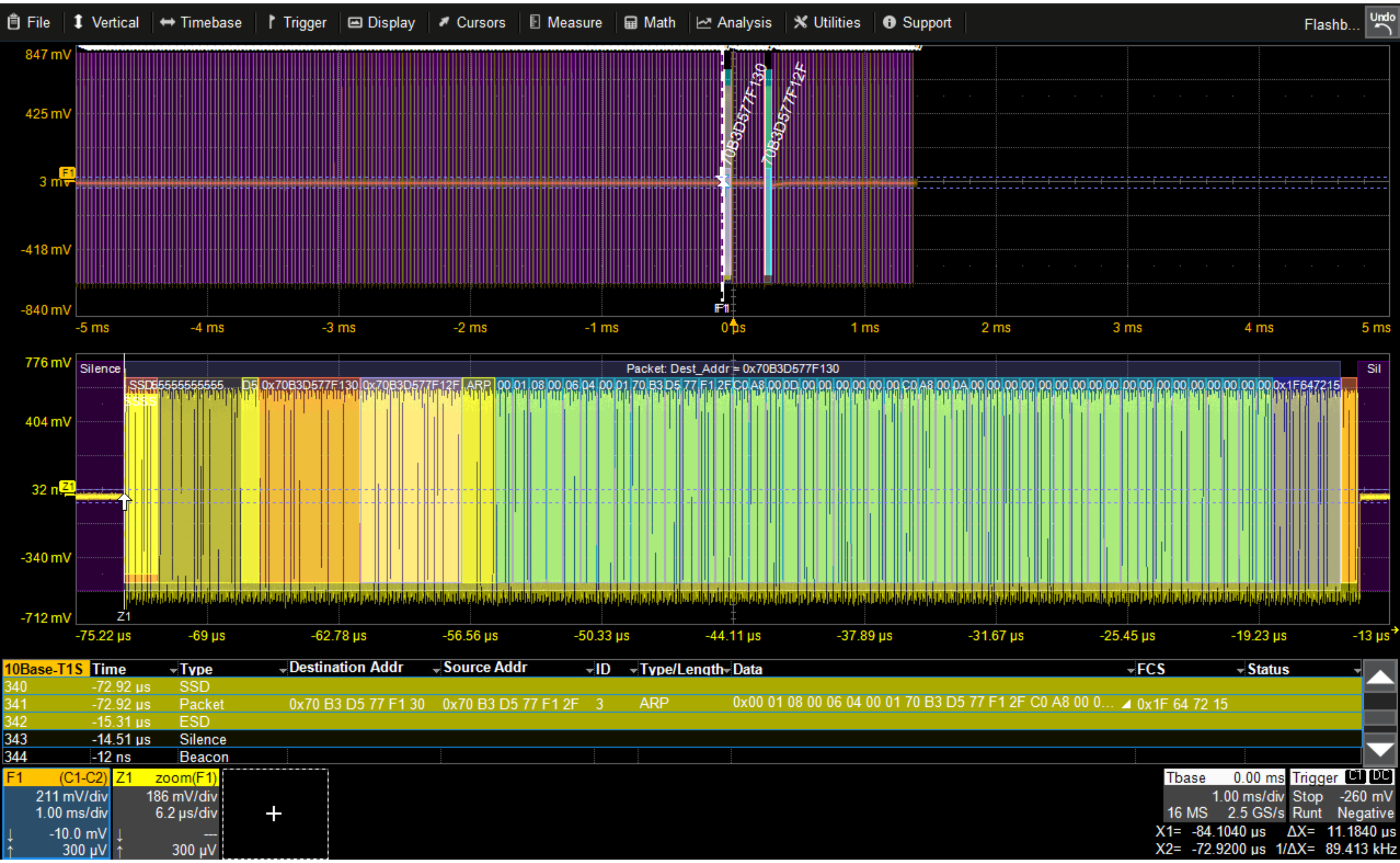


10Base-T1S

Trigger, Decode, Measure/Graph and Eye Diagrams

10Base-T1S TDME Instruction Manual

© 2022 Teledyne LeCroy, Inc. All rights reserved.

Users are permitted to duplicate and distribute Teledyne LeCroy, Inc. documentation for internal educational purposes only. Resale or unauthorized duplication of Teledyne LeCroy publications is strictly prohibited.

Teledyne LeCroy is a trademark of Teledyne LeCroy, Inc., Inc. Other product or brand names are trademarks or requested trademarks of their respective holders. Information in this publication supersedes all earlier versions. Specifications are subject to change without notice.

July, 2022
10base-t1s-tdme-im_19jul22.pdf

Contents

About This Manual

This manual explains the basic procedures for using serial data trigger and decode software options for Teledyne LeCroy oscilloscopes. There are also sections pertaining to the measure, graph and eye diagram capabilities of TDME options. It is assumed that you have a basic understanding of the serial data physical layer specifications, and how to use the oscilloscope on which the option is installed. Only features specific to this product are explained in this manual.

While some images may not exactly match what is on your oscilloscope display—or may show an example taken from another standard—be assured that the functionality is identical. Product-specific exceptions will be noted in the text.

Some capabilities described may only be available with the latest version of our MAUI® software. Updates are available from the software download page at teledynelecroy.com under Oscilloscope Downloads > Firmware Upgrades.

Introducing 10Base-T1S TD and TDME Options

Automotive Ethernet enables faster data communication to meet the demands of today's vehicles and the connected vehicles of the future. Automotive Ethernet can be used to refer to any Ethernet-based network for in-vehicle electrical systems. It encompasses 10Base-T1S, as well as several other variants/speeds of Automotive Ethernet (e.g., 100Base-T1, 1000Base-T1).

The 10Base-T1S decoder applies software algorithms to extract Ethernet control and payload data from 10Base-T1S signals measured on your oscilloscope. When displayed on oscilloscopes or in MAUI® Studio remote oscilloscope software, the extracted information overlays the actual physical layer waveforms, color-coded to provide fast, intuitive understanding of the relationship between message frames and other time-synchronous events.

The trigger and decode (-TD) option enables you to trigger the oscilloscope upon finding beacon packets, commit packets, ESDs, packets containing an individual node ID or node ID plus data pattern, or protocol errors.

The Measure/Graph and Eye Diagram option (-TDME/-DME) adds a set of measurements designed for serial data analysis and protocol-specific eye diagram tests to the standard trigger and decoder capabilities. See Measuring for instructions on using the measure and graphing capabilities. See Eye Diagram Tests for instructions on using the eye diagram tests.

Serial Decode

The methods described here at a high level are used by all Teledyne LeCroy serial decoders, differing only slightly for signals with an embedded clock and separate clock and data signals.

Bit-level Decoding

The first software algorithm examines the embedded clock based on a default or user- specified vertical threshold level. Once the clock signal is extracted, the algorithm examines the traffic to determine whether a data bit is high or low. The default High and Low levels are automatically determined from a measurement of the amplitude of the signals acquired by the oscilloscope. Alternatively, they can be manually set by the user. The algorithm intelligently applies a hysteresis to the rising and falling edge of the serial data signal to minimize the chance of perturbations or ringing on the edge affecting the data bit decoding.

Note: Although the decoding algorithm is based on a clock extraction software algorithm using a vertical level, the results returned are the same as those from a traditional protocol analyzer using sampling point-based decode.

Logical Decoding

After determining individual data bit values, another algorithm performs a decoding of the serial data message after separation of the underlying data bits into logical groups specific to the protocol (Header/ID, Address Labels, Data Length Codes, Data, CRC, Parity Bits, Start Bits, Stop Bits, Delimiters, Idle Segments, etc.).

Message Decoding

Finally, another algorithm applies a color overlay with annotations to the decoded waveform to mark the transitions in the signal. Decoded message data is displayed in tabular form below the grid. Various compaction schemes are utilized to show the data for the duration of the acquisition, from as little as one serial data message acquisition to many thousands. In the case of long acquisitions, only the most important information is highlighted, whereas with the shortest acquisition, all information is displayed with additional highlighting of the complete message frame.

User Interaction

Your interaction with the software in many ways mirrors the order of the algorithms. You will:

- Assign a protocol/encoding scheme and input sources to one of the four decoder panels using the Serial Data and Decode Setup dialogs.
- Complete the remaining subdialogs required by the protocol/encoding scheme.
- Work with the decoded waveform, result table, and measurements to analyze the decoding.

Decoding Workflow

We recommend the following workflow for effective decoding:

1. Set up the decoder using the lowest level decoding mode available, but do not yet enable it.
2. Acquire at least one complete transmission reasonably well centered on screen in both directions, with generous idle segments on both sides.

   Note: See Failure to Decode for more information about the required acquisition settings.

3. Stop acquisition, then enable the decoder. It will operate on the acquisition in buffer.
4. Use the various decoder tools to verify that transitions are being correctly decoded. Tune the decoder settings as needed to produce a satisfactory decoding.
5. Once you are correctly decoding in one mode, continue making small acquisitions of five to eight transmissions and run the decoder in higher level modes.
6. Finally, run the decoder on acquisitions of the desired length.

When you are satisfied the decoder is working properly, you can disable/enable the decoder as desired without having to repeat this tuning process, provided the basic signal characteristics do not change.

Decoder Set Up

Use the Decode Setup dialog and its protocol-related subdialogs to preset decoders for future use. Each decoder can use different protocols and data sources, or have other variations, giving you maximum flexibility to compare different signals or view the same signal from multiple perspectives.

1. Touch the **Front Panel Serial Decode button** (if available on your oscilloscope), or choose **Analysis > Serial Decode** from the oscilloscope menu bar.
2. On the Serial Decode dialog, enable the decoder by checking **On** next to the decoder number. This may be done any time, although we recommend having an acquisition in buffer before enabling the decoder.
3. Click the **Setup** button at the end of the row to open the Decode Setup dialog.

