TELEDYNE LECROY
Everywhereyoulook™

SENTbus

Trigger, Decode, Measure/ Graph and Eye Diagram

SENTbus TDME Instruction Manual

February, 2020
sent-tdme-im-eng_04feb20.pdf

Contents

About This Manual

Teledyne LeCroy offers a wide array of toolsets for decoding and debugging serial data streams. These toolsets may be purchased as optional software packages, or are provided standard with some oscilloscopes.

This manual explains the basic procedures for using serial data decoder and trigger software options. There are also sections pertaining to the measure and graphing capabilities and eye diagram tests.

While some of the images in this manual 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, as much functionality is shared. Product-specific exceptions will be noted in the text.

It is assumed that you have a basic understanding of the serial data physical and protocol layer specifications, and a basic understanding of how to use an oscilloscope, specifically the Teledyne LeCroy oscilloscope on which the option is installed. Only features directly related to serial data triggering and decoding are explained in this manual.

Teledyne LeCroy is constantly expanding coverage of serial data standards and updating software. Some capabilities described in this documentation may only be available with the latest version of our firmware. You can download the free firmware update from:

teledynelecroy.com/support/softwaredownload

About the SENTbus Options

The Teledyne LeCroy SENTbus TD and TDME options are tools aimed at decoding SENT (Single Edge Nibble Transmission) streams emitted by various sensors. The decoder supports the 2008, 2010 and 2016 SENT specifications.

SAE J2716 SENT protocol is a point-to-point scheme for transmitting signal values from a sensor to a controller. It is intended to allow for high-resolution data transmission with a lower system cost than other available serial data solutions. (SENT Protocol, wikipedia.org, accessed 03-06-2013).

The SENT SPC (Short PWM Code) protocol, or synchronized SENT, is a half-duplex variant of the SENT protocol which allows a microcontroller to query multiple sensors over the same wire. The microcontroller initiates the transfer of the data from the sensor to itself by emitting a Master Trigger Pulse (MTP). The sensors respond when queried rather than continuously broadcast as they would in the traditional SENT protocol.

The SENTbus trigger and decode (-TD) option enables you to decode and filter SENT and SENT SPC serial data streams by frame IDs, data patterns, CRCs, Master Trigger Pulse lengths, or protocol errors in . You can also trigger acquisition upon the occurrence of selected SENT protocol errors; start of the next slow channel, fast channel or any type of message; and IDs or data patterns within slow or fast channel messages. Conditional filtering at different levels enables you to target the trigger to a single message or a range of matching data.

Note: The current trigger implementation supports only SENT, not SENT SPC.

The SENTbus trigger, decode, measure/graph and eye diagram option (-TDME) 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 algorithms described here at a high level are used by all Teledyne LeCroy serial decoders sold for oscilloscopes. They differ slightly between serial data signals that have a clock embedded in data and those with separate clock and data signals.

Bit-level Decoding

The first software algorithm examines the embedded clock for each message based on a default or user-specified vertical threshold level. Once the clock signal is extracted or known, the algorithm examines the corresponding data signal at the predetermined vertical level to determine whether a data bit is high or low. The default vertical level is set to 50%, as determined from a measurement of peak amplitude of the signals acquired by the oscilloscope. For most decoders, it can also be set to an absolute voltage level, if desired. 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 during a long acquisition (many hundreds or thousands of serial data messages) or a short acquisition (one serial data message acquisition). In the case of the longest acquisition, only the most important information is highlighted, whereas in the case of 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, an input source, and a clock source (if necessary) to one of the four decoder panels using the Serial Data and Decode Setup dialogs.
- Complete the remaining dialogs 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. Connect your data and strobe/clock lines (if used) to the oscilloscope.
2. Set up the decoder using the lowest level decoding mode available (e.g., Bits or Nibbles).
3. Acquire a sufficient burst of relevant data. The data burst should be 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. A burst might contain at most 100000 transitions, or 32000 bits/1000 words, whichever occurs first. This is more a safety limit for software engineering reasons than a limit based on any protocol. We recommend starting with much smaller bursts.

4. Stop the acquisition, then run the decoder.
5. Use the various decoder tools to verify that transitions are being correctly decoded. Tune the decoder settings as needed.
6. Once you know you are correctly decoding transitions in one mode, continue making small acquisitions of five to eight bursts and running the decoder in higher level modes (e.g., Words). The decoder settings you verify on a few bursts will be reused when handling many packets.
7. 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 set up and tuning process, provided the basic signal characteristics do not change.

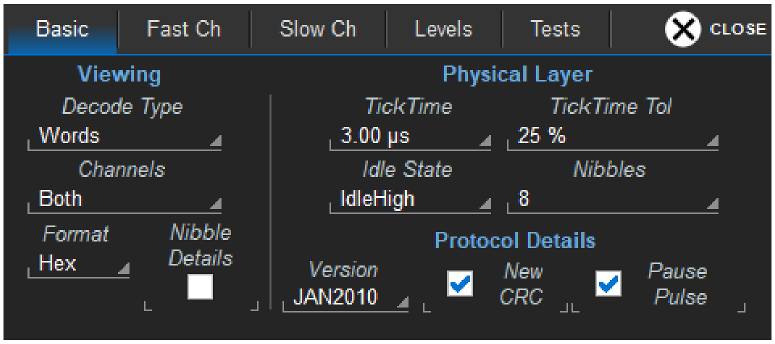
Decoder Set Up

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. Open the **Decode Setup** dialog.
2. From the buttons at the left, select the **Decode #** to set up.
3. Select the **data source (Src 1)** to be decoded and the **Protocol** to decode.
4. If required by the protocol, also select the **Strobe** or **Clock** source. (These controls will simply not appear if not relevant.)
5. Define the bit- and protocol-level decoding on the subdialogs next to the Decode Setup dialog.

Tip: After completing setup for one decoder, you can quickly start setup for the other decoders by using the buttons at the left of the Decode Setup dialog to change the Decode # .

SENT Basic Subdialog



Select a **Decode Type** of Nibbles or Words. A separate row will appear in the decode result table for each occurrence of your selection.

Choose to decode Fast Only, Slow Only, or Both **Channels**.

Choose **Format** in which to show results: Hex(adecimal) or Dec(imal) .

Check **Nibble Details** if you wish to view the raw nibble values before rounding. Values are normally rounded off to the nearest whole number. This selection will insert an individual row for each nibble in the message beneath the word decoding in the SENT result table.

Enter the **Tick Time**—time in seconds between a nibble of value N and a nibble of value N+1.

Set a **Tick Time Tol(erance)**. This defines how the CAL pulse is filtered with respect to the Tick Time. For example, if a Tick Time of 3 µs is set with a tolerance of 10%, the CAL pulse is expected to be 56 * 3 µs ± 10%, therefore 168 µs ± 10%.

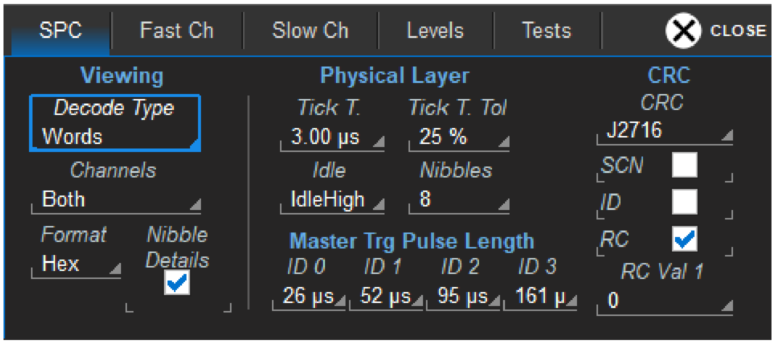
Indicate where the **Idle State**(opposite of the pulse direction) lies , IdleLow or IdleHigh.

Enter the number of **Nibbles** that make up a word, from 5 to 8 (8 is the default).

SENT decoding: Choose the **Version** of the protocol used in the decoded signal. If you are using the Jan 2010 or April 2016 version, also choose to enable or disable the following Protocol Details:

- **New CRC**—the CRC computation will be performed as per the 2010 or 2016 recommended implementation.
- **Pause Pulse**—the algorithm expects a Pause Pulse as per the 2010 section 5.2.6 definition. The Pause Pulse follows the CRC of message N and precedes the CAL pulse of message N+1.

SENT SPC Subdialog



In addition to the above settings, enter the **Master Trigger Pulse Length(s)** that initiate the transfer of data from up to four sensors in ID 0 through ID 3. This will enable you to decode and filter transmissions from particular sensors. You need only enter those pulse lengths that are in use; any that are not found in the stream are ignored, so you can leave them as is.

Select the version of the **CRC** in use.

Select any of the following that should be included in the CRC computation and decoding:

- **SCN**—Status and Communication Nibble.
- **ID**—Identifier (0 - 3) of the sensor sending the message, matching the MTP length you entered on the dialog.

  Note: If you wish to decode sensor IDs, be sure to assign each a unique MTP both in the decoder and in your system design. If you interrogate two sensors with pulses of the same length, the software will decode the response, but the message will fail the CRC check when you have chosen to decode ID, as well.

