Table of Contents

Using this Help 1
What's New in E5071C 8
Precautions 9
Quick Start 21
Measurement 69
Measurement with Option 489
Setting Control Functions 717
Product Information 749
Using Windows 943
Programming 958
Controlling Multiport Test Set 2881

Agilent E5071C ENA Network Analyzer Help

The E5071C help system provides an easy access to the information related to the use of Agilent E5071C ENA Network Analyzer.

You can navigate this help system through the navigation tools provided in the left or you can directly access the chapters:

- Precautions
- Quick Start
- Measurement
- Measurement with Options
- Specifications
- Using this Help (Read me first)
- Setting Control Functions
- Using Windows
- Programming
- Controlling Multiport Test Set
- Product Information
- What's New

To understand the conventions and navigation system used in this help system, refer to Using this Help.

To know about the new features of the E5071C, refer to What's New.

E5071C Firmware Revision	A.09.61
Help Edition No.	7.5
Help Revision Date	10 March, 2011

Agilent Technologies

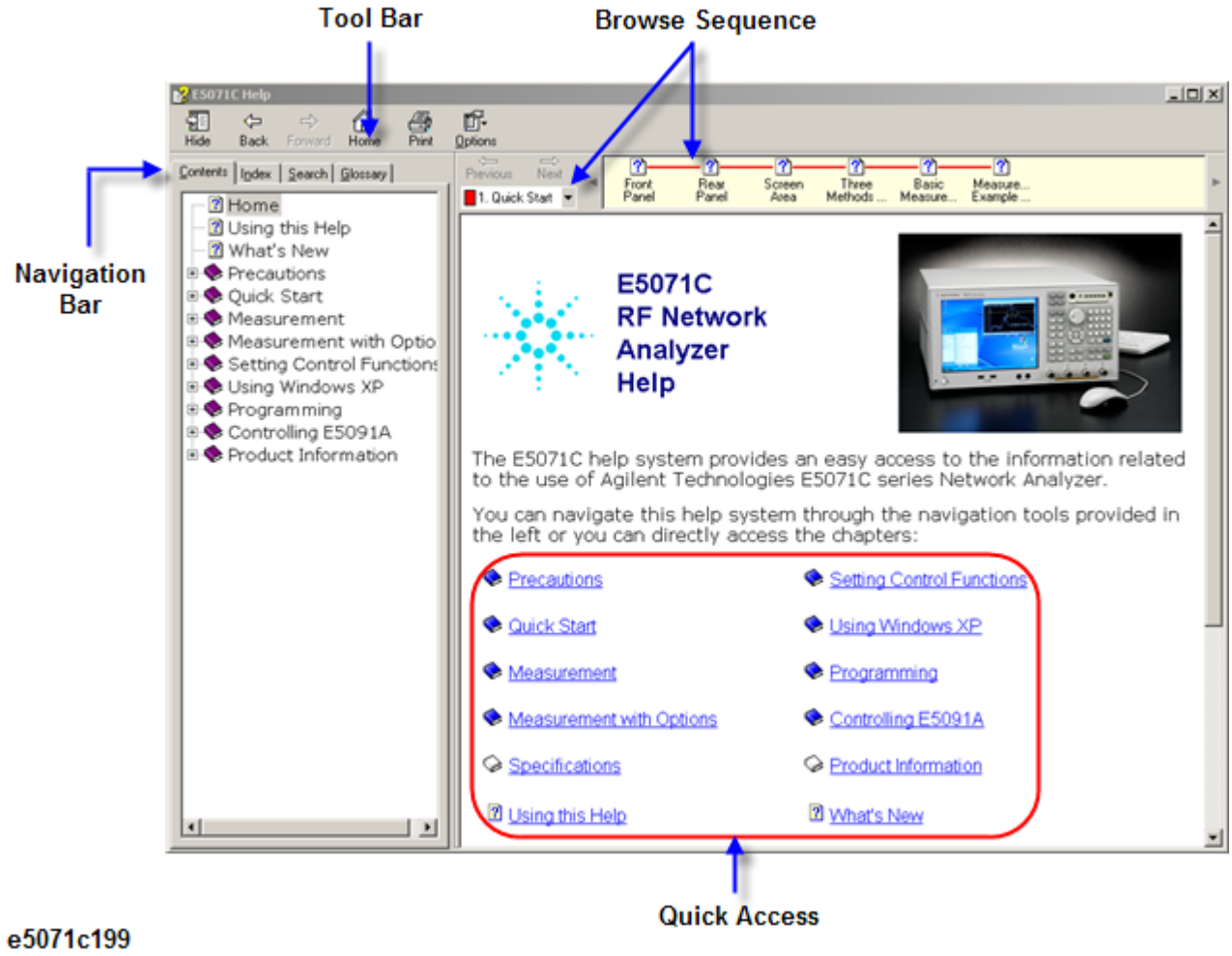
© Copyright 2006-2011 Agilent Technologies

Using this Help

This help provides the user and programming documentation in an searchable electronic format for the E5071C Network Analyzer. This section describes the usage of this help system.

- Opening E5071C Help
- Closing E5071C Help
- Viewing E5071C Help on PC
- Context Sensitive Help
- Switching between E5071C Help and Measurement View
- Navigating E5071C Help
- Conventions used in E5071C Help

E5071C Online Help



e5071c199

Opening E5071C Help

This help system is provided in Microsoft Compressed HTML Help format (ENA_C_Help.chm). This help can be also be viewed on a normal PC.

To open E5071C help, use either of the following methods:

- By pressing the **Help** key located in the **ENTRY** Block.
- By pressing the **F1** key on a keyboard attached with the E5071C.
- By double-clicking **ENA_C_Help.chm** (located in **D:\Agilent\Help**).

Closing E5071C Help

To close the E5071C help, click , located on the top right of the E5071C help viewer.

Viewing E5071C Help on your PC

The E5071C help can be opened and viewed on a normal PC. The help file (**ENA_C_Help.chm**) is located in the E5071C Network Analyzer hard disk at **D:\Agilent\Help**. Copy it to a local hard disk drive on your PC, then double-click it to view.

Context Sensitive Help

Context sensitive help is a special feature of the E5071C help in firmware revision A.08.0x or later. It allows User to get information about the selected softkey by pressing the **Help** key in the E5071C or by pressing **F1** in a keyboard attached to the E5071C or by clicking the help button in a dialog box. This special feature is known as context sensitive help as User receive information quickly depending upon which area the user is working, in the firmware of the E5071C. It provides information relevant to the task that needs to be accomplished and reduces the time to search relevant information required to complete a task.

Switching between E5071C Help and Measurement View

The **Focus** Key, located in the **Entry** Block, can be used to switch between the E5071C Measurement View and Help View.

Navigating E5071C Help

The E5071C help system provides several ways to navigate through the information related to the use of E5071C Network Analyzer. This sections describes the navigation system of the E5071C help which consists of:

- Navigation bar
- Quick Access
- Toolbar
- Browse Sequence

Navigation bar

Navigation bar comprises of tabs related to **Contents**, **Index**, **Search** and **Glossary**. The **Contents** tab contains the main navigational structure of the E5071C help. The **Glossary** tab contains a text explanation of the terms significant to the E5071C Network Analyzer. The **Index** tab is an additional tool that can be used to navigate different topics according to their alphabetical listing. The **Search** tab can be used to search any term/phrase in the E5071C online Help.

Toolbar

The toolbar can be used to navigate through the help. The **Home** option can be used to return to the home page of the E5071C help, containing Quick Access to chapters in the help. The **Back** and **Forward** option can be used to toggle between visited topics. The **Print** option can be used to print the selected, or all topic of the E5071C help.

Browse Sequence

The E5071C help contains pre-defined browse sequence to ease the navigation and to save time needed to perform some commonly used procedures. It also enables users to toggle between steps of the browse sequence.

Quick Access

Quick access, located on the **Home Page** of the E5071C help, is a convenient and quick way to access the contents of the E5071C help.

Conventions used in E5071C Help

Naming Conventions Used in this Help

Convention	Description	Example

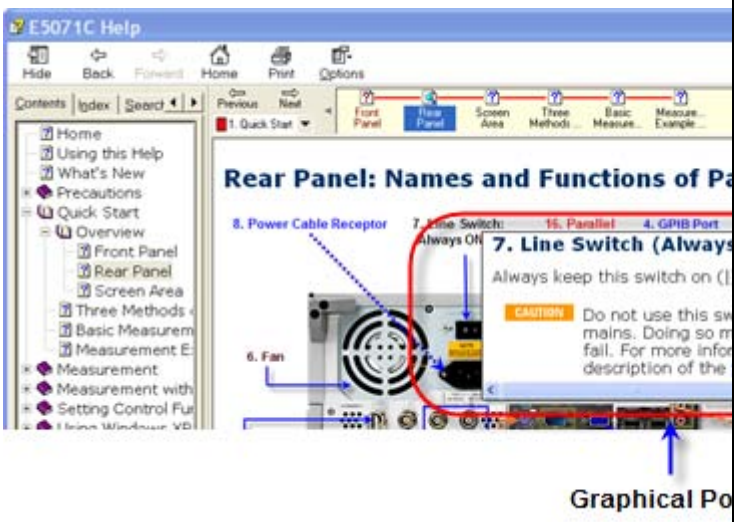
Normal Text	Normal text is displayed in Verdana, 12 pt.	This is normal text.
File Name / Path	File names and path associated with them as displayed as **Bold** in Arial with 12 pt size.	**C:\Documents and Settings\test.txt** **example2.xls**
Hard keys	Hard Keys (Keys located on the Front panel of E5071C) are displayed in Blue color, **Bold** in Verdana with 12 pt size.	**Trace max** **Focus** **Marker**
Softkeys	Also knows as menu keys, are the names of menu appearing in the Firmware (Software) of E5071C and are displayed as **Bold** in Arial with 12 pt size.	**Auto Scale All** **S22** **Target Transition**

Soft Links

Soft links displays a Popup window containing detailed information about a particular topic. There are two type of Popup:

1. Graphical Popup

Graphical Popup windows can contain graphics as well as text.



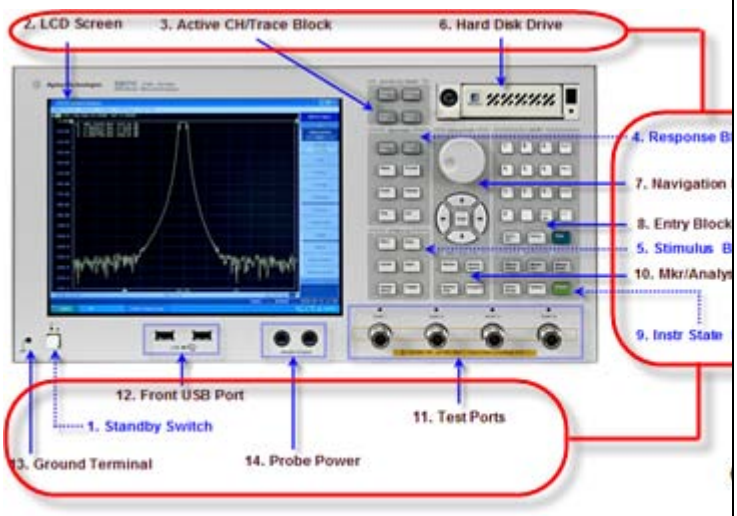
Click here to see an example of a graphical popup window.

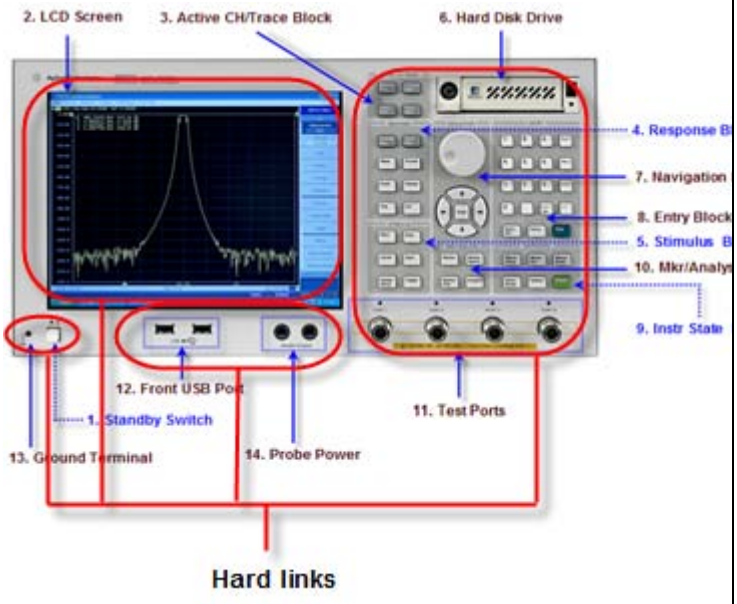
Click here to see an example of a text only popup window.

2. Text Only Popup

Text only Popup windows can contain only text and are displayed with a yellow background.

- In real images of the E5071C, all the text marking are Graphical Popup links.



Hard Links	Hard links shifts the focus of the active window to display more information about a particular topic. • In real images of the E5071C, all the picture part of the components are Hard links. 	
BLOCK Names	E5071C Front Panel is divided into 7 blocks. These blocks are displayed in ALL CAPS, **Green Color**, **Bold** in Arial with 12 pt size, followed by the word 'Block'.	**RESPONSE** Block **ENTRY** Block **STIMULUS** Block
Note	This Note sign denotes important information. It calls attention to a condition that is essential for the user to understand.	• This is a Note.
Caution	This Caution sign denotes a hazard. It calls attention to a procedure, practice, or condition that, if not correctly performed or adhered to, could result in damage to or destruction of part or all of the instrument.	• This is a Caution.
Warning	This warning sign denotes a hazard. It calls attention to a procedure, practice, or condition that, if not correctly performed or adhered to, could result in injury or death to personnel.	• This is a Warning.

What's New in E5071C

The E5071C with A.09.61 comes with the following new features:

- S6P/S8P are supported.

See the revision history for more information.

Precautions

Precautions

- Safety
- Notices
- Installing Software
- Protecting ENA System
- Before Contacting us

Safety

Safety Summary

The following general safety precautions must be observed during all phases of operation, service, and repair of this instrument. Failure to comply with these precautions or with specific WARNINGS elsewhere in this manual may impair the protection provided by the equipment. Such noncompliance would also violate safety standards of design, manufacture, and intended use of the instrument. Agilent Technologies assumes no liability for the customer's failure to comply with these precautions.

- The E5071C complies with INSTALLATION CATEGORY II as well as POLLUTION DEGREE 2 in IEC61010-1. The E5071C is an INDOOR USE product.
- The LEDs in the E5071C are Class 1 in accordance with IEC60825-1, CLASS 1 LED PRODUCT

- Ground the Instrument

  To avoid electric shock, the instrument chassis and cabinet must be grounded with the supplied power cable's grounding prong.
- DO NOT Operate in an Explosive Atmosphere

  Do not operate the instrument in the presence of inflammable gasses or fumes. Operation of any electrical instrument in such an environment clearly constitutes a safety hazard.
- Keep Away from Live Circuits

  Operators must not remove instrument covers. Component replacement and internal adjustments must be made by qualified maintenance personnel. Do not replace components with the power cable connected. Under certain conditions, dangerous voltage levels may remain even after the power cable has been disconnected. To avoid injuries, always disconnect the power and discharge circuits before touching them.
- DO NOT Service or Adjust the Instrument Alone

  Do not attempt internal service or adjustment unless another person, capable of rendering first aid and resuscitation, is present.
- DO NOT Substitute Parts or Modify the Instrument

  To avoid the danger of introducing additional hazards, do not install

substitute parts or perform unauthorized modifications to the instrument. Return the instrument to an Agilent Technologies Sales and Service Office for service and repair to ensure that safety features are maintained in operational condition.

- Dangerous Procedure Warnings

  Warnings, such as the example below, precede potentially dangerous procedures throughout this manual. Instructions contained in the warnings must be followed.

  - Dangerous voltage levels, capable of causing death, are present in this instrument. Use extreme caution when handling, testing, and adjusting this instrument.

Safety Symbols

Instruction Manual symbol: the product is marked with this symbol when it is necessary for the user to refer to the instrument manual.

Alternating current.

Direct current.

On (Supply).

Off (Supply).

In-position of push-button switch.

Out-position of push-button switch.

A chassis terminal; a connection to the instrument's chassis, which includes all exposed metal structure.

Stand-by.

Notices

The information contained in this document is subject to change without notice.

This document contains proprietary information that is protected by copyright. All rights are reserved. No part of this document may be photocopied, reproduced, or translated to another language without the prior written consent of Agilent Technologies.

Microsoft®,MS-DOS®,Windows®,Visual C++®,Visual Basic®,VBA® and Excel® are registered trademarks of Microsoft Corporation.

Java® is registered trademark of Sun Microsystems Corporation.

© Copyright 2002-2010 Agilent Technologies. All rights reserved.

Certification

Agilent Technologies certifies that this product met its published specifications at the time of shipment from the factory. Agilent Technologies further certifies that its calibration measurements are traceable to the United States National Institute of Standards and Technology, to the extent allowed by the Institution's calibration facility or by the calibration facilities of other International Standards Organization members.

Documentation Warranty

The material contained in this document is provided "as is," and is subject to being changed, without notice, in future editions. Further, to the maximum extent permitted by applicable law, Agilent disclaims all warranties, either express or implied with regard to this manual and any information contained herein, including but not limited to the implied warranties of merchantability and fitness for a particular purpose. Agilent shall not be liable for errors or for incidental or consequential damages in connection with the furnishing, use, or performance of this document or any information contained herein. Should Agilent and the user have a separate written agreement with warranty terms covering the material in this document that conflict with these terms, the warranty terms in the separate agreement will control.

Exclusive Remedies

The remedies provided herein are Buyer's sole and exclusive remedies. Agilent Technologies shall not be liable for any direct, indirect, special, incidental, or consequential damages, whether based on contract, tort, or any other legal theory.

Assistance

Product maintenance agreements and other customer assistance agreements are available for Agilent Technologies products.

For any assistance, contact your nearest Agilent Technologies Sales and Service Office.

Sample Program

The customer shall have the personal, non-transferable rights to use, copy, or modify SAMPLE PROGRAMS in this manual for the customer's internal operations. The customer shall use the SAMPLE PROGRAMS solely and exclusively for their own purposes and shall not license, lease, market, or distribute the SAMPLE PROGRAMS or modification of any part thereof.

Agilent Technologies shall not be liable for the quality, performance, or behavior of the SAMPLE PROGRAMS. Agilent Technologies especially disclaims any responsibility for the operation of the SAMPLE PROGRAMS to be uninterrupted or error-free. The SAMPLE PROGRAMS are provided AS IS.

AGILENT TECHNOLOGIES DISCLAIMS ANY IMPLIED WARRANTY OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary right by the SAMPLE PROGRAMS or their use. Agilent Technologies does not warrant that the SAMPLE PROGRAMS are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

VBA Macro

The customer shall have the personal, non-transferable rights to use, copy, or modify the VBA macros for the customer's internal operations.

The customer shall use the VBA macros solely and exclusively for their own purposes and shall not license, lease, market, or distribute the VBA macros or modification of any part thereof.

Agilent Technologies shall not be liable for any infringement of any patent, trademark, copyright, or other proprietary right by the VBA macros or their use. Agilent Technologies does not warrant that the VBA macros are free from infringements of such rights of third parties. However, Agilent Technologies will not knowingly infringe or deliver software that infringes the patent, trademark, copyright, or other proprietary right of a third party.

Installing Software

Updating Pre-Installed Software

Do not update pre-installed software besides Agilent recommends to do so.

Installing User Application Software

Users can install commercial application software for Windows on the E5071C with their responsibilities. Some application software may affect the measurement performance, especially measurement speed.

Protecting ENA System

To protect your ENA, follow the instructions below:

1. Read the warning labels and specifications

Do not exceed the values provided in the specifications guide or as indicated by the yellow warning labels on the front panel of ENA. Refer to the specifications for the conditions required to meet the listed specifications. There will be information regarding ENA settings, and calibration requirements.

2. Do NOT Plug off Power Cable during Shutdown Process

Do Not Plug off Power cable until completely completing the shutdown process.

If you directly interrupt the power supply to the power cable receptacle when the power supply is on, or turn off the Line Switch (Always ON), the shutdown process will not work. This could damage the HDD of the E5071C.

3. Do NOT Modify or Reconfigure the Operating System

The Microsoft Windows operating system has been modified and optimized by Agilent to improve the performance of the ENA.

- Do NOT install a standard version of the Windows operating system on the ENA.
- Do NOT change advanced performance settings or group policies.
- Do NOT add or delete any hard disk drive partitions on the ENA.
- Do NOT delete the Agilent user account.
- Do NOT modify any of the Agilent software registry entries.
- Do NOT change the settings of Standards and Formats in Regional Options and Languages from default setting (English).

4. Install Antivirus Protection

The ENA does NOT have antivirus protection when shipped. Use of an antivirus program is strongly recommended if you connect the ENA to the LAN (Internet).

In addition, the use of a firewall could help to protect the ENA from viruses. However, some firewalls could limit DCOM connectivity of the ENA.

5. Install Windows Critical Updates

The ENA is always shipped with the latest service packs and critical updates that were available at the time when Firmware is updated. We recommend that you maintain the latest available protection for your ENA by automatically accepting and installing the latest critical security patches from the Microsoft Windows Update website: http://windowsupdate.microsoft.com

6. Run Error Check and Disk Defragmenter

When the ENA is shutdown unexpectedly or power is removed without first shutting down, large amounts of Hard Disk Drive space is rendered unusable. If shutdown in this manner enough times, the ENA could become unstable and no longer work.

This HDD space can be recovered by first running Windows Error-checking to find and correct errors on the disk, and then the Disk Defragmenter to recover Hard Disk Drive space. See the Windows Help file for more information about Disk Defragmenter.

To run error check and defragmenter follow the procedure below:

1. On the desktop, double-click **My Computer**.
2. Select **System OS**.
3. Click **File**, then **Properties**.
4. Click the **Tools** tab.

Error-checking

1. Click **Check Now**.
2. Check **Automatically fix file system errors**.
3. Click **Start**.
4. Click **Yes** to run disk check on next restart.
5. Manually restart the ENA. The disk check will run before Windows restarts.

Periodically, check the second box in addition to the first box. The error-checking process takes much longer, but performs a complete check.

Defragmentation

1. Click **Defragment Now...**
2. Click **Defragment** to begin the defragment process.
3. Click **Close** when defragmentation is complete.

7. Precaution for Use of Hard Disk

Do NOT Modify or delete any Files and Folders in Drive other than D and E: Drive.(These two drives, D and E are available to use as Data backup). Doing so will result in malfunctioning of the device. It is

required to execute the System recovery if there are any troubles with the above operations.

- For Option 017 users:
  - Do not remove removable hard disk until completely completing the shutdown process.
  - Do not unlock the removable hard disk key until completely completing the shutdown process.

8. Precaution for RF input connector and cable

- Do not apply DC voltage or current to the test port. Applying DC voltage or current may lead to device failure. In particular, the capacitor might remain charged. Connect the measurement sample (DUT) to the test port (or the test fixture, cables, etc. connected to the test port) after the analyzer has been completely discharged. The maximum DC limit of test port is 35V.

- Do NOT bend, bump or flex any device under test (DUT) connected to the input of ENA (such as filters, couplers etc). This will reduce the amount of strain placed on the input connector and the mounting hardware. Make sure externally connected items are properly supported (not freely suspended) from the input.
- Do NOT bend cables repeatedly, as this may damage the cable instantly. Limit the number of connections and disconnections to reduce wear. Inspect connectors prior to using; look for dirt, nicks, and other signs of damage or wear. A bad connector can ruin a good connector instantly. Clean dirty connectors to prevent poor electrical connections and damage to the connector.

9. Precautions for Electrostatic Discharge (ESD)

ESD can damage or destroy electronic components. Whenever possible, conduct testing at a static-safe workstation. Keep static-generating materials at least one meter away from all components. Before connecting any coaxial cable to an analyzer, momentarily short the center and outer conductors of the cable together.

10. Maintain working environment condition

Control your environment. Maintain temperature & humidity with a satisfactory range within the instruments specification and prevent large fluctuations.

11. Precautions for transportation

- Do NOT pick the instrument with your hand over the front panel. If the instrument slips, damage may occur to the keypad, knob, or input connectors. Lift the Instrument by the handles when transporting.
- Do NOT use styrene pellets as packaging materials as these may cause damage to ENA by generating static electricity.

12. Check for Proper Ventilation

Periodically check and clean the cooling vents of the ENA. Inadequate airflow can result in excessive operating temperatures which can lead to instrument failures. When installing the product in a cabinet, the convection into and out of the instrument must not be restricted.

See the installation guide in detail.

13. Precautions for Proper Grounding

- Proper grounding prevents building-up of static charge which may be harmful to ENA.
- Do NOT defeat the earth-grounding protection by using an extension cable, power cable, or autotransformer without a protective ground conductor.

Before Contacting us

If you encounter the following problems during startup or operation of the E5071C, in which initial registration of the Windows Operating System has been properly performed, execute system recovery and update the Firmware revision.

The system starts up, but the normal measurement screen does not appear

- The system automatically shuts down immediately after the startup, or the startup process stops.
- The measurement screen appears, but "Power on test fail" or "Calibration data lost" is displayed in the instrument message/warning area against a red background in the lower-left part of the screen. The system enters the service mode. (The instrument status bar in the lower-right displays SVC in red).

Unstable Operation

- The system hangs up while the instrument is controlled from VBA or external PCs.
- The blue screen appears and the system hangs up.
- The response is much slower than usual.

When execution of system recovery does not result in normal operation, a failure may have occurred. Contact Agilent customer contacts.

For other problems, refer to Troubleshooting.

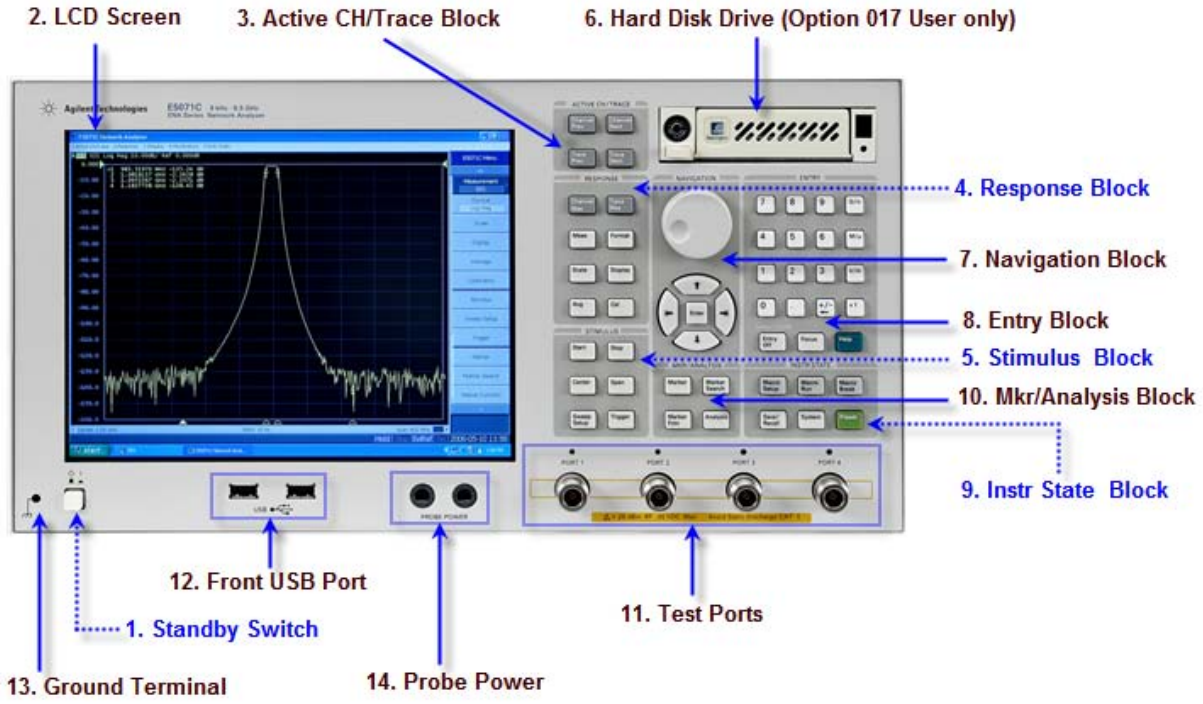
Quick Start

Quick Start helps you to understand the E5071C operation quickly.

- Overview
  - Front Panel
  - Rear Panel
  - Screen Area
- Three Methods of Operation
- Basic Measurement Procedures
- Measurement Example of a Bandpass Filter

Overview

Front Panel: Names and Functions of Parts



1. Standby Switch

- Used for choosing between power-on (|) and standby (O) state of E5071C.

To turn off the power for the E5071C, be sure to follow the steps described below:

1. First, press this standby switch or send a shutdown command from the external controller to activate the shutdown process (the processing of software and hardware necessary to turn off the power supply). This will put the E5071C into the standby state.
2. Next, if necessary, turn off power supply to the Power Cable Receptacle (to LINE) on the rear panel.
    - Under normal use, never directly interrupt the power supply to the power cable receptacle on the rear panel when the power supply is on. Always keep the Line Switch (Always ON) at (|). Never turn it off (O).

      If you directly interrupt the power supply to the power cable receptacle when the power supply is on, or turn off the Line Switch (Always ON), the shutdown process will not work. This

could damage the software and hardware of the E5071C and lead to device failure.

Turning on the power supply after a faulty shutdown may cause the system to start up in a condition called "safe mode." If this occurs, first shut down the system to put it into the standby state and then turn on the power supply again to start up the system in normal mode.

2. LCD Screen

The E5071C is equipped with a 10.4-inch TFT color, touch-sensitive LCD screen for displaying traces, scales, settings, softkeys and other measurement related information. The touch screen LCD allows you to manipulate softkeys by touching the LCD screen directly with a finger. For more on the LCD screen, see Screen Area: Names and Functions of Parts.

- Do not press the surface of the LCD screen with a sharp object (e.g., a nail, pen, or screwdriver). Pressing the surface with a sharp-pointed object will damage the LCD screen surface or cause the screen to fail.
- Valid pixels are 99.998 % and more. Below 0.002 % of fixed points of black, blue, green or red are not regarded as failure.

3. ACTIVE CH/TRACE Block

A group of keys for selecting active channels and traces. For more on the concepts of channels and traces, see Setting Channels and Traces.

Key Name	Description
Channel Next Key	Selects the next channel as the active channel. (Each time the key is pressed causes the active channel to step up from the channel with the currently designated number to one with a larger channel number). A channel must be active before you can define such parameters as the sweep range. To change the settings for a channel, use this key to first make the channel active.
Channel Prev Key	Selects the previous channel as the active channel. (Each time the key is pressed causes the active channel to step down from the channel with the currently designated number to one with a smaller channel number).
Trace Next Key	Selects the next trace as the active trace. (Each time the key is pressed causes the active trace to step up from the trace with the currently designated number to one with a larger channel number.) A trace must be active before you can define measurement parameters and other settings. To change the settings for a trace, use this key to first make the trace active.

Trace Prev Key	Selects the previous trace as the active trace. (Each time the key is pressed causes the active trace to step down from the trace with the currently designated number to one with a smaller trace number.)

4. RESPONSE Block

A group of keys used mainly for setting up response measurements on the E5071C.

Key Name	Description
Channel Max Key	Changes between normal and maximum display of the active channel window. In normal display, all of the defined channel windows (both active and non-active) are displayed in split views on the screen. In maximum display, only the active channel window is displayed over the entire area, with non-active windows not displayed. To maximize the active channel, double-click the channel window frame. Measurements are also carried out on the non-active channels that are not displayed.
Meas Key	Displays the Measurement Menu on the right side of the screen. Manipulating the Measurement Menu enables you to specify the measurement parameters (types of S-parameters) for each trace.
Format Key	Displays the Format Menu on the right side of the screen. Manipulating the Format Menu enables you to specify the data format (data transformation and graph formats) for each trace.
Scale Key	Displays the Scale Menu on the right side of the screen. Manipulating the Scale Menu enables you to specify the scale for displaying a trace (magnitude per division, value of the reference line, etc.) for each trace. You can also specify the electrical delay and phase offset for each trace.
Display Key	Displays the Display Menu on the right side of the screen. Manipulating the Display Menu enables you to specify the number of channels and the channel window array, the number and arrangement of traces, the setup for data math, etc.
Avg Key	Displays the Average Menu on the right side of the screen. Manipulating the Average Menu enables you to define the averaging, smoothing, and IF bandwidth.
Cal Key	Displays the Calibration Menu on the right side of the screen. Manipulating the Calibration Menu enables you to turn the calibration and error correction on/off and change definitions for calibration kits.

5. STIMULUS Block

A group of keys for defining the stimulus values (signal sources and triggers).

Key Name	Description

Start Key	Displays the data entry bar for specifying the start value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu for specifying the sweep range on the right side of the screen.)
Stop Key	Displays the data entry bar for specifying the stop value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu in the same way as **Start** Key.)
Center Key	Displays the data entry bar for specifying the center value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu in the same way as **Start** Key.)
Span Key	Displays the data entry bar for specifying the span value of the sweep range in the upper part of the screen. (It also displays the Stimulus Menu in the same way as **Start** Key.)
Sweep Setup Key	Displays the Sweep Setup Menu on the right side of the screen. Manipulating the Sweep Setup Menu enables you to specify the signal source power level, sweep time, number of points, sweep type, etc.
Trigger Key	Displays the Trigger Menu on the right side of the screen. Manipulating the Trigger Menu enables you to specify the trigger mode and trigger source. You can specify the trigger mode for each channel.

6. Hard Disk Drive (Option 017 User only)

The E5071C comes with a removable hard disk drive. For removal procedure, see Removing/Mounting Removable Hard Disk.

7. NAVIGATION Block

The keys and knob in the **NAVIGATION** Block are used to navigate between softkey menus, tables (limit table, segment table, etc.), or selected (highlighted) areas in a dialog box as well as to change a numeric value in the data entry area by stepping up or down. When selecting one of two or more objects (softkey menus, data entry areas, etc.) to manipulate with the **NAVIGATION** Block keys displayed on the screen, first press the **Focus** key in the ENTRY Block to select the object to be manipulated (placing focus on the object) and then manipulate the **NAVIGATION** Block keys (knob) to move among selected (highlighted) objects or change numeric values.

The following descriptions show how the **NAVIGATION** Block keys work both when the focus is on a softkey menu and when the focus is on the data entry area. For more on manipulating tables and dialog boxes, refer to the manipulation procedure for each of these functions.

When the focus is on a softkey menu (softkey menu is selected)

When the focus is placed on a softkey menu (the menu title area in the uppermost part is displayed in blue), the **NAVIGATION** Block keys work as described below.

Key Name	Description
Knob (turned clockwise or counterclockwise)	Moves the softkey selection (highlighted display) up or down.
Keys	Moves the softkey selection (highlighted display) up or down.
Key	Displays the softkey menu one layer above.
Key	Displays the softkey menu one layer below.
Knob or Enter key (pressed)	Executes the function of the selected softkey.

After pressing the data entry softkey, the focus automatically moves to the data entry area.

When the focus is on the data entry area (data entry area is selected)

When the focus is placed on the data entry area (the data entry bar is displayed in blue), the **NAVIGATION** Block keys work as described below.

Key Name	Description
Knob (turned clockwise or counterclockwise)	Increases or decreases the numeric value in the data entry area in small steps.
Keys	Increases or decreases the numeric value in the data entry area in large steps.
Keys	Moves the cursor (|) in the data entry area laterally back and forth. Use it together with the ENTRY Block keys to change data one character at a time.
Knob or **Enter** key (pressed)	Finishes the entry in the data entry area and moves the focus to the softkey menu.

8. ENTRY Block

A group of keys used for entering numeric data is provided on the front panel of the E5071C.

Key Name	Description
0, 1, 2, 3 ..... 9 , . Keys	Type numeric characters or a decimal point at the position of the cursor in the data entry area.

(numeric keys)	
+/- & ← Key	**+/-** alternately changes the sign (+, -) of a numeric value in the data entry area. ← is the backspace key. **+/-** key works only when there are no numerical values in the entry area. When a numerical character is entered in the entry area, the **+/-** key transforms into (←) backspace key.
G/n, M/u, k/m, x1 Keys	Adds a prefix to the numeric data typed by using the numeric key and **+/-** key and then enters that data. One of the two prefixes written on the surface of the key is automatically selected depending on the parameter to be entered. **x1** is entered without a prefix.
Softkey On/Off , Entry Off Key	Turns off the data entry bar if it is displayed. If the dialog box is displayed, cancels the entry and closes the dialog box. If the data entry bar and dialog box are not displayed, turns the softkey menu display on/off.
Help Key	Displays online help for E5071C.
Focus Key	Changes the selection (focus) among the objects to be manipulated by the **NAVIGATION** Block keys and **ENTRY** Block keys. The objects to be manipulated by the **NAVIGATION** Block keys and **ENTRY** Block keys include softkey menus, data entry areas, tables (e.g., segment tables, limit tables, and marker tables), and dialog boxes. When two or more of these are displayed on the screen and need selecting, use this key to change the selection (focus) among the objects to be manipulated. When a softkey menu is selected, the menu name area at the top of the menu is displayed in blue. When a data entry area is selected, the data entry bar is displayed in blue. When a table is selected, the frame of the table window is displayed in light gray. While a dialog box is displayed, the focus is fixed on the dialog box and cannot be changed.

9. INSTR STATE Block

A group of keys related to the macro function, store and call function, control/management function, and the presetting of the E5071C (returning it to the preset state).

Key Name	Description
Macro Setup Key	Displays the Macro Setup Menu on the right side of the screen. Manipulating the Macro Setup Menu enables you to start up the VBA editor or to create, call, or store a VBA project.
Macro Run Key	Executes a VBA procedure called "main" that has a VBA module named Module1.
Macro Break	Stops the VBA procedure being executed.

Key	
Save/Recall Key	Displays the Save/Recall Menu on the right side of the screen. Manipulating the Save/Recall Menu enables you to store the setup conditions to or read from the storage devices, calibration data, and trace data of the analyzer.
Capture/System Key	First, temporarily saves the data for the image displayed on the LCD screen the moment this key is pressed to the internal memory (clipboard). Immediately after that, displays the System Menu on the right side of the screen. Manipulating the System Menu enables you to define the setup for the limit test and then execute it or to define the setup for the control and management of the analyzer. Using the **Dump Screen Image** option enables you to store the image data in the clipboard to a file on the storage devices. Also, using the **Print** option in the System menu enables you to print the image data in the clipboard to a printer.
Preset Key	Displays the Preset Menu on the right side of the screen. Clicking **OK** in the Preset Menu enables you to return the analyzer to the initial setup state, called the preset setup.

10. MKR/ANALYSIS Block

A group of keys used for analyzing the measurement results by using the markers, fixture simulator, etc. For more on the functions of the keys in the **MKR/ANALYSIS** Block, see Overview of Functions.

Key Name	Description
Marker Key	Displays the Marker Menu on the right side of the screen. Manipulating the Marker Menu enables you to turn the markers on/off and move them by entering stimulus values. You can place up to 10 markers on each trace.
Marker Search Key	Displays the Marker Search Menu on the right side of the screen. Manipulating the Marker Search Menu enables you to move a marker to a specific point (maximum, minimum, peak, and a point with a target value) on a trace. You can also find the bandwidth parameters (up to six) and display them.
Marker Fctn Key	Displays the Marker Function Menu on the right side of the screen. Manipulating the Marker Function Menu enables you to not only specify the marker sweep range and the coupling of markers on a channel but also to display statistics data on traces.
Analysis Key	Displays the Analysis Menu on the right side of the screen. Manipulating the Analysis Menu enables you to use the analytical function called the fixture simulator.

11. Test Port

The port to which the DUT is connected. Number of ports depends on the option. While signals are being output from a test port, the yellow LED above the test port is lit.

Test Port for options 2D5, 4D5, 2K5 and 4K5

Options	Connector type
230, 430, 235, 435, 240, 440, 245, 445, 260, 460, 265, 465, 280, 480, 285, 485	50 ohm, N-type, female
2D5, 4D5, 2K5 and 4K5	50 ohm, 3.5 mm, male

- Do not apply DC voltage or current to the test port. Applying DC voltage or current may lead to device failure. In particular, the capacitor might remain charged. Connect the measurement sample (DUT) to the test port (or the test fixture, cables, etc. connected to the test port) after the analyzer has been completely discharged. The maximum DC limit of test port is 35V.

The test ports comply with Installation Category I of IEC 61010-1.

12. Front USB Port

Two USB (Universal Serial Bus) ports are provided that can be used for connecting to ECal (Electronic Calibration) module, USB, Multiport test set or a printer. Connecting a designated ECal module to this port enables ECal measurements to be taken. Connecting a compatible printer to this port enables screen information on the E5071C to be printed. For more on executing ECal measurements, see Calibration, and for printing, see Printing Displayed Screen.

- The specifications of this port are identical to those of the Rear USB port.

13. Ground Terminal

Connected to the chassis of the E5071C, a ground terminal is provided with the E5071C. You can connect a banana-type plug to this terminal for grounding.

14. Probe Power

The E5071C comes with two ports that can be used to provide power to external probes. See the Data sheet for the voltage and maximum current.

Rear Panel: Names and Functions of Parts

1. Handler I/O Port

The terminal to which an automatic machine (handler) used on a production line is connected. See Handler I/O Port.

Connector type: 36-pin Ribbon (Centronics) connector

2. Ethernet Port

A terminal for connecting the E5071C to a LAN (Local Area Network). Connecting this instrument to a LAN enables you to access the hard disk drive of this instrument from an external PC or to control this instrument by using SICL-LAN or telnet.

Specification	Value
Connector type	8-pin RJ-45 connector
Base standard	10Base-T/100Base-TX Ethernet (automatic data rate selection)

3. External Monitor Output Port (Video)

A terminal to which an external color monitor (display device) can be connected. By connecting a color monitor to this terminal, the same information shown on the LCD screen of the main body can be displayed on an external color monitor.

Connector type: 15-pin VGA connector, female

4. GPIB Connector

The connection of an external controller through General Purpose Interface Bus (GPIB) connector allows you to configure an automatic measurement system.

This GPIB connector is used only for controlling the E5071C from an external controller. Use USB/GPIB interface to control other devices from the E5071C. You cannot control other devices from the E5071C through this GPIB connector.

5. External Trigger Input Port (Ext Trig)

A connector to which external trigger signals are input. This connector detects the downward transition from the HIGH state in TTL signals as the trigger signal. To use this connector to generate a trigger, you must set

the trigger source to the "external" side (key operation: **Trigger** > **Trigger Source** > **External**).

Connector type: BNC connector, female

6. Fan

The cooling fan for controlling the temperature inside the E5071C. This fan exhausts heated air from inside the analyzer to the outside.

7. Line Switch (Always ON)

Always keep this switch on (|).

- Do not use this switch to turn off (O) the mains. Doing so may cause the analyzer to fail. For more information, see the description of the Standby Switch.

8. Power Cable Receptacle (to LINE)

The receptacle (outlet) to which the power cable is connected.

- To connect the device to a power source (outlet), use the supplied three-prong power cable with a ground conductor. The plug attached to the power cable (on the power outlet side or device side of the cable) serves as the disconnecting device (device that cuts off power supply) of the E5071C. When the power supply must be cut off to avoid such danger as electric shock, pull out the power cable plug (on the power outlet side or device side of the cable). For the procedure for turning off the mains in normal use, see the description in Standby Switch.

For more on the power supply, see the Installation Guide.

9. High Stability Frequency Reference Output Port (Ref Oven, Option 1E5 only)

When Option 1E5 (high stability frequency reference) is installed, the reference signal is output from this connector.

Specification	Value
Connector type	BNC connector, female
Output signal (Typical)	10 MHz ± 1 ppm, 0 dBm minimum

- When Option 1E5 (high stability frequency reference) is installed, connect this connector to the External Reference Signal Input Connector (Ref In) by using the BNC(m)-BNC(m) cable included with the option.

10. External Reference Signal Input Port (Ref In)

The reference signal input connector for phase-locking the measurement signal from the E5071C to the external frequency reference signal.

Inputting the reference signal to this connector improves the accuracy and frequency stability of the measurement signal from the E5071C.

Specification	Value
Connector type	BNC connector, female
Input signal (Typical)	10 MHz ± 10 ppm, -3 dBm to +10 dBm

- When the frequency reference signal is input to this connector, the measurement signal from the E5071C is automatically phase-locked to the reference signal. When an input signal is not present, the frequency reference signal inside the E5071C is automatically used. The ExtRef on the instrument status bar is displayed in blue when the system is phase-locked to the external reference signal and in gray when not phase-locked.

  When using Option 1E5 (high stability frequency reference), connect this connector to the High Stability Frequency Reference Output Connector (Ref Oven, Option 1E5 only) by using the BNC(m)-BNC(m) cable included with the option.

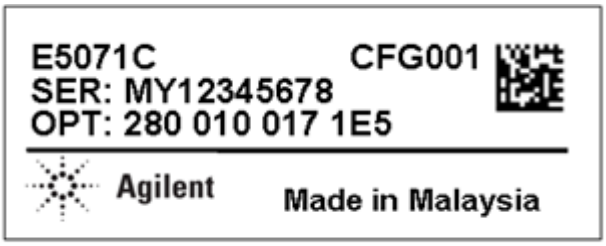
11. Internal Reference Signal Output Port (Ref Out)

A connector for outputting the internal frequency reference signal from the E5071C. By connecting this output connector to the external reference signal input connector of another device, the device can be phase-locked to the internal reference signal of the E5071C and used under this condition.

Specification	Value
Connector type	BNC connector, female
Output signal (Typical)	10 MHz ± 5 ppm, 0 dBm ± 3 dBm
Output impedance (Typical)	50 ohm

12. Serial Number Plate

The label showing the product number, serial number and the installed option number. The accessary and system rack options are not listed on this label. (CFGxxx or ATOxxx in the first line is for Agilent Use Only.)



e5071c501

13. Certificate of Authenticity Label

The label showing the information of the "Certificate of Authenticity."

14. Rear USB Port

Four USB (Universal Serial Bus) ports are provided that can be used for connecting to ECal (Electronic Calibration) module, USB, Multiport test set or a printer. Connecting a designated ECal module to this port enables ECal measurements to be taken. Connecting a compatible printer to this port enables screen information on the E5071C to be printed.

- The specifications of this port are identical to the Front USB Port.

15. Reserved Port (Reserved)

These ports are not available for use (reserved ports).

16. Reserved Port (Reserved)

This port is not available for use (reserved port).

17. Mini-DIN Keyboard Port

A port to which a mini-DIN type keyboard is connected. The keyboard can be used to edit VBA programs inside the E5071C or to enter file names. Since the arrow keys and numeric keys on the keyboard work in the same way as the arrow keys and numeric keys on the front panel of the E5071C, you can use it instead of front panel operation.

- Though E5071C comes with a Mini-DIN Keyboard port, use of a USB Keyboard is recommended.

18. Mini-DIN Mouse Port

The port to which a mini-DIN type mouse is connected. Using a mouse enables you to more efficiently perform the operations of menu bars, softkeys, and dialog boxes as well as selecting an *active channel* or an active trace. The mouse also enables you to move a marker or the scale reference line by using drag-and-drop operations.

- Though E5071C comes with a Mini-DIN Mouse port, use of a USB Mouse is recommended.

19. USB (USBTMC) Interface Port

Through this port, you can control the E5071C from external controllers. For more information on the measurement system using the USB port, see the USB Remote Control System.

Specification	Value
Connector type	Universal serial bus (USB) jack, type B (4 contact positions), Female
Compliance Standards	USBTMC-USB488 and USB2.0

20. External Trigger Output Port

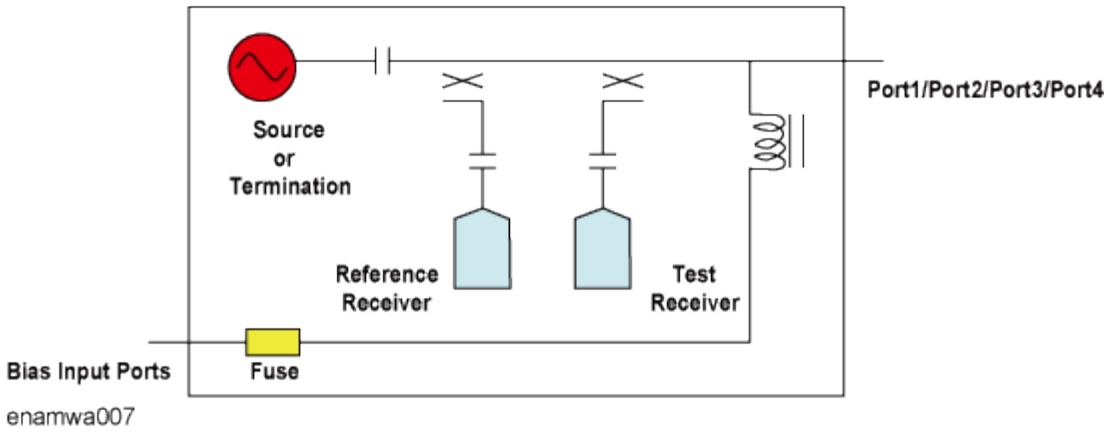
The External Trigger Output Port can output the pulse with the specified polarity either before or after the measurement of each point.

- The External Trigger Output Port can safely handle a maximum Output Current of 50mA.

Symbol	Parameter	Typical Value	Unit	Condition
	Pulse Width	1	μsec	
VOH	HIGH Level Output Voltage	5	Volt	I_{out}=-50μA
VOL	LOW Level Output Voltage	0	Volt	I_{out}=50μA

21. Bias Input Ports

This BNC female connector allows external bias to be applied at the test ports.



- Do not apply DC voltage exceeding 35 Volt.
- Bias Input Port is an optional component of E5071C. Valid for options: 235, 435, 285, 485.

22. Fuse for Bias Input Ports

Fuse for Bias Input Ports is an optional component of E5071C. Valid for options: 235, 435, 285, 485. The fuse holder has been changed. The current fuse case has the black color. There is no functional differences.

Fuse holder in Black color

e5071c493

Specification	Description
Fuse Rating	0.5A/125V
Part Number	2110-0046

23. Aux Input Ports

The Aux Input Ports are used to input DC signal for DC signal measurement.

Screen Area

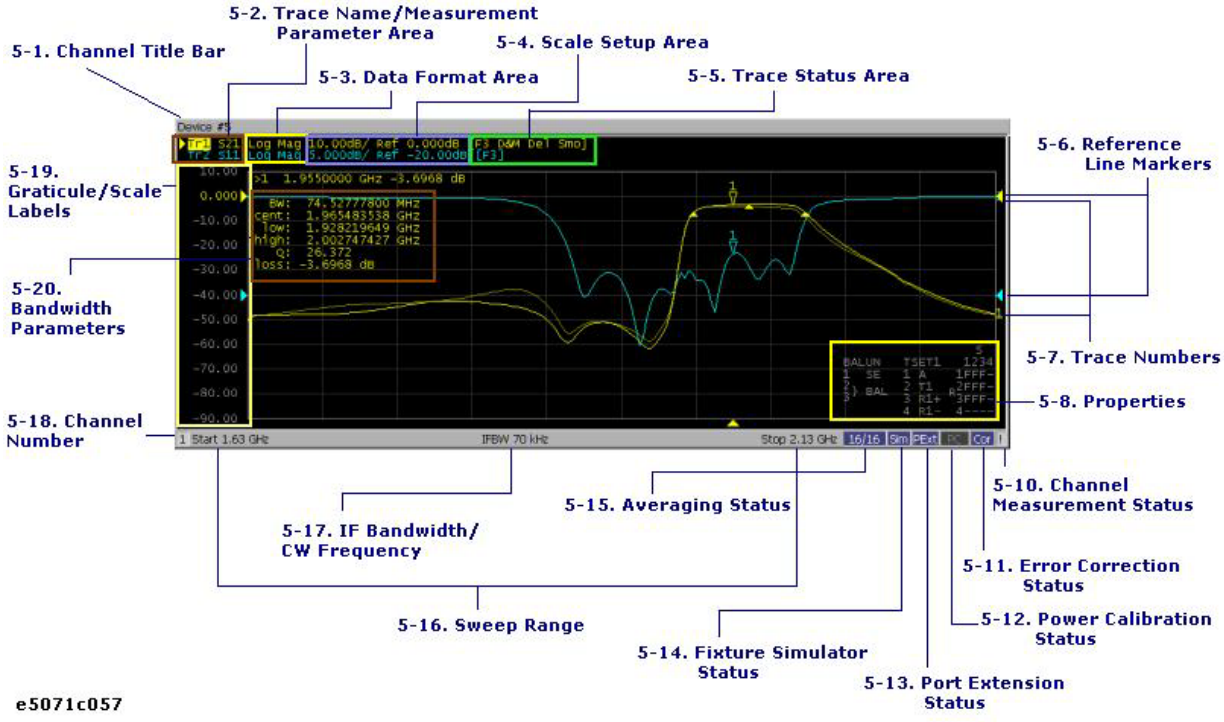
Screen Area: Names and Functions of Parts

Channel Window

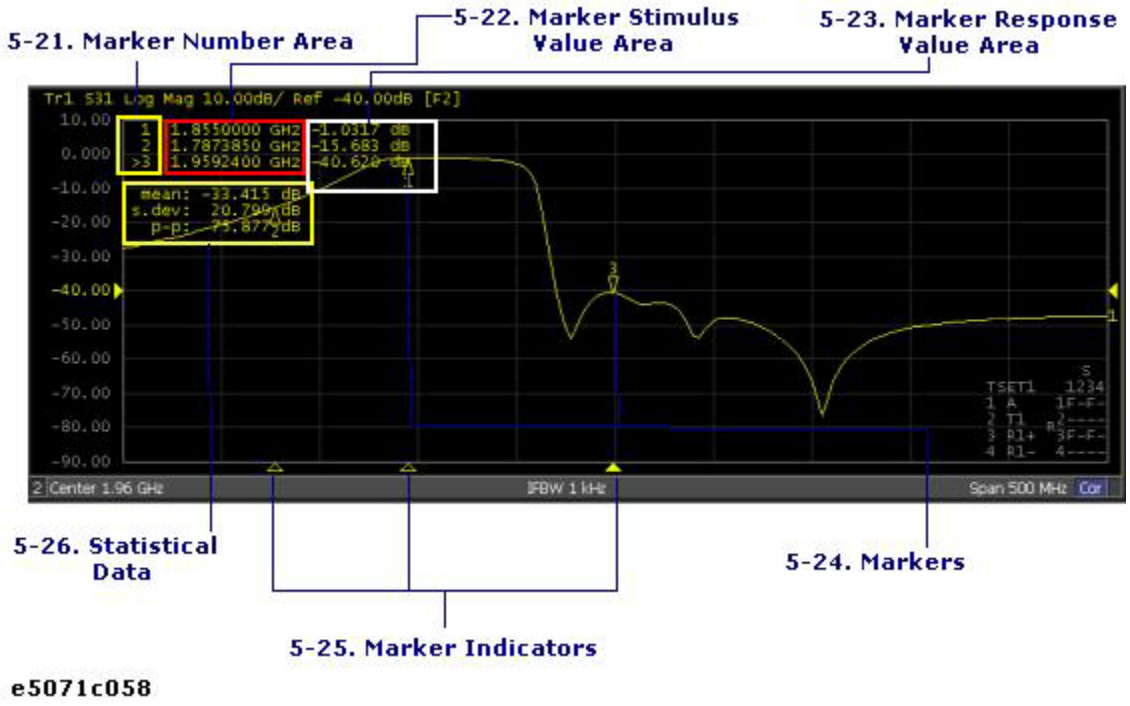
Windows for displaying traces. Because a channel corresponds to a window, it is called a channel window. When the outer frame of a channel window is displayed in light gray, the channel is the active channel (the channel for which setup is being performed). In the following figure, Channel 1 (the upper window) is the active channel. To make a channel active, use **Channel Next** or **Channel Previous** key. Clicking inside a channel window will also make the channel active.

- Channel 1 Window and Channel 2 Window describes different measurement parameters available in the channel measurement window. The measurement parameters described in the Channel 1 and 2 Window correspond to the same channel measurement window and are displayed in separate windows for ease of read.

Channel 1 Window



Channel 2 Window



5-1. Channel Title Bar

You can assign a title to each channel and have the title displayed on the bar. For more on setting up a channel title bar, see Labeling a window.

5-2. Trace Name/Measurement Parameter

The names of the traces (Tr1 through Tr9)on the channel and their measurement parameters are displayed here. ▶ to the right of the trace name indicates the active trace (the trace for which setup is being performed). To make a trace active, use **Trace Next** Key or **Trace Prev** Key. Clicking the line where the trace name is placed (the mouse pointer changes from to) also makes a trace active.

5-3. Data Format

The data format of each trace is displayed here. For more on setting up data formats, see Selecting a Data Format.

5-4. Scale Settings

The scale setting for each trace is displayed here. This example shows that "10.00dB/" corresponds to 10 dB per division. "Ref 0.000dB" shows that the value of the reference line is 0 dB. For more on setting scales, see Setting the Scales.

5-5. Trace Status Area

The setup for each trace is displayed here.

Trace status display

Classification	Contents inside [ ]	Meaning
Error correction (In the option TDR, this information should be ignored.)	**RO**	Error correction: ON (OPEN (n) response calibration)
	RS	Error correction: ON (SHORT (n) response calibration)
	RT	Error correction: ON (THRU (n) response calibration)
	ER	Error correction: ON (Enhanced response calibration)
	F1	Error correction: ON (1-port calibration)
	F2	Error correction: ON (Full 2-port calibration / 2-port TRL calibration)
	F3	Error correction: ON (Full 3-port calibration / 3-port TRL calibration)
	F4	Error correction: ON (Full 4-port calibration / 4-port TRL calibration)
Turning on / off traces	**Nothing**	Data trace: ON, Memory trace: OFF
	M	Data trace: OFF, Memory trace: ON
	D&M	Data trace: ON, Memory trace: ON
	off	Data trace: OFF, Memory trace: OFF
Performing data math	**D+M** (**D+M&M**)	Execution of Data+Mem math

When a memory trace is ON, see the contents inside ()	**D- M** (**D- M&M**)	Execution of Data- Mem math
	D*M (**D*M&M**)	Execution of Data*Mem math
	D/M (**D/M&M**)	Execution of Data/Mem math
Electrical delay	**Del**	A numeric value other than 0 (zero) is specified as the electrical delay or phase offset.
Smoothing	**Smo**	Smoothing: ON
Gating	**Gat**	Gating: ON
Parameter conversion	**Zr**	Conversion: ON (Impedance: Reflection measurement)
	Zt	Conversion: ON (Impedance: Transmission measurement)
	Ztsh	Conversion: ON (Impedance: Transmission-Shunt measurement)
	Yr	Conversion: ON (Admittance: Reflection measurement)
	Yt	Conversion: ON (Admittance: Transmission measurement)
	Ytsh	Conversion: ON (Admittance: Transmission-Shunt measurement)
	1/S	Conversion: ON (Inverse S-parameter)
	Conj	Conversion: ON (Conjugation)
Power correction	**PC**	Error correction: ON (Power calibration)
Receiver correction	**RC**	Error correction: ON

		(Receiver calibration)
Scalar-mixer correction	**Cor**	Error correction: ON (Scalar-mixer calibration)
Equation Editor	**Equ**	Equation Editor: ON

5-6. Reference Line Indicators

The indicators that indicate the position of the reference line for the Y-axis scale in the rectangular display format. One indicator is to the right and the other is to the left of the scale (▶and ◀). To enter a numeric value for the position of the reference line, open the data entry bar using the keys: **Scale** > **Reference Position**. You can also move the position of the reference line by placing the mouse pointer on either of the two reference line indicators (the pointer changes from to .), moving the indicator vertically with the left mouse button kept pressed, and then releasing the button at the desired location (i.e., a drag-and-drop operation).

5-7. Trace Number

In the rectangular display format, the trace number is displayed in the same color as the trace at the right end of each trace.

5-8. Properties

e5071c444

Displays the following properties.

Property Name	Description
Calibration Property	Displays the status of the obtained calibration coefficients on the channel. For details, see Acquisition status of

	calibration coefficient for each channel.
E5091A Property	Displays the assignment information of the test ports on the channel. For details, see Displaying the multiport test set properties.
Balanced Measurement Topology Property	Displays the topology for balanced measurement on the channel. For details, see Checking device type and port assignment.

5-9. Channel Status Bar

The status of each channel is displayed here (see parts 5-10 through 5-16).

5-10. Channel Measurement Status

Displays the update status of traces on the channel.

!	Measurement in progress. When the sweep time exceeds 1.5 seconds, ↑ is displayed at the point on the trace.
#	Invalid traces. The measurement conditions have changed, but the traces on the channel currently displayed have not been updated to match the new conditions.
(No display)	The measurement has not been executed.

5-11. Error Correction Status

Displays the execution status of error correction on the channel. For details, see Execution status of error correction for each channel.

5-12. Power Calibration Status

Displays the execution status of power level error correction on the channel. For details, see Turning Power Level Error Correction ON/OFF.

5-13. Port Extension Status

Shows whether the port extension is turned ON or OFF.

PExt (displayed in blue)	Port extension: ON
(not displayed)	Port extension: OFF

5-14. Fixture Simulator Status

Shows whether the fixture simulator is turned ON or OFF.

Sim (displayed in blue)	Fixture simulator: ON
(not displayed)	Fixture simulator: OFF

5-15. Averaging Status

Displays the averaging factor and averaging count when averaging is turned on.

n/m (displayed in blue)	Averaging: ON (m: averaging factor; n: averaging count)
(not displayed)	Averaging: OFF

5-16. Sweep Range

Indicates the sweep range by using the start/stop or center/span.

5-17. IF Bandwidth/CW Frequency

Indicates the IF bandwidth when the sweep type is linear/log frequency or the CW frequency when the sweep type is power.

5-18. Channel Number

Indicates the channel number.

5-19. Graticule Labels

Y-axis divisions in the rectangular display format. When traces in the rectangular display format are overlaid, the Y-axis divisions for the active trace are displayed. The value of the reference line (the division line between ▶ and ◀) is entered numerically by opening the data entry bar using the keys: **Scale** Key + Reference Value. You can change values of the reference line at one-division intervals by placing the mouse pointer in the area of the graticule label (the pointer changes from to), moving the pointer vertically with the left mouse button pressed, and then releasing the button at the desired location.

5-20. Bandwidth Parameters

Turning on the bandwidth search function displays the bandwidth parameters here. For more on the bandwidth search function, see Determining the Bandwidth of the Trace (Bandwidth Search).

5-21. Marker Numbers

The marker values are displayed in a list at positions 5-21, 5-22, and 5-23. Position 5-21 displays the marker numbers. For the *active marker* (the one for which setup and analysis are being performed), > is displayed to the left of the marker number. For the reference marker, △ is displayed instead of the marker number.

5-22. Marker Stimulus Values

The marker stimulus value for each marker (the frequency/power level at the marker point) is displayed here.

5-23. Marker Response Values

The marker response value for each marker (the measurement value at the marker point) is displayed here. Two (or three) response values are displayed for data in Smith chart or polar display format.

5-24. Markers

The markers used for reading values on a trace. Up to 10 markers can be displayed for each trace.

n ▽	Active marker (the one for which setup and analysis are being performed)
△ n	Non-active marker

Here, "n" denotes a marker number. For the reference marker, however, nothing is displayed at the location of n. Clicking the marker or one of the Marker Indicators makes the marker active.

5-25. Marker Indicators

Indicates the positions of markers on the stimulus axis.

▲	Active marker indicator
△	Non-active marker indicator

You can also move a marker to the desired position by placing the mouse pointer on the marker indicator or position of the marker itself (the pointer changes from to), moving the indicator vertically with the left mouse button pressed, and then releasing the button at the desired location.

5-26. Statistics Data

Turning on the statistics data function displays statistics data here. For more on the statistics data function, see Determining the Mean, Standard Deviation, and p-p of the Trace

Data Entry Bar

Used to enter numeric data into the E5071C. Press a hardkey or softkey to enter data, and the data entry bar will appear at the top of the screen. To assign a title to a channel window, an entry bar that allows you to enter letters and symbols by using the front panel keys or mouse is displayed instead.

Data entry bar

e5070bue054

- To manipulate the data entry bar by using the front panel keys, the data entry bar must be selected as the object to manipulate (with the focus placed on it). When the focus is placed on the data entry bar, the entire bar is displayed in blue. Pressing or clicking **Focus** Key in the ENTRY Block enables you to move the focus to the desired object.

2-1. Parameter Name

Displays the name of the parameter for which data will be entered.

2-2. Data Entry Area

When the data entry bar is displayed for the first time, the current settings are displayed on it. You can change numeric values by typing from the keyboard or in the ENTRY block on the front panel.

You can hide the frequency information in order to ensure its confidentiality or for other reasons. For detailed information, see Hiding Softkey's Frequency Information.

2-3. Step Button (Small)

Increases or decreases the numeric value in the data entry area in small steps. Use the mouse to manipulate this button.

2-4. Step Button (Large)

Increases or decreases the numeric value in the data entry area in large steps. Use the mouse to manipulate this button.

2-5. Enter Button

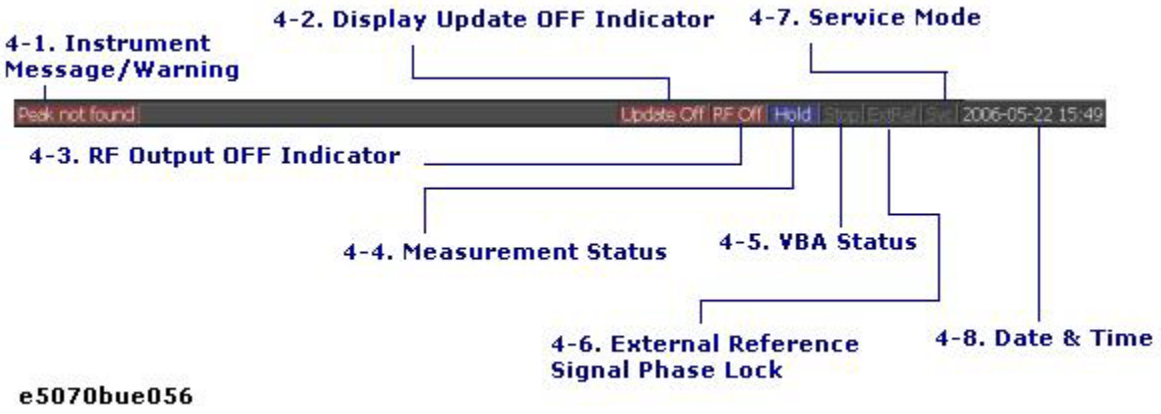
After typing numeric values in the data entry area by using the keyboard or the numeric keys in the **ENTRY** Block on the front panel, press this button to finish the entry. Use the mouse to manipulate this button.

2-6. Close Button

Closes the data entry area (turns off the display). Use mouse to manipulate this button.

Instrument Status Bar

The instrument status bar displays the status of the entire instrument.



4-1. Instrument Message/Warning

Displays instrument messages and warnings. Instrument messages are displayed in gray and warnings in red. For the meanings of the instrument messages and warnings, see Troubleshooting.

4-2. Display Update OFF Indicator

When updating of information displayed on the LCD screen is turned off, this indicator is displayed.

4-3. RF Output OFF Indicator

When RF signal output is turned off, this indicator is displayed.

4-4. Measurement Status

Displays the measurement status of the E5071C.

Value	Description
Setup	Setup for measurement in progress
Hold	Measurement on hold (idling)
Init	Measurement being initialized
Man	The trigger source is set to "Manual" and waiting for trigger.
Ext	The trigger source is set to "External" and waiting for trigger.
Bus	The trigger source is set to "Bus" and waiting for trigger.

Meas	A measurement is in progress.

4-5. VBA Status

Displays the state of the execution of the VBA program in the E5071C.

Value	Description
Run	A VBA program is currently running.
Stop	A VBA program has stopped.

4-6. External Reference Signal Phase Lock

When the frequency reference signal is input to the External Reference Signal Input Connector (Ref In) on the rear panel and the measurement signal of the E5071C is phase-locked to the reference signal, ExtRef is displayed in blue.

Value	Description
ExtRef (displayed in blue)	Measurement signal is phase-locked to the external reference signal.
ExtRef (displayed in gray)	Measurement signal is not phase-locked to the external reference signal.

- Even when the High Stability Frequency Reference Output Connector (Ref Oven, Option 1E5 only) and External Reference Signal Input Connector (Ref In) are connected, phase-locking may not occur immediately after power-on in a low-temperature environment. (The "ExtRef" display remains gray, not blue.) In such a case, wait a few minutes until the instrument has warmed up and the "ExtRef" display turns blue.

4-7. Service Mode

Indicates the service mode status.

Value	Description
SVC (displayed in blue)	The E5071C is in one of the following mode. • Service mode, which is used for self-diagnosis • 8 term calibration mode • Advanced Mode in TDR Measurement performance will not be guaranteed according to the specifications. If, under normal use, the system remains in the

	service mode and does not return to normal operating mode, there is a possibility that the instrument is out of order.
SVC (displayed in red)	An abnormal condition has been detected inside the E5071C. The unit may be damaged. Notify the Customer Contact listed at the end of this manual or the distributor from whom the unit was purchased.
SVC (displayed in gray)	The E5071C is in normal mode.

4-8. Date and Time

Displays the date and time generated by the internal clock. The display format is as follows:

YYYY-MM-DD HH:MM

Where:

YYYY: Year (AD)
MM: Month
DD: Day
HH: MM: Time (0:00 to 23:59)

- You can turn the date and time display on/off by: **System** > **Misc Setup** > **Clock Setup** > **Show Clock.**

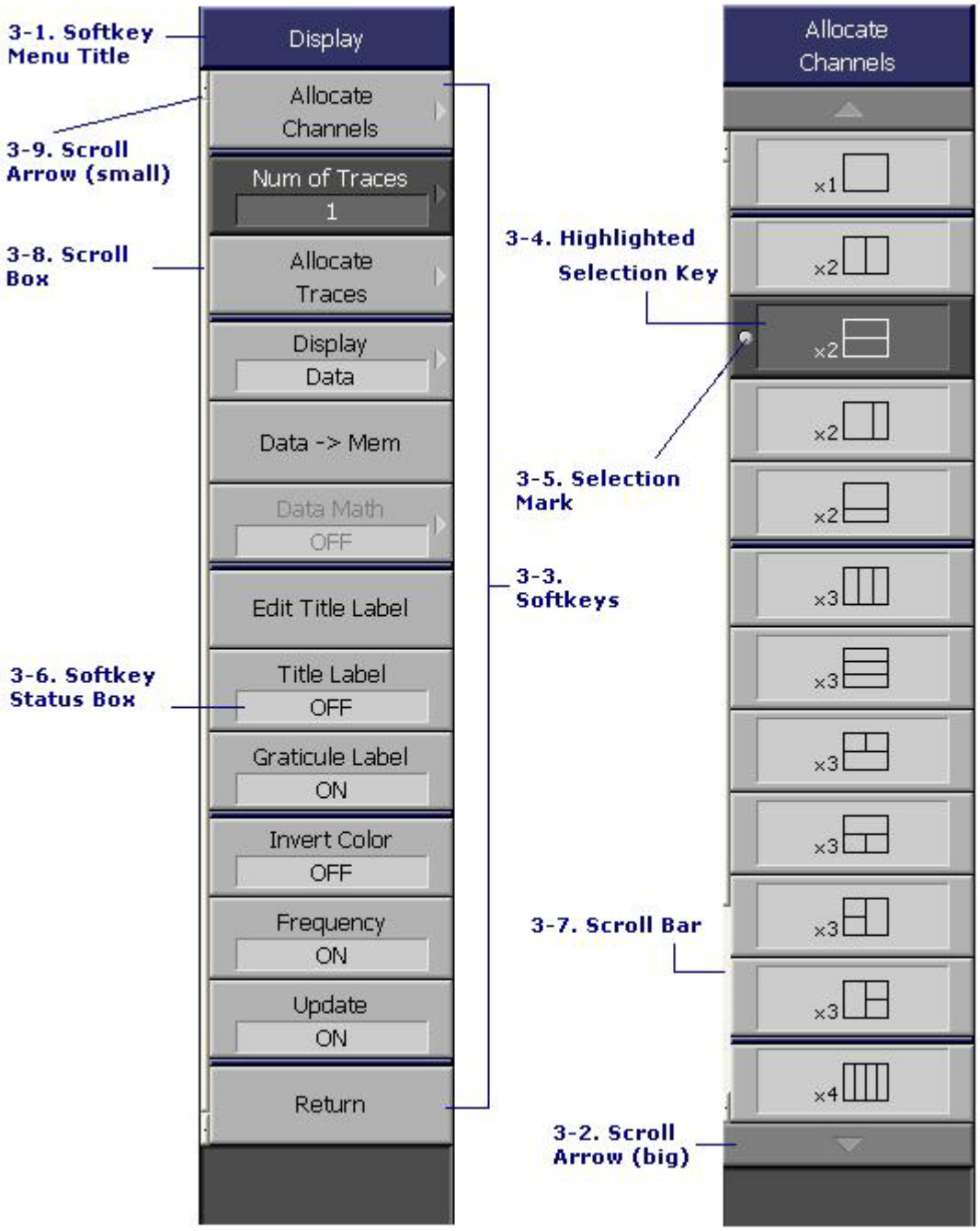
Menu Bar

By using the mouse and keyboard to manipulate the menu bar, you can perform interface operations that are equivalent to those of the keys in the **ACTIVE CH/TRACE** Block, **RESPONSE** Block, **STIMULUS** Block, **MKR/ANALYSIS** Block, and **INSTR STATE** Block on the front panel of the E5071C. The menus on the menu bar correspond to the key blocks, and their submenus to the hardkeys inside the key blocks.

Softkey Menu Bar

A group of keys on the screen called by the softkeys and menu bars. You can manipulate these keys by using the **NAVIGATION** Block keys on the front panel, the mouse, or the keyboard. You can perform manipulations by directly touching the screen with your finger instead of using a mouse.

Softkey Menu Bar



e5070aue075

- To manipulate a menu bar, it has to be selected as the object to manipulate (with the focus placed on it). When the focus is placed on a menu bar, the menu title area at the top is displayed in blue. Pressing or clicking on **Focus** Key in the ENTRY Block enables you to move the focus to the desired object.

3-1. Softkey Menu Title

The title of the softkey menu is displayed here. Double-clicking on this part of the menu bar displays the top layer of softkeys.

3-2. Scroll Arrow (Large)

When the softkeys in a menu overflow the screen, using this key enables you to scroll the menu page by page. Both upward and downward scroll arrows are available. Use the mouse to manipulate these buttons.

3-3. Softkeys

These are the actual keys you would use to perform setup. A ▸ displayed to the right of a softkey indicates that pressing that softkey will display the lower layer of softkeys.

3-4. Highlighted Softkey

Pressing and **Enter** key on the front panel or pressing **Enter** key on the keyboard causes the highlighted (selected) softkey to be executed. You can change which softkey in the menu is highlighted by turning or pressing on the front panel or by pressing on the keyboard. Pressing the key on the front panel or the key on the keyboard brings up the upper level softkey menu, and pressing the key on the front panel or the key on the keyboard brings up the lower level softkey menu.

3-5. Selection Mark

Shows which softkey function is currently selected.

3-6. Softkey Status Display

Displays a softkey's setup status.

You can hide the frequency information in order to ensure its confidentiality or for other reasons. See Hiding Softkey's Frequency Information.

3-7. Scroll Bar

When the softkeys in a menu overflow the screen, clicking on the blank part of the scroll bar enables you to scroll the softkey menu up or down.

3-8. Scroll Box

You can scroll the softkey menu up or down by using the mouse to select and drag the scroll box (pressing the button on the object to be moved and then releasing the button at the desired location). The length and position of the scroll box indicate the length and position of the currently displayed part of the softkey menu relative to the entire menu.

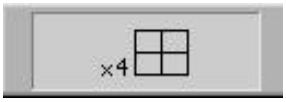
3-9. Scroll Arrow (Small)

Using this button, you can scroll the menu one softkey at a time. Both upward and downward scroll arrows are available. Use the mouse to manipulate these buttons.

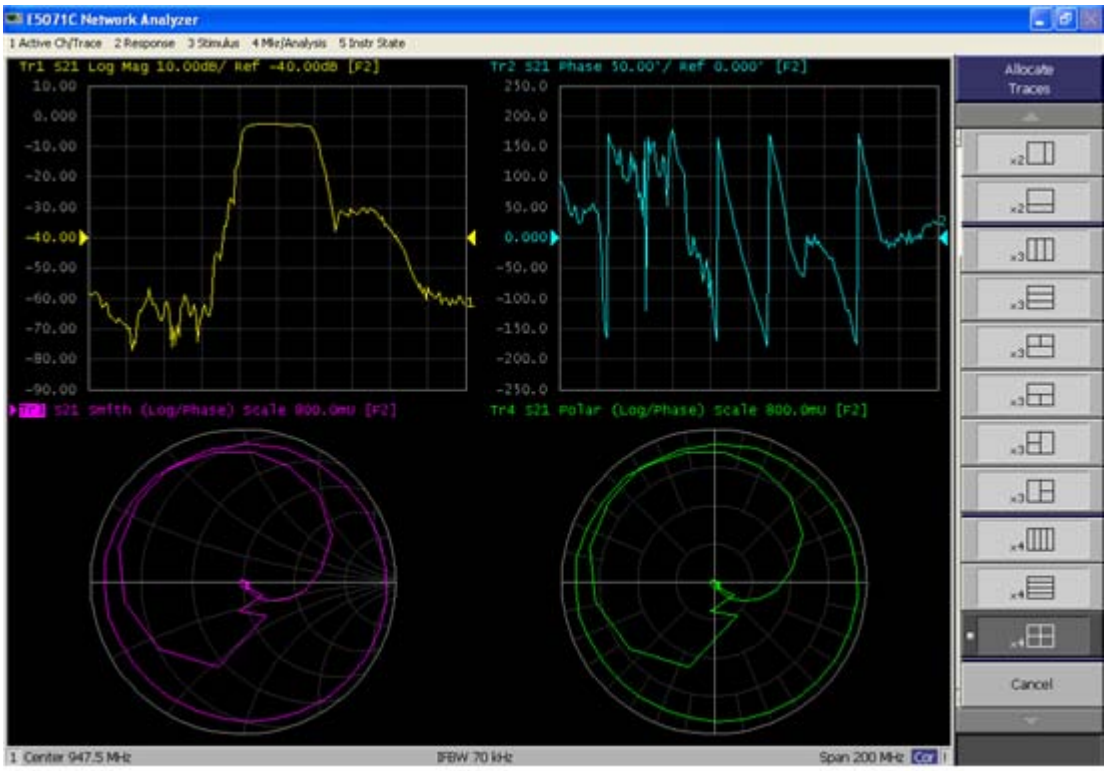
Three Methods of Operation

You can operate the E5071C using one of three operating methods: using keys on the front panel, using a mouse and keyboard, and using the touch screen. This section illustrates these three operating methods through the example in which the channel window layout is set to the four-channel display as shown in the following figure.

In the next section and those following it, a series of operations is expressed as follows:

Display > **Allocate Traces** > 

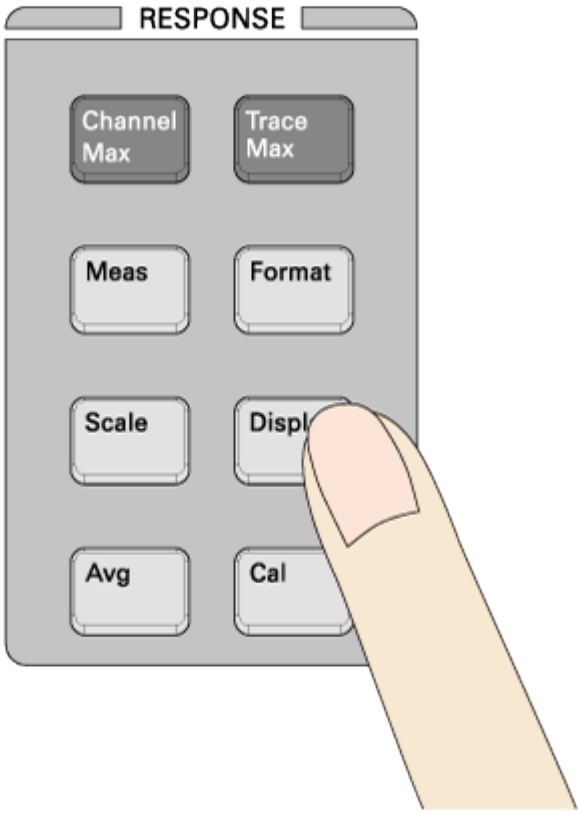
Four-Trace Display



e5071c193

Operating Method Using Keys

1. Press the **DISPLAY** key in the **RESPONSE** Block.



e5071c307

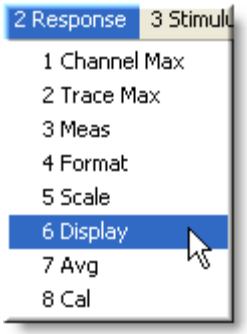
2. Press or key to move the cursor to the **Allocate Traces**, then press **Enter** or ( ) key.

e5071c002

3. Press or key to move the cursor to , then press **Enter** key.

Operation Method Using a Mouse

1. From the **Response** menu, press **Display** key.

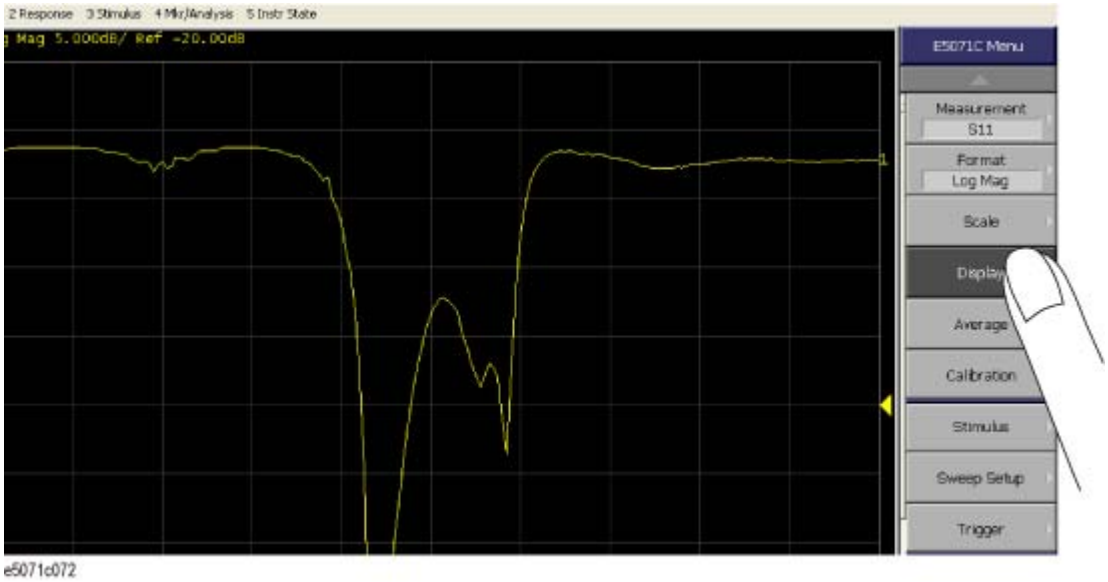


e5071c051

2. Click **Allocate Traces**.
3. Click

Operation Method Using the Touch Screen

1. Press **Display** key after presetting the E5071C (Executing **Preset** key).



e5071c072

2. Click/Press **Allocate Traces.**
3. Click/Press any desired setting.

Basic Measurement Procedures

To better understanding how to use the E5071C, this section describes the basic measurement procedure using the E5071C and presents an example of the transmission measurement of a bandpass filter.

Basic Measurement Flow

1. **Determining measurement conditions**
   - Presetting the E5071C
   - Select:
     - S-parameter
     - Data format
     - Frequency range
     - Number of measurement points
     - Power level
     - IF bandwidth
2. **Calibration**
   - Specify calibration kit
   - Select calibration type
   - Measure calibration data
   - Make error correction ON
   - Save calibartion data
3. **Connecting the Device Under Test (DUT)**
   - Connect DUT
   - Adjust the scale
4. **Analyzing measurement results**
   - Analysis using markers
5. **Outputting measurement results**
   - Store measurement file to a disk

Measurement Example of a Bandpass Filter

This section describes how to measure the transmission characteristics of a 947.5 MHz bandpass filter. The measurement conditions for this measurement example are those suitable for a 947.5 MHz bandpass filter. To measure another device under test (DUT), change the measurement conditions to suit the particular DUT.

STEP 1. Determining Measurement Conditions

1. Preset the E5071C.

   Preset > **OK**

2. Set the S-parameter to S21.

   Meas > **S21**

   3. When measuring the reverse transmission characteristics, set the S-parameter to S12.

3. Set the data format to the log magnitude format

   Format > **Log Mag**

4. Set the center frequency to the bandpass filter center frequency. Next, specify the span frequency, which is set to 200 MHz in this measurement example.

   Center > **9** > **4** > **7** > **. 5** > **M/m**
   Span > **2** > **0** > **0** > **M/m**

   3. When entering the frequency unit using the keyboard, type "G" for GHz, "M" for MHz, and "k" for kHz.

5. Specify the number of measurement points per sweep. The number of measurement points in this measurement example is set to 401.

   Sweep Setup > **Points** > **4** > **0** > **1** > **x1**

6. Specify the power level of the signal source. The power level in this measurement example is set to -10 dBm.

   Sweep Setup > **Power** > **+/-** > **1** > **0** > **x1**

7. Specify the IF bandwidth of the receiver as necessary. In this measurement example, the IF bandwidth is set to 10 kHz because of the need to lower the noise floor.

   Avg > **IF Bandwidth** > **1** > **0** > **k/m**

STEP 2. Calibration

To turn the error correction ON, set the calibration type to the full 2-port calibration and measure the calibration data.

For details about calibration, see Calibration.

1. Select the calibration kit suitable for the measurement cable. In this measurement example, Calibration Kit 85032F is selected.

   Cal > **Cal Kit** > **85032F**

2. Set the calibration type to the full 2-port calibration using the test port 1 and 2.

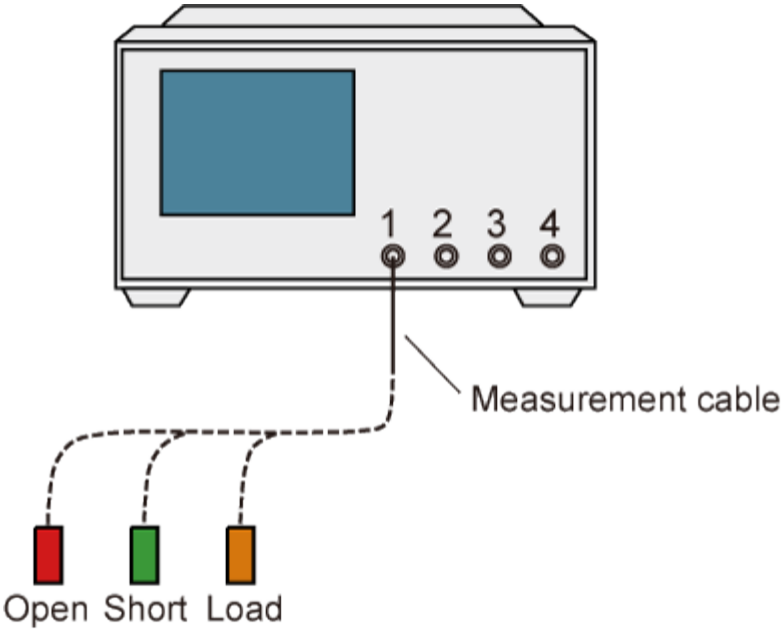
   Cal > **Calibrate** > **2-Port Cal** > **Select Ports - 1-2**

3. Connect the OPEN standard (included in the calibration kit) to the other end of the measurement cable that is connected to the test port 1 as shown in the following figure, and measure the open calibration data at the test port 1. After measuring the open calibration data, a checkmark is displayed to the left of the **Port 1 Open** menu.

   Cal > **Calibrate** > **2-Port Cal** > **Reflection** > **Port1 Open**

   In the same way, measure the calibration data for the SHORT/LOAD standards at the test port 1.

   Connecting the OPEN/SHORT/LOAD standards

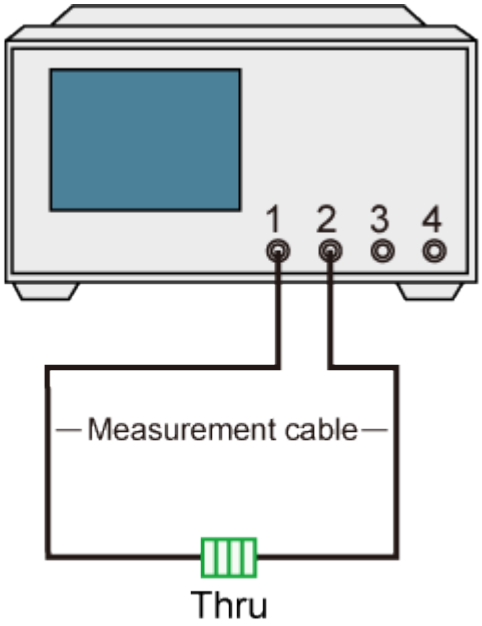


e5071c323

4. In the same way as described above, measure the calibration data for the OPEN/SHORT/LOAD standards at the test port 2.
5. Connect the THRU standard (included in the calibration kit) between the measurement cables as shown in the figure below , and measure the thru calibration data. After measuring the thru calibration data, a checkmark appears is displayed to the left of the **Port 1-2 Thru** button.

Cal > **Calibrate** > **2-Port Cal** > **Transmission** > **Port 1-2 Thru**

Connecting the THRU standard



e5071c316

6. Make the full 2-port calibration measurement DONE. The calibration factor is calculated based on the calibration data acquired, and the error correction is turned ON.

Cal > **Calibrate** > **2-Port Cal** > **Done**

7. Select the type in which the data is to be saved before saving the calibration factor (calculated based on the calibration data).

Save/Recall > **Save Type** > **State & Cal**

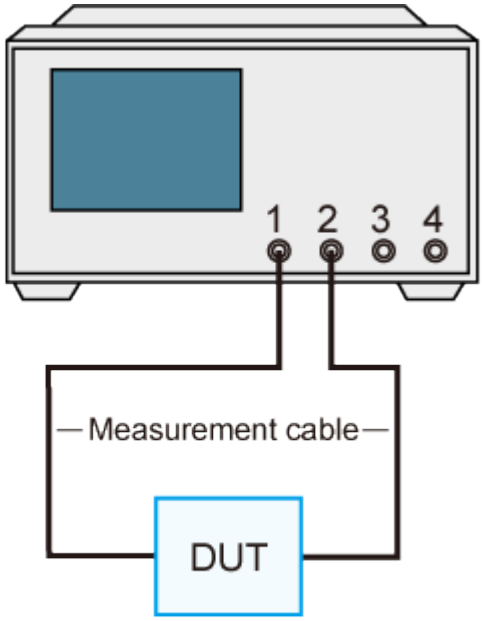
8. Store the calibration file to the disk of the E5071C. The symbol "X" appearing in the operations below represent the assigned numbers to be used when the file is saved.

Save/Recall > **Save State** > **State 0X**

STEP 3. Connecting the Device Under Test (DUT)

1. Connect to the DUT to the E5071C. (See the below figure)

Connecting the DUT



e5071c315

2. Set the appropriate scale by executing the auto scale. (See the below figure)

   Scale > **Auto Scale**

   You can also adjust the scale by entering arbitrary values in the **Scale/Div** button, **Reference Position** button, and **Reference Value**.

S21 trace after executing the auto scale

STEP 4. Analyzing Measurement Results

This section describes how to use the marker function to read out important parameters for the transmission measurement of the bandpassfilter (insertion loss, -3 dB bandwidth).

Measuring the Insertion Loss

1. Display a marker.

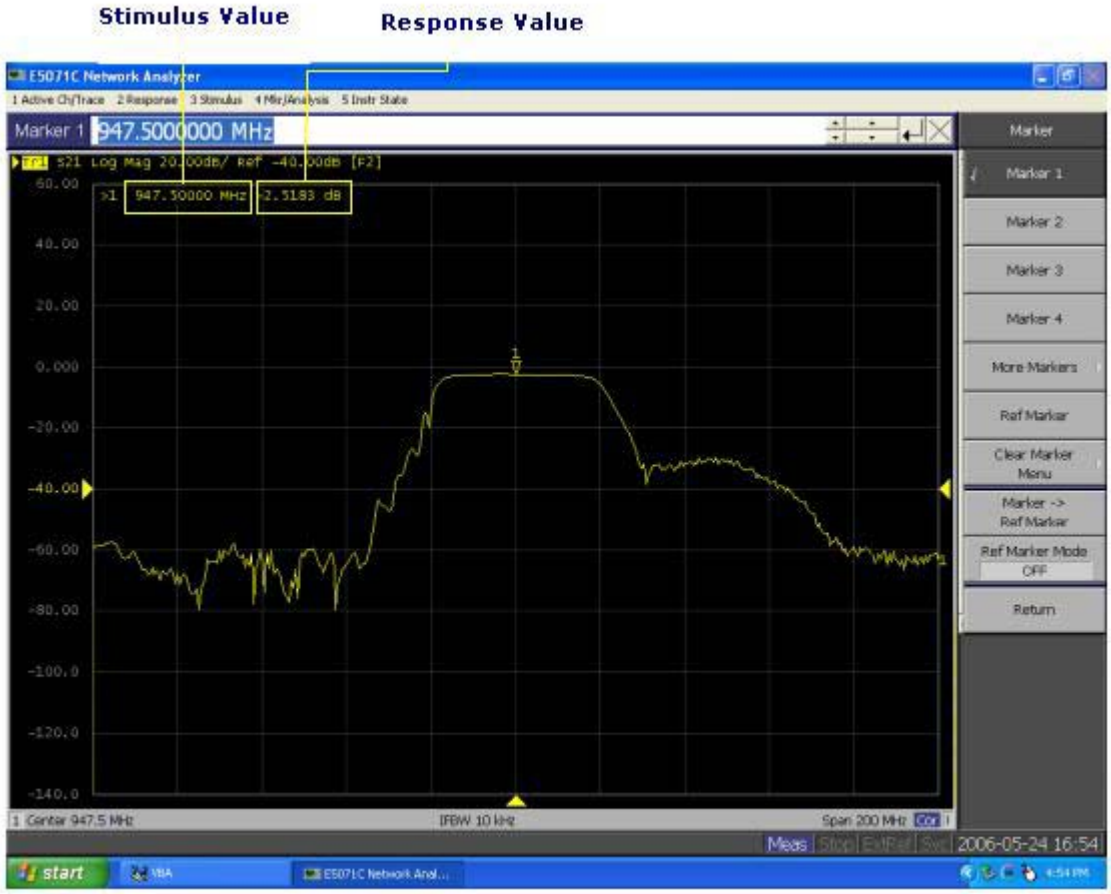
   Marker > **Marker 1**

2. Using one of the following methods, move the marker to the center frequency of the bandpass filter.

- On the entry bar, press **9**> **4** > **7** > **.** > **5** > **M/m**
- Turn the rotary knob on the front panel to set it to the center frequency (947.5 MHz).

3. Read the marker value displayed as shown in the figure below. In this example, the response value denotes the insertion loss.

Measuring an Insertion Loss



e5071c054

Measuring the -3 dB Bandwidth

Using the marker bandwidth search function, the bandwidth, center frequency between two cutoff frequency points, Q value, and insertion loss are all read out. These parameters are described in the following table.

- If the two cutoff frequency points are not found, all data items except the insertion loss revert to zero.

Parameter	Description
BW (Bandwidth)	Stimulus width between two cutoff frequency points (low and high)
cent (Center Frequency)	Center point between cutoff frequency points (low and high)
low (Left-side Cutoff Frequency)	The lower frequency of the two cutoff frequency points
high (Right-side Cutoff	The higher frequency of the two cutoff

Frequency)	frequency points
Q (Q Value)	Q = cent/BW
loss (Insertion Loss)	The measurement value of the active marker.

1. Display a marker.

Marker > **Marker 1**

2. Using one of the following methods, move the marker to the center frequency of the bandpass filter.

- On the entry bar, press **9** > **4** > **7** > **.** > **5** > **M/m**
- Turn the rotary knob on the front panel to set it to the center frequency (947.5 MHz).

3. Specify the bandwidth definition value that defines the pass band of the filter. In this measurement example, it is set to -3 dB.

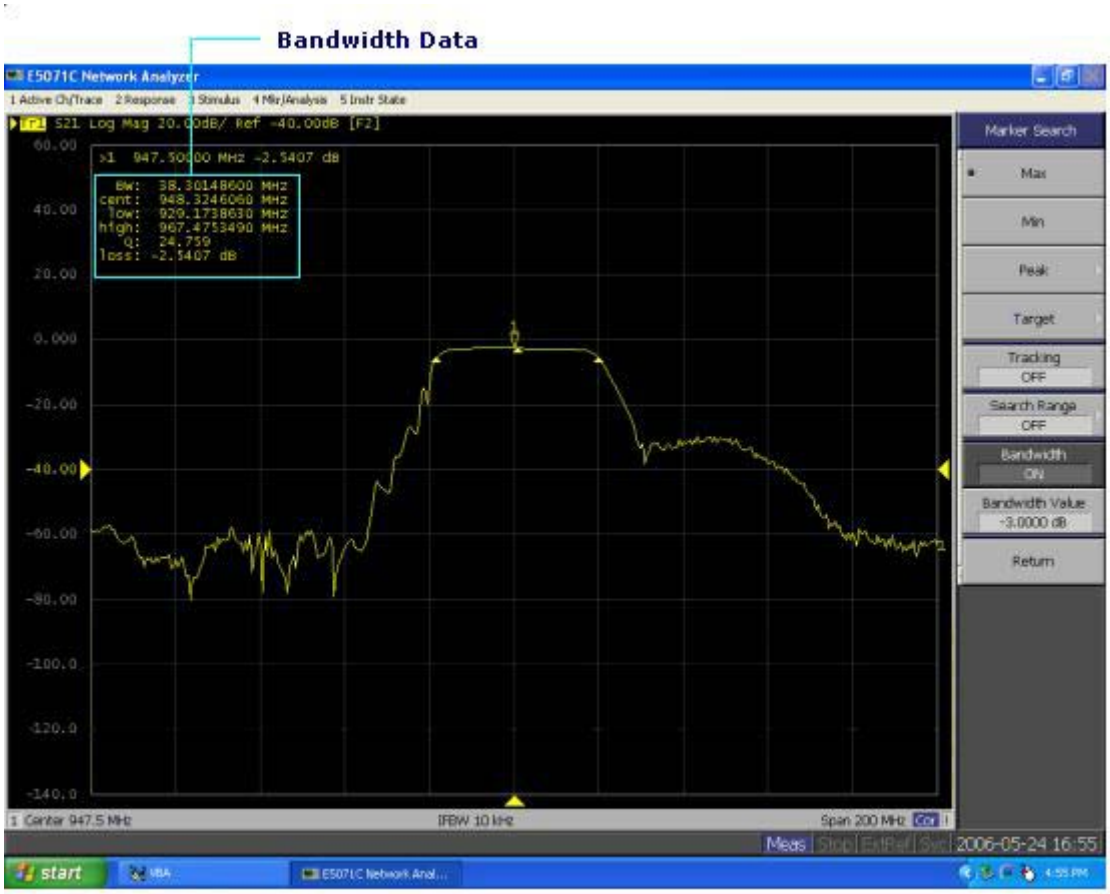
Marker Search > **Bandwidth Value** > **+/-** > **3** > **x 1**

4. Set the bandwidth search function ON.

Marker Search > **Bandwidth**

5. The bandwidth data items (BW, cent, low, high, Q, loss) will be displayed. (See the following figure)

Measuring the -3 dB Bandwidth



e5071c055

STEP 5. Outputting Measurement Results (Save)

You can save not only the internal data but also the measurement results such as trace data and display screens to the disk .

Saving the Trace Data(in CSV format)

You can save the trace data to the disk of the E5071C in CSV file format (extension: .csv). Since the CVS-formatted data to be saved is a text file, you can analyze the data using Microsoft Excel.

Follow the step below to save the trace data:

Save/Recall > **Save Trace Data**

Saving the Display Screen

You can save the screen displayed on the E5071C to the disk of the E5071C in Windows bitmap file format (extension: .bmp) or Portable Network Graphics format (extension: .png).

Follow the step below to save the display screen:

System > **Dump Screen Image**

- The image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture/System** key is pressed) is saved.

Measurement

This chapter provides information related to the measurements done through E5071C.

- Setting Measurement Conditions
- Calibration
- Making Measurements
- Data Analysis
- Data Output
- Optimizing Measurements
- Measurement Examples
- Fixture Simulator

Setting Measurement Conditions

Setting Measurement Conditions

- Initializing Parameters
- Setting Channels and Traces
- Setting the System Z0
- Setting Stimulus Conditions
- Selecting Measurement Parameters
- Selecting a Data Format
- Setting the Scales
- Setting Window Displays

Initializing Parameters

The E5071C has three different initial settings as shown below.

Initial setting	Restore method
Preset state	Press **Preset** > **OK** the front panel or Execute the :SYST:PRES command
*RST state	Execute the *RST command
Factory default setting	E5071C factory (default) settings

The user can set items to be preset freely. For more information, see Setting the user preset function.

Other topics about Setting Measurement Conditions

Setting Channels and Traces

- Overview
- Setting Upper Limit of Number of Channels/Traces
- Setting Channel Display (Layout of Channels)
- Setting Trace Display
- Active Channel
- Setting Parameter for each Setup Item (Analyzer, Channel, Trace)

Other topics about Setting Measurement Conditions

Overview

The E5071C allows you to use up to 160 channels (when the number of traces is up to 9) to perform measurement under 160 different stimulus conditions.

For each channel, up to 4, 6, 9, 12 or 16 traces (measurement parameters) can be displayed. Because multiple traces can be displayed for each channel, no feature is provided to link the stimulus conditions between channels, and each channel is always independent of the others. In other words, for the E5071C, you need to set the measurement conditions and execute calibration for each channel you use for measurement.

With the E5071C, you can change the number of available channels and the upper limit of the number of traces. If you change the upper limit setting, you need to restart the firmware of the E5071C. Therefore, first, set the upper limit appropriately depending on the numbers of channels and traces necessary for your measurement.

When you set items whose setting target is channels/traces (refer to Parameter setting for each setup item), the target is the selected (active) channel/trace. You can specify only the displayed channels/traces as active channels/traces. Therefore, set the display of channels/traces before setting the measurement conditions.

Setting Upper Limit of Number of Channels/Traces

You can select the upper limits of the number of channels and the number of traces from the following combinations.

Channels	Traces	Maximum Number of Points	Supported Revision (Firmware Revision)	Note

1	4	1601	All	You may want to select the "1-channel, 4-trace" or "2-channel, 4-trace" configuration to save the time required to save/recall the instrument state file, since this takes longer with other configurations. More channels and traces requires longer measurement time.
2	4	1601		
4	16	1601		
9	9	1601		
12	6	1601		
16	4	1601		
16	16	1601		
24	12	1601		
36	9	1601		
2	8	1601	From A.09.50	
2	8	10001	From A.09.20 to A.09.4x	
2	16	10001	From A.09.50	
1	4	20001	ALL	

72	9	401	From A.08.10	For these setup, press **System** > **Service Menu** > **Channel/Trace Setup** After the configuration setup is applied, memory swap may occur during the measurement. The instrument may seem to pause during the measurement.
96	9	201		
160	9	201	From A.09.10	

- Its important to realize that the combination used to save a state file must be the same used to recall it; that is, you cannot save a state file with one combination of channels/traces and recall it with another combination. For more details, see Saving and Recalling Instrument State.

The selection procedure is as follows:

1. Press **System** > **Misc Setup** > **Channel/Trace Setup**. (**System** > **Service Menu** > **Channel/Trace Setup** for some settings)
2. Press the desired softkey to select the upper limits of the number of channels and the number of traces.
3. The dialog box that prompts "New Channel/Trace configuration will take effect after firmware restart." appears.
4. Click **OK**.
5. Press any softkeys or hardkeys besides **Preset** key.
6. The dialog box that prompts "Do you want to restart now?" appears.
7. Click **Yes**.
8. On choosing a multi-channel mode, it may take more time to restart the firmware. The screen may seem to freeze but will resume their normal operation once the settings are applied.

Setting Channel Display (Layout of Channels)

The measurement result for each channel is displayed in its dedicated window (channel window). You cannot have a single window display the measurement results from more than one channel. This means that the

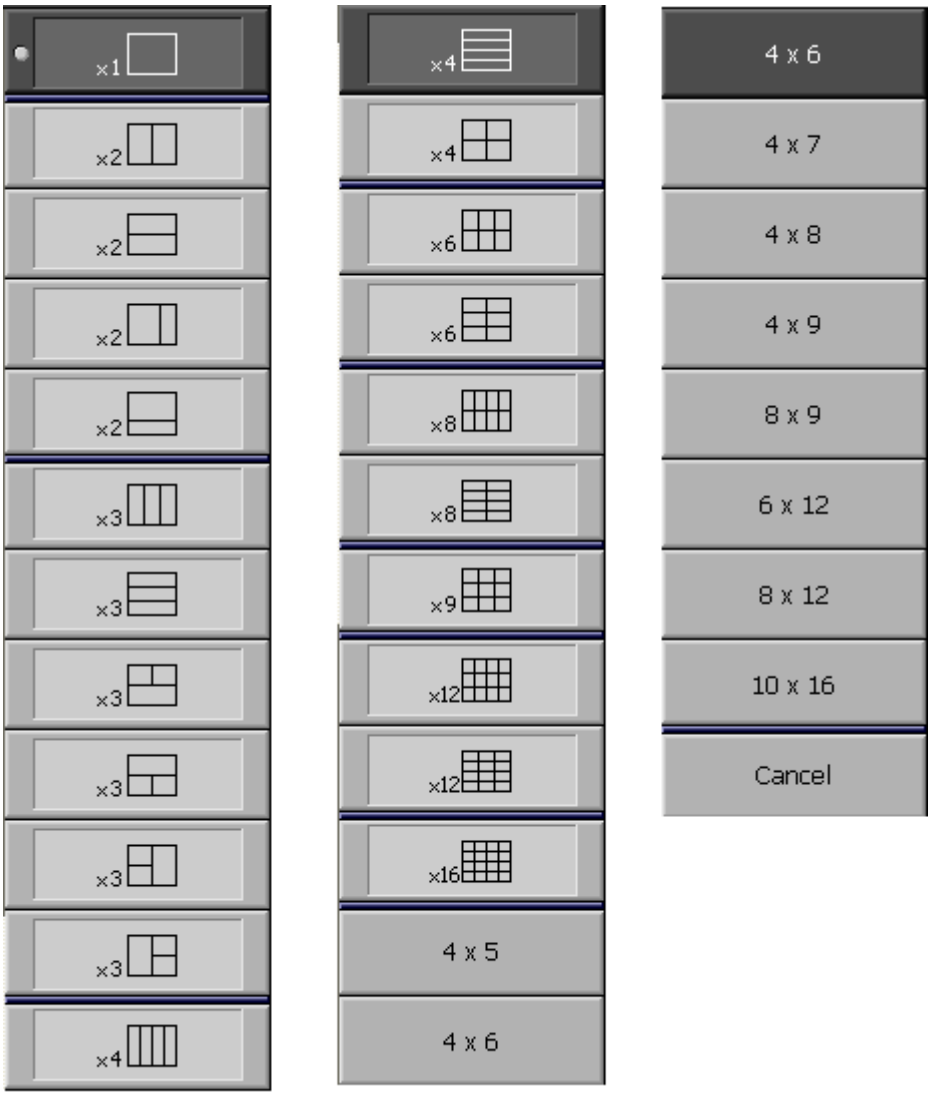
setting of the window layout determines the number of channels displayed on screen.

- The execution of measurement for each channel does not depend on how the channel is displayed (channels that are not displayed can be measured). For information on executing measurement for each channel (trigger mode and trigger source), refer to Making Measurements.

The procedure for setting the window layout is as follows:

1. Press **Display** key > **Allocate Channels**.
2. Press the desired softkey to select the window layout shown below.

Windows Layout



e5071c061

- From Firmware revision A.09.10, channel display configuration of 10×16 is added.

From Firmware revision A.08.10, channel display configuration of 8×9, 6×12 and 8×12 is added.

Setting Trace Display

Setting the number of traces

Depending on the measurement parameters of the traces displayed for each channel, the sweep necessary for each channel is executed. For more information, refer to Sweep Order in Each Channel.

You specify the trace display by setting the number of traces (upper limit of displayed trace numbers). For example, if you set the number of traces to 3, traces 1 through 3 are displayed.

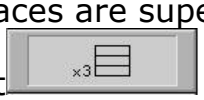
The procedure for setting the number of traces is as follows:

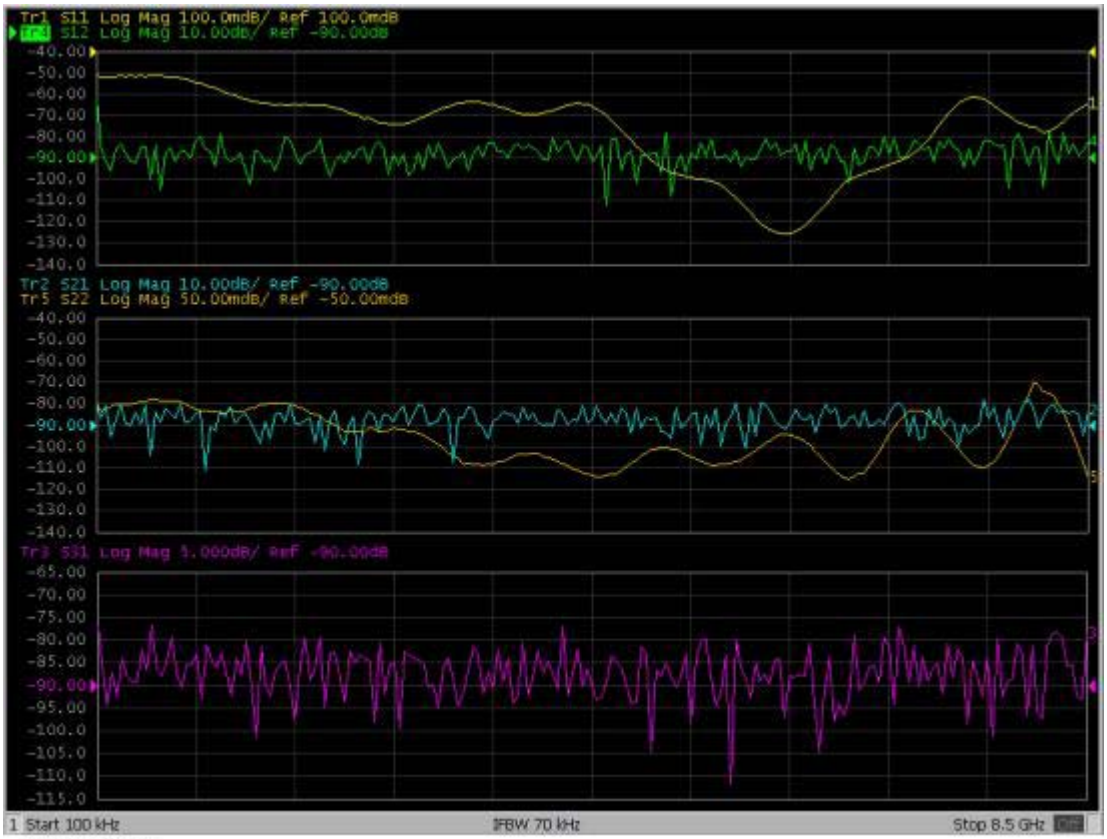
1. Press **Channel Next** or **Channel Prev** to select the channel for which you want to set the number of traces.
2. Press **Display** > **Number of Traces**.
3. Press the desired softkey to set the number of traces.

Setting trace layout (graph layout)

Traces are laid out and displayed in the order of the trace number from graph 1 according to the graph layout in the channel window.

You can select the graph layout from the windows layout.

If the number of traces is less than the number of graphs, nothing is displayed in the remaining area. If the number of traces you set exceeds the number of graphs, excess traces are superimposed from the first graph. For example, if you select  as the graph layout and set the number of traces to 5, graph 1 (Gr1 in Graph layout) and graph 2 (Gr2 in Graph layout) display traces 1 and 4 and traces 2 and 5, respectively, by superimposing, and graph 3 (Gr3 in Graph layout) displays only trace 3 as shown in the figure below.

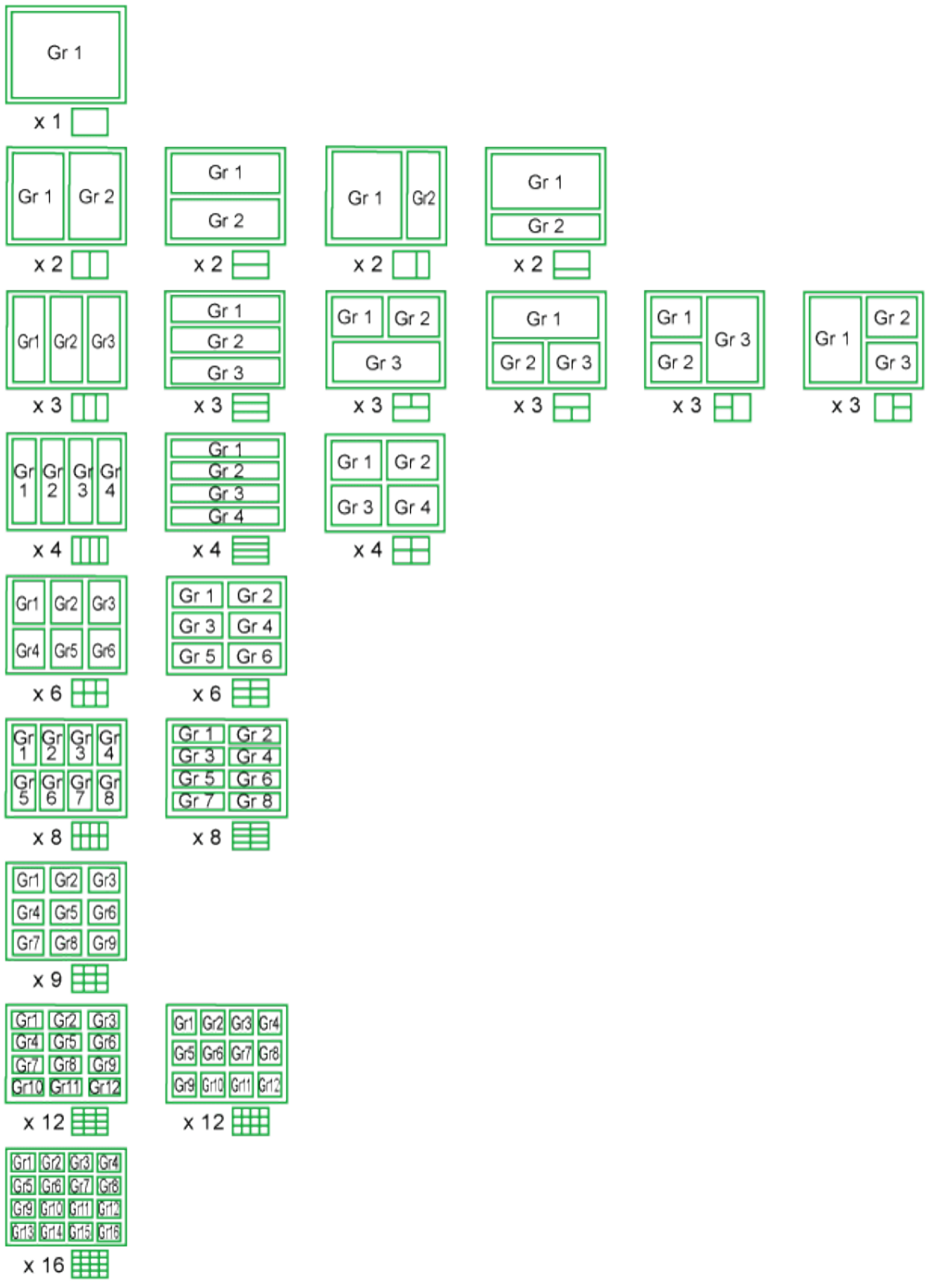


e5071c059

The procedure for setting the graph layout is as follows:

1. Press **Channel Next** or **Channel Prev** to select the channel for which you want to set the graph layout.
2. Press **Display** > **Allocate Traces**.
3. Press the desired softkey to select the graph layout shown below.

Graph Layout


Gr 1
x 1
Gr 1
Gr 2
x 2
Gr 1
Gr 2
x 2
Gr 1
Gr2
x 2
Gr 1
Gr 2
x 2
Gr1
Gr2
Gr3
x 3
Gr 1
Gr 2
Gr 3
x 3
Gr 1
Gr 2
Gr 3
x 3
Gr 1
Gr 2
Gr 3
x 3
Gr 1
Gr 2
Gr 3
x 3
Gr 1
Gr 2
Gr 3
Gr 4
x 4
Gr 1
Gr 2
Gr 3
Gr 4
x 4
Gr 1
Gr 2
Gr 3
Gr 4
x 4
Gr1
Gr2
Gr3
Gr4
Gr5
Gr6
x 6
Gr 1
Gr 2
Gr 3
Gr 4
Gr 5
Gr 6
x 6
Gr 1
Gr 2
Gr 3
Gr 4
Gr 5
Gr 6
Gr 7
Gr 8
x 8
Gr 1
Gr 2
Gr 3
Gr 4
Gr 5
Gr 6
Gr 7
Gr 8
x 8
Gr1
Gr2
Gr3
Gr4
Gr5
Gr6
Gr7
Gr8
Gr9
x 9
Gr1
Gr2
Gr3
Gr4
Gr5
Gr6
Gr7
Gr8
Gr9
Gr10
Gr11
Gr12
x 12
Gr1
Gr2
Gr3
Gr4
Gr5
Gr6
Gr7
Gr8
Gr9
Gr10
Gr11
Gr12
x 12
Gr1
Gr2
Gr3
Gr4
Gr5
Gr6
Gr7
Gr8
Gr9
Gr10
Gr11
Gr12
Gr13
Gr14
Gr15
Gr16
x 16

e5071c471

Active Channel

The *active channel* is the one whose settings can currently be changed. The window frame of the active channel is displayed brighter than the window frames of the other channels. To change the settings specific to a certain channel, you must first activate the channel.

To change the active channel, use the following hardkeys:

Hardkey	Function
Channel Next	Change the active channel to the next channel with the larger channel number.
Channel Prev	Change the active channel to the previous channel with the smaller channel number.

Active trace

The active trace is the one whose settings can currently be changed. The trace name on the screen (for example, Tr3) of the current active trace is highlighted and indicated with ▶ to the left. To change the settings specific to a certain trace, you must first activate the trace.

To select the active trace, use the following hardkeys:

Hardkey	Function
Trace Next	Change the active trace to the next trace with the larger trace number.
Trace Prev	Change the active trace to the previous trace with the smaller trace number.

Setting Parameter for each Setup Item (Analyzer, Channel, Trace)

The following table lists the setting parameters and indicates the setup item (analyzer, channel, or trace) that each parameter controls along with the applicable setup key(s).

Parameter	Controlled Setup Items			Setup Key(s)
	Analyzer	Channel	Trace	
Stimulus Settings				
Sweep range		x		**Start**, **Stop**, **Center**, **Span**
Power, CW frequency		x		**Sweep Setup** > **Power**

Sweep time/Sweep delay time		x		**Sweep Setup** > **Sweep Time, Sweep Delay**
Number of points		x		**Sweep Setup** > **Points**
Segment sweep		x		**Sweep Setup** > **Sweep Type, Edit Segment Table, Segment Display**
Sweep mode		x		**Sweep Setup** > **Sweep Mode**
Trigger Settings				
Trigger source	x			**Trigger** > **Trigger Source/Restart/Trigger**
Trigger mode		x		**Trigger** > **Hold/Hold All Channels/Single/ Continuous/Continuous Disp Channels**
Response Settings				
Measurement parameter			x	**Meas**
Data format			x	**Format**
Scale, Electrical delay, Phase offset			x	**Scale**
Memory trace and data math			x	**Display** > **Display/Data->Mem/ Data Math**
Window title		x		**Display** > **Edit Title Labe/ Title Label (ON/OFF)**
Graticule label in rectangular form		x		**Display** > **Graticule Label (ON/OFF)**
Color inversion	x			**Display** > **Invert Color**
Frequency display ON/OFF	x			**Display** > **Frequency (ON/OFF)**
Display update ON/OFF	x			**Display** > **Update (ON/OFF)**
Averaging		x		**Avg** > **Averaging Restart/ Avg Factor/Averaging**

				(ON/OFF)
Smoothing			x	**Avg** > **Smo Aperture/ Smoothing (ON/OFF)**
IF bandwidth		x		**Avg** > **IF Bandwidth**
Calibration		x		**Cal**
Marker			x	**Marker**, **Marker Search**, **Maker Fctn**
Analysis				
Fixture simulator		x		**Analysis** > **Fixture Simulator**
Time domain			x	**Analysis** > **Gating/ Transform**
Parameter conversion			x	**Analysis** > **Conversion**
Limit test			x	**Analysis** > **Limit Test**
Saving and recalling data	x			**Save/Recall**
Macro	x			**Macro Setup**, **Macro Run**, **Macro Break**
System				
Printing/Saving display Screen/Beeper/GRIB settings/Network Settings/Date & Time/Key Lock/Backlight/Firmware Revision/Service menu	x			**System**
Preset	x			**Preset**

Setting the System Z0

The procedure for setting the system characteristic impedance (Z0) is as follows:

1. Press **Cal** key.
2. Click **Set Z0**, then input the system Z0.

Other topics about Setting Measurement Conditions

Setting Stimulus Conditions

You can set the stimulus condition for each channel independently.

- Setting the Sweep Type
- Setting the Sweep Range
- Enable Stimulus Signal Output
- Setting power level
- Setting fixed frequency at power sweep
- Setting CW Time Sweep
- Setting Power Level with Auto Power Range
- Setting the Number of Points
- Setting the Sweep Time

Other topics about Setting Measurement Conditions

Setting the Sweep Type

You can select the sweep type from the following four types.

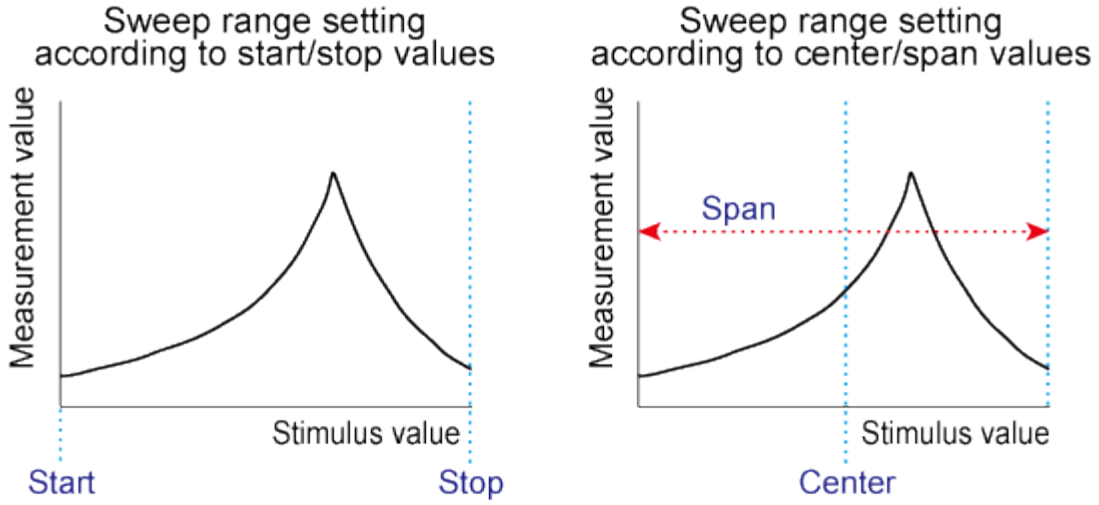
SoftKey	Description
Linear Freq	Sweeps frequencies in linear scale.
Log Freq	Sweeps frequencies in logarithmic scale.
Segment	Performs a sweep with linear sweep conditions (segments) combined. For more information, refer to Performing a Segment-by-Segment Sweep (segment sweep).
Power Sweep	Sweeps power levels in linear scale.

The procedure for selecting the sweep type is as follows:

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to set the sweep type.
2. Press **Sweep Setup** > **Sweep Type**.
3. Press the desired softkey to select the sweep type.
4. To use the E5071C in CW Time sweep, refer to Setting CW Time Sweep

Setting the Sweep Range

There are two ways to set the sweep range: by specifying the lowest and the highest values and by specifying the center value and a span. Once the sweep range is set, it is possible to change the range by substituting the lowest value, the highest value, or the center value with a value (stimulus value) represented by a marker on the trace.



e5071c443

Setting the Sweep Range with the Lowest and Highest Values

1. Press **Channel Next**/**Channel Prev** keys to select the channel whose sweep range will be set.
2. Click **Start**, then input the lowest value.
3. Click **Stop**, then input the highest value.

Setting the Sweep Range with the Center Value and a Span

1. Press **Channel Next**/**Channel Prev** keys to select the channel whose sweep range will be set.
2. Click **Center**, then input the center value.
3. Click **Span**, then input the span value.

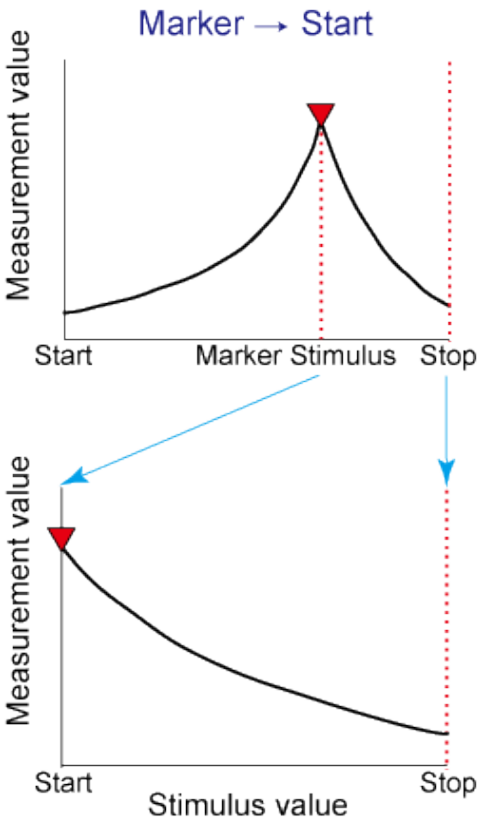
Setting the Sweep Range Using the Marker

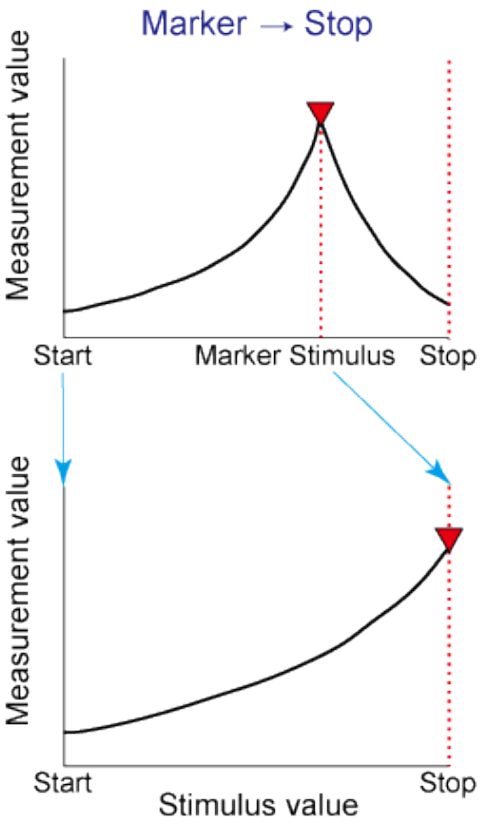
1. In the channel window whose range must be set, place the *active marker* on the active trace to a position that corresponds to the new range (to the lowest, highest, or center value).
2. Press **Marker Fctn** key.
3. Click the softkey that corresponds to each value.

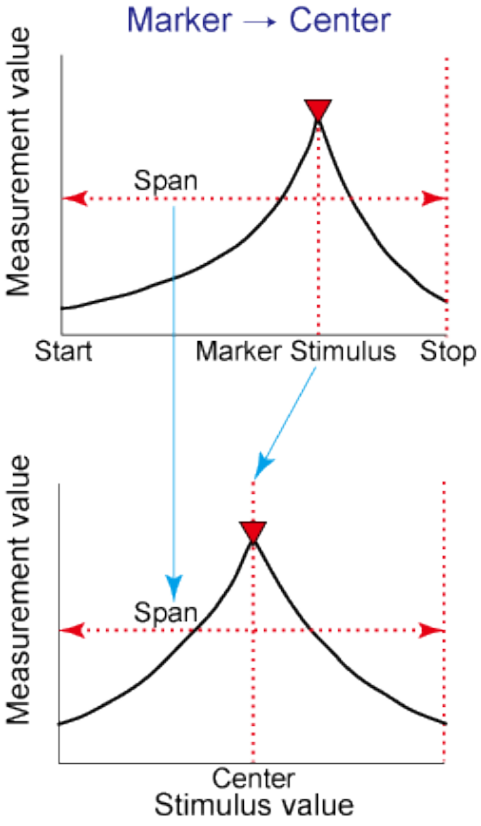
SoftKey	Function
Marker ->	Sets the lowest value to the stimulus value of the active marker

Start	on the currently active trace.
Marker -> Stop	Sets the highest value to the stimulus value of the active marker on the currently active trace.
Marker -> Center	Sets the center value to the stimulus value of the active marker on the currently active trace.

- If the reference marker is on and the stimulus value of the active marker is expressed by a value relative to the reference marker, the absolute stimulus value will be used to set the new sweep rang.


Marker → Start
Measurement value
Start
Marker Stimulus
Stop
Measurement value
Start
Stop
Stimulus value


Marker → Stop
Measurement value
Start
Marker Stimulus
Stop
Measurement value
Start
Stop
Stimulus value


Marker → Center
Measurement value
Span
Start
Marker Stimulus
Stop
Measurement value
Span
Center
Stimulus value

Enable Stimulus Signal Output

You can turn on/off the stimulus signal output, but this will prevent you from performing measurement. Therefore, you will not normally use this feature. This is mainly used to turn the output back to on after it has been turned off by the power trip feature.

Follow these steps to turn the stimulus signal output on/off:

1. Press **Sweep Setup** key.
2. Click **Power** > **RF Out** (Each press toggles between on/off).

When set to off, "RF OFF" is displayed in Instrument Status Bar.

Power trip

The power trip is a feature that the instrument uses to automatically turn off the output of the stimulus signal to protect the instrument when a signal whose level exceeds the upper limit is inputted to the test port.

If the power output is automatically turned off by the power trip feature, remove the cause of the over-input and turn on the power output according to the above steps to restart the measurement.

Setting power level

1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **Port Couple**, then select the on/off setting of the level coupling for all ports.

ON	The same power level is outputted to all ports.
OFF	A specific power level is outputted to each port independently.

- The power level of port 1 is coupled with the power level for all ports.

5. If you change the on/off setting of the level coupling, all ports are automatically changed to the same level value as that of port 1.

4. Follows the following procedure according to Port Couple.

- When setting level for all ports (Port Couple ON)

a. Click **Power**, then enter the power level.

- When setting level for each port (Port Couple OFF)

a. Press **Port Power**, then click the softkey corresponding to each port (Port 1 Power to Port 4 Power).

b. Enter the power level.

Correcting attenuation of power level (using power slope feature)

You can use the power slope feature to correct the attenuation of a power level so that it is simply proportional to the frequency (attenuation due to cables and so on), which improves the accuracy of the level actually applied to the DUT.

Turning power slope feature on/off

1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **Slope [OFF]** (**Slope [ON]**). Each press toggles between on/off.

Setting correction coefficient (correction amount per 1 GHz)

1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **Slope [xxx dB/GHz]** ("xxx" represents the current set value.).
4. Enter the correction coefficient using the **ENTRY** block keys on the front panel.

Setting fixed frequency at power sweep

The procedure for setting the fixed frequency (CW frequency) at the power sweep is as follows:

1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Power** > **CW Freq**, then enter the fixed frequency.

Setting CW Time Sweep

The E5071C supports sweep type of CW Time where the E5071C is set to a single frequency, and the data is displayed versus time. To use the E5071C in CW time, follow the procedure explained below:

1. Press **Channel Next**/**Channel Prev** keys to select the channel whose sweep range will be set.
2. Press **Sweep Setup** > **Sweep Time**, then input the duration of the sweep which is displayed on X-axis.
3. Click **Span**, then input the span value as 0 Hz (zero span).

4. Press **Center**, then input the center value.
5. Press **Marker** to display the marker 1. The marker position shows the time for its measurement point.

Setting Power Level with Auto Power Range (options 2D5, 4D5, 2K5 and 4K5 only)

- Auto Power range function works only with the E5071C options 2D5, 4D5, 2K5 and 4K5 having a mechanical source attenuator. Auto power range functionality is neither required nor available in other E5071C options which are having electrical source attenuator.

When the Auto Power Range set function is effective, the proper source attenuator and power range are selected automatically according to the output power setting of each channel.

Under the following conditions, turn OFF the Auto Power Range set function and set the power range and power level manually:

- When different power ranges are selected in multiple channel measurement settings. In this case, an error message is displayed to avoid source attenuator damage. For example, if Channel 1 power is at -35 dBm with the power range of -35 dBm to 0 dBm, and Channel 2 power is set at -5 dBm with the power range of -25 dBm to +10 dBm, this will causes an error because each selected power range is different. In such a case , turn OFF the Auto Range on Channel 2 and set the range as -35 to 0 manually, to set the same power range as of Channel 1.
    - Continuous switching may damage source attenuator.

Setting the level

1. Press **Channel Prev**/**Channel Next** to select the channel for which you want to set the power level.
2. Press **Sweep Setup** key, then click **Power** > **Port Couple** and select the **ON**/**OFF** setting of the level coupling for all ports.

Value	Description
ON	The same power level is outputted to all ports.
OFF	A specific power level is outputted to each port independently.

1. The power level of port 1 is coupled with the power level for all ports when Port Couple is set to ON.

2. If you change the on/off setting of the level coupling, all ports are automatically changed to the same level value as that of port 1.
3. When setting level for all ports (**Port Couple ON**), click **Power** input the power level using the **ENTRY** block keys on the front panel.

When setting level for each port (**Port Couple OFF**), click **Port Power** and select the softkey corresponding to each port (**Port 1 Power** to **Port 4 Power**).

The table below describes the relation between source power level and step attenuation:

Source Power Range	Step Attenuation
-25 ~ +10 dBm	0 dB
-35 ~ 0 dBm	10 dB
-45 ~ -10 dBm	20 dB
-55 ~ -20 dBm	30 dB
-65 ~ -30 dBm	40 dB
-75 ~ -40 dBm	50 dB
-85 ~ -50 dBm	60 dB

For Auto Power Range setting, the minimum value of step attenuation is selected according to the available source power level. The table below describes the step attenuation selection based on the setting source power level:

Source Power Range	Step Attenuation
-25 ≤ power ≤ +10 dBm	0 dB
-35 ≤ power < -25 dBm	10 dB
-45 ≤ power < -35 dBm	20 dB
-55 ≤ power < -45 dBm	30 dB
-65 ≤ power < -55 dBm	40 dB
-75 ≤ power < -65 dBm	50 dB
-85 ≤ power < -75 dBm	60 dB

When Power Sweep is selected as sweep type, Step attenuation is selected with the start power condition and the maximum value of the Stop power (>= Start Power) is selected with the step ATT value.

For example:

- Case of Start Power= -26 dBm
  - Step ATT = 10 dB, Stop Power = -26 dBm ~ 0 dBm
- Case of Start Power = -25 dBm
  - Step ATT = 0 dB, Stop Power = -25 dBm ~ + 10 dBm
- If the source power range (start/stop) is located on the two ranges, the Step ATT is set with the latest condition as:
  - After setting Start = -26 dBm, set stop = + 10 dBm, then start is reset with - 25 dBm value.
- Available power differs according to the measurement frequency. For example, at 0 dBm Attenuator range, the specified maximum available power level is as follows and reduces in proportion to inserted Attenuator range:

Frequency	Power Level
300 kHz to 1 MHz	+8 dBm
1MHz to 6 GHz	+10 dBm
6GHz to 8 GHz	+9 dBm
8GHz to 10.5 GHz	+7 dBm
10.5GHz to 15 GHz	+3 dBm
15 GHz to 20 GHz	0 dBm

Correcting attenuation of power level (using power slope feature)

You can use the power slope feature to correct the attenuation of a power level so that it is simply proportional to the frequency (attenuation due to cables and so on), which improves the accuracy of the level actually applied to the DUT.

Turning power slope feature ON/OFF

1. Press **Channel Prev**/**Channel Next** to select the channel for which the power slope feature will be turned OFF.
2. Press **Sweep Setup** key, then click **Power** > **Slope** and select the **ON**/**OFF** setting of the power level.

Setting correction coefficient (correction amount for 1 GHz)

1. Press **Channel Prev**/**Channel Next** to select the channel for which you want to set the correction coefficient.
2. Press **Sweep Setup** key, then click **Power** > **Slope [xxx dB/GHz]** where "xxx" represents the current set value and input the correction coefficient using the **ENTRY** block keys on the front panel.

Setting the Number of Points

The number of points is the number of data items collected in one sweep. It can be set to any number from 2 to 1601 for each channel independently.

- To obtain a higher trace resolution against the stimulus value, choose a larger value for number of points.
- To obtain higher throughput, keep the number of points to a smaller value within an allowable trace resolution.
- To obtain higher measurement accuracy after calibration, perform calibration using the same number of points as in actual measurements.
  - When the upper limit of the number of channels and traces is set to **1Ch / 4 Tr 20001 Points**, the maximum number of measurement points will be 20001.

Setting the number of points

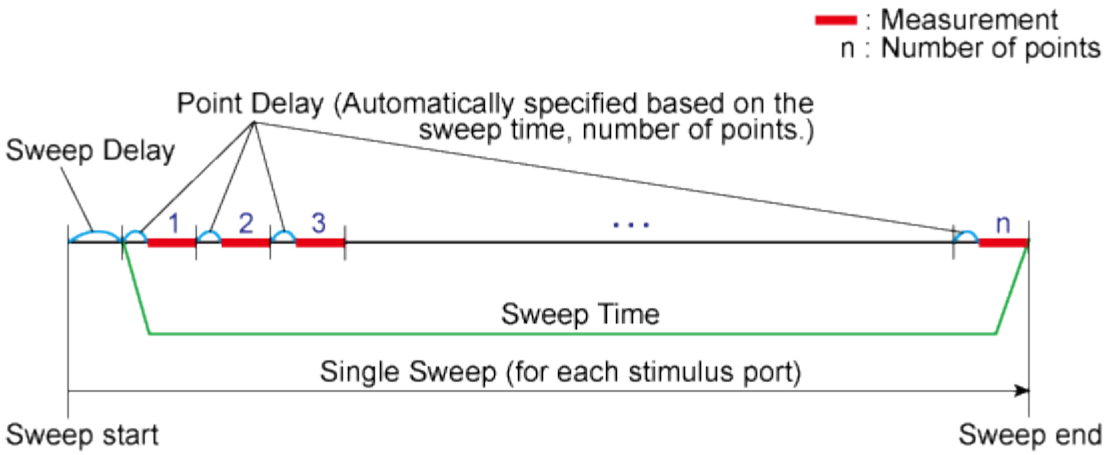
1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key.
3. Click **Points**, then input the desired number of points.

Setting the Sweep Time

Sweep time is the time it takes to complete a sweep for each stimulus (source) port. Two modes are available for setting the sweep time: manual sweep time mode and automatic sweep time mode.

Manual Sweep Time Mode	In this mode, the sweep time is set manually. Once the sweep time is set, changes in measurement conditions will not affect the sweep time as long as it is within the analyzer's capability. If the sweep time becomes lower than the analyzer's lower sweep time limit, the sweep time will be reset to the shortest time within the conditions. If the sweep time exceeds the analyzer's upper sweep time limit, the sweep time will be reset to the longest time within the conditions.
Automatic Sweep Time Mode	The sweep time is always kept to the shortest time possible with the current measurement conditions.

The following figure shows the definitions of the sweep time and the sweep delay time.



e5071c396

Sweep delay is time before starting a sweep for each stimulus (source) port

Setting Up the Sweep Time (Manual Sweep Time Mode)

1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key, then Click **Sweep Time.**
3. Using the **ENTRY** block keys on the front panel, input the desired sweep time (in seconds).

If the previous operation mode was automatic sweep time mode, entering a new sweep time forces the analyzer to switch to manual sweep time mode.

Switching to Automatic Sweep Time Mode

1. Press **Channel Next**/**Channel Prev** keys to select the desired channel.
2. Press **Sweep Setup** key, then click **Sweep Time**, press **0** > **x1**.

By entering zero (seconds), automatic sweep time becomes effective.

Selecting Measurement Parameters

The E5071C allows users to evaluate the DUT (device under test) characteristics by using the following measurement parameters.

- S-parameters
- Absolute
- AUX Input

Other topics about Setting Measurement Conditions

Operation Procedure

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace for which measurement parameters will be set up.
2. Press **Meas**.
3. Click a softkey that corresponds to the desired measurement parameter.

S-parameters

S-parameters (scattering parameters) are used to evaluate how signals are reflected by and transferred through the DUT. An S-parameter is defined by the ratio of two complex numbers and contains information on the magnitude and phase of the signal. S-parameters are typically expressed as follows:

$S_{out\ in}$

out: port number of the DUT from which the signal is output

in: port number of the DUT to which the signal is input

For example, S-parameter S_{21} is the ratio of the output signal of port 2 on the DUT with the input signal of port 1 on the DUT, both expressed in complex numbers.

When the balance-unbalance conversion function is turned on, the Mixed mode S-parameters can be selected. Refer to Evaluating Balanced Devices (balance-unbalance conversion function for more information.

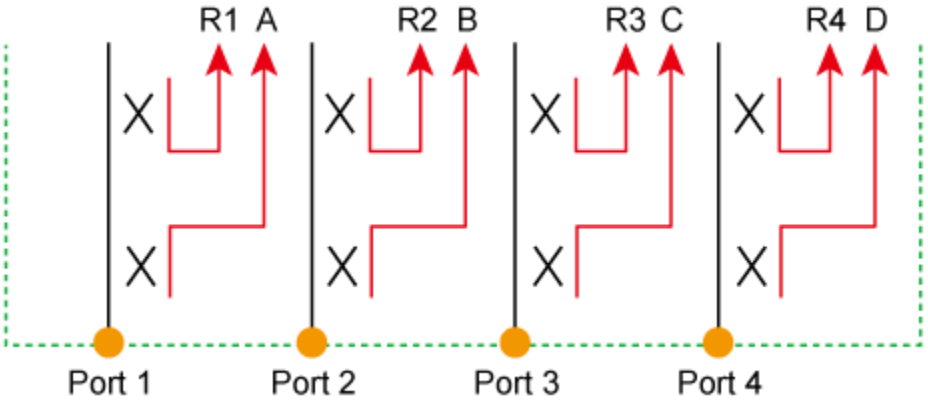
Absolute

Absolute shows the absolute power for reference and received signals on the port.

Softkey	Description
A (n)	Absolute measurement in Port 1, test receiver

B (n)	Absolute measurement in Port 2, test receiver
C (n)	Absolute measurement in Port 3, test receiver
D (n)	Absolute measurement in Port 4, test receiver
R1 (n)	Absolute measurement in Port 1, reference receiver
R2 (n)	Absolute measurement in Port 2, reference receiver
R3 (n)	Absolute measurement in Port 3, reference receiver
R4 (n)	Absolute measurement in Port 4, reference receiver

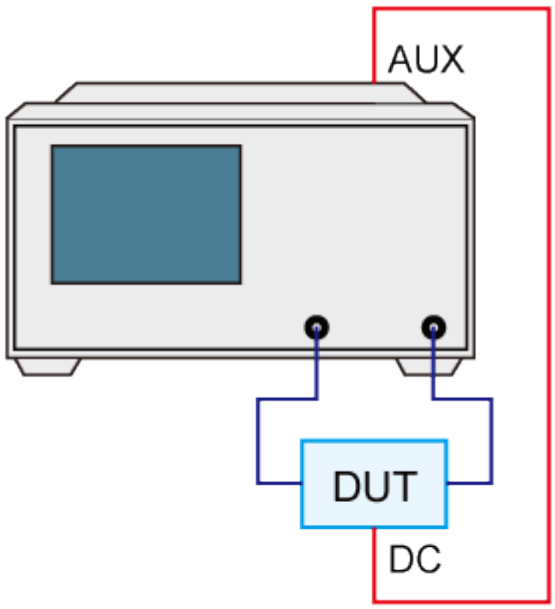
where n in the parentheses is the stimulus port number. For example, R1(1) means the reference level while the signal is output from the port 1, and A(2) means the received signal level into port 1 while the signal is output from the port 2.



AUX Input

The AUX Input Ports can be used to input DC signal for DC signal measurement. This is useful in cases where the DUT (Device Under Test) works on a DC supply and it is required to measure the DC supply along with other measurements of the DUT using the E5071C.

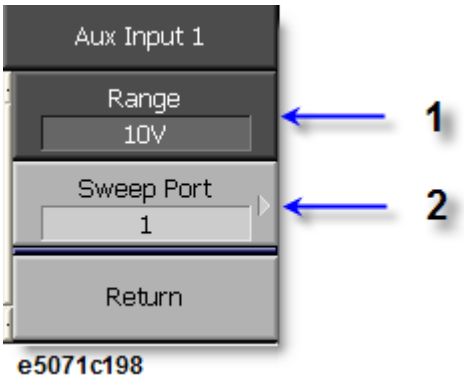
Example of AUX Input Measurement



e5071c209

To use AUX Input Measurement:

1. Select **Aux Input 1** or **Aux Input 2** (depending upon the Aux port used for connection) by pressing the **Meas** key.
2. Select **Range** (*1V* or *10 V*) [**1** in the figure below].



e5071c198

3. Select **Sweep Port** (*1-4*) [**2** in the figure above].
4. Click **Format** > **Real**.
5. Perform measurements as per normal.
6. Sweep Port [**2** in the figure above] specifies the signal output port. For example, if sweep port is set to 1, the AUX input signal is measured while port 1 on the front panel outputs the RF signal.

Selecting a Data Format

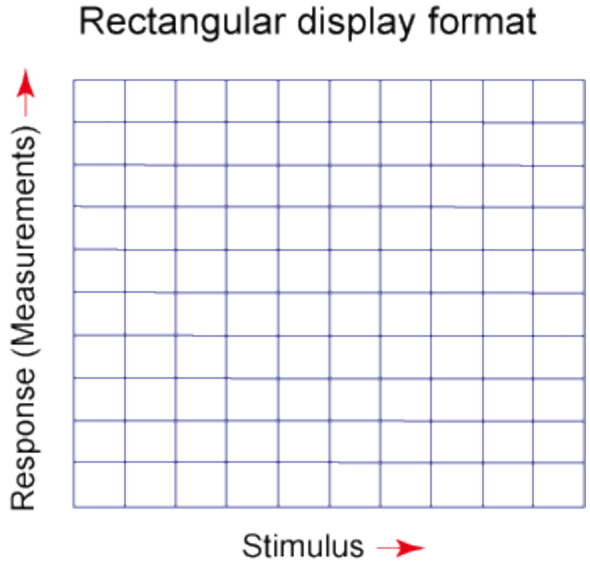
The E5071C allows you to display measured S-parameters by using the following data formats. The data format can be preset to factory settings using the Preset option.

- Rectangular display formats
- Polar format
- Smith chart format

Other topics about Setting Measurement Conditions

Rectangular display formats

Rectangular display formats draw traces by assigning stimulus values (linear scale) to the X-axis and response values to the Y-axis. Eight different formats are available depending on the selection of data for the Y-axis.



e5071c451

Type	Y-axis Data Type	Y-axis Unit	Application Examples
Log magnitude format	Magnitude	dB	• Return loss measurement • Insertion loss measurement (or gain measurement)
Phase format	Phase (displayed in range from -180 ° to +180 °)	Degrees (°)	Measurement of deviation from linear phase

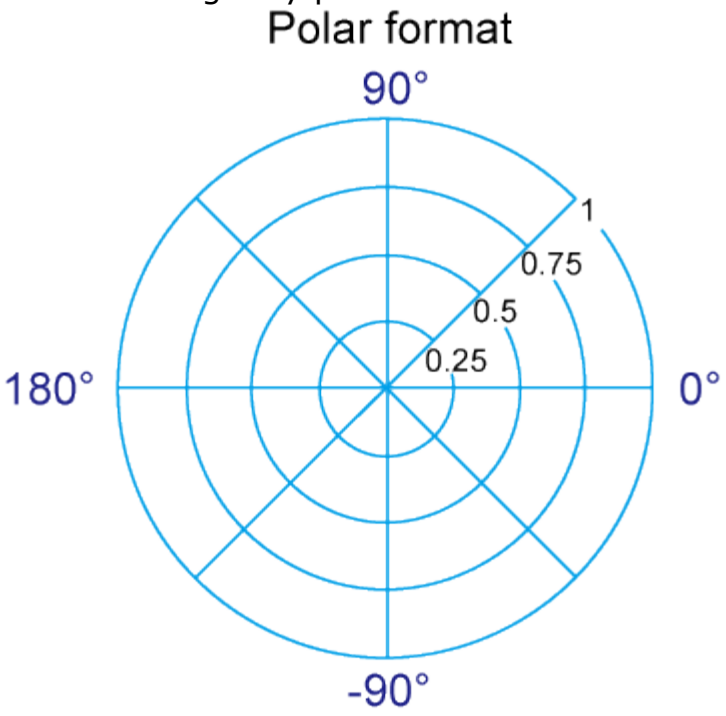
Expanded phase format	Phase (can be displayed above +180 ° and below -180 °)	Degrees (°)	Measurement of deviation from linear phase
Positive phase format	Phase (displayed in range from 0 ° to +360 °)	Degrees (°)	Measurement of deviation from linear phase
Group delay format	Signal transfer delays within the DUT	Seconds (s)	Group delay measurement
Linear magnitude format	Magnitude	(Abstract number)	Reflection coefficient measurement
SWR format	$\frac{1+\rho}{1-\rho}$ (ρ: reflection coefficient)	(Abstract number)	Measurement of standing wave ratio
Real format	Real part of measured complex parameter	(Abstract number)	
Imaginary format	Imaginary part of measured complex parameter	(Abstract number)	

Polar format

In the polar format, traces are drawn by expressing the magnitude as a displacement from the origin (linear) and phase in an angle counterclockwise from the positive X-axis. This data format does not have a stimulus axis, so frequencies must be read by using the marker. The polar format allows users to select one of the following three data groups for displaying the marker response values.

- Linear magnitude and phase (°)
- Log magnitude and phase (°)

- Real and imaginary parts

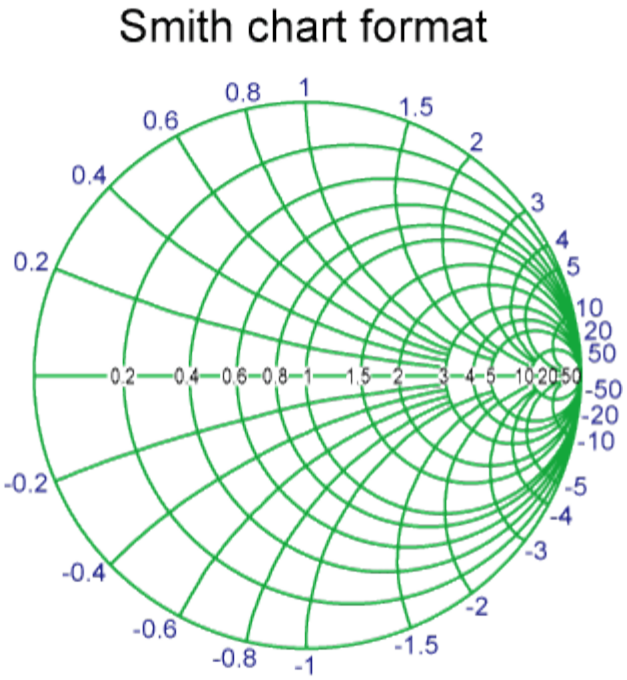


e5071c382

Smith chart format

The Smith chart format is used to display impedances based on reflection measurement data of the DUT. In this format, traces are plotted at the same spots as in the polar format. The Smith chart format allows users to select one of the following five data groups for displaying the marker response values.

- Linear magnitude and phase (°)
- Log magnitude and phase (°)
- Real and imaginary parts
- Resistance (ohm), Reactance (ohm), and inductance (H) or capacitance (F)
- Conductance (S), susceptance (S), and capacitance (F) or inductance (H)



e5071c383

Use the following procedure to select a data format.

1. Press **Channel Next** (or **Channel Prev**) and **Trace Next** (or **Trace Prev**) to select the trace for which the data format will be set.
2. Press **Format.**
3. Press the softkey that corresponds to the desired data format.

Softkey	Function
Log Mag	Selects the log magnitude format
Phase	Selects the phase format
Group Delay	Selects the group delay format
Smith - Lin / Phase	Selects the Smith chart format (with linear magnitude and phase as the marker response values)

Smith - Log / Phase	Selects the Smith chart format (with log magnitude and phase as the marker response values)
Smith - Real / Imag	Selects the Smith chart format (with the real and imaginary parts as the marker response values)
Smith - R + jX	Selects the Smith chart format (with resistance and reactance as the marker response values)
Smith - G + jB	Selects the Smith chart format (with conductance and susceptance as the marker response values)
Polar - Lin / Phase	Selects the polar format (with linear magnitude and phase as the marker response values)
Polar - Log / Phase	Selects the polar format (with log magnitude and phase as the marker response values)
Polar - Real / Imag	Selects the polar format (with the real and imaginary parts as the marker response values)
Lin Mag	Selects the linear magnitude format
SWR	Selects the SWR (standing wave ratio) format
Real	Selects the real format
Imaginary	Selects the imaginary format
Expand Phase	Selects the expanded phase format
Positive Phase	Selects the positive phase format

Setting the Scales

- Auto Scale
- Manual scale adjustment on a rectangular display format

Other topics about Setting Measurement Conditions

Auto scale

The auto scale function is used to tailor each scale (scale/division and the reference line value) automatically in such a way that traces will appear at the proper size on the screen for easy observation.

- The scale data can be preset to factory settings using the Preset option.

Single Trace Auto Scale

Follow the procedure below to perform the auto scale function on a specific trace.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace for which the auto scale function will be performed.
2. Press **Scale** > **Auto Scale**.

Auto Scale on All Traces Within a Channel

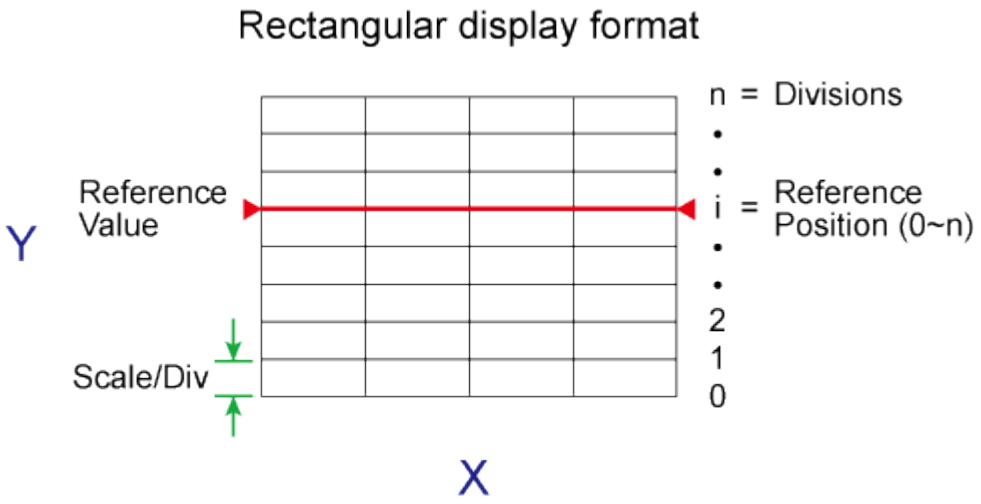
1. Press **Channel Next**/**Channel Prev** keys to select the channel for which the auto scale function will be performed.
2. Press **Scale** > **Auto Scale**.

Manual scale adjustment on a rectangular display format

For a rectangular display format, four parameters are used to manually adjust the scales.

Adjustable feature	Description
Divisions	Defines the number of divisions on the Y-axis. An even number from 4 to 30 must be used. Once set, it is commonly applied to all traces displayed in any rectangular format within that channel.
Scale/Division (Scale/Div)	Defines the number of increments per division on the Y-axis. The value applies only to the active trace.
Reference position	Defines the position of the reference line. The position must be specified using the number assigned to each division on the Y-axis starting at 0 (the least significant) running up to the number of divisions being used (the most significant). The position applies only to the active trace.

Reference line value (Reference Value)	Defines the value corresponding to the reference line. It must be set using the unit on the Y-axis. The reference line value applies only to the active trace.



e5071c385

Manually setting scales on a rectangular display format

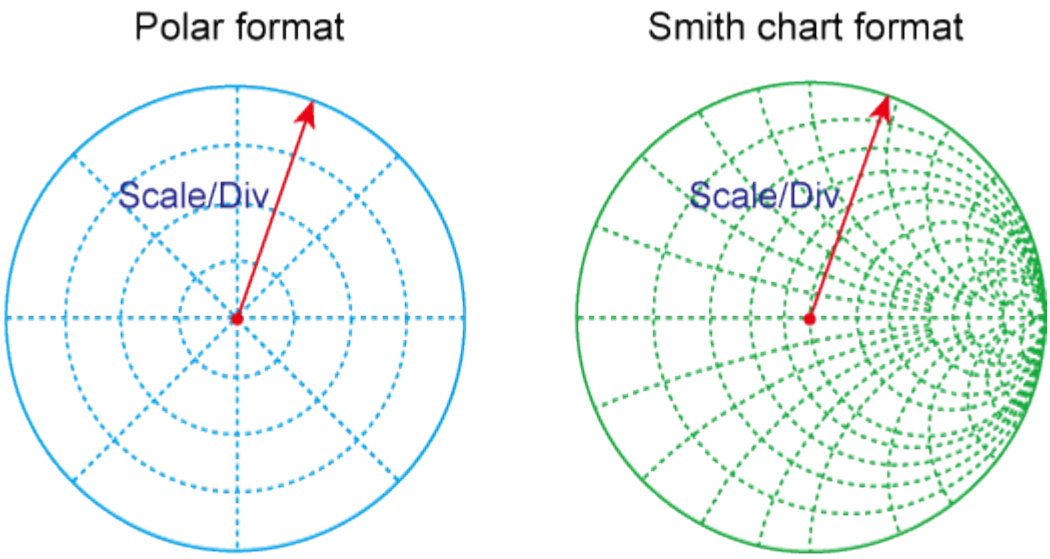
1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace for which scale features will be adjusted.
2. Press **Scale** key.
3. Select the softkey that corresponds to the particular feature that needs to be adjusted.

Softkey	Function
Divisions	Defines the number of divisions on the Y-axis.
Scale/Div	Defines the number of increments per division on the Y-axis.
Reference Position	Defines the position of the reference line.
Reference Value	Defines the value corresponding to the reference line.

- It is also possible to turn off the display of graticule labels. For details, refer to Turning_off_the_display_of_graticule_labels.

Manual scale adjustment on the Smith chart/polar format

Manual scale adjustment on the Smith chart format or the polar format is done by using the displacement (**Scale/Div** of the outermost circle).



e5071c384

Manually setting scales on the Smith chart/polar format

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace for which the scale will be adjusted.
2. Press **Scale** key.
3. Click **Scale/Div**, then input the displacement of the outermost circle.

Setting the value of a reference line using the marker

When using a rectangular display format, it is possible to change the reference line value to be equal to the response value of the *active marker* on the active trace.

Setting the reference line value using the marker

1. Place the active marker on the active trace on the position that corresponds to the new reference line value.
2. Press **Scale** or **Marker Fctn** key.

3. Click **Marker -> Reference** to change the reference line value to the marker response value.
    4. If the reference marker is on and the stimulus value of the active marker is expressed using a value relative to the reference marker, the absolute stimulus value will be used to set the new reference line value.

Setting Window Displays

- Maximizing the specified window trace display
- Turning off the display of graticule labels
- Hiding Frequency Information
- Labeling a Window
- Setting Display Colors
- Setting Display Magnification

Other topics about Setting Measurement Conditions

Maximizing the specified window / trace display

When using multiple channels, it is possible to maximize a specific channel window on the screen. When multiple traces are displayed in a channel window, it is also possible to maximize a specific trace displayed within that channel window.

- The Window/Trace Display data can be preset to factory settings using the Preset option

Maximizing a window

1. Press **Channel Next** (or **Channel Prev**) to select the channel whose window will be maximized.
2. Press **Channel Max** to maximize the channel window.
3. Press **Channel Max** one more time to reduce the window to its previous size.

Maximizing a trace display

1. Press **Channel Next** (or **Channel Prev**) to select the channel to which the trace belongs.
2. Press **Trace Next** (or **Trace Prev**) to select the trace whose display will be maximized.
3. Press **Trace Max** to maximize the trace display.
4. Press **Trace Max** one more time to reduce the display to its previous size.

Turning off the display of graticule labels

When using a rectangular display format, the graph area can be expanded to the left by turning off the display of graticule labels.

Turning off graticule label display

1. Press **Channel Next** (or **Channel Prev**) to select the channel for which graticule label display will be turned on or off.

2. Press **Display**.
3. Click **Graticule Label** to turn graticule label display on or off.

Hiding Frequency Information

You can hide the frequency information from the screen in order to ensure its confidentiality or for other reasons.

Hiding Frequency Information on the Screen

Follow the steps below to hide frequency information on the measurement screen.

1. Press **Display** key.
2. Click **Frequency** to turn off the frequency display.
3. Turning off the frequency display using **Display** > **Frequency** key does not erase the frequency display within the **Stimulus** softkey, which is turned on by pressing **Start**, **Stop**, **Center**, and **Span**. The display of the softkey bar itself can be switched on or off by pressing **Softkey On/Off**.

Hiding Softkey's Frequency Information

You can delete the frequency information from the measurement screen, which changes the frequency information displayed in the Stimulus softkey and the data entry area for Hz unit to asterisks (***).

1. Press **System** key.
2. Click **Service Menu**, then click **Security Level** and select any of the following options for the frequency display.

Softkey	Function
None	Displays the frequency information.
Low	Hides the frequency information with a series of asterisks. This can be turned to OFF by the Security Level menu.
High	Hides the frequency information with a series of asterisks. This cannot be turned to OFF by the Security Level menu. Resetting to OFF is only possible by executing Preset or Recall. o **Save/Recall** > **Save Trace Data** & **Save/Recall** > **Save SnP** is inactive when **Security Level** is set to **Low**/**High**.

Labeling a window

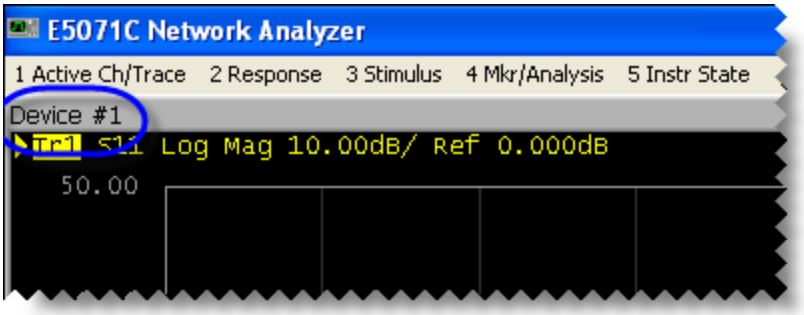
It is possible to assign a unique name to a channel and display it on the screen. This feature is useful in saving and/or printing measurement result for future reference.

Labeling a window

1. Press **Channel Next**/**Channel Prev** keys to select the channel to be labeled.
2. Press **Display** key.
3. Click **Edit Title Label**,then the title label input dialog box appears.

e5071c060

3. Using the keys in the dialog box, type a label and click **Enter**.
4. Click **Title Label** to turn on the title display. The title will appear within a frame at the top of the channel window.



e5071c195

Setting Display Colors

Selecting display mode

You can select the display mode of the LCD display from two modes: normal display (background: black) or inverted display (background: white). In normal display, the colors of items are preset so that you can recognize them easily on the display of the instrument. On the other hand, in inverted display, they are preset to colors obtained by nearly inverting the default settings of the normal display so that you can use data easily when storing it into a graphic file.

The selection procedure is as follows:

1. Press **Display**.
2. Click **Invert Color** to select the display color. OFF indicates the normal display; ON the inverted display.

Setting display color for each item

You can set the display color to the normal display or the inverted display separately for each of the following items.

- Data/memory trace

- Labels and lines of graphs
- File display of the limit test and limit lines
- Background

You set the color of each item by specifying the amounts of red (R), green (G), and blue (B) contained in the color. You can specify each level of R, G, and B in 6 steps (0 to 5). Therefore, 216 colors in total are available by combining them. The table below shows the R, G, and B values for the main colors as a reference.

	R	G	B		R	G	B		R	G	B
White	5	5	5	**Gray**	2	2	2	**Black**	0	0	0
Light red	5	3	3	**Red**	5	0	0	**Dark Red**	2	0	0
Light yellow	5	5	3	**Yellow**	5	5	0	**Dark Yellow**	2	2	0
Light green	3	5	3	**Green**	0	5	0	**Dark Green**	0	2	0
Light cyan	3	5	5	**Cyan**	0	5	5	**Dark cyan**	0	2	2
Light blue	3	3	5	**Blue**	0	0	5	**Dark Blue**	0	0	2
Light magenta	5	3	5	**Magenta**	5	0	5	**Dark Magenta**	2	0	2

The setting procedure is as follows:

1. Press **System** > **Misc Setup** > **Display Setup** > **Color Setup**
2. Click **Normal** (for normal display) or **Invert** (for inverted display).
3. Click the softkey corresponding to the item for which you want to set the display color.

Softkey	Function
Data Trace 1 to 16	Specifies the data trace of traces 1 to 16
Mem Trace 1 to 16	Specifies the memory trace of traces 1 to 16
Graticule Main	Specifies the graticule label and the outer lines of graphs

Graticule Sub	Specifies the grid of graphs
Limit Fail	Specifies the fail display in the limit test result
Limit Line	Specifies the limit line
Background	Specifies the background

4. Click **Red** (or, **Green**, or **Blue**).
5. Select the amount of the selected color from 0 to 5.

Resetting the display colors to the factory state

You can reset the display colors in normal display and inverted display to the preset factory state.

The selection procedure is as follows:

1. Press **System** > **Misc Setup** > **Display Setup** > **Color Setup.**
2. Click **Normal** (for normal display) or **Invert** (for inverted display).
3. Click **Reset Color** > **OK**.

Setting Display Magnification

You can reset the display magnification to Small, Normal or Large.

The selection procedure is as follows:

1. Press **System** > **Misc Setup** > **Display Setup** > **Magnification.**
2. Click **Normal, Small** or **Large**.

Calibration

Calibration

Overview

- Measurement Errors and their Characteristics
- Calibration Types and Characteristics
- Checking Calibration Status
- Clear Calibration

Basic Calibration

- Selecting Calibration Kit
- OPEN/SHORT Response Calibration (reflection test)
- THRU Response Calibration (transmission test)
- Enhanced Response Calibration
- 1-Port Calibration (reflection test)
- Full 2-Port Calibration
- Full 3-Port Calibration
- Full 4-Port Calibration

Calibration with ECal (Electronic Calibration)

- ECal (Electronic Calibration)
- ECal Driver Installation
- 1-Port Calibration Using a 2-Port ECal Module
- Calibration Using 4-port ECal

If you have 2 port Ecal:

- Full 2-Port Calibration Using the 2-Port ECal Module
- Full 3-Port and Full 4-Port Calibration using 2-Port ECal

Advanced Calibration with ECal

- Improving Calibration Accuracy along with ECal
- Confidence Check on Calibration Coefficients Using ECal
- Turning off ECal Auto-detect Function
- User-characterized ECal

Advanced Calibration

- Setting the Trigger Source for Calibration
- Sliding Offset and Load Calibration

- Modifying Calibration Kit Definition
- Specifying Different Standard for Each Frequency
- Receiver Calibration
- Power Calibration
- Partial Overwrite
- Simplified Calibration
- Adapter Removal-Insertion
- Unknown Thru Calibration
- Performing 8 term Calibration using External PC

Mixer Calibration

- Scalar-Mixer Calibration
- Vector-Mixer Calibration

TRL Calibration

- 2-port TRL calibration
- 3-port TRL calibration
- 4-port TRL calibration

Measurement Errors and their Characteristics

- Overview
- Drift Errors
- Random Errors
- Systematic Errors

Other topics about Calibration

Overview

It is important to understand the factors contributing to measurement errors in order to determine the appropriate measures that should be taken to improve accuracy. Measurement errors are classified into three categories:

Drift Errors

Drift errors are caused by deviations in the performance of the measuring instrument (measurement system) that occur after calibration. Major causes are the thermal expansion of connecting cables and thermal drift of the frequency converter within the measuring instrument. These errors may be reduced by carrying out frequent calibrations as the ambient temperature changes or by maintaining a stable ambient temperature during the course of a measurement.

Random Errors

Random errors occur irregularly in the course of using the instrument. Since random errors are unpredictable, they cannot be eliminated by calibration. These errors are further classified into the following sub-categories depending on their causes.

- Instrument noise errors
- Switch repeatability errors
- Connector repeatability errors

Instrument noise errors

Instrument noise errors are caused by electric fluctuations within components used in the measuring instrument. These errors may be reduced by increasing the power of the signal supplied to the DUT, narrowing the IF bandwidth, or enabling sweep averaging.

Switch repeatability errors

Switch repeatability errors occur due to the fact that the electrical characteristics of the mechanical RF switch used in the measuring instrument change every time it is switched on. These errors may be reduced by carrying out measurements under conditions in which no switching operation takes place.

(You don't need to worry about these errors since the E5071C does not have mechanical RF switches).

Connector repeatability errors

Connector repeatability errors are caused by fluctuations in the electrical characteristics of connectors due to wear. These errors may be reduced by handling connectors with care.

Systematic Errors

Systematic errors are caused by imperfections in the measuring instrument and the test setup (cables, connectors, fixtures, etc.). Assuming that these errors are repeatable (i.e., predictable) and their characteristics do not change over time, it is possible to eliminate them mathematically at the time of measurement by determining the characteristics of these errors through calibration. There are six types of systematic errors, as follows.

Errors caused by signal leaks in the measuring system:

- Directivity
- Isolation (cross-talk)

Errors caused by reflections in the measuring system:

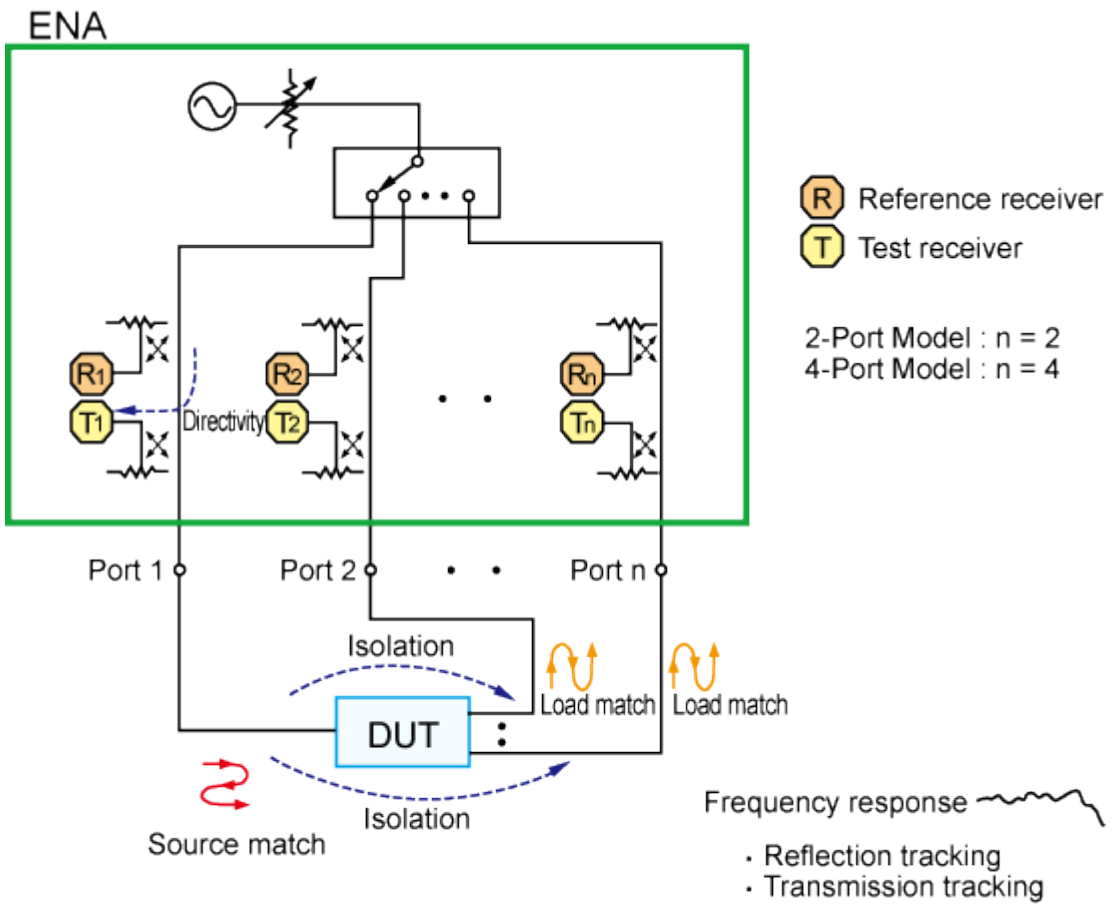
- Source match
- Load match

Errors caused by the frequency response of the receiver within the measuring instrument:

- Reflection tracking
- Transmission tracking

The E5071C has two receivers for each test port: the reference receiver and the test receiver (transmission measurement or reflection measurement). You can perform measurements with both of these receivers at the same time.

E5071C port architecture and systematic errors



e5071c240

Directivity error (Ed)

Directivity errors are caused by the fact that, in a reflection measurement, signals other than the reflection signal from the DUT are received by receiver T1 through the directivity coupler. When a certain port is a stimulus port, this error can be defined as a constant value for each stimulus port because the state of the termination at the other ports does not change. The number of directivity errors of the E5071C is the number of stimulus ports you use.

Ed1	Directivity error of port 1
Ed2	Directivity error of port 2
Ed3	Directivity error of port 3
Ed4	Directivity error of port 4

Isolation error (Ex)

An isolation error (crosstalk error) is caused by signals other than the transmission signal of the DUT leaking to the test receiver of the transmission measurement port in transmission measurements. When a certain port is a stimulus port, an isolation error is defined for each of the other ports. Therefore, the number of isolation errors for the E5071C is the total number of combinations of stimulus ports and response ports.

Ex21, Ex31, and Ex41	Isolation error when port 1 is a stimulus port
Ex12, Ex32, and Ex42	Isolation error when port 2 is a stimulus port
Ex13, Ex23, and Ex43	Isolation error when port 3 is a stimulus port
Ex14, Ex24, and Ex34	Isolation error when port 4 is a stimulus port

Source match error (Es)

A source match error is caused when the reflection signal of the DUT reflects at the signal source and enters the DUT again. When a certain port is a stimulus port, this error can be defined as a constant value for each stimulus port because the state of the signal source switch does not change. The number of source match errors in the E5071C is equivalent to the number of stimulus ports you use.

Es1	Source match error of port 1
Es2	Source match error of port 2
Es3	Source match error of port 3
Es4	Source match error of port 4

Load match error (El)

A load match error is caused when part, but not all, of the signal transmitted in the DUT reflects at a response port is measured by the receiver of the response port. When a certain port is a stimulus port, a load match error is defined for each of the other ports. Therefore, the number of load match errors for the E5071C is the total number of combinations of stimulus ports and response ports.

El21, El31, and El41	Load match error when port 1 is a stimulus port
El12, El32, and El42	Load match error when port 2 is a stimulus port
El13, El23, and El43	Load match error when port 3 is a stimulus port
El14, El24, and El34	Load match error when port 4 is a stimulus port

Reflection tracking error (Er)

A reflection tracking error is caused by the difference in frequency response between the test receiver and the reference receiver of a stimulus port in reflection measurements. This error can be defined as a constant value for each stimulus port because the combination of the test receiver and the reference receiver of a stimulus port is always the same. The number of reflection tracking errors for the E5071C is simply the number of stimulus ports you use.

Er1	Reflection tracking error of port 1
Er2	Reflection tracking error of port 2
Er3	Reflection tracking error of port 3
Er4	Reflection tracking error of port 4

Transmission tracking error (Et)

A transmission tracking error is caused by the difference in frequency response between the test receiver of a response port and the reference receiver of a stimulus port in transmission measurements. When a certain port is a stimulus port, a transmission tracking error is defined for each of the other ports. Therefore, the number of transmission tracking errors for the E5071C is the total number of combinations of stimulus ports and response ports.

Et21, Et31, and Et41	Transmission tracking error when port 1 is a stimulus port
Et12, Et32, and Et42	Transmission tracking error when port 2 is a stimulus port
Et13, Et23, and Et43	Transmission tracking error when port 3 is a stimulus port
Et14, Et24, and Et34	Transmission tracking error when port 4 is a stimulus port

Calibration Types and Characteristics

The table shows the different types of calibrations and the features of each method.

Calibration Method	Standard (s) Used	Corrected Error Factor	Measurement Parameters	Characteristics
No calibration	None	None	All parameters	• Low accuracy • Calibration not required
Response Calibration	• OPEN or SHORT • LOAD	Following 2 error terms: • Reflection Tracking (Er) • Directivity (Ed)	S11(Reflection characteristics at 1 port)	• Medium-level accuracy • Quick calibration • Isolation calibration improves the accuracy in a reflection measurement of a DUT with high return loss
	• THRU • LOAD	Following 2 error terms: • Transmission Tracking (Et) • Isolation (Ex)	S21 (1 direction transmission characteristics at 2 ports)	• Medium-level accuracy • Quick calibration • Isolation calibration improves the accuracy in a transmission measurement of a device with high insertion loss
1-Port Calibration	ECal module (2-port/4-port)	Following 3 error terms: • Directivity (Ed) • Source Match (Es) • Reflection Tracking (Er)	S11 (Reflection characteristics at 1 port)	• 1-port measurement with the highest degree of accuracy • Quick calibration with low

				chance of operator error
	• OPEN • SHORT • LOAD			• Highly accurate 1-port measurement
Enhanced Response Calibration	Ecal module (2-port)	Following 5 error terms: • Directivity (Ed1) • Isolation (Ex21) • Source Match (Es1) • Transmission Tracking (Et21) • Reflection Tracking (Er1)	S11, S21 (1 direction transmission/Reflection characteristics at 2 ports)	• Highly accurate 2-port measurement (higher than response calibration) • Quick calibration with low chance of operator error
	• OPEN • SHORT • LOAD • THRU			• Highhly accurate 2-port measurement (higher than response calibration)
Full 2-Port Calibration	ECal module (2-port/4-port)	Following 12 error terms: • Directivity (Ed1,Ed2) • Isolation (Ex21,Ex12) • Source Match (Es1,Es2) • Load Match (El12,El21) • Transmission Tracking (Et21,Et12) • Reflection Tracking (Er1,Er2)	S11,S21,S12,S22 (All S-parameters at 2 ports)	• Highly accurate 2-port measurement • Quick calibration with low chance of operator error
	• OPEN • SHORT • LOAD • THRU			• Highly accurate 2-port measurement
Full 3-Port Calibration	ECal module (2-port/ 4-port)	Following 27 error terms: • Directivity (Ed1,Ed2,Ed3) • Isolation (Ex21,Ex31,Ex12,Ex32, Ex13,Ex23) • Source Match	S11,S21,S31,S12,S22 ,S32,S13,S23,S33 (All S-parameters at 3 ports)	• Highly accurate 3-port measurement • Quick calibration with low chance of operator error

	• OPEN • SHORT • LOAD • THRU	(Es1,Es2,Es3) • Load Match (El21,El31,El12,El32,El 13,El23) • Transmission Tracking (Et21,Et31,Et12,Et32,Et 13,Et23) • Reflection Tracking (Er1,Er2,Er3)		• Highly accurate 3-port measurement
Simplified full 3-port calibration	• OPEN • SHORT • LOAD • THRU	Same as full 3-port calibration	S11, S21, S31, S12, S22, [S32], S13, [S23], S33 (Part of thru measurement can be skipped)	• High-accuracy 3-port measurement • Simpler procedure by skipping thru measurement
Full 4-Port Calibration	ECal module (2-port/ 4-port)	Following 48 error terms: • Directivity (Ed1,Ed2,Ed3,Ed4) • Isolation (Ex21,Ex31,Ex41,Ex12, Ex32,Ex42 ,Ex13,Ex23,Ex43,Ex14, Ex24,Ex34) • Source Match (Es1,Es2,Es3,Es4) • Load Match (El1,El2,El3,El4) • Transmission Tracking (Et21,Et31,Et41,Et12,Et 32, Et42,Et13,Et23,Et43,Et 14 ,Et24,Et34) • Reflection Tracking (Er1,Er2,Er3,Er4)	S11,S21,S31,S41,S12 ,S22,S32,S42, S13,S23,S33,S43,S14 ,S24,S34,S44 (All S-parameters at 4 ports)	• Highly accurate 4-port measurement • Quick calibration with low chance of operator error
	• OPEN • SHORT • LOAD • THRU			• Highly accurate 4-port measurement
Simplified full 4-port calibration	• Open • Short • Load • Thru	Same as full 4-port calibration	S11, S21, S31, [S41], S12, S22, [S32], [S42], S13, [S23], S33, S43, [S14], [S24], S34, S44 (Part of thru measurement can be skipped)	• High-accuracy 4-port measurement • Simpler procedure by skipping thru measurement
2-Port TRL Calibra	• Reflection (OPEN or	Following 12 error terms: • Directivity (Ed1,Ed2)	S11,S21,S12,S22 (All S-parameters at 2	• Highly accurate 2-port

tion	SHORT) • THRU • LINE • MATCH	• Source Match (Es1,Es2) • Load Match (El1,El2) • Transmission Tracking (Et21,Et12) • Reflection Tracking (Er1,Er2)	ports)	measurement • Effective for non-coaxial device measurement
3-Port TRL Calibration	• Reflection (OPEN or SHORT) • THRU • LINE • MATCH	Following 27 error terms: • Directivity (Ed1,Ed2,Ed3) • Source Match (Es1,Es2,Es3) • Load Match (El21,El31,El12,El32, El13,El23) • Transmission Tracking (Et21,Et31,Et12,Et32, Et13,Et23) • Reflection Tracking (Er1,Er2,Er3)	S11,S21,S31,S12,S22 ,S32,S13,S23,S33 (All S-parameters at 3 ports)	• Highly accurate 3-port measurement • Effective for non-coaxial device measurement
Simplified 3-port TRL calibration	• Reflection (open or short) • Thru • Line • Match	Same as 3-port TRL calibration	S11, S21, S31, S12, S22, [S32], S13, [S23], S33 (Part of thru (or line) and line (or match) measurement can be skipped)	• High-accuracy 3-port measurement Effective for non-coaxial device measurement • Simpler procedure by skipping thru/line/match measurement
4-Port TRL Calibration	• Reflection (OPEN or SHORT) • THRU • LINE • MATCH	Following 48 error terms: • Directivity (Ed1,Ed2,Ed3,Ed4) • Source Match (Es1,Es2,Es3,Es4) • Load Match (El1,El2,El3,El4) • Transmission Tracking (Et21,Et31,Et41,Et12, Et32,Et42, Et13,Et23,Et43,Et14, Et24,Et34)	S11,S21,S31,S41,S12 ,S22,S32,S42,S13, S23,S33,S43,S14,S24 ,S34,S44 (All S-parameters at 4 ports)	Highly accurate 4-port measurement Effective for non-coaxial device measurement

		• Reflection Tracking (Er1,Er2,Er3,Er4)		
Simplified 4-port TRL calibration	• Reflection (open or short) • Thru • Line • Match	Same as 4-port TRL calibration	S11, S21, S31, [S41], S12, S22, [S32], [S42], S13, [S23], S33, S43, [S14], [S24], S34, S44 (Part of thru (or line) and line (or match) measurement can be skipped)	• High-accuracy 4-port measurement Effective for non-coaxial device measurement • Simpler procedure by skipping thru/line/match measurement

Other topics about Calibration

Checking Calibration Status

- Execution Status of Error Correction for Each Channel
- Execution Status of Error Correction for Each Trace
- Acquisition Status of Calibration Coefficient for Each Channel

Other topics about Calibration

Execution Status of Error Correction for Each Channel

You can check the execution status of error correction for each channel with the error correction status.

The error correction status is indicated in the channel status bar in the lower part of the window by the symbols in the below table.

Symbol	Execution status of error correction
Cor (displayed in blue)	Error correction: On (enabled for all traces)
Cor (displayed in gray)	Error correction: On (enabled for some traces)
Off (displayed in gray)	Error correction: Off
--- **(displayed in gray)**	Error correction: On (no calibration data)
C? (displayed in blue)	Error correction: On (Interpolation is being executed or the IF bandwidth, power level, power range, sweep time, sweep delay time, sweep mode, or sweep type is different from that when the calibration was executed.)
C! (displayed in blue)	Error correction: On (Extrapolation is being executed.) In the option TDR, even after making deskew and calibration, this is displayed.

An asterisk (*) in the upper-right of the softkey indicates that the existing calibration coefficient will be cleared if you select the test port and execute acquisition of the calibration coefficient (clicking **Done**).

Execution Status of Error Correction for Each Trace

You can check the status of the error correction actually executed for each trace with the trace status area.

For a trace for which error correction is executed, the applied calibration type is indicated in the trace status area by the symbols in the table below.

Symbol	Calibration type
RO	Open response calibration
RS	Short response calibration
RT	Thru response calibration
ER	Enhanced response calibration
F1	1-port calibration
F2	Full 2-port calibration/2-port TRL calibration
F3	Full 3-port calibration/3-port TRL calibration
F4	Full 4-port calibration/4-port TRL calibration

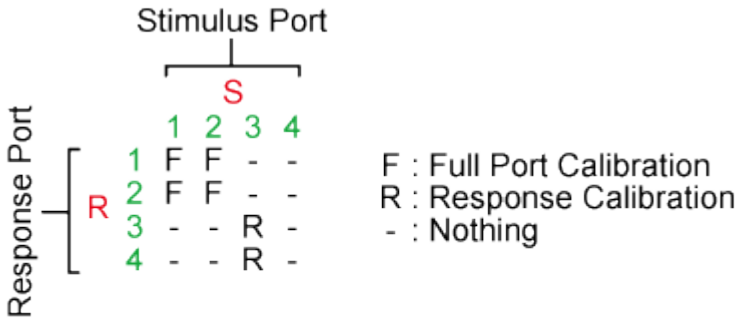
If none of the symbols described above is displayed, error correction is not executed for the trace.

Acquisition Status of Calibration Coefficient for Each Channel

You can check the acquisition status of the calibration coefficient for each channel with the calibration property.

The calibration property displays the acquisition status of the calibration coefficient between test ports for each channel in matrix format. The following figure shows an example of when the calibration coefficients have been acquired for the full 2-port calibration between test ports 1 and 2, the response calibration for test port 3, and the response calibration (THRU) between test ports 4 and 3.

Example of calibration property display



e5071c355

- The simplified full 3/4-port calibration and the simplified 3/4-port TRL calibration are not discriminated from the normal full-port and TRL calibrations, and **F** is displayed.

Procedure to turn on/off calibration property display

Follow these steps to turn on/off the calibration property display.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to turn on/off the calibration property display.
2. Press **Cal** key.
3. Click **Property**. Each press toggles the on/off setting.

Conditions for clearing already acquired calibration coefficients

In the following cases, already acquired calibration coefficients are cleared.

- Executing preset clears all calibration coefficients.
- If S parameters required to calculate the calibration coefficient for the specified calibration type and test ports and those required for the existing calibration coefficient overlap, executing the acquisition of the calibration coefficient (measuring necessary data and then clicking the **Done** softkey) clears the calibration coefficient for which necessary S parameters overlap. Taking the Example of calibration property display as an example, if you acquire the calibration coefficient of the 1-port calibration for test port 4, neither calibration coefficient is cleared. On the other hand, if you acquire the calibration coefficient for the full 2-port calibration between test ports 2 and 3, the calibration coefficient of the full 2-port calibration between test ports 1 and 2 and that of the response calibration for test port 3 are cleared.

Clear Calibration

This softkey clears the user calibration data. When Calibration is done for a particular DUT, the data get stored in the E5071C. To clear this data, **Clear Calibration** can be used which removes the User calibration data from the E5071C.

Other topics about Calibration

Basic Calibrations

Selecting Calibration Kit

- Overview
- Procedure

Other topics about Basic Calibration

Overview

Before executing calibration, you need to select a calibration kit.

If you use a calibration kit other than a predefined one, you need to define it. If the connector type of the standard of the calibration kit you use has polarity (the distinction between male and female), you need to change the standard class definition of the calibration kit depending on the standard you actually use. For more information, see Modifying Calibration Kit Definition.

- If you select a predefined calibration kit, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.
- System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to select the calibration kit.
2. Press **Cal** Key.
3. Click **Cal Kit**, then select the calibration kit.
4. If the name (label) of the calibration kit has been changed, the label is displayed as the softkey.
5. An asterisk (*) on the upper right of the softkey corresponding to a predefined calibration kit indicates that its definition value has been changed from the factory setting by the user.

OPEN/SHORT Response Calibration (reflection test)

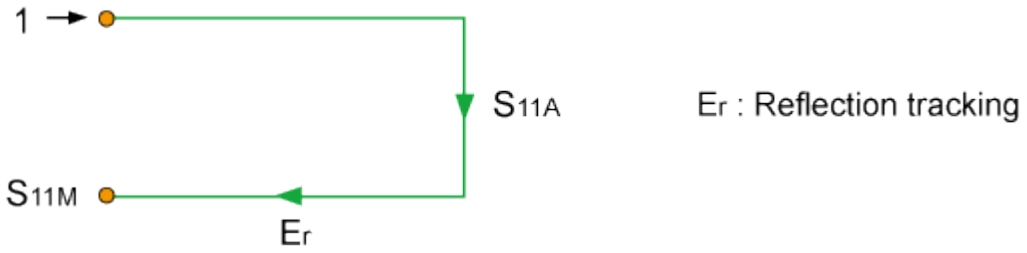
- Overview
- Procedure

Other topics about Basic Calibration

Overview

In OPEN or SHORT response calibration, calibration data are measured by connecting an OPEN or SHORT standard, respectively, to the desired test port. For frequency response, these calibrations effectively eliminate the reflection tracking error from the test setup reflection test using that port.

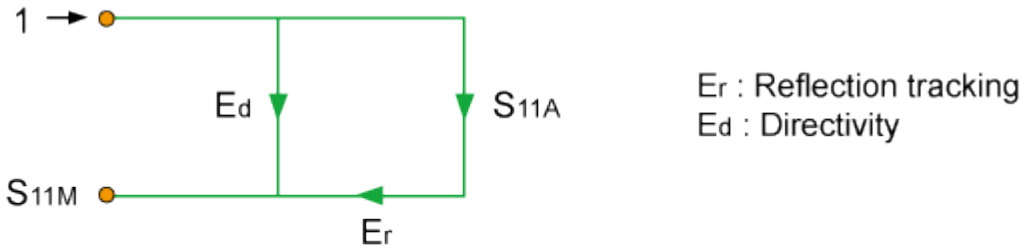
1-Port error model (OPEN/SHORT response)



e5071c398

It is also possible to carry out isolation calibration with a LOAD standard during OPEN/SHORT response calibration. An isolation calibration will eliminate the directivity error from the test setup in a reflection test using that port

1-Port error model (OPEN/SHORT response + isolation)



e5071c399

1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.

Procedure

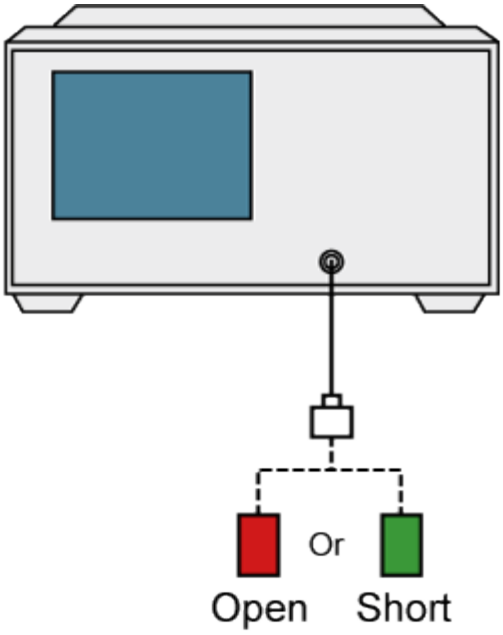
1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.

2. Press **Cal** key.
3. Click **Calibrate**.
4. Select **OPEN or SHORT response calibration**.

Softkey	Function
Response (Open)	Displays softkeys for performing an OPEN response calibration (response calibration with an OPEN standard)
Response (Short)	Displays softkeys for performing a SHORT response calibration (response calibration with a SHORT standard)

5.
6. Click **Select Port**.
7. Select the test port upon which you will perform OPEN/SHORT response calibration.
8. Connect a calibration standard (OPEN or SHORT) to the selected test port (connector to which the DUT is to be connected).
9. Click **Open** or **Short** to start the calibration measurement.
10. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.
11. Connect a LOAD standard to the selected test port (connector to which the DUT is to be connected).
12. Click **Load (Optional)** to start the measurement on the LOAD standard.
13. Click **Done** to terminate the response calibration (and the LOAD isolation calibration) process. Upon pressing this key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

Connecting standards in OPEN/SHORT Response calibration



e5071c318

THRU Response Calibration (transmission test)

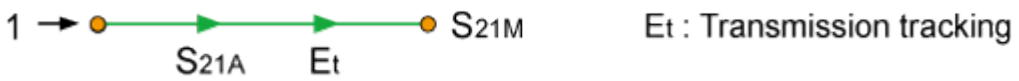
- Overview
- Procedure

Other topics about Basic Calibration

Overview

In THRU response calibration, calibration data are measured by connecting a THRU standard to the desired test port. This calibration effectively eliminates the frequency response transmission tracking error from the test setup in a transmission test using that port.

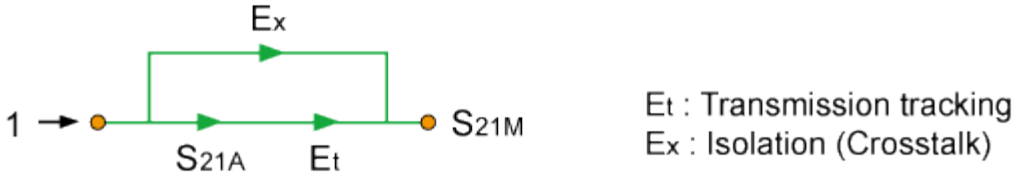
2-Port error model (THRU response)



e5071c400

It is also possible to carry out an isolation calibration using a LOAD standard in the process of THRU response calibration. An isolation calibration will eliminate isolation error (crosstalk error) from the test setup in a transmission test using that port.

2-Port Error model (THRU response + isolation)



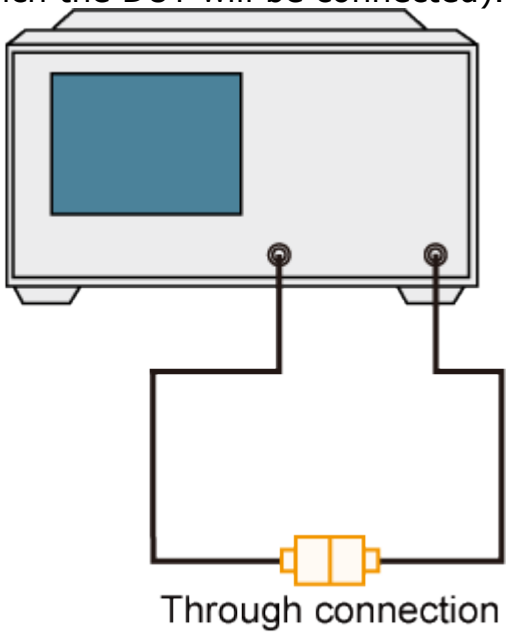
e5071c401

1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **Response (Thru)** > **Select Ports**.

4. Select the test ports (and corresponding S parameters) upon which a THRU response calibration is to be performed.
5. Make a between the selected test ports (between the connectors to which the DUT will be connected).

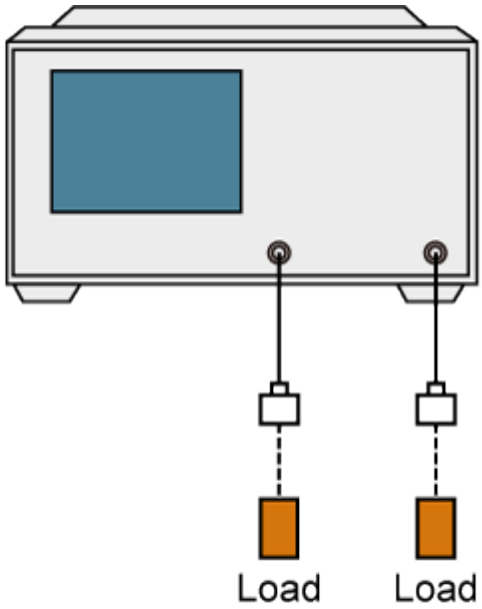


e5071c317

6. Click **Thru** to start the calibration measurement.
    - The sweep is performed twice in the one thru measurement.
7. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.

6.

    a. Connect a LOAD standard to each of the two selected test ports (connectors to which the DUT is to be connected).



e5071c319

b. Click **Isolation (Optional)** to start the calibration measurement.

7. Click **Done** to terminate the response calibration (and the LOAD isolation calibration) process. Upon pressing this key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

Enhanced Response Calibration

- Overview
- Procedure

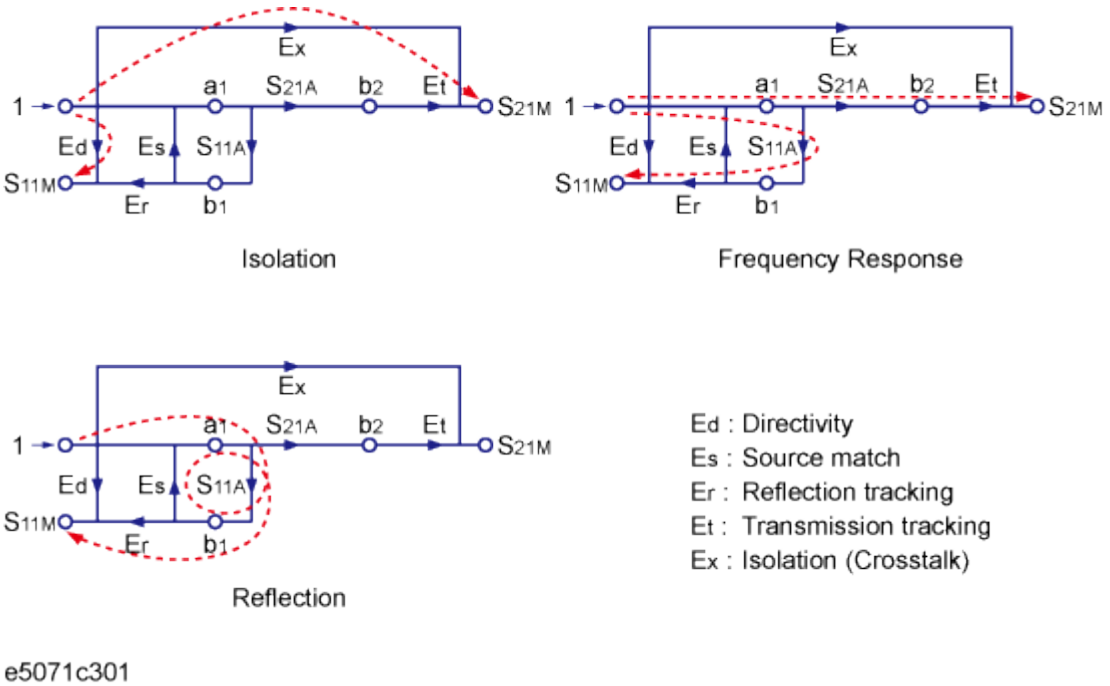
Other topics about Basic Calibration

Overview

In enhanced response calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, or a LOAD standard to the output port (or a THRU standard between two ports).This calibration effectively eliminates the directivity error, crosstalk, source match error, frequency response reflection tracking error, and frequency response transmission tracking error from the test setup in a transmission or reflection test using those ports .

1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.

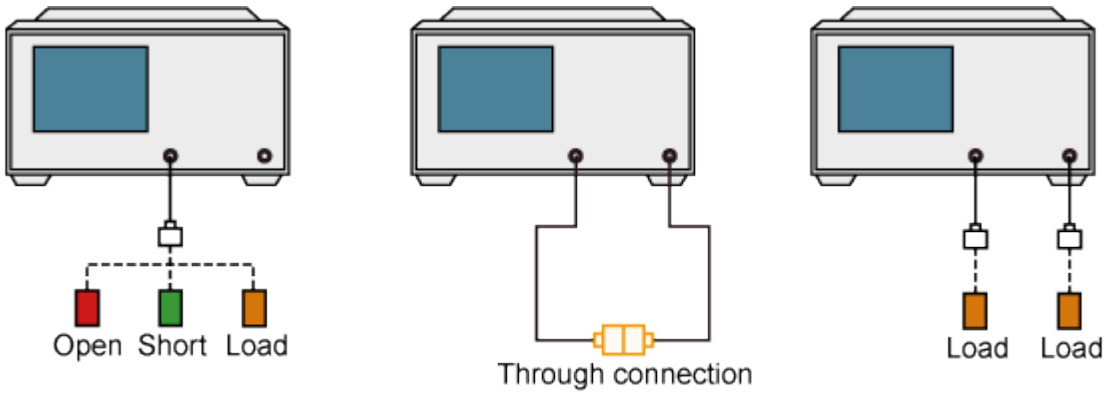
2-Port Error Model (Enhanced Response)



e5071c301

Procedure

Connecting the Standard at Enhanced Response Calibration



e5071c332

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate**.
4. Click **Enhanced Response**.
5. Click **Ports** to select the test ports on which an enhanced response calibration will be performed.
6. Connect an OPEN calibration standard to the output port.
7. Click **Open** to start the calibration measurement.
    - If you select the calibration kit which has different calibration definitions for each gender, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.
8. Disconnect the OPEN calibration standard and replace it with a SHORT calibration standard.
9. Click **Short** to start the calibration measurement.
10. Disconnect the SHORT calibration standard and replace it with a LOAD standard.
11. Click **Load** to start the calibration measurement.
12. Make a THRU connection between the two ports.
13. Click **Thru** to start the calibration measurement.
14. If an isolation calibration must be performed using a LOAD standard, follow the procedure below:

13.

a. Connect a LOAD standard to the two test ports.

b. Click **Isolation (Optional)** to start the calibration measurement.

c. Click **Return**.

14. Click **Done** to terminate the enhanced response calibration process. Upon pressing the key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

1-Port Calibration (reflection test)

- Overview
- Procedure

Other topics about Basic Calibration

Overview

In 1-port calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, and a LOAD standard to the desired test port. This calibration effectively eliminates the frequency response reflection tracking error, directivity error, and source match error from the test setup in a reflection test using that port.

e5071c402

1-Port error model (1-port calibration)

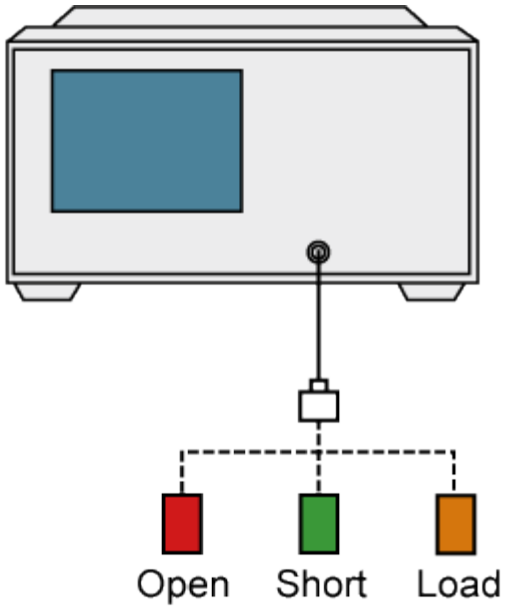
1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **1-Port Cal** > **Select Port**.
4. Select a test port (and corresponding S parameter) on which 1-port calibration will be performed.
5. Connect an OPEN calibration standard to the selected test port (connector to which the DUT is to be connected).
6. Click **Open** to start the calibration measurement.
    - If you select the calibration kit which has different calibration definitions for each gender, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.

7. Connect a SHORT calibration standard to the selected test port (connector to which the DUT is to be connected).
8. Click **Short** to start the calibration measurement.
9. Connect a LOAD calibration standard to the selected test port (connector to which the DUT is to be connected).
10. Click **Load** to start the calibration measurement.
11. Click **Done** to terminate the 1-port calibration process. Upon pressing this key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

Connecting the standard for 1-port calibration



e5071c300

Full 2-Port Calibration

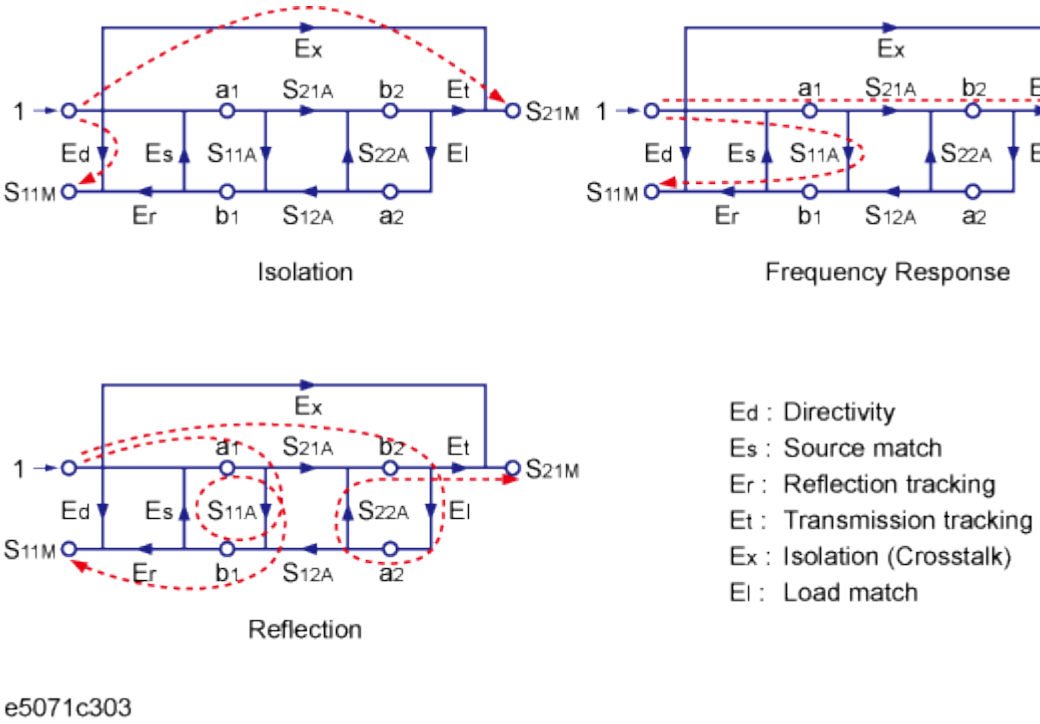
- Overview
- Procedure

Other topics about Basic Calibration

Overview

In full 2-port calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, or a LOAD standard to two desired test ports (or a THRU standard between two ports).This calibration effectively eliminates the directivity error, crosstalk, source match error, frequency response reflection tracking error, and frequency response transmission tracking error from the test setup in a transmission or reflection test using those ports. This calibration makes it possible to perform measurements with the highest possible accuracy. A total of twelve error terms, six each in the forward direction and the reverse direction, are used in a calibration.

1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.



e5071c303

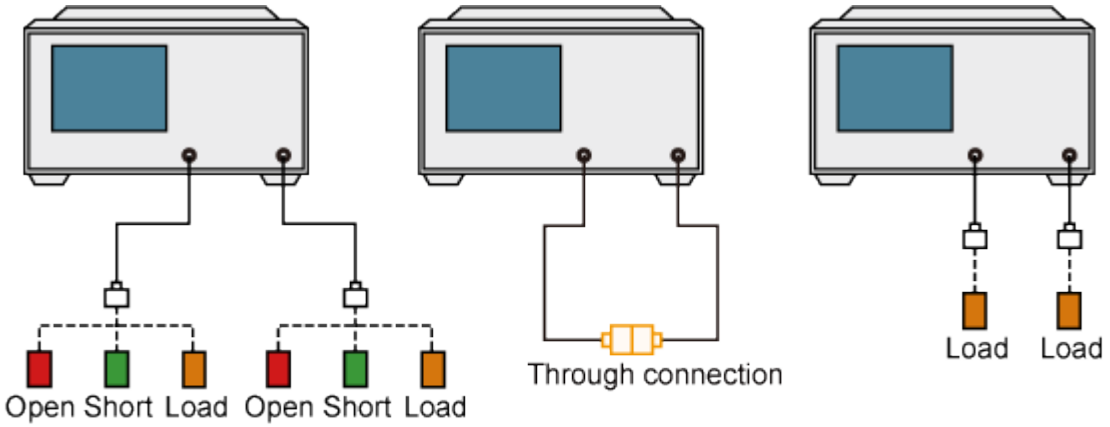
Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.

3. Click **Calibrate** > **2-Port Cal**.
4. Click **Select Ports**, then select the test ports on which you will perform full 2-port calibration. (In the procedure below, the selected test ports are denoted as x and y.)
5. Click **Reflection**.
6. Connect an OPEN calibration standard to test port x (the connector to which the DUT is to be connected).
7. Click **Port x Open** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
    - If you select the calibration kit which has different calibration definitions for each gender, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.
8. Disconnect the OPEN calibration standard and replace it with a SHORT calibration standard.
9. Click **Port x Short** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
10. Disconnect the SHORT calibration standard and replace it with a LOAD standard.
11. Click **Port x Load** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
12. Repeat the above procedure for port y.
13. Click **Return**.
14. Click **Transmission**.
15. Make a THRU connection between ports x and y (between the connectors to which the DUT is to be connected).
16. Click **Port x-y Thru** to start the calibration measurement (**x** and **y** denote the test ports between which the THRU connection is being made).
17. Click **Return**.
18. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.
19. Click **Isolation (Optional)**.
20. Connect a LOAD standard to each of the two test ports (connectors to which the DUT is to be connected).

21. Click **Port x-y Isol** to start the calibration measurement (**x** and **y** denote the port numbers to which the LOAD standard is connected).
22. Click **Return**.
23. Click **Done** to terminate the full 2-port calibration process. Upon pressing this key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

Connecting standards in full 2-port calibration



e5071c331

Full 3-Port Calibration

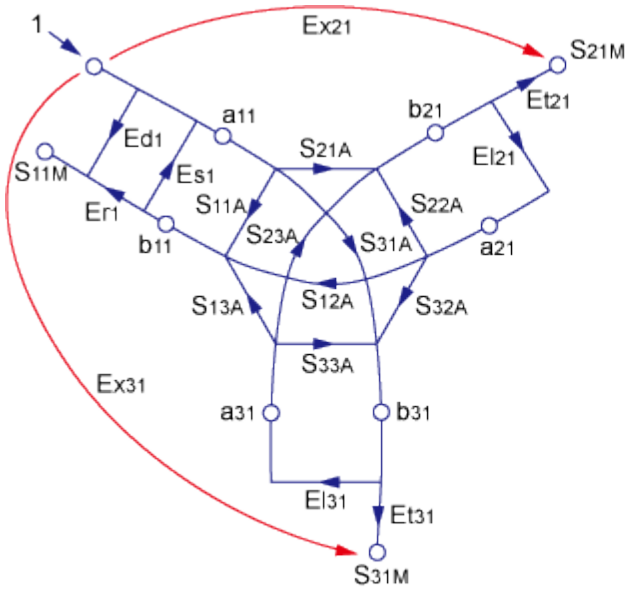
- Overview
- Procedure

Other topics about Basic Calibration

Overview

In full 3-port calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, or a LOAD standard to three desired test ports (or a THRU standard between three ports). This calibration effectively eliminates the directivity error, crosstalk, source match error, load match error, frequency response reflection tracking error, and frequency response transmission tracking error from the test setup in a transmission or reflection test using those ports. As in full 2-port calibration, this calibration method also makes it possible to perform measurements with the highest possible accuracy. There are unique error terms for directivity, source match, and reflection tracking for each stimulus test port (3 × 3 ports = 9). As for isolation, load match, and transmission tracking errors, there are unique terms for each combination between a stimulus port and a response port (3 × 6 combinations = 18). Therefore, in total, 27 error terms are involved in a full 3-port calibration.

1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.



Directivity :	Ed1, Ed2, Ed3
Isolation :	Ex21, Ex31, Ex12, Ex32, Ex13, Ex23,
Source match :	Es1, Es2, Es3
Load match :	El21, El31, El12, El32, El13, El23,
Reflection tracking :	Er1, Er2, Er3
Transmission tracking :	Et21, Et31, Et12, Et32, Et13, Et23,

e5071c304

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **3-Port Cal**.
4. Click **Select Ports**, then select the test ports on which you will perform full 3-port calibration.

Reflection

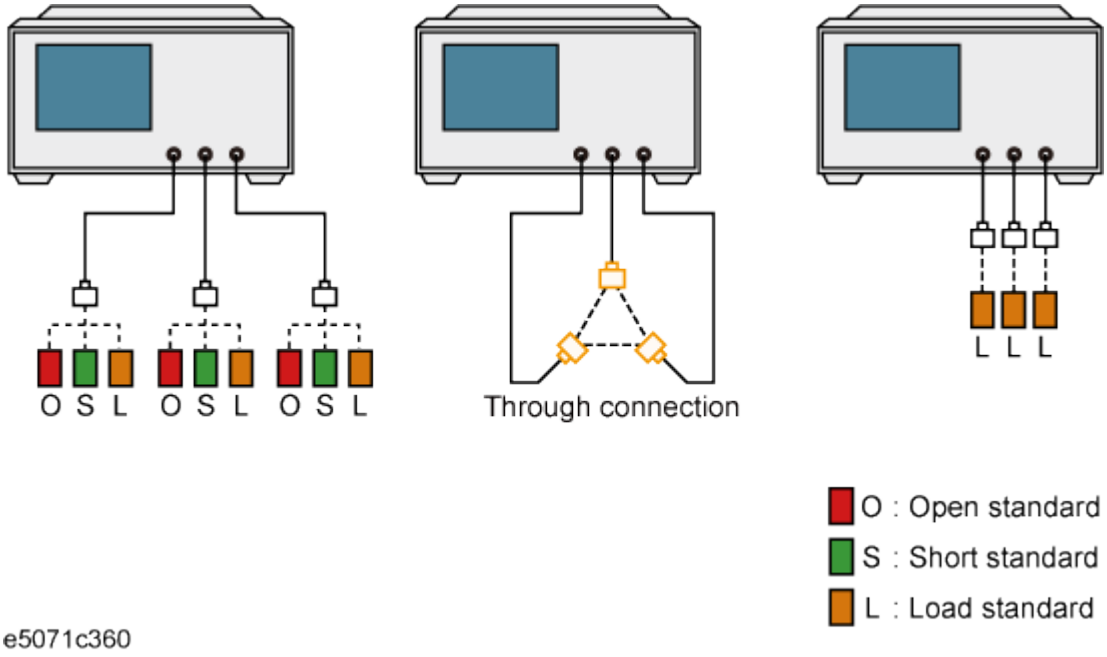
1. Click **Reflection**.
2. Connect an OPEN calibration standard to test port x (the connector to which the DUT is to be connected).
3. Click **Port x Open** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
    - If you select the calibration kit which has different calibration definitions for each gender, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.
4. Disconnect the OPEN calibration standard and replace it with a SHORT calibration standard.
5. Click **Port x Short** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
6. Disconnect the SHORT calibration standard and replace it with a LOAD standard.
7. Click **Port x Load** to start the calibration measurement (**x** denotes the test port to which the standard is connected).
8. Repeat the above procedure for ports y and z.
9. Click **Return**.

Transmission

1. Click **Transmission**.
2. Make a THRU connection between ports x and y (between the connectors to which the DUT is to be connected).
3. Click **Port x-y Thru** to start the calibration measurement (**x** and **y** denote the test ports between which the THRU connection is being made).
4. Repeat above procedure for ports x and z. then for ports y and z.

5. Click **Return**.
6. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.
7. Click **Isolation (Optional)**.
8. Connect a LOAD standard to each of the three test ports (connectors to which the DUT is to be connected).
9. Click **Port x-y Isol** to start the calibration measurement (**x** and **y** denote the port numbers to which the LOAD standard is connected).
10. Click **Return**.
11. Click **Done** to terminate the full 3-port calibration process. Upon pressing this key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

Connecting standards in full 3-port calibration



e5071c360

Full 4-Port Calibration

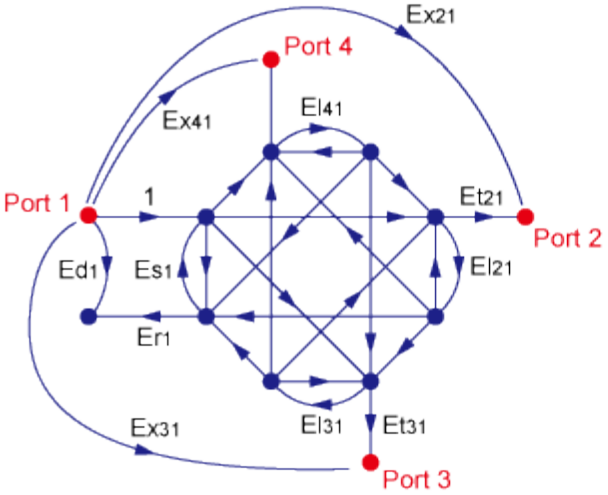
- Overview
- Procedure

Other topics about Basic Calibration

Overview

In full 4-port calibration, calibration data are measured by connecting an OPEN standard, a SHORT standard, or a LOAD standard to the four test ports (or a THRU standard between the four ports). This calibration effectively eliminates the directivity error, crosstalk, source match error, load match error, frequency response reflection tracking error, and frequency response transmission tracking error from the test setup in a transmission or reflection test using those ports. As in full 2-port calibration, this calibration method also makes it possible to perform measurements with the highest possible accuracy. There are unique error terms for directivity, source match, and reflection tracking for each stimulus test port (3 × 4 ports = 12). As for isolation, load match, and transmission tracking errors, there are unique terms for each combination between a stimulus port and a response port (3 × 12 combinations = 36). Therefore, in total, 48 error terms are involved in a full 4-port calibration.

1. System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.



Directivity :	Ed1, Ed2, Ed3, Ed4
Isolation :	Ex21, Ex31, Ex41, Ex12, Ex32, Ex42, Ex13, Ex23, Ex43, Ex14, Ex24, Ex34,
Source match :	Es1, Es2, Es3, Es4
Load match :	El21, El31, El41, El12, El32, El42, El13, El23, El43, El14, El24, El34,
Reflection tracking :	Er1, Er2, Er3, Er4
Transmission tracking :	Et21, Et31, Et41, Et12, Et32, Et42, Et13, Et23, Et43, Et14, Et24, Et34,

e5071c306

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **4-Port Cal**.

Reflection

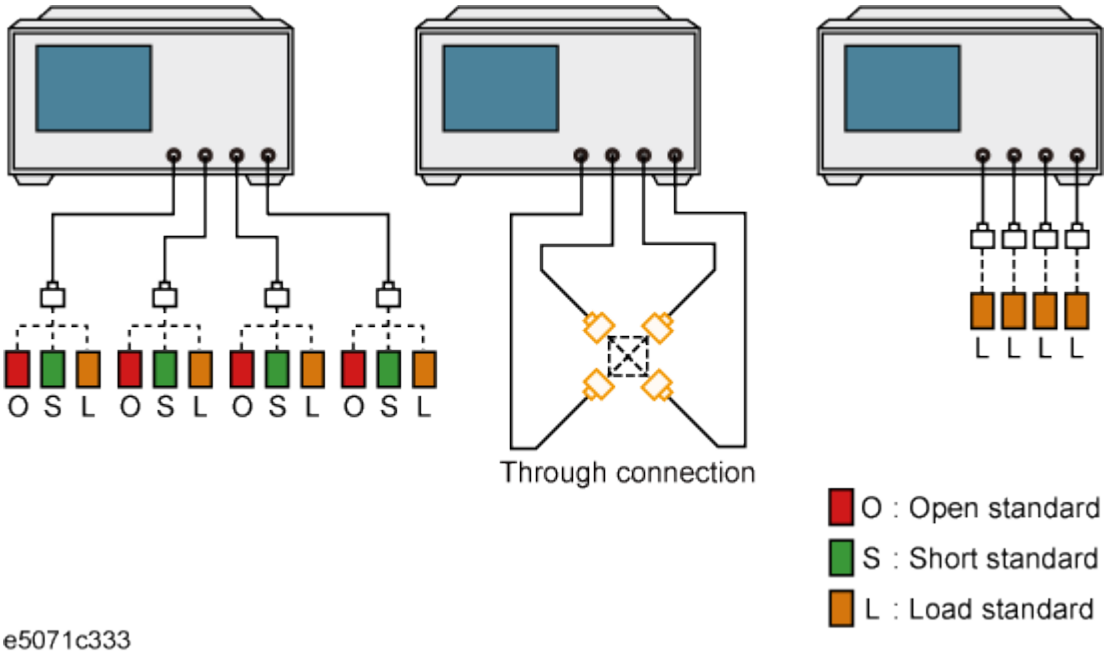
1. Click **Reflection**.
2. Connect an OPEN calibration standard to test port 1.
3. Click **Port 1 Open** to start the calibration measurement.
    - If you select the calibration kit which has different calibration definitions for each gender, (m) and (f) in the name (label) of the standard displayed in the softkey indicate male (m) and female (f) for the analyzer's connector, respectively.
4. Disconnect the OPEN calibration standard and replace it with a SHORT calibration standard.
5. Click **Port 1 Short** to start the calibration measurement.
6. Disconnect the SHORT calibration standard and replace it with a LOAD standard.
7. Click **Port 1 Load** to start the calibration measurement.
8. Repeat the above procedure for ports 2, 3 and 4.
9. Click **Return**.

Transmission

1. Click **Transmission**.
2. Make a THRU connection between ports 1 and 2.
3. Click **Port 1-2 Thru** to start the calibration measurement.
4. Repeat above procedure ports 1 and 3.
5. Repeat the procedure on ports 1 and 4.
6. Repeat the procedure on ports 2 and 3.
7. Repeat the procedure on ports 2 and 4.
8. Repeat the procedure on ports 3 and 4.
9. Click **Return**.
10. If an isolation calibration must be performed using a LOAD standard, follow the procedure below.
11. Click **Isolation (Optional)**.

12. Connect a LOAD standard to each of the four test ports (connectors to which the DUT is to be connected).
13. Click **Port 1-2 Isol** to start the calibration measurement.
14. Perform the each calibrations by clicking **Port 1-3 Isol**, **Port 1-4 Isol**, **Port 2-3 Isol** and **Port 2-4 Isol**.
15. Click **Return**.
16. Click **Done** to terminate the full 4-port calibration process. Upon pressing this key, calibration coefficients will be calculated and saved. The error correction function will also be automatically enabled.

Connecting standards in full 4-port calibration



Calibration with ECal (Electronic Calibration)

ECal (electronic calibration)

ECal is a calibration method that uses solid-state circuit technology. ECal offers the following advantages:

- Simplified calibration process.
- Shorter time required for calibration.
- Reduced chance of erroneous operation.
- Little degradation of performance due to wear because the ECal module employs PIN diodes and FET switches.
    - If the frequency sweep range exceeds the frequency range of the ECal, the calibration data for the minimum frequency or maximum frequency are used for the exceeding frequency range and extrapolation is being executed.
    - When using Ecal with the E5071C options 2D5, 4D5, 2K5 and 4K5 , wait time is inserted automatically for stable calibration results. Approximate calibration time is described in the support ECal page.

Refer the following section for ECal calibration.

- Connecting ECal to E5071C
- 1-Port Calibration Using a 2-Port ECal Module
- Calibration Using 4-port ECal
- Full 2-Port Calibration Using the 2-Port ECal Module
- Full 3-Port and Full 4-Port Calibration using 2-Port ECal
- Improving Calibration Accuracy along with ECal
- Confidence Check on Calibration Coefficients Using ECal
- Turning off ECal auto-detect function
- User-characterized ECal

Other topics about Calibration with ECal

Connecting ECal to E5071C

- ECal Driver Installation
- Connecting Single ECal
- Connecting Multiple ECal

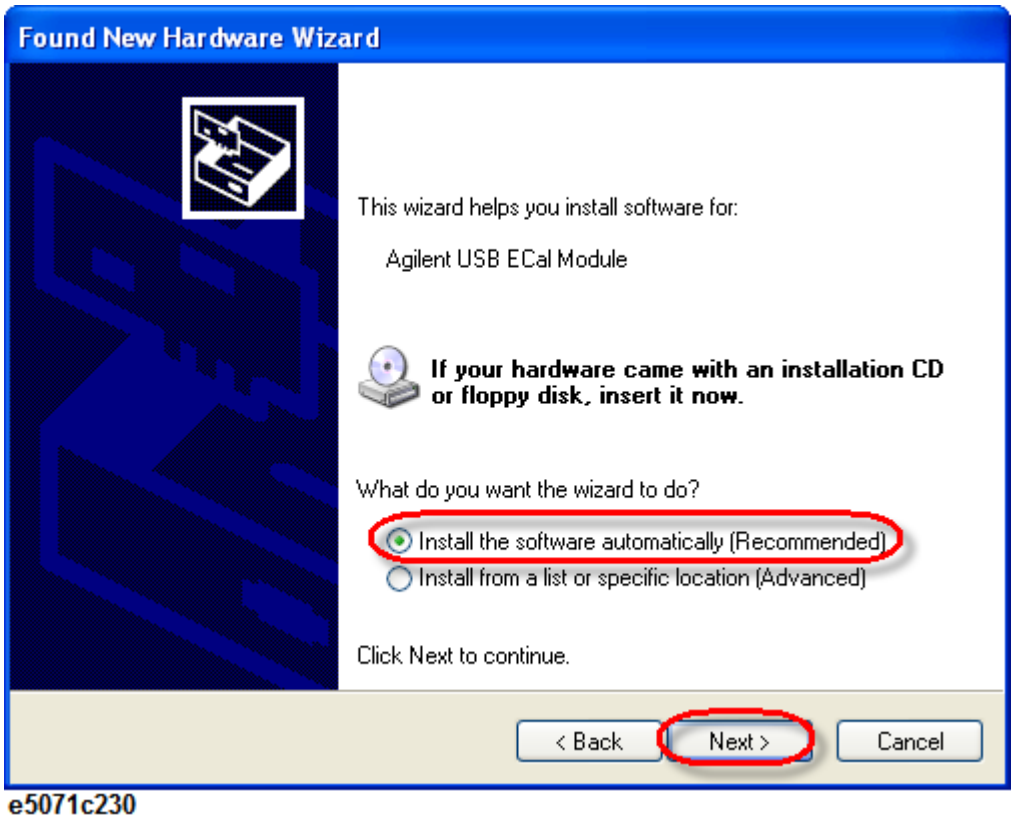
Other topics about Calibration with ECal

ECal Driver Installation

When the ECal is connected with USB ports at the first time, ECal driver installation is required.

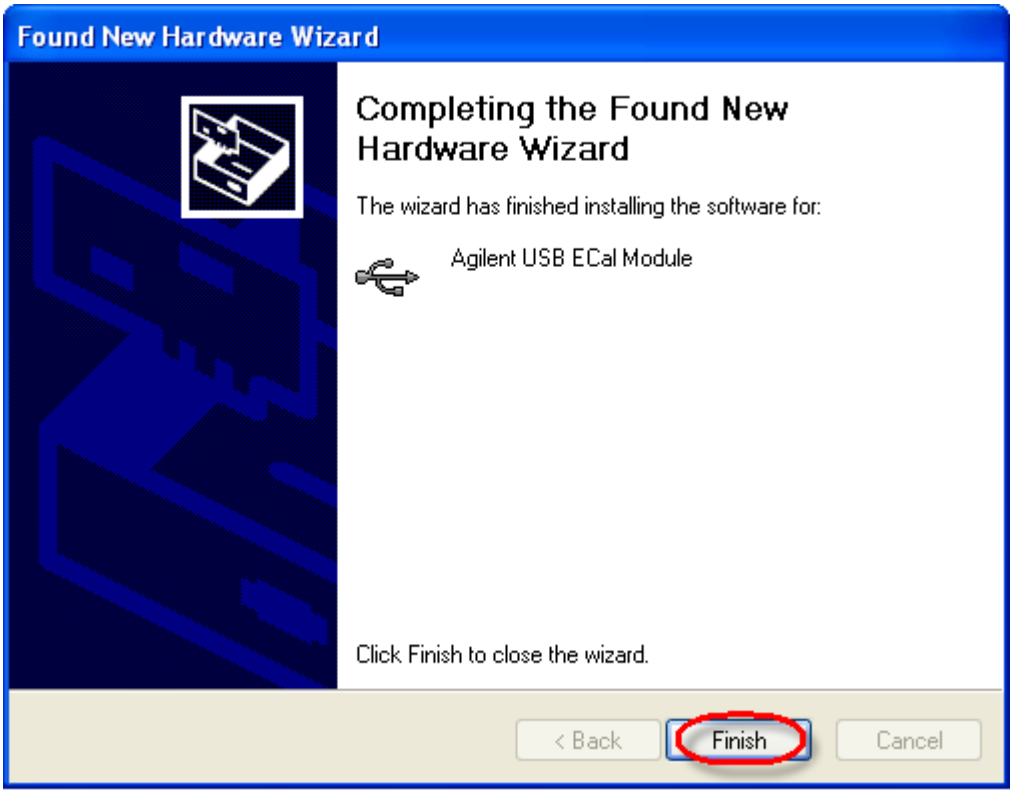
1. Connect a Ecal to the USB port of the E5071C.
2. Select **No, not this time**, then click **Next**.

3. Select **Install the software automatically (Recommended)**, then click **Next**.



e5071c230

4. Click **Finish.**



e5071c231

5. Even if you install the driver on a USB port, you will be asked to install the driver again if you connect the ECal with a different USB port.

Connecting Single ECal

There are several options available in calibrating using ECal module:

- 1-Port Calibration Using a 2-Port ECal Module
- Calibration Using 4-Port ECal
- Full 2-Port Calibration Using the 2-Port ECal Module
- Full 3-Port and Full 4-Port Calibration using 2-Port ECal

Connecting Multiple ECal

The ECal module connected to E5071C is listed here: **Cal** > **ECal** > **ECal**.

If a single ECal is connected, only one ECal is listed here. When multiple ECal are connected E5071C, all the connected ECals are listed here. The first line indicates the model number of the ECal and the second line indicates its serial number, as shown below:

E5071C revision A.9.60 and above allows multiple ECal to be connected at any one time. To perform calibration using your preferred ECal module, select the desired ECal from the list and perform 1-Port, 2-Port, 3-Port or 4-Port calibration just as its performed in single ECal. Once completed, you can select another ECal connected to E5071C and repeat the procedure.

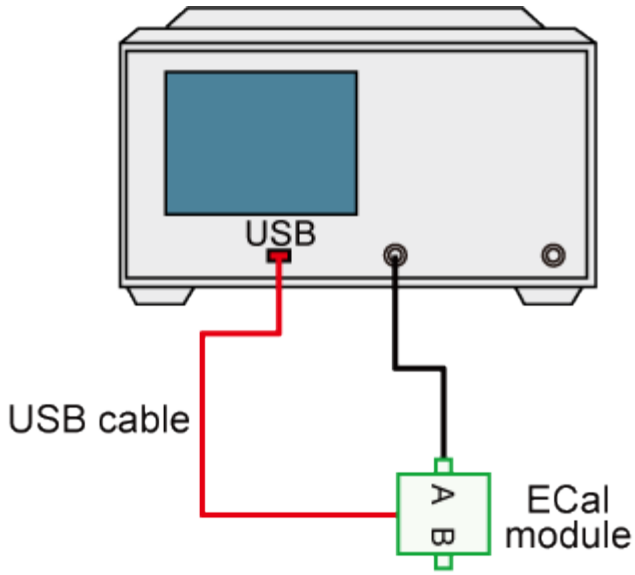
Other topics about Calibration with ECal

1-Port Calibration Using a 2-Port ECal Module

Follow the procedure below to perform a 1-port calibration using the 2-port ECal module.

1. Connect the USB port on the ECal module with the USB port on the E5071C via a USB cable. This connection may be made while the E5071C's power is on.
2. Allows the ECal module to warm up for 15 minutes until the module indicates from WAIT to READY.
3. Connect a port on the ECal module to the test port to be calibrated.
4. If you don't use all of the ECal module's ports, connect terminations to the unused ports.
5. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
6. Press **Cal** key.
7. Click **Ecal** > **1-Port Cal**.
8. Perform a 1-port calibration by clicking the softkey of selected port.
    1. You can connect the ports of the ECal and the test ports of the E5071C arbitrarily. Connected ports can be manually specified although they are automatically detected before data measurement. For more information, Turning off ECal auto-detect function.

Connecting ECal module (1-port calibration)



e5071c325

Other topics about Calibration with ECal

Calibration Using 4-port ECal

- Overview
- Procedure

Other topics about Calibration with ECal

Overview

The E5071C allows you to perform calibration using the 4-port ECal module. It provides much simpler operation than when using the 2-port ECal. Especially when using a multi-port test set, calibration time and operator errors can be reduced significantly.

Procedure

1. Connect the USB cable between the USB port of the 4-port ECal module and the USB port of the E5071C. You can make this connection while the E5071C's power is on.
2. Allows the ECal module to warm up for 20 minutes until the module indicates from WAIT to READY.
3. Connect the ports of the 4-port ECal module to the test ports you want to calibrate.
4. If you don't use all of the ECal module's ports, connect terminations to the unused ports.
5. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
6. Press **Cal** key.
7. Click **Ecal**.
8. Select the **calibration type**.

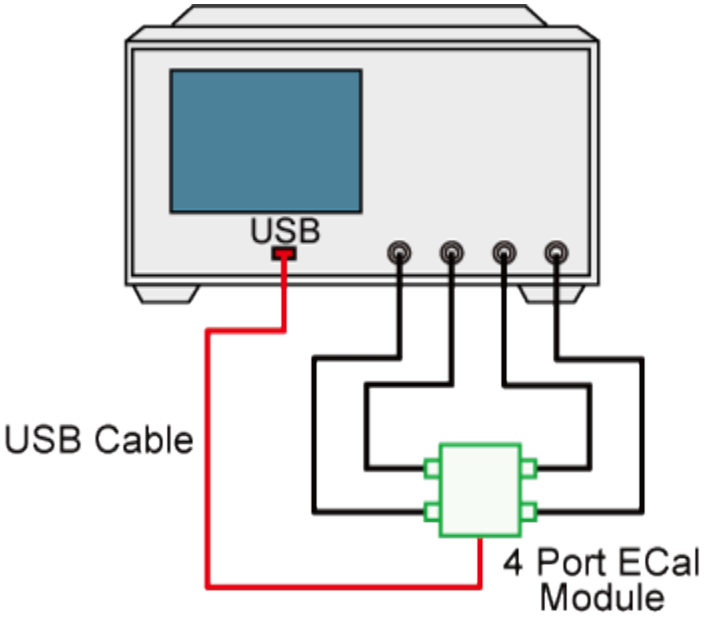
Softkey	Function
1-Port ECal	Selects 1-port calibration
2-Port ECal	Selects full 2-port calibration
3-Port ECal	Selects full 3-port calibration
4-Port ECal	Selects full 4-port calibration
Thru ECal	Selects THRU calibration

9.

10. If you must select a port, the softkey for making this selection is displayed. Select a port and start calibration. If you do not have to select a port, this step is skipped.

11. The E5071C detects the test ports connected to the ECal and then measurement starts. If the selected test ports to be calibrated is not connected to the ECal module, an error occurs.

Connecting 4-port ECal module (for full 4-port calibration)



e5071c326

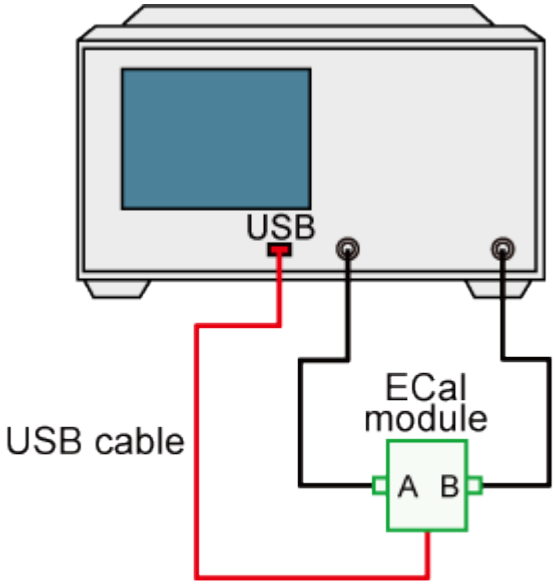
- You can connect the ports of the ECal and the test ports of the E5071C arbitrarily. Connected ports can be manually specified although they are automatically detected before data measurement. For more information, Turning off ECal auto-detect function.

Full 2-Port Calibration Using the 2-Port ECal Module

Follow the procedure below to perform a full 2-port calibration using the 2-port ECal module.

1. Connect the USB port on the ECal module with the USB port on the E5071C via a USB cable. This connection may be done while the E5071C's power is on.
2. Allows the ECal module to warm up for 15 minutes until the module indicates from WAIT to READY.
3. Connect port A and port B on the ECal module to the test ports to be calibrated.
4. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
5. Press **Cal** key.
6. Click **ECal**.
7. Click **2 Port ECal**. When using a 2-port E5071C, pressing this key performs a 2-port ECal. When using a 3-port or 4-port E5071C, click one of the softkeys to start a full 2-port calibration.

Connecting ECal module (full 2-port calibration)



e5071c322

- You can connect the ports of the ECal and the test ports of the E5071C arbitrarily. Connected ports can be manually specified although they are automatically detected before data measurement. For more information, Turning off ECal auto-detect function

Other topics about Calibration with ECal

Full 3-Port and Full 4-Port Calibration using 2-Port ECal

- Overview
- Procedure

Other topics about Calibration with ECal

Overview

A VBA macro (ECal Assistant) is pre-installed in the E5071C to carry out a full 3-port or a full 4-port calibration using the 2-port ECal.

- ECal Assistant does not perform isolation calibration.

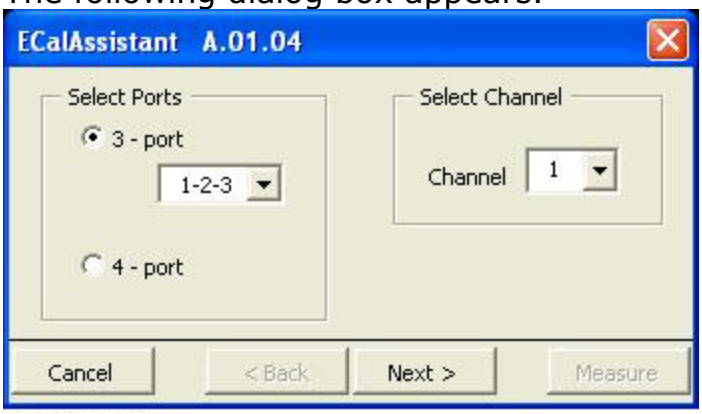
Procedure

1. Connect the USB port of the ECal module to the USB port on the E5071C with a USB cable. The connection may be made while the E5071C's power is on.
2. Allows the ECal module to warm up for 15 minutes until the module indicates from WAIT to READY.
3. Press **Macro Setup** key.
4. Click **Load Project**.
5. From the Open dialog box, select the VBA project file **D:\Agilent\ECalAssistant.vba** and click the **Open** button.
6. Press **Macro Run** key.
7. The following dialog box appears.

e5071c085

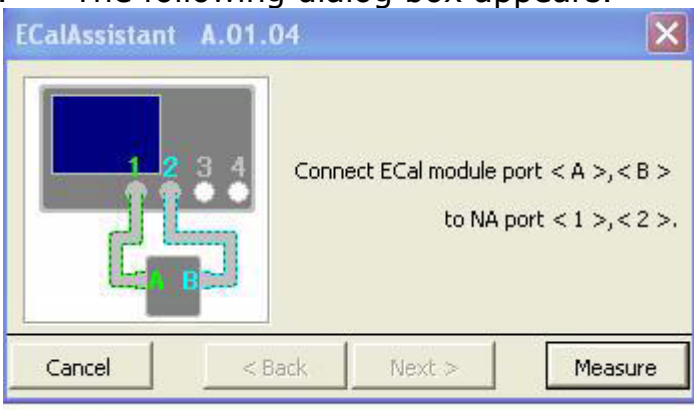
8. Click **Next**.

9. The following dialog box appears.



e5071c086

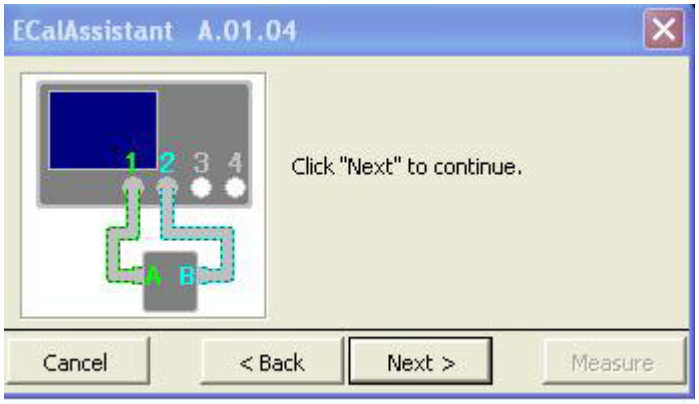
10. In the Select Ports area, click and select the **3-Port** (for a full 3-port calibration) or the **4-Port** (for a full 4-port calibration) radio button.
11. When a full 3-port calibration is carried out on an E5071C, select the test ports to be calibrated from the drop-down list box below the **3-Port** button (either **1-2-3**, **1-2-4**, **1-3-4**, or **2-3-4**).
12. In the Select Channel area, select the channel to be calibrated (one of channels **1** to **9**).
13. Click the **Next** button.
14. The following dialog box appears.



e5071c087

15. Following the connection diagram shown in the dialog box, connect ports A and B of the ECal module to test ports on the E5071C. The connection diagram shown in each dialog box that appears in each step depends on the number of test ports on the E5071C.
16. Click the **Measure** button to start the measurement of calibration data.

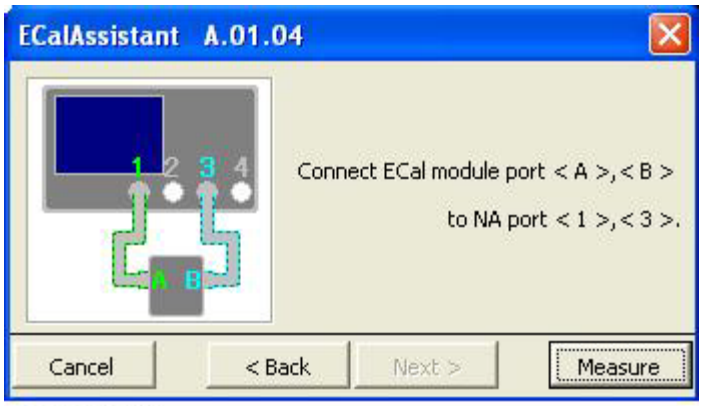
17. Upon completion of measurement, the following dialog box appears.



e5071c088

18. Click the **Next** button.

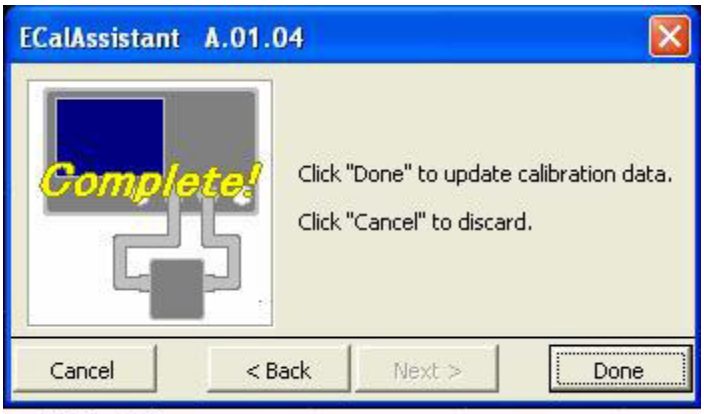
19. The following dialog box appears.



e5071c089

20. Re-connect the ECal module following the instructions given in each dialog box and continue the calibration process.

21. When all calibration data have been collected, a dialog box with the **Complete!** sign appears as shown in the following figure, click **Done** to finish the calibration. If you wish to cancel the calibration, click **Cancel**.


ECalAssistant A.01.04
Complete!
Click "Done" to update calibration data.
Click "Cancel" to discard.
Cancel
< Back
Next >
Done

e5071c099

Improving Calibration Accuracy along with ECal

Inaccuracy caused by Thru calibration in the ECal can be reduced by the using the following methods.

- Partial Overwrite
- Unknown Thru Calibration

Other topics about Calibration with ECal

Partial Overwrite

Using the partial overwrite function allows you to improve the calibration accuracy. For example, follow these steps for full 2-port calibration.

1. Execute full 2-port calibration with ECal and save the calibration coefficients.
2. Execute Partial overwrite procedure with the Thru standard of the calibration kit

Unknown Thru Calibration

E5071C allows you to perform Thru calibration of ECal as the unknown Thru calibration. In this function, the Thru calibration is performed with the Thru standard in ECal, however, the stored Thru calibration data in ECal is not used. E5071C performs the Thru calibration as an unknown Thru calibration.

1. Press **Cal** Key, then click **ECal**.
2. Click **Unknown Thru** to turn on.
3. Perform your desired ECal calibration.

Confidence Check on Calibration Coefficients Using ECal

- Overview
- Procedure

Other topics about Calibration with ECal

Overview

Using the ECal module, the E5071C lets you verify the obtained calibration coefficients to determine whether correct measurement is possible with them.

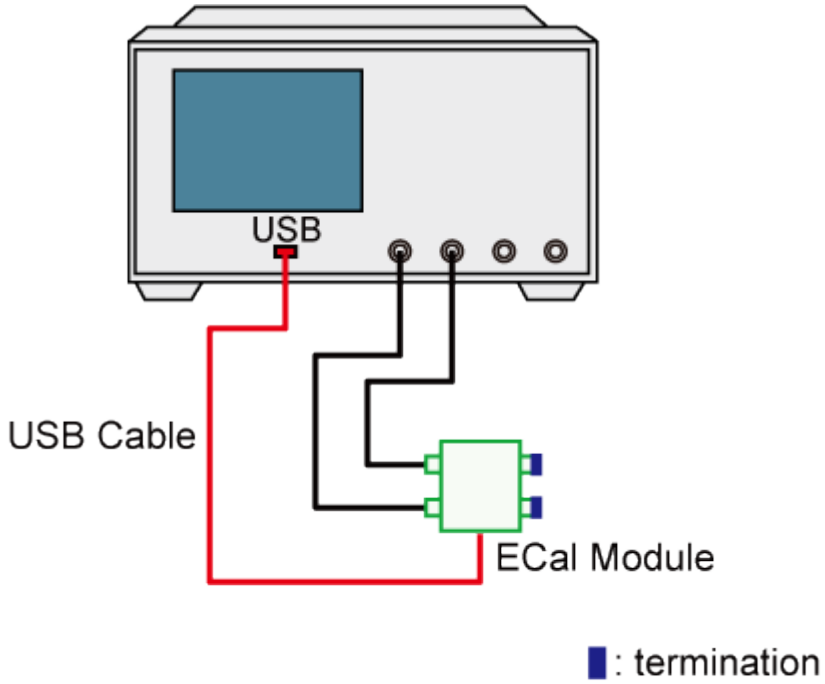
The E5071C can set ECal to the state used to verify the measurement parameters and then copy the appropriate characteristics of that verification state to the memory trace from the ECal's built-in memory. This is done according to the measurement parameters of the active trace of the active channel. While measuring ECal in this specified state, you can compare the measurement results with those of the E5071C and with the appropriate measurement results stored in ECal in several different ways. These include simultaneously displaying the data and memory traces or displaying the math operation results between the data and memory traces. This enables you to verify the correctness of measurement for each measurement parameter when the obtained calibration coefficients are used.

Procedure

1. Connect the USB cable between the USB port of the ECal module and that of the E5071C. You can make this connection while the E5071C's power is ON.
2. Allows the ECal module to warm up for 15 to 20 minutes until the module indicates from WAIT to READY.
3. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the verification.
4. Press **Meas** key.
5. Select the S-parameter you want to verify. You cannot verify the mixed mode S-parameter.
6. Connect the test ports of the E5071C corresponding to the selected S-parameter (for example, ports 1 and 2 when the S-parameter is S21) and the ports of the ECal module.
7. If you don't use all of the ECal module's ports, connect terminations to the unused ports.
8. Press **Cal** key.
9. Click **ECal**.

10. When using an adapter to the ECal, click **Characterization** and then press the softkey corresponding to the characterization for the adapter you are using.
11. Click **Confidence Check**.
12. Compare the data trace and the memory trace and verify whether measurement is correct.
13. The following is the procedure for comparison when simultaneously displaying the data trace and the memory trace.
    a. Press **Display** key.
    b. Click **Display** > **Data & Mem**.
    c. Press **Scale** key.
    d. Click **Auto Scale**.
    e. Determine whether the difference between the traces is acceptable. The difference should be read in terms of linear values instead of dB error. If we consider the magnitude of the linear error as compared to the dB delta, the value is very small. So to evaluate the difference between the traces, a linear error scale should be used for comparison instead of the dB error scale.
14. For all of the parameters you want to verify, repeat the procedure.

Connecting ECal module (for verification of S21)



e5071c328

Turning off ECal auto-detect function

The ECal module automatically detects the connection between E5071C's test ports and ECal module's ports. You can turn off this function to set ports manually.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to turn off the auto-detect function.
2. Press **Cal** key.
3. Click **ECal** > **Orientation**, and select **Manual**.
4. Specify a test port of the E5071C.
5. Specify an Ecal port for the port of the E5071C you specified.
    - Even if the connection is wrong with the auto-detect function turned off, no error is displayed.

Other topics about Calibration with ECal

User-characterized ECal

- Overview
- Precautions to take in using VBA macros
- Storing user characteristics to the ECal module
- Backup/recovery of ECal module's built in flash memory
- Executing User characterized ECal

Other topics about Calibration with ECal

Overview

The E5071C allows you to execute ECal calibration with user-defined characteristics instead of the ECal characteristics defined as the factory default. This feature is called User-characterized ECal, and it is used to execute ECal calibration when an adapter is connected to the ECal module.

Before executing the User-characterized ECal, you have to measure data, such as characteristics when the adapter is connected to the ECal module, and store them to the built-in flash memory of the ECal module as the user characteristics.

Use the following VBA macro to acquire user characteristics and store them to the ECal module's built-in memory.

Storage folder	VBA macro name (project name)
D:\Agilent	EcalCharacterization.vba

Precautions to take in using VBA macros

- Never connect/disconnect the USB cable while executing the VBA macro.
  - In particular, the above precaution must always be observed while the VBA macro is storing data to the ECal module's built-in flash memory; disconnecting the USB cable at this time may damage the ECal module.
- Back up the flash memory contents.
  The VBA macro provides a feature to back up the contents of the ECal module's built-in flash memory. Before storing user characteristics to the ECal module, be sure to use this feature to back up the flash memory's current contents.

Storing user characteristics to the ECal module

Follow these steps to measure characteristics while an adapter is connected to the ECal module and then to store them to the ECal module's built-in flash memory as user characteristics.

- With the 2-port E5071C, you cannot measure the user characteristics of a 4-port ECal module and store them into the memory by using this VBA macro.

1. Connecting ECal Module

Connect the USB cable between the USB port of the ECal module and that of the E5071C. You can make this connection while the E5071C's power is ON.

2. Setting Stimulus Condition

Set the stimulus condition of the channel for which you want to measure the user characteristics. For optimal accuracy, set the IF bandwidth to 1 kHz or less.

3. Executing Calibration

For the channel for which you have set the stimulus condition, execute full 2/4-port calibration with a mechanical calibration kit when characterizing 2/4-port ECal. Define the calibration surface as the connector surface connected to each port of the ECal module in the state used to measure characteristics.

4. Starting the VBA MACRO
   a. Press **Macro Setup** key.
   b. Click **Load Project**.
   c. The Open dialog box appears. Specify the file name "**D:\Agilent\EcalCharacterization.vba**" and click **Open**.
   d. Press **Macro Run** key to start the macro. When more than 1601 measurement points is set for 1 channel and 4 traces, the E5071C VBA macro function may take more time to operate. A warning is also displayed if the user tries to increase the number of points to

more than 1601 with 1 channel and 4 traces. Click **OK**.

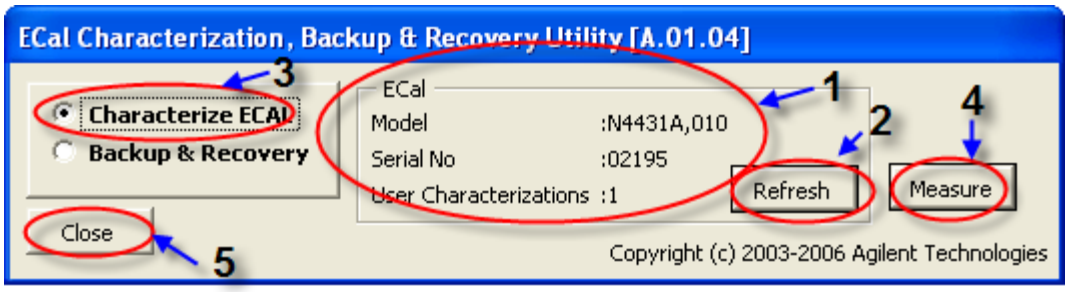
e5071c136

e. The ECal part (1 in the figure below) displays the information of the ECal module connected to the E5071C.

f. Click **Refresh** (2 in the figure below) to update the information if you have connected another ECal module after the macro has been started.

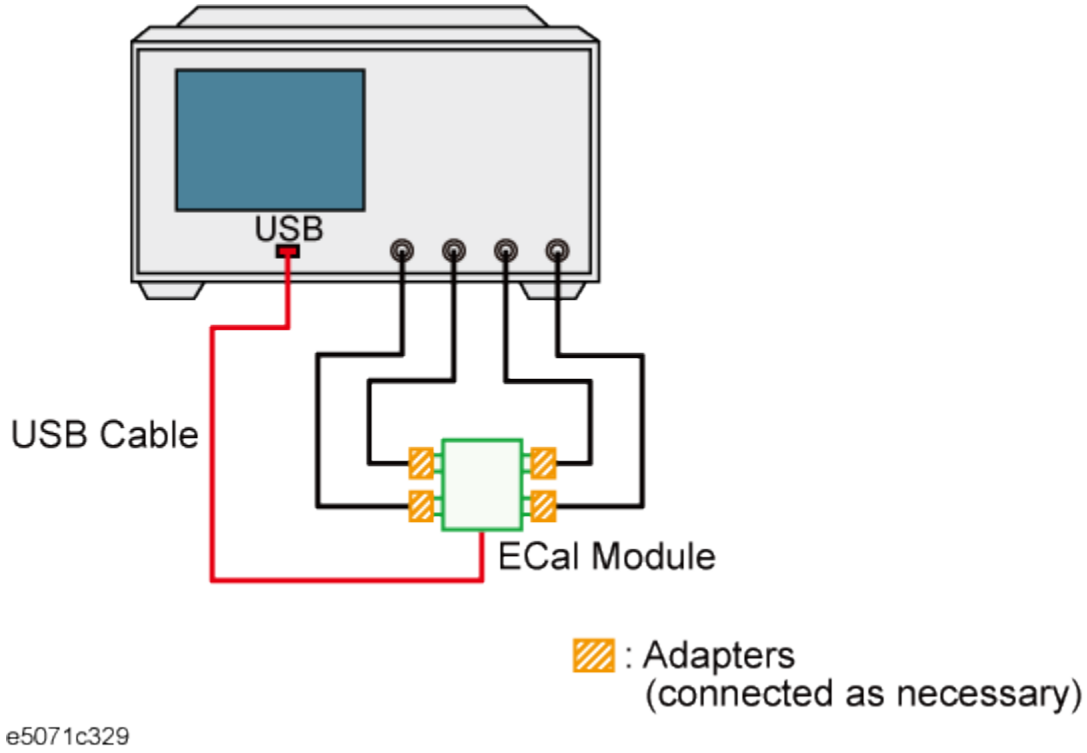
5. Measuring User Characteristics

a. Select Characterize ECAL (3 in the figure below) to display the User Characteristic Measurement screen.



e5071c232

b. After connecting the adapter to the ECal module as necessary, connect each port of the ECal module and the test port of the E5071C.



c. Click **Measure** (4 in the figure) to start measurement.

   a. You can select any port of the ECal module and any test port of the E5071C for connection; the E5071C automatically recognizes the connected ports before measurement.

6. Storing the User Characteristics to the Memory

a. When the measurement is complete, the User Characterization Info screen appears.

b. Enter the following information.

Designation	Category	Description
1	Number	Specify a user number (a location number in the memory where you want to store the user characteristics) . If the specified location number is not used for storage, the

		parts Characterization, Connectors, and Adapter Description are left blank; if already used, the stored contents are displayed.
2	Characterization	Enter the information (operator, used analyzer, and so on) when measuring user characteristics as necessary.
3	Connectors	Select the connector types of the adapters for the ECal module's test ports. Male and female in the list of connected types indicate male and female adapter, respectively. Select "No adapter" if no adapter is used on a port.
4	Adapter Description	Enter the detailed information on the adapters connected to each port as necessary.

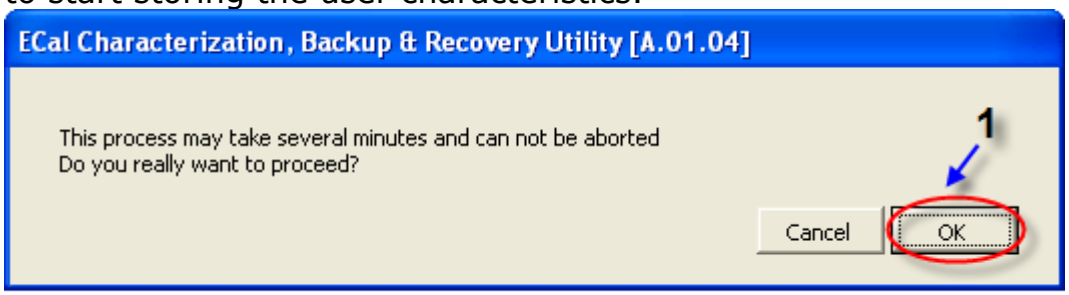
The information you have entered is displayed when checking the user characteristics information by using the key strokes: **Cal** > **Ecal** > **Characterization Info**.

d. Click **Write**.

e. At this time, if user characteristics are already stored for the specified user number, a dialog appears to confirm overwriting. Click **OK**.

a. Although the maximum number of user characteristics stored to the ECal's memory is usually five, this number may be limited by memory size because the size of user-characteristics data is not fixed and increases in proportion to the number of measurement points. An error occurs when the **Write** button is pressed if the total size added the new user characteristics exceeds this limitation due to memory size.

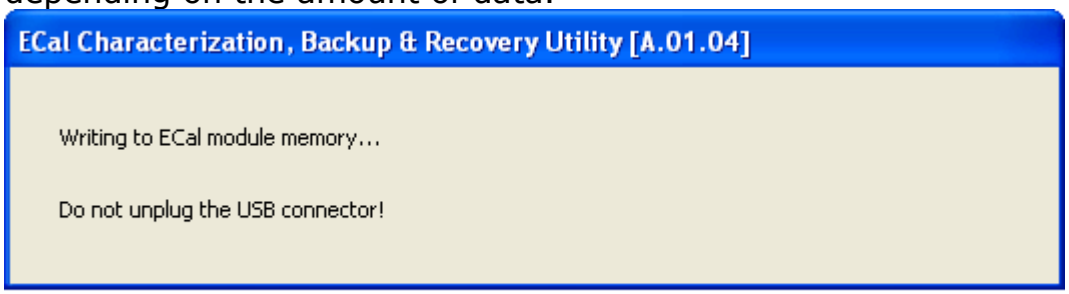
f. The following dialog box is displayed to confirm execution. Click **OK** to start storing the user characteristics.



e5071c234

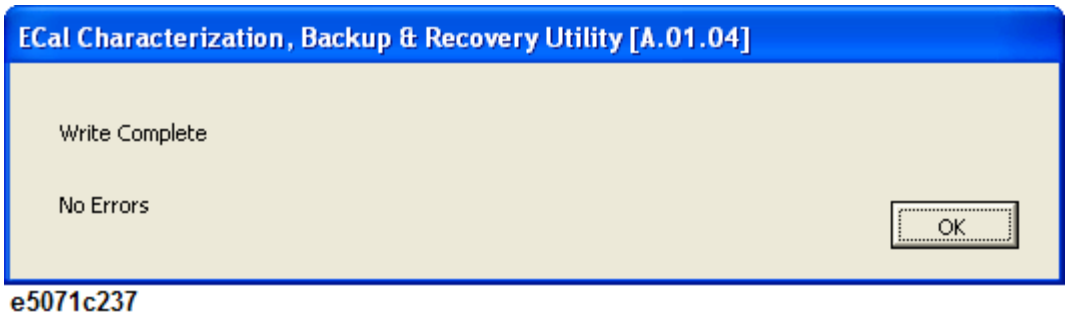
a. Do not disconnect the USB cable or terminate the VBA macro by force while the VBA macro is storing data to the ECal's built-in flash memory. Doing so may damage the ECal module.

g. The following dialog box appears while the VBA macro is storing data to memory. Storing the user characteristics takes a few minutes depending on the amount of data.



e5071c236

h. Another dialog box is displayed to notify completion of data storage. Click **OK**.



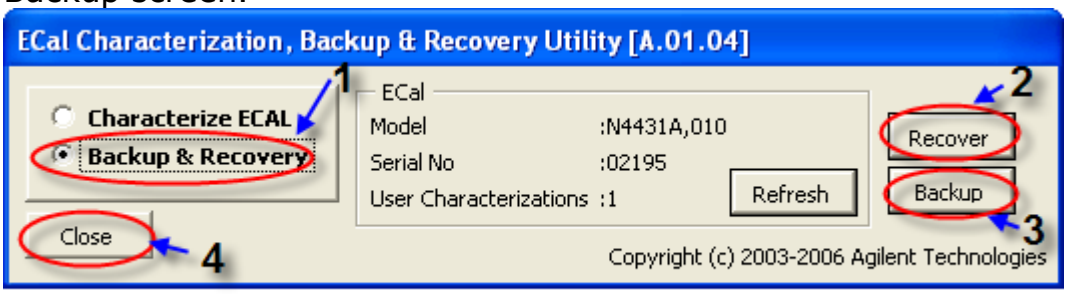
e5071c237

8. Closing the VBA macro

a. Click **Close** (No. 5 in Figure) to close the macro.

Follow these steps to back up the contents of the ECal module's built-in flash memory.

1. Connect the USB cable between the USB port of the ECal module and that of the E5071C. You can make this connection while the E5071C's power is ON.
2. Start the VBA macro according to Starting the VBA MACRO
3. Select **Backup Flash ROM** (No. 1 in the following figure) to display the Backup screen.



e5071c235

Recovery

1. Click **Recover** (No. 2 in the figure above.).
2. The Open dialog box appears. Enter the file name of the contents you want to recover and Click **Open**. If the serial number information stored in the file does not match that of the ECal module connected to the E5071C, a confirmation dialog box appears. Click **OK** to continue the recovery only if a mismatch between these serial numbers is allowed.
3. The dialog box is displayed to confirm execution. Click **OK** to start the recovery of the flash memory. The dialog box appears while the VBA macro is storing data to the memory. The recovery of the flash memory takes a few minutes depending on the amount of data.
4. The Completion screen appears. Click **OK**.

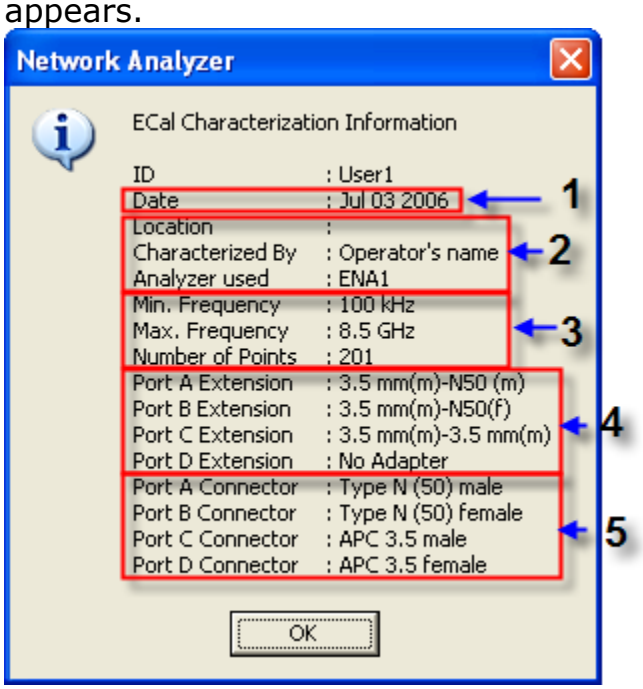
Backup

1. Click **backup**.
2. The Save As dialog box appears. Enter the name of the file you want to save and press **Save**.
3. Click **Close** to close the macro.

Executing User-characterized ECal

The execution procedure for the User-characterized ECal is the same as for normal ECal except that it requires the user characteristics to be selected in advance.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to execute calibration.
2. Press **Cal** key.
3. Click **ECal** > **Characterization**.
4. Select a user characteristic which is specified by User Characterization Info screen.
5. To check the information on the user characteristics you have selected, click **Characterization Info**. The following dialog box appears.



1:	The date when the user characteristics were measured
2:	The information you entered in Characterization (1 of User Characterization Info screen)
3:	The stimulus conditions when the user characteristics were measured
4:	The information you entered in Adapter Description (4 of User Characterization Info screen)

5:	The information you entered in Connectors (3 of User Characterization Info screen)

Advanced Calibrations

Setting the trigger source for calibration

- Overview of Trigger Source
- Setting Trigger Source

Other topics about Advanced Calibration

Overview of Trigger Source

You can set the trigger source for calibration before executing calibration.

You can select it from "Internal" or "System." Setting it to "System" allows you to use the same trigger source setting for calibration and measurement.

You can set the trigger source for calibration (which is usually set to "Internal") to "System," which allows you to generate triggers at any timing from a PC for external control or from the front panel for calibration in the same way as for measurement.

The setting of the point trigger and averaging trigger is also applied to the trigger for calibration. When the trigger source for calibration is set to "System" and the trigger source for measurement is set to "External" or "Manual" with the point trigger function set to ON, a trigger is required for each measurement point during calibration. When the averaging trigger function is set to ON, the sweep is performed the number of times specified by the averaging factor for a single trigger during calibration.

- For the following types of calibration, the setting of the trigger source does not take effect. Those calibration operations are controlled by the internal trigger.
- Calibration using ECal
- Power calibration
- Receiver calibration
- Mixer converter calibration

Setting Trigger Source

Follow these steps to set the trigger source for calibration:

1. Press **Cal** Key.
2. Click **Cal Trig Source**, then select the trigger source you want to use.

Softkey	Function
Internal	Selects "Internal."

System	Selects "System."

Sliding and Offset Load Calibration

- Supported Calibration
- Sliding Load Calibration
- Offset Load Calibration

Other topics about Advanced Calibration

Supported Calibration

The following table shows the supported calibration for the sliding load and offset load.

Calibration	SOLT	Response	Enhanced Response	TRL-Line	Adapter Removal	Scalar Mixer	Vector Mixer (VBA)
Sliding Load	Yes	Yes	Yes	Yes	Yes	No	No
Offset Load	Yes	No	Yes	No	No	No	No

Sliding Load Calibration

A sliding load is defined by making multiple measurements of the device with the sliding load element positioned at various marked positions of a long transmission line. The transmission line is assumed to have zero reflections and the load element has a finite reflection that can be mathematically removed using a least squares circle fitting method.

A sliding load calibration can be very accurate when performed perfectly. It can also be very inaccurate when not using proper technique. For accurate results, follow the users manual instructions closely.

The following calibration kits are supported:

- 85050B 7 mm Calibration Kit
- 85052B 3.5 mm Calibration Kit
- 85054B 50 Ohm Type-N Calibration Kit
- 85056A 2.4 mm Calibration Kit

Procedure

Follow these steps to perform calibration using sliding load. The example demonstrates re-calibration for full 1-port calibration.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **Cal Kit**.
4. Select the calibration kit with sliding load, for example, **85052B**.
5. Press **Cal** key.
6. Click **Calibrate** > **1-Port Cal** > **Select Ports**.
7. Select the test ports for which you want to perform re-calibration for full 1-port calibration.
8. Make a OPEN connection on the selected test port.
9. Click **Open**.
10. Make a SHORT connection on the selected test port.
11. Click **Short**.
12. Make a LOAD connection on the selected test port with a sliding load.
13. Click **Calibrate** > **1-Port Cal** > **Load** > **Load 2 Sliding** for female load OR **Calibrate** > **1-Port Cal** > **Load** > **Load 5** for male load. The assignment of female and male loads varies from one calibration kit to another. In the case of 85052B, load 2 is a female and load 5 is a male load.
14. Slide the load to the desired position and click **Sliding Load**. Repeat this step for at least 5 times (till **Sliding Load 5**), then click **Done**. This step can be repeated at most 10 times only.
15. Click **Return**.
16. Now, all **Short**, **Open** and **Load** should have check mark on their left.
17. Click **Done**.

The sweep range setting should match the frequency range of the sliding load. Else, the load process is not executed, hence no check mark appear on the **Load** softkey. Similarly, if the frequency range is extended, the load process is also not executed. Refer to redefining a calibration kit to learn more about it.

Offset Load Calibration

An offset load is a compound standard consisting of a load element and two known offset elements (transmission lines) of different length. The

shorter offset element can be a zero-length (Flush-thru) offset. The load element is defined as a 1-port reflection standard. An offset load standard is used when the response of the offset elements are more precisely known than the response of the load element. This is the case with waveguide. Measurement of an offset load standard consists of two measurements, one with each of the two offset elements terminated by the load element. The frequency range of the offset load standard should be set so that there will be at least a 20 degree separation between the expected response of each measurement.

To specify more than two offset elements, define multiple offset load standards.

The following calibration kits are supported:

- X11644A X Band 90 Calibration Kit
- P11644A P Band WG 62 Calibration Kit
- K11644A K Band WG 42 Calibration Kit

NOTE The K11644A K Band WG 42 Calibration Kit is not verified.

Procedure

Follow these steps to perform calibration using offset load. The example demonstrates re-calibration for full 1-port calibration.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **Cal Kit**.
4. Select the calibration kit with sliding load, for example, **X11644A**.
5. Press **Cal** key.
6. Click **Calibrate** > **1-Port Cal** > **Select Ports**.
7. Select the test ports for which you want to perform re-calibration for full 1-port calibration.
8. Make a OPEN connection on the selected test port.
9. Click **Open**.
10. Make a SHORT connection on the selected test port.
11. Click **Short**.
12. Click **Calibrate** > **1-Port Cal** > **Load** > **Load 2 Offset Load**.
13. Make a LOAD connection on the selected test port with a offset load (Thru).

14. Click **Offset Line 1 Thru**.
15. Make a LOAD connection on the selected test port with a offset load (X-band Delay line).
16. Click **X-band Delay Line**.
17. Click **Done.**
18. Click **Return** twice.
19. Now, all **Short**, **Open** and **Load** should have check mark on their left.
20. Click **Done**.

The sweep range setting and Thru Offset frequency range should match the frequency range of the offset load. Else, the load process is not executed, hence no check mark appear on the **Load** softkey. Similarly, if the frequency range is extended, the load process is also not executed. Refer to redefining a calibration kit to learn more about it.

Modifying Calibration Kit Definition

- Definition of Terms
- Defining Parameters for Standards
- Redefining a Calibration Kit
- Setting Options for TRL Calibration
- Defining Calibration Kit Gender
- Saving Definition File of Calibration Kit
- Loading Definition File of Calibration Kit
- Restoring Definition File of Calibration Kit

Other topics about Advanced Calibration

In most measurements, the user can use pre-defined calibration kits as they are. However, it may be necessary to change the definition of a calibration kit (or create a new one) when changing the pre-defined connector between male and female (e.g. from OPEN (f) to OPEN (m)) or when a special standard is used or a high degree of accuracy is demanded. When it is necessary to change the definition of a calibration kit that contains a calibration device but no calibration kit model, the user must fully understand error correction and the system error model.

A user-defined calibration kit may be used in the following circumstances.

- When the user wants to use connectors other than those pre-defined in the calibration kits for the E5071C (e.g., a SMA connector).
- When the user wants to use different standards in place of one or more standards pre-defined in the E5071C. For example, when three offset SHORT standards are used instead of OPEN, SHORT, and LOAD standards.
- When the user wants to modify the standard model of a pre-defined calibration kit and turn it into a more accurate model. It is possible to perform better calibration if the performance of the actual standard is better reflected in the standard model. For example, you may need to define the 7-mm LOAD standard as 50.4 ohm instead of 50.0 ohm.

Definition of Terms

The terms used in this section are defined as follows:

Standard

An accurate physical device, for which the model is clearly defined, used to determine system errors. With the E5071C, the user may define up to 21 standards per calibration kit. Each standard is numbered from 1 through

21. For example, standard 1 for the 85033E 3.5-mm calibration kit is a SHORT standard.

Standard type

The type of standard used to classify a standard model based on its form and construction. Six standard types are available: SHORT, OPEN, LOAD, delay/THRU, Unknown Through, and arbitrary impedance.

Standard coefficient

The numeric characteristics of the standard used in the selected model. For example, the offset delay (32 ps) of the SHORT standard in the 3.5-mm calibration kit is a standard coefficient.

Standard class

A group of standards used in a calibration process. For each class, the user must select the standards to use from the 21 available standards.

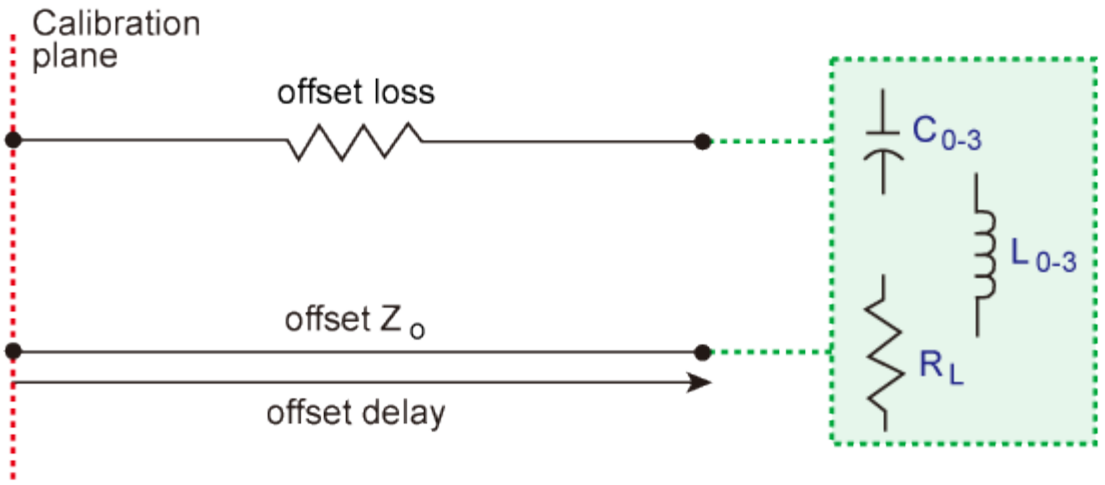
Subclass

You can register up to 8 standard types. This capability lets you specify a different standard for each frequency range. You must assign standards to subclasses you use.

Defining Parameters for Standards

The following figures show the parameters used in defining standards.

Reflection Standard Model (SHORT, OPEN, or LOAD)

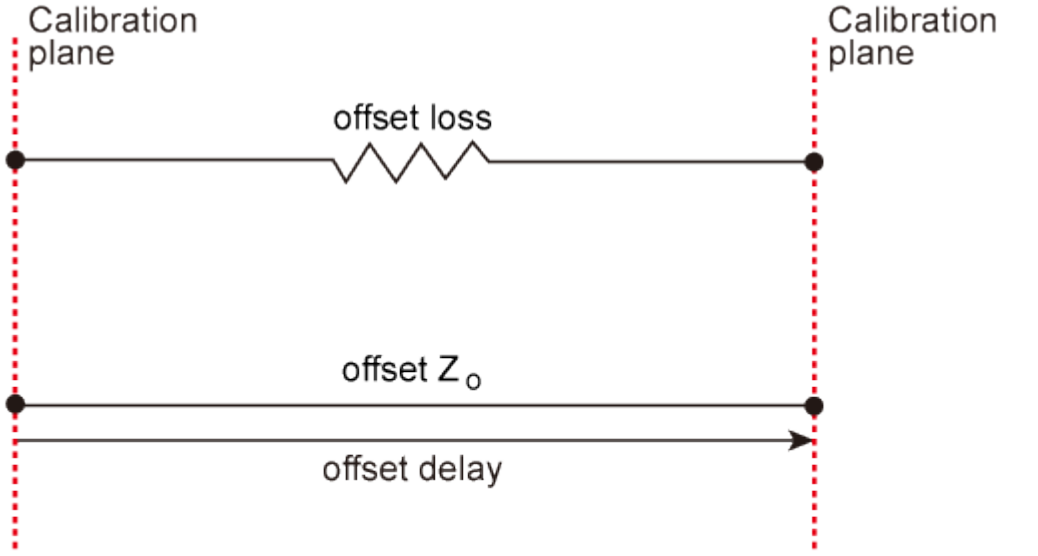


- Frequency range
- Coax or Waveguide
- Fixed, Sliding, or Offset (load only)

Model for Reflection Standard (short, open, load or arbitrary Impedance)

e5071c494

Transmission Standard Model (THRU)



- Frequency range
- Coax or Waveguide

Model for Transmission Standard (Thru)

e5071c495

Offset Z0

The offset impedance between the standard to be defined and the actual measurement plane. Normally, this is set to the system's characteristic impedance.

Offset Delay

The delay that occurs depending on the length of the transmission line between the standard to be defined and the actual measurement plane. In an OPEN, SHORT, or LOAD standard, the delay is defined as one-way propagation time (sec.) from the measurement plane to the standard. In a THRU standard, it is defined as one-way propagation time (sec.) from one measurement plane to the other. The delay can be determined through measurement or by dividing the exact physical length of the standard by the velocity coefficient.

Offset Loss

This is used to determine the energy loss caused by the skin effect along the length (one-way) of the coaxial cable. Loss is defined using the unit of ohm/s at 1 GHz. If you select waveguide as the media type, this parameter is ignored. In many applications, using the value 0 for the loss should not result in significant error. The loss of a standard is determined by measuring the delay (sec.) and the loss at 1 GHz and then substituting them in the formula below.

$$Loss\left(\frac{\Omega}{s}\right) = \frac{loss(dB) \times Z_0(\Omega)}{4.3429(dB) \times delay(s)}$$

C0, C1, C2, C3

It is extremely rare for an OPEN standard to have perfect reflection characteristics at high frequencies. This is because the fringe capacitance of the standard causes a phase shift that varies along with the frequency. For internal calculation of the analyzer, an OPEN capacitance model is used. This model is described as a function of frequency, which is a polynomial of the third degree. Coefficients in the polynomial may be defined by the user. The formula for the capacitance model is shown below:

$$C = (C0) + (C1 \times F) + (C2 \times F^2) + (C3 \times F^3)$$

F: measurement frequency
C0 unit: (Farads) (constant in the polynomial)
C1 unit: (Farads/Hz)

C2 unit: (Farads/Hz^2)
C3 unit: (Farads/Hz^3)

L0, L1, L2, L3

It is extremely rare for a SHORT standard to have perfect reflection characteristics at high frequencies. This is because the residual inductance of the standard causes a phase shift that varies along with the frequency. It is not possible to eliminate this effect. For internal calculation of the analyzer, a short-circuit inductance model is used. This model is described as a function of frequency, which is a polynomial of the third degree. Coefficients in the polynomial may be defined by the user. The formula for the inductance model is shown below.

$$L = (L0) + (L1 \times F) + (L2 \times F^2) + (L3 \times F^3)$$

F: Measurement frequency
L0 unit: [Henry] (the constant in the polynomial)
L1 unit: [Henry/Hz]
L2 unit: [Henry/Hz^2]
L3 unit: [Henry/Hz^3]

In most existing calibration kits, THRU standards are defined as "zero-length THRU," i.e., the delay and loss are both "0". Such a THRU standard does not exist, however. Calibration must be done with two test ports interconnected directly.

- The measurement accuracy depends on the conformity of the calibration standard to its definition. If the calibration standard has been damaged or worn out, the accuracy will decrease.

Redefining a Calibration Kit

This section provides the procedure to change the definition of a calibration kit.

- Procedure to select and define a calibration kit
- Procedure to select the standard type and define standard coefficient
- Procedure to define standard class

Procedure to select and define a calibration kit

1. Press **Cal** key.
2. Click **Cal Kit**, then select the calibration kit to be redefined. If the names (labels) of calibration kits were changed prior to operation, the new names will appear on the respective softkeys.

3. Click **Modify Kit**. To change the pre-defined connector type (e.g. OPEN(f) to OPEN (m)) skip to the specifty classes.
4. Click **Define STDs**.
5. Select the standard to be redefined from among standards numbered 1 through 21.
6. Click **No Name** > **Label** and input a new label for the standard using the keypad displayed on the screen.

Procedure to select the standard type and define standard coefficient

1. Press **Cal** key.
2. Click **Modify Kit**.
3. Click **Define STDs** and select the standard to be redefined from among standards numbered 1 through 21.
4. Click **STD Type**, then select the **type of standard**.

Softkey	Function
Open	Selects the OPEN standard
Short	Selects the SHORT standard
Load	Selects the LOAD standard
Delay/Thru	Selects the delay/THRU standard
Unknown Thru	Selects the Unknown THRU standard
Arbitrary	Selects the arbitrary impedance
None	Selects no standard type

5.
6. Set the **standard coefficient**.

Softkey	Function
C0	Sets C0

C1	Sets C1
C2	Sets C2
C3	Sets C3
L0	Sets L0
L1	Sets L1
L2	Sets L2
L3	Sets L3
Offset Delay	Sets the offset delay
Offset Z0	Sets the offset Z0. Specify 1 ohm when Waveguide is set for the media type.
Offset Loss	Sets the offset loss
Arb. Impedance	Sets an arbitrary impedance
Min. Frequency	Sets a start frequency. Specify cutoff frequency when Waveguide is set for the media type.
Max. Frequency	Sets a stop frequency. Specify cutoff frequency when Waveguide is set for the media type.
Media	Selects a media type, Coaxial or Waveguide.
Length Type	When the standard type is Load, this softkey can be selected from Fixed, Sliding or Offset.

7.

8. Repeat step 1 to step 5 to redefine all standards for which changes are necessary, then click **Return**.

Procedure to define standard class

1. Press **Cal** key.
2. Click **Modify Kit**.
3. Click **Specify CLSs**, then select the **class**.

Softkey	Function
Sub Class	Selects a subclass you want to use.
Open	Selects the OPEN class
Short	Selects the SHORT class
Load	Selects the LOAD class
Thru	Selects the THRU class
TRL Reflect	Selects TRL Reflect.
TRL Line/Match	Selcts TRL Line/Match.

4.
5. For TRL Thru, select **ports**. Select **Set All** to use the same standards for all test ports.

Softkey	Function
Port 1-2	Selects port 1 and port 2.
Port 1-3	Selects port 1 and port 3.
Port 1-4	Selects port 1 and port 4.
Port 2-3	Selects port 2 and port 3.

Port 2-4 Selects port 2 and port 4.

Port 3-4 Selects port 3 and port 4.

6.
7. Select the test port. Select **Set All** to use the same standards for all test ports.
8. Select the standards to be registered in the class from among standards numbered 1 through 21. To change the connectors between male and female (e.g. OPEN (f) to OPEN (m)), select the appropriately labeled standards here. For more information on changing the gender, refer to Defining Calibration Kit Gender.
9. Repeat the procedure until classes are defined for all test ports that need to be redefined, then click **Return**.
10. Repeat the procedure to redefine all classes that need to be modified, then click **Return**.
11. Click **Label Kit** and input a new label for the calibration kit by using the keypad displayed on the screen.

You can save the definition. See Saving Definition File of Calibration Kit.

Example of defining the TRL calibration kit

You need to enter the definition of the TRL calibration kit to perform TRL calibration. Follow these steps to define the following calibration kit given as an example.

- THRU (Delay 0 ps, Offset Loss 1.3 Gohm/s)
- REFLECT (SHORT, Delay 0 ps)
- MATCH (@0 to 2GHz)
- LINE1 (50-ohm transmission line, Delay 54.0 ps @ 2G to 7GHz)
- LINE2 (50-ohm transmission line, Delay 13.0 ps @ 7G to 32GHz)

Procedure to define the name of the calibration kit

1. Press **Cal** key.
2. Click **Cal Kit**, then select a User kit that has not been registered.
3. Click **Modify Kit** > **Label Kit [User]**, then type in a name you want.

Procedure to define Thru and Reflect

1. Click **Define STDs**.
2. Click **1:No Name** > **Label**, then type in **THRU**.

3. Select **STD Type** > **Delay/Thru**.
4. Set **Offset Loss** to 1.3Gohm/s, and **Offset Delay** to 0.
5. Click **Return** to return to the Define STD menu.
6. In the same way, repeat the procedure to enter the definition of REFLECT to No. 2.
7. Select **SHORT** for STD Type.

Procedure to define Match

1. Click **3:No Name** > **Label**.
2. Type in **MATCH <2G**.
3. Click **STD Type** > **Load**.
4. Set **Max Frequency** to 2GHz.
5. Click **Return** to return to the Define Std menu.

Procedure to define Line 1/2

1. Click **4:No Name - Label**.
2. Type in **LINE <7G**.
3. Click **STD Type - Thru**.
4. Set **Offset Delay** to 54 ps.
5. Set **Min Frequency** to 2 GHz.
6. Set **Max Frequency** to 7 GHz.
7. Click **Return** to return to the Define Std menu.
8. Click **5:No Name - Label**.
9. Type in **LINE >7G**.
10. Click **STD Type** > **Thru**.
11. Set **Offset Delay** to 13 ps.
12. Set **Min Frequency** to 7 GHz.
13. Click **Return** to return to the Define Std menu.
14. Click **Return** to return to the Modify Cal Kit menu.

Procedure to register Thru/Reflect/Match to sub class 1

1. Click **Specify CLSs** > **Sub Class.**
2. Select **Sub Class1**.
3. Click **TRL Thru** > **Set All** > **THRU** > **Return**.
4. Click **TRL Reflect** > **REFLECT**.
5. Click **TRL Line/Match** > **Set All** > **MATCH <2G** > **Return.**

Procedure to register Line 1/2 to subclass 2/3

1. Click **Sub Class2**.
2. Click **TRL Line/Match** > **Set All** > **LINE <7G** > **Return.**
3. Click **Sub Class3**.
4. Click **TRL Line/Match** > **Set All** > **LINE >7G** > **Return.**
5. Press **Cal** key and check that the name you specified is selected as Cal Kit.

By assigning Match and Line 1/2 to subclass 1/2/3 respectively, you can calibrate 3 standards with different frequency bands for TRL line/match calibration.

Setting Options for TRL Calibration

This section describes how to set the reference impedance and measurement data of the standard used for calculating the calibration plane for TRL calibration.

1. Press **Cal** key.
2. Click **Cal Kit**, then select a calibration kit.
3. Click **Modify Cal Kit** > **TRL Option**.
4. Click **Impedance** to select the **reference impedance**.

Softkey	Function
Line	Calculates the calibration coefficient using the characteristic impedance of the line standard as the reference impedance. The impedance of the line standard is used as the center of the Smith chart, and the calibration coefficient is calculated using the line standard's as the direction.
System	Calculates the calibration coefficient using the system impedance as the reference impedance. Select this when the line impedance and the

test port impedance differ.

5. Click **Reference Plane** to select the **measurement data of the standard** used for calculating the reference plane.

Softkey	Function
Thru	Uses the length of the thru/line standard to calculate the calibration plane.
Reflect	Uses the reflection coefficient of the reflection standard to calculate the calibration plane.

6.

Setting a media type for the calibration kit

You can set a media type for the standard you use.

1. Press **Cal** key.
2. Click **Cal Kit**, then select a calibration kit.
3. Click **Modify Cal Kit** > **Modify Cal Kit**.
4. Click **Define STDs**, and select a standard.
5. Click **Media**, and select a **media type**.

Softkey	Function
Coaxial	Selects coaxial as the media type.
Waveguide	Selects waveguide as the media type.

6.

7. If you select waveguide as the media type, set the system impedance and the characteristic impedance to 1ohm .
8. Depending on the media type you use, the calculation method of the electrical delay, which is required to correct the phase delay, differs.

You can save the definition. See Saving Definition File of Calibration Kit.

Defining Calibration Kit Gender (m/f)

The gender of a calibration can be set using the subclass function in the ENA. For example, subclass 1 can be used as "male" and subclass 2 can be used as "female" or vice-versa. The display label, for example "**Open (f)**" or "**Open (m)**", is just a display label and can be edited using **Cal** > **Modify Kit** > **Define STDs** and hence cannot be used to change the gender of the calibration kit. To change the gender of the calibration kit, the E5071C subclass function must be used.

For example, the gender of the default calibration kit (85032F) can be checked using the following procedure:

1. Press **Cal** key.
2. Click **Cal Kit** and select **85032F**.
3. Select **Cal** > **Modify Cal Kit** > **Specify CLSs** > **Sub Class**.

The subclass gender is set from the E5071C side and not from the Calibration kit side.

If both male/female are set using subclass in advance, the gender which is used for the calibration can also be changed programmatically using SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SUBClass

- The gender setting option using subclass is defined from the E5071C connector type and not from the standard calibration kit gender type.

Saving Definition File of Calibration Kit

You can save the definition file of the calibration kit for each standard into a file on a storage device, then recall it later to reproduce the definition.

1. Press **Cal** key.
2. Click **Cal Kit**, then select a calibration kit.
3. Click **Modify Cal Kit** > **Export Cal Kit...** to open the dialog box.
4. Specify a folder, enter a file name, and click **Save**.
   - The file is saved with the (.ckx) extension. Do not open and modify it directly.
   - Never modify the contents (folders and files) of drives except for drive D. Modifying the contents of drives other than drive D may give serous damage to the functions and performance of the analyzer.

Loading Definition File of Calibration Kit

1. Press **Cal** key.
2. Click **Cal Kit**, then select a calibration kit.

3. Click **Modify Cal Kit** > **Import Cal Kit...** to open the dialog box.
4. Specify a folder, enter a file name, and click **Open**.

Restoring Definition File of Calibration Kit

1. Press **Cal** key.
2. Click **Cal Kit**, then select a calibration kit.
3. Click **Modify Cal Kit** > **Restore Cal Kit...**.
4. Click **OK** to restore the definition of the calibration kit selected by **Cal Kit** to factory default settings.

Specifying Different Standards for Each Frequency

- Overview
- Defining Standards for Each Subclass
- Disabling Standards Defined for a Subclass
- Possible Cases of Frequency Ranges Using Subclasses

Other topics about Advanced Calibration

Overview

This section demonstrates the procedure to define a different open standard for each frequency band, based on the following information:

Standard	Label name	Frequencies to be defined
Open	Open 3G	1 GHz - 3 GHz
Open	Open 6G	3 GHz - 6 GHz

1. Press **Cal** key.
2. Click **Cal Kit**.
3. Select a calibration kit you want to define.
4. Click **Modify Kit** > **Define STDs**.
5. Select #1, click **Label**, and enter **Open 3G** using the character input pad that appears on the screen.
6. Click **Open** as the standard type (**STD Type**).
7. Set a necessary standard coefficient.
8. Enter **1G** as the minimum frequency for **Min Frequency**.
9. Enter **3G** as the maximum frequency for **Max Frequency**.
10. Click **Return**.
11. In the same way, select #2, press **Label**, and enter **Open 6G** using the character input pad that appears on the screen.
12. Select **Open** as the standard type (**STD Type**).
13. Set a necessary standard coefficient.
14. Enter **3G** as the minimum frequency for **Min Frequency**.
15. Enter **6G** as the maximum frequency for **Max Frequency**.
16. Click **Return**.

Defining Standards for Each Subclass

This section demonstrates the procedure to specify a different OPEN standard for each frequency band, using subclasses #1 and #2. In this example, the standards created in Defining different standard for each frequency band are used.

1. Press **Cal** Key.
2. Click **Cal Kit**.
3. Select a calibration kit you want to use.
4. Click **Modify Kit** > **Specify CLSs**.
5. Click **Sub Class** > **Sub Class 1**.
6. Select the standard type **Open**, press **Set All**, make settings for all ports.
7. Select #1 (**Open 3G**).
8. Click **Return**.
9. Then, press **Sub Class**, and select **Sub Class 2**.
10. Select the standard type **Open**, press **Set All**, make settings for all ports.
11. Select #2 (**Open 6G**).
12. Click **Return**.

Disabling Standards Defined for a Subclass

The following procedure shows how to disable a standard defined for a subclass. You cannot disable subclass 1 because at least one standard must exist. The following procedure shows how to disable an OPEN standard of subclass 2.

1. Press **Cal** key.
2. Click **Cal Kit**.
3. Select a calibration kit you want to use.
4. Click **Modify Kit** > **Specify CLSs**.
5. Click **Sub Class** > **Sub Class 2**.
6. Select the standard type **Open**, click **Set All**, make settings for all ports.
7. Select **None**.
8. Click **Return**.

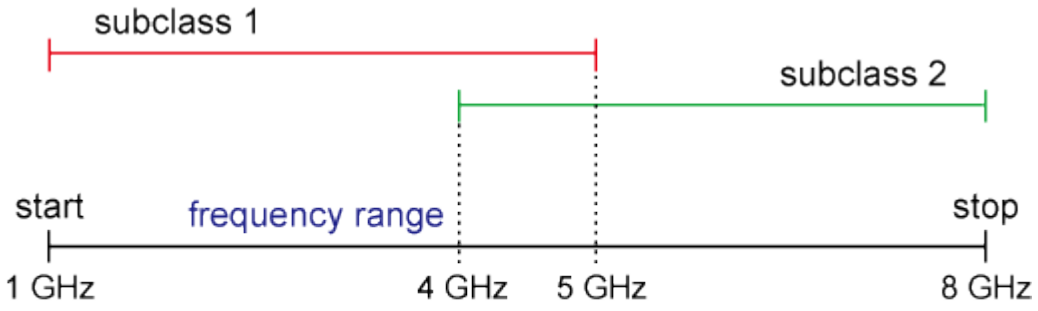
Possible Cases of Frequency Ranges Using Subclasses

By using several subclasses, you can set a different standard for each frequency, but note the following on frequency ranges. The following table shows possible cases and whether calibration is possible.

Possible cases of frequency ranges	Execution of calibration
Example 1, When frequency ranges specified with subclasses overlap	Possible
Example 2, When frequency ranges specified with subclasses do not cover a part of a measurement frequency range	Impossible
Example 3, When a frequency range specified with a subclass lies outside the measurement frequency range	Possible
Example 4, When a part of a frequency range specified with a subclass lies outside the measurement frequency range	Possible

Example 1, When frequency ranges specified with subclasses overlap

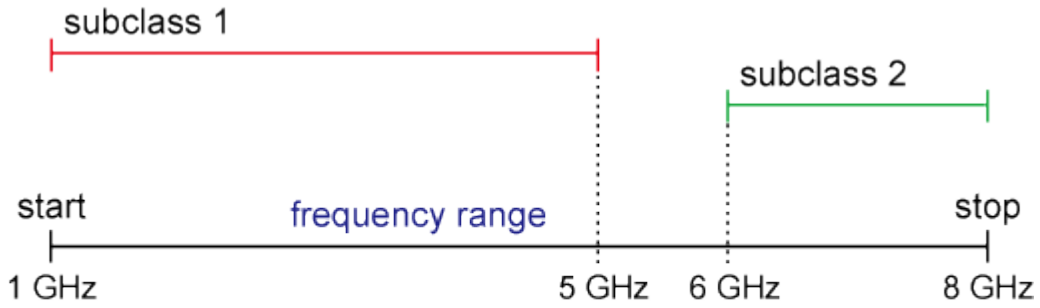
For the measurement frequency range (1 GHz - 8 GHz), if you execute calibration with a standard of subclass 1 (1 GHz-5 GHz) and then with a standard of subclass 2 (4 GHz-8 GHz), the standard last executed is applied for the overlapping portion (4 GHz-5 GHz).



e5071c470

Example 2, When frequency ranges specified with subclasses do not cover a part of a measurement frequency range

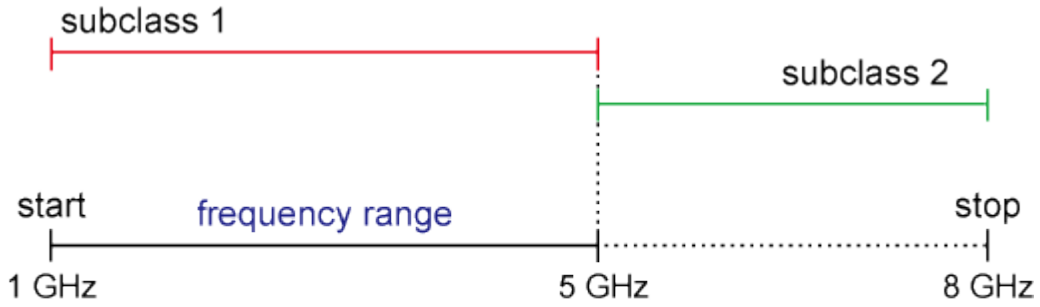
For a measurement frequency range (1 GHz to 8 GHz), if you execute calibration with a standard of subclass 1 (1 GHz to 5 GHz) and then with a standard of subclass 2 (6 GHz to 8 GHz), **Done** is NOT available for the undefined portion (5 GHz-6 GHz).



e5071c468

Example 3, When a frequency range specified with a subclass lies outside the measurement frequency range

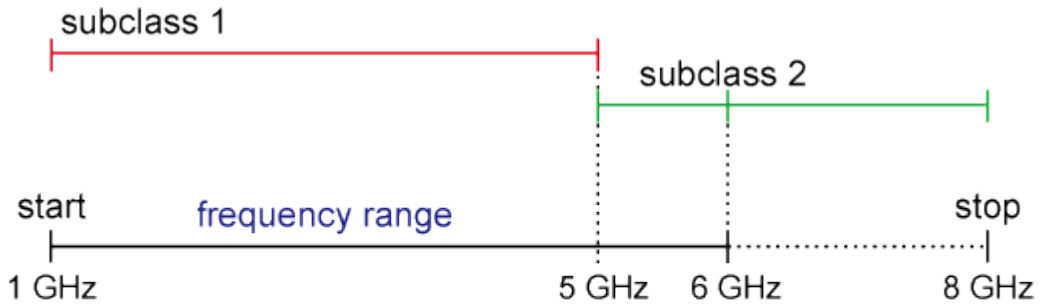
For a measurement frequency range (1 GHz to 5 GHz), if you define a standard of subclass 1 (1 GHz to 5 GHz) and a standard of subclass 2 (5 GHz to 8 GHz), calibration possible for subclass 2 although it is out of the measurement frequency range. The calibration coefficients, however, are disregarded. (There is no check mark on the softkey.) Note that, because the standard of subclass 1 covers the measurement frequency range, **Done** is available.



e5071c467

Example 4, When a part of a frequency range specified with a subclass lies outside the measurement frequency range

For a measurement frequency range (1 GHz to 6 GHz), if you define a standard of subclass 1 (1 GHz - 5 GHz) and a standard of subclass 2 (5 GHz to 8 GHz), a part of subclass 2 (6 GHz to 8 GHz) lies outside the measurement frequency range, but the other part (5 GHz to 6 GHz) lies within it, so calibration is possible.


subclass 1
subclass 2
start
frequency range
stop
1 GHz
5 GHz
6 GHz
8 GHz

e5071c469

Receiver Calibration

- Overview
- Turning Receiver Error Correction ON/OFF
- Selecting Target Port for Error Correction
- Measuring Calibration Data

Other topics about Advanced Calibration

Overview

The E5071C has a function to calibrate the gain of the individual receivers in absolute value measurement.

The receiver calibration function calibrates the gain of the receiver by inputting the output power of a stimulus port that has been assigned a correct value by the power calibration to the port of the receiver you need to calibrate.

- Receiver calibration is valid only for the parameters of absolute value measurement.

Turning Receiver Error Correction ON/OFF

The receiver calibration data are acquired for each channel/test port, so the gain error correction can be turned ON/OFF independently for any channel or port.

The status of the gain error correction for each channel is indicated by one of the symbols in the channel status bar in the lower part of the window, as shown in the following table.

Symbol	Status of receiver error correction
RC (displayed in blue)	Error correction is performed for all of the receiver ports.
RC (displayed in grey)	Error correction is performed for some of the receiver ports.
RC? (displayed in blue)	Error correction is performed for all of the receiver ports. Interpolated calibration data is used.
RC? (displayed in grey)	Error correction is performed for some of the receiver ports. Interpolated calibration data is used.
RC! (displayed in	Error correction is performed for all of the

blue)	receiver ports. Extrapolated calibration data is used.
RC! (displayed in grey)	Error correction is performed for some of the receiver ports. Extrapolated calibration data is used.
--- (displayed in gray)	Error correction is not performed. (At least one receiver port is turned on for error correction, but valid calibration data are not available.)
None	Error correction is not performed. (Error correction is turned off for all of the receiver ports.)

Turning receiver error correction ON/OFF

The error correction of the receiver is automatically turned on when the calibration data are measured. You can also turn this ON/OFF explicitly by following the steps below.

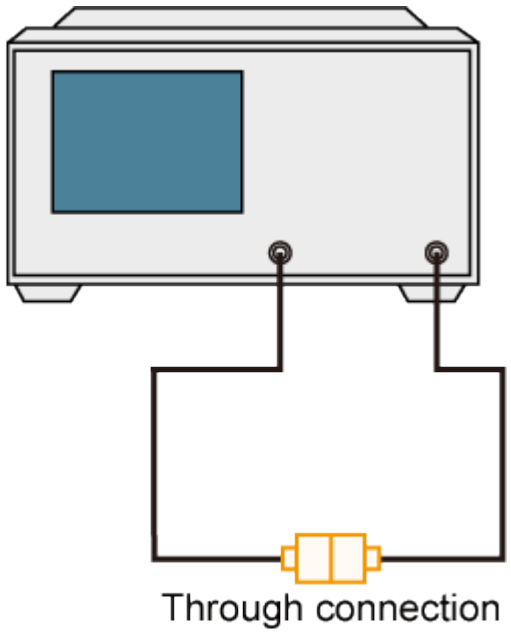
1. Press **Cal** key.
2. Press **Receiver Calibration**.
3. Select a port (see Selecting target port for error correction).
4. Click **Correction**. Each time the key is pressed, ON and OFF switches over alternately.

Selecting Target Port for Error Correction

The error correction of receiver ports is performed for each channel/port, and you can set the following items for them:

- Turning on or off error correction
- Calibration data

1. Press **Cal** key.
2. Press **Receiver Calibration**.
3. Press **Channel Next**/**Channel Prev** keys to select a channel.
4. Click **Select Port**.



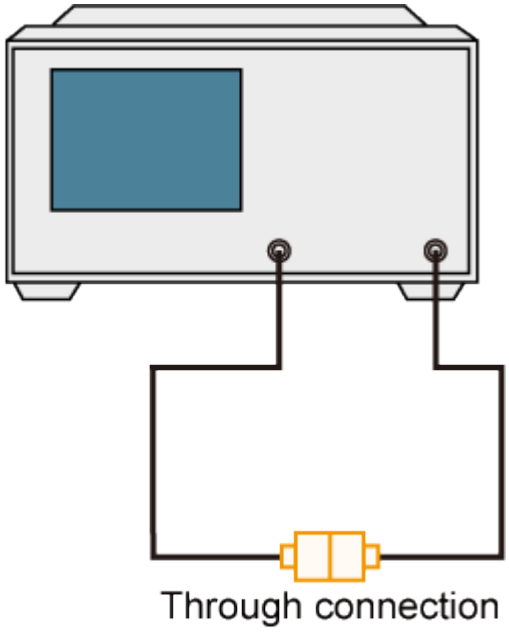
e5071c321

5. Click the softkey corresponding to the port you want to calibrate.

Measuring Calibration Data

- The power calibration information on both the receiver port and source port is used to calculate calibration coefficients. The accuracy of receiver calibration will increase if power calibration is implemented for both the receiver port and the source port before starting receiver calibration. For information on power calibration, refer to Power Calibration.

1. Press **Cal** key.
2. Press **Receiver Calibration**.
3. Select a port (see Selecting target port for error correction).
4. Click **Source Port**.
5. Select the stimulus port for which you want to perform power calibration.
6. Connect the selected stimulus port by cable to any port you want to calibrate.



e5071c321

7. Click **Receiver Calibration** > **Power Offset** to enter the offset value. Receiver calibration is done at the condition of (source power)+(Specified offset).

   For example, if the source Power is -10 dBm, then:

- If specified offset is 0 dBm, then Source Power Reading after calibration would be -10 dBm.
- If specified offset is 10 dBm, then Source Power Reading after calibration would be 0 dBm.
- If specified offset is -10 dBm, then Source Power Reading after calibration would be -20 dBm.

10. The value of offset is independent for each channel.

8. The E5071C provides an option to calibrate both the Receiver and Transmitter end, or only the Receiver end, or only the Transmitter end:

   - Click **Receiver Calibration** > **Calibrate Both** to start measurement of calibration data for both receiver and transmitter end. This step allows the gain of the receiver to be calibrated, turning on the error correction function automatically.

9. From Firmware revision 9.2, **Take Cal Sweep** softkey is renamed as the **Calibrate Both** softkey.

- Click **Receiver Calibration** > **Calibrate R** to calibrate the receiver end only.
- When **Calibrate R** is selected, the **Select Port** setting is used and the **Source Port** setting is ignored.
- Click **Receiver Calibration** > **Calibrate T** to calibrate the transmitter end only.
- When **Calibrate Both** or **Calibrate T** is selected, the source port should be different from measurement port.

Power Calibration

- Overview
- Turning Power Level Error Correction ON/OFF
- Preparing Power Meter and Sensor
- Selecting Target Port of Error Correction
- Setting Loss Compensation
- Measuring Calibration Data

Other topics about Advanced Calibration

Overview

The E5071C has a calibration feature for power level output that uses the power meter (power calibration).

The power calibration function outputs a stimulus signal with a more accurate power level (closer to the set value) by measuring calibration data (power level) in advance with the power meter and sensor. Then this function performs error correction of the power level by using the calibration data.

Turning Power Level Error Correction ON/OFF

Power calibration data are acquired for each channel/test port, and you can turn ON/OFF the power level error correction independently for any channel or test port.

The status of the power level error correction of each channel is indicated by one of the symbols shown in table below in the channel status bar in the lower part of the window.

Symbol	Status of power level error correction
PC (displayed in blue)	Error correction is performed for all stimulus ports
PC (displayed in gray)	Error correction is performed for some stimulus ports
PC? (displayed in blue)	Error correction is performed for all stimulus ports; interpolated calibration data is used. If you turn on the error correction function when the stimulus setting is different from that when the power calibration data were acquired, power level error correction is performed with interpolated calibration data only when calibration data can be interpolated (extrapolation is not performed for calibration data).
PC? (displayed in	Error correction is performed for some stimulus

gray)	ports; interpolated calibration data is used. If you turn on the error correction function when the stimulus setting is different from that when the power calibration data were acquired, power level error correction is performed with interpolated calibration data only when calibration data can be interpolated (extrapolation is not performed for calibration data).
--- (displayed in gray)	Error correction is not performed (some stimulus ports have error correction set to on, but there is no valid calibration data)
None	Error correction is not performed (error correction is off for all stimulus ports)

Procedure to turn ON/OFF power level error correction

Power level error correction is automatically turned on when you execute the measurement of calibration data. You can turn it on or off as necessary by using the following procedure.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Select a port.
4. Click **Correction**. Each press toggles this selection between ON/OFF.

Preparing Power Meter and Sensor

To execute power calibration, you need to prepare the power meter and power sensor used to acquire power calibration data.

- Power calibration may not be executed when the specified frequency and level is out of range of the power sensor specification.

The table below shows available power meters and recommended power sensors for power calibration.

Product Description	Model Number	Supported ENA Firmware Revision
Power meters	Agilent 437B, 438A, E4418A/B, E4419A/B, N1911A/N1912A (w/ USB/GPIB interface)	All
	HP EPM-441A/442A, EPM-P E4416A/E4417A	

	Agilent N1911A/N1912A (USB to USB direct connection)	A.09.2x and above
Power sensors	Agilent 8482A, E4412A, N1921A	All
	HP ECP-E18A	
	Agilent N1922A, E4413A, E9300A, E9301A, E9304A, E9304A-H18/H20, E9300H, E9300H-H19, E9301H, E9300B, E9301B	A.09.1x and above
	Agilent U2000A/B/H, U2001A/B/H, U2002A/H, U2004A	A.09.2x and above

Preparing to control the power meter

When acquiring power calibration data, the power meter is controlled via GPIB or USB (for USB enabled Power Meters) from the E5071C.

To control the power meter through GPIB from the E5071C, connect the USB port of the E5071C and the GPIB/USB connector of the power meter through the USB/GPIB interface and set the GPIB address of the connected power meter with the E5071C.

- The USB/GPIB interface and USB enabled power sensor must be ready to use. When you connect them for the first time, new hardware wizard runs. For the procedure, refer to Setting system controller (USB/GPIB interface). The hardware wizard procedure for USB enabled power sensor is same. At the last step in the procedure, Assign USB device alias dialog box will be displayed. Select "When a new USB device is plugged in" or "Never show this dialog" instead of "Each time a USB devices is plugged in", or ENA sometimes fail to recognize the USB power sensor.

  For detailed information, refer to "Hardware Installation and Configuration", page 15 to 19 of USB Power sensor operation manual, U2000-90405 November 14, 2008 3rd edition.
- Check whether the USB power sensor is properly installed by referring to *"Hardware Installation and Configuration"* of USB Power sensor operation manual, U2000-90405. If you face trouble while connecting the USB power sensor for the first time, or just after reconnecting, try to restart the ENA firmware and re-connect the USB power sensor again.

Selecting the power meter

Use the following procedure to select the type of GPIB or USB enabled power meter:

1. Press **System** key.
2. Click **Misc Setup** > **Power Meter Setup** > **Select type** > **GPIB|USB**
3. From Firmware revision 9.2, E5071C supports USB enabled power sensors.

Using GPIB controlled power meter

Use the following procedure to set the GPIB address of the power meter:

1. Press **System** key.
2. Click **Misc Setup** > **GPIB Setup** > **Power Meter Address**.
3. Enter the GPIB address of the power meter you are using.
4. You can also set the GPIB address of power meter by pressing **System** > **Misc Setup** > **Power Meter Setup** > **GPIB Address**

Setting power sensor calibration factor table

- Before using the power sensor calibration factor table of the E5071C, set the calibration factor to 100% and then calibrate the power sensor.

When you use the 437B or 438A as the power meter, you need to set the power sensor calibration factor table with the E5071C.

If you use a power meter other than the 437B or 438A, refer to the following table.

Power sensor	Setting of calibration factor table
8482A	Set the calibration factor table with the E5071C only when you do not set the calibration factor table with the power meter. If you set the calibration factor table with both the power meter and the E5071C, calibration is executed by both of them and you cannot obtain correct measurement results.
E4412A ECP-E18A	You do not need to set the calibration factor table with the E5071C. Even if you set the calibration factor table with the E5071C, this setting would be ignored.

- If you use the E4418A, E4419A, EPM-441A, or EPM-442A with Firmware revision Ax.02.00 or earlier and you set the calibration factor table with the E5071C, calibration is executed by both the

power meter and the E5071C due to this Firmware revision, and thus you cannot obtain correct measurement results. Therefore, never set the calibration factor table under this condition with the E5071C.

Use the following procedure to set the power sensor's calibration factor table:

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Click **Sensor A Settings** (for power sensor connected to channel A) or **Sensor B Settings** (for power sensor connected to channel B).
4. Click **Ref Cal Factor**.
5. Enter the reference calibration factor (the calibration factor at 50 MHz).
6. According to the calibration factor data attached to the power sensor, set the frequency (**Frequency**) and the coefficient (**Factor**) of the calibration factor table by using the hardkeys and softkeys.
7. Pressing **Preset** > **OK** does not affect the current setting of the reference calibration factor and the calibration factor table.
8. For a frequency other than one set in the table, a value obtained by linear interpolation of the calibration factors at the 2 points adjacent to the frequency is used. If a frequency is lower than the lowest frequency in the table, the calibration factor at the lowest frequency is used; if larger than the highest frequency in the table, the calibration factor at the highest frequency is used.
9. When setting the table by using the front panel keys or the keyboard, you need to first set the focus on (select) the operation target (table or softkey). You can change the focus by pressing **Focus** key in the ENTRY block. When the focus is placed on the table, the window frame of the table is displayed as bright as the window frame of the active channel. When the focus is placed on the softkey menu, the softkey menu title area is displayed in blue.

Saving power sensor calibration factor table

You can save the power sensor calibration factor table as a CSV (Comma Separated Value) format file.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Click **Sensor A Settings** or **Sensor B Settings**.

4. Click **Export to CSV File** to open the Save As dialog box. At this time, CSV File (extension *.csv) is selected as the file type.
5. Enter a file name in the **File Name** box and press the **Save** button to save the power sensor calibration factor table.

Recalling power sensor calibration factor table

By recalling a power sensor calibration factor table saved in the CSV format according to saving power sensor calibration factor table, you can set the power sensor calibration factor table.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Click **Sensor A Settings** or **Sensor B Settings**.
4. Click **Import from CSV File** to open the Open dialog box. At this time, CSV File (extension*.csv) is selected as the file type.
5. Select the CSV format file you want to import and press the **Open** button to recall the power sensor calibration factor table.
6. This operation is not guaranteed under the following two conditions:
   1) you imported a CSV format file created/edited on a spreadsheet program, or
   2) you imported a CSV format file that had been exported according to Saving power sensor calibration factor table but then modified.

Using USB power sensor

Use the following procedure to select the USB power meter:

1. Press **System** key.
2. Click **Misc Setup** > **Power Meter Setup** > **USB**.
3. Select desired USB power sensor.
4. When USB enabled power sensor is plugged to ENA, it will take about 30 seconds to be ready. If you perform calibration or zeroing before the ENA is ready, an error will occur.

Performing Zeroing/Calibration

Zeroing a power sensor is performed in order to reduce zero measurement offset and noise impact to improve the accuracy of RF power measurement. The U2000 Series USB power sensors have two types of zeroing INTernal zeroing and EXTernal zeroing. For more information, refer to http://cp.literature.agilent.com/litweb/pdf/U2000-90405.pdf.

Use the following procedure to perform zeroing/calibration:

1. Press **Cal** key.

2. Click **Power Calibration** > **Sensor A/B Settings** > **Zero Type** > **INT**|**EXT** to select from internal or external zeroing.
3. Click **Power Calibration** > **Sensor A/B Settings** > **Zero/Calibrate Sensor**
4. For U2000 Series USB power sensors, Sensor B setting is disabled.

Selecting Target Port of Error Correction

The power level error correction is executed for each channel/test port. You can set the following items for each channel/test port:

- ON/OFF of error correction
- Setting of loss compensation
- Selection of the power sensor
- Number of power level measurements at one measurement point
- Calibration data

Follow the steps below to select the test port for which you want to set/execute power level error correction.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Press **Channel Next**/**Channel Prev** keys to select the channel.
4. Click **Select Port**.
5. Click the softkey corresponding to the test port you want to select.

Setting Loss Compensation

If you need to apply a signal with a more accurate power level to the DUT, you can use the loss compensation feature to obtain calibration data for correcting the difference in power loss due to the difference in connection method (cable, adapter, etc.) between the time when the power calibration data are measured and when the actual DUT is measured.

The loss compensation feature corrects the power measurement result based on the preset power loss data. More specifically, if you turn on the loss compensation, the measurement result of the power level obtained in the power calibration data measurement is a value obtained by adding the loss value set in the loss compensation table to the measured power value.

Turning ON/OFF loss compensation

Follow the steps below to turn ON/OFF the loss compensation.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Select a port (refer to Selecting target port of error correction).

4. Click **Loss Compen** > **Compensation**. Each press toggles the selection between ON/OFF.

Creating loss compensation table

Follow the steps below to set the loss compensation table.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Select a port (refer to Selecting target port of error correction).
4. Click **Loss Compen**
5. Set the frequency (**Frequency**) and the loss (**Loss**) of the loss compensation table by using the hardkeys and softkeys.
    1. Pressing **Preset** > **OK** does not affect the current setting of the reference calibration factor and the calibration factor table.
    2. For a frequency other than one set in the table, a value obtained by linear interpolation of the calibration factors at the 2 points adjacent to the frequency is used. If a frequency is lower than the lowest frequency in the table, the calibration factor at the lowest frequency is used; if larger than the highest frequency in the table, the calibration factor at the highest frequency is used.
    3. When setting the table by using the front panel keys or the keyboard, you need to first set the focus on (select) the operation target (table or softkey). You can change the focus by pressing **Focus** key in the ENTRY block. When the focus is placed on the table, the window frame of the table is displayed as bright as the window frame of the active channel. When the focus is placed on the softkey menu, the softkey menu title area is displayed in blue.

Saving loss compensation table

You can save the loss compensation table as a CSV (Comma Separated Value) format file.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Select a port (refer to Selecting target port of error correction).
4. Click **Loss Compen**.
5. Click **Export to CSV File** to open the Save As dialog box. At this time, CSV File (extension*.csv) is selected as the file type.

6. Enter a file name in the **File Name** box and press the **Save** button to save the loss compensation table.

Recalling loss compensation table

By recalling a loss compensation table saved in the CSV format according to the Saving loss compensation table, you can set this loss compensation table.

1. It is possible to recall a file from a different channel/port from where it was saved.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Select a port (refer to Selecting target port of error correction).
4. Click **Loss Compen**.
5. Click **Import from CSV File** to open the Open dialog box. At this time, CSV File (extension*.csv) is selected as the file type.
6. Select the CSV format file you want to import and press the **Open** button to recall the loss compensation table.

7. This operation is not guaranteed under the following two conditions: 1) you imported a CSV format file created/edited on a spreadsheet program, or 2) you imported a CSV format file that had been exported according to Saving loss compensation table but then modified.

Setting a tolerance for power calibration

The E5071C provides an option to set the tolerance value for power calibration. This function is to save time for the power calibration by aborting the calibration process once the measurement value goes outside the specified tolerance and is NOT to settle the power level within the specified tolerance. If the power calibration is aborted, the power-level error-correction function is not turned on.

Setting the tolerance

Follow the steps below to set the tolerance for power calibration:

1. Press **Cal** key.
2. Click **Power Calibration** > **Tolerance**.
3. Enter the tolerance value for power calibration.

Measuring Calibration Data

- Before measuring calibration data, you need to execute the zero adjustment and calibration of the power sensor. For information

on how to execute these operations, see the manual of the power meter you are using.

When using the power sensor calibration factor table of the E5071C, set the calibration factor to 100% and execute the calibration of the power sensor.