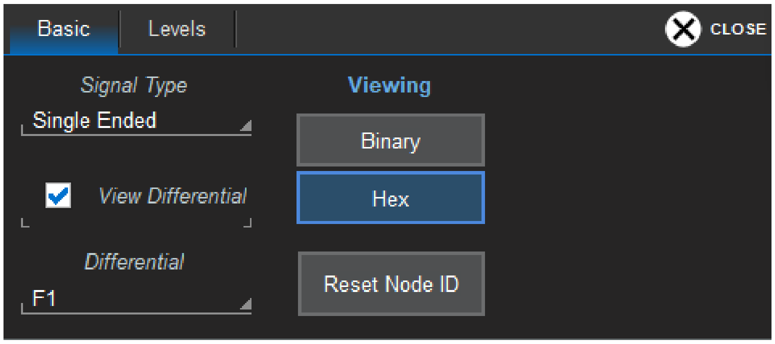
   Note: The full configuration can only be made from Decode Setup.

4. Enter the input channels (sources) and select the **Protocol** to be decoded. This selection will drive the other fields that appear.

   Note: Subdialog selections, such as single-ended or differential probing, may affect the input fields that appear on the Decode Setup dialog.

5. Define the level of decoding on the subdialogs (see below) to the right of the Decode Setup dialog.

Basic Subdialog



Select the **Signal Type**, **Differential** or **Single Ended**.

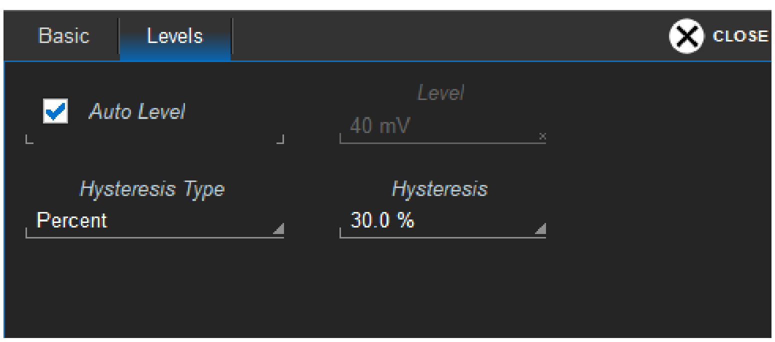
When using single-ended inputs, you can check **View Differential** to generate a differential trace of the inputs. Enter the math function slot to use for the **Differential** trace (F*n*).

If you choose to generate the differential trace, the decoding will be applied to your selected math functions, rather than the single-ended channel inputs. Also, all levels and measurements will be referenced to the differential levels, even if the math trace is not displayed on the screen.

Select how you would like to view payload data, in **Binary** or **Hex**(adecimal) code.

The decoder will learn the ID numbers of nodes based on their Source Address. Select **Reset Node ID** to clear the mappings when there is a change to the addresses of nodes on the bus.

Levels Subdialog



The default **Auto Level** selection will analyze the input signal to determine the logic high and and low levels that produce the best Differential Manchester decoding. For most applications, this will be sufficient, and we recommending leaving it selected.

If the Auto Level selection does not produce a satisfactory decoding, deselect the checkbox and enter the logic threshold **Level**. Optionally, you can also set a **Hysteresis** to prevent false determinations. You may need to tune these settings until a satisfactory decoding appears following acquisition.

Setting Level and Hysteresis

The **Level** setting represents the logical level for bit transition, corresponding to the physical Low and High distinction. Level is normally set as 50% of waveform amplitude, but can sometimes be set as an absolute voltage (with reference to the waveform 0 level).

Percent mode is easy to set up because the software immediately determines the optimal threshold, but in some cases it might be beneficial to switch to Absolute mode when available:

- On poor signals, where Percent mode can fail and lead to bad decodes
- On noisy signals or signals with a varying DC component
- On very long acquisitions, where Percent mode adds computational load

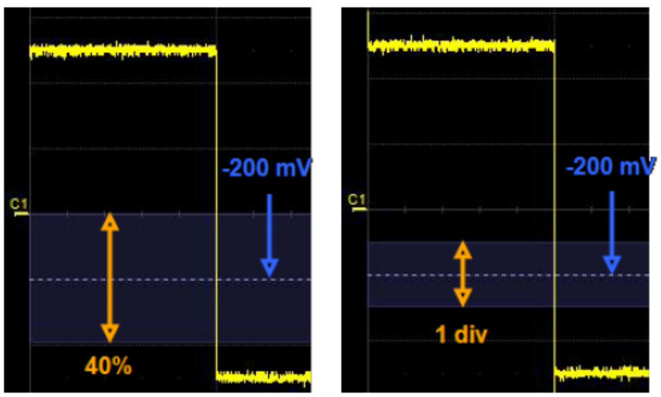
The transition Level appears as a dotted, horizontal line across the oscilloscope grid. If your initial decoding indicates that there are a number of error frames, make sure that Level is set to a reasonable value.

The optional **Hysteresis** setting imposes a limit above and below the measurement level that precludes measurements of noise or other perturbations within this band.

A blue marker around the Level line indicates the area of the hysteresis band. Depending on protocol, the **Hysteresis Type** may be percent amplitude, vertical grid divisions or absolute voltage level.

Observe the following when setting Hysteresis:

- Hysteresis must be larger than the maximum noise spike you wish to ignore.
- The largest usable hysteresis value must be less than the distance from the level to the closest extreme value of the waveform.



Hysteresis set as 40 percent of total waveform amplitude (left) and Hysteresis set as equivalent of 1 grid division (right) around an absolute -200mV Level setting.

Note: Usually, you can set the Level and Hysteresis in different modes. For a few protocols, there is only one option for setting Level or Hysteresis.

Failure to Decode

Several conditions may cause a decoder to fail, in which case a message will appear in the first row of the summary result table, instead of in the message bar as usual. In these cases, the decoding is turned off to protect you from incorrect data. Adjust your acquisition settings accordingly, then re-enable the decoder.

All decoders will test for the condition **Too small amplitude**. If the signal's amplitude is too small with respect to the full ADC range, the message "Decrease V/Div" will appear. The required amplitude to allow decoding is usually one vertical division.

If the decoder incorporates a user-defined bit rate (usually these are protocols that do not utilize a dedicated clock/strobe line), the following two conditions are also tested:

- **Under sampled**. If the sampling rate (SR) is insufficient to resolve the signal adequately based on the bit rate (BR) setup or clock frequency, the message "Under Sampled" will appear. The minimum SR:BR ratio required is 4:1. It is suggested that you use a slightly higher SR:BR ratio if possible, and use significantly higher SR:BR ratios if you want to also view perturbations or other anomalies on your serial data analog signal.

- **Too short acquisition**. If the acquisition window is too short to allow any meaningful decoding, the message "Too Short Acquisition" will appear. The minimum number of bits required varies from one protocol to another, but is usually between 5 and 50.

- **Poor signal quality**. Care must be taken when probing high speed serial data signals (typically with a high bandwidth differential probe). Channel loss, reflections and probe loading can degrade the signal. Its best to probe at the termination of a high speed serial link to minimize probe loading effects and reflections. If the signal has significant channel loss, the CTLE/DFE equalizers in the SDAIII software can be used to improve the quality of the signal being decoded.

Note: It is possible that several conditions are present, but you will only see the first relevant message in the table. If you continue to experience failures, try adjusting the other settings.

Serial Decode Dialog

After decoders have been configured on the Decode Setup dialog, use the Serial Decode dialog to quickly change the input channels (sources) or turn the decoder on/off. If you change protocols, the last settings configured for that protocol will be resumed.

To enable decoders, on the same row as Decode *N*, check **On**. If there is a valid acquisition, a result table and annotated waveform appear.

To turn off decoders, deselect the On boxes individually, or touch **Turn All Off**.

Tip: If you wish to inspect the decoding, best practice is to make single acquisitions, stop, then enable the decoder to apply it to the buffered acquisition. If you wish to accumulate or graph serial data measurements, it may be better to run the decoder on continuous acquisitions.

Reading Waveform Annotations

When a decoder has operated successfully on a valid acquisition, an annotated waveform appears on the oscilloscope display, allowing you to quickly see the relationship between the protocol decoding and the physical layer. A colored overlay marks significant bit-sequences in the source signal: Header/ID, Address, Labels, Data Length Codes, Data, CRC, Parity Bits, Start Bits, Stop Bits, Delimiters, Idle segments, etc. Annotations are customized to the protocol or encoding scheme.

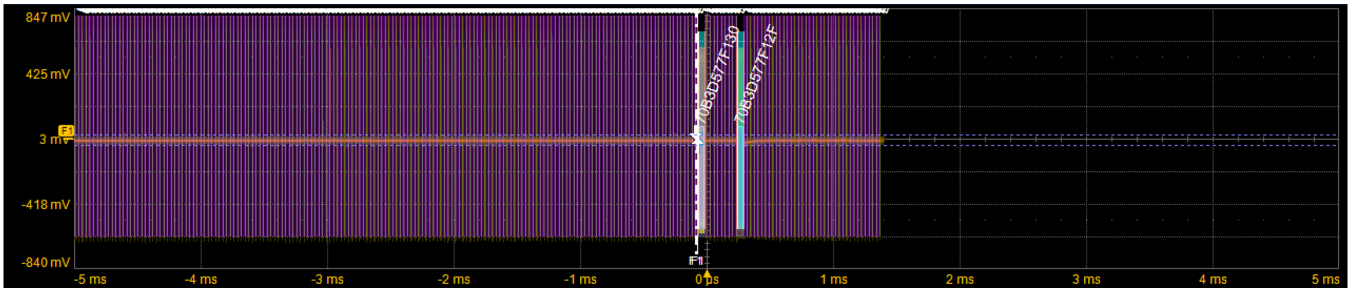
The amount of information shown on an annotation is affected by the width of the rectangles in the overlay, which is determined by the magnification (scale) of the trace and the length of the acquisition. Zooming a portion of the decoded trace by clicking a line in the table will reveal the detailed annotations.