- **RC**—Rolling Counter identifying the (virtual) sequence number of a message from a given sensor. This field is used primarily for tuning the decoder, ensuring that the CRC is being properly computed. To use it, check the box, then enter an **RC Val**(ue) starting from 0 until any CRC error messages in the decoder table disappear.

Fast Channel Subdialog

The Fast Channel dialog will appear if you selected a Decode Type of "Words" and Channels is set to either "Fast" or "Both."

The SENT protocol specifies four user-defined data fields, D0 through D3. Each of these can be used to present the content of the message payload as it was programmed. Select each that you wish to decode, then specify the payload interpretation.

Filter Dx Errors removes potentially incorrect D0-D3 values from the result table when an error occurs in the SENT message so that they cannot be used by downstream processes such as graphs and measurements. If you would prefer to see the error data in the table, deselect this box.

Interpretation of Offset defines the units in which the message Offset is entered: Bits, Nibbles or High Speed nibbles. Select Bits only when there is a need to save bits so that more information can be compacted into the relatively short message. Select High Speed only when decoding High Speed SENT.

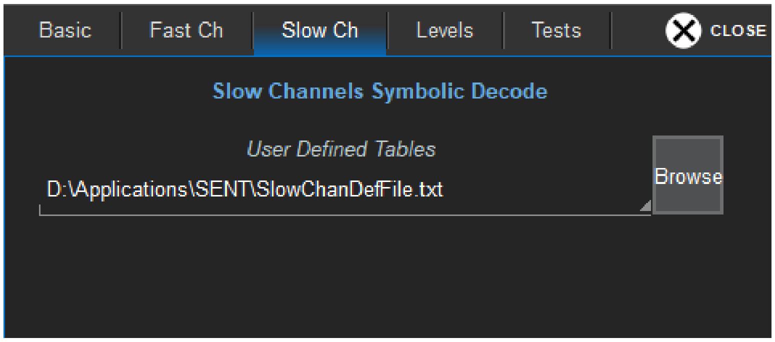
Check **Active** to decode the corresponding user-defined data field.

In **Offset**, enter the number of Bits, Nibbles, or High Speed Bits that constitute the message offset (i.e., the position from which to start decoding content).

Enter the number of **Nibbles** in the payload.

Select whether data is presented in LSN or MSN **Order**.

Slow Channel Subdialog

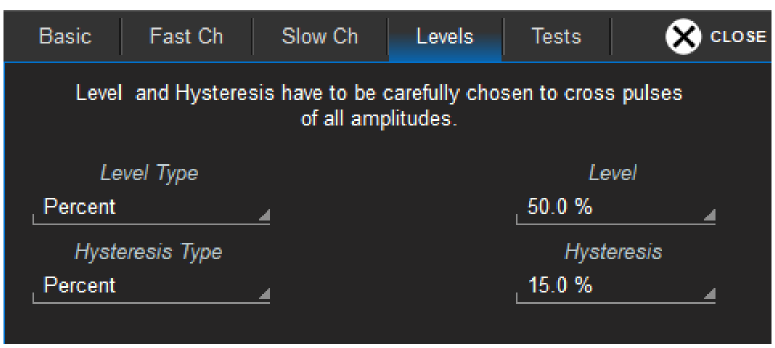


The Slow Channel dialog will appear if you selected a Decode Type of “Words” and Channels is set to either “Slow Only” or “Both.” The Slow Only option takes a single data value from each of 16 or 18 SENT message packets and builds a Slow Channel result, displayed on the decoder result table.

A plain text Slow Channel Definition File (SCDF) is used to decode this data into a more meaningful presentation of the assimilated slow channel data. For example, it is expected that different sensors will require different SCDF files. On the Slow Channel dialog, select the User Defined Tables field and Browse to the SCDF file to use for this purpose.

Tip: To simplify uploading a new SCDF file, first copy it to oscilloscope directory D:\Applications\Sent.

Levels Subdialog

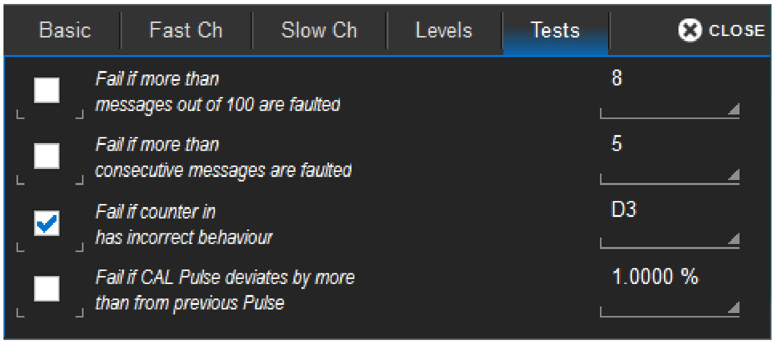


Enter the vertical **Level** used to determine the edge crossings of the signal. This value will be used to determine the bit-level decoding.

Optionally, enter a **Hysteresis** band value. Hysteresis represents the amount the signal may rise or fall from the crossing Level without affecting the bit transition.

For guidelines, see Setting Level and Hysteresis.

Tests Subdialog

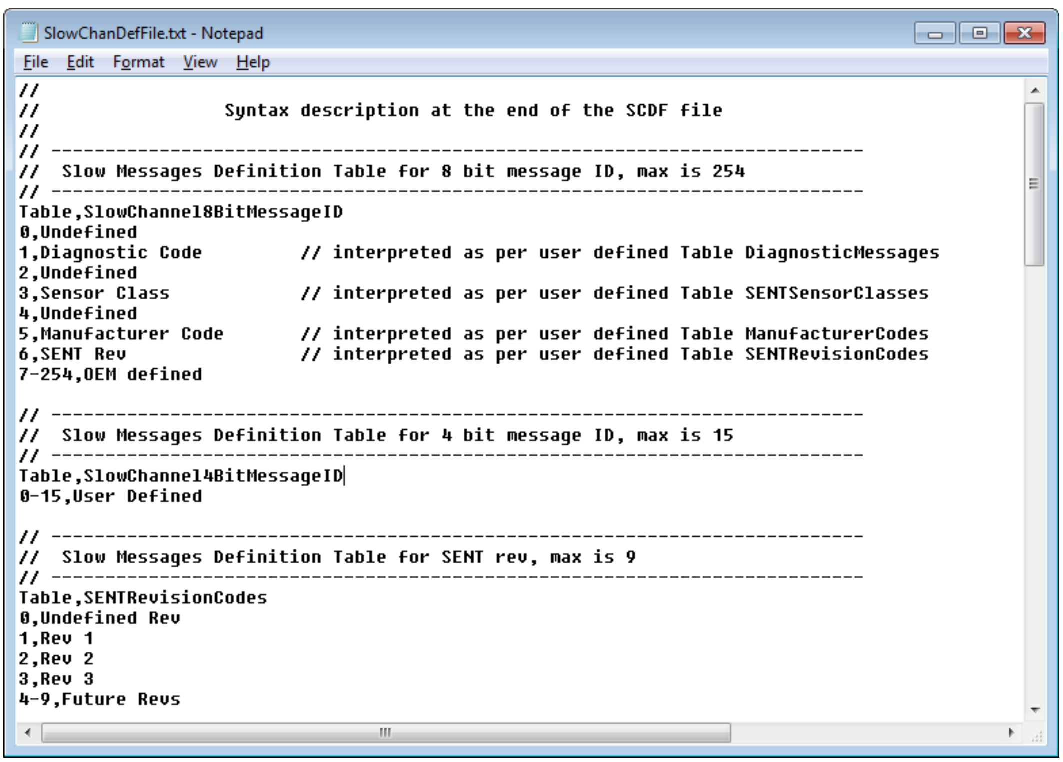


The Test dialog enables you to apply any of four SAE test criteria (the text on the dialog describes the test). Select each test to apply, then enter the value that completes the statement. Results are shown in the decode table as a text message in the Status column and a Boolean output in the S column.

See SAE Test Results for more information.

