

Trigger,
Decode,
Measure/Graph,
Eye Diagram

Instruction Manual

for Embedded Protocols:

I2C

SPI

UART and RS-232



Embedded Protocols TDME Instruction Manual

© 2025 Teledyne LeCroy, Inc. All rights reserved.

This document does not contain export-controlled information.

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

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

November 2025
embedded-tdme-im.pdf

Contents

Introduction	1
About the I2Cbus Options	1
About the SPIbus Options	2
About the UART/RS-232bus Options	2
Serial Trigger	3
Requirements	3
Restrictions	3
Serial Trigger Inputs	3
I2C Trigger Set Up	4
RS-232 Trigger Set Up	6
SPI/SIOP/SPI-DDR/SSPI Trigger Set Up	8
UART Trigger Set Up	11
Using the Decoder with the Trigger	13
Saving Trigger Data	13
Serial Decode	14
Decoding Workflow	15
Serial Decode Dialog	15
Decoder Set Up	16
Setting Level and Hysteresis	20
Decoding Digital Inputs	21
Failure to Decode	22
Reading Waveform Annotations	23
Serial Decode Result Table	26
Searching Decoded Waveforms	32
Decoding in Sequence Mode	33
Improving Decoder Performance	34
Measure/Graph	35
Serial Data Measurements	35
Graphing Measurements	36
Measure/Graph Setup Dialog	36
Filtering Measurements	37
Digital to Analog Conversion	39
Eye Diagrams	42
Eye Diagram Setup	42
Mask Failure Locator	44
Appendix A: Automating the Decoder	45
Configuring the Decoder	45
Accessing the Result Table	45
Reading the Structure of the Result Table	45
Modifying the Result Table	47
Technical Support	48

About This Manual

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

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

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

Introduction

Teledyne LeCroy decoders apply software algorithms to extract serial data information from physical layer waveforms measured on your oscilloscope. When displayed on oscilloscopes or in MAUI® Studio remote oscilloscope software, the extracted information overlays the actual physical layer waveforms, color-coded to provide fast, intuitive understanding of the relationship between message frames and other time-synchronous events.

Trigger *and* decode (-TD) options enable you to trigger the oscilloscope acquisition upon finding specific message frames, data patterns, or errors in serial data streams. Conditional filtering at different levels enables you to target the trigger to a single message or a range of matching data.

The premium -DME and -TDME options add a set of serial data timing and eye diagram measurements to the standard trigger and decoder capabilities. Measurements can be graphed as histogram, track or trend for complete serial data analysis. See [Measuring](#) for instructions on using the measure and graphing capabilities. See [Eye Diagram Tests](#) for instructions on using the eye diagram features.

About the I2Cbus Options

I2C is a standardized protocol created by Philips. NXP (formerly Philips Semiconductors) provides a full description of the standard at: www.nxp.com/documents

The I2Cbus TD and TDME options enable triggering on a range of data bytes corresponding with reads or writes to specific sub-address memory blocks in the EEPROM. It can also aid in monitoring data outputs from I2C- based sensors, such as analog-to-digital converters, and triggering when data is outside a safe operating range. Other features include:

- Address conditions in binary and data conditions in hexadecimal to trigger on a range of addresses using "Don't Care" bits
- Frame length triggering
- EEPROM triggering on up to 96 bits (12 bytes) of data at any location within an I2C frame or at a user-defined location in a 2048 byte window
- All permutations of Read, Write, or R/W Don't Care trigger conditions for 7 and 10-bit addresses
- ACK response triggering



Note: If you have installed other -DME or -TDME options, the dialogs for Measure/Graph and Eye Diagram creation will appear when the decoder is open. They may or may not appear "grayed out." We do not guarantee the correct operation of the functionality unless it is explicitly supported by the installation of a TDME option for this protocol.

About the SPIbus Options

SPI was popularized by Motorola but is not a standard, per se. There are variants with differences characterized by how data is clocked, whether data is MSB or LSB format, and whether it is multislave or single-slave.

While SPI has no formal standard, information is often included in the technical documentation for the microprocessor supporting the protocol.

The "Simplified" SPI (SIOP, SSPI, and SPI-Custom) decoders do not require use of a Chip Select line. In its place is the ability to set a minimum Interframe Time. By eliminating the Chip Select line presence requirement, an additional oscilloscope channel is preserved for use with other analog signals.

The SPIbus TD and TDME options enable trigger on SPI messages, targeted to a specific data pattern or a range of data. As with the decoder, the serial data trigger can be configured for any of the many variants of SPI.



Note: If you have installed other -DME or -TDME options, the dialogs for Measure/Graph and Eye Diagram creation will appear when the decoder is open. They may or may not appear "grayed out." We do not guarantee the correct operation of the functionality unless it is explicitly supported by the installation of a TDME option for this protocol.

About the UART/RS-232bus Options

UART is a generic backbone for many proprietary serial data protocols, each with different physical layers. UART has no formal standard. The protocol evolved from mechanical rotating teletypewriter devices. Formats were formalized with the advent of the first electronic computers.

RS-232 is a special case of UART, with a more defined protocol and specific physical layer. The physical layer is defined in Electronic Industries Association (EIA) EIA-RS-232-C and Telecommunications Industry Association (TIA) TIA-232-F. Its protocol layer is not specified; however, UART is commonly implemented. Resources can be found at www.eia.org and www.tiaonline.org.

The UART/RS-232bus TD and TDME options feature:

- Ability to define the UART byte with 9-bit data, with the 9th data bit functioning as an alert bit with a value setable to 0, 1, or X.
- Ability to define as few as 5 bits of data in the UART byte.
- Polarity configurable to either IdleLow or IdleHigh.
- Decoding in **Binary** or **Hex**(adecimal) formats

Triggers can be set on data patterns or Parity errors. Data triggers can be set to find values relative to ranges or individual values. Additional features include:

- Triggering on up to 12 bytes of data in a string up to 2048 bytes long.
- Ability to define the UART byte messages into a single long message packet.



Note: If you have installed other -DME or -TDME options, the dialogs for Measure/Graph and Eye Diagram creation will appear when the decoder is open. They may or may not appear "grayed out." We do not guarantee the correct operation of the functionality unless it is explicitly supported by the installation of a TDME option for this protocol.

Serial Trigger

"T" options provide advanced serial data triggering in addition to decoding. 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 an acquisition of all input signals as configured in the instrument's acquisition settings. This allows decode and analysis of the signal being triggered on, as well as concomitant data streams and analog signals.



Note: The trigger and decode systems are independent, although they are seamlessly coordinated in the user interface and the architecture. It is therefore possible to use the serial trigger without displaying the decoded 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. See [Serial Trigger Inputs](#) for supported input channels and devices.

Restrictions

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

Low-speed serial trigger is not supported on LabMaster series models, only decode, measure and eye diagrams. Some low-speed serial triggers are not supported on WaveMaster models. See the [Serial TDME Datasheet](#) for support by option.

Serial Trigger Inputs

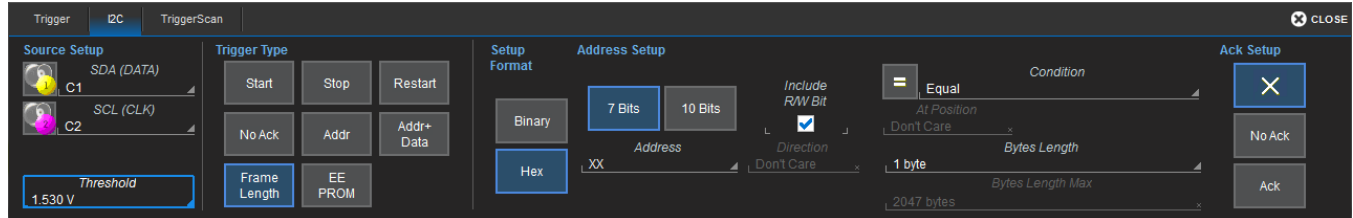
	Analog	Digital
I2C	Any channel or Ext In	Any group or line
SPI/SIOP/SPI-DDR/SSPI	Any channel or Ext In	Any group or line
UART / RS-232	Any channel or Ext In	Any group or line

I2C Trigger Set Up

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 **I2C** Standard button.

Then, working from left to right, make the desired selections from the I2C dialog.



Source Setup

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

In **CLK**, select the clock source input channel.

Use the **Threshold** control to adjust the vertical level for the trigger. This value is used for both data and clock signals.

Trigger Type

These buttons determine which frames/fields are included in the trigger search and which controls are activated on the Trigger setup dialog.

Start triggers upon finding the next Start Frame.

Stop triggers upon finding the next Stop Frame.

Restart triggers upon finding the next Restart.

No Ack triggers upon finding the next packet with no ACK field value.

Address triggers upon finding the next packet containing the specified address. Complete the Address Setup and Ack Setup fields described below.

Address+Data triggers upon finding matching Address field values *and* data patterns. Complete Address Setup, Data Pattern Setup, and Ack Setup fields described below.

Frame Length triggers upon finding packets that satisfy the frame length conditions. Complete the Condition and ByteLength fields under Data Pattern Setup.

EEPROM triggers upon finding matching EEPROM data patterns. Complete Data Pattern Setup and Ack Setup fields described below.

Data Format

For all trigger types that require you to enter a pattern on which to 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.

Address Setup

Choose whether the signal utilizes a **7-bit** or **10-bit** Address length.

Mark **Include R/W bit** if the Read/Write bit is included in the address value (i.e., 8-bit Address values).

Enter the **Address Value**. To use wildcards ("Don't Care" values) in any bit or nibble position, enter an X.



Note: When Hex values are converted to Binary, any non-nibble length wildcards are shown as \$.

Choose a **Direction** of Read, Write, or Don't Care for the Address value. Addresses are always MSB format.

Data Pattern Setup

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 a Data Condition of 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 Posof Value** (instead of the default "Don't Care") to mark a specific **Byte 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).

Length defaults to the length, in bytes, 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.



Note: This field appears as Bytes Length when using a Frame Length trigger. Specify a length from 0 (first byte only) to 2047 bytes (full frame), or to set a range, the **Bytes Length Max**.

Ack Setup

Choose whether to add an Acknowledge bit to the trigger conditions: **X** (Don't Care), **No Ack** or **Ack**.

RS-232 Trigger Set Up

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 **RS-232 Standard** button.

Then, working from left to right, make the desired selections from the RS-232 dialog.

Source Setup

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

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

RS-232 Setup

Use these controls to define the characteristics of the data signal.

Enter the **Bitrate** of the bus to which you are connected. This bitrate selection is dynamically linked to the decoding bitrate (they are always the same value). Use the arrows to scroll a list of standard bit rates, or touch Bitrate and enter a value.

Enter the number of **Data Bits** per byte (not including the START, STOP, or PARITY bits). To trigger on UART messages with a 9th DATA bit used as an Alert, enter 9 in Data Bits, then specify whether the 9th **Alert** bit is a 0, 1, or X (don't care).

Select the **Parity**. Parity must be Odd or Even to trigger on Parity Error.

Enter the number of **Stop Bits**.

Choose either Most Significant Bit (MSB) or Least Significant Bit (LSB) **Bit Order**.



Note: This field defaults to LSB and cannot be changed on an RS-232 trigger.

Select the **Polarity** of the UART signal, either **IdleLow** (Data 1 = High) or **IdleHigh** (Data 1 = Low).



Note: This field defaults to IdleLow and cannot be changed on an RS-232 trigger.

Trigger Type

Choose to trigger on a user-defined **Data** pattern or a **Parity Error**.

Setup Format

Choose to display/enter data 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.

Data Pattern

Use **Data Condition** and **Data Value** together to describe the Data field pattern upon which to trigger. To use a range of values, choose In Range or Out Range. When using a range, also enter the **Data Value To**.

Enter the data **Pattern Length**. This value defaults to the length, in bytes, of the Data Value entry. If the length is changed to a lesser value, it truncates the beginning of the value. If the length is increased, "don't care" bytes are appended to the beginning of the data pattern.

InterFrame Setup

InterFrame Setup is used to define the position of data in RS-232 packets. Choose **None** or **Manual** mode. Manual enables the **Byte Position** and **InterFrame Time** fields for you to enter the values.

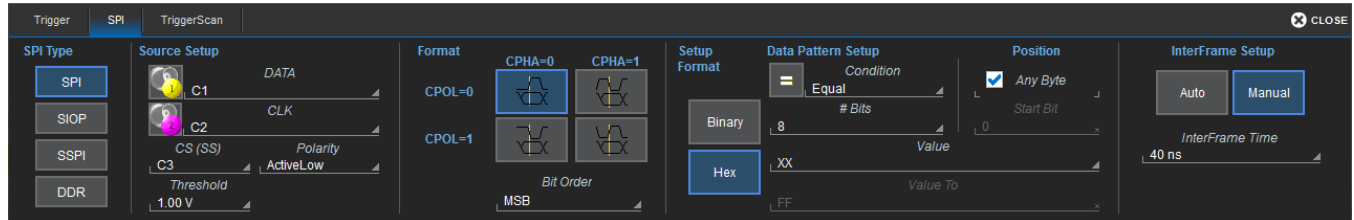
SPI/SIOP/SPI-DDR/SSPI Trigger Set Up

The SPI and Embedded serial decode options enable serial trigger on SPI, SIOP, SPI-DDR or SSPI Command plus Address, or Command, Address and Data patterns.

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 SPI Standard buttons:

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



SPI Type

Choose the type of SPI encoding used on the input signal: **SPI**, **SIOP**, **SPI-DDR**, or **SSPI**.

Source Setup

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

In **CLK** (Clock), select the clock source input channel.

For SPI and SPI-DDR signals, enter the **CS (Chip Select)** input channel and **Polarity**.

Use the **Threshold** control to adjust the vertical level for the trigger. This threshold is used for all input channels.

Format

For SPI and SIOP signals, choose the **CPOL** (Clock Polarity) and **CPHA** (Clock Phase), the phasing of the data to the clock.



Note: SPI microcontrollers and peripherals have settings for CPOL and CPHA that are published in the technical datasheets for those products. The CPOL and CPHA value varies according to the SPI mode in use. See the table below.

SPI Mode	CPOL	CPHA	Data is shifted out on...	Data is sampled on...
0	0	0	falling SCLK and when \overline{CS} activates	rising SCLK
1	0	1	rising SCLK	falling SCLK
2	1	0	rising SCLK and when \overline{CS} activates	falling SCLK
3	1	1	falling SCLK	rising SCLK



Note: There is only one option for SSPI signals. These settings are not used by SPI-DDR.

Choose either MSB or LSB **Bit Order** format.

Setup Format

Setup format determines how you will view/enter the data pattern trigger criteria. Choose either **Binary** or **Hex** (adecimal) format.

Data Pattern Setup

Use **Condition** and **Value** together to describe the data pattern upon which to trigger. All values that meet the criteria will produce a trigger. Up to 12 bytes (96 bits) of data can be entered in Value. If using ranges, also enter the **To Value**.

In **# Bits**, enter the total bits in the trigger pattern (regardless of the data field length in the signal). If the Value entry is shorter than #Bits, it will be padded to this total number using "don't care" characters ("X" if the trigger condition is = or Not =, "0" otherwise).

Position

Check **Any Byte** to trigger on the provided byte pattern, regardless of its position in the frame.

Clearing the Any Byte checkbox will only trigger upon finding the provided pattern when it appears at the given **Start Bit** location.



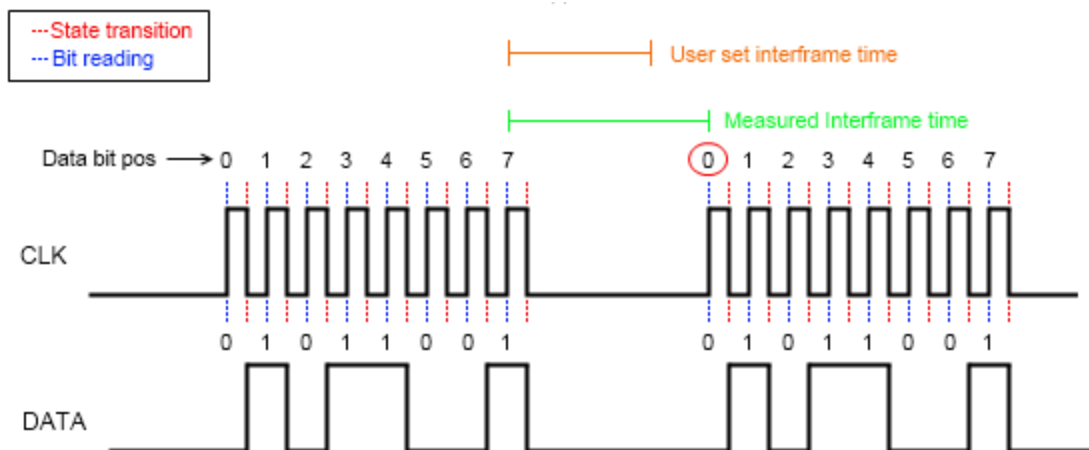
Note: The Start Bit value must respect the pattern's byte boundaries when searching for the provided Value. The trigger will not fire if the Value begins in the middle of a byte.

InterFrame Setup

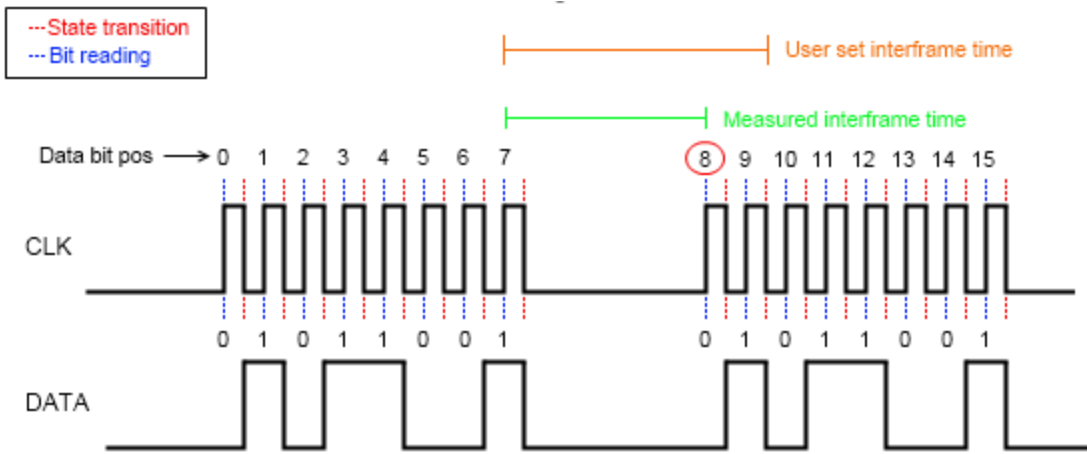
InterFrame Setup is used to determine the position of data in SPI packets. Since packets can include several words, and sometimes a signal is encoded over several words, it's important to establish a bit numbering scheme with a 0 point where counting begins.

Choose **Auto** or **Manual** mode. Manual enables the **InterFrame Time** field where you can provide a specific value. Auto mode sets the InterFrame time to four times the length of a bit.

In Manual mode, the time between each bit reading transition on the CLK signal is read. Inside a word, this time is equal to the length of a bit. At the end of a word, the time until the next transition can be bigger than a bit length. This specific time separation length (i.e., InterFrame Time) defines how the bits are numbered. When the read InterFrame Time is greater than the one you provided, the bit counter is reset to 0 (as shown in the following image).



When the read InterFrame Time is smaller than the one you provided, subsequent bits are considered part of the same packet and continue to accrue sequentially.

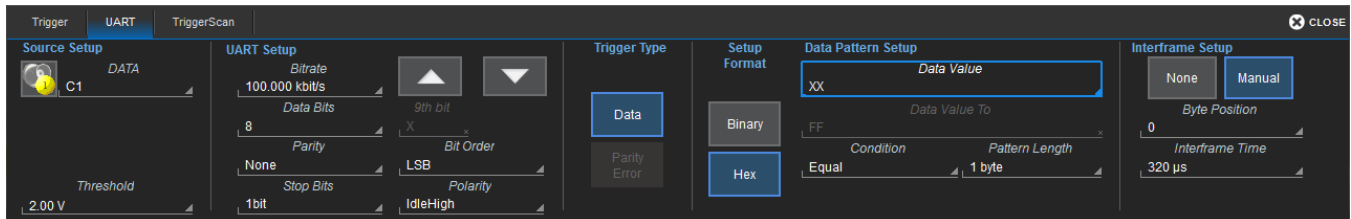


UART Trigger Set Up

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 **UART** Standard button.

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



Source Setup

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

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

UART Setup

Use these controls to define the characteristics of the data signal.

Enter the **Bitrate** of the bus to which you are connected. This bitrate selection is dynamically linked to the decoding bitrate (they are always the same value). Use the arrows to scroll a list of standard bit rates, or touch Bitrate and enter a value.

Enter the number of **Data Bits** per byte (not including the START, STOP, or PARITY bits). To trigger on UART messages with a 9th DATA bit used as an Alert, enter 9 in Data Bits, then specify whether the 9th **Alert** bit is a 0, 1, or X (don't care).

Select the **Parity**. Parity must be Odd or Even to trigger on Parity Error.

Enter the number of **Stop Bits**.

Choose either Most Significant Bit (MSB) or Least Significant Bit (LSB) **Bit Order**.

Select the **Polarity** of the UART signal, either **IdleLow** (Data 1 = High) or **IdleHigh** (Data 1 = Low).

Trigger Type

Choose to trigger on a user-defined **Data** pattern or a **Parity Error**.

Setup Format

Choose to display/enter data 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.

Data Pattern

Use **Data Condition** and **Data Value** together to describe the Data field pattern upon which to trigger. To use a range of values, choose In Range or Out Range. When using a range, also enter the **Data Value To**.

Enter the data **Pattern Length**. This value defaults to the length, in bytes, of the Data Value entry. If the length is changed to a lesser value, it truncates the beginning of the value. If the length is increased, "don't care" bytes are appended to the beginning of the data pattern.

InterFrame Setup

InterFrame Setup is used to define the position of data in UART packets. Choose **None** or **Manual** mode. Manual enables the **Byte Position** and **InterFrame Time** fields for you to enter the values.

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.

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 data sources to one of the four decoder panels on the Serial Data and Decode Setup dialogs. Each decoder can utilize different protocols or data sources, or have other variations, giving you maximum flexibility to compare different signals or view the same signal from multiple perspectives.
- Complete the remaining subdialogs required by the protocol/encoding scheme. Once there is an acquisition in buffer, you will see a [result table](#) and an [annotation overlay](#) on the waveform trace showing the decoded data.
- Work with the annotated waveform, result table and other functionality to analyze the decoding.

Decoding Workflow

We recommend the following workflow for effective decoding:

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



Note: See [Failure to Decode](#) for more information about the required acquisition settings.

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

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

Serial Decode Dialog

Choose **Analysis > Serial Decode** to access the serial decoder dialogs.

The Serial Decode dialog is best used to turn on/off decoders after they've been fully configured on the [Decode Setup](#) dialog. You cannot make all required decoder settings here.

- To turn on decoders, on the same row as Decode *N*, check **On**. If there is a valid acquisition, a result table and annotated waveform will appear.
- To turn off decoders, deselect the On boxes individually, or touch **Turn All Off**.



Note: If you change the **Protocol** to decode, the last settings configured for that protocol will be resumed.

If the decoder supports data search, use the **Search** button to open the Zoom dialogs. Enter search criteria on the Search subdialog to the right of each *Z_n* dialog.

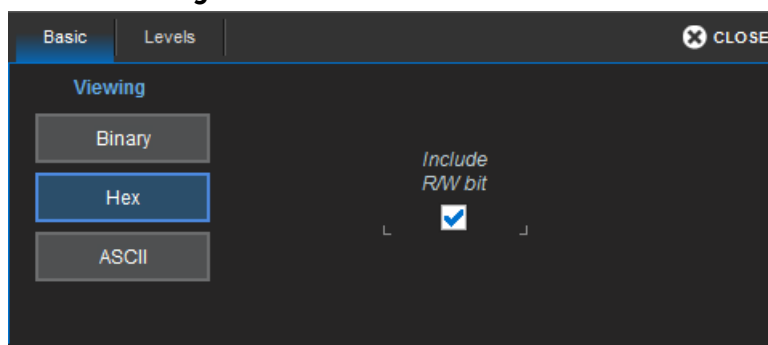
Decoder Set Up

Use Decode Setup and its subdialogs to configure decoders.

1. 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.
4. Select the highest-level **Protocol** to be decoded and the inputs (sources). The Protocol selection will influence the remainder of the set up.

I2C Decoder Settings

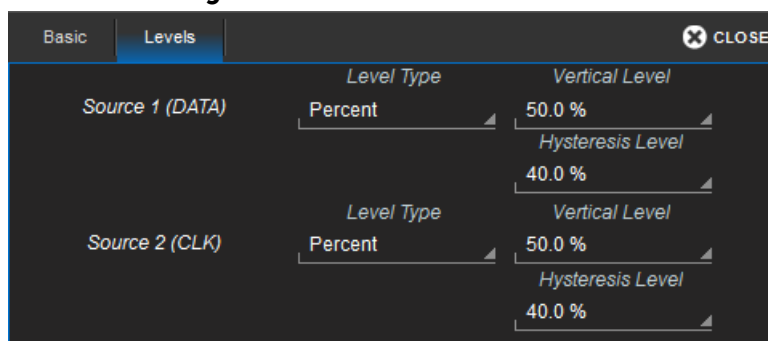
Basic Subdialog



Under **Viewing**, choose to view/enter data in **Binary**, **Hexadecimal**, or **ASCII** formats.

Some engineers consider the address pattern to include the R/W bit (i.e., 8-bits). If your signal utilizes 8-bit addresses, mark **include the R/W bit**.

Levels Subdialog



For both the DATA and CLK (Clock) source signals, 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. For guidelines, see Setting Level and Hysteresis.

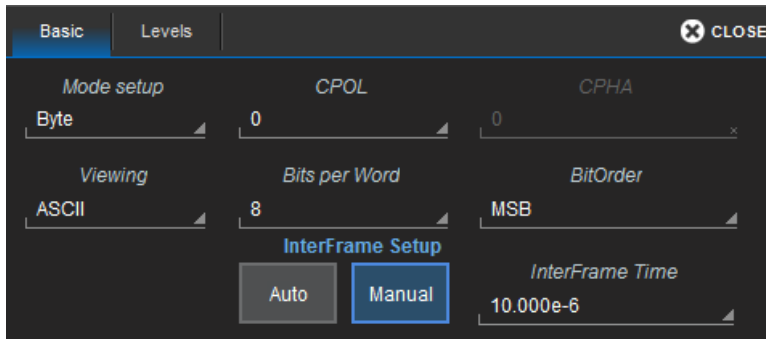


Note: Data and clock sources can be set to different levels, but they are typically the same level.

SPI Decoder Settings

SPI/Basic Subdialog

For SPI and SPI-DDR decoders, the top subdialog will be named after the protocol. For SIOP, SSPI, and SPI-Custom decoders, this dialog will be named Basic. They are the same except for Chip Select and InterFrame settings.



Mode setup - choose to decode by Byte or Frame.

CPOL (Clock Polarity) and **CPHA** (Clock Phase) - enter 0 or 1



Note: SPI microcontrollers and peripherals have settings for CPOL and CPHA that are published in the technical datasheets for those products. The CPOL and CPHA value varies according to the SPI mode in use. See the table below.

SPI Mode	CPOL	CPHA	Data is shifted out on...	Data is sampled on...
0	0	0	falling SCLK and when $\overline{\text{CS}}$ activates	rising SCLK
1	0	1	rising SCLK	falling SCLK
2	1	0	rising SCLK and when $\overline{\text{CS}}$ activates	falling SCLK
3	1	1	falling SCLK	rising SCLK



Note: Identical settings for Clock Polarity and Clock Phase are on the SPI serial trigger setup dialog. If you have chosen to link this decoder to the trigger, shared settings are copied from the trigger setup and dynamically updated in both locations.

In **Viewing**, choose to view/enter decoded values in Binary, Hexadecimal, ASCII, or Decimal format.

Bits per Word and **BitOrder** - Enter the number of bits per word and choose from MSB or LSB bit order format.

SPI and SPI-DDR Protocols

CS Polarity - Set Chip Select Polarity to either ActiveLow or ActiveHigh. Bytes will only be decoded during the selecting polarity.

Decode Outside CS - Check if you want to decode all SPI bytes instead of only those active during the CS Polarity state. When this is selected, the decoding starts at the first transition of CS (in either polarity) and continues until the end of the packet regardless of the state of CS. When it is not selected, decoding will only occur during the CS Polarity state.

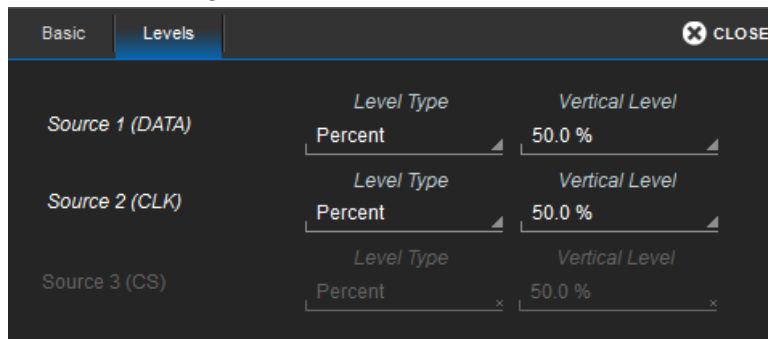


Note: Decoding always starts at the first CS transition. If you have scaled the acquisition so that there is only one packet in either state prior to the transition showing, it will not be decoded.

SIOP and SPI-Custom Protocols

SIOP, SSPI and SPI-Custom decoders do not use Chip Select. Instead, choose to set the InterFrame period **Auto** (matically) or **Manual**(ly). Auto sets a time that is is (typically) 4x a single bit time and less than the interframe time between different message packets. For a manual setting, enter the **InterFrame Time**.

Levels Subdialog

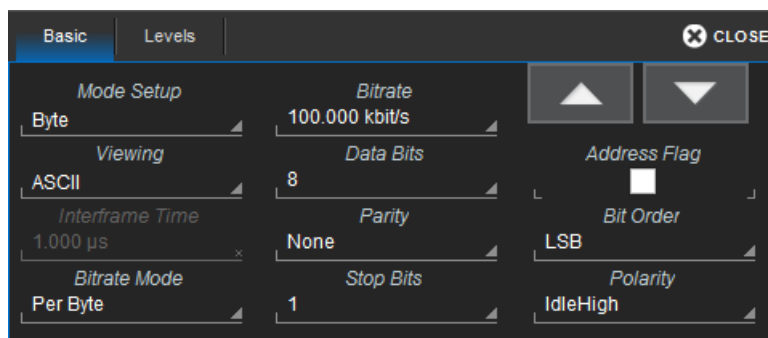


For DATA, CLK (Clock), and CP (Chip Select) source signals, enter the vertical **Level** used to determine the edge crossings of the signal. This value will be used to determine the bit-level decoding. For guidelines, see Setting Level and Hysteresis.

UART/RS-232 Decoder Settings

Basic Subdialog

Use these controls to define the characteristics of the data signal.



In **Mode Setup**, choose to view the decoding per Byte or per Message frame. The result table will show one Byte or Message per line. If you are using Message mode, also enter the **InterFrame Time** between messages.

In **Viewing**, choose to view/enter values in Binary, Hexadecimal, ASCII, or Decimal format.

Choose a **Bitrate Mode** to determine how the bit rate is calculated:

- Per Byte computes bit rate for each byte separately, then takes the average of those to calculate bit rate of frame. This method of computing bit rate ensures that the gap between each byte is not included in the overall frame bit rate.
- Per Message computes bit rate using the full length of frame divided by total number of bits.

Enter the **Bitrate** of the bus to which you are connected. The value should be correct within 5%. A mismatched bit rate will cause various confusing side effects on the decoding, so it is best to take time to correctly adjust this fundamental value. Use the arrows to scroll a list of standard bit rates, or touch Bitrate and enter a value.



Tip: If you are not sure about the value, apply the Bit Rate measurement parameter to a short acquisition to determine the exact bit rate of your signal.

Enter the number of **Data Bits** per byte (not including the START, STOP, or PARITY bits).

Select the **Parity**. Parity must be Odd or Even to trigger on Parity Error.

Enter the number of **Stop Bits**.

Check **Address Flag** to decode the last bit before the STOP bit to determine if it is an address or data byte.

Choose either Most Significant Bit (MSB) or Least Significant Bit (LSB) **Bit Order**.



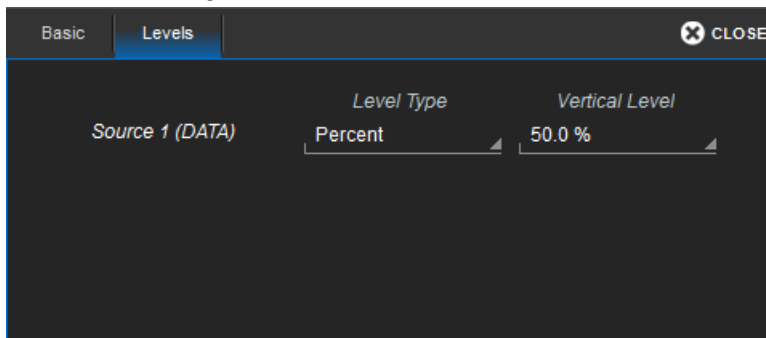
Note: On an RS-232 decoder, this field defaults to LSB and cannot be changed.

Select the **Polarity** of the UART signal, either **IdleLow** (Data 1 = High) or **IdleHigh** (Data 1 = Low).



Note: On an RS-232 decoder, this field defaults to IdleLow and cannot be changed.

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. For guidelines, see Setting Level and Hysteresis.

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

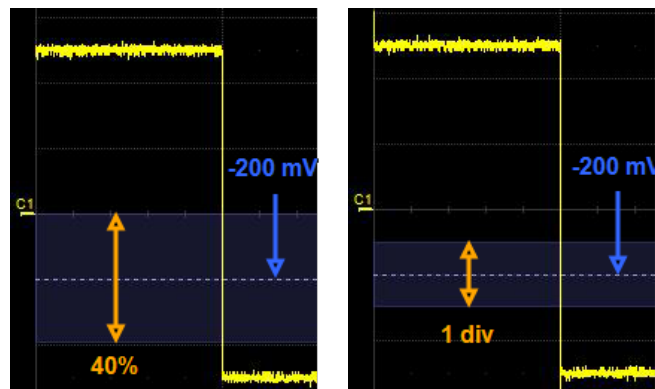
The transition Level appears as a dotted, horizontal line across the oscilloscope grid. If your initial decoding indicates that there are a number of error frames, make sure that Level is set to a reasonable value.

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.

Decoding Digital Inputs

While digital lines can be used as sources for most decoders, it is important to make sure that the logic setup for the digital inputs is consistent with the levels set for the decoder and appropriate for the signal.

If you receive a poor decoding when using digital inputs, do the following to troubleshoot.

Assess Signal Levels Using Analog Input

It is good practice to establish the exact signal characteristics using an analog input method prior to using a digital input method.

Connect the signal to an oscilloscope channel (C_n) and set the amplitude, top and base measurement parameters on it. If the decoder utilizes multiple sources, do this for all signals.

Open the decoder setup dialogs and try decoding the analog signal using the default levels. Adjust Level and Hysteresis according to the measured amplitude until you get a good decoding of the analog signal.

Set Appropriate Logic Levels on Digital Dialogs

When a digital input device is connected to the oscilloscope, setup dialogs for digital groups ($Digital_n$) are added to the Vertical menu. Use the $Digital_n$ dialogs to set the levels used for logic determination by the input device.



Note: Digital1 is turned on by default with all the lines added to the group.

Open the Logic Setup tab and either select a Logic Family that is appropriate to the signal amplitude, or choose User Defined and enter your custom logic Threshold and Hysteresis levels based on what you determined when decoding the analog signal.



Tip: Selecting different Logic Families will display what determination Threshold and Hysteresis values they use on the dialog.

For example, a signal with approximately 1.25 V amplitude, 0 V based, does not match any of the Logic Families but lends itself to a User Defined setup with Threshold at 0.625 V (~50%) and a Hysteresis of 100 mV if the edges are clean.

Continue to adjust your digital group Logic Setup and your decoder Levels in synch until you receive a good digital decoding.



Note: For the MSO Digital Leadset, logic can only be set per digital leadbank, not line by line as on the HDA125. Be sure the lines you use for decoding are all from the same leadbank and set appropriately.

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.

However, there are no tests for the following, so you will want to pay attention to:

- **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, CTLE/DFE equalizers can be used to improve the quality of the signal being decoded. Use the waveform output by the equalizer as the input to the decoder.
- **Incorrect Threshold Levels.** Configuring an incorrect threshold level can result in no, incomplete or incorrect decoding. With analog traces, the threshold level can be adjusted after the acquisition but this is not possible with digital waveforms. You must make sure that the appropriate logic levels are configured for digital inputs, as described in the [Decoding Digital Inputs](#), before data is acquired. When troubleshooting, it is best to use analog channels instead of digital channels because of the additional information available.



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.

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.

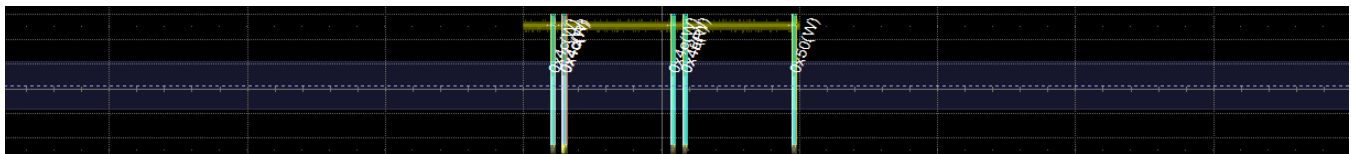
The amount of information shown on an annotation is affected by the width of the rectangles in the overlay, which is determined by the magnification (scale) of the trace and the length of the acquisition. Zooming a portion of the decoded trace by clicking a line in the table will reveal the detailed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

I2C Waveform Annotations

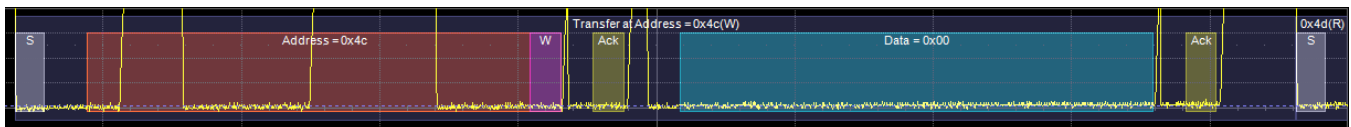
These overlays appear on an I2C waveform or its zoom trace:

Annotation	Overlay Color (1)	Text (2) (3)
Packet	Navy Blue (behind other fields)	<type> = <value> (< R W >)
Start/Stop Bits	Grey	< S P >
R/W Bits	Hot Pink	< R W >
Address	Brick Red	Address = <value>
Acknowledge	Olive	ACK
No Acknowledge	Bright Red	NACK
Payload Data	Aqua Blue	Data = <bytes>
Protocol Error	Bright Red (behind other fields)	<error type>

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. Payload data values shown in binary, hexadecimal, or ASCII depending on your decoder selection. Other values shown in hexadecimal only.



Initial decoding. At this resolution, little information appears on the overlay.



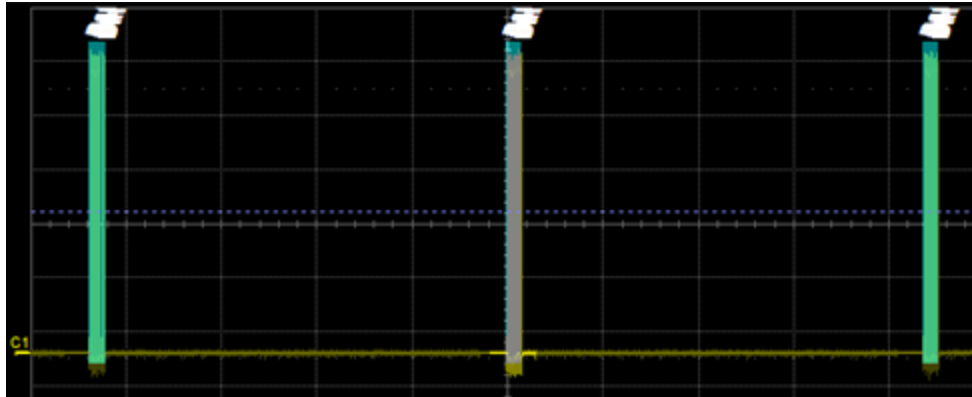
Zoom of single index showing annotation details.

SPI Waveform Annotations

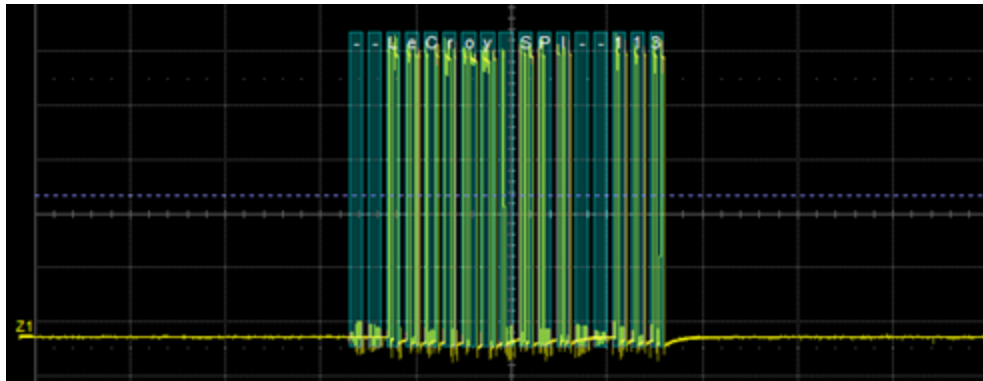
These overlays appear on a SPI waveform or its zoom trace:

Annotation	Overlay Color (1)	Text (2) (3)
Data byte	Green	<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 binary, hexadecimal, ASCII, or decimal depending on your decoder selection.



Initial decoding of SIOP waveform.



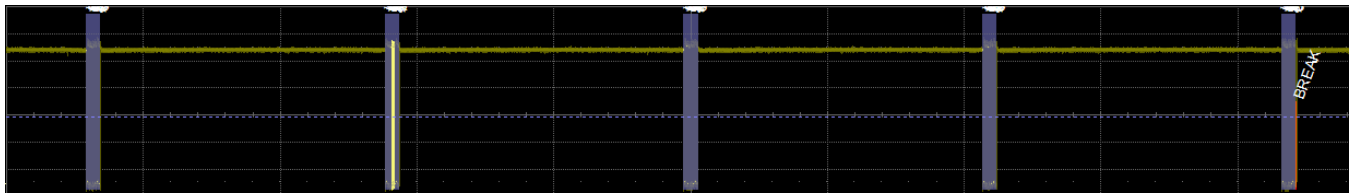
Zoomed SIOP waveform, showing detailed annotations.

UART/RS-232 Waveform Annotations

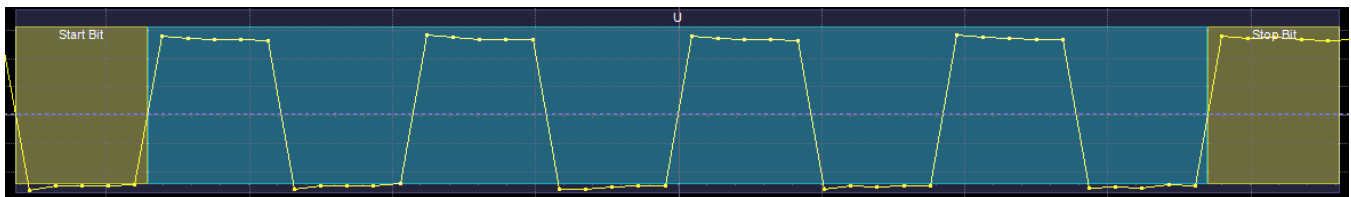
These overlays appear on decoded UART and RS-232 waveforms and their zoom traces.

Annotation	Overlay Color (1)	Text (2) (3)
Byte	Navy Blue (behind other fields)	<value>
Start/Stop Bits	Yellow	< S P >
Payload Data	Aqua Blue	
Protocol Error	Bright Red (behind other fields)	<error type>

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 binary, hexadecimal, ASCII, or decimal depending on your decoder selection.



Initial decoding.



Zoom showing annotation details.

Serial Decode Result Table

When you choose to turn **On** or to **View Decode**, provided there is a valid acquisition to decode using that protocol, a table summarizing the decoder results appears below the grids. This result table shows all data decoded during the most recent acquisition, even when there are too many bursts for the waveform annotation to be legible.



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

Table Rows

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

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



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

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

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

Table Columns

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

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 (if used)
Status	Any decoder messages; content may vary by protocol

Index	Time	Protocol	Message	Data
9	2.62316 ms	I2C	r	
10	2.84000 ms	I2C	r	'eetings from LeCroy..9..Tw..n7i..1ni.g).W.<")...L.Z.O*...f..... 8
11	4.11914 ms	I2C	r	
12	4.32165 ms	I2C	M	' LeCroy I2C 30'
13	43.6681 ms	I2C	N	' LeCroy I2C 31'

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

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

Index	Time	Protocol	Message	Data
10	2.84000 ms	I2C	r	'eetings from LeCroy..9..Tw..n7i..1ni.g}.W.<"')...L.Z.O*...f..... 8
		Addr Length	Address	R/W Len... Data
		10	r	R 255 'eetings from LeCroy..9..Tw..n7i..1ni.g}....
11	4.11914 ms	I2C	r	
12	4.32165 ms	I2C	w	'LeCroy I2C 30'

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

Exporting Result Table Data

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

1. Choose **Analysis > Serial Decode** and 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\



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\

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. From the oscilloscope menu bar, choose **File > Save Table** and select the desired **Decode N** as the source.

I2C Result Table

Column	Extracted or Computed Data
Index (always shown)	Number of the line in the table
Time	Time elapsed from start of acquisition to Start of Frame
Address Length	Number of Address field bits, either 7 or 10
Address	Extracted Address field bits, in either binary or hexadecimal
R/W	Extracted Read/Write field bits
Length	Number of Data field bytes
Data	Extracted Data field bytes
Status	I2C errors found in the decoding

I2C	Time	Addr Length	Address	R/W	Length	Data	Status
1	-16.7062 μs	7	0x4c	W	16	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 30	
2	4.11914 ms	7	0x4c	W	1	0x00	
3	4.32165 ms	7	0x4d	R	16	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 30	
4	43.6681 ms	7	0x4e	W	16	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 31	
5	47.8039 ms	7	0x4e	W	1	0x00	
6	48.0064 ms	7	0x4f	R	16	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 31	
7	87.3530 ms	7	0x50	W	16	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 32	

Section of typical I2C detailed result table.

SPI Result Table

Column	Extracted or Computed Data
Index (always shown)	Number of the line in the table
Time	Time elapsed from start of acquisition to start of message
Data	Extracted data values
Bit Rate/Byte or Bit Rate/Msg	Bit Rate for corresponding byte or message of data, depending on your Mode Setup selection

SIOP	Time	Data	Bit Rate/Byte
1	-103.818 μ s	"	100.015 kbit/s
2	-5.000 μ s	'L'	100.012 kbit/s
3	93.733 μ s	'e'	100.010 kbit/s
4	192.552 μ s	'C'	100.013 kbit/s
5	291.290 μ s	'r'	100.021 kbit/s
6	390.022 μ s	'o'	100.015 kbit/s
7	488.755 μ s	'y'	100.007 kbit/s

Section of typical SPI detailed result table.

UART/RS-232 Result Table

Column	Extracted or Computed Data
Index (always shown)	Number of the line in the table
Time	Time elapsed from start of acquisition to start of message
Data	Decoded data payload bytes
Data Length	Data field length
A/D	If the Address flag is checked, this column will show whether the byte is address or payload data
Parity	Parity bit
Bit Rate	Transmission bit rate per Byte or Message, depending on Mode Setup selection

UART	Time	Data	DataLength	A/D	Parity	Bit Rate
23	-43.31235 ms	'C'	8	1		98.226 kbit/s
24	-43.19834 ms	'r'	8	1		101.257 kbit/s
25	-43.08237 ms	'o'	8	1		98.217 kbit/s
26	-42.96837 ms	'y'	8	1		101.220 kbit/s
27	-42.85239 ms	''	8	1		98.210 kbit/s
28	-42.73835 ms	'U'	8	1		101.242 kbit/s
29	-42.62237 ms	'A'	8	1		98.220 kbit/s

Section of typical UART detailed result table.

Using the Result Table

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

Zoom & Search

Touching any cell of the table opens a zoom centered around the part of the waveform corresponding to the index. The *Zn* dialog opens to allow you to rescale the zoom, or to [Search](#) the decoding. 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 zoom are highlighted, as is the zoomed area of the source waveform. The highlight color indicates the zoom shown (Z1 yellow, Z2 pink, etc.). As you adjust the zoom scale, the highlighted area may expand to several rows or fade to indicate that only a part of that Index is now shown in the zoom.

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



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

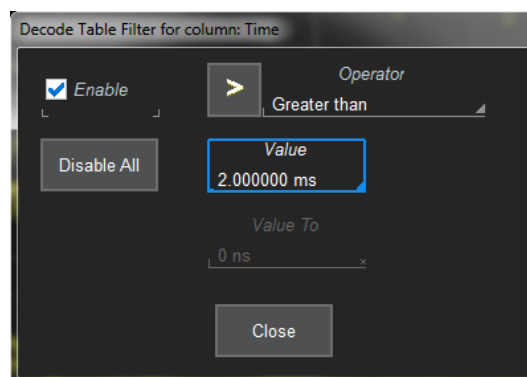
Index	Time	Protocol	Message	Data	CRC	Status
1	-16.706 μs	I2C	0x4c	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 30		
2	-16.706 μs	I2C	'L'	'LeCroy I2C 30'		
3	4.119136 ms	I2C	0x4c	0x00		
4	4.119136 ms	I2C	'L'	'		
5	4.321649 ms	I2C	0x4d	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 30		
6	4.321649 ms	I2C	'M'	'LeCroy I2C 30'		
7	43.66809 ms	I2C	0x4e	0x00 00 4c 65 43 72 6f 79 20 49 32 43 00 00 33 31		

Filter Results

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

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

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

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 coloring in red the Bitrate shown in the table.

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.

i Idx = 15 (decimal) found at Row 55 Column 0 going Left

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 sometimes 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)*.

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.

If your goal is downstream processing, 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 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.

Measure/Graph

Not supported on WaveSurfer and HDO4000 series oscilloscopes.

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



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

Serial Data Measurements

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



Note: When working outside the TDME software, measurements appear in the Serial Decode sub-menu of the Measure Setup menu and may have slightly different names. The measurements are made 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.
Column to Value	Column	Extracts the data in a single column of the result table to a measurement parameter location, with no transformation of value.
Message to Analog	ID, Data, Analog, Column	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, Column	Computes time from start of first message that meets conditions to start of the next message that meets conditions.
Time at Message	ID, Data, Column	Computes time from trigger to start of each message that meets conditions.
Analog to Message	ID, Data, Analog, Column	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, Column	Computes time difference between two messages on a single decoded line.
Bus Load	ID, Data, Column	Computes the load of selected messages on the bus (as a percent).
Message Bitrate	ID, Data, Column	Computes the bitrate of selected messages within the decoded stream.
Number of Messages	ID, Data, Column	Computes the total number of messages in the decoding that meet conditions.

Graphing Measurements

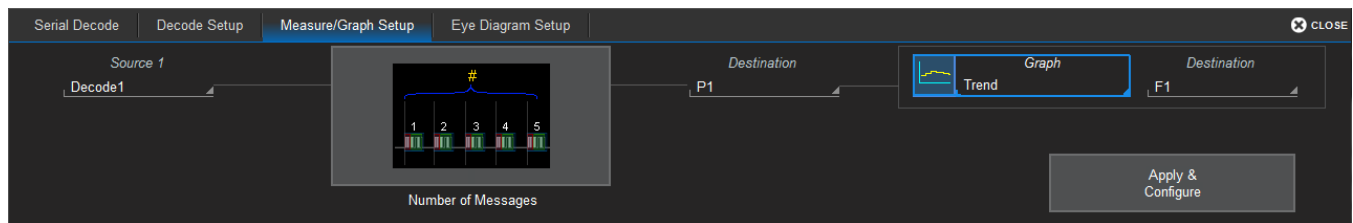
The Measure/Graph package include simplified methods for plotting measurement values as:

- **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 decoding in a separate grid.

Measure/Graph Setup Dialog

Use the Measure/Graph Setup dialog to select the parameter to apply to the decoded waveform while simultaneously graphing the results.



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 those 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 give results for only rows with specified IDs, data patterns or values in selected columns of data. When the measurement uses multiple sources, each source can have the same or a different filter.

Main

- 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.
- From the **Filter** drop-down on the Main subdialog, choose the message elements on which to filter:
 - Any** displays results for any message (no filter).
 - ID** restricts the measurement to messages with a specific ID value.
 - ID + Data** restricts the measurement to messages with extracted data matching the filter.
 - Col(umn)** restricts the measurement to messages with matching values in the selected result table column(s). It can be applied to timing measurements such as MsgtoMsg and MsgtoAnalog. It allows you to use a column other than ID or Data as the start/end point for the measurement.

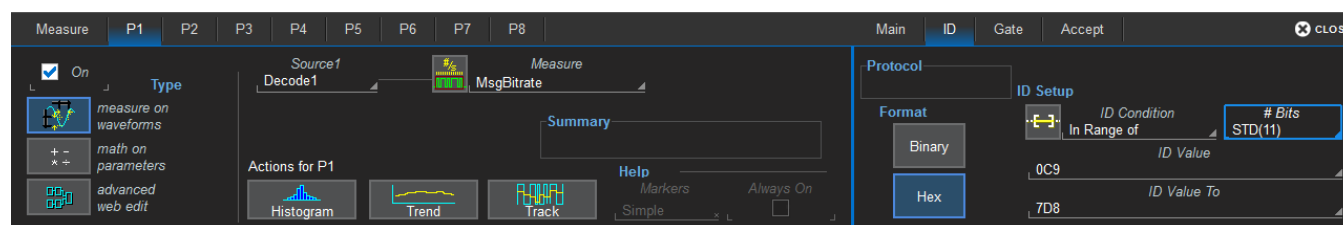


Tip: To make timing measurements on different parts of message frames within the same decoding, choose the same Decode as measurement Source1 and Source2, Column filter both, but choose a different column for Col1 and Col2. See the example below.

- Set filter conditions on the right-hand subdialogs that appear next to the P_n dialogs.

ID Filter

Settings on this dialog may change depending on the protocol.

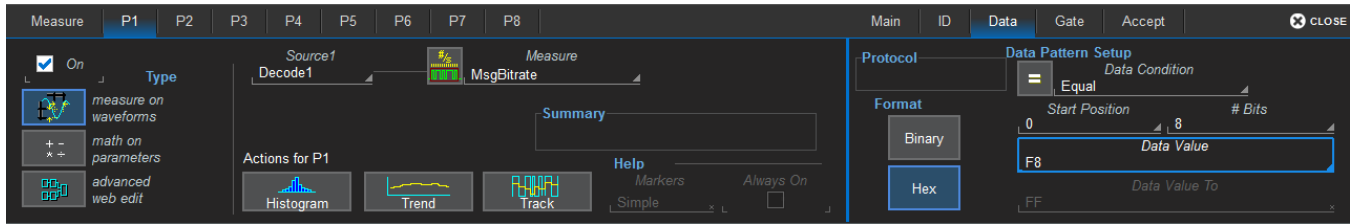



- On the **ID subdialog**, choose to enter the ID in **Binary** or **Hex**(adecimal) format.
- If the field appears, select the **# Bits** used to define the frame ID. (This will change the ID Value field length.)
- Using the **ID Condition** and **ID Value** controls, create a condition statement that 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.

ID + Data Filter

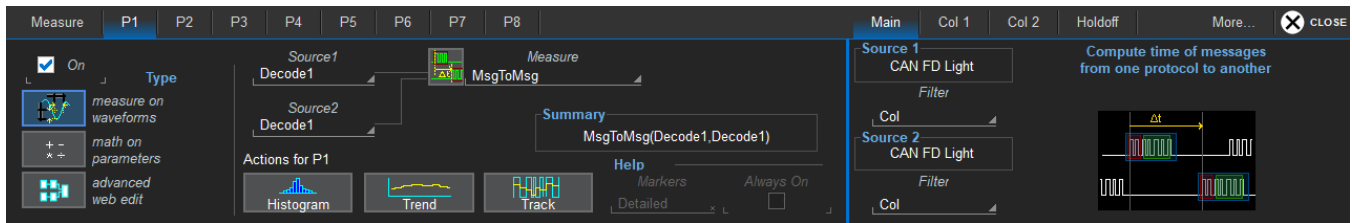


 **Tip:** For a Data only filter, leave the ID subdialog set to ID = ALL.

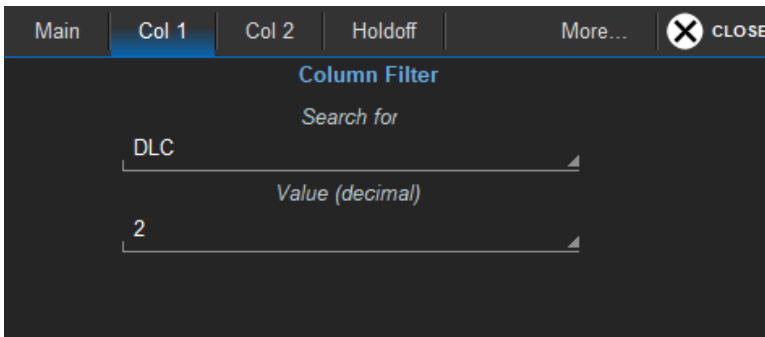
1. On the Data subdialog, choose to enter data in **Binary** or **Hex**(adecimal) format.
2. Create a Boolean **Data Condition** describing the **Data Value(s)** to include in the measurement. Use "X" as a wild card ("Don't Care") in any position where the value doesn't matter.
3. Optionally, enter a **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 filter condition is entered twice: first for the Start Message and then for the End Message. The measurement computes the time between finding a match to each condition.

Column Filter



1. On the Measure/Graph Main subdialog, choose the **Filter** type **Col**(umn) for those sources where you want to specify a column of data (i.e., message frame) as the start/end point for a timing measurement.
2. On the **Col 1** and **Col 2** subdialogs, choose the result table columns to **Search for the Value**.



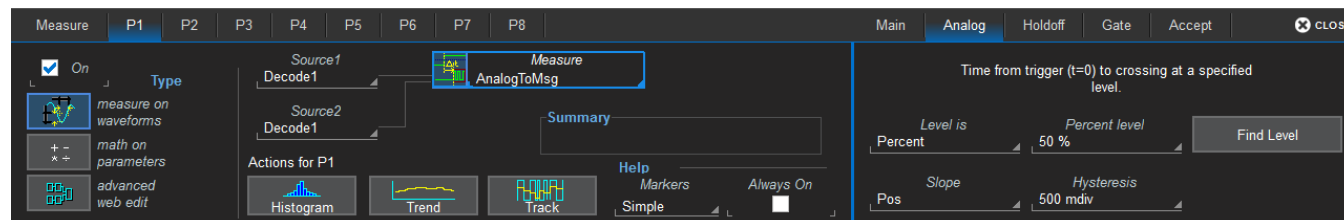
 **Note:** In measurements where both sources are Column filtered, the measurement is made from the first matching value in Col 1 to the first matching value in Col 2.

Analog Filters

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 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.



Note: Examples shown are taken from CAN decoders, but the functionality works the same for all protocols.

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 & Configure**.

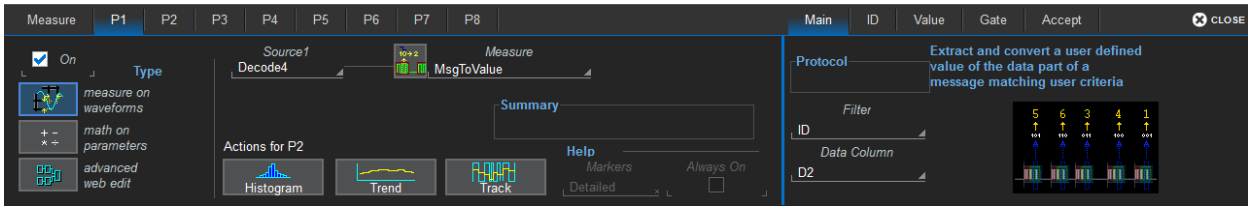
Message to Value

The MsgToValue measurement enables you to graph or test only a subset of a decoder result table. It 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 via a parameter that can be used for other math or measurement processes, such as the Track function.

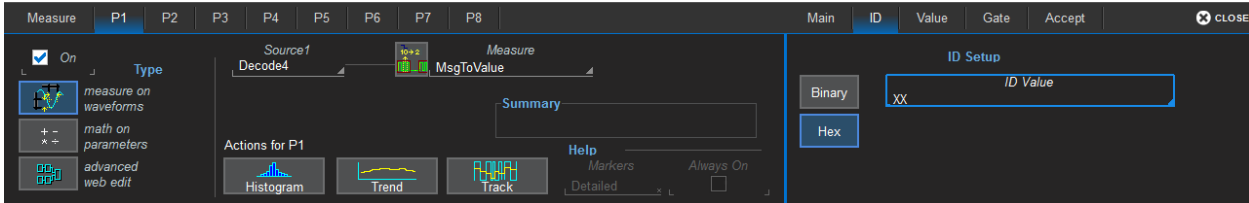
For TDME users, MsgToValue can be assigned to a Source decoder and parameter right on the Measure/Graph Setup dialog. However, MsgToValue requires several filter selections, so it is necessary to select **Apply & Configure** to complete the set up on the Measure set up subdialogs.

continued

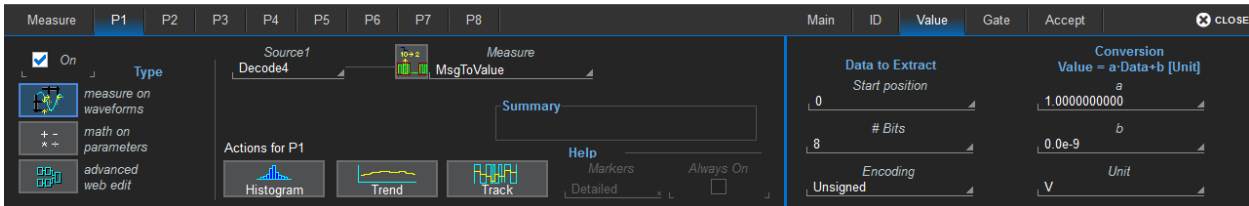
1. Choose whether or not you wish to **Filter** by ID or accept **Any** packets, then select the **Data Column** from which to pass values.



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



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



Follow these steps to define the values to extract:

- a. SYMBOLIC users: touch **Browse DBC/ARXML**. Expand the symbol file, then click on the desired symbol to filter on occurrences of that symbol.



Note: This button will not appear if your installation does not support symbolic decoding.

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

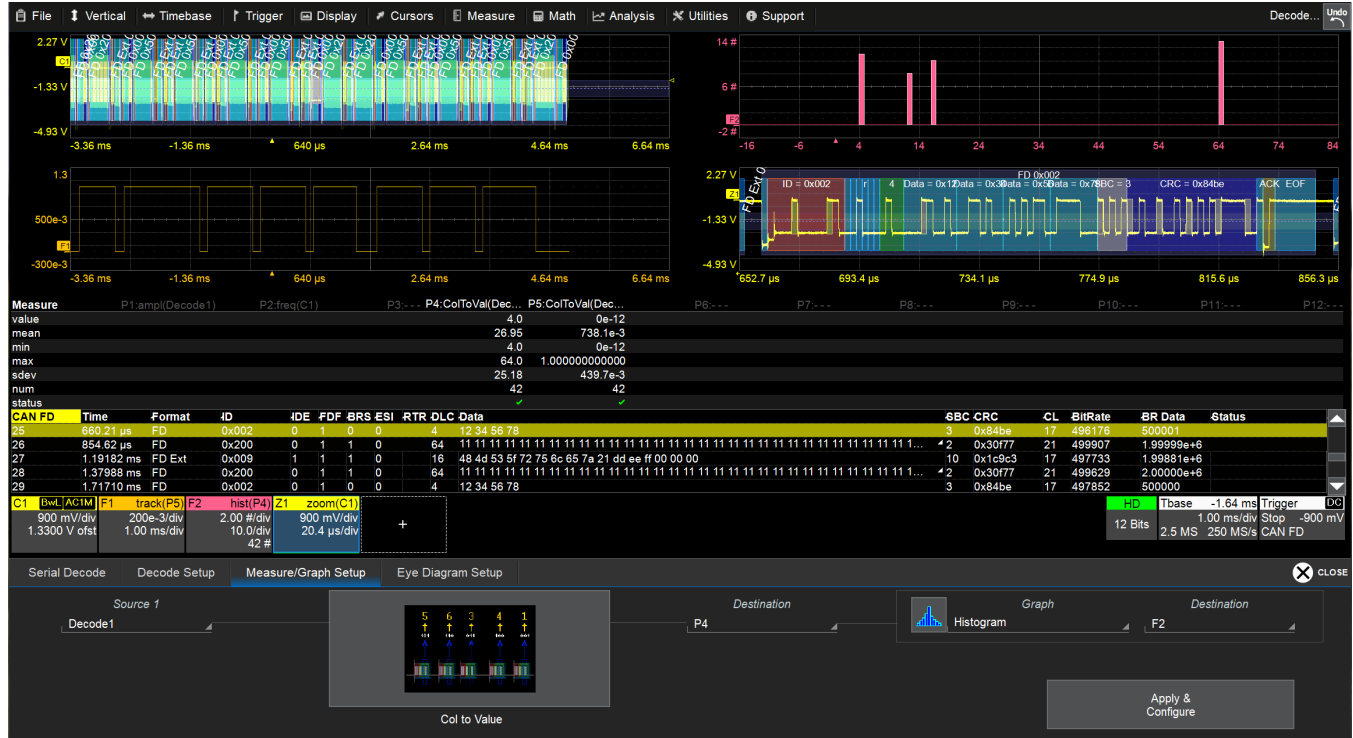
Column to Value

Similar to MsgToValue, ColToValue acts as a special "pass thru" of one column of decoder table values via a parameter configured with this measurement, allowing you to graph, run Pass/Fail tests or perform other calculations on just these values by using the parameter as the source of other functions. It is very similar to using Excel to graph a column of data.

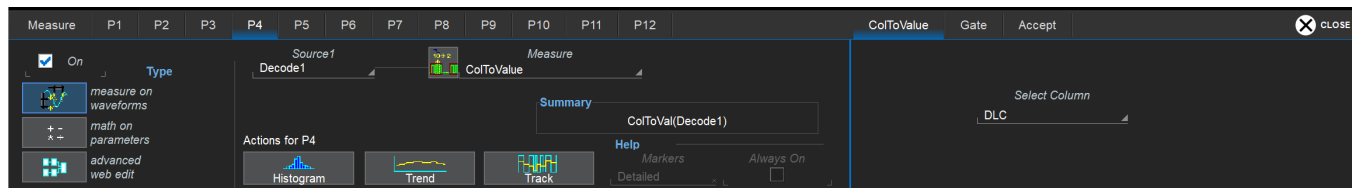
ColToValue acts upon many columns in the table, not just Index or Data. There is no special setup other than the column selection. It is therefore very useful for decodings of continuous data streams that are not "packetized" into separately indexed table rows, or when you want to process values from columns other than Data.

To use this measurement successfully, 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.

In the CAN FD example below, parameters P4 and P5 are both configured with the ColToValue measurement using Source Decode1. P4 is additionally histogrammed (F2), while P5 is tracked (F1).



After beginning to configure the parameter on the Measure/Graph Setup dialog, the **Apply & Configure** button opens the Measure setup ColToValue subdialog, where you **Select (the) Column** whose values are extracted. Parameter P4 is configured to pass the DLC column data, whereas P5 (not shown) passes the BRS column data. This is the data set that appears in Measure table P4 and P5 and is represented by the F2 and F1 functions that graph the parameters.



Eye Diagrams

Not supported on WaveSurfer and HDO4000 series oscilloscopes.

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.

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

The eye can be generated from all transitions in the acquisition or from a "filtered" set of transitions by using the Apply to Zoom feature. With Apply to Zoom, only transitions related to the zoomed (highlighted) table rows are included in the eye diagram.



Note: Serial decode eye diagrams show the decoded signal as it has been configured for the 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.

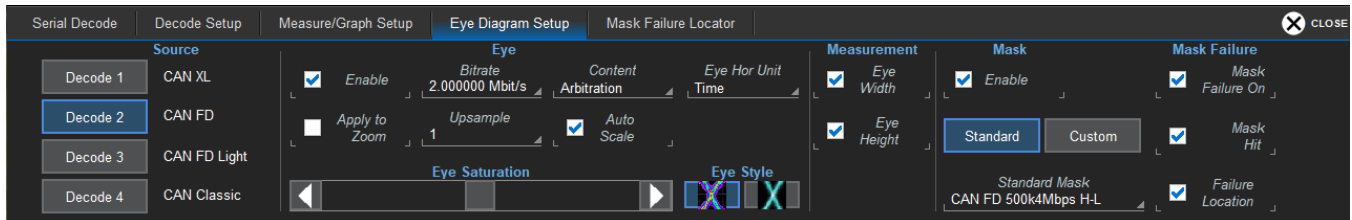
Our recommended approach to using the eye diagram is to:

- Make single acquisitions with decoder and eye diagram enabled to test both are working correctly.
- Make a normal acquisition with Mask Testing and Stop On Failure enabled (if masks have been defined for the protocol), or with a Pass/Fail test set on one of the eye parameters.

Eye Diagram Setup



Note: Examples shown are taken from CAN decoders, but the functionality works the same for all protocols.



Create Eye Diagram

1. Open the **Eye Diagram Setup** dialog and select the **Decode** for which to create an eye diagram.
2. Under Eye, check **Enable** to display the eye diagram.
3. Enter the **Bitrate** of the section of the decoding to be diagrammed. This should match the Content selection in protocols where there is an option.


- Select the type of decoded **Content** (e.g., packet type) from which to draw the eye diagram. The eye will be created solely from transitions of that type, without having to manually adjust the Bitrate. The Content choices will vary by protocol.



Caution: Conflicts can occur if the Content type does not match the table selection when using Apply to Zoom. Zoom only single rows and change Content to match the row zoomed.

- Choose an **Eye Hor(izontal) Unit** of either Time or UI. The eye diagram graticule is redrawn accordingly.
- To diagram only the zoomed row(s) of the decode result table, check **Apply to Zoom**. Eye measurements will also reflect only this zoomed section.
- 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 that can occur when the bit rate is a submultiple of the sampling rate, or the acquisition does not sample a sufficiently large number of UIs.
- To rescale the eye diagram to fit the entire grid, check **Auto Scale**.
- Choose an **Eye Style** of either 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 (lowest) to red (highest).



Analog persistence  uses relative intensities of the same color.

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

Eye Measurements

You can optionally choose to display the **Eye Height** and/or **Eye Width** measurement parameters. The read out will appear in the first open slots of the Measure table, with markers drawn over the eye diagram.

Eye Mask Test

A mask test of the eye diagram is a quick way to verify basic signal integrity. Many standard eye masks have been included for the protocols where masks have been defined. You can also upload custom masks.

- Set up and display the eye, then under Mask, check **Enable** to turn on eye mask testing.
- Select to use either a **Standard** or **Custom** mask, then **Browse** to and select your **Mask File**.
- Optionally, check:
 - Mask Failure On** to mark the parts of the eye diagram that "hit" the mask. Mask violations appear as red circles where the eye diagram intersects the mask.
 - Mask Hits** to display the total number of mask violations on the Measure table.
- To delve deeper into mask hits, check **Failure Location** and use the [Mask Failure Locator](#) feature.

Appendix A: Automating the Decoder

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



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

Configuring the Decoder

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

```
app.SerialDecode.Decoden
```

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.Decoden.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.

CAN FD	Time	Format	ID	IDE	DF	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.Decode1.out.Result.ColumnState contains a pipe-delimited list of all the table columns and their current state (visible=on, hidden=off). For example:

```
app.SerialDecode.Decode1.out.Result.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.

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
support@teledynelecroy.com



700 Chestnut Ridge Road
Chestnut Ridge, NY 10977
USA

teledynelecroy.com