10Base-T1S Waveform Annotations

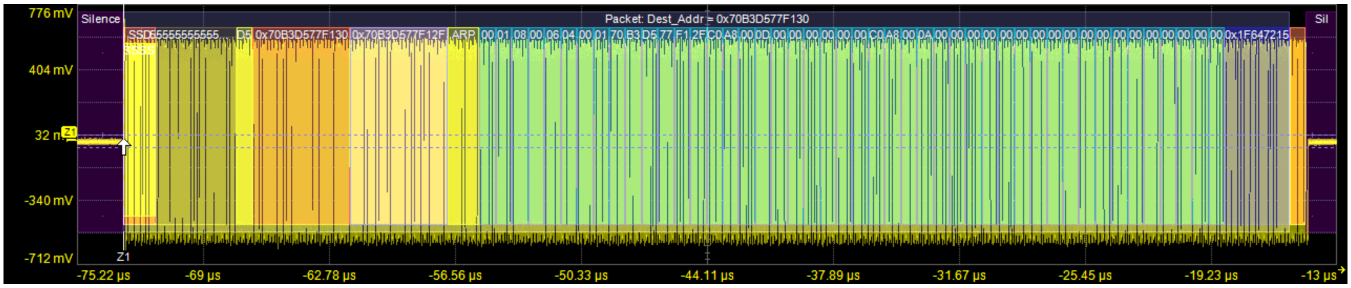
These overlays appear on a decoded 10Base-T1S waveform:

Annotation	Overlay Color (1)	Text (2)
Protocol Error	Bright Red	Error
Inter-transmission Silence	Dark Purple	Silence
Beacon	Charcoal	Beacon
Commit	Brown	COMMIT
Packet	Dark Blue (behind SSD through ESD)	Packet Dest. Address= <hex code>
Start-of-Stream Delimiter	Dark Red (behind SYNC Bytes)	SSD
SYNC Bytes (4)	Gold	SYNC1, SYNC2, SSD1, SSD2
Preamble	Charcoal	Preamble=<hex code>
Start Frame Delimiter	Gold	SFD=<hex code>
Destination Address	Dark Red	Dest_Address=<binary | hex code> (3)
Source Address	Lavender-Gray	Src_Address=<binary | hex code>
Data Type and Length	Gold	Type_Len=<type code>
Payload Data	Aqua Blue	Data=<binary | hex code>
Frame Check Sequence	Royal Blue	FCS=<hex code>
End-of-Stream Delimiter	Dark Red	ESD

1. Combined overlays affect the appearance of colors. Box outline shows true color.
2. Text in brackets < > is variable. The amount of text shown depends on your zoom factors.
3. Viewing setting determines format of data shown, binary or hexadecimal.



Initial 10Base-T1S decoding. At this resolution, very little information is visible on the overlay.



Zoom of packet showing annotations details.

Serial Decode Result Table

When you have selected to turn a decoder **On** or to **View Decode**, and a valid acquisition has been decoded using that protocol, a table summarizing the decoder results appears below the grids. This result table provides a view of data as decoded during the most recent acquisition, even when there are too many bursts for the waveform annotation to be legible.

You can export result table data to a .CSV file.

Tip: The result table does not have to be visible in order for the decoder to function. Hiding the table can improve performance when your aim is to use the decode in downstream processes, like measurements.

Table Rows

Each row of the table represents one index of data found within the acquisition. What exactly this represents depends on the protocol and how you have chosen to "packetize" the data stream when configuring the decoder.

When multiple decoders are run at once, the index rows are interleaved in a summary table, ordered according to their acquisition time. The Protocol column is colorized to match the input source that resulted in that index.

Note: The interleaved summary table will default to the lowest common decoding (e.g., hexadecimal if both support that, but only one supports symbolic).

You can change the number of rows displayed on the table at one time. The default is five rows.

Swipe the table up/down or use the scrollbar at the far right to navigate the table. See Using the Result Table for more information about how to interact with the table rows to view the decoding.

Table Columns

When a single decoder is enabled, the result table shows the protocol-specific details of the decoding. This **detailed result table** may be customized to show only selected columns. A **summary result table** combining results from two decoders always shows these columns.

Column	Extracted or Computed Data
Index	Number of the line in the table
Time	Time elapsed from start of acquisition to start of message
Protocol	Protocol being decoded
Message	Message identifier bits
Data	Data payload
CRC	Cyclic Redundancy Check sequence bits
Status	Any decoder messages; content may vary by protocol

Index	Time	Protocol	Message	Data	CRC	Status
1	-51.064 ms	LIN	0f	3c 5a	69	
2	-2.4322 ms	10Base-T1S	Silence			
3	-2.4321 ms	10Base-T1S	Silence			
4	-2.4178 ms	10Base-T1S	Beacon			
5	-2.4158 ms	10Base-T1S	Silence			

Example summary result table, with results from two decoders interleaved on one table.

When you select the Index number from the summary result table, the detailed results for that index drop-in below it.

Index	Time	Protocol	Message	Data	CRC	Status
1	-51.064 ms	LIN	0f	3c 5a	69	
2	-2.4322 ms	10Base-T1S	Silence			
3	-2.4321 ms	10Base-T1S	Silence			

Time	Break	Synch	ID	Parity	DataLength	Data	Chec…	Bit Rate	Symbol
-51.06 ms	0		0f	03	2	3c 5a	69	19.09 kbit/s	

Example summary result table showing drop-in detailed result table.

This extracted data appears on the 10Base-T1S detailed result table.

Column	Extracted or Computed Data
Index	Number of the row in the table
Time(µs)	Time elapsed from start of acquisition to corresponding transmission
Type	Transmissions type: Silence, Beacon, SSD, Packet, ESD
Destination Addr	Packet destination address
Source Addr	Packet source address
ID	Destination Node ID when using Multidrop topology
Type/Length	Code indicating type and length of packet data (e.g., ARP)
Data	Payload data in binary or hexadecimal, depending on Viewing selection
FCS	Frame Check Sequence hexadecimal code
Status	Errors or other messages regarding the decoding (e.g., No SSD)

10Base-T1S	Time	Type	Destination Addr	Source Addr	ID	Type/Length	Data	FCS	Status
340	-72.92 µs	SSD							
341	-72.92 µs	Packet	0x70 B3 D5 77 F1 30	0x70 B3 D5 77 F1 2F	3	ARP	0x00 01 08 00 06 04 00 01 70 B3 D5 77 F1 2F C0 A8 00 0...	0x1F 64 72 15	
342	-15.31 µs	ESD							
343	-14.51 µs	Silence							
344	-12 ns	Beacon							

Section of typical 10Base-T1S Ethernet detailed result table.

Using the Result Table

Besides displaying the decoded serial data, the result table helps you to inspect the acquisition.

Zoom & Search

Touching any cell of the table opens a zoom centered around the part of the waveform corresponding to the index. The Z*n* dialog opens to allow you to rescale the zoom, or to Search the acquisition. This is a quick way to navigate to events of interest in the acquisition.

Tip: When in a summary table, touch any data cell *other than* Index and Protocol to zoom.

The table rows corresponding to the zoomed area are highlighted, as is the zoomed area of the source waveform. The highlight color reflects the zoom that it relates to (Z1 yellow, Z2 pink, etc.). As you adjust the zoom scale, the highlighted area may expand to several rows of the table, or fade to indicate that only a part of that Index is shown in the zoom.

When there are multiple decoders running, each can have its own zooms of the decoding open at once. In this case, multiple rows of the summary table are highlighted to show which indexes are shown in the zooms. These highlights will be different colors to indicate which rows correspond to each decoder.

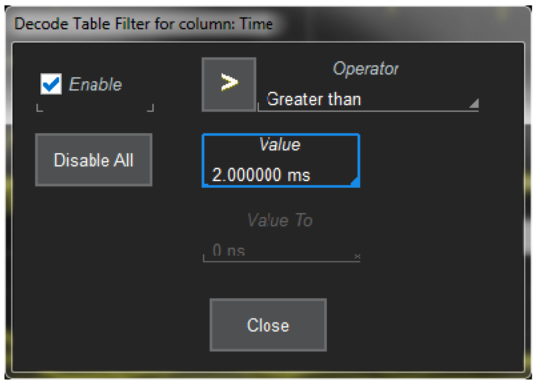
Note: The zoom number is no longer tied to the decoder number. The software tries to match the numbers, but if it cannot it uses the next empty zoom in the sequence.

Example multi-decoder summary table, both indexes highlighted.

Filter Results

Columns of data with a drop-down arrow in the header cell can be filtered: 

Touch the **header cell** to open the Decode Table Filter dialog.



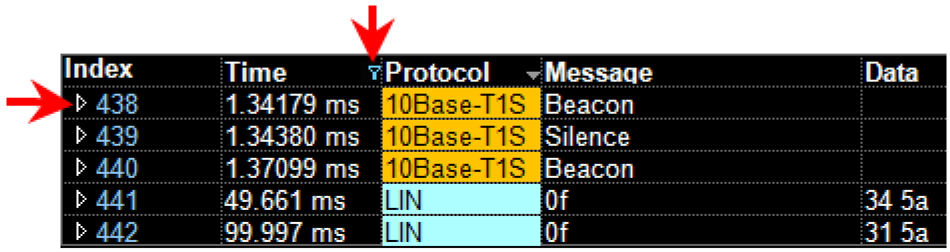
Select a filter **Operator** and enter a **Value** that satisfies the filter condition.

Operators	Data Types	Returns
=, ≠	Numeric or Text	Exact matches only
>, ≥, <, ≤	Numeric	All data that satisfies the operator
In Range, Out Range	Numeric	All data within/without range limits
Equals Any (on List), Does Not Equal Any (on List)	Text	All data that is/is not an exact match to any full value on the list. Enter a comma-delimited list of values, no spaces before or after the comma, although there may be spaces within the strings.
Contains, Does Not Contain	Text	All data that contains or does not contain the string

Note: Once the Operator is selected, the dialog shows the Value format that may be entered. Numeric values must be within .01% tolerance to be considered a match. Text values are case-sensitive, including spaces within the string.

Select **Enable** to turn on the column filter; deselect it to turn off the filter. Use the **Disable All** button to quickly turn off multiple filters. The filter settings remain in place and can be re-enabled on subsequent decodings.

Those columns of data that have been filtered will have a funnel icon (similar to Excel) in the header cell, and the index numbers will be colorized.



Index	Time	Protocol	Message	Data
438	1.34179 ms	10Base-T1S	Beacon	
439	1.34380 ms	10Base-T1S	Silence	
440	1.37099 ms	10Base-T1S	Beacon	
441	49.661 ms	LIN	0f	34 5a
442	99.997 ms	LIN	0f	31 5a

Example filtered decoder table.

On summary tables, only the Time, Protocol, and Status columns can be filtered.