1. Press **Cal** key.
2. Click **Power Calibration**.
3. Select a port (refer to Selecting target port of error correction).
4. Click **Use Sensor**. Each press toggles between channel A and channel B. If you use a power meter with one channel, channel A is always selected.
5. Click **Num of Readings**.
6. Enter the number of power level measurements at one measurement point (averaging factor).
7. Connect the power sensor for the selected channel to the selected port.
8. Click **Take Cal Sweep** to start the measurement of calibration data.
9. You can abort the measurement by pressing **Abort** during measurement.
10. When the measurement is complete, the power level error correction is automatically turned on.

11. If the power meter GPIB address is not set correctly or if the power sensor is not connected to the specified channel, an error occurs and calibration data are not measured.

- If appropriate calibration data cannot be obtained, an error may occur for each sweep after the measurement of calibration data is complete. In this event, turn off the power level error correction, check the connection and setting, and then measure the calibration data again.

Partial Overwrite Calibration

- Overview
- Procedure

Other topics about Advanced Calibration

Overview

The partial overwrite function is used to perform partial measurement after the execution of calibration, and it overwrites the calibration coefficients.

There are three types of calibration coefficients: Er, Es, Ed for reflection, Et for transmission, and Ex for isolation. If some of them do not provide satisfactory calibration, you can use this function to re-calculate the calibration coefficients by measuring an applicable standard only instead of measuring all standards again.

- When the calibration coefficients become inappropriate over time or the status on the E5071C side from the calibration surface changes due to replacement of a cable or connector, you need to also perform thru measurement when partial overwrite is required for reflection or isolation measurement.
- The adapter removal and partial overwrite function is only available when calibration status is [Corr] and not for [C?] or [C!].

  Partial overwrite is not available if no calibration has been done. You cannot append calibration coefficients to previous calibrations. For example, you cannot realize full 4-port calibration by performing additional calibration for 1 port after the execution of full 3-port calibration. The partial overwrite function is used to make measurements for previous calibration coefficients and overwrite them.

Procedure

Follow these steps to execute the partial overwrite function. The example demonstrates re-calibration for full 2-port thru calibration.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **2-Port Cal** > **Select Ports**.
4. Select the test ports for which you want to perform re-calibration for full 2-port calibration. (In the following procedure, the test ports you select are expressed as test ports x and y.)

5. Click **Transmission**.
6. Make a thru connection between selected test ports x and y (between the connectors to which the DUT is connected).
7. Click **Port x-y Thru** (**x** and **y** are the thru-connected port numbers) to start the re-measurement of the calibration standard.
8. Click **Return**.
9. Click **Overwrite** to finish the re-calibration for the full 2-port calibration. At this point, the calibration coefficients are re-calculated and saved.

.

Simplified Calibration

- Overview
- Simplified Full 3/4-port Calibration
- Simplified 3/4-port TRL Calibration

Other topics about Advanced Calibration

Overview

The simplified calibration calculates the calibration coefficients by skipping part of thru measurement (and line measurement for TRL measurement) that is necessary for the full 3/4-port calibration and the 3/4-port TRL calibration.

Simplified Full 3/4-port Calibration

In the simplified full 3/4-port calibration, since the calibration coefficients are calculated while omitting part of the thru measurement data, the effect of errors when acquiring calibration data becomes larger than in the normal full 3/4-port calibration. Therefore, you must pay special attention to the following points when measuring data for the simplified full 3/4-port calibration.

- The standard used for measurement must match its definition value.
  - Use a standard that provides good repeatability (stability).
  - Do not omit the length of the thru when defining the standard.
  - When using a user-created standard, verify the definition value.
  - For the N connector, note the discrimination between male and female.
- Realize high reliability and repeatability for measurement.
  - Reduce the difference in external environment (such as temperature difference) between conditions of measuring calibration data and conditions of measuring actual data.
  - Set the power level of the stimulus signal to a value that does not generate compression.
  - Narrow the IF bandwidth.
  - Increase the averaging factor.
  - Use a cable whose change in amplitude/phase characteristics when bent is small.
  - Use high-precision connectors.

The types of thru measurement you can omit are determined by the ports you select. Some types of thru measurement cannot be omitted. The table below shows the types of thru measurement that can be omitted.

Omissible Thru Type at Full 3-port Calibration

Ports used	Types of thru measurement that can be omitted
Ports 1, 2, 3	2-3
Ports 1, 2, 4	2-4
Ports 1, 3, 4	1-4
Ports 2, 3, 4	2-4

For the simplified full 3-port calibration, (Optional) is displayed on the softkey of the type of thru measurement that can be omitted. For example, when the omissible thru measurement is 2-3, **2-3 (Optional)** is displayed. The display after the execution of the omissible thru measurement is the same as that for a required thru measurement.

Omissible Thru Type at Full 4-port Calibration

Required thru measurements	Types of thru measurement that can be omitted
1-2, 1-3, 3-4	1-4, 2-3, 2-4

For the simplified full 4-port calibration, (Recommended) is displayed on the softkey of the type of thru measurement that can be omitted. For example, when the omissible thru measurement is 2-3, **2-3 (Recommended)** is displayed. The display after the execution of the omissible thru measurement is the same as that for a required thru measurement, and the display for the remaining omissible thru measurement is **2-3 (Optional)**.

For the simplified full 4-port calibration, up to three thru measurements can be omitted, but it is recommended to perform two or more types of calibration because omitting all of them results in poor accuracy.

Simplified 3/4-port TRL Calibration

In the simplified 3/4-port TRL calibration, because the calibration coefficients are calculated while omitting part of the thru/line measurement data and line/match measurement data, the effect of errors when acquiring calibration data becomes larger than in the normal 3/4-port TRL calibration. Therefore, you must give more attention to data measurement for the simplified full 3/4-port calibration than for the ordinary 3/4-port TRL calibration.

The types of thru/line measurement and line/match measurement you can omit are determined by the ports you select. Some types of measurement cannot be omitted. You cannot omit thru/line measurement only or line/match measurement only, even for an omissible measurement path.

Omissible Thru type at 3-port TRL calibration

Ports used	Omissible measurement
Ports 1, 2, 3	2-3
Ports 1, 2, 4	2-4
Ports 1, 3, 4	1-4
Ports 2, 3, 4	2-4

For the simplified 3-port TRL calibration, (Optional) is displayed on the softkey of the omissible thru/line measurement. For example, when the omissible thru/line measurement is 2-3, **2-3 (Optional)** is displayed. The display after the execution of the omissible thru/line measurement is **2-3 Thru/Line**.

For the softkey of an omissible line/match measurement, (Optional) is displayed as in the case of the omissible thru/line measurement. For example, when the omissible line/match measurement is 2-3, **2-3 (Optional)** is displayed. The display after the execution of the omissible line/match measurement is **2-3 Line/Match**.

The softkeys (Fwd and Rvs) for forward measurement and reverse measurement for the omissible line/match measurement do not change.

Omissible measurements at 4-port TRL calibration

Required measurements	Omissible measurements
1-2, 1-3, 3-4	1-4, 2-3, 2-4

For the simplified 4-port TRL calibration, (Recommended) is displayed on the softkey of the omissible thru/line measurement. For example, when the omissible thru/line measurement is 2-3, **2-3 (Recommended)** is displayed. The display after the execution of the omissible thru/line measurement is **2-3 Thru/Line**, and the display for the remaining omissible thru/line measurement is **2-3 (Optional)**.

For the softkey of an omissible line/match measurement, (Recommended) is displayed as in the case of the omissible thru/line measurement. For example, when the omissible line/match measurement is 2-3, **2-3**

(Recommended) is displayed. The display after the execution of the omissible line/match measurement is **2-3 Line/Match**, and the display for the remaining omissible line/match measurement is **2-3 (Optional)**.

The softkeys (Fwd and Rvs) for forward measurement and reverse measurement for the omissible line/match measurement do not change.

For the simplified 4-port TRL calibration, up to three thru/line measurements and line/match measurements can be omitted, but omitting all of them results in poor accuracy.

Adapter Removal-Insertion

- About Adapter Removal
- About Adapter Insertion
- About Adapter Waveguide & Rotate
- Performing Adapter Waveguide Calibration
- Procedure for Adapter Removal/Insertion
- Difference between Traditional Network Analyzer & E5071C Adapter Calibration

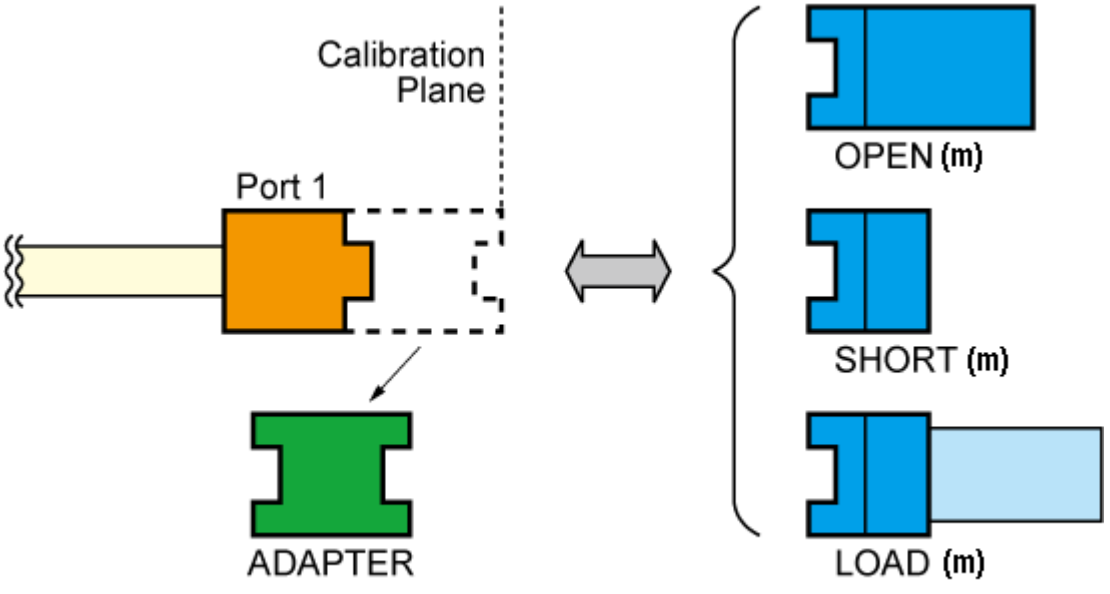
Other topics about Advanced Calibration

About Adapter Removal

Adapter Removal is a technique used to remove any adapter characteristics from the calibration plane. The E5071C uses the following adapter removal process to remove adapter characteristics:

1. Perform calibration with the adapter in use.
2. Remove the adapter from the port and measure Open, Short, and Load values to determine the adapter's characteristics.
3. Remove the obtained adapter characteristics from the error coefficients in a de-embedding fashion.

Open, Short, and Load values measured with the adapter removed



e5071c203

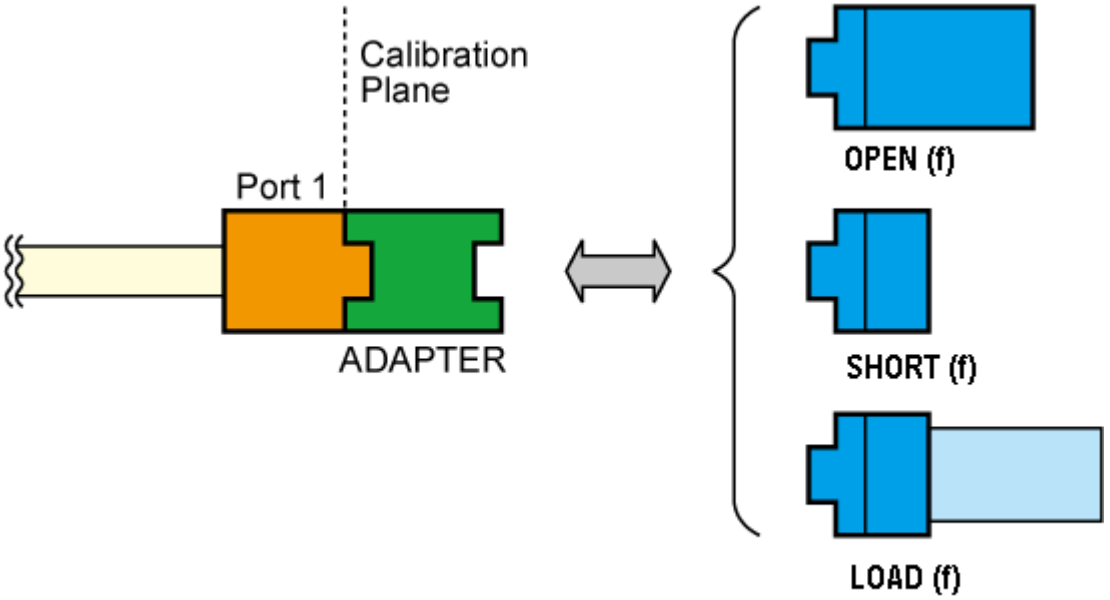
- The adapter removal and partial overwrite function is only available when the calibration status is [Corr] and not for [C?] or [C!].

About Adapter Insertion

This above described method also makes it possible to add adapter characteristics to a port with n-port full calibration. This allows you to make a measurement with the adapter. E5071C uses the following adapter insertion process to insert adapter characteristics:

1. Perform calibration without adapter in use.
2. Insert the adapter to the port and measure Open, Short, and Load values to determine the adapter's characteristics.
3. Insert the obtained adapter characteristics to the error coefficients in an embedding fashion.

Open, Short, and Load values measured with adapter attached



e5071c204

In order to determine the adapter characteristics (with four unknown parameters) by making three measurements (Open, Short, and Load), the adapter must satisfy the following requirements:

1. Adapter must be Reciprocal (with S21 and S12 equal) in nature. It should have a consistent behavior, independent of the direction from which it is used.
2. The electrical length of Adapter should be known at an accuracy of ±1 /4 of wavelength.

About Adapter Waveguide & Rotate

From firmware A.09.10, functionality related to adapter waveguide and adapter rotation is added. Adapter waveguide length and Cut off frequency can be set using the following commands:

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff

Coaxial length can be set using **Cal** > **Calibrate** > **Adapter Removal** > **Coaxial Length**

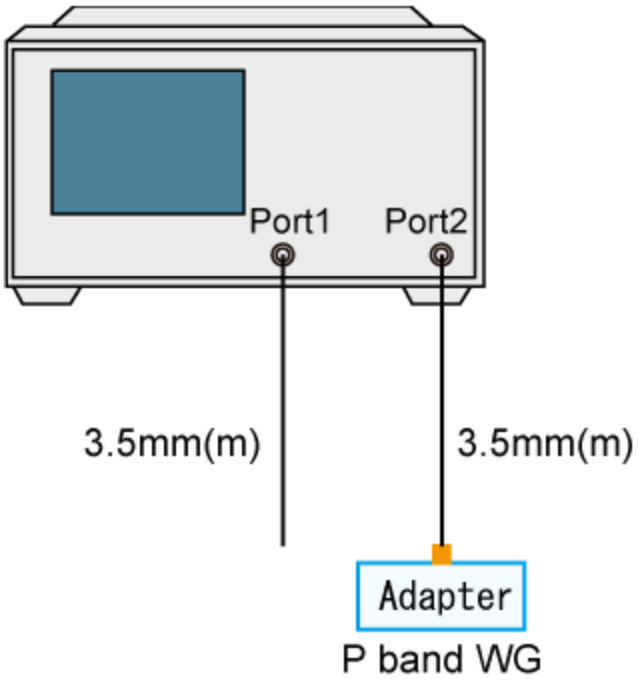
Rotate command executes Adapter Removal/Insertion along with moving the phase of adapter (which is removed or inserted) to 180 degrees. This command is useful in cases where auto judgement of phase fails. This command can be executed several times while the calibration remains valid.

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

- System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.
- Some calibration kits such as the waveguide calibration kit have operational frequency range defined by Minimum and Maximum frequency. When the E5071C stimulus setting is out of the operational frequency range, you cannot click **Done** key or finish the calibration by remote control. In this case, use a calibration kit that has proper frequency range, or change the E5071C stimulus setting to proper range that the calibration kit can cover. Refer to the calibration kit manual or definition in the E5071C with **Cal** > **Modify Cal Kit** > **Define STDs** for the Maximum & Minimum frequency of the calibration kit.

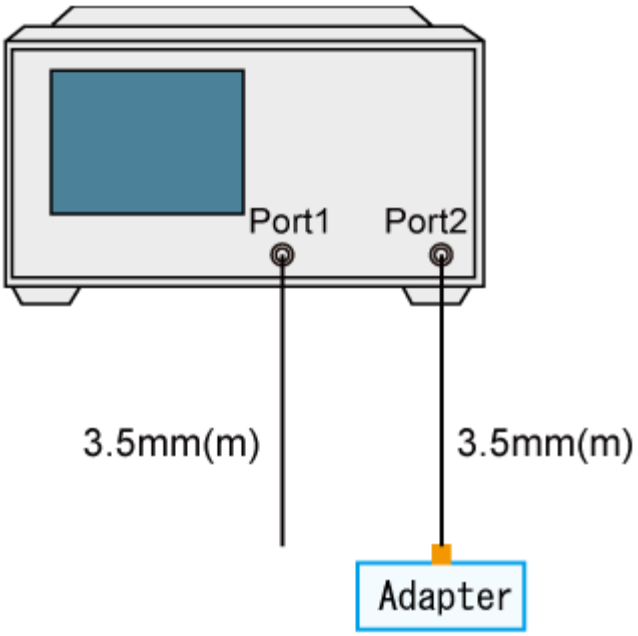
Performing Adapter Waveguide Calibration

This section provides an example of performing adapter waveguide calibration to perform a Full 2-port calibration plane between Port 1 (3.5mm) & Port 2 (P-band wave guide).



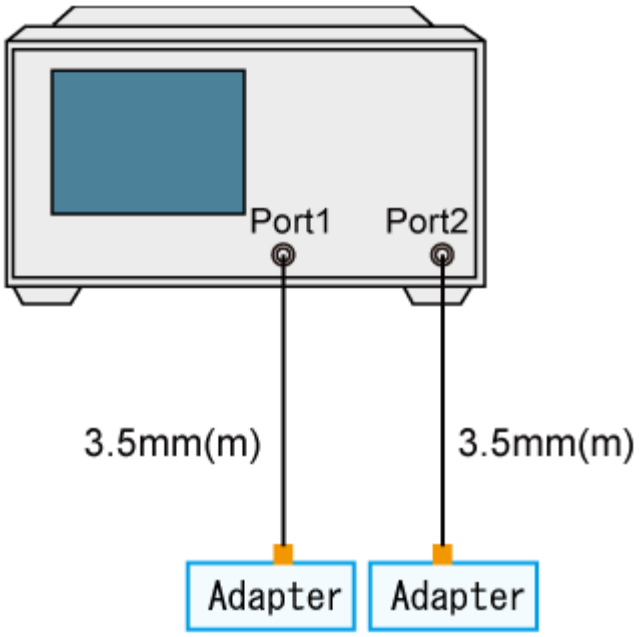
e5071c488

1. Using Adapter Insertion:
    a. Perform a SOLT F2P Calibration at Port 1 (3.5mm) using Z0=50 ohm
    b. Perform Adapter insertion at Port 2 (Z0=1) using P11644A



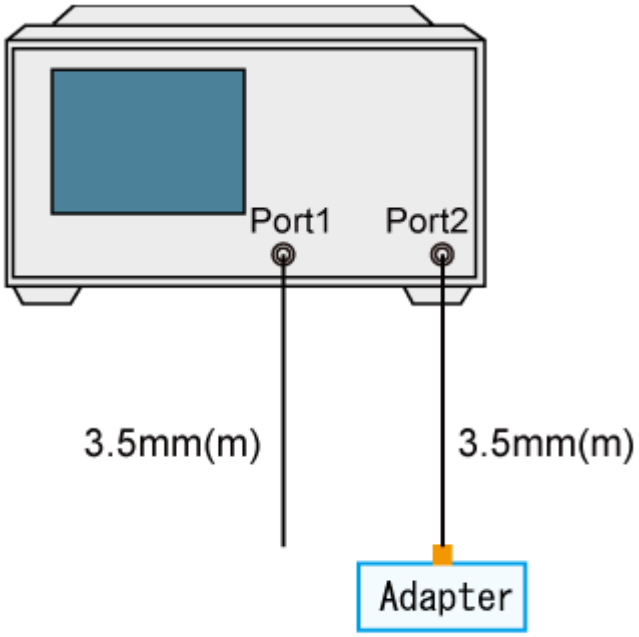
e5071c489

3. Using Adaptor Removal:
    a. Perform a SOLT/TRL F2P Calibration at P band WG (Z0 = 1)
    b. Perform Adapter removal at port 1 (Z0=50) using 85052D



e5071c490

4. Using Port Extension:
    a. Perform a SOLT F2P calibration at Port 1 (3.5mm) with Z0 = 50
    b. Perform Port extension calibration at Port 2 (manual with Coax length, WG length, Cut off). System Z0 can be either 1 oh or 50 ohm; both are fine to use.



e5071c491

Procedure for Adapter Removal/Insertion

The S parameter of a reciprocal adapter can be determined when the following data is available:

1. Open, Short, and Load measurements.
2. Actual values derived from the **CalKit** definitions.
3. An approximate length of the adapter.
4. Nature of the intended operation: removal or insertion.
    - If the frequency setting is not Zero Span, an approximate length of the adapter should be provided otherwise an error might be generated in the measurement.
    - For Adapter Removal, the adapter length is (**+**) Positive. For Adapter Insertion, the Adapter length is (-) Negative.
    - To change gender of the calibration kit, refer to Defining Calibration Kit Gender.

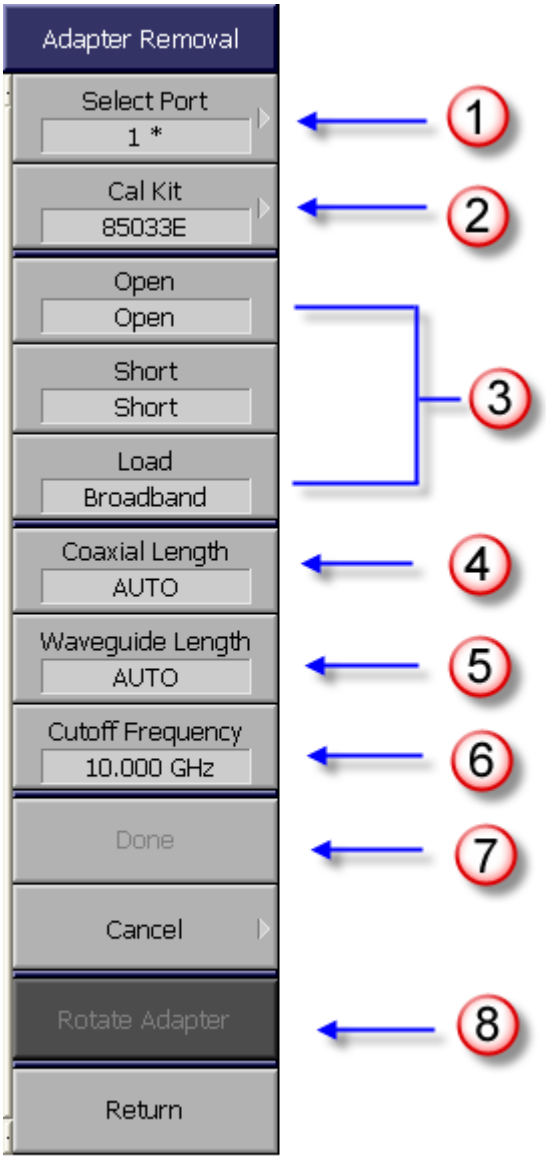
To use Adapter Removal/Insertion, follow the given procedure:

1. Perform a full n-Port calibration using your calibration kit so that the port used to conduct adapter insertion/removal is calibrated.
2. When you need to remove an adapter from the calibration plane (adapter removal) to connect your DUT, perform the calibration with an adapter so that you can make a calibration with your calibration kit. When you need to add an adapter into the

calibration plane (adapter insertion) to connect your DUT, perform the calibration without an adapter.

2. Open the Adapter Removal menu:

   Cal > **Calibrate** > **Adapter Removal**



e5071c194

3. Select the Port in/from which you want to insert/remove Adapter characteristics from **Select Port** (e.g. 1) (**1** in the above figure). A * sign appears in front of the port is a valid port to conduct adapter removal/insertion as the full n-port calibration has been performed on the port.

4. Select the proper standard Calkit you need to use from **CalKit** (e.g. 85033E) (**2** in the above figure) to characterize the adapter. The calibration kit is used at the plane from which adapter is removed (adapter removal), or the plane in which adapter is inserted (adapter insertion).
5. If you are using a Coaxial adapter, select the **Coaxial Length** as **Auto** for Automatic Insertion/Add, or if you want to input a manual Coaxial Length, type a Coaxial length with Negative or Positive sign (**4** in the above figure). For Adapter Removal, the Coaxial length is (+) Positive and for Adapter Insertion, the Coaxial length is (-) Negative.
6. If you are using a Waveguide or a mixed adapter, select the **Cutoff Frequency** of the waveguide you want to enter (**6** in the above figure). The Cutoff Frequency is defined by the waveguide calibration kit in use.
    7. Without entering the value of Cutoff Frequency, the Waveguide Length cannot be entered.
7. If you are using a Waveguide or a mixed adapter, select the **Waveguide Length** as **Auto** for Automatic Insertion/Add, or if you want to input a manual Waveguide Length, type a value for Waveguide length with Negative or Positive sign (**5** in the above figure). For Adapter Removal, the Waveguide length is (**+**) Positive and for Adapter Insertion, the Waveguide length is (-) Negative.
    8. Change the system Z0 to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.
8. Connect Open, Short, and Load of the selected **Calkit** (e.g. 85033E)with the selected port respectively and click/press **Open**, **Short**, and **Load** respectively. ENA measures the cal kit standard, calculates the adapter characteristics, and then conducts adapter removal/insertion.
    9. A checkmark appears in (**Open**, **Short**, and **Load**) menu after each type of calibration is done, respectively.
9. Click **Done** to complete the process (**7** in the above figure).
    10. Some calibration kits such as the waveguide calibration kit have operational frequency range defined by Minimum and Maximum frequency. When the E5071C stimulus setting is out of the operational frequency range, you cannot click **Done** or finish the calibration by remote control. In this case, use a calibration kit that has proper frequency range, or change the E5071C stimulus setting to proper range that the calibration kit can cover. Refer to the calibration kit manual or definition in the

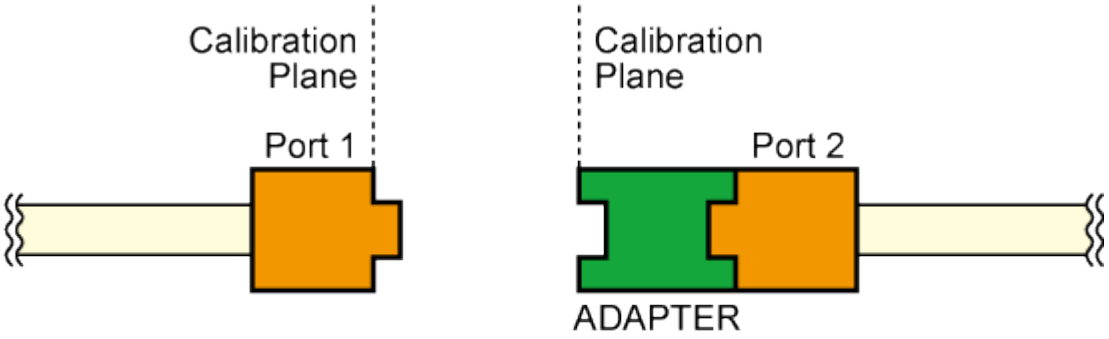
E5071C with **Cal** > **Modify Cal Kit** > **Define STDs** for the Maximum & Minimum frequency of the calibration kit.

11. In cases where auto judgement of phase fails, select **Rotate Adapter** to move the phase of adapter (which is removed or inserted) to 180 degrees.

Difference between Traditional Network Analyzer & E5071C Adapter Calibration

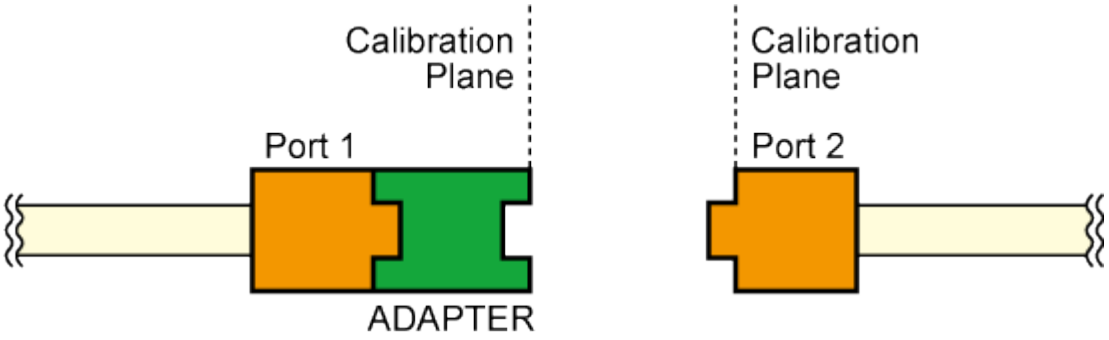
Usually, two-port network analyzers removes adapter characteristics by performing two sets of Full 2 Port Calibration as shown below:

Calibration performed with the adapter connected to Port2



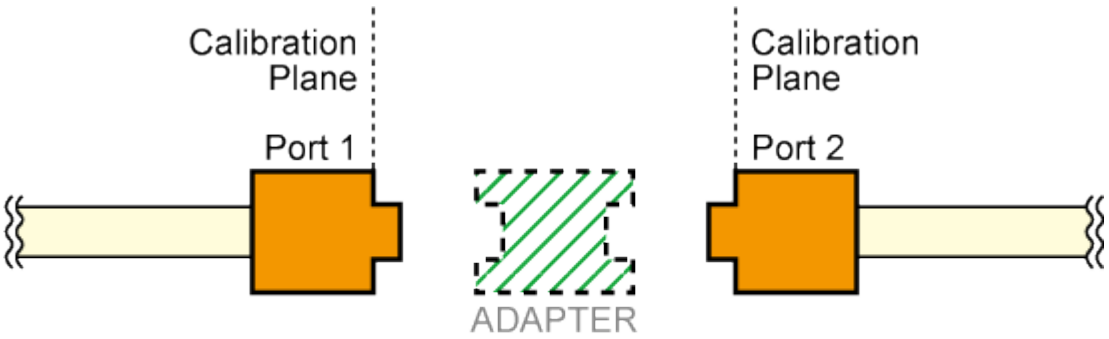
e5071c200

Calibration performed with the adapter connected to Port1



e5071c201

Removing adapter characteristics using two (above) sets of calibration



e5071c202

However, this method is not suitable for a multi-port network analyzer because it will require as many sets of Full 2 Port Calibration as twice the number of port combinations. Therefore, the E5071C uses an advanced method to remove Adapter characteristics described in About Adapter Insertion.

Unknown Thru Calibration

- Overview
- Procedure

Other topics about Advanced Calibration

Overview

Unknown Thru Cal is the preferred THRU method of calibrating the ENA to measure a non-insertable device. The major benefits of using a unknown Thru calibration are:

- It is easy to perform.
- Provides better accuracy than Defined Thru and is usually better than Adapter Removal.
- Does not rely on the existing standard definitions that may no longer be accurate.
- Causes minimal cable movement if the THRU standard has the same footprint as the DUT. In fact, the DUT can often be treated as the THRU standard.

The Thru must satisfy the following requirements:

1. Thru must be Reciprocal (with S21 and S12 equal) in nature. It should have a consistent behavior, independent of the direction from which it is used.
2. The electrical length of the Thru should be known at an accuracy of ±1 /4 of the wavelength.
3. Unknown Thru is NOT supported during the TRL calibration.

Procedure

To measure the nominal electrical delay of the Unknown Thru standard:

1. With port 1 configured, perform a simple S11 response calibration at port 1 by using a short circuit

   Press **Cal** > **Calibrate** > **Response (Short)** > **Short**
2. Connect the Unknown Thru standard to port 1 and terminate the standard with a short circuit.
3. Measure the electrical delay of the Unknown Thru standard

   Press **Cal** > **Port Extensions** > **Auto Port Extension** > **Measure SHORT** > **Port 1**

The electrical delay is automatically measured and can be confirmed.

Press **Cal** > **Port Extensions** > **Extension Port 1** {Value}

4. If there is an offset in the short circuit used to terminate the Unknown Thru standard, remember to subtract its electrical length from the measurement result.

To perform the unknown Thru Calibration, follow the procedure below:

1. Press **Cal** key, then click **Calkit.**
2. Select the desired calibration kit (e.g. 85033D or User Kit)
3. Press **Cal** > **Modify Calkit** > **Define STDs** > **No Name** > **Label**
4. Type a name for unknown Thru Cal standard e.g. {Unknown Thru1}
    - By changing the label to a different name (e.g. Unknown Thru1), the 'No Name' standard will adopt to the specified new name.
5. Press **Cal** > **Modify Calkit** > **Define STDs** > **{Defined Name}** > **STD Type** > **Unknown Thru**
6. Click **Offset Delay**, then enter the approximate offset delay value of the unknown Thru standard.
7. Press **Cal** > **Modify Calkit** > **Specify CLSs** > **Thru** > **Set All** > **{Defined Name}**
6.
    1. Different Unknown Thru can be used for different ports.
7. Press **Cal** > **Calibrate** > **n-Port Cal** (n>1) > **Transmission** > **{Defined Name}** to execute mechanical Unknown Thru calibration.
8. Complete Reflection, and Isolation (optional) calibration for n full port Cal.

Performing 8 term Calibration using External PC

- Overview
- Procedure
- Example

Other topics about Advanced Calibration

Overview

To perform 8-term error model calibration using an external PC (measuring raw measurement data using the E5071C, calculating error term using external PC and then performing correction with calculated error terms and measured raw data), the procedure described in this section must be followed. When you use this method, the measurement specification is not satisfied.

- In most cases, users do NOT need to follow the below procedure as these considerations have already been performed on 8-term error model calibration in E5071C.
- When users perform 12-term error model calibration using external computer such as SOLT calibration, users do NOT need to take following procedure.

Procedure

Turning off the "System Calibration"

In its default setting, the E5071C applies "system calibration" to the measured S-parameter data. Because the "system calibration" is applied only to the S-parameter measurement data and NOT to the absolute value measurement data, it interferes with 8-term error model calibration when calculating error coefficients on external computer. You need to turn OFF this state, all the time for calibration and measurement.

The system calibration can be turned off by executing SCPI.SYSTem.CORRection.STATe

- Once the E5071C receives this command, it will keep this setting until preset. This state is saved in the state file.

Turning off the "Virtual Bridge" (3 GHz, 4.5 GHz, 6.5 GHz and 8.5 GHz option only)

There is a software conversion function called "virtual bridge" on E5071C. The default setting of this state is ON, and this state interferes with 8-term error model calibration when calculating error coefficients on external computer. You need to turn OFF this state, all the time for the calibration and the measurement.

- This function only exist on frequency 3 GHz, 4.5 GHz, 6.5 GHz and 8.5 GHz options. It's not necessary to consider "virtual bridge" on E5070A/71A/70B/71B, and E5071C frequency 14 GHz and 20 GHz options.

The virtual bridge can be turned off by the command: SERVice:ADJust:VIRTual:BRIDge[:STATe] {ON|OFF|1|0}.

- Since this command is under service menu, the "SVC" indicator is turned on when the ENA sets virtual bridge OFF.
- Once the ENA receives this command, it will keep this setting until preset.
- The state of "virtual bridge" is NOT saved in the state file, and the default state of the "virtual bridge" is ON. So you need to send the command to turn OFF the state, every time you recall the state file.

Set the "RF ranging mode" to MANual (3 GHz, 4.5 GHz, 6.5 GHz and 8.5 GHz option only)

There is a hardware function called "RF ranging mode" on E5071C. The default setting of this state is AUTO, and this state interferes with 8-term error model calibration when calculating error coefficients on external computer. You need to turn off the "RF ranging Auto mode".

The "RF ranging mode" can be changed by the command: SERVice:ADJust:SENSe[Ch]:SWEep:RF:RANGe:PORT[Pt]:AUTO {ON|OFF}. This command applies to each channel and measurement port.

- This function only exist on frequency 3 GHz, 4.5 GHz, 6.5 GHz and 8.5 GHz options. It's not necessary to consider "virtual bridge" on E5070A/71A/70B/71B, and E5071C frequency 14 GHz and 20 GHz options.
- Since this command is under service menu, the "SVC" indicator is turned on when the ENA sets RF range auto mode OFF. Once the ENA receives this command, it will keep this setting until preset.
- The state of "RF ranging mode" is NOT saved in the state file, and the default state of the "RF ranging mode" is AUTO. So you need to send the command to set state to AUTO OFF, every time you recall the state file.

You need to keep the state of "RF range Auto Mode" as OFF, all the time for calibration and measurement using this method

- Once the ENA receives above command, it will keep this setting until preset.

- Since the "RF ranging" is hold during the measurement, the system dynamic range of the E5071C will degrade by 6dB, as a result of the 8-term calibration performed externally. However, there is NO dynamic range degrade by 8-term error model calibration with internal calibration of the ENA.

Example

Assume you perform 4-port TRL calibration on channel1, you need to send following command before calibration. Also, you need to keep this state all the time for calibration and measurement.

```
SYSTem:CORRection:STATe OFF  'system correction off
SERVice:ADJust:VIRTual:BRIDge:STATe OFF  'virtual bridge off
SERVice:ADJust:SENSe1:SWEep:RF:RANGe:PORT1:AUTO OFF  'RF ranging mode to manual on port1
SERVice:ADJust:SENSe1:SWEep:RF:RANGe:PORT2:AUTO OFF  'RF ranging mode to manual on port2
SERVice:ADJust:SENSe1:SWEep:RF:RANGe:PORT3:AUTO OFF  'RF ranging mode to manual on port3
SERVice:ADJust:SENSe1:SWEep:RF:RANGe:PORT4:AUTO OFF  'RF ranging mode to manual on port4
```

- Once you preset the system, you need to resend all of the above commands.
- Once you save and recall the state using this method, you need to resend all of the above commands except for system correction off.

Mixer Calibration

Scalar-Mixer Calibration

- Overview
- Confirming Calibration Status
- Procedure Using Mechanical Calibration Kit
- Procedure Using ECal Kit

Other topics about Mixer Calibration

Overview

The E5071C has a scalar-mixer calibration function for measuring frequency conversion devices.

Scalar-mixer calibration allows you to measure the magnitude value and reflection parameter of the mixer's conversion loss with very high accuracy by performing calibration using calibration standards (OPEN/SHORT/LOAD) in combination as well as a power meter.

For measurement of the conversion loss in a frequency conversion device, normal full 2-port calibration is not available because of the frequency difference between the stimulus port and the response port. Therefore, scalar-mixer calibration allows you to correct the error term that resides in a full 2-port error model by using the error model and an expression based on a new concept.

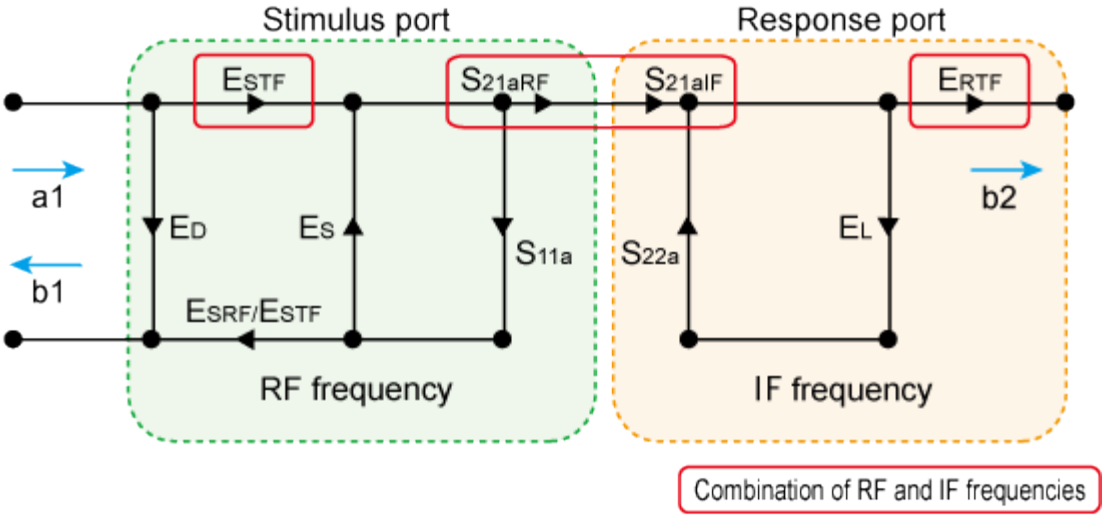
You can correct the following error elements by using the scalar-mixer calibration.

- The reflection between the output port of the network analyzer and the input port of the measured mixer (vector error correction).
- The reflection between the output port of the measured mixer and the input port of the network analyzer (vector error correction).
- Transmission frequency characteristics at different frequencies (scalar error correction).

The frequency-offset error model can be described by the flow graph in the figure below. From the flow graph it can be seen that the model is divided in two halves: the stimulus port and the response port. By conceptualizing the error model in these two halves, each error term can be isolated to either of the model's halves depending on the frequency at which it was generated. The majority of the signals will only affect measurements by causing errors at the same frequency.

Two error signals that are functions of both halves of the frequency-offset model are isolation (EXF) and transmission tracking (ETF). EXF is set to zero because it is cannot be detected by the frequency change. ETF must

be divided into two types of errors: one associated with the stimulus side at the input frequencies and the other associated with the response side at the output frequencies. As previously discussed, calculating the transmission tracking (ETF) based on both stimulus and response sides is the key to using error correction during frequency-offset measurements.



ED Directivity
ES Source match
EL Load match
ETF Transmission tracking = (ESTF x ERTF)
ESTF Transmission tracking source
ERTF Transmission tracking receiver

ERF Reflection tracking = (ESTF x ESRF)
ESRF Reflection tracking source
C21a Conversion loss actual = (S21aRFF x S21aIF)
S21aRF Conversion at RF Freq
S21aIF Conversion at IF Freq

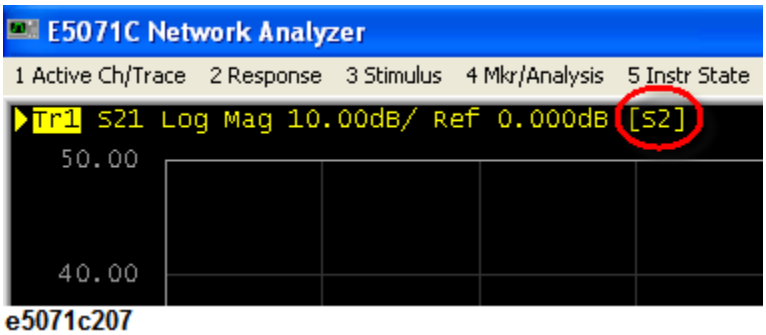
e5071c457

In scalar-mixer calibration, use the power meter to measure the transmission tracking error (ESTF) of the signal source in both the output and input frequencies in order to calculate the transmission tracking in the frequency-offset (ETF). The transmission tracking of the receiver in the output frequency (ERTF) can be determined by dividing the transmission tracking (ETF) calculated based on the output frequency in full 2-port calibration by the receiver's transmission tracking (ESTF). Multiplying the signal source transmission tracking (ESTF) by the receiver transmission tracking (ERTF) provides the transmission tracking in the frequency-offset (ETF). Since ESTF and ERTF are products of two power measurements, the resulting correction coefficient is a scalar parameter.

Confirming Calibration Status

Error correction status of each channel

You can confirm the progress of error correction for each channel by viewing the error calibration status. For the traces in which the scalar-mixer is valid, the status [S2] is added.

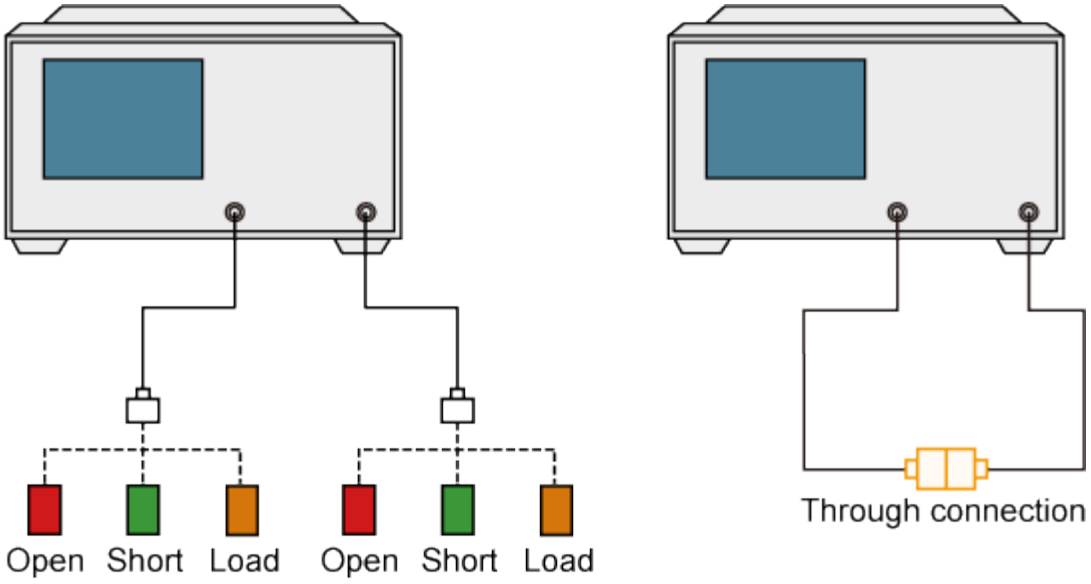


e5071c207

- In scalar-mixer calibration, the normal calibration coefficient is invalid while the frequency-offset sweep is in progress; in this case the scalar-mixer calibration coefficient is used instead. Turning the frequency-offset sweep off switches over to the normal calibration coefficient; however, the information on the scalar-mixer calibration coefficient is retained.

Procedure Using Mechanical Calibration Kit

1. Before starting scalar-mixer calibration, verify that the frequency-offset is activated by clicking **Sweep Setup** key > **Frequency Offset**. If the frequency-offset is not valid, you cannot start scalar-mixer calibration.
2. Press **Channel Next**/**Channel Prev** keys to select the channel you want to calibrate.
3. Press **Cal** key.
4. Click **Mixer/Converter Calibration** > **Scalar Cal (Manual)** > **Select Ports**.
5. Select the test port for starting scalar-mixer calibration.



e5071c330

6. For the scalar-mixer calibration, only one direction with 2-ports calibration is available. "One direction" means the forward or reverse direction. Two directions may be used among the same ports, but this would not be full 2-port scalar-mixer calibration but simply the simultaneous operation of single-direction scalar-mixer calibrations.. .

Reflection

Both **x** and **y** in the following steps are provided for the case where you have selected in the test port selection.

6. Click **Reflection**.
7. Connect the OPEN calibration standard to the test port **x** (connector for the DUT).
8. Click **Port x @Freq x Open** to start measurement of the calibration standard.
9. Click **Port x @Freq y Open** to start measurement of the calibration standard.
10. Disconnect the OPEN calibration standard, then connect SHORT calibration standard in its place.

11. Click **Port x@Freq x Short** to start measurement of the calibration standard.
12. Click **Port x@Freq y Short** to start measurement of the calibration standard.
13. Disconnect the SHORT calibration standard, then connect LOAD calibration standard in its place.
14. Click **Port x@Freq x Load** to start measurement of the calibration standard.
15. Click **Port x@Freq y Load** to start measurement of the calibration standard.
16. Repeat the procedure for port y.
17. Click **Return**.

Transmission

18. Click **Transmission**.
19. Connect a THRU between the test ports x and y (between the connectors for the DUT).
20. Click **Port y-x@Freq y Thru** to start measurement of the calibration standard.
21. Click **Port x-y@Freq x Thru** to start measurement of the calibration standard.
22. Click **Return**.

Power Meter

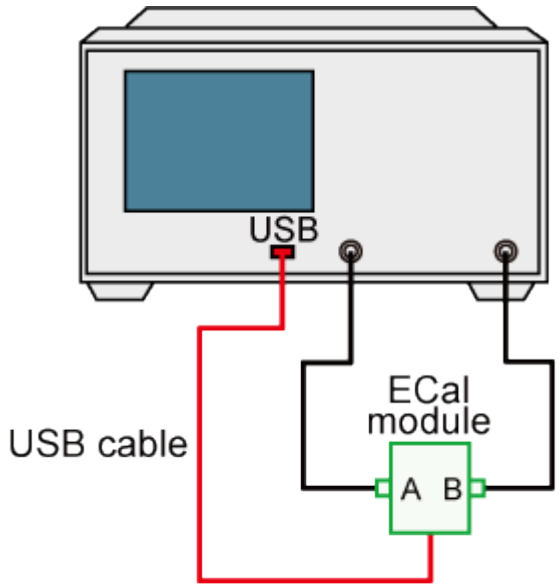
23. Click **Power Meter**. See Preparing to control the power meter for setup the power meter.
24. Click **Use Sensor**. Each time the key is pressed, Channels A and B switch over alternately. If you are using a one-channel power meter, select Channel A.
25. Connect the power sensor for the selected channel to the selected port.
26. Click **Port x@Freq x**.
27. Click **Port x@Freq y**.
28. Click **Port y@Freq x**.
29. Click **Port y@Freq y**.
30. Click **Return**.

Error Correction Done

31. Click **Done** to exit the scalar-mixer calibration. This step allows the calibration coefficient to be calculated, turning on the error correction function automatically.

Procedure Using ECal Kit

1. Before starting scalar-mixer calibration, verify that the frequency-offset is activated by clicking **Sweep Setup** key > **Frequency Offset**. If the frequency-offset is not valid, you cannot start scalar-mixer calibration.
2. Press **Channel Next**/**Channel Prev** keys to select the channel you want to calibrate.
3. Press **Cal** key.
4. Click **Mixer/Converter Calibration** > **Scalar Cal (ECal)** > **Select Ports**.
5. Select the test port for starting scalar-mixer calibration.
6. Click **Power Meter**. See Preparing to control the power meter for setup the power meter.
7. Click **Use Sensor**. Each time the key is pressed, Channels A and B switch over alternately. If you are using a one-channel power meter, select Channel A.
8. Connect the power sensor for the selected channel to the selected port.
9. Click **Port x@Freq x**.
10. Click **Port x@Freq y**.
11. Click **Port y@Freq x**.
12. Click **Port y@Freq y**.
13. Click **Return**.
14. Connect the test port you selected to the ECal module.



e5071c322

15. Click **ECal & Done**. The calibration coefficient will be calculated, turning on the error correction function automatically.
    - If the ECal module is not connected to the E5071C or power calibration have not been done yet, the **ECal & Done** menu item will not be not available. In addition, if the ECal module is not connected to the test port of the calibration target, an error may occur.

Vector-Mixer Calibration

- Overview
- Characterizing Calibration Mixer (with IF filter)
- Executing Characterization of Calibration Mixer
- Characterizing Calibration Mixer (with IF filter) for Balance Mixer Measurement
- Executing Characterization

Other topics about Mixer Calibration

Overview of Vector-Mixer Calibration

The E5071C has a vector-mixer calibration function for use in measuring frequency conversion devices.

The vector-mixer calibration allows you to measure the magnitude, phase and group delay of the mixer's conversion loss by using in combination calibration standards (OPEN/SHORT/LOAD) and calibration mixer with an IF filter, as well as the network de-embedding function incorporated in the E5071C.

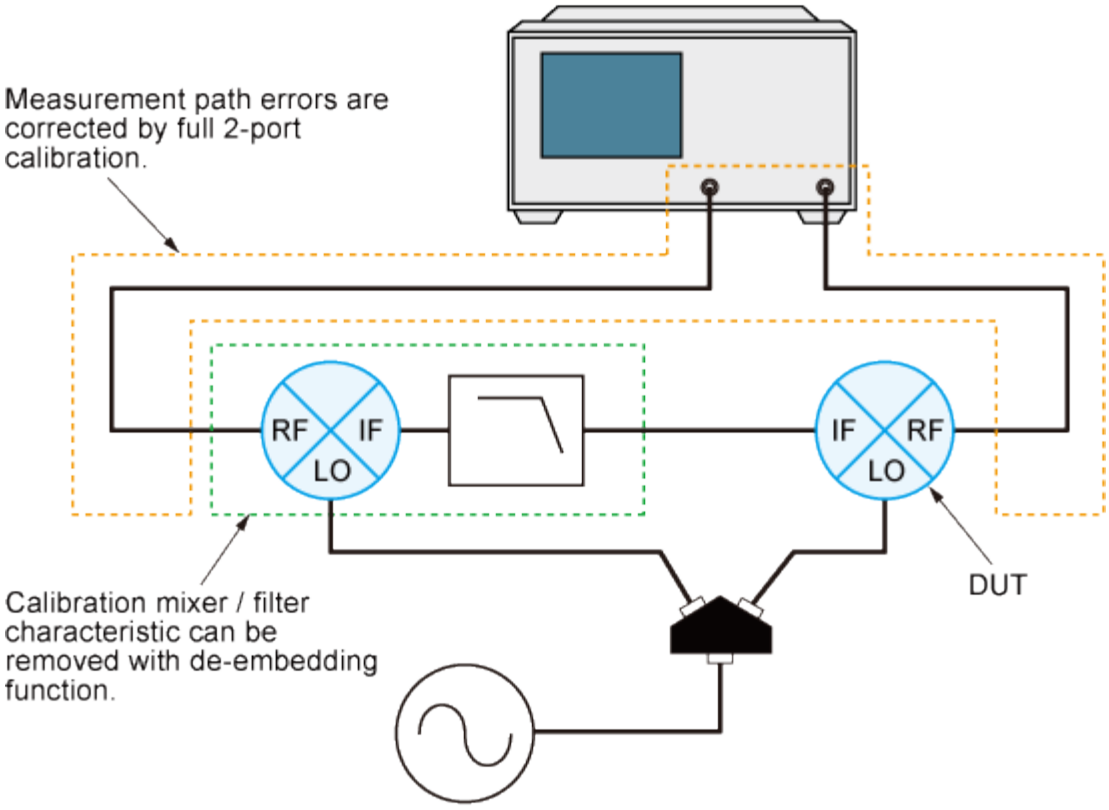
- For Fixed RF measurement (RF: Fixed, LO and IF: Swept), it is NOT possible to perform vector-mixer calibration because touchstone file, which defined with fixed frequency, can not be imported to ENA.
- For Fixed IF measurement (RF and LO: Swept, IF: Fixed), it is possible to perform vector-mixer calibration and to measure conversion loss and return loss, but NOT possible to measure group delay because IF frequency is fixed.
- For Swept IF measurement (RF and IF: Swept, LO: Fixed), it is possible to perform vector-mixer calibration and to measure conversion loss, return loss and group delay.

You can also perform balanced mixer measurements by using two calibration mixers that each has an IF filter.

Vector-mixer calibration is implemented by eliminating the characteristics of the calibration mixer and IF filter by using the network de-embedding function after full 2-port calibration has been completed. Using the up/down conversion method allows you to specify the same sweep measurement frequency for the input and output ports, thus enabling full 2-port calibration at the end of the target port. Consequently, only the characteristics of the measured mixer (DUT) can be obtained by using the network de-embedding function, after eliminating the characteristics of the calibration mixer with an IF filter from all measurement results.

- Since the up/down conversion method is used in vector-mixer calibration, the frequency-offset function is not used. But, option 008 is required to use the VMC macro, SG control, and etc.

Overview of Vector-Mixer Calibration



e5071c340

The vector-mixer calibration requires the characteristics data for the calibration mixer with IF filter.

Measured mixer

A measured mixer (DUT) signifies an unknown target mixer of measurement. However, a measured mixer meeting the requirements for a calibration mixer can be used as a calibration mixer.

Calibration mixer (with IF filter)

The calibration mixer is required for supporting the measurement system of the up/down conversion. You must also evaluate in advance the frequency response characteristics of the calibration mixer. The vector-mixer calibration method obtains the characteristics of the measured mixer alone by using the network de-embedding function to eliminate the

characteristics of the calibration mixer from the measurement result. You can use the IF filter to select any required frequency conversion component such as RF+LO, RF-LO, and LO-RF. The calibration mixer and IF filter can be seen as a part of the test system setup, just like the network analyzer and the test cable; they are connected at the same location during the entire calibration or measurement.

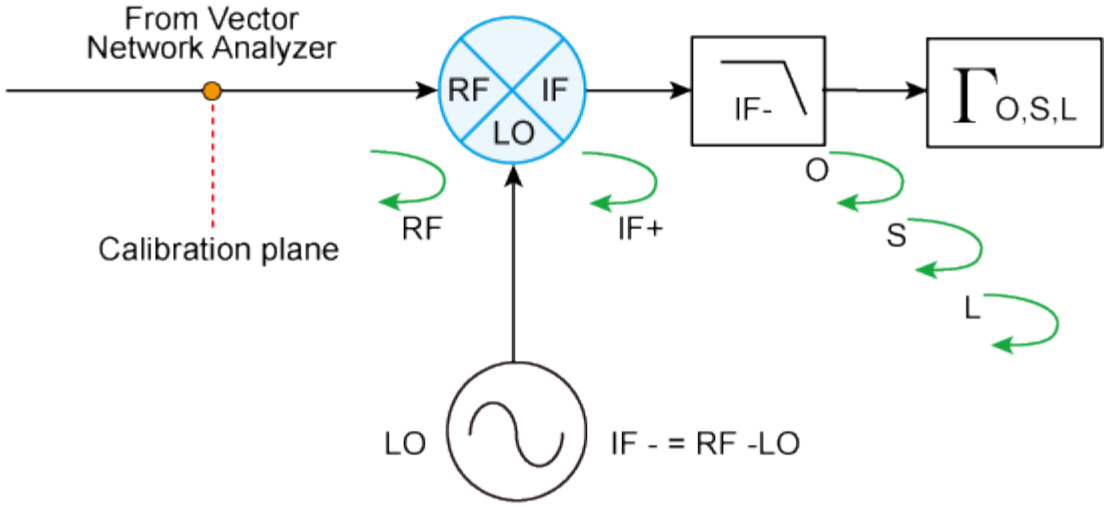
- The frequency range must be equal to or greater than that of the measured mixer. If you want to test multiple mixers with a single setup, select a wide range of calibration mixers that can cover all frequencies of the target test devices.

Characterizing Calibration Mixer (with IF filter)

In vector-mixer calibration, you must characterize the calibration mixer with the IF filter. As shown in the following figure, connect the target mixer (with IF filter) to the port of the network analyzer on which vector calibration has been performed and then connect an OPEN, SHORT or LOAD standard to the end of the IF filter to start reflection measurement. The signals measured at the test port include the reflection signal from the mixer's RF port, the IF signal (IF+) converted by the mixer and then reflected by the IF filter, and the IF signal (IF-) passing through the IF filer and then reflected by the calibration standard.

The characteristics of the calibration mixer can be described in a 1-port error model, and each error item can be determined from Γ_O, Γ_S, and Γ_L, which are obtained in the reflection measurement of individual standards.

Characteristics evaluation of calibration mixer (with IF filter)



e5071c458

- The calibration mixer must be reciprocal. The term "reciprocal" means the magnitude and phase of the conversion loss are equal both in the forward and reverse directions. The forward conversion loss occurs during the measurement of the output signal at the IF port while inputting measurement signals into the RF port. In contrast, the reverse conversion loss occurs during measurement of the output signal at the RF port while inputting measurement signals into the IF port.
- For precise calibration, the conversion loss in each direction must be less than 10 dB using a calibration mixer and IF filter in combination. Exceeding 15 dB of the conversion loss in any direction may deteriorate the calibration accuracy significantly.

Characterizing procedure for calibration mixer (with IF filter)

The E5071C has a pre-installed VBA macro (Vector Mixer Characterization) for characterizing the calibration mixer with an IF filter.

Storage Folder	VBA Macro (Project Name)
D:\Agilent	MixerCharacterization.vba

Executing Characterization of Calibration Mixer

1. Setting Stimulus Conditions

Set the stimulus conditions for the channel you want to calibrate. For the necessary steps, refer to Setting Stimulus Conditions.
You must also set the external signal source in advance. For the necessary steps, refer to Setting External Signal Source.

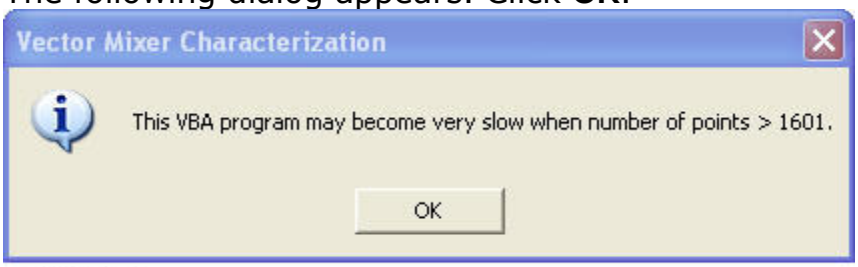
2. Running VBA Macro

1. Full 2-port calibration is recommended for characterizing the calibration mixer with the IF filter, although 1-port calibration is also available. This is because using full 2-port calibration simplifies the evaluation procedures, which are described in Overview of vector-mixer calibration . For detailed information on full 2-port calibration, see Full 2-Port Calibration.

Load the VBA project and then run it.

1. Press **Macro Setup**.
2. Click **Load & Project**.
3. Specify **D:\Agilent\ MixerCharacterization.vba** for the file name in the Open dialog box and then press the **Open** button.
4. Press **Macro Run** to run the macro.

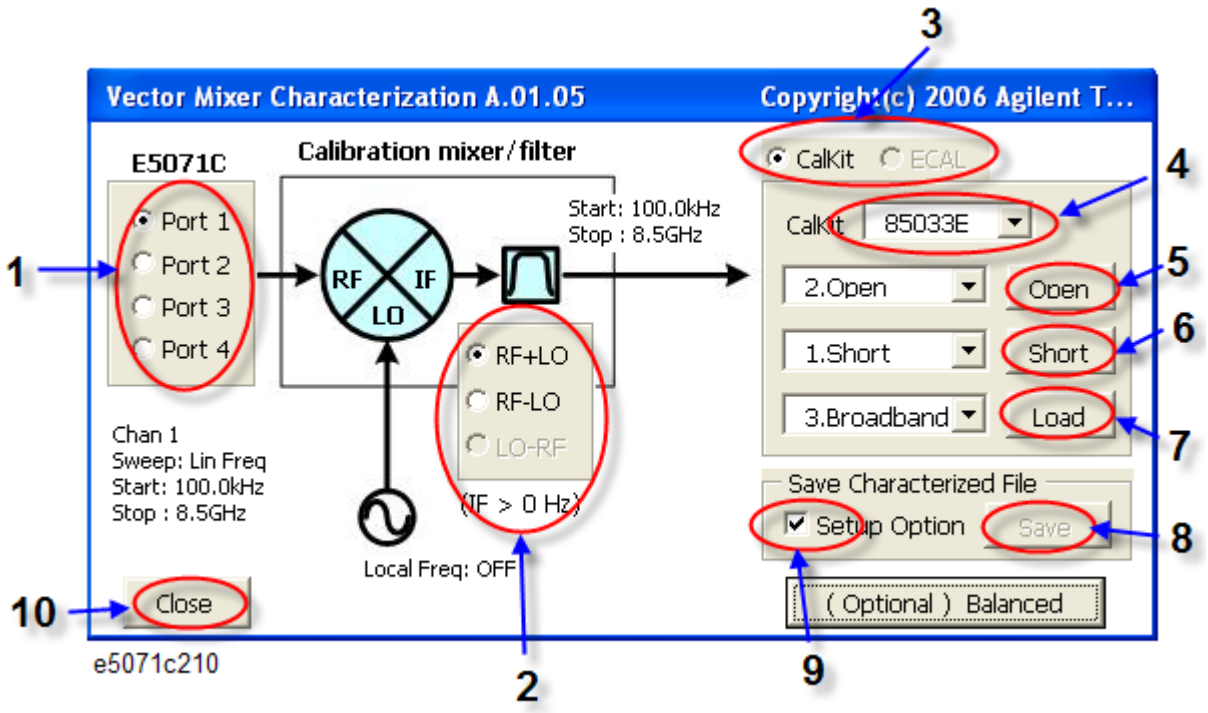
5. The following dialog appears. Click **OK**.



e5071c136

When more than 1601 measurement points is set for 1 channel and 4 traces, the E5071C VBA macro function may take more time to operate.

Vector Mixer Characterization Macro



3. Selecting Measurement Port

Select the 1-port calibration port (1 in menu).

- Select any port if full 2-port calibration is set.

4. Setting IF Frequency

Select IF frequency from RF+LO, RF-LO and LO-RF (2 in menu), depending on the IF frequency of the calibration mixer.

- The number displayed in the Vector Mixer Characterization macro is the frequency set in the E5071C and read from it. You

must also set the minimum IF frequency at more than 0 kHz. IF BW must be set to much smaller value than IF frequency.

5. Selecting a Calibration Kit

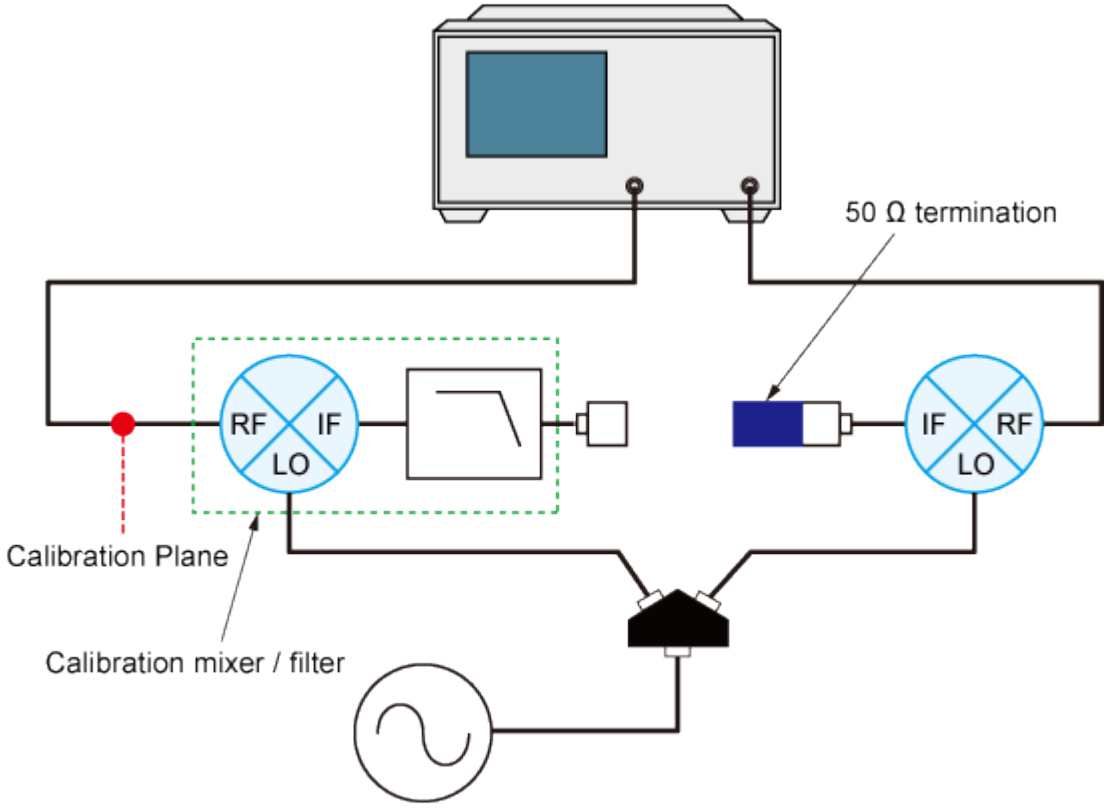
Select a calibration kit (3 in menu).

- The mechanical calibration kit displayed in the Vector Mixer Characterization macro is the frequency registered in the E5071C and read from it. If an ECal module is connected to the E5071C, ECal will be selected automatically.

6. Measuring Calibration Mixer with IF Filter (when using calibration kit)

Connect the calibration mixer to one of the test ports on which 1-port calibration has been done, as shown in the following figure.

Connection of calibration mixer (with IF filter)



e5071c341

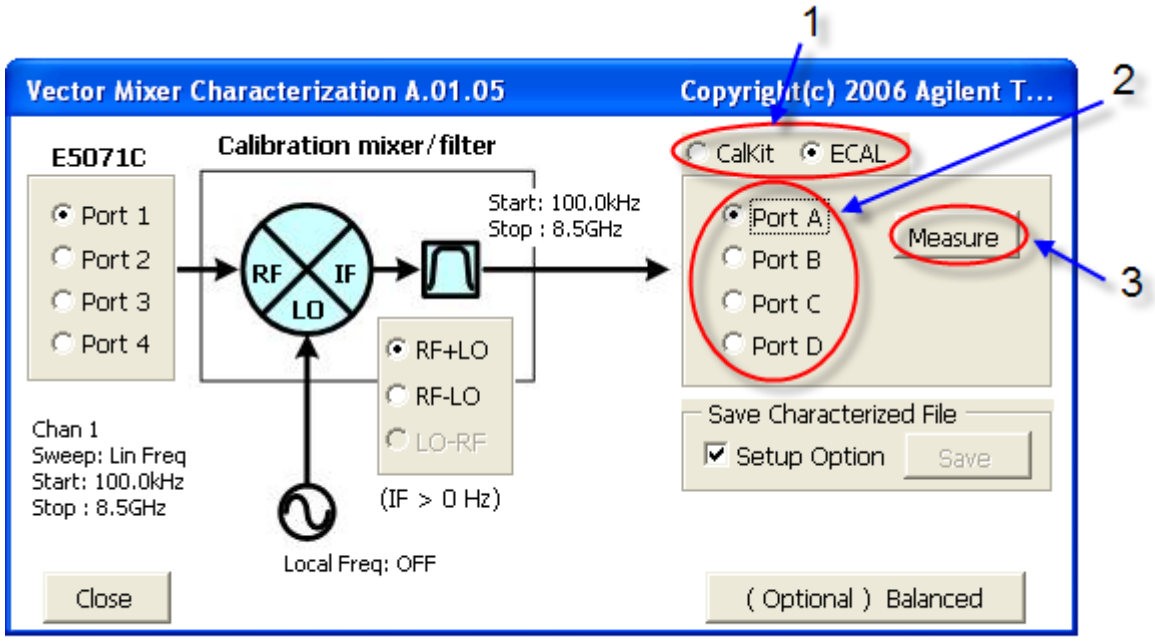
1. Select any port if full 2-port calibration is set.
2. We recommend that you characterize the calibration mixer with an IF filter when the power splitter for distributing the LO signal is connected to the measured mixer. In vector-mixer calibration,

where the up/down conversion method is used, the power of the LO signal is distributed to the calibration mixer and the measured mixer through the power splitter. During a characteristics evaluation of the calibration mixer, the LO power level used by the drive of the calibration mixer must be equal to the LO power level with the measured mixer connected. This is because the mixer's conversion loss and reflection coefficient are significantly affected by the power level of the LO signal.

1. Select **CalKit** (3 in menu).
2. Select the type number of the calibration kit from **CalKit** menu (4 in menu).
3. Click the **Open** button (5 in menu) to start measurement in OPEN.
4. Click the **Short** button (6 in menu) to start measurement in SHORT.
5. Click the **Load** button (7 in menu) to start measurement in LOAD.

7. Measuring Calibration Mixer with IF Filter (when using ECal module)

Characterizing calibration mixer using ECal



e5071c211

1. Select **ECal** (1 in menu).
2. Select the port used for the ECal module (2 in menu).
3. Click the **Measure** button (3 in menu) to start measurement.

8. Saving Characteristic Data and Closing VBA Macro

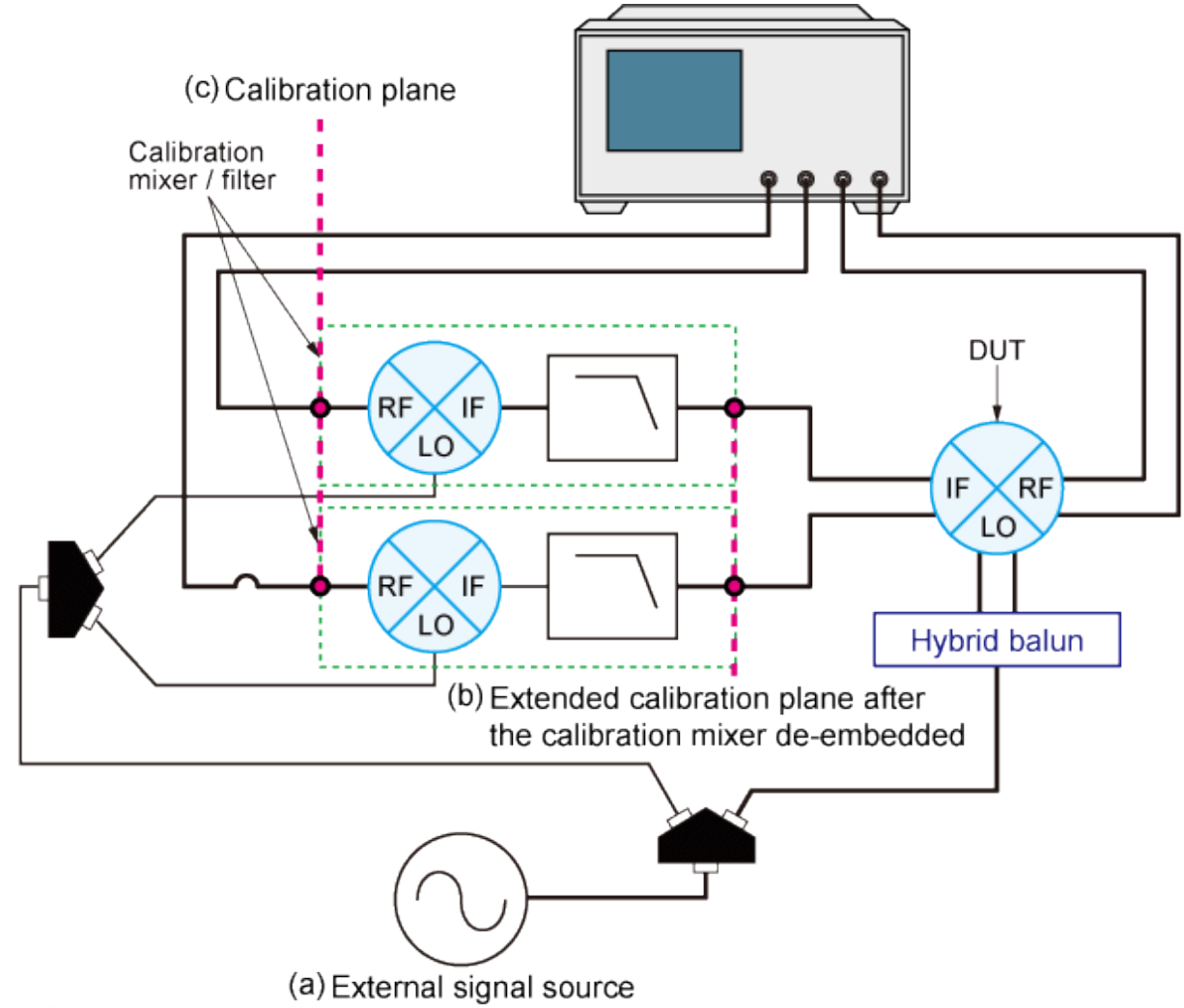
1. Press the Save button (8 in menu) to open the Save screen.
2. Press the Save button to specify a name for the characteristic data of the calibration mixer with IF filter. Then save it to a Touchstone file. If you check the Setup Option (9 in menu), the saved characteristic data will be set for the specified port of the active channel as the characteristic data file of the network de-embedding, and the fixture simulator function will be enabled. If unchecked, only the characteristic data will be saved.
3. Click the **Close** button (10 in menu) to exit the macro.

Characterizing Calibration Mixer (with IF filter) for Balance Mixer Measurement

The VBA macro (Vector Mixer Characterization) provided with the E5071C allows you to characterize the calibration mixer (with IF filter) to be used for the balanced mixer measurement. The characterizing procedures of the calibration mixer with IF filter used for balance mixer measurement are basically the same as those used for normal mixer measurement; however, two characteristic data of the calibration mixer with IF filter are required for balanced mixer measurement, as shown in the following figure.

Connect the target calibration mixer (with IF filter) to the port of the network analyzer on which calibration has been performed and then connect the OPEN, SHORT and LOAD standards to the end of the IF filter to start reflection measurement and characterization. For a balanced mixers, the phase difference of the LO signals between the calibration mixers with IF filter will remain as an error, since each calibration mixer with IF filter is characterized independently. Therefore, you must calibrate the phase difference between the two characterized calibration mixers with IF filters.

Characteristics evaluation of calibration mixer (with IF filter) for balance mixer



- We recommend that you use the same products of calibration mixers, IF filters and cables between the balanced ports to which each calibration mixer (with IF filter) is connected. You should keep the following electrical lengths the same between the two ports to which the calibration mixer is connected as much as possible - the electrical length from the external signal source output port (a) to the extended calibration plane (b), and the electrical length from the calibration plane (c) to the extended calibration plane (b). Large electrical length differences between the two ports to which the calibration mixer is connected could raise 180 degrees phase value error between the two IF ports even though the Balanced Mixer Calibration Macro is executed. You can verify it by swapping the IF cable connections with each other.

Executing Characterization

1. Characterizing Calibration Mixer (with IF filter)

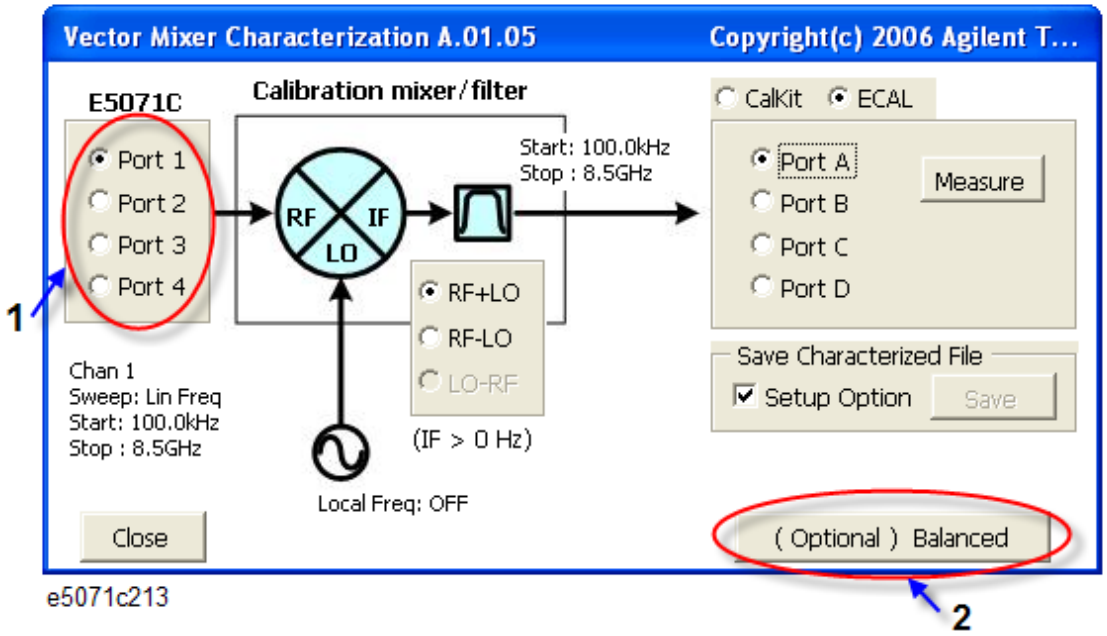
Measure the characteristic data of each calibration mixer with the IF filter used for balanced mixer measurement, using any two ports.

Set the stimulus conditions for the channel you want to calibrate. For the necessary steps, refer to Setting Stimulus Conditions.
You must also set the external signal source in advance. For the necessary steps, refer to Setting External Signal Source.

- If you characterize a calibration mixer with an IF filter, we recommend that you perform full 4-port calibration in advance, since it simplifies the evaluation procedures. For detailed information on full 4-port calibration, see Full 4-Port Calibration.

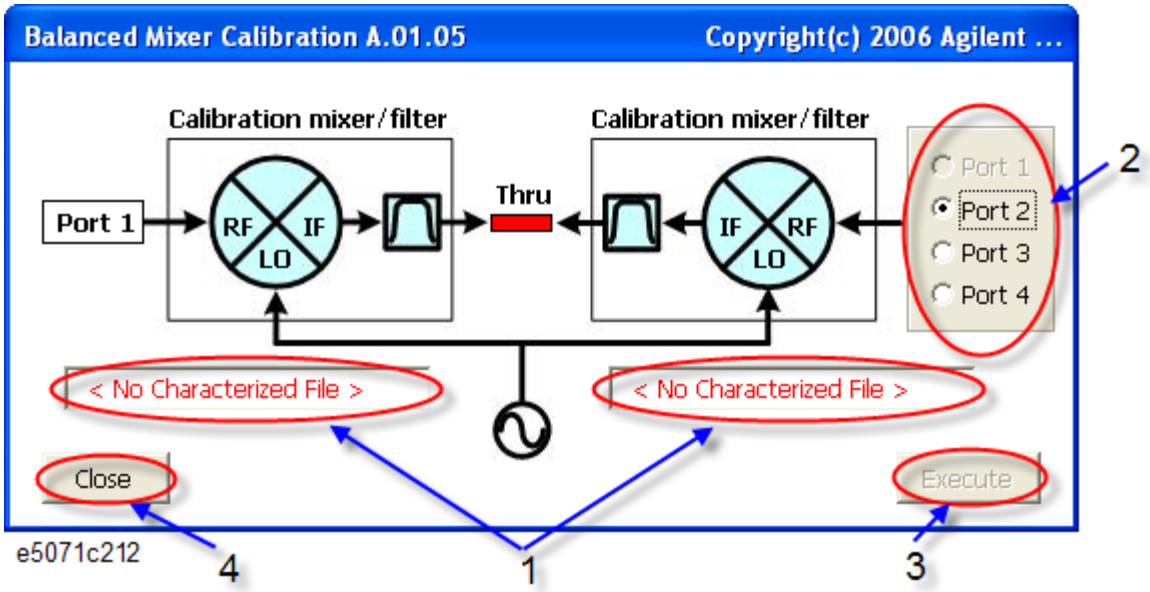
1. Press **Macro Setup** key.
2. Click **Load & Project**.
3. Specify "**D:\ Agilent\ MixerCharacterization.vba**" for the file name in the Open dialog box and then press the **Open** button.
4. Press **Macro Run** to run the macro.
5. Select **Port 1** (1 in menu) to characterize the calibration mixer 1 with IF filter. In this case, the data are saved to a temporary file (MIXER_1.s2p).
6. Select **Port 2** (1 in menu) to characterize calibration mixer 2 with IF filter. Here, the data are also saved to a temporary file (MIXER_2.s2p).
    - For detailed information on characterizing the calibration mixer, see Characterizing procedure for calibration mixer (with IF filter).
7. Click **(Optional) Balanced Mixer** (2 in menu).

Vector Mixer Characterization Macro



8. As the Vector Mixer Characterization Macro is running, the data files of the pre-measured calibration mixer with IF filter (MIXER_1.s2p, MIXER_2.s2p) are read automatically into the macro (1 in menu).
    - If failure occurs when reading the data file for the calibration mixer with IF filter, the characterization may have done by using only one port instead of using two ports.

Balanced Mixer Characterization Macro



1. Select the measurement port (2 in menu) and then connect a THRU between the IF ports of the calibration mixers to correct the phase difference of the LO signals for the calibration mixers with IF filters.

2. Pressing the **Execute** (3 in menu) button executes a phase error correction and overwrites the results on the original data file.
    - You cannot run the **Execute** function when selecting the measurement port if the data file of the calibration mixer with IF filter (*.s2p) has not been set for the network removal function of the fixture simulator.
    - The phase error correction data reflects the phase difference of the LO signals for the phase information of the calibration mixer's data file, which is registered in any two ports.
3. Press the **Close** button (4 in menu) to exit macro.

TRL Calibration

2-port TRL Calibration

- Overview
- Procedure

Other topics about TRL Calibration

Overview

The 2-port TRL calibration function lets you measure calibration data by connecting thru, reflection (open or short), line, or match calibration standards to (between) 2 desired test ports. This calibration provides the most accurate measurement for non-coaxial parts, using 12 error terms model in total for calibration in the same way as full 2-port calibration.

Before executing TRL calibration, you need to modify the calibration kit definition you use (or create a new one). For more information, see Changing the Calibration Kit Definition.

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to perform the calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **2-Port TRL Cal**.
4. Click **Select Ports**, then select the test ports for which you want to execute TRL 2-port calibration. (In the following procedure, test ports you select here are expressed as test ports x and y.)
5. Connect the port.

Thru/Line

1. Make a THRU or Line connection between ports x and y (between the connectors to which the DUT is to be connected).
2. Click **Thru/Line**.
3. Click **Port x-y Thru** to execute a Thru/Line measurement.
4. Click **Return**.

Reflection

1. Click **Reflect**
2. Click **Port x Reflect** to execute a reflection calibration for test port x.
3. Click **Port y Reflect** to execute a reflection calibration for test port y.
4. Click **Return**.

Line/Match

1. Click **Line/Match**.
2. Click **softkey** to execute Line/Match calibration.

Softkey	Function
x-y Line/Match	Executes line/match calibration for test ports x and y. Use this to perform forward-direction measurement and reverse-direction measurement at the same time. This is equivalent to measuring both **x-y Fwd (Syx)** and **x-y Rvs (Sxy)**.
x-y Fwd (Syx)	Executes line/match calibration for test ports x and y. Use this to perform forward-direction measurement only.
x-y Rvs (Sxy)	Executes line/match calibration for test ports x and y. Use this to perform reverse-direction measurement only.

3.
4. Click **Return**.
5. Click **Done** to finish TRL 2-port calibration. At this point, the calibration coefficient is calculated and saved. The error correction function is automatically turned on.

3-port TRL Calibration

- Overview
- Procedure

Other topics about TRL Calibration

Overview

The 3-port TRL calibration function lets you measure calibration data by connecting thru, reflection (open or short), line, or match calibration standards to (between) 3 desired test ports. This calibration provides the most accurate measurement for non-coaxial parts, using 27 error terms model in total for calibration in the same way as full 3-port calibration.

Before executing TRL calibration, you need to modify the calibration kit definition you use (or create a new one). For more information, see Changing the Calibration Kit Definition.

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to execute calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **3-Port TRL Cal** > **Select Ports**.
4. Select test ports for which you want to execute TRL 3-port calibration. (In the following procedure, test ports you select here are expressed as test ports x, y, and z.)
5. Click **Thru/Line** to start the measurement of the calibration standard.
6. Click **Reflect** to start the measurement of the calibration standard.
7. Click **Line/Match** to start the measurement of the calibration standard.

Softkey	Function
x-y Line/Match	Executes line/match calibration for test ports x and y. Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
x-y Fwd (Syx)	Executes line/match calibration for test ports x and y. Use this to perform forward-direction measurement only.
x-y Rvs (Sxy)	Executes line/match calibration for test ports x and y. Use this to perform reverse-direction

	measurement only.
x-z Line/Match	Executes line/match calibration for test ports x and z. Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
x-z Fwd (Szx)	Executes line/match calibration for test ports x and z. Use this to perform forward-direction measurement only.
x-z Rvs (Sxz)	Executes line/match calibration for test ports x and z. Use this to perform reverse-direction measurement only.
y-z Line/Match	Executes line/match calibration for test ports y and z. Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
y-z Fwd (Szy)	Executes line/match calibration for test ports y and z. Use this to perform forward-direction measurement only.
y-z Rvs (Syz)	Executes line/match calibration for test ports y and z. Use this to perform reverse-direction measurement only.

8. Click **Done** to finish TRL 3-port calibration. At this point, the calibration coefficient is calculated and saved. The error correction function is automatically turned on.

4-port TRL Calibration

- Overview
- Procedure

Other topics about TRL Calibration

Overview

The 4-port TRL calibration function lets you measure calibration data by connecting thru, reflection (open or short), line, or match calibration standards to (between) 4 test ports. This calibration provides the most accurate measurement for non-coaxial parts, using 48 error terms model in total for calibration in the same way as full 4-port calibration.

Before executing TRL calibration, you need to modify the calibration kit definition you use (or create a new one). For more information, see Changing the Calibration Kit Definition.

Procedure

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to execute calibration.
2. Press **Cal** key.
3. Click **Calibrate** > **4-Port TRL Cal** > **Select Ports**.
4. Click **Thru/Line** to start the measurement of the calibration standard.
5. Click **Reflect** to start the measurement of the calibration standard.

Softkey	Function
Port 1 Reflect	Executes reflection calibration for test port 1.
Port 2 Reflect	Executes reflection calibration for test port 2.
Port 3 Reflect	Executes reflection calibration for test port 3.
Port 4 Reflect	Executes reflection calibration for test port 4.

6. Click **Line/Match** to start the measurement of the calibration standard.

Softkey	Function
1-2 Line/Match	Executes line/match calibration for test ports 1 and 2.Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
1-2 Fwd	Executes line/match calibration for test ports 1 and

(S21)	2.Use this to perform forward-direction measurement only.
1-2 Rvs (S12)	Executes line/match calibration for test ports 1 and 2.Use this to perform reverse-direction measurement only.
1-3 Line/Match	Executes line/match calibration for test ports 1 and 3.Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
1-3 Fwd (S31)	Executes line/match calibration for test ports 1 and 3.Use this to perform forward-direction measurement only.
1-3 Rvs (S13)	Executes line/match calibration for test ports 1 and 3.Use this to perform reverse-direction measurement only.
1-4 Line/Match	Executes line/match calibration for test ports 1 and 4.Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
1-4 Fwd (S41)	Executes line/match calibration for test ports 1 and 4.Use this to perform forward-direction measurement only.
1-4 Rvs (S14)	Executes line/match calibration for test ports 1 and 4.Use this to perform reverse-direction measurement only.
2-3 Line/Match	Executes line/match calibration for test ports 2 and 3.Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
2-3 Fwd (S32)	Executes line/match calibration for test ports 2 and 3.Use this to perform forward-direction measurement only.
2-3 Rvs (S23)	Executes line/match calibration for test ports 2 and 3.Use this to perform reverse-direction measurement only.
2-4 Line/Match	Executes line/match calibration for test ports 2 and 4.Use this to perform forward-direction measurement

	and reverse-direction measurement at the same time.
2-4 Fwd (S42)	Executes line/match calibration for test ports 2 and 4.Use this to perform forward-direction measurement only.
2-4 Rvs (S24)	Executes line/match calibration for test ports 2 and 4.Use this to perform reverse-direction measurement only.
3-4 Line/Match	Executes line/match calibration for test ports 3 and 4. Use this to perform forward-direction measurement and reverse-direction measurement at the same time.
3-4 Fwd (S43)	Executes line/match calibration for test ports 3 and 4. Use this to perform forward-direction measurement only.
3-4 Rvs (S34)	Executes line/match calibration for test ports 3 and 4. Use this to perform reverse-direction measurement only.

7. Press **Done** to finish TRL 4-port calibration. At this point, the calibration coefficient is calculated and saved. The error correction function is automatically turned on.

Making Measurements

Making Measurements

- Setting Up Trigger and Making Measurements
- Setting Point Trigger
- Setting Low-Latency External Trigger Mode
- Setting Averaging Trigger Function
- Setting External Trigger Output

Setting Trigger and Making Measurements

- Overview
- Sweep Order in Each Channel
- Trigger Source
- Trigger Scope
- Trigger Mode
- Setup Procedure for Trigger

Other topics about Making Measurement

Overview

The E5071C has one trigger source. When this trigger source detects a trigger signal that has occurred, a sweep is performed for channels in the "Initiate" state in the order of channel 1 to channel 36. You set the "Initiate" or "Idle" status of each channel by changing the trigger mode.

- The execution of measurement for each channel does not depend on whether the channel is displayed. Channels that have been activated can be measured even if they are not displayed.
- For each channel, a sweep is performed only for the stimulus ports needed to update the parameters of the displayed trace.

Sweep Order in Each Channel

In a channel, each test port is set to a stimulus port in the order of port number and updates each trace.

Sweep Order	Stimulus Port	Updated Trace
1	Port 1	S11, S21, S31, S41
2	Port 2	S12, S22, S32, S42
3	Port 3	S13, S23, S33, S43
4	Port 4	S14, S24, S34, S44

- If full 2, 3, or 4-port error correction is in effect, no trace between calibrated ports is updated until the last calibrated port is swept as the stimulus port.
- Sweep is not executed for stimulus ports that are not required for updating traces.

Trigger Source

The trigger source generates a cue signal that initiates a measurement process. Four types of trigger sources are available.

Trigger Sources	Function
Internal (**Internal**)	Uses a consecutive signal generated by the firmware as a trigger source. Triggers are sent immediately following the completion of each measurement.
External (**External**)	Uses the external trigger input terminal (BNC) or Handler I/O (Pin No. 18) as a trigger source.
Manual (**Manual**)	A trigger is generated by pressing **Trigger** > **Trigger**.
Bus (**Bus**)	A trigger is generated through GPIB/LAN/USB.

Trigger Scope

The Trigger Scope specifies the scope of the triggering, whether it is for all channels (default) or for the selected channel.

When this function is enabled with a value of "ACTive", only active channel is triggered. When this function is enabled with a value of "ALL", all channels of the E5071C are triggered.

For example, if Trigger Scope is set to "ACTive" when **Trigger** > **Continuous** is selected for all channels, a measurement channel is automatically changed by switching over the active channel.

Trigger Scope	Function
All Channel	Triggers ALL channels which are in "Initiate" status
Active Channel	Triggers Active channel which is in "Initiate" status

Trigger Mode

You can set the trigger mode for each channel independently. This allows you to control the operation of each channel after a trigger signal is detected by setting the channel's status with the trigger mode.

Trigger mode name	Function
Sweep stop (**Hold**)	The status ("Idle" status) in which the sweep is stopped. When a trigger signal is detected, the

	sweep is not performed.
Single sweep (**Single**)	An "Initiate" status. When a trigger signal is detected, a sweep is performed. After completion of the sweep, the "Idle" status is activated.
Continuous sweep (**Continuous**)	An "Initiate" status. When a trigger signal is detected, a sweep is performed. After completion of the sweep, the "Initiate" status is maintained. The sweep is repeated each time a trigger signal is detected.
Hold All Channels	Sets all channel trigger modes to hold sweep mode
Continuous Disp Channels	Sets trigger modes of all displayed channels (**Display** > **Allocate Channels**) to continuous sweep mode

Setup Procedure For Trigger

1. Selecting a Trigger Source

Follow the procedure below to select a trigger source.

1. Press **Trigger**.
2. Click **Trigger Source**.
3. Click the softkey that corresponds to the desired trigger source.
4. When **External** is selected as a trigger source, click **Ext Trig Input** to select trigger polarity.

Softkey	Function
Negative Edge	Triggered at negative edge of an external trigger input signal.
Positive Edge	Triggered at positive edge of an external trigger input signal.

5. The setting for trigger polarity is not valid for the external trigger from Hander I/O.

2. Selecting a Trigger Scope

Follow the procedure below to select a trigger mode.

1. Press **Trigger** key.
2. Click **Trigger Scope**.

3. Click the softkey that corresponds to the desired trigger scope:

Trigger Scope	Function
All Channel	Triggers ALL channels
Active Channel	Triggers Active channel alone.

3. Selecting a Trigger Mode

Follow the procedure below to select a trigger mode.

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which the trigger mode will be set.
2. Press **Trigger** key.
3. Press the softkey that corresponds to the desired trigger mode. Repeat the procedure until each channel is set for its trigger mode.

Softkey	Function
Hold	Sets active channel trigger mode to hold sweep mode
Single	Sets active channel trigger mode to single sweep mode
Continuous	Sets active channel trigger mode to continuous sweep mode
Hold All Channels	Sets all channel trigger modes to hold sweep mode
Continuous Disp Channels	Sets trigger modes of all displayed channels (**Display** > **Allocate Channels**) to continuous sweep mode

4. Generating the Trigger

Next, it is necessary to generate a trigger by using the trigger source selected in Selecting a Trigger Source.

- Once the internal trigger source is selected, a series of triggers is continuously generated as soon as the setting becomes effective.
- Pressing **Trigger** > **Restart** during a sweep forces the analyzer to abort the sweep.

Setting Point Trigger

- Overview
- Setting Point Trigger

Other topics about Making Measurement

Overview

The point trigger provides a point measurement at every trigger, and it can be used to change the trigger event to point trigger mode.

Trigger event name	Function
On Point (On)	Point measurement is performed when trigger is applied.
On Sweep (Off)	Measurement is performed for all measurement points when trigger is applied.

- When the trigger source is selected as the internal trigger, the point trigger does not work.

Setting Point Trigger

1. Press **Trigger** key.
2. Click **Trigger Event**.
3. Click the softkey that corresponds to the desired trigger event.

Softkey	Function
On Point	Measures at each measurement point
On Sweep	Measures all measurement points

Setting Low-Latency External Trigger Mode

- Overview
- Setting Low-Latency External Trigger
- External Trigger Delay Time and Point Trigger Interval

Other topics about Making Measurement

Overview

In the low-latency external trigger mode, variations in delay time between the reception of a trigger and the start of a one-point measurement are decreased for point trigger measurement by using trigger pulses supplied to the external trigger input terminal. In addition, using this mode will allow you to decrease the variations in the time required to be ready for the next measurement trigger after execution of a one-point measurement. This mode allows you to perform continuous point trigger measurement in sync with external trigger pulses and accurately set a trigger delay time at each point-trigger point.

- The time required for measurement at each measurement point varies depending on the settings of the E5071C and the type of measurement. For more information, see External trigger delay time and point trigger interval.

The low-latency external trigger mode becomes available by turning on the point trigger function and setting the trigger source to "External."

Low-latency external trigger mode	Function
ON	Sets the low-latency external trigger mode to ON. The external trigger delay time becomes valid.
OFF	Sets the low-latency external trigger mode to OFF.

- If the trigger source is not the external trigger, the low-latency external trigger mode does not work if it is specified to ON.
- The setting of the low-latency external trigger mode is valid for all channels.

Setting Low-Latency External Trigger

1. Setting the low-latency external trigger

If the trigger source is set to "External" and the trigger event to "On" (On Point), set the low-latency external trigger mode.

1. Press **Trigger** key.
2. Click **Low Latency**.
3. Click **ON** to activate the-low latency external trigger mode.

2. Setting the external trigger delay time

Set the external trigger delay time at each point. The setting resolution is 10 µs.

1. Press **Trigger** key.
2. Click **Trig Delay**.
3. Enter an external trigger delay time.

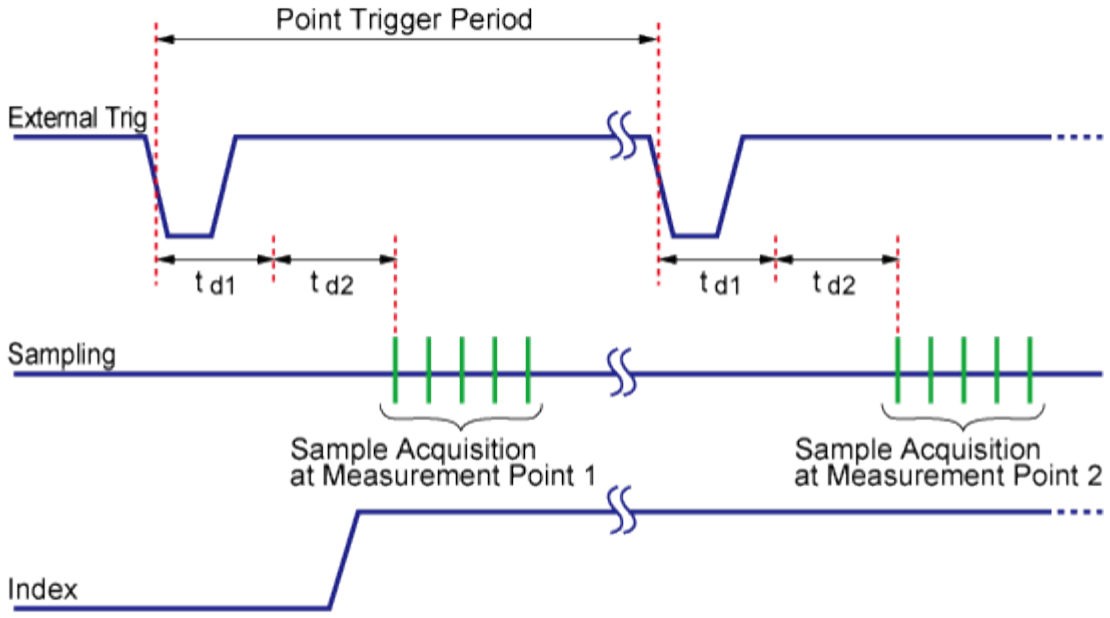
External Trigger Delay Time and Point Trigger Interval

External trigger pulses, supplied until the next measurement becomes ready after the start of a one-point measurement, are ignored, and the next trigger is generated by a pulse supplied after the completion of the one-point measurement.

The time until the next trigger can be accepted after the start of a one-point measurement depends on the IFBW and other settings of the analyzer. For example, in the case of a frequency's zero-span measurement, the time until the next measurement is ready after the start of a one-point measurement is obtained by dividing the time required for a single sweep in On Sweep mode, instead of On Point mode, by the number of measurement points. If you use the point trigger function with external trigger pulses that are wider than this time, point trigger measurement is performed at each pulse input.

The figure below shows the timing chart of an external trigger when the point trigger function is on.

Timing chart of external trigger (trigger event = on, trigger source = external, low-latency external trigger mode = on)



e5071c345

The table below describes signals and time as shown in the above figure.

Signal, time	Description
External Trig	External trigger signal to be supplied.
Sampling	Time while the E5071C is actually performing measurement.
Index	/Index signal of the handler I/O port. When the point trigger function is ON and the low-latency external trigger mode is ON, it goes to the High level only before staring measurement of the first sweep point and returns to the Low level after completing measurement of all measurement points.
Point Trigger Period	Time until the E5071C is ready to accept a trigger for the next measurement point. The value depends on the measurement conditions and the settings of the E5071C.
td1	Time set as the external trigger delay time.
td2	Time that the E5071C requires internally for setting. This depends on the settling time for setting the frequency, IF

	frequency band, and so on. For zero-span measurement, td2 = 13±1 μs (Typ.).

Setting Averaging Trigger Function

- Overview
- Averaging Trigger Function

Other topics about Making Measurement

Overview

The averaging trigger function is used to execute the sweep the number of times specified by the averaging factor with a single trigger when the sweep averaging function is **ON**.

Averaging Trigger	Function
ON	Performs the sweep the number of times specified by the averaging factor with a single trigger.
OFF	Performs the sweep once with a single trigger.

The averaging factor is cleared before the start of measurement.

- When the point trigger function is on, its setting has priority, and you need to generate triggers based on "(number of measurement points)× (averaging factor)".
- When the sweep averaging function is off, sweep is performed only once even if the averaging trigger function is set to on.
- The averaging trigger function is valid for all channels. Note that you can set the sweep averaging function for each channel.

Averaging Trigger Function

1. Setting Averaging Trigger Function

When the sweep averaging function is **ON**, follow these steps to set the averaging trigger function.

1. Press **Avg** key.
2. Click **Avg Trigger**.
3. Click **ON** to activate the averaging trigger.

2. Executing Averaging Measurement

1. Press **Trigger** key.
2. Click **Single**. The averaging factor is cleared before the start of measurement, the sweep is executed the number of times specified by the averaging factor, and then the instrument waits for the next trigger.

Setting External Trigger Output

- Overview
- Setting External Trigger Output

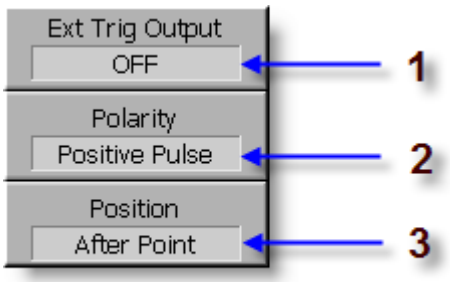
Other topics about Making Measurement

Overview

The External Trigger Output port (located at the rear panel) can be used to provide trigger to an external device. This is useful in cases where an external device needs to be triggered through the E5071C.

Setting External Trigger Output

1. Press **Trigger** key, and then set the value of **Ext Trig Output** (**1**) as **ON**.



e5071c196

2. Click **Polarity**, and then select **Positive Pulse** or **Negative Pulse** (**2**).

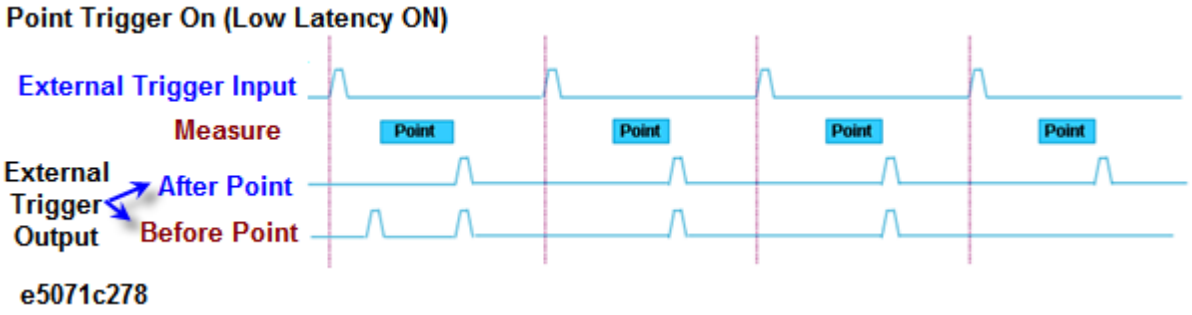
Polarity Property	Description
Positive Pulse	Generates a Positive Pulse as a trigger.
Negative Pulse	Generates a Negative Pulse as a trigger.

3. Click **Position**, and then select **After Point** or **Before Point** (**3**).

Position Property	Description
After Point	Generates a Pulse (trigger) after the measurement points.
Before Point	Generates a Pulse (trigger) before the measurement points.

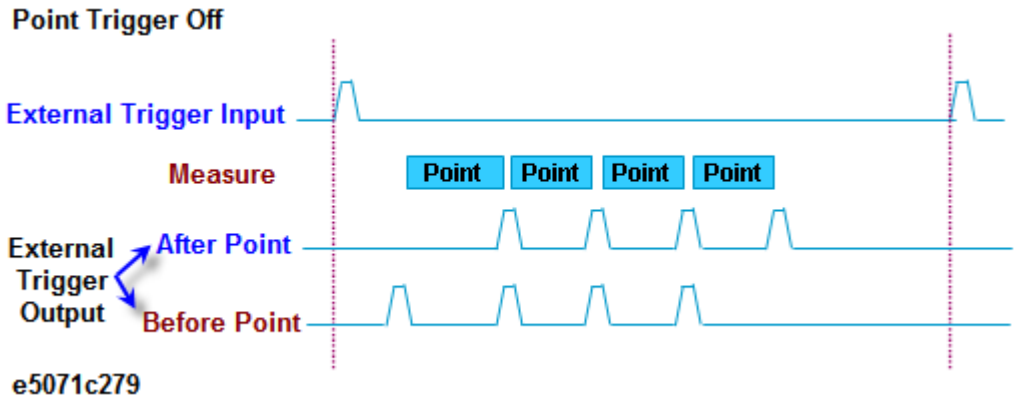
Difference between After and Before Point Settings

External trigger output function displays the difference result by setting the Point trigger as follows:



After Point:Pulse is generated after every measurement point.

Before Point:Pulse is generated before first measurement point. Rest of pulse are same timing as After Point. Number of pulse is same as number of measurement point.



After Point:Pulse is generated after every measurement point.

Before Point:Pulse is generated before every measurement point.

Data Analysis

Data Analysis

Data Analysis

- Analyzing Data on the Trace Using the Marker
- Searching for Positions that Match Specified Criteria
- Determining the Bandwidth of the Trace (Bandwidth Search)
- Determining the Bandwidth of the Trace (Notch Search)
- Determining the Mean, Standard Deviation, and p-p of the Trace
- Comparing Traces/Performing Data Math
- Performing Parameter Conversion of Measurement Results
- Limit Test
- Bandwidth Test
- Ripple Test
- Equation Editor

Analyzing Data on the Trace Using the Marker

- About Marker Functions
- Reading Marker Values on Trace
- Reading Relative Value from Reference Point on Trace
- Reading Actual Measurement Point/Value Interpolated between Measurement Points
- Setting up Markers for Each Trace/Setting up Markers for Coupled Operation between Traces
- Listing Marker Values in All Displayed Channels
- Specifying Display Position of Marker Values
- Aligning Marker Value Display
- Displaying All Marker Values for Displayed Traces

Other topics about Data Analysis

About Marker Functions

The marker can be used in the following ways:

- Reading a measured value as numerical data (as an absolute value or a relative value from the reference point)
- Moving the marker to a specific point on the trace (marker search)
- Analyzing trace data to determine a specific parameter
- Using the value of the marker to change the stimulus (sweep range) and scale (value of the reference line)

For the procedure used to change the sweep range and scale by using the marker, refer to Setting the Sweep Range Using the Marker and Setting the value of a reference line using the marker.

The E5071C is capable of displaying up to 10 markers including the reference marker on each trace. Each marker has a stimulus value (the value on the X-axis in rectangular display format) and a response value (the value on the Y-axis in rectangular display format). The Smith chart and polar formats each have two marker response values (log amplitude and phase).

Reading Marker Values on Trace

You can read the value of a marker displayed on the trace.

In rectangular display format, the marker response value is always in the same data format as that of the Y-axis. On the contrary, one format for the marker response values (two values: main and auxiliary) can be selected from among several types. The selection is performed in the data format.

Softkey for selecting data format	Marker response value	
	Main	Auxiliary
Smith - Lin / Phase	Linear amplitude	Phase
Smith - Log / Phase	Log amplitude	Phase
Smith - Real / Imag	Real component	Imaginary component
Smith - R + jX	Resistance	Reactance
Smith - G + jB	Conductance	Susceptance
Polar - Lin / Phase	Linear amplitude	Phase
Polar - Log / Phase	Log amplitude	Phase
Polar - Real / Imag	Real component	Imaginary component

For setting up data formats, refer to Selecting a Data Format.

Activating the marker on the Trace

1. Press **Channel Next**/**Channel Prev** keys to activate the channel on which a marker is used.
2. Press **Trace Next**/**Trace Prev** to activate the trace on which a marker is used.
3. Press **Marker** key. At this point, marker 1 is turned on and becomes active (you can operate the marker). When using marker 1, you can omit the next step.
4. Select a marker and turn it on. The softkey used to turn on a marker is also used to activate that marker.

Moving the marker

1. Change the marker stimulus value. This operation enables you to move the marker to a point on the selected trace.

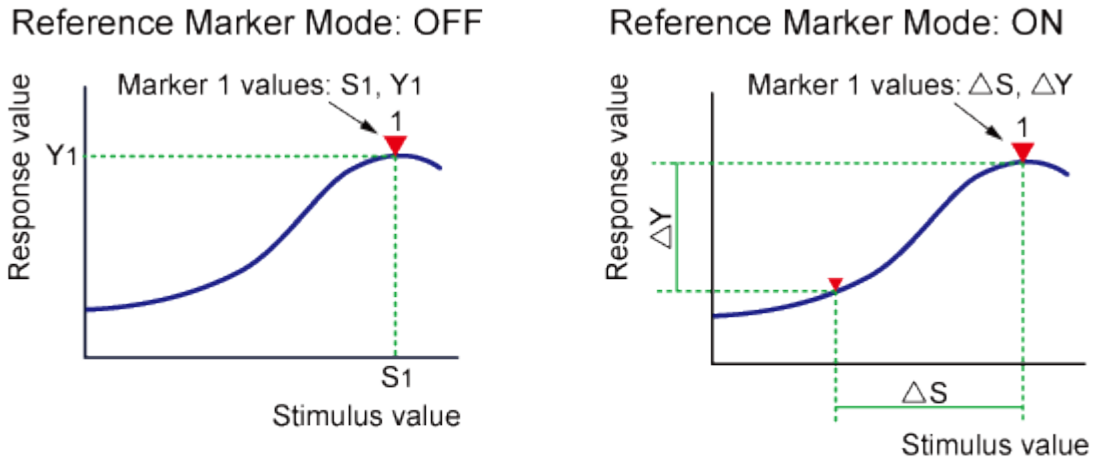
2. Read the marker stimulus value and marker response value displayed in the upper-left part of the trace screen.

Turning off the marker

1. Press **Marker** key.
2. Click **Clear Marker Menu** and then click one of the options.
   - In the preset configuration, the marker settings on traces in a channel are coupled (Marker Couple is turned on). For marker coupling, refer to Setting up markers for each trace/Setting up markers for coupled operations between traces.

Reading Relative Value from Reference Point on Trace

You can convert the marker reading into a relative value from the reference point.



e5071c310

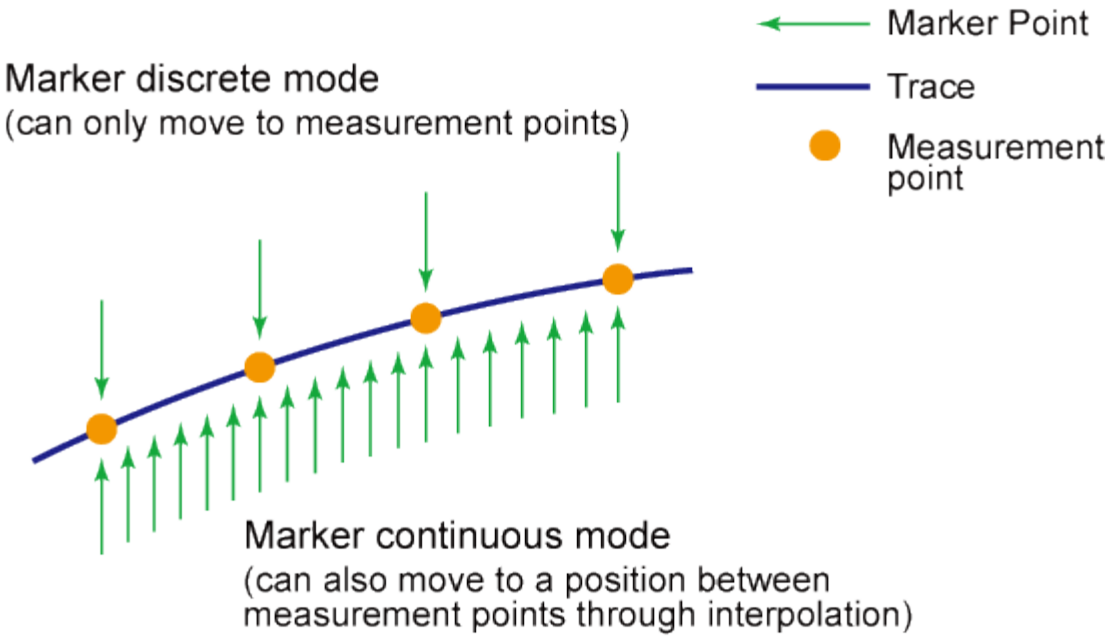
Converting From a Reference Point to a Relative Value

1. Activate the reference marker.
2. Move the reference marker at the point to be used as the reference.
3. Click **Ref Marker Mode** to turn on the reference mode.
4. With the reference mode turned on, the stimulus values and response values are indicated in relative values referred to by the position of the reference marker.
5. Activate your desired marker, then move it to your desired position.
6. Pressing **Marker** and then clicking **Ref Marker** enables you to place the reference marker at the position of the currently *active marker* . The reference mode will then turn on automatically.

Reading Actual Measurement Point/Value Interpolated between Measurement Points

The point on the trace on which a marker can be placed differs depending on how the discrete marker mode is set up.

Value	Description
Turning on discrete mode (**Discrete ON**)	A marker moves only between actual measurement points. When a specific marker stimulus value is specified as a numerical value, the marker is placed at the measurement point closest to the specified value. A marker that is placed between interpolated points with the discrete mode off automatically moves to the nearest measurement point when the discrete mode is turned on.
Turning off discrete mode (**Discrete OFF**)	The marker can move from one actual measurement point to another. Because it is interpolated, it can also move in the space between measurement points.



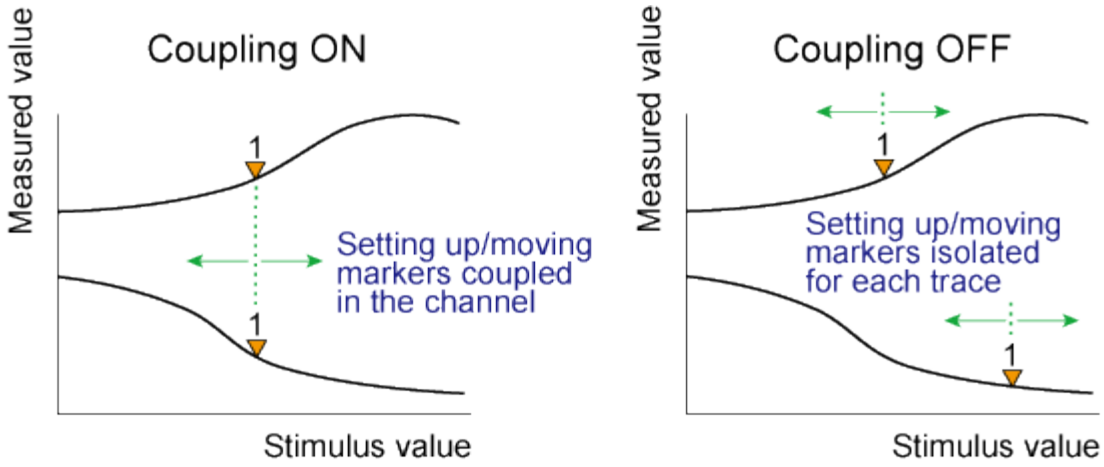
e5071c311

Turning Discrete Mode On or Off

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** to activate the trace on which the discrete mode is set up.
2. Press **Marker Fctn** key.
3. Click **Discrete** to turn the discrete mode on or off.

Setting up Markers for Each Trace / Setting up Markers for Coupled Operation between Traces

Makers can be set up and moved either in coupled operation for all traces in a channel or independently for each trace.



e5071c388

Value	Description
Marker Couple is on (**Coupling ON**)	Markers are set up and moved in coupled operation on all traces in a channel.
Marker Couple is off (**Coupling OFF**)	Markers are set up and moved independently for each trace.

Turning Marker Coupling On or Off

1. Press **Channel Next**/**Channel Prev** keys to activate the channel on which the marker couple will be set.
2. Press **Marker Fctn** key.
3. Click **Couple** to turn the marker coupling on or off.

Listing all Marker Values in all Displayed Channels

You can list all of the marker values in all of the displayed channels on the screen.

Turning On the Marker Table Display

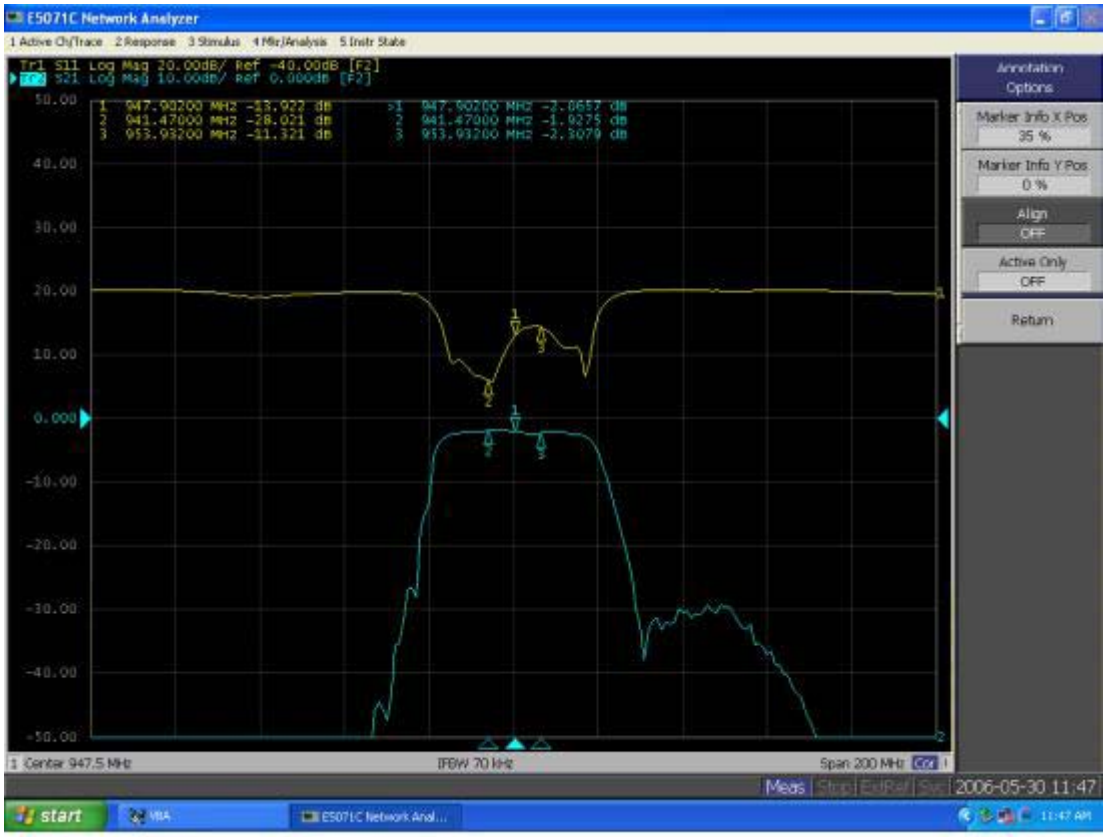
1. Press **Marker Fctn** key.
2. Click **Marker Table** to turn on the marker table display.

The marker table appears in the lower part of the screen.

e5071c063

Specifying Display Position of Marker Values

This section describes how to specify the marker value display position for each active trace.



e5071c064

Value	Description
Marker Info X Pos	Specifies the horizontal display position by the width of the display area as a percentage.
Marker Info Y Pos	Specifies the vertical display position by the height of the display area as a percentage.

Operational procedure

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set marker coupling.
2. Press **Marker Fctn** key.
3. Click **Annotation Options**.
4. Click **Marker Info X Pos** to set the horizontal display position.
5. Click **Marker Info Y Pos** to set the vertical display position.

Aligning Marker Value Display

This section describes how to align maker value displays.

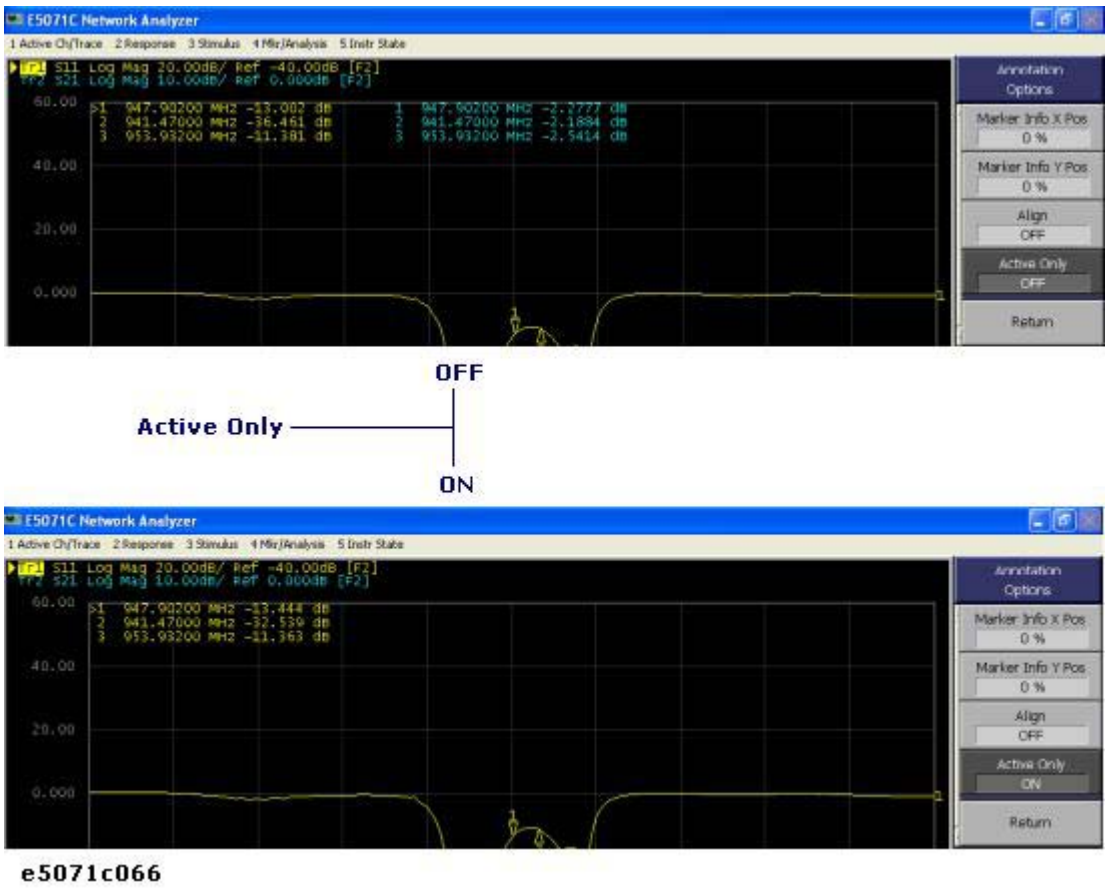
e5071c065

Value	Description
On (**Align ON**)	Displays marker values to align to the display position of trace 1.
Off (**Align OFF**)	Displays marker values in the display position defined for each trace.

1. Press **Marker Fctn** key.
2. Click **Annotation Options**.
3. Click **Align** to toggle ON/OFF.

Displaying All Marker Values for Displayed Traces

This section describes how to display all marker values for displayed traces.



e5071c066

Value	*Description*
Displays all (**Active Only OFF**)	Displays all marker values for displayed traces.
Displays active markers (**Active Only ON**)	Displays markers for the active trace only.

1. Press **Marker Fctn** key.
2. Click **Annotation Options**.
3. Click **Active Only** to toggle ON/OFF.

Searching for Positions that Match Specified Criteria

- Overview
- Setting Search Range
- Setting Multiple Range
- Automatically Executing a Search (Search Tracking)
- Searching for Maximum and Minimum Values
- Searching for the target value (Target search)
- Searching for the peak

Other topics about Data Analysis

Overview

You can search for a position that matches your specified criteria by using the Marker Search feature. Marker Search allows you to search for a position that matches any of the following criteria.

- Maximum value
- Minimum value
- Target (a point that has a target measurement value)
  - Target nearest to marker position
  - Target on left-hand side nearest to marker position
  - Target on right-hand side nearest to marker position
- Peak
  - Maximum peak (for a positive peak), minimum peak (for a negative peak)
  - Peak on left-hand side nearest to marker position
  - Peak on right-hand side nearest to marker position

Setting Search Range

The Marker Search feature allows you to set part of the sweep range as the search target (Partial Search feature) as well as the entire search range. For the Partial Search feature, you can select whether to couple traces in the channel. The Marker Statistics value is calculated by using the search range.

Procedure to Turn ON/OFF Trace Coupling within Search Range

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace for which you want to set the search range.
2. Press **Marker Search** key.

3. Click **Search Range**.
4. Click **Couple** to toggle *ON/OFF* trace coupling within the search range.

Procedure to Set Search Range

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace for which you want to set the search range.
2. Press **Marker Search** key.
3. Click **Search Range**.
4. Click **Search Range** to turn ON the Partial Search feature.
5. Click **Start**, then enter the start value (lower limit) of the search range.
6. Click **Stop**, then enter the stop value (upper limit) of the search range.

Procedure to Show Statistics Value of the Search Range

1. Set and define a search range as described above.
2. Press **Marker Fctn** key.
3. Click **Statistics** to turn ON. The statistics values are calculated and displayed on the screen.

Setting Multiple Range

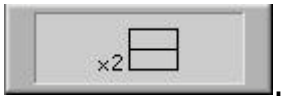
The Multiple Range feature allows you to set up to 16 search range in a channel. You can select more than one marker in a search range. You can select to couple the search range among the traces in the channel.

The Marker Statistics value is calculated by using the search range of the active marker. If no active marker is selected, by default the search range of marker 1 is used.

Procedure to Place Multiple Markers within a Search Range (in Multiple Range)

The below example shows how to place multiple markers within several search range on two traces. In this example, stimulus is set for two. Then, two multiple ranges are set and two markers are assigned to each search range as min and max. The effect of search range coupling and trace coupling on multiple range is also shown in this example.

Setting up traces and stimulus

1. **Display** > **Num of Traces** > **2**.
2. **Display** > **Allocate Traces** > 

.
3. **Start** > **2.0000 GHz**.

4. **Stop** > **8.0000 GHz**.
5. **Scale** > **Autoscale All**.

Turning ON and setting up multiple range

1. Select **Trace 1**.
2. Press **Marker Search** key.
3. Click **Search Range** > **Search Range** > **ON** to turn ON the partial search feature.
4. Click **Multiple Range** > **Multiple Range** > **ON** to turn ON the multiple search range feature.
5. Click **Target Range** and select within 1 to 16 which represents the search range number. In this example, select **1** which represents Search Range 1.
6. Click **Start**, then enter **3.0000 GHz**, the start value (lower limit) of the search range.
7. Click **Stop**, then enter **4.5000 GHz**, the stop value (upper limit) of the search range.
8. Similarly, click **Target Range** > **2**.
9. **Start** > **5.0000 GHz**.
10. **Stop** > **6.5000 GHz**.
11. The search range is set for both, Trace 1 and Trace 2. Note that the search range is not displayed as no Marker is assigned to these search range.

Assigning markers to search range

1. **Marker** > **Marker 1** to turn it ON.
2. **Marker Search** > **Search Range** > **Multiple Range** > **Target Range** > **1** to activate search range 1.
3. Similarly, turn ON **Marker 2**.
4. Activate search range 1, by following Step 2.
5. Note that now, search range 1 is displayed, and marker 1 and marker 2 are placed within search range 1.

Moving markers to min or max

1. **Marker** > **Marker 1** to activate it.
2. **Marker Search** > **Max**. Marker 1 is placed at the maximum point within search range 1.

3. **Marker** > **Marker 2** to activate it.
4. **Marker Search** > **Min**. Marker 2 is placed at the minimum point within search range 1.

Using search range coupling and marker coupling in multiple search range

1. Select **Trace 2**.
2. By default, search range coupling is turned ON. **Marker Search** > **Search Range** > **Couple** > **OFF** to turn OFF search range coupling.
3. **Marker** > **Marker 3** to activate it.
4. Activate search range 2.
5. Note that Trace 1 displays search range 1 while Trace 2 displays search range 2.
6. **Marker Search** > **Max**. Marker 3 is now placed at the maximum point within search range 2. Marker 3 at Trace 1 is in parallel with Marker 3 at Trace 2. However, Trace 1 still displays search range 1 while Trace 2 displays search range 2.
7. By default, marker coupling is turned ON. **Marker Fctn** > **Couple** > **OFF** to turn OFF marker coupling.
8. **Marker** > **Marker 4** to activate it. As marker coupling is turned OFF, marker 4 is activated for Trace 2 only. To learn more about marker coupling, refer to Setting up markers for each trace/Setting up markers for coupled operations between traces.
9. Activate search range 2.
10. **Marker Search** > **Min**. Marker 4 is now placed at the minimum point within search range 2. However, note that Marker 4 is not shown at Trace 1.

Procedure to Show Statistics Value of the Active Search Range

1. Click **Search Range** > **Multiple Range** > **Target Range**.
2. Select the search range number which you want its statistics value to be calculated and shown on the screen.
3. Press **Marker Fctn** key.
4. Click **Statistics** to turn ON the display of statistics data.

Automatically Executing a Search (Search Tracking)

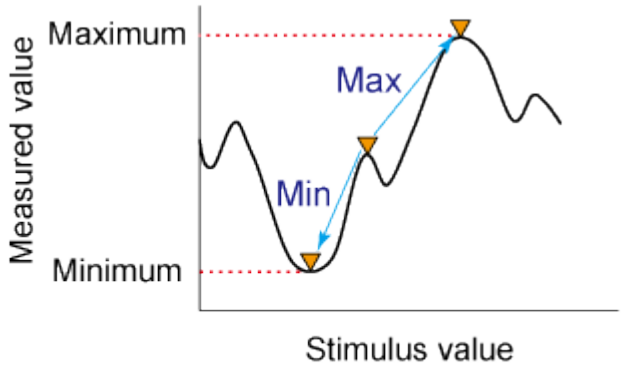
Search tracking is a function that sets a search to be repeated every time a sweep is done even if the execution key for the search (maximum, minimum, peak, and target) is not pressed. This function facilitates observation of measurement results such as the maximum value of traces (e.g., the insertion loss of a band pass filter).

Performing Search Tracking

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace on which you want to set up search tracking.
2. Press **Marker Search** key.
3. Click **Tracking** and turn the search tracking function *ON*/*OFF*.

Searching for Maximum and Minimum Values

You can search for the maximum or minimum measured value on the trace and move a marker to that point.



e5071c386

Search for maximum (**Max**)	Move *active marker* to point on trace where measured value is greatest
Search for minimum (**Min**)	Move active marker to point on trace where measured value is lowest

Procedure

1. Activate the marker you are using to search for the maximum and minimum values.
2. Press **Marker Search** key.
3. Click the corresponding softkey to move the marker to the maximum or minimum measured value.
   - When the data format is in Smith chart or polar format, execute the search only for the main response value.

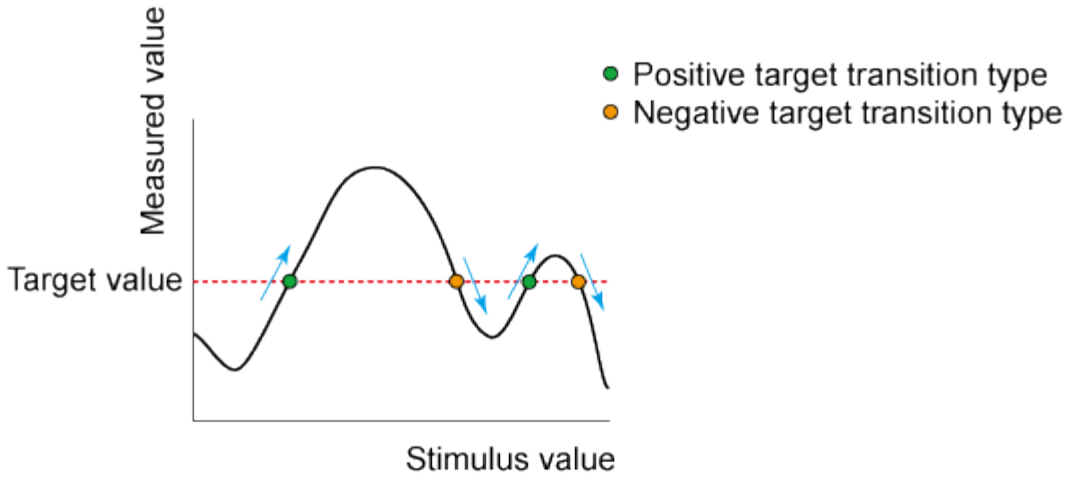
Searching for the target value (target search)

The target search is a function that searches for a target that matches the pre-defined target value and transition type(s) (positive, negative, or both positive and negative) and then moves the marker to that target.

Target and Transition Types

A target is a point that has a specific measured value on the trace. Targets can be divided into the two groups shown below depending on their transition type.

Transition type: Positive (**Positive**)	When the value of the target is larger than the measured value that immediately precedes it (on the left side)
Transition type: Negative (**Negative**)	When the value of the target is smaller than the measured value that precedes immediately it (on the left side)



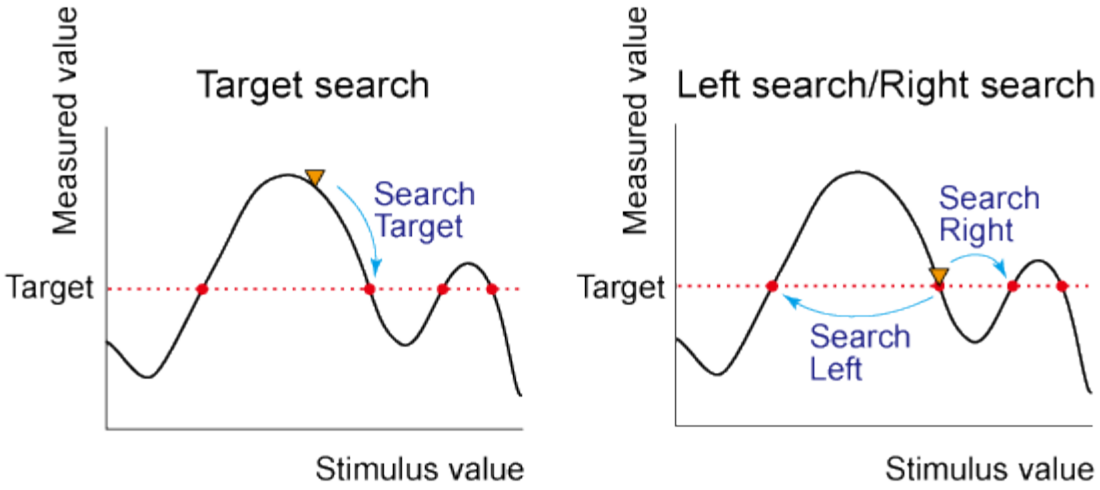
e5071c389

Executing a Target Search

The following three methods are available for executing the target search:

Target search (**Search Peak**)	The marker moves to the peak with maximum response value if the peak polarity is **Positive** or **Both** or to the peak with minimum response value if the peak polarity is **Negative**.
Search left (**Search Left**)	Executes the search from the current marker position to the smaller stimulus values and moves the marker to first target encountered.

Search right (**Search Right**)	Executes the search from the current marker position to the larger stimulus values and moves the marker to first target encountered.



e5071c387

Procedure

1. Activate the marker you are using for the target search.
2. Press **Marker Search** key.
3. Click **Target** > **Target Value**
4. Enter the target value in the entry area that appears. This causes the target search to be executed based on the definitions of the newly set target value and the currently set transition type.
5. Click **Target Transition**.
6. Select a transition type. This sets the target search to be executed based on the definitions of the currently set target value and the newly set transition type.
7. Press the corresponding softkey to move the marker to the target.
   - When the data format is in Smith chart or polar format, execute the search for the main response value of the two marker response values.

Searching for the peak

The peak search function enables you to move the marker to the peak on the trace.

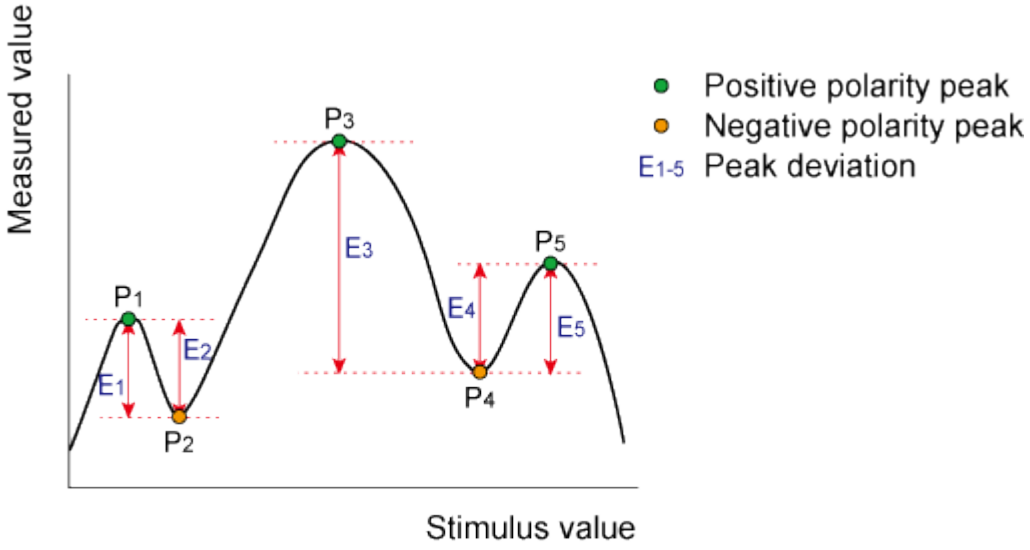
Definition of the peak

A peak is a measurement point whose value is greater or smaller than the adjoining measurement points on its right and left sides. Peaks are classified into the following two types depending on the difference in magnitude from the measurement points on either side of it.

Positive peak (**Positive**)	A peak whose measured value is greater than those of the measurement points on either side of it (peak polarity: positive)
Negative peak (**Negative**)	A peak whose measured value is smaller than those of the measuring points on either side of it (peak polarity: negative)

About Peak Excursion Value

The peak excursion value is the smaller of the differences in measured values from the adjoining peaks of the opposite polarity.



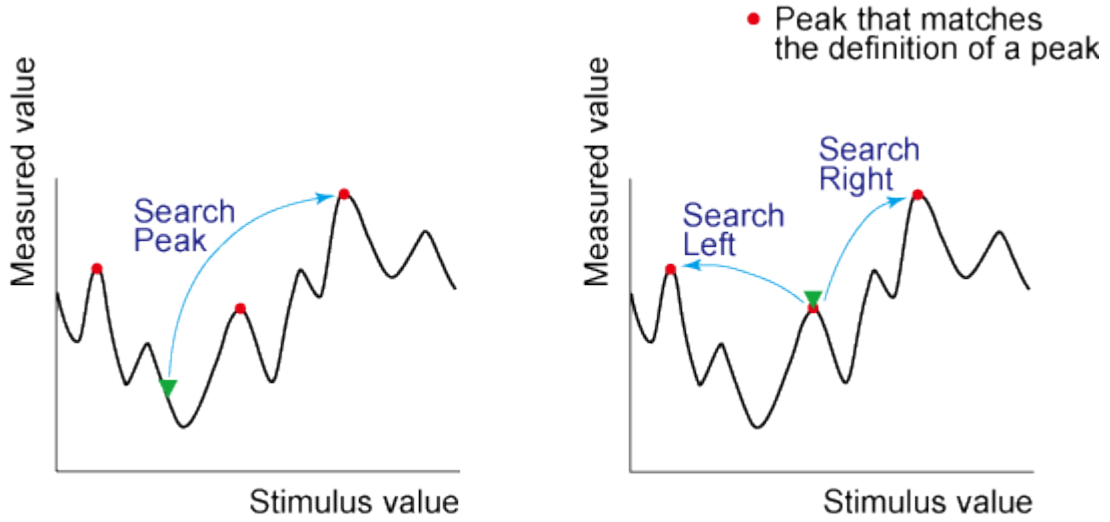
e5071c390

Executing a Peak Search

The following three methods are available for executing the peak search:

Peak search (**Search Peak**)	Moves the marker to the maximum peak when peak polarity is **Positive** or **Both**. Moves the marker to the minimum peak when peak polarity is **Negative**.
Left	Executes the search from current marker position to the

search (**Search Left**)	smaller stimulus values and moves the marker to first peak encountered.
Right search (**Search Right**)	Execute the search from current marker position to the larger stimulus values and moves the marker to first peak encountered.



e5071c452

Procedure

1. Activate the marker you are using for the peak search.
2. Press **Marker Search** key.
3. Click **Peak** > **Peak Excursion**.
4. Enter the lower limit for the peak excursion value. This sets the peak search to be executed based on the definitions of the newly set lower limit for the peak excursion value and the currently set peak polarity.
5. Click **Peak Polarity**.
6. Select a peak polarity. This sets the peak search to be executed based on the definitions of the currently set lower limit for the peak excursion value and the newly set peak polarity.
7. Click the corresponding softkey to move the marker to the peak.
8. When the data format is in Smith chart or polar format, execute the search for the main response value of the two marker response values.

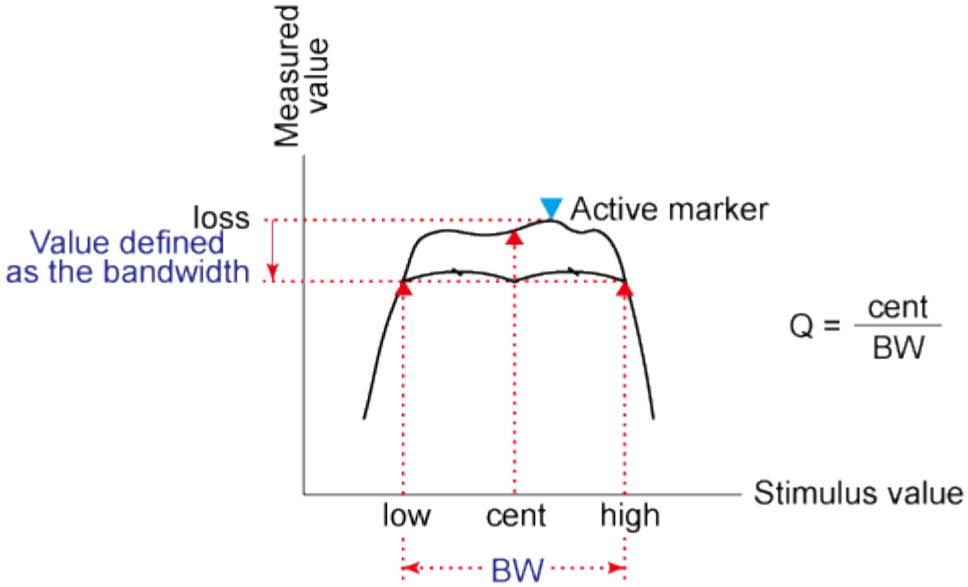
Determining the Bandwidth of the Trace (Bandwidth Search)

- Overview
- Executing a Bandwidth Search

Other topics about Data Analysis

Overview

The bandwidth search is a function for determining the bandwidth of the trace, center frequency, cut-off points (on the higher frequency and the lower frequency sides), Q, and insertion loss, based on the position of the *active marker* . The definitions of the parameters determined through the bandwidth search are shown in below.



e5071c392

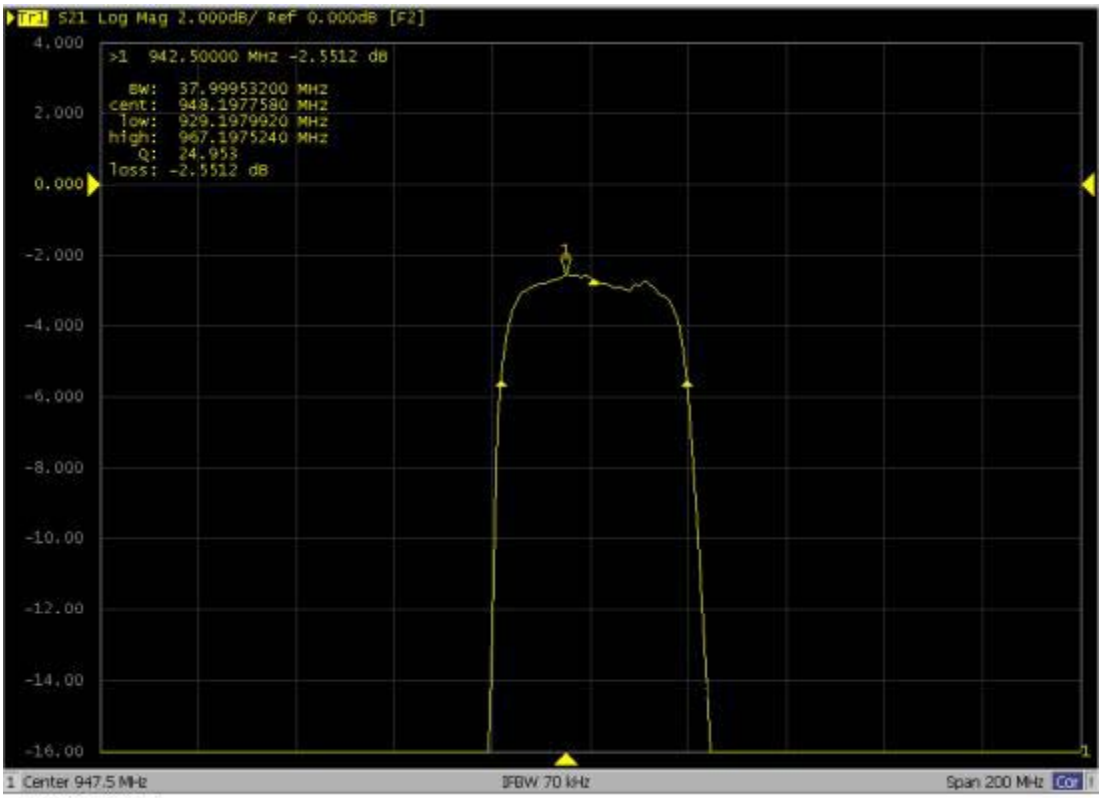
Bandwidth Parameter	Definition
Insertion loss (*loss*)	The measured value of the position of the active marker at the time the bandwidth search is executed.
Lower frequency cut-off point (*low*)	Lowest frequency of two measurement points, both separated by the defined bandwidth value from the active marker position.

Higher frequency cut-off point (*high*)	Highest frequency of two measurement points, both separated by the defined bandwidth value from the active marker position.
Center frequency (*cent*)	Frequency at the midpoint between the lower frequency cut-off and higher frequency cut-off points. (*high+low)/2*
Bandwidth (*BW*)	The difference in frequency between the higher frequency cut-off and lower frequency cut-off points. (*high-low)*
Q	Value obtained by dividing the center frequency by the bandwidth.(*cent*/*BW*)

Executing a Bandwidth Search

1. Place the active marker at the desired point on the trace on which the bandwidth search is executed. The response value of this active marker itself is the insertion loss in the bandwidth search (**loss**).
2. Press **Marker Search** key.
3. Click **Bandwidth Value** and enter the defined bandwidth value in the entry area that appears.
4. Click **Bandwidth** to turn on the bandwidth search. In the upper left of the trace display, six bandwidth parameters are displayed.

Bandwidth search results


Tr1 S21 Log Mag 2.000dB/ Ref 0.000dB [F2]
>1 942.50000 MHz -2.5512 dB
BW: 37.99953200 MHz
cent: 948.1977580 MHz
low: 929.1979920 MHz
high: 967.1975240 MHz
Q: 24.953
loss: -2.5512 dB
4.000
2.000
0.000
-2.000
-4.000
-6.000
-8.000
-10.00
-12.00
-14.00
-16.00
1 Center 947.5 MHz
IFBW 70 kHz
Span 200 MHz
Cor

e5071c077

Determining the Bandwidth of the Trace (Notch Search)

- Overview
- Executing a Notch Search

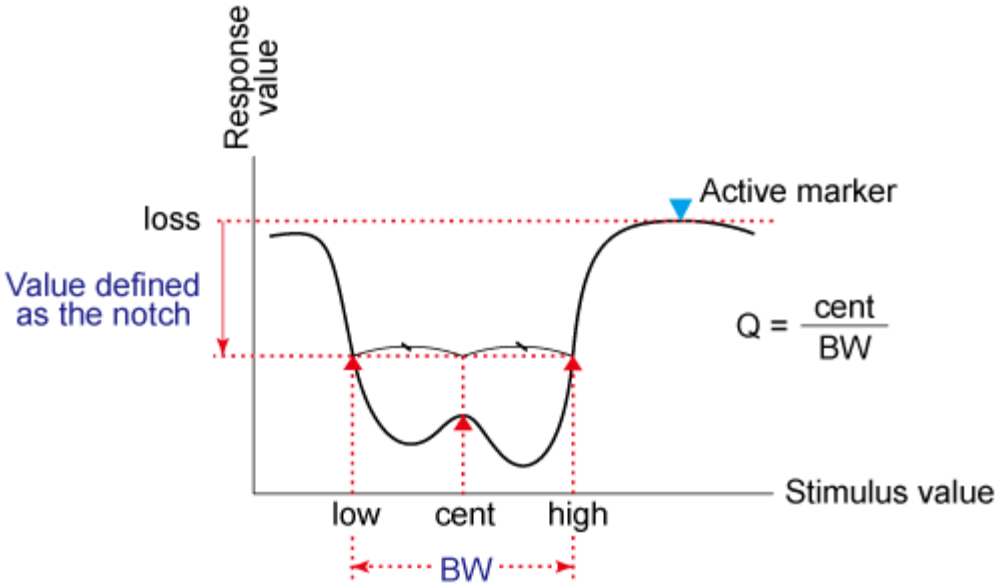
Other topics about Data Analysis

Overview

The notch search function is used to obtain the bandwidth, center frequency, cutoff points (high-frequency side and low-frequency side), Q, and insertion loss of a trace based on the active marker position. The notch search function starts from the left side of the active marker position, and ends when points that meet the conditions are found.

The figure and the table below shows the definition of parameters obtained by notch search function. The notch value in figure below must be specified by the user.

Bandwidth Parameters



e5071c484

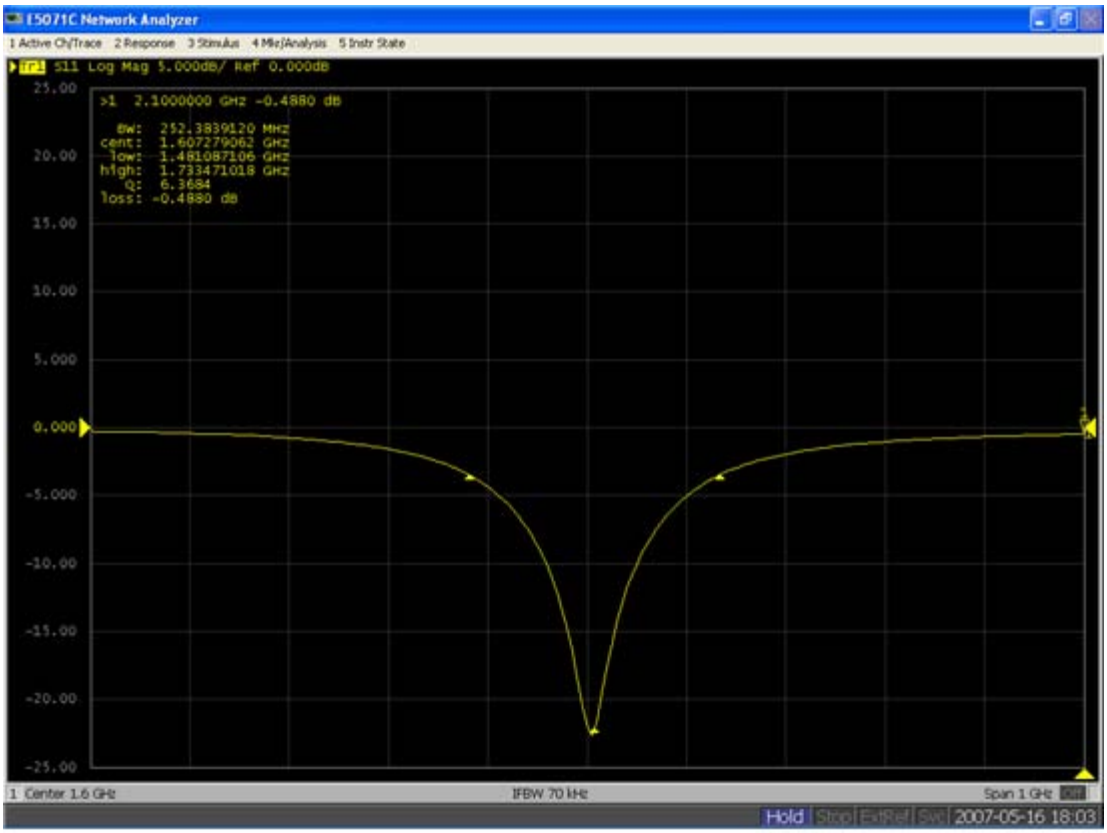
Definition of bandwidth parameters

Bandwidth parameter name	Definition
Insertion loss (loss)	Measurement value at the active marker position when the notch search is executed.

Lower cutoff point (low)	Lower frequency of the 2 points on both sides that have the measurement value apart from the active marker position by the notch value.
Higher cutoff point (high)	Higher frequency of the 2 points on both sides that have the measurement value apart from the active marker position by the notch value.
Center frequency (cent)	Frequency of the middle point between the lower cutoff point and the higher cutoff point (high + low)/2.
Bandwidth (BW)	Frequency difference between the higher cutoff point and the lower cutoff point (high – low).
Q	Value obtained by dividing the center frequency by the bandwidth (cent/BW).

Executing a Notch Search

1. Place the active marker on the desired point on the trace on which the notch search is executed. The response value of this active marker itself is the insertion loss in the notch search (loss).
2. Press **Marker Search.**
3. Press **Notch Value** and enter the notch value in the entry area that appears.
4. Press **Notch** to turn on the notch search. In the upper left of the trace display, six bandwidth parameters are displayed (see the figure below).



e5071c485

5. For more information on displaying the notch search result, see Notch Search

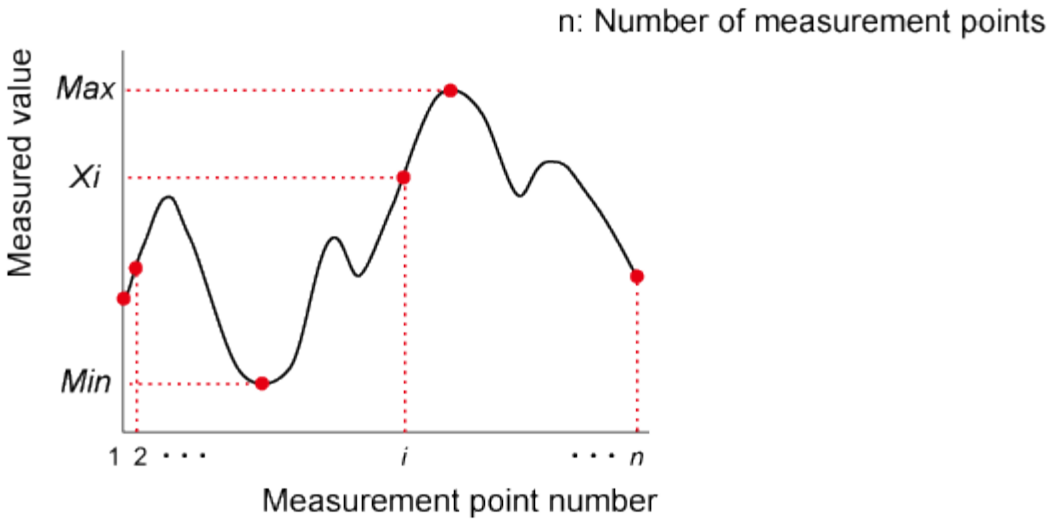
Determining the Mean, Standard Deviation, and p-p of the Trace

- Overview
  - Displaying Statistical Data

Other topics about Data Analysis

Overview

You can easily determine the statistics data for a trace (mean, standard deviation, and peak-to-peak). The definitions for the statistics data elements are shown below.



e5071c391

Statistics data element	Definition
Mean (*mean*)	$\frac{\sum_{i=1}^{n} x_i}{n}$ (n: number of points; x_i: measured value at the i-th measurement point)

Standard deviation (*s. dev*)	$\sqrt{\frac{\sum_{i=1}^{n}(x_i - mean)^2}{n-1}}$ (n: number of points; x_i: measured value at the i-th measurement point; mean: Mean)
Peak-to-peak (*p - p*)	Max - Min (Max: greatest measured value; Min: smallest measured value

Displaying Statistical Data

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace for which statistical data is required.
2. Press **Marker Fctn** key.
3. Click **Statistics** to turn on the display of statistics data.

Comparing Traces/Performing Data Math

- Overview
- Performing Data Math Operations

Other topics about Data Analysis

Overview

Each of the traces for which measured data is displayed is provided with an additional trace, called a memory trace, that temporarily stores measured data. You can use the memory trace to compare traces on the screen or to perform complex data math between the memory trace and measured data.

The following data math operations are available:

Value	Description
Data / Memory	Divides the measured data by the data in the memory trace. This function can be used to evaluate the ratio of two traces (e.g., evaluating gain or attenuation).
Data * Memory	Multiplies the measured data by a memory trace.
Data - Memory	Subtracts a memory trace from the measured data. This function can be used, for example, to subtract a vector error that has been measured and stored (e.g., directivity) from data subsequently measured on a device.
Data + Memory	Adds the measured data and the data in the memory trace.

Performing Data Math Operations

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace to be stored in memory.
2. Press **Display** key.
3. Press **Data -> Mem** to store the measured data in memory.
4. Press **Data Math**.
5. Select the data math operation to perform.
6. Press **Display**.
7. Select the type of data to display on the screen.
8. Send the trigger to make measurements.

Performing Parameter Conversion of Measurement Results

- Overview
- Procedure

Other topics about Data Analysis

Overview

You can use the parameter conversion function to convert the measurement results of the S-parameter (S_{ab}) to the following parameters.

- Equivalent impedance (Z_r) and equivalent admittance (Y_r) in reflection measurement

$$Z_r = Z_{0a} \times \frac{1 + S_{aa}}{1 - S_{aa}}, Y_r = \frac{1}{Z_r}$$

- Equivalent impedance (Z_t) and equivalent admittance (Y_t) in transmission measurement

$$Z_t = \frac{2 \times \sqrt{Z_{0a} \times Z_{0b}}}{S_{ab}} - (Z_{0a} + Z_{0b}), Y_t = \frac{1}{Z_t}$$

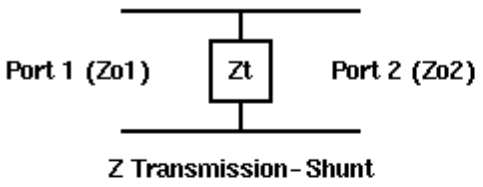
- Inverse S-parameter ($1/S_{ab}$)

where:

Z_{0a}: Characteristic impedance of port a

Z_{0b}: Characteristic impedance of port b

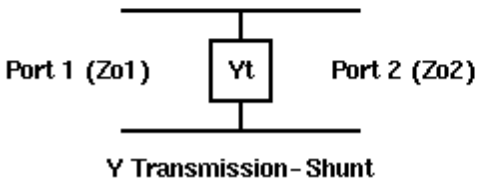
- Z/Y Transmission Shunt



$$Zt = \frac{1}{Yt}$$

$$Yt = \frac{2\sqrt{Yo1 \cdot Yo2}}{S} - (Yo1 + Yo2)$$

$$Yo1 = \frac{1}{Zo1} \quad Yo2 = \frac{1}{Zo2}$$



- Conjugation

  Conjugation converts the measurement value to complex conjugate number.

When the fixture simulator function is ON and the port impedance function is ON, the value set in the port impedance conversion is used. In other cases, the system Z0 (preset value: 50 ohm) is used.

Procedure

ON/OFF

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate a trace on which you want to use the marker.
2. Press **Analysis** key.
3. Click **Conversion**.
4. Click **Conversion** to turn ON the conversion function.

Selecting Conversion Target Parameter

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate a trace on which you want to use the marker.
2. Press **Analysis** key.
3. Click **Conversion** > **Function**.
4. Click the softkey corresponding to the parameter to which you want to convert the result.

When the conversion function is ON, the selected parameter is displayed in Trace Status Area.

Limit Test

- Overview
- Concept of Limit Test
- Displaying Judgement Result of Limit Test
- Defining Limit Line
- Limit line offset
- Initializing the limit table

Other topics about Data Analysis

Overview

The limit test feature allows you to set the limit line for each trace and then perform the pass/fail judgment for the measurement result.

Concept of Limit Test

The limit test is a function to perform pass/fail judgment based on the limit line you set with the limit table.

In the limit test, if the upper limit or lower limit indicated by the limit line is not exceeded, the judgment result is pass; if it is exceeded, the judgment result is fail for all measurement points on the trace. Measurement points in a stimulus range with no limit line are judged as pass.

- The targets of the pass/fail judgment are measurement points only. Parts interpolated between measurement points are not judged.

You define the limit line by specifying the stimulus value (Begin Stimulus) and response value (Begin Response) of the begin point, the stimulus value (End Stimulus) and response value (End Response) of the end point, and the type (lower limit/upper limit). For more information, refer to Defining the limit line.

When the limit test is on, measurement points that fail are displayed in red on the screen and the trace's pass/fail judgment result based on the results of individual measurement points (fail if one or more measurement points on the trace fail) is also displayed. You can check the pass/fail judgment result for the channel (fail if one or more traces fail in any of the limit test, the ripple test or the bandwidth test within the channel) on the screen as well. For more information, refer to Displaying judgment result of limit test.

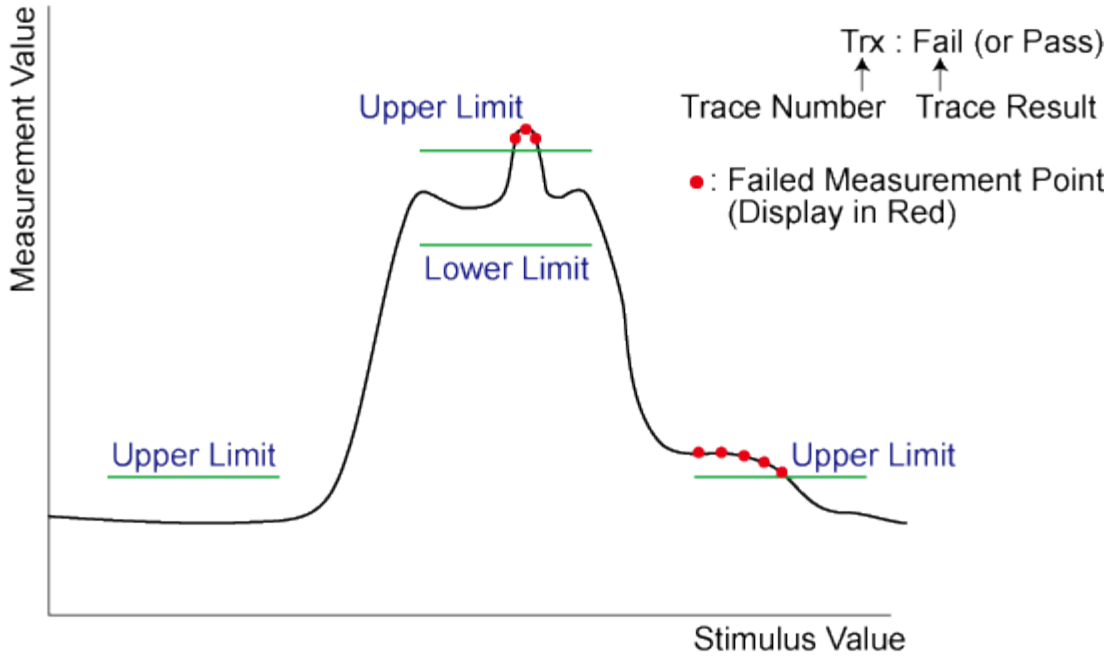
In addition to viewing the screen, you can check the judgment result of the limit test by the following methods.

- Beep that occurs when the judgment result is fail.
- Using the status register.

Displaying Judgement Result of Limit Test

Judgment result of measurement points and trace

Measurement points that fail are displayed in red on the screen. The judgment result of the trace is indicated by Pass or Fail displayed in the upper right part of the graph.



e5071c446

Judgment result of channels

If a channel has a judgment result of fail, the message below appears on the screen (it will be judged as failed if one or more unsatisfactory trace exists in any of the limit test, the ripple test or the bandwidth test within the channel.)

e5070bue065

Follow these steps to turn on/off the display of the channel fail message.

1. Press **Analysis** key.

2. Click **Limit Test**.
3. Click **Fail Sign**. Each press toggles between on/off.

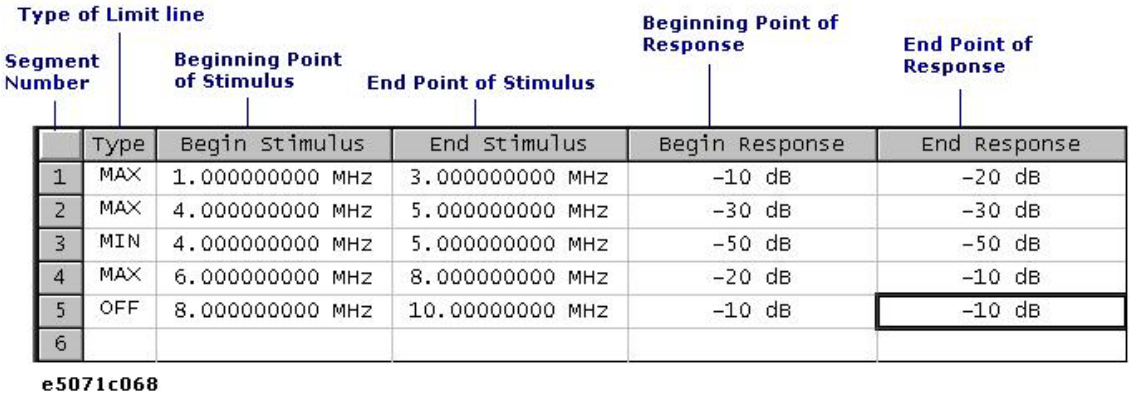
Defining Limit Line

To use the limit test, you must first define the limit line. You can define a limit table for each trace, and you can define up to 100 limit lines (segments) in a limit table.

Defining a segment

The following steps describe how to define a segment.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace on which the limit test function will be used.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Limit Test** to display the softkeys associated with the limit test.
4. Click **Edit Limit Line** to display the limit table.



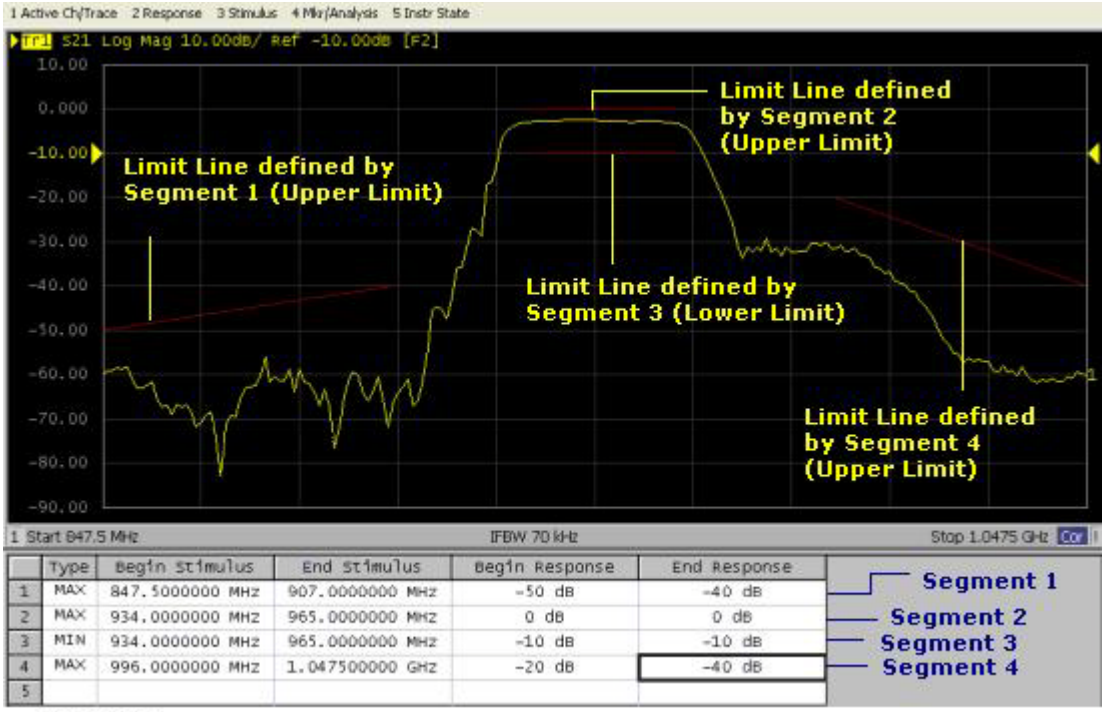
	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	1.000000000 MHz	3.000000000 MHz	-10 dB	-20 dB
2	MAX	4.000000000 MHz	5.000000000 MHz	-30 dB	-30 dB
3	MIN	4.000000000 MHz	5.000000000 MHz	-50 dB	-50 dB
4	MAX	6.000000000 MHz	8.000000000 MHz	-20 dB	-10 dB
5	OFF	8.000000000 MHz	10.00000000 MHz	-10 dB	-10 dB
6					

5. Using the limit table, create/edit a segment. Initially, no segments are entered in the limit table. At the same time, the Edit Limit Line menu used to create/edit the limit table is displayed.
6. Click **Add** to add a segment to the limit table and then specify the segment parameter values shown below.

Segment Parameter	Description
Type	Select the type of segment from the following: **OFF** Segment not used for the limit test **MIN** Segment at which the minimum is specified

	MAX Segment at which the maximum is specified
Begin Stimulus	Specify starting point for stimulus value on the limit line
End Stimulus	Specify ending point for stimulus value on the limit line
Begin Response	Specify starting point for response value on the limit line
End Response	Specify ending point for response value on the limit line

- The range in which stimulus values can be specified is from -500 G to +500 G. When a value outside the range is entered, a suitable value within the range is specified. Once the stimulus value is specified, changing the sweep range of the E5071C does not affect the stimulus value.
- The range in which response values can be specified is from -500 M to +500 M. When a value outside this range is entered, a suitable value within the range is specified. After the response value is specified, changing formats results in changing the units but not the value.



e5071c078

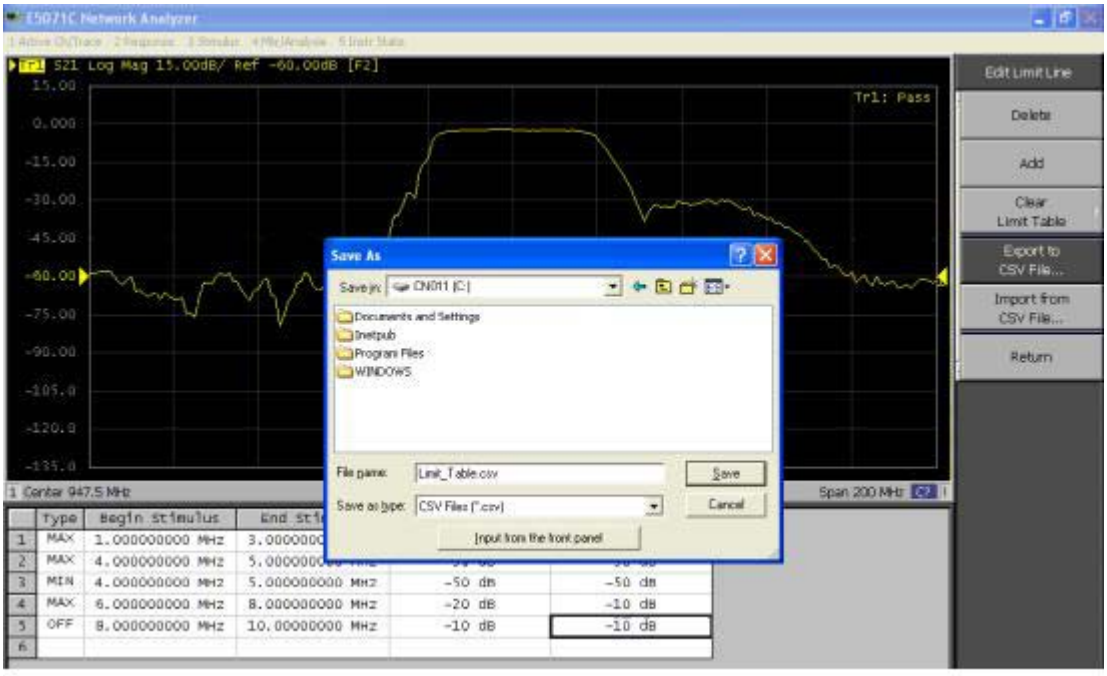
- You can define a limit line that is able to freely overlap the stimulus range of another limit line.

  Defining one limit line having the same type as a second limit line whose stimulus range overlaps with the first one results in two or more limit values at the same measurement point. In this case, the limit value to be used in the limit test is defined as follows:

  - When two or more limit values whose type is set to maximum (MAX) exist, the smallest one is used as the maximum.
  - When two or more limit values whose type is set to minimum (MIN) exist, the largest one is used as the minimum.
- Even if the span of the sweep range on the E5071C is set to 0, enter the two parameters of Begin Stimulus and End Stimulus.
- When two or more response values are returned as a result of using the Smith or polar chart format, the first response value of the marker provides the object of the limit test.

Saving/calling the limit table

You can save the limit table to a file that you can then freely bring up on the screen later and use. You can import a file saved in CSV format (extension: *.csv) into spreadsheet software on a PC for later use (a numerical value will be saved as strings that include its unit).



e5071c069

1. Display the limit table.
2. In the Edit Limit Line menu, press **Export to CSV File** to open the Save As dialog box. In this step, CSV (extension: *.csv) is selected as the file type
3. Specify the folder in which to save the file and enter the file name. Press **Save** to save the limit table displayed on the screen to the file.
4. Conversely, to recall a saved limit table, press **Import from CSV File** in the Edit Limit Line menu to display the Open dialog box. In this step, CSV (extension: *.csv) is selected as the file type.
5. After specifying the folder containing the file, select the file. Press **Open** to display the limit table on the screen.
6. The limit table can be called from any trace of any channel, regardless the channel or trace as of saving in a file.

Limit Table Saved in CSV Format

The limit table is saved in the following format.

- On the first line, the channel number of the active channel that was valid when the file was saved is output.
- On the second line, the trace number of the active trace that was valid when the file was saved is output.
- The third line provides the header showing the items for the segments to be output on the fourth and later lines.
- Data on segments are output on the fourth and later lines.

"# Channel 1"

"# Trace 1"

Type, Begin Stimulus, End Stimulus, Begin Response, End Response

MAX, 200.0000000 MHz, 400.0000000 MHz, -100 dB, -100 dB

MAX, 490.0000000 MHz, 510.0000000 MHz, -10 dB, -10 dB

MIN, 490.0000000 MHz, 510.0000000 MHz, -20 dB, -20 dB

MIN, 600.0000000 MHz, 800.0000000 MHz, -100 dB, -100 dB

Turning the limit test ON/OFF

You can set the limit test ON/OFF for each trace individually.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace on which the limit test function will be used.
2. Press **Analysis** key to display the Analysis menu.
3. Press **Limit Test** to display the Limit Test menu.

Softkey	Function
Limit Test	Sets the limit test ON/OFF.
Limit Line	Sets the limit line display ON/OFF.
Edit Limit Line	Opens the limit table for editing the limit line. See Defining_the_limit_line.

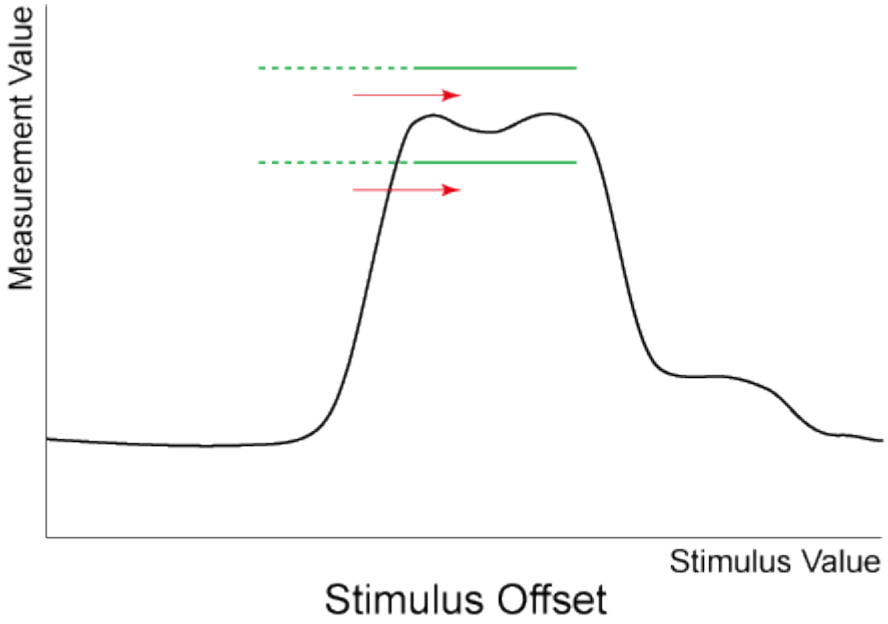
Limit line offset

By adding a certain offset to the limit value, you can adjust the limit line so that it conforms to the device output.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace on which the limit test function will be used.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Limit Test** to display the softkeys for the limit test.
4. Click **Limit Line Offsets** to display the limit line offset function menu. The following functions correspond to each softkey.

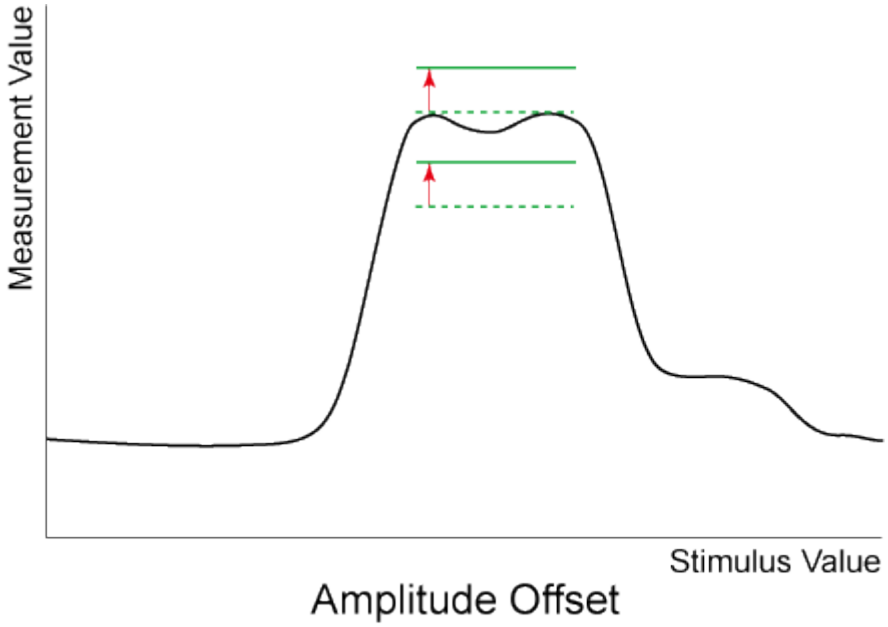
Softkey	Function
Stimulus Offset	Adds a certain offset to the stimulus value of the entire segment in the limit table. (Stimulus offset)
Amplitude Offset	Adds a certain offset to the response value of the entire segment in the limit table. (Amplitude offset)
Marker -> Amp. Ofs.	Adds the amplitude offset by the same amount as the retrieved value of the *active marker* . You can confirm the current value set for the amplitude offset by pressing **Amplitude Offset**. (Marker amplitude offset)

Stimulus offset



e5071c447

Amplitude offset



e5071c448

Initializing the limit table

The following operations initialize the limit table.

- At power-on
- When presetting
- When calling a limit table with zero segments
- When **Clear Limit Table** - **OK** is pressed in the Edit Limit Line menu

Bandwidth Test

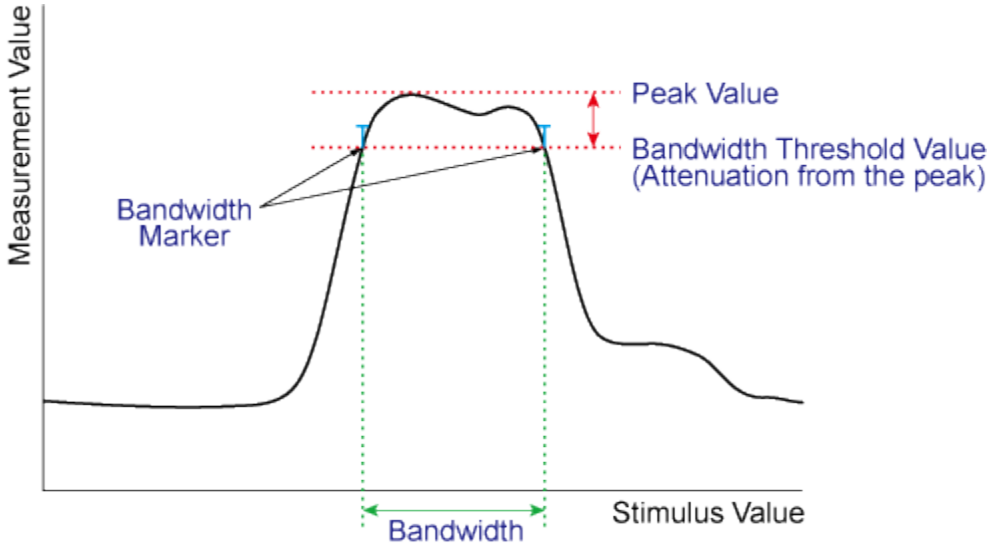
- Overview
- Displaying Bandwidth Test Results
- Bandwidth Test Setup
- Turning Bandwidth Test and Displaying Results ON/OFF

Other topics about Data Analysis

Overview

The bandwidth test function can be used for testing bandwidth for the band-pass filters.

The bandwidth test find the peak of a signal in the passband and locates a point on each side of the passband at an amplitude below the peak specified in test setup. The frequency between these two points is the bandwidth of the filter. Then the obtained bandwidth is compared to minimum and maximum allowable bandwidth that you specify beforehand.



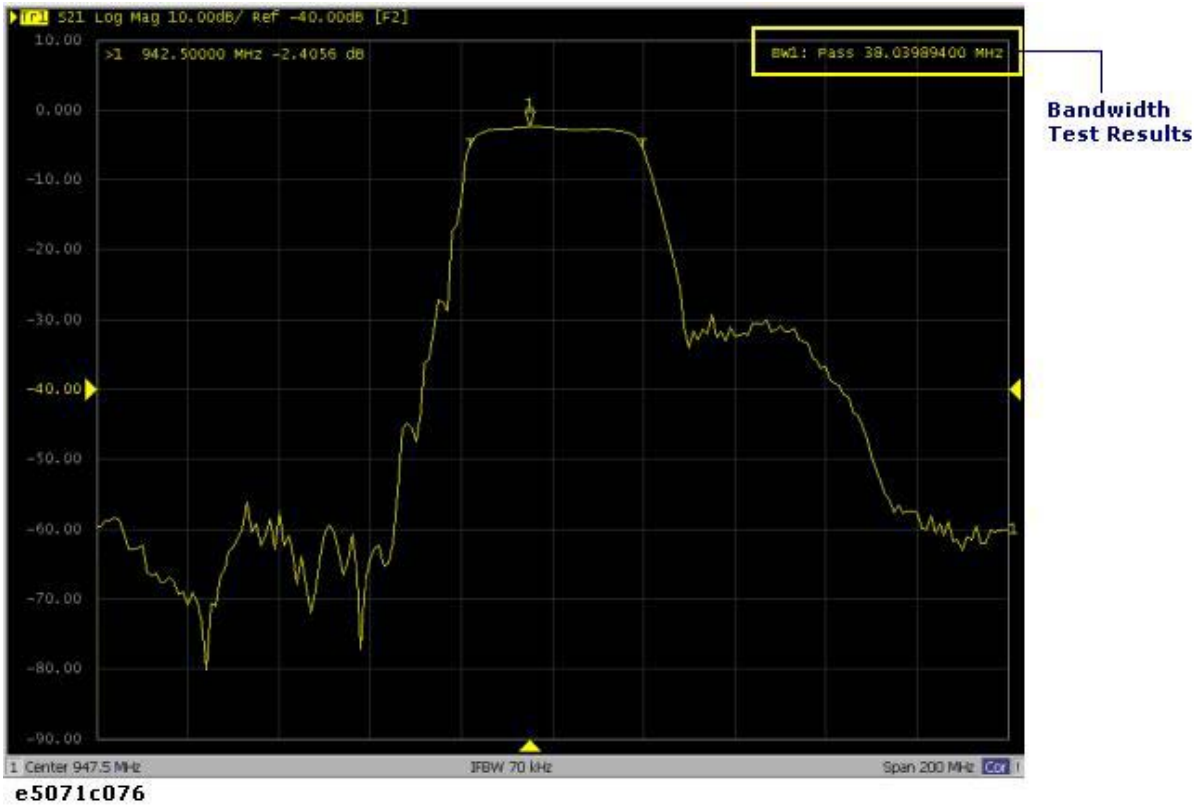
e5071c450

Displaying Bandwidth Test Results

Test Result for Trace

The test result of the trace will be indicated in the upper-right area of the graph for each trace, following BWn:. "n" denotes the trace number. The results are shown as Pass, Wide, Narrow, or >Span (Fail). You can also display the bandwidth value.

For information on how to display the results, see Turning On/Off Bandwidth Test and Displaying Results.



Test Results for Channel

If any channel is unsatisfactory, the message will be displayed as shown in Judgment result of channels. (It will be judged as failed if one or more failed traces are found for the limit test, ripple test, or bandwidth test within the channel.)

You can also specify this On/Off setting from the Fail Sign, which is provided in the limit test menu and ripple test menu. From the bandwidth test menu, follow the steps below to turn it on/off.

1. Press **Analysis** key.
2. Click **Bandwidth Limit**.
3. Click **Fail Sign**. This menu toggles between on and off.

In addition to the screen, the following features also let you confirm the test results:

- Beep notifying the result was unsatisfactory
- Status register

Bandwidth Test Setup

You must set up the bandwidth threshold and the upper and lower limits before you can use the bandwidth test function. You can specify the threshold, upper limit, and lower limit for each trace.

Operational procedure

Follow the steps below to set up the bandwidth test.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace to which you want to apply the bandwidth test function.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Bandwidth Limit** to display the softkeys for the bandwidth test.
4. Click **N dB Points** to specify the bandwidth threshold. The unit is dB.
5. Press **Min Bandwidth** to enter the lower limit for the bandwidth. Similarly, press **Max Bandwidth** to enter the upper limit for the bandwidth. The unit is Hz for both Min and Max bandwidths.
6. If the data format is Smith chart or polar, the limit test is performed for the main response value among the two marker response values.

Turning Bandwidth Test and Displaying Results ON / OFF

You can configure the on/off setting of the bandwidth test function for each trace independently.

Operational procedure

Follow the steps below to configure the on/off setting of the bandwidth test.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace Prev** keys to activate the trace to which you want to apply the bandwidth test function.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Bandwidth Limit** to display the softkeys for the bandwidth test.
4. Click **BW Test** and turn *ON* the bandwidth test. You can display the bandwidth marker on the screen by turning **BW Marker** *ON*. To display the bandwidth value, turn **BW Display** *ON*.

Ripple Test

- Overview
- Concept of Ripple Test
- Displaying Ripple Test Results
- Configuring Ripple Limit
- Saving/Recalling Ripple Limit Table
- Turning ON/OFF Ripple Test and Result Display
- Initializing Limit Table

Other topics about Data Analysis

Overview

Independently of the limit test, you can evaluate the measurement results on a pass/fail basis by setting a limit for the ripple. This function is called the ripple test.

Concept of Ripple Test

The ripple test is a function for evaluating the results on a pass/fail basis based on the ripple limit, which is set using the ripple limit table. You can specify up to 12 frequency bands, which permits a test for each frequency band.

The ripple test judges the measurement as "Pass" when the ripple value specified with the ripple limit is not exceeded by any of the measurement points on the trace; otherwise, it judges the measurement as "Fail." For the measurement points in a stimulus range without a specified ripple limit, the test judges the measurement as "Pass."

- The measurement point alone is the target of evaluation for pass/fail. The interpolated part between measurement points is not evaluated.

The ripple limit is defined with the start point stimulus value, end point stimulus value, ripple limit value, and type (on/off.) For detailed information, see Configuring ripple limit.

While the ripple test function is turned on, the measurement points corresponding to a "fail" judgment will be indicated in red on the screen, and the trace's test results based on the results of each measurement point will be displayed (judged as "fail" if one or more red measurement point exist on the trace). For information on how to display the results, see Turning on/off ripple test and displaying results. You can also confirm the channel test results on the screen (judged as "fail" if one or more failed traces appear in the limit test, ripple test, or bandwidth test within the channel).

Displaying Ripple Test Results

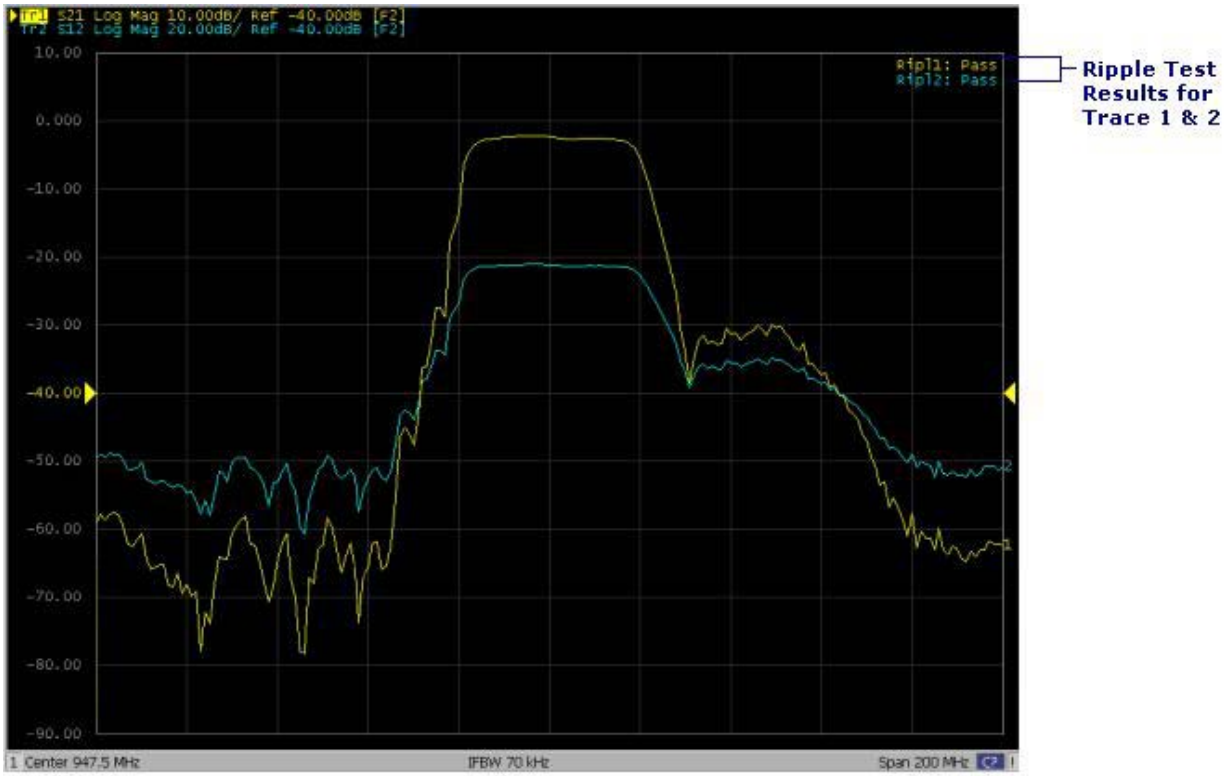
Measurement point and test results

Failed measurement points will be displayed in red on the screen. The test result for the trace will be indicated as Pass or Fail in the upper-right area of the graph. You can also display the ripple value at the selected frequency band.

The result will be displayed as Ripln:Pass (or Fail) for each trace. n denotes the trace number. Bn will be followed by the ripple value (if the ripple display is turned off, only Bn will be displayed without the ripple value).

For example, in the following figure, Ripl1:Pass in the first line indicates the result for trace 1. The value following B3 is the ripple value at the third frequency band specified in the ripple test. Similarly, the second line indicates the test result for trace 2, showing the ripple value at the first frequency band.

For information on how to display the results, see Turning on/off ripple test and displaying results.



Test result for channel

If a channel has a judgment result of "fail," the message will be displayed as shown in Test Result for Trace. (It will be judged as failed if one or more failed traces are found in the limit test, ripple test, or bandwidth test within the channel.)

e5071c071

You can also specify the On/Off setting for the Fail Sign in the limit test menu and as well as in the bandwidth test menu. From the ripple test menu, follow the steps below to turn it on/off.

1. Press **Analysis** key.
2. Click **Ripple Limit**.
3. Click **Fail Sign**. This menu toggles between on and off.

In addition to the screen, the following features also let you confirm the test results:

- Beep notifying that the result was "fail"
- Status register

Configuring Ripple Limit

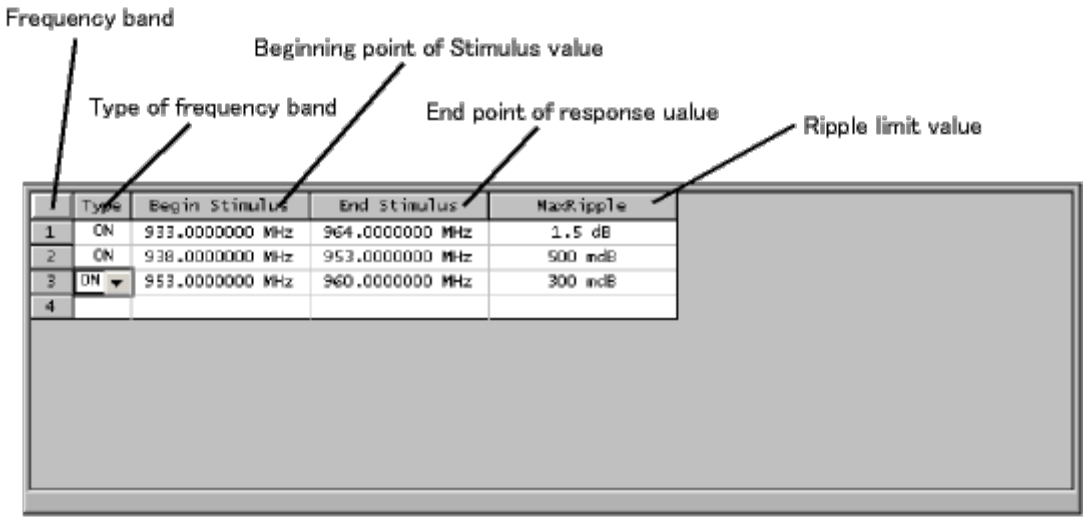
You must configure the ripple limit before you can use the ripple test function. You can specify a ripple limit table for each trace, where up to 12 ripple limit bands (frequency bands) can be configured.

Operational procedure

Follow the steps below to configure the ripple limits.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace on which you want to apply the ripple test function.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Ripple Limit** to display the softkeys for to the ripple test.

4. Click **Edit Ripple Limit** to display the ripple limit table shown below.



e5070bue7018

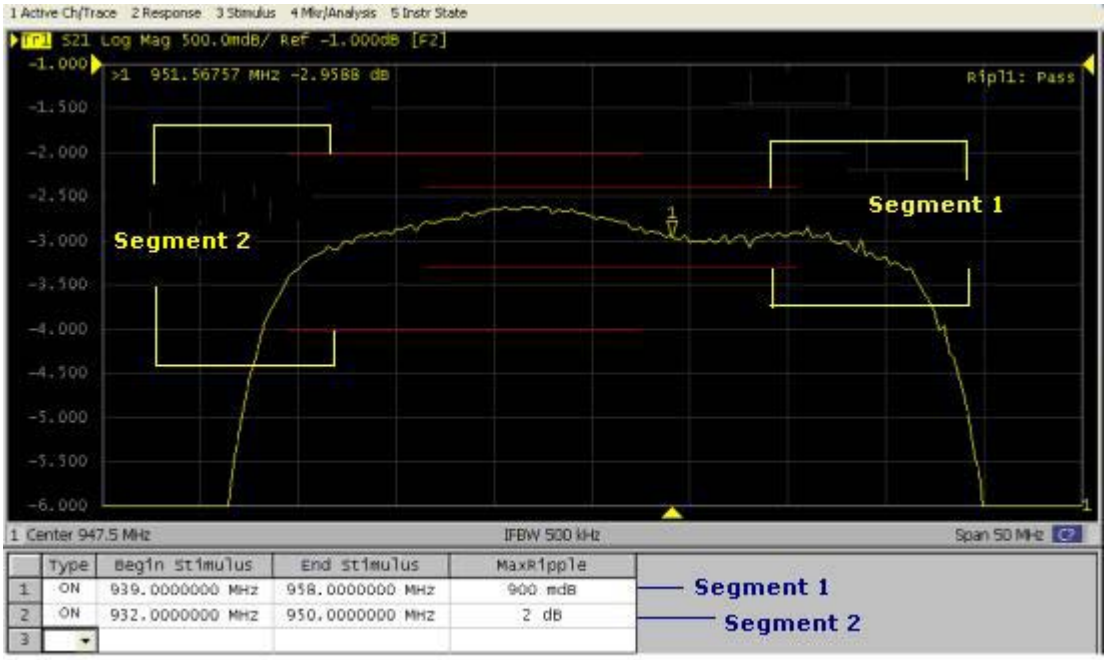
5. Create or edit the bands by using softkeys. Note that no frequency band is provided in the ripple limit table by default. The Edit Ripple Limit menu is also displayed, which can be used for creating or editing the frequency band.
6. Click **Add** to add a frequency band to the ripple limit table and then specify the following parameters for the frequency band:

Parameter	Description
Type	Selects a frequency type, either ON or OFF. **ON** Band used for the ripple test. **OFF** Band not used for the ripple test.
Begin Stimulus	Specifies the start point for the stimulus value in the ripple test.
End Stimulus	Specifies the end point for the stimulus value in the ripple test.
Max Ripple	Specifies the ripple limit value.

- Acceptable range for the stimulus value: -500G to +500G. If any outranging value is specified, it will be reset to fall within the range.

- Even if the E5071C's sweep range is changed after the stimulus value has been set, the stimulus value is not susceptible.

Example of ripple limit configuration



e5071c079

- The individual frequency bands for the ripple test can overlap each other; in this case, the ripple limit test is performed for each frequency band.
- Even if the E5071C's span value is set to zero, you must enter a parameter for both Begin Stimulus and End Stimulus.
- If the data format is Smith chart or polar, the limit test is performed for the main response value among the two marker response values.

Saving/Recalling Ripple Limit Table

The ripple limit table can be saved in a file and recalled later for use on the screen. The file is saved in the csv format (with the extension *.csv), and values are saved as a character string with the unit. The csv formatted file can also be reused in spreadsheet software made for PCs.

Follow the steps below to save/recall the ripple limit table. This operation should be done by using the external keyboard and/or mouse.

1. Display the ripple limit table.
2. Click **Export to CSV File** from the Edit Ripple Limit menu to open the dialog box. At this time, CSV Files (with the extension *.csv) is selected as the file type.

3. Specify any folder in which you want to save the file, and enter the file name. Click **Save** to save the ripple limit table displayed on the screen to a file.
4. To recall the saved ripple limit table, click **Import from CSV File** from the Edit RippleLimit menu to display the Open dialog box. At this time, CSV Files (with the extension *.csv) is selected as the file type.
5. Specify the folder that contains the file, and then select the file. Click **Open** to recall the saved limit table on the screen.
6. You can recall a limit table from a trace on any channel independently of the channel and trace that were active when the limit table was saved to the file.

The ripple limit table is saved in the following format:

- In the first line, the channel number for the active channel at the time of file saving will be output.
- In the second line, the trace number for the active trace at the time of file saving will be output.
- The third line is a header indicating the segment items that are output from the fourth line onward.
- From the fourth line onward, the segment data are output.

Operational procedure

"# Channel 1"

"# Trace 1"

Type, Begin Stimulus, End Stimulus, MaxRipple

ON, 933.0000000 MHz, 964.0000000 MHz, 1.5 dB

ON, 938.0000000 MHz, 953.0000000 MHz, 500 mdB

ON, 953.0000000 MHz, 960.0000000 MHz, 300 mdB

Turning ON/OFF Ripple Test and Result Display

You can set the limit test ON/OFF for each trace individually.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace on which you want to apply the ripple test function.
2. Press **Analysis** key to display the Analysis menu.
3. Click **Ripple Limit** to display the softkeys for to the ripple test.
4. Click the following corresponding softkeys.

Softkey	Function
Ripple	Sets the ripple test ON/OFF.

Test	
Ripple Limit	Sets the ripple limit line display ON/OFF.
Ripple Value	Sets how the ripple values are displayed. Available settings are off, absolute value (difference between maximum and minimum values within the band) display, and margin (difference between absolute value of ripple and ripple limit) display.
Ripple Value Band	Selects the band for which you want to display the ripple value.
Edit Ripple Limit	Opens the ripple limit table for editing the ripple limit. To use the ripple test function, you must first define the ripple limit.

Initializing Limit Table

The following operations initialize the limit table.

- At power-on
- When presetting
- When calling a Ripple table with zero segments
- When **Clear Limit Table** > **OK** is clicked in the Edit Ripple Line menu

Equation Editor

- Overview
- Using Equation Editor
- Equation Editor Examples
- Equation History
- Functions and Constants
- Converting S Parameters to H, Y, Z, F & T Parameters
- Operators used in Equation Editor

Other topics about Data Analysis

Overview

Equation Editor allows you to enter an algebraic equation of standard mathematical operators and functions, referencing data that is available in the E5071C. Once a valid equation is entered and enabled, the display of the active trace is replaced with the results of the equation, and updated in real-time as new data is acquired. For equations that can be expressed with Equation Editor's supported functions, operators, and data, there is no need for off-line processing in a separate program.

For example, on entering the equation "Example=S21/(1-S11)" in the E5071C Equation Editor (**4** in the Figure below), the resulting trace is computed as each S21 data point divided by one minus the corresponding S11 data point. For a 201 point sweep setup, the computation is repeated 201 times, once for each point.

As another example, if you want the ENA to make a directivity measurement of your 3-port DUT, then the desired result is the sum and difference of LogMag formatted traces, which can be expressed as: S12 + S23 - S13.

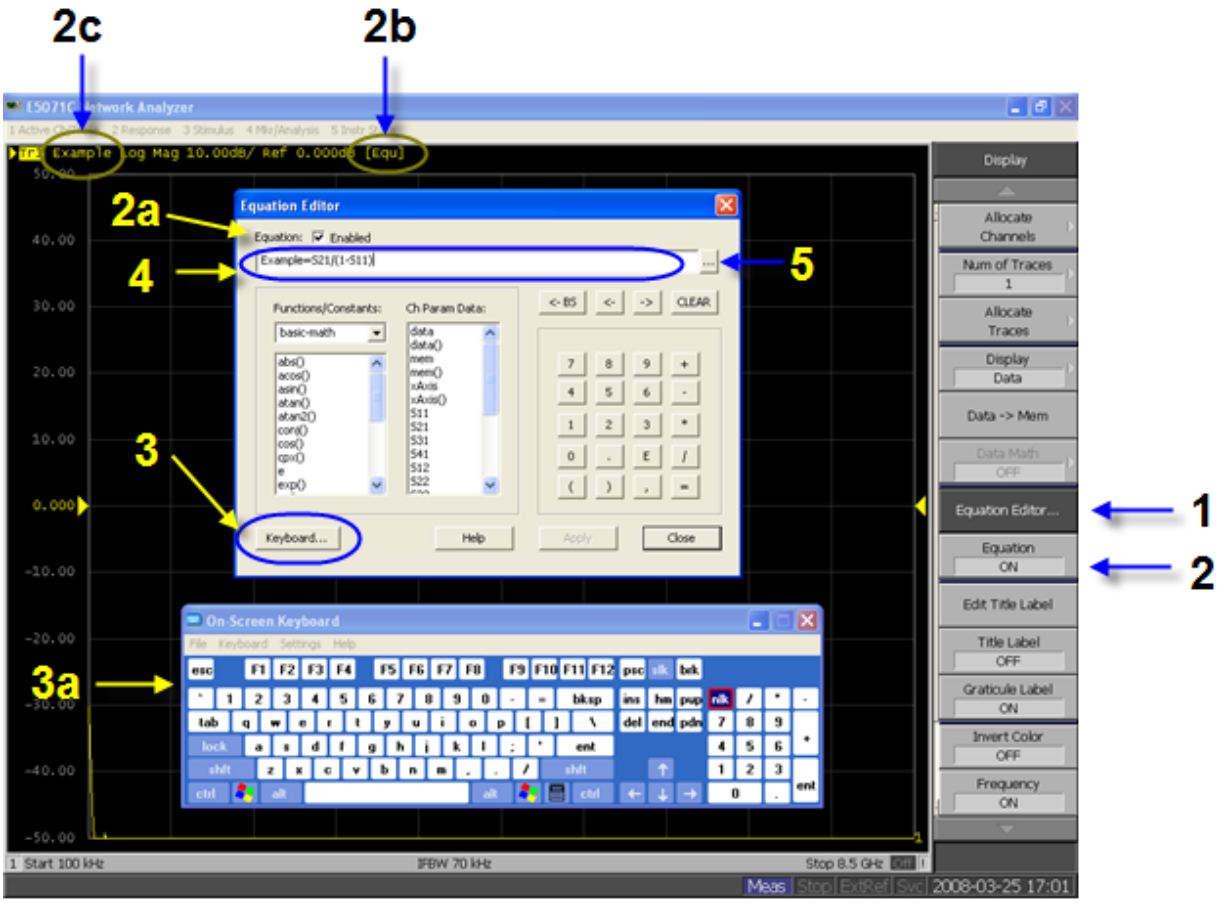
Because Equation Editor operates on unformatted complex data, the required equation is:

DIR = S12 * S23 / S13

DIR becomes a display label to help you identify the computed data trace. On the equation trace, set the format to LogMag.

Using Equation Editor

The step-by-step procedure of using Equation Editor is described below:



e5071c282

1. Select a trace in which you want to enter the equation and activate the trace.
    2. Activating a trace is required as Equation Editor works on traces.
2. Follow the steps below to enter an equation:
    1. Press **Display**.
    2. Click **Equation Editor** (**1** in the figure above). The Equation Editor dialog box appears.
    3. Enter an equation in the equation field (**4** in the figure above).

- Referring to traces in a different channel is NOT available with Equation Editor on the ENA.
- The equation can be entered with the software keyboard enabled by selecting **Keyboard...** (**3** and **3a** in the figure above).

3. Follow the steps below to apply the defined equation. When a valid equation is entered, the Equation Enabled check box becomes available for checking.
    1. Check **Equation Enabled** check box (**2a** in the figure above).
    2. Click **Apply**. The equation becomes visible and annotation of [Equ] (**2b** in the figure above) is displayed in the trace title area.
    3. Click **Close** to hide the dialog box.
    4. The equation can also be applied by selecting **Display** > **Equation ON**.
    5. If error correction is not turned on, then the raw, uncorrected data is used in the equation trace.
    6. If an equation is NOT valid (i.e. referring to a trace that is not measured in the measured channel), annotation of [Equ!] is displayed in the trace title area instead.

Equation Editor Examples

The following examples may help you in getting started with Equation Editor:

Offset each data point in trace 2 from trace 1 by XdB

The equation is entered into trace 2 as "**Offset= data(1)*pow(10,X/20)**" (**4** in the figure e5071c282).

To multiply a trace by X-times in log format

The equation is entered as "**Example_Trace= pow(data,X)**" (**4** in the figure e5071c282).

Differential Return Loss using a 2-port ENA

The equation is entered as "**Sdd11= (S11-S21-S12+S22)/2)**" (**4** in the figure e5071c282).

- For detailed on mixed-mode S-parameters on the 2-port ENA, refer to the related FAQ on the Agilent website.

CMRR of a Balanced Component

The equation is entered as "**CMRR = data(2) / data (1)**" (**4** in the figure e5071c282)

where:

data (1) - a trace that measures Scc21 with Fixture Simulator

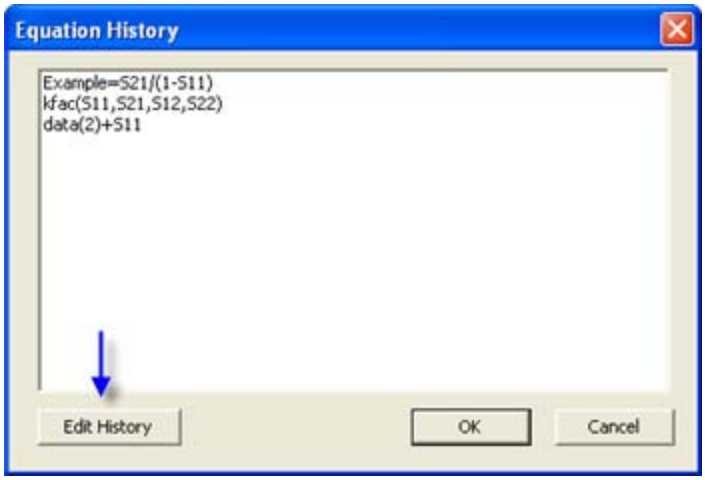
data (2) - a trace that measures Sdd21 with Fixture Simulator

Equation History

Equation Editor has the capability to save and recall all previously defined equations, which can be accessed by clicking the **...** button (shown as **5** in the figure e5071c282). All equations can be viewed in the **Equation History** dialog box.

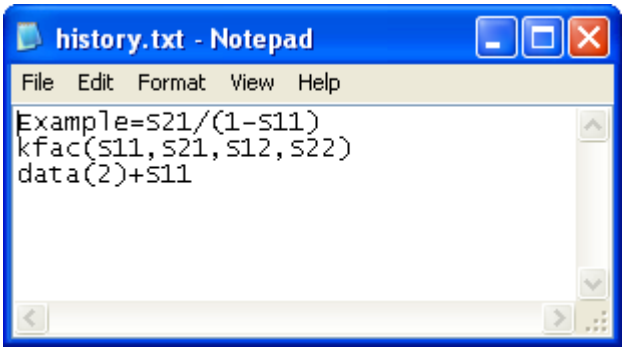
To view the equations in the list, follow this procedure:

1. Open Equation Editor by **Display** > **Equation Editor**
2. Enter an equation and click **Apply** in the Equation Editor dialog box (Figure e5071c282) to save the defined equation in the directory of the ENA. To view a list of saved equations, click the ... button (**5** in Figure e5071c282) to open the Equation History dialog box.



3. To store an equation in the History List, the equation must be applied first. This can be done by clicking on the **Apply** button.

3. To edit the equations in the list, click **Edit History**. The text file of **history list** is opened with Notepad.



4. The **History** List is stored as a text file **D:\Agilent\Equation\history.txt** and can save a maximum of 50 lines (equations) with a maximum of 254 character per line (equation).

Functions and Constants

The following table describes the different functions and constant available in the E5071C Equation Editor. In the following table:

- Function(scalar x) means that the function requires a scalar value. If a complex value is entered, it is automatically converted to a scalar value; complex(x,y) -> scalar(x)
- Function(complex x) means that the function requires a complex value. If a scalar value is entered, it is automatically converted to a complex value; scalar(x) -> complex(x, 0)
- **a,b** are arguments that are used in the function.

Basic Math Functions

Function	Description
abs(complex **a**)	returns the sqrt(a.re*a.re+a.im*a.im)
acos(scalar **a**)	returns the arc cosine of a in radians
asin(scalar **a**)	returns the arc sine of a in radians
atan(scalar **a**)	returns the arc tangent of a in radians
atan2(complex **a**)	returns the phase of a = (re, im) in radians
atan2(scalar **a**, scalar **b**)	returns the phase of (a, b) in radians
conj(complex **a**)	returns the conjugate of a
cos(complex **a**)	takes a in radians and returns the cosine

cpx(scalar **a**, scalar **b**)	returns a complex value (a+ib) from two scalar values
exp(complex **a**)	returns the exponential of a
im(complex **a**)	returns the imaginary part of a as the scalar part of the result (zeroes the imaginary part)
ln(complex **a**)	returns the natural logarithm of a
log10(complex **a**)	returns the base 10 logarithm of a
mag(complex **a**)	returns sqrt(a.re*a.re+a.im*a.im)
phase(complex **a**)	returns atan2(a) in degrees
pow(complex **a**,complex **b**)	returns a to the power b
re(complex **a**)	returns the scalar part of a (zeroes the imaginary part)
sin(complex **a**)	takes a in radians and returns the sine
sqrt(complex **a**)	returns the square root of a, with phase angle in the half-open interval (-pi/2, pi/2]
tan(complex **a**)	takes a in radians and returns the tangent
Constants	
e	2.71828182845904523536
PI	3.14159265358979323846

- Mutual transformation is automatically made for scalar and complex.

  scalar(x) -> complex(x, 0)
  complex(x, y) -> scalar(x)

Advanced Math Functions

In the following table: **a**,**b**,**c**,**d** are arguments of complex value that are used in the function. For 2-port network measurement: **a**,**b**,**c**,**d** corresponds to Sii,Sji, Sij, Sjj respectively.

- The functions can also be defined by scalar arguments with port numbers of the ENA. For example, the function, kfac(1,2)

returns the K-factor of 2-port measurement between port 1 and port 2 of the ENA.

- When a scalar argument of the port number in the function is out of range (i.e. mu1(1,3) for the 2-port ENA), the ENA indicates an error message, "Equation runtime error" and equation is not reflected on the trace.

Z0 means the system characteristic impedance (Z0) which can be accessed under Cal.

- Although the port impedance for each port can be changed with the port impedance conversion function by Fixture Simulator, the change is not reflected on the computed trace by Equation Editor.

Function	Description
A(complex **a**, complex **b**, complex **c**, complex **d**)	F_{11} conversion = ((1+a)*(1-d) + b*c)/(2*b)
A(scalar i, scalar j)	returns A(Sii, Sji, Sij, Sjj)
B(complex **a**, complex **b**, complex **c**, complex **d**)	F_{12} conversion = Z0*((1+a)*(1+d) - b*c)/(2*b)
B(scalar i, scalar j)	returns B(Sii, Sji, Sij, Sjj)
C(complex **a**, complex **b**, complex **c**, complex **d**)	F_{21} conversion = (1/Z0)*((1-a)*(1-d) - b*c)/(2*b)
C(scalar i, scalar j)	returns C(Sii, Sji, Sij, Sjj)
D(complex **a**, complex **b**, complex **c**, complex **d**)	F_{22} conversion = ((1-a)*(1+d) + b*c)/(2*b)
D(scalar i, scalar j)	returns D(Sii, Sji, Sij, Sjj)
H11(complex **a**, complex **b**, complex **c**, complex **d**)	H_{11} conversion = Z0*((1+a)*(1+d) - b*c)/((1-a)*(1+d) + b*c)
H11(scalar i, scalar j)	returns H11(Sii, Sji, Sij, Sjj)
H12(complex **a**, complex **b**, complex **c**, complex **d**)	H_{12} conversion = 2*c/((1-a)*(1+d) + b*c)
H12(scalar i, scalar j)	returns H12(Sii, Sji, Sij, Sjj)

H21(complex **a**, complex **b**, complex **c**, complex **d**)	H_{21} conversion = -2*b/((1-a)*(1+d) + b*c)
H21(scalar i, scalar j)	returns H21(Sii, Sji, Sij, Sjj)
H22(complex **a**, complex **b**, complex **c**, complex **d**)	H_{22} conversion = (1/Z0)*((1-a)*(1-d) - b*c)/((1-a)*(1+d) + b*c)
H22(scalar i, scalar j)	returns H22(Sii, Sji, Sij, Sjj)
kfac(complex **a**, complex **b**, complex **c**, complex **d**)	k-factor = (1 - abs(a)^2 - abs(d)^2 + (abs(a*d - b*c)^2)/(2*abs(b*c))
kfac(scalar i, scalar j)	returns kfac(Sii, Sji, Sij, Sjj)
MAPG(complex **a**, complex **b**, complex **c**, complex **d**)	maximum available power gain = abs(b/c)*(kfac(a,b,c,d) - sqrt(kfac(a,b,c,d)^2 - 1))
MAPG(scalar i, scalar j)	returns MAPG(Sii, Sji, Sij, Sjj)
MSG(complex **a**, complex **b**, complex **c**, complex **d**)	maximum stable power gain = abs(b)/abs(c)
MSG(scalar i, scalar j)	returns MSG(Sii, Sji, Sij, Sjj)
mu1(complex **a**, complex **b**, complex **c**, complex **d**)	μ-factor = (1 - abs(a)^2) / (abs(d - conj(a) * (a*d-b*c)) + abs(b*c))
mu1(scalar i, scalar j)	returns mu1(Sii, Sji, Sij, Sjj)
mu2(complex **a**, complex **b**, complex **c**, complex **d**)	μ-factor = (1 - abs(d)^2) / (abs(a - conj(d) * (a*d-b*c)) + abs(b*c))
mu2(scalar i, scalar j)	returns mu2(Sii, Sji, Sij, Sjj)
T11(complex **a**, complex **b**, complex **c**, complex **d**)	T_{11} conversion = -(a*d - b*c)/b
T11(scalar i, scalar j)	returns T11(Sii, Sji, Sij, Sjj)
T12(complex **a**, complex **b**, complex **c**, complex **d**)	T_{12} conversion = a/b
T12(scalar i, scalar j)	returns T12(Sii, Sji, Sij, Sjj)

T21(complex **a**, complex **b**, complex **c**, complex **d**)	T_{21} conversion = -d/b
T21(scalar i, scalar j)	returns T21(Sii, Sji, Sij, Sjj)
T22(complex **a**, complex **b**, complex **c**, complex **d**)	T_{22} conversion = 1/b
T22(scalar i, scalar j)	returns T22(Sii, Sji, Sij, Sjj)
Y11(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{11} conversion = (1/Z0)*((1-a)*(1+d) + b*c)/((1+a)*(1+d) - b*c)
Y11(scalar i, scalar j)	returns Y11(Sii, Sji, Sij, Sjj)
Y12(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{12} conversion = (1/Z0)*(-2*c)/((1+a)*(1+d) - b*c)
Y12(scalar i, scalar j)	returns Y12(Sii, Sji, Sij, Sjj)
Y21(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{21} conversion = (1/Z0)*(-2*b)/((1+a)*(1+d) - b*c)
Y21(scalar i, scalar j)	returns Y21(Sii, Sji, Sij, Sjj)
Y22(complex **a**, complex **b**, complex **c**, complex **d**)	Y_{22} conversion = (1/Z0)*((1+a)*(1-d) + b*c)/((1+a)*(1+d) - b*c)
Y22(scalar i, scalar j)	returns Y22(Sii, Sji, Sij, Sjj)
Z11(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{11} conversion = Z0*((1+a)*(1-d) + b*c)/((1-a)*(1-d) - b*c)
Z11(scalar i, scalar j)	returns Z11(Sii, Sji, Sij, Sjj)
Z12(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{12} conversion = Z0*(2*c)/((1-a)*(1-d) - b*c)
Z12(scalar i, scalar j)	returns Z12(Sii, Sji, Sij, Sjj)
Z21(complex **a**, complex **b**, complex **c**, complex **d**)	Z_{21} conversion = Z0*(2*b)/((1-a)*(1-d) - b*c)
Z21(scalar i, scalar j)	returns Z21(Sii, Sji, Sij, Sjj)
Z22(complex **a**, complex **b**,	Z_{22} conversion = Z0*((1-a)*(1+d) + b*c)/((1-

complex **c**, complex **d**)	a)*(1-d) - b*c)
Z22(scalar i, scalar j)	returns Z22(Sii, Sji, Sij, Sjj)

1. For both mu1 and mu2, conj is the complex conjugate. For scalars a and b, conj(a+ib)=(a-ib).

Channel Parameter Data	
data	corrected data
data(scalar i)	corrected data of trace i (trace number)
mem	memory data
mem(scalar i)	memory data of trace i (trace number)
xAxis	x-axis data
xAxis(scalar i)	x-axis data of trace i (trace number)
S11 - S44	S-parameter data

- When a trace number is out of range, the ENA indicates an error message, "Equation runtime error" and equation is not reflected on the trace. The maximum number of traces can be set by **System** > **Misc Setup** > **Channel/Trace Setup**.

Converting S-Parameters to H, Y, Z, F & T-Parameters

The following section provides definition of the two port parameters in Equation Editor of the ENA.

1. **Hybrid parameters (H)**

$$\begin{pmatrix} V_1 \\ I_2 \end{pmatrix} = (H)\begin{pmatrix} I_1 \\ V_2 \end{pmatrix} \qquad (H) = \begin{pmatrix} H_{11} & H_{12} \\ H_{21} & H_{22} \end{pmatrix}$$

2. **Admittance parameters (Y)**

$$\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} = (Y)\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} \qquad (Y) = \begin{pmatrix} Y_{11} & Y_{12} \\ Y_{21} & Y_{22} \end{pmatrix}$$

3. Impedance parameters (Z)

$$\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = (Z)\begin{pmatrix} I_1 \\ I_2 \end{pmatrix} \quad (Z) = \begin{pmatrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{pmatrix}$$

Fundamental parameters (F)

$$\begin{pmatrix} V_1 \\ I_1 \end{pmatrix} = (F)\begin{pmatrix} V_2 \\ -I_2 \end{pmatrix} \quad (F) = \begin{pmatrix} A & B \\ C & D \end{pmatrix}$$

4.

$$\begin{pmatrix} b_1 \\ b_2 \end{pmatrix} = (S)\begin{pmatrix} a_1 \\ a_2 \end{pmatrix} \quad (S) = \begin{pmatrix} S_{11} & S_{12} \\ S_{21} & S_{22} \end{pmatrix}$$

$$a_1 = \frac{V_1 + I_1 Z_0}{2 \cdot \sqrt{Z_0}} \qquad a_2 = \frac{V_2 + I_2 Z_0}{2 \cdot \sqrt{Z_0}}$$

$$b_1 = \frac{V_1 - I_1 Z_0}{2 \cdot \sqrt{Z_0}} \qquad b_2 = \frac{V_2 - I_2 Z_0}{2 \cdot \sqrt{Z_0}}$$

5. Scattering transfer parameters (T)

$$\begin{pmatrix} b_1 \\ a_1 \end{pmatrix} = (T)\begin{pmatrix} a_2 \\ b_2 \end{pmatrix} \quad (T) = \begin{pmatrix} T_{11} & T_{12} \\ T_{21} & T_{22} \end{pmatrix}$$

S-parameters of 2-port network can be converted to Hybrid parameters (H), Admittance parameters (Y), Impedance parameters (Z), Fundamental parameters (F) and Scattering transfer parameters (T) using the following functions:

1. **Converting S-parameters to H-parameters**

$$H_{11} = Z_0 \cdot \frac{(1+S_{11})(1+S_{22}) - S_{12}S_{21}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

$$H_{12} = \frac{2 \cdot S_{12}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

$$H_{21} = \frac{-2 \cdot S_{21}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

$$H_{22} = \frac{1}{Z_0} \cdot \frac{(1-S_{11})(1-S_{22}) - S_{12}S_{21}}{(1-S_{11})(1+S_{22}) + S_{12}S_{21}}$$

2. Converting S-parameters to Y-parameters

$$Y_{11} = \frac{1}{Z_0} \cdot \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Y_{12} = \frac{1}{Z_0} \cdot \frac{-2 \cdot S_{12}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Y_{21} = \frac{1}{Z_0} \cdot \frac{-2 \cdot S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

$$Y_{22} = \frac{1}{Z_0} \cdot \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1+S_{11})(1+S_{22})-S_{12}S_{21}}$$

3. Converting S-parameters to Z-parameters

$$Z_{11} = Z_0 \cdot \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

$$Z_{12} = Z_0 \cdot \frac{2 \cdot S_{12}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

$$Z_{21} = Z_0 \cdot \frac{2 \cdot S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

$$Z_{22} = Z_0 \cdot \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{(1-S_{11})(1-S_{22})-S_{12}S_{21}}$$

4. Converting S-parameters to F-parameters

$$A = \frac{(1+S_{11})(1-S_{22})+S_{12}S_{21}}{2\cdot S_{21}}$$

$$B = Z_0 \cdot \frac{(1+S_{11})(1+S_{22})-S_{12}S_{21}}{2\cdot S_{21}}$$

$$C = \frac{1}{Z_0} \cdot \frac{(1-S_{11})(1-S_{22})-S_{12}S_{21}}{2\cdot S_{21}}$$

$$D = \frac{(1-S_{11})(1+S_{22})+S_{12}S_{21}}{2\cdot S_{21}}$$

5. Converting S-parameters to T-parameters

$$T_{11} = \frac{-D_S}{S_{21}}$$

$$T_{12} = \frac{S_{11}}{S_{21}}$$

$$T_{21} = \frac{-S_{22}}{S_{21}}$$

$$T_{22} = \frac{1}{S_{21}}$$

Where:

$D_s = S_{11}S_{22} - S_{12}S_{21}$

Operators used in Equation Editor

Operator	Description
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Power
(	Open parenthesis
)	Close parenthesis
,	Comma - separator for arguments
=	Equal (optional)
E	Exponent (as in 23.45E6)

1. Priority of operators is:

    1. ^
    2. *,/
    3. +,-

Data Output

Data Output

- Saving and Recalling Instrument State
- Saving/Recalling Instrument State for Each Channel into/from Memory
- Saving Trace Data to a File
- Saving the Screen Image to a File
- Printing Displayed Screen

Saving and Recalling Instrument State

- Overview
- Compatibility of Files (Saving and Recalling)
- Saving Data
- Recall Procedure
- Recall Procedure Using "Recall by File Name"
- Priority of Recalling Configuration File at Startup

Other topics about Data Output

Overview

You can save the instrument state of the E5071C into a file on mass storage and then recall it later to reproduce that state. You can select the stored data from the following four types.

Type	Stored data and usage
State only (State Only)	Saves the setting of the E5071C and reproduces the state when it was saved by recalling it later into the E5071C.
State and calibration data (State & Cal)	Saves the setting of the E5071C and calibration data (calibration coefficient array) to reproduce the state when it was saved by recalling it later into the E5071C. At this time, you can perform error correction of measured values by using the recalled calibration data.
State and trace (State & Trace)	Saves the setting of the E5071C and traces (error-corrected data array and error-corrected memory array) to reproduce the state when it was saved by recalling it later into the E5071C. At this time, the traces are also recalled and displayed on the screen.
State, calibration data, and traces (All)	Saves the setting of the E5071C, calibration data, and traces to reproduce the state when it was saved by recalling it later into the E5071C. At this time, the calibration data and traces are also recalled.

In addition, the "user-preset" function is provided to allow the user to freely set up an instrument state recalled when the preset function is executed.

Compatibility of Files (Saving and Recalling)

Files saved with the E5071C cannot be recalled with the E507xA/B. The other compatibility of saving/recalling the instrument state file is as follows:

Compatibility between different frequency models

			Recalling by E5071C									
	Model	Opt.No.	230430	235435	240440	245445	260460	265465	280480	285485	2D54D5	2K54K5
Saved File by	E5071C	230430	Y	N	Y	N	Y	N	Y	N	N	N
		235435	Y	Y	Y	Y	Y	Y	Y	Y	N	N
		240440	N	N	Y	N	Y	N	Y	N	N	N
		245445	N	N	Y	Y	Y	Y	Y	Y	N	N
		260460	N	N	N	N	Y	N	Y	N	N	N

		2654465	N	N	N	N	Y	Y	Y	Y	N	N
		2804480	N	N	N	N	N	N	Y	N	N	N
		2854485	N	N	N	N	N	N	Y	Y	N	N
		2D54D5	N	N	N	N	N	N	N	N	Y	Y
		2K54K5	N	N	N	N	N	N	N	N	N	Y
	E5070A/B		Y	Y	Y	Y	Y	Y	Y	Y	N	N
	E5071A/B		N	N	N	N	N	N	Y	Y	N	N

Y: Recall is possible, N: Recall is impossible

- There is no compatibility of saving/recalling the instrument state between options 2D5, 4D5, 2K5 and 4K5 and ENA-B/Other ENA-C Option.
- When state files with calibration data that are saved with E5070A/B and E5071A/B, only states and trace data are recalled.

Compatibility between different number of ports models

- If the stored data is "state only (**State Only**)," files saved with a model having a smaller number of ports can be recalled with a model having a larger number of ports, but the opposite is not possible.

		Recalling

	Option: number of port	2 Ports	4 Ports
Saved File by	2 Ports	Y	Y/ c
	4 Ports	N	Y

Y: Recall is possible

N: Recall is impossible

Y/ c: Only when the stored data is "state only (**State Only**)," recall is possible.

Compatibility when the maximum number of channels/traces is different

- You cannot recall files saved by specifying all channels/traces (**All**) as the save target.
- If the number of channels/traces at save does not exceed that at recall, you can recall files saved by specifying the displayed channel/trace (**Disp Only**) as the save target.

Compatibility when the Firmware revision is different

- Files that are saved with later firmware revisions cannot be recalled by prior firmware revisions.
- Recalling the state file which is saved by prior firmware revision may cause a sweep time change and C? is displayed. In this case, re-calibration is recommended.

Compatibility when the system spec. version (available with ":SERV:SREV?") is different

- When Files that are saved with different system spec. versions include calibration data, only states and trace data are recalled.
    - If you recall an incompatible file, an error occurs and the device recovers to the presetting.

Saving Data

Selecting Content to be Saved

- This setting takes effect both when saving the entire instrument state into a file and when saving the instrument state for each channel into memory.

1. Press **Save/Recall** key.
2. Click **Save Type**.
3. Click the softkey corresponding to the content of the instrument state you want to save.

Selecting Save Target Channel/Trace

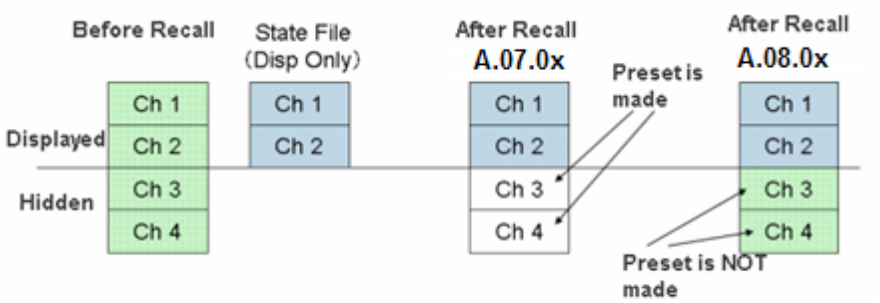
1. Press **Save/Recall** key.

2. Click **Channel/Trace** and select the save target from all channels/traces (**All**) or displayed channel/traces only (**Disp Only**).

If you specify the displayed channel/traces only as the save target, you can reduce the file size. However, for channels/traces that are not displayed, you cannot recall and reproduce the instrument state separately held for each channel/trace at a later time.

Faster Recall

From Firmware revision A.08.0x or later, the recall speed has been increased in the **DISP only** mode. In Firmware revision below A.08.0x, preset was done for hidden channels. In Firmware revision A.08.0x or later, hidden channels are not made preset and results in a faster recall.



e5071c281

Saving Instrument State

Follow the procedure below to save internal data from the E5071C.

1. Press **Save/Recall** key.
2. Click **Save State**.
3. Click the softkey corresponding to the destination you want to save.

Softkey	Description
State01 to **State08**	Save the instrument state into the state number.
Autorec	Save the instrument state as the auto recall setting. The E5071C is automatically configured with this state at the startup. This key saves the state into the "**D:Autorec.sta**". When **Autorec.sta** file is found on the D: drive at startup, the E5071C is automatically configured using the saved settings. To disable the auto recall function, delete the

	Autorec.sta files.
User Pres	Save the instrument state as user preset. The user can preset the analyzer at user saved status.
File Dialog...	Save the instrument state as your desired file name. You can enter a file name using the **Input from the front panel** buttons on the dialog box when storing a file.

- If **D:\Autorec.sta** is found on the system at startup, the E5071C is automatically configured using the saved settings. When the external floppy disk drive is connected as A: drive, then if **A:\Autorec.sta** is found at startup, the E5071C is also automatically configured using the saved settings. If both files are found, **A:\Autorec.sta** is recalled. To disable the auto recall function, delete the **Autorec.sta** files.
- An asterisk (*) in the upper right of the softkey indicates that the corresponding file of the softkey already exists. If you save into the existing file, the existing file is copied as **backup.sta** and then overwritten.

Recall Procedure

Follow the procedure below to recall internal data from the E5071C.

1. If you recall a file that includes traces (its content was set to **State &Trace** or **All** when it was saved), the trigger source is automatically set to Manual.

1. Press **Save/Recall** key.
2. Click **Recall State**.

When you want to recall State01.sta - State08.sta, Autorec.sta

1. Press **State01** - **State08** or **Autorec**.

When you want to recall other files

1. Press **File Dialog...** to open the Open dialog box.
2. Select the folder and the file using the external keyboard and mouse.
3. Click **Open**.

The following warning messages may appear when recall fails:

Warning Message	Note
Model Mismatch	This message is displayed when the state file of E5070B/E5071B is recalled by E5071C options 2D5, 4D5, 2K5 and 4K5 .

Unsupported Version	This message is displayed when the state file generated from a newer version of Firmware is recalled.
Invalid No. of Ports	This message is displayed when the state file generated from a Firmware having different Port Options is recalled.
Source Attenuator Option Mismatch	This message is displayed when the state file generated from a Firmware having different Source Attenuator Option is recalled.
Power Sweep Option Mismatch	This message is displayed when the state file generated from a Firmware having different Power Sweep Option is recalled.
Invalid No. of Points	This message is displayed when the Maximum No. of Points are not enough.
External Test Set Option Mismatch	This message is displayed when the state file generated from a Firmware having different External Test Set Option is recalled.
Frequency Option Mismatch	This message is displayed when the state file generated from a Firmware having different Frequency Option is recalled.
ch x, tr y, z pts required	This message is displayed when the Recall Function cannot recall data with maximum channel or traces. The number of channels/traces should be higher than the recall state file to recover from the error. More information is available in Setting upper limit of channels and traces.

- The warning message listed above:
  1. May not appear in all failures
  2. Appears on the screen only when recall is done from soft key by Recall State or Recall by File Name.
  3. Displays "Recall Failed" once when the recall is done through GPIB or VBA.
  4. Is not kept in Error queue. This means that when SCPI.SYSTem.ERRor is executed, "Recall Failed" is returned.
  5. These messages are supported for state files created in Firmware revision A.09.10 or later.
- When a user file is used in Extending the Calibration Plane Using Network De-embedding, Determining Characteristics After Adding a Matching Circuit, or Determining the Characteristics that Result from Adding a Matching Circuit to a Differential Port and the setup status is saved, a recall error will occur if the user

file is not located in the same folder as when the state was saved.

- Pressing **Save/Recall** > **Explorer** executes Windows Explorer. This helps you to browse the files in the ENA hard disk drive.

Recall Procedure Using "Recall by File Name" Feature

You can use the recall feature with the **Recall by File Name** softkey for files you have named freely and saved in the D:\State folder. This function lets you recall a file you have named freely and saved by simple softkey operation, eliminating annoying operation using the Open dialog box.

- Although there is no limit to the number of files saved in a folder, only up to 50 files are displayed on the softkeys. If more than 50 files are saved in a folder, they are sorted in the order of numbers 0 to 9 and alphabetic characters A to Z and the first 50 files are displayed as softkeys.

  Although there is no limit to the number of characters of a file name, only up to 12 characters are displayed on the softkey. If a file name exceeds 12 characters, the first 12 characters are displayed on the softkey and the remaining characters are omitted and replaced with "...".

- Different files may be displayed on softkeys with the same name or a saved file is not displayed on any softkey because of the above limitations.

1. Press **Save/Recall** key
2. Click **Recall by File Name**.
3. Files that have been named and saved in the D:\State folder are displayed on softkeys. Press the key for the file you want to recall.

Priority of Recalling Configuration File at Startup

If several instrument configuration files exist at the startup of the E5071C, only one file is recalled and set at a time in the following order of priority.

If these files do not exist, the normal preset (factory preset) is executed.

Priority	Recalled file
1	Configuration file for the auto-recall function in the A drive (If external floppy disk drive is connected.)
2	Configuration file for the auto-recall function in the D drive
3	Configuration file for the user-preset function in the D drive.

	Executed when the preset operation mode is User and the file (**D:\UserPreset.sta**) exists.

Saving/Recalling Instrument State for Each Channel into/from Memory

- Overview
- Saving Instrument State for Each Channel
- Recalling Instrument State for Each Channel
- Deleting Saved Instrument State (Clearing all Registers)

Other topics about Data Output

Overview

The E5071C allows you to save/recall the instrument state for each channel independently. This function allows you to save the instrument state of the active channel independently into one of four registers (A to D, volatile memory) and to recall the instrument state from the register to restore it as the state of the currently active channel. As in the case of saving the entire state of the instrument into a file, you can select items to be saved from four kinds.

Since you can recall the instrument state for each channel that was saved with this function from a different channel that the one used to save it, this function is very useful for copying an instrument state between channels.

- Unlike when saving the entire instrument state, the instrument state for each channel is saved into volatile memory instead of a file, so if you turn off the power, this state is lost.

Saving Instrument State for Each Channel

1. Press **Channel Next**/**Channel Prev** keys to activate a channel whose state you want to save.
2. Press **Save/Recall** key.
3. Click **Save Channel**.
4. Click one of **State A** to **State D** to save the instrument state of the active channel to the specified register.
5. For registers having saved data, the * symbol is displayed to the right of their softkey label. If you specify one of these, its content is overwritten.

Recalling Instrument State for Each Channel

1. Press **Channel Next**/**Channel Prev** keys to activate a channel whose state you want to recall and restore.
2. Press **Save/Recall** key.
3. Click **Recall Channel**.

4. Click the softkey of the register in which the state you want to restore is saved. This instrument state is recalled to the active channel.

Deleting Saved Instrument State (Clearing all Registers)

1. Press **Save/Recall** key.
2. Click **Save Channel**.
3. Click **Clear States**. The contents of all the registers are deleted.

Saving Trace Data to a File

- Saving Data in CSV Format
- Saving Data in Touchstone Format

Other topics about Data Output

Saving Data in CSV Format

The E5071C allows the user to save data for the active trace on the active channel to a CSV file (file extension *.csv) and to load the data into PC application software for further processing.

Trace data are saved in the format shown below.

Example of saved trace data

```
"# Channel 1"
"# Trace 1"
Frequency,              Formatted Data,          Formatted Data
+3.00000000000E+005, +1.41837599227E-002, +1.43446459328E-006
+4.27985000000E+007, +1.41275293412E-002, +2.02407834551E-004
+8.52970000000E+007, +1.41334093048E-002, +4.00643331604E-004
+1.27795500000E+008, +1.41240661092E-002, +6.09250514670E-004
+1.70294000000E+008, +1.41402155348E-002, +8.05620003993E-004
```

The first line shows the number of the active channel at the time the data was saved.

The second line shows the number of the active trace at the time the data was saved.

The third line is a header line indicating the contents of each item of trace data written on the fourth line onward.

The fourth line onward shows the trace data. The amount of data is determined by the number of points (frequency) assigned to the trace.

Saving Trace Data

Follow the procedure below to save trace data from the E5071C.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace to be saved.
2. Press **Save/Recall** key.
3. Click **Save Trace Data** to open the Save As dialog box.
4. Select the destination folder and input a file name.
5. Click **Save** to save the file.

Saving Data in Touchstone Format

You can also save trace data of a E5071C active channel to a Touchstone format file, based on 1- to 4-port models.

Touchstone file data format

You can save data in "log magnitude - angle", "linear magnitude - angle", or "real number - imaginary number."

When AUTO is selected, the data format is automatically set according to the display format of the active trace. However, when the display format of the active trace is set to one other than log magnitude format (LogMag), linear magnitude format (LinMag), or real-imaginary number format (Real/Imag), the data format is automatically set to real-imaginary number.

You can use data saved in Touchstone format for a circuit simulator such as Agilent Advanced Design System (ADS) on your PC (personal computer) or workstation. For more information on the ADS, refer to the operation manual that comes with the system.

NOTE The fixture simulator function enables you to recall to the E5071C the s2p (2-port Touchstone file) file type used to save user-defined matching circuit information. However, no other type of file saved in Touchstone format can be recalled to the E5071C.

NOTE The port reference impedance value when outputs the touchstone format file has to be the same impedance value (real) to every measurement port.

NOTE You can output the measurement result when using complex port impedance conversion, or when setting different port impedance at each port. However, the touchstone file is valid only when the same impedance value (real) to every measurement port is used.
When you recall the file saved with above condition, the file is recalled as single normalized reference Z_0 defined at option line, and it provides in-consistency result.

File types of Touchstone files

File types of the E5071C Touchstone files are s1p, s2p, s3p, and s4p. The file type indicates the number of ports of the data structure that is output to the Touchstone file.

Data structure in Touchstone file

Data structure of the Touchstone file consists of a header part and a data part. The contents of the file is text data, which is ready to be read with a general text editor.

The header part consists of the comment line and option line.

- Comment line is preceded by an exclamation mark (!), and includes returned value of *IDN?, file created date, calibration state, list of all S parameters of a specified port, and port impedance setting when saving the data.
- Option line includes format information.

The header parts of s1p and s4p are shown below.

Header of s1p

Header of s1p

```
!Agilent Technologies,E5071C,<ID>,<FW Revision>
!Date <Date>
!Data & Calibration Information
!Freq Sww:Method(Stat)
!PortZ PORT-w:R+jX
!Above PortZ is port Z conversion or system Z0 setting when saving the data.
!When reading, reference impedance value at option line is always used"
# Hz S FMT R Z0
```

Header of s4p

```
!Agilent Technologies,E5071C,<ID>,<FW Revision>
!Date <Date>
!Data & Calibration Information
!Freq Sww:Method(Stat) Sxw:Method(Stat) Syw:Method(Stat) Szw:Method(Stat)
.
.
Swz:Method(Stat) Sxz:Method(Stat) Syz:Method(Stat) Szz:Method(Stat)
!PortZ PORT-w:R+jX      Port-x:R+jX      Port-y:R+jX      Port-z:R+jX
!Above PortZ is port Z conversion or system Z0 setting when saving the data.
!When reading, reference impedance value at option line is always used"
# Hz S FMT R Z0
```

Comment line

Touchstone data files include comments. Comments are preceded by an exclamation mark (!).

Each item has the following meaning:

Parameter	Description
Sww to Szz	S parameters of the selected test port; corresponds in ascending order, beginning with w to z.
Method	Calibration type applied to S parameter.
Stat	State of S parameter calibration and error correction

	setting (ON, OFF, or --) ON = Error correction is set to ON OFF =Error correction is set to OFF -- = Calibration is not performed
PORT-w to PORT-z:	Single-end port impedance setting when saving the data. This is not valid when reading the file. When reading, the reference impedance value Z0 at option line is always used.

Option Line

The option line is formatted as follows.

Hz S FMT R Z0

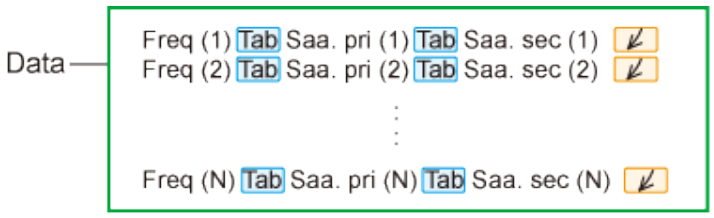
Parameter	Description
FMT	Data format RI = Real number- imaginary number MA = Linear magnitude - angle DB = Log magnitude - angle
Z0	Reference impedance value

The structure of the data part depends on the combination of the selected file type and specified port.

NOTE When the S parameter of a port on which calibration is not performed is specified, if data measured with the S parameter exists, that data are output to a Touchstone file. If no measurement data exists, 0 (for log magnitude - angle, log magnitude = -200 dB) is output in the corresponding field.

The following figures show the data structures of files saved in Touchstone format.

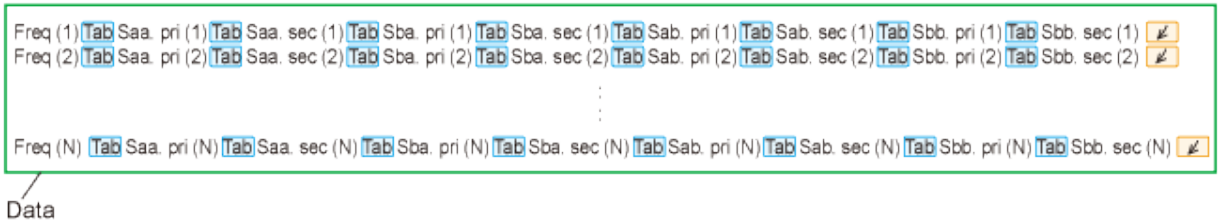
1-port Touchstone file



a : Selected test port number
Freq(n) : Frequency at measurement point n [Hz]
Saa. pri(n) : Real part(RI), linear magnitude(MA) or dB(DB) of measured parameter Saa at measurement point n
Saa. sec(n) : Imaginary part(RI) or phase(MA,DB) of measured parameter Saa at measurement point n
N : Number of measurement points
Tab : Tab
: Line break

e5071c346

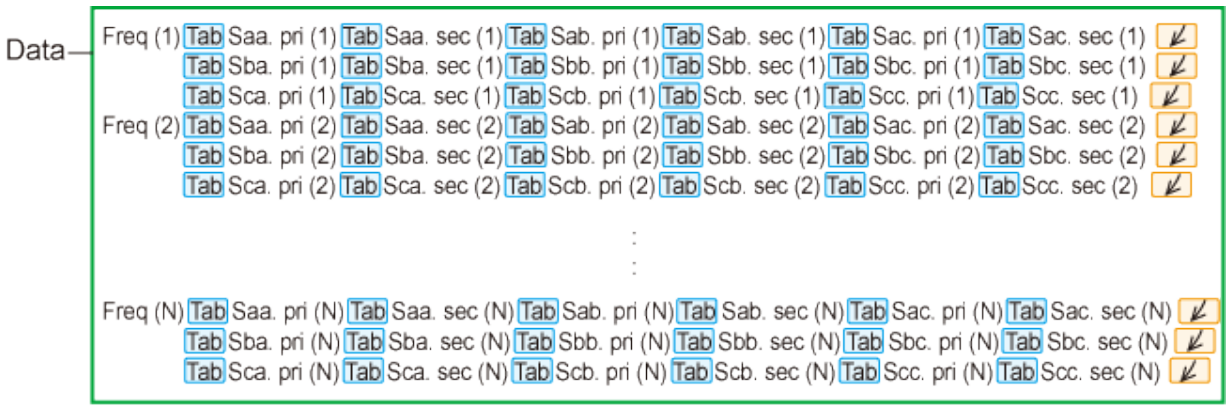
2-port Touchstone file



a - b : Selected test port number (corresponding in ascending order, beginning with 1 to a)
Freq(n) : Frequency at measurement point n [Hz]
Sxy. pri(n) : Real part(RI), linear magnitude(MA) or dB(DB) of measured parameter Sxy at measurement point n
Sxy. sec(n) : Imaginary part(RI) or phase(MA,DB) of measured parameter Sxy at measurement point n
N : Number of measurement points
Tab : Tab
: Line break

e5071c347

3-port Touchstone file



a - c : Selected test port number (corresponding in ascending order, beginning with 1 to a)
Freq(n) : Frequency at measurement point n [Hz]
Sxy. pri(n) : Real part(RI), linear magnitude(MA) or dB(DB) of measured parameter Sxy at measurement point n
Sxy. sec(n) : Imaginary part(RI) or phase(MA,DB) of measured parameter Sxy at measurement point n
N : Number of measurement points
Tab : Tab
: Line break

e5071c348

4-port Touchstone file

Freq(n) : Frequency at measurement point n [Hz]
Sxy. pri(n) : Real part(RI), linear magnitude(MA) or dB(DB) of measured parameter Sxy at measurement point n
Sxy. sec(n) : Imaginary part(RI) or phase(MA,DB) of measured parameter Sxy at measurement point n
N : Number of measurement points
Tab : Tab
: Line break

e5071c349

Restrictions when saving data in Touchstone format

The following restrictions apply when saving measurement data into Touchstone format.

- When both fixture simulation and port impedance conversion are on, all Z0 of the ports to be saved must be set to the same value. If Z0 is different among the ports, no error occurs, but only the Z0 of the smallest port number is output to the header.
- When the time domain function is on, the saved data are not the displayed data but the data of the S parameter before conversion.
- For data saved in touchstone format, data operation, time domain, parameter conversion, data format, electrical delay, equation editor, and smoothing are not reflected in the output data.
- An error occurs when attempting to save data that use the frequency offset function.

Saving procedure

Follow the steps below to save trace data in Touchstone format.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to select the trace to be saved.
2. Press **Save/Recall** key.
3. Click **SnP** > **SnP Format**.
4. Click the softkey that corresponds to the data format you want to save.

Softkey	Function
Auto	Data format is automatically set according to the display format of the active trace. When the display format of the active trace is set to one other than log magnitude format (LogMag), linear magnitude format (LinMag), or real-imaginary number format (Real/Imag), the data format is automatically set to real-imaginary number.
LogMag/Angle	Select "log magnitude - angle" data format
LinMag/Angle	Select "linear magnitude - angle" data format
Real/Imaginary	Select "real - imaginary number" data format

5. Click the softkey for **s1p**, **s2p**, **s3p**, or **s4p** according to the file type you want to save.
6. Click the softkey that corresponds to the combination of ports that you want to save. For example, when pressing **s3p** in Step5, softkeys

displayed are **1-2-3**, **1-2-4**, **1-3-4**, and **2-3-4**. If you want to save the data of the channel for which port 1, port 2, and port 4 are measured, press **1-2-4**.

7. SaveAs dialog box opens. For its operations, use an external keyboard and mouse.
8. Specify the folder to which the file should be saved, enter a file name, and then press **Save** to save the file.
9. When saving data in a sweep process, the data during sweep is saved into a Touchstone file. That is, the previous sweep data is saved as data that has not been swept; or, if sweep was not performed previously, zero data might be saved. Therefore, you should set the active channel to the HOLD state when saving data into a Touchstone file.

Saving the Screen Image to a File

- Overview
- Saving Screen Image as File

Other topics about Data Output

Overview

Along with printing, the E5071C allows the user to save screen images as bitmap (.bmp) or portable network graphics (.png) files. Saved files can be loaded into PC application software for further processing.

Saving Screen Image as File

Follow the procedure below to save a screen image to a file.

1. Display the screen to be saved as a file. If you want to save the screen with a white background, set the display mode to inverted display before saving the screen. For details about display mode, see Setting Display Colors.
2. Press **System** key. The screen image at the time **System** key is pressed is the image that will be saved.
3. Press **Dump Screen Image** to open the Save As dialog box. "Bitmap Files (*.bmp)" or "Portable Network Graphics (*.png)" can be selected as the file type.
4. Select the file type.
5. Select the destination folder and type a file name.
6. Press **Save** to save the screen image of E5071C to a file.

Printing Displayed Screen

- Overview
- Printed/Saved Images
- Print Procedure

Other topics about Data Output

Overview

By connecting a printer to the USB port of the E5071C, you can print the displayed screen of the E5071C.

Printed/Saved Images

The display image saved in the volatile memory (clipboard) is printed/saved. If no image is saved in the clipboard, the image displayed at the time of print execution is printed/saved.

Saving image to clipboard

The **System** key also has a screen capture feature. When you press **System** key, the image displayed on the screen immediately before pressing is saved in the clipboard.

- The image in the clipboard is cleared when you execute print/save.

Print Procedure

Preparation before printing

Follow these steps to prepare for printing:

1. Turns off the E5071C.
2. Turn on the printer and connect it to E5071C.
3. Turn on the E5071C.
4. Press **System** key.
5. Press **Printer Setup**. The Printers window opens. The icons of the printers that have been connected are displayed in the window. When you connect a print for the first time, it is automatically registered and its icon is added in the window.
6. The printer with the check mark (✔) on its icon is selected as the default printer for printing. If you want to change it, select (highlight) the icon of your preferred printer in the Printers window and then click **Set as Default Printer** in the File menu.
7. Click **Printing Preferences...** in the File menu. The Printing Preferences dialog box for the selected printer appears. Set items necessary before printing such as Page Size and then click the **OK** button .

8. Click **Close** in the File menu.

Executing print

Follow these steps to print the screen information:

1. Display the screen you want to print.
2. Press **System** key to save the currently displayed screen onto the clipboard.
3. As necessary, press **Invert Image** to toggle between [OFF] for printing in colors close to the actually displayed screen and [ON] for printing in inverse colors.
4. Click **Print** to start printing.
5. To cancel the printing in progress, press **Abort Printing**.
   - If you start printing when the printer is not ready (for example, it is not turned on) by mistake, the Printers Folder dialog box may appear. In this case, click **Cancel** to close the Printers Folder dialog box, prepare your printer, and then start printing again.

Optimizing Measurements

Optimizing Measurements

- Expanding Dynamic Range
- Reducing Trace Noise
- Improving Phase Measurement Accuracy
- Improving Measurement Throughput
- Performing Segment-by-Segment Sweep (segment sweep)

Expanding Dynamic Range

- Overview
- Lowering Receiver Noise Floor

Other topics about Optimizing Measurement

Overview

The dynamic range is the finite difference between the maximum input power level and the minimum measurement power level (noise floor) of the analyzer. In evaluating a characteristic accompanied by a large change in the amplitude (the pass band and stop band of a filter, for example), it is important to increase the dynamic range.

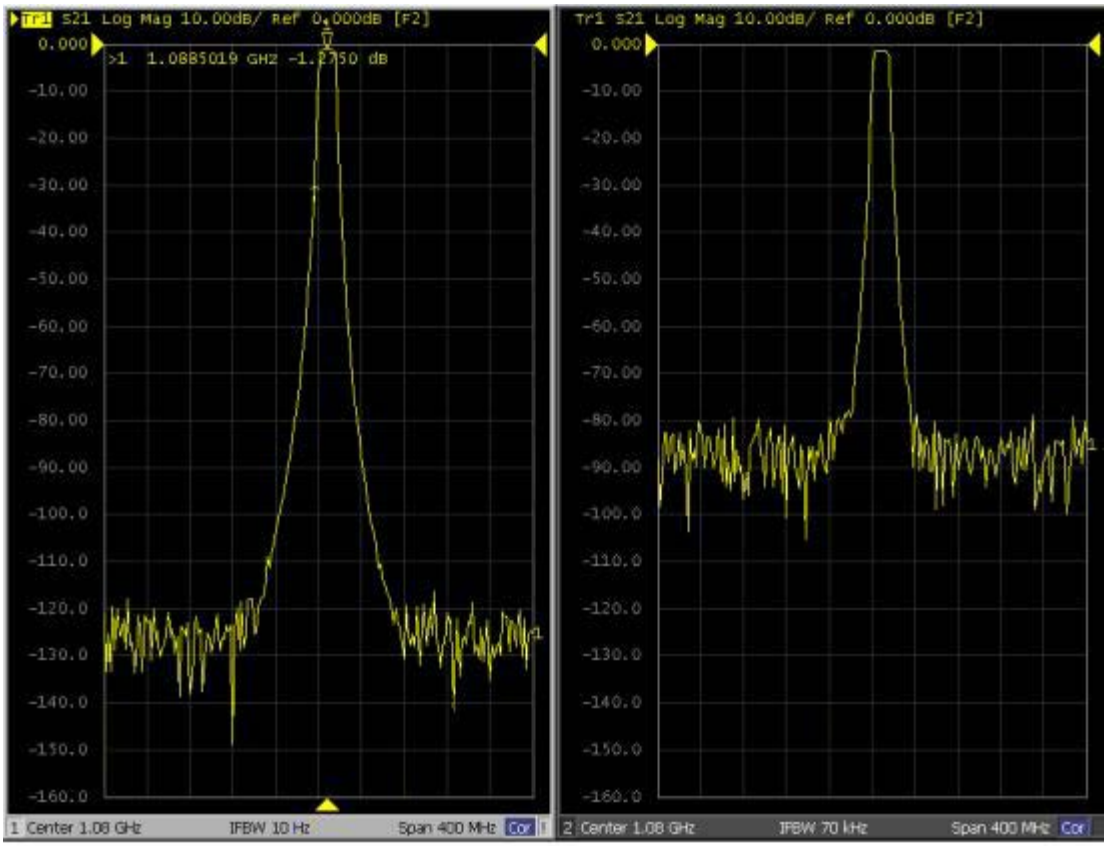
Lowering Receiver Noise Floor

Lowering the noise floor of the receiver enables you to expand the dynamic range. The following methods can be used to lower the receiver noise floor.

- Narrowing the IF bandwidth
- Turning on Sweep Averaging

Narrowing the IF bandwidth

Narrowing the receiver IF bandwidth enables you to reduce the effect of random noise on measurements. Narrowing the IF bandwidth to 1/10 the original bandwidth causes the receiver noise floor to decrease by 10 dB.



e5071c080

To specify the IF bandwidth, follow the steps described below.

1. Press **Channel Next**/**Channel Prev** keys to select a channel on which to specify the IF bandwidth.
2. Press **Avg** key.
3. Click **IF Bandwidth**, then Change the IF bandwidth in the data entry area.

Turning on Sweep Averaging

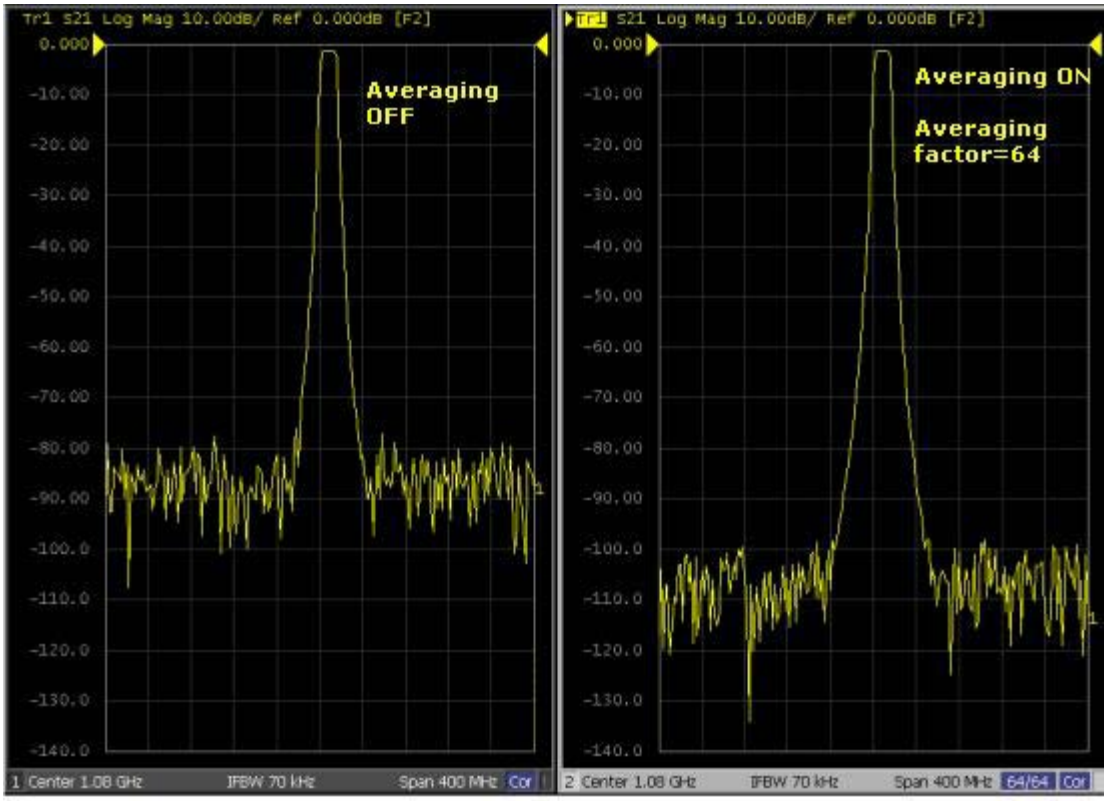
Using sweep averaging also enables you to reduce the effects of random noise on measurements.

Sweep averaging averages data from each point (vector quantity) based on the exponential average of a continuous sweep weighted by the averaging factor specified by the user. Sweep averaging is expressed in following equation.

$$A_n = \frac{S_n}{F} + \left(1 - \frac{1}{F}\right) \times A_{n-1}$$

where:

- $\mathbf{A_n}$ = Result of the calculation of sweep averaging for the nth sweep operation at the point in question (a vector quantity)
- $\mathbf{S_n}$ = Measurement value obtained at the nth sweep operation at the point in question (a vector quantity)
- **F** = Sweep averaging factor (an integer between 1 and 999)



e5071c081

Define the sweep averaging by following the steps below.

1. Press **Channel Next**/**Channel Prev** keys to select the channel on which you want to define the sweep averaging.
2. Press **Avg** key.
3. Click **Avg Factor**, then change the averaging factor in the data entry area.
4. Click **Averaging** to turn ON the averaging.
    5. Clicking **Averaging Restart** resets n to 1 in Sweep Averaging equation in Turning on Sweep Averaging

Reducing Trace Noise

Any of the following methods can be used to lower the trace noise. This section provides the description of Turing on Smoothing.

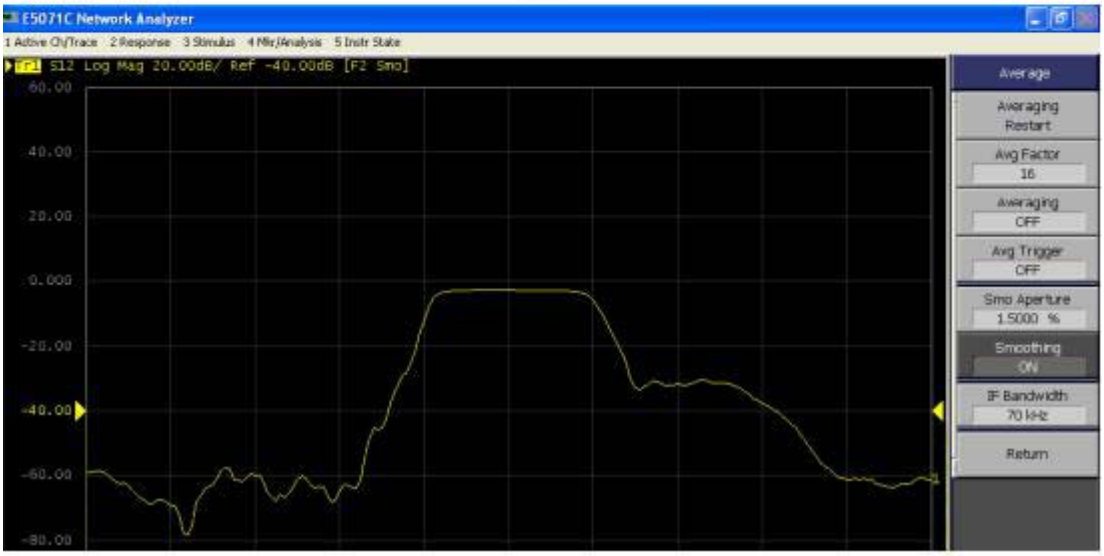
- Turning on Smoothing
- Turning on Sweep Averaging
- Narrowing IF Bandwidth

Other topics about Optimizing Measurements

Turning on Smoothing

Smoothing can be used to reduce noise that has relatively small peaks. By turning on smoothing, the value of each point on a trace is represented by the moving average over the values of several nearby points. The smoothing aperture (percentage of sweep span) defines the range of points to be included in the calculation of the moving average.

- You can define the smoothing trace by trace.



e5071c075

Setting up smoothing

Set up the smoothing operation by following the steps below:

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to activate the trace on which smoothing will be defined.
2. Press **Avg** key.
3. Click **Smo Aperture**, then change the smoothing aperture (%) in the data entry area.
4. Click **Smoothing** to turn ON smoothing.

Improving Phase Measurement Accuracy

This section describes the following functions that can be used to improve phase measurement accuracy.

- Electrical Delay
- Velocity Factor
- Phase Offset
- Port Extensions and Loss Values

Other topics about Optimizing Measurements

Electrical Delay

Electrical Delay is a function that adds or removes a pseudo-lossless transmission line with a variable length corresponding to the receiver input. Using this function enables you to improve the resolution in phase measurement and thereby measure deviation from the linear phase. You can specify the electrical delay trace by trace. Depending on the media type, the calculation method of the electrical delay, which is required to correct the phase delay, differs.

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to activate the phase trace for which you want to specify the electrical delay.
2. Press **Scale** key.
3. Click **Electrical Delay**.
4. Change the electrical delay (in seconds) in the data entry area.
5. Click **Media**, and select a media type for calculating the electrical delay. If the electrical delay is 0 second, the calculation result is always the same regardless of media type.

Softkey	Function
Coaxial	Selects Coaxial as the media type.
Waveguide	Selects Waveguide as the media type.

6. If you have selected **Waveguide** as the media type, click **Cutoff Frequecy**, and specify a cutoff frequency.
7. The cutoff frequency is available only when the media type is **Waveguide**.

For how to determine the deviation from a linear phase, see Measuring the Deviation from a Linear Phase.

Procedure using marker

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to activate the trace for which you want to set the electrical delay.
2. Place the *active marker* in an appropriate position.
3. Press **Maker Fctn**.
4. Click **Marker -> Delay** to set the electrical delay to the group delay value at the position of the active marker (a value smoothed with the aperture of 20% regardless of the smoothing setting).

Phase offset

Phase offset is a function used to add or subtract a predetermined value relative to the frequency to and from the trace. Using this function enables you to simulate the phase offset occurring as a result of, say, adding a cable.

The phase offset can be specified from - 360 ° to +360 ° .

Using the Phase Offset Function

1. Press **Channel Next**/**Channel Prev** keys and **Trace Next**/**Trace prev** keys to activate the trace for which you want to specify the phase offset.
2. Press **Scale** key.
3. Click **Phase Offset**, then enter the phase offset (°) in the data entry area.

Velocity factor

The velocity factor is the ratio of the propagation velocity of a signal in a coaxial cable to the propagation velocity of that signal in free space. The velocity factor for a common cable is about 0.66. The propagation velocity depends on the dielectric constant (ε_r) of the dielectric substance the cable.

$$\text{Velocity factor} = \frac{1}{\sqrt{\varepsilon_r}}$$

By specifying the velocity factor, you can match the equivalent length (in meters) appearing in the data entry area to the actual physical length when using the Electrical Delay or Setting port extensions to specify the electrical delay (in seconds).

You can define the velocity factor channel by channel.

Using the velocity factor

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to specify the velocity factor.

2. Press **Cal** key.
3. Click **Velocity Factor**, then the velocity factor in the data entry area.

Port Extensions and Loss Values

Setting port extensions

Port Extension is a function for moving the calibration reference plane by specifying the electrical delay. This function is useful, for example, when you cannot directly perform calibration at the DUT terminal because the DUT is inside the test fixture. In such a case, this function enables you to first perform calibration at the test fixture terminal and then move the calibration plane to the DUT terminal by extending the port.

Port extension corrects the electrical delay of each test port (phase shift) only. It cannot remove errors caused by the loss in and incorrect matching of cables, adapters, or test fixtures.

- You can define port extension channel by channel. Setting port extension for one particular channel does not affect other channels.
- Auto Port Extension does not supports waveguide port extension.

Operational procedure

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set port extension.
2. Press **Cal** key.
3. Click **Port Extensions**, and then Select **Extension Port 1/2/3/4**.
4. Specify Loss values by clicking **Loss**.

Setting coaxial delay, waveguide delay and cut off frequency

Follow the steps below to set coaxial or waveguide delay:

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set port extension.
2. Press **Cal** key.
3. Click **Port Extensions.**

Softkey	Function
Auto Port Extension	Sets the Auto Port Extension
Extension Port 1	Sets port extension (in seconds) for test port 1.

Extension Port 2	Sets port extension (in seconds) for test port 2.
Extension Port 3	Sets port extension (in seconds) for test port 3.
Extension Port 4	Sets port extension (in seconds) for test port 4.

4. Click **Extension Port n** where n can be from 1 to 4.
5. Click **Waveguide** to set delay in sec.
6. Click **Cutoff Frequency** to set cut off frequency of selected port.
7. Click **Coax. Extension** to set the coaxial extension in sec.
8. If you want to use waveguide port extension, then setting of coaxial delay is not necessary. If only waveguide measurement is needed, coaxial delay should be set to zero. Coaxial extension value should be set while measuring adapters between coaxial and waveguide.

Waveguide delay and Cut off frequency can be set using the following commands:

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

- System Z0 should be changed to 1 ohm before calibration when using waveguide calibration kit and measuring waveguide devices.
- Some calibration kits such as the waveguide calibration kit have operational frequency range defined by Minimum and Maximum frequency. When the E5071C stimulus setting is out of the operational frequency range, you cannot click Done key or finish the calibration by remote control. In this case, use a calibration kit that has proper frequency range, or change the E5071C stimulus setting to proper range that the calibration kit can cover. Refer to the calibration kit manual or definition in the E5071C with **Cal** > **Modify Cal Kit** > **Define STDs** for the Maximum & Minimum frequency of the calibration kit.

Setting loss values

In addition to port extension, you can set loss values for each port. By correcting loss due to port extension, more accurate measurement results are obtained.

There are two types of loss value settings: loss values at two frequency points for a specified port, and a DC loss value. You can make these settings at the same time for each port.

- You can set loss values channel by channel. Setting loss values for one particular channel does not affect other channels.

1. Setting loss values

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set loss values.
2. Press **Cal** key.
3. Click **Port Extensions.**
4. Click **Extension Port n** where n can be from 1 to 4.
5. Click **Loss** to set a loss value.
6. Click **Loss1 [OFF]** to toggle to **Loss1 [ON]** (enabled), and enter a loss value (**Loss1**) and a frequency (**Freq1**).
7. If you want to set loss at two frequency points, press **Loss2 [OFF]** to toggle to **Loss [ON]** (enabled), and enter a loss value (**Loss2**) and a frequency (**Freq2**).
8. If you want to set loss values for other ports, repeat steps 4 to 6.

- Expression to calculate loss using Loss 1:

$$\mathrm{Loss(f)} = \mathrm{Loss1} \times \sqrt{\left(\frac{\mathrm{f}}{\mathrm{Freq1}}\right)}$$

- Expression to calculate loss using Loss 1 and Loss 2:

$$\mathrm{Loss(f)} = \mathrm{Loss1} \times \left(\frac{\mathrm{f}}{\mathrm{Freq1}}\right)^{\mathrm{n}}$$

$$\mathrm{n} = \frac{\log_{10}\left|\frac{\mathrm{Loss1}}{\mathrm{Loss2}}\right|}{\log_{10}\frac{\mathrm{Freq1}}{\mathrm{Freq2}}}$$

- When you specify two frequency points, set the lower frequency to Loss1, and the higher one to Loss2.

2. Setting a DC loss value

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set DC loss values.
2. Press **Cal** key.
3. Click **Port Extensions** > **Loss**.
4. Click **Select Port** to select the port for which you want to set a DC loss value.
5. Click **Loss at DC**, then enter a DC loss value.
6. If you want to set a DC loss value for other ports, repeat steps 4 to 5.

Enabling port extensions and loss values

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to enable port extension.
2. Press **Cal** key.
3. Click **Port Extensions**.
4. Turn on **Extensions**.

Using the auto port extension function

The auto port extension function measures port extension and loss values for each port using the OPEN/SHORT standard connected to the port, automatically calculates them, and set them.

Softkey		Function
Auto Port Extension	**Select Ports**	Sets the Ports (1 to 4)
	Measure OPEN	Performs a OPEN measurement
	Measure SHORT	Performs a Short measurement
	Method	Sets the Span
	Include Loss	Sets Loss ON/OFF
	Adjust Mismatch	Sets Adjust Mismatch ON/OFF

When the auto port extension function is completed, the port extensions and loss values are updated to the calculated values.

- You can use both open and short measurement values in the auto port extension function. Note that in this case, the average value of the calculation results is used for updating.

- You can set the auto port extension function channel by channel. Setting the auto port extension function for one particular channel does not affect other channels.
- When the sweep type is power sweep or the frequency offset function is ON, the auto port extension is not available.
- Auto Port Extension does not supports waveguide port extension.

1. Selecting a port(s)

Select the port(s) for which you want to use the auto port extension function.

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set auto port extension.
2. Press **Cal** key.
3. Click **Port Extensions** > **Auto Port Extension** > **Select Ports** to select the port(s) for which you want to use the auto port extension function.

2. Setting frequencies used for calculation

Set the frequency points with which you want to calculate a loss value.

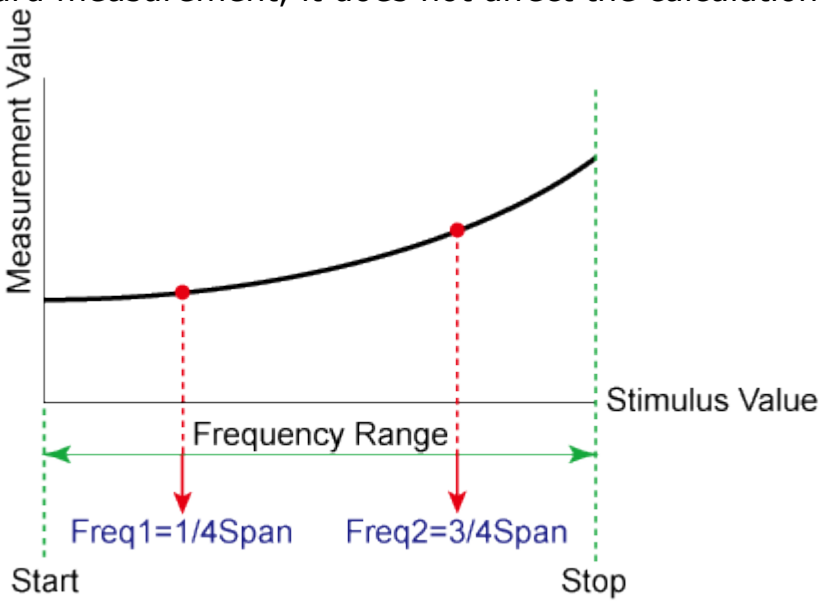
1. Press **Channel Next**/**Channel Prev** keys to activate the channel to set auto port extension.
2. Press **Cal** key.
3. Click **Port Extensions** > **Auto Port Extension** > **Select Ports** > **Method** to set the frequencies used for calculation.

Softkey	Function
Current Span	Executed using the frequency range set currently.
Active Marker	Executed using the frequency at the active marker. In this case, the result is applied to Loss1. Loss2 is ignored.
User Span	Executed using a start value and a stop value you set.

4. If you have selected **User Span**, use **User Span Start** and **User Span Stop** to set a start value and a stop value.

5. For **Current Span** and **User Span**, a frequency point at 1/4 of the frequency range is set to **Freq1**; a frequency point at 3/4 of the frequency range is set to **Freq2**.

6. If the setting is not made before starting OPEN/SHORT standard measurement, it does not affect the calculation result.



e5071c466

3. Specifying a loss value as a calculation target

Specify whether you want to include a loss value in the calculation result.

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set auto port extension.
2. Press **Cal** key.
3. Click **Port Extensions** > **Auto Port Extension** > **Select Ports** > **Include Loss** to turn it on.
   - If the setting is not made before starting the measurement of the OPEN/SHORT standard, it does not affect the calculation result.

4. Specifying a DC loss value as a calculation target

Specify whether you want to include a DC loss value in the calculation result.

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to set auto port extension.
2. Press **Cal** key.
3. Click **Port Extensions** > **Auto Port Extension** > **Adjust Mismatch** to turn it on.
    - If the setting is not made before starting the measurement of the OPEN/SHORT standard, it does not affect the calculation result.

5. Measuring the OPEN/SHORT standard and executing calculation

Calculate port extensions and loss values based on the calculation results using the OPEN/SHORT standard.

1. Press **Channel Next**/**Channel Prev** keys to activate the channel for which you want to want to set auto port extension.
2. Press **Cal** key.
3. Click **Port Extensions** > **Auto Port Extension**.
4. If you use the OPEN standard, click **Measure OPEN**, and select the port(s) for which you want to execute measurement. Execution is restricted to ports selected in Selecting a port(s)
5. If you use the SHORT standard, click **Measure SHORT**, and select the port(s) for which you want to execute measurement. Execution is restricted to ports selected in Selecting a port(s).
    - If a port extension value or loss value has been set, the value is updated to the calculated result.
    - If you execute both open measurement and short measurement, the average of the calculation results is reflected to the port extension and loss value.

6. Deleting the result of open/short measurement

When you exit from the softkey menu in the same level after open/short measurement, the measurement results are deleted. Note that you can use a GPIB command.

- Port extension and loss values that have been calculated are not cleared.

Improving Measurement Throughput

This section explains the methods to improve measurement throughput.

- Change Sweep Mode
- Turning off the updating of information displayed on the LCD screen
- Turning off system error correction

Other topics about Optimizing Measurements

Change Sweep Mode

The E5071C provides the two sweep modes: Stepped Mode and Swept Mode.

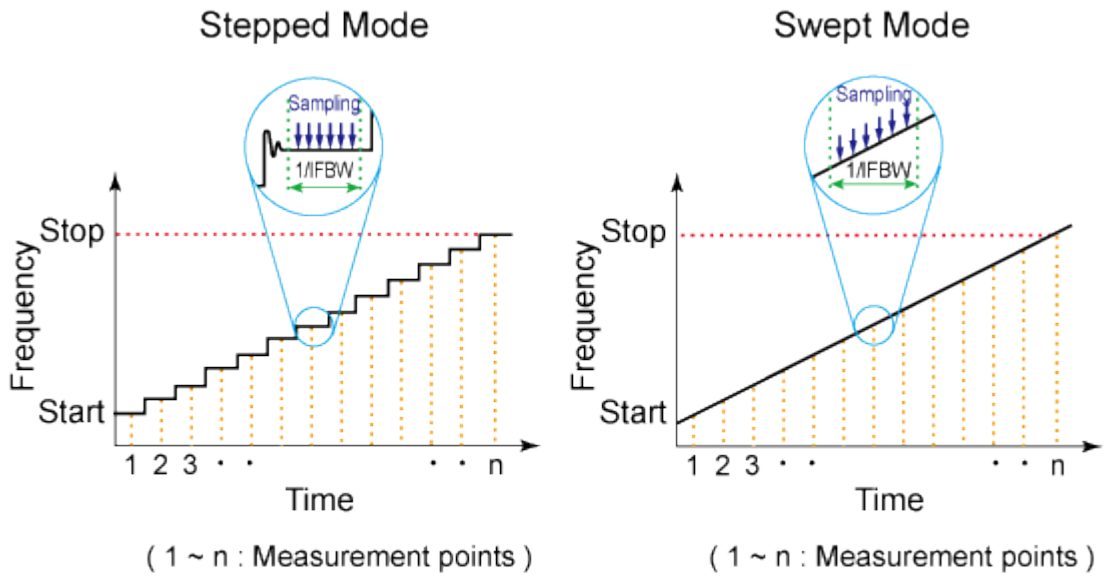
Mode	Stepped	Swept
Measurement Reliability	High	Low
Sweep Time	Long	Short
Restrictions on measurement	No Particular restriction	DUT with long electrical delay cannot be measured correctly.

Procedure to select the sweep mode

1. Press **Sweep Setup** key.
2. Click **Sweep Mode**.
3. Select **Stepped** or **Swept** for your appropriate sweep mode.

Overview

In the stepped mode, the frequency is changed stepwise and sampling is performed at a fixed frequency for each measurement point. On the other hand, in the swept mode, sampling is performed with the frequency always swept for each measurement point.



e5071c397

In the stepped mode, a certain time should be allowed until the frequency becomes stable at each measurement point, since the frequency is changed stepwise. Therefore, the sweep time in the swept mode is generally shorter than that in the stepped mode. However, if the measurement point interval is extremely large, the sweep to the next measurement point frequency cannot be completed within the sampling time due to the limitations of the instrument's frequency sweep speed. This causes a certain waiting time until the start of measurement of the next measurement point, and thus in this case the swept mode actually has a longer sweep time.

Although there is some difference depending on the IF bandwidth setting, when the measurement point interval is approximately 30 MHz or less, the sweep time in the swept mode is shorter than that in the stepped mode.

When you use the swept mode, you should confirm that there is no measurement-related problem before performing the actual measurement.

- The swept mode has the following disadvantages.
- DUTs with long electrical delay time cannot be measured correctly. For more information, refer to Notes for measuring DUTs with long electrical delay time
- Trace noise may increase compared to the stepped mode because sampling is performed while sweeping the frequency.
- The specifications are not guaranteed

Notes for measuring DUTs with long electrical delay time

When sweeping the frequency of a signal applied to the DUT (F), there is a frequency difference between the input side and output side of the DUT due to the delay time that occurs in the DUT (Delta T). This frequency difference (Delta F) becomes larger as the electrical delay time of the DUT becomes longer and the frequency sweep speed becomes faster, as shown in the following equation.

$$\Delta F = \frac{dF}{dt} \times \Delta T$$

When measuring a DUT with long electrical delay time, if you perform measurement (sampling) while sweeping the measurement signal as in the swept mode, a measurement error occurs due to the difference between the frequency outputted from the source port and the frequency actually measured at the receiver port.

Therefore, when measuring a DUT with a long electrical delay time, generally use the stepped mode to prevent the measurement error described above. However, if you need to shorten the sweep time, evaluate the measurement error as described below to determine the sweep mode that should be used.

Turning off the updating of information displayed on the LCD screen

Turning off the updating of information displayed on the LCD screen eliminates the processing time required to update displays within the analyzer, improving measurement throughput. If it is not necessary to check displayed information during measurements, turning off real-time updating is an effective means of improving throughput.

The updating of information displayed on the LCD screen can be switched using the following procedure:

Turning off the updating of information

1. Press **Display** key.
2. Click **Update** to switch the updating of displayed information on the LCD screen on/off.

When the LCD screen update is turned off, **Update Off** appears on Instrument Status Bar.

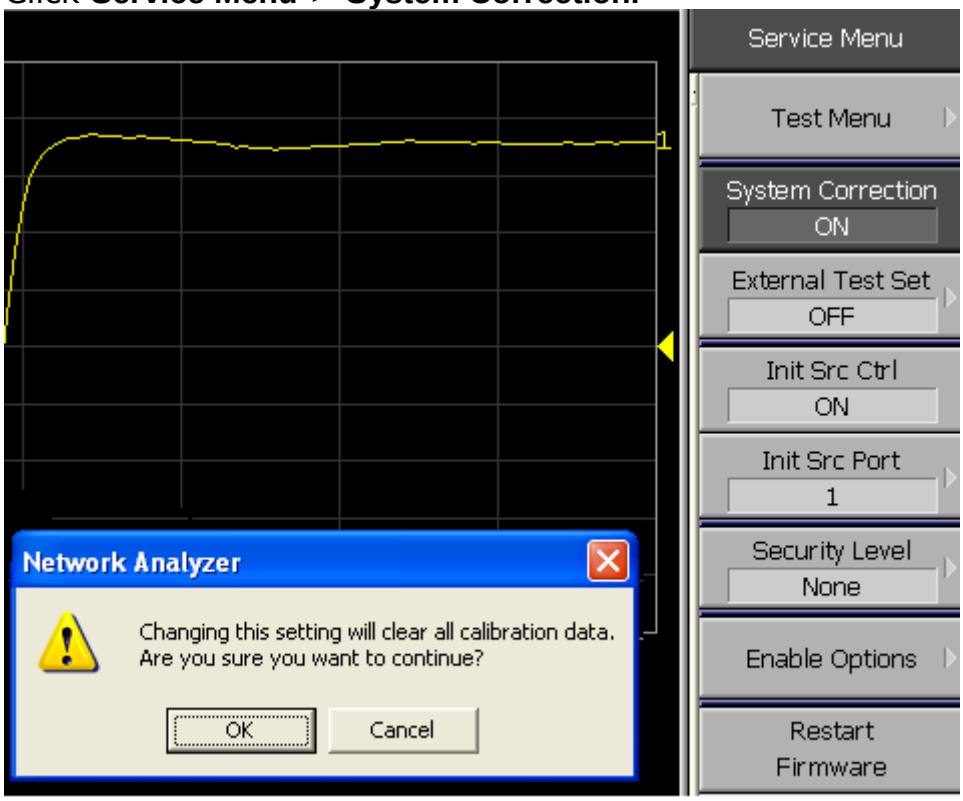
Turning off system error correction

The E5071C executes Port Characteristics Correction in the data processing flow shown in Data Processing Flowchart, by using the system calibration data set at the factory. This system error correction process is not required if the user performs proper calibration by using the **Cal** key and the softkeys that subsequently appear, which automatically turns on error correction.

By turning off system error correction, you can reduce the data processing time needed during measurement and thus improve measurement throughput.

- When you turn ON/OFF system error correction, all calibration data set by user calibration is deleted.

1. Press **System** key.
2. Click **Service Menu** > **System Correction.**



e5071c074

3. Click **OK** to turn ON/OFF system error correction.

Performing a Segment-by-Segment Sweep

This section describes the concept of the segment sweep and how to perform it.

- Concept of Segment Sweep
- Conditions for Setting Segment Sweep
- Items that can be set for Each Segment
- Sweep Delay/Sweep Time in Segment Sweep
- Turning ON/OFF Segments Individually
- Frequency/Order Base Display

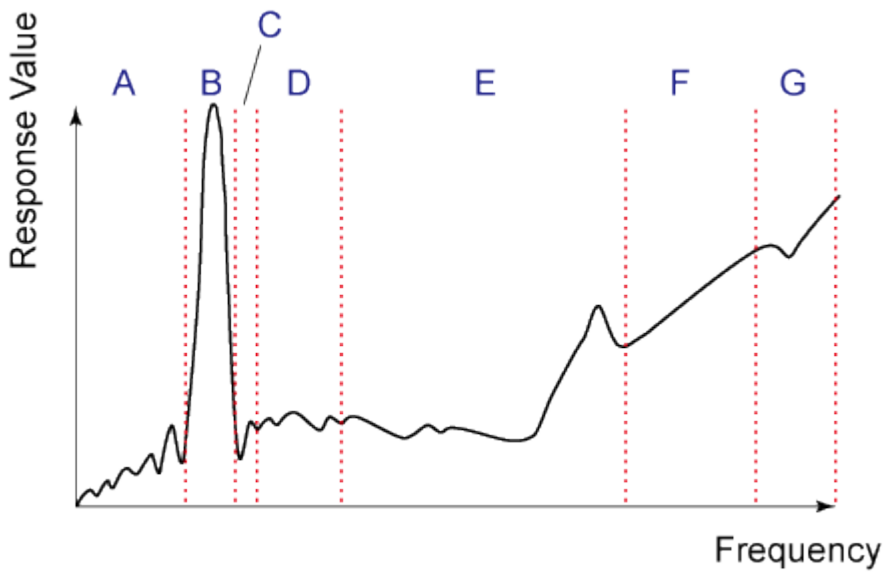
Other topics about Optimizing Measurements

Concept of Segment Sweep

To perform a segment sweep, you must define two or more frequency ranges, called segments, and then specify the number of points, IF bandwidth, power level, sweep mode, sweep delay time, and sweep time for each segment. All segments are swept sequentially as if swept in one sweep operation.

- By skipping the frequency range, which does not need to be measured, you can sweep and measure only the portions you need.
- You can define the optimum measurement conditions for each of the segments you designate. For example, you can specify as many points as possible in a segment requiring high trace resolution and as few points as possible in a segment not requiring high resolution. This shortens the measurement time, enabling you to optimize the overall measurement throughput by not having to perform the entire operation under the same measurement conditions of a particular frequency range.

To evaluate a band pass filter having the transmission characteristics shown in the following figure. for example, you can select the frequency ranges you need from A through G and determine the measurement conditions shown in the table below This enables you to measure them simultaneously in one sweep operation.



e5071c393

Frequency ranges (segments) from the figure above and their measurement conditions						
	Start frequency	**Stop frequency**	**Number of points**	**IF Bandwidth**	**Sweep mode**	**State**
A	440 MHz	915 MHz	50	50 kHz	Stepped	ON
B	915 MHz	980 MHz	130	70 kHz	Stepped	ON
C	980 MHz	1.035 GHz	60	50 kHz	Stepped	ON
E	1.07 GHz	2 GHz	100	70 kHz	Swept	ON
G	2.6 GHz	3 GHz	40	70 kHz	Swept	ON

Conditions for Setting Segment Sweep

The following conditions apply when setting up a segment sweep.

- The frequency range of a segment must not overlap with that of another segment. (The start frequency of a segment must be higher than the stop frequency of the immediately preceding segment).

- The start frequency of segment 1 must be greater than lowest frequency and the stop frequency of the last segment less than highest frequency, as per the frequency range depending on the option.
- When the start frequency and stop frequency of a segment are not the same, you can define from 2 to 1601 points in a segment.
- When the start frequency and stop frequency of a segment are the same, you can define from 1 to 1601 points in a segment.
- You can set the total number of points in the segment table from 2 to 1601.
- You can set the number of segments in the segment table to between 1 and 201.
- You can turn ON or OFF segments.

Items that can be set for Each Segment

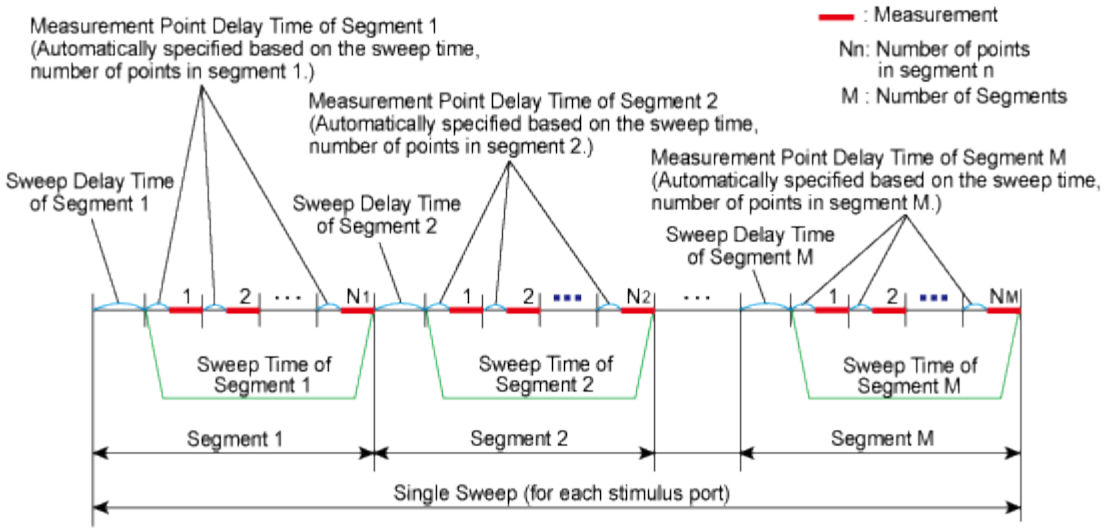
For the segment sweep, you can set the sweep range, the number of points, IF bandwidth, power level, sweep delay time, sweep mode, sweep time and segment ON/OFF for each segment.

You can set the items in the following table to ON/OFF for each segment. If you enable the segment-by-segment setting, you can make the setting for each segment in the segment table; if you disable it, the setting in the following table is used.

Item	When segment-by-segment setting is disabled
IF bandwidth	For all segments, the IF bandwidth for the linear/log sweep (set with **Avg** key > **IF Bandwidth**) is set.
Power level	For all segments, the power level for the linear/log sweep (set with **Sweep Setup** key > **Power**) is set.
Sweep delay time	For all segments, 0 is set.
Sweep mode	For all segments, the sweep mode for the linear/log sweep (set with **Sweep Setup** key > **Sweep Mode**) is set.
Sweep time	For all segments, the auto sweep time mode is set.
Segment ON/OFF	For all segments, ON is set.

Sweep Delay/Sweep Time in Segment Sweep

The definitions for sweep delay time and sweep time, which you can specify in the segment sweep, are shown in figure below.



e5071c308

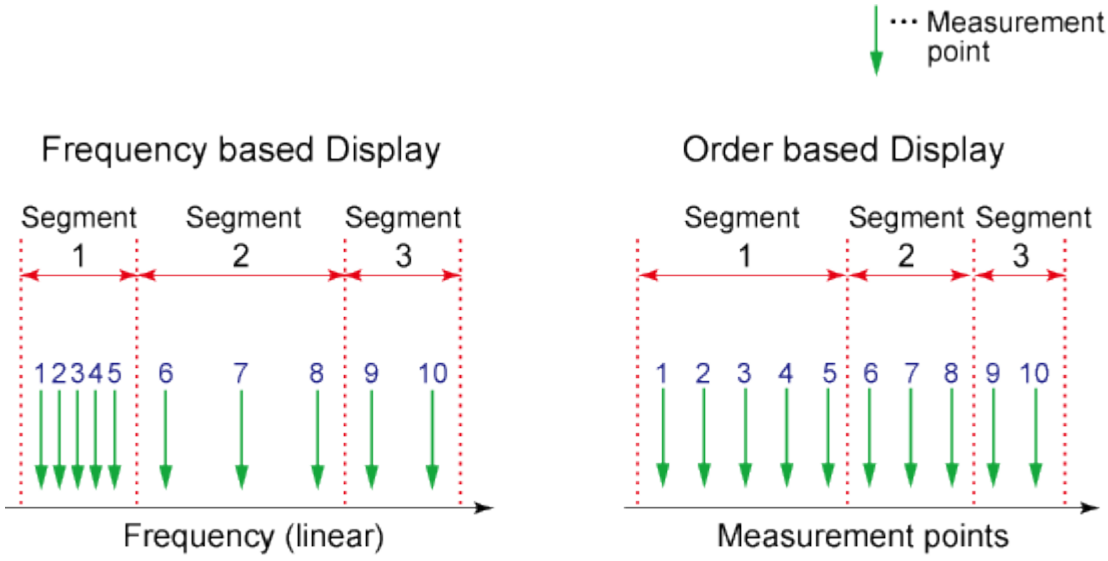
Turning ON/OFF Segments Individually

From the firmware revision A.09.60, each segment can be turned ON or OFF individually. The minimum number of points turned ON must be 2.

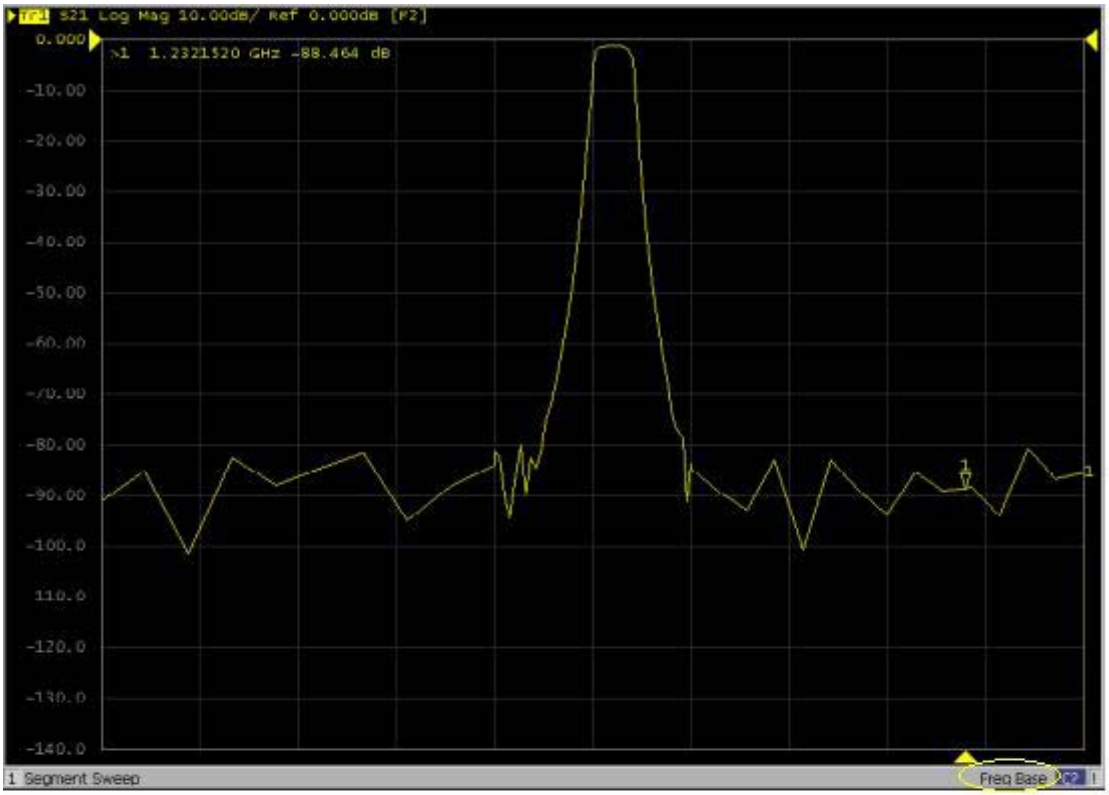
If the total number of points turned ON are less than 2, the segments with the highest frequency ranges is automatically turned ON to meet the minimum number of points.

Frequency/Order Base Display

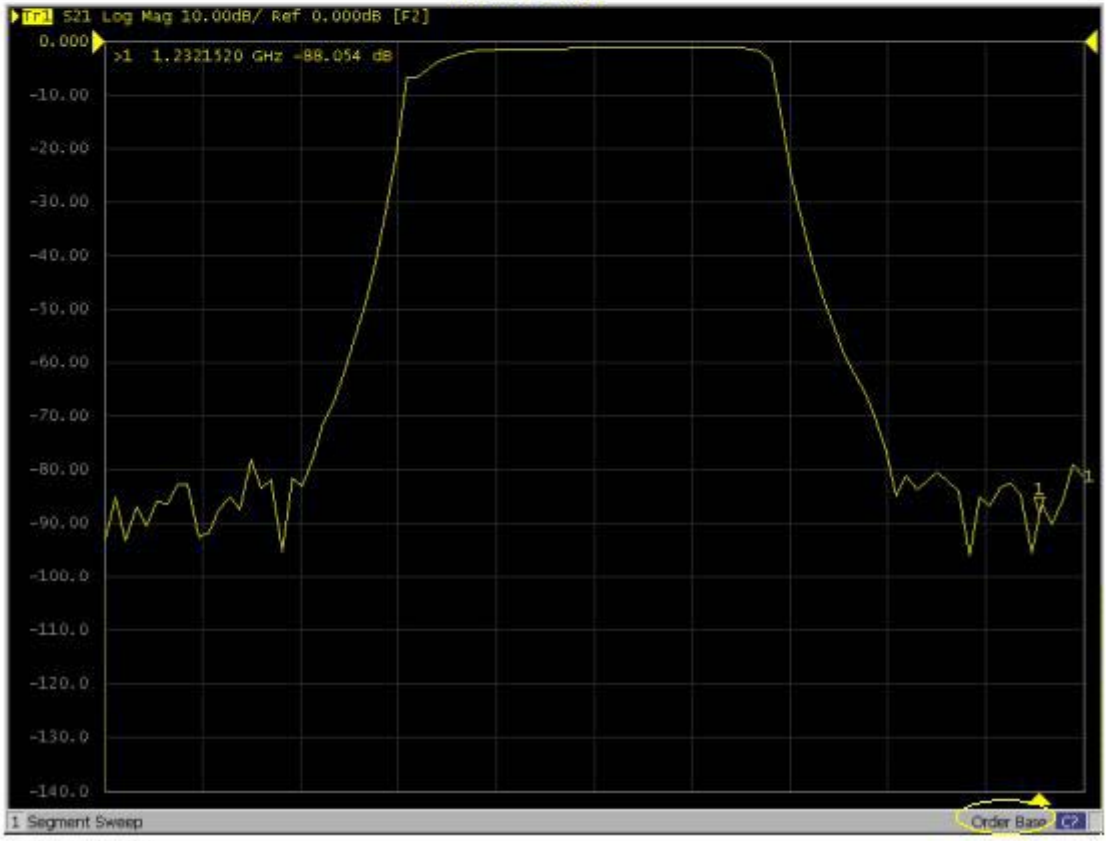
You can choose between frequency-based and order-based display as the method of displaying traces when executing a segment sweep.


Measurement point
Frequency based Display
Segment 1
Segment 2
Segment 3
1 2 3 4 5 6 7 8 9 10
Frequency (linear)
Order based Display
Segment 1
Segment 2
Segment 3
1 2 3 4 5 6 7 8 9 10
Measurement points

e5071c394



Display — Frequency Based / Order Based



e5071c082

Procedure

Creating a segment table

1. Press **Channel Next**/**Channel Prev** keys to select the channel for which you want to create the segment table.
2. Press **Sweep Setup** key.
3. Click **Edit Segment Table**. The segment table appears in the lower part of the screen.
4. To change the frequency range setting mode or to set the IF bandwidth, power level, sweep delay time, sweep mode, sweep time for each segment, and segment ON/OFF use the following softkeys.
    1. When setting the segment table using the front panel keys or keyboard, you need to place focus on (select) the operation target (segment table of softkey) first. You can change the focus by pressing **Focus** key in the **ENTRY** Block. When the focus is placed on the segment table, the window frame of the segment table is displayed as bright as the window frame of the active channel. When the focus is placed on the softkey menu, the softkey menu title area is displayed in blue.

Softkey	Function
Freq Mode	Switches the frequency range setting mode (start/stop or center/span)
List IFBW	Toggles ON/OFF the IF bandwidth setting for each segment; the row for setting (**IFBW**) only appears in the segment table when this is ON
List Power	Toggles ON/OFF the power level setting for each segment; the row for setting (**Power**) only appears in the segment table when this is ON
List Delay	Toggles ON/OFF the sweep delay time setting for each segment; the row for setting (**Delay**) only appears in the segment table when this is ON
List Sweep Mode	Toggles ON/OFF the sweep mode setting for each segment; the row for setting (**Sweep Mode**) only appears in the segment table when this is ON
List Time	Toggles ON/OFF the sweep time setting for each segment; the row for setting (**Time**) only appears in the segment table when this is ON

List State	Toggles ON/OFF the state of each segment; the row for setting (**State**) only appears in the segment table when this is ON

5. Enter each item in the following table for each added segment (line) to create the segment table. To create the segment table, use the hardkeys and softkeys.

Start	Sets the start value of the sweep range
Stop	Sets the stop value of the sweep range
Center	Sets the center value of the sweep range
Span	Sets the span value of the sweep range
Points	Sets the number of points
IFBW	Sets the IF bandwidth
Power	Sets the power level; the power range is common to the settings for the linear/log sweep (**Sweep Setup** key > **Power Ranges**)
Delay	Sets the sweep delay time
Sweep Mode	Sets the sweep mode; you need to select one of the following items: **STEPPED**: Stepped mode **SWEPT**: Swept mode
Time	Sets the sweep time; to specify the auto setting (**AUTO**), enter 0 as the sweep time
State	Sets the state of each segment

Useful functions when using a mouse

By right-clicking on the selected cell, you can use the following shortcut menu.

Shortcut	Function
Copy	Copies the value in the selected cell into the clipboard (internal temporary storage memory)

Paste	Pastes the value data in the clipboard to a newly selected cell
Insert	Adds a new line above the selected cell
Delete	Deletes the line containing the selected cell

In the character-by-character edit mode, you can also use the following shortcut menu.

Shortcut	Function
Undo	Undoes the change and restore the value before the change
Cut	Cuts the selected string and store it into the clipboard (temporary memory)
Copy	Copies the selected string into the clipboard
Paste	Pastes the string in the clipboard to a newly selected cell
Delete	Deletes the selected string
Select All	Selects the entire string in the cell

Executing segment sweep

To execute a segment sweep by using the segment table you have created, you must specify the sweep type for that sweep operation by following the steps below.

1. Press **Channel Next**/**Channel Prev** keys to select the channel on which you will execute the segment sweep operation.
2. Press **Sweep Setup** Key.
3. Press **Sweep Type** > **Segment**.

Setting up the segment display

Define the method of displaying traces when the segment sweep is executed by following the steps described below.

1. Press **Channel Next**/**Channel Prev** keys to select the channel on which you will define the segment display.
2. Press **Sweep Setup** Key.
3. Click **Segment Display**.
4. Select the segment display.

Softkey	Function

Freq Base	Displays the X-axis as the axis for linear frequencies (frequency-based display)
Order Base	Displays the X-axis as the axis for the points (order-based display)

Saving a newly created segment table in CSV format

As discussed in Creating a segment table, you can export the newly created segment table as a CSV (comma-separated value) formatted file (so it can be used easily in software that requires a different format).

1. Press **Sweep Setup** Key.
2. Click **Edit Segment Table** > **Export to CSV File** to open the Save As dialog box. Note that CSV Files (***.csv**) will already be selected as the file type when the dialog box first opens.
3. Type the file name in the File Name area and Click **Save** to save the segment table.

Calling a segment table saved in CSV Format

By importing a segment table file saved by E5071C, you can set up the segment table.

- It is possible to recall a file from a different channel where it was saved.

1. Press **Sweep Setup** Key.
2. Click **Edit Segment Table** > **Import from CSV File** to open the Open dialog box. Note that CSV Files (***.csv**) will already be selected as the file type when the dialog box first opens.
3. Select the CSV format file to be imported, and click **Open** to call up the segment table.
   - You cannot import a CSV-formatted file created/edited in spreadsheet software into the E5071C.
   - You cannot import CSV-formatted files created in firmware revision A.09.60 or later to firmware revisions before revision A.09.60.

Measurement Examples

Measurement Examples

- Measuring the SAW Bandpass Filter Using the Segment Sweep
- Evaluating a Duplexer
- Measuring the Deviation from a Linear Phase
- Measuring an Unbalanced and Balanced Bandpass Filter
- Measuring Parameters with Cable
- Evaluating Transmission Characteristics of a Front End Module
- Executing Power Calibration

Measuring SAW Bandpass Filter using Segment Sweep

- Overview
- Procedure

Other Measurement Examples

Overview

This section illustrates how to use the segment sweep function to evaluate a SAW bandpass filter with a center frequency of 947.5 MHz.

Procedure

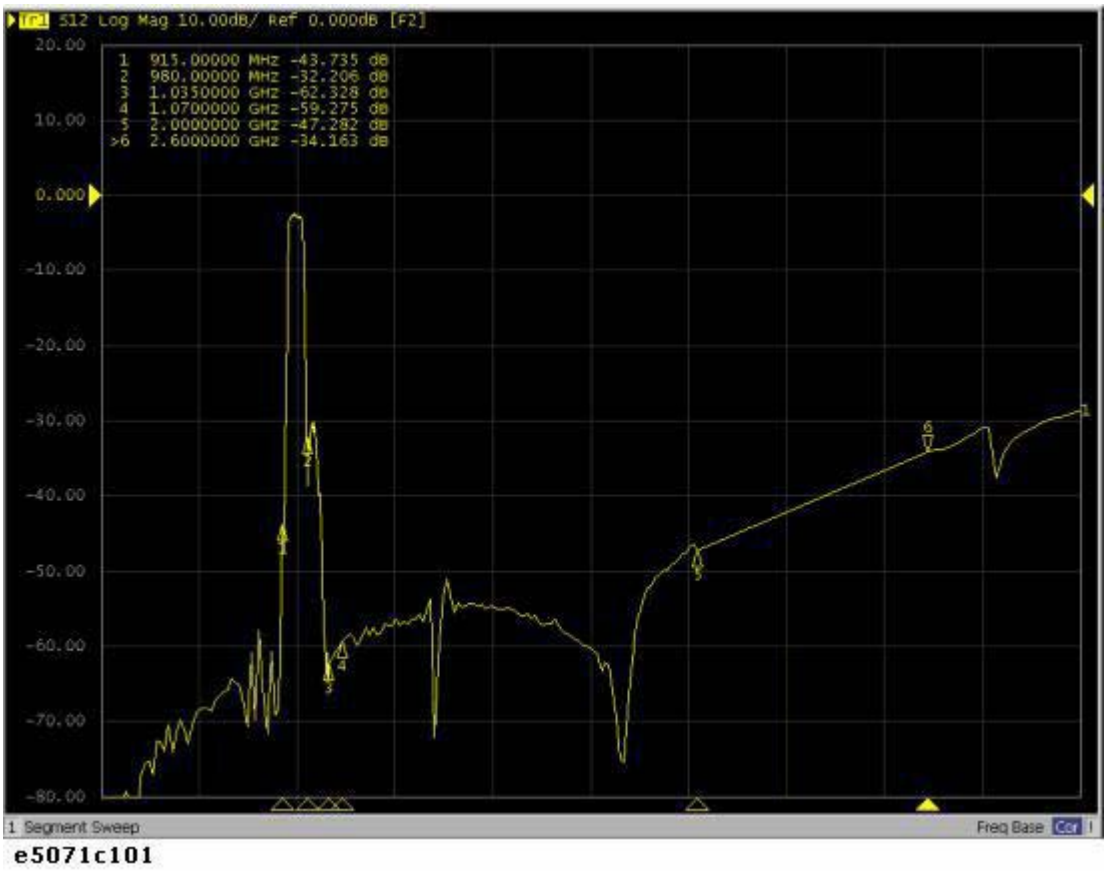
Here, the DUT is evaluated by following the steps.

Step	Description
1. Determine the Segment Sweep Conditions	The segment sweep conditions are determined according to the characteristics of the DUT.
2. Create a Segment Sweep Table	The segment sweep conditions are entered in the E5071C.
3. Select the Segment Sweep as the Sweep Type	The segment sweep is selected as the sweep type.
4. Execute the Calibration	A 2-port ECal is performed between the test ports connecting the DUT.
5. Connect the DUT	The DUT is connected.
6. Execute the Measurement	A trigger is applied to perform the measurement.
7. Define the Setup for Display	The choice is made between frequency base and order base as the method of displaying segments.

1. Determine the Segment Sweep Conditions

The following figure shows the results of evaluating the transmission characteristics of the SAW bandpass filter in the range of 440 MHz to 3 GHz by using the linear sweep.

Transmission characteristics of SAW bandpass filter (440 MHz to 3 GHz, linear sweep)



The measurement conditions are determined for each frequency range. Here, the segment sweep is performed following the sweep conditions shown in the following table.

Frequency Range		Measurement Conditions	
Start	**Stop**	**Number of Points**	**IF Bandwidth**
440 MHz	915 MHz (Marker 1)	47	70 kHz
915 MHz (Marker 1)	980 MHz (Marker 2)	130	100 kHz
980 MHz (Marker 2)	1.035 GHz (Marker 3)	55	70 kHz
1.07 GHz (Marker 4)	2 GHz (Marker 5)	93	70 kHz
2.6 GHz (Marker 6)	3 GHz	41	70 kHz

2. Create a Segment Sweep Table

Follow the steps below to make entries in the segment sweep table.

1. Display the segment table.

Setup Description	Key Operation
Presetting	**Preset** > **OK**
Displaying the segment table	**Sweep Setup** > **Edit Segment Table**

2. Display the IF bandwidth setting column in the segment table.

Setup Description	Key Operation
Moving the focus to the softkey menu	**Focus**
Display of IF bandwidth setting column: ON	**List IFBW** (turn it **ON**)

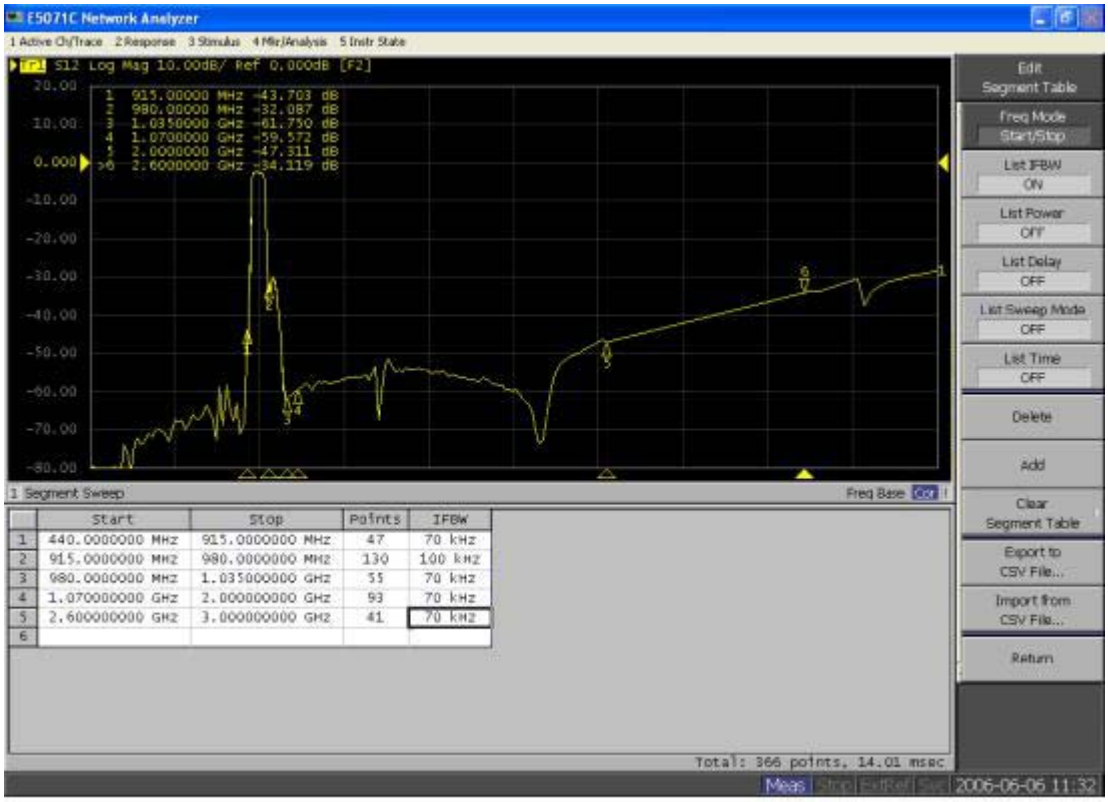
When setup items (power level, delay time, sweep mode, and sweep time in this case) are not displayed in the segment table, the setting for the channel in use applies to all segments.

3. Enter the setup data in the segment table.

Setup Description	Key Operation
Moving the focus to the segment table (select)	**Focus**
Segment 1	
Start frequency: 440 MHz	**4** > **4** > **0** > **M/μ**
Stop frequency: 915 MHz	**9** > **1** > **5** > **M/μ**
Number of points: 47	**4** > **7** > **x1**
IF bandwidth: 70 kHz	**7** > **0** > **k/m**
Segment 2	
Start frequency: 915 MHz	**9** > **1** > **5** > **M/μ**
Stop frequency: 980 MHz	**9** > **8** > **0** > **M/μ**

Number of points: 130	**1** > **3** > **0** > **x1**
IF bandwidth: 100 kHz	**1** > **0** > **0** > **k/m**
Segment 3	
Start frequency: 980 MHz	**9** > **8** > **0** > **M/µ**
Stop frequency: 1.035 GHz	**1** > **.** > **0** > **3** > **5** > **G/n**
Number of points: 55	**5** > **5** > **x1**
IF bandwidth: 70 kHz	**7** > **0** > **k/m**
Segment 4	
Start frequency: 1.07 GHz	**1** > **.** > **0** > **7** > **G/n**
Stop frequency: 2 GHz	**2** > **G/n**
Number of points: 93	**9** > **3** > **x1**
IF bandwidth: 70 kHz	**7** > **0** > **k/m**
Segment 5	
Start frequency: 2.6 GHz	**2** > **.** > **6** > **G/n**
Stop frequency: 3 GHz	**3** > **G/n**
Number of points: 41	**4** > **1** > **x1**
IF bandwidth: 70 kHz	**7** > **0** > **k/m**

Completed segment table



e5071c102

3. Select the Segment Sweep as the Sweep Type

The segment sweep is selected as the sweep type.

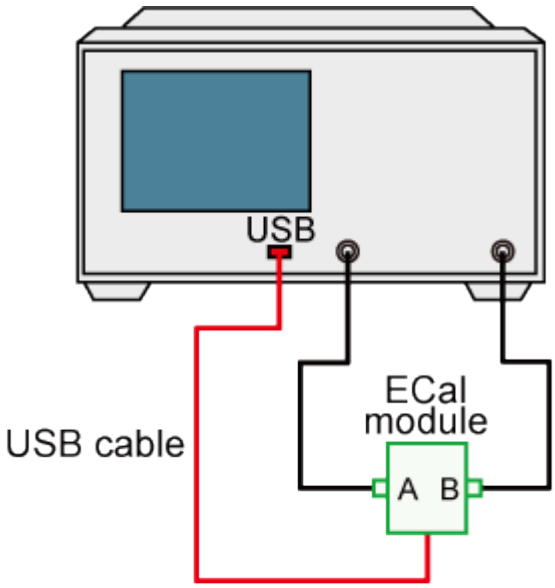
Setup Description	Key Operation
Sweep type: Segment sweep	**Sweep Setup** > **Sweep Type** > **Segment**

4. Execute the Calibration

In this step, a 2-port ECal is executed on the two ports to be used.

1. Connect the ECal module across test ports 1 and 2.

Connecting the ECal module



e5071c322

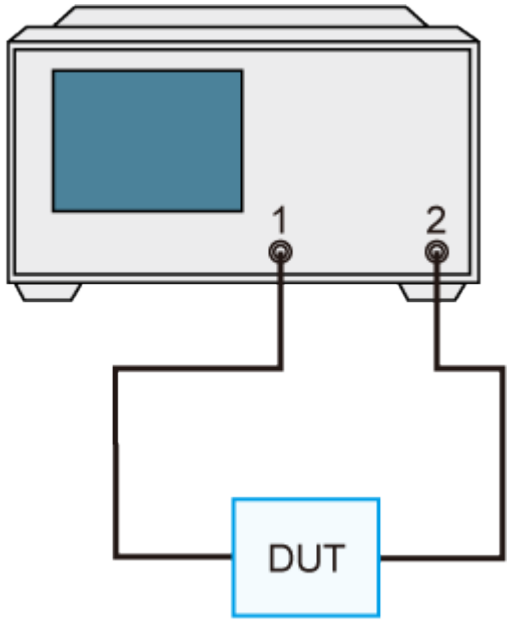
2. Execute the 2-port ECal.

