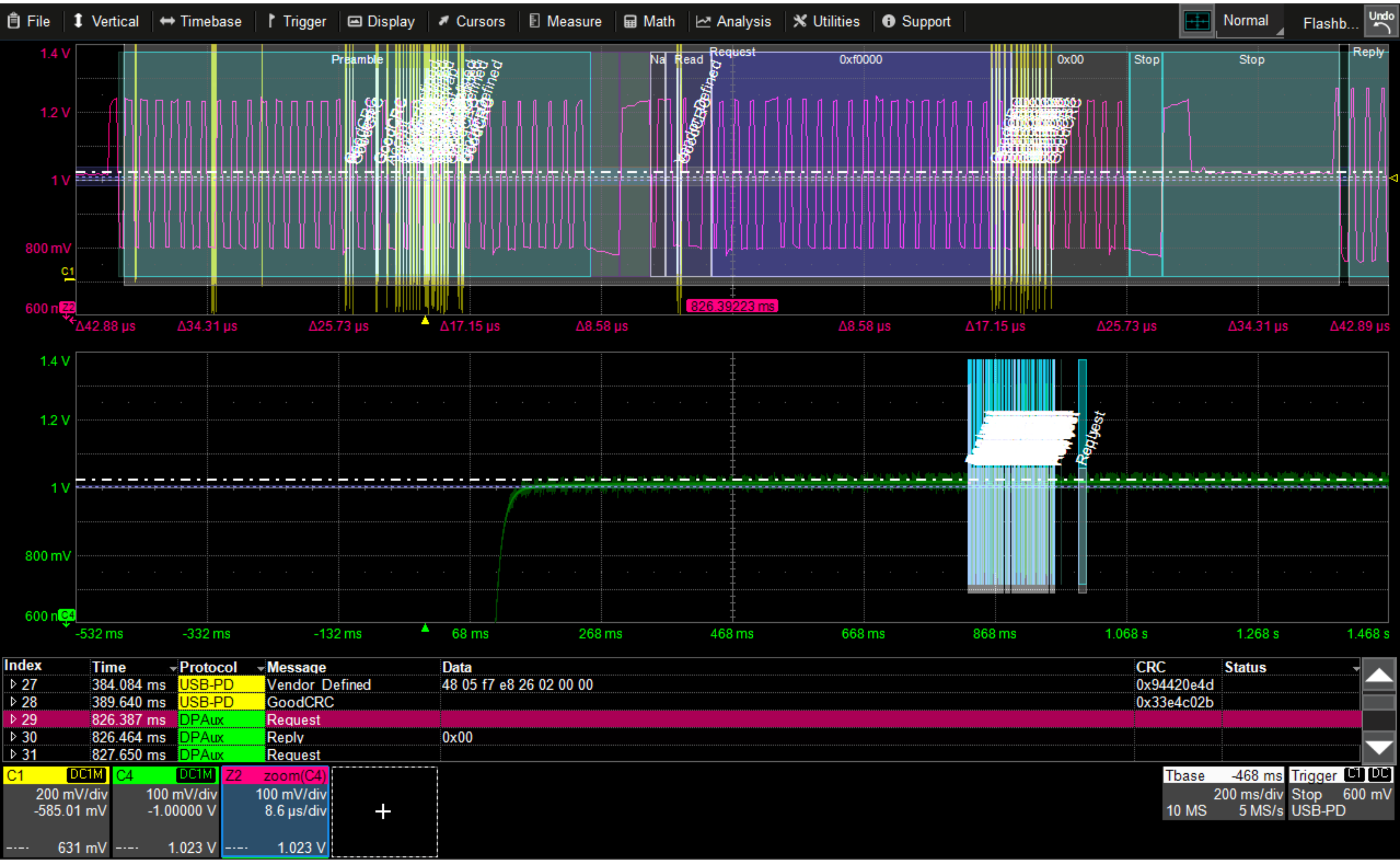


DPAUX

DisplayPort AUX Decode, Measure/Graph and Eye Diagrams

DPAUX DME Instruction Manual

July, 2021
dpaux-dme-im-eng_26jul21.pdf

Contents

About This Manual

This manual explains the basic procedures for using serial data 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 DPAUX D and DME

The Teledyne LeCroy DPAUX (DisplayPort AUX) decoder options apply software algorithms to extract DisplayPort AUX channel information from physical layer waveforms measured on DisplayPort Source and Sink devices. 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 -DME option 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. Connect your signals to the oscilloscope.
2. Set up the decoder using the lowest level decoding mode available, but do not yet enable it.
3. 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.