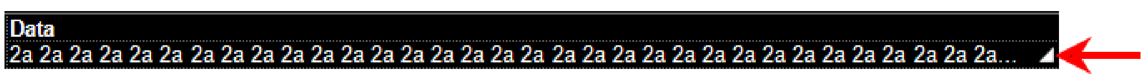
If you apply filters to a single decoder table, the annotation is applied to only that portion of the waveform corresponding to the filtered results, so you can quickly see where those results occurred. Annotations are not affected when a summary table is filtered.

Also, eye diagrams are modified to represent only the filtered results, which can help to identify exactly which indices of data are the cause of signal integrity problems.

View Details

When viewing a summary table, touch the **Index number** in the first column to drop-in the detailed decoding of that record. Touch the Index cell again to hide the details.

If there is more data than can be displayed in a cell, the cell is marked with a white triangle in the lower-right corner. Touch this to open a pop-up showing the full decoding.



Data
2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a 2a...

Navigate

In a single decoder table, touch the **Index column header** (top, left-most cell of the table) to open the Decode Setup dialog. This is especially helpful for adjusting the decoder during initial tuning.

When in a summary table, the Index column header cell opens the Serial Decode dialog, where you can enable/disable all the decoders. Touch the **Protocol** cell to open the Decode Setup dialog for the decoder that produced that index of data.

Customizing the Result Table

You may customize the size of the result table by changing the **Table # Rows** setting on the Decode Setup dialog. Keep in mind that the deeper the table, the more compressed the waveform display on the grid, especially if there are also measurements turned on.

Performance may be enhanced if you reduce the number of columns in the result table to only those you need to see. It is also especially helpful if you plan to export the data.

1. On the Decode Setup tab, touch the **Configure Table** button.
2. On the **View Columns** pop-up dialog, mark the columns you want to appear and clear those you wish to remove. Only those columns selected will appear on the oscilloscope display.

   Note: If a column is not relevant to the decoder as configured, it will not appear.

   To return to the preset display, touch **Default**.
3. Touch the **Close** button when finished.

You may also use the View Columns pop-up to set a **Bit Rate Tolerance** percentage. When implemented, the tolerance is used to flag out-of-tolerance messages (messages outside the user-defined bitrate +- tolerance) by colorizing in red the Bitrate shown in the table.

Exporting Result Table Data

You can manually export the detailed result table data to a .CSV file:

1. Press the Front Panel **Serial Decode button**, or choose **Analysis > Serial Decode**, then open the **Decode Setup tab**.
2. Optionally, touch **Browse** and enter a new **File Name** and output folder.
3. Touch the **Export Table** button.

Export files are by default created in the D:\Applications\<protocol> folder, although you can choose any other folder on the oscilloscope or any external drive connected to a host USB port. The data will overwrite the last export file saved, unless you enter a new filename.

Note: Only rows and columns displayed are exported. When a summary table is exported, a combined file is saved in D:\Applications\Serial Decode. Separate files for each decoder are saved in D:\Applications\<protocol>.

The Save Table feature will automatically create tabular data files with each acquisition trigger. The file names are automatically incremented so that data is not lost. Choose **File > Save Table** from the oscilloscope menu bar and select **Decodex** as the source.

Searching Decoded Waveforms

Touching the Action toolbar **Search button** button on the Decode Setup dialog creates a 10:1 zoom of the center of the decoder source trace and opens the Search subdialog.

Touching the **any cell** of the result table similarly creates a zoom and opens Search, but of only that part of the waveform corresponding to the index (plus any padding).

Tip: In summary table mode, touch any cell *other than* Index and Protocol to create the zoom.

Basic Search

On the Search subdialog, select what type of data element to **Search for**. These basic criteria vary by protocol, but generally correspond to the columns of data displayed on the detailed decoder result table.

Optionally:

- Check **Use Value** and enter the **Value** to find in that column. If you do not enter a Value, Search goes to the beginning of the next data element of that type found in the acquisition.
- Enter a **Left/Right Pad**, the percentage of horizontal division around matching data to display on the zoom.
- Check **Show Frame** to mark on the overlay the frame in which the event was found.

After entering the Search criteria, use the **Prev** and **Next** buttons to navigate to the matching data in the table, simultaneously shifting the zoom to the portion of the waveform that corresponds to the match.

The touch screen message bar shows details about the table row and column where the matching data was found.

Advanced Search

Advanced Search allows you to create complex criteria by using Boolean AND/OR logic to combine up-to-three different searches. On the Advanced dialog, choose the **Col(umns) to Search 1 - 3** and the **Value** to find just as you would a basic search, then choose the **Operator(s)** that represent the relationship between them.

Decoding in Sequence Mode

Decoders can be applied to Sequence Mode acquisitions. In this case, the index numbers on the result table are followed by the segment in which the index was found and the number of the sample within that segment: *index* (*segment-sample*).

Note: For some protocols, the Serial Trigger does not support Sequence Mode acquisitions, although you could still decode Sequence Mode acquisitions made using a different trigger type.

CAN Std	Time	Format	ID	IDE	RTR	DLC	Data
2 (2-1)	9.72882 ms	Std	0x400	0	0	2	6a 6b
3 (3-1)	19.7527 ms	Std	0x400	0	0	2	6a 6b
4 (4-1)	30.2558 ms	Std	0x400	0	0	2	6a 6b
5 (5-1)	40.1663 ms	Std	0x400	0	0	2	6a 6b
6 (6-1)	49.8284 ms	Std	0x400	0	0	2	6a 6b
7 (7-1)	59.8595 ms	Std	0x400	0	0	2	6a 6b
8 (8-1)	69.8913 ms	Std	0x400	0	0	2	6a 6b
9 (9-1)	80.4032 ms	Std	0x400	0	0	2	6a 6b
10 (10-1)	89.9384 ms	Std	0x400	0	0	2	6a 6b
11 (11-1)	99.9688 ms	Std	0x400	0	0	2	6a 6b

Example filtered result table for a sequence mode acquisition.

In the example above, each segment was triggered on the occurrence of ID 0x400, which occurred only once per segment, so there is only one sample per segment. The Time shown for each index in a Sequence acquisition is absolute time from the first segment trigger to the beginning of the sample segment.

Otherwise, the results are the same as for other types of acquisitions and can be zoomed, filtered, searched, or used to navigate. When a Sequence Mode table is filtered, the waveform annotation appears on only those segments and samples corresponding to the filtered results.

Note: Waveform annotations can only be shown when the Sequence Display Mode is Adjacent. Annotations are not adjusted when a Sequence Mode summary table is filtered, only the result table data.

Multiple decoders can be run on Sequence Mode acquisitions, but in a summary table, each decoder will have a first segment, second segment, etc., and there may be any number of samples in each. As in any summary table, the samples will be interleaved and indexed according to their actual acquisition time. So, you may find (3-2) of one decoder before (1-1) of another. Filter on the Protocol column to see the sequential results for only one decoder.

Improving Decoder Performance

Digital oscilloscopes repeatedly capture "windows in time". Between captures, the oscilloscope is processing the previous acquisition.

The following suggestions can improve decoder performance and enable you to better exploit the long memories of Teledyne LeCroy oscilloscopes.

Where possible, **decode Sequence Mode acquisitions.** By using Sequence mode, you can take many shorter acquisitions over a longer period of time, so that memory is targeted on events of interest.

Note: For some protocols, the Serial Trigger does not support Sequence Mode acquisitions, although you could still decode Sequence Mode acquisitions made using a different trigger type.

Parallel test using multiple oscilloscope channels. Up-to-four decoders can run simultaneously, each using different data or clock input sources. This approach is statistically interesting because multi-channel acquisitions occur in parallel. The processing is serialized, but the decoding of each input only requires 20% additional time, which can lessen overall time for production validation testing, etc.

Avoid oversampling. Too many samples slow the processing chain.

Optimize for analysis, not display. The oscilloscope has a preference setting (Utilities > Preference Setup > Preferences) to control how CPU time is allocated. If you are primarily concerned with quickly processing data for export to other systems (such as Automated Test Equipment) rather than viewing it personally, it can help to switch the Optimize For: setting to Analysis.

Decrease the number of rows and columns in tables. Only the result table rows and columns shown are exported. It is best to reduce tables to only the essential columns if the data is to be exported, as export time is proportional to the amount of data exchanged.

Serial Trigger

"T" options provide advanced serial data triggering in addition to decoding. Serial data triggering is implemented directly within the hardware of the oscilloscope acquisition system. The serial data trigger scrutinizes the data stream in real time to recognize "on-the-fly" the user-defined serial data conditions. When the desired pattern is recognized, the oscilloscope takes a real-time acquisition of all input signals as configured in the instrument's acquisition settings. This allows decode and analysis of the signal being triggered on, as well as concomitant data streams and analog signals.

Note: The trigger and decode systems are independent, although they are seamlessly coordinated in the user interface and the architecture. It is therefore possible to use the serial trigger without decoding the acquisition, or to decode acquisitions made without using the serial trigger.

Requirements

Serial trigger options require the appropriate hardware (please consult support), an installed option key, and the latest firmware release.

Restrictions

The serial trigger operates on only one protocol at a time. It is therefore impossible to express a condition such as "trigger on CAN frames with ID = 0x456 followed by LIN packet with Adress 0xEBC."

Linking Trigger and Decoder

A quick way to set up a serial trigger is to link it to a decoder by checking the **Link to Trigger** ("On") box on the Serial Decode dialog. Linking trigger and decoder allows you to configure the trigger with the exact same values that are used for decoding the signal (in particular the bit rate), saving the extra effort needed to re-enter values on the serial trigger set up dialogs.

While the decoder and the trigger have distinct sets of controls, when the link is active, a change to the bit rate in the decoder will immediately propagate to the trigger and vice-versa.

10Base-T1S Serial Trigger Setup

The 10Base-T1S serial trigger starts an acquisition upon finding the specified conditions in the trigger source input channel.

To access the serial trigger dialogs:

- Touch the **Trigger descriptor box** or choose **Trigger > Trigger Setup** from the Menu Bar.
- On the Trigger dialog, touch the **Serial** Type button, then the **10Base-T1S** Standard button.