SCDF File Structure

The mechanism used to translate SENT IDs and values is a simple TXT file containing table definitions. The beginning of the SCDF default file installed on every instrument is shown here:


SlowChanDefFile.txt - Notepad

File Edit Format View Help

```
//
//                  Syntax description at the end of the SCDF file
//
// ---------------------------------------------------------------------------
//  Slow Messages Definition Table for 8 bit message ID, max is 254
// ---------------------------------------------------------------------------
Table,SlowChannel8BitMessageID
0,Undefined
1,Diagnostic Code         // interpreted as per user defined Table DiagnosticMessages
2,Undefined
3,Sensor Class            // interpreted as per user defined Table SENTSensorClasses
4,Undefined
5,Manufacturer Code       // interpreted as per user defined Table ManufacturerCodes
6,SENT Rev                // interpreted as per user defined Table SENTRevisionCodes
7-254,OEM defined

// ---------------------------------------------------------------------------
//  Slow Messages Definition Table for 4 bit message ID, max is 15
// ---------------------------------------------------------------------------
Table,SlowChannel4BitMessageID
0-15,User Defined

// ---------------------------------------------------------------------------
//  Slow Messages Definition Table for SENT rev, max is 9
// ---------------------------------------------------------------------------
Table,SENTRevisionCodes
0,Undefined Rev
1,Rev 1
2,Rev 2
3,Rev 3
4-9,Future Revs
```

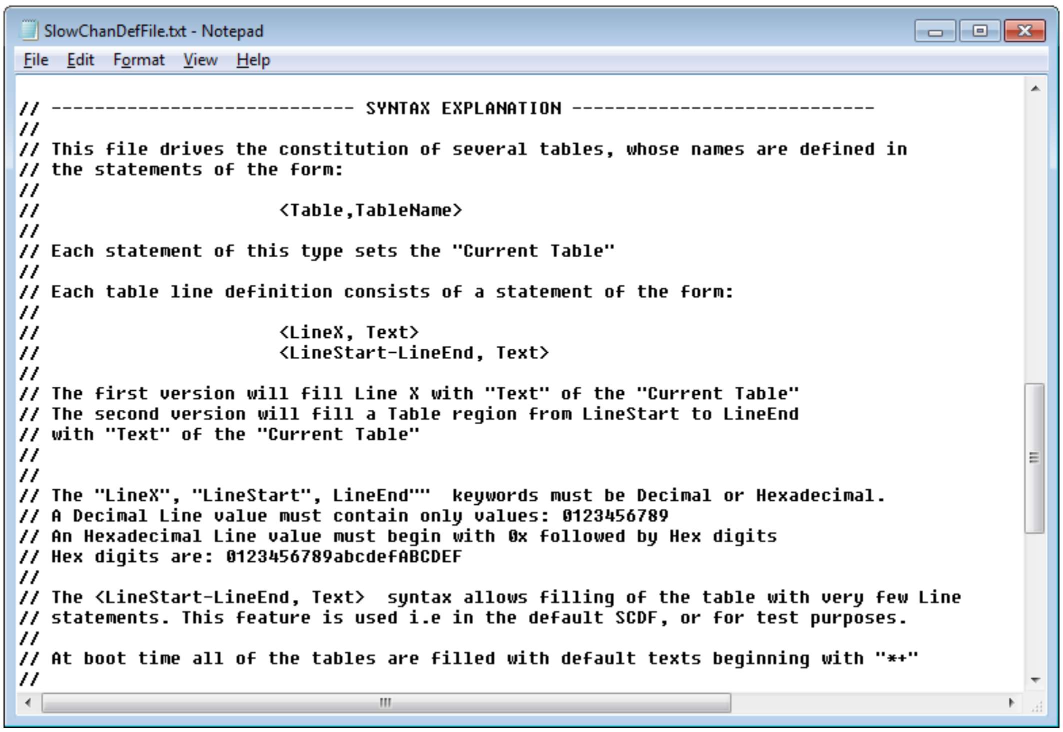
Reserved Names

The SCDF syntax defines several reserved names to identify the auxiliary tables used for:

- Table, SlowChannel8BitMessageID, used to interpret the 8 bit message ID
- Table, SlowChannel4BitMessageID, used to interpret the 8 bit message ID
- Table, SENTRevisionCodes, used to interpret the value conveyed by message ID 6
- Table, SENTSensorClasses, used to interpret the value conveyed by message ID 3
- Table, DiagnosticMessages, used to interpret the value conveyed by message ID 1
- Table, ManufacturerCodes, used to interpret the value conveyed by message ID 5

Syntax Errors

The syntax is documented in the file itself as comments toward the end, as well as the syntax errors detected during the parsing.




```
// ---------------------------- SYNTAX EXPLANATION ----------------------------
//
// This file drives the constitution of several tables, whose names are defined in
// the statements of the form:
//
//                      <Table,TableName>
//
// Each statement of this type sets the "Current Table"
//
// Each table line definition consists of a statement of the form:
//
//                      <LineX, Text>
//                      <LineStart-LineEnd, Text>
//
// The first version will fill Line X with "Text" of the "Current Table"
// The second version will fill a Table region from LineStart to LineEnd
// with "Text" of the "Current Table"
//
//
// The "LineX", "LineStart", LineEnd""  keywords must be Decimal or Hexadecimal.
// A Decimal Line value must contain only values: 0123456789
// An Hexadecimal Line value must begin with 0x followed by Hex digits
// Hex digits are: 0123456789abcdefABCDEF
//
// The <LineStart-LineEnd, Text>  syntax allows filling of the table with very few Line
// statements. This feature is used i.e in the default SCDF, or for test purposes.
//
// At boot time all of the tables are filled with default texts beginning with "*+"
//
```

Parsing errors are listed at the very end of the file. Note that the parsing errors are very useful when beginning to use the SCDF, in order to located syntax errors. The syntax is very strict, in order to keep the parser as simple as possible.

SlowChanDefFile.txt - Notepad

File Edit Format View Help

```
// At boot time all of the tables are filled with default texts beginning with "*+"
//
// --------------------------- SYNTAX ERRORS EXPLANATION ---------------------
//
// 1-Line Value and Range Syntax Errors
//
//  "Line XX, index YYY out of range (0 to Max Line for Current Table)       i.e.  1234,Some 4 bit ID message
//  "Line XX has more then 1 '-' separator in Line Range: "                  i.e.  1-3-8,User Text
//  "Line XX, Line Range has Start:SS > End:EE"                              i.e.  25-4,User Text
//  "Line XX, line contains non digits on position %d: "                     i.e.  2$3, User Text
//
// 2-General Line Syntax Errors
//
//  "Line XX has CC Commas Instead Of 1"                                     i.e.  21,User Text,,,,
//  "Line XX has No Commas"                                                  i.e.  A Line without commas
//  "Table Line Definition XX Has UU Tokens Instead Of 2"                    i.e.  34,User Text1,User Text2
//  "Line XX, Line Definition exceeds 10 characters"                         i.e.  12345-13345,User Text
//  "Line XX, Message Length YY exceeds 60 characters"                       i.e.  3,Some Very Long User Text
//  "Line XX, Warning,empty Message?"                                        i.e.  3,
//
// 3-Table Selection Syntax Errors
//
//  "Line XX, Warning: Table at YY replaced by User Entry "                  Line is redefined
//  "Line XX defines a table line, but no table is selected"                 Line Definition starts before Table Definition
//  "Line XX, unknown error, rejects all tokens"                             Catch all when Line Parser is overwhelmed
//  "Table Selector Definition %d Has %d Tokens Instead Of 2"                i.e. Table,ManufacturerCodes,Some More Useless Text
//  "Line XX specifies unknown Table"                                        i.e. Table,UnknownTableName
```

In addition to the SCDF file, parsing errors are always emitted at the bottom of the oscilloscope screen, with the last error overwriting the previous ones. It is therefore advisable to fix the errors as they occur, beginning with the last emitted error, then reload the file and see if any errors are left.

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 alternatively be set as an absolute voltage (with reference to the waveform 0 level) by changing the **Level Type** to Absolute.

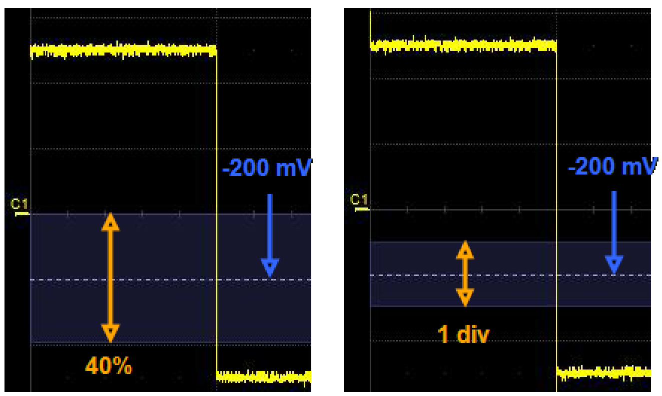
Percent mode is easy to set up because the software immediately determines the optimal threshold, but in some cases it might be beneficial to switch to Absolute mode:

- 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. As with Level Type, **Hysteresis Type** may be either a percentage of amplitude or an absolute number of vertical grid divisions.



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 the same or different modes. For a few protocols, Hysteresis can only be set as a number of mV plus/minus the 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.

Failure to Decode

Three conditions in particular may cause a decoder to fail, in which case a failure message will appear in the first row of the summary result table, instead of in the message bar as usual.

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.

In all the above cases, the decoding is turned off to protect you from incorrect data. Adjust your acquisition settings accordingly, then re-enable the decoder.

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

To first set up a decoder, go to the Decode Setup dialog. Once decoders have been configured, use the Serial Decode dialog to quickly turn on/off a decoder or make minor modifications to the settings.

To turn on decoders:

1. On the same row as the **Decode #**, check **On** to enable the decoder.

   As long as On is checked (and there is a valid acquisition), a result table and decoded waveform appear. The number of rows of data displayed will depend on the **Table #Rows** setting (on the Decode Setup dialog).

2. Optionally, modify the:
   - **Protocol** associated with the decoder.
   - **Data (Source)** to be decoded.

3. Check Link To Trigger **On** to tie this decoder setup to a serial trigger setup.

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

Reading Waveform Annotations

When a decoder is enabled, 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.