4. Stop acquisition, then enable the decoder. It will operate on the acquisition in buffer.
5. Use the various decoder tools to verify that transitions are being correctly decoded. Tune the decoder settings as needed to produce a satisfactory decoding.
6. 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.
7. 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.

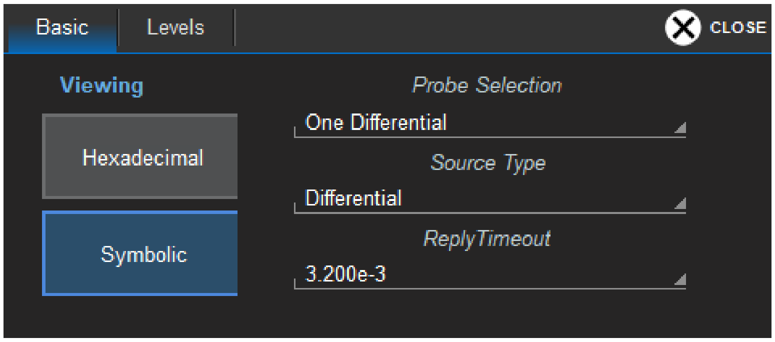
4. Enter the input channels (sources) and select the **Protocol** to be decoded. This selection will drive the other fields that appear.

   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.

DPAUX Decoder Settings

Basic Subdialog



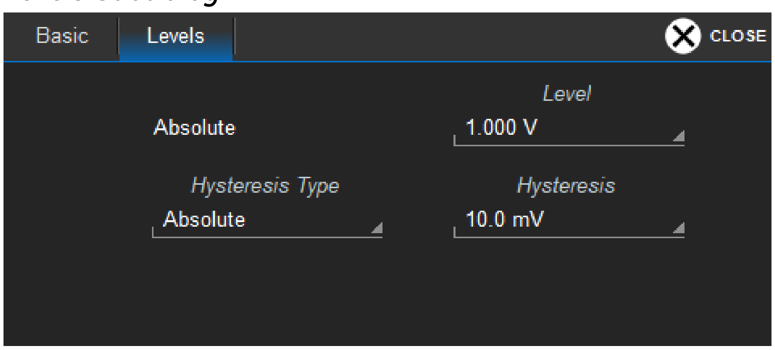
Choose to view decoded data in **Hexadecimal** or **Symbolic** format.

Make a **Probe Selection** matching the physical test setup.

Choose a **Source Type** to decode: only the Aux P line, only the Aux N line, or their Differential.

Enter a **Reply Timeout** after which to stop acquisition if a valid trigger has not been found.

Levels Subdialog



Enter the voltage **Level** that represents the high-low threshold.

Enter a **Hysteresis** the signal must cross to prevent false determinations. Hysteresis can be set in Divisions, Percent (amplitude) or Absolute (voltage level) by changing the **Hysteresis Type**.

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

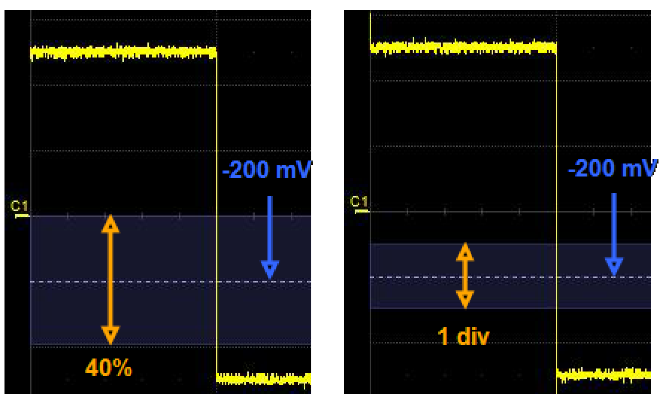
The transition Level appears as a dotted, horizontal line across the oscilloscope grid. If your initial decoding indicates that there are a number of error frames, make sure that Level is set to a reasonable value.