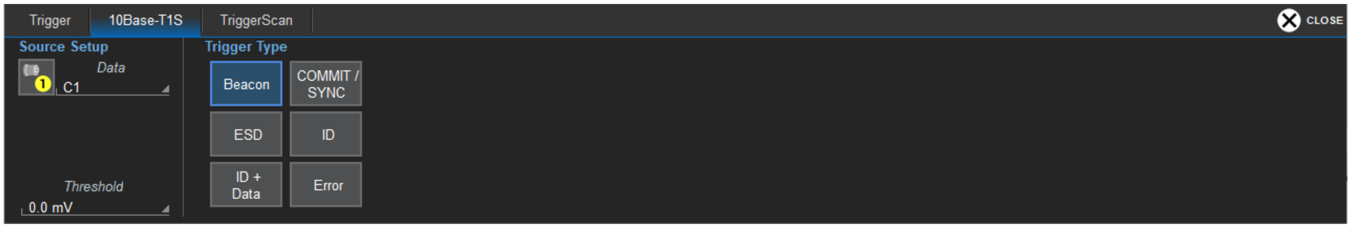
Working from left to right, make the desired selections from the trigger setup dialog.

Source Setup

In **DATA**, select the data source input channel.

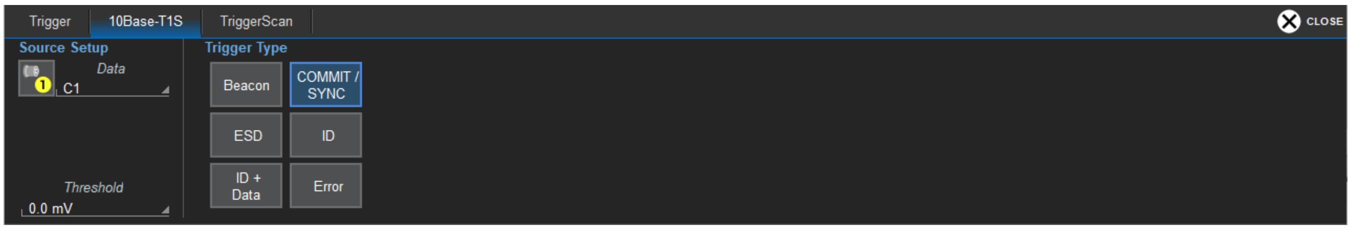
Use the **Threshold** control to adjust the vertical level for the trigger. Much like an Edge trigger, you must specify the level at which to process the incoming signal to determine whether the serial data pattern meets the trigger condition.

Beacon Trigger



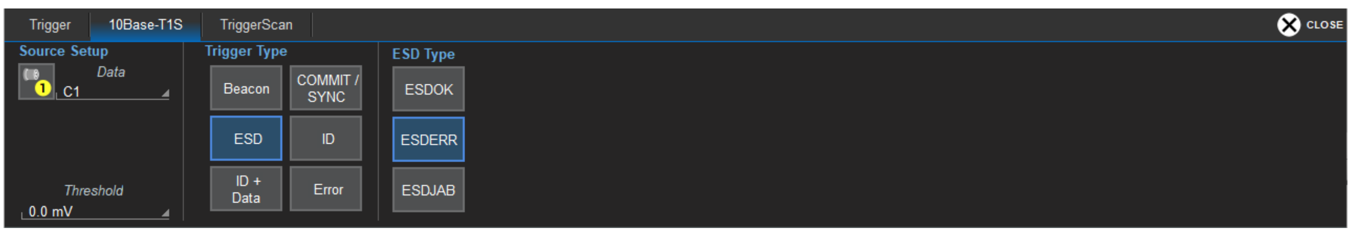
Select **Beacon** to trigger on the next packet sent from the Beacon (node ID0).

COMMIT/SYNC Trigger



Select **COMMIT/SYNC** to trigger on the next commit packet sent from any node.

ESD Trigger

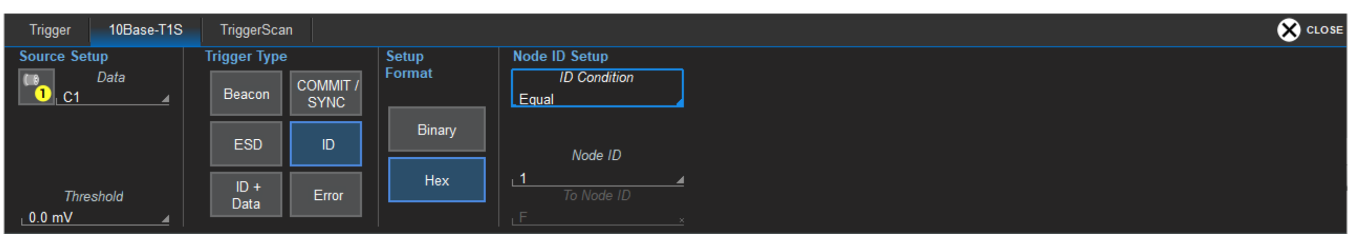


Select ESD to trigger on any of three End of Stream Delimiter conditions:

- **ESDOK** triggers on the next good ESD symbol.
- **ESDERR** triggers on the next ESD error.
- **ESDJAB** triggers on the next ESD Jabber symbol.

Select **ESD** and the trigger condition.

ID Trigger



ID is used to trigger on the next packet from either a specific node ID or any value relative to a reference ID (e.g., greater than *x*).

Setup Format

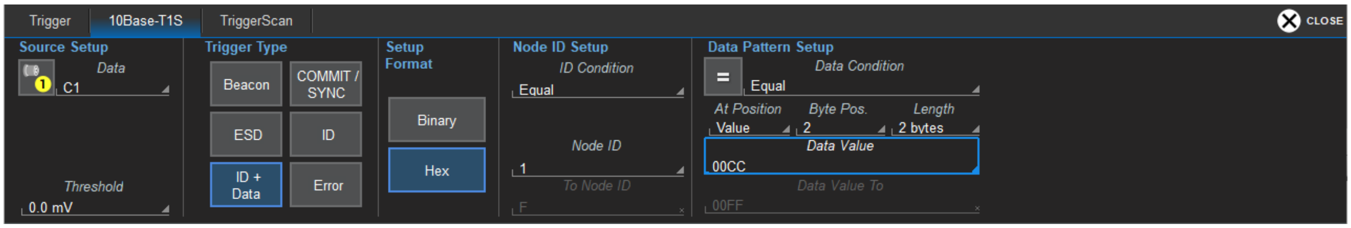
Choose to enter trigger values in Binary or Hex(adecimal) format.

Node ID Setup

1. Choose the ID Condition (Boolean operator) that describes the relationship to the Node ID value that will fire the trigger. To use a range of values, choose In Range or Out Range.

2. Enter the Node ID value. When setting a range, also enter the stop value in To Node ID.

ID + Data Trigger



In addition to defining the node ID fields (above), create a condition statement that describes the Data field pattern upon which to trigger. This condition is added to the ID condition.

1. Choose the Data Condition (Boolean operator) that describes the relationship to the reference Data Value. To use a range of values, choose In Range or Out Range.

2. Use At Position, Byte Pos. and Length to define the location and length of the value to be found in the message data field.

   - An At Positionof Don't Care will trigger upon finding the pattern anywhere in the data field. Choose Value to mark a specific Byte Pos(ition) in the Data field the matching Data Value must occupy.

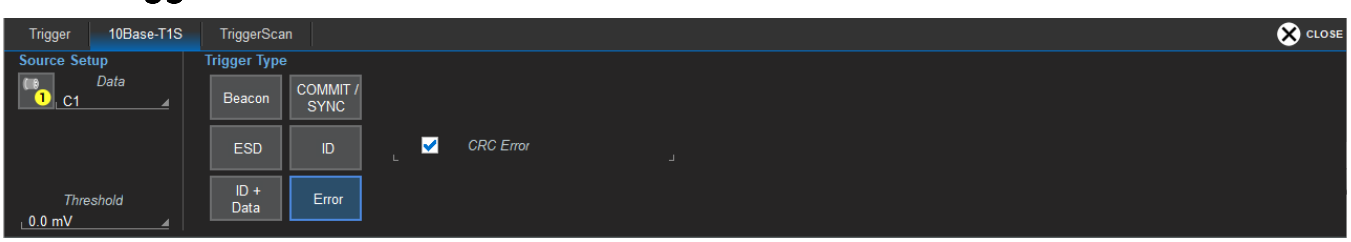
   - The Byte Pos. is the byte number at which the pattern starts.

   - The total Length of the pattern can be from 1 to 12 bytes.

3. Enter the reference Data Value to appear at the specified location. When using a range, enter the start Data Value. For Hexadecimal format values, if desired, you can precede the ID value with 0x, but this is not necessary.

   When using a range, enter the stop value in Data Value To.

Error Trigger



Choose Error to trigger on the next protocol error detected. Check CRC error to include | limit the trigger to Cyclic Redundancy Check errors.

Using the Decoder with the Trigger

A key feature of Teledyne LeCroy trigger and decode options is the integration of the decoder functionality with the trigger. While you may not be interested in the decoded data per se, using the decoded waveform can help with understanding and tuning the trigger.

Stop and Look

Decoding with repetitive triggers can be very dynamic. Stop the acquisition and use the decoder tools such as Search, or oscilloscope tools such as TriggerScan, to inspect the waveform for events of interest. Touch and drag the paused trace to show time pre- or post-trigger.

Optimize the Grid

The initial decoding may be very compressed and impossible to read. Try the following:

- Increase the height of the trace by *decreasing* the gain setting (V/Div) of the decoder source channel. This causes the trace to occupy more of the available grid.
- Change your Display settings to turn off unnecessary grids. The Auto Grid feature automatically closes unused grids. On many oscilloscopes, you can manually move traces to consolidate grids.
- Close setup dialogs.

Use Zoom

The default trigger point is at zero (center), marked by a small triangle of the same color as the input channel at the bottom of the grid. Zoom small areas around the trigger point. The zoom will automatically expand to fit the width of the screen on a new grid. This will help you to see that your trigger is occurring on the bits you specified.

If you drag a trace too far left or right of the trigger point, the message decoding may disappear from the grid. You can prevent "losing" the decode by creating a zoom of whatever portion of the decode interests you. The zoom trace will not disappear when dragged and will show much more detail.