Setup Description	Key Operation
Executing a 2-port ECal between test ports 1 and 2	**Cal** > **ECal** > **2 Port ECal** > **Port 1-2**

5. Connect the DUT

The DUT is connected across test ports 1 and 2.

Connecting the DUT



e5071c320

6. Execute the Measurement

A trigger is applied to perform the measurement.

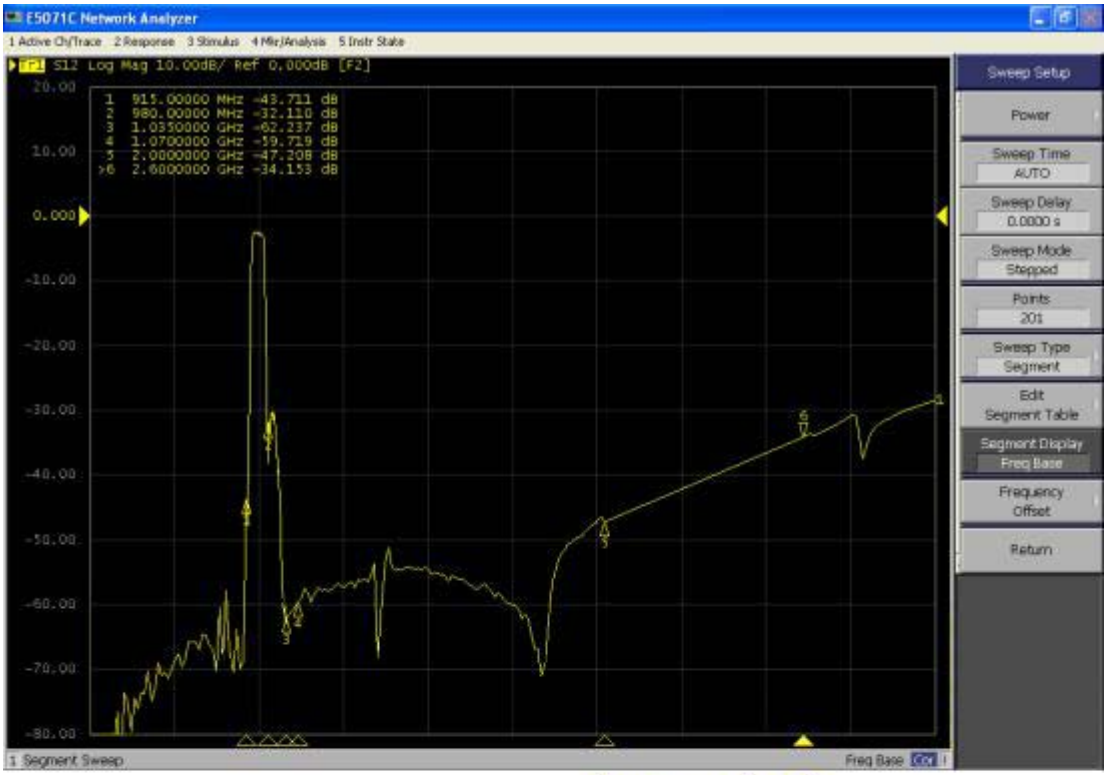
Setup Description	Key Operation
Trigger mode: Single	**Trigger** > **Single** (Or **Continuous**)

7. Define the Setup for Display

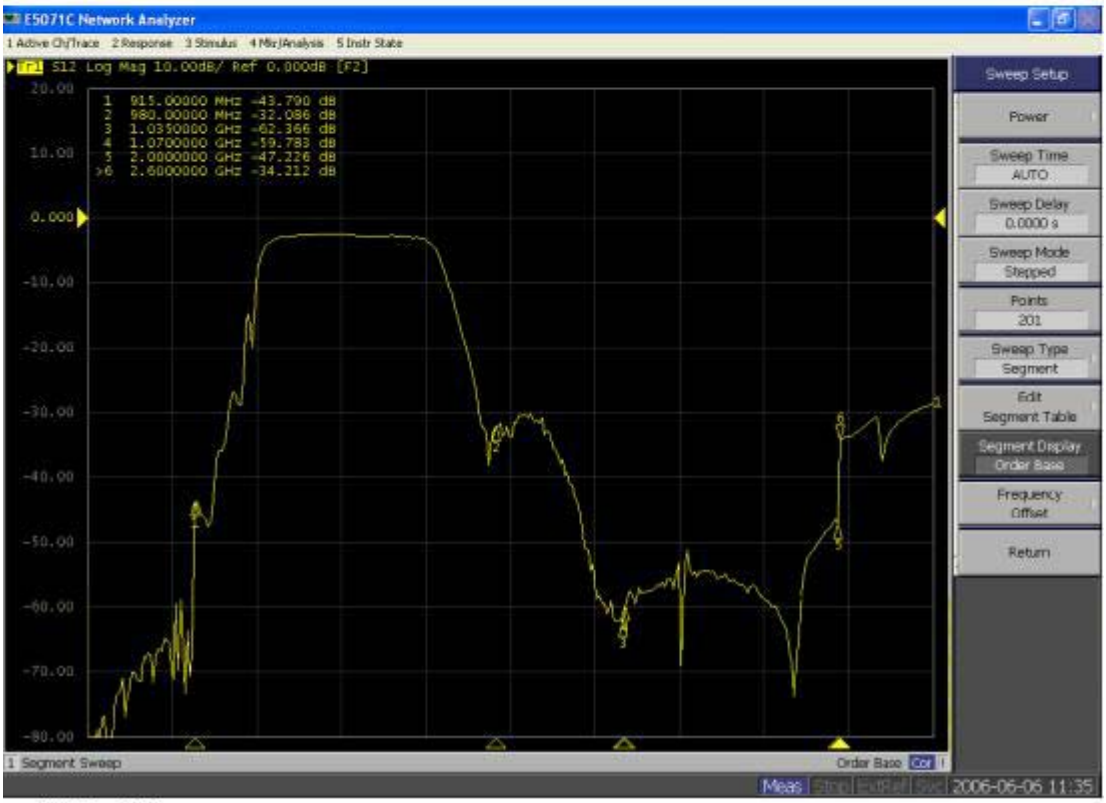
The choice is made between frequency base and order base as the segment display mode.

Setup Description	Key Operation
Segment display: Frequency base or order base	**Sweep Setup** > **Segment Display** > **Frequency Base** | **Order Base**

Segment display: Frequency/Order base



Segment Display — **Frequency Based** / **Order Based**



e5071c103

Evaluating a Duplexer

- Overview
- Procedure

Other Measurement Examples

Overview

This section illustrates how to evaluate a duplexer (Tx center frequency: 1.88 GHz, Rx center frequency: 1.96 GHz).

Procedure

Here, the DUT is evaluated by following the steps described in table below.

Steps	Description
1. Determine the Segment Sweep Conditions	Segment sweep conditions are determined according to the characteristics of the DUT.
2. Create a Segment Sweep Table	The segment sweep conditions are entered in the E5071C.
3. Select the Segment Sweep as the Sweep Type	The segment sweep is selected as the sweep type.
4. Execute the Calibration	A full 3-port calibration is executed by using the 2-port ECal module.
5. Connect the DUT	The DUT is connected.
6. Define the Setup for Display	The number of traces to be displayed, split display, and measurement parameters are specified.
7. Execute the Measurement	A trigger is applied to execute the measurement.
8. Define the Setup for the Segment Display and Scale	The setup for segment display and for the scale are defined.
9. Analyze the Parameters	The evaluation parameters for the duplexer are determined.
10. Define the Setup for a Limit Table	The setup for the limit table is defined.
11. Execute the Limit Test	The limit test is executed.

1. Determine the Segment Sweep Conditions

A segment sweep is performed by following the sweep conditions shown in table below.

Start	Stop	Number of Points
1.73 GHz	1.83 GHz	50
1.83 GHz	2.03 GHz	400
2.03 GHz	2.13 GHz	50
3.65 GHz	4.03 GHz	38
5.5 GHz	6.02 GHz	52

2. Create a Segment Sweep Table

Entries are made in the segment sweep table following the steps described below.

1. Display the segment table.

Setup Description	Key Operation
Presetting	**Preset** > **OK**
Displaying the segment table	**Sweep Setup** > **Edit Segment Table**

2. Enter the setup data in the segment sweep table.

In this step, the IF Bandwidth, power level, delay time, and sweep time are not entered segment by segment. By turning off the display of those parameters on the segment table, you can use, in each segment without making a change:

- the IF Bandwidth (preset value: 70 kHz) of the channel specified by using **Avg** > **IF Bandwidth**
- Power level (preset value: 0 dBm) of the channel specified by using **Sweep Setup** > **Power**
- Sweep delay time (preset value: 0 sec) of the channel specified by using **Sweep Setup** > **Sweep Delay**
- Sweep mode (preset value: Stepped mode) of the channel specified by using **Sweep Setup** > **Sweep Mode**
- Sweep time (preset value: Automatic) of the channel specified by using **Sweep Setup** > **Sweep Time**.

e5071c083

3. Select the Segment Sweep as the Sweep Type

The segment sweep is selected as the sweep type.

Setup Description	Key Operation
Sweep type: Segment sweep	**Focus** (Moves the focus to the softkey menu) > **Return** > **Sweep Type** > **Segment** (or **Sweep Setup** > **Sweep Type** > **Segment**)

4. Execute the Calibration

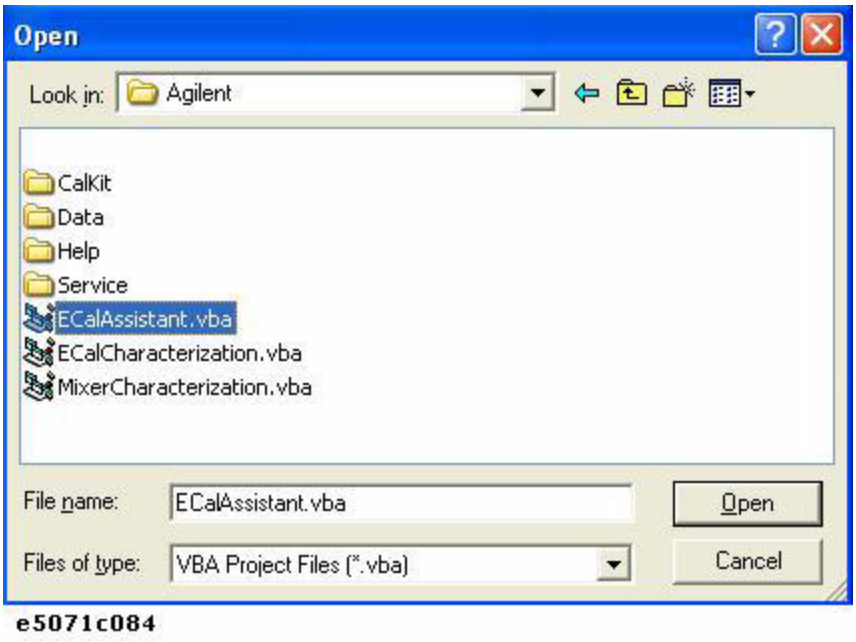
In this step, a 2-port ECal module and 3-/4-port module installed in the E5071C are used to execute calibration on the three ports used in the measurement.

1. Connect the USB port of the 2-port ECal module and the USB port of the E5071C with a USB cable. The connection may be made while the unit is powered.
2. Load and execute the 3-/4-port ECal programs.

Setup Description	Key Operation
Opening the VBA Project Open dialog box	**Macro Setup** > **Load Project**
Loading **ECalAssistant.VBA**	Select **D:\Agilent\ECalAssistant.VBA** and press the **Open** button.

Executing the program	**Macro Run**

3. The EcalAssistant (start) dialog box appears.



e5071c084

4. Click **Next** to display the EcalAssistant (port/channel selection) dialog box.

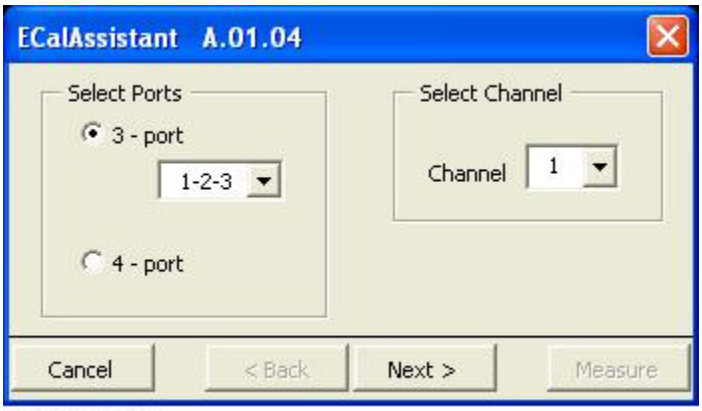
e5071c085

5. Following the instructions in the dialog box, select the type of ECal, test ports, and the channel.

Setup Description	Key Operation
Select Ports	

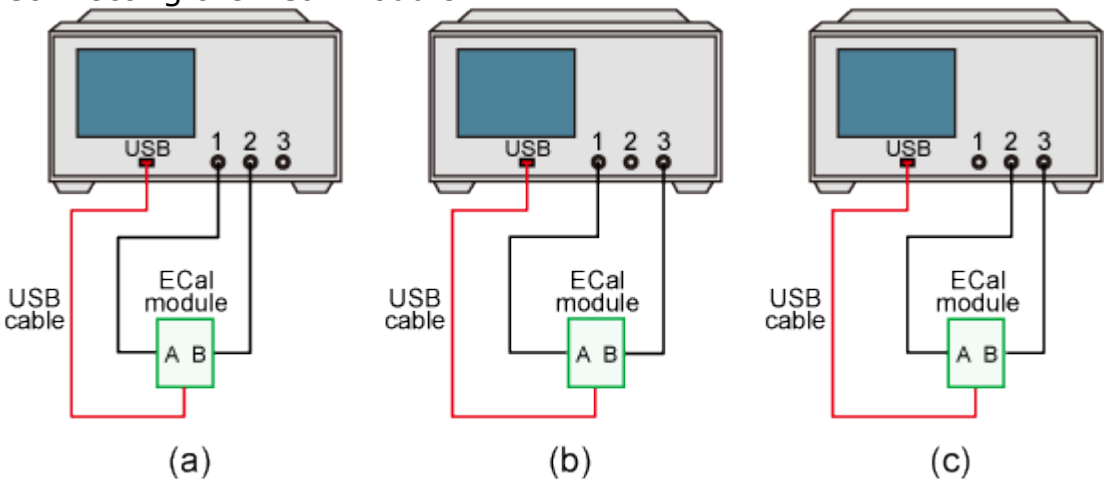
ECal type: Full 3-port calibration	**3 Port**
Test ports to be used for ECal: 1, 2, and 3	**1, 2, 3**
Select Channel	
Channel on which ECal is to be executed: Channel 1	**Channel: 1**

6. Click **Next**. The EcalAssistant (connection) dialog box appears. Select 3-port/Channel:1.



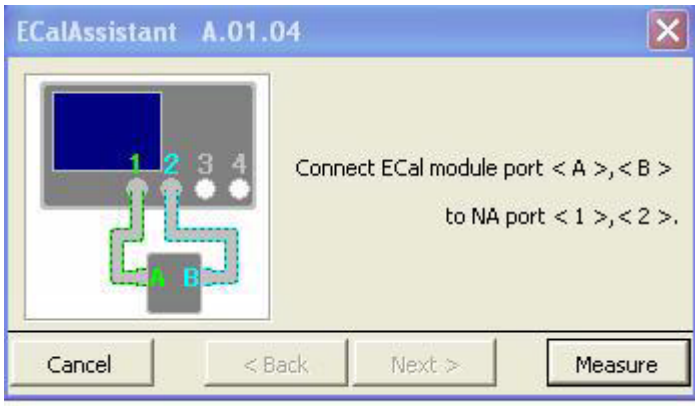
e5071c086

Connecting the ECal module



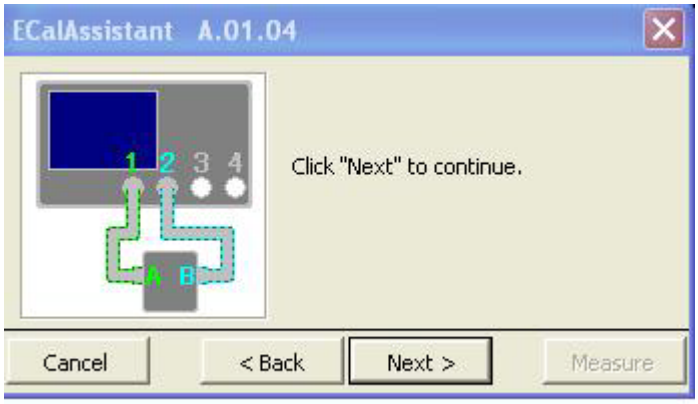
e5071c361

7. Click **Next**. The calibration dialog box for Port 1-2 appears.



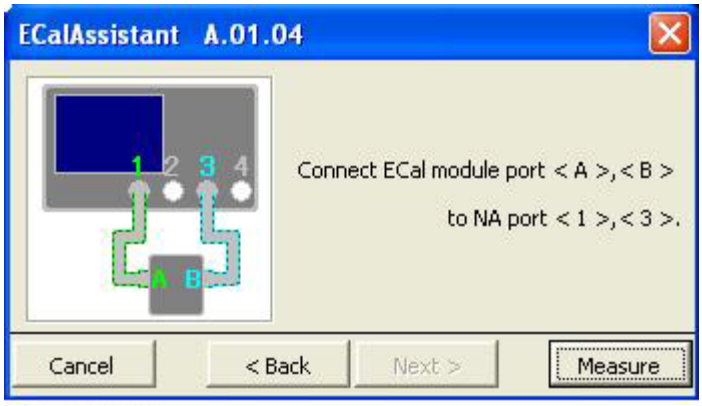
e5071c087

8. Click **Measure** to Calibrate Port 1-2 settings. After successful calibration of Port 1-2, a new dialog box appears.



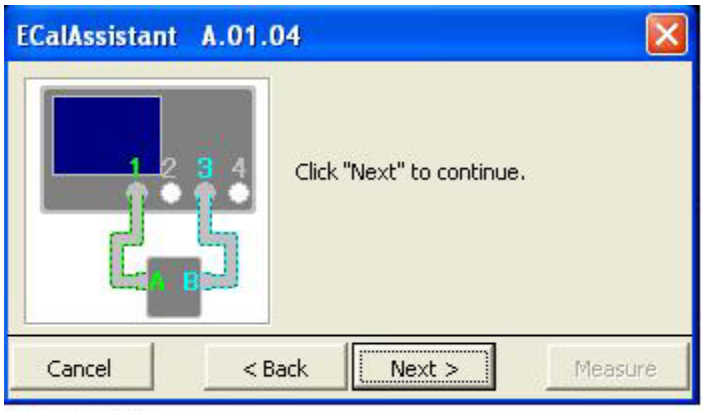
e5071c088

9. Click **Next**. The dialog box for the calibration of port 1-3 appears.



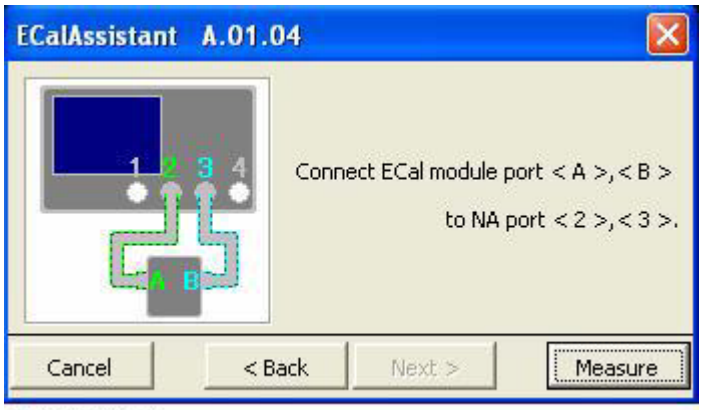
e5071c089

10. Click **Measure** to Calibrate Port 1-3 settings. After successful calibration of Port 1-3, a new dialog box appears.



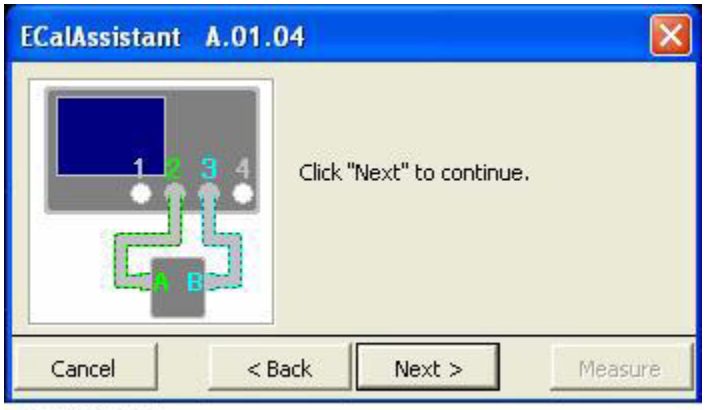
e5071c090

11. Click **Next**. The dialog box for the calibration of port 2-3 appears.



e5071c100

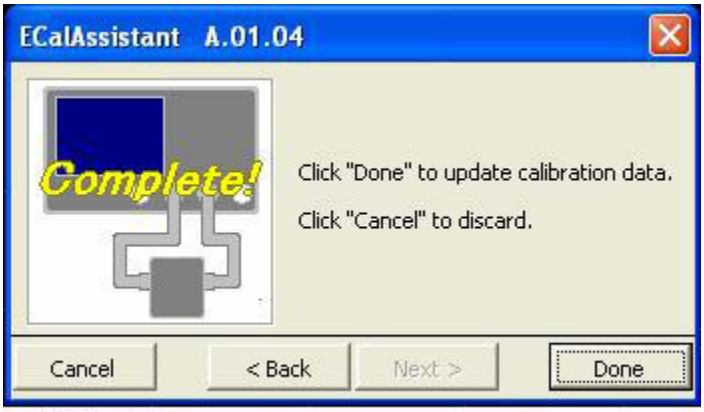
12. Click **Measure** to Calibrate Port 2-3 settings. After successful calibration of Port 2-3, a new dialog box appears.



e5071c098

13. The EcalAssistant (complete) dialog box appears. Click **Done** to terminate the calibration.

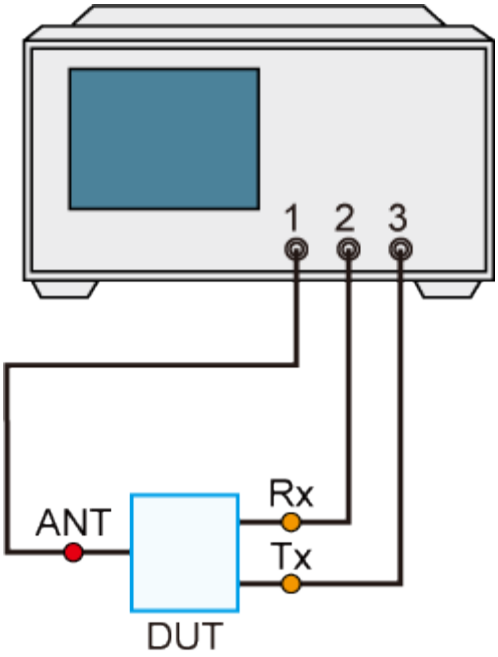
ECalAssistant (complete) dialog box



e5071c099

5. Connect the DUT

The DUT is connected to test ports 1, 2, and 3.



e5071c359

6. Define the Setup for Display

The setup for display is defined.

Setup Description	Key Operation
Number of traces to be displayed: 5	**Display** > **Number of Traces** > **5**

Trace placement: Trisected	**Allocate Traces** >
Measurement Parameter	
Trace 1: S13	**Meas** - **S13**
Trace 2: S21	**Trace Next** > **Meas** > **S21**
Trace 3: S23	**Trace Next** > **Meas** > **S23**
Trace 4: S33	**Trace Next** > **Meas** > **S33**
Trace 5: S11	**Trace Next** > **Meas** > **S11**

7. Execute the Measurement

A trigger is applied to execute the measurement.

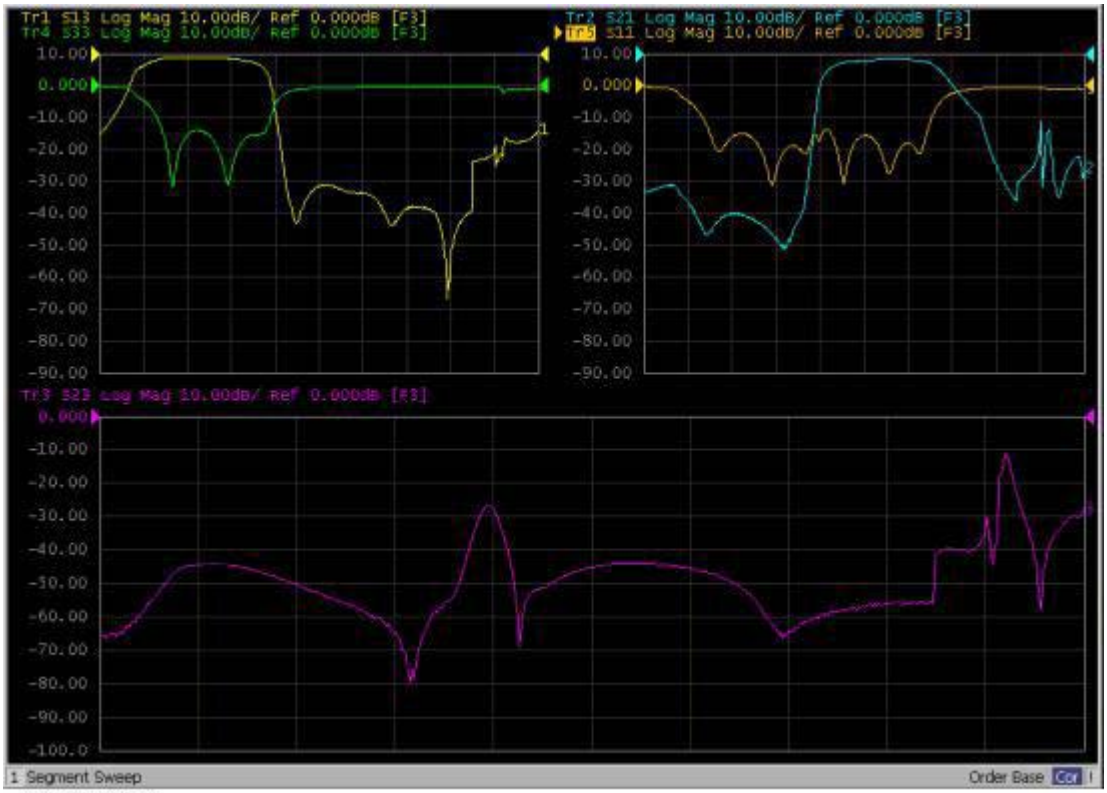
Setup Description	Key Operation
Trigger mode: Single (or continuous)	**Trigger** > **Single** (or **Continuous**)

8. Define the Setup for the Segment Display and Scale

The setup for the scale is defined.

Setup Description	Key Operation
Segment display: Order base	**Sweep Setup** > **Segment Display** > **Order Base**
Reference Line Position	
Trace 1: 10	**Trace Next** > **Scale** > **Reference Position** > **1** > **0** > **x1**
Trace 2: 10	**Trace Next** > **Reference Position** > **1** > **0** > **x1**
Trace 3: 10	**Trace Next** > **Reference Position** > **1** > **0** > **x1**
Trace 4: 9	**Trace Next** > **Reference Position** > **9** > **x1**
Trace 5: 9	**Trace Next** > **Reference Position** > **9** > **x1**

Measurement result (segment display: order base)



9. Analyze the Parameters

The parameters for the duplexer are determined.

1. Determine the insertion loss and 3-dB bandwidth for Tx.

Setup Description	Key Operation
Marker coupling: OFF	**Marker Fctn** > **Couple** (Turn it **OFF**)
Activating Trace 1	**Trace Next**
Marker 1: ON	**Marker**
Search/Tracking: ON	**Marker Search** > **Tracking** (Turn it **ON**)
Moving the marker 1 to the trace maximum	**Max**
Bandwidth search: ON	**Bandwidth** (Turn it **ON**)

The insertion loss (**loss**) in this example is -1.243 dB, and the 3-dB bandwidth (**BW**) is 85.53 MHz.

2. Determine the insertion loss and 3-dB bandwidth for the Rx.

Setup Description	Key Operation
Activating Trace 2	**Trace Next**
Marker 1: ON	**Marker**
Search/Tracking: ON	**Marker Search** > **Tracking** (Turn it **ON**.)
Moving Marker 1 to the trace maximum	**Max**
Bandwidth search: ON	**Bandwidth** (Turn it **ON**)

The insertion loss (**loss**) in this example is -1.627 dB and the 3-dB bandwidth (**BW**) is 71.04 MHz.

3. Determine the isolation between Tx and Rx.

Setup Description	Key Operation
Activating Trace 3	**Trace Next**
Marker 1: ON	**Marker**
Search/Tracking: ON	**Marker Search** > **Tracking** (Turn it **ON**)
Moving Marker 1 to the peak near 1.92 GHz	**Peak** - **Search Left** or **Search Right** (press as many times as necessary)

The isolation (response value of marker 1) in this example is -6.612 dB.

4. Determine the return loss of Tx.

Setup Description	Key Operation
Activating Trace 4	**Trace Next**
Marker 1: ON	**Marker**
Search/Tracking: ON	**Marker Search** > **Tracking** (Turn it **ON**)
Moving Marker 1 to the peak in the pass band	**Peak** > **Search Left** or **Search Right** (press as many times as necessary)

The return loss (response value of Marker 1) in this example is 12.65 dB.

5. Determine the return loss of Rx.

Setup Description	Key Operation
Activating Trace 5	**Trace Next**
Marker 1: ON	**Marker**
Search/Tracking: ON	**Marker Search** > **Tracking** (Turn it **ON**)
Moving Marker 1 to the peak in the pass band	**Peak** > **Search Left**|**Search Right** (press as many times as necessary)

The return loss (response value of Marker 1) in this example is 13.80 dB.

6. Turn on the marker table display.

Setup Description	Key Operation
Marker table display: ON	**Marker Fctn** > **Marker Table** (Turn it **ON**)

Activating Marker Table

e5071c092

10. Define the Setup for a Limit Table

Follow the steps below to make entries in the limit table.

1. Display the limit table for Trace 1 (S13).

Setup Description	Key Operation
Activating Trace 1	**Trace Next**
Displaying a limit table	**Analysis** > **Limit Test** > **Edit Limit Line**

2. Enter the setup data in the limit table for trace 1.

Completed limit table for trace 1

	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	1.730000000 GHz	1.940000000 GHz	0 dB	0 dB
2	MIN	1.850000000 GHz	1.910000000 GHz	-4 dB	-4 dB
3	MAX	1.930000000 GHz	1.990000000 GHz	-35 dB	-35 dB
4	MAX	1.990000000 GHz	2.130000000 GHz	-30 dB	-30 dB
5	MAX	2.130000000 GHz	6.020000000 GHz	-20 dB	-20 dB
6					

e5071c093

3. Display the limit table for Trace 2 (S21).

Setup Description	Key Operation
Activating Trace 2	**Trace Next**

4. Enter the setup data in the limit table for trace 2.

Completed limit table for trace 2

	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	1.730000000 GHz	1.850000000 GHz	-40 dB	-40 dB
2	MAX	1.850000000 GHz	1.910000000 GHz	-45 dB	-45 dB
3	MAX	1.910000000 GHz	6.020000000 GHz	0 dB	0 dB
4	MIN	1.930000000 GHz	1.990000000 GHz	-5 dB	-5 dB
5					

e5071c094

11. Execute the Limit Test

The limit test is executed.

1. Turn on the limit line and limit test for Trace 1.

Setup Description	Key Operation
Activating Trace 1	**Trace Prev**
Limit Line: ON	**Analysis** > **Limit Test** - **Limit Line** (Turn it **ON**)

Limit Test: ON	**Limit Test** (Turn it **ON**)

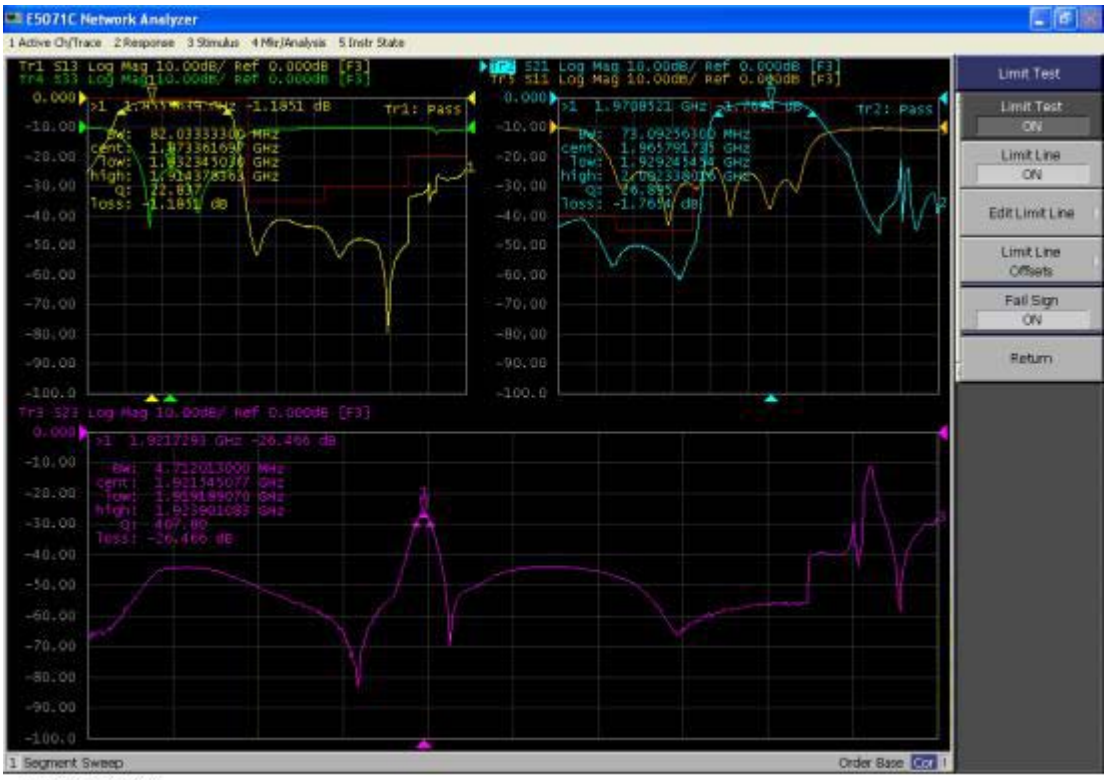
2. Turn on the limit line and limit test for Trace 2.

Setup Description	Key Operation
Activating Trace 2	**Trace Next**
Limit Line: ON	**Limit Line** (Turn it **ON**)
Limit Test: ON	**Limit Test** (Turn it **ON**)

3. Apply a trigger to execute the measurement.

Setup Description	Key Operation
Trigger Mode: Single	**Trigger** > **Single** (or **Continuous**)

Limit test results

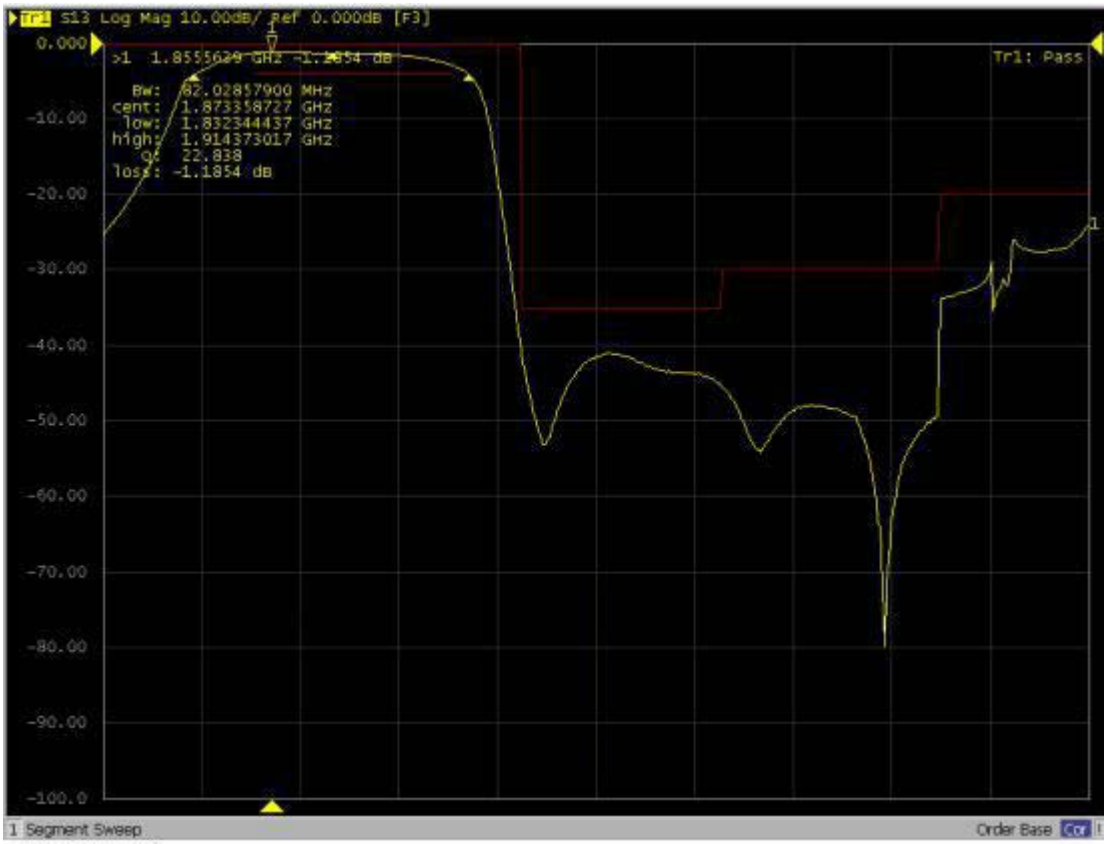


e5071c095

4. Maximize the screen display of Trace 1 to examine its details.

Setup Description	Key Operation
Activating Trace 1	**Trace Prev**
Maximizing the display of Trace 1	**Trace Max**

Enlarged display of trace 1

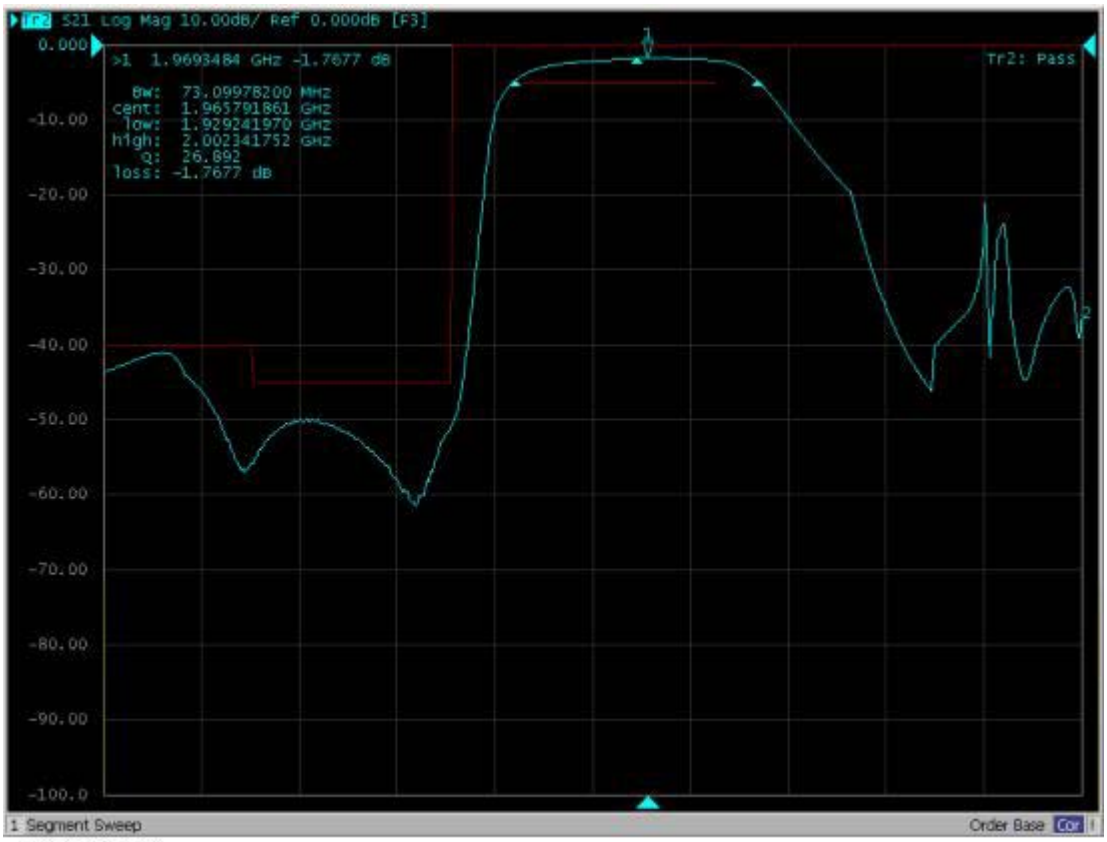


e5071c096

5. Maximize the screen display of Trace 2 to examine its details.

Setup Description	Key Operation
Activating Trace 2	**Trace Prev** (The display of Trace 2 is maximized.)

Enlarged display of trace 2


Tr2 S21 Log Mag 10.00dB/ Ref 0.000dB [F3]
0.000
-10.00
-20.00
-30.00
-40.00
-50.00
-60.00
-70.00
-80.00
-90.00
-100.0
>1 1.9693484 GHz -1.7677 dB
Bw: 73.09978200 MHz
cent: 1.965791861 GHz
low: 1.929241970 GHz
high: 2.002341752 GHz
Q: 26.892
loss: -1.7677 dB
Tr2: Pass
1 Segment Sweep
Order Base Cor

e5071c097

Measuring the Deviation from a Linear Phase

- Overview
- Procedure

Other Measurement Examples

Overview

This section illustrates how to determine the deviation from a linear phase in the pass band of a 1.09-GHz bandpass filter.

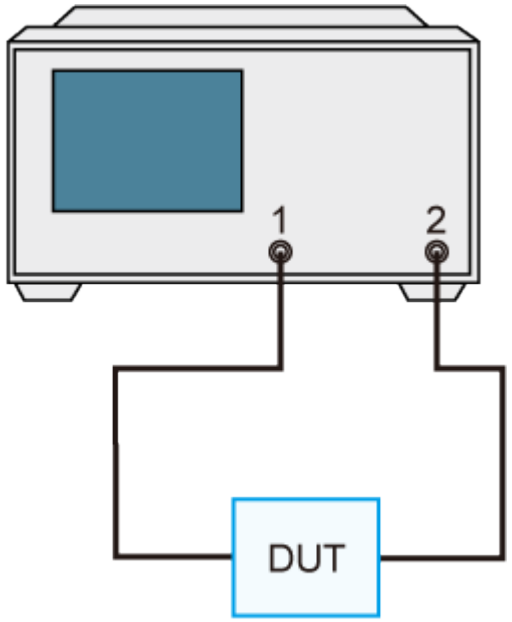
Procedure

Here, the DUT is evaluated by following the steps.

Step	Description
1. Connect the DUT	The DUT is connected.
2. Define the Measurement Conditions	The measurement conditions are defined.
3. Execute the Calibration	The calibration is executed.
4. Connect the DUT and Execute the Auto Scale	The DUT is connected again to execute the auto scale function.
5. Specify the Electrical Delay	The electrical delay is specified.
6. Measure the Deviation from a Linear Phase	The statistics data function (peak-to-peak) is used to determine the deviation from a linear phase.

1. Connect the DUT

Connect the DUT as shown below.



e5071c320

2. Define the Measurement Conditions

The measurement conditions are defined by following the steps described below.

Setup Description	Key Operation
Presetting	**Preset** > **OK**
Center frequency: 1.09 GHz	**Center** > **1** > **.** > **0** > **9** > **G/n**
Frequency span: 20 MHz	**Span** > **2** > **0** > **M/u**
Measurement parameter: S21	**Meas** > **S21**
Data format: Expand Phase	**Format** > **Expand Phase**
Executing the Auto Scale	**Scale** > **Auto Scale**

3. Execute the Calibration

The THRU response calibration is executed.

Setup Description	Key Operation
Executing the THRU response	(A THRU standard is connected instead of a DUT) **Cal** > **Calibrate** > **Response (Thru)** > **Thru** >

	Done

4. Connect the DUT and Execute the Auto Scale

The DUT is connected again as shown in Connect the DUT to execute the auto scale.

Setup Description	Key Operation
Executing the auto scale	**Scale** > **Auto Scale**

e5071c113

5. Specify the Electrical Delay

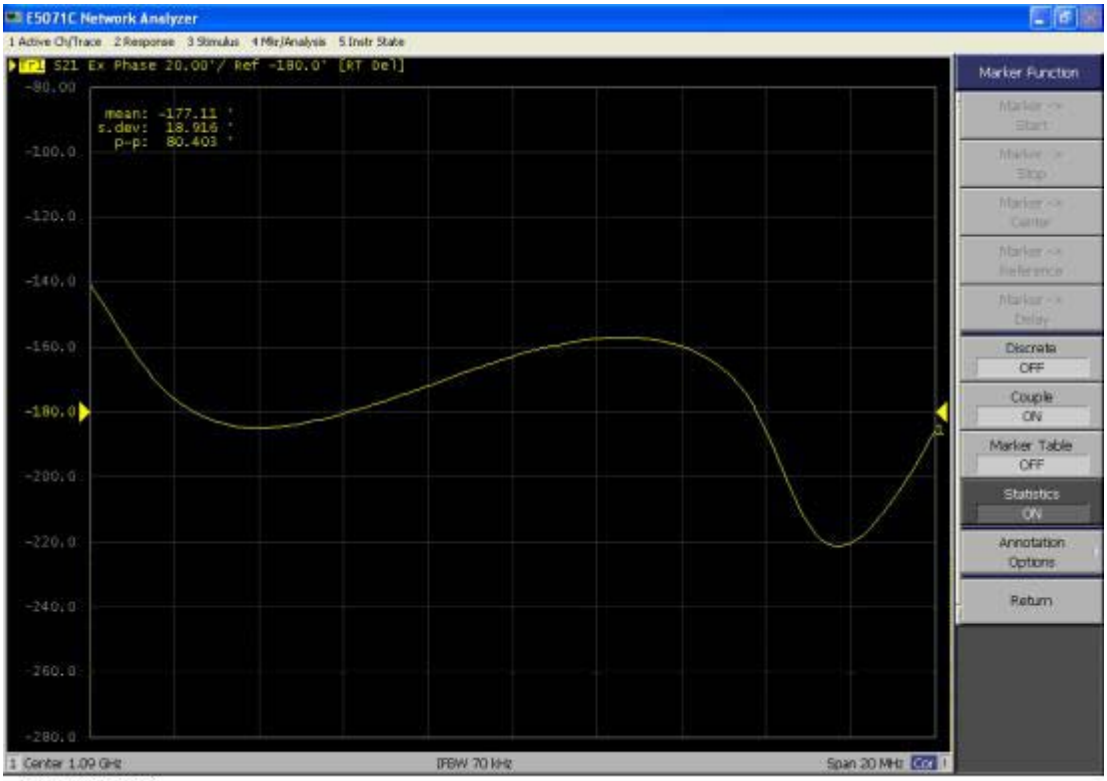
The electrical delay is entered to flatten the phase trace.

Setup Description	Key Operation
Entering the electrical delay	**Scale** > **Electrical Delay** > **Up**/**Down Arrow** keys or Rotary Knob (Flattening a trace)

6. Measure the Deviation from a Linear Phase

The statistics data is used to read the deviation from a linear phase (peak-to-peak).

Setup Description	Key Operation
Executing the auto scale	**Scale** > **Auto Scale**
Displaying the statistics data	**Marker Function** > **Statistics** (Turn it **ON**.)



e5071c114

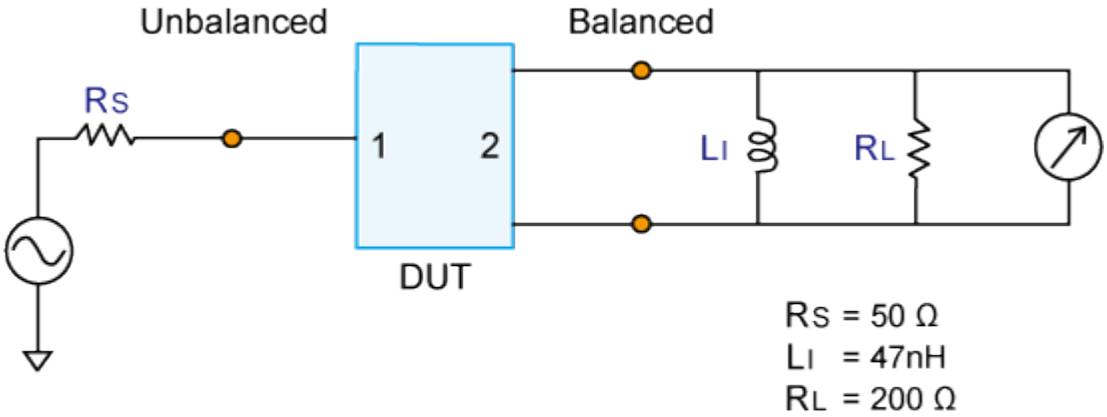
Measuring Unbalanced-Balanced Bandpass Filter

- Overview
- Procedure

Other Measurement Examples

Overview

This section introduces an example of actually evaluating the unbalanced and balanced SAW bandpass filter with a center frequency of 942.5 MHz. The following figure shows the measurement circuit in the condition for evaluating a DUT.



e5071c378

Procedure

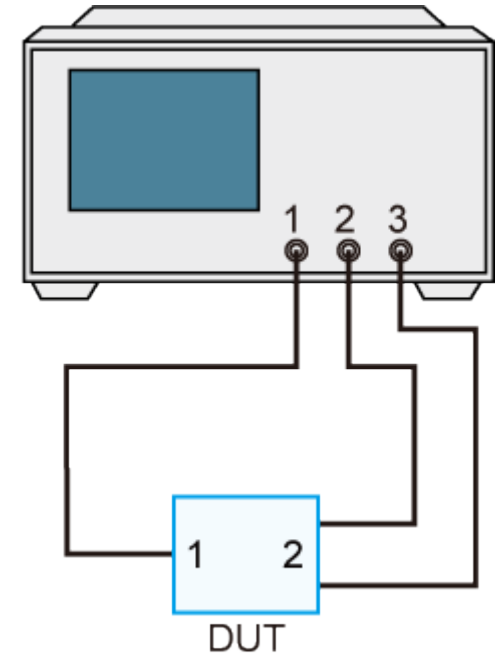
Here, the DUT is evaluated by following the steps.

Step	Description
1. Connecting the DUT	The DUT is connected.
2. Setting the Measurement Conditions	The measurement conditions are defined.
3. Performing Calibration	The full 3-port calibration is executed.
4. Setting a Balance Conversion Topology	The balance conversion topology is specified.

5. Selecting Measurement Parameters	The mixed-mode S-parameters are selected.
6. Extending the Calibration Plane (removing the cause of error)	The calibration reference plane is extended.
7. Setting the Port Reference Impedances	The port reference impedances are specified.
8. Adding a Matching Circuit	A matching circuit is added.

1. Connecting the DUT

Connect the DUT to the E5071C by using the instrument's three test ports.



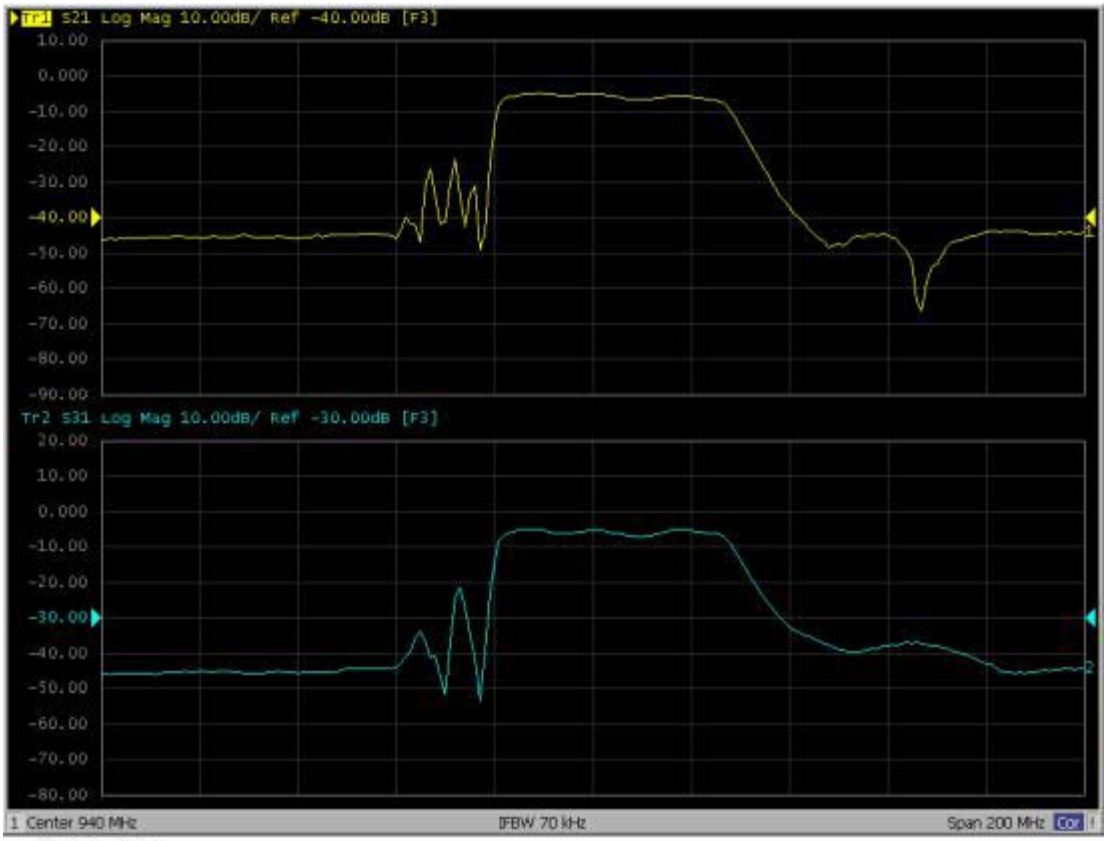
e5071c358

2. Setting the Measurement Conditions

Follow the procedure below to set the measurement conditions. The measurement parameters for balanced measurements should be set after unbalanced-balanced conversion. Here, set the measurement parameters for observing the characteristics achieved during unbalanced measurements.

Setting Description	Key Operation
Preset for setting	**Preset** > **OK**

Center frequency: 940 MHz	**Center** > **9** > **4** > **0** > **M/u**
Frequency span: 200 MHz	**Span** > **2** > **0** > **0** > **M/u**
Number of traces: 2	**Display** > **Num of Traces** > **2**
Trace-1 measurement parameter: S21	**Meas** > **S21**
Trace-2 measurement parameter: S31	**Trace Next** > **Meas** > **S31**
Allocate a trace to upper and lower displays	**Display** > **Allocate Traces** >
Auto-scale all traces	**Scale** > **Auto Scale All**



e5071c104

3. Performing Calibration

Perform a full three-port calibration for the three ports to be used.

1. Set the type and conditions of calibration.

Setting Description	Key Operation
Calibration kit to use: 85033D	**Cal** > **Cal Kit** > **85033D**
Type of calibration: Full three port calibration	**Calibrate** > **3-Port Cal**
Test ports to calibrate: 1, 2, 3	**Select Ports** > **1-2-3** (check only)

2. Perform a reflection calibration.

Setting Description	Key Operation
Select reflection calibration	**Reflection**
Perform Port 1 calibration	(With the OPEN connected) **Port 1 OPEN** (With the SHORT connected) **Port 1 SHORT** (With the LOAD connected) **Port 1 LOAD**
Perform Port 2 calibration	(With the OPEN connected) **Port 2 OPEN** (With the SHORT connected) **Port 2 SHORT** (With the LOAD connected) **Port 2 LOAD**
Perform Port 3 calibration	(With the OPEN connected) **Port 3 OPEN** (With the SHORT connected) **Port 3 SHORT** (With the LOAD connected) **Port 3 LOAD**

3. Perform a transmission calibration.

Setting Description	Key Operation
Select transmission calibration	**Return** > **Reflection**
Perform a Port 1-to-Port 2 calibration	(With thru connection) **Port 1-2 Thru**
Perform a Port 1-to-Port 3 calibration	(With thru connection) **Port 1-3 Thru**
Perform a Port 2-to-Port 3 calibration	(With thru connection) **Port 2-3 Thru**

4. Finish the calibration.

Setting Description	Key Operation
Complete the calibration and then calculate and store calibration coefficients.	**Return** > **Done** (This causes **Correction** to turn **ON**.)
Calibration property display: ON	**Return** > **Return > Property** (Turns it **ON**.)

4. Setting a Balance Conversion Topology

Follow the procedure below to set the balanced conversion topology.

Setting Description	Key Operation
Set port 1 on the DUT to unbalanced and port 2 on the DUT to balanced.	**Analysis** > **Fixture Simulator > Topology > Device > SE-Bal** (check only)
Set the connecting destination of port 1 on the DUT (unbalanced) to test port 1 of the analyzer.	**Port 1 (se)** > **1** (check only)
Set the connecting destination of port 2 on the DUT (balanced) to test ports 2 and 3 of the analyzer.	**Port 2 (bal)** > **2-3** (check only)

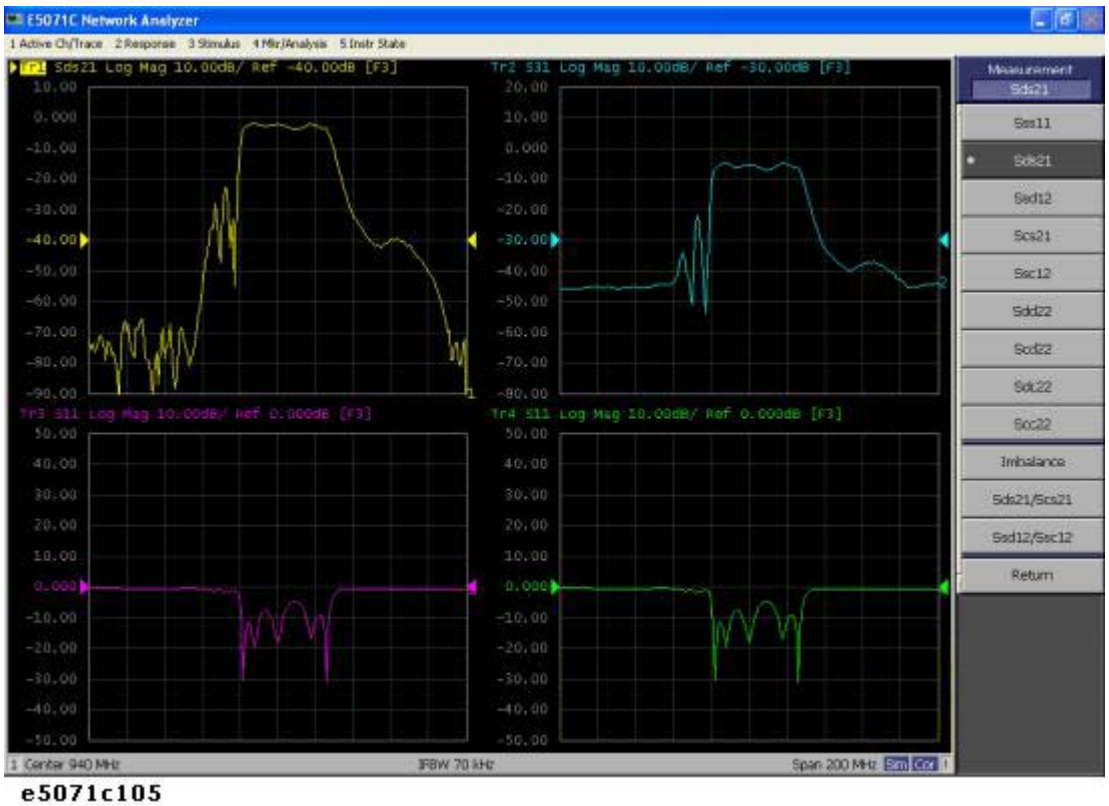
5. Selecting Measurement Parameters

1. Display four traces.

Setting Description	Key Operation
Number of traces: 4	**Display** > **Number of Traces > 4**
Trace allocation: 4-part split	**Allocate Traces** >

2. Set the measurement parameter (mixed mode S-parameter) and data format for trace 1.

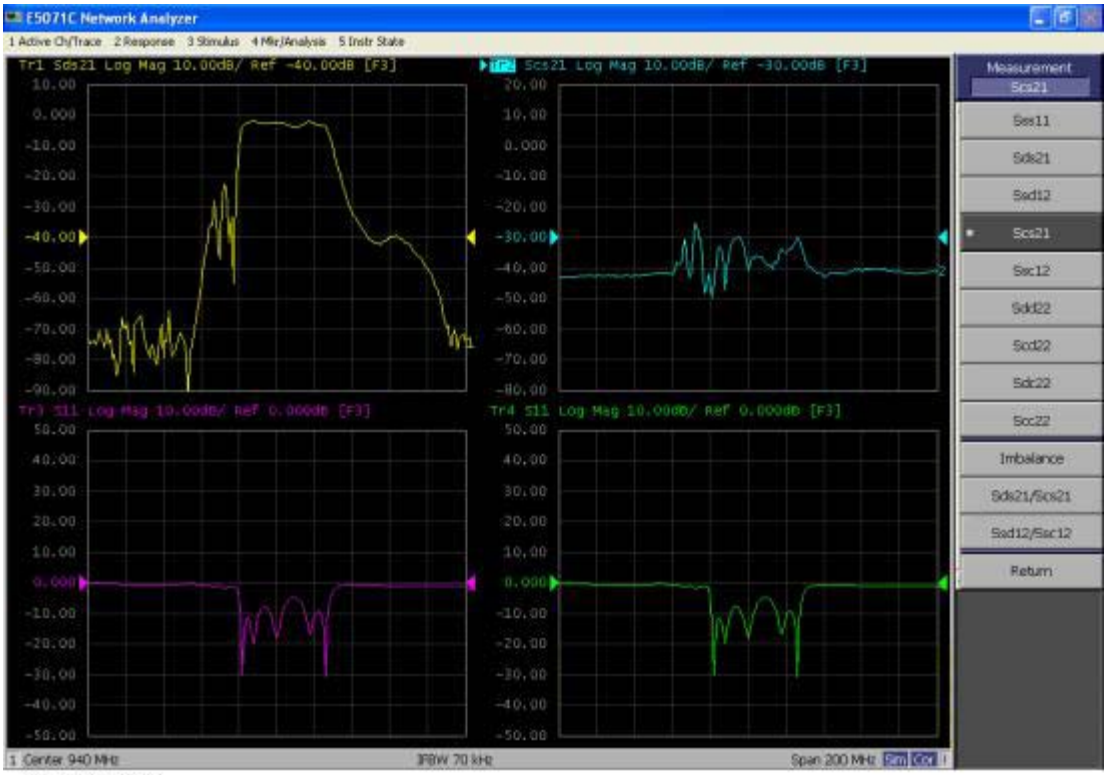
Setting Description	**Key Operation**
Fixture simulator: ON	**Analysis** > **Fixture Simulator** > **Fixture Simulator** (turns it **ON**)
Unbalanced-balanced conversion of trace 1: ON	**BalUn** (turns it **ON**)
Measurement parameter: Sds21	**Meas** > **Sds21**



e5071c105

3. Set the measurement parameter (mixed mode S-parameter) and data format for trace 2.

Setting Description	Key Operation
Unbalanced-balanced conversion of trace 2: ON	**Trace next** > **Analysis** > **Fixture Simulator** > **BalUn** (turns it **ON**)
Measurement parameter: Scs21	**Meas** > **Scs21**

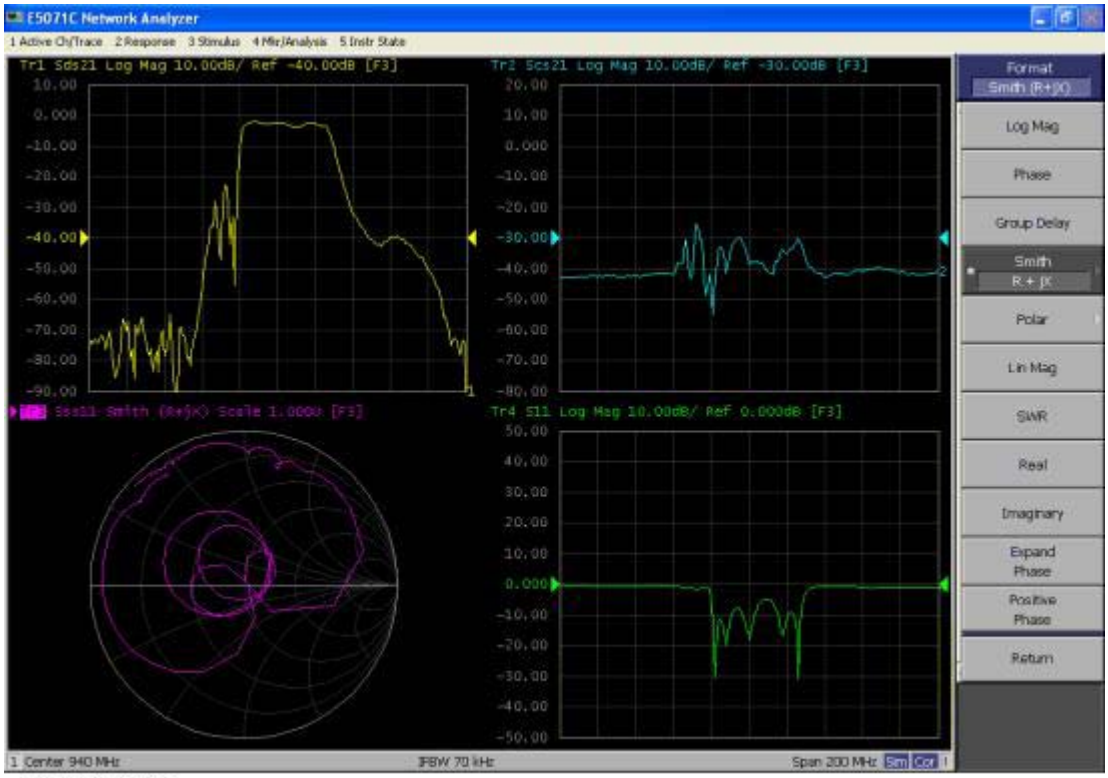


e5071c106

4. Set the measurement parameter (mixed mode S-parameter) and data format for trace 3.

Setting Description	Key Operation
Unbalanced-balanced conversion of trace 3: ON	**Trace next** > **Analysis** > **Fixture Simulator** > **BalUn** (turns it **ON**)
Measurement parameter: Sss11	**Meas** > **Sss11**

Data format: Smith chart (marker display: R+jX)	**Format** > **Smith** > **R + jX**

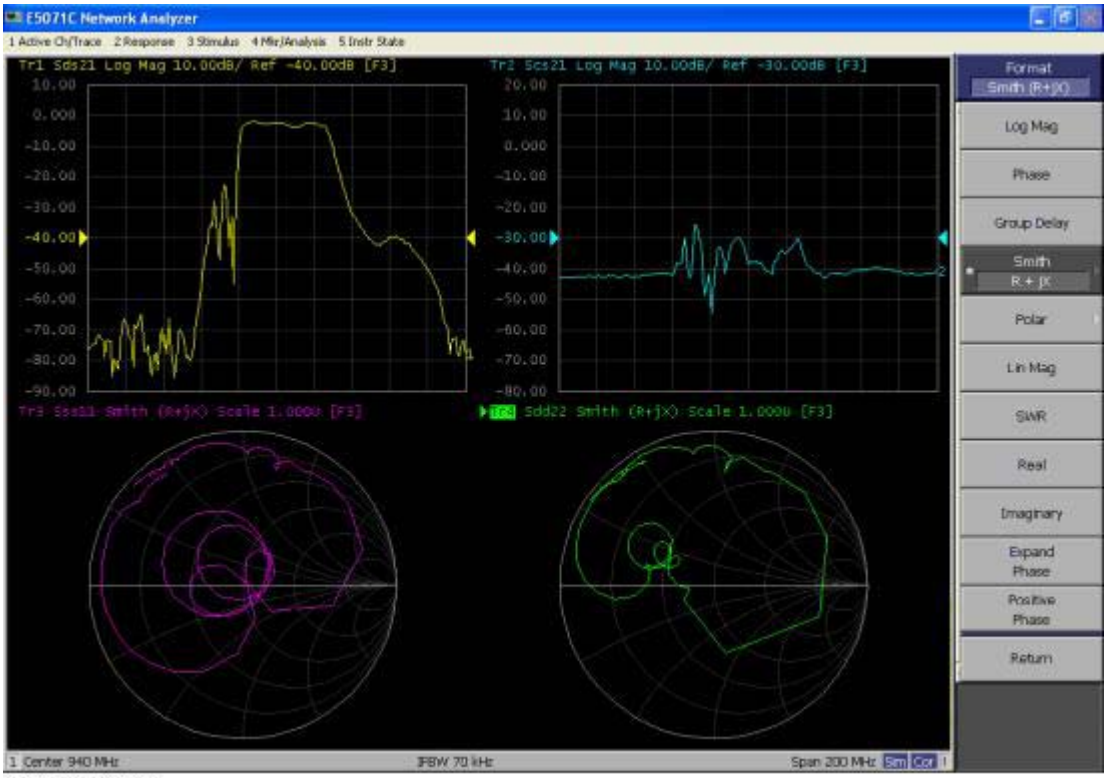


e5071c107

5. Set the measurement parameter (mixed mode S-parameter) and data format for trace 4.

Setting Description	Key Operation
Unbalanced-balanced conversion of trace 4: ON	**Trace next** > **Analysis** > **Fixture Simulator** > **BalUn** (turns it **ON**)
Measurement parameter: Sdd22	**Meas** > **Sdd22**
Data format: Smith chart (marker display: R+jX)	**Format** > **Smith** > **R + jX**

The following figure shows the setting results for each parameter.



e5071c108

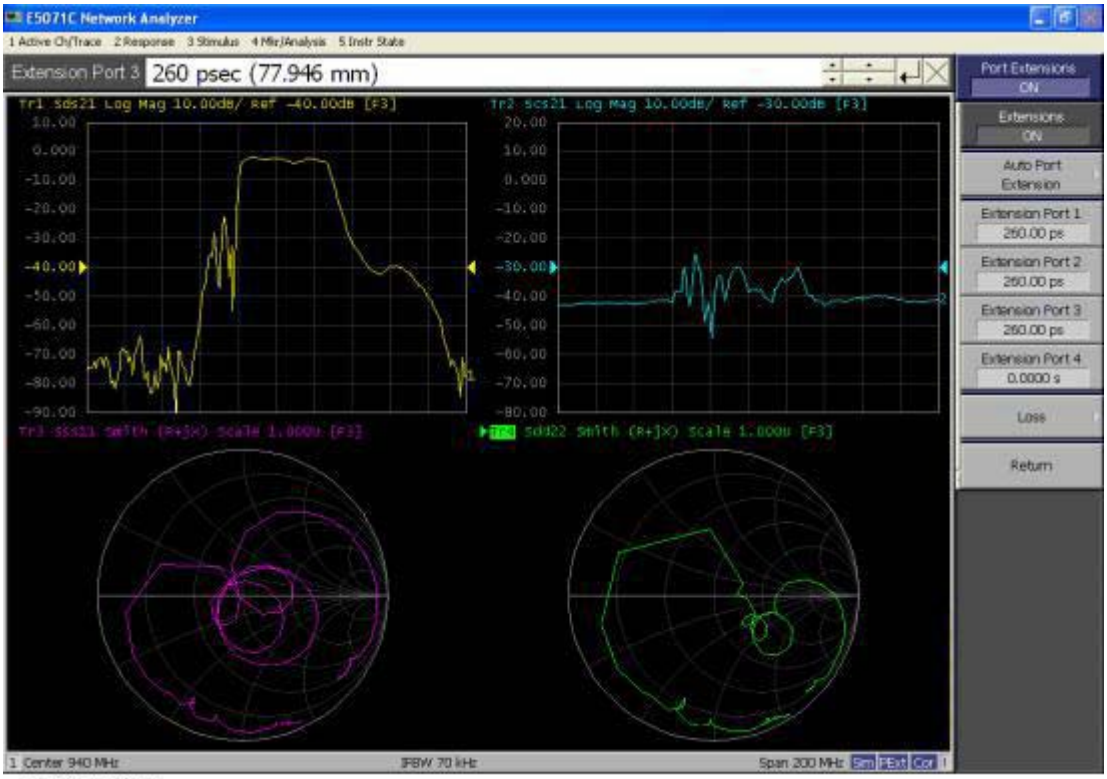
6. Extending the Calibration Plane (removing the cause of error)

In this section you will use the port extension function to remove an electrical delay caused by cables or fixtures located between the calibration reference plane and the DUT to be evaluated. If you can provide a two-port Touchstone data file representing the characteristics of the network to be removed, the network removal function allows you to remove the network and extend the calibration reference plane.

Follow the procedure below to set port extension for each test port.

Setting Description	Key Operation
Port extension of test port 1: 260 ps	**Cal** > **Port Extensions** > **Extension Port 1** > **.** > **2** > **6** > **G/n**
Port extension of test port 2: 260 ps	**Extension Port 2** > **.** > **2** > **6** > **G/n**
Port extension of test port 3: 260 ps	**Extension Port 3** > **.** > **2** > **6** > **G/n**
Port extension: ON	**Extensions** (turns it **ON**)

The following figure shows the results of extending the calibration reference plane.



7. Setting the Port Reference Impedances

With the reference impedances of two test ports in unbalanced measurements set to Z0, conversion of those ports into balanced ports permits the impedance of the balanced ports' common mode to be automatically set to Z0/2 and the impedance of their differential mode to be automatically set to 2Z0.

1. Set the port reference impedance of port 1 on the DUT (unbalanced) to 50 ohm.

Setting Description	Key Operation
Reference impedance of test port 1: 50 ohm	**Analysis** > **Fixture Simulator** > **Port Z conversion** > **Port 1 Z0 Real** > **5** > **0** > **x1**

2. In order to set the impedance of the differential mode of port 2 on the DUT (balanced) to 200 ohm , set the impedances of two unbalanced ports before conversion each to 100 ohm.

Setting Description	Key Operation
Reference impedance of test port 2:	**Port 2 Z0 Real** > **1** > **0** > **0**

100 ohm	> **x1**
Reference impedance of test port 3: 100 ohm	**Port 3 Z0 Real** > **1** > **0** > **0** > **x1**

3. Always set the reference impedances of the two test ports before balanced conversion to the same value.
4. Turn on the port reference impedance conversion function.

Setting Description	Key Operation
Port reference impedance conversion: ON	**Port Z Conversion** (turns it **ON**)

The reference impedance of the command mode of port 2 on the DUT is set to 50 W . The impedance of the differential mode of that port may be set and modified independently of setting the two-port reference impedances before balanced conversion. For more information, see Converting Reference Impedance of Balanced Port.

8. Adding a Matching Circuit

Here, add an inductance of 47 nH in parallel to port 2 on the DUT (balanced). It is also possible to add a matching circuit to the port before unbalanced-balanced conversion. For more information, see Determining the Characteristics that Result from Adding a Matching Circuit to a Differential Port.

- The balance matching circuit (Differential matching circuit embedding) is not applied to single-ended S-parameter results. For example, the balance matching circuit is not applied to the imbalance parameter as it is derived from single-ended S-parameters.

Setting Description	Key Operation
Selecting a matching circuit: Shunt L - Shunt C	**Return** (or **Analysis** > **Fixture Simulator**) > **Diff. Matching** > **Select Circuit** > **Shunt L-Shunt C**

	 e5071c110
Inductance: 47 nH	**L** > **4** > **7** > **G/n**
C=0, G=0, R=0	(checks that **C**, **G**, and **R** have been set to **0**.)
Differential matching circuit function: ON	**Diff. Matching** (turns it **ON**)

Measuring Parameters with Cable

- Overview
- Procedure

Other Measurement Examples

Overview

This section introduces an example of how to detect the location of a mismatch that occurs in a cable by using the time domain function.

Procedure

In this example, a DUT is evaluated according to the steps.

Step	Description
1. Setting the Measurement Conditions	Set the measurement conditions.
2. Executing Calibration	Execute calibration.
3. Connecting the DUT	Connect the DUT.
4. Auto Scale	Execute auto scale.
5. Setting the Time Domain Function	Set the time domain function.

1. Setting the Measurement Conditions

Follow these steps to set the measurement conditions:

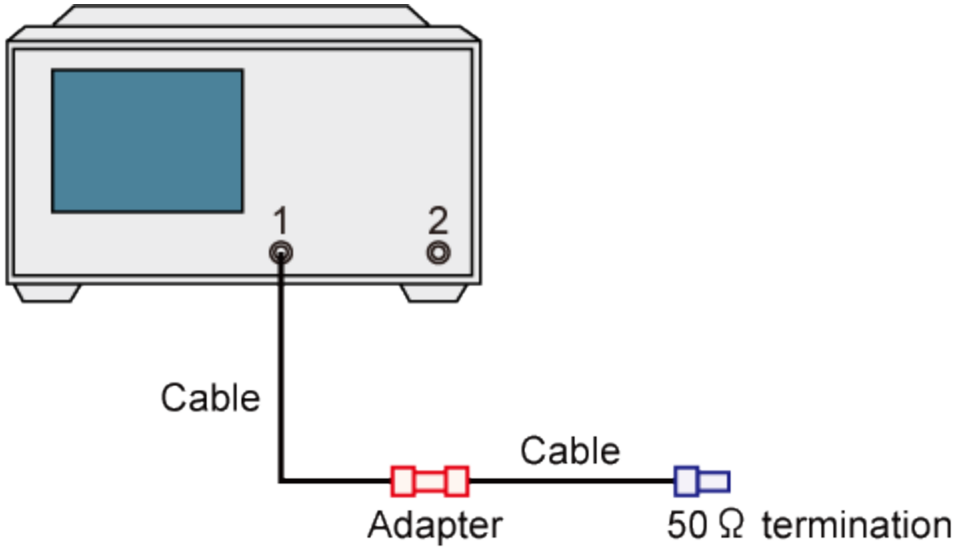
Setting Description	Key Operation
Presetting	**Preset** > **OK**
Stop frequency: 3 GHz	**Stop** > **3** > **G/n**
Number of points: 201	**Sweep Setup** > **Points** > **2** > **0** > **1** > **x1**
Specifying the low-pass mode sweep condition	**Analysis** > **Transform** > **Set Freq Low Pass**
Measurement parameter: S11	**Meas** > **S11**

2. Executing Calibration

By following 1-Port Calibration (reflection test), execute 1-port calibration on port 1.

3. Connecting the DUT

Connect the DUT as shown below.

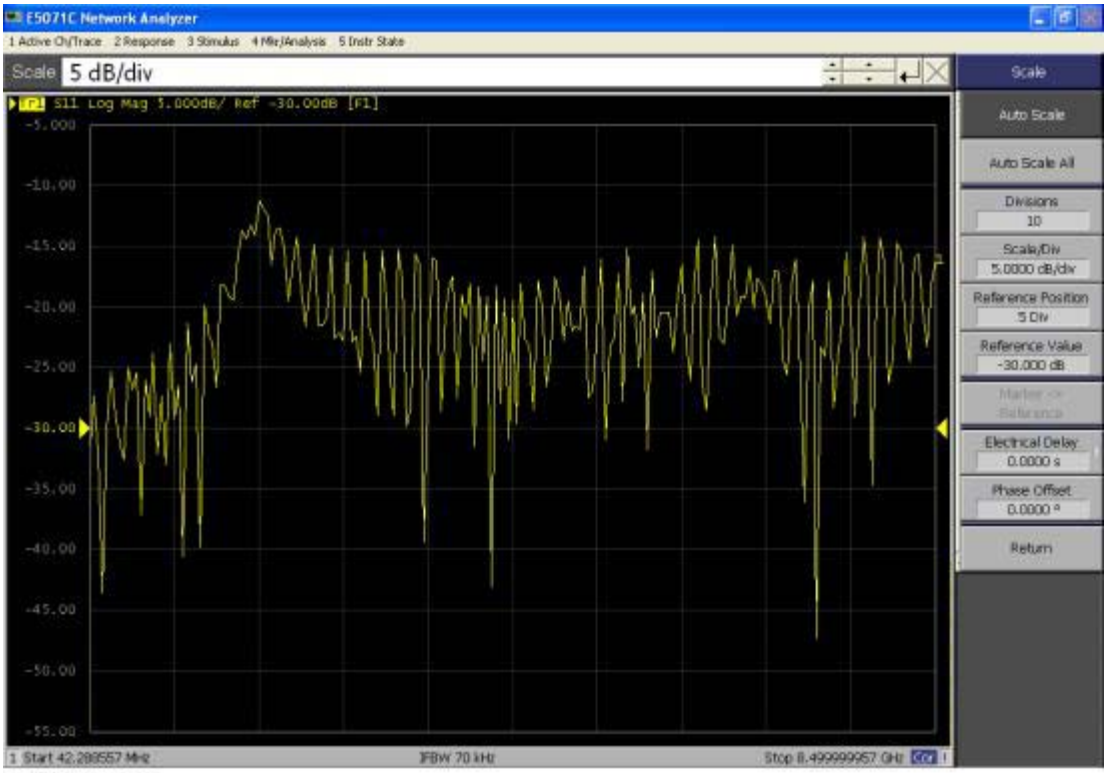


e5071c327

4. Auto Scale

Execute the auto scale function.

Setting Description	Key Operation
Executing auto scale	**Scale** > **Auto Scale**



e5071c111

5. Setting the Time Domain Function

Set the conversion function to display the response in the time domain. If you enable this setting, the response in time domain is displayed as shown figure below. A peak indicating a small mismatch appears at the location of the connector.

Setting Description	Key Operation
Data format: real	**Format** > **Real**
Setting the transformation type to low-pass impulse	**Analysis** > **Transform** > **Type** > **Lowpass Imp**
Setting the window type to maximum.	**Window** > **Maximum**
Setting the display range: from 0 s to 20 ns	**Start** > **0** > **x1**
	Stop > **2** > **0** > **G/n**
Enabling the transformation function	**Transform** (set to **ON**)
Executing auto scale	**Scale** > **Auto Scale**

E5071C Network Analyzer
1 Active Ch/Trace 2 Response 3 Stimulus 4 Mkr/Analysis 5 Instr State
Stop 20 nsec (5.9958 m)
Tr1 S11 Real 10.00mU/ Ref 0.000U [F1]
50.00m
40.00m
30.00m
20.00m
10.00m
0.000
-10.00m
-20.00m
-30.00m
-40.00m
-50.00m
1 Start 0 s
IFBW 70 kHz
Stop 20 ns
Transform
Transform ON
Start 0.0000 s
Stop 20.000 ns
Center 10.000 ns
Span 20.000 ns
Type Lowpass Imp
Window
Return

e5071c112

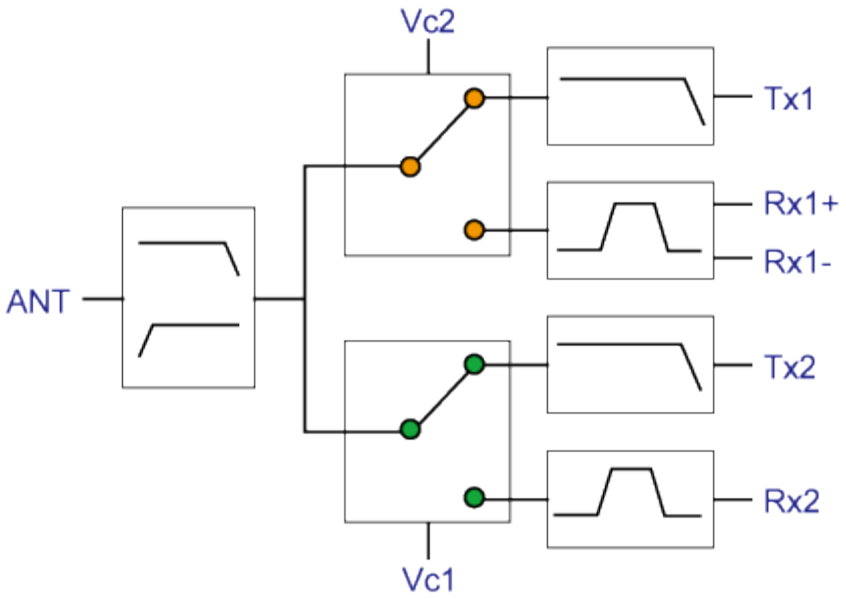
Evaluating Transmission Characteristics of a Front End Module

- Overview
- Procedure

Other Measurement Examples

Overview

This example shows how to measure the transmission characteristics of a 6-port front end module, as shown in the following figure, by using the E5071C and the E5091A.



e5071c472

Procedure

In this example, a DUT is evaluated according to the procedure shown in table below.

Procedure	Description
1. Determining Measurement Conditions	Determine the measurement conditions such as the sweep conditions and measurement ports.
2. Setting Channel Window Allocation	Set the allocation of the channel windows on the screen.
3. Setting the Test Ports	Determine the test port assignment for each channel.

4. Setting Control Line	Set the E5091A's control line.
5. Setting Sweep Conditions	Set the sweep range and the number of points.
6. Setting Balance Conversion Topology	Set the balance port and unbalance port assignment.
7. Selecting Measurement Parameter	Set the measurement parameter.
8. Executing Calibration	Perform calibration using 4-port ECal.
9. Connecting DUT	Connect the DUT.
10. Executing Measurement	Execute the measurement and perform auto scale.

1. Determining Measurement Conditions

In this example, perform measurement under the measurement conditions in table below.

Ch.	Start frequency	Stop frequency	NOP	Test port assignment	Control line	Meas. parameter	Calibration	
							Type	Port
1	400 MHz	1.4 GHz	201	Port 1 - Port A Port 2 - Port T1 Port 3 - Port R1+ Port 4 - Port R1-	Line 1: Low Line 2: High	S12	Full 2-Port	1, 2
2	880 MHz	1 GHz	101		Line 1: Low Line 2: Low	Sds21	Full 3-Port	1, 3, 4
3	1.34 GHz	2.34 GHz	201	Port 1 - Port A Port 2 - Port T2 Port 3 -	Line 1: High Line 2:	S12	Full 2-Port	1, 2

				Port R2+ Port 4 - Port R1-	Low			
4	1.665 GHz	2.015 GHz	101		Line 1: Low Line 2: Low	S31	Full 2-Port	1, 3

2. Setting Channel Window Allocation

Set the screen to split into 2 rows and 2 columns to assign channel windows after preset.

Setting Description	Key Operation
Execute preset	**Preset** > **OK**
Allocate channel windows	**Display** > **Allocate Channels** >

3. Setting the Test Ports

1. Display the E5091A setup menu and select the 9-port model for ID1.

Setting Description	Key Operation
Display the E5091A setup menu	**System** > **Multiport Test Set Setup**
Select the 9-port model for ID1	**Test Set 1 > Select Test Set > E5091_9**

2. Display the E5091A properties.

Setting Description	Key Operation
Display the E5091A properties	**Property**

3. Select the test ports assigned to ports 1 to 4 for channel 1.

Setting Description	Key Operation
Assign test port A to port 1.	**Port1** > **A**

Assign test port T1 to port 2.	**Port2** > **T1**
Assign test port R1+ to port 3.	**Port3** > **R1+**
Assign test port R1- to port 4.	**Port4** > **R1-**

4. Assign test ports for channels 2, 3, and 4. Press the **Channel Next** key to switch the active channel and then make the same setting of channel 1.
5. Enable the control of the E5091A.

Setting Description	Key Operation
Enable the control of the E5091A.	**Control** (set it to **ON**)

4. Setting Control Line

1. Set the bits of the control line for channel 1.

Setting Description	Key Operation
Set channel 1 to the active channel.	**Channel Next**
Line 1: Low	**Control Lines** (check only)
Line 2: High	**Control Lines** > **Line 1** (set it to **High**)

2. Set the control line for channels 2, 3, and 4 according to the same procedure of channel 1.

5. Setting Sweep Conditions

1. Set the sweep conditions for channel 1.

Setting Description	Key Operation
Set channel 1 to the active channel.	**Channel Next**
Start frequency: 400 MHz	**Start** > **4** > **0** > **0** > **M/u**
Stop frequency: 1.4 GHz	**Stop** > **1** > **.** > **4** > **G/n**

Number of points: 201	**Sweep Setup** > **Point** > **2** > **0** > **1** > **x1**

2. Set sweep conditions for channels 2, 3, and 4 according to the same setting of channel 1.

6. Setting Balance Conversion Topology

For channel 2, set the balance conversion topology in order to perform measurement that includes the balanced port.

Setting Description	Key Operation
Set channel 2 to the active channel.	**Channel Next** (press it until channel 2 is activated)
Set DUT port 1 to unbalance and DUT port 2 to balance.	**Analysis** > **Fixture Simulator > Topology > Device > SE-Bal**
Set the destination to which DUT port 1 (unbalance) is connected to test port 1 of the analyzer.	**Port 1 (se)** > **1**
Set the destination to which DUT port 2 (balance) is connected to test ports 3 and 4 of the analyzer.	**Port 2 (bal)** > **3-4**
Unbalance-balance conversion for trace 1: ON	**BalUn** (set it to **ON**)
Fixture simulator: ON	**Fixture Simulator** (set it to **ON**)

7. Selecting Measurement Parameter

1. Select the measurement parameter for trace 1 of channel 1.

Setting Description	Key Operation
Set channel 1 to the active channel.	**Channel Next** (press it until channel 1 is activated)
Measurement parameter for trace 1: S12	**Meas** > **S12**

2. The subscript of the measurement parameter means the test port of the E5071C. Check the test port assignment and select the measurement parameter.

2. Set measurement parameter for channels 2, 3, and 4 according to the same setting of channel 1.

8. Executing Calibration

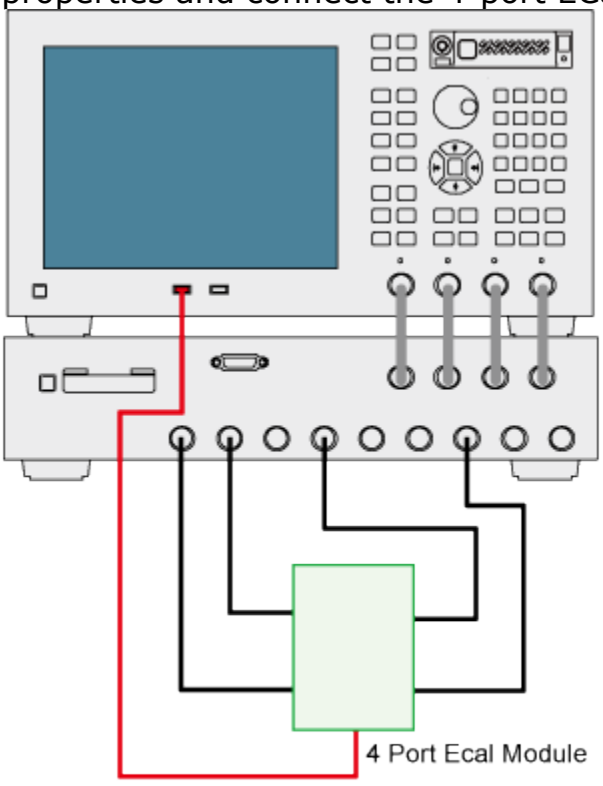
1. Display the Ecal menu.

Setting Description	Key Operation
Display the ECal menu.	**Cal** > **ECal**

2. Set channel 1 to the active channel.

Setting Description	Key Operation
Switch the active channel.	**Channel Next**

3. Check the test ports assigned to ports 1 to 4 in the E5091A properties and connect the 4-port ECal module to those port.



e5071c413

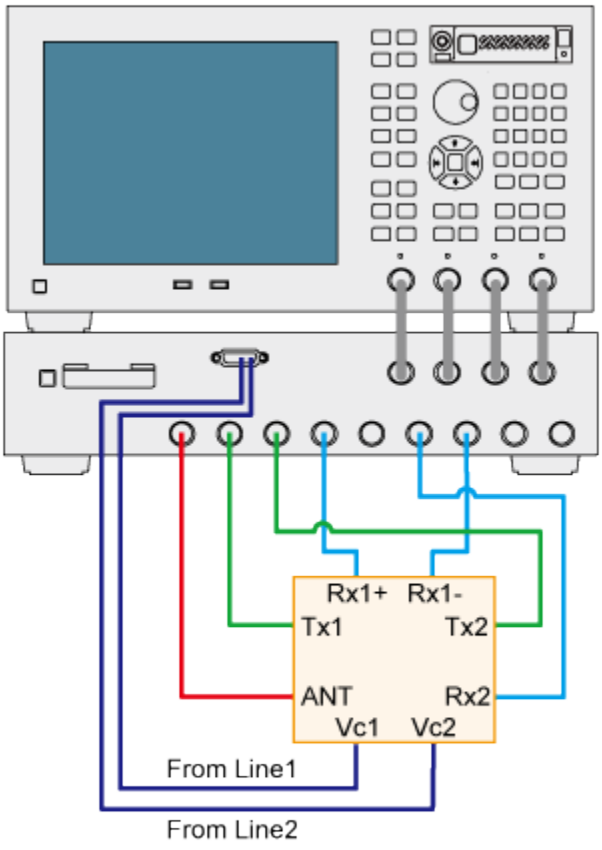
4. Execute the calibration.

Setting Description	Key Operation
Select full 2-port calibration.	**2-Port ECal**
Select the port and execute the calibration.	**1-2**

5. Perform calibration for channels 2, 3, and 4 according to the same procedure as in Step 2 to Step 4.
    - Because the test port assignment setting for channels 1 and 2 and that for channels 3 and 4 are the same, you need not change the ECal connection.

9. Connecting DUT

Connect the DUT as shown below.



e5071c412

10. Executing Measurement

1. Display the trigger menu.

Description	Key Operation

Display the trigger menu.	**Trigger**

2. Set the trigger source to "manual."

Setting Description	Key Operation
Set the trigger source to "manual."	**Trigger Source** > **Manual**

3. Set the trigger mode for channel 1 to "continuous."

Setting Description	Key Operation
Set channel 1 to the active channel.	**Channel Next**
Set the trigger mode to "continuous."	**Continuous**

4. Set the trigger mode for channels 2, 3, and 4 to "continuous" according to the same procedure as in Step 3.
5. Execute the measurement.

Setting Description	Key Operation
Generate a trigger event.	**Trigger**

6. Repeat the following procedure to execute auto scale for all of the channels.

Setting Description	Key Operation
Set the active channel.	**Channel Next**
Execute auto scale.	**Scale** > **Auto Scale**

Executing Power Calibration

- Overview
- Procedure

Other Measurement Examples

Overview

This section shows an example of executing power calibration using the E4418B power meter and the E4412A power sensor.

- Power calibration may not be executed when the specified frequency and level is out of range of the power sensor specification.

Procedure

In this example, power calibration is executed according to the steps shown in table below.

Step	Description
1. Connecting Power Meter	Connects the power meter to the E5071C.
2. Selecting Power Meter a. Setting Address of GPIB Power Meter b. Setting USB Power Meter	Configures the power meter's USB/GPIB interface and GPIB address with the E5071C.
3. Setting Stimulus Condition	Sets conditions such as the power level and frequency.
4. Executing Zero Adjustment and Calibration of Power Sensor	Executes zero adjustment and calibration of the power sensor.
5. Setting Calibration Data Measurement Conditions	Selects the port, selects the power sensor, sets the number of measurements at one point, and sets the tolerance during power calibration.
6. Connecting Power	Connects the power sensor.

Sensor	
7. Measuring Calibration Data	Executes the measurement of calibration data.

1. Connecting Power Meter

Connect the GPIB connector on the E4418B (GPIB address: 14) to the E5071C through USB/GPIB Interface.

- From Firmware revision 9.2, E5071C supports USB enabled power sensors. When you connect USB enabled power sensor for the first time, new hardware wizard runs. For more information, refer to Setting system controller (USB/GPIB interface)

2. Selecting Power Meter

Follow the steps to select the power meter

Setting Description	Key Operation
USB/GPIB	**System** > **Misc Setup** > **Power Meter Setup** > **Select Type** > **USB/GPIB**

Setting Address of GPIB Power Meter

Follow these steps to configure the power meter's GPIB address.

Setting Description	Key Operation
GPIB address of the power meter: 14	**System** > **Misc Setup** > **GPIB Setup** > **Power Meter Address** > **1** > **4** > **x1**

- You can also set the GPIB address of power meter by pressing **System** > **Misc Setup** > **Power Meter Setup** > **GPIB Address**

Setting USB Power Meter

Follow these steps to configure the USB power meter

Setting Description	Key Operation
USB power meter	**System** > **Misc Setup** > **Power Meter Setup** > **USB**

3. Setting Stimulus Condition

Follow these steps to configure the stimulus conditions:

Setting Description	Key Operation

Presetting	**Preset** > **OK**
Sweep type: Power	**Sweep Setup** > **Sweep Type** > **Power Sweep**
Fixed frequency: 1 GHz	**Sweep Setup** > **Power** > **CW Freq** > **1** > **G/n**
Start value: -40 dBm	**Start** > **+/-** > **4** > **0** > **x1**
Stop value: -10 dBm	**Stop** > **+/-** > **1** > **0** > **x1**
Number of points: 61	**Sweep Setup** > **Points** > **6** > **1** > **x1**

4. Executing Zero Adjustment and Calibration of Power Sensor

Execute the zero adjustment and calibration of the power sensor according to the *E4418B Power Meter User's Guide*.

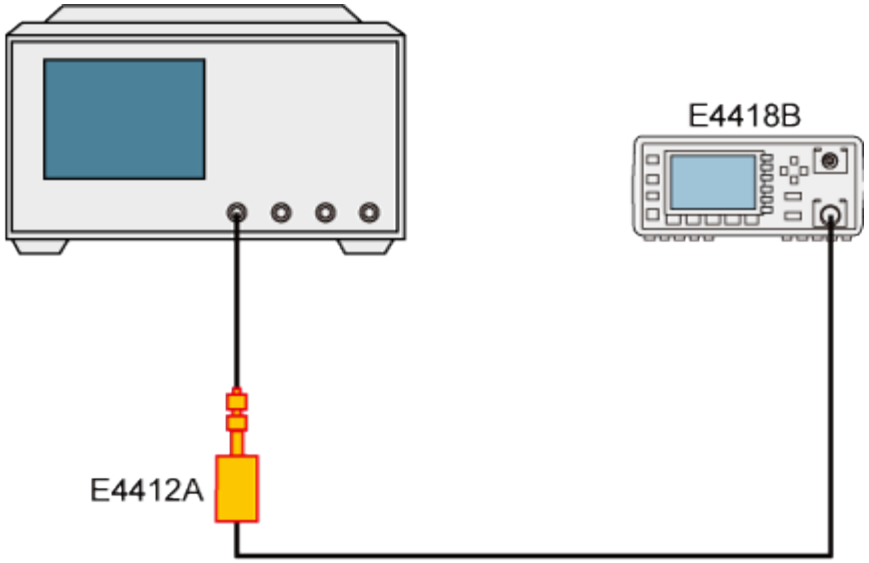
5. Setting Calibration Data Measurement Conditions

Follow these steps to set the calibration data measurement conditions:

Setting Description	Key Operation
Selecting the test port: 1	**Cal** > **Power Calibration** > **Select Port** > **1**
Selection of power sensor: A	**Cal** > **Power Calibration** > **Use Sensor [A]**
Number of measurements at one measurement point: 4	**Cal** > **Power Calibration** > **Num of Readings** > **4** > **x1**
Tolerance during power calibration: 5 dB	**Cal** > **Power Calibration** > **Tolerance** > **5** > **x1**

6. Connecting Power Sensor

Connect the power sensor as shown below.



e5071c335

7. Measuring Calibration Data

Follow this step to measure the calibration data:

Setting Description	Key Operation
Measuring calibration data	Cal > **Power Calibration** > **Take Cal Sweep**

Fixture Simulator

Fixture Simulator

- Overview of Fixture Simulator
- Extending the Calibration Plane using Network De-embedding
- Converting the Port Impedance of the Measurement Result
- Determining Characteristics After Adding a Matching Circuit
- Obtaining Characteristics After Embedding/De-embedding 4-port Network
- Evaluating Balanced Devices (balance-unbalance conversion function)
- Converting Reference Impedance of Balanced Port
- Determining the Characteristics that Result from Adding a Matching Circuit to a Differential Port
- Example of Using Fixture Simulator

Overview of Fixture Simulator

- Overview
- Functions for Single-Ended (Unbalanced) Port

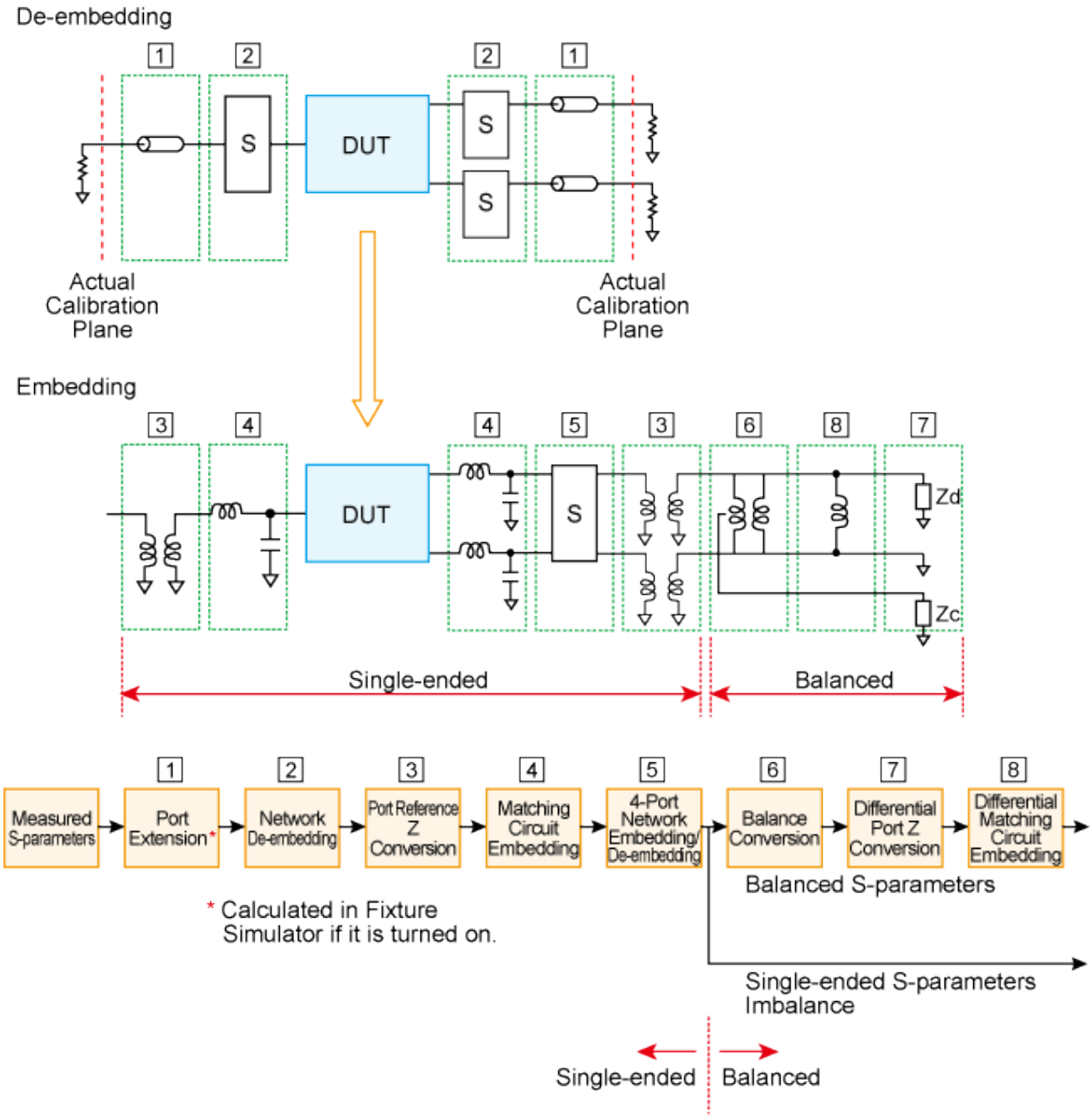
Other topics about Fixture Simulator

Overview

The Fixture Simulator is a function that uses software in the E5071C to simulate various measurement conditions based on the measurement results. The functions available in Fixture Simulator are as follows:

- Network de-embedding
- Port reference impedance conversion
- Matching circuit embedding
- 4-port network embedding/de-embedding
- Balance-unbalance conversion
- Differential/Common port reference impedance conversion
- Differential matching circuit embedding

Data processing flow diagram of fixture simulator



e5071c545

NOTE The E5071C adopts the power wave definition for its S-parameter calculation. Reference document: "Power waves and Scattering Matrix. K. Kurokawa"

Port extension is an independent function from the fixture simulator, but if the fixture simulator function is on, data processing is automatically executed as a function of the fixture simulator to improve the data processing efficiency. (Measurement result is the same as when the fixture simulator is turned off.) Port extension moves the calibration reference location by setting an electrical delay for a single-ended port. Port

extension can eliminate only electrical delay (phase shift) for each single-ended port. Loss or mismatch cannot be eliminated by this function.

Functions for Single-Ended (Unbalanced) Port

The following three functions are applied to single-ended ports (unbalanced ports). Balance-unbalance conversion can additionally be applied to single-ended ports.

Network de-embedding

A function that uses software to remove an arbitrary network (50 ohm system) defined by a two-port Touchstone data file from each test port (single-ended) and to extend the calibration plane. This makes it possible to remove networks that create error elements between the calibration plane and the DUT, thereby enabling a more realistic evaluation of the DUT.

For the setup procedure of the network de-embedding function, see Extending the Calibration Plane Using Network De-embedding.

Port reference impedance conversion

A function that uses software to convert an S-parameter measured with a 50 ohm port reference impedance into a value measured with an arbitrary impedance.

For the setup procedure of port reference impedance conversion, see Converting the Port Impedance of the Measurement Result.

Matching circuit embedding

A function for converting an original measurement result into a characteristic determined under the condition of inserting a matching circuit between the DUT and the test port (single-ended). The matching circuit to be inserted is either selected from the five predetermined circuit models or provided by a designated arbitrary circuit defined in a two-port Touchstone file.

For the setup procedure used for matching circuit embedding, see Determining Characteristics After Adding a Matching Circuit.

4-port network embedding/de-embedding

This is a feature to embed (in terms of numerical calculation) your desired network that you have defined in a 4-port Touchstone data file into measurement results or to de-embed it from them.

When you use this as de-embedding, the embedded matching circuit and a part of networks in the DUT are removed.

For information on how to operate this function, refer to Obtaining Characteristics After Embedding/De-embedding 4-port Network.

Balance-unbalance conversion (4 ports option only)

A function that uses software to convert the measurement results in an unbalanced DUT state, which are obtained by connecting the DUT to the test port of the E5071C, into measurement results in a balanced state. Two test ports of the E5071C are connected to one balanced port of the DUT.

For the setup procedure used for balance-unbalance conversion, see Evaluating Balanced Devices (balance-unbalance conversion function).

Functions for balanced port (4 ports option only)

The following two functions are applied to a balanced (differential) port converted by balance-unbalance conversion.

Differential/Common port impedance conversion (4 ports option only)

A function for converting the differential mode port impedance of a balanced port after an balance-unbalance conversion. Balance-unbalance conversion automatically converts the differential mode port impedance at the balanced port into 2Z0 and the common mode port impedance into Z0/2, compared with the two pre-conversion port impedances of Z0. Differential port impedance conversion further converts a differential port impedance after balance-unbalance conversion into an arbitrary port impedance.

For the setup procedure used for differential port impedance conversion, see Converting Reference Impedance of Balanced Port.

Differential matching circuit embedding (4 ports option only)

A function for converting the measurement results obtained from balance-unbalance conversion into a characteristic under the condition of inserting a matching circuit in the balanced port.

For setup procedure of differential matching circuit embedding, see Determining the Characteristics that Result from Adding a Matching Circuit to a Differential Port.

Extending the Calibration Plane Using Network De-embedding

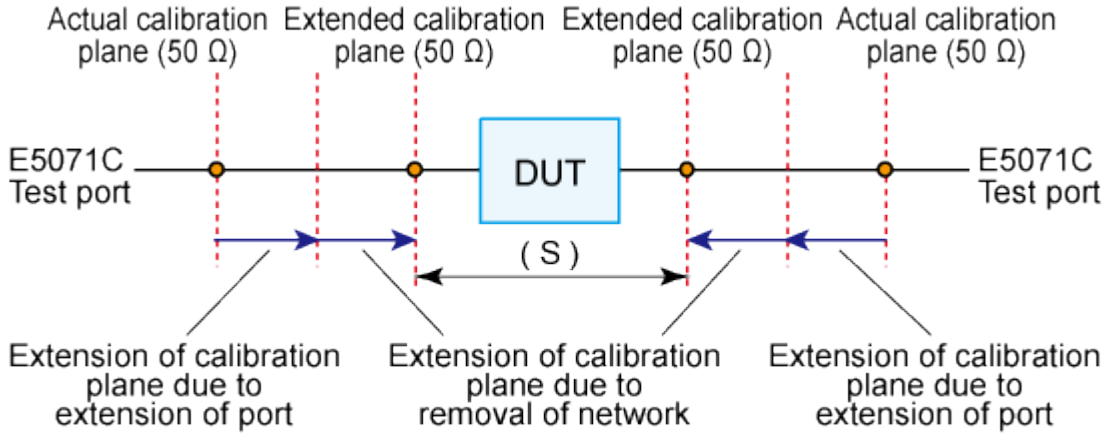
- Overview
- Using Network De-embedding Function

Other topics about Fixture Simulator

Overview

Network de-embedding is a function for performing measurements, test port by test port, by removing the characteristics of an arbitrary network defined by a Touchstone data file. By removing the characteristics of the cable, test fixtures, etc. between the actual calibration plane and the DUT, the calibration plane can be correspondingly extended. The network de-embedding function can be used together with the port extension function.

Port extension and calibration plane extension using network de-embedding



e5071c454

Using Network De-embedding Function

1. Prepare a two-port Touchstone data file (.s2p format) corresponding to the network to be removed.

   NOTE A Touchstone data file is defined for a single normalized impedance value (real). Thus, when preparing a Touchstone data file with the 2-port network de-embedding feature, be sure to specify the same port reference impedance value (real) to both 2-port network in the file.

2. Press **Analysis** key, then click **Fixture Simulator** > **De-Embedding**.
3. Click **Select Port**.

4. Click **1**, **2**, **3**, or **4** to select the test port from which the network de-embedding is performed.
5. Click **User File**.
6. Using the dialog box that appears, select the Touchstone data file defining the characteristics of the network to be removed. Once the file is selected, the selection of **Select Type** automatically changes to **User**. To cancel a user-defined file that has been set up, click **Select Type** > **None**.
7. Repeat the procedure to set up the Touchstone data file for each port from which a network is to be removed.
8. Click **De-Embedding** to turn the network de-embedding function **ON**.
9. Click **Return**.
10. If **Fixture Simulator** is **OFF**, press the key again to turn it **ON**.

Converting the Port Impedance of the Measurement Result

- Overview
- Converting Port Impedance

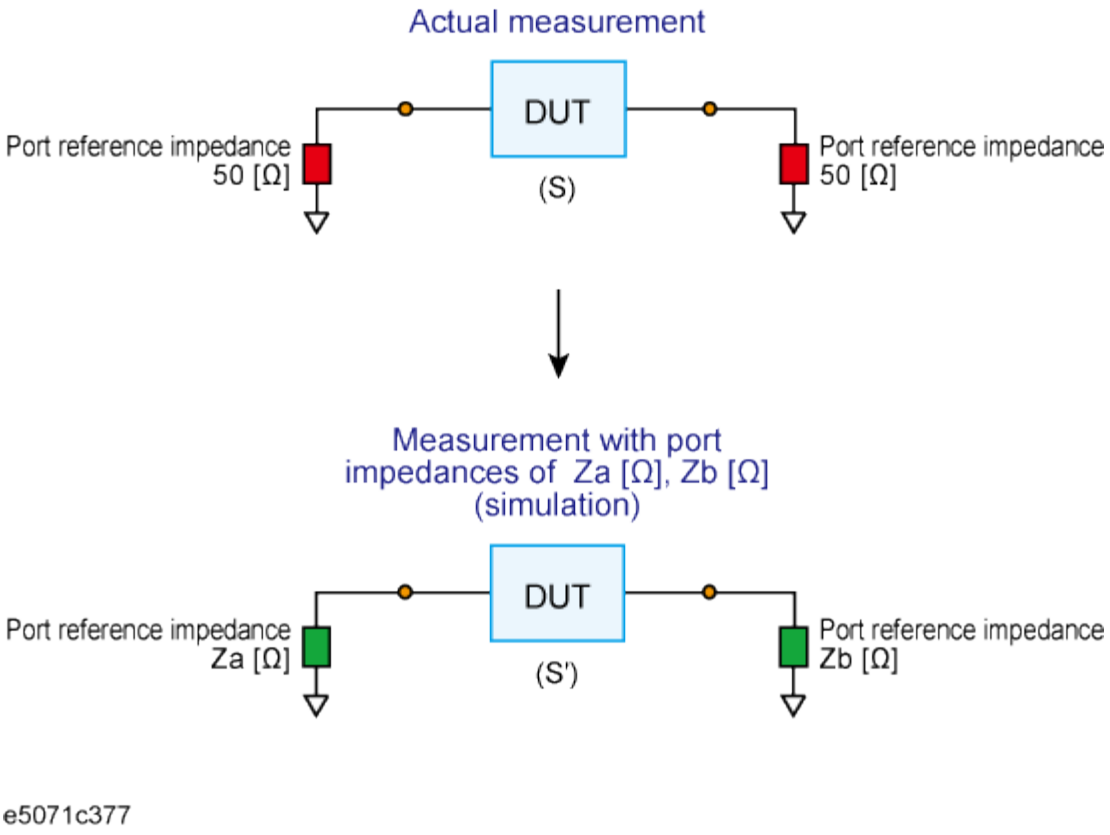
Other topics about Fixture Simulator

Overview

The measured value obtained by using a port impedance of 50 ohm can be converted into a measured value at an arbitrary port impedance.

NOTE When using balance conversion, be sure to set the impedances of the two unbalanced ports equal to each other.

Port impedance conversion function



e5071c377

Converting Port Impedance

1. Press **Analysis** key, the click **Fixture Simulator** > **Port Z Conversion**.
2. Specify the port reference impedance for the required port.
   - Select **Port 1 Z0 Real**, **Port 2 Z0 Real**, **Port 3 Z0 Real**, or **Port 4 Z0 Real** to set the port reference impedance in Real format.

- Select **Port 1 Z0 Real** and **Port 1 Z0 Imag**, **Port 2 Z0 Real** and **Port2 Z0 Imag**, **Port 3 Z0 Real** and **Port 3 Z0 Imag**, or **Port 4 Z0 Real** and **Port 4 Z0 Imag** to set the port reference impedance in Complex format.

3. Click **Port Z Conversion** to change the port impedance conversion function to the **ON** state.
4. Click **Return**.
5. If **Fixture Simulator** is **OFF**, click the key again to turn it **ON**.

Determining Characteristics After Adding a Matching Circuit

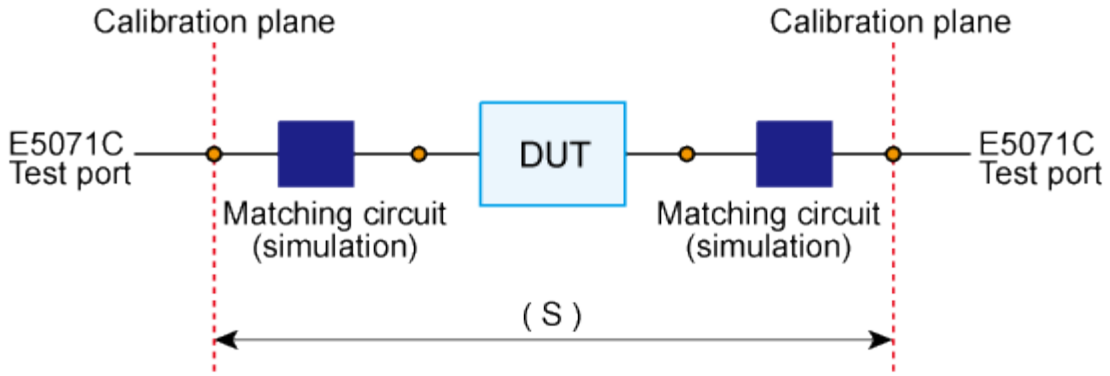
- Overview
- Using Matching Circuit Function

Other topics about Fixture Simulator

Overview

Using the matching circuit embedding function, you can easily obtain the resulting characteristics after adding a matching circuit for each test port.

Matching circuit function



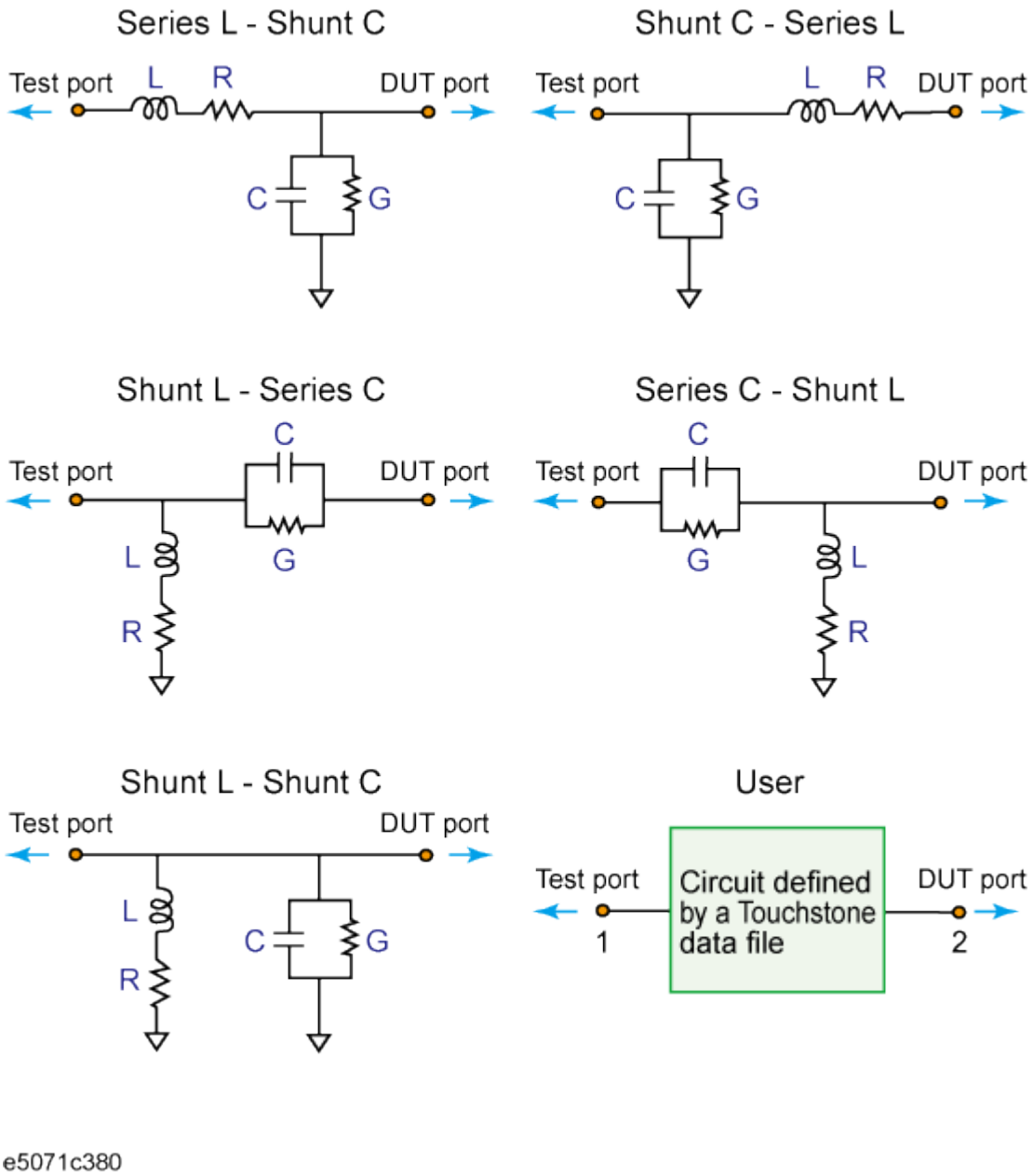
e5071c453

Define the matching circuit to be added by one of the following methods:

- Select one of the five predetermined circuit models and specify the values for the elements in the circuit model.
- Use a user file (in two-port Touchstone data format) that defines the matching circuit to be added.

The circuit models used for defining matching circuits are shown below.

Circuit models for defining matching circuits



e5071c380

- In the 2-port matching circuit embedding or the 2-port network de-embedding feature, if the normalized impedance value specified in the user file (2-port Touchstone data file) is different from the port reference impedance setting value of the analyzer, it is automatically converted to adapt to the analyzer setting.

  When a 2-port Touchstone data file is read in, data for up to 3202 frequency points are read in using interpolation according to the measurement frequency point setting of the analyzer. If

the number of frequency points in the file is greater than 3202, excess data are ignored.

For a network defined in the user file, it is assumed that port 1 is connected to the test port and port 2 is connected to the DUT.

Using Matching Circuit Function

1. Press **Analysis** key.
2. Click **Fixture Simulator** > **Port Matching** > **Select Port**.
3. Click **1**, **2**, **3**, or **4** to select the port to which a matching circuit is to be added.
4. Click **Select Circuit**.
5. Select a matching circuit model.

Softkey	Function
None	No matching circuit is added.
Series L - Shunt C	Selects a circuit model consisting of a series inductor and a shunt capacitor
Shunt C - Series L	Selects a circuit model consisting of a shunt capacitor and a series inductor
Shunt L - Series C	Selects a circuit model consisting of a shunt inductor and a series capacitor
Series C - Shunt L	Selects a circuit model consisting of a series capacitor and a shunt inductor
Shunt L - Shunt C	Selects a circuit model consisting of a shunt inductor and a shunt capacitor
User	Selects the circuit model defined in the user file imported. To add a matching circuit defined in a user file, execute the following operations before selecting this softkey: 1. Press **User File**. 2. In the dialog box that appears, select the two-port Touchstone data file (.s2p format) for the matching circuit to be added.

	NOTE A Touchstone data file is defined for a single normalized impedance value (real). Thus, when preparing a Touchstone data file for the matching circuit feature, be sure to specify the same port reference impedance value (real) to both 2-port network in the file. Once a user file is specified, the selection of **Select Circuit** automatically changes to **User**.

6. For the circuit models, see Circuit models for defining matching circuits.
7. Specify the values of the elements in the selected circuit model.

Softkey	Function
C	Specifies capacitance [F]
G	Specifies conductance [S]
L	Specifies inductance [H]
R	Specifies resistance [ohm]

1. When either of the two components of Series-C, "C" and "G", is set to 0, the component is defined as "Open"; when both components are set to 0, they are defined as "Short."
   When either of the two components of Shunt-L, "L" and "R", is set to 0, the component is defined as "Short"; when both components are set to 0, they are defined as "Open."

8. Repeat the procedure above to set up the matching circuit for each port used.
9. Click **Port Matching** to turn the matching circuit function **ON**.
10. Click **Return**.
11. If **Fixture Simulator** is **OFF**, press the key again to turn it **ON**.

Obtaining Characteristics After Embedding/De-embedding 4/6/8-port Network

- Overview
- Procedure

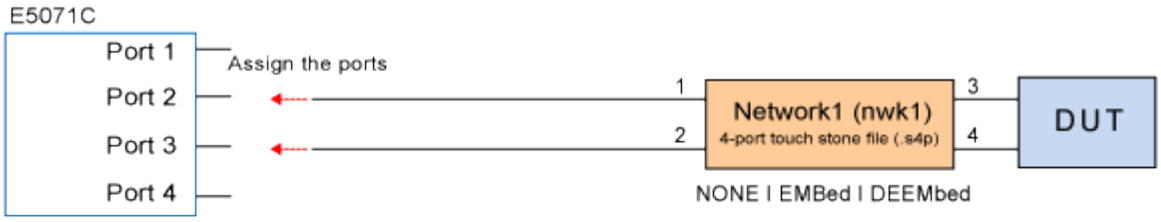
Other topics about Fixture Simulator

Overview

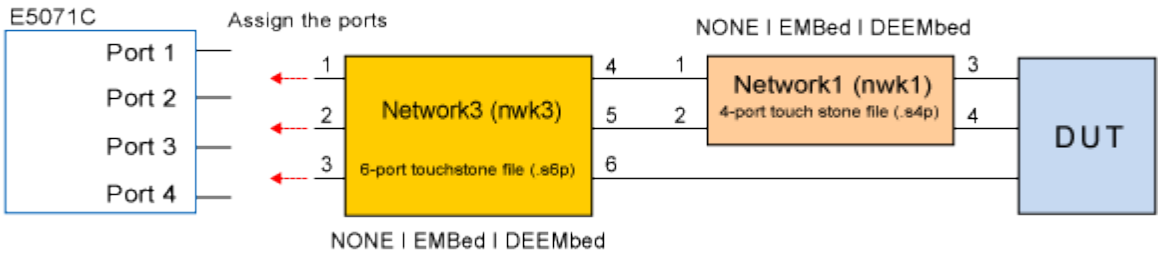
The 4/6/8-port network embedding/de-embedding feature allows you to add (embed) or remove (de-embed) data of your desired 4/6/8-port network defined in a 4/6/8-port Touchstone data file to or from measurement values through software-based processing. The embedding/de-embedding supports three types of connection as shown below.

Connection that enables embedding/de-embedding network (4/6/8-port touchstone data)

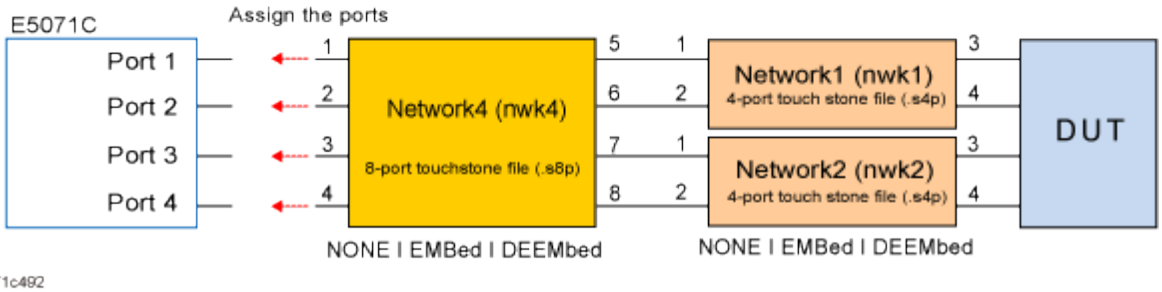
When DUT is connected to 2 ports on analyzer (Topology A)



When DUT is connected to 3 ports on analyzer (Topology B)



When DUT is connected to 4 ports on analyzer (Topology C)



Touchstone Data File Definition

When defined in the 4-port Touchstone data file, a 4-port network assumes that ports 1 and 2 are connected to the analyzer and that ports 3 and 4 are connected to the DUT.

When defined in the 6-port Touchstone data file, a 6-port network assumes that ports 1, 2 and 3 are connected to the analyzer and that ports 4, 5 and 6 are connected to the DUT.

When defined in the 8-port Touchstone data file, a 8-port network assumes that ports 1 to 4 are connected to the analyzer and that ports 5 to 8 are connected to the DUT.

Topology Selection

For measuring a 4-port DUT with a 4-port network embed/de-embed only on one side, select topology C instead of A, set embedding/de-embedding on the desired side, and specify no processing (None) on the other side.

For measuring a 3-port DUT with a 4-port network embed/de-embed on one side and a 2-port network on the other side, select topology B instead of A, embed/de-embed a 4-port network on one side, and embed/de-embed a 2-port network on the other side using the 2-port matching circuit embedding feature or the 2-port network de-embedding feature. Selecting topology A will result in unexpected measurement.

Procedure

1. Prepare a 4/6/8-port touchstone data file (.s4p/.s6p/.s8p format) corresponding to a network you want to embed/de-embed.
2. Press **Analysis** key.
3. Click **Fixture Simulator** > **De-Embedding SnP** > **Topology** > **Select Topology**.
4. Select a network connection type (topology) from A, B, or C.
5. Click **Ports** to select measurement port numbers of the analyzer to which you want to connect ports 1 and 2 of the 4-port network.
   - For topology **A**: Select measurement port numbers of the analyzer to which you want to connect nwk1-1 and nwk1-2. For example, when you connect nwk1-1 and nwk1-2 to measurement ports 1 and 2, respectively, specify Ports ="1-2".
   - For topology **B**: Select measurement port numbers of the analyzer to which you want to connect nwk3-1, nwk3-2, and nwk3-3. For example, when you connect nwk3-1, nwk3-2, and nwk3-3 to measurement ports 1, 2, and 3, respectively, specify Ports="1-2-3".
   - For topology **C**: Select measurement port numbers of the analyzer to which you want to connect nwk4-1, nwk4-2, nwk4-

3, and nwk4-4. For example, when you connect nwk4-1, nwk4-2, nwk4-3, and nwk4-4 to measurement ports 1, 2, 3, and 4, respectively, specify Ports="1-2-3-4".

6. Click **User File (nwk*n*)** and select a Touchstone data file you want to use for network n. For nwk1 and nwk2, a 4-port touchstone data file should be selected. a 6-port and 8-port touchstone data file should be selected for nwk3 and nwk4, respectively.
7. Click **Type (nwk*n*)** and select a **processing type** for network n..

Softkey	Function
None	Specifies no-processing (equivalent to nothing on the network.)
Embed	Selects embedding
De-Embed	Selects de-embedding

8. If you have not specified an appropriate Touchstone data file, you cannot select embedding (**Embed**) or de-embedding (**De-embed**). If you have not specified an appropriate file and you try to select embedding/de-embedding, an error occurs and no-processing (**None**) is automatically selected.
9.
10. Click **De-Embedding SnP** to turn **ON** the 4/6/8-port network embedding/de-embedding feature.
11. Click **Return**.
12. If **Fixture Simulator** is **OFF**, click the key to turn it **ON**.

At the firmware A.9.61 and above, when the 4/6/8-port network embedding/de-embedding feature reads a 4/6/8-port Touchstone data file,it automatically convert the file's normalized impedance value to adapt to the port reference impedance setting value of the analyzer.

At the firmware A.9.61 and above, the data in the 4/6/8-port Touchstone data file has to be defined with a single normalized impedance value (real) written in the option line.

When a 4/6/8-port Touchstone data file is read in, data for up to 3202 frequency points are read in by using interpolation according to the measurement frequency point setting of the analyzer. If the number of frequency points in the file is greater than 3202, excess data are ignored.

Evaluating Balanced Devices (Balance-Unbalance Conversion Function)

- Overview
- Measurement Parameters of Balanced Devices
- Procedure for Balance-Unbalance Conversion
- Measurement Parameter Setup
- Checking Device Type and Port Assignment

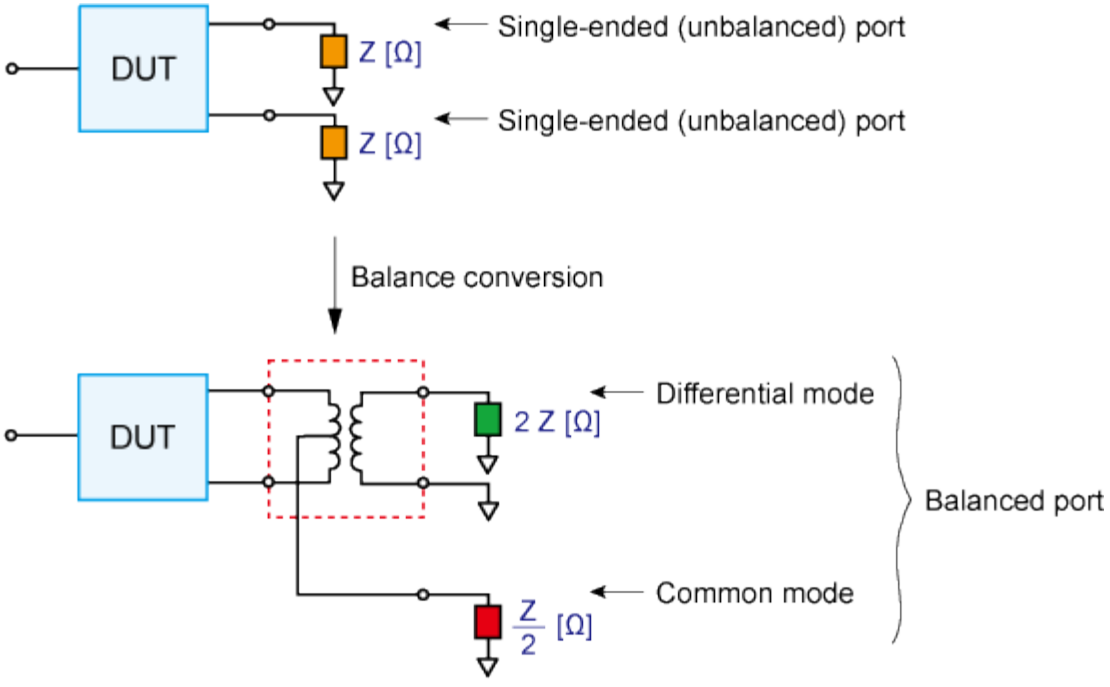
Other topics about Fixture Simulator

Overview

The balance-unbalance conversion function simulates a measurement under a balanced state based on measurement results obtained in an unbalanced state. This function enables you to evaluate devices with balanced ports.

- Be sure to set the impedances of the two unbalanced ports equal to each other. For more details on setting up port impedance for unbalanced ports, refer to Converting the Port Impedance of the Measurement Result.

Balance-unbalance conversion

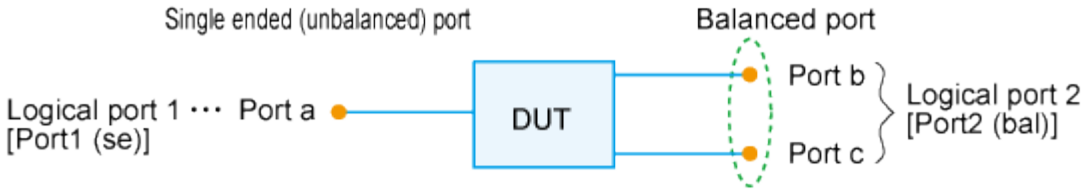


e5071c406

The types of devices that can be evaluated using the E5071C are shown below. To evaluate a balanced device, an E5071C with at least three test ports is required.

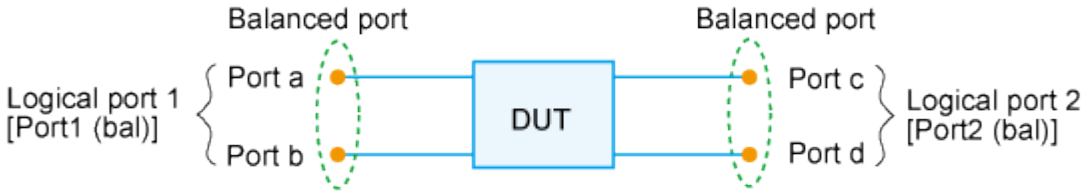
Types of balanced devices that can be evaluated with E5071C

Single ended—Balanced Devices (SE-Bal)



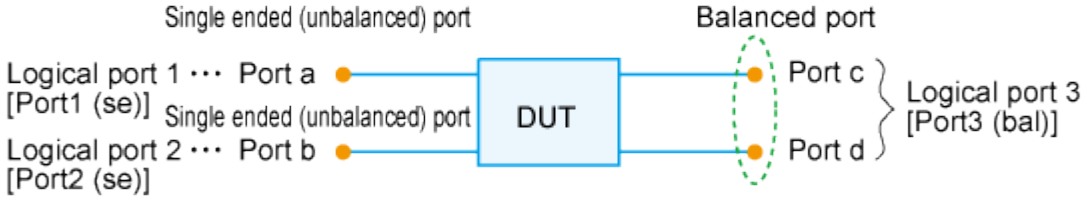
Number of test ports required for analyzer : 3

Balanced—Balanced Devices (Bal-Bal)



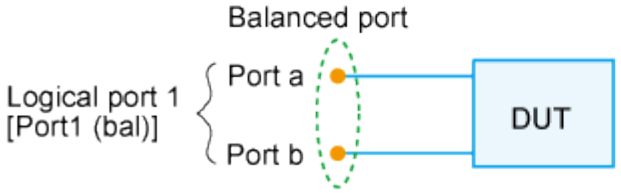
Number of test ports required for analyzer : 4

Single ended—Single ended—Balanced Devices (SE-SE-Bal)



Number of test ports required for analyzer : 4

Balanced Devices (Bal)



Number of test ports required for analyzer : 2

Enclosed with [ ] are softkey labels.

e5071c375

In the terminology of the E5071C, ports after the balance conversion are called logical ports (or DUT ports).

You can freely assign the test ports of the E5071C to logical ports (ports a to d in the figure above).

Measurement Parameters of Balanced Devices

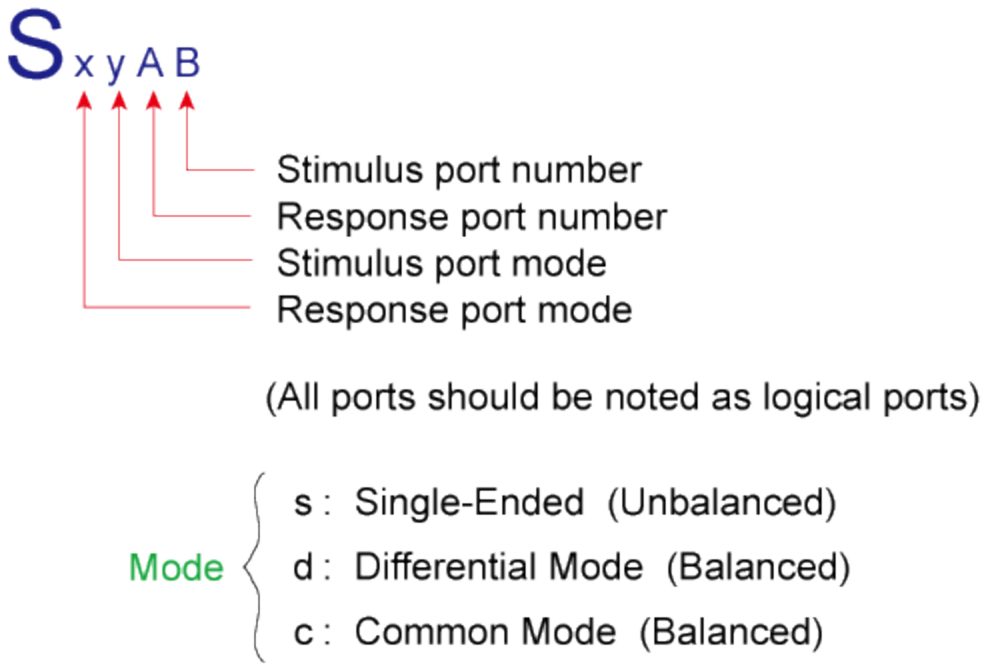
Turn on the balance-unbalance conversion function to measure the following parameters.

- Mixed mode S-parameter
- Imbalance parameter
- CMRR (Common Mode Rejection Ratio)

Mixed mode S-parameter

By turning on the balance-unbalance conversion function, you can obtain the S-parameter of the balanced port separately for two modes, the differential mode and the common mode. The following figure shows the notation of the S-parameter in balance measurement (mixed mode S-parameter).

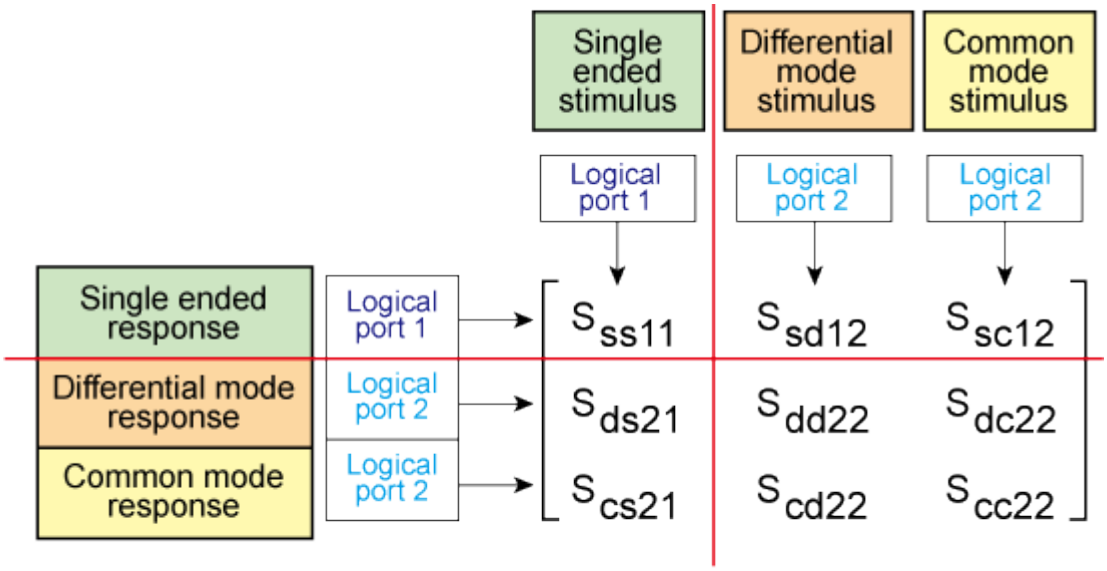
Notation of mixed mode S-parameter



e5071c381

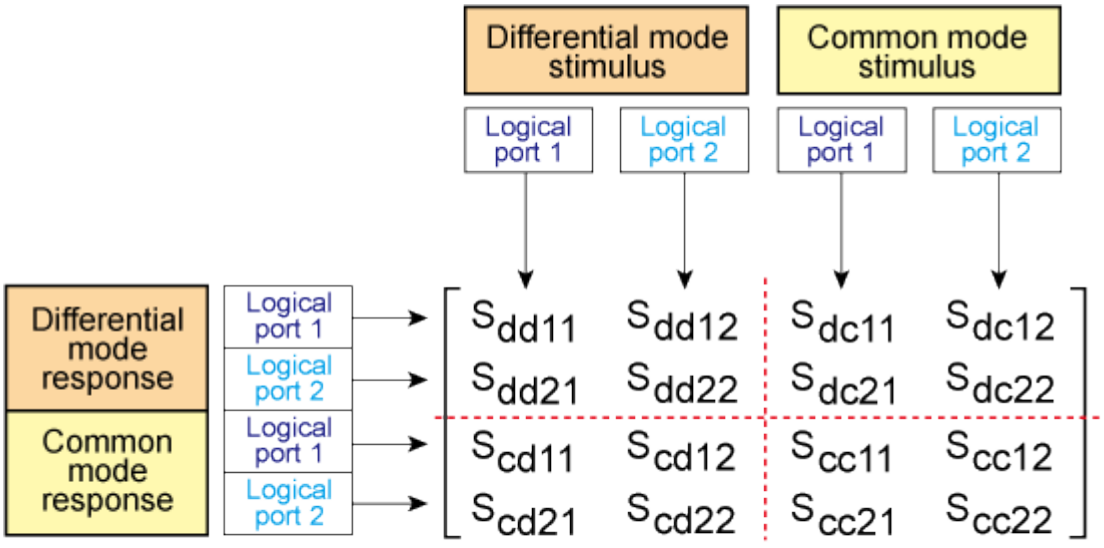
The following two figures show the mixed mode S-parameter when measuring each balanced device.

Mixed mode S-parameter when measuring a single-ended - balanced device



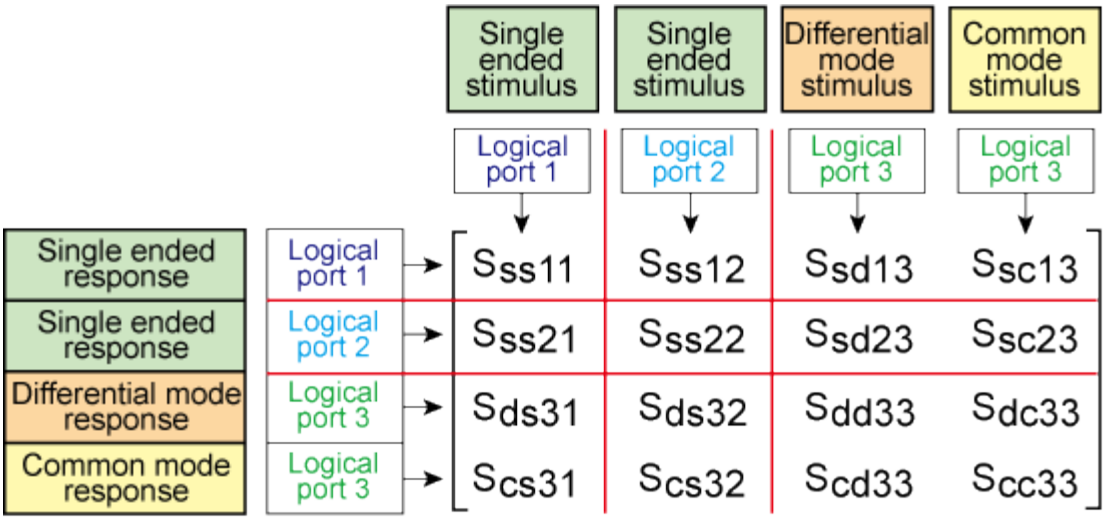
e5071c408

Mixed mode S-parameter when measuring a balanced - balanced device



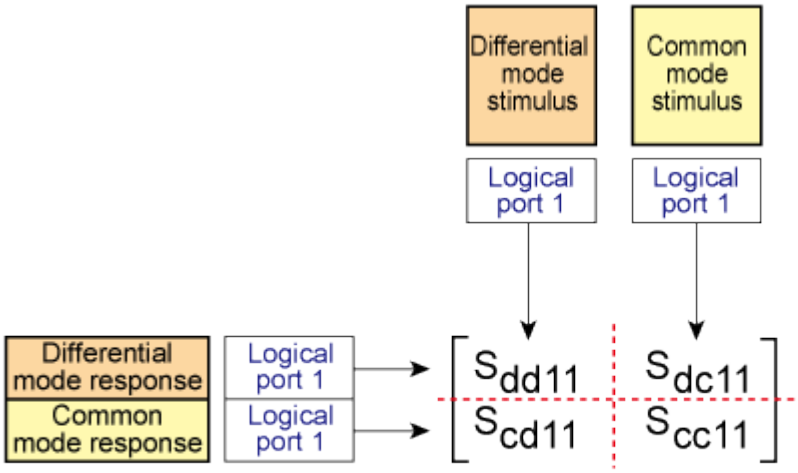
e5071c407

Mixed mode S-parameter when measuring a single-ended - single-ended - balanced device



e5071c409

Mixed mode S-parameter when measuring a balanced device

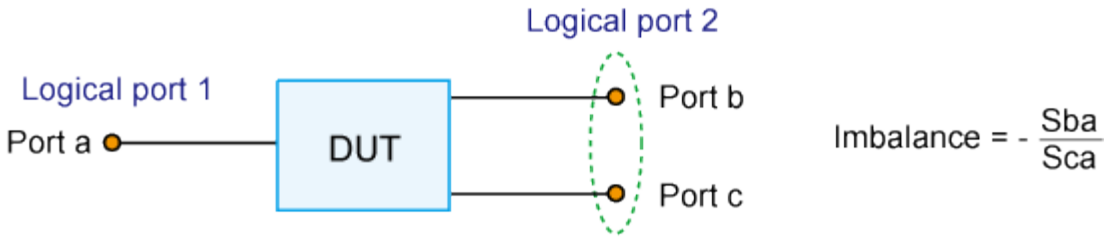


e5071c540

Imbalance Parameter

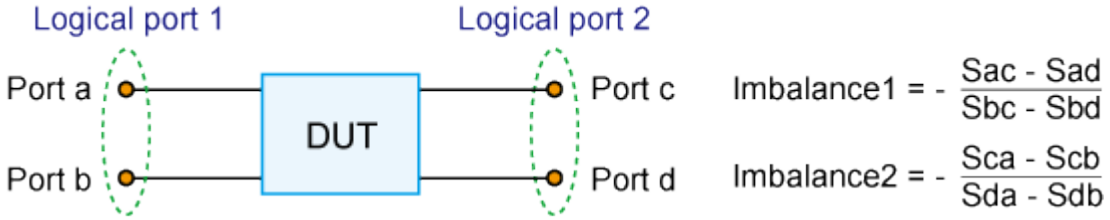
By turning on the balance-unbalance conversion function, you can select the imbalance parameter of the balanced port as the measurement parameter. The following three figures show the imbalance parameter you can select when measuring each balanced device.

Parameter when measuring a single-ended - balanced device (Imbalance)



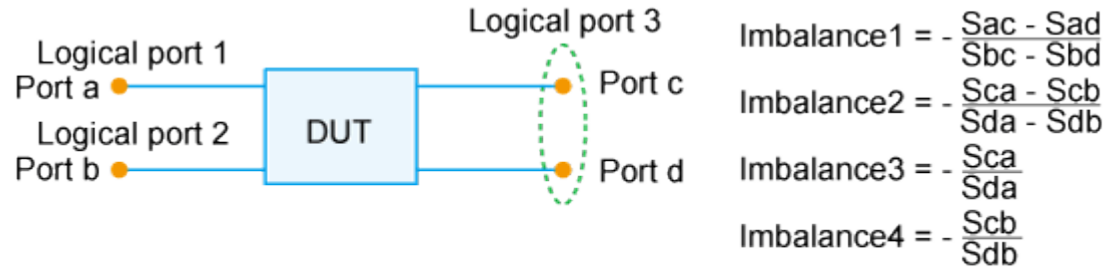
e5071c410

Parameter when measuring a balanced - balanced device (Imbalance1, Imbalance2)



e5071c411

Parameter when measuring a single-ended - single-ended - balanced device (Imbalance1, Imbalance2, Imbalance3, Imbalance4)



e5071c456

- The balance matching circuit (Differential matching circuit embedding) is not applied to single-ended S-parameter results. For example, the balance matching circuit is not applied to the

imbalance parameter as it is derived from single-ended S-parameters.

CMRR (Common Mode Rejection Ratio)

By turning on the balance-unbalance conversion function, you can select CMRR (ratio between the transmission characteristic in the differential mode and that in the common mode) of the balanced port as the measurement parameter. The table below shows the CMRR parameter you can select when measuring each balanced device.

Device Type	CMRR
Single-ended - balanced device	S_{ds21}/S_{cs21} and S_{sd12}/S_{sc12}
Balanced - balanced device	S_{dd21}/S_{cc21}
Single-ended - single-ended - balanced device	S_{ds31}/S_{cs31} and S_{ds32}/S_{cs32}

Procedure for Balance-Unbalance Conversion

- When using three test ports, perform a full three-port calibration on the test ports to be used. When using four test ports, perform a full four-port calibration.

1. Extend the calibration plane, if necessary, by using the port extension function or network de-embedding function. For more on the port extension and network de-embedding functions, see Extending the Calibration Plane Using Network De-embedding.

1. Press **Analysis** key.
2. Click **Fixture Simulator** > **Topology** > **Device**.
3. Select the balanced/unbalanced topology.

Softkey	Function
SE-Bal	Establishes port 1 on the DUT as an unbalanced port and port 2 as a balanced port
Bal-Bal	Establishes both port 1 and port 2 on the DUT as balanced ports
SE-SE-Bal	Establishes port 1 and port 2 on the DUT as unbalanced ports and port 3 as a balanced port

Bal	Establishes port 1 on the DUT as balanced ports

4. Select each port on the analyzer to which a port on the DUT is connected.

- When you have selected **SE-Bal**:

Softkey	Function
Port 1 (se)	Selects a port on the analyzer from among **1**, **2**, **3**, and **4** for connection to logical 1 (Port a)
Port 2 (bal)	Selects two ports on the analyzer from among **1-2**, **1-3**, **1-4**, **2-1**, **2-3**, **2-4**, **3-1**, **3-2**, **3-4**, **4-1**, **4-2**, and **4-3** for connection to logical port 2 (Port b and Port c)

- When you have selected **Bal-Bal**:

Softkey	Function
Port 1 (bal)	Selects two ports on the analyzer from among **1-2**, **1-3**, **1-4**, **2-1**, **2-3**, **2-4**, **3-1**, **3-2**, **3-4**, **4-1**, **4-2**, and **4-3** for connection to logical port 1 (Port a and Port b)
Port 2 (bal)	Selects two ports on the analyzer from among **1-2**, **1-3**, **1-4**, **2-1**, **2-3**, **2-4**, **3-1**, **3-2**, **3-4**, **4-1**, **4-2**, and **4-3** for connection to logical port 2 (Port c and Port d)

- When you have selected **SE-SE-Bal**:

Softkey	Function
Port 1 (se)	Selects a port on the analyzer from among **1**, **2**, **3**, and **4** for connection to logical 1 (Port a)
Port 2 (se)	Selects a port on the analyzer from among **1**, **2**, **3**, and **4** for connection to logical 2 (Port b)
Port 3 (bal)	Selects two ports on the analyzer from among **1-2**, **1-3**, **1-4**, **2-1**, **2-3**, **2-4**, **3-1**, **3-2**, **3-4**, **4-1**, **4-2**, and **4-3** for connection to logical port 3 (Port c and Port d)

- When you have selected **Bal**:

Softkey	Function

Port 1 (bal)	Selects two ports on the analyzer from among **1-2**, **1-3**, **1-4**, **2-1**, **2-3**, **2-4**, **3-1**, **3-2**, **3-4**, **4-1**, **4-2**, and **4-3** for connection to logical port 1 (Port a)

5. Click **Return**.
6. Click **BalUn** to turn on the balanced/unbalanced state conversion function.
7. Click **Fixture Simulator** to turn on the fixture simulator function.

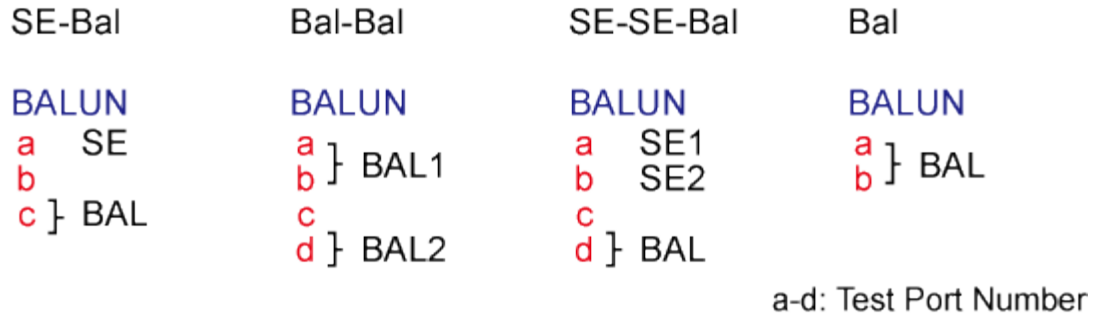
Measurement Parameter Setup

Performing balance-unbalance conversion enables you to make measurements with mixed mode S-parameters, imbalance parameters and CMRR. Parameters that can be used differ depending on the balance-unbalance topology specified in Balance-Unbalance Conversion.

1. Press **Meas** key (or **Analysis** > **Fixture Simulator** > **Measurement**).
2. Select the measurement parameter.

Checking Device Type and Port Assignment

You can check the device type and the port assignment for the balance-unbalance conversion by displaying the balance measurement topology property as shown below.



e5071c445

Procedure to Turn On/Off Balance Measurement Topology Property Display

Follow these steps to turn on/off the balance measurement topology property display.

1. Press **Analysis** key.
2. Click **Fixture Simulator** > **Topology**
3. Click **Property**. Each press toggles between on/off.

Converting Reference Impedance of Balanced Port

- Overview
- Converting Port Reference Impedance in Differential Mode
- Converting Port Reference Impedance in Common Mode

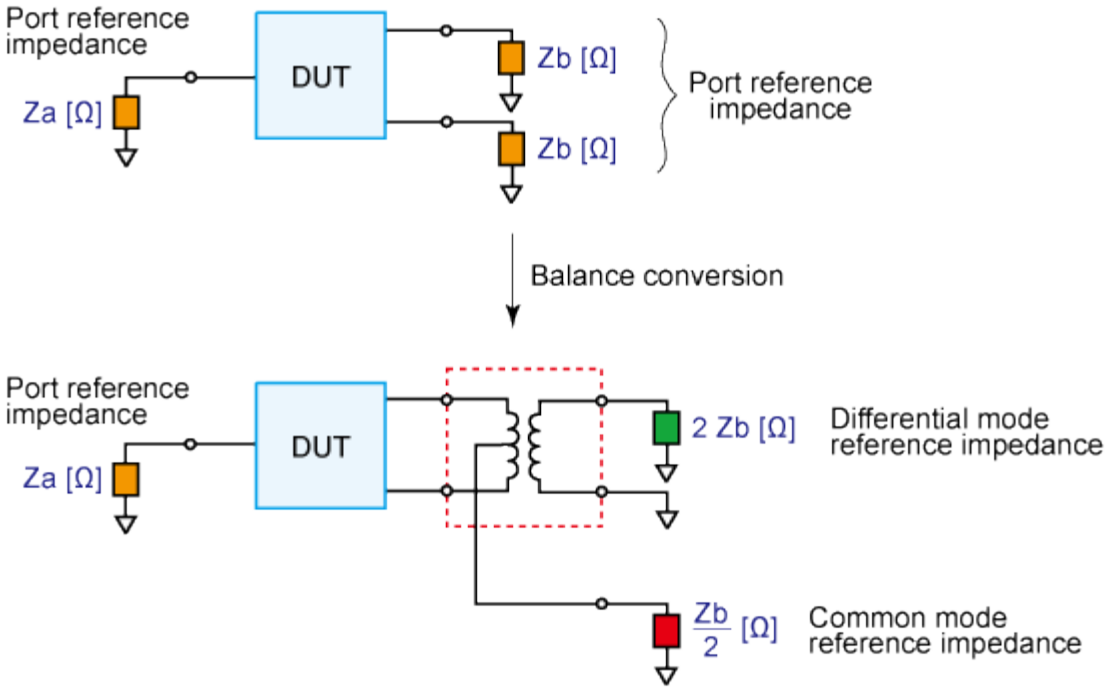
Other topics about Fixture Simulator

Overview

By using the port impedance conversion function, you can specify the impedance of each test port. As a result of this conversion, the impedance of the balanced port in differential mode is set to a value twice as large as the impedance of the two unbalanced ports before conversion, and in common mode to a value one-half as large.

NOTE Be sure to set the impedances of the two unbalanced ports equal to each other. For more details on setting up port impedance for unbalanced ports, refer to Converting the Port Impedance of the Measurement Result.

Port impedance after a balance-unbalance conversion



e5071c403

As described above, the impedance of the balanced port is automatically specified as the result of specifying the impedance of the two unbalanced ports prior to balance-unbalance conversion. However, the port impedance

can be changed to an arbitrary value by using the differential port impedance conversion function and the common port impedance conversion function.

Converting Port Reference Impedance in Differential Mode

If you turn on the differential port impedance conversion function, the port reference impedance in the differential mode is converted to an arbitrary value specified with this function instead of the value in the above figure.

Procedure to Turn On/Off Differential Port Reference Impedance Conversion Function

1. Press **Analysis** key, then click **Fixture Simulator** > **Diff Z Conversion**.
2. Click **Diff Z Conversion** to set the differential impedance conversion function to **ON**.

You can only turn on or off Differential Port Impedance Conversion for all of the balanced ports together, not for each port individually. If you want to turn off only a specific port, set the reference impedance of the port to the value in the above figure.

Procedure to Set Differential Port Reference Impedance

1. Press **Analysis** key, then click **Fixture Simulator** > **Diff Z Conversion**.
2. Specify the port reference impedance in differential mode.
3. Select the balanced port from **Port 1 (bal) Real**, **Port 2 (bal) Real**, or **Port 3 (bal) Real** to set the differential port reference impedance in "Real" format.
4. Select **Port 1 (bal) Real** and **Port 1 (bal) Imag**, **Port 2 (bal) Real** and **Port2 (bal) Imag**, or **Port 3 (bal) Real** and **Port 3 (bal) Imag** to set the differential port reference impedance in "Complex" format.
    5. Ports 1, 2, and 3 refer to the logical ports 1, 2, and 3.

Converting Port Reference Impedance in Common Mode

If you turn on the common port impedance conversion function, the port reference impedance in the common mode is converted to an arbitrary value specified with this function instead of the value the above figure.

Procedure to Turn On/Off Common Port Reference Impedance Conversion

1. Press **Analysis** key, then click **Fixture Simulator** > **Cmn Z Conversion**.
2. Click **Cmn Z Conversion** to set the common port impedance conversion function to **ON**.
3. You can only turn on or off Common Port Impedance Conversion for all of the ports together, not for each port individually. If you want to turn off only a specific port, set the impedance of the port to the value in the above figurerocedure to set common port reference impedance

1. Press **Analysis** key, then click **Fixture Simulator** > **Cmn Z Conversion**.

2. Specify the port reference impedance in common mode.
3. Select the balanced port from **Port 1 (bal) Real**, **Port 2 (bal) Real**, or **Port 3 (bal) Real** to set the common port reference impedance in "Real" format.
4. Select **Port 1 (bal) Real** and **Port 1 (bal) Imag**, **Port 2 (bal) Real** and **Port2 (bal) Imag**, or **Port 3 (bal) Real** and **Port 3 (bal) Imag** to set the common port reference impedance in "Complex" format.
5. Ports 1, 2, and 3 refer to the logical ports 1, 2, and 3.

Determining the Characteristics that Result from Adding a Matching Circuit to a Differential Port

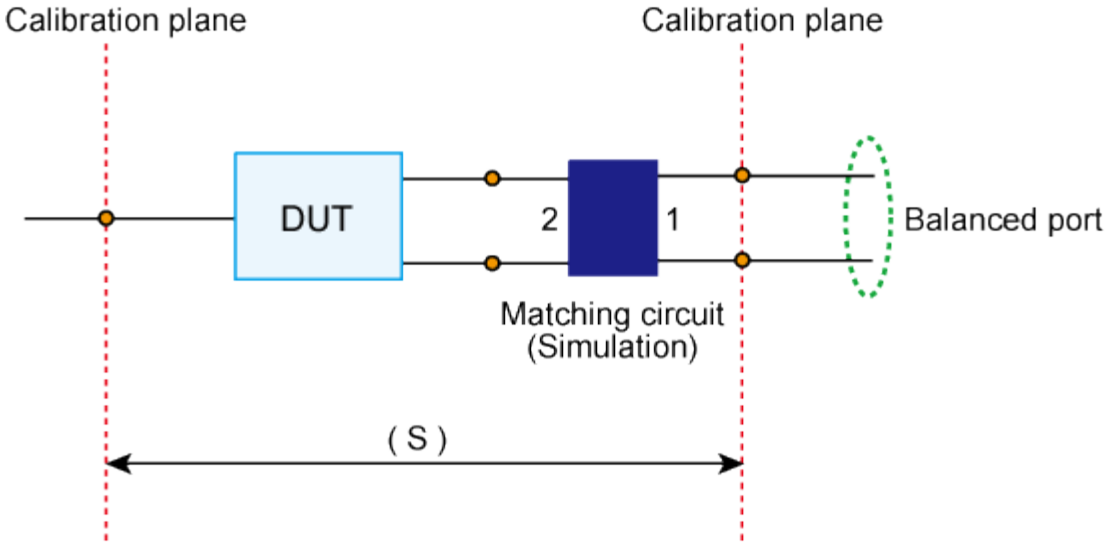
- Overview
- Procedure

Other topics about Fixture Simulator

Overview

You can obtain the characteristics resulting from the pseudo addition of a balance matching circuit to a balanced port created by balance-unbalance conversion. By using the matching circuit function, you can obtain the characteristics resulting from the addition of an arbitrary matching circuit for each test port.

Balance matching circuit function



e5071c376

NOTE The balance matching circuit (Differential matching circuit embedding) is not applied to single-ended S-parameter results. For example, the balance matching circuit is not applied to the imbalance parameter as it is derived from single-ended S-parameters.

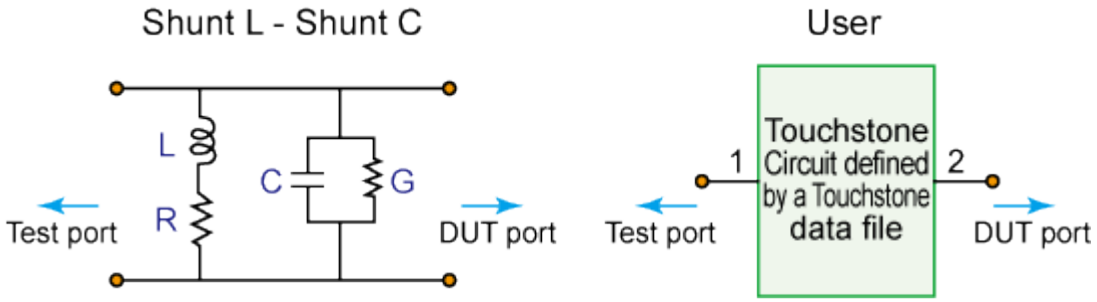
Procedure

Define the matching circuit to be added by one of the following methods:

- Use a predetermined circuit model and specify the values for the elements in the circuit model.
- Use a user file (in two-port Touchstone format) to define the matching circuit to be added.

The following figure shows the circuit models used in defining a balance matching circuit.

Circuit models used to define balance matching circuit



e5071c379

- For a network defined in a user file, it is assumed that port 1 is connected to the test port and port 2 is connected to the DUT.

The setup steps are shown below.

1. Press **Analysis** key, then click **Fixture Simulator** > **Diff. Matching**.
2. Click **Select Port**.
3. Click **1**, **2**, or **3** to select the port on the DUT to which a differential matching circuit will be added.
4. To add a matching circuit defined in a user file, perform the following operations:
   a. Press **User File**.
   b. Using the dialog box that appears, select the 2-port Touchstone data file (.s2p format) for the matching circuit to be added.
   c. Once you have specified the user file, the selection of **Select Circuit** automatically changes to **User**. In this case, you do not have to execute Step 5 and Step 6.
5. Click **Select Circuit**.
6. Select a differential matching circuit model.

Softkey	Function
None	The matching circuit is not added.
Shunt L - Shunt C	Selects a circuit model consisting of a shunt inductor and a shunt capacitor
User	Selects the circuit model defined in the user file imported in Step4.

7. Specify the values for the elements in the circuit model selected.

Softkey	Function
C	Specifies the capacitance [F]
G	Specifies the conductance [S]
L	Specifies the inductance [H]
R	Specifies the resistance [Ω]

6.

- When either of the two components of Shunt-L, "L" and "R", is set to 0, the component is defined as "Short"; when both components are set to 0, they are defined as "Open."

7. Repeat the procedure to set up the differential matching circuit to be added to the selected ports on the DUT.
8. Click **Diff. Matching** to turn the differential matching circuit **ON**.
9. Click **Return**.
10. If **Fixture Simulator** is **OFF**, press the key again to turn it **ON**.

Example of Using Fixture Simulator

- Overview
- Measurement Circuit Example: DUT with Balanced Port
- Evaluation Using Actual Test Fixture
- Problems in Measurement with Actual Test Fixture
- DUT Evaluation using E5071C's Fixture Simulator
- Advantages of Balanced DUT Evaluation using Fixture Simulator

Other topics about Fixture Simulator

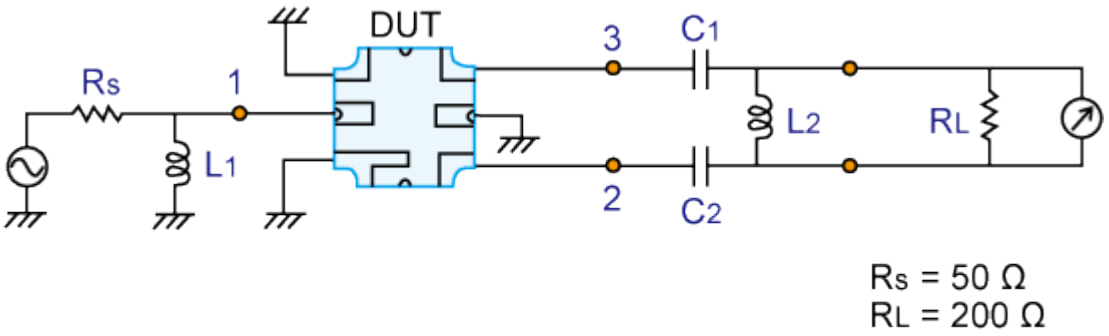
Overview

In this section, the fixture simulation function is explained based on an evaluation example for a DUT (balanced SAW filter) with a balanced port.

Measurement Circuit Example: DUT with Balanced Port

The following figure shows an example of a measurement circuit used to evaluate a balanced SAW filter. DUT port 1 is an unbalanced port connected to source impedance R_S and input matching circuit L_1. DUT port 2 is a balanced port connected to an output matching circuit (C_1, C_2, and L_2) and load resistance R_L.

Measurement circuit of balance SAW filter



e5071c420

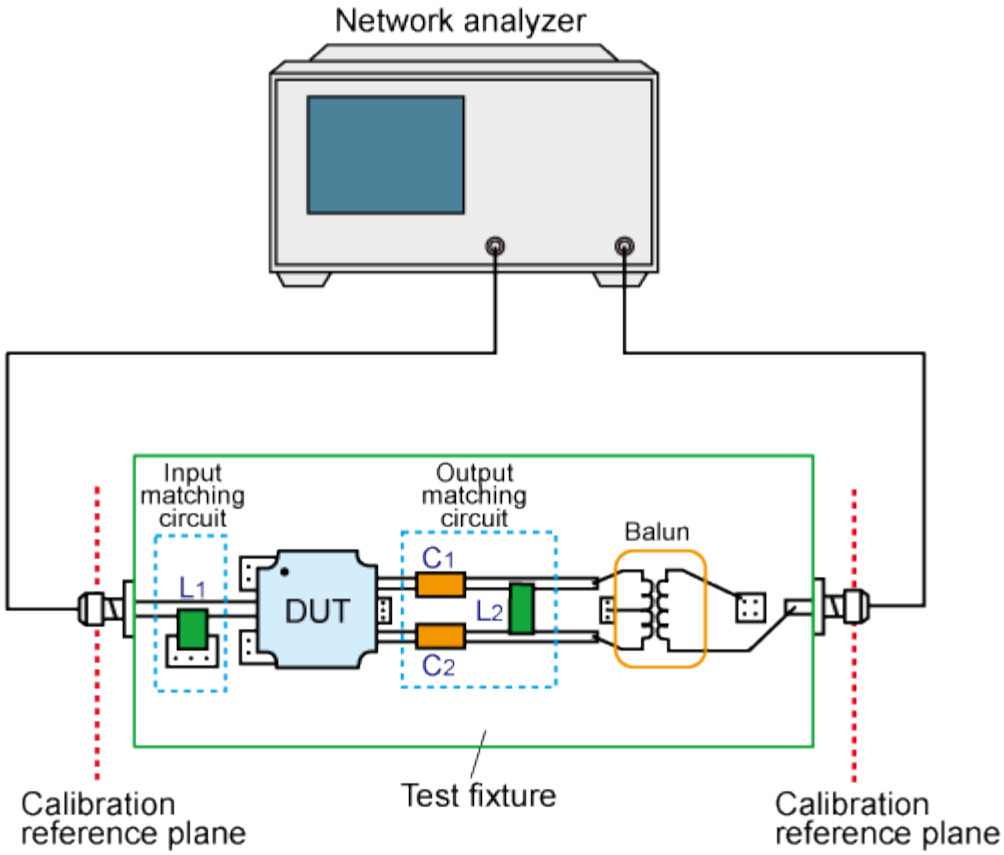
Evaluation Using Actual Test Fixture

Generally, a test fixture as shown in the following figure is fabricated for evaluating the characteristics of a DUT in a measurement circuit by using a network analyzer.

In typical use, a network analyzer performs measurements at a 50 ohm port reference impedance and in a single-ended (unbalanced) state. Therefore, DUT port 1 can be connected directly to the test port of the network analyzer.

On the other hand, DUT port 2 is a balanced port that cannot be connected directly to the test port of the network analyzer. Usually, a balun (BALance-UNbalance transformer) is used to convert the DUT's balanced port to an unbalanced port and to connect the converted port to the test port of the network analyzer. Matching circuits are mounted in the test fixture as shown in the following figure.

DUT evaluation using an actual test fixture



e5071c404

Problems in Measurement with Actual Test Fixture

Evaluating a balanced device with an actual test fixture involves the following problems:

- Calibration cannot be performed at the DUT's terminals. (A DUT's terminals are in the test fixture and calibration standards cannot be connected to them. In addition, it is very difficult to obtain calibration standards that can be used to calibrate a balanced port.) As a result, calibration is performed at appropriate connectors connected to the test fixture, and the network between the calibration reference plane and the DUT's terminals induces measurement errors.

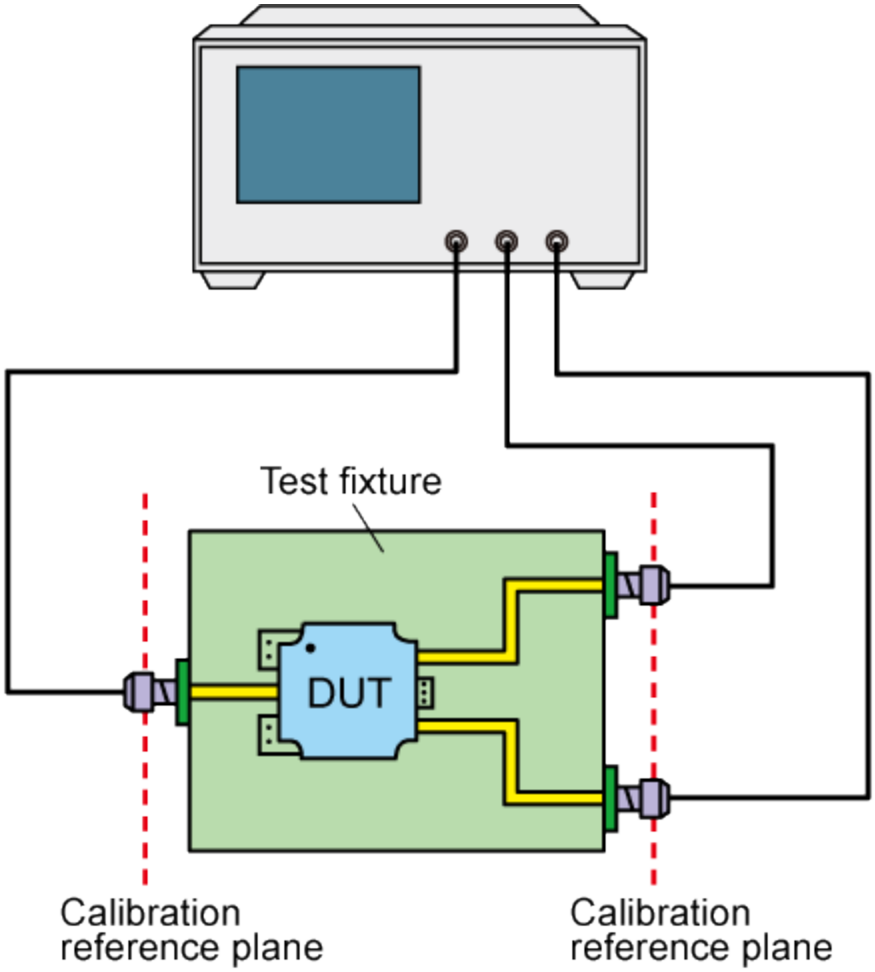
- Different test fixtures must be fabricated for evaluating different types of DUTs because they require different characteristic impedances and matching circuits.
- An actual balun does not have ideal characteristics, so measurement error cannot be avoided. Furthermore, a common mode signal evaluation cannot be performed when an actual balun is used.

DUT Evaluation using E5071C's Fixture Simulator

The E5071C's fixture simulator function simulates a test fixture by using internal software instead of using an actual test fixture for evaluating DUTs.

The following figure shows an example connection for evaluating a DUT with the E5071C's fixture simulator function. The unbalanced port of the DUT should be directly connected to a test port of the E5071C, and the balanced port of the DUT should be connected to two other test ports of the E5071C. The actual measurement by the E5071C is performed at single-ended ports with a 50 ohm port reference impedance.

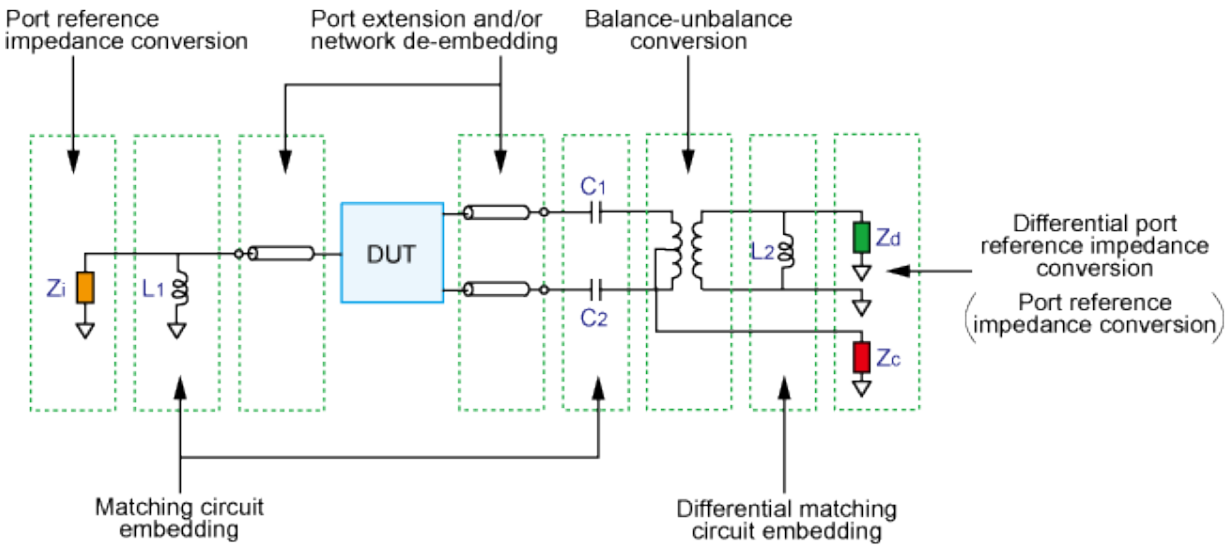
DUT connection when fixture simulator is used



e5071c362

The following figure shows the measurement circuit simulated by the fixture simulator based on actual measurement with the test fixture shown in figure above.

Measurement circuit simulated by fixture simulator



e5071c405

First, the effect of an undesired network can be eliminated by port extension and/or network de-embedding. In the above figure (DUT connection when fixture simulator is used), since calibration standards cannot be connected to the DUT terminals to perform calibration, calibration should be performed at the connectors to the test fixture. Using port extension and/or network de-embedding enables you to remove an undesired network by using data processing and moving the calibration reference plane to the DUT's side equivalently. This function is performed for a single-ended port even if balance-unbalance conversion is applied to the port.

Port reference impedance conversion converts measured S-parameters to those at arbitrary port reference impedance. In the above figure (Measurement circuit simulated by fixture simulator), since the single-ended port of the DUT is connected to the E5071C's test port (50 ohm , single-ended), port reference impedance conversion is not required. This function is performed for a single-ended port even if balance-unbalance conversion is applied to the port.

Matching circuit embedding converts measured S-parameters to those when a matching circuit is added to the DUT's terminal. This function is performed for a single-ended port even if balance-unbalance conversion is applied to the port.

Balance-unbalance conversion converts S-parameters measured at an unbalanced state to mixed-mode S-parameters measured at a balanced state. The balanced port signal can be evaluated by using differential mode and common mode signals.

Differential matching circuit embedding converts measured S-parameters to those when a matching circuit is added to the DUT's differential mode port.(L2 in there figure Measurement circuit simulated by fixture simulator)

Differential port reference impedance conversion converts a differential port reference impedance to an arbitrary impedance. Port reference impedance Z [ohm] at the two single-ended ports before balance conversion is automatically converted to 2Z [ohm] for differential mode port and Z/2 [ohm] for common mode port after balance conversion. Accordingly, if port reference impedance conversion is not performed for the two single-ended ports before balance conversion, differential mode port reference impedance Zd becomes 50 ohm * 2 = 100 ohm , and common mode port reference impedance Zc becomes 50 ohm / 2 = 25 ohm . Since the differential port is terminated with 200 ohm in the figure,Measurement circuit of balance SAW filter, differential port reference impedance Zd should be set to 200 ohm.

Advantages of Balanced DUT Evaluation using Fixture Simulator

Balanced device evaluation using the fixture simulator offers the following advantages:

- Calibration reference plane can be easily moved to the DUT's terminal after calibration is performed at the connectors where calibration standards can be connected. Undesired network can be removed to eliminate measurement errors (port extension, network de-embedding).
- Characteristics of a DUT, including desired matching circuits, can be obtained easily (matching circuit embedding, differential matching circuit embedding). Port reference impedance can be set freely (port reference impedance conversion, differential port reference impedance conversion).
- Differential mode and common mode signal evaluation (mixed-mode S-parameter evaluation) can be performed easily (balance-unbalance conversion).

Setting Control Functions

Remote Control

- Setting the GPIB
- Remote Control Using HTTP

Display

- Turning off the Date/Time Display
- Turning off the LCD Screen Backlight
- Calibration of the Touch Screen

Others

- Initial Source Port Control function
- External Test Set Mode
- Checking the product information
- Activating Software Option
- Locking the Front Keys, Keyboard, and/or Mouse (Touch Screen)
- Setting the Beeper (Built-in Speaker)
- Setting the preset function
- Overload Detection and Power Trip Feature
- Exit/Restart E5071C Measurement Application

Setting the GPIB

- Setting talker/listener GPIB address of E5071C
- Setting system controller (USB/GPIB interface)

Other topics about Setting Control Functions

This section describes how to set the interface necessary to use the GPIB (General Purpose Interface Bus) of the E5071C.

Setting talker/listener GPIB address of E5071C

When controlling the E5071C using GPIB commands from the external controller connected to the GPIB connector, you need to set the talker/listener GPIB address of the E5071C.

Follow these steps to make this setting:

1. Press **System** key.
2. Press **Misc Setup** > **GPIB Setup**> **Talker/Listener Address**.
3. Enter the address using the ENTRY block keys on the front panel.

Setting system controller (USB/GPIB interface)

When controlling an external device from the E5071C, connect the USB port of the E5071C and the GPIB port of the external device through the USB/GPIB interface.

- Do not connect two or more USB/GPIB interfaces.
- In case the volume label of C drive is CN701, CN702, or CN801, the driver for 82357B USB/GPIB is not installed.

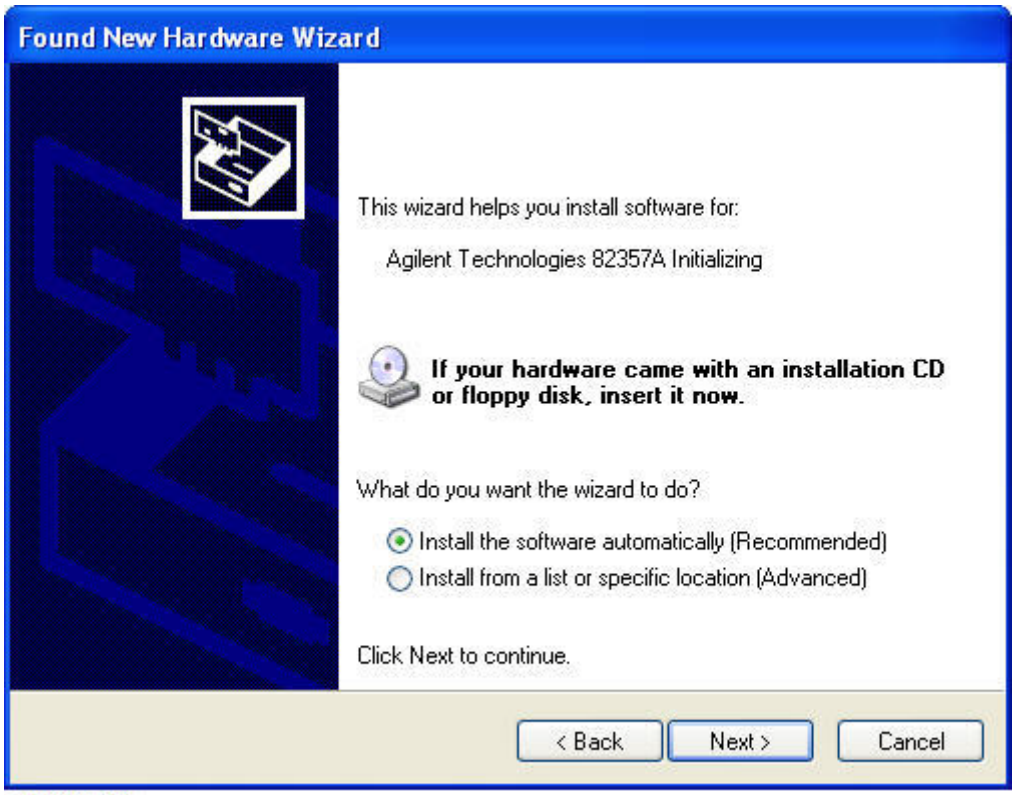
  Before following the steps mentioned below, it is required to install the driver for 82357B.

Follow these steps to set the USB/GPIB interface:

1. Connect the USB port of the E5071C to the USB/GPIB interface.

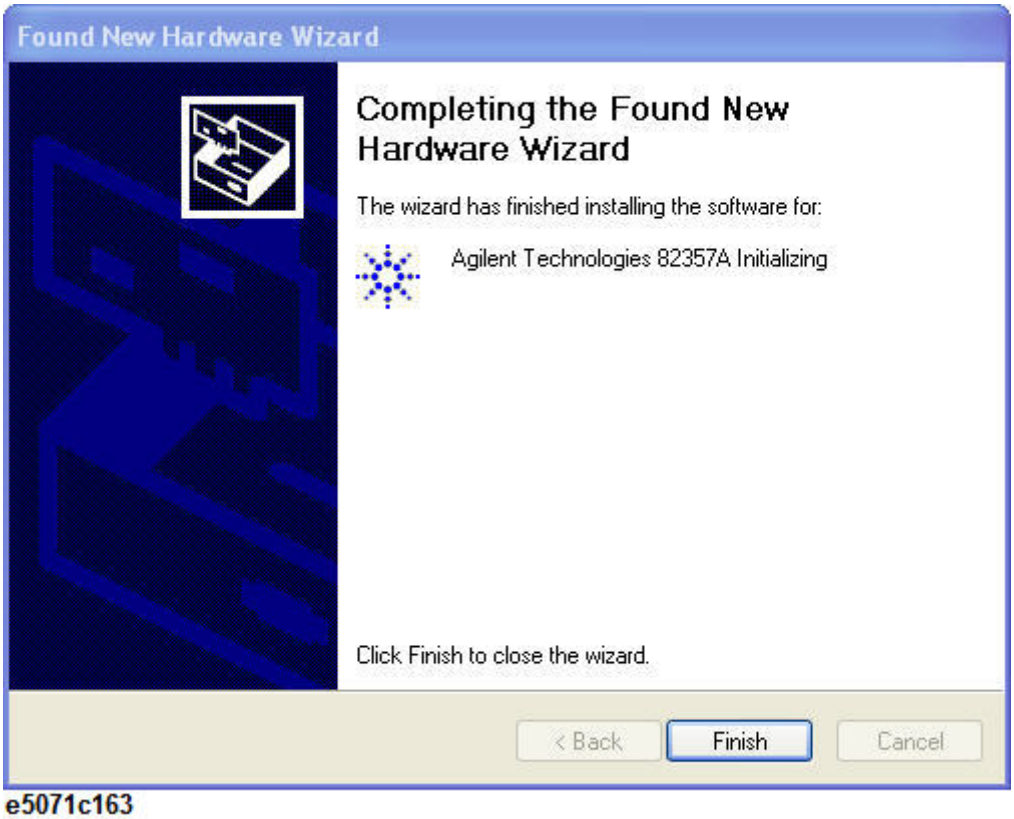
2. Select **No, not this time**, then click **Next**.

3. Select **Install the software automatically (Recommended)**, then click **Next**.



e5071c161

4. Click **Finish**.

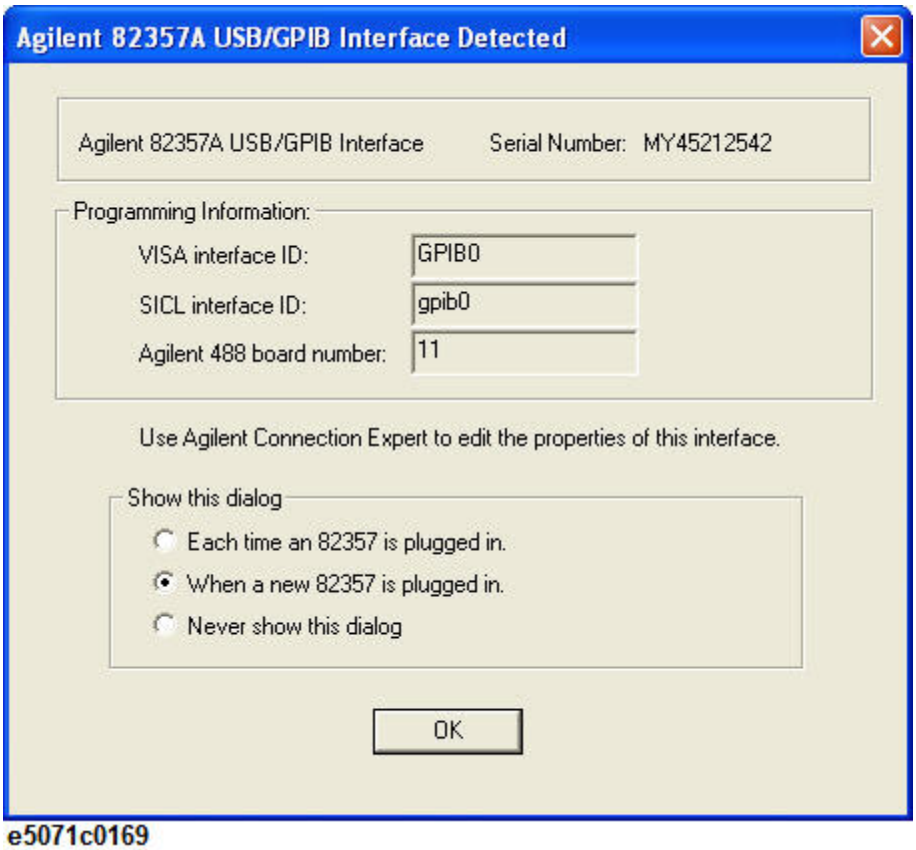


e5071c163

5. The **Found New Hardware Wizard** appears again. Repeat the step 2 to 4 again.

Found New Hardware Wizard
Welcome to the Found New Hardware Wizard
Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).
Read our privacy policy
Can Windows connect to Windows Update to search for software?
Yes, this time only
Yes, now and every time I connect a device
No, not this time
Click Next to continue.
< Back
Next >
Cancel

6. Select **When a new 82357 is plugged in**, then click **OK**.



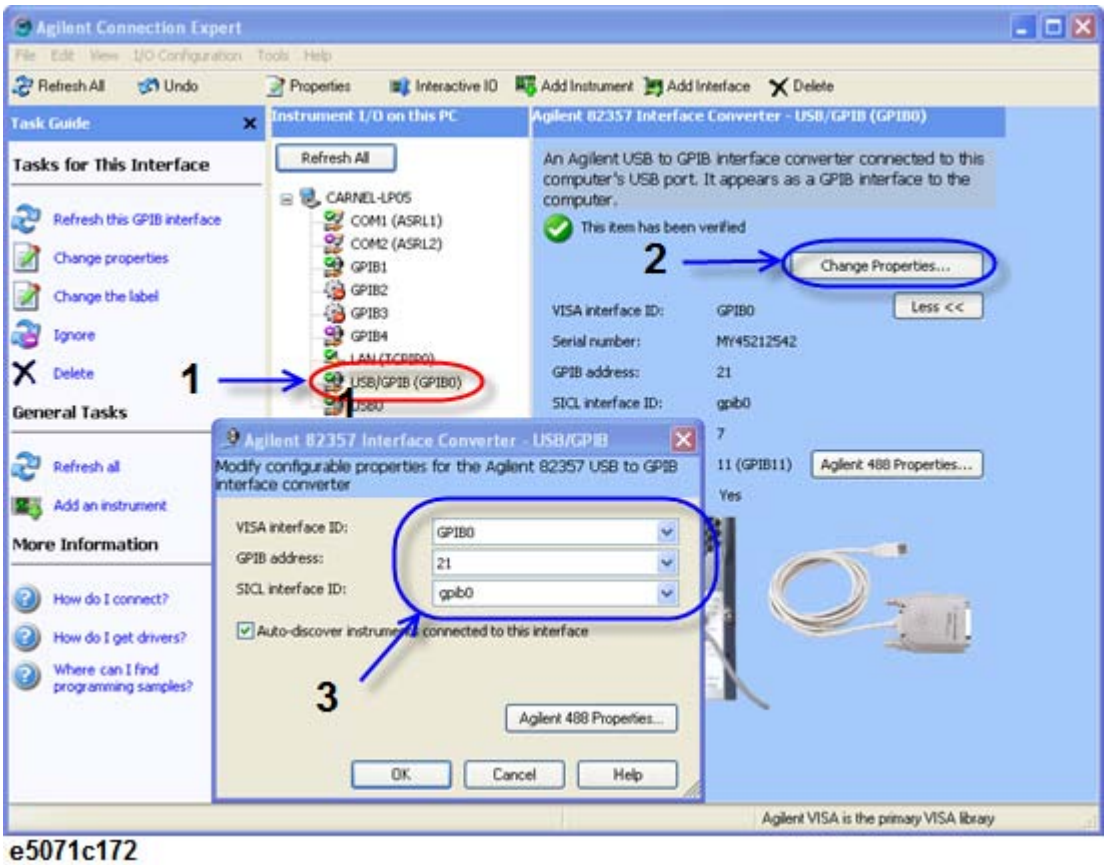
e5071c0169

7. Select "When a new <USB device> is plugged in." or "Never show this dialog" instead of "Each time a USB devices is plugged in".

Changing the setting of the USB/GPIB interface

If you need to check/change the setting of the USB/GPIB interface after connecting the USB/GPIB interface, follow these steps:

1. Press **System** key, then click **Misc Setup** > **GPIB Setup** > **System Controller Configuration**.
2. The Agilent Connection Expert appears. (You can also execute Agilent Connection Expert from Task bar or Start menu in Windows.)



e5071c172

3. Select USB/GPIB (GPIBx)
4. Click Change Propetries...
5. Change the setting of USB/GPIB interface.
    6. It is required to connect the USB/GPIB Interface to "GPIB0" as VISA Interface ID, in order to control:

- Power Meter for the Power Calibration
- External Signal Source for the Frequency Offset

Remote Control Using HTTP

- Enabling Web Server
- Browser Web Control

Other topics about Setting Control Functions

You can access the web page installed in the E5071C by using the hypertext transfer protocol (http) and the E5071C's IP address from the external PC's web browser. This function is called an web-enabled analyzer. Through the built-in web page, you can control the E5071C remotely and display the measurement screen on external PCs.

The following browsers are recommended:

- Internet Explorer 6.0 and later

Enabling Web Server

Network Configuration

To use web server, you have to configure the E5071C's network correctly. For detailed information on configuration and notes, see Configuring the Network.

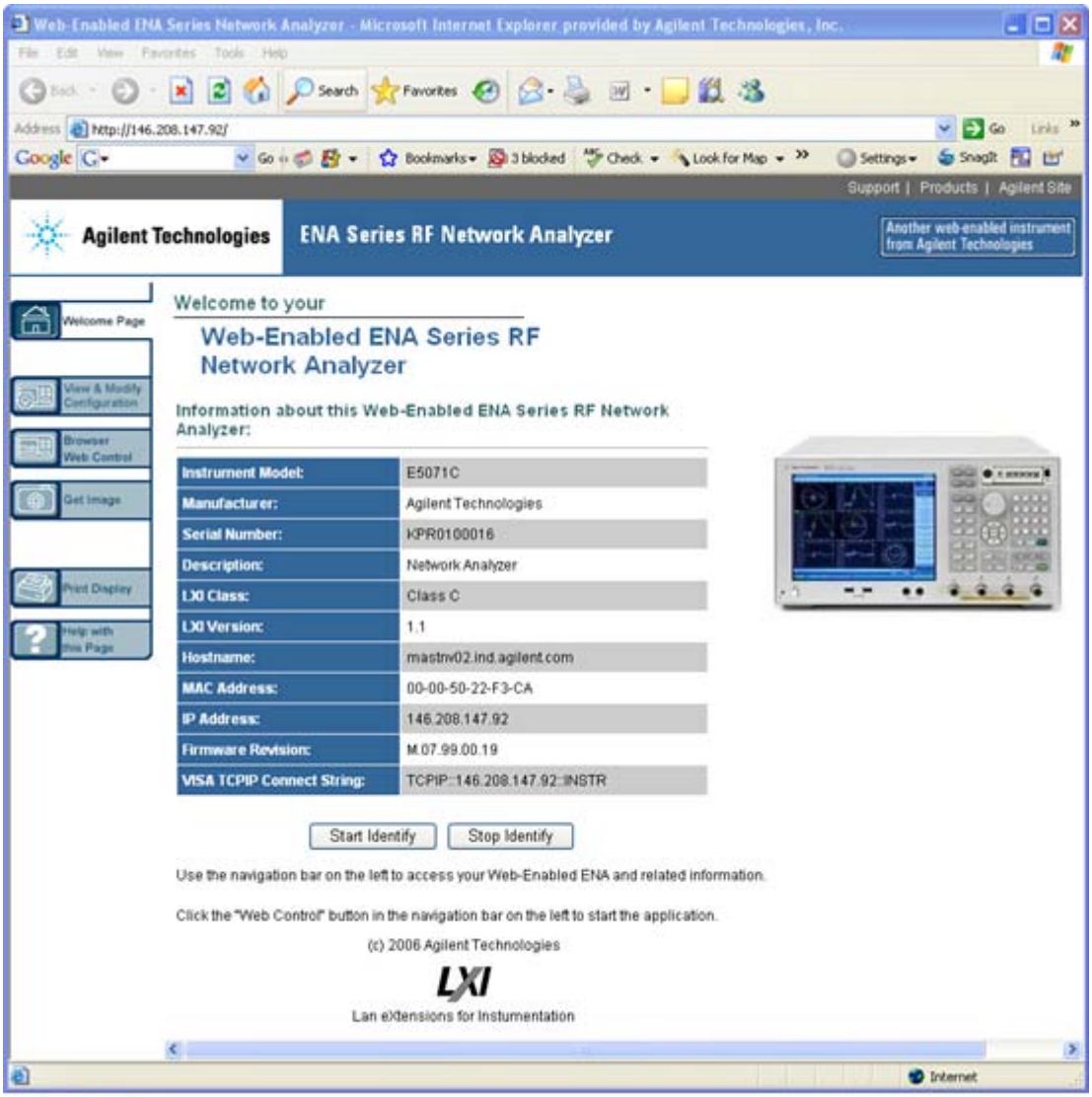
Enabling Web Server

Enable the web server for the E5071C so that it may allow access from an external PC. Follow these steps:

1. Press **System** Key.
2. Click **Misc Setup** > **Network Setup**.
3. Click **Web Server** to turn it **ON**.

Access from an external PC

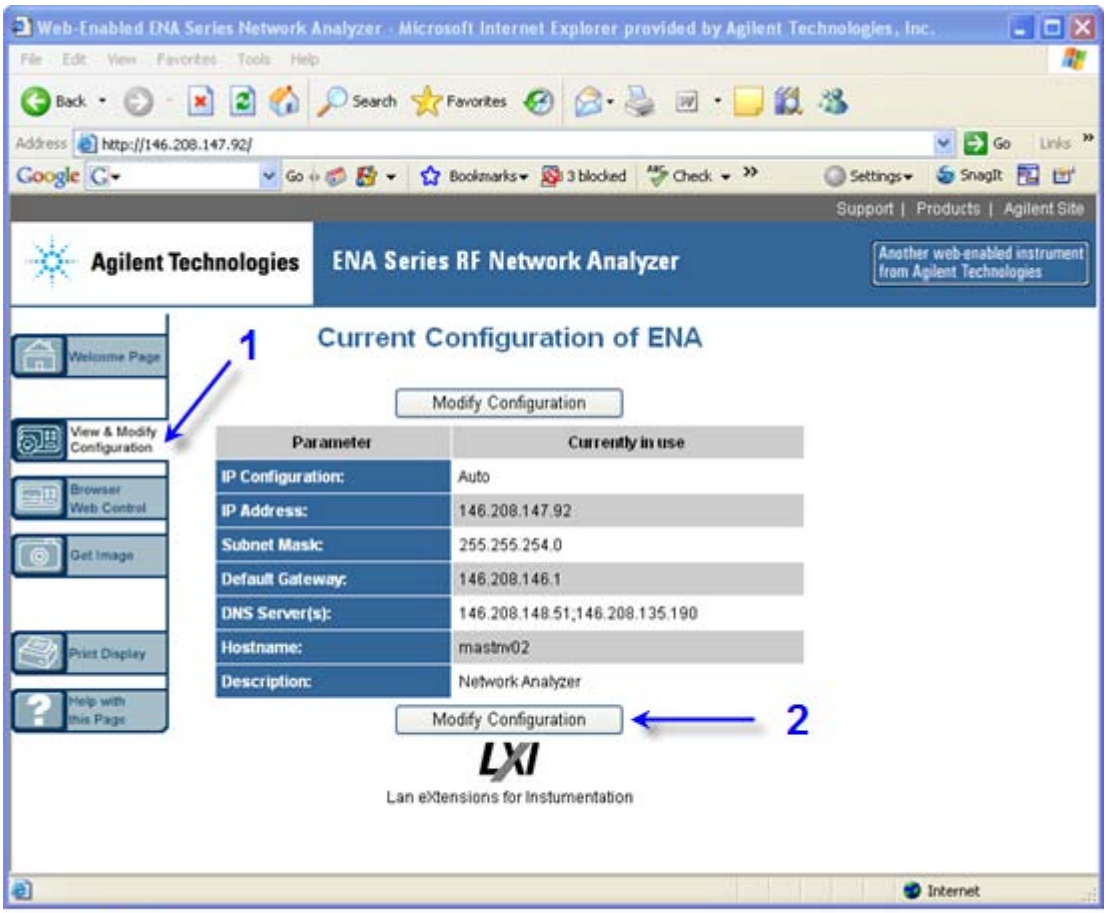
1. Execute web browser on your PC.
2. Check IP address of the E5071C.
3. Enter IP address of the E5071C in the address bar and press Enter. The following screen appears:



e5071c173

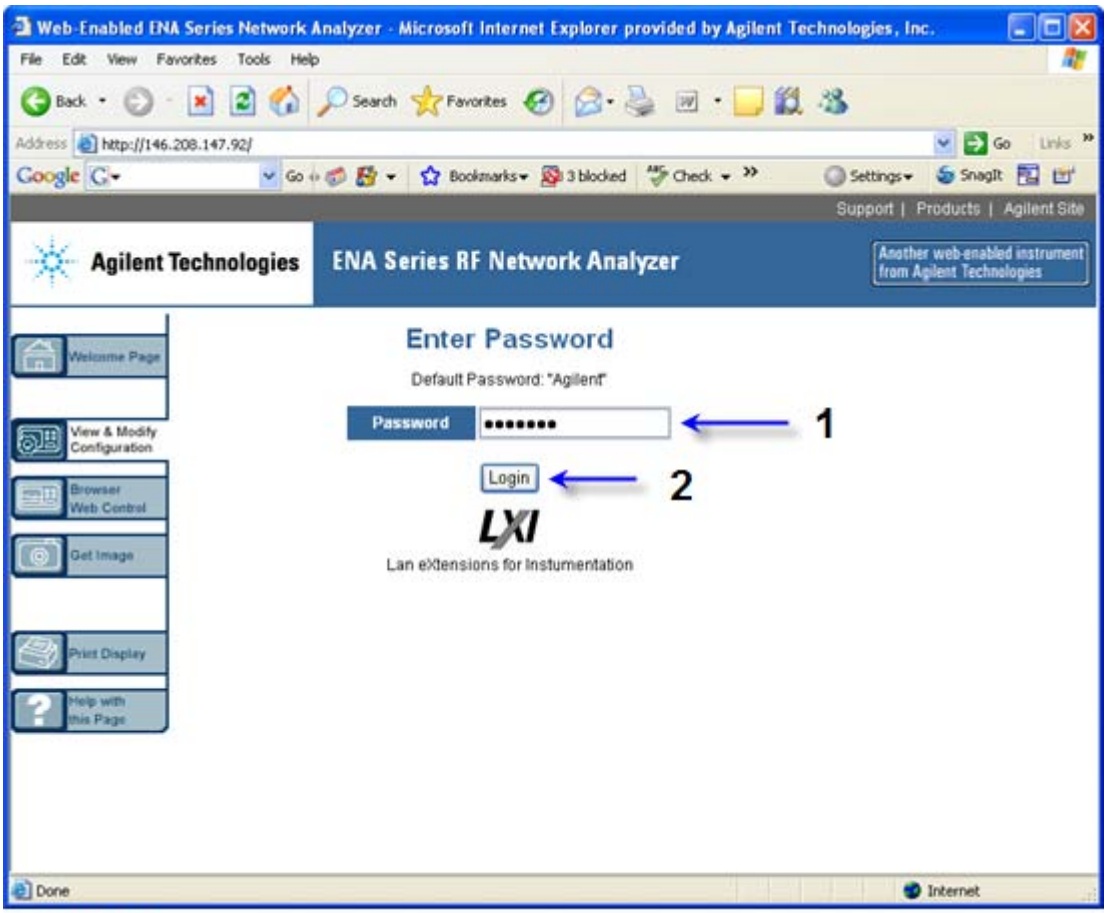
To change LAN (TCP/IP) configration of you ENA.

1. Click **View & Modify Configuration** (**1** in the Figure below). The following screen appears:



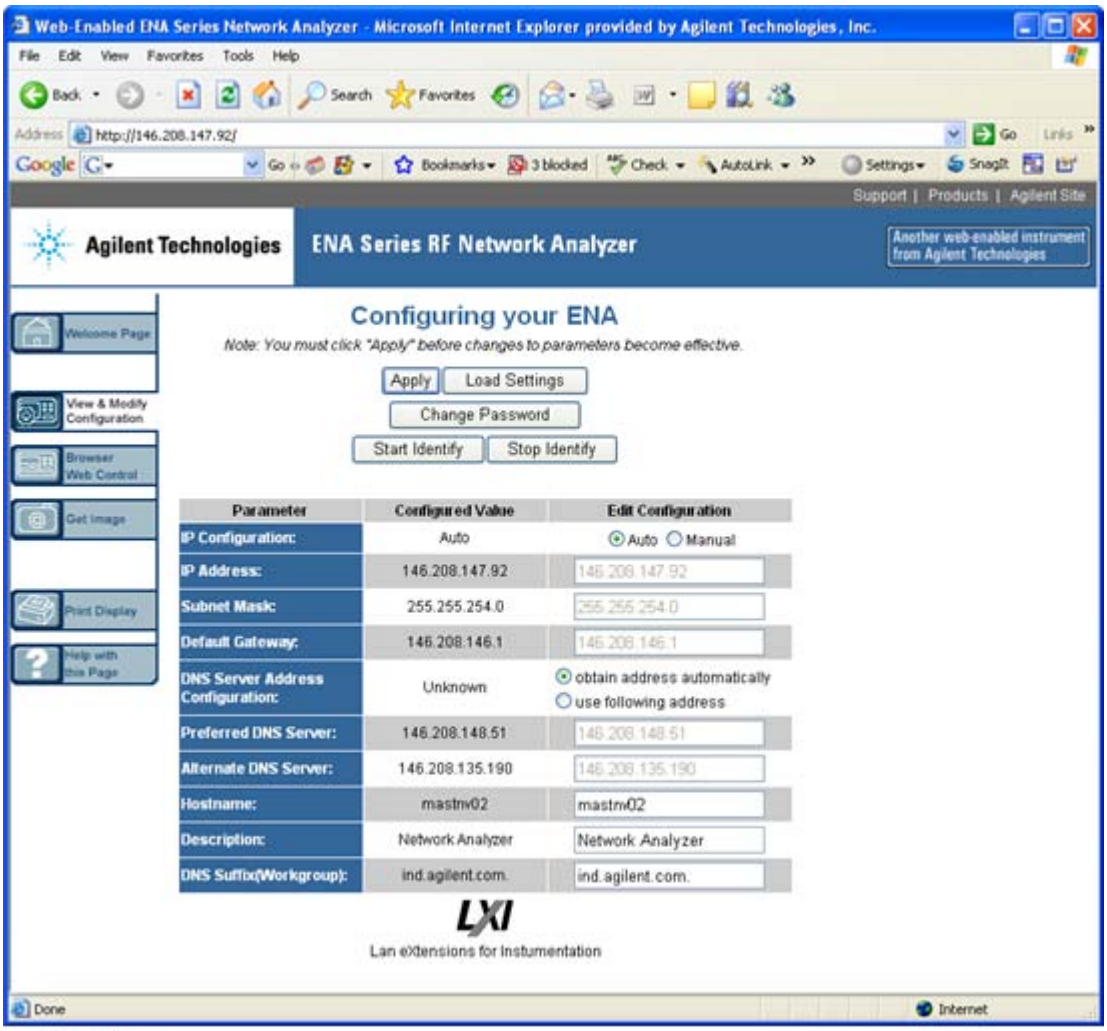
e5071c283

2. Click **Modify Configuration** (**2** in the Figure above). Modifying this setup affects the Windows Internet Protocol (TCP/IP) property. The following screen appears:



e5071c284

3. Enter the password (Default: "Agilent") in the Password field (**1** in the figure above) and click **Login** (**2** in the figure above). The following screen appears:



e5071c285

Browser Web Control

Browser Web Control function allows you to control your ENA from web browser. This function is executed by the VNC server.

- The external PC must have the Java Runtime Environment installed otherwise the Browser Web Control function might not work properly.

The following is a description of how to start the VNC server configuration.

1. Press **System** Key.
2. Click **Misc Setup** > **Network Setup**.

3. Click **VNC Server Configuration** to start the VNC Server Properties.
4. To restrict external access, set a password for the VNC server configuration. The default password at factory shipment is blank.

Turning on/off the Date/Time Display

The date/time display in the instrument status bar can be switched on/off using the following procedure.

1. Press **System** > **Misc Setup** > **Clock Setup**.
2. Click **Show Clock** to toggle the date/time display on/off.

Other topics about Setting Control Functions

Turning off the LCD Screen Backlight

You can switch off the backlight (illumination) of the LCD screen of the E5071C. This extends the life of the backlight when using it continuously over a long period.

Turning off the LCD Screen Backlight

1. Press **System** key.
2. Press **Backlight** to switch the backlight on/off.
3. Switching off the backlight causes indications on the LCD screen to be almost invisible.
4. The backlight that has been switched off can be turned on again by pressing any key from the front panel.

Other topics about Setting Control Functions

Calibration of the Touch Screen

When you have executed system recovery on the E5071C, you have to calibrate the touch screen. Follow the procedure described below to calibrate the touch screen.

1. Press **System** Key.
2. Click **Service Menu** > **Test Menu** > **Adjust Touch Screen**.
3. The touch screen calibration screen appears.
4. Touch the x mark on the upper left with your finger. The mark x appears also on the lower left, upper right, and lower right. Touch the x marks in that order with your finger.
5. Touching the four locations described above with your finger automatically concludes the touch screen calibration.
6. With no operation on the touch screen calibration screen for a preset time, it automatically closes and the previous measurement screen reappears.

Other topics about Setting Control Functions

Initial Source Port Control function

It protects the output amplifier inside the instrument against any potential damage due to transient voltage that may be externally applied. After the firmware is installed, this feature is activated at power-on. When activated, this feature attenuates the output signal and turns off the test port LED each time a sweep finishes. At the same time, it switches the stimulus signal output test port in the trigger hold state to a pre-selected test. A test port 1 is selected at power-on.

The Init Src Ctrl feature lets you select the stimulus signal output test port in the trigger hold state. By setting the stimulus signal output test port to a test port which hardly experience transient voltage, this feature can reduce the possibility that the output amplifiers of the instrument may be damaged by transient voltage. Transient voltage occurs when the active DUT is connected or an external DC power supply is turned on. A test port not used or a test port to which the DUT input terminal is connected can be regarded as a test port which hardly experience transient voltage. The following table shows the degree of tolerance to transient voltage of each test with this feature. Connect the DUT so that transient voltage is not applied to test ports with low tolerance.

Test Port to select **(Stimulus signal output test port)**	Low tolerance test ports Connected to the instrument's output amplifier.(Recommended connection: test port not used or DUT input terminal)	High tolerance test ports connected to 50 ohm termination inside the instrument.(Recommended connection: DUT output terminal)
When test port 1 is selected	Test port 1, test port 3	Test port 2, test port 4
When test port 2 is selected	Test port 2, test port 3	Test port 1, test port 4
When test port 3 is selected	Test port 3, test port 1	Test port 2, test port 4
When test port 4 is selected	Test port 4, test port 1	Test port 2, test port 3

We recommended you to use this feature in single measurement (when performing manual measurement) or :INIT:CONT OFF (when using a program) since the stimulus signal output destination is switched to the selected test port only in the trigger hold state

Usage:

When performing manual measurement

Trigger: **Trigger** > **Single** is recommended.

Turning on the Init Src Ctrl feature: **System** > **Service Menu** > **Init Src Ctrl [ON]**

Specifying the Init Src port: **System** > **Service Menu** > **Init Src Port [1|2|3|4]**

When using a SCPI program. Sample program to set Port 1 to Initial Source Port.

```
:INIT:CONT OFF
:SYSTem:ISPControl[:STATe] ON
:SYSTem:ISPControl:PORT 1
```

To disable the Initial Source Port Control feature:

When using the front panel menu :**System** > **Service** > **Init Src Ctrl [OFF]**

When using the SCPI command : SYSTem:ISPControl[:STATe] OFF

When using the VBA command: SCPI.SYSTem.ISPControl.STATe = False

Other topics about Setting Control Functions

External Test Set Mode

- Overview
- Example of using the external test set mode function
- Setting the external test set mode
- Calibration and Measurement Parameters

Other topics about Setting Control Functions

Overview

To evaluate high-power amplifiers, stimulus signals having greater amplitude than the maximum output power of the E5071C may be required. In such cases, perform the transmission/reflection measurement by configuring an external test set by connecting external couplers/bridges instead of using the S-parameter test set built in the E5071C.

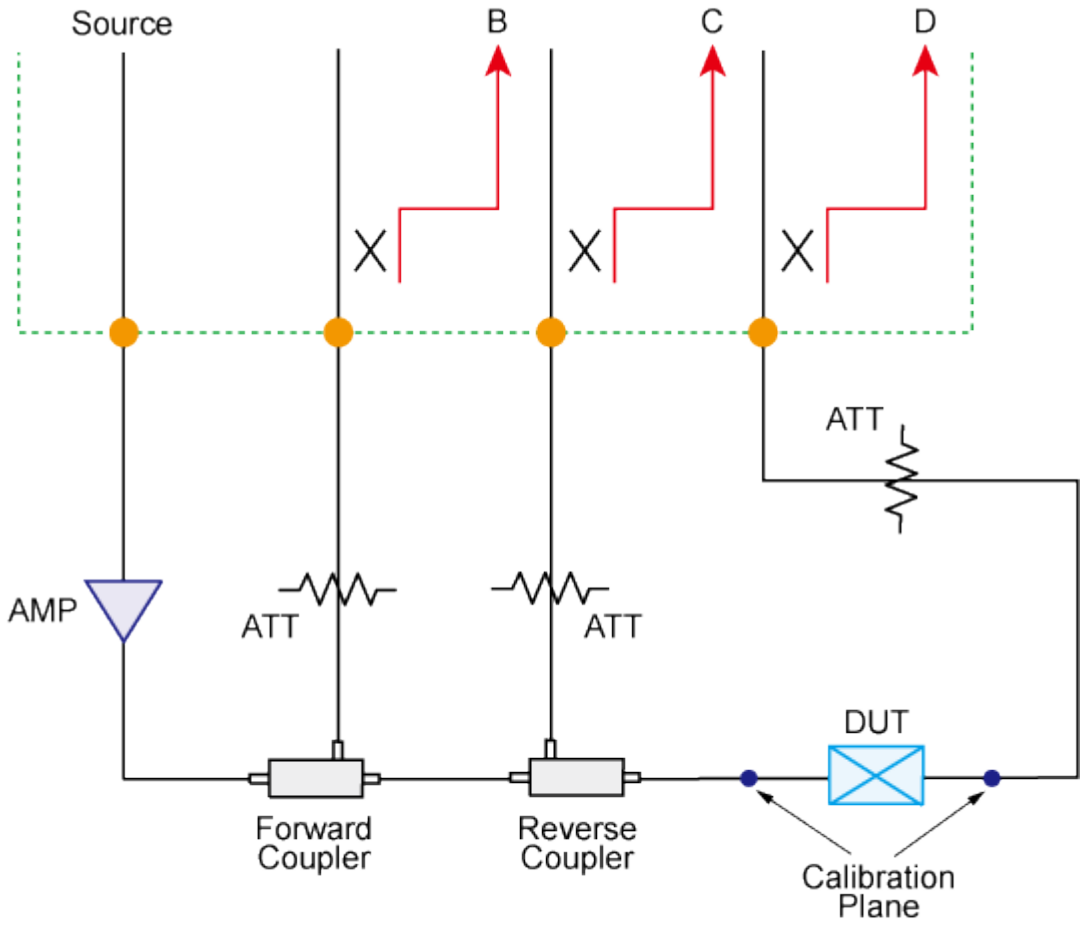
When you want to use the E5071C for such purposes, use the external test set mode function.

Example of using the external test set mode function

To apply amplified output signal of the E5071C to the DUT, an amplifier is connected to the port used for the source to the E5071C. A reverse coupler is inserted to make the reflection measurement. A forward coupler is inserted to use the output of the amplifier as the reference signal. This configuration corrects a source mismatch error caused by the amplifier.

To measure S21 and S11 using this configuration, Mode 1 is selected as the external test set mode.

Configuration using the external test set mode function (Mode 1)

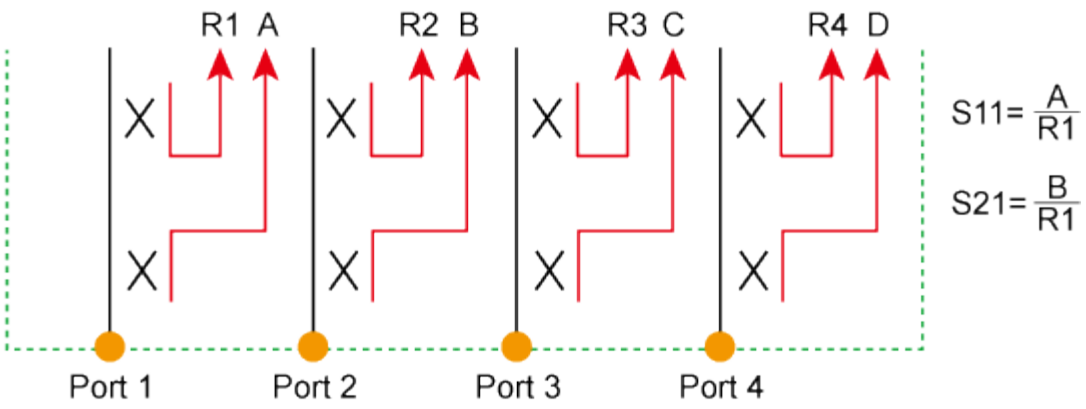


e5071c474

Setting the external test set mode

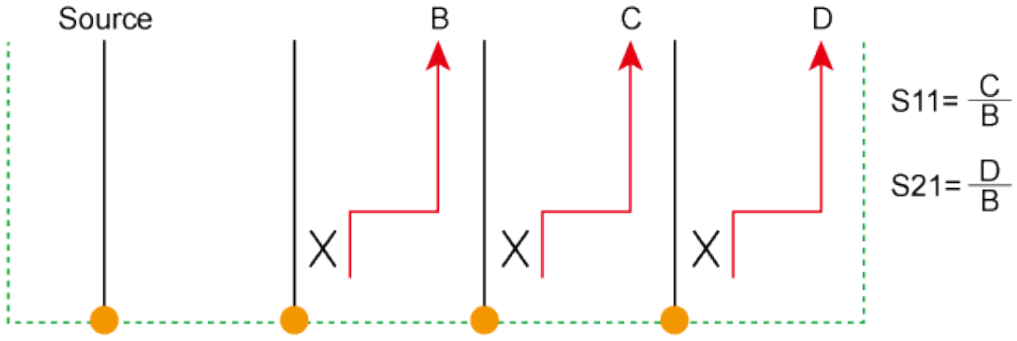
You can select the status of the external test set mode from 3 types: OFF, Mode 1, or Mode 2.

- When OFF is selected, the instrument operates as a network analyzer with a built-in 4-port S-parameter test set.



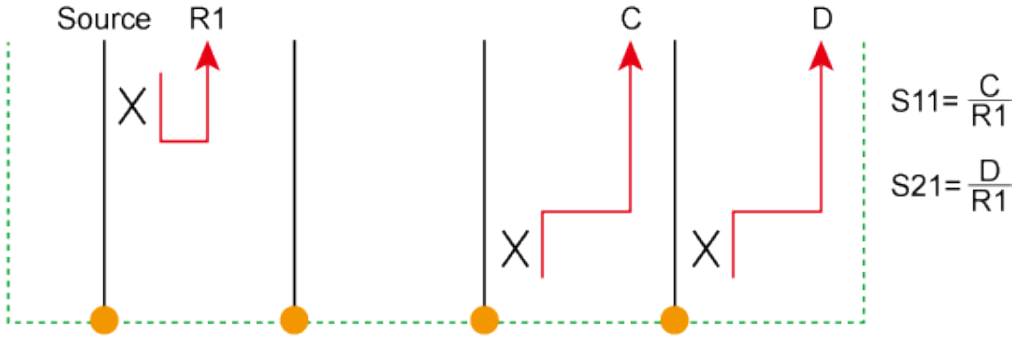
e5071c475

- When Mode 1 or Mode 2 is selected, the instrument operates as a network analyzer with a built-in 2-port T/R test set.
    1. Mode 1



e5071c476

2. Mode 2



e5071c477

Setting procedure

Follow these steps to set the external test set mode.

1. Press **System** > **Service Menu** > **External Test Set**.
2. Select your desired mode.
3. When the setting of the external test set mode is changed, the calibration coefficients acquired using the calibration kit or ECal are cleared, and the error correction function is turned off.

Calibration and Measurement Parameters

When the external test set mode (Mode 1 or Mode 2) is used with the E5071C, only limited types of calibrations and measurement parameters are available.

Calibration when using the external test set

When the external test set mode (Mode 1 or Mode 2) is used with the E5071C, the following calibrations are available. When other calibration methods are used, correct calibration is not performed.

- Open response calibration
- Short response calibration
- Thru response calibration
- Enhanced response calibration
- 1-port calibration
  - When ECal is used with Mode 1 or Mode 2 selected, the connection of test port 4 cannot be detected automatically. When you perform calibration using the 4-port ECal, set the connection of port 4 as follows depending on the connection of test port 1/2/3.

Port of E5071C	Port 1 / 2 / 3	Port 4
Port of ECal	A	B
	B	C
	C	D
	D	A

Measurement parameters

When the external test set mode (Mode 1 or Mode 2) is used with the E5071C, only the S11/S21 measurement and absolute value measurement are available.

When Mode 1 or Mode 2 is selected, the fixture simulator's balance measurement function and frequency offset function are not available.

Checking Product Information

- Checking Serial Number and Options
- Checking Firmware Revision and System Information
- Checking HDD Revision

Other topics about Setting Control Functions

Checking Serial number and Options

The serial number and options of the E5071C can be checked using the following procedure.

1. Press **System** > **Firmware Revision**.
2. The Firmware Revision dialog which shows the serial number and options is displayed.

The information of MWA option is not displayed here as the MWA software is a VBA macro.

To activating software option, see Activating Software Option.

Checking Firmware Revision and System Information

The revision number and system information of the firmware installed in the E5071C can be checked using the following procedure.

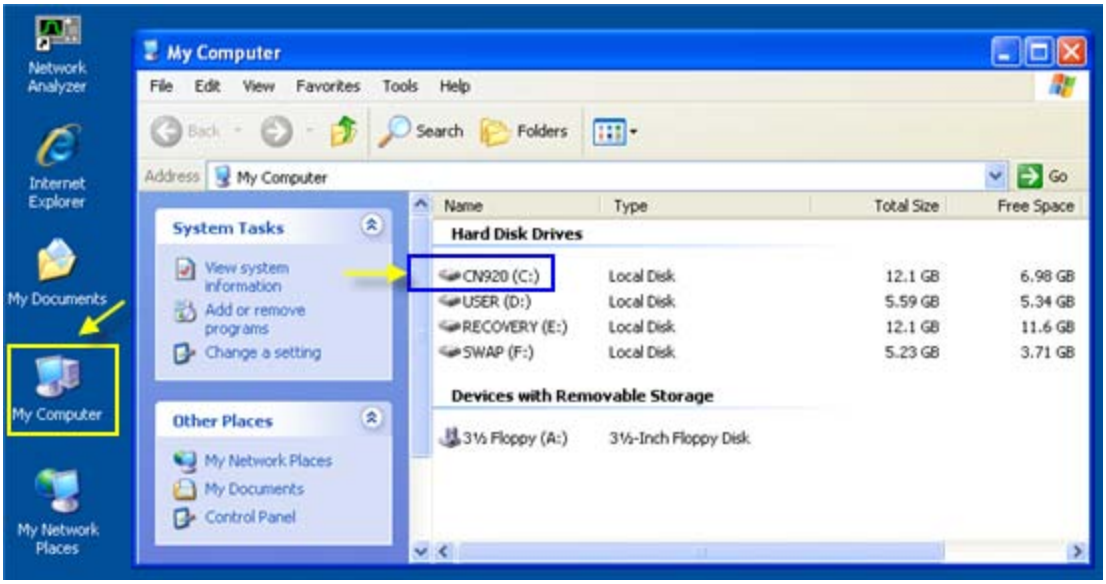
1. Press **System** > **Firmware Revision**.
2. The Firmware Revision dialog is displayed.

NOTE From firmware revision A.08.0x and later, the Firmware Revision dialog box provides information about installed Options, Serial Number, IP Address, Mac Address and USB ID.

NOTE From firmware revision A.09.5x and later,the Firmware Revision dialog box provides information about installed Options, Serial Number, System Information, IP Address, Mac Address and USB ID.

Checking HDD Revision

The HDD revision of the E5071C can be checked using **Windows Start Menu** > **My Computer** > **C:** drive label. For example, the HDD version of the E5071C in the figure below is CN920. For more information, refer to HDD Revision History.


Network Analyzer
Internet Explorer
My Documents
My Computer
My Network Places
My Computer
File Edit View Favorites Tools Help
Back Search Folders
Address My Computer
Go
System Tasks
View system information
Add or remove programs
Change a setting
Other Places
My Network Places
My Documents
Control Panel
Name Type Total Size Free Space
Hard Disk Drives
CN920 (C:) Local Disk 12.1 GB 6.98 GB
USER (D:) Local Disk 5.59 GB 5.34 GB
RECOVERY (E:) Local Disk 12.1 GB 11.6 GB
SWAP (F:) Local Disk 5.23 GB 3.71 GB
Devices with Removable Storage
3½ Floppy (A:) 3½-Inch Floppy Disk

Activating Software Option

- Activating Option
- Backing Up License Key File

Other topics about Setting Control Functions

Activating Option

The software options can be purchased separately to enhance the E5071C measurement functionality. When you purchase the software option upgrade kit, Agilent provides the software entitlement certificate.

Procedure

1.

- Option MWA

  refer to Installation of the MWA Software.

- Other Options
  1. Press **System** > **Service Menu** > **Enable Options** and then select option which you want to activate.
  2. Type the relevant 12 character long license key sent by Agilent in the Key Code entry dialog box, then click **Enter**.
  3. Check the installed option according to Checking_Serial Number and Option.

Note for Option TDR

- Requires the upgrade at Agilent service center.
- Requires the option TDR application on the E5071C.

Locking the Front Keys, Keyboard, and/or Mouse (Touch Screen)

You can lock (disable) the front keys, keyboard, and/or mouse (touch screen). This feature prevents erroneous operation caused by inadvertently touching any of these devices.

Locking the Front Keys, Keyboard, and/or Mouse

1. Press **System** Key.
2. Click **Misc Setup** > **Key Lock**.
3. Click the corresponding key to switch the lock on/off.

Softkey	Function
Front Panel & Keyboard Lock	Switches the lock for the front panel keys and keyboard on/off.
Touch Screen & Mouse Lock	Switches the lock for the touch screen and mouse on/off.

- You cannot use a locked device to unlock that same device. To unlock the front panel keys, keyboard, touch screen and mouse that have been locked, press the Standby switch to turn off the power supply and then turn it on again. When setting at power-on, the front panel keys, keyboard, touch screen and mouse are all in an unlocked condition.

Other topics about Setting Control Functions

Setting the Beeper (Built-in Speaker)

- Setting the Operation Complete Beeper
- Setting the Warning Beeper

Other topics about Setting Control Functions

The E5071C has a built-in speaker that sounds a beep tone. The beeper allows you to make two types of settings.

Type	Function
Operation complete beeper	Sounds a beep tone to inform the user that operations have completed. • When calibration data measurements are done • When data storage has completed
Warning beeper	Sounds a beep tone to prompt the user to use caution. • When an instrument error occurs (An error message appears at the same time.) • When a limit test fails

The warning beeper sounds slightly longer than the operation complete beeper.

Setting the Operation Complete Beeper

1. Press **System** key.
2. Click **Misc Setup** > **Beeper** > **Beep Complete** to switch the operation complete beeper on/off.
3. Clicking **Test Beep Complete** allows you to hear and check the beep tone of the operation complete beeper.

Setting the Warning Beeper

1. Press **System** key.
2. Click **Misc Setup** > **Beeper** > **Beep Warning** to switch the warning beeper on/off.
3. Clicking **Test Beep Warning** allows you to hear and check the beep tone of the warning beeper.

Setting the preset function

- Showing/hiding the confirmation buttons when presetting
- Setting the user preset function
- Saving a user-preset instrument state

Other topics about Setting Control Functions

Showing/hiding the confirmation buttons when presetting

The preset function can be executed without displaying the **OK** and **Cancel** softkey buttons when pressing the preset button of the E5071C.

1. Press **System** key.
2. Click **Misc Setup** > **Preset Setup**.
3. **Confirm** to toggle on (show)/off (hide) the confirmation buttons.

Setting the user preset function

You can save the instrument state of the E5071C into a file in the mass storage, and then recall it with the preset function to reproduce that state.

If no user preset instrument state is stored, you cannot set the user preset function.

1. Press **System** key.
2. Click **Misc Setup** > **Preset Setup** > **State**.
3. Use one of the following keys for the desired setting.

Softkey	Function
Factory	Specifies the normal preset function.
User	Specifies the user-preset function.
Cancel	Returns to the softkey display in one upper level.

Saving a user-preset instrument state

To execute the user-preset function, you must have a preset setting file that has been saved. Follow these steps to save a preset instrument state of the E5071C.

1. Press **Save/Recall** key.
2. Press **Save State** > **User Pres**.

Overload Detection and Power Trip Feature

Overload detection and Power Trip function is a safety feature to protect your E5071C from over-input. When an overload is detected, the power output is automatically turned off by the power trip feature. If the power output is turned off by the power trip, remove the cause of the over-input and turn on the power output to restart the measurement .

When you make a measurement at the frequency between 9.375 MHz and 8.5 GHz, an overload may occurs even if the input signal level is below the maximum input level. In that case, you may turn off overload detection. Even if the overload detection is turned off, overload may occur at the excessive input level.

- When you need to turn off this feature, be sure to control that your input signal is under the maximum input level for specified measurements, and never exceeds the damage level (26 dBm) written on the instrument panel, or the instrument is damaged.

Follow the procedure described below to turn ON/OFF Overload Detection/Power Trip function:

1. Press **System** Key.
2. Click **Service Menu** > **Overload Detect** ON|OFF.

The SCPI command to turn this feature ON/OFF remotely is SCPI.SERVice.POWer:AC:OVERload:DETect:STATe.

Other topics about Setting Control Functions

Exit/Restart E5071C Measurement Application

Pressing **System** > **Service Menu** > **Exit Firmware** shuts down the firmware of E5071C.

Pressing **System** > **Service Menu** > **Restart Firmware** restarts the firmware of the E5071C.

Other topics about Setting Control Functions

Product Information

Product Information

- Options
- Documentations
- Specifications
- Customer Contacts
- Error Messages (Warning Message)
- TDR Error Messages (TDR Warning Message)
- Default Conditions
- Troubleshooting
- Maintenance
- Measurement Accessories
- General Principles of Operation
- Revision History
- Replacing 8753ES with E5071C

Options for E5071C

- Test Port & Frequency Options
- Software Options
- Hardware Options
- Timebase Options
- Accessary Options
- Calibration Option
- System Rack Options

Other topics about Product Information

Test Port & Frequency Options

Frequency Range	Bias Tees	2 Port Test Set	4 Port Test Set	Supported E5071C Firmware Revision
9 kHz to 3 GHz	No	Option 230	Option 430	All
9 kHz to 4.5 GHz		Option 240	Option 440	A.08.00 and above
9 kHz to 6.5 GHz		Option 260	Option 460	A.09.30 and above
9 kHz to 8.5 GHz		Option 280	Option 480	All
100 kHz to 3 GHz	Yes	Option 235	Option 435	All
100 kHz to 4.5 GHz		Option 245	Option 445	A.08.00 and above
100 kHz to 6.5 GHz		Option 265	Option 465	A.09.30 and above
100 kHz to 8.5 GHz		Option 285	Option 485	All
300 kHz to 14 GHz		Option 2D5	Option 4D5	A.09.30 and above
300 kHz to 20 GHz		Option 2K5	Option 4K5	A.09.10 and above

- Options of (9 kHz to 3 GHz) and (100 kHz to 3 GHz) are discontinued and is being replaced by option of (9 kHz to 4.5 GHz) and (100 kHz to 4.5 GHz). Upgrade kit is available. Contact your nearest Agilent Technologies sales office for more details.
- 14/20 GHz options (2D5, 4D5, 2K5 and 4K5) are available to use in -85 dBm to +10 dBm power range. The maximum power level is frequency range dependent.

  Mechanical attenuation is used only in 14/20 GHz options. Due to this, attenuation selection points/timing is different from 4.5/6.5/8.5 GHz options. For more information on attenuation, refer to Setting Power Level with Auto Power Range
-

Software Options

Option Number	Description
008	Frequency Offset Mode
010	Time Domain Analysis
790	Measurement Wizard Assistant Software
TDR	Enhanced Time Domain Analysis

The upgrade (retrofit) kits are available for the software options.

Hardware Options

Option Number	Description
019	Standard (non-removable) HDD
017	Removable HDD

Timebase Options

Option Number	Description
UNQ	Standard Stability Timebase
1E5	High Stability Timebase

Accessary Options

Option Number	Description
810	Add Keyboard
820	Add Mouse

Calibration Option

Option Number	Description
A6J	ANSI Z540 Compliant Calibration
1A7	ISO 17025 Compliant Calibration

System Rack Options

Option Number	Description	Equivalent Agilent Part Number
1CM	Rack Mount Kit	5063-9216
1CN	Front Handle Kit	5063-9229
1CP	Rack Mount and Front Handle Kit	5188-4430

Documentations for E5071C

Manuals for E5071C

The following documentations are provided for E5071C.

Name	Description
Help (This file)	Provides the information about the measurement operation, programming, built-in VBA, I/O interface.
Installation Guide	Provides information about start up setup and system recovery information when the Windows cannot be boot up. This is furnished with E5071C as a paper manual.
Service Guide	Provides information about the parts, troubleshooting, performance test, adjustment and service menu.

The latest revision of Help and Help in PDF, WebHelp formats are also available at the site.

Other topics about Product Information

Specifications

The latest version of data sheet is available at
http://cp.literature.agilent.com/litweb/pdf/5989-5479EN.pdf

For its history, see the Data Sheet Revision History.

Other topics about Product Information

Error Messages

Error Messages

An error message is displayed against a red background in the instrument message/warning area in the lower left part of the screen. Pushing a front panel key or executing :DISP:CCL command clears the error message. Errors caused by the operation of a front panel key simply appear on the display. They are not stored in the error queue with some exceptions.

An error with a positive error number is one uniquely defined for this instrument. On the other hand, an error with a negative error number is basically one defined for common GPIB devices in IEEE488.2

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

A

No.	Message	Description
20	Additional standard needed	The GPIB command that turns ON the calibration function has been sent before all the data measurements required to calculate the calibration factor have been completed. In 1-port calibration, for example, when the measurements are completed for OPEN and SHORT standards but not for LOAD standard. This error occurs when you try to turn ON the calibration function by calculating 1-port calibration coefficient using SENS:CORR:COLL:SAVE or SENS:CORR:OFFS:COLL:SAVE. Be sure to measure all the necessary calibration data before sending commands. This error is not generated by front key operations.
58	Auto port extension not allowed	An auto port extension element has been received at a position where this instrument does not accept one.

B

No.	Message	Description
-	Block	A block-data element has been received at a

168	data not allowed	position where this instrument does not accept one.

C

No.	Message	Description
240	Calibration data lost	This error occurs when a file containing the system calibration data is not found or in a damaged state at time of the startup of this instrument, indicating a failure of this instrument. Contact an Agilent Technology sales office or the company from which you bought the instrument.
22	Calibration method not selected	This error occurs when the command for validating the calibration data, SENS:CORR:COLL:SAVE or SENS:CORR:OFFS:COLL:SAVE is executed before the command for selecting a calibration type, SENS:CORR:COLL:METH:xxxx or SENS:CORR:OFFS:COLL:METH:xxxx, is executed. This error is not generated by front key operations.
-148	Character data not allowed	A character data element (not violating the standard) has been received at a position where this instrument does not accept one. Double quotes (") are omitted where it is necessary to place a parameter in double quotes ("), for example.
-100	Command error	A comprehensive syntax error has occurred showing that this instrument cannot detect a more detailed error. This code simply shows that a command error defined in 11.5.1.1.4, IEEE488.2 has occurred.
60	Continuous switching may damage source attenuator	This error occurs when different power ranges are selected in multiple channel measurement settings to avoid source attenuator damage. • This error occurs in case of options 2D5, 4D5, 2K5 and 4K5.

D

No.	Message	Description
-222	Data out of range	A data element (not violating the standard) outside the range defined by this instrument has been received. This error occurs when an integer-based command for which the parameter can be rounded exceeds the range of -65536 to +65536 or when a real-number-based command for which the parameter can be rounded exceeds the range of -9.9e37 to +9.9e37, for example. This error occurs also when a numeric value other than a specified one is entered into a command in which the "port number" and "CalKit number" are specified as parameters and hence the parameters are not rounded. Such commands are, for example, CALC:FSIM:BAL:TOP:BBAL:PPOR, SENS:CORR:COLL:ACQ:OPEN, SENS:CORR:COLL:ECAL:SOLT3, SENS:CORR:COLL:CKIT:ORD:LOAD, etc.
-104	Data type error	The parser has recognized a data element that must not exist. Block data has been sent instead of numeric value data or character string data that had been expected, for example.
63	Duplicate trace name	This error occurs when a trace name defined with CALCulate[1-160]:PARameter[1-16]:TNAMe:DATA command already exist.

E

No.	Message	Description
32	ECal module not in appropriate RF path	This error occurs when an ECal command, SENS:CORR:COLL:ECAL:SOLTn or SENS:CORR:OFFS:COLL:ECAL:xxxx, is executed with the port on the ECal module not connected correctly to the instrument. When the auto-detect function is turned OFF, (SENS:CORR:COLL :ECAL:ORI OFF), however, this error does not occur even when the port on the ECal module is not connected correctly to the instrument.
502	Equation run time	This error occurs under the following conditions:

	error	• The trace number in data(tr)/mem(tr)/xAxis(tr) is out of range # of trace available depends on the maximum number of channel/traces • The port number in Advanced math function is out of range # of port available depends on the model option
-200	Execution error	An error associated with execution has been generated for which this instrument cannot specify the error message. This code shows that an error associated with execution defined in 11.5.1.1.5, IEEE488.2 has occurred. This error occurs also when a calibration measurement is aborted.
-123	Exponent too large	The absolute value of the exponent exceeds 32,000 (see 7.7.2.4.1, IEEE488.2).
-178	Expression data not allowed	An expression-data element has been received at a position where this instrument does not accept one.
-170	Expression error	When the expression data is put to syntactic analysis, an error not corresponding to one of Error Numbers -171 through -179 occurs.
27	External test set enabled	When the external test set mode is turned ON, the allowable calibration is limited. See Calibration and Measurement Parameters.

F

No.	Message	Description
31	Failed to configure ECal module	This error occurs when the control of the ECal module fails at time of executing an ECal command, SENS:CORR:COLL:ECAL:SOLTn or SENS:CORR:OFFS:COLL:ECAL :xxxx. The failure results from the failure to connect the ECal module to the USB port, failure of the ECal module, etc.
76	Failed to control	This error occurs when the external signal source fails to respond during measurement, though it is

	signal generator	enabled (SENS:OFFS:LOC:STAT ON and SENS:OFFS:LOC:CONT ON). Ensure the connection of the external signal source.
102	Failed to copy file	This error occurs when copying a file (MMEM:COPY command) fails.
104	Failed to create directory	This error occurs when creating a directory (MMEM:MDIR command) fails.
103	Failed to delete file	This error occurs when deleting a file (MMEM:DEL command) fails.
600	Failed to simulate eye	This error occurs when eye diagram simulation fails.
100	Failed to read file	This error occurs when a 2-port touchstone file (CALC:FSIM:SEND:PMC:PORT:USER:FIL command), the formatted data array (MMEM:LOAD:FDAT command) and limit table (MMEM:STOR:LIM command) for the active trace on the active channel, segment sweep table (MMEM:LOAD:SEGM command) for the active channel, a VBA project file (MMEM:LOAD:PROG command), etc. cannot be read normally.
601	Failed to read bit file	This error occurs when reading bit file fails.
101	Failed to write file	This error occurs when the formatted data array (MMEM:STOR:FDATcommand) and limit table (MMEM:STOR:LIM command) for the active trace on the active channel, segment sweep table (MMEM:STOR:SEGM command) for the active channel, display image (MMEM:STOR:IMAG command) for the LCD screen, a VBA project file (MMEM:STOR:PROG command), etc. cannot be written normally.
-	File name	A file name error. This message appears when an error exists in the file name and hence a

257	error	command is not executed correctly. This error occurs when you try to copy to an unsuitable file name, for example.
-256	File name not found	The file name specified is not found and hence the command is not executed correctly. This error occurs when you try to read a file that does not exist in a disk or a disk is not correctly inserted into the drive to read or write a file, for example.
107	File transfer failed	This error occurs when writing data into or reading data from a file (MMEM:TRAN command) fails.
56	Fixture Simulator not allowed	This error occurs when the fixture simulator is executed (CALC:FSIM:STAT ON) under the condition of the frequency offset is enabled (SENS:OFFS ON). The fixture simulator is allowed to use when the frequency offset is not enabled.
24	Frequency offset enabled	This error occurs when the normal calibration (SENS:CORR:COLL:OPEN etc.) is performed under the condition of the frequency offset is enabled (SENS:OFFS ON).
59	Frequency offset not allowed	This error occurs when frequency offset function is not allowed in the current setting.
25	Frequency offset not enabled	This error occurs when the scalar-mixer calibration (SENS:CORR:OFFS:COLL:OPEN etc.) is performed under the condition of the frequency offset is not enabled (SENS:OFFS OFF).
62	Frequency out of range	This error occurs when the setting of frequency exceeds output frequency range under the condition of the frequency offset is enabled (SENS:OFFS ON).

G

No.	Message	Description
-105	GET not	A group execution trigger (GET) has been received in the program message (see 7.7,

	allowed	IEEE488.2).

H

No.	Message	Description
-114	Header suffix out of range	The unit of the header is outside the range. The header is invalid in the unit for numeric parameters following a SCPI command.

I

No.	Message	Description
-224	Illegal parameter value	The parameter value is not suitable. This error occurs when the CALC:PAR:DEF command is used to specify an S-parameter that does not exist in the model (S44 in the case of a 2-port model), for example.
-282	Illegal program name	This error occurs when a nonexistent VBA program name is specified by the PROG:SEL:NAME command.
-213	Init ignored	Because another measurement is in progress, the request for initiating a measurement ("INIT" command) is ignored.
35	Insufficient ECal module memory	This error occurs when the embedded memory is insufficient to save the user property in ECal module during the user definition ECal processing.
-161	Invalid block data	Block data has been expected, but the block data that appears is invalid for some reason (see 7.7.6.2, IEEE488.2). The END message is received before the length of block data has been filled, for example.
28	Invalid calibration method	This error occurs when the type of calibration is not specified or not correct when partial overwrite is executed with the GPIB command. This error does not occur for operation with front keys.
-101	Invalid character	An invalid character exists in the program message character string.

-141	Invalid character data	An invalid character is found in the character data element, or the parameter received is not valid.
-121	Invalid character in number	A character that is invalid for the data type subject to syntactic analysis has been received. For example, a letter is found in a decimal numeric value or a numeric character "9" in octal data.
500	Invalid equation expression	The equation expression used in Equation Editor is not valid.
501	Invalid equation label	The equation label used in Equation Editor is not valid.
-171	Invalid expression	The expression-data element is invalid (see 7.7.7.2, IEEE488.2). Parentheses are not paired, or illegal characters are used, for example.
602	Invalid mask data	This error occurs when invalid mask data is loaded.
-103	Invalid separator	The parser (a syntactic analysis program) had been expecting a delimiter, but a character that is not a delimiter has been sent.
-151	Invalid string data	Character string data has been expected, but the character string data that appears is invalid for some reason (see 7.7.5.2, IEEE488.2). The END message is received before the ending quotation mark character appears, for example.
-131	Invalid suffix	The suffix does not comply with the syntax defined in 7.7.3.2, IEEE488.2. Or it does not suit E5071C.

L

No.	Message	Description
53	Log sweep	The span of sweep range is not satisfied the

	requires 2 octave minimum span	requirement for logarithmic sweep. The sweep type is automatically changed to linear sweep when this error occurs. For example, this error occurs when, with the start and stop frequency are set 1 MHz and 2 MHz respectively, the sweep type is changed to logarithmic sweep. Set the stop frequency to more than four times as many as the start frequency. And then select logarithmic sweep.

M

No.	Message	Description
-109	Missing parameter	The number of parameters is less than that required for the command, or the parameter has not been entered. For example, the command SENS{1-6}:SWE:POIN requires one more parameter. Therefore, when a message "SENS1:SWE:POIN" is sent to a correct program message "SENS1:SWE:POIN 201" this instrument receives the former message as an invalid one because all parameters have not been entered. Enter command parameters correctly.

N

No.	Message	Description
70	No GPIB system controller	The GPIB system controller is not set properly. This error occurs when USB/GPIB interface is not connected or the interface setting is improper, during the measurement for power calibration data or for that in scalar-mixer calibration. It also occurs when USB/GPIB interface is not connected or the interface setting is improper, though the control of the external signal source is enabled.

		Ensure the connection of USB/GPIB interface and its setting.
52	No valid memory trace	This error occurs when you have executed either DISP:WIND:TRAC:MEM ON command to display memory trace, or any other command to enable data calculation using memory trace (CALC:MATH:FUNC command with other than NORM is specified), though no valid data exists in memory trace. This error is not generated by front key operations.
-120	Numeric data error	An error resulting from the numeric value data (including numeric value data having no decimal point representation) has occurred. A numeric value error other than Errors -121 through -129 has occurred.
-128	Numeric data not allowed	An numeric-value-data element (not violating the standard) has been received at a position where this instrument does not accept one.

O

No.	Message	Description
200	Option not installed	The command received has been ignored because of the mismatch between the contents of an option for this instrument and the command. It also occurs when you try to enable the time domain function in a model not having the time domain option. This holds true for the frequency offset option. This error is not generated by front key operations.
-225	Out of memory	Insufficient memory is available in this instrument to perform the required operation.

P

No.	Message	Description

-220	Parameter error	When a parameter-related error other than Errors -221 through -229 occurs, that error is displayed.
-108	Parameter not allowed	The number of parameters exceeds that required for the command. For instance, when a program message ":SENS1:SWE:TYPE LIN, SEGM" is sent instead of a correct program message with a command ":SENS1:SWE:TYPE LIN" which requires a parameter, the instrument receives the message as the number of parameters is invalid. See the command reference to confirm the required number of parameters.
41	Peak not found	This error occurs when, after specifying a peak and executing the CALC:MARK:FUNC:EXEC and CALC:FUNC:EXEC commands, the specified peak is not found in the marker search analysis.
220	Phase lock loop unlocked	This error occurs when the PLL circuit of this instrument becomes unlocked while the measurement is in progress. The measurement value is not correct. This error may occur when an external reference out of specification is connected to this instrument. Should an error occur with an external reference not connected, this instrument is faulty. Contact an Agilent Technology sales office or the company from which you bought the instrument.
221	Port 1 receiver overload	The input to Test Port 1 exceeds the maximum input level. The measurement value is not correct. When a DUT is an amplifier or the like, this error may occur, damaging the receiver in the worst case. Should this error occur with a passive part used as the DUT or with nothing connected to the test port, this

		instrument is faulty. Contact an Agilent Technology sales office or the company from which you bought the instrument. If this error occurs, the stimulus signal output is automatically turned to off. • For more information, see Overload Detection and Power Trip Function.
222	Port 2 receiver overload	The input to Test Port 2 exceeds the maximum input level. The measurement value is not correct. When a DUT is an amplifier or the like, this error may occur, damaging the receiver in the worst case. Should this error occur with a passive part used as the DUT or with nothing connected to the test port, this instrument is faulty. Contact an Agilent Technology sales office or the company from which you bought the instrument. If this error occurs, the stimulus signal output is automatically turned to off. • For more information, see Overload Detection and Power Trip Function.
223	Port 3 receiver overload	The input to Test Port 3 exceeds the maximum input level. The measurement value is not correct. When a DUT is an amplifier or the like, this error may occur, damaging the receiver in the worst case. Should this error occur with a passive part used as the DUT or with nothing connected to the test port, this instrument is faulty. Contact an Agilent Technology sales office or the company from which you bought the instrument. If this error occurs, the stimulus signal output is automatically turned to off. • For more information, see Overload Detection and Power Trip Function.

224	Port 4 receiver overload	The input to Test Port 4 exceeds the maximum input level. The measurement value is not correct. When a DUT is an amplifier or the like, this error may occur, damaging the receiver in the worst case. Should this error occur with a passive part used as the DUT or with nothing connected to the test port, this instrument is faulty. Contact an Agilent Technology sales office or the company from which you bought the instrument. If this error occurs, the stimulus signal output is automatically turned to off. • For more information, see Overload Detection and Power Trip Function.
73	Power meter not settled	The measurement value of the power meter is unstable, or it goes outside the tolerance. This error occurs when the power meter's measurement value does not become stable within about 10 seconds during measurement for power calibration data or that for scalar-mixer calibration data. This error also occurs when the power calibration value exceeds the set tolerance. In addition, it occurs when the power meter fails to respond for any reason, such as power-down for example, during measurement under the same conditions. Ensure a firm connection of the power sensor.
241	Power on test failed	This error occurs when the power-on test fails, indicating a failure of this instrument. Contact an Agilent Technology sales office or the company from which you bought the instrument.
61	Power	The out power level exceeds available

	unleveled	range. For example, if the level after correction exceeds the power level that can be outputted when correcting the power level with the power calibration and power slope feature, this error occurs. Check that the power level is set correctly, the power calibration has been performed correctly and the correction value of the power slope is set correctly.
120	Printer error	This error occurs when the previous printing is still in progress or the printer fails (offline, short of paper, etc.) at time of outputting the display image on the LCD screen to the printer (HCOP:IMM command).
121	Print failed	This error occurs when printing fails for reasons other than Error 120, Printer error.
-284	Program currently running	This error occurs when the PROG:SEL:STAT RUN command is executed with the VBA program in the Run state.
-112	Program mnemonic too long	The length of the header exceeds 12 characters (see 7.6.1.4.1, IEEE488.2).
-286	Program runtime error	An error occurring when VBA is executed.

Q

No.	Message	Description
-430	Query DEADLOCKED	The state that generates a "DEADLOCKED" Query error (see 6.3.1.7, IEEE488.2). This error occurs when both input and output buffers have become full, preventing the instrument from continuing processing, for example.

-400	Query error	A comprehensive query error has occurred showing that this instrument cannot detect a more detailed error. This code simply shows that a query error defined in 11.5.1.1.7 and 6.3, IEEE488.2 has occurred.
-410	Query INTERRUPTED	The state that generates a "INTERRUPTED" Query error (see 6.3.2.3, IEEE488.1). This error occurs when data bytes (DAB) or GET are received before the transmission of the response after a query has not been completed, for example.
-420	Query UNTERMINATED	The state that generates an "UNTERMINATED" Query error (see 6.3.2, IEEE488.2). This error occurs when this instrument is designated as the talker and an incomplete program message is received, for example.
-440	Query UNTERMINATED after indefinite response	After a query asking for an indefinite response has been run, another query is received in the same program message (See 6.5.7.5.7, IEEE488.2).
-350	Queue overflow	The queue contains a specific code in place of the code which caused this error. The code indicates that the error occurred because of no space available in the queue, but the error is not recorded.

R

No.	Message	Description
105	Recall failed	This error occurs when reading an instrument status file (State01.sta, etc.) (MMEM:LOAD:STAT command) fails.

S

No.	Message	Description

106	Save failed	This error occurs when writing an instrument status file (State01.sta, etc.) (MMEM:STOR:STAT command) fails.
33	Selected parameter not valid for confidence check	This error occurs when the mix mode S parameter has been selected for the S parameter you want to check, while using the confidence check function for calibration coefficient.
57	SnP request not valid for selected measurement	This error occurs when you try to save data to a Touchstone file but no measurement has been executed. This error also occurs when you try to save a Touchstone file with power sweep measurement specified or with the frequency offset function set to ON.
50	Specified channel hidden	This error occurs when an attempt is made to activate a channel not on display using the DISP:WIND:ACT command. This error is not generated by front key operations.
23	Specified error term does not exist	The error occurs when the calibration coefficient type of the parameter specified in the read/write command of the calibration coefficient (SENS:CORR:COEF) is invalid for the calibration type selected by the calibration type selection command (SENS:CORR:COEF:METH:xxxx.) This error is not generated by front key operations.
21	Specified ports overlapped	This error occurs when a port number is duplicated in a command requiring two or more port numbers as parameters. Such commands are, for example, CALC:FSIM:BAL:TOP:SSB:PPOR 1,2,3,3. Specify port setup correctly to

		avoid duplication of ports. This error is not generated by front key operations.
51	Specified trace dose not exist	This error occurs when CALC:PAR:SEL command is executed to activate more traces than specified by CALC:PAR:COUN command. This error is not generated by front key operations.
26	Standard not selected	This error occurs when you execute the measurement under the condition that the standard is not selected for the subclass (the standard number is set to 0).
-150	String data error	When a character-string-data element is put to syntactic analysis, an error not corresponding to one of Error Numbers -151 through -159 occurs.
-158	String data not allowed	A character-string-data element has been received at a position where this instrument does not accept one.
-138	Suffix not allowed	A suffix is attached to a numeric value element to which a suffix is not allowed to be attached.
-134	Suffix too long	The unit is too long. The unit is expressed in 12 or more characters (see 7.7.3.4, IEEE488.2).
55	Sweep mode changed to stepped sweep.	You cannot change the sweep mode to the swept mode. This error occurs when you change the sweep mode to the swept mode/high speed swept mode under the conditions where the sweep type is set to power sweep, power slope function is enabled (ON, correction coefficient is other than zero), power calibration is ON, frequency offset is enabled, spurious avoidance of frequency offset is enabled, or external signal source

		control is enabled.
-102	Syntax error	A command or data type that is not recognized exists.
-310	System error	One of the errors designated as "system errors" in this instrument has occurred.

T

No.	Message	Description
40	Target value not found	This error occurs when the target is not found during the marker search analysis after specifying the target and executing the CALC:MARK:FUNC:EXEC and CALC:FUNC:EXEC commands. This error occurs also when the bandwidth is not found after executing the bandwidth marker command, CALC:MARK:BWID:DATA?
242	TDR option requires advanced system calibration	TDR requires the system information PCC2.00 and above. This instrument should be sent to service centre for adjustment (with the firmware Rev 9.50 and above) to suit E5071C-TDR option.
-124	Too many digits	The number of digits of the argument of the decimal numeric-value-data element exceeds 255 with the preceding 0 removed (see 7.7.2.4.1, IEEE488.2).
-223	Too much data	The block-, expression-, or character-string-type program data that has been received conforms with the standard. But it exceeds the amount that can be processed under the condition of the memory or conditions specific to memory-related devices. In this instrument, this error occurs when the number of

		characters exceeds 254 in a character-string parameter.
54	Transform, Gate not allowed	This message appears when you turn on the gating/conversion function of the time domain feature, while frequency offset is enabled, measurement points are set to two, or sweep type is set to other than linear sweep Turn off the frequency offset (SENS:OFFS OFF command), set the measurement points to three or more and the sweep type to linear sweep, and then turn on the gating/conversion function.
-211	Trigger ignored	This instrument receives and detects a trigger command ("TRIG") or an external trigger signal. But it is ignored due to the timing condition (This instrument is not in the wait-for- trigger state, for example). Change the setup so that a trigger command or an external trigger signal can be sent after the instrument has entered the wait-for-trigger state.

U

No.	Message	Description
300	Unable to estimate adapter length	Adapter length can not be estimated at zero span. Change the frequency span other than 0.
-113	Undefined header	A command not defined in this instrument, though not illegal in the syntactic structure, has been received. For example, when a message ":DISP:WIND1:TABL:MEM ON" is sent to a correct program message ":DISP:WIND1:TRAC1:MEM ON," the message sent is received as

		an undefined command by this instrument. See the command reference and use correct commands. This error occurs also when a port not existing on this model is specified in a command specifying a port number as an index. Such commands are CALC:FSIM:SEND:DEEM:PORTn:xxxx, CALC:FSIM:SEND:PMC:PORTn:xxxx, CALC:FSIM:SEND:ZCON:PORTn:Z0:R, and SENS:CORR:EXT:PORTn:TIME; they include PORTn as a part.
34	User characterization not found in module	This error occurs when the selected user profile is not detected in the ECal memory, while reading it from the ECal module, written by the user definition ECal.

V

No.	Message	Description
30	Valid Ecal module not found	This error occurs when the number of ports of the ECal module connected is less than the necessary number of ports. This error occurs, for example, when a 4-port Cal executing command, SENS:CORR:COLL:ECAL:SOLT4, is executed with a 2-port ECal module connected. This error is not generated by front key operations.
71	Valid power meter not found.	Valid power meter is not connected . This error occurs when the E5071C and the power meter is not connected with USB/GPIB interface, the GPIB addresses do not match between the power meter and the E5071C's power meter, or the power meter is turned off, during the measurement for power calibration data or for that in scalar-

		mixer calibration. It also occurs when inoperable power meter is connected or an unspecified power sensor is set using :SYST:COMM:PSEN command. Ensure the connection of the power meter, GPIB addresses and power supply.
72	Valid power sensor not found. No valid sensor is connected to the power meter.	This error occurs when no power sensor is connected to the specified channel of the power meter, during the measurement for power calibration data or for that in scalar-mixer calibration. Ensure the connection of the power sensor.
75	Valid signal generator not found. No valid external signal source is connected.	This error occurs when the specified type of external signal source is not connected to the specified GPIB address, though the control of the external signal source is enabled (SENS:OFFS:LOC:STAT ON and SENS:OFFS:LOC:CONT ON). It also occurs when USB/GPIB interface is set but not connected. Ensure the external signal source and the connection of USB/GPIB interface.

Warning Message

A warning message is displayed in the instrument message/Warning area in the lower left part of the display against a gray background. Pushing a front panel key or executing :DISP:CCL command clears the message.

This message simply appears on the display, being not known to a remote environment such as a GPIB. This message is not displayed when another error (against a red background) has already been displayed in the instrument message/Warning area.

Messages	Description
Auto port extension not allowed	This message appears when you try to execute the auto port extension when the sweep type is power sweep or the frequency offset function is ON. If this message is displayed, the auto port extension is not available.
Calibration extrapolated	The frequency range that can be calibrated with the ECal module is exceeded. In this case, the calibration data for the minimum frequency or maximum frequency are used instead of the data exceeding the allowable frequency range for calibration.
Failed to set zero type of power meter	This error occurs when zero type (External or Internal) of power meter cannot be performed.
Failed to zero and calibrate power meter	This error occurs when zeroing or calibration of power meter cannot be performed.
Fixture simulator not allowed	This message appears when enabling the frequency offset, while the fixture simulator function is turned on. If this message is displayed, the fixture simulator function is automatically turned off.
Log sweep requires 2 octave minimum	If you change the sweep range that does not satisfy the necessary condition of the

span	frequency span (the stop frequency is about 4 times or more of the start frequency) when the sweep type is set to the log sweep, this message is displayed. If this message is displayed, the sweep type is automatically set to the linear sweep. For example, if you attempt to change the log type to the log sweep when the start frequency is set to 1 MHz and the stop frequency to 2 MHz, this error occurs. Set the stop frequency to a value of about 4 times or more the start frequency and then set the low seep.
Peak not found	This warning message is displayed when, with the tracking turned on, the peak specified by the marker search has not been found by the time the sweep is finished (with the tracking executed).
Segment table changed	This warning message is displayed when the setting specified segment by segment in the segment table is automatically changed by a change in the other setting. For example, this warning message is displayed when, with the power specified segment by segment in the segment table, the power setting for a segment is adjusted by a change in the power range setting.
Sweep mode changed to stepped sweep	This message appears when you set the sweep type to power sweep, power slope function to enabled (ON, correction coefficient is other than zero), power calibration to ON, frequency offset to enabled, spurious avoidance of frequency offset to enabled, or external signal source control to enabled, while the sweep mode is set to swept mode. When this message is displayed, the sweep mode is automatically set to step mode (during swept mode).
Target value not	This warning message is displayed when, with

found	the tracking turned on, the target specified by the marker search has not been found by the time the sweep is finished (with the tracking executed). This warning message is displayed also when, with the bandwidth marker displayed, the setting for the bandwidth marker is changed at the end of the sweep, or when, with the *active marker* changed or moved, the bandwidth is not found.
Transform, Gate not allowed	This message appears when you have set the measurement points to two, sweep type to other than linear sweep, or frequency offset to enabled, while the gating/conversion function of the time domain feature is turned on. If this message is displayed, the gating/conversion function is automatically turned off.
User Preset File not found	This warning message is displayed if no user preset file (**D:\UserPreset.sta**) exists when executing the SYST:UPR command. If this message is displayed, a normal preset (SYST:PRES) will be performed.

TDR Error Message

TDR Error Messages

An error message is displayed on a dialog box. In E5071C-TDR application on PC (remote control), the error message is displayed at the Error Message area of the application.

An error with a positive error number is one uniquely defined for this instrument. On the other hand, an error with a negative error number is basically one defined for common GPIB devices in IEEE488.2

A B C D E F G H I J K L M N O P Q R S T U V W X Y Z

B

No.	Message	Description
-168	Block data not allowed	An block-data element has been received at a position where this instrument does not accept one.

C

No.	Message	Description
-148	Character data not allowed	A character data element (not violating the standard) has been received at a position where this instrument does not accept one. Double quotes (") are omitted where it is necessary to place a parameter in double quotes ("), for example.
-100	Command error	A comprehensive syntax error has occurred showing that this instrument cannot detect a more detailed error. This code simply shows that a command error defined in 11.5.1.1.4, IEEE488.2 has occurred.

D

No.	Message	Description
-222	Data out of range	A data element (not violating the standard) outside the range defined by this instrument has been received. This error occurs when an integer-based command for which the parameter can be

		rounded exceeds the range of -65536 to +65536 or when a real-number-based command for which the parameter can be rounded exceeds the range of -9.9e37 to +9.9e37, for example.
-104	Data type error	The parser has recognized a data element that must not exist. Block data has been sent instead of numeric value data or character string data that had been expected, for example.

E

No.	Message	Description
-200	Execution error	An error associated with execution has been generated for which this instrument cannot specify the error message. This code shows that an error associated with execution defined in 11.5.1.1.5, IEEE488.2 has occurred. This error occurs also when a calibration measurement is aborted.
-123	Exponent too large	The absolute value of the exponent exceeds 32,000 (see 7.7.2.4.1, IEEE488.2).
-178	Expression data not allowed	An expression-data element has been received at a position where this instrument does not accept one.
-170	Expression error	When the expression data is put to syntactic analysis, an error not corresponding to one of Error Numbers -171 through -179 occurs.

F

No.	Message	Description
100	Failed to read file	This error occurs when load eye-mask file (:MMEM:LOAD:EYE:MASK) and load user defined bit pattern file (:MMEM:LOAD:EYE:BPAT) cannot be read correctly.
101	Failed to write file	This error also occurs when formatted data (:MMEMy:STOR:FDAT), displayed image (:MMEM:STOR:IMAG) and measurement data (:MMEM:STOR:SNP) cannot be stored correctly.

-257	File name error	A file name error. This message appears when an error exists in the file name and hence a command is not executed correctly. This error occurs when you try to copy to an unsuitable file name, for example.
-256	File name not found	The file name specified is not found and hence the command is not executed correctly. This error occurs when you try to read a file that does not exist in a disk or a disk is not correctly inserted into the drive to read or write a file, for example.

H

No.	Message	Description
-114	Header suffix out of range	The unit of the header is outside the range. The header is invalid in the unit for numeric parameters following a SCPI command.

I

No.	Message	Description
-224	Illegal parameter value	The parameter value is not suitable. This error occurs when the CALC:TRAC:PAR command is used to specify an S-parameter that does not exist in the model (S44 in the case of a 2-port model), for example.
601	Invalid bit pattern	An invalid bit pattern is entered or loaded.
-161	Invalid block data	Block data has been expected, but the block data that appears is invalid for some reason (see 7.7.6.2, IEEE488.2). The END message is received before the length of block data has been filled, for example.
28	Invalid calibration method	This error occurs when execution of load measurement (:SENS:CORR:COLL:DLC:LOAD), open measurement (:SENS:CORR:COLL:DLC:OPEN), thru measurement as part of DLC sequence (:SENS:CORR:COLL:DLC:THRU) or

		registration of DLC results (:SENS:CORR:COLL:DLC:SAVE) is not performed correctly.
-101	Invalid character	An invalid character exists in the program message character string.
-141	Invalid character data	An invalid character is found in the character data element, or the parameter received is not valid.
-121	Invalid character in number	A character that is invalid for the data type subject to syntactic analysis has been received. For example, a letter is found in a decimal numeric value or a numeric character "9" in octal data.
-171	Invalid expression	The expression-data element is invalid (see 7.7.7.2, IEEE488.2). Parentheses are not paired, or illegal characters are used, for example.
602	Invalid mask data	An invalid mask data is loaded.
-103	Invalid separator	The parser (a syntactic analysis program) had been expecting a delimiter, but a character that is not a delimiter has been sent.
-151	Invalid string data	Character string data has been expected, but the character string data that appears is invalid for some reason (see 7.7.5.2, IEEE488.2). The END message is received before the ending quotation mark character appears, for example.
-131	Invalid suffix	The suffix does not comply with the syntax defined in 7.7.3.2, IEEE488.2. Or it does not suit E5071C.

M

No.	Message	Description
-109	Missing	The number of parameters is less than that

	parameter	required for the command, or the parameter has not been entered. For example, the command "SENS:AVER:COUN" requires one more parameter. Therefore, when a message "SENS:AVER:COUN" is sent to a correct program message "SENS:AVER:COUN 16" this instrument receives the former message as an invalid one because all parameters have not been entered. Enter command parameters correctly.

N

No.	Message	Description
-120	Numeric data error	An error resulting from the numeric value data (including numeric value data having no decimal point representation) has occurred. A numeric value error other than Errors -121 through -129 has occurred.
-128	Numeric data not allowed	An numeric-value-data element (not violating the standard) has been received at a position where this instrument does not accept one.

O

No.	Message	Description
-225	Out of memory	Insufficient memory is available in this instrument to perform the required operation.

P

No.	Message	Description
-220	Parameter error	When a parameter-related error other than Errors -221 through -229 occurs, that error is displayed.
-108	Parameter not allowed	The number of parameters exceeds that required for the command. For instance, when a program message

		":CALC:DEV SEND1, SEND2" is sent instead of a correct program message with a command ":CALC:DEV SEND1" which requires a parameter, the instrument receives the message as the number of parameters is invalid. See the command reference to confirm the required number of parameters.
-112	Program mnemonic too long	The length of the header exceeds 12 characters (see 7.6.1.4.1, IEEE488.2).

Q

No.	Message	Description
-430	Query DEADLOCKED	The state that generates a "DEADLOCKED" Query error (see 6.3.1.7, IEEE488.2). This error occurs when both input and output buffers have become full, preventing the instrument from continuing processing, for example.
-400	Query error	A comprehensive query error has occurred showing that this instrument cannot detect a more detailed error. This code simply shows that a query error defined in 11.5.1.1.7 and 6.3, IEEE488.2 has occurred.
-410	Query INTERRUPTED	The state that generates a "INTERRUPTED" Query error (see 6.3.2.3, IEEE488.1). This error occurs when data bytes (DAB) or GET are received before the transmission of the response after a query has not been completed, for example.
-420	Query UNTERMINATED	The state that generates an "UNTERMINATED" Query error (see 6.3.2, IEEE488.2). This error occurs when this instrument is designated as the talker and an incomplete program

		message is received, for example.
-440	Query UNTERMINATED after indefinite response	After a query asking for an indefinite response has been run, another query is received in the same program message (See 6.5.7.5.7, IEEE488.2).
-350	Queue overflow	The queue contains a specific code in place of the code which caused this error. The code indicates that the error occurred because of no space available in the queue, but the error is not recorded.

S

No.	Message	Description
-150	String data error	When a character-string-data element is put to syntactic analysis, an error not corresponding to one of Error Numbers -151 through -159 occurs.
-158	String data not allowed	A character-string-data element has been received at a position where this instrument does not accept one.
-138	Suffix not allowed	A suffix is attached to a numeric value element to which a suffix is not allowed to be attached.
-134	Suffix too long	The unit is too long. The unit is expressed in 12 or more characters (see 7.7.3.4, IEEE488.2).
-102	Syntax error	A command or data type that is not recognized exists.
-310	System error	One of the errors designated as "system errors" in this instrument has occurred.

T

No.	Message	Description
-124	Too many digits	The number of digits of the argument

		of the decimal numeric-value-data element exceeds 255 with the preceding 0 removed (see 7.7.2.4.1, IEEE488.2).
-223	Too much data	The block-, expression-, or character-string-type program data that has been received conforms with the standard. But it exceeds the amount that can be processed under the condition of the memory or conditions specific to memory-related devices. In this instrument, this error occurs when the number of characters exceeds 254 in a character-string parameter.
242	TDR option requires advanced system calibration	Advanced system calibration has to be performed on the E5071C so that the TDR option functions accurately.

U

No.	Message	Description
-113	Undefined header	A command not defined in this instrument, though not illegal in the syntactic structure, has been received. Check the command syntax.

TDR Warning Message

An error message is displayed on a dialog box.

This message simply appears on the display, being not known to a remote environment such as a GPIB. This message is not displayed when another error (against a red background) has already been displayed in the instrument message/Warning area.

Messages	Description
TDR option is not found	This message appears when the E5071C-TDR option is not installed completely or correctly. Reinstall the E5071C-TDR option. Restart the option.
E5071C is not in TDR mode	This message appears when E5071C-TDR option is not turned ON. Turn ON the E5071C-TDR option: **Analysis** > **TDR - ON**. If its already at ON state, then click **Start TDR**.
Failed to connect <*address*>	This message appears when there's start-up error in the E5071C-TDR option remote mode. <*address*> shows the specified VISA address.
E5071C ENA option TDR already exist	This message appears if E5071C-TDR option is turned ON when there's another E5071C-TDR option already turned ON and is running.
Invalid state file	This message appears when an invalid state file is recalled. Recall a valid state file, in *.tdr format.
Unsupported revision	This message appears when the firmware revision does not support the E5071C-TDR option revision, because one of the version is older than the other. Check the compatibility of the firmware revision and E5071C-TDR revision and install accordingly.

Advanced/cal mode mismatch	This message appears in E5071C-TDR option when you recall a setting which was saved in Advanced Mode, in Basic Mode. Note that, when you save the setting in the Advanced Mode, you cannot recall the file in the Basic mode. Recall the setting in the Advanced Mode.

Default Conditions

Analysis

Key Operation			Default Value	Preset	*RST	Save/Recall
Fixture Simulator	Fixture Simulator		OFF	←	←	√
	Topology	Device	SE-BAL	←	←	√
		Port1 (se)	1 (When SE-Bal is selected as the device) (When SE-SE-Bal is selected as the device)	←	←	√
		Port1 (bal)	1-2 (When Bal-Bal is selected as the device)	←	←	√
		Port2 (bal)	2-3 (SE-Bal is selected as the device) 3-4 (Bal-Bal is selected as the device)	←	←	√

		Port2 (se)	2 (When SE-SE-Bal is selected as the device)	←	←	√
		Port3 (bal)	3-4 (When SE-SE-Bal is selected as the device)	←	←	√
		Property	OFF	←	←	√
	BalUn		OFF	←	←	√
	Port Matching	Port Matching	OFF	←	←	√
		Select Port	1	←	←	√
		Select Circuit	None	←	←	√
		C	0.000e-12F	←	←	√
		G	0.000S	←	←	√
		L	0.000e-9H	←	←	√
		R	0.0000 ohm	←	←	√
	Port Z Conversion	Port Z Conversion	OFF	←	←	√
		Port1 Z0 Real	50.000 ohm	←	←	√
		Port1 Z0 Imag	0.000 ohm	←	←	√

		Port2 Z0 Real	50.000 ohm	←	←	√
		Port2 Z0 Imag	0.000 ohm	←	←	√
		Port3 Z0 Real	50.000 ohm	←	←	√
		Port3 Z0 Imag	0.000 ohm	←	←	√
		Port4 Z0 Real	50.000 ohm	←	←	√
		Port4 Z0 Imag	0.000 ohm	←	←	√
	De-Embedding	De-Embedding	OFF	←	←	√
		Select Port	1	←	←	√
		Select Type	None	←	←	√
	Diff Matching	Diff Matching	OFF	←	←	√
		Select Bal Port	2	←	←	√
		Select Circuit	None	←	←	√
		C	0.0000 F	←	←	√
		G	0.0000 S	←	←	√
		L	0.0000 H	←	←	√
		R	0.0000 ohm	←	←	√
	Diff Z Conversion	Diff Z Conversion	OFF	←	←	√
		Port1 (bal) Real	100.00 ohm	←	←	√

		Port1 (bal) Imag	0.00 ohm	←	←	√
		Port2 (bal) Real	100.00 ohm	←	←	√
		Port2 (bal) Imag	0.00 ohm	←	←	√
		Port3 (bal) Real	100.00 ohm	←	←	√
		Port3 (bal) Imag	0.00 ohm	←	←	√
	Cmn ZConversion	Cmn Z Conversion	OFF	←	←	√
		Port1 (bal) Real	25.000 ohm	←	←	√
		Port1 (bal) Imag	0.000 ohm	←	←	√
		Port2 (bal) Real	25.000 ohm	←	←	√
		Port2 (bal) Imag	0.000 ohm	←	←	√
		Port3 (bal) Real	25.000 ohm	←	←	√
		Port3 (bal) Imag	0.000 ohm	←	←	√
	De-Embedding S4P	De-Embedding S4P	OFF			
		Topology	A			

Gating	Gating		OFF	←	←	√
	Start		-10.000 ns	←	←	√
	Stop		10.000 ns	←	←	√
	Center		0.0000 s	←	←	√
	Span		20.000 ns	←	←	√
	Type		Bandpass	←	←	√
	Shape		Normal	←	←	√
Transform	Transform		OFF	←	←	√
	Start		-10.000 ns	←	←	√
	Stop		10.000 ns	←	←	√
	Center		0.0000 s	←	←	√
	Span		20.000 ns	←	←	√
	Type		Bandpass	←	←	√
	Window		Normal	←	←	√
		Impulse Width	230.69 ps	←	←	√
		Kaiser Bata	6.0000	←	←	√
Conversion	Conversion		OFF	←	←	√
	Function		Z:Reflection	←	←	√
Limit Test	Limit Test		OFF	←	←	√
	Limit Line		OFF	←	←	√
	Limit Line Offsets	Stimulus Offset	0.0000 Hz	←	←	√
		Amplitude Offset	0.0000 Hz	←	←	√

	Fail Sign	ON	←	←	√
Ripple Limit	Ripple Limit Test	OFF	←	←	√
	Ripple Limit	OFF	←	←	√
	Ripple value	OFF	←	←	√
	Ripple Band	1	←	←	√
	Fail Sign	ON	←	←	√
Bandwidth Limit	BW Test	OFF	←	←	√
	BW Display	OFF	←	←	√
	BW Marker	OFF	←	←	√
	N dB Points	3.0000 dB	←	←	√
	Min Bandwidth	10.000 kHz	←	←	√
	Max Bandwidth	300.00 kHz	←	←	√
	Fail Sign	ON	←	←	√

Avg

Key Operation	Default Value	Preset	*RST	Save/Recall
Avg Factor	16	←	←	√
Averaging	OFF	←	←	√
Avg Trigger	OFF	←	←	√
SMO Aperture	1.5000%	←	←	√
Smoothing	OFF	←	←	√
IF Bandwidth	70 kHz	←	←	√

Cal

Key Operation			Default Value	Preset	*RST	Save/Recall
Correction			OFF	←	←	√
Calibrate	Response (Open)	Select Port	1	←	←	
	Response (Short)	Select Port	1	←	←	
	Response (Thru)	Select Ports	2-1 (S21)	←	←	
	Enhanced Response	Select Ports	2-1 (S21 S11)	←	←	
	1-Port Cal	Select Port	1	←	←	
	2-Port Cal	Select Ports	1-2	←	←	
	3-Port Cal	Select Ports	1-2-3	←	←	
	2-Port TRL Cal	Select Ports	1-2	←	←	
	3-Port TRL Cal	Select Ports	1-2-3	←	←	
	Adapter Removal	Select Ports	1	←	←	
	Adapter Remov	Cal Kit	85033 E			

	al						
ECal	Enhanced Response			2-1 (S21 S11)	←	←	√
	Isolation			OFF	←	←	√
	Characterization			Factory	←	←	√
	Orientation			Auto	←	←	√
	ECal						
Property				OFF	←	←	√
Cal Kit				85033E	←	←	√
Modify Cal Kit	Define STDs			Define STDs	←	←	√
	Specify CLSs	Sub Class		Sub Class 1	←	←	√
	TRL Reflect			Short			
	TRL Option	Impedance		System			
	TRL Option	Reference Plane		Thru			
	Label Kit			85033E			
Port Extensions	Extensions			OFF	←	←	√
	Auto Port Extension	Select Ports		1,2,3,4	←	←	√
		Method		Current Span	←	←	√
		Method	User Span Start	Minimum value in	←	←	√

				frequency range			
		Method	User Span Stop	Maximum value in frequency range	←	←	√
		Adjust Mismatch		OFF	←	←	√
		Include Loss		OFF	←	←	√
	Extension Port1			0.0000 s	←	←	√
	Extension Port2			0.0000 s	←	←	√
	Extension Port3			0.0000 s	←	←	√
	Extension Port4			0.0000 s	←	←	√
	Loss	Select Port		1	←	←	√
		Loss1		OFF	←	←	√
		Loss1		0.0000 dBm	←	←	√
		Freq1		1.0000 GHz	←	←	√
		Loss2		OFF	←	←	√
		Loss2		0.0000 dBm	←	←	√
		Freq2		1.0000 GHz	←	←	√
		Loss at DC		0.0000 dBm	←	←	√
Velocity Factor				1.0000	←	←	√

Set Z0				50.000 ohm	←	←	√
Power Calibration	Select Port			1	←	←	√
	Correction			OFF	←	←	√
	User Sensor			A	←	←	√
	Num of Readings			1	←	←	√
	Tolerance			5 dB	←	←	√
	Loss Compen	Compensation		OFF	←	←	√
	Sensor A Settings	Ref Cal Factor		100.00 %	Non-changing	←	√
		Zero Type		External			
	Sensor B Settings	Ref Cal Factor		100.00 %	Non-changing	←	√
		Zero Type		External			
Receiver Calibration	Select Port			1	←	←	
	Correction			OFF	←	←	√
	Source Port			1	←	←	
	Power Offset			0	←	←	√
	Calibrate Both						
	Calibrate R						
	Calibrate T						
Mixer/Converter Calibration	Scalar Cal (Manual)	Select Ports		2-1 (fwd)	←	←	
	Scalar	Powe	User	A	←	←	

	Cal (Manual)	r Meter	Sensor				
	Scalar Cal (ECal)	Select Ports		2-1 (fwd)	←	←	
	Scalar Cal (ECal)	Power Meter	Use Sensor	A	←	←	
Cal Trig Source				Internal	←	←	√

Center

Key Operation	Default Value	Preset	*RST	Save/Recall
	(Maximum Frequency - 10E5)/2	←	←	√

Display

Key Operation	Default Value	Preset	*RST	Save/Recall
Allocate Channels		←	←	√
Number of Traces	1	←	←	√
Allocate Traces		←	←	√
Display	Data	←	←	√
Data Math	OFF	←	←	√
Title Label	OFF	←	←	√
Graticule Label	ON	←	←	√
Invert Color	OFF	←	←	√
Frequency	ON	←	←	√
Update	ON	←	←	√
Equation	OFF	←	←	√

Format

Key Operation	Default value	Preset	*RST	Save/Recall
	Log Mag	←	←	√

Macro Setup

Key Operation	Default Value	Preset	*RST	Save/Recall
Echo Window	OFF	←	←	√

Marker Fctn

Key Operation		Default Value	Preset	*RST	Save/Recall
Discrete		OFF	←	←	√
Couple		ON	←	←	√
Marker Table		OFF	←	←	√
Statistics		OFF	←	←	√
Annotation Options	Marker Info X Pos	1 %	←	←	√
	Marker Info Y Pos	1 %	←	←	√
	Align	ON	←	←	√
	Active Only	ON	←	←	√

Marker Search

Key Operation			Default Value	Preset	*RST	Save/Recall
Peak			Search Peak			
	Peak Excursion		3.0000 dB	←	←	√
	Peak Polarity		Positive	←	←	√
Target			Search Target	←	←	√
	Target Value		0.0000 dB (When Target is selected)	←	←	√
	Target Transition		Both (When Target is selected)	←	←	√
Tracking			OFF	←	←	√
Search Range	Search Range		OFF	←	←	√
	Start		0.0000 Hz	←	←	√
	Stop		0.0000 Hz	←	←	√
	Couple		ON	←	←	√
	Multiple Range	Multiple Range	OFF	←	←	√
		Target Range	1	←	←	√
		Start	0.0000 Hz	←	←	√

		Stop	0.0000 Hz	←	←	√
Bandwidth			OFF	←	←	√
Bandwidth Value			-3.0000 dB (When one of the marker is on)	←	←	√
Notch			OFF	←	←	√
Notch Value			-3.0000 dB (When one of the marker is on)	←	←	√

Marker

Key Operation	Default Value	Preset	*RST	Save / Recall
Marker1	Marker 1 is turned on immediately after the marker softkey menu is displayed	←	←	√
Ref Marker Mode	OFF	←	←	√

Meas

Key Operation		Default value	Preset	*RST	Save/Recall
		S11	←	←	√
		Sss11 (When Fixture Simulator is set on, set Topology at SE-Bal, and, BalUn on in Analysis mode)	←	←	√
		Sdd11 (When Fixture Simulator is set on, set Topology at Bal-Bal, and BalUn on in Analysis mode)	←	←	√
		Sss11 (When Fixture Simulator is set on, set Topology at SE-SE-Bal, and BalUn on in Analysis mode)	←	←	√
Absolute		A(1)	←	←	√
Aux Input 1	Range	10 V	←	←	√
	Sweep Port	1	←	←	√
Aux Input 2	Range	10 V	←	←	√
	Sweep Port	1	←	←	√

Save/Recall

Key Operation		Default Value	Preset	*RST	Save/Recall
Save Type		State & Cal	←	←	√
Channel/Trace		Disp Only	←	←	√
Save SnP	SnP Format	Auto	←	←	√

Scale

Key Operation	Default value	Preset	*RST	Save/Recall
Divisions	10	←	←	√
Scale/Div	10.000 dB/div	←	←	√
Reference Position	5 Div	←	←	√
Reference Value	0.0000 dB	←	←	√
Electrical Delay	0.0000 s	←	←	√
Phase Offset	0.0000 °	←	←	√

Span

Key Operation	Default Value	Preset	*RST	Save/Recall
	Maximum Frequency - 10E5	←	←	√

Start

Key Operation	Default Value	Preset	*RST	Save/Recall
	100.00 kHz	←	←	√

Stop

Key Operation	Default Value	Preset	*RST	Save/Recall
	Maximum value in frequency range	←	←	√

Sweep Setup

Key Operation			Default Value	Preset	*RST	Save/Recall
Power	Power		0 dBm	←	←	√
	Port Couple		ON	←	←	√
	Port Power	Port 1 Power	0 dBm	←	←	√
		Port 2 Power	0 dBm	←	←	√
		Port 3 Power	0 dBm	←	←	√
		Port 4 Power	0 dBm	←	←	√
	Slope [xx dB/GHz]		0 dB/GHz	←	←	√
	Slope [ON/OFF]		OFF	←	←	√
	CW Freq		100.00 kHz	←	←	√
	RF Out		ON	←	←	√
Sweep Time			AUTO	←	←	√
Sweep Delay			0.0000 s	←	←	√
Sweep Mode			Stepped	←	←	√
Points			201	←	←	√
Sweep Type			Lin Freq	←	←	√
Edit Segment Table	Freq Mode		Start/Stop	←	←	√
	List IFBW		OFF	←	←	√
	List Power		OFF	←	←	√
	List Delay		OFF	←	←	√
	List Sweep Mode		OFF	←	←	√
	List Time		OFF	←	←	√
Segment Display			Order	←	←	√

			Base			
Frequency Offset	Frequency Offset		OFF	←	←	√
	Port 1	Multiplier	1.0000	←	←	√
		Divisor	1.0000	←	←	√
		Offset	0.0000 Hz	←	←	√
		Start	9.00 KHz	←	←	√
		Stop	8.5000 GHz	←	←	√
	Port 2	Multiplier	1.0000	←	←	√
		Divisor	1.0000	←	←	√
		Offset	0.0000 Hz	←	←	√
		Start	9.00 kHz	←	←	√
		Stop	8.5000 GHz	←	←	√
	Port 3	Offset	0.0000 Hz	←	←	√
		Start	9300 kHz	←	←	√
		Stop	8.5000 GHz	←	←	√
	Port 4	Offset	0.0000 Hz	←	←	√
		Start	9.00 kHz	←	←	√
		Stop	8.5000 GHz	←	←	√
	External Source	LO Frequency	OFF	←	←	√
		Multiplier	0.0000	←	←	√
		Divisor	1.0000	←	←	√
		Offset	0.0000 Hz	←	←	√
		Start	0.0000 Hz	←	←	√
		Stop	0.0000 Hz	←	←	√
		Control	OFF	←	←	√

		Power	-10 dBm	←	←	√
		Slope [xxdB/GHz]	0 dB/GHz	←	←	√
		Slope [ON/OFF]	OFF	←	←	√
	X-Axis		Normal (When Frequency Offset is set off.)	←	←	√
	X-Axis		Response (When Frequency Offset is set on.)	←	←	√

System

Key Operation				Default Value	Preset	*RST	Save/Recall
Invert Image				ON	←	←	√
Multiport Test Set Setup	Test Set 1	Select Test Set		E5091_9	←	←	√
		Control		OFF	←	←	√
		Property		OFF	←	←	√
		Port1		A	←	←	√
		Port2		T1	←	←	√
		Port3		R1+	←	←	√
		Port4		R1-	←	←	√
		Port5		X1	←	←	√
		Port6		Y1	←	←	√
		Port7		Z1	←	←	√
		Control Lines	Line 1	LOW	←	←	√
		Control Lines	...	LOW	←	←	√
		Control Lines	Line 8	LOW	←	←	√

	Test Set 2	Select Test Set		E5091_9	←	←	√
		Control		OFF	←	←	√
		Property		OFF	←	←	√
		Port1		A	←	←	√
		Port2		T1	←	←	√
		Port3		R1+	←	←	√
		Port4		R1-	←	←	√
		Port5		X1	←	←	√
		Port6		Y1	←	←	√
		Port7		Z1	←	←	√
		Control Lines	Line 1	LOW	←	←	√
		Control Lines	...	LOW	←	←	√
		Control Lines	Line 8	LOW	←	←	√
Misc Setup	Beeper	Beep Complete		ON	←	←	√
		Beep Warning		ON	←	←	√
	GPIB Set	Talker/Listener Address		17	Non-changing	←	

	up	Power Meter Address			13	Non-changing	←	
		Signal Generator Address	Address		19	←	←	√
			Switching Time		100.00 ms	←	←	√
			Custom Commands	Preset	""	←	←	√
				Turn RF Out ON	"R3"	←	←	√
				Set Frequency	"FR %f% HZ"	←	←	√
				Set Power Level	"AP %p% DM"	←	←	√
	Power Meter Set up	Select Type			GPIB	Non-changing	←	
		GPIB Address			13	Non-changing	←	
		USB						
	Network Set up	Telnet Server			OFF	Non-changing	←	
		SICL-LAN Server			OFF	Non-changing	←	
		SICN-LAN Address			17	Non-changing	←	

		Web Server			OFF	Non-changing	←	
		MAC Address			-	Non-changing	←	
	Clock Setup	Show Clock			ON	←	←	√
	Key Lock	Front Panel & Keyboard Lock			OFF	←	←	
		Touch Screen & Mouse Lock			OFF	←	←	
	Display Setup	Color Setup	Normal	Data Trace 1	Red:5 Green:5 Blue:0	←	←	√
				Data Trace 2	Red:0 Green:5 Blue:5	←	←	√
				Data Trace 3	Red:5 Green:0 Blue:5	←	←	√
				Data Trace 4	Red:0 Green:5 Blue:0	←	←	√
				Data Trace 5	Red:5 Green:4 Blue	←	←	√

					:0			
				Data Trace 6	Red:5 Green:3 Blue:3	←	←	√
				Data Trace 7	Red:3 Green:4 Blue:5	←	←	√
				Data Trace 8	Red:5 Green:4 Blue:4	←	←	√
				Data Trace 9	Red:3 Green:4 Blue:3	←	←	√
				Mem Trace 1	Red:3 Green:3 Blue:0	←	←	√
				Mem Trace 2	Red:0 Green:3 Blue:3	←	←	√
				Mem Trace 3	Red:3 Green:0 Blue:3	←	←	√

				Mem Trace 4	Red:0 Green:3 Blue:0	←	←	√
				Mem Trace 5	Red:3 Green:2 Blue:0	←	←	√
				Mem Trace 6	Red:3 Green:1 Blue:1	←	←	√
				Mem Trace 7	Red:1 Green:2 Blue:3	←	←	√
				Mem Trace 8	Red:3 Green:2 Blue:2	←	←	√
				Mem Trace 9	Red:1 Green:2 Blue:1	←	←	√
				Graticule Main	Red:3 Green:3 Blue:3	←	←	√
				Graticule Sub	Red:1 Green:1	←	←	√

					Blue :1			
				Limit Fail	Red: 5 Green:0 Blue :0	←	←	√
				Limit Line	Red: 3 Green:0 Blue :0	←	←	√
				Background	Red: 0 Green:0 Blue :0	←	←	√
		Magnification			Normal			
	Channel/Trace Setup				9 Channels 9 Traces	Non-changing	←	
	Preset Setup	State			Factory	Non-changing	←	
		Confirm			ON	Non-changing	←	
Backlight					ON	←	←	
Service Menu	System Correction				ON	←	←	√
	External Test Set				OFF	←	←	√
	Init Src Ctrl				ON	←	←	√
	Init Src Port				1	←	←	√

	Security Level	None	←	←	√

Trigger

Key Operation	Default	Preset	*RST	Save/Recall
Continuous	Continuous (Ch1) Hold (Other Ch)	←	Hold (all channels)	√
Trigger Source	Internal	←	←	√
Trigger Event	On Sweep	←	←	√
Ext Trig Input	Negative Edge	←	←	√
Ext Trig Delay	0.0000 s	←	←	√
Low Latency	OFF	←	←	√
Ext Trig Output	OFF	←	←	√
Polarity	Positive Pulse	←	←	√
Position	After Point	←	←	√
Trigger Scope	All Channel	←	←	√

Troubleshooting

Troubleshooting

Troubleshooting

This section describes the steps you should take when you believe the Agilent E5071C is operating improperly. The results of these simple investigative procedures may help you avoid the down-time and inconvenience of repair service. The troubleshooting instructions are divided into three categories.

When taking all solution but it does not work. Contact Agilent Technology's Customer Contact.

- Troubleshooting during Startup
- Troubleshooting during Operation
- Troubleshooting for External Devices

Troubleshooting during Startup

Symptom	Solution
Turning on (|) the standby switch does not start up the system.	• Confirm that the power cable is properly plugged in. • Confirm that the line switch on the rear panel is turned on.
The system starts up, but it automatically shuts down immediately.	Execute the system recovery.
The system starts up, but it enters the service mode (The instrument status bar in the lower right part of the screen displays SVC in red).	Execute the system recovery.
The measurement screen appears after startup, but the date and time displayed on the instrument status bar in the lower right part of the screen differ greatly from the previous settings.	Execute the system recovery.
The measurement screen appears after startup, but the power-on test is failed, with Error Message 241 appearing against a red background in the instrument	Execute the system recovery.

message/warning area in the lower left part of the screen.	

Troubleshooting during Operation

Symptom	Solution
The Error Message "Port N receiver overload" (N denotes a port number) is Displayed. During the measurement of an amplifier, Error Messages 221 through 224 "Port N receiver overload" (N denotes a port number) are displayed. This error occurs when the input to a test port exceeds the maximum input level in the measurement of an amplifier. The measurement value obtained in such a case is not correct. In the worst case, a failure (damage to the receiver) may occur.	• Disconnect any cables and adapter on test ports. • Change the measurement condition so that the input to the test port does not exceed the maximum input level. • When you measure the 3rd harmonics over 8.5 GHz with options 2D5, 4D5, 2K5 and 4K5, it is recommended to suppress the carrier signal using filter etc before executing the measurement for accurate results.
A Clearly Abnormal Measurement Value The measurement value is not reproducible, or clearly abnormal.	• Confirm that the DUT, connection cables, and other parts are connected correctly. • Confirm that the connectors and cables used to connect the DUT are free from damage and poor contact. • Confirm that the calibration has been executed correctly. If you have not acquired a correct error correction factor, you cannot obtain a correct measurement value. • Confirm that the calibration kit was selected correctly. • Confirm that the calibration kit is defined correctly. • Confirm that if the stimulus

	signal output is turned on.
The System Cannot be Operated Manually (Front Panel Keys, Keyboard, Touch Screen and Mouse) The keyboard or mouse becomes inoperable.	• Confirm that the keyboard or mouse is connected correctly. When it is connected correctly, turn off the power once, and restart the system.
The front panel key or keyboard becomes inoperable.	• Using the mouse, turn **System > Key Lock > Front Panel & Keyboard Lock OFF.**
The touch screen becomes inoperable.	• Using the front panel keys, press **System** > **Key Lock** > **Touch Screen & Mouse Lock** OFF to turn OFF the lock. • Execute the calibration of the touch screen. • After the System Recovery, touch screen may be inoperable. See the Reconfiguring Touch Panel Setting in the installation guide.
The mouse becomes inoperable.	• Using the front panel keys, press **System** > **Key Lock** > **Touch Screen & Mouse Lock** to turn OFF the lock.
All of the front panel keys, keyboard, and mouse become inoperable.	• Confirm that the keyboard or mouse is connected correctly. When it is connected correctly, turn off the power once, and restart the system.
The keyboard and mouse have been connected after power-on.	• Turn off the power once, and restart the system. When taking all these measures does not recover operability, there is a possibility of a failure.

The Screen Freezes and All Operations Become Impossible. The measurement in progress or screen update is stalled and all of the front panel keys, keyboard, mouse, and touch screen are inoperable.	• Press the standby switch to turn off the power once, and restart the system.
The System Freezes while in Operation.	• Press the standby switch to turn off the power once, and restart the system.
The Rear Cooling Fan Does Not Operate.	• There is a possibility of a failure.
The Sweep Action Stops during Measurement or Is Not Executed. An Error or Warning Message Appears.	• There is a possibility of a failure.
An error or warning message is displayed on the instrument message/warning area in the lower part of the screen	• Refer to Error Messages and Warning Messages.

Troubleshooting for External Devices

Symptom	Solution
Cannot Output to a Printer Cannot output a measurement screen or data to a printer. Attempting to output to a printer causes Error Messages 120 and 121 to appear.	• Confirm that the power to the printer is on and that the line cable is connected correctly. • Confirm that the connector cable of the printer is connected correctly. • Confirm that the printer is online. • Confirm that the printer has not run out of paper. • Confirm that the printer has not run out of ink.
Does Not Respond to	• Confirm that the GPIB address is

an External Controller/Fails to Function Normally A GPIB device does not respond to the external controller, or fails to function normally.	defined correctly. • Confirm that the GPIB cable is connected. • Confirm that another instrument connected by the GPIB cable has the same GPIB address. • Confirm that the GPIB cable connection forms a loop.
An ECal module can not be detected.	When you install the firmware A.09.31 and below in the HDD revision CN960 and above, this symptom may occur. Follow the following steps. 1. Disconnect all ECal module from E5071C. 2. Execute the system recovery. 3. Delete two files, AgtEcal.inf and AgtEcal.PNF in C:\WINDOWS\PNPDRVRS\AGILENT. 4. 5. Put this file into C:\WINDOWS\PNPDRVRS\AGILENT. 6. Reinstall the firmware. 7. Install the Ecal Driver.
The E5071C hung up when two (or more) ECals are connected/disconnected at the same time.	When you install the firmware A.09.60 and above in the HDD revision CN954 and below, this symptom may occur. Follow the following steps. 1. Disconnect all ECal module from E5071C. 2. Execute the system recovery. 3. Delete two files, USBECAL.inf and USBECAL.PNF in C:\WINDOWS\PNPDRVRS\AGILENT.

	5. Put these two files into C:\WINDOWS\PNPDRVRS\AGILENT. 6. Reinstall the firmware. 7. Install the Ecal Driver.

Maintenance

Maintenance

- Backing Up the Data
- Cautions Applicable to Requesting Repair, Replacement, Regular Calibration, etc.
- Cleaning this Instrument
- Removing Log Data
- Removing/Mounting Removable Hard Disk
- Replacement of Parts with Limited Service Life
- System Recovery
- Service Menu
- Updating Firmware
- System Verification

Backing Up the Data

Be sure to back up regularly your important data (including program) files in this instrument to a CD-R or other backup medium. Agilent Technologies shall not be liable for any data damages caused by troubles of this instrument.

Making Backup Files

You can make backup files on the hard disk of an external PC using following methods.

You can access to drive D: of the E5071C from an external PC via LAN, and copy your important data files on the drive D: to the external PC. See Accessing Hard Disk of E5071C from External PC for details.

You can transfer your important data files on the drive D: of the E5071C to the external PC using :MMEM:TRAN command via GPIB.

- Do not modify any files and folders in drives other than drive D:. Doing so will cause malfunctions.

Other topics about Maintenance

Cautions Applicable to Requesting Repair, Replacement, Regular Calibration, etc.

- Backing Up Data in the Hard Disk
- Devices to be Sent Back for Repair or Regular Calibration

Other topics about Maintenance

Backing Up Data in the Hard Disk

The user is requested to back up the stored programs and data into external media by using the instrument's storing function before requesting the Company's Service Center to repair the instrument or replace hard disks.

See Making Backup Files for how to make backup files.

Please take note that the Company will not be held liable to any extent for potential erasure or change of stored programs or data due to the repair or replacement of hard disks performed by the Company. When a hard disk itself fails, the programs and data stored in it cannot be recovered.

Devices to be Sent Back for Repair or Regular Calibration

If it is necessary to send the unit to the Service Center of Agilent Technologies for repair or regular calibration, please follow the instructions below.

Equipment to be Sent

When requesting repair or regular calibration of the unit by our Service Center, send only the E5071C main unit without any installed option you may have ordered. Unless specifically instructed, it is not necessary to send accessories and calibration kits.

Packing

Use the original package and shock absorbers, or equivalent anti-static packing materials, when sending the unit.

Shipping Address

For the location of the nearest Agilent Technologies Service Center, contact the Customer Contact.

Recommended Calibration Period

The recommended calibration period for this instrument is one year. The user is recommended to request the Company's Service Center to perform regular calibration every year.

Product Warranty Period

The actual warranty of the instrument depends on the date it was ordered and whether any warranty options were purchased or not at that time. To determine the exact warranty period on your instrument, contact Agilent Technologies with the model and serial number of the instrument.

Cleaning this Instrument

- Cleaning an LCD
- Maintenance of Test Ports and Other Connectors/Ports
- Cleaning a Display Other than an LCD

Other topics about Maintenance

This section describes how to clean the instrument.

To protect yourself from electrical shock, be sure to unplug the power cable from the outlet before cleaning the instrument.

Never clean the internal components of the instrument.

Cleaning an LCD

Use one of the following methods to clean the display surface regularly.

- For normal cleaning, rub the surface gently with a dry, soft cloth.
- When stains are difficult to remove, gently wipe the surface with cloth damped with a small amount of ethanol or isopropyl alcohol. You can clean the standard type LCD (no touch screen function) with a cloth dipped in water and then wrung tightly.
    - Do not use chemicals other than ethanol and isopropyl alcohol to wet the cleaning cloth. To clean the touch screen LCD, do not wet the cloth with water.

Maintenance of Test Ports and Other Connectors/Ports

Test ports on the front panel of the E5071C are fitted with N Type connectors (f). Stains or other damage to these connectors would significantly affect the accuracy in measurements in the RF range. Always pay attention to the following precautions.

- Always keep the connectors free from stains and dust.
- Do not touch the contact surface on the connectors.
- Do not plug damaged or scratched connectors into the test ports.
- Use compressed air for cleaning connectors. Do not use abrasives under any circumstance.

The above precautions must also be observed in maintaining connectors and ports other than these test ports.

Cleaning Parts Other than the LCD, Test Ports, and Other Connectors/Ports

To remove stains on parts other than the LCD, test ports, and other connectors/ports of the instrument, wipe them gently with a soft cloth that is dry or wetted with a small amount of water and wrung tightly.

Removing Log Data

The E5071C creates automatic log of data for troubleshooting purpose. For security reasons, if this data needs to be deleted, then SCPI.SERVice.LOGGing.CLEar command can be used to clear the log recorded by the E5071C.

The log file stores data related to:

- Power ON time
- Number of times of power ON
- Result of power ON test
- Number of switching time of Step Attenuator
- Number of time of overload
- Number of aborted time
- Event Log

Other topics about Maintenance

Removing/Mounting Removable Hard Disk (Option 017 User Only)

- Overview
- Backing_Up_Retributing_Option_Data
- Removing Removable Hard Disk (Type I)
- Mounting Removable Hard Disk (Type I)
- Removing Removable Hard Disk (Type II)
- Mounting Removable Hard Disk (Type II)
- Writing System Correction Data from Backup Memory to Removable Hard Disk

Other topics about Maintenance

Overview

This section provides information on how to mount and remove a removable hard disk as well as write the system correction data.

For calibration or repair of the E5071C, send it with the removable hard disk mounted.

If you replace the removable hard disk, make sure to retrieve the system correction data and option data after mounting the hard disk

Backing Up/Retributing Option Data

1. Back up the license key file for software option before removing hard disk.
2. Place the license key file after mounting the hard disk.

Removing Removable Hard Disk (Type I)

Follow the steps below to remove a removable hard disk.

1. Turn off the power.
2. Insert the key (furnished, Agilent Part Number 0960-2562) in the lock and turn clockwise.
3. Press the button under the lock and remove the hard disk. It may be too rigid to remove by fingers. In that case, use some tool such as a

flat-blade screw driver to remove it.

e5071c190

4. Do not attempt to unlock the key and remove the hard disk when the power is on. Doing so will turn off the power automatically.

- Be careful not to be hurt your hands while removing the hard disk.

Mounting Removable Hard Disk (Type I)

Follow the steps below to mount a removable hard disk.

1. Insert the hard disk into the slot.
2. Turn the key to the left until it is locked.
3. Remove the key.
4. Turn on the E5071C.
5. The "U" is displayed at the 8 segment LED on next to the removal hard disk when E5071C is turned on without key lock.

- The key is available to remove even in Unlocked state. Before power on, confirm if Removable hard disk is locked completely.

Removing Removable Hard Disk (Type II)

Follow the steps below to remove a removable hard disk.

1. Turn off the power.
2. Insert the key (furnished, Agilent Part Number 5188-4426) in the lock and turn anti-clockwise.

ssa0153

3. Open the screw in clockwise direction.

ssa0154

4. Press the button under the screw.

ssa0155

5. Remove the hard disk.

ssa0156

6. Do not attempt to unlock the key and remove the hard disk when the power is on. Doing so will turn off the power automatically.

Mounting Removable Hard Disk (Type II)

Follow the steps below to mount a removable hard disk.

1. Insert the hard disk into the slot.
2. Turn the screw in anti-clockwise direction until it is locked.
3. Turn the key to the right until it is locked.
4. Remove the key.
5. Turn on the E5071C.
6. The key is available to remove even in Unlocked state. Before power on, confirm if Removable hard disk is locked completely.

Writing System Correction Data from Backup Memory to Removable Hard Disk

The following procedure shows how to write the system correction data stored in the backup memory to the newly mounted removable hard disk.

1. Turn on the E5071C.
2. Press **Macro Setup** key.
3. Click **Load Project...** in the softkey menu.
4. Select **RestoreSysCorFile.vba** from the **D:\Agilent\Service** folder and then press the **Open** button.
5. Press **Macro Run** key
6. The RestoreSysCorFile dialog box appears.
7. Click **OK**.
8. Once the program has terminated, restart the E5071C to enable the system correction data.

Replacement of Parts with Limited Service Life

This instrument incorporates parts with limited service life as shown in the following table. Using the recommended replacement time as a guide, request the Agilent Service Center to replace these parts. However, a part may need to be replaced at an earlier time than that listed in the table, depending on such conditions as location, frequency of use, and where it is stored.

- Each service life and recommended replacement time listed below is for reference only and does not imply a guarantee of the part's service life.

Part Name	Service Life (Parts supplier reference value)	Recommended replacement time
Hard Disk Drive (HDD)	5 years or 20,000 operating hours, whichever comes earlier. Exchanging hard disk drives causes the contents written after shipment from the factory (LAN setup, etc.)to be initialized to the state at the time of shipment. The programs and data stored in Drive D (user directory) are erased.	3 years
Main fan	50,000 operating hours. The service life may be significantly shorter when used in a dusty and dirty environment.	5 years
CPU fan		
Battery on Mother board	50,000 operating hours. The service life may be shorter if E5071C power has not been turned on for long time.	5 years
Power supply	50,000 operating hours (Depends on the service life of the power supply cooling fun) The service life may be significantly shorter when used in a dusty and dirty environment.	5 years
LCD screen backlight	50,000 operating hours. When the unit is used for automatic	5 years

	measurements in a production line and the on-screen information is not required, the life of the LCD backlight can be saved by turning it off. As for the method of turning the backlight off, refer to Turning off the LCD Screen Backlight.	
Touch screen (function)	One million times (dotting life)	5 years
USB receptacle	1,500 cycles insertion/extraction. The service life may be shorter when used in a dusty and dirty environment. In case that the insertion/extraction is in heavy usage such as Ecal in the production line, using USB extension cable may save the USB receptacle life.	N/A

Other topics about Maintenance

System Recovery

By executing system recovery, you can return the Windows operating system of the E5071C to the factory state or the user state at the setting the user performed save user state.

NOTE User state recovery function is available at HDD revision CN924 and below.

The procedure of system recovery is described in both Installation Guide and Service Guide.

Other topics about Maintenance

Updating Firmware

- Overview
- Procedure

Other topics about Maintenance

Overview

Users can update E5071C firmware by themselves.

Updating firmware does not include the following software update:

- Windows Operating System
- VBA (Visual Basic for Application)
- Driver for Windows
- Calibration Constant Data

When E5071C firmware is updated from version A.08.12 to A.09.10, the update process may interrupt, and all of update actions may stop without generating any warning message. When such case happens, reboot the Windows. After rebooting, firmware update process will start again.

Procedure

NOTE If you have E5071C-TDR option installed, turn OFF this option before updating the firmware: **Analysis** > **TDR** (**OFF**).

1.
2. Download the execution file (**E5071C_xx.xx.exe**).
3. Run the **E5071C_xx.xx.exe** to extract the **E5071C.msi**.
4. If you use your local PC to download, save the file to a USB mass storage device in order to move it to E5071C. Connect the USB mass storage device into the front USB port of the E5071C.
5. Copy or move the **E5071C.msi** file to D drive.
6. Press **System** > **Service Menu** > **Update Firmware**. Then, the password dialog box appears.
7. Type **e507xa** as the password, then click **OK**.
8. In **Open** dialog box, select **E5071C.msi**, then press **Open**. Then the windows installer appears.
9. Follow the instruction of windows installer.
10. After the installation is finished, the firmware restarts.

11. Press **System** > **Firmware Revision** to confirm the firmware revision you have just installed.

Installing E5071C-TDR Option

1.

2. Download the execution file (**E5071C_opt_TDR_xx.xx.exe**).
3. Ensure that you have updated the E5071C firmware.
4. Press **System** > **Service Menu** > **EXIT** to exit the E5071C measurement application.
5. Run the **E5071C_opt_TDR_xx.xx.exe** to extract the **E5071C_opt_TDR.msi**.
6. If you use your local PC to download, save the file to a USB mass storage device in order to move it to E5071C. Connect the USB mass storage device into the front USB port of the E5071C.
7. Copy or move the **E5071C_opt_TDR.msi** file to D drive.
8. Double-click **E5071C_opt_TDR.msi**.
9. If you are installing for the first time, select **Install Option TDR on E5071C Instrument (default)**.
10. Follow the instruction of window installer.
11. After the installation is finished, double-click the Network Analyzer icon on the desk top to restart the E5071C firmware.
12. Click **About E5071C-TDR** button in E5071C-TDR option tool bar to confirm the option revision you have just installed.

Service Functions

This menu (**System** > **Service Menu** > **Test** and **Service Functions**) provides information related to various test related to the E5071C.

Under service menu, the following topics are described in this help.

Other topics about Maintenance

System Verification

The performance of the network analyzer is specified in two ways: system specifications, and instrument specifications. It is the end users' responsibility to determine which set of specifications is applicable to their use of the E5071C.

A network analyzer measurement "system" includes the analyzer, calibration kit, test cables, and any necessary adapters. The system verification software in the E5071C is used to verify the system's conformance to the "system" specifications. A "pass" result demonstrates that the analyzer, test cables, and adapters, perform correctly as a system. It DOES NOT demonstrate that any one component performs according to its individual specifications. A change to any part of this measurement system requires a re-verification of the system.

Instrument specifications specify the network analyzer's uncorrected measurement port characteristics and its output and input behavior. The E5071C performance tests are used to verify the analyzer's conformance to "instrument" specifications.

The System Verification utility verifies the E5071C's system specifications by automatically measuring the magnitude and phase for all four S-parameters for each verification device, and comparing the values against the following:

- Factory measured data from files on the verification disk
- Limit lines based on the measurement uncertainty

System Verification requires the use of a Calibration Kit and Verification Kit which has been certified within the past 12 months by Agilent. System Verification can NOT be used to perform this kit certification.

- Equipment Used in the System Verification
- Precautions for Handling Airlines
- Flow Diagram of Procedure
- System Verification Program
- Procedure for System Verification
- If the System Fails the Verification Test
- System Verification Results

Other topics about Maintenance

NOTE Although the performance for all S-parameters is measured, the S-parameter phase uncertainties for reflection

measurements are less important for verifying system performance. Therefore, the limit lines will not appear on the printouts.

Equipment Used in the System Verification

Supported Verification Kit:

- 85053B, 3.5 mm, 300 kHz - 26.5 GHz
- 85055A, Type-N, 300 kHz - 18.0 GHz
- 85051B, 7 mm, 300 kHz - 18 GHz

The combination of Calibration Kit and Verification Kit

The combination of Calibration Kit and Verification Kit that can be used is as follows:

Cal kit	Connector	Kit Type	Verification Kit	Minimum Frequency	Maximum Frequency
85033D	3.5 mm	Mechanical	85053B	-	6 GHz
85033E	3.5 mm	Mechanical	85053B	-	9 GHz
85052D	3.5 mm	Mechanical	85053B	-	26.5 GHz
85052C	3.5 mm	Mechanical	85053B	-	26.5 GHz
85093B	3.5 mm	ECal**	85053B	300 kHz	9 GHz
85093C	3.5 mm	ECal**	85053B	300 kHz	9 GHz
N4433A	3.5 mm	ECal**	85053B	300 kHz	20 GHz
N4691A	3.5 mm	ECal**	85053B	10 MHz	26.5 GHz
N4691	3.5 mm	ECal**	85053B	300 kHz	26.6

B					GHz
85032 B/E	Type-N	Mechanical	85055A	-	6 GHz
85032F	Type-N	Mechanical	85055A	-	9 GHz
85054 D	Type-N	Mechanical	85055A	-	18 GHz
85092 B	Type-N	ECal**	85055A	300 kHz	9 GHz
85092 C	Type-N	ECal**	85055A	300 kHz	9 GHz
N4432 A	Type-N	ECal**	85055A	300 kHz	18 GHz
N4690 A	Type-N	ECal**	85055A	10 MHz	18 GHz
N4690 B	Type-N	ECal**	85055A	300 kHz	18 GHz
85031 B	APC-7mm	Mechanical	85051B	-	6 GHz
85050 C	APC-7mm	Mechanical	85051B	-	18 GHz
85050 D	APC-7mm	Mechanical	85051B	-	18 GHz
85091 B	APC-7mm	ECal**	85051B	300 kHz	9 GHz
85091 C	APC-7mm	ECal**	85051B	300 kHz	9 GHz
N4696 A	APC-7mm	ECal**	85051B	10 MHz	18 GHz
N4696	APC-	ECal**	85051B	300 kHz	18 GHz

B	7mm				
N4431 A*	3.5 mm	ECal**	85053B	300 kHz	9 GHz
N4431 B*	3.5 mm	ECal**	85053B	9 kHz	13.5 GHz
N4431 A*	Type-N	ECal**	85055A	300 kHz	9 GHz
N4431 B*	Type-N	ECal**	85055A	9 kHz	13.5 GHz

NOTE The results from the verification kit other than following combination are not warranted specification:

E5071C 3GHz, 4.5GHz, 6.5GHz, 8.5GHz (Option 230/235/240/245/260/265/280/285/430/435/440/445/460/465/480/485)
Calibration Kit: 85032F, 85033E Electronic calibration (ECal) module: 85092C, 85093C

E5071C 14GHz, 20GHz (Option 2D5/2K5/4D5/4K5)

Calibration Kit: 85052D

NOTE * The connection of the N4431x is for the thru path A-B, C-D, A-D, and B-C.

NOTE ** The supported connector type is "male-female".

Cable Substitution

The test port cables specified for the E5071C have been characterized for connector repeatability, magnitude and phase stability with flexing, return loss, insertion loss, and aging rate. Since test port cable performance is a significant contributor to the system performance, cables of lower performance will increase the uncertainty of your measurement. It is highly recommended that the test port cables be regularly tested.

If the system verification is performed with a non-Agilent cable, ensure that the cable meets or exceeds the operation of the specified cable.

Cable Flex Factor

Flex Factor determines how much of the cable phase uncertainty to include in determining the limit lines.

Set to 0% (zero) if the cables are held down in a fixture and are not allowed to move during the calibration and verification.

Set to 100% if the cables are allowed to move a lot.

Calibration Kit Substitution

Non-Agilent calibration kits are not recommended nor supported.

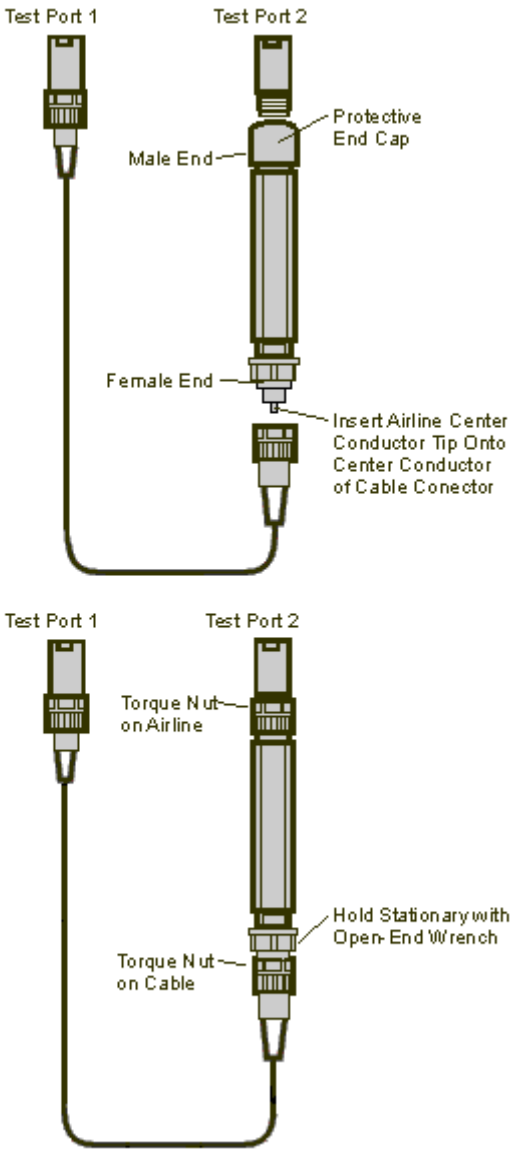
Precautions for Handling Airlines

When you are using the airlines in the Verification Kit, observe the following practices to ensure good measurement techniques.

Be very careful not to drop the airline's center or outer conductor. Damage will result if these devices are dropped.