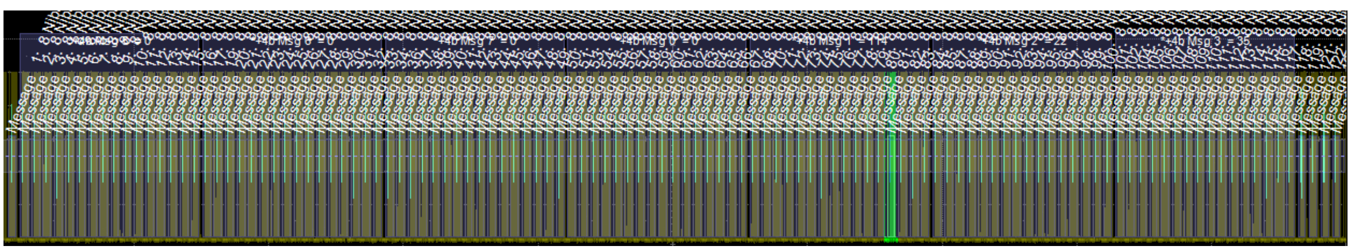
The amount of information shown on an annotation is affected by the width of the rectangles in the overlay, which is determined by the magnification (scale) of the trace and the length of the acquisition. Zooming a portion of the decoder trace will reveal the detailed annotations.

SENT Decoded Waveform Annotations

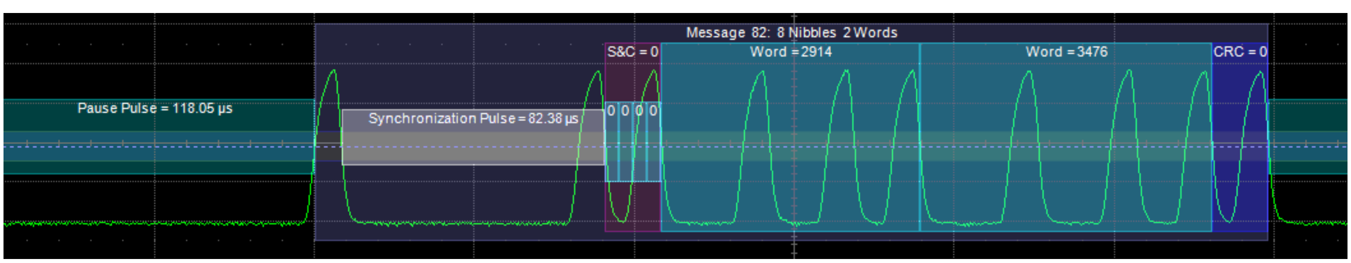
These overlays appear on a SENT trace, or its zoom, to highlight key portions of the decoded signal.

Annotation	Overlay Color (1)	Text (2) (3)
Message Burst	Navy Blue (behind data fields)	Message <ID>
Pause Pulse	Dark Green	Pause Pulse = <time>
Syncronization Pulse	Grey	Synchronization Pulse = <time>
Status & Communication Bits	Purple	S&C = <value>
Payload Data	Aqua Blue	[Word | Nibble] = <value>
Cyclic Redundancy Check	Royal Blue	CRC = <value>

1. Combined overlays affect the appearance of colors.
2. Text in brackets < > is variable. The amount of text shown depends on your zoom factors.
3. Data values are shown in Nibbles or Words depending on your decoder selection.



Decoded waveform. At this resolution, very little information appears on the overlay.



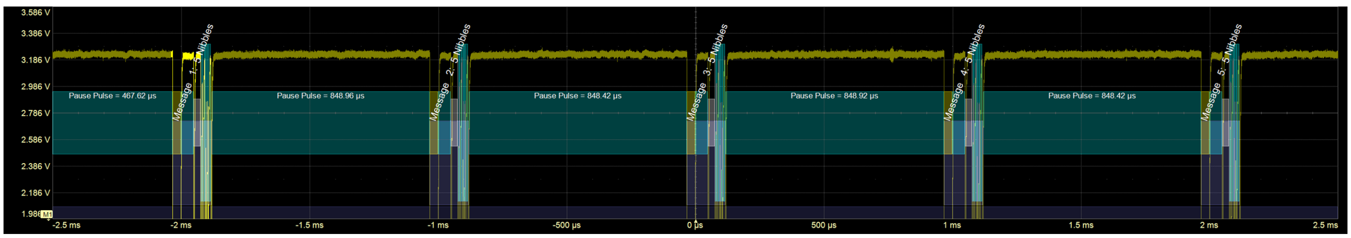
Zoomed waveform annotations, showing decoded data.

SENT SPC Decoded Waveform Annotations

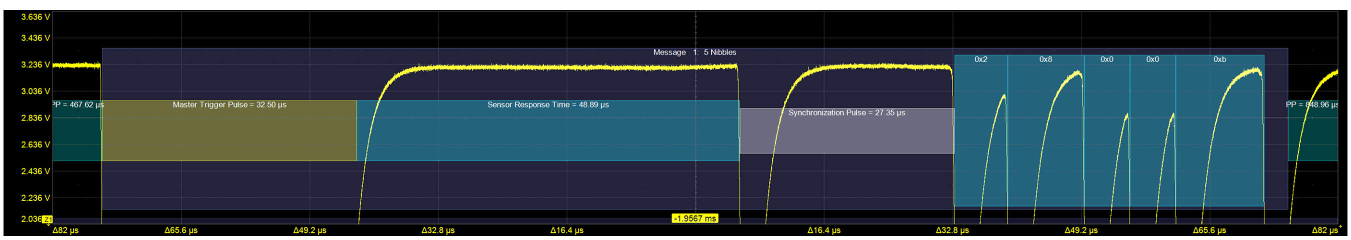
These overlays appear on a SENT SPC decoding, or its zoom trace, to highlight key portions of the decoded signal.

Annotation	Overlay Color (1)	Text (2) (3)
Message Burst	Navy Blue (behind data fields)	Message <ID>
Pause Pulse	Dark Green	Pause Pulse = <time>
Master Trigger Pulse	Olive	Master Trigger Pulse = <time>
Sensor Response Pulse	Aqua Blue	Sensor Response Time = <time>
Syncronization Pulse	Grey	Synchronization Pulse = <time>
Payload Data	Aqua Blue	[Word | Nibble] = <value>

1. Combined overlays affect the appearance of colors.
2. Text in brackets < > is variable. The amount of text shown depends on your zoom factors.
3. Data values are shown in Nibbles or Words depending on your decoder selection.



Decoded waveform. At this resolution, very little information appears on the overlay.



Zoomed waveform annotations, showing decoded data.

Serial Decode Result Table

When **View Decode** is checked on the Decode Setup Dialog *and* a source signal 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. See also Automating the Decoder.

Tip: If any downstream processes such as measurements reference a decoder, 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 export data rather than view the decoding.

Table Rows

Each row of the table represents one index of data found within the acquisition, numbered sequentially. Exactly what this represents depends on the protocol and how you have chosen to "packetize" the data stream when configuring the decoder (frame, message, packet, etc.).

Note: For some decoders, it is even possible to turn off packetization, in which case all the decoded data appears on one row of the table.

When multiple decoders are run at once, the index rows are combined 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.

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
22	-34.621 ms	SENT	Message 22: 8 Nibbl...		0	
23	-31.227 ms	SENT	Message 23: 8 Nibbl...		2	
24	-27.832 ms	SENT	Message 24: 8 Nibbl...		12	
25	-24.635 ms	SENT	Message 1: 8 Nibbl...		6	
26	-24.437 ms	SENT	Message 25: 8 Nibbl...		14	
27	-24.220 ms	SENT	Message 2: 8 Nibbl...		9	
28	-23.796 ms	SENT	Message 3: 8 Nibbl...		12	
29	-23.373 ms	SENT	Message 4: 8 Nibbl...		3	

Example summary result table, with results from two decoders combined 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
22	-34.621 ms	SENT	Message 22: 8 Nibbl...		0	
23	-31.227 ms	SENT	Message 23: 8 Nibbl...		2	
24	-27.832 ms	SENT	Message 24: 8 Nibbl...		12	
25	-24.635 ms	SENT	Message 1: 8 Nibbl...		6	
26	-24.437 ms	SENT	Message 25: 8 Nibbl...		14	

Sync	Msg	Stat	b0	b1	b2	b3	D0	D1	Status
672.4 µs	Message 25: 8 Nibbles 2 Words	9	1	0	0	1	1	3807	

Index	Time	Protocol	Message	Data	CRC	Status
27	-24.220 ms	SENT	Message 2: 8 Nibbl...		9	

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

This extracted data appears on a SENT detailed result table. Columns can be hidden by customizing the result table.