Saving Trigger Data

The message decoding and the result table are dynamic and will continue to change as long as there are new trigger events. As there may be many trigger events in long acquisitions or repetitive waveforms, it can be difficult (if not impossible) to actually read the results on screen unless you stop the acquisition. You can preserve data concurrent with the trigger by using the **AutoSave** feature.

- AutoSave Waveform creates a .trc file that copies the waveform at each trigger point. These files can be recalled to the oscilloscope for later viewing. Choose **File > Save Waveform** and an Auto Save setting of **Wrap** (overwrite when drive full) or **Fill** (stop when drive full). The files are saved in D:\Waveforms.
- AutoSave Table creates a .csv file of the result table data at each trigger point. Choose **File > Save Table** and an Auto Save setting of **Wrap** or **Fill**. The files are saved in D:\Tables.

Caution: If you have frequent triggers, it is possible you will eventually run out of hard drive space. Choose Wrap only if you're not concerned about files persisting on the instrument. If you choose Fill, plan to periodically delete or move files out of the directory.

Measure/Graph

The installation of the Measure/Graph package (included with "ME" and "MP" options) adds a set of measurements and plots designed for serial data analysis to the oscilloscope's standard measurement capabilities. Measurements can be quickly applied without having to leave the waveform or tabular views of the decoding.

Note: This capability will only function properly if an "ME" or "MP" option for the protocol decoded is installed, although the dialogs will appear if any Measure/Graph options are installed.

Serial Data Measurements

These measurements designed for debugging serial data streams can be applied to the decoded waveform. Measurements appear in a tabular readout below the grid (the same as for any other measurements) and are in addition to the result table that shows the decoded data. You can set up as many measurements as your oscilloscope has parameter locations.

Note: Measurements appear in the Serial Decode sub-menu of the Measure Setup menu and may have slightly different names. For example, the CAN sub-menu has measurements for CAN to Value instead of Message to Value, etc. The measurements are the same.

Measurement	Filters	Description
View Serial Encoded Data as Analog Waveform		Simplified set up of a Message to Value parameter and graph. Performs a Digital-to-Analog Conversion (DAC) of the embedded digital data and displays it as an analog waveform.
Message to Value	ID, Value	Extracts a selected portion of the decoded data to a measurement parameter location, with optional conversion of value. Data may be selected by ID and/or data field position.
Message to Analog	ID, Data, Analog	Computes time from start of first message that meets conditions to crossing threshold on an analog signal. If the analog condition precedes the message condition, no measurement is performed.
Message to Message	ID, Data	Computes time from start of first message that meets conditions to start of the next message that meets conditions.
Time at Message	ID, Data	Computes time from trigger to start of each message that meets conditions.
Analog to Message	ID, Data, Analog	Computes time from crossing threshold on an analog signal to start of first message that meets conditions. If the message condition precedes the analog condition, no measurement is performed.
Delta Messages	ID, Data	Computes time difference between two messages on a single decoded line.
Bus Load	ID, Data	Computes the load of selected messages on the bus (as a percent).
Message Bitrate	ID, Data	Computes the bitrate of selected messages within the decoded stream.
Number of Messages	ID, Data	Computes the total number of messages in the decoding that meet conditions.

Graphing Measurements

The Measure/Graph package include simplified methods for plotting measurement values as:

- Histogram - a bar chart of the number of data points that fall into statistically significant intervals or bins. Bar height relates to the frequency at which data points fall into each interval/bin. Histogram is helpful to understand the modality of a parameter and to debug excessive variation.
- Trend - a plot of the evolution of a parameter over time. The graph's vertical axis is the value of the parameter; its horizontal axis is the order in which the values were acquired. Trending data can be accumulated over many acquisitions. It is analogous to a chart recorder.
- Track - a time-correlated accumulation of values for a single acquisition. Tracks are time synchronous and clear with each new acquisition. Track can be used to plot data values and compare them to a corresponding analog signal, or to observe changes in timing. A parameter tracked over a long acquisition could provide information about the modulation of the parameter.

To graph a measurement, just select the plot type from the Measure/Graph dialog when setting up the measurement. All plots are Math functions that open along side the deocoding in a separate grid.

Measure/Graph Setup Dialog

Use the Measure/Graph Setup dialog to apply serial data measurement parameters to the decoded waveform and simultaneously graph the results. This dialog appears behind the Decode Setup dialog and is active when measurements are supported.

1. Select the **Measurement** to apply and the **Destination** parameter (P*n*) to which to assign it.
2. The active decoder is preselected in **Source 1**, indicating the measurement will be applied to the decoder results; change it if necessary. If the measurement requires it, also select an appropriate Source 2 (such as an analog waveform for comparison).
3. Optionally:
   - Touch Graph to select a plot type. Also select a **Destination** function (F*n*) for the plot.
   - Touch **Apply & Configure** to set a filter, gate or other qualifiers on the measurement.

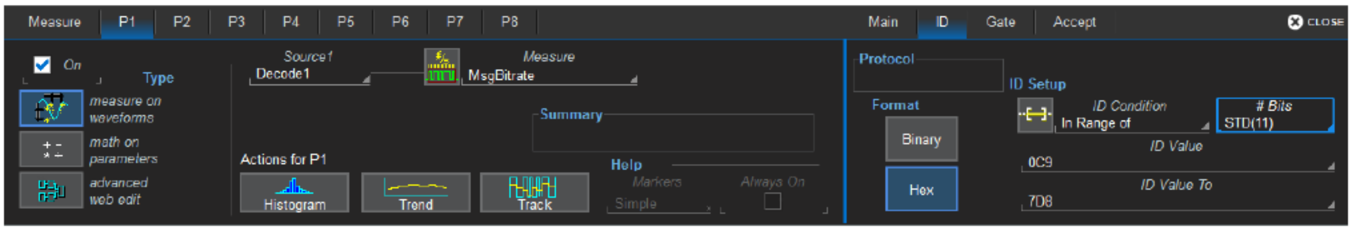
Filtering Measurements

Certain serial decode measurements can be filtered to include only the results from specified IDs or specific data patterns. As with all measurements, you can set a gate to restrict measurements to a horizontal range of the grid corresponding to a specific time segment of the acquisition.

After creating a measurement on the Measure/Graph Setup dialog, touch **Apply&Configure**. The touch screen display will switch to the standard Measure setup dialogs for the parameter you selected. Set filter conditions on the right-hand subdialogs that appear next to the P*n* dialogs.

ID Filter

This filter restricts the measurement to only frames/packets with a specific ID value. Settings on this dialog may change depending on the protocol.

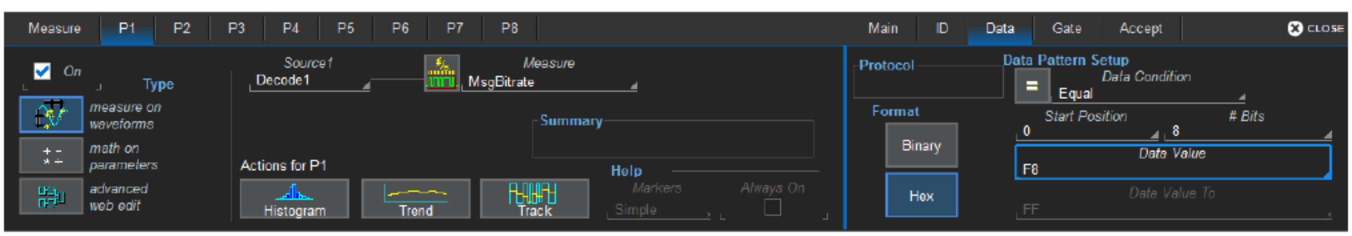


1. On the Main subdialog, choose to **Filter** by **ID** or **ID + Data**.
2. On the **ID subdialog**, choose to enter the ID in **Binary** or **Hex**(adecimal) format.
3. If the field appears, select the **# Bits** used to define the frame ID. (This will change the ID Value field length.)
4. Using the **ID Condition** and **ID Value** controls, create a condition statement that describes the IDs you want included in the measurement. To set a range of values, also enter the **ID Value To**.

> **Tip:** On the value entry pop-up: use the arrow keys to position the cursor; use Back to clear the previous character (like Backspace); use Clear to clear all characters.

Data Filter

This restricts measurements to only frames containing extracted data that matches the filter condition. It can be combined with a Frame ID filter by choosing **ID+Data** on the Main subdialog.



Use the same procedure as above to create a condition describing the **Data Value(s)** to include in the measurement. Use "X" as a wild card ("Don't Care") in any position where the value doesn't matter.

Optionally, enter a **Start Position** within the data field byte to begin seeking the pattern, and the **# Bits** in the data pattern. The remaining data fields positions will autofill with "X".

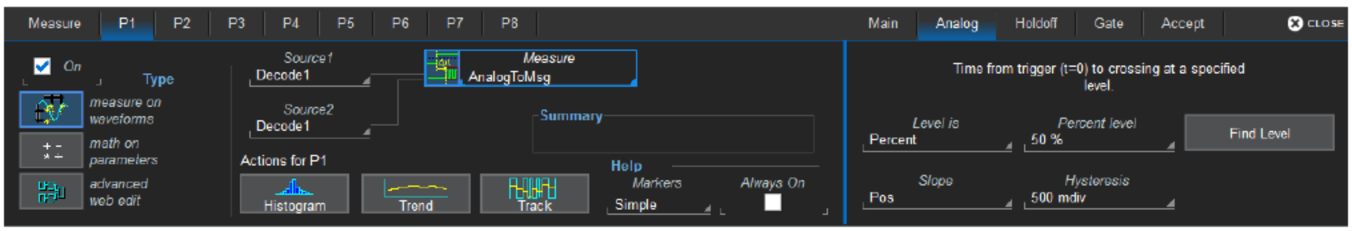
Note: For MsgtoMsg measurements, the data condition is entered twice: first for the Start Message and then for the End Message. The measurement computes the time to find a match to each set of conditions.

Analog Settings

The measurements AnalogToMsg and MsgToAnalog allow you to use crossing level and slope to define the event in the Analog waveform that is to be used as the reference for the measurement.

As with the decoder, Level may be set as a percentage of amplitude (default), or as an absolute voltage level by changing **Level Is** to Absolute. You can also use **Find Level** to allow the oscilloscope to set the level to the mean Top-Base amplitude.