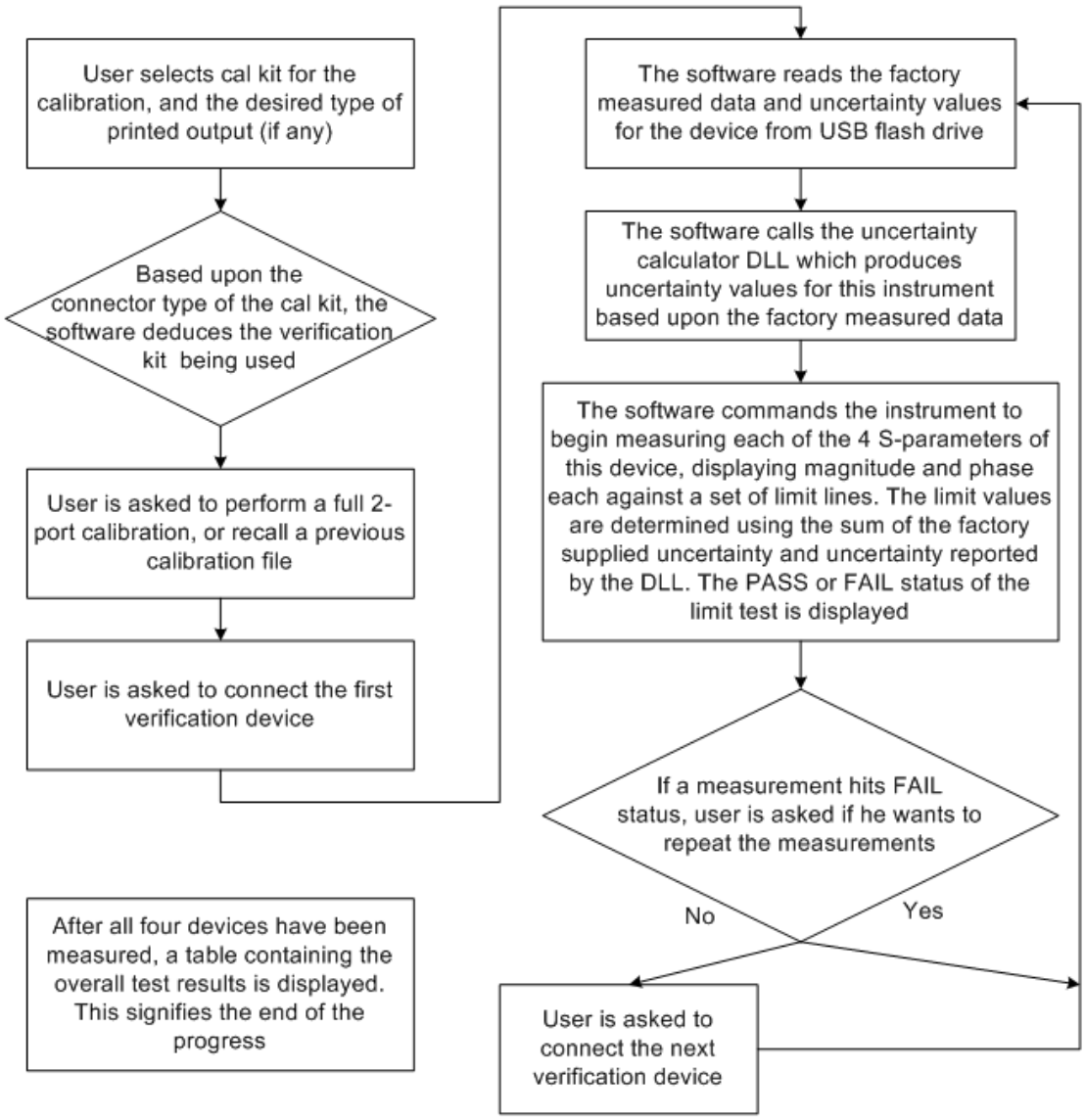
Use proper Electro-Static Discharge (ESD) procedures.

Clean your hands or wear gloves as skin oils will cause a change in electrical performance.



Flow Diagram of Procedure

The operational flow of the software is depicted by the flowchart shown below.

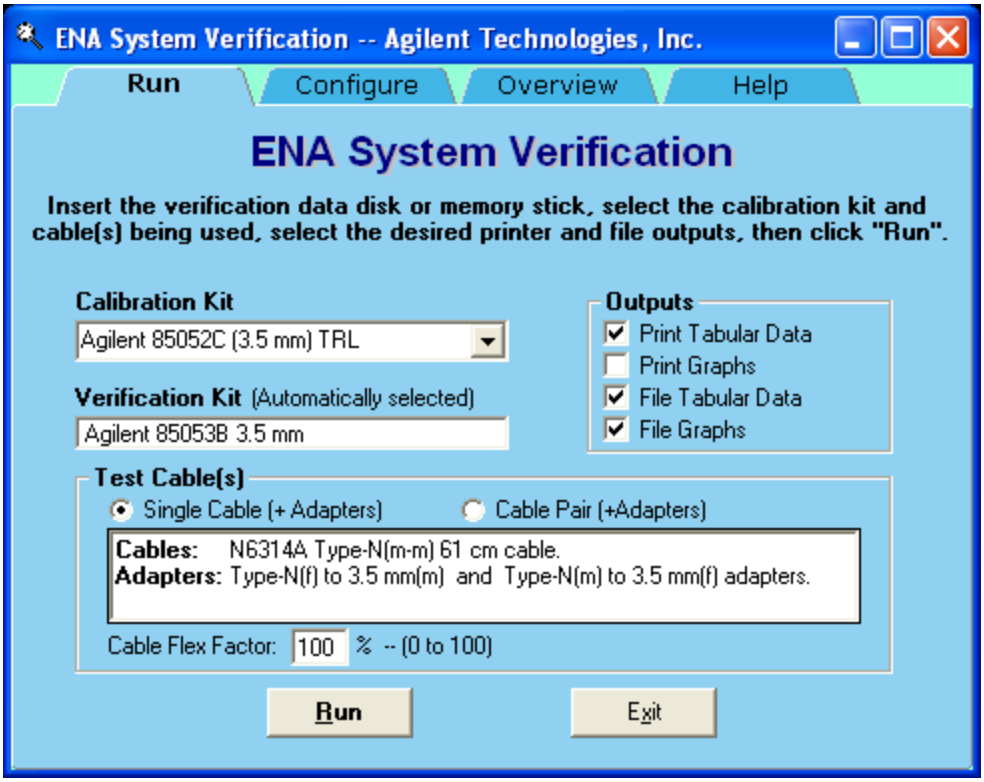


System Verification Program

There are four tabs in the E5071C System Verification program:

- Run
- Configure
- Overview
- Help

Run



1. **Calibration Kit**: Select the calibration kit or ECal module that is being used.
2. **Verification Kit**: The corresponding Verification Kit to use appears in the **Verification Kit** box. Refer to the combination of Calibration Kit and Verification Kit table.
3. **Test Cable(s)**: Select the method of connecting ports and Calibration/Verification Kit of E5071C. It is necessary to connect Calibration Kit and Verification Kit to E5071C with the same cable connection.

When calibration or verification is measured (2 port measurement), E5071C ports are connected to ports of kit by the following options:

- **Single Cable**: One side of the port is connected to the port of E5071C and kit directly. The other port is connected through Cable.
- **Cable Pair**: Both ports are connected to both ports of E5071C and kit through Cable.
- **Cable Flex Factor**: Flex Factor determines how much of the cable phase uncertainty should be included in determining the limit lines.

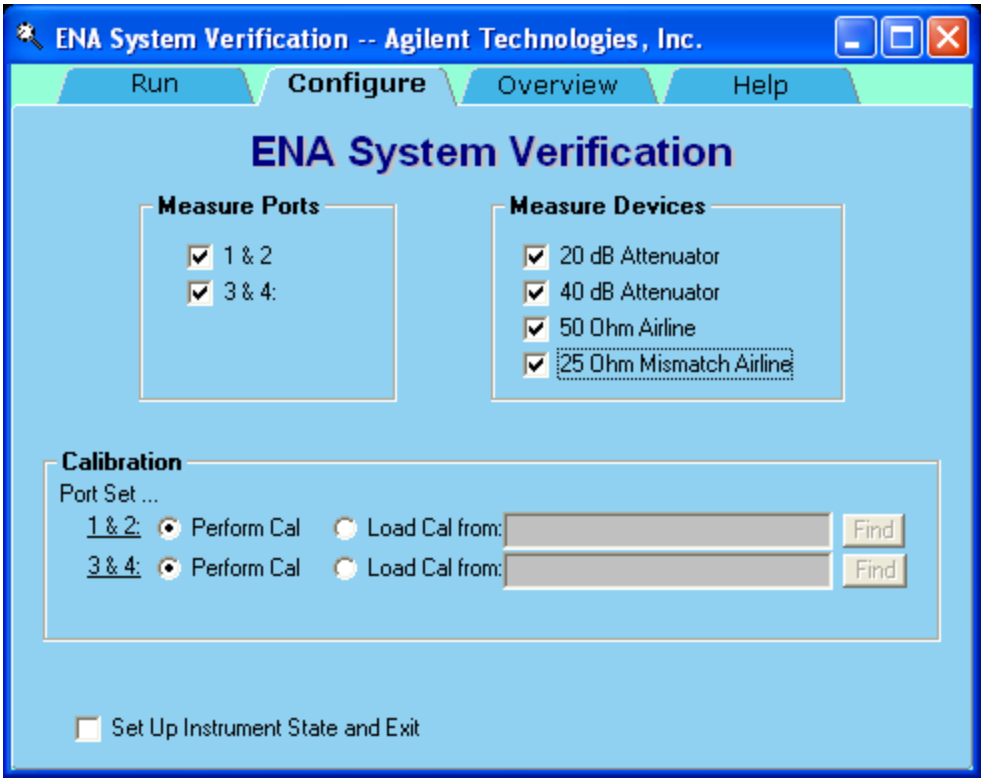
- Set to 0% (zero) if the cables are held down in a fixture and are not allowed to move during the calibration and verification.
- Set to 100% if the cables are allowed to move extensively.

4. **Outputs**: Select the Verification result output format from one of the following options.
   - **Print Tabular Data**: Prints the Verification data in tabular form which includes measured data and uncertainty limits.
   - **Print Graphs**: Prints the Verification data in graphical form. It is the screen image of the Verification result of a measurement is printed as output. The graphic form includes the measured data trace, factory supplied data trace and uncertainty limits.

     NOTE When you select Print Graphs, proceed to the next step once the printing is completed. Otherwise, printer error will occur.
   - **File Tabular Data**: Writes the Verification data in tabular form to a text file. The file is stored as SysVer(Date)(Time)(Ports).txt and saved under the directory of "D:\SysVer", where Date=yyyymmdd (ex. 20101016), Time=hhmm (ex. 1520), Ports=Pnm (ex. P12). An example of a filename is SysVer201010161520P12.txt.
   - **File Graphs**: Saves a screen image in .PNG format. The file is stored as SysVer(Date)(Time)(Ports)(DevNo)(Param).png and saved under the directory of "D:\SysVer", where DevNo=Device Number (ex. Dev1), Param=M or P (M is for LogMag or LinMag, P is for Phase). An example of a filename is SysVer201010161520P12Dev1M.png.
5. Click **Run** button to start the Verification procedure. Follow the instructions on the E5071C for performing the System Verification, inserting the Verification devices as prompted.
6. Click **Exit** button to exit the Verification program.

   NOTE For the System Verification to be truly adequate, the software must measure all devices in the kit with a recent calibration applied. Removing and reattaching any test port cables or adapters invalidates all previous calibrations.

Configure



1. **Measure Ports**: The port pair that executes verification measurement is selected.
   - **1 & 2** : Port 1 and Port 2
   - **3 & 4** : Port 3 and Port 4 (This option is not displayed on 2 port E5071C model)

   In 4 ports E5071C model, both options are selected as default. At least one option should be selected.

   In 2 ports E5071C model, option *1 & 2* is selected and it cannot be changed.
2. **Measure Devices**: Select the device in the verification kit to execute the measurement from the following options. By default, all options are selected and at least one option must be selected. In general, all items should be measured. When you disable all items, *20 dB Attenuator* is automatically selected:
   - **20 dB Attenuator**
   - **40 dB Attenuator**
   - **50 Ohm Airline**
   - **25 Ohm Mismatch Airline**
3. **Calibration**: Select the method of acquiring the calibration data in the Verification procedure from the following options.

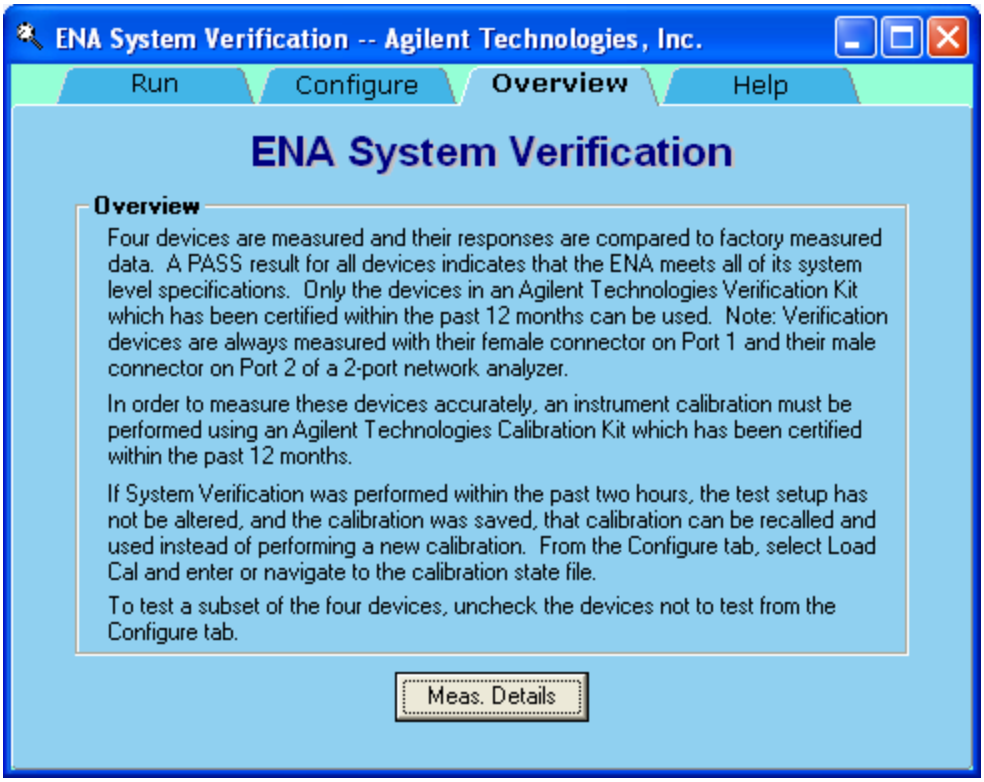
- **Perform Cal**: Calibration data is acquired from the actual Calibration measurement.
- **Load Cal from**: Calibration data is recalled from the State file. The State file should include the full 2 port calibration for the corresponding ports.

The radio button for unselected options are disabled.

In 2 ports E5071C model, the option and entry for Port *3 & 4* is not displayed.

4. **Set Up Instrument State and Exit:** If this check box is checked, only set up is performed and the program is terminated thereafter without executing the verification process. If this check box is un-checked, the verification process is executed once the set up is performed.

Overview



1. This tab displays the overview of the ENA System Verification. Click **Meas. Details** to view the Verification measurement details. Click Meas. Overview to toggle back to ENA System Verification overview.

Help

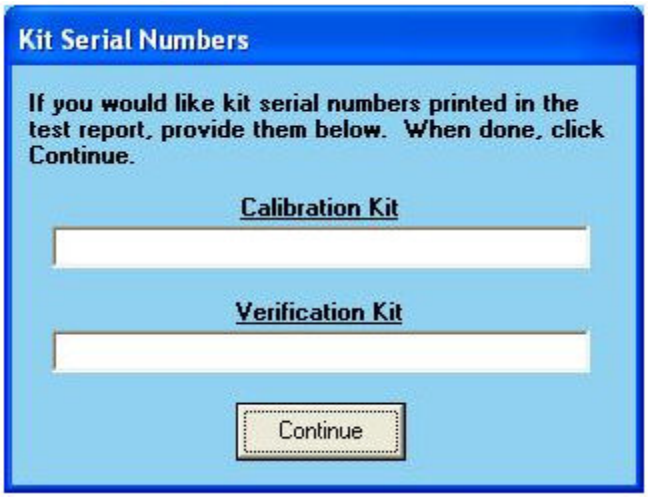
1. This tab displays the information of the ENA System Verification and links to online help topics.

Procedure for System Verification

1. If you want printed test outputs, connect a printer to the E5071C.
2. Let the E5071C warm up for at least 90 minutes.
3. Insert the Verification Kit USB flash drive into the E5071C's USB port.
4. On the **System** menu, point to **Service Menu** > **Test Menu** > **System Verification**. The System Verification window is displayed.
5. Select calibration kit or ECal module in the **Calibration Kit** box.
6. Setup the configuration in **Run** tab and **Configure** tab as required.

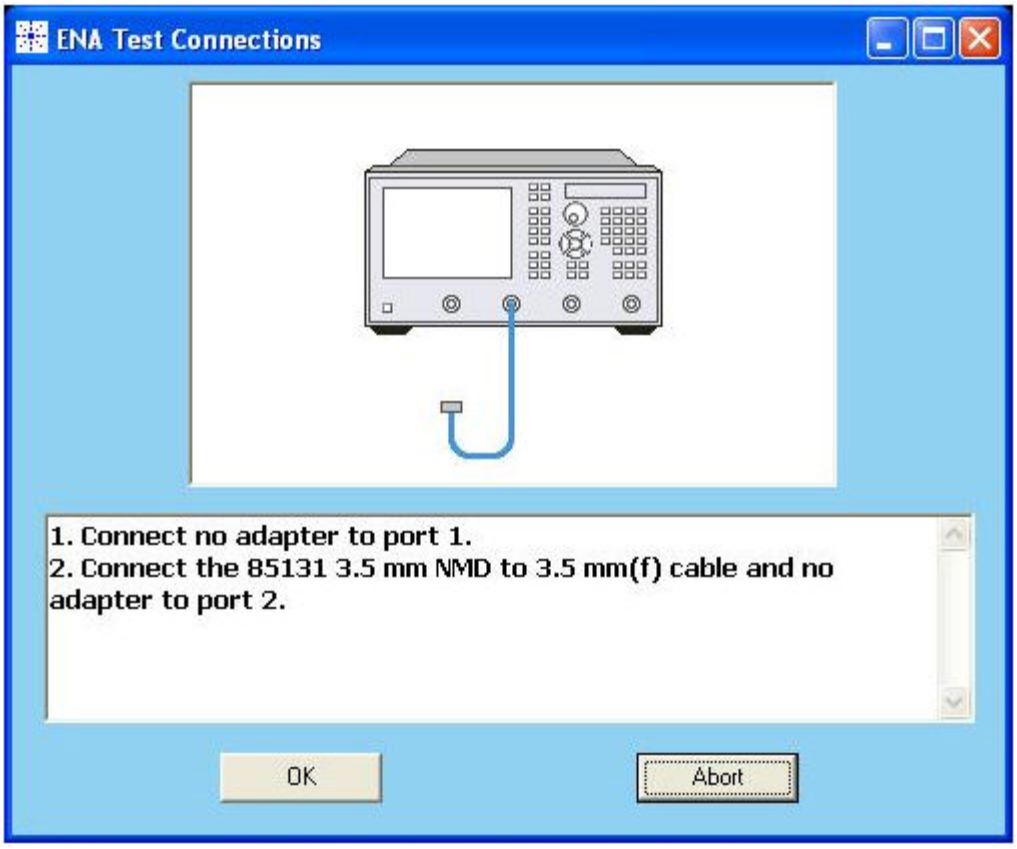
If you click **Cancel**, **Abort**, **Abort All Measurements** button in one of the following steps, the program returns to the main menu.

7. Click the **Run** button.
8. Type the serial number for both Calibration kit and Verification kit, then click the **Continue** button.
9. The user-entered serial number is printed on the Verification result report. If not necessary, blank is acceptable.



10. Connect the cable with E5071C ports, and Calibration Kit (or ECal) with adapter, then click **OK**.
11. The connection is depends on the selected Calibration Kit, test cables configuration (Single Cable or Cable Pair) and E5071C model used. The following figure shows an example case of E5071C 20GHz with Calibration Kit 85033D and Single Cable.

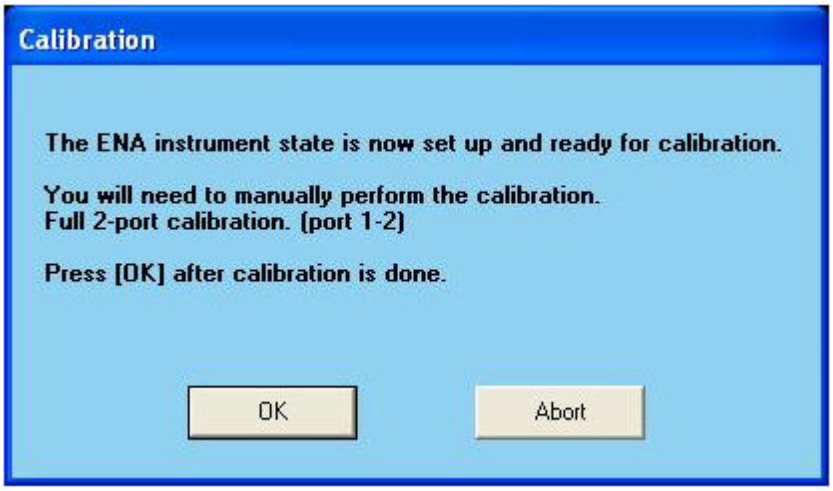
NOTE At single cable connection for the E5071C option x3x/x4x/x6x and x8x, a precision m-m adapter at port 1 and m-f cable at port 2 can be used instead for easy airline connection. The test result is regarded as typical.



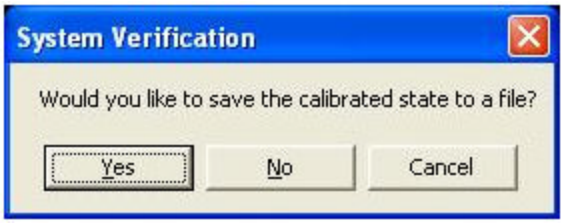
12. Calibration Execution:

a. Press **Cal** key on the E5071C front panel and execute full 2 ports calibration.

NOTE Once you have completed calibration, click **OK** in the following dialog box to proceed with the next step.



b. When the calibration is performed correctly, the following dialog box is displayed. If you want to save the calibration result into the State file, click **Yes**. Otherwise, click **No**.



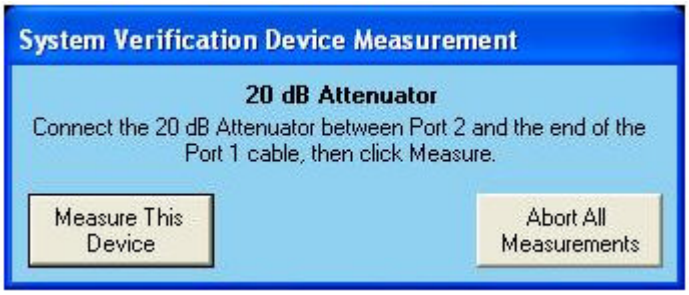
c. If the *Load Cal from* is selected from the **Calibration** option of the **Configure** tab, then this step is skipped. Instead of executing calibration, the selected State file is recalled. Then, the calibration data is set up.

13. Verification Execution:

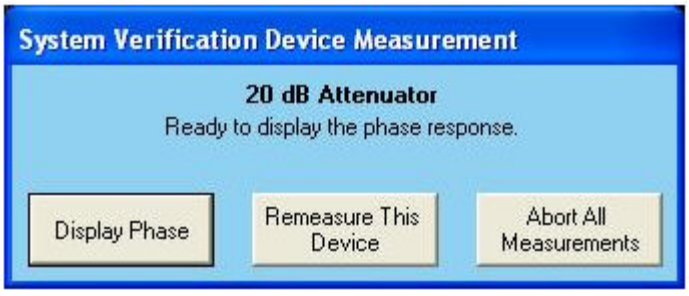
a. Execute the Verification measurements for the devices selected under **Measure Devices** options of the **Configure** tab.

b. Follow the instruction displayed in the dialog.

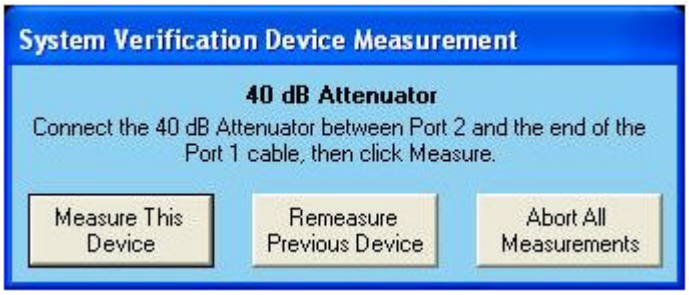
c. Connect the Verification Kit device, then click **Measure This Device** to measure. The result is displayed in the graph display area on the E5071C screen.



d. Once the measurement is completed, the following dialog is displayed:



e. Click **Display Phase** to change the display format of the graph into Phase. Instruction on the next step for the next device measurement (40 dB Attenuator) is displayed in the following dialog box:



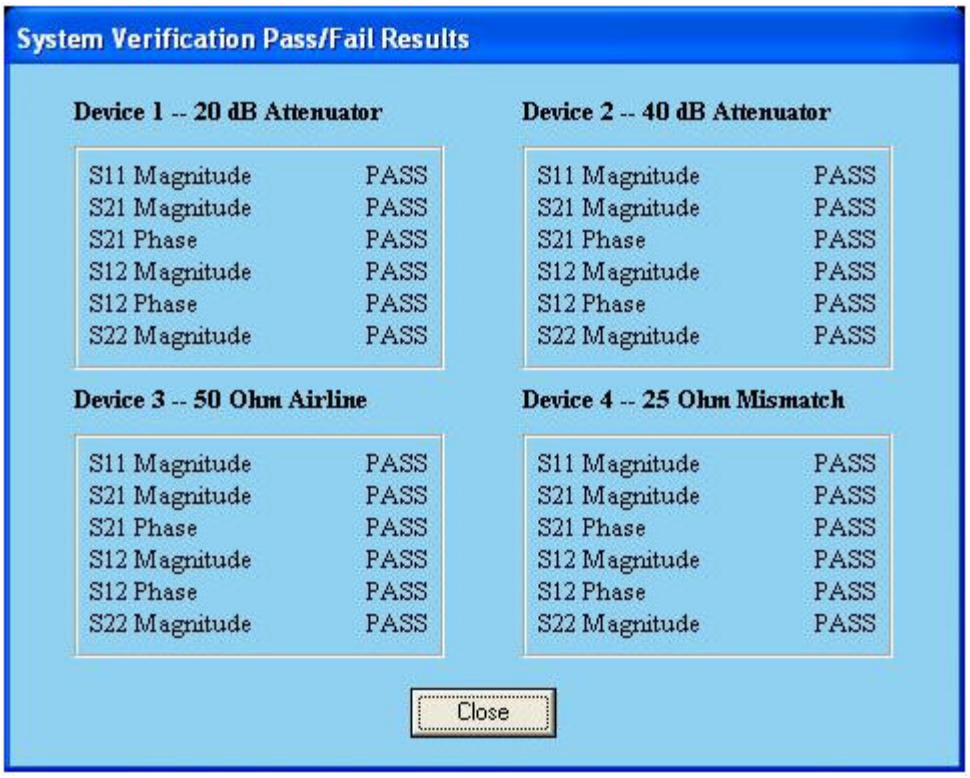
f. Click **Remeasure This Device** to return to the previous dialog so that you can re-measure the same device (20 dB Attenuator).

g. Similarly, repeat step c to e in order to measure all the devices.

h. Click **Remeasure Previous Device** to return to the previous measurement (in this example, 20 dB Attenuator).

14. Displaying the result of Verification Measurement

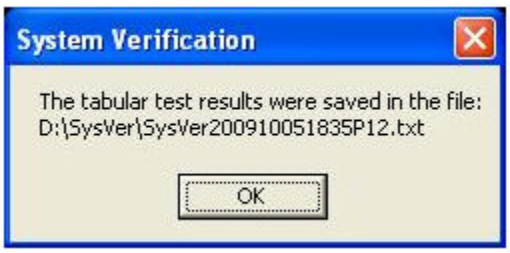
a. Once all the verification measurements for the selected devices are completed, the following result is displayed. It shows the PASS and FAIL status for each measurement.



b. Click **Close** to close the dialog box.

15. Saving the Verification result into a file

a. If the *File Tabular Data* is selected as the **Outputs** option in the **Run** tab, the following dialog is displayed. If it is not selected, this step is skipped.



b. Click **OK** to save the result into the file as shown in the dialog box.

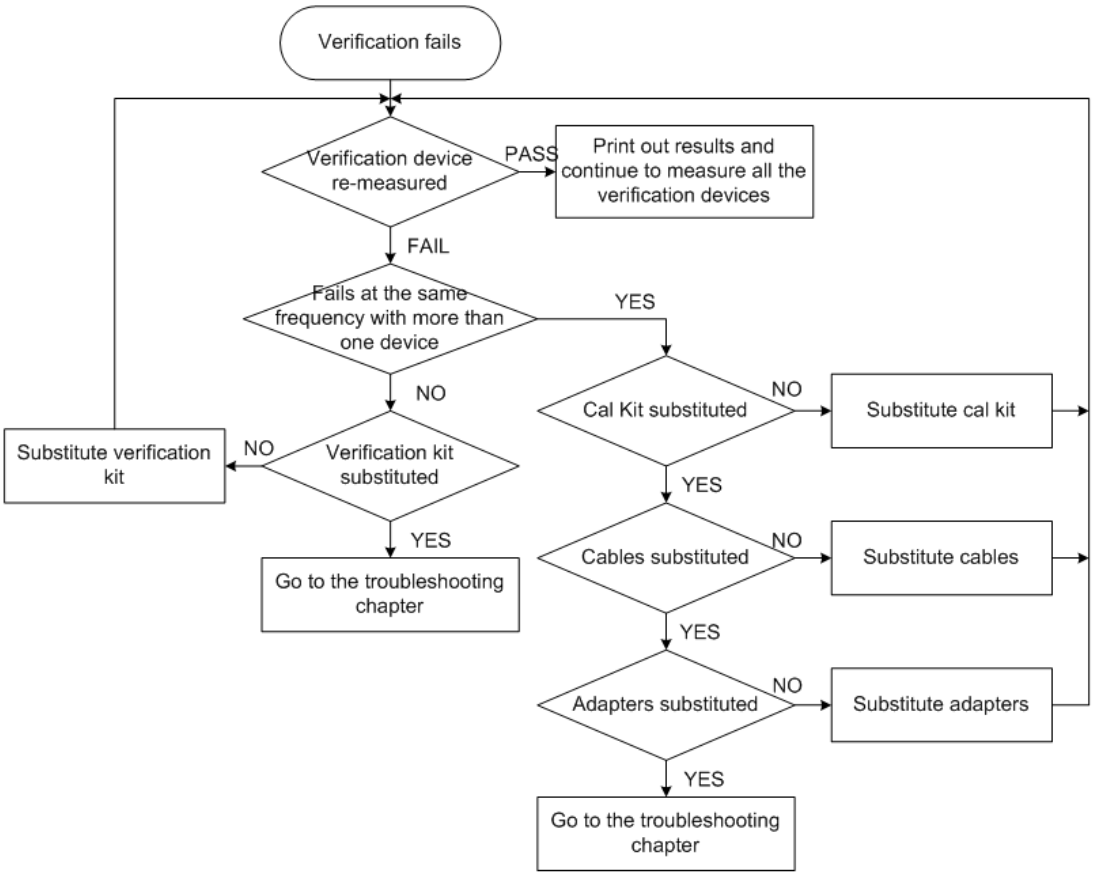
16. If both pairs (1&2 and 3&4) are selected under Measure Ports option of Configure tab for 4 port E5071C, follow the procedure from the step 10 onwards for one pair, then repeat them for another pair.

17. If only one pair is selected, follow the procedure from step 5 onwards.

If the System Fails the Verification Test

NOTE Inspect all connections. Do not remove the cable from the E5071C test port. This will invalidate the calibration that you have done earlier.

1. Repeat this verification test. Make good connections with correct torque specifications for each verification device.
2. Disconnect, clean and reconnect the device that failed the verification test. Then measure the device again.
3. If the analyzer still fails the test, check the measurement calibration.
4. Refer to the graphic below, for additional troubleshooting steps.



System Verification Result

The type of output for the Verification result can be selected under Outputs in the **Run** tab. Use these printouts to determine whether your measured data falls within the total uncertainty limits at all frequencies.

The result includes the following items:

Tabular Data

1. Header information
    - E5071C: model number, serial number, verification test time and day
    - Calibration Kit: model number, (serial number)
    - Verification Kit: model number, (serial number)
    - Measured device, Parameter
2. Data for each test point
    - Frequency

- Lower limit (defined by the total system uncertainty specification)
- Measurement result
- Upper limit (defined by the total system uncertainty specification)
- Test status (PASS or FAIL). Asterisks (**) is displayed at fail points.

Graphs

1. Upper limit line (defined by the total system uncertainty specification)
2. Lower limit line (defined by the total system uncertainty specification)
3. Measured value at factory (Memory trace)
4. Measured value (Data trace)
5. Parameter name (S11, S12 etc.), format (Lin, Log or Phase)
6. Device name and serial number

Measurement Accessories

Measurement Accessories

- Calibration Kits
- Test Port Cables/Adapters
- System Accessories

Calibration Kits

- Overview
- Mechanical Calibration Kit
- ECal

Other topics about Measurement Accessories

Overview

Calibration kits are used to improve the accuracy of the analyzer in various measurements.

The electronic calibration kit reduces the time required for calibration, mis-connections, and wear on connectors since it requires fewer changes of connection than the mechanical type.

Specifications for calibration kits and the availability of particular calibration kits are subject to change without prior notice.

Mechanical Calibration Kit

Model Name	Description	Connector Type	Frequency Range
85031B	Calibration Kit	APC-7	DC to 6 GHz
85032B/E	Calibration Kit	Type-N, 50 ohm	DC to 6 GHz
85032F	Mechanical Calibration Kit	Type-N, 50 ohm	DC to 9 GHz
85033D	Calibration Kit	3.5 mm	DC to 6 GHz
85033E	Mechanical Calibration Kit	3.5 mm	DC to 9 GHz
85036B	Calibration Kit	Type-N, 75 ohm	DC to 3 GHz
85036E	Economy Calibration Kit	Type-N, 75 ohm	DC to 3 GHz
85038A/F/M	Calibration Kit	7-16	DC to 7.5 GHz

85039B	Calibration Kit	Type F , 75 ohm	DC to 3 GHz
85050C	Precision Mechanical Calibration Kit	APC-7	DC to 18 GHz
85050D	Economy Mechanical Calibration Kit	APC-7	DC to 18 GHz
85052D	Economy Mechanical Calibration Kit	3.5 mm	DC to 26.5 GHz
85052C	Precision Mechanical Calibration Kit	3.5 mm	DC to 26.5 GHz
85054D	Economy Calibration Kit	Type-N, 50 ohm	DC to 18 GHz
85056D	Economy Calibration Kit	2.4 mm	DC to 50 GHz
85056K	Mechanical Calibration Kit	2.4 mm and 2.92 mm	DC to 50 GHz
K11644A	Mechanical Calibration Kit (waveguide)	WR-42	18 GHz to 26.5 GHz
P11644A	Mechanical Calibration Kit (waveguide)	WR-62	12.4 GHz to 18 GHz
X11644A	Mechanical Calibration Kit (waveguide)	WR-90	8.2 GHz to 12.4 GHz

- 85054D, 85056D, 85056K, X1164A, P1164A and K1164A Calibration kits are added from Firmware A.09.10.
  Delay value in Open Standard is updated for 85052C ,85052D, 85033D and 85033E calibration kits.

For more information on definition of calibration kit, refer to http://na.tm.agilent.com/pna/caldefs/stddefs.html

- From firmware A.09.10, the number of calibration kit definition is increased from 20 to 30 and number of sub class is increased from 8 to 30.

ECal (electronic calibration) kit

The ECal supported by E5071C are defined below:

Model Name	Connector Type	Frequency Range
85091B	7 mm	300 kHz to 9 GHz
85091C	7 mm	300 kHz to 9 GHz
85092B	Type N	300 kHz to 9 GHz
85092C	Type N	300 kHz to 9 GHz
85093B	3.5 mm	300 kHz to 9 GHz
85093C	3.5 mm	300 kHz to 9 GHz
85096B	Type N, 75 ohm	300 kHz to 3 GHz
85096C	Type N, 75 ohm	300 kHz to 3 GHz
85098B	7-16	300 kHz to 7.5 GHz
85098C	7-16	300 kHz to 7.5 GHz
85099B	Type F	300 kHz to 3 GHz
85099C	Type F	300 kHz to 3 GHz
N4431A	3.5 mm, Type N	300 kHz to 9 GHz
N4431B	3.5 mm, Type N	9 kHz to 13.5 GHz (see note)
N4432A	Type N	300 kHz to 18 GHz
N4433A	3.5 mm	300 kHz to 20 GHz
N4690A	Type N	10 MHz to 18 GHz
N4690B	Type N	300 kHz to 18 GHz

N4691A	3.5 mm	10 MHz to 26.5 GHz
N4691B	3.5 mm	300 kHz to 26.5 GHz
N4692A	2.92 mm	10 MHz to 40 GHz
N4696A	7 mm	10 MHz to 18 GHz
N4696B	7 mm	300 kHz to 18 GHz

- Firmware A.09.10 or above, enable users to calibrate down to 9 kHz using N4431A (4-port).
- The 9 kHz to 300 kHz performance is valid only for the E5071C with Firmware revision A.09.10 or above using N4431B.
- When Ecal is used with options 2D5, 4D5, 2K5 and 4K5 and in the following condition, the waiting time is added automatically.
  - Firmware A.09.10: The maximum frequency of your ECal is over 13 GHz.
  - Firmware A.09.20 and above: The maximum frequency of your ECal is over 13 GHz and the stop frequency on your ENA set over 8.5 GHz. (The waiting time is NOT added when stop frequency is set below 8.5GHz.)
- For scalar mixer calibration, the stop frequency of calibration port is used to determine the waiting time instead of the sweep stop frequency.

The total waiting time of each calibration type (approximately) is as follows:

Calibration Type	Total waiting time for E5071C (sec)
1-port	9.1
2-port	18.1
3-port	27.1
4-port	36.1

- ENA does not support the following ECal:

Model Name	Connector Type	Frequency Range
N4693A	2.4 mm	10 MHz to 50 GHz

N4694A	1.85 mm	10 MHz to 67 GHz

Test Port Cables / Adapters

- Overview
- Test Port Cables
- Adapters

Other topics about Measurement Accessories

Overview

Agilent Technologies provides various cables and adapters. The following are one of them.

Test Port Cables

N6314A 50 ohm N Type RF Cable (300 kHz to 9 GHz)

An RF cable 610 mm in length with male N type connectors on both ends.

N6315A 50 ohm N Type RF Cable (300 kHz to 9 GHz)

An RF cable 610 mm in length with a male and a female N type connector on each end.

Adapters

11853A 50 ohm N Type Accessory Kit

The 11853A kit includes the following items.

Agilent Cat. No.	Description
1250-1472	N type (f)-(f) adaptor kit (two adaptors)
1250-1475	N type (m)-(m) adaptor kit (two adaptors)
11511A	N type (f), short
11512A	N type (m), short

11878A N type to 3.5 mm Adaptor Kit

The 11878A kit includes the following items.

Agilent Cat. No.	Description
1250-1744	3.5 mm (f) to N type 50 ohm (m) adaptor
1250-1743	3.5 mm (m) to N type 50 ohm (m) adaptor
1250-1745	3.5 mm (f) to N type 50 ohm (f) adaptor
1250-1750	3.5 mm (m) to N type 50 ohm (f) adaptor

11854A 50 ohm BNC Accessory Kit

The 11854A kit includes the following items.

Agilent Cat. No.	Description
1250-0929	BNC (m), short
1250-1473	BNC (m) to N type (m) adaptor kit (two adaptors)
1250-1474	BNC (f) to N type (f) adaptor kit (two adaptors)
1250-1476	BNC (f) to N type (m) adaptor kit (two adaptors)
1250-1477	BNC (m) to N type (f) adaptor kit (two adaptors)

11852B Minimum-loss pad

Option	Description
Option 004	N type connectors, 50 ohm (m) to 75 ohm, (f)

System Accessories

- USB/GPIB Interface
- GPIB Cables
- Agilent IO Libraries

Other topics about Measurement Accessories

82357A/B USB/GPIB Interface

The 82357A/B can be used to control external GPIB devices by the E5071C. It can also be used to control E5071C by PC with USB.

GPIB Cables

The following GPIB cables can be used to connect the analyzer with an external device such as a computer.

Product Number	Length
10833A	1.0 m (3.3 ft)
10833B	2.0 m (6.6 ft)
10833C	3.0 m (9.9 ft)
10833D	0.5 m (1.6 ft)

Agilent IO Libraries

Agilent IO Libraries is a collection of libraries that give you the ability to use your instruments from a test and measurement program, and utilities that help you quickly and easily connect your instruments to your PC.

General Principles of Operation

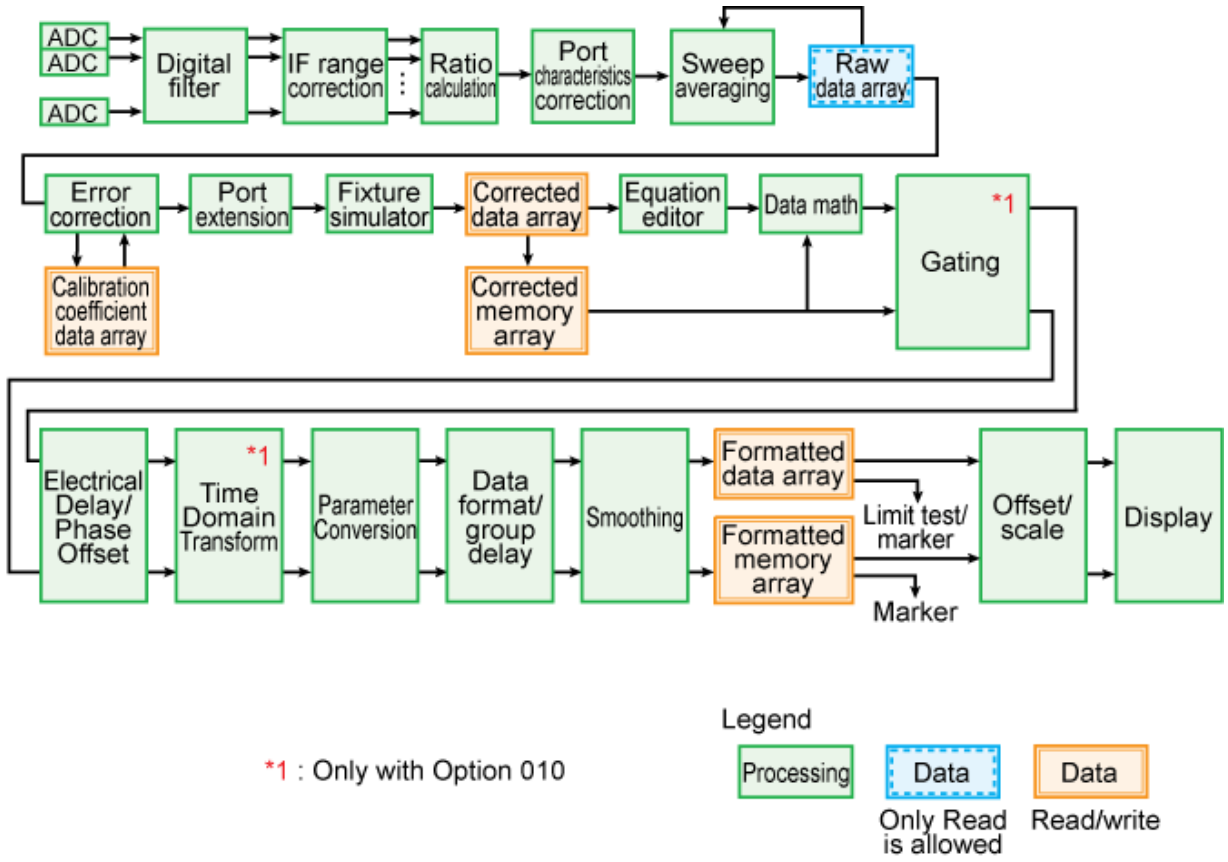
General Principles of Operation

- Data Processing
- System Description

Data Processing

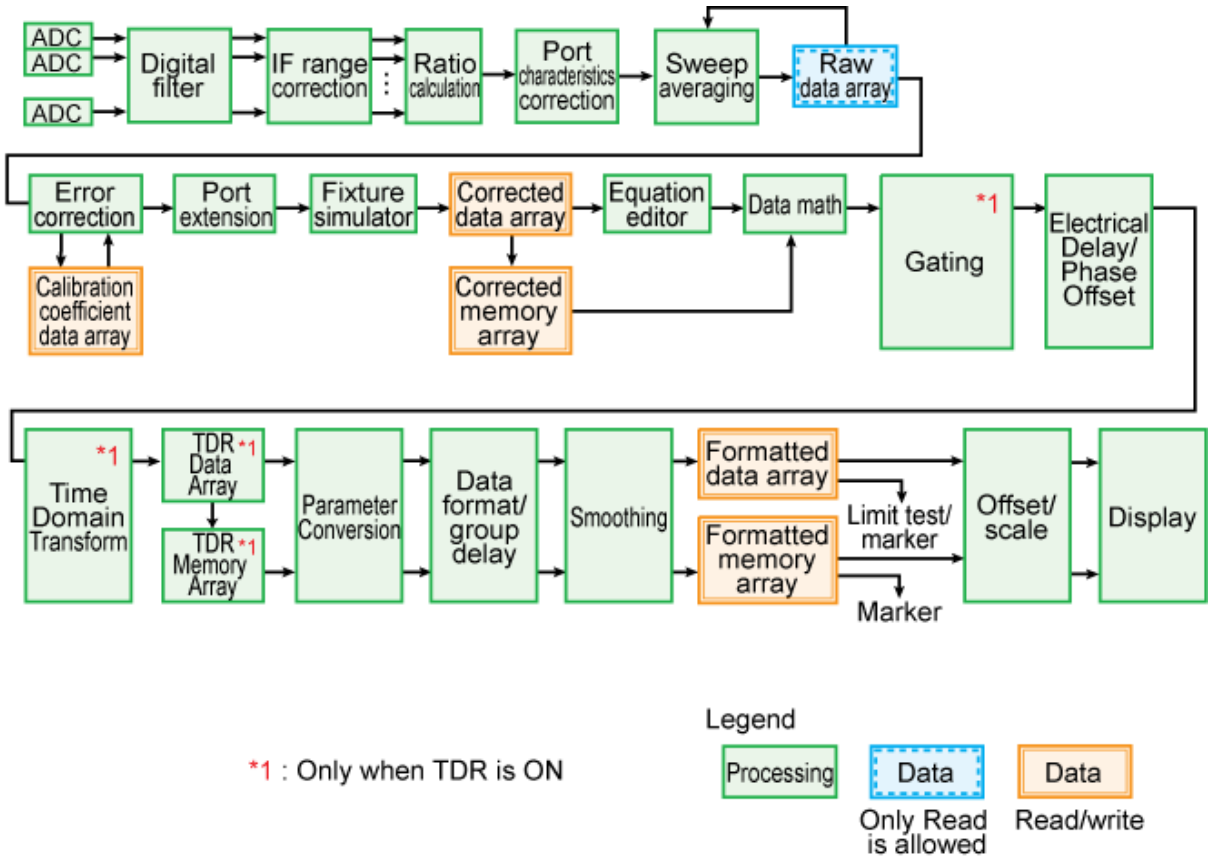
The internal data processing flowchart for the E5071C is shown in the following figure.

Data Processing Flowchart



e5071c574

Data Processing Flowchart (Option TDR)



e5071c575

ADC

The ADCs (analog-to-digital converters) convert analog signals fed to the receiver and converted into IF signals (R_1, R_2, ... , R_n and T_1, T_2, ... , T_n) into digital signals. One ADC is available for each signal and the conversion takes place simultaneously.

Digital Filter

The digital filter performs a discrete Fourier transformation (DFT) and picks up IF signals. Each IF signal is then converted into a complex number that has a real part and an imaginary part. The IF bandwidth of the analyzer is equivalent to the bandwidth of the DFT filter. The IF bandwidth can be set in the range of 10 Hz ~ 100 kHz.

IF Range Correction

Input signals that went through ranging at the receiver are reverted (corrected) to previous values before the ranging.

Ratio Calculation

The ratio between two signals is determined by performing divisions on complex numbers. In the case of absolute measurements (Option 008), the ratio of complex number can not be calculated.

Port Characteristics Correction

The equivalent source match error, the directivity error, and the tracking error of each test port bridge are corrected. In the case of absolute measurements (Option 008), the gain of each test port is corrected.

Sweep Averaging

The average of complex indices is determined based on data obtained from multiple sweep measurements. Sweep averaging is effective in reducing random noise in measurements.

Raw Data Array

The results from all data processing done up to this point are stored in this array as raw data. All prior data processing is performed as each sweep takes place. When the full N-port error correction (N=2 to 4) is enabled, all N × N S parameters are stored in the raw data array and used in error correction. Users are allowed to access (read only) this raw data array.

Error Correction / Calibration Coefficient Data Array

When error correction is enabled, the process eliminates the system errors that are reproducible and stored in the calibration coefficient data array. It accommodates everything from the simple vector normalization to the full 12-term error correction. The user is allowed to access (read/write) this calibration coefficient data array. Gain correction is performed in absolute measurement.

Port Extension

This process carries out a simulation of adding or eliminating a variable length no-loss transmission path on each test port so that the reference plane of calibration is moved. The port extension is defined by an electrical delay (sec).

Fixture Simulator

A data conversion by the fixture simulator function is executed. The fixture simulator function is a collective term for six different functions: balanced-unbalanced conversion, addition of matching circuits, port reference impedance conversion, network elimination, addition of differential matching circuits, and differential reference impedance conversion.

Corrected Data Array

Unlike the raw data array, this array stores the results obtained after error corrections, port extensions, or the fixture simulator functions are applied. The user is allowed to read/write data from/to the corrected data array.

Corrected Memory Array

By pressing **Display** > **Data -> Mem**, the contents of the corrected data array will be copied to this array. The user is allowed to read/write data from/to the corrected memory array.

Equation Editor

Equation Editor is a function available in Firmware revision A.08.0x or later that allows users to use a custom equation to display data in the E5071C. Equation Editor can be accessed through **Display** > **Equation Editor**.

Data Math

Data processing is carried out using the corrected data array and the corrected memory array. Four types of data processing addition, subtraction, multiplication, and division are available.

Electrical Delay / Phase Offset

An electrical delay and a phase offset are applied to each trace. By setting an electrical delay, a linear phase that is proportional to the frequency will be added or subtracted. On the other hand, setting a phase offset adds or subtracts a phase that is constant throughout the frequency range. Incidentally, data processing performed from this point on in the flowchart is applied to both the data array and the memory array.

TDR Data Array

The results from all data processing done up to this point are stored in this array. The user is not allowed to access (read/write) this data array.

TDR Memory Array

By pressing Display > Data -> Mem, the contents of the TDR data array will be copied to this array. The user is not allowed to access (read/write) this memory array.

Data Format / Group Delay

Complex data consisting of the real parts and the imaginary parts are converted into scalar data according to the data format of user's choice. Group delays are also calculated here.

Smoothing

By enabling the smoothing function, each point in a sweep measurement will be replaced by a moving average of several measurement points nearby. The number of points used in calculating a moving average is determined by the smoothing aperture set by the user. The smoothing aperture is defined by a percentage against the sweep span.

Formatted Data Array / Formatted Memory Array

All results from data processing are stored in the formatted data array and the formatted memory array. The marker functions are applied to these

arrays. The limit test is applied to the formatted data array. The user is allowed to read/write data from/to these arrays.

Offset / Scale

Each set of data is processed so that traces can be drawn on the screen. Particular scaling depending on the data format is applied using the position of the reference line, the value of the reference line, and the scale/graticule settings.

Display

The results obtained after data processing are displayed on the screen as traces.

Other topics about General Principles of Operation

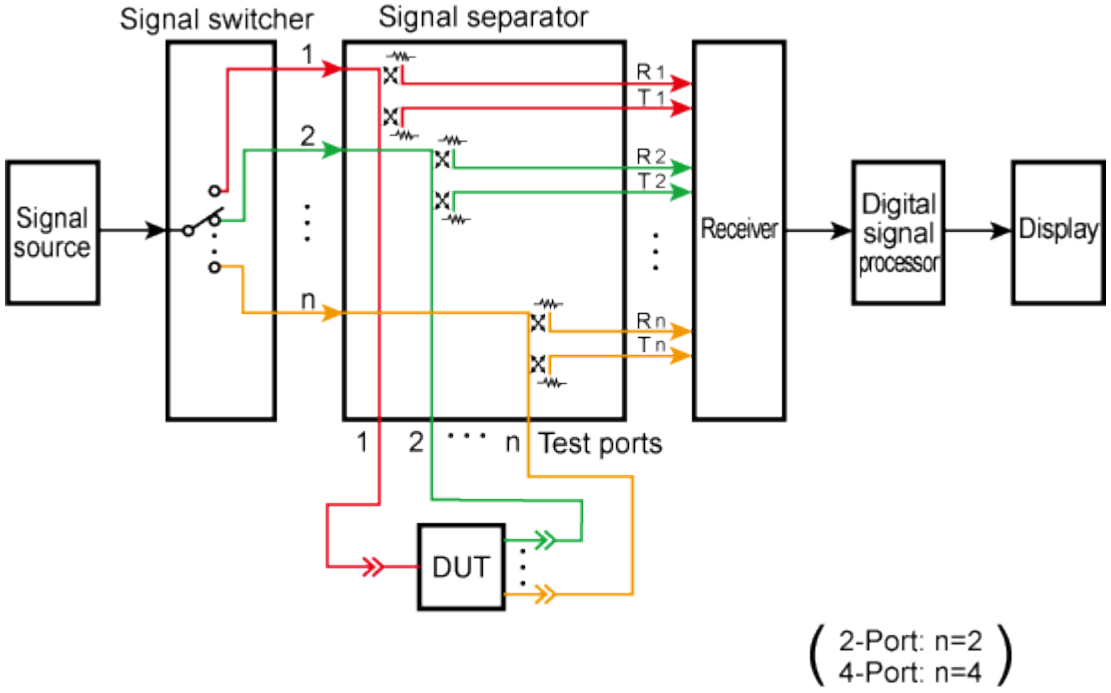
System Description

- Synthesized Source
- Source Switcher
- Signal Separator
- Receiver

Other topics about General Principles of Operation

A network analyzer supplies a sweep signal to a DUT, measures its transmission and reflection, and displays the results as ratios against the input signal from the signal source. The E5071C network analyzer consists of the circuit modules shown in the following figure.

System Diagram for the E5071C Network Analyzer



e5071c241

Synthesized Source

The synthesized source generates an RF sweep signal in the specified frequency range.

The signal source is phase-locked to a highly reliable quartz crystal oscillator to maintain a high level of accuracy in its frequency as well as to achieve precise phase measurements. The level of RF output power is controlled within the range of -15 to +10 dBm by the internal ALC (automatic leveling control) circuit.

Source Switcher

The source switcher is used to switch test ports to which the RF signal is supplied from the source.

Signal Separator

The signal separator consists of directivity couplers that detect input and output signals at the test ports. On a test port to which a signal is output, the output signal and the reflection from the DUT are detected as the reference signal (R) and the test signal (T), respectively. On the other ports, the signal that is transmitted through the DUT is detected as the test signal (T). All signals are then sent to the receiver.

Receiver

Each signal that is sent to the receiver is first converted into an IF signal by a mixer and then converted into a digital signal by an ADC (analog to digital converter). These processes are applied to each signal independently. The digital data is then analyzed by a micro processor and measurement results will be displayed on the screen.

Revision History

Revision History

Revision History

- Firmware Revision History
- HDD Revision History
- Data Sheet Revision History

Other topics about Product Information

Firmware Revision History

- Firmware Revision History
  - A.09.61
  - A.09.60
  - A.09.5x
  - A.09.4x
  - A.09.31
  - A.09.30
  - A.09.2x
  - A.09.1x
  - A.08.1x
  - A.08.0x
  - A.07.0x
- TDR software Revision History
  - A.01.02

Other topics about Revision History

Firmware Revision History

A.09.61

- S6P/S8P are supported.

A.09.60

- Turning ON/OFF segment individually (Segment Sweep) is supported.
- Setting multiple range is allowed.
- Multiple ECal connection is added.
- Sliding load and offset load calibration is supported.
- RAW data array is readable.
- Trace data direct selection is allowed.
- Trace name assignment is allowed.
- The following commands have been added.
  - SCPI.SENSe(Ch).SEGMent.LIST.CONTrol.DATA
  - SCPI.SENSe(Ch).SEGMent.LIST.CONTrol.STATe

- SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.RANGe
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STARt
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STATe
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STOP
- SCPI.SYSTem.COMMunicate.ECAL.CATalog
- SCPI.SYSTem.COMMunicate.ECAL.DEFine
- SCPI.SENSe(Ch).DATA.CORRdata
- SCPI.SENSe(Ch).DATA.RAWData

A.09.5x

- Enhanced Time Domain Analysis Option (Option TDR) is added.
- Bal is added on the balanced devices type.
- The following commands have been added.
  - SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter(Tr).BALanced.DEFine
  - SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.BALanced.PORts
- The parameter of following command has been changed.
  - SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

A.09.4x

- System Verification is added.

A.09.31

- SCPI.CALCulate(Ch).SELected.MARKer(Mk).DATA is added.
- The parameters of SCPI.CALCulate(Ch).SELected.MARKer(Mk).SET are added.

A.09.30

- 6.5 GHz/14 GHz frequency model are supported.

  (The information of option 260, 265, 460, 465, 2D5, 4D5 is added.)

A.09.2x

New Feature

- Support for USB Power Sensor
  - Selecting the power meter

    - Setting USB power meter
    - Performing Zeroing/Calibration
  - Performing 8 term Calibration using External PC
  - Receiver Calibration
    - Power Offset
    - R, T Channel independent Calibration
  - Removing Log Data
  - SCPI Command:
    - SCPI.CALCulate(Ch).SELected.DATA.XAXis
    - SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.RCHannel.ACQuire
    - SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.TCHannel.ACQuire
    - SCPI.SENSe(Ch).CORRection.RECeiver(Pt).OFFset.AMPLitude
    - SCPI.SENSe(Ch).CORRection.COLLect.ECAL.INFormation
    - SCPI.SERVice.ADjust.VIRTual.BRIDge.STATe
    - SCPI.SERVice.ADjust.SENse(Ch).SWEep.RF.RANGe.PORT(Pt).AUTO
    - SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog
    - SCPI.SYSTem.COMMunicate.PSENsor
    - SCPI.SERVice.LOGGing.CLEar

Modifications

- Channel/Traces Setup (Support for 2ch/8Tr 10001 points)
- Reduce waiting time for ECal calibration
- Overload/Detection and Power Trip function

A.09.1x

New Feature

- Supported Frequency and Power Range
- Auto Power Range Function
- E5092A
- Support for Power Sensors
- Channel/Traces Setup
  - Channel Display Setup

    - Channel/Traces Number
- Activating Software Option
- Waveguide
    - SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff
    - SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth
    - SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate
    - SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff
    - SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME
- Overload Detection/Power Trip Function

Modifications

- Equation Editor
- MWA
- Context Sensitive
- Save/Recall
    - Compatibility of Files (Saving/Recalling)
    - Recall Procedure
- Calibration Kit
- Error Message
- Protecting ENA System
- Default Condition
    - Default Condition Finder
    - Preset, *RST , Save/Recall
- SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ISOLation.STATe

A.08.1x

New Feature

- Extended channels (up to 96 Channels)
- Added Channel display (8x9, 6x12 and 8x12)
- Notch Search
    - Determining the Bandwidth of the Trace (Notch Search)

- SCPI.CALCulate(Ch).SELected.MARKer.NOTCh
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.DATA
- SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.THReshold

Modifications

- Parallel Printer Port is not supported (only USB port is supported)
- Removable Hard Disk Type II
- Enhanced Command Finder

A.08.0x

- Context Sensitive Help
- Option up to 4.5 GHz
- Equation Editor
  - Equation Editor
  - SCPI.CALCulate(Ch).SELected.EQUation.STATE
  - SCPI.CALCulate(Ch).SELected.EQUation.TEXT
  - SCPI.CALCulate(Ch).SELected.EQUation.VALid
- LXI
- Trigger Scope
  - Trigger Scope (Active Only Trigger)
  - SCPI.TRIGger:SCOPe
- Faster Recall
- User Variable
  - SCPI.PROGram.VARiable.ARRay(Vnum).DATA
  - SCPI.PROGram.VARiable.ARRay(Vnum).SIZE
  - SCPI.PROGram.VARiable.DOUBle(Vnum).DATA
  - SCPI.PROGram.VARiable.LONG(Vnum).DATA
  - SCPI.PROGram.VARiable.STRing(Vnum).DATA
- Revised Firmware Revision Information
- Updated Data Processing Flowchart
- Calkit Definition for 85038A/F/M
- Support for N1911A and N1912A Power Meter

A.07.0x

This is the first revision of E5071C.

New Features

- Improved Measurement Performance (Dynamic Accuracy, Trace Noise, Measurement Speed)
- DC Measurement (AUX Ports)
- External Trigger Output
- Probe Power
- 65dB Full Range Power Sweep (@ 9 kHz to 5 GHz)
- Extended IF Bandwidth up to 500 kHz
- Extended channels up to 36 Channels
- Unknown Thru Calibration (Unknown Thru Calibration with ECal)
- Adapter Removal-Insertion function
- Absolute Measurement is available even if option 008 is not installed.
- External Test Set Mode
- Windows (Open environment)
- XGA 10.4 inch LCD
- More USB ports (Front 2, Rear 4)
- Online Help
- Internal DC Bias Tee (Option 235, 435, 285, and 485)
- Extended the lowest frequency to 9 KHz (Option 230, 430, 280, and 480)/100 kHz (Option 235, 435, 285, and 485)
- Touch Panel LCD is equipped as a standard.

Deleted Functions from E5070B/E5071B

- Floppy Disk Drive
- Fast Stepped/Fast Swept mode
- Power Range
- Avoid Spurious
- Warm Up Ready
- Adapter Characterization VBA

TDR software Revision History

A.01.02

- Height (dB) is deleted from the result.
- TDR Mask Test: It is required to define the values of the Logic 1 Lv and Logic 0 Lv in the scale tab for Mask.

HDD Revision History

HDD revision is based upon a number of factors such as windows OS, driver upgrade patch which are installed at the factory shipment. Firmware revision denotes E5071C measurement software. To know E5071C HDD revision, refer to Checking the product information.

- CN96x
- CN95x
- CN94x
- CN93x
- CN925
- CN922
- CN80x

Other topics about Revision History

CN96x

- No functionally update

CN95x

- No functionally update

CN94x

- No functionally update

CN93x

- No functionally update

CN925

- Initial registration procedure has been changed
- Factory recovery procedure has been changed
- User recovery capability has been deleted

CN922

- cXL support:cXL software is pre-installed
- Windows SP3

CN80x

- LXI is supported (dotnet framework is installed)

Data Sheet Revision History

The data sheet revision history is shown below. The print date is used to define the revision. See the last page of the data sheet for its print date.

- March 3, 2011
- July 5, 2010
- June, 2009
- February 3, 2009

Other topics about Revision History

March 3, 2011

- Stability per switch of the E5092A is changed.

July 5, 2010

- The information for the E5091A is deleted.
- Test Port Output: Source stability and CW Accuracy is changed from ±5 ppm to ±7 ppm.

June, 2009

- 6.5 GHz/14 GHz frequency model are added
  Option 260, 265, 460, 465, 2D5, 4D5 and its related information are added.
- Corrected system performance with 3.5 mm device connectors, 85052D calibration kit, Figures of "Transmission uncertainty" has been corrected.
- Corrected system performance with 3.5 mm device connectors, N4691B electronic calibration (ECal) module, Figures of "Transmission uncertainty" has been corrected.

February 3, 2009

The following specification has been changed.

Corrected System Performance with Calibration Kit

Option 230/235/240/245/280/285/430/435/440/445/480/485

- The data for 85092C and 85093C are changed.

Option 2k5/4K5

- The data for 85052D and N4691B are changed.

Uncorrected System Performances

Option 2k5/4K5

- At 300 kHz to 1MHz, Directivity and Source Match are changed from 25 dB to 20 dB.

Replacing 8753ES with E5071C

Replacing 8753ES with E5071C

The information necessary to replace Agilent 8753ES with the Agilent E5071C is described below. Agilent provides some tools such as a Code Conversion Assistant Editor for the command translation for programming.

- Important Functional Differences
- Code Migration Tools
- Comparing Functions

Important Functional Differences

This section describes the key differences between the Agilent 8753ES and Agilent E5071C.

- Channel and Trace Concepts
- Measurement Parameters
- Test Port Output Ranges
- Sweep Function
- Calibration
- Trigger System
- Data Flow
- Reading/Writing Data
- Screen Display and Marker Functions
- Math Operation Functions on Traces
- Device Test Functions
- Analytical Functions
- Save/Recall
- Test Sequence Program
- Outputting to a Printer/Plotter
- GPIB Interface
- LAN Interface
- Other Functions

Other topics about Replacing 8753ES with E5071C

Channel and Trace Concepts

In the 8753ES, channels 1 and 2 are independent from each other and have auxiliary channels, channels 3 and 4, respectively. Channels 3 and 4 can be displayed as additions to channels 1 and 2, respectively. This allows up to four channels to be displayed for up to four traces on the screen simultaneously. Channels 1 and 3 and channels 2 and 4 are always coupled, while channels 1 and 2 are independent from each other. This enables you to specify different sweep conditions on each of channels 1 and 2. The E5071C has 4, 9 , 12, 16 and 36 independent channels, each of which allows sweep conditions to be defined different from those on other channels. On the screen you can open up to maximum 36 windows for

channels. The channel allows up to maximum 16 traces to be defined depending on the number of channel.

Measurement Parameters

With the E5071C, you can perform measurements of S parameter and absolute value of the power input into ports. In addition to these, the 8753ES supports measurement parameters such as A, B, R, A/R, B/R, A/B, etc.

Test Port Output Ranges

The 8753ES comes standard with test sets for two ports, while the E5071C comes optionally with test sets for two ports and four ports. Furthermore, the 8753ES comes optionally with a 75 ohm test set (Option 075), while the E5071C does not.

For more about the measurement frequency ranges of the 8753ES and E5071C, see the following table.

Measurement Frequency Ranges

Function	8753ES	E5071C
Measurement frequency range	• 30 kHz to 3 GHz (std.) • 30 kHz to 6 GHz (Option 006)	See Test Port and Frequency Options

For more about the output power levels and output power ranges of the 8753ES and E5071C, see the following table.

Test Port Output Power Levels

Function	8753ES	E5071C
Output power levels	• - 85 dBm to 10 dBm (std.) • - 85 dBm to 8 dBm (Options 014 and 075)	• - 55 dBm to 10 dBm (3 GHz, 4.5 GHz, 6.5 GHz, 8.5 GHz options) • -85 dBm to +10 dBm (14 GHz and 20 GHz options)
Output power ranges	• - 15 dBm to 10 dBm (std.) or -	None

	15 dBm to 8 dBm (Options 014 and 075) • - 25 dBm to 0 dBm • - 35 dBm to - 10 dBm • - 45 dBm to - 20 dBm • - 55 dBm to - 30 dBm • - 65 dBm to - 40 dBm • - 75 dBm to - 50 dBm • - 85 dBm to - 60 dBm	

Sweep Function

The list (segment) sweep function enables you to perform measurements corresponding to two or more sweep conditions in one sweep operation and is supported both on the 8753ES and E5071C. While the 8753ES allows only up to 30 segments per table to be defined, the E5071C allows up to 201 segments to be defined. Furthermore, while two or more commands are needed to create a table using GPIB (SCPI) commands on the 8753ES, only one command does the same function on the E5071C.

The 8753ES supports an IF bandwidth up to 6 kHz, but the E5071C supports an IF bandwidth up to 500 kHz, enabling faster sweep operations than with the 8753ES.

While the 8753ES supports automatically selecting the output power range, the E5071C supports only manual selection.

The 8753ES allows you to measure frequency conversion device such as a mixer; since it supports the frequency offset sweep. The E5071C also supports this function when the option 008 is used.

Calibration

The types of calibration kits supported by the 8753ES and E5071C are shown in the following table.

Supported Calibration Kits

Type of calibration kit	8753ES	E5071C
7 mm	• 85031B	• 85031B • 85050C/D
3.5 mm	• 85033C/D/E	• 85033D/E • 85052D
N type	• 50 ohm : 85032B/E/F • 75 ohm : 85036B/E	• 50 ohm : 85032B/F • 75 ohm : 85036B/E
2.4 mm	• 85056A/D	N/A
TRL 3.5 mm	• 85052C	• 85052C
Others	User-defined calibration kit	User-defined calibration kit

The 8753ES allows only one type of user-defined calibration kit to be saved in the internal memory. Up to 15 classes can be set up when defining calibration kits, including 12 classes (isolation included) to be used for full 2-port calibration and three calibration classes (THRU, reflect, and line/match for TRL*/LRM* calibration. A maximum of seven standards can be defined for each calibration class.

In contrast, the E5071C allows 20 types of user-defined calibration kits to be saved in the internal memory, which include the nine calibration kits registered beforehand. When setting up calibration classes, OPEN, SHORT, and LOAD can be set up on each port and THRU between ports. Eight standards are allowed for each calibration class.

ECal

Both the 8753ES and E5071C support ECal measurement. However, each supports slightly different functions. The 8753ES supports enhanced response calibration, 1-port calibration for S11 and S22, and full 2-port calibration. The E5071C provides multi-port calibrations including enhanced response calibration, 1-port calibration for S11 and S22, full 2-port calibration, full 3-port calibration (with 4 ports model) and full 4-port calibration (with option 4 ports model).

The 8753ES allows a frequency array or module information to be read, but these functions are not supported on the E5071C.

Trigger System

The trigger system detects the signal for starting a measurement (trigger) and controls decisions on whether to measure or not measure.

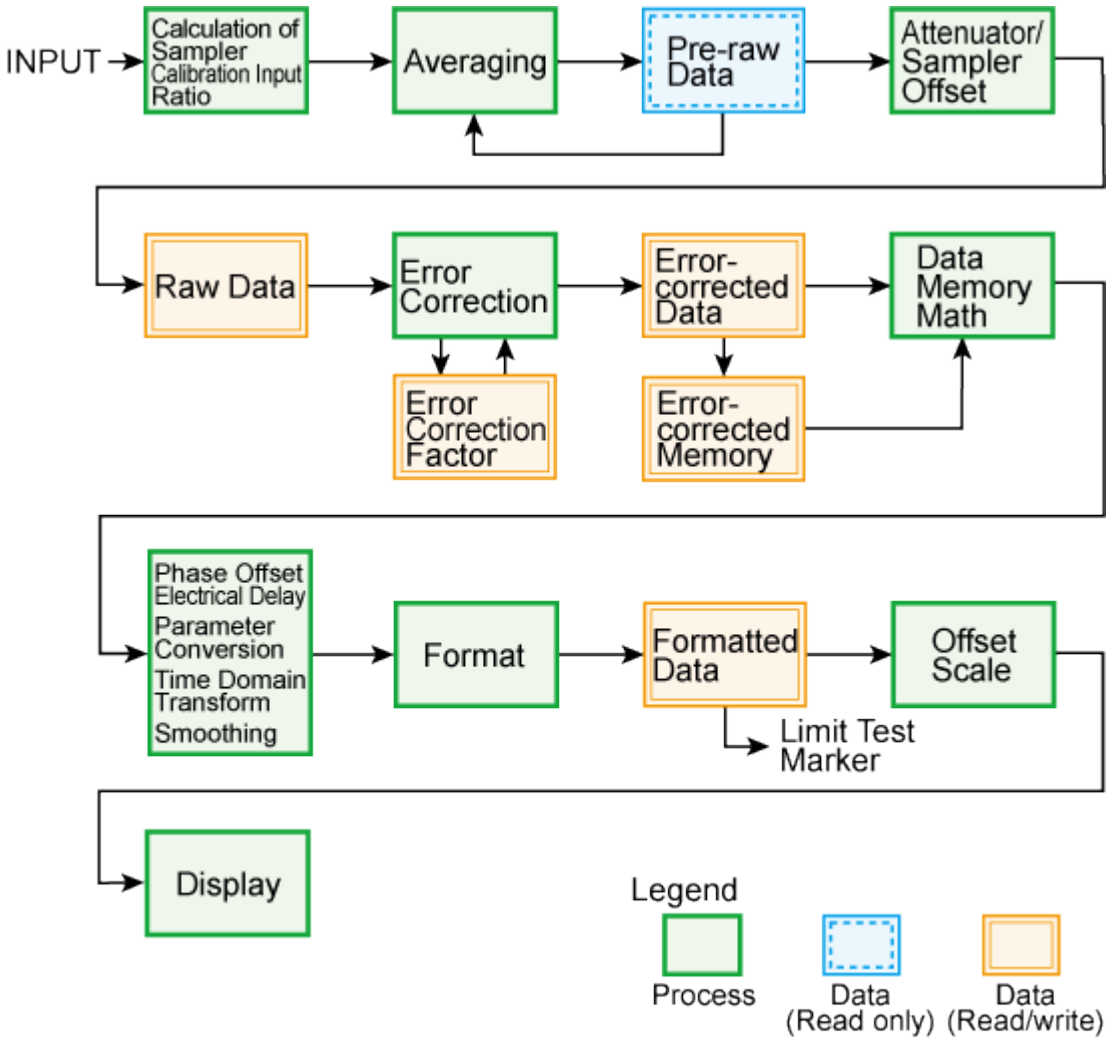
On the 8753ES, the trigger state is available for the pair of a main channel and an auxiliary channel (two pairs: channels 1 and 3 and channels 2 and 4). For each pair, three states are available: Hold, Waiting for Trigger, and Measurement. When a trigger event occurs, one pair of channels in the Waiting for Trigger state are put into a sweep operation. If the other pair is also Waiting for Trigger, then the next trigger event puts it into sweep operation as well. When the sweep condition coupling channels is turned on, the Hold, Waiting for Trigger, and Measurement states are common to all channels. In this case, when a trigger event occurs in the Waiting for Trigger state, all channels are put into sweep operation. For example, when you set channel 1 and 2 to uncouple and sweep each channel, you need to set each channel to Hold state and make trigger events to each channel.

On the E5071C, the trigger system involves states of the entire system and those of each of the channels. Since a trigger event is common to all channels, three system-wide states exist: Hold, Waiting for Trigger, and Measurement. On the other hand, two states exist for each channel: Idle and Startup. For a channel in an Idle state, measurement is not performed at all, while for a channel in Startup state, measurements are started in sequence after an event occurs. When all channels are in an Idle state, the E5071C is in Hold state when viewed as an entire system. If even one Startup state channel exists, the E5071C enters the Waiting for Trigger or Measurement state. Upon a transition from the Waiting for Trigger to the Measurement state, measurement is performed on channels put into Startup state starting with the channel with the smallest channel number.

Data Flow

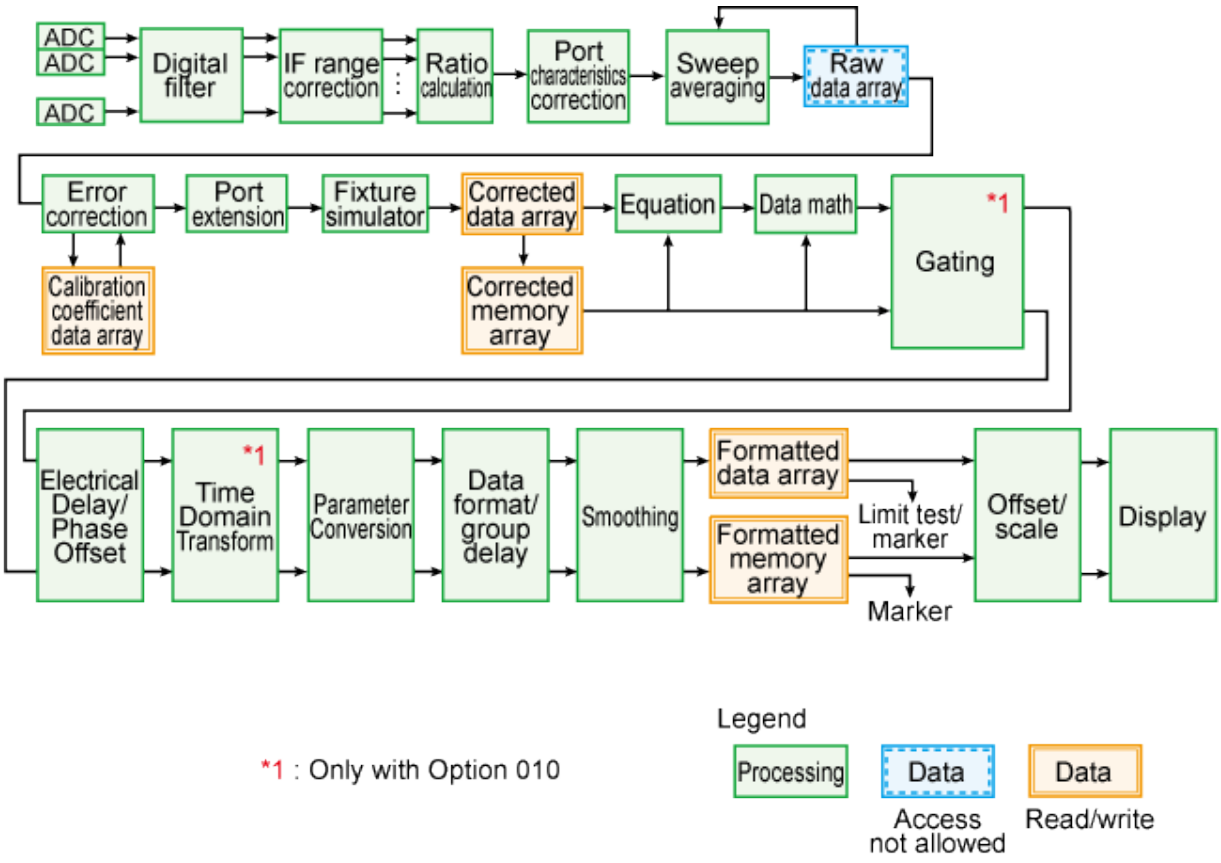
The data flow in the 8753ES is shown in 8753ES Data Flow while the flow in the E5071C is shown in E5071C Data Flow. As described in Reading Writing Data, the types of data you can read/write using the 8753ES differ from those you can read/write using the E5071C.

8753ES Data Flow



e5071c363

E5071C Data Flow



e5071c465

Reading / Writing Data

Types of data that can be handled by the 8753ES and E5071C are listed below.

Reading / Writing Data

Function	8753ES	E5071C
Data transfer format	• Intra-device binary • IEEE 32-bit floating point • IEEE 64-bit floating point • ASCII • PC-DOS 32-bit floating point	• IEEE 64-bit floating point • ASCII
Reading/Writing data	• Raw data array	• Formatted data

	• Calibration coefficient array (before interpolation) • Corrected data array • Formatted memory array • Memory trace • Calibration kit array data • Power meter calibration coefficient array (before interpolation)	array • Formatted memory array • Corrected data array • Corrected memory array • Power meter calibration coefficient array (after interpolation) • Calibration coefficient data array (after interpolation)
Reading data	• Pre-raw data (in Take4 mode) • Calibration coefficient array (after interpolation) • Power meter calibration coefficient array (after interpolation) • Entry area display • All lists in list format	

Screen Display and Marker Functions

The 8753ES allows up to four channels to be displayed on the screen. Up to five markers can be displayed on each channel. Also, one of the displayed markers can be designated as the reference marker. Each

channel also supports a fixed marker that can be established at a fixed position.

In contrast, the E5071C enables you to have all 36 channels displayed by opening up 36 separate windows on the screen. Sixteen traces can be displayed for each channel, and up to nine markers can be displayed for each trace. In addition to the markers displayed, you can also designate one marker as the reference marker. The E5071C, however, does not support fixed markers.

The 8753ES supports the Maximum, Minimum, Target value, and Bandwidth marker functions. The E5071C supports all these in addition to a Peak Search function. Using this function, you can determine whether or not to search for a positive or negative peak. In addition, the 8753ES has an additional function to search for the maximum or minimum bandwidth. While the 8753ES allows a target value or search tracking to be established only on the active marker, the E5071C allows a target value or search tracking to be established on all markers.

Math Operation Functions on Traces

On the 8753ES, each channel is provided with a memory trace. For this reason, math operations between the data trace and memory trace are supported: **Data / Memory** and **Data - Memory** On the E5071C, however, **Data * Memory** and **Data + Memory** are supported along with the division and subtraction operations described above.
The trace displays supported on the 8753ES are: "Data trace only", "Memory trace only", "Both memory trace and data trace" and "Data math only". The E5071C supports these functions in addition to "Both data math and memory trace"

Device Test Functions

The 8753ES supports limit test, ripple test and bandwidth test. The E5071C also supports these tests.

For the limit test on the 8753ES, a limit test table is provided for each channel with up to 18 segments are allowed in each table. In contrast, the E5071C allows a limit test table to be defined for each trace and up to 100 segments to be defined per table.

The 8753ES supports the pass/fail result for each channel, segment or measurement point along with the minimum/maximum measurement value for each segment as retrieved items in test results. On the other hand, the E5071C supports pass/fail results, measurement value, and upper/lower limit values for each measurement point as well as pass/fail results of the active trace for each channel.

Analytical Functions

Although the 8753ES does not support the fixture simulator function, the E5071C does. The fixture simulator supported by the E5071C include the balanced-unbalanced transformation function for analyzing balanced devices, and the matching circuit function.

Save/Recall

For storing data, the 8753ES is provided with an internal register, internal disk drive (floppy disk), and external disk drive (connected through the GPIB). In contrast, the E5071C provides an internal hard disk drive and allows external storage devices such as USB memory. While the 8753ES can save or recall the device setup, screen color settings, and test sequences, the E5071C is able to save or recall the instrument setup, segment sweep table, and limit line table as well as VBA projects. The 8753ES saves display data in JPEG format while the E5071C supports the Windows Bitmap (BMP) and Portable Network Graphics (PNG) format.

Test Sequence Program

Although the 8753ES supports the test sequence program, the E5071C provides an environment for developing VBA programs for automatic measurement.

Outputting to a Printer/Plotter

The 8753ES enables you to establish the print area covering an entire sheet or just a 1/4 sheet, and to define traces, the reference line, and colors of warning messages. In contrast, the E5071C supports only full-sheet output, and an on/off setting for highlighting the entire screen in connection with color setup.

The 8753ES supports parallel ports, serial ports, and GPIB as printer ports, although the E5071C supports only USB-connected printers.

GPIB Interface

While the 8753ES uses pass control to pass the controller information to an external PC or instrument, the E5071C does not support this function.

LAN Interface

Although the 8753ES does not support a LAN interface, support for LANs is standard on the E5071C.

Other Functions

The 8753ES is provided with Take4 mode, mixer measurement function, and harmonics measurement function (Option 002). The E5071C provides the mixer measurement function, and harmonics measurement function with option 008.

Code Migration Tools

This section provides useful information to migrate the automation system from 8753 to ENA.

- Code Conversion Assistant Editor
- ENA 8753 Code Translator (ENA8753cXL) Application

Other topics about Replacing 8753ES with E5071C

Code Conversion Assistant Editor

The code conversion assistant editor helps to convert 8753 GPIB commands to ENA commands. This software consists of a text editor (EmEditor) and a plug-in software, which works on a PC and helps to edit programs in text format. The editor is also useful for 872X analyzers as most of the 872X commands are the same as 8753's.

ENA 8753 Code Translator (ENA8753cXL) Application

The ENA8753cXL is a code emulation application that allows you to use the ENA with the test program that was written in language for 8753 series network analyzer. This application works on background of the ENA, translating incoming 8753 commands via GPIB in real time.

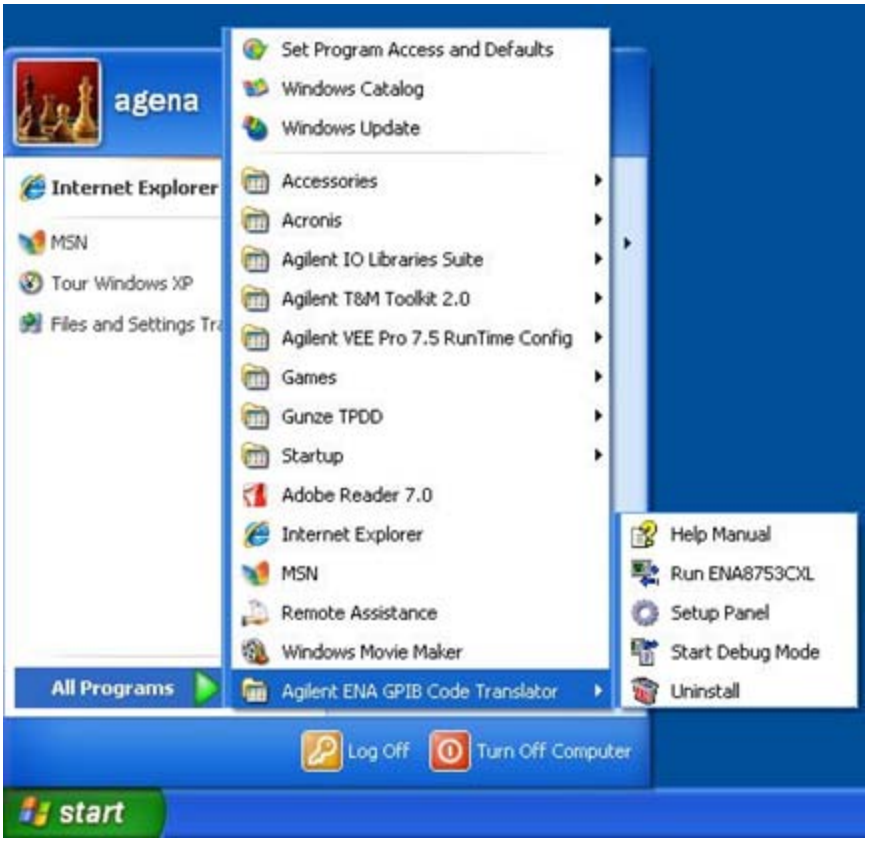
- The ENA8753cXL is designed to work with certain legacy Agilent Network Analyzer products. Not all products may be supported nor can all legacy commands be supported. ALWAYS REFER TO THE INFORMATION PROVIDED IN THE HELP FILE FOR ADDITIONAL INFORMATION REGARDING THE LIST OF SUPPORTED COMMANDS AND PRODUCTS.

On the HDD revision CN920 or above, the installer is pre-loaded on the ENA HDD image (**D:\Agilent\ENACXL\Agilent_ENA8753CXL.msi**). To install the ENA8753cXL application, follow the instruction below.

- If the HDD revision is less than CN920, contact your nearest Agilent Technologies sales representative.

1. Terminate the ENA application: Press **System** key, click **Service** > **Exit**
2. Double-click the Agilent_ENA8753CXL.msi to launch the installer
3. Follow the wizard to install the ENA8753cXL

After the installation is complete, the following files will be available from the Windows start menu. Refer to the "Help Manual" for more details on launching and using the ENA8753cXL.


agena
Internet Explorer
MSN
Tour Windows XP
Files and Settings Tr
All Programs
Set Program Access and Defaults
Windows Catalog
Windows Update
Accessories
Acronis
Agilent IO Libraries Suite
Agilent T&M Toolkit 2.0
Agilent VEE Pro 7.5 RunTime Config
Games
Gunze TPDD
Startup
Adobe Reader 7.0
Internet Explorer
MSN
Remote Assistance
Windows Movie Maker
Agilent ENA GPIB Code Translator
Help Manual
Run ENA8753CXL
Setup Panel
Start Debug Mode
Uninstall
Log Off
Turn Off Computer
start

Comparing Functions

Anaysis	8753ES	E5071C
Fixture simulator	Not available	Available

Calibration	8753ES	E5071C
Calibration kit	Available calibration kits: • 7 mm: 85031B • 3.5 mm: 85033C/D/E • N type (50 ohm): 85032B/E/F • N type (75 ohm): 85036B/E • 2.4 mm: 85056A/D • TRL 3.5 mm: 85052C User defined calibration kit	• 7 mm: 85031B, 85050C/D • 3.5 mm: 85033D/E, 85052D • N type (50 ohm): 85032B/F • N type (75 ohm): 85036B/E • TRL 3.5 mm: 85052C Up to 11 user defined calibration kits can be registered.

Calibration type	Not more than two ports		• Response calibration • Response and isolation calibration • Enhanced response calibration • S11 1-port calibration • S22 1-port calibration • Full 2-port calibration • TRL/LRM Calibration • Receiver calibration • Power meter calibration	• Response calibration (OPEN,SHORT, THRU) • Response and isolation calibration The response calibration includes isolation measurement as an option. • Enhanced response calibration • Full 1-port calibration • Full 2-port calibration • TRL/LRM Calibration • Power calibration • Receiver calibration • Vector-Mixer calibration • Scalar-Mixer calibration • Partial overwrite calibration • Unknown thru

	calibration • Adapter Removal/Insertion calibration

	four ports	Not available	• Full 3-port calibration (Opts. 430/435/480/485) • Full 4-port calibration (Opts. 430/435/480/485) • TRL/LRM Calibration • Power calibration • Receiver calibration • Vector-Mixer calibration • Scalar-Mixer calibration • Partial overwirte calibration • Simplified calibration • Simplified TRL calibration • Unknown thru calibration • Adapter Removal/Insertion calibration

Omission of the isolation measurement	Possible (Can be omitted by designating it using the softkeys on the front panel or sending a GPIB command)	Possible (Isolation measurement is performed as an option accompanying a response calibration and 2-, and 4-port calibration.)
Power meter calibration	Available	Available
Receiver calibration	Available	Available
Adapter removal	Available	Not available
Velocity factor	Definable	Definable
Characteristic impedance of the measurement system	Definable	Definable with fixture simulator
Port extension	Definable for ports 1 and 2 and inputs A and B	Definable for each port per channel
Selection between chop sweep and alternate sweep	Can be changed over.	Chop sweep for traces in a channel and alternate sweep between channels
Error correction	On/Off setting is definable.	Same

Interpolating the calibration coefficient		On/Off setting is definable.	Always On.
Defining the calibration kit	Calibration class	• S11A (OPEN) • S11B (SHORT) • S11C (LOAD) • S22A (OPEN) • S22B (SHORT) • S22C (LOAD) • Forward Transmission • Forward Match • Reverse Transmission • Reverse Match • Response • Response & Isolation • TRL Thru • TRL Reflect • TRL Line/Match	• OPEN • SHORT • LOAD • THRU (OPEN, SHORT, and LOAD are set for each port; THRU is set between ports) • TRL Thru • TRL Reflect • TRL Line/Match
	Class indication label	Editable	Editable

	Number of standards that can be registered in calibration classes		Up to seven types	Up to eight types
	Type of standard		OPEN, SHORT, LOAD, and THRU Arbitrary Impedance	OPEN, SHORT, LOAD, and THRU Arbitrary Impedance, None
	Standard parameters	Common to standards	• Offset delay • Offset loss • Offset characteristic impedance • Frequency range • Offset type (coaxial, waveguide)	Same
		Unique to OPEN	• C0, C1, C2, C3	Same
		Unique to SHORT	• Not available	L0, L1, L2, L3
		Unique to LOAD	Type of standard (fixed, sliding, offset)	Not available (the type of standard is treated as fixed load.)
		Unique to THRU	Not available	Unknown Thru
		Arbitrary Impedance	• Type of standard (fixed,	• Arbitrary

		sliding) • Arbitrary Impedance	Impedance
	Standard label	Editable	Same
ECal	Type of calibration	• Enhanced response calibration • S11 1-port calibration • S22 1-port calibration • Full 2-port calibration	• Enhanced response calibration • Response (thru) calibration • Full 1-port calibration • Full 2-port calibration • Full 3-port calibration (Opts. 430/435/480/485) • Full 4-port calibration (Opts. 430/435/480/485)
	Omission of isolation	Allowed	Allowed
	Manual THRU measurement	Allowed	Allowed
	Reading the frequency array	Allowed	Not allowed
	Reading the module	Allowed	Not allowed

information

Device Test		8753ES	E5071C
Limit test	Limit test table	• One per channel • Number of segments: Up to 18 per table	• One per trace • Number of segments: Up to 100 per table
	Offset	Definable	Definable
	Creating a table by using the GPIB command	Create a table by using two or more commands	Create a table by using one command
	Reading the results	• Per channel, Per segment • Pass/fail at each point, • Number of points per segment that failed the maximum/minimum, • Stimulus value at a failed point	• Pass/Fail of the active trace on each channel • Number of failed points • Stimulus value at a failed point
Ripple test		Available	Available
Bandwidth test		Available	Available

Display		8753ES	E5071C
Memory trace	Number	• One per channel	• One per trace
	Data math	• Data / Memory • Data - Memory	• Data / Memory • Data * Memory • Data - Memory • Data + Memory
	Display	• Data trace only. • Memory trace only. • Simultaneous display of memory traces and data traces. • Data math only.	• Data trace only. • Memory trace only. • Simultaneous display of memory traces and data traces. • Data math only. • Simultaneous display of data math and memory traces.
Frequency display		On/Off setting is definable for all channels.	On/off setting is definable channel by channel.
Graph layout		• Up to four graphs can be displayed.	• Windows are assigned to each channel; up to 36 windows can be displayed. • Up to 16 graphs can be displayed in each window.
Math between data traces		The results for channel 1/channel 2 can be displayed on the trace for channel 2.	Not available
Scale	Auto	Available	Available

scale		
Reference line	Both value and position can be specified.	Both value and position can be specified.
Scales per division	Definable	Definable
Number of divisions	Fixed at 10	Can be set in increments of 2 from 4 to 30.
Turning the softkey area On/Off	Available	Available
List display	Available	Not available
List display of Instrument State	Available	Not available
Color settings	Available	Available
Screen brightness setting	Available	Not available
Turning the LCD On/Off	Available	Available
Turning the update On/Off	Always updates.	Available

Macro	8753ES	E5071C
Creating macros	Uses a test sequence.	Uses VBA.
Automatic execution	Will execute if "AUTO" is given as the name of the sequence 6.	Only auto loading is possible.

Marker		8753ES	E5071C
Number of markers		Up to five markers per channel	Nine markers per trace (exclude reference marker)
Active marker		One marker per channel	One marker per trace
Delta marker (Reference marker)		Designates the marker displayed as the reference marker.	You can designate the reference marker independently from the markers displayed.
Marker coupling		Setting of Couple/Uncouple definable	Setting of Couple/Uncouple definable (coupling available only between markers on the same channel)
Marker move mode		Continuous/discrete (definable channel by channel)	Continuous/discrete (definable trace by trace)
Fixed marker		One marker definable per channel	Not available
Marker value display		On/Off setting for the marker value display definable per channel	All marker values for active traces are displayed on each channel. Display/No Display setting for the marker table displaying all marker values definable.
Marker search	Search function	• Maximum • Minimum • Target value • Bandwidth	• Maximum • Minimum • Target value • Peak
	Object of search	• Active marker	• Arbitrary marker (during remote

			control) • Active marker (when manipulating the front panel).
	Parameters for bandwidth search	• Bandwidth • Bandwidth center frequency • Q factor • Loss	• Bandwidth • Bandwidth center frequency • Upper and lower frequency of the bandwidth • Q factor • Loss
	Target value	Definable for an active marker	Definable for each marker
	Target transition type	Not definable	Setting of Positive/Negative transition definable (per marker)
	Search tracking	On/Off setting definable for the active marker (Markers other than the active marker are always turned off.)	On/Off setting definable for each marker
Marker reading set at a parameter		Values of the start, stop, center, span, CW frequency, and reference line in the sweep range	Values of the start, stop, center, span, and reference line in the sweep range
Reading the marker value (for remote control)		Possible for the active marker	Possible for any marker

Measurement		8753ES	E5071C
Reset		Can be executed by using the front panel and the GPIB command.	Can be executed by using the front panel, the GPIB command, and telnet.
Channel	Number of channels	4 channels	Up to 36 independent measurement channels.
	Coupling between channels	Channels 3 and 4 are auxiliary channels for channels 1 and 2 and subject to the same sweep conditions, etc. applicable to channels 1 and 2. Couple/Uncouple between channels 1 and 2 can be set up.	Channels are independent of each other.
Trace		A trace for each channel	Up to 16 traces.
Measurement parameter		• S-parameters • A • B • R	• S-parameters (Mixed-mode S-parameters) • Absolute

	• A/R • B/R • A/B • analog bus	value meas parameter • DC Measurement
S-parameter conversion	• Impedance (reflection and transmission) • admittance (reflection and transmission) • 1/S	Same
Display format (Data format)	• log magnitude • linear magnitude • phase • group delay • Smith chart • polar format • SWR • real • imaginary	• log magnitude • linear magnitude • phase • expanded phase • positive phase • group delay • Smith chart • polar format • SWR

			• real • imaginary
Test port output	Number of ports	• 2 ports	• 2 and 4 port options are available
	Frequency	• 30 kHz to 3 GHz (Std.) • 30 kHz to 6 GHz (Opt. 006)	The following frequency range is available depending on the option. • 9 kHz to 4.5/6.5/8.5 GHz • 100 kHz to 4.5/6.5/8.5 GHz • 300 kHz to 14/20 GHz
	Power level	• -85 to +10 dBm (Std.) • -85 to +8 dBm (Opts. 014/075)	• -55 to 10 dBm
	Characteristic Impedance	• 50 ohm (Std.) • 75 ohm (Opt. 075)	• 50 ohm
	Coupled/Uncoupled power levels	You can define Couple/Uncouple between	Traces are coupled on the same channel,

			channels and between ports.	not between channels.
	Power slope function		Available	Available
	Turning the output On/Off		Allowed	Available
	Power range	Items to be set up	Per port and per channel.	No Power Range (one range for entire power level.)
		Automatic setting	On/Off setting capability.	
		Setting range	Range setting with 10 dBm resolution is definable from between -15 and +10 dBm down to between -85 and -60 dBm.	
Sweep conditions	Number of points (except for the list frequency sweep operation)		3, 11, 21, 26, 51, 101, 201, 401, 801, 1601	Arbitrary value from 2 to 1601. (Can be set only when the Channel/trace is set to 1 ch/ 4 Tr 20001 points.)
	Sweep type		• linear sweep • log sweep • list sweep • power sweep • CW time sweep	• linear sweep • log sweep • segment sweep • power sweep
	List frequency	Number of	One for channels 1 and	One for each channel (16 in

	sweep	list tables	3 and one for channels 2 and 4, two in total	total)
		Number of segments per table	Up to 30	Up to 201
		Creation of a table using GPIB(SCPI) command	Creating a table using more than one command	Creating a table using a single command
		Number of points	1 to 1601 points per segment. Up to 1601 points in total.	2 to 200001 points per segment. Up to 20001 points in total.
		Stepped/swept mode	stepped mode and swept mode. Selectable only when list frequency sweep is performed	stepped mode and swept mode. Selectable also when list frequency sweep is not performed.
		Sweep of designated segments.	Allowed	Not allowed (Always sweeps all segments.)
	Sweep time		Automatic, Manual (definable from the shortest time to 24 hours)	Automatic, Manual (range of definable sweep depends on sweep condition)
	IF bandwidth		10 Hz, 30 Hz, 100 Hz, 300 Hz, 1 KHz, 3 kHz,	10 Hz, 15 Hz, 20 Hz, 30 Hz, 40 Hz, 50 Hz, 70 Hz, 100 Hz,

		3.7 kHz, 6 kHz	150 Hz, 200 Hz, 300 Hz, 400 Hz, 500 Hz, 700 Hz, 1 kHz, 1.5 kHz, 2 kHz, 3 kHz, 4 kHz, 5 kHz, 7 kHz, 10 kHz, 15 kHz, 20 kHz, 30 kHz, 40 kHz, 50 kHz, 70 kHz, 100 kHz, 150 kHz 200 kHz, 300 kHz, 400 kHz 500 kHz
	Averaging	Can be set from 1 to 999.	Can be set from 1 to 999.
Smoothing		The smoothing aperture can be set from 0.05% to 20%.	Same
Electrical delay	Transmission line	Compatible with coaxial cables and waveguides	Same
	Value	Any value from 0 to 10 seconds	Any value from -10 to +10 seconds
Phase offset		Any value from -360 degree to +360 degree.	Same
Setting sweep conditions for Couple/Unc		Channels 1 and 3 and channels 2 and 4 are always coupled. Channels 1 and 2 can be set	Traces in the same channel are coupled; traces in different channels are

ouple	either at Couple or Uncouple.	not coupled.
Trigger mode	• Continuous • Single • Hold • Specified number of sweeps	• Continuous • Single • Hold
Trigger source	• Internal • External (per sweep, per point) • Manual (per point)	• Internal • External (per sweep) • bus • manual (per sweep)
External trigger line	Can be set to High or Low	Same

Other		8753ES	E5071C
Time domain transformation		Available	Available
Take4 mode		Available	Not available
Mixer measurement function		Available	Available (Option 008)
Harmonics measurement		Available (Opt. 002)	Available (Option 008)
Key manipulation-related GPIB command		GPIB commands that perform the same processing as do the front panel key and softkey manipulation.	Not available
LAN	interface	Not available	• 10 Base-T • 100 Base-TX
	protocol	Not available	• TCP/IP
	function	Not available	• Telnet • SICL-LAN
GPIB	Debug mode	Can be turned On/Off	Not available
	pass control	Available	Not available
	GPIB address setup	The addresses of the main body, controller, external disk, LO control, power meter, printer, and plotter can be specified by using the GPIB command and from the front panel.	The address of the main body can be specified from the front panel. The addresses of the power meter can be specified by using the GPIB command and from the front panel.

	Parallel port	Can be used as a printer or GPIO connector.	Not available.

Printer	8753ES	E5071C
Print range	Entire sheet, 1/4 of a sheet	Entire sheet
Color settings	Settings definable for the colors of a trace, reference line, text, and warning messages	Highlighting/No highlighting of all elements
Line settings	Possible	Always a solid line.
Printer port	• Parallel port • Serial port • GPIB	• USB

Read / Write	8753ES	E5071C
Transfer format	• Intra-device binary • IEEE 32-bit floating point • IEEE 64-bit floating point • ASCII • PC-DOS 32-bit floating point	• IEEE 32-bit floating point • IEEE 64-bit floating point • ASCII • PC-DOS 32-bit floating point
Object to be read/written	• Raw data array • Calibration coefficient data array (before interpolating) • Corrected data array • Formatted memory array • Memory trace Calibration kit array data • Power meter calibration coefficient array (before interpolating)	• Calibration coefficient data array (after interpolating) • Corrected data array • Corrected memory array • Formatted data array • Formatted memory array • Power meter calibration coefficient array (after interpolating)
Object to be read only	• Pre-raw data (in Take4 mode). • Calibration coefficient array (after interpolating). • Power meter calibration coefficient array (after interpolating).	
Reading data for a designated point or scope	Allowed only for data traces	Not allowed

Save/Recall	8753ES	E5071C
Storage	• Internal registers • Internal disk (floppy disk) • External disk (connected through GPIB)	• Internal hard disk • External storage/memory device through USB.
Storage initialization	Possible	Available with mouse
Save only	LCD screen image	• Formatted data array • LCD screen image
Object to be saved/recalled	• Instrument State (You can simultaneously save formatted data array for the active trace, raw data array, corrected data array, and LCD images.) • Color settings • Test sequence	• Instrument State (You can simultaneously save formatted data array for the active trace and calibration data.) • Segment sweep table • Limit line table • VBA project
File formats	• Binary form • ASCII form	• Binary form
Format for saving the screen display	• JPEG format	• Windows Bitmap format • PNG format
File manipulation	• Reading file information • Deleting a file	• Reading file information • Deleting a file or folder • Creating a folder • Copying a file

Status Report	8753ES	E5071C
Register types	• Status byte register. • Service request enable register. • Standard event status register. • Standard event status register B and enable register.	• Status byte register. • Service request validation register. • Standard event status register. • Operation status register and validation register. • Questionable status register and validation register.
For limit test	Included in the standard event status register B and validation register	• Questionable limit status register. • Questionable limit channel status register. • Questionable limit extra channel status register. • Questionable limit channel extra status register. • Questionable ripple limit status register. • Questionable ripple limit channel status register. • Questionable ripple limit extra channel status register. • Questionable ripple limit channel extra status register. • Questionable bandwidth limit status register. • Questionable bandwidth limit channel status register. • Questionable bandwidth

		limit extra channel status register. • Questionable bandwidth limit channel extra status register.

System	8753ES	E5071C
Self-test	Can be executed by using the front panel and SCPI commands.	Can be executed by using the front panel.
Internal clock	Available	Available
Beep sound	On/Off setting definable for completion of actions, warnings, and limit test failures	On/Off setting definable for completion of actions and warnings
Measuring instrument mode	Can be selected from a • standard network analyzer • external signal source (automatic) • external signal source (manual) • tuned receiver.	Always a network analyzer

Using Windows

- Windows Consideration
- Change Date/Time Settings
- User Account and Password
- On-Screen Keyboard
- Configuring Network
- Accessing Hard Disk of E5071C from External PC
- Windows Firewall
- Disabling-Enabling USB Mass Storage Device
- Connecting External Accessories

Windows Consideration

- Using USB
- Plug & Play Stability and Security
- LAN Connections
- Single and Double Click option
- Printing

Other topics about Using Windows

Using USB

The ENA has six USB ports for connecting devices: two in the front panel and four on the rear panel. The main advantages of USB are instant connect and disconnect, and faster data transfer speed. Electronic Calibration modules are now available with USB connections.

The first time you plug a device into a USB port there is some wait time. Windows reports it is identifying the hardware, then searching for the correct driver, then installing the driver (if it was found).

Connecting that same device back into that same port later is quick and easy, but if you move the device to a different USB port, you will have to wait through the hardware ID and driver search again.

- Certain USB devices (such as ECAL modules) require you to be logged on with Administrator's privileges, the first time you plug them into the ENA. This must be done for each serial number. Click **Next** to choose the default settings when installing new USB devices.

Plug & Play Stability and Security

Plug & Play capabilities provide both a stable and secure operating environment. You may notice also that it greatly reduces the number of required reboots.

LAN Connections

Windows supports DHCP and fixed IP addressing. Also, instant connect and disconnect of the LAN cable, as well as a visual indicator of LAN status in system tray area, makes LAN connections more intuitive. In addition, the Hardware Wizard helps users with system hardware configuration.

Single and Double Click option

By default, Windows allows a double-click method of launching icons. To revert to single-clicking, see the Windows help.

Printing

Adding a printer should be done as Windows operation. See Connecting Printer.

Change Date/Time Settings

The E5071C has the built-in clock that keeps track of the date and time. This clock is used for the following functions.

- To display the current date and time in the instrument status bar at the lower part of the screen
- To write date and time information when saving internal data or a VBA program
    - The **Date & Time** of E5071C clock can be changed by double-clicking the clock on the bottom right of the taskbar.

Other topics about Using Windows

User Account and Password

- To Add Password
- To Add New User

Other topics about Using Windows

To Add Password

At the factory default setting, the password is not set and Auto Log On is activated.
If you add a password, Auto Log On is not available.

1. Open **User Accounts** in Control Panel.
2. Click **Change an Account**.
3. Select your account name ("agena"is the default setting).
4. Click **Create a password**.
5. Enter you password and password hint.
6. Click **Create Password**.

To require users to press CTRL+ALT+DELETE before logging on

For added security, you can require users to press CTRL+ALT+DELETE before logging on.

1. Open **User Accounts** in Control Panel.
2. Click **Change the way users log on or off**.
3. Check out **Use the Welcome screen**.
4. Click **Apply Options**.
5. Log off Windows.
    - ENA must be removed from the domain for the procedure to work. When shipped, ENA is configured as a member of a workgroup. If the configuration of ENA is later changed so as to be a member of domain, then name/password must always be entered at startup. This is a security requirement of Windows.
    - If ENA is assigned to a domain, then **Change the way users log on or off** will not be available in **User Accounts** dialogue box.
    - At the log on, you can log in as "agena" or "administrator" (they are default setting). The password of "administrator" is "e507xa". "agena" has no password.

To Add New User

Accounts must be the computer administrator if you make measurements.

1. Open **User Accounts** in Control Panel.
2. Click **Create a new account**.
3. Type the new account name, then click **Next**.
4. Select **Computer Administrator**.
5. Click **Create Account**.

On-Screen Keyboard

- On-Screen Keyboard overview
- To open Windows On-Screen Keyboard

Other topics about Using Windows

On-Screen Keyboard overview

On-Screen Keyboard is a utility provided by Windows that displays a virtual keyboard on the screen. It allows the users of the E5071C to input characters without the need of a keyboard.

- The E5071C does not requires an external keyboard for its operation. Users can input characters using an on-screen keyboard in-built with the E5071C firmware.

e5071c060

To open Windows On-Screen Keyboard

To open On-Screen Keyboard: click **Start** > **All Programs** > **Accessories** > **Accessibility** > **On-Screen Keyboard**.

Configuring Network

- When you use the E5071C by connecting it to your LAN, consult your network administrator and make the setting of the LAN correctly.

This section describes how to set the following basic items necessary to connect the E5071C to the LAN (Local Area Network).

- Enabling/Disabling network
- Check/Reset IP Address
- Setting IP address
- Specifying computer name

If you need detail network settings, consult your network administrator and perform operation in the same way as the Windows PC.

Other topics about Using Windows

Enabling / Disabling Network

You can enable/disable the network connection function of the E5071C.

To enable / disable the network connection function

1. Use the LAN cable to connect the E5071C to the LAN.
2. Press **System** > **Misc Setup** > **Network Setup** > **Network Configuration** to open **Network Connections** window.
3. Perform either of the following:

- When switching from disable to enable:

Double-click the Local Area Connection icon in the **Network Connections** window to enable the network connection function.

- When switching from enable to disable:

Double-click the Local Area Connection icon in the **Network Connections** window. The **Local Area Connection Status** screen appears. Click the **Disable** button to disable the network connection function.

4. Close **Network Connections** window.

Check / Reset IP Address

The IP address of the E5071C can be checked/reset by **System** > **Misc Setup** > **Network Setup** > **LAN Dialog**.

Setting IP address

Follow these steps to set the IP address:

1. Press **System** > **Misc Setup** > **Network Setup** > **Network Configuration**
2. Double-click the Local Area Connection icon in the **Network Connections** window. The **Local Area Connection Status** screen appears.
3. Click **Properties**. The **Local Area Connection Properties** screen appears.
4. Select (highlight) **Internet Protocol (TCP/IP),** and then click **Properties**.
5. The **Internet Protocol (TCP/IP) Properties** appears. Click (select) **Use the following IP address** and then enter the **IP address, subnet mask** and **gateway address**.
6. If the IP address can be obtained automatically (if the DHCP server can be used), click (select) **Obtain an IP address automatically**.
7. In **Internet Protocol (TCP/IP) Properties**, click **OK**.
8. In **Local Area Connection Properties**, click **OK**.
9. In **Local Area Connection Status**, click **Close**.
10. Close **Network Connections** window.

Specifying Computer Name

Follow these steps to specify the computer name:

1. Press **System** > **Misc Setup** > **Network Setup** > **Network Identification**.
2. Enter the desired computer name in Computer description in **Computer Name** tab.
3. Click **OK**.

Accessing Hard Disk of E5071C from External PC

If you connect the E5071C to LAN, you can access the hard disk (D drive) in the E5071C as a network drive from an external PC connected to the same LAN.

- Enabling the access from an external PC
- Accessing hard disk of E5071C from external PC
- E5071C Hard Disk Drive

Other topics about Using Windows

Enabling the access from an external PC

This section shows the procedure to enable the access from the external PC.

1. Press **Save/Recall** > **Explorer** which opens the explorer.
2. Select (highlight) **USER (D:)** and then click **Properties** in the File menu. The USERS(D:) Properties dialog box appears.
3. Select the **Sharing** tab.
4. Click **New Share.** The New Share dialog box appears.
5. Enter the share name (name used when accessed from the external PC) in the **Share Name** box and click **OK** .
6. In the USERS(D:) Properties dialog box, click the **OK** button.

Accessing hard disk of E5071C from external PC

This section describes the procedure to connect to the hard disk (D drive) in the E5071C to which access has been made possible according to the procedure described in Enabling the access form the external PC from the external PC, taking Windows as an example.

- For information on connection, see your PC's operation manual.

1. Right-click start menu, then select **Explorer** to start the explorer.
2. From the Explorer's menu, click **Tools** > **Map Network Drive..** The Map Network Drive dialog box appears.
3. Select an appropriate drive, enter **\\C_NAME\S_NAME** as the network path and then click the **OK** button.
4. The dialog box to enter the user name and the password appears. Enter an appropriate user name and password and then click the **OK** button.

5. **C_NAME** in the network path is the computer name of the E5071C and **S_NAME** is the share name of the D drive. For information on how to set the computer name, refer to Specifying computer name; for information on how to set the share name, refer to Enabling the access from an external PC.
6. The user name and password differ depending on the setting made when enabling access from the external PC. When you have set them according to Enabling the access from an external PC you can make connection using the user name, **agena**, without the password.

E5071C Hard Disk Drive

The E5071C Hard disk contains several memory partitions. The following table explains the different partitions of the E5071C.

Drive	Description
CNxxx (C:)	The system drive is replaced with the original image when system recovery is executed.
USER (D:)	User can keep their files, in this drive. This drive is not replaced even when system recovery is executed.
RECOVERY (E:)	This drive contains license information and back up data for syscal. Also,user can keep their files, in this drive. The drive is not replaced even when system recovery is executed.
SWAP (F:)	This folder is for system recovery. User cannot use this drive.

Windows Firewall

To turn Windows Firewall on or off

1. Open Windows Firewall.
2. On the General tab, click one of the following:
    - **On (recommended)**. This is the setting that you should normally use.
    - **Off (not recommended)**. Turning off Windows Firewall might make your computer (and your network, if you have one) more vulnerable to damage from viruses.
    - To open Windows Firewall, open Control Panel, and then double-click **Windows Firewall**.

Other topics about Using Windows

Disabling-Enabling USB Mass Storage Device

You can disable/enable any USB-compatible external mass storage devices in order to ensure confidentiality or for other reasons.

- To disable a USB Mass Storage Device
- To enable a USB Mass Storage Device
- To disable write access to USB Mass Storage Device

Other topics about Using Windows

To disable a USB Mass Storage Device

1. Press **Save/Recall** key.
2. Press **Explorer...**.and navigate to **D:\Agilent\Service**.
3. Double-click **DisableUsbStorage.exe**
4. Click **OK** in the SUCCEEDED message window that appears. If any USB mass storage device is connected to the E5071C under this condition, the Hardware Wizard will start, but the USB mass storage device will not work.

To enable a USB Mass Storage Device

1. Press **Save/Recall** key.
2. Press **Explorer...**.
3. Double-click **EnableUsbStorage.exe** from **D:\Agilent\Service**.
4. Click **OK** in the SUCCEEDED message window that appears.
5. If you do not want any USB mass storage device to ever be enabled at any time, delete **EnableUsbStorage.exe** from the E5071C after **DisableUsbStorage.exe** has been completed. These two programs will not be recovered automatically by applying the firmware update or other such action. Before deleting any of these programs, you should make a backup copy to a recording medium such as a USB memory and store it separately.
6. If the program fails to run, it is possible that you have not logged in as a user in the Administrators Group. When you want to execute any of the above programs, make sure to log in as a user in the Administrators Group.

To disable write access to USB Mass Storage Device

1. Press **Save/Recall** key.
2. Press **Explorer...**.
3. Double-click **ReadOnlyUsbStorage.exe** from **D:\Agilent\Service**.
4. Click **OK** in the SUCCEEDED message window that appears.
5. The **ReadOnlyUsbStorage.exe** does not allows you to save the file into USB memory from ENA HDD.

Connecting External Accessories

The E5071C can be connected to external input/output devices such as printer, monitor, keyboard and mouse using any of the serial or USB ports provided in the rear or the front panel.

- Connecting a Printer
- Connecting a Mouse
- Connecting a Monitor
- Connecting a Keyboard

Other topics about Using Windows

Connecting a Printer

A printer can be connected to the E5071C using the Universal Serial Bus (USB) ports provided in the front or rear panel.

Connecting a Mouse

A mouse can be connected to the E5071C using the Mini-DIN Keyboard Port provided in the rear panel of the E5071C, or through the Universal Serial Bus (USB) ports provided in the front or rear panel.

Connecting a Monitor

A monitor can be connected to the E5071C using the External Monitor Output Terminal (Video) located in the rear panel of the E5071C.

- An external monitor needs to be connected to the analyzer and turned on before the analyzer is turned on so that the analyzer recognizes the monitor properly.

Connecting a Keyboard

A keyboard can be connected to the E5071C using the Mini-DIN Keyboard Port provided in the rear panel of the E5071C, or through the Universal Serial Bus (USB) ports provided in the front or rear panel.

Programming

- Remote Control
- Built-in VBA Programming
- TDR Remote Control (Option TDR)
- Command Reference
- TDR Command Reference (Option TDR)

Remote Control

Remote Control

- Overview
- Setting up Analyzer
- Performing Calibration
- Making Measurement
- Reading-Writing Measurement Data
- Analyzing Data
- Limit Test
- Saving and Recalling
- Communication with External Devices
- Status Reporting System
- Working with Automatic Test System
- Sample Programs

Other topics about Programming

Overview

Overview

- Types of remote control system
- GPIB remote control system
- LAN remote control system
- USB Remote Control System
- Sending SCPI command messages
- LXI

Types of remote control system

Depending on the system controller and the interface, you can configure 4 types of remote control system as shown in the table below.

System controller	Interface	Overview
External controller (external computer such as PC and workstation)	GPIB (talker/listener mode)	System to control the E5071C and other devices connected via GPIB from the external controller. For more information, refer to GPIB remote control system.
	LAN	System to control the E5071C and other devices connected via LAN from the external controller. For more information, refer to LAN remote control system.
	USB	System to control the E5071C and other devices connected via USB from the external controller. For more information, refer to USB Remote Control System.
E5071C		System to control the E5071C itself using built-in E5071C VBA.
	GPIB (system controller mode)	System to control the E5071C itself and external devices connected via GPIB using built-in E5071C VBA.

Other topics about Overview

GPIB remote control system

- About GPIB
- System Configuration

Other topics about Overview

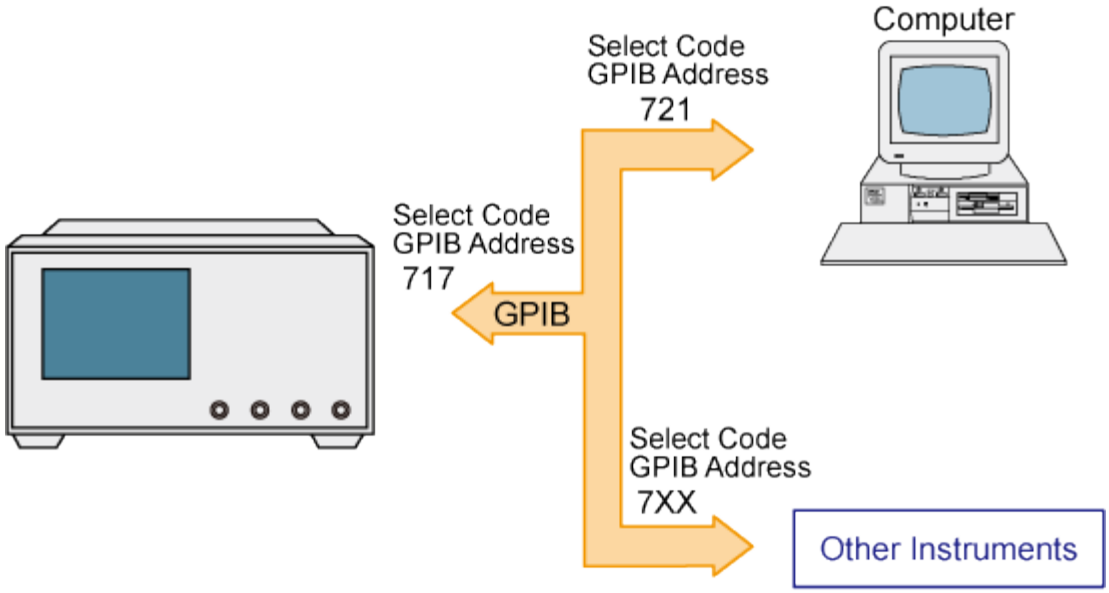
About GPIB

GPIB (General Purpose Interface Bus) is an interface standard for connecting computers and peripherals, which supports the following international standards: IEEE 488.1, IEC-625, IEEE 488.2, and JIS-C1901. The GPIB interface allows you to control the Agilent E5071C from an external computer. The computer sends commands and instructions to the E5071C and receives data sent from the E5071C via GPIB.

System Configuration

Use GPIB cables to connect between the E5071C, the external controller (computer), and peripherals. The following figure shows the overview of the system configuration of the GPIB remote control system.

Configuration of the GPIB remote control system



e5071c337

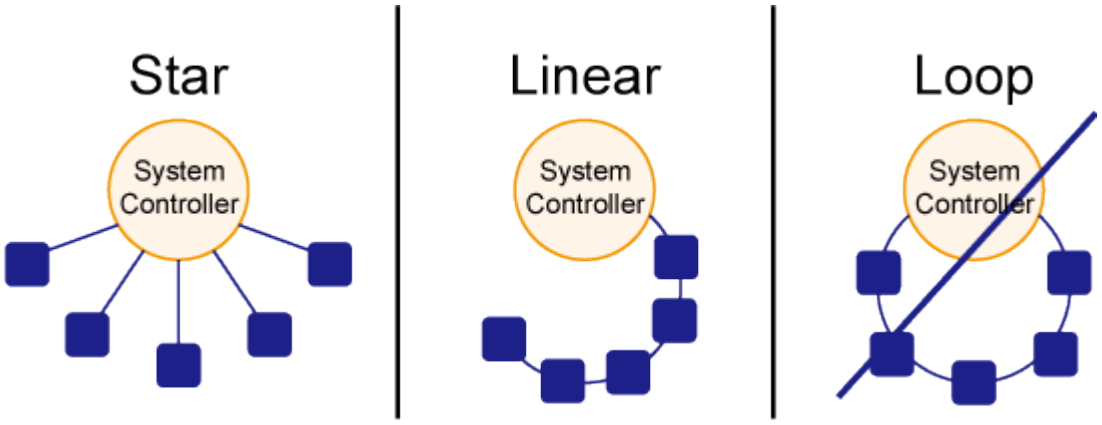
- While the E5071C is turned off, the SRQ status of the E5071C is active. To prevent an incorrect operation on the SRQ of the GPIB remote control system, disconnect the E5071C from the system when the E5071C is turned off.

Required Equipment

- E5071C
- External controller (PC or workstation that can be connected to LAN and Agilent I/O Library is installed into)
- Other devices (other instruments and/or peripherals that serve your purpose)
- GPIB cables

Scale of system you can construct

- You can connect up to 15 devices in a single GPIB system.
- The length of cables to connect between devices must be 4 m or less. The total length of connecting cables in a single GPIB system must be 2 m × the number of connected devices (including the controller) or less. You cannot construct the system in which the total cable length exceeds 20 m.
- The number of connectors connected to an individual device must be 4 or less. If you connect 5 or more connectors, excessive force is applied to the connector part, which may result in failure.
- You can choose the device connection topology from star, linear, and combined. Loop connection is not supported.



e5071c305

Device selector

The device selector is a unique value assigned to each device that is used by the controller to select the control target (to send/receive messages) among devices connected on the GPIB remote control system.

The device selector consists of a select code (usually, 7) and a GPIB address. For example, when the select code is 7 and the GPIB address is 17, the device selector is 717. The select code must be set for each system. The GPIB address must be set to a unique value for each device,

which is used to identify devices on the same system. In the description and sample programs in this manual, it is assumed that the device selector is set to 717.

Setting the GPIB address of E5071C

To set the GPIB address for talker/listener mode, See Setting_talker_listener_GPIB_address_of_E5071C.

LAN remote control system

- Overview
- System Configuration
- Required Equipment
- Control over SICL-LAN Server
- Control using C or Visual Basic
- Control using Agilent VEE
- Control with Telnet Server

Other topics about Overview

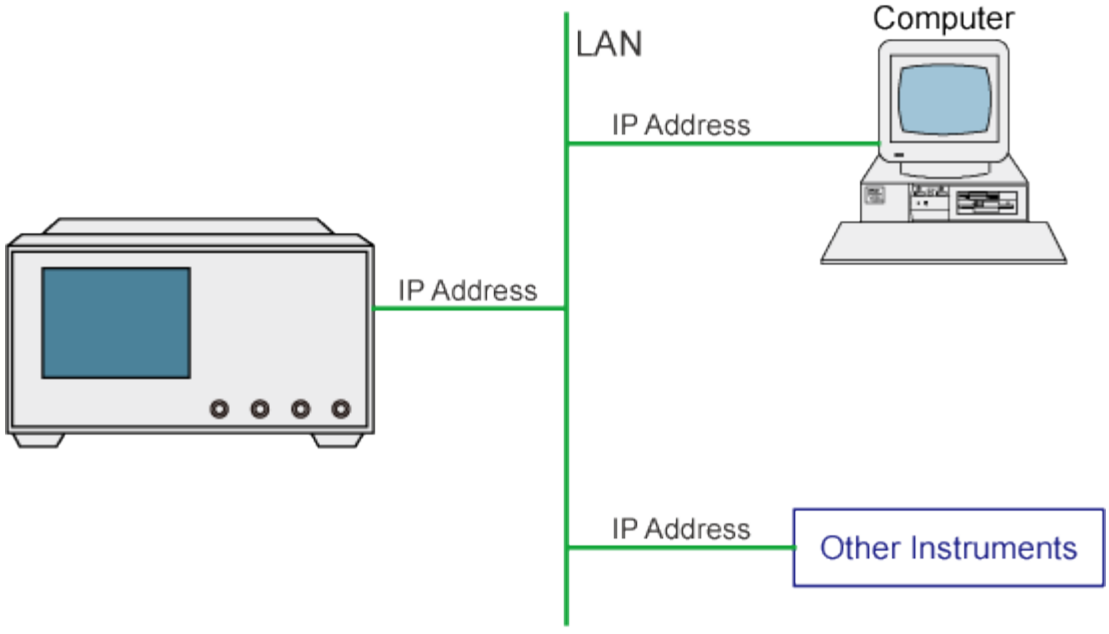
Overview

The LAN (Local Area Network) remote control system provides two methods: controlling the E5071C using the SICL-LAN server and controlling the E5071C using the telnet server.

System Configuration

Use a LAN cable to connect between the E5071C and the external controller (computer). The following figure shows the overview of the system configuration of the LAN remote control system.

Configuration of the LAN remote control system



e5071c338

Required Equipment

- E5071C
- External controller (PC or workstation that can be connected to LAN)
- Other devices (other instruments and/or peripherals that serve your purpose)
- LAN cables

Control over SICL-LAN Server

In the control system using the SICL-LAN server, communication between the external controller (client) and the E5071C (server) is performed using the SICL-LAN protocol. Communication is performed using SICL (Standard Instrument Control Library). You can control the E5071C by programming using SICL or VISA with the C language in the UNIX environment, or Visual C++, Visual Basic, or VEE in the Windows environment.

Preparing the E5071C

To communicate with the external controller, follow these steps to turn on the SICL-LAN server of the E5071C in advance.

1. Turn on the SICL-LAN server of the E5071C.

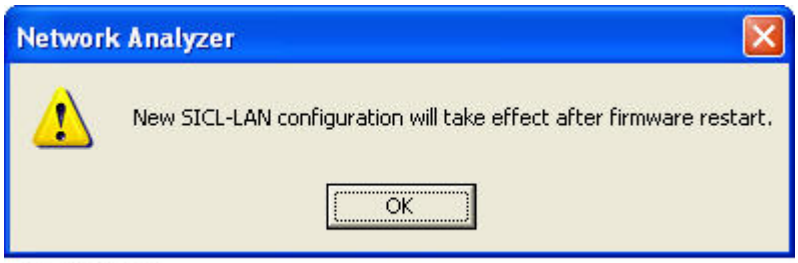
   System > **Misc Setup** > **Network Setup** > **SICL-LAN Server [ON]**

   2. When the SICL-LAN server is turned ON for the first time, the windows firewall setting dialog box appears. To use the SICL-LAN server, select **Unblock** and click **OK**.

2. Set the GPIB address of the E5071C for control with the SICL-LAN server. "XX" represents an address number.

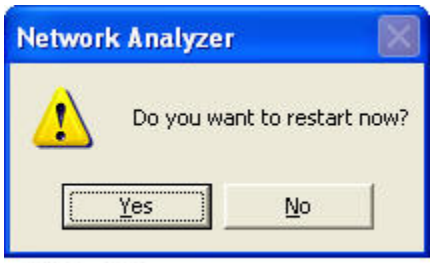
   System > **Misc Setup** > **Network Setup** > **SICL-LAN Address [XX]**

3. By default, the SICL-LAN Address does not changes until the firmware of E5071C is restarted.



e5071c137

4. On pressing any key, a message appears for restarting the firmware. Click **Yes** to restart the firmware.



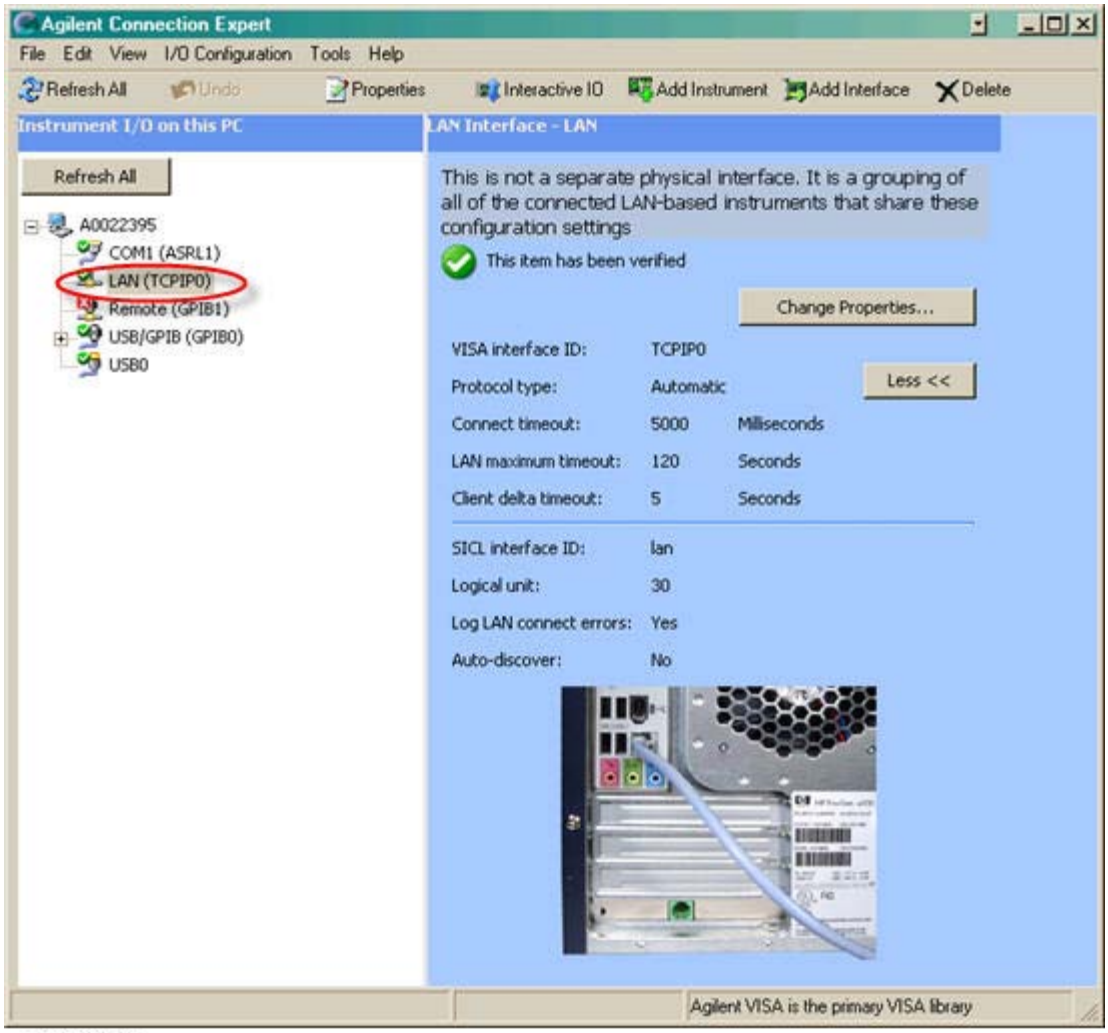
e5071c138

Preparing the external controller

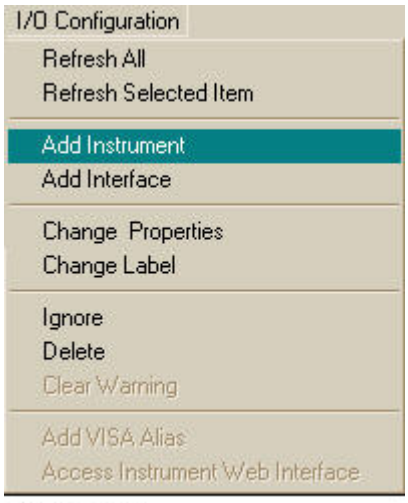
In order to establish communication to the E5071C using the TCP/IP protocol, you need to set the I/O interface of the external controller in advance. This section shows the setting procedure when using the external controller in the Windows environment.

- You must install the Agilent I/O Libraries on your PC in advance. Use Agilent I/O Libraries Suite 14.2 or later.

1. From your PC's Start menu, click **Program** > **Agilent I/O Libraries Suite** > A**gilent Connection Expert** to open the Agilent Connection Expert setting screen.
2. In the Agilent Connection Expert setting screen, select **LAN(TCPIP0)** in the **Instrument I/O on this PC** frame, and then click **I/O Configuration** > **Add Instrument.**

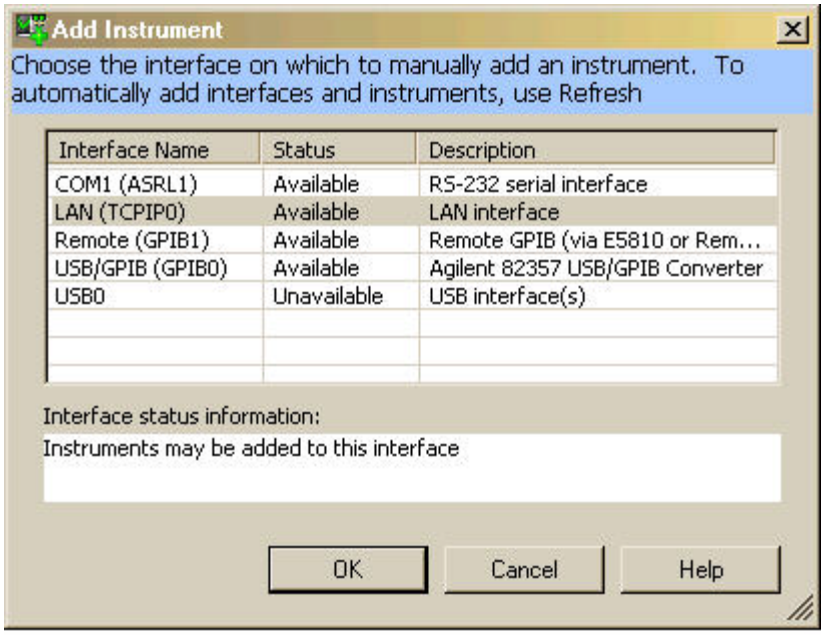


e5071c139



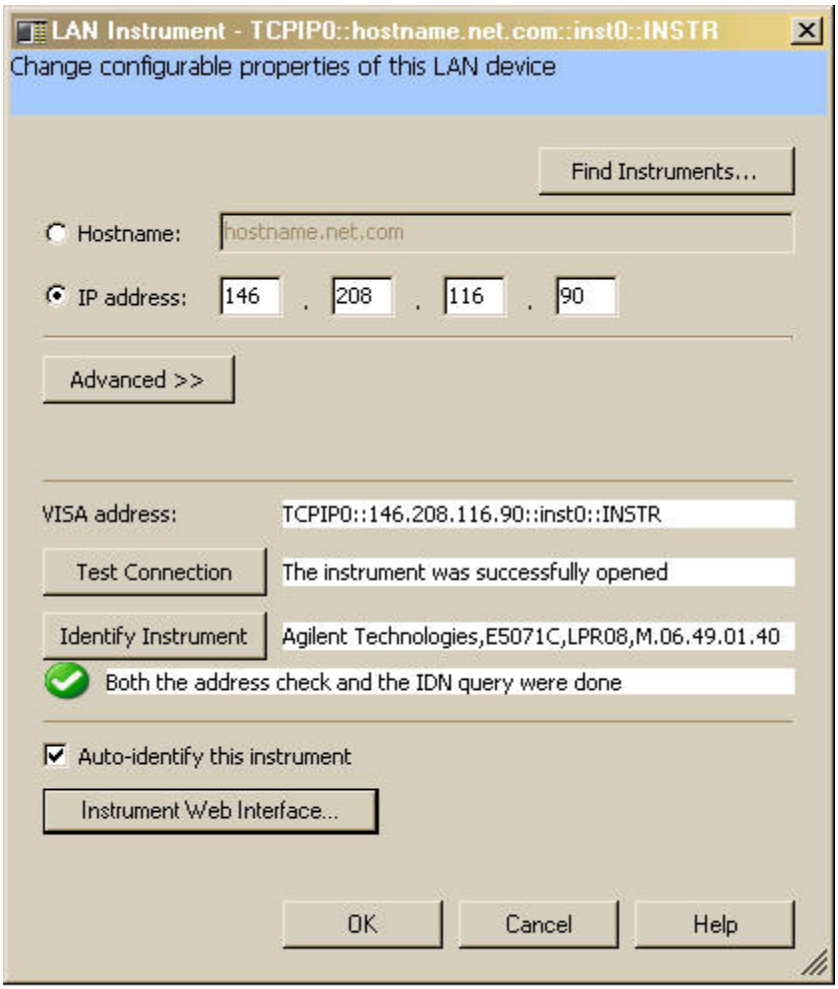
e5071c140

3. In the **Add Instrument** screen, select **LAN** (if it is not selected), and then click **OK**.



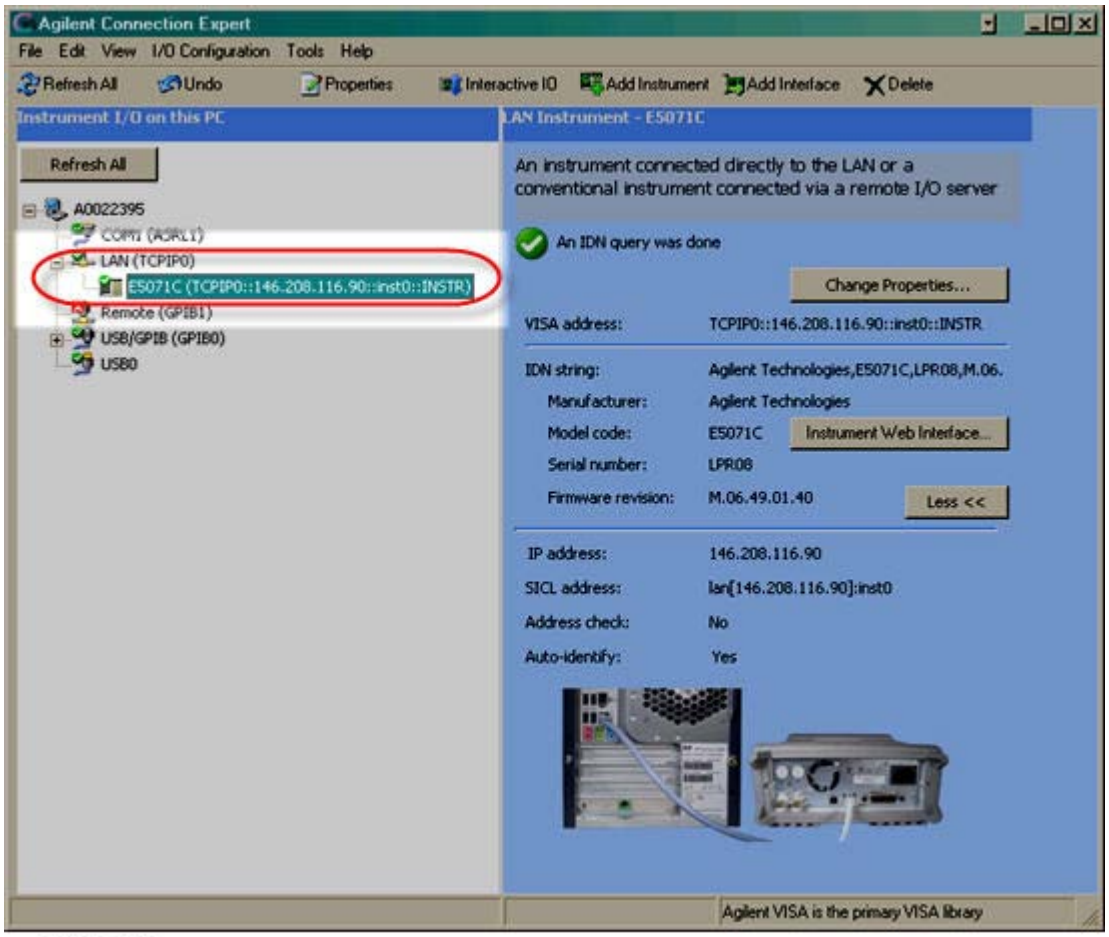
e5071c141

4. In the LAN Instrument Properties screen, set up the IP address of the E5071C and click **OK**. You can change settings as necessary. For details, refer to the Agilent I/O Libraries Suite documentation.



e5071c142

5. In the Agilent Connection Expert screen, check that the E5071C has been added under **LAN(TCPIP0)** in the **Instrument I/O on this PC** frame.



e5071c143

Control using C or Visual Basic

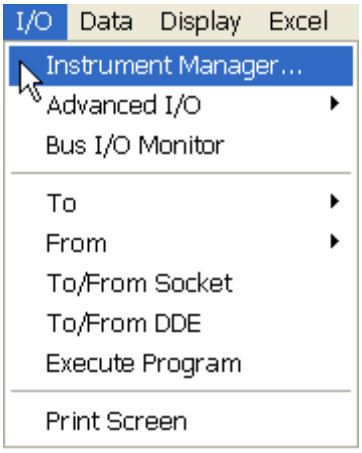
You can control the E5071C by programming using SICL with the C language in the UNIX environment, or Visual C++ or Visual Basic in the Windows environment.

Control using Agilent VEE

Agilent VEE allows you to control the E5071C via the I/O interface. The following example shows how to control the E5071C that is set as follows: the address of the SICL-LAN server is 17 and the IP address is 146.208.116.90.

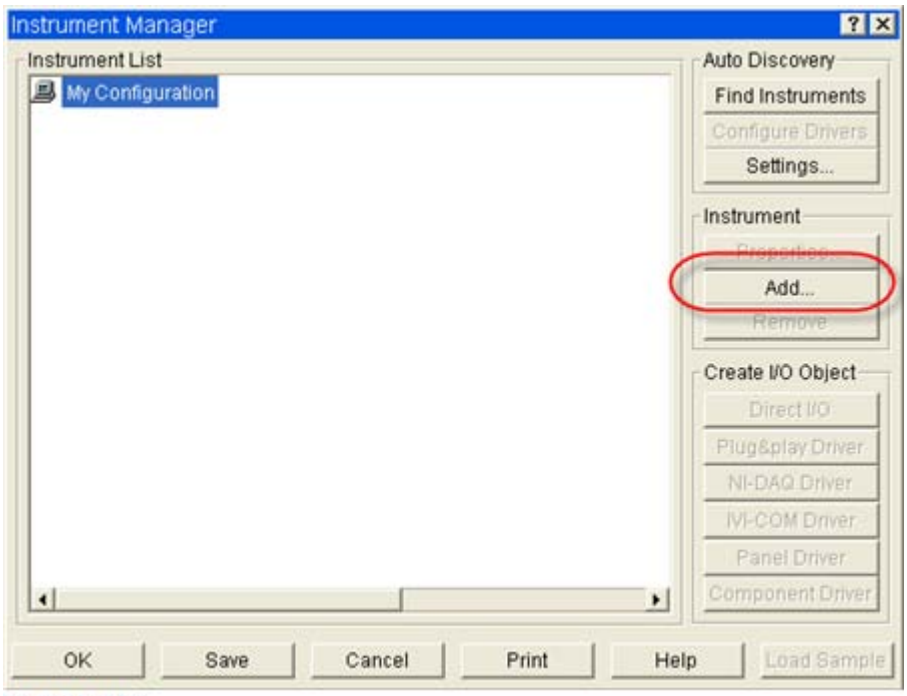
- When using Agilent VEE for PC, use Agilent VEE Pro 7.5 for Windows or later.

1. On the Agilent VEE's **I/O** menu, click **Instrument Manager**



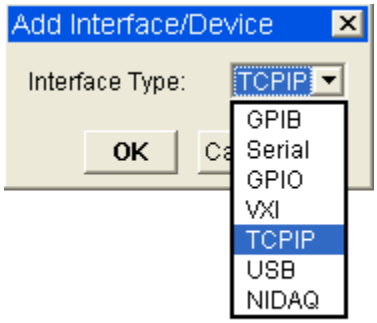
e5071c146

2. In **Instrument Manager**, click **Add...**



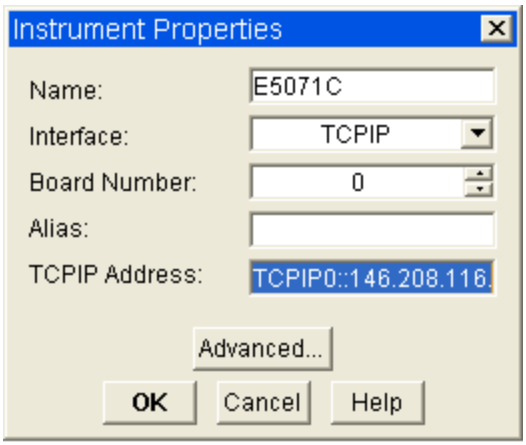
e5071c147

3. A new windows appears for the selection of Interface Type. Select **TCPIP** and click **OK**.



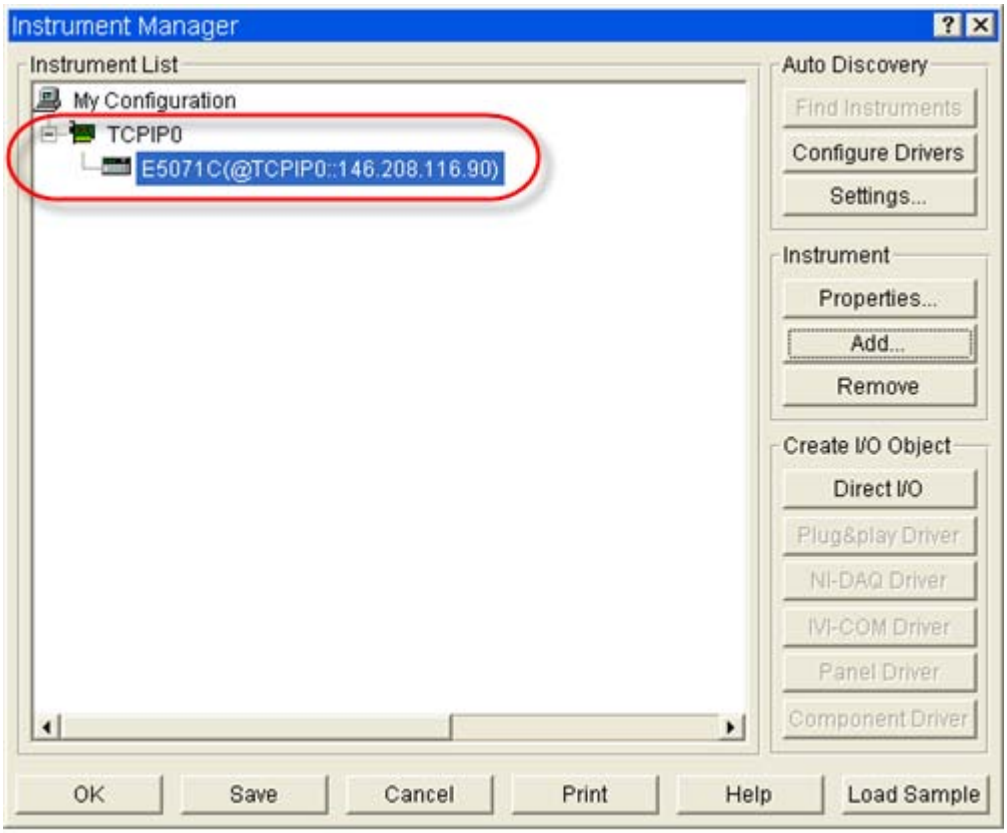
e5071c148

4. In **Instrument Properties**, type any name for the Instrument in Name (for example: ENA or E5071C), and add TCPIP0::<IP Address> in the **TCIP Address,** where <IP Address> is the IP address for E5071C. For example, if the IP address for E5071C is 146.208.116.90, then the value for **TCPIP Address** would be TCPIP0::146.208.116.90. Click **OK** after entering all the parameters.



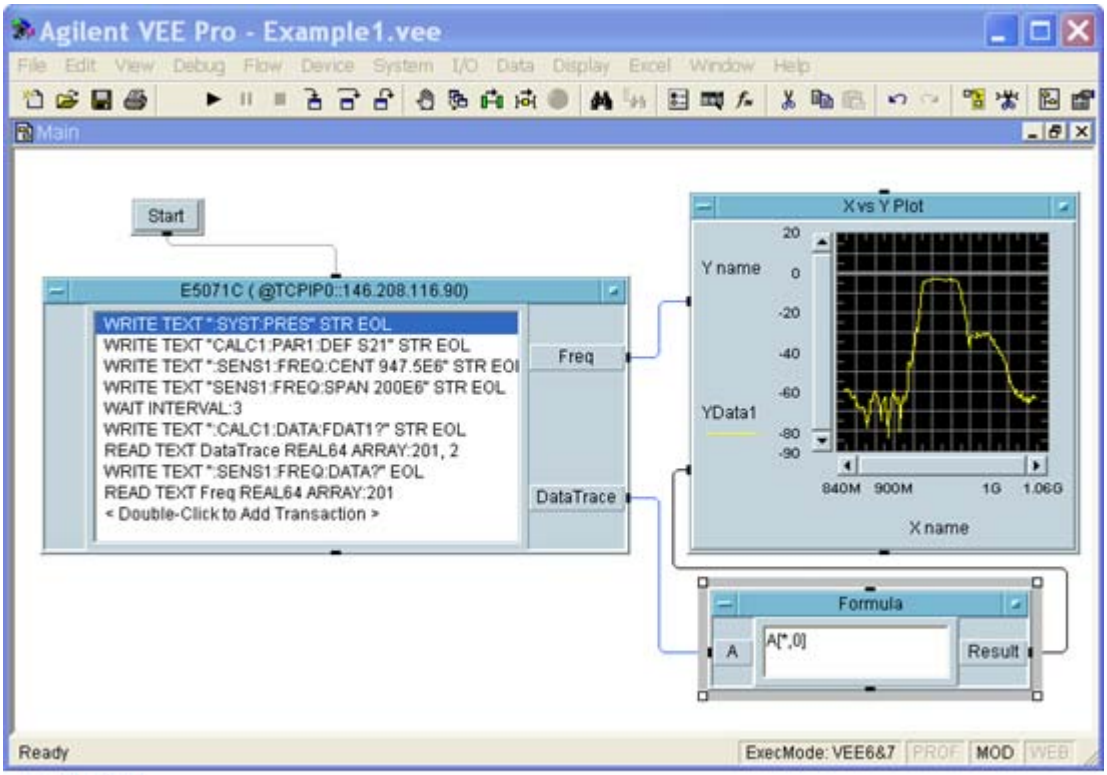
e5071c149

5. The Instrument manager displays the connection with E5071C.



e5071c150

The following figure shows an example of control using the I/O interface that has been set in the above procedure.



e5071c151

Control with Telnet Server

In the control system over telnet server, communications are performed through connection between the sockets provided by the processes of the external controller and the E5071C to establish a network path between them.

A socket is an endpoint for network connection; port 23 and port 5025 are provided for the sockets for the E5071C. Port 23 is provided for conversational control using telnet (user interface program for the TELNET protocol) and port 5025 for control from a program.

- To use telnet, port 23 and 5025 should be opened through Windows firewall.
- By opening port 23 and 5025, the E5071C can be controlled remotely using telnet. It is recommended to close port 23 and 5025 after usage from the security prospective.

Preparing the E5071C

To communicate with the external controller, follow these steps to turn on the telnet server of the E5071C in advance.

System > **Misc Setup** > **Network Setup** > **Telnet Server [ON]**

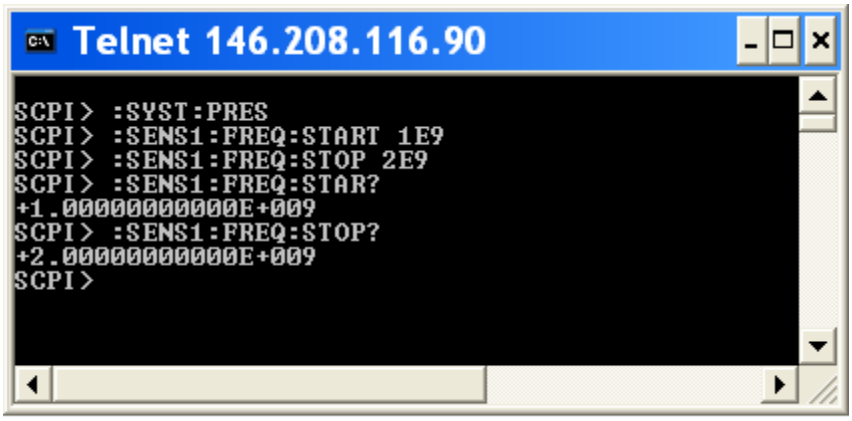
Conversational control using telnet (using port 23)

You can use telnet to perform conversational control by sending SCPI commands to the E5071C on a message-by-message basis. For telnet, the socket of port 23 is used for communications.

In this example, in order to show you the control procedure using telnet, you control the E5071C (IP address: 146.208.116.90 and host name: e5071c) from the external controller in the Windows environment.

1. Open the MS-DOS command prompt screen.
2. At the MS-DOS prompt, type telnet 146.208.116.90 or telnet e5071c and press the return key.
3. The telnet screen opens.
4. Type a command and press the return key; it is sent to the E5071C and executed. If you enter a command that queries some data, the query response is displayed below the line you have entered the command.
5. The following figure shows the screen after using the :SYST:PRES command to reset, the :SENS{1-36}:FREQ:STAR command and :SENS{1-36}:FREQ:STOP command commands to set the sweep start value and stop value to 1 GHz and 2 GHz respectively, and checking the settings.

Example of control using telnet



e5071c153

6. Press] while holding down Ctl in the telnet screen to break the connection to the E5071C. The telnet prompt appears. At the telnet prompt, type quit and press the Enter key. The connection to the E5071C breaks and telnet finishes.)

Control from a program (using port 5025)

When controlling the E5071C from a program on the external controller, use the socket of port 5025 for connection.

- Some functions such as service requests that are available in the GPIB remote control system are not available in control over telnet server.

Control using C or Visual Basic

You can control the E5071C by socket programming using the C language in the UNIX environment, or Visual C++ or Visual Basic in the Windows environment.

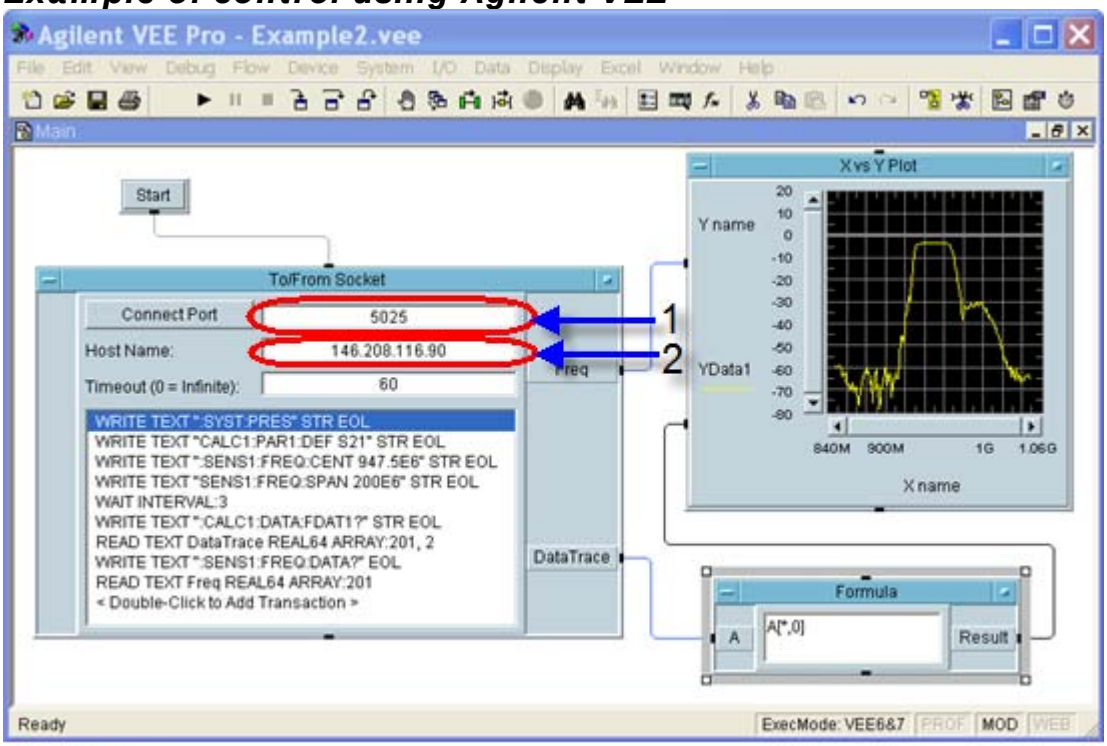
For socket programming, the library for network connection on the TCP/IP protocol is required. For the UNIX environment, BSD (Berkeley Software Distribution) Sockets API is available; for the Windows environment, WinSock (WinSock1.1 and WinSock2.0) created by porting BSD Sockets to Windows and expanding it is available.

For more information on the control method, see a sample program for control using WinSock described in "Controlling Using Telnet Server".

Control using Agilent VEE

Agilent VEE allows you to control the E5071C through the connection to the socket of port 5025 using To/From Socket. The following figure shows an example (when the IP address of the E5071C is 146.208.116.90). Enter 5025 in **Host Name** to specify the port for connection (1 in the following figure) and enter the IP address or host name of the E5071C in the field to specify the host name (2 in the following figure).

Example of control using Agilent VEE



e5071c152

USB Remote Control System

- Overview
- System Configuration

Other topics about Overview

Overview

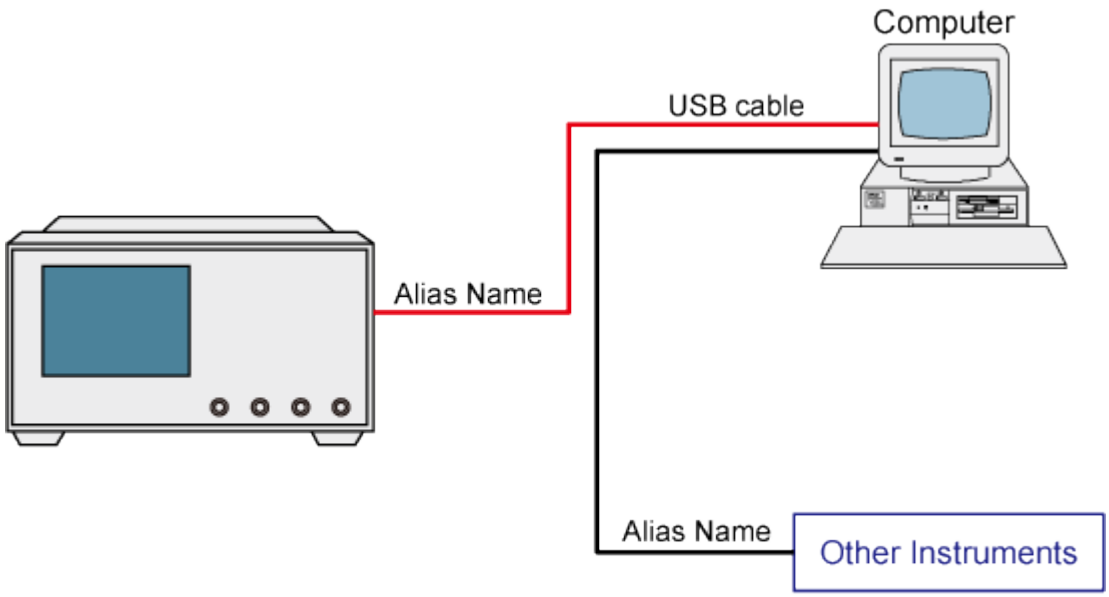
The USB (Universal Serial Bus) remote control system provides device control via USB, which is equivalent to control via GPIB. Connection is made through an interface in compliance with USBTMC-USB488 and USB 2.0.

System Configuration

The USB remote control system controls instruments that use the name "alias." There is no such address for GPIB connections.

Use a USB cable to connect the E5071C to an external controller (personal computer). The following figure shows an overview of the system configuration for the USB remote control system.

USB Remote Control System Configuration



e5071c339

Required Equipment

- E5071C
- External controller (PC with USB host port (type A)).

- Other USB compatible devices (instruments and/or peripherals for specific purposes).
- USB cable connecting E5071C and external controller (with type A/4-prong male or type B/4-prong male connectors depending on device used).

USB Port Types

There are two standard types of USB ports. The external controller (PC) must be connected via the USB host port (type A), while the E5071C and other USB compatible devices must be connected via the USB interface port (type B).

Port Type	Description
	Type A: USB host port
	Type B: USB (USBTMC) interface port

Preparing E5071C

You do not have to configure any softkey or command of the E5071C in order to control the E5071C from an external controller. Simply connect a USB cable to the USB interface port.

Preparing External Controller

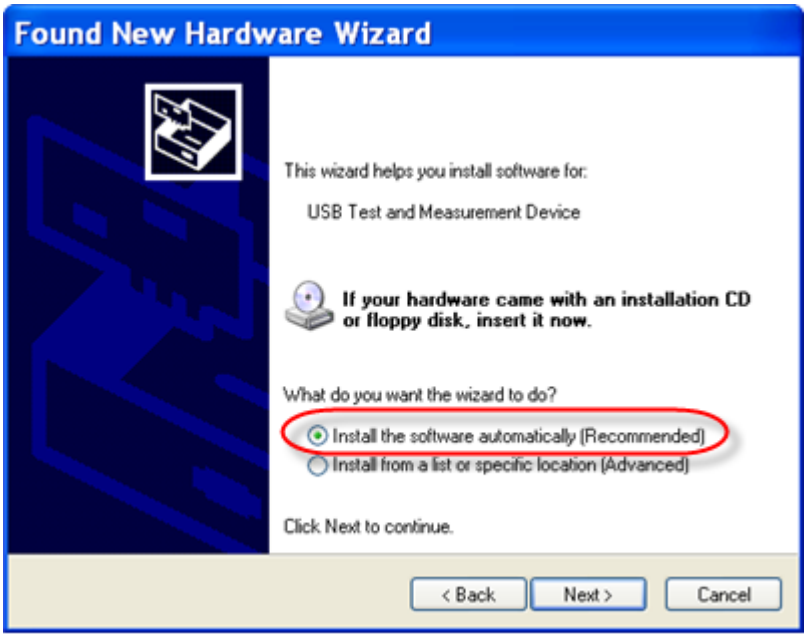
In order to establish communication with the E5071C via USB, you must set up the I/O interface of the external controller in advance. The USB can identify devices automatically, so once you connect a USB cable to a target device, a dialog box will appear for USB device registration.

- The E5071C will be identified as new device if its serial number has been changed.
- You must install the Agilent I/O Libraries on your PC in advance. Use Agilent I/O Libraries Suite 14.2 or later.

1. Setting E5071C when USB Cable Is Connected
    1. When new device is connected via USB cable, the following dialog box will appear automatically. Select **No, not this time**, and then click **Next**.

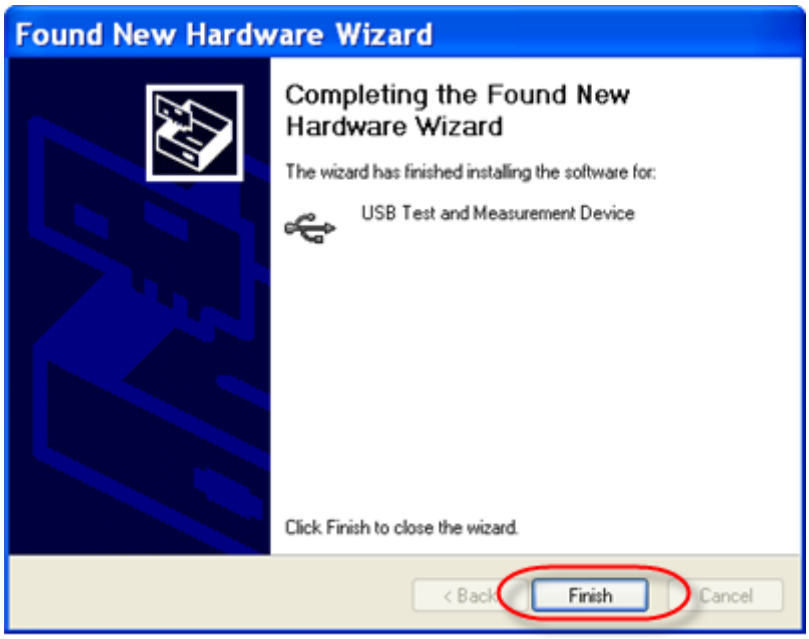
e5071c155

2. Select **Install the software automatically (Recommended)**, and then click **Next**.



e5071c156

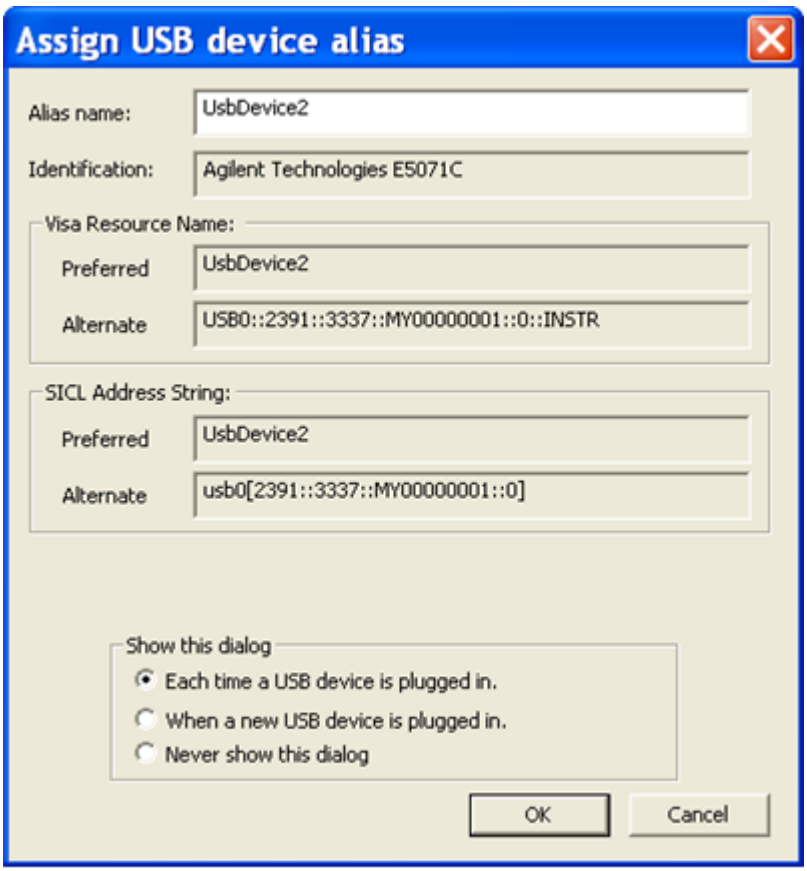
3. The drivers for E5071B are automatically installed and the completion screen appears. Click **Finish** to complete the process.



e5071c157

2. Registering Alias

Just after finishing the setting, another screen appears that can be used to change the Alias for the connected E5071C.



e5071c158

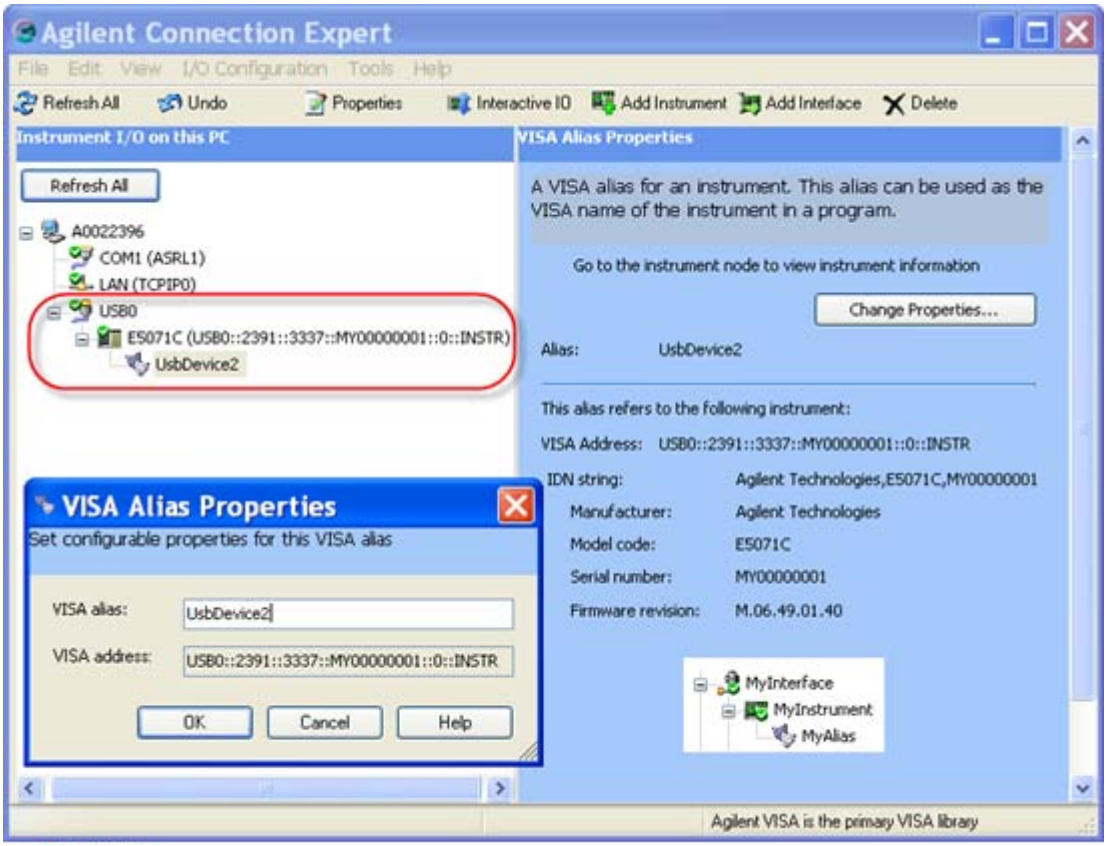
- For alias, use the ASCII format less than 127 digits. Alias is upper/lower case insensitive.
- If **Never show this dialog** is selected in **Show this dialog** frame, the dialog box does not appear even if a new device is connected.
- Once new device is identified, the "New Hardware Search Wizard" will start. Follow the instruction to implement the processing.

3. Changing Alias on Setting Screen

The following are steps using the Agilent I/O Libraries Suite 14.2.

1. From the Start menu of your PC, click **Programs** > **Agilent IO Libraries Suite** > **Agilent Connection Expert** to open the Config setting screen.
2. In the Config setting screen, select the alias names from **USB0** onward in the **Instrument I/O on this PC** frame, and then use the **Change Properties** from **I/O Configuration** on the menu bar.

Changing Alias



e5071c145

Control using C or Visual Basic

You can control the E5071C by programming using Visual C++ or Visual Basic in the Windows environment as well as SICL/VISA.For further information on controlling the E5071C, see the manual of SICL or VISA. For Agilent I/O Libraries, use Agilent I/O Libraries Suite 14.2 or later.

You may use alias in the programming using SICL/VISA.

The following example shows an OPEN command to control the E5071C to which alias is given as ENA_USBIF.

SICL	id = iopen("ENA_USBIF")
VISA	viOpen(...,"ENA_USBIF",...)

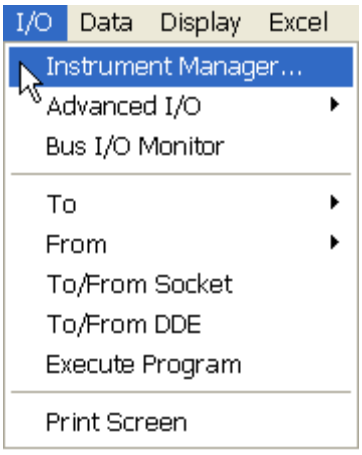
- For further details of the programming using SICL/VISA, see the SICL Users Guide or the VISA Users Guide.

Control using Agilent VEE

Agilent VEE allows you to control the E5071C via the direct I/O interface. The following example shows how to control the E5071C to which alias is given as ENA_USBIF.

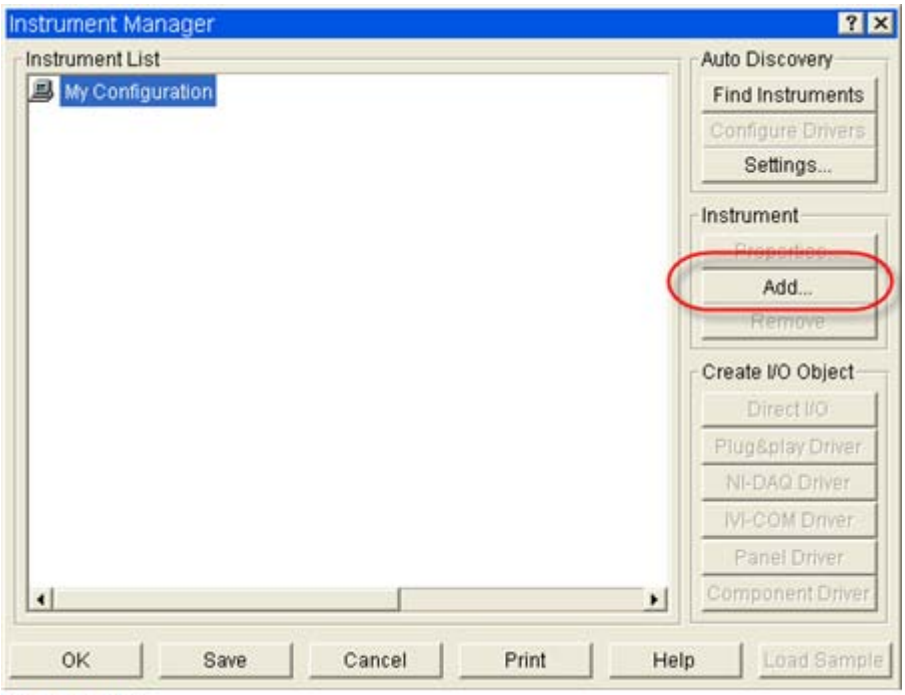
- When using Agilent VEE for PC, use Agilent VEE Pro 7 for Windows or later version.

1. On the Agilent VEE's **I/O** menu, click **Instrument Manager**.



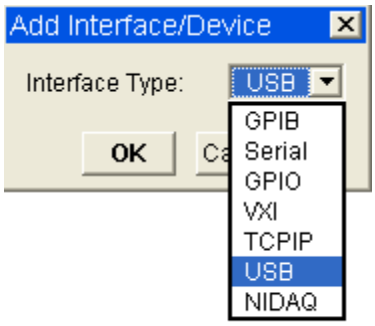
e5071c146

2. In **Instrument Manager**, click **Add...**



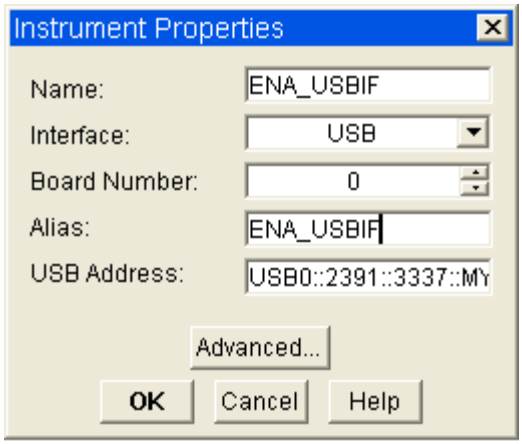
e5071c147

3. A new windows appears for the selection of Interface Type. Select **USB** and click **OK**.



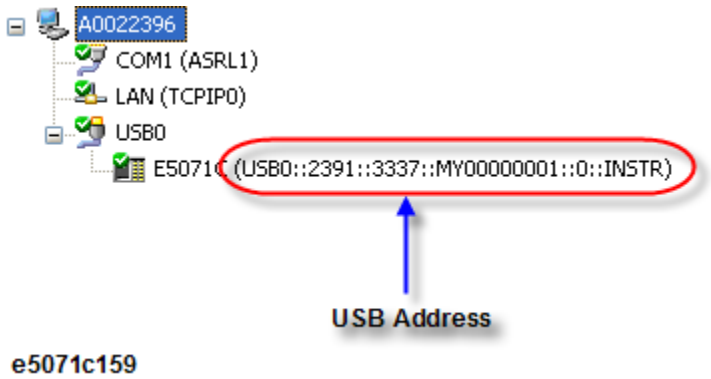
e5071c154

4. In **Instrument Properties**, type any name for the Instrument in Name (for example: ENA_USBIF or E5071C_USB), and add USB Address in the **USB Address.** Click **OK** after entering all the parameters.



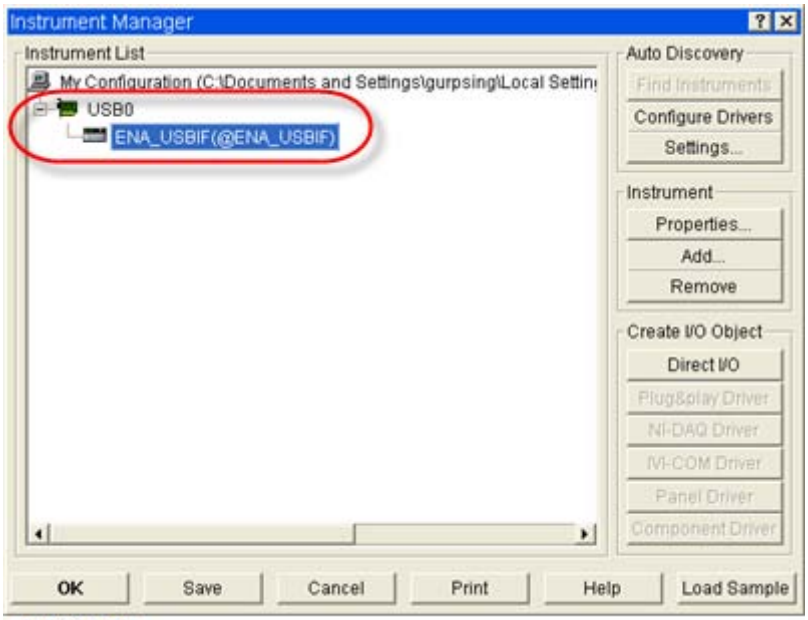
e5071c181

3. The USB address can be retrieved from Agilent Connection Expert.



e5071c159

5. The E5071C successfully appears in the **Instrument Manager**.



e5071c182

1.

Sending SCPI command messages

- Type and Structure of Commands
- Grammar of Messages
- Remote Mode

Other topics about Overview

Type and Structure of Commands

The SCPI commands available for the E5071C are classified into 2 groups as follows.

E5071C commands

Commands specific to the E5071C. They cover all measurement functions that the E5071C has and some general-purpose functions. The commands in this group are arranged in a hierarchical structure called the command tree. Each command consists of character strings (mnemonics) indicating each hierarchical level and colon (:) separators between hierarchical levels.

IEEE common commands

Commands to cover general-purpose functions defined in IEEE488.2 that are available commonly to instruments that support this standard. The commands in this group have an asterisk (*) at the beginning. For the commands in this group, there is no hierarchical structure.

Concepts of the command tree

The commands at the top of the command tree are called "root command" or simply "root." To access lower level commands in the tree, you need to specify a specific path like a directory path in the DOS file system. After power-on or reset, the current path is set to the root. Special characters in messages change the path setting as described below.

Message terminator

A message terminator such as the
<new line> character sets the current path to the root.

Colon (:)

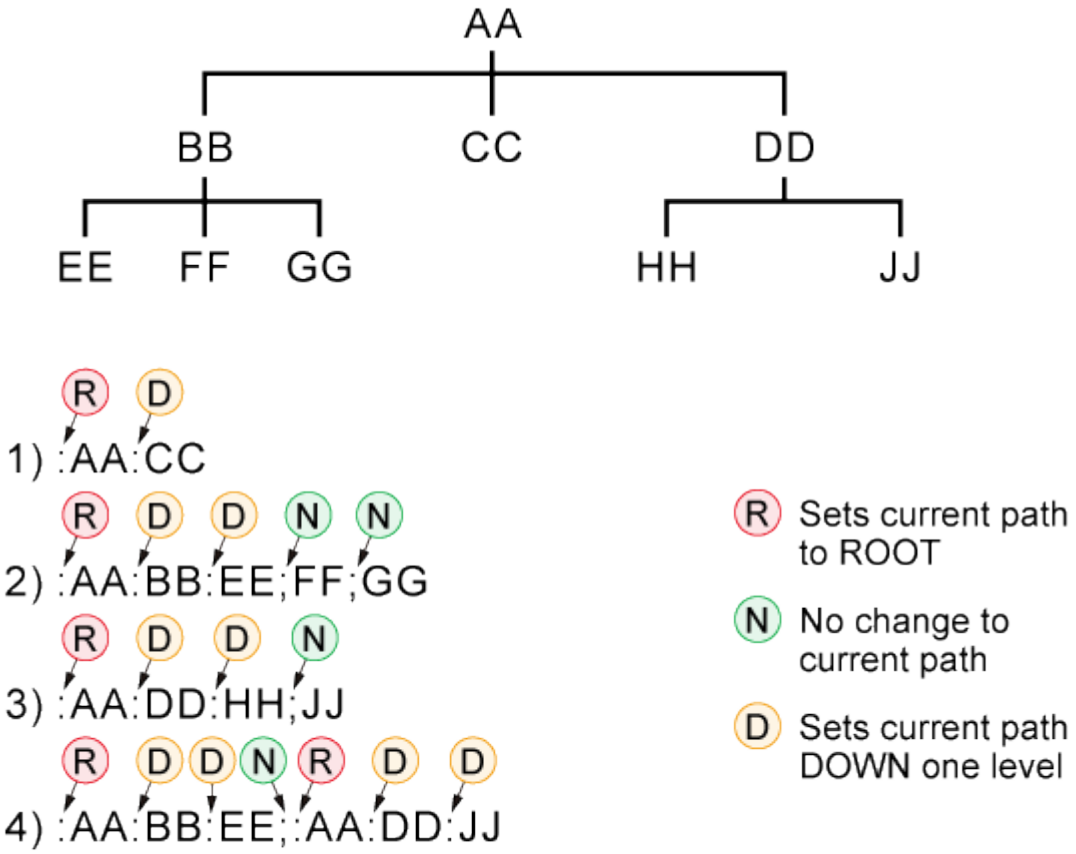
A colon between 2 command mnemonics lowers the level of the current path in the command tree. A colon used as the first character of a command specifies the command mnemonic that follows as the root-level command.

Semicolon (;)

A semicolon does not change the current path and separates 2 commands in the same message.

The following figure shows an example of how to use colons and semicolons to efficiently access commands in the command tree.

Using colons and semicolons



e5071c350

Grammar of Messages

This section describes the grammar to send program messages via GPIB. Program messages are messages that the user sends to the instrument from the external controller to control the instrument. A program message consists of 1 or more commands and their necessary parameters.

Upper/lower case sensitivity

Upper/lower case insensitive.

Program message terminator

A program message must be terminated with one of the 3 program message terminators: <new line>, <^END>, or <new line><^END>. <^END> indicates that EOI on the GPIB interface becomes active at the instant when the immediately previous data byte is sent. For example, the

OUTPUT command of HTBasic automatically sends the message terminator after the last data byte.

Parameters

A space (ASCII code: 32) is required between a command and its first parameter. When sending several parameters in a single command, separate each parameter with a comma (,).

Message including several commands

When sending 2 or more commands in a single message, separate each command with a semicolon (;). The following example shows how to send the *CLS command and the :STAT:PRES command in a single message using HTBasic.

```
OUTPUT 717;"*CLS;:STAT:PRES"
```

Remote Mode

The E5071C does not provide remote mode. Therefore, even if you send a GPIB command, it never enters into remote mode automatically. There is no local key to release remote mode.

If you need to prevent misoperation during remote control due to entry from the front panel or mouse, lock the input devices using the following commands.

- :SYST:KLOC:KBD
- :SYST:KLOC:MOUS

LXI

The E5071C is LXI-C compliant from firmware revision A.08.00 onwards.

About LXI

LXI (LAN eXtensions for Instrumentation) is the LAN-based successor to GPIB and combines the advantages of Ethernet with the simplicity and familiarity of GPIB. The key features of LXI are as follows:

- The speed, simplicity, worldwide reach, low cost, ongoing enhancement and backward compatibility of LAN.
- Quick, easy configuration through the intuitive web interface built into compliant instruments.
- Simplified programming and greater software reuse through IVI drivers.
- The ability to create hybrid systems that include LXI, GPIB, VXI, PXI, CANbus, etc.
- Enhanced system performance and event handling via hardware- and LAN-based triggering modes.
- Synchronization of local and remote instruments through the IEEE 1588 precision time protocol.

Checking LXI Compliance

The E5071C having LXI compliance will show a LXI logo at the start up screen of the firmware

. If this logo does not appears at the start up screen of the firmware, it means that the system is not LXI-C complaint.

Files Required for LXI with HDD revision CN70x

The HDD revision CN70x does not comes with installed Dot Net architecture which is a requirement for LXI. User having HDD revision CN70x need to install the required components for LXI on the E5071C, and then update the firmware to A.08.00 or later to use LXI-C functionality.

1. Update the E5071C with the latest Firmware revision if your firmware revision is A.07.0x.

2. LXI works only on Dot Net platform.

Setting up Analyzer

Setting up Analyzer

- Selecting the Active Channel/Trace
- Configuring Measurement Conditions
- Configuring Display Settings
- Saving and Loading the Settings

Selecting the Active Channel / Trace

You can configure the E5071C by using various commands. Some commands require you to specify and work with a particular channel or trace, while other commands do not have this restriction.

Those commands that do not require you to specify a particular channel or trace apply to the currently active channels and traces. Therefore, before issuing such a command, you must make the appropriate channels and traces active.

To make a channel active, use the following command:

:DISP:WIND{1-160}:ACT

- Only the currently displayed channels can be active channels. Therefore, you must display the desired channels by using the :DISP:SPL command before making them active.

To make a trace active, use the following command:

:CALC{1-160}:PAR{1-16}:SEL

- Only the currently displayed traces can be active traces. Therefore, you must display the desired traces by using the :CALC{1-160}:PAR:COUN command before making them active.

If you are using E5071C revision A.9.60 and above, you can select a trace by the trace name, provided it has been defined earlier. To define a trace name, use the following command:

:CALC{1-160}:PAR{1-16}:TNAMe:DATA

- If the trace name is not defined but used, the following error occurs:

'51, Specified trace does not exist'

- If the defined trace name already exist, the following error occurs:

'63, Duplicate trace name'

If you are using E5071C revision A.9.60 and above, you also have an option to select the trace directly by using TRAC{1-16} for all SELected commands. As such, you do not need to make a trace active before assigning a command to it.

For example, to activate marker 3 in trace 2, channel 1:

- In E5071C revision A.9.60 and above:

:CALC1:TRAC2:MARK3:ACT

- In E5071C revision A.9.5x and below:

:CALC1:PAR2:SEL

:CALC1:MARK3:ACT

Other topics about Setting up Analyzer

Configuring Measurement Conditions

- Setting the Number of Traces
- Selecting Measurement Parameters
- Setting Sweep Condition (Stimulus)
- Configuring Averaging Settings
- Setting the System Z0

Other topics about Setting up Analyzer

Setting the Number of Traces

When you set the number of traces, that setting determines the upper limit trace number; for example, if the setting is 3, traces 1 through 3 will be displayed. To set the number of traces, use the following command:

:CALC{1-36}:PAR:COUN

- Only the currently displayed traces can be active traces. Therefore, you must set the number of traces appropriately before making them active.

Selecting Measurement Parameters

To select the measurement parameter (S parameter) for each trace, use the following command:

:CALC{1-36}:PAR{1-36}:DEF

When you use the Balance-Unbalance Conversion feature, you can select the mixed mode S parameter as well. For more information, refer to Analysis Using the Fixture Simulator.

Setting Sweep Condition (Stimulus)

How you can set the sweep condition depends on the sweep type. You can choose between the following four sweep types:

- Linear sweep
- Log sweep
- Segment sweep
- Power sweep

To select one of the above sweep types, use the following command:

:SENS{1-36}:SWE:TYPE

To select the sweep mode (stepped/swept), use the following command:

:SENS{1-36}:SWE:GEN

Turning On/Off stimulus signal output

To turn on/off the stimulus signal output, use the following commands. For example, if the power output is automatically turned off due to the power trip feature, remove the cause of the over-input and turn on the stimulus signal output by executing the following command. You cannot perform measurement until you turn on the stimulus signal output.

:OUTP

Configuring linear/log sweep settings

To set the sweep range, use the following commands:

Type	Command
Start value	:SENS{1-36}:FREQ:STAR
Stop value	:SENS{1-36}:FREQ:STOP
Center value	:SENS{1-36}:FREQ:CENT
Span value	:SENS{1-36}:FREQ:SPAN

To set the number of measurement points, use the following command:

:SENS{1-36}:SWE:POIN

To set the sweep time, use the following commands:

Type	Command
Sweep time	:SENS{1-36}:SWE:TIME
Turning on/off auto setting	:SENS{1-36}:SWE:TIME:AUTO

To set the sweep delay time, use the following command:

:SENS{1-36}:SWE:DEL

To set the IF bandwidth, use the one of the following commands (both provide the same function):

:SENS{1-36}:BAND

:SENS{1-36}:BWID

Setting power level

To set the power level, use the following command:

:SOUR{1-36}:POW

To select whether to output the same power level (the set value for port 1) or a different power level for each port, use the following command:

:SOUR{1-36}:POW:PORT:COUP

:SOUR{1-36}:POW:PORT{1-4}

To set the correction of power-level attenuation so that it's proportional to the frequency (power slope feature), use the following command:

:SOUR{1-36}:POW:SLOP:STAT

:SOUR{1-36}:POW:SLOP

- If you turn on the power slope feature, the sweep mode is changed to the step mode.

Configuring segment sweep settings

When you opt to use segment sweep, you can set all items (in the segment sweep table) by using a single command:

:SENS{1-36}:SEGM:DATA

Alternatively, you can configure the segment sweep settings based on the data contained in a CSV file by issuing the following command:

:MMEM:LOAD:SEGM

Also, you can save the contents of the current segment sweep table to a file by issuing the following command:

:MMEM:STOR:SEGM

For more information on how to save and load the segment sweep table, refer to Saving and recalling the segment sweep table.

Configuring power sweep settings

To set the sweep range, use the following commands:

Type	Command
Start value	:SOUR{1-36}:POW:STAR
Stop value	:SOUR{1-36}:POW:STOP
Center value	:SOUR{1-36}:POW:CENT
Span value	:SOUR{1-36}:POW:SPAN

To set the fixed frequency (CW frequency), use the following command:

:SENS{1-36}:FREQ

To set the number of points, the sweep time, the sweep delay time, and the IF bandwidth, use the same commands as for the linear/log sweep.

Configuring Averaging Settings

To configure the averaging settings, use the following commands:

Type	Command
On/off	:SENS{1-36}:AVER
Averaging factor	:SENS{1-36}:AVER:COUN
Clear (Restart)	:SENS{1-36}:AVER:CLE

For averaging, normally, the instrument must be triggered according to the number of averaging; however, when the averaging trigger is turned on, sweeps for the number of averaging can be executed by a single trigger. For details on the averaging trigger, refer to Averaging Trigger Function.

Setting the System Z0

- This function is available with the Firmware revision 3.01 or greater.

To set the system characteristic impedance (Z0), use the following command:

:SENS:CORR:IMP

Configuring Display Settings

- Setting the Layout of Windows and Graphs
- Configuring Trace Display Settings
- Setting Display Color

Other topics about Setting up Analyzer

Setting the Layout of Windows and Graphs

You can split the E5071C's LCD screen into multiple windows that display channel-specific result information, and the window layout can be selected from a number of variations. In addition, you can place on screen a segment sweep table or echo window, which you can use to display messages from your custom program.

Selecting the window layout (Channel Display Mode)

One window displays the results for a single channel. You cannot have a single window display the results from more than one channel. This means that setting the window layout determines the number of channels displayed on screen.

To select one of the 19 different window layouts shown in the figure below, use the following command:

:DISP:SPL

Selecting the graph layout (Trace Display Mode)

You can place a number of trace graphs in each window by selecting one of the pre-defined graph layouts. The number of graphs differs depending on your selected graph layout. If the number of graphs is equal to or larger than the number of traces (set by the :CALC{1-36}:PAR:COUN command), each graph always displays one trace. On the other hand, if the number of graphs is smaller than the number of traces, some of the graphs display two or more traces. Graph 1 is populated with trace 1, graph 2 with trace 2, and so on. Traces whose numbers exceed the last graph's number will populate graph 1, graph 2, and so on.

To select one of the 19 different graph layouts shown in the figure below, use the following command:

:DISP:WIND{1-36}:SPL

Maximizing a window or a trace graph

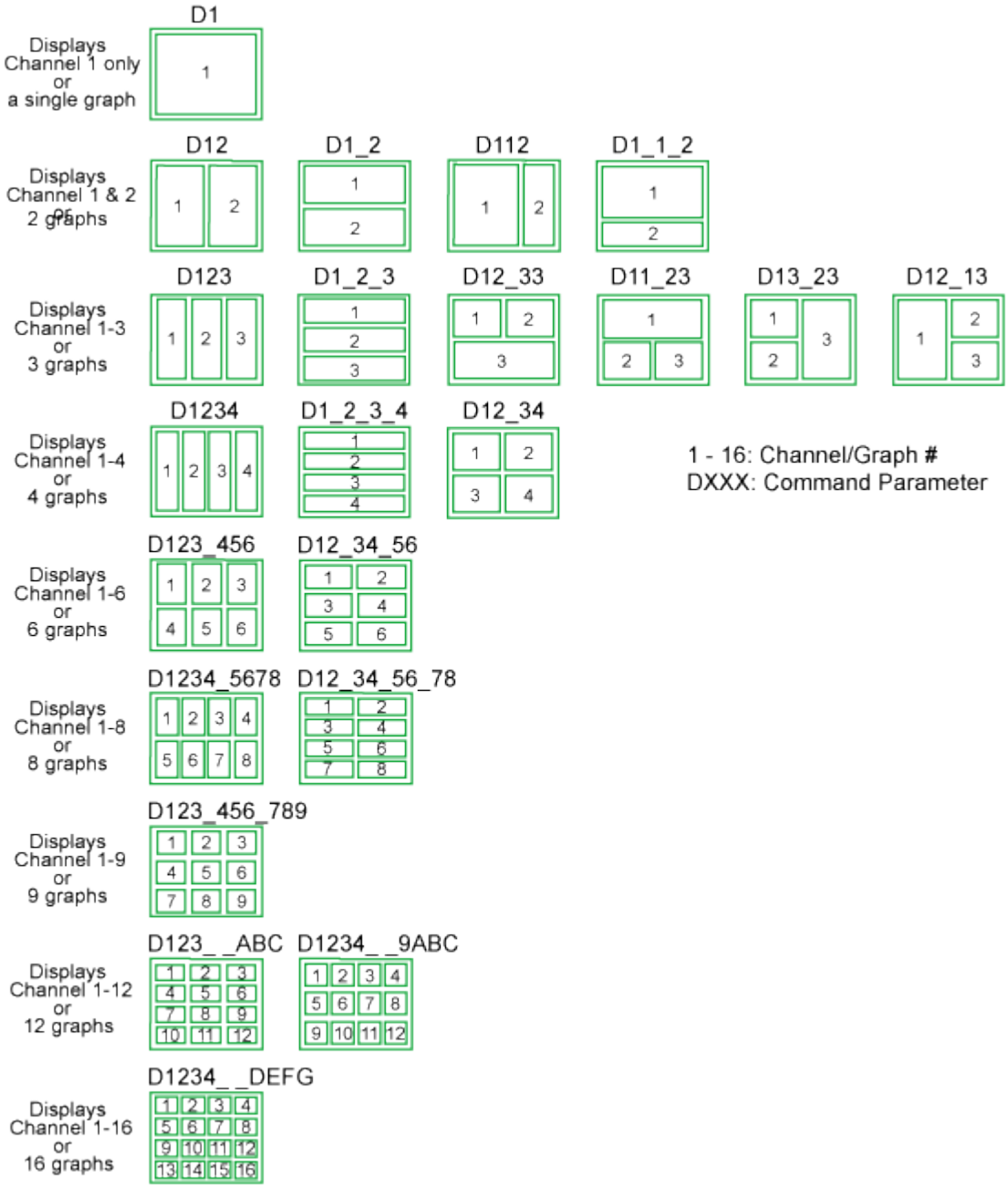
When you have multiple windows displayed, you can maximize the active channel window so that it covers the entire screen area. To maximize a window, use the following command:

:DISP:MAX

Similarly, when you have multiple traces displayed, you can maximize the active trace so that it extends throughout the entire window. To maximize a trace, use the following command:

:DISP:WIND{1-36}:MAX

Window/graph layouts and command parameters



e5071c314

Showing/hiding a table or echo window

You can display the following items at the bottom of the LCD screen:

- Segment sweep table
- Limit table
- Marker list table
- Echo window (a window that displays messages from a custom program)
- Loss compensation table
- Power sensor's calibration factor table

To show or hide each of the above items, use the following command:

:DISP:TABL

You cannot have two or more of the above items displayed at a time. The screen displays only the selected item by using the following command:

:DISP:TABL:TYPE

Showing/hiding softkey labels

You can show or hide the softkey labels placed alongside the right-hand edge of the LCD screen. To show or hide the softkey labels, use the following command:

:DISP:SKEY

Configuring Trace Display Settings

Selecting which traces to display

Each trace has two different representations: data and memory traces. You can show or hide the data and memory traces independently of each other. To show or hide the data or memory traces, use the following commands:

Type	Command
Data trace	:DISP:WIND{1-36}:TRAC{1-36}:STAT
Memory trace	:DISP:WIND{1-36}:TRAC{1-36}:ANN:MARK:POS:X

To copy the data trace to the memory trace, use the following command:

:CALC{1-36}:MATH:MEM

Configuring cross-trace math operations

You can perform math operations between the data and memory traces and have the results displayed as the data trace. To perform cross-trace math operations, use the following command:

:CALC{1-36}:MATH:FUNC

Configuring smoothing settings

To turn on/off smoothing, use the following command:

:CALC{1-36}:SMO

The smoothing aperture is expressed as a percentage with respect to the sweep range. To set the smoothing aperture, use the following command:

:CALC{1-36}:SMO:APER

Selecting the data format

You can select the following data formats:

- Rectangular display formats
  - Log magnitude format
  - Phase format
  - Group delay format
  - Linear magnitude format
  - SWR format
  - Real format
  - Imaginary format
  - Expanded phase format
  - Positive phase format
- Smith chart format
- Polar format

To select the measurement parameter data format, use the following command:

:CALC{1-36}:FORM

Configuring the display scale

Depending on the measurement parameter data format, you can configure the display scale in one of the following two ways:

Rectangular display formats:

When you use one of rectangular display formats (Logarithmic magnitude/Phase/ Group delay/Linear magnitude/SWR/Real/Imaginary/Expanded phase/Positive phase), you can configure the display scale by setting the following four items:

Type	Command
Number of divisions	:DISP:WIND{1-36}:Y:DIV
Scale per division	:DISP:WIND{1-36}:TRAC{1-36}:Y:PDIV
Reference graticule line	:DISP:WIND{1-36}:TRAC{1-36}:Y:RPOS
Reference graticule line value	:DISP:WIND{1-36}:TRAC{1-36}:Y:RLEV

- The number of divisions is a channel-wide setting (shared among all traces), while the remaining three settings are trace-specific.

You can show or hide graticule label (the label on the left-hand side of the graticule lines) by issuing the following command:

:DISP:WIND{1-36}:LAB

Smith chart/Polar formats:

When you are using one of Smith chart/Polar formats, you can only set the full scale value (the outermost circle's value) using the following command:

:DISP:WIND{1-36}:TRAC{1-36}:Y:PDIV

Auto Scale

You can use Auto Scale to automatically set the display scale. This feature works by automatically adjusting the reference division line value and the scale value per division when you are using one of the rectangular display formats or the full scale value when you are using one of Smith chart/Polar formats.

To perform Auto Scale, use the following command:

:DISP:WIND{1-36}:TRAC{1-36}:Y:AUTO

Displaying a message in the echo window

You can display a message in the echo window by issuing the following command:

:DISP:ECHO

You can clear any message displayed in the echo window by issuing the following command:

:DISP:ECHO:CLE

Turning On/Off display update

To turn on/off the update of the LCD screen, use the following command:

:DISP:ENAB

Showing/hiding frequencies

To show or hide frequencies on the LCD screen, use the following command:

:DISP:ANN:FREQ

Showing or hiding the title

To show or hide the title, use the following command:

:DISP:WIND{1-36}:TITL

To define the title string that appears in the title display area, use the following command:

:DISP:WIND{1-36}:TITL:DATA

Configuring date/time display

To show or hide the current date and time on the right-hand side of the instrument status bar, use the following command:

:DISP:CLOC

To set the date and time, use the following command:

:SYST:DATE

:SYST:UPR

Turning On/Off the LCD backlight

To turn on/off the LCD backlight, use the following command (note that turning off the backlight makes the screen unreadable):

:SYST:BACK

Setting Display Color

Selecting Display Mode

You can select the one of two LCD display modes: normal display (black background) or inverted display (white background).

To select the display mode, use the following command:

:DISP:IMAG

Setting display color for each item

To set the display colors, use the following commands:

Data trace	:DISP:COL{1-2}:TRAC{1-36}:DATA
Memory trace	:DISP:COL{1-2}:TRAC{1-36}:MEM
Graph	:DISP:COL{1-2}:GRAT{1-2}
Limit test	:DISP:COL{1-2}:LIM{1-2}
Background	:DISP:COL{1-2}:BACK

Resetting display colors to factory state

You can reset the display colors in normal display and inverted display to the preset factory state.

To reset the display colors, use the following command:

:DISP:COL{1-2}:RES

Saving and Loading the Settings

You can save the settings for measurement conditions and screen display to a file along with other instrument settings, and these settings can later be loaded from the file.

Once you have saved the measurement condition and screen display settings to a file, you can later load them whenever necessary; therefore, you can quickly modify the settings loaded from a file to create new settings without having to issue many commands.

To save the current settings to a file, use the following command:

:MMEM:STOR

To load the settings from a file, use the following command:

:MMEM:LOAD

Other topics about Setting up Analyzer

Performing Calibration

Performing Calibration

- Calibration
- Power Calibration
- Receiver Calibration
- Scalar-Mixer Calibration
- Partial overwrite

Calibration

- Overview
- Performing Calibration
- Defining Calibration Kits
- Standard Definitions
- Reading/Writing Calibration Coefficient Alone
- Clearing Calibration Data and Calibration Coefficients

Other topics about Performing Calibration

Overview

You need to execute calibration to eliminate error elements related to measurement, thus allowing you to perform accurate measurement.

Performing Calibration (Obtaining calibration coefficients)

Selecting a Calibration Kit

To select a calibration kit, use the following command:

:SENS{1-160}:CORR:COLL:CKIT

Selecting a Calibration Type

The calibration coefficients are calculated based on the selected calibration type. Therefore, before you can calculate the calibration coefficients, you must select the appropriate calibration type by using one of the following commands.

Calibration type		Command
Response	OPEN	:SENS{1-160}:CORR:COLL:METH:OPEN
	SHORT	:SENS{1-160}:CORR:COLL:METH:SHOR
	THRU	:SENS{1-160}:CORR:COLL:METH:THRU
Enhanced Response		:SENS{1-160}:CORR:COLL:METH:ERES
1-Port		:SENS{1-160}:CORR:COLL:METH:SOLT1
Full 2-Port		:SENS{1-160}:CORR:COLL:METH:SOLT2
Full 3-Port		:SENS{1-160}:CORR:COLL:METH:SOLT3
Full 4-Port		:SENS{1-160}:CORR:COLL:METH:SOLT4

2-Port TRL	:SENS{1-160}:CORR:COLL:METH:TRL2
3-Port TRL	:SENS{1-160}:CORR:COLL:METH:TRL3
4-Port TRL	:SENS{1-160}:CORR:COLL:METH:TRL4

- To calculate the calibration coefficients for the simplified full 3-port and simplified full 4-port calibrations, select the full 3-port and full-4 port commands, respectively. To calculate the calibration coefficient for the simplified 3-port TRL calibration and the simplified 4-port TRL calibration, select the 3-port TRL and the 4-port TRL commands, respectively.

To check the currently selected calibration type, use the following command:

:SENS{1-160}:CORR:COLL:METH:TYPE?

Setting the trigger source for calibration

To set the trigger source for calibration, use the following command. By setting the trigger source for calibration to "System," setting the trigger source for measurement to "External," and turning on the point trigger function, it becomes possible to use "calibration for each measurement point using the external trigger."

:SENS{1-160}:CORR:TRIG:FREE

- The trigger source for calibration does not function for the calibrations of E-Cal, power, receiver, and scalar mixer.

Measuring Calibration Data

To measure the calibration data, use one of the following commands:

Calibration data items	Command
OPEN	:SENS{1-160}:CORR:COLL:OPEN
SHORT	:SENS{1-160}:CORR:COLL:SHOR
LOAD	:SENS{1-160}:CORR:COLL:LOAD
THRU	:SENS{1-160}:CORR:COLL:THRU
Isolation	:SENS{1-160}:CORR:COLL:ISOL
TRL Thru	:SENS{1-160}:CORR:COLL:TRLT

TRL Reflection	:SENS{1-160}:CORR:COLL:TRLR
TRL Line/Match	:SENS{1-160}:CORR:COLL:TRLL

- You cannot run more than one of the commands listed above at a time; if you issue another command before the currently running command completes successfully, the current command will be aborted. Therefore, when you write a program that issues multiple calibration commands in series, you should use the *OPC? command or some other means to ensure that no command is executed before the preceding command completes itself.

As shown in the table below, the data required to calculate the calibration coefficients differ depending on the selected calibration type.

Calibration type (Selected ports are enclosed in parentheses)		**Data**				
		OPEN	**SHORT**	**LOAD**	**THRU**	**Isolation**
Response	**OPEN (a)**	a	Not required	[a]	Not required	Not required
	SHORT (a)	Not required	a	[a]	Not required	Not required
	THRU (a-b)	Not required	Not required	Not required	a-b	[a-b]
Enhanced Response (a-b)		b	b	b	a-b	[a-b]
1-Port (a)		a	a	a	Not required	Not required
Full 2-Port (a-b)		a, b	a, b	a, b	a-b, b-a	[a-b], [b-a]
Full 3-Port (a-b-c)		a, b, c	a, b, c	a, b, c	a-b, b-a a-c, c-a b-c, c-	[a-b], [b-a] [a-c], [c-a] [b-c],

				b	[c-b]
Simplified Full 3-Port					
(1-2-3)	1, 2, 3	1, 2, 3	1, 2, 3	1-2, 2-1 1-3, 3-1 [2-3], [3-2]	[1-2], [2-1] [1-3], [3-1] [2-3], [3-2]
(1-2-4)	1, 2, 4	1, 2, 4	1, 2, 4	1-2, 2-1 1-4, 4-1 [2-4], [4-2]	[1-2], [2-1] [1-4], [4-1] [2-4], [4-2]
(1-3-4)	1, 3, 4	1, 3, 4	1, 3, 4	1-3, 3-1 [1-4], [4-1] 3-4, 4-3	[1-3], [3-1] [1-4], [4-1] [3-4], [4-3]
(2-3-4)	2, 3, 4	2, 3, 4	2, 3, 4	2-3, 3-2 [2-4], [4-2] 3-4, 4-3	[2-3], [3-2] [2-4], [4-2] [3-4], [4-3]
Full 4-Port	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1-2, 2-1 1-3, 3-1 1-4, 4-1 2-3, 3-2 2-4, 4-2 3-4, 4-3	[1-2], [2-1] [1-3], [3-1] [1-4], [4-1] [2-3], [3-2] [2-4], [4-2] [3-4], [4-3]
Simplified Full 4-Port	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1-2, 2-1	[1-2], [2-1]

				1-3, 3-1 [1-4], [4-1] [2-3], [3-2] [2-4], [4-2] 3-4, 4-3	[1-3], [3-1] [1-4], [4-1] [2-3], [3-2] [2-4], [4-2] [3-4], [4-3]

In the data section in the table, the letter m (for example, 1, a) represents the measurement data at port m; m-n (for example, 1-2, a-b) represents the measurement data between response port m and stimulus port n. You can omit data enclosed in brackets.

Calculating Calibration Coefficients

To calculate the calibration coefficients, use one of the following commands:

Calibration type	Command
Response, 1/2/3/4 port	:SENS{1-160}:CORR:COLL:SAVE
Simplified full 3/4 port Simplified 3/4-port TRL	:SENS{1-160}:CORR:COLL:SIMP:SAVE

Before issuing the above commands, you must measure all required calibration data items according to your selected calibration type. Calculating the calibration coefficients clears all calibration data regardless of whether they are used for the calculation. The calibration type selection is also cleared, which results in a state where no calibration type is selected.

Simplified full 3-/4-port calibration

- This function is available with Firmware revision 3.50 or greater. Note that you can execute this function from the front panel only for Firmware revision A.06.50 or greater.

The simplified full 3-/4-port calibration acquires the calibration coefficients while omitting a part of the thru measurement.

Notes on the simplified full 3-/4-port calibration

Compared to the normal full 3-/4-port calibration, the simplified full 3-/4-port calibration is more sensitive to the error that may arise when acquiring calibration data. This because the calibration coefficients are

calculated without a part of the thru measurement data. Therefore, the following should be considered when measuring data for the simplified full 3-/4-port calibration.

- The standard used for measurement must match its definition value.
  - Use a standard that provides good repeatability (stability).
  - Do not omit the length of the thru when defining the standard.
  - When using a user-created standard, verify the definition value.
  - For the N connector, remember that it has two different types: male and female.
- Assure high reliability and repeatability for measurement.
  - Reduce the difference in external environment (such as temperature difference) between the time when measuring calibration data and when measuring actual data.
  - Set the power level of the stimulus signal sufficiently small so that compression does not occur.
  - Narrow the IF bandwidth.
  - Increase the averaging factor.
  - Use a cable that has robust amplitude/phase characteristics against bending.
  - Use high-precision connectors.

Simplified 3/4-port TRL calibration

- This function is available with Firmware revision A.06.50 or greater.
- The simplified 3/4-port TRL calibration calculates the calibration coefficient by skipping part of the thru/line measurement (or line/match measurement) that is necessary for normal 3/4-port TRL calibration.
- You need to give consideration to the same conditions for the simplified 3/4-port TRL calibration as those for the simplified full 3/4-port calibration listed above.

Turning ON/OFF Error Correction

To turn ON/OFF error correction, use the following command:

:SENS{1-160}:CORR:STAT

Also, once you have calculated the calibration coefficient using the :SENS{1-160}:CORR:COLL:SAVE or :SENS{1-160}:CORR:COLL:SIMP:SAVE command, error correction is automatically turned on.

Using ECal

An ECal (Electronic Calibration) module allows you to perform 1-/2-/3-/4-port calibration and response (THRU) calibration without having to replace the standard device.

ECal works by using the calibration kit data contained in the ECal module instead of the calibration kit data selected for the E5071C. This means that you do not have to define or select a calibration kit when using ECal.

- When two or more ECal modules are connected through the USB port, the system uses the calibration kit data of the first ECal module.

To perform ECal, use one of the following commands:

Calibration type	Command
1-Port Calibration	:SENS{1-160}:CORR:COLL:ECAL :SOLT1
Full 2-Port Calibration	:SENS{1-160}:CORR:COLL:ECAL :SOLT2
Full 3-Port Calibration	:SENS{1-160}:CORR:COLL:ECAL :SOLT3
Full 4-Port Calibration	:SENS{1-160}:CORR:COLL:ECAL :SOLT4
Enhanced Response Calibration	:SENS{1-160}:CORR:COLL:ECAL :ERES
Response Calibration (THRU)	:SENS{1-160}:CORR:COLL:ECAL :THRU

Simply issuing one of the above commands completes all of the tasks necessary for error correction, including measuring the calibration data, calculating the calibration coefficients, and running the error correction feature.

- Once you have initiated ECal, you cannot cancel the operation.
- No command entered following the initiation of ECal will be processed until ECal completes successfully. Accordingly, if you issue a command that queries some data, the system will not respond to the query until ECal is complete.

The below command was intended to turn ON/OFF the isolation measurement for performing ECal. However, as the isolation performance of ENA is better than ECal, this command no longer works. ENA ignores this command.

:SENS{1-160}:CORR:COLL:ECAL:ISOL

- This command takes no action and only exists to maintain backward compatibility.

To select the ECal characteristic for a user-characterized ECal, use the following command:

:SENS{1-160}:CORR:COLL:ECAL:UCH

ECal Auto-detect Function

The ECal module can automatically detect which port of the ECal module is connected to the E5071C test port. Turn off the auto-detect function to specify a port manually.

To turn OFF the auto-detect function, use the following command.

:SENS:CORR:COLL:ECAL:ORI

To turn OFF the auto-detect function and set a port manually, use the following command.

:SENS:CORR:COLL:ECAL:PATH

Checking the Applied Calibration Type

When you turn on error correction, you can check the calibration type actually applied to each trace. To check the calibration type, use the following command:

:SENS{1-160}:CORR:TYPE{1-160}?

The above command reads out the same parameter (SOLT3) for both the full 3-port and simplified full 3-port calibrations, and thus they cannot be discriminated. For the same reason, the following calibrations cannot be discriminated: full 4-port and simplified full 4-port, 3-port TRL and simplified 3-port TRL, and 4-port TRL and simplified 4-port TRL.

Defining Calibration kits

Selecting a Calibration Kit

To select a calibration kit, use the following command:

:SENS{1-160}:CORR:COLL:CKIT

Setting the Calibration Kit Name

To set the name of a calibration kit, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:LAB

Standard Definitions

Selecting a Standard Type

To select a standard type, use the following command:
:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:TYPE

Setting the Standard Name

To set the standard name, use the following command:
:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:LAB

Setting the Standard Value

To set the standard value, use one of the following commands:

Item	Command
C0	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C0
C1	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C1
C2	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C2
C3	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :C3
L0	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L0
L1	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L1
L2	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L2
L3	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :L3
Offset Delay	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :DEL
Offset Loss	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :LOSS
Offset Z0	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :Z0
Arbitrary Impedance	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}: ARB
Start Frequency	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :FMIN
Stop Frequency	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}: FMAX
Media Type	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :CHAR
Length Type	:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30} :LTYP

As shown in the table below, you need to set different items depending on the standard type.

Standard Types	C0 to C3	L0 to L3	Offset Delay	Offset Loss	Offset Z0	Arbitrary Impedance	Min. Frequency	Max. Frequency	Connector Type
OPEN	*		*	*	*		*	*	*
SHORT		*	*	*	*		*	*	*
LOAD			*	*	*		*	*	*
THRU			*	*	*		*	*	*
Arbitrary Impedance			*	*	*	*	*	*	*

You need to set the items identified by * marks in the table above.

Saving/Recalling the Definition File

To save the definition of the selected calibration kit to a file, use the following command.

:MMEM:STOR:CKIT{1-20}

To recall the definition of the calibration kit from a file and set, use the following command.

:MMEM:LOAD:CKIT{1-20}

Defining a Subclass of the Standard

To set the standard type that varies with the frequency range, use the following command to specify the subclass.

:SENS{1-160}:CORR:COLL:SUBC

To select the subclass, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD

To set the start frequency of a specified subclass, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:FMIN

To set the stop frequency of a specified subclass, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:FMAX

Defining a Standard Class Assignment

To select the standard to be applied to the OPEN measurement for each port, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:OPEN

To select the standard to be applied to the SHORT measurement for each port, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:SHOR

To select the standard to be applied to the LOAD measurement for each port, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:LOAD

To select the standard to be applied to the THRU measurement between each pair of ports, use the following command:

:SENS{1-160}:CORR:COLL:CKIT:ORD:THRU

To select the standard to be applied to the THRU measurement for the TRL calibration between each pair of ports, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD:TRLT

To select the standard to be applied to the Reflection measurement for the TRL calibration between each pair of ports, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD:TRLR

To select the standard to be applied to the Line/Match measurement for the TRL calibration between each pair of ports, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:ORD:TRLL

Setting the Standard Media Type

To select the media type of the standard to be used, use the following command.

:SENS{1-160}:CORR:COLL:CKIT:STAN{1-30}:CHAR

Saving and loading calibration coefficients

You can save calibration coefficients to a file along with other instrument settings and then later load them from the file.

By default, the system does not save calibration coefficients when it saves instrument settings. Therefore, to save calibration coefficients, you must explicitly configure the system to save them by issuing the following command:

:MMEM:STOR:STYP

To save calibration coefficients to a file, use the following command:

:MMEM:STOR

To load calibration coefficients from a file, use the following command:

:MMEM:LOAD

For more information on how to save and load calibration coefficients, refer to Saving and recalling instrument status

Reading/Writing Calibration Coefficient Alone

The calibration coefficient alone can be read from and written to the E5071C by using the following command:

:SENS{1-160}:CORR:COEF

To write a positive calibration coefficient, use one of the following commands to declare the calibration type:

:SENS{1-160}:CORR:COEF:METH:ERES

:SENS{1-160}:CORR:COEF:METH:OPEN

:SENS{1-160}:CORR:COEF:METH:SHOR

:SENS{1-160}:CORR:COEF:METH:SOLT1

:SENS{1-160}:CORR:COEF:METH:SOLT2

:SENS{1-160}:CORR:COEF:METH:SOLT3

:SENS{1-160}:CORR:COEF:METH:SOLT4

:SENS{1-160}:CORR:COEF:METH:THRU

To validate the written calibration coefficient, use the following command:

:SENS{1-160}:CORR:COEF:SAVE

About Calibration Types and Coefficients

The following table shows the required calibration coefficients for each calibration type.

Calibration Type	Calibration Coefficient

	ES	ER	ED	EL	ET	EX
Response calibration (OPEN)		*	*			
Response calibration (SHORT)		*	*			
Response calibration (THRU)					*	*
Enhanced response calibration	*	*	*		*	*
1-port calibration	*	*	*			
Full 2-port calibration	*	*	*	*	*	*
Full 3-port calibration	*	*	*	*	*	*
Full 4-port calibration	*	*	*	*	*	*
2-Port TRL calibration	*	*	*	*	*	
3-Port TRL calibration	*	*	*	*	*	
4-Port TRL calibration	*	*	*	*	*	

- If either an invalid calibration coefficient is specified for the writing command or a nonexistent calibration coefficient is specified for its reading command, the following error will occur: 23, Specified error term does not exist

Procedures for Writing Calibration Coefficient

You must follow the steps below to write the calibration coefficient.

1. Declare the calibration type to write.

Execute :SENS{1-160}:CORR:COEF:METH:xxxx command

2. Write any calibration coefficient.

Execute :SENS{1-160}:CORR:COEF command as needed for the written calibration coefficients

3. Validate the calibration coefficients.

Execute :SENS{1-160}:CORR:COEF:SAVE command

- Do not execute any other command while writing the calibration coefficients.This may cause the system to function incorrectly.
- If the fixture simulator is turned on, the calibration coefficient writing will not function correctly. Make sure to turn off the fixture simulator before execution.

Clearing Calibration Data and Calibration Coefficients

Clearing Calibration Data

When the frequency offset function has been disabled

You can use the following command to clear the measurement values of calibration data executed with :SENS{1-160}:CORR:COLL:OPEN command, etc.

:SENS{1-160}:CORR:COLL:CLE

When the frequency offset function has been enabled

You can use the following command to clear the measurement values of calibration data executed with :SENS{1-160}:CORR:OFFS:COLL:OPEN command, etc.

:SENS{1-160}:CORR:OFFS:COLL:CLE

These clear functions make the temporarily settings during the calibration, such as trace number and measurement parameters, recover to the original state.

Clearing Calibration Coefficients

When the frequency offset function has been disabled

You can use the following command to clear the calibration coefficients used.

:SENS{1-160}:CORR:CLE

When the frequency offset function has been enabled

You can use the following command to clear the calibration coefficients used.

:SENS{1-160}:CORR:OFFS:CLE

This command does not clear calibration coefficients related to normal calibration.

Power Calibration

- Overview
- Preparation for Power Calibration
- Loss Compensation
- Turning ON/OFF Power-level Error Correction

Other topics about Performing Calibration

Overview

The E5071C lets you perform power calibration by using a power meter to output a more accurate power level.

Preparation for Power Calibration

Connecting Power Meter and Power Sensor

The E5071C performs power calibration by controlling the power meter via GPIB. Therefore, you need to connect the E5071C and the power meter by using the USB/GPIB interface.

Selecting the Power Meter

Use the following command to select the type of GPIB or USB enabled power meter:

:SYST:COMM:PSENsor

Setting GPIB Address of Power Meter

Use the following command to set the GPIB address of the power meter to the E5071C.

:SYST:COMM:GPIB:PMET:ADDR

Setting Power Sensor Calibration Factor Table

Before performing power calibration, set the reference calibration factor (the calibration factor at 50 MHz) and the calibration factor for each frequency depending on the power sensor you use.

- Depending on the combination of power meter and power sensor that you use, the power sensor calibration factor table may be set automatically by the power meter. In this case, do not set the calibration factor table with the E5071C.

To set the reference calibration factor of the power sensor, use the following commands:

Channel A	:SOUR:POW:PORT:CORR:COLL:ASEN:RCF
Channel B	:SOUR:POW:PORT:CORR:COLL:BSEN:RCF

To set the calibration factor table of the power sensor for each frequency, use the following commands:

Channel A	:SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA
Channel B	:SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA

Alternatively, you can configure the power sensor calibration factor table based on the data contained in a CSV file by issuing the following commands:

Channel A	:MMEM:LOAD:ASCF
Channel B	:MMEM:LOAD:BSCF

Also, you can save the contents of the current power sensor calibration factor table to a file by issuing the following commands:

Channel A	:MMEM:STOR:ASCF
Channel B	:MMEM:STOR:BSCF

For more information on saving/recalling the power sensor calibration factor table, refer to Saving/recalling Power Sensor Calibration Factor Table.

Loss Compensation

You can perform loss compensation by setting in advance the power loss due to the difference in connections (cables, adapters, etc.) between the when the power calibration data are measured and when the actual DUTs are measured.

Setting Loss Compensation Table

To set the loss compensation table for each frequency, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS:DATA

Alternatively, you can configure the loss compensation table based on the data contained in a CSV file by issuing the following command:

:MMEM:LOAD:PLOS{1-4}

Also, you can save the contents of the current loss compensation table to a file by issuing the following command:

:MMEM:STOR:PLOS{1-4}

Turning ON/OFF Loss Compensation

To turn on or off the loss compensation, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS

Measuring power calibration data

- Before measuring power calibration data, you need to execute zero adjustment and calibration of the power sensor. For information on how to perform these procedures, refer to the operation manual of the power meter you are using.
- When using the power sensor calibration factor table of the E5071C, set the reference calibration factor of the power meter to 100% when calibrating the power sensor.

To set the number of power calibration data measurements at one measurement point (averaging factor), use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:AVER

To set a tolerance for power calibration, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL:NTOL

- When a tolerance for power calibration is set, if the measured value does not fall within the tolerance, even after measurement is performed during power calibration the number of times specified by the averaging factor, an error message is displayed and the power calibration is aborted. In this case, the power correction is not turned on.

To measure the power calibration data, use the following command. When the measurement is complete, the power calibration is automatically turned on.

:SOUR{1-36}:POW:PORT{1-4}:CORR:COLL

- If you issue another command during the measurement of the power calibration data by the above command, the measurement may be aborted. Therefore, when you write a program that issues multiple calibration commands in series, you should use the *OPC or some other means to ensure that

no command is executed before the preceding command completes itself.

Turning ON/OFF Power-level Error Correction

To turn on or off the power-level error correction, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR

When the power calibration data measurement initiated with the :SOUR{1-36}:POW:PORT{1-4}:CORR:COLL command is complete, the power calibration is automatically turned on.

Reading/writing power calibration data array

The power calibration data array contains data used to perform error correction for the power level at each measurement point (values obtained by subtracting the value actually measured with the power meter from the set power level value when measuring power calibration data at each measurement point).

The number of power calibration data arrays that are assigned to individual ports of individual channels can be up to 64 (16x4). To read/write one of the power calibration data arrays, use the following command:

:SOUR{1-36}:POW:PORT{1-4}:CORR:DATA

Receiver Calibration

- Overview
- Measurement of Receiver Calibration Data & Calculation of Calibration Coefficient
- Turning ON/OFF Receiver Error Correction

Other topics about Performing Calibration

Overview

The E5071C has a receiver calibration capability to calibrate the gain for each receiver in an absolute measurement.

Measurement of Receiver Calibration Data & Calculation of Calibration Coefficient

- Before starting a measurement of receiver calibration data, you must connect a THRU between the source port at which power calibration was applied and the receiver port on which you want to implement receiver calibration.

Use the following command for the measurement of receiver calibration data and calculation of calibration coefficient. The value following REC signifies a receiver port number, and the value given as an argument is a source port number. Specifying the same port to both the receiver port and source port will cause an error.

:SENS{1-160}:CORR:REC{1-4}:COLL:ACQ

- From Firmware rev 9.2, you can calibrate the receiver and transmitter port independently.

Use the following command to calibrate the receiver port:

:SENS{1-160}:CORR:REC{1-4}:COLL:RCH:ACQ

Use the following command to calibrate the transmitter port:

:SENS{1-160}:CORR:REC{1-4}:COLL:TCH:ACQ

Once the measurement is completed, calculation of the calibration coefficient takes place automatically, turning on receiver error correction automatically.

- The power calibration information on both the receiver port and source port is used to calculate calibration coefficients. The accuracy of receiver calibration will increase if power calibration is implemented for both the receiver port and the source port before starting receiver calibration. For information on power calibration, refer to Power Calibration.
- Verify the frequency of each port before starting a frequency offset sweep. Since this command does not change the frequency setting, you cannot expect a correct measurement

result unless the frequency is the same for both the receiver port and the source port.

- When using power calibration in combination with receiver calibration, you must implement it so that it sufficiently covers the frequencies of both ports or implement it twice, before and after the receiver calibration.

Turning ON/OFF Receiver Error Correction

To turn on or off the receiver error correction, use the following command:

:SENS{1-36}:CORR:REC{1-4}

- Once the measurement of receiver calibration data is completed with the :SENS{1-36}:CORR:REC{1-4}:COLL:ACQ command, receiver error correction is automatically turned on.

Scalar-Mixer Calibration

- Overview
- Measurement of Scalar-Mixer Calibration Data

Other topics about Performing Calibration

Overview

The E5071C's frequency offset function allows you to measure any device that has different input and output frequencies. To correct such measurements, you must determine the transmission frequency characteristics at different frequencies.

Scalar-mixer calibration is a method to calculate the transmission frequency characteristics at different frequencies by using a power meter.

Measurement of Scalar-Mixer Calibration Data

Setting Frequency Offset Function

You can use the following command to enable the frequency offset function before starting a measurement of scalar-mixer calibration data.

:SENS{1-36}:OFFS

- To use scalar-mixer calibration, the frequency offset function must be enabled.

Selecting Calibration Port and Type

You can use the following commands to set the port on which a measurement of scalar-mixer calibration data is implemented and the type of calibration.

- :SENS{1-36}:CORR:OFFS:COLL:METH:SMIX2
- :SENS{1-36}:CORR:OFFS:COLL:METH:SOLT1

"SMIX2" will set a scalar-mixer calibration for 2-port forward, reverse and both directions. "SOLT1" will set 1-port calibration.

- The two ports specified by "SMIX2" must be different from each other.

The setting of forward, reverse and both directions is not dependent on the order of the two specified ports but determined by specifying an appropriate command for calibration data measurement, such as

:SENS{1-36}:CORR:OFFS:COLL:OPEN

Measuring the Calibration Data

You can use the following command to measure the scalar-mixer calibration data.

- :SENS{1-36}:CORR:OFFS:COLL:OPEN
- :SENS{1-36}:CORR:OFFS:COLL:SHOR
- :SENS{1-36}:CORR:OFFS:COLL:LOAD
- :SENS{1-36}:CORR:OFFS:COLL:THRU

Measuring Power

You can use the following command to measure power.

:SENS{1-36}:CORR:OFFS:COLL:PMET

- Power measurement is required for scalar-mixer calibration, but it is not required for 1-port calibration. The setting for power measurement is the same as for the normal power calibration. For information on power calibration, refer to Power Calibration.

Calculation of scalar-mixer calibration coefficient

If the needed measurement is completed for the calibration port and type you have selected, use the following command to start calculation of calibration coefficients.

:SENS{1-36}:CORR:OFFS:COLL:SAVE

- If the necessary measurement is not completed, an error will occur. Once the calculation of calibration coefficients is completed, the measurement values are cleared despite whether they were used and error correction is turned on automatically.

Turning ON/OFF scalar-mixer error correction

To turn on or off scalar-mixer error correction, use the following command. This command is commonly used for the normal calibration; scalar-mixer calibration data are enabled while the frequency offset is turned on, and the normal calibration data are enabled while it is turned off.

:SENS{1-36}:CORR:STAT

- Once the scalar-mixer calibration coefficient has been calculated, scalar-mixer correction is automatically turned on.

Implementing scalar-mixer calibration with ECal

An ECal (Electronic Calibration) module allows you to perform scalar-mixer calibration and 1-port calibration without needing to replace the standard device.

To perform ECal, use one of the following commands:

Calibration type	Command
1-Port Calibration	:SENS{1-36}:CORR:OFFS:COLL:ECAL:SOLT1
Scalar-Mixer Calibration	:SENS{1-36}:CORR:OFFS:COLL:ECAL:SMIX2

Simply issuing one of the above commands completes some of the tasks necessary for error correction, including measuring the calibration data, calculating the calibration coefficients, and running the error correction feature. To implement scalar-mixer calibration, you must measure the power in advance. For 1-port calibration, you do not need to do this.

- Once ECal has started, you cannot interrupt the operation.
- Any command received after ECal has started will not be executed until ECal is completed. In other words, even if you issue a command with a Query, you cannot get the Query response until ECal is finished.

Partial Overwrite

- Overview
- Executing_calculation_of_calibration_coefficients_using_partial_overwrite

Other topics about Performing Calibration

Overview

The E5071C has the following calibration coefficients for full N-port calibration: Er, Es, Ed (reflection), Et (transmission), and Ex (isolation).The partial overwrite function is used to measure some of these calibration coefficients after completion of the initial calibration and then to overwrite them.

The conditions under which the calibration coefficients can be calculated by the partial overwrite are as follows:

- Calibration is completed once and valid (status other than C? or C!)
- One or more measurements for re-calculation are performed.
    - The isolation calibration coefficient, Ex, cannot be returned to the initial value, 0, once it is calculated.

If calculation of the calibration coefficients is attempted without the measurements required to execute the partial overwrite, an error message (20: Additional Standard Needed) is displayed.

Executing calculation of calibration coefficients using partial overwrite

To calculate the calibration coefficients using partial overwrite, use the following command:

:SENS{1-36}:CORR:COLL:PART:SAVE

- Before you can calculate the calibration coefficients with the partial overwrite, you must select the appropriate calibration type in the same way used for normal calibration. If calculation of the calibration coefficients is attempted without selecting the calibration type, an error message (28: Invalid Calibration Method) is displayed.

Making Measurement

Making Measurement

- Trigger System
- Starting a Measurement Cycle (triggering the instrument)
- Waiting for the End of Measurement
- Point Trigger Function
- Averaging Trigger Function

Trigger System

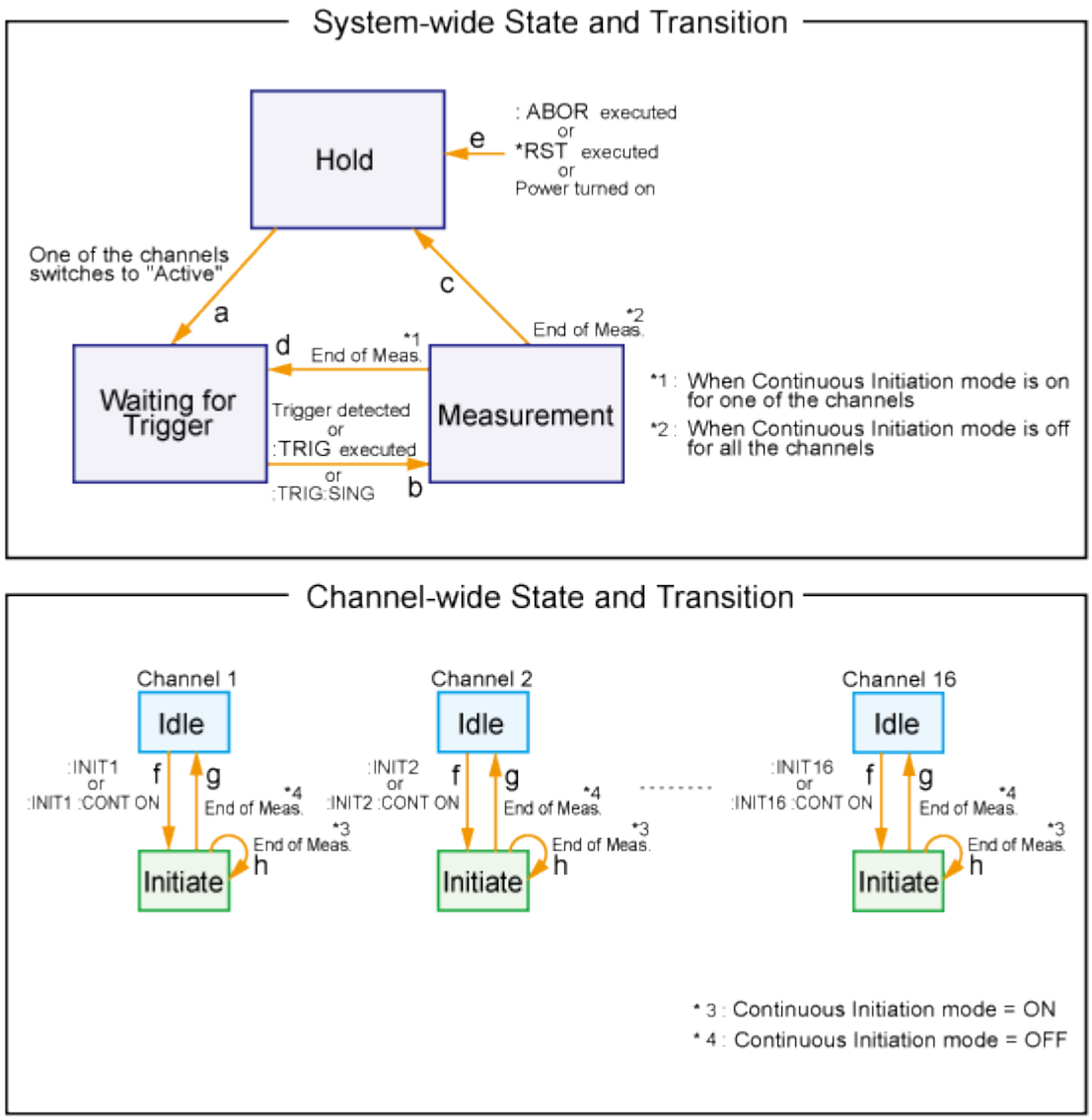
- Overview
- System-Wide States and Transitions
- Channel-wide States and Transitions

Other topics about Making Measurement

Overview

The trigger system is responsible for such tasks as detecting the start of a measurement cycle (triggering) and enabling/disabling measurement on each channel. As shown the following figure, the trigger system has two types of states: system-wide and channel-wide. The system-wide state can be "Hold", "Waiting for Trigger", or "Measurement", while the channel-wide state can be "Idle" or "Initiate".

Trigger system



e5071c312

The following subsections describe each state and explains how the trigger system switches among the states.

System-Wide States and Transitions

"Hold" State

The trigger system switches to "Hold" state when one of the following commands has been executed (arrow "e" in the above figure). Also, turning on the power to the instrument puts the trigger system into "Hold" state. When the power is turned on, however, continuous initiation mode is on for channel 1 and the trigger source is set to "Internal"; accordingly, the trigger system immediately switches to "Waiting for Trigger" state and subsequently repeats transitions between "Measurement" and "Waiting for Trigger" states.

- :ABOR
- *RST

When the trigger system is in "Hold" state and one of the channels switches to "Initiate" state (arrow "f" in the above figure), the trigger system switches to "Waiting for Trigger" state (arrow "a" in the above figure).

"Waiting for Trigger" State

When the trigger system is in "Waiting for Trigger" state and either the instrument is triggered (i.e., a trigger is detected) or one of the following commands is executed, the trigger system switches to "Measurement" state (arrow "b" in the above figure)

- :TRIG
- :TRIG:SING

As shown in the table below, the instrument is triggered differently depending on which trigger source is specified. To specify the trigger source, use the :TRIG:SOUR command.

Trigger Source	How instrument is triggered
Internal trigger	The instrument is automatically triggered within itself.
External trigger	The instrument is triggered when a trigger signal is input through the Ext Trig terminal or the handler interface
Bus trigger	The instrument is triggered when the *TRG command is issued.
Manual trigger	The instrument is triggered when you press **Trigger** > **Trigger** on the front panel.

"Measurement" State

In "Measurement" state, the instrument waits for the elapse of the sweep delay time (set by the :SENS{1-36}:SWE:DEL) and then starts a measurement cycle; this process is performed sequentially on each of those channels that were in "Initiate" state immediately before the transition to this state, in ascending order of channel number.

When the instrument has finished measuring all of the active channels, the trigger system behaves in one of the following ways depending on the setting of the continuous initiation mode.

If continuous initiation mode is off for all channels:

The trigger system switches to "Hold" state (arrow "c" in the above figure).

If continuous initiation mode is on for one of the channels:

The trigger system switches to "Waiting for Trigger" state (arrow "d" in the above figure).

Channel-wide States and Transitions

"Idle" State

A channel switches to "Initiate" state when one of the following commands has been executed (arrow "f" in the above figure).

- :INIT{1-36}
- :INIT{1-36}:CONT("ON" specified)

"Initiate" State

A channel in this state is measured just before the entire system switches to "Measurement" state.

When the instrument has finished measuring a channel, the channel behaves in one of the following ways depending on the setting of the continuous initiation mode (set by the :INIT{1-36}:CONT).

If continuous initiation mode is off:

The channel switches to "Idle" state (arrow "g" in the above figure).

If continuous initiation mode is on:

The channel remains in "Initiate" state (arrow "h" in the above figure).

Starting a Measurement Cycle (Triggering the Instrument)

- Configuring the Instrument
- Starting Measurement on Demand

Other topics about Making Measurement

Configuring the Instrument to Automatically Perform Continuous Measurement

1. Use the :INIT{1-36}:CONT command to turn on continuous initiation mode for the channels you want to measure and turn the mode off for any other channel.
2. Issue the :TRIG:SOUR command to set the trigger source to Internal trigger.

Starting Measurement on Demand

1. Use the :INIT{1-36}:CONT command to turn on continuous initiation mode for the channels you want to measure and turn the mode off for any other channel.
2. Issue the :TRIG:SOUR command to set the trigger source to "Bus Trigger".
3. Trigger the instrument whenever you want to perform measurement. An external controller can trigger the instrument by using one of the following three commands:

Command	Can *OPC? command be used to wait for end of sweep?	Applicable trigger source
*TRG	No	Bus trigger only
:TRIG		External trigger Bus trigger Manual trigger
:TRIG:SING	Yes	

4. Repeat step 3 to start the next measurement cycle.

Waiting for the End of Measurement

- Using the Status Register
- Using :TRIG:SING Command
- Using Wait Time

Other topics about Making Measurement

Using the Status Register

The status of the E5071C can be detected through the status registers. This section explains how to detect the end of measurement by using the status registers.

Measurement status is reported by the operation status condition register. An SRQ (service request) is useful when creating a program that uses the information reported by this register to detect the end of measurement.

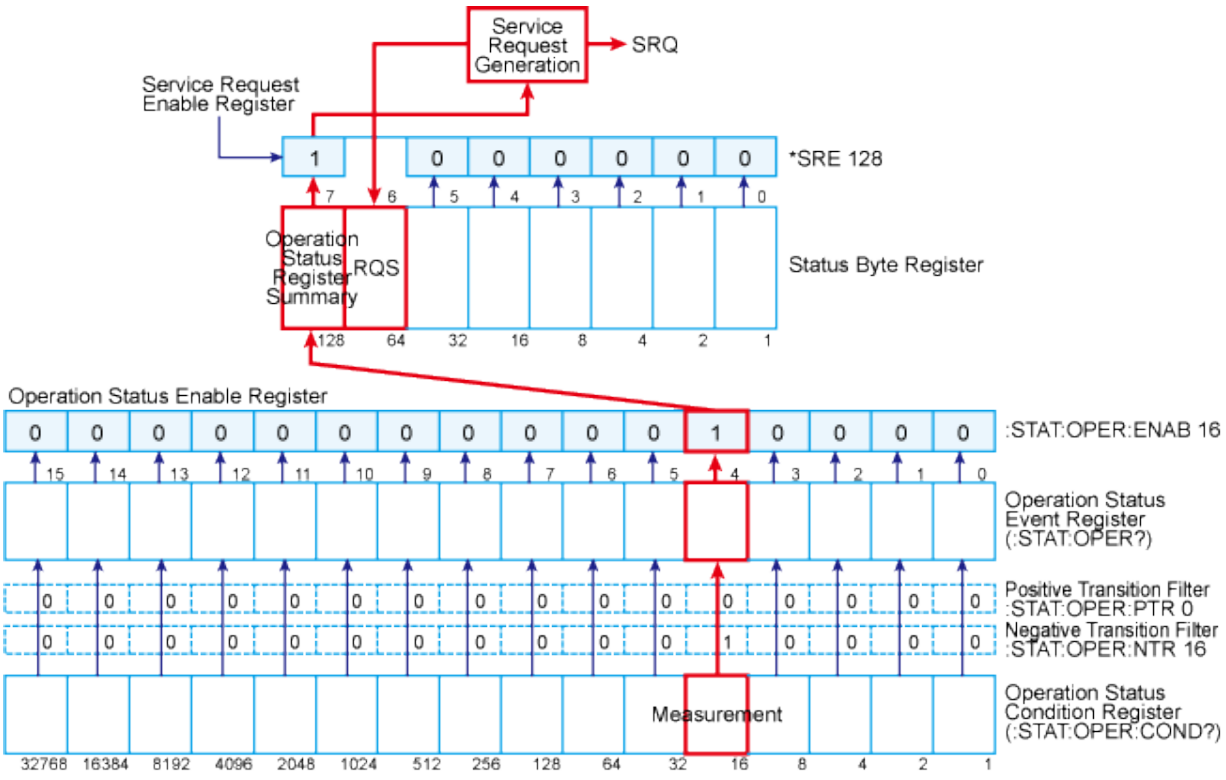
To detect the end of measurement via an SRQ, use one of the following commands:

- *SRE
- :STAT:OPER:ENAB
- :STAT:OPER:PTR
- :STAT:OPER:NTR

Follow these steps to utilize an SRQ:

1. Configure the E5071C so that it generates an SRQ when the operation status condition register's bit 4 (a bit that is set to 1 during measurement) is changed from 1 to 0.
2. Trigger the instrument to start a measurement cycle.
3. When an SRQ is generated, the program interrupts the measurement cycle.

SRQ generation sequence (at end of measurement)



e5071c356

Sample Program

See the Waiting for Trigger (SRQ).

Using :TRIG:SING Command

When you trigger the instrument by issuing the :TRIG:SING command, you can use the *OPC command to wait until the measurement cycle is completed.

Sample Program

See the Waiting for Trigger (OPC?).

Using Wait Time

Before creating your program, actually measure the time between the start and end of the measurement cycle. Then code your program so that the controller waits for the actually measured time by using the appropriate command (for example, the WAIT command for HTBasic). This is a straightforward method, but care must be taken: an incorrect wait time could result in an unexpected error.

Point Trigger Function

- Point Trigger
- Switching Mode

Other topics about Making Measurement

Point Trigger

Point trigger is a method used to measure a single point each time the instrument is triggered. You can perform this type of measurement by turning ON/OFF the point trigger function.

Switching Mode

You can use the following command to toggle between the point trigger's ON/OFF status.

:TRIG:POIN

- If the trigger source is internal, the point trigger function does not work even if it is specified to ON.

Low-latency external trigger

When the trigger source is an external trigger and the point trigger function is on, the low-latency external trigger function is available. The low-latency external trigger function is used to set the external trigger delay time each time a trigger for a measurement point is given.

The setting of the low-latency external trigger function is valid for all channels.

Setting the low-latency external trigger

To toggle ON/OFF the low-latency external trigger function, use the following command:

:TRIG:EXT:LLAT

If the trigger source is not an external trigger, the low-latency external trigger function does not work even if it is specified to ON, nor does it work when the point trigger function is off.

To set the external trigger delay time, use the following command:

:TRIG:EXT:DEL

Averaging Trigger Function

- Averaging Trigger
- Toggling Mode

Other topics about Making Measurement

Averaging Trigger

The averaging trigger function is used to execute the number of sweeps specified as the averaging factor with a single trigger when the sweep averaging function is used. When a trigger is given, the sweep count is cleared, the sweep is executed the number of times specified by the averaging count, and then the instrument goes into Hold status.

The setting of the averaging trigger function is valid for all channels. Note that you can set the sweep averaging function for each channel.

Toggling Mode

To toggle ON/OFF the averaging trigger function, use the following command:

:TRIG:AVER

- When the point trigger function is on, its setting has priority and the setting of the averaging trigger is ignored. More specifically, the required number of triggers is: number of measurement points in a single sweep x averaging factor.
- When the averaging function is off for the active channel, the averaging trigger function does not work for the channel. When a trigger is given, the sweep is executed only once, and then the instrument goes into Hold status.

Reading-Writing Measurement Data

Reading-Writing Measurement Data

- Data Transfer Format
- Internal Data Processing
- Retrieving Measurement Results
- Entering Data into a Trace

Data Transfer Format

- Overview
- ASCII Transfer Format
- Integer Format
- Floating-Point Number Format
- Binary Transfer Format

Other topics about Reading-Writing Measurement Data

Overview

When you transfer data using the one of the following commands, you can choose among ASCII transfer format, IEEE 64-bit floating point binary transfer format and IEEE 32-bit floating point binary transfer format.

- The instrument always uses the ASCII transfer format when you transfer data without using any of the following commands:
  - :CALC{1-160}:BLIM:REP?
  - :CALC{1-160}:DATA:FDAT
  - :CALC{1-160}:DATA:FMEM
  - :CALC{1-160}:DATA:SDAT
  - :CALC{1-160}:DATA:SMEM
  - :CALC{1-160}:FUNC:DATA?
  - :CALC{1-160}:LIM:DATA
  - :CALC{1-160}:LIM:REP?
  - :CALC{1-160}:LIM:REP:ALL?
  - :CALC{1-160}:RLIM:DATA
  - :CALC{1-160}:RLIM:REP?
  - :SENS{1-160}:CORR:COEF?
  - :SENS{1-160}:FREQ:DATA?
  - :SENS{1-160}:SEGM:DATA
  - :SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA
  - :SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA
  - :SOUR{1-160}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS
  - :SOUR{1-160}:POW:PORT{1-4}:CORR:DATA

To set the data transfer format, use the following command:

:FORM:DATA

- Executing the :SYST:PRES or *RST does not affect the current setting of the data transfer format.

ASCII Transfer Format

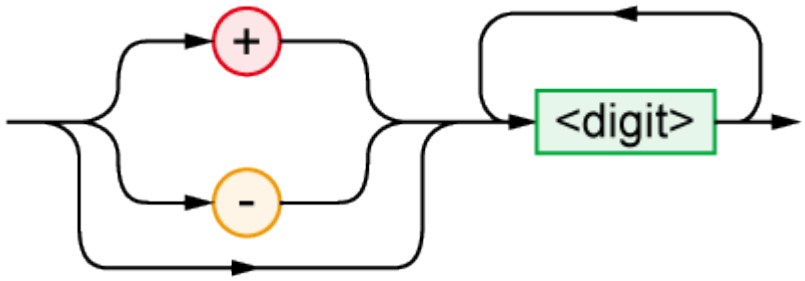
When you select the ASCII transfer format as the data transfer format, numbers are transferred as ASCII bytes, each of which corresponds to one of the formats shown below. Note that numbers are separated from one another with a comma (,) in accordance with the IEEE 488.2 specification.

- Numeric data strings vary in length. Keep this in mind when you extract some data from retrieved numeric data strings in your program.

Integer Format

The figure below shows this format. Numbers are expressed as integers. For example, 201 is expressed as "+201" or "201."

Integer format

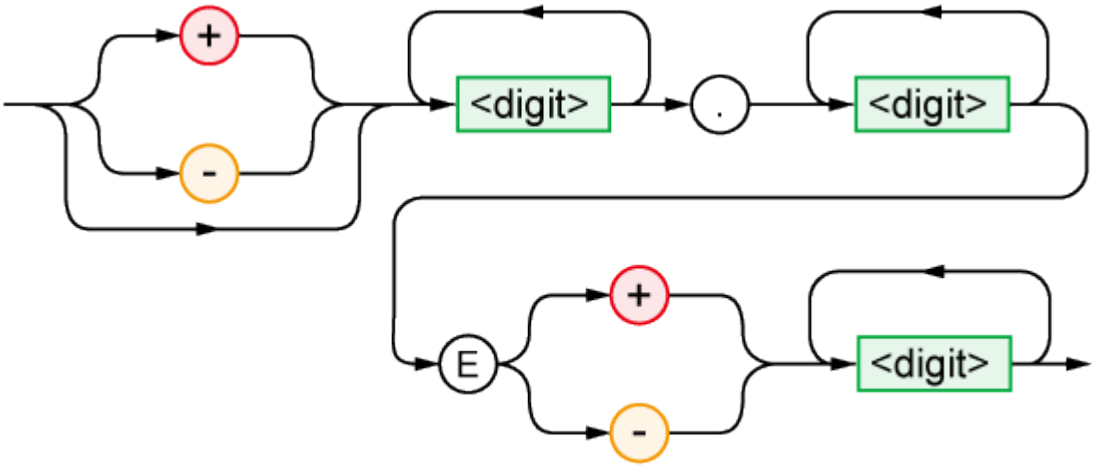


e5071c351

Floating-Point Number Format

The figure below shows this format. Numbers are expressed with floating points. For example, 1000 is expressed as "+1.00000000000E+003."

Floating-point number format



e5071c352

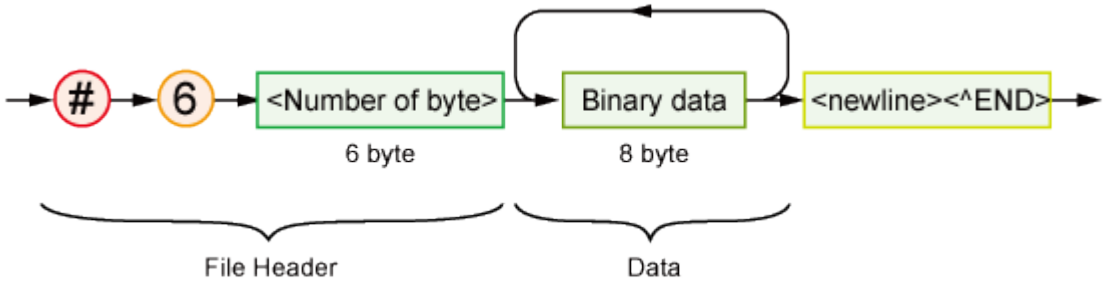
Binary Transfer Format

You can select the binary transfer format from the IEEE 64-bit floating point format or the IEEE 32-bit floating point format depending on the controller you use.

IEEE 64-bit floating point format

When you select the IEEE 64-bit floating point binary transfer format as the data transfer format, numbers are transferred in the format shown in the figure below.

Binary transfer format

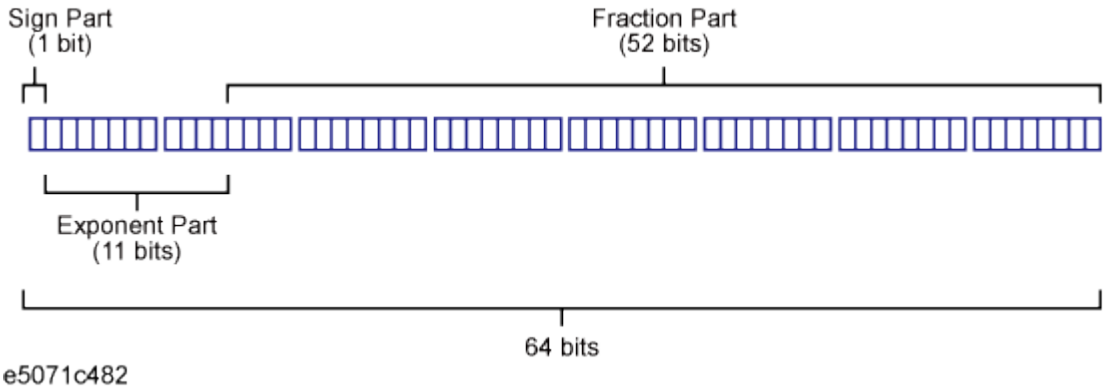


e5071c480

This data transfer format uses a header that consists of a sharp character (#), a number of 6 (which indicates the byte size of the <number of bytes transferred> part), and the <number of bytes transferred> part in this order. The header is followed by the binary data (each number consists of 8 bytes and the total is the byte size indicated by <number of bytes transferred>) and the message terminator <new line>^END.

The binary data is expressed in the IEEE 754 64-bit floating-point number format shown in the figure below.

64-bit floating point format

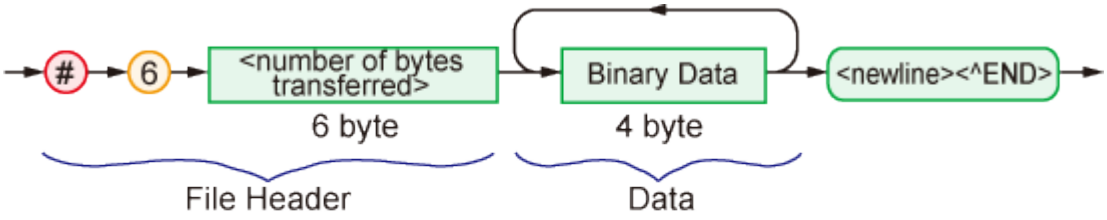


e5071c482

IEEE 32-bit floating point format

When you select the IEEE 32-bit floating point binary transfer format as the data transfer format, numbers are transferred in the format shown in the figure below.

IEEE 32-bit floating point binary transfer format

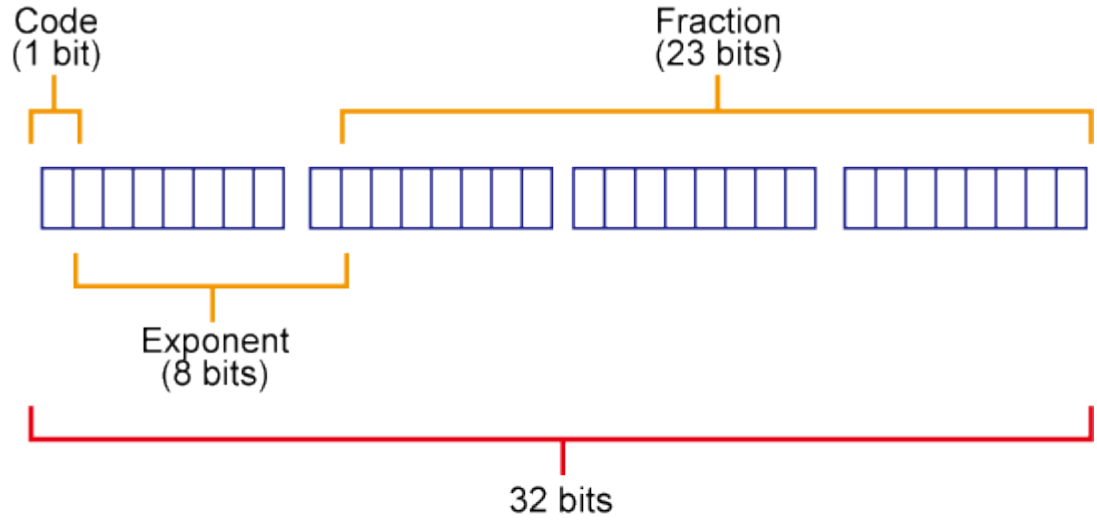


e5071c439

This data transfer format uses a header that consists of a sharp character (#), a number of 6 (which indicates the byte size of the <number of bytes transferred> part), and the <number of bytes transferred> part in this order. The header is followed by the binary data (each number consists of 4 bytes and the total is the byte size indicated by <number of bytes transferred>) and the message terminator <new line>^END.

The binary data is expressed in the IEEE 754 32-bit floating-point number format shown in the figure below.

32-bit floating point data



e5071c440

Byte order

When you opt to perform binary transfer, you can configure the instrument to transfer the bytes of the data in one of the following two byte orders:

NORMal

Transfer begins with the byte that contains the MSB (Most Significant Bit); that is, the leftmost byte in 64 bit floating point format and 32 bit floating point data.

SWAPped

Transfer begins with the byte that contains the LSB (Least Significant Bit); that is, the rightmost byte in 64 bit floating point format and 32 bit floating point data.

To set the byte order, use the following command:

:FORM:BORD

- Executing the :SYST:PRES or *RST does not affect the current setting of the byte order.

Internal Data Processing

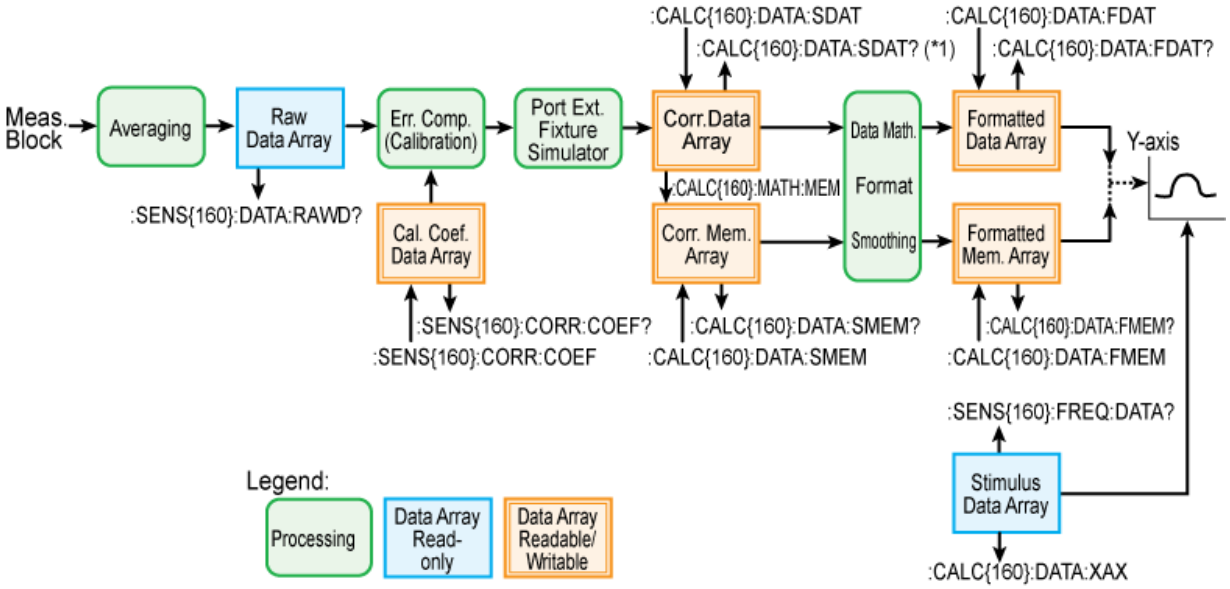
- Data Flow
- Internal Data Arrays

Other topics about Reading-Writing Measurement Data

Data Flow

The following figure provides an overview of the E5071C's internal data processing flow.

E5071C's data processing flow



e5071c576

(*1) :CALC{1-160}:DATA:CORR? allows you to get the other S-parameter than the selected.

Internal Data Arrays

Raw data arrays

Raw data array contains uncorrected data. The data is obtained by performing IF range correction, ratio calculation, port characteristics correction and sweep averaging. This data will be used in error correction.

To read one of the raw data arrays, use the following command:

:SENS{1-160}:DATA:RAWD

Calibration Coefficient Data Arrays

A calibration coefficient data array contains the calibration coefficients calculated based on the results of measurement performed with standard devices.

Commands are available for reading or writing calibration coefficient data arrays. To read or write, first use the following command:

:SENS{1-160}:CORR:COEF

- If any calibration coefficient is interpolated, the interpolated calibration coefficient data array will be retrieved. Once a calibration coefficient data array has been written, execute the command SENS{1-160}:CORR:COEF:SAVE to validate it

Corrected data arrays

A corrected data array contains the corrected data obtained by performing error correction, port extension compensation (calibration), Fixture Simulator operations on the raw measured data of S-Parameter specified for each trace of each channel. Each data element is stored as a complex number (Re/Im).

To read/write one of the corrected data arrays, use the following command:

:CALC{1-160}:DATA:SDAT

Corrected data arrays (S-Parameter)

You can get the S-Parameter measurement result which the analyzer has.

For example, when you measure 2 full ports calibration, the analyzer measures not only the S-Parameter you selected but also four S-Parameters at the measurement. In this case, you can get the other S-Parameter.

To read one of the corrected S-Parameter data arrays, use the following command:

:SENS{1-160}:DATA:CORR

Corrected memory arrays

When the :CALC{1-160}:MATH:MEM command is executed on a particular corrected data array, its copy is stored into the corrected memory array corresponding to that corrected data array.

To read/write one of the corrected data arrays, use the following command:

:CALC{1-160}:DATA:SMEM

Formatted data array

A formatted data array contains the formatted data (values to be displayed) obtained by performing data math operations, measurement parameter conversion, and smoothing on a particular corrected data array.

Regardless of the data format, it contains two data elements per measurement point as shown in the following table:

Data format	Data element (primary value)	Data element (secondary value)
log magnitude	log magnitude	Always 0
Phase	Phase	Always 0
Group delay	Group delay	Always 0
Smith chart (Lin)	Liner magnitude	Phase
Smith chart (Log)	log magnitude	Phase
Smith chart (Re/Im)	Real part of a complex number	Imaginary part of a complex number
Smith chart (R+jX)	Resistance	Reactance
Smith chart (G+jB)	Conductance	Susceptance
Polar (Lin)	Liner magnitude	Phase
Polar (Log)	log magnitude	Phase
Polar (Re/Im)	Real part of a complex number	Imaginary part of a complex number
Liner magnitude	Liner magnitude	Always 0
SWR	SWR	Always 0
Real number	Real part of a complex number	Always 0
Imaginary number	Imaginary part of a complex number	Always 0
Expanded	Expanded phase	Always 0

phase		

To read/write one of the formatted data arrays, use the following command:

:CALC{1-160}:DATA:FDAT

Formatted memory arrays

A formatted memory array contains the formatted data (values to be displayed) obtained by performing data math operations, measurement parameter conversion, and smoothing on a particular corrected memory array.

To read/write one of the formatted memory arrays, use the following command:

:CALC{1-160}:DATA:FMEM

Stimulus data arrays

A stimulus data array contains the stimulus values for all measurement points.

The instrument retains 160 stimulus data arrays at maximum, each of which is associated with one of the 160 channels. Stimulus data arrays are read-only. To retrieve one of the stimulus data arrays, use the following command:

:SENS{1-160}:FREQ:DATA?

Retrieving Measurement Results

- Overview
- Retrieving Internal Data Arrays
- Sample Program

Other topics about Reading-Writing Measurement Data

Overview

Internal data arrays allows you to retrieve all measurement results throughout a particular trace. Alternatively, markers allow you to retrieve measurement results at your specified points. For information on how to retrieve marker values, refer to Retrieving measurement results at marker positions.

Retrieving Internal Data Arrays

You can chose between the ASCII and binary data transfer formats when you retrieve internal data arrays. For more information, please refer to Data Transfer Format.

Sample Program

See Reading Data in Ascii Format and Reading Data in Binary Format.

Entering Data into a Trace

- Overview
- Sample Program

Other topics about Reading-Writing Measurement Data

Overview

You can change the data/memory trace on the LCD by writing the new data into the Formatted data array/Formatted memory arrays.

When you write data into formatted data/memory array, you can choose either the ASCII or binary transfer format (see Data Transfer Format).

Using the ASCII Transfer Format to Write Formatted Data Arrays (write_a.htb) and Using the Binary Transfer Format to Write Formatted Data Arrays (write_b.htb) show sample programs that demonstrate how to write data into formatted data arrays. The sample program in Using the ASCII Transfer Format to Write Formatted Data Arrays (write_a.htb) uses the ASCII transfer format while the sample in Using the Binary Transfer Format to Write Formatted Data Arrays (write_b.htb) uses the binary transfer format. You can find the source files of these programs, named write_a.htb and write_b.htb, on the sample program disk.

Each of the sample programs holds the sweep on channel 1, retrieves the data from a specified file (a file saved measurement data using the :MMEM:STOR:FDAT command), and populates trace 1 for channel 1 with the retrieved data.

Sample Program

See Writing Data in Ascii Format and Writing Data in Binary Format.

Analyzing Data

Analyzing Data

- Retrieving Measurement Results at Specified Points
- Searching for Positions That Match Specified Criteria
- Bandwidth Search
- Notch Search
- Statistical Analysis
- Analysis Using the Fixture Simulator
- Analysis in Time Domain (time domain function)
- Analyzing impedance

Retrieving Measurement Results at Specified Points

- Overview
- Showing/Hiding Markers
- Turning ON/OFF Reference Marker Mode
- Setting and Retrieving Stimulus Value at Marker Positions
- Retrieving Measurement Results at Marker Positions

Other topics about Analyzing Data

Overview

Markers allow you to retrieve measurement results at specified points. You can use up to nine markers for each trace, and you can move them to any point on the trace. In addition to the regular markers, you can use a reference marker.

Showing/Hiding Markers

To show or hide markers, including the reference marker, use the following command:

:CALC{1-36}:MARK{1-10}

- You can move markers or retrieve the data at a marker even when the markers are hidden.
- The display of the reference marker is turned on or off when you turn On or Off Reference Marker mode.

Turning ON/OFF Reference Marker Mode

Turning on Reference Marker mode provides relative marker values with respect to the reference marker (by subtracting the value at the reference marker from the value at a particular marker).

To turn On or Off Reference Marker mode, use the following command:

:CALC{1-36}:MARK:REF

Setting and Retrieving Stimulus Value at Marker Positions

To set (or change along the frequency axis) the stimulus value at a particular marker or the reference maker or to retrieve the current stimulus value, use the following command:

:CALC{1-36}:MARK{1-10}:X

When Reference Marker mode is on, the stimulus value at a regular marker is a relative stimulus value obtained by subtracting the stimulus value at the reference marker from the actual stimulus value at that particular marker.

Retrieving Measurement Results at Marker Positions

To retrieve the measurement results (response values) at a particular marker or the reference marker, use the following command:

:CALC{1-36}:MARK{1-10}:Y?

When Reference Marker mode is on, the response value at a regular marker is a relative value obtained by subtracting the response value at the reference marker from the actual response value at that particular marker.

Searching for Positions Matching Specified Criteria

- Overview
- Using Marker Search
- Analysis Commands
- Sample Program

Other topics about Analyzing Data

Overview

You can search for a position that matches specified criteria by using the Marker Search feature or analysis commands.

Using Marker Search

- Marker Search is available whether the markers are shown or hidden.

Setting the Search Range

You can use either the entire sweep range or a user-defined range for the marker search range by using the following command:

:CALC{1-36}:MARK:FUNC:DOM

When you opt to use a user-defined range, use the following commands to set the range:

Description	Command
Start value (lower limit value)	:CALC{1-36}:MARK:FUNC:DOM :STAR
Stop value (upper limit value)	:CALC{1-36}:MARK:FUNC:DOM :STOP

You can also select whether to specify the marker search range independently for each trace by using the following command.

:CALC{1-36}:MARK:FUNC:DOM:COUP

Selecting a Search Type

Marker Search allows you to choose from the following eight search types:

- Maximum value
- Minimum value
- Peak (3 types)
  - Maximum peak (for a positive peak), minimum peak (for a negative peak)

- Peak nearest to the marker position on its left-hand side
- Peak nearest to the marker position on its right-hand side

▪ Target (3 types)

- Peak nearest to the marker position
- Target nearest to the marker position on its left-hand side
- Target nearest to the marker position on its right-hand side

To select a search type, use the following command:

:CALC{1-36}:MARK{1-10}:FUNC:TYPE

Defining a Peak

You can define a peak by specifying the lower limit for the peak excursion value and polarity (positive or negative peak).

To define a peak, use the following commands:

Lower limit for the peak excursion value	:CALC{1-36}:MARK{1-10}:FUNC:PEXC
Polarity	:CALC{1-36}:MARK{1-10}:FUNC:PPOL

Defining a Target

You can define a target by specifying the target value (response value) and transitional direction (positive or negative value change).

To define a target, use the following commands:

Target value	:CALC{1-36}:MARK{1-10}:FUNC:TARG
Transitional direction	:CALC{1-36}:MARK{1-10}:FUNC:TTR

Performing Marker Search

To perform Marker Search, use the following command:

:CALC{1-36}:MARK{1-10}:FUNC:EXEC

To turn On or Off the Search Tracking feature, which performs Marker Search every time the trace is updated, use the following command:

:CALC{1-36}:MARK{1-10}:FUNC:TRAC

Retrieving Search Results

Performing Marker Search moves the marker to the points that matches the search criteria, so you can obtain the search results by retrieving the

marker value. For information on how to retrieve marker values, refer to Setting (changing) and retrieving stimulus value at marker positions and Retrieving measurement results at marker positions.

Analysis Commands

You can use the analysis Commands to perform search and analysis.

Setting the Search (Analysis) Range

You can use either the entire sweep range or a user-defined range as the search (analysis) range by using the following command:

:CALC{1-36}:FUNC:DOM

When you opt to use a user-defined range, use the following commands to set the range:

Start value (lower limit value)	:CALC{1-36}:FUNC:DOM:STAR
Stop value (upper limit value)	:CALC{1-36}:FUNC:DOM:STOP

You can also select whether to specify the marker search (analysis) range independently for each trace by using the following command:

:CALC{1-36}:FUNC:DOM:COUP

Selecting the Search (Analysis) Type

The analysis commands allows you to choose from the following five search types:

- Maximum value
- Minimum value
- Maximum peak (for a positive peak), minimum peak (for a negative peak)
- All peaks
- All targets

In addition, you can choose from the following three analysis types:

- Difference between the maximum and minimum values
- Standard deviation
- Average

To select the search (analysis) type, use the following command:

:CALC{1-36}:FUNC:TYPE

Defining a Peak

You can define a peak by specifying the lower limit for the peak excursion value and polarity (positive or negative peak).

To define a peak, use the following commands:

Lower limit for the peak excursion value	:CALC{1-36}:FUNC:PEXC
Polarity	:CALC{1-36}:FUNC:PPOL

Defining a Target

You can define a target by specifying the target value (response value) and transitional direction (positive or negative value change).

To define a target, use the following commands:

Target value	:CALC{1-36}:FUNC:TARG
Transitional direction	:CALC{1-36}:FUNC:TTR

Performing Search (Analysis)

To perform search (analysis), use the following command:

:CALC{1-36}:FUNC:EXEC

Retrieving Search (Analysis) Results

To retrieve search (analysis) results, use the following command:

:CALC{1-36}:FUNC:DATA?

The number of data items contained in search (analysis) results differ depending on the search (analysis) type and the number of points found by the search operation. To retrieve the number of data items, use the following command:

:CALC{1-36}:FUNC:POIN?

Sample Program

See Peak Search.

Bandwidth Search

- Overview
- Setting Bandwidth Definition Value
- Retrieving Bandwidth Search Results
- Sample Program

Other topics about Analyzing Data

Overview

The E5071C has a feature called Bandwidth Search. This feature searches for the cutoff points on both right- and left-hand sides of the marker position on the trace.

- Bandwidth (BW= high-low)
- Center frequency (cent= {high+low}/2))
- Q value (Q=cent/BW)
- Loss (response value at the marker position)

Where high is the right-hand cutoff point frequency;low is the left-hand cutoff point frequency.

Setting Bandwidth Definition Value

Bandwidth Search finds a point whose response value is different, by the amount defined as the bandwidth definition value, from the response value at the marker position, and identifies that point as the cutoff point.

To set the bandwidth definition value, use the following command:

:CALC{1-36}:MARK{1-10}:BWID:THR

Retrieving Bandwidth Search Results

Once you have moved the marker to the appropriate position using Marker Search or some other feature, you can retrieve the results of Bandwidth Search by using the following command:

:CALC{1-36}:MARK{1-10}:BWID:DATA?

Also, you can use the following command to control whether to display the results of Bandwidth Search on the LCD:

:CALC{1-36}:MARK:BWID

You can retrieve the results of Bandwidth Search even after you have turned off the display of markers or the results of Bandwidth Search.

Sample Program

See Bandwidth Search.

Notch Search

- Overview
- Setting the notch definition value
- Displaying the notch search result
- Reading out the notch search result

Other topics about Analyzing Data

Overview

The notch search function is used to obtain the bandwidth, center frequency, cutoff points (high-frequency side and low-frequency side), Q and insertion loss of a trace based on the active marker position. The notch search function starts at the left side of the active marker position, and ends when points that meet the condition are found.

- Bandwidth (BW = high – low)
- Center frequency (cent = (high +low)/2)
- Q value (Q = cent/BW)
- Loss (response value at marker position)

Where, high is the right-hand cutoff point frequency, and low is the left-hand cutoff point frequency.

- For more information on notch search, see Determining the bandwidth of a trace (Notch Search)

Setting the notch definition value

The notch search function finds a point whose response value is different, by the amount defined as the notch definition value, than the response value at the marker position, and identifies that point as the cutoff point.

To set the notch definition value, use the following command:

:CALC{Ch}:MARK{1-10}:NOTC:THR

Displaying the notch search result

The following command is used to control whether to display the notch search result on the LCD:

:CALC{Ch}:MARK:NOTC

Reading out the notch search result

Once the marker is moved to an appropriate position using the marker search function or some other function, it is able to retrieve the notch search result using the following command:

:CALC{Ch}:MARK{1-10}:NOTC:DATA

- It is able to retrieve the notch search result regardless of whether the marker display and the notch search result display is on/off.

Statistical Analysis

The E5071C provides a mechanism that analyzes trace statistics. You can analyze the following statistics:

- Average
- Standard deviation
- Difference between the maximum and minimum values

To retrieve the results of statistical analysis, use the following command:

:CALC{1-36}:MST:DATA?

Also, you can use the following command to control whether to display the results of statistical analysis on the LCD:

:CALC{1-36}:MST

- You can retrieve the results of statistical analysis even after you have turned off the display showing these results.

Alternatively, you can use the analysis commands to analyze the trace statistics. When you use the analysis commands, you can analyze the trace data in a specific range as well as the trace data throughout the entire sweep range. For information on how to use the analysis commands, refer to Analysis Commands.

Other topics about Analyzing Data

Analysis Using Fixture Simulator

- Overview
- Matching Circuit Embedding
- Port Impedance Conversion
- Network De-embedding
- 4-Port Network Embedding/De-embedding
- Balance-Unbalance Conversion (only 3-port/4-port models)
- Differential Matching Circuit Embedding (only 3-port/4-port models)
- Differential Port Impedance Conversion (only 3-port/4-port models)
- Sample Program

Other topics about Analyzing Data

Overview

The Fixture Simulator provides the following functions:

- Matching Circuit Embedding
- Port Impedance Conversion
- Network De-embedding
- 4-Port Network Embedding/De-embedding
- Balance-Unbalance Conversion (only 3-port/4-port models)
- Differential Matching Circuit Embedding (only 3-port/4-port models)
- Differential Port Impedance Conversion (only 3-port/4-port models)

Before you can use any of the features listed above, you must turn on the desired feature individually and issue the following command to turn on the Fixture Simulator:

```
:CALC{1-36}:FSIM:STAT
```

Matching Circuit Embedding

The Matching Circuit feature simulates the characteristics that the DUT would exhibit when connected to a matching circuit.

On/Off

To turn on or off the Matching Circuit, use the following command:

```
:CALC{1-36}:FSIM:SEND:PMC:STAT
```

You can only turn on or off Matching Circuit for all the ports, not for each port individually. However, any port whose circuit type is set to "None" behaves as if this feature were permanently off.

Configuring the Matching Circuit Settings

To select a circuit type, use the following command:

:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}

To set the circuit constant, use the following commands:

Circuit constant	Command
C	:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:C
G	:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:G
L	:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:L
R	:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:PAR:R

When you want to use a user file to define the circuit type, specify the file by using the following command:

:CALC{1-36}:FSIM:SEND:PMC:PORT{1-4}:USER:FIL

Port Impedance Conversion

The Port Impedance Conversion feature converts the measurement results with a port impedance of 50 ohm to the characteristics of a different port impedance.

On/Off

To turn on or off Port Impedance Conversion, use the following command:

:CALC{1-36}:FSIM:SEND:ZCON:STAT

You can only turn on or off Port Impedance Conversion for all of the ports, not for each port individually. However, any port with ZO set to 50 ohm behaves as if this feature were permanently off.

Setting the Z0 Value

To set the target port impedance, use the following command:

:CALC{1-36}:FSIM:SEND:ZCON:PORT{1-4}:Z0

:CALC{1-36}:FSIM:SEND:ZCON:PORT{1-4}:REAL

:CALC{1-36}:FSIM:SEND:ZCON:PORT{1-4}:IMAG

Network De-embedding

The Network De-embedding feature eliminates any network that can cause error between the calibration level and the DUT.

On/Off

To turn on or off Network De-embedding, use the following command:

:CALC{1-36}:FSIM:SEND:DEEM:STAT

You can only turn on or off Network De-embedding for all of the ports, not for each port individually. However, any port whose Network De-embedding type is set to "None" behaves as if this feature were permanently off.

Selecting a Type

To select a Network De-embedding type, use the following command:

:CALC{1-36}:FSIM:SEND:DEEM:PORT{1-4}

Specifying the File

To specify the file that defines the criteria for Network De-embedding, use the following command:

:CALC{1-36}:FSIM:SEND:DEEM:PORT{1-4}:USER:FIL

4-Port Network Embedding/De-embedding

The 4-port Network Embedding/De-embedding feature is provided to embed or de-embed a network (defined in a 4-port touchstone file) between the calibration surface and the DUT.

On/Off

To turn on or off the 4-port network embedding/de-embedding feature, use the following command:

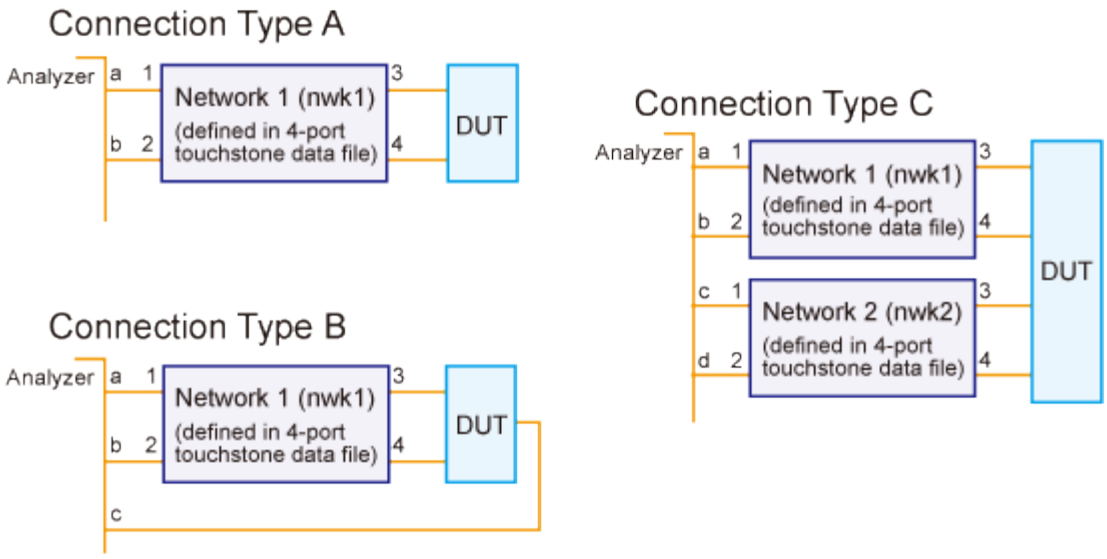
:CALC{1-36}:FSIM:EMB:STAT

Setting Topology (connection method between analyzer and DUT)

To select a connection type, use the following command:

:CALC{1-36}:FSIM:EMB:TYPE

Connection type



e5071c313

To assign the ports (ports a through d in Connection_type), use the appropriate command that matches your selected connection type, as identified in the following table:

Connection type	Command
A	:CALC{1-36}:FSIM:EMB:TOP:A:PORT
B	:CALC{1-36}:FSIM:EMB:TOP:B:PORT
C	:CALC{1-36}:FSIM:EMB:TOP:C:PORT

Selecting Processing Type (embedding/de-embedding)

To select a network processing type, use the following command:

:CALC{1-36}:FSIM:EMB:NETW{1-2}:TYPE

Specifying File

To select a file that defines the criteria for network embedding/de-embedding, use the following command:

:CALC{1-36}:FSIM:EMB:NETW{1-2}:FIL

Balance-Unbalance Conversion

The Balance-Unbalance Conversion feature converts the measurement results obtained in an unbalanced state to the characteristics of a balanced state. You can select mixed mode S parameter, balance and CMRR as the measurement parameter when you turn on Balance-Unbalance Conversion.

On/Off

You can turn on or off Balance-Unbalance Conversion for each trace individually. To turn on or off Balance-Unbalance Conversion, use the following command:

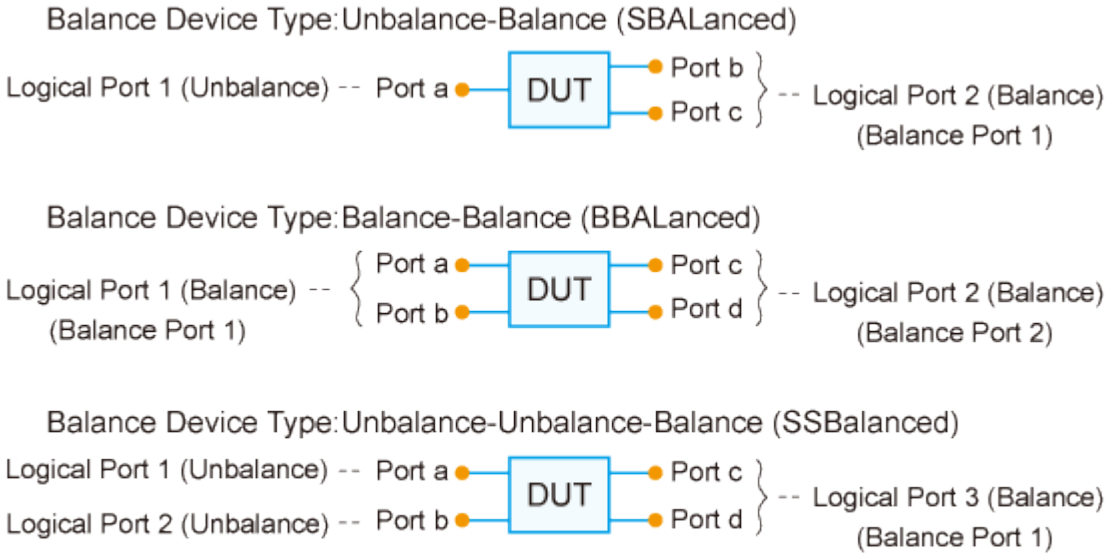
:CALC{1-36}:FSIM:BAL:PAR{1-36}:STAT

Setting the Topology

To select a balance device type, use the following command:

:CALC{1-36}:FSIM:BAL:DEV

Balance device type



e5071c302

To assign the ports (ports a through d in Balance_device_type), use the command that matches your selected device type, as identified in the following table:

Device Type	Command
Unbalance-balance (SBALanced)	:CALC{1-36}:FSIM:BAL:TOP :SBAL
Balance-balance (BBALanced)	:CALC{1-36}:FSIM:BAL:TOP :BBAL
Unbalance-unbalance-balance (SSBalanced)	:CALC{1-36}:FSIM:BAL:TOP :SSB

Selecting the Measurement Parameter

To select the measurement parameter, use the command that matches your selected device type, as identified in the following table:

Device Type	Command
Unbalance-balance	:CALC{1-36}:FSIM:BAL:PAR{1-36} :SBAL
Balance-balance	:CALC{1-36}:FSIM:BAL:PAR{1-36} :BBAL
Unbalance-unbalance-balance	:CALC{1-36}:FSIM:BAL:PAR{1-36} :SSB

Differential Matching Circuit Embedding

The Differential Matching Circuit Embedding feature simulates the characteristics the DUT would exhibit if a balance-unbalance converted differential port were connected to a matching circuit after being subjected to balance-unbalance conversion.

On/Off

To turn on or off Differential Matching Circuit Embedding, use the following command:

:CALC{1-36}:FSIM:BAL:DMC:STAT

You can only turn on or off Differential Matching Circuit Embedding for all of the ports, not for each balance port individually. However, any balance port whose circuit type is set to "None" behaves as if this feature were permanently off.

Configuring the Matching Circuit Settings

To select a circuit type, use the following command:

:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}

To set the circuit constant, use the following commands:

Circuit constant	Command
C	:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:C
G	:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:G
L	:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:L
R	:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:PAR:R

When you want to use a user file to define the circuit type, specify the file using the following command:

:CALC{1-36}:FSIM:BAL:DMC:BPOR{1-2}:USER:FIL

Differential Port Impedance Conversion

The Differential Port Impedance Conversion feature converts the measurement results for a balance-unbalance converted differential port to the characteristics of a different port impedance.

On/Off

To turn on or off Differential Port Impedance Conversion, use the following command:

:CALC{1-36}:FSIM:BAL:DZC:STAT

You can only turn on or off Differential Port Impedance Conversion for all of the balance ports, not for each port individually.

Setting the Z0 Value

To set the target differential port impedance, use the following command:

:CALC{1-36}:FSIM:BAL:DZC:BPOR{1-2}:Z0

:CALC{1-36}:FSIM:BAL:DZC:BPOR{1-2}:REAL

:CALC{1-36}:FSIM:BAL:DZC:BPOR{1-2}:IMAG

Sample Program

See Fixture Simulator.

Analysis in Time Domain (time domain function)

- Overview
- Transforming Measurement Data to Time Domain
- Sample Program

Other topics about Analyzing Data

Overview

The time domain function provides the following functions:

- Transforming measurement data to data in the time domain (Transformation Function)
- Deleting unnecessary measurement data in the time domain (Gating Function)

Transforming Measurement Data to Time Domain

By using the Transformation Function, you can convert the results measured in the frequency domain to data in the time domain and analyze it.

On/Off

To turn on or off the transformation function, use the following command:

:CALC{1-36}:TRAN:TIME:STAT

Selecting Transformation Type

To select the transformation type (band pass/low pass), use the following command:

:CALC{1-36}:TRAN:TIME

To select the stimulus type (impulse/step) when the transformation type is low pass, use the following command:

:CALC{1-36}:TRAN:TIME:STIM

When the transformation type is low pass, you need to execute the following command because each measurement point must be a multiple of the start frequency.

:CALC{1-36}:TRAN:TIME:LPFR

Setting Window Shape

To set the window shape, use one of the following items.

Item	Command
β	:CALC{1-

	36}:TRAN:TIME:KBES
Impulse width	:CALC{1-36}:TRAN:TIME:IMP :WIDT
Rise time of step signal	:CALC{1-36}:TRAN:TIME:STEP :RTIM

The above three items are dependent on each other. When the value of one of them is changed, the values of the other two are automatically changed to corresponding values.

Unlike manual operation, you cannot set the window shape by selecting the window type (maximum/normal/minimum). However, you can set the same shape as each window type by setting β as follows:

	Maximum	Normal	Minimum
Value of β	13	6	0

Setting Display Range

To set the display range after time domain transformation, use the following commands:

Description	Command
Start value	:CALC{1-36}:TRAN:TIME:STAR
Stop value	:CALC{1-36}:TRAN:TIME:STOP
Center value	:CALC{1-36}:TRAN:TIME:CENT
Span value	:CALC{1-36}:TRAN:TIME:SPAN

Deleting unnecessary measurement data in the time domain

You can use the Gating Function to delete unnecessary time domain data.

On/Off

To turn on or off the gating function, use the following command:

:CALC{1-36}:FILT:TIME:STAT

Selecting Gate Type

To select the gate type, use the following command:

:CALC{1-36}:FILT:TIME

Setting Gate Shape

To select the gate shape, use the following command:

:CALC{1-36}:FILT:TIME:SHAP

Setting Gate Range

To set the gate range, use the following commands:

Description	Command
Start value	:CALC{1-36}:FILT:TIME:STAR
Stop value	:CALC{1-36}:FILT:TIME:STOP
Center value	:CALC{1-36}:FILT:TIME:CENT
Span value	:CALC{1-36}:FILT:TIME:SPAN

Sample Program

See the Time Domain.

Analyzing impedance

By turning on the parameter conversion function, you can convert the measurement result of the S parameter to the following parameters.

- Equivalent impedance in reflection measurement
- Equivalent impedance in transmission measurement
- Equivalent admittance in reflection measurement
- Equivalent admittance in transmission measurement
- Inverse of S-parameter (1/S)

To turn On/Off the parameter conversion function, use the following command:

:CALC{1-36}:CONV

To select the parameter to which you want to convert the measurement result, use the following command:

:CALC{1-36}:CONV:FUNC

Other topics about Analyzing Data

Limit Test

Limit Test

- Performing a Limit Test
- Obtaining Test Results

Performing a Limit Test

- Configuring Limit Lines
- Showing/Hiding Limit Lines
- Turning Limit Test Feature ON/OFF
- Showing/Hiding "Fail"

Other topics about Limit Test

Configuring Limit Lines

The Limit Test feature of the E5071C allows you to create up to 100 upper/lower limit lines on each trace and then determine the pass/fail status of the measured data.

When you manually configure limit lines, you must add each segment (limit line) to the limit table and define various conditions that apply to the specific segment. On the other hand, when you use an external controller to configure limit lines, you can use the following command to define all of the segment conditions (all limit lines) in the active table trace at once.

:CALC{1-36}:LIM:DATA

Alternatively, you can configure limit lines based on the data contained in a CSV file by issuing the following command:

:MMEM:LOAD:LIM

Also, you can save the contents of the current limit table to a file by issuing the following command:

:MMEM:STOR:LIM

Showing/Hiding Limit Lines

To turn ON or OFF limit lines, use the following command:

:CALC{1-36}:LIM:DISP

Even when limit lines are hidden, the system performs a limit test if the Limit Test feature is on.

Turning Limit Test Feature ON/OFF

To turn ON or OFF the Limit Test feature, use the following command:

:CALC{1-36}:LIM

Showing/Hiding "Fail"

To turn ON or OFF the "Fail" indicator that is displayed at the center of the window when the test result for the channel is "fail," use the following command:

:DISP:FSIG

Obtaining Test Results

- Overview
- Test Results at each Measurement Point
- Test Results for each Trace
- Test Results for each Channel
- Overall Test Results

Other topics about Limit Test

Overview

You can obtain test results by issuing a result retrieval command or through the status register.

Test Results at each Measurement Point

Using commands that retrieve test results

You can obtain the test results at each measurement point by retrieving the stimulus value at failed measurement points. To retrieve failed measurement points, use the following command:

Stimulus value	:CALC{1-36}:LIM:REP?
Number of measurement points	:CALC{1-36}:LIM:REP:POIN?

Using the status register

You cannot use the status register to obtain the test results at each measurement point.

Test Results for each Trace

Using commands that retrieve test results

You can retrieve the test result for each trace (i.e., the trace-wide result that combines the results for all measurement points in a particular trace) by issuing the following command:

:CALC{1-36}:LIM:FAIL?

Using the status register

The condition register and event register under the questionable limit channel {1-16} status register provide 14 bits that correspond to traces 1 to 14 and contain the test results (0: Pass, 1: Fail) for the respective traces; for example, you can obtain the test result for trace 1 from bit 1 and that for trace 14 from bit 14. The condition register and event register

under the questionable limit channel {1-16} extra status register provide two bits that correspond to traces 15 to 16 and contain the test results (0: Pass, 1: Fail) for the respective traces; for example, you can obtain the test result for trace 15 from bit 1 and that for trace 16 from bit 2.

Every bit of the condition register is set to 0 when a measurement cycle is started. Upon completion of measurement, those bits that correspond to failed traces are set to 1.

If the corresponding bit of the positive transition filter is set to 1 (preset value), each bit of the event register is set to 1 when the corresponding bit of the condition register changes from 0 to 1 (indicating that the corresponding trace failed the test).

To retrieve the registers, use the following commands:

Questionable limit channel {1-16} status register	
Condition register	:STAT:QUES:LIM:CHAN{1-16}:COND?
Event register	:STAT:QUES:LIM:CHAN{1-16}?
Questionable limit channel {1-16} extra status register	
Condition register	:STAT:QUES:LIM:CHAN{1-16}:ECH:COND?
Event register	:STAT:QUES:LIM:CHAN{1-16}:ECH?

- Although up to 36 channel is available on E5071C, the registers for limit test and Ripple/Bandwidth limit tests are for up to 16 channel.

Test Results for each Channel

Using commands that retrieve test results

No command is available that allows you to directly retrieve the test result for each channel (i.e., the channel-wide result that combines the results for all traces in a particular channel).

Using the status register

The questionable limit status event register provides 14 bits that correspond to channels 1 to 14 and contain the test results (0: Pass, 1: Fail) for the respective channels; for example, you can obtain the test result for channel 1 from bit 1 and that for channel 14 from bit 14. The questionable limit extra status event register provides nine bits that correspond to channels 1 to 2 and contain the test results (0: Pass, 1: Fail)

for the respective channels; for example, you can obtain the test result for channel 15 from bit 1 and that for channel 16 from bit 2.

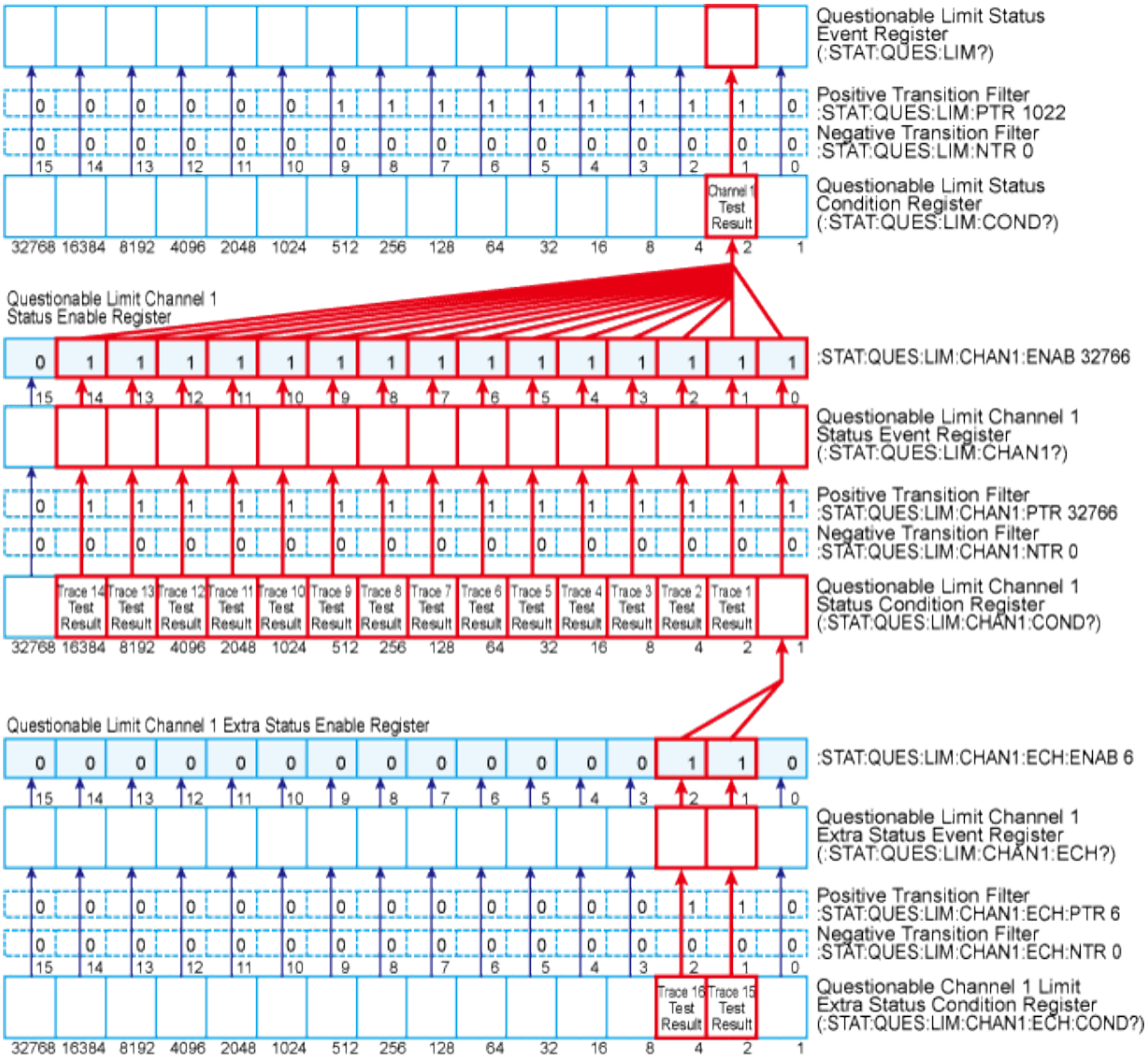
Every bit of the condition register is set to 0 after the event registers are cleared by the *CLS. Upon completion of measurement, if the channel-wide test result that combines the results for all traces in a channel is "fail," the corresponding bit of the condition register is set to 1.

If the corresponding bit of the positive transition filter is set to 1 (preset value), every bit of the event register is set to 1 when the corresponding bit of the condition register changes from 0 to 1.

To retrieve the registers, use the following commands:

Questionable limit status register	
Condition register	:STAT:QUES:LIM:COND?
Event register	:STAT:QUES:LIM?
Questionable limit extra status register	
Condition register	:STAT:QUES:LIM:ELIM:COND?
Event register	:STAT:QUES:LIM:ELIM?

Obtaining test results for a channel (channel 1 in this example) using the status register



e5071c437

Overall Test Results

Using commands that retrieve test results

No command is available that allows you to directly retrieve the overall test results that combine the test results for all channels.

Using the status register

The condition register and event register under the questionable status event register each provides bit 10, from which you can obtain the overall test result (0: Pass, 1: Fail).

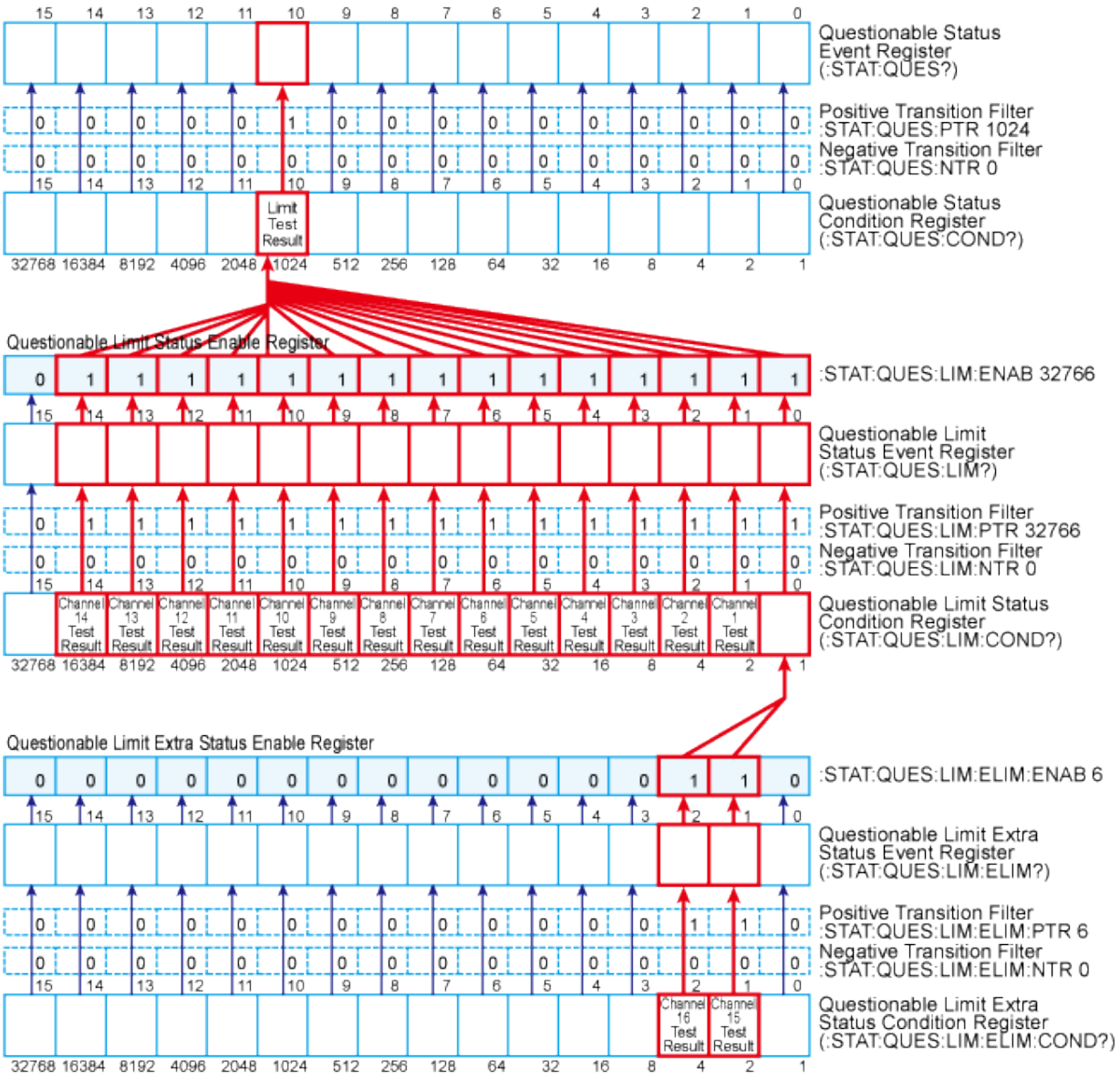
The condition register's bit 10 is set to 0 after the event registers are cleared by the *CLS. Upon completion of measurement, this bit is set to 1 if the overall test result that combines the results for all channels is "fail."

If the positive transition filter's bit 10 is set to 1 (preset value), the event register's bit 10 is set to 1 when the condition register's bit 10 changes from 0 to 1.

To retrieve the condition register and event register under the questionable status event register, use the following commands:

Condition register	:STAT:QUES:COND?
Event register	:STAT:QUES?

Obtaining overall test results using the status register


Questionable Status Event Register (:STAT:QUES?)
Positive Transition Filter :STAT:QUES:PTR 1024
Negative Transition Filter :STAT:QUES:NTR 0
Questionable Status Condition Register (:STAT:QUES:COND?)
Limit Test Result
Questionable Limit Status Enable Register
:STAT:QUES:LIM:ENAB 32766
Questionable Limit Status Event Register (:STAT:QUES:LIM?)
Positive Transition Filter :STAT:QUES:LIM:PTR 32766
Negative Transition Filter :STAT:QUES:LIM:NTR 0
Questionable Limit Status Condition Register (:STAT:QUES:LIM:COND?)
Channel 14 Test Result
Channel 13 Test Result
Channel 12 Test Result
Channel 11 Test Result
Channel 10 Test Result
Channel 9 Test Result
Channel 8 Test Result
Channel 7 Test Result
Channel 6 Test Result
Channel 5 Test Result
Channel 4 Test Result
Channel 3 Test Result
Channel 2 Test Result
Channel 1 Test Result
Questionable Limit Extra Status Enable Register
:STAT:QUES:LIM:ELIM:ENAB 6
Questionable Limit Extra Status Event Register (:STAT:QUES:LIM:ELIM?)
Positive Transition Filter :STAT:QUES:LIM:ELIM:PTR 6
Negative Transition Filter :STAT:QUES:LIM:ELIM:NTR 0
Questionable Limit Extra Status Condition Register (:STAT:QUES:LIM:ELIM:COND?)
Channel 16 Test Result
Channel 15 Test Result

e5071c438

Saving and Recalling

Saving and Recalling

- Saving and Recalling File
- Managing Files

Saving and Recalling File

- Specifying File
- Saving and Recalling Instrument Status
- Saving Measurement Data
- Saving Measurement Data in Touchstone Format
- Saving Images
- Saving and Recalling Segment Sweep Table
- Saving and Recalling Limit Table
- Saving/Recalling Power Sensor Calibration Factor Table
- Saving/Recalling Loss Compensation Table
- Saving/Loading (Importing) a VBA Program

Other topics about Saving and Recalling

Specifying File

When running a command for saving, recalling, and managing files, use a filename with extension to specify a particular file. Specify "D:" in the beginning of the file name, when specifying a file on the user area of hard disk. Also, when specifying a file name with directory, use "/" (slash) or "\" (backslash) as a delimiter.

Saving and Recalling Instrument Status

You can save the instrument state using one of the following 2 methods:

- Saving the entire instrument state into a file
- Saving the state for each channel into registers A to D (volatile memory)

Selecting content to be saved

When saving the instrument status into a file or register, the content to be saved can be selected among the following 4 options:

- Instrument status only
- Instrument status and calibration coefficient array.
- Instrument status, corrected data/memory array (measurement data)
- Instrument status, calibration coefficient array, and corrected data/memory array (measurement data)

To select a content to be saved, use the following command:

:MMEM:STOR:STYP

Selecting Content to be Saved

To select whether to save the setting of all channels/traces or that of the displayed channels/traces, use the following command:

:MMEM:STOR:SALL

Saving and recalling entire instrument status

To save the entire instrument status into a file, use the following command:

:MMEM:STOR

Recalling a file saved with the above command can reproduce the status when it was saved. To recall the settings from a file, use the following command:

:MMEM:LOAD

Auto recall

The file saved with the name autorec.sta or A:autorec.sta will be automatically recalled when the E5071C is powered ON.

Saving the state for each channel into a register

For the active channel, when you want to save the instrument state specific to that channel into only one of the registers A to D, use the following command:

:MMEM:STOR:CHAN

Recalling an instrument state saved in a register can reproduce it as the state of the active channel. To recall a register, use the following command:

:MMEM:LOAD:CHAN

- It is possible to recall a file from a different channel where it was saved.

The contents in the registers are lost when you turn OFF the power. You can delete (clear) the contents of all registers using the following command.

:MMEM:STOR:CHAN:CLE

Saving Measurement Data

Measurement data (in a formatted data array) can be saved to a file in CSV (Comma Separated Value) format.

To save measurement data in a file, use the following command:

:MMEM:STOR:FDAT

Executing the above command will save the measurement data of the active trace. Note that the data saved using the above command cannot be recalled from the E5071C.

Saving Measurement Data in Touchstone Format

Measurement data for the active channel can be saved to a file in touchstone format.

To determine a file type in touchstone file format and specify a port, use one of the following commands according to the number of ports used:

- :MMEM:STOR:SNP:TYPE:S1P
- :MMEM:STOR:SNP:TYPE:S2P
- :MMEM:STOR:SNP:TYPE:S3P
- :MMEM:STOR:SNP:TYPE:S4P

To set a data type for files saved in touchstone format, use the following command:

:MMEM:STOR:SNP:FORM

To save measurement data in touchstone format, use the following command:

:MMEM:STOR:SNP

- Only trace data of frequency sweep can be saved in touchstone format files.Trace data of frequency offset measurement and power sweep measurement cannot be saved.

Saving Images

Images displayed on the LCD screen can be saved to a file in the bitmap (.bmp) or portable network graphics (.png) format.

To save the screen image to a file, use the following command:

:MMEM:STOR:IMAG

Executing the above command will save the screen image when the command is invoked.

- This gives different screen image results from those obtained by pressing the **[Capture]** key on the front panel.

Saving and Recalling Segment Sweep Table

Segment sweep table can be saved in the file with CSV (Comma Separated Value) format.

To save segment sweep table on a file, use the following command:

:MMEM:STOR:SEGM

Executing the above command will save the segment sweep table for the active channel.

Recalling the file saved using the above command can reproduce the segment sweep table on the active channel.

To recall the settings from a file, use the following command:

:MMEM:LOAD:SEGM

- It is possible to recall a file from a different channel where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving and Recalling Limit Table

Limit table can be saved in the file with CSV (Comma Separated Value) format. To save limit table on a file, use the following command:

:MMEM:STOR:LIM

Executing the above command will save the limit table for the active trace of the active channel.

Recalling the file saved using the above command can reproduce the limit table on the active trace of the active channel. To recall the settings from a file, use the following command:

:MMEM:LOAD:LIM

- It is possible to recall a file from a different channel or trace where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Recalling Power Sensor Calibration Factor Table

The power sensor calibration table can be saved in the file with CSV (Comma Separated Value) format. To save the power sensor calibration factor table that is set specifically to the E5071C into a file, use the following commands:

- :MMEM:STOR:ASCF
- :MMEM:STOR:BSCF

Recalling a file saved with the above command can reproduce the power sensor calibration factor table when it was saved. To recall the settings from a file, use the following command:

- :MMEM:LOAD:ASCF
- :MMEM:LOAD:BSCF
    - Recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Recalling Loss Compensation Table

The loss compensation table can be saved in the file with CSV (Comma Separated Value) format.To save the loss compensation table that is set for each port into a file, use the following command:

:MMEM:STOR:PLOS{1-4}

Executing the above command will save the loss compensation table for the active channel.

Recalling the file saved using the above command can reproduce the loss compensation table on the active channel. To recall the settings from a file, use the following command:

:MMEM:LOAD:PLOS{1-4}

- It is possible to recall a file from a different channel or port where it was saved. Note that recalling operation is not guaranteed for the file that might have been modified with editor.

Saving/Loading (Importing) a VBA Program

Saving

Only the VBA project file can be saved using command.

To save the VBA project that is opened on the VBA editor on the file, use the following command.

:MMEM:STOR:PROG

Loading (importing)

To load the VBA project to the VBA editor, or to import the module/form file, use the following command.

:MMEM:LOAD:PROG

Executing above command will load/import the file according to its extension as follows:

Extension	File type
vba	VBA Project
bas	Standard module
frm	User Forms
cls	Class Modules

Sample program

See the Saving Files.

Managing Files

- Various Commands
- Sample Program

Other topics about Saving and Recalling

Various Commands

Creating directory (folder)

To create a directory (folder), use the following command:

:MMEM:MDIR

Deleting file (directory)

To delete a file or a directory, use the following command:

:MMEM:DEL

Copying file

To copy a file, use the following command:

:MMEM:COPY

Transferring files

File transfer from the external controller to the E5071C can be possible by reading data from a file on the controller and then writing them to the file on the E5071C.

:MMEM:TRAN

Also, file transfer from the E5071C to the external controller can be possible by reading data from a file on the E5071C using the commands as query and then writing them to the file on the controller.

Retrieving data from storage

To retrieve information for the storage that is built in the E5071C (usage, property of file located in a specified directory), use the following command;

:MMEM:CAT?

Sample Program

See the Transferring Files.

Communication with External Devices using I/O Port

Communication with External Devices (Handler I/O)

- Handler I/O Port Overview
- I/O Signal Pin Layout and Description
- Inputting/Outputting Data
- Preset states at power-on
- Timing Chart
- Electrical Characteristics

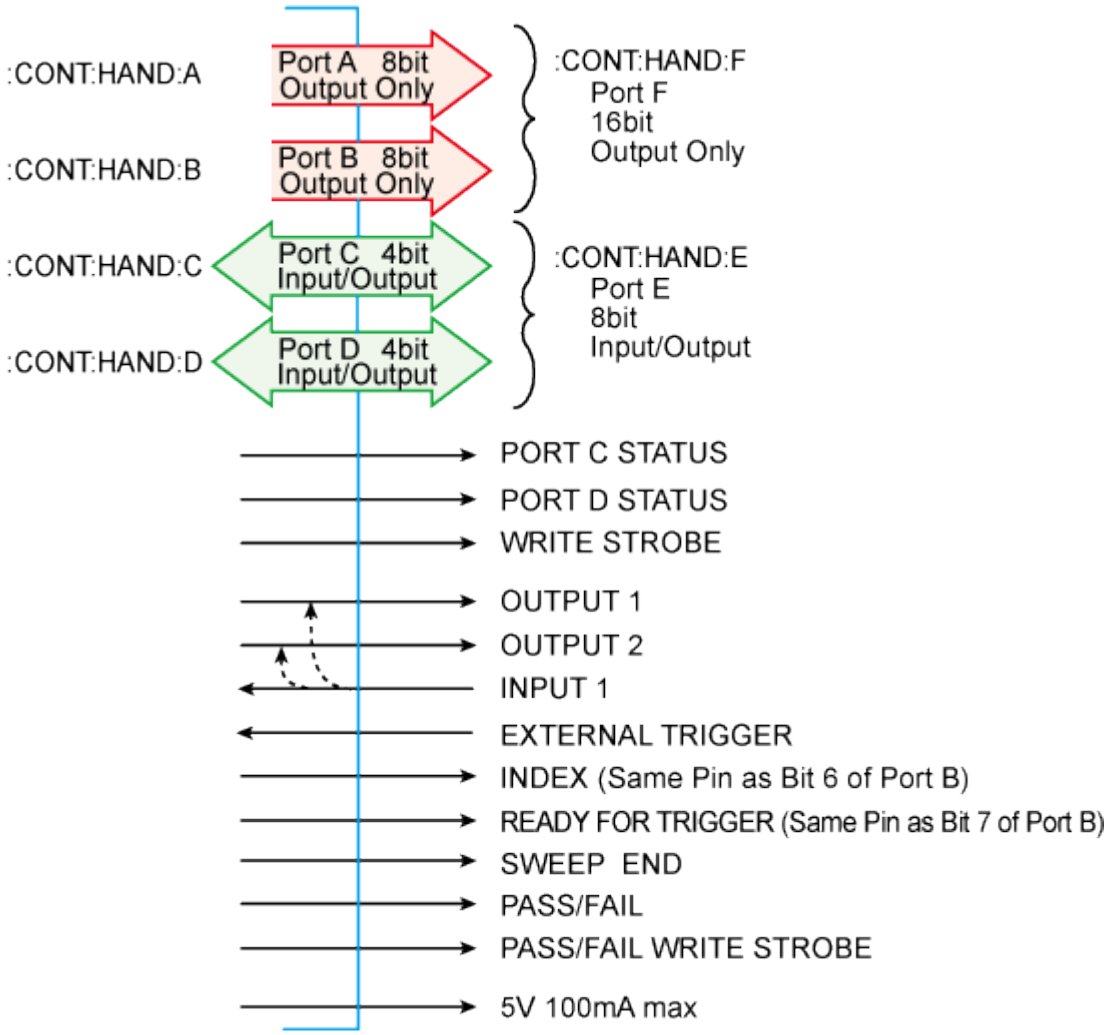
Handler I/O Port Overview

The E5071C handler I/O port provides four independent parallel ports for data I/O associated with several control signal lines and the power line. All signals operate in TTL logic.

The data I/O ports are configured with 2 pairs of 8 bit output port and 2 pairs of 4 bit bi-directional port. Also those ports can cooperate to provide a maximum 16-bit-width output port or a maximum 8-bit-width input port.

The I/O signals operate on the negative logic basis, which can be altered. The control signal lines consist of various control output data, including completion of measurement or control signal for handshaking.

I/O ports and control signal lines

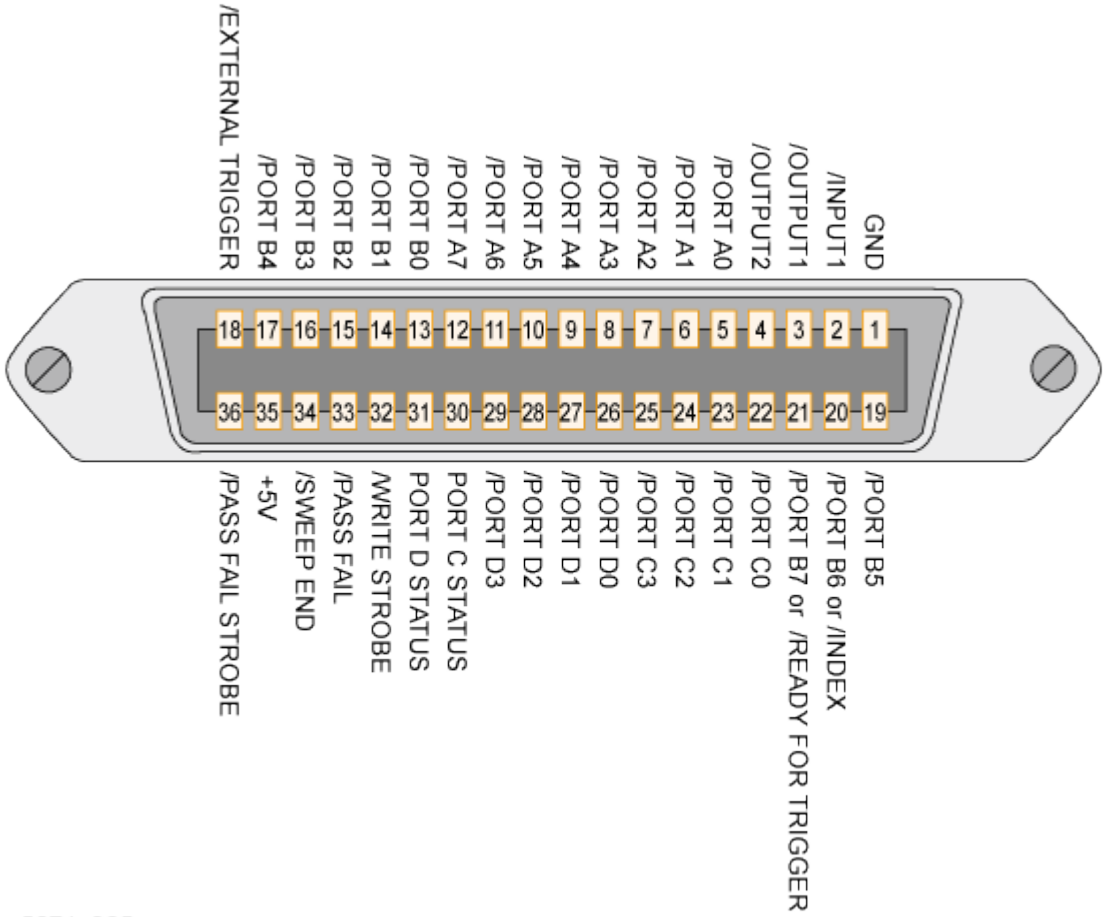


e5071c364

Other topics about Communication with External Devices

I/O Signal Pin Layout and Description

The layout of the I/O signal pins on the handler interface connector and its description are shown below.



e5071c365

A slash (/) symbol preceding signal names means that they are negative logic (active low).