Column	Extracted or Computed Data
Index	Number of the line in the table
Time(µs)	Time (in microseconds) relative to the trigger of the beginning of the SENT burst
Sync	Measured length of the synchronization pulse. The pulse width is measured between the two falling edges of the sync pulse, at the intersection of the signal and the Level selected on the Level dialog. Note that a large hysteresis will impact this value.
Tick	Tick Time; value of the sync pulse divided by 56
Msg	Message summary, with the number of transitions, nibbles and words
RMS	Root Mean Square value of the falling edge crossings, usually in nanoseconds
Pause P(ulse)	Time from the end of one burst to the beginning of the next burst
S	Boolean Status: 0 when Status is empty (no errors), 1 when Status contains text (errors reported). The S column can be used with the oscilloscope measure, math, and analysis features.
Status	Reported errors and warnings
When decoding Nibbles...	
Nibbles	Number of nibbles
When decoding Words...	
Stat	Value of the status and communication (S&C) nibbles. This value is split into its component bits in the next four columns to help interpret the contents. Component bits are chromacoded for quick reference to their result values in the ID, Data, and CRC columns.
b0-b1	Reserved for special applications
b2	Message data bits (slow channels)
b3	Message start bits
D0 - D3	Message data bits (fast channels)
ID	Result value of 8-bit ID

Column	Extracted or Computed Data
Data	Result value of 12-bit data
CRC	Value of the CRC (error detection) nibble compared to values of the other nibbles of the message. If it does not match, an error appears in the Status column. It is normal that the first and last messages of a record, when truncated, generate a CRC error.

SENT	Time	Sync	Tick	Msg	Stat	b0	b1	b2	b3	D0	D1	ID	Data	CRC	RMS	Pause P	S	Status
1	-24.635 ms	82.34 µs	1.47 µs	Message 1: 8 Nibbles 2 Words	0	0	0	0	0	2742	2373			6	19 ns	365.6 µs	0	
2	-24.220 ms	82.35 µs	1.47 µs	Message 2: 8 Nibbles 2 Words	8	0	0	0	1	2744	1861			9	26 ns	116.6 µs	0	
3	-23.796 ms	82.33 µs	1.47 µs	Message 3: 8 Nibbles 2 Words	4	0	0	1	0	2747	1093			12	23 ns	100.3 µs	0	
4	-23.373 ms	82.36 µs	1.47 µs	Message 4: 8 Nibbles 2 Words	0	0	0	0	0	2750	325			3	32 ns	101.8 µs	0	
5	-22.949 ms	82.35 µs	1.47 µs	Message 5: 8 Nibbles 2 Words	4	0	0	1	0	2752	3893			15	30 ns	120.9 µs	0	
6	-22.526 ms	82.32 µs	1.47 µs	Message 6: 8 Nibbles 2 Words	0	0	0	0	0	2755	3125			10	19 ns	97.38 µs	0	
7	-22.102 ms	82.34 µs	1.47 µs	Message 7: 8 Nibbles 2 Words	0	0	0	0	0	2757	2613			0	14 ns	110.6 µs	0	
8	-21.679 ms	82.31 µs	1.47 µs	Message 8: 8 Nibbles 2 Words	0	0	0	0	0	2760	1845			10	23 ns	125.3 µs	0	

Section of typical SENT detailed result table.

SENT	Time	Sync	Tick	Msg	Stat	b0	b1	b2	b3	ID	Data	CRC	RMS	Pause P	S	Status
1	-15.696 ms	162.1 µs	2.89 µs	Message 1: 8 Nibbles 0 Words	0	0	0	0	0			6	4e-3	186.5 µs	0	
				Raw Nibble 0 :-0.0005 Δ: -5.403e-04 Δ^2: 2.919e-07												
				Raw Nibble 1 :04.0026 Δ: 2.591e-03 Δ^2: 6.712e-06												
				Raw Nibble 2 :00.9986 Δ: -1.439e-03 Δ^2: 2.071e-06												
				Raw Nibble 3 :06.0000 Δ: -3.816e-05 Δ^2: 1.456e-09												
				Raw Nibble 4 :04.9985 Δ: -1.550e-03 Δ^2: 2.402e-06												
				Raw Nibble 5 :03.0097 Δ: 9.683e-03 Δ^2: 9.377e-05												
				Raw Nibble 6 :12.0039 Δ: 3.850e-03 Δ^2: 1.483e-05												
				Raw Nibble 7 :05.9996 Δ: -3.984e-04 Δ^2: 1.587e-07												
				RMS: 3.877e-03												
11	-14.878 ms	162.1 µs	2.89 µs	Message 11: 8 Nibbles 0 Words	0	0	0	0	0			6	2e-3	16.41 µs	0	
				Raw Nibble 0 :00.0003 Δ: 2.701e-04 Δ^2: 7.293e-08												

SENT result table with nibble details.

SENT SPC Result Table

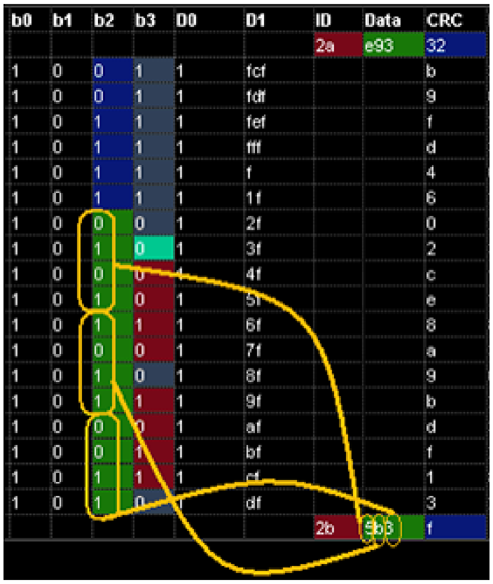
In addition to the columns above, the result table will show the following when decoding SENT SPC:

Column	Extracted or Computed Data
MT Pulse	Measured length of the Master Trigger Pulse (MTP)
RespT	Measured response time from MTP to the sensor response
ID	ID of the sensor sending the message, 0 through 3

SE...	Time	MTPulse	RespT	ID	Sync	Tick	Msg	Nibbles	RMS	Pause P	S	Status
1	-2.0324 ms	32.50 µs	48.89 µs	0	27.35 µs	488 ns	Message 1: 5 Nibbles	2 8 0 0 b	64e-3	467.6 µs	0	
2	-1.0321 ms	32.51 µs	48.83 µs	0	27.34 µs	488 ns	Message 2: 5 Nibbles	2 8 0 0 b	60e-3	849.0 µs	0	
3	-32.439 µs	32.50 µs	48.88 µs	0	27.35 µs	488 ns	Message 3: 5 Nibbles	2 8 0 0 b	62e-3	848.4 µs	0	
4	967.784 µs	32.50 µs	48.88 µs	0	27.34 µs	488 ns	Message 4: 5 Nibbles	2 8 0 0 b	64e-3	848.9 µs	0	
5	1.9675 ms	32.50 µs	48.82 µs	0	27.35 µs	488 ns	Message 5: 5 Nibbles	2 8 0 0 b	66e-3	848.4 µs	0	

SENT SPC detailed result table.

Chromacoding of S&C Bits



Enhanced slow channel message using color to show distribution of Sync, ID, Data, and CRC bits

In addition to the decoded bit values, the SENT result table displays color-coding of Status and Communication (S&C) bits so that it is easy to recognize how they are distributed over many Fast Messages (16 or 18). The resulting bit transmission is shown in the ID, Data, and CRC columns.

Bits	Color Code
Sync pattern	Grey
4-bit or 8-bit ID	Red
12-bit Data	Green
Control bit: When 0, ID is 8-bits and Data 12-bits wide When 1, ID is 4-bits and Data 16-bits wide	Turquoise
4-bit or 6-bit CRC	Blue

SENT Errors

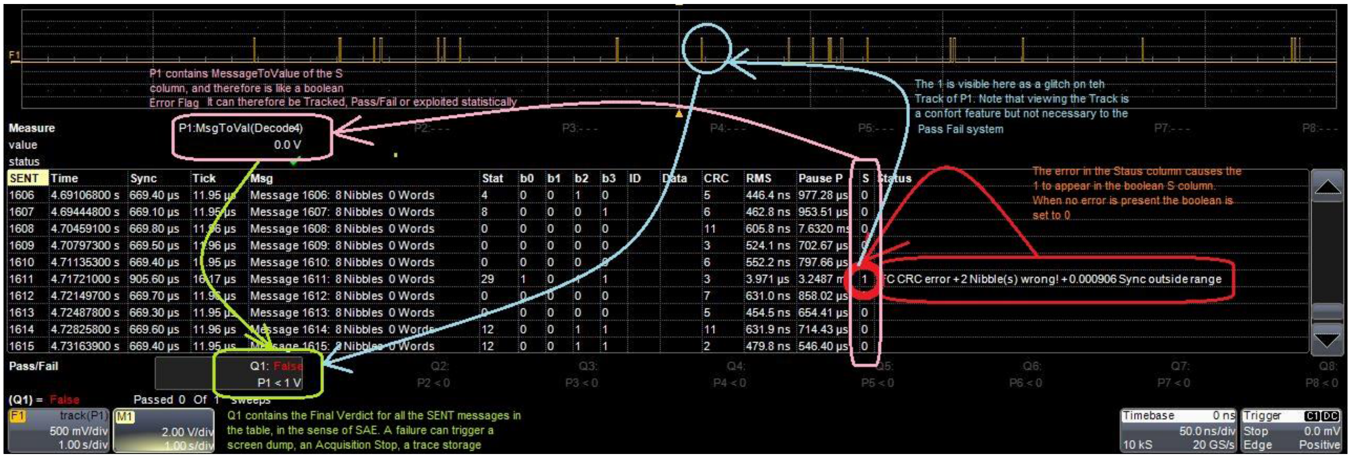
These messages may appear in the **Status** column of the SENT detailed result table:

Channel	Error	Message
Fast	Nibble value outside range 0-15	%d Nibble(s) wrong!
Fast	Sync outside range ± 25% of Sync computed as 56 * user TickTime	%f Sync outside range
Fast	Fast channel CRC error	FC CRC error
Slow	In enhanced Slow Message, bit 7, 13 or 18 is not 0	B [7|13|18] != 0!
Slow	Enhanced Slow Message CRC error	SC(18) CRC Error
Slow	Legacy Slow Message CRC error	SC(16) CRC Error

SAE Test Results

The results of SAE tests are populated to the **Status** and **S** columns of the result table:

- **Status** shows a description of the error
- **S** shows a 0 if there is no error, or 1 when an error appears, making it a Boolean error flag that can be tracked statistically or exploited for measurements and Pass/ Fail testing.