A **Slope** and **Hysteresis** selection is also offered. The width of the Hysteresis band is specified in milli-divisions. See Setting Level and Hysteresis for more information on using these controls.



Digital to Analog Conversion

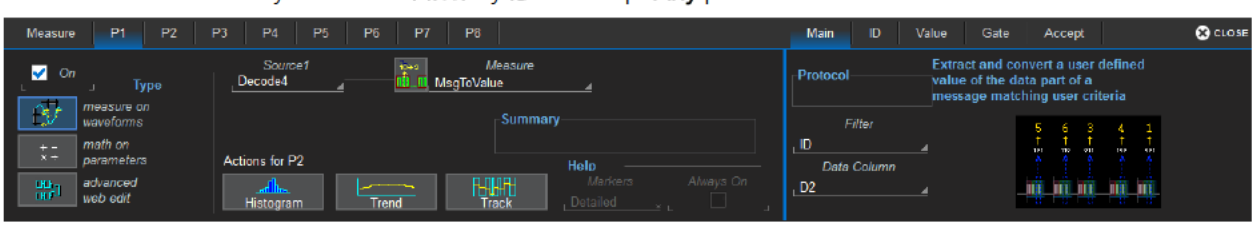
These serial data measurements enable you to take a subset of decoded data (such as sensor data payload) and plot it as a graph. The track of the these measurements is, in effect, a Digital to Analog Converter (DAC) that can display digitally-encoded sensor data as an analog waveform. They are particularly useful for symbolic and higher-level decodings.

Message to Value

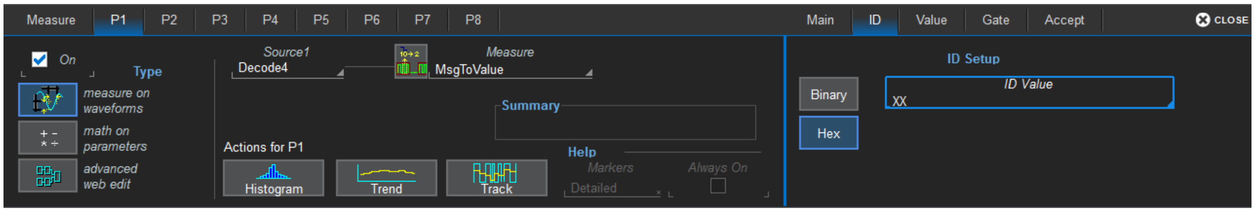
Message to Value enables you to apply oscilloscope features to a subset of the result table and is aimed at protocols with addressed packets containing varying types of data, like CAN, LIN, MIL1553 and many others. With it, you can filter the table by a particular ID to extract and convert decoded data values into a parameter that can be used for other math or measurement processes, in particular the Track function.

Message to Value requires several filter selections from the parameter set up subdialogs:

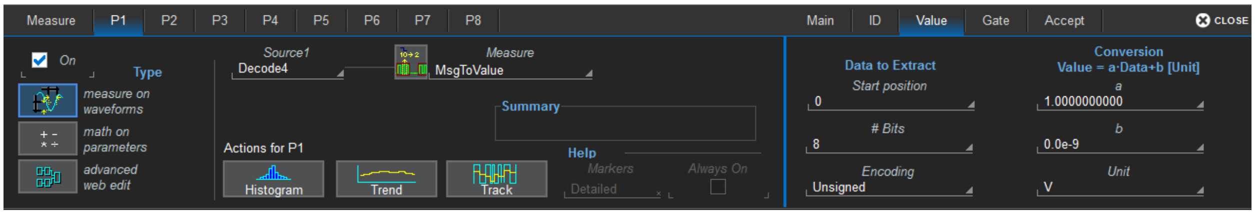
- Choose whether or not you wish to **Filter** by **ID** or accept **Any** packets.



- If you are filtering by ID, enter the desired ID on the ID subdialog.



- On the Value subdialog, enter the **Data to Extract** and any **Conversion** to be made.



Follow these steps to define the values to extract:

1. Under Data to Extract, begin by entering the **Start position** and the **# Bits** to extract.
2. Choose the **Encoding** type if the signal uses encoding, otherwise leave it Unsigned.
3. Under Conversion, enter the **a. Coefficient** and **b. Term** that satisfy the formula:
   *Value = Coefficient * Raw Value + Term*.
4. Optionally, enter a **Unit** for the extracted decimal value.

View Serial Encoded Data as Analog Waveform

This is an alternative set up method for Message to Value, which preselects for the Track graph. You can add filters the same as for Message to Value by clicking Apply and Configure.

Eye Diagrams

The "ME" and "MP" options provide easy eye diagram setup and eye mask testing.

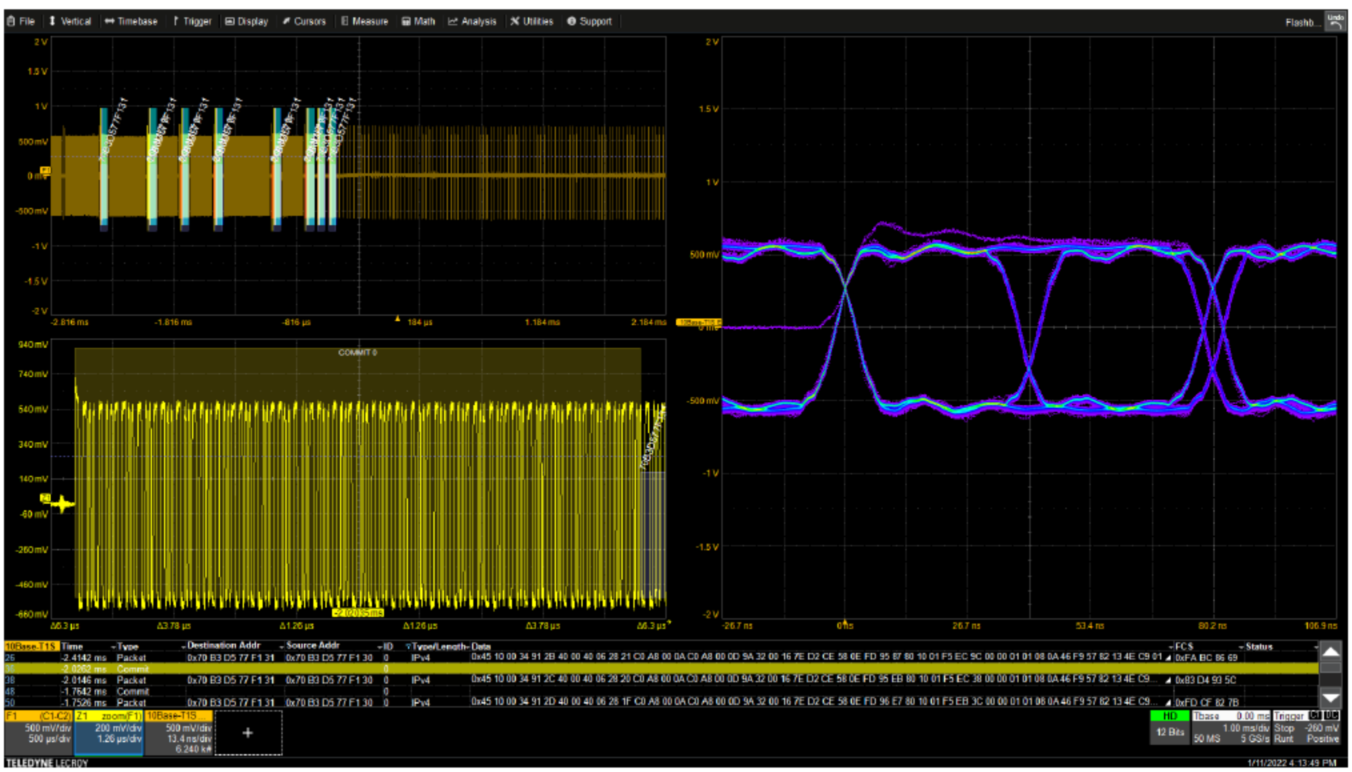
Note: This capability will only function properly if an "ME" or "MP" option for the protocol decoded is installed, although the dialogs will appear if any Measure/Graph options are installed.

Eye diagrams are a key component of serial data analysis. They are used both quantitatively and qualitatively to understand the quality of the signal communications path. Signal integrity effects such as intersymbol interference, loss, crosstalk and EMI can be identified by viewing eye diagrams, such that the eye is typically viewed prior to performing any further analysis.

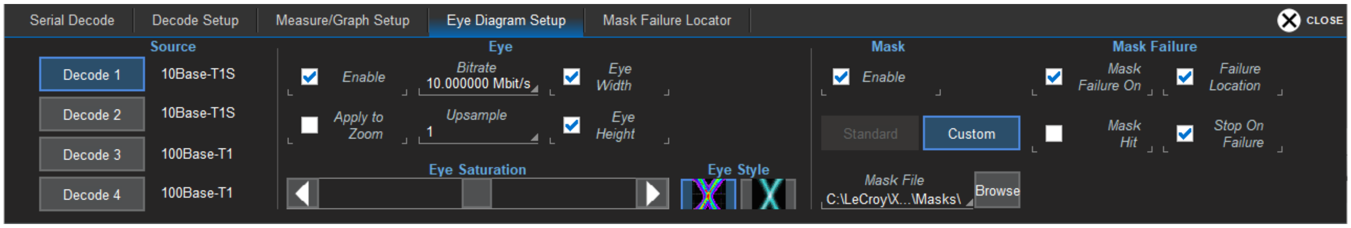
Each pixel in the eye takes on a color that indicates how frequently a signal has passed through the time and voltage specified for that pixel. The eye diagram shows all values a digital signal takes on during a bit period. A bit period (also referred to as unit interval, or UI) is defined by the data clock, whether explicit or extrapolated depending on the protocol.

Eye diagrams show the acquired signal that is currently being shown on the decoder result table. They are not persistent, as are eye diagrams generated in some other serial data analysis software; the eye will change from one acquisition to the next and when the result table is filtered. Our recommended approach for using the eye diagrams is to:

- Make single acquisitions with decoder and eye diagram enabled to test both are working correctly.
- Make a normal acquisition with Mask Testing and Stop On Failure enabled in the Mask Failure Locator, or with a Pass/Fail test set on one of the eye parameters.