Pin number	Signal name	Input/Output	Description
1	GND	N/A	Ground.
2	/INPUT1	Input	When this port receives a negative pulse, /OUTPUT1 and /OUTPUT2 are changed to the Low level.
3	/OUTPUT1	Output	Changes to the Low level when /INPUT1 receives a

			negative pulse. A command can be available for altering the Low/High level logic.
4	/OUTPUT2	Output	Changes to the Low level when /INPUT1 receives a negative pulse.A command can be available for altering the Low/High level logic.
5	/PORT A0	Output	Bit 0 of port A (8 bit parallel output port)
6	/PORT A1	Output	Bit 1 of port A.
7	/PORT A2	Output	Bit 2 of port A.
8	/PORT A3	Output	Bit 3 of port A.
9	/PORT A4	Output	Bit 4 of port A.
10	/PORT A5	Output	Bit 5 of port A.
11	/PORT A6	Output	Bit 6 of port A.
12	/PORT A7	Output	Bit 7 of port A.
13	/PORT B0	Output	Bit 0 of port B (8 bit parallel output port)
14	/PORT B1	Output	Bit 1 of port B.
15	/PORT B2	Output	Bit 2 of port B.
16	/PORT B3	Output	Bit 3 of port B.
17	/PORT B4	Output	Bit 4 of port B.
18	/EXTERNAL TRIGGER	Input	An external trigger signal. When the trigger source is set to the "External," this port generates a trigger in respond to the trailing edge of a negative pulse.

19	/PORT B5	Output	Bit 5 of port B.
20	/PORT B6	Output	Bit 6 of port B.
	/INDEX		Indicates that analog measurement is complete. The /INDEX signal changes to the Low level when analog measurement (all sweeps of all channels) is complete. When the handler receives the signal, it assumes that it is ready to connect the next DUT. However, no measurement data are available until data calculation is completed. When the point trigger function is on, it goes to the High level before staring measurement of the first measurement point and returns to the Low level after completing measurement of all measurement points.
21	/PORT B7	Output	Bit 7 of port B.
	/READY FOR TRIGGER		Indicates that the instrument is ready for triggering.This signal is changed to the Low level when the instrument is ready for receiving a trigger signal. With the point trigger function on, it goes to the Low level when the instrument is ready to accept the trigger signal for the first point and goes

			to the High level when the trigger signal for the first point is received. When measurement of all measurement points is completed and the instrument is ready to receive the trigger signal for the first point of the next sweep, this signal goes to the Low level again.
22	/PORT C0	Input/Output	Bit 0 of port C (4 bit parallel I/O port)
23	/PORT C1	Input/Output	Bit 1 of port C.
24	/PORT C2	Input/Output	Bit 2 of port C.
25	/PORT C3	Input/Output	Bit 3 of port C.
26	/PORT D0	Input/Output	Bit 0 of port D (4 bit parallel I/O port)
27	/PORT D1	Input/Output	Bit 1 of port D.
28	/PORT D2	Input/Output	Bit 2 of port D.
29	/PORT D3	Input/Output	Bit 3 of port D.
30	PORT C STATUS	Output	Port C status signal.This signal is changed to the High level when the port C is configured to output port. It is changed to the Low level when the port is configured to input port.
31	PORT D STATUS	Output	Port D status signal. This signal is changed to the High level when the port D is configured to output port. It is changed to the Low level when the port is

			configured to input port.
32	/WRITE STROBE	Output	A output port write strobe signal. When data is present (that is, output level changes) on any of the output ports, this signal provides a negative pulse.
33	/PASS FAIL	Output	Each limit test's results signal. This signal changes to the High level when limit test, bandwidth test, or ripple test results return FAIL. It changes to the Low level when all limit test results return PASS.
34	/SWEEP END	Output	A sweep completion signal. When measurement (all sweeps of all channels) and data calculation are completed, this signal provides a negative pulse.
35	+5V	Output	Provides +5V DC power supply for external instruments.
36	/PASS FAIL STROBE	Output	Each limit test's results write a strobe signal. When limit test result is present on /PASS FAIL, this signal provides a negative pulse.

Other topics about Communication with External Devices

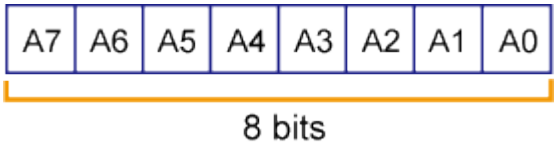
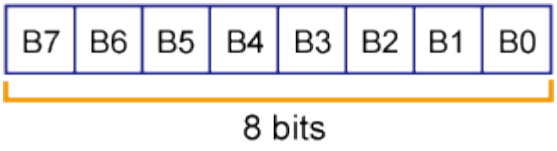
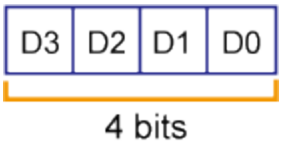
Inputting/Outputting Data

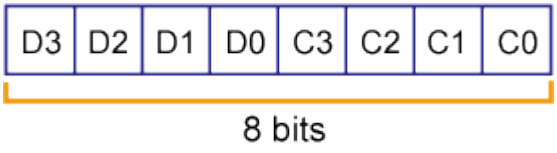
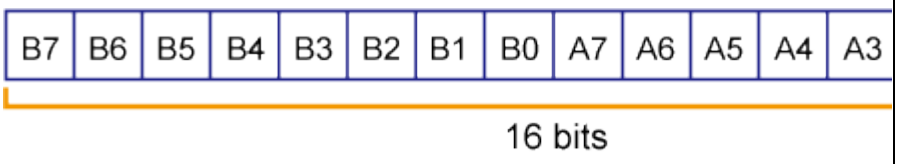
- Overview
- Specifying Signal Direction of Port
- Reading Data Input from Port
- Data Output to Port
- Sample Program

Other topics about Communication with External Devices

Overview

The E5071C handler I/O port provides the ports for data I/O shown below.

Port Name	Usage	Data Structure
Port A	Output	
Port B	Output	
Port C	Input/Output	
Port D	Input/Output	

Port E	Input/Output	
Port F	Output	

Specifying Signal Direction of Port

Signal direction (input/output) can be changed for the ports C, D, and D as shown in I/O ports and control signal lines. Thus, before the ports are used, the directions should be determined according to their usage.

To specify the signal direction for the ports C and D, use the following command. Direction for the port E depends on the setting for the ports C and D.

Port Name	Command
Port C	:CONT:HAND:C:MODE
Port D	:CONT:HAND:D:MODE

Reading Data Input into Port

When the ports C, D, or E is configured to input ports, binary data represented with High(0)/Low(1) of each bit of the port will be read as decimal data.

To retrieve the data, use the following command as query:

Port Name	Command
Port C	:CONT:HAND:C
Port D	:CONT:HAND:D
Port E	:CONT:HAND:E

Data Output to Port

To the ports A through F (the ports C, D, and E should be configured to output ports), binary data (decimal data when output data is specified with

a command) represented with High(0)/Low(1) of each bit of the port can be output.

To output data, use the following command:

Port Name	Command
Port A	:CONT:HAND:A
Port B	:CONT:HAND:B
Port C	:CONT:HAND:C
Port D	:CONT:HAND:D
Port E	:CONT:HAND:E
Port F	:CONT:HAND:F

- The bit 6 of the data output by :CONT:HAND:B (the bit 14 of the data output by :CONT:HAND:F) is ignored when outputting the /INDEX signal is turned on.
- The bit 7 of the data output by :CONT:HAND:B (the bit 15 of the data output by :CONT:HAND:F command) is ignored when outputting the /READY FOR TRIGGER signal is turned on.

Sample Program

See Handler Interface.

Preset states at power-on

The handler I/O port is set at power-on as follows (not affected at reset)

Description	Status
Port A	High (All Bits)
Port B	High (All Bits)
Port C	Input
Port D	Input
Port C STATUS	Low
Port D STATUS	Low
/OUTPUT1	High
/OUTPUT2	High
/SWEEP END	High
/PASS FAIL	High

Other topics about Communication with External Devices

Timing Chart

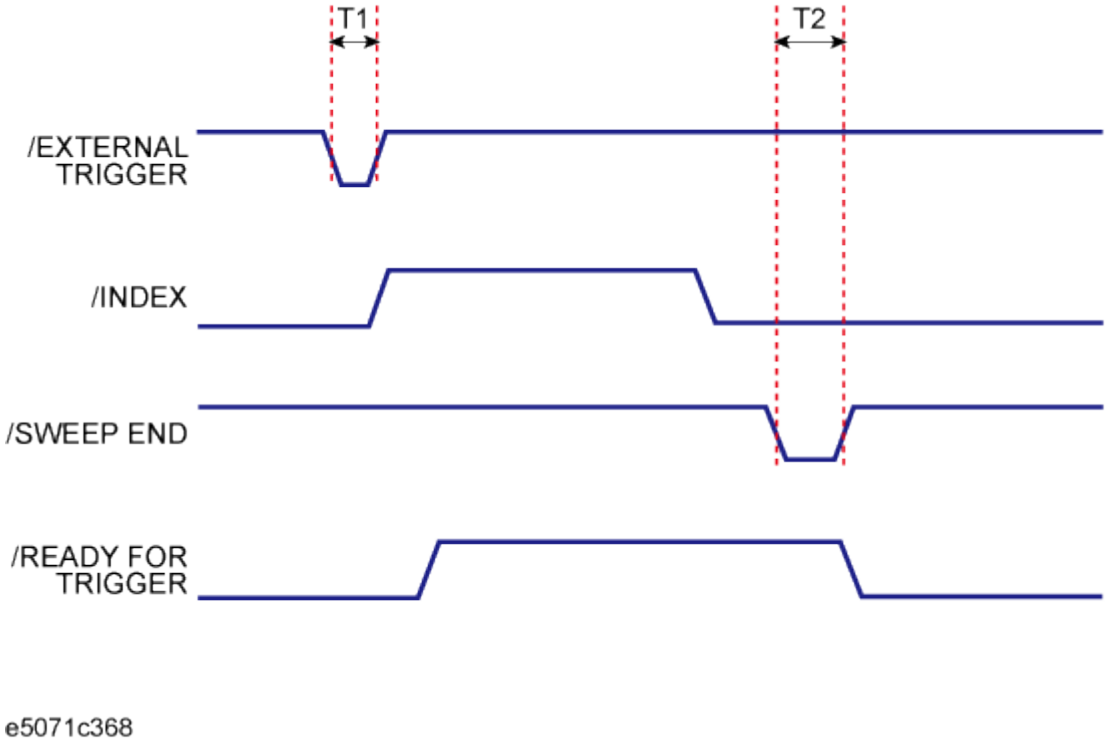
- Overview
- Timing Chart of I/O Port Signal
- Timing Chart of Data Output and Write Strobe Signal
- Timing Chart of Limit Test Result Output and Write Strobe Signal
- Timing Chart of /INPUT1 and /OUTPUT1, /OUTPUT2

Other topics about Communication with External Devices

Overview

This section shows the typical timing chart of I/O port Signal.

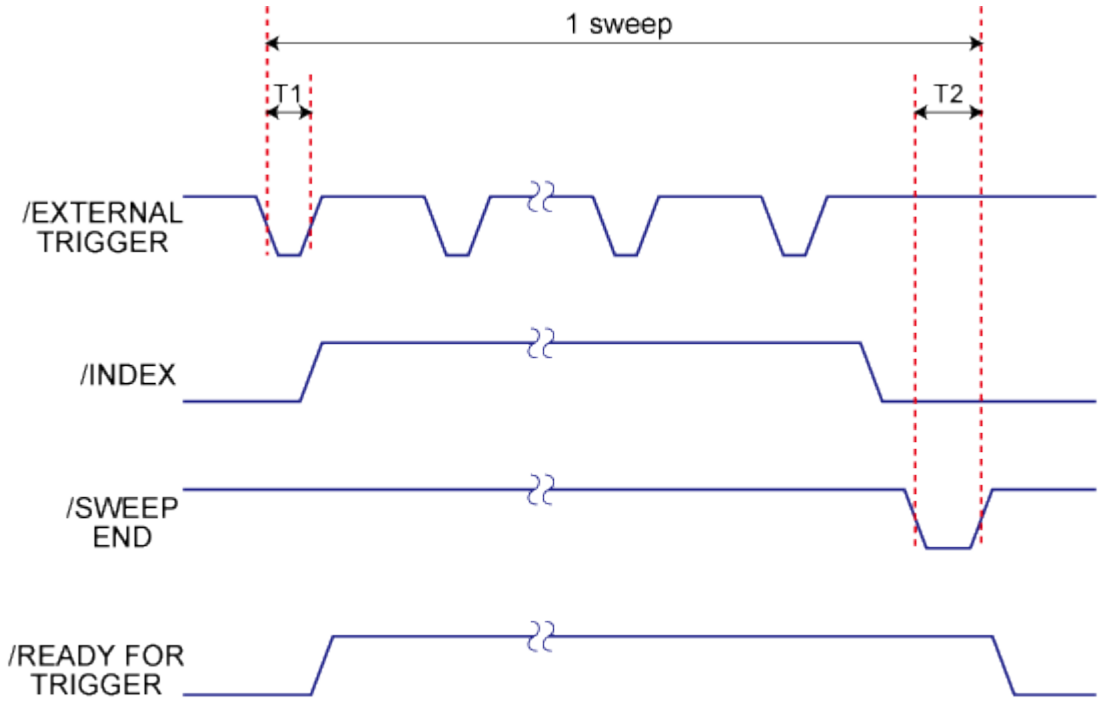
Timing chart of I/O Port Signal (Point trigger function: off)



e5071c368

		Minimum value	Typical Value	Maximum value
T1	Pulse width of /EXTERNAL TRIGGER	1 µs	-	-
T2	Pulse width of /SWEEP END	10 µs	12 µs	-

Timing Chart of I/O Port Signal (Point trigger function:on, Low-latency external trigger mode: on)



e5071c309

When both the point trigger function and the low-latency external trigger mode are on, the /EXTERNAL TRIGGER signal must be inputted for each measurement point during a single sweep. The /INDEX signal goes to the High level before staring measurement of the first measurement point and returns to the Low level after the completing measurement of all measurement points.

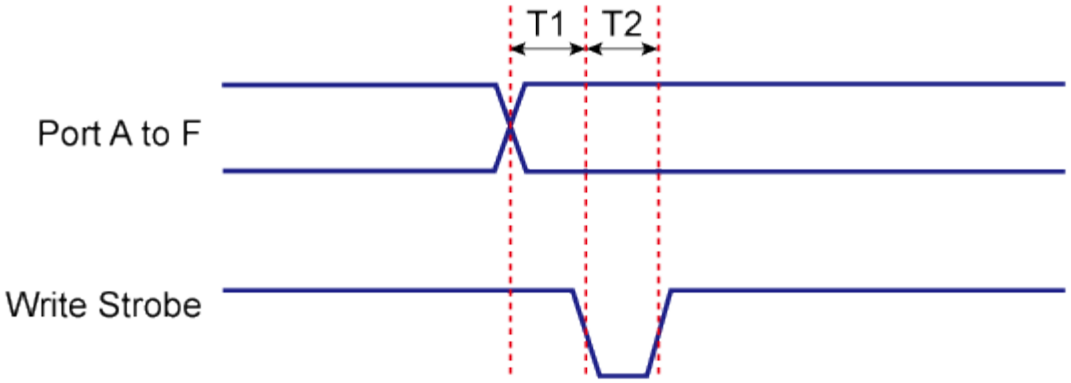
The /READY FOR TRIGGER signal goes to the Low level when the instrument is ready to accept the trigger signal for the first point and then goes to the High level when the trigger signal for the first point is received.

When measurement of all measurement points is completed and the instrument is ready to receive the trigger signal for the first point of the next sweep, this signal goes to the Low level again.

When the point trigger function is on and the low-latency external trigger mode is off, the /READY FOR TRIGGER signal goes High each time a trigger signal is received and goes Low when measurement of each measurement point is completed and the instrument is ready to accept a trigger for the next measurement point.

The times of T1 and T2 are the same as those when the point trigger function is off. For more information, see Timing chart of I/O Port Signal(Point trigger function:off).

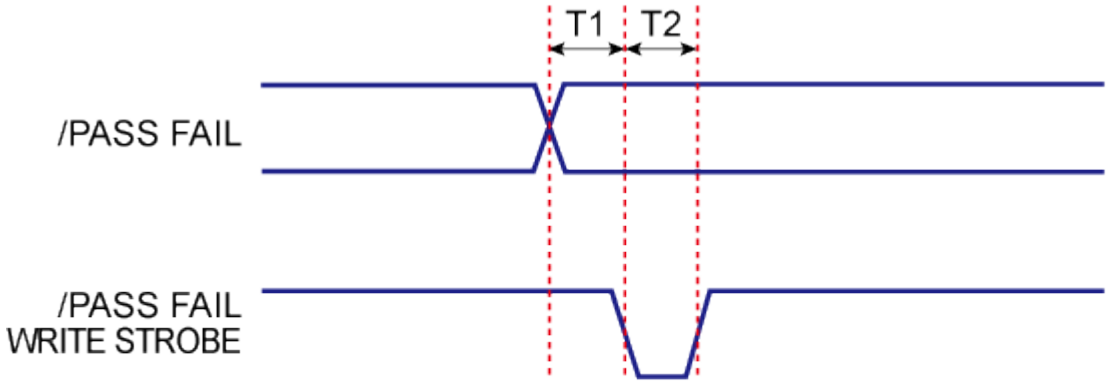
Timing Chart of Data Output and Write Strobe Signal



e5071c366

T1	Response time of write strobe signal	1 µs
T2	Pulse width of write strobe signal	1 µs

Timing Chart of Limit Test Result Output and Write Strobe Signal

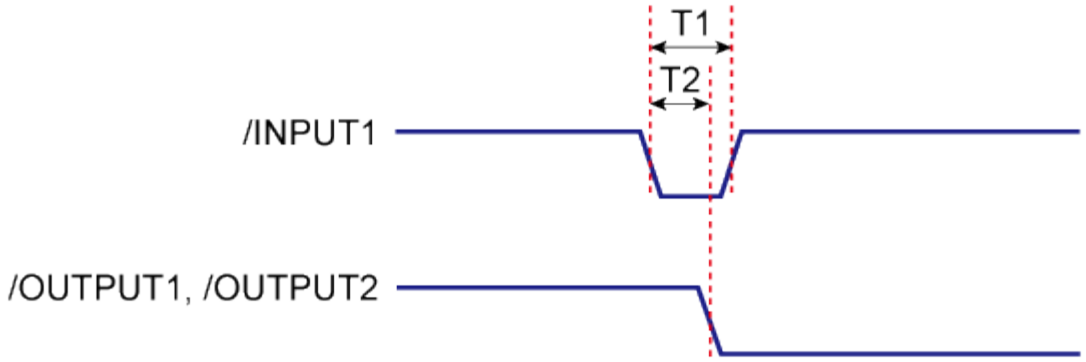


e5071c367

T1	Response time of /PASS FAIL write strobe	1 µs
T2	Pulse width of /PASS FAIL write strobe	1 µs

- When the average trigger function is activated, the fail and write strobe signals are output at the time that the average test result shows "failed" on a certain channel.

Timing Chart of /INPUT1 and /OUTPUT1, /OUTPUT2



e5071c369

		Minimum value	Maximum value
T1	Pulse width of /INPUT1	1 µs	-
T2	Response time of /OUTPUT1, /OUTPUT2	0.4 µs	0.6 µs

Electrical Characteristics

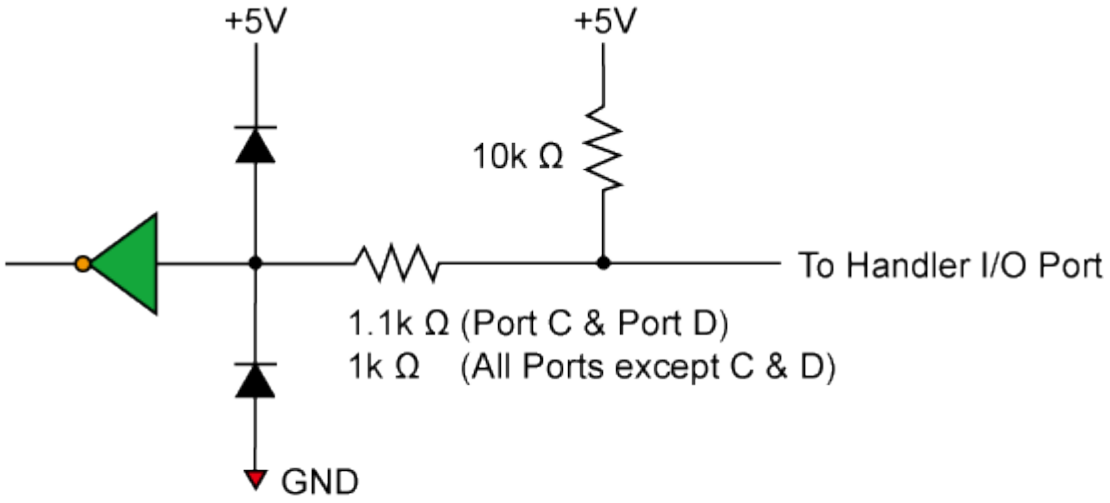
- Input Signal
- Output Signal
- Power Supply (+5 V)

Other topics about Communication with External Devices

Input Signal

All input signals are TTL compatible.

Maximum rate input voltage		-0.5 V to 5.5 V
Input voltage	**High level**	2.0 V to 5.0 V
	Low level	0 V to 0.5 V



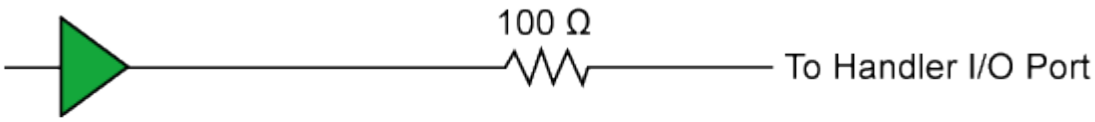
e5071c370

Output Signal

All output signals are TTL compatible.

Maximum rate output current		-10 mA to 10 mA
Output current	**High level**	-5 mA

	Low level	3 mA
Output voltage	**High level**	2.0 V to 3.3 V (when output current is from -5 mA to 0 mA) 3.20 V (when output current is -1 mA) 2.75 V (when output current is -5 mA)
	Low level	0 V to 0.8 V (when output current is from 0 mA to 3 mA) 0.25 V (when output current is 1 mA) 0.55 V (when output current is 3 mA)



e5071c371

Power Supply (+5 V)

The following table shows electrical characteristics of +5 V power supply for external instruments.

Output voltage	4.5 V to 5.5 V
Maximum output current	100 mA

Status Reporting System

Status Reporting System

- General Status Register Model
- Using the Status Reporting System
- Status Register Structure

General Status Register Model

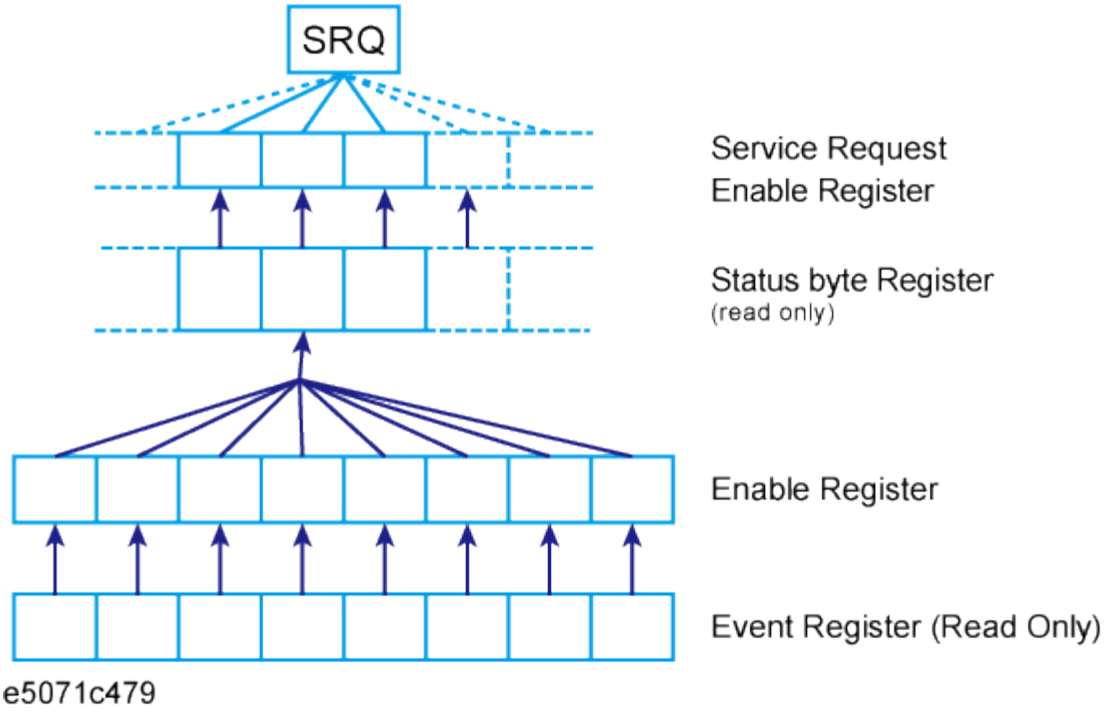
- Overview
- Event Register
- Enable Register
- Status Byte Register
- Condition Register and Transition Filter

Other topics about Status Reporting System

Overview

The Agilent E5071C has a status reporting system to report the condition of the instrument.

General status register model



e5071c479

The status reporting system has a hierarchical structure as shown in the figure above. When the instrument satisfies a particular condition, the corresponding bit of the event register is set to 1. Therefore, you can check the instrument status by reading the event register.

When the event register bit is set to "1" and a corresponding enable register bit (a bit marked with an arrow in General status register model) is also "1," the summary bit of the status byte register is set to "1." You can read the status byte register by using the serial poll.

If the bit of the service request enable register is "1," a service request (SRQ) is generated by the positive transition of the corresponding status byte register bit. By generating SRQ, you can notify the controller that the E5071C is requesting service. In other words, interruption by SRQ can be programmed. For more information on using SRQ, see Using the status register or Using the status reporting system.

Event Register

Reflects the corresponding condition of the E5071C (e.g., occurrence of an event) as a bit status. These bits continuously monitor changes in the E5071C's state and change the bit status when the condition (e.g., change bit status to "1" if a specific event occurs) for each bit is met. You cannot change the bit status by issuing a SCPI command.

Enable Register

Setting the enable register allows you to specify event register bits that can set "1" to the summary bit of the status byte register when an event occurs. The register bits work as mask bits; setting "1" to an enable register will enable a corresponding bit in the event register.

For example, when you want to set "1" as the summary bit in the status byte register by a specific register condition, set the corresponding enable register to "1."

Status Byte Register

If the enabled event register is set to "1," a corresponding bit of the status byte register is also set to "1." This register also indicates the output queue and SRQ status.

The value of the status byte register can be read by using the *STB? command or serial poll (SPOLL statement in HTBasic) from the controller.

Reading the status byte register by using the *STB? command does not affect the contents of the status byte register. However, reading it with the SPOLL statement of HTBasic will clear the RQS bit in the status byte register.

Also, setting the service request enable register using the *SRE command can generate a service request synchronously with the status byte register.

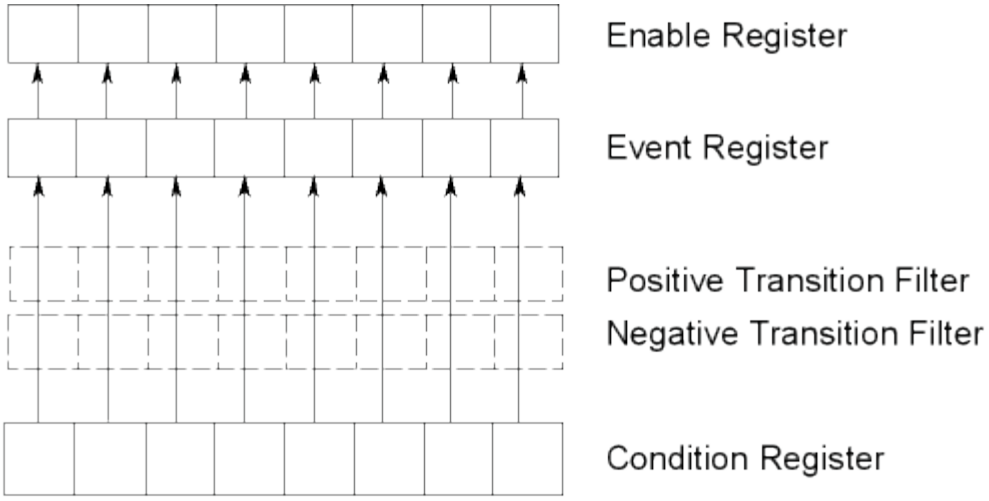
Condition Register and Transition Filter

When the status register has a transition filter, there is a lower register called a condition register under the event register. The transition filter is between the event register and the condition register.

The transition filter enables you to select a positive and/or negative transition of the condition register bit in order to set a bit in the corresponding event register. For example, using the negative transition filter to set bit 3 to "1" causes bit 3 of the event register to be set to "1"

when bit 3 of the condition register makes a negative transition, that is, changes from 1 to 0.

Transition filter and condition register



4294ape022

In the E5071C, the following registers provide a condition register and transition filter:

- Operation status register
- Questionable status register
- Questionable limit status register
- Questionable limit extra status register
- Questionable limit channel {1-16} status register
- Questionable limit channel {1-16} extra status register
- Questionable bandwidth limit status register
- Questionable bandwidth limit extra status register
- Questionable bandwidth limit channel {1-16} status register
- Questionable bandwidth limit channel {1-16} extra status register
- Questionable ripple limit status register
- Questionable ripple limit extra status register
- Questionable ripple limit channel {1-16} status register
- Questionable ripple limit channel {1-16} extra status register

Using the Status Reporting System

You can manage the status report system using the following commands in any combination:

- *CLS
- *SRE
- *STB?
- *ESE
- *ESR?
- :STAT:PRES
- :STAT:OPER:ENAB
- :STAT:OPER:COND?
- :STAT:OPER?
- :STAT:OPER:PTR
- :STAT:OPER:NTR
- :STAT:QUES:ENAB
- :STAT:QUES:COND?
- :STAT:QUES?
- :STAT:QUES:PTR
- :STAT:QUES:NTR
- :STAT:QUES:LIM:ENAB
- :STAT:QUES:LIM:COND?
- :STAT:QUES:LIM?
- :STAT:QUES:LIM:PTR
- :STAT:QUES:LIM:NTR
- :STAT:QUES:LIM:ELIM:ENAB
- :STAT:QUES:LIM:ELIM:COND?
- :STAT:QUES:LIM:ELIM?
- :STAT:QUES:LIM:ELIM:PTR
- :STAT:QUES:LIM:ELIM:NTR
- :STAT:QUES:LIM:CHAN{1-16}:ENAB
- :STAT:QUES:LIM:CHAN{1-16}:COND?
- :STAT:QUES:LIM:CHAN{1-16}?
- :STAT:QUES:LIM:CHAN{1-16}:PTR
- :STAT:QUES:LIM:CHAN{1-16}:NTR

- :STAT:QUES:LIM:CHAN{1-16}:ECH:ENAB
- :STAT:QUES:LIM:CHAN{1-16}:ECH:COND?
- :STAT:QUES:LIM:CHAN{1-16}:ECH?
- :STAT:QUES:LIM:CHAN{1-16}:ECH:PTR
- :STAT:QUES:LIM:CHAN{1-16}:ECH:NTR
- :STAT:QUES:BLIM:ENAB
- :STAT:QUES:BLIM:COND?
- :STAT:QUES:BLIM?
- :STAT:QUES:BLIM:PTR
- :STAT:QUES:BLIM:NTR
- :STAT:QUES:BLIM:ELIM:ENAB
- :STAT:QUES:BLIM:ELIM:COND?
- :STAT:QUES:BLIM:ELIM?
- :STAT:QUES:BLIM:ELIM:PTR
- :STAT:QUES:BLIM:ELIM:NTR
- :STAT:QUES:BLIM:CHAN{1-16}:ENAB
- :STAT:QUES:BLIM:CHAN{1-16}:COND?
- :STAT:QUES:BLIM:CHAN{1-16}?
- :STAT:QUES:BLIM:CHAN{1-16}:PTR
- :STAT:QUES:BLIM:CHAN{1-16}:NTR
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:ENAB
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:COND?
- :STAT:QUES:BLIM:CHAN{1-16}:ECH?
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:PTR
- :STAT:QUES:BLIM:CHAN{1-16}:ECH:NTR
- :STAT:QUES:RLIM:ENAB
- :STAT:QUES:RLIM:COND?
- :STAT:QUES:RLIM?
- :STAT:QUES:RLIM:PTR
- :STAT:QUES:RLIM:NTR
- :STAT:QUES:RLIM:ELIM:ENAB
- :STAT:QUES:RLIM:ELIM:COND?
- :STAT:QUES:RLIM:ELIM?
- :STAT:QUES:RLIM:ELIM:PTR

- :STAT:QUES:RLIM:ELIM:NTR
- :STAT:QUES:RLIM:CHAN{1-16}:ENAB
- :STAT:QUES:RLIM:CHAN{1-16}:COND?
- :STAT:QUES:RLIM:CHAN{1-16}?
- :STAT:QUES:RLIM:CHAN{1-16}:PTR
- :STAT:QUES:RLIM:CHAN{1-16}:NTR
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:ENAB
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:COND?
- :STAT:QUES:RLIM:CHAN{1-16}:ECH?
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:PTR
- :STAT:QUES:RLIM:CHAN{1-16}:ECH:NTR

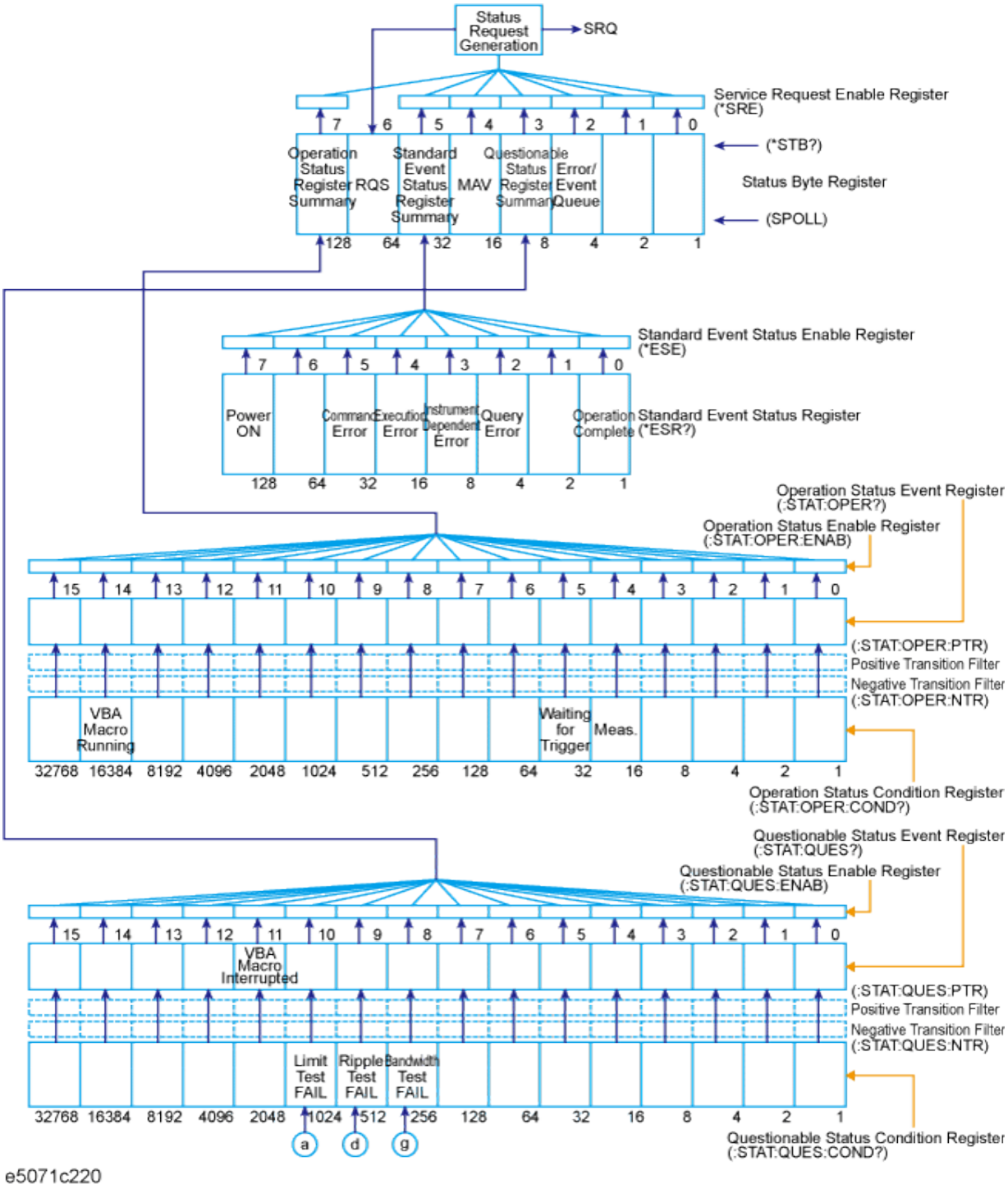
Other topics about Status Reporting System

Status Register Structure

Status Register Structure

- Status Register
- Status Register for Limit Test (Channel)
- Status Register for Limit Test (Trace) (1 of 2)
- Status Register for Limit Test (Trace) (2 of 2)
- Status Register for Bandwidth Limit (Channel)
- Status Register for Bandwidth Limit (Trace) (1 of 2)
- Status Register for Bandwidth Limit (Trace) (2 of 2)
- Status Register for Ripple Limit (Channel)
- Status Register for Ripple Limit (Trace) (1 of 2)
- Status Register for Ripple Limit (Trace) (2 of 2)
    - Although up to 36 channel is available on E5071C, the registers for limit test and Ripple/Bandwidth limit tests are for up to 16 channel.

Status Register



e5071c220

Status Bit Definitions of Status Byte Register

Bit Position	Name	Description

0, 1	Not used	Always 0
2	Error/Event Queue	Set to "1" if the error/event queue contains data; reset to "0" when all the data has been retrieved.
3	Questionable Status Register Summary	Set to "1" when one of the enabled bits in the questionable status register is set to "1."
4	MAV (Message Available)	Set to "1" when the output queue contains data; reset to "0" when all the data has been retrieved.
5	Standard Event Status Register Summary	Set to "1" when one of the enabled bits in the standard event status register is set to "1."
6	RQS	Set to "1" when any of the status byte register bits enabled by the service request enable register is set to "1"; reset to "0" when all the data has been retrieved through serial polling.
7	Operation Status Register Summary	Set to "1" when one of the enabled bits in the operational status register is set to "1."

Issuing the *CLS command will clear all bits from the status byte register.

Status Bit Definitions of Standard Event Status Register

Bit Position	Name	Description
0	Operation Complete	Set to "1" upon completion of all operations done by commands that precede the *OPC? command.
1	Not used	Always 0

2	Query Error	1. Set to "1" when the E5071C receives a data output request but there is no data to output. 2. Set to "1" when the data of the E5071C's output queue has been cleared because of a new message received before the completion of data output.
3	Instrument Dependent Error	Set to "1" when an error has occurred and the error is not a command, query, or execution error.
4	Execution Error	1. Set to "1" when any parameter in an SCPI command exceeds its input range or is inconsistent with the E5071C's capabilities. 2. Set to "1" when an SCPI command cannot be properly executed due to some condition of the E5071C.
5	Command Error	1. Set to "1" when an IEEE 488.2 syntax error occurs (a command sent to the E5071C does not follow the IEEE 488.2 syntax). Possible violations include the command parameter violating the E5071C listening formats or being unacceptable. 2. Set to "1" when a semantic error occurs. Possible causes include a command containing misspellings being sent to the E5071C or an IEEE 488.2 command not supported by the E5071C being sent. 3. Set to "1" when GET (Group Execution Trigger) is input while a program message is being received.
6	Not used	Always 0

7	Power ON	Set to "1" when the E5071C is powered ON, or when the firmware is restarted.

Issuing the *CLS command will clear all bits from the standard event status register.

Status Bit Definitions of the Operation Status Condition Register

Bit Position	Name	Description
0 - 3	Not used	Always 0
4	Measurement	Set to "1" during measurement
5	Waiting for Trigger	Set to "1" while the instrument is waiting for a trigger.
6 - 13	Not used	Always 0
14	VBA Macro Running	Set to "1" while a VBA macro is running.
15	Not used	Always 0

Issuing the *CLS command will clear all bits from the operation status event register.

Status Bit Definitions of the Questionable Status Condition Register

Bit Position	Name	Description
0 - 7	Not used	Always 0
8	Bandwidth Test Fail (Questionable bandwidth limit status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit status event register is set to "1."
9	Ripple Test Fail (Questionable ripple limit status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit status

		event register is set to "1."
10	Limit Test Fail (Questionable limit status register summary)	Set to "1" while one of the enabled bits in the questionable limit status event register is set to "1."
11 - 15	Not used	Always 0

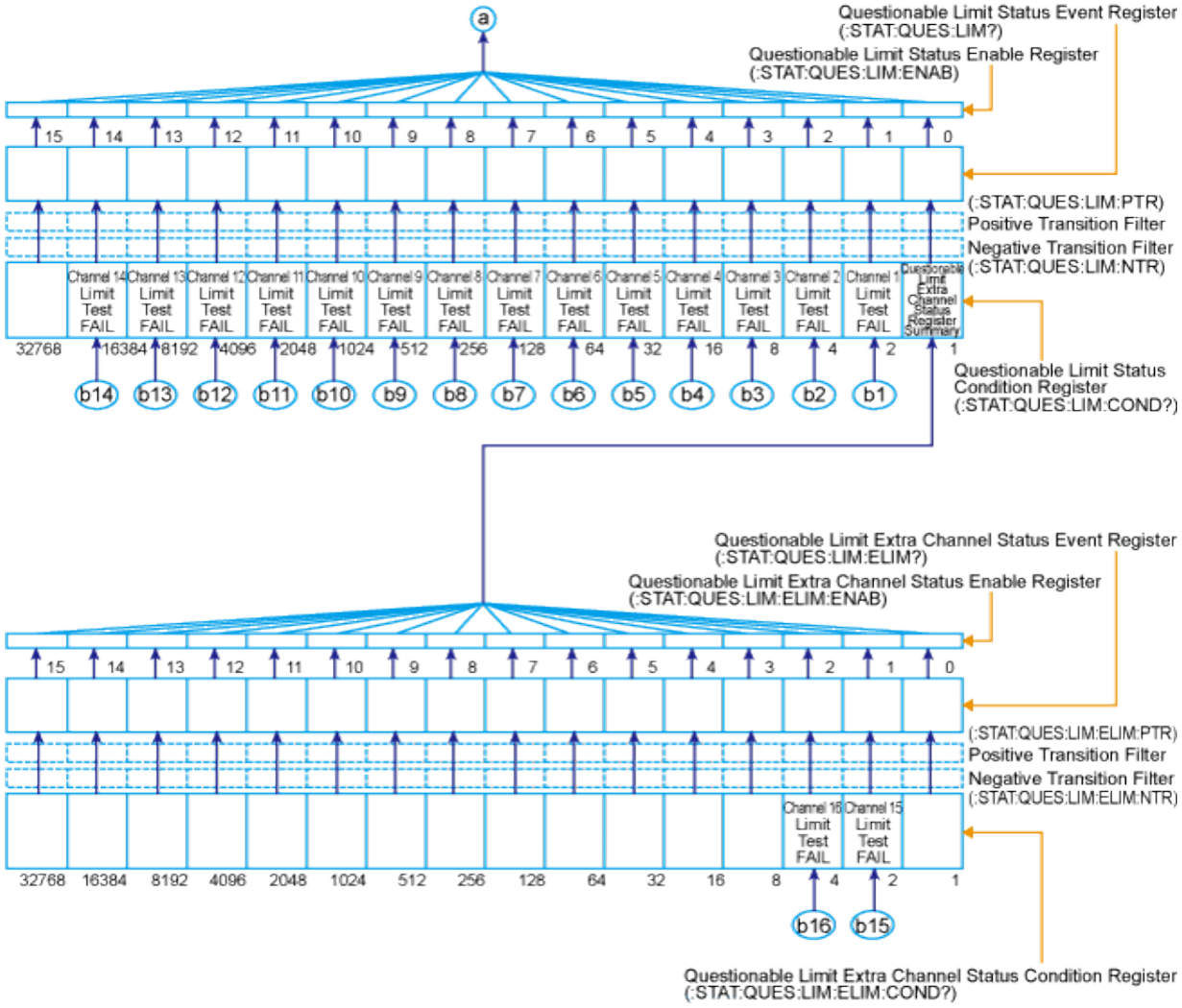
Status Bit Definitions of the Questionable Status Event Register

Bit Position	Name	Description
0 - 7	Not used	Always 0
8	Bandwidth Test Fail (Questionable bandwidth limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
9	Ripple Test Fail (Questionable ripple limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
10	Limit Test Fail (Questionable limit status register summary)	Set to "1" when a transition of the condition register occurs if the transition filters are set as valid values.
11	VBA Macro Interrupted	Set to "1" when a VBA macro is interrupted by one of the following reasons. Occurrence of an execution error Executing "End" statement in the VBA

		Macro Executing :PROG:STAT STOP Operating **CTRL** + **Break** using the keyboard Operating **Marcro Break** or **Macro Setup** > **Stop** using the front panel
12 - 15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable status event register.

Status Register for Limit Test (channel)



e5071c221

Status Bit Definitions of the Questionable Limit Status Condition Register

Bit Position	Name	Description
0	Channel 15, 16 Limit Test summary (questionable limit extra status register summary)	Set to "1" while one of the enabled bits in the questionable limit extra status event register is set to "1."

1	Channel 1 Limit Test Fail (questionable limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 1 status event register is set to "1."
2	Channel 2 Limit Test Fail (questionable limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 2 status event register is set to "1."
3	Channel 3 Limit Test Fail (questionable limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 3 status event register is set to "1."
4	Channel 4 Limit Test Fail (questionable limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 4 status event register is set to "1."
5	Channel 5 Limit Test Fail (questionable limit channel 5 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 5 status event register is set to "1."
6	Channel 6 Limit Test Fail (questionable limit channel 6 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 6 status event register is set to "1."
7	Channel 7 Limit Test Fail (questionable limit channel 7 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 7 status event register is set to "1."
8	Channel 8 Limit Test Fail (questionable limit channel 8 status register	Set to "1" while one of the enabled bits in the questionable limit channel 8

	summary)	status event register is set to "1."
9	Channel 9 Limit Test Fail (questionable limit channel 9 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 9 status event register is set to "1."
10	Channel 10 Limit Test Fail (questionable limit channel 10 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 10 status event register is set to "1."
11	Channel 11 Limit Test Fail (questionable limit channel 11 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 11 status event register is set to "1."
12	Channel 12 Limit Test Fail (questionable limit channel 12 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 12 status event register is set to "1."
13	Channel 13 Limit Test Fail (questionable limit channel 13 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 13 status event register is set to "1."
14	Channel 14 Limit Test Fail (questionable limit channel 14 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 14 status event register is set to "1."
15	Not used	Always 0

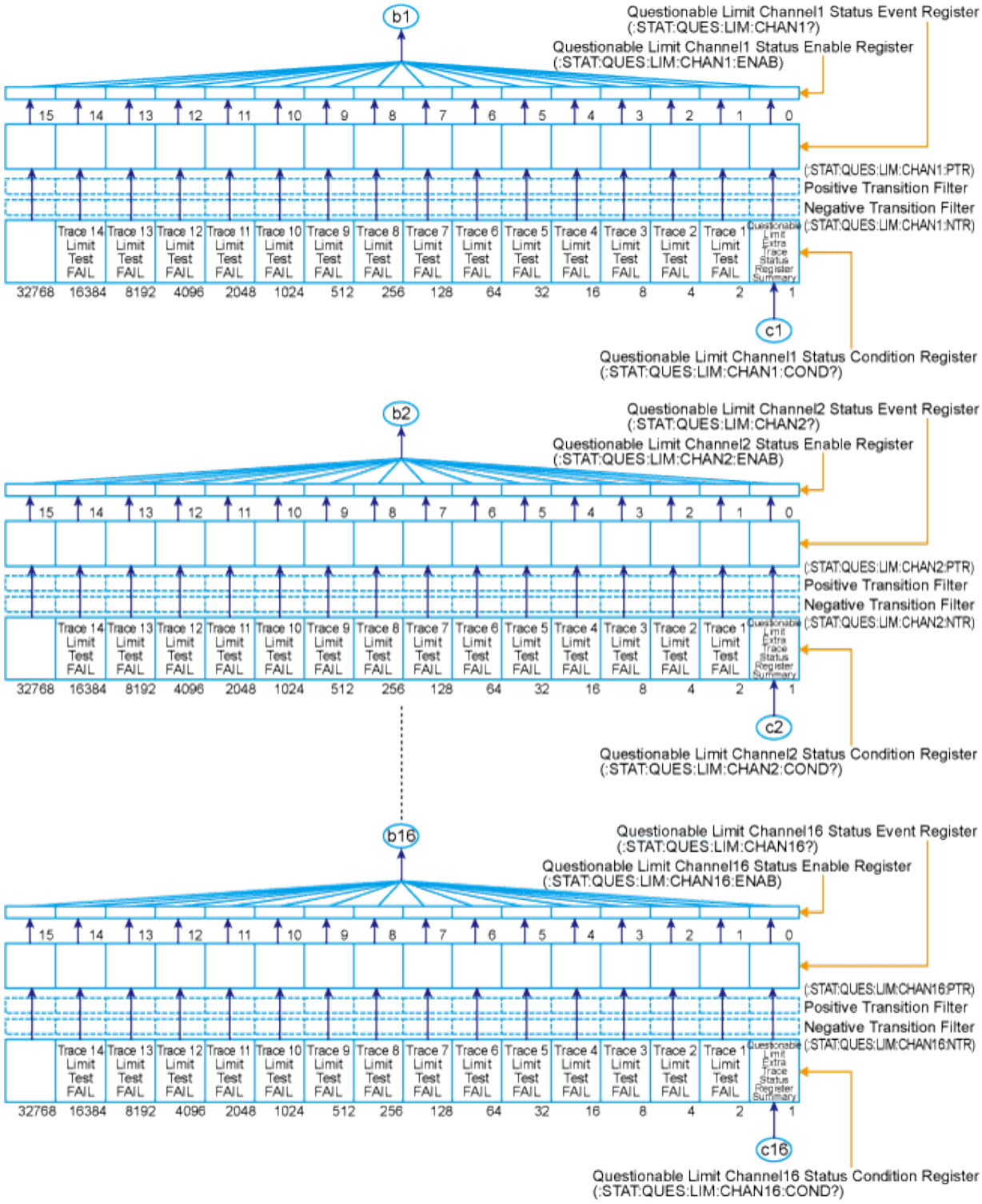
Issuing the *CLS command will clear all bits from the questionable limit status event register.

Status Bit Definitions of the Questionable Limit Extra Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 15 Limit Test Fail (questionable limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 15 status event register is set to "1."
2	Channel 16 Limit Test Fail (questionable limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 16 status event register is set to "1."
3 - 15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable limit extra status event register.

Status Register for Limit Test (Trace) (1 of 2)



Status Bit Definitions of the Questionable Limit Channel {1-16} Status Condition Register

Bit Position	Name	Description
0	Trace 15, 16 Limit Test summary (questionable limit channel {1-16} extra status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel {1-16} extra status event register is set to "1."
1	Trace 1 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 1.
2	Trace 2 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 2.
3	Trace 3 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 3.
4	Trace 4 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 4.
5	Trace 5 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 5.

6	Trace 6 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 6.
7	Trace 7 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 7.
8	Trace 8 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 8.
9	Trace 9 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 9.
10	Trace 10 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 10.
11	Trace 11 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 11.
12	Trace 12 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the

		measurement cycle finishes and returns "fail" as the limit test result for trace 12.
13	Trace 13 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 13.
14	Trace 14 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 14.
15	Not used	Always 0

Issuing the *CLS command will clear all the bits in the questionable limit channel {1-16} status event register.

Status Bit Definitions of the Questionable Limit Channel {1-16} Extra Status Condition Register

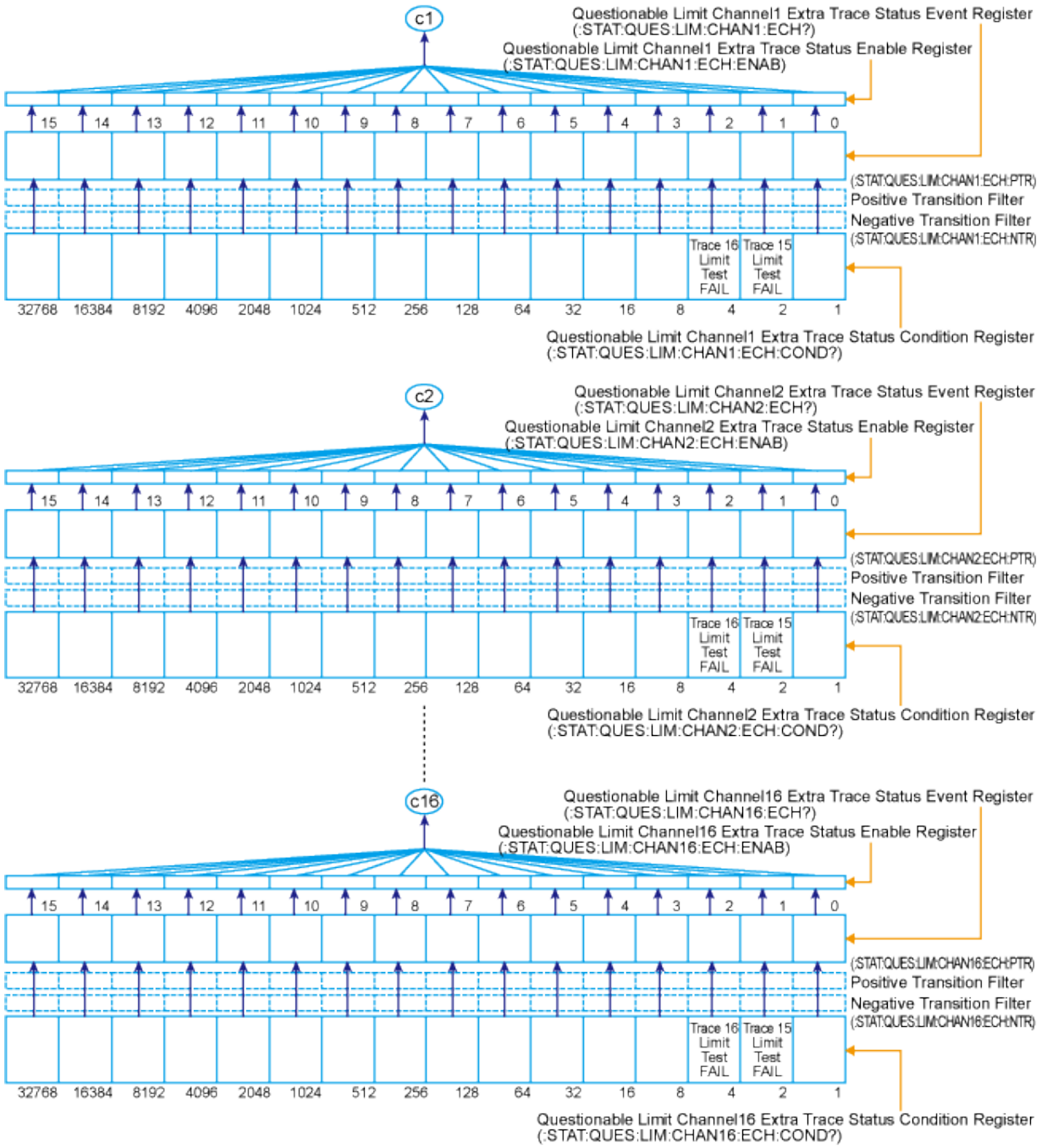
Bit Position	Name	Description
0	Not used	Always 0
1	Trace 15 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 15.
2	Trace 16 Limit Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the limit test result for trace 16.

3 - 15	Not used	Always 0

Issuing the *CLS command will clear all the bits in the questionable limit channel {1-16} extra status event register.

- Although up to 36 channel is available on E5071C, the register for limit test is for up to 16 channel.

Status Register for Limit Test (Trace) (2 of 2)



e5071c223

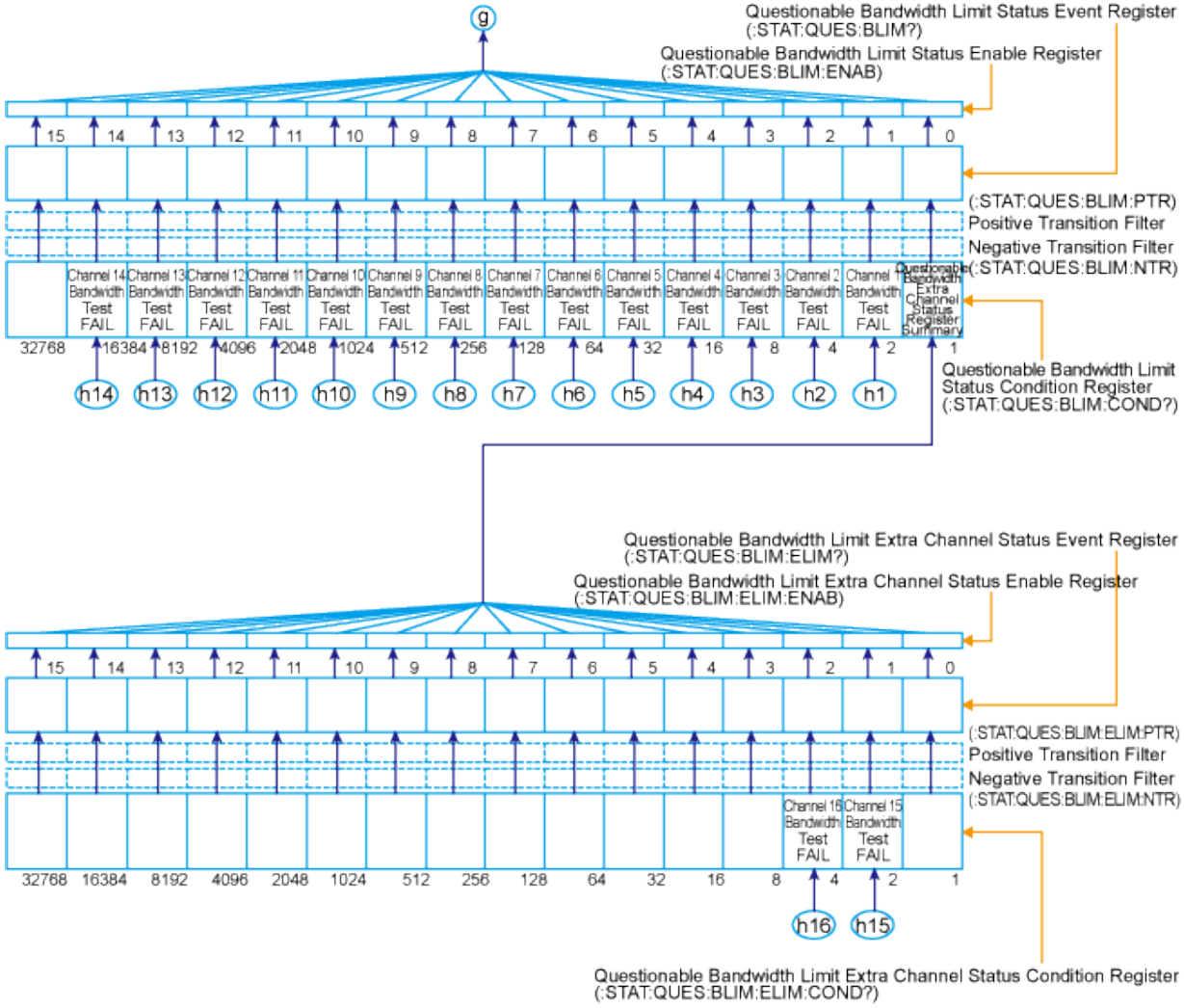
Status Bit Definitions of the Questionable Limit Extra Status Condition Register

Bit Position	Name	Description

0	Not used	Always 0
1	Channel 15 Limit Test Fail (questionable limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 15 status event register is set to "1."
2	Channel 16 Limit Test Fail (questionable limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable limit channel 16 status event register is set to "1."
3 - 15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable limit extra status event register.

Status Register for Bandwidth Limit (Channel)



e5071c226

Status Bit Definitions of the Questionable Bandwidth Limit Channel {1-16} Status Condition Register

Bit Position	Name	Description
0	Trace 15, 16 Bandwidth Test summary (questionable bandwidth limit channel {1-16} extra status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel {1-16} extra status event register is set to "1."
1	Trace 1 Bandwidth Test	Set to "0" when a

	Fail	measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 1.
2	Trace 2 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 2.
3	Trace 3 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 3.
4	Trace 4 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 4.
5	Trace 5 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 5.
6	Trace 6 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for

		trace 6.
7	Trace 7 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 7.
8	Trace 8 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 8.
9	Trace 9 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 9.
10	Trace 10 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 10.
11	Trace 11 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 11.
12	Trace 12 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the

		measurement cycle finishes and returns "fail" as the bandwidth test result for trace 12.
13	Trace 13 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 13.
14	Trace 14 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 14.
15	Not used	Always 0

Issuing the *CLS command will clear all the bits in the questionable bandwidth limit channel {1-16} status event register.

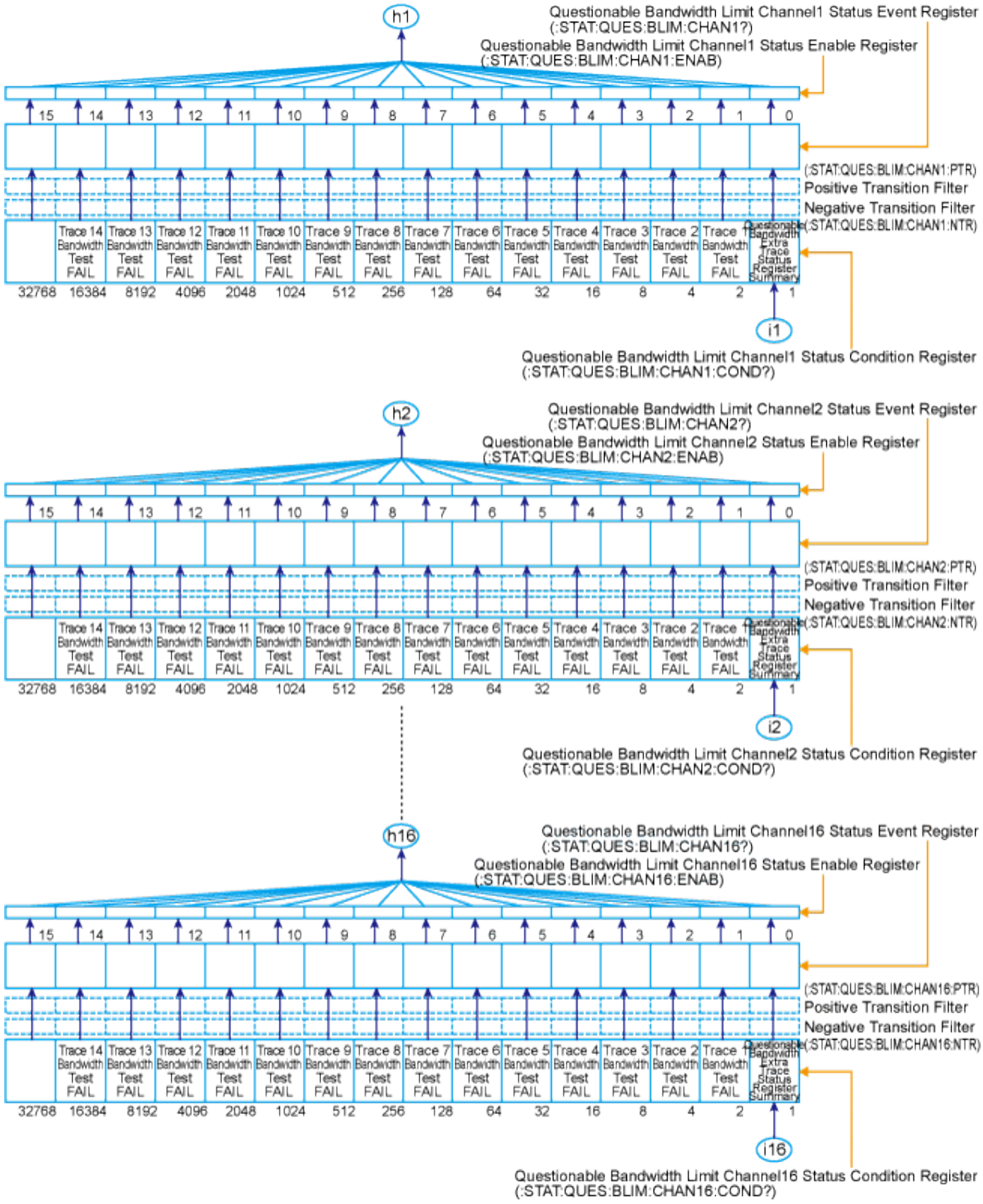
Status Bit Definitions of the Questionable Bandwidth Limit Channel {1-16} Extra Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Trace 15 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 15.
2	Trace 16 Bandwidth Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the bandwidth test result for trace 16.

3 - 15	Not used	Always 0

Issuing the *CLS command will clear all the bits in the questionable bandwidth limit channel {1-16} extra status event register.

Status Register for Bandwidth Limit (Trace) 1



Status Bit Definitions of the Questionable Bandwidth Limit Status Condition Register

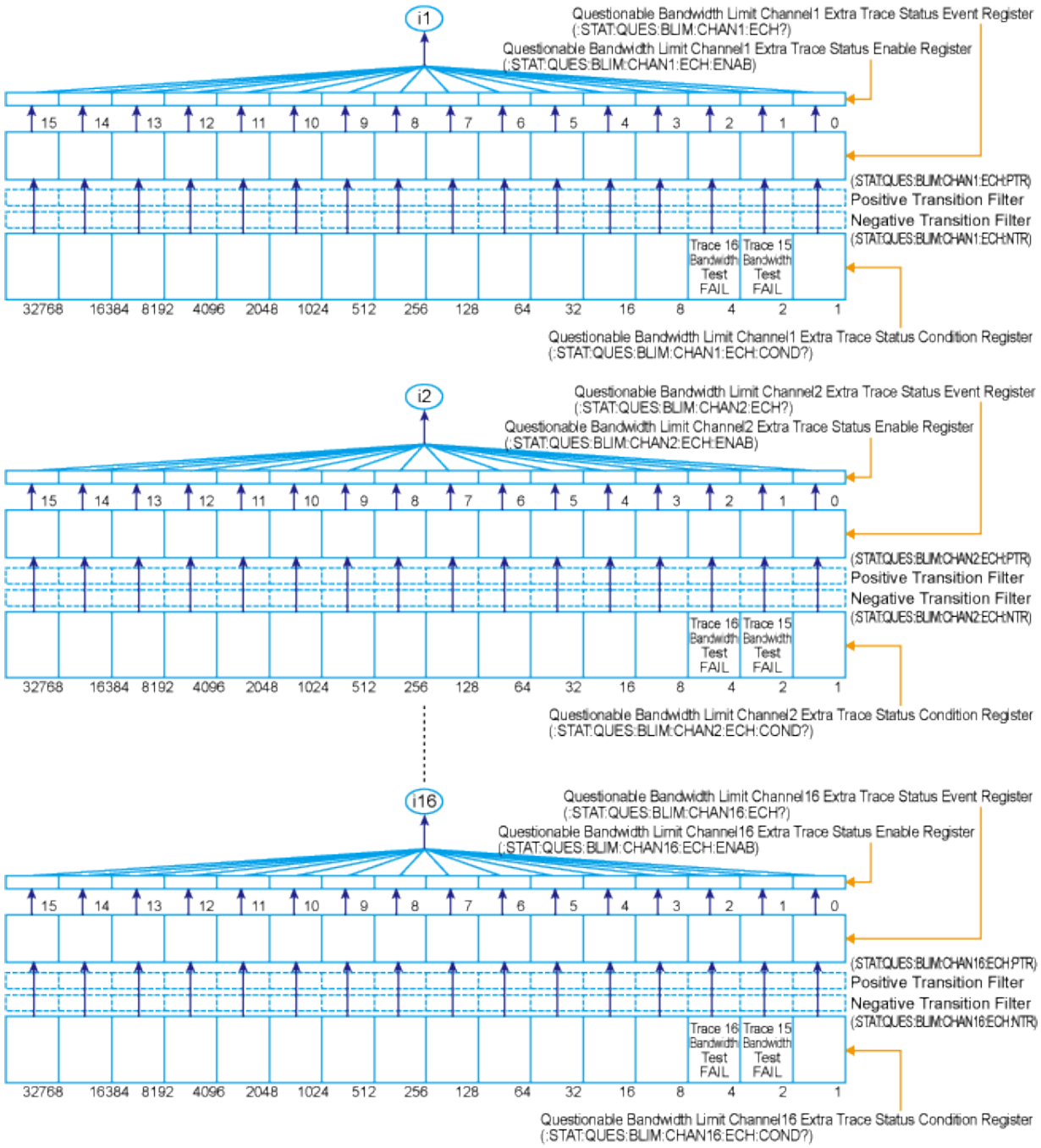
Bit Position	Name	Description
0	Channel 15, 16 Bandwidth test summary (questionable bandwidth limit extra status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit extra status event register is set to "1."
1	Channel 1 Bandwidth Test Fail (questionable bandwidth limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 1 status event register is set to "1."
2	Channel 2 Bandwidth Test Fail (questionable bandwidth limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 2 status event register is set to "1."
3	Channel 3 Bandwidth Test Fail (questionable bandwidth limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 3 status event register is set to "1."
4	Channel 4 Bandwidth Test Fail (questionable bandwidth limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 4 status event register is set to "1."
5	Channel 5 Bandwidth Test Fail (questionable bandwidth limit channel 5 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 5 status event register is set to "1."
6	Channel 6 Bandwidth Test Fail (questionable bandwidth limit channel 6 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 6 status event register is set to "1."

7	Channel 7 Bandwidth Test Fail (questionable bandwidth limit channel 7 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 7 status event register is set to "1."
8	Channel 8 Bandwidth Test Fail (questionable bandwidth limit channel 8 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 8 status event register is set to "1."
9	Channel 9 Bandwidth Test Fail (questionable bandwidth limit channel 9 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 9 status event register is set to "1."
10	Channel 10 Bandwidth Test Fail (questionable bandwidth limit channel 10 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 10 status event register is set to "1."
11	Channel 11 Bandwidth Test Fail (questionable bandwidth limit channel 11 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 11 status event register is set to "1."
12	Channel 12 Bandwidth Test Fail (questionable bandwidth limit channel 12 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 12 status event register is set to "1."
13	Channel 13 Bandwidth Test Fail (questionable bandwidth limit channel 13 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 13 status event register is set to "1."
14	Channel 14 Bandwidth Test Fail (questionable bandwidth limit channel 14	Set to "1" while one of the enabled bits in the questionable bandwidth limit

	status register summary)	channel 14 status event register is set to "1."
15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable bandwidth limit status event register.

Status Register for Bandwidth Limit (Trace) 2



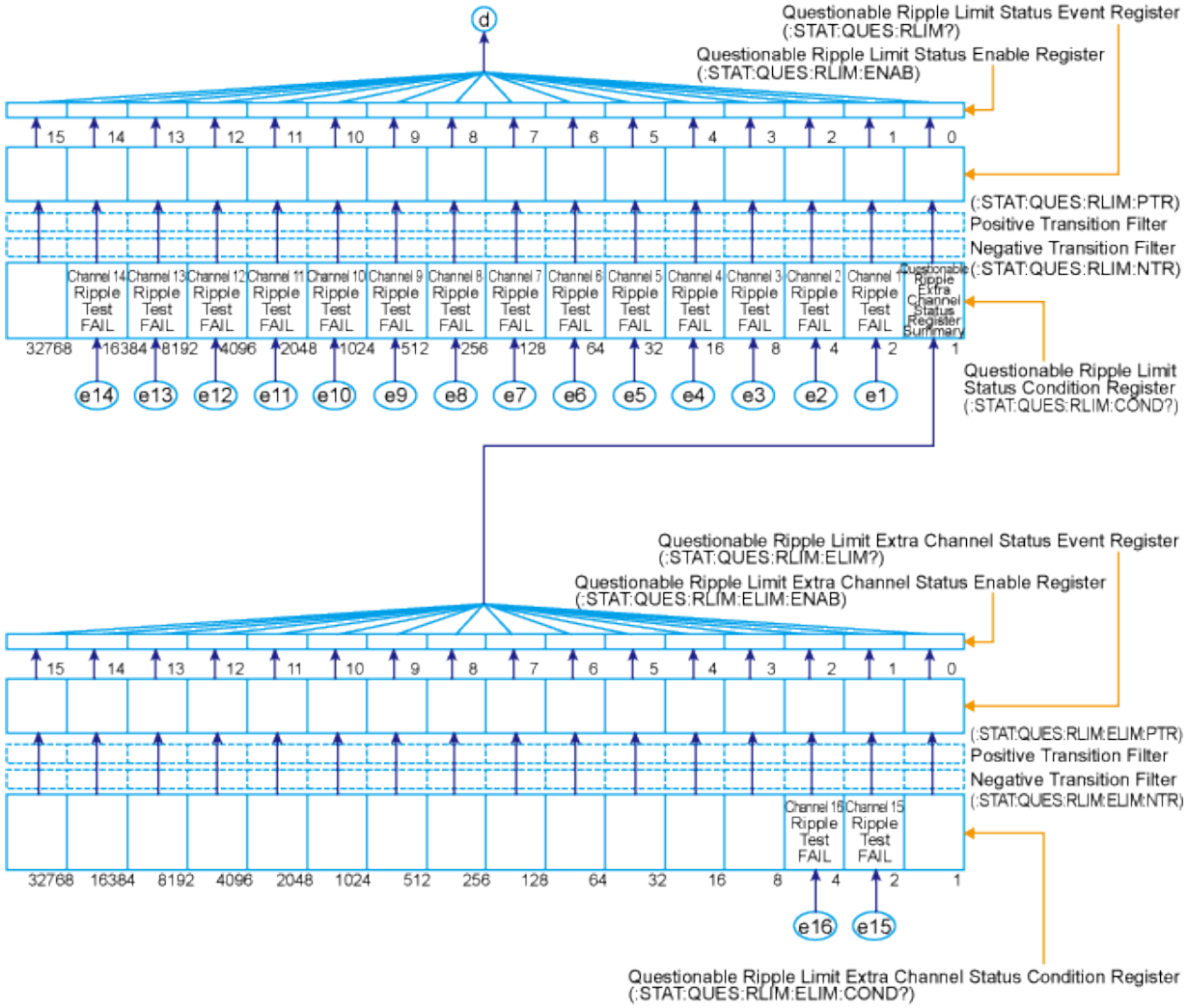
Status Bit Definitions of the Questionable Bandwidth Limit Extra Status Condition Register

Bit Position	Name	Description

0	Not used	Always 0
1	Channel 15 Bandwidth Test Fail (questionable bandwidth limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 15 status event register is set to "1."
2	Channel 16 Bandwidth Test Fail (questionable bandwidth limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable bandwidth limit channel 16 status event register is set to "1."
3 - 15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable bandwidth limit extra status event register.

Status Register for Ripple Limit (Channel)



e5071c224

Status Bit Definitions of the Questionable Ripple Limit Status Condition Register

Bit Position	Name	Description
0	Channel 15, 16 Ripple Test summary (questionable ripple limit extra status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit extra status event register is set to "1."

1	Channel 1 Ripple Test Fail (questionable ripple limit channel 1 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 1 status event register is set to "1."
2	Channel 2 Ripple Test Fail (questionable ripple limit channel 2 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 2 status event register is set to "1."
3	Channel 3 Ripple Test Fail (questionable ripple limit channel 3 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 3 status event register is set to "1."
4	Channel 4 Ripple Test Fail (questionable ripple limit channel 4 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 4 status event register is set to "1."
5	Channel 5 Ripple Test Fail (questionable ripple limit channel 5 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 5 status event register is set to "1."
6	Channel 6 Ripple Test Fail (questionable ripple limit channel 6 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 6 status event register is set to "1."
7	Channel 7 Ripple Test Fail (questionable ripple limit channel 7 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 7 status event register is set to "1."
8	Channel 8 Ripple Test Fail (questionable ripple limit channel 8 status register	Set to "1" while one of the enabled bits in the questionable ripple limit

	summary)	channel 8 status event register is set to "1."
9	Channel 9 Ripple Test Fail (questionable ripple limit channel 9 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 9 status event register is set to "1."
10	Channel 10 Ripple Test Fail (questionable ripple limit channel 10 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 10 status event register is set to "1."
11	Channel 11 Ripple Test Fail (questionable ripple limit channel 11 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 11 status event register is set to "1."
12	Channel 12 Ripple Test Fail (questionable ripple limit channel 12 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 12 status event register is set to "1."
13	Channel 13 Ripple Test Fail (questionable ripple limit channel 13 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 13 status event register is set to "1."
14	Channel 14 Ripple Test Fail (questionable ripple limit channel 14 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 14 status event register is set to "1."
15	Not used	Always 0

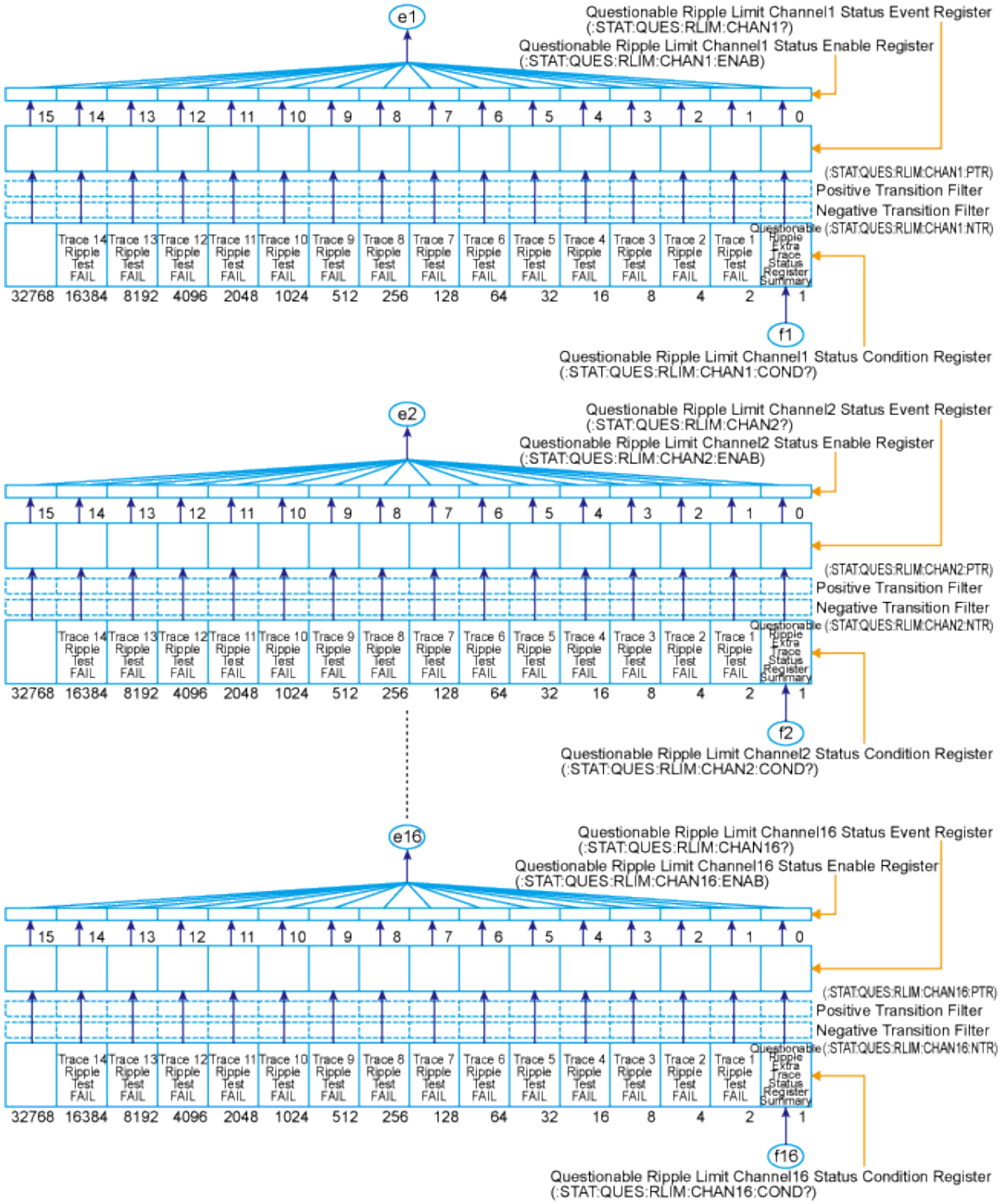
Issuing the *CLS command will clear all bits from the questionable ripple limit status event register.

Status Bit Definitions of the Questionable Ripple Limit Extra Status Condition Register

Bit Position	Name	Description
0	Not used	Always 0
1	Channel 15 Ripple Test Fail (questionable ripple limit channel 15 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 15 status event register is set to "1."
2	Channel 16 Ripple Test Fail (questionable ripple limit channel 16 status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel 16 status event register is set to "1."
3-15	Not used	Always 0

Issuing the *CLS command will clear all bits from the questionable ripple limit extra status event register.

Status Register for Ripple Limit (Trace) (1 of 2)



e5071c324

Status Bit Definitions of the Questionable Ripple Limit Channel {1-16} Status Condition Register

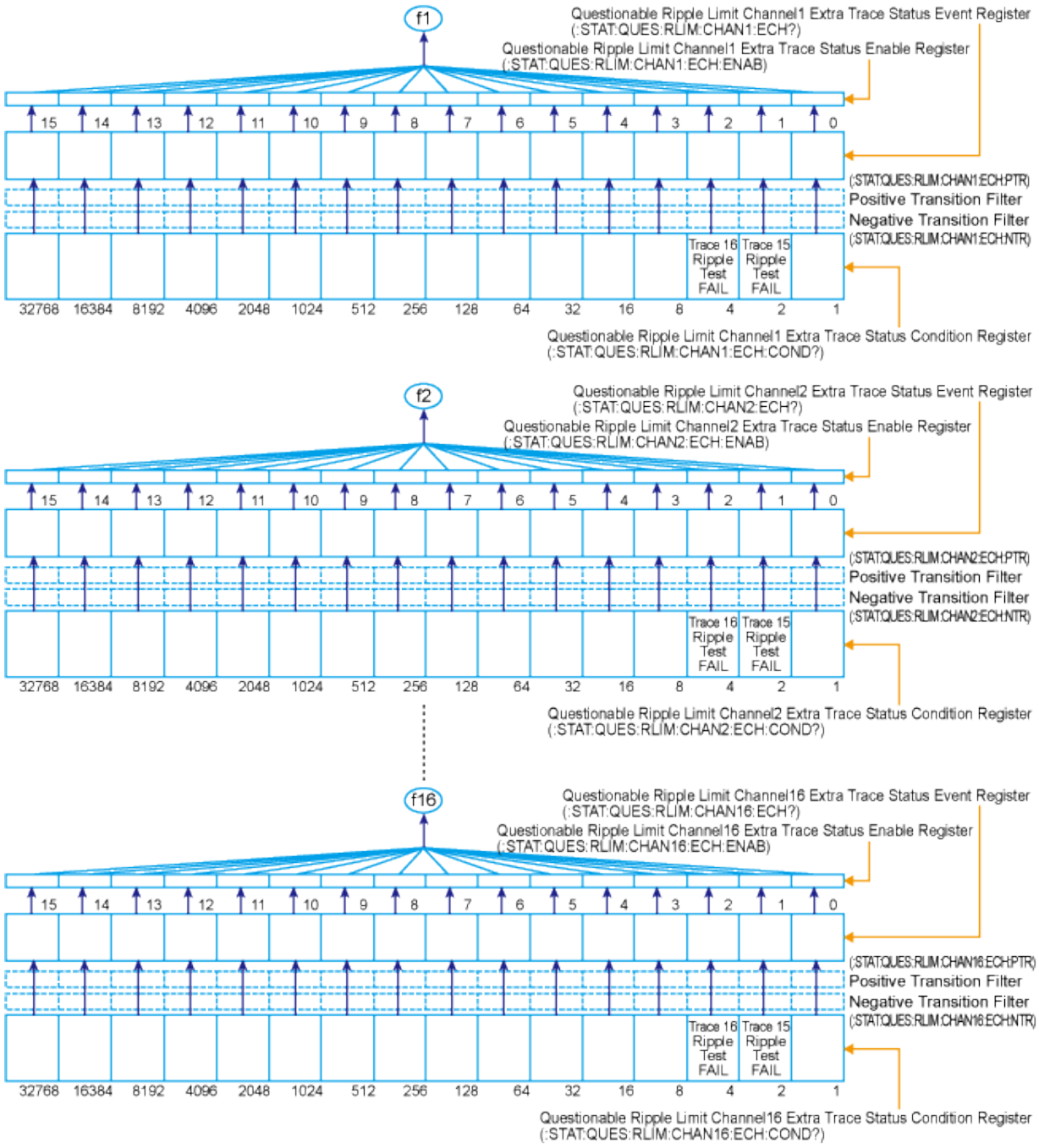
Bit Position	Name	Description
0	Trace 15, 16 Ripple Test summary (questionable ripple limit channel {1-16} extra status register summary)	Set to "1" while one of the enabled bits in the questionable ripple limit channel {1-16} extra status event register is set to "1."
1	Trace 1 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 1.
2	Trace 2 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 2.
3	Trace 3 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 3.
4	Trace 4 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 4.
5	Trace 5 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 5.

6	Trace 6 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 6.
7	Trace 7 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 7.
8	Trace 8 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 8.
9	Trace 9 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 9.
10	Trace 10 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 10.
11	Trace 11 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 11.
12	Trace 12 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the

		measurement cycle finishes and returns "fail" as the ripple test result for trace 12.
13	Trace 13 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 13.
14	Trace 14 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 14.
15	Not used	Always 0

Issuing the *CLS command will clear all the bits in the questionable ripple limit channel {1-16} status event register.

Status Register for Ripple Limit (Trace) (2 of 2)



e5071c225

Status Bit Definitions of the Questionable Ripple Limit Channel {1-16} Extra Status Condition Register

Bit Position	Name	Description

0	Not used	Always 0
1	Trace 15 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 15.
2	Trace 16 Ripple Test Fail	Set to "0" when a measurement cycle begins; set to "1" when the measurement cycle finishes and returns "fail" as the ripple test result for trace 16.
3-15	Not used	Always 0

Issuing the *CLS command will clear all the bits in the questionable ripple limit channel {1-16} extra status event register.

Working with Automatic Test System

Working with Automatic Test System

- Preventing Erroneous Key Operation on the Front Panel (key lock feature)
- Improving Command Processing Speed
- Detecting Occurrence of an Error

Preventing Erroneous Key Operation on the Front Panel (key lock feature)

When no operation is required from the front panel controls, the mouse, or the keyboard, disabling these input devices can prevent any erroneous operation that might be caused by accidentally touching the devices.

To turn on or off Key Locking, use the following commands:

Command	Description
:SYST:KLOC:KBD	Locking the front panel controls and the keyboard
:SYST:KLOC:MOUS	Locking the mouse and the touch screen

Improving Command Processing Speed

SCPI commands should be processed quickly to improve throughput when such commands are frequently executed (for example, reading out traces for each measurement).

With the E5071C, the processing time for SCPI commands can be improved by decreasing the refresh rate of the LCD display.

Measurement results (trace) do not need to be updated

When the measurement trace does not need to be updated, turn off the updating of the LCD display. This improves the processing speed of SCPI commands and eliminates the time used for updating the screen.

To turn off the updating of the LCD display, use the following command:

:DISP:ENAB

Measurement results (trace) need to be updated

When the measurement trace needs to be updated, the processing speed of SCPI commands can still be improved by controlling the update timing of the LCD display:

1. Execute all SCPI commands that are required before measurement, including commands setting conditions.
2. Turn OFF the update of the LCD display.
3. Perform the measurement.
4. Execute the commands for reading out measurement result or analyzing the result. Note that reading out the result in binary format will accelerate data transfer.
5. Execute the following command to update the LCD display once
   :DISP:UPD
6. Return to Step 3.

Sample program

See Control LCD Update Timing.

Detecting Occurrence of an Error

- Using Status Reporting System
- Using Error Queue
- Sample Program

Other topics about Working with Automatic Test System

Using Status Reporting System

The status of the E5071C can be detected through the status registers. This section describes how to detect the end of measurement by using the status registers.

The occurrence of an error will be present in the standard event status register. An SRQ (service request) is useful when you create a program that uses the information reported by this register to detect the occurrence of an error.

To detect the end of sweep via an SRQ, use one of the following commands:

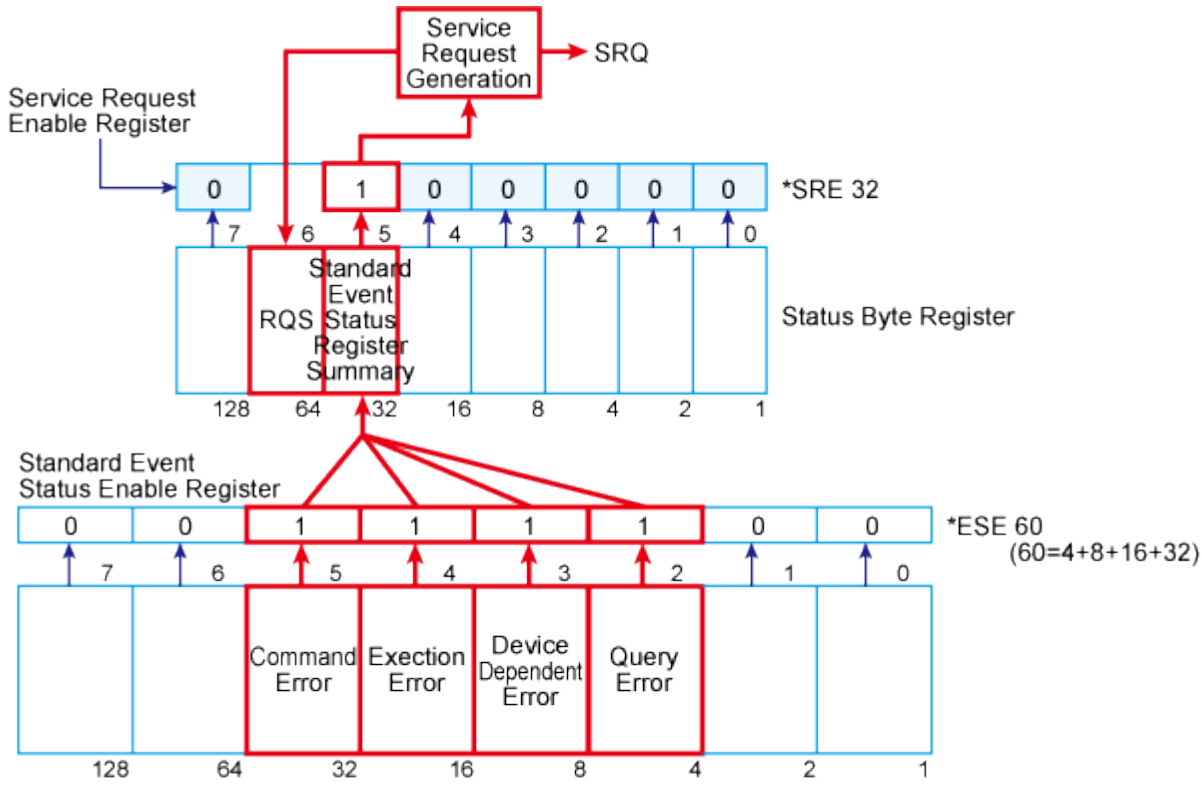
*SRE

*ESE

Follow these steps:

1. Set the E5071C so that it generates an SRQ when any of the error occurrence bits is set to 1 in the standard event status register.
2. When an SRQ is generated, the program interrupts the measurement cycle.

SRQ generation sequence (when an error occurs)



e5071c357

Using Error Queue

An error queue holds the number for the error and the error message. Reading the error queue allows the user to verify the error that has occurred. To retrieve the content of an error queue, use the following command:

:SYST:ERR?

The error queue can be used in the following ways:

1. It is used as a branch for error handling. When an error queue is retrieved, it returns 0 as the error number and "No error" as the error message if no error is detected. This can be used for detecting of an error and for branching the flow of a program. This is also useful when you wish to handle a specific error(s). Note that this method will not allow the user to perform any processing during the occurrence of an error.
2. When an error is detected using SRQ, the error queue is used to examine the error. Refer to the sample program in this section.

Sample Program

See Error Detection (SRQ).

Sample Programs

Sample Programs

This section shows sample programs with the SCPI commands which can be executed from the external controller. See Application Programs under VBA Programming about the sample programs for built-in VBA.

- Analyzer Setup
- Calibration
- ECal
- Power Calibration
- Reading/Writing Error Coefficient
- Waiting for Trigger (OPC?)
- Waiting for Trigger (SRQ)
- Error Detection (SRQ)
- Reading Data in Ascii Format
- Reading Data in Binary Format
- Writing Data in Ascii Format
- Writing Data in Binary Format
- Peak Search
- Bandwidth Search
- Limit Test
- Saving Files
- Transferring Files
- Fixture Simulator
- Time Domain
- Control Using SICL-LAN Server
- Controlling Using Telnet Server
- Handler Interface
- Controlling E5091A

Analyzer Setup

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The program listed in this section is a sample program that demonstrates how to configure measurement conditions.

The sample program puts the instrument into the preset state, configures it as shown in table below, and saves the settings to a file named **sample.sta**.

See Setting up Analyzer for this programming.

Target settings

Item		Setting
Window Layout		Channel 1 in the upper window (2/3 of the screen height) and channel 2 in the lower window (1/3 of the screen height)
Channel 1	Sweep type	Segment
	Sweep range	See Segment table.
	Number of measurement points	
	IF bandwidth	
	Power	
	Number of traces	4

	Graph Layout		Four graphs at upper left, upper right, lower left, and lower right.
	Trace 1	Measurement parameter	S11
		Data format	Smith chart (Lin)
		Full-scale value	2
	Trace 2	Measurement parameter	S21
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 3	Measurement parameter	S12
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm

	Trace 4	Measurement parameter	S22
		Data format	Smith chart (Lin)
		Full-scale value	2
Channel 2	Sweep type		Linear
	Sweep range	Center value	1.9 GHz
		Span value	500 MHz
	Number of measurement points		101
	IF bandwidth		70 kHz
	Power		0 dBm
	Number of traces		4
	Graph Layout		Two graphs at left and right
	Trace 1	Measurement parameter	S21
		Data format	Log magnitude
		Reference division line number	9
		Reference division line value	2
		Scale per division	10 dBm
	Trace 2	Measurement parameter	S22
		Data format	Smith chart (Lin)
		Full-scale value	2

Segment table for channel 1

Segment Number	Start value	Stop value	Number of measurement points	IF bandwidth	Power
1	1.7 GHz	1.9 GHz	21	50 kHz	0 dBm
2	1.9 GHz	2 GHz	101	10 kHz	-10 dBm
3	2 GHz	2.2 GHz	21	50 kHz	0 dBm

Sample Program in Excel VBA

```
Sub Setup()
  '
  Dim defrm As Long
  Dim vi As Long
  Const TimeOutTime = 20000

  Dim Allocate1 As String, Allocate2 As String, File As String
  Dim Para1(4) As String, Para2(2) As String
  Dim Fmt1(4) As String, Fmt2(2) As String
  Dim Star1(3) As String, Stop1(3) As String
  Dim IfBw1(3) As Double, IfBw2 As Double
  Dim Power1(3) As Double, Power2 As Double
  Dim Cent2 As Double, Span2 As Double
  Dim RefLev1(4) As Double, RefLev2(2) As Double, Scale1(4) As Double, Scale2(2) As Double
  Dim Segm As Integer, Nop1(3) As Integer, Nop2 As Integer
  Dim NumOfTr1 As Integer, NumOfTr2 As Integer
  Dim RefPos1(4) As Integer, RefPos2(2) As Integer
  Dim SendData As String
  '
  Segm = 3 ' Number of Segment Ch.1 : 3
  Star1(1) = "1.7E9"      ' Start Frequency Ch.1 Segm.1: 1.7 GHz
  Star1(2) = "1.9E9"      ' Segm.2: 1.9 GHz
  Star1(3) = "2E9"        ' Segm.3: 2 GHz
  Stop1(1) = "1.9E9"      ' Stop Frequency Ch.1 Segm.1: 1.9 GHz
```

```
Stop1(2) = "2E9"        ' Segm.2: 2 GHz
Stop1(3) = "2.2E9"      ' Segm.3: 2.2 GHz
Cent2 = 1900000000#     ' Center Frequency Ch.2 : 1.9 GHz
Span2 = 500000000#      ' Span Ch.2 : 500 MHz
Nop1(1) = 21            ' Number of points Segm.1: 21
Nop1(2) = 101           '              Segm.2: 101
Nop1(3) = 21            '              Segm.3: 21
Nop2 = 101              '       Ch.2 : 101
IfBw1(1) = 50000#       ' IFBW  Ch.1    Segm.1: 50 kHz
IfBw1(2) = 10000#       '               Segm.2: 10 kHz
IfBw1(3) = 50000#       '               Segm.3: 50 kHz
IfBw2 = 70000#          '       Ch.2 : 70 kHz
Power1(1) = 0           ' Power Ch.1    Segm.1: 0 dBm
Power1(2) = -10         '               Segm.2: -10 dBm
Power1(3) = 0           '               Segm.3: 0 dBm
Power2 = 0              '       Ch.2 : 0 dBm
NumOfTr1 = 4            ' Number of Trace Ch.1 : 4
NumOfTr2 = 2            ' Ch.2 : 2
Allocate1 = "D12_34"    ' Allocate Traces Ch.1 : D12_34
Allocate2 = "D12"       ' Ch.2 : D12
Para1(1) = "S11"        ' Measurement Ch.1 Trace1: S11
Para1(2) = "S21"        ' Parameter Trace2: S21
Para1(3) = "S12"        ' Trace3: S12
Para1(4) = "S22"        ' Trace4: S22
Para2(1) = "S31"        ' Ch.2 Trace1: S31
Para2(2) = "S33"        ' Trace2: S33
Fmt1(1) = "SLIN"        ' Data Format Ch.1 Trace1: Smith(Lin/Phase)
Fmt1(2) = "MLOG"        ' Trace2: Log Mag
Fmt1(3) = "MLOG"        ' Trace3: Log Mag
Fmt1(4) = "SLIN"        ' Trace4: Smith(Lin/Phase)
Fmt2(1) = "MLOG"        ' Ch.2 Trace1: Log Mag
Fmt2(2) = "SLIN"        ' Trace2: Smith(Lin/Phase)
RefPos1(1) = 9          ' Reference Ch.1 Trace2: 9
RefPos1(2) = 9          ' Position Trace3: 9
RefPos2(1) = 9          ' Ch.2 Trace1: 9
RefLev1(1) = 0          ' Reference Level Ch.1 Trace2: 0 dBm
RefLev1(2) = 0          ' Trace3: 0 dBm
```

```
    RefLev2(1) = 0          ' Ch.2 Trace1: 0 dBm
    Scale1(1) = 2           ' Scale Ch.1 Trace1: 2
    Scale1(2) = 10          ' Trace2: 10 dBm
    Scale1(3) = 10          ' Trace3: 10 dBm
    Scale1(4) = 2           ' Trace4: 2
    Scale2(1) = 10          ' Ch.2 Trace1: 10 dBm
    Scale2(2) = 2           ' Trace2: 2
    StaFileName = "sample.sta" ' Save File Name : sample.sta
    '
    ' Assigns a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime) ' Set time out
    '
    Call viVPrintf(vi, ":SYST:PRES" + vbLf, 0)
    Call viVPrintf(vi, ":DISP:SPL D1_1_2" + vbLf, 0) 'Allocate Channel
    Call viVPrintf(vi, ":INIT1:CONT ON" + vbLf, 0) 'Turn on Continuous Activation mode for channel 1
    Call viVPrintf(vi, ":INIT2:CONT ON" + vbLf, 0) 'Turn on Continuous Activation mode for channel 2
    ' ===============
    ' Setup Channel 1
    ' ===============
    Call viVPrintf(vi, ":SENS1:SWE:TYPE SEGM" + vbLf, 0) 'Sets channel 1 sweep type to segment
    '
    ' Create the data string for Segment Table
    SendData = "5,0,1,1,0,0," & Str(Segm)
    For i = 1 To Segm
        SendData = SendData + "," & Star1(i) & "," + Stop1(i) & "," & CStr(Nop1(i)) & "," & CStr(IfBw1(i)) &
"," & CStr(Power1(i))
    Next i
    Call viVPrintf(vi, ":SENS1:SEGM:DATA " + SendData + vbLf, 0)
    '
    Call viVPrintf(vi, ":CALC1:PAR:COUN " & CStr(NumOfTr1) & vbLf, 0) 'Set number of traces
    Call viVPrintf(vi, ":DISP:WIND1:SPL " & Allocate1 & vbLf, 0) 'Set graph layout


    For i = 1 To NumOfTr1
        Call viVPrintf(vi, ":CALC1:PAR" & CStr(i) & ":DEF " & Para1(i) & vbLf, 0) 'Set measurement
parameter
        Call viVPrintf(vi, ":CALC1:PAR" & CStr(i) & ":SEL" & vbLf, 0) ' Make trace active
```

```
        Call viVPrintf(vi, ":CALC1:FORM " & Fmt1(i) & vbLf, 0) ' Set data format
        '
        Select Case Fmt1(i)
        Case "SLIN", "SLOG", "SCOM", "SMIT", "SADM", "PLIN", "PLOG", "POL"
        ' If data format is neither Smith chart nor polar, sets reference division line number and scale per
division
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:PDIV " + CStr(Scale1(i)) + vbLf, 0)
        Case Else
        ' If data format is Smith chart or polar, set full-scale value
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:RPOS " & CStr(RefPos1(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:RLEV " & CStr(RefLev1(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND1:TRAC" & CStr(i) & ":Y:PDIV " & CStr(Scale1(i)) & vbLf, 0)
        End Select
    Next i
    ' ===============
    ' Setup Channel 2
    ' ===============
    Call viVPrintf(vi, ":SENS2:SWE:TYPE LIN " + vbLf, 0) ' Set sweep type to linear
    Call viVPrintf(vi, ":SENS2:FREQ:CENT " + CStr(Cent2) + vbLf, 0) ' Set center frequency
    Call viVPrintf(vi, ":SENS2:FREQ:SPAN " + CStr(Span2) + vbLf, 0) ' Set span frequency
    Call viVPrintf(vi, ":SENS2:SWE:POIN " + CStr(Nop2) + vbLf, 0) ' Set number of points
    Call viVPrintf(vi, ":SENS2:BAND " + CStr(IfBw2) + vbLf, 0) ' Set IFBW
    Call viVPrintf(vi, ":SOUR2:POW " + CStr(Power2) + vbLf, 0) ' Set power level
    Call viVPrintf(vi, ":CALC2:PAR:COUN " & CStr(NumOfTr2) & vbLf, 0) ' Set number of traces
    Call viVPrintf(vi, ":DISP:WIND2:SPL " & Allocate2 & vbLf, 0) 'Set graph layout


    For i = 1 To NumOfTr2
        Call viVPrintf(vi, ":CALC2:PAR" & CStr(i) & ":DEF " & Para2(i) & vbLf, 0) 'Set measurement
parameter
        Call viVPrintf(vi, ":CALC2:PAR" & CStr(i) & ":SEL" & vbLf, 0) ' Make trace active
        Call viVPrintf(vi, ":CALC2:FORM " & Fmt2(i) & vbLf, 0) ' Set data format
        '
        Select Case Fmt2(i)
        Case "SLIN", "SLOG", "SCOM", "SMIT", "SADM", "PLIN", "PLOG", "POL"
        ' If data format is neither Smith chart nor polar, sets reference division line number and scale per
division
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:PDIV " + CStr(Scale2(i)) + vbLf, 0)
        Case Else
```

```
        ' If data format is Smith chart or polar, set full-scale value
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:RPOS " & CStr(RefPos2(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:RLEV " & CStr(RefLev2(i)) & vbLf, 0)
            Call viVPrintf(vi, ":DISP:WIND2:TRAC" & CStr(i) & ":Y:PDIV " & CStr(Scale2(i)) & vbLf, 0)
        End Select
    Next i
    '
    Call viVPrintf(vi, ":MMEM:STOR """ & StaFileName & """ & vbLf, 0) ' Save ENA settings to file
    ' Close IO
    Call viClose(vi)
    Call viClose(defrm)

End Sub
```

Sample Program in HT Basic (setup.htb)

```
10 DIM Allocate1$[9],Allocate2$[9],File$[20]
20 DIM Para1$(1:4)[9],Para2$(1:2)[9],Fmt1$(1:4)[9],Fmt2$(1:2)[9]
30 REAL Star1(1:3),Stop1(1:3),Pow1(1:3),Cent2,Span2,Pow2
40 REAL Ref_rev1(1:4),Ref_rev2(1:2),Scale1(1:4),Scale2(1:2)
50 INTEGER Segm,Nop1(1:3),Nop2,Num_of_tr1,Num_of_tr2
60 INTEGER Ref_pos1(1:4),Ref_pos2(1:2),I
70 ASSIGN @Agte507x TO 717
80 !
90 Segm=3 ! Number of Segment Ch.1 : 3
100 Star1(1)=1.7E+9 ! Start Frequency Ch.1 Segm.1: 1.7 GHz
110 Star1(2)=1.9E+9 ! Segm.2: 1.9 GHz
120 Star1(3)=2.E+9 ! Segm.3: 2 GHz
130 Stop1(1)=1.9E+9 ! Stop Frequency Ch.1 Segm.1: 1.9 GHz
140 Stop1(2)=2.E+9 ! Segm.2: 2 GHz
150 Stop1(3)=2.2E+9 ! Segm.3: 2.2 GHz
160 Cent2=1.9E+9 ! Center Frequency Ch.2 : 1.9 GHz
170 Span2=5.00E+8 ! Span Ch.2 : 500 MHz
180 Nop1(1)=21 ! Number Ch.1 Segm.1: 21
190 Nop1(2)=101 ! of Points Segm.2: 101
200 Nop1(3)=21 ! Segm.3: 21
210 Nop2=101 ! Ch.2 : 101
220 If_bw1(1)=5.0E+4 ! IF Bandwidth    Ch.1 Segm.1: 50 kHz
230 If_bw1(2)=1.0E+4 ! Segm.2: 10 kHz
```

```
240 If_bw1(3)=5.0E+4 ! Segm.3: 50 kHz
250 If_bw2=7.0E+4 ! Ch.2 : 70 kHz
260 Pow1(1)=0 ! Power Ch.1 Segm.1: 0 dBm
270 Pow1(2)=-10 ! Segm.2: -10 dBm
280 Pow1(3)=0 ! Segm.3: 0 dBm
290 Pow2=0 ! Ch.2 : 0 dBm
300 Num_of_tr1=4 ! Number Ch.1 : 4
310 Num_of_tr2=2 ! of Traces Ch.2 : 2
320 Allocate1$="D12_34" ! Allocate Traces Ch.1 : D12_34
330 Allocate2$="D12" ! Ch.2 : D12
340 Para1$(1)="S11" ! Measurement Ch.1 Trace1: S11
350 Para1$(2)="S21" ! Parameter Trace2: S21
360 Para1$(3)="S12" ! Trace3: S12
370 Para1$(4)="S22" ! Trace4: S22
380 Para2$(1)="S31" ! Ch.2 Trace1: S31
390 Para2$(2)="S33" ! Trace2: S33
400 Fmt1$(1)="SLIN" ! Data Format Ch.1 Trace1: Smith(Lin/Phase)
410 Fmt1$(2)="MLOG" ! Trace2: Log Mag
420 Fmt1$(3)="MLOG" ! Trace3: Log Mag
430 Fmt1$(4)="SLIN" ! Trace4: Smith(Lin/Phase)
440 Fmt2$(1)="MLOG" ! Ch.2 Trace1: Log Mag
450 Fmt2$(2)="SLIN" ! Trace2: Smith(Lin/Phase)
460 Ref_pos1(2)=9 ! Reference Ch.1 Trace2: 9
470 Ref_pos1(3)=9 ! Position Trace3: 9
480 Ref_pos2(1)=9 ! Ch.2 Trace1: 9
490 Ref_lev1(2)=0 ! Reference Level Ch.1 Trace2: 0 dBm
500 Ref_lev1(3)=0 ! Trace3: 0 dBm
510 Ref_lev2(1)=0 ! Ch.2 Trace1: 0 dBm
520 Scale1(1)=2 ! Scale Ch.1 Trace1: 2
530 Scale1(2)=10 ! Trace2: 10 dBm
540 Scale1(3)=10 ! Trace3: 10 dBm
550 Scale1(4)=2 ! Trace4: 2
560 Scale2(1)=10 ! Ch.2 Trace1: 10 dBm
570 Scale2(2)=2 ! Trace2: 2
580 File$="sample.sta" ! Save File Name : sample.sta
590 !
600 OUTPUT @Agte507x;":SYST:PRES"
```

```
610 !
620 OUTPUT @Agte507x;":DISP:SPL D1_1_2"
630 OUTPUT @Agte507x;":INIT1:CONT ON"
640 OUTPUT @Agte507x;":INIT2:CONT ON"
650 !
660 ! Channel 1
670 !
680 OUTPUT @Agte507x;":SENS1:SWE:TYPE SEGM"
690 OUTPUT @Agte507x;":SENS1:SEGM:DATA 5,0,1,1,0,0,";Segm;",";
700 FOR I=1 TO Segm-1
710 OUTPUT @Agte507x;Star1(I);",";Stop1(I);",";Nop1(I);",";If_bw1 (I);",";Pow1(I);",";
720 NEXT I
730 OUTPUT @Agte507x;Star1(Segm);",";Stop1(Segm);",";Nop1(Segm);","
;If_bw1(Segm);",";Pow(Segm)
740 !
750 OUTPUT @Agte507x;":CALC1:PAR:COUN ";Num_of_tr1
760 OUTPUT @Agte507x;":DISP:WIND1:SPL "&Allocate1$
770 FOR I=1 TO Num_of_tr1
780 OUTPUT @Agte507x;":CALC1:PAR"&VAL$(I)&":DEF "&Para1$(I)
790 OUTPUT @Agte507x;":CALC1:PAR"&VAL$(I)&":SEL"
800 OUTPUT @Agte507x;":CALC1:FORM "&Fmt1$(I)
810 SELECT Fmt1$(I)
820 CASE "SLIN","SLOG","SCOM","SMIT","SADM","PLIN","PLOG","POL"
830 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:PDIV "; Scale1(I)
840 CASE ELSE
850 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:RPOS "; Ref_pos1(I)
860 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:RLEV "; Ref_rev1(I)
870 OUTPUT @Agte507x;":DISP:WIND1:TRAC"&VAL$(I)&":Y:PDIV "; Scale1(I)
880 END SELECT
890 NEXT I
900 !
910 ! Channel 2
920 !
930 OUTPUT @Agte507x;":SENS2:SWE:TYPE LIN"
940 OUTPUT @Agte507x;":SENS2:FREQ:CENT ";Cent2
950 OUTPUT @Agte507x;":SENS2:FREQ:SPAN ";Span2
960 OUTPUT @Agte507x;":SENS2:SWE:POIN ";Nop2
```

```
970 OUTPUT @Agte507x;":SENS2:BAND ";If_bw2
980 OUTPUT @Agte507x;":SOUR2:POW ";Pow2
990 !
1000 OUTPUT @Agte507x;":CALC2:PAR:COUN ";Num_of_tr2
1010 OUTPUT @Agte507x;":DISP:WIND2:SPL "&Allocate2$
1020 FOR I=1 TO Num_of_tr2
1030 OUTPUT @Agte507x;":CALC2:PAR"&VAL$(I)&":DEF "&Para2$(I)
1040 OUTPUT @Agte507x;":CALC2:PAR"&VAL$(I)&":SEL"
1050 OUTPUT @Agte507x;":CALC2:FORM "&Fmt2$(I)
1060 SELECT Fmt2$(I)
1070 CASE "SLIN","SLOG","SCOM","SMIT","SADM","PLIN","PLOG","POL"
1080 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:PDIV "; Scale2(I)
1090 CASE ELSE
1100 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:RPOS "; Ref_pos2(I)
1110 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:RLEV "; Ref_rev2(I)
1120 OUTPUT @Agte507x;":DISP:WIND2:TRAC"&VAL$(I)&":Y:PDIV "; Scale2(I)
1130 END SELECT
1140 NEXT I
1150 !
1160 OUTPUT @Agte507x;":MMEM:STOR """"&File$&""""
1170 END
```

Calibration

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program performs calibration with the specified calibration type.

See Calibration for this programming.

Sample Program in Excel VBA

```
Sub Cal_Click()
    Dim defrm As Long          'Session to Default Resource Manager
    Dim vi As Long             'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(4) As String

    Const TimeOutTime = 40000   'timeout time.
    Const Cal85032F = 4         'cal kit number.

    Ch = Cells(5, 5)           'Select channel
    Port(1) = Cells(3, 6)       'Sets the select port 1.
    Port(2) = Cells(3, 7)       'Sets the select port 2.
    Port(3) = Cells(3, 8)       'Sets the select port 3.
    Port(4) = Cells(3, 9)       'Sets the select port 4.
    CalKit = Cal85032F          'Sets cal kit (85032F)

    Call viOpenDefaultRM(defrm)    'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)    'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)   'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)   'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:CKIT " & CalKit & vbLf, 0)  'Select the calibration kit
```

```
    Select Case Cells(3, 5)
        Case "Response (Open)"   'Perform response calibration (OPEN).
            Call Cal_Resp(vi, Ch, "OPEN", Port(1))
        Case "Response (Short)"   'Perform response calibration (SHORT).
            Call Cal_Resp(vi, Ch, "Short", Port(1))
        Case "Response (Thru)"    'Perform response calibration (Thru).
            Call Cal_RespThru(vi, Ch, "Thru", Port(1), Port(2))
        Case "Full 1 Port"    'Perform 1-port calibration.
            Call Cal_Slot(vi, Ch, 1, Port)
        Case "Full 2 Port"     'Perform full 2-port calibration.
            Call Cal_Slot(vi, Ch, 2, Port)
        Case "Full 3 Port"     'Perform full 3-port calibration.
            Call Cal_Slot(vi, Ch, 3, Port)
        Case "Full 4 Port"    'Perform full 4-port calibration.
            Call Cal_Slot(vi, Ch, 4, Port)
    End Select

    Call viClose(vi)  'Closes the resource manager session.
    Call viClose(defrm)  'Breaks the communication and terminates the VISA system.

    End                                                    'End
End Sub

Sub Cal_Resp(vi As Long, Ch As String, CalType As String, Port As String)
    Dim Dummy As Variant   'Variant to receive the result

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:" & CalType & " " & Port & vbLf, 0)  'Sets the
calibration type.

    MsgBox ("Set " & CalType & " to Port " & Port & ". then click [OK] button")  'Display the message box.

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:" & CalType & " " & Port & vbLf, 0) 'Measurement the
calibration data.
    Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC? result.
    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)      'Calculating the calibration
coefficients.

    Call ErrorCheck(vi)      'Checking the error.
```

```
End Sub

Sub Cal_RespThru(vi As Long, Ch As String, CalType As String, Port1 As String, Port2 As String)
    Dim Dummy As Variant   'Variant to receive the result.

    If Port1 <> Port2 Then
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:" & CalType & " " & Port1 & "," & Port2 &
vbLf, 0)   'Sets the calibration type

        MsgBox ("Set " & CalType & " to Port " & Port1 & "&" & Port2 & ". then click [OK] button")   'Display
the message box.

        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:" & CalType & " " & Port1 & "," & Port2 & vbLf, 0)
 'Measurement the calibration data.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)      'Reads the *OPC? result.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)     'Calculating the calibration
coefficients.

        Call ErrorCheck(vi)     'Checking the error.
    Else
        MsgBox ("Thru calibration select port error!")   'Displaying the error message when selected same
ports.
        Exit Sub
    End If

End Sub
Sub Cal_Slot(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy
    Dim i As Integer, j As Integer

    Select Case NumPort
        Case 1
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & vbLf,
0)   'Set the 1-port calibration type.
        Case 2
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & vbLf, 0)   'Set the full 2-port calibration type.
        Case 3
```

```
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & "," & Port(3) & vbLf, 0)   'Set the full 3-port calibration type.
      Case 4
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT4 1,2,3,4" & vbLf, 0)   'Set the full 4-port calibration type.
    End Select
    'Reflection
    For i = 1 To NumPort
      MsgBox ("Set Open to Port " & Port(i) & ". then click [OK] button")   'Display the message box.
      Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:OPEN " & Port(i) & vbLf, 0)    'Measurement the OPEN calibration.
      Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC? result.

      MsgBox ("Set Short to Port " & Port(i) & ". then click [OK] button")    'Display the message box.
      Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SHORT " & Port(i) & vbLf, 0)  'Measurement the SHORT calibration.
      Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC? result.

      MsgBox ("Set Load to Port " & Port(i) & ". then click [OK] button")   'Display the message box.
      Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:LOAD " & Port(i) & vbLf, 0)    'Measurement the LOAD calibration.
      Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)     'Reads the *OPC? result.
    Next i
    'Transmission
    For i = 1 To NumPort - 1
      For j = i + 1 To NumPort
        MsgBox ("Set Thru to Port " & Port(i) & "&" & Port(j) & ". then click [OK] button")   'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(i) & "," & Port(j) & vbLf, 0)
  'Measurement the THRU calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC result.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(j) & "," & Port(i) & vbLf, 0)
  'Measurement the THRU calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)    'Reads the *OPC result.
      Next j
    Next i
    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)  'Calculating the calibration coefficients.

    Call ErrorCheck(vi)    'Checking the error.
```

```
End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)    'Reads error message.
    ErrNo = Split(err, ",")    'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)    'Display the message box.
    End If
End Sub
```

Sample Program in HT Basic (cal.htb)

```
10 DIM File$[20],Ch$[9],Inp_char$[9]
20 INTEGER Cal_kit,Cal_type,Port(1:4)
30 !
40 ASSIGN @Agte507x TO 717
50 File$="Ex_4_1.sta"
60 Ch$="1"
70 !
80 Select_cal_kit(@Agte507x,Ch$)
90 !
100 CLEAR SCREEN
110 ON ERROR GOTO Type_select
120 Type_select: !
130 PRINT "## Calibration Type Selection ##"
140 PRINT " 1: Response (Open)"
150 PRINT " 2: Response (Short)"
160 PRINT " 3: Response (Thru)"
170 PRINT " 4: Full 1 Port"
180 PRINT " 5: Full 2 Port"
190 PRINT " 6: Full 3 Port"
200 PRINT " 7: Full 4 Port"
210 PRINT ""
220 PRINT "Input 1 to 7"
230 INPUT "Input number? (1 to 7)",Inp_char$
240 Cal_type=IVAL(Inp_char$,10)
```

```
250 IF Cal_type<1 OR Cal_type>7 THEN Type_select
260 OFF ERROR
270 !
280 SELECT Cal_type
290 CASE 1
300 Select_port(1,Port(*))
310 Cal_resp(@Agte507x,Ch$,"OPEN",Port(1))
320 CASE 2
330 Select_port(1,Port(*))
340 Cal_resp(@Agte507x,Ch$,"SHOR",Port(1))
350 CASE 3
360 Select_port(2,Port(*))
370 Cal_resp_thru(@Agte507x,Ch$,Port(1),Port(2))
380 CASE 4
390 Select_port(1,Port(*))
400 Cal_solt(@Agte507x,Ch$,1,Port(*))
410 CASE 5
420 Select_port(2,Port(*))
430 Cal_solt(@Agte507x,Ch$,2,Port(*))
440 CASE 6
450 Select_port(3,Port(*))
460 Cal_solt(@Agte507x,Ch$,3,Port(*))
470 CASE 7
480 Select_port(4,Port(*))
490 Cal_solt(@Agte507x,Ch$,4,Port(*))
500 END SELECT
510 !
520 OUTPUT @Agte507x;":MMEM:STOR:STYP CST"
530 OUTPUT @Agte507x;":MMEM:STOR """&File$&"""
540 END
550 !=========================================
560 ! Calibration Kit Selection Function
570 !=========================================
580 SUB Select_cal_kit(@Agte507x,Ch$)
590 DIM Cal_kit_lbl$(1:10)[20],Inp_char$[9]
600 INTEGER Cal_kit,I
610 CLEAR SCREEN
```

```
620 !
630 FOR I=1 TO 10
640 OUTPUT @Agte507x;":SENS1:CORR:COLL:CKIT ";I
650 OUTPUT @Agte507x;":SENS1:CORR:COLL:CKIT:LAB?"
660 ENTER @Agte507x;Cal_kit_lbl$(I)
670 NEXT I
680 ON ERROR GOTO Kit_select
690 Kit_select: !
700 PRINT "## Calibration Kit Selection ##"
710 FOR I=1 TO 10
720 PRINT USING "X,2D,A,X,20A";I,":",Cal_kit_lbl$(I)
730 NEXT I
740 PRINT ""
750 PRINT "Input 1 to 10"
760 INPUT "Input number? (1 to 10)",Inp_char$
770 Cal_kit=IVAL(Inp_char$,10)
780 IF Cal_kit<1 OR Cal_kit>10 THEN Kit_select
790 OFF ERROR
800 !
810 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:CKIT ";Cal_kit
820 SUBEND
830 !==========================================
840 ! Port Selection Function
850 !==========================================
860 SUB Select_port(INTEGER Num_of_ports,INTEGER Port(*))
870 DIM Inp_char$[9]
880 !
890 CLEAR SCREEN
900 IF Num_of_ports=4 THEN
910 Port(1)=1
920 Port(2)=2
930 Port(3)=3
940 Port(4)=4
950 ELSE
960 PRINT "## Test Ports Selection ##"
970 ON ERROR GOTO Port_select
980 FOR I=1 TO Num_of_ports
```

```
990 PRINT "Port("&VAL$(I)&"):";
1000 Port_select:!
1010 INPUT "Number?",Inp_char$
1020 Port(I)=IVAL(Inp_char$,10)
1030 IF Port(I)<1 OR Port(I)>4 THEN Port_select
1040 FOR J=1 TO I-1
1050 IF Port(I)=Port(J) THEN Port_select
1060 NEXT J
1070 PRINT Port(I)
1080 NEXT I
1090 OFF ERROR
1100 END IF
1110 SUBEND
1120 !============================================
1130 ! Response (Open/Short) Calibration Function
1140 !============================================
1150 SUB Cal_resp(@Agte507x,Ch$,Type$,INTEGER Port)
1160 DIM Buff$[9]
1170 !
1180 PRINT "## Response ("&Type$&") Calibration ##"
1190 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:METH:"&Type$&" ";Port
1200 PRINT "Set "&Type$&" to Port "&VAL$(Port)&". Then push [Enter] key."
1210 INPUT "",Buff$
1220 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:"&Type$&" ";Port
1230 OUTPUT @Agte507x;"*OPC?"
1240 ENTER @Agte507x;Buff$
1250 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SAVE"
1260 PRINT "Done"
1270 SUBEND
1280 !============================================
1290 ! Response (Thru) Calibration Function
1300 !============================================
1310 SUB Cal_resp_thru(@Agte507x,Ch$,INTEGER Port1,Port2)
1320 DIM Buff$[9]
1330 !
1340 PRINT "## Response (Thru) Calibration ##"
1350 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:METH:THRU ";Port1;","; Port2
```

```
1360 PRINT "Set THRU between Port "&VAL$(Port1)&" and Port "&VAL$(Port2 )&". Then push [Enter] key."
1370 INPUT "",Buff$
1380 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port1;",";Port2
1390 OUTPUT @Agte507x;"*OPC?"
1400 ENTER @Agte507x;Buff$
1410 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SAVE"
1420 PRINT "Done"
1430 SUBEND
1440 !==========================================
1450 ! Full n Port Calibration Function
1460 !==========================================
1470 SUB Cal_solt(@Agte507x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
1480 DIM Buff$[9]
1490 INTEGER I,J
1500 !
1510 PRINT "## Full "&VAL$(Num_of_ports)&" Port Calibration ##"
1520 !
1530 ! Calibration Type Selection
1540 !
1550 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:METH:SOLT"&VAL$(Num_of_ ports)&" ";
1560 FOR I=1 TO Num_of_ports-1
1570 OUTPUT @Agte507x;Port(I);",";
1580 NEXT I
1590 OUTPUT @Agte507x;Port(Num_of_ports)
1600 !
1610 ! Reflection Measurement
1620 !
1630 FOR I=1 TO Num_of_ports
1640 PRINT "Set OPEN to Port "&VAL$(Port(I))&". Then push [Enter] key."
1650 INPUT "",Buff$
1660 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:OPEN ";Port(I)
1670 OUTPUT @Agte507x;"*OPC?"
1680 ENTER @Agte507x;Buff$
1690 PRINT "Set SHORT to Port "&VAL$(Port(I))&". Then push [Enter] key."
1700 INPUT "",Buff$
1710 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SHOR ";Port(I)
```

```
1720 OUTPUT @Agte507x;"*OPC?"
1730 ENTER @Agte507x;Buff$
1740 PRINT "Set LOAD to Port "&VAL$(Port(I))&". Then push [Enter] key."
1750 INPUT "",Buff$
1760 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:LOAD ";Port(I)
1770 OUTPUT @Agte507x;"*OPC?"
1780 ENTER @Agte507x;Buff$
1790 NEXT I
1800 !
1810 ! Transmission Measurement
1820 !
1830 FOR I=1 TO Num_of_ports-1
1840 FOR J=I+1 TO Num_of_ports
1850 PRINT "Set THRU between Port "&VAL$(Port(I))&" and Port "& VAL$(Port(J))&". Then push [Enter]
key."
1860 INPUT "",Buff$
1870 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(I);"," ;Port(J)
1880 OUTPUT @Agte507x;"*OPC?"
1890 ENTER @Agte507x;Buff$
1900 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(J);"," ;Port(I)
1910 OUTPUT @Agte507x;"*OPC?"
1920 ENTER @Agte507x;Buff$
1930 NEXT J
1940 NEXT I
1950 !
1960 ! Done
1970 !
1980 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SAVE"
1990 PRINT "Done"
2000 SUBEND
```

ECal

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program performs 1-port or 2-port calibration using ECal.

See Calibration for this programming.

Sample Program in Excel VBA

```
Sub ECal_Click()
    Dim defrm As Long          'Session to Default Resource Manager
    Dim vi As Long             'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(4) As String
    Const TimeOutTime = 40000   'timeout time.

    Ch = Cells(5, 5)           'Select channel
    Port(1) = Cells(3, 6)      'Sets the select port 1.
    Port(2) = Cells(3, 7)      'Sets the select port 2.
    Port(3) = Cells(3, 8)      'Sets the select port 3.
    Port(4) = Cells(3, 9)      'Sets the select port 4.

    Call viOpenDefaultRM(defrm)     'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)      'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)   'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)   'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.

    Select Case Cells(3, 5)
        Case "1 Port"
            Call ECal(vi, Ch, 1, Port)   'Perform 1-port calibration.
        Case "2 Port"
            Call ECal(vi, Ch, 2, Port)   'Perform full 2-port calibration.
```

```
        Case "3 Port"
            Call ECal(vi, Ch, 3, Port)   'Perform full 3-port calibration.
        Case "4 Port"
            Call ECal(vi, Ch, 4, Port)   'Perform full 4-port calibration.
    End Select

    Call viClose(vi)    'Closes the resource manager session.
    Call viClose(defrm)    'Breaks the communication and terminates the VISA system.

    End
End Sub
Sub ECal(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy As Variant
    Dim i As Integer, j As Integer

    Select Case NumPort
        Case 1
            MsgBox ("Connect Port " & Port(1) & ". then click [OK] button")  'Display the message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & vbLf,
0) 'Execute the 1-port calibration.
        Case 2
            MsgBox ("Connect Port " & Port(1) & " and Port " & Port(2) & ". then click [OK] button")   'Display
the message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & vbLf, 0) 'Execute the full 2-port calibration.
        Case 3
            MsgBox ("Connect Port " & Port(1) & "," & Port(2) & " and Port " & Port(3) & ". then click [OK]
button")  'Display the message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:ECAL:SOLT" & NumPort & " " & Port(1) & "," &
Port(2) & "," & Port(3) & vbLf, 0)   'Execute the full 3-port calibration.
        Case 4
            MsgBox ("Connect Port 1, 2, 3 and 4. then click [OK] button")    'Display the message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:ECAL:SOLT4 1,2,3,4" & vbLf, 0) 'Execute the full
4-port calibration.
    End Select

    Call ErrorCheck(vi)      'Checking the error.

End Sub
```

```
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)    'Reads error message.
    ErrNo = Split(err, ",")   'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)    'Display the message box.
    End If
End Sub
```

Sample Program in HT Basic (ecal.htb)

```
10 DIM File$[20],Ch$[9],Inp_char$[9]
20 INTEGER Cal_kit,Cal_type,Port(1:4)
30 !
40 ASSIGN @Agte507x TO 717
50 File$="Ex_4_2.sta"
60 Ch$="1"
70 !
80 CLEAR SCREEN
90 ON ERROR GOTO Type_select
100 Type_select: !
110 PRINT "## Calibration Type Selection ##"
120 PRINT " 1: Full 1 Port"
130 PRINT " 2: Full 2 Port"
140 PRINT " 3: Full 3 Port"
150 PRINT " 4: Full 4 Port"
160 PRINT ""
170 PRINT "Input 1 to 4"
180 INPUT "Input number? (1 to 4)",Inp_char$
190 Cal_type=IVAL(Inp_char$,10)
200 IF Cal_type<1 OR Cal_type>4 THEN Type_select
210 OFF ERROR
220 !
230 Select_port(Cal_type,Port(*))
240 Ecal(@Agte507x,Ch$,Cal_type,Port(*))
250 !
260 OUTPUT @Agte507x;":MMEM:STOR:STYP CST"
```

```
270 OUTPUT @Agte507x;":MMEM:STOR """"&File$&""""""
280 END
290 !=========================================
300 ! Port Selection Function
310 !=========================================
320 SUB Select_port(INTEGER Num_of_ports,INTEGER Port(*))
330 DIM Inp_char$[9]
340 !
350 CLEAR SCREEN
360 IF Num_of_ports=4 THEN
370 Port(1)=1
380 Port(2)=2
390 Port(3)=3
400 Port(4)=4
410 ELSE
420 PRINT "## Test Ports Selection ##"
430 ON ERROR GOTO Port_select
440 FOR I=1 TO Num_of_ports
450 PRINT "Port("&VAL$(I)&"):";
460 Port_select: !
470 INPUT "Number?",Inp_char$
480 Port(I)=IVAL(Inp_char$,10)
490 IF Port(I)<1 OR Port(I)>4 THEN Port_select
500 FOR J=1 TO I-1
510 IF Port(I)=Port(J) THEN Port_select
520 NEXT J
530 PRINT Port(I)
540 NEXT I
550 OFF ERROR
560 END IF
570 SUBEND
580 !=========================================
590 ! Electronic Calibration Function
600 !=========================================
610 SUB Ecal(@Agte507x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
620 DIM Buff$[9],Err_msg$[100]
630 INTEGER Err_no,Port1
```

```
640 !
650 PRINT "## Full "&VAL$(Num_of_ports)&" Port ECal ##"
660 !
670 OUTPUT @Agte507x;"*CLS"
680 SELECT Num_of_ports
690 CASE 1
700 PRINT "Connect Port "&VAL$(Port(1))&" to ECal Module."
710 PRINT "Then push [Enter] key."
720 INPUT "",Buff$
730 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:ECAL:SOLT1 ";Port(1)
740 CASE 2
750 PRINT "Connect Port "&VAL$(Port(1));
760 PRINT " and Port "&VAL$(Port(2))&" to ECal Module."
770 PRINT "Then push [Enter] key."
780 INPUT "",Buff$
790 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:ECAL:SOLT2 ";Port(1); ",";Port(2)
800 CASE 3
810 PRINT "Connect Port "&VAL$(Port(1));
820 PRINT ", Port "&VAL$(Port(2));
830 PRINT " and Port "&VAL$(Port(3))&" to ECal Module."
840 PRINT "Then push [Enter] key."
850 INPUT "",Buff$
860 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:ECAL:SOLT3 ";Port(1); ",";Port(2);",";Port(3)
870 CASE 4
880 PRINT "Connect Port 1, Port 2, Port 3 and Port 4 to to ECal Mod ule."
890 PRINT "Then push [Enter] key."
900 INPUT "",Buff$
910 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:ECAL:SOLT4 1,2,3,4"
920 END SELECT
930 PRINT "Executing ..."
940 OUTPUT @Agte507x;":SYST:ERR?"
950 ENTER @Agte507x;Err_no,Err_msg$
960 IF Err_no<>0 THEN
970 PRINT "Error occurred!!"
980 PRINT " No:";Err_no,"Description: "&Err_msg$
990 PRINT "ECAL INTERRUPT!!"
1000 ELSE
```

```
1010 PRINT "Done"
1020 END IF
1030 SUBEND
```

Power Calibration

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

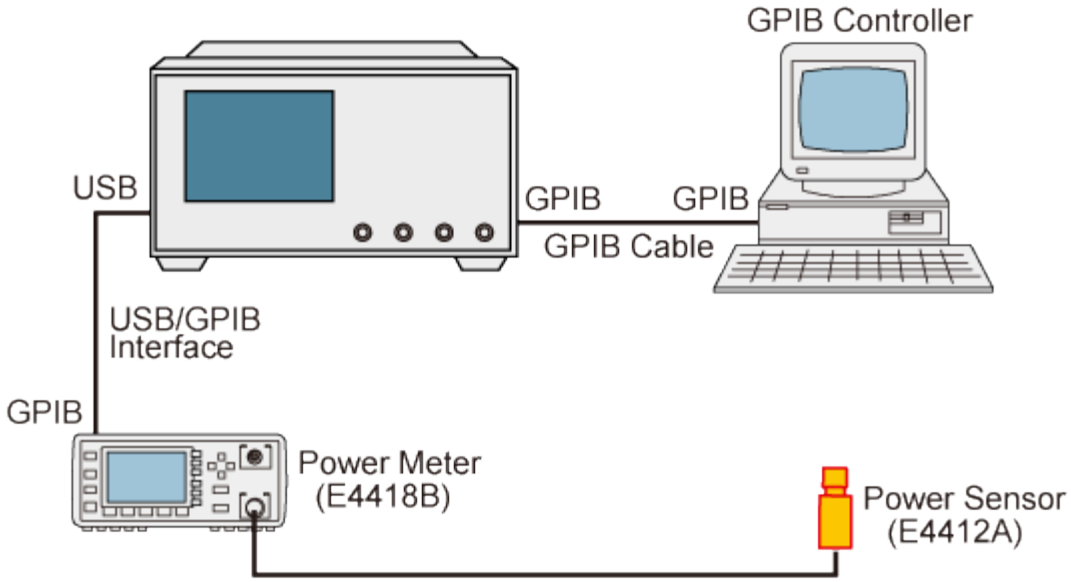
Other topics about Sample Programs

Overview

This program that demonstrates how to perform power calibration.

This program, as shown in the figure below, is run by making connections between the E5071C and the power meter (E4418B) through the USB/GPIB interface and between the E5071C and the external controller through the GPIB cable. Then the program executes the power calibration of the E5071C by using the power sensor (E4412A). The obtained power calibration data array is saved into a file.

Connecting E5071C, power meter, and external controller



e5071c334

See Power Calibration for this programming.

Sample Program in Excel VBA

```
Sub pow_cal_Click()
  '
  Dim defrm As Long
  Dim vi As Long
  Dim SwpType As String, StartPower As String, StopPower As String, CwFreq As String
  Dim Nop As Long, NumOfAve As String, Limit As Double, CorrData() As Double
```

```
    Dim Result As String * 10000, OpcRes As String * 2, Res As Variant
    Dim i As Long, Stat As VbMsgBoxResult
    Dim err As String * 50, ErrNo As Variant
    Const TimeOutTime = 50000 ' TimeOut time should be greater than the measurement time.
    ' Assign a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    SwpType = "POW"         ' Sweep type              : POWER
    Nop = 21                ' Number of points        : 41
    StartPower = "-20"      ' Start Power             : -20 dBm
    StopPower = "-10"       ' Stop Power              : -10 dBm
    CwFreq = "1E9"          ' CW frequency            : 1 GHz
    NumOfAve = "4"          ' Number of averaging     : 4
    Limit = 10              ' Limit for corrected data : 10 dBm
    '
    Call viVPrintf(vi, ":SYST:PRES" + vbLf, 0) ' Presetting the analyzer
    Call viVPrintf(vi, ":SYST:COMM:GPIB:PMET:ADDR 13" + vbLf, 0) ' Setting GPIB address of the power
meter to ENA
    '
    ' Setting measurement conditions
    Call viVPrintf(vi, ":SENS1:SWE:TYPE " & SwpType & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:SWE:POIN " & CStr(Nop) & vbLf, 0)
    Call viVPrintf(vi, ":SOUR1:POW:STAR " & StartPower & vbLf, 0)
    Call viVPrintf(vi, ":SOUR1:POW:STOP " & StopPower & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:FREQ " & CwFreq & vbLf, 0)


    Stat = MsgBox("Do you perform zeroing and calibrating the power sensor?", vbYesNo)


    If Stat = vbYes Then
       MsgBox "Zero and calibrate the power sensor by using the power meter, then press [OK] key.",
vbOKOnly
    End If


MeasStart:
    '
    ' Connecting the power sensor A to the port-1 of ENA
```

```
    Call viVPrintf(vi, "*CLS" + vbLf, 0)
    Stat = MsgBox("Set the power sensor connected to the port 1 in the ENA, then press [OK] key.",
vbOKOnly)
    '
    ' Performing power calibration measurement
    Call viVPrintf(vi, ":SOUR1:POW:PORT1:CORR:COLL:AVER " & NumOfAve & vbLf, 0)
    Call viVPrintf(vi, ":SOUR1:POW:PORT1:CORR:COLL ASEN" + vbLf, 0)
    Call viVPrintf(vi, "*OPC?" + vbLf, 0)
    Call viVScanf(vi, "%t", OpcRes)
    '
    ' Error hnadling at power meter measurement
    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
    ErrNo = Split(err, ",")
    '
    If Val(ErrNo(0)) = 0 Then
        ReDim CorrData(Nop)

        Call viVPrintf(vi, ":FORM:DATA ASC" + vbLf, 0)
        Call viVPrintf(vi, ":SOUR1:POW:PORT1:CORR:DATA?" + vbLf, 0)
        Call viVScanf(vi, "%t", Result)
        Res = Split(Result, ",")

        If fnLim(vi, Nop, Limit, Res) Then
            MsgBox "Power meter calibration measurement is complete.", vbOKOnly
            For i = 1 To Nop
                Cells(i + 5, 2) = i
                Cells(i + 5, 3) = Res(i - 1)
            Next i
        Else
            GoTo ReCalibration
        End If
    Else
        MsgBox "Error", vbOKOnly
        GoTo ReCalibration
    End If
ProgEnd:
    Call viClose(vi)
```

```
    Call viClose(defrm)
    Exit Sub
ReCalibration:
    Stat = MsgBox("Do you perform the power meter calibration measurement again?", vbYesNo)
    If Stat = vbYes Then
        GoTo MeasStart
    Else
        GoTo ProgEnd
    End If
End Sub
Function fnLim(vi As Long, Nop As Long, Limit As Double, Res As Variant) As Boolean
    For i = 1 To Nop
        If Abs(Res(i - 1)) > Limit Then
            Call viVPrintf(vi, "SOUR1:POW:PORT1:CORR OFF" + vbLf, 0)
            MsgBox "The corrected data is out of limit!", vbOKOnly
            fnLim = False
            Exit Function
        End If
    Next i
    fnLim = True
End Function
```

Sample Program in HT Basic (pow_cal.htb)

```
10 DIM Swp_type$[11],Inp_char$[9],Buff$[9],Err_mes$[50],File$[20]
20 DIM Corr_data(1:1601)
30 REAL Start_p,Stop_p,Cw_freq,Limit
40 INTEGER Nop,Pow_rang,Num_avg,Err_no,Verifier,Data_size,I
50 CLEAR SCREEN
60 !
70 ASSIGN @Agte507x TO 717
80 !
90 Swp_type$="POW" !Sweep type : POWER
100 Nop=41 !Number of points : 41
110 Pow_rang=0 !Power Range :-20 to +12 dBm
120 Start_p=-2.0E+1 !Start Power :-20 dBm
130 Stop_p=-1.0E+1 !Stop Power :-10 dBm
140 Cw_freq=1.0E+9 !CW frequency : 1 GHz
150 Num_avg=4 !Number of averaging : 4
```

```
160 Limit=10 !limit for corrected data : 10 dBm
170 !
180 ! Presetting the analyzer
190 !
200 OUTPUT @Agte507x;":SYST:PRES"
210 !
220 ! Setting GPIB address of the power meter to E5071C
230 !
240 OUTPUT @Agte507x;":SYST:COMM:GPIB:PMET:ADDR 13"
250 !
260 ! Setting measurement conditions
270 !
280 OUTPUT @Agte507x;":SENS1:SWE:TYPE "&Swp_type$
290 OUTPUT @Agte507x;":SENS1:SWE:POIN ";Nop
300 OUTPUT @Agte507x;":SOUR1:POW:ATT ";Pow_rang
310 OUTPUT @Agte507x;":SOUR1:POW:STAR ";Start_p
320 OUTPUT @Agte507x;":SOUR1:POW:STOP ";Stop_p
330 OUTPUT @Agte507x;":SENS1:FREQ ";Cw_freq
340 !
350 PRINT "Do you perform zeroing and calibrating the power sensor?"
360 PRINT
370 INPUT "[Y/N]",Inp_char$
380 IF UPC$(Inp_char$)="Y" THEN
390 PRINT "Zero and calibrate the power sensor by using the power meter, then press [Enter] key."
400 PRINT
410 INPUT "",Inp_char$
420 END IF
430 !
440 Meas_start: !
450 !
460 ! Connecting the power sensor to the port 1 in the ENA
470 !
480 OUTPUT @Agte507x;"*CLS"
490 PRINT "Set the power sensor connected to the port 1 in the ENA, then press [Enter] key."
500 PRINT
510 INPUT "",Inp_char$
520 !
```

```
530 ! Performing power calibration measurement
540 !
550 OUTPUT @Agte507x;":SOUR1:POW:PORT1:CORR:COLL:AVER ";Num_avg
560 OUTPUT @Agte507x;":SOUR1:POW:PORT1:CORR:COLL ASEN"
570 OUTPUT @Agte507x;"*OPC?"
580 ENTER @Agte507x;Buff$
590 !
600 ! Error hnadling at power meter measurement
610 !
620 OUTPUT @Agte507x;":SYST:ERR?"
630 ENTER @Agte507x;Err_no,Err_mes$
640 !
650 IF Err_no=0 THEN
660 REDIM Corr_data(1:Nop)
670 OUTPUT @Agte507x;":FORM:DATA ASC"
680 OUTPUT @Agte507x;":SOUR1:POW:PORT1:CORR:DATA?"
690 ENTER @Agte507x;Corr_data(*)
700 Verifier=FNLim(@Agte507x,Nop,Limit,Corr_data(*))
710 IF Verifier=-1 THEN
720 PRINT "Do you perform the power meter calibration measurement again?"
730 PRINT
740 INPUT "[Y/N]",Inp_char$
750 IF UPC$(Inp_char$)="Y" THEN GOTO Meas_start
760 IF UPC$(Inp_char$)<>"Y" THEN GOTO Prog_stop
770 END IF
780 PRINT "Power meter calibration measurement is complete."
790 PRINT
800 ELSE
810 PRINT "Error: "&Err_mes$
820 PRINT
830 PRINT "Do you perform the power meter calibration measurement again?"
840 PRINT
850 INPUT "[Y/N]",Inp_char$
860 IF UPC$(Inp_char$)="Y" THEN GOTO Meas_start
870 IF UPC$(Inp_char$)<>"Y" THEN GOTO Prog_stop
880 END IF
890 !
```

```
900 ! Installing the corrected data to a file
910 !
920 File$="CORR_DATA"
930 Data_size=Nop*8
940 ON ERROR GOTO Skip_purge
950 PURGE File$
960 Skip_purge: OFF ERROR
970 PRINT "The file installing power correction data: "&File$
980 PRINT
990 CREATE File$,Data_size
1000 ASSIGN @File TO File$;FORMAT ON
1010 FOR I=1 TO Nop
1020 OUTPUT @File USING "3D,3X,MD.4DE";I,Corr_data(I)
1030 NEXT I
1040 ASSIGN @File TO *
1050 !
1060 PRINT "Installing the corrected data to the file is DONE."
1070 !
1080 GOTO Prog_end
1090 !
1100 Prog_stop: !
1110 PRINT "Program Interruption"
1120 !
1130 Prog_end: !
1140 !
1150 END
1160 !
1170 ! Limit Test Function for the Corrected Data
1180 !
1190 DEF FNLim(@Agte507x,INTEGER Nop,REAL Limit,REAL Corr_data(*))
1200 INTEGER I
1210 !
1220 FOR I=1 TO Nop
1230 IF ABS(Corr_data(I))>Limit THEN
1240 OUTPUT @Agte507x;"SOUR1:POW:PORT1:CORR OFF"
1250 PRINT "The corrected data is out of limit!"
1260 PRINT
```

```
1270 RETURN -1
1280 GOTO Fn_exit
1290 END IF
1300 NEXT I
1310 !
1320 RETURN 0
1330 !
1340 Fn_exit: !
1350 !
1360 FNEND
```

Reading / Writing Error Coefficient

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program reads/writes the error coefficient.

This program will set measurement conditions and perform full 2-port calibration, preset the E5071C with the read error coefficient to be written, and then again read the error coefficient.

- The error coefficient read from the E5071C will be displayed in a graph.

Sample Program in Excel VBA

```
Sub Err_Term_Click()
    Dim defrm As Long          'Session to Default Resource Manager
    Dim vi As Long             'Session to instrument
    Dim Ch As String
    Dim CalKit As Integer
    Dim Port(2) As String
    Dim Result As String * 10
    Dim tNop As Long
    Dim Respons As String
    Dim Stimulus As String
    Dim ErrTerm As String

    Const TimeOutTime = 40000   'timeout time.
    Const Cal85032F = 4         'cal kit number

    Ch = Cells(2, 6)           'Select channel
    Port(1) = Cells(4, 6)       'Sets the select port 1.
    Port(2) = Cells(5, 6)       'Sets the select port 2.
    Respons = Cells(6, 6)       'Sets the respons port.
    Stimulus = Cells(7, 6)      'Sets the stimulus port.
    ErrTerm = Cells(8, 6)       'Sets the error term.

    CalKit = Cal85032F          'Set cal kit (85032F)
```

```
    Call viOpenDefaultRM(defrm)    'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)    'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)      'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)   'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.

    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:CKIT " & CalKit & vbLf, 0)  'Select the calibration kit.

    Call Set_sgm_tbl(vi, Ch)     'Configures the segment table.

    Select Case Cells(3, 6)    'Sets the read/write.
       Case "Read"
          Call Cal_Slot(vi, Ch, 2, Port)  'Full 2-Port Calibration.
       Case "Write"
          Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF:METH:SOLT2 1,2" & vbLf, 0)   'Sets the
calibration type to the full 2-port calibration.
    End Select

    Call viVPrintf(vi, ":SENS" & Ch & ":SEGM:SWE:POIN?" & vbLf, 0)  'Reads out the total number of the
measurement points of all segments.
    Call viVScanf(vi, "%t", Result)

    Call Exec_Error_Term(vi, Ch, Val(Result), ErrTerm, Respons, Stimulus)    'Reads the error coefficient.

    Call viClose(vi)      'Closes the resource manager session.
    Call viClose(defrm)      'Breaks the communication and terminates the VISA system.

    End
End Sub
Sub Exec_Error_Term(vi As Long, Ch As String, Nop As Long, ErrTerm As String, Respons As String,
Stimulus As String)

    Dim Error_Term_Data As Variant
    Dim Freq_Data As Variant
    Dim i As Integer, j As Integer
```

```
    Dim SelMode As String
    Dim Result As String * 10000
    Dim RealData As Double
    Dim ImagData As Double
    Dim FreqData As Double

    ReDim Error_Term_Data(Nop * 2) As String        'Defines the stock variables for the error coefficient
as needed for NOP.
    ReDim Freq_Data(Nop) As String                  'Defines the stock variables for the frequency values.

    SelMode = Cells(3, 6)   'Reads the read/write mode.

    Select Case SelMode
        Case "Read"                             'Reads the error coefficient from the ena.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF? " & ErrTerm & "," & Respons & "," & Stimulus &
vbLf, 0)  'Read the calibration coefficient data.
            Call viVScanf(vi, "%t", Result)
            Error_Term_Data = Split(Result, ",")    'Splits the read data by comma.

            Freq_Data = Make_Freq(vi, Nop)       'Calculates the frequency values.

            For i = 0 To Nop - 1
                RealData = CDbl(Error_Term_Data(i * 2))       'Reads the real data from error coefficient items.
                ImagData = CDbl(Error_Term_Data(i * 2 + 1))  'Reads the imag data from error coefficient
items.
                FreqData = CDbl(Freq_Data(i + 1))              'Reads the frequency values.
                Cells(10 + i, 2) = RealData                   'Displays the real data to the excel sheet.
                Cells(10 + i, 3) = ImagData                   'Displays the imag data to the excel sheet.
                Cells(10 + i, 1) = FreqData                   'Displays the frequency values to the excel sheet.
            Next i

            Call Data_Plot(vi, Nop, ErrTerm)                   'Displays the graph to the excel sheet.

        Case "Write"     'Write the error coefficient to the ena.
            Error_Term_Data = ErrTerm & "," & Respons & "," & Stimulus     'Sets the command parameter.
            For i = 0 To Nop - 1
                RealData = Cells(10 + i, 2)                   'Retrieves the real data from the excel sheet.
                ImagData = Cells(10 + i, 3)                   'Retrieves the imag data from the excel sheet.
```

```
            Error_Term_Data = Error_Term_Data & "," & RealData & "," & ImagData         'Sets the
command parameter.
        Next i

        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF " & Error_Term_Data & vbLf, 0)    'Write the
calibration coefficient data.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COEF:SAVE" & vbLf, 0)          'Calculates the
calibration coefficients.

    End Select
End Sub
Function Make_Freq(vi As Long, tPoint As Long) As Variant
    Dim start_freq As Double
    Dim stop_freq As Double
    Dim Nop As Integer
    Dim fStep As Double
    Dim fPoint As Double
    Dim freq_arry() As Variant
    Dim MeasPoint As Integer

    Const SegmentCnt = 2        'number of segment table.

    ReDim freq_arry(tPoint) As Variant

    MeasPoint = 1

    For j = 1 To SegmentCnt
        start_freq = Cells(3 + j - 1, 9)                    'Sets the start frequency of segment table.
        stop_freq = Cells(3 + j - 1, 10)                    'Sets the stop frequency of segment table.
        Nop = Cells(3 + j - 1, 13)                          'Sets the nop of segment table.
        fStep = (stop_freq - start_freq) / (Nop - 1)              'Calculate the frequency step.

        fPoint = start_freq                                 'Sets the frequency start point.
        For i = 1 To Nop
            freq_arry(MeasPoint) = fPoint                         'Sets the frequency value.
            fPoint = fPoint + fStep                         'Calculate the frequency points.
            MeasPoint = MeasPoint + 1                             'Add to measurement points.
        Next i
```

```
    Next j

    Make_Freq = freq_arry                                   'Sets the frequency data array.

End Function
Sub Data_Plot(vi As Long, Nop As Long, ErrTerm As String)
    Range("B10:C" & Nop + 9 & "").Select    'Select the error coefficient.
    Charts.Add
    ActiveChart.ChartType = xlLineStacked     'Sets the chart type.
    ActiveChart.SetSourceData Source:=Sheets("Error Term").Range("A9:C" & Nop + 9 & "") 'Sets the
error coefficient and displays the graph.
    ActiveChart.Location Where:=xlLocationAsObject, Name:="Error Term"

    ActiveChart.Axes(xlCategory).Select    'Select the formatting of X-axis.
    With Selection
        .TickLabelPosition = xlLow     'Displays the frequency values to low area.
    End With

    With ActiveChart
        .HasTitle = True
        .ChartTitle.Characters.Text = "Error Term " & ErrTerm     'Display the title.
    End With
End Sub
Sub Set_sgm_tbl(vi As Long, Ch As String)
    Dim Star1(2) As Double, Stop1(2) As Double, Pow1(2) As Double, If_bw1(2) As Double
    Dim Segm As Integer, Nop1(2) As Integer, Num_of_tr1 As Integer
    Dim i As Integer

    Segm = 2
    Star1(1) = Cells(3, 9)              'Sets the start frequency of segment 1 table.
    Stop1(1) = Cells(3, 10)              'Sets the stop frequency of segment 1 table.
    Pow1(1) = Cells(3, 11)               'Sets the power of segment 1 table.
    If_bw1(1) = Cells(3, 12)             'Sets the ifbw of segment 1 table.
    Nop1(1) = Cells(3, 13)               'Sets the nop of segment 1 table.
    Star1(2) = Cells(4, 9)              'Sets the start frequency of segment 2 table.
    Stop1(2) = Cells(4, 10)              'Sets the stop frequency of segment 2 table.
    Pow1(2) = Cells(4, 11)               'Sets the power of segment 2 table.
```

```
    If_bw1(2) = Cells(4, 12)            'Sets the ifbw of segment 2 table.
    Nop1(2) = Cells(4, 13)              'Sets the nop of segment 2 table.


    Call viVPrintf(vi, ":SENS" & Ch & ":SWE:TYPE SEGM" & vbLf, 0)   'Sets sweep type to segment.
    Call viVPrintf(vi, ":SENS" & Ch & ":SEGM:DATA 5,0,1,1,0,0," & Segm & ",", 0)       'Sets the header of segment table.
    Call viVPrintf(vi, Star1(1) & "," & Stop1(1) & "," & Nop1(1) & "," & If_bw1(1) & "," & Pow1(1) & ",", 0)  'Sets the 1st parameter.
    Call viVPrintf(vi, Star1(2) & "," & Stop1(2) & "," & Nop1(2) & "," & If_bw1(2) & "," & Pow1(2) & vbLf, 0)  'Sets the 2nd parameter.


    Call ErrorCheck(vi)   'Checking the error.


End Sub
Sub Cal_Slot(vi As Long, Ch As String, NumPort As String, Port() As String)
    Dim Dummy
    Dim i As Integer, j As Integer


    Select Case NumPort
        Case 1
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & vbLf, 0)   'Set the 1-port calibration type.
        Case 2
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & vbLf, 0)  'Set the full 2-port calibration type.
        Case 3
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT" & NumPort & " " & Port(1) & "," & Port(2) & "," & Port(3) & vbLf, 0)   'Set the full 3-port calibration type.
        Case 4
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:METH:SOLT4 1,2,3,4" & vbLf, 0)    'Set the full 4-port calibration type.
    End Select
    'Reflection
    For i = 1 To NumPort
        MsgBox ("Set Open to Port " & Port(i) & ". then click [OK] button")    'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:OPEN " & Port(i) & vbLf, 0)      'Measurement the OPEN calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)     'Reads the *OPC? result.
```

```
        MsgBox ("Set Short to Port " & Port(i) & ". then click [OK] button")  'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SHORT " & Port(i) & vbLf, 0)  'Measurement the
SHORT calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)  'Reads the *OPC? result.


        MsgBox ("Set Load to Port " & Port(i) & ". then click [OK] button")   'Display the message box.
        Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:LOAD " & Port(i) & vbLf, 0)   'Measurement the
LOAD calibration.
        Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)   'Reads the *OPC? result.
    Next i
    'Transmission
    For i = 1 To NumPort - 1
        For j = i + 1 To NumPort
            MsgBox ("Set Thru to Port " & Port(i) & "&" & Port(j) & ". then click [OK] button")  'Display the
message box.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(i) & "," & Port(j) & vbLf, 0)
 'Measurement the THRU calibration.
            Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)   'Reads the *OPC? result.
            Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:THRU " & Port(j) & "," & Port(i) & vbLf, 0)
 'Measurement the THRU calibration.
            Call viVQueryf(vi, "*OPC?" & vbLf, "%t", Dummy)      'Reads the *OPC? result.
        Next j
    Next i


    Call viVPrintf(vi, ":SENS" & Ch & ":CORR:COLL:SAVE" & vbLf, 0)    'Calculating the calibration
coefficients.


    Call ErrorCheck(vi)   'Checking the error.


End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response


    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)   'Reads error message.
    ErrNo = Split(err, ",")      'Gets the error code.


    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)    'Display the message box.
    End If
```

```
End Sub
```

Sample Program in HT Basic (ErrTerm.htb)

```
2000 Main:!
2010 INTEGER Agte507x,Ii,Nop
2020 INTEGER Respons,Stimulas
2030 INTEGER Port(1:2)
2040 REAL Stok(12,1:5000)
2050 REAL Stok2(12,1:5000)
2060 REAL Stok3(12,1:5000)
2070 DIM Ch$[10],Wk$[128]
2080 !
2090 ! PC's Monitor Clear
2100 CLEAR SCREEN
2110 GINIT
2130 !
2140 ! Set ENA++'s Addr
2150 Agte507x=717
2160 !
2170 Ch$="1"
2180 !
2190 ! Set ENA++'s I/O Path
2200 ASSIGN @Agte507x TO Agte507x
2210 !
2220 ON TIMEOUT SC(@Agte507x),15 RECOVER Tout
2230 !
2240 ! Set Start Port and End Port
2250 Port(1)=1
2260 Port(2)=2
2270 !
2280 ! Setup Segment Table
2290 CALL Set_sgm_tbl(@Agte507x)
2300 !
2310 ! Select Cal Kit
2320 CALL Select_cal_kit(@Agte507x,Ch$)
2330 !
2340 ! Execute Full-2Port Calibration
2350 CALL Cal_solt(@Agte507x,Ch$,2,Port(*))
```

```
2360 !
2370 ! Get All Segment's Points
2380 CALL Get_nop(@Agte507x,Nop,Ch$)
2390 !
2400 REDIM Stok(12,1:Nop*2)
2410 REDIM Stok2(12,1:Nop*2)
2420 REDIM Stok3(12,1:Nop*2)
2430 !
2440 CALL Exec_error_term(@Agte507x,"READ","ES",Ch$,1,Nop,1,1,Stok(*))
2450 CALL Exec_error_term(@Agte507x,"READ","ES",Ch$,2,Nop,2,2,Stok(*))
2460 CALL Exec_error_term(@Agte507x,"READ","ER",Ch$,3,Nop,1,1,Stok(*))
2470 CALL Exec_error_term(@Agte507x,"READ","ER",Ch$,4,Nop,2,2,Stok(*))
2480 CALL Exec_error_term(@Agte507x,"READ","ED",Ch$,5,Nop,1,1,Stok(*))
2490 CALL Exec_error_term(@Agte507x,"READ","ED",Ch$,6,Nop,2,2,Stok(*))
2500 !
2510 CALL Exec_error_term(@Agte507x,"READ","EL",Ch$,7,Nop,1,2,Stok(*))
2520 CALL Exec_error_term(@Agte507x,"READ","EL",Ch$,8,Nop,2,1,Stok(*))
2530 CALL Exec_error_term(@Agte507x,"READ","ET",Ch$,9,Nop,1,2,Stok(*))
2540 CALL Exec_error_term(@Agte507x,"READ","ET",Ch$,10,Nop,2,1,Stok(*))
2550 !
2560 CLEAR SCREEN
2570 PRINT "Push [Preset] - OK of ENA. Then push [Enter] key."
2580 INPUT "",Wk$
2590 !
2600 CALL Set_sgm_tbl(@Agte507x)
2610 !
2620 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COEF:METH:SOLT2 ";Port(1);",";Port(2)
2630 !
2640 CALL Exec_error_term(@Agte507x,"WRITE","ES",Ch$,1,Nop,1,1,Stok(*))
2650 CALL Exec_error_term(@Agte507x,"WRITE","ES",Ch$,2,Nop,2,2,Stok(*))
2660 CALL Exec_error_term(@Agte507x,"WRITE","ER",Ch$,3,Nop,1,1,Stok(*))
2670 CALL Exec_error_term(@Agte507x,"WRITE","ER",Ch$,4,Nop,2,2,Stok(*))
2680 CALL Exec_error_term(@Agte507x,"WRITE","ED",Ch$,5,Nop,1,1,Stok(*))
2690 CALL Exec_error_term(@Agte507x,"WRITE","ED",Ch$,6,Nop,2,2,Stok(*))
2700 !
2710 CALL Exec_error_term(@Agte507x,"WRITE","EL",Ch$,7,Nop,1,2,Stok(*))
2720 CALL Exec_error_term(@Agte507x,"WRITE","EL",Ch$,8,Nop,2,1,Stok(*))
```

```
2730 CALL Exec_error_term(@Agte507x,"WRITE","ET",Ch$,9,Nop,1,2,Stok(*))
2740 CALL Exec_error_term(@Agte507x,"WRITE","ET",Ch$,10,Nop,2,1,Stok(*))
2750 !
2760 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COEF:SAVE"
2770 !
2780 CALL Exec_error_term(@Agte507x,"READ","ES",Ch$,1,Nop,1,1,Stok2(*))
2790 CALL Exec_error_term(@Agte507x,"READ","ES",Ch$,2,Nop,2,2,Stok2(*))
2800 CALL Exec_error_term(@Agte507x,"READ","ER",Ch$,3,Nop,1,1,Stok2(*))
2810 CALL Exec_error_term(@Agte507x,"READ","ER",Ch$,4,Nop,2,2,Stok2(*))
2820 CALL Exec_error_term(@Agte507x,"READ","ED",Ch$,5,Nop,1,1,Stok2(*))
2830 CALL Exec_error_term(@Agte507x,"READ","ED",Ch$,6,Nop,2,2,Stok2(*))
2840 !
2850 CALL Exec_error_term(@Agte507x,"READ","EL",Ch$,7,Nop,1,2,Stok2(*))
2860 CALL Exec_error_term(@Agte507x,"READ","EL",Ch$,8,Nop,2,1,Stok2(*))
2870 CALL Exec_error_term(@Agte507x,"READ","ET",Ch$,9,Nop,1,2,Stok2(*))
2880 CALL Exec_error_term(@Agte507x,"READ","ET",Ch$,10,Nop,2,1,Stok2(*))
2890 !
2900 ASSIGN @Agte507x TO *
2910 !
2920 DISP CHR$(139)&" Done ..."&CHR$(136)
2930 STOP
2940 !
2950 Tout: OFF TIMEOUT SC(@Agte507x)
2960 !
2970 ASSIGN @Agte507x TO *
2980 !
2990 PRINT CHR$(137)&" ENA Timeout ..."&CHR$(136)
3000 END
3010!
3020 Set_sgm_tbl: SUB Set_sgm_tbl(@Agte507x)
3030 REAL Star1(1:2),Stop1(1:2),Pow1(1:2)
3040 INTEGER Segm,Nop1(1:2),Num_of_tr1
3050 INTEGER I
3060 !
3070 CLEAR SCREEN
3080 DISP CHR$(138)&" Wait ..."&CHR$(136)
3090 !
```

```
3100 Segm=2 ! Number of Segment Ch.1 : 2
3110 Star1(1)=3.E+6 ! Start Frequency Ch.1 Segm.1: 3.0 MHz
3120 Star1(2)=5.0E+7 ! Segm.2: 50.0 MHz
3130 Stop1(1)=1.0E+7 ! Stop Frequency Ch.1 Segm.1: 10.0 MHz
3140 Stop1(2)=8.E+9 ! Segm.2: 8.0 GHz
3150 Nop1(1)=2 ! Number Ch.1 Segm.1: 2
3160 Nop1(2)=170 ! of Points Segm.2: 170
3170 If_bw1(1)=7.0E+4 ! IF Bandwidth Ch.1 Segm.1: 70 kHz
3180 If_bw1(2)=7.0E+4 ! Segm.2: 70 kHz
3190 Pow1(1)=0 ! Power Ch.1 Segm.1: 0 dBm
3200 Pow1(2)=0 ! Segm.2: 0 dBm
3210 !
3220 OUTPUT @Agte507x;":SYST:PRES"
3230 !
3240 WAIT 5
3250 !
3260 ! Channel 1
3270 !
3280 OUTPUT @Agte507x;":SENS1:SWE:TYPE SEGM"
3290 OUTPUT @Agte507x;":SENS1:SEGM:DATA 5,0,1,1,0,0,";Segm;",";
3300 FOR I=1 TO Segm-1
3310 OUTPUT @Agte507x;Star1(I);",";Stop1(I);",";Nop1(I);",";If_bw1(I);",";Pow1(I);",";
3320 NEXT I
3330 OUTPUT
@Agte507x;Star1(Segm);",";Stop1(Segm);",";Nop1(Segm);",";If_bw1(Segm);",";Pow1(Segm)
3340 !
3350 OUTPUT @Agte507x;":CALC1:PAR:COUN ";Num_of_tr1
3360 FOR I=1 TO Num_of_tr1
3370 OUTPUT @Agte507x;":CALC1:PAR"&VAL$(I)&":SEL"
3380 NEXT I
3390 SUBEND
3400!
3410 Select_cal_kit: SUB Select_cal_kit(@Agte507x,Ch$)
3420 !=========================================
3430 ! Calibration Kit Selection Function
3440 !=========================================
3450 !
```

```
3460 DIM Cal_kit_lbl$(1:10)[20],Inp_char$[9]
3470 DIM Msg$[80],Wk$[10]
3480 INTEGER Cal_kit,I,Noc
3490 !
3500 ! PC's Monitor Clear
3510 CLEAR SCREEN
3520 !
3530 ! Number of Cal Kid
3540 Noc=10
3550 !
3560 FOR I=1 TO Noc
3570 OUTPUT @Agte507x;":SENS1:CORR:COLL:CKIT ";I
3580 OUTPUT @Agte507x;":SENS1:CORR:COLL:CKIT:LAB?"
3590 ENTER @Agte507x;Cal_kit_lbl$(I)
3600 NEXT I
3610 ON ERROR GOTO Kit_select
3620 !
3630 PRINT "## Calibration Kit Selection ##"
3640 FOR I=1 TO Noc
3650 PRINT USING "X,2D,A,X,20A";I,":",Cal_kit_lbl$(I)
3660 NEXT I
3670 PRINT ""
3680 PRINT "Input 1 to "&VAL$(Noc)
3690 !
3700 Msg$="Input number? (1 to "&VAL$(Noc)&") "
3710 LOOP
3720 LOOP
3730 DISP Msg$;
3740 INPUT Inp_char$
3750 Cal_kit=IVAL(Inp_char$,10)
3760 EXIT IF 1<=Cal_kit AND Cal_kit<=Noc
3770 Kit_select:!
3780 BEEP
3790 END LOOP
3800 !
3810 Wk$=""
3820 PRINT TABXY(1,Cal_kit+1);
```

```
3830 PRINT USING "X,B,2D,A,X,20A,B";139,Cal_kit,":",Cal_kit_lbl$(Cal_kit),136
3840 INPUT "Sure ? [Y/N]",Wk$
3850 EXIT IF (UPC$(Wk$)="Y")
3860 PRINT TABXY(1,Cal_kit+1);
3870 PRINT USING "X,2D,A,X,20A";Cal_kit,":",Cal_kit_lbl$(Cal_kit)
3880 BEEP
3890 BEEP
3900 END LOOP
3910 OFF ERROR
3920 !
3930 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:CKIT ";Cal_kit
3940 SUBEND
3950!
3960 Cal_solt: SUB Cal_solt(@Agte507x,Ch$,INTEGER Num_of_ports,INTEGER Port(*))
3970 !=========================================
3980 ! Full n Port Calibration Function
3990 !=========================================
4000 !
4010 DIM Buff$[9]
4020 INTEGER I,J
4030 !
4040 ! PC's Monitor Clear
4050 CLEAR SCREEN
4060 !
4070 PRINT "## Full "&VAL$(Num_of_ports)&" Port Calibration ##"
4080 !
4090 ! Calibration Type Selection
4100 !
4110 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:METH:SOLT"&VAL$(Num_of_ports)&" ";
4120 FOR I=1 TO Num_of_ports-1
4130 OUTPUT @Agte507x;Port(I);",";
4140 NEXT I
4150 OUTPUT @Agte507x;Port(Num_of_ports)
4160 !
4170 ! Reflection Measurement
4180 !
4190 FOR I=1 TO Num_of_ports
```

```
4200 PRINT "Set OPEN to Port "&VAL$(Port(I))&". Then push [Enter] key."
4210 INPUT "",Buff$
4220 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:OPEN ";Port(I)
4230 OUTPUT @Agte507x;"*OPC?"
4240 ENTER @Agte507x;Buff$
4250 PRINT "Set SHORT to Port "&VAL$(Port(I))&". Then push [Enter] key."
4260 INPUT "",Buff$
4270 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SHOR ";Port(I)
4280 OUTPUT @Agte507x;"*OPC?"
4290 ENTER @Agte507x;Buff$
4300 PRINT "Set LOAD to Port "&VAL$(Port(I))&". Then push [Enter] key."
4310 INPUT "",Buff$
4320 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:LOAD ";Port(I)
4330 OUTPUT @Agte507x;"*OPC?"
4340 ENTER @Agte507x;Buff$
4350 NEXT I
4360 !
4370 ! Transmission Measurement
4380 !
4390 FOR I=1 TO Num_of_ports-1
4400 FOR J=I+1 TO Num_of_ports
4410 PRINT "Set THRU between Port "&VAL$(Port(I))&" and Port "&VAL$(Port(J))&". Then push [Enter]
key."
4420 INPUT "",Buff$
4430 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(I);",";Port(J)
4440 OUTPUT @Agte507x;"*OPC?"
4450 ENTER @Agte507x;Buff$
4460 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:THRU ";Port(J);",";Port(I)
4470 OUTPUT @Agte507x;"*OPC?"
4480 ENTER @Agte507x;Buff$
4490 NEXT J
4500 NEXT I
4510 !
4520 ! Done
4530 !
4540 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COLL:SAVE"
4550 PRINT "Done"
```

```
4560 SUBEND
4570!
4580 Get_nop: SUB Get_nop(@Agte507x,INTEGER Nop,Ch$)
4590 ! Get All Segment's Points
4600 OUTPUT @Agte507x;":SENS"&Ch$&":SEGM:SWE:POIN?"
4610 ENTER @Agte507x;Nop
4620 SUBEND
4630 Exec_error_term: SUB Exec_error_term(@Agte507x,Rw$,Id$,Ch$,INTEGER
Idx,Nop,Respons,Stimulas,REAL Stok(*))
4640 INTEGER Ii
4650 REAL Error_term_data(1:5000)
4660 !
4670 DISP CHR$(138)&" Wait ..."&CHR$(136)
4680 !
4690 REDIM Error_term_data(1:Nop*2)
4700 !
4710 SELECT Rw$
4720 CASE "WRITE"
4730 FOR Ii=1 TO Nop
4740 Error_term_data(2*Ii-1)=Stok(Idx,2*Ii-1)
4750 Error_term_data(2*Ii)=Stok(Idx,2*Ii)
4760 NEXT Ii
4770 !
4780 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COEF
"&Id$&",";Respons;",";Stimulas;",";Error_term_data(*)
4790 !
4800 CASE "READ"
4810 FOR Ii=1 TO Nop
4820 Error_term_data(2*Ii-1)=-999
4830 Error_term_data(2*Ii)=-999
4840 NEXT Ii
4850 !
4860 OUTPUT @Agte507x;":SENS"&Ch$&":CORR:COEF? "&Id$&",";Respons;",";Stimulas
4870 ENTER @Agte507x;Error_term_data(*)
4880 !
4890 CALL Data_plot(Id$,Respons,Stimulas,Nop,Error_term_data(*))
4900 !
4910 FOR Ii=1 TO Nop
```

```
4920 Stok(Idx,2*Ii-1)=Error_term_data(2*Ii-1)
4930 Stok(Idx,2*Ii)=Error_term_data(2*Ii)
4940 NEXT Ii
4950 !
4960 END SELECT
4970 SUBEND
4980!
4990 Data_plot: SUB Data_plot(Error_term$,INTEGER Respons,Stimulas,Nop,REAL
Error_term_data(*))
5000 INTEGER Ii,Pen(1:2)
5010 REAL Y_minmax(1:2)
5020 DIM Wk$[20]
5030 !
5040 CLEAR SCREEN
5050 GINIT
5060 GCLEAR
5070 !
5080 Pen(1)=3
5090 Pen(2)=4
5100 !
5110 ! Get Min Value and Max Value from all data
5120 Y_minmax(1)=MIN(Error_term_data(*))
5130 Y_minmax(2)=MAX(Error_term_data(*))
5150 !
5160 IF (Y_minmax(1)=Y_minmax(2)) AND (Y_minmax(1)=0) THEN
5170 Y_minmax(1)=1
5180 Y_minmax(2)=-1
5190 ELSE
5200 IF (Y_minmax(1)=Y_minmax(2)) THEN
5210 Y_minmax(1)=Y_minmax(1)*.5
5220 Y_minmax(2)=Y_minmax(2)*1.5
5230 END IF
5240 END IF
5250 !
5260 VIEWPORT 25*RATIO,80*RATIO,40,90
5270 WINDOW 1,Nop,Y_minmax(1),Y_minmax(2)
5280 FRAME
```

```
5290 !
5300 VIEWPORT 80*RATIO,100*RATIO,40,90
5310 WINDOW 0,2,0,2
5320 PEN Pen(1)
5330 CSIZE 2.5
5340 LORG 2
5350 MOVE .2,1.5
5360 DRAW .4,1.5
5370 MOVE .5,1.5
5380 PEN 1
5390 LABEL ":Real Value"
5400 !
5410 PEN Pen(2)
5420 MOVE .2,1
5430 DRAW .4,1
5440 MOVE .5,1
5450 PEN 1
5460 LABEL ":Image Value"
5470 !
5480 VIEWPORT 25*RATIO,80*RATIO,90,100
5490 WINDOW 0,2,0,2
5500 CSIZE 3
5510 LORG 5
5520 MOVE 1,1.2
5530 LABEL "Error Term:"&Error_term$
5540 !
5550 MOVE 1,.5
5560 LABEL "Respons Port:"&VAL$(Respons)&" Stimulas Port:"&VAL$(Stimulas)
5570 !
5580 VIEWPORT 0,25*RATIO,40,90
5590 WINDOW 0,2,0,2
5600 CLIP -10,10,-10,10
5610 LORG 8
5620 CSIZE 3
5630 !
5640 MOVE 1.9,0
5650 LABEL VAL$(Y_minmax(1))
```

```
5660 MOVE 1.9,2
5670 LABEL VAL$(Y_minmax(2))
5680 !
5690 VIEWPORT 25*RATIO,80*RATIO,30,40
5700 WINDOW 0,2,0,2
5710 CLIP -10,10,-10,10
5720 LORG 5
5730 MOVE 0,1.5
5740 LABEL VAL$(1)
5750 MOVE 2,1.5
5760 LABEL VAL$(Nop)
5770 !
5780 VIEWPORT 25*RATIO,80*RATIO,40,90
5790 WINDOW 1,Nop,Y_minmax(1),Y_minmax(2)
5800 FOR Ii=2 TO Nop
5820 PEN Pen(1)
5830 MOVE Ii-1,Error_term_data(2*(Ii-1)-1)
5840 DRAW Ii,Error_term_data(2*Ii-1)
5860 !
5870 PEN Pen(2)
5880 MOVE Ii-1,Error_term_data(2*(Ii-1))
5890 DRAW Ii,Error_term_data(2*Ii)
5900 NEXT Ii
5910 !
5920 PEN 1
5930 BEEP
5940 INPUT "Cont:push [Enter] key",Wk$
5950 SUBEND
5960!
```

Waiting for Trigger (OPC?)

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use the :TRIG:SING command to wait until the measurement cycle is completed.

- This sample program correctly runs when the maximum number of channels/traces is set to 9 channels/9 traces.

The sample program uses the :TRIG:SING command to start a sweep (measurement) cycle, uses the *OPC command to wait until the measurement cycle is completed, then prints a message and exits.

See Waiting for the End of Measurement for this programming.

Sample Program in Excel VBA

```
Sub trg_sing_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim ContMode(9) As String
    Dim Result As String * 10
    Dim i As Integer
    Const TimeOutTime = 100000 ' TimeOut time should be greater than the measurement time.
    '
    ' Assign a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Store the settings of continuous initiation mode for eachchannel
    ' (on for channels 1 and 2; off for channels 3 through 9)
    ' into the array variable ContMode().
    ContMode(1) = "ON"
    ContMode(2) = "ON"
    For i = 3 To 9
        ContMode(i) = "OFF"
    Next i
```

```
    '
    ' Turn on or off continuous initiation mode for each channel
    ' depending on the value of ContMode(*).
    For i = 1 To 9
       Call viVPrintf(vi, ":INIT" & CStr(i) & ":CONT " & ContMode(i) & vbLf, 0)
    Next i
    ' Set the trigger source to Bus Trigger.
    Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)
    '
    ' Trigger the instrument to start a sweep cycle.
    Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)
    '
    ' Execute the *OPC? command and wait until the command
    ' returns 1 (i.e., the measurement cycle is completed).
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    '
    ' Display a measurement completion message.
    Stat = MsgBox("Measurement complete", vbOKOnly)
    Call viClose(vi)
    Call viClose(defrm)
End Sub
```

Sample Program in HT Basic (trg_sing.htb)

```
10 DIM Cont_mode$(1:9)[9],Buff$[9]
20 INTEGER I
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 Cont_mode$(1)="ON"
70 Cont_mode$(2)="ON"
80 Cont_mode$(3)="OFF"
90 Cont_mode$(4)="OFF"
100 Cont_mode$(5)="OFF"
110 Cont_mode$(6)="OFF"
120 Cont_mode$(7)="OFF"
130 Cont_mode$(8)="OFF"
140 Cont_mode$(9)="OFF"
```

```
150 !
160 FOR I=1 TO 9
170 OUTPUT @Agte507x;":INIT"&VAL$(I)&":CONT "&Cont_mode$(I)
180 NEXT I
190 OUTPUT @Agte507x;":TRIG:SOUR BUS"
200 !
210 OUTPUT @Agte507x;":TRIG:SING"
220 OUTPUT @Agte507x;"*OPC?"
230 ENTER @Agte507x;Buff$
240 !
250 PRINT "Measurement complete"
260 END
```

Waiting for Trigger (SRQ)

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use an SRQ to detect the end of measurement.

- This sample program correctly runs when the maximum number of channels/traces is set to 9 channels/9 traces.

The sample program sets up the trigger system, configures the instrument to properly generate an SRQ, and then triggers the instrument. When the instrument has generated an SRQ that indicates the end of measurement, the program exits after printing a measurement completion message.

See Waiting for the End of Measurement for this programming.

Sample Program in Excel VBA

```
Sub srq_meas_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim ContMode(9) As String
    Dim Result As String * 10
    Dim i As Integer, StbStatus As Integer
    Const TimeOutTime = 100000 ' TimeOut time should be greater than the measurement time.
    '
    ' Assign a GPIB address to the I/O pass.
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Store the settings of continuous initiation mode for eachchannel
    ' (on for channels 1 and 2; off for channels 3 through 9)
    ' into the array variable ContMode().
    ContMode(1) = "ON"
    ContMode(2) = "ON"
    For i = 3 To 9
      ContMode(i) = "OFF"
    Next i
```

```
    '
    ' Turn on or off continuous initiation mode for each channel
    ' depending on the value of ContMode(*).
    For i = 1 To 9
        Call viVPrintf(vi, ":INIT" & CStr(i) & ":CONT " & ContMode(i) & vbLf, 0)
    Next i
    '
    ' Set the trigger source to Bus Trigger.
    Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)
    '
    ' .
    Call viVPrintf(vi, ":STAT:OPER:PTR 0" & vbLf, 0) 'Set 0 at all bits of Position Transition Filter
    Call viVPrintf(vi, ":STAT:OPER:NTR 16" & vbLf, 0) 'Set 1 at bit 4 of Negative Transition Filter
    Call viVPrintf(vi, ":STAT:OPER:ENAB 16" & vbLf, 0) 'Set 1 at bit 4 of Operation status enable
    Call viVPrintf(vi, "*SRE 128" & vbLf, 0) 'Set 1 at bit 7 of Service Request Enable Register
    Call viVPrintf(vi, "*CLS" & vbLf, 0) ' Clear Register.
    '
    Call viVPrintf(vi, "*TRG" & vbLf, 0) 'Make a trigger
    ' Wait until Status Byte Register became 192
    Do
        Call viReadSTB(vi, StbStatus) ' Read Status Byte Register
        Range("B5").Value = StbStatus
    Loop Until StbStatus = 192
    '
    ' Display a measurement completion message.
    Stat = MsgBox("Measurement complete", vbOKOnly)
    ' Close IO
    Call viClose(vi)
    Call viClose(defrm)
End Sub
```

Sample Program in HT Basic (srq_meas.htb)

```
10 DIM Cont_mode$(1:9)[9],Buff$[9]
20 INTEGER I
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 Cont_mode$(1)="ON"
```

```
70 Cont_mode$(2)="ON"
80 Cont_mode$(3)="OFF"
90 Cont_mode$(4)="OFF"
100 Cont_mode$(5)="OFF"
110 Cont_mode$(6)="OFF"
120 Cont_mode$(7)="OFF"
130 Cont_mode$(8)="OFF"
140 Cont_mode$(9)="OFF"
150 !
160 FOR I=1 TO 9
170 OUTPUT @Agte507x;":INIT"&VAL$(I)&":CONT "&Cont_mode$(I)
180 NEXT I
190 OUTPUT @Agte507x;":TRIG:SOUR BUS"
200 !
210 OUTPUT @Agte507x;":STAT:OPER:PTR 0"
220 OUTPUT @Agte507x;":STAT:OPER:NTR 16"
230 OUTPUT @Agte507x;":STAT:OPER:ENAB 16"
240 OUTPUT @Agte507x;"*SRE 128"
250 OUTPUT @Agte507x;"*CLS"
260 OUTPUT @Agte507x;"*OPC?"
270 ENTER @Agte507x;Buff$
280 !
290 ON INTR 7 GOTO Meas_end
300 ENABLE INTR 7;2
310 OUTPUT @Agte507x;"*TRG"
320 PRINT "Waiting..."
330 Meas_wait: GOTO Meas_wait
340 Meas_end: OFF INTR 7
350 PRINT "Measurement Complete"
360 END
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 140

These lines store the settings of continuous initiation mode for each channel (on for channels 1 and 2; off for channels 3 through 9) into the array variable Cont_mode$(*).

Lines 160 to 180

These lines turn on or off continuous initiation mode for each channel depending on the value of Cont_mode$(*).

Line 190

Sets the trigger source to "Bus Trigger".

Lines 210 to 220

These lines configure the instrument so that operation status event register's bit 4 is set to 1 only when the operation status condition register's bit 4 is changed from 1 to 0 (negative transition).

Lines 230 to 240

These lines enable the operation status event register's bit 4 and status byte register's bit 7.

Lines 250 to 270

These lines clear the status byte register and operation status event register.

Lines 290 to 300

These lines set the branch target for an SRQ interrupt to enable SRQ interruptions.

Lines 310 to 320

These lines trigger the instrument and wait until the measurement cycle finishes.

Line 350

Displays a measurement completion message.

Error Detection (SRQ)

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use an SRQ to detect the occurrence of an error.

This program sets SRQs and then intentionally sends an invalid parameter to generate an error to be handled by this program. In the error handling part, this program examines the error, displays the error number and error message, and then displays the message indicating the suspension of the program. See Detecting Occurrence of an Error for this programming.

- The sequence interception by an error can not be performed on Excel VBA.

Sample Program in HT Basic (srq_err.bas)

```
10 DIM Buff$[9],Err_mes$[50]
20 INTEGER Err_no
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 OUTPUT @Agte507x;"*ESE 60"
70 OUTPUT @Agte507x;"*SRE 32"
80 OUTPUT @Agte507x;"*CLS"
90 OUTPUT @Agte507x;"*OPC?"
100 ENTER @Agte507x;Buff$
110 !
120 ON INTR 7 GOTO Err_proc
130 ENABLE INTR 7;2
140 OUTPUT @Agte507x;":CALC1:PAR:COUN 2"
150 PRINT "Trace 1 Meas.Para: S21"
160 PRINT "Trace 1 Format : Log Mag"
170 OUTPUT @Agte507x;":CALC1:PAR1:DEF S21"
180 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
190 OUTPUT @Agte507x;":CALC1:FORM MLOG"
200 PRINT "Trace 2 Meas.Para: S11"
210 PRINT "Trace 2 Format : Log Mag"
220 OUTPUT @Agte507x;":CALC1:PAR2:DEF S11"
```

```
230 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
240 OUTPUT @Agte507x;":CALC1:FORM LOG"
250 OUTPUT @Agte507x;"*OPC?"
260 ENTER @Agte507x;Buff$
270 GOTO Skip_err_proc
280 Err_proc: OFF INTR 7
290 OUTPUT @Agte507x;";:SYST:ERR?"
300 ENTER @Agte507x;Err_no,Err_mes$
310 PRINT "Error occurred!!"
320 PRINT " No:";Err_no,"Description: "&Err_mes$
330 PRINT "PROGRAM INTERRUPT!!"
340 GOTO Prog_end
350 Skip_err_proc: PRINT "PROGRAM DONE."
360 Prog_end: END
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 70

These lines enable bits 2, 3, 4 and 5 in the standard event status register and set bit 5 to 1 in the service request enable register.

Lines 80 to 100

These lines clear the status byte register, the standard event status register, and the error queue.

Lines 120 to 130

These lines set the branch target for an SRQ interrupt to enable SRQ interruptions.

Lines 140 to 260

These lines set the measurement parameters and their data formats for traces 1 and 2. An invalid parameter is given to the data format setting for trace 2, causing an error.

Lines 280 to 330

These lines define an error handler in the following way.

Lines 290 to 300: These lines retrieve the error number and error messages for the error from the error queue.

Lines 310 to 330 These lines display the message indicating the occurrence of the error, the error number, the error message, and the message showing that the program is suspended.

Line 350

Displays a closing message. Note that this message will not display unless this program is re-executed after setting a corrected parameter to the data format setting for trace 2.

Reading Data in ASCII Format

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to retrieve formatted data arrays in the ASCII transfer format.

This program holds the sweep on channel 1, then retrieves and displays the stimulus array for channel 1 and the formatted data array for trace 1.

See Retrieving Measurement Results for this programming.

Sample Program in Excel VBA

```
Sub read_asc_Click()
    '
    Dim defrm As Long
    Dim vi As Long
    Dim Result As String * 10000
    Dim Res As Variant
    Dim Res2 As Variant
    Dim Nop As Long
    Const TimeOutTime = 10000
    '
    ' Open the Analyzer
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    ' Select Parameter 1
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" + vbLf, 0)
    Call viVPrintf(vi, ":INIT1:CONT OFF" + vbLf, 0)
    Call viVPrintf(vi, ":ABOR" + vbLf, 0)
    '
    ' Read out NOP Data in ASCII transfer format
    Call viVPrintf(vi, ":SENS1:SWE:POIN?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Nop = Val(Result)
```

```
    ReDim FMTData(Nop, 2)
    ReDim Freq(Nop)
    '
    ' Read out Measurement Data in ASCII transfer format
    Call viVPrintf(vi, ":FORM:DATA ASC" + vbLf, 0)
    Result = ""
    Call viVPrintf(vi, ":CALC1:DATA:FDAT?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Res = Split(Result, ",")
    '
    Range("A6:D1607").Clear 'Clear cells of Excel
    '
    ' Write data in cells of Excel
    j = 0
    For i = 1 To Nop
       Cells(i + 5, 1) = i
       Cells(i + 5, 3) = Val(Res(j))
       Cells(i + 5, 4) = Val(Res(j + 1))
       j = j + 2
    Next i
    '
    ' Read out Measurement Frequency Data in ASCII transfer format
    Result = ""
    Call viVPrintf(vi, ":SENS1:FREQ:DATA?" + vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Res2 = Split(Result, ",")
    '
    '  Write data in cells of Excel
    For i = 1 To Nop
       Cells(i + 5, 2) = Val(Res2(i - 1))
    Next i
    '
    ' Close the Analyzer
    Call viClose(vi)
    Call viClose(defrm)
End Sub
```

Sample Program in HT Basic (read_asc.htb)

```
10 REAL Fdata(1:1601,1:2),Freq(1:1601)
20 DIM Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
80 OUTPUT @Agte507x;":INIT1:CONT OFF"
90 OUTPUT @Agte507x;":ABOR"
100 OUTPUT @Agte507x;":SENS1:SWE:POIN?"
110 ENTER @Agte507x;Nop
120 REDIM Fdata(1:Nop,1:2),Freq(1:Nop)
130 !
140 ! Reading out in ASCII transfer format
150 !
160 OUTPUT @Agte507x;":FORM:DATA ASC"
170 !
180 OUTPUT @Agte507x;":CALC1:DATA:FDAT?"
190 ENTER @Agte507x;Fdata(*)
200 OUTPUT @Agte507x;":SENS1:FREQ:DATA?"
210 ENTER @Agte507x;Freq(*)
220 !
230 ! Displaying
240 !
250 OUTPUT @Agte507x;":CALC1:FORM?"
260 ENTER @Agte507x;Fmt$
270 SELECT Fmt$
280 CASE "MLOG","PHAS","GDEL","MLIN","SWR","REAL","IMAG","UPH"
290 Img$="MD.4DE,2X,MD.6DE"
300 PRINT " Frequency Data"
310 FOR I=1 TO Nop
320 PRINT USING Img$;Freq(I),Fdata(I,1)
330 NEXT I
340 CASE ELSE
350 Img$="MD.4DE,2X,MD.6DE,2X,MD.6DE"
360 PRINT " Frequency Data1 Data2"
370 FOR I=1 TO Nop
```

```
380 PRINT USING Img$;Freq(I),Fdata(I,1),Fdata(I,2)
390 NEXT I
400 END SELECT
410 !
420 END
```

Reading Data in Binary Format

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to retrieve formatted data arrays in the Binary transfer format.

This program holds the sweep on channel 1, then retrieves and displays the stimulus array.

See Retrieving Measurement Results for this programming.

Sample Program in Excel VBA

```
Sub read_bin_Click()
    Dim defrm As Long
    Dim vi As Long
    Dim Result As String * 10000
    Dim Res() As Double
    Dim Nop As Long
    Const TimeOutTime = 10000
    ' Open Analyzer
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    '
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" + vbLf, 0)
    Call viVPrintf(vi, ":INIT1:CONT OFF" + vbLf, 0)
    Call viVPrintf(vi, ":ABOR" + vbLf, 0)
    '
    ' Reading out Measurement Frequency Data in Binary transfer format
    Call viVPrintf(vi, ":FORM:DATA REAL" + vbLf, 0)
    Call viVPrintf(vi, ":CALC1:DATA:FDAT?" + vbLf, 0)
    Call Scpi_read_binary_double_array(vi, Res, Nop)
    '
    ' Write data in cells of Excel
    Range("A6:D1607").Clear
    For i = 0 To Nop - 1
        j = i Mod 2
```

```
        k = i \ 2
        Cells(k + 6, j + 3).Value = Res(i)
    Next i
    '
    ' Read out Measurement Frequency Data in Binary transfer format
    Call viVPrintf(vi, ":SENS1:FREQ:DATA?" + vbLf, 0)
    Call Scpi_read_binary_double_array(vi, Res, Nop)
    '
    ' Write data in cells of Excel
    For i = 0 To Nop - 1
        Cells(i + 6, 1) = i + 1
        Cells(i + 6, 2).Value = Res(i)
    Next i
    ' Close
    Call viClose(vi)
    Call viClose(defrm)
End Sub
    '=================================
    ' BinaryAry Read Subroutine
    '=================================
Sub Scpi_read_binary_double_array(vi As Long, data() As Double, Nop As Long)
    Dim dblArray(10000) As Double
    Dim paramsArray(3) As Long
    Dim err As Long
    Dim i As Long
    Dim lf_eoi As String * 1

    Nop = UBound(dblArray) - LBound(dblArray) + 1
    paramsArray(0) = VarPtr(Nop)
    paramsArray(1) = VarPtr(dblArray(0))
    err = viVScanf(vi, "%#Zb%1t", paramsArray(0))
    If err <> 0 Then MsgBox "Binary Error"

    ReDim data(Nop - 1)
    For i = 0 To Nop - 1
        data(i) = dblArray(i)
    Next
```

End Sub

Sample Program in HT Basic (read_bin.htb)

```
10 REAL Fdata(1:1601,1:2),Freq(1:1601)
20 DIM Buff$[9],Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte507x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 !
80 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
90 OUTPUT @Agte507x;":INIT1:CONT OFF"
100 OUTPUT @Agte507x;":ABOR"
110 OUTPUT @Agte507x;":SENS1:SWE:POIN?"
120 ENTER @Agte507x;Nop
130 REDIM Fdata(1:Nop,1:2),Freq(1:Nop)
140 !
150 ! Reading out in binary transfer format
160 !
170 OUTPUT @Agte507x;":FORM:DATA REAL"
180 !
190 OUTPUT @Agte507x;":CALC1:DATA:FDAT?"
200 ENTER @Agte507x USING "#,8A";Buff$
210 ENTER @Binary;Fdata(*)
220 ENTER @Agte507x USING "#,1A";Buff$
230 OUTPUT @Agte507x;":SENS1:FREQ:DATA?"
240 ENTER @Agte507x USING "#,8A";Buff$
250 ENTER @Binary;Freq(*)
260 ENTER @Agte507x USING "#,1A";Buff$
270 !
280 ! Displaying
290 !
300 OUTPUT @Agte507x;":CALC1:FORM?"
310 ENTER @Agte507x;Fmt$
320 SELECT Fmt$
330 CASE "MLOG","PHAS","GDEL","MLIN","SWR","REAL","IMAG","UPH"
340 Img$="MD.4DE,2X,MD.6DE"
350 PRINT " Frequency Data"
```

```
360 FOR I=1 TO Nop
370 PRINT USING Img$;Freq(I),Fdata(I,1)
380 NEXT I
390 CASE ELSE
400 Img$="MD.4DE,2X,MD.6DE,2X,MD.6DE"
410 PRINT " Frequency Data1 Data2"
420 FOR I=1 TO Nop
430 PRINT USING Img$;Freq(I),Fdata(I,1),Fdata(I,2)
440 NEXT I
450 END SELECT
460 !
470 END
```

Writing Data in Ascii Format

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates to Write Formatted Data Arrays in Using the ASCII Transfer Format

See Entering Data into a Trace for this programming.

Sample Program in HT Basic (write_a.htb)

```
10 REAL Freq,Fdata(1:1601,1:2)
20 DIM File$[300]
30 INTEGER Nop
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 CALL Inp_file_name(File$)
80 !
90 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
100 OUTPUT @Agte507x;":INIT1:CONT OFF"
110 OUTPUT @Agte507x;":ABOR"
120 !
130 OUTPUT @Agte507x;":SENS1:SWE:POIN?"
140 ENTER @Agte507x;Nop
150 REDIM Fdata(1:Nop,1:2)
160 !
170 ON ERROR GOTO File_error
180 ASSIGN @File TO File$
190 ENTER @File USING "K";Buff$
200 ENTER @File USING "K";Buff$
210 ENTER @File USING "K";Buff$
220 FOR I=1 TO Nop
230 ENTER @File USING "19D,2X,19D,2X,19D";Freq,Fdata(I,1),Fdata (I,2)
240 NEXT I
250 ASSIGN @File TO *
260 OFF ERROR
270 !
280 OUTPUT @Agte507x;":FORM:DATA ASC"
```

```
290 !
300 OUTPUT @Agte507x;":CALC1:DATA:FDAT ";Fdata(*)
310 !
320 GOTO Prog_end
330 !
340 File_error: OFF ERROR
350 PRINT "############ ERROR ############"
360 PRINT File$&" is NOT exist."
370 PRINT " or"
380 PRINT File$&" has UNSUITABLE data."
390 !
400 Prog_end: END
410 !=========================================
420 ! File Name Input Function
430 !=========================================
440 SUB Inp_file_name(Inp_name$)
450 DIM Inp_char$[9]
460 ON ERROR GOTO Inp_start
470 Inp_start: !
480 PRINT "Input File Name!"
490 INPUT "Name?",Inp_name$
500 PRINT "Input Name: "&Inp_name$
510 INPUT "OK? [Y/N]",Inp_char$
520 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
530 OFF ERROR
540 SUBEND
```

Description

Line 50

Assigns a GPIB address to the I/O pass.

Line 70

Passes control to a subprogram named Inp_file_name, which lets the user input a file name, and then stores the returned file name into the File$ variable. For more information on the Inp_file_name subprogram, refer to the description in Using the Binary Transfer Format to write Formatted Data Arrays.

Lines 90 to 110

These lines set channel 1's active trace to trace 1 and hold the sweep.

Lines 130 to 140

These lines retrieve the number of points in channel 1 and stores that number into the Nop variable.

Line 150

Resizes the Fdata array based on the value of the Nop variable (the number of points).

Line 170

This line points to the statement block to be executed if an error occurs in retrieving data from the file (for example, if no file matches File$).

Lines 180 to 260

These lines retrieve the formatted data from the file identified by File$, and store the data into the Fdata array.

Line 280

Sets the data transfer format to ASCII.

Line 300

Writes Fdata into the formatted data array for the active trace (trace 1) in channel 1.

Lines 340 to 380

This statement block is executed if an error occurs in retrieving data from the file.

Writing Data in Binary Format

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates to Write Formatted Data Arrays in Using the Binary Transfer Format

See Entering Data into a Trace for this programming.

Sample Program in HT Basic (write_b.htb)

```
10 REAL Freq,Fdata(1:1601,1:2)
20 DIM File$[300],Header$[10]
30 INTEGER Nop
40 !
50 ASSIGN @Agte507x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 CALL Inp_file_name(File$)
80 !
90 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
100 OUTPUT @Agte507x;":INIT1:CONT OFF"
110 OUTPUT @Agte507x;":ABOR"
120 !
130 OUTPUT @Agte507x;":SENS1:SWE:POIN?"
140 ENTER @Agte507x;Nop
150 REDIM Fdata(1:Nop,1:2)
160 !
170 ON ERROR GOTO File_error
180 ASSIGN @File TO File$
190 ENTER @File USING "K";Buff$
200 ENTER @File USING "K";Buff$
210 ENTER @File USING "K";Buff$
220 FOR I=1 TO Nop
230 ENTER @File USING "19D,2X,19D,2X,19D";Freq,Fdata(I,1),Fdata (I,2)
240 NEXT I
250 ASSIGN @File TO *
260 OFF ERROR
270 !
280 OUTPUT @Agte507x;":FORM:DATA REAL"
```

```
290 Header$="#6"&IVAL$(8*2*Nop,10)
300 OUTPUT @Agte507x;":CALC1:DATA:FDAT ";Header$;
310 OUTPUT @Binary;Fdata(*),END
320 GOTO Prog_end
330 !
340 File_error: OFF ERROR
350 PRINT "############ ERROR ############"
360 PRINT File$&" is NOT exist."
370 PRINT " or"
380 PRINT File$&" has UNSUITABLE data."
390 !
400 Prog_end: END
410 !=========================================
420 ! File Name Input Function
430 !=========================================
440 SUB Inp_file_name(Inp_name$)
450 DIM Inp_char$[9]
460 ON ERROR GOTO Inp_start
470 Inp_start: !
480 PRINT "Input File Name!"
490 INPUT "Name?",Inp_name$
500 PRINT "Input Name: "&Inp_name$
510 INPUT "OK? [Y/N]",Inp_char$
520 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
530 OFF ERROR
540 SUBEND
```

Description

Lines 50 to 60

Assigns a GPIB address to the I/O pass.

Line 70

Passes control to a subprogram named Inp_file_name, which lets the user input a file name, and then stores the returned file name into the File$ variable.

Lines 90 to 110

These lines set channel 1's active trace to trace 1 and hold the sweep.

Lines 130 to 140

These lines retrieve the number of points in channel 1 and stores that number into the Nop variable.

Line 150

Resizes the Fdata array based on the value of the Nop variable (the number of points).

Line 170

This line points to the statement block to be executed if an error occurs in retrieving data from the file (for example, if no file matches File$).

Lines 180 to 260

These lines retrieve the formatted data from the file identified by File$, and store the data into the Fdata array.

Line 280

Sets the data transfer format to binary.

Line 290

Creates the data header and stores it into the Header$ variable.

Line 300

Sends the command that writes data into the formatted data array for the active trace (trace 1) in channel 1, following it with the data header (Header$).

Line 310

Sends the data itself (Fdata), following it with a message terminator.

- Because binary data must be written without being formatted, the program uses an I/O path (@Binary) that is configured to support writing unformatted data.

Lines 340 to 380

This statement block is executed if an error occurs in retrieving data from the file.

The Inp_file_name subprogram in lines 440 to 540, which is used to enter a save filename, is described below.

Line 460

Allows the user to return to the entry start line and re-enter the data if an error (such as an invalid entry) occurs while entering the target file name.

Lines 480 to 490

These lines prompt the user to enter the target file name. The program does not continue till the user actually enters the file name.

Lines 500 to 510

These lines display the entered file name and waits for a confirmation entry (y/n key).

Line 520

Returns to the entry start line if the key the user pressed in line 870 is not the y key.

Peak Search

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to search for peaks using the Marker Search feature and analysis commands.

This program works in two steps: first, it uses Marker Search to search for the maximum positive peak and displays the results; second, it uses analysis commands to search for all positive peaks and displays the results.

See Searching for Positions That Match Specified Criteria for this programming.

Sample Program in Excel VBA

```
Sub PeakSearch_Click()
    Dim defrm As Long
    Dim vi As Long
    Const TimeOutTime = 20000
    '
    Dim Buff As String, Img As String, Err_msg As String
    Dim Excursion As String, Freq As String * 20, Resp As Variant, PeakPoint As Variant
    Dim Poin As String * 5, Result As String * 1000, errmsg As String * 20

    Excursion = "0.5"
    ' Open Analyzer
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    Call viVPrintf(vi, "*CLS" & vbLf, 0)
    '
    ' Setup Analyzer
    Call viVPrintf(vi, ":SENS1:FREQ:CENT 947.5E6" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:FREQ:SPAN 200E6" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:DEF S11" & vbLf, 0)
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)
    '
```

```
Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:TYPE PEAK" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:PEXC " & Excursion & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:PPOL POS" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:MARK1:FUNC:EXEC" & vbLf, 0)
Call ErrorCheck(vi)
Call viVPrintf(vi, ":CALC1:MARK1:X?" & vbLf, 0)
Call viVScanf(vi, "%t", Freq)
'
Call viVPrintf(vi, ":CALC1:MARK1:Y?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
'
Resp = Split(Result, ",")
Cells(5, 5).Value = Val(Freq)
Cells(5, 6).Value = Val(Resp(0))
'
Call viVPrintf(vi, ":CALC1:FUNC:DOM OFF" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:TYPE APE" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:PEXC " & Excursion & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:PPOL NEG" & vbLf, 0)
Call viVPrintf(vi, ":CALC1:FUNC:EXEC" & vbLf, 0)
Call ErrorCheck(vi)
'
Call viVPrintf(vi, ":CALC1:FUNC:POIN?" & vbLf, 0)
Call viVScanf(vi, "%t", Poin)
Call viVPrintf(vi, ":CALC1:FUNC:DATA?" & vbLf, 0)
Call viVScanf(vi, "%t", Result)
PeakPoint = Split(Result, ",")
'
j = 1
   For i = 1 To Val(Poin) / 2
   Cells(6 + i, 5).Value = Val(PeakPoint(j))
   Cells(6 + i, 6).Value = Val(PeakPoint(j + 1))
   j = j + 2
Next i
'
Call viClose(vi)
```

```
    Call viClose(defrm)
    '
End Sub
Sub ErrorCheck(vi)
    Dim err As String * 50, ErrNo As Variant, Response As VbMsgBoxResult
    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
    ErrNo = Split(err, ",")
    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
    End If
End Sub
```

Sample Program in HT Basic (search.htb)

```
10 DIM Buff$[9],Img$[50],Err_msg$[100]
20 REAL Excursion,Freq,Resp,Result(1:100,1:2)
30 INTEGER Poin,Err_no
40 !
50 ASSIGN @Agte507x TO 717
60 Excursion=.5
70 !
80 OUTPUT @Agte507x;"*ESE 60"
90 OUTPUT @Agte507x;"*SRE 32"
100 OUTPUT @Agte507x;"*CLS"
110 OUTPUT @Agte507x;"*OPC?"
120 ENTER @Agte507x;Buff$
130 ON INTR 7 GOTO Err
140 ENABLE INTR 7;2
150 !
160 PRINT "Maximum Peak Search using Marker 1"
170 !
180 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
190 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:TYPE PEAK"
200 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:PEXC ";Excursion
210 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:PPOL POS"
220 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:EXEC"
230 OUTPUT @Agte507x;":CALC1:MARK1:X?"
240 ENTER @Agte507x;Freq
250 OUTPUT @Agte507x;":CALC1:MARK1:Y?"
```

```
260 ENTER @Agte507x;Resp
270 Img$="8A,MD.4DE,2X,MD.6DE"
280 PRINT " Frequency Response"
290 PRINT USING Img$;"Peak: ",Freq,Resp
300 !
310 PRINT "All Peaks Search using Command"
320 !
330 OUTPUT @Agte507x;":CALC1:FUNC:DOM OFF"
340 OUTPUT @Agte507x;":CALC1:FUNC:TYPE APE"
350 OUTPUT @Agte507x;":CALC1:FUNC:PEXC ";Excursion
360 OUTPUT @Agte507x;":CALC1:FUNC:PPOL POS"
370 OUTPUT @Agte507x;":CALC1:FUNC:EXEC"
380 OUTPUT @Agte507x;":CALC1:FUNC:POIN?"
390 ENTER @Agte507x;Poin
400 REDIM Result(1:Poin,1:2)
410 OUTPUT @Agte507x;":CALC1:FUNC:DATA?"
420 ENTER @Agte507x;Result(*)
430 Img$="4A,2D,2A,MD.4DE,2X,MD.6DE"
440 PRINT " Frequency Response"
450 FOR I=1 TO Poin
460 PRINT USING Img$;"Peak",I,": ",Result(I,2),Result(I,1)
470 NEXT I
480 GOTO No_err
490 Err: OFF INTR 7
500 OUTPUT @Agte507x;";:SYST:ERR?"
510 ENTER @Agte507x;Err_no,Err_msg$
520 PRINT "Error occurred!!"
530 PRINT " No:";Err_no,"Description: "&Err_msg$
540 No_err: OFF INTR 7
550 END
```

Limit Test

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to perform limit tests.

The sample program creates a limit table as shown in the following two tables, turns on the Limit Test feature, performs one cycle of measurement, and then displays the test results.

No.	Type	Begin Stimulus	End Stimulus	Begin Response	End Response
1	MAX	847.5 MHz	905.0 MHz	-55.0 dBm	-55.0 dBm
2	MIN	935.0 MHz	960.0 MHz	-3.5 dBm	-3.5 dBm
3	MAX	935.0 MHz	960.0 MHz	0 dBm	0 dBm
4	MAX	980.0 MHz	1047.5 MHz	-25.0 dBm	-25.0 dBm

See Limit Test for this programming.

Sample Program in HT Basic (lim_test.htb)

```
10 DIM Param1$[9],Param2$[9],Fmt1$[9],Fmt2$[9],Buff$[9]
20 REAL Cent,Span,Lim1(1:4,1:5),Lim2(1:3,1:5),Fail_data(1:1601)
30 INTEGER Num_of_seg1,Num_of_seg2,Segment,Column,Fail_point
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 Cent=9.475E+8
80 Span=2.00E+8
90 Param1$="S21"
100 Param2$="S11"
110 Fmt1$="MLOG"
120 Fmt2$="MLOG"
130 !
140 ! == Trace 1 Limit Line ==
```

```
150 Num_of_seg1=4 ! Number of segments: 4
160 ! -- Segment 1 --
170 Lim1(1,1)=1 ! Type : Maximum
180 Lim1(1,2)=8.475E+8 ! Frequency Start: 847.5 MHz
190 Lim1(1,3)=9.050E+8 ! Stop : 905.0 MHz
200 Lim1(1,4)=-55 ! Response Start: -55 dBm
210 Lim1(1,5)=-55 ! Stop : -55 dBm
220 ! -- Segment 2 --
230 Lim1(2,1)=2 ! Type : Minimum
240 Lim1(2,2)=9.350E+8 ! Frequency Start: 935.0 MHz
250 Lim1(2,3)=9.600E+8 ! Stop : 960.0 MHz
260 Lim1(2,4)=-3.5 ! Response Start: -3.5 dBm
270 Lim1(2,5)=-3.5 ! Stop : -3.5 dBm
280 ! -- Segment 3 --
290 Lim1(3,1)=1 ! Type : Maximum
300 Lim1(3,2)=9.350E+8 ! Frequency Start: 935.0 MHz
310 Lim1(3,3)=9.600E+8 ! Stop : 960.0 MHz
320 Lim1(3,4)=0 ! Response Start: 0 dBm
330 Lim1(3,5)=0 ! Stop : 0 dBm
340 ! -- Segment 4 --
350 Lim1(4,1)=1 ! Type : Maximum
360 Lim1(4,2)=9.800E+8 ! Frequency Start: 980.0 MHz
370 Lim1(4,3)=1.0475E+9 ! Stop : 1047.5 MHz
380 Lim1(4,4)=-25 ! Response Start: -25 dBm
390 Lim1(4,5)=-25 ! Stop : -25 dBm
400 ! == Trace 2 Limit Line ==
410 Num_of_seg2=3 ! Number of segments: 3
420 ! -- Segment 1 --
430 Lim2(1,1)=1 ! Type : Maximum
440 Lim2(1,2)=8.475E+8 ! Frequency Start: 847.5 MHz
450 Lim2(1,3)=9.250E+8 ! Stop : 925.0 MHz
460 Lim2(1,4)=0 ! Response Start: 0 dBm
470 Lim2(1,5)=0 ! Stop : 0 dBm
480 ! -- Segment 2 --
490 Lim2(2,1)=1 ! Type : Maximum
500 Lim2(2,2)=9.350E+8 ! Frequency Start: 935.0 MHz
510 Lim2(2,3)=9.600E+8 ! Stop : 960.0 MHz
```

```
520 Lim2(2,4)=-9.5 ! Response Start: -9.5 dBm
530 Lim2(2,5)=-9.5 ! Stop : -9.5 dBm
540 ! -- Segment 3 --
550 Lim2(3,1)=1 ! Type : Maximum
560 Lim2(3,2)=9.700E+8 ! Frequency Start: 970.0 MHz
570 Lim2(3,3)=1.0475E+9 ! Stop : 1047.5 MHz
580 Lim2(3,4)=0 ! Response Start: 0 dBm
590 Lim2(3,5)=0 ! Stop : 0 dBm
600 !
610 OUTPUT @Agte507x;":SENS1:FREQ:CENT ";Cent
620 OUTPUT @Agte507x;":SENS1:FREQ:SPAN ";Span
630 OUTPUT @Agte507x;":CALC1:PAR1:COUN 2"
640 OUTPUT @Agte507x;":DISP:WIND1:SPL D1_2"
650 OUTPUT @Agte507x;":TRIG:SOUR BUS"
660 OUTPUT @Agte507x;":INIT1:CONT ON"
670 !
680 ! Trace 1
690 !
700 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
710 !
720 OUTPUT @Agte507x;":CALC1:PAR1:DEF "&Param1$
730 OUTPUT @Agte507x;":CALC1:FORM "&Fmt1$
740 !
750 OUTPUT @Agte507x;":CALC1:LIM:DATA ";Num_of_seg1;
760 FOR Segment=1 TO Num_of_seg1
770 FOR Column=1 TO 5
780 OUTPUT @Agte507x;",";Lim1(Segment,Column);
790 NEXT Column
800 NEXT Segment
810 OUTPUT @Agte507x;""
820 OUTPUT @Agte507x;":CALC1:LIM:DISP ON"
830 OUTPUT @Agte507x;":CALC1:LIM ON"
840 !
850 ! Trace 2
860 !
870 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
880 !
```

```
890 OUTPUT @Agte507x;":CALC1:PAR2:DEF "&Param2$
900 OUTPUT @Agte507x;":CALC1:FORM "&Fmt2$
910 !
920 OUTPUT @Agte507x;":CALC1:LIM:DATA ";Num_of_seg2;
930 FOR Segment=1 TO Num_of_seg2
940 FOR Column=1 TO 5
950 OUTPUT @Agte507x;",";Lim2(Segment,Column);
960 NEXT Column
970 NEXT Segment
980 OUTPUT @Agte507x;""
990 OUTPUT @Agte507x;":CALC1:LIM:DISP ON"
1000 OUTPUT @Agte507x;":CALC1:LIM ON"
1010 !
1020 ! Setting status registers
1030 !
1040 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1:ENAB 6"
1050 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1:PTR 6"
1060 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1:NTR 0"
1070 OUTPUT @Agte507x;":STAT:QUES:LIM:PTR 2"
1080 OUTPUT @Agte507x;":STAT:QUES:LIM:NTR 0"
1090 OUTPUT @Agte507x;"*CLS"
1100 !
1110 OUTPUT @Agte507x;":TRIG:SING"
1120 OUTPUT @Agte507x;"*OPC?"
1130 ENTER @Agte507x;Buff$
1140 !
1150 ! Checking test results
1160 !
1170 OUTPUT @Agte507x;":STAT:QUES:LIM?"
1180 ENTER @Agte507x;Reg_val
1190 Ch1_judge=BIT(Reg_val,1)
1200 OUTPUT @Agte507x;":STAT:QUES:LIM:CHAN1?"
1210 ENTER @Agte507x;Reg_val
1220 Tr1_judge=BIT(Reg_val,1)
1230 Tr2_judge=BIT(Reg_val,2)
1240 !
1250 ! Displaying test results
```

```
1260 !
1270 IF Ch1_judge=0 THEN
1280 PRINT "## PASS! ##"
1290 ELSE
1300 PRINT "## FAIL! ##"
1310 IF Tr1_judge=0 THEN
1320 PRINT " Trace1(S21): PASS"
1330 ELSE
1340 PRINT " Trace1(S21): FAIL"
1350 !
1360 ! Reading and displaying frequency at failed points
1370 !
1380 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
1390 OUTPUT @Agte507x;":CALC1:LIM:REP:POIN?"
1400 ENTER @Agte507x;Fail_point
1410 REDIM Fail_data(1:Fail_point)
1420 OUTPUT @Agte507x;":CALC1:LIM:REP?"
1430 ENTER @Agte507x;Fail_data(*)
1440 PRINT " Frequency:"
1450 FOR I=1 TO Fail_point
1460 PRINT USING "3X,MD.4DE";Fail_data(I)
1470 NEXT I
1480 END IF
1490 IF Tr2_judge=0 THEN
1500 PRINT " Trace2(S11): PASS"
1510 ELSE
1520 PRINT " Trace2(S11): FAIL"
1530 !
1540 ! Reading and displaying frequency at failed points
1550 !
1560 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
1570 OUTPUT @Agte507x;":CALC1:LIM:REP:POIN?"
1580 ENTER @Agte507x;Fail_point
1590 REDIM Fail_data(1:Fail_point)
1600 OUTPUT @Agte507x;":CALC1:LIM:REP?"
1610 ENTER @Agte507x;Fail_data(*)
1620 PRINT " Frequency:"
```

```
1630 FOR I=1 TO Fail_point
1640 PRINT USING "3X,MD.4DE";Fail_data(I)
1650 NEXT I
1660 END IF
1670 END IF
1680 END
```

Description

Line 50

Assigns a GPIB address to the I/O pass.

Lines 70 to 120

These lines store the sweep center value, sweep span value, trace 1 measurement parameter, trace 2 measurement parameter, trace 1 data format, and trace 2 data format into the variables Cent, Span, Param1$, Param2$, Fmt1$, and Fmt2$, respectively.

Line 150

Stores the number of segments in trace 1 limit table into the Num_of_seg1 variable.

Lines 160 to 390

These lines store the settings in trace 1 limit table into the Lim1(*) variable.

Line 410

Stores the number of segments in trace 2 limit table into the Num_of_seg2 variable.

Lines 420 to 590

These lines store the settings in trace 2 limit table into the Lim2(*) variable.

Lines 610 to 620

These lines configure the sweep range for channel 1's sweep range using the center and span values contained in the Cent and Span values.

Lines 630 to 660

These lines configure channel 1 so that it contains 2 traces, displays graphs in two windows tiled horizontally (i.e., with the screen split into upper and lower halves), uses a bus trigger source, and works in continuous activation mode.

Line 700

Sets channel 1's active trace to trace 1.

Lines 720 to 730

These lines store trace 1's measurement parameter and data format into the variables Param1$ and Fmt1$, respectively.

Lines 750 to 810

These lines set up the limit table for trace 1.

Line 750: Sends the command that sets up a limit table along with the Num_of_seg1 variable that contains the number of segments.

Lines 770 to 790: Sends five data items (type, start point stimulus value, end point stimulus value, start point response value, and end point response value) for each segment.

Lines 820 to 830

These lines turn on the display of limit lines and the Limit Test feature for trace 1.

Line 870

Sets channel 1's active trace to trace 2.

Lines 890 to 900

These lines set trace 2's measurement parameter and data format to Param2$ and Fmt2$, respectively.

Lines 920 to 980

These lines set up the limit table for trace 2.

Lines 990 to 1000

These lines turn on the display of limit lines and the Limit Test feature for trace 2.

Lines 1040 to 1060

These lines set, under the questionable limit channel 1 status register, the enable register and positive transition filter to 6 (0000000000000110 in binary notation) while setting the negative transition filter to 0 so that the questionable limit status condition register's bit 1 is set to 1 when the test result that combines the results for trace 1 and trace 2 is "fail."

The sample program provides an example of explicitly configuring the register bits so that they reflect the test results that only cover trace 1 and trace 2. However, because the results for traces 3 to 9 will never be "fail" as long as the Limit Test feature is disabled for those traces, the register bits would reflect the test result that is limited to traces 1 and 2, even if the default setting is not changed.

Lines 1070 to 1080

These lines set transition filters so that the questionable limit status event register's bit 1 is set to 1 when the questionable limit status condition register's bit 1 changes from 0 to 1.

Line 1090

Clears the questionable limit status event register and questionable limit channel 1 status event register.

Lines 1110 to 1130

These lines trigger the instrument and wait until the sweep cycle is completed.

Lines 1170 to 1190

These lines retrieve the value of the questionable limit status event register and store the setting of bit 1 of the value into Ch1_judge.

Lines 1200 to 1230

These lines retrieve the value of the questionable limit channel 1 status event register and store the settings of bit 1 and bit 2 of the value into Tr1_judge and Tr2_judge, respectively.

Line 1280

Displays a message indicating that the DUT has passed the limit test if the test result for channel 1 is "Pass" (i.e., if Ch1_judge returns 0).

Lines 1300 to 1660

These lines are executed if the test result for channel 1 is "Fail" (i.e., if Ch1_judge returns 1).

Line 1300: Notifies the user that the limit test result is "Fail".

Line 1320: Displays a message indicating that trace 1 has passed the limit test if the test result for trace 1 is "Pass" (i.e., if Tr1_judge returns 0).

Lines 1340 to 1470: These lines are executed if the test result for trace 1 is "Fail" (i.e., if Tr1_judge returns 1). The lines notify the user that the test result for trace 1 is "Fail" and then retrieve and display the frequencies at the failed measurement points on trace 1.

Line 1340: Notifies the user that the limit test result for trace 1 is "Pass."

Line 1380: Sets channel 1's active trace to trace 2.

Lines 1390 to 1410: These lines retrieve the number of failed measurement points on trace 1 and, based on that number, resize the array that contains retrieved frequencies.

Lines 1420 to 1470: These lines retrieve and display the frequencies at the failed measurement points on trace 1.

Line 1500: Displays a message indicating that trace 2 has passed the limit test if the test result for trace 2 is "Pass" (i.e., if Tr2_judge returns 0).

Lines 1520 to 1650: If the test result for trace 2 is "Fail" (i.e., if Tr2_judge returns 1), these lines notify the user that trace 2 has failed to pass the limit test and then retrieve and display the frequencies at the failed measurement points on trace 2.

Bandwidth Search

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to perform Bandwidth Search.

The sample program moves the marker to the maximum value position and then retrieves and displays the results of Bandwidth Search.

See Bandwidth Search for this programming.

Sample Program in Excel VBA

```
Sub Bandwid_click()
    Dim vi As Long
    Dim defrm As Long
    Dim Threshhold As Long
    Dim Result As String * 1000
    Dim Bdata As Variant
    Const TimeOutTime = 10000
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    Threshhold = -3
    Call viVPrintf(vi, ":SENS1:FREQ:CENT 947.5E6" & vbLf, 0)
    Call viVPrintf(vi, ":SENS1:FREQ:SPAN 200E6" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:DEF S21" & vbLf, 0)
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:FUNC:TYPE MAX" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:FUNC:EXEC" & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:BWID:THR " + CStr(Threshhold) & vbLf, 0)
    Call viVPrintf(vi, ":CALC1:MARK1:BWID:DATA?" & vbLf, 0)
    Call viVScanf(vi, "%t", Result)
    Bdata = Split(Result, ",")
    Call ErrorCheck(vi)
    Cells(5, 2).Value = Val(Bdata(0))
    Cells(6, 2).Value = Val(Bdata(1))
```

```
    Cells(7, 2).Value = Val(Bdata(2))
    Cells(8, 2).Value = Val(Trim(Bdata(3)))
    Call viClose(vi)
    Call viClose(defrm)

End Sub
Sub ErrorCheck(vi)
    Dim err As String * 50, ErrNo As Variant, Response
    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)
    ErrNo = Split(err, ",")
    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)
    End If
End Sub
```

Sample Program in HT Basic (bandwid.htb)

```
10 DIM Buff$[9],Err_msg$[100]
20 REAL Threshold,Bwid,Cent,Q,Loss
30 INTEGER Err_no
40 !
50 ASSIGN @Agte507x TO 717
60 Threshold=-3
70 !
80 OUTPUT @Agte507x;"*ESE 60"
90 OUTPUT @Agte507x;"*SRE 32"
100 OUTPUT @Agte507x;"*CLS"
110 OUTPUT @Agte507x;"*OPC?"
120 ENTER @Agte507x;Buff$
130 ON INTR 7 GOTO Err
140 ENABLE INTR 7;2
150 !
160 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
170 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:TYPE MAX"
180 OUTPUT @Agte507x;":CALC1:MARK1:FUNC:EXEC"
190 OUTPUT @Agte507x;":CALC1:MARK1:BWID:THR ";Threshold
200 OUTPUT @Agte507x;":CALC1:MARK1:BWID:DATA?"
210 WAIT .5
220 ENTER @Agte507x;Bwid,Cent,Q,Loss
```

```
230 !
240 PRINT "## Bandwidth Search ##"
250 PRINT "Bandwidth : ",Bwid
260 PRINT "Center Frequency: ",Cent
270 PRINT "Q : ",Q
280 PRINT "Loss : ",Loss
290 !
300 GOTO No_err
310 Err: OFF INTR 7
320 OUTPUT @Agte507x;":SYST:ERR?"
330 ENTER @Agte507x;Err_no,Err_msg$
340 PRINT "Error occurred!!"
350 PRINT " No:";Err_no,"Description: "&Err_msg$
360 No_err: OFF INTR 7
370 END
```

Saving Files

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to save a file. This program saves selected content on a file with a specified name.

See Saving and Recalling File for this programming.

Sample Program in Excel VBA

```
Sub File_Save()
  ' Declare two string variables for file name and file type
  Dim File_Name As String
  Dim File_Type As String
  Dim defrm As Long
  Dim vi As Long
  Const TimeOutTime = 10000
  ' Check whether file name textbox is empty or not
  If TextBox1.Text <> "" Then
    File_Name = Trim(TextBox1.Text)
    File_Type = Trim(frmFileSave.ComboBox1.Value)
  ' Open connection to the E5071C
    Call viOpenDefaultRM(defrm)
    Call viOpen(defrm, "GPIB0::18::INSTR", 0, 0, vi)
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)
    Select Case File_Type
      Case "1: State (State only)"
        Call viVPrintf(vi, ":MMEM:STOR:STYP STAT" + vbLf, 0)
        Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
      Case "2: State (State & Cal)"
        Call viVPrintf(vi, ":MMEM:STOR:STYP CST" + vbLf, 0)
        Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
      Case "3: State (State & Trace)"
        Call viVPrintf(vi, ":MMEM:STOR:STYP DST" + vbLf, 0)
        Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
      Case "4: State (State & Cal & Trace)"
```

```
            Call viVPrintf(vi, ":MMEM:STOR:STYP CDST" + vbLf, 0)
            Call viVPrintf(vi, ":MMEM:STOR """ & File_Name & ".sta""" + vbLf, 0)
         Case "5: State (Trace Data (CSV))"
            Call viVPrintf(vi, ":MMEM:STOR:FDAT """ & File_Name & ".csv""" + vbLf, 0)
         Case "6: State (Screen)"
            Call viVPrintf(vi, ":MMEM:STOR:IMAG """ & File_Name & ".bmp""" + vbLf, 0)
         Case Else
            msgbox "Error in code"
      End Select
   Call viClose(defrm)
   Else
      msgbox "Please enter a filename"
   End If
End Sub
```

Sample Program in HT Basic (file_sav.htb)

```
10 DIM File$[300],Inp_char$[30]
20 INTEGER Content
30 CLEAR SCREEN
40 ASSIGN @Agte507x TO 717
50 !
60 ON ERROR GOTO Content_select
70 Content_select: !
80 PRINT "## Save Content Selection ##"
90 PRINT "Select Content"
100 PRINT " 1: State (State only)"
110 PRINT " 2: State (State & Cal)"
120 PRINT " 3: State (State & Trace)"
130 PRINT " 4: State (State & Cal & Trace)"
140 PRINT " 5: Trace Data (CSV)"
150 PRINT " 6: Screen"
160 PRINT ""
170 PRINT "Input 1 to 6"
180 INPUT "Number?",Inp_char$
190 Content=IVAL(Inp_char$,10)
200 IF Content<1 OR Content>6 THEN Content_select
210 OFF ERROR
220 !
```

```
230 CALL Inp_file_name(File$)
240 !
250 SELECT Content
260 CASE 1
270 OUTPUT @Agte507x;":MMEM:STOR:STYP STAT"
280 OUTPUT @Agte507x;":MMEM:STOR """&File$&".sta"""
290 CASE 2
300 OUTPUT @Agte507x;":MMEM:STOR:STYP CST"
310 OUTPUT @Agte507x;":MMEM:STOR """&File$&".sta"""
320 CASE 3
330 OUTPUT @Agte507x;":MMEM:STOR:STYP DST"
340 OUTPUT @Agte507x;":MMEM:STOR """&File$&".sta"""
350 CASE 4
360 OUTPUT @Agte507x;":MMEM:STOR:STYP CDST"
370 OUTPUT @Agte507x;":MMEM:STOR """&File$&".sta"""
380 CASE 5
390 OUTPUT @Agte507x;":MMEM:STOR:FDAT """&File$&".csv"""
400 CASE 6
410 OUTPUT @Agte507x;":MMEM:STOR:IMAG """&File$&".bmp"""
420 END SELECT
430 !
440 END
450 !=========================================
460 ! File Name Input Function
470 !=========================================
480 SUB Inp_file_name(Inp_name$)
490 DIM Inp_char$[9]
500 ON ERROR GOTO Inp_start
510 Inp_start: !
520 PRINT "## File Name Input ##"
530 PRINT "Input Save File Name (without Extension)"
540 INPUT "Name?",Inp_name$
550 PRINT "Input Name: "&Inp_name$
560 INPUT "OK? [Y/N]",Inp_char$
570 IF UPC$(Inp_char$)<>"Y" THEN Inp_start
580 OFF ERROR
590 SUBEND
```

Transferring Files

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates to transfer files between the external controller and the E5071C.

This program reads out data from a specified file on the external controller (or the E5071C), then write them to a specified file on the E5071C (or the external controller).

See Managing Files for this programming.

Sample Program in HT Basic (file_xfr.htb)

```
10 DIM Src_file$[50],Dst_file$[50],Src_size_char$[50],Inp_char$[30]
20 INTEGER Direction
30 ASSIGN @Agte507x TO 717
40 !
50 CLEAR SCREEN
60 ON ERROR GOTO Direct_select
70 Direct_select: !
80 PRINT "#### File Transfer ####"
90 PRINT " 1: E507x -> Controller"
100 PRINT " 2: Controller -> E507x"
110 PRINT ""
120 PRINT "Input 1 or 2"
130 INPUT "Number?",Inp_char$
140 Direction=IVAL(Inp_char$,10)
150 IF Direction<1 OR Direction>2 THEN Direct_select
160 OFF ERROR
170 !
180 PRINT ""
190 PRINT " Input source file name. ";
200 INPUT "Name?",Src_file$
210 PRINT ": "&Src_file$
220 !
230 IF Direction=2 THEN
240 PRINT " Input source file size. ";
250 INPUT "Size[Byte]?",Src_size_char$
```

```
260 PRINT ": "&Src_size_char$&"[Byte]"
270 END IF
280 !
290 PRINT " Input destination file name. ";
300 INPUT "Name?",Dst_file$
310 PRINT ": "&Dst_file$
320 PRINT ""
330 !
340 IF Direction=1 THEN
350 Copy_to_contr(@Agte507x,Src_file$,Dst_file$)
360 ELSE
370 Copy_to_e507x(@Agte507x,Src_file$,Src_size_char$,Dst_file$)
380 END IF
390 !
400 END
410 !=========================================
420 ! File Transfer Function (E507x -> Controller)
430 !=========================================
440 SUB Copy_to_contr(@Agte507x,Src_file$,Dst_file$)
450 DIM Img$[32],Src_size_char$[10],Buff$[9],Err_msg$[100]
460 INTEGER Max_bsize,Block_size,Err_no
470 REAL Src_size
480 !
490 ON ERROR GOTO Skip_purge
500 PURGE Dst_file$
510 Skip_purge: OFF ERROR
520 CREATE Dst_file$,1
530 ASSIGN @Dst_file TO Dst_file$
540 Max_bsize=24576 ! 24KByte
550 !
560 OUTPUT @Agte507x;"*ESE 60"
570 OUTPUT @Agte507x;"*SRE 32"
580 OUTPUT @Agte507x;"*CLS"
590 OUTPUT @Agte507x;"*OPC?"
600 ENTER @Agte507x;Buff$
610 !
620 ON INTR 7 GOTO Err
```

```
630 ENABLE INTR 7;2
640 PRINT "Now Copying: "&Src_file$&"(@E507x) -> "&Dst_file$&"(@Contro ller)"
650 OUTPUT @Agte507x;":MMEM:TRAN? """&Src_file$&""""
660 WAIT .1
670 ENTER @Agte507x USING "#,A";Buff$
680 ENTER @Agte507x USING "#,A";Digit$
690 Img$="#,"&Digit$&"A"
700 ENTER @Agte507x USING Img$;Src_size_char$
710 !
720 Src_size=VAL(Src_size_char$)
730 WHILE Src_size>0
740 IF Src_size>Max_bsize THEN
750 Block_size=Max_bsize
760 ELSE
770 Block_size=Src_size
780 END IF
790 !
800 ALLOCATE Dat$[Block_size]
810 Img$="#,"&VAL$(Block_size)&"A"
820 ENTER @Agte507x USING Img$;Dat$
830 OUTPUT @Dst_file USING Img$;Dat$
840 DEALLOCATE Dat$
850 !
860 Src_size=Src_size-Block_size
870 END WHILE
880 !
890 PRINT "Done"
900 ENTER @Agte507x USING "#,A";Buff$
910 ASSIGN @Dst_file TO *
920 !
930 GOTO Skip_error
940 Err: OFF INTR 7
950 OUTPUT @Agte507x;":SYST:ERR?"
960 ENTER @Agte507x;Err_no,Err_msg$
970 PRINT "Error occurred!!"
980 PRINT " No:";Err_no,"Description: "&Err_msg$
990 Skip_error: OFF INTR 7
```

```
1000 SUBEND
1010 !=========================================
1020 ! File Transfer Function (Controller -> E507x)
1030 !=========================================
1040 SUB Copy_to_e507x(@Agte507x,Src_file$,Src_size_char$,Dst_file$)
1050 DIM Img$[32],Header$[10],Buff$[9],Err_msg$[100]
1060 INTEGER Max_bsize,Block_size,Err_no
1070 REAL Src_size
1080 !
1090 ON ERROR GOTO File_error
1100 ASSIGN @Src_file TO Src_file$
1110 OFF ERROR
1120 Max_bsize=24576 ! 24KByte
1130 !
1140 OUTPUT @Agte507x;"*CLS"
1150 OUTPUT @Agte507x;"*OPC?"
1160 ENTER @Agte507x;Buff$
1170 !
1180 PRINT "Now Copying: "&Src_file$&"(@Controller) -> "&Dst_file$&"(@ E507x)"
1190 Header$="#"&VAL$(LEN(Src_size_char$))&Src_size_char$
1200 OUTPUT @Agte507x;":MMEM:TRAN """&Dst_file$&""","&Header$;
1210 !
1220 Src_size=VAL(Src_size_char$)
1230 WHILE Src_size>0
1240 IF Src_size>Max_bsize THEN
1250 Block_size=Max_bsize
1260 ELSE
1270 Block_size=Src_size
1280 END IF
1290 !
1300 ALLOCATE Dat$[Block_size]
1310 Img$="#,"&VAL$(Block_size)&"A"
1320 ENTER @Src_file USING Img$;Dat$
1330 OUTPUT @Agte507x USING Img$;Dat$
1340 DEALLOCATE Dat$
1350 !
1360 Src_size=Src_size-Block_size
```

```
1370 END WHILE
1380 !
1390 OUTPUT @Agte507x;"",END
1400 ASSIGN @Src_file TO *
1410 !
1420 OUTPUT @Agte507x;";:SYST:ERR?"
1430 ENTER @Agte507x;Err_no,Err_msg$
1440 IF Err_no=0 THEN
1450 PRINT "Done"
1460 ELSE
1470 PRINT "Error occurred!!"
1480 PRINT " No:";Err_no,"Description: "&Err_msg$
1490 END IF
1500 GOTO Skip_error
1510 File_error:OFF ERROR
1520 PRINT "File name NOT found!"
1530 Skip_error:!
1540 SUBEND
```

Description

Line 40

Assigns a GPIB address to the I/O pass.

Lines 60 to 130

These line allow the user to return to the entry start line and re-enter the data if an error (such as an invalid entry) occurs while entering the number that indicates the transfer direction. Then, these line display the list of transfer directions and prompt the user to input a selected number.

Lines 80 to 130

These lines display the list of transfer directions, and prompt the user to choose one of the items by typing in the appropriate number.

Lines 140 to 150

Converts the entered value into an integer and stores it into the Direction variable. Returns to the entry start line if an invalid value is contained in Direction.

Lines 180 to 210

These lines obtain the name of the source file for copying from the user input, store it into the Src_file$ variable, and display the value of Src_file$.

Lines 180 to 210

These lines obtain the name of the source file for copying from the user input, store it into the Src_file$ variable, and display the value of Src_file$.

Lines 230 to 270

If Direction is equal to 2 (from the external controller to the E5071C), these lines obtain the size of the source file for copying, store it into the Src_size_char$, and display the value of Src_size_char$.

Lines 290 to 320

These lines obtain the name of the destination file for copying from the user input, store it into the Dst_file$ variable, and display the value of Dst_file$.

Line 350

If Direction is equal to 1 (from the E5071C to the external controller), these lines use the subprogram Copy_to_contr to transfer (copy) a file with the name Src_file$ on the E5071C to a file with the name Dst_file$ on the external controller.

Line 370

If Direction is equal to 2, these lines use the subprogram Copy_to_e507x to transfer (copy) a file with the name Src_file$ on the external controller to a file with the name Dst_file$ on the E5071C.

Copy_to_contr, a subprogram for transferring files from the E5071C to the external controller that appears in lines 440 to 1000, is described below.

Lines 490 to 520

If any file with the name File$ already exists, these lines delete the file and newly create a file with the name File$.

Line 530

Assigns a destination file for copying to the I/O pass.

Line 540

This line stores a maximum number of transferred data (in bytes) per one transfer, that is 24 KByte to meet the size limitation of string arrays in the HTBasic, into Max_bsize variable.

Lines 560 to 600

These lines configure the system to generate an SRQ when it cannot find a source file for copying due to an error.

Lines 620 to 630

These lines set the branch target for an SRQ interrupt to enable SRQ interrupts.

Lines 640 to 650

These lines display a message showing that the transfer has started, and execute commands for reading data from a file on the E5071C.

Lines 670 to 680

These lines read the header symbol (#) in a block data, read number of digits (characters) indicating the size of data in bytes, then store it into Digit$ variable.

Line 690

This line creates a format for reading characters in Digit$.

Line 700

This line reads the data size in byte and stores it into Src_size_char$ variable.

Line 720

This line converts Src_size_char$ to a real number and stores it into Src_size variable.

Lines 730 to 870

These lines repeat the procedures below until Src_size reaches 0.

Lines 740 to 780: If Src_size is greater than Max_bsize, these lines assign the value of the Max_bsize to Block_size variable (transferred data in bytes). If Src_size is equal or less than Max_bsize, assign the value of Src_size to Block_size.

Line 800 This line defines Dat$ string variable with the size as large as Block_size and reserves memory area.

Line 810 This line creates a format for reading characters as many as Block_size characters.

Line 820 This line reads data from the file on the E5071C, then stores them into Dat$.

Line 830 This line writes the contents of Dat$ to the file on the external controller.

Lines 840 to 860 These lines free the memory area for Dat$ and subtract Block_size from Src_size.

Lines 890 to 900

These lines display a message showing the completion of transfer, then read a message terminator at the end of the data.

Lines 940 to 980

These lines define an error handler that retrieves and displays the number and message of an error that has occurred.

Copy_to_e507x, a subprogram for transferring files from the external controller to the E5071C that appears in lines 1040 to 1540, is described below.

Lines 1090 to 1110

Assigns a destination file for copying to the I/O pass.

Line 1120

This line stores a maximum number of transferred data (in bytes) per one transfer, that is 24 KByte, into Max_bsize variable.

Lines 1140 to 1160

Clears the error queue.

Line 1180

Displays a measurement start message.

Lines 1190 to 1200

These lines create the header part indicating that data will be sent as many as Src_size_char$ bytes, then send the header part of the command and its parameters for writing the data to the file on the E5071C.

Line 1220

This line converts Src_size_char$ to a real number and stores it into Src_size variable.

Lines 1230 to 1370

These lines repeat the procedures below until Src_size reaches 0.

Lines 1240 to 1280: If Src_size is greater than Max_bsize, these lines assign the value of the Max_bsiize to Block_size variable (transferred data in bytes). If Src_size is equal or less than Max_bsize, assign the value of Src_size to Block_size.

Line 1300 This line defines Dat$ string variable with the size as large as Block_size and reserves memory area.

Line 1310 This line creates a format for reading characters as many as Block_size characters.

Line 1320 This line reads data from the file on the external controller, then stores them into Dat$.

Line 1330 This line writes the contents of Dat$ to the file on the E5071C.

Lines 1340 to 1360 These lines free the memory area for Dat$ and subtract Block_size from Src_size.

Line 1390

This line sends a message terminator at the end of data.

Lines 1420 to 1430

These lines retrieve the error number and error message from the error queue, and then store them into the variables Err_no and Err_msg$, respectively.

Lines 1440 to 1490

If Err_no is equal to 0 (no error occurred), these lines display the message indicating completion of transfer, and if Err_no is not equal to 0 (an error occurred), display Err_no along with Err_msg$.

Lines 1510 to 1520

These lines handle the case with no source file for copying is found.

Fixture Simulator

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

The sample program demonstrates how to use the Fixture Simulator.

This configures the Balance-Unbalance Conversion, Differential Matching Circuit Embedding, Port Impedance Conversion, and Differential Port Impedance Conversion features so that the instrument can correctly deal with an unbalance-balance (3-port) DUT.

See Analysis Using the Fixture Simulator for this programming.

Sample Program in Excel VBA

```
Sub Fixture_Click()
    Dim defrm As Long          'Session to Default Resource Manager
    Dim vi As Long             'Session to instrument
    Dim Para As String
    Dim Dev_Type As String
    Dim Dmc_c As Double
    Dim Dmc_g As Double
    Dim Dmc_l As Double
    Dim Dmc_r As Double
    Dim Z0_se As Double
    Dim Z0_diff As Double
    Dim Port(3) As Integer
    Dim Result As String

    Const TimeOutTime = 20000

    Dev_Type = "SBAL"              'Divice Type     : SE-Bal
    Port(1) = 1                    'Port1(SE)       : 1
    Port(2) = 2                    'Port2(Bal)      : 2,3
    Port(3) = 3                    '
    Para = "SDS21"                 'Meas. Parameter : Sds21
    Dmc_c = 0.000000000001         'Diff.        C : 1 pF
    Dmc_g = 0.002                  'Matching     G : 2 mS
    Dmc_l = 0.00000003             'Circuit      L : 30 nH
```

```
    Dmc_r = 0.004                '          R : 4 mohm
    Z0_se = 100                 'Z Conversion Z0 : 100 ohm
    Z0_diff = 210               'Diff. Z Conv.Z0 : 210 ohm

    Call viOpenDefaultRM(defrm)          'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)         'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)         'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)    'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)    'Clears the all status register.

    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DEV " & Dev_Type & vbLf, 0)   'Sets the balance device type.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:TOP:" & Dev_Type & " " & Port(1) & "," & Port(2) & "," & Port(3) &
vbLf, 0)  'Sets the port assignment.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:PAR1:" & Dev_Type & " " & Para & vbLf, 0)   'Sets the
measurement parameter of port 1.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:PAR1:STAT ON" & vbLf, 0)    'Turns on balance-unbalance
conversion.
    Call ErrorCheck(vi)   'Checking the error.

    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1 PLPC" & vbLf, 0)   'Specifies the type of differential
matching circuit.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C " & Dmc_c & vbLf, 0)  'Sets the differential
matching circuit constants C.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G " & Dmc_g & vbLf, 0)  'Sets the differential
matching circuit constants G.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:L " & Dmc_l & vbLf, 0)  'Sets the differential
matching circuit constants L.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R " & Dmc_r & vbLf, 0)   'Sets the differential
matching circuit constants R.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DMC:STAT ON" & vbLf, 0)  'Turns of differential matching circuit
embedding.
    Call ErrorCheck(vi)   'Checking the error.

    Call viVPrintf(vi, ":CALC1:FSIM:SEND:ZCON:PORT2:Z0 " & Z0_se & vbLf, 0)   'Sets the port
impedance of port 2.
    Call viVPrintf(vi, ":CALC1:FSIM:SEND:ZCON:PORT3:Z0 " & Z0_se & vbLf, 0)   'Sets the port
impedance of port 3.
    Call viVPrintf(vi, ":CALC1:FSIM:SEND:ZCON:STAT ON" & vbLf, 0)  'Turns on port impedance
conversion.
```

```
    Call ErrorCheck(vi)    'Checking the error.

    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DZC:BPOR1:Z0 " & Z0_diff & vbLf, 0)  'Sets the differential port impedance of balance port 1.
    Call viVPrintf(vi, ":CALC1:FSIM:BAL:DZC:STAT ON" & vbLf, 0)   'Turns on differential port impedance conversion.

    Call viVPrintf(vi, ":CALC1:FSIM:STAT ON" & vbLf, 0)  'Turns on the fixture simulator.
    Call ErrorCheck(vi)   'Checking the error.

    Call viClose(vi)   'Closes the resource manager session.
    Call viClose(defrm)   'Breaks the communication and terminates the VISA system.

    End
End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err) 'Reads error message.
    ErrNo = Split(err, ",")  'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)  'Display the message box.
    End If
End Sub
```

Sample Program in HT Basic (fixture.htb)

```
10 DIM Dev_type$[9],Para$[9]
20 REAL Dmc_c,Dmc_g,Dmc_l,Dmc_r,Z0_se,Z0_diff
30 INTEGER Port(1:3)
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 Dev_type$="SBAL" ! Device Type : SE-Bal
80 Port(1)=1 ! Port1(SE) : 1
90 Port(2)=2 ! Port2(Bal) : 2,3
100 Port(3)=3 !
110 Para$="SDS21" ! Meas. Parameter : Sds21
120 Dmc_c=1.E-12 ! Diff. C: 1 pF
```

```
130 Dmc_g=2.E-3 ! Matching G: 2 mS
140 Dmc_l=3.E-8 ! Circuit L: 30 nH
150 Dmc_r=4.E-3 ! R: 4 mohm
160 Z0_se=100 ! Z Conversion Z0: 100 ohm
170 Z0_diff=210 ! Diff. Z Conv. Z0: 210 ohm
180 !
190 ! Balance-Unbalance Conversion Setting
200 !
210 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DEV "&Dev_type$
220 OUTPUT @Agte507x;":CALC1:FSIM:BAL:TOP:"&Dev_type$&" ";Port(*)
230 OUTPUT @Agte507x;":CALC1:FSIM:BAL:PAR1:"&Dev_type$&" "&Para$
240 OUTPUT @Agte507x;":CALC1:FSIM:BAL:PAR1:STAT ON"
250 !
260 ! Diff. Matching Circuit Setting
270 !
280 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DMC:BPOR1 PLPC"
290 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C ";Dmc_c
300 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G ";Dmc_g
310 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:L ";Dmc_l
320 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R ";Dmc_r
330 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DMC:STAT ON"
340 !
350 ! Z Conversion Setting
360 !
370 OUTPUT @Agte507x;":CALC1:FSIM:SEND:ZCON:PORT2:Z0 ";Z0_se
380 OUTPUT @Agte507x;":CALC1:FSIM:SEND:ZCON:PORT3:Z0 ";Z0_se
390 OUTPUT @Agte507x;":CALC1:FSIM:SEND:ZCON:STAT ON"
400 !
410 ! Diff. Z Conversion Setting
420 !
430 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DZC:BPOR1:Z0 ";Z0_diff
440 OUTPUT @Agte507x;":CALC1:FSIM:BAL:DZC:STAT ON"
450 !
460 ! Fixture Simulator On/Off
470 !
480 OUTPUT @Agte507x;":CALC1:FSIM:STAT ON"
490 !
```

```
500 END
```

Time Domain

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program demonstrates how to use the transformation function of the time domain function.

This program executes calibration (ECal), performs measurement once, converts the results to data in the time domain, and displays this data.

See Analysis in Time Domain (time domain function) for this programming.

Sample Program in Excel VBA

```
Sub Time_Domain_Click()
    Dim defrm As Long                'Session to Default Resource Manager
    Dim vi As Long                   'Session to instrument
    Dim Para As String
    Dim Tran_Type As String
    Dim Stim_Type As String
    Dim stop_freq As Double
    Dim Win_Beta As Double
    Dim Star_Time As Double
    Dim Stop_Time As Double
    Dim Result As String

    Const TimeOutTime = 40000        'timeout time.

    stop_freq = 3000000000#          'Stop Frequendy  : 3 GHz
    Nop = 201                        'Nop             : 201
    Para = "S11"                     'Meas. Parameter : S11
    Tran_Type = "LPAS"               'Transform Type  : Lowpass
    Stim_Type = "IMP"                'Stimulus Type   : Impulse
    Win_Beta = 13                    'Window Beta     : 13 (Maximum Type)
    Star_Time = 0                    'Start time      : 0 sec
    Stop_Time = 0.00000001           'Stop time       : 10 nsec

    Call viOpenDefaultRM(defrm)                          'Initializes the VISA system.
```

```
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)                    'Opens the session to the
specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)             'The state of an attribute
for the specified session.

    Call viVPrintf(vi, ":SYST:PRES" & vbLf, 0)                          'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)                                'Clears the all status register.

    Call viVPrintf(vi, ":SENS1:FREQ:STOP " & stop_freq & vbLf, 0)       'Sets the sweep stop
frequency.
    Call viVPrintf(vi, ":SENS1:SWE:POIN " & Nop & vbLf, 0)              'Sets the number of points.

    Call viVPrintf(vi, ":CALC1:TRAN:TIME:LPFR" & vbLf, 0)               'Sets a measurement point.

    Call viVPrintf(vi, ":CALC1:PAR1:DEF " & Para & vbLf, 0)             'Sets the measurement
parameter.
    Call viVPrintf(vi, ":TRIG:SOUR BUS" & vbLf, 0)                      'Sets the trigger source to BUS.
    Call ErrorCheck(vi)                                                 'Checking the error.

    MsgBox "Connect Port1 to Ecal Module. Then click OK button.", vbOKOnly      'Display the message
box.

    Call viVPrintf(vi, ":SENS1:CORR:COLL:ECAL:SOLT1 1" & vbLf, 0)       'Execute the 1-port
calibration.
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)                               'Sets the *OPC? command.
    Call viVScanf(vi, "%t", Result)                                     'Reads the *OPC? result.
    Call ErrorCheck(vi)                                                 'Checking the error.
    MsgBox "Set DUT. Then click OK button.", vbOKOnly                   'Display the message box.

    Call viVPrintf(vi, ":TRIG:SING" & vbLf, 0)                          'Execute the trigger.
    Call viVPrintf(vi, "*OPC?" & vbLf, 0)                               'Sets the *OPC? command.
    Call viVScanf(vi, "%t", Result)                                     'Reads the *OPC? result.
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)            'Execute auto scale.

    MsgBox "Click OK button. < Time Domain Transform >", vbOKOnly       'Display the message
box.

    Call viVPrintf(vi, ":CALC1:TRAN:TIME " & Tran_Type & vbLf, 0)       'Sets the transformation
type.
```

```
    Call viVPrintf(vi, ":CALC1:TRAN:TIME:STIM " & Stim_Type & vbLf, 0)          'Sets the stimulus type.
    Call viVPrintf(vi, ":CALC1:TRAN:TIME:KBES " & Win_Beta & vbLf, 0)          'Sets the beta value of
the window.
    Call viVPrintf(vi, ":CALC1:TRAN:TIME:STAR " & Star_Time & vbLf, 0)         'Sets the start value of
the display range.
    Call viVPrintf(vi, ":CALC1:TRAN:TIME:STOP " & Stop_Time & vbLf, 0)         'Sets the end value of
the display range.
    Call viVPrintf(vi, ":CALC1:TRAN:TIME:STAT ON" & vbLf, 0)                   'Turns on the time domain
function.
    Call ErrorCheck(vi)                                                        'Checking the error.

    Call viVPrintf(vi, ":CALC1:PAR1:SEL" & vbLf, 0)                            'Sets the active trace.
    Call viVPrintf(vi, ":CALC1:FORM REAL" & vbLf, 0)                           'Sets the real data format.
    Call viVPrintf(vi, ":DISP:WIND1:TRAC1:Y:AUTO" & vbLf, 0)                   'Execute auto scale.
    Call ErrorCheck(vi)                                                        'Checking the error.

    Call viClose(vi)                                                           'Closes the resource manager session.
    Call viClose(defrm)                                                        'Breaks the communication and terminates
the VISA system.

    End                                                                        'End
End Sub
```

Sample Program in HT Basic

```
10 DIM Para$[9],Tran_type$[9],Stim_type$[9],Buff$[9],Inp_ch ar$[9]
20 REAL Stop_freq,Win_beta,Star_time,Stop_time
30 INTEGER Nop
40 !
50 ASSIGN @Agte507x TO 717
60 !
70 Stop_freq=3.E+9 ! Stop Frequency : 3 GHz
80 Nop=201 ! Nop : 201
90 Para$="S11" ! Meas. Parameter : S11
100 !
110 Tran_type$="LPAS" ! Transform Type : Lowpass
120 Stim_type$="IMP" ! Stimulus Type : Impulse
130 Win_beta=13 ! Window Beta : 13 (Maximum Type)
140 Star_time=0 ! Start time : 0 s
150 Stop_time=1.E-8 ! Stop time : 10 ns
160 !
```

```
170 OUTPUT @Agte507x;":SYST:PRES"
180 OUTPUT @Agte507x;":SENS1:FREQ:STOP ";Stop_freq
190 OUTPUT @Agte507x;":SENS1:SWE:POIN ";Nop
200 !
210 OUTPUT @Agte507x;":CALC1:TRAN:TIME:LPFR"
220 !
230 OUTPUT @Agte507x;":CALC1:PAR1:DEF "&Para$
240 OUTPUT @Agte507x;":TRIG:SOUR BUS"
250 !
260 ! 1 Port Full Calibration (ECal)
270 !
280 PRINT "Connect Port 1 to ECal Module. Then push [Enter] key."
290 INPUT "",Buff$
300 OUTPUT @Agte507x;":SENS1:CORR:COLL:ECAL:SOLT1 1"
310 OUTPUT @Agte507x;":SYST:ERR?"
320 ENTER @Agte507x;Buff$
330 !
340 ! Measurement
350 !
360 PRINT "Set DUT. Then Push [Enter] key."
370 INPUT "",Inp_char$
380 !
390 OUTPUT @Agte507x;":TRIG:SING"
400 OUTPUT @Agte507x;"*OPC?"
410 ENTER @Agte507x;Buff$
420 !
430 OUTPUT @Agte507x;":DISP:WIND1:TRAC1:Y:AUTO"
440 PRINT "Push [Enter] key. -> [Time Domain Transform]"
450 INPUT "",Inp_char$
460 !
470 ! Time Domain Transform
480 !
490 OUTPUT @Agte507x;":CALC1:TRAN:TIME "&Tran_type$
500 OUTPUT @Agte507x;":CALC1:TRAN:TIME:STIM "&Stim_type$
510 OUTPUT @Agte507x;":CALC1:TRAN:TIME:KBES ";Win_beta
520 OUTPUT @Agte507x;":CALC1:TRAN:TIME:STAR ";Star_time
530 OUTPUT @Agte507x;":CALC1:TRAN:TIME:STOP ";Stop_time
```

```
540 OUTPUT @Agte507x;":CALC1:TRAN:TIME:STAT ON"
550 !
560 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
570 OUTPUT @Agte507x;":CALC1:FORM REAL"
580 OUTPUT @Agte507x;":DISP:WIND1:TRAC1:Y:AUTO"
590 END
```

Control Using SICL-LAN Server

- Overview
- Sample Program in Excel VBA

Other topics about Sample Application Program

Overview

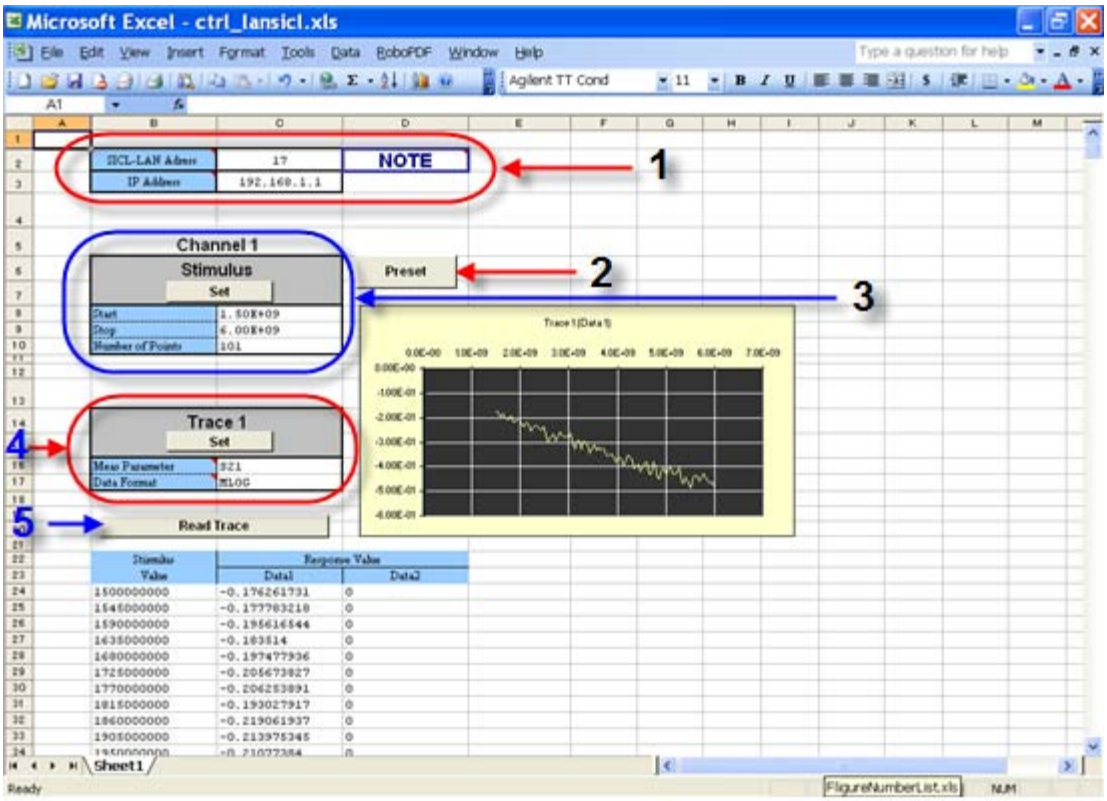
This section explains how to control the E5071C by using SICL in the Windows environment.

- To control the E5071C using the SICL-LAN server, you need to make the preparations described in Control over SICL-LAN server.

Sample Program in Excel VBA

Opening **ctrl_lansicl.xls** in Microsoft Excel displays a screen as shown in the figure below:

ctrl_lansicl.xls



e5071c191

For how to use each element in ctrl_lansicl.xls, refer to the following description.

1. In part 1, in the cell to the right of the SICL-LAN Address, enter the address of the E5071C for control with the SICL-LAN server. This address is **XX**, which has been set with the command **System** > **Misc**

Setup > **Network Setup** > **SICL-LAN Address** [**XX**]. Enter the IP address of the E5071C in the cell to the right of the IP Address. This VBA macro will not work properly without the correct values in these two cells.

2. Click Preset in part 2 to execute the presetting operation.
3. In part 3, the sweep range (start and stop points) and the number of measurement points for channel 1 are set. Click **Set** to execute the setting as shown in the setting table.
4. Part 4 sets the measurement parameters and data format for trace 1 in channel 1. Click **Set** to execute the setting as shown in the setting table.
5. Click **Read Trace** in part 5 retrieves the formatted data array of trace 1 in channel 1. The data is displayed in tabular and graphical format.

Description of Operation in VBA macro

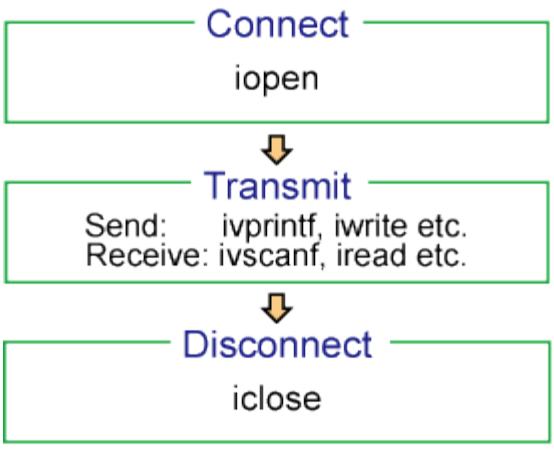
This section describes the operation of the VBA macro, focusing on the part related to control with SICL.

- In order to use SICL in your VBA macro, you must declare functions and define variables with a SICL definition file (for VB).
- In the VBA macro, ctrl_sicllan.xls, the standard module whose object name is "SICL," is the definition file**.**

The basic control flow with SICL is shown in Flow of control using SICL.

- In this sample program, the **ivprintf** function, the **ivscanf** function, and the **iread** function are used in its communication part; you can use other SICL functions as well. For details, refer to sicl.hlp (the online help of SICL).

Flow of control using SICL



e5071c441

- For more information on how to use each function of SICL, refer to the SICL manual.

The procedures of each step in Flow of control using SICL are described below.

Connection

The procedure corresponding to connection is OpenSession (OpenSession). OpenSession establishes a connection to the E5071C with the **iopen** function of SICL, using the SICL-LAN Address and IP Address entered in part 1 in ctrl_lansicl.xls. The **iopen** function takes the address information of the E5071C you specify as its parameters.

Syntax

addr = iopen(*dev*)

Variable	**addr**
Description	Session information (output)
Data type	Integer type
	dev
Description	Address information of the instrument you specify (input)
Data type	Character string type

Grammar	*sicl-name [ip-address]:interface, sicl-lan-address*

For example, if the parameter (*dev*) is "lan[192.168.0.1]:hpib9,17," connection is made to the address of **17** of the interface of **hpib9** with the E5071C whose IP address is **192.168.0.1** by using the external controller whose SICL interface name is **lan**.

OpenSession

```
Function OpenSession() As Integer
Dim ServAddr As String
Dim IpAddr As String
On Error GoTo ErrHandler
'"Get Sicl-Lan Address
Sheets("Sheet1").Select
Range("C2").Select
ServAddr = ActiveCell.FormulaR1C1
'"Get Ip Address
Sheets("Sheet1").Select
Range("C3").Select
IpAddr = ActiveCell.FormulaR1C1

OpenSession = iopen("lan[" & IpAddr & "]:hpib9," & ServAddr)
Call itimeout(OpenSession, 10000)
Exit Function
ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End
End Function
```

Sending

The procedure corresponding to sending in communication is OutputSiclLan. OutputSiclLan uses the **ivprintf** function of SICL to send messages (SCPI commands). The **ivprintf** function takes the session information output from the **iopen** function and a program message as its parameters.

Syntax

Status = ivprintf(*addr*,*mes*)

Variable	**Status**
Description	Return value of the function (output)
Data type	Integer type
	addr
Description	Session information (input)
Data type	Integer type
	mes
Description	Program message (input)
Data type	Character string type

OutputSiclLan

```
Sub OutputSiclLan(addr As Integer, message As String)

Dim Status As Integer
Dim actualcnt As Long
Dim length As Long

On Error GoTo ErrHandler
length = Len(message)
Status = ivprintf(addr, message & Chr$(10))
Exit Sub

ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End
```

```
End Sub
```

Receiving

The procedure corresponding to receiving ASCII format messages in communication is EnterSiclLan. EnterSiclLan uses the **ivscanf** function of SICL to receive a message in ASCII format and store it into the output variable. The **ivscanf** function takes the session information output from the **iopen** function, the format for output, and the data to be output as its parameters.

Syntax

Status = ivscanf(*addr*,*fmt*,*ap*)

Variable	**fmt**
Description	Format for output (input)
Data type	Character string type
	ap
Description	Data to be output (output)
Data type	Character string type

For information on the variable (*Status*) and the variable (*addr*), refer to Variable.

In Visual Basic, variables must be declared as a fixed-length string when receiving string data using the **ivscanf** function.

EnterSiclLan

```
Sub EnterSiclLan(addr As Integer, Query As String)

Dim Status As Integer
Dim actualcnt As Long
Dim res As String * 256
```

```
On Error GoTo ErrHandler
Status = ivscanf(addr, "%t", res)
Query = Trim(res)
Exit Sub

ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End

End Sub
```

The procedure corresponding to receiving array data in communication is EnterSiclLanArrayReal64, which uses the **iread** function of SICL to receive array data in the IEEE 64-bit floating point binary transfer format and store it into the output variable. The **iread** function takes the session information output from the **iopen** function, the data to be output, the number of data bytes, the condition to finish reading data, and the number of data bytes actually read out as its parameters.

Syntax

Status = iread(*addr*,*buf*,*bufsize*,*reason*,*actual*)

Variable	**buf**
Description	Data to be output (output)
Data type	Character string type
	bufsize
Description	The number of data bytes (input)
Data type	Long integer type
	reason
Description	The condition to finish reading out data (input)
Data type	Integer type

	actual
Description	The number of data bytes actually read out (output)
Data type	Long integer type

For information on the variable (*Status*) and the variable (*addr*), refer to Variable.

Each functional part of EnterSiclLanArrayReal64 is described below.

(1) Retrieves the data header.

(2) Stores the number of data bytes into the size variable in the header part.

(3) Retrieves the formatted data array for trace 1 in channel 1 and stores it into the databuf variable.

(4) Retrieves the message terminator at the end of the data.

EnterSiclLanArrayReal64

Function EnterSiclLanArrayReal64(addr As Integer, databuf() As Double) As Long

```
Dim Status As Integer
Dim actualcnt As Long
Dim buf As String * 8
Dim size As Long
On Error GoTo ErrHandler
'"Read header info of "#6NNNNNN"
Status = iread(addr, buf, 8, I_TERM_MAXCNT, actualcnt) '.....(1)
size = Val(Mid$(buf, 3, 6)) '.....(2)
'"Read data
Status = iread(addr, databuf, size, I_TERM_MAXCNT, actualcnt) '.....(3)
'"Read ending LF
Status = iread(addr, buf, 1, I_TERM_MAXCNT, actualcnt) '.....(4)
EnterSiclLanArrayReal64 = size / 8
Exit Function
```

```
ErrHandler:
MsgBox "*** Error : " & Error$
Call siclcleanup
End
End Function
```

Disconnection

The **iclose** function of SICL is used to disconnect communication. The **iclose** function takes the session information output from the **iopen** function as its parameter.

Syntax

Status = iclose(*addr*)

For information on the variable (*Status*) and the variable (*addr*), refer to Variable

Sample control

The E5071C can be controlled by executing the above procedures in order, following the control flow in Flow of control using SICL. This is demonstrated by the Preset procedure (a procedure that is executed when the Preset button is clicked) as described in Preset.

Preset

```
Sub Preset()

''' Open Session
E507x = OpenSession

'''Presetting the analyzer
Call OutputSiclLan(E507x, ":SYST:PRES")
'''Close Session
Call iclose(E507x)

End Sub
```

Control Using Telnet Server

- Overview
- Sample Program in Excel VBA

Other topics about Sample Programs

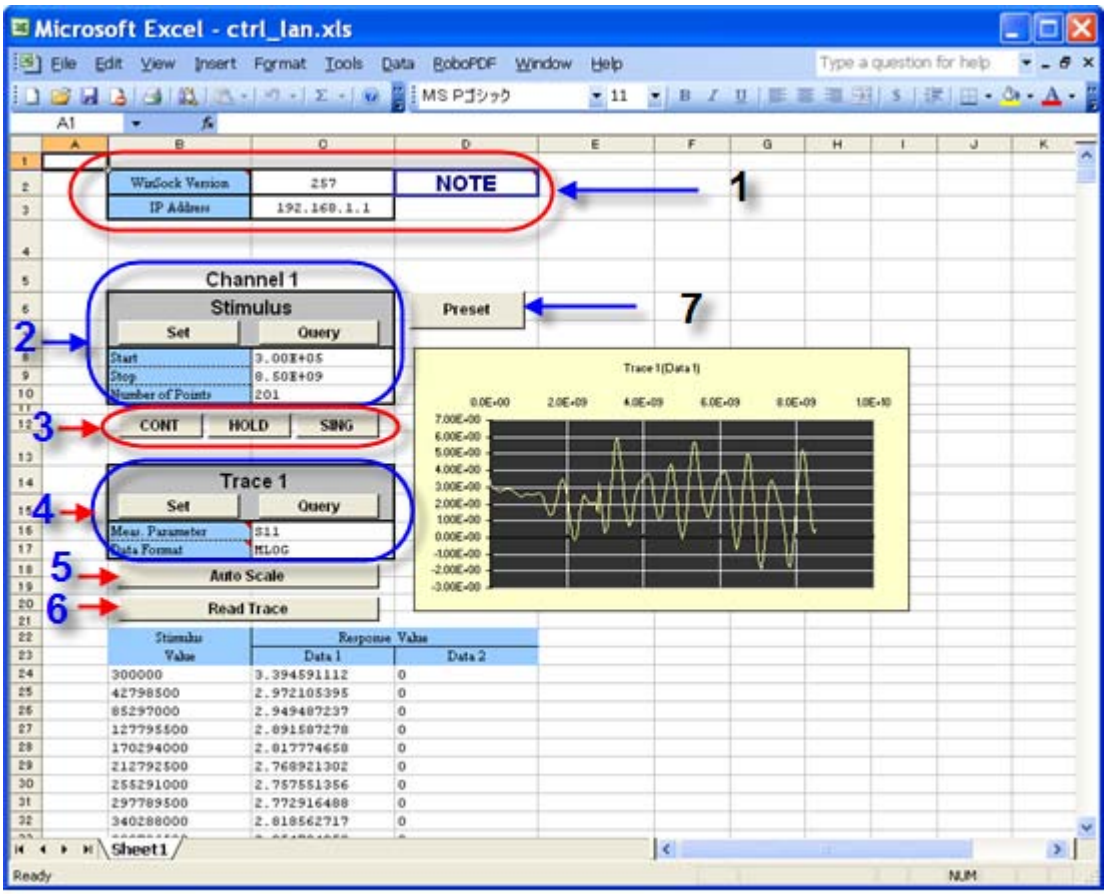
Overview

This section explains how to control the E5071C by using WinSock API in the Windows environment.

Sample Program in Excel VBA

Opening **ctrl_lan.xls** in Microsoft Excel displays the screen shown in the figure below.

ctrl_lan.xls



e5071c192

For how to use each element in ctrl_lan.xls, refer to the following description.

1. Enter the version number of WinSock API in the cell to the right side of "Winsock Version." The version number is obtained by multiplying 256 by the major version and then adding the minor version. For

example, when the version of your Winsock API is 1.1, the version number is obtained as follows: 256*1+1=257. Enter the IP address of the E5071C in the cell to the right side of "IP Address." This VBA macro will not work properly without the correct values in these two cells.

2. In part 2, the sweep range (start and stop points) and the number of measurement points are set. Click **Set** to execute the setting operation as specified with the setting table, while clicking the button labeled "Query" retrieves the current settings of the E5071C.
3. Part 3 is dedicated to setting the trigger mode.
4. Part 4 sets the measurement parameters and data format for trace 1 in channel 1. Click **Set** to execute the setting operation as specified with the setting table, while clicking the button labeled "Query" retrieves the current settings of the E5071C.
5. In part 5, click **Auto Scale** to execute auto scaling for trace 1 in channel 1.
6. Click **Read Trace** in part 6 to retrieve the formatted data of trace 1 in channel 1. The data is displayed in tabular and graphical formats.
7. Click **Preset** to execute the presetting operation.

Description of operation in VBA macro

This section describes the operation of the VBA macro, focusing on the part related to control with WinSock API.

In order to use WinSock API, you must declare functions and define variables with a definition file of WinSock API, as shown in Definition file of WinSock API.

Definition file of WinSock API

```
'This is the Winsock API definition file for Visual Basic


'Setup the variable type 'hostent' for the WSAStartup command
Type Hostent
h_name As Long
h_aliases As Long
h_addrtype As String * 2
h_length As String * 2
h_addr_list As Long
End Type
Public Const SZHOSTENT = 16
```

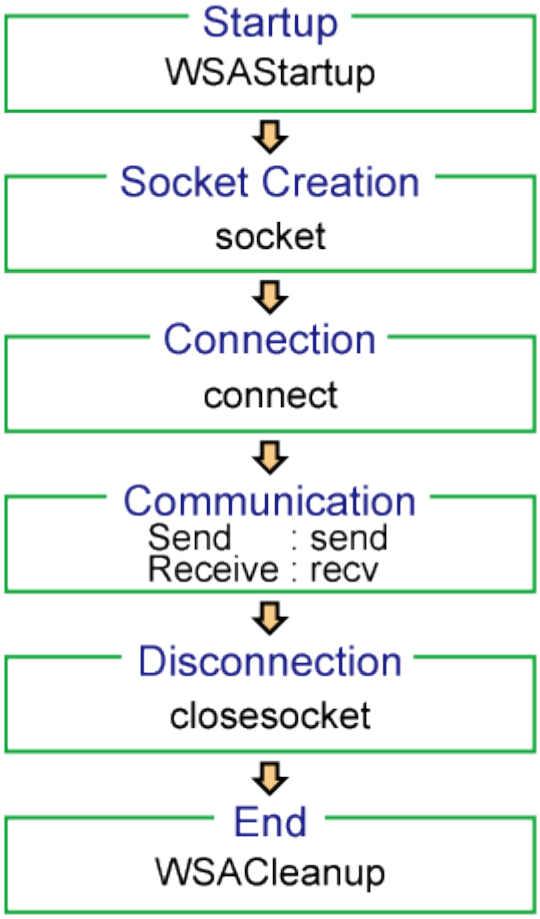
```
'Set the Internet address type to a long integer (32-bit)
Type in_addr
s_addr As Long
End Type

'A note to those familiar with the C header file for Winsock
'Visual Basic does not permit a user-defined variable type
'to be used as a return structure. In the case of the
'variable definition below, sin_addr must
'be declared as a long integer rather than the user-defined
'variable type of in_addr.
Type sockaddr_in
sin_family As Integer
sin_port As Integer
sin_addr As Long
sin_zero As String * 8
End Type

Public Const WSADESCRIPTION_LEN = 256
Public Const WSASYS_STATUS_LEN = 128
Public Const WSA_DescriptionSize = WSADESCRIPTION_LEN + 1
Public Const WSA_SysStatusSize = WSASYS_STATUS_LEN + 1

'Setup the structure for the information returned from
'the WSAStartup() function.
Type WSAData
wVersion As Integer
wHighVersion As Integer
szDescription As String * WSA_DescriptionSize
szSystemStatus As String * WSA_SysStatusSize
iMaxSockets As Integer
iMaxUdpDg As Integer
lpVendorInfo As String * 200
End Type

'Define socket return codes
```

```
Public Const INVALID_SOCKET = &HFFFF
Public Const SOCKET_ERROR = -1

'Define socket types
Public Const SOCK_STREAM = 1 'Stream socket
Public Const SOCK_DGRAM = 2 'Datagram socket
Public Const SOCK_RAW = 3 'Raw data socket
Public Const SOCK_RDM = 4 'Reliable Delivery socket
Public Const SOCK_SEQPACKET = 5 'Sequenced Packet socket

'Define address families
Public Const AF_UNSPEC = 0 'unspecified
Public Const AF_UNIX = 1 'local to host (pipes, portals)
Public Const AF_INET = 2 'internetwork: UDP, TCP, etc.
Public Const AF_IMPLINK = 3 'arpanet imp addresses
Public Const AF_PUP = 4 'pup protocols: e.g. BSP
Public Const AF_CHAOS = 5 'mit CHAOS protocols
Public Const AF_NS = 6 'XEROX NS protocols
Public Const AF_ISO = 7 'ISO protocols
Public Const AF_OSI = AF_ISO 'OSI is ISO
Public Const AF_ECMA = 8 'european computer manufacturers
Public Const AF_DATAKIT = 9 'datakit protocols
Public Const AF_CCITT = 10 'CCITT protocols, X.25 etc
Public Const AF_SNA = 11 'IBM SNA
Public Const AF_DECnet = 12 'DECnet
Public Const AF_DLI = 13 'Direct data link interface
Public Const AF_LAT = 14 'LAT
Public Const AF_HYLINK = 15 'NSC Hyperchannel
Public Const AF_APPLETALK = 16 'AppleTalk
Public Const AF_NETBIOS = 17 'NetBios-style addresses
Public Const AF_MAX = 18 'Maximum # of address families

'Setup sockaddr data type to store Internet addresses
Type sockaddr
sa_family As Integer
sa_data As String * 14
End Type
```

```
Public Const SADDRLEN = 16

'Declare Socket functions

Public Declare Function closesocket Lib "wsock32.dll" (ByVal s As Long) As Long

Public Declare Function connect Lib "wsock32.dll" (ByVal s As Long, addr As sockaddr_in, ByVal
namelen As Long) As Long

Public Declare Function htons Lib "wsock32.dll" (ByVal hostshort As Long) As Integer

Public Declare Function inet_addr Lib "wsock32.dll" (ByVal cp As String) As Long

Public Declare Function recv Lib "wsock32.dll" (ByVal s As Long, ByVal buf As Any, ByVal buflen As
Long, ByVal flags As Long) As Long

Public Declare Function recvB Lib "wsock32.dll" Alias "recv" (ByVal s As Long, buf As Any, ByVal buflen
As Long, ByVal flags As Long) As Long

Public Declare Function send Lib "wsock32.dll" (ByVal s As Long, buf As Any, ByVal buflen As Long,
ByVal flags As Long) As Long

Public Declare Function socket Lib "wsock32.dll" (ByVal af As Long, ByVal socktype As Long, ByVal
protocol As Long) As Long

Public Declare Function WSAStartup Lib "wsock32.dll" (ByVal wVersionRequired As Long, lpWSAData
As WSAData) As Long

Public Declare Function WSACleanup Lib "wsock32.dll" () As Long

Public Declare Function WSAUnhookBlockingHook Lib "wsock32.dll" () As Long

Public Declare Sub CopyMemory Lib "kernel32" Alias "RtlMoveMemory" (hpvDest As Any, hpvSource As
Any, ByVal cbCopy As Long)
```

The basic control flow with WinSock API is shown in the figure below:

Control flow with WinSock API



e5071c372

The procedures of each step in Control flow with WinSock API are described below.

Startup

The procedure corresponding to Startup is StartIt. StartIt launches and initializes WinSock API with **WSAStartup**, whose version is shown in part 1 of ctrl_ lan.xls. The function WSAStartup should always be used when initiating WinSock. This function takes the version number (input) and launching information (output) as its parameters.

StartIt

```
Sub StartIt()

Dim StartUpInfo As WSAData
'Version 1.1 (1*256 + 1) = 257
```

```
'version 2.0 (2*256 + 0) = 512
'Get WinSock version
Sheets("Sheet1").Select
Range("C2").Select
version = ActiveCell.FormulaR1C1
'Initialize Winsock DLL
x = WSAStartup(version, StartUpInfo)

End Sub
```

Socket Creation and Connection

The procedure for Socket Creation and Connection is OpenSocket. OpenSocket makes a connection to an instrument associated with the IP address specified with the input parameter Hostname. It uses a socket of the port specified with the input parameter PortNumber. Each functional part of OpenSocket is described below.

In (1), the inet_aadr function of WinSock API is used to convert an IP address delimited by "." to an Internet address.

In (2), a new socket is created with the **socket** function of WinSock API and its socket descriptor is obtained. If an error occurs, the control returns to the main program with a message. The socket function takes an address family (input), a socket type (input), and a protocol number (input) as its parameters.

In (3), the socket address is specified. Note that htons, which is used for specifying the port number, is a function of WinSock API. This function converts a 2-byte integer from the Windows byte order (little endian) to the network byte order (big endian).

In (4), a connection to the E5071C is made by using the **connect** function of WinSock API. If an error occurs, the control returns to the main program with a message. The connect function takes a socket descriptor (input), a socket address (input), and the size of the socket address (input) as its parameters.

OpenSocket

```
Function OpenSocket(ByVal Hostname As String, ByVal PortNumber As Intege r) As Integer
Dim I_SocketAddress As sockaddr_in
Dim ipAddress As Long
ipAddress = inet_addr(Hostname) '.....(1)
```

```
'Create a new socket
socketId = socket(AF_INET, SOCK_STREAM, 0) '
If socketId = SOCKET_ERROR Then '
MsgBox ("ERROR: socket = " + Str$(socketId)) '.....(2)
OpenSocket = COMMAND_ERROR '
Exit Function '
End If '

'Open a connection to a server

I_SocketAddress.sin_family = AF_INET '
I_SocketAddress.sin_port = htons(PortNumber) '.....(3)
I_SocketAddress.sin_addr = ipAddress '
I_SocketAddress.sin_zero = String$(8, 0) '

x = connect(socketId, I_SocketAddress, Len(I_SocketAddress)) '
If socketId = SOCKET_ERROR Then '
MsgBox ("ERROR: connect = " + Str$(x)) '..(4)
OpenSocket = COMMAND_ERROR '
Exit Function '
End If '

OpenSocket = socketId

End Function
```

Communication

The procedure corresponding to Communication is SendCommand. SendCommand transmits a message (SCPI command) specified with the input parameter "command" to the E5071C using the **send** function of WinSock API. The send function takes a socket descriptor (input), a message to be transmitted (input), message length (input) and a flag (input) as its parameters.

SendCommand

```
Function SendCommand(ByVal command As String) As Integer
```

```
Dim strSend As String
strSend = command + vbCrLf
count = send(socketId, ByVal strSend, Len(strSend), 0)
If count = SOCKET_ERROR Then
MsgBox ("ERROR: send = " + Str$(count))
SendCommand = COMMAND_ERROR
Exit Function
End If
SendCommand = NO_ERROR

End Function
```

The procedure corresponding to the Receiving part of communication is RecvAscii and other functions. RecvAscii receives a message in ASCII format and stores it in the dataBuf output parameter. Maximum length of the message is specified with the maxLength input parameter. Each functional part of RecvAscii is described below.

In (1), a message (a response to a query for a SCPI command) is received from the E5071C as a series of characters using the **recv** function of WinSock API. If an error occurs, the control returns to the main program with a message. The recv function takes a socket descriptor (input), a message to be received (input), message length (input) and a flag (input) as its parameters.

In (2), it is determined whether each received character is LF (ASCII code: 10). When it is LF, receiving is terminated by adding NULL (ASCII code: 0) to the end of the dataBuf string and the control returns to the main program.

In (3), the number of the last characters that were read out is added to the count value for checking the number of received characters, and the characters are appended to the end of the dataBuf string.

RecvAscii

```
Function RecvAscii(dataBuf As String, ByVal maxLength As Integer) As Integer

Dim c As String * 1
Dim length As Integer
dataBuf = ""
While length < maxLength
DoEvents
count = recv(socketId, c, 1, 0) '
If count < 1 Then '
```

```
RecvAscii = RECV_ERROR '.....(1)
dataBuf = Chr$(0) '
Exit Function '
End If '
If c = Chr$(10) Then '
dataBuf = dataBuf + Chr$(0) '.....(2)
RecvAscii = NO_ERROR '
Exit Function '
End If '
length = length + count '.....(3)
dataBuf = dataBuf + c '
Wend
RecvAscii = RECV_ERROR
End Function
```

Disconnection

The procedure corresponding to Disconnection is CloseConnection. CloseConnection disconnects communication and removes a socket using the **closesocket** function of WinSock API. The closesocket function takes a socket descriptor (input) as its parameter.

c

```
Sub CloseConnection()

x = closesocket(socketId)
If x = SOCKET_ERROR Then
MsgBox ("ERROR: closesocket = " + Str$(x))
Exit Sub
End If

End Sub
End
```

The procedure corresponding to End is EndIt. EndIt disconnects WinSock API using the **WSACleanup** function of WinSock API. The function WSACleanup should always be used when terminating WinSock.

EndIt

```
Sub EndIt()
```

```
'Shutdown Winsock DLL
x = WSACleanup()

End Sub
```

Example of control

The E5071C can be controlled by executing the above procedures in order, following the control flow in Control flow with WinSock API. This is demonstrated by the procedure autoscale (a procedure that is executed when the Auto Scale button is clicked) as described in autoscale.

autoscale

```
Sub autoscale()
'
' auto scaling
'
Call StartIt
Call get_hostname
x = OpenSocket(Hostname$, ScpiPort)
x = SendCommand(":DISP:WIND1:TRAC1:Y:AUTO")
Call CloseConnection
Call EndIt
End Sub
```

When you execute more than one command by connecting and disconnecting a socket for every command, the sequence of execution may change.

Control LCD Update Timing

- Overview
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This sample program is provided in this section where the command processing time is improved by controlling the update timing of the LCD display.

- This sample program correctly runs when the maximum number of channels/traces is set to 9 channels/9 traces.

This program sets the necessary measurement conditions and then turns off the updating of the LCD display. Next, it performs measurement, reads out the result, and updates the screen once. This program repeats this measurement procedure ten times.

See Improving Command Processing Speed for this programming.

Sample Program in HT Basic (cont_upd.htb)

```
10 REAL Trace1(1:201,1:2),Trace2(1:201,1:2)
20 DIM Buff$[9],Img$[30]
30 INTEGER Nop,I
40 !
50 ASSIGN @Agte507x TO 717
60 ASSIGN @Binary TO 717;FORMAT OFF
70 !
80 OUTPUT @Agte507x;":SENS1:SWE:TYPE LIN"
90 OUTPUT @Agte507x;":SENS1:FREQ:CENT 950E6"
100 OUTPUT @Agte507x;":SENS1:FREQ:SPAN 100E6"
110 OUTPUT @Agte507x;":SENS1:SWE:POIN 201"
120 OUTPUT @Agte507x;":TRIG:SOUR BUS"
130 OUTPUT @Agte507x;":INIT1:CONT ON"
140 FOR I=2 TO 9
150 OUTPUT @Agte507x;":INIT"&VAL$(I)&":CONT OFF"
160 NEXT I
170 !
180 OUTPUT @Agte507x;":DISP:SPL D1"
190 OUTPUT @Agte507x;":DISP:WIND1:SPL D1_2"
200 !
210 OUTPUT @Agte507x;":CALC1:PAR:COUN 2"
```

```
220 OUTPUT @Agte507x;":CALC1:PAR1:DEF S21"
230 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
240 OUTPUT @Agte507x;":CALC1:FORM MLOG"
250 OUTPUT @Agte507x;":CALC1:PAR2:DEF S11"
260 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
270 OUTPUT @Agte507x;":CALC1:FORM MLOG"
280 !
290 OUTPUT @Agte507x;":DISP:ENAB OFF"
300 OUTPUT @Agte507x;":FORM:DATA REAL"
310 !
320 FOR I=1 TO 10
330 OUTPUT @Agte507x;":TRIG:SING"
340 OUTPUT @Agte507x;"*OPC?"
350 ENTER @Agte507x;Buff$
360 !
370 ! Read Trace Data
380 !
390 OUTPUT @Agte507x;":CALC1:PAR1:SEL"
400 OUTPUT @Agte507x;":CALC1:DATA:FDAT?"
410 ENTER @Agte507x USING "#,8A";Buff$
420 ENTER @Binary;Trace1(*)
430 ENTER @Agte507x USING "#,1A";Buff$
440 !
450 OUTPUT @Agte507x;":CALC1:PAR2:SEL"
460 OUTPUT @Agte507x;":CALC1:DATA:FDAT?"
470 ENTER @Agte507x USING "#,8A";Buff$
480 ENTER @Binary;Trace2(*)
490 ENTER @Agte507x USING "#,1A";Buff$
500 !
510 ! Update Display
520 !
530 OUTPUT @Agte507x;":DISP:UPD"
540 NEXT I
550 END
```

Description

Lines 50 to 60

Assigns a GPIB address to the I/O pass.

Lines 80 to 110

These lines set the sweep type to linear sweep, the sweep center value to 950 MHz, the sweep span value to 100 MHz, and the number of measurement points to 201.

Lines 120 to 160

These lines set the trigger source to bus trigger, turn on Continuous Activation mode for channel 1, and turn the mode off for channels 2 through 9.

Lines 180 to 190

These lines display the window for channel 1 only and arrange two graphs tiled horizontally.

Lines 210 to 270

These lines set the number of traces for channel 1 to 2, the measurement parameter and its data format for trace 1 to S21 and Log Mag, respectively, and those for trace 2 to S11 and Log Mag, respectively.

Line 290

This line turns Off the updating of the LCD screen.

Line 300

This line sets the data transfer format to binary.

Lines 320 to 540

These lines repeat the following procedure ten times.

Lines 340 to 360: These lines trigger the instrument and wait until the measurement cycle finishes.

Lines 400 to 440: Reads out the formatted data array of trace 1 in channel 1.

Lines 460 to 500: Reads out the formatted data array of trace 2 in channel 1.

Line 540: This line updates the LCD screen once.

Handler Interface

- Overview
- Program Code

Other topics about Sample Programs

Overview

The sample program communicates with an external instrument through the handler I/O port.

This program outputs 5 (sets bit 2 and bit 0 to Low, and the other bits to High) to the port A of the handler I/O port, then waits until the bit 3 of the port C is set to Low.

See Inputting/Outputting Data for this programming.

Program Code

Excel VBA

```
Sub Handler_Click()
    Dim defrm As Long          'Session to Default Resource Manager.
    Dim vi As Long             'Session to instrument.
    Dim Out_Data As Integer    'Decimal value.
    Dim In_Data As Long
    Dim Bit_stat As Integer
    Dim Flag_bit As Integer
    Dim Out_Data_Bin As String
    Dim Ret As Long            'Return value.
    Dim i As Long
    Dim X As Long
    Const TimeOutTime = 40000    'timeout time.
    Out_Data_Bin = "00000101"     'Store the output data on the port A (binaly).
    Flag_bit = 3                  'Bit location (bit 3).

    Call viOpenDefaultRM(defrm)    'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)   'Opens the session to the specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)   'The state of an attribute for the
specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)    'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)   'Clears the all status register.
```

```
    Call viVPrintf(vi, ":CONT:HAND:C:MODE INP" & vbLf, 0)     'Configures the port C to input port.
    Call viVPrintf(vi, ":CONT:HAND:IND:STAT ON" & vbLf, 0)    'Line enable /INDEX signal.
    Call viVPrintf(vi, ":CONT:HAND:RTR:STAT ON" & vbLf, 0)   'Line enable /READY FOR TRIGGER
signal.

    For i = 1 To Len(Out_Data_Bin)    'Convert Out_Dara_Bin to a decimal value.
        If Mid(Out_Data_Bin, Len(Out_Data_Bin) - i + 1, 1) = "1" Then
            X = 2 ^ (i - 1)
            Ret = Ret + X
        End If
    Next i
    Out_Data = Ret       'Sets the decimal value.
    Call viVPrintf(vi, ":CONT:HAND:A " & Ret & vbLf, 0)    'Sets to the port A.

    Call viVPrintf(vi, ":CONT:HAND:C?" & vbLf, 0)   'Outputs data to output port C.
    Call viVScanf(vi, "%t", In_Data)    'Reads data from the port C.

    Call ErrorCheck(vi)      'Checking the error.

    Call viClose(vi)          'Closes the resource manager session.
    Call viClose(defrm)     'Breaks the communication and terminates the VISA system.

    End
End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)    'Reads error message.
    ErrNo = Split(err, ",")  'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)  'Display the message box.
    End If
End Sub
```

HT Basic (handler.htb)

```
10 INTEGER Out_data,In_data,Bit_stat
20 DIM Out_data_bin$[9]
```

```
30 !
40 ASSIGN @Agte507x TO 717
50 !
60 Out_data_bin$="00000101"
70 Flag_bit=3
80 !
90 OUTPUT @Agte507x;":CONT:HAND:C:MODE INP"
100 OUTPUT @Agte507x;":CONT:HAND:IND:STAT ON"
110 OUTPUT @Agte507x;":CONT:HAND:RTR:STAT ON"
120 !
130 Out_data=IVAL(Out_data_bin$,2)
140 OUTPUT @Agte507x;":CONT:HAND:A ";Out_data
150 !
160 REPEAT
170 OUTPUT @Agte507x;":CONT:HAND:C?"
180 ENTER @Agte507x;In_data
190 Bit_stat=BIT(In_data,Flag_bit)
200 UNTIL Bit_stat=1
210 END
```

Controlling E5091A

- Overview
- Sample Program in Excel VBA
- Sample Program in HT Basic

Other topics about Sample Programs

Overview

This program assigns Port 1 of the E5091A to A, Port 2 to T2, Port 3 to R2, and Port 4 to R2 and sets Line 1 and Line 3 of the control line to HIGH.

See Controlling E5091A by Programming for this programming.

Sample Program in Excel VBA

```
Sub e5091ctrl_Click()
    Dim defrm As Long          'Session to Default Resource Manager
    Dim vi As Long             'Session to instrument
    Dim Port1 As String
    Dim Port2 As String
    Dim Port3 As String
    Dim Port4 As String
    Dim Line1 As String
    Dim Line2 As String
    Dim Line3 As String
    Dim Line4 As String
    Dim Line5 As String
    Dim Line6 As String
    Dim Line7 As String
    Dim Line8 As String
    Dim Data_Bin As String
    Dim Data_Dec As Long
    Dim i As Integer
    Dim X As Long
    Dim Model As String

    Const TimeOutTime = 20000

    Model = "E5091_9"                       'Model    E5091A-009

    Port1 = "A"                             'Port1 = A
    Port2 = "T2"                            'Port2 = T2
```

```
    Port3 = "R2"                                    'Port3 = R2
    Port4 = "R2"                                    'Port4 = R2

    Line1 = "1"                                     'Line1 = High
    Line2 = "0"                                     'Line2 = Low
    Line3 = "1"                                     'Line3 = High
    Line4 = "0"                                     'Line4 = Low
    Line5 = "0"                                     'Line5 = Low
    Line6 = "0"                                     'Line6 = Low
    Line7 = "0"                                     'Line7 = Low
    Line8 = "0"                                     'Line8 = Low

    Call viOpenDefaultRM(defrm)                             'Initializes the VISA system.
    Call viOpen(defrm, "GPIB0::17::INSTR", 0, 0, vi)        'Opens the session to the
specified instrument.
    Call viSetAttribute(vi, VI_ATTR_TMO_VALUE, TimeOutTime)         'The state of an attribute
for the specified session.

    Call viVPrintf(vi, "*RST" & vbLf, 0)                    'Presets the setting state of the ENA.
    Call viVPrintf(vi, "*CLS" & vbLf, 0)                    'Clears the all status register.

    Call viVPrintf(vi, ":SENS1:MULT1:NAME " & Model & vbLf, 0)          'Specifies the test set
name.

    Call viVPrintf(vi, ":SENS:MULT:PORT1 " & Port1 & vbLf, 0)       'Sets the port assigned to port 1
to A
    Call viVPrintf(vi, ":SENS:MULT:PORT2 " & Port2 & vbLf, 0)       'Sets the port assigned to port 2
to T2
    Call viVPrintf(vi, ":SENS:MULT:PORT3 " & Port3 & vbLf, 0)       'Sets the port assigned to port 3
to R2
    Call viVPrintf(vi, ":SENS:MULT:PORT4 " & Port4 & vbLf, 0)       'Sets the port assigned to port 4
to R2

    Call ErrorCheck(vi)                                     'Checking the error.

    Data_Bin = Line8 & Line7 & Line6 & Line5 & Line4 & Line3 & Line2 & Line1

    For i = 1 To Len(Data_Bin)                              'Creates a decimal setting value.
        If Mid(Data_Bin, Len(Data_Bin) - i + 1, 1) = "1" Then
```

```
            X = 2 ^ (i - 1)
            Data_Dec = Data_Dec + X
        End If
    Next i
    Call viVPrintf(vi, ":SENS:MULT:OUTP " & CStr(Data_Dec) & vbLf, 0)        'Sets the control line.

    Call viVPrintf(vi, ":SENS:MULT:DISP ON" & vbLf, 0)                       'Turns on the E5091A property
display.
    Call viVPrintf(vi, ":SENS:MULT:STAT ON" & vbLf, 0)                       'Turns on the control of the
E5091A.

    Call ErrorCheck(vi)                                                      'Checking the error.

    Call viClose(vi)                                                        'Closes the resource manager session.
    Call viClose(defrm)                                                      'Breaks the communication and terminates
the VISA system.

    End                                                                     'End
End Sub
Sub ErrorCheck(vi As Long)
    Dim err As String * 50, ErrNo As Variant, Response

    Call viVQueryf(vi, ":SYST:ERR?" & vbLf, "%t", err)                       'Reads error message.
    ErrNo = Split(err, ",")                                                 'Gets the error code.

    If Val(ErrNo(0)) <> 0 Then
        Response = MsgBox(CStr(ErrNo(1)), vbOKOnly)                          'Display the message box.
    End If
End Sub
```

Sample Program in HT Basic (e5091ctr_2.htb)

```
10 DIM Port1$[3],Port2$[3],Port3$[3],Port4$[3],Data_bin$[9]
20 DIM Line1$[3],Line2$[3],Line3$[3],Line4$[3]
30 DIM Line5$[3],Line6$[3],Line7$[3],Line8$[3]
40 DIM Model$[8]
50 INTEGER Data_dec
60 !
70 ASSIGN @Agte507x TO 717
80 !
```

```
90 Model$="E5091_9" ! Model:E5091A-009
100 !
110 Port1$="A" ! Port1: A
120 Port2$="T2" ! Port2: T2
130 Port3$="R2" ! Port3: R2
140 Port4$="R2" ! Port4: R2
150 !
160 Line1$="1" ! Line1: HIGH
170 Line2$="0" ! Line2: Low
180 Line3$="1" ! Line3: HIGH
190 Line4$="0" ! Line4: Low
200 Line5$="0" ! Line5: Low
210 Line6$="0" ! Line6: Low
220 Line7$="0" ! Line7: Low
230 Line8$="0" ! Line8: Low
240 !
250 OUTPUT @Agte507x;":SENS1:MULT1:NAME "&Model$
260 !
270 OUTPUT @Agte507x;":SENS1:MULT1:PORT1 "&Port1$
280 OUTPUT @Agte507x;":SENS1:MULT1:PORT2 "&Port2$
290 OUTPUT @Agte507x;":SENS1:MULT1:PORT3 "&Port3$
300 OUTPUT @Agte507x;":SENS1:MULT1:PORT4 "&Port4$
310 !
320 Data_bin$=Line8$&Line7$&Line6$&Line5$&Line4$&Line3$&Line2$&Line1$
330 Data_dec=IVAL(Data_bin$,2)
340 OUTPUT @Agte507x;":SENS1:MULT1:OUTP ";Data_dec
350 !
360 OUTPUT @Agte507x;":SENS:MULT1:DISP ON"
370 OUTPUT @Agte507x;":SENS:MULT1:STAT ON"
380 !
390 END
```

Built-in VBA Programming

Built-in VBA Programming

- Introduction to VBA Programming
- Operation Basics
- Controlling E5071C
- Controlling Peripherals
- Application Programs
- Complex Operation Library
- Waveform Analysis Library

Other topics about Programming

Introduction to VBA Programming

Introduction to VBA Programming

- Introduction of the E5071C Macro Function
- An Overview of a Control System Based on the Macro Function
- Overview of E5071C COM Object

Introduction of the E5071C Macro Function

- Overview
- Macro Function

Other topics about Introduction to VBA Programming

Overview

The E5071C has a built-in macro function that allows a single instruction to substitute for multiple instructions. You can have the E5071C automatically execute your own macro program that contains a series of VBA (Visual Basic for Application) statements. The macro function allows you to run a variety of applications; you can control not only the E5071C but also various peripherals from your own macro code.

The VBA is based on the VB (Visual Basic) programming language. Although the VBA is similar to the VB, they are not the same. The VBA is decreased some of the VB's features and added characteristic features for each application. The E5071C VBA is added features for controlling the E5071C. For details of difference between the VBA and the VB, refer to Microsoft official guides, and various books on VBA.

For information on the basic operating procedures for the E5071C's VBA, see Operation Basics. This manual is not meant to be an in-depth guide to VBA programming basics and the syntax of VBA functions and commands. Such in-depth information is covered in VBA Help, Microsoft official guides, and various books on VBA.

NOTE The built-in VBA can not control the ENA option TDR application.

Macro Function

The macro function allows you to control the E5071C itself as well as various peripherals. You can do the following:

1. Automate repetitive tasks
   You can use the E5071C's macro function to combine several processes into one. Automating repetitive tasks provides higher efficiency and eliminates human error. Once you have contained repetitive tasks in Sub procedures, you can later call the procedures from other programs, thus allowing effective reuse of programming assets.
2. Implement a user interface
   The E5071C VBA supports user forms that simplify creating a visual user interface. User forms guide users through common tasks such as performing measurement and entering data, without requiring

familiarity with the E5071C, thus minimizing the possibility of human error.

An Overview of a Control System Based on the Macro Function

- Overview
- Implementing a Control System
- Required Equipment
- Control Methods

Other topics about Introduction to VBA Programming

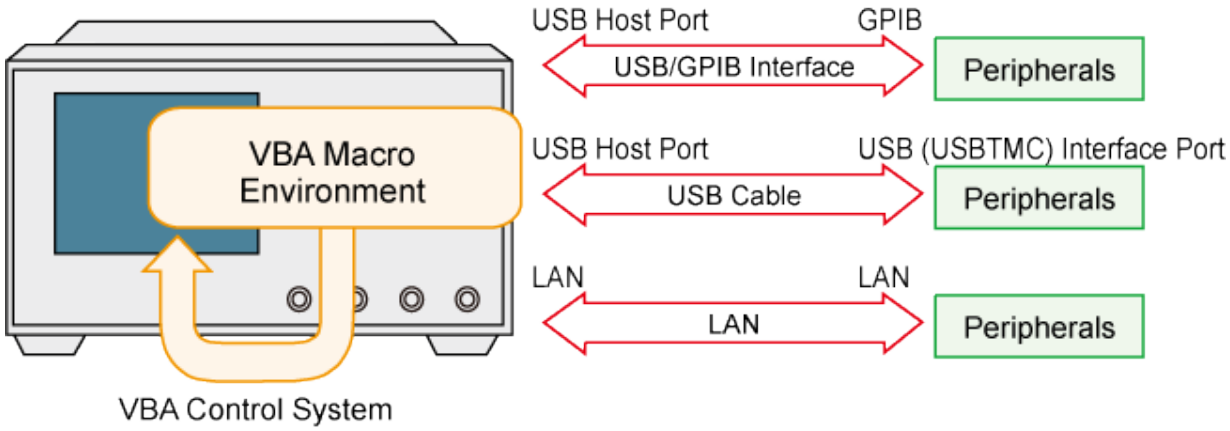
Overview

This section describes how you can use the E5071C's built-in VBA macro function to implement a system that controls the E5071C and peripherals, and what command sets are available for such purposes.

Implementing a Control System

Macro-based control systems are classified into two types: As shown in the following figure, a VBA control system controls the E5071C itself while a VBA remote control system controls peripherals. When you use the macro function to control peripherals, you must connect the E5071C with the peripherals through USB/GPIB interface, USB or LAN, and configure them to communicate over VISA (Virtual Instrument Software Architecture). For information on programming using the VISA library, refer to Programming with VISA.

Configuration example of control system using macro environment



e5071c478

Required Equipment

- E5071C
- Peripherals and/or other instruments that serve your purpose
- USB/GPIB interface, USB Cable, or LAN

Control Methods

The command set you can use differs depending on whether you use the macro function to control the E5071C or a peripheral.

Controlling the E5071C

When you want to control the E5071C itself, you can create a program using COM objects within the E5071C VBA environment. COM objects that come with the E5071C include seven objects specific to the COM interface and COM objects that correspond to SCPI commands.

Controlling a Peripheral

When you want to control a peripheral, you can create a program using VISA library functions within the E5071C VBA environment.

For information on using the VISA library, see Controlling Peripherals. For a complete description of VISA functions, refer to the VISA library's online help.

For information on the GPIB commands available with a particular peripheral, refer to the documentation that comes with the peripheral.

Overview of E5071C COM Object

- Overview
- About COM Object
- Property
- Method
- Event
- Using COM Object to Control E5071C
- Major Control Difference between COM Object and SCPI Command

Other topics about Introduction to VBA Programming

Overview

The E5071C VBA environment provides COM objects that support controlling the E5071C. This section provides an overview of COM objects as well as considerations for using the E5071C's COM objects.

The definitions and specifications of COM are beyond the scope of this guide. Such in-depth information is covered in a variety of books on COM.

About COM Object

When you control the E5071C through the macro function, you can use COM objects as components of your application. The functionality of the E5071C's COM objects is exposed through properties and methods.

Property

A property allows you to read or write a setting or attribute of an object. With the E5071C, you can use properties to set or read the settings of the E5071C.

You can find properties in the list of object types in COM Object Reference.

Method

A method allows you to manipulate an object in a particular way. With the E5071C, you can use methods to perform specific tasks.

You can find methods in the list of object types in COM Object Reference.

Event

An event means an operation from outside that the program can recognize such as clicking a mouse. The E5071C detects events that a specific softkey is pressed using the UserMenu_OnPress(ByVal Key_id As Long) procedure to execute the assigned procedure.

You can find events in the list of object types in COM Object Reference.

Using COM Object to Control E5071C

When you want to control the E5071C, you can use COM objects alone or in conjunction with SCPI commands and the Parse object. The latter method is a little slower than the former method because the Parse object

is used to parse the messages of SCPI commands. For instructions on using the E5071C's VBA Editor to create a program that uses COM objects, refer to Operation Basics of the E5071C's VBA.

Major Control Difference between COM Object and SCPI Command

While the control using SCPI commands allows SRQ (Service Request) interrupts through the status reporting mechanism, the control using COM objects does not support SRQ interrupts. Instead of SRQ interrupts, you can use the WaitOnSRQ object to suspend the program until the E5071C is put into the desired state.

Operation Basics

Operation Basics

- Displaying Visual Basic Editor
- Closing Visual Basic Editor
- Switching to the E5071C Measurement Screen
- Making a Preparation Before Coding
- Coding a VBA Program
- Saving a VBA program
- Loading a VBA Program
- Running a VBA Program
- Stopping a VBA Program
- Errors and Debugging
- Printing Output Values in the Echo Window
- Uses Advanced Techniques
- Using VBA Online Help

Displaying Visual Basic Editor

- Overview
- Initial Screen of Visual Basic Editor

Other topics about Operation Basics

Overview

This section describes how to launch Visual Basic Editor.

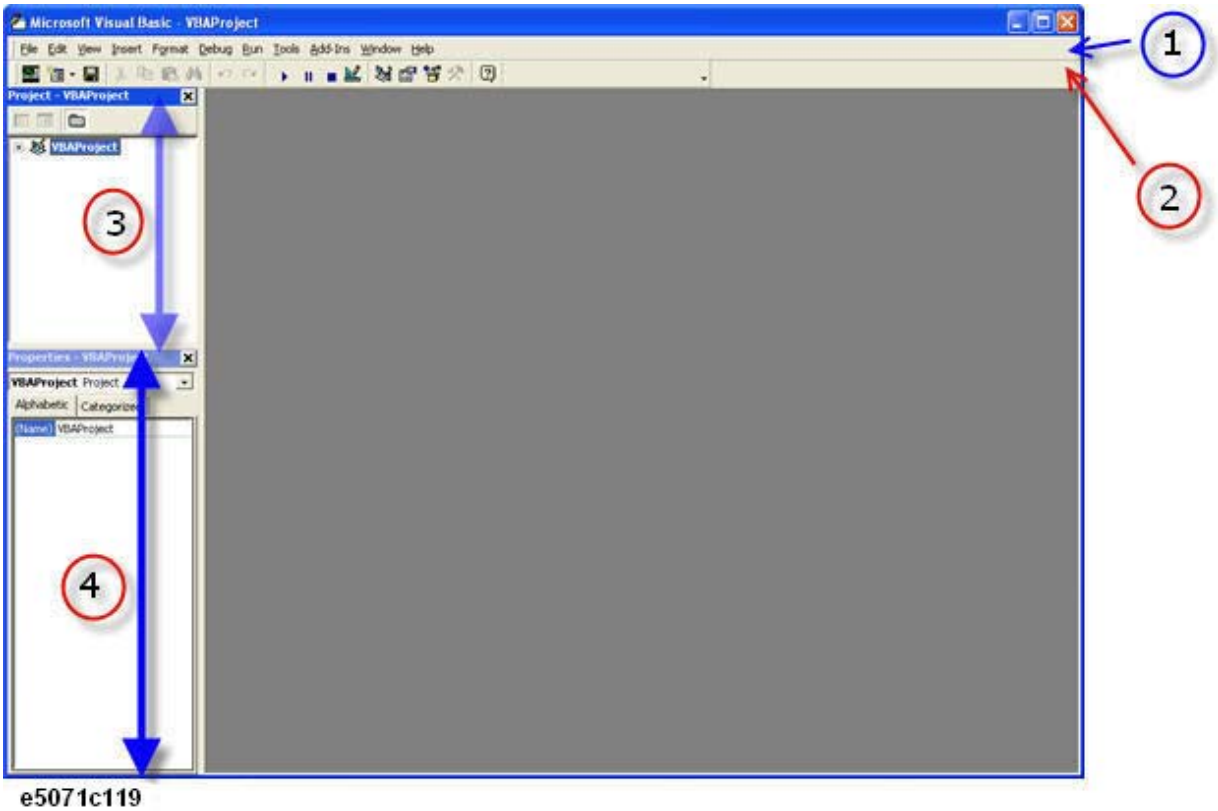
From the E5071C measurement screen, launch Visual Basic Editor using one of the following methods:

1. **Macro Setup** > **VBA Editor**
2. Press **Alt** + **F11** keys on the keyboard.

Initial Screen of Visual Basic Editor

When you launch Visual Basic Editor, it displays the initial screen, which contains a number of windows as shown in the following figure. The initial screen provides the following GUI elements:

Example of Visual Basic Editor initial screen



e5071c119

1. Menu bar
2. Toolbar
3. Project Explorer

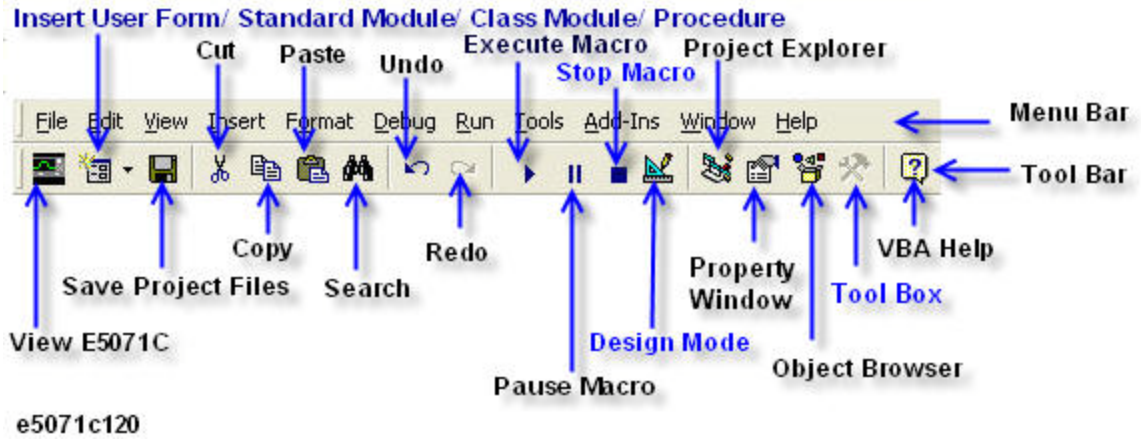
4. Property Window

Menu Bar

Clicking one of the menu labels brings up the corresponding menu. The menu bar can be used as the primary method to navigate through E5071C's VBA environment.

Toolbar

The toolbar provides access to commonly used commands via icon buttons; these commands are a subset of the commands accessible from the menu bar.



Project Explorer

Within the E5071C's VBA environment, you can develop your application as a project that consists of a number of files (modules). Project Explorer shows a list of all files (modules) that make up a project. The list also includes files (modules) created or loaded in Visual Basic Editor. For information on modules, refer to A Project and Three Types of Module.

To display the project explorer, do one of the following:

1. On the **View** menu, click **Project Explorer**.
2. Press **Ctrl** + **R** keys on the keyboard.
3. On the toolbar, click .

Property Window

A property window shows the settings (label, font, color, size, etc.) of a control (such as a command button or text box) placed on the user form. For information on user forms, refer to User Form.

You can also set properties by programming in the code window.

To display the properties window, do one of the following:

1. On the **View** menu, click **Properties Window**.
2. Press **F4** key on the keyboard.
3. On the toolbar, click .

Closing Visual Basic Editor

This section describes how to quit Visual Basic Editor. Close the Visual Basic Editor using either one of the following methods:

- On Visual Basic Editor's **File** menu, click **Close and Return to E5071**.
- Within Visual Basic Editor, press **Alt** + **Q** keys on the keyboard.
- **Macro Setup** > **Close Editor** (E5071C measurement screen)
  - Whenever you launch Visual Basic Editor, it automatically displays the project files you were working with in the previous session. However, once you turn off the power to the E5071C, the project files kept in memory will be lost; therefore, it is strongly recommended to save your VBA programs before you turn off the power.

Other topics about Operation Basics

Switching to the E5071C Measurement Screen

You can switch to the E5071C measurement screen without closing Visual Basic Editor.

- On the **View** menu, click **E5071**.
- Press **Alt** + **F11** keys on the keyboard.
- On the toolbar, click "E5071C" icon.
- Press **Focus** key on the E5071C front panel.

Other topics about Operation Basics

Making a Preparation Before Coding

- A Project and Three Type of Modules
- Displaying a Code Window

Other topics about Operation Basics

A Project and Three Type of Modules

Project Explorer displays a list of files (modules) that are used in the E5071C VBA. This section describes a project composed of a number of files (modules) and three types of modules ("user form", "standard," and "class"). Each type of module serves its own purposes as described below.

Project

When you develop an application within the E5071C's VBA environment, you use a number of VBA program files (modules), and manage them as one project. The project is saved with the file extension ".vba".

User Form

A user form contains controls such as buttons and text boxes. You can code event-driven procedures that are invoked when a particular event occurs on a particular control, thereby creating a user interface. The user form is saved with the file extension ".frm".

Standard module

A standard module contains a collection of one or more procedures (subprograms enclosed between Sub and End Sub). One typical use of a standard module is to contain shared subroutines and globally called functions. The standard module is saved with the file extension ".bas".

Class Module

A class module contains both data and procedures and acts as one object. Once you have created a class module that serves as an object, you can create any number of instances of that object by naming each instance as an object variable. While each procedure must be unique in a standard module, you can have multiple instances of an object created through a class module. The class module is saved with the file extension ".cls".

Displaying a Code Window

The code windows appear on the Visual Basic Editor by inserting the modules in a project. You can do coding (programming) on this code windows practically.

The E5071C's VBA environment does not allow you to manage multiple projects. When the current project is existing in the Visual Basic Editor by

loading the saved project file, you can replace the current project with a new project by the following method from the E5071C measurement screen.

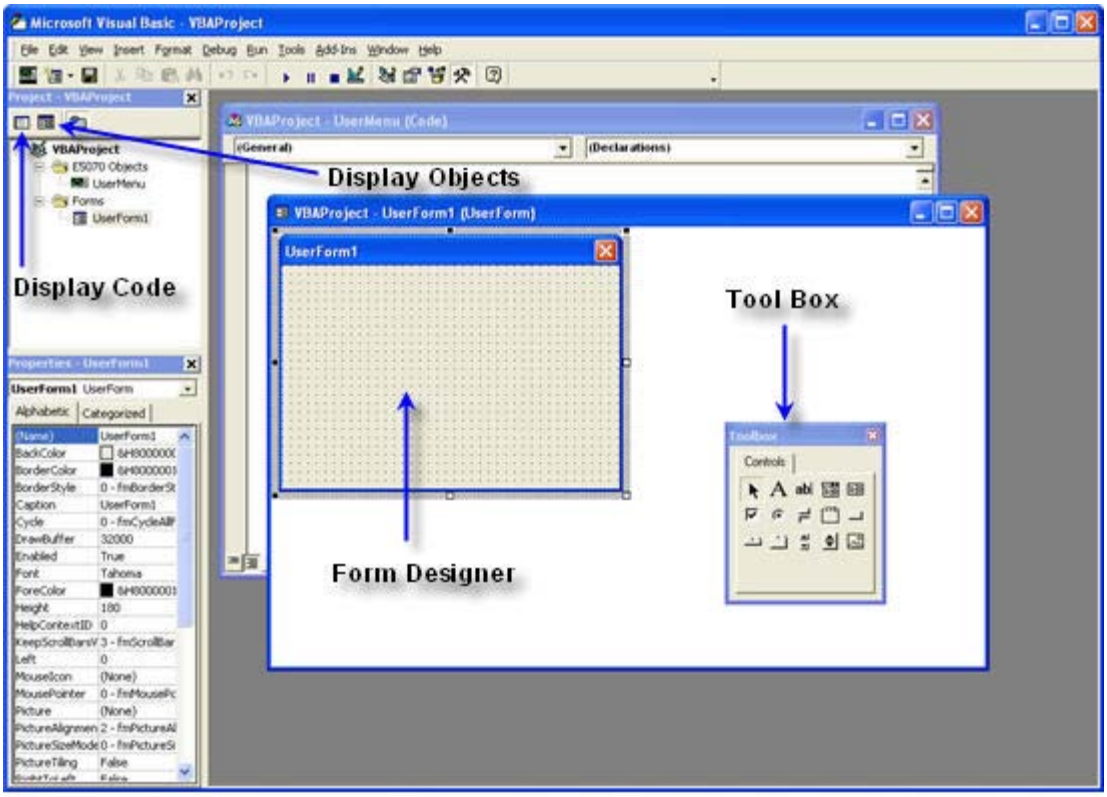
1. **Macro Setup** > **New Project**
2. When you replace the current project with a new project, the message whether or not the current project is saved may appear. If you want to save the project, click **Yes** button to display a dialog box for saving. For saving the project, see Saving a Project.

Inserting the User Form

Within Visual Basic Editor, do one of the following to add a user form to your project.

1. On the **Insert** menu, click **UserForm**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **UserForm**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert** > **UserForm**.
4. Adding a user form does not automatically open the code window for that user form. To open the code window, click the "Display Code" icon on Project Explorer in the following figure or double-click a control placed on the user form.

Adding a user form

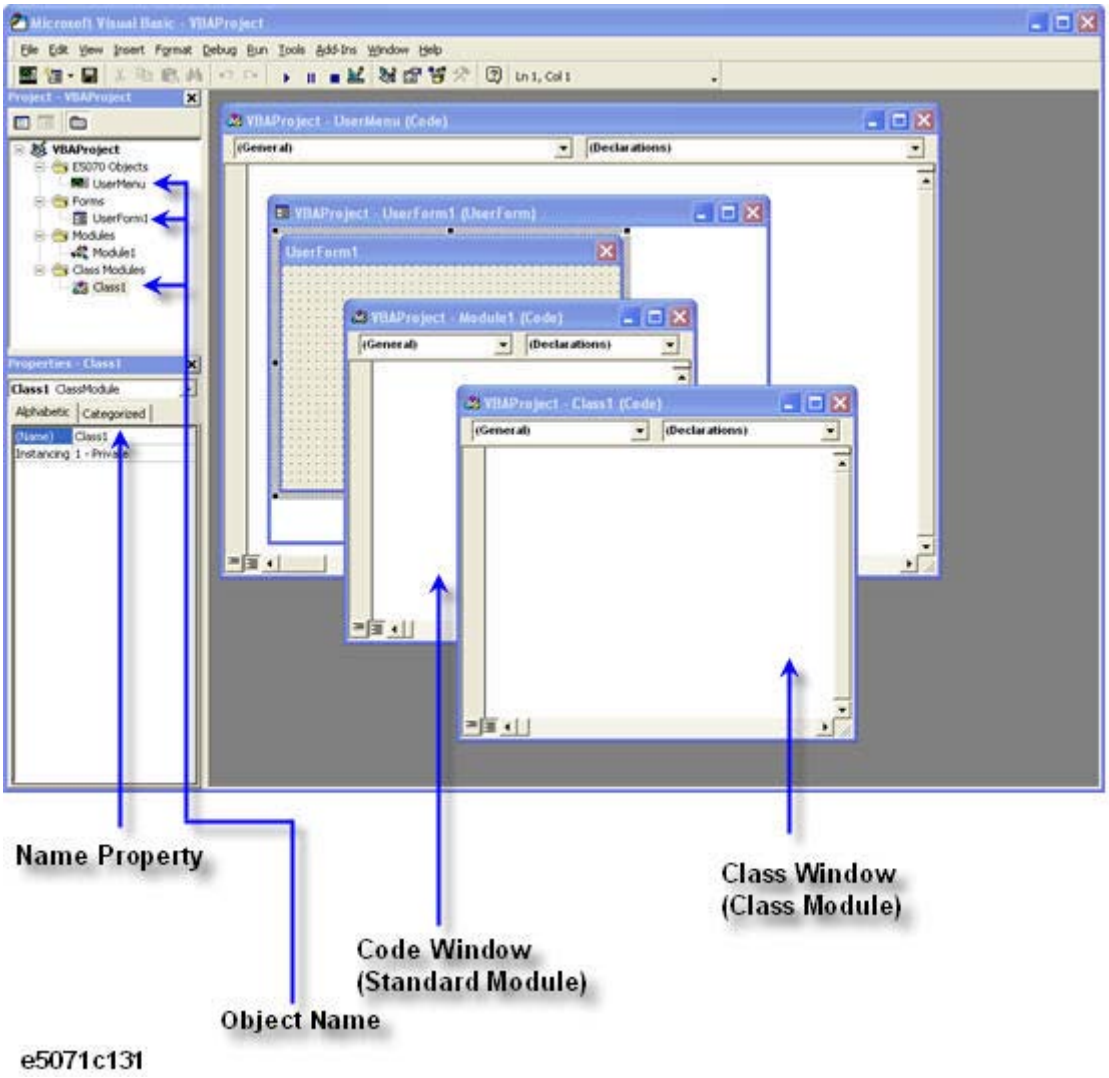


Inserting the Standard Module

Within Visual Basic Editor, do one of the following to add a standard module to your project.

1. On the **Insert** menu, click **Module**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **Module**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert** > **Module**.

