PicoScope 7 Software

User's Guide

The PicoScope 7 User's Guide is still in development, but we've released this early version to help you get started with the software. We're continuing to work on the full guide and welcome your feedback to help improve it. Thank you for sharing your experiences as you use PicoScope 7.

Contents

1. Introduction

Welcome to PicoScope 7, the PC oscilloscope software from Pico Technology. This user's guide is designed to help you use both versions of the software: PicoScope 7 T&M, for test & measurement users, and PicoScope 7 Automotive, for vehicle diagnostics users.

With an oscilloscope device from Pico Technology, PicoScope 7 turns your PC into a powerful PC oscilloscope and spectrum analyzer, with all the features and performance of a benchtop oscilloscope.

Benefits include fast capture rates, fast data processing, clear graphics and text, touchscreen support, regular free-of-charge updates and the ability to easily save, print and share your data.

PicoScope 7 supports all real-time USB PicoScopes and runs on Windows, macOS and Linux operating systems. To find software and documentation for other Pico products please refer to ***picotech.com*** or ***picoauto.com***

For help getting started, see ***Using PicoScope for the first time*** and the ***PicoScope and oscilloscope primer***. Finally, for step-by-step tutorials, see the ***How to...*** section.

PicoScope 7 software has a wide range of features, including:

- **Advanced UI:**
  - Touchscreen friendly: optimized for use on devices with touch controls.
  - Full-screen mode: use the whole screen for waveform display.
  - Zoom and pan: easily zoom in and pan around waveforms.
  - Cursors and markers: measure specific points on waveforms.
  - Annotations and notes: add notes and labels to waveforms.
  - Customizable user interface: allows users to personalize their workspace.
  - Free software updates: continuously receive new features and improvements.
  - Probe mismatch detection: simplifies setup by alerting users to mismatched probes.
  - Printing and exporting: print waveforms or export them in various formats.
  - Multi-language support: available in 28 languages.
  - Cross-platform compatibility: available on Windows, macOS and Linux.
- **Triggering:**
  - Rise/Fall time: capture and analyze signals based on their transition times
  - Runt: detects a pulse that crosses the first threshold but not the second.
  - Pulse Width: enables you to trigger exclusively on pulses with a specified range of widths.
- **Measurements:**
  - Measurements Logging: log automatic oscilloscope measurements to a .CSV file for long-term analysis.
  - Measurements Limits & Actions: automatically test measurement results against user-defined limits and trigger actions on violations.
  - Improved measurement functions: comprehensive set of inbuilt measurements with an easy-to-read display.
- **Automotive:**
  - Guided tests: updated and improved guided tests for setting up the scope and explaining what to check.
  - Waveform library: comprehensive, easy to use, and faster than ever before.
  - Reference Waveforms: enables holding a waveform on screen for comparison.
- **Waveform analysis:**
  - Serial protocol decoders: decode over 40 serial protocols.
  - Math channels: perform mathematical operations on waveforms.
  - Spectrum analysis: analyze the frequency content of signals.
  - XY view: display one channel against another.
  - Persistence display: overlay multiple waveforms to visualize changes over time.

2. Legal and system information

2.1 Legal Statement

Grant of license. The material contained in this release is licensed, not sold. Pico Technology Limited ('Pico') grants a license to the person who installs this software, subject to the conditions listed below.

Access. The licensee agrees to allow access to this software only to persons who have been informed of and agree to abide by these conditions.

Usage. The software in this release is for use only with Pico products or with data collected using Pico products.

Copyright. Pico claims the copyright of, and retains the rights to, all material (software, documents etc.) contained in this release.

Liability. Pico and its agents shall not be liable for any loss or damage, howsoever caused, related to the use of Pico Technology equipment or software, unless excluded by statute.

Fitness for purpose. No two applications are the same, so Pico cannot guarantee that its equipment or software is suitable for a given application. It is therefore the user's responsibility to make sure the product is suitable for the user's application.

Mission-critical applications. Because the software runs on a computer that may be running other software products, and may be subject to interference from these other products, this license specifically excludes usage in 'mission-critical' applications, for example life-support systems.

Viruses. This software was continuously monitored for viruses during production. However, the user is responsible for virus checking the software once it is installed.

Support. No software is ever error-free, but if you are dissatisfied with the performance of this software, please contact our technical support staff.

2.2 Updates

We provide software updates, free of charge, from our web site at ***www.picotech.com***, ***www.picoauto.com*** and through the software itself.

2.3 Trademarks

Pico Technology, PicoScope, PicoLog, FlexRes and ConnectDetect are internationally registered trademarks or trademarks of Pico Technology Limited, registered in the United Kingdom and other countries.

PicoScope and Pico Technology are registered in the U.S. Patent and Trademark Office.

DeepMeasure is a trademark of Pico Technology Limited.

2.4 System requirements

To ensure PicoScope runs correctly, your computer must meet at least the minimum requirements for a supported operating system listed below. Performance improves with a more powerful PC, especially one with a multi-core processor.

Item	Specification
Operating system (all 64-bit only)	Microsoft Windows 10 or 11, 64-bit only Linux: Ubuntu or openSUSE, 64-bit only macOS, 64-bit only See ***picotech.com/downloads*** or ***picoauto.com/downloads*** for full details.
Processor	As required by the operating system
Memory	
Free disk space	
Interfaces	USB 3.0 or USB 2.0 port(s)

3. Using PicoScope for the first time

We have designed PicoScope to be as easy as possible to use, even for newcomers to oscilloscopes. Once you have followed the introductory steps listed below, you will soon be on your way to becoming a PicoScope expert.

1. Download the latest PicoScope software from *picotech.com/downloads* or *picoauto.com/downloads*
2. Install the newly-downloaded software.
3. Connect your PicoScope to your PC or Laptop
4. Open your newly installed software
5. PicoScope 7 software will detect your PicoScope and you're ready to start taking measurements

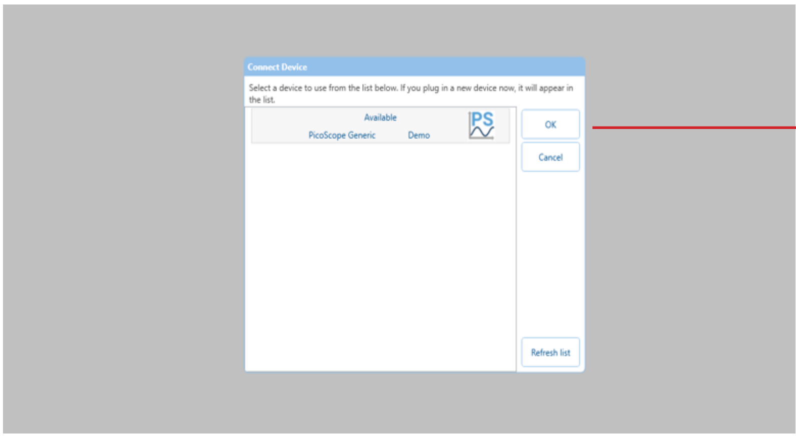
If you haven't already received your PicoScope and you want to try the software, don't worry you can still download, install and run the software in demo mode, see section 3.1 below

Problems?

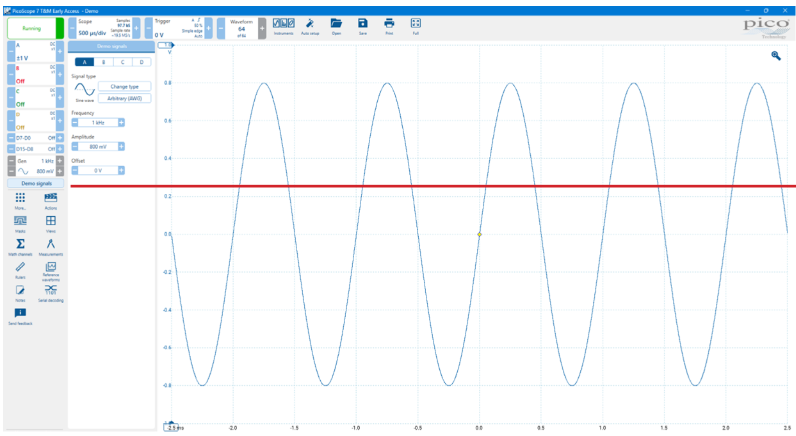
Help is at hand! Our technical support staff are always ready to answer your telephone call during office hours (09:00 to 17:00, UK time and 09:00 to 17:00 US Central time, Mon-Fri: see the contact details on the last page of this document). At other times, you can leave a message on our support forum or send us an email.

3.1 Demo mode

If you start PicoScope software without a device connected, you will see the following screens:



Click Ok



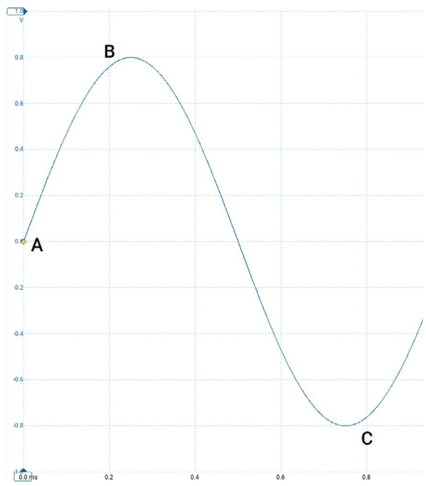
Once the demo version is running click the **Demo signals** button that appears on the left hand side. This will give you access to a range of simulated signals that are useful for evaluation of PicoScope 7 functions and for education purposes.

4. PicoScope and oscilloscope primer

This chapter covers the essential concepts you should understand before using PicoScope 7. If you're already familiar with PC oscilloscopes, you can skip this section.

4.1 Oscilloscope basics

An oscilloscope is a measuring tool that graphically displays how a signal changes over time. By connecting probes or sensors to its input channels, it can capture signals such as voltage, current, pressure and vibration. For example, the image below shows a typical waveform when a varying voltage is applied to one of the inputs.



Oscilloscope displays are always read from left to right. The voltage-time characteristic of the signal, or waveform, is drawn as a line called the trace. In this example, the trace is blue and begins at point **A**. If you look to the left of this point, you will see the number **0.0** on the vertical axis, which tells you that the voltage is 0.0 V (volts). If you look below point **A**, you will see another number **0.0**, this time on the horizontal axis, which tells you that the time is 0.0 ms (milliseconds) at this point.

At point **B**, 0.25 ms later, the voltage has risen to a positive peak of 0.8 V. At point **C**, 0.75 ms after the start, the voltage has dropped to a negative peak of –0.8 V. After 1 ms, the voltage has risen back to 0.0 V and a new cycle is about to begin. This type of signal is called a sine wave, and is one of a limitless range of signal types that you will encounter.

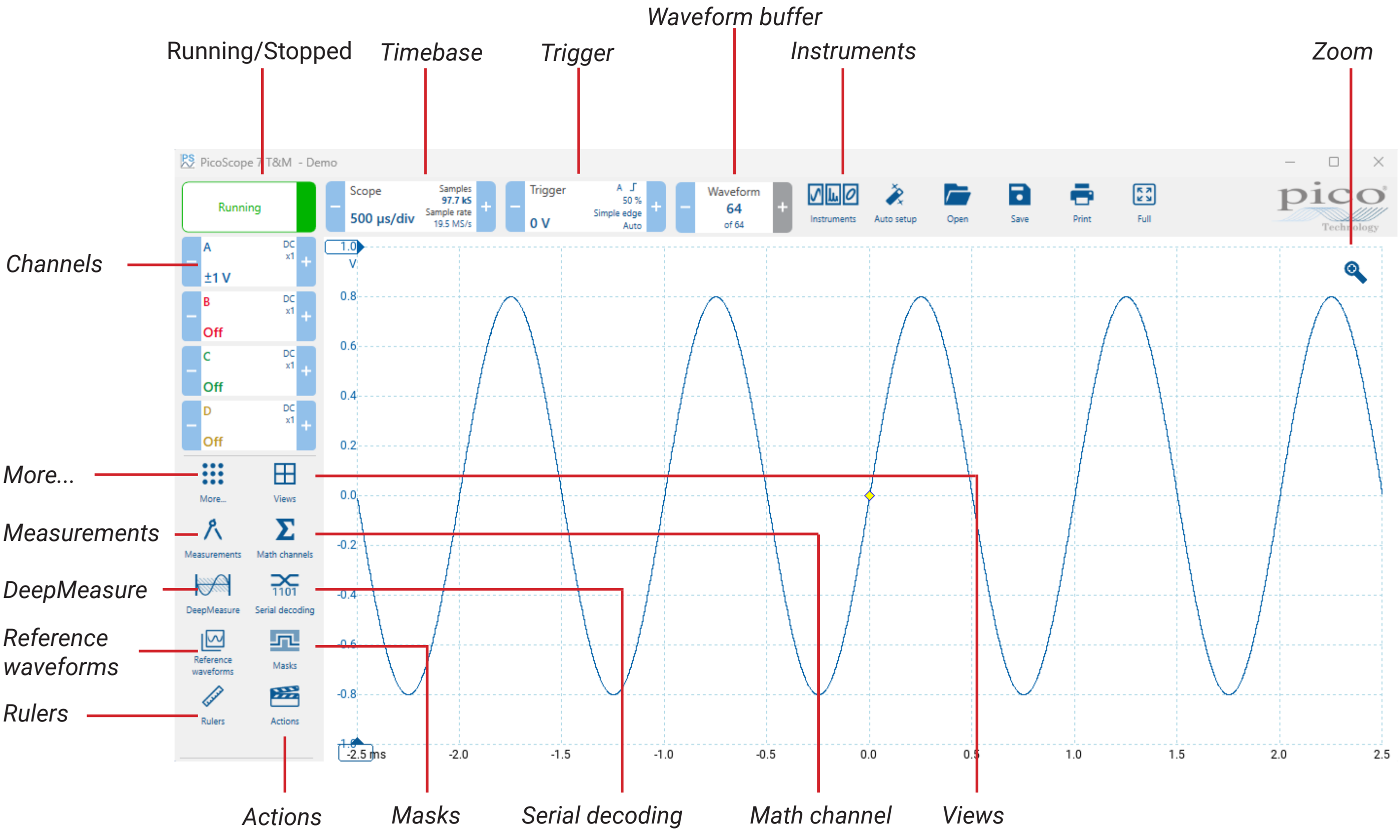
Most oscilloscopes allow you to adjust the vertical and horizontal scales of the display. The vertical scale is called the **input range**, and can be measured in a number of different units: the example above uses volts. The horizontal scale is called the **Timebase** and is measured in units of time - in this example, milliseconds.

4.2 PC oscilloscope basics

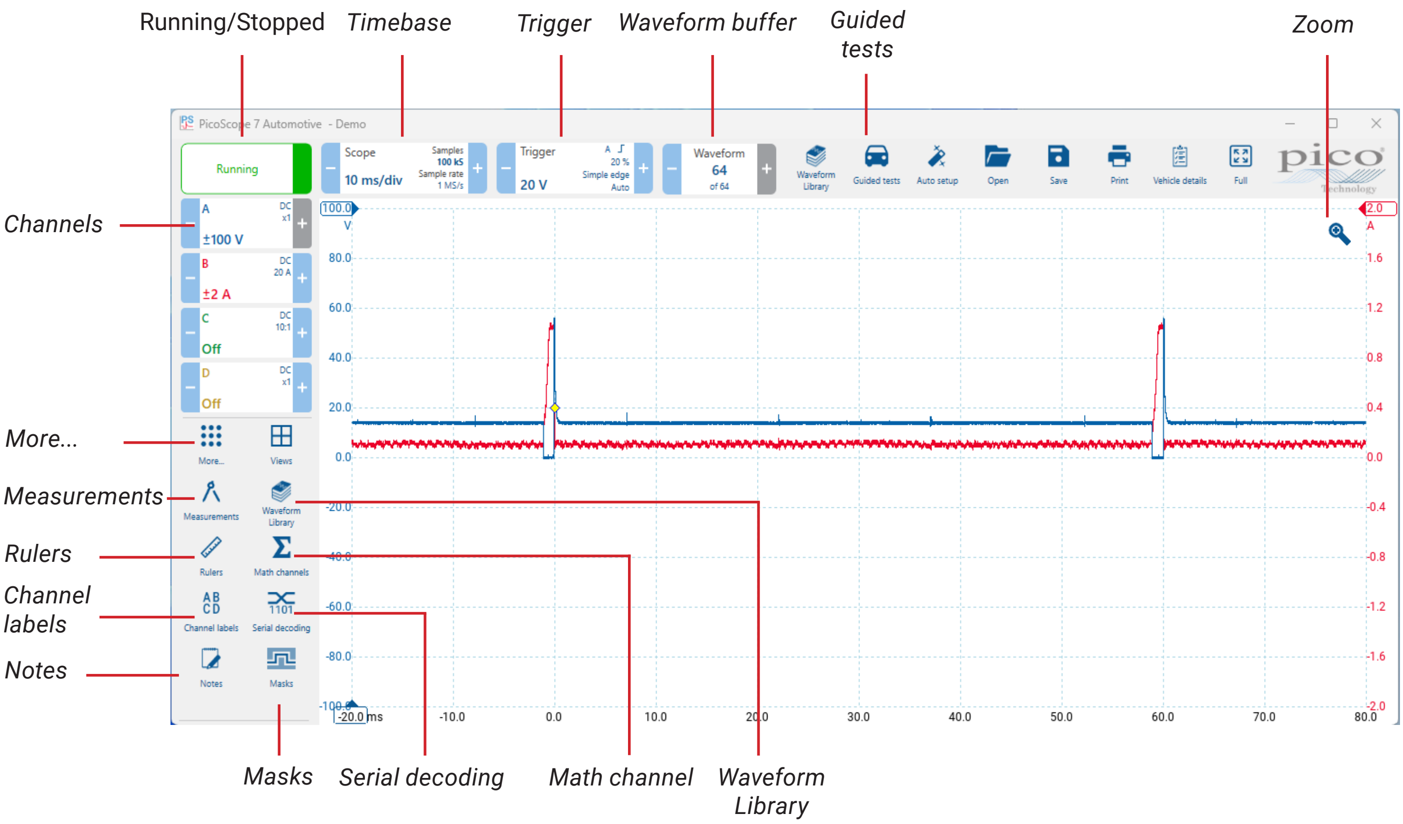
A PC oscilloscope combines a hardware data acquisition device with oscilloscope software running on a computer. Unlike traditional standalone oscilloscopes, which offered limited functionality and costly storage, PC-based oscilloscopes take advantage of modern digital technology to deliver powerful features at a lower cost. By pairing Pico Technology’s PicoScope hardware with your PC, you gain a flexible, high-performance, measurement system that’s easy to use and upgrade.

4.3 PicoScope basic overview

PicoScope can produce a simple display such as the example in the Oscilloscope basics topic, but it also has many advanced features. The screenshot below shows the PicoScope window.



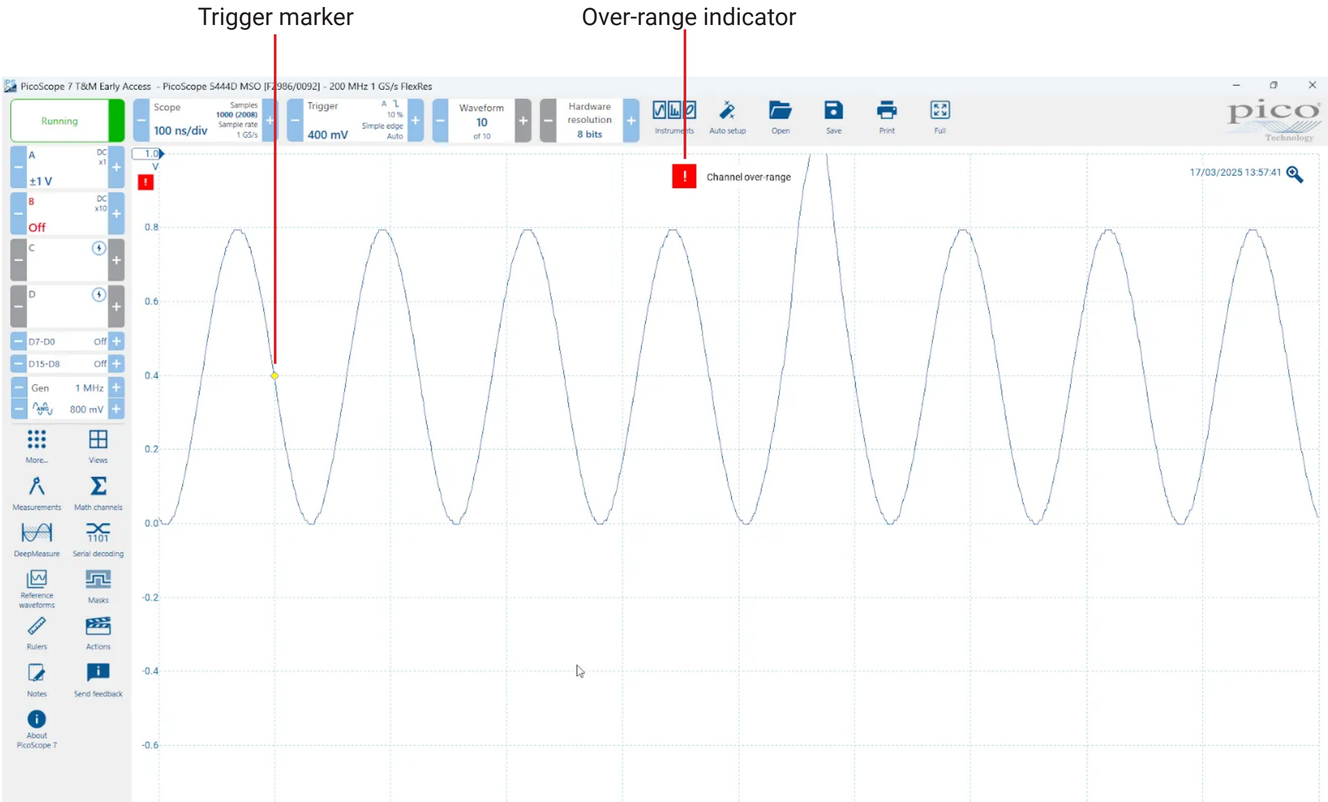
PicoScope 7 T&M



PicoScope 7 Automotive

4.3.1 Scope view

A scope view shows the data captured from the scope as a graph of signal amplitude against time. PicoScope opens with a single view, but you can add more views by using the views option panel. Similar to the screen of a conventional oscilloscope, a scope view shows you one or more waveforms with a common horizontal time axis, with signal level shown on one or more vertical axes. Each view can display up to 8 waveforms, including math channels and reference waveforms. To see more waveforms at once, just open another view.



4.3.2 Over-range indicator

If the signal voltage exceeds the measurement input range, PicoScope displays the red over-range indicator in the top center of the display, with the message "Channel over-range". A smaller version appears next to the vertical axis of the affected channel. The waveform will be clipped: no data outside the measurement input range will be shown. Increase the input range of the affected channel until the indicator disappears.

Scopes with differential inputs (PicoScope 3425 and 4444 only)

If the common-mode voltage of the differential input exceeds the oscilloscope's common-mode input range, the yellow common-mode over-range indicator appears in the top center of the PicoScope display, with the message "Common-mode over-range". Again, a smaller version appears next to the vertical axis of the affected channel. Exceeding the common-mode input range of the scope causes inaccurate measurements and can lead to severe signal distortion.

Scopes with floating inputs (PicoScope 4225, 4225A, 4425 and 4425A only)

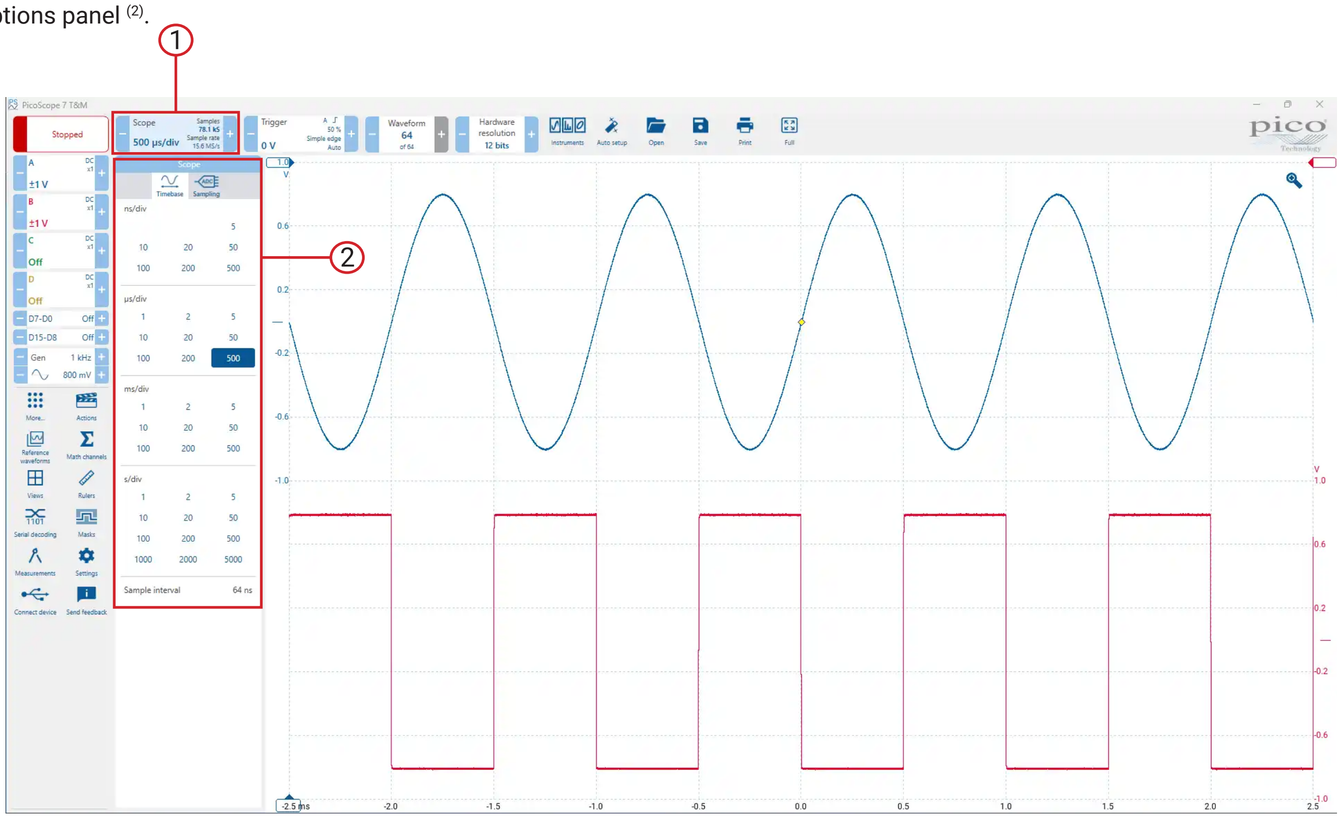
If the voltage between the BNC shell and the oscilloscope chassis exceeds 30 V, the yellow warning indicator appears in the top center of the PicoScope display. A smaller version appears next to the vertical axis of the affected channel, and the LED next to that channel's input on the scope device will turn solid red. When this warning appears, check the orientation of the test connections against the wiring diagrams. In some cases, you may need to provide a 0 V reference to the scope, using the M4 screw terminal on the rear of the unit.

When an over-range or common-mode over-range condition is present, displayed waveforms and measurements may be incorrect and the oscilloscope is unable to indicate the true input voltage which may be hazardous. Select a larger voltage range to achieve a within-range measurement.

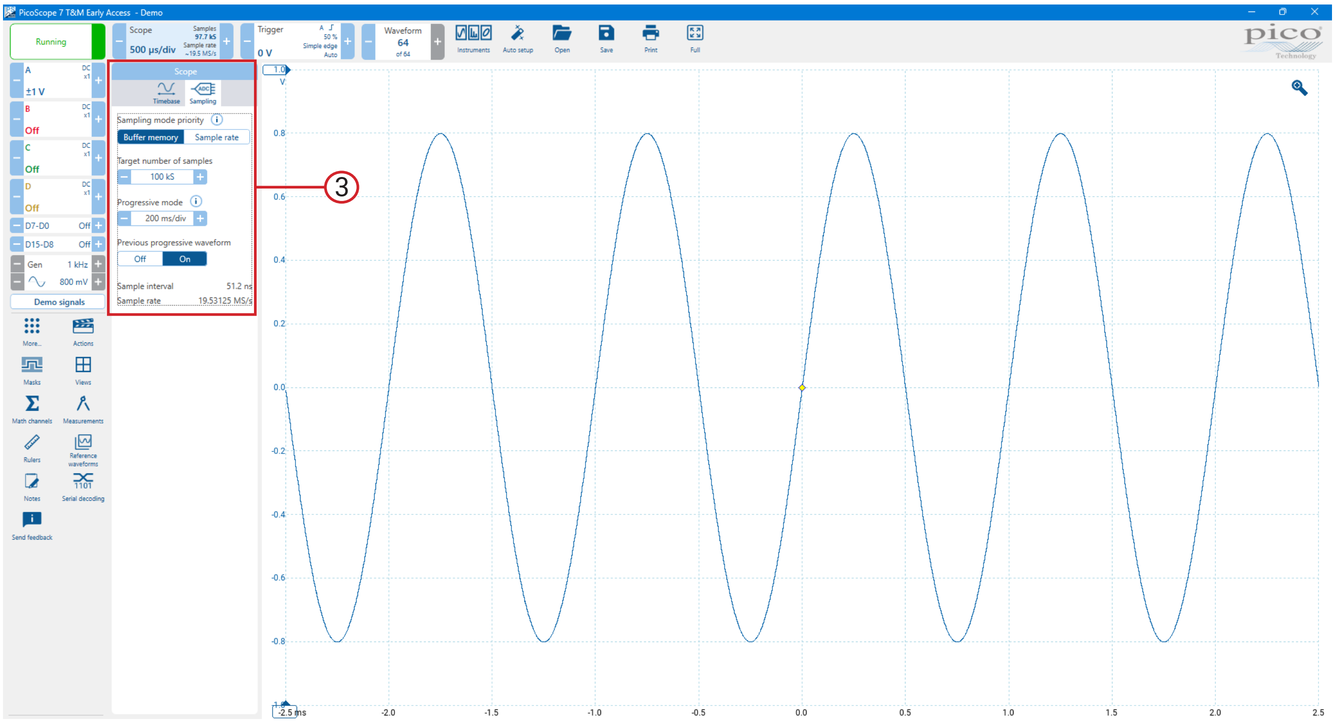
4.3.3 Timebase and Sampling controls

The timebase sets the time interval across the horizontal axis. By default it is scaled in time per division. As there are ten horizontal divisions, the total collection time is 10 times the time per division.

To change the Timebase settings, click the Scope lozenge (1) on the top row, then select the appropriate time from the Timebase options panel (2).

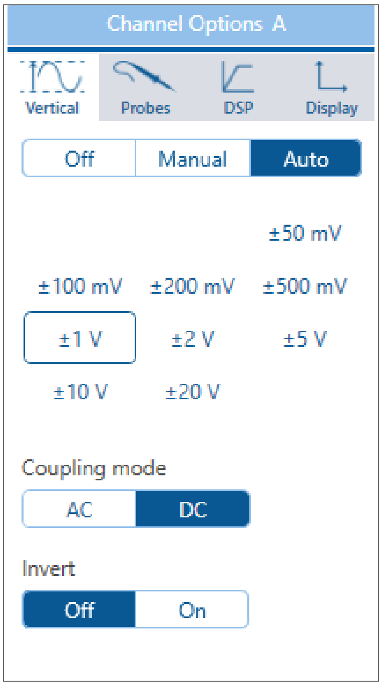


To change the Sampling settings, click Sampling (3) to adjust the Sampling mode priority, Target number of samples, Progressive mode and Previous progressive waveform from the Sampling options panel.



4.3.4 Channel controls (basic)

When clicking each input channel in the side bar, an options panel opens allowing you to adjust the settings for that input channel.

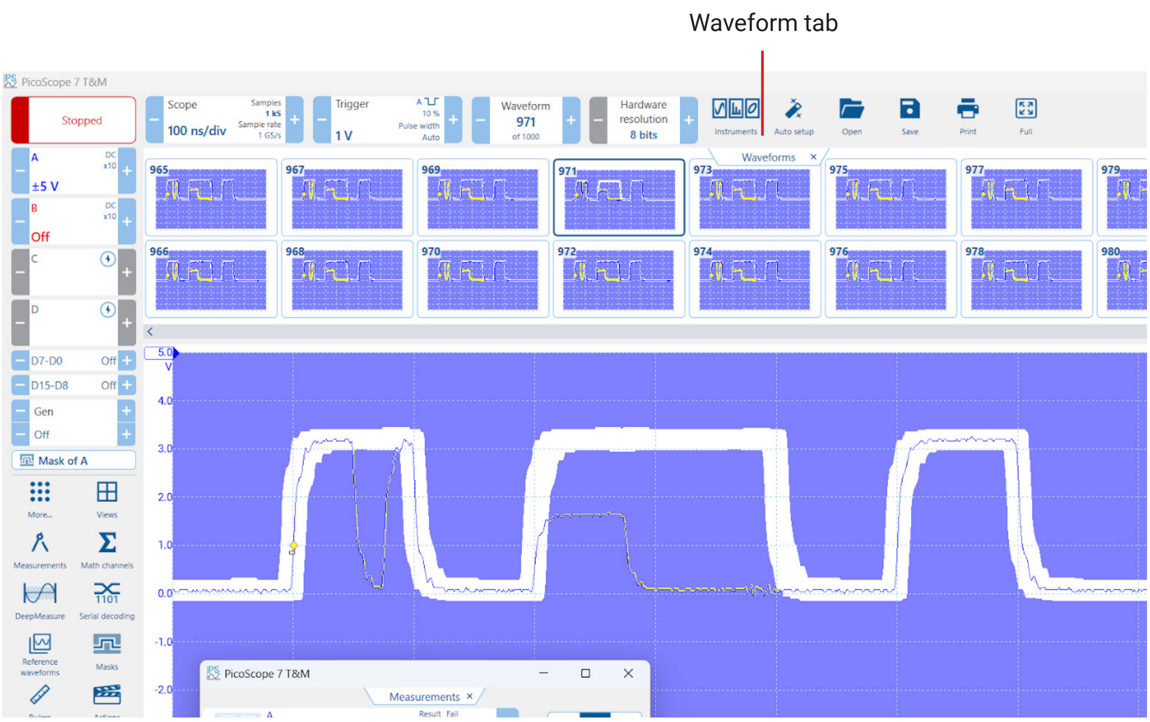


To get started, click "Auto" to turn the channel on and have PicoScope manage the input range settings according to the signal detected, or click a voltage range to manually select the input span.

Further options are available in the tabs at the top of the options panel. See section *5.1.1* for further information on the channel controls.

4.3.5 Waveform buffer

The PicoScope Waveform buffer can hold up to 40 000 waveforms, subject to the amount of available memory in the oscilloscope. The Waveform tab helps you to scroll through the buffer quickly to find the waveform you want.



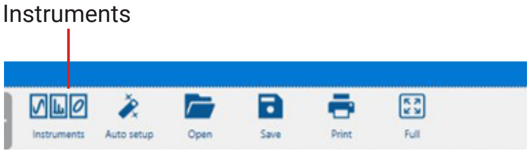
Click on any one of the visible waveforms to bring it to the front of the overview for closer inspection, or if there are multiple waveforms, then scroll using the on-screen controls.

For more information, see *www.picotech.com/library/knowledge-bases/oscilloscopes/waveform-buffer*

Note: additional buttons may appear in the main PicoScope window depending on the oscilloscope that is connected, and on the settings applied to the PicoScope software.

4.3.6 Instruments

PicoScope can operate in four modes: Scope mode, Spectrum mode, XY mode and Persistence mode. The modes are selected by pressing the Instruments button.



In Scope mode, PicoScope displays a main scope view, optimizes its settings for use as a PC Oscilloscope, and allows you to directly set the Timebase. You can still display one or more secondary Scope, Spectrum or XY views.

In Spectrum mode, PicoScope displays a main spectrum view, optimizes its settings for spectrum analysis, and allows you to directly set the frequency range in a similar way to a dedicated spectrum analyzer. You can still display one or more secondary Scope, Spectrum or XY views.

In XY mode, PicoScope XY plots one channel against another on the screen. This plot could be the I–V curve of a component such as a capacitor, inductor or diode, or a Lissajous figure showing the phase difference between two or more periodic waveforms.

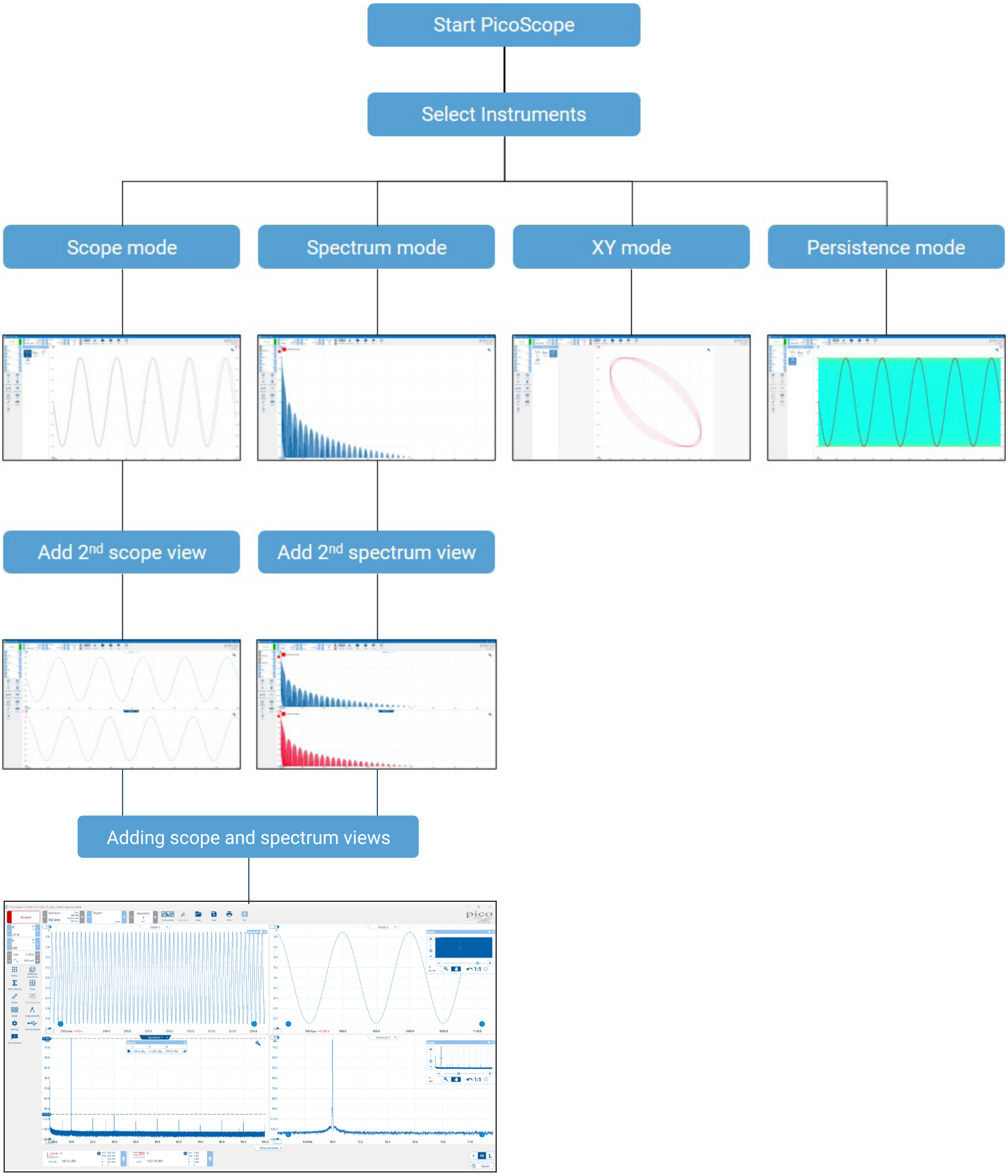
In Persistence mode, PicoScope displays a single, modified scope view in which many captures of the same waveform are overlaid, with their color or intensity proportional to how often each part of the waveform occurs, or how recently it last occurred.

Saving a file in PicoScope saves the data or settings for the instrument that is currently in use, including all viewports and settings within that instrument. To save data for multiple instruments, switch to that instrument and save another file.

4.3.7 How do Instruments work with views?

Selecting Instruments tells PicoScope whether you are mainly interested in viewing waveforms or frequency plots. When you select an Instrument, PicoScope sets up the hardware appropriately and then shows you a view that matches the capture mode (a scope view if you selected Scope mode or XY mode or a spectrum view if you selected Spectrum mode). The rest of this section does not apply in Persistence mode, which allows only a single view.

Once PicoScope has shown you the first view, you can, if you wish, add more scope or spectrum views, regardless of the mode you are in. You can add and remove as many extra views as you wish, as long as one view remains that matches the mode, as illustrated in the diagram below. For more information on views, ***click here***



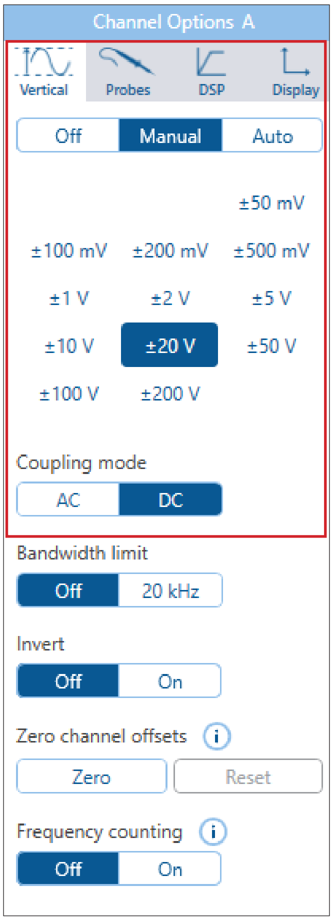
The example above is showing how you might select Instruments and open additional views in PicoScope

5. Waveform capture

5.1 Analog channels setup

5.1.1 Channel controls

The **Channel controls** allow you to adjust the settings for each input channel. Different scope devices may have different numbers of channels.

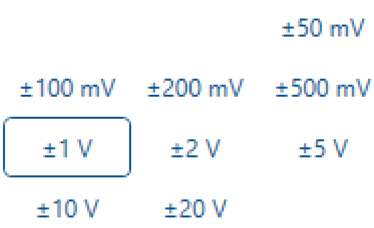


Each channel has its own set of buttons:

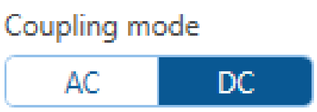
Channel Options tabs. Opens the **Channel Options** with options for **Vertical**, **Probes, DSP** and **Display**.

Off/Manual/Auto control:

- **Off -** Turns off the channel
- **Manual -** Allows the user to manually adjust the vertical scale
- **Auto -** PicoScope will continually adjust the vertical scale so the height of the waveform fills as much of the view as possible



Input Range control. Sets up the scope device to capture signals over the specified range of values. The list of options depends on the selected scope device and probe. The red overrange indicator appears if the input signal exceeds the selected range.

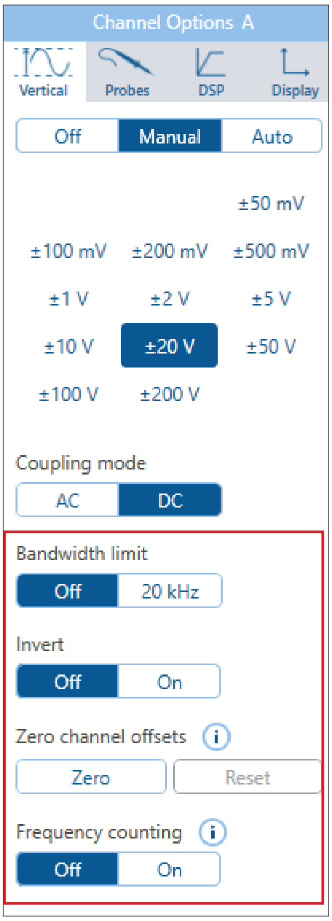


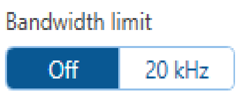
Coupling control. Sets up the input circuitry. Available coupling types vary depending on scope model.

AC coupling: rejects frequencies below about 1 Hz.

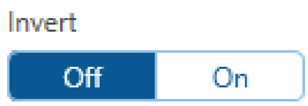
DC coupling: accepts all frequencies from DC to the scope's maximum bandwidth.

50Ω DC: matched impedance coupling for high frequency signals.



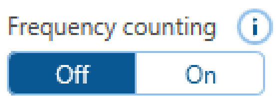


Bandwidth limit - One or more fixed hardware bandwidth limits, depending on your scope model. This means any parts of your measured signal that have a frequency above the limit will be reduced in amplitude. And the higher their frequency above the bandwidth limit, the greater the reduction. Software low-pass filtering is also available in the DSP tab providing adjustable cut-off frequency.



Invert - Switching this On will turn your signal upside down.

Zero channel offsets:
Zero - runs a self-calibration routine to digitally remove any offset from the input channel.
Reset - removes any applied zero-offset corrections.



Frequency counting - Hardware frequency counting is available on Automotive scopes only and only one channel at a time. Use the math channels freq() function on other scopes or to plot the frequency of multiple channels.

5.1.2 Resolution enhancement

Resolution enhancement is a technique for increasing the effective vertical resolution of the scope at the expense of high-frequency detail. In some scope operating modes, PicoScope may reduce the number of samples available to maintain display performance.

For this technique to work, the signal must contain a very small amount of Gaussian noise, but for many practical applications this is generally supplied by the scope itself and the noise inherent in normal signals.

The **Resolution enhancement** feature uses a flat moving-average filter. This acts as a lowpass filter with good step response characteristics and a very slow roll-off from the pass-band to the stop-band.

Some side effects will be observed when using **Resolution enhancement**. These are normal and can be counteracted by reducing the amount of enhancement used, increasing the number of samples captured or changing the collection time. Trial and error is usually the best way to find the optimum **Resolution enhancement** for your application. The side effects include:

· Widened and flattened impulses (spikes)

· Vertical edges, such as those of square waves, turned into straight-line slopes

· Inversion of the signal, sometimes making it look as if the **trigger point** is on the wrong edge

· A reduction in signal amplitude, or even a flat line, when there are not enough samples in the waveform

Procedure

· Click **Channel Options button, DSP tab**

· Use the **Resolution enhancement** control to select the effective number of bits, which can be equal to or greater than the vertical resolution of your scope device.

Quantifying Resolution enhancement

The table below shows the number of values, ***n***, used by the moving-average filter for each **resolution enhancement** setting, ***e***. A larger number of values in the filter requires a higher sampling rate to represent a given signal without significant side effects (as detailed above).

Resolution enhancement ***e*** (bits)	Number of values ***n***
0.5	2
1.0	4
1.5	8
2.0	16
2.5	32
3.0	64
3.5	128
4.0	256

Example

Your scope device is a PicoScope 2208B (resolution = 8 bits). You have selected an effective resolution of 9.5 bits.

The **Resolution enhancement** is therefore:

e = 9.5 – 8.0 = 1.5 bits

The table shows that this is achieved using a moving average of:

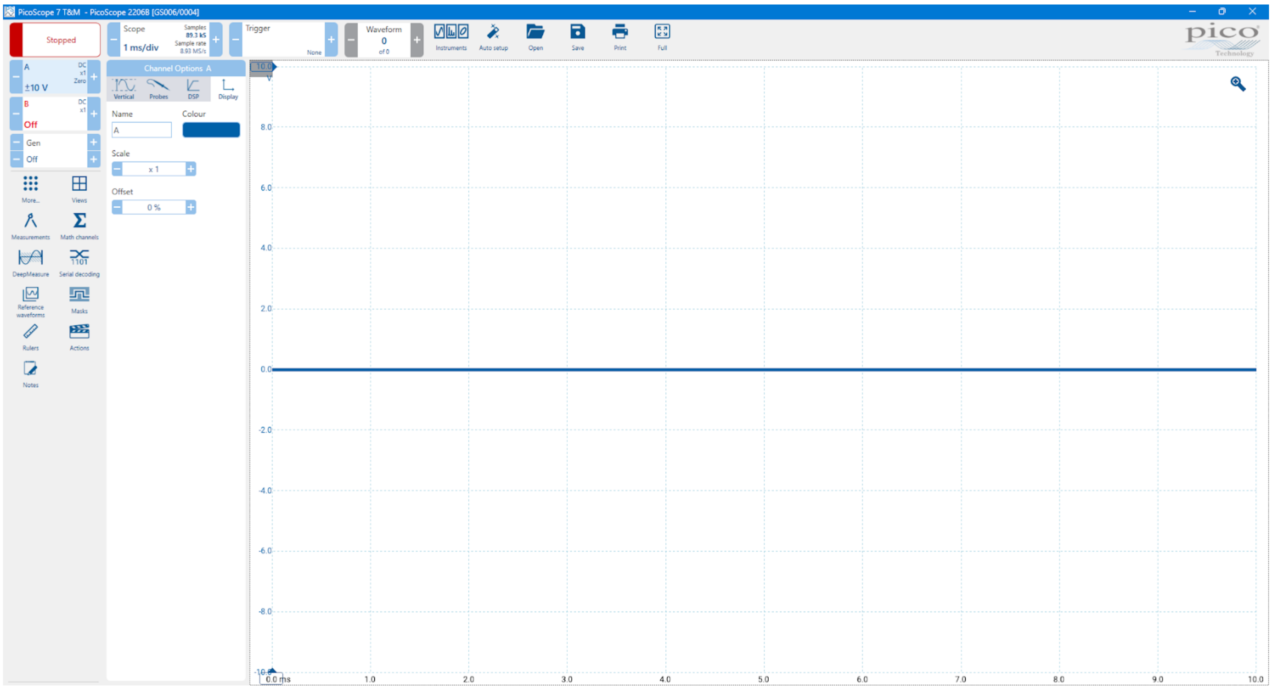
n = 8 samples

This number gives a guide to what sort of filtering effect the **Resolution enhancement** will have on the signal. The best way of seeing the actual lowpass filter effect is to add a **Spectrum view** and look at the shape of the noise floor (try dragging the y-axis upwards to see the noise more clearly).

5.1.3 Axis scaling controls

Location: **Channel button >** Display

The **axis scaling controls** let you change the **scale** and **offset** of each vertical axis individually. If the axis belongs to a **reference waveform**, you can also adjust its delay relative to the live waveforms. You can use the **axis scaling controls** in any capture mode.



Name

Choose a custom name to help identify each channel signal. The name is shown wherever the channel is used, for example in measurements.

Color

Configure a custom trace color for the channel. Channel colors are saved as a user preference, so PicoScope will always use your own selected colors even when sharing files with other users.

Scale.

Increase to magnify the waveform, decrease to reduce it. The vertical axis rescales accordingly so that you can always read the correct voltage from the axis. The scaling button always shows the selected scale. You can also scale the vertical axis by using the mouse wheel or pinch zoom on a touchscreen over the axis itself, which has the same effect as this control.

Offset.

Increase to move the waveform up the display, decrease to move it down. The vertical axis shifts accordingly so that you can always read the correct voltage from the axis. You can also adjust the offset by dragging the vertical axis up or down the display.

Delay control
(Reference waveform only)

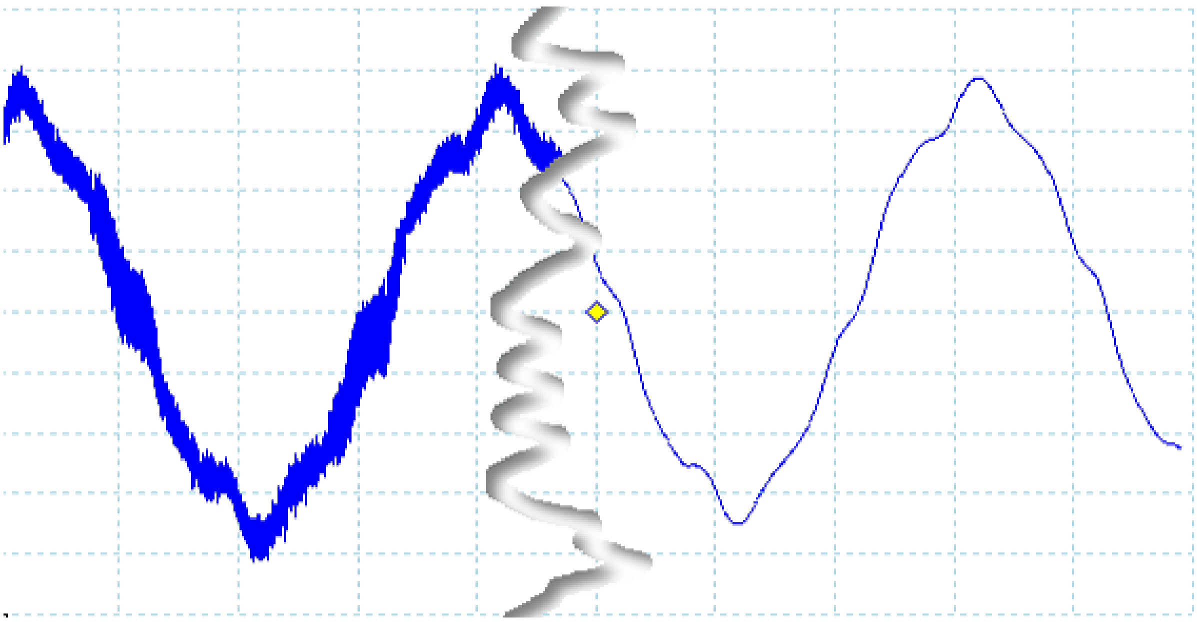
Increase to move the waveform to the left relative to the timing reference point, decrease to move it to the right.

The location of the timing reference point depends on which trigger mode PicoScope is in. If the trigger mode is None then the delay is measured relative to the left-hand edge of the display. In all other trigger modes, the delay is measured relative to the trigger marker.

5.1.4 Lowpass filtering

The lowpass filtering feature can reject high frequencies from any selected input channel. The filtering control is found in the DSP tab of the channel options panel, for the relevant channel. The control determines the cut-off frequency of the filter, which must be below half the sampling rate shown in the timebase control.

Lowpass filtering is useful for rejecting noise. The split screenshot below shows the effect of applying a 1 kHz lowpass filter on a noisy signal. The underlying shape of the signal is preserved but the high-frequency noise is eliminated:



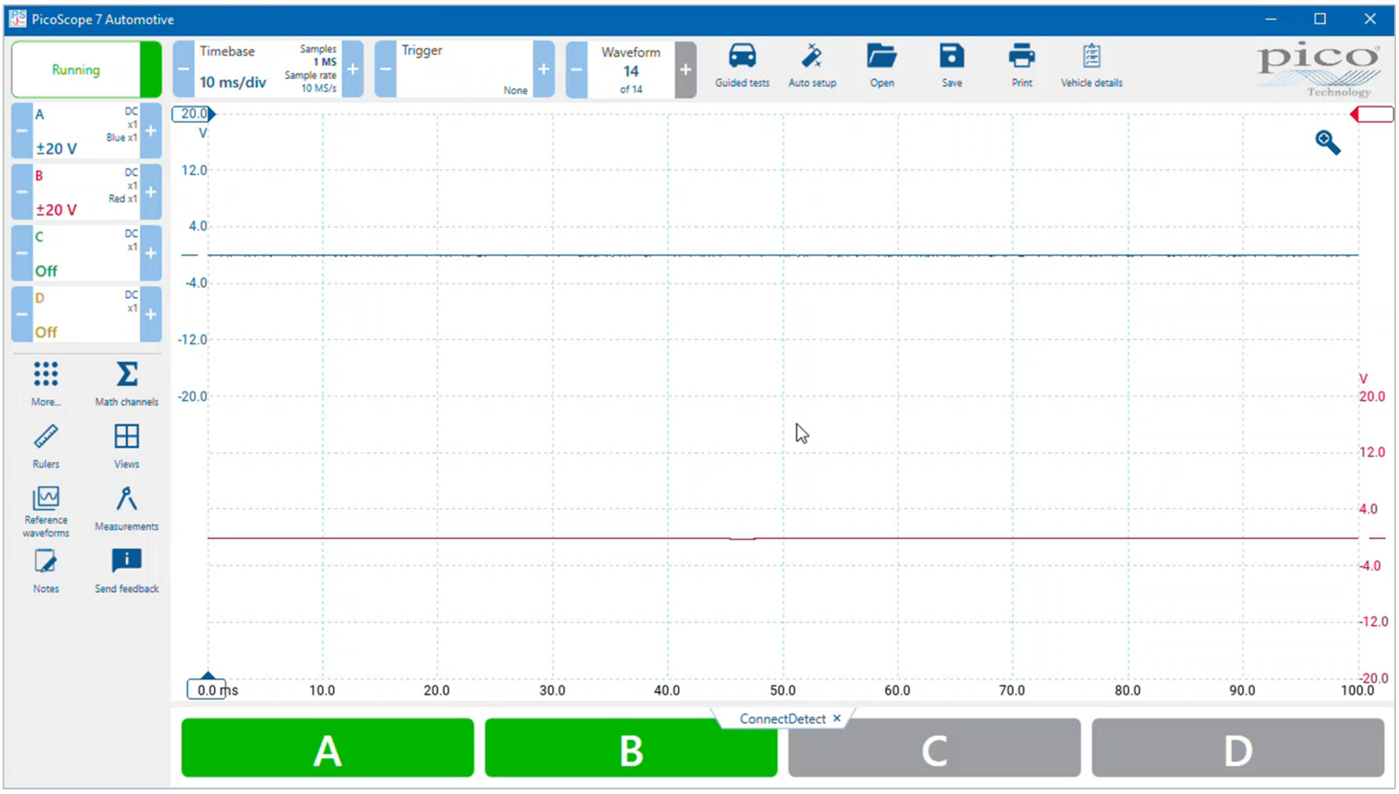
Left: before lowpass filtering. Right: after 1 kHz lowpass filtering.

5.1.5 ConnectDetect®

Location: **More... -> ConnectDetect** (automotive oscilloscopes only)

Purpose: Indicates whether or not a test probe has a good physical connection to the component under test

Applicability: PicoScope 4225, 4225A, 4425 and 4425A Automotive oscilloscopes only, DC coupling mode only.



When ConnectDetect is activated, the on-screen indicator for each channel is either green, to indicate that the test probe is directly connected across a component, or red to indicate that it is not. Channels which are turned off are shown in grey, and are not tested by ConnectDetect.

ConnectDetect

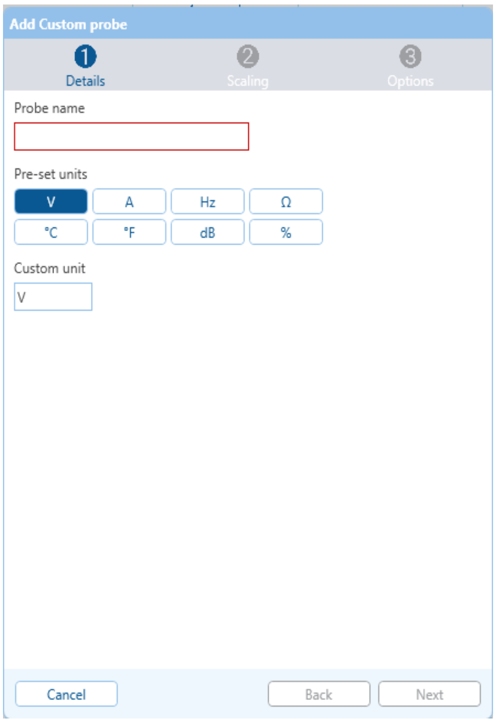
Note: ConnectDetect can only be used with probes that come into direct contact with the component under test. Sensors such as the TA204 coil-on-plug and signal probe, which works using electrical induction from the ignition coils attached to the spark plugs, do not make such a connection, so ConnectDetect will not work with them.

5.1.6 Custom Probes dialog

Location: **Channel button > Probes > Manage Probes > Add** or **Edit**

A probe is any transducer, measuring device or other accessory that you connect to an input channel of your scope device. PicoScope has a built-in library of common probe types, such as the x1 and x10 voltage probes used with most oscilloscopes, but if your probe is not included in this list you can use the Manage Probes dialog to define a new one. Custom probes can have any input range within the capabilities of the oscilloscope, display in any units, and have either linear or nonlinear characteristics.

Custom probe definitions are particularly useful when you wish to display the probe's output in units other than volts, or to apply linear or nonlinear corrections to the data.

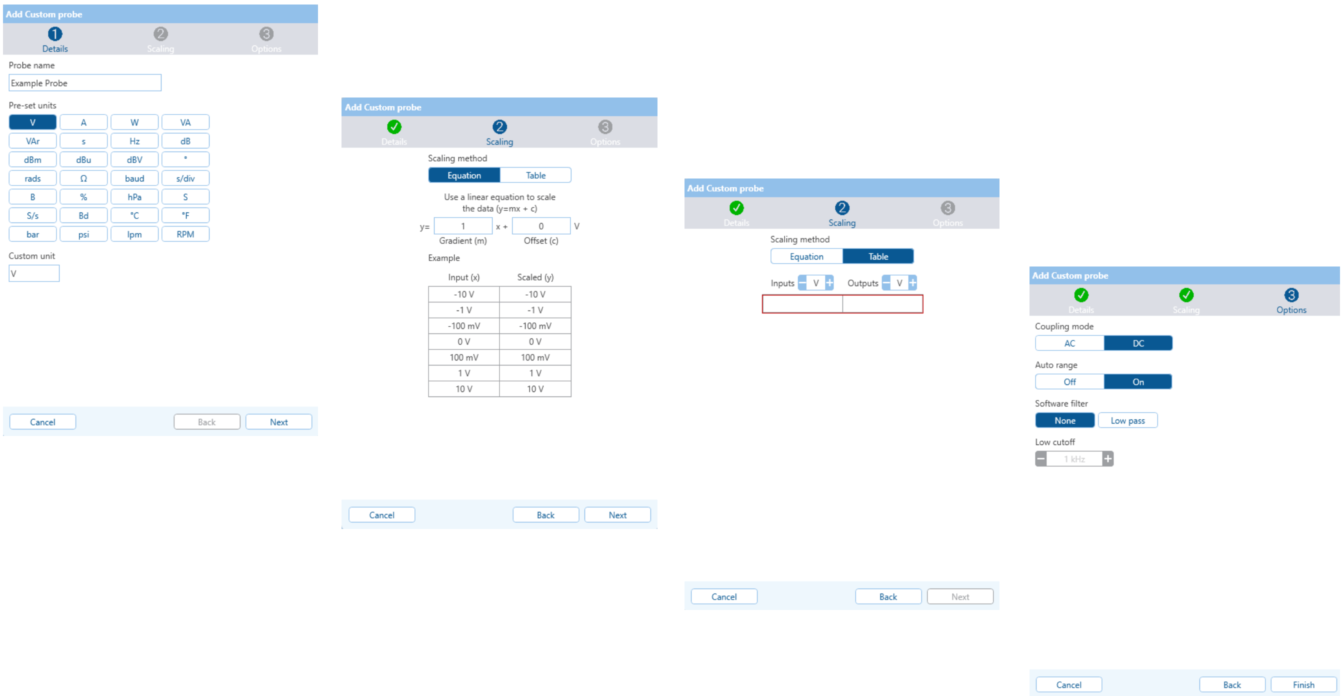


The selection of built-in probes shown may vary depending on the version of the PicoScope software that you are using.

All the probes that PicoScope knows about are listed in sections according to the type of probe. The probe list is preserved between sessions, so that PicoScope will never forget your custom probes unless you delete them.

Adding a probe to your library

Click **Add** in the manage probes section to add a custom probe, now go through the custom probe wizard and create a name, add one of the Pre-set units, select the scaling method, Coupling mode, Auto range and Software filter.

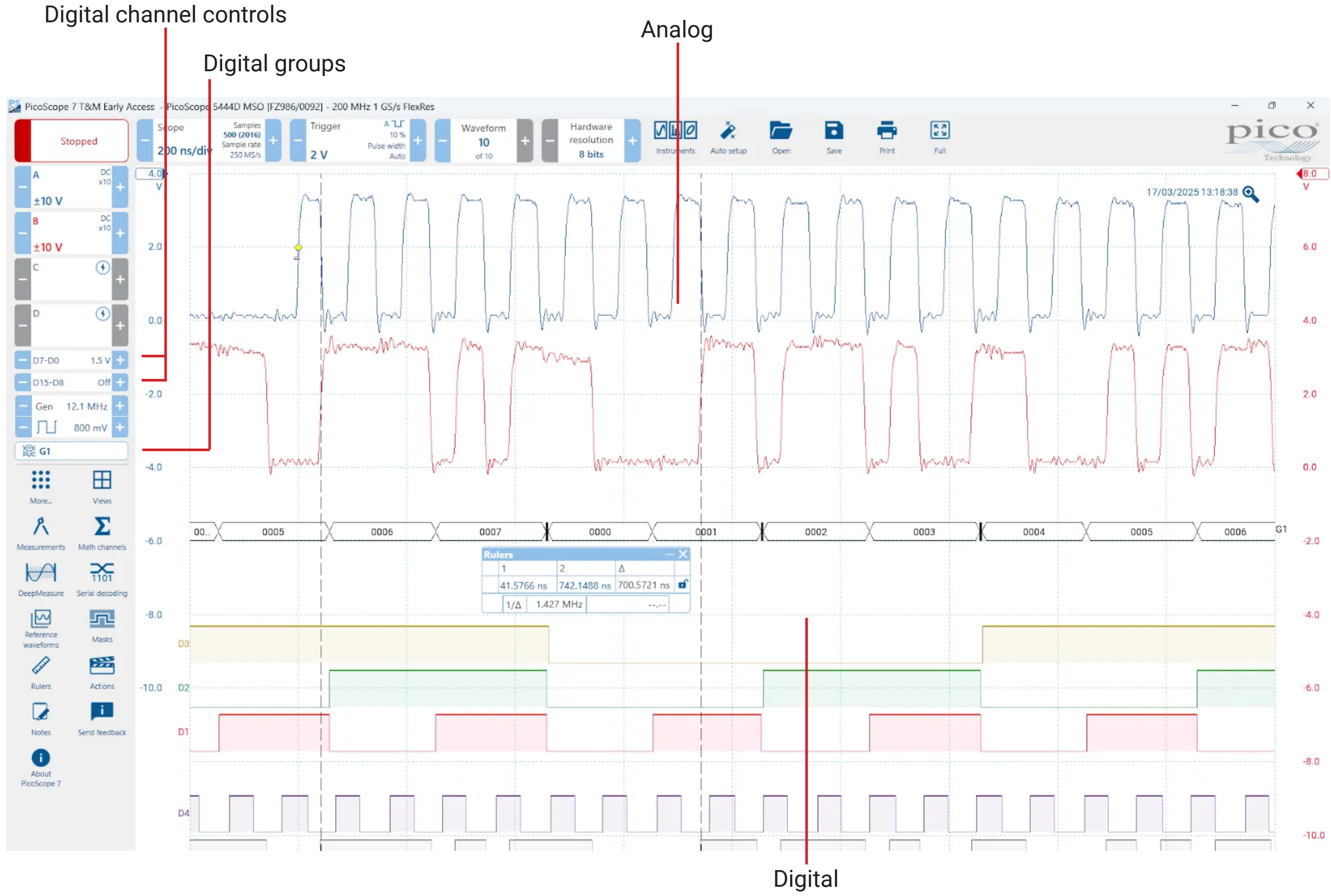


5.2 Digital channels setup

5.2.1 Digital channels

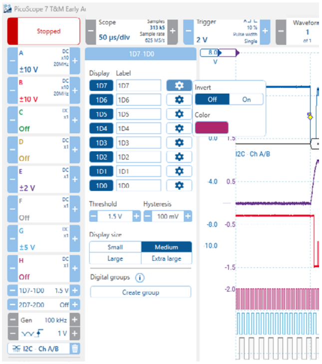
Applicability: mixed-signal oscilloscopes (MSOs) only

The scope view shows mixed analog and digital data on the same horizontal scale.



5.2.2 Digital channel options panel

Location: Scope view

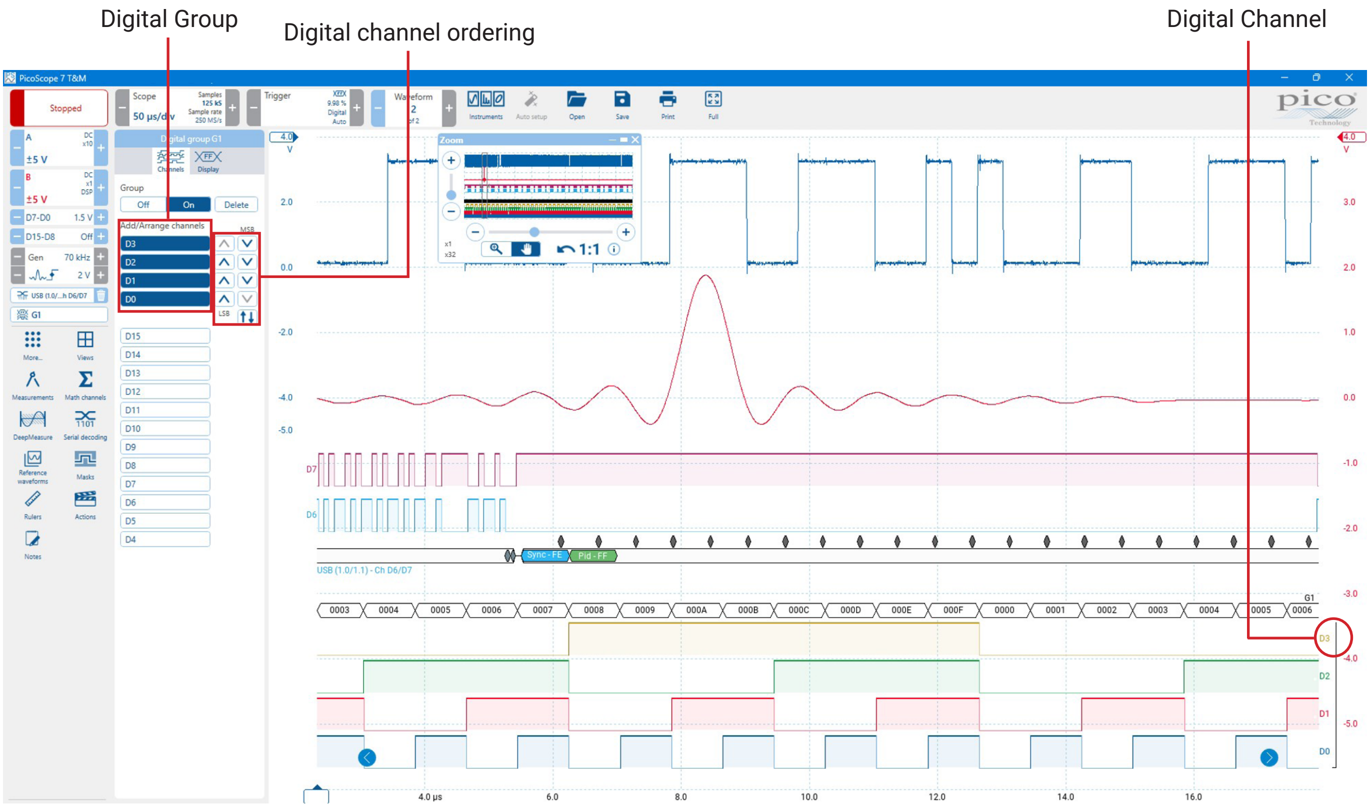


The changes that you can make, are the following:

- **Display** - Clicking the display channel button will turn the channel on or off.
- **Label** - Change the Label name to make it more personally identifiable.
- **Color** and **Invert** - Change the color of the signal and decide if you want to invert or not.
- **Threshold** - The threshold is the voltage at which the digital inputs distinguish a logic '1' from a logic '0'
- **Hysteresis** - Hysteresis allows the digital inputs to reject spurious transitions caused by noise. Only the 6000E series has adjustable hysteresis, on other PicoScopes the hysteresis is fixed in hardware and this control is not shown.
- **Display size** - Change the height of each digital channel trace.
- **Create a group** - Create digital groups to combine multiple digital channels and display them in your chosen format: hexadecimal, binary or analog level.

5.2.3 Digital group options

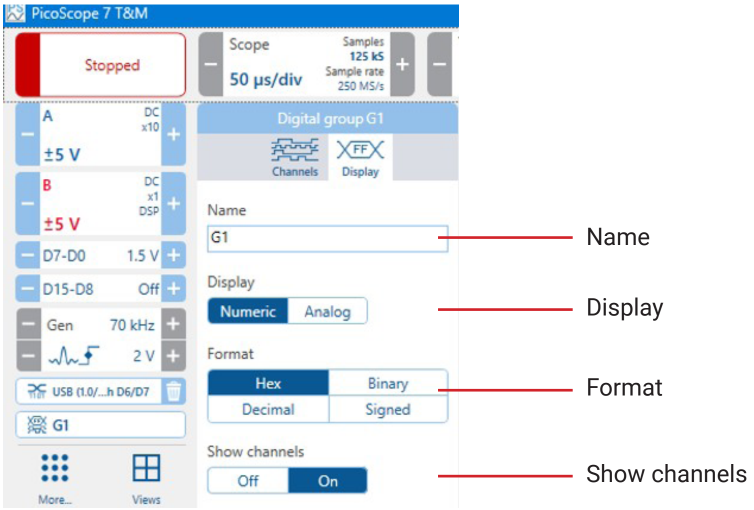
Location: Scope view



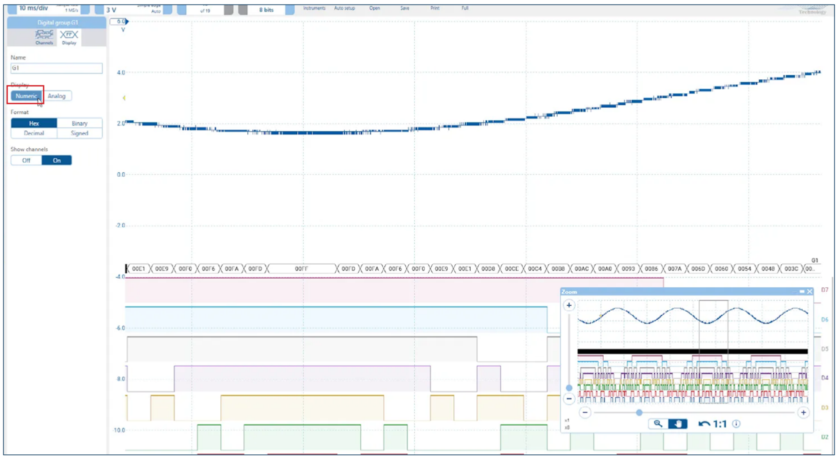
Digital Channel	Displayed in the order in which they appear in the **Digital group** options panel, the digital channels are displayed individually and/or grouped together in a bus display, all the time correlated with the analog channels around the same trigger point.
Digital Group	Groups are created and named in the **Digital group** options panel.
Digital channel ordering	Move channel up/down in group, and reverse order of channels in group.

5.2.4 Digital group display options

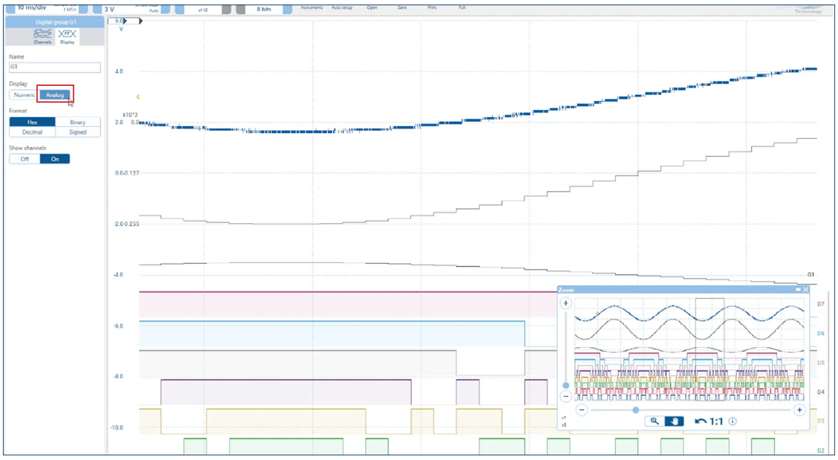
Location: Digital Group options panel > Display



- **Name** - Here you can change the name to make this group more personally identifiable.
- **Display** - Switch between Numeric or Analog display.



Numeric



Analog

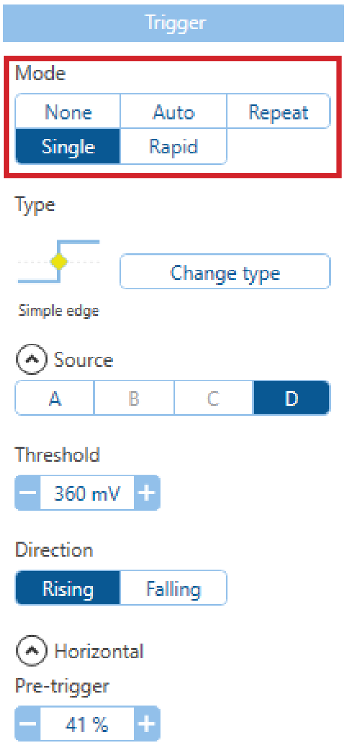
- **Format** - These are the numerical format in which group values are displayed in the scope view and when exporting data.
- **Show channels** - Allows you to turn the display of the individual digital channel signals forming the group, beneath the group value on or off.

5.3 Triggering

The trigger lozenge shows the main trigger settings and opens the trigger options panel when clicked, allowing control of all trigger settings.

5.3.1 Trigger controls

The **Trigger** controls tell the scope device when to start capturing data.



Trigger Mode. The list of available modes varies depending on the type of scope device in use.

None: PicoScope acquires waveforms continually, without waiting for an event to **trigger** on.

Auto: PicoScope waits for a trigger event before capturing data. If there is no trigger event within a reasonable time, it captures data anyway. It repeats this process until you click the **Stop** button. **Auto** mode does not set the trigger level automatically.

Repeat: PicoScope waits indefinitely for a trigger event before displaying data. It repeats this process until you click the **Stop** button. If there is no trigger event, PicoScope displays nothing.

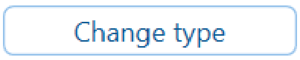
Single: PicoScope waits for one occurrence of a trigger event, then stops sampling. To repeat the process, click the **Go** button. The **Single trigger** is the only type that allows one capture to fill the entire buffer memory.

Rapid: PicoScope instructs the scope device to acquire a sequence of waveforms with the minimum possible delay between them. The display is not updated until the last waveform in the sequence has been captured. When the operation is finished, you can step through the waveforms using the **Buffer Navigation** toolbar.

Note: rapid triggering is available only on certain devices and with collection times of **1 ms/div** or shorter.

ETS: **Equivalent time sampling**. PicoScope captures many cycles of a repetitive signal, then combines the results to produce a single waveform with higher time resolution than is possible with a single capture. For accurate results, the signal must be perfectly repetitive and the trigger must be stable. **ETS** is not available on some PicoScopes or mixed-signal oscilloscopes when digital channels are enabled.

Note: if you select **ETS** when an **advanced trigger** type is enabled, the trigger type will revert to **simple edge** as advanced triggers are not available in ETS mode.

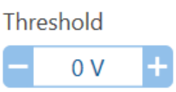


Advanced Triggers. Clicking change type gives you extra trigger types beyond the **simple edge trigger**. The selection of advanced triggers available varies depending on the scope model connected.

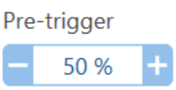
Trigger Channel. This is the channel that PicoScope monitors for the trigger condition. You can set it to any of the scope channels, or to **Ext** or **AuxIO ,** or to a digital input on MSO devices. Some advanced trigger types are only available on analog channels, for example Window trigger types which use more than one trigger threshold.

Rising Edge. Click to trigger on the **rising edge** of the waveform.

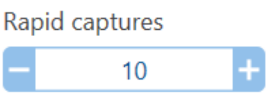
Falling Edge. Click to trigger on the **falling edge** of the waveform. For advanced trigger types, different directions are available.



Threshold. Sets the **voltage threshold** for the trigger. You can also set the **threshold** by dragging the trigger marker up or down the screen.



Pre-trigger (0% to 100%). This parameter controls how much of the waveform appears before the trigger point. It defaults to 50%, which puts the **trigger marker** in the middle of the screen. You can also control this parameter by dragging the **trigger marker** left or right.



Rapid Captures. In **Rapid trigger** mode, this is the number of waveforms to capture in a sequence. They will be captured with the minimum possible dead time between them.

5.3.2 Trigger type selection

Location: **Trigger controls > Change type**

Purpose: allows you to set up more complex trigger types

For more detailed information on Trigger types, descriptions and setup, ***click here***

5.3.3 Trigger marker

The **trigger marker** shows the level and timing of the trigger point.

The height of the marker on the vertical axis shows the level at which the trigger is set, and its position on the time axis shows the time at which it occurs.

You can move the **trigger marker** by dragging it with the mouse or, for greater precision, using the **Trigger** controls.

Other forms of trigger marker

If the scope view is zoomed and panned so that the trigger point is off the screen, the off-screen **trigger marker** (shown above) appears at the side of the graticule to indicate the trigger level.

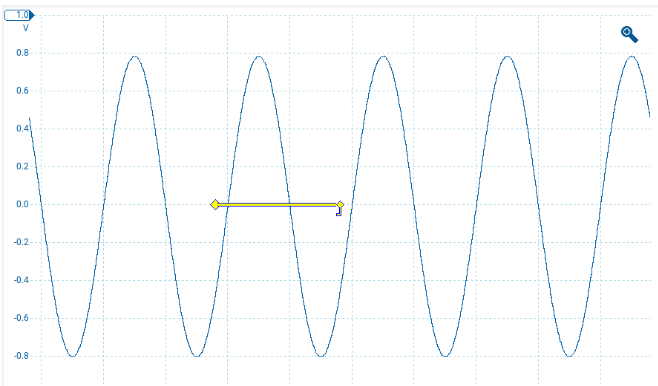
If you have set a time delay, the **trigger marker** is temporarily replaced by the time-delay arrow while you adjust the time delay.

When some advanced trigger types are in use, the **trigger marker** changes to a window marker, which shows the upper and lower trigger thresholds.

For more information, see **Trigger** controls and its subsections.

5.3.4 Time-delay arrow

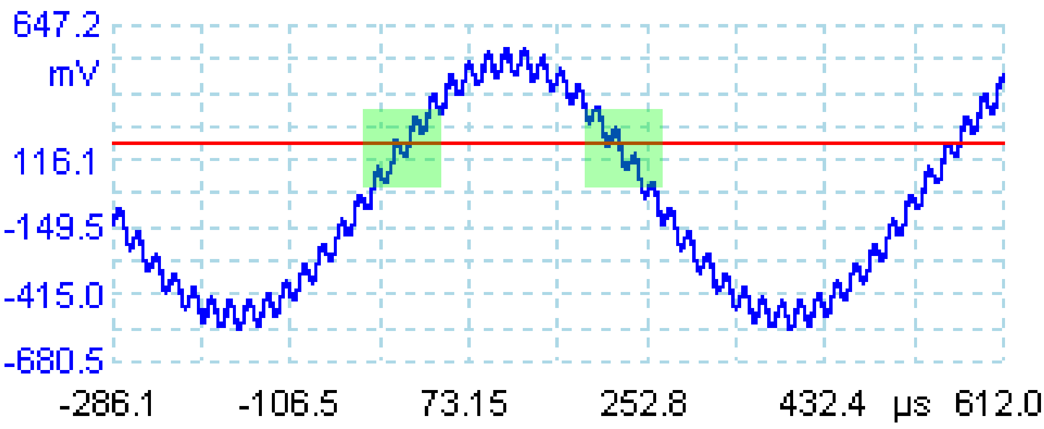
The **time-delay arrow** is a modified form of the **trigger marker** that appears temporarily on a scope view while you are setting up a time delay, or dragging the **trigger marker** after setting up a time delay.



The left-hand end of the arrow indicates the trigger point, and is aligned with zero on the time axis. The right-hand end of the arrow (temporarily replacing the trigger marker) indicates the trigger reference point. You can set up a time delay using the Trigger controls.

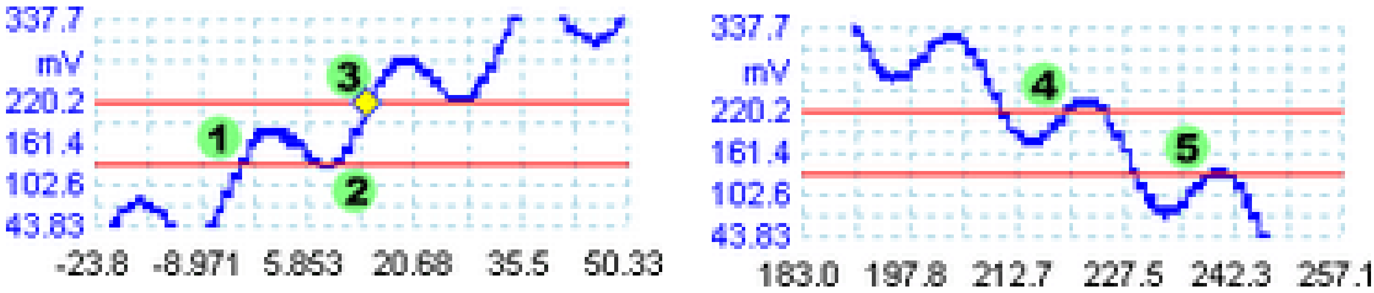
5.3.5 Hysteresis

Hysteresis is a feature of all trigger types in PicoScope software that reduces false triggering on noisy signals. When hysteresis is enabled, a second trigger threshold voltage is used in addition to the main trigger threshold. The trigger fires only when the signal crosses the two thresholds in the correct order. The first threshold arms the trigger, and the second causes it to fire. An example will help to illustrate how this works.



Noisy signal with a single threshold

Consider the very noisy signal above. It is difficult to trigger reliably on this signal with a normal **rising edge trigger** because it crosses the **trigger threshold**, the red line in this picture, several times in one cycle. If we zoom in on the highlighted parts of the signal, we will see how **hysteresis** can help.



Noisy signal with hysteresis threshold

In these zoomed-in views, the original **threshold** is the upper red line, which passes through the trigger diamond. The lower red line is the second **threshold** used by the **hysteresis trigger**.

The signal rises across the lower threshold at (1) and (2), arming the trigger but not firing it. At (3) the signal finally crosses the upper threshold, firing the trigger. On the falling edge of the signal, at (4) and (5), rising edges of noise pulses cause the signal to cross the upper and lower thresholds, but in the wrong order, so the trigger is not armed and does not fire. Thus the trigger occurs at only one well-defined point in the cycle (3), despite the noise on the signal.

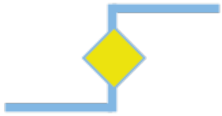
Hysteresis is enabled by default for all trigger types. It is set automatically for simple edge trigger, and can be adjusted for other trigger types by using the Hysteresis controls in the Trigger options panel to let you change the hysteresis voltage as a percentage of full scale. The trigger marker shows the size of the hysteresis window.

In PicoScope, hysteresis is measured as a fraction of the full voltage range. For example, if the scope is set to a voltage range of ±1 V, then a hysteresis setting of 1% gives a hysteresis of 1% x 2 V = 20 mV.

5.3.6 Simple Edge

As an oscilloscope user, you will be familiar with the standard type of triggering provided on all instruments. PicoScope software calls this the Simple Edge trigger, and uses it by default when you enable triggering. The scope monitors the incoming signal and waits for the voltage to rise above (or fall below) a set threshold, then causes the scope to capture and display the waveform. This method is adequate when the signal consists of repetitive pulses or cycles like those in a pure sine or square wave.

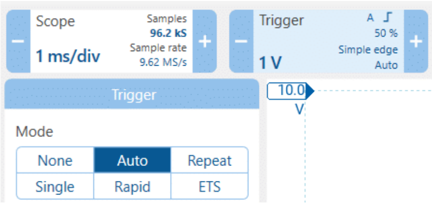
What is the symbol for Simple Edge trigger:



Configuring the Simple Edge trigger:

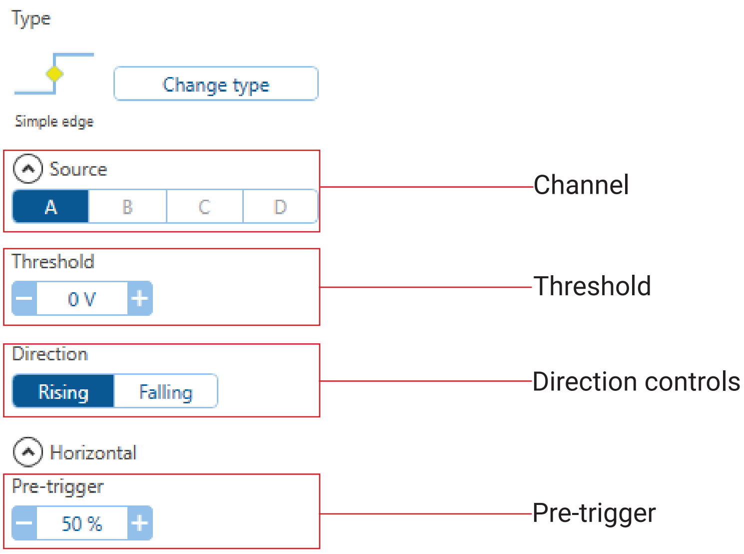
PicoScope software lets you avoid the complexity of advanced triggers unless you really need to use them. We will first look at the trigger mode control and the Simple Edge trigger types.

Trigger mode control



The trigger mode control tells the scope whether to wait for a trigger condition and what to do next. None, Single, Repeat and Auto modes are described in 5.3.1 Trigger Controls. Depending on the PicoScope model and timebase settings, certain PicoScope models have special modes including Rapid and ETS (Equivalent Time Sampling).

Standard triggering controls



Channel - the input to trigger on. This can be one of the analog input channels, or a digital input pin on MSOs, or an Ext or AuxIO trigger input if available.

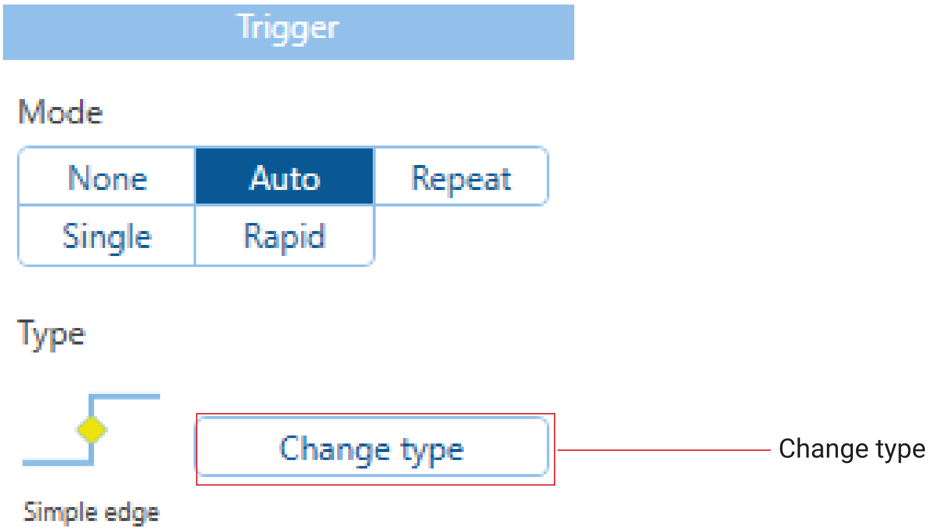
Threshold - the voltage at which the scope will trigger

Direction - whether the scope should trigger when the signal voltage rises above the threshold, or when it falls below it.

Pre-trigger - the proportion of the display which is before the trigger point. The pre-trigger can be set from 0 % (meaning the whole display is after the trigger event) to 100% (meaning the whole display shows what leads up to the trigger event). The default 50% pre-trigger means the trigger event is exactly in the middle of the display.

5.3.7 Advanced trigger types

To access advanced trigger types, click change type, as shown below:

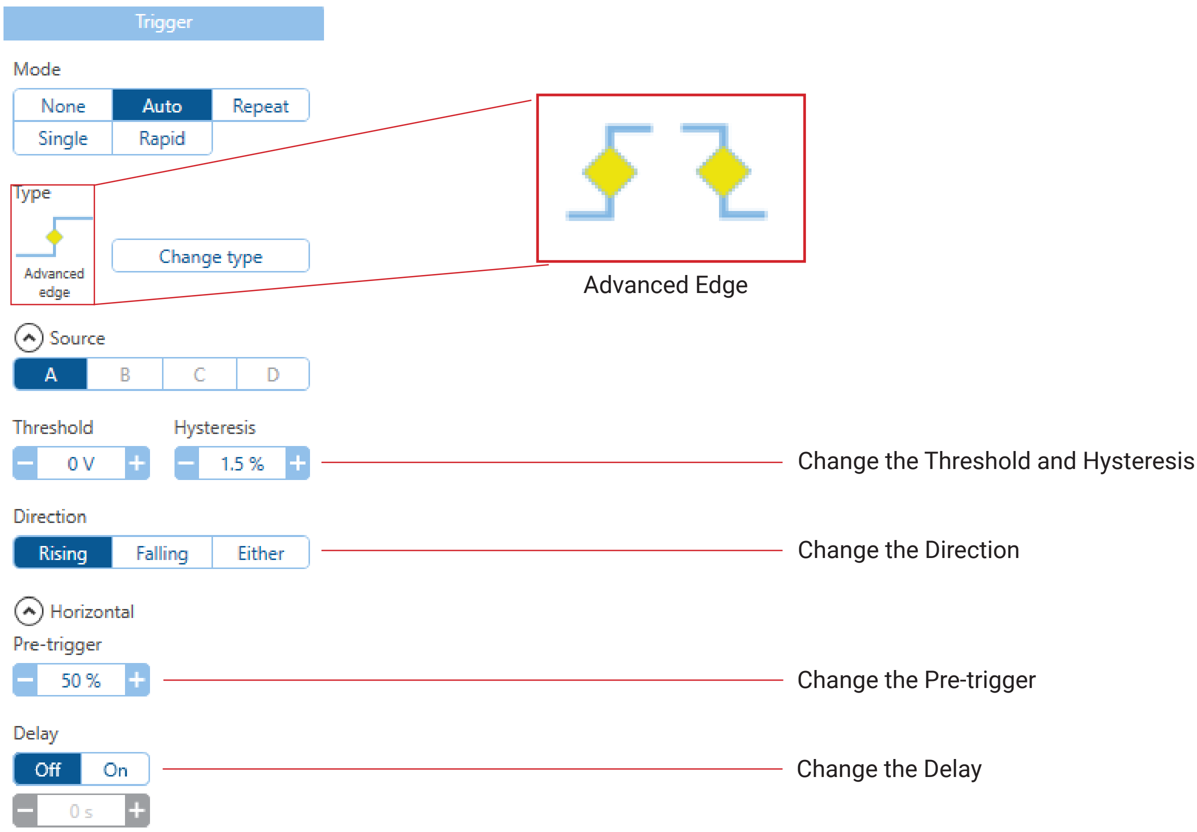


Below is an interactive list of advanced trigger types that are currently available, click each button to go to more information on that trigger type:

Simple edge trigger	Advanced edge trigger	Window trigger	Interval trigger
Pulse width trigger	Window pulse width trigger	Level dropout trigger	Window dropout trigger
Rise/Fall time trigger	Digital trigger	Logic trigger	

Not all PicoScope models support all trigger types. Refer to your product's datasheet for full specifications.

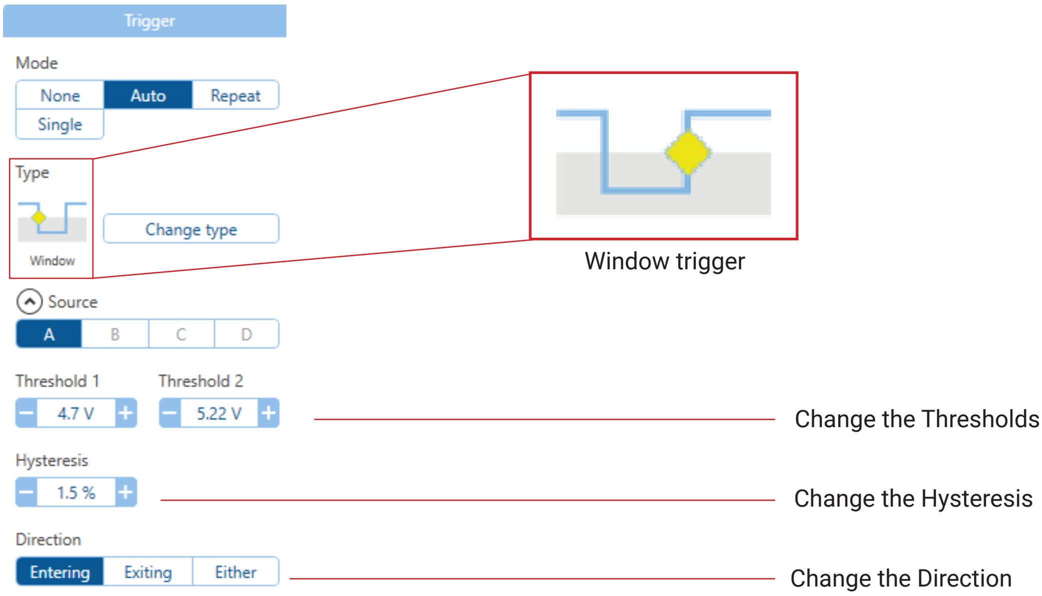
5.3.8 Advanced edge trigger



The Advanced Edge trigger provides the standard rising, falling and dual (either) edge conditions. Dual edge triggering allows you to check the widths and voltages of positive and negative pulses as the same time. It is useful for rapidly spotting jitter and noise problems, and for displaying eye patterns.

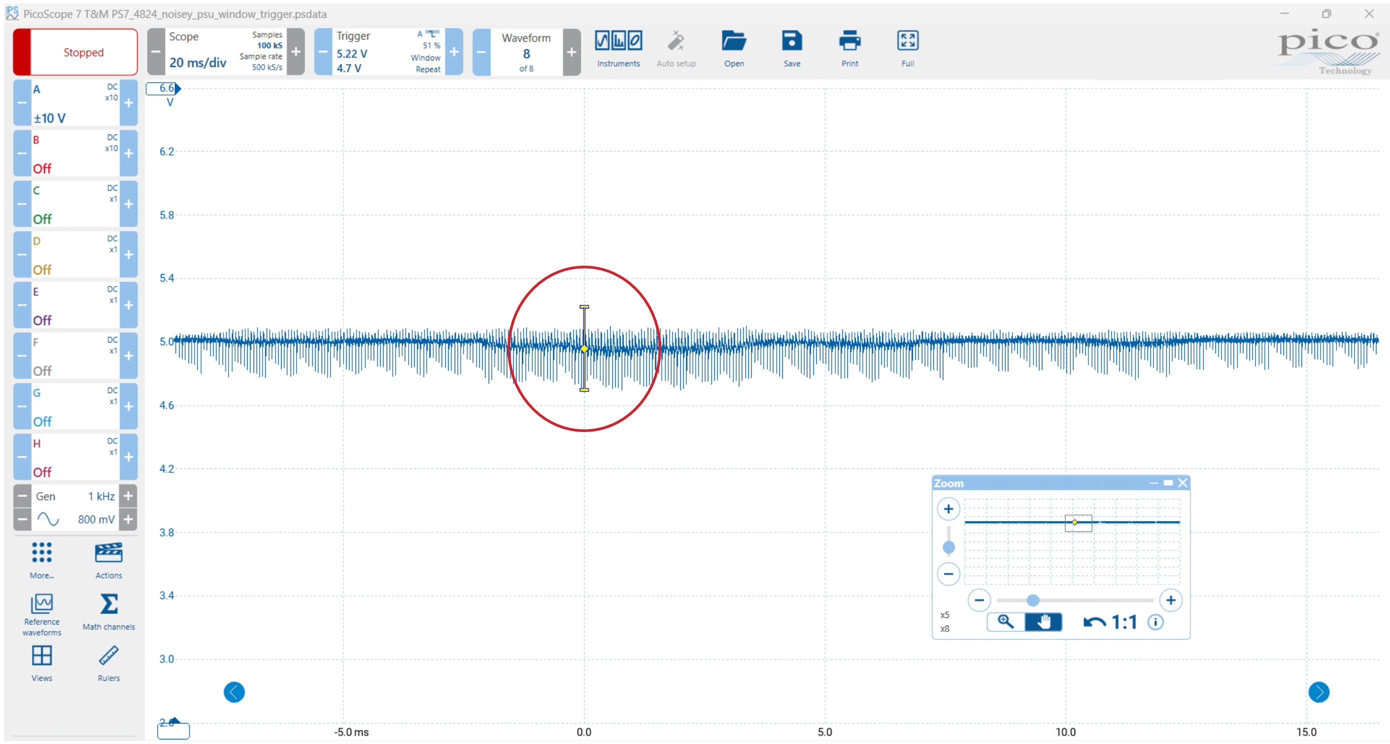
The Advanced Edge trigger also provides adjustable hysteresis. For more information, see section ***Hysteresis***

5.3.9 Window trigger



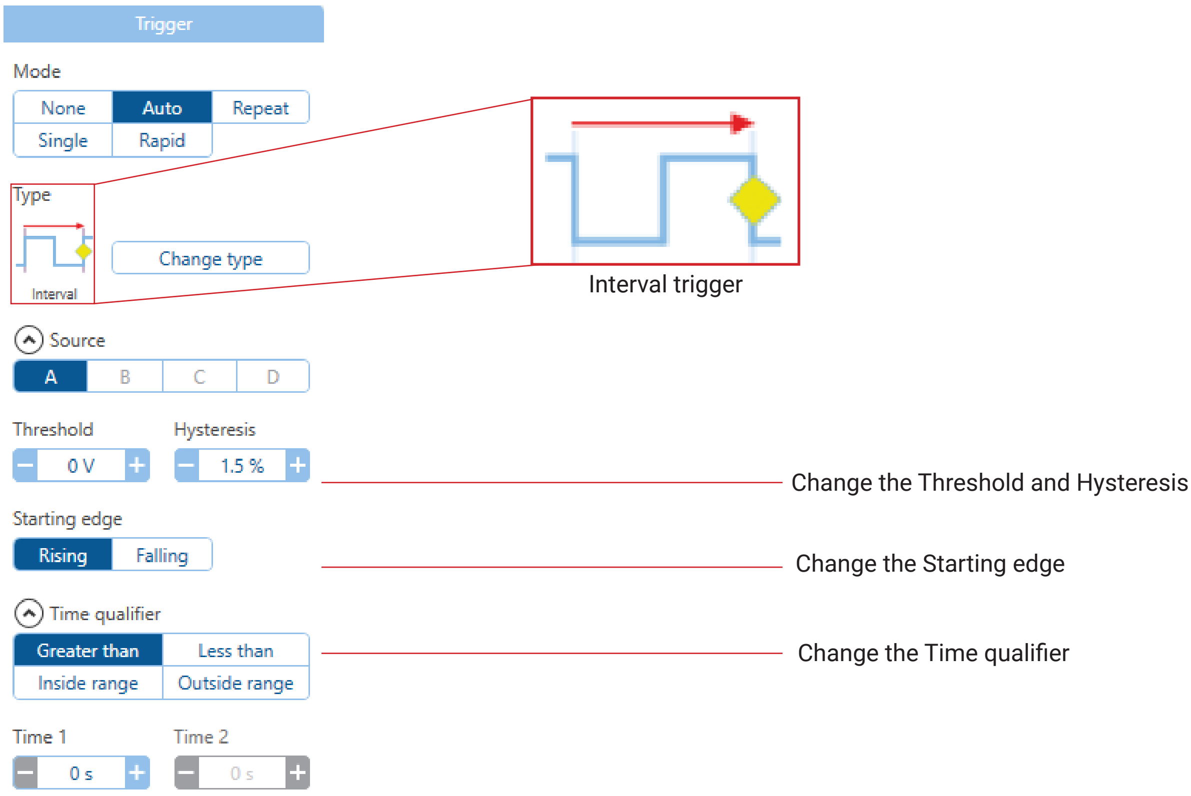
This trigger detects the moment when the waveform enters or leaves a voltage range, defined as being between Threshold 1 and Threshold 2. This allows you to search for overvoltages and undervoltages at the same time. In this example a 5 volt power supply is monitored with thresholds of 4.7 and 5.22 volts. The window trigger will detect both the positive and negative excursions outside this range.

The Direction control specifies whether the trigger operates when the signal enters the window, exits it, or both. The Threshold 1 and Threshold 2 controls define the upper and lower limits of the voltage window. It's not important which threshold is the larger voltage.



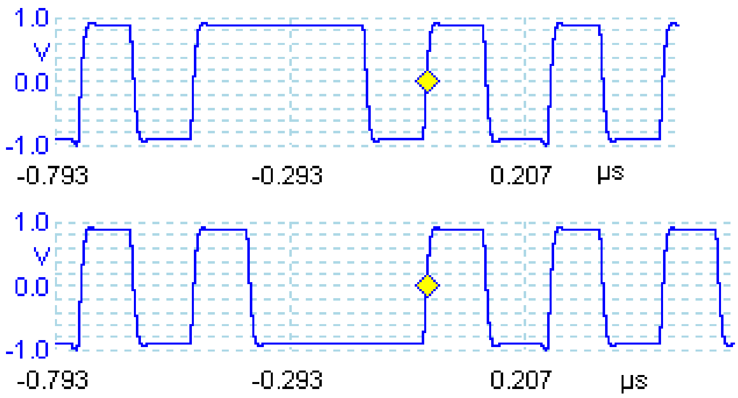
Window trigger marker in the above image shows the upper and lower thresholds defining the window.

5.3.10 Interval trigger

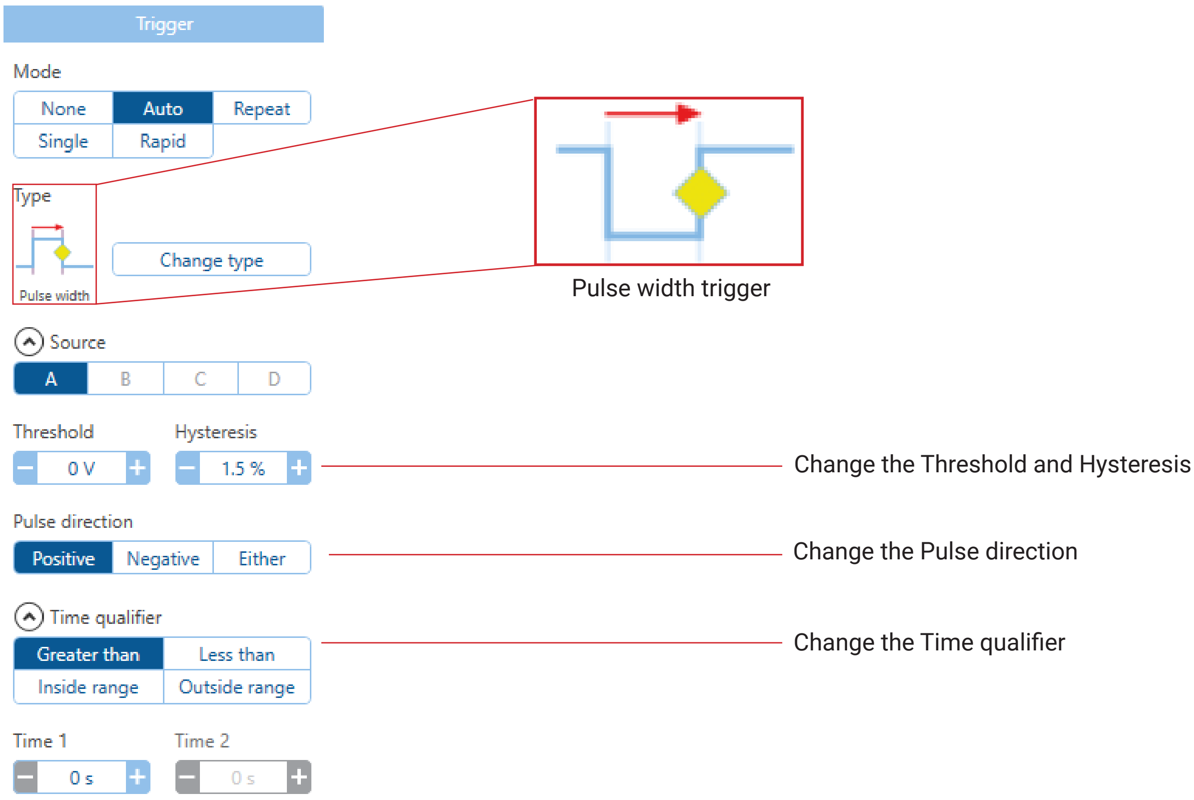


Interval trigger triggers when the time between two consecutive rising edges, or two consecutive falling edges, meets a time condition. This trigger helps you find missing or mistimed edges, or changes in signal frequency.

The picture shows two examples of a 4 MHz clock waveform with a missing pulse. You could use a pulse width trigger to search for the extended high pulse in the top example or the extended low pulse in the bottom example, but the interval trigger lets you find both errors without having to change the trigger type. Setting an interval trigger of "rising, greater than 300 ns" will detect both cases. The trigger point is set to the first rising edge after the long interval.



5.3.11 Pulse width trigger



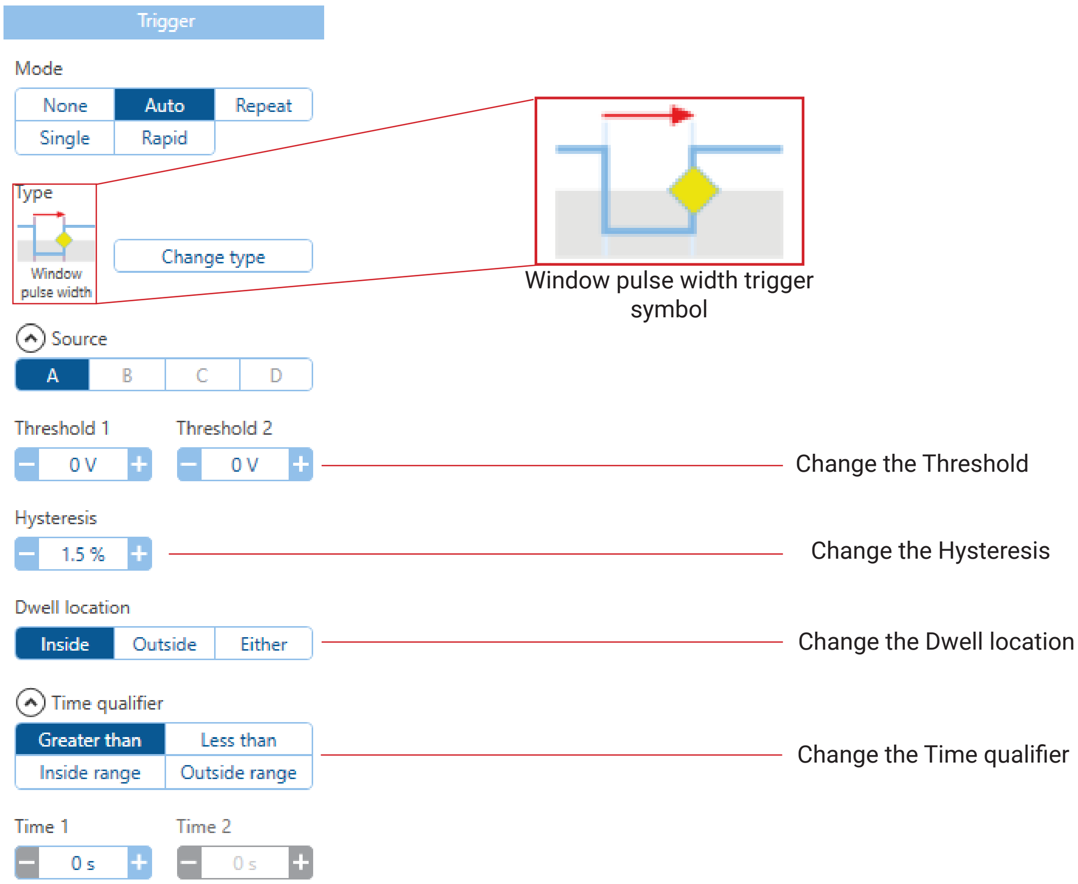
This trigger enables you to trigger exclusively on pulses with a specified range of widths. This can be useful for finding glitches, rare events in synchronous control signals such as write-enables, or extreme values in pulse-width modulated (PWM) signals.

The Pulse direction control specifies whether you want to trigger on either positive or negative pulses. A positive pulse is a period in which the signal remains continuously above the threshold.

The Condition control specifies whether you are looking for pulses wider or narrower than a specified width, or inside or outside a specified range of widths.

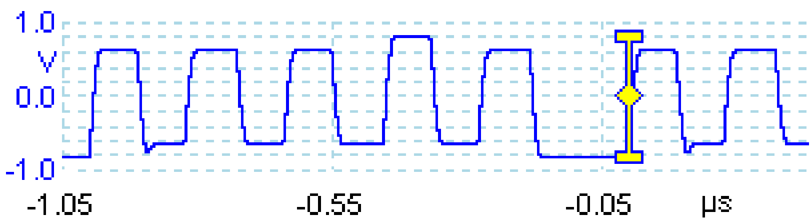
The pulse width trigger evaluates the time condition at the end of each pulse in the input signal. If the pulse meets the chosen time condition, the scope triggers.

5.3.12 Window pulse width trigger

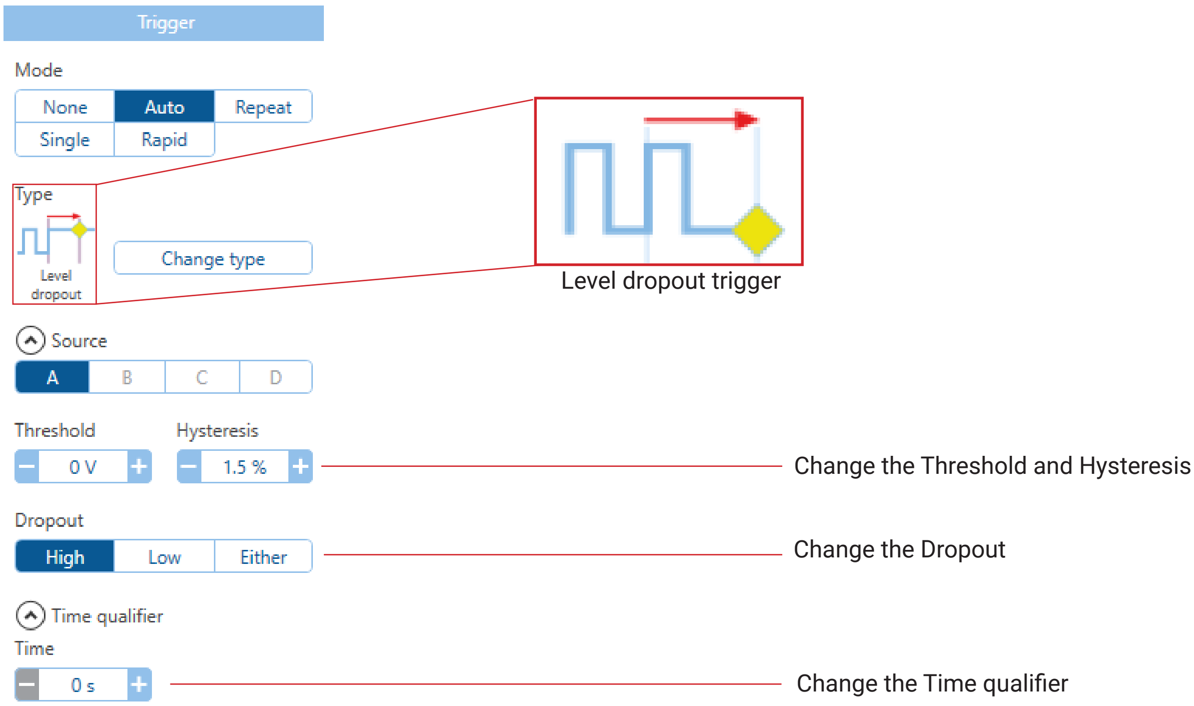


This trigger is a combination of the window trigger and the pulse width trigger. It detects when the signal enters or leaves a voltage range for a specified period of time. In the example shown a nominally ±700 mV signal has occasional overvoltages and undervoltages, but we have set the dwell time to “greater than 100 ns” so that only abnormally wide pulses that go outside this range are detected.

This trigger can be used to detect runt pulses, but the Runt trigger described later is easier to use in simple cases.



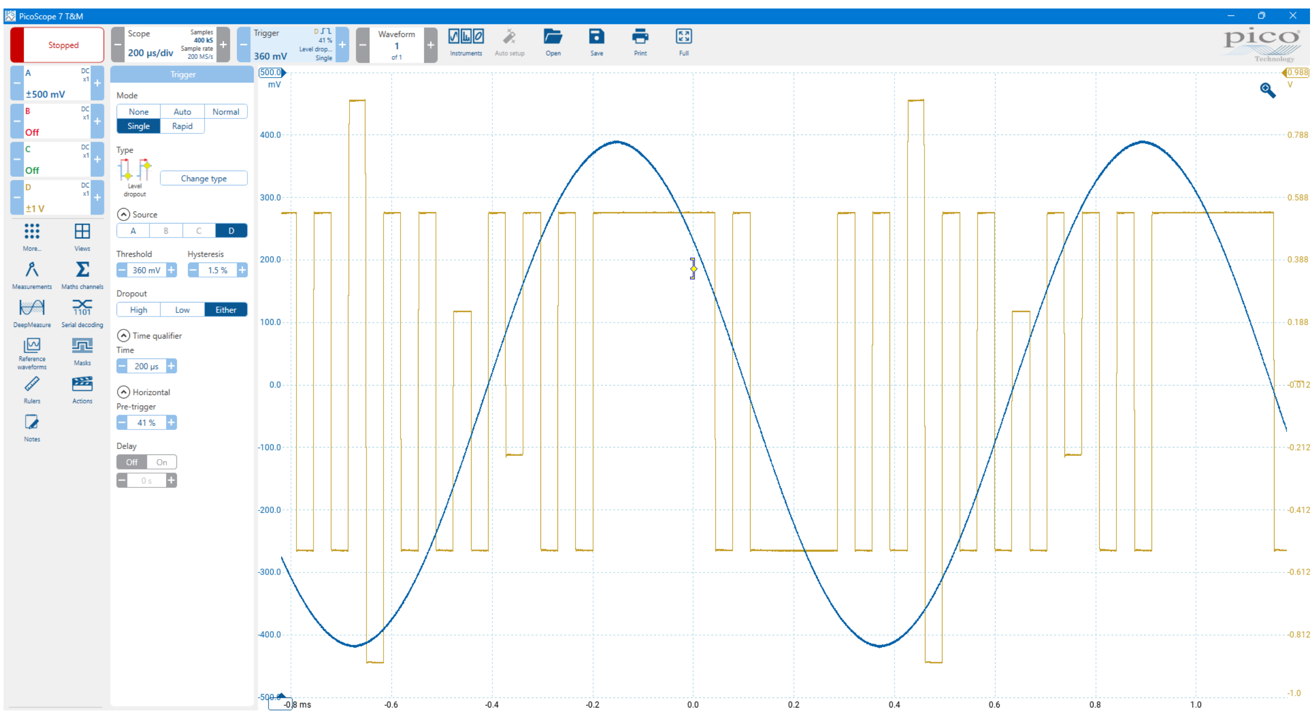
5.3.13 Level dropout trigger



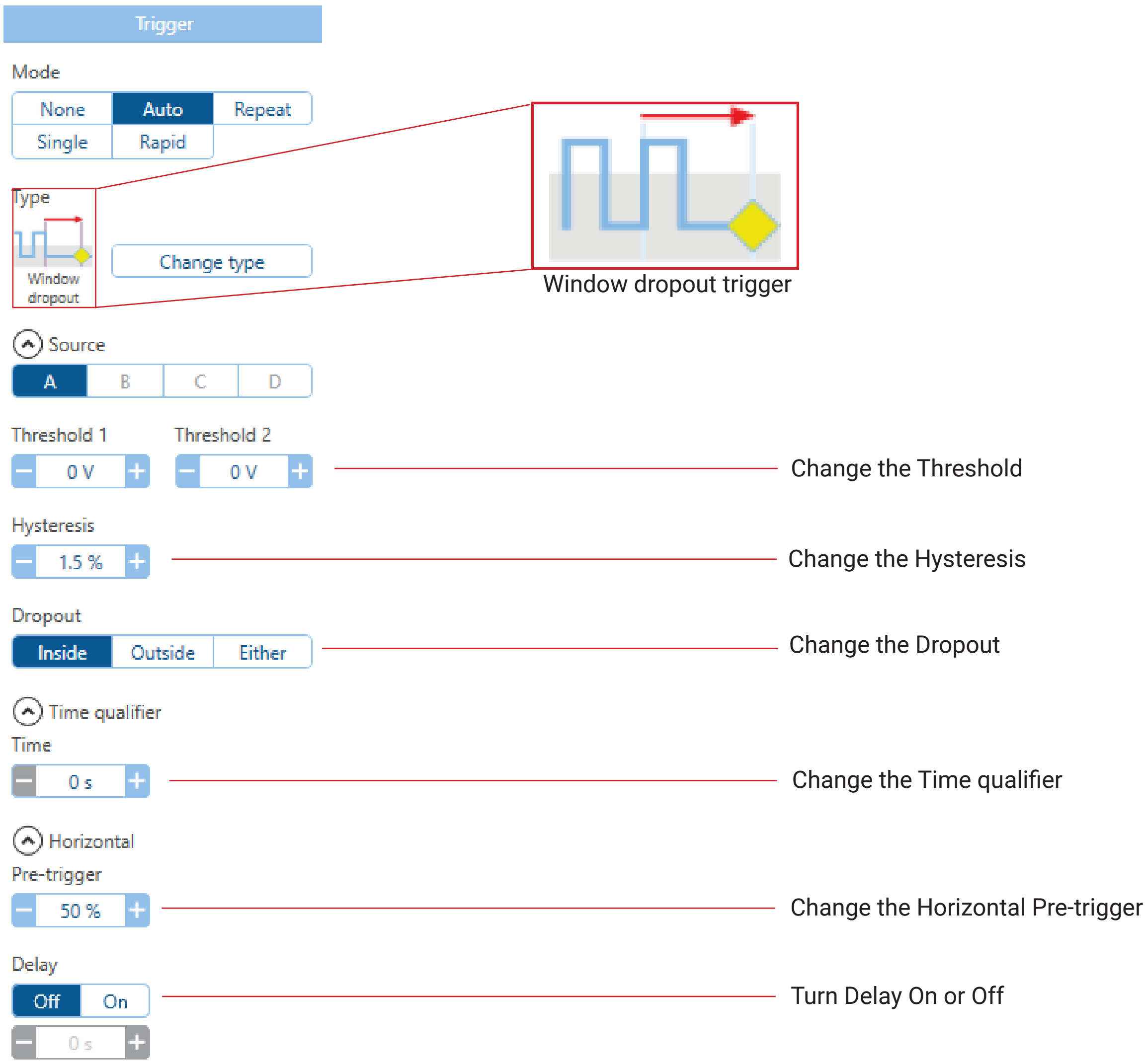
This trigger detects an edge followed by a specified time with no edges. This is useful for triggering on the end of a pulse train. This image shows the end of a pulse train in which edges normally occur every 20 µs. The dropout trigger has detected that no edges have occurred within 40 µs.

The Dropout control specifies whether to trigger when the signal remains high, low, or in either state relative to the threshold.

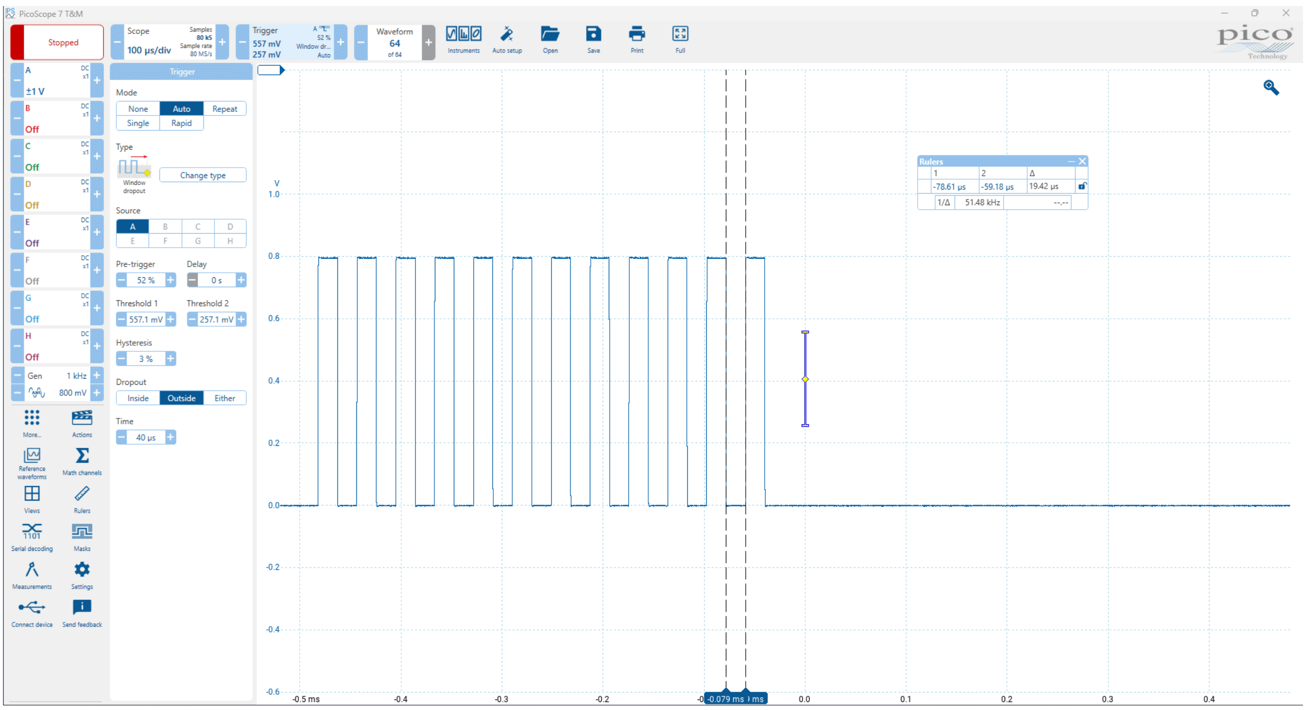
Unlike the pulse width trigger, the dropout trigger triggers as soon as the time condition has passed without waiting for the end of a "pulse".



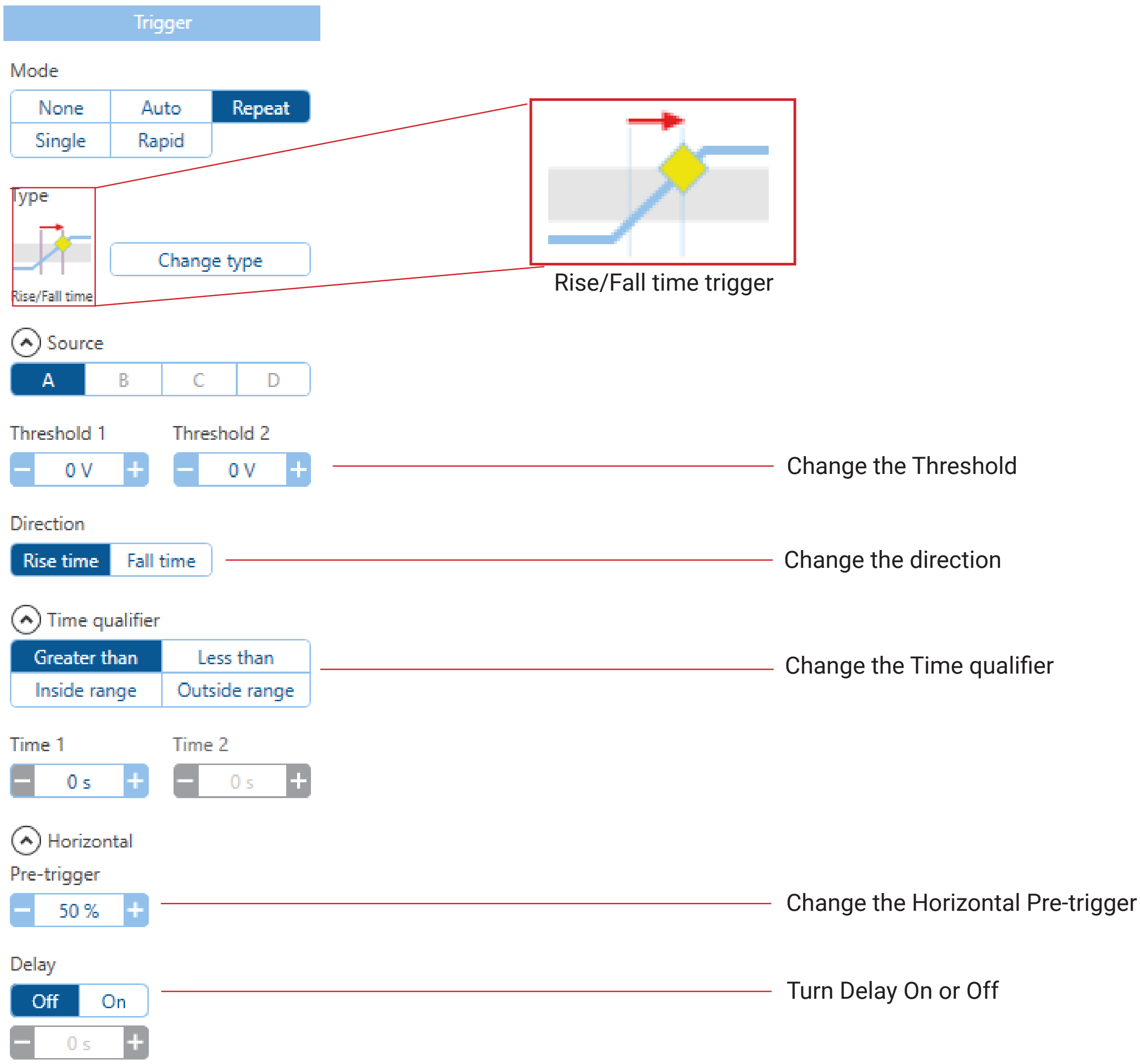
5.3.14 Window dropout trigger



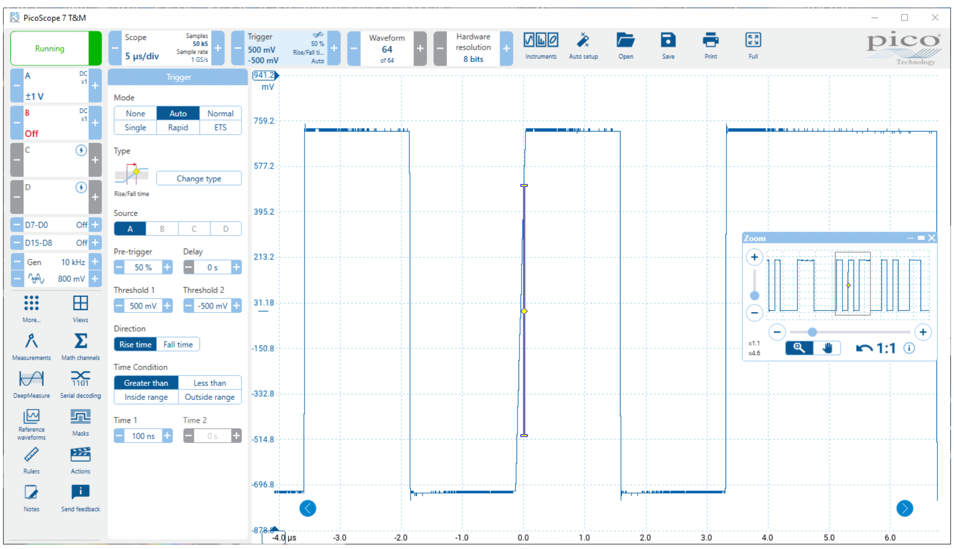
This trigger is a combination of the Window and Level Dropout triggers. It detects when the signal enters or leaves a voltage range and stays there for a specified time. This is useful for detecting when a signal gets stuck at a particular voltage. In this example the window dropout trigger was set to a 40 µs delay with a +250 mV to +550 mV window. It ignored the pulses, which entered and exited the window rapidly, but detected when the signal remained outside the window for more than 40 µs.



5.3.15 Rise/Fall time trigger

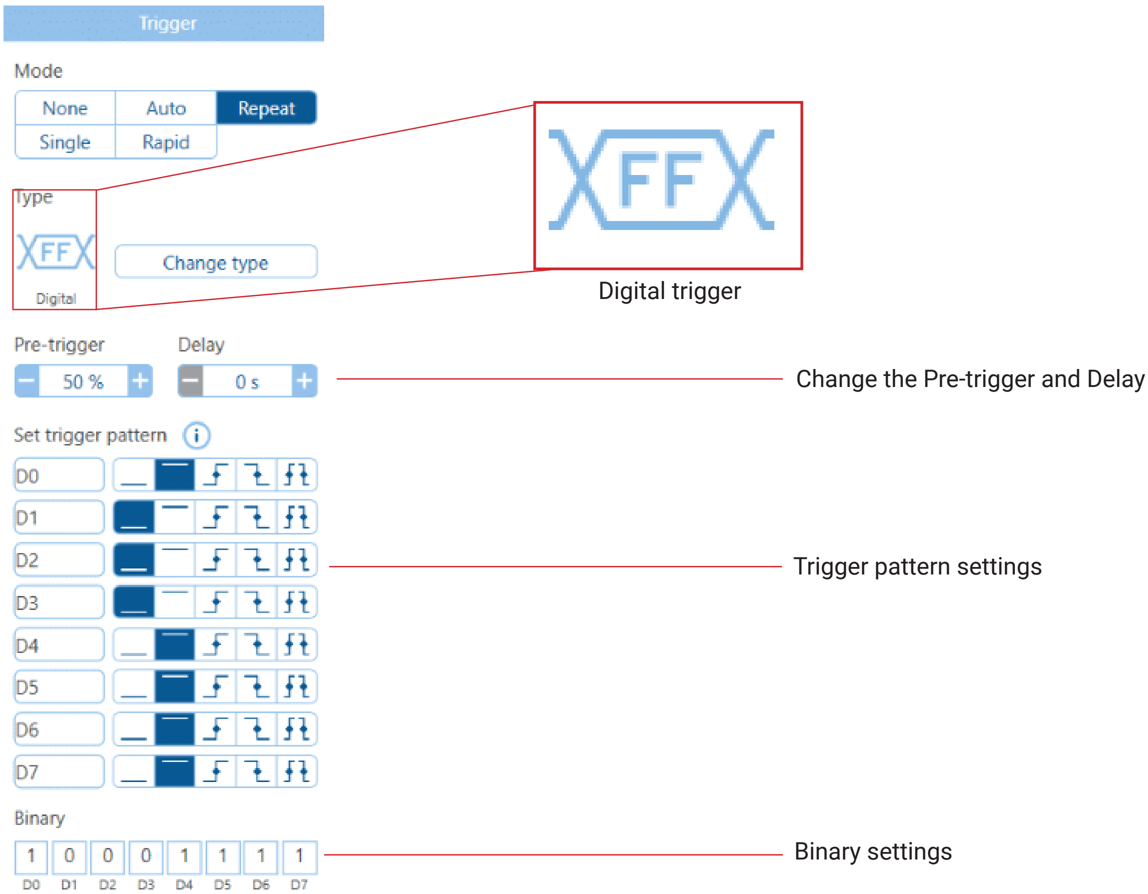


This trigger measures the time the signal takes to transition between two voltage thresholds: from a lower threshold to an upper threshold in the case of rise time, or the opposite in the case of fall time. The scope triggers when the measured transition time meets a user-specified time condition: greater than, less than, inside a range or outside a range of times. Unlike window pulse width trigger, rise/fall-time trigger allows you to trigger specifically on rising edges or on falling edges which meet a time condition. This can be useful if the expected slew rate of rising and falling edges is different in your system.



In this example above the rise time trigger was set to trigger on rising edges which transitioned from -500 mV to +500 mV in greater than 100 ns. It ignored the rising edges which were faster than 100 ns, and the falling edges regardless of their rate, and triggered only on the slower rising edge.

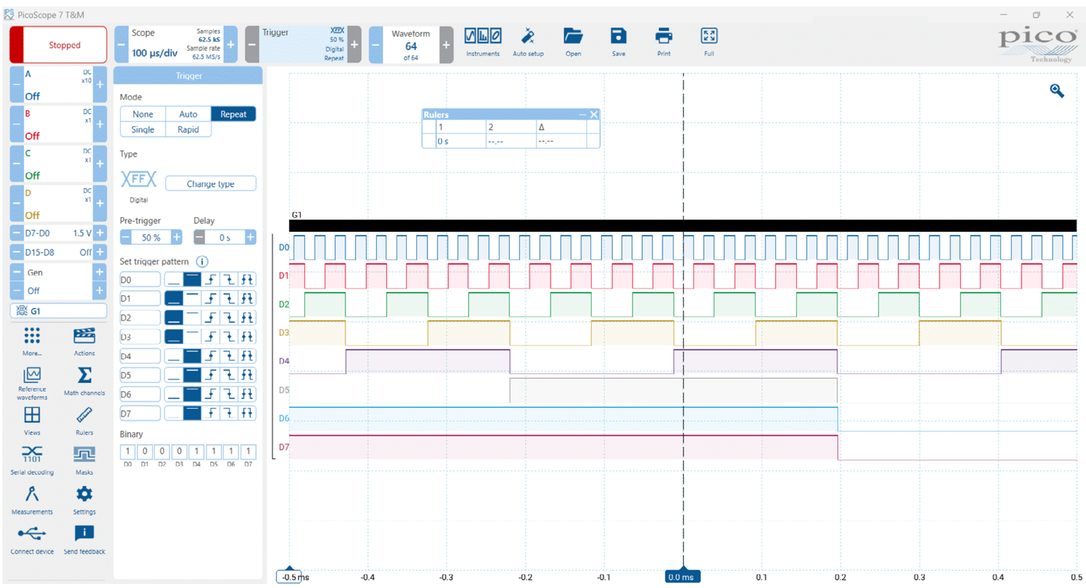
5.3.16 Digital trigger



The digital trigger is available on PicoScope mixed signal oscilloscopes (MSOs). It detects a pattern, and optionally a transition, on the digital inputs. If no state is set for a particular digital bit, that bit is "don't care" and is not included in the pattern. All bits with a state set must match the pattern simultaneously (on the same sample) for the scope to trigger.

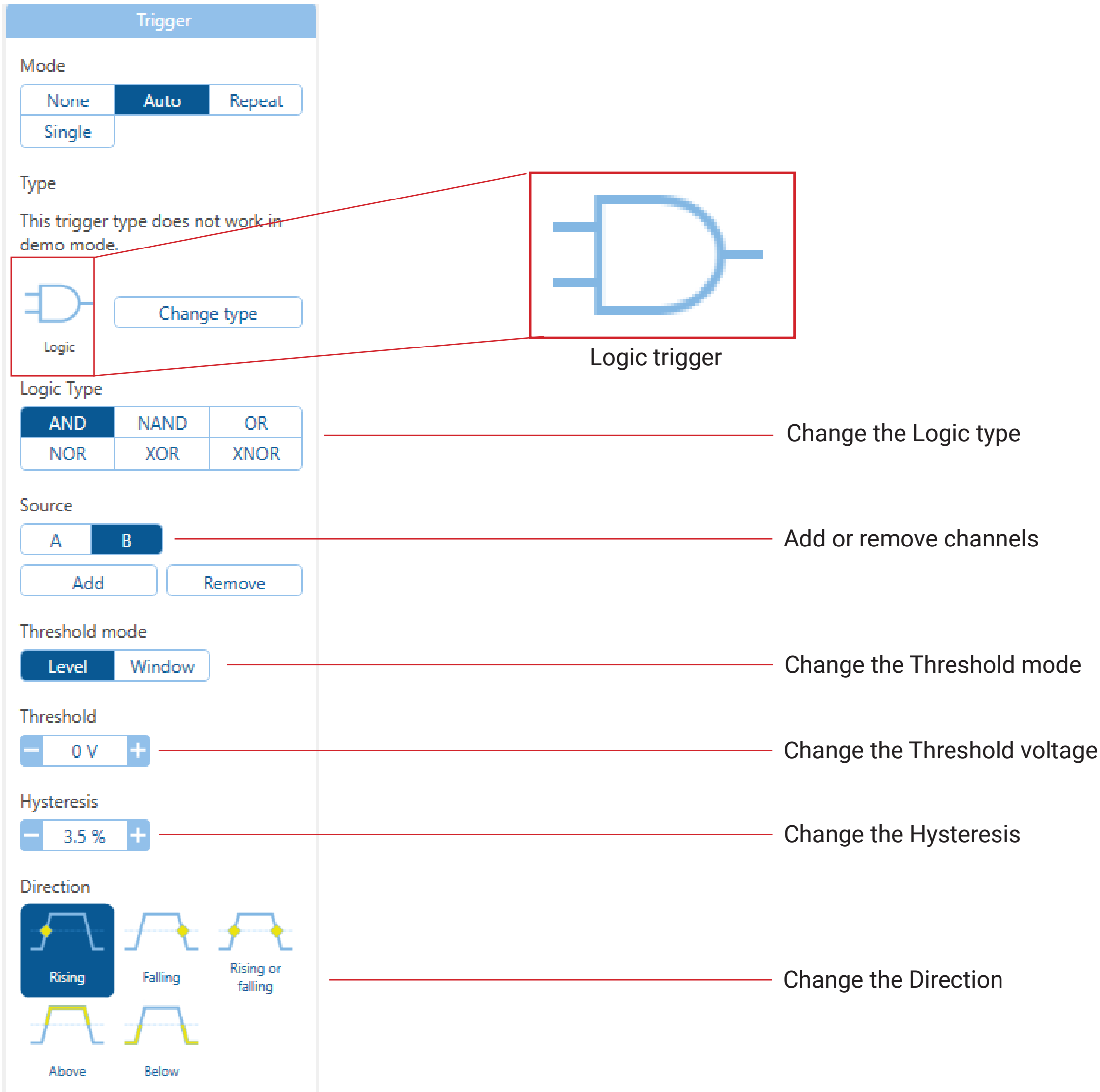
At most one bit can be set to a transition (rising, falling, or either edge) as part of the pattern. This might be used for a clock signal, for example, so the scope triggers only when the pattern is met on the rising edge of a clock.

Here we set the trigger to look for the pattern 10001111 on bits D0...D7.



The Digital trigger can be fed into the Logic trigger, described in section ***Logic trigger***, to allow you to look for composite conditions of the analog and digital inputs.

5.3.17 Logic trigger



This trigger detects a logical combination of up to four of the scope's inputs.

Instead of the AND function, you can combine the channels using NAND, OR, NOR, XOR or XNOR.

Use the Add and Remove buttons to select the channels to be included in the logical function. The scope will trigger when the chosen logical combination of the selected channels returns "true". For example with the AND function, all the selected channels must simultaneously meet their trigger conditions on the same sample. With the OR function, the scope triggers when any one or more of the selected channels meets its condition.

The channel A to B (or D, or H, depending on the number of inputs the oscilloscope has) controls have the most options. With level-qualification selected, the channel triggers when above or below the threshold. With window-qualification selected, the channel triggers when inside or outside the specified voltage window. The Ext and AuxIO settings controls have only the level-qualified options.

The Logic control specifies how the inputs are combined to produce a trigger condition using the Boolean functions AND, NAND, OR, NOR, XOR and XNOR. For example: to trigger on all of the channel conditions being met, choose AND; or to trigger on exactly one of two (or an odd number out of more than two sources) being met, select XOR.

6. Waveform display

6.1 Zoom

PicoScope 7 offers several intuitive ways to zoom in and out of your waveforms for detailed analysis.

- **Mouse Wheel** Scrolling up zooms in, while scrolling down zooms out.
- **Touchscreen** Pinch to zoom
- **Zoom button** Dedicated zoom button is available in the righthand corner of the display, see Fig.1.
- **Zoom Overview** A miniature overview of the entire waveform is displayed when zoom is in use. You can directly select and drag a region within this overview to zoom in on the corresponding portion of the main waveform display, see Fig.2
- **Panning** Once zoomed in, you can easily pan left or right across the waveform using the horizontal arrows or by dragging the waveform itself using the **hand** tool within the display area.

These versatile zoom features allow you to precisely examine specific areas of interest within your captured waveforms, facilitating accurate measurements and analysis of signal characteristics.

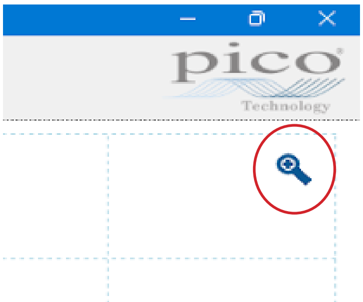


Fig.1

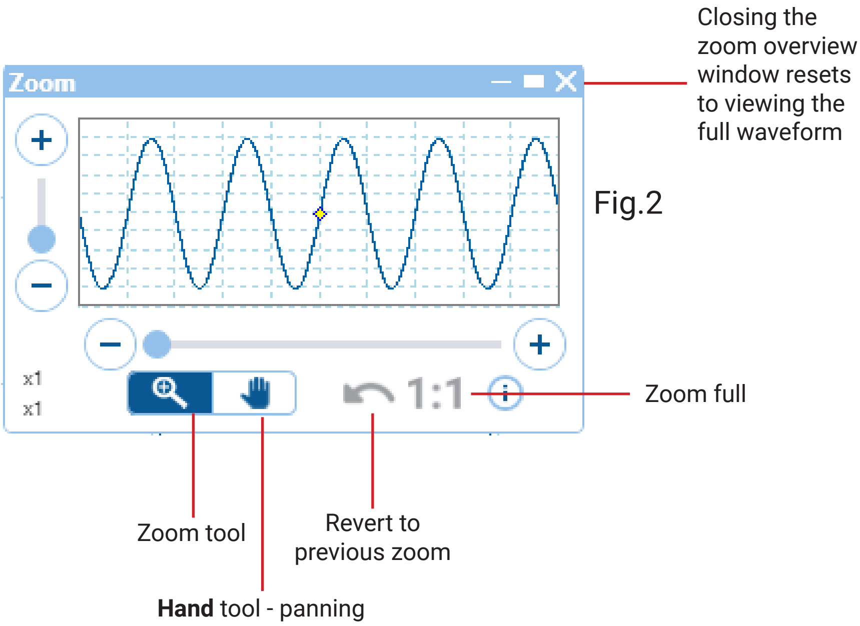
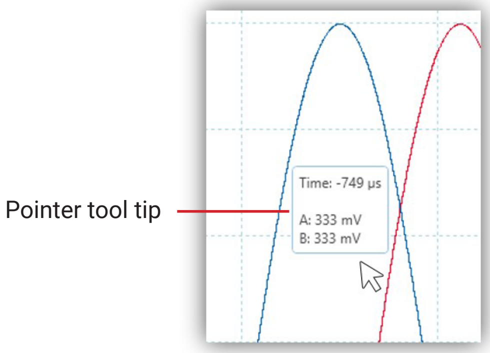


Fig.2

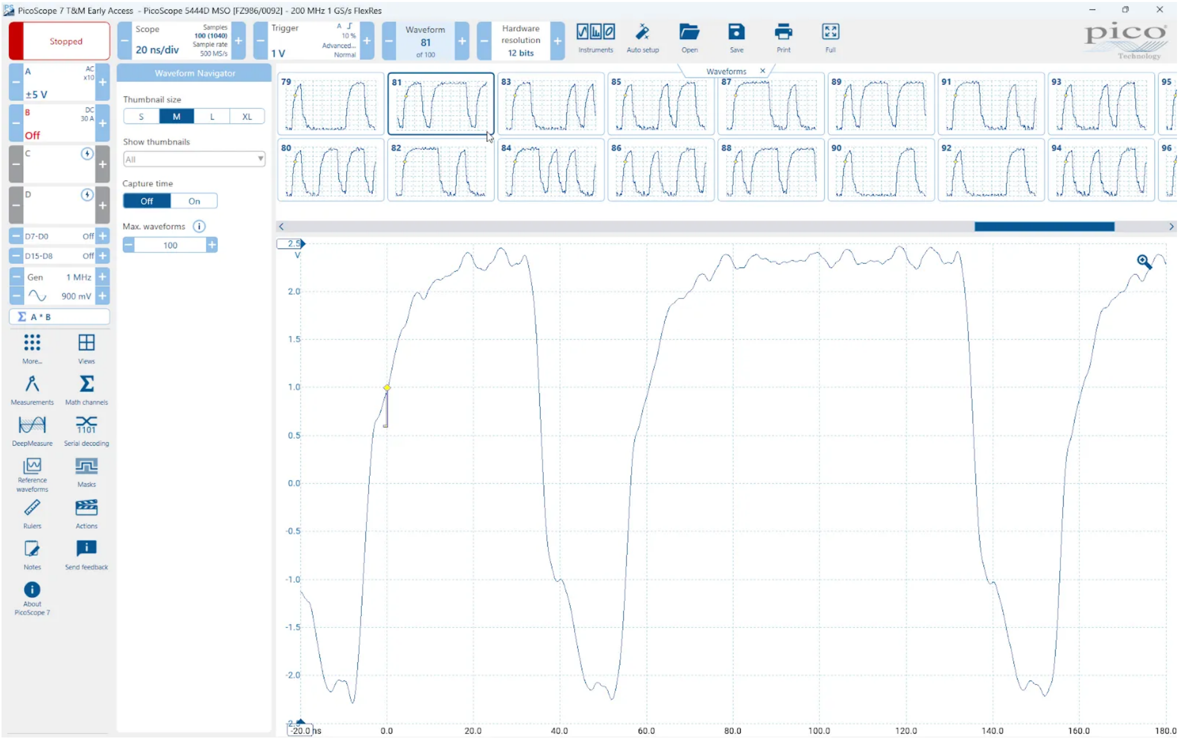
6.2 Pointer tool tip

The pointer tool tip is a box that displays the horizontal and vertical axis values at the mouse pointer location. It appears temporarily when you click the background of a view.



6.3 Buffer navigation toolbar

The **Buffer Navigation** toolbar allows you to select a waveform from the **waveform buffer**.



What is the waveform buffer?

The PicoScope 7 waveform buffer is a powerful tool that can be used to capture, store, and analyze waveforms. It is a circular buffer that can store up to 40,000 waveforms. This allows you to capture and review waveforms even if you miss a glitch or other transient event. The waveform buffer can also be used to perform mask limit testing and other advanced analysis tasks.

The waveform buffer is enabled by default in PicoScope 7. To view the waveform buffer, click on the **Waveform** tab in the main window. This will open the Waveform Buffer Navigator. The Waveform Buffer Navigator allows you to view and navigate through the waveforms in the buffer. When saving a PicoScope data file, you can save all or a selection of the waveforms in the buffer to review or process later.
The waveform buffer is a valuable tool for anyone who needs to capture and analyze waveforms. It is especially useful for troubleshooting electronic circuits and systems.

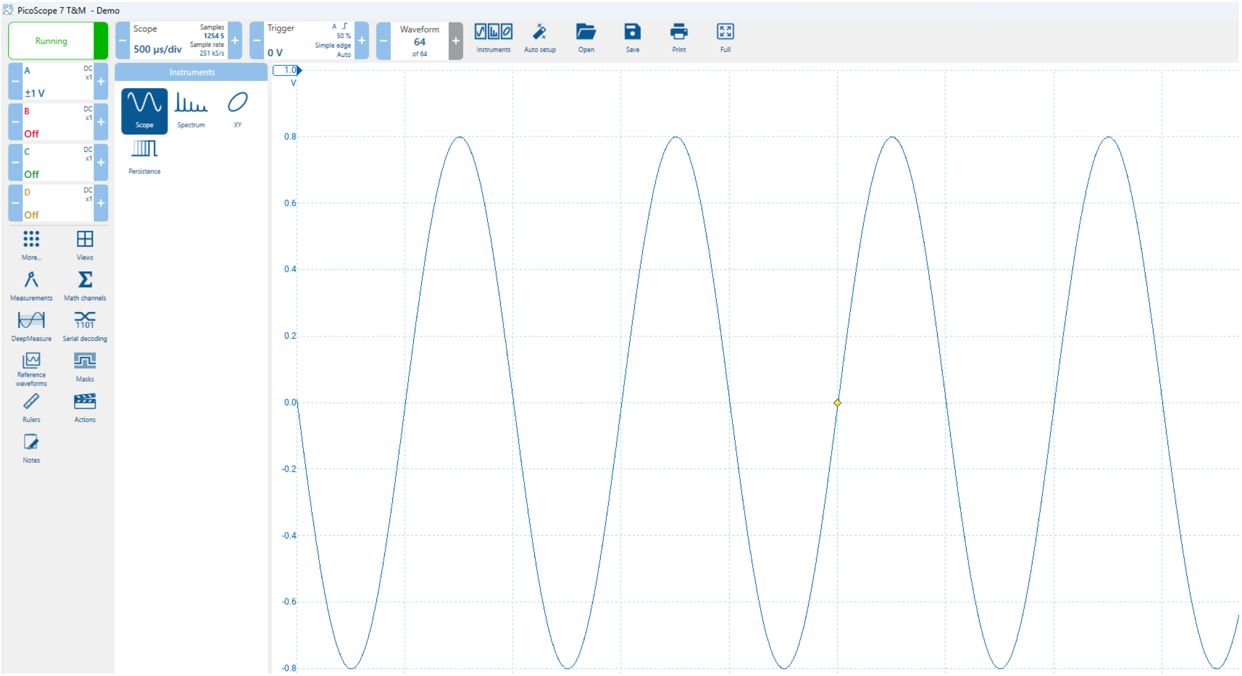
6.4 Instrument setup

Location: **Instruments > Scope/Spectrum/XY/Persistence**

The **Instruments Setup** controls the time-related or frequency-related settings of your oscilloscope.

Selecting an Instrument chooses the main operating mode of the PicoScope software, similar to changing to a different piece of benchtop instrumentation.

6.4.1 Scope View

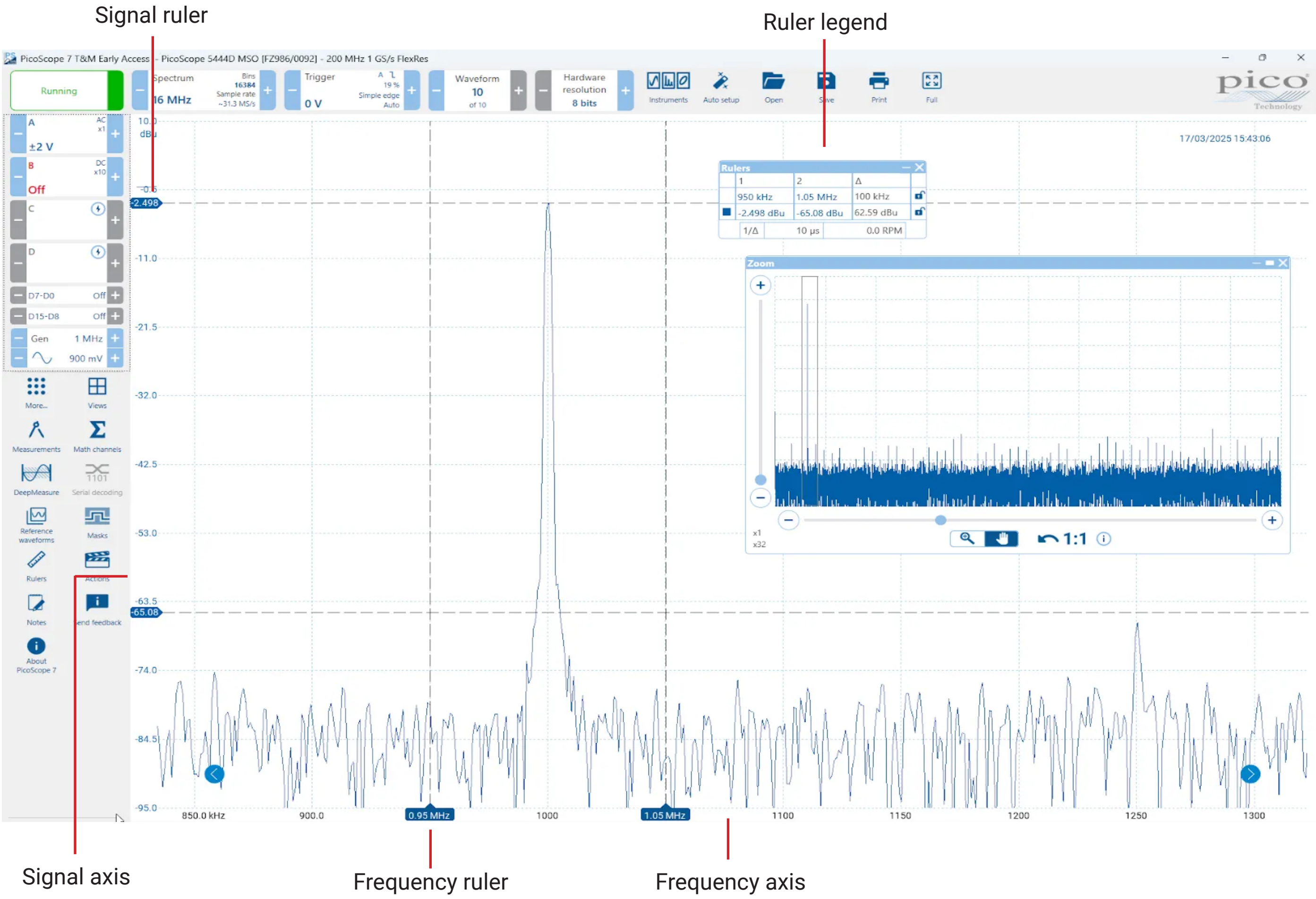


A scope view (shown above) shows the data captured from the scope as a graph of signal amplitude against time. A scope view shows you one or more waveforms with a common horizontal time axis, with signal level shown on one or more vertical axes.

Each view can display to 8 waveforms, including math channels and reference waveforms. To see more waveforms at once, just open another view.

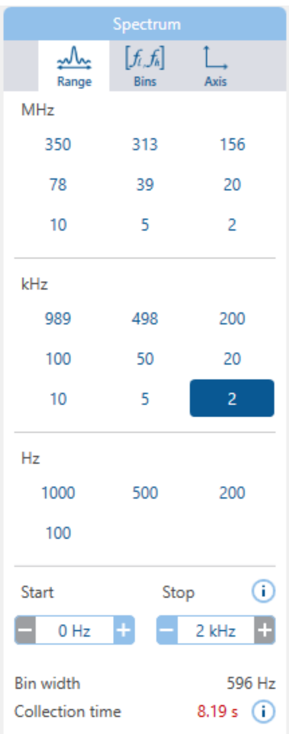
6.4.2 Spectrum view

A Spectrum view is one view of the data from a scope device. A spectrum is a diagram of signal level on a vertical axis plotted against frequency on the horizontal axis. PicoScope opens with a scope view, but you can add a Spectrum view by using the Views option panel. Similar to the screen of a traditional spectrum analyzer, a Spectrum view shows you one or more spectra with a common frequency axis. Each view can have as many spectra as the scope device has channels. Spectrum views are available regardless of which mode - Scope Mode or Spectrum Mode - is active.

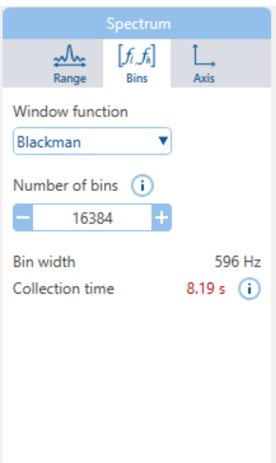


Spectrum controls

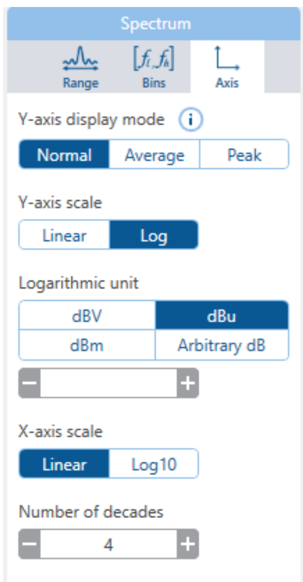
Range



Bins

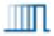


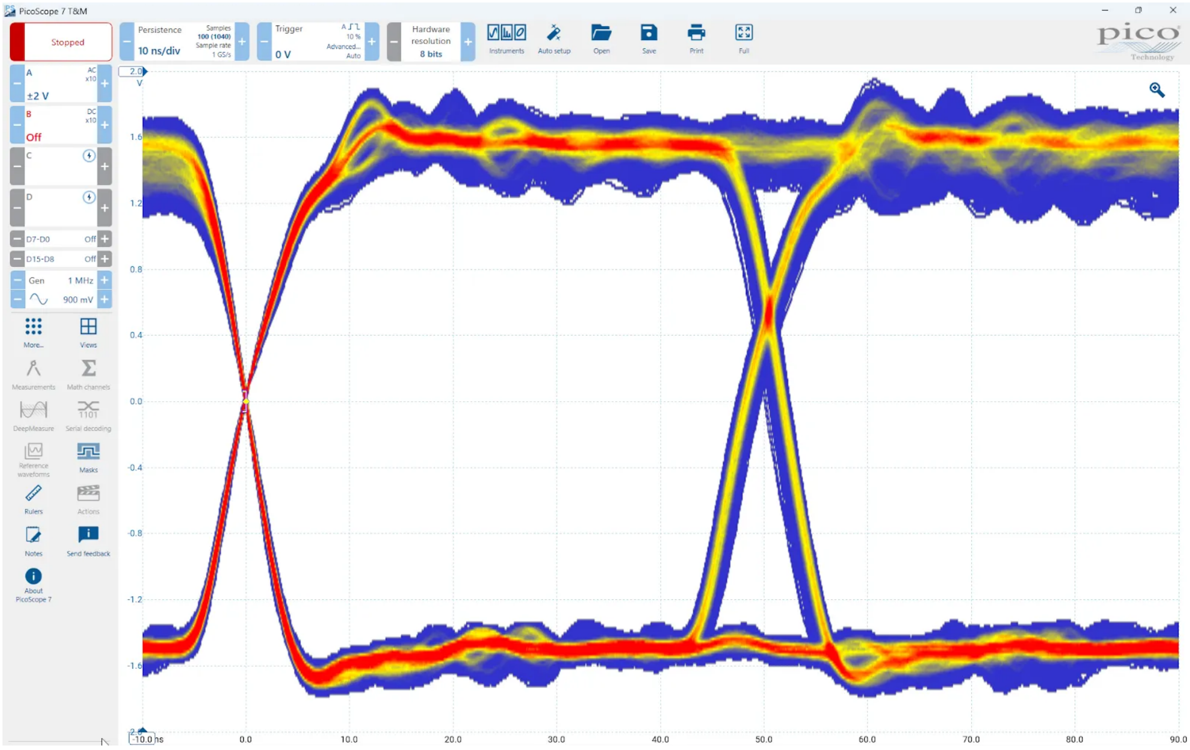
Axis



6.4.3 Persistence view

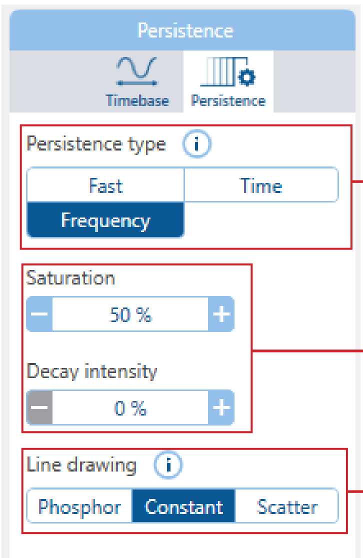
Persistence mode superimposes multiple waveforms on the same view, with more frequent data or newer waveforms drawn in brighter colors than older ones. This is useful for spotting glitches, when you need to see a rare fault event hidden in a series of repeated normal events.

Enable Persistence mode by clicking the Persistence mode button  in the Instruments option panel. With the persistence options set at their default values, the screen will look something like this:



The colors indicate how frequently each point in the waveform occurs. Red is used for the most frequently occurring parts of the signal, with yellow for intermediate and blue for the least frequently occurring. In the example above, the waveform spends most of its time in the red region, but noise causes it to wander occasionally into the blue and yellow regions.

This example shows Persistence mode in its most basic form. See below the Persistence Options panel for ways to modify the display to suit your application.



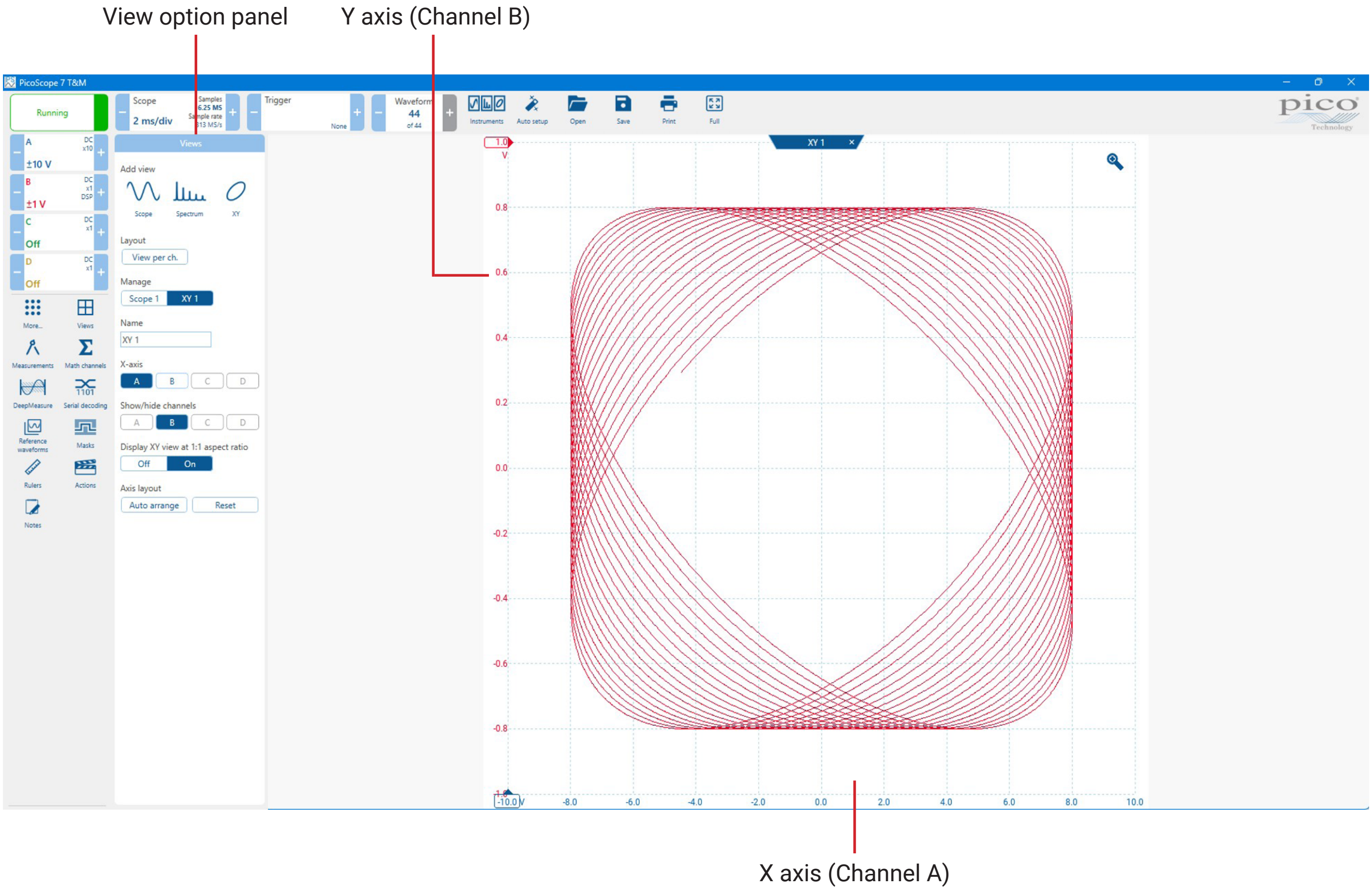
Persistence type:
- Fast - providing a reduced configuration and simplified display options to enable the fastest possible update rate.
- Time - uses color intensity to indicate the age of the waveform data.
- Frequency - uses a range of colors to indicate the frequency of waveform data.

Saturation and Decay intensity allow the user to adjust the visibility of the least frequent waveform data.

Line drawing:
- Phosphor: Joins each pair of sample points with a line whose intensity varies inversely with the slew rate. Emulating a traditional phosphor display.
- Constant: Joins each pair of sample points with a line uniform in color.
- Scatter: Draws sample points as unconnected dots. Providing the user the raw data points.

6.4.4 XY view

An **XY view**, in its simplest form, shows a graph of one channel plotted against another. XY mode is useful for showing relationships between periodic signals (using Lissajous figures) and for plotting I-V (current-voltage) characteristics of electronic components.



There are two ways to create an **XY view**.

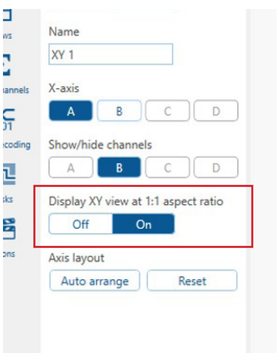
- Use the **Add View > XY** command on the Views options panel. This adds a new **XY view** to the PicoScope window without altering the original scope or spectrum view or views. It automatically chooses the two most suitable channels to place on the X and Y axes. Optionally, you can change the X axis channel assignment using the X-Axis command (see below).
- Select "XY" on the "Instruments" options panel, from the top toolbar. If you currently have a single view, this changes the view to be an XY type. If you've already customized your layout with multiple views, it instead adds an XY view if you don't already have one.

Configuring the X-axis

- Use the **X-Axis** command on the Views option panel. You can only change the X axis if the view is already an XY view. With this method, you can even assign a math channel or a reference waveform to the X axis.

Display XY view at 1:1 aspect ratio

- On (default) - the view is square with the vertical and horizontal axes the same length, even if this doesn't fill the screen area, as shown in the screenshot.
- Off - the view fills the available screen area



6.5 Views

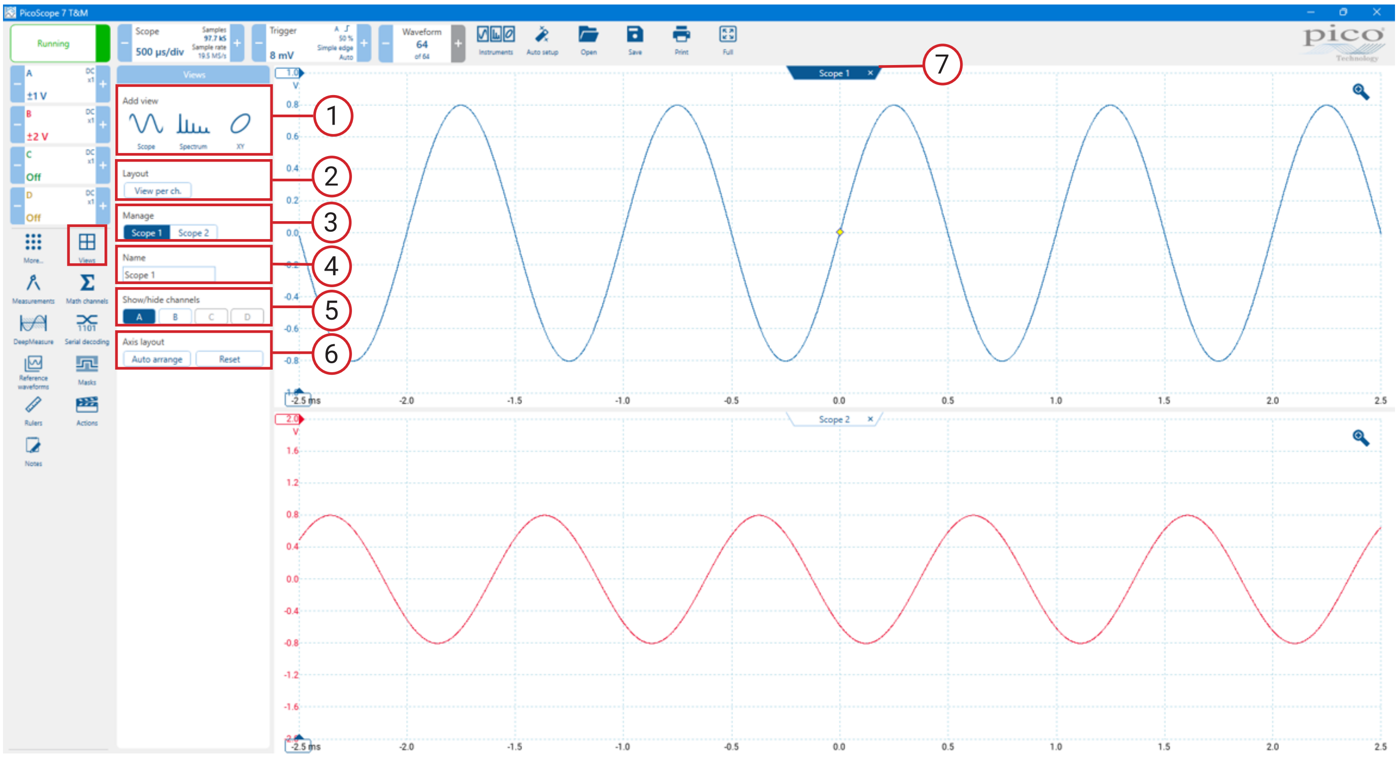
6.5.1 Views option panel

Location: **More...** or Side bar

Purpose: controls the layout of the current view, which is a rectangular area of the PicoScope window that displays scope, spectrum or other kinds of data

To arrange the views within the PicoScope window

If the PicoScope window contains more than one view, PicoScope arranges them in a grid. This is arranged automatically, but you can customize it if you wish. Each rectangular space in the grid is called a viewport. You can move a view to a different viewport by dragging its name tab, in addition to this, you have the ability to move it outside the PicoScope window. You can also place more than one view in a viewport, for more information on this *see How to drag and drop a viewport*



1.	**Add view**	Add a view of the selected type (scope, XY or spectrum).
2.	**Layout**	Clicking **View per ch.** will change the layout to each active channel in its own view e.g. if you're only using two channels then the view will change to show channel A view and channel B view
3.	**Manage**	This allows you to highlight the different views, this will also show views that aren't visible on screen if you have over nine views.
4.	**Name**	Change the standard Scope or Spectrum label to a title of your choice.
5.	**Show/hide channels**	Select which channels are visible in the current view. Each view, when created, shows all the input channels, but you can switch them on and off using this command. Only the input channels that are enabled are available for viewing.
6.	**Axis layout**	**Auto arrange** scales and offsets all traces to fill the view and avoid overlaps. **Reset** resets the scale factor and offset of each trace in the selected view to their default values, with each axis filling the full height of the viewport.
7.	**Close view**	Remove a view from the PicoScope window.

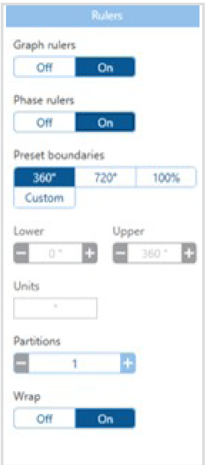
7. Waveform evaluation

7.1 Rulers

7.1.1 Ruler settings

Location: Side bar or **More... > Rulers**

The Ruler options panel allows you to control the behavior of the Channel rulers, Time/Frequency rulers and Phase rulers.



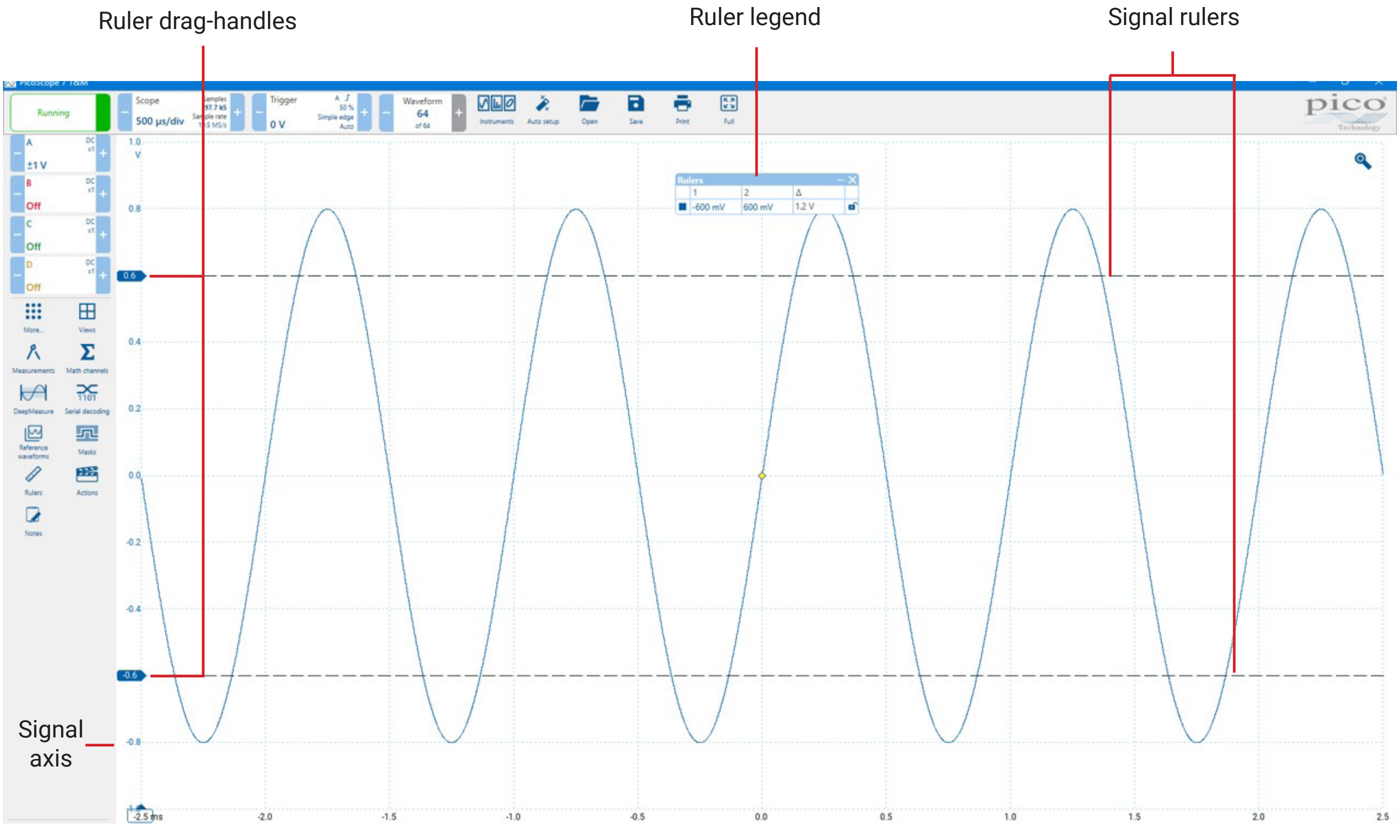
7.1.2 Signal rulers

The signal rulers help you measure absolute and relative signal levels on a scope, XY or spectrum view.

In scope view, the Pointers to the left of the vertical axis are the ruler drag-handles for Channel A. Drag one of these downwards from its resting position in the top left corner, and a signal ruler (a horizontal dashed line) will extend from it.

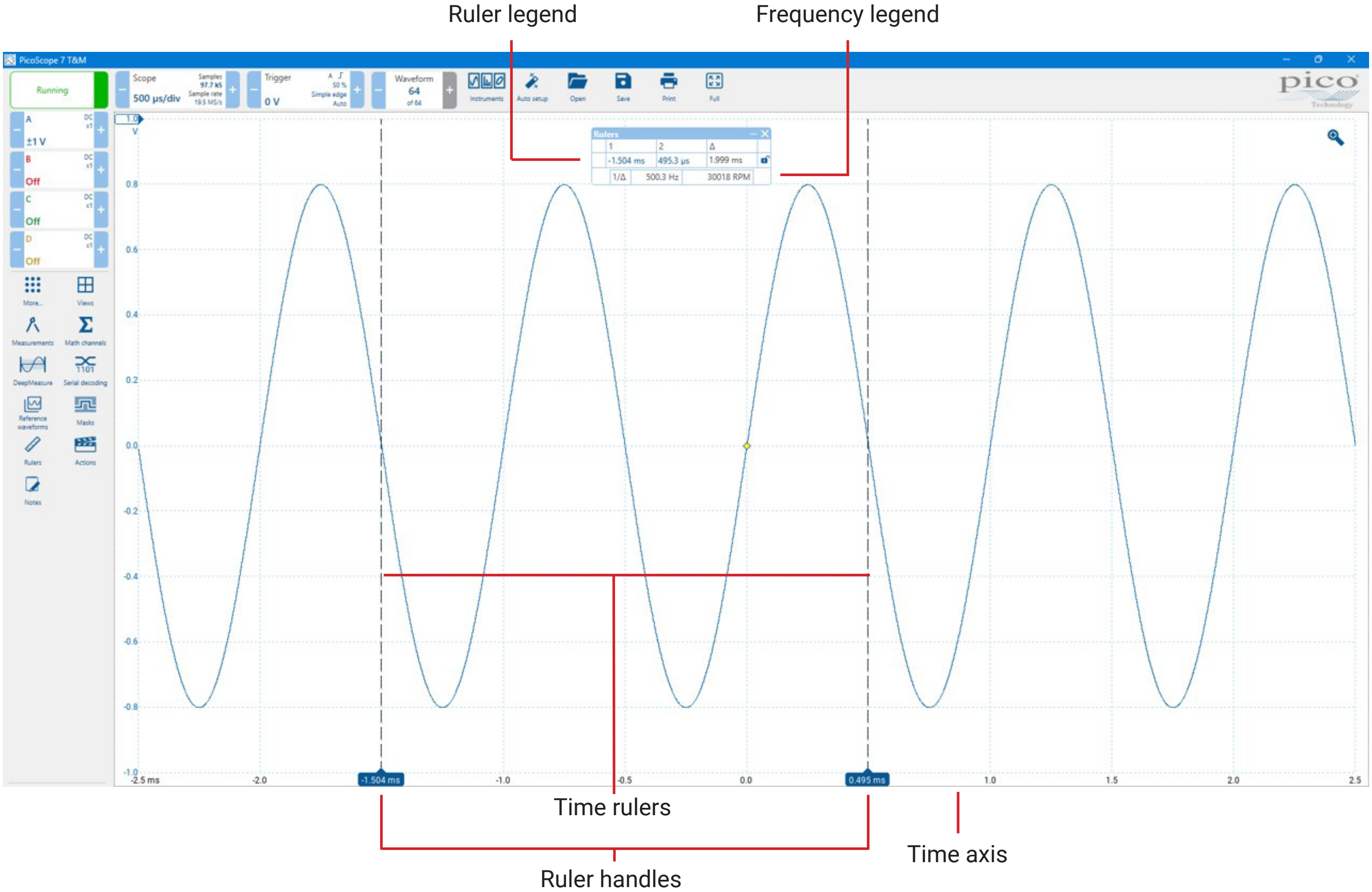
Whenever one or more signal rulers is in use, the ruler legend appears. This is a table showing all of the signal ruler values. If you close the ruler legend using the X button, all the rulers are deleted.

Signal rulers also work in spectrum and XY views.



7.1.3 Time and frequency rulers

The time and frequency rulers are vertical dashed lines that measure time on a scope view and frequency on a spectrum view.



In the scope view above, the two blue rectangles on the time axis are the time ruler handles. When you drag these to the right from the bottom left corner, the time rulers appear. The rulers work in the same way on a spectrum view, but the ruler legend shows their horizontal positions in units of frequency rather than time.

Ruler legend

The table at the top of the view is the Ruler legend. In this example, the table shows that time ruler 1 is at –1.504 milliseconds, ruler 2 is at 495.3 microseconds and that the difference between them is 1.999 milliseconds. Closing the Ruler legend using the button deletes all the rulers.

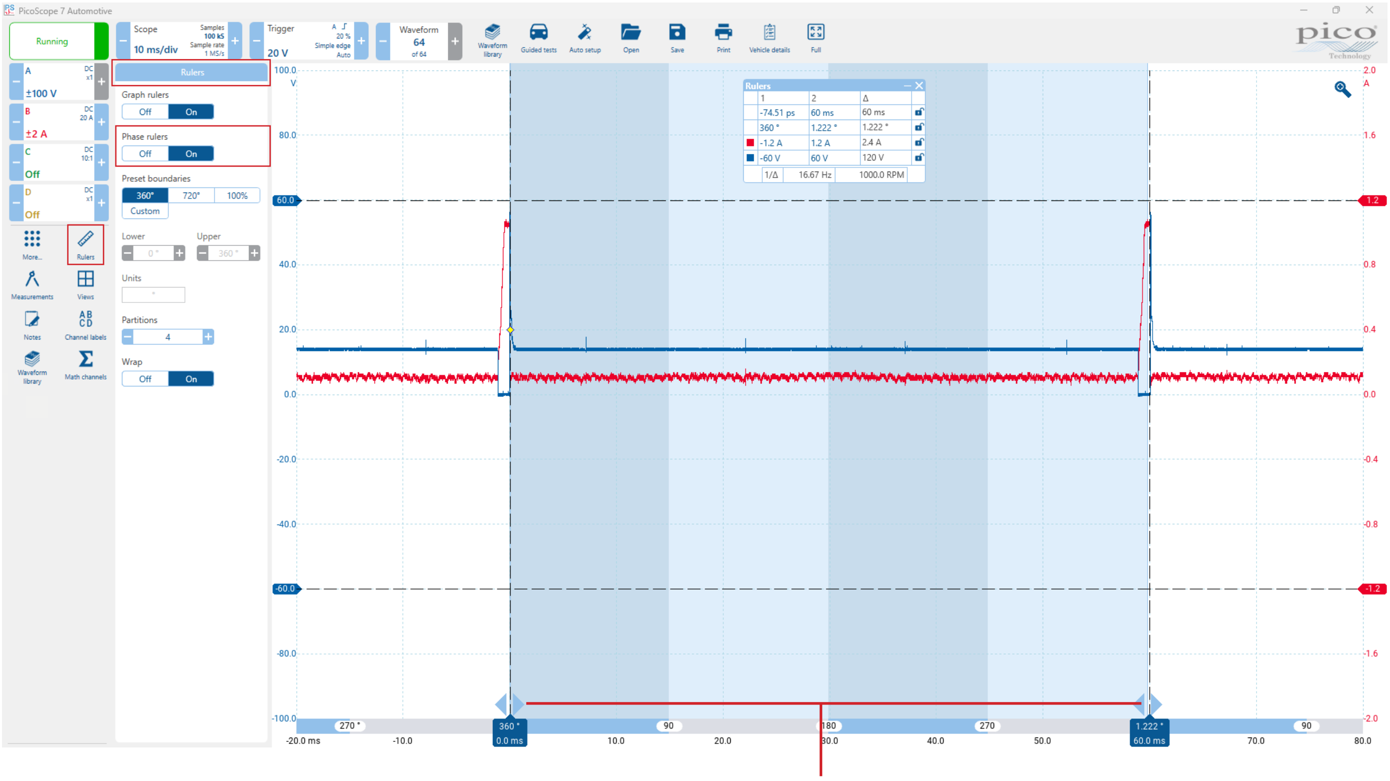
Frequency legend

The Frequency legend in the bottom of the ruler legend shows 1/Δ, where Δ is the difference between the two time rulers. This represents the frequency of a signal where the time rulers are positioned at the start and end of a single cycle. The accuracy of this calculation depends on the accuracy with which you have positioned the rulers. For greater accuracy with periodic signals, use the frequency measurement function built in to PicoScope. The Frequency legend displays values in hertz or revolutions per minute (RPM).

7.1.4 Phase rulers

The Phase rulers help to measure the timing of a cyclic waveform on a Scope view. Instead of measuring relative to the trigger point, as time and Frequency rulers do, phase rulers measure relative to the start and end of a time interval that you specify. Measurements may be shown in degrees, percent or a custom unit as selected in the Ruler settings box. Phase rulers are not available in Spectrum mode.

To use the Phase rulers, select **More...** in the side option panel > **Rulers** > turn Phase rulers **on**, now drag the two phase ruler handles to mark a section of the waveform as shown below:



Drag to position
Phase rulers

Partition

Increasing this value above 1 causes the space between the two Phase rulers to be partitioned equally into the specified number of intervals. The intervals are marked by alternately-shaded regions between the Phase rulers. The lines help you to interpret complex waveforms such as the vacuum pressure of a four-stroke engine with its intake, compression, ignition and exhaust phases, or a commutated AC waveform in a switch mode power supply.

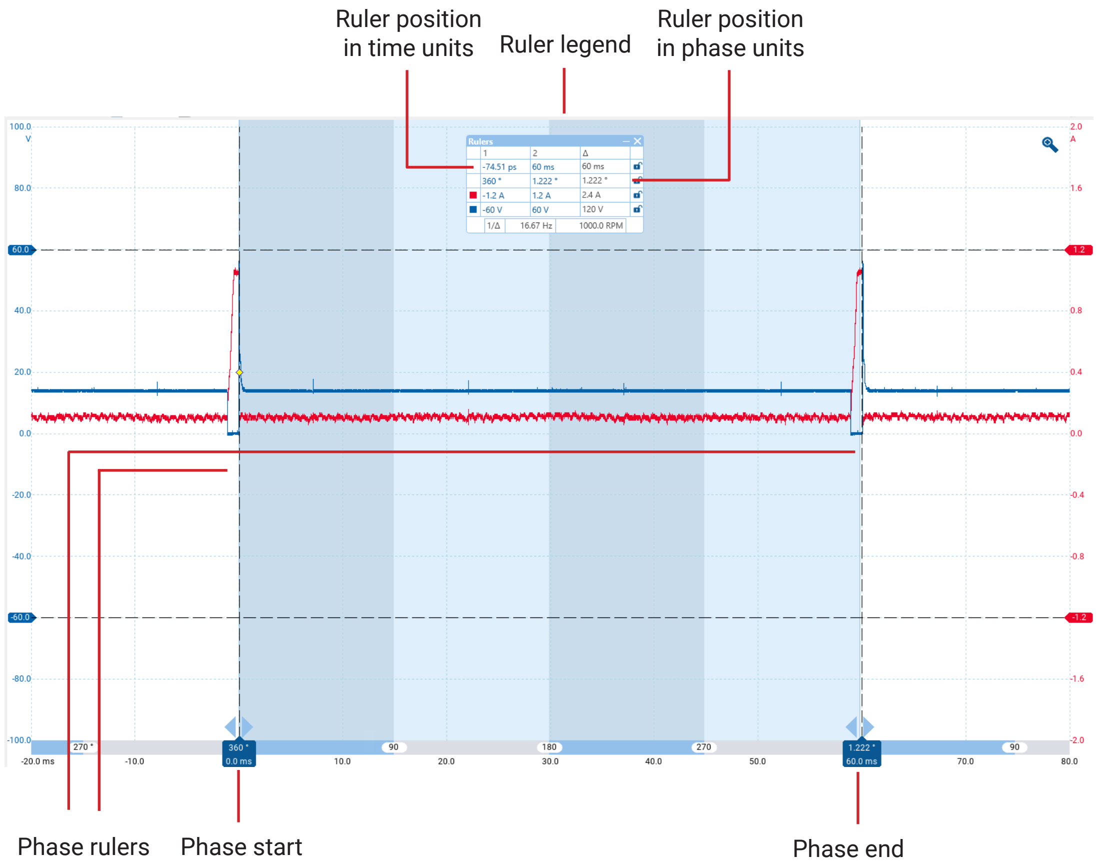
Units

You can choose between Degrees, Percent or Custom. Custom allows you to enter your own unit symbol or name.

Wrap

If Wrap is **on**, Time ruler values outside the range set by the Phase rulers are wrapped back into that range. For example, if the Phase rulers are set to 0° and 360°, the value of a Time ruler just to the right of the 360° Phase ruler will be 0°, and the value of a Time ruler just to the left of the 0° Phase ruler will be 359°. If Wrap is **off**, ruler values are unconstrained, meaning a time ruler just to the right of the 360° phase ruler will show 361°.

When you have dragged both Phase rulers into position, the scope view will look like this (we also added two time rulers, for a reason that we will explain later):



In the Scope view above, the two Phase rulers have been dragged into place to mark the start and end of a cycle.

The start and end phase values of 0° and 360° are shown below the rulers and can be edited to any custom value. For example, when measuring timings on a four-stroke engine, it is customary to show the end phase as 720° as one cycle comprises two rotations of the crankshaft.

Ruler legend

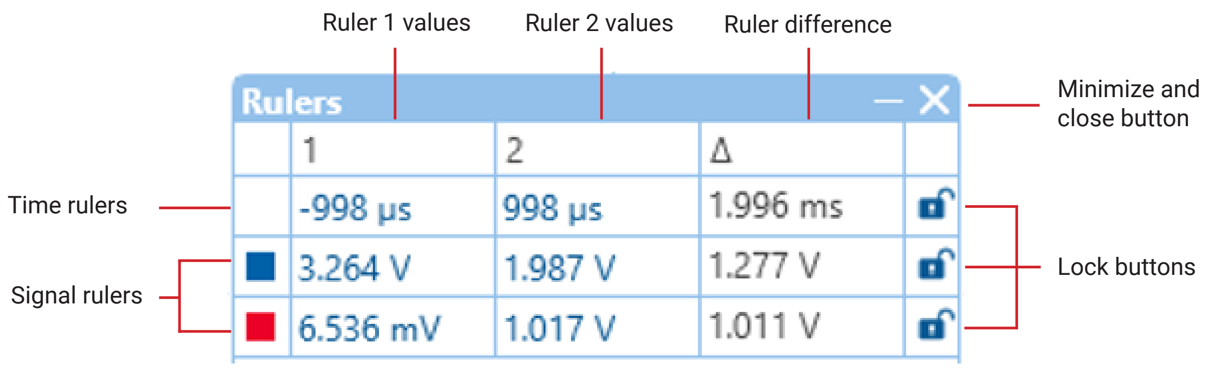
The Phase rulers become more powerful when used in conjunction with Time rulers. When both types of rulers are used together, as shown above, the Ruler legend displays the positions of the Time rulers in phase units as well as time units. If two Time rulers are positioned, the legend also shows the phase difference between them. To close the Rulers they have to be turned off in the option panel.

Ruler settings

Options for the phase (rotation) rulers are configured by the Rulers options panel, which is called up by the Rulers button on the side toolbar.

7.1.5 Ruler legend

The Ruler legend is a box that displays the positions of all the rulers you have placed on the view. It appears automatically whenever you position a ruler on the view:



Editing

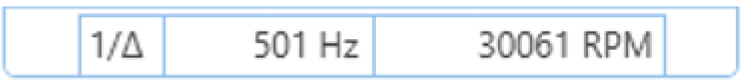
You can adjust the position of a ruler by editing any value in the first two columns. To insert a Greek µ (the micro symbol, meaning one millionth or × 10^{-6}), type the letter u.

Locking rulers

When two rulers have been positioned on one channel, the Lock button appears next to that Ruler in the Ruler legend. Clicking this button causes the two rulers to track each other, dragging one then causes the other to follow it, maintaining a fixed separation. The button changes when the Rulers are locked.

TOP TIP: To set up a locked pair of rulers with a known distance between them, first click the Lock button, then edit the two values in the Ruler legend so that the rulers are the desired distance apart.

7.1.6 Frequency legend



The Frequency legend appears when you have placed two Time rulers on a Scope view. It shows 1/Δ in hertz (the SI unit of frequency, equal to cycles per second), where Δ is the time difference between the two rulers. You can use this to estimate the frequency of a periodic waveform, but you will get more accurate results by creating a frequency measurement using the Measurements button.

For frequencies up to 1.666 kHz, the Frequency legend can also show the frequency in RPM (revolutions per minute).

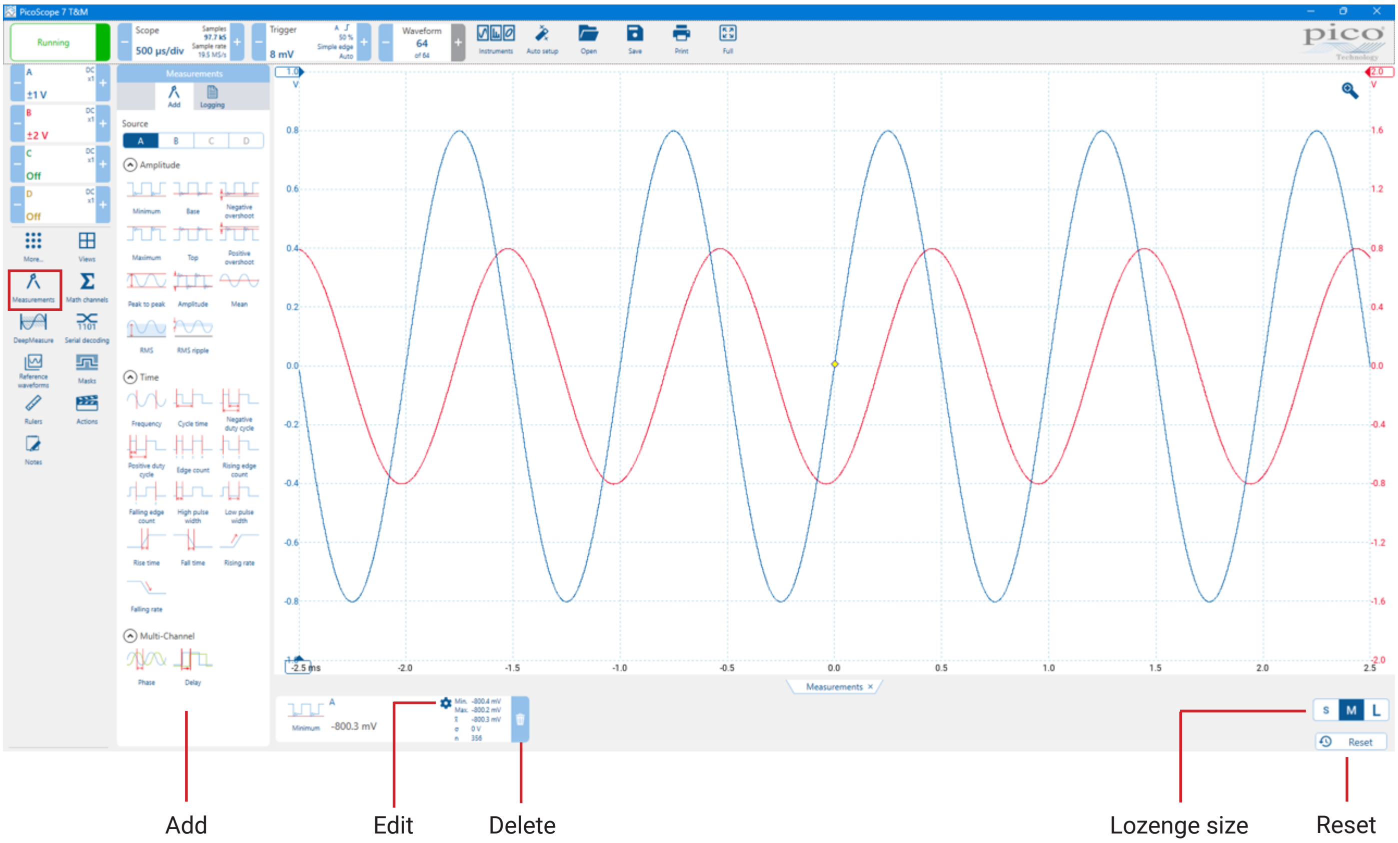
7.2 Measurements

7.2.1 Measurements option panel

Location: Side bar > **Measurements**

More... > Measurements

Purpose: controls the different measurements available including **Amplitude**, **Time**, **Multi-channel** and **Power**.



Add

Clicking a measurement icon adds that measurement on the source channel selected at the top of the panel.

Edit

This takes you to the Edit Measurement dialog, see section ***7.2.2***

Delete

Removes the selected lozenge from the Measurements tab.

Lozenge size

Sets the font and lozenge size and level of detail for the entries in the Measurements tab.

- Small: Only shows the current value
- Medium: Shows all statistical parameters for each measurement
- Large: Shows the current value in a large font.

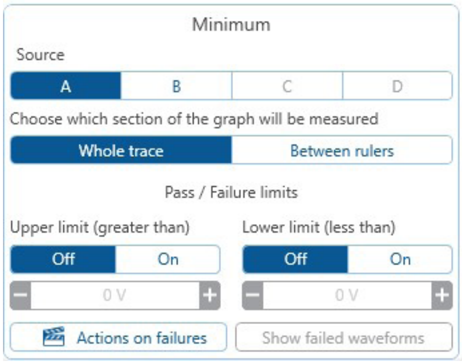
Reset

Resets all the Measurement statistics in the Measurements tab

7.2.2 Edit Measurement dialog

Location: **Measurements** buttons > **Add Measurement**
More... > Add Measurement or click on an existing measurement lozenge

Purpose: allows you to configure the settings for a new or existing measurement.



If this is the first measurement for the active view, PicoScope will create a new measurements tab; otherwise, it will add the new measurement to the existing tab. PicoScope automatically refreshes the measurement every time it updates the waveform.

Source - Allows the user to choose which of the scope device's channels to measure. You can also add measurements on math channels.

Choose which section of the graph will be measured - Measure the whole trace, the section between two time or frequency rulers or a single cycle marked by one ruler. Available options depend on the selected measurement.

Pass / Failure limits - Allows the user to set their own upper and lower limits.

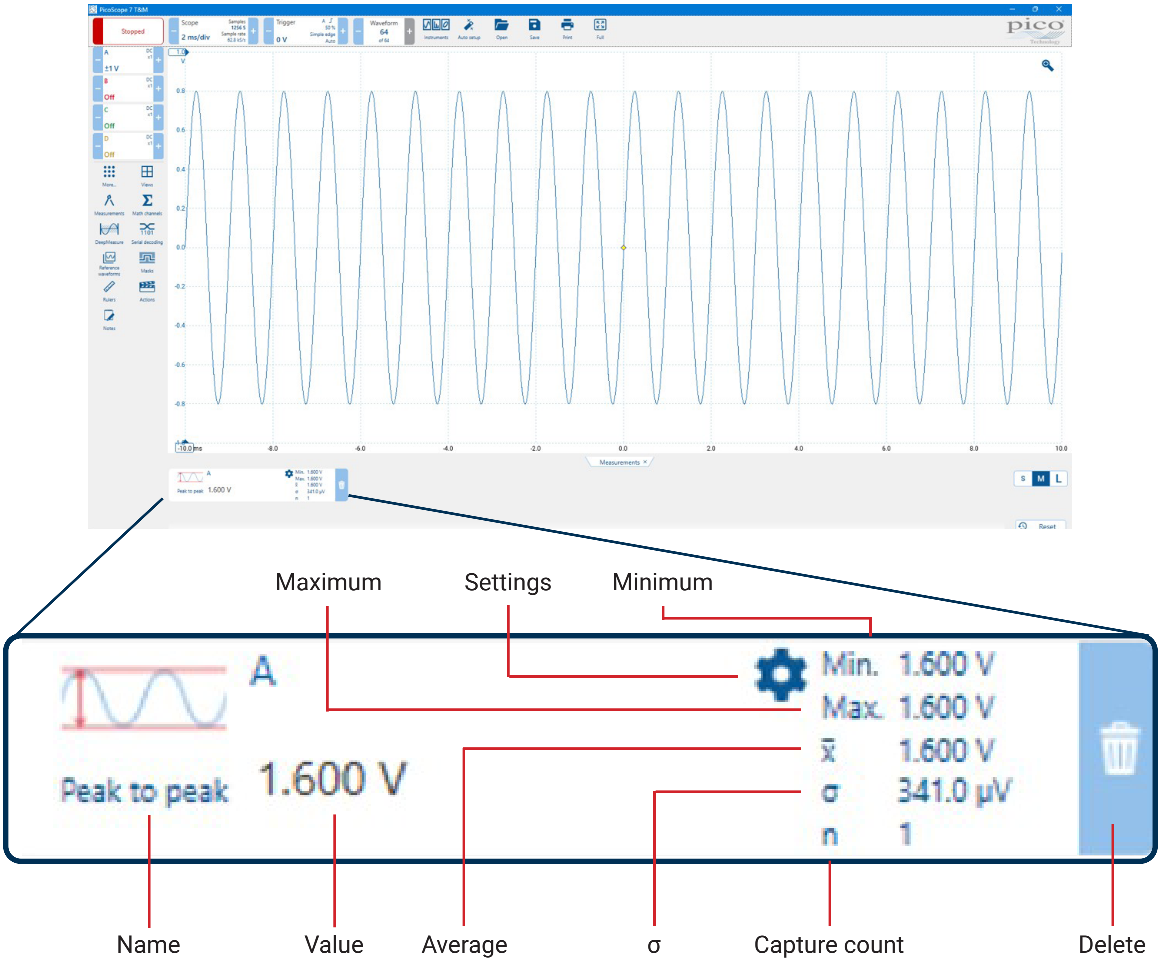
- For an upper limit, specify a threshold value for the failure condition, which will occur whenever a waveform is captured that causes the measurement value to go above the specified value
- For a lower limit, specify a threshold value for the failure condition, which will occur whenever a waveform is captured that causes the measurement value to go below the specified value.

Measurement Options

Some measurements have additional settings to choose from, for more information on the measurements available within the PicoScope software, please click here *https://www.picotech.com/library/knowledge-bases/oscilloscopes/measurements*

7.2.3 Measurement button

Automatic measurements can be added using the Measurements button in the side toolbar options panel. Each measurement is shown as a lozenge in the measurements pane, which can be configured to the size and level of detail you require.



Name	The name of the measurement
Value	The live value of the measurement, from the latest capture
Minimum	The minimum value of the measurement since measuring began
Maximum	The maximum value of the measurement since measuring began
Average	The arithmetic mean of the measurements from the last n captures
σ	The standard deviation of the measurements from the last n captures
Capture count (n)	The number of captures used to create the statistics above
Settings	Allows you to add limits, actions, etc.
Measurement lozenges	Sets the font and lozenge size for the entries in the **Measurements** tab 
Delete	Delete the selected measurement

7.2.4 Spectrum measurements

The measurements available in a **Spectrum view** are as follows.

Note: the term **datum** refers to the noise-free signal.

Frequency at peak. The frequency at which the **peak signal value** appears.

Amplitude at peak. The amplitude of the **peak signal value**.

Average amplitude at peak. The amplitude of the **peak signal value** averaged over a number of captures.

Total power. The power of the whole signal captured in the **Spectrum view**, calculated by adding the powers in all of the **spectrum bins**.

Total harmonic distortion (THD). The ratio of the sum of **harmonic powers** to the power at the **fundamental frequency**.

$$THD = 20\log_{10}\left(\frac{\sqrt{V_2^{\,2} + V_3^{\,2} + V_4^{\,2} + V_5^{\,2} + V_6^{\,2} + V_7^{\,2}}}{V_1}\right)$$

Total harmonic distortion plus noise (THD+N). The ratio of the **harmonic power** plus noise to the **fundamental power**. THD+N values are always greater than the THD values for the same signal.

$$THD + N = 20\log_{10}\left(\frac{\sqrt{\text{sum of squares of RMS values excluding datum}}}{\text{RMS value of datum}}\right)$$

Spurious-free dynamic range (SFDR). This is the ratio of the amplitude of the highest peak in the spectrum to that of the second highest peak. The second peak is not necessarily a harmonic of the **fundamental frequency**. For example, it might be a strong, independent noise signal.

Signal to noise and distortion ratio (SINAD). The ratio, in decibels, of the signal-plus-noise-plus-distortion to noise-plus-distortion.

$$SINAD = 20\log_{10}\left(\frac{\sqrt{\text{sum of squares of all RMS components}}}{\sqrt{\text{sum of squares of all RMS components except datum}}}\right)$$

Signal to noise ratio (SNR). The ratio, in decibels, of the mean signal power to the mean noise power. Hanning or Blackman windows are recommended because of their low noise.

$$SNR = 20\log_{10}\left(\frac{\text{RMS value of datum}}{\sqrt{\text{sum of squares of all values excluding datum and harmonics}}}\right)$$

Intermodulation distortion (IMD). A measure of the distortion caused by the nonlinear mixing of two tones. When multiple signals are injected into a device, modulation or nonlinear mixing of these two signals can occur. For input signals at frequencies f1 and f2, the two second-order distortion signals will be found at frequencies: f3 = (f1 + f2) and f4 = (f1 - f2).

IMD is expressed as the dB ratio of the RMS sum of the distortion terms to the RMS sum of the two input tones. IMD can be measured for distortion terms of any order, but the second-order terms are most commonly used. In the second-order case, the intermodulation distortion is given by:

$$IMD = 20\log_{10}\sqrt{\frac{F_3^{\,2} + F_4^{\,2}}{F_1^{\,2} + F_2^{\,2}}}$$

where F3 and F4 are the amplitudes of the two second-order distortion terms (at frequencies f3 and f4 defined above) and F1 and F2 are the amplitudes of the input tones (at frequencies f1 and f2, as marked by the frequency rulers in the spectrum window).

For reference, the third-order terms are at frequencies (2f1 + f2), (2f1 – f2), (f1 + 2f2) and (f1 – 2f2).

Note: Hanning or Blackman windows are recommended because of their low noise. An FFT size of 4096 or greater is recommended in order to provide adequate spectral resolution for the IMD measurements.

Mask Failures. A special measurement that counts the number of failed waveforms during **mask limit testing**. This measurement is added to the table automatically when you use **mask limit testing**, so there is usually no need to select it manually.

7.3 DeepMeasure™

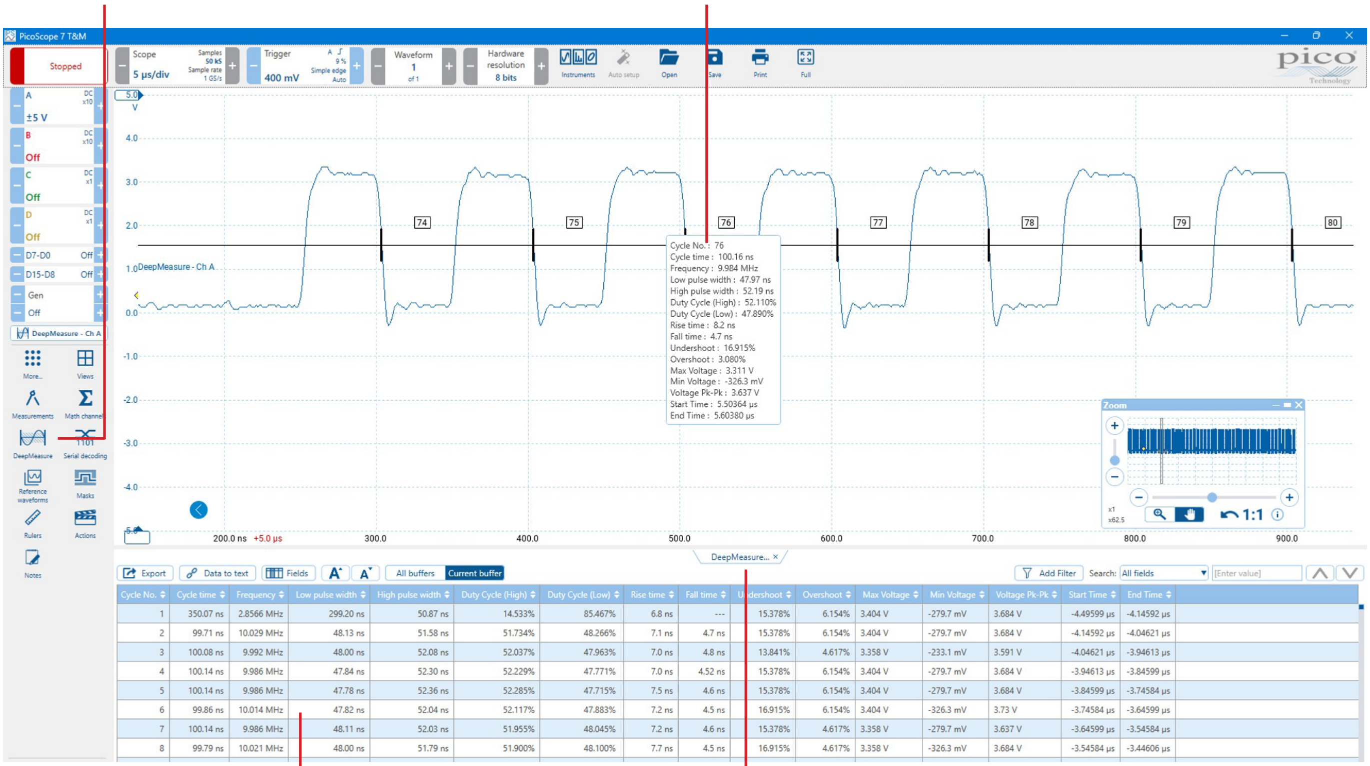
The measurement of waveform pulses and cycles is key to verifying the performance of electrical and electronic devices.

DeepMeasure delivers automatic measurements of important waveform parameters, such as pulse width, rise time and voltage, for every individual cycle in the captured waveforms. Up to a million cycles can be displayed with each triggered acquisition or combined across multiple acquisitions. Results can be easily sorted, analyzed and correlated with the waveform display, or exported as a .CSV file or spreadsheet for further analysis.

For example, use DeepMeasure with PicoScope's rapid trigger mode to capture 40 000 pulses and quickly find those with the largest or smallest amplitude, or use your scope's deep memory to record a million cycles of one waveform and export the rise time of every single edge for statistical analysis.

Select DeepMeasure

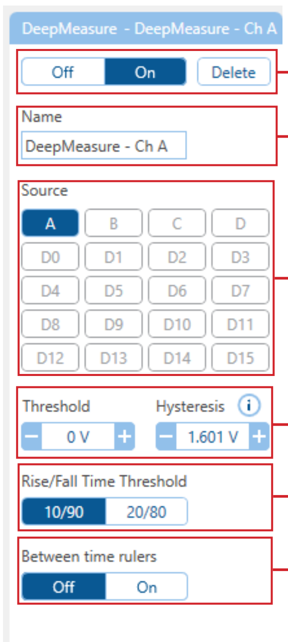
Hover over waveform to view summary



DeepMeasure data table

DeepMeasure tab

To start using DeepMeasure, click **DeepMeasure** from the side bar or click **More... > DeepMeasure**. You can simultaneously run as many instances of DeepMeasure as you like, with different settings or on different channels, and switch between them using the **DeepMeasure tabs**.



Switch DeepMeasure On or Off

Personalization

Select the channel

Adjust the Threshold and Hysteresis used to determine the boundary between cycles of your waveform.

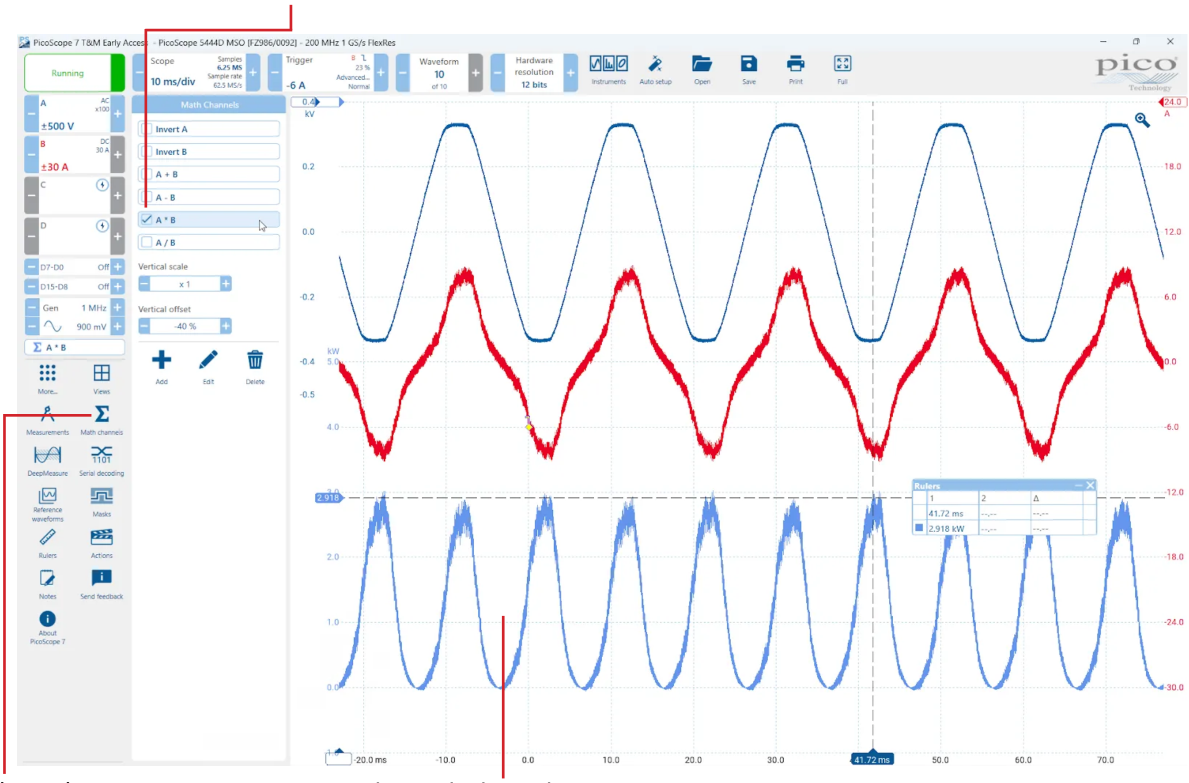
Adjust the Rise/Fall time Threshold

Select whether DeepMeasure operates on the whole trace (off) or just the portion of the trace between the two time rulers (on).

7.4 Math Channels

A Math channel is a mathematical function of one or more input signals. It can be displayed in a Scope, XY or Spectrum view in the same way as an input signal, and like an input signal it has its own measurement axis, Scaling and Offset controls and color. PicoScope has a set of built-in math functions for the most important functions, such as Invert A, A+B and A–B. You can also define your own functions using the equation editor.

Here is a three-step guide to using math channels:

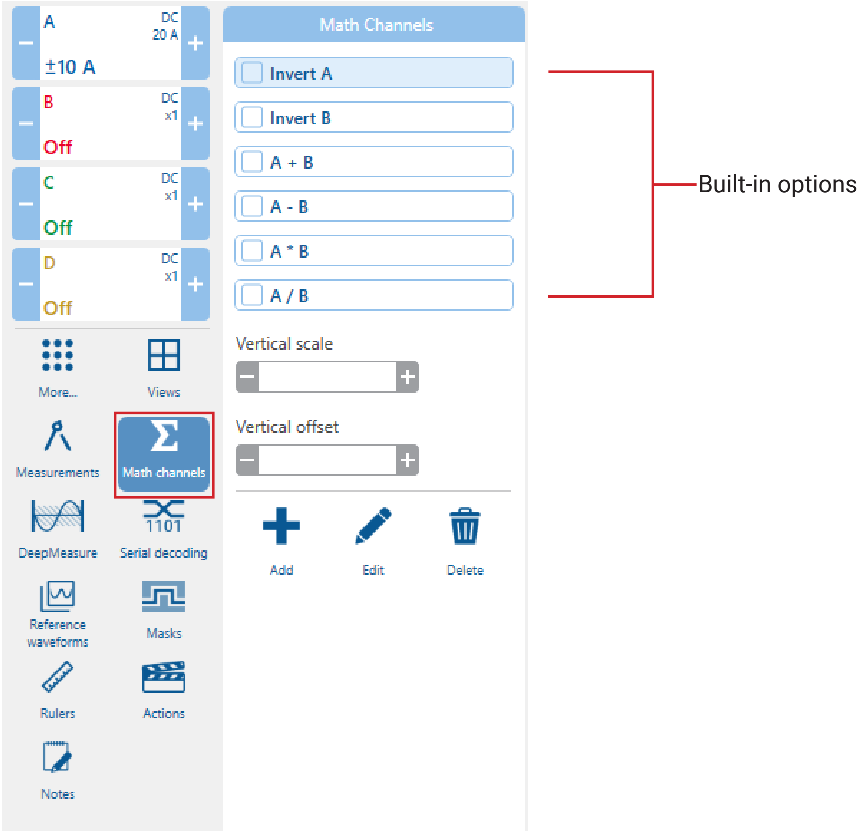


1. Math Channels command in the side bar. Click this to open the Math Channels option panel, shown in the image above.
2. Math Channels option panel. This lists all the available math channels and allows you define new ones. In the example above, only the built-in functions are listed.
3. Math channel. Once enabled, a math channel appears in the selected Scope or Spectrum view. You can change its scale and offset as with any other channel. In the example above, the new math channel (bottom) is defined as A-B, the difference between input channels A (top) and B (middle).

7.4.1 Math Channels

Location: **Tools > Math Channels**

Purpose: creating, editing and controlling math channels, which are virtual channels generated by mathematical functions of input channels



Math Channels

The main area of the Math Channels options panel is the Math Channels list, which shows all the built-in Math Channels options. To choose whether or not a channel appears in the main PicoScope window, click the appropriate check box. You can have up to eight channels in any view. If you enable a ninth channel, PicoScope opens a new view.

Add guides you through the process of creating or editing a math channel. The new channel will appear in the Math Channels list.

Edit Allows you to edit the selected math channel. You must first select a channel in the Math Channels list and then click **Edit**.

Delete Permanently deletes the selected math channel.

7.4.2 Math Channel Equation

Location: **Math Channel > Add** or **select an existing math channel > Edit**

Purpose: allows you to enter or edit the equation for a math channel

You can type directly into the equation box, or click the calculator buttons and let the program insert the symbols for you. When you are happy with your equation (Fig.1), click **Next** to continue to the Math Channel wizard Name dialog (Fig.2).

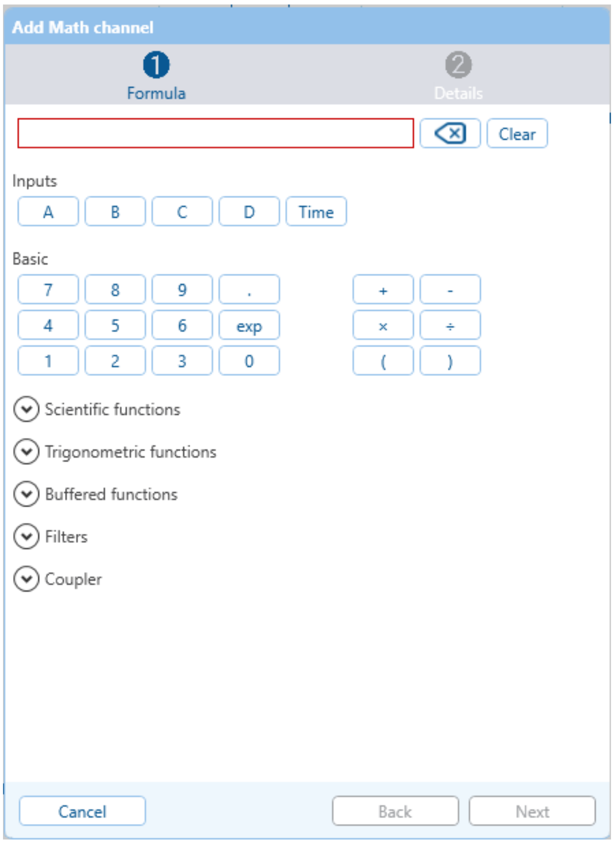


Fig.1

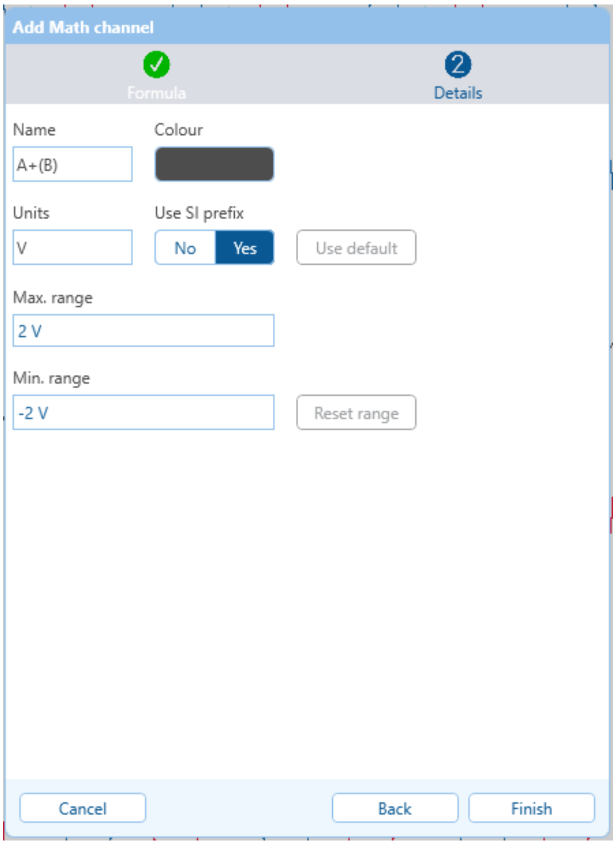
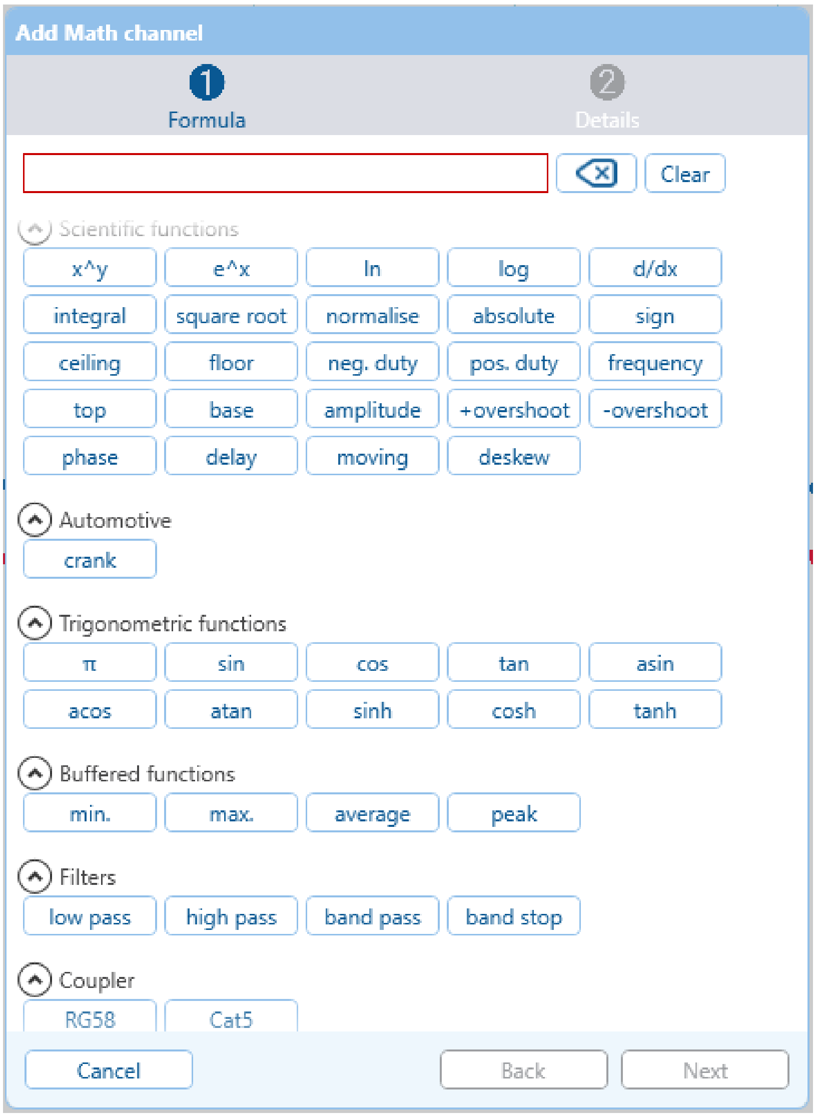


Fig.2

The available math functions are grouped into sections for easy access. Each section has an arrow button to show or hide its functions.



Advanced buttons (Scientific functions)

Button	Equation	Description
exp	E	**Exponent.** Only applicable to constants. 1E6 is shorthand for 1x10^6 or 1 million.
x^y	^	**Power.** Raise x to the power of y.
e^x	exp()	**Natural exponent**. Raise e, the base of the natural logarithm, to power of x.
ln	ln()	**Natural logarithm**
log	log()	**Logarithm**. Base-10 logarithm
d/dx	derivative()	**Derivative**. Calculated with respect to the x-axis.
Note: the derivative of a sampled signal contains a large amount of noise, so it is advisable to apply digital lowpass filtering to all channels used as inputs to this function.		
integral	Integral()	Along the x-axis.
square root	sqrt()	**Square root**

Advanced buttons cont. (Scientific functions)

Button	Equation	Description
normalize	norm()	PicoScope calculates the maximum and minimum values of the argument over the capture period, and then scales and offsets the argument so that it exactly fits the range [0, +1] units.
absolute	abs()	**Absolute value**
sign	sign()	The function simply gives the sign for the given values of x, -1 if negative, 0 if exactly zero, +1 if positive.
ceiling	ceil()	A mathematical function that returns the smallest integer that is greater than or equal to a given real number
floor	floor()	Returns the largest integer less than or equal to the operand
-duty	negduty()	Cycle-by-cycle measurement of negative duty cycle
+duty	posduty()	Cycle-by-cycle measurement of positive duty cycle
frequency	freq()	The number of waves produced by a source each second in hertz (Hz).
cycle time	cycletime()	The period of each cycle of the input signal in seconds
+ pulse width	highpulsewidth()	The width in seconds of the high part of each input waveform cycle
- pulse width	lowpulsewidth()	The width in seconds of the low part of each input waveform cycle
top	top()	Measures the high levels of a signal, excluding any noise or ringing present on those levels. This differs from the Maximum which does include any noise or ringing on those levels.
base	base()	Measures the low levels of a signal, excluding any noise or ringing present on those levels. This differs from the Minimum which does include any noise or ringing on those levels.
amplitude	amplitude()	is a measurement of the maximum distance a waveform is from its equilibrium position. It can be used to visualize how the amplitude of an electrical signal changes over time.
rms	rms()	is a mathematical way to find the DC equivalent voltage of an AC waveform. This is a critical step in completing power calculations because using a simple average or a peak value would not properly calculate the effective value in an AC circuit.
rms ripple	rmsripple()	The RMS value of each cycle of the input signal after first subtracting the DC component (mean average).
phase	phase(<signal 1>, <signal 2>)	Cycle-by-cycle measurement of the phase difference between signal 1 and signal 2.
delay	delay(<signal 1>, <signal 2>)	Delay is the measure of time between equivalent reference points on 2 waveforms (usually rising or falling edges).
moving	moving(<signal>, <resolution (power of 2)>)	Moving-average filter of <signal> over <resolution> samples. Resolution should be a power of two, or will be rounded to the nearest power of two if not.
deskew	[-<time offset (s)>]	Shifts the signal in time, for example to remove the time lag between probes on an oscilloscope. This is important for accurate power measurements, especially when using separate voltage and current probes.
true power	truepower(<volts>, <amperes>)	Power converted to useful output, delivered to a load. True power is also referred to as the following: real, active, working, actual, useful, in-phase.
apparent power	apparentpower(<volts>, <amperes>)	the total electrical power present in an AC circuit or system.

Advanced buttons cont. (Scientific functions)

Button	Equation	Description
reactive power	reactivepower(<volts>, <amperes>)	In an AC power transmission system, reactive power is the component of power that oscillates between the source and load without being consumed
pwr factor	powerfactor(<volts>, <amperes>)	Power factor is the ratio between true and apparent power, which is a measure of an electrical network's operational efficiency.
area AC	areaac()	The area under the curve of a waveform after removing any DC component. The math channel value at each point is the total area from the start of the graph to that point, with sections above the DC average counting positively and those below it counting negatively.
+area AC	areaacpositive()	The area under the positive parts of a waveform after removing any DC component. The math channel value at each point is the total area from the start of the graph to that point, with only sections above the DC average being counted.
-area AC	areaacnegative()	The area above the negative parts of a waveform after removing any DC component. The math channel value at each point is the total area from the start of the graph to that point, with only sections under the DC average being counted.
abs area AC	areaacabs()	The absolute area under the curve of a waveform after removing any DC component. The math channel value at each point is the total area from the start of the graph to that point, with each segment of the area counted positively whether above the DC average or below.
area DC	areadc()	The area between the curve of a waveform and zero. The math channel value at each point is the total area from the start of the graph to that point, with sections above zero counting positively and those below it counting negatively.
+area DC	areadcpositive()	The area under the positive parts of a waveform only. The math channel value at each point is the total area from the start of the graph to that point, with only sections above zero being counted.
-area DC	areadcnegative()	The area above the negative parts of a waveform. The math channel value at each point is the total area from the start of the graph to that point, with only sections below zero being counted.
abs area DC	areadcabs()	The absolute area between the curve of a waveform and zero. The math channel value at each point is the total area from the start of the graph to that point, with each segment of the area counted positively whether above zero or below.
crest factor	crestfactor()	The crest factor of a waveform is the ratio of the peak value to the RMS value of the waveform.
DC power	<volts>*<amperes>	This function is simply a shortcut for multiplying the channels representing voltage and current to give power.
+overshoot	posovershoot()	The Positive overshoot measurement measures the difference in the Maximum and the Top as a percentage of the Amplitude and so the calculation is (Maximum – Top) / (Top – Base) x 100.
-overshoot	negovershoot()	The Negative overshoot measurement measures the difference in the Base and the Minimum as a percentage of the Amplitude and so the calculation is (Base – Minimum) / (Top – Base) x 100.

Advanced buttons (Trigonometric functions)

Button	Equation	Description
π	pi	The mathematical constant pi
sin	sin()	**Sine.** The operand is in radians.
cos	cos()	**Cosine**. The operand is in radians.
tan	tan()	**Tangent**. The operand is in radians.
sinh	sinh()	**Hyperbolic sine**.
cosh	cosh()	**Hyperbolic cosine**.

Advanced buttons cont. (Trigonometric functions)

Button	Equation	Description
tanh	tanh()	**Hyperbolic tangent**.
asin	asin()	**arcsine** returns the inverse sine of the number, in radians between $-\pi / 2 <= r <= \pi / 2$
acos	acos()	**arccos** returns the inverse cosine of the number, in radians between $0 <= r <= \pi$
atan	atan()	**arctangent** returns the inverse tangent of the number, in radians between $-\pi / 2 <= r <= \pi / 2$

Advanced buttons (Buffered function)

Button	Equation	Description
min	min()	**Minimum**. Negative peak detect of all previous waveforms.
max	max()	**Maximum**. Positive peak detect of all previous waveforms.
average	average()	**Average**. Arithmetic mean of all previous waveforms
peak	peak()	**Peak detect**. Display maximum-to-minimum range of all previous waveforms.

Advanced buttons (Filters)

Parameters:

i is the input channel or other operand (see under **Basic buttons** above)
f (or $\mathbf{f}_1$ and $\mathbf{f}_2$) are the –3 dB cutoff frequencies of the filter, in hertz

Button	Equation	Description
low pass	lowPass(i,f)	**lowpass filter**. Attenuates high frequencies.
high pass	highPass(i,f)	**Highpass filter**. Attenuates low frequencies.
band pass	BandPass(i,f1,f2)	**Bandpass filter**. Attenuates high and low frequencies outside the specified range.
band stop	BandStop(i,f_1,f_2)	**Bandstop filter**. Attenuates mid-band frequencies inside the specified range.

For more information on how to set up a specific Math channel, *click here.*
Advanced buttons (Couplers)

Parameters:

$\mathbf{s}_1$ is the master channel
$\mathbf{s}_2$ is the slave channel
d is the distance between the master and slave probing points, in metres

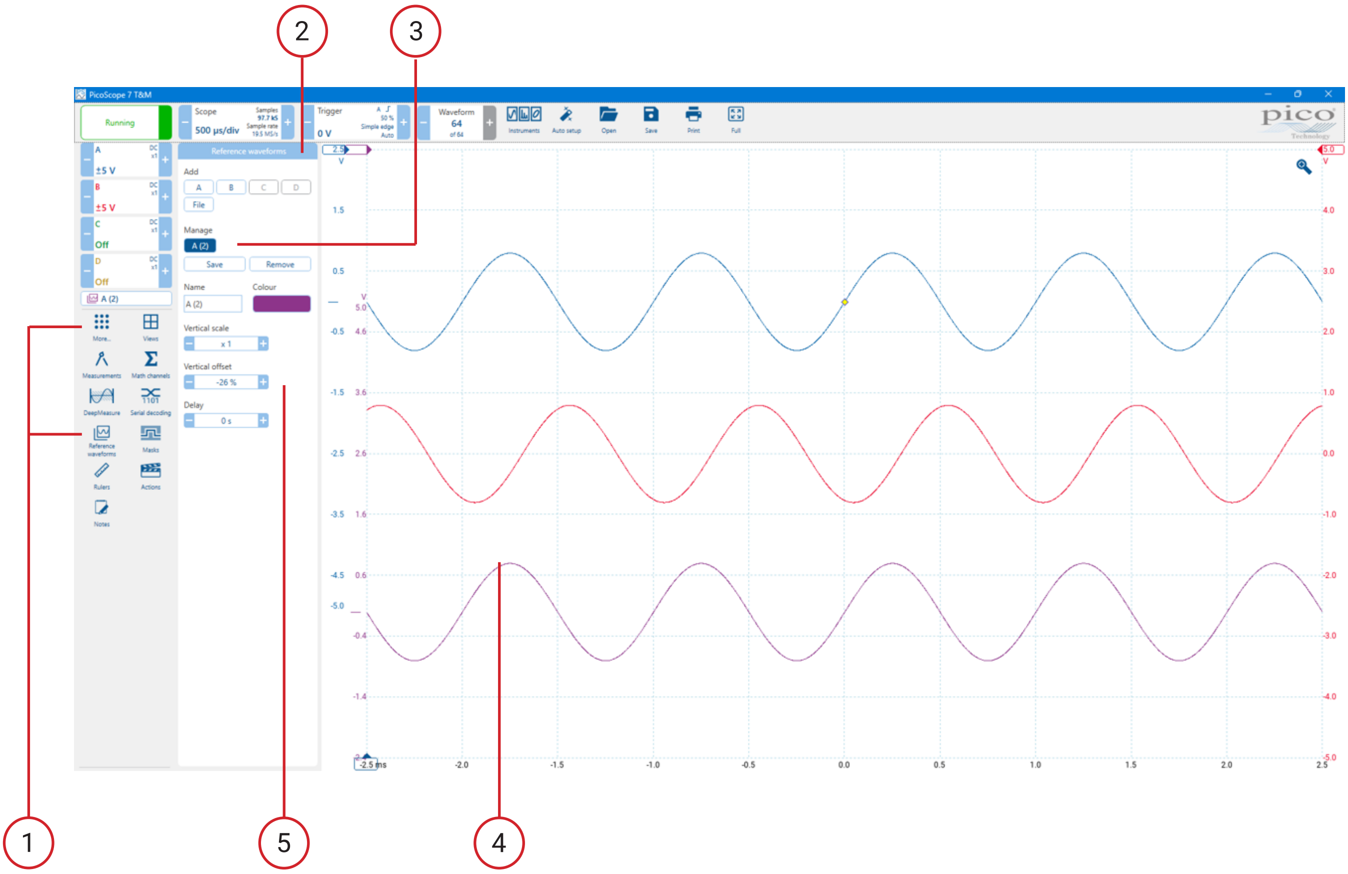
Button	Equation	Description
RG58	Coupler(s1,s2,RG58,d)	**RG58 coupler**. For coaxial cables.
Cat5	Coupler(s1,s2,Cat5,d)	**Cat5 coupler**. For twisted-pair cables.

These are directional coupler functions for extracting **BroadR-Reach** signals from a bidirectional bus. One differential probe (connected to channel s1) is placed on the cable near the master device, and another (connected to channel s2) on the cable near the slave device. The output of the coupler is then used as the input to the **BroadR-Reach** protocol decoder.

7.5 Reference waveforms

A Reference waveform is a stored version of an input signal. You can create one by selecting **More...** > **Reference waveforms**. It can be displayed in a Scope or Spectrum view in the same way as an input signal, and similarly it has its own Measurement axis, scaling, offset and delay controls, and color. The Reference waveform may have fewer samples than the original.

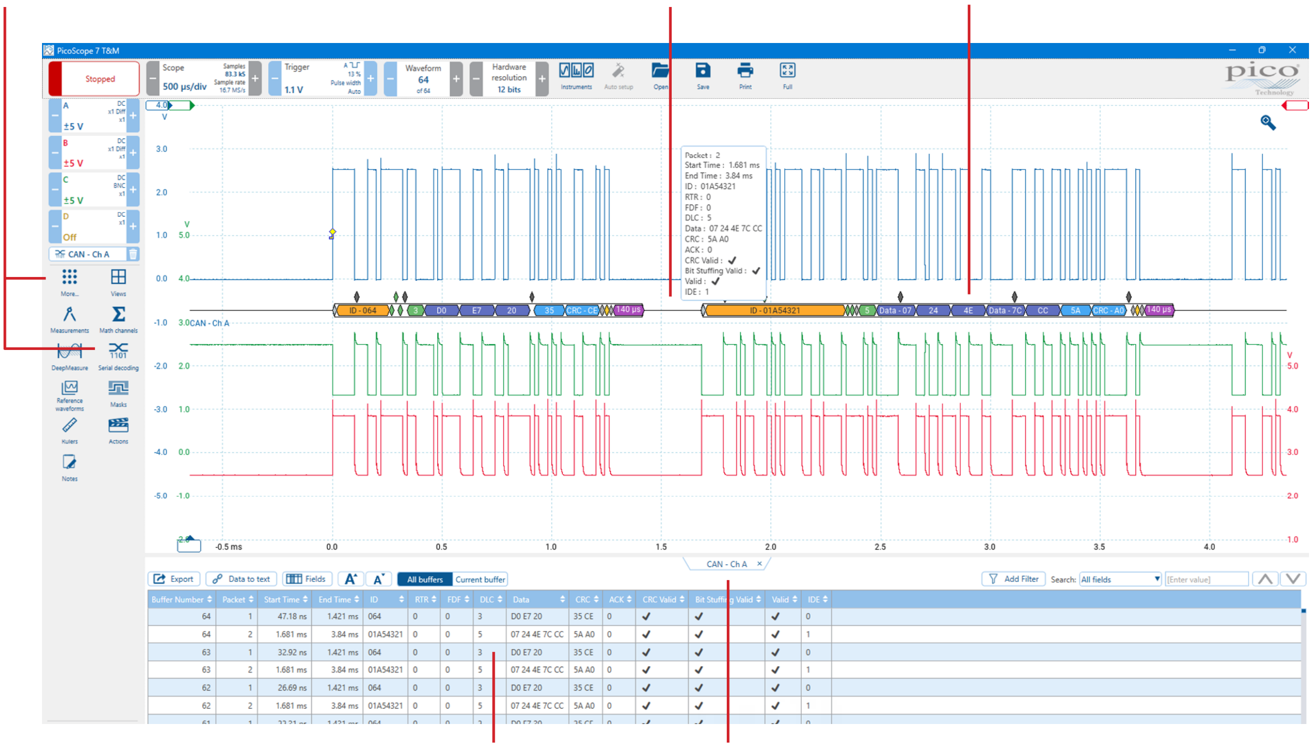
For more control over Reference waveforms, use the Reference Waveforms options panel as shown below.



1. **More... > Reference waveforms** command or use the side bar. Click this to open the Reference Waveforms options panel, shown in the image above.
2. Reference Waveforms options panel. This lists all the available input channels and Reference waveforms. In the example above, input channels A and B are switched on, so they appear in the **Add** section.
3. **Manage** section. This shows all your Reference waveforms. You have the option to save or remove the Reference waveform here too.
4. Reference waveform. Once enabled, a Reference waveform appears in the selected scope or spectrum view. In the example above, the new Reference waveform (bottom) is a copy of Channel A.
5. Axis control. Allowing you to adjust the scale, offset and delay for this waveform, and above that you can rename and change the color of the new Reference waveform.

7.6 Serial decoding

You can use PicoScope to decode data from a serial bus such as I2C or CAN Bus. Unlike a conventional bus analyzer, PicoScope lets you see the high-resolution electrical waveform at the same time as the data. The data is integrated into the Scope view, with color-coded packets, so there's no need to learn a new screen layout.



To start decoding your data, click **More... > Serial Decoding** or use the side bar. There are many different protocols available, and the data can be viewed in Graph or Table formats, or both. You can simultaneously decode multiple channels in different formats, and switch between them using the Decoding tabs.

Tech tip - Link files: Link files help to speed analysis by cross referencing hexadecimal field values into human readable form. So, for example, instead of displaying "Address: 7E" in the Table View, the corresponding text "Set Motor Speed" will be shown instead, or whatever is appropriate. The Link File template with all field headings can be created directly from the serial table toolbar, and edited manually as a spreadsheet to apply the cross reference values.

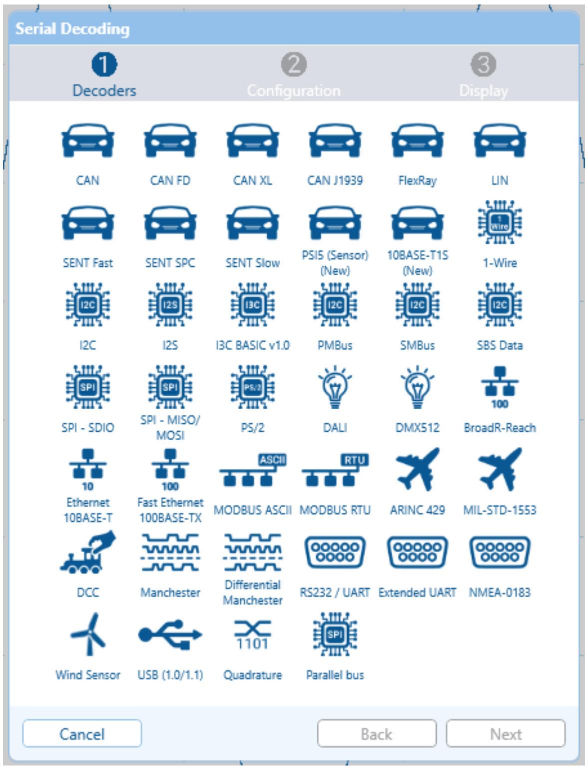
T19

	G	H	I	J	K	L	M	N	O	P
1	ID	ID Description	RTR	RTR Description	FDF	FDF Description	DLC	DLC Description	Data	Data Description
2	540	Key Position							C0 49 FF 00 FF 00 00 0F	Standby
3	6D1	Cycle Start								
4	288	Crank Speed								
5	320	Oil Temp								
6	588	Water Temp								
7	3E2	MAF								
8	1A0	Indicator LHS								
9	5A0	Indicator RHS								
10	280	Throttle Position								
11	5E7	Fuel Pressure								

7.6.1 Serial decoding dialog

Location: **More...** or Side bar > **Serial Decoding**

Purpose: lets you set up which channels and protocols to use for **serial decoding.**



Setting up a new serial decoder

1. Click **Serial decoding** in the side bar

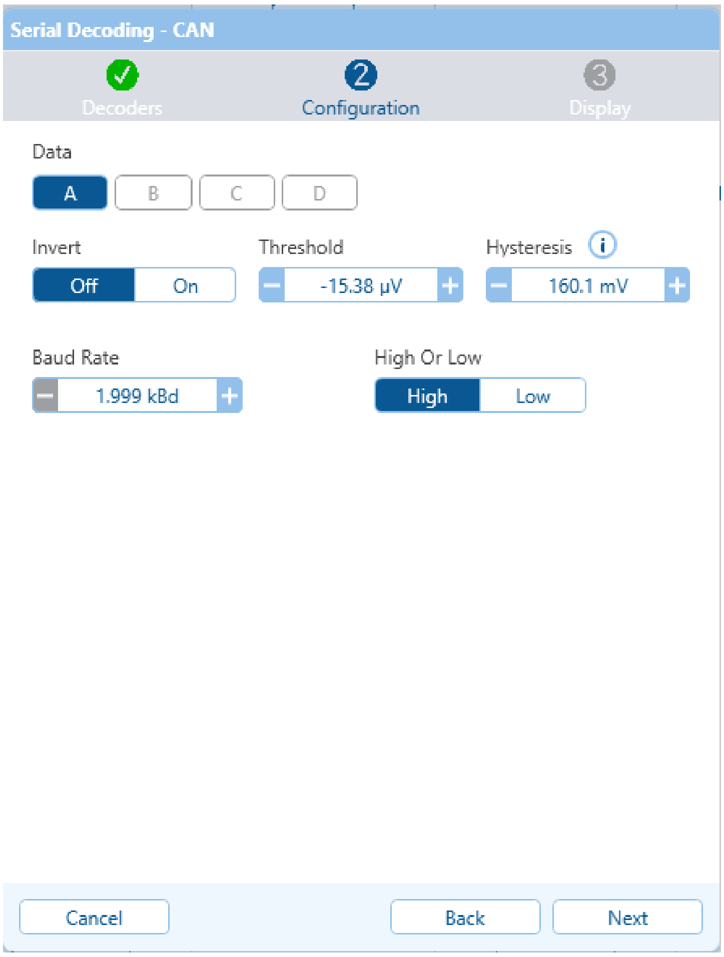
2. Select the **serial decoding** protocol you would like to use from the list.

3. Now follow the wizard regarding your required configuration and display.

For more information on setting up specific serial decoders, *see how to setup serial decoders.*

7.6.2 Decoder settings

Different settings options are available for each **protocol decoder**



If the **serial decoding** parameters entered are incorrect, PicoScope will display an error message.

Below is a list of the various different settings that are available including descriptions.

Data	One or more data channels are required depending on the protocol.
Clock	**Clock** for various protocols
Invert	Reverses the polarity of the signal
Threshold	The voltage that defines the transition in either direction between high and low logic states
Hysteresis	Eliminates the effect of noise on the **threshold**, in a similar way to trigger **hysteresis**. The value you enter here is a range, to be divided equally either side of the **threshold**.
Differential signal	Coupler output for **BroadR-Reach decoding**. First create a **Coupler math channel**, then select this channel as the input to the decoder.
Tx (Master)	Master signal for **Modbus** protocols
Rx (Slave)	Slave signal for **Modbus** protocols
Word Select	For the **I2S** protocol. Indicates which stereo channel the data belongs to.
Chip Select	For the **SPI** protocol. If there is no **Chip Select** signal, clear the check box.

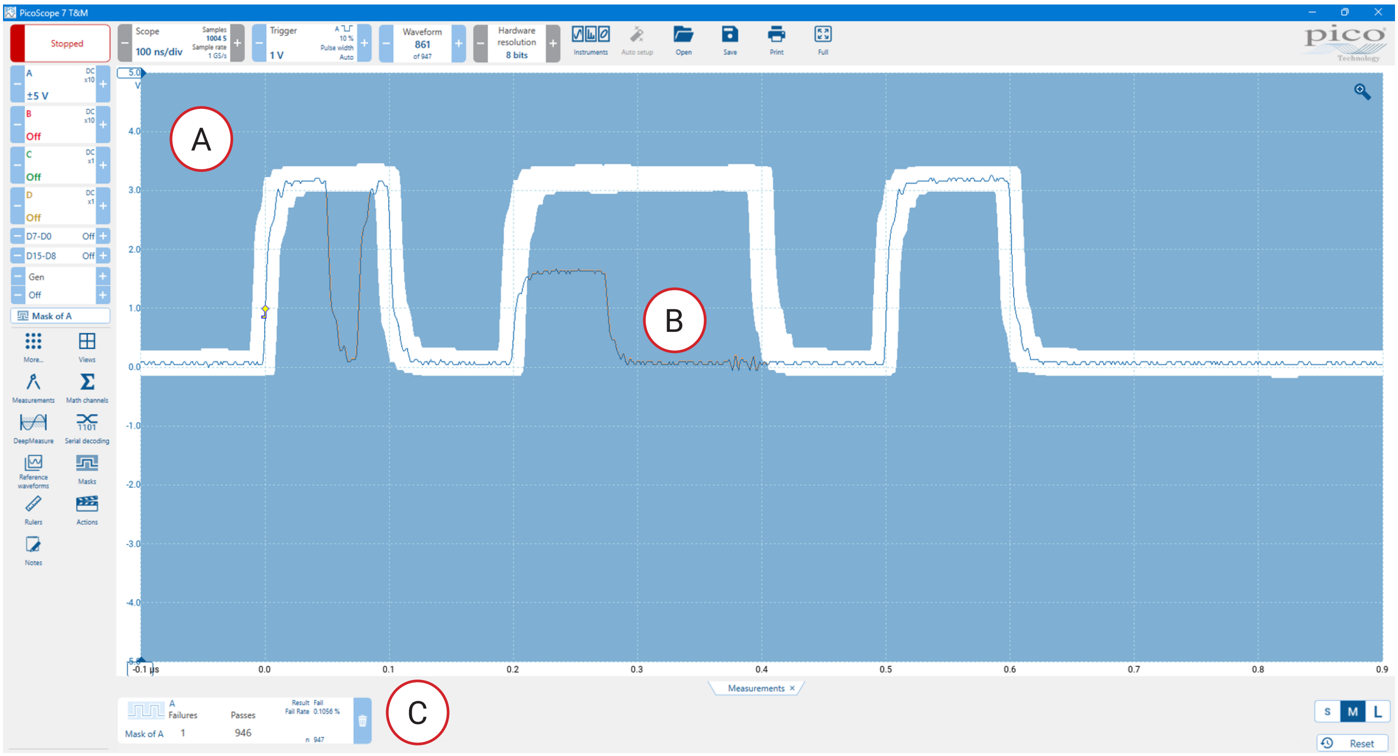
DataPlus & **DataMinus**	For the **USB (1.0/1.1)** protocol.
Baud Rate	The symbol frequency in baud. You can choose one of the standard values from the drop-down list or enter an arbitrary value
Bit Order	Whether the most or least significant bit occurs first For protocols with application-specific bit order, such as **UART**, you must set this to match to the data format
Bus Speed	The maximum data rate of the **I2C serial bus**
Checksum Length	The number of words following the data field to decode as the **checksum**.
Chip Select State	The polarity of the **Chip Select (CS)** signal. Data is decoded only while the Chip Select signal is in the specified state
Clock Edge	Whether to sample the data on the falling or rising edge of the clock (SCK)
Data Bits	The number of bits in the data payload of each packet
Data Bit Rate	The bit rate (in baud) of the data phase of the signal, independent of bit encoding and decoding
Data Encoding	Encoding of the **ARINC 429** data field: **binary coded decimal (BCD)**, **binary number representation (BNR)**, or discrete data representation
Data Length	The number of words decoded as the data field following the header
Data Length Type	**Fixed:** The data field is a fixed length as specified. Used in protocols such as **MILSTD-1553** **Remaining:** The data field is a variable length, equal to the packet length minus the lengths of the other fields. Used in protocols such as **Ethernet 10BaseT**.
Display 4B/5B Code Groups	Show the **4B/5B** block coding scheme output (rather than Ethernet packets) for the **Fast Ethernet** protocol
End Of Packet Byte	For the **Modbus ASCII** protocol. Byte value of the **line feed (LF)** byte.
First Slot	The first **DMX channel** to display
Full-speed (12 Mbit/s)	Full speed **USB 1.0 or 1.1** protocol
Header Length	The number of words decoded as the packet header after the preamble.
High or Low	The signal polarity of the **CAN** or **CAN FD** protocol
High Threshold	The voltage level of the high signal state
Id Bytes	The number of bytes used to identify slave devices in the **Modbus** protocols. Using one-byte slave IDs imposes a limit of 247 slave devices on the network; using two bytes extends this to 65535 devices.

Last Slot	The last **DMX channel** to display
Low-speed (1.5 Mbit/s)	Low speed **USB 1.0 or 1.1** protocol
Low Threshold	The voltage level of the low signal state
Master	Check to decode the master signals on a **BroadR-Reach bus**. Uncheck to decode the slave signals.
Parity	The type of error-correction bit added to each packet
Packet Split Interval	The time, in bit periods, between packets.
Preamble Length	The preamble is a fixed bit pattern that identifies the start of a frame.
Sample time from falling edge	The time since the last **falling edge** at which to sample the **1-Wire bus**
Sensor Type	The type of **SENT** sensor attached. Defines the contents of the packet data nibbles.
Start Bits	The number of fixed bits (1s) at the start of each packet
Stop Bits	The number of fixed bits at the end of each packet
Tick Time	**SENT** transmitter-specific nominal clock period (tick)
Word Length	For **I2S** and **Manchester encoding**. Word length used in the packet.
Display Name	A user-defined name for each protocol setup, which is displayed in the scope view to aid identification when multiple buses are being decoded.
Display packets in	**Graph display format:** shows the data in logic-analyzer style, on the same time axis as the analog waveform. Choose a format for the decoded data: Hex, Binary, Decimal, or ASCII. Selecting "Off" disables in-graph display. • Hover the mouse pointer over any decoded packet to display its contents. • Click-and-drag the decoded data up or down the scope view. • If the Table display is visible, double-click on any packet to highlight it in the table. **Table display format:** displays the decoded data in a table in the serial data window, with advanced search and filtering functions. Choose a format for the decoded data: Hex, Binary, Decimal, or ASCII. Selecting "Off" disables table display.
Between time rulers	Only decodes the section of waveform shown between the time rulers.

7.7 Mask limit testing

Mask limit testing is a feature that tells you when a waveform or spectrum goes outside a specified area, called a mask, drawn on the Scope view or Spectrum view. PicoScope can draw the mask automatically by tracing a captured waveform. Mask limit testing is useful for spotting intermittent errors during debugging, and for finding faulty units during production testing.

When you have selected, loaded or created a mask, the Scope view will appear:



A. Mask: Shows the allowed area (in white) and the disallowed area (in blue). Clicking the **Mask** command in the option panel takes you to the Edit Mask dialog. *See 5.2.8 Masks.*

B. Failed waveforms: If the waveform enters the disallowed area, it is counted as a failure. The part of the waveform that caused the failure is highlighted, and persists on the display until the capture is restarted.

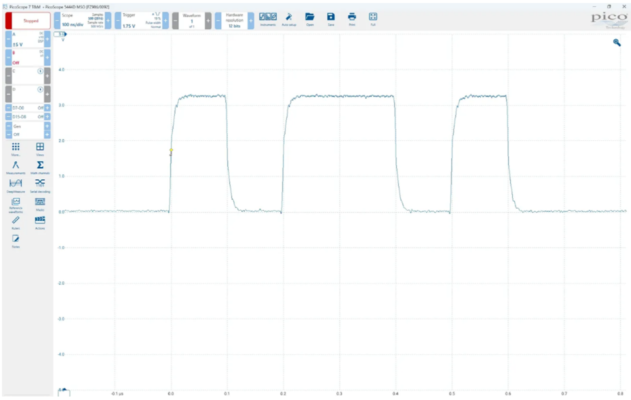
C. Mask failure measurement: The number of failures since the start of the current scope run is shown as a measurement. The measurements tab can display other measurements at the same time as the mask failure count.

7.7.1 Creating and editing masks

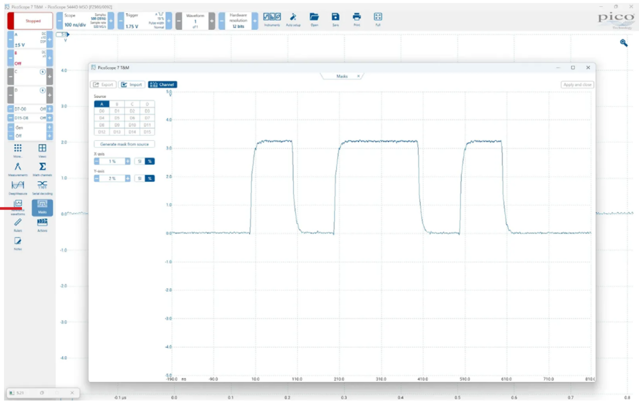
Location: More... Side bar > **Masks**

Purpose: allows you to **create, export** and **import** masks for **mask limit testing**

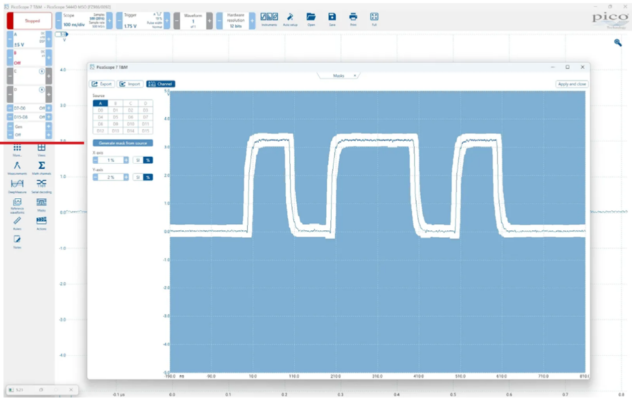
Capturing the golden waveform



Opening up Masks by click the Masks in the side toolbar



Creating the Mask by clicking **Generate mask from source**



Mask button descriptions below:

Channel Displays the options for creating a mask from scope channel data

Import Load a mask that has previously been saved as a .mask file

Export Save a mask as a .mask file for future import

Source Select the channel from whose data you want to auto-generate a mask.
This need not be the same channel you will apply the mask to.

Generate Create a new mask based on the last waveform captured from the selected channel.

X-axis: The horizontal distance between the waveform and the mask.
Y-axis: The vertical distance between the waveform and the mask.

Toggle the offset value between **absolute units (SI)** and **relative units (% of full scale)**.

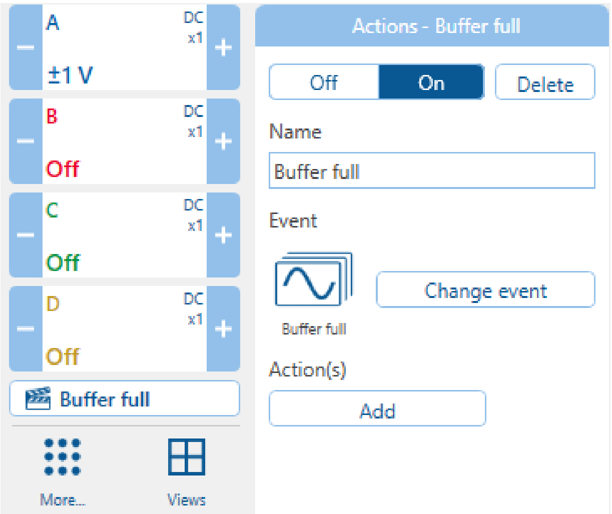
Apply and close Use the selected mask on the selected channel and close the Mask dialog

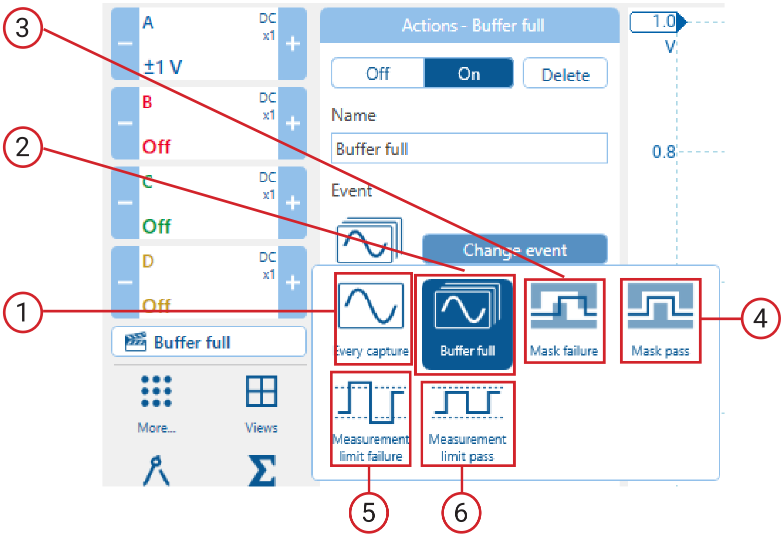
7.8 Actions

The actions feature lets you program PicoScope 7 to carry out one or more "Actions" each time a chosen "Event" occurs.

Location: **More...** or Side bar > **Actions**

Purpose: gives access to the **actions** feature, which specifies actions to be taken on various events



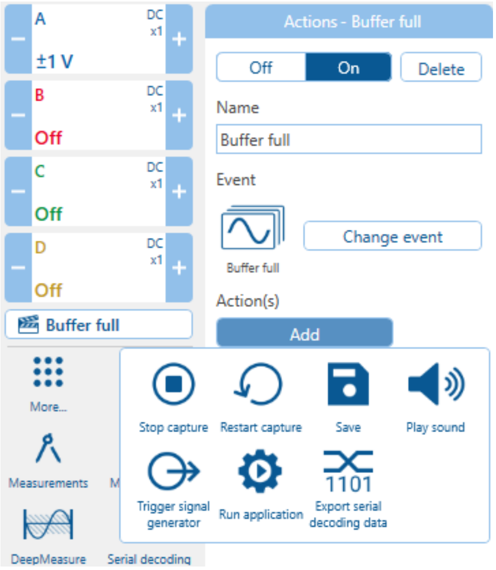


Event(s) that will trigger an **action**:

1. **Every capture:** every time the oscilloscope has captured a complete waveform.
2. **Buffer full:** when the waveform buffer becomes full. This event only happens once per run, unless you also add an action to restart capturing.
3. **Mask failure:** when a waveform fails a mask limit test, on any active mask on any channel.
4. **Mask pass:** when a waveform passes a mask limit test, on any active mask on any channel.
5. **Measurement limit failure:** when a waveform fails at least one active measurement limit.
6. **Measurement limit pass:** when a waveform passes at least one active measurement limit.

Actions list: Add **actions** to this list by clicking **Add**. Whenever the event selected above occurs, PicoScope will execute the all the actions in the list from top to bottom.

Add: Add an action to the **Actions list**. Possible actions are:



Play Sound: specify the name of a **.mp3** sound file to play or choose from the built-in sounds.

Stop Capture: equivalent to pressing the red **Stop** button.

Restart Capture: resume after stopping. This restarts the waveform buffer, meaning any **Buffer full** event can run again. Any following **Save all buffers** actions will contain only new waveforms.

Save: save the current waveform or all waveforms from the buffer as a .psdata file, or in numerous other binary, text and image formats. You can use the %buffer% variable to insert the buffer index number into the file name, or the %time% variable to insert the time of capture. PicoScope will add an incrementing number to the filename if the action runs multiple times.

Trigger Signal Generator: if the scope device has a triggerable signal generator, start generating a signal. The signal generator's Trigger Source option must be set to Manual for this operation to work.

Run application: run the specified program file. You can type the %file% variable after the program name to pass the name of the last file saved as an argument to the program. PicoScope will stop capturing while the program runs, and resume after the program terminates. Please note: Pico Technology are not liable for the actions of any external program file you call from an Action.

Enable external code execution: As a security feature, to enable Run Application you must also select the Enable external code execution option. This option is automatically cleared when loading a psdata or pssettings file to avoid inadvertent code execution.

8. Waveform import/export

8.1 File formats

PicoScope 7 supports saving data in various file formats, including PicoScope's proprietary .psdata format, as well as common formats like CSV, MATLAB (.mat), and PNG graphical format. You can also save settings to a .pssettings file.

The .psdata format stores all capture and view settings, data, and other information for later use and analysis.

Here's a detailed breakdown:

- **.psdata:**

This is the primary format for saving PicoScope data, including capture settings, data channels, measurements, and annotations. It allows for easy sharing and reloading of waveforms with all associated information.

- **CSV:**

Comma-separated values, useful for exporting data to other applications or for further analysis.

- **MATLAB (.mat)**:

A format for data storage and analysis within the MATLAB environment.

- **Graphical format (PNG):**

Saves the displayed waveform as an image for inclusion in reports or presentations.

8.2 Open dialog

Location: **Top toolbar -> Open**

Purpose: allows you to open saved waveforms and settings. You can open files in the following formats:

- **Data files (.psdata)** Contains waveform data and settings (including any custom probes, active math channels etc.) from the scope device used to capture the data. You can create your own files using the **Save** command.
- **Settings files (.pssettings)** Contains all settings, including any custom probes, active math channels etc. from the current scope device. Does not contain any waveform data. You can create your own files using the **Save** command.

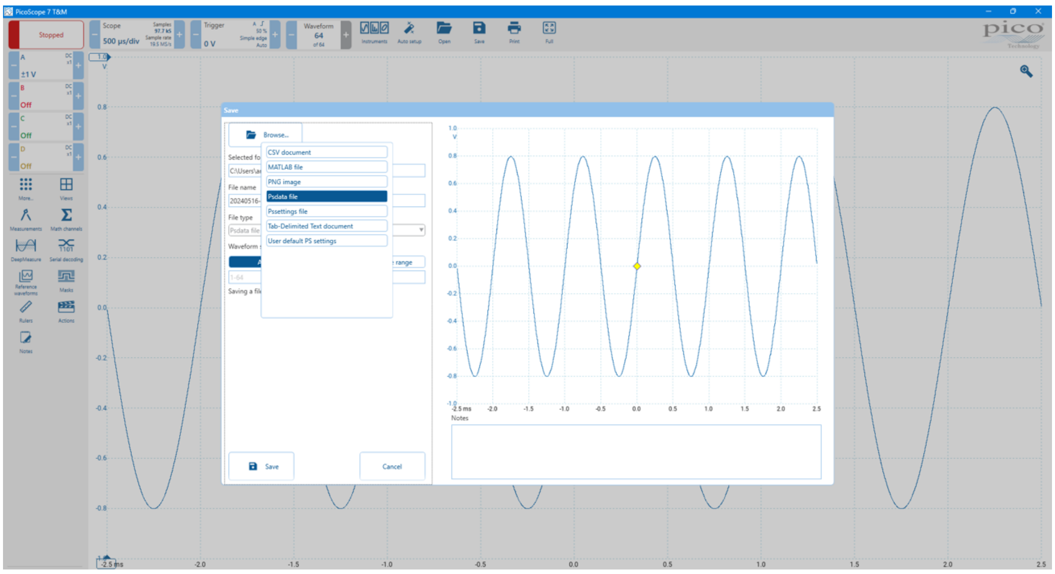
If you open a .psdata or .pssettings file that was saved using a different scope device from the one connected, PicoScope opens the file as if on the device that saved it.

Note: If a device is already connected when a .psdata file is loaded, the channel settings are changed to match the .psdata when you start the scope running with the new settings. The resulting settings are displayed on the Channels lozenge.

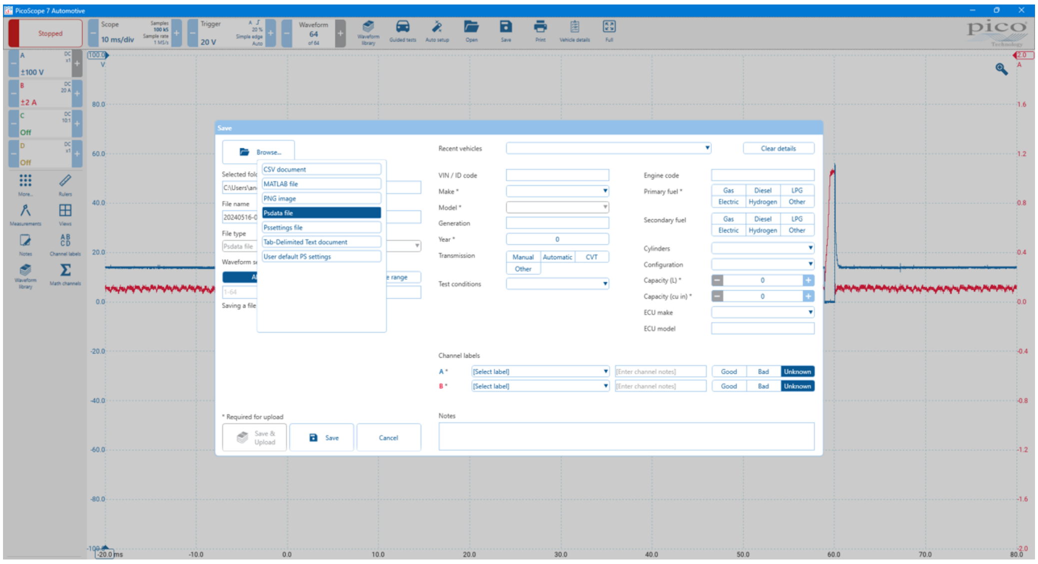
8.3 Save dialog

Location: **Top toolbar -> Save**

Purpose: allows you to save your waveforms and settings to a file in various formats



PicoScope 7 T&M



PicoScope 7 Automotive

- **Data files (.psdata)** Stores waveforms and settings (including any custom probes, active math channels etc.) from the current scope device. Can be opened on any computer running PicoScope, whether or not a PicoScope device is connected.
- **Settings files (.pssettings)** Stores all settings, including any custom probes, active math channels etc., from the current scope device. Does not save any waveform data. Can be opened on any computer running PicoScope.
- **CSV (Comma delimited) files (.csv)** Stores waveforms as a text file with comma-separated values. This format is suitable for importing into spreadsheets such as Microsoft Excel. The first value on each line is the time stamp, and it is followed by one value for each active channel, including currently displayed math channels.
- **Text (Tab delimited) files (.txt)** Stores waveforms as a text file with tab-separated values. The columns are the same as those in the CSV format.
- **MATLAB** files **(.mat)** Stores the waveform data in MATLAB format.
- **User default PS settings** Stores these settings to be recalled every time you launch the PicoScope 7 software. If you have multiple versions of PicoScope 7 installed, for example Early Access and Stable versions, each uses its own default settings file.

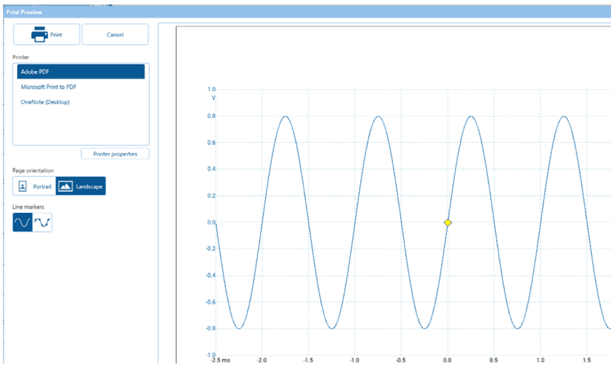
Waveform selection options

The three options control what happens when the waveform buffer contains more than one waveform:

- **All** saves all of the waveforms in the selected file format. If the file format is .psdata, all of the waveforms are collected in a single file. You can then load this into PicoScope and step through the waveforms using the buffer navigation controls. If the selected file format does not support multiple waveforms, PicoScope creates a new directory containing multiple files.
- **Current** only saves the single waveform that is currently on view.
- **Choose range** Saves the specified list or range of waveforms. Each waveform is identified by its index number. Example format: 1, 2, 10-20.

8.4 Printing dialog

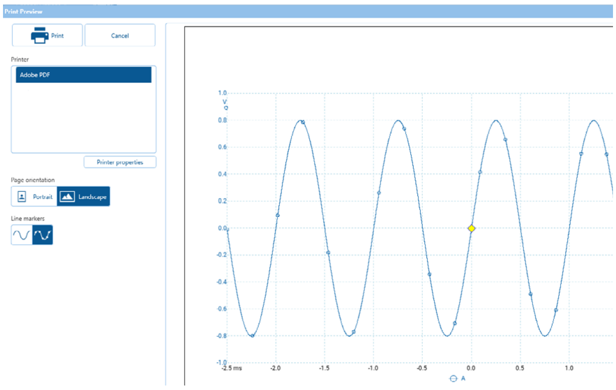
Location: Top bar > **Print**



Printer Allows you select what medium to print to.

Page orientation Portrait or Landscape.

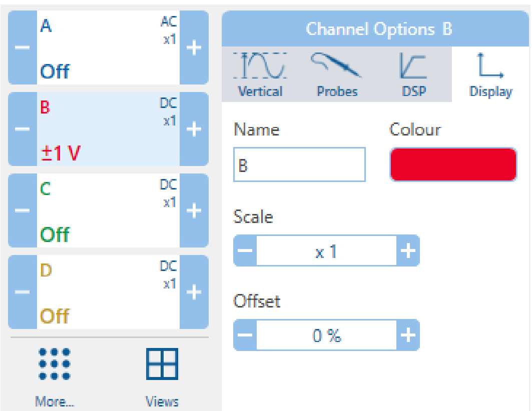
Line markers Allow the line markers to be included or not, shown in the image below:



8.5 Channel display options

Location: **Channel button** > **Display**

Purpose: control how the channel trace is represented on screen



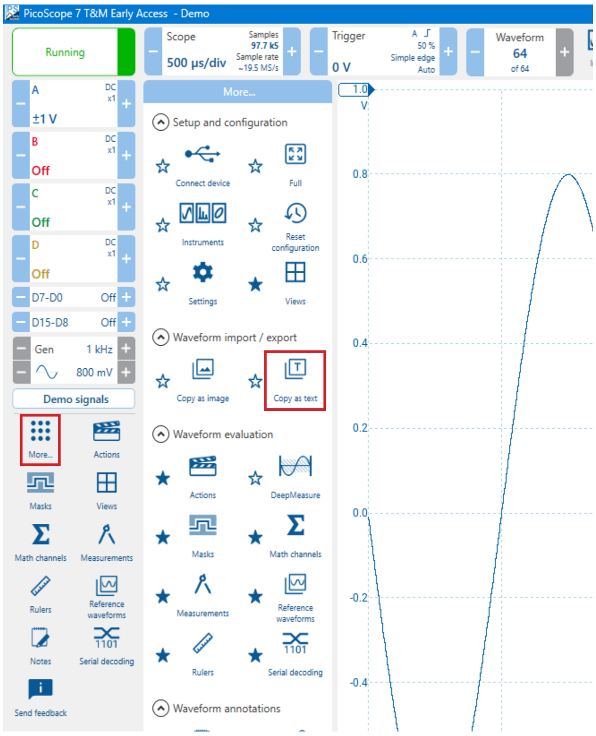
Channels the trace color for each scope channel can be set here.

Scale increases/decrease the waveform size on the screen.

Offset moves the waveform display vertically on screen.

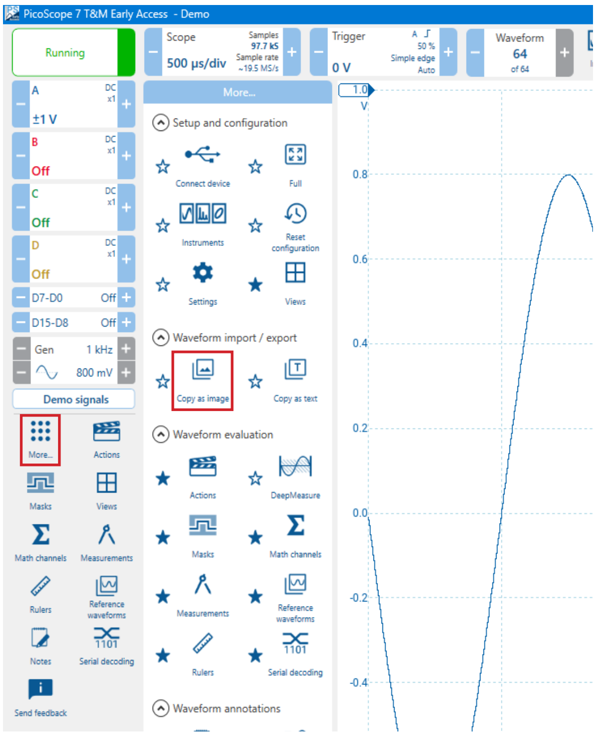
8.6 Copy as text

In PicoScope 7, you can copy waveform data as text using the **Copy as text** option, which is found by clicking **More... > Copy as text**. This function allows you to copy the raw waveform data in text format to the clipboard, making it easy to paste the data into other applications like spreadsheets or text editors.



8.7 Copy as image

In PicoScope 7, you can copy waveform data as an image using the **Copy as image** option, which is found by clicking **More... > Copy as image**. Copy as Image will include an image of the graph, with any measurements and included notes.

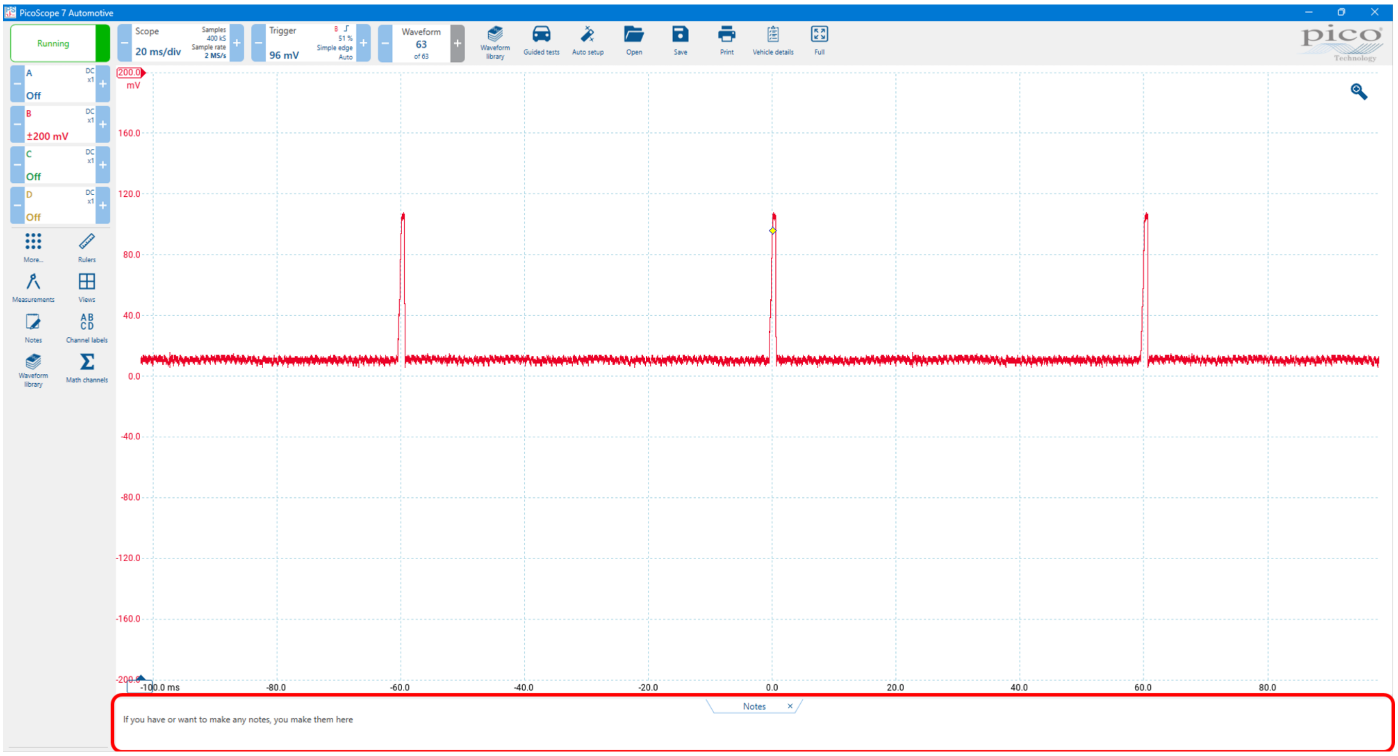


9. Waveform annotations

9.1 Notes

Location: **More... > Notes**

Purpose: provides a text box for typing your own notes



You can place a Notes area at the bottom of the PicoScope window (see image above), and you can enter any text you wish in it or paste text into it from other programs. This text is included when you save the waveform as a file. For more information, see *www.picotech.com/library/knowledge-bases/oscilloscopes/annotations*

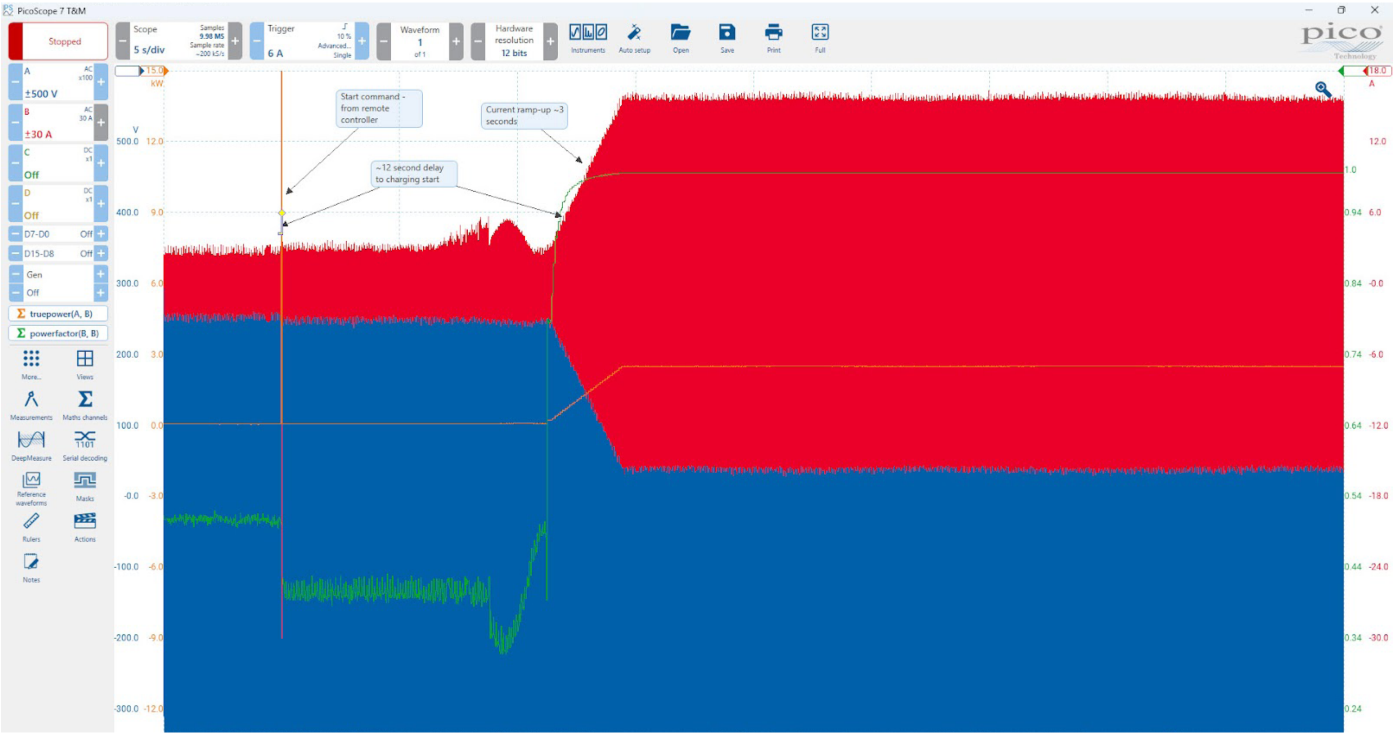
9.2 Annotations

Location: **More... > Annotations**

Waveform annotations provide a way to make notes directly onto the graph. Add freeform text boxes onto the graph, edit them and help explain or draw attention to specific events or anomalies in the data by dragging fixed pinpoint arrows to them.

Additionally, these annotations are visible on printouts and image exports, which might be required for sharing and distribution.

To add an annotation, press the Waveform annotation tool from the More Tools options panel to quickly add a new annotation text box into the graph.



Easily resize the annotation using the handles that appear when it's selected, and move it around by grabbing the sides of the annotation, allowing you to position it where it fits appropriately within the data

Quickly add precision arrows by clicking the arrow above/below the annotation and dragging it to specific anomalies or events of focus within the waveform. Multiple arrows can be added for each annotation.

These arrows will be fixed to the data even as you zoom in or move the annotation textbox around the graph.

If you no longer need an annotation and want to remove it, simply use the delete button next to the annotation to remove it from the graph.

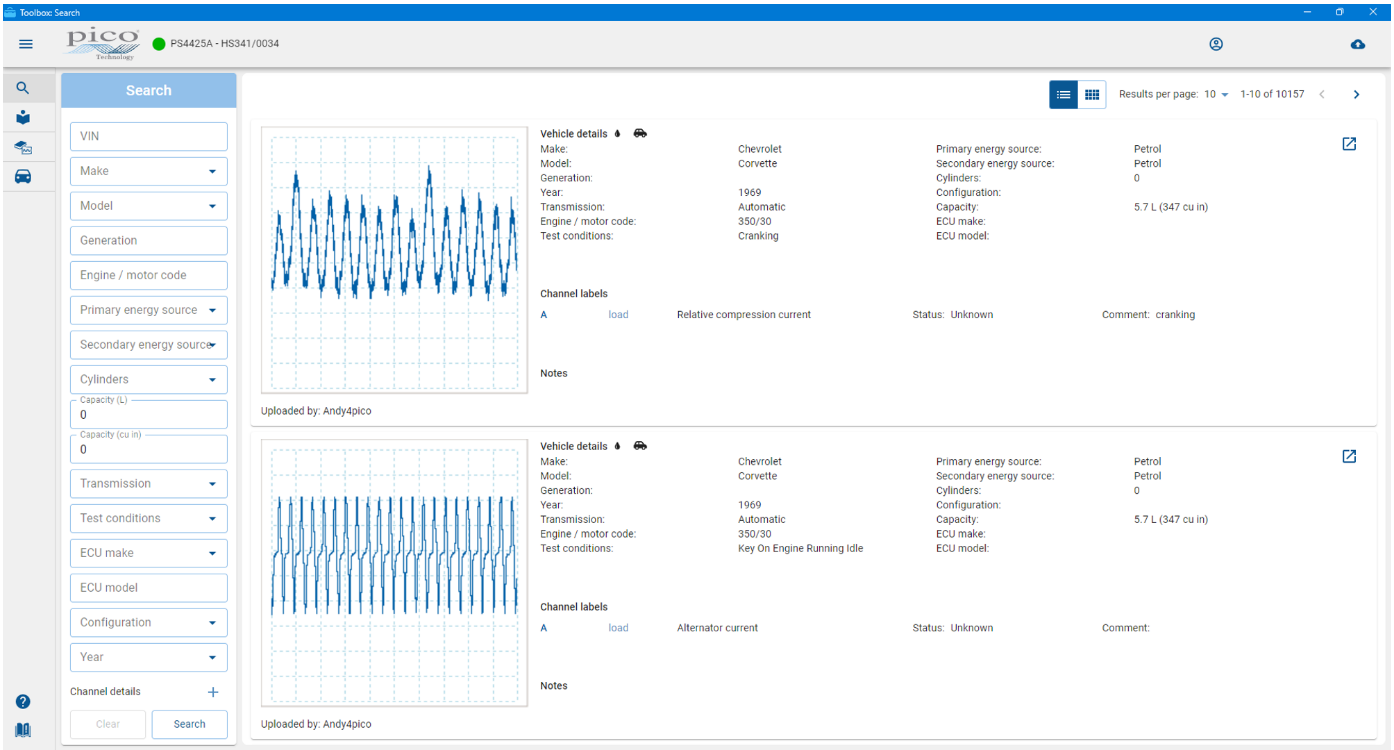
10. Automotive diagnostics

10.1 Waveform Library

Location: **Top bar > Waveform Library**

Purpose: allows you to search hundreds of user-uploaded waveforms by entering various fields of required data

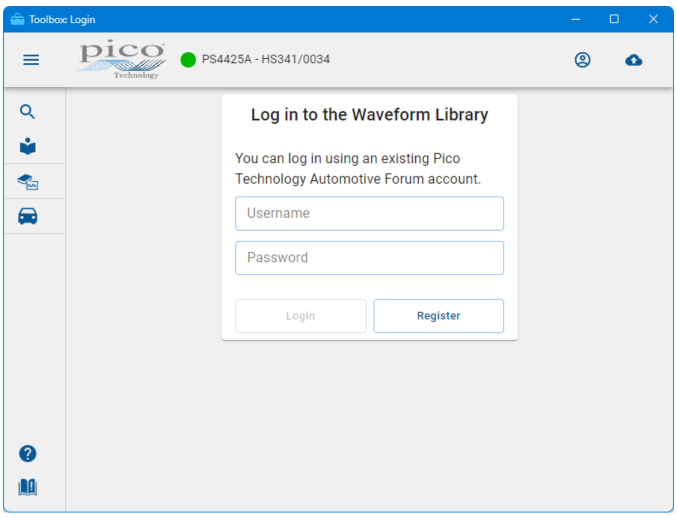
Applicability: PicoScope Automotive software only



Users of PicoScope Automotive oscilloscopes have access to the online Waveform Library. This is a collection of user-generated waveforms, with information about the vehicle under test. You can search the Waveform Library for waveforms from the vehicle and component you want to test, and also share your own examples. You can also treat the Waveform Library as an online backup for your own waveform files.
Once you have found a waveform, you can preview or open it. You can also use the waveform's individual channels as reference waveforms for your own signal, in PicoScope software.

Login details

When you open the Waveform Library, a login page appears. Enter your username and password, and the software will take you to the page shown below:



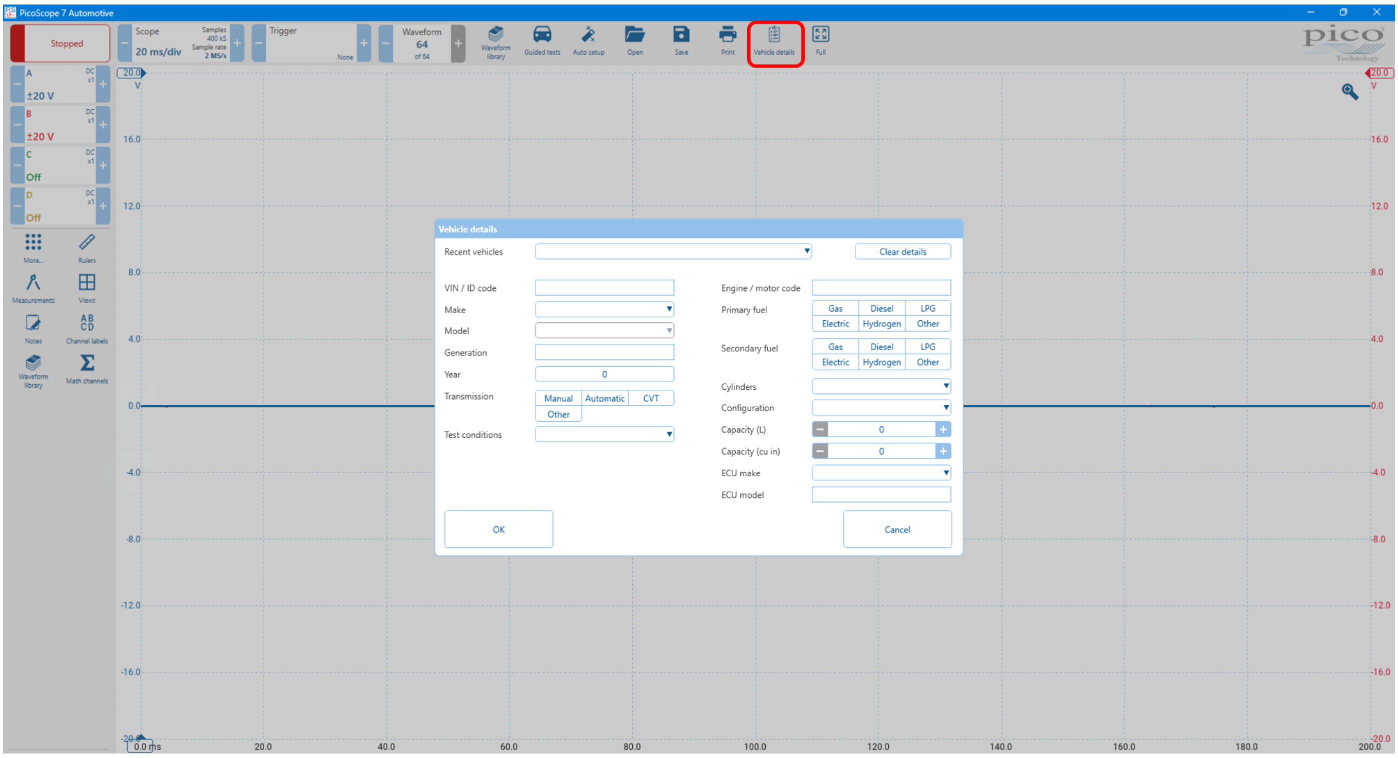
The Waveform Library uses the same account information as the Pico automotive forum: if you are not already a member, you can register for free by following the link on the Waveform Library Login page. For more information see, *www.picoauto.com/library/picoscope/reference-waveform-library*

10.2 Vehicle details

Location: **Top bar > Vehicle details**

Purpose: allows you to record vehicle details, before saving a file

Applicability: PicoScope Automotive only



All the details you need to identify a vehicle can be entered into Vehicle details. The details can then be stored when you save your waveforms to a PicoScope data (psdata) file or upload them to the Waveform Library.

Often, it is critical that you have the correct reference waveforms for your vehicle, otherwise you may have no way of telling good from bad and diagnosing a fault. Furthermore, as you (and other Waveform Library users) build your waveform collection over time, it can be difficult to look back and know which waveforms came from which vehicle so your saved waveforms can become less useful as a reference tool. Vehicle details solves this problem: by completing the Vehicle details dialog prior to testing a vehicle, you will automatically have the reference vehicle information available when you come to save or upload your captured waveforms.

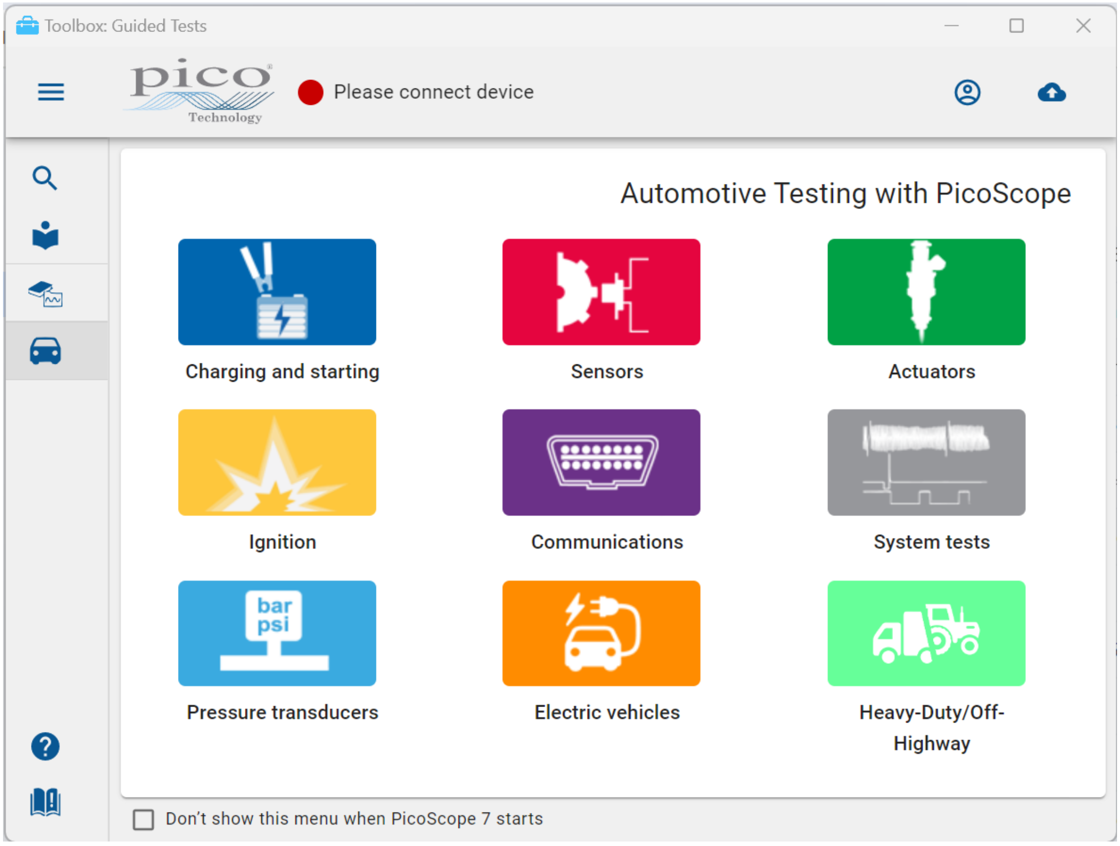
The vehicle information stored in uploaded files is also used by the Waveform Library to identify suitable reference waveforms when you (or others) use the search tool to find them.

10.3 Automotive - Guided tests

Location: Top bar > **Guided tests**, this will also appear every time you open the PicoScope 7 Automotive software

Purpose: gives access to a wide variety of preset **guided tests**

Applicability: PicoScope Automotive software only



The Automotive Toolbox panel contains over 160 preset guided tests ready for you to perform on vehicles.

To run a guided test

1. Select the category of test you wish to run.

2. The Guided Test application will provide a list of tests related to the category selected.

3. When you have selected a test, the options will be narrowed down to even more specific tests. These tests will provide you with two options, **Load Settings File** and **Guide and Setting File**.

If you have selected **Guide and Setting File**, the Guided Test program will guide you through setting up the PicoScope software to be able to perform the relevant Guided Test.

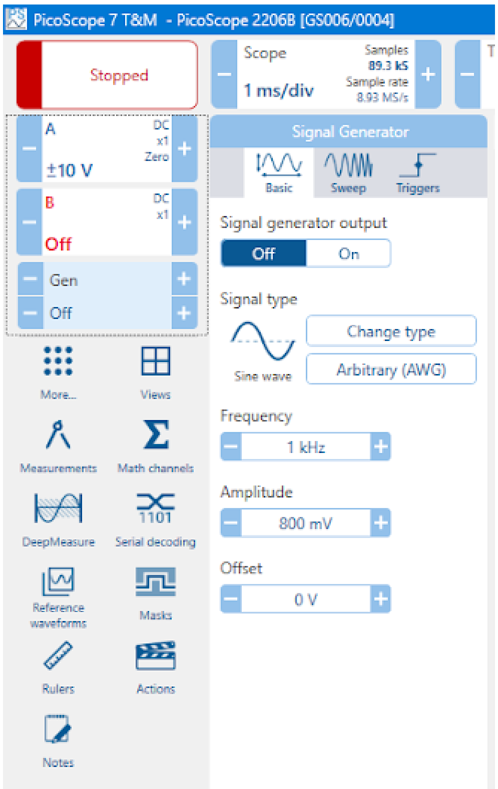
If you have selected the **Load Settings File**, the PicoScope software will load the pre-set waveforms and settings to enable you to carry out the test.

For more information, see *www.picoauto.com/library/automotive-guided-tests*

11. Signal Generator

11.1 Signal Generator button

The **Signal Generator** button allows you to set up your scope's test signal generator.



If your scope has a built-in **signal generator,** clicking the **Signal Generator** button opens the **Signal Generator options panel**

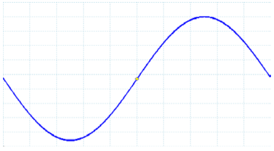
If PicoScope is in demo mode then clicking the **Demo signals** button opens the **Demo Signals** option panel with which you can configure a virtual input signal for each channel of the demo scope.

Signal generator output	Click **On** to enable the **signal generator**
Signal Type	Select the type of signal to be generated. The list of signal types depends on the capabilities of the scope device.
Arbitrary...	Opens the **Arbitrary Waveform Generator** window. This button is available only if your scope has an **AWG**.
Frequency	Type in this box or use the spin buttons to select the frequency. If the scope device has a **frequency sweep generator**, then this box sets the start frequency of the sweep.
Amplitude	The peak amplitude of the waveform. For example, if **Amplitude** is 1 V and **Offset** is 0 V, the output will have a negative peak of –1 V and a positive peak of +1 V.
Offset	The mean value of the signal. For example, when Offset is 0 V, a sine or square wave will have equal positive and negative peak voltages.

11.2 Signal generator waveform types

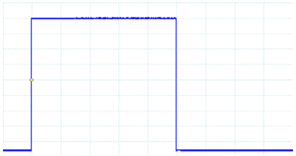
The list of waveform types available in the **Signal Generator** varies according to the type of oscilloscope connected. The full list is as follows:

Sine



Sinusoid

Square



Square wave

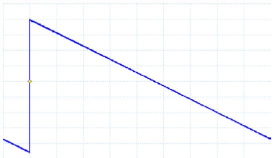
Triangle

Symmetrical triangle wave

Ramp Up

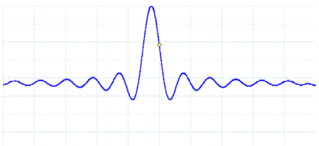
Rising sawtooth

Ramp Down



Falling sawtooth

Sin (x)/x



sin(x)/x, truncated on the x axis

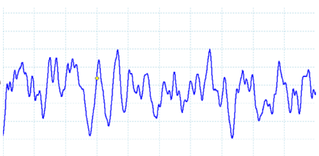
Gaussian

The 'bell curve' of the normal distribution, truncated on the x axis

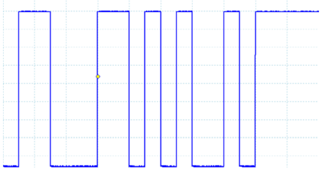
Half Sine

A rectified sinusoid

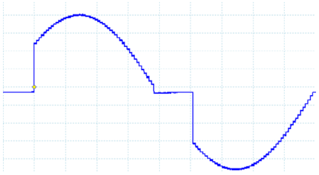
White Noise



Random samples at the maximum update rate of the **AWG**

PRBS  Pseudo-random binary sequence - a random sequence of bits with adjustable bit rate

DC Voltage Constant voltage, adjustable using the **Offset** control

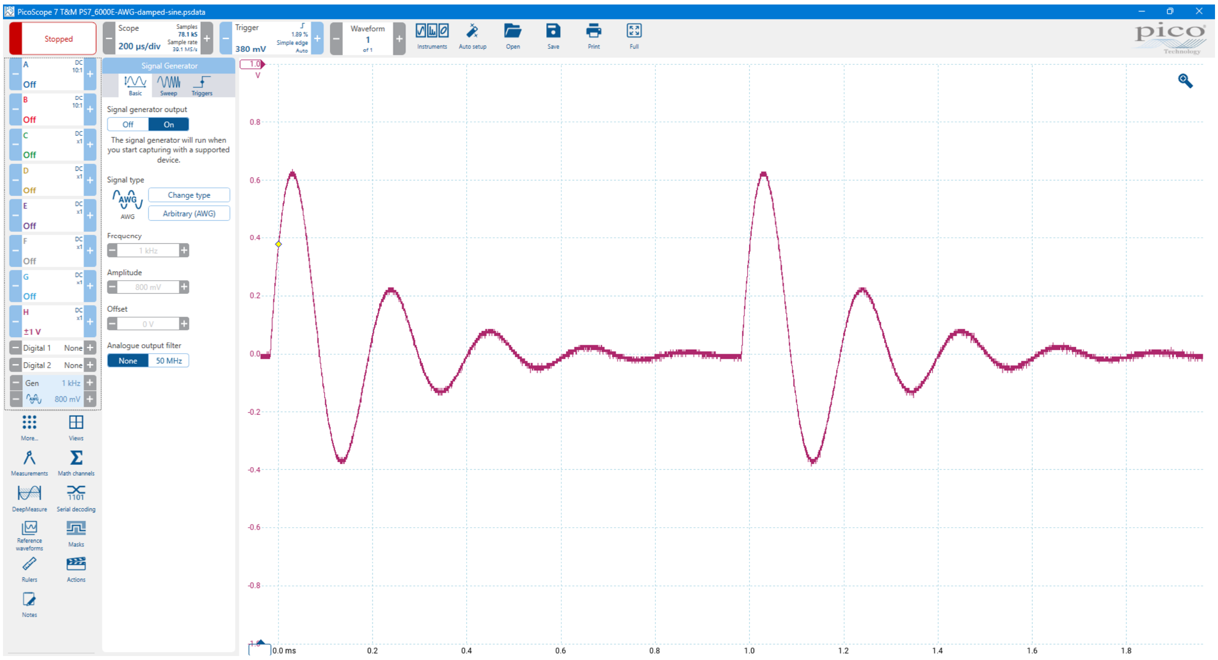
Arbitrary  Any waveform created by the **arbitrary waveform** editor

11.3 Arbitrary Waveform Generator window

Location: **Signal Generator > Arbitrary**

Purpose: allows you to import, edit, draw and export arbitrary waveforms to load into your scope's arbitrary waveform generator. You can also import and export the data in CSV format for use in other applications.

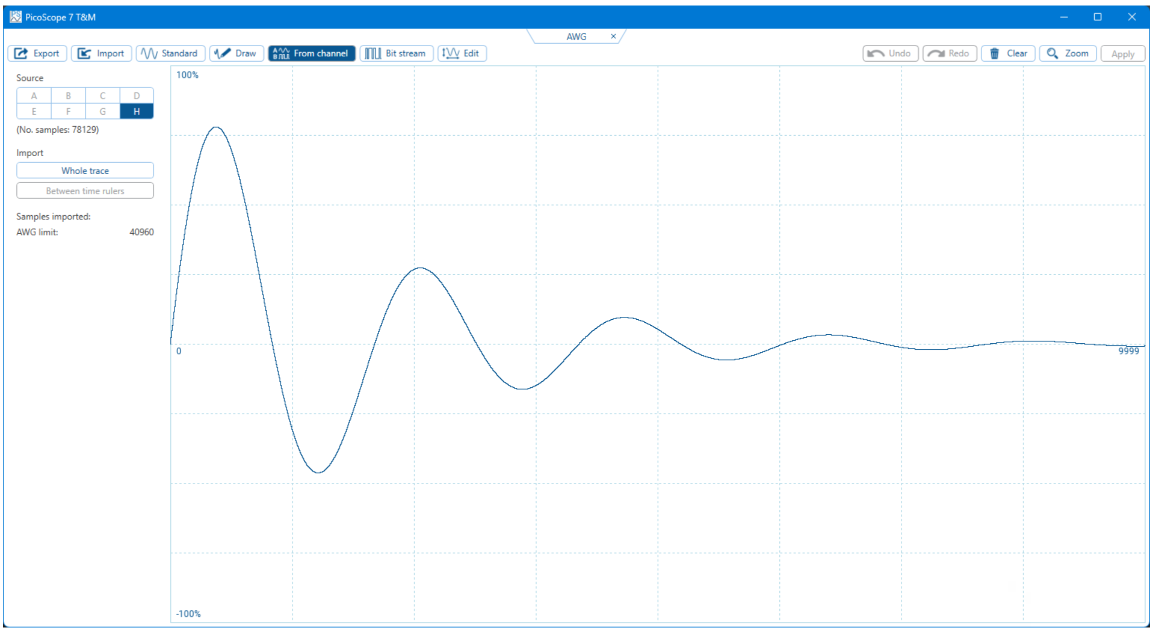
Many PicoScope PC oscilloscopes have an arbitrary waveform generator (AWG), which is enabled using the Signal Generator options panel. You can program the AWG to a standard waveform shape, such as a sine or square wave, use the Smooth Drawing and Line Drawing controls to customize one of these standard waveforms or to create an arbitrary waveform, or import an arbitrary waveform from a CSV file or one of the oscilloscope channels.



11.3.1 Import from a Channel dialog

Location: **Arbitrary Waveform Generator** window > **From Channel** button

Purpose: allows you to copy captured data from a scope channel to the **Arbitrary Waveform Generator** window



From channel You can import the latest waveform from any available channel. This control allows you to specify a subset of the capture, either the whole trace or between **time rulers**. The subset will be scaled to fit the number of samples specified in the **Samples** control in the **Arbitrary Waveform** window.

11.3.2 Import from CSV

Location: **Arbitrary Waveform Generator** window > **Import** button

Purpose: allows you to copy an existing **arbitrary waveform** to the **Arbitrary Waveform Generator** window

You can import an existing **arbitrary waveform** into PicoScope software as a **CSV** file, which must be laid out as a single column of decimal values. The minimum number of values a file may have is 10; the maximum is determined by the size of the **AWG** buffer (specified in the **data sheet** for your scope device).

The values are samples between –1.0 and +1.0 and are equally spaced in time. The output is scaled to the amplitude selected in the Signal Generator options panel and the selected offset is added if necessary. For example, if the **signal generator** amplitude is set to 1 V and the offset to 0 V, then a sample value of –1.0 corresponds to an output of –1.0 V and a sample of +1.0 corresponds to an output of +1.0 V.

Your file should contain exactly one cycle of the waveform, which PicoScope will then play back at the **Start Frequency** specified in the Signal Generator options panel. In the example above, the **signal generator** was set to 1 kHz, so one cycle of the waveform lasts for 1 ms. There are 10 samples in the waveform, so each sample lasts for 0.1 ms.

To **import** a file, click the **Import** button in the **Arbitrary Waveform Generator** window. An **Open** dialog will appear, allowing you to find and select your file. The waveform appears in the window: click **OK** or **Apply** to start using it.

11.3.3 Export as CSV

Location: **Arbitrary Waveform Generator** window > **Export** button

Purpose: allows you to **save** a copy of your **arbitrary waveform**

PicoScope software allows you to save the waveforms you create in the **Arbitrary Waveform Generator** window.

When you have created an **arbitrary waveform** and are happy with the result, click the **Export** button to open a **Save As... dialog** and save your work as a **CSV** file in the location of your choice.

You can now use the **Import** from **CSV** button to play back your waveform whenever you like.

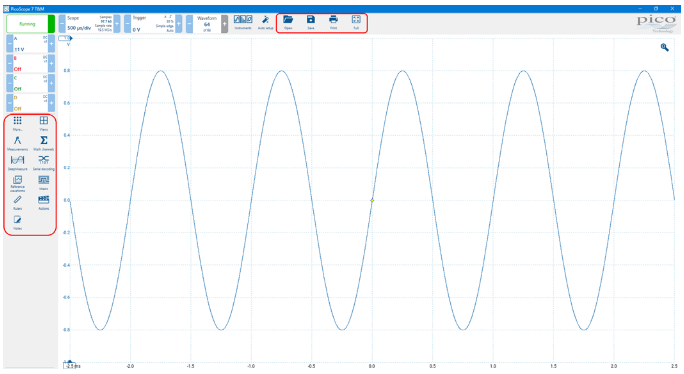
The file will contain values of your choice between –1.0 and +1.0, evenly spaced in time. The output is scaled to the amplitude selected in the Signal Generator options panel and the selected offset is added.

12. Setup and configuration

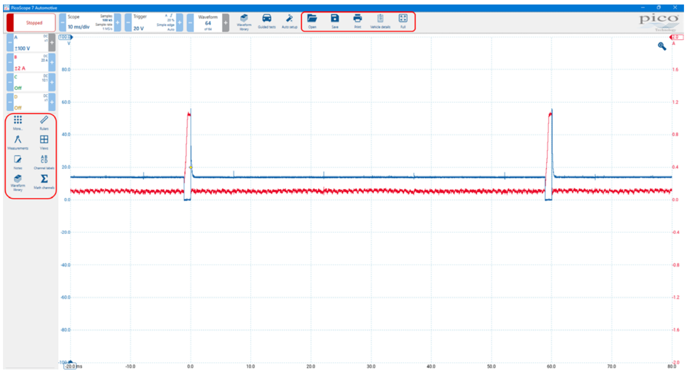
Controls for the main features of PicoScope are always visible in the top and side toolbars as shown in Figure 1.

The side bar allows the user to view other functions within the PicoScope software in more detail. The **More...** button will expand the side bar option panel (Fig.2) when selected to show other functions which are not initially displayed on the Top bar or Side bar when the PicoScope software is opened. The user can 'Star' each of the function which they use more frequently or need to prioritize, this will create a personalized Side bar for ease of access to these prioritized or frequently used functions.

Fig.1

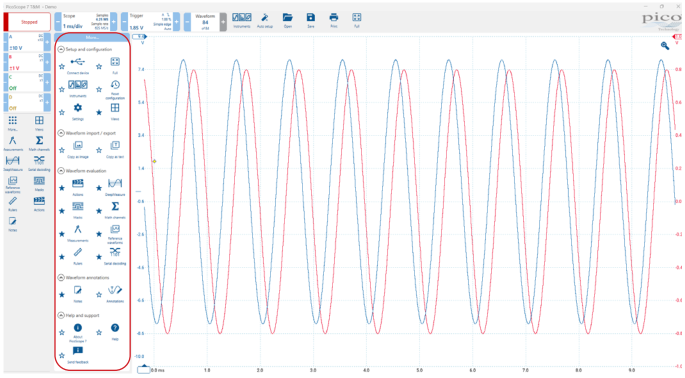


PicoScope 7 T&M

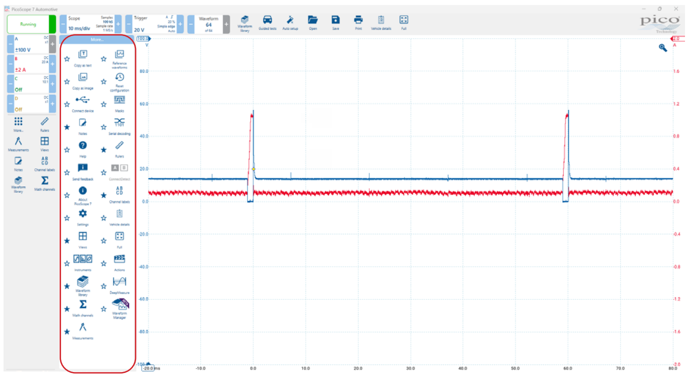


PicoScope 7 Automotive

Fig.2



PicoScope 7 T&M



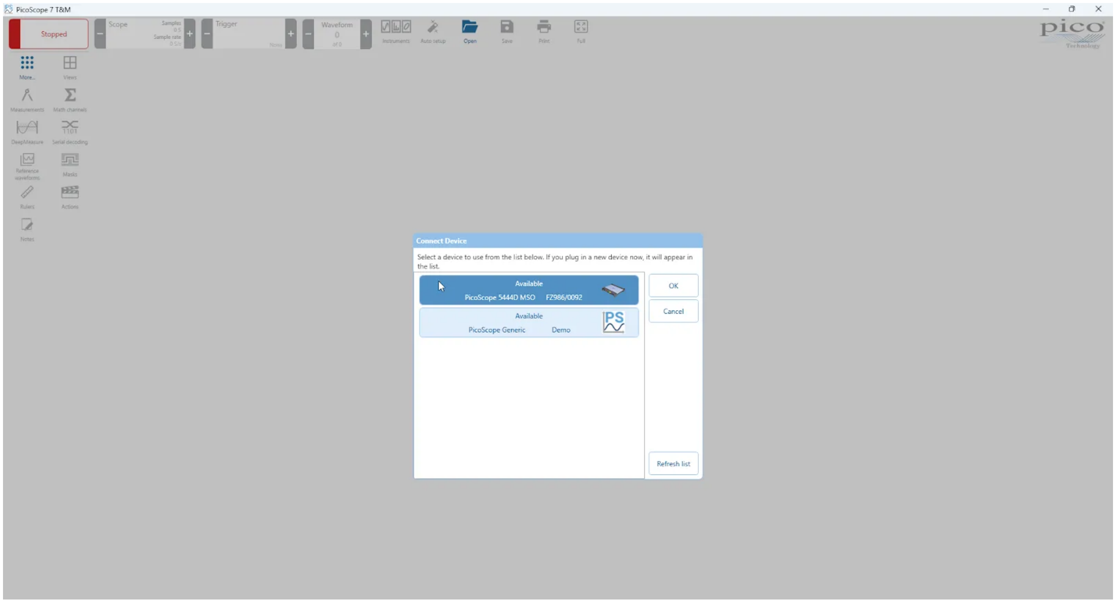
PicoScope 7 Automotive

The list of items in the top bar may vary depending on the windows that you have open in PicoScope.

12.1 Connect Device dialog

Location: **More... > Connect Device** or start PicoScope with more than one scope device connected

Purpose: when PicoScope finds more than one available scope device, this dialog allows you to select which one to use



Note: PicoScope Automotive software is only compatible with Pico's range of automotive oscilloscopes. Non-automotive oscilloscopes, will appear as unsupported in the list of available devices, as they cannot be used with the PicoScope Automotive software.

You can switch to a different scope device, or between Demo and a real scope, by clicking **Connect Device**.

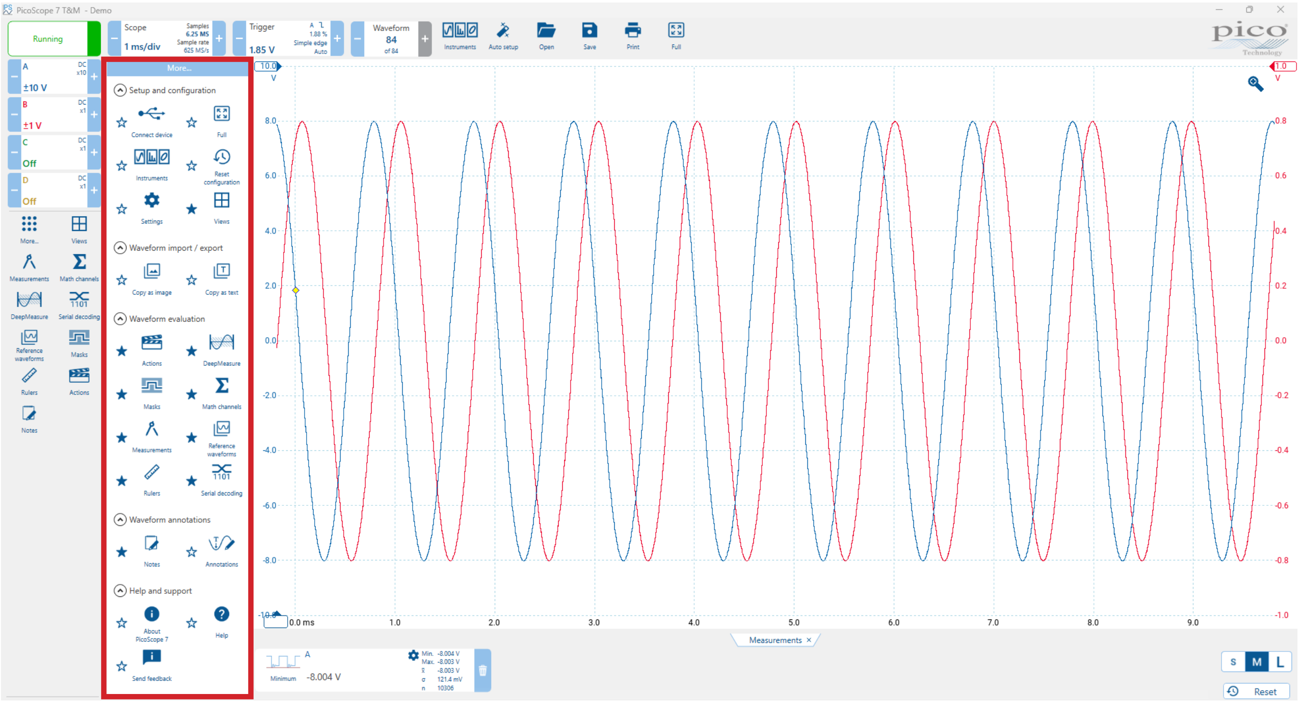
Connecting a new device

1. Open the Connect Device dialog and wait for the list of scope devices to appear. This may take a few seconds.
2. Select a device and click **OK**.
3. PicoScope will open a scope view for the selected device.
4. Use the option panel to set up the device and the scope view to display your signals.

12.2 More... options panel

Location: **More... >**

Purpose: gives access to assorted tools for signal analysis



Setup and configuration:

Connect device - This will show a list of any PicoScope devices you have connected to your PC or Laptop.

Full - expands the graph area of your chosen View so it fills the whole screen.

Instruments - Allows the user to select Scope, Spectrum, XY or Persistence.

Reset configuration - Resets to the same configuration chosen in preferences for start-up: factory settings, user-saved default settings, or last session settings.

Settings - Set various options that control PicoScope's behaviors.

Views - provides different ways to view, arrange and group your waveforms and analysis tools.

Waveform import / export:

Copy as image

Copy as text

Waveform evaluation:

Actions - can be programmed and will execute when certain events occur.

DeepMeasure - For more information, see *www.picotech.com/deepmeasure*

Masks - For more information, see *www.picotech.com/mask-limit-testing*

Math Channels - For more information, see *www.picotech.com/math-and-measurements*

Measurements - For more information, see *www.picotech.com/math-and-measurements*

Reference waveforms - For more information, see *www.picotech.com/reference-waveforms*

Rulers - to take measurements between different points on your waveforms.

Serial decoding - For more information, see *www.picotech.com/serial-protocol-decoding*

Waveform annotations:

Notes - is a free text area that can be used to describe an entire dataset.

Annotations - provide a way to make notes directly onto the graph.

Help and support:

About PicoScope 7 - Software version, Legal notices, update checker, Driver version, etc.

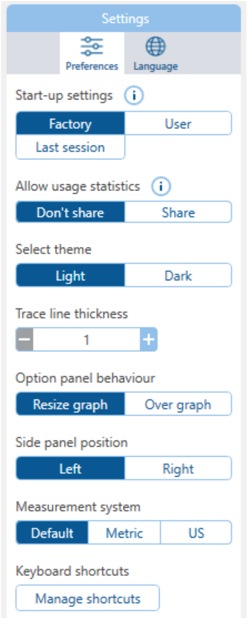
Help - This takes you to *www.picotech.com/library/knowledge-bases/oscilloscopes*

Send feedback

12.3 Settings options panel

Location: **More... > Settings > Preferences**

Purpose: Allows you to set options for the PicoScope software.



Start-up settings

Factory - recall the default PicoScope settings each time you start the PicoScope application

User - recall your personalized settings each time you start the PicoScope application. To use this feature you must also save your User default settings using the Save dialog

Last Session - recall your last session each time you start the PicoScope application

Allow usage statistics

PicoScope 7 shares anonymous usage statistics, this helps us improve reliability and performance

Select theme

This allows you to display the software in light or dark theme

Option panel behavior

Choose whether opening the option panel shrinks the graph area, or extends over the graph.

Trace line thickness

This allows you to increase or decrease the line thickness

Side panel position

This allows you to move the side panel from the left hand side of the software window to the right hand side and vice versa

Measurement system

This allows you to choose your preferred measurement system for temperature and pressure values:
- Metric: Bar, Celsius
- US / Imperial: PSI, Fahrenheit.
- Electrical quantities are always measured in SI units.

Keyboard shortcuts

This allows you to manage and toggle between **Basic** and **Advanced (QWERTY)** keyboard shortcut settings

12.3.1 Languages

Location: **More...** > **Settings > Language**

Purpose: lets you select the language settings for PicoScope's user interface

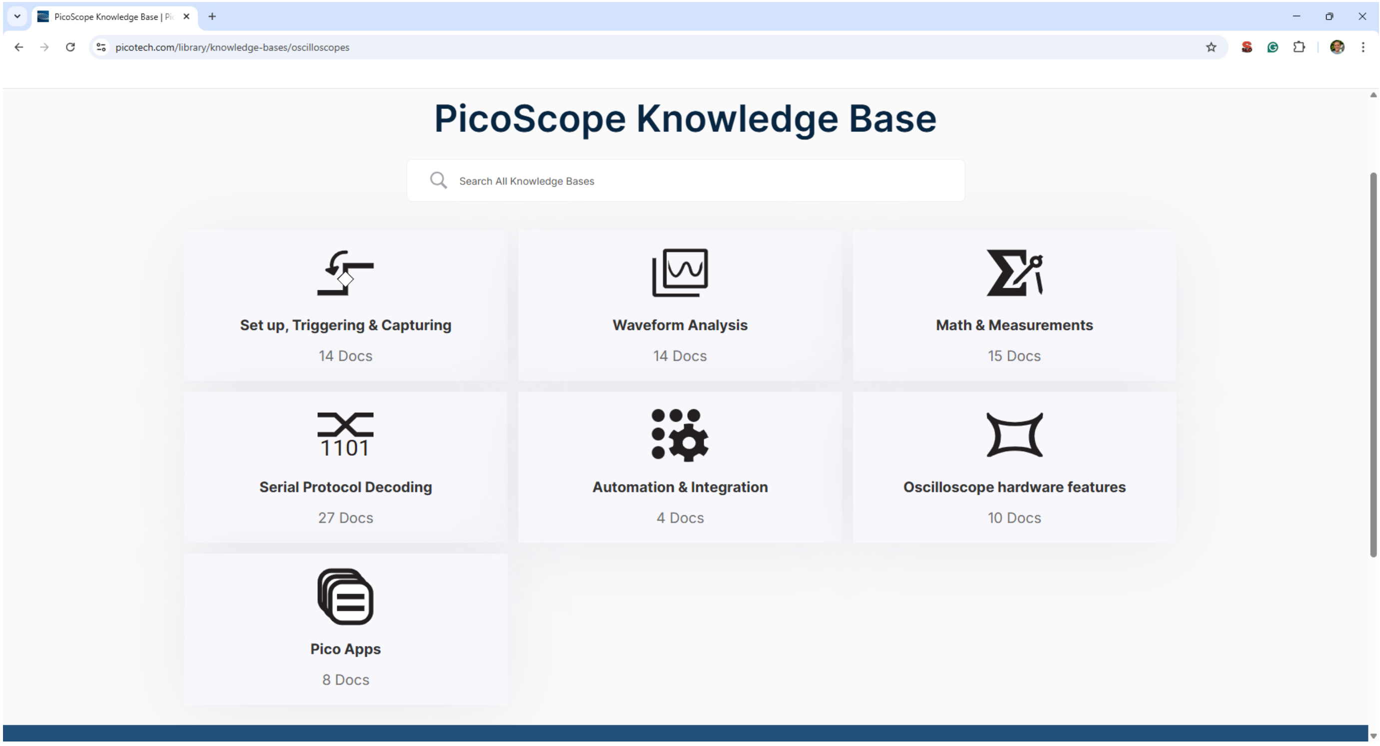
Please note: In Early Access releases of PicoScope, the latest new features may be labelled in English regardless of the selected language.

13. Help and support

13.1 Help - T&M

Location: **More... > Help**

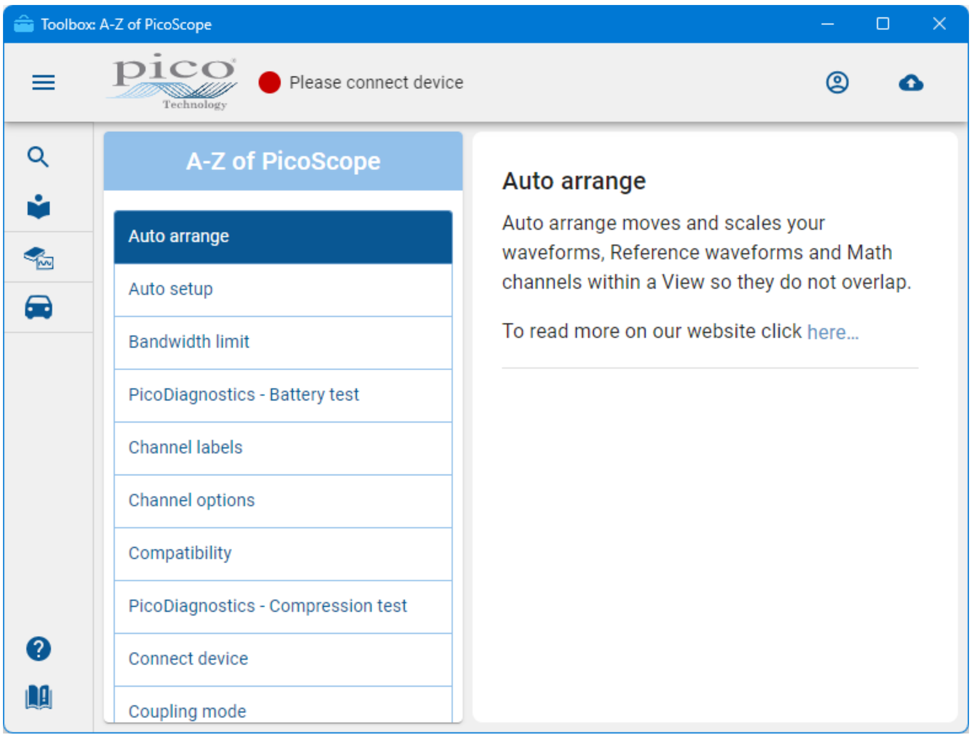
Purpose: gives access to the PicoScope software Knowledge Base and related information



13.2 Help - Automotive

Location: **More... > Help**

Purpose: gives access to the A to Z of PicoScope.



13.3 Send Feedback dialog

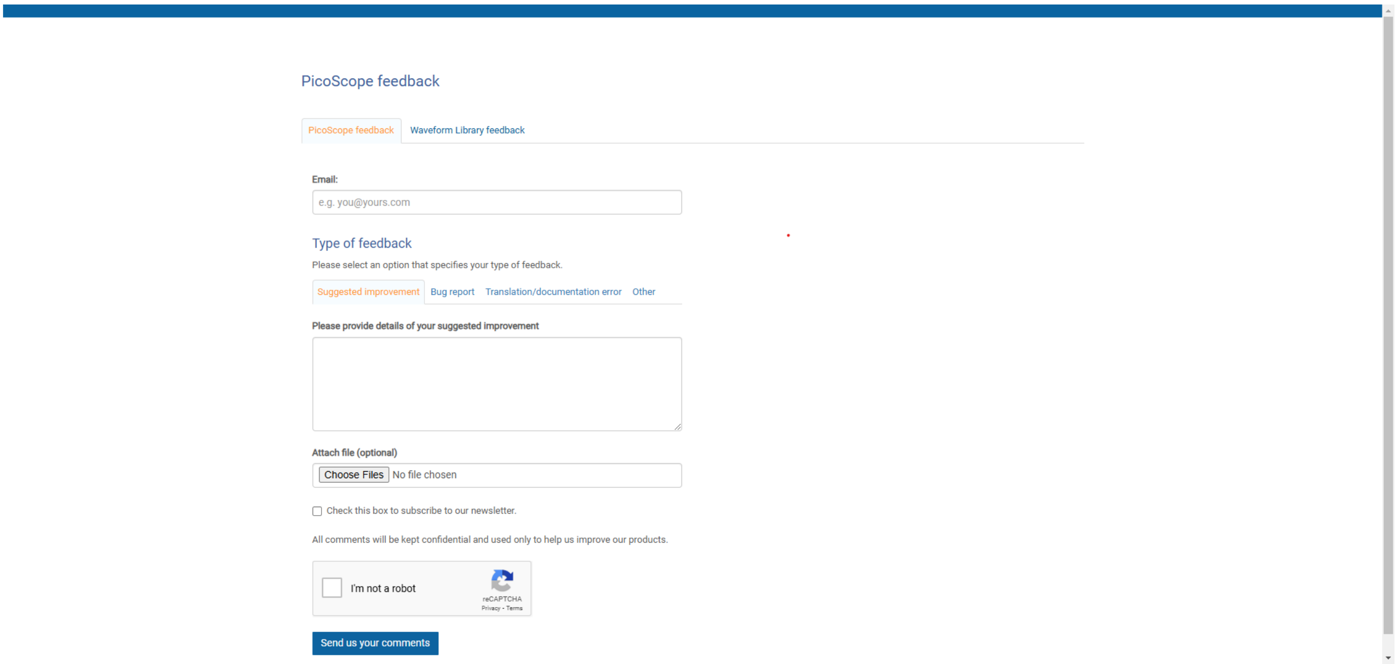
Location: **Help > Send Feedback**

Purpose: allows you to send comments, queries and bug reports to Pico Technology

The **Send Feedback** command opens a web page (requires an internet connection) through which you can let us know about your experience using PicoScope software.

If you are reporting a bug, please let us know what you were doing, what happened and what should have happened.

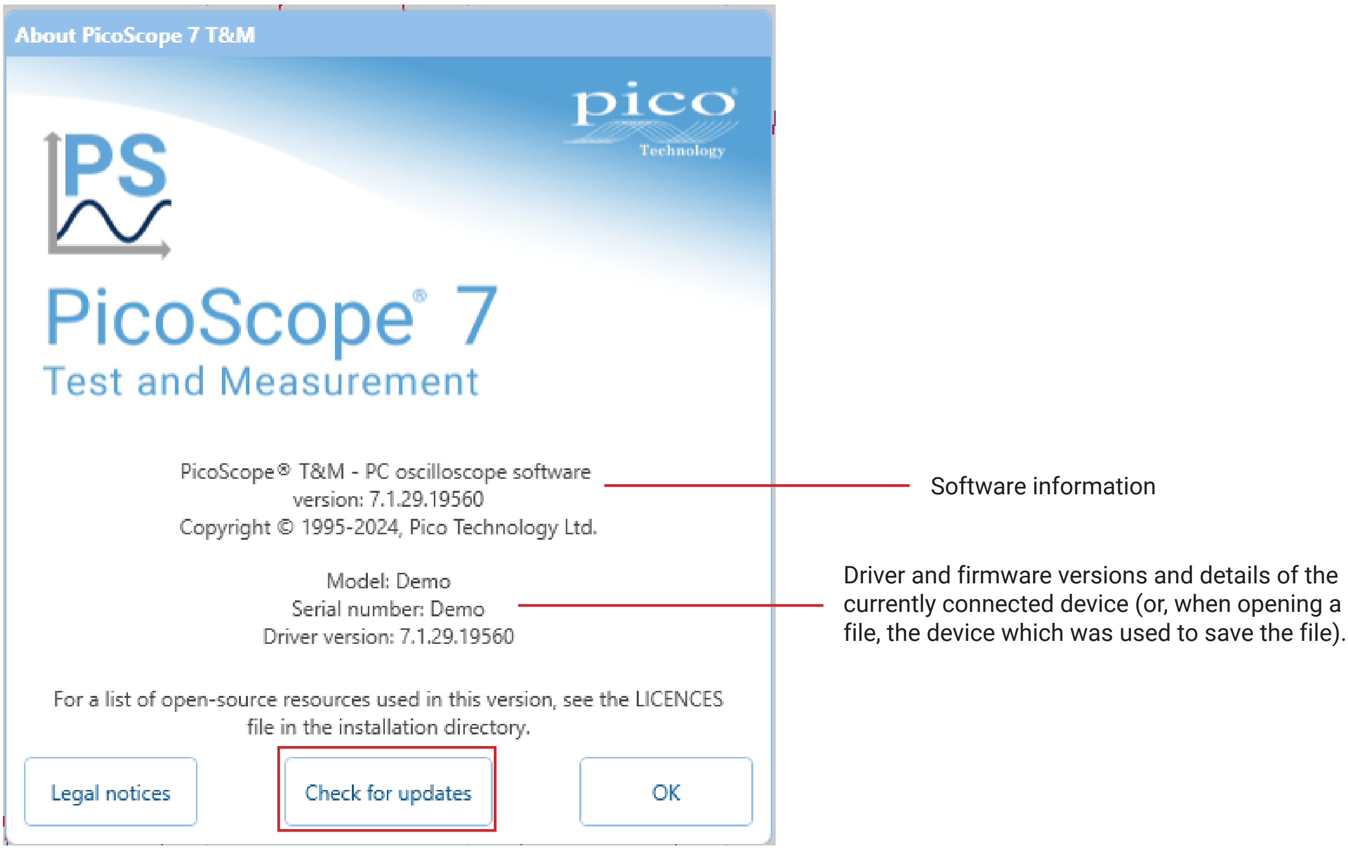
If you are reporting a translation or documentation error, please tell us where the error is, what it says and what you believe it should say.



13.4 About PicoScope dialog

Location: **More... > About PicoScope 7**

Purpose: Allows you to check for updates and other related services

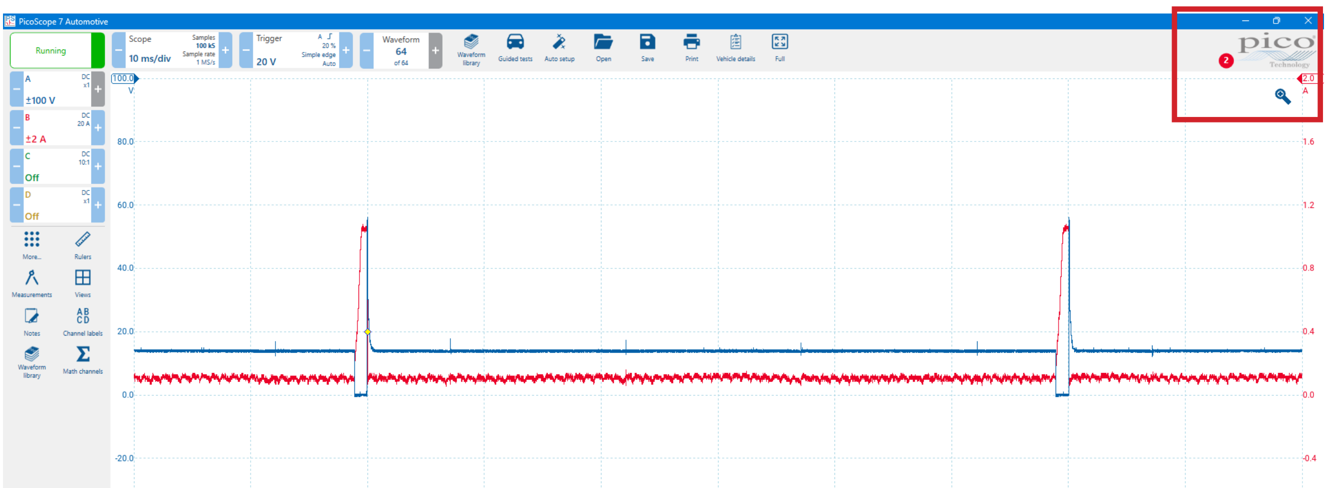


Legal notices	This takes you to our Legal information page
Check for updates	Allows you to check for updates yourself, by clicking the **Check for updates** command

Note: PicoScope 7 will automatically check for software updates.

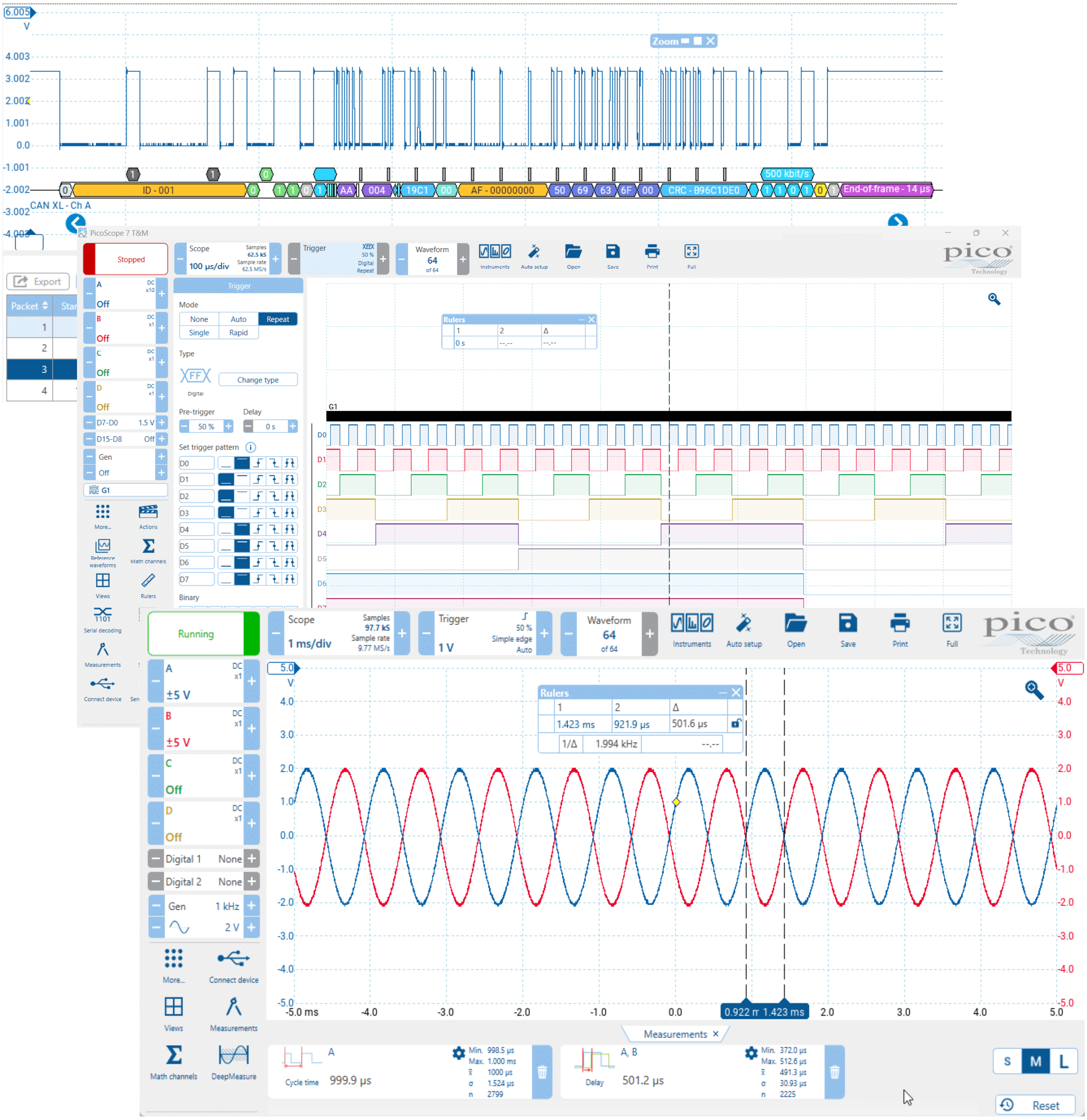
If an update is found:

- You'll see a red circle with a number in it near the Pico Technology logo in the top right corner of the screen.
- Clicking this number will open a window.
- This window will tell you what's new in the update and give you the option to install it.



14. How to...

This How to section of the PicoScope 7 user manual provides a guide to leveraging the software's powerful analytical capabilities. This section moves beyond basic signal capture, delving into the application of mathematical functions and measurements to extract detailed insights from waveforms. You will learn how to create and apply custom functions, perform complex calculations, and utilize built-in measurement tools to analyze signal characteristics with precision. Whether you're aiming to perform frequency analysis, derive custom parameters, or automate measurement tasks, this section equips you with the knowledge to effectively utilize PicoScope 7's function-based analysis tools.



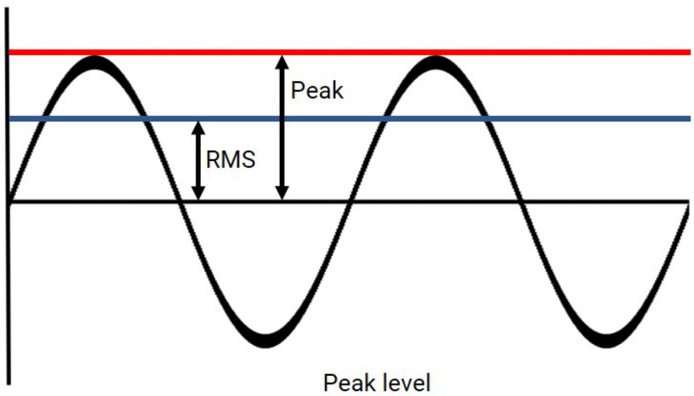
14.1 Math and Measurement

This section covers how to use all the available Math and Measurement functions. Many of these functions are available both as a measurement and as a math channel. As a math channel, they operate on each individual cycle of the waveform, showing any variation over the course of a multi-cycle waveform capture. As a measurement, they show an aggregate value for the whole trace (or section of the trace, when using rulers to set measurement bounds) and the measurement statistics operate on these aggregate values. Both math and measurements versions will be covered in the calculation section.

14.1.1 Crest Factor

The crest factor of a waveform is the ratio of the peak value to the RMS value of the waveform. This indicates how extreme the peaks are versus the effective value of the waveform.

What does crest factor look like:



$$\text{Crest factor} = \frac{\text{Peak level}}{\text{RMS level}} = \text{xxx}$$

Calculation and algorithm:

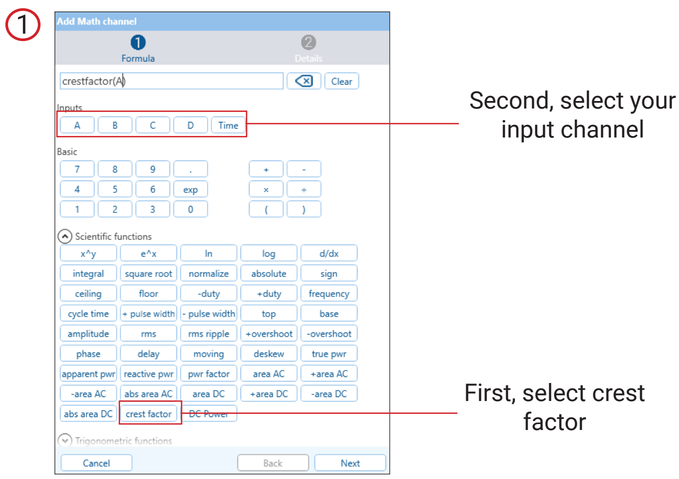
The crest factor is simply a ratio between the peak of a waveform and the RMS value.

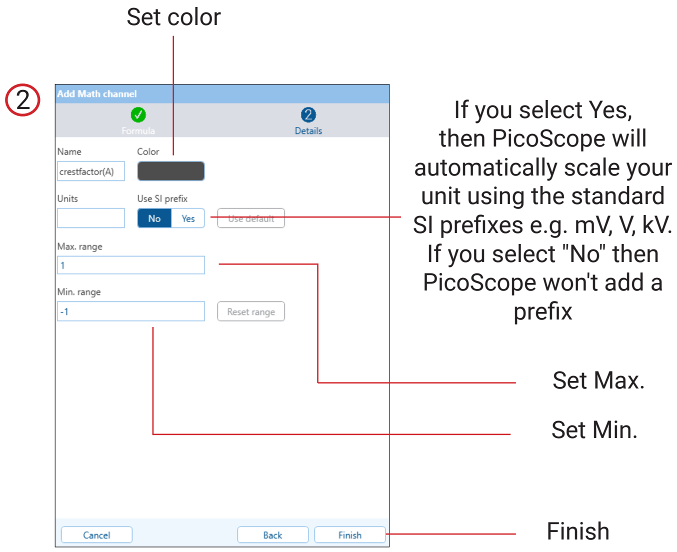
The peak value is calculated by taking both the minimum and maximum values as absolute, then selecting the largest.

Therefore the calculation is to take the absolute value of the peak of the waveform and divide it by the root mean square of the waveform. As a math channel, the trace value indicates the crest factor of each cycle of the input signal. As a measurement, the crest factor is calculated from the peak and RMS values for the whole waveform (or section of waveform to be measured).

Setup:

To access the wizard click the Math channel icon > **Add**

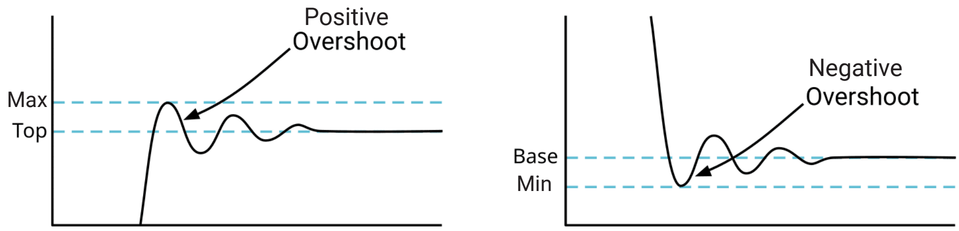




14.1.2 Positive and Negative Overshoot

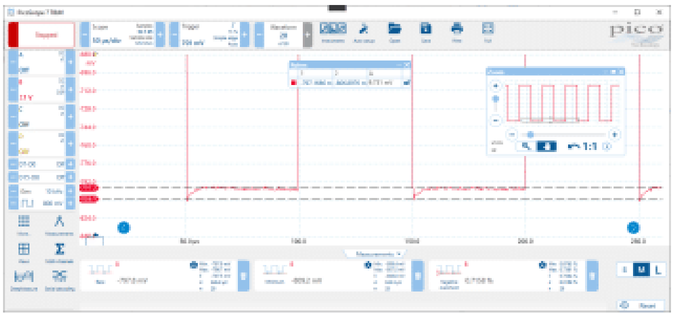
An overshoot is a short term spike or transient which rises above the top, or drops below the base of the waveform. Typically an overshoot occurs at the top of a fast rising edge or at the bottom of a fast falling edge, like we see in a square wave.

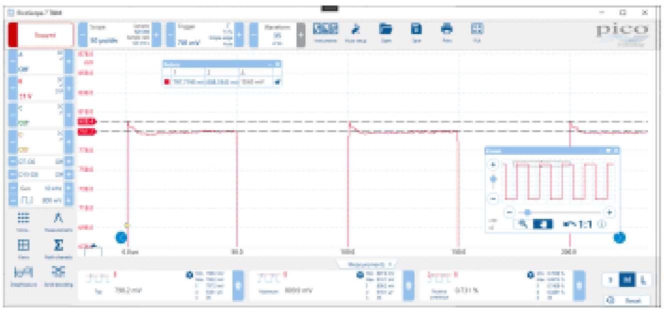
What does overshoot look like:



Interpreting results:

Positive and Negative overshoot measurement results are shown in their own respective measurement lozenge in the Measurements viewport. By default, the measurement is calculated over the whole trace but this can be changed to other options selected in the settings popup which is accessed from the measurement lozenge itself.





Calculation and algorithm:

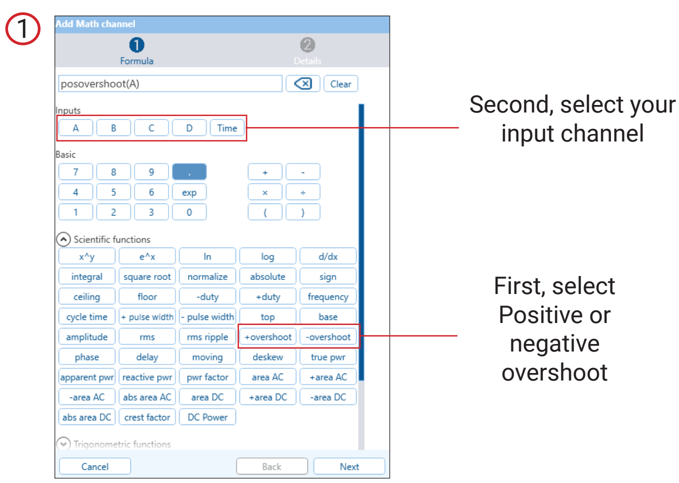
The Positive overshoot measurement is calculated as the difference in the Maximum measurement and the Top measurement as a percentage of the Amplitude, i.e. (Maximum – Top) / (Top – Base) x 100.

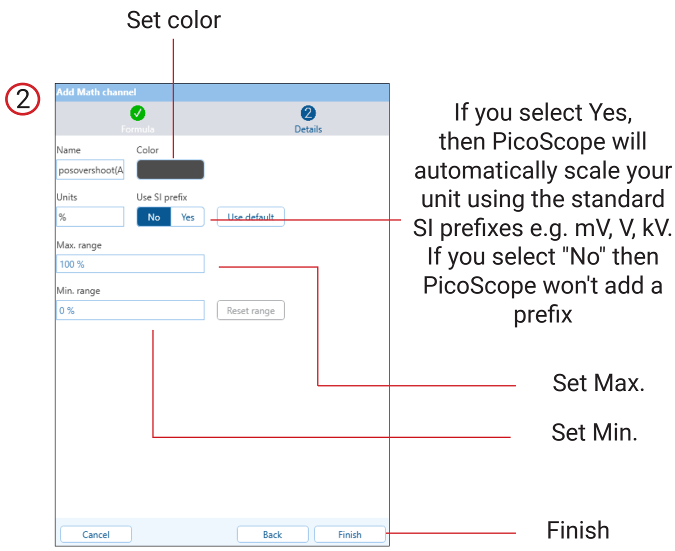
The Negative overshoot measurement is calculated as the difference in the Base measurement and the Minimum measurement as a percentage of the Amplitude, i.e. (Base – Minimum) / (Top – Base) x 100.

As a math channel, the math channel value indicates the overshoot of each cycle of the input signal. As a measurement, the overshoot value for each waveform is the average overshoot of the cycles which were measured.

Setup:

To access the wizard click the Math channel icon > **Add**

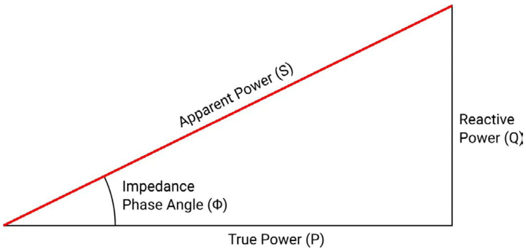




14.1.3 Apparent Power

Apparent power is the total electrical power present in an AC circuit or system. This figure is important as it determines the sizing of electrical infrastructure required to power the load. The apparent power of the system is made up of both active/true power and reactive power components.

What does Apparent power look like:



Interpreting results:

The apparent power measurement result is shown in its own respective measurement lozenge in the Measurements viewport. By default the measurement is calculated over the whole trace but this can be changed to other options selected in the settings popup which is accessed from the measurement lozenge. The measurement lozenge also provides other statistical calculations across buffers about the measurement, e.g. maximum, minimum, standard deviation, mean and number of measurements taken.

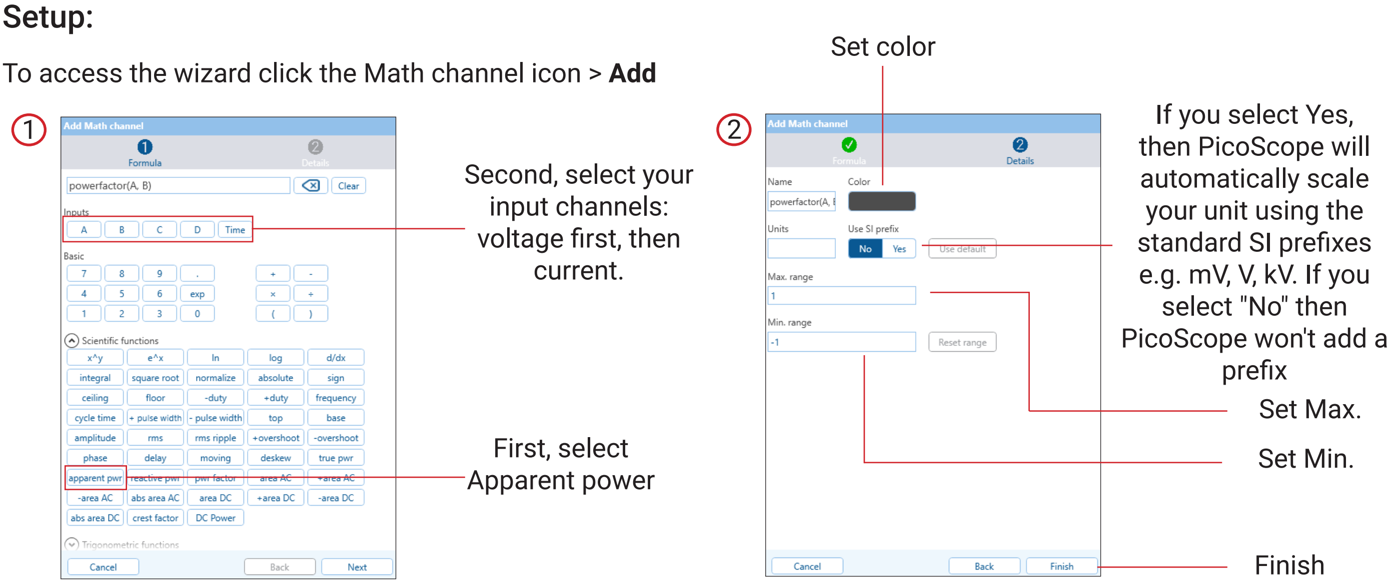
Calculation and algorithm:

The apparent power measurement is calculated as the RMS of one channel, multiplied by the RMS of the second channel, in this example 238.3 * 3.530 = 841.2. The current and voltage values used internally for the calculation are more precise than the four significant figures displayed, hence PicoScope's calculation of apparent power as 841.4 is more accurate than our calculation using the displayed values above.The cycle bounds of the waveform are calculated so as to exclude partial cycles from biassing the output result. For example, in the following example image, if the measurement is set to measure the whole trace, data before -49.14ms and after 49.33ms would be ignored as being incomplete cycles of the waveform. This same rule also applies when the measurement is set to Between Rulers, bounds to measure across will be automatically determined to best fit the cycles of the waveform available between the ruler positions.

As a math channel, the trace value indicates the apparent power of each cycle of the input signal. As a measurement, the apparent power is calculated from the RMS values of voltage and current for the whole waveform (or section of waveform to be measured).

Setup:

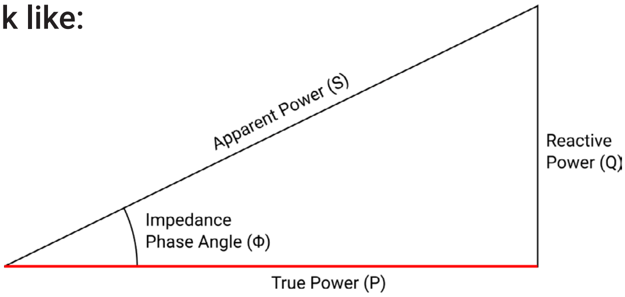
To access the wizard click the Math channel icon > **Add**



14.1.4 True Power

True power plays a crucial role in the efficient operation of electrical networks. To comprehend true power, it's essential to understand its relationship with apparent power and reactive power.

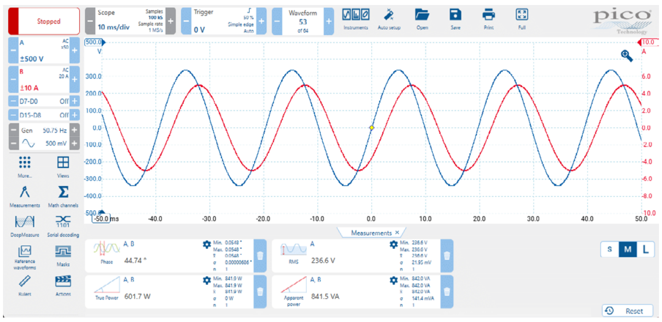
What does True power look like:

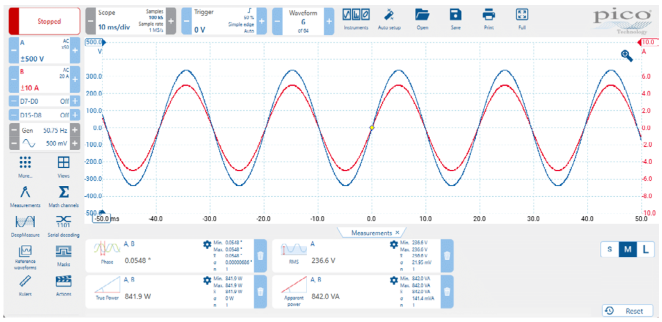


Interpreting results:

True power is measured on a per-cycle basis and then averaged across all cycles in the current buffer. Similarly to other measurements, global statistics are displayed on the Measurements lozenge, which are calculated over all captured buffers. In the two examples below the right-hand one shows the voltage and current are nearly perfectly in-phase and the left-hand image shows them not in phase.

When voltage and current are in-phase, this results in a power factor close to 1. As explained in the *power measurements concepts article*, power factor is the ratio between true and apparent power. Given this the expected true power is roughly equal to the apparent power. Furthermore this is intuitive as a power factor nearing 1, indicates that the true power (utilized power) makes up nearly all of the apparent power (total power).





Calculation and algorithm:

1. Identify primary channel cycles, i.e. ignore samples that are not contained within full cycles. This gives the bounds for rest of the calculation.
2. Multiply the voltage and current samples within the bounds, to get the instantaneous power values.
3. Average the power values where V is the collection of voltage samples, I is the collection of current samples and n is the number of samples, identified within the bounds.

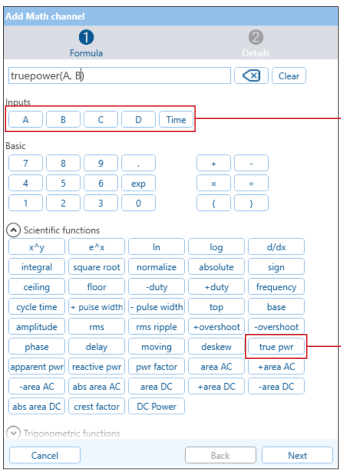
As a math channel, the trace value indicates the true power over each cycle of the input signal. As a measurement, the values for each cycle in the whole waveform (or section of waveform to be measured) are averaged to give the measured value.

Setup:

To access the wizard click the Math channel icon > **Add**

Set color

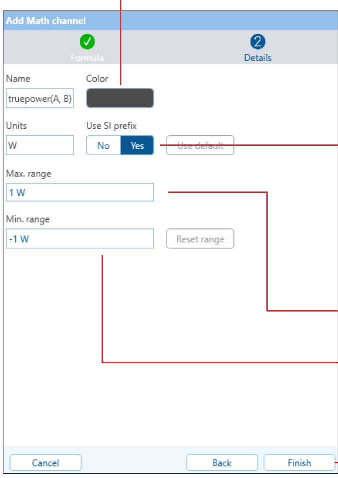
1



Second, select your input channels: voltage first, then current.

First, select True pwr

2



If you select Yes, then PicoScope will automatically scale your unit using the standard SI prefixes e.g. mV, V, kV. If you select "No" then PicoScope won't add a prefix

Set Max.

Set Min.

Finish

14.1.5 Reactive Power

In an AC power transmission system, reactive power is the component of power that oscillates between the source and load without being consumed, whereas true power is the component consumed by the load hence performing useful work. A level of reactive power is necessary in the system to ensure functional operation and to maintain stable voltage levels.

What does Reactive power look like:

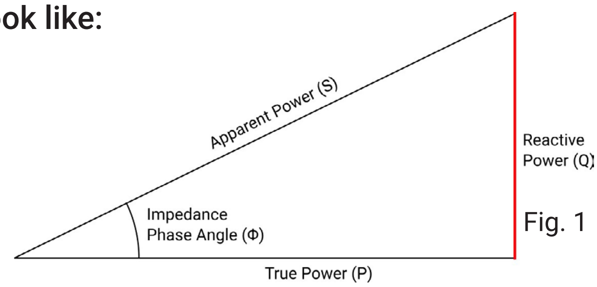
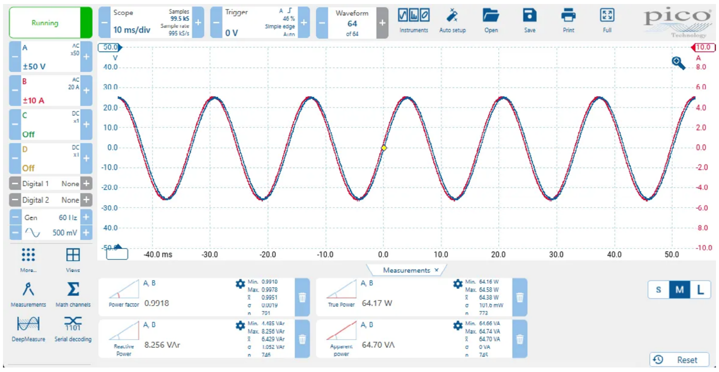


Fig. 1

Interpreting results:

Reactive power is measured on a per-cycle basis and then averaged across all cycles in the current buffer. Similarly to other measurements, global statistics are displayed on the Measurements lozenge, which are calculated over all captured buffers. In the below example, the primary and secondary data sources (voltage and current respectively) are nearly perfectly in-phase. This results in a power factor close to 1, meaning the reactive power will be very low.



Calculation and algorithm:

1. Identify primary channel cycles, i.e. ignore samples that are not contained within full cycles. This gives the bounds for rest of the calculation
2. Calculate the true power average over cycles
3. Calculate the apparent power average over cycles
4. Calculate the reactive power (opposite) within the power triangle (Fig. 1) using the Pythagorean theorem:

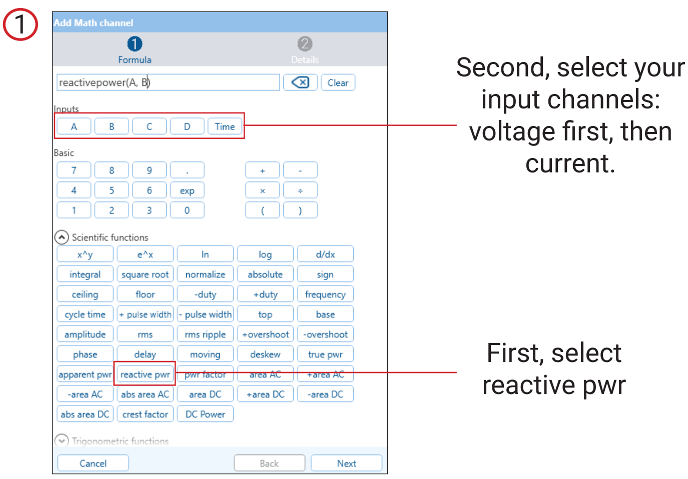
$$Q = \sqrt{S^2 - P^2}$$

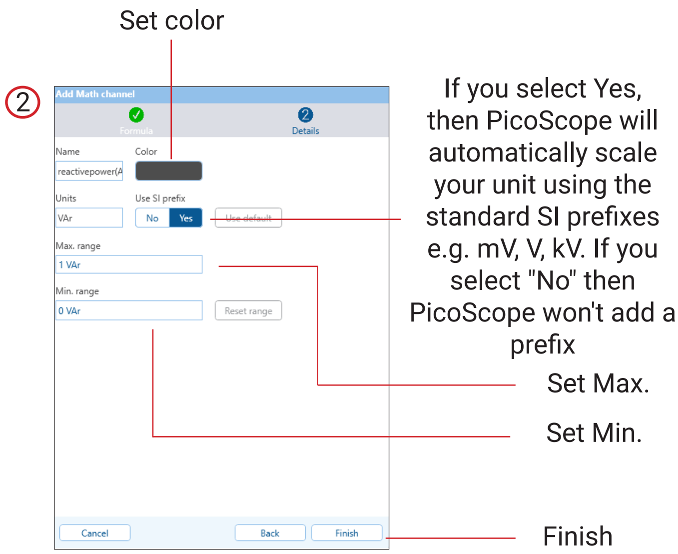
where Q is reactive power, S is apparent power, P is true power.

As a math channel, the trace value indicates the reactive power over each cycle of the input signal. As a measurement, the values for each cycle in the whole waveform (or section of waveform to be measured) are averaged to give the measured value.

Setup:

To access the wizard click the Math channel icon > **Add**



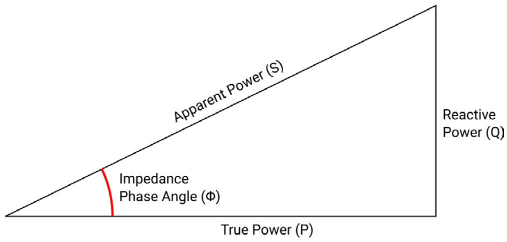


14.1.6 Power Factor

Power factor is the ratio between true and apparent power, which is a measure of an electrical network's operational efficiency. Recall that true power is power converted to useful output, delivered to a load. Apparent power is total electrical power present in an AC circuit or system and consists of both true and reactive power components.

Power factor is calculated using the following formula:

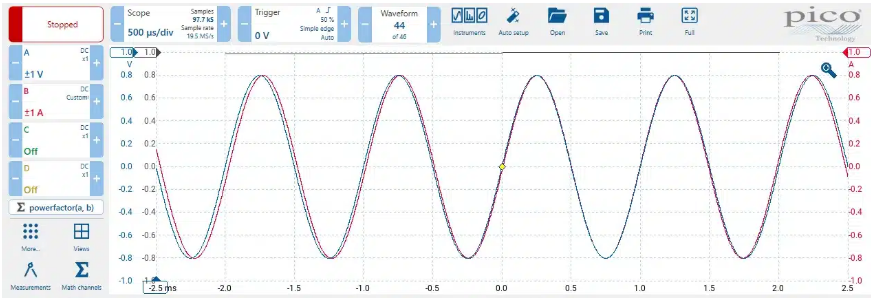
$$PF = \frac{P}{S}$$



where P is true power and S is apparent power.

Interpreting results:

Power factor is measured on a per-cycle basis. The result for each cycle is displayed as a horizontal line overlaid on the graph. In the below example, the primary and secondary data sources (voltage and current) are channels A and B in blue and red respectively. The result is displayed in black.

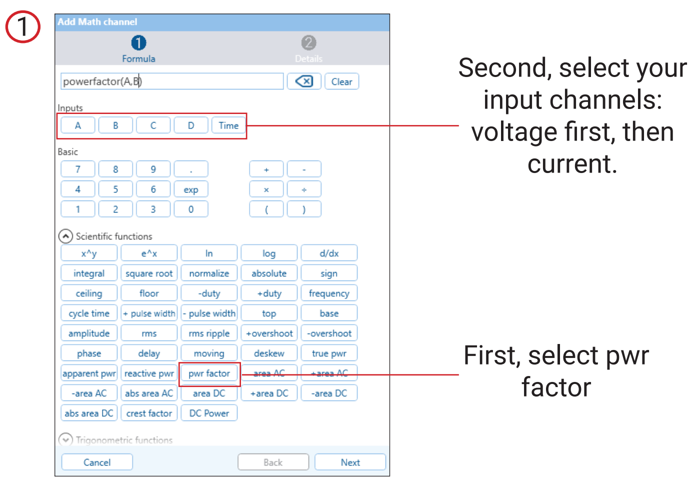


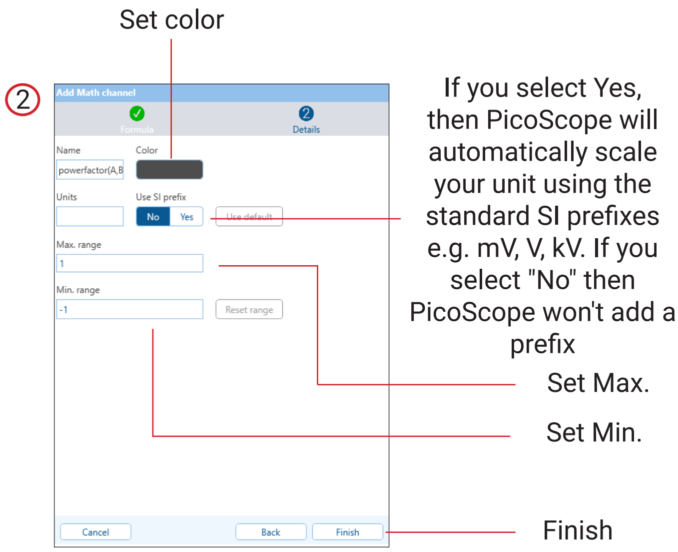
Calculation and algorithm:

1. Identify primary channel cycles, i.e. ignore samples that are not contained within full cycles. This gives the bounds for the rest of the calculation.
2. Calculate the *true power* values
3. Calculate the *apparent power* values
4. Calculate the ratio for true and apparent power, this gives the math channel value per cycle of the input waveform.
5. Take the average of the ratios, for each cycle in the whole waveform (or section of waveform to be measured) to give the measurement value.

Setup:

To access the wizard click the Math channel icon > **Add**

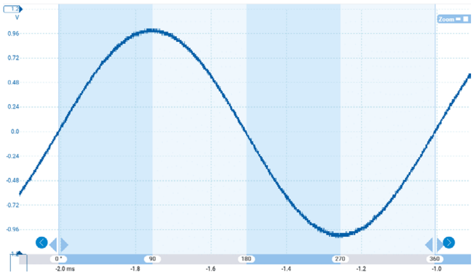




14.1.7 Phase

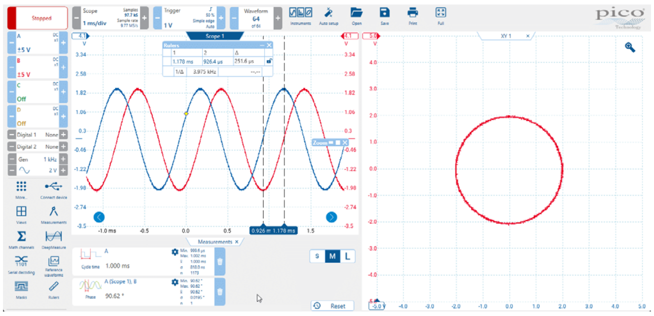
Phase is an angular representation of a point, within a periodic waveform cycle. More specifically a complete cycle/period will be represented by 360° (2π radians or some other rotational unit) and phase is a fraction of this overall value, describing how much of the period has elapsed. The phase measurement or math channel calculates the phase difference between two channels. If the signal is identical on both channels the phase difference is zero, and if the signals are exactly opposite it is 180 degrees.

What does phase look like:



Interpreting results:

Phase is measured on a per-cycle basis and then averaged across all cycles in the current buffer. Similarly to other measurements, global statistics are displayed on the Measurements lozenge, which are calculated over all captured buffers. In the image below, the secondary data source is roughly ¼ out of phase with the primary data source, which given the current output mapping (0-360°) is 90°.



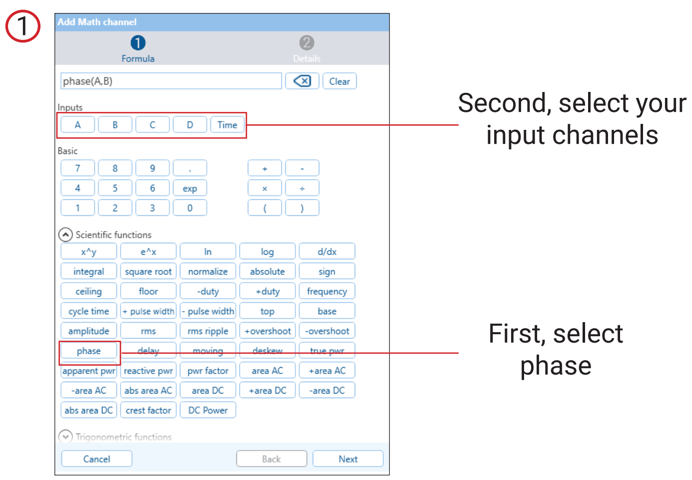
Calculation and algorithm:

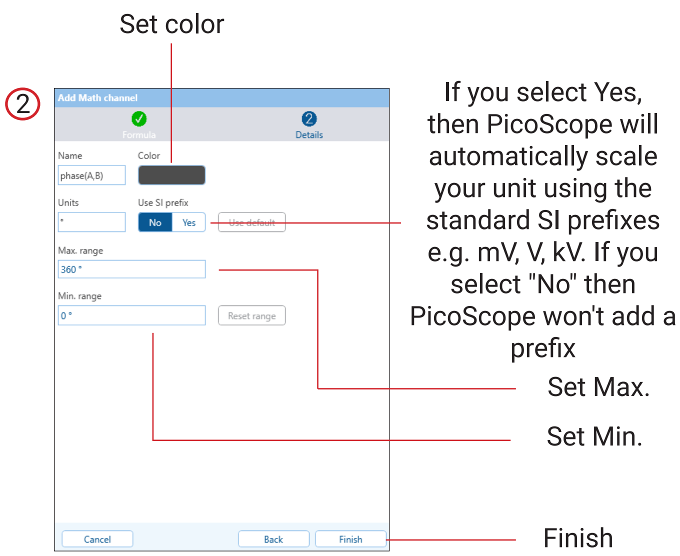
1. Identify primary channel crossing points.
2. Capture the cycle boundaries using either rising or falling crossing points.
3. Capture the secondary channel crossing points, that correspond to the primary channel crossing points (this acts as the corresponding reference point, that is at an equivalent proportion through the cycle when compared to the primary data source's reference point).
4. Map the delay time between corresponding crossing points, to a fraction of the primary channel cycle.
5. Average the phase difference over all cycles and map to desired range. For the 0:360° output range, the following algorithm can be used: $\phi = (t2-t1)/T(360)$ where T is the period and t1 and t2 are the respective data source crossing points.

As a math channel, the trace value indicates the phase difference at each cycle of the input signals, so you can see a varying phase relationship across a waveform. As a measurement, the values for each cycle in the whole waveform (or section of waveform to be measured) are averaged to give the measured value.

Setup:

To access the wizard click the Math channel icon > Add

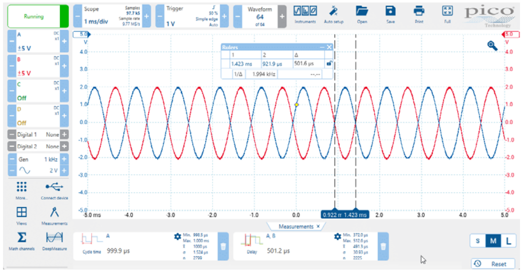




14.1.8 Delay

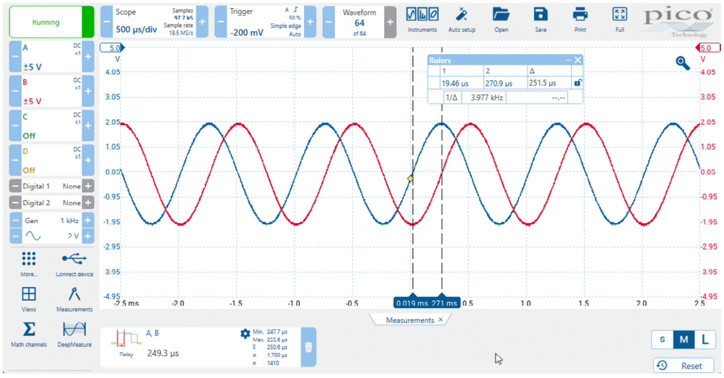
Delay is the measure of time between equivalent reference points on 2 waveforms (usually rising or falling edges). This concept is closely related to phase, but the interval is measured in time, rather than an angle representing how much of the cycle has elapsed.

What does Delay look like:



Interpreting results:

Delay is measured on a per-cycle basis, on either a rising or falling pair of edges, and then averaged across all cycles in the current buffer. Similarly to other measurements, global statistics are displayed on the Measurements lozenge, which are calculated over all captured buffers. In the below example (Fig.4), the secondary data source is out of phase by roughly ¼ of a cycle with the primary data source, and given the current time window, this is calculated as 250 µs.



Calculation and algorithm:

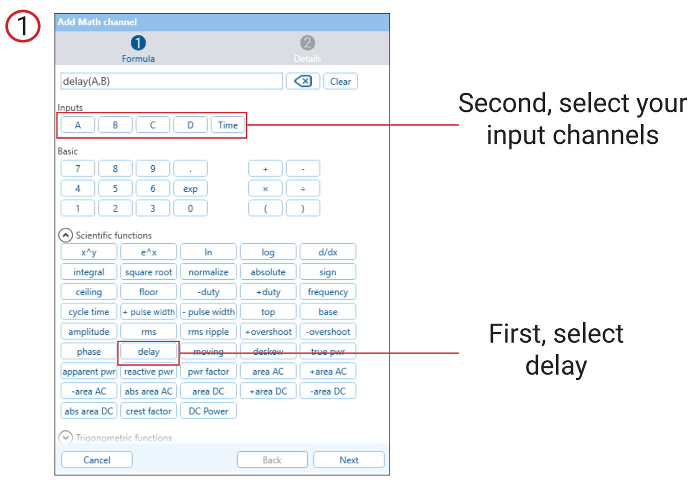
1. Identify primary and secondary channel crossing points
2. Using the crossing points, capture the cycle boundaries for both channels, using either rising or falling crossing points
3. Rising and falling edge counts will be compared and the most numerous edges will be used for the rest of the calculation
4. Calculate the delay for each cycle, using the secondary channel corresponding edge, subtracted from the primary channel edge. This gives the math channel value per cycle of the input waveform
5. Average each of the delay values for each cycle in the whole waveform (or section of waveform to be measured) to give the measurement value.

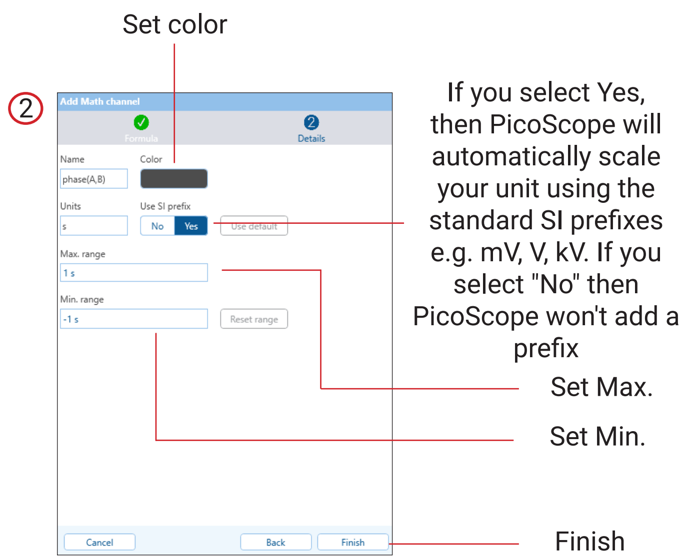
Using the scenario in in the image above, we can calculate the delay for a single pair of rising edges, where the crossing points are marked by the rulers (3rd pair of rising edges). The delay can be calculated using the information in the ruler grid:

270.9 µs – 19.46 µs ≈ 251 µs

Setup:

To access the wizard click the Math channel icon > **Add**

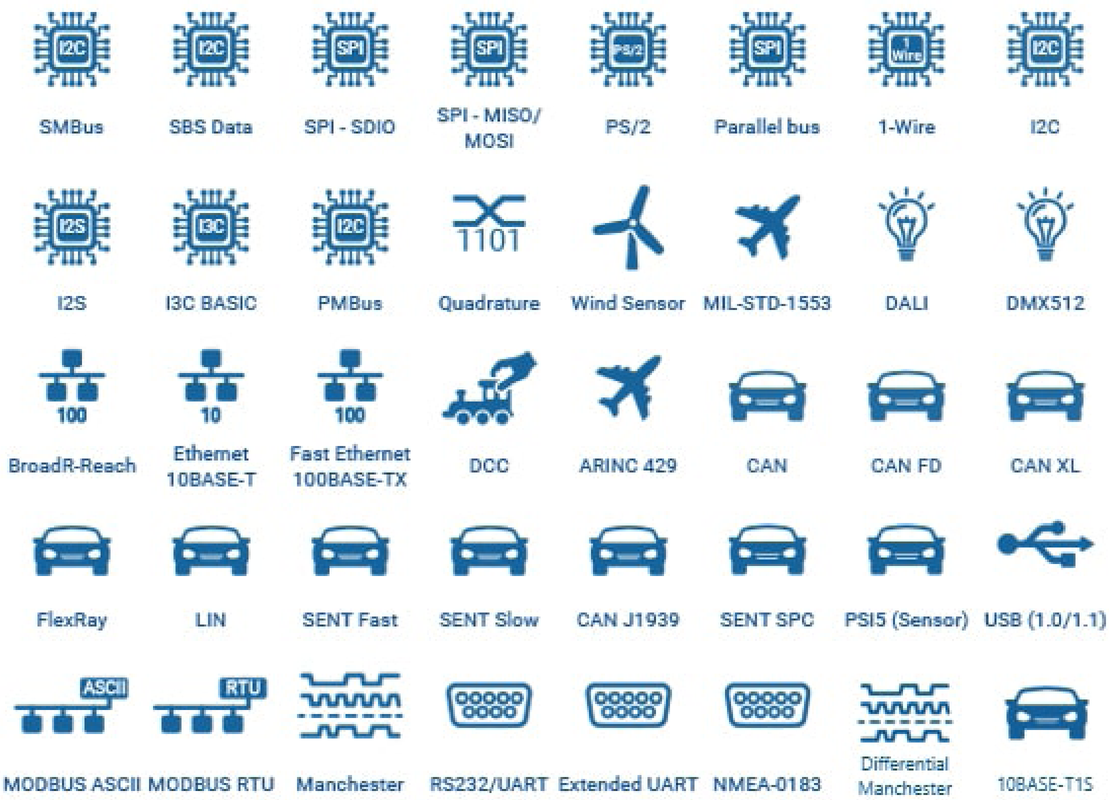




14.2 Serial Decoders

Serial communication buses are used extensively in modern electronic designs. Serial buses offer significant cost advantages and some performance improvements over parallel bus communications. First off, there are fewer signals to route on the board, so PCB costs are lower. Fewer I/O pins on each device are needed, which simplifies component packaging and so reduces component cost. Some serial buses use differential signaling which improves noise immunity.

PicoScope includes decoding and analysis of popular serial standards to help engineers see what is happening in their design to identify programming and timing errors and check for other signal integrity issues. Timing analysis tools help to show performance of each design element, enabling the engineer to identify those parts of the design that need to be improved to optimize overall system performance.

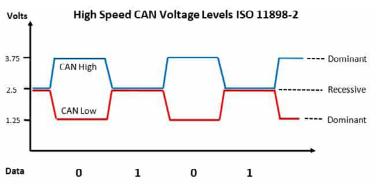


14.2.1 Controller Area Network (CAN)

CAN bus (Controller Area Network) is a serial data standard originally developed in the 1980s by Robert Bosch GmbH for use in automotive applications. Today it is also widely used in industrial process control and aerospace applications.

It allows microcontrollers and electronic devices to communicate with each other without using a host computer and provides fast and reliable data transfer in electrically noisy environments at low cost and with minimal wiring.

What does CAN look like:



Step one – probing

CAN is a differential signal, CAN Low being the inverse of CAN High. Viewing the difference between the two removes any common-mode interference encountered by the signal during transmission.

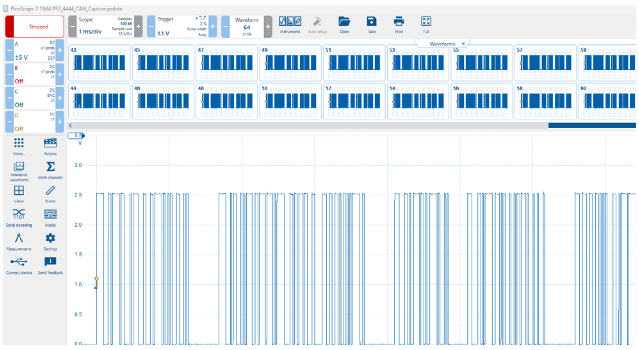
Best results will be achieved by acquiring the signal difference between CAN Low and CAN High using a differential probe or a differential input oscilloscope such as the PicoScope 4444.

The signal can still be acquired using a single-ended probe connected to either CAN Low or CAN high, but any common-mode noise will be displayed and may cause errors in decoding on the oscilloscope which would not affect the CAN receiver.

Step two – acquire the CAN data signal

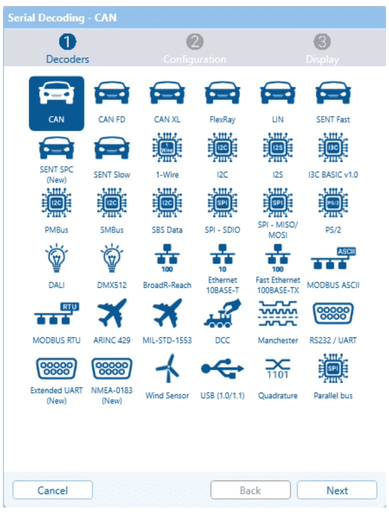
Set memory length to enough to acquire as many frames as required, and with the resolution required to resolve individual bits.

Or use the waveform buffer to capture short bursts of frames while ignoring any dead time in between.



Step three – set up a decoder

1

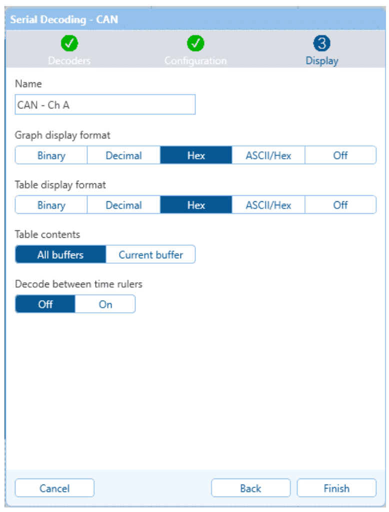


Open Serial decoding from the option panel, then select CAN

2

Select the Data channel (A in this example) and High Or Low if probing single ended, or High if probing differentially.

3



In the Display page keep on or turn off Graph and/or Table display by selecting the required format—Hex/Binary/Decimal/ASCII or Off.

14.2.2 CAN FD

Bosch followed the CAN standard with CAN FD 1.0 or CAN with Flexible Data-Rate, which later became part of the ISO 11898-1:2015 standard. This specification allows for increased data lengths as well as optionally switching to a faster bit rate after the arbitration is decided.

CAN FD meets the growing need to transfer more data, more quickly, in automotive (and other) systems of increasing complexity.

CAN FD is reverse-compatible with existing CAN 2.0 networks so new CAN FD devices can coexist on the same network with classical CAN devices.

Step one – probing

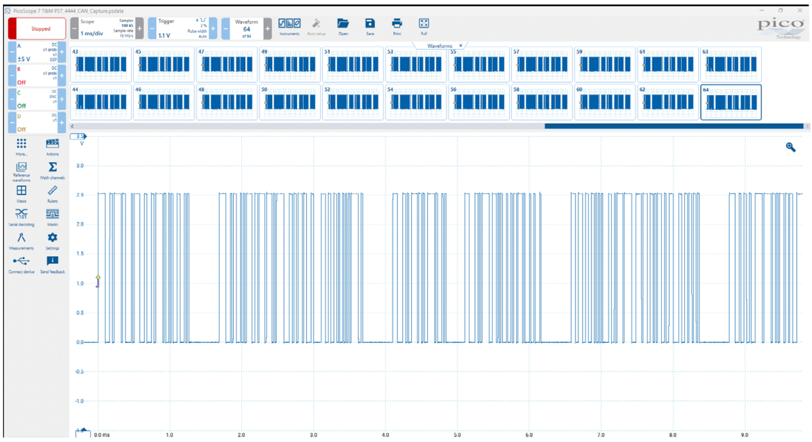
CAN is a differential signal, CAN Low being the inverse of CAN High. Viewing the difference between the two removes any common-mode interference encountered by the signal during transmission.

Best results will be achieved by acquiring the signal difference between CAN Low and CAN High using a differential probe or a differential input oscilloscope such as the PicoScope 4444.

The signal can still be acquired using a single-ended probe connected to either CAN Low or CAN high, but any common-mode noise will be displayed and may cause errors in decoding on the oscilloscope which would not affect the CAN receiver.

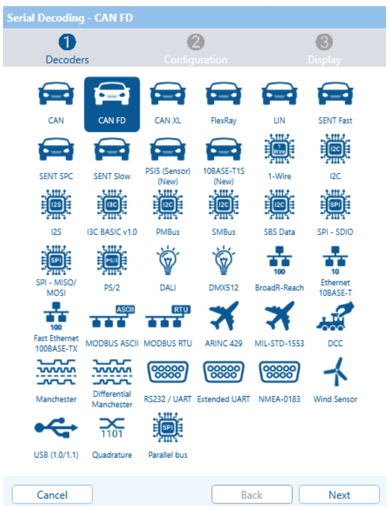
Step two – acquire the CAN data signal

Set memory length to enough to acquire as many frames as required, and with the resolution required to resolve individual bits. Or use the buffer memory index to capture short bursts of frames while ignoring any dead time in between.



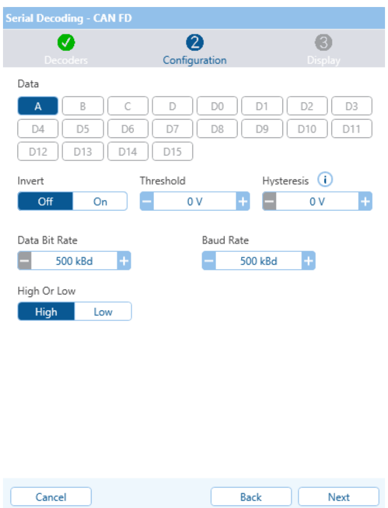
Step three – set up a decoder

1



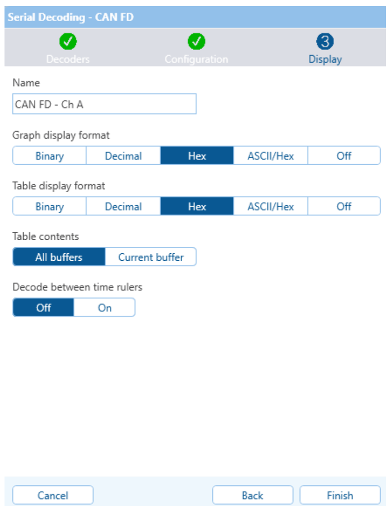
Open Serial decoding from the option panel, then select CAN FD

2



Select the Data channel (A in this example) and High Or Low if probing single ended, or High if probing differentially.

3



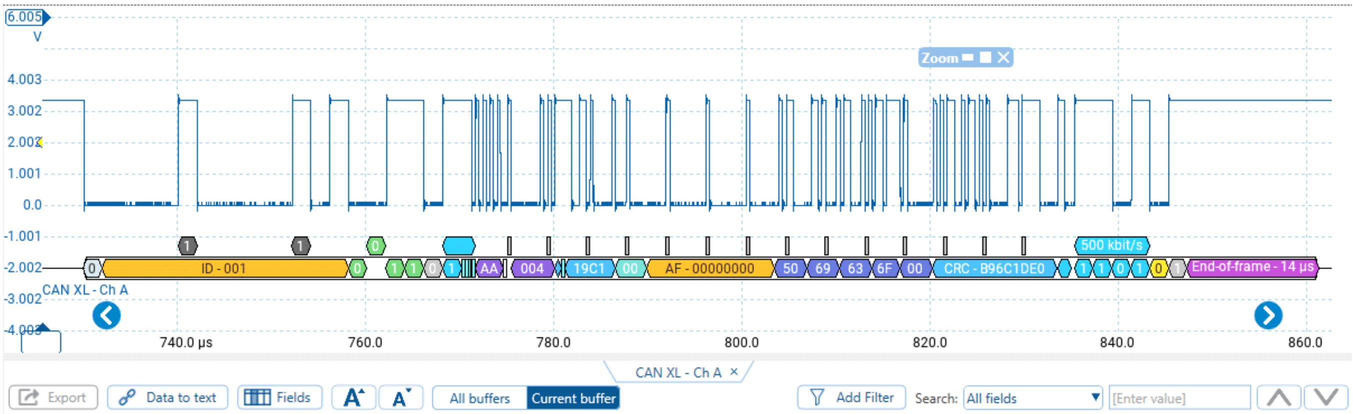
In the Display page keep on or turn off Graph and/or Table display by selecting the required format—Hex/Binary/Decimal/ASCII or Off.

14.2.3 CAN XL

CAN XL is the 3rd generation controller-area network protocol, building on and supporting backward compatibility with Classic CAN and CAN FD networks. It is targeted at in-vehicle networks, connecting multiple controllers and sensors using a single differential-mode bus. Due to the high level of robustness and use of the bus topology with minimal need for wiring, the controller-area network protocols are increasingly finding their way into new industrial applications.

CAN XL provides support for higher data bit rates and longer data payloads than its predecessors, allowing transfer rates up to 20Mbit/s and up to 2048 bytes per frame. In order to support the higher data transfer rates, a new CAN SIC XL transceiver type has been introduced; providing fast signal edges, low ringing, and symmetry required for the higher speed data transfer.

CAN XL can still be used with typical CAN, High Speed CAN or CAN SIC transceivers on mixed-mode buses or if a high bitrate is not required for the application.

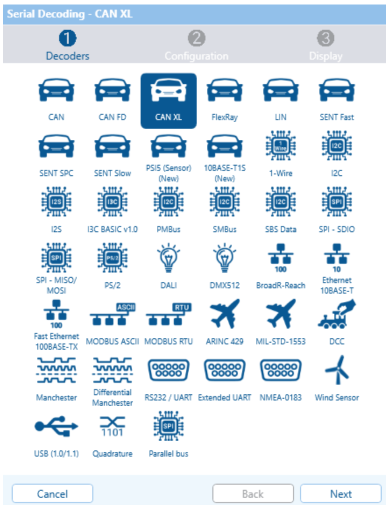


Packet	Start Time	End Time	ID	XLF	DLC	Data	CRC	ACK	CRC Valid	SBC	SBC Parity	IDE	SDT	SEC	VCID	AF	PCRC
1	1.804 ns	176.4 µs	001		5	50 69 63 6F 00	5C 3D	0	✓			0					
2	301.3 µs	507.8 µs	001		5	50 69 63 6F 00	0 06 B5	0	✓	4	0	0					
3	730 µs	861.3 µs	001	1	004	50 69 63 6F 00	B96C1DE0	0	✓	2	1	0	AA	0	00	00000000	19C1
4	1.086 ms	1.217 ms	001	1	004	50 69 63 6F 00	B96C1DE0	0	✓	2	1	0	AA	0	00	00000000	19C1

How to decode CAN XL in PicoScope 7

CAN XL decoding is included in PicoScope 7 as standard. To decode CAN XL waveforms in PicoScope 7, select Serial decoding from the More… options panel to launch the serial decoding dialog.

①

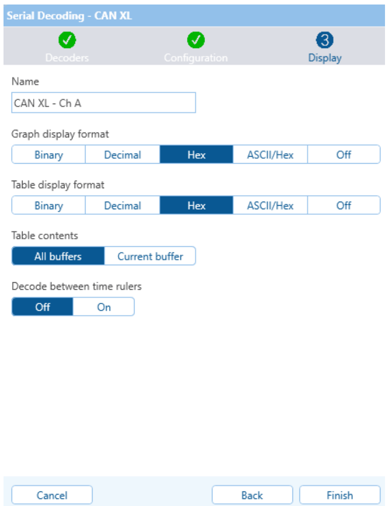


Select CAN XL from the list of available protocols and click 'Next' to proceed.

②

Select the corresponding PicoScope input channel for the CAN XL data signal. The data source may be the CAN XL TXD signal, CAN H, or CAN L from the differential bus.

③



Once the data channel has been chosen and set up, configure the following options as per the CAN XL bus under test.

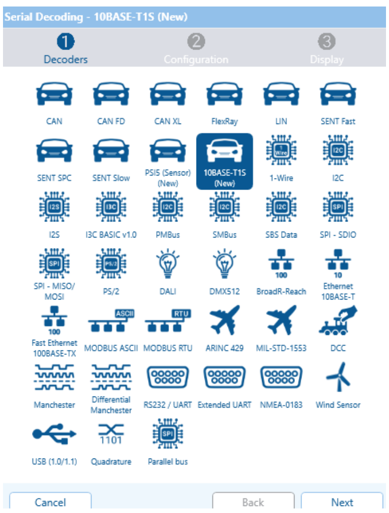
14.2.4 10BASE-T1S

10BASE-T1S is an Ethernet physical layer, standardized by the Institute of Electrical and Electronics Engineers (IEEE), designed to provide 10 Mbps communication over a single twisted pair of wires. This makes it particularly suitable for environments where space and weight constraints are critical, such as in automotive networks and industrial automation systems.

For signaling, it utilizes the Biphase-Mark variant of Differential Manchester Encoding (DME) where the presence or absence of transitions in the middle of the bit period indicates logical values (zero or one). As a result it eliminates any baseline wandering, and data is immune to noise and polarity inversions due to the use of transitions, not absolute voltage levels.

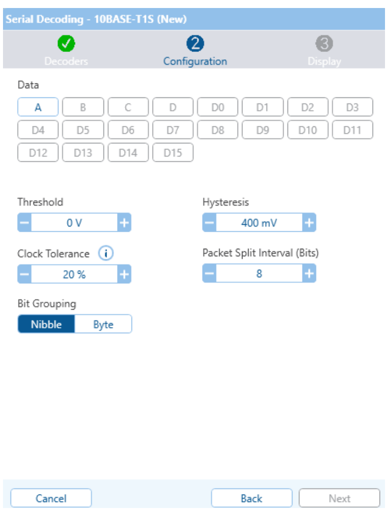
Capturing and analyzing 10BASE-T1S communications with PicoScope

①



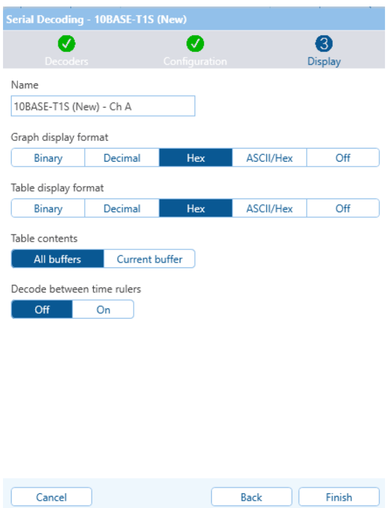
Select 10BASE-T1S from the list of available protocols and click 'Next' to proceed.

②



In the Configuration tab, select the corresponding PicoScope input channel for Data and provide the appropriate values for the relevant 10Base-T1S fields.

③

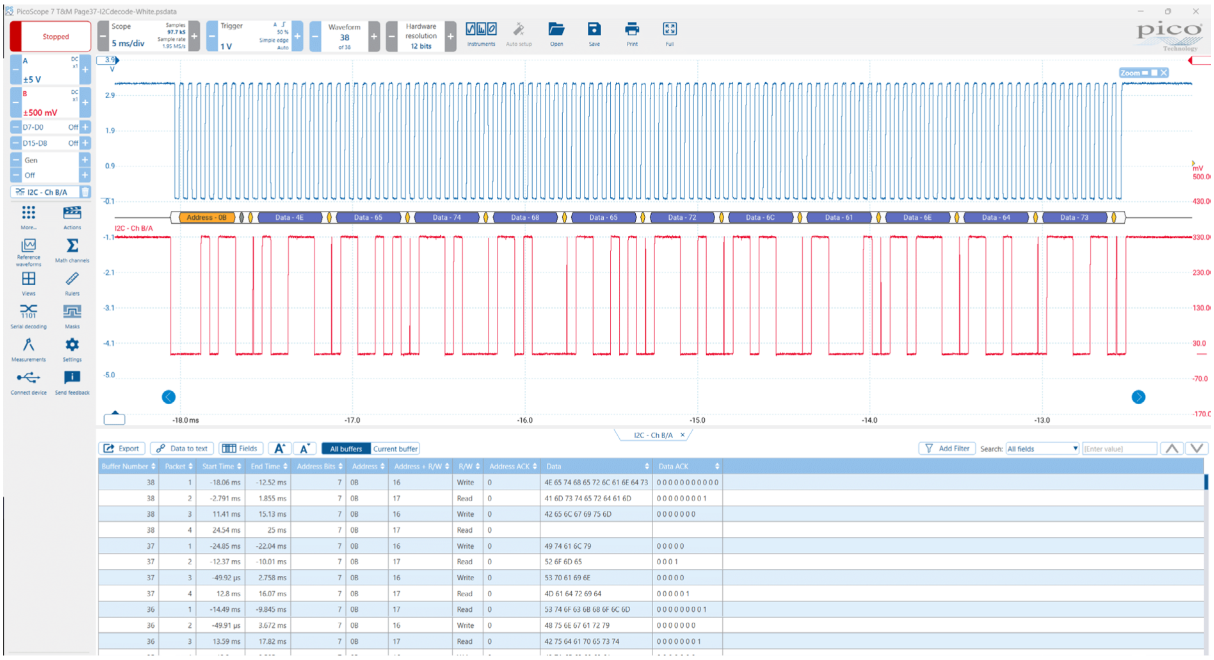


In the Display tab, select the desired Graph and Table display format options to display 10BASE-T1S packets in the appropriate locations.

14.2.5 I²C Decoder

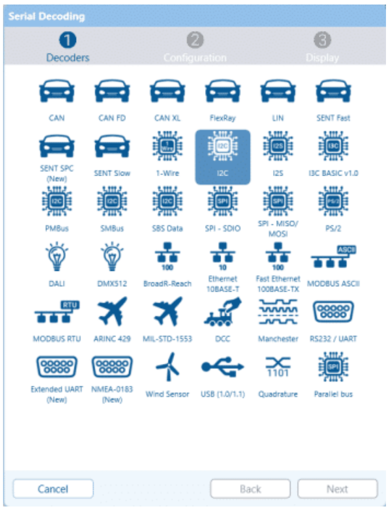
I²C (Inter Integrated Circuit) is a low-speed serial data protocol, commonly used to transfer data between multiple components and modules within a single device.

Developed in the early 1980s by Philips Semiconductors (now NXP), I²C employs two signal wires to transfer "packets" of information between one or more "master" devices such as microcontrollers, and multiple "slave" devices such as sensors, memory chips, ADC and DACs.



I²C decoding in PicoScope

①



Select I²C from the list of available protocols and click 'Next' to proceed.

②

In the Configuration page, select the data and clock channels. The Threshold, Hysteresis will be automatically detected, but can be adjusted along with the Invert toggle and Bus Speed if not standard-mode.

③

On the Display page, give your decoder a name, and set the formats for both Graph and Table displays – Hex/Binary/Decimal/ASCII or Off. Your decoding selection can also be adjusted to a single or all buffers, and/or between rulers if setup.

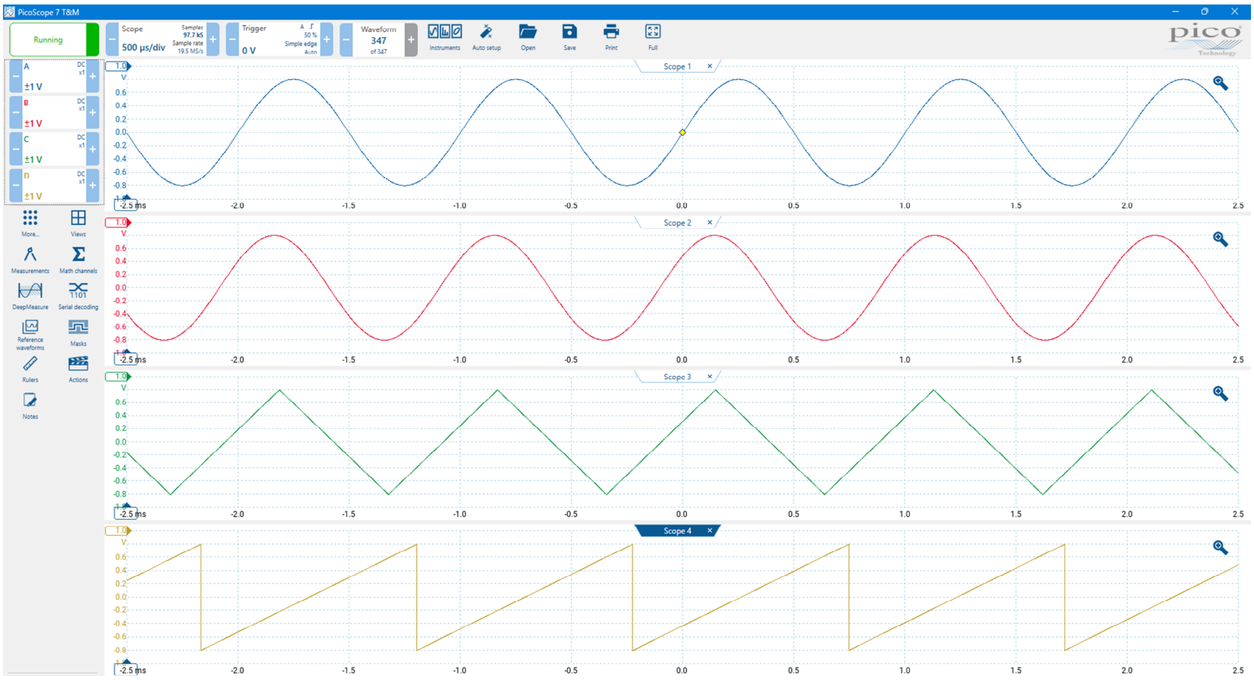
14.3 How to drag and drop a viewpoint

When dragging and dropping viewpoints, it's worth noting the below:

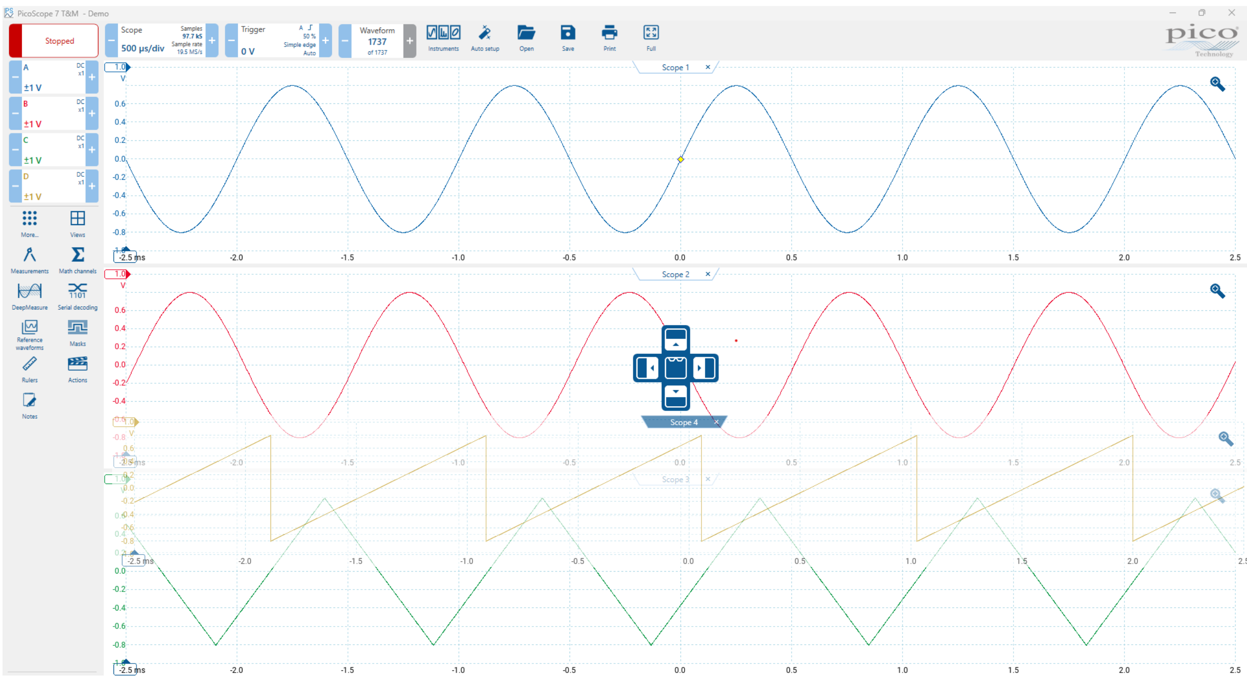
- PicoScope can show up to 8 viewports (2 columns x 4 rows) in one window. To display more than 8 viewports simultaneously, PicoScope will open another window.
- You can pull viewports out into their own windows, and still dock multiple viewports in that window
- You can dock multiple graphs in the same viewport as separate tabs
- Viewports include anything that has the tab at the top, including graphs, measurements, AWG dialog, serial decoding, notes, waveform navigator, etc. All of these can be pulled out and docked in whatever arrangement you like.

You can easily drag a view from one viewport to another. This example shows four view ports, which contain **scope views** labeled **Scope 1** to **Scope 4**. Suppose that you wish to move the **Scope 4** view to the top left view port.

Step 1. Click on the name tab of the **Scope 4 view** and hold down the left mouse button.



Step 2. Drag the tab until the icon in the middle of the screen appears, drop the tab you are dragging onto a zone to dock it above, below, to the left or right of the viewport you're dragging it onto, or drop it on the center zone to dock it as a tab in the same viewport.

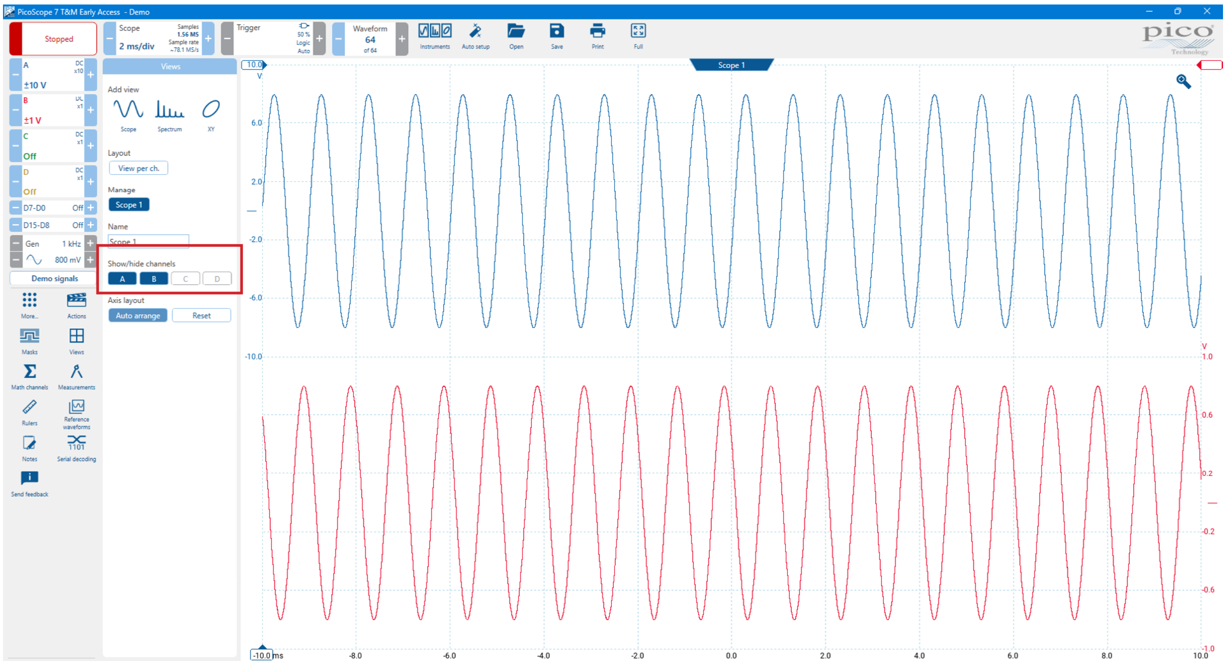


14.4 How to scale and offset a signal

PicoScope offers several ways to change the size and position of a signal during or after capture. These methods apply equally to scope views and spectrum views. They do not change the stored data, only the way in which it is displayed. These options are provided in addition to the analog offset capability of some scopes.

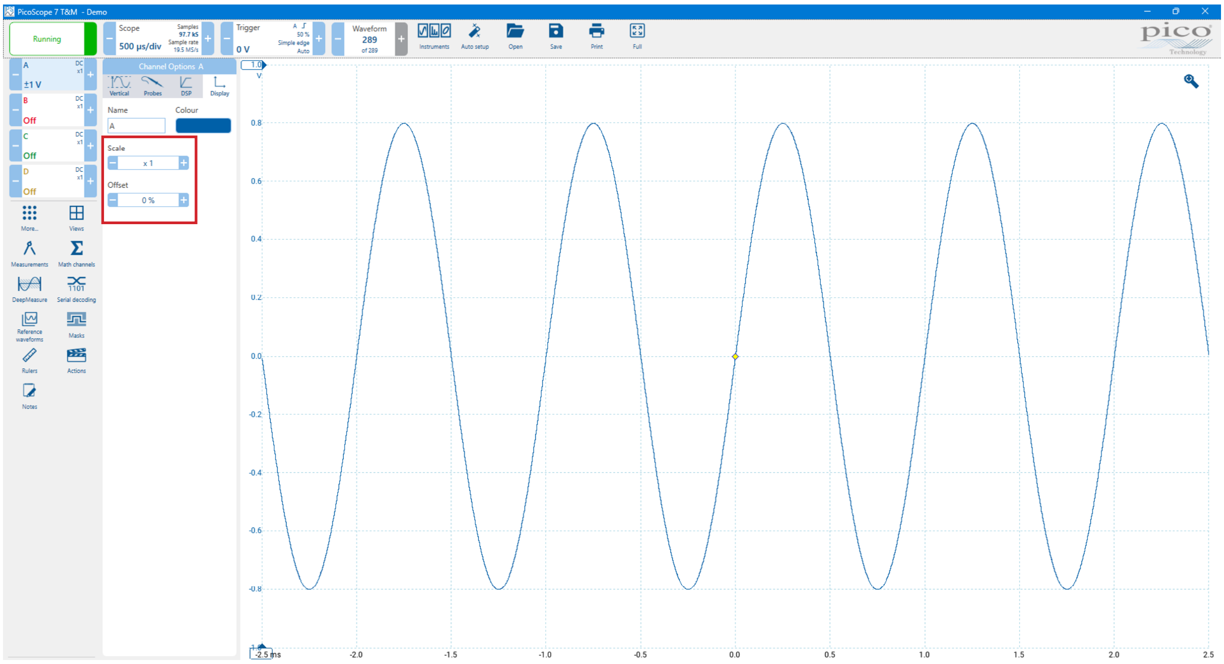
14.4.1 Auto-arrange axes

Open **Views**, and select **Auto-arrange axes**. PicoScope will adjust the displayed height and vertical position of each channel so that the channel axes do not overlap.



14.4.2 Axis scaling and offset

Click the relevant Channel, this will open the Channel Options panel, then select the Display tab. The Scale and Offset options will appear and can be manipulated as required.



Use these tools if **Auto-arrange axes (see above)** does not give you the results you want. It allows you to position channels individually on the view (unlike the **global zooming** and **scrolling** tools, which are applied to all of the channels at the same time).

To adjust the offset without using the **axis scaling** controls, click on the **vertical axis** and drag it up or down. To adjust the scale, you can use the mouse wheel or pinch-zoom on the vertical axis.

How is this different from scaling my data with a Custom Probe?

You can create a **Custom Probe** to apply scaling to the raw data. A **Custom Probe** may change the scale and position of data on the graph but it has a few important differences from the other scaling methods.

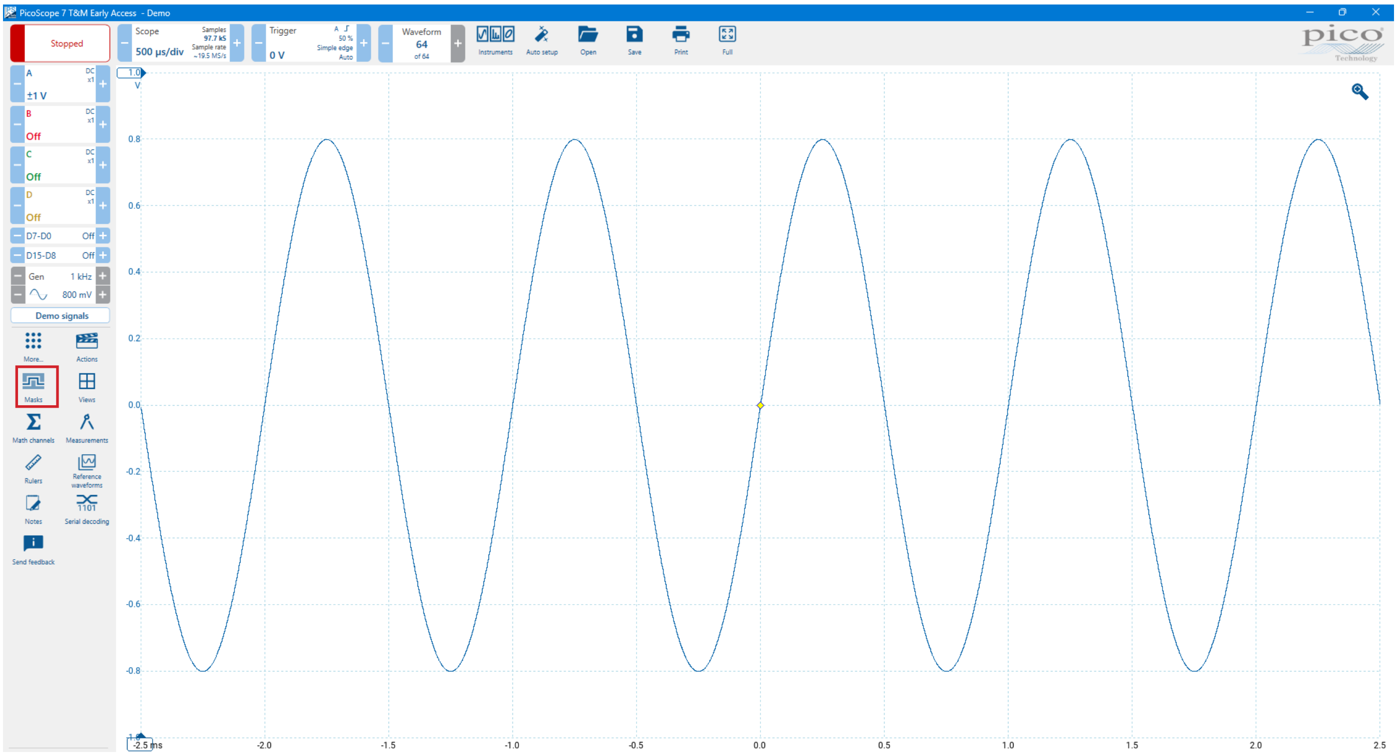
- The actual data values themselves are changed, so the graph axes may no longer display the original input range of the device.
- Custom Probe scaling can be nonlinear and so may alter the shape of the signal.

Custom Probes are useful when you want to represent the characteristics of a physical probe or transducer that you plug into your scope device. All of the **zooming, scrolling, scaling** and **offset** tools still apply to data that has been scaled with a **Custom Probe** in exactly the same way that they would apply to the raw data.

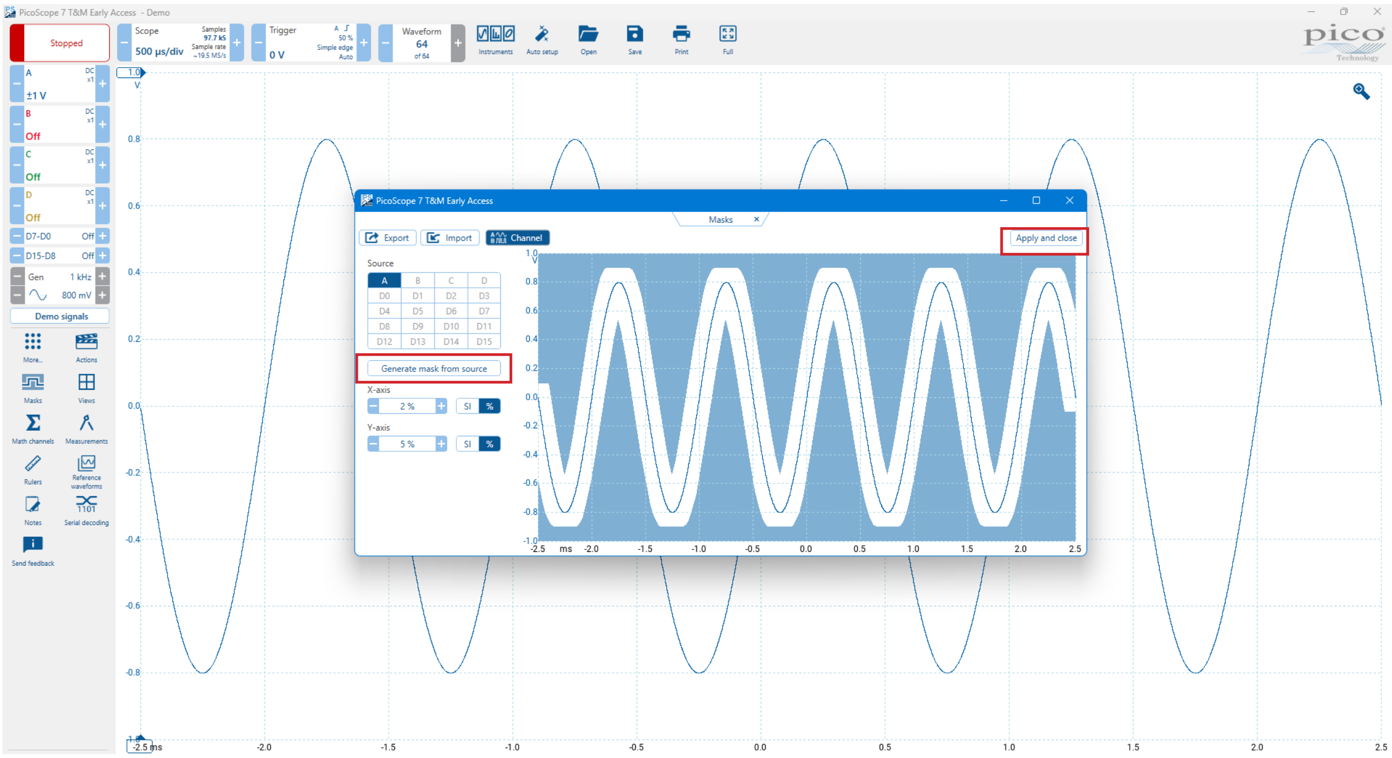
14.5 How to set up a mask limit test

You can carry out a **mask limit test** in a **Scope** and **Spectrum** view. In this example, we will use a **Scope view**.

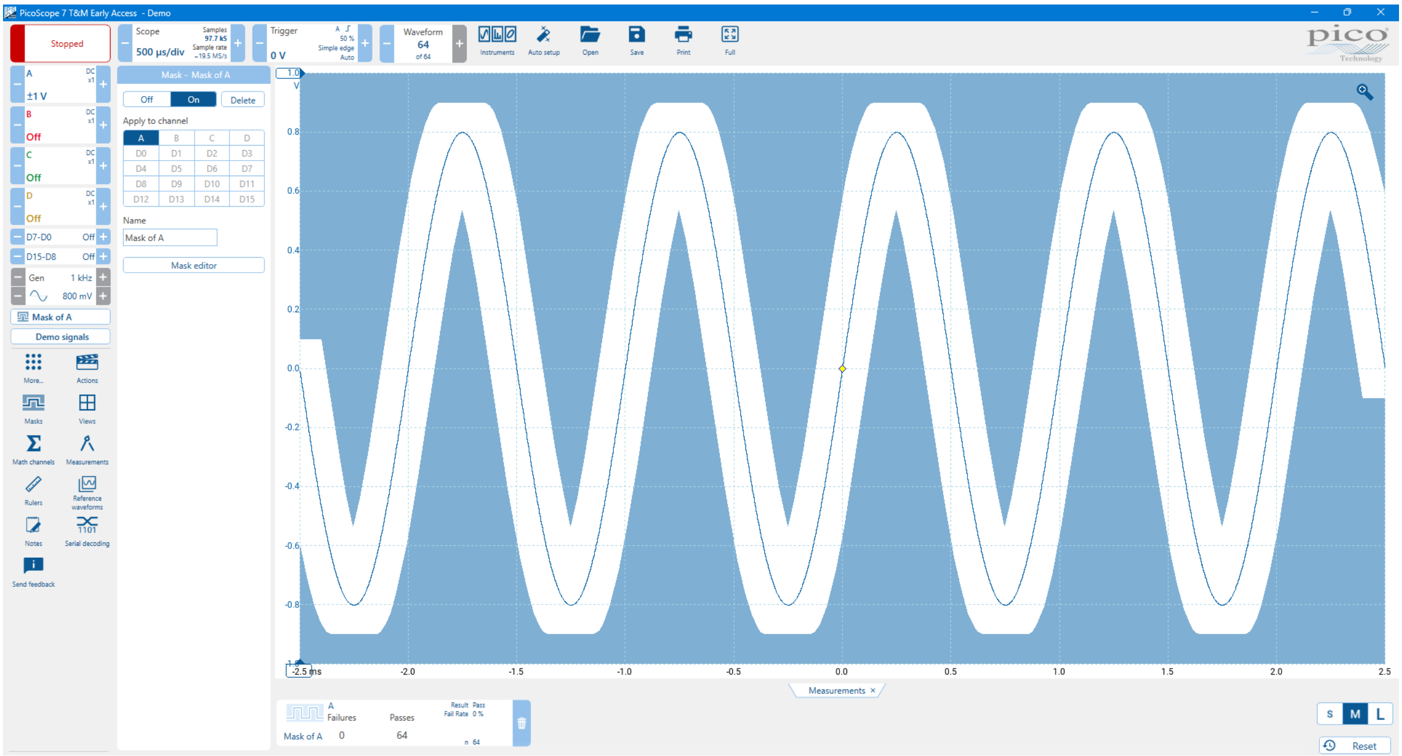
1. Set up the display to show a stable waveform. Adjust the input range and collection time so that the feature of interest fills most of the view. Select **Masks.**



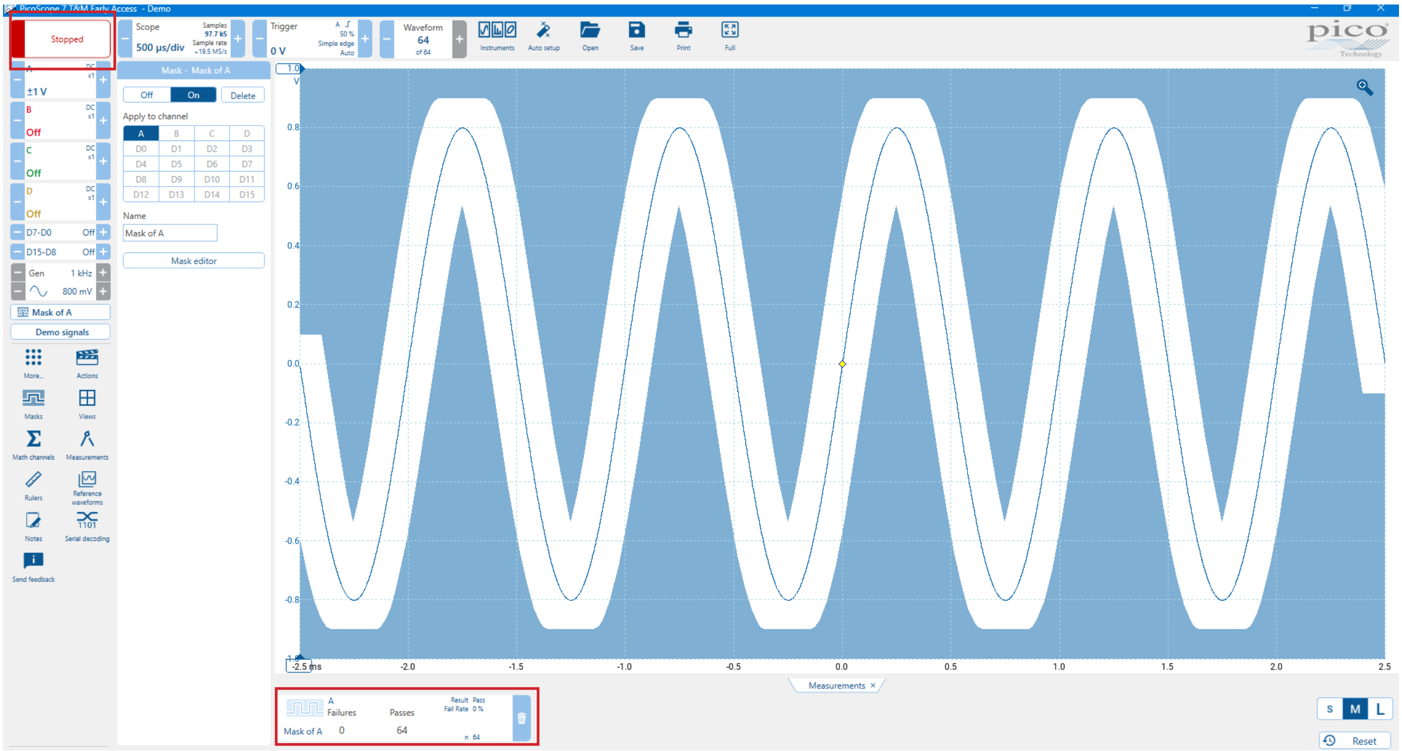
2. The **Mask dialog** will open and Channel A is selected by default, you can change this to any active channel. Click the **Generate mask from source** button to create a mask around the current waveform on that channel. You can use the X-axis and Y-axis controls to adjust the space between the waveform and the mask. As you change the controls, the Mask dialog updates showing a preview of the mask you are creating. Then click **Apply and close.**



3. You now have a mask drawn around the original waveform.



4. PicoScope stops capturing when you enter the **Mask dialog**, so click the **Stop/Running** button to restart. If any captured waveform fails to fit inside the mask, the offending parts are drawn in a contrasting color. The **Measurements** button shows the number of failures:



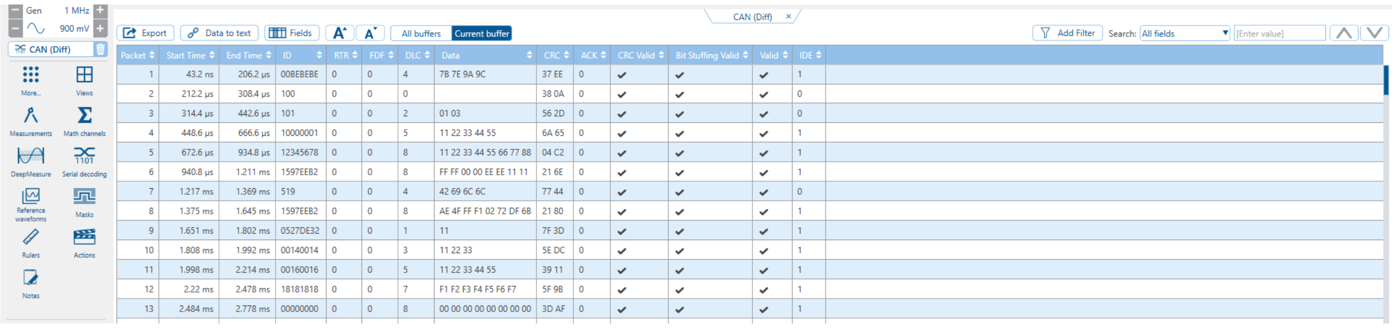
You now have a functioning mask limit test.

14.6 How to create a Data to text file

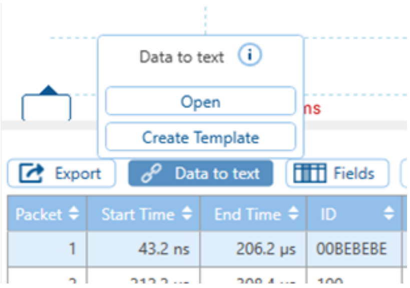
When you perform **Serial decoding**, the **Table** view uses a **Data to text** file to convert numeric data to text. This is useful for displaying decoded packets in a human-readable format.

1. Use the **Serial Decoding dialog** to select the required **serial protocol** and create a **serial decoder**. Make sure the **Table** view is enabled. For more information on creating a Serial decoder click *here*

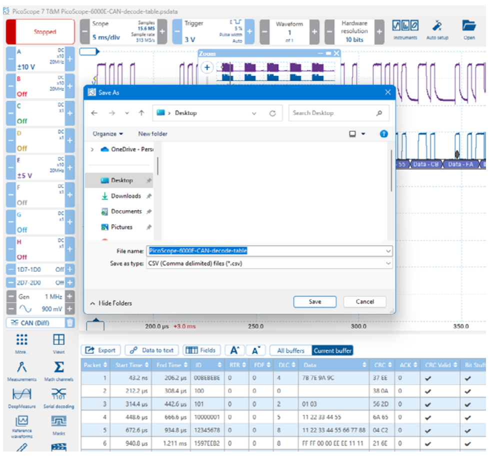
2. PicoScope will show the decoded data in numeric format:



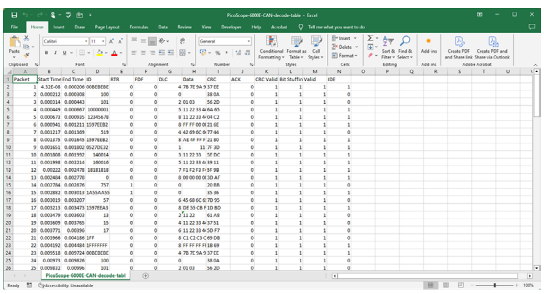
3. To create a **Data to text** file, begin by creating a blank template. Click the **Data to text** button in the **Serial Data Table** toolbar and select **Create Template**:



4. A **Save As** dialog will appear. Choose the file type from the **Save As Type** box (**.ods** for OpenDocument Spreadsheet format, or **.csv** for comma-separated text format) and save the **Data to text** file to a convenient location.



5. Open the **Data to text** file with an appropriate text or spreadsheet editor, and add number/string pairs appropriate to the target system



15. Glossary

AC coupling. In this mode, the scope device rejects very low signal frequencies below about 1 hertz. This allows you to use the full resolution of the scope to measure AC signals accurately, ignoring any DC offset. You cannot measure the signal level with respect to ground in this mode.

AWG. An **arbitrary waveform generator** (**AWG**) is a circuit that can generate a waveform of almost any shape. It is programmed with a data file, supplied by the user, which defines the output voltage at a number of equally spaced points in time. The circuit uses this data to reconstruct the waveform with a specified amplitude and frequency.

Axis. A line marked with measurements. PicoScope shows one vertical axis for each channel that is enabled in a view, giving measurements in volts or other units. Each view also has a single horizontal axis, which is marked in units of time for a scope view, or units of frequency for a spectrum view.

Channel. A scope device has one or more channels, each of which can sample one signal. High-speed scope devices typically have one BNC connector per channel.

Collection Time. The time represented by the horizontal axis of the PicoScope display, when the horizontal zoom control is set to ×1. Also known as the timebase.

CSV. Comma-separated values. A text file containing tabulated data, with columns separated by commas and rows by line-breaks. CSV format is used for importing and exporting PicoScope arbitrary waveform files. You can also export PicoScope waveforms in CSV format. CSV files can be imported into spreadsheets and other programs.

Cycle Time. PicoScope will attempt to find a repeated pattern in the waveform and measure the duration of one cycle.

DC Average. The mean value of the waveform.

DC coupling. In this mode, the scope device measures the signal level relative to signal ground. This shows both DC and AC components.

Dead time. The time between the end of one capture and the start of the next. To obtain the minimum possible dead time, use Rapid trigger mode.

Demonstration (demo) mode. If PicoScope is started when no scope device is plugged in, it allows you to select a "demo device", a virtual scope unit that you can use to test the software. The program is then in demo mode. This mode provides a simulated, configurable signal source for each input channel of the demo device.

Duty Cycle. The amount of time that a signal spends above its mean value, expressed as a percentage of the signal period. A duty cycle of 50% means that the high time is equal to the low time.

Edge Count. Counts the number of rising and falling edges in the signal.

ETS. Equivalent time sampling (ETS) is a method of increasing the effective sampling rate of the scope. In a scope view, PicoScope captures several cycles of a repetitive signal, then combines the results to produce a single waveform with higher time-resolution than a single capture. For accurate results, the signal must be perfectly repetitive and the trigger must be stable.

In focus. PicoScope can display several views, but only one view is in focus at any time. When you click a toolbar button, it will usually affect only the view that is in focus. To bring a view into focus, click on it.

Low Pulse Width. The amount of time that the signal spends below its mean value. This is the average value for all the pulses in a waveform that have a falling edge.

Falling Edge Count. Counts the number of falling edges in the signal.

Falling Rate. The rate at which the signal level falls, in signal units per second.

Fall Time. The time the signal takes to fall from the upper threshold to the lower threshold.

Floating inputs. A feature of the PicoScope 4225 and 4425 oscilloscopes. These are inputs that do not share a common measurement ground. The channels are separated by high impedances and their measurement grounds can be connected to any voltage within the specification range. It is important, however, that every input in use has both a signal and a ground connection.

Frequency. The number of cycles of the waveform per second.

Graticule. The horizontal and vertical dashed lines in every view. These help you estimate the amplitude and time or frequency of features on the waveform.

High Pulse Width. The amount of time that the signal spends above its mean value. This is the average value for all the pulses in a waveform that have a rising edge.

Mask Failures. A special measurement that counts the number of failed waveforms during mask limit testing.

Maximum. The highest level reached by the signal.

Minimum. The lowest level reached by the signal.

MSO. **Mixed-signal oscilloscope**. An instrument that captures and displays analog and digital signals on the same horizontal scale.

PC Oscilloscope. A measuring instrument consisting of a scope device and the PicoScope software running on a PC. A PC Oscilloscope has the same functions as a traditional bench-top oscilloscope but is more flexible and cost-effective.

Peak To Peak. The difference between **maximum** and **minimum**. It is equivalent to a ripple measurement.

Probe. An accessory that attaches to your oscilloscope and picks up a signal to be measured. Probes are available to pick up any form of signal, but they always deliver a voltage signal to the oscilloscope. PicoScope has built-in definitions of standard probes, but also allows you to define custom probes.

Resolution enhancement. Collecting samples at a faster rate than requested, then combining the excess samples by averaging. Resolution enhancement can increase the effective resolution of a scope device when there is a small amount of noise on the signal.

Rise Time. The time the signal takes to rise from the lower threshold to the upper threshold.

Rising Edge Count. Counts the number of rising edges in the signal.

Rising Rate. The rate at which the **signal level** rises, in signal units per second.

RMS. The root mean square value of the waveform, including the DC component.

RMS ripple. The root mean square (RMS) value of the waveform minus the DC Average.

Ruler. A vertical or horizontal dashed line that can be dragged into place on a waveform in a view. PicoScope displays the signal level, time value or frequency value of all rulers in the Ruler Legend.

Scope device. The box from Pico Technology that you plug into the USB or parallel port of your computer. With the help of the PicoScope software, the scope device turns your computer into a PC Oscilloscope.

Progressive mode. Normally, PicoScope redraws the waveform in a scope view many times every second. However, at collection times longer than 200 ms/div (adjustable on the Sampling tab of the Scope options panel), it switches to progressive mode. In this mode, PicoScope updates the scope view continuously as each capture progresses, rather than waiting for a complete capture before updating the view.

Standard deviation. A statistical measure of the spread of a set of samples. The standard deviation of the set is defined as:

$$SD = \sqrt{\frac{1}{(n-1)}\sum_{i=0}^{n-1}(y_i - \bar{y})^2}$$

where $\bar{y}$ is the arithmetic mean of all the samples. The units of the standard deviation value are the same as those of the original samples.

Trigger. The part of an oscilloscope that monitors an incoming signal and decides when to begin a capture. Depending on the trigger condition that you set, the scope may trigger when the signal crosses a threshold, or may wait until a more complex condition is satisfied.

Vertical resolution. The number of bits that the scope device uses to represent the signal level. This number depends on the design of the device, but can be boosted in some cases by using resolution enhancement.

View. A presentation of data from a scope device. A view may be a scope view, an XY view or a spectrum view.

Viewport. The views in the PicoScope window are arranged in a grid, and each rectangular area in the grid is called a viewport.

UK global headquarters:

☎ +44 (0) 1480 396 395
✉ sales@picotech.com

Pico Technology
James House
Colmworth Business Park
St. Neots
Cambridgeshire
PE19 8YP
United Kingdom

North America regional office:

☎ +1 800 591 2796
✉ sales@picotech.com

Pico Technology
320 N Glenwood Blvd
Tyler
TX 75702
United States

Germany regional office and
EU Authorised Representative:

☎ +49 (0) 5131 907 62 90
✉ info.de@picotech.com

Pico Technology GmbH
Emmericher Str. 60
47533 Kleve
Germany

Asia-Pacific regional office:

☎ +86 21 2226-5152
✉ pico.asia-pacific@picotech.com

www.picotech.com

www.picotech.com/tech-support