In the example above:

- A glitch in record 1611 appears as 1 in the S column.
- P1 is MsgToValue set to "pass thru" column S values.
- Pass/ Fail test Q1 is set to fail whenever P1 is *not* "less than 1" (P1 < 1 = False), meaning it is not an error free 0. The "fail" could in turn be set up to trigger any number of other actions.

Note: It is not necessary to display the measurement source trace in order for the SAE test results to be used this way.

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 zoom of the decoding highlighted on the summary table at the same time.

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 zoom that is not yet turned on.

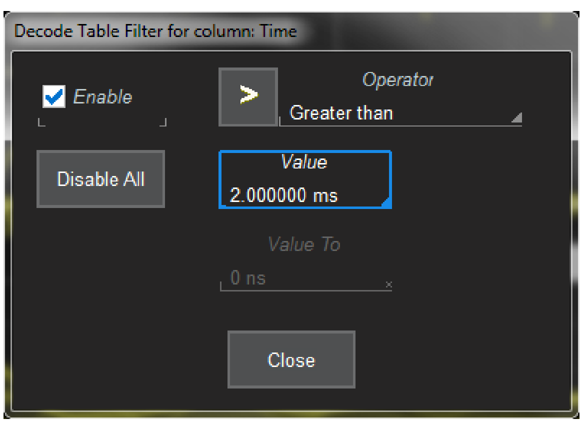
Index	Time	Protocol	Message	Data	CRC	Status
32	-22.102 ms	SENT	Message 7: 8 Nibbl...		0	
33	-21.679 ms	SENT	Message 8: 8 Nibbl...		10	
34	-21.256 ms	SENT	Message 9: 8 Nibbl...		12	
35	-21.042 ms	SENT	Message 26: 8 Nibbl...		8	
36	-20.832 ms	SENT	Message 10: 8 Nibbl...		0	
37	-20.409 ms	SENT	Message 11: 8 Nibbl...		3	
38	-19.985 ms	SENT	Message 12: 8 Nibbl...		6	
39	-19.562 ms	SENT	Message 13: 8 Nibbl...		12	

Example multi-decoder summary table, both zoomed indexes highlighted.

Filter Results

Those columns of data that have a drop-down arrow in the header cell can be filtered: 

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



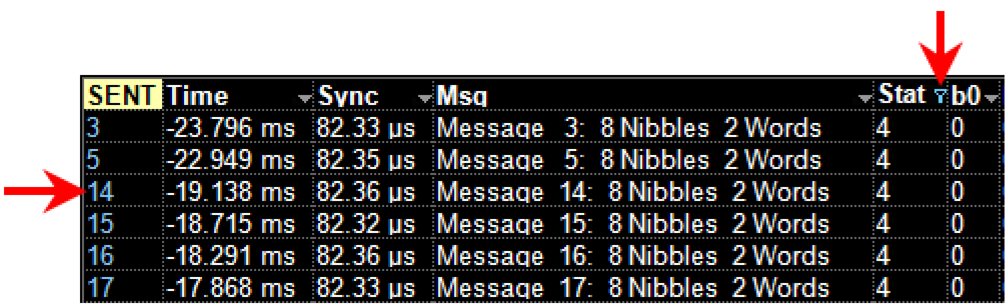
Select a filter **Operator** and enter a **Value** that satisfies the filter condition.

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 format that may be entered in Value for that column of data. Numeric values must be within .01% tolerance of a result 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 until changed 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.



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.

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

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.

On some decoders, 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.

The SENT decoder does not utilize the Bit Rate Tolerance setting.

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.

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 **Decode*x*** as the source.

Using ColToValue and MsgToValue

Column to Value

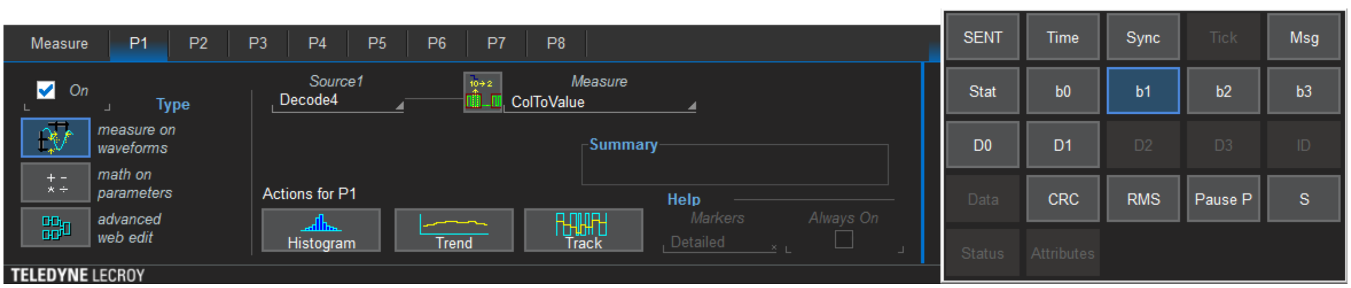
ColToValue acts as a special "pass thru" of one column of decoder table values, allowing you to graph or run Pass/Fail tests on the values via a parameter that is configured with this measurement.

ColToValue acts upon any columns in the table (except those with text values), but does not perform any filtering or extraction of the results. There is no special setup other than the column selection. It is very similar to using Excel to graph a column of data. ColToValue works best if all the lines have similar content. When they do not, use MsgToValue to first filter lines with similar content.

Tip: SENT fast channels are most efficiently tracked using ColToValue.

ColToValue can only be selected from the standard measurement set up dialogs under the Serial Decode group, not from the Measure/Graph Setup dialog that is activated with Measure options. There are two keys to using this parameter successfully:

- Because it is not set up like other serial decode measurements, the parameter source is not automatically set on a decoder. You must be sure to select the correct decoder (Decode1 - Decode4), rather than an input channel, as the parameter source. This will expose the decoder table columns for selection on the ColToValue subdialog.
- Be sure there is an active decoding. If there is no data in the result table, the Select Column field on the ColToValue subdialog shows "undef", and the selector is inactive. As long as there is data in the table, you should see a selection of those columns populated by the decoder.



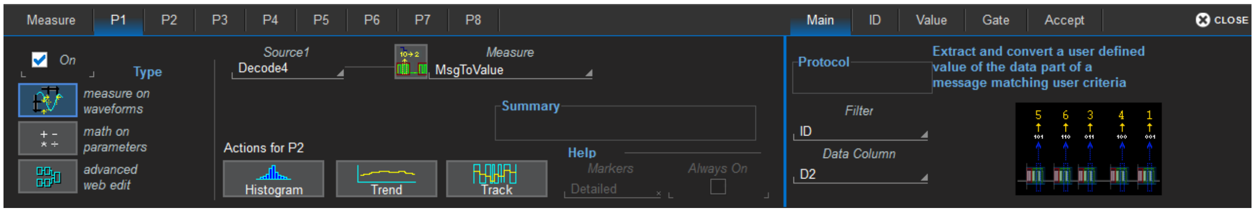
Message to Value

MsgToValue enables you to apply oscilloscope features to a subset of the result table (in SENT usually only the data columns) 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. The track of the Msg to Value parameter is, in effect, a Digital to Analog Converter (DAC) that can display digitally-encoded sensor data as an analog waveform.

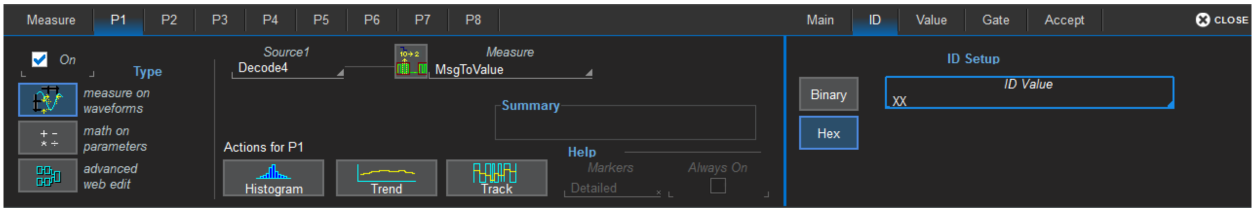
Tip: To track SENT Slow Channels, use MsgToValue to filter by Slow Channel ID. Also, when SAE tests are turned on, the S column of the result table contains the Boolean result of the tests. You can configure a parameter with MsgToValue assigned to decoder column S, then set a Pass/Fail test on the occurrence of 1 or 0 in that parameter.