Adding a standard module/class module



Inserting the Class Module

Within Visual Basic Editor, do one of the following to add a class module to your project.

1. On the **Insert** menu, click **ClassModule**.
2. On the toolbar, click "Insert User Form/Standard Module/Class Module/Procedure" icon, and click **ClassModule**.
3. In Project Explorer, right-click the "VBAProject" icon, and click **Insert** > **ClassModule**.

Deleting Modules

You can delete any unnecessary module from the project within Visual Basic Editor. The following procedure assumes that you want to delete a class module named "Class1".

1. In Project Explorer, click the "Class1" class module under the "Class Modules" icon to highlight it.
2. Delete the "Class1" class module using one of the following methods:
    a. On the **File** menu, click **Remove Class1...**.
    b. Click the right mouse button, and click **Remove Class1...**.
3. When you are prompted to confirm whether to export (save) "Class1", click **No**. Alternatively, you can click **Yes** if you want to save the module.

Coding a VBA Program

- Overview
- User Interface Elements of a Code Window
- Creating a Simple VBA Program
- Auto-complete Feature

Other topics about Operation Basics

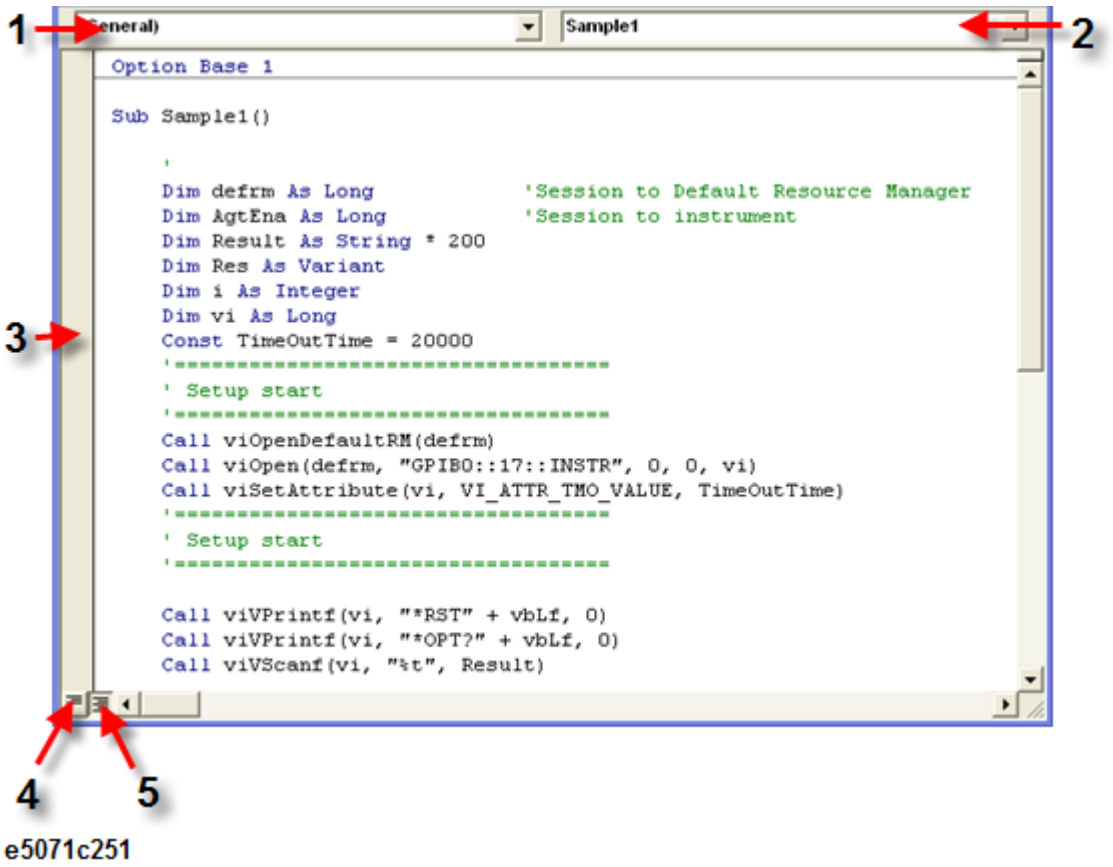
Overview

This section provides descriptive information on the user interface elements of a code window that lets you code a VBA program, and walks through a sample program (procedure) that finds the maximum value contained in an array so you can gain insight into how to create your own programs.

User Interface Elements of a Code Window

A code window is where you code a VBA program. When you are working with a user form, you can open the code window for that user form by double-clicking a control (such as a button or text box) placed on the form. Similarly, when you are working with a standard or class module, you can open the code window associated with that module by double-clicking the module's icon in Project Explorer.

Code window for a standard module



e5071c251

1. Object box

Provides a list of objects currently used within the code window.

2. Procedure box

Provides a list of procedures that reside within the code window. When you are working with a user form, this provides a list of events (actions such as click or double-click).

3. Margin indicator bar

Primarily intended for use when debugging a program.

4. Show Procedure button

Displays only the procedure at the cursor position.

5. Show Module button

Displays the entire program contained in the code window.

Creating a Simple VBA Program

This section walks through a sample program that finds the maximum value contained in an array while breaking down the code into a number of blocks and describing what they do. Line numbers are added for description purpose only, and do not appear in the actual program source code.

Sample program that finds the maximum value contained in an array

```
10| Option Explicit
20|
30| Sub Maximum()
40|
50| Dim q As Variant
60| Dim x(100) As Integer
70| Dim i As Integer, n As Integer
80| Dim Start As Integer, Last As Integer, Num As Integer
90| Dim Maximum As Integer
100|
110| ' Defining the array
120| q = Array(7, -2, 3, -20, 15, -6, 27, -12, 9, -5, 18, 23, _
130| 9, -16, 22, 0)
140|
150| Start = LBound(q)
160| Last = UBound(q)
170| Num = Last - Start + 1
180|
190| For i = Start To Last
200| x(i) = q(i)
210| Next i
220|
230| Maximum = x(Start)
240|
250| For n = Start + 1 To Last
260| If x(n) > Maximum Then Maximum = x(n)
270| Next n
280|
290| MsgBox Maximum
300|
310| End Sub
```

Let us break down the code into a number of blocks and see what they do.

Line 10

This instruction mandates explicit declaration of variables.

Lines 30 to 310

The code enclosed between Sub Maximum() and End Sub will be executed within the E5071C's macro environment. Thus enclosed code is called a procedure. In this example, "Maximum" is the procedure name.

Lines 50 to 90

These lines declare data types of variables using Dim statements. A statement is the minimum instruction unit based on the syntax. The sample program declares the variable "q" as Variant, and the variables "x(100)", "i", "n", "Start", "Last", "Num", and "Maximum" as Integer. For a complete list of statements and data types supported by VBA, see VBA Online Help.

Line 110

Any text preceded by a comment indicator (') is treated as a comment.

Lines 120 to 130

These lines use VBA's Array function to initialize the array. The q() array contains elements delimited with commas in the ascending order of index numbers (zero-based). A combination of a space and underscore (_) is used to continue the statement across two or more lines.

Line 150

Stores the starting index number of the q array into the Start variable.

Line 160

Stores the last index number of the q array into the Last variable.

Line 170

Stores the number of elements in the q array into the Num variable.

Lines 190 to 210 and Lines 250 to 270

The code within each For ...Next statement is iterated until the counter reaches the specific number.

Line 200

Stores the contents of the q array (Variant) into the x variable (Integer).

Line 230

Uses the first element of the x array as the tentative maximum value.

Line 260

Compares the tentative maximum value with each of elements that follow; if an element is larger than the tentative maximum value, then that element is used as the tentative maximum value.

Line 290

Uses a message box function to display the maximum value. For a complete list of functions supported by VBA, see VBA Online Help.

- The above sample program consists of a single procedure contained in a single module. However, when you deal with procedures and variables across multiple modules, you should be aware of the scope of variables and procedures.

Auto-complete Feature

When you use COM objects in Visual Basic Editor, the editor's auto-complete feature allows you to easily type in keywords without misspelling them.

The following procedure assumes that you are entering the SCPI.INITiate(Ch).CONTinuous object.

1. In a standard module, type **sub main** and press the **Enter** key. **End Sub** is automatically added.
2. Typing **scpi** followed by a dot (.) brings up a list of classes under the SCPI class.
3. Typing **in** automatically moves focus to **INITiate** in the list box.
4. Typing **(** brings up a list of indexes.
5. Typing **1).** brings up a list of classes under the INITiate class.
6. Typing **c** automatically moves focus to **CONTinuous** in the list box.
7. Typing **=** brings up a list box for setting a Boolean value (**True/False**).
8. Typing **t** automatically moves focus to **True**.
9. Pressing the **Enter** key completes the statement: SCPI.INITiate(1).CONTinuous = True.

Saving a VBA program

- Overview
- Saving a Project
- Saving a Module (Exporting)

Other topics about Operation Basics

Overview

You can save VBA programs either as one complete project or on a module by module basis.

Saving a Project

When you opt to save your program as one complete project, you can have the files (modules) making up the project into a single package. A project is saved as a .vba file. You can save your program to a project file using one of the following two methods:

Saving a Project from Visual Basic Editor

1. Open the Save As dialog box by doing one of the following:
   - On the **File** menu, click **Save xxx.VBA**. "**xxx**" represents the file name.
   - On the toolbar, click "Save Project File" icon.
   - Press **Ctrl** + **S** keys on the keyboard.
2. The Save As dialog box appears. Specify the file name and location (drive or folder) and click **Save**.

E5071C Saving a Project from the E5071C Measurement Screen

1. Display the E5071C measurement screen following the instructions given in Switching to the E5071C Measurement Screen.
2. Open the Save As dialog box using the following key sequence:
   - **Macro Setup** > **Save Project**
3. The **Save As dialog box** appears. Specify the file name and location (drive or folder) and click Save.

Saving a Module (Exporting)

Alternatively, you can save each module (user form, standard, or class) of your VBA program individually. To save a module, you must use Visual Basic Editor. User forms are saved as .frm files, standard modules as .bas files, and class modules as .cls files.

a. In Project Explorer, click the file name that appears under the desired module icon to highlight it.

b. Open the Export File dialog box by doing one of the following:
    - On the **File** menu, click **Export File...**.
    - Click the right mouse button, and click **Export File...**.
    - Press **Ctrl** + **E** keys on the keyboard.
c. The **Export File dialog box** appears. Specify the file name and location (drive or folder) and click **Save**.

Loading a VBA Program

- Overview
- Loading a Project

Other topics about Operation Basics

Overview

Once you have saved a project or module file, you can load it later whenever necessary.

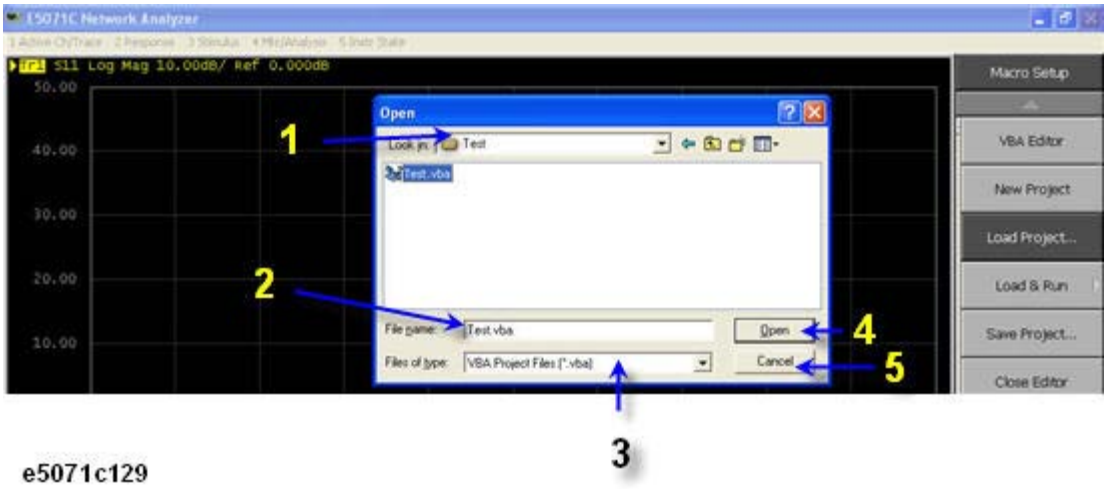
Loading a Project

You can load a saved project file either from the E5071C measurement screen or by specifying that the project file be automatically loaded when the power is turned on.

Loading a Project from the E5071C Measurement Screen

1. Press **Macro Setup** key, then click **Load Project**.
    - When the another project has already been loaded on the Visual Basic Editor, the message whether or not the current project is saved may appear. If you want to save the project, click **Yes** button to display a dialog box for saving. For saving the project, see Saving a Project.
2. The Open dialog box appears. Specify the file name and location (drive or folder) of the file you want to load and click **Open**.

Open dialog box



e5071c129

The Open dialog box has the following user interface elements:

1. **Look in**: Specify the location (drive or folder) where the project resides.
2. **File name**: Specify the file name of the project you want to load.

3. **Files of type**: Select the type of the file you want load. Normally, you should select **VBA Project Files [*.vba]**.
4. **Open**: Clicking this button loads the project.
5. **Cancel**: Clicking this button closes the Open dialog box and brings you back to the main screen.

Automatically Loading a Project at Power-On

Once you have saved a project file that satisfies the following conditions, the project will be automatically loaded whenever the power is turned ON.

Auto-loaded project	Conditions
Directory where the project resides.	**A:** or **D:**
Project file name	autoload.vba

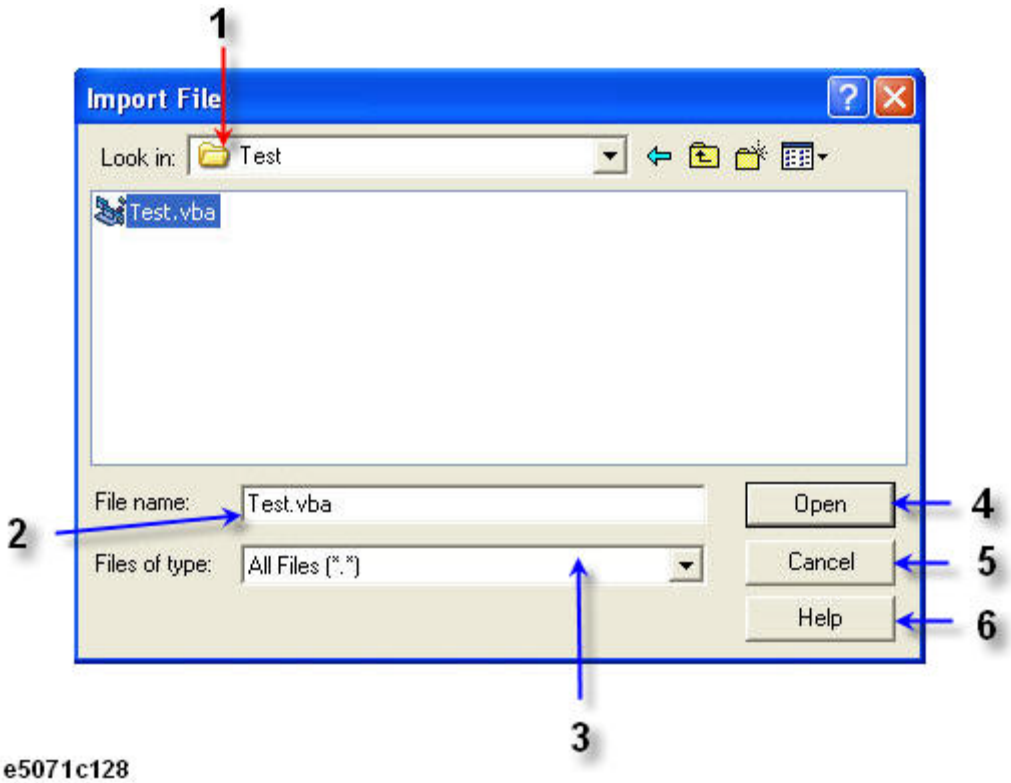
If there is the file named "autoload.vba" in both the A drive and the D drive, the file in the A drive is used.

Loading a Module (Importing)

To load a saved module into a project, you must use Visual Basic Editor.

1. In Project Explorer, click the file name that appears under the desired module icon to highlight it.
2. Open the Import File dialog box by doing one of the following:
   - On the **File** menu, click **Import File...**.
   - In Project Explorer, right-click the "VBAProject" icon, and click **Import File...**.
   - Press **Ctrl** + **M** keys on the keyboard.
3. The Import File dialog box appears. Specify the file name and location (drive or folder) of the file (module) you want to load and click **Open**.
4. The Import File dialog box has the following user interface elements:

Import File dialog box



e5071c128

The Import File dialog box has the following user interface elements:

1. **Look in**: Specify the location (drive or folder) where the module resides.
2. **File name**: Specify the file name of the module you want to load.
3. **Files of type**: Select the type of the file you want load. Normally, you should select **VB Files [*.frm,*.bas,*.cls]**.
4. **Open**: Clicking this button loads the module.
5. **Cancel**: Clicking this button closes the Import File dialog box and brings you back to the main screen.
6. **Help**: Clicking this button brings up VBA Online Help.

Running a VBA Program

- Overview
- Running a Previously Loaded VBA Program
- Running a Program from the_E5071C_Measurement_Screen
- Loading and Executing Programs in Batch Process

Other topics about Operation Basics

Overview

The E5071C provides 2 methods to execute a VBA program: executing a program that you previously loaded and loading and executing a program in a batch process. The execution status of the VBA program is indicated in the instrument status bar, as shown in the following figure. "Run" indicates that the program is running while "Stop" indicates that the program is stopped.

Instrument status bar indicating the status of the VBA program

e5071c132

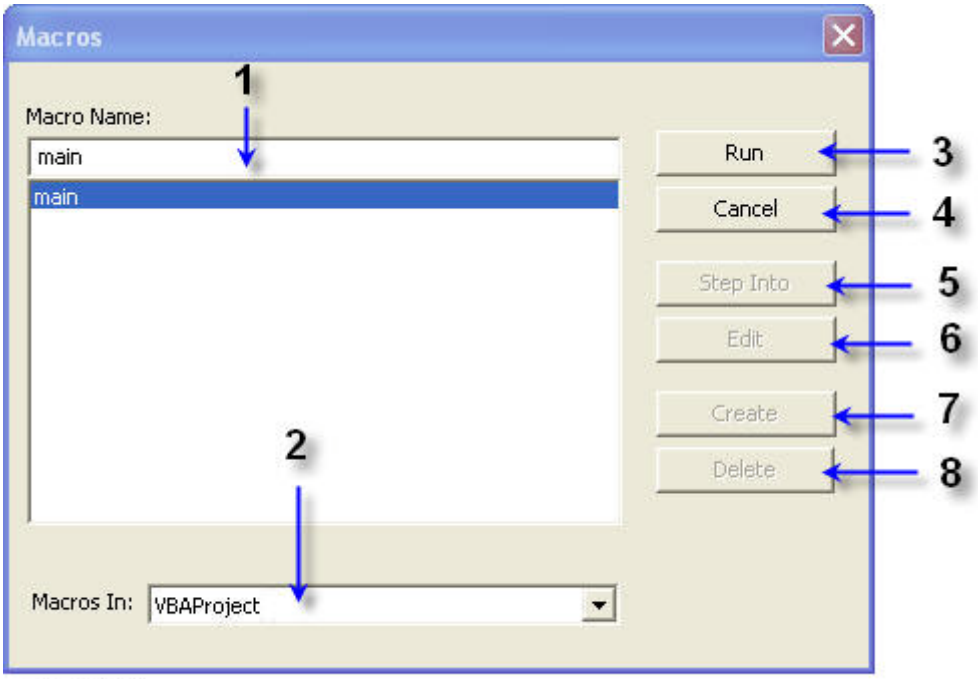
Running a Previously Loaded VBA Program

Running a Program from Visual Basic Editor

The E5071C allows you to run a previous loaded VBA program using one of the four methods listed below.

1. Open the Macros dialog box by doing either one of the following:
   - On the **Run** menu, click **Run Macro**.
   - On the **Tools** menu, click **Macros...**.
   - On the toolbar, click "Run Macro" icon.
   - Press **F5** key on the keyboard.
     1. Doing the above steps with the cursor positioned within a procedure in the code window immediately runs the program without displaying the Macros dialog box.
2. In the Macros dialog box, select the VBA program (procedure name) you want to run, and click the **Run** button.

Macros dialog box



e5071c133

The Macros dialog box has the following user interface elements:

1. **Macro Name**: Select the VBA program (procedure name) you want to run from the list box so its name appears here.
2. **Macro In**: Specify the project that contains the VBA program you want to run. Normally, use the default.
3. **Run**: Clicking this button runs the selected VBA program (procedure).
4. **Cancel**: Clicking this button closes the Macros dialog box and brings you back to the main screen.
5. **Step Into**: Clicking this button brings up Visual Basic Editor and put it into step-in mode, where the selected VBA program is run step by step. This mode is primarily intended for use when debugging a VBA program. For more information on step-in mode, see Debug Toolbar.
6. **Edit**: Displays the code of the selected VBA program. You can use this for re-editing your code.
7. **Create**: This button is normally dimmed.
8. **Delete**: Clicking this button deletes the selected VBA program. Take care not to inadvertently delete your VBA program before saving it.

The Macros dialog provides access to subprograms (procedures enclosed between Sub and End Sub) created in a standard module.

Running a Program from the E5071C Measurement Screen

The E5071C allows you to run a program from E5071C screen using one of the four methods listed below.

1. Display the E5071C measurement screen following the instructions given in Switching to the E5071C Measurement Screen.
2. Run the VBA program (procedure) using the following key sequence:

- **Macro Setup** > **Select Macro - Module xxx**
  where "**Module**" is the object name (Name property shown in the property window) and "**xxx**" is the procedure name.
- Press the **Macro Run** key on the E5071C front panel. For a program to be run from the measurement screen, its procedure name must be "Main" (subprogram enclosed between Sub Main() and End Sub), and its object name (Name property as displayed in the property window) must be "Module1".
  - When you are working with the E5071C measurement screen, the E5071C's macro environment only provides access to those VBA programs that are created as subprograms (enclosed between Sub and End Sub) in a standard module.

Loading and Executing Programs in Batch Process

This section describes how to load and execute a program (VBA project) in a batch process by pressing the softkey corresponding to the program name.

1. Save the VBA program (VBA project file) into the following folder.
   D:\VBA
   - This feature is available only for programs saved in **D:\VBA**. This feature is not available for programs saved in subfolders of **D:\VBA**.
   - When copying a VBA program to D:\VBA from another folder, copy all the files necessary to execute the program to appropriate folders. When copying a factory-installed VBA program into D:\VBA, choose only its VBA project file.
2. Press **Macro Seup** key.
3. Click **Load & Run**.
4. Press the softkey corresponding to the VBA project file name of the program you want to execute. The pressed VBA project is loaded and the program whose procedure name is set to "Main" (subprogram enclosed between Sub Main() and End Sub) and whose object name (Name property as displayed in the property window) is set to "**Module1**" is executed.

5. There is no limit to the number of VBA project files that can be saved in **D:\VBA**. However, the maximum number of programs that can be displayed as softkeys is 50.

- File names of the VBA projects saved in **D:\VBA** are displayed as softkeys in alphabetical order.
- The maximum number of characters that can be displayed in a softkey is 12. If a file name has 13 or more characters, "..." is added to the 12th character from the beginning of the program name and displayed. In this case a .vba extension is omitted.

Stopping a VBA Program

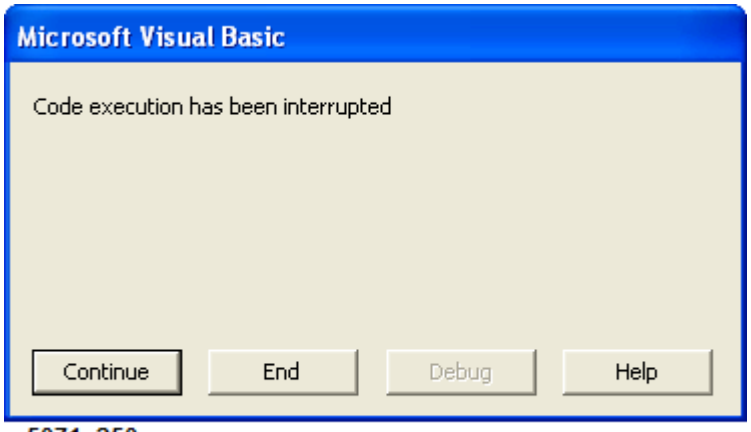
- Stopping a Procedure
- Abruptly Terminating a VBA Program

Other topics about Operation Basics

Stopping a Procedure

This section describes how to break a procedure during the execution of a VBA program.

1. To break the running VBA program, do one of the following:
   - On the **Run** menu, click **Break**.
   - On the toolbar, click "Break Macro" icon.
   - Press **Ctrl** + **Break** keys on the keyboard.
   - **Macro Setup** > **Stop** (E5071C measurement screen)
   - Press **Macro Break** key on the E5071C front panel.
2. A dialog box is displayed through forced interrupts, and the program is suspended.



e5071c250

Select one of the following:

- **Continue**: Resumes the execution of the program.
- **End**: Terminates the VBA program.
- **Debug**: Displays a run-time error.
- **Help**: Brings up VBA Online Help.

Abruptly Terminating a VBA Program

This section describes how to abruptly terminate a running procedure. When abruptly terminating the VBA program by the below methods, the "Program interrupted" message is shown in the instrument status bar on the bottom of the LCD display.

To terminate the running VBA program, do one of the following:

- On the **Run** menu, click **Reset**.
- On the toolbar, click "Reset Macro" icon.
- Insert an *End* statement into your code.

Errors and Debugging

- Type of Errors
- Using a Debug Tool

Other topics about Operation Basics

Type of Errors

Errors in VBA programs are classified into the following two types:

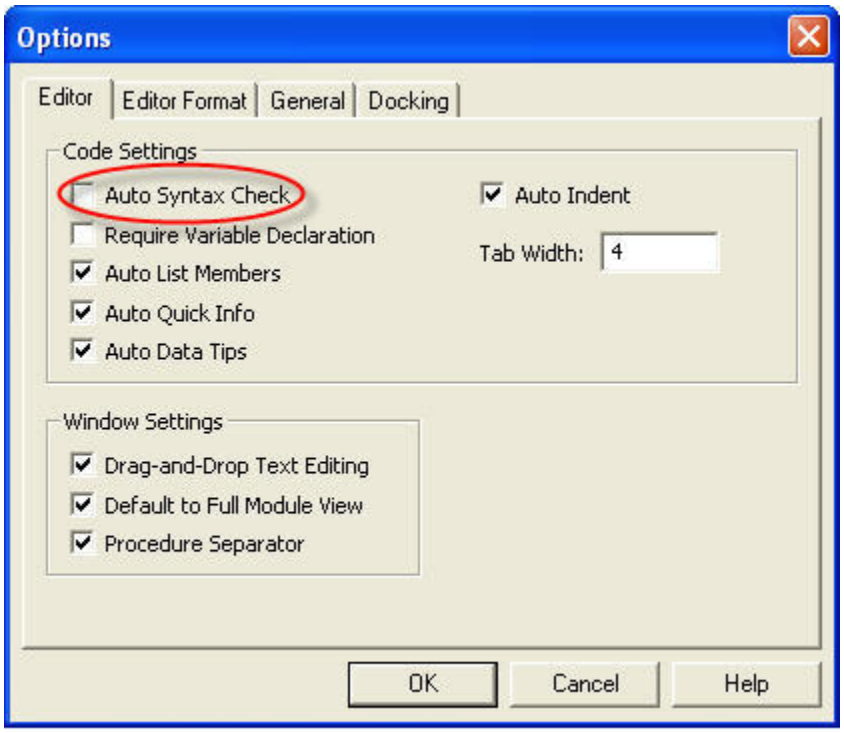
Syntax errors

A syntax error is generated when Visual Basic Editor detects an invalid statement that violates the Visual Basic syntax rules. For example, misspelled keywords generate syntax errors. An error dialog box appears that indicates the error message, and highlight the invalid statement in red. To get detailed information on the error, click the **HELP** button in the error dialog box to display the help topic on the error. You cannot run the macro until you correct the syntax error.

The E5071C VBA environment is by default configured to automatically check for syntax errors, but you can disable the auto syntax check feature using the following steps:

1. On the **Tools** menu, click **Options...**.

2. On the **Editor** tab, clear the **Auto Syntax Check** check box.



e5071c121

3. Click the **OK** button.

Run-time Errors

A run-time error is generated when a VBA program attempts to execute an invalid statement at run time. When a run-time error is generated, the program is stopped at the invalid statement, and an error dialog box appears. You can terminate the program by clicking the END button in the error dialog box. Also, you can click the DEBUG button in the error dialog box to identify the statement that caused the error. In this case, the statement in question is highlighted in yellow.

- Some run-time errors occur under particular conditions, even though a program run without occurring the errors under normal conditions. For example, the "Target value not found" error that occurs when a program that analyzes the results using the Marker Bandwidth Search feature fail to perform bandwidth search because the marker is not in the appropriate position, the "Ecal module not in RF path" error that occurs when a program that performs calibrations using a ECal module fail to measure the calibration data because the ECal module is not appropriately connected to test ports, and so on.

Using a Debug Tool

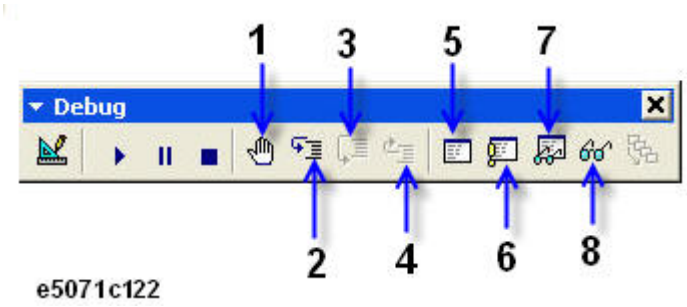
The E5071C's VBA environment provides a variety of debug tools that help you identify logical errors. Detailed information on using the debug tools is covered in VBA Online Help and books on VBA.

Debug Toolbar

The debug toolbar provides tool buttons that allow you to easily access various debug tools.

On the **View** menu, click **Toolbars** > **Debug**.

Debug toolbar



1. Set/clear break points (keyboard **F9**): Puts a break point at the cursor position or clears an existing break point.
2. Step-in (keyboard: **F8**): Runs the VBA program step by step. If the current program contains a call to another procedure, that procedure is also run step by step.
3. Step-over (keyboard: **Shift** + **F8**): Runs the VBA program step by step. If the current program contains a call to another procedure, that procedure is run as one line.
4. Step-out (keyboard: **Ctrl** + **Shift** + **F8**): Executes the remaining lines of the function where the execution point is currently placed.
5. Local window: Opens the local window that shows the current values of local variables.
6. Immediate window (keyboard: **Ctrl** + **G**): Opens the immediate window that evaluates entered values of variables or expressions.
7. Watch window: Opens the watch window that displays the current value of a specified expression.
8. Quick window (keyboard: **Shift** + **F9**): Displays the current value of a specified expression in a dialog box.

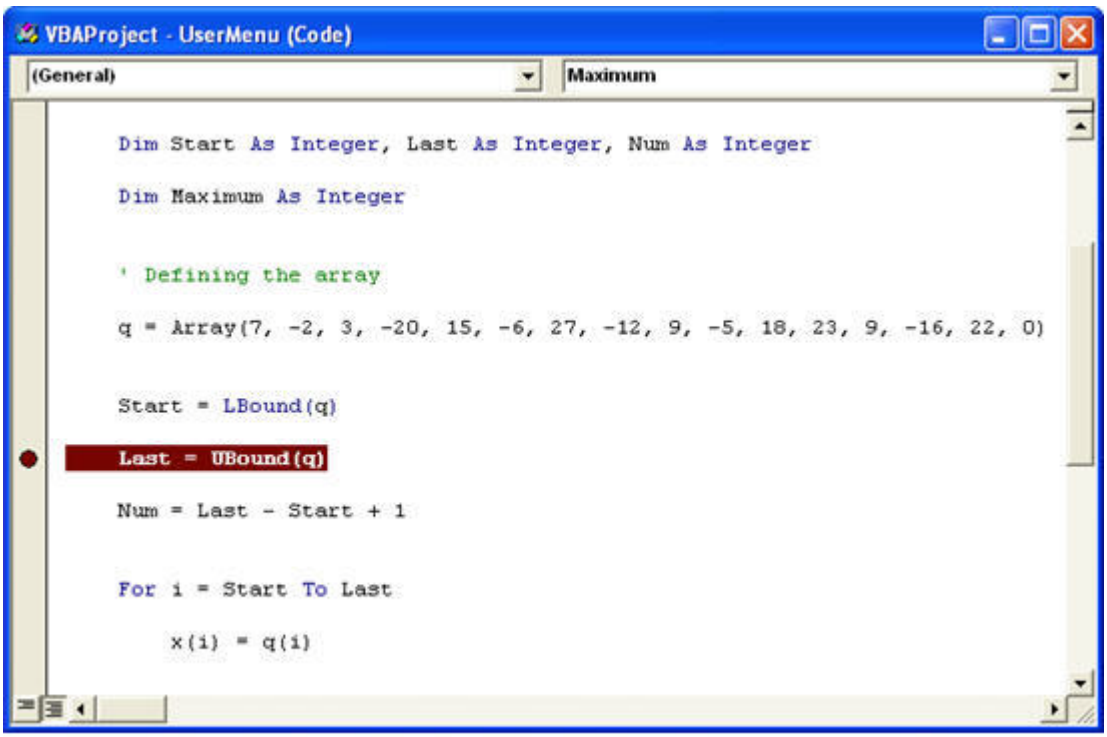
Setting a Break Point

By placing a break point at a particular statement in a VBA program, you can automatically suspend the program when it is executed to that statement.

When you put a break point at a line, the line is highlighted in umber as shown in the following figure. To set a break point, do one of the following:

- Place the cursor at the desired line of code, and click the "Set/clear break points" button on the debug toolbar.
- Click anywhere in the margin indicator bar of the code window.

Setting a break point



e5071c123

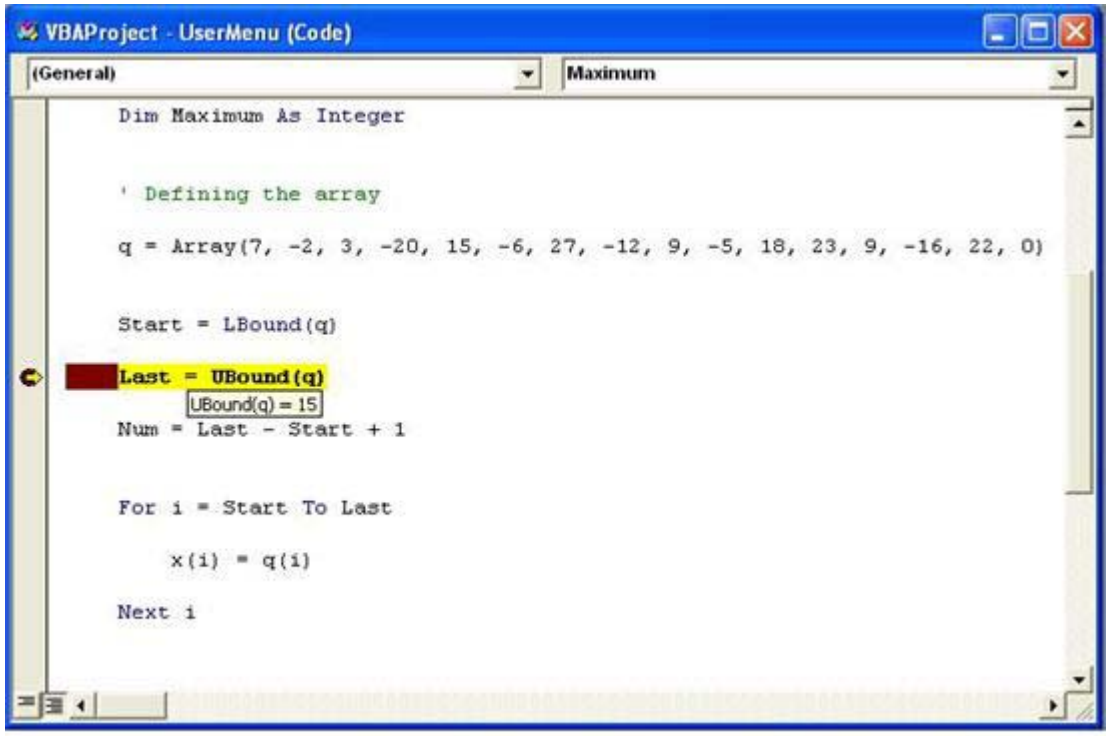
Monitoring Variable or Property Values

With your VBA program suspended, you can use the following debug tool to monitor variables or properties. To do this, you must set a break point, run the VBA program, and suspend it.

Data Hint

When you point to the variable or expression of interest, Data Hint shows the current value as shown in the following figure.

Data Hint



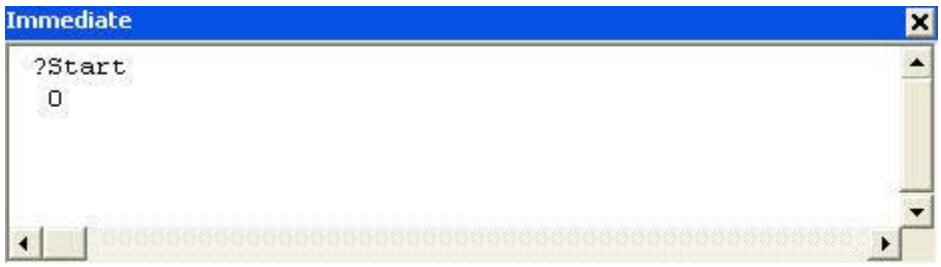
e5071c127

Immediate Window

To display the immediate window, click on the debug toolbar.

In the immediate window, enter a question mark (?) followed by the variable or expression whose value you want to check, and press the Enter key, as shown in the following figure. The current value appears in the line that follows.

Immediate window

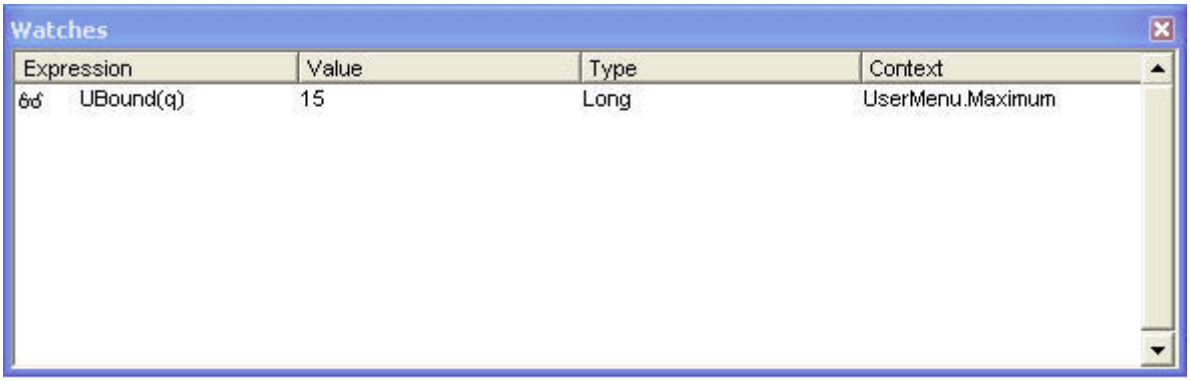


e5071c124

Watch Window

To display the watch window, click the "Watch Window" button on the debug toolbar.

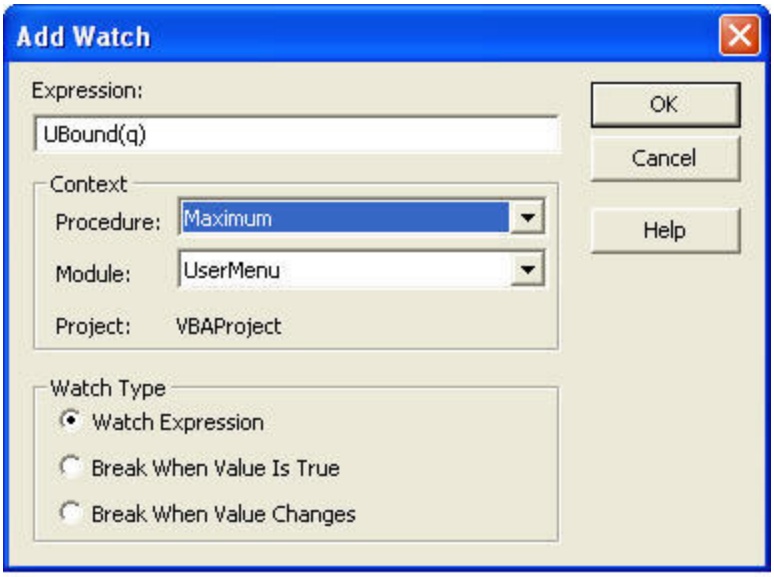
Watch window



e5071c126

1. On the **Debug** menu, click **Add Watch...** to open the Add Watch dialog box.
2. As shown in the following figure, you can specify an expression of interest as a watch expression to always monitor its value.
3. Click the **OK** button.

Add Watch dialog box



e5071c125

Quick Watch

In the code window, select a variable or expression whose value you want to watch. On the debug toolbar, click the "Quick Watch" button to open the Quick Watch dialog box. The dialog box displays the current value of your specified variable or expression.

Also, you can click the **Add** button in the Quick Watch dialog box to specify the current expression as a watch expression.

Printing Output Values in the Echo Window

- Overview
- Entering Values Output to Echo Window
- Opening Echo Window
- Clearing Values Output from Echo Window

Other topics about Operation Basics

Overview

The echo window, which appears in the lower part of the E5071C measurement screen, can be used to display a message or the return value (data) of an object.

Entering Values Output to Echo Window

You can use the COM objects listed below to enter values output to the echo window.

- ECHO
- SCPI.DISPlay.ECHO.DATA

Opening Echo Window

You can use the COM objects listed below to open the echo window.

- SCPI.DISPlay.TABLe.TYPE
- SCPI.DISPlay.TABLe.STATe

Alternatively, you can also open the echo window using the following key sequence:

Macro Setup > **Echo Window (ON)**

Clearing Values Output from Echo Window

You can use the COM object shown below to clear values output from the echo window.

- SCPI.DISPlay.ECHO.CLEar

Alternatively, you can also clear values output from the echo window using the following key sequence:

Macro Setup > **Clear Echo**

Uses Advanced Techniques

- Accessing a List of E5071C COM Objects
- Using Automatic Library References

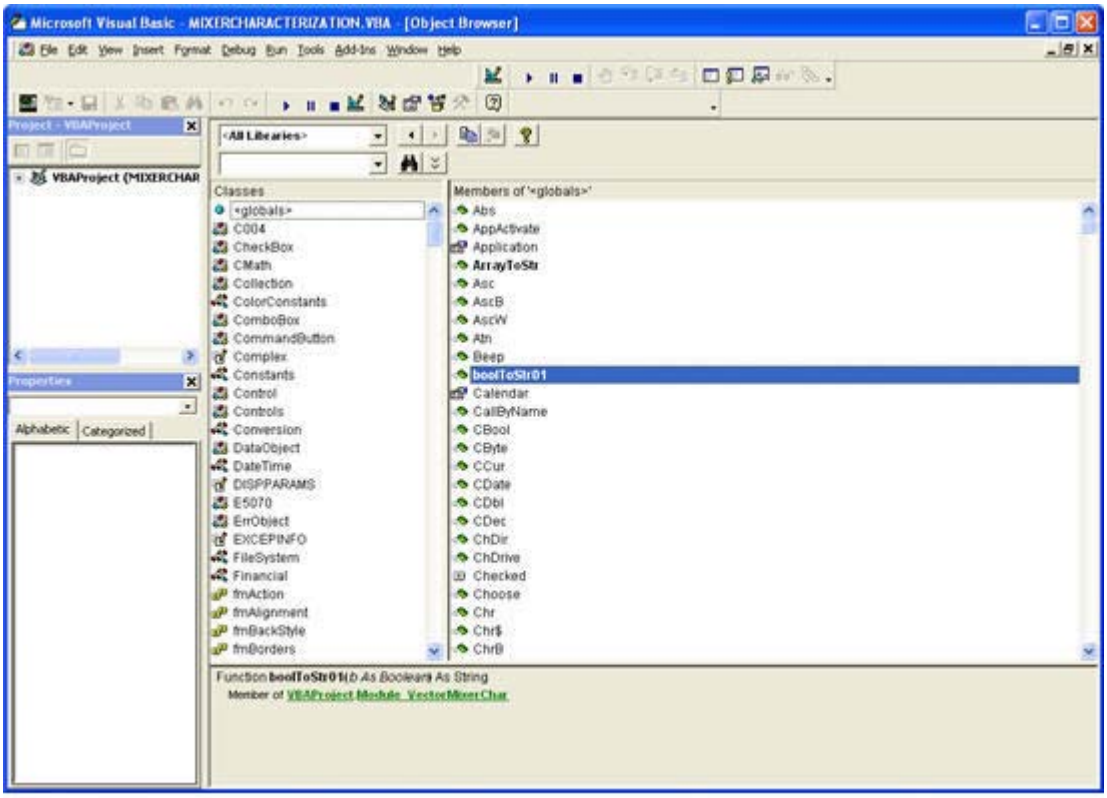
Other topics about Operation Basics

Accessing a List of E5071C COM Objects

The E5071C VBA environment provides COM objects that support controlling the E5071C. When you are developing a program using E5071C COM objects, you can access a list of E5071C COM objects by opening Object Browser within Visual Basic Editor.

1. To open Object Browser, do one of the following:
   - On the **View** menu, click **Object Browser**.
   - On the toolbar, click "Object Browser" icon.
2. Select **E5071Lib** from the Project/Library box to display the E5071C library as shown in the following figure.
3. There are some COM objects NOT used in controlling with E5071C VBA in the list of the E5071C COM objects displayed on the Object Browser. The COM objects NOT used in controlling with E5071C VBA are not described in the COM object reference.

How to use Object Browser



e5071c134

Using Automatic Library References

For libraries that satisfy the following conditions, the library reference will be automatically set whenever a new project is created and loaded (**Macro Setup** > **New Project**).

Automatically referenced libraries	**Conditions**
Directory where the library resides.	**D:\Agilent**
Extensions of libraries	olb, tlb, dll, or ocx

To check the library reference setting, you must use Visual Basic Editor.

Follow these steps to check the library reference setting.

- On the **Tools** menu, click **References...**.
  - A project sets the library reference when the project is created. Therefore, if the existing project is loaded, libraries added after

the development of the project are not automatically set in the library reference.

Using VBA Online Help

- Overview
- Accessing VBA Online Help

Other topics about Operation Basics

Overview

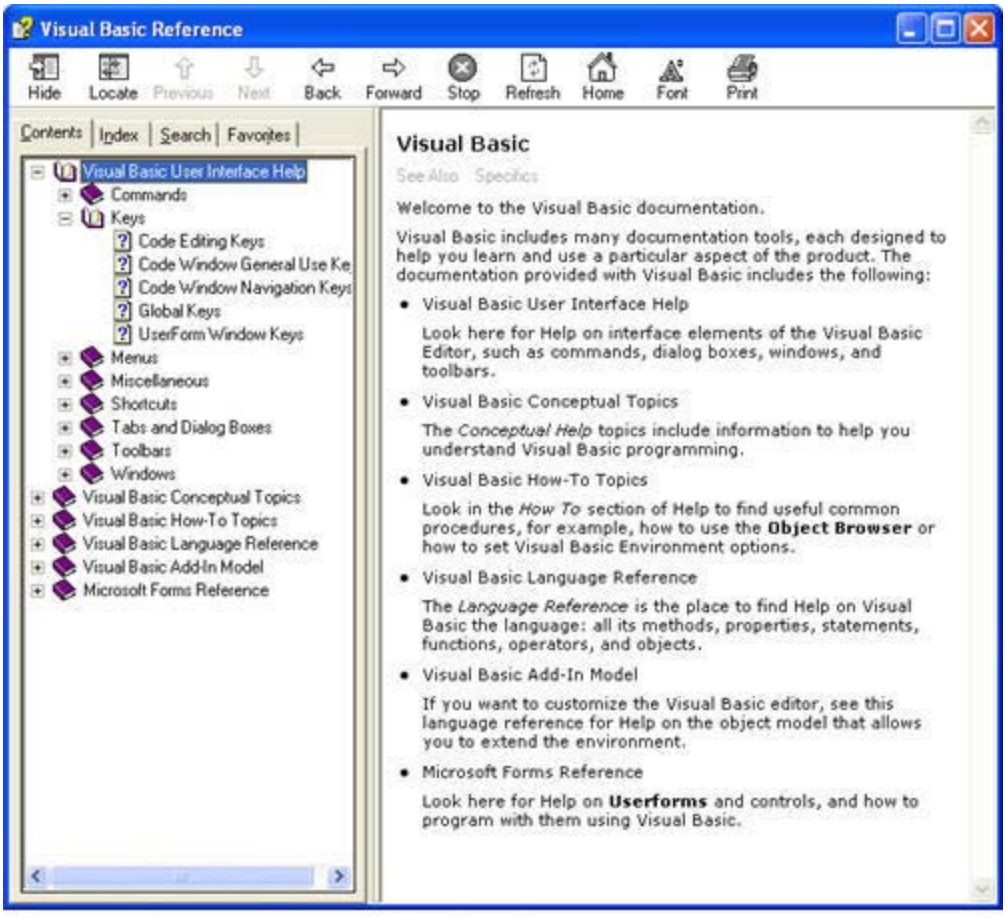
VBA Online Help provides useful topics, such as the VBA terminology or how to use a particular feature. In VBA Online Help, you can find a topic of interest through the Contents or by entering specific keywords.

Accessing VBA Online Help

From Visual Basic Editor, do one of the following to access the VBA Online Help screen.

- On the **Help** menu, click **Microsoft Visual Basic Help**.
- Press **F1** key on the keyboard.
- On the toolbar, click "VBA Help" icon.

VBA Online Help screen



e5071c135

Using the Contents Tab

Clicking the **Contents** tab in the VBA Online Help screen brings up the items listed below. The E5071C VBA Online Help has a hierarchical table of contents. Click an item to expand it, and then find a topic of interest.

- Visual Basic User Interface Help
- Visual Basic Conceptual Topics
- Visual Basic How-To Topics
- Visual Basic Language Reference
- Visual Basic Add-In Model
- Microsoft Forms Reference

When you need information on using Visual Basic Editor, use User Interface Help and How-To Topics as primary sources of information. Format of VBA program is covered in Visual Basic Conceptual Topics. Properties and methods supported by VBA are covered in Visual Basic Language Reference and Visual Basic Add-In Model. Information on using user forms is covered in Microsoft Forms Reference.

Using the Index Tab

In the VBA Online Help screen, click the **Index** tab, and enter a keyword(s) into the text box. For example, you may wish to search for "Sub" or "With" when you are writing your own code.

Looking up a Keyword in the Code within Visual Basic Editor

When you want to know the usage or meaning of a keyword contained in a sample program or some other code, you can quickly access the help topic on that keyword by moving the cursor to the keyword and pressing **F1** key.

Controlling E5071C

Controlling E5071C

- Detecting the End of Measurement
- Reading/Writing Measurement Data
- Executing a Procedure with a Softkey (User Menu Function)

Detecting End of Measurement

- Overview
- Using Status Register
- Using SCPI.TRIGger.SEQuence.SINGle Object

Other topics about Controlling E5071C

Overview

This chapter uses sample programs to demonstrate how to trigger the instrument to start a new measurement cycle and how to detect the end of a measurement cycle. The trigger system is responsible for such tasks as detecting the start of a measurement cycle (triggering) and enabling/disabling measurement on each channel.

You can detect the end of measurement by using either the status register or the SCPI.TRIGger.SEQuence.SINGle object.

Using Status Register

The status of the E5071C can be detected through the status register. If your program is based on SPCI commands, you can use SRQ (Service Request) interrupts to detect the end of measurement.

On the other hand, if your program is based on COM objects, SRQ interrupts are not available; instead, you can use the following object to suspend the program until SRQs are generated upon completion of measurement.

- WaitOnSRQ

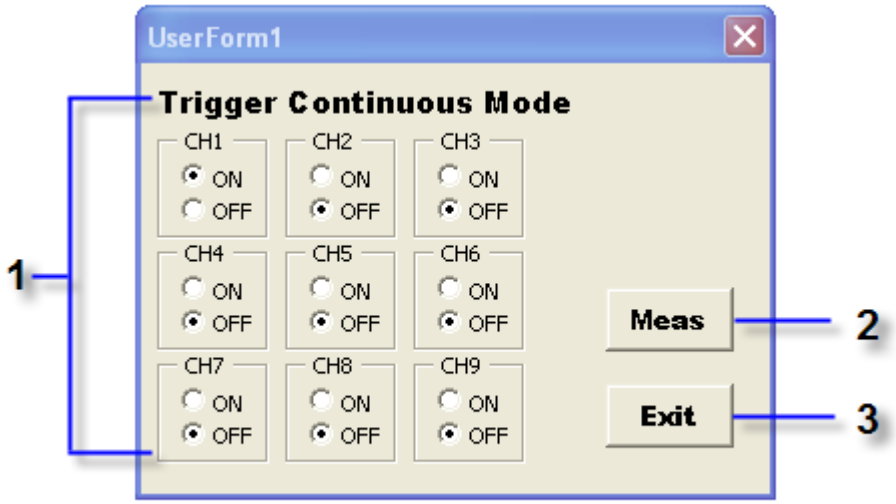
Sample program is available to download from the Agilent Support page, named **meas_srq.vba**, that demonstrates how to use the status register to suspend the program until the end of measurement. This VBA program consists of the following modules:

Object name	Module type	Content
frmSrqMeas	UserForm	Uses the status register to wait for the end of measurement.
mdlSrqMeas	Standard module	Invokes a UserForm.

- This sample program correctly runs when the maximum number of channels/traces is set 9 channels/9 traces.

When you run this VBA program, the following UserForm appears.

The UserForm when running the meas_srq.vba program



e5071c184

For how to use each element, see the following description.

1. The program turns on/off Continuous Activation mode for each channel and determines whether to enable or disable each channel for measurement.
2. The program triggers the instrument to start a new measurement cycle, waits for the end of measurement, and then displays a message. For detail, see the description of the code window.
3. The program exits, and the UserForm disappears.

In Visual Basic Editor, open the UserForm (object name: frmSrqMeas), and double-click the **Meas** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Exit** button on the UserForm (lines 10 to 50)

Line 30

Unloads the UserForm from the memory, and terminates the program.

Procedure called when the user clicks the **Meas** button on the UserForm (lines 70 to 390)

Line 110

Hides the UserForm (object name: frmSrqMeas) from the screen.

Line 130

Displays 9 channel windows.

Line 140

Sets the trigger source to "bus".

Lines 160 to 240

These lines turn on or off Continuous Activation mode for each channel depending on whether the corresponding option buttons are on or off. By default, the mode is turned on for channel 1 only.

Lines 260 to 270

These lines configure the instrument so that operation status event register's bit 4 is set to 1 only when operation status condition register's bit 4 is changed from 1 to 0 (negative transition).

Line 280

Enables the operation status event register's bit 4.

Line 290

Enables the status byte register's bit 7.

Line 300

Clears the status byte register and operation status event register.

Line 310

Triggers the instrument to start a measurement cycle.

Line 320

Verifies that the instrument is in a measurement cycle, and suspends the program until the end of measurement. The time-out is set to 100 seconds (maximum value).

Lines 330 to 350

These lines display a measurement completion message upon detecting the end of measurement.

Line 370

Displays the UserForm (object name :frmSrqMeas) on the screen.

Using SRQs to detect the end of measurement (object name: frmSrqMeas)

```
10| Private Sub cmdExit_Click()
20|
30| Unload Me
40|
50| End Sub
60|
70| Private Sub cmdMeas_Click()
80|
90| Dim Cond As Boolean
100|
110| frmSrqMeas.Hide
120|
130| SCPI.DISPlay.Split = "d123_456_789"
140| SCPI.TRIGger.SEQuence.Source = "bus"
150|
160| SCPI.INITiate(1).CONTinuous = optOn1.Value
170| SCPI.INITiate(2).CONTinuous = optOn2.Value
180| SCPI.INITiate(3).CONTinuous = optOn3.Value
190| SCPI.INITiate(4).CONTinuous = optOn4.Value
200| SCPI.INITiate(5).CONTinuous = optOn5.Value
210| SCPI.INITiate(6).CONTinuous = optOn6.Value
220| SCPI.INITiate(7).CONTinuous = optOn7.Value
230| SCPI.INITiate(8).CONTinuous = optOn8.Value
240| SCPI.INITiate(9).CONTinuous = optOn9.Value
250|
260| SCPI.STATus.OPERation.PTRansition = 0
270| SCPI.STATus.OPERation.NTRansition = 16
280| SCPI.STATus.OPERation.ENABle = 16
290| SCPI.IEEE4882.SRE = 128
300| SCPI.IEEE4882.CLS
310| SCPI.IEEE4882.TRG
320| WaitOnSRQ Cond, 100000
330| If Cond = True Then
340| MsgBox "Measurement Completion"
```

```
350| End If
360|
370| frmSrqMeas.Show
380|
390| End Sub
```

Using the SCPI.TRIGger.SEQuence.SINGle Object

When you trigger the instrument by issuing the SCPI.TRIGger.SEQuence.SINGle object, you can use the SCPI.IEEE4882.OPC object to suspend the program until the end of measurement.

The sample program is available to download from the Agilent Support page, named "meas_sing.vba", that demonstrates how to use the SCPI.TRIGger.SEQuence.SINGle object to suspend the program until the end of measurement. This VBA program consists of the following modules:

Object name	Module type	Content
frmSingMeas	UserForm	Uses the SCPI.TRIGger.SEQuence.SINGle and SCPI.IEEE4882.OPC objects to suspend the program until the end of measurement.
mdlSingMeas	Standard module	Invokes a UserForm.

- This sample program correctly runs when the maximum number of channels/traces is set 9 channels/9 traces.

When you run this VBA program, a Userform appears.

In Visual Basic Editor, open the UserForm (object name:frmSingMeas), and double-click the **Meas** or **Exit** button to bring up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Exit** button on the UserForm (lines 10 to 50)

Line 30

Unloads the UserForm from the memory, and terminates the program.

Procedure called when the user clicks the **Meas** button on the UserForm (lines 70 to 330)

Line 110

Hides the UserForm (object name: frmSingMeas) from the screen.

Line 130

Displays 9 channel windows.

Line 140

Sets the trigger source to "bus".

Lines 160 to 240

These lines turn on or off Continuous Activation mode for each channel depending on whether the corresponding option buttons are on or off. By default, the mode is turned on for channel 1 only.

Line 260

Triggers the instrument to start a measurement cycle.

Line 270

Executes the SCPI.IEEE4882.OPC object to suspend the program until the value of 1 is returned indicating the end of measurement.

Line 290

Displays a measurement completion message.

Line 310

Displays the UserForm (object name: frmSingMeas) on the screen.

Using the SCPI.TRIGger.SEQuence.SINGle object to suspend the program until the end of measurement (object name:frmSingMeas)

```
10| Private Sub cmdExit_Click()
20|
30| Unload Me
40|
50| End Sub
60|
70| Private Sub cmdMeas_Click()
80|
90| Dim Dmy As Long
100|
```

```
110| frmSingMeas.Hide
120|
130| SCPI.DISPlay.Split = "d123_456_789"
140| SCPI.TRIGger.SEQuence.Source = "bus"
150|
160| SCPI.INITiate(1).CONTinuous = optOn1.Value
170| SCPI.INITiate(2).CONTinuous = optOn2.Value
180| SCPI.INITiate(3).CONTinuous = optOn3.Value
190| SCPI.INITiate(4).CONTinuous = optOn4.Value
200| SCPI.INITiate(5).CONTinuous = optOn5.Value
210| SCPI.INITiate(6).CONTinuous = optOn6.Value
220| SCPI.INITiate(7).CONTinuous = optOn7.Value
230| SCPI.INITiate(8).CONTinuous = optOn8.Value
240| SCPI.INITiate(9).CONTinuous = optOn9.Value
250|
260| SCPI.TRIGger.SEQuence.SINGle
270| Dmy = SCPI.IEEE4882.OPC
280|
290| MsgBox "Measurement Completion"
300|
310| frmSingMeas.Show
320|
330| End Sub
```

Reading / Writing Measurement Data

- Overview
- Sample Program

Other topics about Controlling E5071C

Overview

This section describes how to process the E5071C's internal data. You can use these internal data arrays: corrected data arrays, corrected memory arrays, formatted data arrays, formatted memory arrays, and stimulus data arrays. For more information on the internal data arrays, see Internal Data Processing.

To read/write a formatted data array, formatted memory array, corrected data array, or corrected memory array use the following objects:

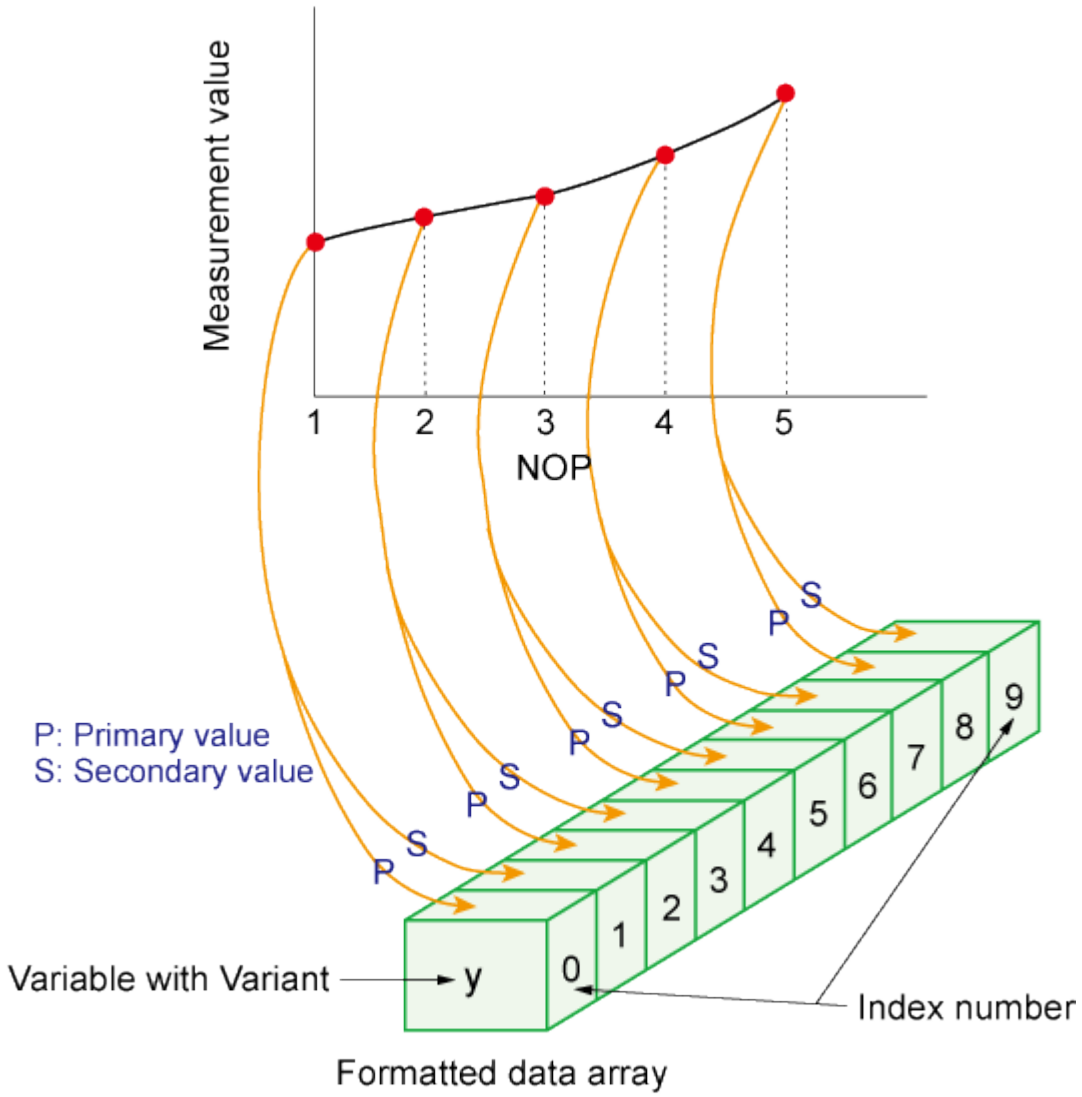
- SCPI.CALCulate(Ch).SELected.DATA.FDATa
- SCPI.CALCulate(Ch).SELected.DATA.FMEMory
- SCPI.CALCulate(Ch).SELected.DATA.SDATa
- SCPI.CALCulate(Ch).SELected.DATA.SMEMory

To read a stimulus data array, use the following objects:

- SCPI.SENSe(Ch).FREQuency.DATA

The E5071C VBA allows you to deal with multiple pieces of data through variables of Variant type. Variant variables can contain any type of data, allowing you to deal with array data without being aware of the number of elements. For example, a formatted data array that includes 5 measurement points is stored as shown in the following figure. Note that a formatted data array always contains 2 data items per measurement point, whichever data format is used. For more information on contained data, see Internal Data Processing. you can find a table that describes the relationship between contained data items and data formats.

Example storing data into a Variant variable



e5071c436

- When you use one of the objects listed above, the base index number of the array is always 0 even if the declaration section contains the "Option Base 1" statement, which specifies the use of the base array index of 1.

For example, you may wish to read the formatted data array for a particular trace in its entirety (including all measurement points), display the data in the echo window, and then write the data into another trace. How to implement such a process can be better understood with the aid of a sample program.

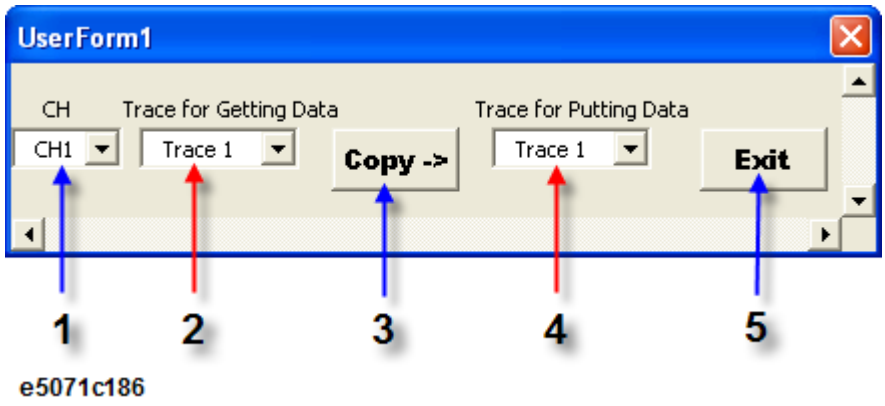
Sample program is available to download from the Agilent Support page, named "**read_write.vba**", that demonstrates how to read and write measurement data. This VBA program consists of the following modules:

Object name	Module type	Content
frmReadWrite	UserForm	Reads, displays, and writes a formatted data array.
mdlReadWrite	Standard module	Invokes a UserForm.

- This sample program correctly runs when the maximum number of channels/traces is set 9 channels/9 traces.

When you run this VBA program, a following window appears.

UserForm of read_write.vba program



e5071c186

The program lets the user specify the channel to be controlled.

1. The program lets the user specify which trace's formatted data array to read (source trace).
2. The program reads the formatted data array for the trace specified by the user, display the measurement results in the echo window, and write the data into the trace specified by the user. For detail, see the description of the code window.
3. The program lets the user specify which trace's formatted data array to overwrite (target trace).
4. The program exits, and the window disappears.

In Visual Basic Editor, open the UserForm (object name: frmReadWrite), and double-click the entire UserForm or the **Copy ->** or **Exit** button to bring

up the code window. The following is the description of the subprograms associated with the respective buttons.

Procedure called when the user clicks the **Copy** button on the UserForm (lines 10 to 520)

Lines 90 to 160

These lines identify the selected items in each list and store them into the variables TrGet, TrPut, and ActCh.

Lines 180 to 210

If the specified target trace is not displayed, these lines display that trace.

Lines 230 to 250

These lines make active the specified trace (TrGet: source trace) in the specified channel(ActCh) and hold the sweep.

Line 260

Reads the number of measurement points for the specified channel (ActCh) and stores that number into the Nop variable.

Line 280

Reads the formatted data array for the active trace (source trace) and store the data into the FmtData variable.

Line 290

Reads the stimulus array for the specified channel (ActCh) and stores the data into the Freq variable.

Line 330

Reads the data format for the active trace (source trace) and store it into the Fmt variable.

Lines 340 to 350

These lines display the echo window in the lower part of the LCD screen.

Lines 360 to 470

The lines display, in the echo window, each point along with one measured value (the odd part of the index is always 0) and a frequency if the Fmt is "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL", "IMAG", or "UPH"; or

along with two measured values and a frequency if Fmt$ returns any other string.

Line 490

Makes active the specified trace (TrPut: target trace) in the specified channel(ActCh).

Line 500

Writes the formatted data array (FmtData) into the active trace (target trace).

Procedure called when the user clicks the **Exit** button on the UserForm (lines 540 to 580)

Line 560

Unloads the UserForm from the memory, and terminates the program.

Procedure that initializes the UserForm (lines 600 to 1020)

Lines 620 to 1000

When the program is launched, these lines add each list item and set the default value for each list.

Sample Program

Reading/displaying/writing a formatted data array (read_write.frm)

```
10| Private Sub cmdCopy_Click()
20|
30| Dim X As Integer, Y As Integer, Z As Integer, I As Integer
40| Dim ActCh As Long, TrGet As Long, TrPut As Long
50| Dim TrCont As Long, Nop As Long
60| Dim FmtData As Variant, Freq As Variant
70| Dim Fmt As String
80|
90| X = cboCh.ListIndex
100| ActCh = X + 1
110|
120| Y = cboGet.ListIndex
130| TrGet = Y + 1
140|
150| Z = cboPut.ListIndex
```

```
160| TrPut = Z + 1
170|
180| TrCont = SCPI.CALCulate(ActCh).PARameter.Count
190| If TrCont < TrPut Then
200| SCPI.CALCulate(ActCh).PARameter.Count = TrPut
210| End If
220|
230| SCPI.CALCulate(ActCh).PARameter(TrGet).SELect
240| SCPI.INITiate(ActCh).CONTinuous = False
250| SCPI.ABORt
260| Nop = SCPI.SENSe(ActCh).SWEep.POINts
270|
280| FmtData = SCPI.CALCulate(ActCh).SELected.Data.FDATa
290| Freq = SCPI.SENSe(ActCh).FREQuency.Data
300|
310| "'Displays the formatted data
320|
330| Fmt = SCPI.CALCulate(ActCh).SELected.Format
340| SCPI.DISPlay.TABLe.TYPE = "ECHO"
350| SCPI.DISPlay.TABLe.STATe = True
360| Select Case Fmt
370| Case "MLOG", "PHAS", "GDEL", "MLIN", "SWR", "REAL", "IMAG", "UPH"
380| ECHO "Nop", "Frequency(GHz)", "Data"
390| For I = 0 To Nop - 1
400| ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I)
410| Next I
420| Case Else
430| ECHO "Nop", "Frequency(GHz)", "Data1", "Data2"
440| For I = 0 To Nop - 1
450| ECHO I + 1, Freq(I) / 1000000000#, FmtData(2 * I), FmtData(2 * I + 1)
460| Next I
470| End Select
480|
490| SCPI.CALCulate(ActCh).PARameter(TrPut).SELect
500| SCPI.CALCulate(ActCh).SELected.Data.FDATa = FmtData
510|
520| End Sub
```

```
530|
540| Private Sub cmdExit_Click()
550|
560| Unload Me
570|
580| End Sub
590|
600| Private Sub UserForm_Initialize()
610|
620| With cboCh
630| .AddItem "CH1"
640| .AddItem "CH2"
650| .AddItem "CH3"
660| .AddItem "CH4"
670| .AddItem "CH5"
680| .AddItem "CH6"
690| .AddItem "CH7"
700| .AddItem "CH8"
710| .AddItem "CH9"
720| End With
730|
740| With cboGet
750| .AddItem "Trace 1"
760| .AddItem "Trace 2"
770| .AddItem "Trace 3"
780| .AddItem "Trace 4"
790| .AddItem "Trace 5"
800| .AddItem "Trace 6"
810| .AddItem "Trace 7"
820| .AddItem "Trace 8"
830| .AddItem "Trace 9"
840| End With
850|
860| With cboPut
870| .AddItem "Trace 1"
880| .AddItem "Trace 2"
890| .AddItem "Trace 3"
```

```
900| .AddItem "Trace 4"
910| .AddItem "Trace 5"
920| .AddItem "Trace 6"
930| .AddItem "Trace 7"
940| .AddItem "Trace 8"
950| .AddItem "Trace 9"
960| End With
970|
980| cboCh.ListIndex = 0
990| cboGet.ListIndex = 0
1000| cboPut.ListIndex = 0
1010|
1020| End Sub
```

Executing a Procedure with a Softkey (User Menu Function)

- Overview
- Preparing for User Menu Function
- Using User Menu Function
- Sample Program

Other topics about Controlling E5071C

Overview

The E5071C lets you perform procedures assigned to specific softkeys (**Macro Setup** > **User Menu** > **Button 1/2/3/4/5/6/7/8/9/10**), without using user forms, when that softkey is pressed. This function is called the user menu function.

- You do not have to execute any VBA program when using the user menu function.

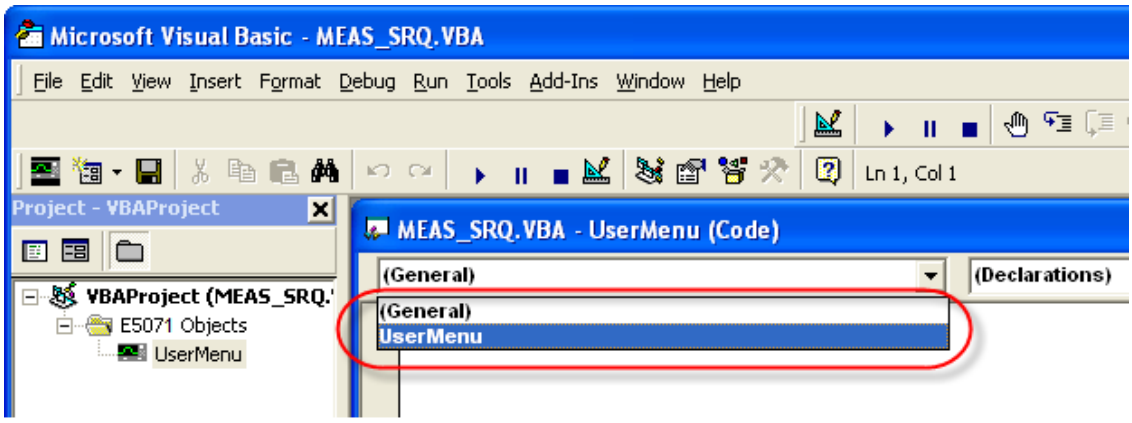
Preparing for User Menu Function

Before using the user menu function, perform the following preparation.

Coding of a Procedure Assigned to a Softkey

Follow these steps to create a procedure assigned to a specific softkey in the "UserMenu" object in the "E5071 Objects" folder.

1. Double-click UserMenu to open the code window.
2. In the object box in the code window, select **UserMenu** as shown below:



e5071c185

3. In the UserMenu_OnPress(ByVal Key_id As Long) procedure, create a program you want to assign to a specific softkey (specify with the *id* variable). For actual use example, see Line 70 to 430 in the Sample program using user menu UserMenu object.

4. During processing an event (during execution of a procedure for a key pressed), another event (an interrupt by a procedure for another softkey pressed) cannot be accepted.
5. You cannot save (export) the "UserMenu" object by module basis; save it by project basis.

Settings for Softkey Label and Softkey Enabled/Disabled

When you want to change the softkey labels for the user menu function, use the following COM object.

- UserMenu.Item(Key_id).Caption

When you want to set the softkey enabled/disabled for the user menu function, use the following COM object.

- UserMenu.Item(Key_id).Enabled

Moreover, when you want to preset the above settings for the user menu function, use the following COM object.

- UserMenu.PRESet
  - The above user menu setting is also preset by pressing **Macro Setup** > **Preset User Menu** on the E5071C front panel.

Using User Menu Function

To execute a procedure assigned to a softkey, you need to generate an event of pressing the softkey. To generate an event, the manual method and the COM object method are available.

Method by Manual Operation

Click the specific softkey as follows:

Macro Setup > **User Menu** > **Button <No>.**

"**No.**" represents a button number. You can set the label for "**Button No.**" as you like. For detail, refer to the "Settings for Softkey Label and Softkey Enabled/Disabled." section.

Method by COM Object

You can use the following COM object to perform the same operation as pressing a specific softkey.

- UserMenu.Press(Key_id)

Sample Program

Sample program is available to download from the Agilent Support page, named **meas_user.vba**, that demonstrates how to use the user menu

function. This VBA program consists of the following standard module and the "UserMenu" object.

Object name	Module type	Content
mdlUserMenu	Standard module	Sets the softkey labels and enables interrupts from the softkeys.

The program (object name: mdlUserMenu) is described in detail below:

Line 70

Stores True into the State variable.

Lines 90 to 150

Sets the first to third softkey (*id*: 1 to 3) enabled, and sets the fourth to tenth softkey (*id*: 4 to 10) disabled.

Lines 170 to 190

Sets the first softkey label (*id*: 1) to "Setup" the second softkey label (*id*: 2) to "Meas" the third softkey label (*id*: 3) to "Exit".

Line 210

Displays the buttons for the user menu function in the softkey area.

Lines 230 to 250

Processing repeated until the State variable is True (State = True).

Line 240

Detects an event that a specific softkey is pressed and enables the interrupt from the event.

Sample program using user menu (object name: mdlUserMenu)

```
10| Public State As Boolean
20|
30| Sub Main()
40|
50| Dim I As Long, J As Long
60|
70| State = True
80|
```

```
 90| For I = 1 To 3
100| UserMenu.Item(I).Enabled = True
110| Next I
120|
130| For J = 4 To 10
140| UserMenu.Item(J).Enabled = False
150| Next J
160|
170| UserMenu.Item(1).Caption = "Setup"
180| UserMenu.Item(2).Caption = "Meas"
190| UserMenu.Item(3).Caption = "Exit"
200|
210| UserMenu.Show
220|
230| Do While State
240| DoEvents
250| Loop
260|
270| End Sub
```

UserMenu object

The procedures of the "UserMenu" object are described below.

Lines 70 to 190

The procedure when the first softkey (*id*: 1) is pressed.

Line 90: Returns the E5071C to the preset state.

Lines 110 to 130 For channel 1, sets the sweep start value to 1.73 GHz, the sweep stop value to 1.83 GHz, and the number of measurement points to 51.

Lines 150 to 170 After aborting the measurement, sets the trigger source to the bus trigger and turns on the continuous trigger startup mode for channel 1.

Line 190: Displays the buttons for the user menu function in the softkey area.

Lines 210 to 320

The procedure when the second softkey (*id*: 2) is pressed.

Lines 230 to 240 Generates a trigger to start a single sweep and waits until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Line 260: Retrieves the number of points in channel 1 and stores that number into the Nop variable.

Lines 280 to 290 Specifies trace 1 of channel 1 to the active trace, retrieves the formatted data array, and stores the data into the FmtData variable.

Lines 310 to 320 Displays the echo window in the lower part of the LCD screen.

Lines 340 to 360: Displays 2 measurement data values (primary value and secondary value) for each measurement point in the echo window.

Lines 380 to 430

The procedure when the third softkey (*id*: 3) is pressed.

Line 400: Displays a program closing message.

Line 410: Stores False into the sta variable to terminate the main program.

Sample program using user menu ("UserMenu" object)

```
10| Private Sub UserMenu_OnPress(ByVal id As Long)
20|
30| Dim I As Integer
40| Dim Nop As Long, Dmy As Long
50| Dim FmtData As Variant
60|
70| If id = 1 Then
80|
90| SCPI.SYSTem.PRESet
100|
110| SCPI.SENSe(1).FREQuency.STARt = 1730000000#
120| SCPI.SENSe(1).FREQuency.STOP = 1830000000#
130| SCPI.SENSe(1).SWEep.POINts = 51
140|
150| SCPI.ABORt
160| SCPI.TRIGger.SEQuence.Source = "BUS"
170| SCPI.INITiate(1).CONTinuous = True
180|
```

```
190| UserMenu.Show
200|
210| ElseIf id = 2 Then
220|
230| SCPI.TRIGger.SEQuence.SINGle
240| Dmy = SCPI.IEEE4882.OPC
250|
260| Nop = SCPI.SENSe(1).SWEep.POINts
270|
280| SCPI.CALCulate(1).PARameter(1).SELect
290| FmtData = SCPI.CALCulate(1).SELected.DATA.FDATa
300|
310| SCPI.DISPlay.TABLe.TYPE = "ECHO"
320| SCPI.DISPlay.TABLe.State = True
330|
340| For I = 1 To Nop - 1
350| ECHO FmtData(2 * I - 2), FmtData(2 * I - 1)
360| Next I
370|
380| ElseIf id = 3 Then
390|
400| MsgBox "Program ended!"
410| State = False
420|
430| End If
440|
450| End Sub
```

User Defined Variable

The E5071C is having an area in which a User can set any value. These area are divided by the different data format of the values. A maximum of ten (1 to 10) such areas can be used by each command.

For example, after setting the value (data) obtained using VBA of the E5071C to the User defined variable, this value is available to an external controller (program) through the use of these User defined variables.

- Turning E5071C power ON/OFF initializes the User defined variables. They are not initialized through executing Preset.

  These commands does not refers to or change the results of the E5071C.

  - SCPI.PROGram.VARiable.ARRay(Vnum).DATA
  - SCPI.PROGram.VARiable.ARRay(Vnum).SIZE
  - SCPI.PROGram.VARiable.DOUBle(Vnum).DATA
  - SCPI.PROGram.VARiable.LONG(Vnum).DATA
  - SCPI.PROGram.VARiable.STRing(Vnum).DATA

Controlling Peripherals

Controlling Peripherals

- Overview
- Programming with VISA

Overview

- Overview
- Preparation

Other topics about Controlling Peripherals

Overview

The E5071C macro function (E5071C VBA) can be used not only to automate measurements but also to control external measurement instruments connected via USB/GPIB interface by acting as a self-contained system controller (see An Overview of a Control System Based on the Macro Function).

The E5071C macro function (E5071C VBA) performs communications via the COM interface when controlling the E5071C itself, but it communicates via VISA (Virtual Instrument Software Architecture) when controlling external measurement instruments.

Preparation

Importing Definition Files

To use the VISA library in the E5071C macro (E5071C VBA), you need to import two definition files into your project with the Visual Basic editor to define the VISA functions and perform other tasks. The definition files are stored on the sample programs disk under the following filenames (for information on importing modules, refer to Saving a Module (Exporting).

- **visa32.bas**
- **vpptype.bas**

Programming with VISA

- Overview
- Starting VISA
- Connection
- Communication
- Disconnection
- Sample Program

Other topics about Controlling Peripherals

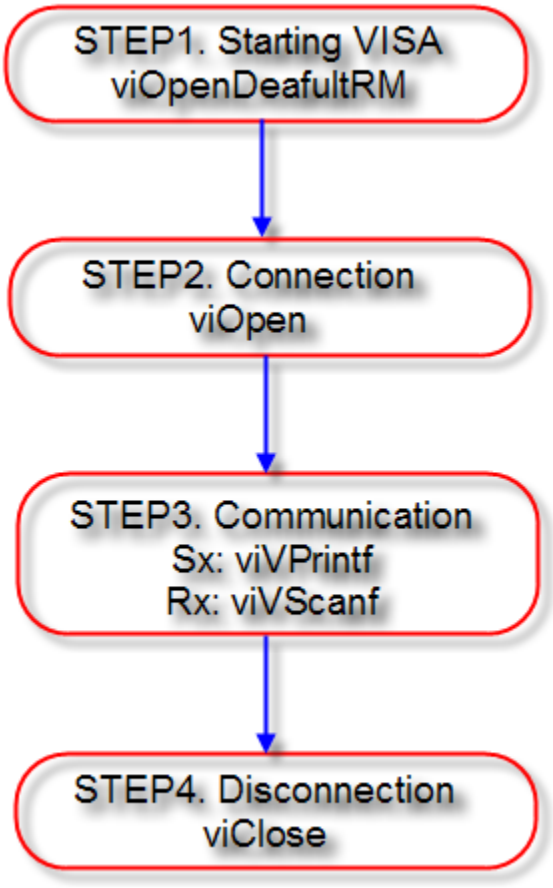
Overview

The following figure shows the flow of controlling the instrument with VISA. When developing a VISA program in the Visual Basic language, a special programing notice (in the readme text filie listed below) must be reviewed.

For details on the use of the VISA library and the programing notice for using the VISA library with the E5071C macro (E5071C VBA), refer to the following files contained in IO library CD-ROM.

- **visa.hlp** (on-line help for the VISA library)
- **vbreadme.txt** (notes on using the VISA library with VB)

Flow of instrument control with VISA



e5071c183

STEP 1. Starting VISA

The VISA system startup session is processed in Line 90 in the sample program **ctrl_ext.vba**. VISA's viOpenDefaultRM function initializes and starts up the VISA system. The viOpenDefaultRM function must be executed before other VISA functions are called, and the parameter of this function is startup information (Defrm in **ctrl_ext.vba**).

Syntax

viOpenDefaultRM(*param*)

Parameter

Parameter	(*param*)
Description	Startup information (output)
Data type	Long integer type

STEP 2. Connection

The connection session is handled in Line 130. VISA's viOpen function makes connection with the specified instrument. The viOpen function

returns a value so that the VISA functions can apply it to the specified instrument. The parameters of this function are startup information (Defrm), the address information of the specified instrument ("GPIB0::17::INSTR" in **ctrl_ext.vba**), access mode (0 in **ctrl_ext.vba**), timeout (0 in **ctrl_ext.vba**), and connection information (Equip in **ctrl_ext.vba**).

Syntax

viOpen(*param1*, *param2*, *param3*, *param4*, *param5*)

Parameters

Parameter	(*param1*)
Description	Startup information (input)
Data type	Long integer type

Parameter	(*param2*)
Description	Address information of the specified instrument (input)
Data type	Character string type
Syntax	"GPIB0::*gpib address*::INSTR" "USB0::*manufacturer ID*::*model code*::*serial number*::0::INSTR" (ex. "USB0::2391::2312::MY12345678::0::INSTR") "TCPIP0::*IP address*::inst0::INSTR"

Parameter	(*param3*)
Description	Access mode (Enter 0)

Parameter	(*param4*)
Description	Timeout (Enter 0)

Parameter	(*param5*)

Description	Connection information (output)
Data type	Long integer type

STEP 3. Communication

The communication session is conducted in Line 170. VISA's viVPrintf function sends a program message (GPIB command) to the specified instrument. The parameters of this function are connection information (Equip), the program message (*IDN?), and the variable to be formatted (0 in **ctrl_ext.vba**).

- To input/output GPIB commands, the viVPrintf function and the viVScanf function are mainly used, but other VISA functions are also available. For more information, refer to **visa.hlp** (online help for the VISA library).

Syntax

viVPrintf(*param1*, *param2*, *param3*)

Parameters

Parameter	(*param1*)
Description	Connection information (input)
Data type	Long integer type

Parameter	(*param2*)
Description	Program message (input) When sending a program message of the GPIB command, a message terminator is required at the end of the message (Chr$(10) in **ctrl_ext.vba**)
Data type	Character string type

Parameter	(*param3*)
Description	A variable to be formatted. If not applicable, enter 0.
Data type	Specified data type

The receiving session is controlled in Line 210. VISA's viVScanf function receives the result from the specified instrument and stores it in the output variable. The parameters of this function are connection information (Equip

in **ctrl_ext.vba**), the format parameter for the output variable (%t in **ctrl_ext.vba**), and the output variable (Prod in **ctrl_ext.vba**).

Syntax

viVScanf(*param1*, *param2*, *param3*)

Parameters

Parameter	(*param1*)
Description	Connection information (input)
Data type	Long integer type

Parameter	(*param2*)
Description	Format parameter for the output variable
Data type	Character string type

Parameter	(*param3*)
Description	Output variable (output)
Data type	Character string type

STEP 4. Disconnection

The disconnection session is handled in Line 280. VISA's viClose function disconnects communication and terminates the VISA system. The parameter of this function is startup information (Defrm in **ctrl_ext.vba**).

Syntax

viClose(*param*)

Parameter

Parameter	(*param*)
Description	Startup information (input)
Data type	Long integer type

Sample Program to Read Out the Product Information of Peripheral (Instrument)

The **ctrl_ext.vba** is a sample program to control instruments connected through USB/GPIB interface cable using the E5071C as the system controller. This VBA program consists of the following modules.

Object name	Module type	Content
mdlVisa	Standard module	Reads out the product information of external instrument.
Module1 Module2	Standard module	Two definition files to use VISA library

- When you control peripherals from E5071C VBA, use the GPIB commands provided for the instrument to communicate over VISA. On the other hand, when you control the E5071C itself from E5071C VBA, use the COM objects provided for the E5071C to communicate.

Lines 90 to 100

Initializes and starts up the VISA system and outputs the startup information to the Defrm variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Lines 130 to 140

Establishes the connection to the external instrument (GPIB address: 17) connected via GPIB and outputs the connection information to the Equip variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Lines 170 to 180

Queries the product information of the external instrument connected via USB/GPIB interface cable using VISA. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Lines 210 to 250

Retrieves the product information through VISA and outputs it into the Prod variable. Displays the read-out result in the message box. During this process, if an error occurs, the program goes to the error handling routine (Lines 320 to 360).

Line 280

Breaks the communication and terminates the VISA system.

Lines 320 to 360

If an error occurs in a VISA function, displays the detail of the error and terminates the program.

Read out the product information (ctrl_ext.vba)

```
10| Sub Main()
20|
30| Dim status As Long 'VISA function status return code
40| Dim Defrm As Long 'Session to Default Resource Manager
50| Dim Equip As Long 'Session to instrument
60| Dim Prod As String * 100 'String to receive the result
70|
80| ' Initializes the VISA system.
90| status = viOpenDefaultRM(Defrm)
100| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
110|
120| ' Opens the session to the specified instrument.
130| status = viOpen(Defrm, "GPIB0::17::INSTR", 0, 0, Equip)
140| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
150|
160| ' Asks for the instrument's product information.
170| status = viVPrintf(Equip, "*IDN?" & Chr$(10), 0)
180| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
190|
200| ' Reads the result.
210| status = viVScanf(Equip, "%t", Prod)
220| If (status <> VI_SUCCESS) Then GoTo VisaErrorHandler
230|
240| ' Displays the result.
250| MsgBox Prod
260|
270| ' Closes the resource manager session (which closes everything)
280| Call viClose(Defrm)
290|
300| GoTo Prog_end
310|
```

```
320| VisaErrorHandler:
330| Dim VisaErr As String * 200
340| Call viStatusDesc(Defrm, status, VisaErr)
350| MsgBox "Error : " & VisaErr, vbExclamation
360| Exit Sub
370|
380| Prog_end:
390|
400| End Sub
```

Application Programs

Application Programs

- Basic Measurement (measuring a band-pass filter)
- Measuring a Multi-port Device
- Measurement Using E5091A (measuring FEM)
- Executing Power Calibration
- Connecting Hard Disk of External PC (shared folder)

Basic Measurement (measuring a band-pass filter)

- Overview
- Overview of the Program
- Description of the Program

Other topics about Application Programs

Overview

The **apl_bsc.vba** shows a sample program (VBA program) that demonstrates how to perform the basic measurement of the band-pass filter. This VBA program consists of the following standard module.

Object name	Module type	Content
mdlBscMeas	Standard module	Performs basic measurement of band-pass filter

Overview of the Program

The sample program performs full 2-port calibration using the 85032F calibration kit, measures a band-pass filter (center frequency: 947.5 MHz), and calculates and displays its bandwidth, insertion loss, and so on. This measurement is the same as Measurement Example of a Bandpass Filter in the Quick Start.

Description of the Program

When you run this VBA program, reset is performed, the measurement conditions are automatically set, and the message "Perform the full 2-port calibration" is displayed. To perform the full 2-port calibration, click **Yes**; otherwise click **No**.

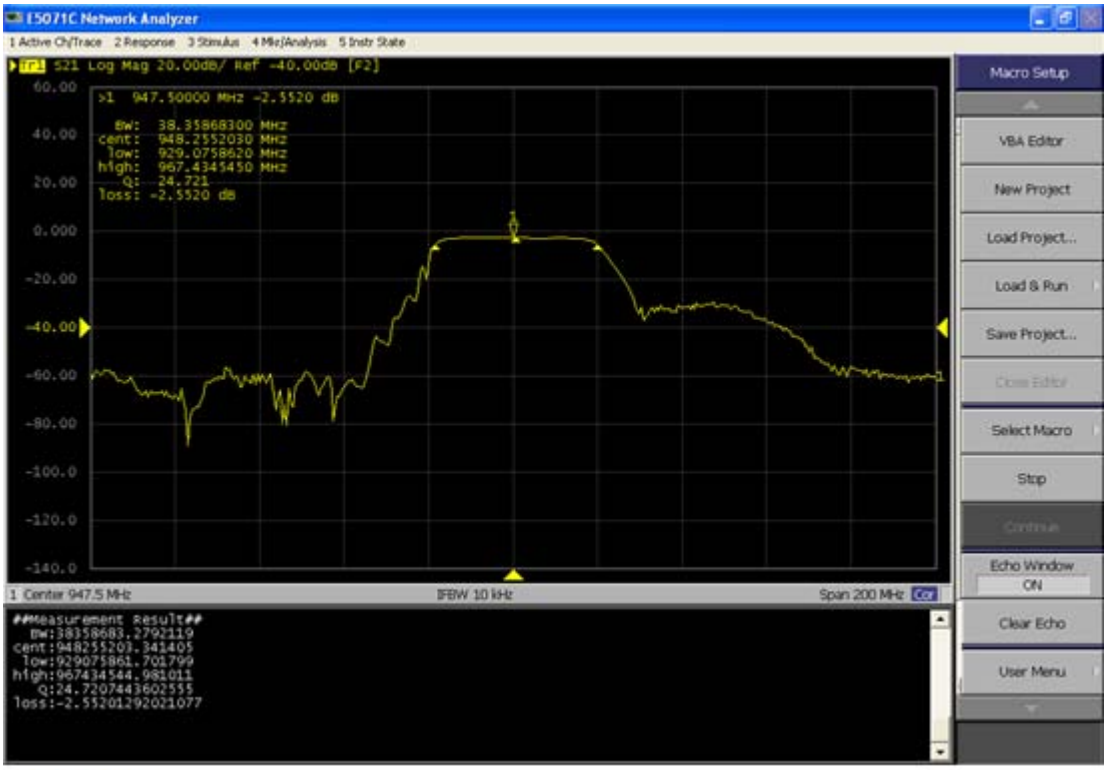
To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port, and then click **OK** to measure the calibration data. Click **Cancel** to return to the beginning of the calibration. You cannot skip the isolation calibration. When the calibration data measurement for all standards is complete, the message "All calibration data completion" is displayed, and the calibration coefficient is calculated.

- When you cancel the calibration data measurement before completing the measurement of necessary calibration data, the settings condition may not be returned to its former state.

Then, the message "Connect DUT, and then press **Macro Setup** > **Continue**" is displayed in the instrument status bar in the lower part of the LCD display. Connect a DUT and perform **Macro Setup** > **Continue**. After the measurement, the search result is displayed in the echo window, as shown

below. If no bandwidth search target is found, only the result of the insertion loss obtained with the marker is displayed.

Example of display after executing the program in "apl_bsc.vba"



e5071c188

The basic measurement program (object name: mdlBscMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Lines 120 to 160

Store the sweep center value (947.5 MHz), the sweep span value (200 MHz), the number of measurement points (401), the IF bandwidth (10 kHz), and the power level (-10 dBm) into the variables Center, Span, Nop, IfBw, and Pow, respectively.

Lines 170 to 210

Store the number of traces (1), the measurement parameter (S21), the data format (log amplitude), the calibration kit number (4: 85032F), and the save file name (State08.sta) into the variables, NumTrac, Par, Fmt, CalKit, and File, respectively.

Line 250

Returns the E5071C to the preset state.

Lines 290 to 300

For channel 1, turn on the continuous trigger startup mode to On and set the trigger source to the bus trigger.

Lines 320 to 360

For channel 1, set the sweep center value to the Center variable, the sweep span value to the Span variable, the number of measurement points to the Nop variable, the IF bandwidth to the IfBw variable, and the power level to the Pow variable.

Lines 380 to 410

For channel 1, set the number of traces to the NumTrac variable, the measurement parameter to the Par variable, and the data format to the Fmt variable.

Line 450

Stores the calibration kit number for channel 1 into the CalKit variable.

Line 460

Stores 1 and 2 into the Port variable that indicates ports used for the full 2-port calibration.

Line 480

Calls the Calib_Solt procedure (lines 1200 to 2130).

Lines 520 to 530

Save the instrument setting and the calibration coefficient into a file whose name is specified with the File variable.

Line 580

Displays a message that prompts you to connect a DUT (Device Under Test) in the instrument status bar in the lower part of the LCD display and waits for the operation of **Macro Setup** > **Continue** after the connection.

Lines 620 to 630

Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Line 650

For trace 1 of channel 1, executes auto scale to set the optimum scale.

Lines 690 to 710

Display marker 1 and move it so that the stimulus value becomes equal to the value of the Center variable. Then, these lines read out the response value of marker 1 and store it into the MkrVal variable.

Line 730

Enables the error handling routine starting from Bw_Err (lines 890 to 950). If a runtime error occurs, the program goes to the error handling routine.

Lines 750 to 770

Set the bandwidth definition value to -3 dB and the bandwidth search result display to on, read out the bandwidth search result (bandwidth, center frequency, Q value, and insertion loss), and store it into the BwData variable.

Lines 790 to 840

Based on the bandwidth search result, these lines store the bandwidth to the Bw variable, the center frequency to the Cent variable, the Q value to the Qfac variable, and the insertion loss to the Loss variable. Then, the program goes to the processing starting from Skip_Bw_Err.

Lines 880 to 960

Define a runtime error handler. These lines read out and display the error number and error message of the error that occurred and store 0 to the Bw, Cent, and Qfac variables and the response value of marker 1 (MkrVal(0) variable) to the Loss variable. Then, the program finishes the error handling and proceeds to the next processing.

Lines 1000 to 1010

Calculate the 2 (higher and lower) cutoff frequencies from the values in the Bw and Cent variables and store them into the CutLow and CutHigh variables.

Lines 1030 to 1110

Display the search result (the values of the Bw, Cent, CutLow, CutHigh, Qfac, and Loss variables) in the echo window.

Lines 1130 to 1160

Display the message asking whether you want to perform measurement again. Click **Yes** to return to the DUT connection section, otherwise click **No** to terminate the program.

Procedure: Calib_Solt (lines 1200 to 2130).

Lines 1260 to 1300

Display the message that prompts for the execution of the full n-port calibration (specified with the SoltType variable). Click **Cancel** to cancel the calibration.

Lines 1320 to 1410

Set the calibration type to the full n-port calibration for the port specified with the Port variable.

Lines 1450 to 1520

Display the message that prompts for connecting the open standard to the specified port. These lines start the measurement of the open calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1540 to 1610

Display the message that prompts for connecting the short standard to the specified port. These lines start the measurement of the short calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1630 to 1700

Display the message that prompts for connecting the load standard to the specified port. These lines start the measurement of the load calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1750 to 1840

Display the message that prompts for connecting the thru standard between the specified ports. These lines start the measurement of the thru calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 1880 to 2060

When the calibration type is not the 1-port calibration (a value other than 1 is specified for the SoltType variable), displays the message asking you whether you want to measure the isolation calibration data. When **Yes** is clicked, displays the message that prompts for connecting the load standard to the specified two ports (specified with the Port(I-1) and Port(J-1) variables). These lines start the measurement of the isolation calibration data initiated by clicking **OK** after the connection and wait for the completion of the measurement. Click **Cancel** to return to the beginning of the calibration.

Lines 2080 to 2090

Calculate the calibration coefficients from the measured calibration data and turn on the error correction function. Then, these lines display a calibration completion message.

Measuring a band-pass filter (object name: mdlBscMeas)

```
10| Sub Main()
```

```
20|
30| Dim Par As String, Fmt As String, File As String
40| Dim Center As Double, Span As Double, IfBw As Double, Pow As Double
50| Dim Bw As Double, Cent As Double
60| Dim CutLow As Double, CutHigh As Double
70| Dim Qfac As Double, Loss As Double
80| Dim MkrVal As Variant, BwData As Variant
90| Dim Nop As Long, NumTrac As Long, CalKit As Long, Buff As Long
100| Dim Port As Variant, Error As Variant
110|
120| Center = 947500000# 'Center freq : 947.5 MHz
130| Span = 200000000# 'Span freq : 200 MHz
140| Nop = 401 'Number of points : 401
150| IfBw = 10000# 'IF bandwidth : 10 kHz
160| Pow = -10 'Power level : -10dBm
170| NumTrac = 1 'Number of traces : 1
180| Par = "S21" 'Meas. parameter : S21
190| Fmt = "MLOG" 'Data format : Log Mag
200| CalKit = 4 'Calibration kit : 85032F
210| File = "State08.sta" 'Saved file name : State08.sta
220|
230| "'Presetting the E5071C
240|
250| SCPI.SYSTem.PRESet
260|
270| "'Setting measurement conditions
280|
290| SCPI.INITiate(1).CONTinuous = True
300| SCPI.TRIGger.SEQuence.Source = "BUS"
310|
320| SCPI.SENSe(1).FREQuency.Center = Center
330| SCPI.SENSe(1).FREQuency.Span = Span
340| SCPI.SENSe(1).SWEep.POINts = Nop
350| SCPI.SENSe(1).BANDwidth.RESolution = IfBw
360| SCPI.Source(1).POWer.LEVel.IMMediate.AMPLitude = Pow
370|
380| SCPI.CALCulate(1).PARameter.Count = NumTrac
```

```
390| SCPI.CALCulate(1).PARameter(1).DEFine = Par
400| SCPI.CALCulate(1).PARameter(1).Select
410| SCPI.CALCulate(1).SELected.Format = Fmt
420|
430| "'Performing full 2-port calibration
440|
450| SCPI.SENSe(1).CORRection.COLLect.CKIT.Select = CalKit
460| Port = Array(1, 2)
470|
480| Calib_Solt 1, 2, Port
490|
500| "'Saving state & cal data
510|
520| SCPI.MMEMory.STORe.STYPe = "CST"
530| SCPI.MMEMory.STORe.STATe = File
540|
550| "'Connecting DUT
560|
570| Meas_Start:
580| Prompt ("Connect DUT, and then press [Macro Setup]-Continue button.")
590|
600| "'Performing single sweep
610|
620| SCPI.TRIGger.SEQuence.SINGle
630| Dmy = SCPI.IEEE4882.OPC
640|
650| SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
660|
670| "'Analyzing the results
680|
690| SCPI.CALCulate(1).SELected.MARKer(1).STATe = True
700| SCPI.CALCulate(1).SELected.MARKer(1).X = Center
710| MkrVal = SCPI.CALCulate(1).SELected.MARKer(1).Y
720|
730| On Error GoTo Bw_Err
740|
750| SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold = -3
```

```
760| SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.STATe = True
770| BwData = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.DATA
780|
790| Bw = BwData(0)
800| Cent = BwData(1)
810| Qfac = BwData(2)
820| Loss = BwData(3)
830|
840| GoTo Skip_Bw_Err
850|
860| Bw_Err:
870|
880| Error = SCPI.SYSTem.Error
890| MsgBox "Error No:" & Error(0) & " , Description:" & Error( 1)
900|
910| Bw = 0
920| Cent = 0
930| Qfac = 0
940| Loss = MkrVal(0)
950|
960| Resume Skip_Bw_Err
970|
980| Skip_Bw_Err:
990|
1000| CutLow = Cent - Bw / 2
1010| CutHigh = Cent + Bw / 2
1020|
1030| ECHO "##Measurement Result##"
1040| ECHO " BW:" & Bw
1050| ECHO "cent:" & Cent
1060| ECHO " low:" & CutLow
1070| ECHO "high:" & CutHigh
1080| ECHO " Q:" & Qfac
1090| ECHO "loss:" & Loss
1100| SCPI.DISPlay.TABLe.TYPE = "ECHO"
1110| SCPI.DISPlay.TABLe.STATe = True
1120|
```

```
1130| Buff = MsgBox("Do you make another measurement?", vbYesNo, "Bandpass fileter
measurement")
1140| If Buff = vbYes Then
1150| GoTo Meas_Start
1160| End If
1170|
1180| End Sub
1190|
1200| Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port As Variant)
1210|
1220| Dim Dmy As Long, I As Long, J As Long, Buff As Long
1230|
1240| Cal_Start:
1250|
1260| Buff = MsgBox("Perform the full " & SoltType & "-port cali bration.", vbOKCancel, "Full" & SoltType
& "-port calibration")
1270|
1280| If Buff = vbCancel Then
1290| GoTo Cal_Skip
1300| End If
1310|
1320| Select Case SoltType
1330| Case 1
1340| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT1 = Port(0)
1350| Case 2
1360| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT2 = Port
1370| Case 3
1380| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT3 = Port
1390| Case 4
1400| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT4 = Port
1410| End Select
1420|
1430| For I = 1 To SoltType
1440|
1450| Buff = MsgBox("Connect the Open standard to Port " & CS tr(Port(I - 1)) & ".", _
1460| vbOKCancel, "Full" & SoltType & "-port calibration")
1470| If Buff = vbOK Then
1480| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.OPEN = Port(I - 1)
```

```
1490| Dmy = SCPI.IEEE4882.OPC
1500| Else
1510| GoTo Cal_Start
1520| End If
1530|
1540| Buff = MsgBox("Connect the Short standard to Port " & CStr(Port(I - 1)) & ".", _
1550| vbOKCancel, "Full" & SoltType & "-port calibration")
1560| If Buff = vbOK Then
1570| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Short = Port(I - 1)
1580| Dmy = SCPI.IEEE4882.OPC
1590| Else
1600| GoTo Cal_Start
1610| End If
1620|
1630| Buff = MsgBox("Connect the Load standard to Port " & CStr(Port(I - 1)) & ".", _
1640| vbOKCancel, "Full" & SoltType & "-port calibration")
1650| If Buff = vbOK Then
1660| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Load = Port(I - 1)
1670| Dmy = SCPI.IEEE4882.OPC
1680| Else
1690| GoTo Cal_Start
1700| End If
1710| Next I
1720|
1730| For I = 1 To SoltType - 1
1740| For J = I + 1 To SoltType
1750| Buff = MsgBox("Connect the Thru standard between Por t " & CStr(Port(I - 1)) & _
1760| " and Port " & CStr(Port(J - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port calibration")
1770| If Buff = vbOK Then
1780| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(I - 1 ), Port(J - 1))
1790| Dmy = SCPI.IEEE4882.OPC
1800| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(J - 1 ), Port(I - 1))
1810| Dmy = SCPI.IEEE4882.OPC
1820| Else
1830| GoTo Cal_Start
1840| End If
1850| Next J
```

```
1860| Next I
1870|
1880| If SoltType <> 1 Then
1890| Buff = MsgBox("Do you measure the Isolation (Optional) ?", vbYesNo, "Full" & SoltType & "-port
calibration")
1900| If Buff = vbYes Then
1910| For I = 1 To SoltType - 1
1920| For J = I + 1 To SoltType
1930| Buff = MsgBox("Connect the Load standard to Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".",
_
1940| vbOKCancel, "Full" & Solt Type & "-port calibration")
1950| If Buff = vbOK Then
1960| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port( I - 1), Port(J - 1))
1970| Dmy = SCPI.IEEE4882.OPC
1980| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port( J - 1), Port(I - 1))
1990| Dmy = SCPI.IEEE4882.OPC
2000| Else
2010| GoTo Cal_Start
2020| End If
2030| Next J
2040| Next I
2050| End If
2060| End If
2070|
2080| SCPI.SENSe(1).CORRection.COLLect.SAVE
2090| MsgBox "All calibration data completion."
2100|
2110| Cal_Skip:
2120|
2130| End Sub
```

Measuring a Multi-port Device

- Overview
- Overview of the Program
- Description of the Program
- Sample Program

Other topics about Application Programs

Overview

The **apl_sys.vba** shows a sample program (VBA program) that demonstrates how to measure a (3-terminal) duplexer. This VBA program consists of the following modules:

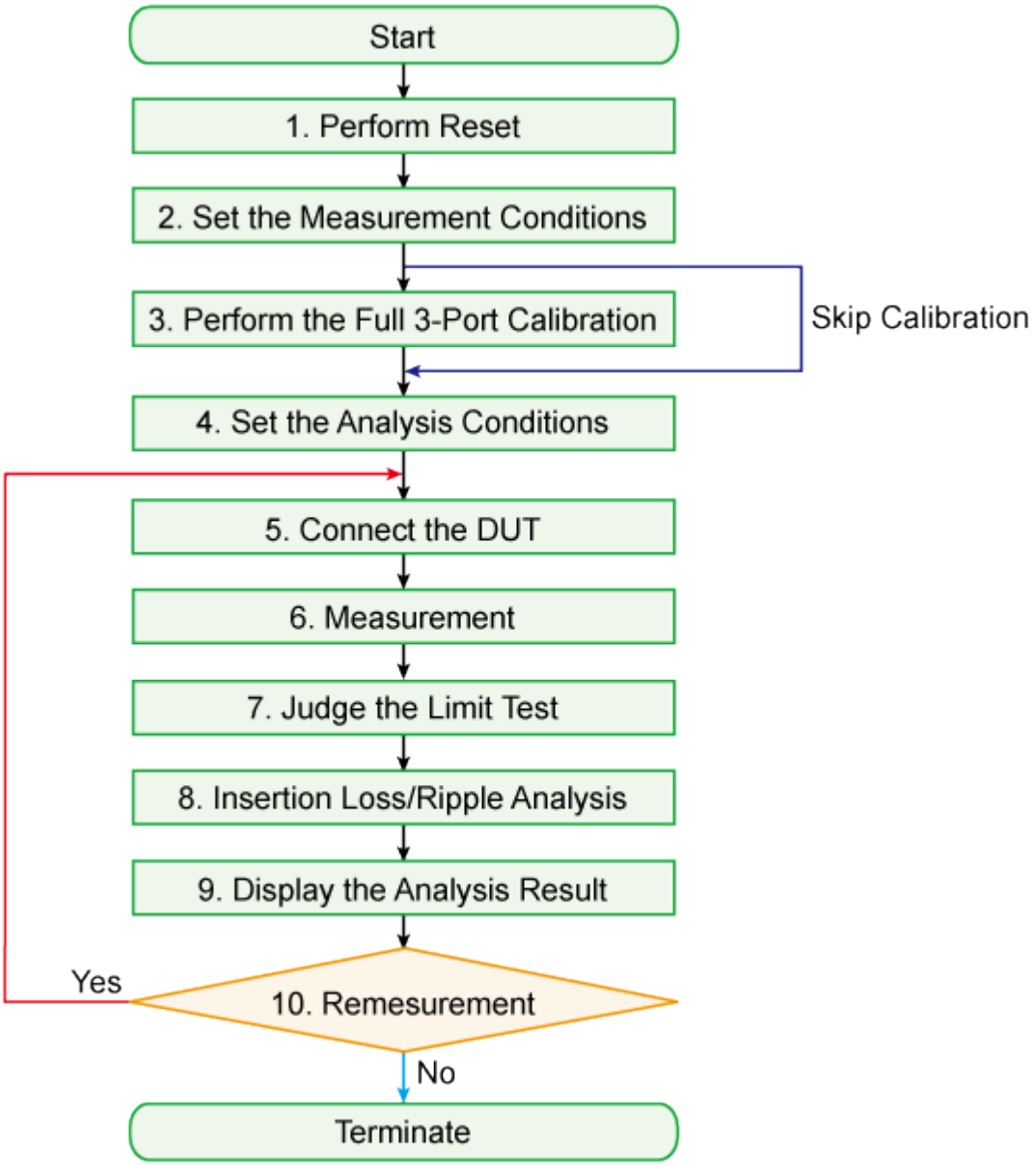
- For the E5071C with 2-port S-parameter test set, a runtime error occurs because there are parameters that cannot be measured.

Object name	Module type	Description
frmDupRes	User form	Displays the analysis result
mdlDupMeas	Standard module	Performs duplexer measurement

Overview of the Program

The program performs full 3-port calibration using the 85032F calibration kit, measures a (3-terminal) duplexer, and calculates and displays the limit test result, insertion loss, and band-pass ripple. The following figure shows the simple flow of the (3-terminal) duplexer measurement program:

Flow of duplexer measurement program



e5071c435

Description of the Program

When you run this VBA program, reset is performed, the measurement conditions are set, and the message "Perform the full 3-port calibration" is displayed. To perform the full 3-port calibration, click **Yes**, otherwise click **No**.

To perform the calibration, follow the onscreen messages to connect each standard of the Agilent 85032F calibration kit to the specified port and then click **OK** to measure the calibration data. Click **Cancel** to return to the beginning of the calibration. You cannot skip the isolation calibration. When

the calibration data measurement for all standards is complete, the message "All calibration data completion" is displayed, and the calibration coefficient is calculated.

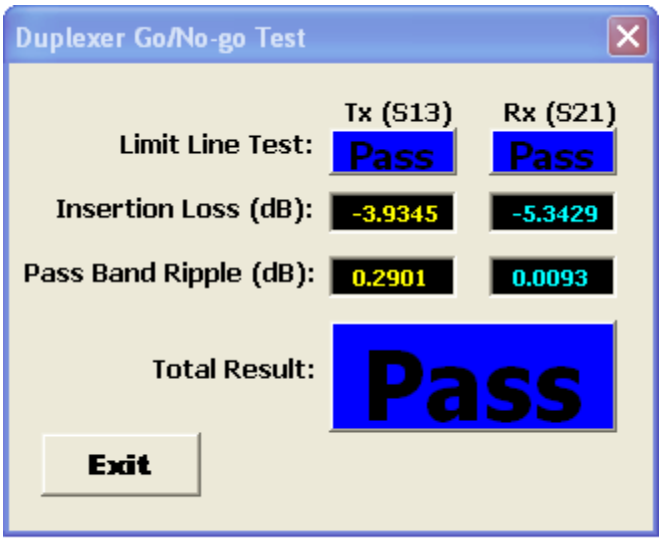
- When you cancel the calibration data measurement before completing the measurement of necessary calibration data, the settings conditions may not return to its original state.

Then, the limit line is set and the setting required for the limit test judgment is performed.

The message "Set the DUT between test cables." is displayed. Connect the DUT (duplexer) between the test cables and then click **Yes**. The limit line is set and a single sweep is executed. Then, for both trace 1 (Tx: S13) and trace 2 (Rx: S21), the Pass/Fail judgment of the limit test and the insertion loss and ripple analysis result within the pass band are displayed.

Click **Exit** on the user form displaying the analysis result. The message prompting for remeasurement is displayed. To perform remeasurement, click **Yes,** otherwise click **No** to terminate the program. The details of the program within the user form for displaying the analysis result (object name: frmDupRes) are not described here.

Display of the execution result of the program of "apl_sys.vba"



e5071c187

The duplexer measurement program (object name: mdlDupMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Line 90

Stores the calibration kit number (4: 85032F) into the CalKit variable.

Line 110

Turns off display update. Turning off display update shortens drawing time and object processing time.

Line 140

Returns the E5071C to the preset state.

Lines 170 to 180

For channel 1, these lines turn the continuous trigger startup mode to On and sets the trigger source to the bus trigger.

Line 200

Calls the Setup_Parameter procedure (lines 910 to 1140). For information on the Setup_Parameter procedure, see the description later.

Line 220

Calls the Setup_Segment procedure (lines 1160 to 1530). For information on the Setup_Segment procedure, see the description later.

Line 250

Stores the calibration kit number for channel 1 into the CalKit variable.

Line 260

Stores 1, 2, and 3 into the Port variable that indicates the ports used for the full 3-port calibration.

Line 280

Calls the Calib_Solt procedure.

Line 310

Calls the Setup_Limitline procedure (lines 2420 to 3180). For information on the Setup_Limitline procedure, see the description later.

Line 330

Calls the Setup_Register procedure (lines 3200 to 3260). For information on the Setup_Register procedure, see the description later.

Line 380

Displays the message that prompts for connecting a DUT (Device Under Test) and waits for the **OK** button to be clicked after the connection.

Line 410

Clears the questionable limit status event register and questionable limit channel 1 status event register.

Lines 420 to 430

Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Lines 450 to 460

For traces 1 and 2, these lines executes the auto scale to set the optimum scale.

Lines 490 to 500

Read out the value of the questionable limit status event register and store the AND of the read-out value and 2 (the value in which only bit 1 is 1) into the Test_Ch1 variable.

Lines 510 to 530

Read out the value of the questionable limit channel 1 status event register and store the AND of the read-out value and 2 (the value in which only bit 1 is 1) into the Test_Tr1 variable and the AND of the read-out value and 4 (the value in which only bit 2 is 1) into the Test_Tr2 variable.

Lines 560 to 610

Specify trace 1 as the active trace and set the analysis range (start point: 1.85 GHz and stop point: 1.91 GHz). Then these lines sets the polarity of the peak search (both the positive peak and the negative peak) and the lower limit of the peak excursion value (0).

Lines 620 to 640

Search for the minimum value within the analysis range and store the analysis result into the IlossTx variables.

Line 650

Uses the ripple analysis function to store the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range into the RipTx variables.

Lines 670 to 720

Specify trace 2 as the active trace and set the analysis range (start point: 1.93 GHz and stop point: 1.99 GHz). Then these lines set the polarity of the peak search (both the positive peak and the negative peak) and the lower limit of the peak excursion value (0).

Lines 730 to 750

Search for the minimum value within the analysis range and store the analysis result into the IlossRx variables.

Line 760

Uses the ripple analysis function to store the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range into the RipRx variables.

Line 790

Calls the Display_Update procedure (lines 3280 to 3620). For information on the Display_Update procedure, see the description later.

Line 810

Displays the user form (object name: frmDupRes) on the screen to display the analysis result.

Lines 830 to 870

Display the message asking whether you want to perform measurement again. Click the **Yes** button to return to the DUT connection section. Click the **No** button to terminate the program.

Procedure: Setup_Parameter (lines 910 to 1140)

Lines 970 to 1020

Store the channel layout ("D1": 1-channel display), graph layout ("D1_2": upper/lower 2 part split display), measurement parameter of trace 1 (S13), measurement parameter of trace 2 (S21), data format of trace 1 (MLOG), and data format of trace 2 (MLOG) into the ChDisp, TracDisp, Par(0), Par(1), Fmt(0), and Fmt(1) variables, respectively.

Lines 1040 to 1060

Set the number of traces for channel 1 to 2, the channel layout to the ChDisp variable, and the graph layout for channel 1 to the TracDisp variable.

Lines 1080 to 1120

Set the measurement parameter for trace 1 to the Par(0) variable, the data format for trace 1 to the Fmt(0) variable, the measurement parameter for trace 2 to the Par(1) variable, and the data format for trace 2 to the Fmt(1) variable.

Procedure: Setup_Segment (lines 1160 to 1530)

Lines 1200 to 1260

Store the conditions for channel 1's segment table setting into the SegmData(0) to SegmData(6) variables. The settings are as follows. Stimulus setting mode: start/stop value. IF bandwidth setting for each segment: off. Power setting for each segment: off. Sweep delay time setting for each segment: off. Sweep time setting for each segment: off. Number of segments: 5.

Lines 1280 to 1470

Store the sweep start value, the sweep stop value, and the number of measurement points for channel 1's segments 1 through 5 into the SegmData(7) to SegmData(21) variables, respectively.

Line 1490

Sets channel 1's segment table to the SegmData variable.

Line 1500

Sets channel 1's sweep type to "segment".

Line 1510

Sets the channel 1 graph's horizontal axis display method to the order base (the axis on which the measurement point numbers are placed evenly in the order of measurement).

Procedure: Calib_Solt (lines 1550 to 2420)

See Lines 1200 to 2130 of Calib_Solt.

Procedure: Setup_Limitline (lines 2440 to 3180)

Line 2490

Stores the number of lines (5) in trace 1 limit table into the LimDataS13(0) variable.

Lines 2500 to 2790

Store the settings in trace 1 limit table into the LimDataS13(1) to LimDataS13(25) variables.

Line 2820

Stores the number of lines (4) in trace 2 limit table into the LimDataS21(0) variable.

Lines 2830 to 3060

Store the settings in trace 2 limit table into the LimDataS21(1) to LimDataS21(20) variables.

Lines 3080 to 3110

Specify trace 1 as the active trace, store trace 1's limit line into the LimDataS13 variable, and display it. Then, these lines turn on the limit test function for trace 1.

Lines 3130 to 3160

Specify trace 2 as the active trace, store trace 2's limit line into the LimDataS21 variable, and display it. Then, these lines turn on the limit test function for trace 2.

Procedure: Setup_Register (lines 3200 to 3260)

Lines 3220 to 3230

Set the instrument so that the questionable limit channel status event register's bits 1 and 2 are set to 1 only when the questionable limit channel status register's bits 1 and 2 are changed from 0 to 1 (positive transition).

Line 3240

Enables the questionable limit channel status event register's bits 1 and 2.

Procedure: Display_Update (lines 3280 to 3620).

Line 3300

Updates the display on the LCD screen once.

Lines 3320 to 3380

When trace 1's limit test result is Fail (Test_Tr1 = 2), these lines display Tx(S13) "Limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when trace 1's limit test result is Pass (Test_Tr1 ¼ 2), they display Tx(S13) "Limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3400 to 3460

When trace 2's limit test result is Fail (Test_Tr2 = 4), these lines display Rx(S21) "Limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when trace 1's limit test result is Pass (Test_Tr2 ≠ 4), they display Rx(S21) "Limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3480 to 3540

When channel 1's limit test result is Fail (Test_Ch1 = 2), these lines display "Overall limit test result: Fail" on the user form (object name: frmDupRes) against a red background. On the other hand, when channel 1's limit test result is Pass (Test_Ch1 ≠ 2), they display "Overall limit test result: Pass" on the user form (object name: frmDupRes) against a blue background.

Lines 3560 to 3600

Display the analysis results for traces 1 and 2 (insertion loss and band-pass ripple) as Tx(S13) and Rx(S21), respectively, on the user form (object name: frmDupRes).

Sample Program

Duplexer measurement (object name: mdlDupMeas)

```
10| Sub Main()
20|
```

```
30| Dim CalKit As Long, Dmy As Long, Rgst As Long, I As Long, Buff As Long
40| Dim Test_Ch1 As Integer, Test_Tr1 As Integer, Test_Tr2 As Integer
50| Dim IlossTx As Variant, IlossRx As Variant
60| Dim RipTx As Double, RipRx As Double
70| Dim Port As Variant
80|
90| CalKit = 4 'Calibration kit :85032F
100|
110| SCPI.DISPlay.ENABle = False
120|
130| "'Presetting the E5071C
140| SCPI.SYSTem.PRESet
150|
160| "'Setting measurement conditions
170| SCPI.INITiate(1).CONTinuous = True
180| SCPI.TRIGger.SEQuence.Source = "BUS"
190|
200| Setup_Parameter
210|
220| Setup_Segment
230|
240| "'Full 3-port calibration
250| SCPI.SENSe(1).CORRection.COLLect.CKIT.Select = CalKit
260| Port = Array(1, 2, 3)
270|
280| Calib_Solt 1, 3, Port
290|
300| "'Setting analysis conditions
310| Setup_Limitline
320|
330| Setup_Register
340|
350| Meas_Start:
360|
370| "'Connecting DUT
380| MsgBox "Connect DUT between test cables."
390|
```

```
400| "'Performing single sweep
410| SCPI.IEEE4882.CLS
420| SCPI.TRIGger.SEQuence.SINGle
430| Dmy = SCPI.IEEE4882.OPC
440|
450| SCPI.DISPlay.WINDow(1).TRACe(1).Y.SCALe.AUTO
460| SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO
470|
480| "'Judging limit test
490| Rgst = SCPI.STATus.QUEStionable.LIMit.EVENt
500| Test_Ch1 = CInt(Rgst And 2)
510| Rgst = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENt
520| Test_Tr1 = CInt(Rgst And 2)
530| Test_Tr2 = CInt(Rgst And 4)
540|
550| "'Analyzing insertion loss/bandpass ripple
560| SCPI.CALCulate(1).PARameter(1).Select
570| SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 1850000000#
580| SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 1910000000#
590| SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True
600| SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "both"
610| SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 0
620| SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "MIN"
630| SCPI.CALCulate(1).SELected.FUNCtion.EXECute
640| IlossTx = SCPI.CALCulate(1).SELected.FUNCtion.DATA
650| RipTx = MaxPeakToPeak(1)
660|
670| SCPI.CALCulate(1).PARameter(2).Select
680| SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 1930000000#
690| SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 1990000000#
700| SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True
710| SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "both"
720| SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 0
730| SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "MIN"
740| SCPI.CALCulate(1).SELected.FUNCtion.EXECute
750| IlossRx = SCPI.CALCulate(1).SELected.FUNCtion.DATA
760| RipRx = MaxPeakToPeak(1)
```

```
770|
780| "'Displaying the results
790| Display_Update Test_Tr1, Test_Tr2, Test_Ch1, IlossTx, IlossRx, RipTx, RipRx
800|
810| frmDupRes.Show
820|
830| Buff = MsgBox("Do you make another measurement?", vbYesNo, "Duplexer Measurement")
840|
850| If Buff = vbYes Then
860| GoTo Meas_Start
870| End If
880|
890| End Sub
900|
910| Private Sub Setup_Parameter()
920|
930| Dim I As Long
940| Dim ChDisp As String, TracDisp As String
950| Dim Par(1) As String, Fmt(1) As String
960|
970| ChDisp = "D1"
980| TracDisp = "D1_2"
990| Par(0) = "S13"
1000| Par(1) = "S21"
1010| Fmt(0) = "MLOG"
1020| Fmt(1) = "MLOG"
1030|
1040| SCPI.CALCulate(1).PARameter.Count = 2
1050| SCPI.DISPlay.Split = ChDisp
1060| SCPI.DISPlay.WINDow(1).Split = TracDisp
1070|
1080| For I = 1 To 2
1090| SCPI.CALCulate(1).PARameter(I).DEFine = Par(I - 1)
1100| SCPI.CALCulate(1).PARameter(I).Select
1110| SCPI.CALCulate(1).SELected.Format = Fmt(I - 1)
1120| Next I
1130|
```

```
1140| End Sub
1150|
1160| Private Sub Setup_Segment()
1170|
1180| Dim SegmData(21) As Variant
1190|
1200| SegmData(0) = 5 'Anytime 5 is set at seg ment settings
1210| SegmData(1) = 0 'Allows stimulus range to be set using Start/Stop frequency
1220| SegmData(2) = 0 'Not allows IF bandwidth to be set
1230| SegmData(3) = 0 'Not allows power to be set
1240| SegmData(4) = 0 'Not allows delay time to be set
1250| SegmData(5) = 0 'Not allows sweep time to be set
1260| SegmData(6) = 5 'Number of segments
1270|
1280| "'Segment 1
1290| SegmData(7) = 1730000000# 'Start frequency
1300| SegmData(8) = 1830000000# 'Stop frequency
1310| SegmData(9) = 50 'Number of points
1320| "'Segment 2
1330| SegmData(10) = 1830000000# 'Start frequency
1340| SegmData(11) = 2030000000# 'Stop frequency
1350| SegmData(12) = 400 'Number of points
1360| "'Segment 3
1370| SegmData(13) = 2030000000# 'Start frequency
1380| SegmData(14) = 2130000000# 'Stop frequency
1390| SegmData(15) = 50 'Number of points
1400| "'Segment 4
1410| SegmData(16) = 3650000000# 'Start frequency
1420| SegmData(17) = 4030000000# 'Stop frequency
1430| SegmData(18) = 38 'Number of points
1440| "'Segment 5
1450| SegmData(19) = 5500000000# 'Start frequency
1460| SegmData(20) = 6020000000# 'Stop frequency
1470| SegmData(21) = 52 'Number of points
1480|
1490| SCPI.SENSe(1).SEGMent.DATA = SegmData
1500| SCPI.SENSe(1).SWEep.TYPE = "SEGM"
```

```
1510| SCPI.DISPlay.WINDow(1).X.SPACing = "OBAS"
1520|
1530| End Sub
1540|
1550| Private Sub Calib_Solt(Chan As Long, SoltType As Long, Port A s Variant)
1560|
1570| Dim Dmy As Long, I As Long, J As Long, Buff As Long
1580|
1590| Cal_Start:
1600|
1610| Buff = MsgBox("Perform the full " & SoltType & "-port calib ration.", vbOKCancel, "Full" & SoltType
& "-port calibration")
1620| If Buff = vbCancel Then
1630| GoTo Cal_Skip
1640| End If
1650|
1660| Select Case SoltType
1670| Case 1
1680| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT1 = Port(0)
1690| Case 2
1700| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT2 = Port
1710| Case 3
1720| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT3 = Port
1730| Case 4
1740| SCPI.SENSe(Chan).CORRection.COLLect.METHod.SOLT4 = Port
1750| End Select
1760|
1770| For I = 1 To SoltType
1780|
1790| Buff = MsgBox("Connect the Open standard to Port " & CS tr(Port(I - 1)) & ".", vbOKCancel, "Full"
& SoltType & "-port calibr ation")
1800| If Buff = vbOK Then
1810| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.OPEN = Port(I - 1)
1820| Dmy = SCPI.IEEE4882.OPC
1830| Else
1840| GoTo Cal_Start
1850| End If
1860|
```

```
1870| Buff = MsgBox("Connect the Short standard to Port " & C Str(Port(I - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port calib ration")
1880| If Buff = vbOK Then
1890| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Short = Port(I - 1)
1900| Dmy = SCPI.IEEE4882.OPC
1910| Else
1920| GoTo Cal_Start
1930| End If
1940|
1950| Buff = MsgBox("Connect the Load standard to Port " & CS tr(Port(I - 1)) & ".", vbOKCancel, "Full" & SoltType & "-port calibr ation")
1960| If Buff = vbOK Then
1970| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.Load = Port(I - 1)
1980| Dmy = SCPI.IEEE4882.OPC
1990| Else
2000| GoTo Cal_Start
2010| End If
2020| Next I
2030|
2040| For I = 1 To SoltType - 1
2050| For J = I + 1 To SoltType
2060| Buff = MsgBox("Connect the Thru standard between Por t " & CStr(Port(I - 1)) & " and Port " & CStr(Port(J - 1)) & ".", vb OKCancel, "Full" & SoltType & "-port calibration")
2070| If Buff = vbOK Then
2080| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(I - 1), Port(J - 1))
2090| Dmy = SCPI.IEEE4882.OPC
2100| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.THRU = Array(Port(J - 1), Port(I - 1))
2110| Dmy = SCPI.IEEE4882.OPC
2120| Else
2130| GoTo Cal_Start
2140| End If
2150| Next J
2160| Next I
2170|
2180| If SoltType <> 1 Then
2190| Buff = MsgBox("Do you measure the Isolation (Optional)? ", vbYesNo, "Full" & SoltType & "-port calibration")
2200| If Buff = vbYes Then
```

```
2210| For I = 1 To SoltType - 1
2220| For J = I + 1 To SoltType
2230| Buff = MsgBox("Connect the Load standard to Port " & Port(I - 1) & " and Port " & Port(J - 1) & ".", vbOKCancel, "Full" & SoltType & "-port calibration")
2240| If Buff = vbOK Then
2250| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port( I - 1), Port(J - 1))
2260| Dmy = SCPI.IEEE4882.OPC
2270| SCPI.SENSe(Chan).CORRection.COLLect.ACQuire.ISOLation = Array(Port( J - 1), Port(I - 1))
2280| Dmy = SCPI.IEEE4882.OPC
2290| Else
2300| GoTo Cal_Start
2310| End If
2320| Next J
2330| Next I
2340| End If
2350| End If
2360|
2370| SCPI.SENSe(1).CORRection.COLLect.SAVE
2380| MsgBox "All calibration data completion."
2390|
2400| Cal_Skip:
2410|
2420| End Sub
2430|
2440| Private Sub Setup_Limitline()
2450|
2460| Dim LimDataS13(25) As Variant, LimDataS21(20) As Variant
2470|
2480| '"Limit line for S13
2490| LimDataS13(0) = 5 'Number of segment
2500| '"Limit_line 1
2510| LimDataS13(1) = 1 'Maximum
2520| LimDataS13(2) = 1730000000# 'Beginning of stimulus
2530| LimDataS13(3) = 1930000000# 'End of stimulus
2540| LimDataS13(4) = 0 'Beginning of response
2550| LimDataS13(5) = 0 'End of response
2560| '"Limit_line 2
```

```
2570| LimDataS13(6) = 2 'Minimum
2580| LimDataS13(7) = 1850000000# 'Beginning of stimulus
2590| LimDataS13(8) = 1910000000# 'End of stimulus
2600| LimDataS13(9) = -8 'Beginning of response
2610| LimDataS13(10) = -8 'End of response
2620| '"Limit_line 3
2630| LimDataS13(11) = 1 'Maximum
2640| LimDataS13(12) = 1930000000# 'Beginning of stimulus
2650| LimDataS13(13) = 1990000000# 'End of stimulus
2660| LimDataS13(14) = -35 'Beginning of response
2670| LimDataS13(15) = -35 'End of response
2680| '"Limit_line 4
2690| LimDataS13(16) = 1 'Maximum
2700| LimDataS13(17) = 1990000000# 'Beginning of stimulus
2710| LimDataS13(18) = 2130000000# 'End of stimulus
2720| LimDataS13(19) = -40 'Beginning of response
2730| LimDataS13(20) = -40 'End of response
2740| '"Limit_line 5
2750| LimDataS13(21) = 1 'Maximum
2760| LimDataS13(22) = 2130000000# 'Beginning of stimulus
2770| LimDataS13(23) = 6020000000# 'End of stimulus
2780| LimDataS13(24) = -20 'Beginning of response
2790| LimDataS13(25) = -20 'End of response
2800|
2810| '"Limit line for S21
2820| LimDataS21(0) = 4 'Number of segment
2830| '"Limit_line 1
2840| LimDataS21(1) = 1 'Maximum
2850| LimDataS21(2) = 1730000000# 'Beginning of stimulus
2860| LimDataS21(3) = 1850000000# 'End of stimulus
2870| LimDataS21(4) = -40 'Beginning of response
2880| LimDataS21(5) = -40 'End of response
2890| '"Limit_line 2
2900| LimDataS21(6) = 1 'Maximum
2910| LimDataS21(7) = 1850000000# 'Beginning of stimulus
2920| LimDataS21(8) = 1910000000# 'End of stimulus
2930| LimDataS21(9) = -40 'Beginning of response
```

```
2940| LimDataS21(10) = -40 'End of response
2950| "'Limit_line 3
2960| LimDataS21(11) = 1 'Maximum
2970| LimDataS21(12) = 1910000000# 'Beginning of stimulus
2980| LimDataS21(13) = 6020000000# 'End of stimulus
2990| LimDataS21(14) = 0 'Beginning of response
3000| LimDataS21(15) = 0 'End of response
3010| "'Limit_line 4
3020| LimDataS21(16) = 2 'Minimum
3030| LimDataS21(17) = 1930000000# 'Beginning of stimulus
3040| LimDataS21(18) = 1990000000# 'End of stimulus
3050| LimDataS21(19) = -10 'Beginning of response
3060| LimDataS21(20) = -10 'End of response
3070|
3080| SCPI.CALCulate(1).PARameter(1).Select
3090| SCPI.CALCulate(1).SELected.LIMit.DATA = LimDataS13
3100| SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe = True
3110| SCPI.CALCulate(1).SELected.LIMit.STATe = True
3120|
3130| SCPI.CALCulate(1).PARameter(2).Select
3140| SCPI.CALCulate(1).SELected.LIMit.DATA = LimDataS21
3150| SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe = True
3160| SCPI.CALCulate(1).SELected.LIMit.STATe = True
3170|
3180| End Sub
3190|
3200| Private Sub Setup_Register()
3210|
3220| SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition = 6
3230| SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition = 0
3240| SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABle = 6
3250|
3260| End Sub
3270|
3280| Sub Display_Update(Test_Tr1 As Integer, Test_Tr2 As Integer, Test_Ch1 As Integer, IlossTx As
Variant, IlossRx As Variant, RipTx As Variant, RipRx As Variant)
3290|
```

```
3300| SCPI.DISPlay.UPDate.IMMediate
3310|
3320| If Test_Tr1 = 2 Then
3330| frmDupRes.lblJudgeS13.BackColor = RGB(255, 0, 0)
3340| frmDupRes.lblJudgeS13.Caption = "Fail"
3350| Else
3360| frmDupRes.lblJudgeS13.BackColor = RGB(0, 0, 255)
3370| frmDupRes.lblJudgeS13.Caption = "Pass"
3380| End If
3390|
3400| If Test_Tr2 = 4 Then
3410| frmDupRes.lblJudgeS21.BackColor = RGB(255, 0, 0)
3420| frmDupRes.lblJudgeS21.Caption = "Fail"
3430| Else
3440| frmDupRes.lblJudgeS21.BackColor = RGB(0, 0, 255)
3450| frmDupRes.lblJudgeS21.Caption = "Pass"
3460| End If
3470|
3480| If Test_Ch1 = 2 Then
3490| frmDupRes.lblResult.BackColor = RGB(255, 0, 0)
3500| frmDupRes.lblResult.Caption = "Fail"
3510| Else
3520| frmDupRes.lblResult.BackColor = RGB(0, 0, 255)
3530| frmDupRes.lblResult.Caption = "Pass"
3540| End If
3550|
3560| frmDupRes.txtIlossS13.Text = Format(IlossTx(0), "0.#### ")
3570| frmDupRes.txtIlossS21.Text = Format(IlossRx(0), "0.#### ")
3580|
3590| frmDupRes.txtRipS13.Text = Format(RipTx, "0.####")
3600| frmDupRes.txtRipS21.Text = Format(RipRx, "0.####")
3610|
3620| End Sub
```

Measurement Using E5091A (measuring FEM)

- Overview
- Sample Program

Other topics about Application Programs

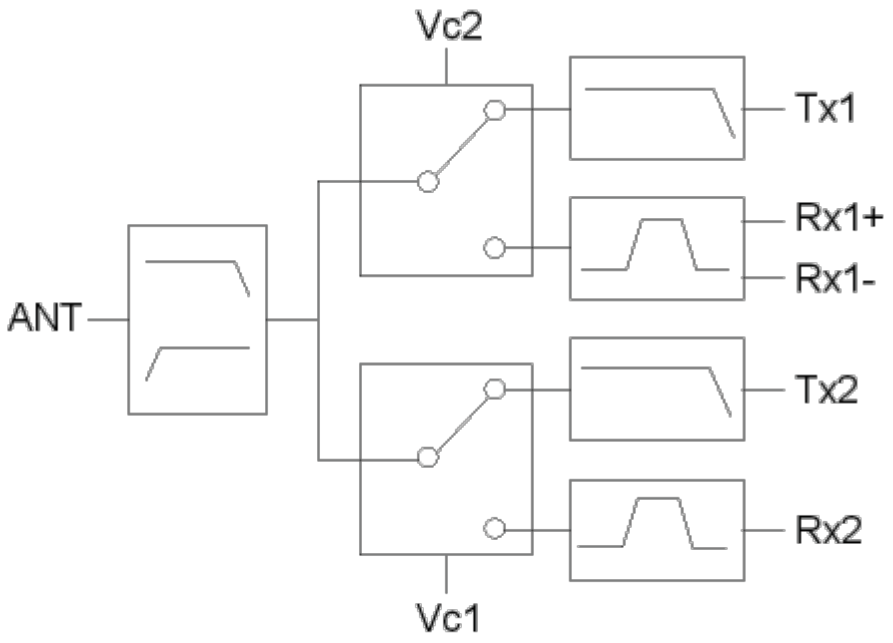
Overview

The **apl_fem.vba** shows front end module (FEM) measurement as a sample program of measurement using the E5091A.

- If the E5071C does not have 4-port S parameter test set, a runtime error occurs because there are parameters that cannot be measured.
- This VBA program cannot control the E5091A-016, the multiport test set.

Object name	Module type	Description
mdlFemMeas	Standard module	Performs measurement of FEM

This program calibrates each channel using the ECal module and then measures the transmission characteristics EGSM:Tx-Antenna (channel 1), EGSM:Antenna-Rx (channel 2), GSM1800:Tx-Antenna (channel 3), and GSM1800:Antenna-Rx (channel 4) of the 6-port dual-band FEM as shown in the below figure.



e5070buj073

When you start the program, "Connect A and T1 to ECal Module" is displayed. Connect the cables from A and T1 of the E5091A to the ECal module and then press the **OK** key to calibrate channel 1. If trouble occurs due to a problem in the connection to the ECal module, an error message is displayed. You can execute calibration again by clicking the **Retry** button. If you want to abort the program, click the **Cancel** button. For channels 2 to 4, execute the calibration in the same way.

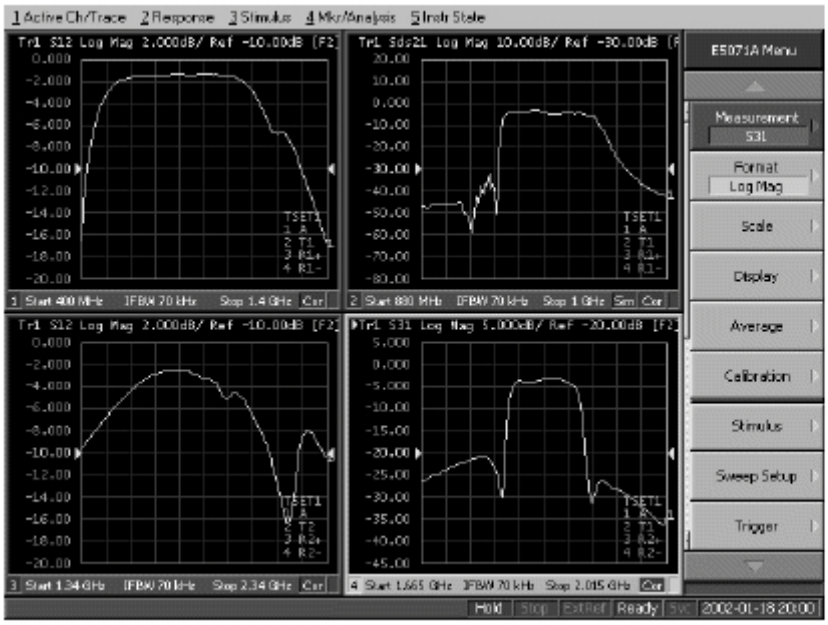
When calibration is complete, "Set DUT" is displayed. Connect the DUT (FEM) and the E5091A as shown below and click the **OK** button to start measurement.

FEM		**E5091A**
Antenna		**A**
EGSM	Tx	T1
	Rx+	R1+
	Rx-	R1-
GSM1800	Tx	T2
	Rx	R2+

Vc1	Control Line 1
Vc2	Control Line 2

The following figure shows a sample display of the LCD screen after the program exits execution.

Example of display after execution of program in "apl_fem.vba"



The FEM measurement program (object name: mdlFemMeas) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Lines 140 to 330

Set the ports assigned to Port 1 to Port 4 of the E5091A and the control line setting (table below) into the variables.

Channel number	Port 1	Port 2	Port 3	Port 4	Control Lines
1	A	T1	R1+	R1-	0 (00000000)
2	A	T1	R1+	R1-	2

					(00000010)
3	A	T2	R2+	R2-	0 (00000000)
4	A	T2	R2+	R2-	1 (00000001)

Lines 340 to 660

Set the settings required for the measurement conditions in the table below to the variables.

Channel number	Sweep range		Number of points	Number of traces	Measurement parameter
	Start	Stop			
1	400 MHz	1.4 GHz	51	1	S12
2	880 MHz	1 GHz	101	1	Sds21
3	1.34 GHz	2.34 GHz	201	1	S12
4	1.665 GHz	2.015 GHz	101	1	S31

Channel number	Fixture simulator		Title
	ON / OFF	Topology	
1	OFF	E/span> E/span>	[EGSM] Tx-Antenna
2	ON	SE:1, Bal:3,4	[EGSM] Antenna-Rx
3	OFF	E/span> E/span>	[GSM1800] Antenna-Rx

4	OFF	E/span> E/span>	[GSM1800] Tx-Antenna

Line 710

Puts the instrument into preset state.

Line 720

Allocates the windows to the upper left, upper right, lower left, and lower right.

Lines 740 to 1020

Repeat the following for channels 1 to 4, where Ch is the channel number.

Lines 780 to 810: For the E5091A whose ID is 1, set the port assigned to port 1 to Port1(Ch-1), the port assigned to port 2 to Port2(Ch-1), the Port assigned to port 3 to Port3(Ch-1), and the port assigned to port 4 to Port4(Ch-1).

Line 820: Sets the control line of the E5091A whose ID is 1 to Clines(Ch-1).

Lines 860 to 890: Set the sweep start value to Freq_star(Ch-1), the sweep stop value to Freq_stop(Ch-1), the number of points to Nop(Ch-1), and the number of traces to N_tr(Ch-1).

Lines 910 to 950: If the fixture simulator function is ON (Fsim(Ch-1) is True), these lines set the fixture simulator function to ON, the device type to Dev(Ch-1), the port assignment to Tpl(Ch-1), the balance-unbalance conversion to ON, and the measurement parameter (mix mode S-parameter) to Trc(Ch-1).

Line 970: If the fixture simulator function is OFF (Fsim(i) is False), sets the measurement parameter (S-parameter) to Trc(Ch-1).

Lines 990 to 1010: Set the title label to Ttl(Ch-1), the title display to ON, and the continuous startup mode to ON.

Line 1040

Sets the trigger source to "Bus."

Lines 1050 to 1060

For the E5091A whose ID is 1, set the property display to ON and the control to ON.

Lines 1100 to 1120

Recall the Function procedure: ECal_solt (Lines 1410 to 1770) to execute the calibration of channel 1 with the ECal module (full 2-port calibration of

ports A and T1). If the calibration is not completed correctly, these lines abort the program.

Lines 1130 to 1210

Execute the calibration of channels 2 to 4 in the same way.

Line 1260

Displays the message that prompts for connecting a DUT (Device Under Test) and waits for the **OK** button to be clicked after the connection.

Lines 1280 to 1290

Generate a trigger to start a single sweep and wait until the measurement finishes (1 is read out with the SCPI.IEEE4882.OPC object).

Lines 1310 to 1330

Execute auto scale for trace 1 of channels 1 to 4.

Line 1350

Displays the message asking whether you want to perform measurement again.

Line 1360

If the **Yes** button is clicked, returns to the DUT connection section.

Function procedure: ECal_solt (lines 1410 to 1770)

Line 1460

Clears the error queue.

Lines 1460 to 1480

Display the message that prompts for connecting the Tset_Port of the E5091A to the ECal module and wait for the **OK** button to be clicked after the connection.

Line 1500

Enables the error handling routine starting from Ecal_Err (lines 1670 to 1740). If a runtime error occurs, the program goes to the error handling routine.

Line 1540

If solt is 1, executes the ECal command that performs 1-port calibration on port Ana_port(0) of channel Ch.

Line 1560

If solt is 2, executes the ECal command that performs full 2-port calibration on port Ana_port of channel Ch.

Line 1580

If solt is 3, executes the ECal command that performs full 3-port calibration on port Ana_port of channel Ch.

Line 1600

If solt is 4, executes the ECal command that performs full 4-port calibration on port Ana_port of channel Ch.

Line 1630

Sets the return value of ECal_solt to 0.

Lines 1670 to 1740

Define a runtime error handler.

Lines 780 to 810: For the E5091A whose ID is 1, these lines set the port assigned to port 1 to Port1(Ch-1), the port assigned to port 2 to Port2(Ch-1), the Port assigned to port 3 to Port3(Ch-1), and the port assigned to port 4 to Port4(Ch-1).

Line 1670: Retrieves the error number and error message from the error queue.

Line 1680: Displays the error message.

Line 1700: When the **Retry** button is clicked, the program disables the error handler routine and then returns to the connection part and repeats ECal.

Lines 1720 to 1730: When the **Cancel** button is clicked, the program sets the return value of ECal_solt to the error number and disables the error handler routine.

Sample Program

Measurement of FEM (object name: mdlFemMeas)

```
10| Sub Main()
20|
30| Dim Port1(3) As String, Port2(3) As String
40| Dim Port3(3) As String, Port4(3) As String
50| Dim Trc(3) As String, Fsim(3) As Boolean, Dev(3) As String
60| Dim Tpl(3) As Variant, Ttl(3) As String, Inp_char As String
70| Dim Freq_star(3) As Double, Freq_stop(3) As Double
80| Dim CLines(3) As Long, Nop(3) As Long, N_tr(3) As Long
90| Dim Ch As Long, Res As Long, Buff As Long, Dmy As Long
100| Dim AnaPort As Variant
110| '
120| ' E5091A Setup
130| '
```

```
140| Port1(0) = "A" '[Ch1] Port1: A
150| Port2(0) = "T1" ' Port2: T1
160| Port3(0) = "R1" ' Port3: R1+
170| Port4(0) = "R1" ' Port4: R1-
180| CLines(0) = 0 ' Control Lines: 0
190| Port1(1) = "A" '[Ch2] Port1: A
200| Port2(1) = "T1" ' Port2: T1
210| Port3(1) = "R1" ' Port3: R1+
220| Port4(1) = "R1" ' Port3: R1-
230| CLines(1) = 2 ' Control Lines: 2 (Line1:HIGH)
240| Port1(2) = "A" '[Ch3] Port1: A
250| Port2(2) = "T2" ' Port2: T2
260| Port3(2) = "R2" ' Port3: R2+
270| Port4(2) = "R2" ' Port4: R2- (Dummy)
280| CLines(2) = 0 ' Control Lines: 2 (Line1:HIGH)
290| Port1(3) = "A" '[Ch4] Port1: A
300| Port2(3) = "T2" ' Port2: T2
310| Port3(3) = "R2" ' Port3: R2+
320| Port4(3) = "R2" ' Port4: R2- (Dummy)
330| CLines(3) = 1 ' Control Lines: 1 (Line0:HIGH)
340| '
350| ' Measurement Condition
360| ' [Ch1]
370| Freq_star(0) = 400000000# ' Start frequency : 400 MHz
380| Freq_stop(0) = 1400000000# ' Stop frequency : 1.4 GHz
390| Nop(0) = 51 ' Number of points : 51
400| N_tr(0) = 1 ' Number of traces : 1
410| Fsim(0) = False ' Fixture Simulator : OFF
420| Trc(0) = "S12" ' Meas. parameter : S12
430| Ttl(0) = "[EGSM] Tx-Antenna" ' Title
440| ' [Ch2]
450| Freq_star(1) = 880000000# ' Start frequency : 880 MHz
460| Freq_stop(1) = 1000000000# ' Stop frequency : 1 GHz
470| Nop(1) = 101 ' Number of points : 101
480| N_tr(1) = 1 ' Number of traces : 1
490| Fsim(1) = True ' Fixture Simulator : ON
500| Dev(1) = "SBAL" ' Bal. Device Type : SE-Bal
```

```
510| Tpl(1) = Array(1, 3, 4) ' Topology : SE:1, Bal:3-4
520| Trc(1) = "SDS21" ' Meas. parameter : Sds21
530| Ttl(1) = "[EGSM] Antenna-Rx" ' Title
540| ' [Ch3]
550| Freq_star(2) = 1340000000# ' Start frequency : 1.34 GHz
560| Freq_stop(2) = 2340000000# ' Stop frequency : 2.34 GHz
570| Nop(2) = 201 ' Number of points : 201
580| N_tr(2) = 1 ' Number of traces : 1
590| Fsim(2) = False ' Fixture Simulator : OFF
600| Trc(2) = "S12" ' Meas. parameter : S12
610| Ttl(2) = "[GSM1800] Tx-Antenna" ' Title
620| ' [Ch4]
630| Freq_star(3) = 1665000000# ' Start frequency : 1.665 GHz
640| Freq_stop(3) = 2015000000# ' Stop frequency : 2.015 GHz
650| Nop(3) = 101 ' Number of points : 101
660| N_tr(3) = 1 ' Number of traces : 1
670| Fsim(3) = False ' Fixture Simulator : OFF
680| Trc(3) = "S31" ' Meas. parameter : S31
690| Ttl(3) = "[GSM1800] Antenna-Rx" ' Title
700|
710| SCPI.SYSTem.PRESet
720| SCPI.DISPlay.Split = "D12_34"
730|
740| For Ch = 1 To 4
750| '
760| ' E5091A Setup
770| '
780| SCPI.SENSe(Ch).MULTiplexer(1).TSET9.Port1 = Port1(Ch - 1)
790| SCPI.SENSe(Ch).MULTiplexer(1).TSET9.Port2 = Port2(Ch - 1)
800| SCPI.SENSe(Ch).MULTiplexer(1).TSET9.Port3 = Port3(Ch - 1)
810| SCPI.SENSe(Ch).MULTiplexer(1).TSET9.Port4 = Port4(Ch - 1)
820| SCPI.SENSe(Ch).MULTiplexer(1).TSET9.OUTPut.DATA = CLines(Ch - 1)
830| '
840| ' Measurement Condition
850| '
860| SCPI.SENSe(Ch).FREQuency.STARt = Freq_star(Ch - 1)
870| SCPI.SENSe(Ch).FREQuency.STOP = Freq_stop(Ch - 1)
```

```
880| SCPI.SENSe(Ch).SWEep.POINts = Nop(Ch - 1)
890| SCPI.CALCulate(Ch).PARameter.Count = N_tr(Ch - 1)
900| If Fsim(Ch - 1) = True Then
910| SCPI.CALCulate(Ch).FSIMulator.STATe = True
920| SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice = Dev(Ch - 1)
930| SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SBALanced.PPOR ts = Tpl(Ch - 1)
940| SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter(1).STATe = Tr ue
950| SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter(1).SBALanced. DEFine = Trc(Ch - 1)
960| Else
970| SCPI.CALCulate(Ch).PARameter(1).DEFine = Trc(Ch - 1)
980| End If
990| SCPI.DISPlay.WINDow(Ch).TITLe.DATA = Ttl(Ch - 1)
1000| SCPI.DISPlay.WINDow(Ch).TITLe.STATe = True
1010| SCPI.INITiate(Ch).CONTinuous = True
1020| Next Ch
1030|
1040| SCPI.TRIGger.SEQuence.Source = "BUS"
1050| SCPI.SENSe.MULTiplexer(1).DISPlay.STATe = True
1060| SCPI.SENSe.MULTiplexer(1).STATe = True
1070| '
1080| ' Calibration
1090| '
1100| AnaPort = Array(1, 2)
1110| Res = ECal_Solt(1, 2, AnaPort, "A and T1")
1120| If Res <> 0 Then GoTo Prg_end
1130| AnaPort = Array(1, 3, 4)
1140| Res = ECal_Solt(2, 3, AnaPort, "A, R1+ and R1-")
1150| If Res <> 0 Then GoTo Prg_end
1160| AnaPort = Array(1, 2)
1170| Res = ECal_Solt(3, 2, AnaPort, "A and T2")
1180| If Res <> 0 Then GoTo Prg_end
1190| AnaPort = Array(1, 3)
1200| Res = ECal_Solt(4, 2, AnaPort, "A and R2+")
1210| If Res <> 0 Then GoTo Prg_end
1220| '
1230| ' Measurement
1240| '
```

```
1250| Meas_Start:
1260| MsgBox "Connect DUT.", vbOKOnly, "Measurement"
1270|
1280| SCPI.TRIGger.SEQuence.SINGle
1290| Dmy = SCPI.IEEE4882.OPC
1300|
1310| For Ch = 1 To 4
1320| SCPI.DISPlay.WINDow(Ch).TRACe(1).Y.SCALe.AUTO
1330| Next Ch
1340|
1350| Buff = MsgBox("Do you make another measurement?", vbYesNo)
1360| If Buff = vbYes Then GoTo Meas_Start
1370|
1380| Prg_end:
1390| End Sub
1400|
1410| Function ECal_Solt(Ch As Long, Solt As Long, AnaPort As Variant, TsetPort As String) As Long
1420| Dim Err_info As Variant
1430| Dim Buff As Long
1440|
1450| Ecal_start:
1460| SCPI.IEEE4882.CLS
1470|
1480| MsgBox "Connect " + TsetPort + " to ECal Module."
1490|
1500| On Error GoTo Ecal_err
1510|
1520| Select Case Solt
1530| Case 1
1540| SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1 = AnaPort(0)
1550| Case 2
1560| SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2 = AnaPort
1570| Case 3
1580| SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT3 = AnaPort
1590| Case 4
1600| SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT4 = AnaPort
1610| End Select
```

```
1620|
1630| ECal_Solt = 0
1640| GoTo Ecal_end
1650|
1660| Ecal_err:
1670| Err_info = SCPI.SYSTem.Error
1680| Buff = MsgBox("Error: " + Err_info(1), vbRetryCancel)
1690| If Buff = vbRetry Then
1700| Resume Ecal_start
1710| Else
1720| ECal_Solt = Err_info(0)
1730| Resume Ecal_end
1740| End If
1750|
1760| Ecal_end:
1770| End Function
```

Executing Power Calibration

- Overview
- Program Overview
- Program Description

Other topics about Application Programs

Overview

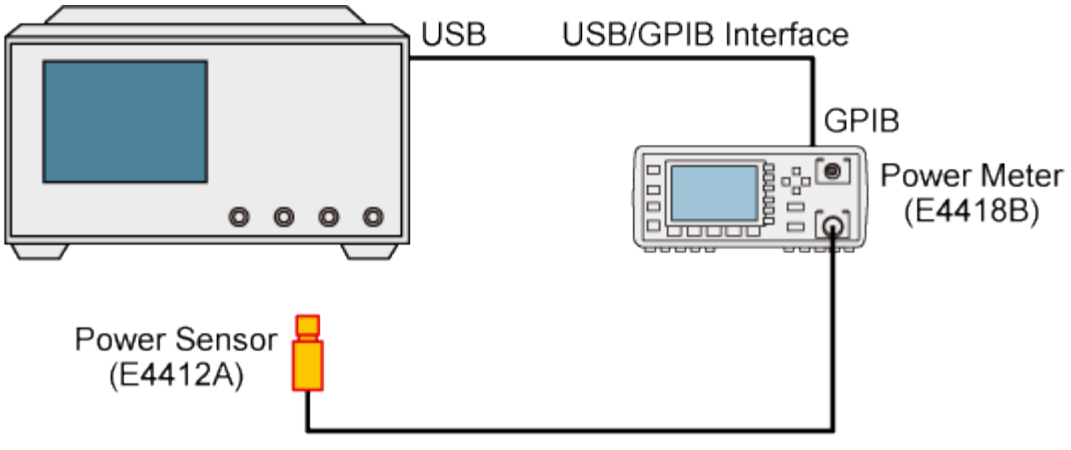
The **pow_cal.vba** shows a sample program (VBA program) for executing power calibration using the E4418B power meter and the E4412A power sensor. This VBA program consists of the following modules:

Object name	Module type	Description
mdlPowCal	Standard module	Executes power calibration
Module1 Module2	Standard module	Definition files for using VISA library

Program Overview

The program connects the E5071C and the E4418B (GPIB address: 13) through the USB/GPIB interface as shown below. Then, the program executes the zero adjustment and calibration of the power sensor (E4412A) connected to the power meter (E4418B) as necessary. Finally, it executes the power calibration of the E5071C and saves the obtained power calibration data array into a file.

Connection between E5071C and power meter



e5071c336

Program Description

When you run this VBA program, reset is performed, the GPIB address of the power meter to be controlled and the measurement conditions of the E5071C are set, and the message **"Do you want to zero and calibrate the power sensor?"** is displayed. If you want to execute the zero adjustment and the calibration of the power sensor, click **Yes**, otherwise click **No**.

If you want to execute the zero adjustment and calibration of the power sensor, follow the displayed messages to connect the power sensor connected to the A channel of the power meter to the POWER RF port of the power meter and click **OK**. When the zero adjustment and calibration of the power sensor is complete, the message "**Zeroing and calibrating the power sensor is complete**" is displayed.

When a message that prompts you to connect the power sensor to port 1 of the E5071C is displayed, make the connection and then click **OK**. The power calibration data measurement is executed immediately and the obtained power calibration data array is saved in a file named **CORR_DATA**.

The power calibration execution program (object name: mdlPowCal) is described in detail below. Line numbers are added for description purpose only and do not appear in the actual program source code.

Lines 100 to 150

Assign the sweep type (power sweep), the number of points (41), the power range (-20 to +12 dBm), the sweep start value (-20 dBm), the sweep stop value (-10 dBm), and the fixed frequency (1 GHz) to the Swp_type, Nop, Pow_rang, Start_p, Stop_p, and Cw_freq variables, respectively.

Lines 160 to 170

Assign the number of power calibration data measurements for each measurement point (4) and the limit value of the power calibration data array (10 dBm) into the Num_avg and Limit variables, respectively.

Line 200

Returns the E5071C to the preset state.

Line 230

Sets the power meter's GPIB address (13) in the E5071C.

Lines 260 to 310

For channel 1, these lines assign the sweep type to the Swp_type variable, the number of points to the Nop variable, the power range to the Pow_rang variable, the sweep start value to the Start_p variable, the

sweep stop value to the Stop_p variable, and the fixed frequency to the Cw_freq variable.

Lines 340 to 370

Display a message asking whether to execute the zero adjustment and calibration of the power sensor. If the **Yes** button is clicked, these lines call the procedure: Control_PowerMeter (Line 900 to Line 1340). On the other hand, if the **No** button is clicked, the program skips the zero adjustment and calibration of the power sensor.

Lines 410 to 420

Clear the error queue. Then, these lines prompt you to connect the power sensor to port 1 of the E5071C and wait for the **OK** button to be clicked after the connection.

Lines 450 to 470

For port 1 of channel 1, these lines set the number of power calibration data measurements for each measurement point to the Num_avg variable and then start the measurement of the power calibration data and wait for the completion of the measurement.

Line 500

Reads out an error that has occurred in the E5071C during the measurement of the power calibration data and sets it in the Err variable.

Lines 510 to 590

If no error has occurred, these lines read out the power calibration data array and set it in the Corr_data variable. In addition, the program uses the Limit_Test function to check whether the read out power calibration data array exceeds the specified limit value. If the limit value is exceeded, the return value of the Limit_Test function, False, is returned. Then, a message is displayed asking whether to perform the power calibration again. Click the **Yes** button to return to the start of the power calibration data measurement. Click the **No** button to terminate the program.

Lines 610 to 660

If an error occurs, display an error message and a message asking whether to execute the power calibration again. Click the **Yes** button to return to the start of the power calibration data measurement. Click the **No** button to terminate the program.

Lines 690 to 790

Write the read out power calibration data array into a file named **CORR_DATA**. Then, these lines display a message that saving into a file has been successfully completed.

Procedure: Control_PowerMeter (lines 900 to 1340)

Lines 990 to 1000

Initialize and start up the VISA system and output the startup information to the Defrm variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 1260 to 1300).

Lines 1030 to 1040

Establish the connection to the power meter in use (GPIB address: 13) and output the connection information to the E4418 variable. During this process, if an error occurs, the program goes to the error handling routine (Lines 1260 to 1300).

Lines 1070 to 1090

Return the power meter to the preset state through VISA and clear the status byte register and the standard event status register. Then, these lines enable the standard event status register's bit 0.

Line 1100

Prompts you to connect the power sensor to the POWER REF port of the power meter and waits for the **OK** button to be clicked after the connection.

Lines 1110 to 1120

Execute the zero adjustment and calibration of the power sensor through VISA. These lines make the setting so that 1 is set to bit 0 of the standard event status register when all pending operations are completed.

Lines 1130 to 1190

Retrieve the value of the status byte register through VISA and set it into the Rgst variable. These lines set the AND of the read-out value and 32 (the value in which only bit 5 is 1) into the Rslt variable and display a message that the zero adjustment and calibration of the power sensor is completed when Rslt becomes 1 (when the zero adjustment and calibration of the power sensor is completed).

Line 1220

Breaks the communication and terminates the VISA system.

Lines 1260 to 1300

If an error occurs in a VISA function, these lines display the details of the error and terminate the program.

Function: Limit_Test (Lines 1360 to 1530)

Lines 1420 to 1470

If the absolute value of the read out power calibration data array exceeds the specified limit value, these lines turn off the power calibration function and return the value of False. Otherwise, the value of True is returned.

Measurement of power calibration (object name: mdlPowCal)

```
10| Sub Main()
20|
30| Dim Swp_type As String, File As String
40| Dim Start_p As Double, Stop_p As Double, Cw_freq As Double, Limi t As Double
50| Dim Nop As Long, Pow_rang As Long, Num_avg As Long, Data_size As Long, Buff As Long, Dmy
As Long
60| Dim Corr_data As Variant, Err As Variant
70| Dim Verifier As Boolean
80| Dim FileNo As Integer, I As Integer
90|
100| Swp_type = "POW" 'Sweep type : POWER
110| Nop = 41 'Number of points : 41
120| Pow_rang = 0 'Power range : -20 to +12 dBm
130| Start_p = -20# 'Start power : -20 dBm
140| Stop_p = -10# 'Stop power : -10 dBm
150| Cw_freq = 1000000000# 'CW frequency : 1 GHz
160| Num_avg = 4 'Number of averaging : 4
170| Limit = 10# 'limit for corrected data : 10 dBm
180|
190| "'Presetting the ENA
200| SCPI.SYSTem.PRESet
210|
220| "'Setting GPIB address of the power meter to ENA
230| SCPI.SYSTem.COMMunicate.GPIB.PMETer.address = 13
240|
250| "'Setting measurement conditions
260| SCPI.SENSe(1).SWEep.TYPE = Swp_type
270| SCPI.SENSe(1).SWEep.POINts = Nop
```

```
280| SCPI.Source(1).POWer.ATTenuation.DATA = Pow_rang
290| SCPI.Source(1).POWer.STARt = Start_p
300| SCPI.Source(1).POWer.STOP = Stop_p
310| SCPI.SENSe(1).FREQuency.CW = Cw_freq
320|
330| "'Performing a calibration in the power meter
340| Buff = MsgBox("Do you perform zeroing and calibrating the power sensor?", vbYesNo, "Power meter calibration")
350| If Buff = vbYes Then
360| Control_PowerMeter
370| End If
380|
390| Meas_Start:
400| "'Connecting the power sensor to the port 1 in the ENA
410| SCPI.IEEE4882.CLS
420| MsgBox "Set the power sensor connected to the port 1 in the ENA.", vbOKOnly, "Power meter calibration"
430|
440| "'Performing power calibration measurement
450| SCPI.Source(1).POWer.PORT(1).CORRection.COLLect.AVERage.count = Num_avg
460| SCPI.Source(1).POWer.PORT(1).CORRection.COLLect.ACQuire = "ASEN "
470| Dmy = SCPI.IEEE4882.OPC
480|
490| "'Error handling at power meter calibration
500| Err = SCPI.SYSTem.Error
510| If Err(0) = 0 Then
520| Corr_data = SCPI.Source(1).POWer.PORT(1).CORRection.DATA
530| Verifier = Limit_Test(Nop, Limit, Corr_data)
540| If Verifier = False Then
550| Buff = MsgBox("Do you perform the power meter calibration me asurement again?", vbYesNo, "Power meter calibration")
560| If Buff = vbYes Then GoTo Meas_Start
570| If Buff = vbNo Then GoTo Prog_Stop
580| End If
590| MsgBox "Power meter calibration measurement is complete.", vbO KOnly, "Power meter calibration"
600|
610| Else
620| MsgBox "Error: " & Err(1)
```

```
630| Buff = MsgBox("Do you perform the power meter calibration meas urement again?", vbYesNo,
"Power meter calibration")
640| If Buff = vbYes Then GoTo Meas_Start
650| If Buff = vbNo Then GoTo Prog_Stop
660| End If
670|
680| "'Installing the corrected data to a file
690| FileNo = FreeFile
700| File = "CORR_DATA"
710|
720| Open File For Output As FileNo
730|
740| For I = 0 To Nop - 1
750| Write #FileNo, Val(Corr_data(I))
760| Next I
770| Close #FileNo
780|
790| MsgBox "Installing the corrected data to the file is DONE.", vbO KOnly, "Power meter calibration"
800|
810| GoTo Prog_End
820|
830| Prog_Stop:
840| MsgBox "Program Interruption", vbOKOnly, "Power meter calibratio n"
850|
860| Prog_End:
870|
880| End Sub
890|
900| Private Sub Control_PowerMeter()
910|
920| Dim Status As Long 'VISA function status return
930| Dim Defrm As Long 'Session to default resource code
940| Dim E4418 As Long 'Session to power meter
950| Dim Rslt As Integer
960| Dim Rgst As String * 10
970|
980| "'Initializing the VISA system
```

```
990| Status = viOpenDefaultRM(Defrm)
1000| If (Status <> VI_SUCCESS) Then GoTo VisaErrorHandler
1010|
1020| "'Opening the session to the power meter
1030| Status = viOpen(Defrm, "GPIB0::13::INSTR", 0, 0, E4418)
1040| If (Status <> VI_SUCCESS) Then GoTo VisaErrorHandler
1050|
1060| "'Zeroing and calibratingthe power meter
1070| Status = viVPrintf(E4418, "SYST:PRES" & Chr$(10), 0)
1080| Status = viVPrintf(E4418, "*CLS" & Chr$(10), 0)
1090| Status = viVPrintf(E4418, "*ESE 1" & Chr$(10), 0)
1100| MsgBox "Set the power sensor connected to the POWER REF port in the power meter.",
vbOKOnly, "Power meter calibration"
1110| Status = viVPrintf(E4418, "CAL1:ALL" & Chr$(10), 0)
1120| Status = viVPrintf(E4418, "*OPC" & Chr$(10), 0)
1130| Do While Rslt = 0
1140| Status = viVPrintf(E4418, "*STB?" & Chr$(10), 0)
1150| Status = viVScanf(E4418, "%t", Rgst)
1160| Rslt = CInt(CInt(Rgst) And 32)
1170| Loop
1180|
1190| MsgBox "Zeroing and Calibrating the power sensor is complete.", vbOKOnly, "Power meter
calibration"
1200|
1210| "'Closing the resource manager session
1220| Call viClose(Defrm)
1230|
1240| GoTo Prog_End
1250|
1260| VisaErrorHandler:
1270| Dim VisaErr As String * 200
1280| Call viStatusDesc(Defrm, Status, VisaErr)
1290| MsgBox "Error : " & VisaErr, vbExclamation
1300| End
1310|
1320| Prog_End:
1330|
1340| End Sub
```

```
1350|
1360| Function Limit_Test(Nop As Long, Limit As Double, Corr_data As Var iant) As Boolean
1370|
1380| Dim I As Integer
1390|
1400| For I = 0 To Nop - 1
1410|
1420| If Abs(Corr_data(I)) > Limit Then
1430| SCPI.Source(1).POWer.PORT(1).CORRection.STATe = False
1440| MsgBox "The corrected data is out of limit!", vbExclamation, "Power meter calibration"
1450| Limit_Test = False
1460| Exit Function
1470| End If
1480|
1490| Next I
1500|
1510| Limit_Test = True
1520|
1530| End Function
```

Connecting Hard Disk of External PC (shared folder)

- Overview
- Using VBA Program
- Description of VBA Program

Other topics about Application Programs

Overview

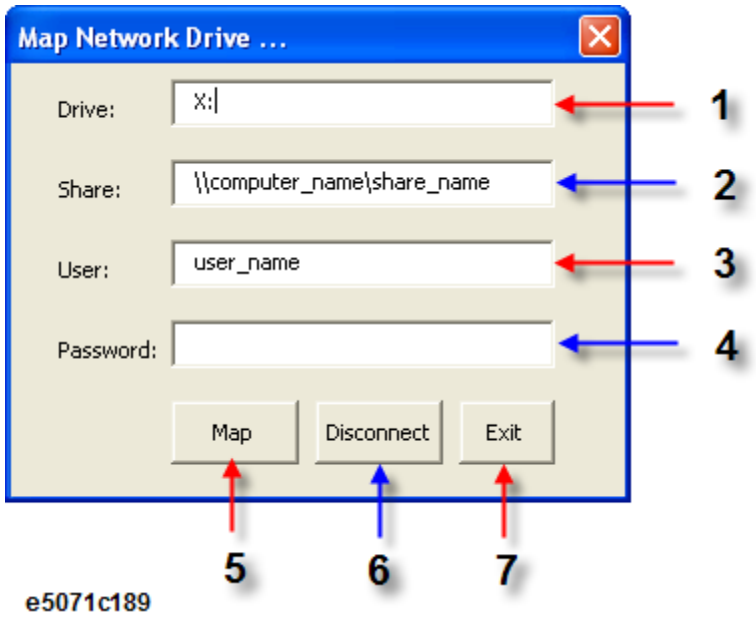
The **map_drive.vba** shows a sample program (VBA program) that demonstrates how to connect a hard disk (a shared folder) of an external PC to the E5071C. This VBA program consists of the following modules:

Object name	Module type	Description
frmMapDrive	User form	Connects or disconnects a hard disk
Module1	Standard module	Displays frmMapDrive

Using VBA Program

Load the **map_drive.vba** and press **Macro Run** key. The following macro appears.

Shared folder connection macro



Connecting (Mapping)

Enter the drive letter for the shared folder (1), share name of the shared folder (2), user name (3), password (4), and then click **Map** (5).

- Consult your network administrator and enter the settings in the same way as done in the Windows PC operating system. If you enter an incorrect setting, an error might occur and the program might be interrupted.

Disconnecting

1. Enter the drive letter for the shared folder (1), and then click **Disconnect** (6).
2. Click **Exit** (7) to exit from the program.

Description of VBA Program

The program (object name: frmMapDrive) is described in detail below:

Sub CommandButton1_Click

This procedure is called when the user clicks the **Map** button. It checks whether the drive letter is used by using the IsDriveNameInUse procedure. Then the procedure connects the shared folder using the MapDrive procedure if the drive letter is not used or otherwise displays a message to show the drive letter is used.

Sub CommandButton2_Click

This procedure is called when the user clicks the **Disconnect** button. The procedure disconnects the shared folder by using the DisconnectDrive procedure.

Function IsDriveNameInUse

This procedure checks if the txtDrive.Text (the drive letter specified by 1) is used.

Sub MapDrive

This procedure connects the shared folder as the txtDrive.Text (the drive letter specified by 1) drive by using the parameters: txtShare.Text (the share name specified by 2), txtUser.Text (the user name specified by 3), and txtPasswd.Text (the password specified by 4).

Sub DisconnectDrive

This procedure disconnects the txtDrive.Text (the drive letter specified by 1) drive.

Sub CommandButton3_Click

This procedure is called when the user clicks the **Exit** button. This procedure ends the program.

Connecting the hard disk of an external PC (Object name: frmMapDrive)

```
Private Sub CommandButton1_Click()
If Not IsDriveNameInUse Then
Call MapDrive
Else
MsgBox "Drive """ & txtDrive.Text & """ is Already used", vb Critical
End If
End Sub

Private Sub CommandButton2_Click()
Call DisconnectDrive
End Sub

Private Function IsDriveNameInUse() As Boolean
Set fso = CreateObject("Scripting.FileSystemObject")
IsDriveNameInUse = fso.DriveExists(txtDrive.Text)
End Function

Private Sub MapDrive()
Set network = CreateObject("wscript.network")
Call network.MapNetworkDrive(txtDrive.Text, txtShare.Text, vbFal se, txtUser. Text, txtPasswd.Text)
End Sub

Private Sub DisconnectDrive()
Set network = CreateObject("wscript.network")
network.RemoveNetworkDrive txtDrive.Text
End Sub

Private Sub CommandButton3_Click()
Unload Me
End Sub
```

Complex Operation Library

Complex Operation Library

By using the complex operation library, you can perform operations of complex numbers.

Data of the complex type

In the complex operation library, you can use the complex type (Complex) as a data type. Data of the complex type consists of a real part (.real) and an imaginary part (.imag) as shown in the following example.

```
Dim Num as Complex
Num.real=1.0
Num.imag=2.0
```

List of procedures

The following table lists the procedures included in the complex operation library.

Procedure name	Function
ComplexSet(x,y)	Sets a complex number. (Specify a real part and an imaginary part.)
ComplexPolar(x,y)	Sets a complex number. (Specify an absolute value and a phase angle.)
ComplexSetArray(x)	Converts a variant type or double floating point type array to a complex type array.
ComplexAdd(x,y)	Returns the result of the addition.
ComplexSub(x,y)	Returns the result of the subtraction.
ComplexMul(x,y)	Returns the result of the multiplication.
ComplexDiv(x,y)	Returns the result of the division.
ComplexAbs(x)	Returns the absolute value.
ComplexArg(x)	Returns the phase angle.
ComplexNorm(x)	Returns the square of the absolute value.
ComplexConj(x)	Returns the conjugate complex number.
ComplexCos(x)	Returns the cosine.

ComplexCosh(x)	Returns the hyperbolic cosine.
ComplexSin(x)	Returns the sine.
ComplexSinh(x)	Returns the hyperbolic sine.
ComplexExp(x)	Returns e^x.
ComplexLog(x)	Returns the natural logarithm.
ComplexLog10(x)	Returns the common logarithm.
ComplexSqrt(x)	Returns the square root.

Sample Program

```
:
:

Dim Dmy As Long
Dim s21_raw As Variant
Dim s31_raw As Variant
Dim s21_Comp As Complex
Dim s31_Comp As Complex
Dim trAce_ratio_comp As Complex
Dim trAce_ratio(401) As Double

SCPI.DISPlay.Split = "D1"
SCPI.DISPlay.WINDow(1).Split = "D12_34"
SCPI.CALCulate(1).PARameter.Count = 2
SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
SCPI.CALCulate(1).PARameter(2).DEFine = "s31"
SCPI.SENSe(1).SWEep.POINts = 201

:
:
:

SCPI.TRIGger.SEQuence.Source = "bus"
SCPI.TRIGger.SEQuence.SINGle
Dmy = SCPI.IEEE4882.OPC
```

```
'''' Get corrected data array
SCPI.CALCulate(1).PARameter(1).SELect
s21_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
SCPI.CALCulate(1).PARameter(2).SELect
s31_raw = SCPI.CALCulate(1).SELected.DATA.SDATa

For i = 0 To 200

'''' Copy corrected data array to the complex data array
'''' to take advantage of complex operation library
s21_Comp = ComplexSet(s21_raw(2 * i), s21_raw(2 * i + 1))
s31_Comp = ComplexSet(s31_raw(2 * i), s31_raw(2 * i + 1))

'''' Calculate the ratio of S31 and S21
'''' S31/S21
trAce_ratio_comp = ComplexDiv(s31_Comp, s21_Comp)

trAce_ratio(2 * i) = trAce_ratio_comp.real
trAce_ratio(2 * i + 1) = trAce_ratio_comp.imag

Next i

SCPI.CALCulate(1).PARameter.Count = 4

'''' Write "S31/S21" data to corrected data array for the trace 3 (LogMag)
SCPI.CALCulate(1).PARameter(3).SELect
SCPI.CALCulate(1).SELected.Format = "MLOG"
SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio

'''' Write "S31/S21" data to corrected data array for the trace 4 (Phase)
SCPI.CALCulate(1).PARameter(4).SELect
SCPI.CALCulate(1).SELected.Format = "PHASe"
SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio

:
:
```

Complex Operation Library

By using the complex operation library, you can perform operations of complex numbers.

Data of the complex type

In the complex operation library, you can use the complex type (Complex) as a data type. Data of the complex type consists of a real part (.real) and an imaginary part (.imag) as shown in the following example.

```
Dim Num as Complex
Num.real=1.0
Num.imag=2.0
```

List of procedures

The following table lists the procedures included in the complex operation library.

Procedure name	Function
ComplexSet(x,y)	Sets a complex number. (Specify a real part and an imaginary part.)
ComplexPolar(x,y)	Sets a complex number. (Specify an absolute value and a phase angle.)
ComplexSetArray(x)	Converts a variant type or double floating point type array to a complex type array.
ComplexAdd(x,y)	Returns the result of the addition.
ComplexSub(x,y)	Returns the result of the subtraction.
ComplexMul(x,y)	Returns the result of the multiplication.
ComplexDiv(x,y)	Returns the result of the division.
ComplexAbs(x)	Returns the absolute value.
ComplexArg(x)	Returns the phase angle.
ComplexNorm(x)	Returns the square of the absolute value.
ComplexConj(x)	Returns the conjugate complex number.
ComplexCos(x)	Returns the cosine.
ComplexCosh(x)	Returns the hyperbolic cosine.

ComplexSin(x)	Returns the sine.
ComplexSinh(x)	Returns the hyperbolic sine.
ComplexExp(x)	Returns e^x.
ComplexLog(x)	Returns the natural logarithm.
ComplexLog10(x)	Returns the common logarithm.
ComplexSqrt(x)	Returns the square root.

Sample Program

```
:
:

Dim Dmy As Long
Dim s21_raw As Variant
Dim s31_raw As Variant
Dim s21_Comp As Complex
Dim s31_Comp As Complex
Dim trAce_ratio_comp As Complex
Dim trAce_ratio(401) As Double

SCPI.DISPlay.Split = "D1"
SCPI.DISPlay.WINDow(1).Split = "D12_34"
SCPI.CALCulate(1).PARameter.Count = 2
SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
SCPI.CALCulate(1).PARameter(2).DEFine = "s31"
SCPI.SENSe(1).SWEep.POINts = 201

:
:
:

SCPI.TRIGger.SEQuence.Source = "bus"
SCPI.TRIGger.SEQuence.SINGle
Dmy = SCPI.IEEE4882.OPC

"" Get corrected data array
```

```
SCPI.CALCulate(1).PARameter(1).SELect
s21_raw = SCPI.CALCulate(1).SELected.DATA.SDATa
SCPI.CALCulate(1).PARameter(2).SELect
s31_raw = SCPI.CALCulate(1).SELected.DATA.SDATa

For i = 0 To 200

"" Copy corrected data array to the complex data array
"" to take advantage of complex operation library
s21_Comp = ComplexSet(s21_raw(2 * i), s21_raw(2 * i + 1))
s31_Comp = ComplexSet(s31_raw(2 * i), s31_raw(2 * i + 1))

"" Calculate the ratio of S31 and S21
"" S31/S21
trAce_ratio_comp = ComplexDiv(s31_Comp, s21_Comp)

trAce_ratio(2 * i) = trAce_ratio_comp.real
trAce_ratio(2 * i + 1) = trAce_ratio_comp.imag

Next i

SCPI.CALCulate(1).PARameter.Count = 4

"" Write "S31/S21" data to corrected data array for the trace 3 (LogMag)
SCPI.CALCulate(1).PARameter(3).SELect
SCPI.CALCulate(1).SELected.Format = "MLOG"
SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio

"" Write "S31/S21" data to corrected data array for the trace 4 (Phase)
SCPI.CALCulate(1).PARameter(4).SELect
SCPI.CALCulate(1).SELected.Format = "PHASe"
SCPI.CALCulate(1).SELected.DATA.SDATa = trAce_ratio

:
:
```

Procedure Reference

ComplexAbs(*x*)

Syntax

Result = ComplexAbs(*x*)

Description

Returns the absolute value of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double
a = ComplexSet(1.5, 2.0)
b = ComplexAbs(a)
```

ComplexAdd(*x*,*y*)

Syntax

Result = ComplexAdd(*x*,*y*)

Description

Returns the result (x+y) of the addition of a complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexAdd(a, b)
```

ComplexArg(*x*)

Syntax

Result = ComplexArg(*x*)

Description

Returns the phase angle (radian) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double, c As Double, pi As Double
a = ComplexSet(1.5, 2.0)
b = ComplexArg(a)
pi = 3.14159265
c = b * 180 / pi      ‘ radian -> degree
```

ComplexConj(*x*)

Syntax

Result = ComplexConj(*x*)

Description

Returns the conjugate complex number of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexConj(a)
```

ComplexCos(*x*)

Syntax

Result = ComplexCos(*x*)

Description

Returns the cosine (cos(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexCos(a)
```

ComplexCosh(*x*)

Syntax

Result = ComplexCosh(*x*)

Description

Returns the hyperbolic cosine (cosh(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexCosh(a)
```

ComplexDiv(*x*,*y*)

Syntax

Result = ComplexDiv(*x*,*y*)

Description

Returns the result (x/y) of the division of a complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexDiv(a, b)
```

ComplexExp(*x*)

Syntax

Result = ComplexExp(*x*)

Description

Returns e*x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexExp(a)
```

ComplexLog(*x*)

Syntax

Result = ComplexLog(*x*)

Description

Returns the natural logarithm (log(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexLog(a)
```

ComplexLog10(*x*)

Syntax

Result = ComplexLog(*x*)

Description

Returns the common logarithm (log 10 (*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)
Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexLog10(a)
```

ComplexMul(*x*,*y*)

Syntax

Result = ComplexMul(*x*,*y*)

Description

Returns the result (x\ y) of the multiplication of a complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexMul(a, b)
```

ComplexNorm(*x*)

Syntax

Result = ComplexNorm(*x*)

Description

Returns the square of the absolute value of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Double precision floating point type (Double)

Example of use

```
Dim a As Complex, b As Double
a = ComplexSet(1.5, 2.0)
b = ComplexNorm(a)
```

ComplexPolar(*x*,*y*)

Syntax

z = ComplexPolar(*x*,*y*)

Description

Sets a complex number to a complex type variable *z*. Specify a complex number with an absolute value *x* and a phase angle *y (radian)*.

Data type

x : Double precision floating point type (Double)

y : Double precision floating point type (Double)

z : Complex type (Complex)

Example of use

```
Dim a As Complex, pi As Double
pi = 3.14159265
a = ComplexPolar(2.5, 60 * pi / 180)
```

ComplexSet(*x*,*y*)

Syntax

z = ComplexSet(*x*,*y*)

Description

Sets a complex number to a complex type variable *z*. Specify a complex number with a real part *x* and an imaginary part *y*. (Sets *x* and *y* to *z.real* and *z.imag* respectively.)

Data type

x : Double precision floating point type (Double)

y : Double precision floating point type (Double)

z : Complex type (Complex)

Example of use

```
Dim a as Complex
a = ComplexSet(1.5, 2.0)
```

ComplexSetArray(*x*)

Syntax

y = ComplexSetArray(*x*)

Description

Converts a variant type or double floating point type array *x* that contains complex numbers using 2 elements to store each complex number in the order of the real part and imaginary part to a complex type array *y*.

Data type

x : Variant type (Variant) array or Double precision floating point type (Double) array

y : Complex type (Complex) array

Example of use

```
Dim a as Variant, b as Complex
a = SCPI.CALCulate(1).SELected.DATA.SDATa
b = ComplexSetArray(a)
```

ComplexSin(*x*)

Syntax

Result = ComplexSin(*x*)

Description

Returns the sine (sin(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSin(a)
```

ComplexSinh(*x*)

Syntax

Result = ComplexSinh(*x*)

Description

Returns the hyperbolic sine (sinh(*x*)) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSinh(a)
```

ComplexSqrt(*x*)

Syntax

Result = ComplexSqrt(*x*)

Description

Returns the square root ( ) of a complex number *x*.

Data type

x : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSqrt(a)
```

ComplexSub(*x*,*y*)

Syntax

Result = ComplexSub(*x*,*y*)

Description

Returns the result (x - y) of the subtraction of a complex number *x* and another *y*.

Data type

x : Complex type (Complex)

y : Complex type (Complex)

Result : Complex type (Complex)

Example of use

```
Dim a As Complex, b As Complex, c As Complex
a = ComplexSet(1.5, 2.0)
b = ComplexSet(0.5, 3.5)
c = ComplexSub(a, b)
```

Waveform Analysis Library

Waveform (Ripple) Analysis Library

- Overview
- Flow of Programming using Ripple Analysis Library
- Condition Setting before using Ripple Analysis Library
- List of Ripple Analysis Library
- Sample Program

Other topics about VBA Programming

Overview

By combining the COM objects provided for the E5071C and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

Flow of Programming using Ripple Analysis Library

Below table shows the flow of program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

1. Condition setting before using the ripple analysis library
    - Specifying the analysis range
    - Setting the peak definition
2. Using the ripple analysis library

Condition Setting before using Ripple Analysis Library

Since the analysis conditions are not specified in the ripple analysis library, before using the procedure for ripple analysis, set up the analysis range and the peak definition using COM objects.

Specifying the Analysis Range

Use the following COM objects to specify the analysis range for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion

- SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

List of Ripple Analysis Library

Use the provided procedures for ripple analysis to analyze the ripple of waveforms and output the result. All procedures perform analysis only within the stimulus range for the specified channel.

Function Name	Description
MaxPeakToPeak(Chan)	Returns the maximum value of the difference between a positive peak and a negative peak
MaxRightGap(Chan)	Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.
MaxLeftGap(Chan)	Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.
MaxGap(Chan)	Returns the maximum value of the difference between a positive peak and its adjacent negative peak.
MaxEnvelopeGap(Chan)	Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.
GapMean(Chan)	Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRippleValue(Chan)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRipplePoint(Chan,Stim)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks and the stimulus value (*Stim*) of the valley

	of the ripple.
Pole(Chan,D,LeftStim,LeftValue, RightStim,RightValue)	Returns the values (*LeftValue* and *RightValue*) and the stimulus values (*LeftStimulus* and *RightStimulus*) of the right and left negative peaks detected first below the specified value (*D*) relative to the maximum value.
FirstRightGap(Chan)	Returns the difference between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftGap(Chan)	Returns the difference between the positive peak detected first when searched from the right edge toward the left edge and its left adjacent negative peak.
FirstRightInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its left adjacent negative peak.

Sample Program

Here is a simple sample program using the ripple analysis procedures.

```
Sub Sample()

Dim Val As Double    (1)

SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 1.5    (2)
SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "BOTH"    (2)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 935E6    (3)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 960E6    (3)
```

```
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True    (3)
  .
  .
Val = MaxPeakToPeak(1)     (4)

End Sub
```

Let us break down the code into a number of blocks and see what they do.

1. Defines a variable Val as Double.
2. Sets the lower limit of the peak excursion value and polarity for the peak search to 1.5 and both of positive peak and negative peak, respectively.
3. Sets the analysis range for channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitutes the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

Waveform (Ripple) Analysis Library

- Overview
- Flow of Programming using Ripple Analysis Library
- Condition Setting before using Ripple Analysis Library
- List of Ripple Analysis Library
- Sample Program

Other topics about VBA Programming

Overview

By combining the COM objects provided for the E5071C and the ripple analysis library, you can easily perform the ripple analysis of waveforms.

Flow of Programming using Ripple Analysis Library

Below table shows the flow of program development using the ripple analysis library. First, set up the analysis range and peak definition to use the procedures for ripple analysis.

1. Condition setting before using the ripple analysis library
    - Specifying the analysis range
    - Setting the peak definition
2. Using the ripple analysis library

Condition Setting before using Ripple Analysis Library

Since the analysis conditions are not specified in the ripple analysis library, before using the procedure for ripple analysis, set up the analysis range and the peak definition using COM objects.

Specifying the Analysis Range

Use the following COM objects to specify the analysis range for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe
- SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

Setting the Peak Definition

Use the following COM objects to set up the peak definition for ripple analysis.

- SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion
- SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

List of Ripple Analysis Library

Use the provided procedures for ripple analysis to analyze the ripple of waveforms and output the result. All procedures perform analysis only within the stimulus range for the specified channel.

Function Name	Description
MaxPeakToPeak(Chan)	Returns the maximum value of the difference between a positive peak and a negative peak
MaxRightGap(Chan)	Returns the maximum value of the difference between a positive peak and its right adjacent negative peak.
MaxLeftGap(Chan)	Returns the maximum value of the difference between a positive peak and its left adjacent negative peak.
MaxGap(Chan)	Returns the maximum value of the difference between a positive peak and its adjacent negative peak.
MaxEnvelopeGap(Chan)	Returns the maximum value of the vertical distance between a line segment connecting 2 adjacent positive peaks and the negative peak between them.
GapMean(Chan)	Returns the mean value of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRippleValue(Chan)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks.
MaxRipplePoint(Chan,Stim)	Returns the maximum value of the total of the differences between a negative peak and its right and left adjacent positive peaks and the stimulus value (*Stim*) of the valley of the ripple.

Pole(Chan,D,LeftStim,LeftValue, RightStim,RightValue)	Returns the values (*LeftValue* and *RightValue*) and the stimulus values (*LeftStimulus* and *RightStimulus*) of the right and left negative peaks detected first below the specified value (*D*) relative to the maximum value.
FirstRightGap(Chan)	Returns the difference between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftGap(Chan)	Returns the difference between the positive peak detected first when searched from the right edge toward the left edge and its left adjacent negative peak.
FirstRightInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its right adjacent negative peak.
FirstLeftInterval(Chan)	Returns the difference of the stimulus value between the positive peak detected first when searched from the left edge toward the right edge and its left adjacent negative peak.

Sample Program

Here is a simple sample program using the ripple analysis procedures.

```
Sub Sample()

Dim Val As Double    (1)

SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 1.5    (2)
SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "BOTH"    (2)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 935E6    (3)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 960E6    (3)
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True    (3)
 .
```

```
.
Val = MaxPeakToPeak(1)     (4)

End Sub
```

Let us break down the code into a number of blocks and see what they do.

1. Defines a variable Val as Double.
2. Sets the lower limit of the peak excursion value and polarity for the peak search to 1.5 and both of positive peak and negative peak, respectively.
3. Sets the analysis range for channel 1 to 935 MHz to 960 MHz.
4. For channel 1, substitutes the return value from the MaxPeakToPeak function (procedure) in the ripple analysis library to the Val variable.

Procedure Reference

FirstLeftGap(*Chan*)

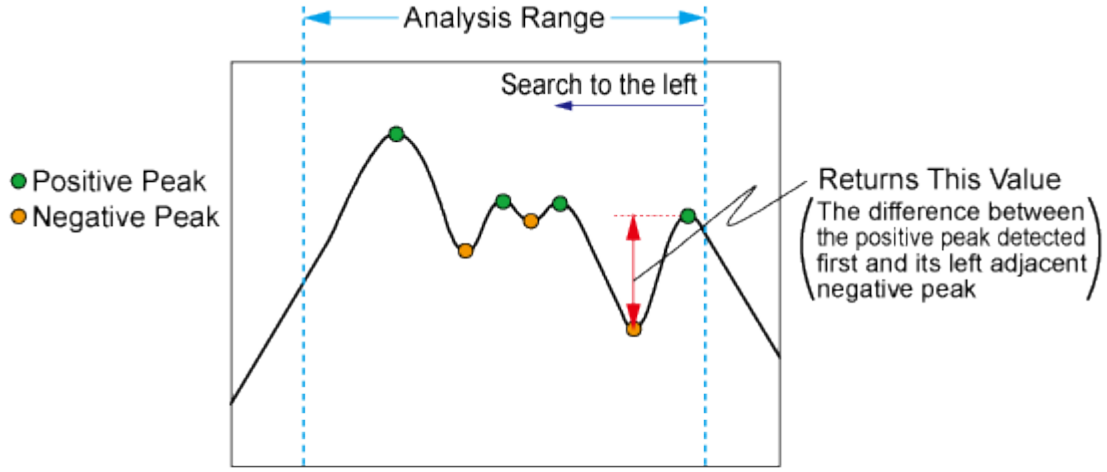
Syntax

Value = FirstLeftGap(*Chan*)

Description

Returns the response difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

FirstLeftGap



e5071c431

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the response difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = FirstLeftGap(1)
MsgBox "First Left Gap =" & Value
```

FirstLeftInterval(*Chan*)

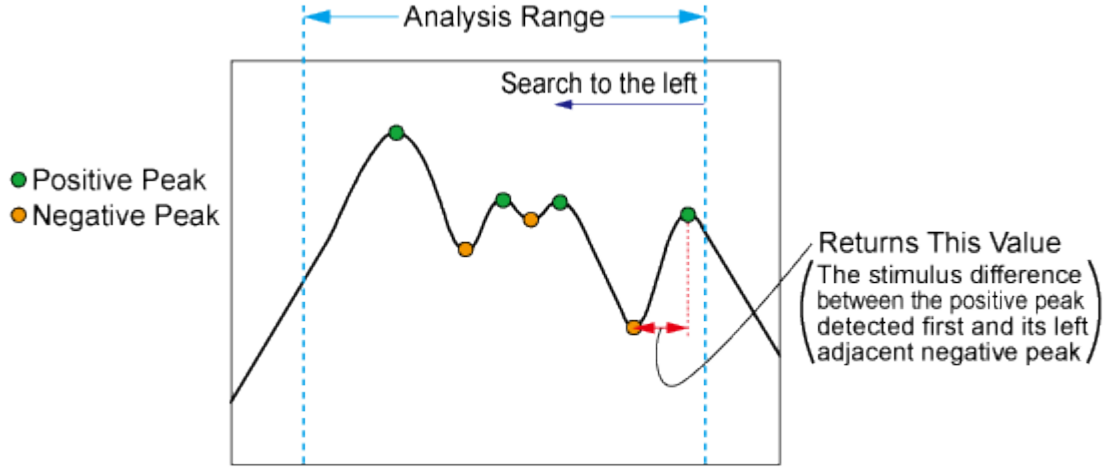
Syntax

Value = FirstLeftInterval(*Chan*)

Description

Returns the stimulus difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.

FirstLeftInterval



e5071c432

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the stimulus difference between the positive peak detected first when searched from the right edge toward the left edge within the analysis range and its left adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = FirstLeftInterval(1)
MsgBox "First Left Interval =" & Value
```

FirstRightGap(*Chan*)

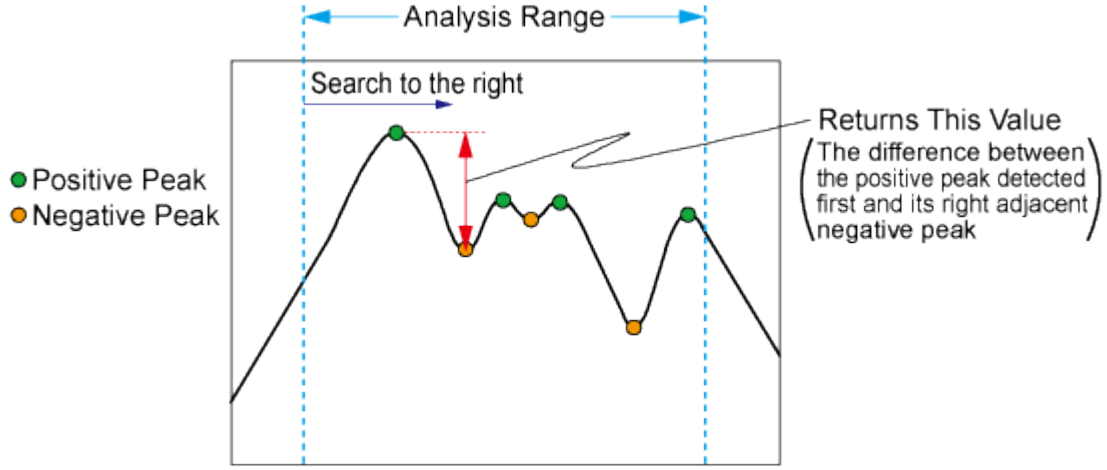
Syntax

Value = FirstRightGap(*Chan*)

Description

Returns the response difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

FirstRightGap



e5071c434

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the response difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = FirstRightGap(1)
MsgBox "First Right Gap =" & Value
```

FirstRightInterval(*Chan*)

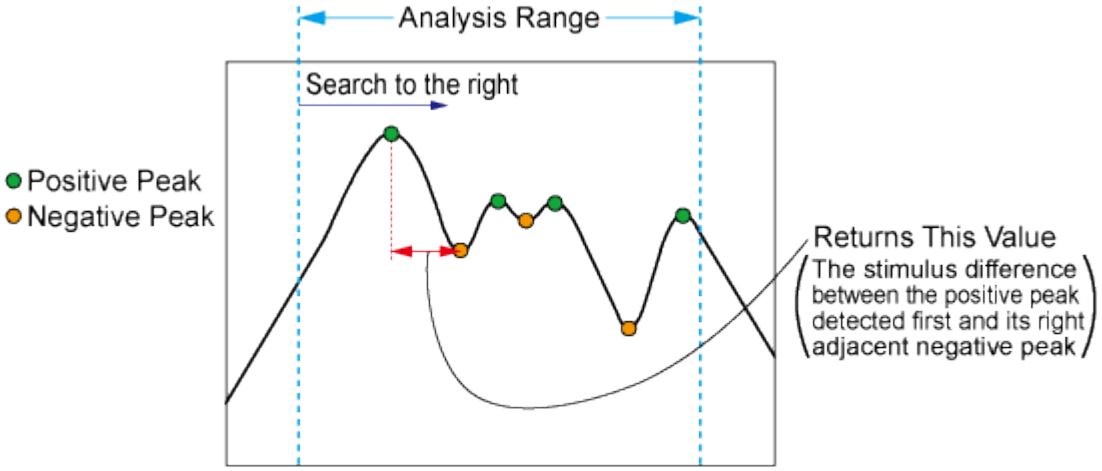
Syntax

Value = FirstRightInterval(*Chan*)

Description

Returns the stimulus difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.

FirstRightInterval



e5071c433

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the stimulus difference between the positive peak detected first when searched from the left edge toward the right edge within the analysis range and its right adjacent negative peak.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = FirstRightInterval(1)
MsgBox "First Right Interval =" & Value
```

GapMean(*Chan*)

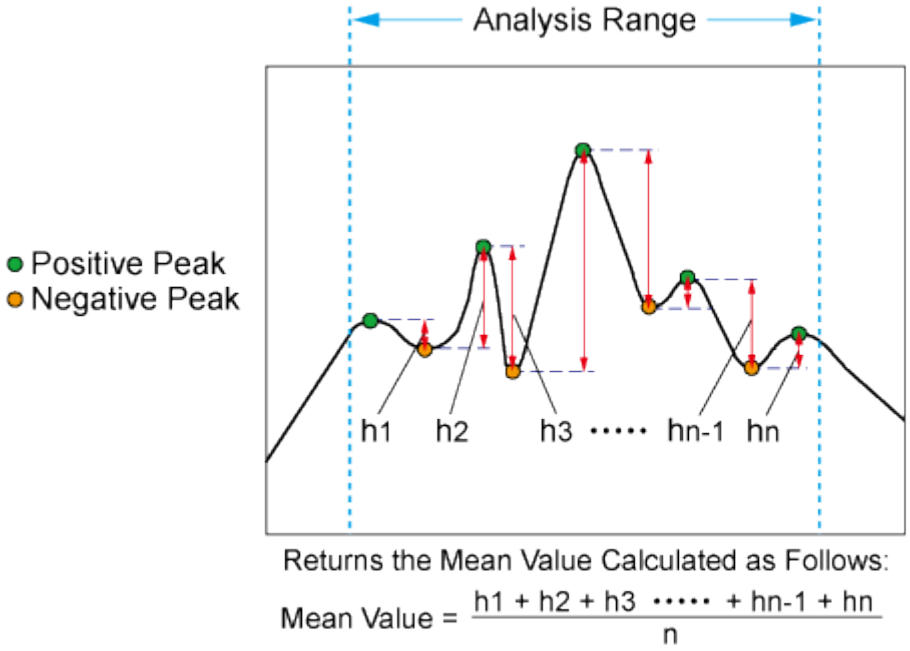
Syntax

Value = GapMean(*Chan*)

Description

Returns the mean value of the response differences between the negative peaks and its adjacent positive peaks within the analysis range.

GapMean



e5071c427

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the mean value of the response differences between the negative peaks and its right and left adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = GapMean(1)
MsgBox "Gap Mean =" & Value
```

MaxEnvelopeGap(*Chan*)

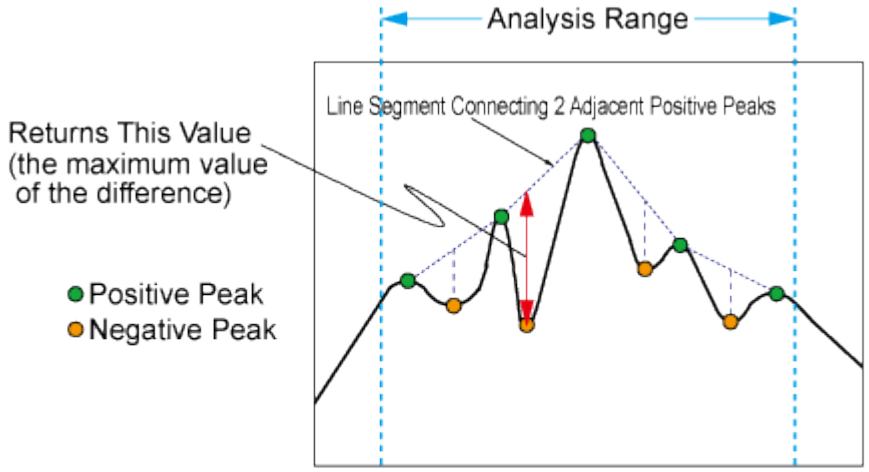
Syntax

Value = MaxEnvelopeGap(*Chan*)

Description

Returns the maximum value of the vertical distance between the line segments connecting 2 adjacent positive peaks and the negative peaks between them within the analysis range.

MaxEnvelopeGap



e5071c426

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the vertical distance between the line segments connecting 2 adjacent positive peaks and the negative peaks between them.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxEnvelopeGap(1)
MsgBox "Max Envelope Gap =" & Value
```

MaxGap(*Chan*)

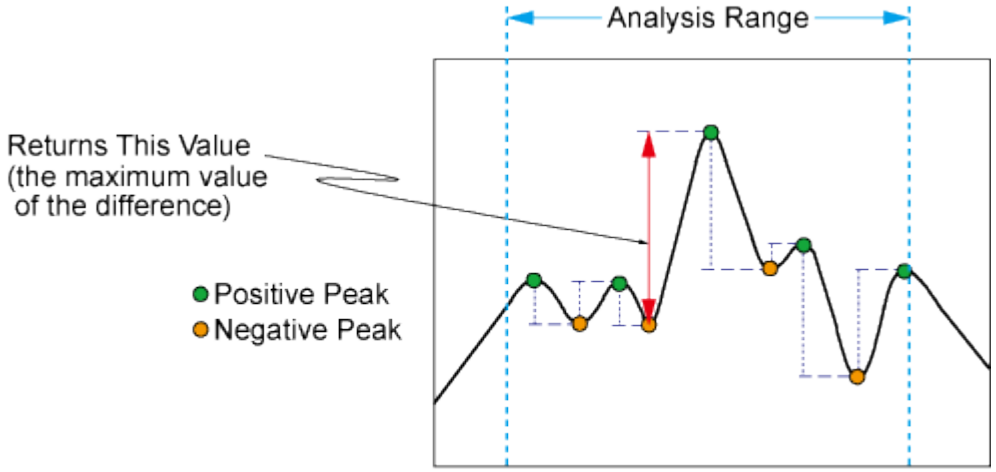
Syntax

Value = MaxGap(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its adjacent negative peaks within the analysis range.

MaxGap



e5071c425

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and its adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxGap(1)
MsgBox "Max Gap =" & Value
```

MaxLeftGap(*Chan*)

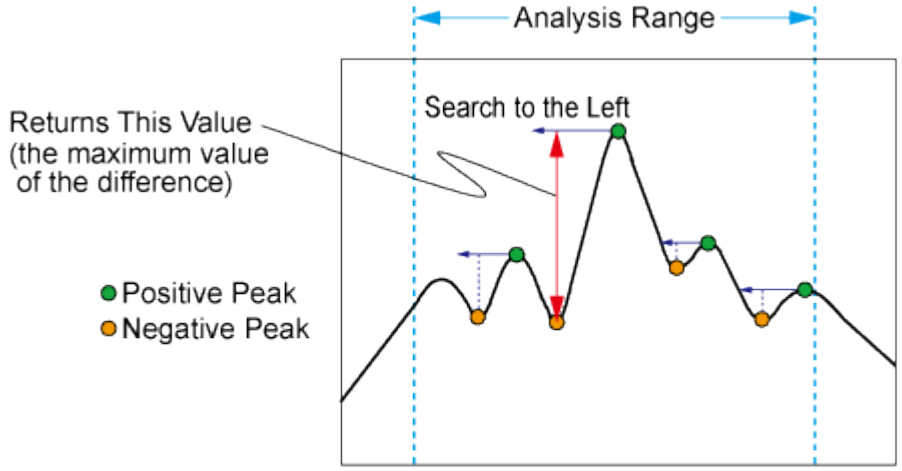
Syntax

Value = MaxLeftGap(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its left adjacent negative peaks within the analysis range.

MaxLeftGap



e5071c424

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and its left adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxLeftGap(1)
MsgBox "Max Left Gap =" & Value
```

MaxPeakToPeak(*Chan*)

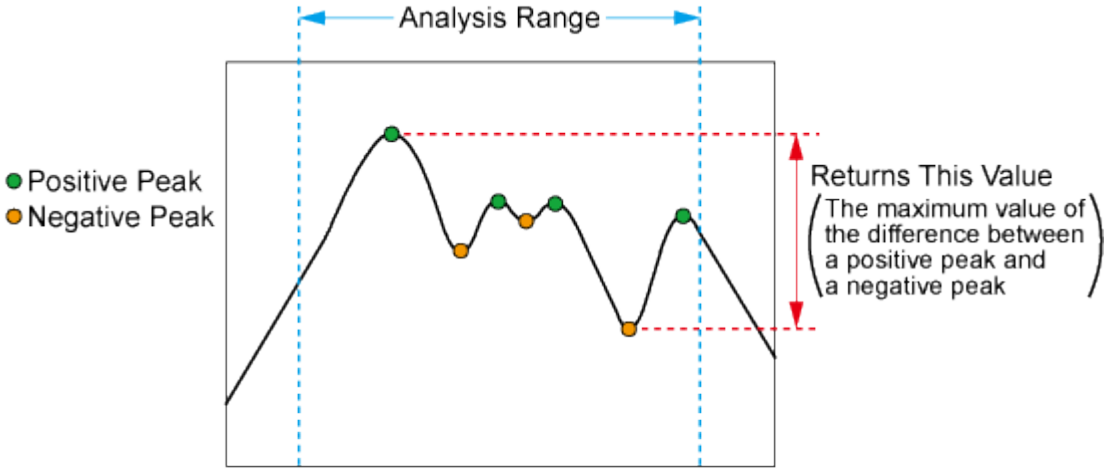
Syntax

Value = MaxPeakToPeak(*Chan*)

Description

Returns the maximum value of the response differences between the positive peaks and the negative peaks within the analysis range.

MaxPeakToPeak



e5071c422

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

1490

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and the negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxPeakToPeak(1)
MsgBox "Max Peak To Peak =" & Value
```

MaxRightGap(*Chan*)

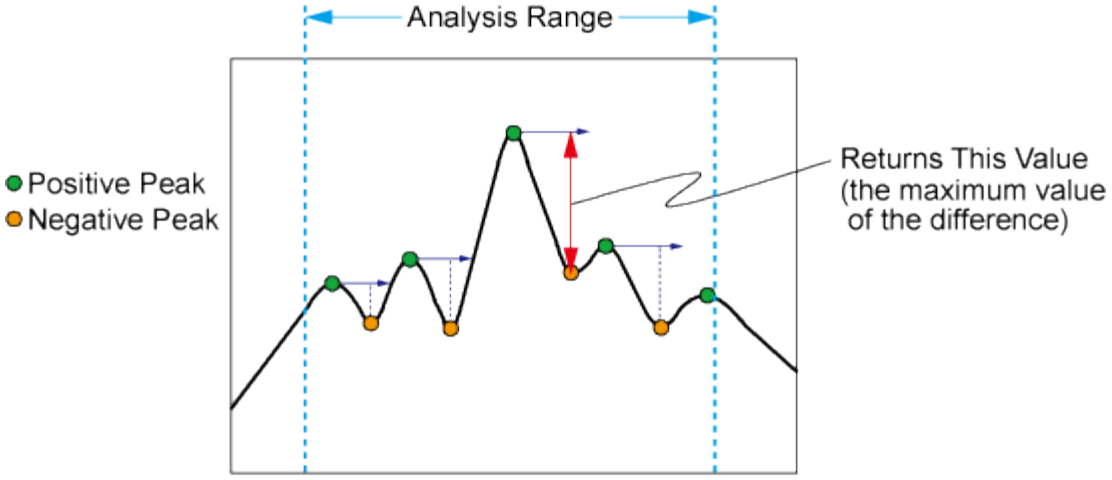
Syntax

Value = MaxRightGap(*chan*)

Description

Returns the maximum value of the response differences between the positive peaks and its right adjacent negative peaks within the analysis range.

MaxRightGap



e5071c423

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the response differences between the positive peaks and its right adjacent negative peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double

Value = MaxRightGap(1)
MsgBox "Max Right Gap =" & Value
```

MaxRipplePoint(*Chan*,*Stim*)

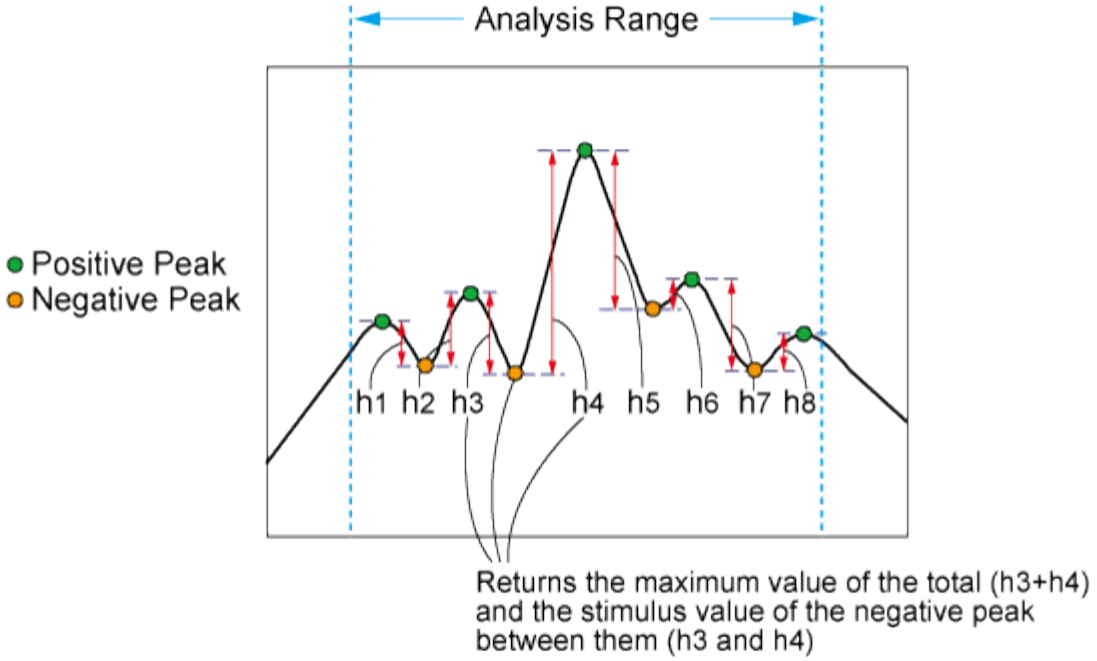
Syntax

Value = MaxRipplePoint(*Chan*,*Stim*)

Description

Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks and the stimulus value of the applicable negative peaks within the analysis range.

MaxRipplePoint



e5071c428

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*Stim*
Description	Returns the stimulus value of the negative peak at which the sum of the response differences between the negative peak and its adjacent positive peaks is maximum.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Dim Stim As Double

Value = MaxRipplePoint(1, Stim)
MsgBox "Max Ripple Value =" & Value & " , Stimulus =" & Stim
```

MaxRippleValue(*Chan*)

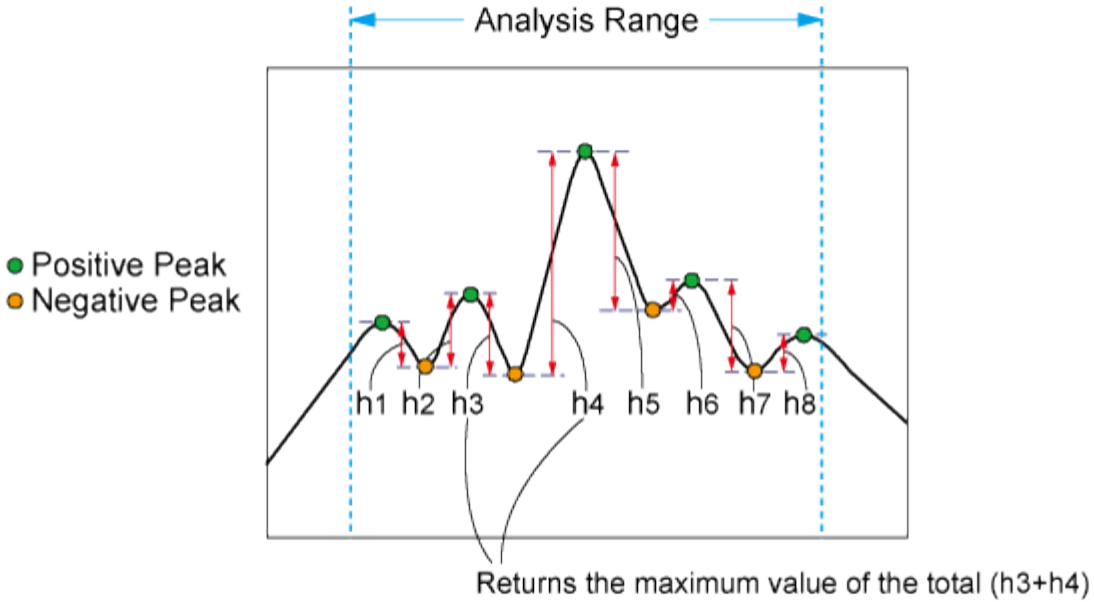
Syntax

Value = MaxRippleValue(*Chan*)

Description

Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks within the analysis range.

MaxRippleValue



e5071c429

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)
Range	1 to 9

Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Return value

Parameter	*Value*
Description	Returns the maximum value of the sum of the response differences between the negative peaks and its adjacent positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim Value As Double
Value = MaxRippleValue(1)
MsgBox "Max Ripple Value =" & Value
```

Pole(*Chan*,*D*,*LeftStim*,*LeftValue*,*RightStim*,*RightValue***)**

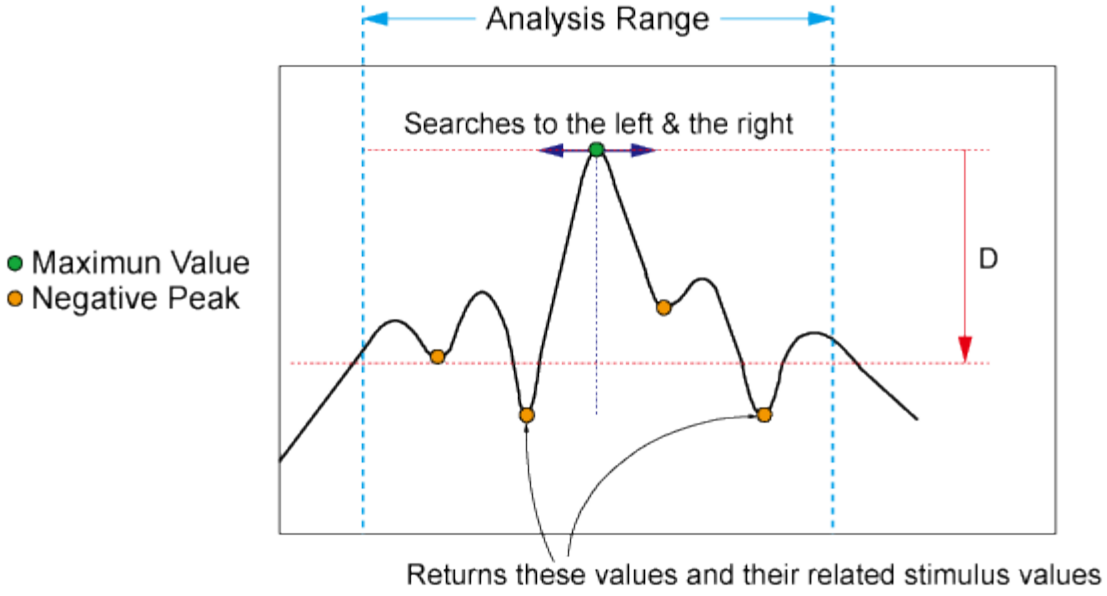
Syntax

Call Pole(*Chan*,*D*,*LeftStim*,*LeftValue*,*RightStim*,*RightValue*)

Description

For the negative peaks below the specified value (*D*) relative to the maximum value of the positive peaks within the analysis range, returns the response value (*LeftValue*) and stimulus value (*LeftStimulus*) of the negative peak first detected when searched to the left from the maximum value of the positive peaks, and the response value (*RightValue*) and stimulus value (*RightStimulus*) of the negative peak first detected when searched to the right from the maximum value of the positive peaks.

Pole



e5071c430

Variable

Parameter	*Chan*
Description	Specifies the channel number.
Data type	Integer type (Integer)

Range	1 to 9
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*D*
Description	Specifies the difference from the maximum value.
Data type	Double precision floating point type (Double)

Return value (arguments)

Parameter	*LeftStim*
Description	Returns the stimulus value of the negative peak first detected to the left from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*LeftValue*
Description	Returns the response value of the negative peak first detected to the left from the maximum value of the positive peaks.

Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*RightStim*
Description	Returns the stimulus value of the negative peak first detected to the right from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Parameter	*RightValue*
Description	Returns the response value of the negative peak first detected to the right from the maximum value of the positive peaks.
Data type	Double precision floating point type (Double)
Note	If no applicable point is detected, 0 is returned.

Example of use

```
Dim LeftStim As Double
Dim LeftValue As Double
Dim RightStim As Double
Dim RightValue As Double

Call Pole(1, 1, LeftStim, LeftValue, RightStim, RightValue)
```

```
MsgBox "Left Pole =" & LeftStim & ":" & LeftValue
MsgBox "Right Pole =" & RightStim & ":" & RightValue
```

TDR Remote Control

Option TDR Remote Control

- Concepts of Remote Control in TDR
- Setup and Connection of TDR
- TDR Sample Program

Other topics about Programming

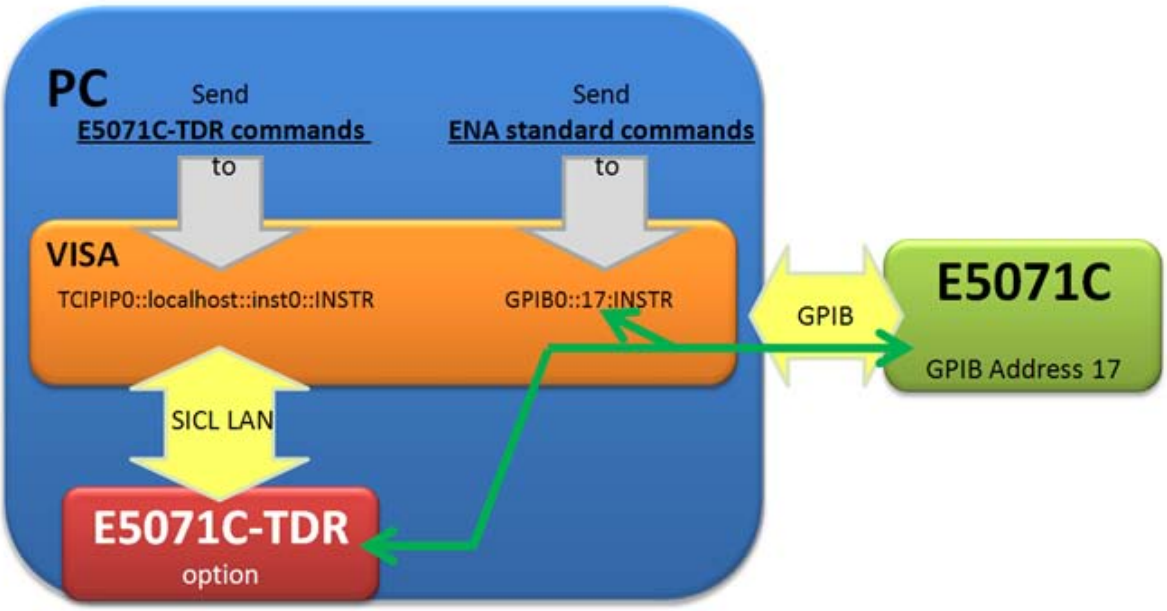
Concept of Remote Control in TDR

Enhanced Time Domain Analysis (E5071C-TDR) can be controlled remotely.

The figure below shows the connection between a PC, E5071C and Enhanced Time Domain Analysis option. VISA application treats both E5071C and Enhanced Time Domain Analysis option as instruments.

As such, VISA application connects to E5071C via GPIB through GPIB0::17:INSTR address. Standard E5071C commands are sent through GPIB to the E5071C.

Whereas, VISA application connects to Enhanced Time Domain Analysis option via SICL LAN through TCPIP0::localhost::inst0::INSTR address. E5071C-TDR commands are sent through SICL LAN to the Enhanced Time Domain Analysis option.



To learn more about other remote control system, refer to Types of remote control system.

SCPI commands and COM commands are used to control E5071C remotely. However, only SCPI commands can be used to control the Enhanced Time Domain Analysis option. Therefore, it is not possible to control it by builit-in VBA.

Other topics about TDR Remote Control

Setup and Connection of TDR

- Installing E5071C-TDR application to PC
- Connecting E5071C-TDR with PC

Other topics about TDR Remote Control

Installing E5071C-TDR application to PC

Three applications need to be installed in order for E5071C-TDR to be worked correctly:

Installing .Net Framework Version 2.0 and above

1. Ensure that .Net Framework Version 2.0 and above is installed in the PC. Go to Windows Control Panel > Add or Remove Programs to confirm if it is installed. If it is not installed, go to below URL and download the latest version.

Installing Agilent IO Library

1. Ensure that Agilent IO Library (Version 15.5) is installed in the PC. If it is not installed, go to below URL and download the latest Agilent IO Libraries Suite executable file:

IO library 15.5 download

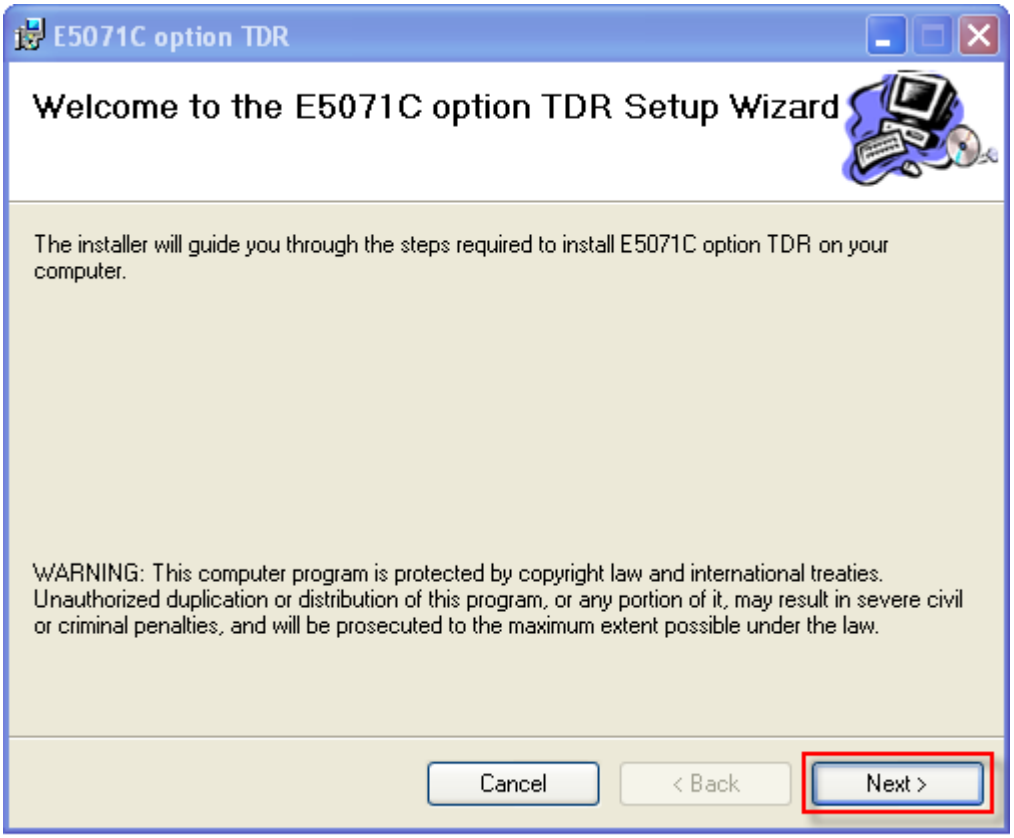
2. Double-click on the executable file for installation. Follow the instruction of the installation wizard.

Installing E5071C-TDR Software

1.

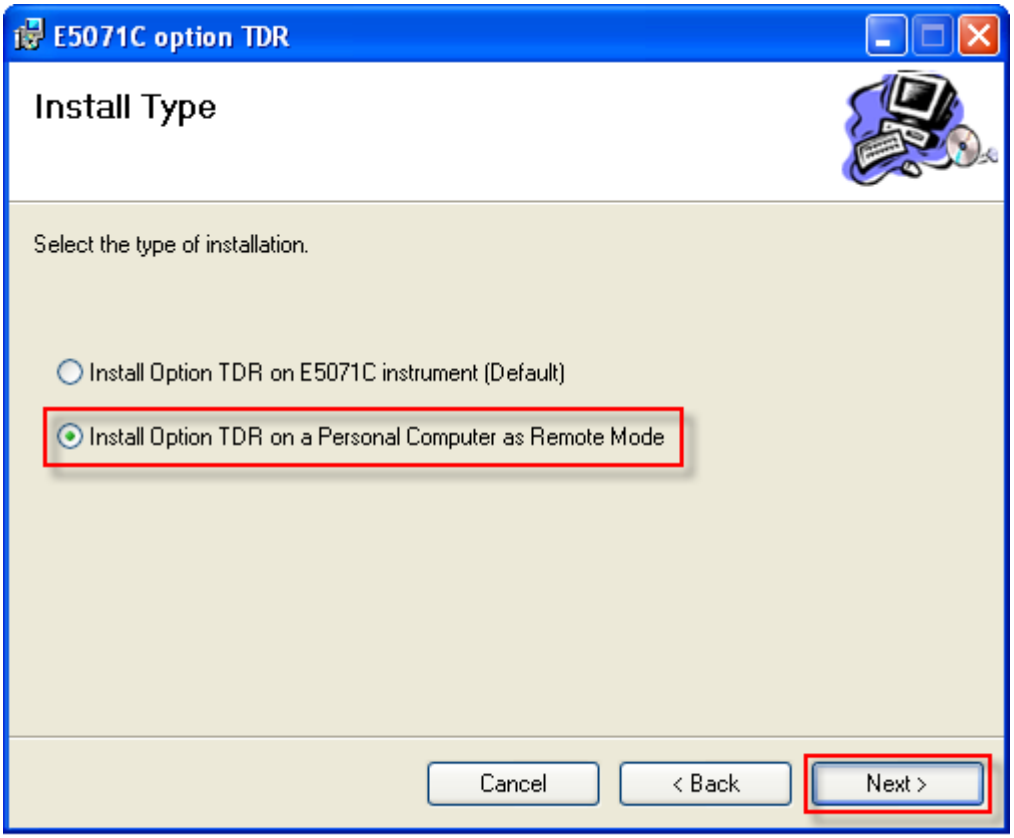
NOTE Install the same revision of the E5071C-TDR application into the instrument and the PC. This ensure the remote access to the instrument works correctly

2. Download the execution file (**E5071C_opt_TDR_xx.xx.exe**).
3. Double-click on the executable file, **E5071C_opt_TDR.msi**.
4. Click **Next >**.



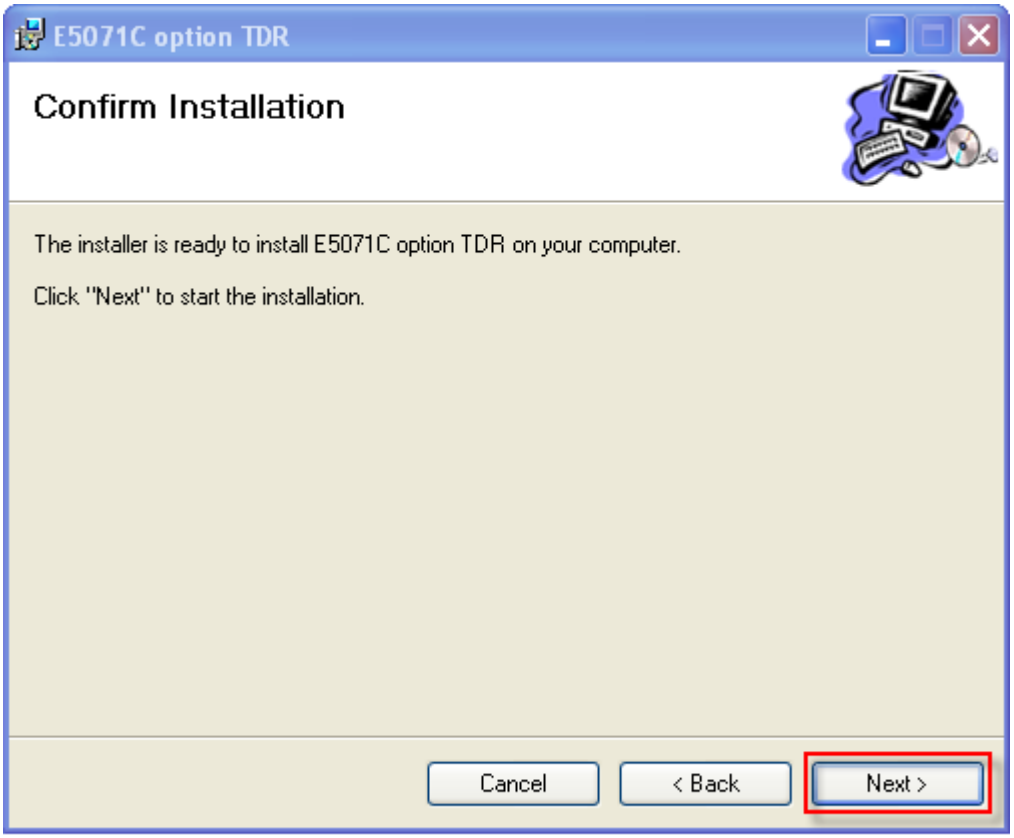
e5071c521

3. Select Install Option TDR on a Personal Computer as Remote Mode and click **Next >**.



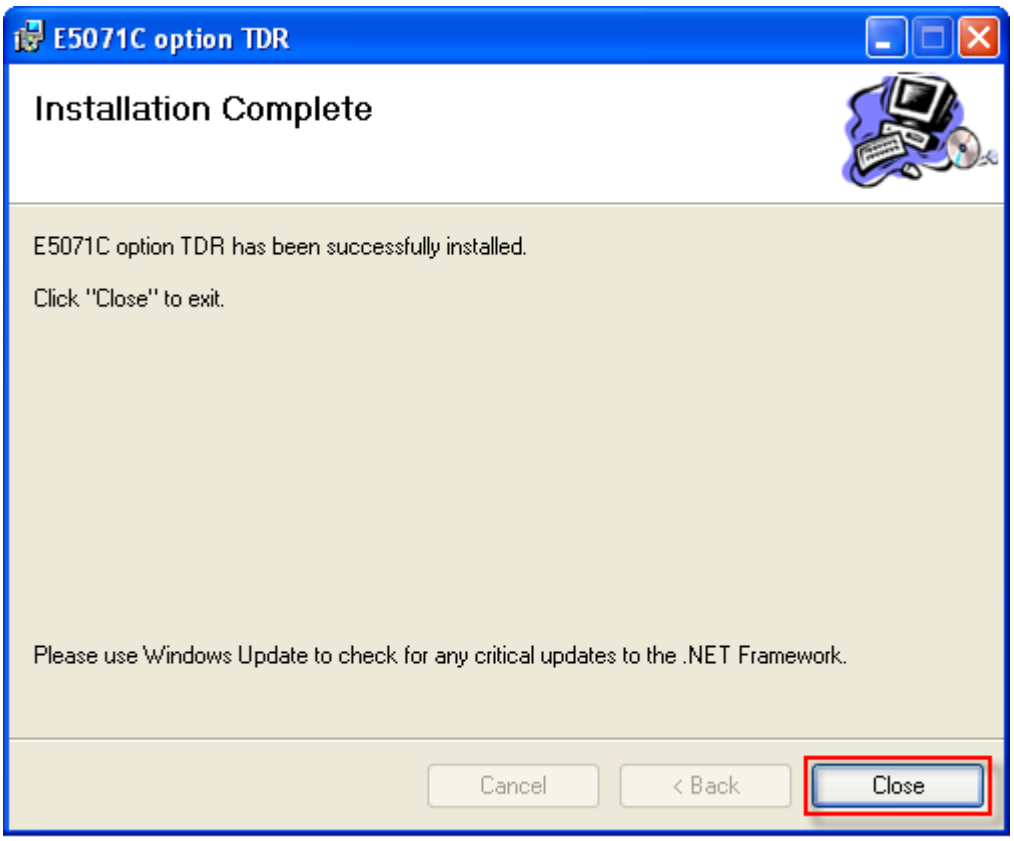
e5071c522

4. Click **Next >**.



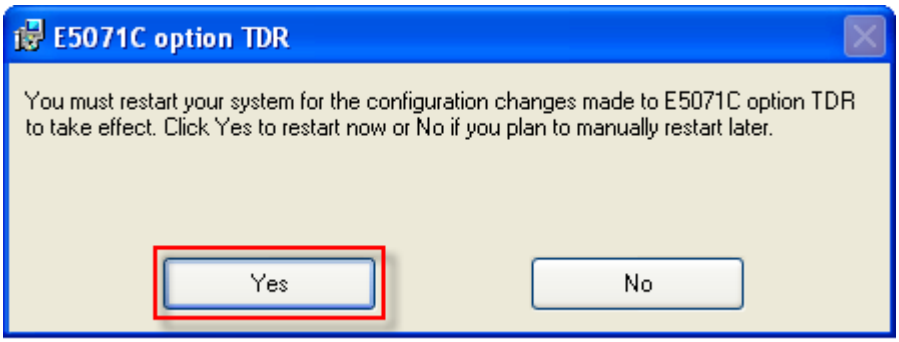
e5071c523

5. Click **Close** to complete the installation.



e5071c524

6. E5071C option TDR window appears. Press **Yes** when ready to reboot.



e5071c525

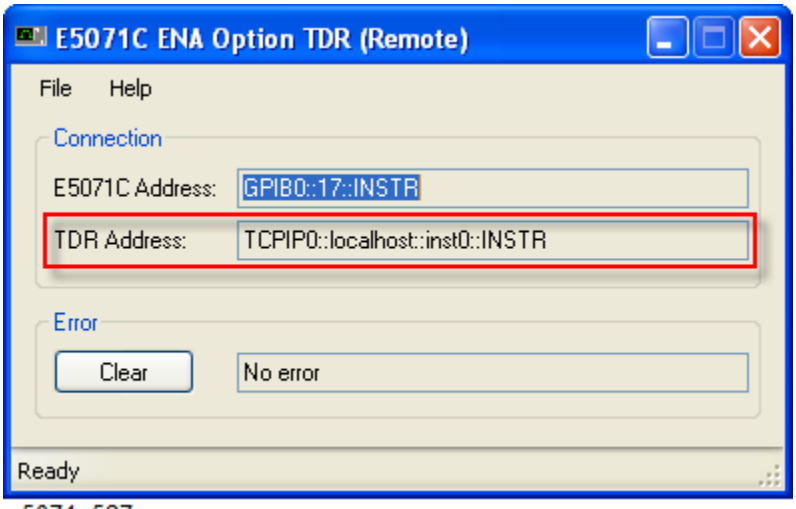
Connecting E5071C-TDR with PC

Below steps are needed to access the instrument remotely.

1. Connect the E5071C with the PC through either GPIB or USB.
2. From Windows Start, **Start** > **All Programs** > **Agilent E5071C ENA Option TDR** > **TDR Remote**.
3. Type the VISA address of E5071C and click **OK**. (This dialog box is displayed once. The next time TDR Remote is launched, the previous VISA address is used to connect to the instrument.)

e5071c526

4. E5071C ENA Option TDR (Remote) window appears. Control the E5071C TDR application with the address in TDR address.



e5071c527

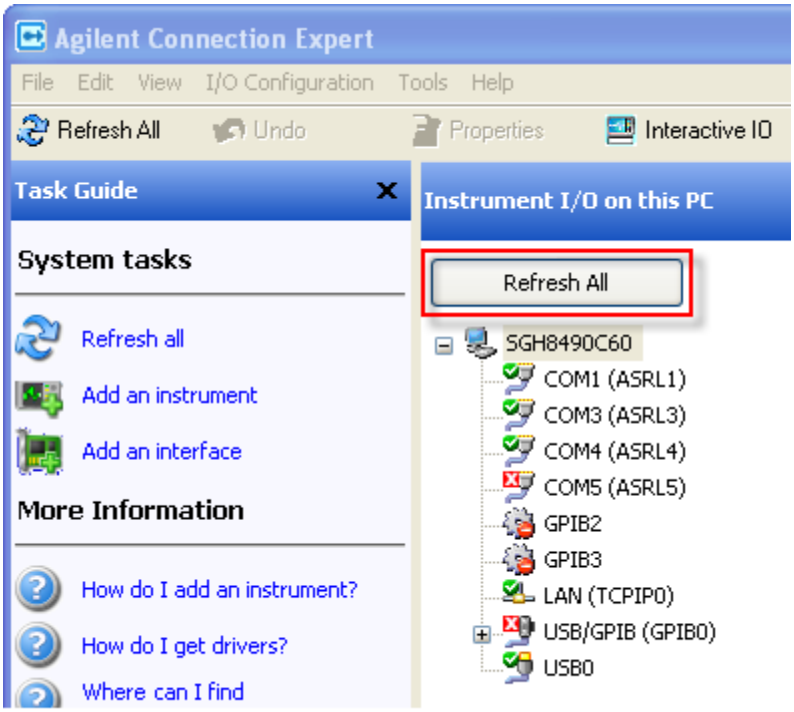
5. E5071C-TDR GUI is closed in the instrument when TDR Remote is launched.

NOTE Restart E5071C-TDR GUI in instrument by clicking **Analysis** > **Start TDR**. (Advanced Mode)

Configuring IO Library Setup

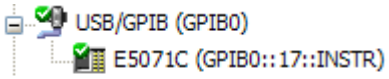
This is optional as it needs Agilent Connection Expert.

1. Click **Start** > **All Programs** > **Agilent IO Libraries Suite** > **Agilent Connection Expert**.
2. Click **Refresh All**.



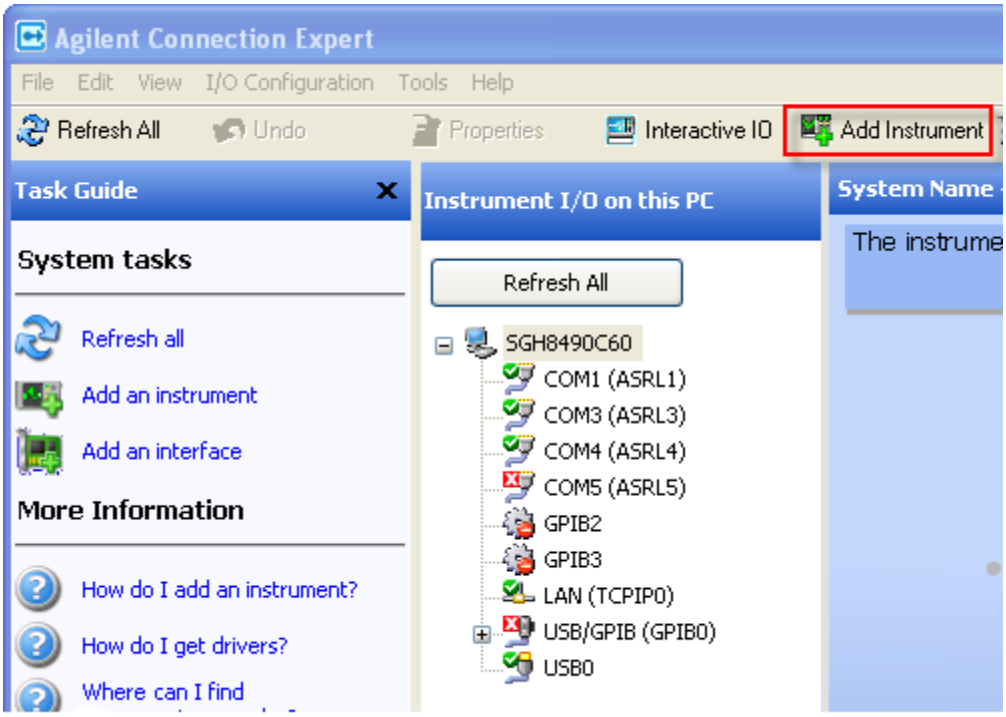
e5071c528

3. Refresh the system. The E5071C is now available under GPIB/USB category.

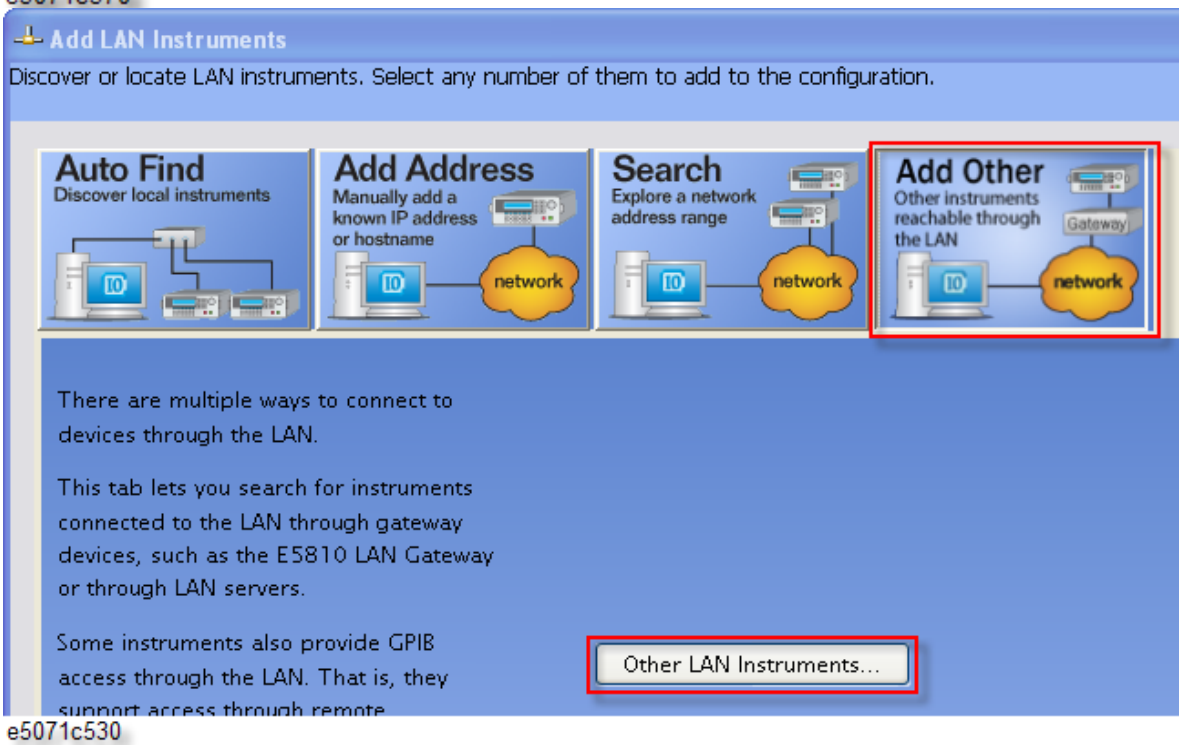


e5071c529

4. Click on LAN (TCPIP0).
5. Select **Add Instrument** > **Add Other** > **Other LAN Instruments...** to connect to E5071C-TDR.

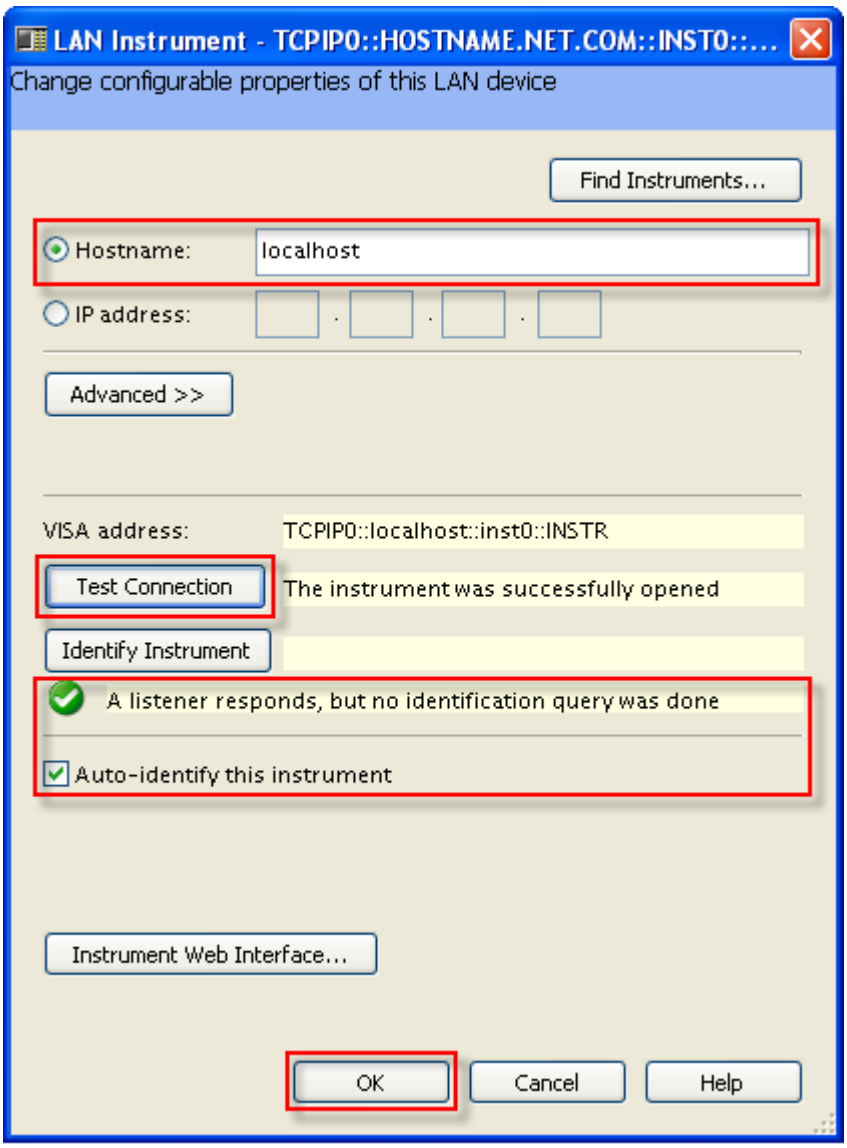


e5071c570



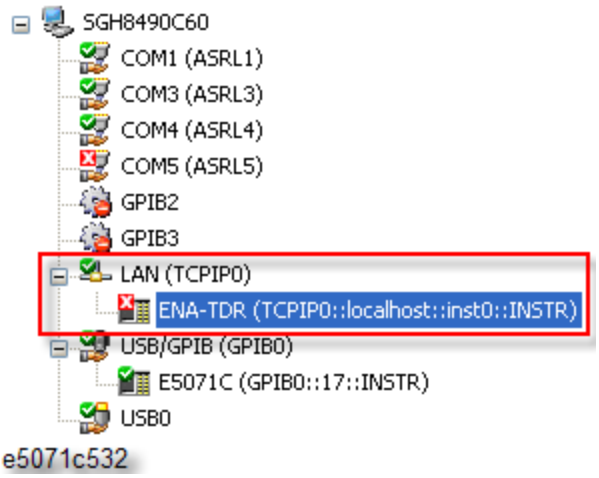
e5071c530

5. Type localhost in Hostname.
6. Click **Test Connection** to confirm that the connection is working.
7. Select out Auto-identify this instrument and click **OK**.



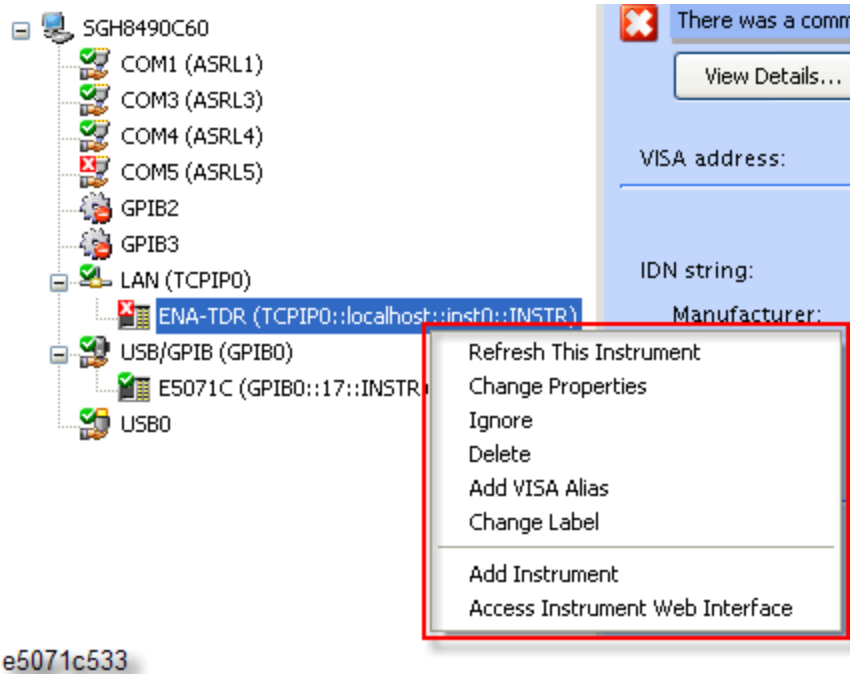
e5071c531

8. Notice that ENA-TDR connection fails in Agilent Connection Expert main window.



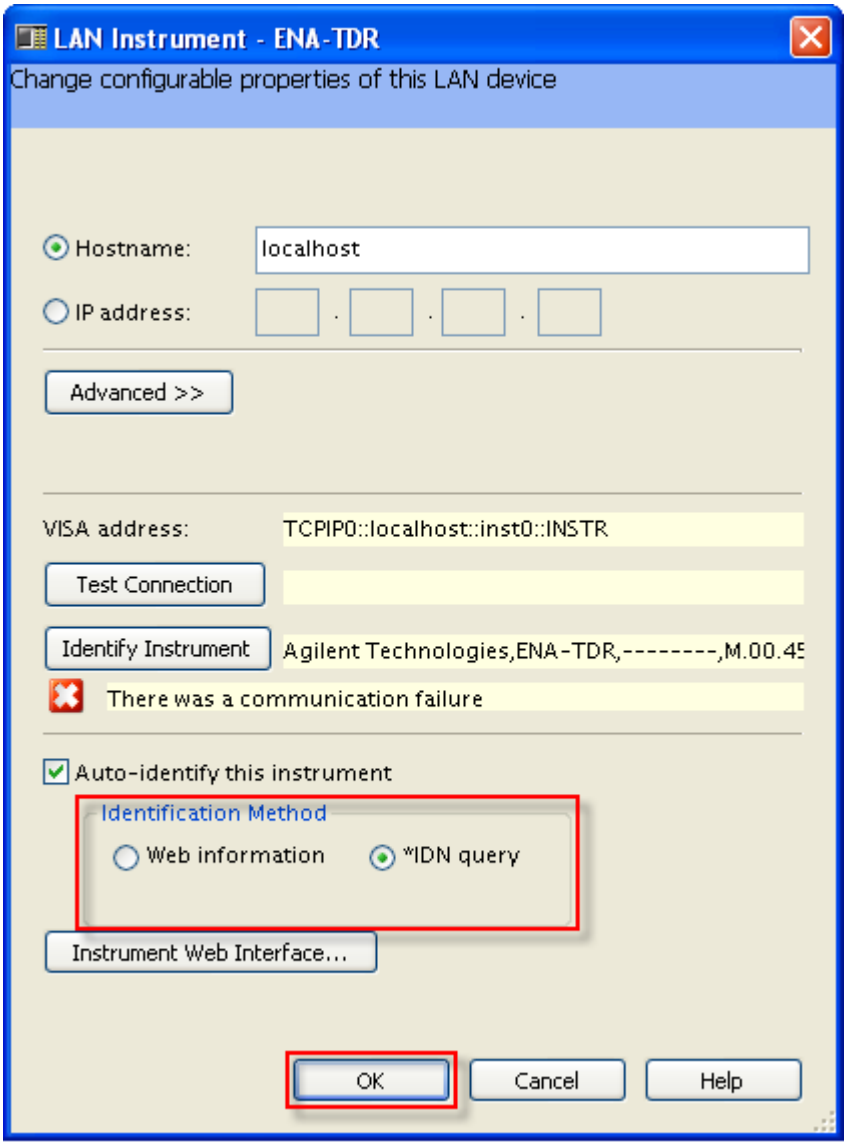
e5071c532

9. Right-click on ENA-TDR option and select **Change Properties.**



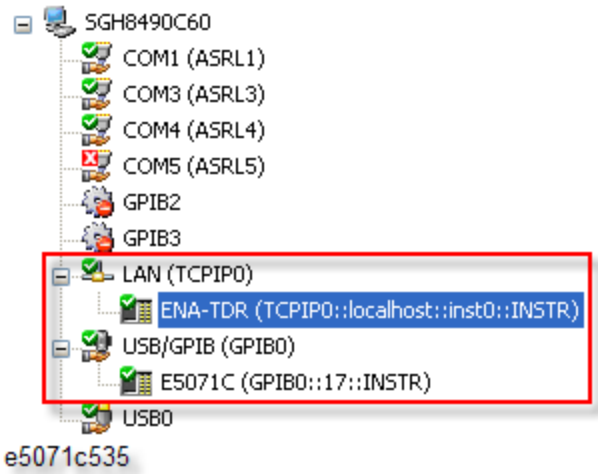
e5071c533

10. Select ***IDN query** under **Identification Method** at LAN Instrument window and click **OK.**



e5071c534

11. Both E5071C and ENA-TDR should be active in Agilent Connection Expert main window as shown below.


SGH8490C60
COM1 (ASRL1)
COM3 (ASRL3)
COM4 (ASRL4)
COM5 (ASRL5)
GPIB2
GPIB3
LAN (TCPIP0)
ENA-TDR (TCPIP0::localhost::inst0::INSTR)
USB/GPIB (GPIB0)
E5071C (GPIB0::17::INSTR)
USB0
e5071c535

TDR Sample Programs

TDR Sample Programs

- TDR/TDT Measurement
- Simulated Eye Diagram
- 2 Channels Measurement

TDR/TDT Measurement

Overview

This sample program demonstrates how to perform TDR/TDT setup.

Sample Program in Excel VBA (VISA-COM)

Global variables

```
Dim rm As VisaComLib.ResourceManager
Dim ENATDR As VisaComLib.FormattedIO488
Dim NumDmy As Integer
```

TDR-TDT Measurement

```
Sub TDRTDTMeasure()
    On Error GoTo errorhandler
    Set rm = New VisaComLib.ResourceManager
    Set ENATDR = New VisaComLib.FormattedIO488

    ' TDR address is not necessary to change.
    Set ENATDR.IO = rm.Open("TCPIP0::localhost::inst0::INSTR")
    ENATDR.IO.Timeout = 50000

    ' Clear Excel Sheet Cells
    Range("F5:F5").ClearContents

    With ENATDR
        'Preset the ENA
        .WriteString ":SYST:PRES"
        'TDR Trigger Hold
        .WriteString ":TRIG:MODE HOLD"

        ' Set DUT Topology to Differential 2-Port
        .WriteString ":CALC:DEV DIF2"
        .WriteString "*OPC?"
         NumDmy = .ReadNumber
        ' Execute Deskew (Deskew & Loss)
        MsgBox "[Deskew] Disconnect the cables from DUT to make open condition.", vbOKOnly
        .WriteString ":SENS:CORR:EXT:AUTO:STAN OPEN"
        .WriteString ":SENS:CORR:EXT:AUTO:PORT1 ON"
```

```
.WriteString ":SENS:CORR:EXT:AUTO:PORT2 ON"
.WriteString ":SENS:CORR:EXT:AUTO:PORT3 ON"
.WriteString ":SENS:CORR:EXT:AUTO:PORT4 ON"
.WriteString ":SENS:CORR:EXT:AUTO:IMM"
.WriteString "*OPC?"
 NumDmy = .ReadNumber
' Execute Thru Measurement: Connect a Thru between Port 1 and Port 3
MsgBox "Connect a Thru between Port 1 and Port 3.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:THRU TH13"
.WriteString "*OPC?"
NumDmy = .ReadNumber
' Execute Thru Measurement: Connect a Thru between Port 2 and Port 4
MsgBox "Connect a Thru between Port 2 and Port 4.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:THRU TH24"
.WriteString "*OPC?"
NumDmy = .ReadNumber

' Measure Load: Connect a Load to Port 1
MsgBox "Connect a Load to Port 1.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 1"
.WriteString "*OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 2
MsgBox "Connect a Load to Port 2.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 2"
.WriteString "*OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 3
MsgBox "Connect a Load to Port 3.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 3"
.WriteString "*OPC?"
NumDmy = .ReadNumber
' Measure Load: Connect a Load to Port 4
MsgBox "Connect a Load to Port 4.", vbOKOnly
.WriteString ":SENS:CORR:COLL:DLC:LOAD 4"
 ' Save the result of loss compensation
.WriteString ":SENS:CORR:COLL:DLC:SAVE"
```

```
    .WriteString "*OPC?"
    NumDmy = .ReadNumber

    'Measure DUT Length
    MsgBox "Connect the DUT to the cables.", vbOKOnly
    .WriteString ":SENS:DLEN:AUTO:IMM"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
    'Set Rise Time
    .WriteString ":CALC:TRAC1:TIME:STEP:RTIM:DATA 35E-12"
    .WriteString ":CALC:TRAC3:TIME:STEP:RTIM:DATA 35E-12"
    .WriteString ":CALC:TRAC5:TIME:STEP:RTIM:DATA 35E-12"
    .WriteString ":CALC:TRAC7:TIME:STEP:RTIM:DATA 35E-12"
    'Set Definition
    .WriteString ":CALC:TRAC1:TIME:STEP:RTIM:THR T1_9"
    .WriteString ":CALC:TRAC3:TIME:STEP:RTIM:THR T1_9"
    .WriteString ":CALC:TRAC5:TIME:STEP:RTIM:THR T1_9"
    .WriteString ":CALC:TRAC7:TIME:STEP:RTIM:THR T1_9"
    .WriteString ":TRIG:SING"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber

    'Set Auto Scale for all traces
    .WriteString ":DISP:ATR:SCAL:AUTO"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber

     'Read Rise Time
    .WriteString ":CALC:TRAC3:TTIM:STAT ON"
    .WriteString ":CALC:TRAC3:TTIM:THR T1_9"
    .WriteString ":CALC:TRAC3:TTIM:DATA?"
    Cells(5, 6) = .ReadNumber

    MsgBox "TDT/TDT Measurement Setup is completed.", vbOKOnly
End With
ENATDR.IO.Close
Exit Sub
```

```
errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext

End Sub
```

Simulated Eye Diagram

Overview

This sample program demonstrates how to perform simulated eye diagram.

Sample Program in Excel VBA (VISA-COM)

Global variables

```
Dim rm As VisaComLib.ResourceManager
Dim ENATDR As VisaComLib.FormattedIO488
Dim NumDmy As Integer
```

Open

```
Sub OpenInstrument()
    On Error GoTo errorhandler
    Set rm = New VisaComLib.ResourceManager
    Set ENATDR = New VisaComLib.FormattedIO488

    ' TDR address is not necessary to change.
    Set ENATDR.IO = rm.Open("TCPIP0::localhost::inst0::INSTR")
    ENATDR.IO.Timeout = 50000

    ' Clear Excel Sheet Cells
    Range("F5:F22").ClearContents
    MsgBox "E5071C-TDR is opened.", vbOKOnly

    Exit Sub
errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Setup

```
Sub setup()
    On Error GoTo errorhandler
    With ENATDR
    'Preset the ENA
    .WriteString ":SYST:PRES"
    .WriteString "*OPC?"
```

```
    NumDmy = .ReadNumber

    'TDR Trigger Hold
    .WriteString ":TRIG:MODE HOLD"
    ' Set DUT Topology to Differential 2-Port
    .WriteString ":CALC:DEV DIF2"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
    ' Execute Deskew (Deskew)
    MsgBox "[Deskew] Disconnect the cables from DUT to make open condition.", vbOKOnly
    .WriteString ":SENS:CORR:EXT:AUTO:STAN OPEN"
    .WriteString ":SENS:CORR:EXT:AUTO:PORT1 ON"
    .WriteString ":SENS:CORR:EXT:AUTO:PORT2 ON"
    .WriteString ":SENS:CORR:EXT:AUTO:PORT3 ON"
    .WriteString ":SENS:CORR:EXT:AUTO:PORT4 ON"
    .WriteString ":SENS:CORR:EXT:AUTO:IMM"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
    'Measure DUT Length
    MsgBox "Connect the DUT to the cables.", vbOKOnly
    .WriteString ":SENS:DLEN:AUTO:IMM"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber

    'Set Rise Time
    .WriteString ":CALC:TRAC1:TIME:STEP:RTIM:DATA 35E-12"
    'Set Definition
    .WriteString ":CALC:TRAC1:TIME:STEP:RTIM:THR T1_9"
    'Activate Trace 3
    .WriteString ":CALC:ATR:ACT 3"
    .WriteString ":TRIG:SING"
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
    MsgBox "Setup is completed.", vbOKOnly

    End With
  Exit Sub
```

```
errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Simulated Eye Diagram

```
Sub SimEyeDiagram()
Dim EyeResult() As Double
Dim NoR As Integer, i As Integer
On Error GoTo errorhandler
    With ENATDR
        ' Set Bit Pattern Parameters
        ' Set Type to PRBS
        .WriteString ":CALC:EYE:INP:BPAT:TYPE PRBS"
        ' Set Length
        .WriteString ":CALC:EYE:INP:BPAT:LENG 7"
        ' Set Amplitude
        .WriteString ":CALC:EYE:INP:OLEV 200e-3"
        ' Set Data Rate
        .WriteString ":CALC:EYE:INP:RTIM:THR T1_9"
        .WriteString "*OPC?"
         NumDmy = .ReadNumber

       ' Execute Draw Eye
        .WriteString ":CALC:EYE:STAT ON"
        .WriteString ":CALC:EYE:EXEC"
        .WriteString "*OPC?"
        NumDmy = .ReadNumber

        'Read Eye Result
        .WriteString ":CALC:EYE:RES:DATA?"
        EyeResult() = .ReadList(ASCIIType_R8, ",")

        NoR = 18
        i = 0
        Do While i < NoR
        Cells(i + 5, 6) = EyeResult(i)
```

```
        i = i + 1
      Loop
      .WriteString "*OPC?"
      NumDmy = .ReadNumber
      MsgBox "Eye Diagram Simulation is completed.", vbOKOnly
  End With
 Exit Sub

errorhandler:
   MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
 End Sub
```

Close

```
Sub CloseInstrument()
     ENATDR.IO.Close
End Sub
```

2 Channels Measurement

- Overview
- Sample Program in Excel VBA (VISA-COM)

Other topics about TDR Sample Program

Overview

2 channel measurement allows you to makes TDR measurement on channel 1 and more customized S-parameter measurement on channel2. This is a program example of the 2 Channel Measurement Example. In 2 channel measurement, you need to setup two logical instrument, E5071C and E5071C-TDR. In this sample program, the E5071C is assigned as ENA and E5071C-TDR is assigned as ENATDR.

The TDR commands should be sent to ENATDR and The E5071C standard commands should be sent to ENA.

Note for 2 Channels Measurement

- Put *OPC? to make sure to finish the process before you change the control from ENA to ENATDR or ENATDR to ENA.
- DUT topology setting should be executed at the beginning of sequence, because changing DUT Topology presets the E5071C.

Sample Program in Excel VBA (VISA-COM)

Grobal valiables

```
Dim rm As VisaComLib.ResourceManager
Dim ENA As VisaComLib.FormattedIO488
Dim ENATDR As VisaComLib.FormattedIO488
Dim NumDmy As Integer
```

Open Instruments

```
Sub OpenInstrument()
    On Error GoTo errorhandler
    Set rm = New VisaComLib.ResourceManager
    Set ENA = New VisaComLib.FormattedIO488
    Set ENATDR = New VisaComLib.FormattedIO488

    ' Change the address for your configuration
    Set ENA.IO = rm.Open("GPIB0::17::INSTR")
    ' Set time out time for ENA at 30 second
    ENA.IO.Timeout = 30000
```

```
    ' TDR address is not necessary to change.
    Set ENATDR.IO = rm.Open("TCPIP0::localhost::inst0::INSTR")
   ' Set time out time for ENATDR at 70 second
    ENATDR.IO.Timeout = 70000

    ' Clear Excel Sheet Cells
    Range("D7:H10010").ClearContents
    Range("E4:E4").ClearContents
    Range("H4:H4").ClearContents

    MsgBox "E5071C and E5071C-TDR are opened.", vbOKOnly
    Exit Sub
errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Preparation for 2 Channels

```
Sub Preparation()
    On Error GoTo errorhandler
    ' Note: Must put *OPC? when you change the control between ENA and ENATDR.
    With ENA
        ' ENA- Preset ENA
        .WriteString "*RST"
        ' *OPC? for checking the process finish
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
    End With

    With ENATDR
        ' TDR- Set Diff 2 port as DUT
        .WriteString ":CALC:DEV DIF2"
        ' TDR- Wait Until Topology Setting is finished.
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
        '
        ' TDR- Trigger Hold
        .WriteString ":TRIG:MODE HOLD"
```

```
    '
    ' *OPC? for checking the process finish
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
  End With

  ' ENA- Setup ENA
  With ENA
    ' Allocate channel must be done after TDR DUT Type setting.
    .WriteString ":DISP:SPL D12"
    '
    ' ENA- Set Trigger at BUS
    .WriteString ":TRIG:SOUR BUS"

    ' ENA- Set Trigger scope at Active channel only
    .WriteString ":TRIG:SCOP ACT"
    '
    ' ENA- Beep Warning OFF
    .WriteString ":SYST:BEEP:WARN:STAT OFF"
    '
    ' *OPC? for checking the process finish
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
  End With

  MsgBox "Preparation for 2ch is done.", vbOKOnly
  Exit Sub

errorhandler:
  MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext

End Sub
```

Setup for Channel 1

```
Sub SetupCh1()
  On Error GoTo errorhandler
```

```
    With ENATDR
        '
        ' TDR- Execute Deskew
        FrmDeSkew.Show  ' Call DeSkew subroutine in Form
        '
        ' TDR- Execute Auto DUT Length
        FrmDUTLength.Show  ' Call DUT Length subroutine in From
        '
        ' TDR- Set rise time for Traces 1 and 2
        .WriteString ":CALC:TRAC1:TIME:STEP:RTIM:THR T2_8"
        .WriteString ":CALC:TRAC1:TIME:STEP:RTIM:DATA 50e-12"
        .WriteString ":CALC:TRAC2:TIME:STEP:RTIM:THR T2_8"
        .WriteString ":CALC:TRAC2:TIME:STEP:RTIM:DATA 50e-12"
        '
        ' *OPC? for checking the process finish
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
    End With
    '
    ' Advance mode setting (such as Limit Test) for channel 1 (TDR) should be setup through ENA
    '
    With ENA
        ' ENA- Limit Test ON
        .WriteString ":CALC1:LIM ON"
        ' ENA- Limit Line On
        .WriteString ":CALC1:LIM:DISP ON"
        ' ENA- Edit Limit Line
        .WriteString ":CALC1:LIM:DATA 2,1,0,1e-9,105,105,2,0,1e-9,75,75"
        '
        ' *OPC? for checking the process finish
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
    End With


    MsgBox "Setup for Ch1 is done.", vbOKOnly
    Exit Sub
```

```
errorhandler:
   MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Deskew (This is called from SetupCh1)

```
Sub DeSkew()
   With ENATDR
      ' TDR- Execute Descrew
      .WriteString ":SENS:CORR:EXT:AUTO:IMM"
      ' *OPC? for checking the process finish
      .WriteString "*OPC?"
      NumDmy = .ReadNumber
   End With
End Sub
```

DutLength (This is called from SetupCh1)

```
Sub DUTLength()
   With ENATDR
      ' TDR- Execute Auto DUT length
      .WriteString ":SENS:DLEN:AUTO:IMM"
      ' *OPC? for checking the process finish
      .WriteString "*OPC?"
      NumDmy = .ReadNumber
   End With
End Sub
```

Measurement for Channel 1

```
Sub MeasCh1()
     On Error GoTo errorhandler
   '
   Dim TimeData() As Double, Impedance() As Double
   Dim Nop As Integer, PassFail As Integer, i As Integer, k As Integer
   '
   ' Making Measurement for TDR (Channel 1)

   With ENA
```

```
    ' ENA- Set active channel at channel 1
    .WriteString ":DISP:WIND1:ACT"
    ' *OPC? for checking the process finish
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
End With
'
With ENATDR
    ' TDR- Single Trigger
    .WriteString ":TRIG:SING"
    ' TDR- Wait for measurement end
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
    ' TDR- Autoscale
    .WriteString ":DISP:ATR:SCAL:AUTO"
    ' *OPC? for checking the process finish
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
End With
'
' Read TDR data (Channel1) - TDR Data should be read through ENA.
'
With ENA
    ' Query Number of Test Points
    .WriteString ":SENS1:SWE:POIN?"
    Nop = .ReadNumber
    ReDim TimeData(Nop - 1)
    ReDim Impedance(Nop * 2 - 1)
    ' Get the data for X axis.
    .WriteString ":CALC1:SEL:DATA:XAX?"
    TimeData() = .ReadList(ASCIIType_R8, ",")
    ' Select the trace 1.
    .WriteString ":CALC1:PAR1:SEL"
    ' Get the data for Y axis for active trace.
    .WriteString ":CALC1:DATA:FDAT?"
    Impedance() = .ReadList(ASCIIType_R8, ",")
    ' Get limit line test result
```

```
        .WriteString ":CALC1:LIM:FAIL?"
        PassFail = .ReadNumber

        k = 0
        For i = 0 To Nop - 1
            Cells(i + 7, 4) = TimeData(i)
            Cells(i + 7, 5) = Impedance(k)
            k = k + 2
        Next i
        Cells(4, 5) = PassFail
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
    End With

    'MsgBox "Measurement for Ch1 is done.", vbOKOnly
    Exit Sub

errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext

End Sub
```

Setup for Ch2

```
Sub SetupCh2()
    On Error GoTo errorhandler
    With ENA
        '
        ' Setup Network measurement(S-Parameter) on channel 2.
        '
        ' ENA- Set Start Frequency
        .WriteString ":SENS2:FREQ:STAR 1E9"
        ' ENA- Set Stop Frequency
        .WriteString ":SENS2:FREQ:STOP 3E9"
        ' ENA- Set IFBW
        .WriteString ":SENS2:BAND 1E3"
        ' ENA- Fixture Simulator ON
```

```
        .WriteString ":CALC2:FSIM:STAT ON"
        ' ENA- Set topology at Balance-Balance
        .WriteString ":CALC2:FSIM:BAL:DEV BBAL"
        ' ENA- Assign Phisical Ports
        .WriteString ":CALC2:FSIM:BAL:TOP:BBAL 1,2,3,4"
        ' ENA- Balance-Balance Conversion ON
        .WriteString ":CALC2:FSIM:BAL:PAR1:STAT ON"
        ' ENA- Select Measurement Parameter at Sdd21
        .WriteString ":CALC2:FSIM:BAL:PAR1:BBAL SDD21"
        '
        ' ENA- Limit Test ON
        .WriteString ":CALC2:LIM ON"
        ' ENA- Limit Line On
        .WriteString ":CALC2:LIM:DISP ON"
        ' ENA- Edit Limit Table
        .WriteString ":CALC2:LIM:DATA 3,2,100e6,1.25e9,-1.5,-5,2,1.25e9,2.5e9,-5,-7.5,2,2.5e9,7.5e9,-
7.5,-25"
        '
        ' *OPC? for checking the process finish
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
    End With

    FrmEcal.Show

    MsgBox "Setup Ch2 is done.", vbOKOnly
    Exit Sub

errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

EcalCalibration (This is called from SetupCh2)

```
Sub EcalCalibration()
    With ENA
        ' ENA- 4 Port Ecal
```

```
    .WriteString ":SENS2:CORR:COLL:ECAL:SOLT4 1,2,3,4"
    ' *OPC? for checking the process finish
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
  End With
End Sub
```

Measurement for channel 2

```
Sub MeasCh2()
  On Error GoTo errorhandler
  Dim FreqData() As Double, InsersionLoss() As Double
  Dim Nop As Integer, PassFail As Integer, i As Integer, k As Integer
  '
  With ENA
    ' ENA- Set active channel at channel 2
    .WriteString ":DISP:WIND2:ACT"
    ' ENA- Single Trigger
    .WriteString ":INIT2;:TRIG:SING"
    ' ENA- Wait for measurement end
    .WriteString "*OPC?"
    NumDmy = .ReadNumber
  End With
  '
  ' Read Newtwork Meaurement (Channel 2) data
  '
  With ENA
    ' ENA- Get Number of Points of data
    .WriteString ":SENS2:SWE:POIN?"
    Nop = .ReadNumber
    ' Get the data for x-axis (Frequency)
    ReDim FreqData(Nop - 1)
    .WriteString ":SENS2:FREQ:DATA?"
    FreqData() = .ReadList(ASCIIType_R8, ",")
    ' Select the trace 1.
    .WriteString ":CALC2:PAR1:SEL"
    ' Get the data for Y-axis (Sdd21 - LogMag)
    ReDim InsersionLoss(Nop * 2 - 1)
```

```
        .WriteString ":CALC2:DATA:FDATA?"
        InsersionLoss() = .ReadList(ASCIIType_R8, ",")
        ' Get limit line test result
        .WriteString ":CALC2:LIM:FAIL?"
        PassFail = .ReadNumber
        '
        k = 0
        For i = 0 To Nop - 1
            Cells(i + 7, 7) = FreqData(i)
            Cells(i + 7, 8) = InsersionLoss(k)
            k = k + 2
        Next i
        Cells(4, 8) = PassFail
        ' *OPC? for checking the process finish
        .WriteString "*OPC?"
        NumDmy = .ReadNumber
    End With

    Exit Sub

errorhandler:
    MsgBox Err.Description, vbExclamation, "Error Occurred", Err.HelpFile, Err.HelpContext
End Sub
```

Close Instrument

```
Sub CloseInstrument()

    ENA.IO.Close
    ENATDR.IO.Close
    MsgBox "E5071C and E5071C-TDR are closed.", vbOKOnly

End Sub
```

Command Reference

Command Reference

- Notational Conventions
- COM Object Model
- Command Finder

Other topics about Programming

Notational Conventions

This section describes the notational conventions used for the description of commands reference.

Object Type

Object type describes different types of E5071C COM objects. The E5071C provides properties and methods as COM objects. COM objects which set (send)/read (return) the state of the E5071C using variables are defined as property and COM objects which does other processing are defined as method.

COM objects used only to read the state of the E5071C are indicated with "**Read-only**" and ones used only to set the state of the E5071C are indicated by "**Write-only**" . COM object that can both read and write data to the E5071C are indicated by '**Read-Write**'.

Syntax

Syntax describes the syntax for sending a COM object from the E5071C VBA to the E5071C. The syntax consists of two parts: the object part and the set/read part, with an equal "=" inserted between them. Variables are indicated by italicized letters. Variables with () are indices. For indices with () having their preset values, you can omit "(*variable*)," and, if omitted, the preset values are automatically set.

The following table describes the 3 types of syntax for coding using objects:

Type	Description
"Object (property) = *variable*":	Set the stat of the E5071C.
variable=object (property):	Read the stat of the E5071C.
"Object (method)":	Perform some processing in the E5071C.

Description

Description describes how to use the COM object or the operation when executed.

Variable

Variable provides description about different variables that can be used with the COM objects. It gives the description, data type, allowable range, preset value, unit, resolution, and notes for *variable (italic)* shown in the syntax.

- Variables declared as the string data type (String) are not case-sensitive. For variables of the string type that indicate arguments (written as *Param* in the syntax), you can omit lower-case letters.

The data types of the E5071C COM objects include 5 types as shown in the following table. Before using variables, declare the data type of each variable. If you do not declare the data type of a variable, it is automatically processed as a variant type.

Data type	Name	Consumed memory	Range
Long	Long integer type	4 bytes	-2,147,483,648 to 2,147,483,647
Double	Double precision floating point type	8 bytes	For a negative value: -1.79769313486232E+308 to -4.94065645841247E-324 For a positive value: -1.79769313486232E+308 to -4.94065645841247E-324
Boolean	Boolean type	2 bytes	For COM: True or False (For SCPI: ON or OFF)
String	Character string type	1 byte / alphanumeric character	Up to approximately 2 billion characters
Variant	Variant type	16 bytes	No limitation

Examples

Examples provides a sample example of using the object through coding with the E5071C VBA.

Related Objects

Related objects provides information about other objects that are similar/related with the object.

Equivalent Key

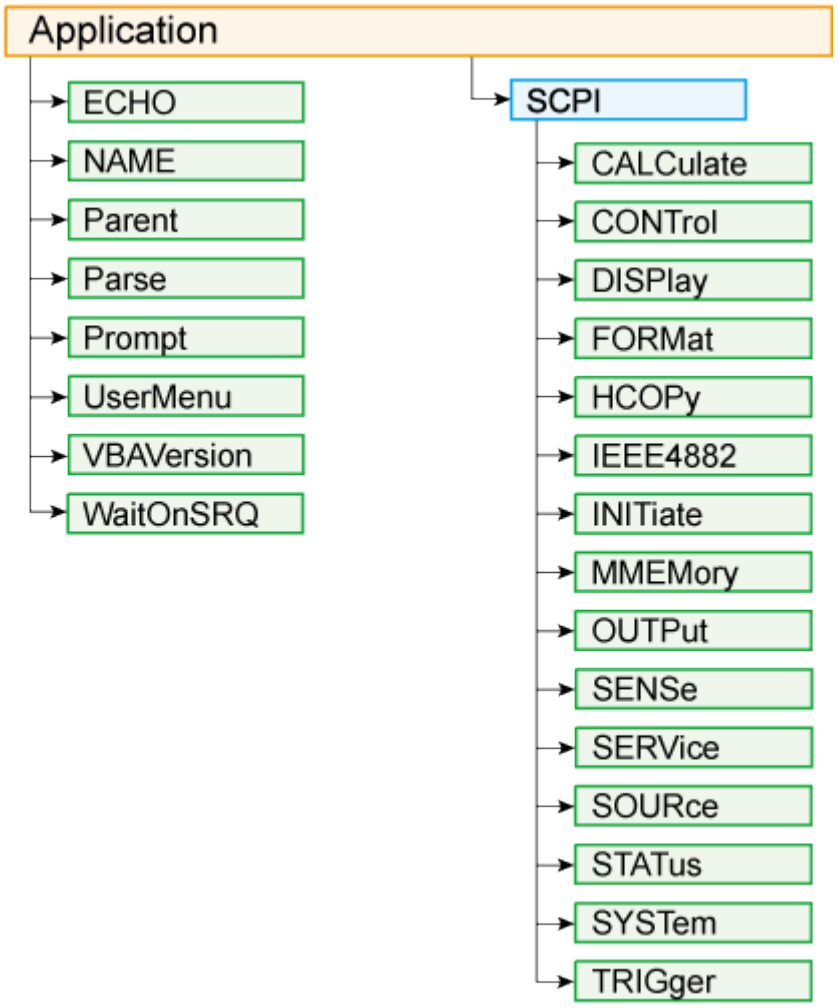
Equivalent key shows the operational procedure of the front panel keys that has the same effect as this object.

Equivalent SCPI command

Equivalent SCPI command shows the SCPI command to execute from an external controller. Its syntax, query response, and example of use are provided.

COM Object Model

The COM objects provided for the E5071C are structured hierarchically as shown below.



e5071c473

Application Objects

The Application objects are at the top of the hierarchy of the E5071C COM object model. They consist of 7 objects dedicated to the COM interface and SCPI objects corresponding to SCPI commands.

SCPI Objects

The SCPI objects are created to realize the SCPI commands of the E5071C with the COM interface.

The conversion rules from the SCPI commands when writing SCPI object messages are as follows:

- SCPI. must be at the beginning. Notice that the IEEE common commands start with SCPI.IEEE4882. and "*" is omitted.
- Replace colons (:) used as the hierarchical separator symbol with dots (.).
- The number written in the object message is specified with ( ).
- You cannot omit the command message in the syntax.

SCPI command	COM object
OUTPUT 717;":SOUR1:POW -10"	SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude = -10
OUTPUT 717;":SENS1:CORR:COLL:METH:TYPE?" ENTER 717;A$	A = SCPI.SENSe(1).CORRection.COLLect.METHod:TYPE
OUTPUT 717;"*CLS"	SCPI.IEEE4882.CLS

Application Objects

ECHO

Object type

Method (**Write-only**)

Syntax

ECHO *V1,V2, ... ,V10*

ECHO *SCPI object*

Description

Provides display in the echo window. (No read)

This command is difference from SCPI.DISPlay.ECHO.DATA.

- Up to 10 data items can be displayed.
- Data is displayed as the declared data type without a cast.

Parameter	*V1,V2,* **.....** *,V10*
Description	Data you want to display in the echo window.
Data type	Variant type (Variant)

Examples

```
Dim Nop As Long
Dim i As Integer
Dim Fdata As Variant
Nop = SCPI.SENSe(1).SWEep.POINts
Fdata = SCPI.CALCulate(1).SELected.DATA.FDATa
ECHO "Test Results"
For i=1 to Nop
    ECHO i, Fdata(2*i-2), Fdata(2*i-1)
Next i


ECHO SCPI.SYSTem.ERRor
```

Related objects

SCPI.DISPlay.ECHO.DATA

Equivalent key

No equivalent key is available on the front panel.

NAME

Object type

Property

Syntax

App = NAME

Description

Reads out the application name of VBA. E5071C is always read out. (Read only)

Variable

Parameter	*App*
Description	Application name
Data type	Character string type (String)

Examples

```
Dim Inst As String
Inst = NAME
ECHO Inst
```

Equivalent key

No equivalent key is available on the front panel.

Parse

Object type

Method (**Write-only**)

Syntax

Parse(*Scpi*)

Return = Parse(*Scpi?*)

Description

Executes an SCPI command of the E5071C.

The Parse object is a little slower in the execution speed than the COM object which has the same function as the SCPI command because it must parse the message string of the SCPI command.

Variable

Parameter	*Scpi*
Description	SCPI command
Data type	Character string type (String)

Parameter	*Return*
Description	Response (query) of the SCPI command
Data type	Character string type (String)

Examples

```
Dim Start As String
Parse(":SENS1:FREQ:STAR 100E6")
Start = Parse(":SENS1:FREQ:STAR?")
```

```
Dim TtlLbl As String
Parse(":DISP:WIND1:TITL:DATA ""filter""")
TtlLbl = Parse(":DISP:WIND1:TITL:DATA?")
```

```
Dim Fmt As String
Parse(":CALC1:PAR2:SEL")
```

```
Parse(":CALC1:FORM SMIT")
Fmt = Parse(":CALC1:FORM?")
```

```
Dim BckLght As String
Parse(":SYST:BACK OFF")
BckLght = Parse(":SYST:BACK?")
```

Equivalent key

No equivalent key is available on the front panel.

Prompt

Object type

Method (**Write-only**)

Syntax

Prompt(*Mes*)

Description

Displays the message you specify on the instrument status bar (at the bottom of the LCD display) and suspends the program until the **Macro Setup** > **Continue** button is pressed. (No read)

- When using this object, execute the program with the Visual Basic closed since you need to press the **Macro Setup** > **Continue**. For more information, see Running a Program from the E5071C Measurement Screen. If you need to abort the program, see Stopping with the Dialog Box Appeared.

Variable

Parameter	*Mes*
Description	Message
Data type	Character string type (String)

Examples

Prompt ("Connect DUT, and then press [Continue]")

Equivalent key

No equivalent key is available on the front panel.

UserMenu.Item(*Key_id*).Caption

Object type

Property

Syntax

UserMenu.Item(*Key_id*).Caption = *Lbl*

Lbl = UserMenu.Item(*Key_id*).Caption

Description

Sets the label name of the user menu function softkeys 1 to 10 (*Key_id*).

Variable

Parameter	*Key_id*
Description	Softkey number for the user menu function
Data type	Long integer type (Long)
Range	1 to 10
Note	You cannot omit this because it does not have a preset value. If the specified variable is out of the valid setting range, an error occurs when executed.

Parameter	*Lbl*
Description	Softkey label name for the user menu function
Data type	Character string type (String)
Preset value	Varies depending on the specified softkey number.

Examples

```
Dim KeyLbl As String
UserMenu.Item(1).Caption = "Meas"
KeyLbl = UserMenu.Item(1).Caption
```

Equivalent key

No equivalent key is available on the front panel.

UserMenu.Item(*Key_id*).Enabled

Object type

Property

Syntax

UserMenu.Item(*Key_id*).Enabled = *Status*

Status = UserMenu.Item(*Key_id*).Enabled

Description

Makes the user menu function softkeys 1 to 10 (*Key_id*) enabled/disabled. The softkey label disabled is displayed with the grey color and its softkey cannot be pressed.

Variable

Parameter	*Status*
Description	Enabled/disabled for the user menu function softkey
Data type	Boolean type (Boolean)
Range	Select from the following. • True or -1 Makes the softkey enabled. • False or 0 Makes the softkey enabled.
Preset value	True or -1

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
Dim KeyEna As Boolean
UserMenu.Item(10).Enabled = False
KeyEna = UserMenu.Item(10).Enabled
```

Related objects

UserMenu.Press

Equivalent key

No equivalent key is available on the front panel.

UserMenu.PRESet

Object type

Method (**Write-only**)

Syntax

UserMenu.PRESet

Description

Presets the label name and enabled/disabled settings for the user menu softkeys. (No read)

Examples

UserMenu.PRESet

Related objects

UserMenu.Item.Caption

UserMenu.Item.Enabled

Equivalent key

Macro Setup > **Preset User Menu**

UserMenu.Press(*Key_id*)

Object type

Method (**Write-only**)

Syntax

UserMenu.Press(*Key_id*)

Description

Presses one of the user menu function softkeys 1 to 10 (*id*). (No read)

Variable

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
UserMenu.Press(1)
```

Related objects

UserMenu.Item.Enabled

Equivalent key

Macro Setup > **User Menu** > **Button 1** to **Button 10**

UserMenu.Show

Object type

Method (**Write-only**)

Syntax

UserMenu.Show

Description

Displays the user menu function softkeys in the softkey area. (No read)

Examples

UserMenu.Show

Equivalent key

Macro Setup > **User Menu**

UserMenu_OnPress(ByVal *Key_id* As Long)

Object type

Event

Description

Executes the processing when one of the user menu function softkeys 1 to 10 (*Key_id*) is pressed. Write the processing in the "UserMenu" object. For more information on its use, see Executing a Procedure with a Softkey (User Menu Function).

Variable

For information on the variable (*Key_id*), see UserMenu.Item.Caption.

Examples

```
Private Sub UserMenu_OnPress(ByVal id As Long)
If id = 1 Then
MsgBox "Button 1 was pressed."
ElseIf id = 10 Then
MsgBox "Button 10 was pressed."
End If
End Sub
```

Equivalent key

No equivalent key is available on the front panel.

VBAVersion

Object type

Property

Syntax

Vers = VBAVersion

Description

Reads out the version information of VBA installed in the E5071C. (Read only)

Variable

Parameter	*Vers*
Description	VBA version information
Data type	Character string type (String)

Examples

```
Dim Version As String
Version = VBAVersion
ECHO Version
```

Equivalent key

From the **Help** menu of the Visual Basic editor, click **About Microsoft Visual Basic...**.

WaitOnSRQ

Object type

Method (**Write-only**)

Syntax

WaitOnSRQ *Status*, *Timeout*

Description

Suspends the program for specified time until the RQS/MSS bit (bit 6) of the status byte register changes to 1. (No read)

Variable

Parameter	*Status*
Description	State of the RQS/MSS bit (read only)
Data type	Boolean type (Boolean)
Range	One of the following is returned. • True or -1 1 has been received within the specified time. • False or 0 1 has not been received within the specified time due to timeout or abort.

Parameter	*Timeout*
Description	Timeout time
Data type	Long integer type (Long)
Range	0 to 2,147,483,647
Preset value	-1 (infinity)
Unit	ms (millisecond)

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Stat As Boolean
SCPI.IEEE4882.CLS
SCPI.STATus.OPERation.PTRansition = 0
SCPI.STATus.OPERation.NTRansition = 16
SCPI.STATus.OPERation.ENABle = 16
SCPI.IEEE4882.SRE = 128
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMediate
WaitOnSRQ Stat, 10000
If Stat = True Then
MsgBox "Done"
End If
```

Equivalent key

No equivalent key is available on the front panel.

ABORT

SCPI.ABORt

Object type

Method (**Write-only**)

Syntax

SCPI.ABORt

Description

This command aborts the measurement and changes the trigger sequence for all channels to idle state.

The channels for which the continuous startup mode is set to *ON* (setting to start up the trigger system continuously), changes immediately from idle to startup state. See Trigger System for details.

Examples

SCPI.ABORt

Related objects

SCPI.INITiate(Ch).IMMediate

SCPI.INITiate(Ch).CONTinuous

Equivalent key

Trigger > **Restart**

Equivalent SCPI command

Syntax

:ABORt

Example of use

```
10 OUTPUT 717;":ABORt"
```

CALCULATE

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).IMAGinary

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).IMAGinary = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).IMAGinary

Description

This command sets/gets the impedance value (imaginary part) for the common port impedance conversion function, for balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

Variable

Parameter	*Ch*
Description	Channel number
Data type	Long integer type (Long)
Range	1 to 160
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Bpt*
Description	Balance port number
Data type	Long integer type (Long)

Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	Impedance value (imaginary part) for the common port impedance conversion function
Data type	Double precision floating point type (Double)
Range	-1E+18 to 1E+18
Preset value	0
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CImag As Double
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORt(1).IMAGinary = 1E5
CImag = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORt(1).IMAGinary
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).REAL

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Cmn ZConversion** > **Port1(bal) Imag**|**Port2(bal) Imag**|**Port3(bal) Imag**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:IMAGinary <numeric>

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:IMAGinary?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:IMAG 30E6"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:IMAG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).REAL

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).REAL = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).REAL

Description

This command sets/gets the impedance value (real part) for the common port impedance conversion function, for balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

- This command performs in the same way as SCPI.CALCulate(Ch).FSIM-ulator.BALun.CZConversion. BPORt(Bpt).Z0.R

Variable

Parameter	*Value*
Description	Impedance value (real part) for the common port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	25
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CReal As Double
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORt(1).REAL = 30
CReal = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORt(1).REAL
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Cmn ZConversion** > **Port1(bal) Real**|**Port2(bal) Real**|**Port3(bal) Real**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:REAL <numeric>

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:REAL?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:REAL 30E6"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:REAL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).Z0.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).Z0.R = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.BPORt(*Bpt*).Z0.R

Description

This command sets/gets the impedance value for the common port impedance conversion function, for balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

- This command clears setting value of SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion. BPORt(Bpt).IMAGinary

Variable

Parameter	*Value*
Description	Impedance value for the common port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	25
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CZ0 As Double
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORt(1).Z0.R = 30
CZ0 = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.BPORt(1).Z0.R
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).REAL

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Cmn ZConversion** > **Port1(bal) Real|Port2(bal) Real|Port3(bal) Real**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:Z0[:R] <numeric>

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:BPORt{[1]|2}:Z0[:R]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:Z0 30"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:BPOR1:Z0?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.BALun.CZConversion.STATe

Description

This command turns ON/OFF the common port impedance conversion function when the fixture simulator function is *ON*, for all the balance ports of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the common port impedance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the common port impedance conversion function. • False or OFF: Turns OFF the common port impedance conversion function.
Preset value	False or OFF

Examples

```
Dim ComZcon As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.STATe = True
ComZcon = SCPI.CALCulate(1).FSIMulator.BALun.CZConversion.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).REAL

SCPI.CALCulate(Ch).FSIMulator.BALun.CZConversion.BPORt(Bpt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Cmn ZConversion > Cmn ZConversion

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:BALun:CZConversion:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:CZC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:CZC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVice

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVice = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DEVice

Description

This command selects the balance device type of the fixture simulator function, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Balance device type
Data type	Character string type (String)
Range	Select from the following. • "SBALanced" Specifies the unbalance-balance (3 ports). • "BBALanced" Specifies the balance-balance (4 ports). • "SSBalanced" Specifies the unbalance-unbalance-balance (4 ports). • "BALanced" Specifies the unbalance (2 ports). (Rev. A.9.50 and above)
Preset value	For 4 port models, "SBALanced" For 2 port models, "BALanced"

Examples

```
Dim BalDev As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
BalDev = SCPI.CALCulate(1).FSIMulator.BALun.DEVice
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.BBALanced.PPORts

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SBALanced.PPORts

SCPI.CALCulate(Ch).FSIMulator.BALun.TOPology.SSBalanced.PPORts

Equivalent key

Analysis > Fixture Simulator > Topology > Device

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DEVice
{SBALanced|BBALanced|SSBalanced|BALanced}
:CALCulate{[1]-160}:FSIMulator:BALun:DEVice?
```

Query response

```
{SBAL|BBAL|SSB|BAL}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DEV BBAL"
20 OUTPUT 717;":CALC1:FSIM:BAL:DEV?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.C

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.C = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.C

Description

This command sets/gets the C value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit. BPORt(Bpt).TYPE object), for balance port 1 and balance port 2 (*Bpt*) of the selected channel (*Ch*),

Variable

Parameter	*Value*
Description	C value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	F (farad)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcC As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.C = 12E-12
DmcC = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.C
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **C**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:C <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:C?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:C?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.G

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.G = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.G

Description

This command sets/gets the G value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit. BPORt(Bpt).TYPE object), for the balance port 1 and balance port 2 (*Bpt*) of selected channel (*Ch*),

Variable

Parameter	*Value*
Description	G value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	S (siemens)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcG As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.G = 12E-12
DmcG = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.G
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **G**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:G <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:G?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:G?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.L

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.L = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.L

Description

This command sets/gets the L value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit. BPORt(Bpt).TYPE object), for the balance port 1 and balance port 2 (*Bpt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	L value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	H (henry)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcL As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.L = 12E-12
DmcL = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.L
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **L**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:L <numeric>
```

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:L?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:L 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:L?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.R = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).PARameters.R

Description

This command sets/gets the R value of the differential matching circuit consisting of shunt L and shunt C (PLPC is specified with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit. BPORt(Bpt).TYPE object), for the balance port 1 and balance port 2 (*Bpt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	R value of the differential matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	ohm
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DmcR As Double
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.R = 12E-12
DmcR = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).PARameters.R
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **R**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:R <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:PARameters:R?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R 12.3"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:PAR:R?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).TYPE

Description

For balance ports 1 and 2 (*Bpt*) of selected channel (*Ch*), selects the type of the differential matching circuit.

Variable

Parameter	*Param*
Description	Type of the differential matching circuit
Data type	Character string type (String)
Range	Select from the following: • "NONE": Specifies no-circuit. • "PLPC": Specifies the circuit that consists of shunt L and shunt C. • "USER": Specifies the user-defined circuit.
Preset value	"NONE"
Note	If you want to select the user-defined circuit, you must specify the 2-port touchstone file in which the proper information on the user-defined circuit is saved in advance. If you do not specify the appropriate file and you select the user-defined circuit, an error occurs when executed and NONE is automatically selected.

Examples

```
Dim CirType As String
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).TYPE = "plpc"
CirType = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).TYPE
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **Select Circuit**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}[:TYPE] {NONE|PLPC|USER}

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}[:TYPE]?

Query response

{NONE|PLPC|USER}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1 PLPC"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).USER.FILename

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).USER.FILename = *File*

File = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.BPORt(*Bpt*).USER.FILename

Description

This command specifies the file in which the information on the user-defined differential matching circuit is saved (2-port touchstone file with the .s2p extension), for the balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\ " (back slash), or "/" (slash).

Even if the specified file does not exist, no error occurs when you execute this object. However, when you set the type of the differential matching circuit to the user-defined circuit with the SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit. BPORt(Bpt).TYPE object, an error occurs when executed.

Variable

Parameter	*File*
Description	2-port touchstone file name (extension: .s2p) for the differential matching circuit
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim DmcUser As String
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).USER.FILename = "dmc.s2p"
```

DmcUser = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).USER.FILename
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.BPORt(1).TYPE = "user"

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **User File**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:USER:FILename <string>

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:BPORt{[1]|2}
:USER:FILename?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:USER:FIL ""Match_d.s2p"""
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:BPOR1:USER:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DMCircuit.STATe

Description

This command turns ON/OFF the differential matching circuit embedding function when the fixture simulator function is ON, for all the balance ports of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the differential matching circuit embedding function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the differential matching circuit embedding function. • False or OFF: Turns OFF the differential matching circuit embedding function.
Preset value	False or OFF

Examples

```
Dim DifMch As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.STATe = True
DifMch = SCPI.CALCulate(1).FSIMulator.BALun.DMCircuit.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.BALun.DMCircuit.BPORt(Bpt).TYPE

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter(Tr).STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff Matching** > **Diff Matching**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:BALun:DMCircuit:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DMC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:DMC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).IMAGinary

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).IMAGinary = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).IMAGinary

Description

This command sets/gets the impedance value (imaginary part) for the differential port impedance conversion function, for the balance ports 1 and 2 (*Bpt*) of selected channel (*Ch*),

Variable

Parameter	*Value*
Description	Impedance value (imaginary part) for the differential port impedance conversion function
Data type	Double precision floating point type (Double)
Range	-1E+18 to 1E+18
Preset value	0
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DImag As Double
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORt(1).IMAGinary = 200
DImag = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORt(1).IMAGinary
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).REAL

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff ZConversion** > **Port1(bal) Imag|Port2(bal) Imag|Port3(bal) Imag**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:IMAGinary <numeric>

:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:IMAGinary?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:IMAG 300"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:IMAG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).REAL

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).REAL = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).REAL

Description

This command sets/gets the impedance value (real part) for the differential port impedance conversion function, for the balance ports 1 and 2 (*Bpt*) of the selected channel (*Ch*).

- This command performs in the same way as "SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion. BPORt(Bpt).Z0.R"

Variable

Parameter	*Value*
Description	Impedance value (real part) for the differential port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	100
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DReal As Double
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORt(1).REAL = 200
DReal = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORt(1).REAL
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff ZConversion** > **Port1(bal) Real|Port2(bal) Real|Port3(bal) Real**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:REAL <numeric>
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:REAL?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:REAL 300"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:REAL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).Z0.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).Z0.R = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.BPORt(*Bpt*).Z0.R

Description

This command sets/gets the impedance value for the differential port impedance conversion function, for the balance ports 1 and 2 (*Bpt*) of selected channel (*Ch*).

- This command clears setting value of "SCPI.CALCulate(Ch).FSIMula-tor.BALun.DZConversion. BPORt(Bpt).IMAGinary"

Variable

Parameter	*Value*
Description	Impedance value for the differential port impedance conversion function
Data type	Double precision floating point type (Double)
Range	1E-3 to 1E7
Preset value	100
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim DZ0 As Double
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORt(1).Z0.R = 200
DZ0 = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.BPORt(1).Z0.R
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).REAL

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Diff ZConversion** > **Port1(bal) Real|Port2(bal) Real|Port3(bal) Real**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:Z0[:R] <numeric>
```

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion
:BPORt{[1]|2}:Z0[:R]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:Z0 300"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:BPOR1:Z0?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.BALun.DZConversion.STATe

Description

This command turns ON/OFF the differential port impedance conversion function when the fixture simulator function is ON, for all the balance ports of selected channel (*Ch)*.

Variable

Parameter	*Status*
Description	ON/OFF of the differential port impedance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the differential port impedance conversion function. • False or OFF: Turns OFF the differential port impedance conversion function.
Preset value	False or OFF

Examples

```
Dim DifZcon As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.STATe = True
DifZcon = SCPI.CALCulate(1).FSIMulator.BALun.DZConversion.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).REAL

SCPI.CALCulate(Ch).FSIMulator.BALun.DZConversion.BPORt(Bpt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Diff ZConversion > Diff ZConversion

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion:STATe
{ON|OFF|1|0}
:CALCulate{[1]-160}:FSIMulator:BALun:DZConversion:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:DZC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:DZC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).XXXX.PARameter.TNAMe.XXXX

As mentioned in Selecting the Active Channel/Trace, if you are using E5071C revision A.9.60 and above, you can select a trace by the trace name, provided it has been defined earlier. To define a trace name, use the following command:

:CALC{[1]-160}:PAR{[1]-16}:TNAMe:DATA

If the trace name is not defined but used, the following error occurs:

'51, Specified trace does not exist'

If the defined trace name already exist, the following error occurs:

'63, Duplicate trace name'

The following commands allow you to select the active trace directly by using TNAMe:

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAME.BALanced.DEFine

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAME.BBALanced.DEFine

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAME.SBALanced.DEFine

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAME.SSBALanced.DEFine

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAME.STATe

SCPI.CALCulate(Ch).PARameter.TNAME.DEFine

SCPI.CALCulate(Ch).PARameter.TNAME.SELect

SCPI.CALCulate(Ch).PARameter.TNAME.SPORt

SCPI.CALCulate*(Ch)*.FSIMulator.BALun.PARameter.TNAMe.BALanced.DEFine*(Name)*

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.BALanced.DEFine(*Name*) = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.BALanced.DEFine(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "balance", for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SDD11": Specifies Sdd11. • "SCD11": Specifies Scd11. • "SDC11": Specifies Sdc11. • "SCC11": Specifies Scc11.

Preset value	"SDD11"

Examples

```
Dim BalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bal"
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.BALanced.DEFine("duptrace") = "sdd11"
BalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.BALanced.DEFine("duptrace")
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator|[Meas]** > **Sdd11|Scd11|Sdc11|Scc11**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:BALanced[:DEFine] <string>,{SDD11|SCD11|SDC11|SCC11}
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:BALanced[:DEFine]? <string>
```

Query response

```
{SDD11|SCD11|SDC11|SCC11}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:BAL ""DUPTRACE"",SDD21"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:BAL? ""DUPTRACE"""
40 ENTER 717;A$
```

SCPI.CALCulate*(Ch)*.FSIMulator.BALun.PARameter.TNAMe.BBALanced.DEFine(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.BBALanced.DEFine(*Name*) = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.BBALanced.DEFine(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "balance-balance", for the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SDD11": Specifies Sdd11. • "SDD21": Specifies Sdd21. • "SDD12": Specifies

	Sdd12. • "SDD22": Specifies Sdd22. • "SCD11": Specifies Scd11. • "SCD21": Specifies Scd21. • "SCD12": Specifies Scd12. • "SCD22": Specifies Scd22. • "SDC11": Specifies Sdc11. • "SDC21": Specifies Sdc21. • "SDC12": Specifies Sdc12. • "SDC22": Specifies Sdc22. • "SCC11": Specifies Scc11. • "SCC21": Specifies Scc21. • "SCC12": Specifies Scc12. • "SCC22": Specifies Scc22. • "IMB1":Specifies Imbalance1. • "IMB2":Specifies Imbalance2. • "CMRR": Specifies CMRR (Sdd21/Scc21).
Preset value	"SDD11"

Examples

```
Dim BbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.BBALanced.DEFine("duptrace") = "sdd21"
BbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter.BBALanced.DEFine("duptrace")
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sdd11**|**Sdd21**|**Sdd12**|**Sdd22**|**Scd11**|**Scd21**|**Scd12**|**Scd22**|**Sdc11**| **Sdc21**|**Sdc12**|**Sdc22**|**Scc11**|**Scc21**|**Scc12**|**Scc22**|**Imbalance1**|**Imbalance2**|**Sdd21/Scc21**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:BBALanced[:DEFine] <string>, {SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC21|SDC12|SDC22|SCC11|SCC21|SCC12|SCC22|IMB1|IMB2| CMRR}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:BBALanced[:DEFine]? <string>

Query response

{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC21|SDC12|SDC22|SCC11|SCC21|SCC12|SCC22|IMB1|IMB2|CMRR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:BBAL ""DUPTRACE"",SDD21"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:BBAL? ""DUPTRACE"""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAMe.SBALanced.DEFine(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.SBALanced.DEFine(*Name*) = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.SBALanced.DEFine(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-balance", for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SSS11": Specifies Sss11. • "SDS21": Specifies Sds21. • "SSD12": Specifies Ssd12. • "SCS21": Specifies Scs21. • "SSC12": Specifies Ssc12.

	• "SDD22": Specifies Sdd22. • "SCD22": Specifies Scd22. • "SDC22": Specifies Sdc22. • "SCC22": Specifies Scc22. • "IMB": Specifies Imbalance. • "CMRR": Specifies CMRR (Sds21/Scs21). • "CMRR2": Specifies CMRR2 (Ssd12/Ssc12).
Preset value	"SSS11"

Examples

```
Dim SbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "sbal"
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.SBALanced.DEFine("duptrace") = "scs21"
SbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.SBALanced.DEFine("duptrace")
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sss11**|**Sds21**|**Ssd12**|**Scs21**|**Ssc12**|**Sdd22**|**Scd22**|**Sdc22**|**Scc22**|**Imbalance**|**Sds21/Scs21**|**Ssd12/Ssc12**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:SBALanced[:DEFine] <string>, {SSS11|SDS21|SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|CMRR|CMRR2}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:SBALanced[:DEFine]? <string>

Query response

{SSS11|SDS21|SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|CMRR|CMRR2}<newline> <^END>

Example of use

1596

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:SBAL ""DUPTRACE"",SDS21"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:SBAL? ""DUPTRACE"""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAMe.SSBALanced.DEFine(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.SSBalanced.DEFine(*Name*) = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.SSBalanced.DEFine(*Name*)

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-unbalance-balance", for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SSS11": Specifies Sss11. • "SSS21": Specifies Sss21. • "SSS12": Specifies Sss12. • "SSS22": Specifies Sss22. • "SDS31": Specifies Sds31.

	• "SDS32": Specifies Sds32. • "SSD13": Specifies Ssd13. • "SSD23": Specifies Ssd23. • "SCS31": Specifies Scs31. • "SCS32": Specifies Scs32. • "SSC13": Specifies Ssc13. • "SSC23": Specifies Ssc23. • "SDD33": Specifies Sdd33. • "SCD33": Specifies Scd33. • "SDC33": Specifies Sdc33. • "SCC33": Specifies Scc33. • "IMB1": Specifies Imbalance1. • "IMB2": Specifies Imbalance2. • "IMB3": Specifies Imbalance3. • "IMB4": Specifies Imbalance4. • "CMRR1": Specifies CMRR (Sds31/Scs31). • "CMRR2": Specifies CMRR (Sds32/Scs32).
Preset value	"SSS11"

Examples

```
Dim SsbPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "ssb"
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.SSBalanced.DEFine("duptrace") = "sds31"
SsbPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.SSBalanced.DEFine("duptrace")
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sss11**|**Sss21**|**Sss12**|**Sss22**|**Sds31**|**Sds32**|**Ssd13**|**Ssd23**|**Scs31**|**Scs32**|**Ssc13**|**Ssc23**|**Sdd33**|**Scd33**|**Sdc33**|**Scc33**|**Imbalance1**|**Imbalance2**|**Imbalance3**|**Imbalance4**|**Sds31/Scs31**|**Sds32/Scs32**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:SSBalanced[:DEFine] <string>,
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS32|SSC13|SSC23|SDD33|SCD33|
SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CMRR1|CMRR2}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:SSBalanced[:DEFine]? <string>

Query response

{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS32|SSC13|SSC23|SDD33|SCD33|SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CMRR1|CMRR2}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:SSB ""DUPTRACE"",SDS31"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:SSB? ""DUPTRACE"""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAMe.STATe(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.STATe(*Name*) = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter.TNAMe.STATe(*Name*)

Description

This command turns ON/OFF the balance-unbalance conversion function when the fixture simulator function is ON, for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Status*
Description	ON/OFF of the balance-unbalance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the balance-unbalance conversion function. • False or OFF: Turns OFF the balance-unbalance conversion function.
Preset value	False or OFF

Examples

```
Dim BalMode As Boolean
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"

SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.STATe("duptrace") = True
BalMode = SCPI.CALCulate(1).FSIMulator.BALun.PARameter.TNAMe.STATe("duptrace")
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **BalUn**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:STATe
<string>, {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter:TNAMe:STATe?
<string>
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR:TNAM:STAT ""DUPTRACE"",ON"
30 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:STAT? ""DUPTRACE"""
40 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).BALanced.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).BALanced.DEFine = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).BALanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "balance", for the selected trace (*Tr*) of the selected channel (*Ch*).

This command is available at the firmware revision A.9.50 and above.

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SDD11": Specifies Sdd11. • "SCD11": Specifies Scd11. • "SDC11": Specifies Sdc11. • "SCC11": Specifies Scc11.
Preset value	"SDD11"

Examples

```
Dim BalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bal"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).BALanced.DEFine = "sdd11"
BalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).BALanced.DEFine
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sdd11**|**Scd11**|**Sdc11**|**Scc11**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:BALanced[:DEFine] {SDD11|SCD11|SDC11|SCC11}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:BALanced[:DEFine]?

Query response

{SDD11|SCD11|SDC11|SCC11}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BAL SDD21"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).BBALanced.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).BBALanced.DEFine *= Param*

Param =
SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).BBALanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "balance-balance", for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SDD11": Specifies Sdd11. • "SDD21": Specifies Sdd21. • "SDD12": Specifies Sdd12. • "SDD22": Specifies Sdd22. • "SCD11": Specifies Scd11. • "SCD21": Specifies Scd21. • "SCD12": Specifies Scd12. • "SCD22": Specifies Scd22. • "SDC11": Specifies Sdc11. • "SDC21": Specifies Sdc21. • "SDC12": Specifies Sdc12. • "SDC22": Specifies Sdc22. • "SCC11": Specifies Scc11. • "SCC21": Specifies Scc21.

	• "SCC12": Specifies Scc12. • "SCC22": Specifies Scc22. • "IMB1":Specifies Imbalance1. • "IMB2":Specifies Imbalance2. • "CMRR": Specifies CMRR (Sdd21/Scc21).
Preset value	"SDD11"

Examples

```
Dim BbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).BBALanced.DEFine = "sdd21"
BbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).BBALanced.DEFine
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sdd11**|**Sdd21**|**Sdd12**|**Sdd22**|**Scd11**|**Scd21**|**Scd12**|**Scd22**|**Sdc11**| **Sdc21**|**Sdc12**|**Sdc22**|**Scc11**|**Scc21**|**Scc12**|**Scc22**|**Imbalance1**|**Imbalance2**|**Sdd21/Scc21**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:BBALanced[:DEFine]
{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC
21|SDC12|SDC22|SCC11|
SCC21|SCC12|SCC22|IMB1|IMB2| CMRR}
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:BBALanced[:DEFine]?
```

Query response

```
{SDD11|SDD21|SDD12|SDD22|SCD11|SCD21|SCD12|SCD22|SDC11|SDC
21|SDC12|SDC22|
SCC11|SCC21|SCC12|SCC22|IMB1|IMB2|CMRR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BBAL SDD21"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:BBAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).SBALanced.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).SBALanced.DEFine *= Param*

Param =
SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).SBALanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-balance", for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SSS11": Specifies Sss11. • "SDS21": Specifies Sds21. • "SSD12": Specifies Ssd12. • "SCS21": Specifies Scs21. • "SSC12": Specifies Ssc12. • "SDD22": Specifies Sdd22. • "SCD22": Specifies Scd22. • "SDC22": Specifies Sdc22. • "SCC22": Specifies Scc22. • "IMB": Specifies Imbalance. • "CMRR": Specifies CMRR (Sds21/Scs21). • "CMRR2": Specifies CMRR2 (Ssd12/Ssc12).
Preset	"SSS11"

value	

Examples

```
Dim SbalPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "sbal"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).SBALanced.DEFine = "scs21"
SbalPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).SBALanced.DEFine
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sss11**|**Sds21**|**Ssd12**|**Scs21**|**Ssc12**|**Sdd22**|**Scd22**|**Sdc22**|**Scc22**|**Imbalance**|**Sds21/Scs21**|**Ssd12/Ssc12**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:SBALanced[:DEFine] {SSS11|SDS21|
SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|CMRR|CMRR2}
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:SBALanced[:DEFine]?
```

Query response

```
{SSS11|SDS21|SSD12|SCS21|SSC12|SDD22|SCD22|SDC22|SCC22|IMB|
CMRR|CMRR2}<newline> <^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SBAL SDS21"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SBAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).SSBalanced.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).SSBalanced.DEFine *= Param*

Param =
SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).SSBalanced.DEFine

Description

This command sets/gets the measurement parameter when the balance device type is "unbalance-unbalance-balance", for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select from the following. • "SSS11": Specifies Sss11. • "SSS21": Specifies Sss21. • "SSS12": Specifies Sss12. • "SSS22": Specifies Sss22. • "SDS31": Specifies Sds31. • "SDS32": Specifies Sds32. • "SSD13": Specifies Ssd13. • "SSD23": Specifies Ssd23. • "SCS31": Specifies Scs31. • "SCS32": Specifies Scs32. • "SSC13": Specifies Ssc13. • "SSC23": Specifies Ssc23. • "SDD33": Specifies Sdd33. • "SCD33": Specifies Scd33. • "SDC33": Specifies Sdc33.

	• "SCC33": Specifies Scc33. • "IMB1": Specifies Imbalance1. • "IMB2": Specifies Imbalance2. • "IMB3": Specifies Imbalance3. • "IMB4": Specifies Imbalance4. • "CMRR1": Specifies CMRR (Sds31/Scs31). • "CMRR2": Specifies CMRR (Sds32/Scs32).
Preset value	"SSS11"

Examples

```
Dim SsbPara As String
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "ssb"
SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).SSBalanced.DEFine = "sds31"
SsbPara = SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).SSBalanced.DEFine
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator**|**[Meas]** > **Sss11**|**Sss21**|**Sss12**|**Sss22**|**Sds31**|**Sds32**|**Ssd13**|**Ssd23**|**Scs31**|**Scs32**|**Ssc13**|**Ssc23**|**Sdd33**|**Scd33**|**Sdc33**|**Scc33**|**Imbalance1**|**Imbalance2**|**Imbalance3**|**Imbalance4**|**Sds31/Scs31**|**Sds32/Scs32**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:SSBalanced[:DEFine]
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS3
2|SSC13|SSC23|SDD33|SCD33|
SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CMRR1|CMRR2}
```

```
:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}
:SSBalanced[:DEFine]?
```

Query response

```
{SSS11|SSS21|SSS12|SSS22|SDS31|SDS32|SSD13|SSD23|SCS31|SCS3
2|SSC13|SSC23|SDD33|SCD33|SDC33|SCC33|IMB1|IMB2|IMB3|IMB4|CM
RR1|CMRR2}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SSB SDS31"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:SSB?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*). STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.BALun.PARameter(*Tr*).STATe

Description

This command turns ON/OFF the balance-unbalance conversion function when the fixture simulator function is ON, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the balance-unbalance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the balance-unbalance conversion function. • False or OFF: Turns OFF the balance-unbalance conversion function.
Preset value	False or OFF

Examples

```
Dim BalMode As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).STATe = True
BalMode = SCPI.CALCulate(1).FSIMulator.BALun.PARameter(1).STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **BalUn**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:BALun:PARameter{[1]-16}:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:PAR1:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BALanced.PPORts

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BALanced.PPORts = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BALanced.PPORts

Description

This command assigns each port when the balance device type is "balance", for the selected channel (*Ch*).

To set the balance device type to "balance," specify BAL with the SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice object.

This command is available at the firmware revision A.9.50 and above.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0)* :Port number assigned to port a in Balance device type. • *Ports(1)* :Port number assigned to port b in Balance device type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 2
Preset value	Ports(0):1 / Ports(1):2
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim BbalPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bal"
```

```
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORts = Array(1,4)
BbalPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORts
```

```
Dim BbalPort(1) As Variant
Dim Ref As Variant
BbalPort(0) = 1
BbalPort(1) = 4
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORts = BbalPort
Ref = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BALanced.PPORts
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BALanced [:PPORts] <numeric 1>, <numeric 2>

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BALanced [:PPORts]?

Query response

{numeric 1},{numeric 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BAL 1,4"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BAL?"
30 ENTER 717;A,B
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BBALanced.PPORts

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BBALanced.PPORts = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.BBALanced.PPORts

Description

This command assigns each port when the balance device type is "balance-balance", for the selected channel (*Ch*).

To set the balance device type to "balance-balance," specify BBAL with the SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice object.

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port number). • *Ports(0)* :Port number assigned to port a in Balance device type. • *Ports(1)* :Port number assigned to port b in Balance device type. • *Ports(2)* :Port number assigned to port c in Balance device type. • *Ports(3)* :Port number assigned to port d in Balance device type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2 / *Ports*(2):3 / Ports(3):4
Resolution	1

Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim BbalPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORts = Array(3,4,1,2)
BbalPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORts
```

```
Dim BbalPort(3) As Variant
Dim Ref As Variant
BbalPort(0) = 3
BbalPort(1) = 4
BbalPort(2) = 1
BbalPort(3) = 2
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "bbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORts = BbalPort
Ref = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORts
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(bal)**

Analysis > **Fixture Simulator** > **Topology** > **Port2(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BBALanced [:PPORts] <numeric 1>, <numeric 2>,<numeric 3>,<numeric 4>

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:BBALanced [:PPORts]?

Query response

{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BBAL 1,2,3,4"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:BBAL?"
30 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.PROPerty.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.PROPerty.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.PROPerty.STATe

Description

This command turns on/off the property display for the topology setting when using the balance-unbalance conversion, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	On/off of the property display of the topology setting
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns on the property display. • False or OFF :Turns off the property display.
Preset value	False or OFF

Examples

```
Dim TopProp As Boolean
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.PROPerty.STATe = True
TopProp = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.PROPerty.STATe
```

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Property**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:BALun:TOPology
:PROPerty:STATe {ON|OFF|1|0}
```

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology
:PROPerty:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:PROP:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:PROP:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORts

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORts = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SBALanced.PPORts

Description

This command assigns each port when the balance device type is "unbalance-balance", for the selected channel (*Ch*).