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.

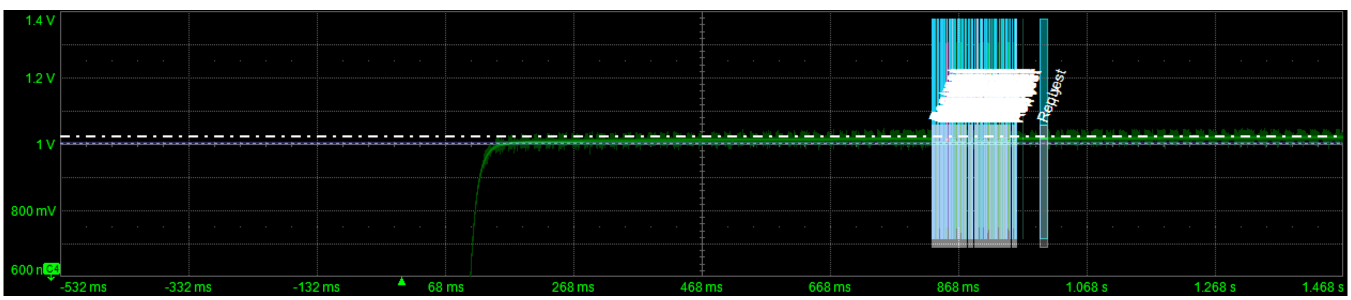
Reading Waveform Annotations

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.

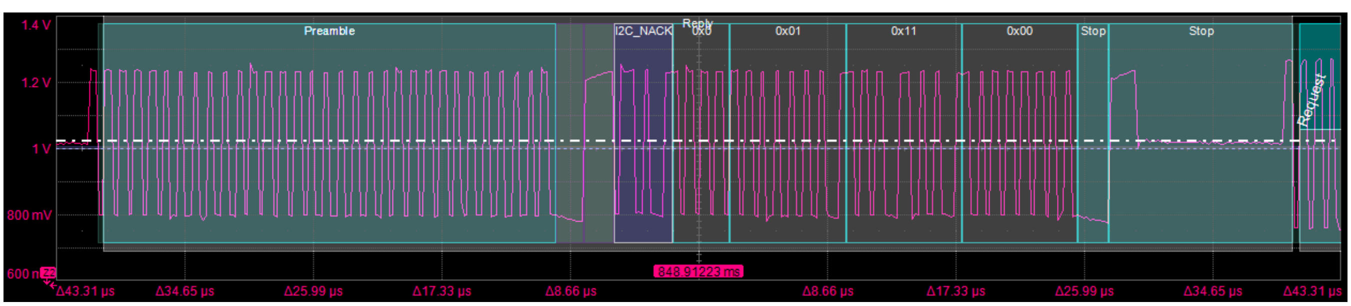
The amount of information shown on an annotation is affected by the width of the rectangles in the overlay, which is determined by the magnification (scale) of the trace and the length of the acquisition. Zooming a portion of the decoded trace by clicking a line in the table will reveal the detailed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

Annotation	Overlay Color (1)	Text (2)
Packet	Navy Blue (behind other fields)	<packet type>
Preamble	Aqua Blue	Preamble
Command Type	Charcoal Grey	<command type>
Command	Purple	<command>
Config Req | Address	Charcoal Grey	<config req> | <address byte>
Stop	Aqua Blue	Stop
Protocol Error	Bright Red (behind other fields)	<error type>

1. Combined overlays affect the appearance of colors. Rectangle border represents actual color.
2. Text in brackets < > is variable. The amount of text shown depends on your zoom factors.



Initial decoding. At this resolution, the overlay is difficult to read.



Zoom of single index showing annotation 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: 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.

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.

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.