MsgToValue requires several selections from the parameter set up subdialogs:

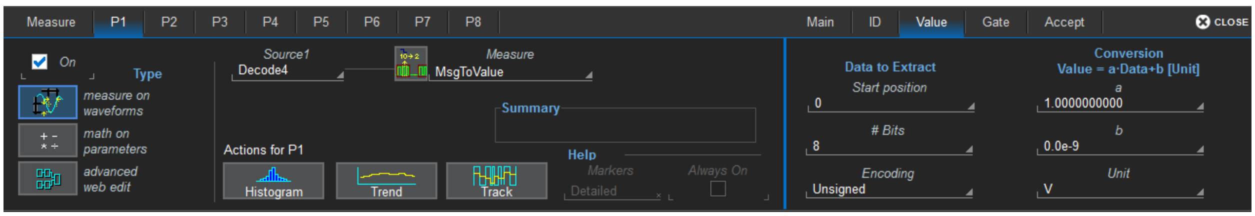
- Choose whether or not you wish to **Filter** by **ID** or accept **Any** packets, and select the **Data Column** from those that have been populated with data.



- 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.

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

Not supported on legacy WaveSurfer3000 models.

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.

Turn off tables, annotations, and waveform traces. As long as downstream processes such as measurements or Pass/Fail tests reference a decoder, the decoder can function without actually displaying results. If you do not need to see the results but only need the exported data, you can deselect View Decode, or minimize the number of lines in a table. Closing input traces also helps.

Decrease the number of 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.

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 XStreamBrowser 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 remote PC if you have a DCOM 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;-4.55526	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;2.274027	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;2.62041	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;0.005388911407171		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:

CAN FD	Time	Format	ID	IDE	FDF	BRS	ESI	RTR	DLC	Data
1	-7.4822 ms	FD	0x01f	0	1	1	0		6	ae 8f a0 a3 00 06
2	-4.5915 ms	FD	0x0be	0	1	1	0		8	00 00 00 00 00 00 00 00
3	-4.4762 ms	FD	0x266	0	1	1	0		6	00 00 00 00 00 00
4	-4.3729 ms	FD	0x02c	0	1	1	0		8	00 00 00 00 00 00 00 00
5	-27.74 µs	Std	0x145	0	0			0	8	45 50 80 00 00 00 00 00
6	2.58442 ms	FD	0x01f	0	1	1	0		6	80 48 61 44 00 06
7	5.35321 ms	FD	0x02c	0	1	1	0		8	00 00 00 00 00 00 00 00

The following table shows example VBS queries you might add to a remote control program to read data from the decoder result table.

Remote Queries	Returned Value (s)	What Is Read by Query
vbs? 'return=app.SerialDecode.Decode1.out.Result.rows'	8	Number of table rows (incl. header Row 0)
vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,0)' vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,1)' vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,2)'	6 2.58442...E-03 FD	Value in first 3 columns of Row 6, including: Index # in Row 6 Col 0 Time in Row 6 Col 1 Format in Row 6 Col 2
vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,12)(0,0)'	128	Data value in ColB0 of Row 6 Col 12
vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,12)(1,0)'	2.62461...E-03	StartTime of Data in ColB0 of Row 6 Col 12 (hidden)
vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,12)(2,0)'	2.62911...E-03	StopTime of Data in ColB0 of Row 6 Col 12
vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,12)(0,1)'	72	Data value in ColB1 of Row 6 Col 12
vbs? 'return=app.SerialDecode.Decode1.out.Result.cellvalue(6,12)(1,1)'	2.62911...E-03	StartTime of Data in ColB1 of Row 6 Col 12 (hidden)

Modifying the Result Table

The CVAR app.SerialDecode.Decode*n*.Decode.ColumnState contains a pipe-delimited list of all the table columns and their current state (visible=on, hidden=off). For example:

app.SerialDecode.Decode1.Decode.ColumnState = "Idx=On|Time=On|Data=On|..."

If you wish to hide or display table columns, send the full string with the state changed from "on" to "off", or vice versa, rather than remove any column from the list.

Serial Trigger

Note: These instructions pertain only to the -TD and -TDME options for those protocols and encoding schemes where serial trigger is supported: 8b10b, 64b66b, 80-bit NRZ, AudioBus (I2S/RJ/LJ), CAN, CAN FD, FlexRay, I2C, I3C, LIN, MIL-1553, SENT, SMBus, SPI, SPMI, UART-RS232 and USB2.

TD 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.

The serial trigger supports fairly simple conditions, such as "trigger at the beginning of any packet," but the conditions can be made more restrictive depending on the protocol and the available filters, such as "trigger on packets with ID = 0x456". The most complex triggers incorporate a double condition on the ID and data, for example "trigger on packets with ID = 0x456 and when data in position 27 exceeds 1000".

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 only operates on 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.

SENT Trigger Setup

To access the serial trigger dialogs:

- Touch the Trigger descriptor box or choose **Trigger > Trigger Setup** from the Menu Bar.
- Touch the **Serial** Type button, and the **SENT** Standard button.

Then, working from left to right, make the desired selections from the SENT dialog.

Source Setup

In **SRC 1 (Data)**, select the data source input channel.

Use the **Threshold** control to adjust the vertical level for the trigger.

Physical Layer and Protocol Settings

Enter the **Tick Time** of the input signal.

Enter a **Tick Time Tolerance** (in percent). This defines how the CAL pulse is filtered with respect to the Tick Time. For example, if a Tick Time of 3 µs is set with a tolerance of 10%, the CAL pulse is expected to be 56 * 3 µs ± 10%, therefore 168 µs ± 10%.

Enter the number of **Nibbles** in one Word (default is 8).

Check **Pause Pulse** if using a Pause Pulse as per the 2010 specification, section 5.2.6.

Check **New CRC** if performing CRC computation as per the recommendations of the 2010 specification, section 5.4.2.2. Otherwise, the trigger will follow the guidelines of the 2008 specification, section 5.4.2.1 (Legacy).

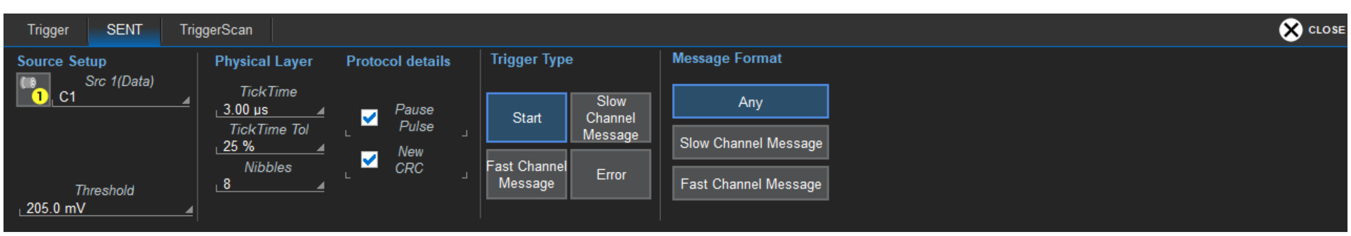
Trigger Type

Choose the type of event on which to trigger:

- **Start** of the selected message type.
- **Slow Channel Message** or **Fast Channel Message** data pattern. Enter the pattern as shown below.
- **Protocol Error**.

Start Trigger

Choose to trigger on the Start of **Any Message**, the next **Slow Channel Message** or the next **Fast Channel Message**.



Slow/ Fast Channel Trigger

First choose to enter and display values in **Binary** or **Hex**(adecimal) format. The selection propagates throughout the entire trigger setup. Toggling between formats does not result in loss of information, but will transform the appearance of values.

For Fast Channel Messages, enter the **Status Nibble** value.

For Slow Channel Messages, choose whether to test **Short Serial Message**(s) or **Enhanced Serial Message**(s) of **4** or **8 Bits**, then enter the frame **ID** value.

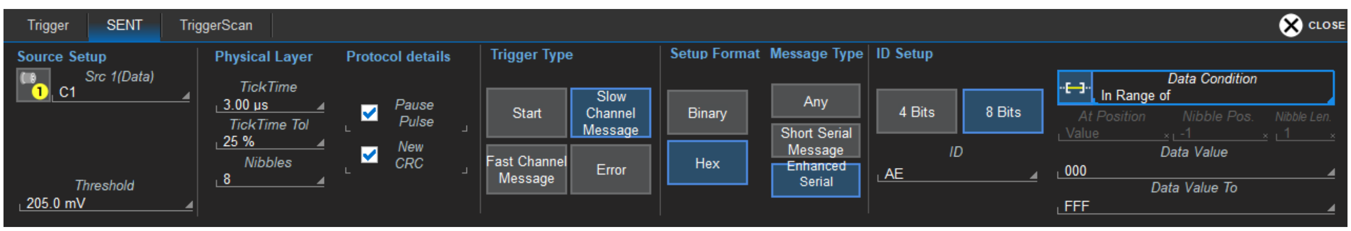
Use **Data Condition** (Boolean operator) and **Data Value** together to specify the data pattern upon which to trigger. The pattern is assumed to begin at the 0 (i.e., first) data byte in the message. If this is not desired, then add preceding or trailing wildcard (X) nibbles to the pattern.

To specify a range of values that may fire the trigger, choose In Range or Out Range. When setting a range, enter the start value in Data Value and the stop value in **Data Value To**.

Note: When more than one data byte is entered, the data is treated as Most Significant Byte (MSB) First. In Hexadecimal format, data must be entered as full bytes even though the minimum acceptable entry is a nibble. If less than a full byte is entered, wildcards (XX) precede the pattern values entered.

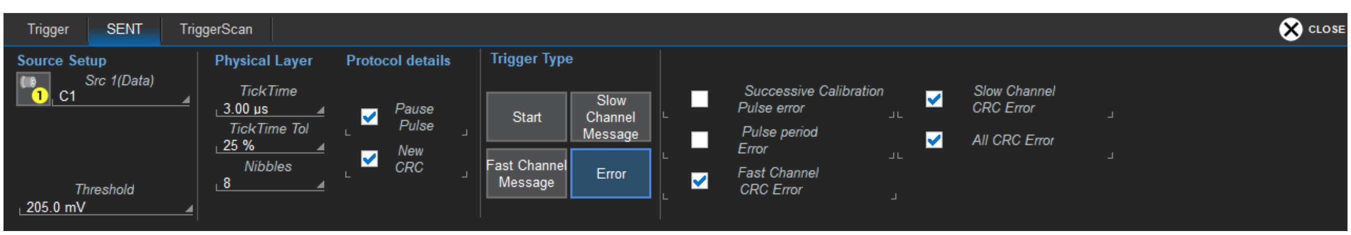
Use an **At Position** of Value (instead of the default "Don't Care") to mark a specific **Nibble Pos**(ition) in the Data field the matching Data Value must occupy. You can select any position up to the maximum valid for that protocol, starting with Byte 0 (the first data byte).

Nibble Length defaults to the length, in nibbles, of the pattern set in Data Value. If the length is changed to a lesser value, the start of the Data Value is truncated by the number of bytes equal to the difference. If the length is increased, wildcards (XX) equal to the difference are appended to the beginning of the value.



Error Trigger

The Error trigger fires whenever a protocol error of the selected type is found. Select all the error types you wish to trigger upon.



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 any -DME or -TDME option) 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 functionality was formerly offered as part of -TDM options and the ProtoBus MAG software option. The features described in this section should be present If you have either of these installed on your oscilloscope.

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 CANtoValue instead of MsgToValue, etc. The measurements are the same.

Measurement	Filters	Description
AnalogToMsg	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.
BusLoad	ID, Data	Computes the load of selected messages on the bus (as a percent).
ColToValue	Column	Extracts the data in a single column of the result table to a measurement parameter location, with no transformation of value.
DeltaMsg	ID, Data	Computes time difference between two messages on a single decoded line.
MsgBitrate	ID, Data	Computes the bitrate of selected messages within the decoded stream.
MsgToAnalog	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.
MsgToMsg	ID, Data	Computes time from start of first message that meets conditions to start of the next message that meets conditions.
MsgToValue	ID, Value	Extracts a selected portion of the data to a measurement parameter location, with optional conversion of value. Data may be selected by ID and/or data field position.
NumMessages	ID, Data	Computes the total number of messages in the decoding that meet conditions.
Time@Msg	ID, Data	Computes time from trigger to start of each message that meets 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).

   Note: It is important to select the correct SENT decoder to use as the source for the MsgtoValue parameter, as this causes the table column selections to appear on the subdialog next to Measure/Graph Setup. Choose the column of data that will "pass thru" to other operations via the parameter .

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.

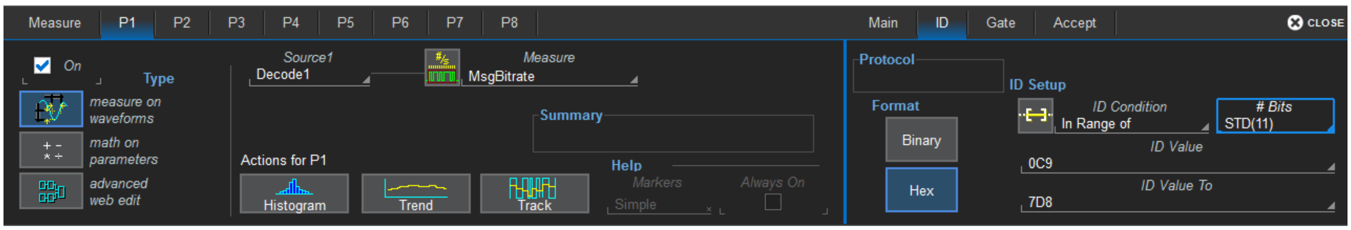
Filtering Measurements

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.

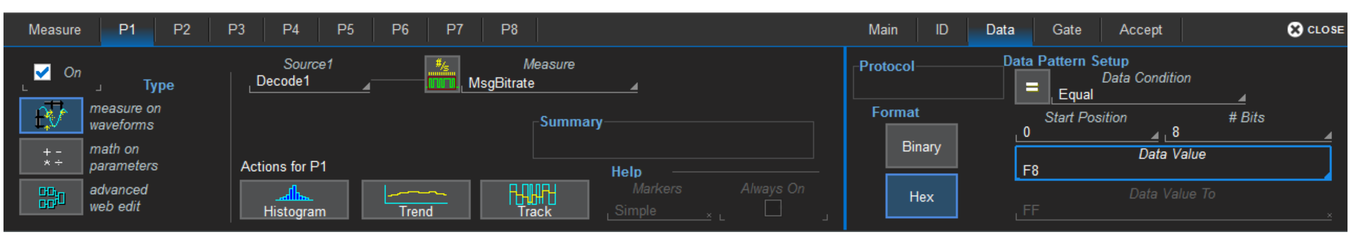


1. On the Main subdialog, choose to **Filter** by **ID** or **ID + Data**.
2. On the **ID subdialog**, choose to enter the ID in **Binary** or **Hex**(adecimal) format.
3. If the field appears, select the **# Bits** used to define the frame ID. (This will change the ID Value field length.)
4. Using the **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".

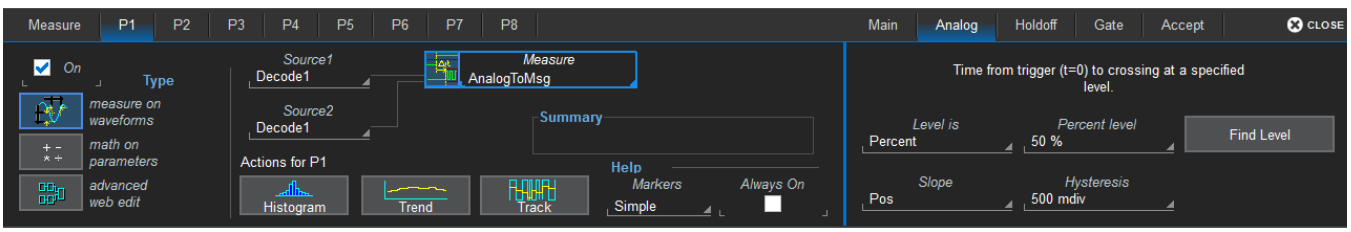
Note: For MsgtoMsg measurements, the data condition is entered twice: first for the Start Message and then for the End Message. The measurement computes the time to find a match to each set of conditions.

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.



Eye Diagram Tests

The -DME and -TDME options provide easy eye diagram setup and eye mask testing.

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 shot acquisitions with decoder and eye diagram enabled to check that both are working correctly.
- Run 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

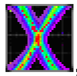
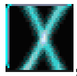
Open the **Eye Diagram Setup** dialog and select the **Decode** for which to create an eye diagram.

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

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.

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.

Choose to display the **Eye Height**, **Eye Width**, or **Mask Hit(s)** measurement parameters. These are added to the Measure table in the first open parameter slots.

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.

Mask Failure Locator Dialog

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

In **Trace Width**, enter the number of UIs surrounding the mask violation to display as "padding."

Check **Stop On Failure** to stop acquisition whenever an eye mask failure occurs.

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.

Technical Support

Live Support

Registered users can contact their local Teledyne LeCroy service center at the number listed on our website.

You can also submit Technical Support requests via the website at:

teledynelecroy.com/support/techhelp

Resources

Teledyne LeCroy publishes a free Technical Library on its website. Manuals, tutorials, application notes, white papers, and videos are available to help you get the most out of your Teledyne LeCroy products. Visit:

teledynelecroy.com/support/techlib

The Datasheet published on the product page contains the detailed product specifications.

Service Centers

For a complete list of offices by country, including our sales & distribution partners, visit:

teledynelecroy.com/support/contact

Teledyne LeCroy
700 Chestnut Ridge Road
Chestnut Ridge, NY, 10977, USA
teledynelecroy.com

Sales and Service:
Ph: 800-553-2769 / 845-425-2000
FAX: 845-578-5985
contact.corp@teledynelecroy.com

Support:
Ph: 800-553-2769
customersupport@teledynelecroy.com

TELEDYNE LECROY
Everywhereyoulook™

700 Chestnut Ridge Road
Chestnut Ridge, NY 10977
USA

teledynelecroy.com