To set the balance device type to "unbalance-balance," specify SBAL with the SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice object.

Variable

Parameter	*Ports*
Description	Indicates 3-element array data (port number). • *Ports(0)* :Port number assigned to port a in Balance device type. • *Ports(1)* :Port number assigned to port b in Balance device type. • *Ports(2)* :Port number assigned to port c in Balance device type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2 / *Ports*(2):3
Resolution	1

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim SbalPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "sbal"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.SBALanced.PPORts = Array(1,3,4)
SbalPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.BBALanced.PPORts
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(se)**

Analysis > **Fixture Simulator** > **Topology** > **Port2(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SBALanced [:PPORts] <numeric 1>, <numeric 2>,<numeric 3>

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SBALanced [:PPORts]?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SBAL 1,2,3"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SBAL?"
30 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SSBalanced.PPORts

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SSBalanced.PPORts = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.BALun.TOPology.SSBalanced.PPORts

Description

This channel assigns/reads each port when the balance device type is "unbalance-unbalance-balance", for the selected channel (*Ch*).

To set the balance device type to "unbalance-unbalance-balance," specify SSB with the SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice object.

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port number). • *Ports(0)* :Port number assigned to port a in Balance device type. • *Ports(1)* :Port number assigned to port b in Balance device type. • *Ports(2)* :Port number assigned to port c in Balance device type. • *Ports(3)* :Port number assigned to port d in Balance device type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2 / *Ports*(2):3 / Ports(3):4
Resolution	1

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set. If you specify the same port number to 2 or more port numbers, an error occurs when executed.

Examples

```
Dim SsbPort As Variant
SCPI.CALCulate(1).FSIMulator.BALun.DEVice = "ssb"
SCPI.CALCulate(1).FSIMulator.BALun.TOPology.SSBalanced.PPORts = Array(1,4,2,3)
SsbPort = SCPI.CALCulate(1).FSIMulator.BALun.TOPology.SSBalanced.PPORts
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.BALun.DEVice

Equivalent key

Analysis > **Fixture Simulator** > **Topology** > **Port1(se)**

Analysis > **Fixture Simulator** > **Topology** > **Port2(se)**

Analysis > **Fixture Simulator** > **Topology** > **Port3(bal)**

- When performing the operation from the front panel, set each port separately.

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SSBalanced [:PPORts] <numeric 1>, <numeric 2>,<numeric 3>,<numeric 4>

:CALCulate{[1]-160}:FSIMulator:BALun:TOPology:SSBalanced [:PPORts]?

Query response

{numeric 1},{numeric 2},{numeric 3},{numeric 4}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SSB 1,2,3,4"
20 OUTPUT 717;":CALC1:FSIM:BAL:TOP:SSB?"
30 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename = *File*

File = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).FILename

Description

This command specifies a file in which the information of network 1 to 4 (*Nwk*) (which you want to embed/de-embed using the 4 (or 6, 8)-port network embedding/de-embedding feature) is saved for the selected channel (*Ch*). The file is saved as a 4 (or 6, 8)-port touchstone file with the ".s4p" (or ".s6p", ".s8p") extension.

For information on network numbers, refer to Obtaining Characteristics After Embedding/De-embedding 4/6/8-port Network.

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\ " (back slash) or "/" (slash).

This function is available with the firmware revision 3.50 or above. The 6/8-port touchstone file is supported with the firmware revision 9.61 or above.

Variable

Parameter	*Nwk*
Description	Number of network
Data type	Long integer type (Long)
Range	1 to 4 1, 2 (4-port touchstone file only) 3 (6-port touchstone file only) 4 (8-port touchstone file only)
Preset value	1
Note	If the specified variable is out of the allowable setup

	range, an error occurs when executed.

Parameter	*File*
Description	4/6/8-port touchstone file name
Data type	Character string type (String)
Range	254 characters or less
Preset value	""
Note	When the processing type is set to NONE, even if the specified file does not exist, no error occurs when you execute this object. However, when you set the processing type to embedding/de-embedding with the SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE object, an error occurs.

Examples

```
Dim Emb As String
SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).FILename = "network.s4p"
Emb = SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).FILename
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE

SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe

Equivalent key

Firmware 9.61 or above

Analysis > **Fixture Simulator** > **De-Embedding SxP** > **Topology** > **User File (nwk1)**|**User File (nwk2)**|**User File (nwk3)**|**User File (nwk4)**

Firmware 9.60 or below

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** > **User File (nwk1)**|**User File (nwk2)**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}: FILename <string>

:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}: FILename?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:FIL ""Network.s4p"""
20 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).TYPE = *File*

File = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.NETWork(*Nwk*).TYPE

Description

This command sets/gets the processing type for networks 1 and 4 (*Nwk*), for the 4/6/8-port network embedding/de-embedding feature for the selected channel (*Ch*).

This function is available with the firmware revision 3.50 or above. The 6/8-port touchstone file is supported with the firmware revision 9.61 or above.

Variable

Parameter	*Param*
Description	Processing type
Data type	Character string type (String)
Range	Select from the following. • "NONE": Specifies no-processing. • "EMBed": Specifies embedding. • "DEEMbed": Specifies de-embedding.
Preset value	"NONE"
Note	Before selecting embedding/de-embedding, use the SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).FILename object to specify the 4/6/8-port touchstone file in which the information on the network is saved. If you do not specify the appropriate file and you select embedding/de-embedding, a runtime error occurs and NONE is automatically selected.

Examples

```
Dim EmbType As String
SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).FILename = "network.s4p"
```

```
SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).TYPE = "deem"
EmbType = SCPI.CALCulate(1).FSIMulator.EMBed.NETWork(1).TYPE
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).FILename

SCPI.CALCulate(Ch).FSIMulator.EMBed.STATe

Equivalent key

Firmware 9.61 or above

Analysis > **Fixture Simulator** > **De-Embedding SxP** > **Topology** > **Type (nwk1)**|**Type (nwk2)**|**Type (nwk3)**|**Type (nwk4)** > **None**|**Embed**|**De-Embed**

Firmware 9.60 or below

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** > **Type (nwk1)**|**Type (nwk2)**> **None**|**Embed**|**De-Embed**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}:TYPE {NONE|EMBed|DEEMbed}

:CALCulate{[1]-160}:FSIMulator:EMBed:NETWork{[1]-4}:TYPE?

Query response

{NONE|EMB|DEEM}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:TYPE DEEM"
20 OUTPUT 717;":CALC1:FSIM:EMB:NETW1:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.STATe

Description

This command turns ON/OFF the 4-port network embedding/de-embedding feature when the fixture simulator feature is *ON*, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Turns ON/OFF the 4-port network embedding/de-embedding feature.
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the 4-port network embedding/de-embedding feature. • False or OFF: Turns OFF the 4-port network embedding/de-embedding feature.
Preset value	False or 0

Examples

```
Dim Emb As Boolean
SCPI.CALCulate(1).FSIMulator.EMBed.STATe = True
Emb = SCPI.CALCulate(1).FSIMulator.EMBed.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).FILename

SCPI.CALCulate(Ch).FSIMulator.EMBed.NETWork(Nwk).TYPE

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **De-Embedding S4P**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:EMBed:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:EMB:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.A.PORTs

Description

This command sets/gets the test port assignment when the connection type (Topology) is set to A, for the 4-port network embedding/de-embedding feature for selected channel (*Ch*).

For information on the connection type (Topology), refer to Connection Type.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port numbers). • *Ports(0)* :Port number assigned to port a in Connection Type. • *Ports(1)* :Port number assigned to port b in Connection Type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2
Resolution	1
Note	If the specified variable is out of the allowable setting range, an error occurs when executed. If you specify an identical port number to multiple ports, a runtime error occurs.

Examples

```
Dim EnbPort As Variant
SCPI.CALCulate(1).FSIMulator.ENBed.TYPE = "a"
SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.A.PORTs = Array(2,1)
EnbPort = SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.A.PORTs
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** > **Ports** > **1-2**|**1-3**|**1-4**|**2-1**|**2-3**|**2-4**|**3-1**|**3-2**|**3-4**|**4-1**|**4-2**|**4-3**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:A:PORTs <value 1>,<value 2>

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:A:PORTs?

Query response

{value 1},{value 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TOP:A:PORT 1,2"
20 OUTPUT 717;":CALC1:FSIM:EMB:TOP:A:PORT?"
30 ENTER 717;A,B
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.B.PORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.B.PORTs = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.B.PORTs

Description

This command sets/gets test port assignment when the connection type (Topology) is set to B, for the 4-port network embedding/de-embedding feature for selected channel (*Ch*).

For information on the connection type (Topology), refer to Connection Type.

Variable

Parameter	*Ports*
Description	Indicates 3-element array data (port numbers). • *Ports(0)* :Port number assigned to port a in Connection Type. • *Ports(1)* :Port number assigned to port b in Connection Type. • *Ports(2)* :Port number assigned to port c in Connection Type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2 / *Ports*(2):3
Resolution	1
Note	If the specified variable is out of the allowable setting range, an error occurs when executed. If you specify an identical port number to multiple ports, a runtime error occurs.

Examples

```
Dim EnbPort As Variant
SCPI.CALCulate(1).FSIMulator.ENBed.TYPE = "b"
SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.B.PORTs = Array(1,3,2)
EnbPort = SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.B.PORTs
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** > **Ports** > **1-2-3**|**1-2-4**|**1-3-2**|**1-3-4**|**1-4-2**|**1-4-3**|**2-1-3**|**2-1-4**|**2-3-1**|**2-3-4**|**2-4-1**|**2-4-3**| **3-1-2**|**3-1-4**|**3-2-1**|**3-2-4**|**3-4-1**|**3-4-2**|**4-1-2**|**4-1-3**|**4-2-1**|**4-2-3**|**4-3-1**|**4-3-2**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:B:PORTs <value 1>,<value 2>,<value 3>

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:B:PORTs?

Query response

{value 1},{value 2},{value 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TOP:B:PORT 1,2,3"
20 OUTPUT 717;":CALC1:FSIM:EMB:TOP:B:PORT?"
30 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.C.PORTs

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.C.PORTs = *Ports*

Ports = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TOPology.C.PORTs

Description

This command sets/gets test port assignment when the connection type (Topology) is set to C, for the 4-port network embedding/de-embedding feature for selected channel (*Ch*).

For information on the connection type (Topology), refer to Connection Type.

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port numbers). • *Ports(0)* :Port number assigned to port a in Connection Type . • *Ports(1)* :Port number assigned to port b in Connection Type . • *Ports(2)* :Port number assigned to port c in Connection Type . • *Ports(3)* :Port number assigned to port d in Connection Type . The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Preset value	Ports(0):1 / Ports(1):2 / *Ports*(2):3 / Ports(3):4
Resolution	1
Note	If the specified variable is out of the allowable setting range, an error occurs when executed. If you specify an identical port number to multiple ports, a runtime error occurs.

Examples

```
Dim EnbPort As Variant
SCPI.CALCulate(1).FSIMulator.ENBed.TYPE = "c"
SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.C.PORTs = Array(1,4,2,3)
EnbPort = SCPI.CALCulate(1).FSIMulator.ENBed.TOPology.C.PORTs
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.TYPE

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** > **Ports** > **1-2-3-4|1-2-4-3|1-3-2-4|1-3-4-2|1-4-2-3|1-4-3-2|2-1-3-4|2-1-4-3|2-3-1-4|2-3-4-1| 2-4-1-3|2-4-3-1|3-1-2-4|3-1-4-2|3-2-1-4|3-2-4-1|3-4-1-2|3-4-2-1|4-1-2-3|4-1-3-2| 4-2-1-3|4-2-3-1|4-3-1-2|4-3-2-1**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:C:PORTs <value 1>,<value 2>,<value 3>,<value 4>

:CALCulate{[1]-160}:FSIMulator:EMBed:TOPology:C:PORTs?

Query response

{value 1},{value 2},{value 3},{value 4}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TOP:C:PORT 1,2,3,4"
20 OUTPUT 717;":CALC1:FSIM:EMB:TOP:C:PORT?"
30 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TYPE = *File*

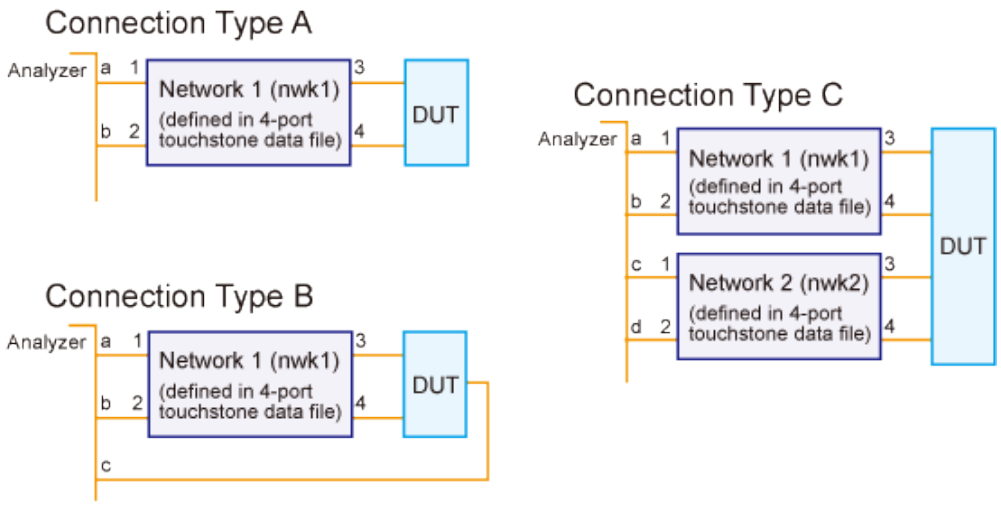
File = SCPI.CALCulate(*Ch*).FSIMulator.EMBed.TYPE

Description

This command selects a connection type (Topology), for the 4-port network embedding/de-embedding feature for channels 1 to 36 (*Ch*).

- This function is available with the Firmware revision 3.50 or greater.

Connection Type



e5071c313

Variable

Parameter	*Param*
Description	Connection type (refer to Connection Type)
Data type	Character string type (String)
Range	Select from the following. • "A": Specifies connection

	type A. • "B": Specifies connection type B. • "C":Specifies connection type C.
Preset value	"A"

Examples

```
Dim EmbType As String
SCPI.CALCulate(1).FSIMulator.EMBed.TYPE = "b"
EmbType = SCPI.CALCulate(1).FSIMulator.EMBed.TYPE
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.A.PORTs

SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.B.PORTs

SCPI.CALCulate(Ch).FSIMulator.EMBed.TOPology.C.PORTs

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding S4P** > **Topology** > **Select Topology** > **A|B|C**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:EMBed:TYPE {A|B|C}

:CALCulate{[1]-160}:FSIMulator:EMBed:TYPE?

Query response

{A|B|C}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:EMB:TYPE A"
20 OUTPUT 717;":CALC1:FSIM:EMB:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).TYPE

Description

This command sets/gets the type of the network de-embedding, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Pt*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Param*
Description	Type of the network de-embedding
Data type	Character string type (String)
Range	Select from the following. • "NONE": Specifies no network de-embedding. • "USER": Specifies the user-defined network de-embedding.

Preset value	"NONE"
Note	If you want to select the user-defined network de-embedding, you must specify the 2-port touchstone file in which the information on the user-defined network is saved in advance. If you do not specify the appropriate file and you select the user-defined network de-embedding, an error occurs when executed and NONE is automatically selected.

Examples

```
Dim DeemType As String
SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.PORT(1).USER.FILename = "network.s2p"
SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.PORT(1).TYPE = "user"
DeemType = SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.PORT(1).TYPE
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.PORT(Pt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.STATe

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ORIentation.STATe

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding** > **Select Type**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDed:DEEMbed:PORT{[1]|2|3|4}[:TYPE] {NONE|USER}
:CALCulate{[1]-160}:FSIMulator:SENDed:DEEMbed:PORT{[1]|2|3|4}[:TYPE]?
```

Query response

```
{NONE|USER}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1 USER"
20 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).USER.FILename

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).USER.FILename = *File*

File = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.PORT(*Pt*).USER.FILename

Description

This command specifies the file in which the information on the user-defined network for the network de-embedding function is saved for ports 1 and 4 (*Pt*) of selected channel (*Ch*). This file is saved as a 2-port touchstone file with the .s2p extension.

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\ " (back slash), or "/" (slash).

Even if the specified file does not exist, no error occurs when you execute this object. However, when you set the type of the network de-embedding to the user-defined network with the SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed. PORT(Pt).TYPE object, an error occurs.

Variable

Parameter	*File*
Description	2-port touchstone file name (extension: .s2p) for the network de-embedding function
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim DeemUser As String
SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.PORT(1).USER.FILename = "D:\folder\network.s2p"
```

DeemUser = SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.PORT(1).USER.FILename
SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.PORT(1).TYPE = "user"

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.STATe

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding** > **User File**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:DEEMbed:PORT{[1]|2|3|4}:USER:FILename <string>

:CALCulate{[1]-160}:FSIMulator:SENDed:DEEMbed:PORT{[1]|2|3|4}:USER:FILename?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1:USER:FIL ""D:\folder\Network.s2p"""
20 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:PORT1:USER:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.DEEMbed.STATe

Description

This command turns ON/OFF the network de-embedding function when the fixture simulator function is ON, for all the ports of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the network de-embedding function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the network de-embedding function. • False or OFF: Turns OFF the network de-embedding function.
Preset value	False or 0

Examples

```
Dim Deemb As Boolean
SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.STATe = True
Deemb = SCPI.CALCulate(1).FSIMulator.SENDed.DEEMbed.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.PORT(Pt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **De-Embedding** > **De-Embedding**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:DEEMbed:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:SENDed:DEEMbed:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:SEND:DEEM:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.C

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.C *= Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.C

Description

This command sets the C value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit. PORT(Pt).TYPE object, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	C value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	F (farad)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PmcC As Double
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.C = 12E-12
PmcC = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.C
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).TYPE
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.G
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.L
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.R
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **C**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:C <numeric>

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:C?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:C 12.3"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:C?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.G

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.G *= Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.G

Description

This command sets/gets the G value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit. PORT(Pt).TYPE object, for the ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	G value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	S (siemens)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PmcG As Double
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.G = 12E-12
PmcG = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.G
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).TYPE
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.C
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.L
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.R
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **G**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:G <numeric>

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:G?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:G 12.3"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:G?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.L

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.L = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.L

Description

This command sets the L value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit. PORT(Pt).TYPE object, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	L value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	H (henry)
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PmcL As Double
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.L = 12E-12
PmcL = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.L
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).TYPE
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.C
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.G
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.R
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **L**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:L <numeric>

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:L?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:L 12.3"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:L?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).PARameters.R = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*). PARameters.R

Description

This command sets/gets the R value of the matching circuit specified with the SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit. PORT(Pt).TYPE object, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	R value of the matching circuit
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	0
Unit	ohm
Resolution	1E-18
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PmcR As Double
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.R = 12E-12
PmcR = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).PARameters.R
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).TYPE
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.C
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.G
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.L
SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **R**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:R <numeric>

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:PARameters:R?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:R 12.3"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:PAR:R?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).TYPE

Description

This command selects the type of the matching circuit, for ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Type of the matching circuit
Data type	Character string type (String)
Range	Select from the following. • "NONE": Specifies no-circuit. • "SLPC": Specifies the circuit that consists of series L and shunt C. • "PCSL": Specifies the circuit that consists of shunt C and series L. • "PLSC": Specifies the circuit that consists of shunt L and series C. • "SCPL": Specifies the circuit that consists of series C and shunt L. • "PLPC": Specifies the circuit that consists of shunt L and shunt C. • "USER": Specifies the user-defined circuit.
Preset value	"NONE"
Note	If you want to select the user-defined circuit, you must specify the 2-port touchstone file in which the proper information on the user-defined circuit is saved in advance. If you do not specify the appropriate file and you select the user-defined circuit, an error occurs when

	executed and NONE is automatically selected.

Examples

```
Dim CirType As String
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).TYPE = "slpc"
CirType = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).TYPE
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **Select Circuit**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}[:TYPE] {NONE|SLPC|PCSL|PLSC|SCPL|PLPC|USER}

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}[:TYPE]?

Query response

{NONE|SLPC|PCSL|PLSC|SCPL|PLPC|USER}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1 SLPC"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).USER.FILename

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).USER.FILename = *File*

File = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.PORT(*Pt*).USER.FILename

Description

This command specifies the file in which the information on the user-defined matching circuit is saved (2-port touchstone file), for the port 1 and 4 (*Pt*) of selected channel (*Ch*).

Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\ " (back slash), or "/" (slash).

Even if the specified file does not exist, no error occurs when you execute this object. However, when you set the type of the matching circuit to the user-defined circuit with the SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit. PORT(Pt).TYPE object, an error occurs.

Variable

Parameter	*File*
Description	2-port touchstone file name (extension: .s2p) for the matching circuit
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim PmcUser As String
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).USER.FILename = "match.s2p"
```

PmcUser = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).USER.FILename
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.PORT(1).TYPE = "user"

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **User File**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:USER:FILename <string>

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:PORT{[1]|2|3|4}:USER:FILename?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:USER:FIL ""Match.s2p"""
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:PORT1:USER:FIL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.PMCircuit.STATe

Description

This command turns ON/OFF the matching circuit embedding function when the fixture simulator function is ON, for all the ports of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the matching circuit embedding function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the matching circuit embedding function. • False or OFF: Turns OFF the matching circuit embedding function.
Preset value	False or OFF

Examples

```
Dim Pmcir As Boolean
SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.STATe = True
Pmcir = SCPI.CALCulate(1).FSIMulator.SENDed.PMCircuit.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).TYPE

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.C

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.G

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.L

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).PARameters.R

SCPI.CALCulate(Ch).FSIMulator.SENDed.PMCircuit.PORT(Pt).USER.FILename

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port Matching** > **Port Matching**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:SENDed:PMCircuit:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:PMC:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:SEND:PMC:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).IMAGinary

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).IMAGinary = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).IMAGinary

Description

This command sets/gets the impedance value (imaginary part) for the port impedance conversion function, for the ports 1 and 4 (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Impedance value (imaginary part) for the port impedance conversion function
Data type	Double precision floating point type (Double)
Range	-1E+18 to 1E+18
Preset value	0
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ZImag As Double
SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.PORT(1).IMAGinary = -9.2E10
ZImag = SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.PORT(1).IMAGinary
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).REAL

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > Fixture Simulator > Port ZConversion > Port1 Z0 Imag|Port2 Z0 Imag|Port3 Z0 Imag|Port4 Z0 Imag

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion
:PORT{[1]|2|3|4}:IMAGinary <numeric>

:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion
:PORT{[1]|2|3|4}:IMAGinary?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:IMAG 75"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:IMAG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).REAL

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).REAL = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).REAL

Description

This command sets/gets the impedance value (real part) for the port impedance conversion function, for ports 1 and 4 (*Pt*) of the selected channel (*Ch*).

- This command performs in the same way as "SCPI.CALCulate(Ch).FSIM-ulator.SENDed.ZCONversion. PORT(Pt).Z0.R"

Variable

Parameter	*Value*
Description	Impedance value (real part) for the port impedance conversion function
Data type	Double precision floating point type (Double)
Range	0.001 to 1E7
Preset value	50
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ZReal As Double
SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.PORT(1).REAL = 3.7E5
ZReal = SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.PORT(1).REAL
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port ZConversion** > **Port1 Z0 Real**|**Port2 Z0 Real**|**Port3 Z0 Real**|**Port4 Z0 Real**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion
:PORT{[1]|2|3|4}:REAL <numeric>
```

```
:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion
:PORT{[1]|2|3|4}:REAL?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:REAL 75"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:REAL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).Z0.R

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).Z0.R = *Value*

Value = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.PORT(*Pt*).Z0.R

Description

This command sets/gets the impedance value for the port impedance conversion function, for ports 1 and 4 (*Pt*) of the selected channel (*Ch*).

- This command clears setting value of SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion. PORT(Pt).IMAGinary

Variable

Parameter	*Value*
Description	Impedance value for the port impedance conversion function
Data type	Double precision floating point type (Double)
Range	0.001 to 1E7
Preset value	50
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ZconR As Double
SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.PORT(1).Z0.R = 75
ZconR = SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.PORT(1).Z0.R
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).REAL

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.STATe

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port ZConversion** > **Port1 Z0 Real**|**Port2 Z0 Real**|**Port3 Z0 Real**|**Port4 Z0 Real**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion
:PORT{[1]|2|3|4}:Z0[:R] <numeric>

:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion
:PORT{[1]|2|3|4}:Z0[:R]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:Z0 75"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:PORT1:Z0?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.SENDed.ZCONversion.STATe

Description

This command turns ON/OFF the port impedance conversion function when the fixture simulator function is ON, for all the ports of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the port impedance conversion function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the port impedance conversion function. • False or OFF: Turns OFF the port impedance conversion function.
Preset value	False or OFF

Examples

```
Dim Zcon As Boolean
SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.STATe = True
Zcon = SCPI.CALCulate(1).FSIMulator.SENDed.ZCONversion.STATe
```

Related objects

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).IMAGinary

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).REAL

SCPI.CALCulate(Ch).FSIMulator.SENDed.ZCONversion.PORT(Pt).Z0.R

SCPI.CALCulate(Ch).FSIMulator.STATe

Equivalent key

Analysis > **Fixture Simulator** > **Port ZConversion** > **Port ZConversion**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:SENDed:ZCONversion:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:SEND:ZCON:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).FSIMulator.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).FSIMulator.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).FSIMulator.STATe

Description

This command turns ON/OFF the fixture simulator function of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the fixture simulator function
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the fixture simulator function. • False or OFF: Turns OFF the fixture simulator function.
Preset value	False or OFF

Examples

```
Dim FxtSim As Boolean
SCPI.CALCulate(1).FSIMulator.STATe = True
FxtSim = SCPI.CALCulate(1).FSIMulator.STATe
```

Equivalent key

Analysis > **Fixture Simulator** > **Fixture Simulator**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:FSIMulator:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:FSIMulator:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FSIM:STAT ON"
20 OUTPUT 717;":CALC1:FSIM:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARameter.TNAMe.DEFine(*Name*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter.TNAMe.DEFine(*Name*) = *Param*

Param = SCPI.CALCulate(*Ch*).PARameter.TNAMe.DEFine(*Name*)

Description

This command sets/gets the measurement parameter of the selected trace for the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select either one of the following: • "S<XY>" Where: x=1to 4 Y=1to 4 • A|B|C|D • R<X> (X=1-4) • AUX1 or AUX2

Preset value	"S11"

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"
SCPI.CALCulate(1).PARameter.TNAMe.DEFine("duptrace") = "S21"
MeasPara = SCPI.CALCulate(1).PARameter.TNAMe.DEFine("duptrace")
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA

SCPI.CALCulate(Ch).PARameter.TNAME.SELect

SCPI.CALCulate(Ch).PARameter.TNAME.SPORt

Equivalent key

Meas > **S<XY>** {X=1-4;Y=1-4}|

Meas > **Absolute** > **R<X>** {X=1-4}|**A**|**B**|**C**|**D**|

Meas > **Aux Input 1**|**Aux Input 2**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter:TNAMe:DEFine <string>, {S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44|A|B|C|D|R1|R2|R3|R4|Aux1|AUX2}

:CALCulate{[1]-160}:PARameter:TNAMe:DEFine? <string>

Query response

{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44|A|B|C|D|R1|R2|R3|R4|AUX1|AUX2}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:PAR:TNAM:DEF ""DUPTRACE"", S21"
30 OUTPUT 717;":CALC1:PAR:TNAM:DEF? ""DUPTRACE"""
40 ENTER 717;A$
```

SCPI.CALCulate(Ch).PARameter.TNAMe.SELect

Object type

Method (**Write Only**)

Syntax

SCPI.CALCulate(*Ch*).PARameter.TNAMe.SELect = *Name*

Description

This command sets the measurement parameter of the selected trace for the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command.

Variable

Parameter	*Name*
Description	Name of the trace
Data type	Character string type (String)

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARameter.TNAMe.SELect = "duptrace"
```

Equivalent key

Trace Prev / Trace Next

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter:TNAMe:SELect <string>

Example of use

```
10 OUTPUT 717;":CALC1:PAR:TNAM:SEL ""DUPTRACE"""
```

SCPI.CALCulate(Ch).FSIMulator.BALun.PARameter.TNAMe.SPORt(*Name*)

Type of object

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter.TNAMe.SPORt(*Name*) = *Value*

Value = SCPI.CALCulate(*Ch*).PARameter.TNAMe.SPORt(*Name*)

Description

This command sets/gets the output port used for absolute or AUX measurements, for the selected trace of the selected channel (*Ch*). The trace can be selected by the trace name once its defined earlier with SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA command. This function is available in the revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace Name
Data type	Character string type (String)

Parameter	*Value*
Description	Setting of the output port
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	You need to set the measurement parameter for absolute/AUX measurements with the "SCPI.CALCulate(Ch).PARameter(Tr).DEFine" command.

Example of use

```
Dim Sport As Long
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "duptrace"
```

```
SCPI.CALCulate(1).PARameter.TNAMe.DEFine("duptrace") = "B"
SCPI.CALCulate(1).PARameter.TNAMe.SPORt("duptrace") = 4
Sport = SCPI.CALCulate(1).PARameter.TNAMe.SPORt("duptrace")
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).DEFine

Equivalent key

Meas > **Absolute** > **A(x)** …**D(x)**…**R1(x)** …**R4(x)** (x: 1 to 4)

Meas > **AUX Input 1** or **AUX Input 2** > **Sweep Port**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter:TNAMe:SPORt <string>,<numeric>

:CALCulate{[1]-160}:PARameter:TNAMe:SPORt? <string>

Query response

{value}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR2:TNAM:DATA ""DUPTRACE"""
20 OUTPUT 717;":CALC1:PAR:TNAM:DEF ""DUPTRACE"", B"
30 OUTPUT 717;":CALC1:PAR:TNAM:SPOR ""DUPTRACE"", 4"
40 OUTPUT 717;":CALC1:PAR:TNAM:SPOR? ""DUPTRACE"""
50 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine = *Param*

Param = SCPI.CALCulate(*Ch*).PARameter(*Tr*).DEFine

Description

This command sets/gets the measurement parameter of the selected trace (*Tr*), for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select either one of the following: • "S<XY>" Where: x=1to 4 Y=1to 4 • A|B|C|D • R<X> (X=1-4) • AUX1 or AUX2
Preset value	"S11"

Examples

```
Dim MeasPara As String
SCPI.CALCulate(1).PARameter(1).DEFine = "s21"
MeasPara = SCPI.CALCulate(1).PARameter(1).DEFine
```

Equivalent key

Meas > **S<XY>** {X=1-4;Y=1-4}|

Meas > **Absolute** > **R<X>** {X=1-4}|**A**|**B**|**C**|**D**|

Meas > **Aux Input 1**|**Aux Input 2**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter{[1]-16}:DEFine <string>{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44|A|B|C|D|R1|R2|R3|R4|Aux1|AUX2}

:CALCulate{[1]-160}:PARameter{[1]|-16}:DEFine?

Query response

{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44|A|B|C|D|R1|R2|R3|R4|AUX1|AUX2}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:DEF S21"
20 OUTPUT 717;":CALC1:PAR1:DEF?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SELect

Object type

Method (**Write-only**)

Syntax

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SELect

Description

This command sets/gets the selected trace (*Tr*) of selected channel (*Ch*) to the active trace.

You can set only a trace displayed to the active trace. If this object is used to set a trace not displayed to the active trace, an error occurs when executed and the object is ignored. (No read).

In revision A.9.60 and above, you can select the trace by the trace name, provided it has been defined earlier. To define a trace name, execute SCPI.CALCulate(Ch).PARameter(Tr).TNAMe.DATA. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

In revision A.9.60 and above, you can execute the action directly by using direct channel and trace assign commands like SCPI.CALCulate(Ch).TRACe(Tr).XXXX.XXXX.

Variable

Parameter	*Tr*
Description	Trace number
Data type	Long integer type (Long)
Range	1 to 16
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

SCPI.CALCulate(2).PARameter(2).SELect

Related objects

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.MEDium

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.TIME

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.WGCutoff

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

Equivalent key

Trace Prev / Trace Next

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter{[1]-16}:SELect

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:SEL"
```

SCPI.CALCulate(*Ch*).PARameter(*Tr*).TNAMe.DATA

Object type

Property (**Read-Write**)

Syntax

This option is only available In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).PARameter(*Tr*).TNAMe.DATA = *Name*

Name = SCPI.CALCulate(*Ch*).PARameter(*Tr*).TNAMe.DATA

Description

This command defines/gets the name for the active trace of selected channel (*Ch*).

This command is only available in revision A.9.60 and above.

Variable

Parameter	*Name*
Description	Trace name for the active trace
Data type	Character string type (String)

Examples

```
Dim TraceName As String
SCPI.CALCulate(1).PARameter(1).TNAMe.DATA = "DuplexTrace"
TraceName = SCPI.CALCulate(1).PARameter(Tr).TNAMe.DATA = Param
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter{[1]-16}:TNAMe:DATA = *Param*

:CALCulate{[1]-160}:PARameter{[1]-16}:TNAMe:DATA?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:TNAM:DATA ""DUPLEXTRACE"""
20 OUTPUT 717;":CALC1:PAR1:TNAM:DATA?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORt

Type of object

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORt = *Value*

Value = SCPI.CALCulate(*Ch*).PARameter(*Tr*).SPORt

Description

This command sets/gets the output port used for absolute or AUX measurements, for the selected trace (*Tr*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of the output port
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	You need to set the measurement parameter for absolute/AUX measurements with the "SCPI.CALCulate(Ch).PARameter(Tr).DEFine" command.

Example of use

```
Dim Sport As Long
SCPI.CALCulate(1).PARameter(1).DEFine = "B"
SCPI.CALCulate(1).PARameter(1).SPORt = 4
Sport = SCPI.CALCulate(1).PARameter(1).SPORt
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).DEFine

Equivalent key

Meas > **Absolute** > **A(x)** …**D(x)**…**R1(x)** …**R4(x)** (x: 1 to 4)

Meas > **AUX Input 1** or **AUX Input 2** > **Sweep Port**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter{[1]-16}:SPORt <numeric>

:CALCulate{[1]-160}:PARameter{[1]-16}:SPORt?

Query response

{value}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR1:DEF B"
20 OUTPUT 717;":CALC1:PAR1:SPOR 4"
30 OUTPUT 717;":CALC1:PAR1:SPOR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).PARameter.COUNt

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).PARameter.COUNt = *Value*

Value = SCPI.CALCulate(*Ch*).PARameter.COUNt

Description

This command sets/gets the number of traces of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of traces
Data type	Long integer type (Long)
Range	Varies depending on the upper limit setting for the channel/trace number.
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TraceNum As Long
SCPI.CALCulate(1).PARameter.COUNt = 4
TraceNum = SCPI.CALCulate(1).PARameter.COUNt
```

Equivalent key

Display > **Num of Traces**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}:PARameter:COUNt <numeric>

:CALCulate{[1]-160}:PARameter:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:PAR:COUN 4"
20 OUTPUT 717;":CALC1:PAR:COUN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).TRACe(*Tr*).XXXX.XXXX

As mentioned in Selecting the Active Channel/Trace, if you are using E5071C revision A.9.60 and above, you can select the trace directly by using TRAC{1-16} for all SELected commands. As such, you do not need to make a trace active before assigning a command to it.

The following commands allow you to select the trace directly by using TRAC{1-16}. Note that the active trace is not changed by this command:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DB
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.MARKer
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.VALue
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.FAIL
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MAXimum
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.MINimum
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.REPort.DATA
SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.STATe
SCPI.CALCulate(Ch).TRACe(Tr).CONVersion.FUNCtion
SCPI.CALCulate(Ch).TRACe(Tr).CONVersion.STATe
SCPI.CALCulate(Ch).TRACe(Tr).CORRection.EDELay.MEDium
SCPI.CALCulate(Ch).TRACe(Tr).CORRection.EDELay.TIME
SCPI.CALCulate(Ch).TRACe(Tr).CORRection.EDELay.WGCutoff
SCPI.CALCulate(Ch).TRACe(Tr).CORRection.OFFSet.PHASe
SCPI.CALCulate(Ch).TRACe(Tr).DATA.FDATa
SCPI.CALCulate(Ch).TRACe(Tr).DATA.FMEMory
SCPI.CALCulate(Ch).TRACe(Tr).DATA.SDATa
SCPI.CALCulate(Ch).TRACe(Tr).DATA.SMEMory
SCPI.CALCulate(Ch).TRACe(Tr).DATA.XAXis
SCPI.CALCulate(Ch).TRACe(Tr).EQUation.STATE
SCPI.CALCulate(Ch).TRACe(Tr).EQUation.TEXT
SCPI.CALCulate(Ch).TRACe(Tr).EQUation.VALid
SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. CENTer
SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. SHAPe
SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. SPAN

SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. STARt
SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. STATe
SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. STOP
SCPI.CALCulate(Ch).TRACe(Tr).FILTer.GATE.TIME. TYPE
SCPI.CALCulate(Ch).TRACe(Tr).FORMat
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.DATA
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.DOMain.COUPle
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.DOMain.STARt
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.DOMain.STATe
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.DOMain.STOP
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.EXECute
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.PEXCursion
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.POINts
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.PPOLarity
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.TARGet
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.TTRansition
SCPI.CALCulate(Ch).TRACe(Tr).FUNCtion.TYPE
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.DATA
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.DISPlay.STATe
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.FAIL
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.AMPLitude
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.MARKer
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.OFFSet.STIMulus
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.ALL
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.DATA
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.REPort.POINts
SCPI.CALCulate(Ch).TRACe(Tr).LIMit.STATe
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).ACTivate
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).BWIDth.DATA
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.BWIDth.STATe
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).BWIDth.THReshold
SCPI.CALCulate(Ch).TRACe(Tr).MARKer.COUPle
SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).DATA

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).DISCrete

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.DOMain.MULTiple.RANGe

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.DOMain.MULTiple.STARt

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.DOMain.MULTiple.STATe

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.DOMain.MULTiple.STOP

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.EXECute

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.PEXCursion

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.PPOLarity

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.TARGet

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.TRACking

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.TTRansition

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.NOTCh

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).NOTCh.DATA

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).NOTCh.THReshold

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).SET

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).STATe

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).X

SCPI.CALCulate(Ch).TRACe(Tr).MARKer(Mk).Y

SCPI.CALCulate(Ch).TRACe(Tr).MARKer.REFerence.STATe

SCPI.CALCulate(Ch).TRACe(Tr).MATH.FUNCtion

SCPI.CALCulate(Ch).TRACe(Tr).MATH.MEMorize

SCPI.CALCulate(Ch).TRACe(Tr).MIXer.XAXis

SCPI.CALCulate(Ch).TRACe(Tr).MSTatistics.DATA

SCPI.CALCulate(Ch).TRACe(Tr).MSTatistics.STATe

SCPI.CALCulate(Ch).TRACe(Tr).OFFset.XAXis

SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DATA
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.LINE
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.SELect
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.DISPlay.VALue
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.FAIL
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.REPort.DATA
SCPI.CALCulate(Ch).TRACe(Tr).RLIMit.STATe
SCPI.CALCulate(Ch).TRACe(Tr).SMOothing.APERture
SCPI.CALCulate(Ch).TRACe(Tr).SMOothing.STATe
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.CENTer
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.IMPulse.WIDTh
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.KBESsel
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.LPFRequency
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.SPAN
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.STARt
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.STATe
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.STEP.RTIMe
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.STIMulus
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.STOP
SCPI.CALCulate(Ch).TRACe(Tr).TRANsform.TIME.TYPE

SCPI.CALCulate(*Ch*).SELected.BLIMit.DB

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.BLIMit.DB = Value

Value = SCPI.CALCulate(*Ch*).SELected.BLIMit.DB

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.DB = Value

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.DB

Description

This command sets/gets the bandwidth threshold value (attenuation from the peak) of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and set/get the bandwidth threshold value for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Bandwidth N dB points.
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	0
Unit	dB

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimDB As Double
SCPI.CALCulate(1).SELected.BLIMit.DB = 3
BLimDB = SCPI.CALCulate(1).SELected.BLIMit.DB
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

Analysis > **Bandwidth Limit** > **N dB Points**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit:DB <numeric>

:CALCulate{[1]-160}[:SELected]:BLIMit:DB?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DB <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DB?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DB 3"
20 OUTPUT 717;":CALC1:BLIM:DB?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.MARKer

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer = Status

Status = SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.MARKer = Status

Status = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.MARKer

Description

This command turns ON/OFF the marker display of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the marker display of the bandwidth tests for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the bandwidth marker.
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the bandwidth marker. • False or OFF: Turns OFF the bandwidth marker.
Preset	False or OFF

value	

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimMk As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.DISPlay.MARKer = True
BLimMk = SCPI.CALCulate(1).SELected.BLIMit.DISPlay.MARKer
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue

Equivalent key

Analysis > **Bandwidth Limit** > **BW Marker**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit:DISPlay:MARKer {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:BLIMit:DISPlay:MARKer?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:MARKer {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:MARKer?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DISP:MARK ON"
20 OUTPUT 717;":CALC1:BLIM:DISP:MARK?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.DISPlay.VALue

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue = Status

Status = SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.VALue = Status

Value = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.DISPlay.VALue

Description

This command turns ON/OFF the bandwidth value display of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the bandwidth value display of the bandwidth tests for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the bandwidth display of the bandwidth test.
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the bandwidth display. • False or OFF: Turns OFF the bandwidth display.
Preset value	False or OFF

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimVal As Boolean
SCPI.CALCulate(1).PARameter(1).SELectSCPI.CALCulate(1).SELected.BLIMit.DISPlay.VALue = True
BLimVal = SCPI.CALCulate(1).SELected.BLIMit.DISPlay.VALue
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer

Equivalent key

Analysis > **Bandwidth Limit** > **BW Display**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit:DISPlay:VALue {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:BLIMit:DISPlay:VALue?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:VALue {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:DISPlay:VALue?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:DISP:VAL ON"
20 OUTPUT 717;":CALC1:BLIM:DISP:VAL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.FAIL

Object type

Property (**Read Only**)

Syntax

All revision:

Status = SCPI.CALCulate(Ch).SELected.BLIMit.FAIL

In revision A.9.60 and above, the following syntax is also supported:

Status = SCPI.CALCulate(Ch).TRACe(Tr).BLIMit.FAIL

Description

This command get the bandwidth limit test results, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and get the bandwidth limit test results for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	The bandwidth limit test result
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the bandwidth limit test result is FAIL. • False or OFF: Turns OFF the bandwidth limit test result is PASS.
Note	When the bandwidth limit test if set to OFF, False or OFF is always read out.

For information on the variable (*Ch*), see Variable.

Examples

```
Dim Result As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.STATe = True
Result = SCPI.CALCulate(1).SELected.BLIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit:FAIL?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:FAIL?

Query response

{1|0}<newline><^END>

	Description
1	The bandwidth test result is FAIL.
0	The bandwidth test result is PASS.

When the bandwidth limit test is set to OFF, 0 is always read out.

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum = Value

Value = SCPI.CALCulate(*Ch*).SELected.BLIMit.MAXimum

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.MAXimum = Value

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.MAXimum

Description

This command sets/gets the upper limit value of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and set/get the upper limit value of the bandwidth test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Maximum bandwidth
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	3E5
Unit	Hz (hertz), dBm or second

- For the variable (*Ch*), the channel number can be selected from 1 to 16, not 1 to 36.

Examples

```
Dim BLimMax As Double
SCPI.CALCulate(1).SELected.BLIMit.MAXimum = 1E9
BLimMax = SCPI.CALCulate(1).SELected.BLIMit.MAXimum
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.MINimum

Equivalent key

Analysis > **Bandwidth Limit** > **Max Bandwidth**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit:MAXimum <numeric>

:CALCulate{[1]-160}[:SELected]:BLIMit:MAXimum?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MAXimum <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MAXimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:MAX 3E5"
20 OUTPUT 717;":CALC1:BLIM:MAX?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum = Value

Value = SCPI.CALCulate(*Ch*).SELected.BLIMit.MINimum

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.MINimum = Value

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.MINimum

Description

This command sets/gets the lower limit value of the bandwidth test, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and set/get the lower limit value of the bandwidth test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Minimum bandwidth
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	1E4
Unit	Hz (hertz), dBm or second

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimMin As Double
SCPI.CALCulate(1).SELected.BLIMit.MINimum = 1E6
BLimMin = SCPI.CALCulate(1).SELected.BLIMit.MINimum
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum

Equivalent key

Analysis > **Bandwidth Limit** > **Min Bandwidth**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit:MINimum <numeric>

:CALCulate{[1]-160}[:SELected]:BLIMit:MINimum?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MINimum <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:MINimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:MIN 1E4"
20 OUTPUT 717;":CALC1:BLIM:MIN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.REPort.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.BLIMit.REPort.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.REPort.DATA

Description

This command reads the bandwidth value of the bandwidth test, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and read the bandwidth test just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	The bandwidth value of the bandwidth
Data type	Double precision floating point type (Double)

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BWData As Double
SCPI.CALCulate(1).PARameter(1).SELect
BWData = SCPI.CALCulate(1).SELected.BLIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:BLIMit:REPort[:DATA]?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit:REPort[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM:REP?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe = Status

Status = SCPI.CALCulate(*Ch*).SELected.BLIMit.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.STATe = Value

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).BLIMit.STATe

Description

This command turns ON/OFF the bandwidth test function, for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the bandwidth test function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF the bandwidth test function.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the bandwidth test function. • False or OFF: Turns OFF the bandwidth test function.
Preset value	False or OFF

For information on the variable (*Ch*), see Variable.

Examples

```
Dim BLimTest As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.BLIMit.STATe = True
BLimTest = SCPI.CALCulate(1).SELected.BLIMit.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.BLIMit.DB

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.MARKer

SCPI.CALCulate(Ch).SELected.BLIMit.DISPlay.VALue

SCPI.CALCulate(Ch).SELected.BLIMit.FAIL

SCPI.CALCulate(Ch).SELected.BLIMit.MAXimum

SCPI.CALCulate(Ch).SELected.BLIMit.MINimum

SCPI.CALCulate(Ch).SELected.BLIMit.REPort.DATA

Equivalent key

Analysis > **Bandwidth Limit** > **BW Test**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:BLIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:BLIMit[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:BLIMit[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:BLIM ON"
20 OUTPUT 717;":CALC1:BLIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion

Object type

Property (**Read-Write**)

Syntax

All Revision

SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.CONVersion.FUNCtion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.FUNCtion = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.FUNCtion

Description

This command sets/gets the parameter after conversion using the parameter conversion function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the parameter after conversion using the parameter conversion function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	The parameter after conversion
Data type	Character string type (String)
Range	Select from the following: • "ZREFlection": Specifies the equivalent impedance in reflection measurement. • "ZTRansmit": Specifies the equivalent impedance(series) in transmission measurement. • "YREFlection": Specifies the equivalent admittance in reflection measurement. • "YTRansmit": Specifies the equivalent admittance(series) in transmission measurement. • "INVersion": Specifies the inverse S-

	parameter. • "ZTSHunt": Specifies the equivalent impedance(shunt) in transmission measurement. • "YTSHunt": Specifies the equivalent admittance(shunt) in transmission measurement. • "CONJugation": Specifies the conjugate.
Preset value	"ZREFlection"

Examples

```
Dim Func As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.CONVersion.FUNCtion = "ztr"
Func = SCPI.CALCulate(1).SELected.CONVersion.FUNCtion
```

Related objects

SCPI.CALCulate(Ch).SELected.CONVersion.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

Equivalent key

Analysis > **Conversion** > **Z:Reflection**|**Z:Transmission**|**Y:Reflection**|**Y:Transmission**|**1/S**| **Z:Trans-Shunt**|**Y:Trans-Shunt**|**Conjugation**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:CONVersion:FUNCtion {ZREFlection|ZTRansmit|YREFlection|YTRansmit|INVersion|ZTSHunt|YTSHunt|CONJugation}

:CALCulate{[1]-160}[:SELected]:CONVersion:FUNCtion?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion:FUNCtion {ZREFlection|ZTRansmit|YREFlection|YTRansmit|INVersion|ZTSHunt|YTSHunt|CONJugation}

:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion:FUNCtion?

Query response

{ZREF|ZTR|YREF|YTR|INV|ZTSH|YTSH|CONJ}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CONV:FUNC ZTR"
20 OUTPUT 717;":CALC1:CONV:FUNC?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.CONVersion.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CONVersion.STATe

Description

This command turns ON/OFF the parameter conversion function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the parameter conversion function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the parameter conversion function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the parameter conversion function. • False or OFF: Turns OFF the parameter conversion function.
Preset value	False or OFF

Examples

```
Dim Conv As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.CONVersion.STATe = True
Conv = SCPI.CALCulate(1).SELected.CONVersion.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Conversion** > **Conversion**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:CONVersion[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:CONVersion[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:CONVersion[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CONV ON"
20 OUTPUT 717;":CALC1:CONV?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.MEDium

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.MEDium = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.MEDium

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.MEDium = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.MEDium

Description

This command select the media type, for calculating the electrical delay time of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the media type for calculating the electrical delay time just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Select the media type for calculating the electrical delay time.
Data type	Character string type (String)
Range	Select from the following: • "COAXial": Selects coaxial as a media type. • "WAVeguide": Selects waveguide as a media type.
Preset value	"COAXial"

Examples

```
Dim EdelMed As String
SCPI.CALCulate(1).SELected.CORRection.EDELay.MEDium = "WAVeguide"
EdelMed = SCPI.CALCulate(1).SELected.CORRection.EDELay.MEDium
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.TIME

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.WGCutoff

Equivalent key

Scale > **Electrical Delay** > **Media**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:CORRection:EDELay:MEDium {COAXial|WAVeguide}

:CALCulate{[1]-160}[:SELected]:CORRection:EDELay:MEDium?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:MEDium {COAXial|WAVeguide}

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:MEDium?

Query response

{COAXial|WAVeguide}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:MED WAV"
20 OUTPUT 717;":CALC1:CORR:EDEL:MED?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.TIME

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.TIME = *Value*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.MEDium

Description

This command sets/gets the electrical delay time of the active trace of channels 1 to 160 (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the electrical delay time for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Electrical delay time
Data type	Double precision floating point type (Double)
Range	-10 to 10
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

	upper limit of the range is exceeded) is set.

Examples

```
Dim Edel As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.CORRection.EDELay.TIME = 0.2
Edel = SCPI.CALCulate(1).SELected.CORRection.EDELay.TIME
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.MEDium

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.WGCutoff

Equivalent key

Scale > **Electrical Delay**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:CORRection:EDELay:TIME <numeric>

:CALCulate{[1]-160}[:SELected]:CORRection:EDELay:TIME?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:TIME <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:TIME?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:TIME 0.2"
20 OUTPUT 717;":CALC1:CORR:EDEL:TIME?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.WGCutoff

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.WGCutoff = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.CORRection.EDELay.WGCutoff

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.WGCutoff = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.EDELay.WGCutoff

Description

This command sets/gets the cut-off frequency when waveguide is selected as an electrical delay time of channels 1 to 160 (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the cut-off frequency just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Cut-off frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

	upper limit of the range is exceeded) is set.

Examples

```
Dim EdelWgc As Double
SCPI.CALCulate(1).SELected.CORRection.EDELay.WGCutoff = 1E9
Edel = SCPI.CALCulate(1).SELected.CORRection.EDELay.WGcutoff
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.MEDium

SCPI.CALCulate(Ch).SELected.CORRection.EDELay.TIME

Equivalent key

Scale > **Electrical Delay** > **Cutoff Frequency**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:CORRection:EDELay:WGCutoff <numeric>

:CALCulate{[1]-160}[:SELected]:CORRection:EDELay:WGCutoff?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:WGCutoff <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:EDELay:WGCutoff?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CORR:EDEL:WGC 1E9"
20 OUTPUT 717;":CALC1:CORR:EDEL:WGC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.CORRection.OFFSet.PHASe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.OFFSet.PHASe = Value

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).CORRection.OFFSet.PHASe

Description

This command sets/gets the phase offset of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the phase offset of the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Phase offset
Data type	Double precision floating point type (Double)
Range	-360 to 360
Preset value	0
Unit	° (degree)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

	upper limit of the range is exceeded) is set.

Examples

```
Dim Offset As Double
SCPI.CALCulate(2).PARameter(1).SELect
SCPI.CALCulate(2).SELected.CORRection.OFFSet.PHASe = 2.5
Offset = SCPI.CALCulate(2).SELected.CORRection.OFFSet.PHASe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Scale > **Phase Offset**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:CORRection:OFFSet:PHASe <numeric>

:CALCulate{[1]-160}[:SELected]:CORRection:OFFSet:PHASe?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:OFFSet:PHASe <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:CORRection:OFFSet:PHASe?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:CORR:OFFS:PHAS 2.5"
20 OUTPUT 717;":CALC1:CORR:OFFS:PHAS?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.DATA.FDATa

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.DATA.FDATa = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.FDATa

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FDATa = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FDATa

Description

This command sets/gets the formatted data array, for the active trace of selected channel (*Ch*).
The array data element varies in the data format (specified with the SCPI.CALCulate(Ch).SELected.FORMat object). For more information on the formatted data array, see Internal Data Processing.

- If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the formatted data array for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (formatted data array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. • *Data(n*2-2) :*Data (primary value) at the n-th measurement point. • *Data(n*2-1) :*Data (secondary value) at the n-th measurement point. Always 0 when the data format is not the Smith chart format or the polar format. The index of the array starts from 0.

Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))*2 when setting a formatted data array, an error occurs when executed and the object is ignored.

Examples

```
Dim FmtData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
SCPI.CALCulate(1).PARameter(1).SELect
FmtData = SCPI.CALCulate(1).SELected.DATA.FDATa
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.DATA.FDATa = FmtData
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.CALCulate(Ch).SELected.DATA.FMEMory

SCPI.CALCulate(Ch).SELected.DATA.SDATa

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:DATA:FDATa <numeric1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}[:SELected]:DATA:FDATa?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FDATa <numeric1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FDATa?

Query response

{numeric 1},… ,{numeric NOP*2}<newline><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:FDAT?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.FMEMory

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FMEMory = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.FMEMory

Description

This command sets/gets the formatted memory array, for the active trace of selected channel (*Ch*).

The array data element varies in the data format (specified with the SCPI.CALCulate(Ch).SELected.FORMat object). For more information on the formatted memory array, see Internal Data Processing.

- If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the formatted memory array for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (formatted memory array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. • *Data(n*2-2) :*Data (primary value) at the n-th measurement point. • *Data(n*2-1) :*Data (secondary value) at the n-th measurement point. Always 0 when the data format is not the Smith chart format or the polar format. The index of the array starts from 0.

Data type	Variant type (Variant)
Note	If there is no array data of NOP (number of measurement point))*2 when setting a formatted memory array, an error occurs when executed and the object is ignored.

Examples

```
Dim FmtMem As Variant
SCPI.SENSe(1).SWEep.POINts = 201
SCPI.CALCulate(1).PARameter(1).SELect
FmtMem = SCPI.CALCulate(1).SELected.DATA.FMEMory
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.DATA.FMEMory = FmtMem
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.CALCulate(Ch).SELected.DATA.FDATa

SCPI.CALCulate(Ch).SELected.DATA.SMEMory

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:DATA:FMEMory <numeric 1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}[:SELected]:DATA:FMEMory?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FMEMory <numeric1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:FMEMory?

Query response

{numeric 1},… ,{numeric NOP*2}<newline><^END>

Example of use

1722

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:FMEM?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.SDATa

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.DATA.SDATa = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.SDATa

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.SDATa = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.SDATa

Description

This command sets/gets the corrected data array, for the active trace of selected channel (*Ch*).

For more information on the corrected data array, see Internal Data Processing

- If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the corrected data array for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. • *Data(n*2-2) :*Real part of the data (complex number) at the n-th measurement point. • *Data(n*2-1)* :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Note	If there is no array data of NOP (number of measurement point))* 2 when setting a corrected data array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
CorData = SCPI.CALCulate(1).SELected.DATA.SDATa
SCPI.SENSe(2).SWEep.POINts = 201
SCPI.CALCulate(2).SELected.DATA.SDATa = CorData
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.DATA.SMEMory

SCPI.CALCulate(Ch).SELected.DATA.FDATa

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:DATA:SDATa <numeric 1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}[:SELected]:DATA:SDATa?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SDATa <numeric1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SDATa?

Query response

<numeric 1>,… ,<numeric NOP*2><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:SDAT?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.DATA.SMEMory

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.SMEMory = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.SMEMory

Description

This command sets/gets the corrected memory array, for the active trace of selected channel (*Ch*).

For more information on the corrected memory array, see Section Internal Data Processing.

- If valid data is not calculated because of the invalid measurement, "1.#QNB" is read out.

In revision A.9.60 and above, you can select the trace and set/get the corrected memory array for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (corrected memory array) of NOP (number of measurement points)x2. Where n is an integer between 1 and NOP. • *Data(nx2-2)* :Real part of the data (complex number) at the n-th measurement point. • *Data(nx2-1)* :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Note	If there is no array data of NOP (number of measurement point))*2 when setting a corrected memory array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorMem As Variant
SCPI.SENSe(1).SWEep.POINts = 201
CorMem = SCPI.CALCulate(1).SELected.DATA.SMEMory
SCPI.SENSe(2).SWEep.POINts = 201
SCPI.CALCulate(1).SELected.DATA.SMEMory = CorMem
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.POINts

SCPI.CALCulate(Ch).SELected.DATA.SDATa

SCPI.CALCulate(Ch).SELected.DATA.FMEMory

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:DATA:SMEMory <numeric 1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}[:SELected]:DATA:SMEMory?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SMEMory <numeric1>,… ,<numeric NOP*2>

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:SMEMory?

Query response

<numeric 1>,… ,<numeric NOP*2><^END>

Example of use

```
10 DIM A(1:201,1:2)
20 OUTPUT 717;":CALC1:DATA:SMEM?"
30 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.DATA.XAXis

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.DATA.XAXis

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).DATA.XAXis

Description

This command reads the data of measurement points of X axis, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the data of measurement points of X axis for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (measurement points) of X axis
Data type	Variant type (Variant)

Examples

```
Dim AnaData As Variant
AnaData = SCPI.CALC1.SEL.DATA.XAX
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:DATA:XAXis?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:DATA:XAXis?

Query response

<numeric 1>,… ,<numeric N><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SEL:DATA:XAX?"
20 ENTER 717;A(*)
```

SCPI.CALCulate(*Ch*).SELected.EQUation.STATE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.EQUation.STATE= *Data*

Data = SCPI.CALCulate(*Ch*).SELected.EQUation.STATE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.STATE= *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.STATE

Description

This command enable/disable the Equation Editor.

In revision A.9.60 and above, you can select the trace and enable/disable the Equation Editor for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	**Ch**
Description	Channel Number
Range	1 to 36

Parameter	*Data*
Description	Sets/Gets the state of equation in the Equation Editor
Data type	Boolean type (Boolean)

Range	Select from either of the following: • True or ON: Turns ON the Equation Editor. • False or OFF: Turns OFF the Equation Editor.
Preset value	False or OFF

Examples

```
Dim EqState As Boolean
EqState = ON
SCPI.CALCulate(1).SELected.EQUation.STATE = EqState
EqState = SCPI.CALCulate(1).SELected.EQUation.TEXT
```

Related objects

SCPI.CALCulate(Ch).SELected.EQUation.TEXT

SCPI.CALCulate(Ch).SELected.EQUation.VALid

Equivalent key

Display > **Equation**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:EQUation:STATE {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:EQUation:STATE?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-160}:EQUation:STATE {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-160}:EQUation:STATE?

Query response

{1/0} <newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SELected:EQUation:STATE ON"
20 OUTPUT 717;":CALC1:SELected:EQUation:STATE?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.EQUation.TEXT

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.EQUation.TEXT = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.EQUation.TEXT

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.TEXT = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.TEXT

Description

This command sets/gets the equation in the Equation Editor. For valid parameters that can be used in this equation, refer to the Equation Editor.

In revision A.9.60 and above, you can select the trace and set/get the equation in the Equation Editor for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	**Ch**
Description	Channel Number
Range	1 to 36

Parameter	*Data*
Description	Sets/Gets the equation in the Equation Editor
Data type	Character string type (String)

Examples

```
Dim strEq As String
```

```
SCPI.CALCulate(1).SELected.EQUation.TEXT = “Example=S21/(1-S11)”
strEq = SCPI.CALCulate(1).SELected.EQUation.TEXT
```

Related objects

SCPI.CALCulate(Ch).SELected.EQUation.STATE

SCPI.CALCulate(Ch).SELected.EQUation.VALid

Equivalent key

No equivalent key is available on the front panel for Equation Text but Equation Editor can be accessed through **Display** > **Equation Editor**.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:EQUation:TEXT <string1>

:CALCulate{[1]-160}[:SELected]:EQUation:TEXT?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:EQUation:TEXT <string1>

:CALCulate{[1]-160}:TRACe{[1]-16}:EQUation:TEXT?

Query response

<string 1><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SELected:EQUation:TEXT?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.EQUation.VALid

Object type

Property (**Read Only**)

Syntax

In revision A.9.5x and below:

Data = SCPI.CALCulate(*Ch*).SELected.EQUation.VALid

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).EQUation.VALid

Description

This command returns False when the equation expression and label are correct but the required S-parameter data is not measured or if it refers the invalid corrected memory array. Annotation of '**Equ!**' is displayed when this command returns a False value.

- Equation Editor can refer S parameter data and data present in corrected memory array.

In revision A.9.60 and above, you can select the trace and check the validity of the equation for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	**Ch**
Description	Channel Number
Range	1 to 36

Parameter	*Data*
Description	A boolean value which gets the state of the equation in the Equation Editor as invalid (False) or valid (True)
Data type	Boolean type (Boolean)

Range	True or ON: Valid Spara data False or OFF: Invalid Spara data
Preset value	False or OFF

Examples

```
Dim EqState As Boolean
EqState = SCPI.CALCulate(1).SELected.EQUation.VALid
```

Related objects

SCPI.CALCulate(Ch).SELected.EQUation.STATE

SCPI.CALCulate(Ch).SELected.EQUation.TEXT

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:EQUation:VALID?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:EQUation:VALID?

Query response

<1/0><^END>

Example of use

```
10 OUTPUT 717;":CALC1:EQUation:VALid?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.CENTer

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.CENTer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.CENTer

Description

This command sets/gets the center value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the center value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The center value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim FilCent As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.CENTer = 1E-8
FilCent = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.CENTer
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.SPAN

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Center**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:CENTer <numeric>

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:CENTer?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:CENTer <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:CENT 1E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME. SHAPe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SHAPe = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SHAPe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SHAPe = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SHAPe

Description

This command set/get the shape of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	The shape of the gate
Data type	Character string type (String)
Range	Select from the following: • "MAXimum": Specifies the maximum shape. • "WIDE": Specifies the wide shape. • "NORMal": Specifies the normal shape. • "MINimum": Specifies the minimum shape.
Preset	"NORMal"

value	

Examples

```
Dim FilShape As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe = "wide"
FilShape = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.TYPE

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Shape** > **Maximum**|**Wide**|**Normal**|**Minimum**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:SHAPe {MAXimum|WIDE|NORMal|MINimum}

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:SHAPe?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SHAPe {MAXimum| WIDE|NORMal|MINimum}

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SHAPe?

Query response

{MAX|WIDE|NORM|MIN}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:SHAP WIDE"
20 OUTPUT 717;":CALC1:FILT:TIME:SHAP?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME. SPAN

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.SPAN

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.SPAN

Description

This command sets/gets the span value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the span value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The span value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	2E-8
Unit	s (second)

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim FilStar As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SPAN = 1E-8
FilStar = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SPAN
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.CENTer

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Span**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:SPAN <numeric>

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:SPAN?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SPAN <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:SPAN 1E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME. STARt

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STARt

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STARt

Description

This command sets/gets the start value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the start value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The start value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	-1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim FilCent As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STARt = 0
FilCent = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STOP

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Start**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:STARt <numeric>

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:STARt?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STARt <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STAR 0"
20 OUTPUT 717;":CALC1:FILT:TIME:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME. STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STATe

Description

This command turns ON/OFF the gating function of the time domain function, for the active trace of selected channel (*Ch*).

You can turn ON the gating function only when the sweep type is the linear sweep and the number of points is 3 or more. If you execute this object to try to turn ON the gating function when the sweep type is other than the linear sweep or the number of points is less than 3, an error occurs and the object is ignored.

When the sweep type is the power sweep, you cannot turn on the gating function. If you execute this object trying to turn on the gating function during the power sweep, an error occurs and the object is ignored.

In revision A.9.60 and above, you can select the trace and turn ON/OFF the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the gating function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the gating function.

	• False or OFF: Turns OFF the gating function.
Preset value	False or OFF

Examples

```
Dim Gating As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STATe = True
Gating = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

Analysis > **Gating** > **Gating**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:STATe?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STAT ON"
20 OUTPUT 717;":CALC1:FILT:TIME:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME. STOP

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.STOP

Description

This command sets/gets the stop value of the gate used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the stop value of the gate used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The stop value of the gate
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim FilStop As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STOP = 2E-8
FilStop = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STARt

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Stop**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:STOP <numeric>

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME:STOP?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STOP <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME:STOP 2E-8"
20 OUTPUT 717;":CALC1:FILT:TIME:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME. TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FILTer.GATE.TIME.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FILTer.GATE.TIME.TYPE

Description

This command sets/gets the gate type used for the gating function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the gate type used for the gating function of the time domain function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	The gate type
Data type	Character string type (String)
Range	Select from the following. • "BPASs": Specifies the band-pass type. • "NOTCh": Specifies the notch type.
Preset value	"BPASs"

Examples

```
Dim FilType As String
SCPI.CALCulate(1).PARameter(1).SELect
```

```
SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe = "notc"
FilType = SCPI.CALCulate(1).SELected.FILTer.GATE.TIME.SHAPe
```

Related objects

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.SHAPe

SCPI.CALCulate(Ch).SELected.FILTer.GATE.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Gating** > **Type**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME[:TYPE] {BPASs|NOTCh}

:CALCulate{[1]-160}[:SELected]:FILTer[:GATE]:TIME[:TYPE]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME[:TYPE] {BPASs|NOTCh}

:CALCulate{[1]-160}:TRACe{[1]-16}:FILTer[:GATE]:TIME[:TYPE]?

Query response

{BPAS|NOTC}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FILT:TIME NOTC"
20 OUTPUT 717;":CALC1:FILT:TIME?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FORMat

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FORMat = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FORMat

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FORMat = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FORMat

Description

This command sets/gets the data format of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the data format of the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Data format
Data type	Character string type (String)
Range	Select from the following: • "MLOGarithmic": Specifies the log magnitude format. • "PHASe": Specifies the phase format. • "GDELay": Specifies the group delay format. • "SLINear": Specifies the Smith chart format (Lin/Phase). • "SLOGarithmic": Specifies the Smith chart format (Log/Phase). • "SCOMplex": Specifies the Smith chart format

	(Re/Im). • "SMITh": Specifies the Smith chart format (R+jX). • "SADMittance": Specifies the Smith chart format (G+jB). • "PLINear": Specifies the polar format (Lin/Phase). • "PLOGarithmic": Specifies the polar format (Log/Phase). • "POLar": Specifies the polar format (Re/Im). • "MLINear": Specifies the linear magnitude format. • "SWR": Specifies the SWR format. • "REAL": Specifies the real format. • "IMAGinary": Specifies the imaginary format. • "UPHase": Specifies the expanded phase format. • "PPHase": Specifies the positive phase format.
Preset value	"MLOGarithmic"

Examples

```
Dim Fmt As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FORMat = "smit"
Fmt = SCPI.CALCulate(1).SELected.FORMat
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Format > **Log Mag**|**Phase**|**Group Delay**|**Lin Mag**|**SWR**|**Real**|**Imaginary**|**Expand Phase**| **Positive Phase**

Format > **Smith** > **Lin/Phase**|**Log/Phase**|**Real/Imag**|**R+jX**|**G+jB**

Format > **Polor** > **Lin/Phase**|**Log/Phase**|**Real/Imag**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FORMat {MLOGarithmic|PHASe|GDELay|SLINear|SLOGarithmic|SCOMplex|SMITh|SADMittance|PLINear|PLOGarithmic|POLar|MLINear|SWR|REAL| IMAGinary|UPHase|PPHase}

:CALCulate{[1]-160}[:SELected]:FORMat?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FORMat {MLOGarithmic|PHASe|GDELay|SLINear|SLOGarithmic|SCOMplex|SMITh|SADMittance|PLINear|PLOGarithmic|POLar|MLINear|SWR|REAL| IMAGinary|UPHase|PPHase}

:CALCulate{[1]-160}:TRACe{[1]-16}:FORMat?

Query response

{MLOG|PHAS|GDEL|SLIN|SLOG|SCOM|SMIT|SADM|PLIN|PLOG|POL|MLIN|SWR| REAL|IMAG|UPH|PPH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FORM SLIN"
20 OUTPUT 717;":CALC1:FORM?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DATA

Description

This command reads the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (analysis result) of N (number of data pairs)x2. N (number of data pairs) can be read out with the SCPI.CALCulate(Ch).SELected.FUNCtion.POINts object. Where n is an integer between 1 and N. • *Data(n*2-2)* :Response value or analysis result of the searched n-th measurement point. • *Data(n*2-1)* :Stimulus value of the searched n-th measurement point. Always 0 for the analysis of the mean value, the standard deviation, and the difference between the maximum value and the minimum value. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim AnaData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "mean"
SCPI.CALCulate(1).SELected.FUNCtion.EXECute
AnaData = SCPI.CALCulate(1).SELected.FUNCtion.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

SCPI.CALCulate(Ch).SELected.FUNCtion.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:FUNCtion:DATA?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DATA?

Query response

<numeric 1>,… ,<numeric N*2><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:POIN?"
20 ENTER 717;A
30 REDIM B(1:2*A)
40 OUTPUT 717;":CALC1:FUNC:DATA?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.COUPle

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.COUPle

Description

This command specifies whether to set the coupling of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for all traces, for the selected channel (*Ch*).

In revision A.9.60 and above, even though you can select the trace and set the coupling of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for all traces just by executing the above TRACe(Tr) command, the *Tr* value is omitted. This is because the command is applied to all the traces.

Variable

Parameter	*Status*
Description	On/off of the trace coupling of the analysis range.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON:Specifies the analysis range with the trace coupling. • False or OFF:Specifies the analysis range for each trace.
Preset	True or ON

value	

Examples

```
Dim TrCpl As Boolean
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.COUPle = False
TrCpl = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.COUPle
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:COUPle {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:COUPle?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:COUPle {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:COUPle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:COUP OFF"
20 OUTPUT 717;":CALC1:FUNC:DOM:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STARt

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.STARt

Description

This command sets/gets the start value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the selected channel (*Ch*).

When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the start value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace (when trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Start value of the analysis range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim AnaStar As Double
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 1.5E9
AnaStar = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:STARt <numeric>

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:STARt?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:STARt <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:STAR 1.7E9"
20 OUTPUT 717;":CALC1:FUNC:DOM:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.STATe

Description

This command sets/gets whether to use an arbitrary range when executing the analysis with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the selected channel (*Ch*).

When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get whether to use an arbitrary range when executing the analysis with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace (when trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	Selection of the analysis range
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies an arbitrary range • False or OFF: Specifies the entire sweep range

Preset value	False or OFF

Examples

```
Dim AnaRnge As Boolean
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STARt = 1.5E9
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 1.8E9
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe = True
AnaRnge = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.DOMain.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.DOMain.STOP

Description

This command sets/gets the stop value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the selected channel (*Ch*).

When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the stop value of the analysis range of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace (when trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Stop value of the analysis range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim AnaStop As Double
SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP = 1.8E9
AnaStop = SCPI.CALCulate(1).SELected.FUNCtion.DOMain.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.COUPle

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:STOP <numeric>

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:STOP?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:STOP <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM:STOP 1.8E9"
20 OUTPUT 717;":CALC1:FUNC:DOM:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.EXECute

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.EXECute

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.EXECute

Description

This command executes the analysis specified with the SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE object, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and execute the analysis specified with the SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Examples

SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.EXECute

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.DOMain.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:EXECute

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:EXECute

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:EXEC"
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.PEXCursion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value (the minimum value of the difference relative to the right and left adjacent measurement points) when executing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the lower limit of peak excursion value when executing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Lower limit of peak excursion value
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3

Unit	Varies depending on the data format. • Log magnitude (MLOG) : dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) :° (degree) • Group delay (GDEL) : s (second) • Others : No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion = 1.5
PeakExc = SCPI.CALCulate(1).SELected.FUNCtion.PEXCursion
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:PEXCursion <numeric>

:CALCulate{[1]-160}[:SELected]:FUNCtion:PEXCursion?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:PEXCursion <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:PEXCursion?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:PEXC 0.2"
20 OUTPUT 717;":CALC1:FUNC:PEXC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.POINts

Object type

Property (**Read Only**)

Syntax

*I*n revision A.9.5x and below:

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.POINts

In revision A.9.60 and above, the following syntax is also supported:

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.POINts

Description

This command reads the number of data pairs of the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of selected channel (*Ch*).

For the analysis of the mean value or the search of the maximum value, 1 is always read out; for the search of all peaks or the search of all targets, the total number of searched measurement points is read out.

In revision A.9.60 and above, you can select the trace and read the number of data pairs of the analysis result of the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Number of analyzed data pairs
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim AnaPoin As Long
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "ape"
SCPI.CALCulate(1).SELected.FUNCtion.EXECute
AnaPoin = SCPI.CALCulate(1).SELected.FUNCtion.POINts
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:FUNCtion:POINts?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCTion:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:POIN?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FUNCtion.PPOLarity

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.PPOLarity

Description

This command sets/gets the polarity when performing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the polarity when performing the peak search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Polarity for peak search
Data type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive peak. • "NEGative": Specifies the negative peak. • "BOTH": Specifies both the positive peak and the negative peak.
Preset	"POSitive"

value	

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity = "both"
PeakPol = SCPI.CALCulate(1).SELected.FUNCtion.PPOLarity
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-160}[:SELected]:FUNCtion:PPOLarity?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:PPOLarity?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:PPOL BOTH"
20 OUTPUT 717;":CALC1:FUNC:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TARGet

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.TARGet

Description

This command sets/gets the target value when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the target value when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Target value
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0

Unit	Varies depending on the data format. • Log magnitude (MLOG) : dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH) : ° (degree) • Group delay (GDEL) : s (second) • Others : No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "atar"
SCPI.CALCulate(1).SELected.FUNCtion.TARGet = -12.5
TargVal = SCPI.CALCulate(1).SELected.FUNCtion.TARGet
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:TARGet <numeric>

:CALCulate{[1]-160}[:SELected]:FUNCtion:TARGet?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:TARGet <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:TARGet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TARG -12.5"
20 OUTPUT 717;":CALC1:FUNC:TARG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TTRansition

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.TTRansition

Description

This command sets/gets the transition type when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object, for the active trace of the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the transition type when performing the target search with the SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute object for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Transition type for search
Data type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.
Preset	"BOTH"

value	

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "atar"
SCPI.CALCulate(1).SELected.FUNCtion.TTRansition = "pos"
TargTran = SCPI.CALCulate(1).SELected.FUNCtion.TTRansition
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:TTRansition {POSitive|NEGative|BOTH}

:CALCulate{[1]-160}[:SELected]:FUNCtion:TTRansition?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:TTRansition {POSitive|NEGative|BOTH}

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:TTRansition?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TTR NEG"
20 OUTPUT 717;":CALC1:FUNC:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.FUNCtion.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).FUNCtion.TYPE

Description

This command sets/gets the type of analysis, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the type of analysis for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Analysis type
Data type	Character string type (String)
Range	Select from the following: • "PTPeak": Specifies the analysis of the difference between the maximum value and the minimum value (Peak to Peak). • "STDEV": Specifies the analysis of the standard deviation. • "MEAN": Specifies the analysis of the mean value. • "MAXimum": Specifies the search for the maximum value. • "MINimum": Specifies the search for the

	minimum value. • "PEAK": Specifies the search for the peak. • "APEak": Specifies the search for all peaks. • "ATARget": Specifies the search for all targets.
Preset value	"PTPeak"

Examples

```
Dim AnaType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.FUNCtion.TYPE = "atar"
AnaType = SCPI.CALCulate(1).SELected.FUNCtion.TYPE
```

Related objects

SCPI.CALCulate(Ch).SELected.FUNCtion.DATA

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.FUNCtion.PEXCursion

SCPI.CALCulate(Ch).SELected.FUNCtion.PPOLarity

SCPI.CALCulate(Ch).SELected.FUNCtion.TARGet

SCPI.CALCulate(Ch).SELected.FUNCtion.TTRansition

SCPI.CALCulate(Ch).SELected.FUNCtion.EXECute

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:FUNCtion:TYPE {PTPeak| STDEV|MEAN| MAXimum|MINimum|PEAK|APEak|ATARget}

:CALCulate{[1]-160}[:SELected]:FUNCtion:TYPE?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtionTYPE {PTPeak| STDEV|MEAN| MAXimum|MINimum|PEAK|APEak|ATARget}

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:TYPE?

Query response

{PTP|STDEV|MEAN|MAX|MIN|PEAK|APE|ATAR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:TYPE PEAK"
20 OUTPUT 717;":CALC1:FUNC:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.LIMit.DATA

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.LIMit.DATA = *Data*

Data = SCPI.CALCulate(*Ch*).SELected.LIMit.DATA

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.DATA = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.DATA

Description

This command sets/gets the limit table for the limit test, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the limit table for the limit test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (for limit line) of 1 + Num (number of limit lines)*5. Where n is an integer between 1 and Num. • *Data(0)* :The number of limit lines you want to set. Specify an integer ranging 0 to 100. When the number of limit lines is set to 0 (clears the limit table), the variable Data is only required with *Data(0)*. • *Data(n*5-4)* :The type of the n-th line. Specify an integer 0 to 2 as follows. 0: OFF 1: Upper limit line 2: Lower limit line • *Data(n*5-3)* :The value on the horizontal axis (frequency/power/time) of the start point of the

	n-th line. • *Data(n*5-2)* :The value on the horizontal axis (frequency/power/time) of the end point of the n-th line. • *Data(n*5-1)* :The value on the vertical axis of the start point of the n-th line. • *Data(n*5)* :The value on the vertical axis of the end point of the n-th line. The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If there is no array data of 1+Num (number of set lines))*5 when setting a formatted memory array, an error occurs when executed and the object is ignored. For *Data(n*5-4)* in the array data, if you specify an integer other than 0, 1 or 2, an error occurs when executed. For *Data(n*5-3)*, *Data(n*5-2)*, *Data(n*5-1)*, and *Data(n85)* in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim LimData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = Array(1,1,1e6,1e9,0,0)
LimData = SCPI.CALCulate(1).SELected.LIMit.DATA

SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = Array(0) 'Clear Limit Table

Dim LimData(5) As Variant
Dim Ref As Variant
LimData(0) = 1
LimData(1) = 1
LimData(2) = 1e6
LimData(3) = 1e9
LimData(4) = 0
LimData(5) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = LimData
Ref = SCPI.CALCulate(1).SELected.LIMit.DATA
```

```
Dim LimData(0) As Variant
LimData(0) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.DATA = LimData 'Clear Limit Table
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe

SCPI.CALCulate(Ch).SELected.LIMit.FAIL

Equivalent key

Analysis > **Limit Test** > **Edit Limit Line**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:DATA <numeric 1>, ... ,<numeric 1+(N*5)>

:CALCulate{[1]-160}[:SELected]:LIMit:DATA?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DATA <numeric 1>, ... ,<numeric 1+(N*5)>

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DATA?

Query response

{numeric 1}, ... ,{numeric 1+(N*5)}<newline><^END>

Example of use

```
10 DIM B(1:2,1:5)
20 OUTPUT 717;":CALC1:LIM:DATA 2,1,1E9,3E9,0,0,2,1E9,3E9,-3,-3"
30 OUTPUT 717;":CALC1:LIM:DATA?"
40 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":CALC1:LIM:DATA 0" ! Clear Limit Table
```

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.DISPlay.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.DISPlay.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.DISPlay.STATe

Description

This command turns ON/OFF the limit line display, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the limit line display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	Limit line display
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the limit line display. • False or OFF: Turns OFF the limit line display.
Preset value	False or OFF

Examples

```
Dim LimDisp As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit = True
LimDisp = SCPI.CALCulate(1).SELected.LIMit.DISPlay.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

Analysis > **Limit Test** > **Limit Line**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:DISPlay[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:LIMit:DISPlay[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DISPlay[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:DISPlay[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:DISP ON"
20 OUTPUT 717;":CALC1:LIM:DISP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

Object type

Property (**Read Only**)

Syntax

All revision:

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.FAIL

In revision A.9.60 and above, the following syntax is also supported:

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.FAIL

Description

This command reads the limit test result, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the limit test result for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	Limit test result
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: The limit test result is FAIL. • False or OFF: The limit test result is PASS.
Note	When the limit test is set to OFF, False or OFF is always read out.

Examples

```
Dim Result As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
Result = SCPI.CALCulate(1).SELected.LIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:LIMit:FAIL?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:FAIL?

Query response

{1|0}<newline><^END>

	Description
1	The limit test result is FAIL.
0	The limit test result is PASS.

When the limit test is set to OFF, 0 is always read out.

Example of use

```
10 OUTPUT 717;":CALC1:LIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.AMPLitude

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude = *Value*

Value = SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.OFFSet.AMPLitude = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.OFFSet.AMPLitude

Description

This command sets/gets the limit line offset of response for the selected active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the selected channel (Ch).

The setting of the limit line doesn't change even if the offset value is changed.

In revision A.9.60 and above, you can select the trace and set/get the limit line offset of response for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The limit line offset of Response (Vertical offset)
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	dB

For information on the variable (*Ch*), see Variable.

Examples

```
Dim LimOffset As Double
SCPI.CALCulate(1).SELected.LIMit.OFFSet.AMPLitude = -10
LimOffset = SCPI.CALCulate(1).SELected.LIMit.OFFSet.AMPLitude
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Amplitude Offset**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:OFFSet:AMPLitude <numeric>

:CALCulate{[1]-160}[:SELected]:LIMit:OFFSet:AMPLitude?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:AMPLitude <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:AMPLitude?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:AMPL -10"
20 OUTPUT 717;":CALC1:LIM:OFFS:AMPL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.MARKer

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.OFFSet.MARKer

Description

This command sets the *active marker* value to amplitude offset using the limit line, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

- The setting of the limit line does not change even if the offset value is changed.

  When the markers are not displayed, this command does not operate.

In revision A.9.60 and above, you can select the trace and set the active marker value to amplitude offset using the limit line for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

For information on the variable (*Ch*), see Variable.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.OFFSet.MARKer
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Marker -> Amplitude Offset**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:OFFSet:MARKer

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:MARKer

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:MARK"
```

SCPI.CALCulate(*Ch*).SELected.LIMit.OFFSet.STIMulus

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus = *Value*

Value = SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.STIMulus

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.OFFSet.STIMulus = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.OFFSet.STIMulus

Description

This command sets/gets the stimulus offset of the limit line, for the selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

- The setting of the limit line doesn't change even if the offset value is changed.

In revision A.9.60 and above, you can select the trace and set/get the stimulus offset of the limit line for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The stimulus offset of the limit line
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz), dBm or second

For information on the variable (*Ch*), see Variable.

Examples

```
Dim LimOffset As Double
SCPI.CALCulate(1).SELected.LIMit.OFFSet.STIMulus = 1E9
LimOffset = SCPI.CALCulate(1).SELected.LIMit.OFFSet.STIMulus
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.AMPLitude

SCPI.CALCulate(Ch).SELected.LIMit.OFFSet.MARKer

Equivalent key

Analysis > **Limit Test** > **Limit Line Offsets** > **Stimulus Offset**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:OFFSet:STIMulus <numeric>

:CALCulate{[1]-160}[:SELected]:LIMit:OFFSet:STIMulus?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:STIMulus <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:OFFSet:STIMulus?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:OFFS:STIM 5E3"
20 OUTPUT 717;":CALC1:LIM:OFFS:STIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.ALL

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(Ch).SELected.LIMit.REPort.ALL

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.REPort.ALL

Description

This command reads the bandwidth test results (stimulus value, limit test result, upper limit value and lower limit value of all measurement points), for the active trace of selected channel (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command).

In revision A.9.60 and above, you can select the trace and read the bandwidth test results for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	Data
Description	Indicates the array data (for limit line) of NOP (number of measurement points)x4. Where n is an integer between 1 and NOP. • *Data(nx4-3)* The stimulus value for the measurement point. • *Data(nx4-2)* The limit test result. Specify an integer -1 to 1 as follows. -1: No limit 0: Fail 1: Pass • *Data(nx4-1)* The upper limit value at the measurement point. (If there is no limit at this point, reads out the 0.) • *Data(nx4)* The lower limit value at the measurement point. (If there is no limit at this point, reads out the 0.)

	The index of the array starts from 0.
Data type	Variant type (Variant)

For information on the variable (*Ch*), see Variable.

Examples

```
Dim LimData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
LimData = SCPI.CALCulate(1).SELected.LIMit.REPort.ALL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.LIMit.REPort.DATA

SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:REPort:ALL?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:REPort:ALL?

Example of use

```
10 OUTPUT 717;":SENS1:SWE:POIN?"
20 ENTER 717;A
30 REDIM B(1:4*A)
40 OUTPUT 717;":CALC1:LIM:REP:ALL?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.REPort.DATA

Description

This command reads the stimulus values (frequency, power level or time) at all the measurement points that failed the limit test, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the stimulus values at all the measurement points that failed the limit test, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data for failed measurement points (can be read out with the SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts object).
Data type	Variant type (Variant)

Examples

```
Dim FailData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
FailData = SCPI.CALCulate(1).SELected.LIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.REPort.POINts

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit:REPort[:DATA]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:REPort[:DATA]?

Query response

{numeric 1},… ,{numeric N}<newline><^END>

Where N is the number of the measurement points that failed (can be read out with the :CALC{1-160}:LIM:REP:POIN? command).

Example of use

```
10 OUTPUT 717;":CALC1:LIM:REP:POIN?"
20 ENTER 717;A
30 REDIM B(1:A)
40 OUTPUT 717;":CALC1:LIM:REP?"
50 ENTER 717;B(*)
```

SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

Object type

Property (**Read Only**)

Syntax

All revision:

Value = SCPI.CALCulate(*Ch*).SELected.LIMit.REPort.POINts

In revision A.9.60 and above, the following syntax is also supported:

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.REPort.POINts

Description

This command reads the number of the measurement points that failed the limit test, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the number of the measurement points that failed the limit test, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Number of measurement points that failed
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim FailPoin As Long
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
FailPoin = SCPI.CALCulate(1).SELected.LIMit.REPort.POINts
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:LIMit:REPort:POINts?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit:REPort:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM:REP:POIN?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.LIMit.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.LIMit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.LIMit.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).LIMit.STATe

Description

This command turns ON/OFF the limit test function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the limit test function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the limit test function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the limit test function. • False or OFF: Turns OFF the limit test function.
Preset value	False or OFF

Examples

```
Dim LimTest As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.LIMit.STATe = True
LimTest = SCPI.CALCulate(1).SELected.LIMit.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.LIMit.DISPlay.STATe

SCPI.DISPlay.FSIGn

Equivalent key

Analysis > **Limit Test** > **Limit Test**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:LIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:LIMit[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:LIMit[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:LIM ON"
20 OUTPUT 717;":CALC1:LIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).ACTivate

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).ACTivate

Description

This command sets the marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the *active marker* , for the active trace of selected channel (*Ch*).

- If you set a marker not displayed to the active marker, the marker display is automatically set to ON.

In revision A.9.60 and above, you can select the trace and set the marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the *active marker*, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Mk*
Description	Marker number
Data type	Long integer type (Long)
Range	1 to 10 Notice that 10 is for the reference marker.
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).ACTivate
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:ACTivate

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:ACTivate

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:ACT"
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).BWIDth.DATA

Description

This command reads the bandwidth search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

If the bandwidth search is impossible, an error occurs when executed and the object is ignored.

In revision A.9.60 and above, you can select the trace and read the bandwidth search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates 4-element array data (bandwidth search result). • *Data(0)* :The bandwidth. • *Data(1)* :Center point frequency of the 2 cutoff frequency points. • *Data(2)* :The Q value. • *Data(3)* :Insertion loss The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim BandData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
BandData = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.THReshold

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:BWIDth:DATA?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:BWIDth:DATA?

Query Response

{numeric1},{numeric2},{numeric3},{numeric4},<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:BWID:DATA?"
20 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.BWIDth.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.BWIDth.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.BWIDth.STATe

Description

This command turns ON/OFF the bandwidth search result display, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the bandwidth search result display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the bandwidth search result display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the bandwidth search result display. • False or OFF: Turns OFF the bandwidth search result display.
Preset value	False or OFF

Examples

```
Dim BandSrch As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.BWIDth.STATe = True
BandSrch = SCPI.CALCulate(1).SELected.MARKer.BWIDth.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.DATA

SCPI.CALCulate(Ch).SELected.MARKer(Mk).BWIDth.THReshold

Equivalent key

Marker Search > **Bandwidth**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:BWIDth[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer:BWIDth[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:BWIDth[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:BWIDth[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:BWID ON"
20 OUTPUT 717;":CALC1:MARK:BWID?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).BWIDth.THReshold

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).BWIDth.THReshold = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).BWIDth.THReshold

Description

This command sets/gets the bandwidth definition value (the value to define the pass-band of the filter) of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the bandwidth definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Bandwidth definition value (the value to define the pass band of the filter)
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	-3

Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim BandVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold = -6
BandVal = SCPI.CALCulate(1).SELected.MARKer(1).BWIDth.THReshold
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.BWIDth.STATe

Equivalent key
Marker Search > Bandwidth Value
Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:BWIDth: THReshold <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:BWIDth: THReshold?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:BWIDth: THReshold <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:BWIDth: THReshold?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:BWID:THR 6"
20 OUTPUT 717;":CALC1:MARK1:BWID:THR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.COUPle

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.COUPle

Description

This command turns ON/OFF the marker coupling between traces, for the selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the marker coupling between traces for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the marker coupling between traces
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the marker coupling. • False or OFF: Turns OFF the marker coupling.
Preset value	True or ON

Examples

```
Dim MkrCpl As Boolean
SCPI.CALCulate(1).SELected.MARKer.COUPle = False
MkrCpl = SCPI.CALCulate(1).SELected.MARKer.COUPle
```

Equivalent key

Marker Fctn > **Couple**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:COUPle {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer:COUPle?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:COUPle {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:COUPle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:COUP OFF"
20 OUTPUT 717;":CALC1:MARK:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).DATA

Description

This command reads the response and stimulus value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

When the reference marker mode is ON ("True" is specified with the SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe object), the readout value is the value relative to the reference marker.

In revision A.9.60 and above, you can select the trace and read the response and stimulus value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (response and stimulus of marker). • *Data(0)* :Response value (primary value) at the marker position. • *Data(1)* :Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format. • *Data(2)* :Stimulus value at the marker position. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim MkrData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
MkrData = SCPI.CALCulate(1).SELected.MARKer(1).DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).X

SCPI.CALCulate(Ch).SELected.MARKer(Mk).Y

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:DATA?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:DATA?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:DATA?"
30 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DISCrete

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DISCrete = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).DISCrete

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).DISCrete = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).DISCrete

Description

This command turns ON/OFF the discrete mode (mode in which the marker moves only at the measurement points) with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the discrete mode with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the marker discrete mode
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the discrete mode. • False or OFF: Turns OFF the discrete mode.
Preset value	False or OFF

Examples

```
Dim MkrDsc As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).DISCrete = True
MkrDsc = SCPI.CALCulate(1).SELected.MARKer(1).DISCrete
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Marker Fctn > **Discrete**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:DISCrete {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:DISCrete?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:DISCrete {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:DISCrete?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:DISC OFF"
20 OUTPUT 717;":CALC1:MARK1:DISC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.COUPle

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.COUPle = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.COUPle

Description

This command sets/gets the coupling of the marker search range for all traces, for the selected channel (*Ch*).

In revision A.9.60 and above, even though you can select the trace and set/get the coupling of the marker search range for all traces just by executing the above TRACe(Tr) command, the *Tr* value is omitted. This is because the command is applied to all the traces.

Variable

Parameter	*Status*
Description	On/off of the trace coupling of the marker search range.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies the search range with the trace coupling. • False or OFF :Specifies the search range for each trace.
Preset value	True or ON

Examples

```
Dim TrCpl As Boolean
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.COUPle = False
TrCpl = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.COUPle
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Couple**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain:COUPle {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain:COUPle?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain:COUPle {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain:COUPle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:COUP OFF"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:COUP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.RANGe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.RANGe = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.RANGe

The following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.DOMain.MULTiple.RANGe = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr)*.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.RANGe

Description

This command sets/gets the target range of search analysis for the selected marker, for the selected channel (*Ch*) in revision A.9.60 and above.

Even though you can select the trace and set/get the target range of search analysis for all traces just by executing the above TRACe(Tr) command, the *Tr* value is omitted. This is because the command is applied to all the traces.

Variable

Parameter	*Param*
Description	Target search range of search analysis.
Data type	Integer
Range	1 to 16

Examples

```
Sub Multiple Range()
```

```
'Turn ON partial search feature and multiple search range
SCPI.CALCulate.TRACe(2).MARKer.FUNCtion.DOMain.STATe = True
SCPI.CALCulate.TRACe(2).MARKer.FUNCtion.DOMain.MULTiple.STATe = True

'Set the start and stop value for search range 3
SCPI.CALCulate.TRACe(2).MARKer(2).FUNCtion.DOMain.MULTiple.STARt(3) = 1000000000#
SCPI.CALCulate.TRACe(2).MARKer(2).FUNCtion.DOMain.MULTiple.STOP(3) = 3000000000#

'Assign the target range to the active marker
SCPI.CALCulate.TRACe(2).MARKer(2).ACTivate
SCPI.CALCulate.TRACe(2).MARKer(2).FUNCtion.DOMain.MULTiple.RANGe = 3

'Move marker to max within the search range
SCPI.CALCulate.TRACe(2).MARKer(2).FUNCtion.TYPE = "MAXimum"
SCPI.CALCulate.TRACe(2).MARKer(2).FUNCtion.EXECute

End Sub
```

Refer to Setting Multiple Range for the operations procedure to place multiple markers within a search range

Related objects

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STARt

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STOP

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STATe

Equivalent key

Marker Search > **Search Range** > **Multiple Range** > **Target Range[1~16]**

Equivalent SCPI command

Syntax

```
:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:RANGe <numeric>
:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:RANGe?
```

The following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:RANGe <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:RANGe?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK2:FUNC:DOM:MULT:RANG 2"
20 OUTPUT 717;":CALC1:MARK2:FUNC:DOM:MULT:RANG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STARt(*Param*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STARt(*Param*) = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STARt(*Param*)

The following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STARt(*Param*) = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STARt(*Param*)

Description

This command sets/gets the start value of the target range in the multiple search range for the selected channel (*Ch*) in revision A.9.60 and above.

When the trace coupling is off, the active trace is the target to be set.

You can select the trace and set/get the start value of the target range in the multiple search range for the trace (when trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Target search range of search analysis.
Data type	Integer
Range	1 to 16

Parameter	*Value*
Description	Start value of the target range in the multiple search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

see the example in SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.RANGe

Related objects

SCPI.CALCULATE(Ch).SELected.MARKer.FUNCtion.DOMain.MULTiple.RANGe

SCPI.CALCULATE(Ch).SELected.MARKer.FUNCtion.DOMain.MULTiple.STATe

SCPI.CALCULATE(Ch).SELected.MARKer.FUNCtion.DOMain.MULTiple.STOP

Equivalent key

Marker Search > **Search Range** > **Multiple Range** > **Start**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STARt{[1]-16} <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STARt{[1]-16}?

The following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STARt{[1]-16} <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STARt{[1]-16}?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STAR(2) 1.7E9"
20 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STAR(2)?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.MULTiple.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.MULTiple.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.MULTiple.STATe

The following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.MULTiple.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.MULTiple.STATe

Description

This command sets/gets the state of the multiple search range for the active trace of the selected channel (*Ch*) in revision A.9.60 and above.

You can select the trace and set/get the multiple search range for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	Selection of the analysis range
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies an arbitrary range • False or OFF: Specifies the entire sweep range
Preset value	False or OFF

Examples

see the example in SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.RANGe

Related objects

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STARt

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.STOP

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.RANGe

Equivalent key

Marker Search > **Search Range** > **Multiple Range** > **Multiple Range**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:MULTiple[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:FUNCtion:DOMain:MULTiple[:STATe]?

The following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:MULTiple[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:FUNCtion:DOMain:MULTiple[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STOP(*Param*)

Object type

Property (**Read-Write**)

Syntax

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STOP(*Param*) = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STOP(*Param*)

The following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STOP(*Param*) = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.DOMain.MULTiple.STOP(*Param*)

Description

This command sets/gets the stop value of the target range in the multiple search range, for the selected channel (*Ch*) in revision A.9.60 and above.

- When the trace coupling is off, the active trace is the target to be set.

You can select the trace and set/get the stop value of the target range in the multiple search range for the trace (when the trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Target search range of search analysis.
Data type	Integer
Range	1 to 16

Parameter	*Value*
Description	Stop value of the target range in the multiple search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

see the example in SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.DOMain.MULTiple.RANGe

Related objects

SCPI.CALCULATE(Ch).SELected.MARKer.FUNCtion.DOMain.MULTiple.RANGe

SCPI.CALCULATE(Ch).SELected.MARKer.FUNCtion.DOMain.MULTiple.STATe

SCPI.CALCULATE(Ch).SELected.MARKer.FUNCtion.DOMain.MULTiple.START

Equivalent key

Marker Search > **Search Range** > **Multiple Range** > **Stop**

Equivalent SCPI command

Syntax

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STOP{[1]-16} <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STOP{[1]-16}?

The following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STOP{[1]-16} <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:DOMain:MULTiple:STOP{[1]-16}?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STOP(2) 1.8E9"
20 OUTPUT 717;":CALC1:MARK1:FUNC:DOM:MULT:STOP(2)?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STARt

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STARt

Description

This command sets/gets the start value of the marker search range, for the selected channel (*Ch*).

- When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the start value of the marker search range for the trace (when the trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The start value of the search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim SchStar As Double
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STARt = 1.7E9
SchStar = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Start**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain:STARt <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain:STARt?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain:STARt <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STAR 1.7E9"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STATe

Description

This command sets/gets whether to use an arbitrary range when executing the marker search, for the selected channel (*Ch*).

- When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get whether to use an arbitrary range when executing the marker search for the trace (when the trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	Selects the search range.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies an arbitrary range. • False or OFF: Specifies the entire sweep range.
Preset value	False or OFF

Examples

```
Dim SchRnge As Boolean
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STARt = 1.5E9
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STOP = 1.8E9
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STATe = True
SchRnge = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STARt

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STOP

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Search Range [ON/OFF]**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM ON"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer.FUNCtion.DOMain.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.FUNCtion.DOMain.STOP

Description

This command sets/gets the stop value of the marker search range, for the selected channel (*Ch*).

- When the trace coupling is off, the active trace is the target to be set.

In revision A.9.60 and above, you can select the trace and set/get the stop value of the marker search range for the trace (when the trace coupling is off) just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Stop value of the search range
Data type	Double precision floating point type (Double)
Preset value	0
Unit	Hz (hertz), dBm or s (second)

Examples

```
Dim SchStop As Double
SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STOP = 1.8E9
SchStop = SCPI.CALCulate(1).SELected.MARKer.FUNCtion.DOMain.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STARt
SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STATe
SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Search Range** > **Stop**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain:STOP <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer:FUNCtion:DOMain:STOP?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain:STOP <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:FUNCtion:DOMain:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STOP 1.8E9"
20 OUTPUT 717;":CALC1:MARK:FUNC:DOM:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.EXECute

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.EXECute

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.EXECute

Description

This command executes search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

To specify the type of the search, use the SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion. TYPE object. (No read)

In revision A.9.60 and above, you can select the trace and execute search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "maximum"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.EXECute
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer.FUNCtion.DOMain.STATe

Equivalent key

Marker Search > **Max**|**Min**

Marker Search > **Peak** > **Search Peak**|**Search Left**|**Search Right**

Marker Search > **Target** > **Search Target**|**Search Left**|**Search Right**

When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:EXECute

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:EXECute

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:EXEC"
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PEXCursion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.PEXCursion = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.PEXCursion

Description

This command sets/gets the lower limit of peak excursion value when executing the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*). Peak excursion value is the minimum value of the difference relative to the right and left adjacent measurement points.

In revision A.9.60 and above, you can select the trace and set/get the lower limit of peak excursion value when executing the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Lower limit of peak excursion value
Data type	Double precision floating point type (Double)
Range	0 to 5E8
Preset value	3

Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): × (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PeakExc As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PEXCursion = 0.2
PeakExc = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PEXCursion
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PPOLarity

Equivalent key

Marker Search > **Peak** > **Peak Excursion**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion: PEXCursion <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion: PEXCursion?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion: PEXCursion <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion: PEXCursion?

Query response

1838

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:PEXC 0.2"
20 OUTPUT 717;":CALC1:MARK1:FUNC:PEXC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.PPOLarity

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.PPOLarity = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.PPOLarity

Description

This comamnd set/get the polarity of the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the polarity of the peak search with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Polarity for peak search
Data type	Character string type (String)
Range	Select from the following: • "POSitive" Specifies the positive peak. • "NEGative" Specifies the negative peak. • "BOTH" Specifies both the positive peak and the negative peak.
Preset	"POSitive"

value	

Examples

```
Dim PeakPol As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "peak"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PPOLarity = "both"
PeakPol = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.PPOLarity
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PEXCursion

Equivalent key

Marker Search > **Peak** > **Peak Polarity**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:PPOLarity?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:PPOLarity {POSitive|NEGative|BOTH}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:PPOLarity?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:PPOL NEG"
20 OUTPUT 717;":CALC1:MARK1:FUNC:PPOL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TARGet

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TARGet = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TARGet

Description

This command sets/gets the target value to be searched with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the target value to be searched with marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Target value for target search
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0

Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TargVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TARGet = -12.5
TargVal = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TARGet
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TTRansition

Equivalent key

Marker Search > **Target** > **Target Value**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TARGet <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TARGet?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TARGet <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TARGet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TARG -12.5"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TARG?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TRACking

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TRACking = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TRACking

Description

This command turns ON/OFF the search tracking (function to repeat search for each sweep) for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the search tracking for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the marker search tracing
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the search tracking. • False or OFF: Turns OFF the search tracking.
Preset value	False or OFF

Examples

```
Dim SrchTrac As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
```

```
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "targ"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TRACking = True
SrchTrac = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TRACking
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Tracking**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TRACking {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TRACking?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TRACking {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TRACking?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TRAC ON"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TRAC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TTRansition

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TTRansition = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TTRansition

Description

This command selects the transition type of the target search, for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and select the transition type of the target search, for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Transition type for search
Data type	Character string type (String)
Range	Select from the following: • "POSitive": Specifies the positive transition. • "NEGative": Specifies the negative transition. • "BOTH": Specifies both the positive transition and the negative transition.

Preset value	"BOTH"

Examples

```
Dim TargTran As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "targ"
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TTRansition = "neg"
TargTran = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TTRansition
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TYPE

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TARGet

Equivalent key

Marker Search > **Target** > **Target Transition**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TTRansition {POSitive| NEGative|BOTH}

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TTRansition?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TTRansition {POSitive| NEGative|BOTH}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TTRansition?

Query response

{POS|NEG|BOTH}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TTR NEG"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TTR?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).FUNCtion.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).FUNCtion.TYPE

Description

This command selects the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and select the search type for marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Search type of marker
Data type	Character string type (String)
Range	Select from the following: • "MAXimum": Sets the search type to the maximum value. • "MINimum": Sets the search type to the minimum value. • "PEAK": Sets the search type to the peak search. • "LPEak": Sets the search type to the peak search to the left from the marker position. • "RPEak": Sets the search type to the peak

	search to the right from the marker position. • "TARGet": Sets the search type to the target search. • "LTARget": Sets the search type to the target search to the left from the marker position. • "RTARget": Sets the search type to the target search to the right from the marker position.
Preset value	"MAXimum"

Examples

```
Dim SrchType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE = "targ"
SrchType = SCPI.CALCulate(1).SELected.MARKer(1).FUNCtion.TYPE
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PEXCursion

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.PPOLarity

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TARGet

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.TTRansition

SCPI.CALCulate(Ch).SELected.MARKer(Mk).FUNCtion.EXECute

Equivalent key

Marker Search > **Max**|**Min**

Marker Search > **Peak** > **Search Peak**|**Search Left**|**Search Right**

Marker Search > **Target** > **Search Target**|**Search Left**|**Search Right**

When performing the operation from the front panel, you select the search type and execute the search at the same time.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion
:TYPE {MAXimum|
MINimum|PEAK|LPEak|RPEak|TARGet|LTARget|RTARget}

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:FUNCtion:TYPE?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion
:TYPE {MAXimum|
MINimum|PEAK|LPEak|RPEak|TARGet|LTARget|RTARget}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:FUNCtion:TYPE?

Query response

{MAX|MIN|PEAK|LPE|RPE|TARG|LTAR|RTAR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:FUNC:TYPE PEAK"
20 OUTPUT 717;":CALC1:MARK1:FUNC:TYPE?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(Mk).NOTCh.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(Mk).NOTCh.DATA

Description

This command reads the notch search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

If the notch search is impossible, an error occurs and the command is ignored. In this case, no query response is obtained.

In revision A.9.60 and above, you can select the trace and read the notch search result of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates 4-element array data (notch bandwidth search result). • *Data(0)* :The bandwidth. • *Data(1)* :Center point frequency of the 2 cutoff frequency points. • *Data(2)* :The Q value. • *Data(3)* :Insertion loss The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim NotchData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
BandData = SCPI.CALCulate(1).SELected.MARKer(1).NOTCH.DATA
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.THReshold

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:NOTCh:DATA?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:NOTCh:DATA?

Query response

{value 1},{value 2},{value 3},{value 4}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1 :NOTC :DATA?"
20 ENTER 717;A,B,C,D
```

SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.NOTCh

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.NOTCh = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.NOTCh

Description

This command turns ON/OFF the notch search result display, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the notch search result display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the notch search result display
Data type	Boolean type (Boolean)
Range	Select from either of the following: • ON or 1: Turns ON the notch search result display. • OFF or 0: Turns OFF the notch search result display.
Preset value	OFF or 0

Examples

```
Dim NotchSrch As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
```

```
SCPI.CALCulate(1).SELected.MARKer.NOTCH = True
NotchSrch = SCPI.CALCulate(1).SELected.MARKer.NOTCH
```

Related Objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.DATA

SCPI.CALCulate(Ch).SELected.MARKer(Mk).NOTCh.THReshold

Equivalent key

Marker Search > **Notch**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:NOTCh[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer:NOTCh[:STATe]?

In revision A.9.60 and above:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:NOTCh[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer:NOTCh[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK :NOTC ON"
20 OUTPUT 717;":CALC1:MARK :NOTC?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).NOTCh.THReshold

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).NOTCh.THReshold = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).NOTCh.THReshold

Description

This command sets/gets the notch definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*), specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command.

In revision A.9.60 and above, you can select the trace and set/get the notch definition value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Notch definition value
Range	-5E8 to 5E8
Preset Value	-3
Unit	Varies depending on the data format as follows: • Amplitude (MLOG):dB (decibel) • Phase (PHAS), Expanded phase (UPH),Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second)

	• Others: No unit

- If the specified parameter is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim NotchVal As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).NOTCH.THReshold = -6
NotchVal = SCPI.CALCulate(1).SELected.MARKer(1).NOTCH.THReshold
```

Related Objects

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh

SCPI.CALCulate(Ch).SELected.MARKer.NOTCh.DATA

Equivalent key

Marker Search > **Notch Value**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:NOTCh:THReshold <value>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:NOTCh:THReshold?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:NOTCh:THReshold <value>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:NOTCh:THReshold?

Query response

{value}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK :NOTC :THR 5"
20 OUTPUT 717;":CALC1:MARK :NOTC :THR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).SET

Object type

Property (**Write Only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).SET = *Param*

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).SET = *Param*

Description

This command sets the value at the position of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the value of the instrument setting item (*Param*), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set the value at the position of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10) to the value of the instrument setting item (*Param*), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Instrument setting item
Data type	Character string type (String)
Range	Select from the following: • "STARt": Sets the sweep start value to the stimulus value at the selected marker position. • "STOP": Sets the sweep stop value to the stimulus value at the selected marker position. • "CENTer": Sets the sweep center value to the stimulus value at the selected marker position. • "RLEVel": Sets the reference line value to the response value at the selected marker position. • "DELay": Sets the electrical delay time value

	to the value of the group delay at the selected marker position (a value smoothed with the aperture of 20%). • "RMARker": Sets the reference marker position to the stimulus value at the marker position. • "SPAN": Sets the span at the between the selected marker and reference marker. If reference marker (*Mk*=10) is selected, span is set at 0.

Examples

```
Dim MkrTo As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).SET = "cent"
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

Equivalent key

Marker Fctn > **Marker -> Start** | **Marker -> Stop** | **Marker -> Center** | **Marker -> Reference** | **Marker -> Delay**

- There are no equivalent keys for the parameters of "RMARker" and "SPAN".

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:SET {STARt|STOP|CENTer|RLEVel|DELay|RMARker|SPAN}

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:SET {STARt|STOP|CENTer|RLEVel|DELay|RMARker|SPAN}

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:SET CENT"
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).STATe

Description

This command turns ON/OFF the display of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the display of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of display of markers 1 to 9 and reference marker
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display of the marker. • False or OFF: Turns OFF the display of the marker.
Preset value	False or OFF

Examples

```
Dim Mkr As Boolean
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.MARKer(10).STATe = True
Mkr = SCPI.CALCulate(1).SELected.MARKer(10).STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

When turning ON the display of the marker

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

When performing the operation from the front panel, a marker set to ON is automatically set to the *active marker* .

When turning OFF the display of the marker

Marker > **Clear Marker Menu** > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**|**Ref Marker**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1 ON"
20 OUTPUT 717;":CALC1:MARK1?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).X

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).X = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).X

Description

This command set the stimulus value for marker 1 to 9 (*Mk*) and reference marker (*Ch*:10), for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set the stimulus value for marker 1 to 9 (*Mk*) and reference marker (*Ch*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Stimulus value of the marker
Data type	Double precision floating point type (Double)
Range	Sweep start value to sweep stop value
Preset value	Sweep start value
Unit	Hz (hertz), dBm or s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

	upper limit of the range is exceeded) is set.

Examples

```
Dim MkrX As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer(1).X = 1E9
MkrX = SCPI.CALCulate(1).SELected.MARKer(1).X
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).Y

SCPI.CALCulate(Ch).SELected.MARKer(Mk).DATA

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

When performing the operation from the front panel, you turn ON the marker and set the stimulus value at the same time.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:X <numeric>

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:X?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:X <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:X?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:X 1E9"
20 OUTPUT 717;":CALC1:MARK1:X?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.MARKer(*Mk*).Y

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer(*Mk*).Y

Description

This command reads the response value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the active trace of selected channel (*Ch*).

When the reference marker mode is ON ("True" is specified with the SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe object), the readout value is the value relative to the reference marker.

In revision A.9.60 and above, you can select the trace and read the response value of marker 1 to 9 (*Mk*) and reference marker (*Mk*:10), for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates 2-element array data (response value of marker). • *Data(0)* :Response value (primary value) at the marker position. • *Data(1)* :Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim MkrY As Variant
SCPI.CALCulate(1).PARameter(1).SELect
MkrY = SCPI.CALCulate(1).SELected.MARKer(1).Y
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MARKer.REFerence.STATe

SCPI.CALCulate(Ch).SELected.MARKer(Mk).X

SCPI.CALCulate(Ch).SELected.MARKer(Mk).DATA

Equivalent key

Marker > **Marker 1**|**Marker 2**|**Marker 3**|**Marker 4**|**Ref Marker**

Marker > **More Markers** > **Marker 5**|**Marker 6**|**Marker 7**|**Marker 8**|**Marker 9**

When performing the operation from the front panel, you turn ON the marker and read the response value on the display.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer{[1]-10}:Y?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MARKer{[1]-10}:Y?

Query response

{numeric 1},{numeric 2}<newline><^END>

	Description
{numeric 1}	Response value (primary value) at the marker position.
{numeric 2}	Response value (secondary value) at the marker position. Always 0 when the data format is not the Smith chart format or the polar format.

Example of use

```
10 OUTPUT 717;":CALC1:MARK1:Y?"
30 ENTER 717;A,B
```

SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MARKer.REFerence.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.REFerence.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MARKer.REFerence.STATe

Description

This command turns ON/OFF the reference marker mode , for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the reference marker mode for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the reference marker mode
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the reference marker mode. • False or OFF: Turns OFF the reference marker mode.
Preset value	False or OFF

Examples

```
Dim RefMode As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MARKer.REFerence.STATe = True
RefMode = SCPI.CALCulate(1).SELected.MARKer.REFerence.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Marker > **Ref Marker Mode**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MARKer:REFerence[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MARKer:REFerence[:STATe]?

In revision A.9.60 and above:

:CALCulate{[1]-160}:TRACe(*Tr*):MARKer:REFerence[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe(*Tr*):MARKer:REFerence[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MARK:REF ON"
20 OUTPUT 717;":CALC1:MARK:REF?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MATH.FUNCtion

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MATH.FUNCtion = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MATH.FUNCtion

Description

This command sets/gets the data trace display method (math method between measurement data and memory trace data), for the active trace of selected channel (*Ch*).

- The math result according to this setting is displayed on the data trace.

In revision A.9.60 and above, you can select the trace and set/get the data trace display method for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Math method between measurement data and memory trace data
Data type	Character string type (String)
Range	Select from the following: • "NORMal": Specifies *Data* (no math). • "DIVide": Specifies *Data* / *Mem*. • "MULTiply": Specifies *Data* x *Mem*. • "SUBTract": Specifies *Data* - *Mem*. • "ADD": Specifies *Data* + *Mem*. Where *Data* is the measurement data (corrected data

	array) and *Mem* is the data stored in the memory trace (corrected memory array).
Preset value	"NORMal"

Examples

```
Dim MathFunc As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MATH.FUNCtion = "div"
MathFunc = SCPI.CALCulate(1).SELected.MATH.FUNCtion
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Display > **Data Math** > **OFF**|**Data / Mem**|**Data * Mem**|**Data - Mem**|**Data + Mem**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MATH:FUNCtion {NORMal|SUBTract|DIVide|ADD|MULTiply}

:CALCulate{[1]-160}[:SELected]:MATH:FUNCtion?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MATH:FUNCtion {NORMal|SUBTract|DIVide|ADD|MULTiply}

:CALCulate{[1]-160}:TRACe{[1]-16}:MATH:FUNCtion?

Query response

{NORM|DIV|MULT|SUBT|ADD}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MATH:FUNC DIV"
20 OUTPUT 717;":CALC1:MATH:FUNC?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MATH.MEMorize

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MATH.MEMorize

Description

This command copies the measurement data to the memory trace, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and copy the measurement data to the memory trace, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Examples

```
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MATH.MEMorize
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Display > **Data** - **> Mem**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MATH:MEMorize

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MATH:MEMorize

Example of use

```
10 OUTPUT 717;":CALC1:MATH:MEM"
```

SCPI.CALCulate(*Ch*).SELected.MIXer.XAXis

Type of object

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MIXer.XAXis = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.MIXer.XAXis

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).MIXer.XAXis = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MIXer.XAXis

Description

This command sets/gets the X-axis frequency when the frequency offset feature is off, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the X-axis frequency when the frequency offset feature is off for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Selects the X-axis frequency (frequency offset: off)
Data type	Character string type (String)
Range	Select from the following: • "NORMal": Specifies the normal frequency. • "RFPLo": Specifies the frequency *RF* + *LO*. • "RFMLo": Specifies the frequency *RF* - *LO*. • "LOMRf": Specifies the frequency *LO* - *RF*.
Preset value	"NORMal"

Example of use

```
Dim Xaxis As String
SCPI.SENSe.OFFSet.STATe = False
```

```
SCPI.CALCulate(1).SELected.MIXer.XAXis = "RFPLo"
Xaxis = SCPI.CALCulate(1).SELected.MIXer.XAXis
```

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.CALCulate(Ch).SELected.OFFset.XAXis

Equivalent key

Sweep Setup > **Frequency Offset** > **X-Axis** > **Normal**|**RF+LO**|**RF-LO**|**LO-RF**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MIXer:XAXis {NORMal|RFPLo|RFMLo|LOMRf}

:CALCulate{[1]-160}[:SELected]:MIXer:XAXis?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MIXer:XAXis {NORMal|RFPLo|RFMLo|LOMRf}

:CALCulate{[1]-160}:TRACe{[1]-16}:MIXer:XAXis?

Query response

{NORM|RFPL|RFML|LOMR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MIX:XAX RFPL"
20 OUTPUT 717;":CALC1:MIX:XAX?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

Object type

Property (**Read Only**)

Syntax

All revision:

Data = SCPI.CALCulate(*Ch*).SELected.MSTatistics.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MSTatistics.DATA

Description

This command reads the statistics values of the active trace of selected channel (*Ch*). The statistical values contains: mean value, standard deviation and the difference between the maximum value and the minimum value.

In revision A.9.60 and above, you can select the trace and read the statistics values of the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (statistics value). • *Data(0)* :Mean value • *Data(1)* :Standard deviation • *Data(2)* :Difference between the maximum value and the minimum value (Peak to Peak) The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim MstData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
MstData = SCPI.CALCulate(1).SELected.MSTatistics.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MSTatistics.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:
:CALCulate{[1]-160}[:SELected]:MSTatistics:DATA?

In revision A.9.60 and above, the following syntax is also supported:
:CALCulate{[1]-160}:TRACe{[1]-16}:MSTatistics:DATA?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

	Description
{numeric 1}	Mean value
{numeric 2}	Standard deviation
{numeric 3}	Difference between the maximum value and the minimum value (Peak to Peak)

Example of use

```
10 OUTPUT 717;":CALC1:MST:DATA?"
20 ENTER 717;A,B,C
```

SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.MSTatistics.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr)*.MSTatistics.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).MSTatistics.STATe

Description

This command turns ON/OFF the statistics values display, for the active trace of selected channel (*Ch*). The statistical values contains: mean value, standard deviation and the difference between the maximum value and the minimum value.

In revision A.9.60 and above, you can select the trace and turn ON/OFF the statistics values display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the statistics value display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the statistics value display. • False or OFF: Turns OFF the statistics value display.

Preset value	False or OFF

Examples

```
Dim Mst As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.MSTatistics.STATe = True
Mst = SCPI.CALCulate(1).SELected.MSTatistics.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.MSTatistics.DATA

Equivalent key

Marker Fctn > **Statistics**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:MSTatistics[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:MSTatistics[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:MSTatistics[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:MSTatistics[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:MST ON"
20 OUTPUT 717;":CALC1:MST?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.OFFset.XAXis

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.OFFSet.XAXis = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.OFFSet.XAXis

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).OFFSet.XAXis = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).OFFSet.XAXis

Description

This command sets/gets the X-axis frequency when the frequency offset feature is on, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the X-axis frequency when the frequency offset feature is on for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	Selects the X-axis frequency (frequency offset: on)
Data type	Character string type (String)
Range	Select from the following: • "BASE": Sets the normal frequency. • "STIMulus": Sets the frequency for the *source port* for the specified measurement parameter. • "RESPonse": Sets the frequency for *the receiver port for the specified measurement parameter*.
Preset value	"RESPonse"

Example of use

```
Dim Xaxis As String
SCPI.SENSe.OFFSet.STATe = True
SCPI.CALCulate(1).SELected.OFFSet.XAXis = "STIMulus"
Xaxis = SCPI.CALCulate(1).SELected.OFFSet.XAXis
```

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.CALCulate(Ch).SELected.MIXer.XAXis

Equivalent key

Sweep Setup > **Frequency Offset** > **X-Axis** > **Base**|**Stimulus**|**Response**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:OFFSet:XAXis {BASE|STIMulus|RESPonse}

:CALCulate{[1]-160}[:SELected]:OFFSet:XAXis?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:OFFSet:XAXis {BASE|STIMulus|RESPonse}

:CALCulate{[1]-160}:TRACe{[1]-16}:OFFSet:XAXis?

Query response

{BASE|STIMulus|RESPonse}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:OFFS:XAX RESP"
20 OUTPUT 717;":CALC1:OFFS:XAX?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DATA

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.RLIMit.DATA = *Data*

Data = SCPI.CALCulate(Ch).SELected.RLIMit.DATA

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DATA = *Data*

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DATA

Description

This command sets/gets the ripple limit table for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command)of selected channel (*Ch*).

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.

In revision A.9.60 and above, you can select the trace and set/get the ripple limit table for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	Indicates the array data (for ripple line) of 1 + Num (number of limit lines)\ 4. Where n is an integer between 1 and Num. • *Data(0)* :The number of limit lines you want to set. Specify an integer ranging 0 to 12. When the number of limit lines is set to 0 (clears the limit table), the variable Data is only required with *Data(0)*. • *Data(nx4-3)* :The type of the n-th line. Specify an integer 0 to 1 as follows. 0: OFF 1: ON • *Data(nx4-2)* :The value on the horizontal axis (frequency/power/time) of the start point of the n-

	th line. • *Data(n×4-1)* :The value on the horizontal axis (frequency/power/time) of the end point of the n-th line. • *Data(n×4)* :The ripple line value (dB) of the n-th line. The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If there is no array data of 1+Num (number of set lines))*4 when setting a formatted memory array, an error occurs when executed and the object is ignored. For *Data(n \ 4-3)* in the array data, if you specify an integer other than 0 or 1, an error occurs when executed. For *Data(n* 4-2)* and *Data(n * 4-1)* in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples (1)

```
Dim RLimData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = Array(1,1,1E6,1E9,0)
RLimData = SCPI.CALCulate(1).SELected.RLIMit.DATA

SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = Array(0) "' Clear Ripple Limit Table
```

Examples (2)

```
Dim RlimData(5) As Variant
Dim Ref As Variant
RLimData(0) = 1
RLimData(1) = 1
RLimData(2) = 1e6
RLimData(3) = 1e9
RLimData(4) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = RLimData
Ref = SCPI.CALCulate(1).SELected.RLIMit.DATA

Dim RLimData(0) as Variant
RLimData(0) = 0
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.DATA = RLimData "' Clear Ripple Limit Table
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

Equivalent key

Analysis > **Ripple Limit** > **Edit Ripple Limit** > **Add**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit:DATA <numeric 1>,… ,<numeric 1+(N\ 4)>

:CALCulate{[1]-160}[:SELected]:RLIMit:DATA?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DATA <numeric 1>,… ,<numeric 1+(N\ 4)>

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DATA?

Query response

{numeric 1},… ,{numeric 1+(Nx4)}<newline><^END>

Example of use

```
10 DIM B(1:2,1:4)
20 OUTPUT 717;":CALC1:RLIM:DATA 2,1,1E9,3E9,3,1,5E9,7E9,3"
30 OUTPUT 717;":CALC1:RLIM:DATA?"
40 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":CALC1:RLIM:DATA 0" ! Clear Ripple Limit Table
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.LINE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE = *Status*

Status = SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DISPlay.LINE = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DISPlay.LINE

Description

This command turns ON/OFF the ripple limit line display, for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the ripple limit line display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF the ripple limit line display.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the ripple limit line display. • False or OFF: Turns OFF the ripple limit line display.
Preset value	False or OFF

Examples

```
Dim RLimDisp As Boolean
SCPI.CALCulate(1).SELected.RLIMit.DISPlay.LINE = True
RLimDisp = SCPI.CALCulate(1).SELected.RLIMit.DISPLay.LINE
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

Equivalent key

Analysis > **Ripple Limit** > **Ripple Limit**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit:DISPlay:LINE {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:RLIMit:DISPlay:LINE?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:LINE {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:LINE?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:LINE ON"
20 OUTPUT 717;":CALC1:RLIM:DISP:LINE?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.SELect

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect = *Value*

Value = SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DISPlay.SELect = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DISPlay.SELect

Description

This command sets/gets the ripple limit band for ripple value display for selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the ripple limit band for ripple value display for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The ripple limit band.
Data type	Long integer type (Long)
Range	1 to 12
Preset value	1

Examples

```
Dim RBand As Long
SCPI.CALCulate(1).SELected.RLIMit.DISPlay.SELect = 2
RBand = SCPI.CALCulate(1).SELected.RLIMit.DISPlay.SELect
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

Equivalent key

Analysis > **Ripple Limit** > **Ripple Band**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit:DISPlay:SELect <numeric>

:CALCulate{[1]-160}[:SELected]:RLIMit:DISPlay:SELect?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:SELect <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:SELect?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:SEL 5"
20 OUTPUT 717;":CALC1:RLIM:DISP:SEL?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.DISPlay.VALue

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue = *Param*

Param = SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DISPlay.VALue = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.DISPlay.VALue

Description

This command sets/gets the display type of ripple value for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command)of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the display type of ripple value for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	The displaying type of ripple value.
Data type	Character string type (String)
Range	Select from the following: • "OFF": Specifies the display off. • "ABSolute": Specifies the absolute value for display type. • "MARgin": Specifies the margin for display type.
Preset	"OFF"

value	

Examples

```
Dim RDisp As String
SCPI.CALCulate(1).SELected.RLIMit.DISPlay.VALue = "ABSolute"
RDisp = SCPI.CALCulate(1).SELected.RLIMit.DISPlay.VALue
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

Equivalent key

Analysis > **Ripple Limit** > **Ripple Value > OFF|Absolute|Margin**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit:DISPlay:VALue {OFF|ABS|MAR}

:CALCulate{[1]-160}[:SELected]:RLIMit:DISPlay:VALue?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:VALue {OFF|ABS|MAR}

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:DISPlay:VALue?

Query response

{OFF|ABS|MAR}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:DISP:VAL ABS"
20 OUTPUT 717;":CALC1:RLIM:DISP:VAL?"
30 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.FAIL

Object type

Property (**Read Only**)

Syntax

All revision:

Status = SCPI.CALCulate(Ch).SELected.RLIMit.FAIL

In revision A.9.60 and above, the following syntax is also supported:

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.FAIL

Description

This command reads the ripple test result for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command)of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and read the ripple test result for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	The ripple test result
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the ripple test result is FAIL. • False or OFF: Turns OFF the ripple test result is FAIL.
Note	When the ripple test if set to OFF, False or OFF is always read out.

Examples

```
Dim Result As Boolean
Result = SCPI.CALCulate(1).SELected.RLIMit.FAIL
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit:FAIL?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:FAIL?

Example of use

```
10 OUTPUT 717;":CALC1:RLIM:FAIL?"
20 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.REPort.DATA

Object type

Property

Syntax

All revision:

Data = SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA

In revision A.9.60 and above, the following syntax is also supported:

Data = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.REPort.DATA

Description

This command reads the ripple value of the ripple test for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command)of selected channel (*Ch*).

The data transfer format when this command is executed depends on the setting with the SCPI.FORMat.DATA command.(Read only)

In revision A.9.60 and above, you can select the trace and read the ripple value of the ripple test for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Data*
Description	Indicates the array data (for ripple line) of 1 + Num (number of limit lines)*3. Where n is an integer between 1 and 12. • *Data(0)* :Number of ripple limit line. • *Data(nx3-2)* :Number of ripple limit bands. • *Data(nx3-1)* :Ripple value. • *Data(nx3)* :Results of ripple test. Select from the following. 0:PASS 1:FAIL. The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim RData As Variant
SCPI.CALCulate(1).PARameter(1).SELect
RData = SCPI.CALCulate(1).SELected.RLIMit.REPort.DATA
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit:REPort[:DATA]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit:REPort[:DATA]?

Query response

{numeric 1},… ,{numeric 1+Nx3}<newline><^END>

Type	Description
<numeric 1>	Number of ripple limit line (1 to 12)
<numeric 1+(n\ 3)-2>	Number of ripple limit bands
<numeric 1+(n\ 3)-1>	Ripple value
<numeric 1+(n\ 3)>	Results of ripple test 0: Pass 1: Fail

Where N is the number of lines (specified with <numeric 1>) and n is an integer between 1 and 12.

Example of use

```
10 DIM B(1:2,1:3)
20 OUTPUT 717;":CALC1:RLIM:REP?"
30 ENTER 717;A,B(*)
```

SCPI.CALCulate(*Ch*).SELected.RLIMit.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(Ch).SELected.RLIMit.STATe = *Status*

Status = SCPI.CALCulate(Ch).SELected.RLIMit.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).RLIMit.STATe

Description

This command turns ON/OFF the ripple test function for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the ripple test function for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF the ripple test function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the ripple test function. • False or OFF: Turns OFF the ripple test function.
Preset	False or OFF

value	

Examples

```
Dim RLimTest As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.RLIMit.STATe = True
RLimTest = SCPI.CALCulate(1).SELected.RLIMit.STATe
```

Related Objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.RLIMit.DATA

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.LINE

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.SELect

SCPI.CALCulate(Ch).SELected.RLIMit.DISPlay.VALue

SCPI.CALCulate(Ch).SELected.RLIMit.FAIL

SCPI.CALCulate(Ch).SELected.RLIMit.REPort.DATA

Equivalent key

Analysis > **Ripple Limit** > **Ripple Limit Test**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:RLIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:RLIMit[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:RLIMit[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:RLIM ON"
20 OUTPUT 717;":CALC1:RLIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.SMOothing.APERture

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.APERture = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.APERture

Description

This command sets/gets the smoothing aperture (percentage to the sweep span value) of the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the smoothing aperture of the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Smoothing aperture
Data type	Double precision floating point type (Double)
Range	0.05 to 25
Preset value	1.5
Unit	% (percent)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the

	upper limit of the range is exceeded) is set.

Examples

```
Dim SmoAper As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.SMOothing.APERture = 2.5
SmoAper = SCPI.CALCulate(1).SELected.SMOothing.APERture
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.SMOothing.STATe

Equivalent key

Avg > **Smo Aperture**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:SMOothing:APERture <numeric>

:CALCulate{[1]-160}[:SELected]:SMOothing:APERture?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing:APERture <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing:APERture?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SMO:APER 2.5"
20 OUTPUT 717;":CALC1:SMO:APER?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.SMOothing.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).SMOothing.STATe

Description

This command turns ON/OFF the smoothing, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and turn ON/OFF the smoothing for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the smoothing
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the smoothing. • False or OFF: Turns OFF the smoothing.
Preset value	False or 0

Examples

```
Dim Smo As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.SMOothing.STATe = True
Smo = SCPI.CALCulate(1).SELected.SMOothing.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.SMOothing.APERture

Equivalent key

Avg > **Smoothing**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:SMOothing[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:SMOothing[:STATe]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing[:STATe] {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:SMOothing[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:SMO:STAT ON"
20 OUTPUT 717;":CALC1:SMO:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CENTer

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.CENTer

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.CENTer = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.CENTer

Description

This command sets/gets the center value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the center value for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Center value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Cent As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.CENTer = 1E-8
Cent = SCPI.CALCulate(1).SELected.TRANsform.TIME.CENTer
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.SPAN

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Center**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:CENTer <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:CENTer?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:CENTer <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:CENT 1E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:CENT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.IMPulse.WIDTh

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.IMPulse.WIDTh = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.IMPulse.WIDTh

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.IMPulse.WIDTh = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.IMPulse.WIDTh

Description

This command sets/gets the shape of the Kayser Bessel window using the impulse width used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the Kayser Bessel window using the impulse width used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Impulse width
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and transformation type.
Preset value	Varies depending on the frequency span and transformation type.
Unit	s (second)

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim ImpWid As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.IMPulse.WIDTh = 1E-10
ImpWid = SCPI.CALCulate(1).SELected.TRANsform.TIME.IMPulse.WIDTh
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME. KBESsel

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STEP.RTIMe

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Center**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:IMPulse:WIDTh <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:IMPulse:WIDTh?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:IMPulse:WIDTh <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:IMPulse:WIDTh?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:IMP:WIDT 1E-10"
20 OUTPUT 717;":CALC1:TRAN:TIME:IMP:WIDT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.KBESsel

Object Type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.KBESsel = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.KBESsel

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.KBESsel = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.KBESsel

Description

This command sets/gets the shape of the Kayser Bessel window used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the Kayser Bessel window used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The value of b
Data type	Double precision floating point type (Double)
Range	0 to13
Preset value	6
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Beta As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.KBESsel = 3
Beta = SCPI.CALCulate(1).SELected.TRANsform.TIME.KBESsel
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME. IMPulse.WIDTh

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STEP.RTIMe

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Center**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:KBESsel <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:KBESsel?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:KBESsel <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:KBESsel?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:KBES 3"
20 OUTPUT 717;":CALC1:TRAN:TIME:KBES?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.LPFRequency

Object type

Method (**Write-only**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.LPFRequency = *Value*

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.LPFRequency = *Value*

Description

This command changes the frequency range to match with the low-pass type transformation of the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and change the frequency range to match with the low-pass type transformation of the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Examples

SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.LPFRequency

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.TYPE

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Set Freq Low pass**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:LPFRequency

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:LPFRequency

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.SPAN

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.SPAN

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.SPAN = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.SPAN

Description

This command sets/gets the span value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the span value for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Span value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	2E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Span As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.SPAN = 1E-8
Cent = SCPI.CALCulate(1).SELected.TRANsform.TIME.SPAN
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.CENTer

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Center**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:SPAN <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:SPAN?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:SPAN <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:SPAN 1E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:SPAN?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STARt

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STARt

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STARt = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STARt

Description

This command sets/gets the start value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the start value for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Start value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	-1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Star As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STARt = 0
Star = SCPI.CALCulate(1).SELected.TRANsform.TIME.STARt
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STOP

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Start**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STARt <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STARt?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STARt <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STAR 0"
20 OUTPUT 717;":CALC1:TRAN:TIME:STAR?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STATe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STATe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STATe = *Status*

Status = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STATe

Description

This command turns ON/OFF the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

You can enable the transformation function only when the sweep type is the linear sweep and the number of points is three or more. If you execute this object to try to enable the transformation function when the sweep type is other than the linear sweep or the number of points is less than three, an error occurs and the object is ignored.

- When the sweep type is the power sweep, you cannot turn on the transformation function. If you execute this object trying to turn on the transformation function during the power sweep, an error occurs and the object is ignored.

In revision A.9.60 and above, you can select the trace and turn ON/OFF the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Status*
Description	ON/OFF of the gating function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the transformation function.

	• False or OFF: Turns OFF the transformation function.
Preset value	False or OFF

Examples

```
Dim Trans As Boolean
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STATe = True
Trans = SCPI.CALCulate(1).SELected.TRANsform.TIME.STATe
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

Analysis > **Transform** > **Transform**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STATe?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STATe {ON|OFF|1|0}

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STAT ON"
20 OUTPUT 717;":CALC1:TRAN:TIME:STAT?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STEP.RTIMe

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STEP.RTIMe = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STEP.RTIMe

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STEP.RTIMe = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STEP.RTIMe

Description

This command sets/gets the shape of the Kayser Bessel window using the rise time of step signal used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the shape of the Kayser Bessel window using the rise time of step signal used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	The rise time of step signal
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span.
Preset value	Varies depending on the frequency span.
Unit	s (second)

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim RTime As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STEP.RTIMe = 1E-10
RTime = SCPI.CALCulate(1).SELected.TRANsform.TIME.STEP.RTIMe
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.IMPulse.WIDTh

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.KBESsel

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Center**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STEP:RTIMe <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STEP:RTIMe?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STEP:RTIMe <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STEP:RTIMe?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STEP:RTIM 1E-10"
20 OUTPUT 717;":CALC1:TRAN:TIME:STEP:RTIM?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STIMulus

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STIMulus = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STIMulus

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STIMulus = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STIMulus

Description

This command sets/gets the stimulus type used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the stimulus type used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	The stimulus type
Data type	Character string type (String)
Range	Select from the following: • "IMPulse": Specifies the impulse • "STEP": Specifies the step
Preset value	"IMPulse"

Examples

```
Dim StimType As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STIMulus = "step"
StimType = SCPI.CALCulate(1).SELected.TRANsform.TIME.STIMulus
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.TYPE

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Type** > **Bandpass**|**Lowpass Step**|**Lowpass Imp.**

- When performing this operation from the front panel, you select the transformation type at the same time.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STIMulus {IMPulse|STEP}

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STIMulus?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STIMulus {IMPulse|STEP}

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STIMulus?

Query response

{IMP|STEP}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME LPAS"
20 OUTPUT 717;":CALC1:TRAN:TIME:STIM STEP"
30 OUTPUT 717;":CALC1:TRAN:TIME:STIM?"
40 ENTER 717;A$
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STOP

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.STOP

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STOP = *Value*

Value = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.STOP

Description

This command sets/gets the stop value used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the stop value used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Value*
Description	Stop value
Data type	Double precision floating point type (Double)
Range	Varies depending on the frequency span and the number of points.
Preset value	1E-8
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range

	is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Span As Double
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.STOP = 2E-8
Cent = SCPI.CALCulate(1).SELected.TRANsform.TIME.STOP
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STARt

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Stop**

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STOP <numeric>

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME:STOP?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STOP <numeric>

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME:STOP 2E-8"
20 OUTPUT 717;":CALC1:TRAN:TIME:STOP?"
30 ENTER 717;A
```

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.TYPE

Object type

Property (**Read-Write**)

Syntax

All revision:

SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).SELected.TRANsform.TIME.TYPE

In revision A.9.60 and above, the following syntax is also supported:

SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.TYPE = *Param*

Param = SCPI.CALCulate(*Ch*).TRACe(*Tr*).TRANsform.TIME.TYPE

Description

This command sets/gets the transformation type used for the transformation function of the time domain function, for the active trace of selected channel (*Ch*).

In revision A.9.60 and above, you can select the trace and set/get the transformation type used for the transformation function of the time domain function, for the trace just by executing the above TRACe(Tr) command. You do not need to execute SCPI.CALCulate(Ch).PARameter(Tr).SELect.

Variable

Parameter	*Param*
Description	The transformation type
Data type	Character string type (String)
Range	Select from the following: • "BPASs": Specifies the band-pass • "LPASs": Specifies the low-pass
Preset value	"BPASs"

Examples

```
Dim Typ As String
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.CALCulate(1).SELected.TRANsform.TIME.SHAPe = "lpas"
Typ = SCPI.CALCulate(1).SELected.TRANsform.TIME.SHAPe
```

Related objects

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STIMulus

SCPI.CALCulate(Ch).SELected.TRANsform.TIME.STATe

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Analysis > **Transform** > **Type** > **Bandpass**|**Lowpass Step**|**Lowpass Imp.**

When performing this operation from the front panel, you select the stimulus type at the same time.

Equivalent SCPI command

Syntax

All revision:

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME[:TYPE] {BPASs|LPASs}

:CALCulate{[1]-160}[:SELected]:TRANsform:TIME[:TYPE]?

In revision A.9.60 and above, the following syntax is also supported:

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME[:TYPE] {BPASs|LPASs}

:CALCulate{[1]-160}:TRACe{[1]-16}:TRANsform:TIME[:TYPE]?

Query response

{BPAS|LPAS}<newline><^END>

Example of use

```
10 OUTPUT 717;":CALC1:TRAN:TIME LPAS"
20 OUTPUT 717;":CALC1:TRAN:TIME?"
30 ENTER 717;A$
```

CONTROL

SCPI.CONTrol.HANDler.A.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.CONTrol.HANDler.A.DATA = *Value*

Description

This command sets/gets information of output port A (A0 to A7) of the handler I/O. Port information is output as 8-bit binary data using A0 as LSB and A7 as MSB.

Variable

Parameter	*Value*
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
SCPI.CONTrol.HANDler.A.DATA = 15
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CONTrol:HANDler:A[:DATA] <numeric>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:A 15"
20 OUTPUT 717;":CONT:HAND:A:DATA 15"
```

SCPI.CONTrol.HANDler.B.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.CONTrol.HANDler.B.DATA = *Value*

Description

This command sets/gets information of output port B (B0 to B7) of the handler I/O. Port information is output as 8-bit binary data using B0 as LSB and B7 as MSB.

- The bit 6 of the data outputted by this project is ignored when outputting the INDEX signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.INDex.STATe object).
- The bit 7 of the data outputted by this project is ignored when outputting the READY FOR TRIGGER signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe object).

Variable

Parameter	*Value*
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
SCPI.CONTrol.HANDler.B.DATA = 15
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CONTrol:HANDler:B[:DATA] <numeric>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:B 15"
20 OUTPUT 717;":CONT:HAND:B:DATA 15"
```

SCPI.CONTrol.HANDler.C.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.C.DATA = *Value* (for output port)

Value = SCPI.CONTrol.HANDler.C.DATA (for input port)

Description

When input/output port C of the handler I/O is set to the output port, outputs port information to output port C (C0 to C3).

When input/output port C of the handler I/O is set to the input port, reads out port information inputted to port C (C0 to C3).

Port information is input/output as 4-bit binary data, using C0 as LSB and C3 as MSB.

Variable

Parameter	*Value*
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 15
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
SCPI.CONTrol.HANDler.C.MODE = "outp"
SCPI.CONTrol.HANDler.C.DATA = 8
```

```
Dim HdlCinp As Long
SCPI.CONTrol.HANDler.C.MODE = "inp"
HdlCinp = SCPI.CONTrol.HANDler.C.DATA
```

Related objects

SCPI.CONTrol.HANDler.C.MODE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:C[:DATA] <numeric>

:CONTrol:HANDler:C[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:C 15"
```

```
10 OUTPUT 717;":CONT:HAND:C:MODE INP"
20 OUTPUT 717;":CONT:HAND:C?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.C.MODE

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.C.MODE = *Param*

Param = SCPI.CONTrol.HANDler.C.MODE

Description

This command sets/gets the input/output direction of port C of the handler I/O.

Variable

Parameter	*Param*
Description	Input/output direction of port C
Data type	Character string type (String)
Range	Select from the following: "INPut":Sets the port C to input. "OUTPut":Sets the port C to output.
Preset value	"INPut"

Examples

```
Dim HdlCmode As String
SCPI.CONTrol.HANDler.C.MODE = "OUTP"
HdlCmode = SCPI.CONTrol.HANDler.C.MODE
```

Related objects

SCPI.CONTrol.HANDler.C.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:C:MODE {INPut|OUTPut}

:CONTrol:HANDler:C:MODE?

Query response

{INP|OUTP}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:C:MODE?"
30 ENTER 717;A$
```

SCPI.CONTrol.HANDler.D.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.D.DATA = *Value* (for output port)

Value = SCPI.CONTrol.HANDler.D.DATA (for input port)

Description

When input/output port D of the handler I/O is set to the output port, outputs port information to output port D (D0 to D3).

When input/output port D of the handler I/O is set to the input port, reads out port information inputted to port D (D0 to D3).

Port information is output as 4-bit binary data using D0 as LSB and D3 as MSB.

Variable

Parameter	*Value*
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 15
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
SCPI.CONTrol.HANDler.D.MODE = "outp"
SCPI.CONTrol.HANDler.D.DATA = 8
```

```
Dim HdlDinp As Long
SCPI.CONTrol.HANDler.D.MODE = "inp"
HdlDinp = SCPI.CONTrol.HANDler.D.DATA
```

Related objects

SCPI.CONTrol.HANDler.D.MODE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:D[:DATA] <numeric>

:CONTrol:HANDler:D[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:D:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:D 15"
```

```
10 OUTPUT 717;":CONT:HAND:D:MODE INP"
20 OUTPUT 717;":CONT:HAND:D?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.D.MODE

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.D.MODE = *Param*

Param = SCPI.CONTrol.HANDler.D.MODE

Description

This command sets/gets the input/output direction of port D of the handler I/O.

Variable

Parameter	*Param*
Description	Input/output direction of port D
Data type	Character string type (String)
Range	Select from the following: "INPut": Sets the port D to input. "OUTPut":Sets the port D to output.
Preset value	"INPut"

Examples

```
Dim HdlDmode As String
SCPI.CONTrol.HANDler.D.MODE = "OUTP"
HdlDmode = SCPI.CONTrol.HANDler.D.MODE
```

Related objects

SCPI.CONTrol.HANDler.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:C:MODE {INPut|OUTPut}

:CONTrol:HANDler:C:MODE?

Query response

{INP|OUTP}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:D:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:D:MODE?"
30 ENTER 717;A$
```

SCPI.CONTrol.HANDler.E.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.E.DATA = *Value* (for output)

Value = SCPI.CONTrol.HANDler.E.DATA (for input port)

Description

When input/output port E (port C + port D) of the handler I/O is set to the output port, outputs port information to output port E (C0 to D3).

When input/output port E of the handler I/O is set to the input port, reads out port information inputted to port E (C0 to D3).

Port information is output as 8-bit binary data using C0 as LSB and D3 as MSB.

Variable

Parameter	*Value*
Description	Port information (output/input)
Data type	Long integer type (Long)
Range	0 to 255
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
SCPI.CONTrol.HANDler.C.MODE = "outp"
SCPI.CONTrol.HANDler.D.MODE = "outp"
SCPI.CONTrol.HANDler.E.DATA = 128
```

```
Dim HdlEinp As Long
SCPI.CONTrol.HANDler.C.MODE = "inp"
SCPI.CONTrol.HANDler.D.MODE = "inp"
HdlEinp = SCPI.CONTrol.HANDler.E.DATA
```

Related objects

SCPI.CONTrol.HANDler.C.MODE

SCPI.CONTrol.HANDler.D.MODE

SCPI.CONTrol.HANDler.C.DATA

SCPI.CONTrol.HANDler.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:E[:DATA] <numeric>

:CONTrol:HANDler:E[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:C:MODE OUTP"
20 OUTPUT 717;":CONT:HAND:D:MODE OUTP"
30 OUTPUT 717;":CONT:HAND:E 128"
```

```
10 OUTPUT 717;":CONT:HAND:C:MODE INP"
20 OUTPUT 717;":CONT:HAND:D:MODE INP"
30 OUTPUT 717;":CONT:HAND:E?"
40 ENTER 717;A
```

SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.EXTension.INDex.STATe = *Status*

Status = SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Description

Turns ON/OFF output of the INDEX signal to B6 of the handler I/O.

- When you use port B6 as the output port, turn OFF the INDEX signal output. When outputting the INDEX signal is turned ON, the bit 6 of the data output by the SCPI.CONTrol.HANDler.B.DATA object (the bit 14 of the data outputted by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the INDEX signal output
Data type	Boolean type (Boolean)
Range	Select from the following: True or ON:Turns ON the INDEX signal output. False or OFF: Turns OFF the INDEX signal output.
Preset value	False or OFF

Examples

```
Dim Indx As Boolean
SCPI.CONTrol.HANDler.EXTension.INDex.STATe = 1
Indx = SCPI.CONTrol.HANDler.EXTension.INDex.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler[:EXTension]:INDex:STATe {ON|OFF|1|0}

:CONTrol:HANDler[:EXTension]:INDex:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:IND:STAT ON"
20 OUTPUT 717;":CONT:HAND:IND:STAT?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = *Status*

Status = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe

Description

This command turns ON/OFF the output of READY FOR TRIGGER signal to B7 of the handler I/O.

- When you use port B7 as the output port, turn OFF the READY FOR TRIGGER signal output. When outputting the READY FOR TRIGGER signal is turned ON, the bit 7of the data output by the SCPI.CONTrol.HANDler.B.DATA object (the bit 15 of the data output by the SCPI.CONTrol.HANDler.F.DATA object) is ignored.

Variable

Parameter	*Status*
Description	ON/OFF of the READY FOR TRIGGER signal output
Data type	Boolean type (Boolean)
Range	Select from the following: True or ON:Turns ON the READY FOR TRIGGER signal output. False or OFF:Turns OFF the READY FOR TRIGGER signal output.
Preset value	False or OFF

Examples

```
Dim RdyTrig As Boolean
SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe = 0
RdyTrig = SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe
```

Related objects

SCPI.CONTrol.HANDler.EXTension.INDex.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler[:EXTension]:RTRigger:STATe {ON|OFF|1|0}

:CONTrol:HANDler[:EXTension]:RTRigger:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:HAND:RTR:STAT ON"
20 OUTPUT 717;":CONT:HAND:RTR:STAT?"
30 ENTER 717;A
```

SCPI.CONTrol.HANDler.F.DATA

Object type

Method (**Write Only**)

Syntax

SCPI.CONTrol.HANDler.F.DATA = *Value*

Description

Outputs port information to output port F (port A + port B) of the handler I/O. Port information is output as 16-bit binary using A0 as LSB and B7 as MSB.

- The bit 14 of the data output by this project is ignored when outputting the INDEX signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.INDex.STATe object).
- The bit 15 of the data output by this project is ignored when outputting the READY FOR TRIGGER signal is turned ON (specifying True with the SCPI.CONTrol.HANDler.EXTension.RTRigger.STATe object).

Variable

Parameter	*Value*
Description	Port information (output)
Data type	Long integer type (Long)
Range	0 to 65535
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
SCPI.CONTrol.HANDler.F.DATA = 511
```

Related objects

SCPI.CONTrol.HANDler.A.DATA

SCPI.CONTrol.HANDler.B.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:HANDler:F[:DATA] <numeric>

Example of use

```
10 OUTPUT 717;":CONT:HAND:F 511"
```

SCPI.CONTrol.HANDler.OUTPut(*Num*).DAT-

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.HANDler.OUTPut(*Num*) = *Value*

Value = SCPI.CONTrol.HANDler.OUTPut(*Num*)

Description

This command sets/gets data to OUTPUT1 or OUTPUT2 of the handler I/O.

Variable

Parameter	*Num*
Description	Number of the OUTPUT terminal
Data type	Long integer type (Long)
Range	1 or 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	Polarity (High/Low)
Data type	Long integer type (Long)
Range	Select from the following: • 1: Specifies

	LOW. • 0: Specifies HIGH.

Examples

```
Dim HdlPol As Long
SCPI.CONTrol.HANDler.OUTPut(1).DATA = 1
HdlPol = SCPI.CONTrol.HANDler.OUTPut(1).DATA
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:CONTrol:HANDler:OUTPut{[1]|2}[:DATA] {1|0}
:CONTrol:HANDler:OUTPut{[1]|2}[:DATA]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":CONT:HAND:OUTP1 1"
20 OUTPUT 717;":CONT:HAND:OUTP1?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.DATA

Description

This command sets/gets output port data for group A with ID of the multiport test set .

Variable

Parameter	*Value*
Description	Output port data for group A
Data type	Long integer type (Long)
Range	0 to 255
Note	This command is available only for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.A.DATA =150
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.A.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:A 0"
20 OUTPUT 717;":CONT:MULT1:OUTP:A?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA

Description

This command sets/gets the output voltage of the group A with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output voltage for group A
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.A.VOLTage.DATA = 5
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.A.VOLTage.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A:VOLTage[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:A:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:A:VOLT 2.3"
20 OUTPUT 717;":CONT:MULT1:OUTP:A:VOLT?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.DATA

Description

This command sets/gets the output port data for group B with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output port data for group B
Data type	Long integer type (Long)
Range	0 to 15
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.B.DATA =12
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.B.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:B 10"
20 OUTPUT 717;":CONT:MULT1:OUTP:B?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA

Description

This command sets/gets the output voltage of the group B with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output voltage for group B
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.B.VOLTage.DATA =0
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.B.VOLTage.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B:VOLTage[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:B:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:B:VOLT 0.5"
20 OUTPUT 717;":CONT:MULT1:OUTP:B:VOLT?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.DATA

Description

This command sets/gets the output port data for group C with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output port data for group C
Data type	Long integer type (Long)
Range	0 to 15
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.C.DATA =10
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.C.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:C 5"
20 OUTPUT 717;":CONT:MULT1:OUTP:C?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.C.VOLTage.DATA

Description

This command sets/gets the output voltage of the group C with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output voltage for group C
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.C.VOLTage.DATA =5.2
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.C.VOLTage.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C:VOLTage[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:C:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:C:VOLT 4.5"
20 OUTPUT 717;":CONT:MULT1:OUTP:C:VOLT?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.DATA

Description

This command sets/gets the output port data for group D with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output port data for group D
Data type	Long integer type (Long)
Range	0 to 15
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.D.DATA = 5
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.D.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:D 4"
20 OUTPUT 717;":CONT:MULT1:OUTP:D?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA = *Value*

Value = SCPI.CONTrol.MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA

Description

This command sets/gets the output voltage of the group D with ID of the multiport test set.

Variable

Parameter	*Value*
Description	Output voltage for group D
Data type	Double precision floating point type (Double)
Range	0 to 5.2V
Resolution	10 mV
Preset value	0V
Note	This command is available for E5092A multiport test set.

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.D.VOLTage.DATA = 5.2
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.D.VOLTage.DATA
```

Related objects

SCPI.CONTrol.MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.CONTrol.MULTiplexer(Id).OUTPut.C.VOLTage.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D:VOLTage[:DATA] <numeric>

:CONTrol:MULTiplexer{[1]|2}:OUTPut:D:VOLTage[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":CONT:MULT1:OUTP:D:VOLT 1.2"
20 OUTPUT 717;":CONT:MULT1:OUTP:D:VOLT?"
30 ENTER 717;A
```

SCPI.CONTrol.MULTiplexer(Id).PORT(Pt).SELect

Object type

Property (**Write Only**)

Syntax

SCPI.CONTrol.MULTiplexer(*Id*).PORT(Pt).SELect

Description

This command sets the multiport test set port.

Variable

Parameter	*Pt*
Description	Test set port
Data type	
Note	This command is available for E5092A multiport test set.

Examples

SCPI.CONTrol.MULTiplexer(1).PORT(1).SELect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:CONTrol:MULTiplexer{[1]|2}:PORT:SELect

Example of use

```
10 OUTPUT 717;":CONT:MULT1:PORT:SEL"
```

DISPLAY

SCPI.DISPlay.ANNotation.FREQuency.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.ANNotation.FREQuency.STATe = *Status*

Status = SCPI.DISPlay.ANNotation.FREQuency.STATe

Description

This command turns ON/OFF the frequency display on the LCD display.

Variable

Parameter	*Status*
Description	Sets/Gets ON/OFF state of the frequency display
Data type	Boolean type (Boolean)
Range	Select either of the following: • True or ON: Turns ON the frequency display. • False or OFF: Turns OFF the frequency display.
Preset value	True or ON

Examples

```
Dim DispFreq As Boolean
SCPI.DISPlay.ANNotation.FREQuency.STATe = 0
DispFreq = SCPI.DISPlay.ANNotation.FREQuency.STATe
```

Equivalent key

Display > **Frequency**

Equivalent SCPI command

Syntax

:DISPlay:ANNotation:FREQuency[:STATe] {ON|OFF|1|0}

:DISPlay:ANNotation:FREQuency[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:ANN:FREQ OFF"
20 OUTPUT 717;":DISP:ANN:FREQ?"
30 ENTER 717;A
```

SCPI.DISPlay.CCLear

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.CCLear

Description

This command clears the error message displayed in the status bar (at the bottom of the LCD display).

Examples

SCPI.DISPlay.CCLear

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:CCLear

Example of use

```
10 OUTPUT 717;":DISP:CCL"
```

SCPI.DISPlay.CLOCk

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.CLOCk = *Status*

Status = SCPI.DISPlay.CLOCk

Description

This command turns ON/OFF the clock display in the instrument status bar (at the right bottom of the LCD display).

Variable

Parameter	*Status*
Description	ON/OFF of the clock display
Data type	Boolean type (Boolean)
Range	Select either of the following: • True or ON: Turns ON the clock display. • False or OFF: Turns OFF the clock display.
Preset value	True or ON

Examples

```
Dim DispTime As Boolean
SCPI.DISPlay.CLOCk = ON
DispTime = SCPI.DISPlay.CLOCk
```

Equivalent key

System > **Misc Setup** > **Clock Setup** > **Show Clock**

Equivalent SCPI command

Syntax

:DISPlay:CLOCk {ON|OFF|1|0}

:DISPlay:CLOCk?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:CLOC OFF"
20 OUTPUT 717;":DISP:CLOC?"
30 ENTER 717;A
```

SCPI.DISPlay.COLor(*Dnum*).BACK

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).BACK = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).BACK

Description

This command sets/gets the background color for normal display (*Dnum*:1) and inverted display (Dnum:2).

Variable

Parameter	*Dnum*
Description	Select either of the following: 1: Normal display 2: Inverted display
Data type	Long integer type (Long)
Range	1 or 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when the command is executed.

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :* Sets amount of red. • *Data(1) :* Sets amount of green. • *Data(2) :* Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)

Range	• *Data(0)* 0 to 5 • *Data(1)* 0 to 5 • *Data(2)* 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim BackColor As Variant
SCPI.DISPlay.COLor(1).BACK = Array(1,2,3)
BackColor = SCPI.DISPlay.COLor(1).BACK
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Display Setup** > **Color Setup** > **Normal|Invert** > **Background**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:BACK <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:BACK?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL:BACK 1,2,3"
20 OUTPUT 717;":DISP:COL:BACK?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*)

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*) = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).GRATicule(*Gnum*)

Description

This command sets/gets:

1. Color of the graticule label.
2. Outer frame line of the graph (*Gnum*:1).
3. Color of the grid line of the graph (*Gnum*:2).

for the normal display (*Dnum*:1) and inverted display (*Dnum*:2).

Variable

Parameter	*Gnum*
Description	The number of items: 1: The outer frame line of the graph 2: The color of the grid line of the graph
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :* Sets amount of red. • *Data(1) :* Sets amount of green.

	• *Data(2) :* Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :* 0 to 5 • *Data(1) :* 0 to 5 • *Data(2) :* 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim GritColor As Variant
SCPI.DISPlay.COLor(1).GRATicule(1) = Array(1,2,3)
GritColor = SCPI.DISPlay.COLor(1).GRATicule(1)
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Display Setup** > **Color Setup** > **Normal**|**Invert** > **Graticule Main**|**Graticule Sub**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:GRATicule{[1]|2} <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:GRATicule{[1]|2}?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:GRAT1 1,2,3"
20 OUTPUT 717;":DISP:COL1:GRAT1?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*)

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*) = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).LIMit(*Lnum*)

Description

This command sets/gets:

- Fail display color used for the limit test result, Bandwidth test result and Ripple test result (Lnum:1)
- Color of the limit line (Lnum:2)
  for normal display (*Dnum*:1) and inverted display (*Dnum*: 2).

Variable

Parameter	*Lnum*
Description	The number of item 1: The limit test result (Fail/Pass) 2: The limit line
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :* Sets amount of red. • *Data(1) :* Sets amount of green.

	• *Data(2) :* Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :* 0 to 5 • *Data(1) :*0 to 5 • *Data(2) :*0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim LimColor As Variant
SCPI.DISPlay.COLor(1).LIMit(1) = Array(1,2,3)
LimColor = SCPI.DISPlay.COLor(1).LIMit(1)
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal**|**Invert** > **Limit Fail**|**Limit Line**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:LIMit{[1]|2} <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:LIMit{[1]|2}?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:LIM1 1,2,3"
20 OUTPUT 717;":DISP:COL1:LIM1?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).RESet

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).RESet

Description

This command resets the display color settings for all the items to the factory preset state, for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Examples

SCPI.DISPlay.COLor(1).RESet

Related objects

SCPI.DISPlay.COLor(Dnum).BACK

SCPI.DISPlay.COLor(Dnum).GRATicule(Gnum)

SCPI.DISPlay.COLor(Dnum).LIMit(Lnum)

SCPI.DISPlay.COLor(Dnum).TRACe(Tr).DATA

SCPI.DISPlay.COLor(Dnum).TRACe(Tr).MEMory

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal**|**Invert** > **Reset Color** > **OK**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:RESet

Example of use

```
10 OUTPUT 717;":DISP:COL1:RES"
```

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).DATA

Description

This command sets/gets the color of the data trace of traces 1 to 36 (*Tr*), for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :*Sets amount of red. • *Data(1) :*Sets amount of green. • *Data(2) :*Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :* 0 to 5 • *Data(1) :* 0 to 5 • *Data(2) :* 0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TrColor As Variant
SCPI.DISPlay.COLor(1).TRACe(1).DATA = Array(1,2,3)
TrColor = SCPI.DISPlay.COLor(1).TRACe(1).DATA
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal**|**Invert** > **Data Trace 1**|**Data Trace 2**|
Data Trace 3|**Data Trace 4**|**Data Trace 5**|**Data Trace 6**|**Data Trace 7**|**Data Trace 8**|**Data Trace 9**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:DATA <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:DATA?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:TRAC1:DATA 1,2,3"
20 OUTPUT 717;":DISP:COL1:TRAC1:DATA?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory = *Data*

Data = SCPI.DISPlay.COLor(*Dnum*).TRACe(*Tr*).MEMory

Description

This command sets/gets the color of the memory trace of traces 1 to 36 (*Tr*), for normal display (*Dnum*: 1) and inverted display (*Dnum*: 2).

Variable

Parameter	*Data*
Description	Indicates 3-element array data. • *Data(0) :*Sets amount of red. • *Data(1) :*Sets amount of green. • *Data(2) :*Sets amount of blue. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0) :*0 to 5 • *Data(1) :*0 to 5 • *Data(2) :*0 to 5
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim TrColor As Variant
SCPI.DISPlay.COLor(1).TRACe(1).MEMory = Array(1,2,3)
TrColor = SCPI.DISPlay.COLor(1).TRACe(1).MEMory
```

Related objects

SCPI.DISPlay.COLor(Dnum).RESet

Equivalent key

System > **Misc Setup** > **Color Setup** > **Normal**|**Invert** > **Mem Trace 1**|**Mem Trace 2**|
Mem Trace 3|**Mem Trace 4**|**Mem Trace 5**|**Mem Trace 6**|**Mem Trace 7**|**Mem Trace 8**|**Mem Trace 9**

Equivalent SCPI command

Syntax

:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:MEMory <numeric 1>,<numeric 2>,<numeric 3>

:DISPlay:COLor{[1]|2}:TRAC{[1]-36}:MEMory?

Query response

{numeric 1},{numeric 2},{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:COL1:TRAC1:MEM 1,2,3"
20 OUTPUT 717;":DISP:COL1:TRAC1:MEM?"
30 ENTER 717;A,B,C
```

SCPI.DISPlay.ECHO.CLEar

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.ECHO.CLEar

Description

This command clears all character strings displayed in the echo window.

Examples

SCPI.DISPlay.ECHO.CLEar

Related objects

SCPI.DISPlay.ECHO.DATA

Equivalent key

Macro Setup > **Clear Echo**

Equivalent SCPI command

Syntax

:DISPlay:ECHO:CLEar

Example of use

```
10 OUTPUT 717;":DISP:ECHO:CLE"
```

SCPI.DISPlay.ECHO.DATA

Object type

Property (**Write Only**)

Syntax

SCPI.DISPlay.ECHO.DATA = *Cont*

Description

This command displays a character string in the echo window. This command is different from ECHO command as it displays a single character string.

Variable

Parameter	*Cont*
Description	String you want to display in the echo window.
Data type	Character string type (String)
Range	254 characters or less

Examples

```
SCPI.DISPlay.ECHO.DATA = "Test Result"
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
```

Related objects

ECHO

SCPI.DISPlay.TABLe.TYPE

SCPI.DISPlay.TABLe.STATe

SCPI.DISPlay.ECHO.CLEar

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:ECHO[:DATA] <string>

Example of use

```
10 OUTPUT 717;":DISP:ECHO ""TEST RESULT"""
```

SCPI.DISPlay.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.ENABle = *Status*

Status = SCPI.DISPlay.ENABle

Description

This command turns ON/OFF the display update on the E5071C measurement screen.

Variable

Parameter	*Status*
Description	ON/OFF of the display update of the E5071C measurement screen
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display update. • False or OFF : Turns OFF the display update.
Preset value	True or ON

Examples

```
Dim DispUpdt As Boolean
SCPI.DISPlay.ENABle = False
DispUpdt = SCPI.DISPlay.ENABle
```

Equivalent key

Display > **Update**

Equivalent SCPI command

Syntax

:DISPlay:ENABle {ON|OFF|1|0}

:DISPlay:ENABle?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:ENAB OFF"
20 OUTPUT 717;":DISP:ENAB?"
30 ENTER 717;A
```

SCPI.DISPlay.FSIGn

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.FSIGn = *Status*

Status = SCPI.DISPlay.FSIGn

Description

This command turns ON/OFF the "Fail" display on the LCD screen when the limit test, bandwidth test and ripple test fails.

Variable

Parameter	*Status*
Description	ON/OFF of the "Fail" display when the limit test fails
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the "Fail" display. • False or OFF : Turns OFF the "Fail" display.
Preset value	True or ON

On/off of the Fail display cannot be set at each test. When the Fail display of either of test is turned on, the Fail display of other tests turns on, too.

Examples

```
Dim DispFail As Boolean
SCPI.DISPlay.FSIGn = False
DispFail = SCPI.DISPlay.FSIGn
```

Related objects

SCPI.CALCulate(Ch).SELected.LIMit.STATe

SCPI.CALCulate(Ch).SELected.RLIMit.STATe

SCPI.CALCulate(Ch).SELected.BLIMit.STATe

Equivalent key

Analysis > **Limit Test > Fail Sign**

Analysis > **Ripple Limit > Fail Sign**

Analysis > **Bandwidth limit > Fail Sign**

Equivalent SCPI command

Syntax

```
:DISPlay:FSIGn {ON|OFF|1|0}
:DISPlay:FSIGn?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:FSIG OFF"
20 OUTPUT 717;":DISP:FSIG?"
30 ENTER 717;A
```

SCPI.DISPlay.IMAGe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.IMAGe = *Param*

Param = SCPI.DISPlay.IMAGe

Description

This command sets/gets the display type of the LCD display.

Variable

Parameter	*Param*
Description	Display type of the LCD display
Data type	Character string type (String)
Range	Select from the following: • "NORMal" : Specifies the normal display (background color: black). • "INVert": Specifies the display in which the color of the normal display is inversed (background color: white).
Preset value	"NORMal"

Examples

```
Dim DispImg As String
SCPI.DISPlay.IMAGe = "inv"
DispImg = SCPI.DISPlay.IMAGe
```

Equivalent key

Display > **Invert Color**

Equivalent SCPI command

Syntax

:DISPlay:IMAGe {NORMal|INVert}

:DISPlay:IMAGe?

Query response

{NORM|INV}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:IMAG INV"
20 OUTPUT 717;":DISP:IMAG?"
30 ENTER 717;A$
```

SCPI.DISPlay.MAXimize

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.MAXimize = *Status*

Status = SCPI.DISPlay.MAXimize

Description

This command turns ON/OFF the window maximization of the active channel.

If the maximization is set to ON, only the window of the active channel is maximized on the LCD display and the windows of the other channels are not displayed.

Variable

Parameter	*Status*
Description	ON/OFF of the window maximization
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the window maximization. • False or OFF: Turns OFF the window maximization.
Preset value	False or OFF

Examples

```
Dim ChMax As Boolean
SCPI.DISPlay.SPLit = "d1_2"
SCPI.DISPlay.WINDow(2).ACTivate
SCPI.DISPlay.MAXimize = True
ChMax = SCPI.DISPlay.MAXimize
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Channel Max

Equivalent SCPI command

Syntax

```
:DISPlay:MAXimize {ON|OFF|1|0}
:DISPlay:MAXimize?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:MAX ON"
20 OUTPUT 717;":DISP:MAX?"
30 ENTER 717;A
```

SCPI.DISPlay.SKEY.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.SKEY.STATe = *Status*

Status = SCPI.DISPlay.SKEY.STATe

Description

This command turns ON/OFF the display of the softkey menu bar.

Variable

Parameter	*Status*
Description	ON/OFF of the softkey menu bar display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the softkey menu bar display. • False or OFF: Turns OFF the softkey menu bar display.
Preset value	True or ON

Examples

```
Dim DispSkey As Boolean
SCPI.DISPlay.SKEY.STATe = False
DispSkey = SCPI.DISPlay.SKEY.STATe
```

Equivalent key

Entry Off

Equivalent SCPI command

Syntax

:DISPlay:SKEY[:STATe] {ON|OFF|1|0}

:DISPlay:SKEY[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:MAX ON"
20 OUTPUT 717;":DISP:MAX?"
30 ENTER 717;A
```

SCPI.DISPlay.SPLit

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.SPLit = *Param*

Param = SCPI.DISPlay.SPLit

Description

Sets the layout of the channel windows on the LCD display.

Variable

Parameter	*Param*
Description	Layout of channel windows
Data type	Character string type (String)
Range	Select from the following: • "D1": See Channel graph window layouts. • "D12": See Channel graph window layouts. • "D1_2": See Channel graph window layouts. • "D112": See Channel graph window layouts. • "D1_1_2": See Channel graph window layouts. • "D123": See Channel graph window layouts. • "D1_2_3": See Channel graph window layouts. • "D12_33": See Channel graph window layouts. • "D11_23": See Channel graph window layouts. • "D13_23": See Channel graph window layouts. • "D12_13": See Channel graph window layouts. • "D1234": See Channel graph window

	layouts. • "D1_2_3_4": See Channel graph window layouts. • "D12_34": See Channel graph window layouts. • "D123_456": See Channel graph window layouts. • "D12_34_56": See Channel graph window layouts. • "D1234_5678": See Channel graph window layouts. • "D12_34_56_78": See Channel graph window layouts. • "D123_456_789": See Channel graph window layouts. • "D123__ABC": See Channel graph window layouts. • "D1234__9ABC": See Channel graph window layouts. • "D1234__CDEF": See Channel graph window layouts. • "D1234__DEFG": See Channel graph window layouts. • "D1X1" • "D1X2" • "D1X3" • "D1X4" • "D2X1" • "D2X2" • "D2X3" • "D2X4" • "D3X1" • "D3X2" • "D3X3" • "D3X4"

	• "D4X1" • "D4X2" • "D4X3" • "D4X4" • "D4X5" • "D4X6" • "D4X7" • "D4X8" • "D4X9" • "D8X9" • "D6X12" • "D8X12" • "D10X16"
Preset value	"D1"

- From Firmware revision A.08.10, channel display configuration of 8x9, 6x12 and 8x12 is added.

Examples

```
Dim ChanAloc As String
SCPI.DISPlay.SPLit = "d12_34"
ChanAloc = SCPI.DISPlay.SPLit
```

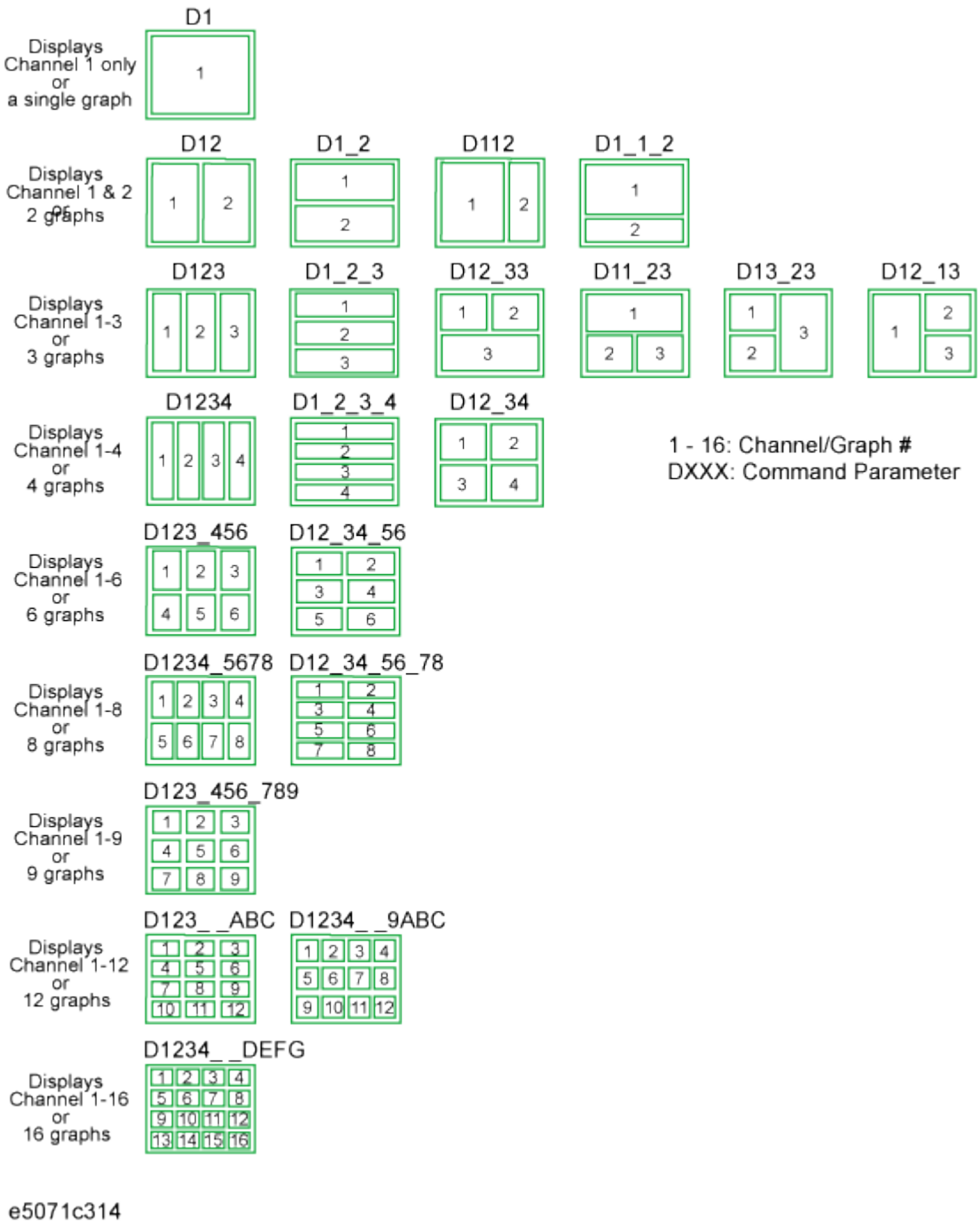
Related objects

SCPI.DISPlay.WINDow(Ch).SPLit

Equivalent key

Display > **Allocate Channels**

Channel/graph window layouts



e5071c314

Equivalent SCPI command

Syntax

:DISPlay:SPLit

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|
D1234|D1_2_3_4|D12_34|D123_456|D12_34_56|D1234_5678|D12_34_56_78|D123_456_789|

D123__ABC|D1234__9ABC|D1234__CDEF|D1234__DEFG|D1X1|D1X2|D1X3|D1X4|D2X1|D2X2|D2X3|D2X4|D3X1|
D3X2|D3X3|D3X4|D4X1|D4X2|D4X3|D4X4|D4X5|D4X6|D4X7|D4X8|D4X9
}

:DISPlay:SPLit?

Query response

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|
D1234|D1_2_3_4|D12_34|D123_456|D12_34_56|D1234_5678|D12_34_56_78|D123_456_789|
D123__ABC|D1234__9ABC|D1234__CDEF|D1234__DEFG|D1X1|D1X2|D1X3|D1X4|D2X1|D2X2|D2X3|D2X4|D3X1|
D3X2|D3X3|D3X4|D4X1|D4X2|D4X3|D4X4|D4X5|D4X6|D4X7|D4X8|D4X9
}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:SPL D1_2"
20 OUTPUT 717;":DISP:SPL?"
30 ENTER 717;A$
```

SCPI.DISPlay.TABLe.POSition.RECTangle

Object type

Property (**Read Only**)

Syntax

Param = SCPI.DISPlay.TABLe.POSition.RECTangle

Description

This command reads the display coordinates position of Table area (the top left of the display is [0, 0]). If SCPI.DISPlay.TABLe.STATe is OFF, 0, 0, 0, 0 will be returned. .

Variable

Parameter	*Param*
Description	Indicates the coordinates position of Table Area. • *Param(0)* : coordinates X position of top left of Table Area. • *Param(1)* : coordinates Y position of top left of Table Area. • *Param(2)* : coordinates X position of bottom right of Table Area. • *Param(3)* : coordinates Y position of bottom right of Table Area. The index of the array starts from 0.
Data type	Variant type (Variant)
Resolution	1

Examples

```
Dim TablePos() As Variant
SCPI.DISPlay.TABLe.STATe = True
TablePos = SCPI.DISPlay.TABLe.POSition.RECTangle
```

Related objects

SCPI.DISPlay.TABLe.STATe

Equivalent key

None

Equivalent SCPI command

Syntax

:DISPlay:TABLe:POSition[:RECTangle]?

Query response

{numeric1},{numeric2},{numeric3},{numeric4},<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL:STAT ON"
20 OUTPUT 717;":DISP:TABL:POS?"
20 ENTER 717;A, B, C, D
```

SCPI.DISPlay.TABLe.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.TABLe.STATe = *Status*

Status = SCPI.DISPlay.TABLe.STATe

Description

This command turns ON/OFF the display of the window that appears in the lower part of the LCD display (specified by SCPI.DISPlay.TABLe.TYPE object).

Variable

Parameter	*Status*
Description	ON/OFF of the display of the window that appears in the lower part of the LCD display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the display. • False or OFF: Turns OFF the display.
Preset value	False or OFF

Examples

```
Dim DispTbl As Boolean
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
DispTbl = SCPI.DISPlay.TABLe.STATe
```

Related objects

SCPI.DISPlay.TABLe.TYPE

Equivalent key

Sweep Setup > **Edit Segment Table**

Marker Fctn > **Marker Table**

Analysis > **Limit Test** > **Edit Limit Line**

Analysis > **Ripple Limit** > **Edit Ripple Line**

Macro Setup > **Echo Window**

Cal > **Power Calibration** > **Loss Compen**

Cal > **Power Calibration** > **Sensor A Settings | Sensor B Settings**

When performing the operation from the front panel, you select the type of the window that appears in the lower part of the LCD display and turn ON/OFF the display at the same time.

Equivalent SCPI command

Syntax

:DISPlay:TABLe[:STATe] {ON|OFF|1|0}

:DISPlay:TABLe[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL ON"
20 OUTPUT 717;":DISP:TABL?"
30 ENTER 717;A
```

SCPI.DISPlay.TABLe.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.TABLe.TYPE = *Param*

Param = SCPI.DISPlay.TABLe.TYPE

Description

This command selects the type of the window that appears in the lower part of the LCD display.

Variable

Parameter	*Param*
Description	Window type
Data type	Character string type (String)
Range	Select from the following: • "MARKer": Specifies the marker table window. • "LIMit": Specifies the limit test table window. • "SEGMent": Specifies the segment table window. • "ECHO": Specifies the echo window. • "PLOSs": Specifies the loss compensation table window. • "SCFactor": Specifies the power sensor's calibration factor table window. • "RLIMit": Specifies the ripple test table window.
Preset value	"MARKer"

Examples

```
Dim TblType As String
SCPI.DISPlay.TABLe.TYPE = "echo"
SCPI.DISPlay.TABLe.STATe = True
TblType = SCPI.DISPlay.TABLe.TYPE
```

Related objects

SCPI.DISPlay.TABLe.STATe

Equivalent key

Sweep Setup > **Edit Segment Table**

Marker Fctn > **Marker Table**

Analysis > **Limit Test** > **Edit Limit Line**

Analysis > **Ripple Limit** > **Edit Ripple Line**

Macro Setup > **Echo Window**

Cal > **Power Calibration** > **Loss Compen**

Cal > **Power Calibration** > **Sensor A Settings**|**Sensor B Settings**

When performing the operation from the front panel, you select the type of the window that appears in the lower part of the LCD display and turn ON/OFF the display at the same time.

Equivalent SCPI command

Syntax

:DISPlay:TABLe:TYPE
{MARKer|LIMit|SEGMent|ECHO|PLOSs|SCFactor|RLIMit}

:DISPlay:TABLe:TYPE?

Query response

{MARK|LIM|SEGM|ECHO|PLOS|SCF|RLIM}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:TABL:TYPE SEGM"
20 OUTPUT 717;":DISP:TABL:TYPE?"
30 ENTER 717;A$
```

SCPI.DISPlay.UPDate.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.UPDate.IMMediate

Description

This command executes the display update once when the display update of the LCD screen is set to OFF (specifying False with the SCPI.DISPlay.ENABle object),

Examples

```
SCPI.DISPlay.ENABle = False
SCPI.DISPlay.UPDate.IMMediate
```

Related objects

SCPI.DISPlay.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:DISPlay:UPDate[:IMMediate]

Example of use

```
10 OUTPUT 717;":DISP:UPD"
```

SCPI.DISPlay.WINDow(*Ch*).ACTivate

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ACTivate

Description

This command specifies selected channel (*Ch*) as the active channel.

- You can set only a channel displayed to the active channel. If this object is used to set a channel not displayed to the active channel, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.SPLit = "d1_2"
SCPI.DISPlay.WINDow(2).ACTivate
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Channel Prev / **Channel Next**

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:ACTivate
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ACT"
```

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.ALIGn.STATe

Description

This command turn ON/OFF the mode that align the marker display position of each trace based on trace 1, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the mode that align the marker display position of each trace based on trace 1
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON:Turns ON the mode that allign marker display position based on trace 1. • False or OFF:Turns OFF the alignment.
Preset value	True or ON

Examples

```
Dim AnnMarkAlig As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe = False
AnnMarkAlig = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.ALIGn.STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Align**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:ALIGn[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:ALIGn[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ANN:MARK:ALIG OFF"
20 OUTPUT 717;":DISP:WIND1:ANN:MARK:ALIG?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).ANNotation.MARKer.SINGle.STATe

Description

This command turns ON/OFF the display of the marker value of only active traces, for the selected channel (*Ch*).

- If the function is turned OFF, marker values of all traces (markers) are displayed.

Variable

Parameter	*Status*
Description	ON/OFF the display of the marker value of only active
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Displays the marker values of only active traces.(ON) • False or OFF: Displays the marker values of all traces. (OFF)
Preset value	True or ON

Examples

```
Dim AnnMarkAlig As Boolean
SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe = False
AnnMarkAlig = SCPI.DISPlay.WINDow(1).ANNotation.MARKer.SINGle.STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Active Only**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:SINGle[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:ANNotation:MARKer:SINGle[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:ANN:MARK:SING OFF"
20 OUTPUT 717;":DISP:WIND1:ANN:MARK:SING?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).LABel = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).LABel

Description

This command turns ON/OFF the graticule label display of the graph of channels 1 to 36 (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the graticule label display of the graph
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the graticule label display. • False or OFF: Turns OFF the graticule label display.
Preset value	True or ON

Examples

```
Dim DispGrat As Boolean
SCPI.DISPlay.WINDow(1).LABel = False
DispGrat = SCPI.DISPlay.WINDow(1).LABel
```

Equivalent key

Display > **Graticule Label**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:LABel {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:LABel?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:LAB ON"
20 OUTPUT 717;":DISP:WIND1:LAB?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).MAXimize

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).MAXimize = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).MAXimize

Description

This command turns ON/OFF the maximization of the active trace of selected channel (*Ch*).

- If you turned ON the maximization, only the maximized active trace is displayed in the window and the other traces are not displayed.

Variable

Parameter	*Status*
Description	ON/OFF of the maximization of the active trace
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the maxim display. • False or OFF: Turns OFF the maxim display.
Preset value	False or OFF

Examples

```
Dim TracMax As Boolean
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.DISPlay.WINDow(1).MAXimize = True
TracMax = SCPI.DISPlay.WINDow(1).MAXimize
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.DISPlay.MAXimize

Equivalent key

Trace Max

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:MAXimize {ON|OFF|1|0}
:DISPlay:WINDow{[1]-160}:MAXimize?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:MAX ON"
20 OUTPUT 717;":DISP:WIND1:MAX?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).SPLit

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).SPLit = *Param*

Param = SCPI.DISPlay.WINDow(*Ch*).SPLit

Description

This command sets/gets the graph layout of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Graph layout
Data type	Character string type (String)
Range	Select from the following: • "D1": See Channel graph window layouts. • "D12": See Channel graph window layouts. • "D1_2": See Channel graph window layouts. • "D112": See Channel graph window layouts. • "D1_1_2": See Channel graph window layouts. • "D123": See Channel graph window layouts. • "D1_2_3": See Channel graph window layouts. • "D12_33": See Channel graph window layouts. • "D11_23": See Channel graph window layouts. • "D13_23": See Channel graph window layouts. • "D12_13": See Channel graph window layouts. • "D1234": See Channel graph window layouts. • "D1_2_3_4": See Channel graph window

	layouts. • "D12_34": See Channel graph window layouts. • "D123_456": See Channel graph window layouts. • "D12_34_56": See Channel graph window layouts. • "D1234_5678": See Channel graph window layouts. • "D12_34_56_78": See Channel graph window layouts. • "D123_456_789": See Channel graph window layouts. • "D123__ABC": See Channel graph window layouts. • "D1234__9ABC": See Channel graph window layouts. • "D1234__DEFG": See Channel graph window layouts.
Preset value	"D1"

Examples

```
Dim TracAloc As String
SCPI.DISPlay.WINDow(1).SPLit = "d1_2"
TracAloc = SCPI.DISPlay.WINDow(1).SPLit
```

Related objects

SCPI.DISPlay.SPLit

Equivalent key

Display > **Allocate Traces**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-36}:SPLit {D1|D12|D1_2|D112|D1_1_2|
D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|D1234|D1_2_3_4|D12_3
4|D123_456|D12_34_56|
D1234_5678|D12_34_56_78|D123_456_789|D123__ABC|D1234__9ABC|
D1234__DEFG}

:DISPlay:WINDow{[1]-36}:SPLit?

Query response

{D1|D12|D1_2|D112|D1_1_2|D123|D1_2_3|D12_33|D11_23|D13_23|D12_13|D1234|D1_2_3_4|
D12_34|D123_456|D12_34_56|D1234_5678|D12_34_56_78|D123_456_789|D123__ABC|
D1234__9ABC|D1234__DEFG}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND:SPL D1_2"
20 OUTPUT 717;":DISP:WIND:SPL?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA = *Lbl*

Lbl = SCPI.DISPlay.WINDow(*Ch*).TITLe.DATA

Description

This command sets/gets the title label displayed in the title area of selected channel (*Ch*).

Variable

Parameter	*Lbl*
Description	Title label
Data type	Character string type (String)
Range	254 characters or less
Preset value	""

Examples

```
Dim TtlLbl As String
SCPI.DISPlay.WINDow(1).TITLe.DATA = "Filter"
SCPI.DISPlay.WINDow(1).TITLe.STATe = True
TtlLbl = SCPI.DISPlay.WINDow(1).TITLe.DATA
```

Related objects

SCPI.DISPlay.WINDow(Ch).TITLe.STATe

Equivalent key

Display > **Edit Title Label**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TITLe:DATA <string>

:DISPlay:WINDow{[1]-160}:TITLe:DATA?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TITL:DATA ""Title"""
20 OUTPUT 717;":DISP:WIND1:TITL?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).TITLe.STATe

Description

This command turns ON/OFF the title label display in the title area of channels 1 to 160 (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the title label display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the title label display. • False or OFF: Turns ON the title label display.
Preset value	False or OFF

Examples

```
Dim DispTtl As Boolean
SCPI.DISPlay.WINDow(1).TITLe.DATA = "Filter"
SCPI.DISPlay.WINDow(1).TITLe.STATe = True
DispTtl = SCPI.DISPlay.WINDow(1).TITLe.STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).TITLe.DATA

Equivalent key

Display > **Title Label**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TITLe[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:TITLe[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TITL ON"
20 OUTPUT 717;":DISP:WIND1:TITL?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

This command sets/gets the display position of the marker value on the X-axis by a percentage of a width of the display span, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Display position of the marker value on the X-axis.
Data type	Double precision floating point type (Double)
Range	-15 to 100
Preset value	1
Unit	% (percent)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AnnMPosX As Double
SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.X = 15
AnnMPosX = SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.X
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.Y

Equivalent key

Marker Fctn > **Annotation Options** > **Marker Info X Pos**

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16}:ANNotation:MARKer:POSition:X <numeric>
:DISPlay:WINDow{[1]-160}:TRACe{[1]-16}:ANNotation:MARKer:POSition:X?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:X 33"
20 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:X?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.Y = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).ANNotation.MARKer.POSition.X

Description

This command sets/gets the display position of the marker value on the X-axis by a percentage of a width of the display span, for the selected trace (*Tr*) of selected channel (*Ch*), and the marker value on Y axis by a percentage of a height of the display span.

Variable

Parameter	*Value*
Description	Display position of the marker value on the Y-axis.
Data type	Double precision floating point type (Double)
Range	-15 to 100
Preset value	1
Unit	% (percent)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AnnMPosY As Double
SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.Y = 23
AnnMPosY = SCPI.DISPlay.WINDow(1).TRACe(1).ANNotation.MARKer.POSition.Y
```

Related objects

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.ALIGn.STATe

SCPI.DISPlay.WINDow(Ch).ANNotation.MARKer.SINGle.STATe

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).ANNotation.MARKer.POSition.X

Equivalent key

Marker Fctn > **Annotation Options** > **Marker Info X Pos**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16}:ANNotation:MARKer:POSition:Y <numeric>

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16}:ANNotation:MARKer:POSition:Y?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:Y 33"
20 OUTPUT 717;":DISP:WIND1:TRAC1:ANN:MARK:POS:Y?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory. STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory.STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).MEMory.STATe

Description

This command turns ON/OFF the memory trace display, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the memory trace display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the memory trace display. • False or OFF: Turns OFF the memory trace display.
Preset value	False or OFF

Examples

```
Dim DispMem As Boolean
SCPI.DISPlay.WINDow(1).TRACe(2).MEMory.STATe = True
DispMem = SCPI.DISPlay.WINDow(1).TRACe(2).MEMory.STATe
```

Related objects

SCPI.CALCulate(Ch).SELected.MATH.MEMorize

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).STATe

Equivalent key

Display > **Display** > **Mem** (when the data trace display is OFF)

Display > **Display** > **Data & Mem** (when the data trace display is ON)

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :MEMory[:STATe] {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :MEMory[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:MEM ON"
20 OUTPUT 717;":DISP:WIND1:TRAC1:MEM?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe = *Status*

Status = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).STATe

Description

This command turns ON/OFF the data trace display, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the data trace display
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the data trace display. • False or OFF: Turns OFF the data trace display.
Preset value	True or ON

Examples

```
Dim DispTrac As Boolean
SCPI.DISPlay.WINDow(1).TRACe(2).STATe = False
DispTrac = SCPI.DISPlay.WINDow(1).TRACe(2).STATe
```

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).MEMory.STATe

Equivalent key

Display > **Display** > **Data** (when the memory trace display is OFF)

Display > **Display** > **Data & Mem** (when the memory trace display is ON)

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :STATe {ON|OFF|1|0}

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:STAT ON"
20 OUTPUT 717;":DISP:WIND1:TRAC1:STAT?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO

Object type

Method (**Write-only**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.AUTO

Description

This command executes the auto scale function, for the selected trace (*Tr*) of selected channel (*Ch*). The Auto Scale function automatically adjusts the value of the reference division line and the scale per division to display the trace appropriately.

Examples

SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.AUTO

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

Equivalent key

Scale > **Auto Scale**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:AUTO

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:AUTO"
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe. PDIVision

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.PDIVision

Description

For the selected trace (*Tr*) of selected channel (*Ch*), when the data format is not the Smith chart format or the polar format, sets the scale per division. When the data format is the Smith chart format or the polar format, sets the full scale value (the value of the outermost circumference).

Variable

Parameter	*Value*
Description	Scale value
Data type	Double precision floating point type (Double)
Range	1E-18 to 1E8
Preset value	Varies depending the data format. • Log magnitude: 10 • Phase, Expanded phase or Positive phase: 90 • Group delay: 1E-8 • Smith chart or Polar or SWR: 1 • Linear magnitude: 0.1 • Real or Imaginary: 0.2
Unit	Varies depending on the data format. • Log magnitude: dB (decibel) • Phase, Expanded phase or Positive phase: ° (degree) • Group delay: s (second) • Others: No unit

Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pdiv As Double
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.FORMat = "gdel"
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.PDIVision = 1E-9
Pdiv = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.PDIVision
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

Equivalent key

Scale > **Scale/Div**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:PDIVision <numeric>

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:PDIVision?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:PDIV 2.5"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:PDIV?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RLEVel

Description

This command sets/gets the value of the reference division line, for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of reference division line
Data type	Double precision floating point type (Double)
Range	-5E8 to 5E8
Preset value	0
Unit	Varies depending on the data format. • Log magnitude (MLOG): dB (decibel) • Phase (PHAS), Expanded phase (UPH) or Positive phase (PPH): ° (degree) • Group delay (GDEL): s (second) • Others: No unit
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim RefLvl As Double
SCPI.CALCulate(1).PARameter(2).SELect
SCPI.CALCulate(1).SELected.FORMat = "phas"
```

```
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RLEVel = 90
Pdiv = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RLEVel
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

Equivalent key

[Scale] > **Reference Value**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:RLEVel <numeric>

:DISPlay:WINDow{[1]-160}:TRACe{[1]-16} :Y[:SCALe]:RLEVel?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RLEV 1E2"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RLEV?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).TRACe(*T36r*).Y.SCALe.RPOSition = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).TRACe(*Tr*).Y.SCALe.RPOSition

Description

This command specifies the position of a reference division line with its number (an integer assigned starting from 0 from the lowest division), for the selected trace (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Position of reference division line
Data type	Long integer type (Long)
Range	0 to the number of divisions
Preset value	5
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim RefPos As Long
SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RPOSition = 6
RefPos = SCPI.DISPlay.WINDow(1).TRACe(2).Y.SCALe.RPOSition
```

Related objects

SCPI.CALCulate(Ch).SELected.FORMat

SCPI.DISPlay.WINDow(Ch).Y.SCALe.DIVisions

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

Equivalent key

Scale > **Reference Position**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:TRACe{[1]-36} :Y[:SCALe]:RPOSition <numeric>

:DISPlay:WINDow{[1]-160}:TRACe{[1]-36} :Y[:SCALe]:RPOSition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RPOS 6"
20 OUTPUT 717;":DISP:WIND1:TRAC1:Y:RPOS?"
30 ENTER 717;A
```

SCPI.DISPlay.WINDow(*Ch*).X.SPACing

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).X.SPACing = *Param*

Param = SCPI.DISPlay.WINDow(*Ch*).X.SPACing

Description

This command selects the display type of the graph horizontal axis of selected channel (*Ch*) for segment sweep.

Variable

Parameter	*Param*
Description	Horizontal axis display type of the graph for segment sweep
Data type	Character string type (String)
Range	Select from the following: • "LINear": Specifies the frequency base (linear frequency axis with the minimum frequency at the left edge and the maximum frequency at the right edge). • "OBASe" : Specifies the order base (axis in which the measurement point numbers are positioned evenly in the order of measurement).
Preset value	"OBASe"

Examples

```
Dim DispSegm As String
SCPI.SENSe(1).SWEep.TYPE = "segm"
SCPI.DISPlay.WINDow(1).X.SPACing = "obas"
DispSegm = SCPI.DISPlay.WINDow(1).X.SPACing
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Segment Display**

Equivalent SCPI command

Syntax

:DISPlay:WINDow{[1]-160}:X:SPACing {LINear|OBASe}

:DISPlay:WINDow{[1]-160}:X:SPACing?

Query response

{LIN|OBAS}<newline><^END>

Example of use

```
10 OUTPUT 717;":DISP:WIND1:X:SPAC OBAS"
20 OUTPUT 717;":DISP:WIND1:X:SPAC?"
30 ENTER 717;A$
```

SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

Object type

Property (**Read-Write**)

Syntax

SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions = *Value*

Value = SCPI.DISPlay.WINDow(*Ch*).Y.SCALe.DIVisions

Description

This command sets/gets the number of divisions in all the graphs, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of divisions of graph
Data type	Long integer type (Long)
Range	4 to 30
Preset value	10
Resolution	2
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Divs As Long
SCPI.DISPlay.WINDow(1).Y.SCALe.DIVisions = 12
Divs = SCPI.DISPlay.WINDow(1).Y.SCALe.DIVisions
```

Related objects

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.PDIVision

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RLEVel

SCPI.DISPlay.WINDow(Ch).TRACe(Tr).Y.SCALe.RPOSition

Equivalent key

Scale > **Divisions**

Equivalent SCPI command

Syntax

```
:DISPlay:WINDow{[1]-160}:Y[:SCALe]:DIVisions <numeric>
:DISPlay:WINDow{[1]-160}:Y[:SCALe]:DIVisions?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":DISP:WIND1:Y:DIV 12"
20 OUTPUT 717;":DISP:WIND1:Y:DIV?"
30 ENTER 717;A
```

FORMAT

SCPI.FORMat.BORDer

Object type

Property (**Read-Write**)

Syntax

SCPI.FORMat.BORDer = *Param*

Param = SCPI.FORMat.BORDer

Description

This command sets/gets the transfer order of each byte in the output data (byte order), when the data transfer format is set to binary mode (by specifying "REAL" with SCPI.FORMat.DATA object).

- This object is NOT used when controlling the E5071C using COM objects through E5071C VBA.

Variable

Parameter	*Param*
Description	Byte order
Data type	Character string type (String)
Range	Select from the following: "NORMal": Specifies the byte order in which transfer starts from the byte including MSB (Most Significant Bit). "SWAPped": Specifies the byte order in which transfer starts from the byte including LSB (Least Significant Bit).
Preset value	"NORMal"

Examples

```
Dim BitOrd As String
SCPI.FORMat.BORDer "swap"
BitOrd = SCPI.FORMat.BORDer
```

Related objects

SCPI.FORMat.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:FORMat:BORDer {NORMal|SWAPped}

:FORMat:BORDer?

Query response

{NORM|SWAP}<newline><^END>

Example of use

```
10 OUTPUT 717;":FORM:BORD SWAP"
20 OUTPUT 717;":FORM:BORD?"
30 ENTER 717;A$
```

SCPI.FORMat.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.FORMat.DATA = *Param*

Param = SCPI.FORMat.DATA

Description

This command can be used to set/get format data using the following SCPI commands:

:CALC{1-160}:DATA:FDAT
:CALC{1-160}:DATA:FMEM
:CALC{1-160}:DATA:SDAT?
:CALC{1-160}:DATA:SMEM?
:CALC{1-160}:FUNC:DATA?
:CALC{1-160}:LIM:DATA
:CALC{1-160}:LIM:REP?
:CALC{1-160}:LIM:REP:ALL?
:CALC{1-160}:BLIM:REP?
:CALC{1-160}:RLIM:DATA?
:CALC{1-160}:RLIM:REP?
:SENS{1-160}:CORR:COEF
:SENS{1-160}:FREQ:DATA?
:SENS{1-160}:SEGM:DATA
:SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA
:SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA
:SOUR{1-160}:POW:PORT{1-4}:CORR:COLL:TABL:LOSS:DATA
:SOUR{1-160}:POW:PORT{1-4}:CORR:DATA

- ASCII transfer format must be specified when controlling the E5071C using SCPI commands with the Parse object in the E5071C VBA.

Variable

Parameter	*Param*
Description	Data transfer format
Data type	Character string type (String)

Range	Select from the following: "ASCii": Specifies the ASCII transfer format. "REAL": Specifies the IEEE 64-bit floating point binary transfer format. "REAL32": Specifies the IEEE 32-bit floating point binary transfer format.
Preset value	"ASCii"

Examples

```
Dim Fmt As String
SCPI.FORMat.DATA = "ASC"
Fmt = SCPI.FORMat.DATA
```

Related objects

SCPI.FORMat.BORDer

Parse

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:FORMat:DATA {ASCii|REAL|REAL32}

:FORMat:DATA?

Query response

{ASC|REAL|REAL32}<newline><^END>

Example of use

```
10 OUTPUT 717;":FORM:DATA REAL"
20 OUTPUT 717;":FORM:DATA?"
30 ENTER 717;A$
```

HCOPY

SCPI.HCOPy.ABORt

Object type

Method (**Write-only**)

Syntax

SCPI.HCOPy.ABORt

Description

This command aborts the print output.

Examples

SCPI.HCOPy.ABORt

Related objects

SCPI.HCOPy.IMMediate

Equivalent key

System > **Abort Printing**

Equivalent SCPI command

Syntax

:HCOPy:ABORt

Example of use

```
10 OUTPUT 717;":HCOP:ABOR"
```

SCPI.HCOPy.IMAGe

Object type

Property (**Read-Write**)

Syntax

SCPI.HCOPy.IMAGe = *Param*

Param = SCPI.HCOPy.IMAGe

Description

This command sets/gets the print color for output (to the printer).

Variable

Parameter	*Param*
Description	Print color for output to the printer.
Data type	Character string type (String)
Range	Select from the following: • "NORMal": Specifies printing in close color to the display color. • "INVert": Specifies printing in the inverted color of the display color.
Preset value	"INVert"

Examples

```
Dim Img As String
SCPI.HCOPy.IMAGe = "norm"
Img = SCPI.HCOPy.IMAGe
```

Related objects

SCPI.HCOPy.IMMediate

Equivalent key

System > **Invert Image**

Equivalent SCPI command

Syntax

:HCOPy:IMAGe {NORMal|INVert}

:HCOPy:IMAGe?

Query response

{NORM|INV}<newline><^END>

Example of use

```
10 OUTPUT 717;":HCOP:IMAG NORM"
20 OUTPUT 717;":HCOP:IMAG?"
30 ENTER 717;A$
```

SCPI.HCOPy.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.HCOPy.IMMediate

Description

This command outputs the display image on the LCD display to the printer connected to the E5071C.

- When printing the E5071C measurement screen, execute the VBA program with the Visual Basic editor closed. For the method, see Running a Program from the E5071C Measurement Screen.

Examples

SCPI.HCOPy.IMMediate

Related objects

SCPI.HCOPy.ABORt

SCPI.HCOPy.IMAGe

Equivalent key

[System] > **Print**

When performing the operation from the front panel, the image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **[Capture] ([System])** key is pressed) is printed. If no image is memorized in the clipboard, in the same way as the SCPI.HCOPy.IMMediate object, the image on the LCD display at the execution is memorized in the clipboard and then it is printed.

Equivalent SCPI command

Syntax

:HCOPy[:IMMediate]

Example of use

```
10 OUTPUT 717;":HCOP"
```

IEEE

SCPI.IEEE4882.CLS

Object type

Method (**Write-only**)

Syntax

SCPI.IEEE4882.CLS

Description

This command clears the following:

- Error Queue
- Status Byte Register
- Standard Event Status Register
- Operation Status Event Register
- Questionable Status Event Register
- Questionable Limit Status Event Register
- Questionable Limit Extra Status Event Register
- Questionable Limit Channel Status Event Register
- Questionable Limit Channel Extra Status Event Register
- Questionable Bandwidth Limit Status Event Register
- Questionable Bandwidth Limit Extra Status Event Register
- Questionable Bandwidth Limit Channel Status Event Register
- Questionable Bandwidth Limit Channel Extra Status Event Register
- Questionable Ripple Limit Status Event Register
- Questionable Ripple Limit Extra Status Event Register
- Questionable Ripple Limit Channel Status Event Register
- Questionable Ripple Limit Channel Extra Status Event Register

Examples

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*CLS

Example of use

```
10 OUTPUT 717;"*CLS"
```

SCPI.IEEE4882.ESE

Object type

Property (**Read-Write**)

Syntax

SCPI.IEEE4882.ESE = *Value*

Value = SCPI.IEEE4882.ESE

Description

This command sets/gets the value of the Standard Event Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Standard Event Status Enable Register
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the result of bitwise AND with 255 (0xff) is set.

Examples

```
Dim Stat As Long
SCPI.IEEE4882.ESE = 16
Stat = SCPI.IEEE4882.ESE
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*ESE <numeric>

*ESE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*ESE 16"
20 OUTPUT 717;"*ESE?"
30 ENTER 717;A
```

SCPI.IEEE4882.ESR

Object type

Property (**Read-only**)

Syntax

Value = SCPI.IEEE4882.ESR

Description

This command reads the value of the Standard Event Status Register. Execution of this command clears the register value.

Variable

Parameter	*Value*
Description	Value of the Standard Event Status Register
Data type	Long integer type (Long)
Preset value	128

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.ESR
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*ESR?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*ESR?"
20 ENTER 717;A
```

E5071C

SCPI.IEEE4882.IDN

Object type

Property (**Read-only**)

Syntax

Cont = SCPI.IEEE4882.IDN

Description

This command reads the product information (manufacturer, model number, serial number, and firmware revision number) of the E5071C.

Variable

Parameter	*Cont*
Description	Product information ("{string 1},{string 2},{string 3},{string 4}") • {string 1}: Manufacturer. Agilent Technologies is always read out. • {string 2}: Model number. E5071C is always read out. • {string 3}: Serial number (example: MY123400101). • {string 4}: Firmware revision number (example: A.07.00).
Data type	Character string type (String)

Examples

```
Dim Who As String
Who = SCPI.IEEE4882.IDN
```

Equivalent key

System > **Firmware Revision**

System > **Service Menu** > **Enable Options** > **Serial Number**

Equivalent SCPI command

Syntax

*IDN?

Query response

{string 1},{string 2},{string 3},{string 4}<newline><^END>

Example of use

2046

```
10 OUTPUT 717;"*IDN?"
20 ENTER 717;A$
```

E5071C

SCPI.IEEE4882.OPC

Object type

Property (**Read-Write**)

Syntax

(1)SCPI.IEEE4882.OPC = *Dummy*

(2)*Value* = SCPI.IEEE4882.OPC

Description

Case (1):

This command sets/gets 1 the OPC bit (bit 0) of the Standard Event Status Register when all of pending operations complete. For information on the structure of the status register, see Status Reporting System.

Variable

Case (2):

Parameter	*Value*
Description	1 returned when all pending operations are complete
Data type	Long integer type (Long)

Examples

SCPI.IEEE4882.OPC = 1

Dmy = SCPI.IEEE4882.OPC

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.ISOLation

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.THRU

SCPI.TRIGger.SEQuence.SINGle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*OPC

Example of use

2048

```
10 OUTPUT 717;"*OPC"
```

SCPI.IEEE4882.OPT

Object type

Property (**Read-only**)

Syntax

Cont = SCPI.IEEE4882.OPT

Description

This command reads the identification numbers of options installed in the E5071C.

Variable

Parameter	*Cont*
Description	Identification numbers of installed options
Data type	Character string type (String)
Note	If there is no installed option, 0 is read out.

Examples

```
Dim OptNum As String
OptNum = SCPI.IEEE4882.OPT
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*OPT?

Query response

{numeric}<newline><^END>

Example of use

```
Call viVPrintf(vi, "*OPT?" + vbLf, 0)
Call viVScanf(vi, "%t", Result)
```

SCPI.IEEE4882.RST

Object type

Method (**Write-only**)

Syntax

SCPI.IEEE4882.RST

Description

This command presets the E5071C to its default settings and is different from setting state preset with the SCPI.SYSTem.PRESet object as the continuous initiation mode (see SCPI.INITiate(Ch).CONTinuous) of channel 1 is set to OFF.

Examples

SCPI.IEEE4882.RST

Related objects

SCPI.INITiate(Ch).CONTinuous

SCPI.SYSTem.PRESet

SCPI.SYSTem.UPReset

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*RST

Example of use

```
10 OUTPUT 717;"*RST"
```

SCPI.IEEE4882.SRE

Object type

Property (**Read-Write**)

Syntax

SCPI.IEEE4882.SRE = *Value*

Value = SCPI.IEEE4882.SRE

Description

This command sets/gets the value of Service Request Enable Register.

Variable

Parameter	*Value*
Description	Value of the Service Request Enable Register
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the result of bitwise AND with 255 (0xff) is set. Note that bit 6 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.IEEE4882.SRE = 8
Stat = SCPI.IEEE4882.SRE
```

Related objects

SCPI.IEEE4882.ESE

SCPI.STATus.OPERation.ENABle

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*SRE <numeric>

*SRE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*SRE 128"
20 OUTPUT 717;"*SRE?"
30 ENTER 717;A
```

SCPI.IEEE4882.STB

Object type

Property (**Read-only**)

Syntax

Value = SCPI.IEEE4882.STB

Description

This command reads the value of Status Byte Register.

Variable

Parameter	*Value*
Description	Value of the Status Byte Register
Data type	Long integer type (Long)
Preset Value	0

Examples

```
Dim Stat As Long
Stat = SCPI.IEEE4882.STB
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*STB?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;"*STB?"
20 ENTER 717;A
```

SCPI.IEEE4882.TRG

Object type

Method (**Write-only**)

Syntax

SCPI.IEEE4882.TRG

Description

This command triggers the E5071C if the trigger source is set to GPIB/LAN (set to BUS with the SCPI.TRIGger.SEQuence.SOURce).

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.IEEE4882.TRG
```

Related objects

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*TRG

Example of use

```
10 OUTPUT 717;"*TRG"
```

SCPI.IEEE4882.WAI

Object type

Method (**Write-only**)

Syntax

SCPI.IEEE4882.WAI

Description

This command waits for the execution of all objects sent before this command to be completed.

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.TRIGger.SEQuence.SINGle
SCPI.IEEE4882.WAI
MsgBox "Done"
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

*WAI

Example of use

```
10 OUTPUT 717;"*WAI"
```

INIT

SCPI.INITiate(*Ch*).CONTinuous

Object type

Property (**Read-Write**)

Syntax

SCPI.INITiate(*Ch*).CONTinuous = *Status*

Status = SCPI.INITiate(*Ch*).CONTinuous

Description

This command turns ON/OFF the continuous initiation mode (setting by which the trigger system initiates continuously) of selected channel (*Ch*) in the trigger system. For more information on the trigger system, see Section Trigger System.

Variable

Parameter	*Status*
Description	ON/OFF of the continuous initiation mode
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the continuous initiation mode. • False or OFF: Turns OFF the continuous initiation mode.
Preset value	Varies depending on [variable (*Ch*)]

Examples

```
Dim ContMode As Boolean
SCPI.INITiate(2).CONTinuous = True
ContMode = SCPI.INITiate(2).CONTinuous
```

Related objects

SCPI.INITiate(Ch).IMMediate

Equivalent key

Trigger > **Continuous** (continuous initiation mode ON)

Trigger > **Hold** (continuous initiation mode OFF)

Equivalent SCPI command

Syntax

:INITiate{[1]-160}:CONTinuous {ON|OFF|1|0}

:INITiate{[1]-160}:CONTinuous?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":INIT1:CONT OFF"
20 OUTPUT 717;":INIT1:CONT?"
30 ENTER 717;A
```

SCPI.INITiate(*Ch*).IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.INITiate(*Ch*).IMMediate

Description

This command changes the state of each channel of selected channel (*Ch*) to the initiation state in the trigger system.

When this object is executed for a channel in the idle state in the trigger system, it goes into the initiation state immediately. Then, after measurement is executed once, it goes back to the idle state.

If this object is executed for a channel that is not in the idle state or a channel for which the continuous initiation mode is set to ON (setting by which the trigger system initiates continuously) in the trigger system, an error occurs when executed and the object is ignored. For more information on the trigger system, see Trigger System.

Examples

```
SCPI.INITiate(1).CONTinuous = False
SCPI.INITiate(1).IMMediate
```

Related objects

SCPI.INITiate(Ch).CONTinuous

Equivalent key

Trigger > **Single**

Equivalent SCPI command

Syntax

:INITiate{[1]-160}[:IMMediate]

Example of use

```
10 OUTPUT 717;":INIT1"
```

MEMORY

:MMEMory:LOAD:PROGram

No equivalent COM Commands

Syntax

:MMEMory:LOAD:PROGram <string>

Description

This command loads (or imports) a VBA project (a file with the .vba extension), a module (a file with the .bas extension), a user form (a file with the .frm extension) or a class module (a file with the .cls extension). If the specified file does not exist, an error occurs and the command is ignored.

Variables

Parameter	*String*
Description	File name in which you want to load the VBA project
Range	254 characters or less
Preset value	""

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:PROG ""Test1/Test1_01.vba"""
10 OUTPUT 717;":MMEM:LOAD:PROG ""A:Test1_01.vba"""
```

Related commands

:MMEM:STOR:PROG

Equivalent key

Macro Setup > **Load VBA Project**

:MMEMory:STORe:PROGram

No equivalent COM Commands

Syntax

:MMEMory:STORe:PROGram <string>

Description

This command saves a VBA project opened on the VBA editor into a file. The file name need to have a .vba extension. If a file with the specified file name exists, its contents are overwritten.

Variables

Parameter	*String*
Description	File name in which you want to save the VBA project
Range	254 characters or less
Preset value	""

Example of use

```
10 OUTPUT 717;":MMEM:STOR:PROG ""Test1/Test1_01.vba"""
10 OUTPUT 717;":MMEM:STOR:PROG ""D:Test1_01.vba"""
```

Related commands

:MMEM:LOAD:PROG

Equivalent key

Macro Setup > **Save VBA Project**

:MMEMory:TRANsfer

No equivalent COM Commands

Syntax

:MMEMory:TRANsfer <string>,<block>

:MMEMory:TRANsfer? <string>

Description

This command sets/gets data to/from a file on the built-in storage device of the E5071C. By reading out data with this command and writing it to a file on the external controller, file transfer from the E5071C to the external controller can be realized.

On the other hand, by reading out data from the external controller and writing it to a file on the E5071C with this command, file transfer from the external controller to the E5071C can be realized.

When you use directory names and file name, separate them with "/" (slash) or "\" (backslash). If a file with the specified file name already exists for writing or if the specified file does not exist for reading out (Query), an error occurs and the command is ignored.

Variables

Parameter	*String*
Description	File name on the built-in storage device
Range	254 characters or less

Parameter	*block*
Description	Data written on/read out from the file
Range	GPIB: 20 Mbytes or less LAN: 100 Kbytes or less

Query response

{block}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:TRAN ""Trace01.csv"",#6012345";Dat$
10 OUTPUT 717;":MMEM:TRAN? ""Trace01.csv"""
```

```
20 ENTER 717 USING "#,A";A$
30 ENTER 717 USING "#,A";Digit$
40 Img$="#,"&Digit$&"A"
50 ENTER 717 USING Img$;Byte$
60 Img$=Byte$&"A"
70 ALLOCATE Dat$[VAL(Byte$)]
80 ENTER 717 USING Img$;Dat$
```

Equivalent key

No equivalent key is available on the front panel.

SCPI.MMEMory.CATalog(*Dir*)

Object type

Property (**Read Only**)

Syntax

Cont = SCPI.MMEMory.CATalog(*Dir*)

Description

This command reads the following information on the built-in storage device of the E5071C:

- Space in use
- Available space
- Name and size of all files (including directories) in the specified directory
  - To read out the information in the root directory (folder), specify "\ " (backslash). Separate between directory names (file name) with "\ " (back slash), or "/" (slash).

Variable

Parameter	*Cont*
Description	Directory information ("{A},{B},{Name 1},,{Size 1},{Name 2},,{Size 2}, ... ,{Name N},,{Size N}") Where N is the number of all files in the specified directory and n is an integer between 1 and N. {A}: Space in use of the built-in storage device (byte). {B}: Available space of the built-in storage device (byte). {Name n}: Name of the n-th file (directory). {Size n}: Size (byte) of the n-th file (directory). Always 0 for directories.
Data type	Character string type (String)

Parameter	*Dir*
Description	Directory name whose information you want to

	read out
Data type	Character string type (String)
Range	254 characters or less

Examples

```
Dim DirCont As String
DirCont = SCPI.MMEMory.CATalog("d:\")
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:MMEMory:CATalog? <string 1>

Query response

{string 2}<newline><^END>

The format of the readout character string is as follows:

"{used_size},{free_size},{name 1},,{size 1}, ... ,{name N},,{size N}"

Where:

N is the number of all files in the specified directory and n is an integer between 1 and N.

{used_size}:

Space in use of the built-in storage device (byte).

{free_size}:

Available space of the built-in storage device (byte).

{name n}:

Name of the n-th file (directory).

{size n}:

Size (byte) of the n-th file (directory). Always 0 for directories.

Example of use

```
10 DIM A$[1000]
20 OUTPUT 717;":MMEM:CAT? ""\"""
30 ENTER 717;A$
```

SCPI.MMEMory.COPY

Object type

Method (**Write Only**)

Syntax

SCPI.MMEMory.COPY = *File*

Description

This command copies a file.

- Specify the file name with the extension. When you use directory names (folder names) and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	Indicates 2 file names (copy source and copy destination). *File(0):* Copy source file name *File(1):* Copy destination file name The index of the array starts from 0.
Data type	Variant type (Variant)
Range	254 characters or less
Note	If the specified copy source file does not exist, an error occurs when executed and the object is ignored. Notice that, if a file with the same name as the specified copy destination file name exists, its contents are overwritten.

Examples

```
SCPI.MMEMory.COPY = Array("test/state01.sta","d:test01.sta")
```

```
Dim File(1) As Variant
File(0) = "test/state01.sta"
File(1) = "d:test01.sta"
SCPI.MMEMory.COPY = File
```

Equivalent key

Save/Recall > **Save State** > **File Dialog...**

Equivalent SCPI command

Syntax

:MMEMory:COPY <string 1>,<string 2>

Example of use

```
10 OUTPUT 717;":MMEM:COPY ""Test1/State01.sta"","""d:Test1_01.sta"""""
```

SCPI.MMEMory.DELete

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.DELete = *File*

Description

This command deletes an existing file or directory (folder).

- When you delete a directory, all the files and directories in it are deleted.

  Specify the file name with the extension.

  When you specify a file (directory) under an existing directory, separate them with "\ " (back slash), or "/" (slash).

  To delete all files in the directory (folder), specify "\ " (backslash).

Variable

Parameter	*File*
Description	File name or directory name you want to delete
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file or directory does not exist, an error occurs when executed and the object is ignored.

Examples

SCPI.MMEMory.DELete = "d:\"

SCPI.MMEMory.DELete = "test/state01.sta"

Equivalent key

Save/Recall > **Save State** > **File Dialog...**

Equivalent SCPI command

Syntax

:MMEMory:DELete <string>

Example of use

```
10 OUTPUT 717;":MMEM:DEL ""Test1/State01.sta"""
10 OUTPUT 717;":MMEM:DEL ""D:State01.sta"""
```

SCPI.MMEMory.LOAD.ASCFactor

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.ASCFactor = *File*

Description

This command recalls the file (file with the ".csv" extension saved with the SCPI.MMEMory.STORe.ASCFactor object) you want to specify as the table for the reference calibration coefficient and calibration coefficient table for power sensor A.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash). (No read)

Variable

Parameter	*File*
Description	A file name (extension ".csv") of the reference calibration coefficient and the calibration coefficient table for power sensor A.
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples

SCPI.MMEMory.LOAD.ASCFactor = "d:\sensor01.csv"

SCPI.MMEMory.LOAD.ASCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEMory.STORe.ASCFactor

Equivalent key

Cal > **Power Calibration** > **Sensor A Settings** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:ASCFactor <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:ASCF ""Test1/Sensor01.csv"""
10 OUTPUT 717;":MMEM:LOAD:ASCF ""D:Sensor01.csv"""
```

SCPI.MMEMory.LOAD.BSCFactor

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.BSCFactor = *File*

Description

This command recalls the file (file with the ".csv" extension saved with the SCPI.MMEMory.STORe.BSCFactor) you want to specify as the table for the reference calibration coefficient and the calibration coefficient table for power sensor B.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	A file name (extension ".csv") of the reference calibration coefficient and the calibration coefficient table for power sensor B.
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples

SCPI.MMEMory.LOAD.BSCFactor = "d:\sensor01.csv"

SCPI.MMEMory.LOAD.BSCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEMory.STORe.BSCFactor

Equivalent key

Cal > **Power Calibration** > **Sensor B Settings** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:BSCFactor <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:BSCF ""Test1/Sensor01.csv"""
10 OUTPUT 717;":MMEM:LOAD:BSCF ""D:Sensor01.csv"""
```

SCPI.MMEMory.LOAD.CHANnel.STATe

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.CHANnel.STATe = *Register*

Description

This command recalls the instrument state for an individual channel (saved with the SCPI.MMEMory.STORe.CHANnel.STATe object) from the specified register as the setting of the active channel.

Variable

Parameter	*Register*
Description	Register
Data type	Character string type (String)
Range	Select from the following: "A": Specifies register A. "B": Specifies register B. "C" :Specifies register C. "D" :Specifies register D.
Note	If no instrument state has been saved in the specified register, an error occurs and the object is ignored.

Examples

```
SCPI.MMEMory.LOAD.CHANnel.STATe = "a"
```

Related objects

SCPI.MMEMory.STORe.CHANnel.STATe

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Save/Recall > **Recall Channel** > **A**|**B**|**C**|**D**

Equivalent SCPI command

Syntax

```
:MMEMory:LOAD:CHANnel[:STATe] {A|B|C|D}
```

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:CHAN A"
```

SCPI.MMEMory.LOAD.CKIT(*Ckit*)

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.CKIT(*Ckit*) = *File*

Description

This command recalls the definition file of the specified calibration kit (file with the ".ckx" extension saved with the SCPI.MMEMory.STORe.CKIT(Ckit)).

Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*Ckit*
Description	Number of calibration kit
Data type	Long integer type (Long)
Range	1 to 30
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*File*
Description	File name of the definition table of a calibration kit
Data type	Character string type (String)
Range	254 characters or less

Note	If the specified file does not exist, a runtime error occurs.

Examples

SCPI.MMEMory.LOAD.CKIT(1) = "Test1/Ckit01.ckx"

SCPI.MMEMory.LOAD.CKIT(1) = "A:\Ckit01.ckx"

Related objects

SCPI.MMEMory.STORe.CKIT(Ckit)

Equivalent key

Cal > **Modify Cal Kit** > **Import Cal Kit...**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:CKIT{[1]-30} <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:CKIT1 ""Test1/Ckit01.ckx"""
10 OUTPUT 717;":MMEM:LOAD:LIM ""A:Ckit01.ckx"""
```

SCPI.MMEMory.LOAD.LIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.LIMit = *File*

Description

This command recalls the specified limit table file (file with the .csv extension saved with SCPI.MMEMory.STORe.LIMit), from the limit table for the active trace of the active channel.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.LIMit = "d:\limit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.LIMit = "test/limit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.STORe.LIMit

Equivalent key

Analysis > **Limit Test** > **Edit Limit Line** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:LIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:LIM ""Test1/Limit01.csv"""
10 OUTPUT 717;":MMEM:LOAD:LIM ""D:Limit01.csv"""
```

SCPI.MMEMory.LOAD.PLOSs(*Pt*)

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.PLOSs(*Pt*) = *File*

Description

This command recalls the specified loss compensation table file (a file with the ".csv" extension saved with SCPI.MMEMory.STORe.PLOSs(Pt)), as the loss compensation table for the active channel, for the selected port (*Pt*).

Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of the loss compensation table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.PLOSs(1) = "d:\loss01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.PLOSs(1) = "test/loss01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.PLOSs(Pt)

Equivalent key

Cal > **Power Calibration** > **Loss Compen** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:PLOSs{[1]|2|3|4} <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:PLOS1 ""Test1/Loss01.csv"""
10 OUTPUT 717;":MMEM:LOAD:PLOS1 ""D:Loss01.csv"""
```

SCPI.MMEMory.LOAD.RLIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.RLIMit = File

Description

This command recalls the specified ripple limit table file (file with the .csv extension saved with SCPI.MMEMory.STORe.RLIMit) of the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command), as ripple limit table for the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect).

- Specify the file name with the extension. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).
  If the specified file does not exist, an error occurs and the command is ignored.

Variable

Parameter	*File*
Description	File name of the ripple limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.RIMit = "D:\Rlimit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTive
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.LOAD.RLIMit = "test/Rlimit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

E5071C

SCPI.MMEMory.STORe.RLIMit

Equivalent key

Analysis > **Ripple Limit** > **Edit Ripple Line** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:RLIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:RLIM ""RTest1/Rlim01.csv"""
10 OUTPUT 717;":MMEM:LOAD:RLIM ""D:Rlim01.csv"""
```

SCPI.MMEMory.LOAD.SEGMent

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.SEGMent = *File*

Description

This command recalls the specified segment sweep table file (file with a .csv extension saved with the SCPI.MMEMory.STORe.SEGMent), as the segment sweep table of the active channel.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of segment sweep table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.SEGMent = "d:\segm01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.LOAD.SEGMent = "test/segm01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SEGMent

Equivalent key

Sweep Setup > **Edit Segment Table** > **Import from CSV File**

Equivalent SCPI command

Syntax

:MMEMory:LOAD:SEGMent <string>

Example of use

```
10 OUTPUT 717;":MMEM:LOAD:SEGM ""Test1/Segm01.csv"""
10 OUTPUT 717;":MMEM:LOAD:SEGM ""D:Segm01.csv"""
```

SCPI.MMEMory.LOAD.STATe

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.LOAD.STATe = *File*

Description

This command recalls the specified instrument state file (file with a .sta extension saved with SCPI.MMEMory.STORe.STATe).

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name of instrument state (extension ".sta")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, an error occurs when executed and the object is ignored.

Examples

SCPI.MMEMory.LOAD.STATe = "d:\state01.sta"

SCPI.MMEMory.LOAD.STATe = "test/state01.sta"

Related objects

SCPI.MMEMory.STORe.STATe

Equivalent key

Save/Recall > **Recall State**

Equivalent SCPI command

Syntax

:MMEMory:LOAD[:STATe] <string>

Example of use

10 OUTPUT 717;":MMEM:LOAD ""Test1/State01.sta"""

```
10 OUTPUT 717;":MMEM:LOAD ""d:State01.sta"""
```

SCPI.MMEMory.MDIRectory

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.MDIRectory = *File*

Description

This command creates a new directory (folder).

- When you create a directory under an existing directory, separate between the directory names with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	Directory name you want to create
Data type	Character string type (String)
Range	254 characters or less
Note	If a directory with the same name as the specified directory name exists, an error occurs when executed and the object is ignored.

Examples

SCPI.MMEMory.MDIRectory = "d:\test"

SCPI.MMEMory.MDIRectory = "test"

Equivalent key

Save/Recall > **Save State** > **File Dialog...**

Equivalent SCPI command

Syntax

:MMEMory:MDIRectory <string>

Example of use

10 OUTPUT 717;":MMEM:MDIR ""Test1"""

10 OUTPUT 717;":MMEM:MDIR ""d:Test1"""

SCPI.MMEMory.STORe.ASCFactor

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.ASCFactor = *File*

Description

This command saves the reference calibration coefficient and the calibration coefficient table for power sensor A into a CSV file (extension ".csv").

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	A file name (extension ".csv") to save the reference calibration coefficient and the calibration coefficient table for power sensor A.
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

SCPI.MMEMory.STORe.ASCFactor = "d:\sensor01.csv"

SCPI.MMEMory.STORe.ASCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEMory.LOAD.ASCFactor

Equivalent key

Cal > **Power Calibration** > **Sensor A Settings** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:ASCFactor <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:ASCF ""Test1/Sensor01.csv"""
10 OUTPUT 717;":MMEM:STOR:ASCF ""D:Sensor01.csv"""
```

SCPI.MMEMory.STORe.BSCFactor

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.BSCFactor = *File*

Description

This command saves the reference calibration coefficient and the calibration coefficient table for power sensor B into a CSV file (extension ".csv").

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	A file name (extension ".csv") to save the reference calibration coefficient and the calibration coefficient table for power sensor B.
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

SCPI.MMEMory.STORe.BSCFactor = "d:\sensor01.csv"

SCPI.MMEMory.STORe.BSCFactor = "test/sensor01.csv"

Related objects

SCPI.MMEMory.LOAD.BSCFactor

Equivalent key

Cal > **Power Calibration** > **Sensor B Settings** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:BSCFactor <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:BSCF ""Test1/Sensor01.csv"""
10 OUTPUT 717;":MMEM:STOR:BSCF ""D:Sensor01.csv"""
```

SCPI.MMEMory.STORe.CHANnel.CLEar

Object type

Method (**Write-only**)

Syntax

SCPI.MMEMory.STORe.CHANnel.CLEar

Description

This command deletes the instrument state for each channel (saved with the SCPI.MMEMory.STORe.CHANnel.STATe object) in all the registers.

Examples

SCPI.MMEMory.STORe.CHANnel.CLEar

Related objects

SCPI.MMEMory.STORe.CHANnel.STATe

Equivalent key

Save/Recall > **Save Channel** > **Clear States** > **OK**

Equivalent SCPI command

Syntax

:MMEMory:STORe:CHANnel:CLEar

Example of use

```
10 OUTPUT 717;":MMEM:STOR:CHAN:CLE"
```

SCPI.MMEMory.STORe.CHANnel.STATe

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.CHANnel.STATe = *Register*

Description

This command saves the instrument state of the items set for the active channel specific to that channel only into the specified register (volatile memory).

Variable

Parameter	*Register*
Description	Register
Data type	Character string type (String)
Range	Select from the following: "A": Specifies register A. "B": Specifies register B. "C": Specifies register C. "D": Specifies register D.
Note	If an instrument state has been saved already in the specified register, its contents are overwritten.

Examples

SCPI.MMEMory.STORe.CHANnel.STATe = "a"

Related objects

SCPI.MMEMory.LOAD.CHANnel.STATe

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

Save/Recall > **Save Channel** > **A|B|C|D**

Equivalent SCPI command

Syntax

:MMEMory:STORe:CHANnel[:STATe] {A|B|C|D}

Example of use

```
10 OUTPUT 717;":MMEM:STOR:CHAN A"
```

SCPI.MMEMory.STORe.CKIT(*Ckit*)

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.CKIT(*Ckit*) = *File*

Description

This command saves the definition table of the calibration kit to a file.

- Specify the file name with the .ckx extension. When you use a directory name and file name, separate them with "/" (slash) or "\" (backslash).

  If a file with the specified file name exists, its contents are overwritten.

Variable

Parameter	*File*
Description	A file name used to save the definition of the calibration kit.
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

- For information on the variable (*Ckit*), see Val Ckit.

Examples

SCPI.MMEMory.STORe.CKIT(1) = "d:\Ckit01.ckx"

SCPI.MMEMory.STORe.CKIT(1) = "test/trace01.csv"

Related objects

SCPI.MMEMory.LOAD.CKIT(Ckit)

Equivalent key

Cal > **Modify Cal Kit** > **Export Cal Kit...**

Equivalent SCPI command

Syntax

:MMEMory:STORe:CKIT{[1]-30} <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:CKIT1 ""Test1/Ckit01.ckx"""
10 OUTPUT 717;":MMEM:STOR:CKIT1 ""d:Ckit01.ckx"""
```

SCPI.MMEMory.STORe.FDATa

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.FDATa = *File*

Description

This command saves the formatted data array into a file in the CSV format (extension ".csv"), for the active trace of the active channel.

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name in which you want to save the formatted data array (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.FDATa = "d:\trace01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.FDATa = "test/trace01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

Save/Recall > **Save Trace Data**

Equivalent SCPI command

Syntax

:MMEMory:STORe:FDATa <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:FDAT ""Result/Trace01.csv"""
10 OUTPUT 717;":MMEM:STOR:FDAT ""D:Trace01.csv"""
```

SCPI.MMEMory.STORe.IMAGe

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.IMAGe = *File*

Description

This command saves the display image on the LCD display at the execution of the object into a file in the bitmap (extension ".bmp") or portable network graphics (extension ".png") format.

- When saving the E5071C measurement screen, execute the VBA program with the Visual Basic editor closed. For more information, see Running a Program from the E5071C Measurement Screen.

  Specify the file name with the extension When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name in which you want to save the display image on the LCD display (extension ".bmp" or ".png")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.MMEMory.STORe.IMAGe = "d:\image01.bmp"
SCPI.MMEMory.STORe.IMAGe = "test/image01.png"
```

Equivalent key

System > **Dump Screen Image**

When performing the operation from the front panel, the image on the LCD display memorized in the volatile memory (clipboard) (the image on the LCD display when the **Capture** **(System)** key is pressed) is saved.

If no image is memorized in the clipboard, in the same way as the SCPI.MMEMory.STORe.IMAGe object, the image on the LCD display at the execution is memorized in the clipboard and then it is saved.

Equivalent SCPI command

Syntax

:MMEMory:STORe:IMAGe <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:IMAG ""Result/Image01.bmp"""
10 OUTPUT 717;":MMEM:STOR:IMAG ""D:Image01.png"""
```

SCPI.MMEMory.STORe.LIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.LIMit = *File*

Description

This command saves the limit table of the active trace of the active channel into a file in the CSV format (extension ".csv").

- Specify the file name with the extension.When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name to save the limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.LIMit = "d:\limit01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.LIMit = "test/limit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.LOAD.LIMit

Equivalent key

Analysis > **Limit Test** > **Edit Limit Line** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:LIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:LIM ""Test1/Limit01.csv"""
10 OUTPUT 717;":MMEM:STOR:LIM ""D:Limit01.csv"""
```

SCPI.MMEMory.STORe.PLOSs(*Pt*)

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.PLOSs(*Pt*) = *File*

Description

This command saves the loss compensation table of the active channel into a file in the CSV format (extension ".csv"), for the selected port (*Pt*).

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	A file name to save the loss compensation table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.PLOSs(1) = "d:\loss01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.PLOSs(1) = "test/loss01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.LOAD.PLOSs(Pt)

Equivalent key

Cal > **Power Calibration** > **Loss Compen** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:PLOSs{[1]|2|3|4} <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:PLOS1 ""Test1/Loss01.csv"""
10 OUTPUT 717;":MMEM:STOR:PLOS1 ""D:Loss01.csv"""
```

SCPI.MMEMory.STORe.RLIMit

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.RLIMit = File

Description

This command saves the ripple limit table of the active trace (specified with the SCPI.CALCulate(Ch).PARameter(Tr).SELect command) of the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the CSV format.

- Specify the file name with the .sta extension. When you write directory names and file name, separate them with "/" (slash) or "\" (backslash).

  If a file with the specified file name already exists, its contents will be overwritten.

Variable

Parameter	*File*
Description	File name used to save the ripple limit table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If the specified file does not exist, a runtime error occurs.

Examples (1)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.RLIMit = "D:\Rlimit01.csv"
```

Examples (2)

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.CALCulate(1).PARameter(1).SELect
SCPI.MMEMory.STORe.RLIMit = "test/Rlimit01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

E5071C

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.MMEMory.LOAD.RLIMit

Equivalent key

Analysis > **Ripple Limit** > **Edit Ripple Line** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:RLIMit <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:RLIM ""RTest1/Rlim01.csv"""
10 OUTPUT 717;":MMEM:STOR:RLIM ""D:Rlim01.csv"""
```

SCPI.MMEMory.STORe.SALL

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SALL = *Status*

Status = SCPI.MMEMory.STORe.SALL

Description

This command sets/gets whether to save the settings of all channels/traces or that of the displayed channels/traces only, as the instrument state to be saved.

Variable

Parameter	*Status*
Description	Selecting content to be saved as the instrument state setting.
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Specifies the setting of all channels/traces as the target to be saved. • False or OFF: Specifies the setting of displayed channels/traces only as the target to be saved.
Preset value	False or OFF

Examples

```
Dim Obj As Boolean
SCPI.MMEMory.STORe.SALL = True
Obj = SCPI.MMEMory.STORe.SALL
```

Related objects

SCPI.MMEMory.STORe.STATe

Equivalent key

Save/Recall > **Channel/Trace**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SALL {ON|OFF|1|0}

:MMEMory:STORe:SALL?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SALL ON"
20 OUTPUT 717;":MMEM:STOR:SALL?"
30 ENTER 717;A
```

SCPI.MMEMory.STORe.SEGMent

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.SEGMent = *File*

Description

This command saves the segment sweep table of the active channel into a file in the CSV format (extension ".csv").

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name to save segment sweep table (extension ".csv")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SEGMent = "d:\segm01.csv"
```

```
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SEGMent = "test/segm01.csv"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.LOAD.SEGMent

Equivalent key

Sweep Setup > **Edit Segment Table** > **Export to CSV File**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SEGMent <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SEGM ""Test1/Segm01.csv"""
10 OUTPUT 717;":MMEM:STOR:SEGM ""D:Segm01.csv"""
```

SCPI.MMEMory.STORe.SNP.DATA

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.SNP.DATA = *File*

Description

Saves the measurement data for the active channel (specified with the SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

You need to specify a file format and file type before saving a file. The extension differs depending on file types as shown in the followig table:

<file type>	<extension>
When specifying one port	s1p
When specifying two ports	s2p
When specifying three ports	s3p
When specifying four ports	s4p

- When you use directory names and file name, separate them with "/" (slash) or "\" (back slash).

  If a file with the specified file name already exists, its contents are overwritten.

Variable

Parameter	*File*
Description	File name you want to use when saving file in the touchstone format
Range	254 characters or less

- When invalid extension is specified, an error message appears and the command is ignored.

Examples

```
Dim SnpPorts(2) As Variant
SCPI.DISPlay.WINDow(1).ACTivate
SCPI.MMEMory.STORe.SNP.FORMat = "RI"
SnpPorts(0) = 1
SnpPorts(1) = 2
SnpPorts(2) = 4
SCPI.MMEMory.STORe.SNP.TYPE.S3P = SnpPorts
SCPI.MMEMory.STORe.SNP.DATA = "SNP01.s3p"
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.FORMat

SCPI.MMEMory.STORe.SNP.TYPE.S1P

SCPI.MMEMory.STORe.SNP.TYPE.S2P

SCPI.MMEMory.STORe.SNP.TYPE.S3P

SCPI.MMEMory.STORe.SNP.TYPE.S4P

Equivalent key

After a file type is specified, a dialog box will appear.

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP[:DATA] <string>

Example of use

```
10 OUTPUT 717;"DISP:WIND1:ACT"
20 OUTPUT 717;":MMEM:STOR:SNP:FORM RI"
30 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P 1,2,4"
40 OUTPUT 717;":MMEM:STOR:SNP ""SNP01.s3p"""
```

SCPI.MMEMory.STORe.SNP.FORMat

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.FORMat = *Param*

Param = SCPI.MMEMory.STORe.SNP.FORMat

Description

This command sets/gets the data format for saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Param*
Description	Touchstone file format
Data type	Character string type (String)
Range	Select from the following: "AUTO": Specifies data format automatically according to the display format of the active trace. "MA": Specifies data format "log magnitude > angle". "DB": Specifies data format "linear magnitude > angle". "RI": Specifies data format "real part > imaginary part".
Preset value	"AUTO"

Examples

```
Dim Fmt As String
SCPI.MMEMory.STORe.SNP.FORMat = "MA"
Fmt = SCPI.MMEMory.STORe.SNP.FORMat
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

Equivalent key

Save/Recall > **Save Snp > Snp Format > AUTO|LogMag/Angle|LinMag/Angle|Real/Imaginary**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:FORMat {AUTO|MA|DB|RI}

:MMEMory:STORe:SNP:FORMat?

Query response

{AUTO|MA|DB|RI}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:FORM MA"
20 OUTPUT 717;":MMEM:STOR:SNP:FORM?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.SNP.TYPE.S1P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.TYPE.S1P = *Port*

Port = SCPI.MMEMory.STORe.SNP.TYPE.S1P

Description

This command sets/gets the specified port to the file type (1 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Port*
Description	Port number
Range	1 to 4
Resolution	1

Examples

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P 2"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P?"
30 ENTER 717;A$
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save SnP** > **S1P** > **1**|**2**|**3**|**4**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S1P <numeric>

:MMEMory:STORe:SNP:TYPE:S1P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P 2"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S1P?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.SNP.TYPE.S2P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.TYPE.S2P = *Ports*

Ports = SCPI.MMEMory.STORe.SNP.TYPE.S2P

Description

This command sets/gets the specified port to the file type (2 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for file type. • *Ports(1):* Specifies the other port for file type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two ports, an error occurs during execution. The order of the two port numbers to be specified is arbitrary.

Examples

```
Dim Ports(1) As Long
Ports(0) = 2
Ports(1) = 3
SCPI.MMEMory.STORe.SNP.TYPE.S2P = Ports
Ports = SCPI.MMEMory.STORe.SNP.TYPE.S2P
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S2p** > **1-2**|**1-3**|**1-4**|**2-3**|**2-4**|**3-4**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S2P <numeric1>, <numeric 2>

:MMEMory:STORe:SNP:TYPE:S2P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P 2,3"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S2P?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.SNP.TYPE.S3P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.TYPE.S3P = *Ports*

Ports = SCPI.MMEMory.STORe.SNP.TYPE.S3P

Description

This command sets/gets the specified port to the file type (3 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Ports*
Description	Indicates 3-element array data (port number). • *Ports(0)* :Specifies a port for file type. • *Ports(1)* :Specifies a port for file type. • *Ports(2)* :Specifies a port for file type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two port numbers, an error occurs during execution. The order of the three port numbers to be specified is arbitrary.

Examples

```
Dim Ports(2) As Long
Ports(0) = 2
Ports(1) = 3
Ports(2) = 4
SCPI.MMEMory.STORe.SNP.TYPE.S3P = Ports
Ports = SCPI.MMEMory.STORe.SNP.TYPE.S3P
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S3p** > **1-2-3**|**1-2-4**|**1-3-4**|**2-3-4**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S3P <numeric 1>, <numeric 2>, <numeric 3>

:MMEMory:STORe:SNP:TYPE:S3P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P 2,3,4"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.SNP.TYPE.S4P

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.SNP.TYPE.S4P = *Ports*

Ports = SCPI.MMEMory.STORe.SNP.TYPE.S4P

Description

This command sets/gets the specified port to the file type (4 port) when saving measurement data for the active channel (specified with SCPI.DISPlay.WINDow(Ch).ACTivate command) into a file in the touchstone format.

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port number). • *Ports(0)* :Specifies a port for file type. • *Ports(1)* :Specifies a port for file type. • *Ports(2)* :Specifies a port for file type. • *Ports(3)* :Specifies a port for file type. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when it is executed. If you specify the same port number to two port numbers, an error occurs during execution. The order of the four port numbers to be specified is arbitrary.

Examples

```
Dim Ports(3) As Long
Ports(0) = 1
Ports(1) = 2
Ports(2) = 3
```

Ports(3) = 4
SCPI.MMEMory.STORe.SNP.TYPE.S4P = Ports
Ports = SCPI.MMEMory.STORe.SNP.TYPE.S4P

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

SCPI.MMEMory.STORe.SNP.DATA

SCPI.MMEMory.STORe.SNP.FORMat

Equivalent key

Save/Recall > **Save Snp** > **S4p** > **1-2-3-4**

Equivalent SCPI command

Syntax

:MMEMory:STORe:SNP:TYPE:S4P 1, 2, 3, 4

:MMEMory:STORe:SNP:TYPE:S4P?

Example of use

```
10 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S4P 1,2,3,4"
20 OUTPUT 717;":MMEM:STOR:SNP:TYPE:S3P?"
30 ENTER 717;A$
```

SCPI.MMEMory.STORe.STATe

Object type

Property (**Write Only**)

Syntax

SCPI.MMEMory.STORe.STATe = *File*

Description

This command saves the instrument state (contents to be saved specified with the SCPI.MMEMory.STORe.STYPe object) into a file (file with the .sta extension).

- Specify the file name with the extension. When you use directory names and file name, separate them with "\ " (back slash), or "/" (slash).

Variable

Parameter	*File*
Description	File name to save the instrument state (extension ".sta")
Data type	Character string type (String)
Range	254 characters or less
Note	If a file with the same name as the specified file name exists, its contents are overwritten.

Examples

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
SCPI.MMEMory.STORe.STATe = "d:\state01.sta"
```

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
SCPI.MMEMory.STORe.STATe = "test/state01.sta"
```

Related objects

SCPI.MMEMory.STORe.STYPe

SCPI.MMEMory.LOAD.STATe

Equivalent key

Save/Recall > **Save State**

Equivalent SCPI command

Syntax

:MMEMory:STORe[:STATe] <string>

Example of use

```
10 OUTPUT 717;":MMEM:STOR ""Test1/State01.sta"""
10 OUTPUT 717;":MMEM:STOR ""D:State01.sta"""
```

SCPI.MMEMory.STORe.STYPe

Object type

Property (**Read-Write**)

Syntax

SCPI.MMEMory.STORe.STYPe = *Param*

Param = SCPI.MMEMory.STORe.STYPe

Description

Selects the contents saved when saving the instrument state into a file with the SCPI.MMEMory.STORe.STATe object.

Variable

Parameter	*Param*
Description	Data of instrument state
Data type	Character string type (String)
Range	Select from the following. • "STATe": Specifies the save of the measurement conditions only. • "CSTate":Specifies the save of the measurement conditions and the calibration state. • "DSTate": Specifies the save of the measurement conditions and the formatted data array. • "CDSTate": Specifies the save of the measurement conditions, the calibration state, and the formatted data array.
Preset value	"CSTate"

Examples

```
Dim StaType As String
SCPI.MMEMory.STORe.STYPe = "cdst"
StaType = SCPI.MMEMory.STORe.STYPe
```

Related objects

SCPI.MMEMory.STORe.STATe

Equivalent key

Save/Recall > Save Type > State Only|State & Cal|State & Trace|All

Equivalent SCPI command

Syntax

:MMEMory:STORe:STYPe {STATe|CSTate|DSTate|CDSTate}

:MMEMory:STORe:STYPe?

Query response

{STAT|CST|DST|CDST}<newline><^END>

Example of use

```
10 OUTPUT 717;":MMEM:STOR:STYP CDST"
20 OUTPUT 717;":MMEM:STOR:STYP?"
30 ENTER 717;A$
```

OUTPUT

SCPI.OUTPut.STATe

Object type

Property (**Read Write**)

Syntax

SCPI.OUTPut.STATe = *Status*

Status = SCPI.OUTPut.STATe

Description

This command turns on/off of the stimulus signal output. Measurement cannot be made until the stimulus signal output is turned ON.

Variable

Parameter	*Status*
Description	On/off of the stimulus signal output
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns on the stimulus signal. • False or OFF: Turns off the stimulus signal.
Preset value	True or ON

Examples

```
Dim Outp As Boolean
SCPI.OUTPut.STATe = True
Outp = SCPI.OUTPut.STATe
```

Equivalent key

Sweep Setup > **Power** > **RF Out**

Equivalent SCPI command

Syntax

:OUTPut[:STATe] {ON|OFF|1|0}

:OUTPut[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":OUTP ON"
20 OUTPUT 717;":OUTP?"
30 ENTER 717;A
```

PROGRAM

:PROGram:CATalog?

No equivalent COM Commands

Syntax

:PROGram:CATalog?

Description

This command reads the list of all executable VBA macros (procedures defined by Public including the VBA project loaded on the VBA editor).

Query response

{string}<newline><^END>

The character string in the following format, in which each macro is separated by a comma (.), is read out.

"{macro 1},{macro 2}, ... ,{macro N}"

Where N is the total number of VBA macros.

{macro n}: VBA macro name (module name.procedure name)

Example of use

```
10 DIM A$[1000]
20 OUTPUT 717;":PROG:CAT?"
30 ENTER 717;A$
```

:PROGram[:SELected]:NAME

No equivalent COM Commands

Syntax

:PROGram[:SELected]:NAME <string>

:PROGram[:SELected]:NAME?

Description

This command sets/gets the VBA macro controlled with the :PROG:STAT command.

- Selectable VBA macro names can be read with the :PROG:CAT? command.

Variable

Parameter	*String*
Description	VBA macro name (module name.procedure name)
Range	254 characters or less
Preset value	""

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":PROG:NAME ""Module1.main"""
20 OUTPUT 717;":PROG:NAME?"
30 ENTER 717;A$
```

Related commands

:PROG:CAT?

:PROG:STAT

Equivalent key

Macro Setup > **Select Macro**

- When performing a similar operation from the front panel, you should select the VBA macro and execute it at the same.

:PROGram[:SELected]:STATe

No equivalent COM Commands

Syntax

:PROGram[:SELected]:STATe {STOP|RUN}

:PROGram[:SELected]:STATe?

Description

This command sets/gets the control/state of the VBA macro selected with the :PROG:STAT command.

Variable

Range	STOP: Stop the program. RUN: Run the Program.
Preset value	STOP

Query response

{STOP|RUN}<newline><^END>

Example of use

```
10 OUTPUT 717;":PROG:STAT RUN"
20 OUTPUT 717;":PROG:STAT?"
30 ENTER 717;A$
```

Related commands

:PROG:NAME

Equivalent key

Macro Break (to stop)

Macro Setup > **Select Macro** (to run)

- When performing the operation from the front panel, you select the VBA macro and execute it at the same time.

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.ARRay(Vnum).DATA = Data

Data = SCPI.PROGram.VARiable.ARRay(Vnum).DATA

Description

This command sets/gets an array of Data that can be exchanged between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Data*
Description	"n" is the number obtained from the value specified with the SCPI.PROGram.VARiable.ARRay(Vnum).SIZE object. • Data(0): The first array data • Data(n-1): The n-th array data The index of the array starts from 0.
Data Type	Variant type (Variant)
Note	If the data size is not specified, an error occurs when executed.

Examples

```
Dim Var1(2) As Long
Dim Var2 as Variant
Dim ArraySize as Long
ArraySize=3
```

```
Var1(1) = 2
Var1(2) = 56
SCPI.PROGram.VARiable.ARRay(1).SIZE = ArraySize
SCPI.PROGram.VARiable.ARRay(1).DATA = Var1
Var2= SCPI.PROGram.VARiable.ARRay(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:ARRay{[1] - 10}:DATA <Data Array>

:PROGram:VARiable:ARRay{[1] - 10}:DATA ?

Example of use

```
10 OUTPUT 717;"PROG:VAR:ARR2:SIZE 4"
20 OUTPUT 717;"PROG:VAR:ARR2 1.0,2.0,3.0,4.0"
30 OUTPUT 717;"PROG:VAR:ARR2?"
40 ENTER 717;A(*)
```

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE = Value

Value = SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

Description

This command sets/gets the size of an array of Data, specified by SCPI.PROGram.VARiable.ARRay(1-10).DATA, that can be exchanged between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of data size
Data Type	Long integer type (long)
Range	1 to 40002
Preset value	402
Resolution	1

Examples

```
Dim Var1(2) As Long
Dim Var2 as Variant
```

```
Dim ArraySize as Long
ArraySize=3
Var1(1) = 2
Var1(2) = 56
SCPI.PROGram.VARiable.ARRay(1).SIZE = ArraySize
SCPI.PROGram.VARiable.ARRay(1).DATA = Var1
Var2= SCPI.PROGram.VARiable.ARRay(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:ARRay{[1] - 10}:SIZE <Data Array Size>

:PROGram:VARiable:ARRay{[1] - 10}:SIZE ?

Example of use

```
10 OUTPUT 717;"PROG:VAR:ARR2:SIZE 4"
20 OUTPUT 717;"PROG:VAR:ARR2 1.0,2.0,3.0,4.0"
30 OUTPUT 717;"PROG:VAR:ARR2?"
40 ENTER 717;A(*)
```

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA = Value

Value = SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

Description

This command can be used to exchange Double type data between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of the double precision floating point type
Data type	Double precision floating point type (Double)
Range	Compliant with the double precision floating point type
Preset value	0

Examples

```
Dim DblVal As Double
DblVal =55.7890

SCPI.PROGram.VARiable.DOUBle(1).DATA = DblVal
DblVal= SCPI.PROGram.VARiable.DOUBle(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.LONG(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:DOUBle{[1]-10}:DATA <Numeric value, Double Type>

:PROGram:VARiable:DOUBle{[1]-10}:DATA ?

Example of use

```
10 OUTPUT 717;"PROGram:VARiable:DOUBle1:DATA 12345.89607"
20 OUTPUT 717;":PROGram:VARiable:DOUBle1:DATA?"
30 ENTER 717;A$
```

SCPI.PROGram.VARiable.LONG(Vnum).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.LONG(Vnum).DATA = Value

Value = SCPI.PROGram.VARiable.LONG(Vnum).DATA

Description

This command can be used to exchange Long type data between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

The SCPI.PROGram.VARiable.LONG(Vnum).DATA command accepts the values from -2,147,483,648 to 2,147,483,647 as a long type variable. However, the range of scpi command (:PROG:VAR:LONG:DATA) is -65536 to 65536. Therefore, you can exchange the value from -65536 to 65536. If you set the value out of the range by the SCPI.PROGram.VARiable.LONG(Vnum).DATA command, the returned value of :PROG:VAR:LONG:DATA is -65536 or 65536.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of the long integer type
Data type	Long integer type (Long)
Range	-65536 to 65536
Preset value	0

Examples

```
Dim LngVal As Long
LngVal = 65536

SCPI.PROGram.VARiable.LONG(1).DATA = LngVal
LngVal= SCPI.PROGram.VARiable.LONG(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:LONG{[1] - 10}:DATA <Numeric value>

:PROGram:VARiable:LONG{[1] - 10}:DATA?

Example of use

```
10 OUTPUT 717;":PROGram:VARiable:LONG1:DATA 65536"
20 OUTPUT 717;":PROGram:VARiable:LONG1:DATA?"
30 ENTER 717;A
```

SCPI.PROGram.VARiable.STRing(Vnum).DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.PROGram.VARiable.STRing(Vnum).DATA = Value

Value = SCPI.PROGram.VARiable.STRing(Vnum).DATA

Description

This command can be used to exchange String type data between an external PC and E5071C built-in VBA using GPIB/LAN/USB.

Variable

Parameter	*Vnum*
Description	Variable Number
Range	1 to 10
Preset value	1

Parameter	*Value*
Description	The value of the character string type
Data type	Character string type (String)
Range	Compliant with the character string type
Preset value	""

Examples

```
Dim StrVal As String
StrVal = "TestS11"

SCPI.PROGram.VARiable.STRing(1).DATA = StrVal
StrVal= SCPI.PROGram.VARiable.String(1).DATA
```

Related objects

SCPI.PROGram.VARiable.ARRay(Vnum).DATA

SCPI.PROGram.VARiable.ARRay(Vnum).SIZE

SCPI.PROGram.VARiable.DOUBle(Vnum).DATA

SCPI.PROGram.VARiable.LONG(Vnum).DATA

Equivalent key

None

Equivalent SCPI command

Syntax

:PROGram:VARiable:STRing{[1] - 10}:DATA <Character value, String Type>

:PROGram:VARiable:STRing{[1] - 10}:DATA ?

Example of use

```
10 OUTPUT 717;" PROG:VAR:STR ""TEST DATA"""
20 OUTPUT 717;" PROG:VAR:STR?"
30 ENTER 717;A$
```

SENSE

SCPI.SENSe(*Ch*).AVERage.CLEar

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).AVERage.CLEar

Description

This command resets the data count to 0, used for averaging of selected channel (*Ch*). Measurement data before the execution of this object is not used for averaging.

Examples

SCPI.SENSe(1).AVERage.CLEar

Related objects

SCPI.SENSe(Ch).AVERage.COUNt

SCPI.SENSe(Ch).AVERage.STATe

Equivalent key

Avg > Averaging Restart

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:AVERage:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:AVER:CLE"
```

SCPI.SENSe(*Ch*).AVERage.COUNt

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).AVERage.COUNt = *Value*

Value = SCPI.SENSe(*Ch*).AVERage.COUNt

Description

This command sets/gets the averaging factor of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Averaging factor
Data type	Long integer type (Long)
Range	1 to 999
Preset value	16
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AvgCnt As Long
SCPI.SENSe(1).AVERage.COUNt = 4
AvgCnt = SCPI.SENSe(1).AVERage.COUNt
```

Related objects

SCPI.SENSe(Ch).AVERage.STATe

SCPI.SENSe(Ch).AVERage.CLEar

Equivalent key

Avg > **Avg Factor**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:AVERage:COUNt <numeric>

:SENSe{[1]-160}:AVERage:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:AVER:COUN 4"
20 OUTPUT 717;":SENS1:AVER:COUN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).AVERage.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).AVERage.STATe = *Status*

Status = SCPI.SENSe(*Ch*).AVERage.STATe

Description

This command sets/gets the averaging function of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the averaging function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the averaging function. • False or OFF: Turns OFF the averaging function.
Preset value	False or OFF

Examples

```
Dim Avg As Boolean
SCPI.SENSe(1).AVERage.STATe = True
Avg = SCPI.SENSe(1).AVERage.STATe
```

Related objects

SCPI.SENSe(Ch).AVERage.COUNt

SCPI.SENSe(Ch).AVERage.CLEar

Equivalent key

Avg > **Averaging**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:AVERage[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:AVERage[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:AVER ON"
20 OUTPUT 717;":SENS1:AVER?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).BANDwidth.RESolution

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).BANDwidth.RESolution = *Value*

Value = SCPI.SENSe(*Ch*).BANDwidth.RESolution

Description

This command sets/gets the IF bandwidth of selected channel (*Ch*).

- This command is similar to SCPI.SENSe(Ch).BWIDth.RESolution.

Variable

Parameter	*Value*
Description	IF bandwidth
Data type	Double precision floating point type (Double)
Range	10 to 500000
Preset value	70000
Unit	Hz (hertz)
Resolution	In steps of 1, 1.5, 2, 3, 4, 5, or 7
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim IfBw As Double
SCPI.SENSe(1).BANDwidth.RESolution = 1.5E3
IfBw = SCPI.SENSe(1).BANDwidth.RESolution
```

Related objects

SCPI.SENSe(Ch).BWIDth.RESolution

Equivalent key

Avg > **IF Bandwidth**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:BANDwidth[:RESolution] <numeric>
:SENSe{[1]-160}:BANDwidth[:RESolution]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:BAND 1.5E3"
20 OUTPUT 717;":SENS1:BAND?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).BWIDth.RESolution

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).BWIDth.RESolution = *Value*

Value = SCPI.SENSe(*Ch*).BWIDth.RESolution

Description

This command sets/gets the IF bandwidth of selected channel (*Ch*).

- This command is similar to SCPI.SENSe(Ch).BANDwidth.RESolution.

Variable

Parameter	*Value*
Description	IF bandwidth
Data type	Double precision floating point type (Double)
Range	10 to 500000
Preset value	70000
Unit	Hz (hertz)
Resolution	In steps of 1, 1.5, 2, 3, 4, 5, or 7
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim IfBw As Double
SCPI.SENSe(1).BWIDth.RESolution = 1.5E3
IfBw = SCPI.SENSe(1).BWIDth.RESolution
```

Related objects

SCPI.SENSe(Ch).BANDwidth.RESolution

Equivalent key

Avg > IF Bandwidth

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:BWIDth[:RESolution] <numeric>

:SENSe{[1]-160}:BWIDth[:RESolution]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:BWID 1.5E3"
20 OUTPUT 717;":SENS1:BWID?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.CLEar

Description

This command clears the error coefficient for calibration when the frequency offset feature is set to OFF, for the selected channel (*Ch*).

Example of use

SCPI.SENSe(1).CORRection.CLEar

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.CLEar

Equivalent key

Cal > **Clear** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:CORR:CLE"
```

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.DATA (Str, Int1 , Int2) = Array

Array = SCPI.SENSe(Ch).CORRection.COEFficient.DATA (Str, Int1, Int2)

Description

This command sets/gets the calibration coefficient data for specified channel.

- When the calibration factor is interpolated, the interpolated calibration coefficient array is read. Similarly, when the calibration factor is not interpolated, a non-interpolated calibration coefficient array is read.

  After writing the calibration coefficient array, the written value becomes effective only after the (SCPI.SENSe(Ch).CORRection.COEFficient.SAVE) command is executed.

Variable

Parameter	*Array*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)X2. Where n is an integer between 1 and NOP. *Data(nX2-2)* Real part of data (complex number) at the n-th measurement point. *Data(nX2-1)* Imaginary part of data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Parameter	*Str*

Description	Calibration type
Data type	Character string type (String)
Range	Select from the following: • "ES": Source match • "ER": Reflection tracking • "ED": Directivity • "EL": Load match • "ET": Transmission tracking • "EX": Isolation

Parameter	*Int1*
Description	Response port
Data type	Integer type (Integer)
Range	1 to 4
Note	If ES, ER, or ED is used, the response port and the stimulus port must be the same, while EL, ET, or EX is used, the response port and the stimulus port must be different.

Parameter	*Int2*
Description	Stimulus port
Data type	Integer type (Integer)

Range	1 to 4
Note	If ES, ER, or ED is used, the response port and the stimulus port must be the same, while EL, ET, or EX is used, the response port and the stimulus port must be different.

For information on the variable (*Ch*), see Ch

Examples

```
DIM Array As Variant
Array = SCPI.SENs(1).CORRection.COEFficient.DATA("EL", 1, 2)
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.ERESponse

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT3

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT4

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COEFficient[:DATA]
{ES|ER|ED|EL|ET|EX},<numeric 1>, <numeric 2>, <numeric 3>,...,
<numeric 3 n*2>
:SENSe{[1]-160}:CORRection:COEFficient[:DATA]?
{ES|ER|ED|EL|ET|EX},<numeric 1>, <numeric 2>
```

Query response

```
{numeric 1}, ... ,{numeric NOP*2}<newline><^END>
```

	Description
{numeric nX 2-1}	Real part of data (complex number) at the n-th measurement point.
{numeric nX 2}	Imaginary part of data (complex number) at the n-th measurement point.

Because the calibration coefficient array is expressed by a complex number, the real part and the imaginary part of one measurement point are returned and obtained as a value. Here, NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:CORR:COEF? EL,1,2"
30 ENTER 717;A(*)
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.ERESponse

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.ERESponse = Ports

Description

This command sets the calibration type to the enhanced response calibration between the two specified ports when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel Ch.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port. • *Ports(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COEFficient.METHod.ERESponse = ERESPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod:ERESponse <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:ERES 1,2"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN = Port

Description

This command sets the calibration type to the response calibration (open) of the specified port when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel Ch.

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Val Ch (Ch) and Val Port respectively.

Examples

SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.OPEN = 1

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod[:RESPonse]:OPEN <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:OPEN 1"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.SHORt

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.SHORt = Port

Description

This command sets the calibration type to the response calibration (short) of the specified port for the selected channel, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command.

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Val Ch (Ch) and Val Port.

Examples

SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.SHORt = 1

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod[:RESPonse]:SHORt <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SHOR 1"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.THRU

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.THRU = Ports

Description

This command sets the calibration type to the response calibration (thru) between the two specified ports when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the input port. • *Ports(1):* Specifies the output port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.THRU = Array(2, 1)

Dim ThruPort(1) As Variant
ThruPort(0) = 2
ThruPort(1) = 1
SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.THRU = ThruPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod[:RESPonse]:THRU <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:THRU 2,1"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT1

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT1 = Port

Description

This command sets the calibration type to the full 1-port calibration of the specified port, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command, for the selected channel.

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Val Ch (Ch) and Val Port respectively.

Examples

SCPI.SENSe(1).CORREction.COEFficient.METHod.SLOT1 = 1

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT1 1"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2 = Ports

Description

This command sets the calibration type to full 2-port calibration between the two specified ports, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command for the selected channel.

Variable

Parameter	Ports
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for full 2-port calibration. • *Ports(1):* Specifies the other port for full 2-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples (1)

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT2 = Array(1, 2)
```

Examples (2)

```
Dim CalPort(1) As Variant
CalPort(0) = 1
CalPort(1) = 2
SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT2 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod:SOLT2 <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT2 1,2"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT3

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT3 = Ports

Description

This command sets the calibration type to full 3-port calibration between the three specified ports, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command for the selected channel.

Variable

Parameter	*Ports*
Description	Indicates 3-element array data (port number). • *Ports(0):* Specifies a port for full 3-port calibration. • *Ports(1):* Specifies a port for full 3-port calibration. • *Ports(2):* Specifies a port for full 3-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples (1)

SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT3 = Array(1, 2, 3)

Examples (2)

```
Dim CalPort(2) As Variant
CalPort(0) = 1
CalPort(1) = 2
CalPort(2) = 3
SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT3 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod:SOLT3 <numeric 1>,<numeric 2>, <numeric 3>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT3 1,2,3"
```

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT4

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT4 = Ports

Description

This command sets the calibration type to full 4-port calibration, when the calibration coefficient data array is written with the SCPI.SENSe(Ch).CORRection.COEFficient.DATA command for the selected channel.

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port number). • *Ports(0):* Specifies a port for full 4-port calibration. • *Ports(1):* Specifies a port for full 4-port calibration. • *Ports(2):* Specifies a port for full 4-port calibration. • *Ports(3):* Specifies a port for full 4-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples (1)

```
SCPI.SENSe(1).CORRection.COEFficient.METHod.SOLT4 = Array(1, 2, 3, 4)
```

Examples (2)

```
Dim CalPort(3) As Variant
CalPort(0) = 1
CalPort(1) = 2
CalPort(2) = 3
CalPort(3) = 4
SCPI.SENSe(1).CORRection.COEFfient.METHod.SLOT4 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:METHod:SOLT4 <numeric 1>, <numeric 2>, <numeric 3>, <numeric 4>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:METH:SOLT4 1,2,3,4"
```

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

Description

This command enables the calibration coefficients depending on the selected calibration type from the writing calibration data.

- Enabling the calibration coefficients clears all calibration data regardless of whether the data are used for the calculation and also clears the calibration type selections.

  If you execute this command before all calibration data needed for calculating the calibration coefficients are written, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Dmy As Long
Dim Data(3) as Variant
Data(0) = -1.123
Data(1) = 2.456
Data(2) = -2.249
Data(3) = 2.608
SCPI.SENSe(1).CORRection.COEFficient.METHod.RESPonse.THRU = Array(2, 1)
SCPI.SENSe(1).CORRection.COEFficient("ET", 2, 1) = Data
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).CORRection.COEFficient.SAVE
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.ERESponse

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.OPEN

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.SHORt

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.RESPonse.THRU

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT2

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT3

SCPI.SENSe(Ch).CORRection.COEFficient.METHod.SOLT4

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COEFficient:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COEF:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.ISOLation

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.ISOLation = *Ports*

Description

This command measures the calibration data of the isolation from the specified stimulus port to the specified response port, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port number. • *Ports(1):* Specifies the stimulus port number. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.ISOLation = Array(1,2)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim IsPort(1) As Variant
Dim Dmy As Long
IsPort(0) = 1
IsPort(1) = 2
```

```
SCPI.SENSe(1).CORRection.COLLect.ACQuire.ISOLation = IsPort
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Thru)** > **Isolation (Optional)**

Cal > **Calibrate** > **n-Port Cal** > **Isolation (Optional)** > **Port m-n Isol**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:ISOLation <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ISOL 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.LOAD

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.LOAD = *Port*

Description

This command measures the calibration data of the load standard for the specified port, for the selected channel (*Ch*).

Variable

Parameter	*Port*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.LOAD = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Open)**|**Response (Short)** > **Load (Optional)**

Cal > **Calibrate** > **1-Port Cal** > **Load**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Load**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:LOAD <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:LOAD 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LINE

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LINE = *Standard*

Standard = SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LINE

Description

This command selects the offset line to be connected with the load for the selected channel (*Ch*).

Variable

Parameter	*Standard*
Description	Standard number
Data type	Long integer type (Long)
Range	1 to 30
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.OLOad.LINE = 1
```

Related objects

Equivalent key

Cal > **Calibrate** > **Response (Open)**|**Response (Short)** > **Load (Optional)**

Cal > **Calibrate** > **1-Port Cal** > **Load**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Load**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OLOad:LINE <numeric>

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OLOad:LINE?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LINE 1"
20 OUTPUT 717;":SENS1:CORR:COLL:OLO:LINE?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.CLEar

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.CLEar = *Port*

Description

This command clears the acquired offset load data for the selected port and channel (*Ch*).

Variable

Parameter	*Port*
Description	Port number for clear
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.OLOad.LOAD.CLEar = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOAd.LOAD.DONE

Equivalent key

Cal > **Calibrate** > **Enhanced Response** > **Load** > **Load n - Offset Load** > **Clear**

Cal > **Calibrate** > **1-Port Cal** > **Load** > **Load n - Offset Load** > **Clear**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Load** > **Load n - Offset Load** > **Clear**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OLOad:LOAD:CLEar <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD:CLE 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.COUNt(*Pt*)

Object type

Property (**Read-only**)

Syntax

Count = SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.COUNt(*Pt*)

Description

This command returns the number of count for the acquired offset calibration data for the offset load standard for the selected port (*Pt*) and channel (*Ch*).

In the other word, number of times SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute is executed. The maximum number of count for the acquired offset calibration data is two. If it is exceeded, the error occurs.

Variable

Parameter	*Count*
Description	Number of count for the offset calibration acquisition
Data type	Long integer type (Long)
Range	
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Count As Long
Count = SCPI.SENSe(1).CORRection.COLLect.ACQuire.OLOad.LOAD.COUNt(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute

Equivalent key

No Equivalent Softkey

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OLOad:LOAD:COUNt? <numeric>

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD:COUN? 1"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.DONE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.DONE = *Port*

Description

This command finalizes the offset load calibration for the selected channel (*Ch*), then calculate the load correction coefficient data.

Variable

Parameter	*Port*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

See SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute

Equivalent key

Cal > **Calibrate** > **Enhanced Response** > **Load** > **Load n - Offset Load** > **Done**

Cal > **Calibrate** > **1-Port Cal** > **Load** > **Load n - Offset Load** > **Done**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Load** > **Load n - Offset Load** > **Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OLOad:LOAD:DONE <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD:DONE 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OLOad.LOAD.EXECute = *Port*

Description

This command measures the calibration data of a offset load standard for the specified port, for the selected channel (*Ch*).

Variable

Parameter	*Port*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Dmy As Long
' Select X11644 Calibration Kit
SCPI.SENSe.CORRection.COLLect.CKIT.SELect = 15
' Select Load 2 in Load
SCPI.SENSe.CORRection.COLLect.ACQuire.SUBClass = 2
' Select Offset Line 1 (Thur).  The "Thru" is defined as the standard No. 6 in the Define Standard.
SCPI.SENSe.CORRection.COLLect.ACQuire.OLOad.LINE = 6
MsgBox "Connect Offset Load (Thru)"
' Make a measurement for Offset Line 1
SCPI.SENSe.CORRection.COLLect.ACQuire.OLOad.Load.EXECute = 1
' Wait for measurement end.
Dmy = SCPI.IEEE4882.OPC
' Select Offset Line 2 (X-Band Delay Line).The "X-Band Delay Line" is defined as the standard No. 7 in
the Define Standard.
SCPI.SENSe.CORRection.COLLect.ACQuire.OLOad.LINE = 7
MsgBox "Connect Offset Load (X-Band Delay Line)"
' Make a measurement for Offset Line 2
SCPI.SENSe.CORRection.COLLect.ACQuire.OLOad.Load.EXECute = 1
' Wait for measurement end.
Dmy = SCPI.IEEE4882.OPC
```

```
' Finalize the Offset load measurement.
SCPI.SENSe.CORRection.COLLect.ACQuire.OLOad.Load.DONE = 1
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOAD.LINE

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOAd.LOAD.DONE

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.COUNt

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.OLOad.LOAD.CLEar

SCPI.IEEE4882.OPC

Equivalent key

Cal > Calibrate > Enhanced Response > Load > Load n - Offset Load> Offset Line n

Cal > Calibrate > 1-Port Cal > Load > Load n - Offset Load > Offset Line n

Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Offset Load > Offset Line n

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OLOad:LOAD[:EXECute] <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OLO:LOAD 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

Refer to Example for COM.

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OPEN

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.OPEN = *Port*

Description

This command measures the calibration data of the open standard for the specified port, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Ch and Port respectively.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.OPEN = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Open)**|**1-Port Cal** > **Open**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Open**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:OPEN <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:OPEN 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SHORt

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SHORt = *Port*

Description

This command measures the calibration data of the short standard for the specified port, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Ch and Port.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.SHORt = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Short)**|**1-Port Cal** > **Short**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Short**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SHORt <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SHOR 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.CLEar

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.CLEar

Description

This command clears the acquired sliding load data for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.ACQuire.SLOad.CLEar

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.DONE

Equivalent key

Cal > **Calibrate** > **Response (Open)**|**Response (Short)** > **Load** > **Load n - Sliding Load** > **Clear**

Cal > **Calibrate** > **Enhanced Response** > **Load** > **Load n - Sliding Load** > **Clear**

Cal > **Calibrate** > **1-Port Cal** > **Load** > **Load n - Sliding Load** > **Clear**

Cal > **Calibrate** > **n-Port Cal** > **Reflection** > **Port m Load** > **Load n - Sliding Load** > **Clear**

Cal > **Calibrate** > **n-Port TRL Cal** > **Line/Match** > **x-y Line/Match | x-y Fwd (sxy) | x-y Rvs (syx)** > **Line/Match n - Sliding Load** > **Clear**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SLOad:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:CLE"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.COUNt

Object type

Property (**Read-only**)

Syntax

Count = SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.COUNt

Description

This command returns the number of count for the acquired sliding calibration data for the sliding load standard for the selected channel (*Ch*).

In the other word, the number of times SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute or SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute is executed. The maximum number of count for the acquired offset calibration data is ten. If it is exceeded, the error occurs.

Variable

Parameter	*Count*
Description	Number of count for the sliding calibration acquisition
Data type	Long integer type (Long)
Range	
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim Count As Long
Count = SCPI.SENSe(1).CORRection.COLLect.ACQuire.SLOad.COUNt
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute

Equivalent key

No Equivalent Softkey

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SLOad:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:COUN?"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.DONE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.DONE

Description

This command finalizes the sliding load calibration for the selected channel (*Ch*), then calculate the load correction coefficient data.

Examples

See SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute

SCPI.IEEE4882.OPC

Equivalent key

Cal > Calibrate > Response (Open)|Response (Short) > Load > Load n - Sliding Load > Done

Cal > Calibrate > Enhanced Response > Load > Load n - Sliding Load > Done

Cal > Calibrate > 1-Port Cal > Load > Load n - Sliding Load > Done

Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Sliding Load > Done

Cal > Calibrate > n-Port TRL Cal > Line/Match > x-y Line/Match | x-y Fwd (sxy) | x-y Rvs (syx) > Line/Match n - Sliding Load > Done

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SLOad:DONE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:DONE"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.EXECute = *Port*

Description

This command measures the calibration data of a sliding load standard for the specified port, for the selected channel (*Ch*). From 5 to 10 times of the sliding measurement is required.

Variable

Parameter	*Port*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
' Select the 85052B Calibration Kit
SCPI.SENSe.CORRection.COLLect.CKIT.SELect = 19
' Select Load 5 (Sliding Load (m)) in load
SCPI.SENSe.CORRection.COLLect.ACQuire.SUBClass = 5
For i = 1 To 5
   MsgBox "Set Sliding Load position"
   ' Make a measurement for sliding load calibration
   SCPI.SENSe.CORRection.COLLect.ACQuire.SLOad.Load.EXECute = 1
   Dmy = SCPI.IEEE4882.OPC
Next i
SCPI.SENSe.CORRection.COLLect.ACQuire.SLOad.DONE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.DONE

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.COUNt

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.CLEar

SCPI.IEEE4882.OPC

Equivalent key

Cal > Calibrate > Response (Open)|Response (Short) > Load > Load n - Sliding Load > Sliding Load

Cal > Calibrate > Enhanced Response > Load > Load n - Sliding Load > Sliding Load

Cal > Calibrate > 1-Port Cal > Load > Load n - Sliding Load > Sliding Load

Cal > Calibrate > n-Port Cal > Reflection > Port m Load > Load n - Sliding Load > Sliding Load

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SLOad:LOAD[:EXECute] <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:LOAD 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

See Example for COM

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute = *Ports*

Description

This command measures the calibration data of a sliding load standard for TRL line from for the specified ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the input port. • *Ports(1):* Specifies the output port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim Port(1) As Variant, Dmy As Long
Port(0) = 2
Port(1) = 1
SCPI.SENSe(1).CORRection.COLLect.ACQuire.SLOad.TRLLine.EXECute  = Port
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.LOAD.EXECute

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.DONE

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.CLEar

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SLOad.COUNt

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **n-Port TRL Cal** > **Line/Match** > **x-y Line/Match** | **x-y Fwd (sxy)** | **x-y Rvs (syx)** > **Line/Match n - Sliding Load**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SLOad:TRLLine[:EXECute] <numeric (input port)>,<numeric (output port)>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SLO:TRLL 2,1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SUBClass

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SUBClass = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.SUBClass

Description

This command sets/gets the subclass which is used for the calibration of the selected channel (*Ch*). For example, if two different subclasses are set in advance, say Thru 1 & Thru 2, which are visible at the calibration softkey, this command can select either Thru1 or Thru2. When performing Thru cal, either Thru 1 or Thru 2 set with this command is used for the calibration.

Variable

Parameter	*Value*
Description	The setting number of the standard subclass for the calibration.
Data type	Long integer type (Long)
Range	1 to 30
Preset value	1
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Subc As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.SUBClass = 3
Subc = SCPI.SENSe(1).CORRection.COLLect.ACQuire.SUBClass
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SUBClass <numeric>

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:SUBClass?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SUBC 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.THRU

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.THRU = *Ports*

Description

This command measures the calibration data of the Thru standard from the specified stimulus port to the specified response port, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the input port. • *Ports(1):* Specifies the output port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.THRU = Array(2,1)
Dmy = SCPI.IEEE4882.OPC

Dim ThruPort(1) As Variant
Dim Dmy As Long
ThruPort(0) = 2
ThruPort(1) = 1
```

```
SCPI.SENSe(1).CORRection.COLLect.ACQuire.THRU = ThruPort
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Calibrate** > **Response (Thru)** > **Thru**

Cal > **Calibrate** > **n-Port Cal** > **Transmission** > **Port m-n Thru**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:THRU <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:THRU 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.TRLLine

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.TRLLine = *Ports*

Description

This command executes LINE or MATCH measurement of the TRL calibration for the selected calibration kit, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the input port. • *Ports(1):* Specifies the output port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLLine = Array(1,2)
Dmy = SCPI.IEEE4882.OPC
```

```
Dim Trll(1) As Variant
Dim Dmy As Long
Trll(0) = 1
Trll(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLLine = Trll
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLThru

Equivalent key

Cal > Calibrate > 2-Port TRL Cal > Line/Match > x-y Line/Match|x-y Fwd (Syx)|x-y Rvs (Sxy)

Cal > Calibrate > 3-Port TRL Cal > Line/Match > x-y Line/Match|x-y Fwd (Syx)|x-y Rvs (Sxy)|x-z Line/Match|x-z Fwd (Szx)|x-z Rvs (Sxz)|y-z Line/Match|y-z Fwd (Szy)|y-z Rvs (Syz)

Cal > Calibrate > 4-Port TRL Cal > Line/Match > x-y Line/Match|x-y Fwd (Syx)|x-y Rvs (Sxy) > x-z Line/Match|x-z Fwd (Szx)|x-z Rvs (Sxz) > x-w Line/Match|x-w Fwd (Swx)|x-w Rvs (Sxw) > y-z Line/Match|y-z Fwd (Szy)|y-z Rvs (Syz) > y-w Line/Match|y-w Fwd (Swy)|y-w Rvs (Syw) > z-w Line/Match|z-w Fwd (Swz)|z-w Rvs (Szw)

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:TRLLine <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:TRLL 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.TRLReflect

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.TRLReflect = *Port*

Description

This command executes the reflection measurement of the TRL calibration for the selected calibration kit, for the selected channel (*Ch*).

Variable

Parameter	*Port*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLReflect = 1
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLLine

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLThru

Equivalent key

Cal > **Calibrate** > **2-Port TRL Cal** > **Reflect** > **Portx Reflect**|**Porty Reflect**

Cal > **Calibrate** > **3-Port TRL Cal** > **Reflect** > **Portx Reflect**|**Porty Reflect**|**Portz Reflect**

Cal > **Calibrate** > **4-Port TRL Cal** > **Reflect** > **Portx Reflect**|**Porty Reflect**|**Portz Reflect**|**Portw Reflect**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:TRLReflect <numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:TRLR 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.TRLThru

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ACQuire.TRLThru = *Ports*

Description

This command executes the THRU measurement of the TRL calibration for the selected calibration kit, for the selected channel (*Ch*).

- Both forward and backward way are required .

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the input port. • *Ports(1):* Specifies the output port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLThru = Array(1,2)
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLThru = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
```

```
Trlt1(0) = 1
Trlt1(1) = 2
Trlt2(0) = 2
Trlt2(1) = 1
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLThru = Trlt1
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).CORRection.COLLect.ACQuire.TRLThru = Trlt2
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLLine

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.TRLReflect

Equivalent key

Cal > **Calibrate** > **2-Port TRL Cal** > **Thru/Line** > **Port x-y Thru**

Cal > **Calibrate** > **3-Port TRL Cal** > **Thru/Line** > **Port x-y Thru**|**Port x-z Thru**|**Port y-z Thru**

Cal > **Calibrate** > **4-Port TRL Cal** > **Thru/Line** > **Port x-y Thru**|**Port x-z Thru**|**Port x-w Thru**|**Port y-z Thru**|**Port y-w Thru**|**Port z-w Thru**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect[:ACQuire]:TRLThru <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:TRLT 1,2"
20 OUTPUT 717;"*OPC?"
30 OUTPUT 717;":SENS1:CORR:COLL:TRLT 2,1"
40 OUTPUT 717;"*OPC?"
50 ENTER 717;A
```

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).LENGth

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).LENGth=*length*

Description

This command sets/displays the approximate length of the adaptor, for the selected channel (*Ch*) and for the selected port.

Variable

Parameter	*length*
Description	Adapter Length
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Note	Adapter length is positive for adaptor removal and negative for adaptor insertion.

Examples

SCPI.SENSe(1).CORRection.COLLect.ADAPter(2).LENGth = 0.01

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ADAPter.REMoval

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Adapter Length**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:LENGth <+ or -><value of length>

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ADAP2:LENG 0.01"
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP2:LENG?"
30 ENTER 717;A$
```

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

Description

This command executes Adapter Removal/Insertion along with moving the phase of adapter (which is removed or inserted) to 180 degrees. This command is useful in cases where auto judgement of phase fails. This command can be executed several times while the calibration remains valid.

- If user cannot execute this command, then "Execution error" is displayed.

Examples

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

Related Object

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Rotate Adapter**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:ROTate

Example of use

```
10 OUTPUT 717;"SENS1:CORR:COLL:ADAP2:ROT"
```

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff = *Cfreq*

Cfreq = SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff

Description

This command sets/gets approximate cutoff frequency of the waveguide, for the selected channel (Ch) and for the selected port.

Variable

Parameter	*Cfreq*
Description	Waveguide cutoff frequency
Data type	Double precision floating point type (Double)
Range	0 to 20E9
Unit	Hz
Preset value	0 Hz

Examples

```
Dim CFreq As Double
SCPI.SENSe(1).CORRection.COLLect.ADAPter(2).WAVeguide.CUToff = 700
CFreq= SCPI.SENSe(1).CORRection.COLLect.ADAPter(2).WAVeguide.CUToff
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Cutoff Frequency**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:WAVeguide:CUToff <numeric>

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:WAVeguide:CUToff?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:CUT 2000"
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:CUT?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth = *length*

length = SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.LENGth

Description

This command sets/gets approximate adapter length of the waveguide, for the selected channel (Ch) and for the selected port.

Variable

Parameter	*length*
Description	Waveguide Adapter Length
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Preset value	0 second
Note	Adapter length is positive for adaptor removal and negative for adaptor insertion.

Examples

```
Dim CLen As Double
SCPI.SENSe(1).CORRection.COLLect.ADAPter(2).WAVeguide.LENGth = -5
CLen= SCPI.SENSe(1).CORRection.COLLect.ADAPter(2).WAVeguide.LENGth
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).ROTate

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).WAVeguide.CUToff

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Waveguide Length**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:WAVeguide:LENGth <numeric>

:SENSe{[1]-160}:CORRection:COLLect:ADAPter{[1]-4}:WAVeguide:LENGth?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:LEN 5"
20 OUTPUT 717;":SENS1:CORR:COLL:ADAP:WAV:LEN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel = *Lbl*

Lbl = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.LABel

Description

This command sets/gets the calibration kit name for the calibration kit selected for channels 1 to 36 (*Ch*).

Variable

Parameter	*Lbl*
Description	Calibration kit name
Data type	Character string type (String)
Range	254 characters or less
Preset value	Varies depending on the calibration kit number: • 85031B • 85032B/E • 85032F • 85033D • 85033E • 85036B • 85036E • 85038A/F/M • 85039B • 85050C • 85050D • 85052D • 85052C • 85054D

	• 85056D • 85056K • K11644A • P11644A • X11644A • <User Defined Cal Kits>

Examples

```
Dim CalLbl As String
SCPI.SENSe(1).CORRection.COLLect.CKIT.LABel = "User 1"
CalLbl = SCPI.SENSe(1).CORRection.COLLect.CKIT.LABel
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Label Kit**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:LABel <string>
:SENSe{[1]-160}:CORRection:COLLect:CKIT:LABel?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:LAB ""MY_KIT"""
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:LAB?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.LOAD(*Cpt*)

Description

This command sets/gets the standard used for the load measurement of the specified port (*Cpt*), for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Cpt*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanLoad As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.LOAD(1) = 10
StanLoad = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.LOAD(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Load** > **Set All**|**Port 1**|**Port 2**|**Port 3**|**Port 4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:LOAD <numeric 1>,<numeric 2>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:LOAD? <numeric 1>

Query response

{numeric 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:LOAD 1,9"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:LOAD? 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.OPEN(*Cpt*)

Description

This command sets/gets the standard used for the open measurement of the specified port (*Cpt*), for a calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Examples

```
Dim StanOpen As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.OPEN(1) = 10
StanOpen = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.OPEN(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Open** > **Port 1**|**Port 2**|**Port 3**|**Port 4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:OPEN <numeric 1>,<numeric 2>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:OPEN? <numeric 1>

Query response

{numeric 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:OPEN 1,2"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:OPEN? 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SELect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SELect = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SELect

Description

This command sets/gets a subclass in the subclass menu. When sending this command, it selects the subclass from 1 to 8.

Variable

Parameter	*Value*
Description	The number of the standard subclass for the calibration.
Data type	Long integer type (Long)
Range	1 to 8
Preset Value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim CKitSel As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.SELect = 3
CKitSel = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.SELect
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ACQuire.SUBClass

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Sub Class** > **Sub Class 1**| **...** |**Sub Class 8**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:SELect <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:SELect?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SEL 1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SEL?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.SHORt(*Cpt*)

Description

This command sets/gets the standard used for the short measurement of the specified port (*Cpt*), for the calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variable (*Cpt*) has no preset value, you cannot omit it. If you omit the variable (*Cpt*), an error occurs when executed.

Examples

```
Dim StanShor As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.SHORt(1) = 10
StanShor = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.SHORt(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Short** > **Port 1**|**Port 2**|**Port 3**|**Port 4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:SHORt <numeric 1>,<numeric 2>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:SHORt? <numeric 1>

Query response

{numeric 2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SHOR 1,1"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:SHOR? 1"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Cpt_m*,*Cpt_n*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Cpt_m*,*Cpt_n*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.THRU(*Cpt_m*,*Cpt_n*)

Description

This set/get the standard used for the thru measurement between the specified 2 ports (*Cpt_m* and *Cpt_n*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Cpt_m*, *Cpt_n*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

- Since the variables (*Cpt_m* and *Cpt_n*) have no preset value, you cannot omit them. If you omit the variables (*Cpt_m* and *Cpt_n*) or if you specify the same port number to 2 port numbers, an error occurs when executed. Notice that when you specify 2 ports with the variables (*Cpt_m* and *Cpt_n*), the order of the 2 port numbers is arbitrary.

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)

Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanThru As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.THRU(1,2) = 10
StanThru = SCPI.SENSe(1).CORRection.COLLect.CKIT.ORDer.THRU(1,2)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **Thru** > **Port 1-2**|**Port 1-3**|**Port 1-4**|**Port 2-3**|**Port 2-4**|**Port 3-4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:THRU <numeric 1>,<numeric 2>, <numeric 3>
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:THRU? <numeric 1>,<numeric 2>
```

Query response

```
{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:THRU 1,2,11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:THRU? 1,2"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLLine(*Cpt_m*,*Cpt_n*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLLine(*Cpt_m*,*Cpt_n*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLLine(*Cpt_m*,*Cpt_n*)

Description

This command sets/gets the standard used for the line measurement of TRL calibration between the specified 2 ports (*Cpt_m* and *Cpt_n*), for the calibration kit selected for selected channel (*Ch*).

- If the standard number is 0, the standard is invalid for the subclass.

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanTrll As Long
SENSe(1).CORRection.COLLect.CKIT.ORDer.TRLLine(1,2) = 10
StanTrll = SENSe(1).CORRection.COLLect.CKIT.ORDer.TRLLine(1,2)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **TRL Line/Match** > **Set All**|**Port 1-2**|**Port 1-3**|**Port 1-4**|**Port 2-3**|**Port 2-4**|**Port 3-4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLLine <numeric 1>,<numeric 2>, <numeric 3>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLLine? <numeric 1>,<numeric 2>

Query response

{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLL 1,2,11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLL? 1,2"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLReflect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLreflect = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLreflect

Description

This command sets/gets the standard used for the reflection measurement of the TRL calibration between the specified 2 ports, for the calibration kit selected for selected channel (*Ch*).

- If the standard number is 0, the standard is invalid for the subclass.

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanTrlr As Long
SENSe(1).CORRection.COLLect.CKIT.ORDer.TRLReflect = 5
StanTrlr = SENSe(1).CORRection.COLLect.CKIT.ORDer.TRLReflect
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **TRL Reflect**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLReflect
<numeric>
:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLReflect?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLR 11"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLThru(*Cpt_m*,*Cpt_n*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLThru(*Cpt_m*,*Cpt_n*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.ORDer.TRLThru(*Cpt_m*,*Cpt_n*)

Description

This command sets/gets the standard for the THRU measurement of the TRL calibration between the specified 2 ports (*Cpt_m* and *Cpt_n*).

- If the standard number is 0, the standard is invalid for the subclass.

Variable

Parameter	*Value*
Description	Standard number
Data type	Long integer type (Long)
Range	0 to 21
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim StanTrlt As Long
SENSe(1).CORRection.COLLect.CKIT.ORDer.TRLThru(1,2) = 3
StanTrlt = SENSe(1).CORRection.COLLect.CKIT.ORDer.TRLThru(1,2)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

Equivalent key

Cal > **Modify Cal Kit** > **Specify CLSs** > **TRL Thru** > **Set All**|**Port 1-2**|**Port 1-3**|**Port 1-4**|**Port 2-3**|**Port 2-4**|**Port 3-4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLThru <numeric 1>, <numeric 2>, <numeric 3>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:ORDer:TRLThru? <numeric 1>,<numeric 2>

Query response

{numeric 3}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLT 1,2,5"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:ORD:TRLT? 1,2"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.RESet

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.RESet

Description

This command resets the calibration kit selected for selected channel (*Ch*) to the default factory setting state.

Examples

SCPI.SENSe(1).CORRection.COLLect.CKIT.RESet

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:RESet

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:RES"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.SELect

Description

This command sets/gets the calibration kit of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of calibration kit
Data type	Long integer type (Long)
Range	1 to 30
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
Dim CalKit As Long
SCPI.SENSe(1).CORRection.COLLect.CKIT.SELect = 3
CalKit = SCPI.SENSe(1).CORRection.COLLect.CKIT.SELect
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.LOAD(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.OPEN(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.SHORt(Cpt)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.THRU(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLLine(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLReflect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.ORDer.TRLThru(Cpt_m,Cpt_n)

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).DELay

Equivalent key

Cal > **Cal Kit**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT[:SELect] <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT[:SELect]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT 3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).ARBitrary

Description

This command sets/gets the value of the arbitrary impedance of the standards (*Std*), for the calibration kit selected for the selected channel (*Ch*).

Variable

Parameter	*Std*
Description	Standard number
Data type	Long integer type (Long)
Range	1 to 21
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	Value of arbitrary impedance
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.

Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanArbt As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).ARBitrary = 50.5
StanArbt = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).ARBitrary
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name**> **Arb. Impedance**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:ARBitrary <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:ARBitrary?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:ARB 50.5"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:ARB?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C0

Description

This command sets/gets the value of the CO value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	C0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	fF (femto farad): 1E-15 F (farad)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanC0 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C0 = 12.3
StanC0 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C0
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C0**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C0 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C0?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C0 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C0?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C1

Description

This command sets/gets the value of the C1 value of the standards (*Std*), for the selected calibration kit and channel (*Ch*).

Variable

Parameter	*Value*
Description	C1
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-27 F/Hz (1E-27 farad / hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanC1 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C1 = 12.3
StanC1 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C1
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name**> **C1**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C1 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C1?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C1 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C2

Description

This command sets/gets the value of the C2 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	C2
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-36 F/Hz^2 (1E-36 farad /hertz^2)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanC2 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C2 = 12.3
StanC2 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C2
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C2**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C2 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C2?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C2 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C2?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).C3

Description

This command sets/gets the value of the C3 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	C3
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-45 F/Hz^3 (1E-45 farad / hertz^3)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanC3 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C3 = 12.3
StanC3 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).C3
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **C3**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C3 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:C3?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C3 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:C3?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).CHARacter

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).CHARacter = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).CHARacter

Description

This command sets/gets the media type of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Select media type of standard.
Data type	Character string type (String)
Range	Select from either of the following: • "COAXial": Selects coaxial as the media type. • "WAVeguide": Selects waveguide as the media type.
Preset value	"COAXial"

Examples

```
Dim StanChar As Double
SCPI.SENSe(1).SENSe.CORRection.COLLect.CKIT.STAN.CHARacter = "WAVeguide"
StanChar = SCPI.SENSe.CORRection.COLLect.CKIT.STAN.CHARacter
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).DELay

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Media**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:CHARacter {COAXial|WAVeguide}

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:CHARacter?

Query response

{COAXial|WAVeguide}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:CHAR WAV"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:CHAR?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).DELay

Description

This command sets/gets the value of the offset delay of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Offset delay
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanDel As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).DELay = 12.3
StanDel = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).DELay
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).CHARacter

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Offset Delay**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:DELay <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:DELay?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:DEL 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:DEL?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMAXimum

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMAXimum = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMAXimum

Description

This command sets/gets the value of the stop frequency of the standard (*Std*), for the calibration kit selected for selected channel (*Ch*).

- When media type of standard is "Waveguide", sets the stop frequency of the cutoff frequency.

Variable

Parameter	*Value*
Description	Stop frequency of the selected standard.
Data type	Double precision floating point type (Double)
Range	0 to 999E9
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanFMax As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).FMAXimum = 13.2E9
StanFMax = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).FMAXimum
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).FMINimum

Equivalent key

[Cal] > **Modify Cal Kit** > **Define STDs** > **no. name** > **Max. Frequency**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMAXimum
<numeric>
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMAXimum?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMAX 5E9"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMAX?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMINimum

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMINimum = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).FMINimum

Description

This command sets/gets the value of the start frequency of the standard *(Std*), for the selected calibration kit and channel (*Ch*).

- When media type of standard is "Waveguide", sets the start frequency of the cutoff frequency.

Variable

Parameter	*Value*
Description	Start frequency of the selected standard.
Data type	Double precision floating point type (Double)
Range	0 to 999E9
Preset value	Varies depending on the specified calibration kit and standard.
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanFMin As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).FMINimum = 600E6
StanFMax = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).FMINimum
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.STAN(Std).FMAXimum

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Min. Frequency**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMINimum
<numeric>
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:FMINimum?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMIN 1E9"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:FMIN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L0

Description

This command sets/gets the value of the L0 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	L0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	pH (pico henry)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanL0 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L0 = 12.3
StanL0 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L0
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L0**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L0 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L0?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L0 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L0?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L1

Description

This command sets/gets the value of the L1 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	L1
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-24 H/Hz (1E-24 henry / hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanL1 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L1 = 12.3
StanL1 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L1
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L1**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L1 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L1?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L1 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L2

Description

This command sets/gets the value of the L2 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	L2
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-33 H/Hz^2 (1E-33 henry / hertz^2)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanL2 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L2 = 12.3
StanL2 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L2
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L2**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L2 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L2?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L2 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L2?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).L3

Description

This command sets/gets the value of the L3 value of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	L3
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	1E-42 H/Hz^3 (1E-42 henry / hertz^3)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanL3 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L3 = 12.3
StanL3 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).L3
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **L3**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L3 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:L3?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L3 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:L3?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel = *Lbl*

Lbl = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LABel

Description

This command sets/gets the name of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Lbl*
Description	Standard name
Data type	Character string type (String)
Range	254 characters or less
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanLbl As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LABel = "OPEN 3.5mm"
StanLbl = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LABel
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Label**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:LABel <string>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:LABel?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LAB ""OPEN"""
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LAB?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LOSS

Description

This command sets/gets the value of the offset loss of the standards 1 to 21 (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Offset loss
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	ohm/s
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanLoss As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LOSS = 12.3
StanLoss = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LOSS
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > Modify Cal Kit > Define STDs > no. name > Offset Loss

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-21}:LOSS <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-21}:LOSS?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LOSS 12.3"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LOSS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LTYPe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LTYPe = *Type*

Type = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).LTYPe

Description

This command sets/gets the length type for the load standard of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Type*
Description	Length type
Data type	Character string type (String)
Range	Select from either of the following: • "FIXed": Normal load • "SLIDing": Sliding load • "OFFSet": Offset load.
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanType As String
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LTYPe = "SLID"
StanType = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).LTYPe
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Length Type**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:LTYPe {FIXed|SLIDing|OFFSet}

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:LTYPe?

Query response

{FIX|SLID|OFFS}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LTYP SLID"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:LTYP?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).TYPE

Description

This command sets/gets the standard type of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Standard type
Data type	Character string type (String)
Range	Select from either of the following: • "OPEN": Specifies open. • "SHORt": Specifies short. • "LOAD": Specifies load. • "THRU": Specifies Thru. • "UTHRu": Specifies Unknown Thru. • "ARBI": Specifies arbitrary impedance. • "NONE": Specifies DUT of which theoretical value is 0.
Preset value	Varies depending on the specified calibration kit and standard.

Examples

```
Dim StanType As String
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).TYPE = "OPEN"
StanType = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).TYPE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **STD Type**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:TYPE
{OPEN|SHORt|LOAD|THRU|UTHRu|ARBI|NONE}
:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:TYPE?
```

Query response

```
{OPEN|SHORt|LOAD|THRU|UTHRu|ARBI|NONE}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TYPE OPEN"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:TYPE?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0 = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.STAN(*Std*).Z0

Description

This command sets/gets the value of the offset Z0 of the standards (*Std*), for the calibration kit selected for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Offset Z0
Data type	Double precision floating point type (Double)
Range	-1E18 to 1E18
Preset value	Varies depending on the specified calibration kit and standard.
Unit	ohm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim StanZ0 As Double
SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).Z0 = 50
StanZ0 = SCPI.SENSe(1).CORRection.COLLect.CKIT.STAN(5).Z0
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.SELect

Equivalent key

Cal > **Modify Cal Kit** > **Define STDs** > **no. name** > **Offset Z0**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:Z0 <numeric>

:SENSe{[1]-160}:CORRection:COLLect:CKIT:STAN{[1]-30}:Z0?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:Z0 50"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:STAN1:Z0?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.TRLoption.IMPedance

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.TRLoption.IMPedance = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.TRLoption.IMPedance

Description

This command sets/gets the reference impedance during the TRL calibration, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Selects the reference impedance during the TRL calibration.
Data type	Character string type (String)
Range	Select from either of the following: • "SYSTem": Calculate the error coefficients by setting the system impedance to the reference impedance. • "LINE" :Calculate the error coefficients by setting the characteristic impedance of the line standard to the reference impedance.
Preset value	"SYSTem"

Examples

```
Dim TrlImp As String
SCPI.SENSe(1).CORRection.COLLect.CKIT.TRLoption.IMPedance = “LINE“
TrlImp = SCPI.SENSe(1).CORRection.COLLect.CKIT.TRLoption.IMPedance
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.TRLoption.RPLane

Equivalent key

Cal > **Modify Cal Kit** > **TRL Option** > **Impedance**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLoption:IMPedance {LINE|SYSTem}

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLoption:IMPedance?

Query response

{LINE|SYST}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:IMP LINE"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:IMP?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.TRLoption.RPLane

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.TRLoption.RPLane = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.CKIT.TRLoption.RPLane

Description

This command sets/gets the calculation method of the calibration plane, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Selects the calculation method of the calibration plane.
Data type	Character string type (String)
Range	Select from either of the following: • "THRU": Uses the length of the THRU and LINE standard to calculate the calibration plane. • "REFLect": Uses the reflection coefficient of the reflection standard to calculate the calibration plane.
Preset value	"THRU"

Examples

```
Dim TrlRpl As String
SCPI.SENSe(1).SENSe(1).CORRection.COLLect.CKIT.TRLoption.RPLane = “REFLect“
TrlRpl = SCPI.SENSe(1).CORRection.COLLect.CKIT.TRLoption.RPLane
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.CKIT.TRLoption.IMPedance

Equivalent key

Cal > **Modify Cal Kit** > **TRL Option** > **Reference Plane**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLoption:RPLane {THRU|REFLect}

:SENSe{[1]-160}:CORRection:COLLect:CKIT:TRLoption:RPLANE?

Query response

{THRU|REFL}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:RPL REFL"
20 OUTPUT 717;":SENS1:CORR:COLL:CKIT:TRL:RPL?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.CLEar

Description

This command clears the calibration measurement data when the frequency offset feature is off, for the selected channel (*Ch*).

- Settings that have been temporarily changed due to measurement for each standard (number of traces, measurement parameters, and so on) return to their original values.

Example of use

SCPI.SENSe(1).CORRection.COLLect.CLEar

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

Equivalent key

Cal > **Calibrate** > **Responce(Open)** > **Cancel** > **OK**

Cal > **Calibrate** > **Responce(Short)** > **Cancel** > **OK**

Cal > **Calibrate** > **Responce(Thru)** > **Cancel** > **OK**

Cal> **Calibrate** > **1-Port Cal** > **Cancel** > **OK**

Cal> **Calibrate** > **2-Port Cal** > **Cancel** > **OK**

Cal> **Calibrate** > **3-Port Cal** > **Cancel** > **OK**

Cal> **Calibrate** > **4-Port Cal** > **Cancel** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:CLE"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.CCHeck.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.CCHeck.ACQuire

Description

This command executes the confidence check of the calibration coefficients for selected channel (*Ch*), using ECal (Electronic Calibration). In other words, this command sets the data measured with the analyzer and the data stored in ECal so that they can be compared).

- If you execute this object when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

  This function is available with the Firmware revision 3.50 or greater.

Example of use

SCPI.SENSe(1).CORRection.COLLect.ECAL.CCHeck.ACQuire

Equivalent key

Cal > **ECal** > **Confidence Check**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:CCHeck[:ACQuire]

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:CCH"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ERESponse

Object type

Property (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ERESponse = Eports

Description

This command executes enhanced response calibration between the two specified ports of selected channel using the ECal (Electrical Calibration) module.

- If you execute this command when the ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 2-element array data (port number). • *EPorts(0):* Specifies the response port. • *EPorts(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.ERESponse = ERESPort
```

Equivalent key

Cal > ECal > Enhanced Response > 2-1 (S21 S11)|3-1 (S31 S11)|...|3-4 (S34 S44)

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ERESponse <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ERES 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.INFormation

Object type

Property (**Read Only**)

Syntax

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.INFormation?

Description

This command gets information of the connected ECAL modules to the E5071C.

Variable

The command return information in a string variable <param> in the following syntax:

[For 2-port ECal]

- ModelNumber
- SerialNumber
- PortAConnector
- PortBConnector
- MinFreq
- MaxFreq
- NumberOfPoints
- Calibrated
- ID
- PortAExtension
- PortBExtension
- Analyzer
- Operator
- Location

[For 4-port ECal]

- ModelNumber
- SerialNumber
- PortAConnector
- PortBConnector
- PortCConnector
- PortDConnector
- MinFreq
- MaxFreq

- NumberOfPoints
- Calibrated
- ID
- PortAExtension
- PortBExtension
- PortCExtension
- PortDExtension
- Analyzer
- Operator
- Location

Examples

```
Dim Cal_Info As String
Cal_Info = SCPI.SENS1.CORRection.COLLect.ECAL.INFormation
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:INFormation?

Query response

{Model number,Serial number,Connector type, Calibration date, Min and max frequency}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:INF?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ISOLation.STATe

Description

From revision A.9.00, this command ignores ON/OFF of the isolation measurement when executing Ecal for the selected channel (*Ch*), because the isolation of the ENA is better than that of the ECal. This command takes no action and only exists to maintain backward compatibility.

Variable

Parameter	*Status*
Description	ON/OFF of the isolation measurement when executing ECal
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the isolation measurement. • False or OFF: Turns OFF the isolation measurement.
Preset value	False or OFF

Examples

```
Dim EcalIso As Boolean
SCPI.SENSe1.CORRection.COLLect.ECAL.ISOLation.STATe = False
EcalIso = SCPI.SENSe1.CORRection.COLLect.ECAL.ISOLation.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT1

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT2

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT3

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.SOLT4

Equivalent key

No equivalent key is available on front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ISOLation[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ISOLation[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ISOL.STATE OFF"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ISOL.STATE?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.ORIentation.STATe

Description

This command turns ON/OFF the ECal auto detect function.

Variable

Parameter	*Status*
Description	ON/OFF the ECal auto detect funcion.
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the auto detect function. • False or OFF: Turns OFF the auto detect function.
Preset value	True or ON

For information on the variable (*Ch*), see *Ch*.

Examples

```
Dim EcalOri As Boolean
SCPI.SENSe(1).CORRection.COLLect.ECAL.ORIentation.STATe = True
EcalOri = SCPI.SENSe(1).CORRection.COLLect.ECAL.ORIentation.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(Cpt)

SCPI.CALCulate(Ch).FSIMulator.SENDed.DEEMbed.PORT(Pt).TYPE

Equivalent key

Cal > **ECal** > **Orientation**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ORIentation[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:COLLect:ECAL:ORIentation[:STATe]?

Query response

{0|1}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ORI ON"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:ORI?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.PATH(*Cpt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(*Cpt*) = *Value*

Value = SCPI.SENSe(Ch).CORRection.COLLect.ECAL.PATH(*Cpt*)

Description

This command set/get the ECal module n port number which is connected to a specified port.

Variable

Parameter	*Value*
Description	Port of ECal module.
Data type	Long integer type (Long)
Range	One of the following is set/get: • 0: Nothing is connected. • 1: Port A is connected. • 2: Port B is connected. • 3: Port C is connected. • 4: Port D is connected.

For information on the variable (*Cpt*), see *Cpt*.

Examples

```
Dim ECalPort As Long
SCPI.SENSe.CORRection.COLLect.ECAL.PATH(1) = 3
ECalPort = SCPI.SENSe.CORRection.COLLect.ECAL.PATH(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ECAL.ORIentation.STATe

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:PATH <numeric 1>,<numeric 2>

:SENSe{[1]-160}:CORRection:COLLect:ECAL:PATH? <numeric 2>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:PATH 1,2"

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT1

Object type

Property (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT1 = *Eport*

Description

This command executes 1-port calibration of the specified port of selected channel (*Ch*) using the ECal (Electronic Calibration) module.

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	*Eport*
Description	Port number
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT1 = 1
```

Equivalent key

Cal > **ECal** > **1-Port Cal** > **Port 1**|**Port 2**|**Port 3**|**Port 4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:SOLT1 <numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT1 1"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT2

Object type

Property (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT2 = *Eports*

Description

Executes full 2-port calibration between the specified 2 ports of channels 1 to 36 (*Ch*) using the ECal (Electronic Calibration) module.

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 2-element array data (port number). • *EPorts(0)* *EPorts*(1) Specifies the port numbers for 2-port ECal. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. the order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT2 = Array(1,2)
```

```
Dim EcalPort(1) As Variant
EcalPort(0) = 1
EcalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT2 = EcalPort
```

Equivalent key

Cal > ECal > 2-Port Cal > Port 1-2|Port 1-3|Port 1-4|Port 2-3|Port 2-4|Port 3-4

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:SOLT2 <numeric
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT2 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT3

Object type

Property (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT3 = *Eports*

Description

This command executes full 3-port calibration between the specified 3 ports of channels 1 to 36 (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the 4-port ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 3-element array data (port number). • *EPorts(0)* *EPorts*(1) *EPorts*(2) Specifies the port numbers for 3-port ECal. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to more than 2 port numbers, an error occurs when executed. the order of the 3 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT3 = Array(1,2,3)
```

Equivalent key

Cal > **ECal** > **3-Port Cal** > **Port 1-2-3**|**Port 1-2-4**|**Port 1-3-4**|**Port 2-3-4**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:SOLT3 <numeric
1>,<numeric 2>,<numeric 3>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT3 1,2,3"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

```
Dim EcalPort(2) As Variant
EcalPort(0) = 1
EcalPort(1) = 2
EcalPort(2) = 3
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT3 = EcalPort
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT4

Object type

Property (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.SOLT4 = *Eports*

Description

This command executes full 4-port calibration for selected channel (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the 4-port ECal module is not connected or when ports are not connected each other appropriately, an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 4-element array data (port number). • *EPorts(0)* *EPorts*(1) *EPorts*(2) *EPorts*(3) Specifies the port numbers for 4-port ECal. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to more than 2 port numbers, an error occurs when executed. the order of the 4 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT4 = Array(1,2,3,4)

Dim EcalPort(3) As Variant
EcalPort(0) = 1
EcalPort(1) = 2
```

```
EcalPort(2) = 3
EcalPort(3) = 4
SCPI.SENSe(1).CORRection.COLLect.ECAL.SOLT4 = EcalPort
```

Equivalent key

Cal > ECal > 4-Port Cal

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:SOLT4 1,2,3,4
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT4 1,2,3,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.THRU

Object type

Property (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.THRU = *Eports*

Description

This command executes response calibration (thru) between the specified 2 ports of selected channel (*Ch*) using the ECal (Electronic Calibration) module.

If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, , an error occurs and the command is ignored.

Variable

Parameter	*Eports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port number. • *Ports(1):* Specifies the stimulus port number. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.ECAL.THRU = Array(1,2)
Dim EcalPort(1) As Variant
EcalPort(0) = 1
```

```
EcalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.ECAL.THRU = EcalPort
```

Equivalent key

Cal > ECal > Thru Cal > 2-1 (S21)|3-1 (S31)|4-1 (S41)|1-2 (S12)|3-2 (S32)| 4-2 (S42)| 1-3 (S13)|2-3 (S23)|4-3 (S43)|1-4 (S14)|2-4 (S24)|3-4 (S34)

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:THRU <numeric
1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:SOLT2 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UCHar

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UCHar = *Param*

Description

This command sets the ECal characteristic used when executing the user-defined ECal, for the selected channel (*Ch*).

- The user-defined ECal is a type of ECal that is executed using the characteristic that has been acquired by the user and stored in the memory for ECal.

  When the ECal module is not connected or the characteristic is not stored at the specified location number, executing this object will cause a runtime error.
- This command is available with the Firmware revision 3.50 or higher.

Variable

Parameter	*Param*
Description	Characteristic used when executing ECal (user characterization)
Data type	Character string type (String)
Range	Select from either of the following: • "CHAR0": Uses the factory-default characteristic. (Normal ECal) • "CHAR1": Uses the characteristic stored at location number 1 in the ECal's flash memory. • "CHAR2": Uses the characteristic stored at location number 2 in the ECal's flash memory. • "CHAR3": Uses the characteristic stored at location number 3 in the ECal's flash memory. • "CHAR4": Uses the characteristic stored at location number 4 in the ECal's flash memory. • "CHAR5": Uses the characteristic stored at

	location number 5 in the ECal's flash memory.
Preset value	"CHAR0"

Examples

```
Dim UserChar As String
SCPI.SENSe(1).CORRection.COLLect.ECAL.UCHar = "CHAR2"
UserChar = SCPI.SENSe(1).CORRection.COLLect.ECAL.UCHar
```

Equivalent key

Cal > **ECal** > **Characterization** > **Factory**|**User1**|**User2**|**User3**|**User4**|**User5**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:ECAL:UCHar {CHAR0| CHAR1
|CHAR2|CHAR3|CHAR4|CHAR5}
:SENSe{[1]-160}:CORRection:COLLect:ECAL:UCHar?
```

Query response

```
{CHAR0|CHAR1|CHAR2|CHAR3|CHAR4|CHAR5}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UCH CHAR1"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UCH?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UTHRU.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UTHRU.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.COLLect.ECAL.UTHRU.STATe

Description

This command sets Unknown Thru On/Off for Ecal.

Variable

Parameter	*Status*
Description	Sets/gets Unknown Thru ON/Off for Ecal
Data type	Boolean
Range	Select from either of the following: • True or ON: Turns ON the Unknown Thru function. • False or OFF: Turns OFF the Unknown Thru function.

Examples

```
Dim EcalPort As Boolean
EcalPort = 0
SCPI.SENSe(1).CORRection.COLLect.ECAL.UTHRU.STATe = 0
EcalPort=SCPI.SENSe(1).CORRection.COLLect.ECAL.UTHRU.STATe
```

Equivalent key

Cal > **ECal** > **Unknown Thru**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:ECAL:UTHRU:STATe {ON|OFF|1|0}

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UTHRU:STAT 1"
20 OUTPUT 717;":SENS1:CORR:COLL:ECAL:UTHRU:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ADAPter.REMoval

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ADAPter.REMoval = *Port*

Description

This command sets the port in which adaptor needs to be added/removed.

Variable

Parameter	*Port*
Description	Port in which Adaptor needs to be added/removed
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1

Examples

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ADAPter.REMoval = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.ADAPter(Pt).LENGth

Equivalent key

Cal > **Calibrate** > **Adapter Removal** > **Select Port**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:ADAPter:REMoval {1-4}

Example of use

```
Call viVPrintf(vi, ":SENS1:CORR:COLL:METH:ADAPter:REMoval  1"+vbcr, 0)
```

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ERESponse

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(Ch).CORRection.COLLect.METHod.ERESponse = Ports

Description

This command sets the calibration type to the enhanced response calibration between the two specified ports, for the selected channel.

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies the response port. • *Ports(1):* Specifies the stimulus port. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	For each parameter, you must specify a different port number. If you specify the same port number for 2 or more parameters, an error occurs and the command is ignored.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim ERESPort(1) As Variant
ERESPort(0) = 1
ERESPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.METHod.ERESponse = ERESPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Enhanced Response** > **Select Ports** > **2-1 (S21 S11)**|**3-1 (S31 S11)**|…|**3-4 (S34 S44)**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:ERESponse <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:ERES 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.OPEN

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.OPEN = *Port*

Description

This command sets the calibration type to the response calibration (open) of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.OPEN = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Response (Open) > Select Port**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod[:RESPonse]:OPEN <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:OPEN 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.SHORt

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.SHORt = *Port*

Description

This command sets the calibration type to the response calibration (short) of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.SHORt = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Response (Short) > Select Port**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod[:RESPonse]:SHORt <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SHOR 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.THRU

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.RESPonse.THRU = *Ports*

Description

This command sets the calibration type to the response calibration (thru) between the specified 2 ports, for the selected channel (*Ch*).

Variable

For information on the variable (*Ch*) and the variable (*Port*), see Ch and Port.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.THRU = Array(2,1)
```

```
Dim ThruPort(1) As Variant
ThruPort(0) = 2
ThruPort(1) = 1
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.THRU = ThruPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **Response (Thru) > Select Ports**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:COLLect:METHod[:RESPonse]:THRU
<numeric 1>,<numeric 2>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:THRU 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT1

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT1 = *Port*

Description

This command sets the calibration type to the 1-port calibration of the specified port, for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT1 = 1

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **1-Port Cal > Select Port**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT1 1"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT2

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT2 = *Ports*

Description

This command sets the calibration type to the full 2-port calibration between the specified 2 ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for full 2-port calibration. • *Ports(1):* Specifies the other port for full 2-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT2 = Array(1,2)

Dim CalPort(1) As Variant
CalPort(0) = 1
CalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT2 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > Calibrate > 2-Port Cal > Select Ports

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:SOLT2 <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT2 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT3

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT3 = *Ports*

Description

This command sets the calibration type to the full 3-port calibration between the specified 3 ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 3-element array data (port number). • *Ports(0):* Specifies a port for full 3-port calibration. • *Ports(1):* Specifies a port for full 3-port calibration. • *Ports(2):* Specifies a port for full 3-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 3 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT3 = Array(1,2,3)

Dim CalPort(2) As Variant
CalPort(0) = 1
CalPort(1) = 2
CalPort(2) = 3
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT3 = CalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **3-Port Cal > Select Ports**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:SOLT3 <numeric 1>, <numeric 2>,<numeric 3>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT3 1,2,3"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT4

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.SOLT4 = *Ports*

Description

This command sets the calibration type to the full 4-port calibration, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port number). • *Ports(0):* Specifies a port for full 4-port calibration. • *Ports(1):* Specifies a port for full 4-port calibration. • *Ports(2):* Specifies a port for full 4-port calibration. • *Ports(3):* Specifies a port for full 4-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 4 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT4 = Array(1,2,3,4)

Dim CalPort(3) As Variant
CalPort(0) = 1
CalPort(1) = 2
```

CalPort(2) = 3
CalPort(3) = 4
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT4 = CalPort

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TYPE

Equivalent key

Cal > **Calibrate** > **4-Port Cal**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:SOLT4 1,2,3,4

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:SOLT3 1,2,3,4"

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TRL2

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TRL2 = *Ports*

Description

This command sets the calibration type to the TRL calibration between the 2 specified ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 2-element array data (port number). • *Ports(0):* Specifies a port for TRL 2-port calibration. • *Ports(1):* Specifies a port for TRL 2-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 2 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.TRL2 = Array(1,2)
```

```
Dim TrlCalPort(1) As Variant
TrlCalPort(0) = 1
TrlCalPort(1) = 2
SCPI.SENSe(1).CORRection.COLLect.METHod.TRL2 = TrlCalPort
```

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3
SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4
SCPI.SENSe(Ch).CORRection.COLLect.SAVE
SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

Cal > **Calibrate** > **2-Port TRL Cal** > **Select Ports** > **1-2**|**1-3**|**1-4**|**2-3**|**2-4**|**3-4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect
:METHod:TRL2 <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TRL2 1,2"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TRL3

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TRL3 = *Ports*

Description

This command sets the calibration type to the TRL calibration between the 3 specified ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 3-element array data (port number). • *Ports(0):* Specifies a port for TRL 3-port calibration. • *Ports(1):* Specifies a port for TRL 3-port calibration. • *Ports(2):* Specifies a port for TRL 3-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 3 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.TRL3 = Array(1,2,3)

Dim TrlCalPort(2) As Variant
TrlCalPort(0) = 1
TrlCalPort(1) = 2
```

TrlCalPort(2) = 3
SCPI.SENSe(1).CORRection.COLLect.METHod.TRL3 = TrlCalPort

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

Cal > **Calibrate** > **3-Port TRL Cal** > **Select Ports** > **1-2-3**|**1-2-4**|**1-3-4**|**2-3-4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect
:METHod:TRL3 <numeric 1>,<numeric 2>,<numeric 3>

Example of use

10 OUTPUT 717;":SENS1:CORR:COLL:METH:TRL3 1,2,3"

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TRL4

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CCORRection.COLLect.METHod.TRL4 = *Ports*

Description

This command sets the calibration type to the TRL calibration between the 4 specified ports, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Indicates 4-element array data (port number). • *Ports(0):* Specifies a port for TRL 4-port calibration. • *Ports(1):* Specifies a port for TRL 4-port calibration. • *Ports(2):* Specifies a port for TRL 4-port calibration. • *Ports(3):* Specifies a port for TRL 4-port calibration. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Resolution	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. If you specify the same port number to 2 or more port numbers, an error occurs when executed. The order of the 4 port numbers to be specified is arbitrary.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.TRL4 = Array(1,2,3,4)

Dim TrlCalPort(3) As Variant
TrlCalPort(0) = 1
TrlCalPort(1) = 2
```

TrlCalPort(2) = 3
TrlCalPort(3) = 4
SCPI.SENSe(1).CORRection.COLLect.METHod.TRL4 = TrlCalPort

Related objects

SCPI.SENSe(Ch).CORRection.COEFficient.DATA

SCPI.SENSe(Ch).CORRection.COEFficient.SAVE

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

Cal > **Calibrate** > **4-Port TRL Cal**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect
:METHod:TRL4 <numeric 1>,<numeric 2>,<numeric 3>,<numeric 4>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TRL4 1,2,3,4"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TYPE

Object type

Property (**Read Only**)

Syntax

Param = SCPI.SENSe(*Ch*).CORRection.COLLect.METHod.TYPE

Description

This command reads the selected calibration type of selected channel (*Ch*).

- This object is used to check the selected calibration type for calculating the calibration coefficients. To check the applied calibration type (error correction on), use the SCPI.SENSe(Ch).CORRection.TYPE(Tr) object.

Variable

Parameter	*Param*
Description	Calibration type
Range	AREM – Adaptor Removal ERES – Enhanced response calibration NONE – None RESPO – Response calibration (open) RESPS – Response calibration (short) RESPT – Response calibration (thru) SOLT1 – 1-port calibration. SOLT2 – Full 2-port calibration. SOLT3 – Full 3-port calibration. SOLT4 – Full 4-port calibration.

	TRL2	TRL 2-port calibration.
	TRL3	TRL 3-port calibration.
	TRL4	TRL 4-port calibration.
Preset Value	NONE	
Data type	Character string type (String)	

Examples

```
Dim CalType As String
CalType = SCPI.SENSe(1).CORRection.COLLect.METHod.TYPE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

SCPI.SENSe(Ch).CORRection.TYPE(Tr)

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:METHod:TYPE?

Query response

{AREM|ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|SOLT3|SOLT4|TRL2|TRL3|TRL4}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:METH:TYPE?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.COLLect.PARTial.SAVE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.PARTial.SAVE

Description

This command, used for partial overwrite, recalculates the calibration coefficients depending on the selected calibration type from the measured calibration data.

- Calculating the calibration coefficients clears all calibration data regardless of whether they are used for the calculation and also clears the calibration type selections.

  If partial overwrite is executed before selecting the calibration type, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

SCPI.SENSe(1).CORRection.COLLect.PARTial.SAVE

Equivalent key

Cal > **Calibrate** > **n-Port Cal** > **Overwrite**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:PARTial:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:PART:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.SAVE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.SAVE

Description

This command calculates the calibration coefficients depending on the calibration type selection, from the measured calibration data.

- Calculating the calibration coefficients clears all the measured calibration data whether or not used for the calculation and also clears the calibration type selection.

  If this command is executed before all necessary calibration data for calculating the calibration coefficients is measured, an error occurs and the command is ignored.

Variable

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Dmy As Long
SCPI.SENSe(1).CORRection.COLLect.METHod.RESPonse.THRU = Array(2,1)
SCPI.SENSe(1).CORRection.COLLect.ACQuire.THRU = Array(2,1)
Dmy = SCPI.IEEE4882.OPC
SCPI.SENSe(1).CORRection.COLLect.SAVE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.OPEN

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.SHORt

SCPI.SENSe(Ch).CORRection.COLLect.METHod.RESPonse.THRU

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT4

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4

Equivalent key

Cal > **Calibrate** > **Response**|**n-Port Cal** > **Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.COLLect.SIMPlified.SAVE

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.COLLect.SIMPlified.SAVE

Description

This command calculates the calibration coefficients for the simple full 3 port calibration or the simple full 4 port calibration from the measured calibration data, when the full 3/4 port calibration is selected as the calibration type.

- If the response calibration or the 1/2 port calibration is selected as the calibration type, this command behaves similarly to SCPI.SENSe(Ch).CORRection.COLLect.SAVE.

  After the calibration coefficients are calculated, the measured data and the calibration type setting are cleared.

  If this command is executed before all the necessary calibration data for calculating the calibration coefficients for the simple full 3 port calibration or the simple full 4 port calibration is measured, a runtime error occurs.

  This function is available with the Firmware revision 3.50 or greater.

Variable

For information on the variable (*Ch*), refer to Ch.

Examples

```
SCPI.SENSe(1).CORRection.COLLect.METHod.SOLT3 = Array(1,2,3)
SCPI.SENSe(1).CORRection.COLLect.SIMPlified.SAVE
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.SOLT4

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4

SCPI.SENSe(Ch).CORRection.COLLect.SAVE

Equivalent key

Cal > **Calibrate** > **n-Port Cal**|**n-Port TRL** > **Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:COLLect:SIMPlified:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:COLL:SIMP:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.CONFig

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.CONFig = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.CONFig

Description

This command sets/gets the frequency point to calculate the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	The frequency point to calculate the auto port extension
Data type	Character string type (String)
Range	Select from either of the following: • "CSPN": Uses the frequency of the current sweep range. • "AMKR": Use the frequency of the active marker. This is applied to Loss 1 and Loss 2 is ignored. • "USPN": This is executed with the arbitrary specified start frequency and stop frequency.
Preset value	"CSPN"

Examples

```
Dim Conf As String
SCPI.SENSe(1).CORRection.EXTension.AUTO.CONFig = "AMKR"
Conf = SCPI.SENSe(1).CORRection.EXTension.AUTO.CONFig
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Method** > **Current Span**|**Active Marker**|**User Span**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:AUTO:CONFig {CSPN|AMKR|USPN}

:SENSe{[1]-160}:CORRection:EXTension:AUTO:CONFig?

Query response

{CSPN|AMKR|USPN}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:CONF CSPN"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:CONF?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.DCOFfset

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.DCOFfset = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.DCOFfset

Description

This command enables/disables or gets the usage of DC Offset value for the results of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the usage of DC Offset value for the results of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Uses the DC Offset value for the results. • False or OFF: Does not use the DC Offset value for the results.
Preset value	False or OFF

Examples

```
Dim Dcof As Boolean
SCPI.SENSe(1).CORRection.EXTension.AUTO.DCOFfset = True
Dcof = SCPI.SENSe(1).CORRection.EXTension.AUTO.DCOFfset
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

SCPI.CALCulate(Ch).SELected.CONVersion.FUNCtion

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Adjust Mismatch**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:AUTO:DCOFfset {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:EXTension:AUTO:DCOFfset?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:DCOF ON"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:DCOF?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.LOSS

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.LOSS = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.LOSS

Description

This command turns ON/OFF or gets the status of the loss compensation for the results of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF the loss compensation for the results of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the loss compensation • False or OFF: Turns off the loss compensation
Preset value	False or OFF

Examples

```
Dim AutoLoss As Boolean
SCPI.SENSe(1).CORRection.EXTension.AUTO.LOSS = True
AutoLoss = SCPI.SENSe(1).CORRection.EXTension.AUTO.LOSS
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > Port Extensions > Auto Port Extension > Include Loss

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:AUTO:LOSS {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:EXTension:AUTO:LOSS?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:LOSS ON"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:LOSS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.MEASure

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.MEASure = *Param*

Param = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.MEASure

Description

This command measures the calibration data of the OPEN standard or SHORT standard of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Standard type of the auto port extension
Data type	Character string type (String)
Range	Select from either of the following: • "OPEN": Measures the calibration data of the OPEN standard • "SHORt": Measures the calibration data of the SHORT standard
Preset value	"SHORt"

Examples

```
Dim AutoMeas As String
SCPI.SENSe(1).CORRection.EXTension.AUTO.MEASure = "OPEN"
AutoLoss = SCPI.SENSe(1).CORRection.EXTension.AUTO.LOSS
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > Port Extensions > Auto Port Extension > Measure OPEN|Measure Short- All|Port 1|Port 2|Port 3|Port 4

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:AUTO:MEASure {OPEN|SHORt}
:SENSe{[1]-160}:CORRection:EXTension:AUTO:MEASure?
```

Query response

```
{OPEN|SHOR}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:MEAS OPEN"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:MEAS?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*) = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.PORT(*Pt*)

Description

This command turns ON/OFF or gets the status of the auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	On/off of the auto port extension
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the auto port extension • False or OFF: Turns OFF the auto port extension
Preset value	True or ON

For information on the variable (*Ch*) or Pt, refer to Ch or Pt.

Examples

```
Dim APort As Boolean
SCPI.SENSe(1).CORRection.EXTension.AUTO.PORT(1) = True
APort = SCPI.SENSe(1).CORRection.EXTension.AUTO.PORT(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Select Ports** > **Port 1**|**Port 2**|**Port 3**|**Port 4**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:AUTO:PORT{[1]|2|3|4} {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:EXTension:AUTO:PORT{[1]|2|3|4}?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:PORT1 ON"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:PORT1?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.RESet

Object type

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.RESet

Description

This command deletes the finished measurement data (OPEN and SHORT), for the selected channel (*Ch*).

Examples

SCPI.SENSe(1).CORRection.EXTension.AUTO.RESet

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:AUTO:RESet

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:RESet"
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STARt = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STARt

Description

This command get/set the start frequency within the frequency range of the user specified auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	3E5
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AStart As Double
SCPI.SENSe(1).CORRection.EXTension.AUTO.STARt = 1E9
AStart = SCPI.SENSe(1).CORRection.EXTension.AUTO.STARt
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure
SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)
SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet
SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STOP

Equivalent key

Cal > Port Extensions > Auto Port Extension > Method > User Span Start

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:AUTO:STARt <numeric>
:SENSe{[1]-160}:CORRection:EXTension:AUTO:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STAR 1.2E9"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STAR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STOP = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.AUTO.STOP

Description

This command sets/gets the stop frequency within the frequency range of the user specified auto port extension, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	Maximum frequency in the range
Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AStop As Double
SCPI.SENSe(1).CORRection.EXTension.AUTO.STOP = 250E8
AStop = SCPI.SENSe(1).CORRection.EXTension.STOP.STARt
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.CONFig

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)
SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet
SCPI.SENSe(Ch).CORRection.EXTension.AUTO.STARt

Equivalent key

Cal > **Port Extensions** > **Auto Port Extension** > **Method** > **User Span Stop**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:AUTO:STOP <numeric>
:SENSe{[1]-160}:CORRection:EXTension:AUTO:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STOP 1.5E9"
20 OUTPUT 717;":SENS1:CORR:EXT:AUTO:STOP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).FREQuency(*Fq*)

Description

This command sets/gets the frequency used for calculation of the loss value of the frequency 1 and 2 (*Fq*) of the selected port (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Fq*
Description	Frequency number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	Frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	0

Unit	Hz (hertz)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PortFreq As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).FREQuency(1) = 500E6
PortFreq = SCPI.SENSe(1).CORRection.EXTension.PORT(1).FREQuency(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3** | **Extension Port4** > **Loss** > **Freq1**|**Freq2**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:FREQuency{[1]|2}
<numeric>
:SENSe{[1]-
160}:CORRection:EXTension:PORT{[1]|2|3|4}:FREQuency{[1]|2}?
```

Query response

```
<numeric><newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:FREQ1 10E6"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:FREQ1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*II*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*II*).STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).INCLude(*II*).STATe

Description

This command turns ON/OFF the set of loss value and frequency value of include 1 and 2 (*II*) of the port 1 to 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*II*
Description	Include number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Status*
Description	ON/OFF the set of loss value and frequency value.
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the loss value and frequency value. • False or OFF: Turns OFF the loss value and

	frequency value.
Preset value	False or OFF

Examples

```
Dim PortIncl As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).INCLude(1).STATe = 500E6
PortIncl = SCPI.SENSe(1).CORRection.EXTension.PORT(1).INCLude(1).STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3** | **Extension Port4** > **Loss** > **Loss1**|**Loss2**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:INCLude{[1]|2}[:STATe] {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:INCLude{[1]|2}[:STATe]?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:INCL1 ON"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:INCL1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LDC

Description

This command sets/gets the DC loss value of the port 1 to 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	The loss value of DC.
Data type	Double precision floating point type (Double)
Range	-90 to 90
Preset value	0
Unit	dBm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PLdc As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).LDC = 45
PLdc = SCPI.SENSe(1).CORRection.EXTension.PORT(1).LDC
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3** | **Extension Port4** > **Loss** > **Loss at DC**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LDC <numeric>

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LDC?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LDC 1.2"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LDC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(Loss)

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Loss*) = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).LOSS(*Loss*)

Description

This command sets/gets the loss value of the loss 1 to 4 of the port 1 to 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Loss*
Description	Loss number
Data type	Long integer type (Long)
Range	1 to 2
Preset value	1
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Parameter	*Value*
Description	The loss value
Data type	Double precision floating point type (Double)
Range	-90 to 90
Preset value	0

Unit	dBm
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PLoss As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).LOSS(1) = -45
PLoss = SCPI.SENSe(1).CORRection.EXTension.PORT(1).LOSS(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3** | **Extension Port4** > **Loss** > **Loss1**|**Loss2**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LOSS{[1]|2} <numeric>

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:LOSS{[1]|2}?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LOSS1 0.8"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:LOSS1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.EXTension.PORT(*Pt*).TIME

Description

This command sets/gets or gets the status of the delay time for the port extension of ports 1 and 4 (*Pt*), for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Delay time
Data type	Double precision floating point type (Double)
Range	-10 to 10
Preset value	0
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PortExt As Double
SCPI.SENSe(1).CORRection.EXTension.PORT(1).TIME = 1E-3
PortExt = SCPI.SENSe(1).CORRection.EXTension.PORT(1).TIME
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)

Equivalent key

Cal > **Port Extensions** > **Extension Port1** | **Extension Port2** | **Extension Port3** | **Extension Port4** > **Coax. Extension**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:TIME <numeric>

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:TIME?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:TIME 1e-3"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:TIME?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff = *Cfreq*

Cfreq = SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

Description

This command sets/gets cutoff frequency of the waveguide of the selected port (*Pt*), for the selected channel (Ch).

Variable

Parameter	*Cfreq*
Description	Waveguide cutoff frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Unit	Hz
Preset value	0 Hz

Examples

```
Dim CFreq As Double
SCPI.SENSe(2).CORRection.EXTension.PORT(1).WAVeguide.CUToff = 1E5
CFreq = SCPI.SENSe(1).CORRection.EXTension.PORT(3).WAVeguide.CUToff
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME

Equivalent key

Cal > **Port Extensions** > **Extension Port 1**|**Extension Port 2**|**Extension Port 3**|**Extension Port 4** > **Cutoff Frequency**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:CUToff <numeric>

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:CUToff?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:CUT 1e3"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:CUT?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME = *Value*

Value = SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.TIME

Description

This command sets/gets the port extension of the selected port (*Pt*), for the selected channel (Ch).

Variable

Parameter	*Value*
Description	Port extension
Data type	Double precision floating point type (Double)
Range	-10 to +10
Unit	Second
Preset value	0 second

Examples

```
Dim WgTime As Double
SCPI.SENSe(2).CORRection.EXTension.PORT(1).WAVeguide.TIME = 5
WgTime = SCPI.SENSe(1).CORRection.EXTension.PORT(3).WAVeguide.TIME
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).WAVeguide.CUToff

Equivalent key

Cal > **Port Extensions** > **Extension Port 1**|**Extension Port 2**|**Extension Port 3**|**Extension Port 4** > **Waveguide**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:TIME <numeric>

:SENSe{[1]-160}:CORRection:EXTension:PORT{[1]|2|3|4}:WAVeguide:TIME?

Query response

<numeric><newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:TIME -4"
20 OUTPUT 717;":SENS1:CORR:EXT:PORT1:WAV:TIME?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.EXTension.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.EXTension.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.EXTension.STATe

Description

This command turns ON/OFF or returns the status of the port extension, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the port extension correction
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Turns ON the port extension. • False or OFF: Turns OFF the port extension.
Preset value	False or OFF

Examples

```
Dim Ext As Boolean
SCPI.SENSe(1).CORRection.EXTension.STATe = True
Ext = SCPI.SENSe(1).CORRection.EXTension.STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.DCOFfset

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.LOSS

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.MEASure

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.PORT(Pt)

SCPI.SENSe(Ch).CORRection.EXTension.AUTO.RESet

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).FREQuency(Fq)

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).INCLude(Il).STATe

SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LDC
SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).LOSS(Loss)
SCPI.SENSe(Ch).CORRection.EXTension.PORT(Pt).TIME

Equivalent key

Cal > **Port Extensions** > **Extensions**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:EXTension[:STATe] {ON|OFF|1|0}
:SENSe{[1]-160}:CORRection:EXTension[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:EXT ON"
20 OUTPUT 717;":SENS1:CORR:EXT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.IMPedance.INPut.MAGNitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).CORRection.IMPedance.INPut.MAGNitude = *Value*

Value = SCPI.SENSe(Ch).CORRection.IMPedance.INPut.MAGNitude

Description

This command sets/gets the system characteristic impedance (Z0) value.

Variable

Parameter	*Value*
Description	System Z0 value
Data type	Double precision floating point type (Double)
Range	1E-3 to 1000
Preset value	50
Unit	ohm
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SysZ0 As Double
SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude = 75
SysZ0 = SCPI.SENSe.CORRection.IMPedance.INPut.MAGNitude
```

Equivalent key

Cal > **Set Z0**

Equivalent SCPI command

Syntax

:SENSe:CORRection:IMPedance[:INPut][:MAGNitude] <numeric>

:SENSe:CORRection:IMPedance[:INPut][:MAGNitude]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:CORR:IMP 75"
20 OUTPUT 717;":SENS:CORR:IMP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.CLEar

Description

This command clears the error coefficient for calibration when the frequency offset feature is on, for the selected channel (*Ch*).

- This command does not clear the error coefficient when the frequency offset mode is OFF.

Example of use

SCPI.SENSe(1).CORRection.OFFSet.CLEar

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.CLEar

Equivalent key

Cal > **Mixer/Converter Calibration** > **Clear** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:CLEar

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:CLE"
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.LOAD

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.LOAD = *Ports*

Description

This command measures the calibration data of the load standard of the specified port when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Provides 2-element array data (port). Ports(0): Measurement port number Ports(1): Frequency port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency and its subcommands" for the port specified by the frequency port number.

Example of use (1)

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.LOAD = Array(1,2)
```

Example of use (2)

```
Dim Port As Variant
Port(0) = 1
Port(1) = 2
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.LOAD = Port
```

Related objects

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Reflection** > **PortX@FreqY Broadband**

- The value of PortX and FreqY changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:LOAD <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:LOAD 1,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.OPEN

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.OPEN = *Ports*

Description

This command measures the calibration data of the open standard of the specified port when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Provides 2-element array data (port). Ports(0): Measurement port number Ports(1): Frequency port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency and its subcommands" for the port specified by the frequency port number.

Example of use (1)

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.OPEN = Array(1,2)
```

Example of use (2)

```
Dim Port As Variant
Port(0) = 1
Port(1) = 2
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.OPEN = Port
```

Related objects

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Reflection** > **PortX@FreqY Open**

- The value of PortX and FreqY changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:OPEN <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:OPEN 1,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.PMETer

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.PMETer = *Params*

Description

This command measures the scalar-mixer calibration data using the power meter when the frequency offset feature is *ON*, for the selected port and channel (*Ch*).

Variable

Parameter	*Params*
Description	Provides 3-element array data (port). Params(0): Measurement port number Params(1): Frequency port number Params(2): Power sensor number in use The index of the array starts from 0.
Data type	Variant type (Variant)
Range	Measurement port number/frequency port number 1 to 4 Power sensor number in use • "ASENsor": Specifies power sensor A. • "BSENsor": Specifies power sensor B.
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency and its subcommands" for the port specified by the frequency port number. The setting of the power sensor is common to that for power meter calibration.

Example of use

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.PMETer = Array(1,2,"ASENsor")
```

```
Dim Params As Variant
Params(0) = 1
Params(1) = 2
Params(2) = "ASENsor"
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.PMETer = Params
```

Related objects

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Power Meter** > **Use Sensor A|Use Sensor B**

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Power Meter** > **PortX@FreqY**

- The values of PortX and FreqY change depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:PMETer
<numeric 1>,<numeric 2>,{ASENsor|BSENsor}
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:PMET 1,4,ASEN"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.SHORt

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.SHORt = *Port*s

Description

This command measures the calibration data of the short standard of the specified port when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Provides 2-element array data (port). Ports(0): Measurement port number Ports(1): Frequency port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	The value used as the frequency is the value specified with frequency setting commands "SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency and its subcommands" for the port specified by the frequency port number.

Example of use

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.SHORt = Array(1,2)


Dim Port As Variant
Port(0) = 1
Port(1) = 2
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.SHORt = Port
```

Related objects

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Reflection** > **PortX@FreqY Short**

- The value of PortX and FreqY changes depending on the selected calibration method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:SHORt <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:SHOR 1,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.THRU

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ACQuire.THRU = *Ports*

Description

This command measures the calibration data of the thru standard of the specified port when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Provides 2-element array data (port). Ports(0): Response port number Ports(1): Stimulus port number The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	For example, when THRU 1,2 is specified, S22 and S12 are measured; when THRU 2,1 is specified, S11 and S21 are measured. If you specify the same port number to the 2 port numbers, an error occurs when executed.

Example of use

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.THRU = Array(1,2)


Dim Port As Variant
Port(0) = 1
Port(1) = 2
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ACQuire.THRU = Port
```

Related objects

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Transmission** > **PortX-Y@FreqZ Thru**

- The value of PortX-Y and FreqZ changes depending on the selected calibra-tion method and port number.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect[:ACQuire]:THRU <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:THRU 1,4"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.CLEar

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.CLEar

Description

This command clears the calibration measurement data when the frequency offset feature is *ON*, for the selected channel (*Ch*).

- This command also clears the measurement data of the power meter.

  Settings that have been temporarily changed due to measurement for each standard (number of traces, measurement parameters, and so on) return to their original values.

Example of use

SCPI.SENSe(1).CORRection.OFFSet.COLLect.CLEar

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.CLEar

Equivalent key

Cal > **Mixer/Converter Calibration** > **Scalar Cal(Manual)** > **Cancel** > **OK**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:CLEar

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:CLE"

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ECAL.SMIX2

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ECAL.SMIX2 = *Ports*

Description

This command executes calibration for the specified 2 ports when the frequency offset feature is on for selected channel (*Ch*) using ECal (Electronic Calibration).

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

Variable

Parameter	*Ports*
Description	Provides 2-element array data (port). Ports(0): Port number 1 Ports(1): Port number 2 The direction (forward or reverse) is determined by the presence/absence of power meter measurement data instead of port 1 or port 2 specified here. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	Before executing this command, the "SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.PMETer" command must be executed. If you specify the same port number to the 2 port numbers, an error occurs when executed.

Example of use

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ECAL.SMIX2 = Array(1,2)

Dim Port As Variant
Port(0) = 1
```

Port(1) = 2
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ECAL.SMIX2 = Port

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.PMETer

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ECAL.SOLT1

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(ECal)** > **ECal & Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:ECAL:SMIX2 <numeric 1>,<numeric 2>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:ECAL:SMIX2 1,2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ECAL.SOLT1

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.ECAL.SOLT1 = *EPort*

Description

This command executes 1-port calibration for the specified port when the frequency offset feature is on for selected channel (*Ch*) using ECal (Electronic Calibration).

- If this command is executed when the ECal module is not connected or when ports are not connected each other appropriately, a runtime error occurs.

Variable

Parameter	*EPort*
Description	Executes 1-port calibration for frequency offset measurement.
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use (1)

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.ECAL.SOLT1 = 1
```

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ECAL.SMIX2

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(ECal)** > **Select Ports** > **Port n**

Cal > **Mixer/Converter Caribration** > **Scalar Cal(ECal)** > **ECal & Done**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:OFFSet:COLLect:ECAL:SOLT1 <numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:ECAL:SOLT1 2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.METHod.SMIX2

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.METHod.SMIX2 = *Port*s

Description

This command executes 2-port calibration for the specified 2 ports when the frequency offset feature is on for selected channel (*Ch*).

Variable

Parameter	*Ports*
Description	Provides 2-element array data (port). Ports(0): Port number 1 Ports(1): Port number 2 The direction (forward, reverse, or both) is determined depending on the port number specified with the "SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD", "SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN", "SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt", and "SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.THRU" command instead of port number 1 or port number 2 specified here. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	1 to 4
Note	If you specify the same port number to the 2 port numbers, an error occurs when executed.

Example of use

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.METHod.SMIX2 = Array(1,2)


Dim Port As Variant
Port(0) = 1
```

Port(1) = 2
SCPI.SENSe(1).CORRection.OFFSet.COLLect.METHod.SMIX2 = Port

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.THRU

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Select Ports** > **2-1(fwd)|1-2(rev)|1-2(both**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:METHod:SMIX2 <numeric 1>,<numeric 2>

Example of use

10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:METH:SMIX2 1,2"

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.METHod.SOLT1

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.METHod.SOLT1 = *Port*

Description

Executes 1-port calibration for the specified port when the frequency offset feature is on for selected channel (*Ch*). (No read)

Variable

Parameter	*Port*
Description	Executes 1-port calibration for frequency offset measurement.
Data type	Long integer type (Long)
Range	1 to 4
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use (1)

```
SCPI.SENSe(1).CORRection.OFFSet.COLLect.METHod.SOLT1 = 2
```

Related objects

SCPI.SENSe(Ch).OFFSet.STATe

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.METHod.SMIX2

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Select Ports** > **Port n**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:METHod:SOLT1 <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:METH:SOLT1 2"
```

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.SAVE

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.OFFSet.COLLect.SAVE

Description

This command calculates the calibration coefficient for the selected calibration type from the calibration data measured with the frequency offset feature, for the selected channel (*Ch*).

- After the calibration coefficient is calculated, the measured data and the calibration type setting are cleared.

  If this command is executed before all necessary calibration data for calculating the calibration coefficient is measured, an error occurs when executed.

Example of use

SCPI.SENSe(1).CORRection.OFFSet.COLLect.SAVE

Related objects

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.METHod.SMIX2

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.METHod.SOLT1

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.LOAD

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.OPEN

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.SHORt

SCPI.SENSe(Ch).CORRection.OFFSet.COLLect.ACQuire.THRU

Equivalent key

Cal > **Mixer/Converter Caribration** > **Scalar Cal(Manual)** > **Done**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:OFFSet:COLLect:SAVE

Example of use

```
10 OUTPUT 717;":SENS1:CORR:OFFS:COLL:SAVE"
```

SCPI.SENSe(*Ch*).CORRection.PROPerty

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.PROPerty = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.PROPerty

Description

This command turns ON/OFF or returns the display of the calibration property, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the display of the calibration property
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the display of the calibration property. • False or OFF: Turns OFF the display of the calibration property.
Preset value	False or OFF

Examples

```
Dim CalProp As Boolean
SCPI.SENSe(1).CORRection.PROPerty = True
CalProp = SCPI.SENSe(1).CORRection.PROPerty
```

Equivalent key

Cal > **Property**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:PROPerty {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:PROPerty?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:PROP ON"
20 OUTPUT 717;":SENS1:CORR:PROP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.ACQuire = *Src*

Description

This command executes receiver calibration, for measurement the selected port (*Pt*) of selected channel (*Ch*).

- Since information of power calibration for both the measurement port and the source port is used for error coefficient calculation, the precision of receiver calibration is improved by executing power calibration for both ports before executing receiver calibration.

Variable

Parameter	*Src*
Description	Specifies the source port for receiver correction.
Data type	Long integer type (Long)
Range	1 to 4
Note	If you specify the same port number as the measurement port number, an error occurs when executed.

Example of use

SCPI.SENSe(1).CORRection.RECeiver(2).COLLect.ACQuire = 4

Related objects

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).STATe

Equivalent key

Cal > Receiver Calibration > Calibrate Both

- From Firmware rev 9.2, the **Take Cal Sweep** softkey is renamed to **Calibrate Both** softkey.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:RECeiver{[1]|2|3|4}:COLLect:ACQuire <numeric>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:COLL:ACQ 2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.RCHannel.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.RChannel.ACQuire

Description

This command executes receiver calibration (R channel only), for measurement the selected port (*Pt*) of selected channel (*Ch*).

- Since information of power calibration for both the measurement port and the source port is used for error coefficient calculation, the precision of receiver calibration is improved by executing power calibration for both ports before executing receiver calibration.
- When **Calibrate R** is selected, the **Select Port** setting is used and the **Source Port** setting is ignored.

Example of use

SCPI.SENSe(1).CORRection.RECeiver(2).COLLect.RCHannel.ACQuire

Related objects

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).STATe

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.ACQuire

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.TCHannel.ACQuire

Equivalent key

Cal > Receiver Calibration > Calibrate R

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:RECeiver{[1]|2|3|4}:COLLect:RCHannel:ACQuire

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:COLL:RCH:ACQ"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.TCHannel.ACQuire

Type of object

Method (**Write-only**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).COLLect.TChannel.ACQuire = *Src*

Description

This command executes receiver calibration (T channel only), for measurement the selected port (*Pt*) of selected channel (*Ch*).

- Since information of power calibration for both the measurement port and the source port is used for error coefficient calculation, the precision of receiver calibration is improved by executing power calibration for both ports before executing receiver calibration.
- When **Calibrate Both** or **Calibrate T** is selected, the source port should be different from measurement port.

Variable

Parameter	*Src*
Description	Specifies the source port for receiver correction.
Data type	Long integer type (Long)
Range	1 to 4
Note	If you specify the same port number as the measurement port number, an error occurs when executed.

Example of use

```
SCPI.SENSe(1).CORRection.RECeiver(2).COLLect.TCHannel.ACQuire = 4
```

Related objects

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).STATe

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.ACQuire

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.RCHannel.ACQuire

Equivalent key

Cal > Receiver Calibration > Calibrate T

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:CORRection:RECeiver{[1]|2|3|4}:COLLect:TCHannel:ACQuire
<numeric>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:COLL:TCH:ACQ 2"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RECeiver(Pt).OFFset.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RECeiver(Pt).OFFset.AMPLitude = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.RECeiver(Pt).OFFset.AMPLitude

Description

This command sets/gets power offset value for Receiver Calibration. Receiver calibration is done at the condition of (source power)+(Specified offset), for selected channel (*Ch*).

For example, if the source Power is -10 dBm, then:

- If specified offset is 0 dBm, then Source Power Reading after calibration would be -10 dBm.
- If specified offset is 10 dBm, then Source Power Reading after calibration would be 0 dBm.
- If specified offset is -10 dBm, then Source Power Reading after calibration would be -20 dBm.

Variable

Parameter	*Value*
Description	Specified Offset
Data type	Double precision floating point type (Double)
Range	-200 to 200
Preset value	0
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Vel As Double
SCPI.SENSe(1).CORRection.RECeiver(1).OFFset.AMPLitude = 50
Vel = SCPI.SENSe(1).CORRection.RECeiver(1).OFFset.AMPLitude
```

Equivalent key

Cal > **Receiver Calibration** > **Power Offset**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:CORRection:RECeiver([1]-4):OFFset:AMPLitude
<numeric>
:SENSe{[1]-160}:CORRection:RECeiver([1]-4):OFFset:AMPLitude?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1:OFF:AMP 50"
20 OUTPUT 717;":SENS1:CORR:REC1:OFF:AMP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.RECeiver(*Pt*).STATe

Description

This command turns ON/OFF or gets the error correction for receiver calibration, for the measurement the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	On/off of error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the receiver calibration error correction. • False or OFF: Turns off the receiver calibration error correction.
Preset value	False or OFF

Example of use

```
Dim Corr As Boolean
SCPI.SENSe(1).CORRection.RECeiver(1).STATe = True
Corr = SCPI.SENSe(1).CORRection.RECeiver(1).STATe
```

Related objects

SCPI.SENSe(Ch).CORRection.RECeiver(Pt).COLLect.ACQuire

Equivalent key

Cal > **Receiver Calibration** > **Correction**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:RECeiver{[1]|2|3|4}[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:RECeiver{[1]|2|3|4}[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:REC1 ON"
20 OUTPUT 717;":SENS1:CORR:REC1?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX = *Value*

Value = SCPI.SENSe(*Ch*).CORRection.RVELocity.COAX

Description

This command sets/gets the velocity factor, for selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Velocity factor
Data type	Double precision floating point type (Double)
Range	0 to 10
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Vel As Double
SCPI.SENSe(1).CORRection.RVELocity.COAX = 0.5
Vel = SCPI.SENSe(1).CORRection.RVELocity.COAX
```

Equivalent key

Cal > **Velocity Factor**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:RVELocity:COAX <numeric>

:SENSe{[1]-160}:CORRection:RVELocity:COAX?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:RVEL:COAX 0.7"
20 OUTPUT 717;":SENS1:CORR:RVEL:COAX?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.STATe

Description

This turns ON/OFF or gets the status of the error correction, for the active trace of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the error correction. • False or OFF: Turns OFF the error correction.
Preset value	False or OFF

Examples

```
Dim Corr As Boolean
SCPI.SENSe(1).CORRection.STATe = True
Corr = SCPI.SENSe(1).CORRection.STATe
```

Equivalent key

Cal > **Correction**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:STATe {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:STAT ON"
20 OUTPUT 717;":SENS1:CORR:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.TRIGger.FREE.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).CORRection.TRIGger.FREE.STATe = *Status*

Status = SCPI.SENSe(*Ch*).CORRection.TRIGger.FREE.STATe

Description

This command sets/gets the trigger source for calibration to internal (ON) or to the state of the trigger source (SCPI.TRIGger.SEQuence.SOURce) when measurement is made (OFF).

- When you change the trigger source during sweep, the sweep is canceled.

Variable

Parameter	*Status*
Description	ON/OFF of the trigger source at calibration
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Specifies internal for the trigger source for calibration. This corresponds to the softkey "Internal." • False or OFF: Matches the state of the trigger source when measurement is made. This corresponds to the softkey "System."
Preset value	True or ON

Examples

```
Dim TrigSour As Boolean
SCPI.SENSe(1).CORRection.TRIGger.FREE.STATe = False
TrigSour = SCPI.SENSe(1).CORRection.TRIGger.FREE.STATe
```

Equivalent key

Cal > **Cal Trigger Source** > **Internal**|**System**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:TRIGger:FREE[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:CORRection:TRIGger:FREE[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:CORR:TRIG:FREE OFF"
20 OUTPUT 717;":SENS1:CORR:TRIG:FREE?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

Object type

Property (**Read-only**)

Syntax

Data = SCPI.SENSe(*Ch*).CORRection.TYPE(*Tr*)

Description

This command reads the information (calibration type, port numbers) of the applied calibration coefficients for the actual error correction, for traces 1 to 36 (*Tr*) of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates 5 array data items (the calibration type and the port information to which the calibration is applied). • *Data(0):* The calibration type applied. For detail, refer to the Range section. • *Data(1):* The port number to which the calibration is applied (0 when the calibration type is NONE). • *Data(2)":* The port number to which the calibration is applied (0 when the calibration type is not SOLT2, SOLT3, or SOLT4). • *Data(3):* The port number to which the calibration is applied (0 when the calibration type is not SOLT3 or SOLT4). • *Data(4):* The port number to which the calibration is applied (0 when the calibration type is not SOLT4). The array index starts from 0.
Range	One of the following is read out as *Data(0)*. • "ERES": The enhanced response calibration is applied. • "NONE": Nothing is applied. • "RESPO": The response calibration (open) is applied. • "RESPS": The response calibration (short) is

	applied. • "RESPT": The response calibration (thru) is applied. • "SMIX2": The scalar-mixer calibration is applied. • "SOLT1": The 1-port calibration is applied. • "SOLT2": The full 2-port calibration is applied. • "SOLT3": The full 3-port calibration is applied. • "SOLT4": The full 4-port calibration is applied.
Data type	Variant type (Variant)

Examples

```
Dim CalType As Variant
CalType = SCPI.SENSe(1).CORRection.TYPE(1)
```

Related objects

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL2

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL3

SCPI.SENSe(Ch).CORRection.COLLect.METHod.TRL4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:CORRection:TYPE{[1]-36}?

Query response

{ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|SOLT3|SOLT4|SMIX2},{numeric 1},{numeric 2}, {numeric 3},{numeric 4}<newline><^END>

{ERES|NONE|RESPO|RESPS|RESPT|SOLT1|SOLT2|SOLT3|SOLT4|SMIX2}:

	Description
ERES	The enhanced response calibration is applied.
NONE	Any calibration is not applied.

RESPO	The response calibration (open) is applied.
RESPS	The response calibration (short) is applied.
RESPT	The response calibration (thru) is applied.
SMIX2	The scalar-mixer calibration is applied.
SOLT1	The 1-port calibration is applied.
SOLT2	The full 2-port calibration is applied.
SOLT3	The full 3-port calibration is applied.
SOLT4	The full 4-port calibration is applied.

{numeric 1}:

the calibration port number
(This parameter is 0 when the first parameter is NONE.)
the response port number when the SMIX2 is selected.

{numeric 2}:

the calibration port number
(This parameter is 0 when the first parameter is not ERES, RESPT, SOLT2, SOLT3, SOLT4 and SMIX2.)
the stimulus port number when the SMIX2 is selected.

{numeric 3}:

the calibration port number
(This parameter is 0 when the first parameter is not SOLT3 and SOLT4.)

{numeric 4}:

the calibration port number
(This parameter is 0 when the first parameter is not SOLT4.)

Example of use

```
10 OUTPUT 717;":SENS1:CORR:TYPE1?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).DATA.CORRdata(*Param*)

Object type

Property (**Read only**)

Syntax

Data = SCPI.SENSe(*Ch*).DATA.CORRdata(*Param*)

Description

This command gets S-Parameter data. Refer to Internal Data Processing.

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select the following: • "S<XY>" Where: x=1to 4 Y=1to 4
Preset value	"S11"

Parameter	*Data*
Description	Indicates the array data (corrected data array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. • *Data(n*2-2) :*Real part of the data (complex number) at the n-th measurement point. • *Data(n*2-1)* :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Note	If there is no array data of NOP (number of measurement point))* 2 when setting a corrected data array, an error occurs when executed and the object is ignored.

Examples

```
Dim CorData As Variant
CorData = SCPI.SENSe.DATA.CORRdata("S21")
```

Related objects

SCPI.SENSe(Ch).DATA.RAWData

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe:DATA:CORRdata?
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44}
```

Query response

```
{array}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS:DATA:CORR? S21"
20 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).DATA.RAWData(*Param*)

Object type

Property (**Read only**)

Syntax

Data = SCPI.SENSe(*Ch*).DATA.RAWData(*Param*)

Description

This command gets the raw data. Refer to Internal Data Processing.

Variable

Parameter	*Param*
Description	Measurement parameter
Data type	Character string type (String)
Range	Select the following: • "S<XY>" Where: x=1to 4 Y=1to 4
Preset value	"S11"

Parameter	*Data*
Description	Indicates the array data (raw data array) of NOP (number of measurement points)*2. Where n is an integer between 1 and NOP. • *Data(n*2-2) :*Real part of the data (complex number) at the n-th measurement point. • *Data(n*2-1)* :Imaginary part of the data (complex number) at the n-th measurement point. The index of the array starts from 0.
Data type	Variant type (Variant)

Note	If there is no array data of NOP (number of measurement point))* 2 when setting a raw data array, an error occurs when executed and the object is ignored.

Examples

```
Dim RawData As Variant
RawData = SCPI.SENSe.DATA.RAWData("S21")
```

Related objects

SCPI.SENSe(Ch).DATA.CORRdata

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe:DATA:RAWData?
{S11|S21|S31|S41|S12|S22|S32|S42|S13|S23|S33|S43|S14|S24|S34|S44}
```

Query response

{array}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:DATA:RAWD? S21"
20 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).FREQuency.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.CENTer = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.CENTer

Description

This command sets/gets the center value of the sweep range of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Center value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	(Maximum Frequency - 10E5)/2
Unit	Hz (hertz)
Resolution	0.5 or 1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Cntr As Double
SCPI.SENSe(1).FREQuency.CENTer = 2E9
Cntr = SCPI.SENSe(1).FREQuency.CENTer
```

Related objects

SCPI.SENSe(Ch).FREQuency.SPAN

Equivalent key

Center

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:FREQuency:CENTer <numeric>
:SENSe{[1]-160}:FREQuency:CENTer?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:CENT 2E9"
20 OUTPUT 717;":SENS1:FREQ:CENT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.CW

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.CW = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.CW

Description

This command sets/gets the fixed frequency (CW frequency) for the power sweep for channels 1 to 36 (*Ch*).

- This object provides the same function as the SCPI.SENSe(Ch).FREQuency.FIXed object.

Variable

Parameter	*Value*
Description	Fixed frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CwFreq As Double
SCPI.SENSe(1).FREQuency.CW = 1E9
CwFreq = SCPI.SENSe(1).FREQuency.CW
```

Related objects

SCPI.SENSe(Ch).FREQuency.FIXed

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Power** > **CW Freq**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency[:CW|FIXed] <numeric>

:SENSe{[1]-160}:FREQuency[:CW|FIXed]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ 1E9"
20 OUTPUT 717;":SENS1:FREQ?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.DATA

Object type

Property (**Read-only**)

Syntax

Data = SCPI.SENSe(Ch).FREQuency.DATA

Description

This command reads the frequencies at all measurement points of channels 1 to 36 (Ch).

Variable

Parameter	*Data*
Description	Indicates the array data (frequency) of NOP (number of measurement points). Where n is an integer between 1 and NOP. • *Data(n-1):* Frequency at the n-th measurement point The index of the array starts from 0.
Data type	Variant type (Variant)

Examples

```
Dim FreqData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
FreqData = SCPI.SENSe(1).FREQuency.DATA
```

Related objects

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency:DATA?

Query response

{numeric 1},□E/span> ,{numeric NOP}<newline><^END>

	Description

{numeric n}	Frequency at the n-th measurement point

Where NOP is the number of measurement points and n is an integer between 1 and NOP.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:FREQ:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).FREQuency.FIXed

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.FIXed = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.FIXed

Description

This command sets/gets the fixed frequency (CW frequency) for the power sweep for channels 1 to 36 (*Ch*).

- This object provides the same function as SCPI.SENSe(Ch).FREQuency.CW.

Variable

Parameter	*Value*
Description	Fixed frequency
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CwFreq As Double
SCPI.SENSe(1).FREQuency.FIXed = 1E9
CwFreq = SCPI.SENSe(1).FREQuency.FIXed
```

Related objects

SCPI.SENSe(Ch).FREQuency.CW

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Power** > **CW Freq**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency[:CW|FIXed] <numeric>

:SENSe{[1]-160}:FREQuency[:CW|FIXed]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ 1E9"
20 OUTPUT 717;":SENS1:FREQ?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.SPAN

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.SPAN = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.SPAN

Description

This command sets/gets the span value of the sweep range of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Span value
Data type	Double precision floating point type (Double)
Range	0 to (Maximum Frequency - 10E5)
Preset value	Maximum Frequency - 10E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Span As Double
SCPI.SENSe(1).FREQuency.SPAN = 1E9
Span = SCPI.SENSe(1).FREQuency.SPAN
```

Related objects

SCPI.SENSe(Ch).FREQuency.CENTer

Equivalent key

Span

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency:SPAN <numeric>

:SENSe{[1]-160}:FREQuency:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:SPAN 1E9"
20 OUTPUT 717;":SENS1:FREQ:SPAN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.STARt = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.STARt

Description

This command sets/gets the start value of the sweep range of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Start value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Start As Double
SCPI.SENSe(1).FREQuency.STARt = 100E6
Start = SCPI.SENSe(1).FREQuency.STARt
```

Related objects

SCPI.SENSe(Ch).FREQuency.STOP

Equivalent key

Start

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency:STARt <numeric>

:SENSe{[1]-160}:FREQuency:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:STAR 100E6"
20 OUTPUT 717;":SENS1:FREQ:STAR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).FREQuency.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).FREQuency.STOP = *Value*

Value = SCPI.SENSe(*Ch*).FREQuency.STOP

Description

This command sets/gets the stop value of the sweep range of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Stop value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	Maximum frequency in the range
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Stp As Double
SCPI.SENSe(1).FREQuency.STOP = 3E9
Stp = SCPI.SENSe(1).FREQuency.STOP
```

Related objects

SCPI.SENSe(Ch).FREQuency.STARt

Equivalent key

Stop

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:FREQuency:STOP <numeric>

:SENSe{[1]-160}:FREQuency:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:FREQ:STOP 100E6"
20 OUTPUT 717;":SENS1:FREQ:STOP?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(*Id*).COUNt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).COUNt

Description

This command reads the number of ports of the Multiport test set whose ID is 1 or 2 (*Id*).

Variable

Parameter	*Value*
Description	The number of ports
Data type	Long integer type (Long)
Range	28: E5092A 16: E5091A-016 9: E5091A-009 0: E5091A/E5092A is not connected.

Examples

```
Dim NPort As Long
NPort = SCPI.SENSe.MULTiplexer(1).COUNt
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:COUN?"
20 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(*Id*).DISPlay.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).DISPlay.STATe = *Status*

Status = SCPI.SENSe(Ch).MULTiplexer(*Id*).DISPlay.STATe

Description

This command turns ON/OFF or gets the status of the property display (the state of the port assignment) of the Multiport test set whose ID is *Id*.

Variable

Parameter	*Status*
Description	ON/OFF of the property display
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: Turns ON the property display. • False or OFF:Turns OFF the property display.
Preset value	False or OFF

Examples

```
Dim Prop As Boolean
SCPI.SENSe.MULTiplexer(1).DISPlay.STATe = True
Prop = SCPI.SENSe.MULTiplexer(1).DISPlay.STATe
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).CATalog

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).SELect

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Property**

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:DISPlay[:STATe] {ON|OFF|1|0}

:SENSe:MULTiplexer{[1]|2}:DISPlay[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:DISP ON"
20 OUTPUT 717;":SENS:MULT1:DISP?"
30 ENTER 717;A
```

E5071C

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.A.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.A.DATA

Description

This command sets/gets the output port data for group A with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output port data for group A
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0

Examples

```
Dim Obj As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.A.DATA =10
Obj = SCPI.SENSe.MULTiplexer1.OUTPut.A.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line A** > **Line** [1-8] <**LOW**|**HIGH**>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:A[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:A[:DATA]?

Query response

2420

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:A.DATA 10"
20 OUTPUT 717;":SENS:MULT1:OUTP:A.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.A.VOTLage.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.A.VOLTage.DATA

Description

This command sets/gets the output voltage of the group A with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output voltage for group A
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.A.VOLTage.DATA =0
Obj = SCPI.SENSe.MULTiplexer1.OUTPut.A.VOLTage.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line A** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:A.VOLTage.[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:A.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:A.VOLT.DATA 0"
20 OUTPUT 717;":SENS:MULT1:OUTP:A.VOLT.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.B.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.B.DATA

Description

This command sets/gets the output port data for group B with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output port data for group B
Data type	Long integer type (Long)
Range	0 to 15
Preset value	0

Examples

```
Dim Prop As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.B.DATA = 10
Prop = SCPI.SENSe.MULTiplexer1.OUTPut.B.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line B** > **Line** [1-4] <**LOW**|**HIGH**>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:B[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:B[:DATA]?

Query response

2424

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:B:DATA 10"
20 OUTPUT 717;":SENS:MULT1:OUTP:B:DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.B.VOTLage.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.B.VOLTage.DATA

Description

This command sets/gets the output voltage of the group B with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output voltage for group B
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.B.VOLTage.DATA =0
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.B.VOLTage.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line B** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:B.VOLTage.[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:B.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:B.VOLT.DATA 0"
20 OUTPUT 717;":SENS:MULT1:OUTP:B.VOLT.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.C.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.C.DATA

Description

This command sets/gets the output port data for group C with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output port data for group C
Data type	Long integer type (Long)
Range	0 to 15
Preset value	0

Examples

```
Dim Prop As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.C.DATA = 10
Prop = SCPI.SENSe.MULTiplexer(1).OUTPut.C.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line C** > **Line** [1-4] <**LOW**|**HIGH**>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:C[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:C[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:C.DATA 10"
20 OUTPUT 717;":SENS:MULT1:OUTP:C.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.C.VOTLage.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.C.VOLTage.DATA

Description

This command sets/gets the output voltage of the group C with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output voltage for group C
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.C.VOLTage.DATA =0
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.C.VOLTage.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line C** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:C.VOLTage.[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:C.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:C.VOLT.DATA 0"
20 OUTPUT 717;":SENS:MULT1:OUTP:C.VOLT.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.D.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.D.DATA

Description

This command sets/gets the output port data for group D with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output port data for group D
Data type	Long integer type (Long)
Range	0 to 15
Preset value	0

Examples

```
Dim Prop As Long
SCPI.SENSe.MULTiplexer(1).OUTPut.D.DATA = 10
Prop = SCPI.SENSe.MULTiplexer(1).OUTPut.D.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line D** > **Line** [1-4] <**LOW**|**HIGH**>

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:D[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:D[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:D.DATA 10"
20 OUTPUT 717;":SENS:MULT1:OUTP:D.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.D.VOLTage.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.D.VOTLage.DATA = *Value*

Value = SCPI.SENSe(Ch).MULTiplexer(*Id*).OUTPut.D.VOLTage.DATA

Description

This command sets/gets the output voltage of the group D with ID of the E5092A.

Variable

Parameter	*Value*
Description	Output voltage for group D
Data type	Double precision floating point type (Double)
Range	0 to 5.2 V
Resolution	10 mV
Preset value	0 V

Examples

```
Dim Obj As Double
SCPI.SENSe.MULTiplexer(1).OUTPut.D.VOLTage.DATA =0
Obj = SCPI.SENSe.MULTiplexer(1).OUTPut.D.VOLTage.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.A.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.B.VOLTage.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.C.VOLTage.DATA

Equivalent key

System > **Multiport Test Set setup** > **Test Set[1|2]** > **Control Line D** > **DC Source**

- This command is available only for E5092A multiport test set.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:OUTPut:D.VOLTage.[:DATA] <numeric>

:SENSe:MULTiplexer{[1]|2}:OUTPut:D.VOLTage.[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:OUTP:D.VOLT.DATA 0"
20 OUTPUT 717;":SENS:MULT1:OUTP:D.VOLT.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).OUTPut.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).OUTPut.DATA = *Value*

Value = SCPI.SENSe(*Ch*).MULTiplexer(*Id*).OUTPut.DATA

Description

This command sets/gets or returns the status of HIGH/LOW of all the control lines of the E5091A whose ID is 1 to 2 (*Id*) when measuring selected channel (*Ch*) in the measurement using the E5091A.

- To set the control lines, use the values obtained by converting 8-bit binary values expressed HIGH (1) / LOW (0) of individual lines to decimal values, assuming line 1 as LSB and line 8 as MSB.

Variable

Parameter	*Value*
Description	Sets/gets control line value
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim OData As Long
SCPI.SENSe.MULTiplexer1.OUTPut.DATA = "0
MultName = SCPI.SENSe.MULTiplexer1.OUTPut.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Control Lines**

- This command is available only for E5091A multiport test set.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:MULTiplexer{[1]|2}:OUTPut[:DATA] <numeric>
:SENSe{[1]-160}:MULTiplexer{[1]|2}:OUTPut[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:OUTP.DATA 5"
20 OUTPUT 717;":SENS1:MULT1:OUTP.DATA?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).PORT(*Pt*).CATalog

Object type

Property (**Read-only**)

Syntax

String = SCPI.SENSe.MULTiplexer(*Id*).PORT(*Pt*).CATalog

Description

This command returns a port assigned to Port 1 to 20 (*Pt*) of the E5091A whose ID is 1 to 2 (*Id*) when measuring selected channel (*Ch*) in the measurement using the E5091A.

Examples

```
Dim PortCat As String
PortCat = SCPI.SENSe.MULTiplexer1.PORT4.CATalog
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

SCPI.SENSe(Ch).MULTiplexer(Id).DISPlay.STATe

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).SELect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:MULTiplexer{[1]|2}:PORT{[1]|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
16|17|18|19|20}:CATalog?
```

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:PORT1:CAT?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).PORT(*Pt*).SELect

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe.MULTiplexer(*Id*).PORT(*Pt*).SELect = *String*

String = SCPI.SENSe.MULTiplexer(*Id*).PORT(*Pt*).SELect

Description

This command sets/gets a port name assigned to Port 1 to 20 (*Pt*) of the Multiport test set whose ID is 1 to 2 (*Id*) when measuring selected channel (*Ch*) in the measurement using the multiport test set.

- The range of this command for port 3 and port 4 is R1/R2/R3/R4 (without the + or - sign). However, the firmware displays the value with - or + sign, for example R1+ or R1-. For using this command, the + or - character is not required in the value and the command should be sent without any sign (+ or -).

Variable

When the 9-port option of the E5091A (E5091A-009) is selected

Parameter	*String*			
Description	Sets a port name assigned to Port 1 to 20			
Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	A or **T1**	**T1** or **T2**	**R1**,**R2**,**R3**	**R1**,**R2**,**R3**
Preset value	A	T1	R1	R1

When the 13-port configuration of the E5091A-016 (E5091_13) or E5092A (E5092_13) is selected

Parameter	*String*
Description	ON/OFF of the control of the E5091A\E5092A

Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	A,T1,T2,T3	**T1,T2,T3,T4**	**R1,R2,R3,R4**	**R1,R2,R3,R4**
Preset value	A	T1	R1	R1

When the 16-port configuration of the E5091A-016 (E5091_16) is selected

Parameter	*String*						
Description	ON/OFF of the control of the E5091A						
Data type	String type						
Range	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7
	A1(A),A2,A3, A4,T1,T2,T3	**B1(T4),B2,B3, B4,T1,T2,T3**	**R1,R2 ,R3,R 4**	**R1,R2 ,R3,R 4**	**X 1** and **X 2**	**Y1** and **Y2**	**Z1** and **Z2**
Preset value	A1	B1	R1	R1	X 1	Y1	Z1

When the 16-port configuration of the E5092A (E5092_16) is selected

Parameter	*String*						
Description	ON/OFF of the control of the E5092A						
Data type	String type						
Range	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7
	A1,A2,A3,A4	**B1,B2,B3,B4**	**R1,R2,R3,R4**	**R1,R2,R3,R4**	**X1** and **X2**	**Y1** and **Y2**	**Z1** and **Z2**
Preset value	A1	B1	R1	R1	X1	Y1	Z1

When the 22-port configuration of the E5092A (E5092_22) is selected

Parameter	*String*			
Description	ON/OFF of the control of the E5092A			
Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4
	A1,A2,A3,A4,A5,A6	**A7,A8,A9,A10,A11**	**B1,B2,B3,B4,B5,B6**	**B7,B8,B9,B10,B11**

Preset value	A1	A7	B1	B7

When the 28-port configuration of the E5092A (E5092_28) is selected

Parameter	*String*									
Description	ON/OFF of the control of the E5092A									
Data type	String type									
Range	Port 1	Port 2	Port 3	Port 4	Port 5	Port 6	Port 7	Port 8	Port 9	Port 10
	A,B, C,D	**A,B, C,D**	**A,B, C,D**	**A,B, C,D**	**A, B**	**A, B**	**A, B**	**A, B**	**A, B**	**A, B**
Preset value	A	A	A	A	A	A	A	A	A	A

When the 10-port full crossbar configuration of the E5092A (E5092_X10) is selected

Parameter	*String*			
Description	ON/OFF of the control of the E5092A			
Data type	String type			
Range	Port 1	Port 2	Port 3	Port 4

	1,3,5,7	2,4,6,8	2,4,6,10	1,3,5,9
Preset value	1	2	2	1

For information on the variable (*Ch*), (*Id*) and (*Pt*), see Ch, Id and Pt.

Examples

```
Dim PortSel As String
SCPI.SENSe.MULTiplexer1.PORT4.SELect = "R2"
PortSel = SCPI.SENSe.MULTiplexer1.PORT4.SELect
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

SCPI.SENSe(Ch).MULTiplexer(Id).DISPlay.STATe

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).CATalog

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Port 1**|**Port 2**|**Port 3**|**Port 4**|**Port 5**|**Port 6**|**Port 7**|**Port 8**|**Port 9**|**Port 10**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-
160}:MULTiplexer{[1]|2}:PORT{[1]|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
16|17|18|19|20}[:SELect] <string>
```

```
:SENSe{[1]-
160}:MULTiplexer{[1]|2}:PORT{[1]|2|3|4|5|6|7|8|9|10|11|12|13|14|15|
16|17|18|19|20}[:SELect]?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:PORT1 A"
20 OUTPUT 717;":SENS1:MULT1:PORT1?"
30 ENTER 717;A$
```

SCPI.SENSe(Ch).MULTiplexer(*Id*).STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(Ch).MULTiplexer(*Id*).STATe = *Status*

Status = SCPI.SENSe(Ch).MULTiplexer(*Id*).STATe

Description

This command turns ON/OFF or return the status of the control (switching the internal switch that connects between the ports and changing control line output) of the Multiport test set whose ID is *Id*.

Variable

Parameter	*Status*
Description	ON/OFF of the control of the Multiport test set
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the control of the multiport test set. • False or OFF: Turns OFF the control of the multiport test set.
Preset value	False or OFF

Examples

```
Dim Cont As Boolean
SCPI.SENSe.MULTiplexer1.STATe = True
Cont = SCPI.SENSe.MULTiplexer1.STATe
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).OUTPut.DATA

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).SELect

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.OUTPut.DATA

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Control**

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:STATe {ON|OFF|1|0}

:SENSe:MULTiplexer{[1]|2}:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:STAT ON"
20 OUTPUT 717;":SENS:MULT1:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.OUTPut.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.OUTPut.DATA = *Value*

Value = SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.OUTPut.DATA

Description

This command sets/gets the HIGH/LOW of all the control line of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

- To set the control lines, use values obtained by converting 8-bit binary values expressed by HIGH (1)/LOW (0) of individual lines to decimal values, assuming line 1 as LSB and line 8 as MSB.

Variable

Parameter	*Value*
Description	Setting value the control line
Data type	Long integer type (Long)
Range	0 to 255
Preset value	0
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim C_line As Long
SCPI.SENSe(1).MULTiplexer1.TSET9.OUTPut.DATA = 5
C_line = SCPI.SENSe(1).MULTiplexer1.TSET9.OUTPut.DATA
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

Equivalent key

System > Multiport Test Set Setup > Test Set 1|Test Set 2 > Control Lines > Line 1|...|Line 8

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:OUTPut[:DATA] <numeric>

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:OUTPut[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:TSET9:OUTP 5"
20 OUTPUT 717;":SENS1:MULT1:TSET9:OUTP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT1

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT1 = *Param*

Param = SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT1

Description

This command sets/gets a port assigned to Port 1 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

- If the port assigned to Port 2 is T1 and you select T1 as the port assigned to Port 1, the port assigned to Port 2 is changed to T2 automatically.

Variable

Parameter	*Param*
Description	The port assigned to Port 1
Data type	Character string type (String)
Range	Select from either of the following: • "A": Selects A as the port assigned to Port 1. • "T1": Selects T1 as the port assigned to Port 1.
Preset Value	A

Examples

```
Dim Port As String
SCPI.SENSe(1).MULTiplexer1.TSET9.PORT1 = "T1"
Port = SCPI.SENSe(1).MULTiplexer1.TSET9.PORT1
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

SCPI.SENSe(Ch).MULTiplexer(Id).DISPlay.STATe

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Port 1** > **A**|**T1**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT1 {A|T1}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT1?

Query response

{A|T1}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT1 T1"
20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT1?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT2

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT2 = *Param*

Param = SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT2

Description

This command sets/gets a port assigned to Port 2 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

- If the port assigned to Port 1 is T1 and you select T1 as the port assigned to Port 2, the port assigned to Port 1 is changed to A automatically.

Variable

Parameter	*Param*
Description	The port assigned to Port 2
Data type	Character string type (String)
Range	Select from either of the following: • "T1": Selects T1 as the port assigned to Port 2. • "T2": Selects T2 as the port assigned to Port 2.
Preset Value	T1

Examples

```
Dim Port As String
SCPI.SENSe(1).MULTiplexer(1).TSET9.PORT2 = "T2"
Port = SCPI.SENSe(1).MULTiplexer(1).TSET9.PORT2
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

E5071C

SCPI.SENSe(Ch).MULTiplexer(Id).DISPlay.STATe

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Port2** > **T1**|**T2**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT2 {T1|T2}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT2?

Query response

{A|T1}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT2 T2"
20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT2?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT3

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT3 = *Param*

Param = SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT3

Description

This command sets/gets a port assigned to Port 3 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

Variable

Parameter	*Param*
Description	The port assigned to Port 3
Data type	Character string type (String)
Range	Select from either of the following: • "R1": Selects R1+ as the port assigned to Port 3. • "R2": Selects R2+ as the port assigned to Port 3. • "R3": Selects R3+ as the port assigned to Port 3.
Preset Value	R1

Examples

```
Dim Port As String
SCPI.SENSe(1).MULTiplexer(1).TSET9.PORT3 = "R2"
Port = SCPI.SENSe(1).MULTiplexer(1).TSET9.PORT3
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

SCPI.SENSe(Ch).MULTiplexer(Id).DISPlay.STATe

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Port3** > **R1+**|**R2+**|**R3+**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT3 {R1|R2}R3}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT3?

Query response

{R1|R2}R3}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT3 R2"
20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT3?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT4

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT3 = *Param*

Param = SCPI.SENSe(*Ch*).MULTiplexer(*Id*).TSET9.PORT3

Description

This command sets/gets a port assigned to Port 4 of the E5091A whose ID is *Id* when measuring selected channel (*Ch*) in the measurement using the E5091A.

Variable

Parameter	*Param*
Description	The port assigned to Port 4
Data type	Character string type (String)
Range	Select from either of the following: • "R1": Selects R1- as the port assigned to Port 4. • "R2": Selects R2- as the port assigned to Port 4. • "R3": Selects R3- as the port assigned to Port 4.
Preset Value	R1

Examples

```
Dim Port As String
SCPI.SENSe(1).MULTiplexer(1).TSET9.PORT4 = "R2"
Port = SCPI.SENSe(1).MULTiplexer(1).TSET9.PORT4
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).STATe

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).DISPlay.STATe

SCPI.SENSe(Ch).MULTiplexer(Id).PORT(Pt).CATalog

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Port4** > **R1-**|**R2-**|**R3-**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT4 {R1|R2|R3}

:SENSe{[1]-160}:MULTiplexer{[1]|2}:TSET9:PORT4?

Query response

{R1|R2}R3}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:MULT1:TSET9:PORT4 R2"
20 OUTPUT 717;":SENS1:MULT1:TSET9:PORT4?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.CONTrol.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.CONTrol.STATe = *Status*

Status = SCPI.SENSe(*Ch*).OFFSet.LOCal.CONTrol.STATe

Description

This command turns ON/OFF or return the status of the external signal source control mode, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	On/off of the external signal source
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the external signal source control mode. • False or OFF: Turns off the external signal source control mode.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSe(1).OFFSet.LOCal.CONTrol.STATe = False
State = SCPI.SENSe(1).OFFSet.LOCal.CONTrol.STATe
```

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Control**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal:CONTrol[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:OFFSet:LOCal:CONTrol[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:CONT ON"
20 OUTPUT 717;":SENS1:OFFS:LOC:CONT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.DATA

Type of object

Property (**Read-only**)

Syntax

Data = SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.DATA

Description

This command acquires the external signal source frequency data, for the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Reads out the frequencies of all measurement points as an array.
Data type	Variant type (Variant)

Example of use

```
Dim LoData As Variant
LoData = SCPI.SENSe(1).OFFSet.LOCal.FREQuency.DATA
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:DATA?

Query response

{value 1},□E/span> ,{value NOP}<newline><^END>

NOTE Where, NOP is the number of points.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:OFFS:LOC:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.DIVisor

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.DIVisor = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.DIVisor

Description

This command sets/gets the external signal source frequency by using a divisor value for the basic frequency, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a divisor value for the basic frequency
Data type	Double precision floating point type (Double)
Range	1 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Divisor As Double
SCPI.SENSe(1).OFFSet.LOCal.FREQuency.DIVisor = 50
Divisor = SCPI.SENSe(1).OFFSet.LOCal.FREQuency.DIVisor
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Divisor**

- The basic frequency range is set by using SCPI.SENSe(Ch).FREQuency.STARt and SCPI.SENSe(Ch).FREQuency.STOP.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:DIVisor <numeric>

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:DIVisor?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:DIV 1E1"
20 OUTPUT 717;":SENS1:OFFS:LOC:DIV?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.MULTiplier

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.MULTiplier = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.MULTiplier

Description

This command sets/gets the external signal source frequency by using a multiplier value for the basic frequency, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a multiplier value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-100 to 100
Preset value	0
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Multiplier As Double
SCPI.SENSe(1).OFFSet.LOCal.FREQuency.MULTiplier = -10
Multiplier = SCPI.SENSe(1).OFFSet.LOCal.FREQuency.MULTiplier
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Multiplier**

- The basic frequency range is set by using “SCPI.SENSe(Ch).FREQuency.STARt”and “SCPI.SENSe(Ch).FREQuency.STOP”.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:MULTiplier <numeric>

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:MULTiplier?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:MULT -1E1"
20 OUTPUT 717;":SENS1:OFFS:LOC:MULT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.OFFSet

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.OFFSet = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.OFFSet

Description

This command sets/gets the external signal source frequency is set by using an offset value for the basic frequency, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of an offset value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Offset As Double
SCPI.SENSe(1).OFFSet.LOCal.FREQuency.OFFSet = -10
Offset = SCPI.SENSe(1).OFFSet.LOCal.FREQuency.OFFSet
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Offset**

- The basic frequency range is set by using “SCPI.SENSe(Ch).FREQuency.STARt” and “SCPI.SENSe(Ch).FREQuency.STOP”.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:OFFSet <numeric>

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:OFFSet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:OFFS 5E9"
20 OUTPUT 717;":SENS1:OFFS:LOC:OFFS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.STARt

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.STARt = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.STARt

Description

This command sets/gets a start value for the external signal source frequency setting, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a start value for the external signal source
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Start As Double
SCPI.SENSe(1).OFFSet.LOCal.FREQuency.STARt = 100E6
Start = SCPI.SENSe(1).OFFSet.LOCal.FREQuency.STARt
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Start**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:STARt <numeric>

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:STAR 5E9"
20 OUTPUT 717;":SENS1:OFFS:LOC:STAR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.STOP

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.STOP = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.FREQuency.STOP

Description

This command sets/gets a stop value for the external signal source frequency setting, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a stop value for the external signal source
Data type	Double precision floating point type (Double)
Range	0 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Stop As Double
SCPI.SENSe(1).OFFSet.LOCal.FREQuency.STOP = 100E6
Stop = SCPI.SENSe(1).OFFSet.LOCal.FREQuency.STOP
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.LOCal.FREQuency.STARt

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Stop**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:STOP <numeric>

:SENSe{[1]-160}:OFFSet:LOCal[:FREQuency]:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:STOP 5E9"
20 OUTPUT 717;":SENS1:OFFS:LOC:STOP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level value for the external signal source power setting, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a power level value for the external signal source
Data type	Double precision floating point type (Double)
Range	-150 to 30
Preset value	-10
Unit	dBm
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Power As Double
SCPI.SENSe(1).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude = -10
Power = SCPI.SENSe(1).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

SCPI.SENSe(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > Frequency Offset > External Source > Power

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel][:IMMediate] [:AMPLitude] <numeric>

:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel][:IMMediate] [:AMPLitude]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:POW -12.5"
20 OUTPUT 717;":SENS1:OFFS:LOC:POW?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Description

This command sets/gets a power slope value for the external signal source, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a power slope value for the external signal source
Data type	Double precision floating point type (Double)
Range	-2 to 2
Preset value	0
Unit	dB/GHz
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Slope As Double
SCPI.SENSe(1).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA = -1
Slope = SCPI.SENSe(1).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

SCPI.SENSe(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > Frequency Offset > External Source > Slope

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel]:SLOPe[:DATA]
<numeric>
:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel]:SLOPe[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP 1.0"
20 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe = *Status*

Status = SCPI.SENSe(*Ch*).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe

Description

This command turns ON/OFF or returns the power slope value setting mode for the external signal source, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Turn the power slope setting ON/OFF, for the external signal source.
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the power slope mode. • False or OFF: Turns off the power slope mode.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSe(1).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe = False
State = SCPI.SENSe(1).OFFSet.LOCal.POWer.LEVel.SLOPe.STATe
```

Related objects

SCPI.SENSe(Ch).OFFSet.LOCal.POWer.LEVel.IMMediate.AMPLitude

SCPI.SENSe(Ch).OFFSet.LOCal.POWer.LEVel.SLOPe.DATA

Equivalent key

Sweep Setup > **Frequency Offset** > **External Source** > **Slope**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel]:SLOPe:STATe {ON|OFF|1|0}

:SENSe{[1]-160}:OFFSet:LOCal:POWer[:LEVel]:SLOPe:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP:STAT ON"
20 OUTPUT 717;":SENS1:OFFS:LOC:POW:SLOP:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.LOCal.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.LOCal.STATe = *Status*

Status = SCPI.SENSe(*Ch*).OFFSet.LOCal.STATe

Description

This command turns ON/OFF or return the external signal source frequency setting mode regardless of status of the frequency offset, for the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	On/off of external signal source frequency setting
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the frequency setting. • False or OFF: Turns off the frequency setting.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSe(1).OFFSet.LOCal.STATe = False
State = SCPI.SENSe(1).OFFSet.LOCal.STATe
```

Equivalent key

Setup > **Frequency Offset** > **External Source** > **LO Frequency**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:LOCal:STATe {ON|OFF|1|0}

:SENSe{[1]-160}:OFFSet:LOCal:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:LOC:STAT ON"
20 OUTPUT 717;":SENS1:OFFS:LOC:STAT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.DATA

Type of object

Property (**Read-only**)

Syntax

Data = SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.DATA

Description

This command reads the frequency data for the selected port (*Pt*) when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Reads out the frequencies of all measurement points as an array.
Data type	Variant type (Variant)

Example of use

```
Dim Freqdata As Variant
Freqdata = SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.DATA
```

Related objects

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:DATA?

Query response

{value 1},...,{value NOP}<newline><^END>

Where, NOP is the number of points.

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SENS1:OFFS:PORT2:DATA?"
30 ENTER 717;A(*)
```

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.DIVisor

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.DIVisor = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.DIVisor

Description

This command sets/gets the selected port (*Pt*) frequency by using a divisor value for the basic frequency when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a frequency divisor value for the basic frequency
Data type	Double precision floating point type (Double)
Range	1 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Divisor As Double
SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.DIVisor = 50
Divisor = SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.DIVisor
```

Related objects

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **Port n** > **Divisor**

- The basic frequency range is set by using “SCPI.SENSe(Ch).FREQuency.STARt” and “SCPI.SENSe(Ch).FREQuency.STOP”.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:DIVisor <numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:DIVisor?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:DIV 1E1"
20 OUTPUT 717;":SENS1:OFFS:PORT2:DIV?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.MULTiplier

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.MULTiplier = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.MULTiplier

Description

This command sets/gets the selected port (*Pt*) frequencies by using a multiplier value for the basic frequency when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a frequency multiplier value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-100 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Multiplier As Double
SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.MULTiplier = -10
Multiplier = SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.MULTiplier
```

Related objects

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **Port n** > **Multiplier**

- The basic frequency range is set by using "SCPI.SENSe(Ch).FREQuency.STARt" and "SCPI.SENSe(Ch).FREQuency.STOP".

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency] :MULTiplier <numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency] :MULTiplier?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:MULT 1E1"
20 OUTPUT 717;":SENS1:OFFS:PORT2:MULT?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.OFFSet

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.OFFSet = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.OFFSet

Description

This command sets/gets the selected port (*Pt*) frequency by using an offset value for the basic frequency when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a frequency offset value for the basic frequency
Data type	Double precision floating point type (Double)
Range	-1E12 to 1E12
Preset value	0
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Offset As Double
SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.OFFSet = 1E9
Offset = SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.OFFSet
```

Related objects

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STARt

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Equivalent key

Sweep Setup > **Frequency Offset** > **Port n** > **Offset**

- The basic frequency range is set by using “SCPI.SENSe(Ch).FREQuency.STARt” and “SCPI.SENSe(Ch).FREQuency.STOP”.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:OFFSet <numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:OFFSet?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:OFFS 1E9"
20 OUTPUT 717;":SENS1:OFFS:PORT2:OFFS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.STARt

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.STARt = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.STARt

Description

This command sets/gets a frequency start value for the selected port (*Pt*) when the frequency offset feature is ON, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a frequency start value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	1E5
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Start As Double
SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.STARt = 100E6
Start = SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.STARt
```

Related objects

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STOP

Equivalent key

Sweep Setup > Frequency Offset > Port n > Start

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STARt <numeric>

:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STARt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:STAR 300000"
20 OUTPUT 717;":SENS1:OFFS:PORT2:STAR?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.STOP

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.STOP = *Value*

Value = SCPI.SENSe(*Ch*).OFFSet.PORT(Pt).FREQuency.STOP

Description

This command sets/gets the frequency stop value for the selected port (*Pt*) when the frequency offset feature is on, for the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Setting of a frequency stop value
Data type	Double precision floating point type (Double)
Range	See frequency range for each option
Preset value	Maximum frequency in the range
Unit	Hz (hertz)
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Example of use

```
Dim Stop As Double
SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.STOP = 100E6
Stop = SCPI.SENSe(1).OFFSet.PORT(1).FREQuency.STOP
```

Related objects

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.DIVisor

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.MULTiplier

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.OFFSet

SCPI.SENSe(Ch).OFFSet.PORT(Pt).FREQuency.STARt

Equivalent key

Sweep Setup > **Frequency Offset** > **Port n** > **Stop**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STOP
<numeric>
:SENSe{[1]-160}:OFFSet:PORT{[1]|2|3|4}[:FREQuency]:STOP?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:OFFS:PORT2:STOP 1e+009"
20 OUTPUT 717;":SENS1:OFFS:PORT2:STOP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).OFFSet.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).OFFSet.STATe = *Status*

Status = SCPI.SENSe(*Ch*).OFFSet.STATe

Description

This command turns ON/OFF or returns the status of the frequency offset, for the selected channel (*Ch*).

- When the frequency offset feature is ON, different frequencies can be used for measurement for each port. Frequencies set for each port are used regardless of whether the port is on the stimulus side or response side.

Variable

Parameter	*Status*
Description	On/off of the frequency offset
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the frequency offset. • False or OFF: Turns off the frequency offset.
Preset value	False or OFF

Example of use

```
Dim State As Boolean
SCPI.SENSe(1).OFFSet.STATe = False
State = SCPI.SENSe(1).OFFSet.STATe
```

Equivalent key

Sweep Setup > **Frequency Offset** > **Frequency Offset**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:OFFSet[:STATe] {ON|OFF|1|0}

:SENSe{[1]-160}:OFFSet[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:OFFS ON"
20 OUTPUT 717;":SENS1:OFFS?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).ROSCillator.SOURce

Object type

Property (**Read-only**)

Syntax

Param = SCPI.SENSe(*Ch*).ROSCillator.SOURce

Description

This command reads whether the external reference signal is inputted to the Ref In connector on the rear panel.

Variable

Parameter	*Param*
Description	Whether the external reference signal is inputted or not.
Data type	Character string type (String)
Range	Select from either of the following: • "INTernal": The external reference signal is not inputted. • "EXTernal": The external reference signal is inputted.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Ref As String
Ref = SCPI.SENSe(1).ROSCillator.SOURce
```

Equivalent key

Displayed on the instrument status bar (at the bottom of the LCD display).

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:ROSCillator:SOURce?

Query response

{INTernal|EXTernal}<newline><^END>

	Description

INTernal	The external reference signal is not inputted.
EXTernal	The external reference signal is inputted.

Example of use

```
10 OUTPUT 717;":SENS1:ROSC:SOUR?"
20 ENTER 717;A$
```

SCPI.SENSe(*Ch*).SEGMent.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SEGMent.DATA = *Data*

Data = SCPI.SENSe(*Ch*).SEGMent.DATA

Description

This command creates the segment sweep table of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data arranged in the following order (for the segment sweep table); where N is the number of segments (specified with <segm>) and n is an integer between 1 and N. *Data* = {<buf>,<stim>,<ifbw>,<pow>,<del>,<swp>,<time>,<segm>, <star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<swp 1>,<time 1>,... , <star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<swp n>,<time n>,.... , <star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<swp N>,<time N>} Each parameter in the above array data is detailed below: • <buf>: Always specify 5 or 6. You have to specify 6 if you need to set up the sweep mode setting for each segment. • <stim>: Stimulus setting mode 0: Specifies with start/stop values 1: Specifies with center/span values • <ifbw>: ON/OFF of the IF bandwidth setting for each segment 0: OFF, 1: ON • <pow>: ON/OFF of the power setting for each segment 0: OFF, 1: ON • <del>: ON/OFF of the sweep delay time setting for

each segment
0: OFF, 1: ON

- <swp>: ON/OFF of the sweep mode setting for each segment
0: OFF, 1: ON
Not necessary when <buf> is 5.
- <time>: ON/OFF of the sweep time setting for each segment
0: OFF, 1: ON
- <segm>: Number of segments
Specify an integer ranging 1 to 201.
- <star n>: Start value/center value of the n-th segment
- <stop n>: Stop value/span value of the n-th segment
- <nop n>: Number of measurement points of the n-th segment
- <ifbw n>: IF bandwidth of the n-th segment Not necessary when the IF bandwidth setting for each segment is OFF (<ifbw>:0).
- <pow n>: Power of the n-th segment Not necessary when the power setting for each segment is OFF (<pow>:0).
- <del n>: Sweep delay time of the n-th segment Not necessary when the sweep delay time setting for each segment is OFF (<del>:0).
- <swp n> : Sweep mode of the n-th segment
0: Stepped mode
1: Swept mode
2: Stepped mode
3: Swept mode
Not necessary when <buf> is 5 or the sweep mode setting for each segment is OFF (<del>:0). 2 or 3 can be acceptable for compatibility with E5071C.
- <time n>
  - Sweep time of the n-th segment is not necessary when the sweep time setting for each segment is OFF (<time>:0).

Data type	Variant type (Variant)
Note	If the necessary amount of array data for the specified number of segments is not available while setting the segment sweep table, an error occurs when executed and the object is ignored. For <stim>, <ifbw>, <pow>, <del>, <swp>, and <time>, if the specified value is not the allowable integer, an error occurs when executed. For <star n>, <stop n>, <nop n>, <ifbw n>, <pow n>, <del n>, and <time n> in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SegmData As Variant
SCPI.SENSe(1).SEGMent.DATA = Array(5,0,0,1,0,0,2, _
100E6,1E9,31,0,2E9,3E9,51,-10)
SegmData = SCPI.SENSe(1).SEGMent.DATA
```

```
Dim SegmData(14) As Variant
Dim Ref As Variant
SegmData(0) = 5
SegmData(1) = 0
SegmData(2) = 0
SegmData(3) = 1
SegmData(4) = 0
SegmData(5) = 0
SegmData(6) = 2
SegmData(7) = 100E6
SegmData(8) = 1E9
SegmData(9) = 31
SegmData(10) = 0
SegmData(11) = 2E9
SegmData(12) = 3E9
SegmData(13) = 51
SegmData(14) = -10
SCPI.SENSe(1).SEGMent.DATA = SegmData
Ref = SCPI.SENSe(1).SEGMent.DATA
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Edit Segment Table**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SEGMent:DATA
5,<mode>,<ifbw>,<pow>,<del>,<time>,<segm>,
<star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<time 1>,?E/span> ,
<star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<time n>,?E/span> ,
<star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<time N>

:SENSe{[1]-160}:SEGMent:DATA
6,<mode>,<ifbw>,<pow>,<del>,<swp>,<time>,<segm>,
<star 1>,<stop 1>,<nop 1>,<ifbw 1>,<pow 1>,<del 1>,<swp 1>,<time 1>,?E/span> ,
<star n>,<stop n>,<nop n>,<ifbw n>,<pow n>,<del n>,<swp n>,<time n>,?E/span> ,
<star N>,<stop N>,<nop N>,<ifbw N>,<pow N>,<del N>,<swp N>,<time N>

:SENSe{[1]-160}:SEGMent:DATA?

Where N is the number of segments (specified with <segm>) and n is an integer between 1 and N.

Query response

When the sweep mode setting for each segment is OFF:

5,{mode},{ifbw},{pow},{del},{time},{segm},
{star 1},{stop 1},{nop 1},{pow 1},{del 1},{time 1},?E/span> ,
{star n},{stop n},{nop n},{pow n},{del n},{time n},?E/span> ,
{star N},{stop N},{nop N},{pow N},{del N},{time N}<newline><^END>

When the sweep mode setting for each segment is ON:

6,{mode},{ifbw},{pow},{del},{swp},{time},{segm},
{star 1},{stop 1},{nop 1},{pow 1},{del 1},{swp 1},{time 1},?E/span> ,
{star n},{stop n},{nop n},{pow n},{del n},{swp n},{time n},?E/span> ,
{star N},{stop N},{nop N},{pow N},{del N},{swp N},{time N}<newline><^END>

Example of use

```
10 DIM H(1:3,1:4)
20 OUTPUT 717;":SENS1:SEGM:DATA 5,0,1,0,0,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,";
50 OUTPUT 717;"4E9,6E9,11,70e3"
60 OUTPUT 717;":SENS1:SEGM:DATA?"
70 ENTER 717;A,B,C,D,E,F,G,H(*)
```

```
10 DIM H(1:3,1:5)
20 OUTPUT 717;":SENS1:SEGM:DATA 6,0,1,0,0,1,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,2,";
50 OUTPUT 717;"4E9,6E9,11,70e3,3"
60 OUTPUT 717;":SENS1:SEGM:DATA?"
70 ENTER 717;A,B,C,D,E,F,G,H(*)
```

SCPI.SENSe(*Ch*).SEGMent.SWEep.POINts

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SENSe(*Ch*).SEGMent.SWEep.POINts

Description

This command reads the total number of the measurement points of all segments, for the segment sweep table of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Total number of measurement points of all segments
Data type	Long integer type (Long)
Preset Value	2

Examples

```
Dim SegmPoin As Long
SegmPoin = SCPI.SENSe(1).SEGMent.SWEep.POINts
```

Related objects

SCPI.SENSe(Ch).SEGMent.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SEGMent:SWEep:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:POIN?"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.DATA

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SENSe(*Ch*).SEGMent.SWEep.TIME.DATA

Description

This command reads the total sweep time (including sweep delay time) of all segments, for the segment sweep table of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Total sweep time of all segments
Data type	Double precision floating point type (Double)

Examples

```
Dim SegmTime As Double
SegmTime = SCPI.SENSe(1).SEGMent.SWEep.TIME.DATA
```

Related objects

SCPI.SENSe(Ch).SEGMent.DATA

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SEGMent:SWEep:TIME[:DATA]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SEGM:SWE:TIME?"
20 ENTER 717;A
```

SCPI.SENSe(*Ch*).SEGMent.LIST.CONTrol.DATA

Object type

Property (**Read-Write**)

Syntax

This option is only available in revision A.9.60 and above:

SCPI.SENSe(*Ch*).SEGMent.LIST.CONTrol.DATA = *Data*

Data = SCPI.SENSe(*Ch*).SEGMent.LIST.CONTrol.DATA

Description

This command sets/gets the state of each segment in the segment sweep table of selected channel (*Ch*).

This command is only available in revision A.9.60 and above.

Variable

Parameter	*Data*
Description	Indicates the state array data; where N is the number of segments and n is an integer between 1 and N. *Data* = {<state1>,... ,<state n>,.... ,<state N>}
Data type	Variant type (Variant)

Examples

```
Dim SegmContrData As Variant
SCPI.SENSe(1).SEGMent.LIST.CONTrol.DATA = Array(1, 1, 0, 0)
SegmContrData = SCPI.SENSe(1).SEGMent.LIST.CONTrol.DATA
SCPI.SENSe(1).SEGMent.LIST.CONTrol.DATA = SegmContrData
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

Equivalent key

Sweep Setup > **Edit Segment Table**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SEGMent:LIST:CONTrol:DATA <state 1>, <state n>, <state N>

:SENSe{[1]-160}:SEGMent:LIST:CONTrol:DATA?

Where N is the number of segments and n is an integer between 1 and N.

Example of use

```
10 DIM H(1:3)
20 OUTPUT 717;":SENS1:SEGM:DATA 5,0,1,0,0,0,3,";
30 OUTPUT 717;"1E9,3E9,11,70e3,";
40 OUTPUT 717;"3E9,4E9,51,7e3,";
50 OUTPUT 717;"4E9,6E9,11,70e3"
60 OUTPUT 717;":SENS1:SEGM:LIST:CONT:DATA 1, 0, 0"
70 OUTPUT 717;":SENS1:SEGM:LIST:CONT:DATA?"
80 ENTER 717;H(*)
```

SCPI.SENSe(Ch).SEGMent.LIST.CONTrol.STATe

Object type

Property (**Read-Write**)

Syntax

This option is only available in revision A.9.60 and above:

SCPI.SENSe(*Ch*).SEGMent.LIST.CONTrol.STATe = *Status*

Status= SCPI.SENSe(*Ch*).SEGMent.LIST.CONTrol.STATe

Description

This command turns ON or OFF the individual segment state control in the segment sweep table.

This command is only available in revision A.9.60 and above.

Variable

Parameter	*Status*
Description	ON/OFF of the individual segment state control
Data type	Boolean type (Boolean)
Range	Select from either of the following: • ON or 1: Turns ON the individual segment state cor • OFF or 0: Turns OFF the individual segment state control.
Preset value	OFF or 0

Examples

```
Dim SegmState As Boolean
SCPI.SENSe(1).SEGMent.LIST.CONTrol.STATe = True
SegmState= SCPI.SENSe(1).SEGMent.LIST.CONTrol.STATe
```

Related objects

SCPI.SENSe(Ch).SEGMent.LIST.CONTrol.DATA

Equivalent key

Sweep Setup > **Edit Segment Table** > **List Control**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SEGMent:LIST:CONTrol:STATe: {ON|OFF|1|0}
:SENSe{[1]-160}:SEGMent:LIST:CONTrol:STATe?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SEGMent:LIST:CONTrol:STATe ON"
20 OUTPUT 717;":SENS1:SEGMent:LIST:CONTrol:STATe?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.ASPurious

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.ASPurious = *Status*

Status = SCPI.SENSe(*Ch*).SWEep.ASPurious

Description

This command does not performs any operation. This command is provided because of its command compatibility with E5070B/E5071B.

Variable

Parameter	*Status*
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Select from the following. • True or ON: No meaning • False or OFF: No meaning
Preset value	True or OFF

Examples

```
Dim ASpurious As Boolean
SCPI.SENSe(1).SWEep.ASPurious = False
ASpurious = SCPI.SENSe(1).SWEep.ASPurious
```

Equivalent key

None

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SWEep:ASPurious {ON|OFF|1|0}

:SENSe{[1]-160}:SWEep:ASPurious?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:ASP OFF"
20 OUTPUT 717;":SENS1:SWE:ASP?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.DELay

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.DELay = *Value*

Value = SCPI.SENSe(*Ch*).SWEep.DELay

Description

This command sets/gets the sweep delay time of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Sweep delay time
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0
Unit	s (second)
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SweDel As Double
SCPI.SENSe(1).SWEep.DELay = 0.05
SweDel = SCPI.SENSe(1).SWEep.DELay
```

Equivalent key

Sweep Setup > **Sweep Delay**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SWEep:DELay <numeric>

:SENSe{[1]-160}:SWEep:DELay?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:DEL 0.05"
20 OUTPUT 717;":SENS1:SWE:DEL?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.GENeration

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.GENeration = *Param*

Param = SCPI.SENSe(*Ch*).SWEep.GENeration

Description

This command sets/gets the sweep mode of selected channel (*Ch*).

- When the sweep type is the power sweep (POW specified with the SCPI.SENSe(Ch).SWEep.TYPE object), when the power calibration is on (ON specified with the SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe object), or the power slope value is other than 0 and the power slope function is on (ON specified with the SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe object), if you execute this object to try to set the sweep mode to the swept mode, an error occurs and the sweep mode is automatically set to the step mode.

Variable

Parameter	*Param*
Description	Sweep mode
Data type	Character string type (String)
Range	Select from either of the following: • "STEPped": Sets the sweep mode to the stepped mode. • "ANALog": Sets the sweep mode to the swept mode. • "FSTepped": Sets the sweep mode to the stepped mode. This parameter is provided because of its command compatibility with E5070B/E5071B. • "FANalog": Sets the sweep mode to the swept mode. This parameter is provided because of its command compatibility with E5070B/E5071B.

Preset value	"STEPped"

Examples

```
Dim SwptMode As String
SCPI.SENSe(1).SWEep.GENeration = "anal"
SwptMode = SCPI.SENSe(1).SWEep.GENeration
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > **Sweep Mode** > **Stepped|Swept**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SWEep:GENeration
{STEPped|ANALog|FSTepped|FANalog}
:SENSe{[1]-160}:SWEep:GENeration?
```

Query response

```
{STEP|ANAL|FST|FAN}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SWE:GEN ANAL"
20 OUTPUT 717;":SENS1:SWE:GEN?"
30 ENTER 717;A$
```

E5071C

SCPI.SENSe(*Ch*).SWEep.POINts

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.POINts = *Value*

Value = SCPI.SENSe(*Ch*).SWEep.POINts

Description

This command sets/gets the number of measurement points of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Number of measurement points
Data type	Long integer type (Long)
Range	2 to 20001
Preset value	201
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Nop As Long
SCPI.SENSe(1).SWEep.POINts = 801
Nop = SCPI.SENSe(1).SWEep.POINts
```

Equivalent key

Sweep Setup > **Points**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SWEep:POINts <numeric>

:SENSe{[1]-160}:SWEep:POINts?

Query response

2512

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:POIN 801"
20 OUTPUT 717;":SENS1:SWE:POIN?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.TIME.AUTO

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.TIME.AUTO = *Status*

Status = SCPI.SENSe(*Ch*).SWEep.TIME.AUTO

Description

This command sets/gets whether to automatically set the sweep time of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of the auto setting of the sweep time
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the auto setting. • False or OFF: Turns OFF the auto setting.
Preset value	True or OFF

Examples

```
Dim SweAuto As Boolean
SCPI.SENSe(1).SWEep.TIME.AUTO = False
SweAuto = SCPI.SENSe(1).SWEep.TIME.AUTO
```

Related objects

SCPI.SENSe(Ch).SWEep.TIME.DATA

Equivalent key

Sweep Setup > **Sweep Time**

- When performing the operation from the front panel, the auto setting of the sweep time is turned ON by setting the sweep time to 0 s.

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SWEep:TIME:AUTO {ON|OFF|1|0}

:SENSe{[1]-160}:SWEep:TIME:AUTO?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TIME:AUTO ON"
20 OUTPUT 717;":SENS1:SWE:TIME:AUTO?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.TIME.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.TIME.DATA = *Value*

Value = SCPI.SENSe(*Ch*).SWEep.TIME.DATA

Description

This command sets/gets the sweep time of selected channel (*Ch*).

- Before using this object to set the sweep time, turns OFF the auto setting of the sweep time (specify False with the SCPI.SENSe(Ch).SWEep.TIME.AUTO object).

Variable

Parameter	*Value*
Description	Sweep time
Data type	Double precision floating point type (Double)
Range	Varies depending on the measurement conditions
Preset value	Varies depending on the measurement conditions
Unit	s (second)
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SweTime As Double
SCPI.SENSe(1).SWEep.TIME.AUTO = False
SCPI.SENSe(1).SWEep.TIME.DATA = 1.5
SweTime = SCPI.SENSe(1).SWEep.TIME.DATA
```

Related objects

SCPI.SENSe(Ch).SWEep.TIME.AUTO

Equivalent key

Sweep Setup > Sweep Time

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:SWEep:TIME[:DATA] <numeric>
:SENSe{[1]-160}:SWEep:TIME[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TIME 1.5"
20 OUTPUT 717;":SENS1:SWE:TIME?"
30 ENTER 717;A
```

SCPI.SENSe(*Ch*).SWEep.TYPE

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).SWEep.TYPE = *Param*

Param = SCPI.SENSe(*Ch*).SWEep.TYPE

Description

This command sets/gets the sweep type of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	Sweep type
Data type	Character string type (String)
Range	Select from either of the following: • "LINear": Sets the sweep type to the linear sweep. • "LOGarithmic": Sets the sweep type to the log sweep. • "SEGMent": Sets the sweep type to the segment sweep. • "POWer": Sets the sweep type to the power sweep.
Preset value	"LINear"

Examples

```
Dim SweType As String
SCPI.SENSe(1).SWEep.TYPE = "segm"
SweType = SCPI.SENSe(1).SWEep.TYPE
```

Equivalent key

Sweep Setup > **Sweep Type** > **Lin Freq**|**Log Freq**|**Segment**

Equivalent SCPI command

Syntax

:SENSe{[1]-160}:SWEep:TYPE {LINear|LOGarithmic|SEGMent|POWer}

:SENSe{[1]-160}:SWEep:TYPE?

Query response

{LIN|LOG|SEGM|POW}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS1:SWE:TYPE SEGM"
20 OUTPUT 717;":SENS1:SWE:TYPE?"
30 ENTER 717;A$
```

SCPI.SENSe(*Ch*).VOLTage(*Auxpt*).DC.RANGe.UPPer

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe(*Ch*).VOLTage(Auxpt).DC.RANGe.UPPer=Param

Param=SCPI.SENSe(*Ch*).VOLTage(Auxpt).DC.RANGe.UPPer

Description

This command sets/gets the DC voltage range of selected channel (*Ch*).

Variable

Parameter	*Param*
Description	AUX DC Input range
Data type	Double precision floating point type (Double)
Unit	Volts (V)
Range	1 or 10
Preset value	10

Parameter	*Auxpt*
Description	AUX port
Data type	Long Integer Type (Long)
Range	1 or 2
Preset value	None

Examples

```
Dim DcRange As variant
SCPI.SENSe(1).VOLTage(2).DC.RANGe.UPPer=1
DcRange = SCPI.SENSe(1).VOLTage(1).DC.RANGe.UPPer
```

Equivalent key

Meas > **Aux Input** (1 or 2) > **Range**

Equivalent SCPI command

Syntax

```
:SENSe{[1]-160}:VOLTage{[1]|2}:DC.RANGe:UPPer
:SENSe{[1]-160}:VOLTage{[1]|2}:DC.RANGe:UPPer?
```

Query response

```
{1|10}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SENS1:VOLT1:DC.RANG:UPP 10"
20 OUTPUT 717;":SENS1:VOLT1:DC.RANG:UPP?"
30 ENTER 717;A
```

SCPI.SENSe.MULTiplexer(*Id*).INCount

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SENSe.MULTiplexer(*Id*).INCount

Description

This command reads the number of input ports of ID 1 to 2 (*Id*), from the E5091A.

Examples

```
Dim INCount As Long
INCount = SCPI.SENSe(1).MULTiplexer(1).INCount
```

Related objects

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT1

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT2

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT3

SCPI.SENSe(Ch).MULTiplexer(Id).TSET9.PORT4

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:INCount?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:INC?"
20 ENTER 717;A
```

SCPI.SENSe.MULTiplexer(*Id*).NAME

Object type

Property (**Read-Write**)

Syntax

SCPI.SENSe.MULTiplexer(*Id*).NAME = *Param*

Param = SCPI.SENSe.MULTiplexer(*Id*).NAME

Description

This command sets/gets the configuration of the multiport test set.

- The upper case and lower case are not distinguished.

Variable

Parameter	*Param*
Description	The name of the multiport test set
Data type	Character string type (String)
Range	Select from either of the following: • "E5091_9": Selects the 9-port option of the E5091A (E5091A-009). • "E5091_13": Selects the 13-port configuration of the E5091A-016. • "E5091_16": Selects the 16-port configuration of the E5091A-016. • "E5092_13": Selects the 13-port configuration of the E5092A (E5092A-020). Equivalent to E5091_13 of the E5091A-016. • "E5092_16": Selects the 16-port configuration of the E5092A (E5092A-020). Equivalent to E5091_16 of the E5091A-016. • "E5092_22":Selects the 22-port configuration of the E5092A (E5092A-020). • "E5092_28":Selects the switching independently in the E5092A (E5092A-020). • "E5092_X10":Selects the 10-port full crossbar configuration of the E5092A (E5092A-020).
Preset	E5091_9

Value	

For information on the variable (*Id*), see Id.

Examples

```
Dim MultName As String
SCPI.SENSe.MULTiplexer(1).NAME = "E5091_16"
MultName = SCPI.SENSe.MULTiplexer(1).NAME
```

Related objects

SCPI.SENSe.MULTiplexer.CATalog

SCPI.SENSe.MULTiplexer(Id).INCount

Equivalent key

System > **Multiport Test Set Setup** > **Test Set 1**|**Test Set 2** > **Select Test Set** > **E5091_9**|**E5091_13**|**E5091_16**|**E5092_13**|**E5092_16**|**E5092_22**|**E5092_28**|**E5092_X10**

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:NAME {E5091_9|E5091_13|E5091_16|E5092_13|E5092_16|E5092_22|E5092_28|E5092_X10}

:SENSe:MULTiplexer{[1]|2}:NAME?

Query response

{E5091_9|E5091_13|E5091_16|E5092_13|E5092_16|E5092_22|E5092_28|E5092_X10}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT1:NAME E5091_9"
20 OUTPUT 717;":SENS:MULT1:NAME?"
30 ENTER 717;A$
```

SCPI.SENSe.MULTiplexer.CATalog

Object type

Property (**Read-only**)

Syntax

Param = SCPI.SENSe.MULTiplexer.CATalog

Description

This command reads the name of the multiport test set.

Examples

```
Dim MultCat As String
MultCat = SCPI.SENSe.MULTiplexer.CATalog
```

Related objects

SCPI.SENSe.MULTiplexer(Id).NAME

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SENSe:MULTiplexer{[1]|2}:CATalog?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SENS:MULT:CAT?"
20 ENTER 717;A$
```

SERVICE

SCPI.SERVice.ADJust.SENSe(Ch).SWEep.RF.RANGe.PORT(Pt).AUTO

Object type

Property (**Read-Write**)

Syntax

Value = SCPI.SERVice.ADJust.SENSe(Ch).SWEep.RF.RANGe.PORT(Pt).AUTO

SCPI.SERVice.ADJust.SENSe(Ch).SWEep.RF.RANGe.PORT(Pt).AUTO = *Value*

Description

This command sets/gets Receiver RF range mode ON/OFF status. This command is used in the 8 term Calibration using External PC.

Variable

Parameter	*Value*
Description	ON/OFF of Receiver RF Range Mode
Data type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	ON
Note	The setting for this command is set at ON by the preset, but it is not saved in the state file. Even if save command is executed after calibration with RF range Auto mode as OFF, it is not possible to measure the value correctly only by recalling. To recall the state, the status of RF range Auto should be set as OFF.

Examples

```
Dim Val As Boolean
SCPI.SERVice.ADJust.SENSe(1).SWEep.RF.RANGe.PORT(2).AUTO = True
Val = SCPI.SERVice.ADJust.SENSe(1).SWEep.RF.RANGe.PORT(2).AUTO
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:ADJust:SENSe([1]-160):SWEep:RF:RANGe:PORT([1]-4):AUTO {ON|OFF|1|0}

:SERVice:ADJust:SENSe([1]-160):SWEep:RF:RANGe:PORT([1]-4):AUTO?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:ADJ:SENS1:SWE:RF:RANG:PORT2:AUTO ON"
20 OUTPUT 717;":SERV:ADJ:SENS2:SWE:RF:RANG:PORT3:AUTO?"
30 ENTER 717;A
```

SCPI.SERVice.ADJust.VIRTual.BRIDge.STATe

Object type

Property (**Read-Write**)

Syntax

Value = SCPI.SERVice.ADJust.VIRTual.BRIDge.STATe

SCPI.SERVice.ADJust.VIRTual.BRIDge.STATe = *Value*

Description

This command sets/gets virtual bridge correction ON/OFF status. This command is used in the 8 term Calibration using External PC.

- This command does not works in options 2D5, 4D5, 2K5 and 4K5.

Variable

Parameter	*Value*
Description	ON/OFF of Virtual bridge correction
Data type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	ON
Note	The setting for this command is set at ON by the preset, but it is not saved in the state file. Even if save command is executed after calibration with virtual bridge OFF, it is not possible to measure the value correctly only by recalling. To recall the state the status of virtual bridge should be set as OFF.

Examples

```
Dim Val As Boolean
SCPI.SERVice.ADJust.VIRTual.BRIDge.STATe = True
Val = SCPI.SERVice.ADJust.SERVice.VIRTual.BRIDge.STATe
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:ADJust:VIRTual:BRIDge[:STATe]{ON|OFF|1|0}

:SERVice:ADJust:VIRTual:BRIDge[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:ADJ:VIRT:BRID ON"
20 OUTPUT 717;":SERV:ADJ:VIRT:BRID?"
30 ENTER 717;A
```

SCPI.SERVice.CHANnel(*Ch*).TRACe.ACTive

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.CHANnel(*Ch*).TRACe.ACTive

Description

This command reads the active trace number of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Active trace number
Data type	Long integer type (Long)

Examples

```
Dim ActTrac As Long
ActTrac = SCPI.SERVice.CHANnel(1).TRACe.ACTive
```

Related objects

SCPI.CALCulate(Ch).PARameter(Tr).SELect

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:CHANnel{[1]-36}:TRACe:ACTive?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN1:TRAC:ACT?"
20 ENTER 717;A
```

SCPI.SERVice.CHANnel.ACTive

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.CHANnel.ACTive

Description

This command reads the active channel number.

Variable

Parameter	*Value*
Description	Active channel number
Data type	Long integer type (Long)

Examples

```
Dim ActChan As Long
ActChan = SCPI.SERVice.CHANnel.ACTive
```

Related objects

SCPI.DISPlay.WINDow(Ch).ACTivate

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:CHANnel:ACTive?

Query response

{numeric}<newline><^END>

SCPI.SERVice.CHANnel.COUNt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.CHANnel.COUNt

Description

This reads the upper limit of the number of channels of the E5071C.

Variable

Parameter	*Value*
Description	Upper limit of the number of channels.
Data type	Long integer type (Long)

Examples

```
Dim MaxChan As Long
MaxChan = SCPI.SERVice.CHANnel.COUNt
```

Equivalent key

System > **Misc Setup** > **Channel/Trace Setup**

Equivalent SCPI command

Syntax

:SERVice:CHANnel:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:CHAN:COUN?"
20 ENTER 717;A
```

SCPI.SERVice.CHANnel.TRACe.COUNt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.CHANnel.TRACe.COUNt

Description

This command reads the upper limit of the number of traces per channel.

Variable

Parameter	*Value*
Description	Upper limit of the number of traces.
Data type	Long integer type (Long)

Examples

```
Dim MaxTrac As Long
MaxTrac = SCPI.SERVice.CHANnel.TRACe.COUNt
```

Equivalent key

System > **Misc Setup** > **Channel/Trace Setup**

Equivalent SCPI command

Syntax

```
:SERVice:CHANnel:TRACe:COUNt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SERV:CHAN:TRAC:COUN?"
20 ENTER 717;A
```

SCPI.SERVice.LOGGing.CLEar

Object type

Method (**Write-only**)

Syntax

SCPI.SERVice.LOGGIng.CLEar

Description

This command clears the log data of the E5071C. For more information about the log file, refer to Removing Log Data.

Examples

```
Dim ClearLog As Long
ClearLog = SCPI.SERVice.LOGGing.CLEar
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:LOGGing:CLEar

Example of use

```
10 OUTPUT 717;":SERV:LOGG:CLE"
```

SCPI.SERVice.PORT.COUNt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.PORT.COUNt

Description

This command reads the number of ports of the E5071C.

Variable

Parameter	*Value*
Description	Number of ports
Data type	Long integer type (Long)

Examples

```
Dim MaxPort As Long
MaxPort = SCPI.SERVice.PORT.COUNt
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:PORT:COUNt?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:PORT:COUN?"
20 ENTER 717;A
```

SCPI.SERVice.POWer.AC.OVERload.DETect.STATe

Object type

Method (**Read / Write**)

Syntax

Value = SCPI.SERVice.POWer.AC.OVERload.DETect.STATe

SCPI.SERVice.POWer:AC.OVERload.DETect.STATe= Value

Description

This command activates/deactivates the Overload Detection/Power Trip function. For more information, refer to Overload Detection and Power Trip Function.

- This function is valid only for E5071C and is not valid for E5070B/E5071B.

Variable

Parameter	*Status*
Description	Turns ON/OFF Overload detection/Power Trip function
Data type	Boolean type (Boolean)
Range	Select from the following: • True or ON: Turns ON the Overload detect • False or OFF: Turns OFF the Overload detect
Preset value	True or ON

Examples

```
Dim Value As Bool
Value=1
SCPI.SERVice.POWer.AC.OVERload.DETect.STATe=Value
Value=SCPI.SERVice.POWer.AC.OVERload.DETect.STATe
```

Equivalent key

System > **Service Menu** > **Overload Detect**

Equivalent SCPI command

Syntax

:SERVice:POWer:AC:OVERload:DETect:STATe {ON|OFF}

:SERVice:POWer:AC:OVERload:DETect:STATe?

Query response

{1|0}<newline><^END>

SCPI.SERVice.SREVision

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.SREVision

Description

This command reads the system spec version of the E5071C.

Variable

Parameter	*Value*
Description	1 means applying new system specifications. 0 means applying old system specifications.
Data type	Long integer type (Long)

Examples

```
Dim SystemRevision As Long
SystemRevision = SCPI.SERVice.PORT.COUNt
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SREVision?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SREV?"
20 ENTER 717;A
```

SCPI.SERVice.SWEep.FREQency.MAXimum

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.SWEep.FREQency.MAXimum

Description

This command reads the upper limit of measurement frequency of E5071C.

Variable

Parameter	*Value*
Description	Upper limit of measurement frequency
Data type	Double precision floating point type (Double)

Examples

```
Dim MaxFreq As Double
MaxFreq = SCPI.SERVice.SWEep.FREQency.MAXimum
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SWEep:FREQency:MAXimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:FREQ:MAX?"
20 ENTER 717;A
```

SCPI.SERVice.SWEep.FREQency.MINimum

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.SWEep.FREQency.MINimum

Description

This command reads the lower limit of measurement frequency of E5071C.

Variable

Parameter	*Value*
Description	Lower limit of measurement frequency
Data type	Double precision floating point type (Double)

Examples

```
Dim MinFreq As Double
MinFreq = SCPI.SERVice.SWEep.FREQency.MINimum
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SWEep:FREQency:MINimum?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:FREQ:MIN?"
20 ENTER 717;A
```

SCPI.SERVice.SWEep.POINts

Object type

Property (**Read-only**)

Syntax

Value = SCPI.SERVice.SWEep.POINts

Description

This command reads the upper limit of the number of measurement points at the current configuration.

Variable

Parameter	*Value*
Description	Upper limit of the number of points
Data type	Long integer type (Long)

Examples

```
Dim MaxPoin As Long
MaxPoin = SCPI.SERVice.SWEep.POINts
```

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SERVice:SWEep:POINts?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SERV:SWE:POIN?"
20 ENTER 717;A
```

SOURCE

SCPI.SOURce(*Ch*).POWer.ATTenuation.AUTO

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.ATTenuation.AUTO = *Status*

Status = SSCPI.SOURce(*Ch*).POWer.ATTenuation.AUTO

Description

This command turns on/off the Auto Power Range set function for the channels (Ch).

When the Auto Power Range set function is ON, the minimum value of step attenuation is selected according to the available source power range as described below:

Source Power Range	Step Attenuation
-25 ~ +10 dBm	0 dB
-35 ~ -25 dBm	10 dB
-45 ~ -35 dBm	20 dB
-55 ~ -45 dBm	30 dB
-65 ~ -55 dBm	40 dB
-75 ~ -65 dBm	50 dB
-85 ~ -75 dBm	60 dB

The power sweep setting is set according to the start power level and maximum value of stop power level is set according to step attenuation. For more information on power attenuation, refer to Setting Power Level with Auto Power Range.

- This command works only with the E5071C options 2D5, 4D5, 2K5 and 4K5 having a mechanical source attenuator. This command is neither required nor available in other E5071C options which are having electrical source attenuator and is provided to maintain backward compatibility.

Variable

Parameter	*Status*
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turn on the Auto Power Range • False or OFF: Turn off the Auto Power Range
Preset value	True or ON

Related objects

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

Equivalent key

Sweep Setup > **Power** > **Auto Range**

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:ATTenuation:AUTO {ON|OFF|1|0}
:SOURce{[1]-160}:POWer:ATTenuation:AUTO?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:ATT:AUTO ON"
20 OUTPUT 717;":SOUR1:POW:ATT:AUTO?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA = *Value*

Value = SCPI.SOURce(*Ch*).POWer.ATTenuation.DATA

Description

This command sets/gets the power range for the channel (Ch).

The power range is selected depending on the setting of the attenuator. The relationship between the attenuator value and the power range is available below:

Source Power Range	Step Attenuation
-25 ~ +10 dBm	0 dB
-35 ~ 0 dBm	10 dB
-45 ~ -10 dBm	20 dB
-55 ~ -20 dBm	30 dB
-65 ~ -30 dBm	40 dB
-75 ~ -40 dBm	50 dB
-85 ~ -50 dBm	60 dB

When Auto Power Range set function is effective, this command is ignored and an attenuator level and a power range are selected automatically according to a maximum frequency and a maximum setting power. For more information on power attenuation, refer to Setting Power Level with Auto Power Range.

- This command works with E5070B/E5071B and ENA options 2D5, 4D5, 2K5 and 4K5 .

Variable

Parameter	*Value*
Description	Dummy Parameter

Data type	Long integer type (Long)
Range	0 to 60
Preset value	0
Unit	dB
Resolution	10

Examples

```
Dim Att As Long
SCPI.SOURce(1).POWer.ATTenuation.DATA = 10
Att = SCPI.SOURce(1).POWer.ATTenuation.DATA
```

Related objects

SCPI.SOURce(Ch).POWer.ATTenuation.AUTO

Equivalent key

Sweep Setup > **Power** > **Power Ranges**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:ATTenuation[:DATA] <numeric>
:SOURce{[1]-160}:POWer:ATTenuation[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:ATT.DATA 20"
20 OUTPUT 717;":SOUR1:POW:ATT.DATA?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.CENTer

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.CENTer = *Value*

Value = SCPI.SOURce(*Ch*).POWer.CENTer

Description

This command sets/gets the center value of the sweep range for the power sweep, for channels 1 to 36 (*Ch*).

Variable

Parameter	*Value*
Description	Center value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz –55 to 10 dBm 5 to 6 GHz –55 to 9 dBm 6 to 7 GHz –55 to 8 dBm 7 to 8.5 GHz –55 to 7 dBm
Preset value	-7.5
Unit	dBm
Resolution	0.05 or 0.025
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pcntr As Double
SCPI.SOURce(1).POWer.CENTer = 0
Pcntr = SCPI.SOURce(1).POWer.CENTer
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.SPAN

Equivalent key

Center

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:CENTer <numeric>

:SOURce{[1]-160}:POWer:CENTer?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:CENT 0"
20 OUTPUT 717;":SOUR1:POW:CENT?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude = *Value*

Value = SCPI.SOURce(*Ch*).POWer.LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Power level
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range.
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PowLev As Double
SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude = -10
PowLev = SCPI.SOURce(1).POWer.LEVel.IMMediate.AMPLitude
```

Related objects

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.ATTenuation.AUTO

Equivalent key

Sweep Setup > **Power**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer[:LEVel][:IMMediate]
[:AMPLitude] <numeric>
:SOURce{[1]-160}:POWer[:LEVel][:IMMediate]
[:AMPLitude]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW -12.5"
20 OUTPUT 717;":SOUR1:POW?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA = *Value*

Value = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.DATA

Description

This command sets/gets the correction value of the power slope feature of channels 1 to 36 (*Ch*).

Variable

Parameter	*Value*
Description	Correction value of the power slope feature
Data type	Double precision floating point type (Double)
Range	-2 to 2
Preset value	0
Unit	dB/GHz
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim SlopLev As Double
SCPI.SOURce(1).POWer.LEVel.SLOPe.DATA = 0.1
SlopLev = SCPI.SOURce(1).POWer.LEVel.SLOPe.DATA
```

Related objects

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.STATe

Equivalent key

Sweep Setup > **Power** > **Slop [xxx dB/GHz]**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer[:LEVel]:SLOPe[:DATA] <numeric>
:SOURce{[1]-160}:POWer[:LEVel]:SLOPe[:DATA]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SLOP 0.1"
20 OUTPUT 717;":SOUR1:POW:SLOP?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe = *Status*

Status = SCPI.SOURce(*Ch*).POWer.LEVel.SLOPe.STATe

Description

This command turns ON/OFF or gets the status of the power slope feature, for the selected channel (*Ch*). This command corrects the attenuation of simple power level proportional to the frequency (attenuation due to cables etc).

Variable

Parameter	*Status*
Description	On/off of the power slope feature
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the power slop feature. • False or OFF: Turns off the power slop feature.
Preset value	False or OFF

Examples

```
Dim Slop As Boolean
SCPI.SOURce(1).POWer.LEVel.SLOPe.STATe = True
Slop = SCPI.SOURce(1).POWer.LEVel.SLOPe.STATe
```

Related objects

SCPI.SOURce(Ch).POWer.LEVel.SLOPe.DATA

SCPI.SENSe(Ch).SWEep.GENeration

Equivalent key

Sweep Setup > **Power** > **Slop [ON/OFF]**

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer[:LEVel]:SLOPe:STATe {ON|OFF|1|0}

:SOURce{[1]-160}:POWer[:LEVel]:SLOPe:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SLOP:STAT ON"
20 OUTPUT 717;":SOUR1:POW:SLOP:STAT?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.ACQuire

Object type

Property (**Write-only**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.ACQuire = *Param*

Description

This command sets the power calibration data using the specified power sensor, for the selected port (*Pt*) of the selected channel (*Ch*). When the measurement is complete successfully, the power level error correction is automatically turned ON.

- If the power meter is not connected correctly, an error occurs and this command is ignored.

Variable

Parameter	*Param*
Description	Selection of the power sensor
Data type	Character string type (String)
Range	Select from either of the following: • "ASENsor": Specifies power sensor A. • "BSENsor": Specifies power sensor B.

Examples

```
Dim Dmy As Long
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.ACQuire = "asen"
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.IEEE4882.OPC

Equivalent key

Cal > **Power Calibration** > **Take Cal Sweep**

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect[:ACQuire] {ASENsor|BSENsor}

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL ASEN"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.AVERage.COUNt

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.AVERage.COUNt = *Value*

Value = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.AVERage.COUNt

Description

This command sets/gets the number of power calibration data measurements per measurement point (averaging factor), for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Averaging factor
Data type	Long integer type (Long)
Range	1 to 100
Preset value	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim AvgCnt As Long
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.AVERage.COUNt = 6
AvgCnt = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.AVERage.COUNt
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLect.ACQuire

Equivalent key

Cal > **Power Calibration** > **Num of Readings**

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:AVERage[:COUNt] <numeric>

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:AVERage[:COUNt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:AVER 6"
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:AVER?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.TABLe.LOSS.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.TABLe.LOSS.DATA = *Data*

Data = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.TABLe.LOSS.DATA

Description

This command sets/gets the loss compensation table, for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data (for the loss compensation table) of 1 + Num (number of set data items)X2. Where n is an integer between 1 and Num. • *Data(0):* The number of data items you want to set. Specify an integer between 0 to 100. When you set the number of data items to 0 (to clear the loss compensation table), you specify only *Data(0)* as the Data variable. • *Data(nX2-1):* The frequency of the n-th data item (1 kHz to 500 GHz). • *Data(nX2):* The loss of the n-th data item (-100 dB to 100 dB). The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If the array data does not contain 1+Num (number of set data items)X2 when setting a loss compensation table, a runtime error occurs. For *Data(nX2-1)* and *Data(n X2)* in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Loss As Variant
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.DATA = Array(2,1e8,0.5
,1e9,0.8)
Loss = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.DATA
'"Clear Loss Table
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.DATA = Array(0)
```

```
Dim Loss(4) As Variant
Dim Ref As Variant
Loss(0) = 2
Loss(1) = 1e8
Loss(2) = 0.5
Loss(3) = 1e9
Loss(4) = 0.8
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.DATA = Loss
Ref = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.DATA
'"Clear Loss Table
Dim Loss(0) As Variant
Loss(0) = 0
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.DATA = Loss
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLect.TABLe.LOSS.STATe

Equivalent key

Cal > **Power Calibration** > **Loss Compen** > **Delete | Add | Clear Loss Table**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:TABLe:LOSS:DATA <numeric 1>,...,<numeric 1+(Nx2)>
```

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:TABLe:LOSS:DATA?
```

Query response

```
{numeric 1},...,{numeric 1+(Nx2)}<newline><^END>
```

Example of use

```
10 DIM B(1:2,1:2)
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA 2,";
30 OUTPUT 717;"1E8,0.5,1E9,0.8"
40 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA?"
50 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:LOSS:DATA 0" ! Clear Table
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.TABLe.LOSS.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.TABLe.LOSS.STATe = *Status*

Status = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.COLLect.TABLe.LOSS.STATe

Description

This command turns ON/OFF or returns the status of the loss compensation, for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Status*
Description	ON/OFF of loss compensation
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the loss compensation. • False or OFF: Turns off the loss compensation.
Preset value	False or OFF

Examples

```
Dim LComp As Boolean
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.STATe = True
LComp = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.TABLe.LOSS.STATe
```

Equivalent key

Cal > **Power Calibration** > **Loss Compen** > **Compensation**

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect:TABLe:LOSS[:STATe] {ON|OFF|1|0}

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:COLLect
:TABLe:LOSS[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:TABL:LOSS ON"
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:TABL:LOSS?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.DATA = *Data*

Data = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.DATA

Description

This command sets/gets the power calibration data array, for the selected port (*Pt*) of selected channel (*Ch*).

Variable

Parameter	*Data*
Description	Indicates the array data (power calibration data array) of NOP (number of points). Where n is an integer between 1 and NOP. *Data(n-1)* Data at the n-th measurement point The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If the array data does not contain NOP (number of measurement point) when setting a power calibration data array, a runtime error occurs.

Examples

```
Dim FreqData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
FreqData = SCPI.SENSe(1).FREQuency.DATA
```

```
Dim CorData As Variant
SCPI.SENSe(1).SWEep.POINts = 201
CorData = SCPI.SOURce(1).POWer.PORT(1).CORRection.DATA
SCPI.SOURce(1).POWer.PORT(2).CORRection.DATA = CorData
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.STATe

SCPI.SENSe(Ch).SWEep.POINts

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:DATA <numeric 1>, ..,<numeric NOP>

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection:DATA?

Query response

{numeric 1},...,{numeric NOP}<newline><^END>

Example of use

```
10 DIM A(1:201)
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:DATA?"
30 ENTER 717;A(*)
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.STATe = *Status*

Status = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).CORRection.STATe

Description

This command turns ON/OFF or returns the status of the power level error correction, for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	*Status*
Description	Turns ON/OFF the power level error correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the power level error correction. • False or OFF: Turns off the power level error correction.
Preset value	False or OFF

Examples

```
Dim PowCorr As Boolean
SCPI.SOURce(1).POWer.PORT(1).CORRection.STATe = True
PowCorr = SCPI.SOURce(1).POWer.PORT(1).CORRection.STATe
```

Equivalent key

Cal > **Power Calibration** > **Correction**

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection[:STATe] {ON|OFF|1|0}

:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}:CORRection[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR ON"
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude = *Value*

Value = SCPI.SOURce(*Ch*).POWer.PORT(*Pt*).LEVel.IMMediate.AMPLitude

Description

This command sets/gets the power level, for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Power level at the specified port.
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range.
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim PowLev As Double
SCPI.SOURce(1).POWer.PORT.COUPle = False
SCPI.SOURce(1).POWer.PORT(1).LEVel.IMMediate.AMPLitude = -12.5
PowLev = SCPI.SOURce(1).POWer.PORT(1).LEVel.IMMediate.AMPLitude
```

Related objects

SCPI.SOURce(Ch).POWer.PORT.COUPle

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

Equivalent key

Sweep Setup > Power > Port Power > Port 1 Power | Port 2 Power | Port 3 Power |Port 4 Power

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}[:LEVel][:IMMediate][:AMPLitude] <numeric>
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}[:LEVel][:IMMediate][:AMPLitude]?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1 -12.5"
20 OUTPUT 717;":SOUR1:POW:PORT1?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.PORT.COUPle

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.PORT.COUPle = *Status*

Status = SCPI.SOURce(*Ch*).POWer.PORT.COUPle

Description

This command sets/gets whether to output the same power level for each port of channels 1 to 36 (*Ch*).

- When the power slope feature is ON, the same power level is always output to all ports regardless of this setting because different power levels cannot be output for each port.

Variable

Parameter	*Status*
Description	Turns ON/OFF the coupling between ports for the power level output
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Outputs the same power level to individual ports. • False or OFF: Outputs different power levels to individual ports.
Preset value	True or ON

Examples

```
Dim OutCpl As Boolean
SCPI.SOURce(1).POWer.PORT.COUPle = False
OutCpl = SCPI.SOURce(1).POWer.PORT.COUPle
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).LEVel.IMMediate.AMPLitude

Equivalent key

Sweep Setup > **Power** > **Port Couple**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT:COUPle {ON|OFF|1|0}
:SOURce{[1]-160}:POWer:PORT:COUPle?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT:COUP OFF"
20 OUTPUT 717;":SOUR1:POW:PORT:COUP?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.SPAN

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.SPAN = *Value*

Value = SCPI.SOURce(*Ch*).POWer.SPAN

Description

This command sets/gets the span value of the sweep range for the power sweep for the channels 1 to 36 (*Ch*).

Variable

Parameter	*Value*
Description	Span value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz –55 to 10 dBm 5 to 6 GHz –55 to 9 dBm 6 to 7 GHz –55 to 8 dBm 7 to 8.5 GHz –55 to 7 dBm
Preset value	15
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pspan As Double
SCPI.SOURce(1).POWer.SPAN = 10
Pspan = SCPI.SOURce(1).POWer.SPAN
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.CENTer

Equivalent key

Span

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:SPAN <numeric>

:SOURce{[1]-160}:POWer:SPAN?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:SPAN 10"
20 OUTPUT 717;":SOUR1:POW:SPAN?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.STARt

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.STARt = *Value*

Value = SCPI.SOURce(*Ch*).POWer.STARt

Description

This command sets/gets the start value of the sweep range for the power sweep for channels 1 to 36 (*Ch*).

Variable

Parameter	*Value*
Description	Start value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz –55 to 10 dBm 5 to 6 GHz –55 to 9 dBm 6 to 7 GHz –55 to 8 dBm 7 to 8.5 GHz –55 to 7 dBm
Preset value	-15
Unit	dBm
Resolution	0.05

Examples

```
Dim Pstart As Double
SCPI.SOURce(1).POWer.STARt = -10
Pstart = SCPI.SOURce(1).POWer.STARt
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.STOP

Equivalent key

Start

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:STARt <numeric>
:SOURce{[1]-160}:POWer:STARt?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:STAR -10"
20 OUTPUT 717;":SOUR1:POW:STAR?"
30 ENTER 717;A
```

SCPI.SOURce(*Ch*).POWer.STOP

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce(*Ch*).POWer.STOP = *Value*

Value = SCPI.SOURce(*Ch*).POWer.STOP

Description

This command sets/gets the stop value of the sweep range for the power sweep for channels 1 to 36 (*Ch*).

Variable

Parameter	*Value*
Description	Stop value
Data type	Double precision floating point type (Double)
Range	Varies depending on the power range: 9 kHz to 5 GHz –55 to 10 dBm 5 to 6 GHz –55 to 9 dBm 6 to 7 GHz –55 to 8 dBm 7 to 8.5 GHz –55 to 7 dBm
Preset value	0
Unit	dBm
Resolution	0.05
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Pstop As Double
SCPI.SOURce(1).POWer.STOP = 10
Pstop = SCPI.SOURce(1).POWer.STOP
```

Related objects

SCPI.SENSe(Ch).SWEep.TYPE

SCPI.SOURce(Ch).POWer.ATTenuation.DATA

SCPI.SOURce(Ch).POWer.STARt

Equivalent key

Stop

Equivalent SCPI command

Syntax

:SOURce{[1]-160}:POWer:STOP <numeric>

:SOURce{[1]-160}:POWer:STOP?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":SOUR1:POW:STOP 10"
20 OUTPUT 717;":SOUR1:POW:STOP?"
30 ENTER 717;A
```

SCPI.SOURce.POWer.PORT.CORRection.COLLect.ASENsor.RCFactor

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce.POWer.PORT.CORRection.COLLect.ASENsor.RCFactor = *Value*

Value = SCPI.SOURce.POWer.PORT.CORRection.COLLect.ASENsor.RCFactor

Description

This command sets/gets the reference calibration coefficient (the calibration coefficient at 50 MHz) for power sensor A.

Variable

Parameter	*Value*
Description	Reference calibration coefficient
Data type	Double precision floating point type (Double)
Range	1 to 150
Preset value	100
Unit	% (percent)
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CalRef As Double
SCPI.SOURce.POWer.PORT.CORRection.COLLect.ASENsor.RCFactor = 99.5
CalRef = SCPI.SOURce.POWer.PORT.CORRection.COLLect.ASENsor.RCFactor
```

Related objects

SCPI.SOURce.POWer.PORT.CORRection.COLLect.BSENsor.RCFactor

Equivalent key

2576

Cal > Power Calibration > Sensor A Settings > Ref Cal Factor

Equivalent SCPI command

Syntax

```
:SOURce:POWer:PORT:CORRection:COLLect:ASENsor:RCFactor
<numeric>
:SOURce:POWer:PORT:CORRection:COLLect:ASENsor:RCFactor?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:ASEN:RCF 99.5"
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:ASEN:RCF?"
30 ENTER 717;A
```

SCPI.SOURce.POWer.PORT.CORRection.COLLect.BSENsor.RCFactor

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce.POWer.PORT.CORRection.COLLect.BSENsor.RCFactor = *Value*

Value = SCPI.SOURce.POWer.PORT.CORRection.COLLect.BSENsor.RCFactor

Description

This command sets/gets the reference calibration coefficient (the calibration coefficient at 50 MHz) for power sensor B.

Variable

Parameter	*Value*
Description	Reference calibration coefficient
Data type	Double precision floating point type (Double)
Range	1 to 150
Preset value	100
Unit	% (percent)
Resolution	0.01
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CalRef As Double
SCPI.SOURce.POWer.PORT.CORRection.COLLect.BSENsor.RCFactor = 99
CalRef = SCPI.SOURce.POWer.PORT.CORRection.COLLect.BSENsor.RCFactor
```

Related objects

SCPI.SOURce.POWer.PORT.CORRection.COLLect.ASENsor.RCFactor

Equivalent key

Cal > **Power Calibration** > **Sensor B Settings** > **Ref Cal Factor**

Equivalent SCPI command

Syntax

```
:SOURce:POWer:PORT:CORRection:COLLect:BSENsor:RCFactor
<numeric>
:SOURce:POWer:PORT:CORRection:COLLect:BSENsor:RCFactor?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:BSEN:RCF 99.5"
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:BSEN:RCF?"
30 ENTER 717;A
```

SCPI.SOURce.POWer.PORT.CORRection.COLLect.NTOLerance

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce.POWer.PORT.CORRection.COLLect.NTOLerance= *Value*

Value = SCPI.SOURce.POWer.PORT.CORRection.COLLect.NTOLerance

Description

This command sets/gets the tolerance of power calibration data for each measurement point, for the selected port (*Pt*) of the selected channel (*Ch*).

Variable

Parameter	*Value*
Description	Tolerance of power calibration
Data type	Double precision floating point type (Double)
Range	0 to 100
Preset value	5
Unit	dB
Resolution	0.001
Note	When the measurement result is beyond the tolerance, an error message appears and the power correction function does not turn on.

Examples

```
Dim TorVal As Long
SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.NTOLerance = 10
TorVal = SCPI.SOURce(1).POWer.PORT(1).CORRection.COLLect.NTOLerance
```

Related objects

SCPI.SOURce(Ch).POWer.PORT(Pt).CORRection.COLLect.ACQuire

Equivalent key

Cal > **Power Calibration** > **Tolerance**

Equivalent SCPI command

Syntax

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}
:CORRection:COLLect:NTOLerance <numeric>
```

```
:SOURce{[1]-160}:POWer:PORT{[1]|2|3|4}
:CORRection:COLLect:NTOLerance?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:NTOL 6"
20 OUTPUT 717;":SOUR1:POW:PORT1:CORR:COLL:NTOL?"
30 ENTER 717;A
```

SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA = *Data*

Data = SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA

Description

This command sets/gets the calibration coefficient table for power sensor A.

Variable

Parameter	*Data*
Description	Indicates the array data (for the calibration coefficient table) of 1 + Num (number of set data items)X2. Where n is an integer between 1 and Num. • *Data(0):* The number of data items you want to set. Specify an integer between 0 to 100. When you set the number of data items to 0 (to clear the calibration coefficient table), you specify only *Data(0)* as the Data variable. • *Data(nX2-1):* The frequency of the n-th data item (1 kHz to 500 GHz). • *Data(nX2):* The calibration coefficient of the n-th data item (1% to 150%). • The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If the array data does not contain 1+Num (number of set data items)x 2 when setting a calibration coefficient table, a runtime error occurs. For *Data(nx2-1)* and *Data(nx2)* in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CalFac As Variant
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA = Array(2,1e7,99.8,
1e9,98.7)
CalFac = SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA
'"Clear Cal Factor Table
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA = Array(0)
```

```
Dim CalFac(4) As Variant
Dim Ref As Variant
CalFac(0) = 2
CalFac(1) = 1e7
CalFac(2) = 99.8
CalFac(3) = 1e9
CalFac(4) = 98.7
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA = CalFac
Ref = SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA
'"Clear Cal Factor Table
Dim CalFac(0) As Variant
CalFac(0) = 0
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA = CalFac
```

Related objects

SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA

Equivalent key

Cal > Power Calibration > Sensor A Settings > Delete | Add | Clear Cal Factor Table

Equivalent SCPI command

Syntax

```
:SOURce:POWer:PORT:CORRection:COLLect:TABLe:ASENsor:DATA
<numeric 1>,...,<numeric 1+(Nx2)>
```

```
:SOURce:POWer:PORT:CORRection:COLLect:TABLe:ASENsor:DATA?
```

Query response

```
{numeric 1},...,{numeric 1+(Nx2)}<newline><^END>
```

Example of use

```
10 DIM B(1:3,1:2)
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA 3,";
30 OUTPUT 717;"1E6,98.5,1E7,99,1E8,99.5"
40 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA?"
50 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:ASEN:DATA 0" !Clear Table
```

SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA

Object type

Property (**Read-Write**)

Syntax

SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA = *Data*

Data = SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA

Description

This command sets/gets the calibration coefficient table for power sensor B.

Variable

Parameter	*Data*
Description	Indicates the array data (for the calibration coefficient table) of 1 + Num (number of set data items)X2. Where n is an integer between 1 and Num. • *Data(0):* The number of data items you want to set. Specify an integer between 0 to 100. When you set the number of data items to 0 (to clear the calibration coefficient table), you specify only *Data(0)* as the Data variable. • *Data(nX2-1):* The frequency of the n-th data item (1 kHz to 500 GHz). • *Data(nX2):* The calibration coefficient of the n-th data item (1% to 150%). The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If the array data does not contain 1+Num (number of set data items)X2 when setting a calibration coefficient table, a runtime error occurs. For *Data(nX2-1)* and *Data(nX2)* in the array data, if the specified value is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim CalFac As Variant
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA = Array(2,1e7,99.8,
1e9,98.7)
CalFac = SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA
'"Clear Cal Factor Table
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA = Array(0)
```

```
Dim CalFac(4) As Variant
Dim Ref As Variant
CalFac(0) = 2
CalFac(1) = 1e7
CalFac(2) = 99.8
CalFac(3) = 1e9
CalFac(4) = 98.7
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA = CalFac
Ref = SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA
'"Clear Cal Factor Table
Dim CalFac(0) As Variant
CalFac(0) = 0
SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.BSENsor.DATA = CalFac
```

Related objects

SCPI.SOURce.POWer.PORT.CORRection.COLLect.TABLe.ASENsor.DATA

Equivalent key

Cal > Power Calibration > Sensor B Settings > Delete | Add | Clear Cal Factor Table

Equivalent SCPI command

Syntax

:SOURce:POWer:PORT:CORRection:COLLect:TABLe:BSENsor:DATA <numeric 1>,...,<numeric 1+(Nx2)>

:SOURce:POWer:PORT:CORRection:COLLect:TABLe:BSENsor:DATA?

<numeric 1> parameter.

Query response

{numeric 1},...,{numeric 1+(Nx2)}<newline><^END>

Example of use

```
10 DIM B(1:3,1:2)
20 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA 3,";
30 OUTPUT 717;"1E6,98.5,1E7,99,1E8,99.5"
40 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA?"
50 ENTER 717;A,B(*)
```

```
10 OUTPUT 717;":SOUR:POW:PORT:CORR:COLL:TABL:BSEN:DATA 0" !Clear Table
```

STATUS

SCPI.STATus.OPERation.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.OPERation.CONDition

Description

This command reads the value of the Operation Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Operation Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.OPERation.CONDition
```

Related objects

SCPI.STATus.OPERation.NTRansition

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:COND?"
20 ENTER 717;A
```

SCPI.STATus.OPERation.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.ENABle = *Value*

Value = SCPI.STATus.OPERation.ENABle

Description

This command sets/gets the value of Operation Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Operation Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 3, bit 6 to13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.ENABle = 16
Stat = SCPI.STATus.OPERation.ENABle
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:ENABle <numeric>

:STATus:OPERation:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:ENAB 16"
20 OUTPUT 717;":STAT:OPER:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.OPERation.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.OPERation.EVENt

Description

This command reads the value of Operation Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Operation Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.OPERation.EVENt
```

Related objects

SCPI.IEEE4882.CLS

SCPI.STATus.OPERation.NTRansition

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER?"
20 ENTER 717;A
```

SCPI.STATus.OPERation.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.NTRansition = *Value*

Value = SCPI.STATus.OPERation.NTRansition

Description

This command sets/gets the value of negative transition filter of the Operation Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 3, bit 6 to13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.NTRansition = 16
Stat = SCPI.STATus.OPERation.NTRansition
```

Related objects

SCPI.STATus.OPERation.EVENt

SCPI.STATus.OPERation.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:NTRansition <numeric>

:STATus:OPERation:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:NTR 16"
20 OUTPUT 717;":STAT:OPER:NTR?"
30 ENTER 717;A
```

SCPI.STATus.OPERation.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.OPERation.PTRansition = *Value*

Value = SCPI.STATus.OPERation.PTRansition

Description

This command sets/gets the value of positive transition filter of the Operation Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	16432
Note	The bit 0 to 3, bit 6 to13 and bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.OPERation.PTRansition = 0
Stat = SCPI.STATus.OPERation.PTRansition
```

Related objects

SCPI.STATus.OPERation.EVENt

SCPI.STATus.OPERation.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:OPERation:PTRansition <numeric>

:STATus:OPERation:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:OPER:PTR 16"
20 OUTPUT 717;":STAT:OPER:PTR?"
30 ENTER 717;A
```

SCPI.STATus.PRESet

Object type

Method (**Write-only**)

Syntax

SCPI.STATus.PRESet

Description

This command initializes all the status registers.

Examples

SCPI.STATus.PRESet

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:PRESet

Equivalent key

No equivalent key is available on the front panel.

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATUS.QUEStionable.BLIMit.CHANnel(Ch).CONDition

Description

This command reads the value of the Questionable Bandwidth Limit Channel Status Condition Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Status Condition Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.CONDition

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Extra Status Condition Register" of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Extra Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUStionable.BLIMit.CHANnel(1).ECHannel.CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.ENABle = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.ENABle

Description

This command sets/gets the value of the Questionable Bandwidth Limit Channel Extra Status Enable Register of the specified channel.

Variable

Parameter	*Value*
Description	Value of the enable register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.ENABle = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.ENABle
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:ENABle <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:ENAB 6"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENt

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Extra Status Event Register" of the specified channel.

Variable

Parameter	*Value*
Description	the value of the Questionable Bandwidth Limit Channel Extra Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition

Description

This command sets/gets the value of the negative transition filter of the "Questionable Bandwidth Limit Channel Extra Status Register" of the specified channel.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Channel Extra Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENt

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:NTR 6"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.PTRansition

Description

This command sets/gets the value of the positive transition filter of the "Questionable Bandwidth Limit Channel Extra Status Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Channel Extra Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ECHannel.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.EVENt

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ECHannel.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ECHannel:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:PTR 6"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ECH:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).ENABle

Description

This command sets/gets the value of the "Questionable Bandwidth Limit Channel Status Enable Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ENABle = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).ENABle
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ENABle <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ENAB 16"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

Description

This command reads the value of the "Questionable Bandwidth Limit Channel Status Event Register" of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Channel Status Event Register
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Description

This command sets/gets the value of the negative transition filter of the "Questionable Bandwidth Limit Channel Status Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Channel Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).NTRansition = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).PTRansition

Description

This command sets/gets the value of the positive transition filter of the "Questionable Bandwidth Limit Channel Status Register" for the specified channel.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Channel Status Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).PTRansition = 0
Stat = SCPI.STATus.QUEStionable.BLIMit.CHANnel(1).PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.BLIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:CHANnel{[1]-160}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.CONDition

Description

This command reads the value of the Questionable Bandwidth Limit Status Condition Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.NTRansition

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.CONDition

Description

This command reads the value of the Questionable Bandwidth Limit Extra Status Condition Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Extra Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ELIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABle = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABle

Description

This command sets/gets the value of the Questionable Bandwidth Limit Extra Status Enable Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Extra Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABle = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ELIMit:ENABle <numeric>

:STATus:QUEStionable:BLIMit:ELIMit:ENABle?

Query response

2614

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:ENAB 6"
20 OUTPUT 717;":STAT:QUES:BLIM:ELIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENt

Description

This command reads the value of the Questionable Bandwidth Limit Extra Status Event Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Extra Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ELIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Bandwidth Limit Extra Status Register.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENt

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ELIMit:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:ELIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:NTR 6"
20 OUTPUT 717;":STAT:QUES:BLIM:ELIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Bandwidth Limit Extra Status Register.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.ELIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.ELIMit.EVENt

SCPI.STATus.QUEStionable.BLIMit.ELIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ELIMit:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:ELIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ELIM:PTR 6"
20 OUTPUT 717;":STAT:QUES:BLIM:ELIM:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.ENABle = Value

Value = SCPI.STATus.QUEStionable.BLIMit.ENABle

Description

This command sets/gets the value of the Questionable Bandwidth Limit Status Enable Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.ENABle = 16
Stat = SCPI.STATus.QUEStionable.BLIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:ENABle <numeric>

:STATus:QUEStionable:BLIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:ENAB 16"
20 OUTPUT 717;":STAT:QUES:BLIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.BLIMit.EVENt

Description

This command reads the value of the Questionable Bandwidth Limit Status Event Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Bandwidth Limit Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.BLIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Bandwidth Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Bandwidth Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.EVENt

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:NTRansition <numeric>

:STATus:QUEStionable:BLIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:NTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.BLIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.BLIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.BLIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Bandwidth Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Bandwidth Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.BLIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.BLIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.BLIMit.EVENt

SCPI.STATus.QUEStionable.BLIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:BLIMit:PTRansition <numeric>

:STATus:QUEStionable:BLIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:BLIM:PTR 16"
20 OUTPUT 717;":STAT:QUES:BLIM:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.CONDition

Description

This command reads the value of the Questionable Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.CONDition
```

Related objects

SCPI.STATus.QUEStionable.NTRansition

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.ENABle

Description

This command sets/gets the value of the Questionable Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 0 to 7 and bit 12 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.ENABle = 6
Stat = SCPI.STATus.QUEStionable.ENABle
```

Related objects

SCPI.IEEE4882.SRE

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:ENABle <numeric>

:STATus:QUEStionable:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:ENAB 16"
20 OUTPUT 717;":STAT:QUES:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.EVENt

Description

This command reads the value of the Questionable Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.EVENt
```

Related objects

SCPI.IEEE4882.CLS

SCPI.STATus.QUEStionable.NTRansition

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Description

This command reads the value of the Questionable Limit Channel Status Condition Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).CONDition

Description

This command reads the value of the Questionable Limit Channel Extra Status Condition Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Extra Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ECHannel.NTRansition

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.ENABle

Description

This command sets/gets the value of the Questionable Limit Channel Extra Status Enable Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Extra Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.ENABle = 6
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.ENABle
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:ENABle <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:ENAB 6"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.EVENt

Description

This command reads the value of the Questionable Limit Channel Extra Status Event Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Extra Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Channel Extra Status Register the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ECHannel.EVENt

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:NTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:NTR 6"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ECHannel.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Limit Channel Extra Status Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ECHannel.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ECHannel.EVENt

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).ECHannel.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:PTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ECHannel:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:PTR 6"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ECH:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).ENABle

Description

This command sets/gets the value of the Questionable Limit Channel Status Enable Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABle = 16
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).ENABle
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ENABle <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ENAB 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).EVENt

Description

This command reads the value of the Questionable Limit Channel Status Event Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Channel Status Event Register of the specified channel
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Channel Status Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition = 16
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:NTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:NTRansition?

Query response

2644

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.CHANnel(*Ch*).PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Limit Channel Status Register of the specified channel (*Ch*).

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition = 0
Stat = SCPI.STATus.QUEStionable.LIMit.CHANnel(1).PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.LIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:PTRansition <numeric>

:STATus:QUEStionable:LIMit:CHANnel{[1]-160}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.CONDition

Description

This command reads the value of the Questionable Limit Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.NTRansition

SCPI.STATus.QUEStionable.LIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.CONDition

Description

This command reads the value of the Questionable Limit Extra Status Condition Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Extra Status Condition Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ELIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:COND?"
20 ENTER 717;A
```

E5071C

SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABle

Description

This command sets/gets the value of the Questionable Limit Extra Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Extra Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABle = 6
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ELIMit:ENABle <numeric>

:STATus:QUEStionable:LIMit:ELIMit:ENABle?

Query response

2650

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:ENAB 6"
20 OUTPUT 717;":STAT:QUES:LIM:ELIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENt

Description

This command reads the value of the Questionable Limit Extra Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Extra Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ELIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Extra Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENt

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ELIMit:NTRansition <numeric>

:STATus:QUEStionable:LIMit:ELIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:NTR 6"
20 OUTPUT 717;":STAT:QUES:LIM:ELIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Limit Extra Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.ELIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.ELIMit.EVENt

SCPI.STATus.QUEStionable.LIMit.ELIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ELIMit:PTRansition <numeric>

:STATus:QUEStionable:LIMit:ELIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ELIM:PTR 6"
20 OUTPUT 717;":STAT:QUES:LIM:ELIM:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.ENABle = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.ENABle

Description

This command sets/gets the value of the Questionable Limit Status Enable Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Status Enable Register
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.ENABle = 16
Stat = SCPI.STATus.QUEStionable.LIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:ENABle <numeric>

:STATus:QUEStionable:LIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:ENAB 16"
20 OUTPUT 717;":STAT:QUES:LIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.LIMit.EVENt

Description

This command reads the value of the Questionable Limit Status Event Register.

Variable

Parameter	*Value*
Description	Value of the Questionable Limit Status Event Register
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.LIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Limit Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.EVENt

SCPI.STATus.QUEStionable.LIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:NTRansition <numeric>

:STATus:QUEStionable:LIMit:NTRansition?

Query response

2660

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.LIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.LIMit.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.LIMit.PTRansition

Description

This command sets/gets the value of positive transition filter of the Questionable Limit Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting for the channel/trace number.
Note	The bit 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.LIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.LIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.LIMit.EVENt

SCPI.STATus.QUEStionable.LIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:LIMit:PTRansition <numeric>

:STATus:QUEStionable:LIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:LIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.NTRansition

Description

This command sets/gets the value of negative transition filter of the Questionable Status Register.

Variable

Parameter	*Value*
Description	Value of the negative transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	The bit (0 to 9) and bit (11 to 15) can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENt

SCPI.STATus.QUEStionable.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:NTRansition <numeric>

:STATus:QUEStionable:NTRansition?

Query response

2664

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:NTR 16"
20 OUTPUT 717;":STAT:QUES:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.PTRansition = *Value*

Value = SCPI.STATus.QUEStionable.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Status Register.

Variable

Parameter	*Value*
Description	Value of the positive transition filter
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	1024
Note	The bit 0 to 9 and bit 11 to 15 can not be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.EVENt

SCPI.STATus.QUEStionable.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:PTRansition <numeric>

:STATus:QUEStionable:PTRansition?

Query response

2666

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:PTR 16"
20 OUTPUT 717;":STAT:QUES:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).CONDition

Description

This command reads the value of the Questionable Ripple Limit Channel Status Condition Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Status Condition Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.CONDition

Description

This command reads the value of the Questionable Ripple Limit Channel Extra Status Condition Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Extra Status Condition Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.NTRansition

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.ENABle = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Channel Extra Status Enable Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Extra Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.ENABle = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.ENABle
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:ENABle <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:ENAB 6"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENt

Description

This command reads the value of the Questionable Ripple Limit Channel Extra Status Event Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Extra Status Event Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ECHannel.NTRansition = *Value*

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(*Ch*).ECHannel.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Channel Extra Status Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Ripple Limit Channel Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENt

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:NTR 6"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Channel Extra Status Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Ripple Limit Channel Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ECHannel.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.EVENt

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ECHannel.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ECHannel:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:PTR 6"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ECH:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Channel Status Enable Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ENABle = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).ENABle
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ENABle <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ENAB 16"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt

Description

This command reads the value of the Questionable Ripple Limit Channel Status Event Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Channel Status Event Register.
Data type	Long integer type (Long)

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Channel Status Register for the specified channel.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Ripple Limit Channel Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).NTRansition = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:NTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Channel Status Register of the specified channel.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Ripple Limit Channel Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

For information on the variable (*Ch*), see Ch.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).PTRansition = 0
Stat = SCPI.STATus.QUEStionable.RLIMit.CHANnel(1).PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).EVENt

SCPI.STATus.QUEStionable.RLIMit.CHANnel(Ch).NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:CHANnel{[1]-160}:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:PTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:CHAN1:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.CONDition

Description

This command reads the value of the Questionable Ripple Limit Status Condition Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.NTRansition

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.CONDition

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.CONDition

Description

This command reads the value of the Questionable Ripple Limit Extra Status Condition Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Extra Status Condition Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.CONDition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ELIMit:CONDition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:COND?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABle = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Extra Status Enable Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Extra Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABle = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ELIMit:ENABle <numeric>

:STATus:QUEStionable:RLIMit:ELIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:ENAB 6"
20 OUTPUT 717;":STAT:QUES:RLIM:ELIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENt

Description

This command reads the value of the Questionable Ripple Limit Extra Status Event Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Extra Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ELIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Extra Status Register.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Ripple Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENt

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ELIMit:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:ELIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:NTR 6"
20 OUTPUT 717;":STAT:QUES:RLIM:ELIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Extra Status Register.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Ripple Limit Extra Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 0 and 3 to 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.ELIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.ELIMit.EVENt

SCPI.STATus.QUEStionable.RLIMit.ELIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ELIMit:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:ELIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ELIM:PTR 6"
20 OUTPUT 717;":STAT:QUES:RLIM:ELIM:PTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.ENABle

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.ENABle = Value

Value = SCPI.STATus.QUEStionable.RLIMit.ENABle

Description

This command sets/gets the value of the Questionable Ripple Limit Status Enable Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Status Enable Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.ENABle = 16
Stat = SCPI.STATus.QUEStionable.RLIMit.ENABle
```

Related objects

SCPI.STATus.QUEStionable.ENABle

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:ENABle <numeric>

:STATus:QUEStionable:RLIMit:ENABle?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:ENAB 16"
20 OUTPUT 717;":STAT:QUES:RLIM:ENAB?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.EVENt

Object type

Property (**Read-only**)

Syntax

Value = SCPI.STATus.QUEStionable.RLIMit.EVENt

Description

This command reads the value of the Questionable Ripple Limit Status Event Register.

Variable

Parameter	*Value*
Description	The value of the Questionable Ripple Limit Status Event Register.
Data type	Long integer type (Long)

Examples

```
Dim Stat As Long
Stat =SCPI.STATus.QUEStionable.RLIMit.EVENt
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit[:EVENt]?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM?"
20 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.NTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.NTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.NTRansition

Description

This command sets/gets the value of the negative transition filter of the Questionable Ripple Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the negative transition filter of the Questionable Ripple Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	0
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.NTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.NTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.EVENt

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:NTRansition <numeric>

:STATus:QUEStionable:RLIMit:NTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:NTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:NTR?"
30 ENTER 717;A
```

SCPI.STATus.QUEStionable.RLIMit.PTRansition

Object type

Property (**Read-Write**)

Syntax

SCPI.STATus.QUEStionable.RLIMit.PTRansition = Value

Value = SCPI.STATus.QUEStionable.RLIMit.PTRansition

Description

This command sets/gets the value of the positive transition filter of the Questionable Ripple Limit Status Register.

Variable

Parameter	*Value*
Description	The value of the positive transition filter of the Questionable Ripple Limit Status Register.
Data type	Long integer type (Long)
Range	0 to 65535
Preset value	Varies depending on the upper limit setting of the number of channels/traces.
Note	Bits 15 cannot be set to 1.

Examples

```
Dim Stat As Long
SCPI.STATus.QUEStionable.RLIMit.PTRansition = 6
Stat = SCPI.STATus.QUEStionable.RLIMit.PTRansition
```

Related objects

SCPI.STATus.QUEStionable.RLIMit.EVENt

SCPI.STATus.QUEStionable.RLIMit.NTRansition

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:STATus:QUEStionable:RLIMit:PTRansition <numeric>

:STATus:QUEStionable:RLIMit:PTRansition?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":STAT:QUES:RLIM:PTR 16"
20 OUTPUT 717;":STAT:QUES:RLIM:PTR?"
30 ENTER 717;A
```

E5071C

SYSTEM

SCPI.SYSTem.BACKlight

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BACKlight = *Status*

Status = SCPI.SYSTem.BACKlight

Description

This command turns ON/OFF or return the status of the backlight of the LCD display.

- When the backlight is OFF, you cannot read the information on the display.

Variable

Parameter	*Status*
Description	ON/OFF of the backlight
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the backlight. • False or OFF: Turns OFF the backlight.
Preset value	True or ON

Examples

```
Dim BckLght As Boolean
SCPI.SYSTem.BACKlight = False
BckLght = SCPI.SYSTem.BACKlight
```

Equivalent key

System > **Backlight**

- To turn the backlight ON, press any key on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:BACKlight {ON|OFF|1|0}

:SYSTem:BACKlight?

2700

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:BACK OFF"
20 OUTPUT 717;":SYST:BACK?"
30 ENTER 717;A
```

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Description

This command generates a beep for the notification of the completion of an operation.

Examples

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Related objects

SCPI.SYSTem.BEEPer.COMPlete.STATe

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Equivalent key

System > **Misc Setup** > **Beeper** > **Test Beep Complete**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:COMPlete:IMMediate

Example of use

```
10 OUTPUT 717;":SYST:BEEP:COMP:IMM"
```

SCPI.SYSTem.BEEPer.COMPlete.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BEEPer.COMPlete.STATe = *Status*

Status = SCPI.SYSTem.BEEPer.COMPlete.STATe

Description

This command turns ON/OFF or returns the status of the beeper for the notification of the completion of the operation.

Variable

Parameter	*Status*
Description	ON/OFF of the beeper for the notification of the completion of the operation
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the beeper for the notification of the completion of the operation. • False or OFF: Turns OFF the beeper for the notification of the completion of the operation.
Preset value	True or ON

Examples

```
Dim BeepComp As Boolean
SCPI.SYSTem.BEEPer.COMPlete.STATe = False
BeepComp = SCPI.SYSTem.BEEPer.COMPlete.STATe
```

Related objects

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

SCPI.SYSTem.BEEPer.WARNing.STATe

Equivalent key

System > **Misc Setup** > **Beeper** > **Beep Complete**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:COMPlete:STATe {ON|OFF|1|0}

:SYSTem:BEEPer:COMPlete:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:BEEP:COMP:STAT OFF"
20 OUTPUT 717;":SYST:BEEP:COMP:STAT?"
30 ENTER 717;A
```

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Description

This command generates a beep for the notification of warning/limit test results.

Examples

SCPI.SYSTem.BEEPer.WARNing.IMMediate

Related objects

SCPI.SYSTem.BEEPer.WARNing.STATe

SCPI.SYSTem.BEEPer.COMPlete.IMMediate

Equivalent key

System > **Misc Setup** > **Beeper** > **Test Beep Warning**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:WARNing:IMMediate

Example of use

```
10 OUTPUT 717;":SYST:BEEP:WARN:IMM"
```

SCPI.SYSTem.BEEPer.WARNing.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.BEEPer.WARNing.STATe = *Status*

Status = SCPI.SYSTem.BEEPer.WARNing.STATe

Description

This command turns ON/OFF or return the status of the beeper for the notification of warning/limit test results.

Variable

Parameter	*Status*
Description	ON/OFF of the beeper for the notification of warning/limit test result
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the beeper for the notification of warning/limit test result. • False or OFF: Turns OFF the beeper for the notification of warning/limit test result.
Preset value	True or ON

Examples

```
Dim BeepWarn As Boolean
SCPI.SYSTem.BEEPer.WARNing.STATe = False
BeepWarn = SCPI.SYSTem.BEEPer.WARNing.STATe
```

Related objects

SCPI.SYSTem.BEEPer.WARNing.IMMediate

SCPI.SYSTem.BEEPer.COMPlete.STATe

Equivalent key

System > **Misc Setup** > **Beeper** > **Beep Warning**

Equivalent SCPI command

Syntax

:SYSTem:BEEPer:WARNing:STATe {ON|OFF|1|0}

:SYSTem:BEEPer:WARNing:STATe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:BEEP:WARN:STAT OFF"
20 OUTPUT 717;":SYST:BEEP:WARN:STAT?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.ECAL.CATalog

Object type

Property (**Read-only**)

Syntax

Ctlg = SCPI.SYSTem.COMMunicate.ECAL.CATalog

Description

This command reads the ID string of ECal that are connected to the ENA through USB. Use the list to select an ECal.

Variable

Parameter	*Ctlg*
Description	Returns a string containing the following parameters for all the connected ECals: 1. Model Number 2. Serial Number
Data type	Character string type (String)

Examples

```
Dim Ctlg As String
Ctlg = SCPI.SYSTem.COMMunicate.ECAL.CATalog
```

Related objects

SCPI.SYSTem.COMMunicate.ECAL.DEFine

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:ECAL:CATalog

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYSTem:COMMunicate:ECAL:CATalog?"
20 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.ECAL.DEFine

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.ECAL.DEFine = *ECalId*

ECalId = SCPI.SYSTem.COMMunicate.ECAL.DEFine

Description

This command sets/gets the ID string of the ECal being used at that time.

Variable

Parameter	*ECalId*
Description	Returns ECalId string containing the following parameters: 1. Model Number 2. Serial Number
Data type	Character string type (String)

Examples

```
Dim ECal As String, ECalReturn As String
SCPI.SYSTem.COMMunicate.ECAL.DEFine = "N4431B 03605"
ECalReturn = SCPI.SYSTem.COMMunicate.ECAL.DEFine
```

Related objects

SCPI.SYSTem.COMMunicate.ECAL.CATalog

Equivalent key

Cal > **ECal** > **ECal**

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:ECAL:DEFine <string>

:SYSTem:COMMunicate:ECAL:DEFine?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYSTem:COMMunicate:ECAL:DEFine ""N4431B 03605"""
10 OUTPUT 717;":SYSTem:COMMunicate:ECAL:DEFine?"
20 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDRess

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDRess = *Value*

Value = SCPI.COMMunicate.GPIB.PMETer.ADDRess

Description

This command sets/gets the GPIB address of the power meter in use.

Variable

Parameter	*Value*
Description	GPIB address of the power meter
Data type	Long integer type (Long)
Range	0 to 30
Preset value	13
Note	If the specified parameter is out of the allowable setting range, a runtime error occurs.

Examples

```
Dim Paddr As Long
SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDRess = 15
Paddr = SCPI.SYSTem.COMMunicate.GPIB.PMETer.ADDRess
```

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Power Meter Address**

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:GPIB:PMETer:ADDRess <numeric>

:SYSTem:COMMunicate:GPIB:PMETer:ADDRess?

Query response

{numeric}<newline><^END>

Example of use

2710

```
10 OUTPUT 717;":SYST:COMM:GPIB:PMET:ADDR 10"
20 OUTPUT 717;":SYST:COMM:GPIB:PMET:ADDR?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDRess

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDRess = *Value*

Value = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDRess

Description

This command sets/gets the GPIB address of the external signal source in use.

Variable

Parameter	*Value*
Description	The GPIB address of the external signal source
Data type	Long integer type (Long)
Range	0 to 30
Preset value	19
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGaddr As Long
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDRess = 20
SGaddr = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.ADDRess
```

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Signal Generator Address**

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:GPIB:SGENerator:ADDRess <numeric>

:SYSTem:COMMunicate:GPIB:SGENerator:ADDRess?

Query response

{value}<newline><^END>

Example of use

2712

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:ADDR 19"
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:ADDR?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency = *Param*

Param = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

Description

This command sets/gets the output frequency setting command for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	*Value*
Description	Frequency setting command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	"FR %f% HZ"
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. The output frequency in the command string is written as "%f%." The frequency of the external signal source, which changes for each measurement point, is set and the setting command is sent to the external signal source.

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency = "FREQ %f%HZ"
SGcomm = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Signal Generator Address** > **Custom Commands** > **Set Frequency**

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:FREQuency <string>

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:FREQuency?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:FREQ ""FREQ %f%HZ"""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:FREQ?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer = *Param*

Param = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

Description

This command sets/gets the output level setting command for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	*Value*
Description	Level setting command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	"AP %p% DM"
Note	If the specified variable is out of the allowable setup range, an error occurs when executed. The output level in the command string is written as "%p%." The output level of the external signal source, which changes for each measurement point, is set and the setting command is sent to the external signal source.

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer = "AMPL %p%DBM"
SGcomm = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System> Misc Setup > GPIB Setup > Signal Generator Address > Custom Commands > Set Power Level

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:POWer <string>

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:POWer?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:POW ""AMPL %p%DBM"""""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:POW?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet = *Param*

Param = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

Description

This command sets/gets the preset command for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	*Value*
Description	Preset command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	""
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet =
"*RST"
SGcomm =
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > **Misc Setup** > **GPIB Setup** > **Signal Generator Address** > **Custom Commands** > **Preset**

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:PRESet <string>

:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:PRESet?

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:PRES ""*RST"""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:PRES?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON = *Param*

Param = SCPI.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

Description

This command sets/gets the command to turn on RF for the external signal source in use.

- This command is available when the type of the external signal source is 1 (user-defined external signal source).

Variable

Parameter	*Value*
Description	RF:on setting command for the user-defined external signal source
Data type	Character string type (String)
Range	254 characters or less
Preset value	"R3"
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGcomm As String
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON = "AMPL:STATE ON"
SGcomm = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address > Custom Commands > Turn RF Out On

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:RFON <string>
:SYSTem:COMMunicate:GPIB:SGENerator:CCOMmand:RFON?
```

Query response

```
{string}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:RFON ""AMPL:STATE ON"""
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:CCOM:RFON?"
30 ENTER 717;A$
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl = *Value*

Value = SCPI.COMMunicate.GPIB.SGENerator.DWELl

Description

This command sets/gets the wait time after setting the frequency or output level for the external signal source in use.

Variable

Parameter	*Value*
Description	Wait time setting for the user-defined external signal source
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0.1
Unit	s (second)
Resolution	0.001
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGdelay As Double
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl = 0.2
SGdelay = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address > Switching Time

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:DWELl <numeric>
:SYSTem:COMMunicate:GPIB:SGENerator:DWELl?
```

Query response

```
{numeric}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:DWEL 0.2"
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:DWEL?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE

Type of object

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE = *Value*

Value = SCPI.COMMunicate.GPIB.SGENerator.TYPE

Description

This command sets/gets the type of the external signal source in use.

Variable

Parameter	*Value*
Description	Type setting of the external signal source 1: User-defined 2: 8643A, 8644B, 8664A, 8665A/B 3: 8648A/B/C/D, ESG Series, PSG Series
Data type	Long integer type (long)
Range	1 to 3
Preset value	3
Note	If the specified variable is out of the allowable setup range, an error occurs when executed.

Example of use

```
Dim SGtype As long
SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE = 2
SGtype = SCPI.SYSTem.COMMunicate.GPIB.SGENerator.TYPE
```

Related objects

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.FREQuency

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.POWer

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.PRESet

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.CCOMmand.RFON

SCPI.SYSTem.COMMunicate.GPIB.SGENerator.DWELl

Equivalent key

System > Misc Setup > GPIB Setup > Signal Generator Address > Custom Commands|8643A,8644B,8664A,8665A/B|8648A/B/C/D,ESG Series,PSG Series

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:GPIB:SGENerator:TYPE <numeric>
:SYSTem:COMMunicate:GPIB:SGENerator:TYPE?
```

Query response

```
{value}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:GPIB:SGEN:TYPE 2"
20 OUTPUT 717;":SYST:COMM:GPIB:SGEN:TYPE?"
30 ENTER 717;A
```

SCPI.SYSTem.COMMunicate.PSENsor

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.COMMunicate.PSENsor = <*char*>,<*string*>

Param = SCPI.SYSTem.COMMunicate.PSENsor

Description

This command specifies the type and location of the power meter to be used in a source power calibration

Variable

Parameter	*char*
Description	Power Meter Type
Data type	Character string type (String)
Range	Select from either of the following: • "GPIB": GPIB Power Meter. • "USB": USB Power Meter.

Parameter	*string*
Description	Address/ID string of Power Meter
Data type	Character string type (String)
Range	For char=GPIB, address of the power meter. Choose any integer between 0 and 30. For char=USB, the ID string of the power meter or power sensor. Use SYST:COMM:USB:PMET:CAT? to see a list of ID strings of connected power meters and sensors.

Examples

```
Dim PsenModel As String
PsenModel = SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog
SCPI.SYSTem.COMMunicate.PSENsor = Array("USB", PsenModel)
```

Related objects

SYSTem.COMMunicate.USB.PMETer.CATalog

Equivalent key

System > **Misc Setup** > **Power Meter Setup** > **Select Type** > **GPIB|USB**

Equivalent SCPI command

Syntax

```
:SYSTem:COMMunicate:PSENsor {GPIB|USB},{string (address/ID string)}
:SYSTem:COMMunicate:PSENsor?
```

Query response

```
{GPIB|USB},{string (address/ID string)}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:COMM:USB:PMET:CAT?"
20 ENTER 717;A$
30 OUTPUT 717;":SYST:COMM:PSEN USB,"&A$
```

SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog

Object type

Property (**Read-only**)

Syntax

Ctlg = SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog

Description

This command reads the ID string of power meters / sensors that are connected to the ENA through USB. Use the list to select a power sensor for a source power cal.

Variable

Parameter	*Ctlg*
Description	Returns a string containing the following parameters for all the connected power meters/sensors: 1. Agilent Technologies 2. Model Number 3. Serial Number 4. Firmware Revision
Data type	Character string type (String)

Examples

```
Dim Ctlg As String
Ctlg = SCPI.SYSTem.COMMunicate.USB.PMETer.CATalog
```

Related objects

SCPI.SYSTem.COMMunicate.PSENsor

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:COMMunicate:USB:PMETer:CATalog

Query response

{string}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYSTem:COMMunicate:USB:PMETer:CATalog?"
20 ENTER 717;A$
```

SCPI.SYSTem.CORRection.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.CORRection.STATe = *Status*

Status = SCPI.SYSTem.CORRection.STATe

Description

This command turns ON/OFF or returns the status of the system correction.

Variable

Parameter	*Status*
Description	ON/OFF of the system correction
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns ON the system correction. • False or OFF: Turns OFF the system correction.
Preset value	True or ON

Examples

```
Dim SysCal As Boolean
SCPI.SYSTem.CORRection.STATe = False
SysCal = SCPI.SYSTem.CORRection.STATe
```

Equivalent key

System > **Misc Setup** > **Service Menu** > **System Correction**

Equivalent SCPI command

Syntax

:SYSTem:CORRection[:STATe] {ON|OFF|1|0}

:SYSTem:CORRection[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:CORR OFF"
20 OUTPUT 717;":SYST:CORR?"
30 ENTER 717;A
```

SCPI.SYSTem.DATE

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.DATE = *Data*

Data = SCPI.SYSTem.DATE

Description

This command sets/gets the date of the clock built in the E5071C.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (date of the built-in clock). • *Data(0)* Sets year. • *Data(1)* Sets month. • *Data(2)* Sets day. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	• *Data(0)* 1980 to 2099 • *Data(1)* 1 to 12 • *Data(2)* 1 to 31
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Day As Variant
SCPI.SYSTem.DATE = Array(2001,12,24)
Day = SCPI.SYSTem.DATE
```

```
Dim Day(2) As Variant
Dim Ref As Variant
Day(0) = 2001
Day(1) = 12
Day(2) = 24
SCPI.SYSTem.DATE = Day
Ref = SCPI.SYSTem.DATE
```

Related objects

SCPI.SYSTem.TIME

SCPI.DISPlay.CLOCk

Equivalent key

System > **Misc Setup** > **Clock Setup** > **Set Date and Time**

Equivalent SCPI command

Syntax

```
:SYSTem:DATE <numeric 1>,<numeric 2>,<numeric 3>
:SYSTem:DATE?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:DATE 2002,1,1"
20 OUTPUT 717;":SYST:DATE?"
30 ENTER 717;A,B,C
```

SCPI.SYSTem.ERRor

Object type

Property (**Read-only**)

Syntax

Err = SCPI.SYSTem.ERRor

Description

This command reads the oldest error from the list of errors stored in the error queue of the E5071C. The read-out error is deleted from the error queue. The size of the error queue is 100.

Executing SCPI.IEEE4882.CLS command clears the errors stored in the error queue.

- This object can not return an error that occurs by the manual operation or the SCPI command used in controlling the E5071C from the external con-troller.

Variable

Parameter	*Err*
Description	Indicates 2-element array data (for error). • *Err(0)* :Error number • *Err(1)* :Error message The index of the array starts from 0.
Data type	Variant type (Variant)
Note	If no error is stored in the error queue, 0 and "No error" are read out as the error number and the error message.

Examples

```
Dim Err As Variant
Err = SCPI.SYSTem.ERRor
```

Related objects

SCPI.IEEE4882.CLS

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:SYSTem:ERRor?

Query response

{numeric},{string}<newline><^END>

{numeric}:

Error number

{string}:

Error message (a character string with double quotation marks (""))

If no error is stored in the error queue, 0 and "No error" are read out as the error number and the error message.

Example of use

```
10 OUTPUT 717;":SYST:ERR?"
20 ENTER 717;A,B$
```

SCPI.SYSTem.ISPControl.PORT

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.ISPControl.PORT = *Value*

Status = SCPI.SYSTem.ISPControl.PORT

Description

This command sets/gets a test port to be selected for stimulus destination when the Initial Source Port Control feature is on.

Variable

Parameter	*Value*
Description	Selected Port
Data type	Long integer type (Long)
Range	1to 4
Preset value	1

- For option 2D5, 4D5, 2K5 and 4K5, the preset value is 1 and cannot be changed. When this command is send to the option 2D5, 4D5, 2K5 and 4K5, an error of "Option Not Installed" is displayed.

Examples

```
SCPI.INIT(ch).CONT = False
SCPI.SYST.ISPC.STAT = True
SCPI.SYST.ISPC.PORT = 1
```

Related objects

SCPI.SYSTem.ISPControl.STATe

Equivalent key

System > **Service** > **Init Src Port [1|2|3|4]**

- For option 2D5, 4D5, 2K5 and 4K5, the key is not displayed.

Equivalent SCPI command

Syntax

:SYSTem:ISPControl:PORT <numeric>

:SYSTem:ISPControl:PORT?

Query response

{1-4}<newline><^END>

Example of use

```
10 :INIT:CONT OFF
20 :SYSTem:ISPControl[:STATe] ON
30 :SYSTem:ISPControl:PORT 1
```

SCPI.SYSTem.ISPControl.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.ISPControl.STATe = *Status*

Status = SCPI.SYSTem.ISPControl.STATe

Description

This command turns ON/OFF or returns the status of the Initial Source Port Control feature (to switch the stimulus output in the trigger hold state to a test port).

Variable

Parameter	*Status*
Description	ON/OFF Initial Source Port Control feature
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: ON Control feature. • False or OFF: OFF Control feature.
Preset value	True or ON

- The default value is ON for all options.

Examples

```
SCPI.INIT(ch).CONT = False
SCPI.SYST.ISPC.STAT = True
SCPI.SYST.ISPC.PORT = 1
```

Related objects

SCPI.SYSTem.ISPC.PORT

Equivalent key

System > **Service** > **Init Src Ctrl [ON]**

Equivalent SCPI command

Syntax

:SYSTem:ISPControl [:STATe] {ON|OFF|1|0}

:SYSTem:ISPControl [:STATe] ?

Query response

{1|0}<newline><^END>

Example of use

```
10 :INIT:CONT OFF
20 :SYSTem:ISPControl[:STATe] ON
30 :SYSTem:ISPControl:PORT 1
```

SCPI.SYSTem.KLOCk.KBD

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.KLOCk.KBD = *Status*

Status = SCPI.SYSTem.KLOCk.KBD

Description

This command sets/gets whether to lock the operation of the front panel (key and rotary knob) and keyboard.

Variable

Parameter	*Status*
Description	ON/OFF of lock
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Specifies lock. • False or OFF: Specifies unlock.
Preset value	False or OFF

Examples

```
Dim FKLock As Boolean
SCPI.SYSTem.KLOCk.KBD = True
FKLock = SCPI.SYSTem.KLOCk.KBD
```

Related objects

SCPI.SYSTem.KLOCk.MOUSe

Equivalent key

System > **Misc Setup** > **Front Panel & Keyboard Lock**

Equivalent SCPI command

Syntax

:SYSTem:KLOCk:KBD {ON|OFF|1|0}

:SYSTem:KLOCk:KBD?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:KLOC:KBD ON"
20 OUTPUT 717;":SYST:KLOC:KBD?"
30 ENTER 717;A
```

SCPI.SYSTem.KLOCk.MOUSe

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.KLOCk.MOUSe = *Status*

Status = SCPI.SYSTem.KLOCk.MOUSe

Description

This command sets/gets whether to lock the operation of the mouse and touch screen.

Variable

Parameter	*Status*
Description	ON/OFF of lock
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Specifies lock. • False or OFF: Specifies unlock.
Preset value	False or OFF

Examples

```
Dim MTLock As Boolean
SCPI.SYSTem.KLOCk.MOUSe = True
MTLock = SCPI.SYSTem.KLOCk.MOUSe
```

Related objects

SCPI.SYSTem.KLOCk.KBD

Equivalent key

System > **Key Lock** > **Mouse Lock**

Equivalent SCPI command

Syntax

:SYSTem:KLOCk:MOUSe {ON|OFF|1|0}

:SYSTem:KLOCk:MOUSe?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:KLOC:MOUS ON"
20 OUTPUT 717;":SYST:KLOC:MOUS?"
30 ENTER 717;A
```

SCPI.SYSTem.POFF

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.POFF

Description

This command turns OFF the E5071C.

Examples

SCPI.SYSTem.POFF

Equivalent key

Standby switch

Equivalent SCPI command

Syntax

:SYSTem:POFF

Example of use

```
10 OUTPUT 717;":SYST:POFF"
```

SCPI.SYSTem.PRESet

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.PRESet

Description

This command presets the setting state of the E5071C to the original factory setting (Default Conditions). This command is different from SCPI.IEEE4882.RSTas the continuous startup mode (see SCPI.INITiate(Ch).CONTinuous) of channel 1 is set to ON.

Examples

SCPI.SYSTem.PRESet

Related objects

SCPI.IEEE4882.RST

SCPI.SYSTem.UPReset

Equivalent key

Preset > **OK**

Equivalent SCPI command

Syntax

:SYSTem:PRESet

Example of use

```
10 OUTPUT 717;":SYST:PRES"
```

SCPI.SYSTem.SECurity.LEVel

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.SECurity.LEVel = Param

Param = SCPI.SYSTem.SECurity.LEVel

Description

This command sets/gets the security level of the E5071C.

Variable

Parameter	*Param*
Description	The security level.
Data type	Character string type (String)
Range	Select from either of the following: • "NONE": Specifies OFF to the security level. • "LOW": Specifies LOW level to the security level. • "HIGH": Specifies HIGH level to the security level.
Preset value	"NONE"
Note	When the setting is LOW, it is able to change to NONE or HIGH. But when this setting is HIGH, it is not able to change NONE or LOW. The setting can be turned NONE by executing the preset or recalling when the setting of frequency blank function is HIGH. Even if the setting is LOW and HIGH, the command that reads out the frequency is not influenced.

Examples

```
Dim SecLev As String
SCPI.SYSTem.SECurity.LEVel = "LOW"
SecLev = SCPI.SYSTem.SECurity.LEVel
```

Equivalent key

System > **Service Menu** > **Security Level** > **None**|**Low**|**High**

Equivalent SCPI command

Syntax

```
:SYSTem:SECurity:LEVel {NONE|LOW|HIGH}
:SYSTem:SECurity:LEVel?
```

Query response

```
{NONE|LOW|HIGH}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:SEC:LEV LOW"
20 OUTPUT 717;":SYST:SEC:LEV?"
30 ENTER 717;A$
```

E5071C

SCPI.SYSTem.SERVice

Object type

Property (**Read-only**)

Syntax

Status = SCPI.SYSTem.SERVice

Description

This command reads whether the E5071C is in the service mode or not.

Variable

Parameter	*Status*
Description	Whether to be in the service mode
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: In the service mode. • False or OFF : Not in the service mode.
Preset Value	False or OFF

Examples

```
Dim SvMode As Boolean
SvMode = SCPI.SYSTem.SERVice
```

Equivalent key

Displayed on the instrument status bar (at the bottom of the LCD display).

Equivalent SCPI command

Syntax

:SYSTem:SERVice?

Query response

{1|0}<newline><^END>

Example of use

2748

```
10 OUTPUT 717;":SYST:SERV?"
30 ENTER 717;A
```

E5071C

SCPI.SYSTem.TEMPerature.HIGH

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.TEMPerature.HIGH = *Status*

Status = SCPI.SYSTem.TEMPerature.HIGH

Description

This command does not works in E5071C. This command is provided because of its command compatibility with E5070B/E5071B. On execution, this command does not works and generates no error.

Variable

Parameter	*Status*
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: No Meaning. • False or OFF: No meaning.
Preset value	False or OFF

Examples

```
Dim TempMode As Boolean
SCPI.SYSTem.TEMPerature.HIGH = True
TempMode = SCPI.SYSTem.TEMPerature.HIGH
```

Equivalent key

None

Equivalent SCPI command

Syntax

:SYSTem:TEMPerature:HIGH {ON|OFF|1|0}

:SYSTem:TEMPerature:HIGH?

Query response

{1|0}<newline><^END>

2750

Example of use

```
10 OUTPUT 717;":SYST:TEMP:HIGH ON"
20 OUTPUT 717;":SYST:TEMP:HIGH?"
30 ENTER 717;A
```

SCPI.SYSTem.TEMPerature.STATe

Object type

Property (**Read-only**)

Syntax

Status = SCPI.SYSTem.TEMPerature.STATe

Description

This command does not works in E5071C. This command is provided because of its command compatibility with E5070B/E5071B. On execution, this command does not works and generates no error.

Variable

Parameter	*Status*
Description	Dummy Parameter
Data type	Boolean type (Boolean)
Range	Always True

Examples

```
Dim WarmUp As Boolean
WarmUp = SCPI.SYSTem.TEMPerature.STATe
```

Equivalent key

None

Equivalent SCPI command

Syntax

:SYSTem:TEMPerature[:STATe]?

Query response

{1}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:TEMP?"
30 ENTER 717;A
```

SCPI.SYSTem.TIME

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.TIME = *Data*

Data = SCPI.SYSTem.TIME

Description

This command sets/gets the time of the clock built in the E5071C.

Variable

Parameter	*Data*
Description	Indicates 3-element array data (time of the built-in clock). *Data(0):* Sets hour (24-hour basis) *Data(1)* :Sets minute. *Data(2)* :Sets second. The index of the array starts from 0.
Data type	Variant type (Variant)
Range	*Data(0)* :0 to 23 *Data(1)* :0 to 59 *Data(2)* :0 to 59
Resolution	1
Note	If the specified variable is out of the allowable setup range, the minimum value (if the lower limit of the range is not reached) or the maximum value (if the upper limit of the range is exceeded) is set.

Examples

```
Dim Time As Variant
SCPI.SYSTem.TIME = Array(21,30,0)
Time = SCPI.SYSTem.TIME
```

```
Dim Time(2) As Variant
Dim Ref As Variant
Time(0) = 21
Time(1) = 30
Time(2) = 0
SCPI.SYSTem.TIME = Time
Ref = SCPI.SYSTem.TIME
```

Related objects

SCPI.SYSTem.DATE

SCPI.DISPlay.CLOCk

Equivalent key

System > **Misc Setup** > **Clock Setup** > **Set Date and Time**

Equivalent SCPI command

Syntax

```
:SYSTem:TIME <numeric 1>,<numeric 2>,<numeric 3>
:SYSTem:TIME?
```

Query response

```
{numeric 1},{numeric 2},{numeric 3}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":SYST:TIME 17,30,0"
20 OUTPUT 717;":SYST:TIME?"
30 ENTER 717;A,B,C
```

SCPI.SYSTem.TSET.EXTernal

Object type

Property (**Read-Write**)

Syntax

SCPI.SYSTem.TSET.EXTernal = *Param*

Param = SCPI.SYSTem.TSET.EXTernal

Description

This command turns on/off the external connection test set, and selects the setting of the test set mode. When the external connection test set is on (MODE1/MODE2), only the S11, S21,or absolute value measurements are available. The fixture simulator's balance measurement function and frequency offset function are not available. When the external connection test set is changed, the calibration coefficients are cleared, and the error correction function is turned off.

Variable

Parameter	*Param*
Description	External test set mode
Data type	Character string type (String)
Range	Select from either of the following: • "OFF": Turns off the external test set. • "MODE1": Turns on the external test set, and sets the test set mode to MODE1. • "MODE2": Turns on the external test set, and sets the test set mode to MODE2.
Preset value	"OFF"

Examples

```
Dim ExtTset As String
SCPI.SYSTem.TSET.EXTernal = "MODE1"
ExtTset = SCPI.SYSTem.TSET.EXTernal
```

Equivalent key

System > **Service Menu** > **External Test Set**

Equivalent SCPI command

Syntax

:SYSTem:TSET:[EXTernal] {OFF|MODE1|MODE2}

:SYSTem:TSET:[EXTernal]?

Query response

{OFF|MODE1|MODE2}<newline><^END>

Example of use

```
10 OUTPUT 717;":SYST:TSET MODE1"
20 OUTPUT 717;":SYST:TSET?"
30 ENTER 717;A
```

SCPI.SYSTem.UPReset

Object type

Method (**Write-only**)

Syntax

SCPI.SYSTem.UPReset

Description

This command presets the E5071C with the user settings. The command is executed regardless of the operation mode in preset state.

- If you try to specify a file for a preset (D:\UserPreset.sta) that does not exist, a warning message will be displayed and SCPI.SYSTem.PRESet will be executed.

Examples

CPI.SYSTem.UPReset

Related objects

SCPI.IEEE4882.RST

SCPI.SYSTem.PRESet

Equivalent key

Preset > **OK**

Equivalent SCPI command

Syntax

:SYSTem:UPReset

Example of use

```
10 OUTPUT 717;":SYST:UPR"
```

TRIGGER

SCPI.TRIGger.OUTPut.POLarity

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.POLarity=*param*

param=SCPI.TRIGger.OUTPut.POLarity

Description

This command sets/gets the polarity of the pulse generated by the External Trigger Output Port.

Variable

Parameter	*Param*
Description	Sets/Gets polarity of the pulse generated by the output trigger.
Data type	Character string type (String)
Range	Select from either of the following: • "POSitive": Generates a Positive pulse. • "NEGative": Generates a Negative pulse.
Preset value	"POSitive"

Examples

```
Dim TrigPol As String
TrigPol="NEGative"
SCPI.TRIGger.OUTPut.POLarity=TrigPol
```

Related objects

SCPI.TRIGger.OUTPut.POSition

SCPI.TRIGger.OUTPut.STATe

Equivalent key

Trigger > **Polarity**

Equivalent SCPI command

Syntax

:TRIGger:OUTPut:POLarity {POSitive|NEGative}

:TRIGger:OUTPut:POLarity?

Query response

{POSitive|NEGative}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:POL POS"
20 OUTPUT 717;":TRIG:OUTP:POL?"
30 ENTER 717;A$
```

SCPI.TRIGger.OUTPut.POSition

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.POSition=Param

Param=SCPI.TRIGger.OUTPut.POSition

Description

This command sets/gets the position of the External Trigger Output Port.

Variable

Parameter	*Param*
Description	Sets/Gets (after or before measurement point) position of the output trigger.
Data type	Character string type (String)
Range	Select from either of the following: • "AFTer": Generates a Pulse (trigger) after the measurement points. • "BEFore": Generates a Pulse (trigger) before the measurement points.
Preset value	"AFTer"

Examples

```
Dim TrigPos As String
TrigPos="BEFore"
SCPI.TRIGger.OUTPut.POSition=TrigPos
```

Related objects

SCPI.TRIGger.OUTPut.POLarity

SCPI.TRIGger.OUTPut.STATe

Equivalent key

Trigger > **Position**

Equivalent SCPI command

Syntax

```
:TRIGger:OUTPut:POSition {AFTer|BEFore}
:TRIGger:OUTPut:POSition?
```

Query response

```
{AFTer|BEFore}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:POS AFT"
20 OUTPUT 717;":TRIG:OUTP:POS?"
30 ENTER 717;A$
```

SCPI.TRIGger.OUTPut.STATe

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.OUTPut.STATe=Param

Param=SCPI.TRIGger.OUTPut.STATe

Description

This command sets/gets the External Trigger Output Port state.

Variable

Parameter	*Param*
Description	Sets/Gets external trigger output state.
Data type	Boolean
Range	Select from either of the following: • True or ON: Turns ON the External Trigger Output • False or OFF : Turns OFF the External Trigger Output
Preset value	"OFF"

Examples

```
Dim TrigStat As boolean
TrigStat=1
SCPI.TRIGger.OUTPut.STATe=TrigStat
```

Related objects

SCPI.TRIGger.OUTPut.POLarity

SCPI.TRIGger.OUTPut.POSition

Equivalent key

Trigger > **Ext Trig Output**

Equivalent SCPI command

Syntax

:TRIGger:OUTPut[:STATe] {ON|1|OFF|0}

:TRIGger:OUTPut[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:OUTP:STAT ON"
20 OUTPUT 717;":TRIG:OUTP:STAT?"
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.AVERage

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.AVERage = *Status*

Status = SCPI.TRIGger.SEQuence.AVERage

Description

This command turns ON/OFF or gets the status of the averaging trigger function.

- The sweep averaging feature must be set to *ON* when turning on the averaging trigger feature.

Variable

Parameter	*Status*
Description	On/off status of the averaging trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following. • True or ON: Turns on the averaging trigger. • False or OFF: Turns off the averaging trigger.
Preset value	False or OFF

Example of use

```
Dim Avetrig As Boolean
SCPI.TRIGger.SEQuence.AVERage = True
Avetrig = TRIGger.SEQuence.AVERage
```

Related objects

SCPI.SENSe(Ch).AVERage.STATe

Equivalent key

Ave > **Ave Trigger**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:AVERage {ON|OFF|1|0}

:TRIGger[:SEQuence]:AVERage?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:AVER ON"
20 OUTPUT 717;":TRIG:AVER?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTernal.DELay

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.EXTernal.DELay = *Value*

Value = SCPI.TRIGger.SEQuence.EXTernal.DELay

Description

This command sets/gets the time that it takes from receiving the trigger to starting measurement when the trigger source is external.

Variable

Parameter	*Value*
Description	External trigger delay time
Data type	Double precision floating point type (Double)
Range	0 to 1
Preset value	0
Unit	s (second)
Resolution	10 µ

Example of use

```
Dim Extdel As Double
SCPI.TRIGger.SEQuence.EXTernal.DELay = 0.05
Extdel = SCPI.TRIGger.SEQuence.EXTernal.DELay
```

Related objects

SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe

SCPI.TRIGger.SEQuence.POINt

Equivalent key

Trigger > **Ext Trigger Delay**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:EXTernal:DELay <numeric>

:TRIGger[:SEQuence]:EXTernal:DELay?

Query response

{numeric}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:EXT:DEL 0.05"
20 OUTPUT 717;":TRIG:EXT:DEL?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe = *Status*

Status = SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe

Description

This command turns ON/OFF or returns the status of the low-latency external trigger feature.

When turning on the low-latency external trigger feature, the point trigger feature must be set to on and the trigger source must be set to external trigger.

- When the point trigger feature is set to off, or when the trigger source is set to one other than the external trigger, the change is ignored.

Variable

Parameter	*Status*
Description	On/off of the low-latency external trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following: • True or ON: Turns on the low-latency external trigger. • False or OFF: Turns off the low-latency external trigger.
Preset value	False or OFF

Example of use

```
Dim Llat As Boolean
SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe = True
Llat = SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe
```

Related objects

SCPI.TRIGger.SEQuence.EXTernal.DELay

SCPI.TRIGger.SEQuence.POINt

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

Trigger > **Low Latency**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:EXTernal:LLATency[:STATe] {ON|OFF|1|0}

:TRIGger[:SEQuence]:EXTernal:LLATency[:STATe]?

Query response

{1|0}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:EXT:LLAT ON"
20 OUTPUT 717;":TRIG:EXT:LLAT?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.EXTernal.SLOPe

Object type

Property

Syntax

SCPI.TRIGger.SEQuence.EXTernal.SLOPe

Description

This command sets/gets the polarity of the External Trigger Input Port.

Variable

Parameter	*Param*
Description	Sets/Gets polarity of the external input trigger.
Data type	Character string type (String)
Range	Select from either of the following: • "POSitive": Sets/Gets Positive polarity. • "NEGative": Sets/Gets Negative polarity.
Preset value	"NEGative"

Examples

```
Dim InputTrigPol As String
InputTrigPol="NEGative"
SCPI.TRIGger.SEQuence.EXTernal.SLOPe=InputTrigPol
```

Related objects

SCPI.TRIGger.SEQuence.EXTernal.DELay

SCPI.TRIGger.SEQuence.EXTernal.LLATency.STATe

Equivalent key

Trigger > **Ext Trig Input**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:EXTernal:SLOPe {POSitive|NEGative}

:TRIGger[:SEQuence]:EXTernal:SLOPe?

Query response

{POSitive|NEGative}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:SEQ:EXT:SLOP POS"
20 OUTPUT 717;":TRIG:SEQ:EXT:SLOP?"
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.IMMediate

Object type

Method (**Write-only**)

Syntax

SCPI.TRIGger.SEQuence.IMMediate

Description

This command generates a trigger immediately and executes a measurement, regardless of the setting of the trigger mode.

This command is different from SCPI.TRIGger.SEQuence.SINGle as the execution of the object finishes at the time of a trigger.

- If you execute this object when the trigger system is not in the trigger wait state (trigger event detection state), an error occurs when executed and the object is ignored.

Examples

```
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.IMMediate
```

Related objects

SCPI.TRIGger.SEQuence.IMMediate

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence][:IMMediate]

Example of use

```
10 OUTPUT 717;":TRIG:IMM"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.POINt

Type of object

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.POINt = *Status*

Status = SCPI.TRIGger.SEQuence.POINt

Description

This command turns ON/OFF or returns the status of the point trigger feature.

- When the point trigger feature is turned on, if the sweep mode is Swept, it changes to Stepped.
  When the trigger source is set to the internal trigger (Internal), the setting is ignored.

Variable

Parameter	*Status*
Description	Turns ON/OFF the point trigger
Data type	Boolean type (Boolean)
Range	Select from either of the following: True or ON: Turns on the point trigger. False or OFF: Turns off the point trigger.
Preset value	False or OFF
Note	

Example of use

```
Dim Ptrig As Boolean
SCPI.TRIGger.SEQuence.POINt = True
Ptrig = TRIGger.SEQuence.POINt
```

Related objects

SCPI.TRIGger.SEQuence.SOURce

Equivalent key

Trigger > **Trigger Event**

Equivalent SCPI command

Syntax

```
:TRIGger[:SEQuence]:POINt {ON|OFF|1|0}
:TRIGger[:SEQuence]:POINt?
```

Query response

```
{1|0}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:POIN ON"
20 OUTPUT 717;":TRIG:POIN?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.SCOPe

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.SCOPe = *Param*

Param = SCPI.TRIGger.SEQuence.SCOPe

Description

This command sets/gets the effective scope of triggering. When this function is enabled with a value of "ACTive", only active channel is triggered. When this function is enabled with a value of "ALL", all channels of the E5071C are triggered.

For example, if TRIGger.SCOPe value is "ACTive" when :INIT:CONT command is turned on for all channels, a measurement channel will automatically be changed by switching over the active channel.

Variable

Parameter	*Param*
Description	Trigger source
Data type	Character string type (String)
Range	Select from either of the following: • "ALL": Specifies trigger to all channels. • "ACTive": Specifies trigger to active channel.
Preset value	"ALL"

Examples

```
Dim TrigScope As Enum
SCPI.TRIGger.SEQuence.SCOPe = "ACTive"
TrigScope = SCPI.TRIGger.SEQuence.SCOPe
```

Equivalent key

Trigger > **Trigger Scope** > **All Channel**|**Active Channel**

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:SCOPe {ALL|ACTive}

:TRIGger[:SEQuence]:SCOPe?

Query response

{ALL|ACTive}<newline><^END>

Example of use

```
10 OUTPUT 717;":TRIG:SEQ:SCOP ACT"
20 OUTPUT 717;":TRIG:SEQ:SCOP?"
30 ENTER 717;A$
```

SCPI.TRIGger.SEQuence.SINGle

Object type

Method (**Write-only**)

Syntax

SCPI.TRIGger.SEQuence.SINGle

Description

This command generates a trigger immediately and executes a measurement, regardless of the setting of the trigger mode.

This command is different from SCPI.TRIGger.SEQuence.IMMediate as the execution of the object finishes when the measurement (all of the sweep) initiated with this object is complete. In other words, you can wait for the end of the measurement using the SCPI.IEEE4882.OPC object.

If you execute this object when the trigger system is not in the trigger wait state (trigger event detection state), an error occurs when executed and the object is ignored.

For details about the trigger system, see Trigger System.

Examples

```
Dim Dmy As Long
SCPI.TRIGger.SEQuence.SOURce = "bus"
SCPI.INITiate(1).CONTinuous = True
SCPI.TRIGger.SEQuence.SINGle
Dmy = SCPI.IEEE4882.OPC
```

Related objects

SCPI.TRIGger.SEQuence.IMMediate

SCPI.IEEE4882.OPC

Equivalent key

No equivalent key is available on the front panel.

Equivalent SCPI command

Syntax

:TRIGger[:SEQuence]:SINGle

Example of use

```
10 OUTPUT 717;":TRIG:SING"
20 OUTPUT 717;"*OPC?"
30 ENTER 717;A
```

SCPI.TRIGger.SEQuence.SOURce

Object type

Property (**Read-Write**)

Syntax

SCPI.TRIGger.SEQuence.SOURce = *Param*

Param = SCPI.TRIGger.SEQuence.SOURce

Description

This command sets/gets the trigger source from the following 4 types:

1. Internal Trigger
   Uses the internal trigger to generate continuous triggers automatically.
2. External Trigger
   Generates a trigger when the trigger signal is inputted externally via the Ext Trig connector or the handler interface.
3. Manual Trigger
   Generates a trigger when the key operation of **Trigger** > **Trigger** is executed from the front panel.
4. Bus Trigger
   Generates a trigger when the SCPI.IEEE4882.TRG object is executed.
5. When you change the trigger source during sweep, the sweep is aborted.

Variable

Parameter	*Param*
Description	Trigger source
Data type	Character string type (String)
Range	Select from either of the following: • "INTernal": Specifies internal trigger. • "EXTernal": Specifies external trigger. • "MANual": Specifies manual trigger. • "BUS": Specifies bus trigger.

Preset value	"INTernal"

Examples

```
Dim TrigSour As String
SCPI.TRIGger.SEQuence.SOURce = "bus"
TrigSour = SCPI.TRIGger.SEQuence.SOURce
```

Equivalent key

Trigger > **Trigger Source** > **Internal**|**External**|**Manual**|**Bus**

Equivalent SCPI command

Syntax

```
:TRIGger[:SEQuence]:SOURce {INTernal|EXTernal|MANual|BUS}
:TRIGger[:SEQuence]:SOURce?
```

Query response

```
{BUS|EXT|INT|MAN}<newline><^END>
```

Example of use

```
10 OUTPUT 717;":TRIG:SOUR BUS"
20 OUTPUT 717;":TRIG:SOUR?"
30 ENTER 717;A$
```

TDR Command Reference

Option TDR Command Reference

The commands for TDR measurement (Option TDR) should be sent via SICL-LAN through TCPIP0::localhost::inst0::INSTR address. The TDR application must be installed and executed on your PC before you send the TDR commands. These commands can not be used in the built-in VBA.

The Command Finder helps you to find the command.

The TDR Notation Convention helps you understand the commands.

Other topics about Programming

Notational Conventions

This section describes the notational conventions used for the description of TDR commands reference.

Command Type

Object type describes different types of E5071C- SCPI commands. The E5071C provides properties and methods as SCPI commands. SCPI commands which set (send)/read (return) the state of the E5071C using variables are defined as property and SCPI commands which does other processing are defined as method.

SCPI commands used to read the state of the E5071C are indicated with "**Query**" and ones used only to set the state of the E5071C are indicated by "**Command**" . SCPI commands that can both read and write data to the E5071C are indicated by '**Command / Query**'.

Syntax

Syntax describes the syntax for sending an SCPI command from the controller (PC) to the E5071C. The syntax consists of two parts: the object part and the set part, with a space " " inserted between them. As for query, the syntax consists of object part followed by a question mark, with no space in between them.

Description

Description describes how to use the SCPI commands or the operation when executed.

Parameters

Parameters provide description about different parameters that can be used with the SCPI commands. It gives the description, data type, allowable range, preset value, unit, resolution, and notes (if any).

The data types of the E5071C SCPI commands include 5 types as shown in the following table.

Data type	Name	Consumed memory	Range
Long	Long integer type	4 bytes	-2,147,483,648 to 2,147,483,647
Double	Double precision floating point type	8 bytes	For a negative value: -1.79769313486232E+308 to -4.94065645841247E-324

			For a positive value: -1.79769313486232E+308 to -4.94065645841247E-324
Boolean	Boolean type	2 bytes	1 (ON) or 0 (OFF)
String	Character string type	1 byte / alphanumeric character	Up to approximately 2 billion characters
Variant	Variant type	16 bytes	No limitation

Examples

Examples provides a sample example of using the object through coding with the Basic language.

Related Commands

Related commands provide information about other commands that are similar/related with the command.

Equivalent GUI

Equivalent GUI shows the operational procedure by using the E5071C-TDR graphic user interface that has the same effect as the object.

CALCULATE

:CALCulate:ALLocate

Type

Command Only

Syntax

:CALCulate:ALLocate {SPARameters|TPARameters|MIXed}

Description

This command sets type of the parameter & format allocation for each traces.

Parameter

Description	Type of the parameter & format allocation
Data Type	Character string type (String)
Range	SPARameters: All S TPARameters: All T MIXed: Mixed
Preset Value	MIXed
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":CALC:ALL SPAR"
```

Equivalent GUI

TDR/TDT > **Trace Control** > **Mixed** OR **All T** OR **All S** button (under **Allocation**)

:CALCulate:ATRaces:ACTive

Type

Command/Query

Syntax

:CALCulate:ATRaces:ACTive <numeric>

:CALCulate:ATRaces:ACTive?

Description

This command sets active trace number.

Parameter

Description	Active trace number
Data Type	Long integer type (Long)
Range	1 to 16
Preset Value	1
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:ATR:ACT 7"
20 OUTPUT 717;":CALC:ATR:ACT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:ATRaces:COUNt

Equivalent GUI

Trace > Select the trace number OR

Trace Next or **Trace Prev**

:CALCulate:ATRaces:COUNt

Type

Query Only

Syntax

:CALCulate:ATRaces:ACTive?

Description

This command returns the number of trace.

Parameter

Description	Number of Trace.
Data Type	Long integer type (Long)
Range	1 to 16
Preset Value	Depending on DUT topology setting
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:ATR:ACT?"
20 ENTER 717;A
```

Related Commands

:CALCulate:ATRaces:ACTive

Equivalent GUI

No equivalent GUI is available.

:CALCulate:ATRaces:MARKer:COUPle

Type

Command/Query

Syntax

:CALCulate:ATRaces:MARKer:COUPle {ON|OFF|1|0}

:CALCulate:ATRaces:MARKer:COUPle?

Description

This command sets state for the marker couple mode.

Parameter

Description	State of marker coupling
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:ATR:MARK:COUP ON"
20 OUTPUT 717;":CALC:ATR:MARK:COUP?"
30 ENTER 717;A
```

Related Commands

:CALCulate:ATRaces:TIME:COUPle

Equivalent GUI

TDR/TDT > **Trace Control** > **Marker** (Under **Coupling**)

:CALCulate:ATRaces:TIME:COUPle

Type

Command/Query

Syntax

:CALCulate:ATRaces:TIME:COUPle {ON|OFF|1|0}

:CALCulate:ATRaces:TIME:COUPle?

Description

This command sets state for the transform couple mode.

Parameter

Description	State of the time coupling.
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:ATR:TIME:COUP ON"
20 OUTPUT 717;":CALC:ATR:TIME:COUP?"
30 ENTER 717;A
```

Related Commands

:CALCulate:ATRaces:MARKer:COUPle

Equivalent Key

TDR/TDT > **Trace Control** > **Time** (Under **Coupling**)

:CALCulate:ATRaces:TIME:STEP:AMPLitude

Type

Command/Query

Syntax

:CALCulate:ATRaces:TIME:STEP:AMPLitude <numeric>

:CALCulate:ATRaces:TIME:STEP:AMPLitude?

Description

This command sets step amplitude value for the transform function.

Parameter

Description	Stimulus amplitude level
Data Type	Double precision floating point type (Double)
Range	1m to 5
Preset Value	200m
Unit	volt
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:ATR:TIME:STEP:AMPL 100e-3"
20 OUTPUT 717;":CALC:ATR:TIME:STEP:AMPL?"
30 ENTER 717;A
```

Equivalent GUI

Setup > **Stim. Ampl.** (under **Basic**)

:CALCulate:DEVice

Type

Command/Query

Syntax

:CALCulate:DEVice {SEND1|SEND2|DIF1|SEND4|DIF2}

:CALCulate:DEVice?

Description

This command sets DUT topology.

Parameter

Description	DUT topology
Data Type	Character string type (String)
Range	SEND1: Single-Ended 1-Port SEND2: Single-Ended 2-Port DIF1: Deferential 1-Port SEND4: Single-Ended 4-Port DIF2: Deferential 2-Port
Preset Value	SEND1
Unit	-
Resolution	-

Query Response

{SEND1|SEND2|DIF1|SEND4|DIF2}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:DEV SEND1"
20 OUTPUT 717;":CALC:DEV?"
30 ENTER 717;A$
```

Equivalent GUI

Setup > **DUT Topology** (under **Basic**)

:CALCulate:EYE:ABORt

Type

Command Only

Syntax

:CALCulate:EYE:ABORt

Description

This command aborts calculations for the simulated eye diagram.

Examples

```
10 OUTPUT 717;":CALC:EYE:ABOR"
```

Related Commands

:CALCulate:EYE:EXECute

Equivalent Key

Eye/Mask > **Abort** (Under **Eye Control**)

:CALCulate:EYE:EXECute

Type

Command Only

Syntax

:CALCulate:EYE:EXECute

Description

This command performs the calculation for the simulated eye diagram for the active trace.

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":CALC:EYE:EXEC"
```

Related Commands

:CALCulate:EYE:STATe

:CALCulate:ATRaces:ACTive

:CALCulate:EYE:ABORt

Equivalent Key

Eye/Mask > **Draw Eye** (Under **Eye Control**)

:CALCulate:EYE:INPut:BPATtern:LENGth

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:BPATtern:LENGth <numeric>

:CALCulate:EYE:INPut:BPATtern:LENGth?

Description

This command sets bits' power of 2 for PRBS pattern. This value is used only when the bit pattern type is selected at PRBS.

Parameter

Description	Bit length
Data Type	Long integer type (Long)
Range	3 to 15
Preset Value	7
Unit	-
Resolution	2

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE PRBS"
20 OUTPUT 717;":CALC:EYE:INP:BPAT:LENG 3"
```

Related Commands

:CALCulate:EYE:INPut:BPATtern:TYPE

Equivalent GUI

Eye Mask > **Length** (under **Bit Pattern**)

:CALCulate:EYE:INPut:BPATtern:TYPE

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:BPATtern:TYPE {PRBS|K285|USER}

:CALCulate:EYE:INPut:BPATtern:TYPE?

Description

This command sets the bit pattern type for the simulated eye function.

Parameter

Description	Bit pattern type
Data Type	Character string type (String)
Range	PRBS: Pseudo-Random Bit Sequence K285: K 28.5 USER: User Custom
Preset Value	PRBS
Unit	-
Resolution	-

Query Response

{PRBS|K285|USER}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE K285"
20 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:EYE:INPut:BPATtern:LENGth

:MMEMory:LOAD:EYE:BPATtern

Equivalent GUI

Eye Mask > **Type** (under **Bit Pattern**)

:CALCulate:EYE:INPut:DRATe

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:DRATe <numeric>

:CALCulate:EYE:INPut:DRATe?

Description

This command sets the bit rate for the simulated eye function.

Parameter

Description	Bit Rate
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	1G
Unit	bits/sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:DRAT 1.1e9"
20 OUTPUT 717;":CALC:EYE:INP:DRAT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:EYE:INPut:OLEVel

:CALCulate:EYE:INPut:ZLEVel

:CALCulate:EYE:INPut:RTIMe:DATA

:CALCulate:EYE:INPut:RTIMe:THReshold

Equivalent GUI

Eye/Mask > **Data Rate** (Under **Bit Pattern**)

:CALCulate:EYE:INPut:OLEVel

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:OLEVel <numeric>

:CALCulate:EYE:INPut:OLEVel?

Description

This command sets the voltage level for bit "1" for the simulated eye function.

Parameter

Description	Level for bit "1"
Data Type	Double precision floating point type (Double)
Range	-5 to +5
Preset Value	200m
Unit	volt
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:OLEV 100e-3"
20 OUTPUT 717;":CALC:EYE:INP:ZLEV -100e-3"
30 OUTPUT 717;":CALC:EYE:INP:OLEV?"
40 ENTER 717;A
50 OUTPUT 717;":CALC:EYE:INP:ZLEV?"
60 ENTER 717;B
```

Related Commands

:CALCulate:EYE:INPut:ZLEVel

:CALCulate:EYE:INPut:OFFSet

Equivalent GUI

Eye Mask > **One Lv.** (Under **Bit Pattern**)

:CALCulate:EYE:INPut:RTIMe:DATA

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:RTIMe:DATA <numeric>

:CALCulate:EYE:INPut:RTIMe:DATA?

Description

This command sets the rise time value for the simulated eye function.

Parameter

Description	Rise time
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	35p
Unit	second
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:RTIM:DATA 90e-12"
20 OUTPUT 717;":CALC:EYE:INP:RTIM:DATA?"
30 ENTER 717;A
```

Related Commands

:CALCulate:EYE:INPut:RTIMe:THReshold

Equivalent GUI

Eye Mask > **Rise Time** (Under **Bit Pattern**)

:CALCulate:EYE:INPut:RTIMe:THReshold

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:RTIMe:THReshold {T1_9|T2_8}

:CALCulate:EYE:INPut:RTIMe:THReshold?

Description

This command sets the rise time threshold for the simulated eye.

Parameter

Description	Treshold levels define rise time
Data Type	Character string type (String)
Range	T1_9: 10%-90% T2_8: 20%-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:RTIM:THR T1_9"
20 OUTPUT 717;":CALC:EYE:INP:RTIM:THR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:EYE:INPut:RTIMe:DATA

Equivalent GUI

Eye Mask > **Rise Time** (Under **Bit Pattern**)

:CALCulate:EYE:INPut:ZLEVel

Type

Command/Query

Syntax

:CALCulate:EYE:INPut:ZLEVel <numeric>

:CALCulate:EYE:INPut:ZLEVel?

Description

This command sets the voltage level for bit "0" for the simulated eye function.

Parameter

Description	level for bit "0"
Data Type	Double precision floating point type (Double)
Range	-5 to 5
Preset Value	0
Unit	volt
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:INP:OLEV 100e-3"
20 OUTPUT 717;":CALC:EYE:INP:ZLEV -100e-3"
30 OUTPUT 717;":CALC:EYE:INP:OLEV?"
40 ENTER 717;A
50 OUTPUT 717;":CALC:EYE:INP:ZLEV?"
60 ENTER 717;B
```

Related Commands

:CALCulate:EYE:INPut:OLEVel

:CALCulate:EYE:INPut:OFFSet

Equivalent GUI

Eye Mask > **Zero Lv.** (Under **Bit Pattern**)

:CALCulate:EYE:MASK:FAIL

Type

Query Only

Syntax

:CALCulate:EYE:MASK:FAIL?

Description

This command returns the mask test result.

Parameter

Description	Mask test result
Data Type	Boolean type (Boolean)
Range	1: Mask Test Fail 0: Mask Test Pass
Preset Value	-
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":MMEM:LOAD:EYE:MASK ""D:\TDR\FC0133.msk"""
40 OUTPUT 717;":CALC:EYE:MASK:STAT ON"
50 OUTPUT 717;":CALC:EYE:EXEC""
60 OUTPUT 717;":CALC:EYE:MASK:FAIL?"
70 ENTER 717;A
```

Related Commands

:CALCulate:EYE:MASK:STATe

Equivalent GUI

Eye/Mask > Mask Test Result in windows

:CALCulate:EYE:MASK:STATe

Type

Command/Query

Syntax

:CALCulate:EYE:MASK:STATe {ON|OFF|1|0}

:CALCulate:EYE:MASK:STATe?

Description

This command sets mask test with simulated eye on/off.

Parameter

Description	Mask Test ON/OFF
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":MMEM:LOAD:EYE:MASK ""D:\TDR\FC0133.msk"""
40 OUTPUT 717;":CALC:EYE:MASK:STAT ON"
50 OUTPUT 717;":CALC:EYE:EXEC"
```

Related Commands

:CALCulate:EYE:MASK:STATe

:CALCulate:EYE:EXECute

:CALCulate:EYE:MASK:FAIL

:MMEMory:LOAD:EYE:MASK

Equivalent GUI

Eye Mask > **Mask Test** (Under **Result**)

:CALCulate:EYE:RESults:DATA

Type

Query Only

Syntax

:CALCulate:EYE:RESults:DATA?

Description

This command returns the results of eye measurement. The 18 values are returned. The minimum and maximum values are returned in addition of the displayed results (16 values) on the TDR application GUI.

Parameter

Description	Results of eye measurement
Data Type	Variant type Array (Range)
Range	-
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric (Minimum Value)},{numeric (Maximum Value)},{numeric (Level Zero)},{numeric (Level One)}, {numeric (Level Mean)},{numeric (Amplitude)},{numeric (Height)},{numeric (always 0)}, {numeric (Width)},{numeric (Opening Factor)}, {numeric (Signal/Noise Ratio)}, {numeric (Duty Cycle Distortion)},{numeric (Duty Cycle Distortion-%)},{numeric (Rise Time)},{numeric (Fall Time)},{numeric (Jitter peak to peak)},{numeric (Jitter RMS)},{numeric (Crossing Percentage)}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":CALC:EYE:RES:THR T2_8"
30 OUTPUT 717;":CALC:EYE:EXEC"
40 OUTPUT 717;":CALCulate:EYE:RESults:DATA?"
50 ENTER 717;A(*)
```

Related Commands

:CALCulate:EYE:RESults:THReshold

E5071C

:CALCulate:EYE:EXECute

Equivalent GUI

Eye Mask > Result Window (Under **Result**)

Eye Mask > **Export to CSV** (Under **Result**)

(Not exactly equivalent)

:CALCulate:EYE:RESults:DISPlay:STATe

Type

Command/Query

Syntax

:CALCulate:EYE:RESults:DISPlay:STATe {ON|OFF|1|0}

:CALCulate:EYE:RESults:DISPlay:STATe?

Description

This command turns the overlay on/off.

Parameter

Description	Overlay on/off
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:STAT ON"
20 OUTPUT 717 ":CALC:ATR:ACT 3"
30 OUTPUT 717;":CALC:EYE:EXEC"
40 OUTPUT 717;":CALC:EYE:RES:DISP:STAT OFF"
```

Related Commands

:CALCulate:EYE:RESults:DATA

Equivalent GUI

Eye Mask > **Overlay** (Under **Result**)

:CALCulate:EYE:RESults:THReshold

Type

Command/Query

Syntax

:CALCulate:EYE:RESults:THReshold {T1_9|T2_8}

:CALCulate:EYE:RESults:THReshold?

Description

This command sets the rise time threshold level for the results of eye measurement.

Parameter

Description	Rise time threshold level
Data Type	Character string type (String)
Range	T1_9: 10-90% T2_8: 20-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:EYE:RES:THR T1_9"
20 OUTPUT 717;":CALC:EYE:RES:THR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:EYE:RESults:DATA

Equivalent GUI

Eye Mask > Rise Time Def. (Under **Result**)

:CALCulate:EYE:STATe

Type

Command/Query

Syntax

:CALCulate:EYE:STATe {ON|OFF|1|0}

:CALCulate:EYE:STATe?

Description

This command displays the EYE/MASK window.

Parameter

Description	Status of Eye/Mask window
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

See the example in :CALCulate:EYE:EXECute

Related Commands

:CALCulate:EYE:EXECute

Equivalent GUI

Eye Mask

:CALCulate:TRACe{Tr}:AMARkers:ACTive

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:AMARkers:ACTive <numeric>

:CALCulate:TRACe{Tr}:AMARkers:ACTive?

Description

This command sets active marker number.

Parameter

Description	Marker number to active
Data Type	Long integer type (Long)
Range	0 to 10
Preset Value	0
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:AMAR:ACT 1"
20 OUTPUT 717;":CALC:TRAC1:AMAR:ACT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:EXECute

Equivalent GUI

Marker > Select the marker number OR

Marker > Select the marker number

:CALCulate:TRACe{Tr}:CONVersion:PEELing:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:CONVersion:PEELing:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:CONVersion:PEELing:STATe?

Description

This command sets state for the peeling function.

Parameter

Description	Peeling function on/off.
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:CONV:PEEL:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:CONV:PEEL:STAT?"
30 ENTER 717;A
```

Equivalent GUI

TDR/TDT > **Parameters** > **Peeling**

:CALCulate:TRACe{Tr}:DTIMe:DATA

Type

Query Only

Syntax

:CALCulate:TRACe{Tr}:DTIMe:DATA?

Description

This command gets delta time result value. You can get the result even if :CALCulate:TRACe{Tr}:DTIMe:STATe is off.

Parameter

Description	Value of delta time result.
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:DTIM:TARG 5"
20 OUTPUT 717;":CALC:TRAC1:DTIM:POS 50"
30 OUTPUT 717;":CALC:TRAC1:DTIM:DATA?"
40 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:DTIMe:POSition

:CALCulate:TRACe{Tr}:DTIMe:STATe

:CALCulate:TRACe{Tr}:DTIMe:TARGet

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:DTIMe:POSition

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:DTIMe:POSition <numeric>

:CALCulate:TRACe{Tr}:DTIMe:POSition?

Description

This command sets delta time reference position.

Parameter

Description	Delta time reference position.
Data Type	Double precision floating point type (Double)
Range	0 to 100
Preset Value	50
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:DTIM:POS 0"
20 OUTPUT 717;":CALC:TRAC1:DTIM:POS?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:DTIMe:DATA

:CALCulate:TRACe{Tr}:DTIMe:STATe

:CALCulate:TRACe{Tr}:DTIMe:TARGet

Equivalent GUI

Marker Search > **Δ Time** > **Position(%)** (In Δ Time Dialog box)

:CALCulate:TRACe{Tr}:DTIMe:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:DTIMe:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:DTIMe:STATe?

Description

This command displays the delta time marker in the marker search..

Parameter

Description	Status of display of delta time marker
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0} <newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:DTIM:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:DTIM:STAT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:DTIMe:DATA

:CALCulate:TRACe{Tr}:DTIMe:POSition

:CALCulate:TRACe{Tr}:DTIMe:TARGet

Equivalent GUI

Marker Search > **Δ Time** > **Δ Time** (Checkbox)

:CALCulate:TRACe{Tr}:DTIMe:TARGet

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:DTIMe:TARGet <numeric>

:CALCulate:TRACe{Tr}:DTIMe:TARGet?

Description

This command sets target trace number for the delta time function. The {Tr} is the trace number starting point for delta time. The <numeric> is the trace number stopping point for delta time.

Parameter

Description	Target trace number for the delta time function.
Data Type	Long integer type (Long)
Range	1 to 16
Preset Value	1
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:DTIM:TARG 5"
20 OUTPUT 717;":CALC:TRAC1:DTIM:TARG?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:DTIMe:DATA

:CALCulate:TRACe{Tr}:DTIMe:POSition

:CALCulate:TRACe{Tr}:DTIMe:STATe

Equivalent GUI

Marker Search > **Δ Time** > trace selection (In Δ Time Dialog box)

:CALCulate:TRACe{Tr}:FORMat

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:FORMat
{IMPedance|VOLT|MLOGarithmic|MLINear|REAL|IMAGinary|GDELay|SWR|PHASe|UPHase|PPHase|SLINear|SLOGarithmic|SCOMplex|SMITh|SADMittance|PLINear|PLOGarithmic|POLar}

:CALCulate:TRACe{Tr}:FORMat?

Description

This command sets trace format.

Parameter

Description	Format for the trace.
Data Type	Character string type (String)
Range	For S-Parameter Measurement: IMPedance|MLOGarithmic|MLINear|REAL|IMAGinary|GDELay|SWR|PHASe|UPHase|PPHase|SLINear|SLOGarithmic|SCOMplex|SMITh|SADMittance|PLINear|PLOGarithmic|POLar For Time Domain measurement: IMPedance|VOLT|MLOGarithmic|MLINear|REAL
Preset Value	Depending on trace.
Unit	-

Resolution	-

Query Response

{IMPedance|VOLT|MLOGarithmic|MLINear|REAL|IMAGinary|GDELay|SWR|PHASe|UPHase|PPHase|SLINear|SLOGarithmic|SCOMplex|SMITh|SADMittance|PLINear|PLOGarithmic|POLar}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:FORM IMP"
20 OUTPUT 717;":CALC:TRAC1:FORM?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:PARameter

Equivalent Key

TDR/TDT > **Parameters**

:CALCulate:TRACe{Tr}:GATE:STARt

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:STARt <numeric>

:CALCulate:TRACe{Tr}:GATE:STARt?

Description

This command sets start value for the gating function.

Parameter

Description	Starting point of the gate
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"

50 OUTPUT 717;":CALC:TRAC1:GATE:STAR?"
60 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:GATE:STOP

:CALCulate:TRACe{Tr}:GATE:TYPE

:CALCulate:TRACe{Tr}:GATE:STATe

Equivalent GUI

TDR/TDT > **Gating** > **Start** virtual knob OR

TDR/TDT > **Gating** > type in the text box under the **Start** virtual knob OR
Right-click on the time domain graph plot and select **Set Gating Range**

:CALCulate:TRACe{Tr}:GATE:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:GATE:STATe?

Description

This command sets state for the gating function.

Parameter

Description	Activating the gating function
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"

50 OUTPUT 717;":CALC:TRAC1:GATE:STAT?"
60 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:GATE:STARt

:CALCulate:TRACe{Tr}:GATE:STOP

:CALCulate:TRACe{Tr}:GATE:TYPE

Equivalent GUI

TDR/TDT > **Gating** > select the **Gating** check box OR

Right-click on the time domain active trace and select **Switch Gating State**

:CALCulate:TRACe{Tr}:GATE:STOP

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:STOP <numeric>

:CALCulate:TRACe{Tr}:GATE:STOP?

Description

This command sets stop value for the gating function.

Parameter

Description	Stopping point of the gate
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"
50 OUTPUT 717;":CALC:TRAC1:GATE:STOP?"
60 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:GATE:STARt

:CALCulate:TRACe{Tr}:GATE:TYPE

:CALCulate:TRACe{Tr}:GATE:STATe

Equivalent GUI

TDR/TDT > **Gating** > **Stop** virtual knob OR

TDR/TDT > **Gating** > type in the text box under the **Stop** virtual knob OR

Right-click on the time domain graph plot and select **Set Gating Range**

:CALCulate:TRACe{Tr}:GATE:TYPE

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:GATE:TYPE {BPASs|NOTCh}

:CALCulate:TRACe{Tr}:GATE:TYPE?

Description

This command sets gate type for the gating function

Parameter

Description	Gate type for the gating function
Data Type	Character string type (String)
Range	BPASs: Band Pass NOTCh: Notch
Preset Value	BPASs
Unit	-
Resolution	-

Query Response

{BPASs|NOTCh}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:GATE:STAR 2e-9"
20 OUTPUT 717;":CALC:TRAC1:GATE:STOP 6e-9"
30 OUTPUT 717;":CALC:TRAC1:GATE:TYPE BPAS"
40 OUTPUT 717;":CALC:TRAC1:GATE:STAT ON"
50 OUTPUT 717;":CALC:TRAC1:GATE:TYPE?"
60 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:GATE:STARt

:CALCulate:TRACe{Tr}:GATE:STOP

:CALCulate:TRACe{Tr}:GATE:STATe

Equivalent GUI

TDR/TDT > **Gating** > **Notch** OR **Bandpass** (under **Type**)

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:EXECute

Type

Command Only

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:EXECute

Description

This command executes marker Max or Min for a selected marker function. This is different than tracking mode because moving the marker to another point, the marker does not point back to Max or Min that is previously set.

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE MAX"
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:EXEC"
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TRACking

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TYPE

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TRACking

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TRACking {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TRACking?

Description

This command sets state for the marker tracking mode. Using this command makes the marker constantly pointing to the point whereby the marker is either set to Max or Min and is not able to move to other desire point.

Parameter

Description	Status for the marker tracking mode
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE MAX"
20 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TRAC ON"
30 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TRAC?
40 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:EXECute

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TYPE

Equivalent GUI

Marker Search > **Max** or **Min**
(:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TYPE is used together when equivalent GUI is clicked)

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TYPE

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TYPE {MAXimum|MINimun}

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TYPE?

Description

This command sets type for the marker function. The equivalent GUI use the tracking mode command. Alternatively, execute command can be used.

Parameter

Description	Marker function type
Data Type	Character string type (String)
Range	MAX: The marker moves to the maximum value of the trace. MIN: The marker moves to the minimum value of the trace.
Preset Value	MAXimum
Unit	-
Resolution	-

Query Response

{MAXimum|MINimun}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE MAX"
20 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TRAC ON"
30 OUTPUT 717;":CALC:TRAC1:MARK1:FUNC:TYPE?"
40 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:EXECute

:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TRACking

Equivalent GUI

Marker Search > **Max** or **Min**
(:CALCulate:TRACe{Tr}:MARKer{Mk}:FUNCtion:TRACking is used together when equivalent GUI is clicked)

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe?

Description

This command sets state for the reference marker display.

Parameter

Description	Status of the reference marker display
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:REF:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:MARK1:REF:STAT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe

Equivalent GUI

Marker > **Ref** OR

Setup > **Advanced Mode** > **Marker** > **Marker Ref**

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:MARKer{Mk}:STATe?

Description

This command sets state for the marker display.

Parameter

Description	Status for the marker display
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:MARK1:STAT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:REFerence:STATe

Equivalent GUI

Marker OR

Setup > **Advanced Mode** > **Marker**

:CALCulate:TRACe{Tr}:MARKer{Mk}:X

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:X <numeric>

:CALCulate:TRACe{Tr}:MARKer{Mk}:X?

Description

This command sets stimulus value of the marker.

Parameter

Description	Marker stimulus value (X-Axis)
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:X 50"
20 OUTPUT 717;":CALC:TRAC1:MARK1:X?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:Y

Equivalent GUI

Setup > **Advanced Mode** > **Marker** > Select Marker (Entry box above the graph plot)

:CALCulate:TRACe{Tr}:MARKer{Mk}:Y

Type

Query Only

Syntax

:CALCulate:TRACe{Tr}:MARKer{Mk}:Y?

Description

This command gets response values of the marker.

Parameter

Description	Marker response value (Y-axis)
Data Type	Double precision floating point type (Double)
Range	- -
Preset Value	- -
Unit	- -
Resolution	- -

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:MARK1:Y?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:MARKer{Mk}:X

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:MATH:MEMorize

Type

Command Only

Syntax

:CALCulate:TRACe{Tr}:MATH:MEMorize

Description

This command copies measurement data to memory trace.

Examples

10 OUTPUT 717;":CALC:TRAC1:MATH:MEM"

Related Commands

:DISPlay:TRACe{Tr}:DMEMory:TYPE

Equivalent GUI

Data Mem > **Data -> Mem**

:CALCulate:TRACe{Tr}:PARameter

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:PARameter {String}

:CALCulate:TRACe{Tr}:PARameter?

Description

This command sets measurement parameter. See Measurement Parameters of Balanced Devices for balanced parameters.

Parameter

Description	Measurement parameter
Data Type	Character string type (String)
Range	For Time Domain • Txy • Tddxy • Tdcxy • Tcdxy • Tccxy For S-parameter • Sxy • Sddxy • Sdcxy • Scdxy • Sccxy x: 1 to 4 y: 1 to 4
Preset Value	Depending on Trace and Selected DUT topology
Unit	-
Resolution	-

Query Response

{String}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:PAR T11"
20 OUTPUT 717;":CALC:TRAC1:PAR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:FORMat

Equivalent GUI

TDR/TDT > **Parameter**

:CALCulate:TRACe{Tr}:SMOothing:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:SMOothing:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:SMOothing:STATe?

Description

This command sets state for smoothing.

Parameter

Description	Status for smoothing
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:SMO:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:SMO:STAT?"
30 ENTER 717;A
```

Equivalent GUI

TDR/TDT > **Parameters** > **Smoothing**

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh <numeric>

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh?

Description

This command sets impulse width value for the transform function.

Parameter

Description	Transform function impulse width value
Data Type	Double precision floating point type (Double)
Range	30.2p to 20.8n
Preset Value	35p
Unit	sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TIME:IMP:WIDT 17e-12"
20 OUTPUT 717;":CALC:TRAC1:TIME:IMP:WIDT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:TIME:STIMulus

Equivalent GUI

TDR/TDT > **Parameters** > **Lowpass Impulse** (In **Stimulus**)

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:DATA

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:DATA <numeric>

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:DATA?

Description

This command sets rise time value for the transform function.

Parameter

Description	Transform function rise time value
Data Type	Double precision floating point type (Double)
Range	15.9p to 20.8n
Preset Value	35p
Unit	sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:DATA 0"
20 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:DATA?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:THReshold

:CALCulate:TRACe{Tr}:TIME:STIMulus

Equivalent GUI

TDR/TDT > **Parameters** > Entry box (In **Rise Time**)

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:THReshold

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:THReshold {T1_9|T2_8}

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:THReshold?

Description

This command sets rise time threshold for the transform function.

Parameter

Description	Transform function rise time threshold
Data Type	Character string type (String)
Range	T1_9 : 10-90% T2_8 : 20-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:THR T1_9"
20 OUTPUT 717;":CALC:TRAC1:TIME:STEP:RTIM:THR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:DATA

:CALCulate:TRACe{Tr}:TIME:STIMulus

Equivalent GUI

TDR/TDT > **Parameters** > **10-90%** or **20-80%** (In **Stimulus**)

:CALCulate:TRACe{Tr}:TIME:STIMulus

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TIME:STIMulus {LPSTep|LPIMpulse}

:CALCulate:TRACe{Tr}:TIME:STIMulus?

Description

This command sets stimulus type for the transform function.

Parameter

Description	Transform function stimulus type
Data Type	Character string type (String)
Range	LPSTep: Lowpass Step LPIMpulse: Lowpass Impulse
Preset Value	LPSTep
Unit	-
Resolution	-

Query Response

{LPSTep|LPIMpulse}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TIME:STIM LPSTep"
20 OUTPUT 717;":CALC:TRAC1:TIME:STIM?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:TIME:IMPulse:WIDTh

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:DATA

:CALCulate:TRACe{Tr}:TIME:STEP:RTIMe:THReshold

Equivalent GUI

TDR/TDT > **Parameters** > **Lowpass Step** or **Lowpass Impulse** (In **Stimulus**)

:CALCulate:TRACe{Tr}:TTIMe:DATA

Type

Query Only

Syntax

:CALCulate:TRACe{Tr}:TTIMe:DATA?

Description

This command returns the rise time result value for marker search. You can get the data even if :CALCulate:TRACe{Tr}:TTIMe:STATe is off.

Parameter

Description	Value of transition time result.
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TTIM:DATA?"
20 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:TTIMe:STATe

:CALCulate:TRACe{Tr}:TTIMe:THReshold

Equivalent GUI

No equivalent GUI is available.

:CALCulate:TRACe{Tr}:TTIMe:STATe

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TTIMe:STATe {ON|OFF|1|0}

:CALCulate:TRACe{Tr}:TTIMe:STATe?

Description

This command displays the rise time marker.

Parameter

Description	Status of display of rise time marker.
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0} <newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TTIM:STAT ON"
20 OUTPUT 717;":CALC:TRAC1:TTIM:STAT?"
30 ENTER 717;A
```

Related Commands

:CALCulate:TRACe{Tr}:TTIMe:DATA

:CALCulate:TRACe{Tr}:TTIMe:THReshold

Equivalent GUI

Marker Search > **Rise Time (10-90%)** or **Rise Time (20-80%)**

:CALCulate:TRACe{Tr}:TTIMe:THReshold

Type

Command/Query

Syntax

:CALCulate:TRACe{Tr}:TTIMe:THReshold {T1_9|T2_8}

:CALCulate:TRACe{Tr}:TTIMe:THReshold?

Description

This command sets the rise time threshold for the rise time in the marker search function.

Parameter

Description	Transition time function rise time threshold
Data Type	Character string type (String)
Range	T1_9 : 10-90% T2_8 : 20-80%
Preset Value	T1_9
Unit	-
Resolution	-

Query Response

{T1_9|T2_8}<newline><^END>

Examples

```
10 OUTPUT 717;":CALC:TRAC1:TTIM:THR T1_9"
20 OUTPUT 717;":CALC:TRAC1:TTIM:THR?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:TTIMe:DATA

:CALCulate:TRACe{Tr}:TTIMe:STATe

Equivalent GUI

Marker Search > **Rise Time (10-90%)** or **Rise Time (20-80%)**

DISPLAY

:DISPlay:ATRaces:SCALe:AUTO

Type

Command Only

Syntax

:DISPlay:ATRaces:SCALe:AUTO

Description

This command executes auto scale for all traces.

Examples

```
10 OUTPUT 717;":DISP:ATR:SCAL:AUTO"
```

Related Commands

:DISPlay:TRACe{Tr}:X:SCALe:AUTO

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Equivalent GUI

Auto Scale > **All Traces** OR

Right-click on the graph plot > **Auto Scale All**

:DISPlay:ATRaces:SCALe:RPOSition

Type

Command/Query

Syntax

:DISPlay:ATRaces:SCALe:RPOSition {LEFT|CENTer}

:DISPlay:ATRaces:SCALe:RPOSition?

Description

This command sets x-axis reference position for the time domain measurement.

Parameter

Description	Sets x-axis reference position for the time domain measurement
Data Type	Character string type (String)
Range	LEFT: The reference position is left edge CENTer: The reference position is center
Preset Value	LEFT
Unit	-
Resolution	-

Query Response

{LEFT|CENTer}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:ATR:SCAL:RPOS LEFT"
20 OUTPUT 717;":DISP:ATR:SCAL:RPOS?"
30 ENTER 717;A$
```

Equivalent GUI

TDR/TDT > or (Under **Horizontal**)

:DISPlay:IMAGe

Type

Command/Query

Syntax

:DISPlay:IMAGe {NORMal|INVert}

:DISPlay:IMAGe?

Description

This command changes the back ground color of screen.

Parameter

Description	Back ground color
Data Type	Character string type (String)
Range	NORMal: Black background color INVert: While background color
Preset Value	NORMal
Unit	-
Resolution	-

Query Response

{NORMal|INVert}<newline><^END>

Examples

```
10 OUTPUT 717;":DISPlay:IMAGe NORMal"
20 OUTPUT 717;":DISPlay:IMAGe?"
30 ENTER 717;A$
```

Related Commands

:MMEMory:STORe:IMAGe

Equivalent GUI

File > **Invert Color**

:DISPlay:TRACe{Tr}:DMEMory:TYPE

Type

Command/Query

Syntax

:DISPlay:TRACe{Tr}:DMEMory:TYPE {OFF|DATA|MEMory|DMEMory}

:DISPlay:TRACe{Tr}:DMEMory:TYPE?

Description

This command sets type of data/memory display.

Parameter

Description	Sets the display to off, data type, memory type or data and memory type
Data Type	Character string type (String)
Range	OFF: Nothing is displayed on the graph plot. DATA: Data only is displayed on the graph plot. MEMory: Memory only is displayed on the graph plot. DMEMory: Data and Memory are displayed on the graph plot.
Preset Value	DATA
Unit	-
Resolution	-

Query Response

{OFF|DATA|MEMory|DMEMory}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:TRAC1:DMEM:TYPE OFF"
20 OUTPUT 717;":DISP:TRAC1:DMEM:TYPE?"
30 ENTER 717;A$
```

Related Commands

:CALCulate:TRACe{Tr}:MATH:MEMorize

Equivalent GUI

Data Mem > Select **Off**, **Data**, **Memory** or **Data & Memory**.

:DISPlay:TRACe{Tr}:X:SCALe:AUTO

Type

Command Only

Syntax

:DISPlay:TRACe{Tr}:X:SCALe:AUTO

Description

This command executes x-axis auto scaling.

Examples

```
10 OUTPUT 717;":DISP:TRAC1:X:SCAL:AUTO"
```

Related Commands

:DISPlay:ATRaces:SCALe:AUTO

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Equivalent GUI

Auto Scale > **X** OR

Right-click on the graph plot > **Auto Scale X**

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision

Type

Command/Query

Syntax

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision <numeric>

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision?

Description

This command sets value of x-axis scale per division.

Parameter

Description	X-axis scale per division
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	2n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:TRAC1:X:SCAL:PDIV 1e-9"
20 OUTPUT 717;":DISP:TRAC1:X:SCAL:PDIV?"
30 ENTER 717;A
```

Related Commands

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision

Equivalent GUI

TDR/TDT > input box under the left knob (Under **Horizontal**)

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel

Type

Command/Query

Syntax

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel <numeric>

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel?

Description

This command sets value of x-axis reference line.

Parameter

Description	Value at X-axis reference line
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	-10n
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:TRAC1:X:SCAL:RLEV 20e-9"
20 OUTPUT 717;":DISP:TRAC1:X:SCAL:RLEV?"
30 ENTER 717;A
```

Related Commands

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel

Equivalent GUI

TDR/TDT > input box under the right knob (Under **Horizontal**)

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Type

Command Only

Syntax

:DISPlay:TRACe{Tr}:Y:SCALe:AUTO

Description

This command executes y-axis auto scaling.

Examples

```
10 OUTPUT 717;":DISP:TRAC1:Y:SCAL:AUTO"
```

Related Commands

:DISPlay:ATRaces:SCALe:AUTO

:DISPlay:TRACe{Tr}:X:SCALe:AUTO

Equivalent GUI

Auto Scale > **Y** OR

Right-click on the graph plot > **Auto Scale Y**

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision

Type

Command/Query

Syntax

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision <numeric>

:DISPlay:TRACe{Tr}:Y:SCALe:PDIVision?

Description

This command sets value of Y-axis scale per division.

Parameter

Description	Y-axis scale per division value.
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	20
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:TRAC1:Y:SCAL:PDIV 20"
20 OUTPUT 717;":DISP:TRAC1:Y:SCAL:PDIV?"
30 ENTER 717;A
```

Related Commands

:DISPlay:TRACe{Tr}:X:SCALe:PDIVision

Equivalent GUI

TDR/TDT > input box under the left knob (Under **Vertical**)

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel

Type

Command/Query

Syntax

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel <numeric>

:DISPlay:TRACe{Tr}:Y:SCALe:RLEVel?

Description

This command sets value of y-axis reference line.

Parameter

Description	Value at Y-axis reference line
Data Type	Double precision floating point type (Double)
Range	-
Preset Value	100
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":DISP:TRAC1:Y:SCAL:RLEV 0.01"
20 OUTPUT 717;":DISP:TRAC1:Y:SCAL:RLEV?"
30 ENTER 717;A
```

Related Commands

:DISPlay:TRACe{Tr}:X:SCALe:RLEVel

Equivalent GUI

TDR/TDT > input box under the right knob (Under **Vertical**)

IEEE

*CLS

Type

Command Only

Syntax

*CLS

Description

This command clears the status.

Examples

```
10 OUTPUT 717;"*CLS"
```

Equivalent GUI

No equivalent GUI is available.

*IDN

Type

Query Only

Syntax

*IDN?

Description

This command returns the product information and revision number of E5071C-TDR application.

Parameter

Description	Gets model information
Data Type	Character string type (String)
Range	-
Preset Value	"Agilent Technologies,ENA-TDR,--------,<revision number>"
Unit	-
Resolution	-

Query Response

{String}<newline><^END>

Examples

```
10 OUTPUT 717;"*IDN?"
20 ENTER 717;A$
```

Equivalent GUI

E5071C OPT. TDR logo at the upper left corner on GUI.

*OPC

Type

Query Only

Syntax

*OPC?

Description

This command detects the operation complete.

Parameter

Description	Operation complete
Data Type	Long integer type (Long)
Range	1 is returned when all pending operations are complete
Preset Value	-
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;"*OPC?"
20 ENTER 717;A
```

Equivalent GUI

No equivalent GUI is available.

MMEMORY

:MMEMory:LOAD:EYE:BPATtern

Type

Command Only

Syntax

:MMEMory:LOAD:EYE:BPATtern <string>

Description

This command loads the specified user bit pattern file. The extension of file should be .txt. The bit pattern editing is not available through the command.

Parameter

Description	File name of user bit pattern (.txt)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":MMEM:LOAD:EYE:BPAT ""D:\TDR\userbit.txt"""
20 OUTPUT 717;":CALC:EYE:INP:BPAT:TYPE USER"
```

Related Commands

:CALCulate:EYE:INPut:BPATtern:TYPE

Equivalent GUI

Eye/Mask > **User Pattern** (Under **Bit Pattern**) > **Load**

:MMEMory:LOAD:EYE:MASK

Type

Command Only

Syntax

:MMEMory:LOAD:EYE:MASK <string>

Description

This command loads eye-mask file. The format of eye mask file should be the same as ones of Infiniium DCA (86100C). The extension of file should be .msk. The MASK pattern editing is not available through the command.

Parameter

Description	File name of eye mask (.msk)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

See the example in :CALCulate:EYE:MASK:STATe

Related Commands

:CALCulate:EYE:MASK:STATe

:CALCulate:EYE:EXECute

:CALCulate:EYE:STATe

Equivalent GUI

Eye Mask > **Mask Pattern** (under **Eye Control**) > **File Tab** > **Load**

:MMEMory:STORe:FDATa

Type

Command Only

Syntax

:MMEMory:STORe:FDATa <string>

Description

This command stores measurement trace data.

Parameter

Description	File name of data file (.csv)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":MMEMory:STORe:FDATa ""D:\TDR\dataresult.csv"""""
```

Related Commands

:MMEMory:STORe:SNP

Equivalent GUI

File > **Save Trace Data**

:MMEMory:STORe:IMAGe

Type

Command Only

Syntax

:MMEMory:STORe:IMAGe <string>

Description

This command stores a display image into a bmp/png format file.

Parameter

Description	File name of image (.bmp or .png)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":MMEM:STOR:IMAG ""D:\TDR\figure1.png"""""
```

Related Commands

:DISPlay:IMAGe

Equivalent GUI

File > **Save Image**

:MMEMory:STORe:SNP

Type

Command Only

Syntax

:MMEMory:STORe:SNP <string>

Description

This command stores the touchstone format file. The file extension should be changed according to the setting of DUT topology.

File Extension	DUT Topology
.s1p	Single Ended 1
.s2p	Single Ended 2, Differential 1
.s4p	Single Ended 4, Differential 2

Parameter

Description	File name of touchstone file (.snp)
Data Type	Character string type (String)
Range	-
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":MMEM:STOR:SNP ""D:\TDR\touchstone.s2p"""""
```

Related Commands

:MMEMory:STORe:FDATa

Equivalent GUI

File > **Save Touchstone**

SENSE

:SENSe:AVERage:CLEar

Type

Command Only

Syntax

:SENSe:AVERage:CLEar

Description

This command clears average count.

Examples

10 OUTPUT 717;":SENS:AVER:CLE"

Related Commands

:SENSe:AVERage:COUNt

:SENSe:AVERage:STATe

Equivalent GUI

No equivalent GUI is available.

:SENSe:AVERage:COUNt

Type

Command/Query

Syntax

:SENSe:AVERage:COUNt <numeric>

:SENSe:AVERage:COUNt?

Description

This command sets averaging factor value.

Parameter

Description	Number of Averaging factor.
Data Type	Long integer type (Long)
Range	1 to 999
Preset Value	16
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:AVER:COUN 16"
20 OUTPUT 717;":SENS:AVER:STAT ON"
```

Related Commands

:SENSe:AVERage:STATe

:SENSe:AVERage:CLEar

Equivalent GUI

Setup > **Averaging** > Type in the average factor in to entry box beside **Averaging**

:SENSe:AVERage:STATe

Type

Command/Query

Syntax

:SENSe:AVERage:STATe {ON|OFF|1|0}

:SENSe:AVERage:STATe?

Description

This command activates the averaging.

Parameter

Description	Status of averaging on or off
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:AVER:COUN 16"
20 OUTPUT 717;":SENS:AVER:STAT ON"
```

Related Commands

:SENSe:AVERage:CLEar

:SENSe:AVERage:COUNt

:TRIGger:AVERage

Equivalent GUI

Setup > **Averaging**

:SENSe:CORRection:COLLection:ADVanced:IMMediate

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:ADVanced:IMMediate

Description

This command clears the calibration in the basic mode and allows you to use the Advance Calibration Mode. Before you perform the calibration, such as TRL calibration, for channel 1 (TDR), this command should be executed.

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:ADV:IMM "
```

Equivalent GUI

Advanced Mode > check box (on) of **Advanced Calibration Methods**

:SENSe:CORRection:COLLection:DLComp:LOAD

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:DLComp:LOAD

Description

This command executes load measurement, as a part of Loss Compensation sequence.

Parameter

Description	Port number where the load measurement is executed.
Data Type	Long integer type (Long)
Range	1 to 4
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:DLC:LOAD 1 "
20 OUTPUT 717; ":SENS:CORR:COLL:DLC:SAVE"
```

Related Commands

:SENSe:CORRection:COLLection:DLComp:OPEN

:SENSe:CORRection:COLLection:DLComp:THRU

:SENSe:CORRection:COLLection:DLComp:SAVE

Equivalent GUI

Setup > **Deskew & Loss Compensation** (under **Basic**) > **Measure** > **Port 1** and/or **Port 2** and/or **Port 3** and/or **Port 4**

:SENSe:CORRection:COLLection:DLComp:OPEN

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:DLComp:OPEN

Description

This command executes open measurement, as a part of Loss Compensation sequence.

Parameter

Description	Port number where the open measurement is executed.
Data Type	Long integer type (Long)
Range	1 to 4
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:DLC:OPEN 1"
20 OUTPUT 717; ":SENS:CORR:COLL:DLC:SAVE"
```

Related Commands

:SENSe:CORRection:COLLection:DLComp:LOAD

:SENSe:CORRection:COLLection:DLComp:THRU

:SENSe:CORRection:COLLection:DLComp:SAVE

Equivalent GUI

Setup > **Deskew & Loss Compensation** (under **Basic**) > **Measure**

NOTE This applies to Single-Ended 1-Port topology only.

:SENSe:CORRection:COLLection:DLComp:SAVE

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:DLComp:SAVE

Description

This command saves the result of Loss Compensation sequence.

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:DLC:SAVE"
```

Related Commands

:SENSe:CORRection:COLLection:DLComp:OPEN

:SENSe:CORRection:COLLection:DLComp:LOAD

:SENSe:CORRection:COLLection:DLComp:THRU

Equivalent GUI

Setup > **Deskew & Loss Compensation** (under **Basic**) > **Measure** > **Port 1** and/or **Port 2** and/or **Port 3** and/or **Port 4** > **Apply**

:SENSe:CORRection:COLLection:DLComp:THRU

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:DLComp:THRU

Description

This command executes a thru measurement, as a part of Loss Compensation sequence.

Parameter

Description	Port number where the thru measurement is executed.
Data Type	Character string type (String)
Range	TH12: Thru between ports 1 and 2. TH13: Thru between ports 1 and 3. TH24: Thru between ports 2 and 4.
Preset Value	-
Unit	-
Resolution	-

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:DLC:THRU TH12"
20 OUTPUT 717; ":SENS:CORR:COLL:DLC:SAVE"
```

Related Commands

:SENSe:CORRection:COLLection:DLComp:OPEN

:SENSe:CORRection:COLLection:DLComp:LOAD

:SENSe:CORRection:COLLection:DLComp:SAVE

Equivalent GUI

Setup > **Deskew & Loss Compensation** (under **Basic**) > **Measure**

NOTE This applies to all other topologies except for Single-Ended 1-Port.

:SENSe:CORRection:COLLection:ECAL:FCOMp

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:ECAL:FCOMp

Description

This command executes fixture compensation after ECal

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:ECAL:FCOM"
```

Equivalent GUI

Setup > **ECal** (under **Basic**) > **Calibrate** > **Fixture Comp**

:SENSe:CORRection:COLLection:ECAL:IMMediate

Type

Command Only

Syntax

:SENSe:CORRection:COLLection:ECAL:IMMediate

Description

This command executes full calibration using the ECal module

Examples

```
10 OUTPUT 717;":SENS:CORR:COLL:ECAL:IMM"
```

Equivalent GUI

Setup > **ECal** (under **Basic**) > **Calibrate**

:SENSe:CORRection:DCONstant

Type

Command/Query

Syntax

:SENSe:CORRection:DCONstant <numeric>

:SENSe:CORRection:DCONstant?

Description

This command sets dielectric constant value.

Parameter

Description	Dielectric constant value
Data Type	Double precision floating point type (Double)
Range	10m to 100
Preset Value	1000m
Unit	-
Resolution	-

Query Response

{numeric} <newline><^END>

Examples

```
10 OUTPUT 717;":SENS:CORR:DCON 0.01"
20 OUTPUT 717;":SENS:CORR:DCON?"
30 ENTER 717;A
```

Related Commands

:SENSe:CORRection:RIMPedance

Equivalent GUI

Setup > **Dielectric Const.** (under **More Functions**)

:SENSe:CORRection:EXTension:AUTO:IMMediate

Type

Command Only

Syntax

:SENSe:CORRection:EXTension:AUTO:IMMediate

Description

This command executes deskew (auto port extension).

Examples

```
10 OUTPUT 717;":SENS:CORR:EXT:AUTO:IMM"
```

Related Commands

:SENSe:CORRection:EXTension:AUTO:PORT{pt}

:SENSe:CORRection:EXTension:AUTO:STANdard

Equivalent GUI

Setup > **Deskew** (under **Basic**).

:SENSe:CORRection:EXTension:AUTO:PORT{pt}

Type

Command/Query

Syntax

:SENSe:CORRection:EXTension:AUTO:PORT{[1]-4} {ON|OFF|1|0}

:SENSe:CORRection:EXTension:AUTO:PORT{[1]-4}?

Description

This command sets ON/OFF or gets the status of auto port extension.

Parameter

Description	Status of auto port extension
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	ON
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:CORR:EXT:AUTO:PORT1 ON"
20 OUTPUT 717;":SENS:CORR:EXT:AUTO:PORT1?"
30 ENTER 717;A
```

Related Commands

:SENSe:CORRection:EXTension:AUTO:IMMediate

:SENSe:CORRection:EXTension:AUTO:STANdard

Equivalent GUI

Setup > **Deskew** (under **Basic**) > **Option** > **Port1** and/or **Port2** and/or **Port3** and/or **Port4** OR

Setup > **Deskew & Loss Compensation** (under **Basic**) > **Option** > **Port1** and/or **Port2** and/or **Port3** and/or **Port4**

:SENSe:CORRection:EXTension:AUTO:STANdard

Type

Command/Query

Syntax

:SENSe:CORRection:EXTension:AUTO:STANdard {OPEN|SHORt}

:SENSe:CORRection:EXTension:AUTO:STANdard?

Description

This command sets the standard for auto port extension.

Parameter

Description	Standard for Deskew
Data Type	Character string type (String)
Range	OPEN: Open termination SHORt: Short termination
Preset Value	OPEN
Unit	-
Resolution	-

Query Response

{OPEN|SHORt}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:CORR:EXT:AUTO:STAN OPEN"
20 OUTPUT 717;":SENS:CORR:EXT:AUTO:STAN?"
30 ENTER 717;A$
```

Related Commands

:SENSe:CORRection:EXTension:AUTO:PORT{pt}

:SENSe:CORRection:EXTension:AUTO:IMMediate

Equivalent GUI

Setup > **Deskew** (under **Basic**) > **Option** > select **Standard Type**

Setup > **Deskew & Loss Compensation** (under **Basic**) > **Option** > select **Standard Type**

:SENSe:CORRection:RIMPedance

Type

Command/Query

Syntax

:SENSe:CORRection:RIMPedance <numeric>

:SENSe:CORRection:RIMPedance?

Description

This command sets reference impedance value.

Parameter

Description	Reference impedance value
Data Type	Double precision floating point type (Double)
Range	1m to 10M
Preset Value	50
Unit	-
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:CORR:RIMP 0.001"
20 OUTPUT 717;":SENS:CORR:RIMP?"
30 ENTER 717;A
```

Related Commands

:SENSe:CORRection:DCONstant

Equivalent GUI

Setup > **Ref. Z** (under **More Functions**)

:SENSe:DLENgth:AUTO:IMMediate

Type

Command Only

Syntax

:SENSe:DLENgth:AUTO:IMMediate

Description

This command executes auto DUT length setting

Examples

```
10 OUTPUT 717;":SENS:DLEN:AUTO:IMM"
```

Related Commands

:SENSe:DLENgth:DATA

Equivalent GUI

TDR/TDT > **DUT Length:** **Auto** (under **Basic**) > **Measure**

:SENSe:DLENgth:DATA

Type

Command/Query

Syntax

:SENSe:DLENgth:DATA <numeric>

:SENSe:DLENgth:DATA?

Description

This command sets DUT length value.

Parameter

Description	DUT length value
Data Type	Double precision floating point type (Double)
Range	6.26n to 416n
Preset Value	6.26n
Unit	sec
Resolution	-

Query Response

{numeric}<newline><^END>

Examples

```
10 OUTPUT 717;":SENS:DLEN:DATA 6.26e-9"
20 OUTPUT 717;":SENS:DLEN:DATA?"
30 ENTER 717;A
```

Related Commands

:SENSe:DLENgth:AUTO:IMMediate

Equivalent GUI

TDR/TDT > **DUT Length** (under **Basic**) > Enter the DUT length

SYSTEM

:SYSTem:ERRor

Type

Query Only

Syntax

:SYSTem:ERRor?

Description

This command gets error code & description

Parameter

Description	Error code
Data Type	Double precision floating point type (Double)
Range	- -
Preset Value	0 "No error"
Unit	- -
Resolution	- -

Query Response

{ - -} <newline><^END>

Examples

```
10 OUTPUT 717;":SYST:ERR?"
30 ENTER 717;A
```

Equivalent GUI

No equivalent GUI is available.

:SYSTem:PRESet

Type

Command Only

Syntax

:SYSTem:PRESet

Description

This command presets the E5071C setting. Calibration data will be lost.

Examples

```
10 OUTPUT 717;":SYST:PRES"
```

Equivalent GUI

Setup > **Preset** (under **Basic**)

TRIGGER

:TRIGger:AVERage

Type

Command/Query

Syntax

:TRIGger:AVERage {ON|OFF|1|0}

:TRIGger:AVERage?

Description

This command sets averaging trigger on/off. When averaging trigger is on, one trigger makes one averaging measurement. For example, if the averaging factor is set at 16, one trigger makes a measurement for 16 times.

Parameter

Description	Status of Averaging trigger
Data Type	Boolean type (Boolean)
Range	ON|OFF|1|0
Preset Value	OFF
Unit	-
Resolution	-

Query Response

{1|0}<newline><^END>

Examples

```
10 OUTPUT 717;":TRIG:MODE HOLD"
20 OUTPUT 717;":SENS:AVER:COUN 16"
30 OUTPUT 717;":SENS:AVER:STAT ON"
40 OUTPUT 717;":TRIG:AVER ON"
50 OUTPUT 717;":TRIG:SING"
```

Related Commands

:SENSe:AVERage:COUNt

:SENSe:AVERage:STATe

:TRIGger:SINGle

Equivalent GUI

No equivalent GUI is available.

:TRIGger:MODE

Type

Command/Query

Syntax

:TRIGger:MODE {HOLD|SINGle|RUN}

:TRIGger:MODE?

Description

This command sets trigger mode.

Parameter

Description	Trigger mode
Data Type	Character string type (String)
Range	HOLD: Trigger is on hold until the conditions are meet, the trigger event starts. SINGle: Trigger event is run once. RUN: Trigger event is running continuous.
Preset Value	RUN
Unit	-
Resolution	-

Query Response

{HOLD|SINGle|RUN}<newline><^END>

Examples

```
10 OUTPUT 717;":TRIG:MODE HOLD"
20 OUTPUT 717;":TRIG:MODE?"
30 ENTER 717;A$
```

Related Commands

:TRIGger:SINGle

Equivalent GUI

Stop Single or **Run**

:TRIGger:SINGle

Type

Command Only

Syntax

:TRIGger:SINGle

Description

This command executes single trigger.

Examples

```
10 OUTPUT 717;":TRIG:SING"
```

Related Commands

:TRIGger:MODE

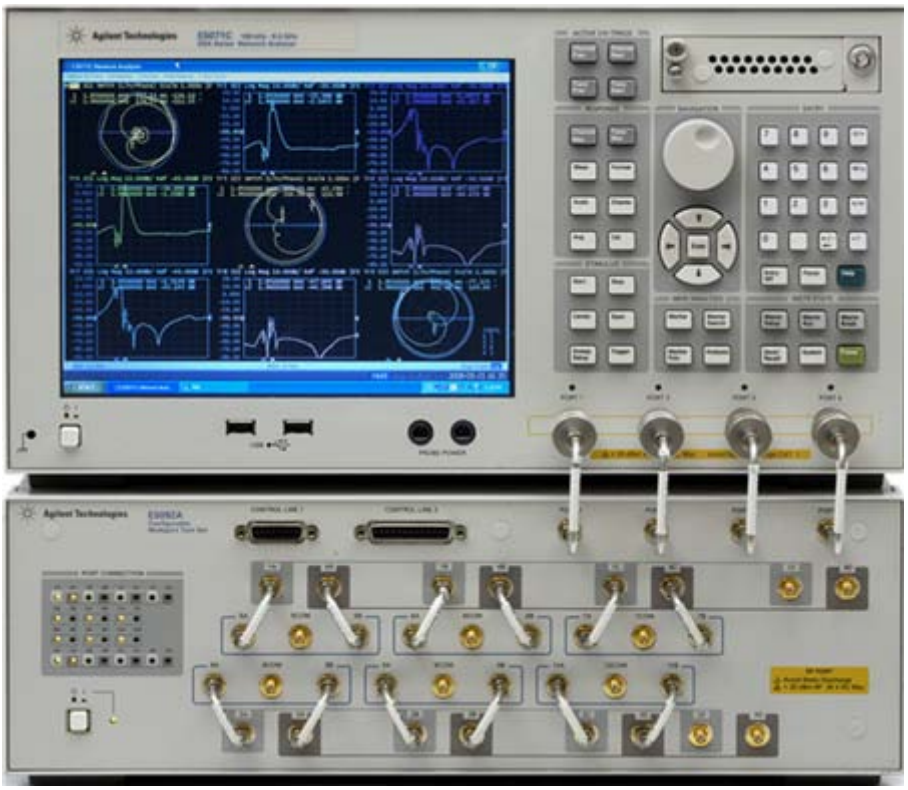
Equivalent GUI

No equivalent GUI is available.

Controlling Multiport Test Set

Controlling Multiport Test Set Overview

This section provides information about controlling multiport test set (E5092A/E5091A) with E5071C.



- Connecting E5071C and Multiport Test Set
- Setting Multiport Test Set
- Performing Measurement
- Controlling Multiport Test Set by Programming

Connecting E5071C and Multiport Test Set

- Connecting E5071C (Option 440/445/460/465/480/485) and E5092A
- Connecting E5071C (Option 4D5/4K5) and E5092A
- Connecting E5071C (Option 440/445/460/465/480/485) and E5091A
- Connecting Two Multiport Test Sets
- Recommended Wrench

Other topics about Controlling Multiport Test Set

Connecting E5071C (Option 440/445/460/465/480/485) and E5092A

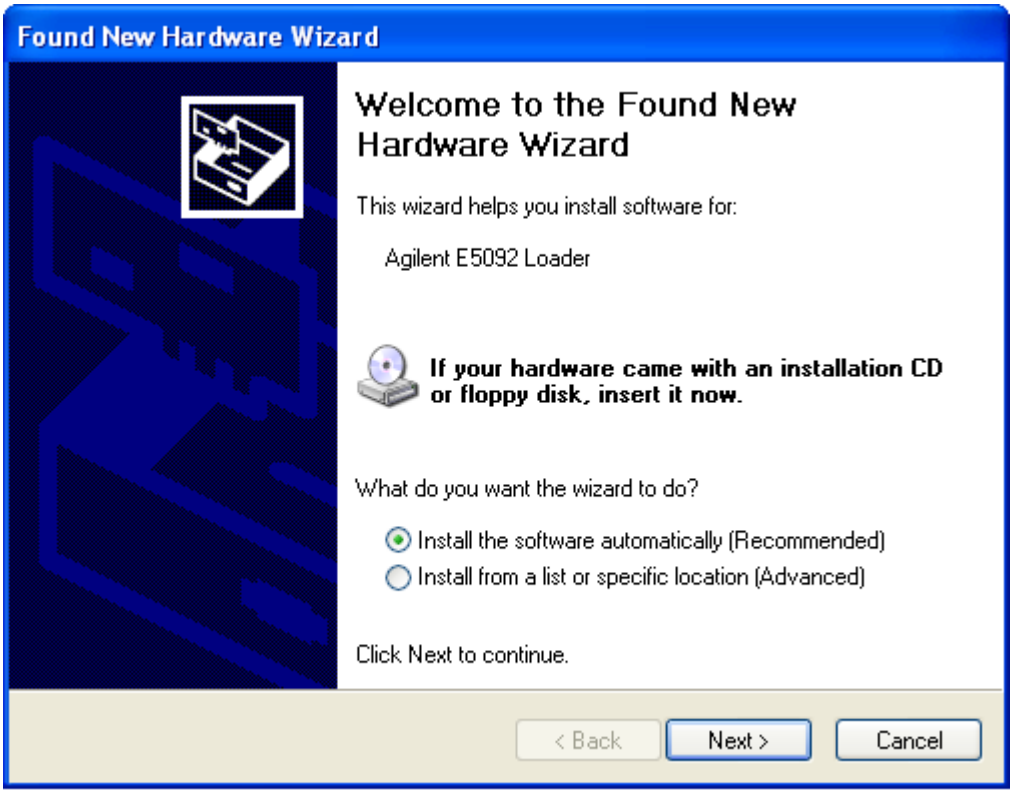
Required Devices

The devices required to connect the E5071C (option 440/445/460/465/480/485) to the E5092A configurable multiport test set are listed below:

- E5071C (option 440/445/460/465/480/485)
- E5092A (option 020)
- Type-N to SMA coaxial adapter (included in the E5092A-08C for connection with the E5071C, Agilent part number: 1250-2879)
- SMA semi-rigid cable (included in the E5092A-08C for connection with the E5071C, Agilent part number: E5092-61652)
- USB cable (supplied with the E5092A, Agilent part number: 8121-1695)
    - For the rackmount usage, it is recommended to use semi-rigid cable (Agilent part number: E5092-61654, included in the E5092A-08C) for connection with the E5071C.

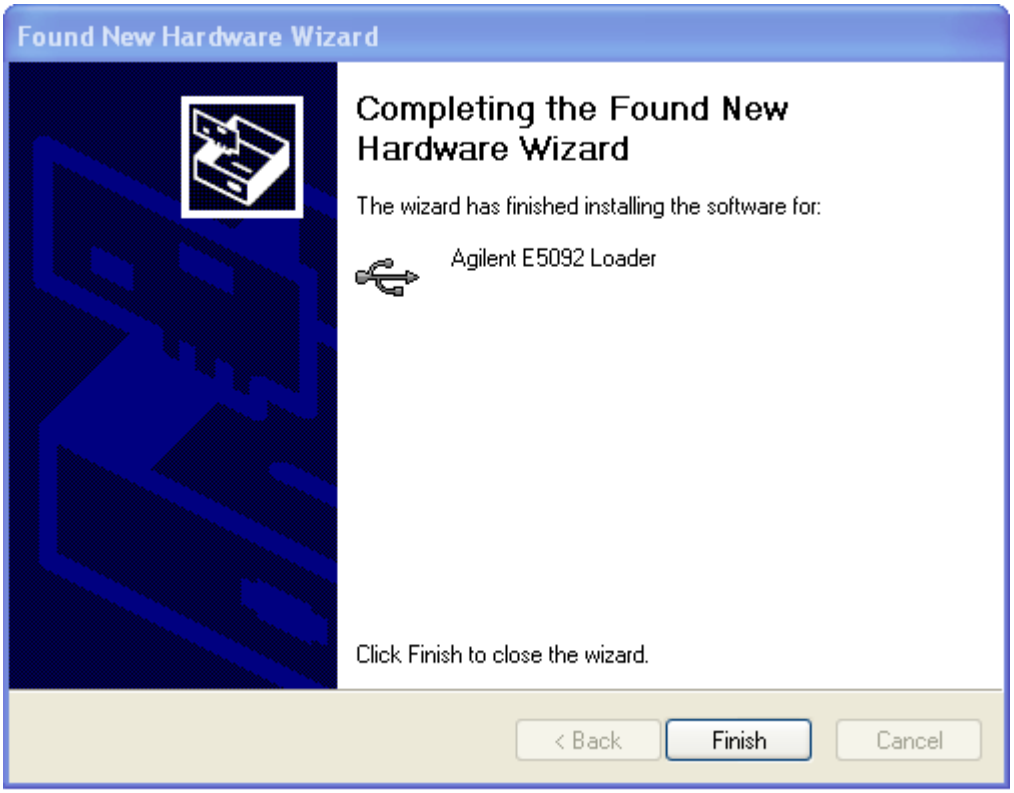
USB Cable Connections and Driver Installation

1. Connect the USB cable between one of USB ports on the E5071C rear panel and that of the E5092A.
2. Turn on the E5071C and the E5092A.
3. The **Found New Hardware Wizard** appears. Select install the software automatically, then click **Next**.



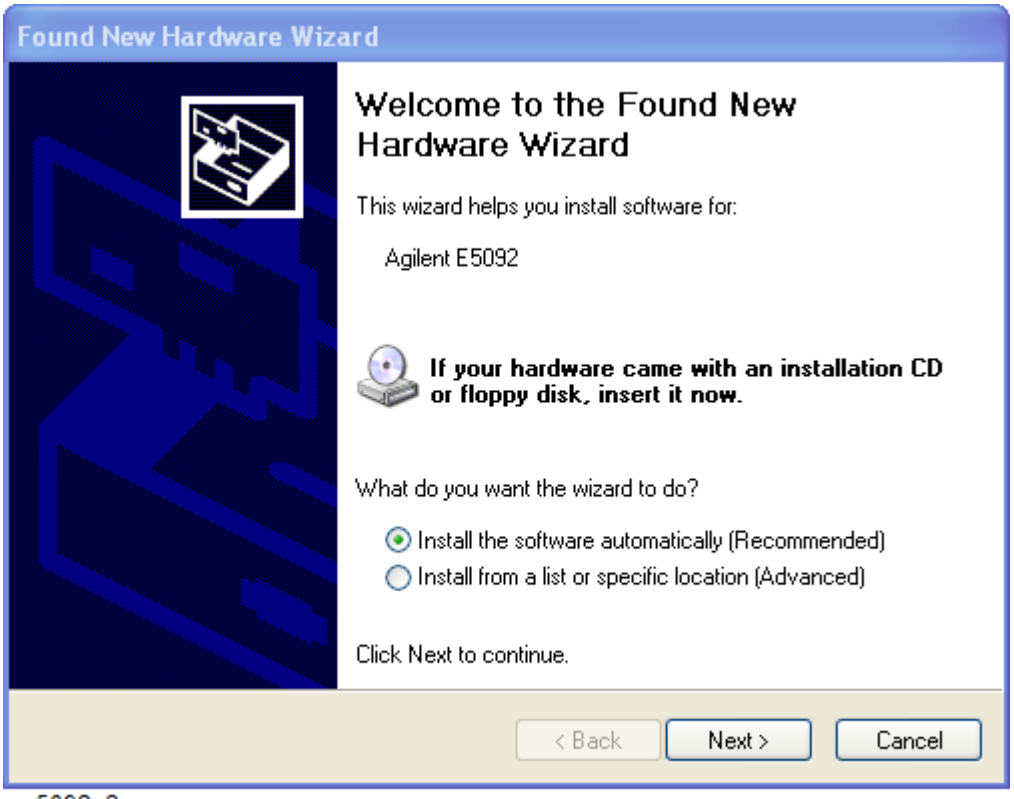
e5092a1

4. Click **Finish.**



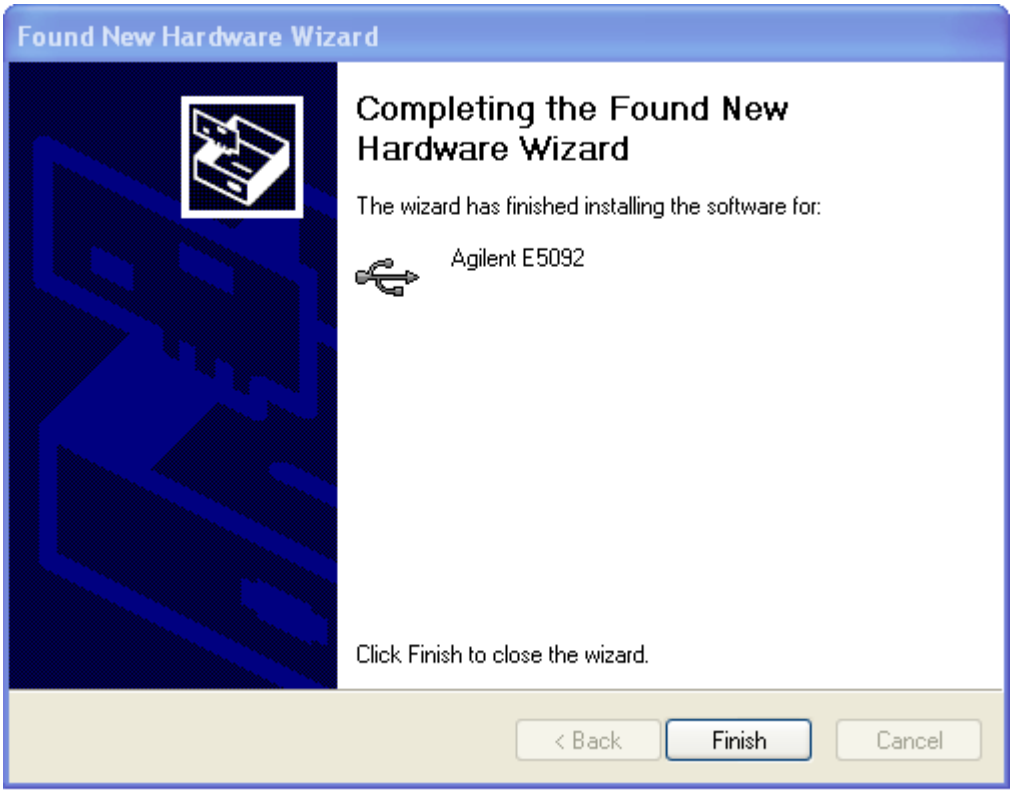
e5092a2

5. The **Found New Hardware Wizard** appears again. Select install the software automatically, then click **Next.**



e5092a3

6. Click **Finish.**

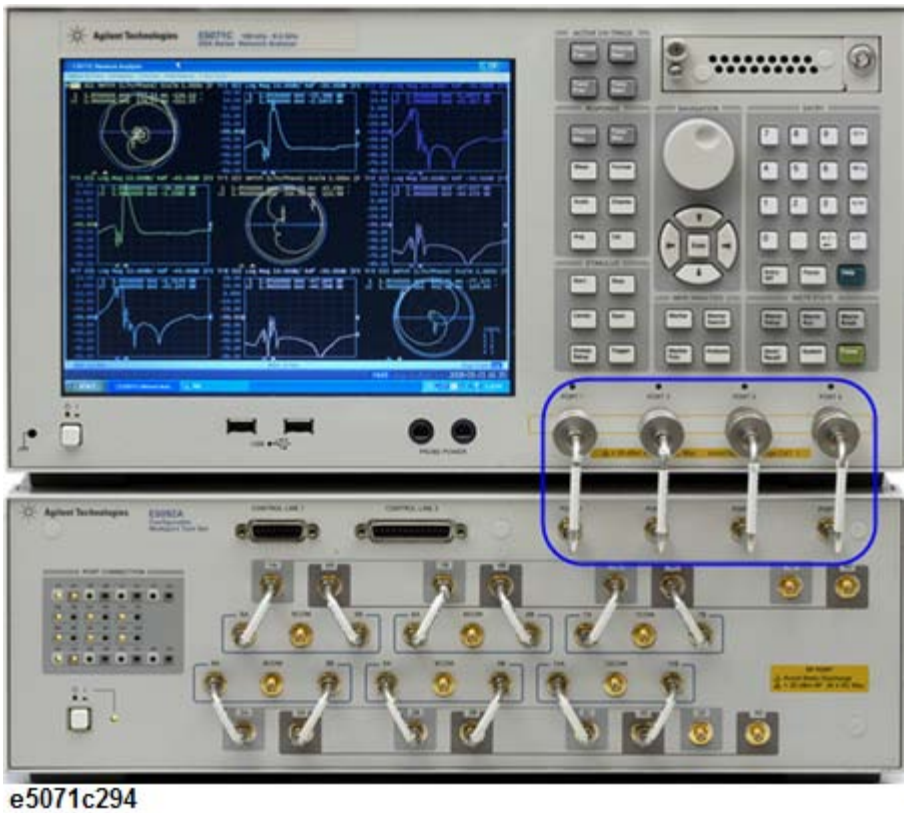


e5092a4

7. After the E5071C detects the E5092A, the LEDs that indicate the connected test ports remain on.
8. E5071C should be reset once the multiport test set driver is updated.

RF Cable Connection

1. Connect the type-N to SMA adapter on the test ports (PORT 1 to 4) of the E5071C.
2. Connect the SMA semi-rigid cables between the adapters on the E5071C and the test ports (PORT 1 to 4) of the multiport test set. Make the connection so that the numbers of the ENA ports and the test set ports match.



Connecting E5071C (Option 4D5/4K5) and E5092A

Required Devices

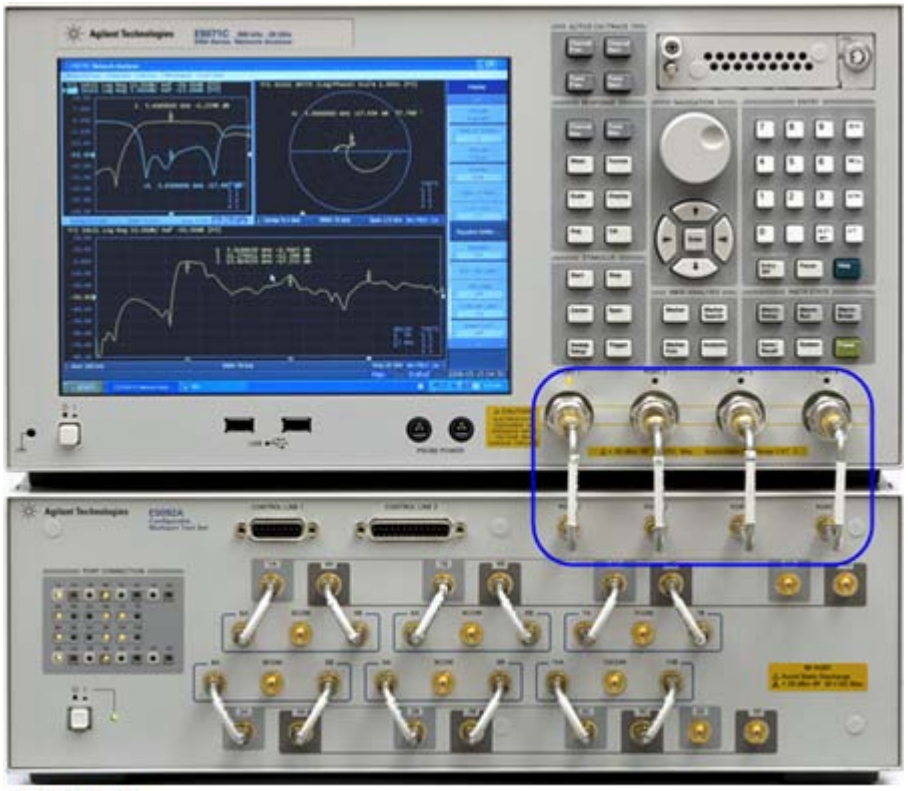
The devices required to connect the E5071C (option 4D5/4K5) to the E5092A configurable multiport test set are listed below:

- E5071C (option 4K5)
- E5092A (option 020)
- 3.5mm (female) to 3.5mm (female) coaxial adapter (included in the E5092A-20C for connection with the E5071C, Agilent part number: 85027-60005)
- Semi-rigid cable (included in the E5092A-20C for connection with the E5071C, Agilent part number: E5092-61653)
- USB cable (supplied with the E5092A, Agilent part number: 8121-1695)
    - For the rackmount usage, it is recommended to use semi-rigid cable (Agilent part number: E5092-61655, included in the E5092A-20C) for connection with the E5071C.

USB Cable Connections and Driver Installation

RF Cable Connection

1. Connect the coaxial adapter on the test ports (PORT 1 to 4) of the E5071C.
2. Connect the SMA semi-rigid cables between the adapters on the E5071C and the test ports (PORT 1 to 4) of the multiport test set. Make the connection so that the numbers of the ENA ports and the test set ports match.



e5071c295

Connecting E5071C (Option 440/445/460/465/480/485) and E5091A

Required Devices

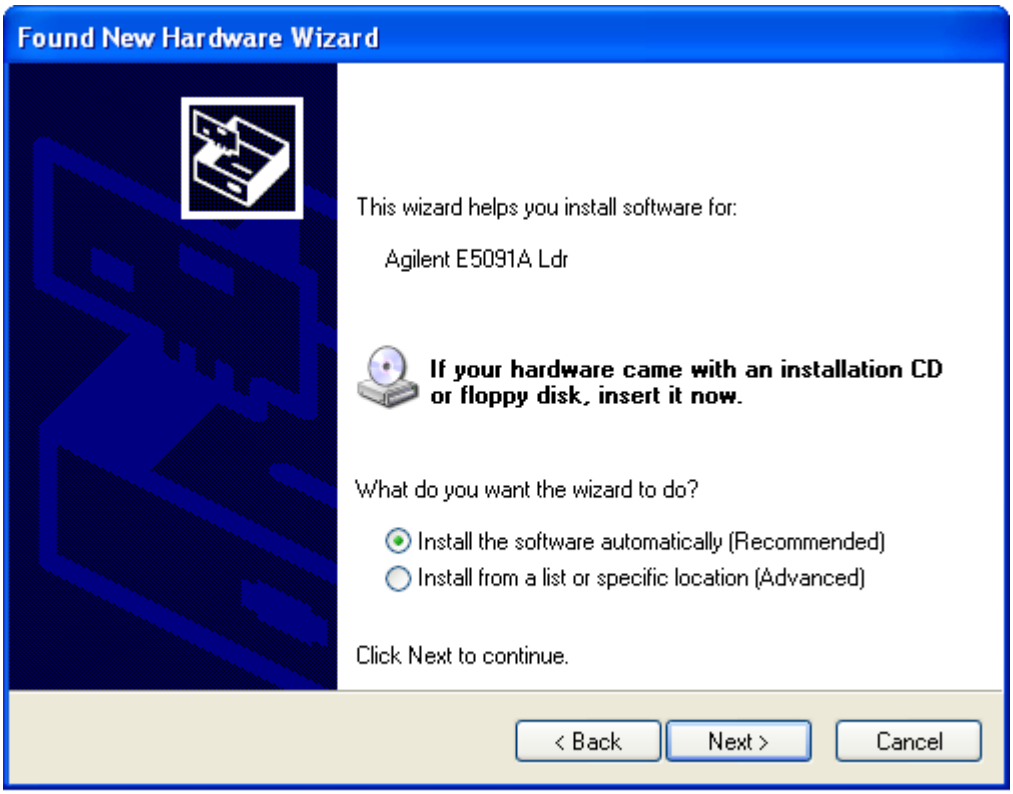
The devices required to connect the E5071C (option 440/445/460/465/480/485) to the E5091A multiport test set are listed below:

- E5071C (option 440/445/460/465/480/485)
- E5091A (option 009/016)
- Type-N to Type-N cable (supplied with the E5091A option 009 or 016 for connection with the E5071C, Agilent part number: 8120-4782)
- USB cable (supplied with the E5091A, Agilent part number: 8121-0770)

USB Cable connections and Driver Installation

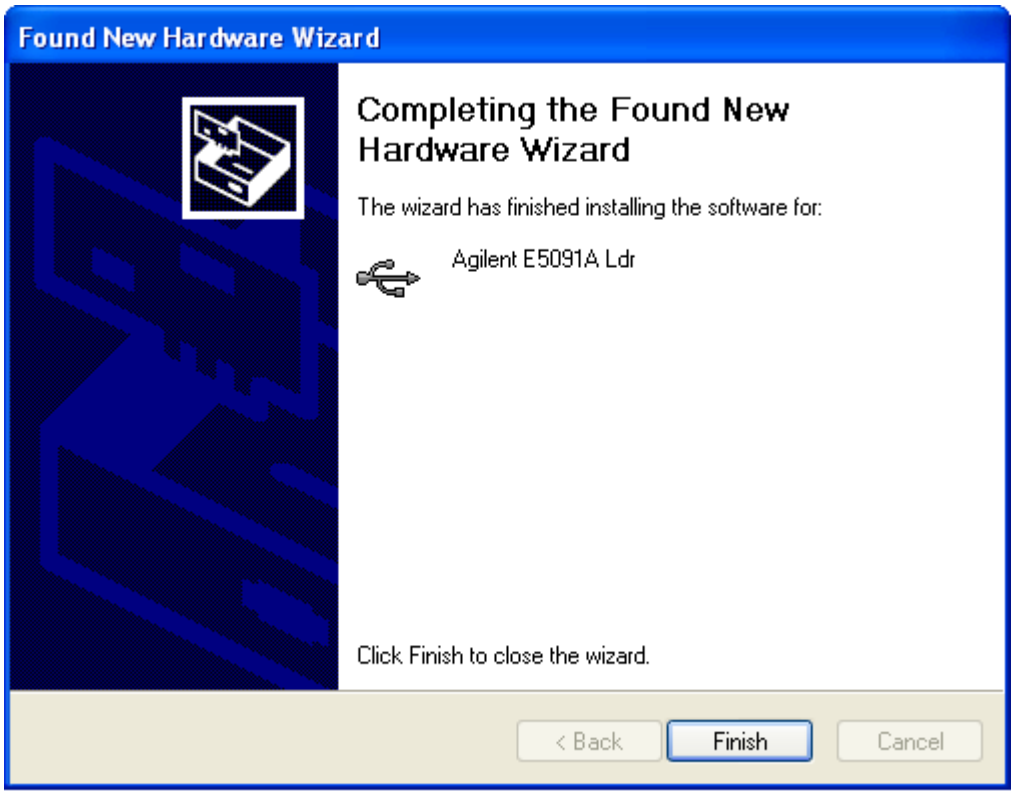
1. Connect the USB cable between one of USB ports on the E5071C rear panel and that of the E5091A.
2. Turn on the E5071C and the E5091A.
3. The **Found New Hardware Wizard** appears. Select **No, not this time**, then click **Next**.

4. Select Install the software automatically, then click **Next**.



e5071c174

5. Click **Finish**.

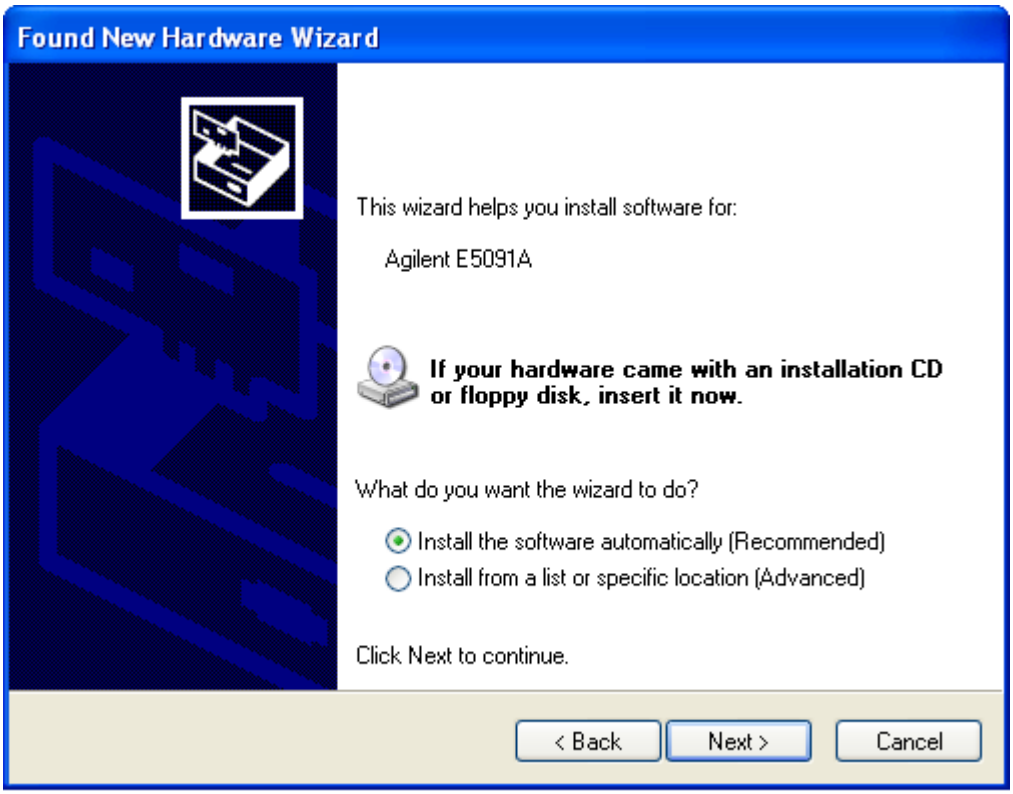


e5071c176

6. The **Found New Hardware Wizard** appears again. Select **No, not this time**, then click **Next**.

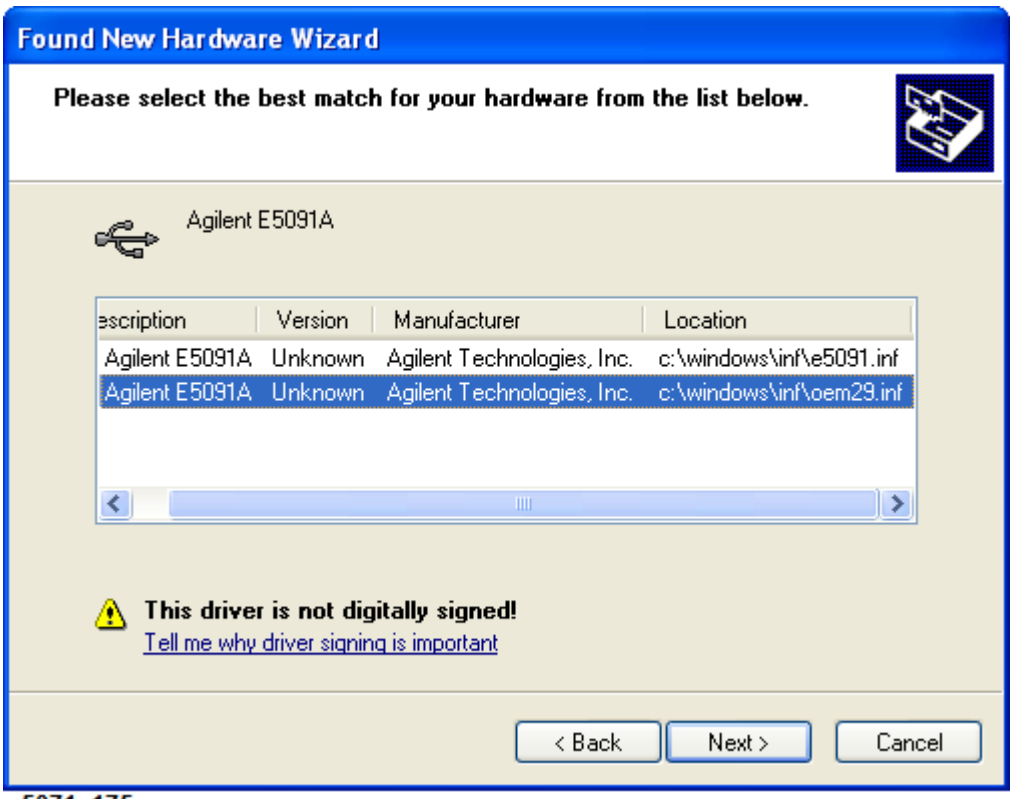
Found New Hardware Wizard
Welcome to the Found New Hardware Wizard
Windows will search for current and updated software by looking on your computer, on the hardware installation CD, or on the Windows Update Web site (with your permission).
Read our privacy policy
Can Windows connect to Windows Update to search for software?
Yes, this time only
Yes, now and every time I connect a device
No, not this time
Click Next to continue.
< Back
Next >
Cancel

7. Select Install the software automatically, then click **Next**.



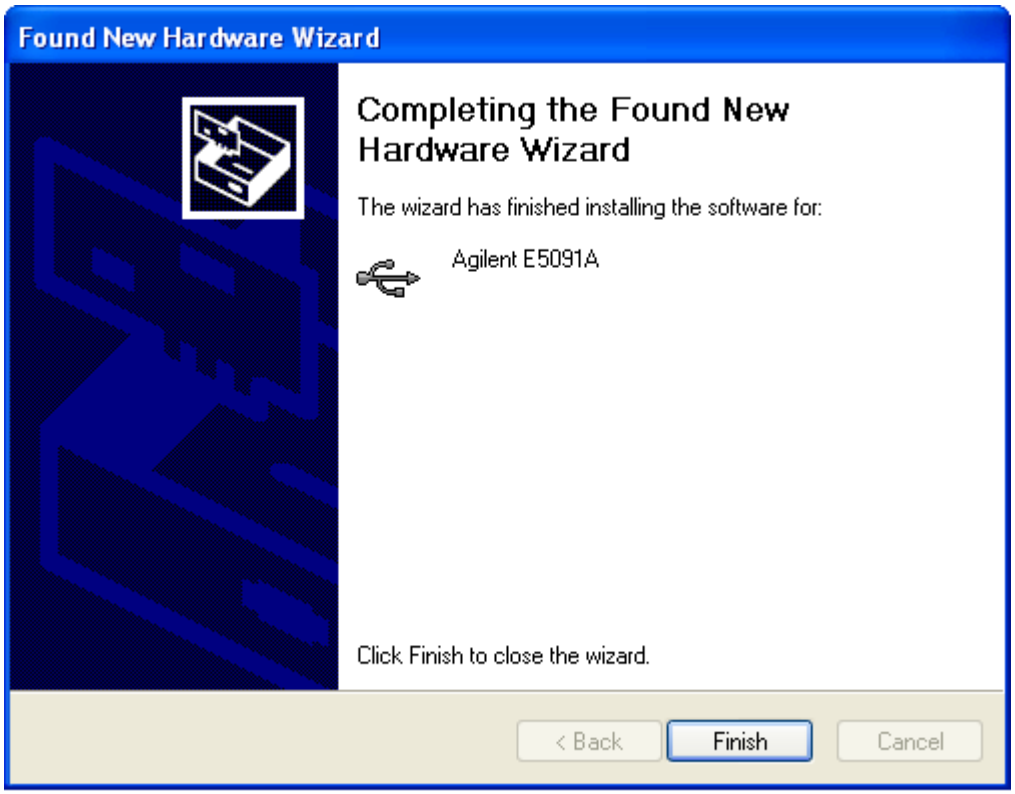
e5071c170

8. Select **Agilent E5091A** of Location **C:\windows\inf\oemxx.inf** (xx is numeric number e.g.: **oem29.inf**), then click **Next**.


Found New Hardware Wizard
Please select the best match for your hardware from the list below.
Agilent E5091A
escription
Version
Manufacturer
Location
Agilent E5091A
Unknown
Agilent Technologies, Inc.
c:\windows\inf\e5091.inf
Agilent E5091A
Unknown
Agilent Technologies, Inc.
c:\windows\inf\oem29.inf
This driver is not digitally signed!
Tell me why driver signing is important
< Back
Next >
Cancel

e5071c175

9. Click **Finish**.

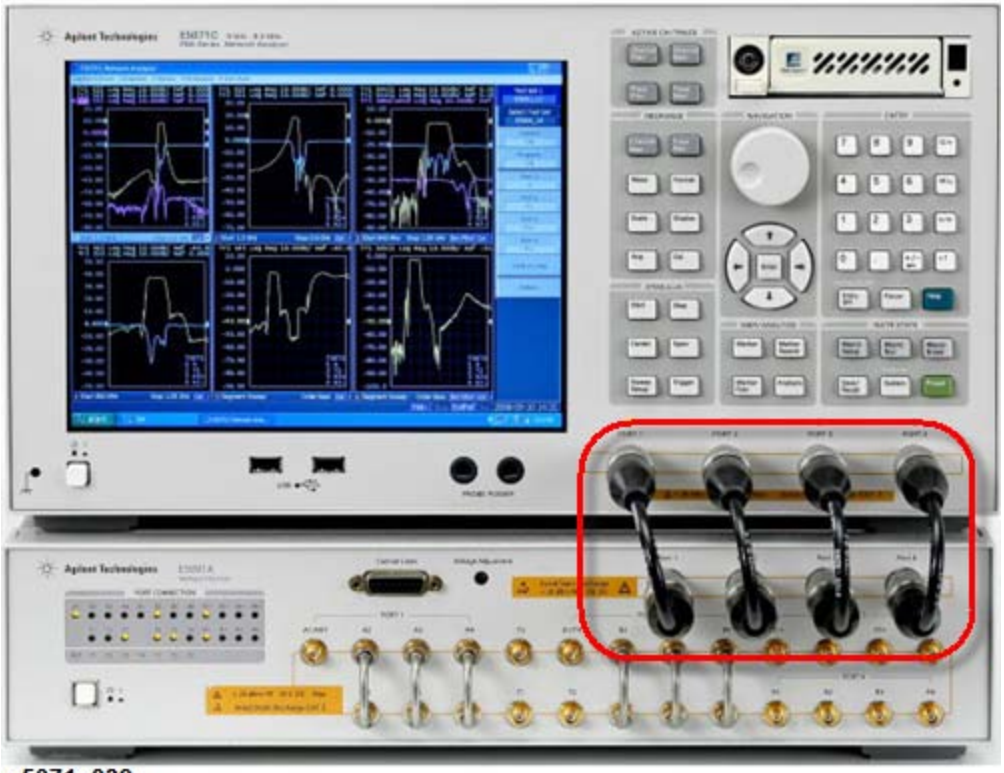


e5071c171

10. After the E5071C detects the E5091A, the LEDs that indicate the connected test ports remain on.
    - Do not switch on/off devices connected via the USB ports (front or rear panel) or connect/disconnect devices to the USB ports while the E5071C is measuring with the E5091A.
    - Even if you install the driver on a USB port, you will be asked to install driver again if you connect E5091A with a different USB port.

RF Cable Connection

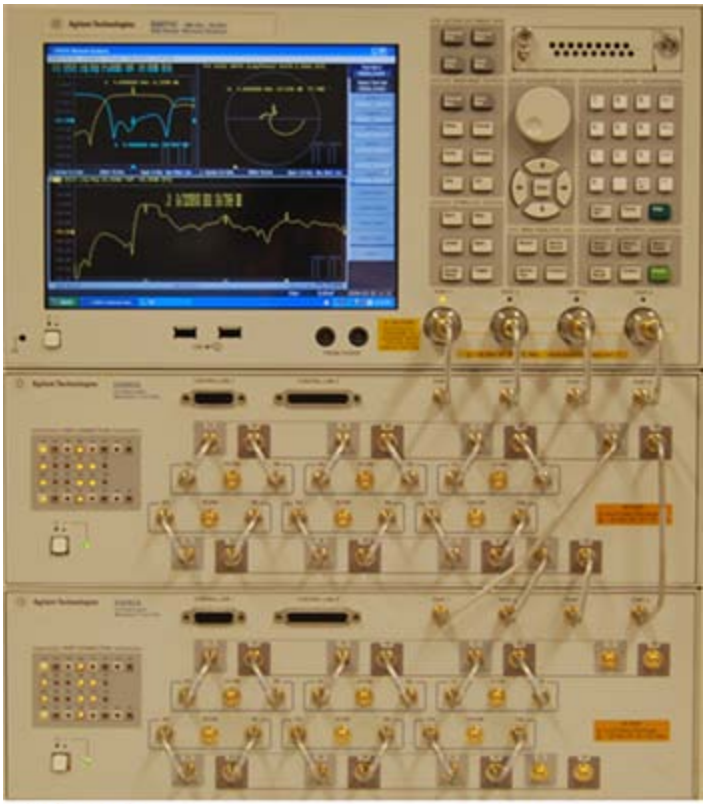
Connect the type-N cables between the test ports (PORT 1 to 4) of the E5071C and the test ports (PORT 1 to 4) of the multiport test set. Make the connection so that the numbers of the ENA ports and the test set ports match.



e5071c039

Connecting Two Multiport Test Sets

Up to two multiport test sets can be connected to one ENA for measurements with more multiple test ports.



e5071c293

1. Set IDs of connected multiport test sets to different values. The instruments will not work correctly if their IDs are the same. For more information, see Selecting ID for Multiport test Set.
2. Press **System** key > **Multiport Test Set Setup** to display the multiport test set setup menu.
3. Select **Test Set 1** for the test set with ID 1 and **Test Set 2** for ID 2.
    4. The E5091A (option 009 and 016) and the E5092A (option 020) can be used at the same time.

Recommended Wrench

When using a semi-rigid cable, connect the semi-rigid cable to a test port of the multiport test set with a specified torque using a torque wrench.

Specified torque	5.7 kgf-cm/ 56 N-cm / 5 in-lb
Recommended wrench	Wrench (Agilent part number 8710-1582)

Setting the Multiport Test Set

This section describes the settings of the multiport test set. The following table shows the flow used for item setting:

Setting flow for multiport test set

Item	Description
Selecting ID for Multiport Test Set	Selects the ID of the multiport test set you want to set
Selecting the Configuration of the Multiport Test Set	Selects the configuration of the multiport test set you want to set
Assigning Test Ports	Assigns test ports of the E5071C and those of the multiport test set
Displaying the Properties	Displays the multiport test set property to check the port setting
Setting Control Line	Makes the setting of the control line that controls the DUT

Other topics about Controlling Multiport Test Set

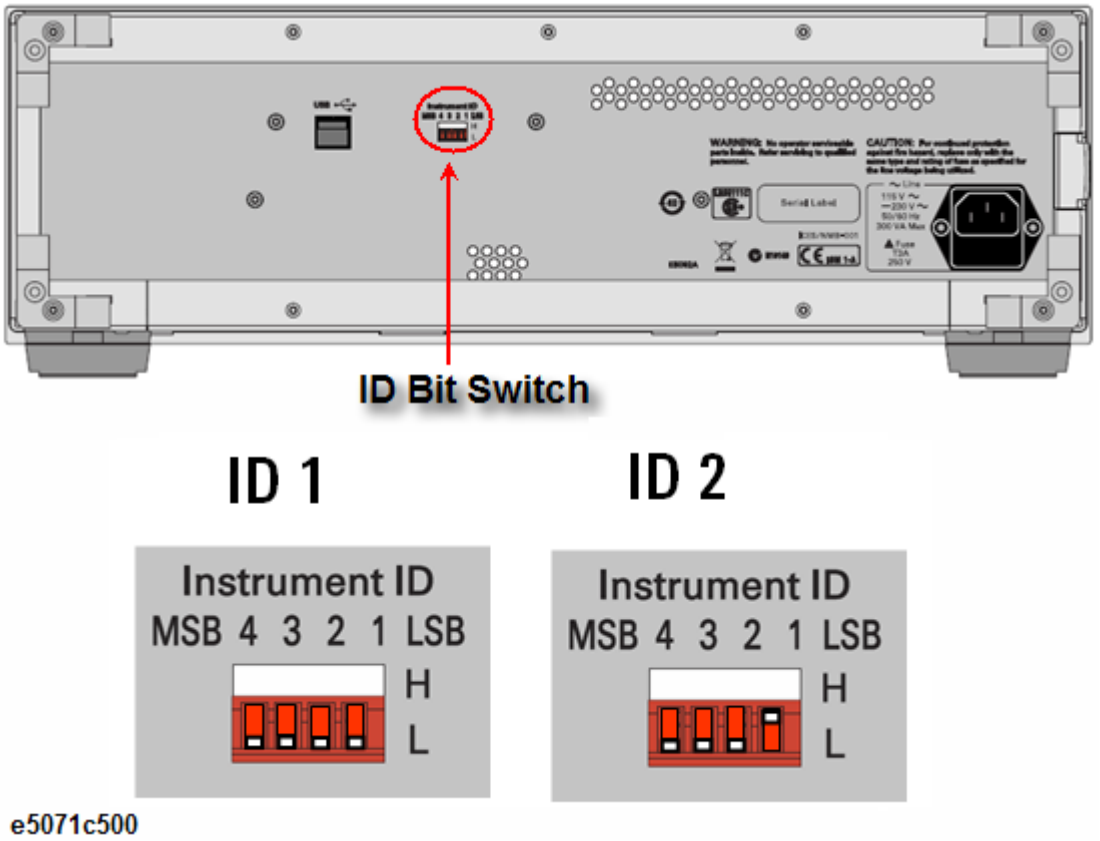
Selecting ID for Multiport Test Set

Operational Procedure

1. Set the target ID to the ID of the connected multiport test set.
2. Press **System** key > **Multiport Test Set Setup** to display the multiport test set setup menu.
3. Select **Test Set 1** for ID 1 and **Test Set 2** for ID 2. The ID is set with the bit switch on the rear panel of the multiport test set.

ID Bit Switch of E5091A

ID Bit Switch of E5092A



e5071c500

- Change the ID bit switch setting while the E5071C is turned off.
- Two multiport test set can be controlled from one E5071C by assigning an ID number (ID 1 or ID 2) on each test set.
- The E5091A (option 009 and 016) and the E5092A (option 020) can be used at the same time.

Selecting the Configuration of the Multiport Test Set

Select the configuration of the multiport test set you want to set.

Operational Procedure

1. Press **System** >**Multiport Test Set Setup** > **Test Set 1** or **Test Set 2**.
2. Click **Select Test Set**.
3. Select the configuration of the multiport test set from the following selection:

Test set	Softkey	Configuration

E5092A	E5092_13	Select the 13-port configuration of the E5092A (E5092A-020). Equivalent to E5091_13 of the E5091A-016.
	E5092_16	Select the 16-port configuration of the E5092A (E5092A-020). Equivalent to E5091_16 of the E5091A-016.
	E5092_22	Select the 22-port configuration of the E5092A (E5092A-020).
	E5092_28	Select the switching independently in the E5092A (E5092A-020).
	E5092_X10	Select the 10-port full crossbar configuration of the E5092A (E5092A-020).
E5091A	E5091_9	Select the 9-port option of the E5091A (E5091A-009).
	E5091_13	Select the 13-port configuration of the E5091A-016.
	E5091_16	Select the 16-port configuration of the E5091A-016.

4. Click **Control** to enable (**ON**) control of the multiport test set.

5. The enable (**ON**)/disable (**OFF**) setting of the control function of the multiport test set is executed for all channels.

- If the model you use and the selected softkey is different, the configuration will not be reflected. Also, no error message will appear. For the correlation between the model and the softkey, refer to the following table:

	E5091_9	**E5091_13**	**E5091_16**	**E5092_13**	**E5092_16**	**E5092_22**	**E5092_28**	**E5092_X10**
E5091A (Option 009)	Yes	No	No	No	No	No	No	No
E5091A (Opti	No	Yes	Yes	No	No	No	No	No

on 016)								
E5092A (Option 020)	No	No	No	Yes	Yes	Yes	Yes	Yes

Assigning Test Ports

Before calibration and measurement, you need to assign the test ports of the multiport test set. You can set the connection ports for each channel and perform measurement while switching the connection for each channel.

Operational Procedure

1. Press **System** key, then click **Multiport Test Set Setup** > **Test Set 1** or **Test Set 2**.
2. Press **Channel Next** (or **Channel Prev)** to activate the channel for which you want to set the connection ports.
3. Use the corresponding softkey to assign the connected test port of the multport test set.
4. Execute Step 2 through Step 3 for all channels for which you want to perform sweep.

When the E5091_9 is Selected

Softkey	Function
Port 1	Selects a test port of the multiport test set connected to the port 1. You can select the port from A or T1.
Port 2	Selects a test port of the multiport test set connected to the port 2. You can select the port from T1 or T2.
Port 3	Selects a test port of the multiport test set connected to the port 3. You can select the port from R1+, R2+, or R3+.
Port 4	Selects a test port of the multiport test set connected to the port 4. You can select the port from R1-, R2-, or R3-.

When the E5092_13/E5091_13 is Selected

Softkey	Function

Port 1	Selects a test port of the multiport test set connected to the port 1. You can select the port from A, T1, T2, or T3.
Port 2	Selects a test port of the multiport test set connected to the port 2. You can select the port from T1, T2, T3, or T4.
Port 3	Selects a test port of the multiport test set connected to the port 3. You can select the port from R1+, R2+, R3+, or R4+.
Port 4	Selects a test port of the multiport test set connected to the port 4. You can select the port from R1-, R2-, R3-, or R4-.

When the E5091_16 is Selected

Softkey	Function
Port 1	Selects a test port of the multiport test set connected to the port 1. You can select the port from A1, A2, A3, A4, A, T1, T2, or T3.
Port 2	Selects a test port of the multiport test set connected to the port 2. You can select the port from B1, B2, B3, B4, T1, T2, T3, or T4.
Port 3	Selects a test port of the multiport test set connected to the port 3. You can select the port from R1+, R2+, R3+, or R4+.
Port 4	Selects a test port of the multiport test set connected to the port 4. You can select the port from R1-, R2-, R3-, or R4-.
Port 5	Selects a test port of the switch #13 (SW13). You can select the port from X1 or X2.
Port 6	Selects a test port of the switch #14 (SW14). You can select the port from Y1 or Y2.
Port 7	Selects a test port of the switch #15 (SW15). You can select the port from Z1 or Z2.

- The same test ports cannot be connected to each port. In such a case, the other test ports settings will be automatically changed.

When the E5092_16 is Selected

Softkey	Function
Port 1	Selects a test port of the multiport test set connected to the port 1. You can select the port from A1, A2, A3, or A4.
Port 2	Selects a test port of the multiport test set connected to the port 2. You can select the port from B1, B2, B3, or B4.
Port 3	Selects a test port of the multiport test set connected to the port 3. You

	can select the port from R1+, R2+, R3+, or R4+.
Port 4	Selects a test port of the multiport test set connected to the port 4. You can select the port from R1-, R2-, R3-, or R4-.
Port 5	Selects a test port of the switch #8 (SW8). You can select the port from X1 or X2.
Port 6	Selects a test port of the switch #9 (SW9). You can select the port from Y1 or Y2.
Port 7	Selects a test port of the switch #10 (SW10). You can select the port from Z1 or Z2.

When the E5092_22 is Selected

Softkey	Function
Port 1	Selects a test port of the multiport test set connected to the port 1. You can select the port from A1, A2, A3, A4, A5,or A6.
Port 2	Selects a test port of the multiport test set connected to the port 2. You can select the port from A7, A8, A9, A10, or A11.
Port 3	Selects a test port of the multiport test set connected to the port 3. You can select the port from B1, B2, B3, B4, B5, or B6.
Port 4	Selects a test port of the multiport test set connected to the port 4. You can select the port from B7, B8, B9, B10, or B11.

When the E5092_28 is Selected

Softkey	Function
Port 1	Selects a test port of the switch #1 (SW1). You can select the port from A, B, C, or D.
Port 2	Selects a test port of the switch #2 (SW2). You can select the port from A, B, C, or D.
Port 3	Selects a test port of the switch #3 (SW3). You can select the port from A, B, C, or D.
Port 4	Selects a test port of the switch #4 (SW4). You can select the port from A, B, C, or D.
Port 5	Selects a test port of the switch #5 (SW5). You can select the port from A or B.
Port 6	Selects a test port of the switch #6 (SW6). You can select the port from A or B.

Port 7	Selects a test port of the switch #7 (SW7). You can select the port from A or B.
Port 8	Selects a test port of the switch #8 (SW8). You can select the port from A or B.
Port 9	Selects a test port of the switch #9 (SW9). You can select the port from A or B.
Port 10	Selects a test port of the switch #10 (SW10). You can select the port from A or B.

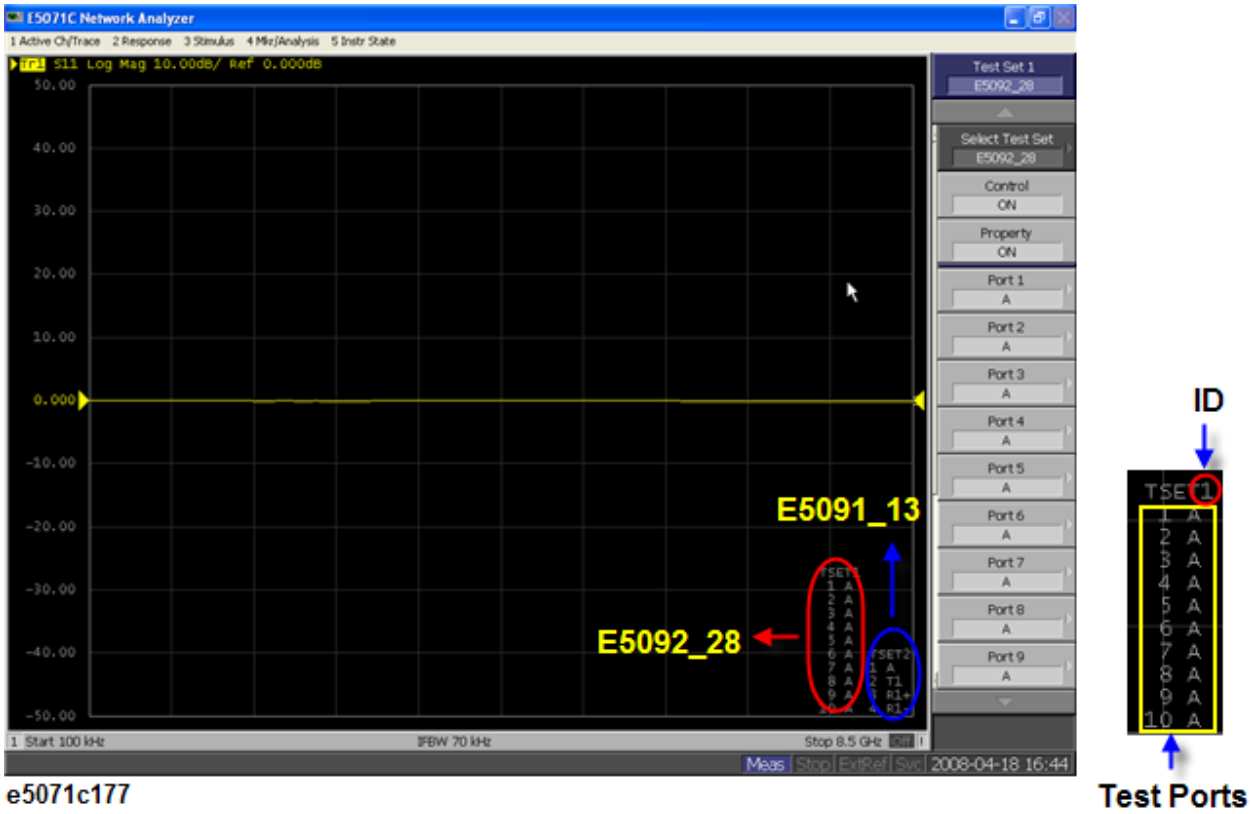
When the E5092_X10 is Selected

Softkey	Function
Port 1	Selects a test port of the multiport test set connected to the port 1. You can select the port from 1, 3, 5, or 7.
Port 2	Selects a test port of the multiport test set connected to the port 2. You can select the port from 2, 4, 6, or 8.
Port 3	Selects a test port of the multiport test set connected to the port 3. You can select the port from 2, 4, 6, or 10.
Port 4	Selects a test port of the multiport test set connected to the port 4. You can select the port from 1, 3, 5, or 9.

Displaying the Properties

By displaying the multiport test set properties shown below, you can obtain the assignment information of the test ports for each channel. This is useful when you need to check the test port assignment, for example, when you perform calibration.

Multiport Test Set properties



e5071c177

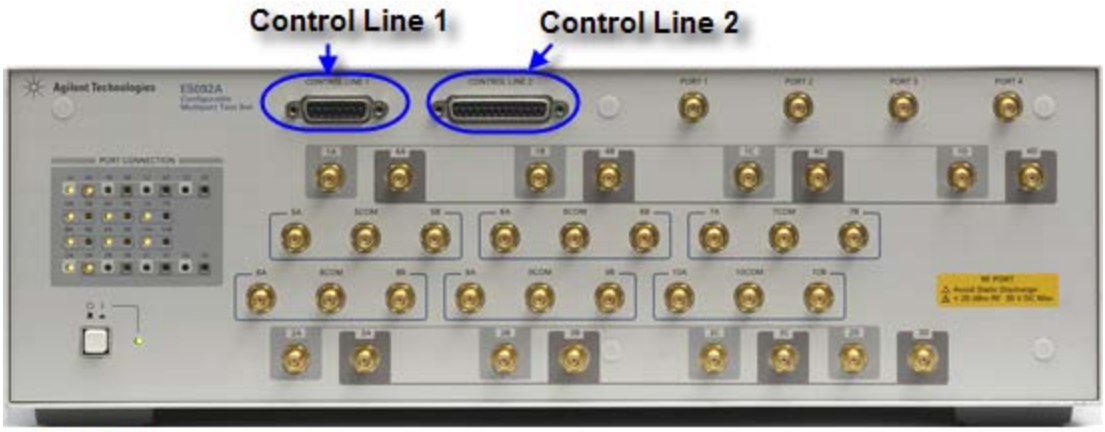
Operational Procedure

1. Press **System** Key, then click **Multiport Test Set Setup** > **Test Set 1** or **Test Set 2**.
2. Click **Property** to enable it (**ON**) to display the multiport test set's properties.
    - The enable (**ON**)/disable (**OFF**) setting of the multiport test set properties display is executed for all channels.

Setting Control Line

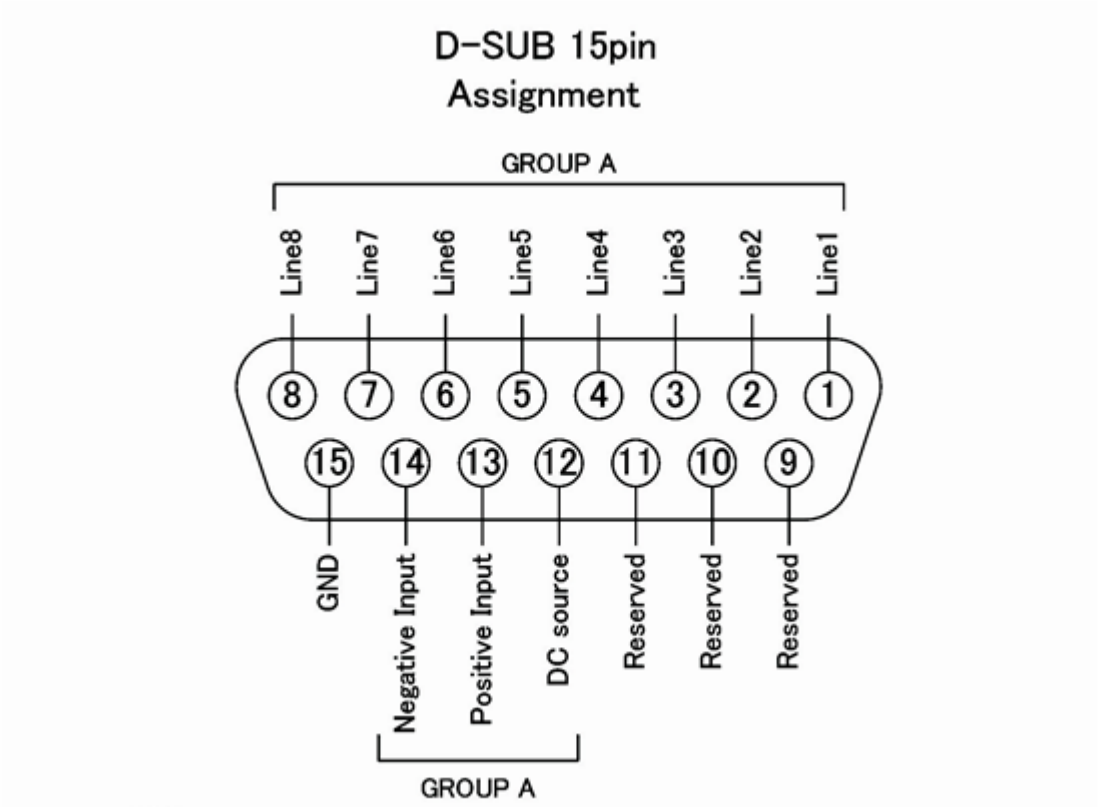
The E5071C can control the output from the control line of the multiport test set and control the DUT (for example, switching the frequency band of the front end module).

E5092A Front Panel

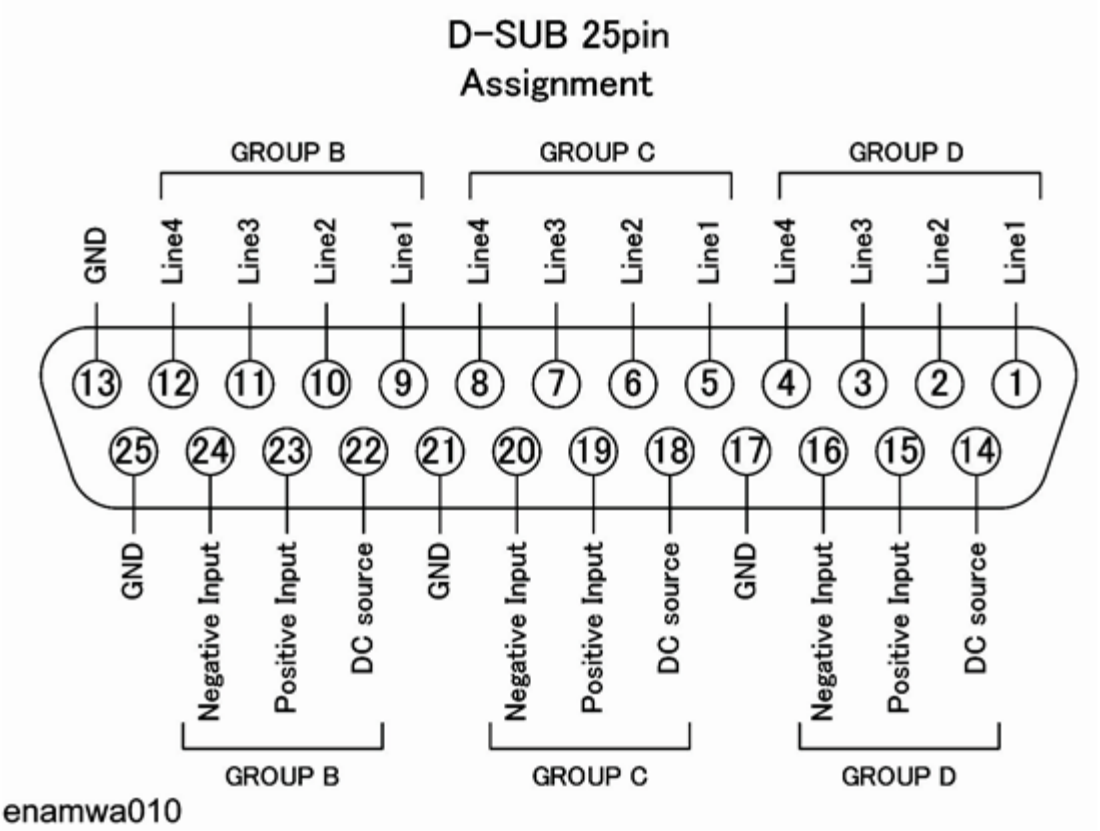


e5071c292

E5092A Control Line



enamwa009



enamwa010

E5091A Front Panel

Operational Procedure

1. Press **System** Key, then click **Multiport Test Set Setup** > **Test Set 1** or **Test Set 2**.
2. Press **Channel Next** (or **Channel Prev)** to activate the channel for which you want to set the control line.
3. Click **Control Lines** for E5091A or **Control Line A/B/C/D** for E5092A to display the settings menu for the DUT control line.
4. Use the corresponding softkey to set the control line of the multiport test set.

Softkey	Function
Line 1 to **Line 8**	Set **High/Low** of each line of the control line.
DC source (only for E5092A)	Set DC source voltage.

5. Execute Step 3 through Step 4 for all channels that you want to sweep.

Performing Measurement

- Calibration
- Making Measurements

Other topics about Controlling Multiport Test Set

Calibration

Follow these steps to perform calibration with the multiport test set connected:

1. Press **Channel Next** (or **Channel Prev**) key to set the channel that you want to calibrate to the active channel.
2. Follow Displaying the properties to display the multiport test set properties.
3. Perform the calibration. Check the corresponding multiport test set test ports shown in the calibration properties as the port names of the E5071C are displayed on the calibration menu, connect the calibration standard to the corresponding test ports of the multiport test set, and perform the calibration.

Making Measurements

Making measurements with multiport test set is the same as E5071C standalone measurement. Perform operations by referring to Making Measurements.

Trigger state and switching the setting of the Multiport test set

The following table shows how the setting in the multiport test set is switched from when the trigger state is the stop state.

Trigger state	Switching the setting of multiport test set
Stop	The setting is not switched.
Trigger wait	The setting of the internal switch and the output of the control line are switched according to the setting of the channel swept first. The connections of the test ports and the output of the control line are switched according to the setting of the channel swept first.
Measurement	Measurement is performed following the procedure below. 1. Execute a sweep for the first channel. 2. Set the connections of the test ports and the output of the control line according to the settings of the channel swept

	second. 3. Execute a sweep for the second channel. 4. Set the connections of the test ports and the output of the control line according to the setting of the channel swept last. 5. Execute a sweep for the last channel.
Stop or trigger wait	The setting is not switched for the stop state; it is switched for the trigger wait state.

Controlling Multiport Test Set by Programming

- Selecting Test Set
- Selecting ports assigned to Port 1 to Port n
- Difference between E5092A and E5091A
- Sample program

Other topics about Controlling Multiport Test Set

Selecting Test Set

The multiport test set provides test set 1 and test set 2. To select the test set, use the following command:

:SENS:MULT{1-2}:NAME

The following models are available for the test set:

Multiport Test Set		Option
E5092A	E5092_13	Option 020 (13-port measurement)
	E5092_16	Option 020 (16-port measurement)
	E5092_22	Option 020 (22-port measurement)
	E5092_28	Option 020 (28-port measurement)
	E5092_X10	Option 020 (X10-port measurement)
E5091A	E5091_9	Option 009
	E5091_13	Option 013 (13-port measurement)
	E5091_16	Option 016 (16-port measurement)

Checking the name of available test set

To check the name of the available multiport test set, use the following command:

:SENS:MULT:CAT?

Turning control ON/OFF

To turn ON/OFF the control of the multiport test set, use the following command:

:SENS:MULT{1-2}:STAT

If you turn OFF the control of the multiport test set, it does not affect the operation of the E5071C, even if it is connected. You can control test set 1 and test set 2 separately.

Selecting ports assigned to Port 1 to Port n

Selecting the connection ports

You can select the ports assigned to Port 1 to Port n for each channel. To select the ports, use the following commands:

:SENS{1-36}:MULT{1-2}:PORT{1-20}

- The connection between the assigned ports and Port 1 to Port n inside the E5091A is not changed when one of the above commands is executed but it is changed immediately before a sweep for each channel.

Turning ON/OFF state display of connection ports (properties display)

You can display the state of the ports assigned to Port 1 to Port n (multiport test set properties) in the lower right part in the window for each channel. To turn ON/OFF the properties display, use the following command:

:SENS:MULT{1-2}:DISP

Checking number of ports

To check the number of the multiport test set connected ports, use the following command:

:SENS:MULT{1-2}:COUN?

Checking the available port name

To check the name of the multiport test set connected ports, use the following command:

:SENS:MULT{1-2}:PORT{1-20}:CAT?

Checking number of input ports

To check the number of the multiport test set input ports, use the following command:

:SENS:MULT{1-2}:INC?

Setting control line

You can set the HIGH/LOW state of each line of the control line for each channel. To set the HIGH/LOW of each line, use the following command:

:SENS:MULT{1-2}:TSET9:OUTP

- The HIGH/LOW state of each line of the E5091A is not changed when the above command is executed but it is changed immediately before a sweep for each channel.

Difference between E5092A and E5091A

Difference	E5092A	E5091A
Options	13 port, 16 port, 22 port, 28 port and X10 port	9 port, 13 port and 16 port
Number of control lines	2	1

- The channel number for E5092A has increased because the number of supported test sets has increased.

Controlling E5092A

- The following commands are available only in E5092A.

Setting output data/voltage and switch

To set the value of output data/voltage and switch, use the following command:

:CONT:MULT[1-2]:OUTP:[A-D][:DATA]

:CONT:MULT[1-2]:OUTP:[A-D]:VOLT[:DATA]

:CONT:MULT[1-2]:PORT[1-20][:SEL]

:SENS[1-160]:MULT[1-2]:OUTP[:DATA]

:SENS[1-160]:MULT[1-2]:OUTP:[A-D][:DATA]

:SENS[1-160]:MULT[1-2]:OUTP:[A-D]:VOLT[:DATA]

:SENS[1-160]:MULT[1-2]:PORT[1-20][:SEL]

Selecting test set

To select the test set, use the following command:

:SENS[1-160]:MULT[1-2]:NAME

For E5091A test set, only the following two commands are available:

:CONT:MULT[1-2]:PORT[1-20][:SEL]

:CONT:MULT[1-2]:OUTPut:A[:DATA]

Sample program

See Control E5091A.