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

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
191	929.728 ms	DPAux	Request			
192	929.826 ms	DPAux	Reply	0x01 0x10 0x00 0x00 0x80 0x00		
193	930.700 ms	USB-PD	GoodCRC		0xda87651e	
194	931.043 ms	DPAux	Request			
195	931.195 ms	DPAux	Reply	0x10 0x2d 0x07 0x2c 0x45 0x00 0x52 0x50 0x21 0x00 0x00 0x02 0x2a 0x44 0x80 0xa0		

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
192	929.826 ms	DPAux	Reply	0x01 0x10 0x00 0x00 0x80 0x00		
193	930.700 ms	USB-PD	GoodCRC		0xda87651e	
194	931.043 ms	DPAux	Request			

PacketType	Command...	Command	DPConfigReq / I2C Address	Details
Request	Native	Read	0x01400	Request to Read 16 Byte(s) at Addr 0x01400

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

Note: Decoding DPAUX alongside USB-PD requires both the DPAUX D/DME and USB-PD D/DME option keys.

This extracted data appears on a USB-PD detailed result table.

Column	Extracted or Computed Data
Index (always shown)	Number of the line in the result table
Time	Time elapsed from start of acquisition to start of transmission
Packet Type	Reply | Request.
Command Type	Commad type (e.g., Native or Over 12C)
Command	Command (e.g., Read, Write, ACK, NACK)
DPConfigReq / I2C Address	If Native command, DP config request bytes. If Over I2C command, destination address byte.
Details	Description of packet contents.

DPAux	Time	PacketType	CommandType	Command	DPConfigReq / I2C Address	Details
28	846.124538 ms	Reply	Over I2C	I2C_NACK	0x00000	Invalid I2C reply transaction. AUX_ACK is not present in reply
29	846.210035 ms	Request	Over I2C	Write	0x00b0d	Request to Write 14 Byte(s) at Addr 0x00b0d
30	847.487138 ms	Reply	Over I2C	I2C_NACK	0x00b0d	Invalid I2C reply transaction. AUX_ACK is not present in reply
31	847.579036 ms	Request	Over I2C	Write	0x00100	Request to Write 1 Byte(s) at Addr 0x00100
32	848.906539 ms	Reply	Over I2C	I2C_NACK	0x00100	Invalid I2C reply transaction. AUX_ACK is not present in reply

Section of typical DPAux 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.

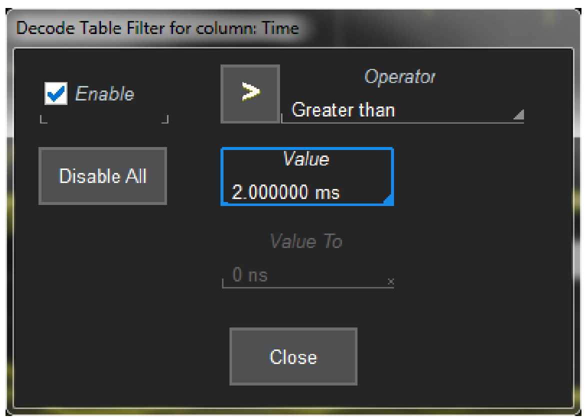
When there are multiple decoders running, each can have its own zooms of the decoding open at once. In this case, multiple rows of the summary table are highlighted to show which indexes are shown in the zooms. These highlights will be different colors to indicate which rows correspond to each decoder.

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 zoom that is not yet turned on.

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.

DPAux	Time	PacketType	CommandType	Command
42	856.605338 ms	Reply	Over I2C	I2C ACK
43	856.691438 ms	Request	Over I2C	Write
44	857.812738 ms	Reply	Over I2C	I2C NACK
45	857.895236 ms	Request	Over I2C	Write
46	859.070739 ms	Reply	Over I2C	I2C ACK

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 **Decode*x*** 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.

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;-4.55526	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;2.274			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;0.00538891140717		8;5.389408	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.

Measure/Graph

The installation of the Measure/Graph package (included with -DME/-DMP or -TDME/-TDMP 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 functionality appears if you have any "M" options installed. However, it will only function properly where the protocol selected supports eye diagrams. Otherwise, the controls will appear shaded. See the TDME catalog or product page for a list of options with "M" support.

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

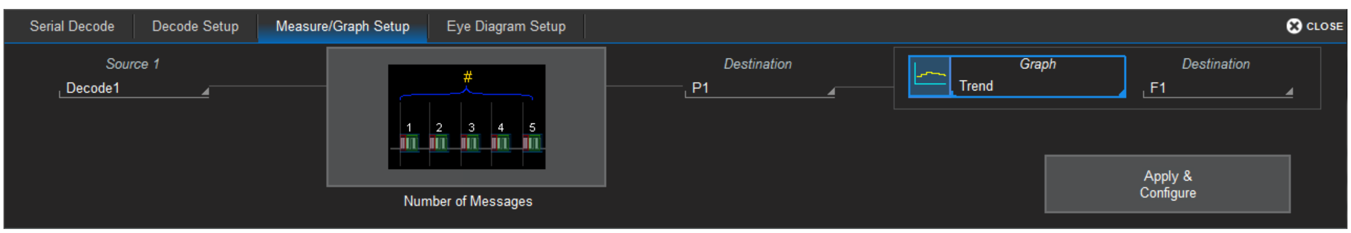
The Measure/Graph package include simplified methods for plotting measurement values as:

- **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.

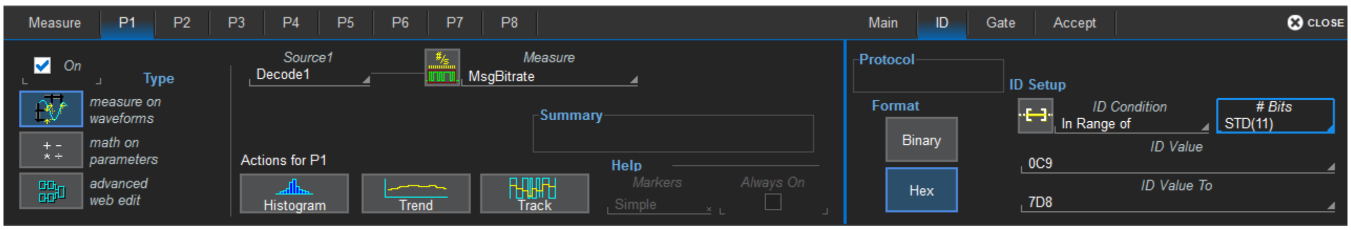
Filtering Measurements

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.

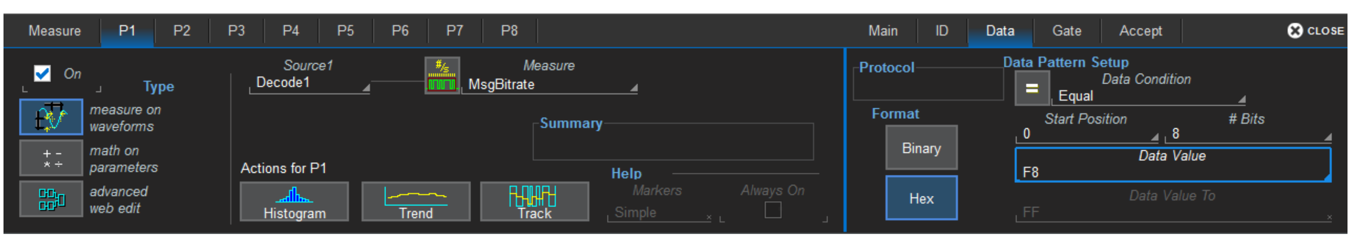


1. On the Main subdialog, choose to **Filter** by **ID** or **ID + Data**.
2. On the **ID subdialog**, choose to enter the ID in **Binary** or **Hex**(adecimal) format.
3. If the field appears, select the **# Bits** used to define the frame ID. (This will change the ID Value field length.)
4. Using the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

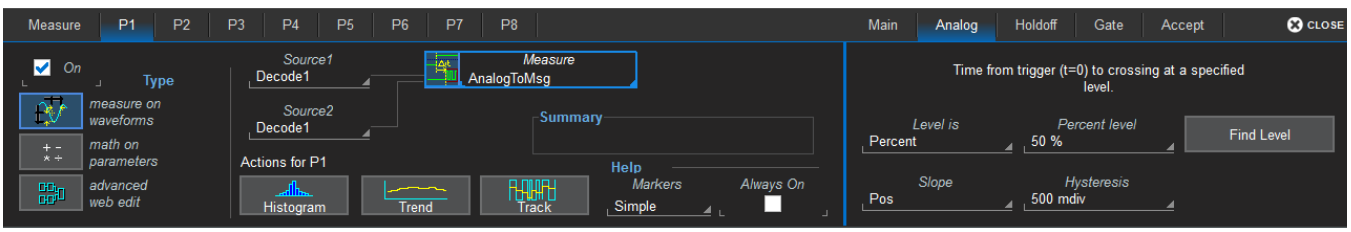
Note: For MsgtoMsg measurements, the data condition is entered twice: first for the Start Message and then for the End Message. The measurement computes the time to find a match to each set of conditions.

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

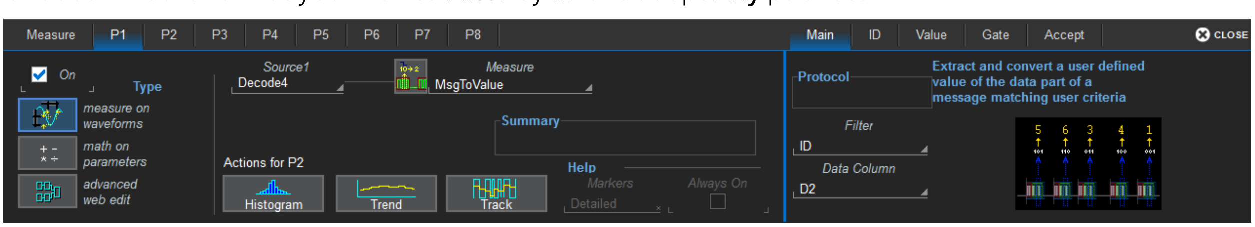
These serial data measurements enable you to take a subset of decoded data (such as sensor data payload) and plot it as a graph. The track of the these measurements is, in effect, a Digital to Analog Converter (DAC) that can display digitally-encoded sensor data as an analog waveform. They are particularly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

Message to Value

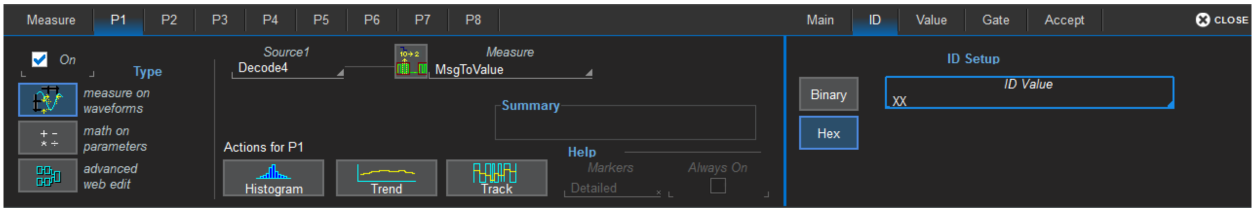
Message to Value enables you to apply oscilloscope features to a subset of the result table and is aimed at protocols with addressed packets containing varying types of data, like CAN, LIN, MIL1553 and many others. With it, you can filter the table by a particular ID to extract and convert decoded data 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:

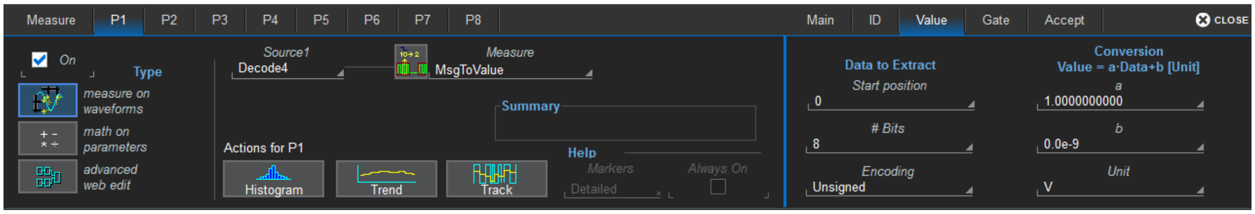
- Choose whether or not you wish to **Filter** by **ID** or accept **Any** packets.



- 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.

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