Eye Diagram Setup Dialog



Create Eye Diagram

Open the **Eye Diagram Setup** dialog and select the **Decode** for which to create an eye diagram.

Under Eye, check **Enable** to display the eye diagram.

Check **Apply to Zoom** to eye diagram only the zoomed section of the source decode waveform. Eye measurements will also reflect only this zoomed section.

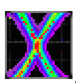
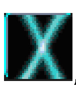
The **Bitrate** is automatically read from the decoder setup. This value is linked to the decoder bit rate setting, and changing it in either place will update both settings.

The **Upsample** factor increases the number of sample points used to compose the eye diagram. Increase from 1 to a higher number (e.g. 5) to fill in gaps. Gaps can occur when the bitrate is extremely close to a submultiple of the sampling rate, such that the sampling of the waveform does not move throughout the entire unit interval. Gaps can also occur when using a record length that does not sample a sufficiently large number of unit intervals.

Choose to display the **Eye Height** or **Eye Width** measurement parameters. These are added to the Measure table in the first open parameter slots.

Check **Auto Scale** to rescale the eye diagram to fit the entire grid.

The **Eye Style** may utilize color-graded or analog persistence:

- With **color-graded** persistence , pixels are given a color based on the pixel's relative population and the selected Eye Saturation. The color palette ranges from violet to red.
- With **analog** persistence , the color used mimicks the relative intensity that would be seen on an analog oscilloscope.

Use the **Eye Saturation** slider to adjust the color grading or intensity. Slide to the left to reduce the threshold required to reach saturation.

Eye Mask Test

Under Mask, check **Enable** to turn on eye mask testing.

Select to use either a **Standard** or **Custom** mask, then either select the **Standard Mask** or **Browse** to and select your custom **Mask File**.

Tip: Masks previously created on the instrument are stored in D:\Masks. For ease of selection, copy other .msk files to this location.

Check **Mask Failure On** to mark the parts of the eye diagram that fail the mask test. Mask violations appear as red failure indicators where the eye diagram intersects the mask.

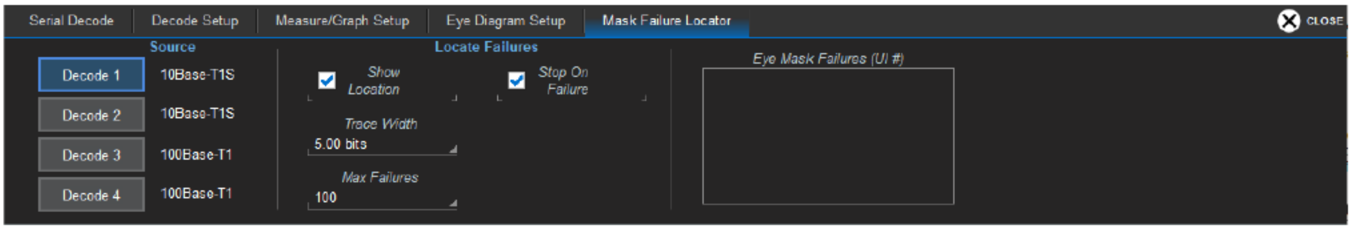
Check **Failure Location** to display the Mask Failure Locator dialog.

Check **Mask Hit(s)** to display the number of hits in the Measure table.

Check **Stop on Failure** to stop acquisition when a mask violation occurs.

Mask Failure Locator Dialog

Use this dialog to quickly search the acquisition for eye diagram mask test failures.



Check **Failure Location** to mark the failure on the source trace. Check **Stop On Failure** to stop acquisition whenever an eye mask failure occurs.

In **#UI Displayed**, enter the number of UIs surrounding the mask violation to display as "padding."

Enter the **Max Failures** to retain in the Eye Mask Failure list.

Select from the **Eye Mask Failure** list to mark and zoom to the location of that failure. Yellow circles appear over the red failure indicators to show the location of the failure.

Appendix A: Automating the Decoder

As with all other oscilloscope settings, decoder features such as result table configuration and export can be configured remotely using COM Automation.

Note: The examples shown here were taken from a CAN FD decoding, but all decoder result tables share the same Automation structure.

Configuring the Decoder

The object path to the decoder Control Variables (CVARs) is:

app.SerialDecode.Decode*n*

Where *n* is the decoder number, 1 to 4. All relevant decoder objects will be nested under this. Use the MAUI Browser utility (installed on the oscilloscope desktop) to view the entire object hierarchy.

Accessing the Result Table

The decoder Result Table is a complex matrix with secondary tables nested within some of its cells. The table data can be accessed using the Automation object:

app.SerialDecode.Decode*n*.out.Result.cellvalue(*RowA*, *ColA*)(*RowB*, *ColB*)

Where:

n:= 1 to 4

RowA:= 0 to K (0=Row Index Number)

ColA:= 0 to L (0=Column Header)

RowB:= 0=MeasuredValue, 1=StartTime, 2=StopTime

ColB:= 0 to M

Complicating the matter of accessing the table is that there are two types of cell that may appear in the Result table, Simple Cell and Table Cell, which are accessed in slightly different ways, and that some columns are always hidden from view, yet they are still counted among the columns when querying.

Reading the Structure of the Result Table

In order to successfully access the data, it is necessary to first ascertain how many rows and columns are actually in your decoder result table, and what cell type is used for the column of data you wish to read.

To do this, we have provided the script, **ExampleTableSerialDecode.vbs**, which by default installs into oscilloscope: C:\LeCroy\XStream\Scripts\Automation\ExampleTableSerialDecode.vbs.

Tip: This script may also be used as a basis for your own remote control programs, or used as is to read decoder table data.

With the decoder table populated, run the script from the oscilloscope (or a PC if you have a remote connection to the oscilloscope). The script will generate the comma-delimited file, **ExampleTableSerialDecode.txt**, which may be imported into Excel or other spreadsheet software to show the table structure.

Result.Rows: 8												
Result.Columns: 34												
CAN FD	Time	Format	PRIO	ID	SRC	IDE	FDF	BRS	ESI	RTR	DLC	Data
1	-7.48E-03	FD		31;-7.48024976		0;-7.4522	1;-7.4502	1;-7.4462	0;-7.44455189402753		6;-7.44405	174;-7.44205462545681E-03;-7.43805466420848E-03;143;-7.43805466420848E-03;-7.43405452651836E-03;
2	-4.59E-03	FD		190;-4.5894578		0;-4.5634	1;-4.5614	1;-4.5574	0;-4.55575966881187		8;-4.5552	0;-4.55326066828695E-03;-4.54826025140856E-03;0;-4.54826025140856E-03;-4.54376023553057E-03;0;-4.
3	-4.48E-03	FD		614;-4.4741593		0;-4.4501	1;-4.4481	1;-4.4441	0;-4.44245973215831		6;-4.44195	0;-4.43996012906669E-03;-4.43546015204063E-03;0;-4.43546015204063E-03;-4.43046028360746E-03;0;-4.
4	-4.37E-03	FD		44;-4.37086003		0;-4.3448	1;-4.3428	1;-4.3388	0;-4.33716034438709		8;-4.33666	0;-4.33466132067482E-03;-4.32966079442937E-03;0;-4.32966079442937E-03;-4.32516075451995E-03;0;-4.
5	-2.77E-05	Std		325;-2.5744779		0;-1.7452	0;2.54715560793993E-07;2.27			0;-3.74528	8;2.27402	69;1.02543932013556E-05;2.62548705473216E-05;80;2.62548705473216E-05;4.22739967647582E-05;128;4
6	2.58E-03	FD		31;2.586421380		0;2.61441	1;2.61641	1;2.6204	0;2.62211966414495E		6;2.62261	128;2.62461650529113E-03;2.62911896054307E-03;72;2.62911896054307E-03;2.63311895907048E-03;97;2
7	5.35E-03	FD		44;5.355213390		0;5.38121	1;5.38320	1;5.3872	0;0.00538891140717		8;5.38940	0;5.39141045889392E-03;5.39641119645398E-03;0;5.39641119645398E-03;5.40091119496943E-03;0;5.400

Example spreadsheet after importing ExampleTableSerialDecode.txt.

The first two rows of the imported file will show the total number of rows and columns in the table, in this example 8 rows and 34 columns. This indicates the range of your *RowA* and *ColA* keys.

The third row of the imported file will replicate the column headers of the Result Table (0), with individual records (frames, messages, etc., depending on how you have "packetized" the decoding) appearing in subsequent rows (1-*n*).

Counting from 0 at the far left (Row Index Number), find the column of the data you wish to access. That will be the *ColA* key in your script.

Note: Do not confuse the number/letter of the cells in the imported file with the rows/columns of the Result Table.

Hidden columns (whether hidden by you or the software) must still be counted, so, in the example above, PRIO is column 3, making ID column 4, and so forth. So, if you wished to access the ID of record 6, the first argument of your query would be: (6,4)

Within each column, Simple Cells contain a single value that appears at the specified location in the table. In the above example, columns 0 through 2 are Simple Cells. Simple Cell VBS access syntax is:

vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(*RowA*,*ColA*)'

However, many cells of the Result Table are the Table Cell type, nested tables that may contain multiple "B" columns and always three "B" rows that, when coupled with the column key, each return a different component of the measurement: (0,*ColB*) = MeasuredValue, (1,*ColB*)= StartTime, (2,*ColB*) = StopTime. These cells can be identified by the list of semi-colon delimited values within them. The first three values in the list are Col0, the second three values are Col1, and so forth.

To access Table Cells, the (*RowB*,*ColB*) argument is sent in a second parenthesis, following the A "locators":

vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(*RowA*,*ColA*)(*RowB*, *ColB*)'

Although the image above does now show it, the ID and IDE columns each contain a single-column, three-row nested table. To read the *values* from such columns, you would add the argument (0,0) following your "locators": (*RowA*,4),(0,0) and (*RowA*,6),(0,0) respectively.

Reading the Data column (*RowA*,12) is more complicated, because it contains a *multi-column*, three-row nested table, as indicated by the longer list of values. To access the full Data column value for each record, all *ColB*s must be called by your script.

For example, if these were your decoder results: