



# PicoScope<sup>®</sup> 6

PC Oscilloscope Software

User's Guide



# Table of Contents

1 Welcome .....	1
2 Introduction .....	4
1 Legal statement .....	4
2 Updates .....	4
3 Trademarks .....	4
4 System requirements .....	5
3 Using PicoScope for the first time .....	6
4 PicoScope and oscilloscope primer .....	7
1 Oscilloscope basics .....	7
2 PC Oscilloscope basics .....	8
3 PicoScope basics .....	9
1 Capture modes .....	10
4 PicoScope window .....	12
5 Scope view .....	13
6 Overrange indicator .....	13
7 MSO view .....	14
1 Digital view .....	15
2 Digital context menu .....	15
8 XY view .....	16
9 Trigger marker .....	17
10 Time-delay arrow .....	17
11 Spectrum view .....	18
12 Persistence mode .....	19
13 Measurements table .....	20
14 DeepMeasure™ .....	21
15 Pointer tool tip .....	22
16 Signal rulers .....	22
17 Time and frequency rulers .....	23
18 Phase (rotation) rulers .....	24
19 Ruler settings .....	26
20 Ruler legend .....	27
21 Frequency legend .....	27
22 Properties sheet .....	28
23 Custom probes .....	29
24 Math channels .....	30
25 Reference waveforms .....	31
26 Serial decoding .....	32
27 Mask limit testing .....	33
28 Alarms .....	34
29 Buffer Overview .....	34

5 Menus .....	36
1 File menu .....	36
1 Connect Device dialog .....	38
2 Open dialog .....	40
3 Save As... dialog .....	41
4 Start-up Settings menu .....	46
5 Waveform Library Browser .....	47
2 Edit menu .....	48
1 Notes .....	49
2 Details dialog .....	50
3 Views menu .....	52
1 Custom grid layout dialog .....	53
4 Measurements menu .....	54
1 Add/Edit Measurement dialog .....	55
2 DeepMeasure™ dialog .....	58
5 Tools menu .....	61
1 Custom Probes dialog .....	63
2 Math Channels dialog .....	78
3 Reference Waveforms dialog .....	90
4 Serial Decoding dialog .....	92
5 Alarms dialog .....	98
6 Masks menu .....	100
7 Macro Recorder .....	103
8 Preferences dialog .....	104
6 Automotive menu .....	119
7 Help menu .....	120
1 Send Feedback dialog .....	121
6 Toolbars and buttons .....	122
1 Capture Setup toolbar .....	122
1 Spectrum Options dialog .....	124
2 Persistence Options dialog .....	126
2 Buffer Navigation toolbar .....	128
3 Zooming and Scrolling toolbar .....	129
1 Zoom Overview window .....	130
4 Channels toolbar (PicoScope devices) .....	131
1 Channel controls (PicoScope devices) .....	131
2 Signal Generator button .....	143
5 PicoLog 1000 Series Channels toolbar .....	151
1 PicoLog 1000 Series Digital Outputs dialog .....	151
6 USB DrDAQ Channels toolbar .....	152
1 USB DrDAQ Signal Generator dialog .....	153
2 USB DrDAQ RGB LED control .....	154

3 USB DrDAQ Digital Outputs dialog .....	155
7 Trigger toolbar .....	156
1 Stop and Go buttons .....	156
2 Trigger controls .....	156
3 Measurements buttons .....	166
4 Other buttons .....	166
7 How to... .....	167
1 How to change to a different device .....	167
2 How to use rulers to measure a signal .....	167
3 How to measure a time difference .....	169
4 How to move a view .....	170
5 How to scale and offset a signal .....	172
6 How to set up the spectrum view .....	175
7 How to find a glitch using persistence mode .....	176
8 How to set up a mask limit test .....	179
9 How to save on trigger .....	182
10 How to create a link file .....	187
11 How to convert files in Windows File Explorer .....	189
8 Reference .....	191
1 Command-line syntax .....	191
2 Flexible power .....	194
3 Hydraulic waveforms: Examples and analysis .....	196
1 Hydraulic ram on a tractor's three-point hitch .....	196
2 Hydraulic system on telehandler .....	197
3 Hydraulic clutch pack .....	200
4 Measurement types .....	202
1 Scope measurements .....	202
2 Spectrum measurements .....	203
5 Serial protocols .....	205
1 1-Wire protocol .....	206
2 ARINC 429 protocol .....	206
3 BroadR-Reach .....	206
4 CAN protocol .....	207
5 CAN FD protocol .....	207
6 DALI protocol .....	208
7 DCC protocol .....	209
8 DMX512 protocol .....	209
9 Ethernet 10BASE-T protocol .....	210
10 Fast Ethernet 100BASE-TX protocol .....	210
11 FlexRay protocol .....	211
12 I <sup>2</sup> C protocol .....	211
13 I <sup>2</sup> S protocol .....	212
14 LIN protocol .....	212

---

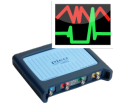
15 Manchester encoding .....	213
16 Modbus ASCII protocol .....	214
17 Modbus RTU protocol .....	215
18 PS/2 protocol .....	216
19 SENT Fast protocol .....	216
20 SENT Slow protocol .....	217
21 SPI protocol .....	217
22 UART protocol .....	218
23 USB (1.0/1.1) protocol .....	218
6 Signal generator waveform types .....	219
7 Spectrum window functions .....	220
8 Device feature table .....	221
9 Glossary .....	222
Index .....	225

# 1 Welcome

Welcome to PicoScope 6, the PC Oscilloscope software from Pico Technology. This user's guide is designed to help you use both versions of the software: **PicoScope 6**, for test and measurement users and **PicoScope 6 Automotive**, for vehicle diagnostics users.



With a scope device from Pico Technology, [PicoScope](#) turns your PC into a powerful [PC oscilloscope](#) and [spectrum analyzer](#), with all the features and performance of a benchtop [oscilloscope](#) at a fraction of the cost.

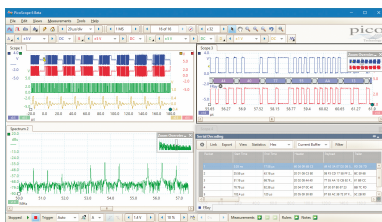


Benefits include fast capture rates, fast data processing, clear graphics and text, touchscreen support, frequent free-of-charge updates and the ability to easily save, print and share your data.

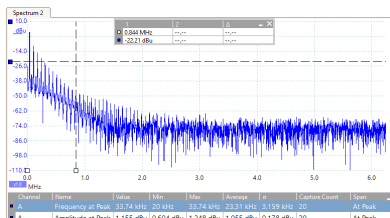
PicoScope 6 supports the devices listed in the [device feature table](#) and runs on Windows operating systems, with beta versions for macOS and Linux. See [System requirements](#) for further recommendations.

For help getting started, see [Using PicoScope for the first time](#) and the [PicoScope and oscilloscope primer](#). For more in-depth information, refer to the sections on [Menus](#), [Toolbars and buttons](#) and [Reference](#). Finally, for step-by-step tutorials, see the [How to...](#) section.

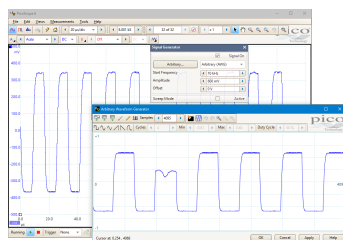
Here are some of the key features of the software:



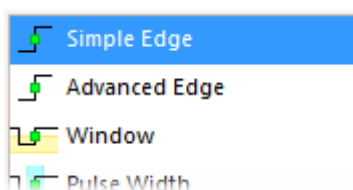
[Scope, spectrum and XY views](#) show your data in both time and frequency domains at once.



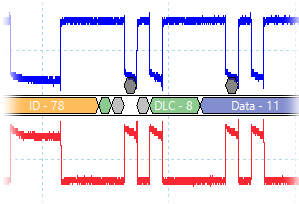
View and analyze your data in the frequency domain with a fully optimized [spectrum analyzer](#).



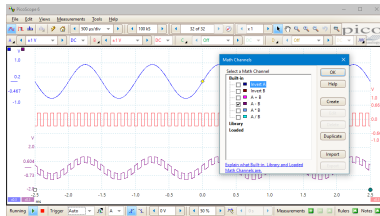
[Signal generator](#) for oscilloscopes with a built-in arbitrary waveform generator (AWG).



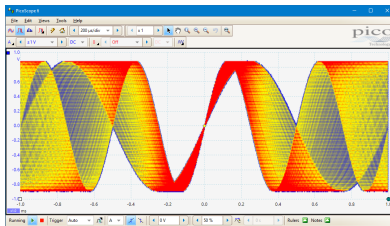
[Advanced triggering](#) includes options for pulse, window, digital and logic triggers.



Real-time [serial decoding](#) supports up to 16 different protocols, with color-coded packet fields.



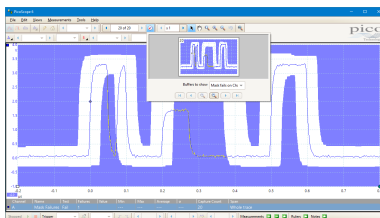
[Math channels](#) let you display mathematical functions of input channels.



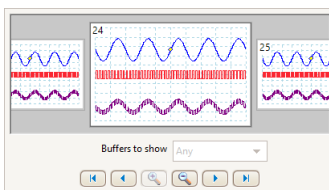
[Persistence mode](#) superimposes new waveforms on old, making glitches and dropouts stand out.

	Max	Average	σ
	660 ns	660 ns	0
V	4.734 V	4.734 V	0
ms	2.324 μs	2.319 μs	2

Add, remove and edit a range of [automatic measurements](#) in both time and frequency domains.



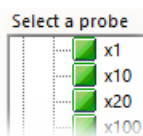
Set up [mask limit testing](#) to see when a signal goes out of bounds.



Identify waveform glitches and mask test failures by searching with the [buffer overview](#).

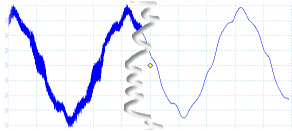


Use [rapid trigger mode](#) to capture a sequence of waveforms with the minimum possible [dead time](#).

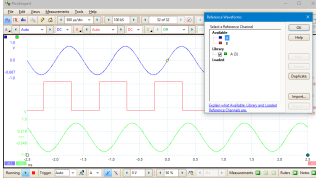


The [custom probes dialog](#) makes it easy to use your own probes and sensors with PicoScope.

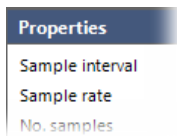




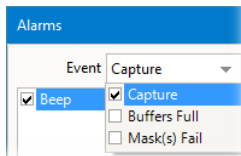
Apply [lowpass filtering](#) to individual channels.



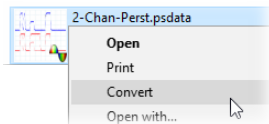
Use [reference waveforms](#) to compare stored copies of input channel waves with the live signal.



The [properties sheet](#) displays all your settings at a glance.



Set [alarms](#) to alert you when specified events occur.



[Windows Explorer integration](#) lets you convert .psdata files to other formats.



You can also run PicoScope 6 in the [command line](#).

## 2 Introduction

### 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](http://www.picotech.com) and [through the software](#) itself. We reserve the right to charge for updates or replacements sent out on physical media.

### 2.3 Trademarks

*Pico Technology*, *PicoScope*, *PicoLog*, *DrDAQ* 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



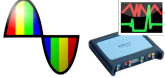
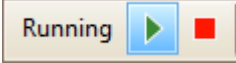

To make sure that PicoScope operates correctly, you must have a computer with at least the minimum system requirements to run your operating system, which must be one of the systems listed in the following table. The performance of the oscilloscope will be better with a more powerful PC, and will benefit from a multi-core processor.

Item	Specification
<b>Operating system</b>	Windows 7, Windows 8, Windows 10*. 32 bit and 64 bit versions. Beta software is also available for Linux and macOS operating systems.
<b>Processor</b>	As required by the operating system
<b>Memory</b>	
<b>Free disk space</b>	
<b>Ports</b>	USB 3.0 or USB 2.0 port(s)

\* PicoScope version 6.11 and PicoSDK are compatible with Windows XP SP3 and Vista SP2 in addition to the Windows versions listed above. For best performance we recommend Windows 7 or later.

## 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.

-  Download the latest version of PicoScope 6 from [www.picotech.com/downloads](http://www.picotech.com/downloads) and install it.
-  Plug in your scope device. Windows will recognize it and prepare your computer to work with it. Wait until Windows tells you that the device is ready to use.
-  Click the new PicoScope icon on your Windows desktop.
-  PicoScope will detect your scope device and prepare to display a waveform. The "Running" label will appear in the bottom left corner of the PicoScope window, and the green [Go](#) button will be highlighted.
-  Connect a signal to one of the scope device's input channels and see your first waveform! To learn more about using PicoScope, please read the [PicoScope Primer](#).

### Problems?

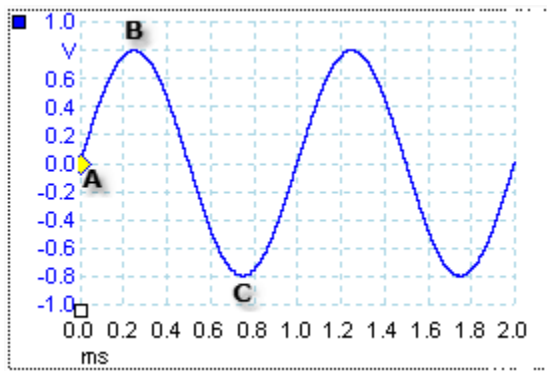
Help is at hand! Our technical support staff are always ready to answer your telephone call during office hours (09:00 to 17:00, UK time and 09:00 to 17:00 US Central time, Mon-Fri: see the contact details on the last page of this document). At other times, you can leave a message on our [support forum](#)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

## 4 PicoScope and oscilloscope primer

This chapter explains the fundamental concepts that you will need to know before working with the PicoScope software. If you have used an oscilloscope before, then most of these ideas will be familiar to you. You can skip the [Oscilloscope basics](#) section and go straight to the [PicoScope-specific information](#). If you are new to oscilloscopes, please take a few minutes to read at least the [Oscilloscope basics](#) and [PicoScope basics](#) topics.

### 4.1 Oscilloscope basics

An oscilloscope is a measuring instrument that shows, graphically, how a signal varies with time. Probes and sensors attached to its input channels enable it to plot various different signals (most commonly voltage or current, although pressure and vibration sensors are available for the automotive oscilloscopes). For example, the picture below shows a typical display on an oscilloscope screen when a varying voltage is connected to one of its input channels.



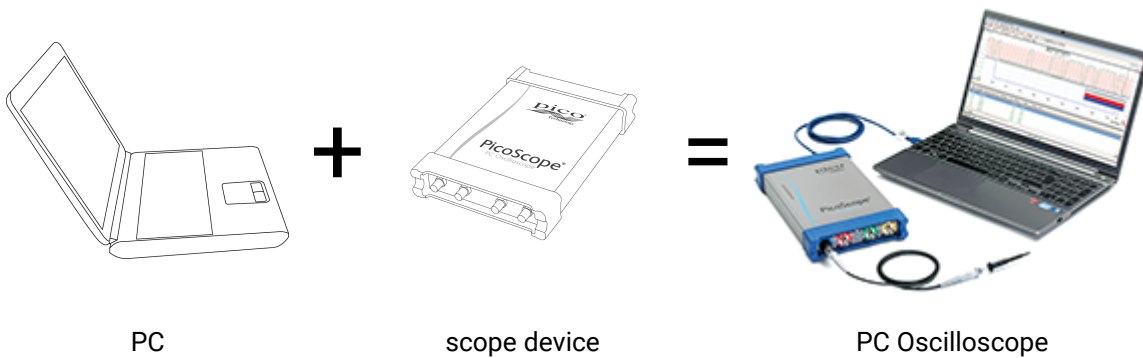
Oscilloscope displays are always read from left to right. The voltage-time characteristic of the signal, or waveform, is drawn as a line called the *trace*. In this example, the trace is blue and begins at point **A**. If you look to the left of this point, you will see the number **0.0** on the vertical [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 **collection time** and is measured in units of time - in this example, milliseconds.

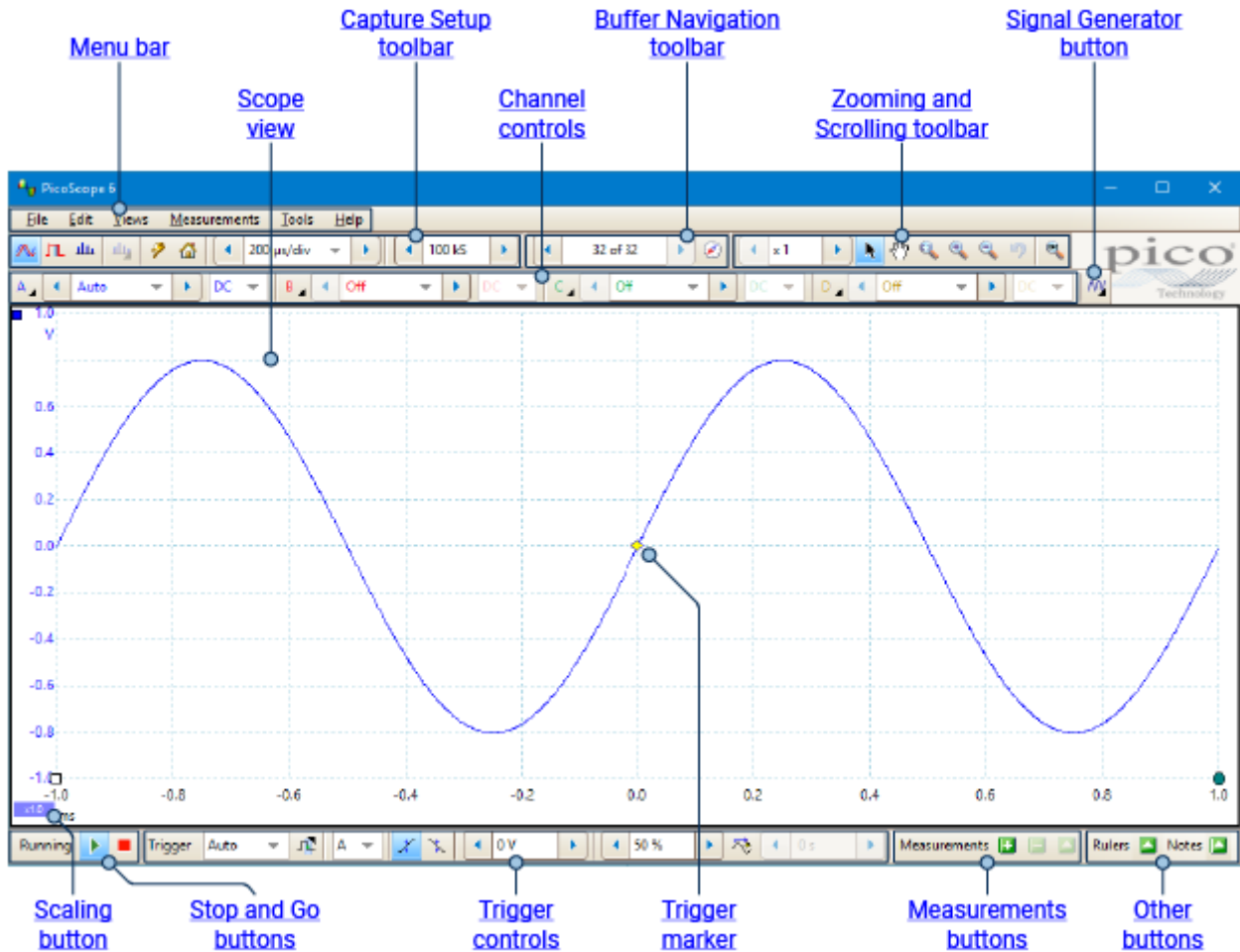
## 4.2 PC Oscilloscope basics

A **PC Oscilloscope** is a measuring instrument that consists of a hardware scope device and an oscilloscope program running on a PC. Oscilloscopes were originally stand-alone instruments with no signal processing or measuring abilities, and with storage only available as an expensive extra. Later oscilloscopes began to use new digital technology to introduce more functions, but they remained highly specialized and expensive instruments. **PC Oscilloscopes** are the latest step in the evolution of oscilloscopes, combining the measuring power of Pico Technology's scope devices with the convenience of the PC that's already on your desk.



## 4.3 PicoScope basics

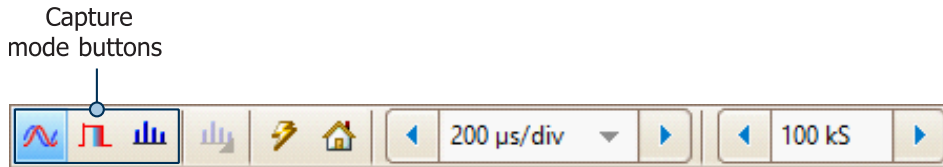
PicoScope can produce a simple display such as the example in the [Oscilloscope basics](#) topic, but it also has many advanced features. The screenshot below shows the PicoScope window. Click on any of the underlined labels to learn more. See [PicoScope window](#) for an explanation of these important concepts.



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

### 4.3.1 Capture modes

PicoScope can operate in three capture modes: **scope mode**, **spectrum mode** and **persistence mode**. The mode is selected by buttons in the [Capture Setup toolbar](#).



- 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 capture time. You can still display one or more secondary spectrum 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 views.
- In **persistence mode**, PicoScope displays a single, modified scope view in which old waveforms remain on the screen in faded colors while new waveforms are drawn in brighter colors. See also [How to find a glitch using persistence mode](#) and [Persistence Options dialog](#).

When you [save waveforms and settings](#), PicoScope only saves data for the mode that is currently in use. If you wish to save settings for both capture modes, then you need to switch to the other mode and save your settings again.

See also [How do capture modes work with views?](#)

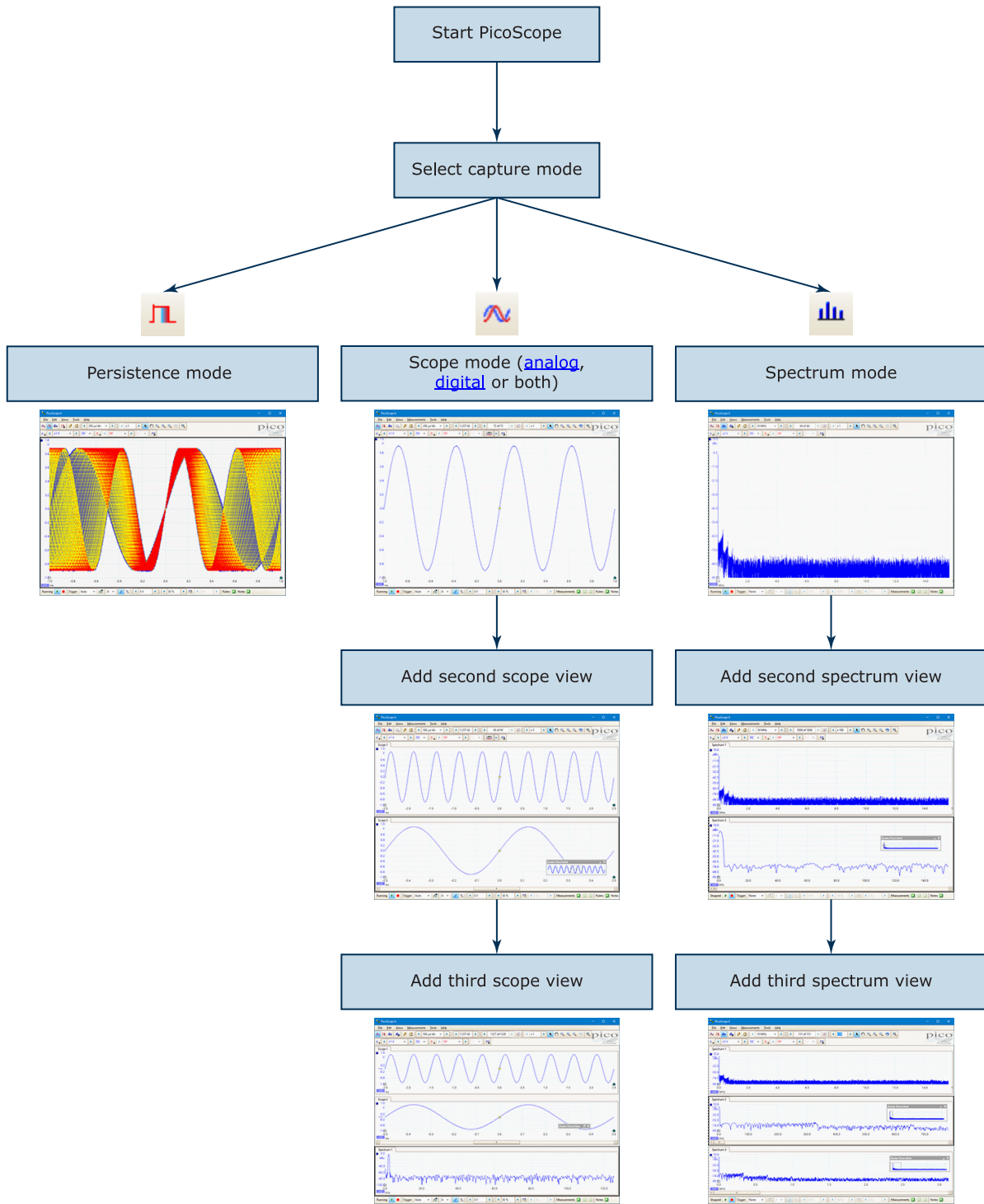
#### 4.3.1.1 How do capture modes work with views?

The [capture mode](#) tells PicoScope whether you are mainly interested in viewing waveforms ([scope mode](#) or [persistence mode](#)) or frequency plots ([spectrum mode](#)). When you select a capture mode, 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 [persistence 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 capture 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 capture mode, as illustrated in the diagram below.

When using a secondary view type (a spectrum view in scope mode, or a scope view in spectrum mode), you may see the data compressed horizontally rather than displayed neatly as in a primary view. You can usually overcome this by using the zoom tools.

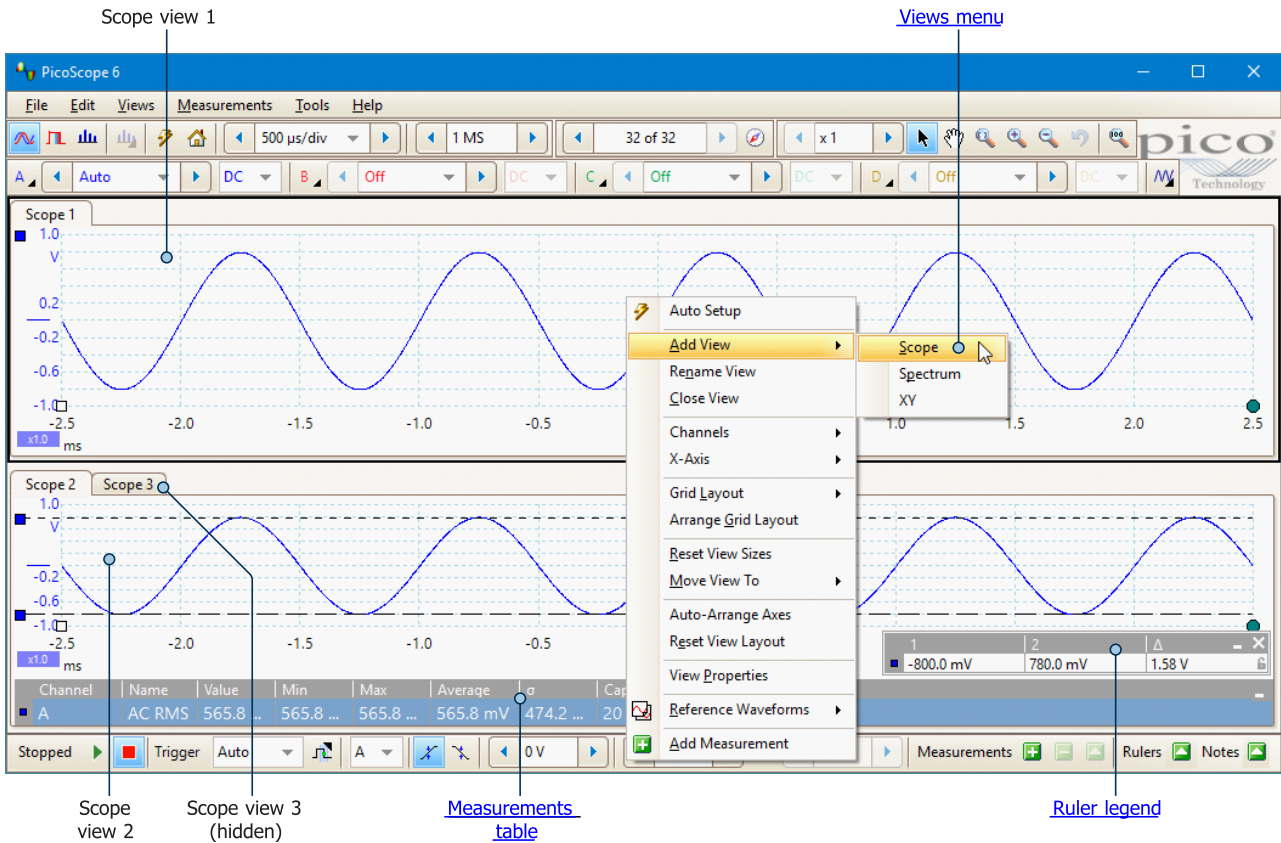




Examples showing how you might select the capture mode and open additional views in PicoScope

## 4.4 PicoScope window

The **PicoScope window** shows a block of data captured from the [scope device](#). When you first open PicoScope it contains one [scope view](#), but you can add more views by clicking **Add view** in the [Views menu](#). The screenshot below shows all the main features of the PicoScope window. Click on the underlined labels for more information.



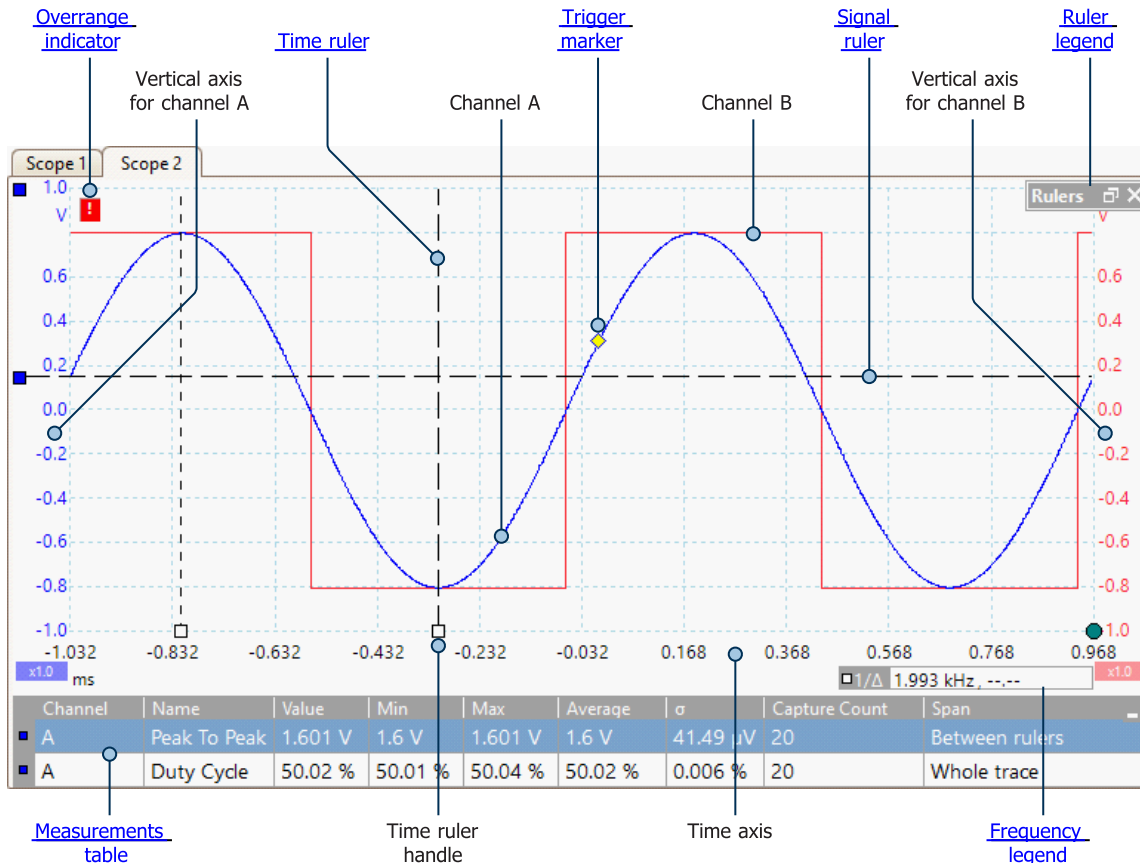
### To arrange the views within the PicoScope window

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, but you cannot move it outside the PicoScope window. You can also place more than one view in a viewport, by dragging a view and dropping it on top of another.

For further options, right-click on a view, or select **Views** from the [Menu bar](#), to open the [Views menu](#).


## 4.5 Scope view

A **scope view** shows the data captured from the scope as a graph of signal amplitude against time. (See [Oscilloscope basics](#) for more on these concepts). PicoScope opens with a single view, but you can add more views by using the [views menu](#). 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 have as many waveforms as the scope device has channels. Click on one of the labels below to learn more about a feature.




Scope views are available regardless of which mode - [scope mode](#) or [spectrum mode](#) - is active.


## 4.6 Overrange indicator

If the signal voltage exceeds the measurement input range, PicoScope displays the red overrange indicator  in the top left corner of the display, with the message "Channel overrange". 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 overrange indicator  appears in the top left corner of the PicoScope display, with the message "Common-mode overrange". 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.

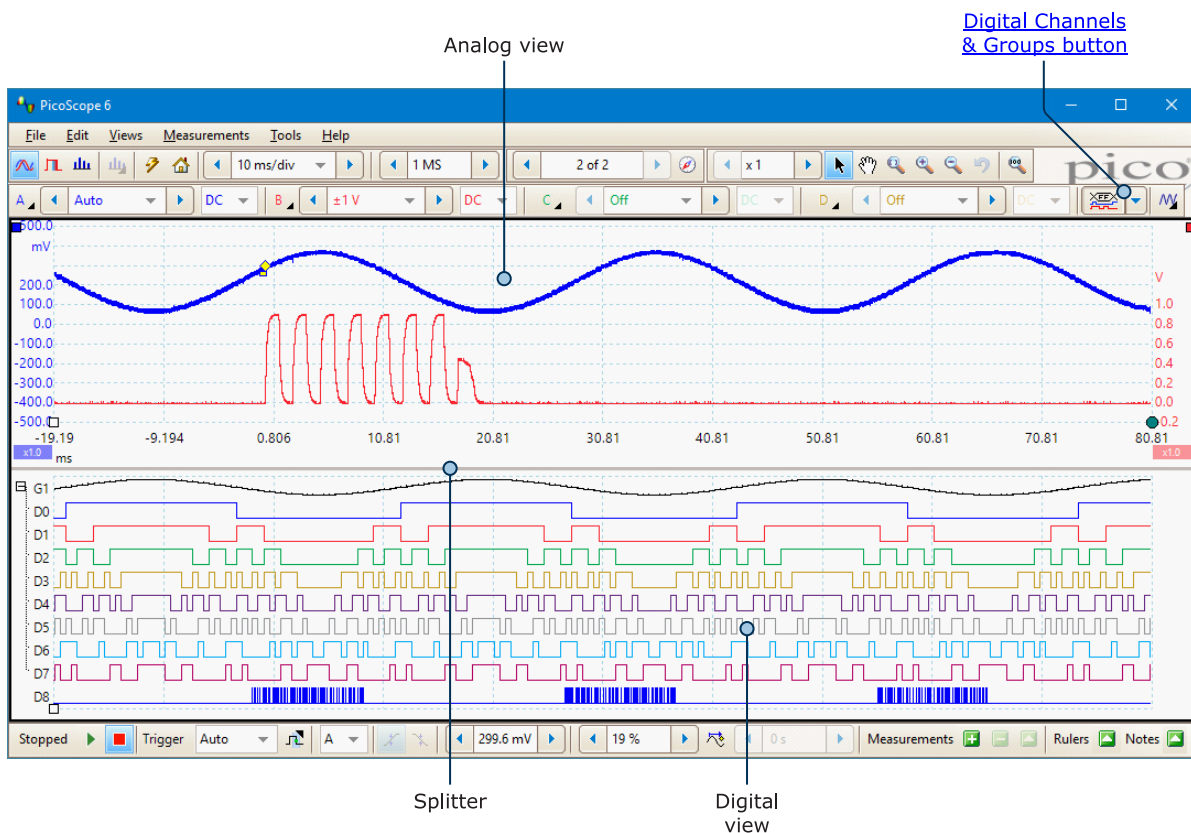
### Scopes with [floating inputs](#) (PicoScope 4225 and 4425 only)

If the voltage between the BNC shell and the oscilloscope chassis exceeds 30 V, the yellow warning indicator  appears in the top left corner 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 bolt on the rear of the unit.

## 4.7 MSO view

Applicability: mixed-signal oscilloscopes ([MSOs](#)) only

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



#### [Digital Channels & Groups button](#)

Switches the [digital view](#) on and off, and opens the [Select Digital Channels/Groups dialog](#).

#### Analog view

Shows the analog channels. The same as a standard [scope view](#).

#### Splitter

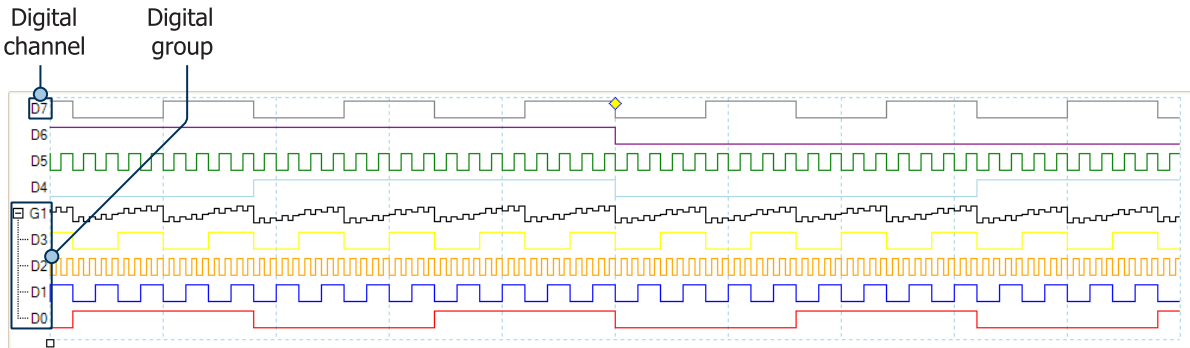
Drag up and down to move the partition between analog and digital sections.

#### Digital view

Shows the digital channels and groups. See [Digital view](#).

## 4.7.1 Digital view



Location: [MSO view](#)



Note 1: you can right-click on the digital view to obtain the [Digital context menu](#).

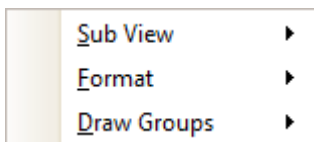
Note 2: if the digital view is not visible when required, check that (a) the [Digital Channels & Groups button](#) is activated and (b) at least one digital channel is selected for display in the [Select Digital Channels/Groups dialog](#).

**Digital channel:** Displayed in the order in which they appear in the [Select Digital Channels/Groups dialog](#), where they can be renamed.

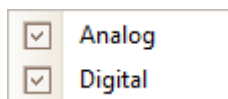
**Digital group:** Groups are created and named in the [Select Digital Channels/Groups dialog](#). You can expand and collapse them within the digital view using the  and  buttons.

## 4.7.2 Digital context menu

Location: right-click on the [digital view](#)



**Sub View:**

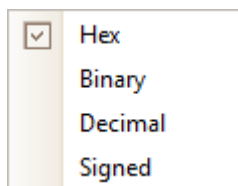


**Analog.** View or hide the [analog scope view](#)

**Digital.** View or hide the [digital scope view](#)

Also available from the [Views menu](#)

**Format:**



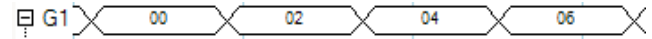
The numerical format in which group values are displayed in the [digital scope view](#)

Draw Groups:



**By Values.**

Draw groups with transitions only where the value changes:



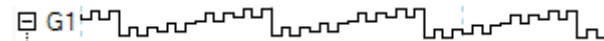
**By Time.**

Draw groups with transitions spaced equally in time, once per sampling period. You will usually need to zoom in to see the individual transitions:



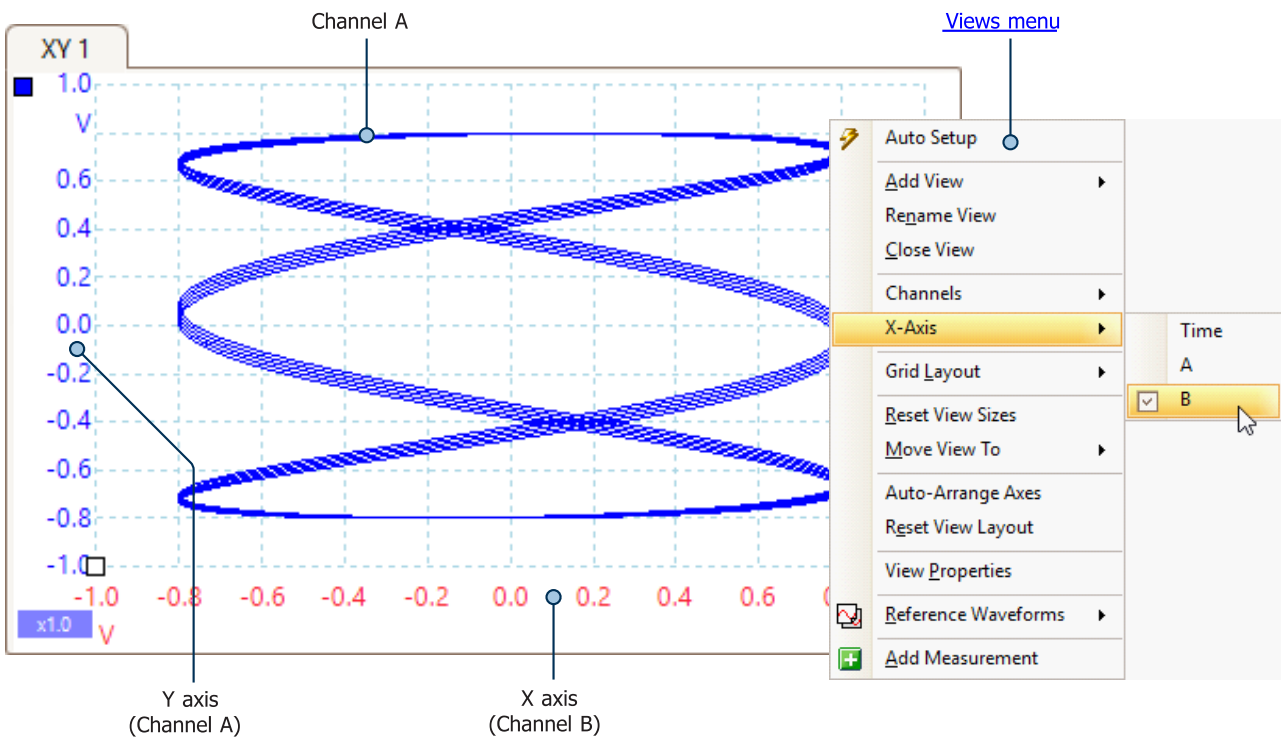
**By Level.**

Draw groups as analog levels derived from the digital data:



## 4.8 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.



In the example above, two different periodic signals have been fed into the two input channels. The smooth curvature of the trace tells us that the inputs are roughly or exactly sine waves. The three loops in the trace show that Channel B has about three times the frequency of Channel A. We can tell that the ratio is not exactly three because the loops do not line up exactly. Since an XY view has no time axis, it tells us nothing about the absolute frequencies of the signals. To measure frequency, we need to open a [Scope view](#).

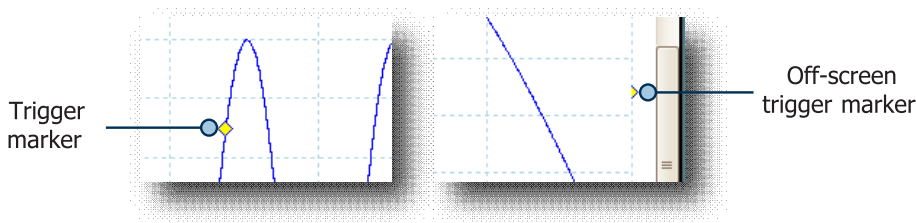
### How to create an XY view

There are two ways to create an XY view.

- Use the **Add View > XY** command on the [Views menu](#). 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).
- Use the **X-Axis** command on the [Views menu](#). This converts the current scope view into an XY view. It maintains the existing Y axes and allows you to choose any available channel for the X axis. With this method, you can even assign a [math channel](#) or a [reference waveform](#) to the X axis.

## 4.9 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.

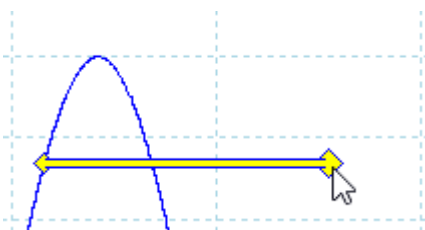
If you have set a [time delay](#), the trigger marker is temporarily replaced by the [time-delay arrow](#) while you adjust the time delay.

When some [advanced trigger types](#) are in use, the trigger marker changes to a window marker, which shows the upper and lower trigger thresholds.

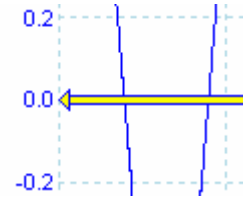
For more information, see [Trigger controls](#) and its subsections.

## 4.10 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. If zero on the time axis is outside the [scope view](#), then the left-hand end of the time-delay arrow appears like this:

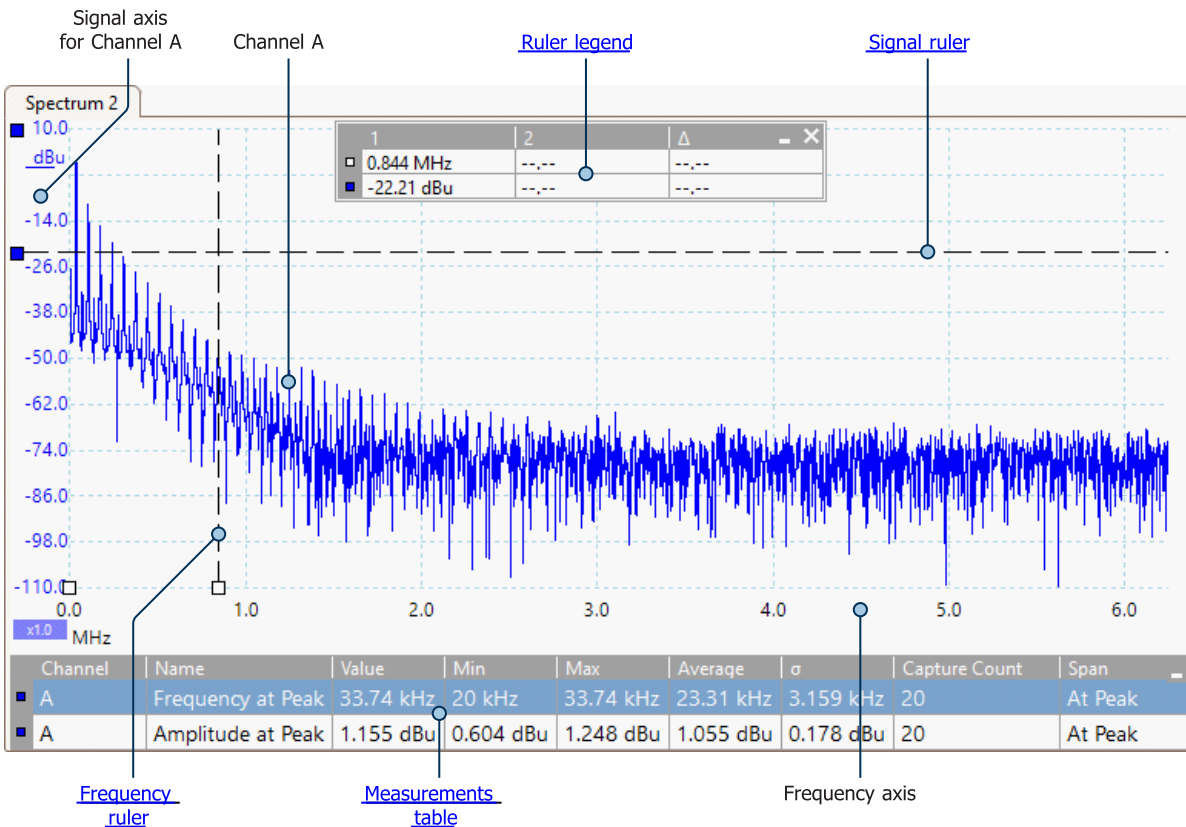


The right-hand end of the arrow (temporarily replacing the [trigger marker](#)) indicates the trigger reference point.

You can set up a time delay using the [Trigger controls](#). For more information on time delays, see [Trigger timing](#).

## 4.11 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 menu](#). 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. Click on one of the labels below to learn more about a feature.




Spectrum views are available regardless of which mode - [Scope Mode](#) or [Spectrum Mode](#) - is active.

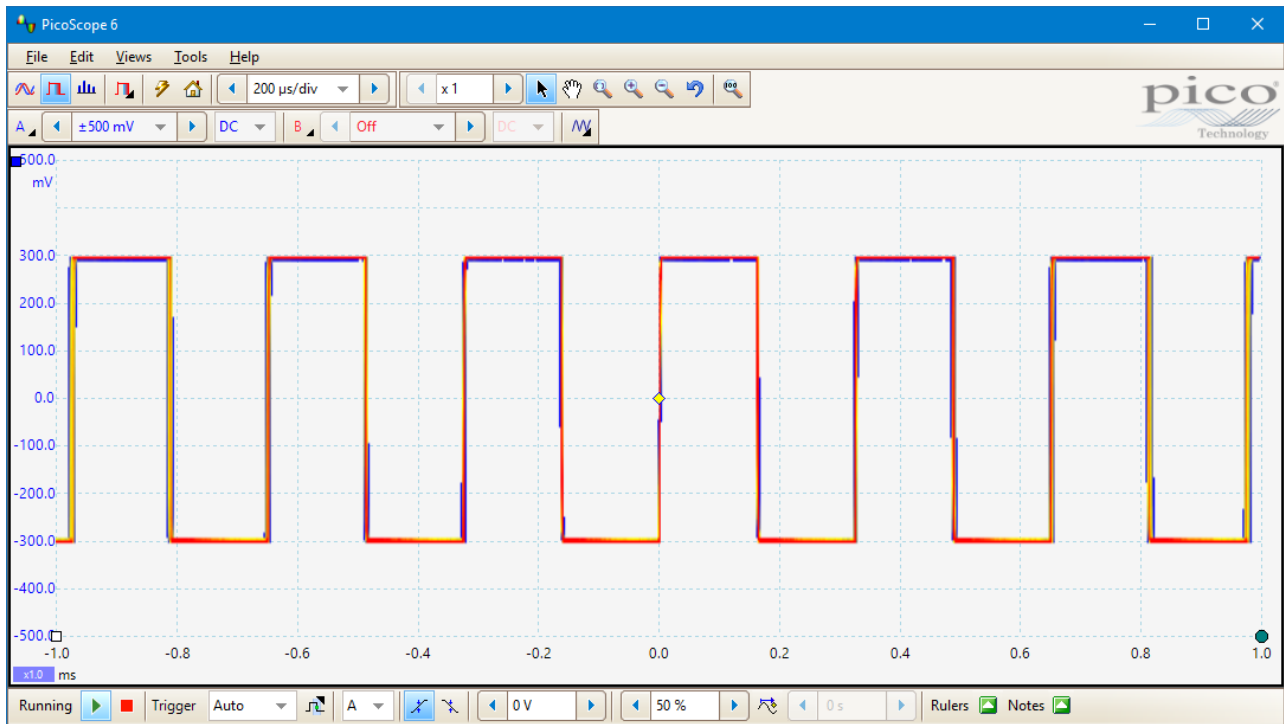
For more information, see: [How to set up the spectrum view](#) and [Spectrum Options dialog](#).



## 4.12 Persistence mode

**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**  on the [Capture Setup toolbar](#). With the [persistence options](#) set at their default values, the screen will look something like this:



The colors indicate the frequency of the data. Red is used for the highest-frequency data, with yellow for intermediate frequencies and blue for the least frequent data. 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. These are the default colors, but you can change them using the [Colors page](#) of the [Preferences dialog](#).

This example shows persistence mode in its most basic form. See the [Persistence Options dialog](#) for ways to modify the display to suit your application, and [How to find a glitch using persistence mode](#) for a worked example.

## 4.13 Measurements table

A **measurements table** displays the results of automatic measurements. Each [view](#) can have its own table, and you can add, delete or edit measurements from this table.

You can choose from [18 measurements in Scope view](#), plus another [11 in Spectrum view](#).

Channel	Name	Span	Value	Min	Max	Average	$\sigma$	Capture Count
A	AC RMS	Whole trace	707.7 mV	707.5 mV	714.3 mV	712.5 mV	2.824 mV	20
A	Frequency	Whole trace	4.567 kHz	3.333 kHz	7.336 kHz	5.296 kHz	997.5 Hz	20
A	Rise Time [80/20%]	Whole trace	187.6 $\mu$ s	2.436 $\mu$ s	472.8 $\mu$ s	140.5 $\mu$ s	123.7 $\mu$ s	20

### Measurements table columns

<b>Name</b>	The name of the measurement that you selected in the <a href="#">Add/Edit Measurement</a> dialog.
<b>Span</b>	The section of the waveform or spectrum that you want to measure. This is 'Whole trace' by default.
<b>Value</b>	The live value of the measurement, from the latest capture
<b>Min</b>	The minimum value of the measurement since measuring began
<b>Max</b>	The maximum value of the measurement since measuring began
<b>Average</b>	The arithmetic mean of the measurements from the last $n$ captures, where $n$ is the number of <b>Statistics Captures</b> set on the <a href="#">General page</a> of the <a href="#">Preferences dialog</a>
<b><math>\sigma</math></b>	The <a href="#">standard deviation</a> of the measurements from the last $n$ captures, where $n$ is the number of <b>Statistics Captures</b> set on the <a href="#">General page</a> of the <a href="#">Preferences dialog</a>
<b>Capture Count</b>	The number of captures used to create the statistics above. This starts at 0 when triggering is enabled, and counts up to the number of <b>Statistics Captures</b> specified on the <a href="#">General page</a> of the <a href="#">Preferences dialog</a> .

### To add, edit or delete measurements

There are three ways to add, edit and delete measurements. You can right-click on the view, open the [Measurements menu](#), or use the [Measurements buttons](#).

### To change the width of a measurement column

First make sure that the **Column Auto-width** option is not enabled in the [Measurements menu](#). If necessary, click the option to switch it off. Then drag the vertical separator between column headings to resize the columns, as shown opposite.

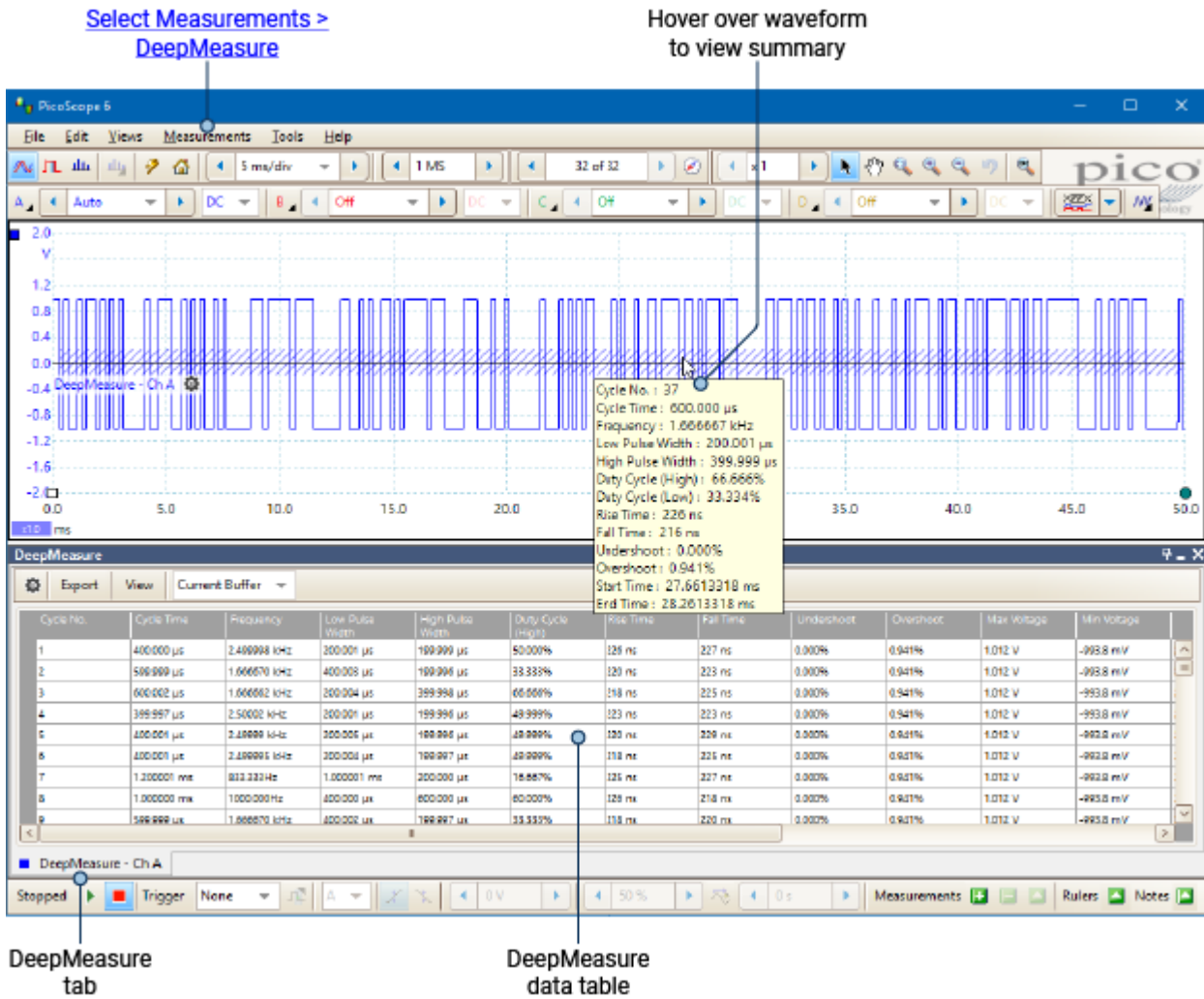
	Max	Average
V	705.8 mV	705.5 mV
s	1 ms	1 ms
V	-8.829 mV	-9.39 mV

### To change the update rate of the statistics

The statistics (**Min**, **Max**, **Average**, **Standard Deviation**) are based on the number of captures shown in the **Capture Count** column. You can change the maximum capture count using the **Statistics Captures** control on the [General page](#) of the [Preferences dialog](#).

## 4.14 DeepMeasure™

The standard measurements available in PicoScope 6 are average values taken across either a whole waveform or across the interval between two time or frequency rulers. This means that making measurements at the level of the individual wave cycle is time-consuming and complicated. DeepMeasure allows you to make up to 1 million cycle-level measurements of 12 different parameters over a single waveform or over the entire waveform buffer in seconds, and you can set it up to run multiple configurations on one or more active channels. Please refer to the [Device feature table](#) for a list of supported devices.

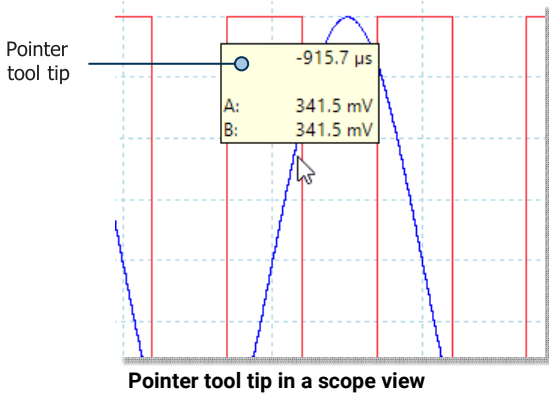


To start using DeepMeasure, click **Measurements > DeepMeasure** from the Menu bar. You can simultaneously run as many instances of DeepMeasure as you like, with different settings or on different channels, and switch between them using the **DeepMeasure tabs**.

For more information, see [DeepMeasure™ dialog](#).

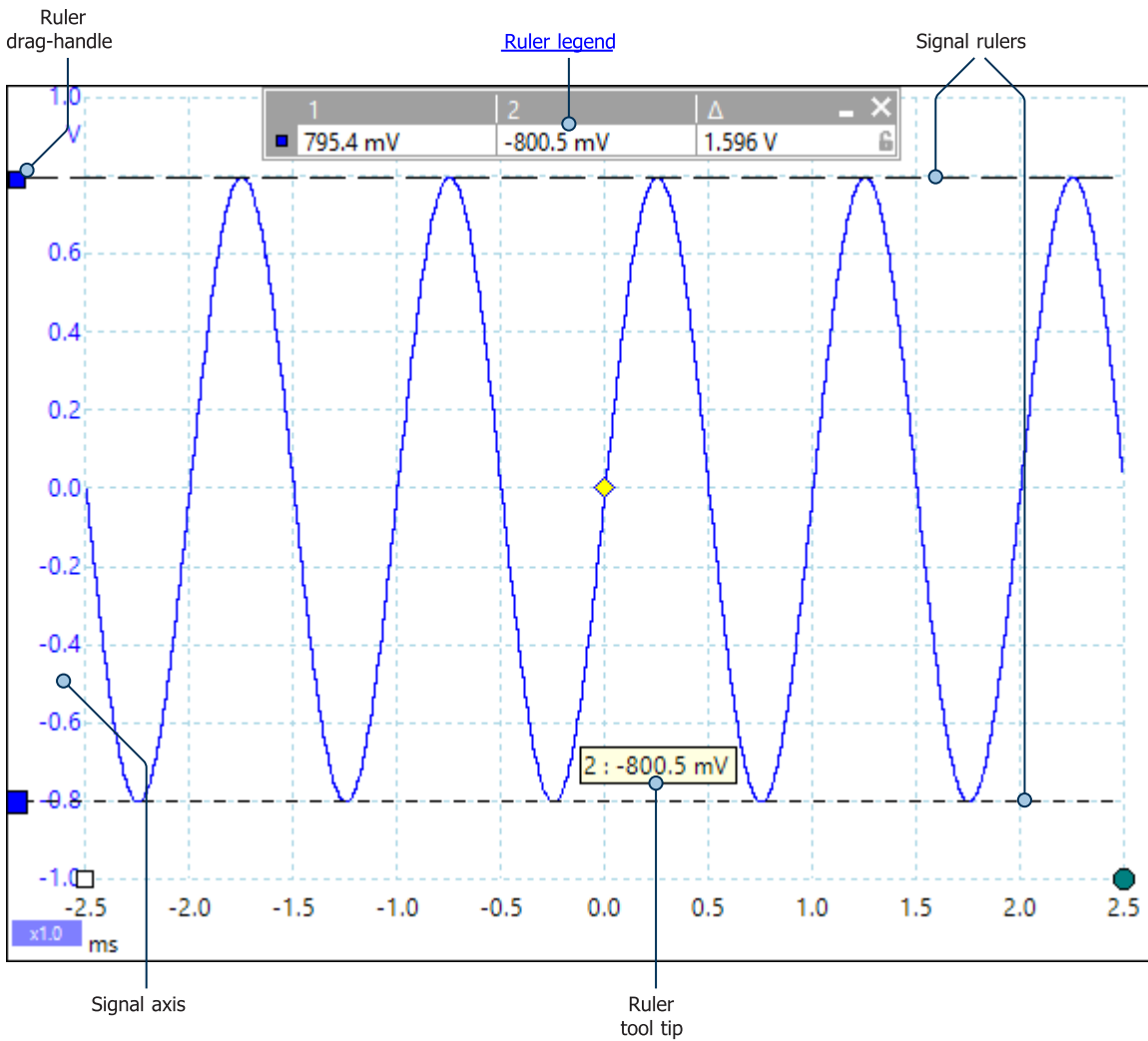
### 4.15 Pointer tool tip

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](#).




### 4.16 Signal rulers

The signal rulers help you measure absolute and relative signal levels on a [scope](#), [XY](#) or [spectrum view](#).



In the [scope view](#) above, the two colored squares 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  button, all the rulers are deleted.

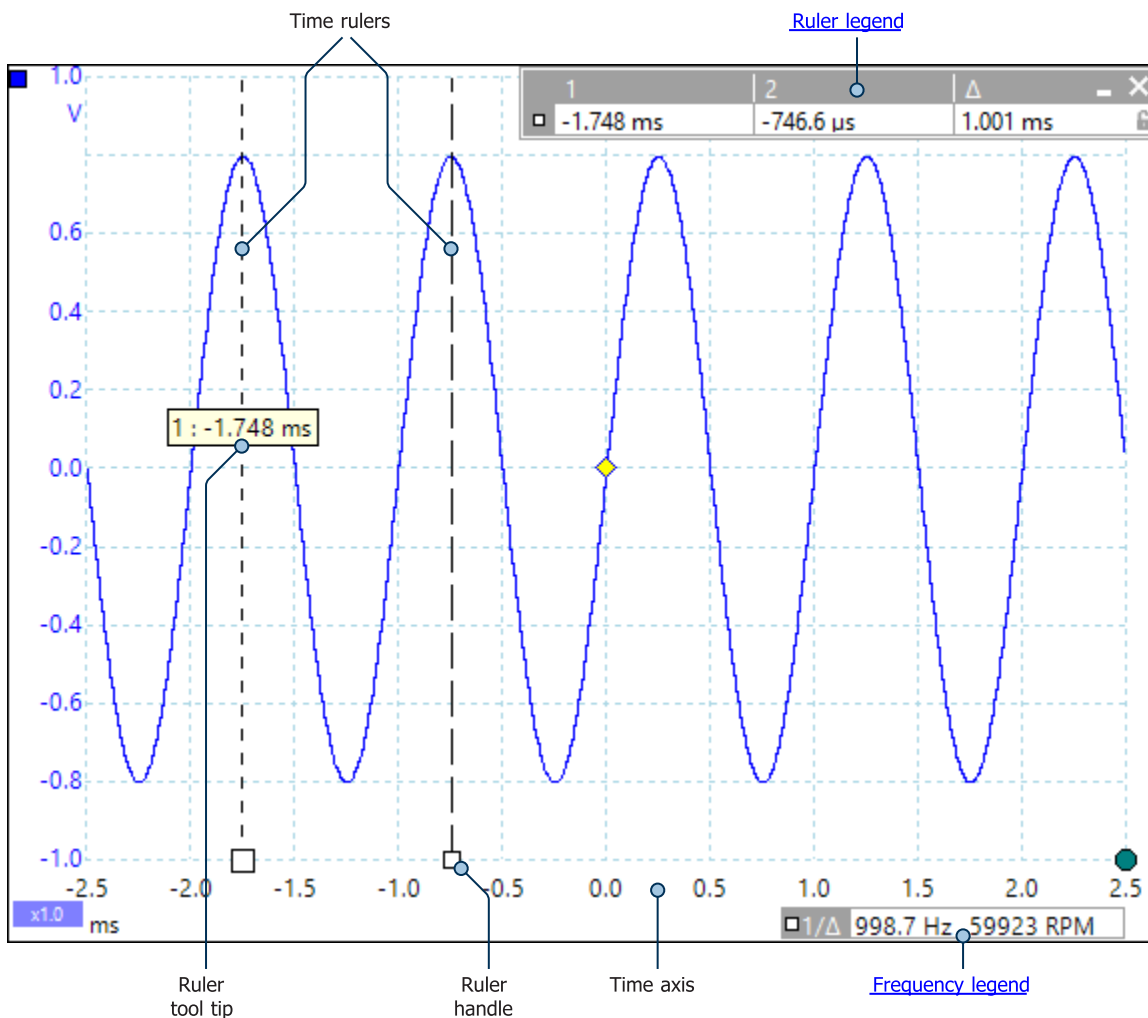
Signal rulers also work in [spectrum](#) and [XY](#) views.

### Ruler tool tip

If you move the mouse pointer over one of the rulers, PicoScope displays a [tool tip](#) with the ruler number and the signal level of the ruler. You can see an example of this in the picture above.

## 4.17 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 white squares 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 tool tip

If you hold the mouse pointer over one of the rulers, as we did in the example above, PicoScope displays a [tool tip](#) with the ruler number and the time or frequency value at the ruler.

### 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 -1.748 milliseconds, ruler 2 is at -746.6 microseconds and that the difference between them is 1.001 milliseconds. Closing the ruler legend using the  button deletes all the rulers.

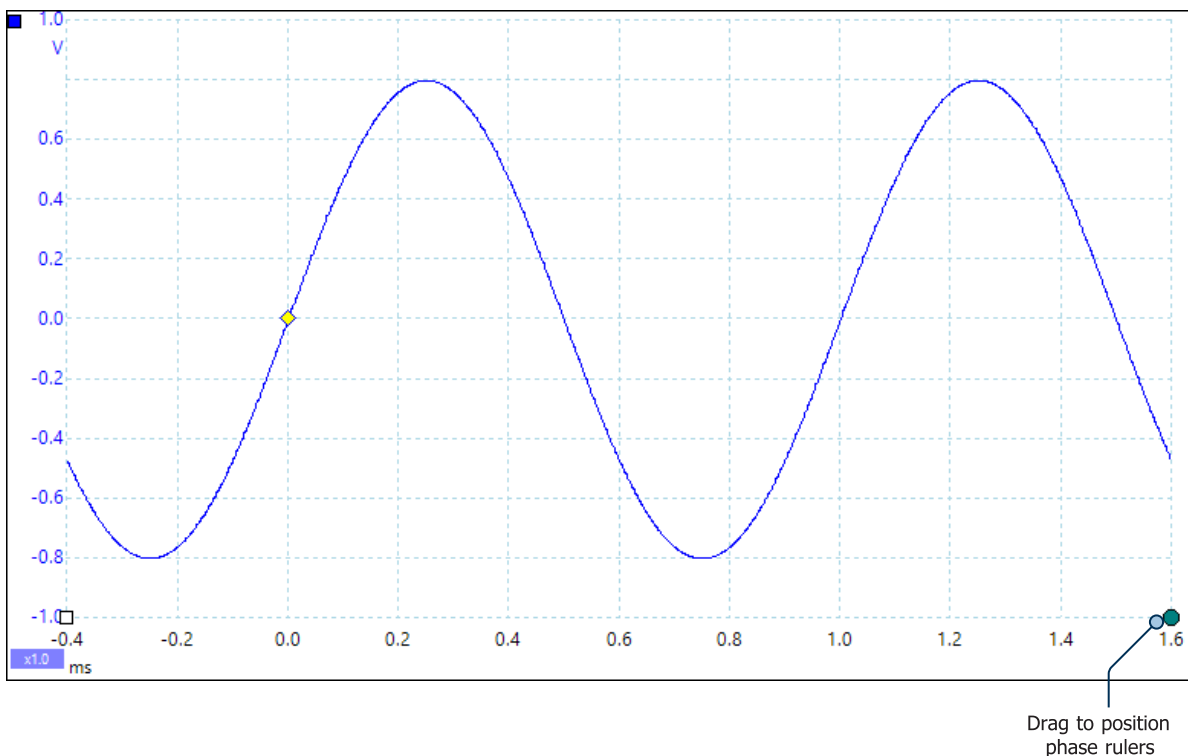
### Frequency legend

The **frequency legend** in the bottom right-hand corner of a scope view shows  $1/\Delta$ , where  $\Delta$  is the difference between the two time rulers. 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 (if selected on the [Options page](#) of the [Preferences dialog](#)) revolutions per minute (RPM).

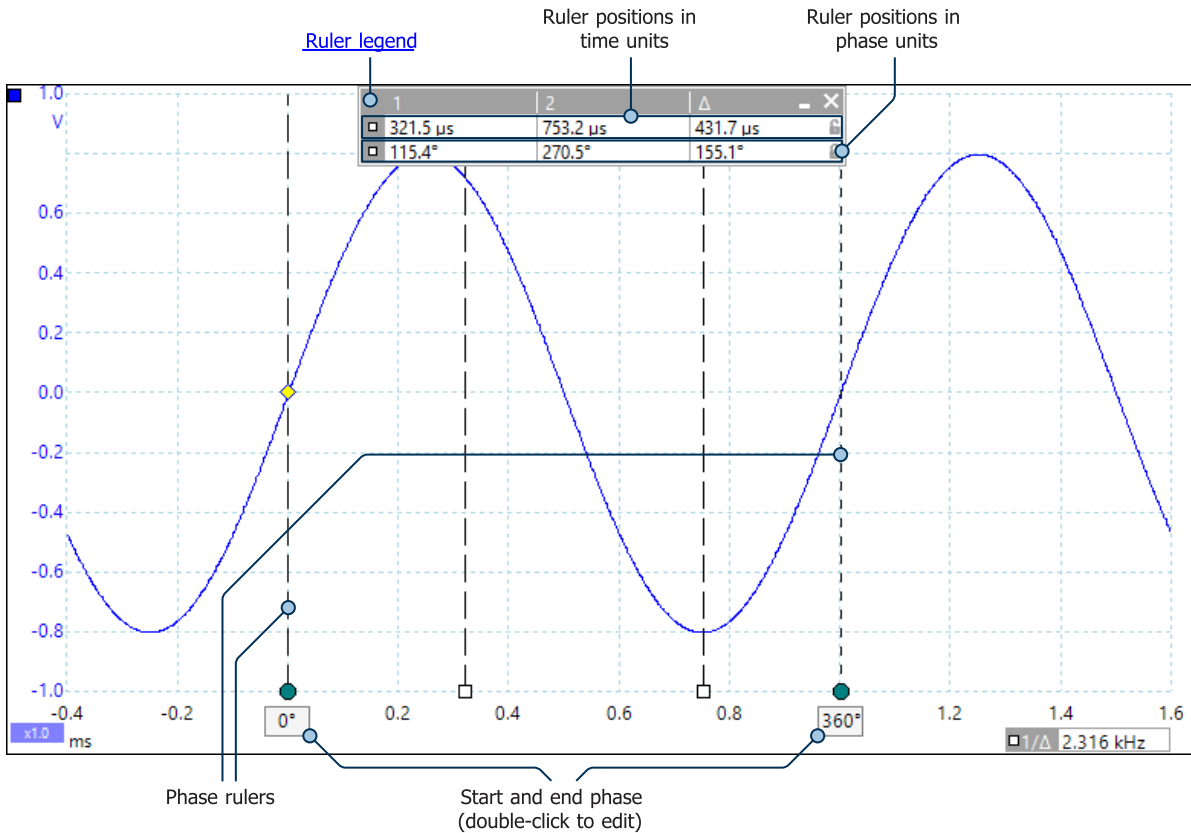
## 4.18 Phase (rotation) rulers

The **phase rulers** (called **rotation rulers** in PicoScope 6 Automotive) 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, drag the two phase ruler handles onto the waveform from their inactive position as shown below:



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 default 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. Closing the ruler legend dismisses all rulers, including the phase rulers.

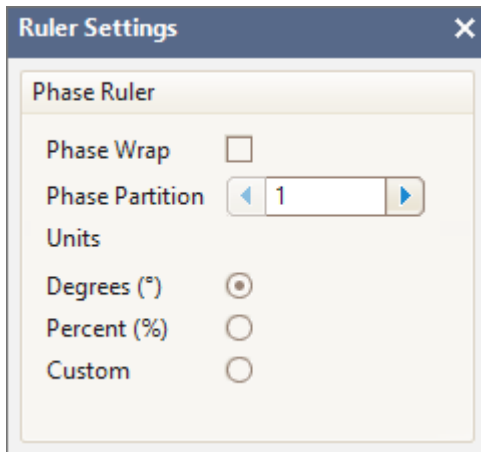
### Ruler settings

Options for the **phase (rotation) rulers** are configured by the [Ruler Settings dialog](#), which is called up by the [Rulers button](#) on the [Trigger toolbar](#).

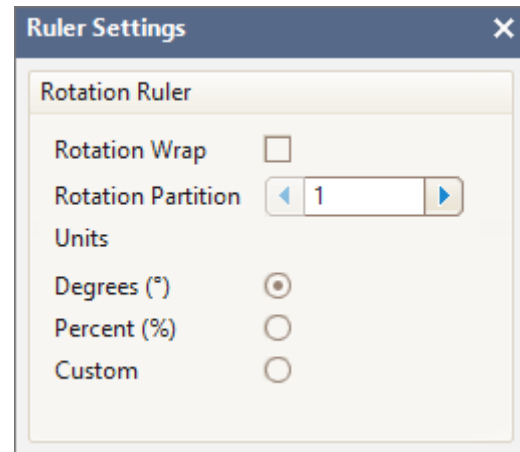
## 4.19 Ruler settings

Location: [Other buttons](#) > **Rulers**

The **Ruler Settings** box allows you to control the behavior of the [time rulers](#) and [phase rulers](#) (called **rotation rulers** in PicoScope 6 Automotive).



Ruler Settings in PicoScope 6



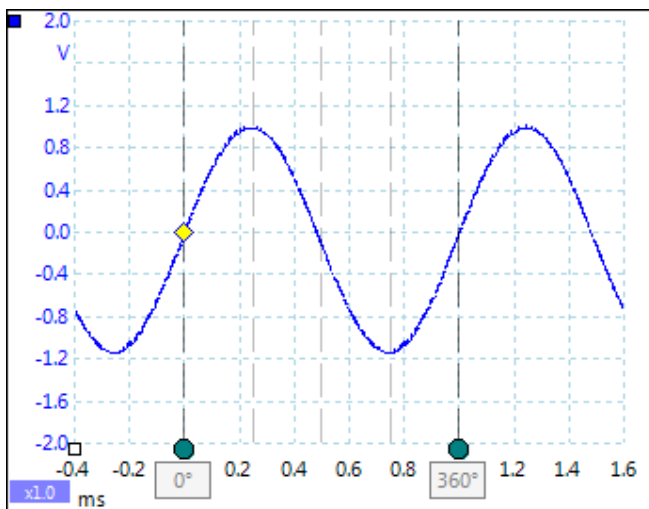
Ruler Settings in PicoScope 6 Automotive

### Phase (Rotation) Wrap

If this box is checked, [time ruler](#) values outside the range set by the [phase \(rotation\) rulers](#) are wrapped back into that range. For example, if the phase (rotation) rulers are set to  $0^\circ$  and  $360^\circ$ , the value of a time ruler just to the right of the  $360^\circ$  phase (rotation) ruler will be  $0^\circ$ , and the value of a time ruler just to the left of the  $0^\circ$  phase (rotation) ruler will be  $359^\circ$ . If this box is unchecked, ruler values are unconstrained.

### Phase (Rotation) Partition

Increasing this value above 1 causes the space between the two phase (rotation) rulers to be partitioned equally into the specified number of intervals. The intervals are marked by broken lines between the phase (rotation) 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.



Phase (rotation) rulers with 4 partitions



## Units

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

## 4.20 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:

	Ruler 1 values	Ruler 2 values	Ruler difference	
Time or frequency rulers	1 -16.99 $\mu$ s	2 -11.78 $\mu$ s	$\Delta$ 5.21 $\mu$ s	Minimize and Close buttons
Signal rulers	226.0 mV	-423.0 mV	649.0 mV	Lock button
	72.0 mV	-403.0 mV	475.0 mV	

### Editing

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

### Tracking 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 causes the other to follow it, maintaining a fixed separation. The button changes to  when the rulers are locked.

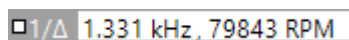
**TIP:** To set up a pair of tracking rulers with a known distance between them, first click the **Lock** button, then edit the two values in the ruler legend so that the rulers are the desired distance apart.

### Phase (rotation) rulers

When [phase rulers](#) (called **rotation rulers** in PicoScope 6 Automotive) are in use, the ruler legend displays additional information.

See also [frequency legend](#).

## 4.21 Frequency legend



The **frequency legend** appears when you have placed two [time rulers](#) on a [scope view](#). It shows  $1/\Delta$  in hertz (the SI unit of frequency, equal to cycles per second), where  $\Delta$  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 [Add Measurements button](#).

For frequencies up to 1.666 kHz, the frequency legend can also show the frequency in RPM (revolutions per minute). The RPM display can be enabled or disabled in the [Preferences > Options dialog](#).

## 4.22 Properties sheet

Location: [Views](#) > **View Properties**

Purpose: shows a summary of the settings that PicoScope 6 is using

The **Properties** sheet appears on the right-hand side of the PicoScope window.

The screenshot shows the 'Properties' dialog box with the following sections and settings:

Section	Property	Value
Capture settings	Sample interval	64 ns
	Sample rate	15.63 MS/s
	No. samples	781,250
	Hardware Resolution	12 bits
Spectrum settings	Window	Blackman
	No. bins	16384
	Bin width	476.8 Hz
	Time gate	2.097 ms
Channel settings	Channel	A
	Range	±2 V
	Coupling	DC
	Res. enhancement	13.0 bits
	Channel	B
	Range	±10 mV
	Coupling	DC
	Effective res.	11 bits
Signal generator settings	Signal type	Square
	Frequency	1 kHz
	Amplitude	1 V
	Offset	0 V
Time stamp	Δ First Trigger	3.999ms
	Δ Previous Trigger	999.985μs
	Capture Date	20/01/2020
	Capture Time	11:04:55
Capture rate (persistence mode only)	Capture Rate	14

### No. samples

The number of samples captured. This may be lower than the number requested in the [Maximum Samples](#) control. A number in brackets is the number of interpolated samples, if [interpolation](#) is enabled.

### Window

[Spectrum views](#) only. The [window function](#) applied to the data before computing the spectrum. This is selected in the [Spectrum options dialog](#).

<b>Time gate</b>	<a href="#">Spectrum views</a> only. The time taken to collect all the data for the fast Fourier transform (FFT), measured from the start of the capture.
<b>Res. enhancement (resolution enhancement)</b>	Indicates that you are using <a href="#">resolution enhancement</a> . The number of bits, including resolution enhancement, selected in the <a href="#">Channel Options dialog</a> .
<b>Effective res.</b>	Effective resolution: applies to <a href="#">flexible resolution</a> oscilloscopes only. PicoScope tries to use the value specified by the <b>Hardware Resolution</b> control in the <a href="#">Capture Setup toolbar</a> , but on some input ranges the hardware delivers a lower effective resolution. The available resolutions are specified in the data sheet for the scope device.
<b>Δ First Trigger</b>	High-resolution start time of this waveform relative to the last reset of the oscilloscope timer. The timer is reset when settings are changed or a new rapid triggering sequence is started. <a href="#">Selected PicoScope models only.</a>
<b>Δ Previous Trigger</b>	High-resolution start time of this waveform relative to the previous one. <a href="#">Selected PicoScope models only.</a>
<b>Capture rate</b>	<a href="#">Persistence mode</a> only. The number of waveforms being captured per second.

## 4.23 Custom probes

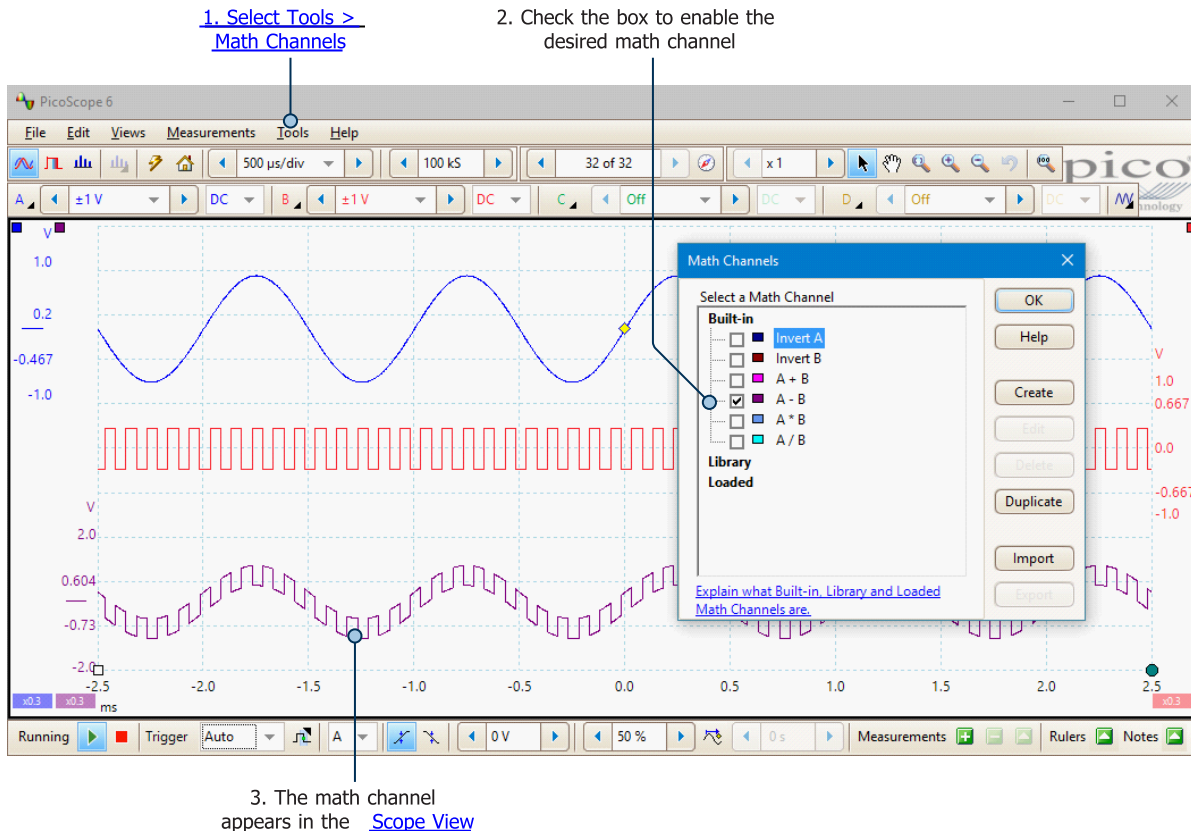
A **probe** is any transducer, measuring device or other accessory that you connect to an input channel of your [scope device](#). PicoScope has a built-in library of common probe types, such as the x1 and x10 voltage probes used with most oscilloscopes, but if your probe is not included in this list you can use the [Custom 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.


## 4.24 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 button](#) and [color](#). PicoScope 6 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](#), or [load predefined math channels from files](#).

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



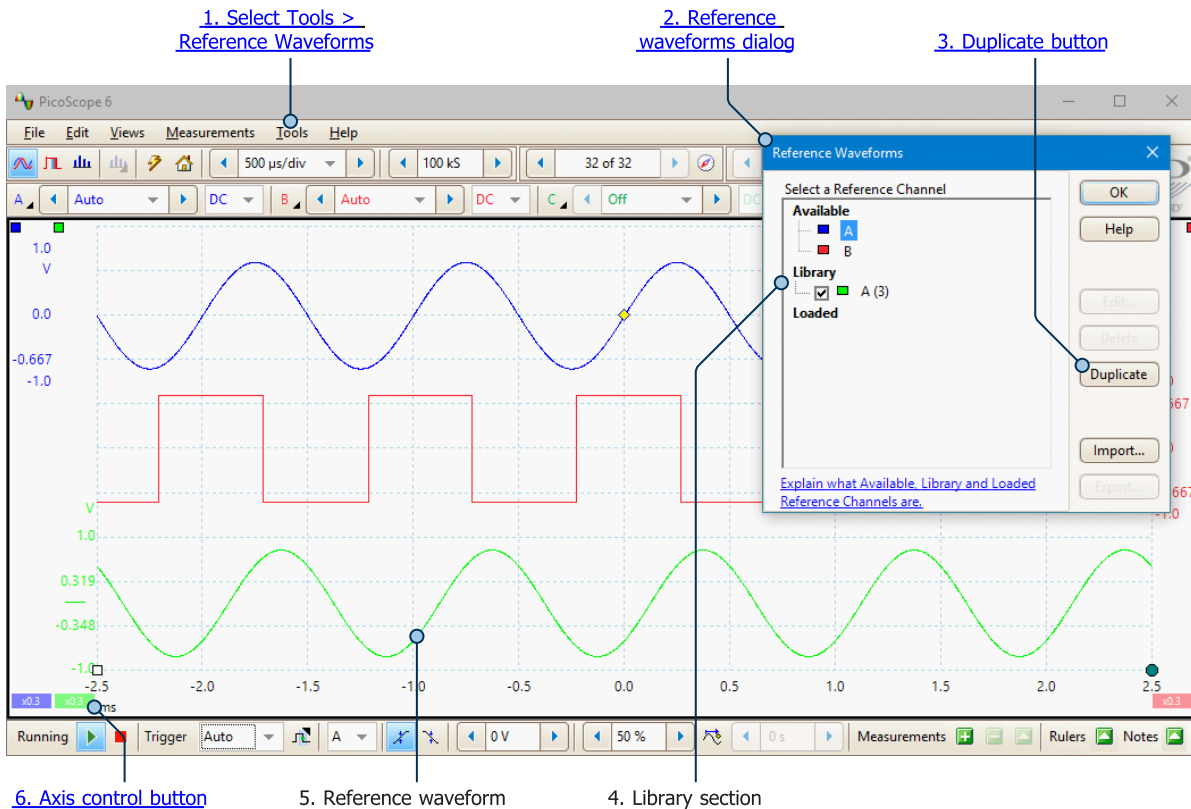
1. **Tools > Math Channels command.** Click this to open the [Math Channels dialog](#), shown in the image above.
2. **Math Channels dialog.** This lists all the available math channels and allows you define new ones. In the example above, only the built-in functions are listed.
3. **Math channel.** 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).

You may occasionally see a flashing warning triangle -  - at the bottom of the math channel axis. This means that the channel cannot be displayed because an input source is missing. For example, this occurs if you enable the **A+B** function while Channel B is set to **Off**.

## 4.25 Reference waveforms

A **reference waveform** is a stored version of an input signal. You can create one by right-clicking on the view, selecting the **Reference Waveforms** command and selecting which channel to copy. 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 and offset button](#), and [color](#). The reference waveform may have fewer samples than the original.

For more control over Reference Waveforms, use the [Reference Waveforms dialog](#) as shown below.



- Tools > Reference Waveforms command.** Click this open the [Reference Waveforms dialog](#), shown in the image above.
- Reference Waveforms dialog.** 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 **Available** section. The **Library** section is empty to begin with.
- Duplicate button.** When you select an input channel or reference waveform and click this button, the selected item is copied to the **Library** section.
- Library section.** This shows all your reference waveforms. Each one has a check box that controls whether or not the waveform appears on the display.
- Reference waveform.** Once enabled, a reference waveform 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 reference waveform (bottom) is a copy of Channel A.
- Axis control button.** Opens an axis scaling dialog allowing you to adjust the scale, offset and delay for this waveform.

## 4.26 Serial decoding

You can use PicoScope to decode data from a serial bus such as I<sup>2</sup>C 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.

1. [Select Tools > Serial Decoding](#)

Hover over packet data to view summary

"Graph" serial data

Decoding tab

"Table" serial data

Packet	Start Time	End Time	ID	RTR	IDE	DLC	Data	CRC	ACK
1	16.67 ns	206.2 μs	00 BE BE BE	0	1	4	7B 7E 9A 9C	37 EE	0
2	212.2 μs	308.4 μs	1 00	0	0	0		38 0A	0
3	314.3 μs	442.6 μs	1 01	0	0	2	01 03	56 2D	0
4	448.5 μs	666.6 μs	10 00 00 01	0	1	5	11 22 33 44 55	6A 65	0
5	672.6 μs	934.7 μs	12 34 56 78	0	1	8	11 22 33 44 55 66 77..	04 C2	0
6	940.7 μs	1.211 ms	15 97 EE B2	0	1	8	FF FF 00 00 EE EE 11..	21 6E	0
7	1.217 ms	1.369 ms	5 19	0	0	4	42 69 6C 6C	77 44	0
8	1.375 ms	1.645 ms	15 97 EE B2	0	1	8	AE 4F FF F1 02 72 D..	21 80	0

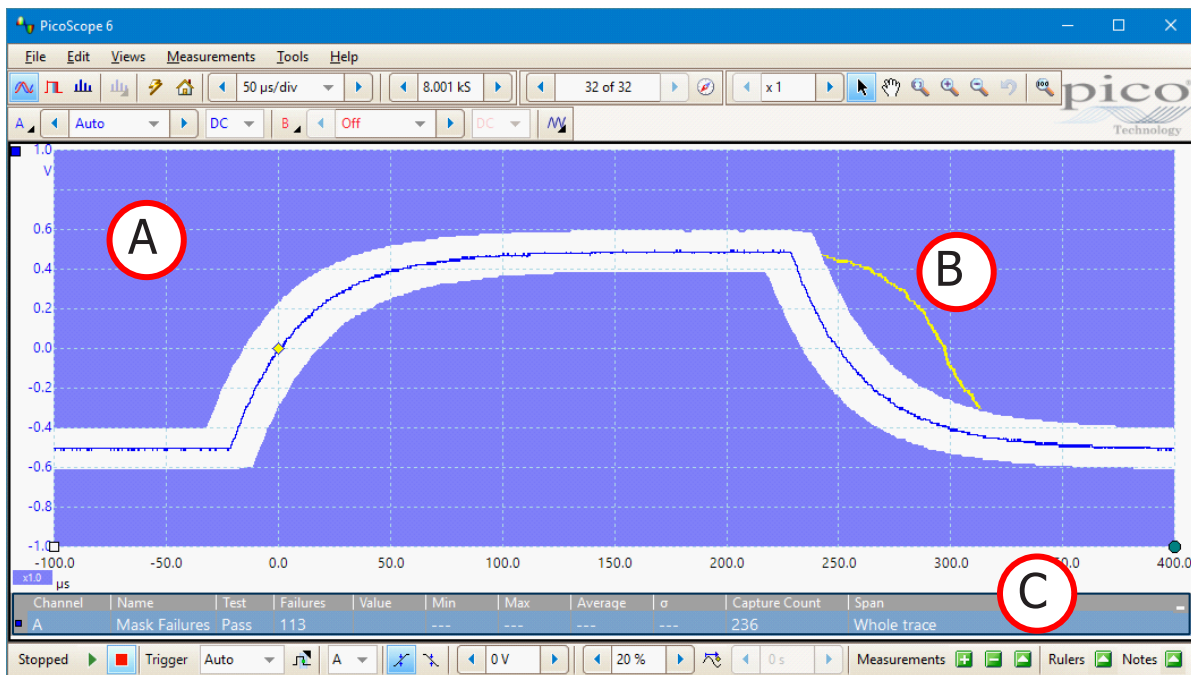
To start decoding your data, click [Tools > Serial Decoding](#) from the Menu bar. There are many [different protocols](#) available, and the data can be viewed in **Graph** or **Table** formats, or both. You can simultaneously decode multiple channels in different formats, and switch between them using the **Decoding tabs**.

For more information, see [Serial Decoding dialog](#).

## 4.27 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, or you can draw it manually. Mask limit testing is useful for spotting intermittent errors during debugging, and for finding faulty units during production testing. For step-by-step instructions, see [How to set up a Mask Limit Test](#).

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). Right-clicking the mask area and selecting the **Edit Mask** command takes you to the [Edit Mask dialog](#). You can change the mask colors in the [Colors page](#) of the [Preferences dialog](#) and add, remove, hide, display and save masks using the [Masks menu](#).

### (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) Measurements table

The number of failures since the start of the current scope run is shown in the [Measurements table](#). You can clear the failure count by stopping and restarting the capture using the [Stop and Go buttons](#). The measurements table can display [other measurements](#) at the same time as the mask failure count.

For detailed information on setting up a mask limit test, see [How to set up a Mask Limit Test](#).

## 4.28 Alarms

**Alarms** are actions that you can program PicoScope to execute when certain events occur. Use the **Tools > Alarms** command to open the [Alarms dialog](#), which configures this function.

The events that can trigger an alarm are:

- Capture - when the oscilloscope has [captured](#) a complete waveform or block of waveforms
- Buffers Full - when the [waveform buffer](#) becomes full
- Mask(s) Fail - when a waveform fails a [mask limit test](#)


The actions that PicoScope can execute are:

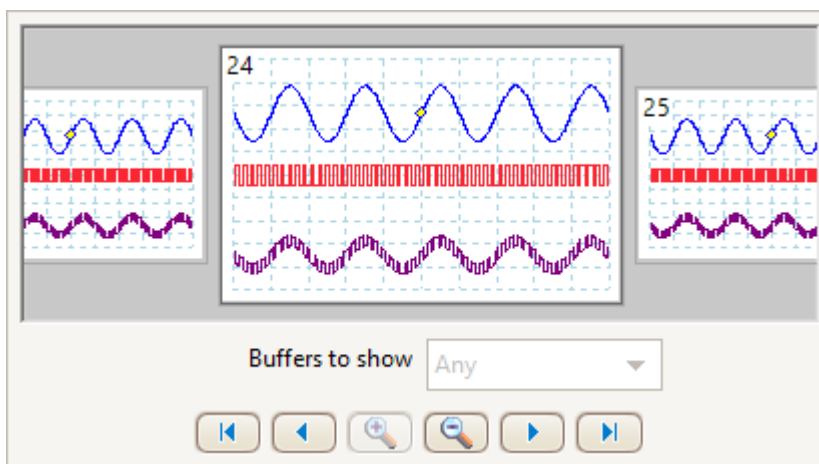
- Beep
- Play Sound
- Stop Capture
- Restart Capture
- Run Executable
- Save Current Buffer
- Save All Buffers
- Trigger Signal Generator

See [Alarms dialog](#) for more details.

## 4.29 Buffer Overview

The PicoScope waveform buffer can hold up to 10 000 waveforms, subject to the amount of available memory in the oscilloscope. The Buffer Overview helps you to scroll through the buffer quickly to find the waveform you want.

To begin, click the **Buffer Overview**  button in the [Buffer Navigation toolbar](#). This opens the Buffer Overview window:



Click on any one of the visible waveforms to bring it to the front of the overview for closer inspection, or use the controls:

**Buffers to show** If any of the channels has a [mask](#) applied, then you can select the channel from this list. The Buffer Overview will then show only the waveforms that failed the mask test on that channel.





Scroll to waveform number 1.



Scroll to the next waveform on the left.



Change the scale of the waveforms in the Buffer Overview view. There are three zoom levels:

**Large:** default view. One waveform fills the height of the window.

**Medium:** a medium-sized waveform above a row of small waveforms.

**Small:** a grid of small waveforms. Click on the top or bottom row of images to scroll the grid up or down.



Scroll to the next waveform on the right.

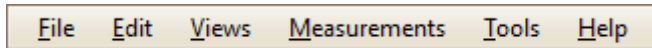


Scroll to the last waveform in the buffer. (The number of waveforms depends on the [Tools > Preferences > General > Maximum Waveforms](#) setting and on the type of scope connected).

Click anywhere on the main PicoScope window to close the Buffer Overview window.

## 5 Menus

**Menus** are the quickest way to get to PicoScope's main features. The **Menu bar** is always present at the top of the PicoScope main window, just below the window's title bar. You can click any of the menu items, or press the **Alt** key and then navigate to the menu using the arrow keys or by typing the underlined letter in the desired menu item.



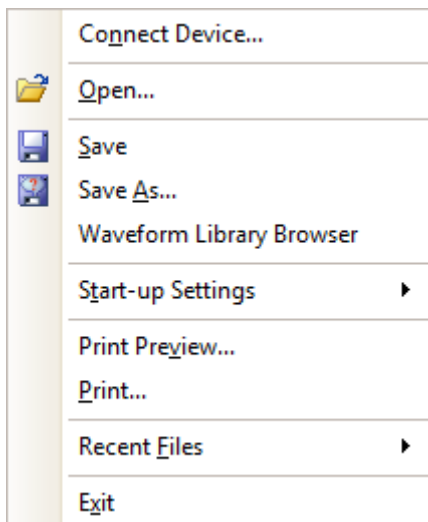
Click on a menu for more information

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

### 5.1 File menu

Location: [Menu bar](#) > **File**

Purpose: gives access to file input and output operations



**Connect Device.** This command appears only when there is no scope device connected or PicoScope is in Demo mode. It opens the [Connect Device dialog](#), which allows you to select the scope device you wish to use.



**Open...** Takes you to the [Open dialog](#), where you can open data, settings, math formula and reference waveform files.



**Save.** Saves all waveforms using the file name shown in the title bar. If you haven't entered a file name yet, the [Save As dialog](#) opens to prompt you for one.



**Save As.** Opens the [Save As dialog](#), which allows you to save the settings, waveforms, custom probes and math channels for all views in various formats. Only the waveforms for the mode currently in use ([Scope Mode](#) or [Spectrum Mode](#)) will be saved.

In [persistence mode](#), this command is called **Save Persistence As** and saves only the data for this mode.

**Waveform Library Browser** (PicoScope 6 Automotive only). Accesses the [Waveform Library Browser](#). Only available when a scope device is connected.

**Start-up Settings**. Opens the [Start-up Settings menu](#).

**Print Preview**. Opens the **Print Preview window**, which allows you to see how your workspace will be printed when you select the **Print** command.

**Print**. Opens a standard Windows Print dialog, which allows you to choose a printer, set printing options and then print the selected view.

**Recent Files**. A list of recently opened or saved files, with a thumbnail image of the waveform. This list is compiled automatically, but you can clear it using the **Recent Files** control on the [Options page](#) of the [Preferences dialog](#).

**Exit**. Close PicoScope without saving any data.

## 5.1.1 Connect Device dialog

Location: [File](#) > **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



The Connect Device dialog, as seen in the Test and Measurement software

Note: PicoScope 6 Automotive is only compatible with Pico's range of automotive oscilloscopes. Non-automotive oscilloscopes, such as the two illustrated above, will not appear in the list of available devices, as they cannot be used with the PicoScope 6 Automotive software.

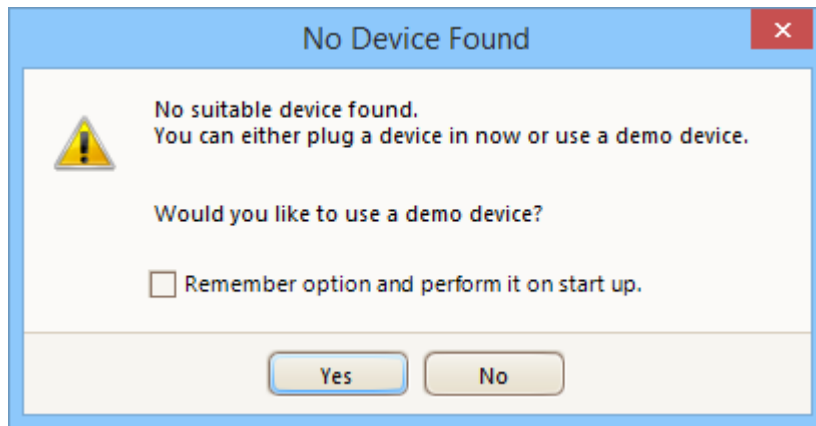
See [How to change to a different device](#) if you wish to switch to a different scope device later.

### 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 [toolbars](#) to set up the device and the [scope view](#) to display your signals.

### 5.1.1.1 Demo mode

If you start PicoScope without a device connected, the **No Device Found** dialog will automatically appear.



- Select **Yes** to start PicoScope 6 in Demo mode.
- Select **No** to start PicoScope 6 without a signal, allowing access to a limited selection of menu items. Select **File > Connect Device** to change to Demo mode or open another connected device.
- You can also select a Demo device from the [Connect Device dialog](#).

When in Demo mode, PicoScope adds a [Demo Signal Generator button](#)  to the toolbar. Use this button to set up test signals using the Demo device.

## 5.1.2 Open dialog

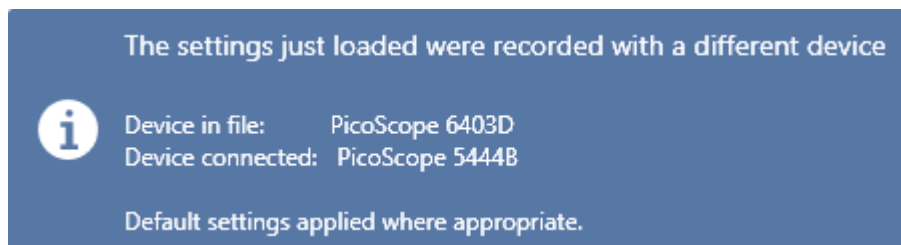
Location: [File > Open](#)

Purpose: allows you to open saved waveforms and settings (including custom probes and active math channels)

Allows you to select the file you want to open. You can open files in the following formats:

- |  |   |
|--|---|
| <b>Data files (.psdata)</b>                    | 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 <a href="#">Save</a> and <a href="#">Save As...</a> commands.      |
| <b>Settings files (.pssettings)</b>            | 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 <a href="#">Save</a> and <a href="#">Save As...</a> commands. |
| <b>Math Formula files (.psmaths)</b>           | Select this option to open a math channel exported from the <a href="#">Math Channels dialog</a> .  |
| <b>Reference Waveform files (.psreference)</b> | Select this option to open a waveform exported from the <a href="#">Reference Waveforms dialog</a> .  |

If you open a .psdata or .pssettings file that was saved using a different scope device from the one connected, PicoScope may need to modify the saved settings to suit the present device. A notification like the one below will appear in the bottom right corner of the PicoScope display:



Once you have opened a .psdata file, if you are using the default [keyboard map](#), you can use the **Page Up** and **Page Down** keys to cycle through all the other data and settings files in the same directory.

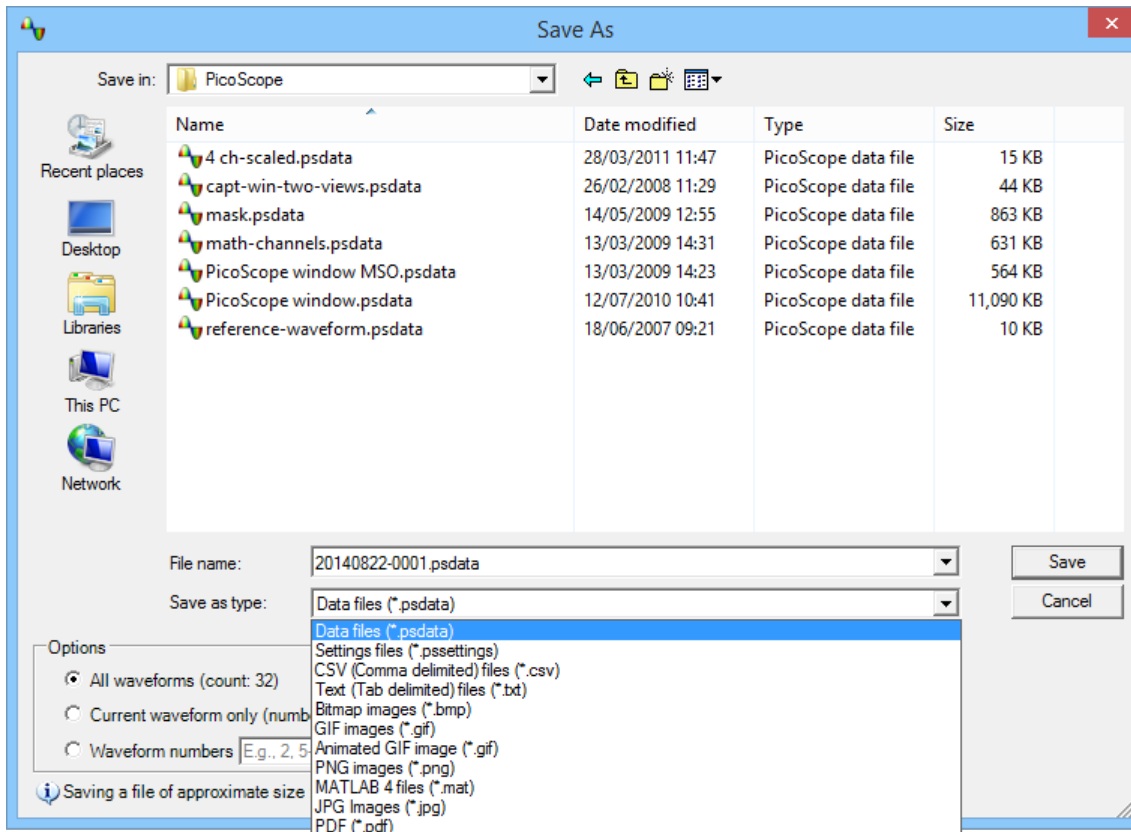
Note: channel information for .psdata files is displayed in the [Properties sheet](#). If a device is already connected when a .psdata file is loaded, the channel settings are changed to match the .psdata file where possible. The resulting settings are displayed on the Channels toolbar.

## 5.1.3 Save As... dialog

Location: [File > Save As...](#)

Purpose: allows you to save your waveforms and settings (including custom probes and active math channels) to a file in [various formats](#)

PicoScope 6 Automotive only: the [Options page](#) of the [Preferences dialog](#) allows you to set the [Details dialog](#) to appear before the **Save As** dialog, enabling you to record details about the vehicle and the customer.



Type your chosen file name in the **File name** box, and then select a file format in the **Save as type** box. You can save data in the following formats:

**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.

**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. See [Text formats](#) for more information.

<b>Text (Tab delimited) files (.txt)</b>	Stores waveforms as a text file with tab-separated values. The columns are the same as those in the CSV format. See <a href="#">Text formats</a> for more information.
<b>Bitmap images (.bmp)</b>	Stores pictures of the waveforms, <a href="#">graticule</a> and <a href="#">rulers</a> in Windows BMP format. The image is displayed in 16 million colors and is uncompressed.
<b>GIF images (.gif)</b>	Stores pictures of the waveforms, <a href="#">graticule</a> and <a href="#">rulers</a> in CompuServe GIF format. The image is displayed in 256 colors and is compressed.
<b>Animated GIF image (.gif)</b>	Creates an animated GIF that displays all of the waveforms in the buffer in sequence. Each waveform is formatted as in the single GIF format described above.
<b>PNG images (.png)</b>	Stores pictures of the waveforms, <a href="#">graticule</a> and <a href="#">rulers</a> in PNG format. The image is displayed in 16 million colors, with lossless compression.
<b>MATLAB 4 files (.mat)</b>	Stores the waveform data in <a href="#">MATLAB 4 format</a> .
<b>JPEG (.jpg)</b>	Stores pictures of the waveforms, <a href="#">graticule</a> and <a href="#">rulers</a> in JPG format. The image is displayed in 16 million colors, with lossy compression.
<b>PDF (.pdf)</b>	Creates a PDF document with a images of each of the waveforms in the buffer, along with their <a href="#">graticules</a> and <a href="#">rulers</a> . There is one image on each page, displayed in 16 million colors.

### Options

The first three options control what happens when the [waveform buffer](#) contains more than one waveform:

<b>All waveforms</b>	Saves all of the waveforms in the selected file format. If the file format is <code>.psdata</code> , all of the waveforms are collected in a single file. You can then load this into PicoScope and step through the waveforms using the <a href="#">buffer navigation controls</a> . If the selected file format does not support multiple waveforms, PicoScope creates a new directory containing multiple files.
<b>Current waveform only</b>	Saves the single waveform that is currently on view.
<b>Waveform numbers</b>	Saves the specified list or range of waveforms. Each waveform is identified by its index number. For example:  <div style="margin-left: 40px;">1, 2, 9, 10</div> <div style="margin-left: 40px;">2, 5-10</div>

### 5.1.3.1 File formats for exported data

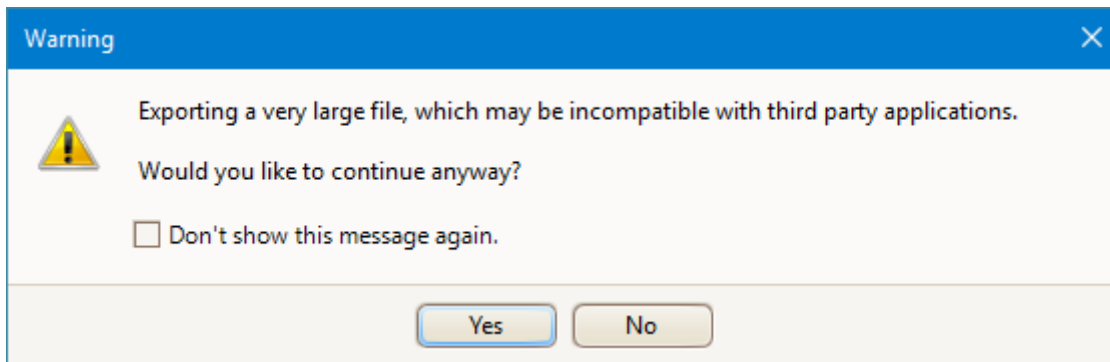
PicoScope 6 can export raw data in either text or binary format:

#### Text-based file formats

- Easy to read without special tools
- Can be imported into standard spreadsheet applications
- Files are very large if there are many samples in the data



There is no maximum file size when you export a text-based file, but not all editing applications can open large files. When you save a text-based file with more than 1 million values per channel, PicoScope 6 will display the following warning:



### [Text file format details](#)

#### Binary file format

- Files remain relatively small and can even be compressed in some situations
- Either a special application is required to read the files or the user must write a program to read the data from the file
- Recommended for large data sets

### [Binary file format details](#)

#### Data types for storing PicoScope 6 data

Regardless of whether the data types were loaded from a binary file or from a text-based file, we recommend the following data formats for storing the values loaded from a PicoScope 6 data file:

- Sampled data (such as voltages) should use 32-bit single-precision floating-point data types
- Times should use 64-bit double-precision floating-point data types

#### 5.1.3.1.1 Text formats

Text-format [files exported by PicoScope 6](#) are encoded in [UTF-8](#) format by default. This is a popular format which is capable of representing a huge range of characters, whilst still retaining some compatibility with the ASCII character set if only standard Western European characters and numbers are used in the file.

#### CSV (comma-separated values)

CSV files store data in the following format:

```
Time, Channel A, Channel B
(µs), (V), (V)
-500.004, 5.511, 1.215
-500.002, 4.724, 2.130
-500, 5.552, 2.212
...
```

There is a comma after each value on a line to represent a column of data and a carriage return at the end of the line to represent a new row of data.

### Tab-delimited

Tab-delimited files store data in the following format:

Time ( $\mu$ s)	Channel A (V)	Channel B (V)
500.004	5.511	1.215
-500.002	4.724	2.130
-500	5.552	2.212
...		

The files have a tab character after each value on a line to represent a column of data and a carriage return at the end of the line to represent a new row of data.

#### 5.1.3.1.2 Binary formats

PicoScope 6 can [export data](#) in version 4 of the .mat binary file format. This is an open format and the full specification is available as PDF documentation from the [www.mathworks.com](http://www.mathworks.com) website. PicoScope 6 saves data into the MAT-file in a specific way, which is detailed below.

Note: data arrays saved in version 4 of the .mat file format must not exceed the maximum size of 100 million elements. MATLAB cannot read arrays with more elements than this in this format.

### Importing into MATLAB

Load the file into your workspace using this syntax:

```
load myfile
```

Each channel's data is stored in an array variable named by the channel. So, the sampled data for channels A to D would be in four arrays named **A**, **B**, **C** and **D**.

There is only one set of time data for all channels and this is loaded in one of two possible formats:

- A start time, an interval and a length. The variables are named **Tstart**, **Tinterval** and **Length**.
- An array of times (sometimes used for ETS data). The time array is named **T**.

If the times are loaded in as **Tstart**, **Tinterval** and **Length** then you can use the following command to create the equivalent array of times:

```
T = [Tstart : Tinterval : Tstart + (Length - 1) * Tinterval];
```

Note: the size of the largest file that MATLAB can open depends on the resources of the computer. It is therefore possible for PicoScope to create a MATLAB file that some installations of MATLAB would be unable to open. Please be aware of this risk when saving critical data.

The other three variables used in the file are:

<b>ExtraSamples</b>	the number of samples added, for example when filtering or resolution enhancement is enabled
<b>RequestedLength</b>	the number of samples. This will be the same as <b>No. samples</b> in the PicoScope 6 <a href="#">Properties sheet</a> . Note that if the signal is not acquired in its entirety, this will be different from <b>Length</b> .
<b>Version</b>	the exact version of PicoScope 6 used to export the file

## Exploring the file format

The full file specification, available from [www.mathworks.com](http://www.mathworks.com), is comprehensive so this guide does not describe the entire format. Instead, this guide describes enough of the format to allow you to get data from the file and use it in your own program.

The variables described above (under [Importing into MATLAB](#)) are stored in a series of data blocks, each preceded by a header. Each variable has its own header and data block and the corresponding variable names are stored with them (such as **A**, **B**, **Tstart**). The following sections describe how to read each variable from the file.

The order of the data blocks is not specified, so programs should look at the variable names to decide which variable is currently being loaded.

## The header

The file consists of a number of data blocks preceded by 20-byte headers. Each header contains five 32-bit integers (as described in the table below).

Bytes	Value
0 to 3	Data format (0, 10 or 20)
4 to 7	Number of values
8 to 11	1
12 to 15	0
16 to 19	Name length

## Data format

The **Data format** in bytes 0 to 3 describes the type of numerical data in the array.

Value	Description
0	Double (64-bit floating point)
10	Single (32-bit floating point)
20	Integer (32-bit)

## Number of values

The **Number of values** is a 32-bit integer describing the number of numerical values in the array. This value may be 1 for variables that only describe one value; but for arrays of samples or times, expect this to be a large number.

## Name length

The **Name length** is the length of the name of the variable as a null-terminated 1-byte per character ASCII string. The last null terminating character ( $\backslash 0$ ) is included in the **Name length** so if the variable name is **Tstart** (same as **Tstart\0**) then the name length will be 7.

## The data block

The data block begins with the name of the variable (such as **A**, **Tinterval**) and you should read in the number of bytes described by the *Name length* part of the header (not forgetting that the last byte in the string is  $\backslash 0$  if your programming language needs to take account of this).

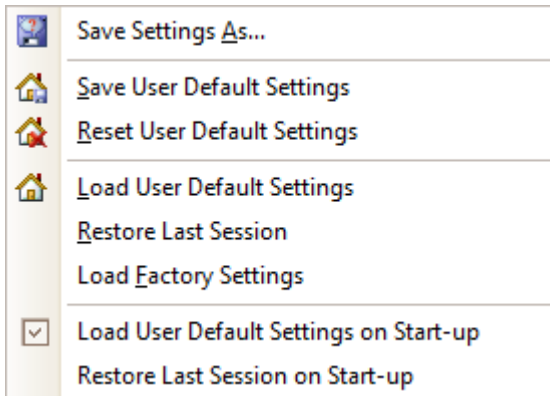
The remaining part of the data block is the actual data itself, so read in the number of values described in the *Number of values* part of the header. Remember to take account of the size of each value as described in the 'Data format' part of the header.

Channel data such as voltages, in variables such as **A** and **B**, are stored as 32-bit single-precision floating-point data types. Times such as **Tstart**, **Tinterval** and **T** are stored as 64-bit double-precision floating-point data types. **Length** is stored as a 32-bit integer.

## 5.1.4 Start-up Settings menu

Location: [File](#) > **Start-up Settings**

Purpose: allows you to load, save and restore the PicoScope 6 startup settings



- Save Settings As...** Stores all the current settings (for example collection time, input range, trigger), which can then be opened on any computer running PicoScope 6. This allows you to create a library of different setups.
- Save User Default Settings** Saves the current settings as the default. This means you only have to configure the oscilloscope once in order to use these settings every time you open PicoScope 6.
- Reset User Default Settings** This replaces your saved default settings with the original factory settings.
- Load User Default Settings** Resets all settings to the saved default. You can also use the home button on the toolbar to achieve this.
- Restore Last Session** This option returns the settings to those from your last session.
- Load Factory Settings** Returns the settings to the factory original.
- Load User Default Settings on Start-up** Selecting this option makes sure that the next time you start up PicoScope 6, the saved user default settings will be applied.
- Restore Last Session on Start-up** You should check this box if you want PicoScope 6 to open with the settings you used for your last session.

## 5.1.5 Waveform Library Browser

Location: [File](#) > **Waveform Library Browser**

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

Applicability: PicoScope 6 Automotive only  
Oscilloscope must be connected

The screenshot shows the Waveform Library browser interface. On the left, there are search options categorized into Vehicle details, Engine details, and Channel details. The main area displays two waveform entries. The top entry is for a 2009 Ford Falcon, showing a waveform and a 'Car Details' sidebar with information like Make/Model, Year, Transmission, and Engine code. Below the waveform is a list of channels, including 'Camshaft sensor (Inductive)' with status 'Good' and 'Bad'. The bottom entry is for a 2002 Volkswagen Bora, showing a waveform and a 'Car Details' sidebar with information like Make/Model, Year, Transmission, and Engine code. Below the waveform is a list of channels, including 'Engine Speed Sensor G28' with a note about a math conversion.

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 6.

### Login details

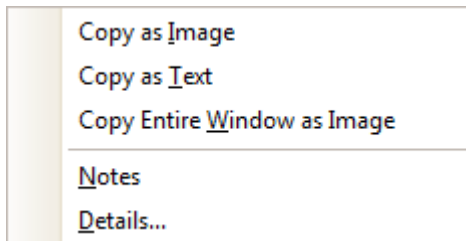
When you open the Waveform Library, a login page appears. Enter your username and password, and the software will take you to the page shown below.

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.

## 5.2 Edit menu

Location: [Menu bar](#) > **Edit**

Purpose: gives access to the clipboard-related and note-editing functions



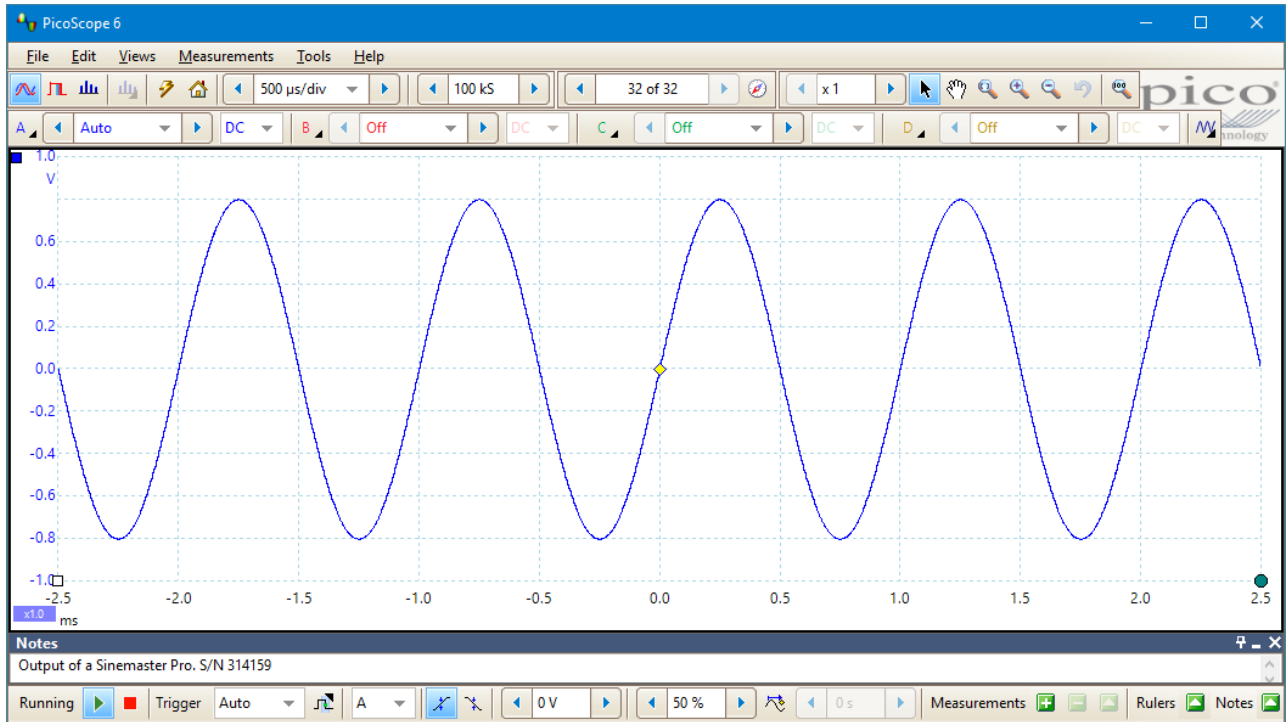
- |                                    |   |
|------------------------------------|---|
| <b>Copy as Image</b>               | Copies the active view to the clipboard as a bitmap. You can then paste the image into any application that accepts bitmap images.  |
| <b>Copy as Text</b>                | Copies the data in the active view to the clipboard as text. You can paste the data into a spreadsheet or other application. The text format is the same as that used by the <a href="#">Save As dialog</a> when you select the .txt file type.   |
| <b>Copy Entire Window as Image</b> | This copies a picture of the PicoScope window to the clipboard. It is provided as an alternative equivalent to pressing <b>Alt + Print Screen</b> for users of laptops without a <b>Print Screen</b> key. You can paste the picture into any application that can display pictures, such as a word processor or desktop publishing program. |
| <b>Notes</b>                       | Opens a <a href="#">Notes area</a> at the bottom of the PicoScope window. You can type or paste your own notes in this area.  |
| <b>Details...</b>                  | PicoScope 6 Automotive only. Opens the <a href="#">Details dialog</a> , which allows you to enter or view vehicle details, customer details, <a href="#">notes</a> and <a href="#">channel labels</a> .   |

## 5.2.1 Notes

Location: [Edit > Notes](#)

[Trigger toolbar > Other buttons > Notes](#) button

Purpose: provides a text box for typing your own notes



You can place a **Notes** area at the bottom of the PicoScope window, 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.

## 5.2.2 Details dialog

Location: [File](#) > [Save As](#)

[Edit](#) > **Details**

Purpose: allows you to record channel labels and vehicle and customer details, before saving a file

Applicability: PicoScope 6 Automotive only

**Details**

Vehicle Customer

Recent Vehicles: [Select a recent vehicle]

VIN / ID Code

Engine Code

Make [Select Make] ...

Primary Fuel

Model  ...

Secondary Fuel

Generation

Cylinders

Year

Configuration

Transmission

Capacity (L)

Capacity (cu in)

Test Conditions

ECU Make [Select Make] ...

ECU Model

<b>A</b>	Injector current	looks like stuck injector	Bad
<b>B</b>	Injector voltage	normal	Good
<b>C</b>	Ignition coil primary	ask Phil	Unknown
<b>D</b>	Ignition coil secondary	HT too low	Bad

Notes

OK Cancel



You can enter information about the vehicle under test in the form shown above. You can also view and edit your [notes](#) and [channel labels](#) at the bottom of the PicoScope window by clicking the **Notes** and **Channel Labels** buttons in the Trigger toolbar.

Click the **Customer** tab to note down the customer's details. All of this information will be saved in the .psdata file. You can then view it in PicoScope using the [Edit > Details](#) menu command.

### 5.2.2.1 Channel Labels

Location: [Edit > Details](#)

[Trigger toolbar](#) > [Other buttons](#) > **Channel Labels** button

Purpose: allows you to supply and view information on the waveforms

Applicability: PicoScope 6 Automotive only

Channel Labels			
A	Injector current	looks like stuck injector	Bad
B	Injector voltage	normal	Good
C	Ignition coil primary voltage	ask Phil	Unknown
D	Ignition coil secondary voltage	HT too low	Bad

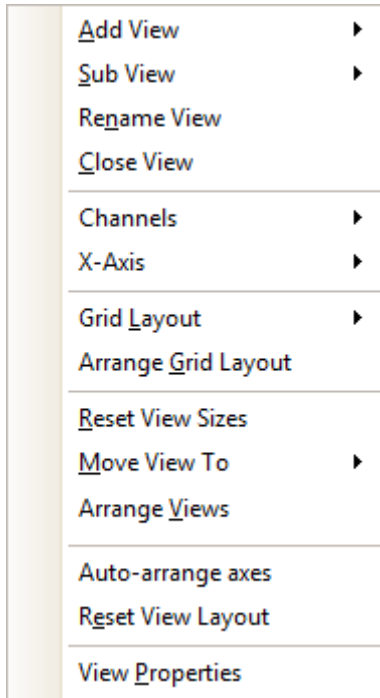
Channel labels can be displayed at the bottom of the PicoScope window.

- Label** Select a standard label from the drop-down list or type any text in the box.
- Description** Type any text in the box to describe the waveform.
- Status** Select **Good**, **Bad** or **Unknown**. This helps other engineers to identify whether the waveform is from a working engine or a faulty one.

## 5.3 Views menu

Location: [Menu bar](#) > **Views**, or right-click on a [view](#)

Purpose: controls the layout of the current [view](#), which is a rectangular area of the PicoScope window that display scope, spectrum or other kinds of data



The contents of the Views menu may vary depending on where you click and how many views are open. If the current view contains a [Measurements table](#), a combined [Measurements menu](#) and Views menu will appear.

### Add View

Add a view of the selected type ([scope](#), [XY](#) or [spectrum](#)). In automatic grid layout mode (the default), PicoScope rearranges the grid to make room for the new view, up to a limit of nine views. Any further views will be added as tabs in existing [viewports](#). If you have selected a fixed grid layout, PicoScope will not change it.

### Sub View

[Mixed signal oscilloscopes](#) only: switch the [analog view](#) and [digital view](#) on and off independently

### Rename View

Change the standard *Scope* or *Spectrum* label to a title of your choice

### Close View

Remove a view from the PicoScope window. In automatic grid layout mode (the default), PicoScope rearranges the grid to make the best use of the remaining space. In fixed grid layout mode (if you have selected a fixed grid layout), PicoScope will not change the grid.

### Channels

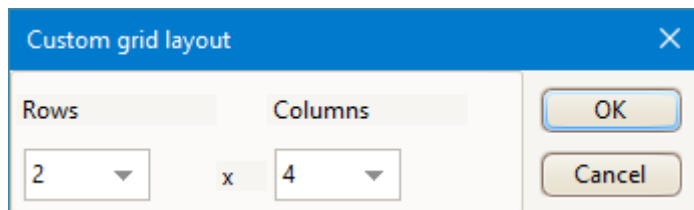
Select which channels are visible in the current view. Each view, when created, shows all the input channels, but you can switch them on and off using this command. Only the input channels that are enabled (not set to **Off** in the [Channel Setup Toolbar](#)) are available for viewing. The **Channels** menu also list [math channels](#) and [reference waveforms](#). You can select up to 8 channels in any view.

<b>X-Axis</b>	Select any suitable channel to drive the X axis. By default, the X axis represents time. If you select an input channel instead, the scope view will become an <a href="#">XY view</a> that plots one input against another. A quicker way to create an XY view is to use the <b>Add View</b> command (see above).
<b>Grid Layout</b>	The grid layout defaults to <i>Automatic</i> mode, in which PicoScope automatically arranges viewports in a grid. You can also select one of the standard grid layouts or create a custom layout, which PicoScope will preserve as you add or remove views.
<b>Arrange Grid Layout</b>	Adjust the grid layout to fit the number of views. Moves any tabbed views to empty viewports. Overrides any previous choice of grid layout.
<b>Reset View Sizes</b>	If you have resized any of the views by dragging the vertical or horizontal separator bars between viewports, this option resets all the viewports to their original sizes.
<b>Move View To</b>	Move a view to a specified viewport. You can achieve the same effect by dragging the view by its name tab and dropping it in a new viewport: see <a href="#">How to move a view</a> .
<b>Arrange Views</b>	If multiple views are stacked in the same viewport, move them into their own viewports
<b>Auto-arrange axes</b>	Scale and offsets all traces to fill the view and avoid overlaps
<b>Reset View Layout</b>	Reset the scale factor and offset of the selected view to their default values
<b>View Properties</b>	Show the <a href="#">Properties sheet</a> , which lists scope settings that are normally hidden

### 5.3.1 Custom grid layout dialog

Location: [Views menu](#) > **Grid Layout** > **Custom layout...**

Purpose: if the **Grid Layout** section of the [Views menu](#) does not contain the layout you want, this dialog gives further options

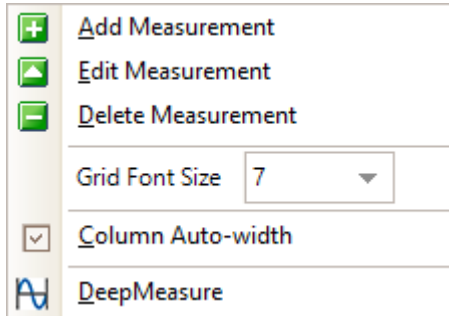


You can lay out the [view](#) grid with any number of rows and columns up to 4 by 4. You can then drag the views to different locations in the grid.

## 5.4 Measurements menu

Location: [Menu bar](#) > Measurements

Purpose: controls the [Measurements table](#)



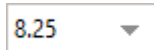
**Add Measurement.** Adds a row to the [measurements table](#), and opens the [Add Measurement dialog](#).



**Edit Measurement.** This takes you to the [Edit Measurement dialog](#). You can also edit a measurement by double-clicking on a row of the [measurements table](#).



**Delete Measurement.** Removes the selected row from the [measurements table](#).



**Grid Font Size.** Sets the font size for the entries in the [measurements table](#).

**Column Auto-width.** If this box is checked, the columns of the [measurements table](#) will continually adjust to fit the contents whenever the table changes. Click again to release the button.



**DeepMeasure™.** Allows you to make and analyze a number of automatic measurements over up to 1 million individual cycles.

You can also access the **Add Measurement**, **Edit Measurement** and **Delete Measurement** commands from the [Measurements buttons](#), on the [Trigger toolbar](#), or by right-clicking on a [view](#).

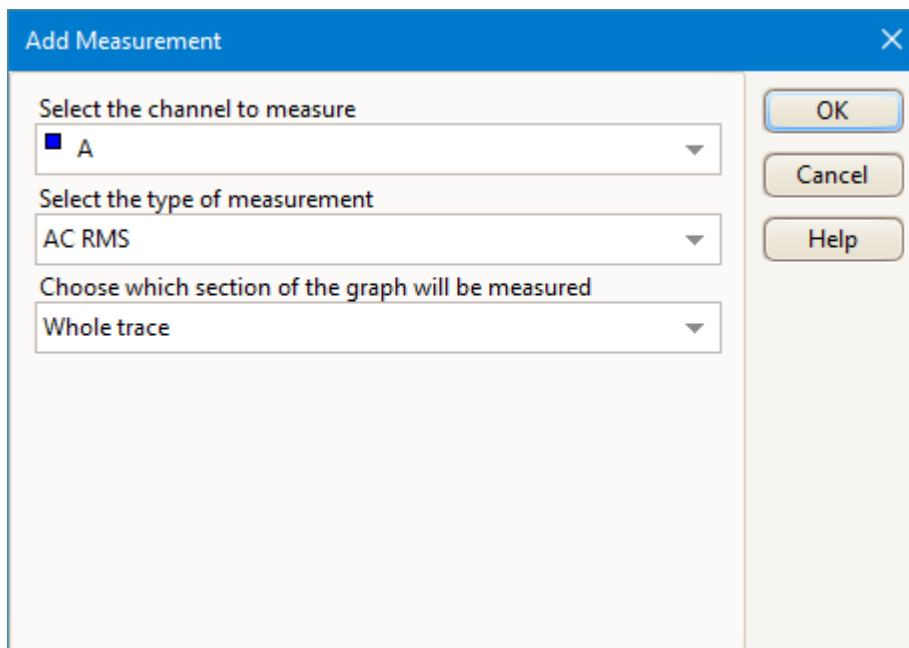
## 5.4.1 Add/Edit Measurement dialog

Location: [Trigger toolbar](#) > [Measurements buttons](#) >  Add Measurement or  Edit Measurement button

[Measurements menu](#) >  Add Measurement or  Edit Measurement button

Double-click a measurement in the [measurements table](#)

Purpose: allows you to add a measurement of a waveform to the selected [view](#), or edit an existing measurement



If this is the first measurement for the active view, PicoScope will create a new [measurements table](#); otherwise, it will add the new measurement to the bottom of the existing table. PicoScope automatically refreshes the measurement every time it updates the waveform.

### Select the channel to measure

Choose which of the [scope device's](#) channels to measure.

### Select the type of measurement

PicoScope can calculate a wide range of measurements for waveforms. See [scope measurements](#) (for use with [scope views](#)) or [spectrum measurements](#) (for use with [spectrum views](#)) for a list.

### 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.

### Measurement Options

Some measurements have additional settings to choose from: see [Scope measurement options](#) and [Spectrum measurement options](#) for more information.

### 5.4.1.1 Scope measurement options

#### Threshold and hysteresis (Edge Count, Falling Edge Count and Rising Edge Count only)

The screenshot shows the 'Add Measurement' dialog box with the following settings:

- Select the channel to measure:** A
- Select the type of measurement:** Edge Count
- Choose which section of the graph will be measured:** Whole trace
- Measurement Options:**
  - Threshold:**  Automatic,  Use Signal Rulers
  - Hysteresis:** 1.5 %

Buttons: OK, Cancel, Help

Use these options to control the threshold method and the degree of [hysteresis](#) for Edge Count, Falling Edge Count and Rising Edge Count measurements.

#### Threshold (Falling Rate, Fall Time, Rise Time and Rising Rate only)

The screenshot shows the 'Add Measurement' dialog box with the following settings:

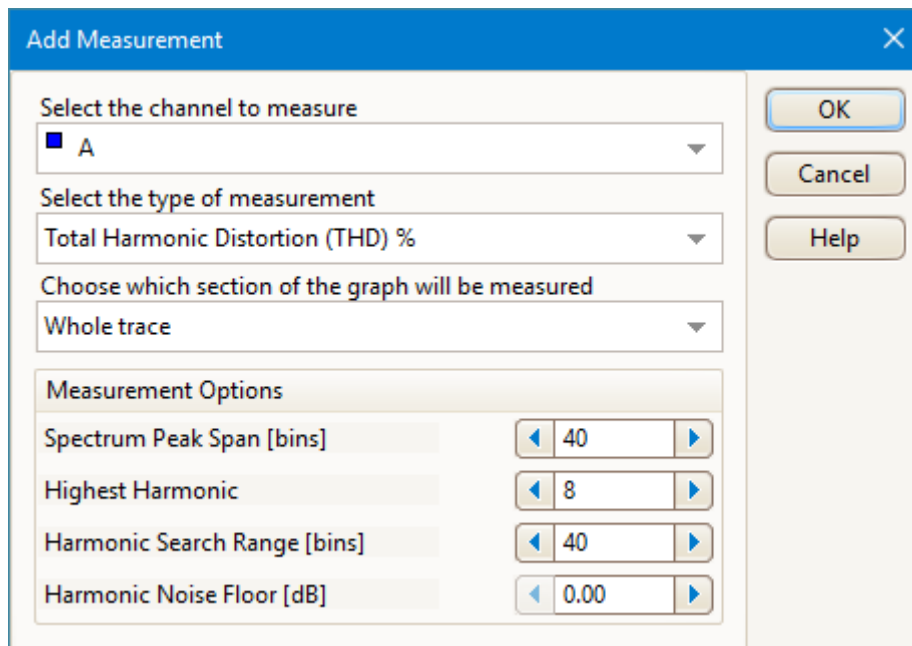
- Select the channel to measure:** A
- Select the type of measurement:** Falling Rate
- Choose which section of the graph will be measured:** Whole trace
- Measurement Options:**
  - Threshold:** Threshold 80%/20%

Buttons: OK, Cancel, Help

Falling Rate, Fall Time, Rise Time and Rising Rate measurements can be made with different thresholds. Use this control to select one.

When comparing rise and fall times with manufacturers' specifications, it is important to use the same thresholds for all measurements.

### 5.4.1.2 Spectrum measurement options



#### **Spectrum Span (all [Spectrum measurements](#))**

When measuring peak-related parameters in a [spectrum view](#), PicoScope can search for a peak near the specified [frequency ruler](#) location. This option tells PicoScope how many frequency bins to search.

#### **Harmonic control options (Total Harmonic Distortion [all variants], SINAD and Signal to Noise Ratio [SNR] only)**

The following options apply to distortion measurements. You can specify which harmonics PicoScope uses for these measurements.

##### **Highest Harmonic**

The highest harmonic to include when calculating distortion power.

##### **Search Range**

The number of frequency bins to search, centered on the expected frequency, when looking for a harmonic peak.

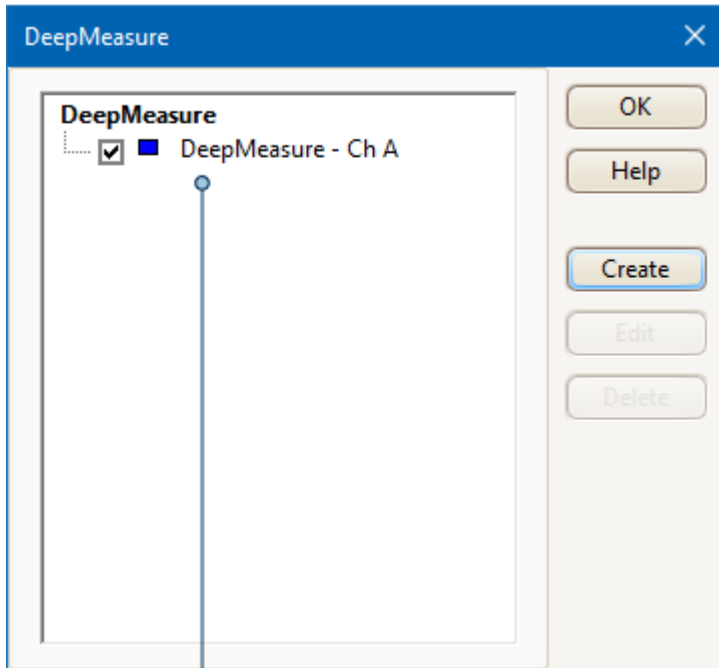
##### **Harmonic Noise Floor**

The level in dB above which signal peaks will be counted as harmonics.

## 5.4.2 DeepMeasure™ dialog

Location: [Measurements](#) > **DeepMeasure**

Purpose: lets you set up which channel to use for [DeepMeasure](#)



List of instances of DeepMeasure

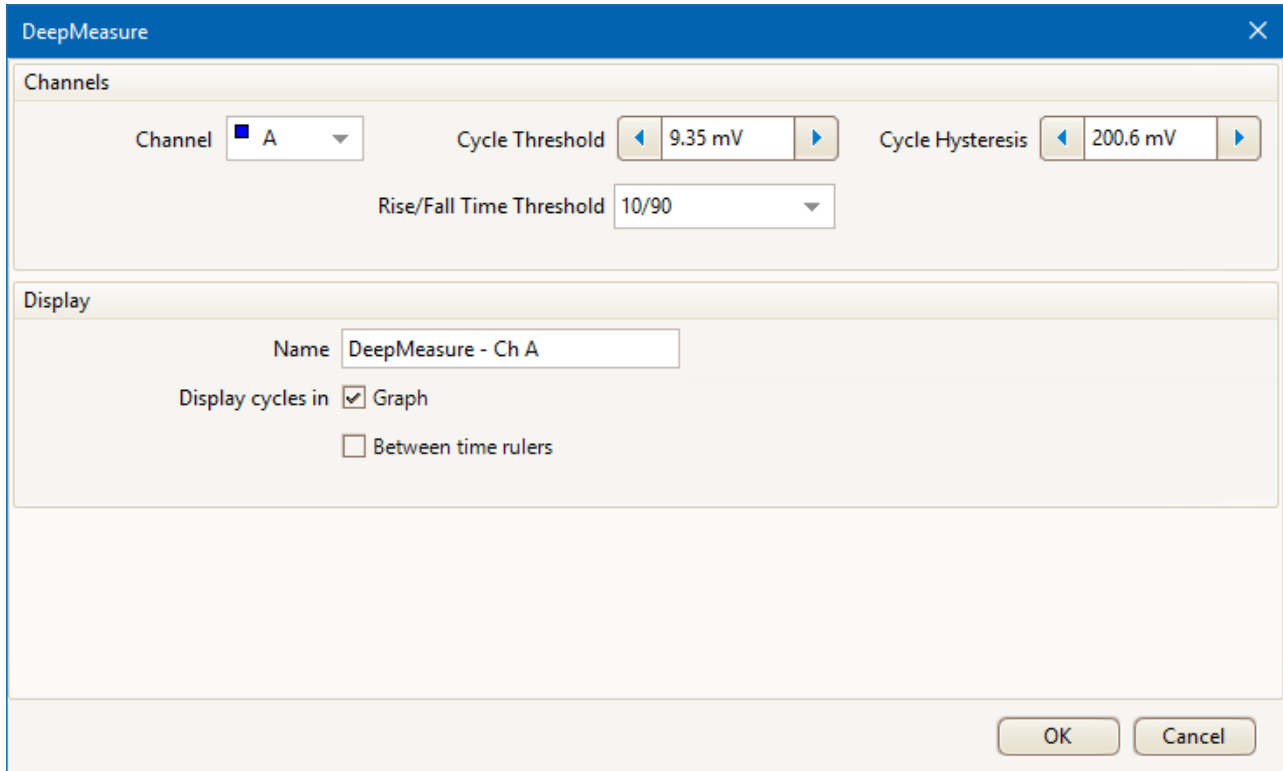
### Adding a new instance of DeepMeasure

1. Make sure the channel you want to measure is enabled, using the [Channels toolbar](#).
2. Click the **Create** button.
3. This opens the [DeepMeasure Settings](#) dialog: use this to select which channel to measure and define the waveform cycles and how you would like to view the data.
4. Click **OK**. The new instance of DeepMeasure will appear in the list, as shown above, where you can edit or delete it, or add more.



### 5.4.2.1 DeepMeasure™ Settings

Use this dialog to set up the channel and waveform characteristics to measure, and set up the display options.



#### Channels

- Data** The input channel to measure
- Cycle Threshold** The signal level that defines the beginning and end of the cycle. The signal will cross this threshold three times in every cycle.
- Cycle Hysteresis** Eliminates the effect of noise on the **Cycle 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.
- Rise/Fall Time Threshold** High and low measurement thresholds as a percentage of waveform height, excluding ringing. Choose lower and upper step values of 10/90 or 20/80.

#### Display

- Name** Enter a name for this instance of DeepMeasure. This is optional, but can be useful if you set up more than one instance for the same channel.
- If you do not enter a name, PicoScope will generate one automatically.
- Display cycles in** **Graph:** Enables the [DeepMeasure overlay](#).
- Between time rulers:** Only measures the section of waveform shown between the [time rulers](#).

### 5.4.2.2 DeepMeasure™ data window

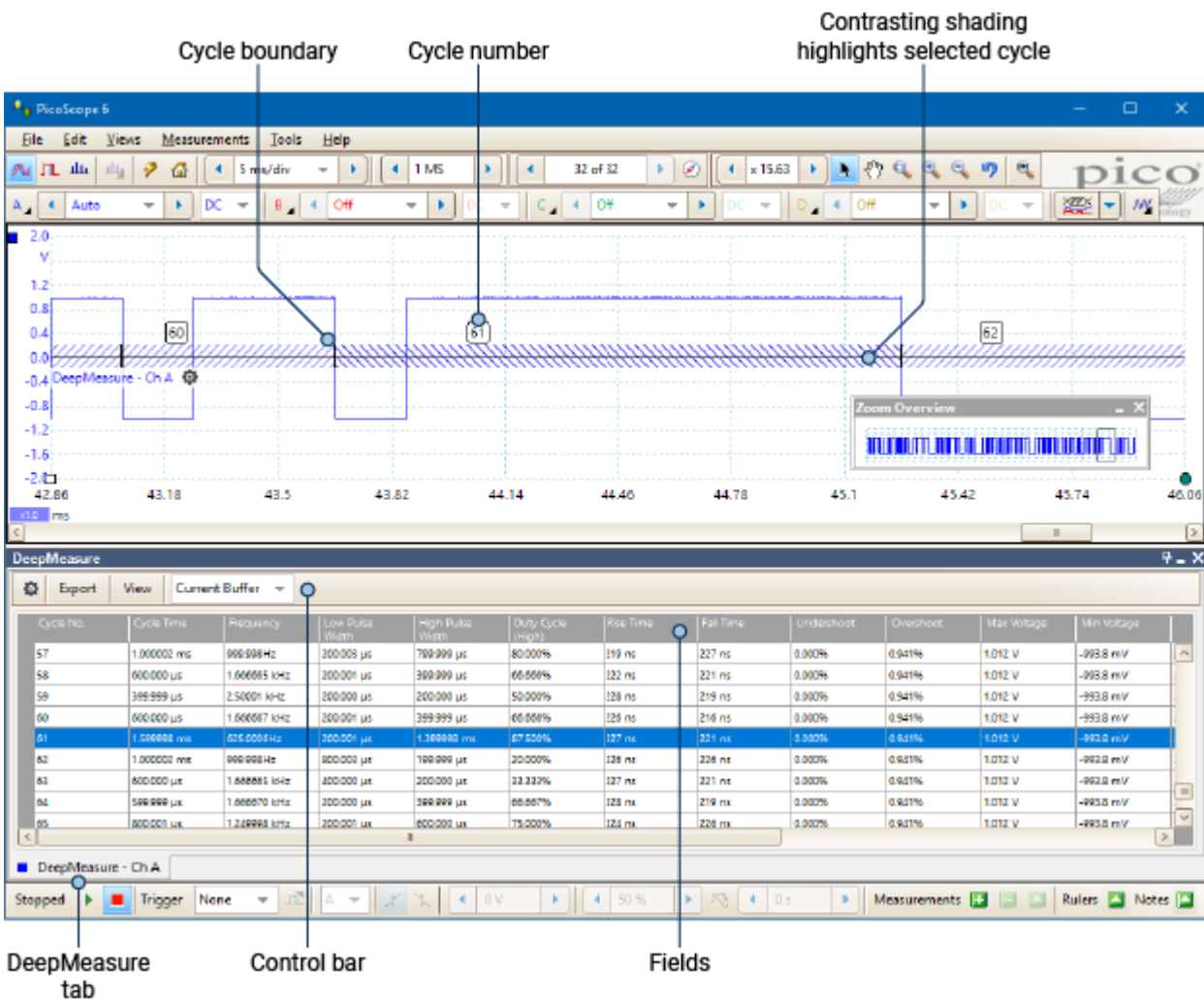
When you use DeepMeasure, PicoScope automatically displays the measurement data in a table below your waveform, as shown below. You can sort this data by any of the fields.

Double-click one of the waveform cycles in the scope view to zoom in on it and go to the corresponding row in the table, or double-click a row in the table to zoom in on the selected cycle in the scope view.

#### DeepMeasure overlay

This feature highlights individual waveform cycles in the graph.

- Boundaries between cycles are marked with thick black lines.
- Each cycle is numbered so you can easily match the waveforms in the graph to the entries in the data table.
- Hover the mouse pointer over any cycle to display a summary of its measurements.
- Double-click any cycle in the graph or the data table to highlight it in the graph and zoom on it, and to highlight it in the data table.



#### DeepMeasure tabs

Each instance of DeepMeasure is displayed in a separate table. Use these tabs to switch between them.

**Control bar**

Open the [DeepMeasure Settings](#) dialog

**Export**

Save the decoded data in a link file, as an OpenDocument Spreadsheet (ODS) or Comma-Separated Values (CSV) text file

**View**

**Fields:** Choose which data fields are displayed

**Grid Font Size:** Change the size of the characters in the table

**Buffer**

Choose whether to display packets from the **Current Buffer** or **All Buffers**

**Fields**

Each field in the data table represents one type of measurement.

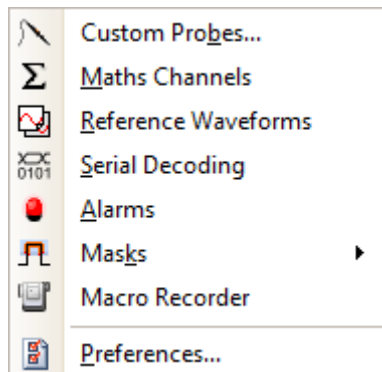
Click and drag the edges of each column to change the width, and make viewing the field data clearer.

Click the field header to sort the table by that field, in ascending order. Click again to sort in descending order.

## 5.5 Tools menu

Location: [Menu bar](#) > **Tools**

Purpose: gives access to assorted tools for signal analysis



[Custom Probes](#). Define new probes and copy, delete, move and edit existing ones.



[Math Channels](#). Add or edit a channel that is a mathematical function of one or more other channels.



[Reference Waveforms](#). Create, load or save a channel as a copy of an existing channel.



[Serial Decoding](#). Decode and display the contents of serial data streams such as CAN bus.



[Alarms](#). Specify actions to be taken on certain events.



[Masks](#). Perform [mask limit testing](#) on a waveform. This detects when the waveform departs from a specified shape.



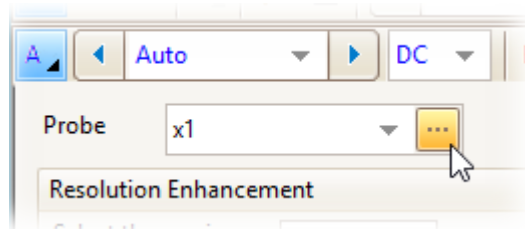
**[Macro Recorder](#)**. Save a frequently-used sequence of operations.



**[Preferences](#)**. Set various options that control PicoScope's behavior.

## 5.5.1 Custom Probes dialog

Location: [Tools > Custom Probes](#)  
or open it from the [Channel Options menu](#):



Purpose: allows you to set up and edit [custom probes](#)

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

### Understanding the probe list

All the probes that PicoScope knows about are listed under three main headings: **Built-in**, **Library** and **Loaded**. The probe list is preserved between sessions, so that PicoScope will never forget your custom probes unless you delete them.

- **Built-in probes.** The built-in probes are supplied by Pico Technology and do not change unless you download an authorized update from us. As a safeguard, PicoScope does not allow you to edit or delete these probes. If you want to modify one of them, you can copy it to your library by clicking **Duplicate**, and then edit the copy in your library.
- **Library probes.** These are the probes that you have created using any of the methods described in this topic. You can edit, delete or duplicate any of these probes by clicking the appropriate button in this dialog.
- **Loaded probes.** Probes in PicoScope data files (.psdata) or settings files (.pssettings) that you have opened appear here until you copy them to your library. You cannot edit or delete these probes directly, but you can click **Save to Library** to copy them to your library where you can edit them.

### Adding a new probe to your library

There are three ways to create a new probe:

- Use the **Duplicate** button as described above.
- Click **New Probe...** to define a new probe.
- Click **Import** to load a probe definition from a \*.psprobe file and add it to your library. These files are normally supplied by Pico, but you can also create your own by defining a new probe and then clicking **Export**.

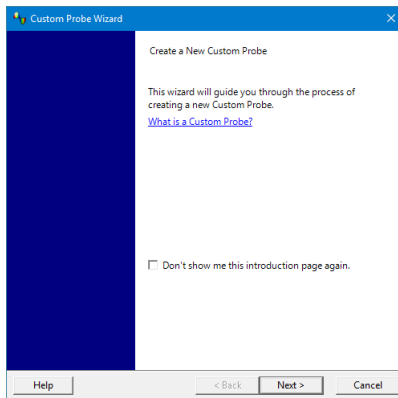
The last two methods open the [Custom Probe wizard](#) to guide you through the probe definition process.

## 5.5.1.1 Custom Probe wizard

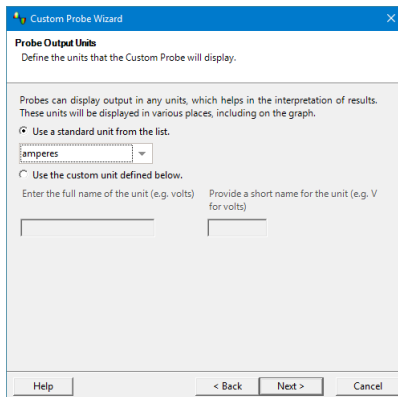
Location: [Custom Probes dialog](#) > **New Probe**

Purpose: allows you to define [custom probes](#) and set up custom ranges

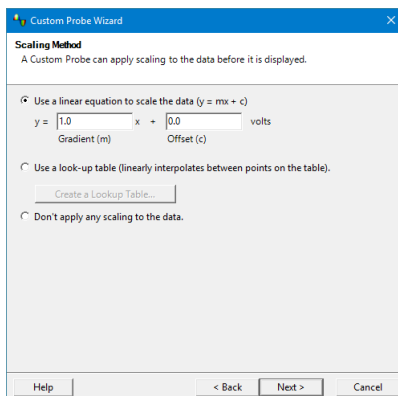
The first dialog in the series is either the [Custom Probe wizard introduction dialog](#) or the [Edit an existing Custom Probe dialog](#).



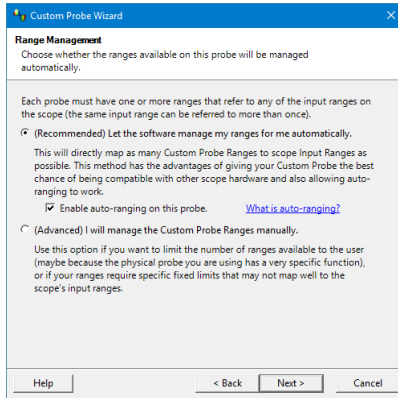
1. [Introduction](#) or [Edit](#)



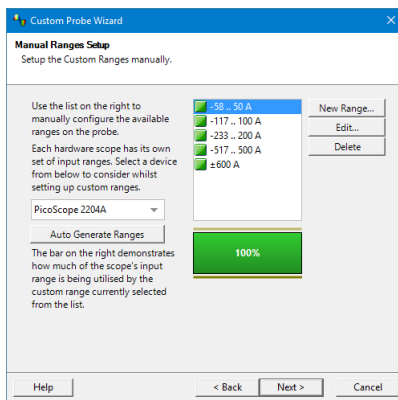
2. [Probe output units](#)



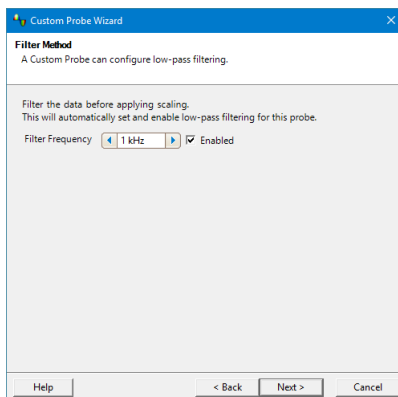
3. [Scaling method](#)



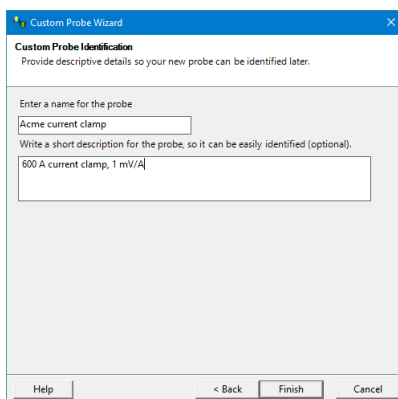
4. [Range management](#)



5. [Manual ranges setup](#)



6. [Filter method](#)

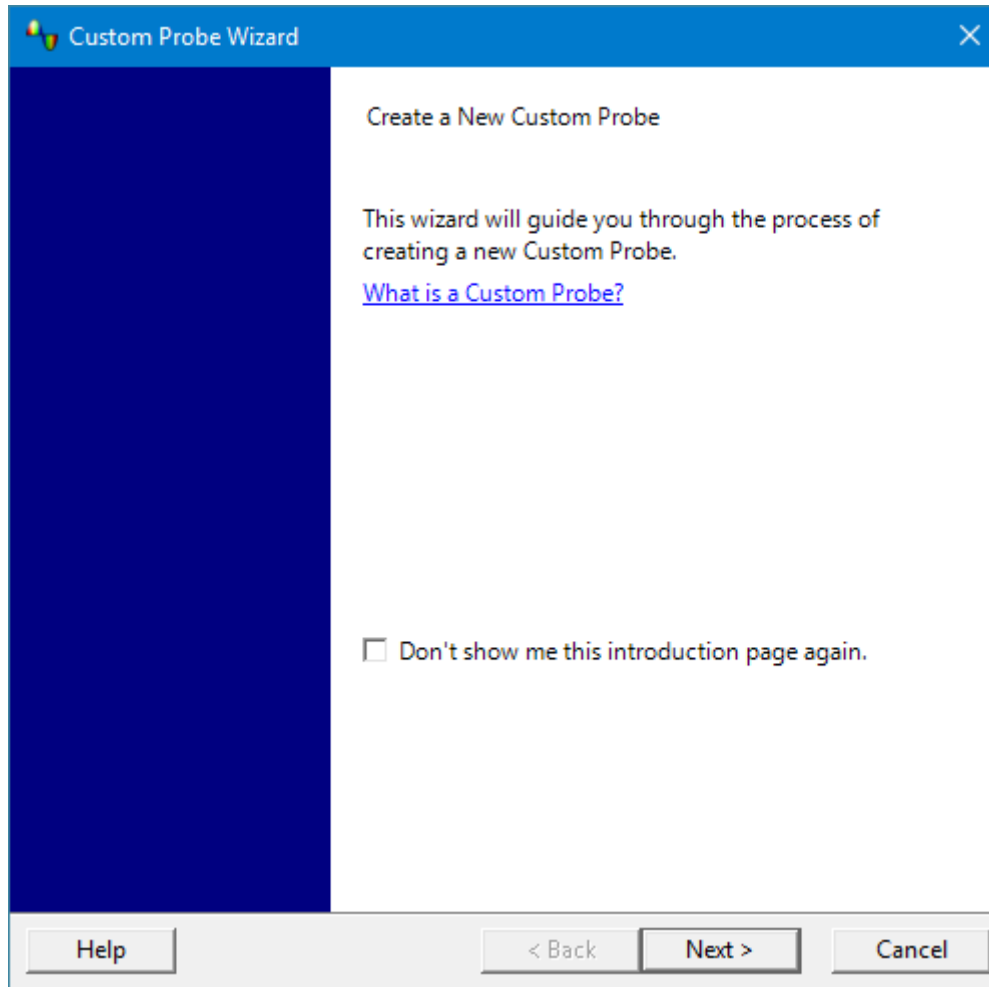


7. [Custom probe identification](#)

### 5.5.1.1.1 Custom Probe wizard introduction dialog

Location: [Custom Probes dialog](#) > **New Probe...**

Purpose: introduces you to the process for creating a new custom probe



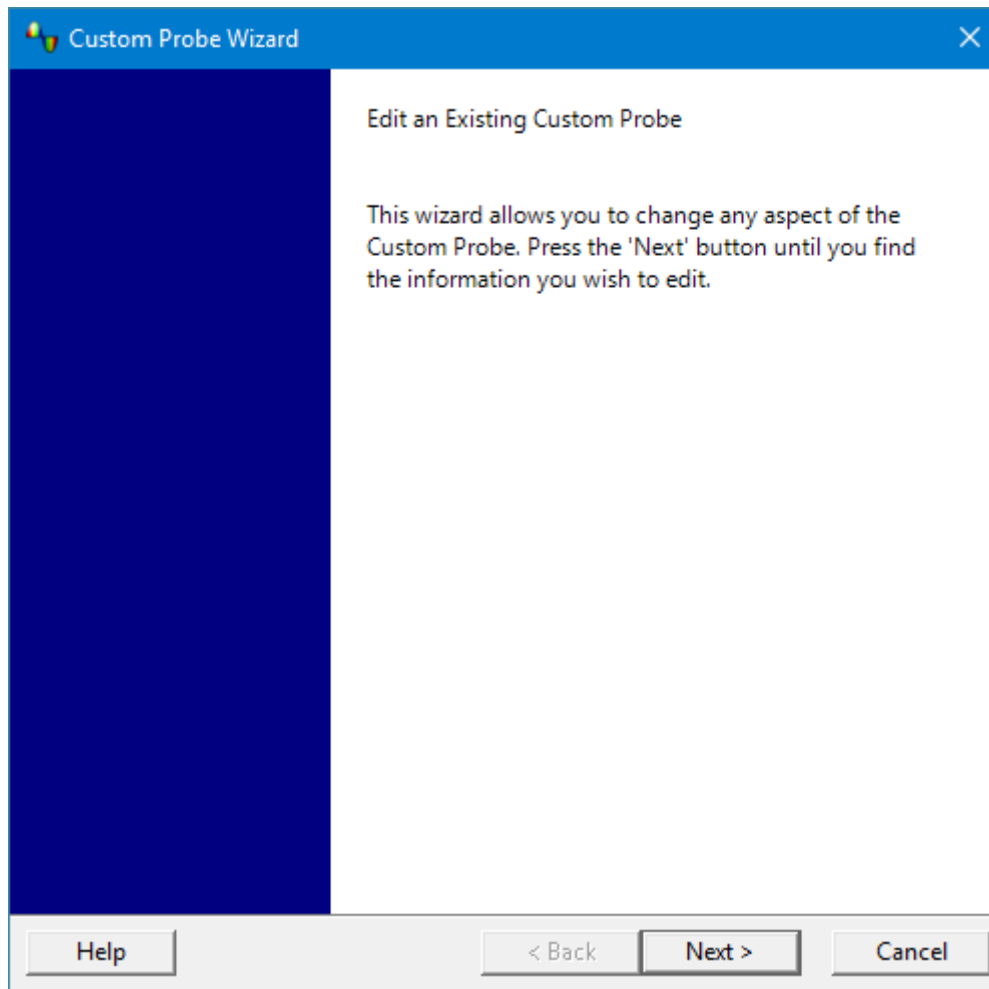
Click **Next** to continue to the [Probe Output Units dialog](#).



### 5.5.1.1.2 Edit Existing Custom Probe dialog

Location: [Custom Probes dialog](#) > **Edit...**

Purpose: introduces you to the process for editing an existing [custom probe](#)

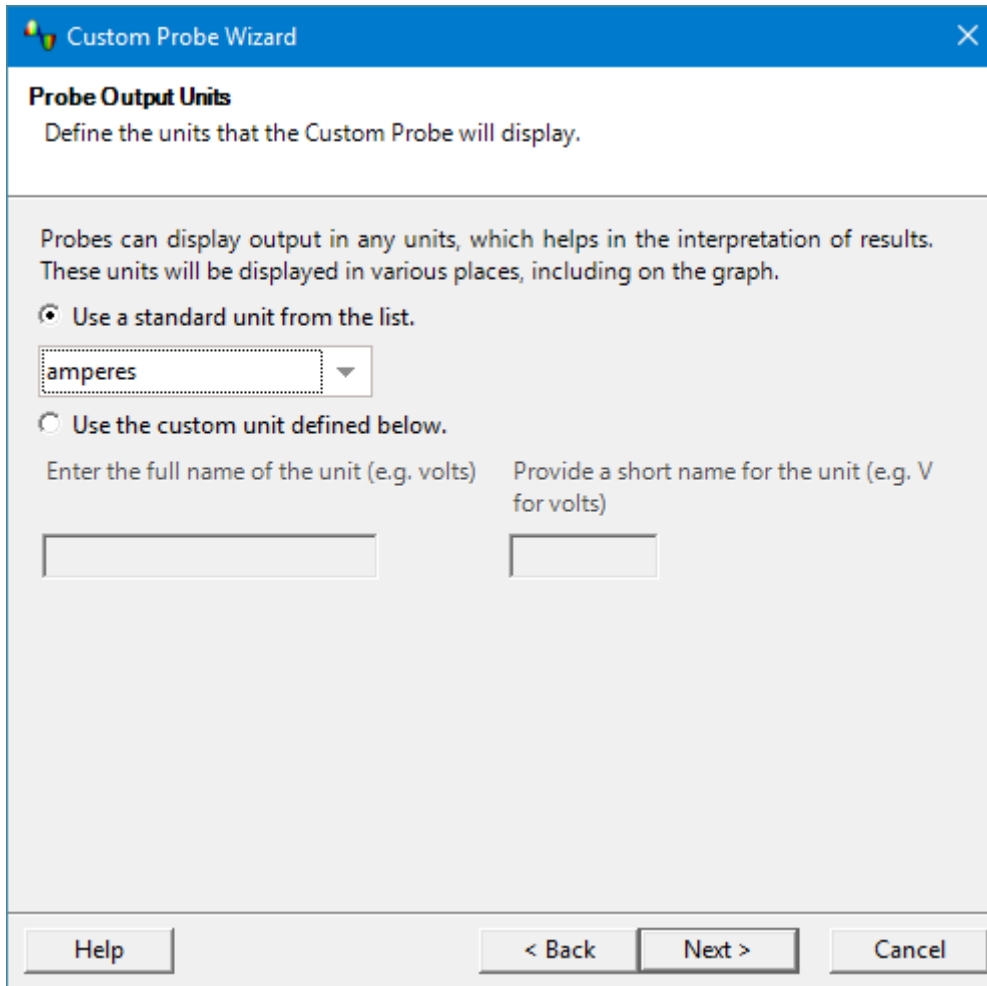


Click **Next** to continue to the [Probe Output Units dialog](#).

### 5.5.1.1.3 Probe Output Units dialog

Location: [Custom Probe wizard](#)

Purpose: allows you to choose the units that PicoScope will use to display the output of your [custom probe](#)



The screenshot shows the 'Custom Probe Wizard' dialog box with the 'Probe Output Units' section active. The title bar reads 'Custom Probe Wizard' with a close button. The main heading is 'Probe Output Units' with the instruction 'Define the units that the Custom Probe will display.' Below this, a text block explains: 'Probes can display output in any units, which helps in the interpretation of results. These units will be displayed in various places, including on the graph.' There are two radio button options: 'Use a standard unit from the list.' (which is selected) and 'Use the custom unit defined below.' Under the first option, a dropdown menu shows 'amperes'. Under the second option, there are two text input fields: 'Enter the full name of the unit (e.g. volts)' and 'Provide a short name for the unit (e.g. V for volts)'. At the bottom, there are four buttons: 'Help', '< Back', 'Next >', and 'Cancel'.

To choose a standard SI unit, click **Use a standard unit from the list** and select one from the list.

To enter a custom unit, click **Use the custom unit defined below** and type the unit name and symbol.

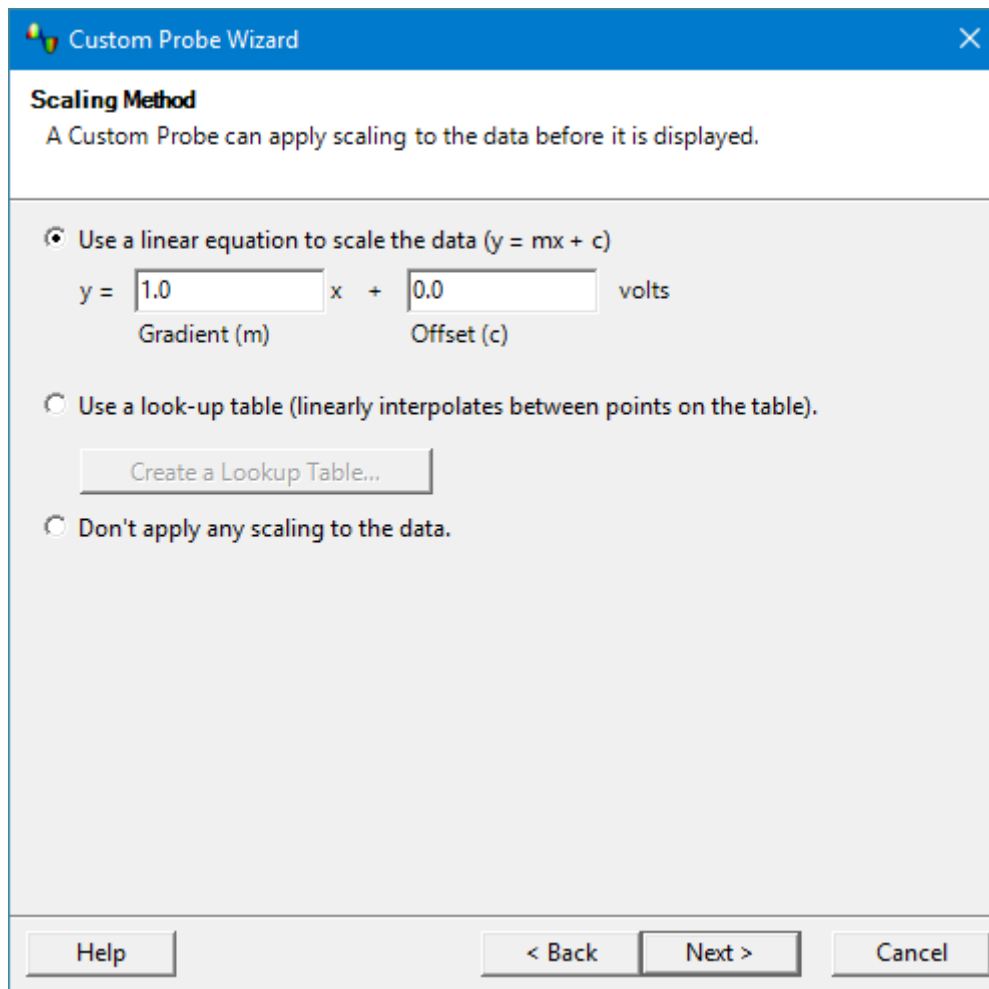
Click **Next** to continue to the [Scaling Method dialog](#).

Click **Back** to return to the [Custom Probe wizard introduction dialog](#) if this is a new probe, or the [Edit Existing Custom Probe dialog](#) if this is an existing probe.

## 5.5.1.1.4 Scaling Method dialog

Location: [Custom Probe wizard](#)

Purpose: allows you to define the characteristic that PicoScope will use to convert the [custom probe's](#) voltage output to a measurement on the display



If you do not require any scaling or offset, select **Don't apply any scaling to the data**.

If the probe requires linear scaling, select **Use a linear equation to scale the data** and enter the gradient (or scale factor)  $m$  and the offset  $c$  in the equation  $y = mx + c$ , where  $y$  is the displayed value and  $x$  is the probe's voltage output.

If you wish to apply a nonlinear function to the probe's output, choose **Use a look-up table**, then click the **Create a Lookup Table...** button to create a new lookup table. This will take you to the [Lookup-table Scaling dialog](#).

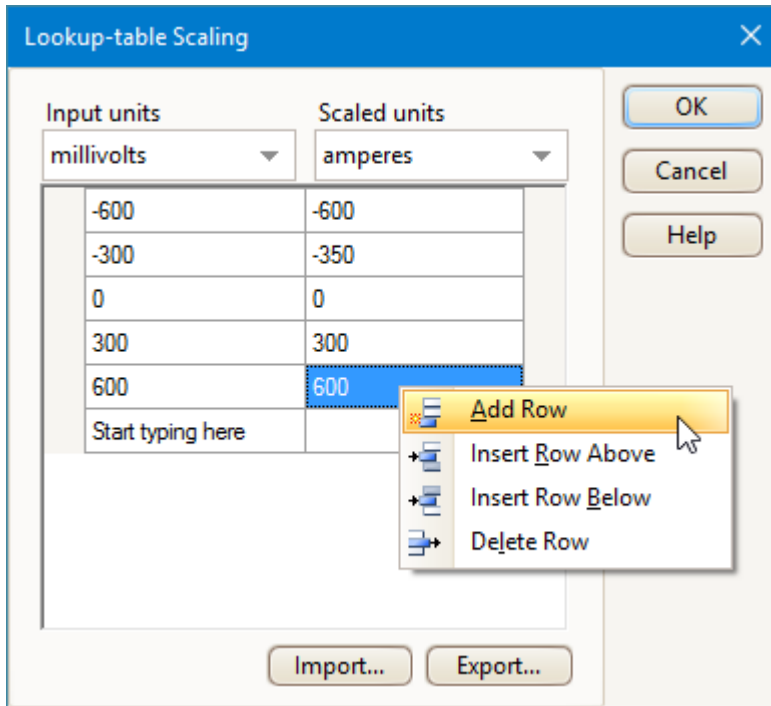
Click **Next** to continue to the [Range Management dialog](#).

Click **Back** to return to the [Probe Output Units dialog](#).

### 5.5.1.1.4.1 Lookup-table Scaling dialog

Location: [Custom Probe wizard](#) > [Scaling Method dialog](#) > **Create a Lookup Table...** or **Edit the Lookup Table...**

Purpose: creates a lookup table to calibrate a [custom probe](#)



#### Editing the Lookup Table

First, select suitable values in the **Input units** and **Scaled units** drop-down boxes. For example, if your probe is a current clamp that outputs one millivolt per ampere over the range  $-600$  to  $+600$  amperes, select **Input units** of millivolts and **Scaled units** of amperes.

Next, enter some data in the scaling table. Click the first empty cell at the top of the table and type  $-600$ , then hit the **Tab** key and type  $-600$ . When you are ready to enter the next pair of values, press the **Tab** key again to start a new row. You can also right-click on the table to obtain a more detailed menu of options, as shown in the picture. In the example above, we have entered a slightly nonlinear response; if the response had been linear then it would have been easier to use the linear option in the [Scaling Method dialog](#).

#### Import.../Export...

Using the **Import...** button, you can fill the lookup table from data in a comma-separated or tab-delimited text file. The **Export...** button allows you to save the lookup table to a new file.

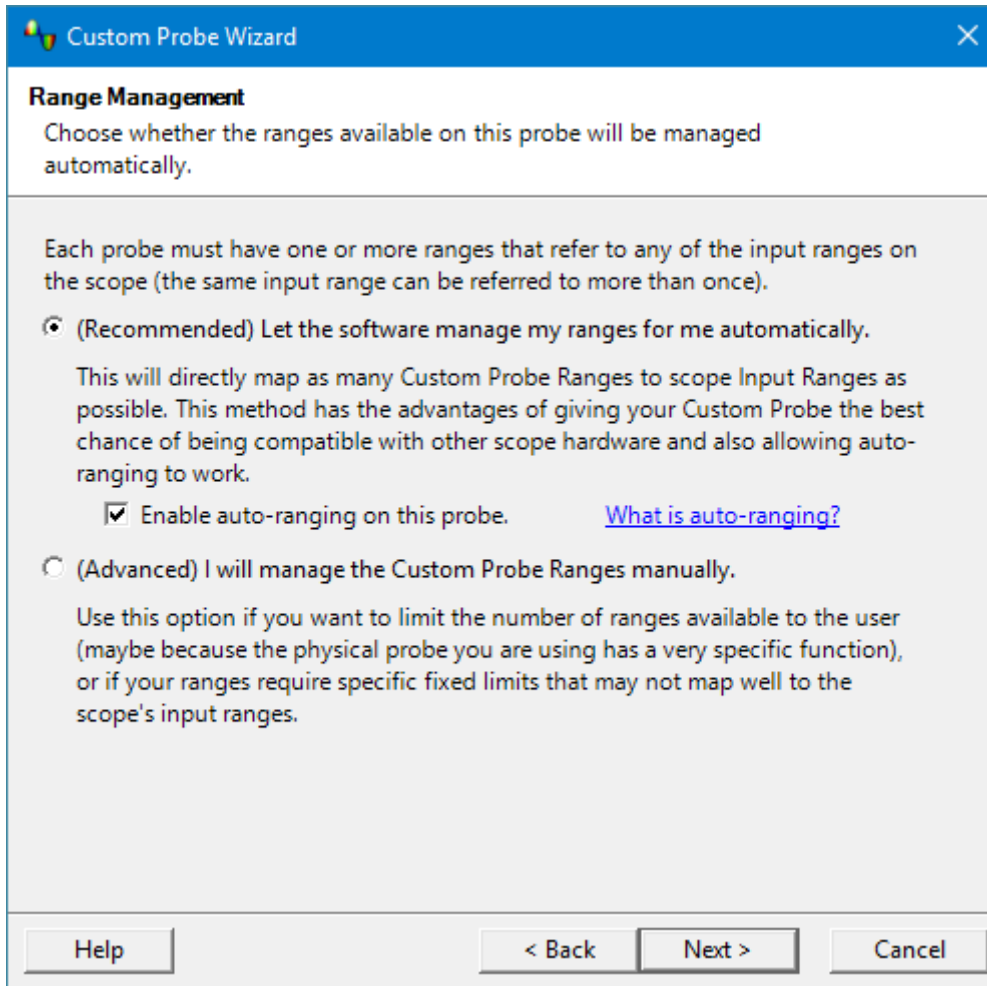
#### Finishing

Clicking **OK** or **Cancel** will return you to the [Scaling Method dialog](#).

### 5.5.1.1.5 Range Management dialog

Location: [Custom Probe wizard](#)

Purpose: allows you to override PicoScope's automatic range-creation feature for custom probes. In most cases, the automatic procedure will be sufficient.



If you select **Let the software manage my ranges for me automatically**, clicking **Next** will take you to the [Custom Probe Identification dialog](#). PicoScope's automatic ranges should be ideal for most applications.

If you select **I will manage the Custom Probe Ranges manually**, clicking **Next** will take you to the [Manual Ranges Setup dialog](#).

Click **Back** to return to the [Scaling Method dialog](#).

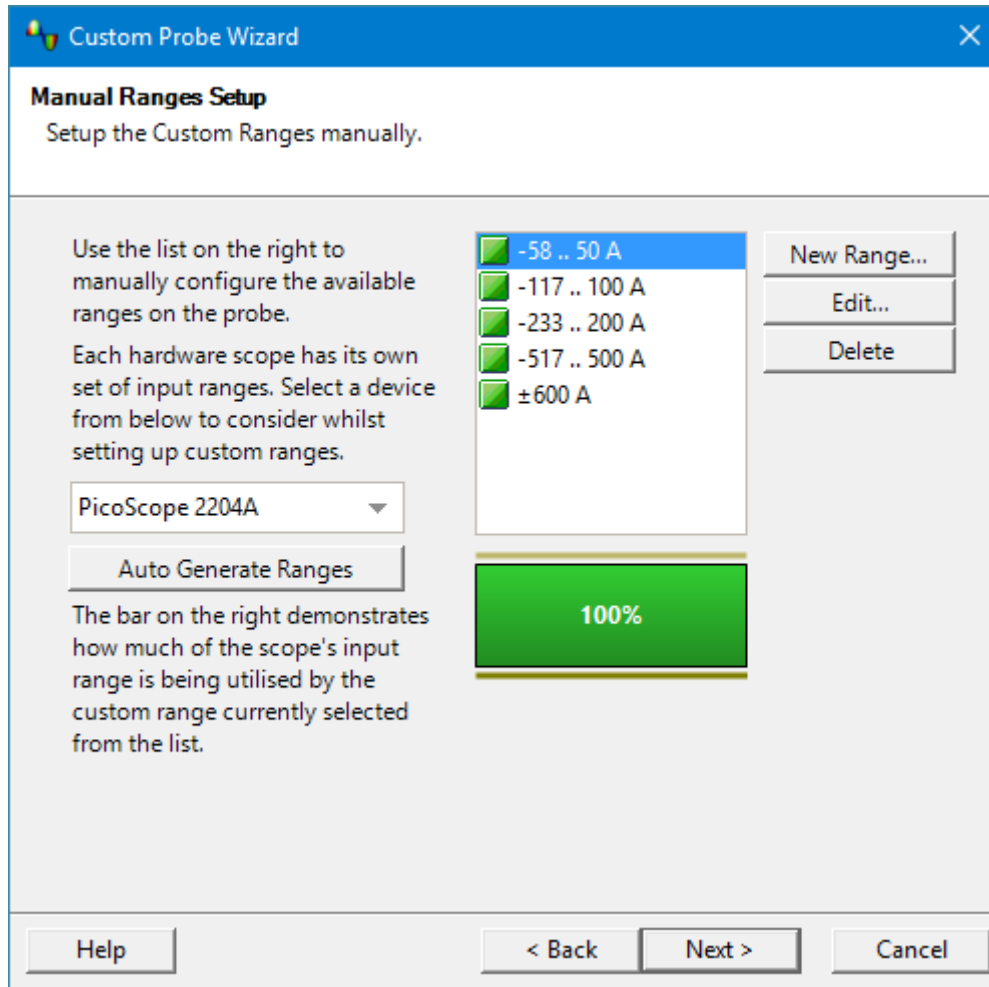
#### What is Auto-ranging?

When the **Auto-ranging** function is selected, PicoScope continually monitors the input signal and adjusts the range when necessary to allow it to display the signal with maximum resolution. This function is available on all standard ranges, and can be used with custom ranges if you select **Let the software manage my ranges for me automatically** in this dialog.

### 5.5.1.1.6 Manual Ranges Setup dialog

Location: [Custom Probe wizard](#) > [Range Management dialog](#) > **Advanced** > **Next**

Purpose: creates ranges manually for your [custom probe](#)



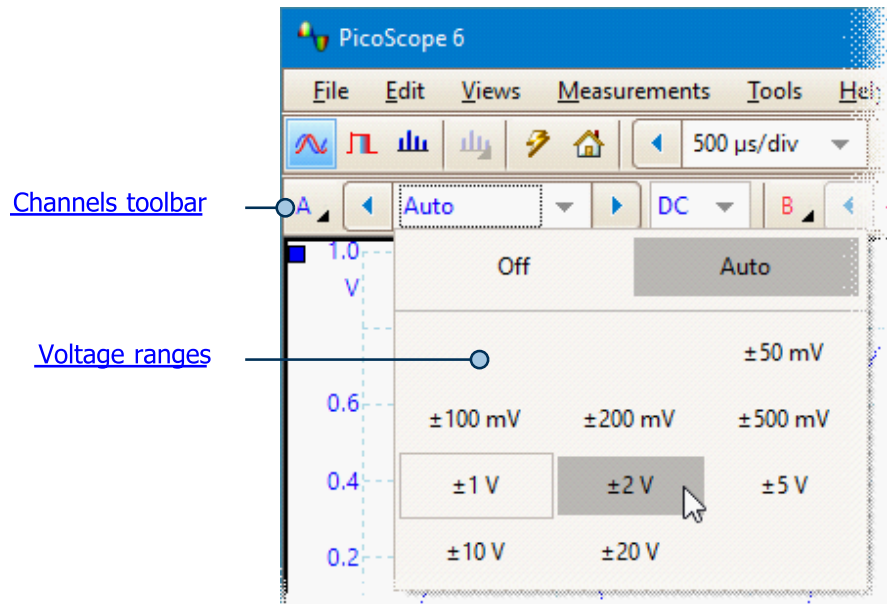
If you wish, you can click **Auto Generate Ranges** and the software will create a number of ranges for the selected device. This will create the same list of ranges that you would have obtained by selecting **Let the software manage my ranges for me automatically** in the previous dialog. When you select a range, a diagram below the list will show its relationship to the scope device's input range – this is explained further under [Edit range dialog](#). You can then edit the ranges by clicking **Edit**, or you can also add a new range by clicking **New Range**. Both of these buttons take you to the [Edit Range dialog](#).

Click **Next** to continue to the [Filter Method dialog](#).

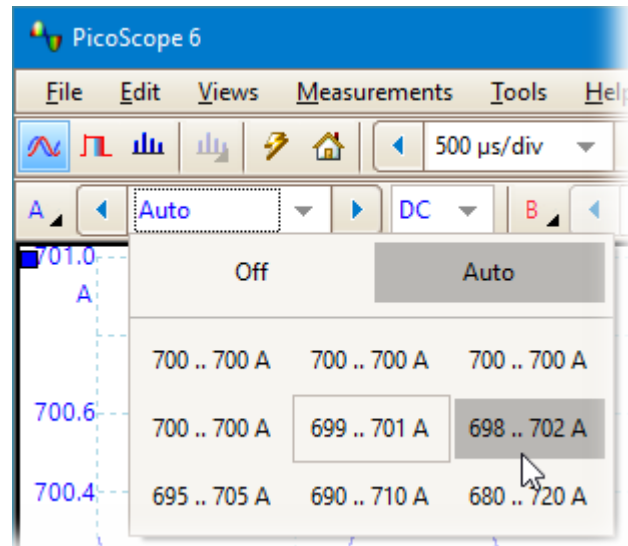
Click **Back** to return to the [Range Management dialog](#).

### How to use a new custom range

Here is the standard input range menu for the built-in x1 probe:



When you create a custom input range, it appears in the drop-down list of ranges in the [channels toolbar](#), like this:



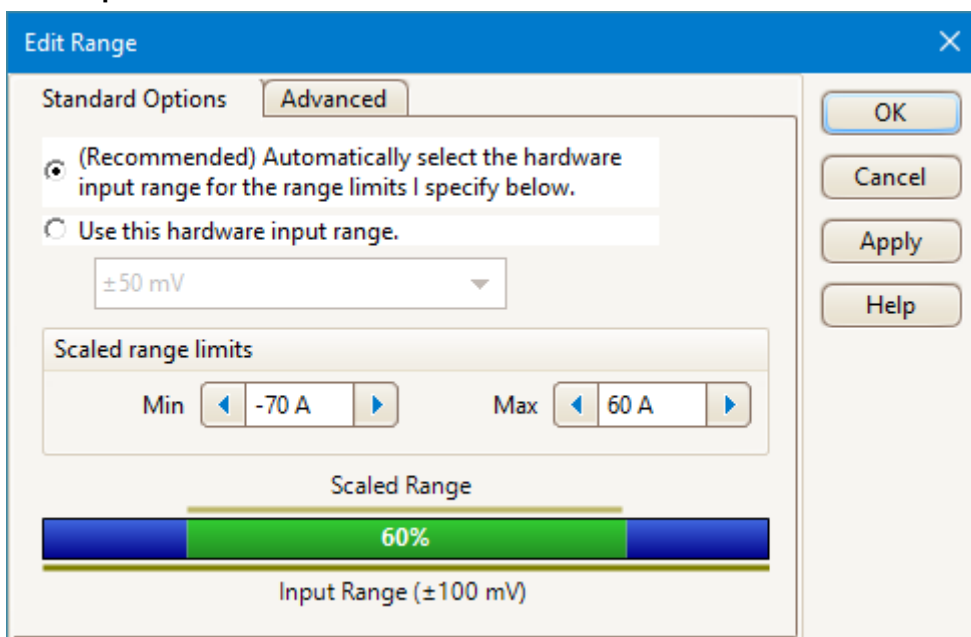
## 5.5.1.1.6.1 Edit Range dialog

Location: [Custom Probe wizard](#) > [Range Management dialog](#) > [Manual Ranges Setup dialog](#) > [Edit Range](#) or [New Range](#)

Purpose: editing a manual range for a [custom probe](#)

An **input range** is the signal range, usually in volts, on the input channel of the [scope device](#). The **scaled range** is the range that will appear on the vertical axis of the scope display when the probe is selected. Your scaled range should match the input range as closely as possible, to make the most of the scope's resolution.

The scaling that you chose on the [Scaling Method](#) page defines the relationship between the input range and the scaled range. This dialog enables you to set up ranges to display the scaled data on the scope view.

**Standard Options tab****Automatically select the hardware input range for the range limits I specify below**

If you leave this radio button pressed, PicoScope will automatically determine the best hardware input range for the device as you change the **Scaled range limits**. This is the best mode to use for almost all ranges. You should set the **Scaled range limits** to the maximum and minimum values you wish to see on the vertical axis of the scope display.

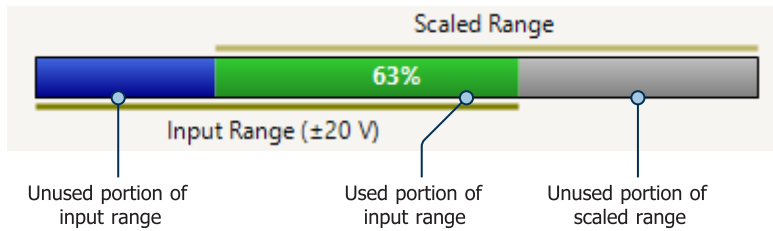
**Use this hardware input range**

If you press this radio button and select a hardware input range from the drop-down box, PicoScope will then use that hardware input range whatever scaled range limits you choose. Set the upper and lower scaled range limits to the limits you wish to appear at the top and bottom of the vertical axis in PicoScope's [scope view](#).

**The range utilization bar**

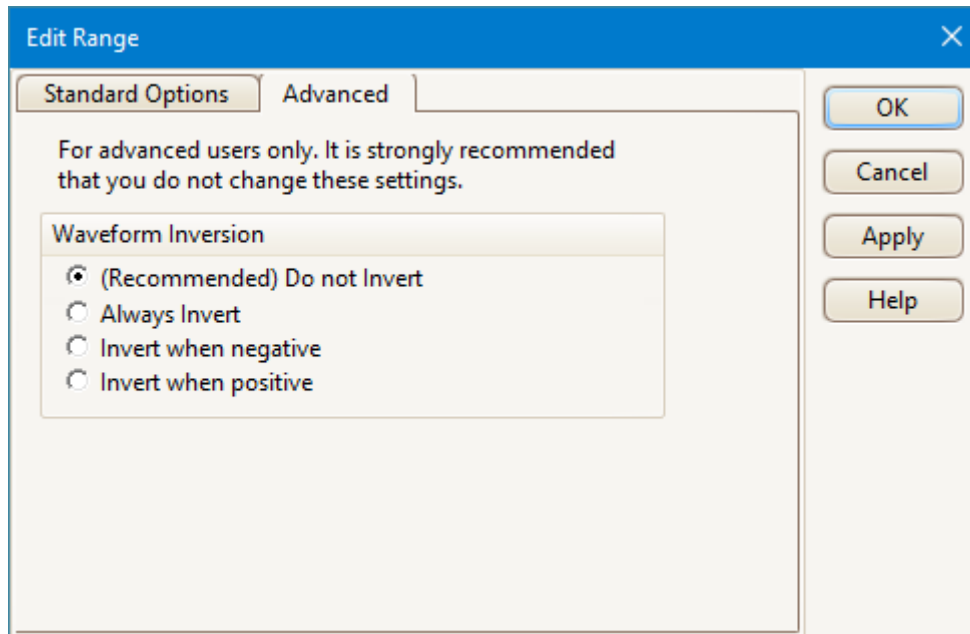
This diagram at the bottom of the dialog shows how well the input range of the device is matched to the scaled range.





- **Green** - The section of the input range that is used by the scaled range. This should be as large as possible, to maximize the use of the scope device's resolution.
- **Blue** - Areas of the input range that are not being used. These indicate wasted resolution.
- **Grey** - Parts of the scaled range that are not covered by the input range. These will result in wasted space on the graph. The range utilization bar may not represent these areas accurately when non-linear scaling is being used, so you should always test the scaled range limits on the scope view.

### Advanced tab



The **Advanced** tab allows you to configure advanced options for [custom probes](#). However, these options are for factory use and we recommend that you do not change them.

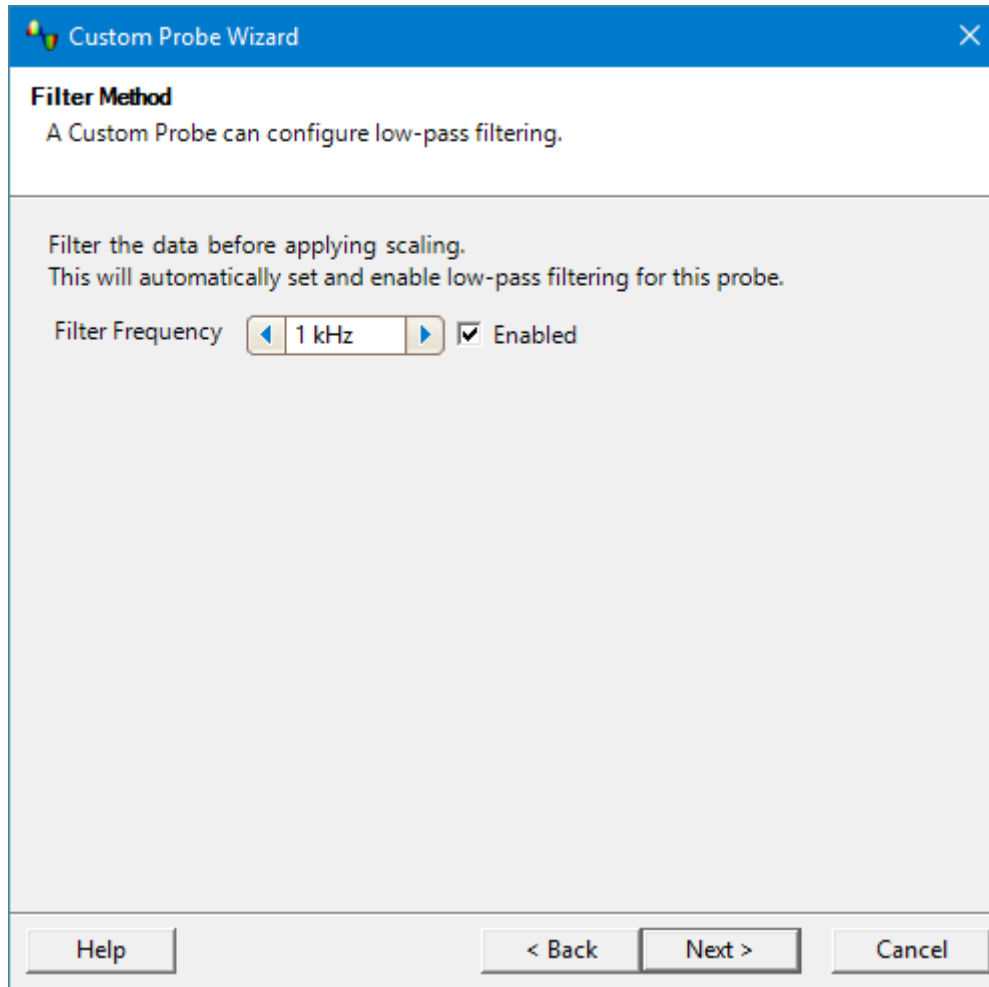
### Finishing

Clicking **OK** or **Cancel** will return you to the [Manual Ranges Setup dialog](#).

### 5.5.1.1.7 Filter Method dialog

Location: [Custom Probe wizard](#)

Purpose: sets up lowpass filtering for this custom probe



This dialog has the same effect as manually enabling the [Lowpass Filtering](#) option in the [Channel Options dialog](#). Filtering will only occur if the attached scope device supports filtering.

Click **Back** to go to the [Manual Range Setup dialog](#).

Click **Next** to go to the [Custom Probe Identification dialog](#).

## 5.5.1.1.8 Custom Probe Identification dialog

Location: [Custom Probe wizard](#)

Purpose: entering text to identify the [custom probe](#)

Custom Probe Wizard

**Custom Probe Identification**  
Provide descriptive details so your new probe can be identified later.

Enter a name for the probe

Acme current clamp

Write a short description for the probe, so it can be easily identified (optional).

600 A current clamp, 1 mV/A

Help < Back Finish Cancel

Click **Back** to return to the [Filter Method dialog](#).

The **probe name** will appear in the probe list.

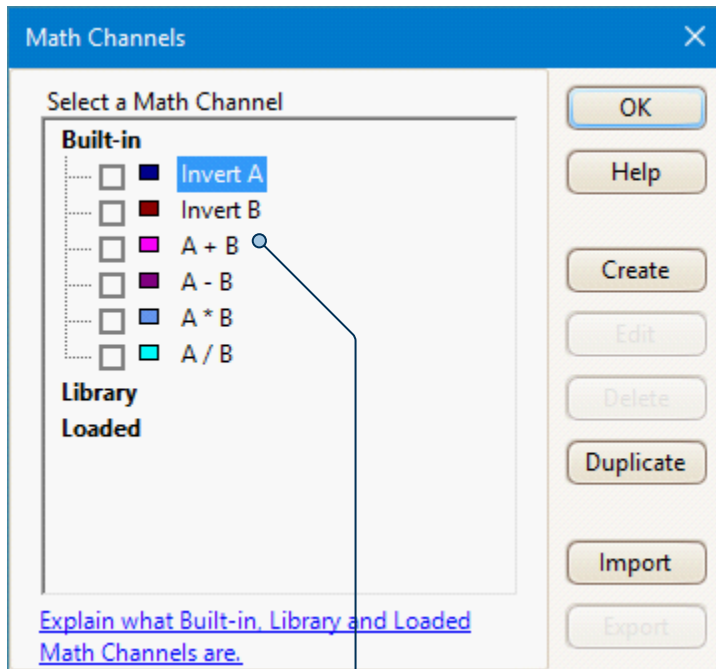
The optional **description** is not used in the current version of the software.

Click **Finish** to accept your custom probe settings and return to the [Custom Probes dialog](#).

## 5.5.2 Math Channels dialog

Location: [Tools > Math Channels](#)

Purpose: [creating, editing](#) and controlling [math channels](#), which are virtual channels generated by mathematical functions of input channels



Math Channel list

### Math Channel list

The main area of the **Math Channels dialog** is the **Math Channel list**, which shows all the built-in, library and loaded [math channels](#). To choose whether or not a channel appears in the main [PicoScope window](#), click the appropriate check box and then **OK**. You can have up to eight channels in any view, including input channels and math channels. If you enable a ninth channel, PicoScope opens a new view.

**Built-in:** these math channels are defined by PicoScope and cannot be changed.

**Library:** these are the math channels that you define using the **Create**, **Duplicate** or **Edit** buttons, or load with the **Import** button.

**Loaded:** these are the math channels present in any PicoScope settings or data files that you have loaded.

### Create

Opens the [Math Channel wizard](#), which guides you through the process of creating or editing a math channel. The new channel will appear under **Library** in the **Math Channel list**.

### Edit

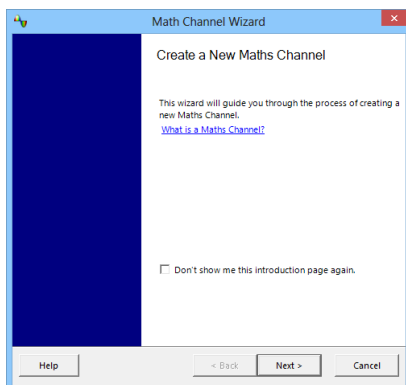
Opens the [Math Channel wizard](#) to allow you to edit the selected math channel. You must first select a channel in the **Library** section of the **Math Channel list**. If the channel you want to edit is in the **Built-in** or **Loaded** section, first copy it to the **Library** section by clicking **Duplicate**, then select it and click **Edit**.

<b>Delete</b>	Permanently deletes the selected math channel. Only math channels in the <b>Library</b> section can be deleted.
<b>Duplicate</b>	Creates a copy of the selected math channel. The copy is placed in the <b>Library</b> section, where you can edit it by clicking <b>Edit</b> .
<b>Import</b>	Opens a .psmaths math channel file and places the math channels it contains in the <b>Library</b> section.
<b>Export</b>	Saves the highlighted math channel from the <b>Library</b> section to a new .psmaths file.

### 5.5.2.1 Math Channel wizard

Location: [Channel Setup toolbar](#) > **Math Channels** button

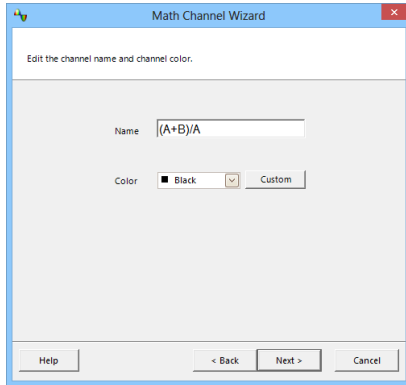
Purpose: creating, editing and controlling math channels, which are virtual channels generated by mathematical functions of input channels



#### 1. [Introduction](#)



#### 2. [Equation](#)



Math Channel Wizard

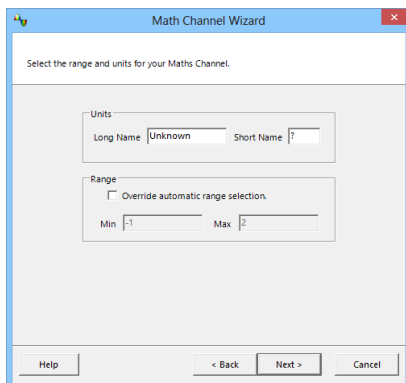
Edit the channel name and channel color.

Name

Color  Black  Custom

Help < Back Next > Cancel

3. [Channel name](#)



Math Channel Wizard

Select the range and units for your Maths Channel.

Units

Long Name  Short Name 

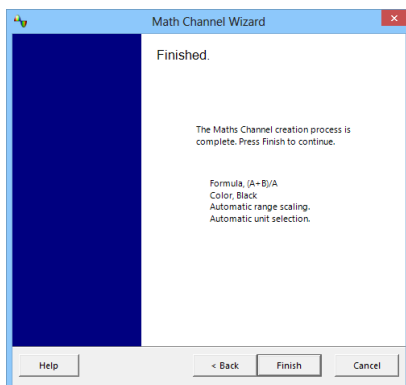
Range

Override automatic range selection.

Min  Max

Help < Back Next > Cancel

4. [Units and range](#)



Math Channel Wizard

Finished.

The Maths Channel creation process is complete. Press Finish to continue.

Formula: (A-B)/A  
Color: Black  
Automatic range scaling.  
Automatic unit selection.

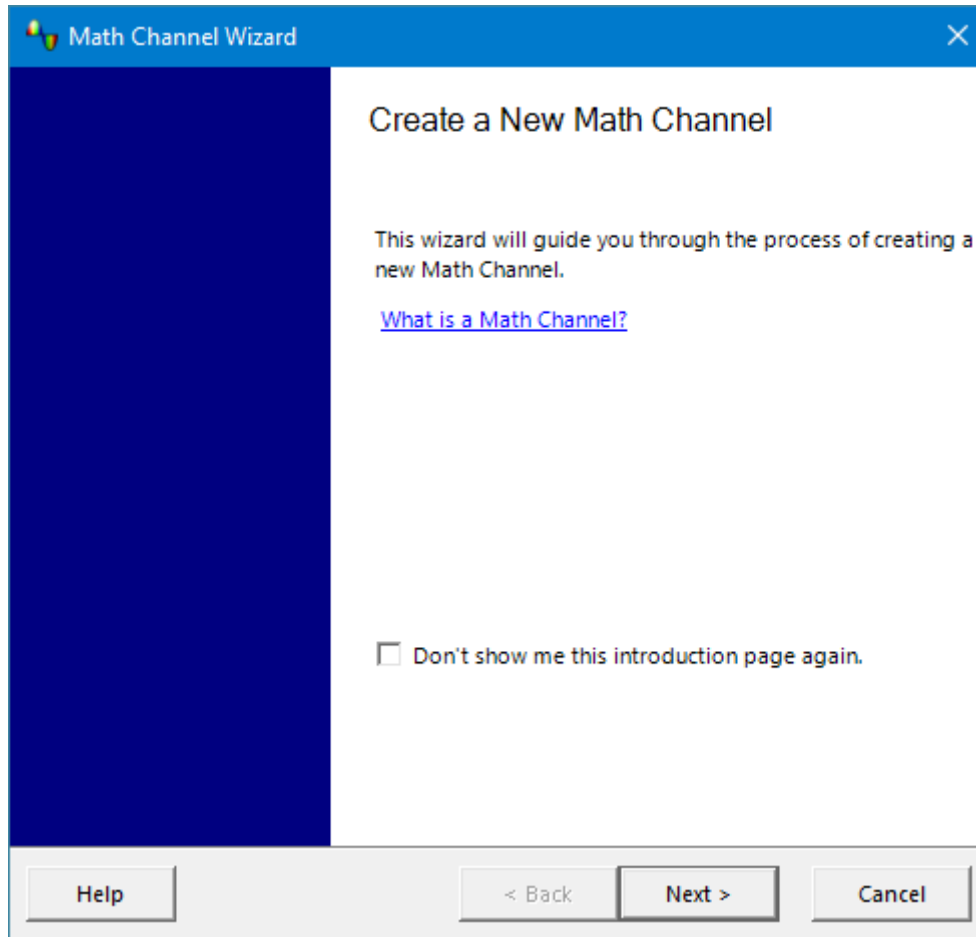
Help < Back Finish Cancel

5. [Finished](#)

## 5.5.2.1.1 Math Channel wizard introduction dialog

Location: [Math Channels dialog](#) > Create

Purpose: introduces you to the process for creating a new math channel




Click **Next** to continue to the [Math Channel wizard Equation dialog](#).

### 5.5.2.1.2 Math Channel wizard Equation dialog

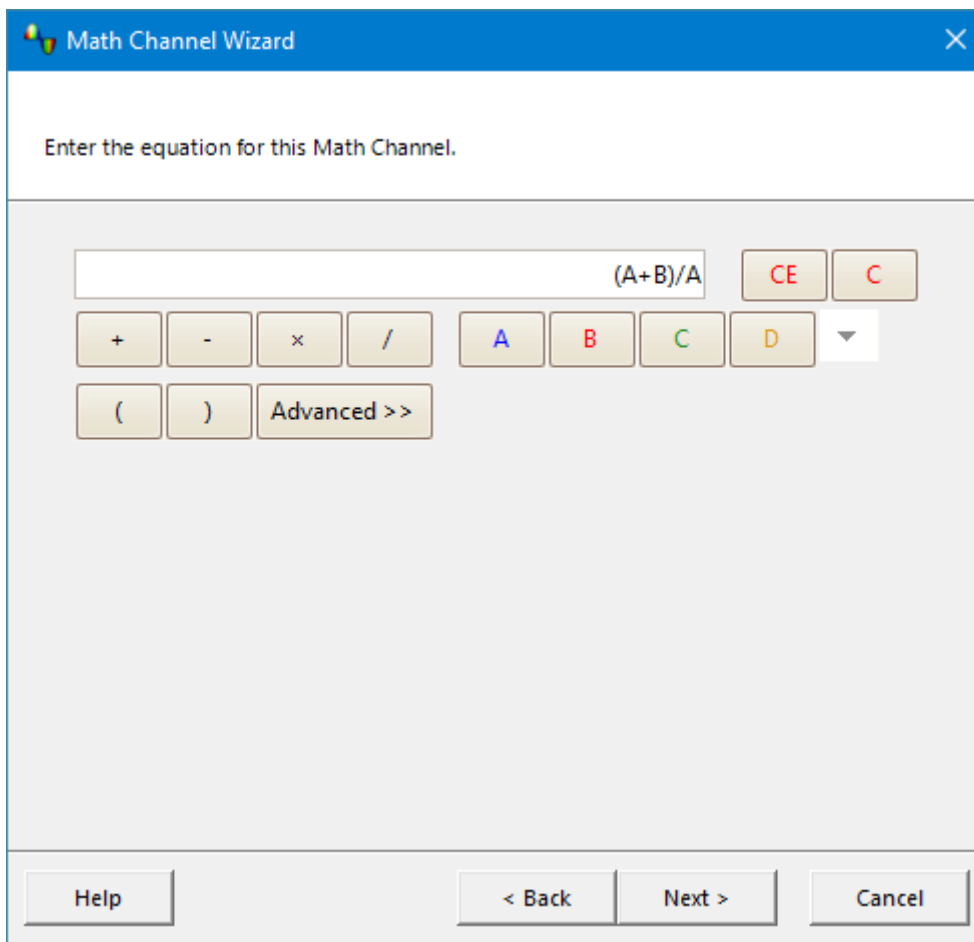
Location: [Math Channel wizard](#)

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. A red error indicator  will appear to the right of the equation box if the equation contains a syntax error.




When you are happy with your equation, click **Next** to continue to the [Math Channel wizard Name dialog](#).

#### Basic view











Math Channel wizard Equation dialog, basic view

#### Basic buttons

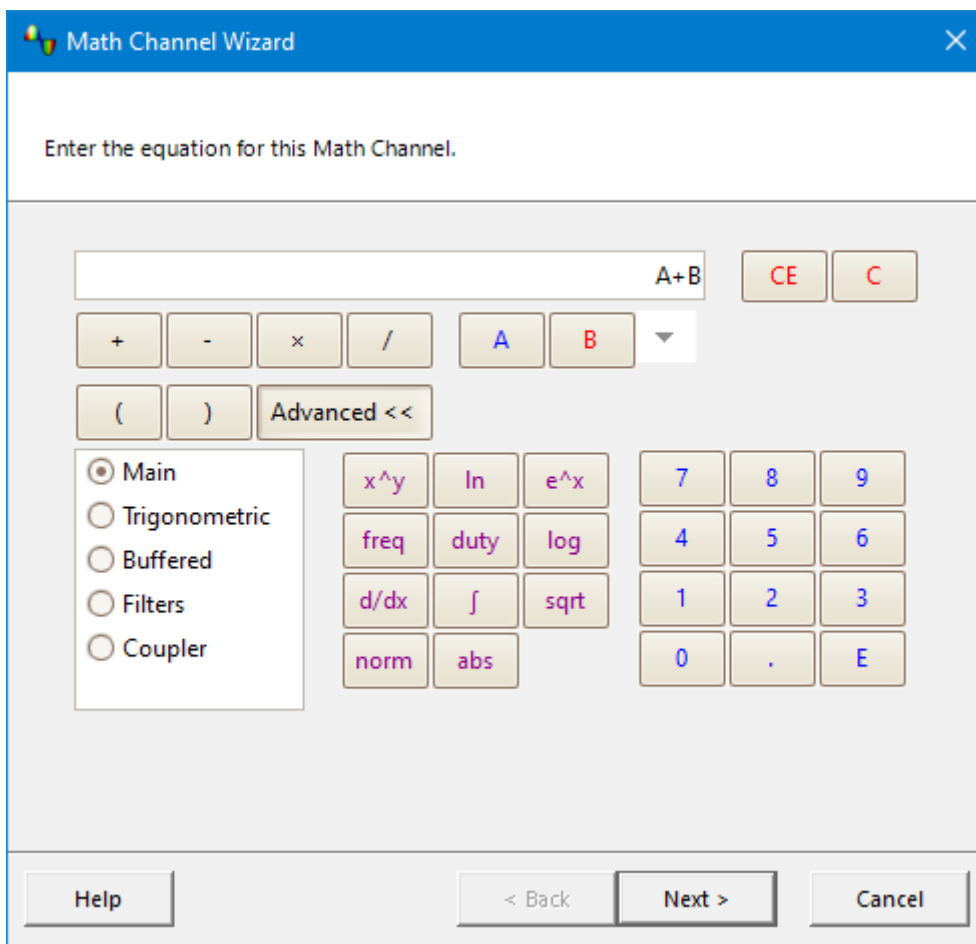
Button	Equation	Description
		<b>Clear Equation.</b> Clears the entire contents of the equation box.
		<b>Clear.</b> Clears the single character to the left of the cursor.
	+	<b>Add</b>



	-	<b>Subtract</b> (or negate)
	*	<b>Multiply</b>
	/	<b>Divide</b>
 ... 	A...D	<b>Input Channels.</b> The selection available depends on the number of enabled channels on the scope device.
	{...}, T	<b>Other Operands.</b> Shows a drop-down list of available inputs to equations, including <a href="#">reference waveforms</a> and <b>Time</b> .
 	(...)	<b>Brackets.</b> Expressions within brackets will be evaluated before the expressions to either side.





**Advanced view**

Clicking the **Advanced** button reveals more function buttons. Initially these are the **Main** group of buttons, as shown below.






Math Channel wizard Equation dialog, advanced view, showing Main group of buttons

**Advanced buttons (numeric keypad)**









Button	Equation	Description
 ... 	0..9	<b>0 to 9.</b> The decimal digits.
	.	<b>Decimal point</b>
	E	<b>Exponent.</b> $aEb$ means $a \times 10^b$ .

**Advanced buttons (Main group)**

Button	Equation	Description
	$x^y$	<b>Power.</b> Raise $x$ to the power of $y$ .
	$\ln()$	<b>Natural logarithm</b>
	$e^x$	<b>Natural exponent.</b> Raise $e$ , the base of the natural logarithm, to the power of $x$ .
	$\text{freq}()$	<b>Frequency.</b> Calculated in hertz.
	$\text{duty}()$	<b>Duty.</b> Calculate the duty cycle as a percentage.
	$\log()$	<b>Logarithm.</b> Base-10 logarithm.
	$d/dx$	<b>Derivative.</b> 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 <a href="#">digital lowpass filtering</a> to all channels used as inputs to this function.
	$\int$	<b>Integral.</b> Along the $x$ -axis.
	$\text{sqrt}()$	<b>Square root</b>
	$\text{norm}()$	<b>Normalize.</b> 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.
	$\text{abs}()$	<b>Absolute value</b>
	$\text{crank}()$	<b>Crank speed (PicoScope Automotive only).</b> This function requires two arguments: the channel, and the number of teeth on the crank sensor (including the spaces for missing teeth). PicoScope then calculates and plots the crank speed in RPM.





Click **Trigonometric**, **Buffered**, **Filters** or **Coupler** to reveal alternative groups of buttons:

#### Advanced buttons (Trigonometric group)

Button	Equation	Description
	pi	<b>Pi.</b> The ratio of a circle's circumference to its diameter.
		<b>Inverse.</b> Modifies the <b>sin</b> , <b>cos</b> and <b>tan</b> buttons to the inverse trigonometric functions <b>asin</b> , <b>acos</b> and <b>atan</b> .
	sin()	<b>Sine.</b> The operand is in radians.
	cos()	<b>Cosine.</b> The operand is in radians.
	tan()	<b>Tangent.</b> The operand is in radians.
	sinh()	<b>Hyperbolic sine.</b>
	cosh()	<b>Hyperbolic cosine.</b>
	tanh()	<b>Hyperbolic tangent.</b>

#### Advanced buttons (Buffered group)

When the scope is running, these functions operate continuously on all waveforms since the scope started capturing. If a math channel containing these functions is enabled when the scope is stopped, it operates on the contents of the waveform buffer instead.


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




#### Advanced buttons (Filters group)

Parameters:

**i** is the input channel or other operand (see under **Basic buttons** above)

**f** (or **f<sub>1</sub>** and **f<sub>2</sub>**) are the -3 dB cutoff frequencies of the filter, in hertz

Button	Equation	Description
	HighPass( <i>i</i> , <i>f</i> )	<b>Highpass filter.</b> Attenuates low frequencies.

Button	Equation	Description
	$\text{LowPass}(i,f)$	<b>Lowpass filter.</b> Attenuates high frequencies.
	$\text{BandPass}(i,f_1,f_2)$	<b>Bandpass filter.</b> Attenuates high and low frequencies outside the specified range.
	$\text{BandStop}(i,f_1,f_2)$	<b>Bandstop filter.</b> Attenuates mid-band frequencies inside the specified range.

These are digital filters with a finite number of taps and therefore cannot attenuate down to DC. They have a minimum cutoff frequency of 1/64 000 of the scope's sampling rate. You can find the current sampling rate displayed in the [Properties sheet](#).

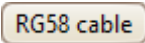
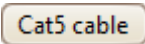
### Advanced buttons (Coupler group)

Parameters:

$s_1$  is the master channel

$s_2$  is the slave channel

$d$  is the distance between the master and slave probing points, in metres

Button	Equation	Description
	$\text{Coupler}(s_1,s_2,RG58,d)$	<b>RG58 coupler.</b> For coaxial cables.
	$\text{Coupler}(s_1,s_2,Cat5,d)$	<b>Cat5 coupler.</b> For twisted-pair cables.

These are directional coupler functions for extracting [BroadR-Reach](#) signals from a bidirectional bus. One differential probe (connected to channel  $s_1$ ) is placed on the cable near the master device, and another (connected to channel  $s_2$ ) on the cable near the slave device. The output of the coupler is then used as the input to the BroadR-Reach protocol decoder (see [Serial Decoding](#)).

### Additional functions

There are a few operators that can be entered only by using the equation box:

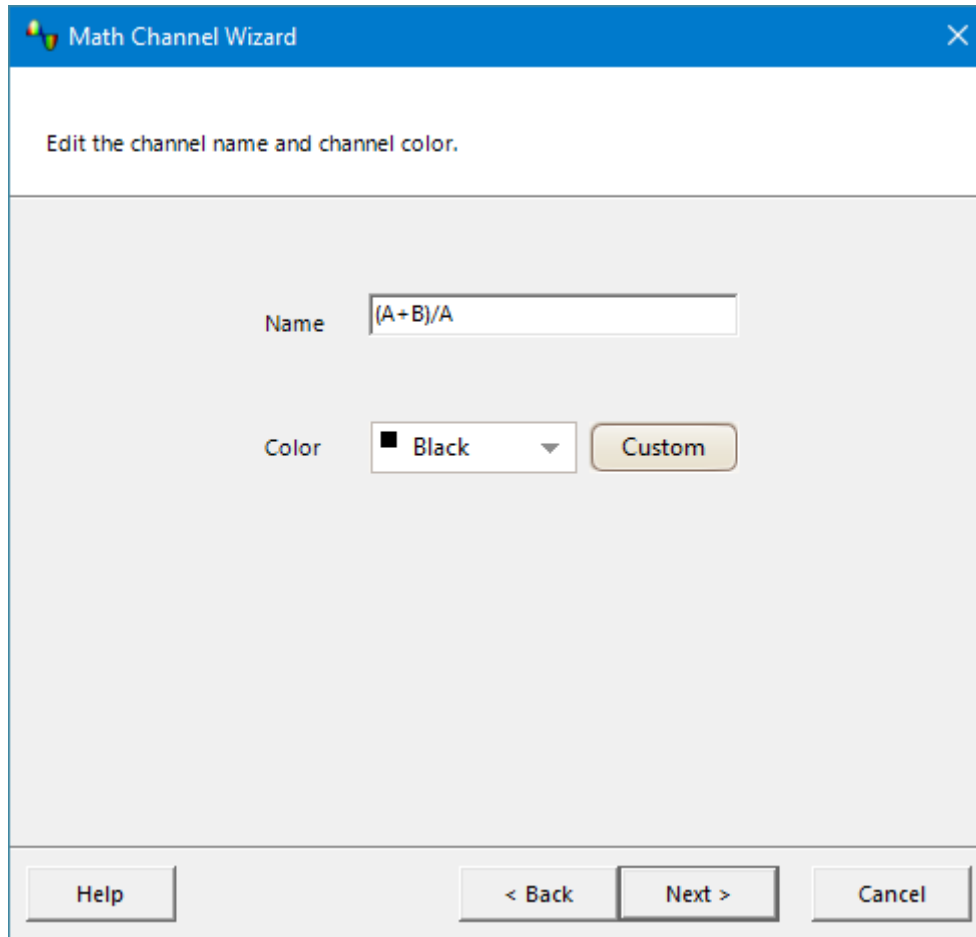
**Signum function.** The **sign()** operator returns the sign of its input. The result is +1 when the input is positive, -1 when the input is negative, and 0 when the input is 0.

**Advance/Delay.** To deskew a signal, select the channel and add the deskew time in seconds, in square brackets. For example, to advance Channel A by 1 millisecond, enter **A[0.001]**, and to delay Channel A by 1 millisecond, enter **A[-0.001]**. Using time offsets to adjust one channel relative to another can compensate for differences in cable lengths or skew in trace lengths.

## 5.5.2.1.3 Math Channel wizard Name dialog

Location: [Math Channel wizard](#)

Purpose: allows you to enter or edit the name and color of a [math channel](#)



PicoScope initially sets the name to the text of the equation, but you can edit it to anything you like. The name will appear in the channel list in the [Math Channels dialog](#). You can set the color of the trace to one of the standard colors in the drop-down-list, or click **Custom** to choose any possible color allowed by Windows.

Click **Next** to continue to the [Math Channel wizard Units and Range dialog](#).

#### 5.5.2.1.4 Math Channel wizard Units and Range dialog

Location: [Math Channel wizard](#)

Purpose: allows you to specify the measurement units and the range of values to display for a [math channel](#)

Math Channel Wizard

Select the range and units for your Math Channel.

Units

Long Name  Short Name 

Range

Override automatic range selection.

Min  Max

Help < Back Next > Cancel

#### Units

**Long Name:** This is for your reference only.

**Short Name:** This will be displayed on the vertical axis in [scope](#) and [spectrum](#) views, in the [ruler legend](#) and in the [measurements table](#).

#### Range

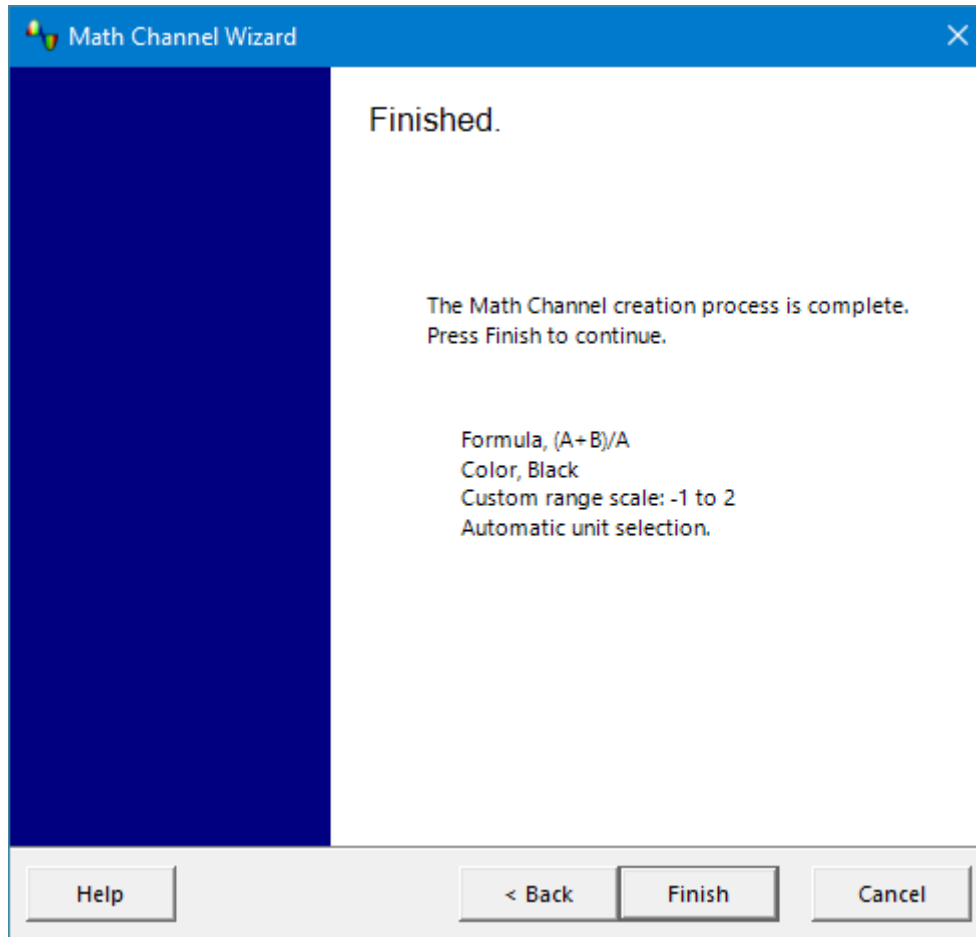
If you leave the **Override automatic range selection** check box empty, PicoScope will choose the most appropriate range for the vertical axis. If you prefer to set your own minimum and maximum values for the axis, check the box and enter them in the **Min** and **Max** boxes.

Click **Next** to continue to the [Math Channel wizard Finished dialog](#).

## 5.5.2.1.5 Math Channel wizard Finished dialog

Location: [Math Channel wizard](#)

Purpose: shows you the settings for the [math channel](#) that you have just created or edited



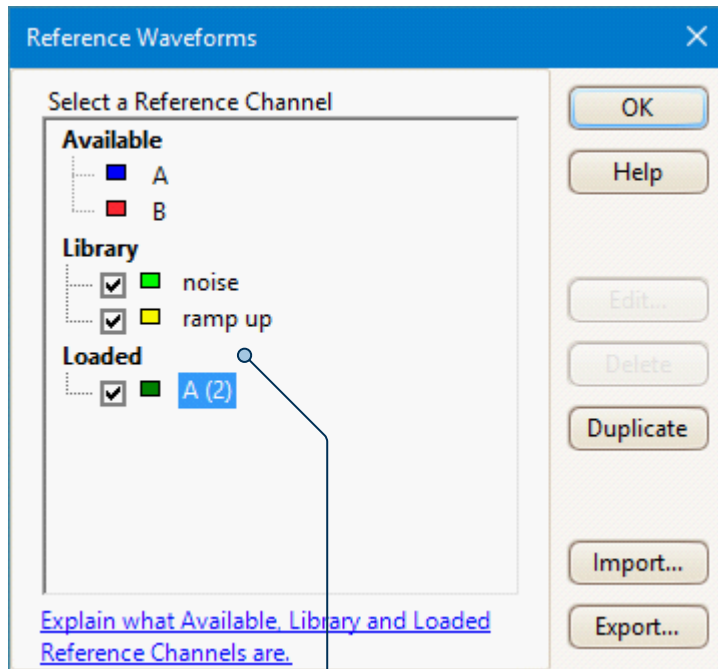
Click **Back** to return to previous dialogs in the [Math Channel wizard](#) if you wish to change any of the settings.

Click **Finish** to accept the settings shown and return to the [Math Channels dialog](#). If you want the new or edited channel to appear on the scope or spectrum display, remember to check the appropriate box in the channel list. You can change them later by returning to the Math Channels dialog.

### 5.5.3 Reference Waveforms dialog

Location: [Tools](#) > [Reference Waveforms](#)

Purpose: enables you to create, [edit](#) and control [reference waveforms](#), which are stored copies of input channels



Reference Waveforms list

**Reference Waveforms list** The main area of the **Reference Waveforms dialog** is the **Reference Waveforms list**, which shows all the available input channels and the library and loaded [reference waveforms](#). To choose whether or not a waveform appears in the main [PicoScope window](#), click the appropriate check box and then **OK**. You can have up to eight channels in any view, including input channels, math channels and reference waveforms. If you try to enable a ninth channel, PicoScope will open another view.

**Available:** these input channels are suitable as sources for reference waveforms

**Library:** these are the reference waveforms that you have defined using the **Duplicate** button or loaded with the **Import** button

**Loaded:** these are the reference waveforms present in any PicoScope settings or data files that you have loaded

**Edit** Opens the [Edit Reference Waveform dialog](#) to allow you to edit the selected reference waveform. You must first select a waveform in the **Library** section of the **Reference Waveforms list**. If the waveform you want to edit is in the **Loaded** section, first copy it to the **Library** section by clicking **Duplicate**, then select it and click **Edit**.

**Delete** Permanently deletes the selected reference waveform. Only reference waveforms in the **Library** section can be deleted.

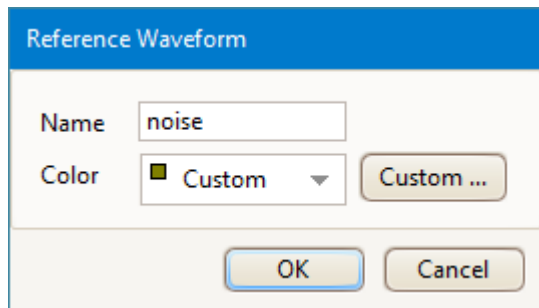


- Duplicate** Creates a copy of the selected input channel or reference waveform. The copy is placed in the **Library** section, from where you can edit it by clicking **Edit**. A quicker way to do the same thing is to right-click on the view, select **Reference Waveforms** and then click the channel that you wish to copy.
- Import** Opens a .psreference reference waveform file and places the waveforms it contains in the **Library** section.
- Export** Saves the highlighted reference waveform from the **Library** or **Loaded** sections to a new .psreference or MATLAB 4 .mat file.

### 5.5.3.1 Edit Reference Waveform dialog

Location: [Reference Waveforms dialog](#) > **Edit**

Purpose: allows you to edit the name and color of a [reference waveform](#)

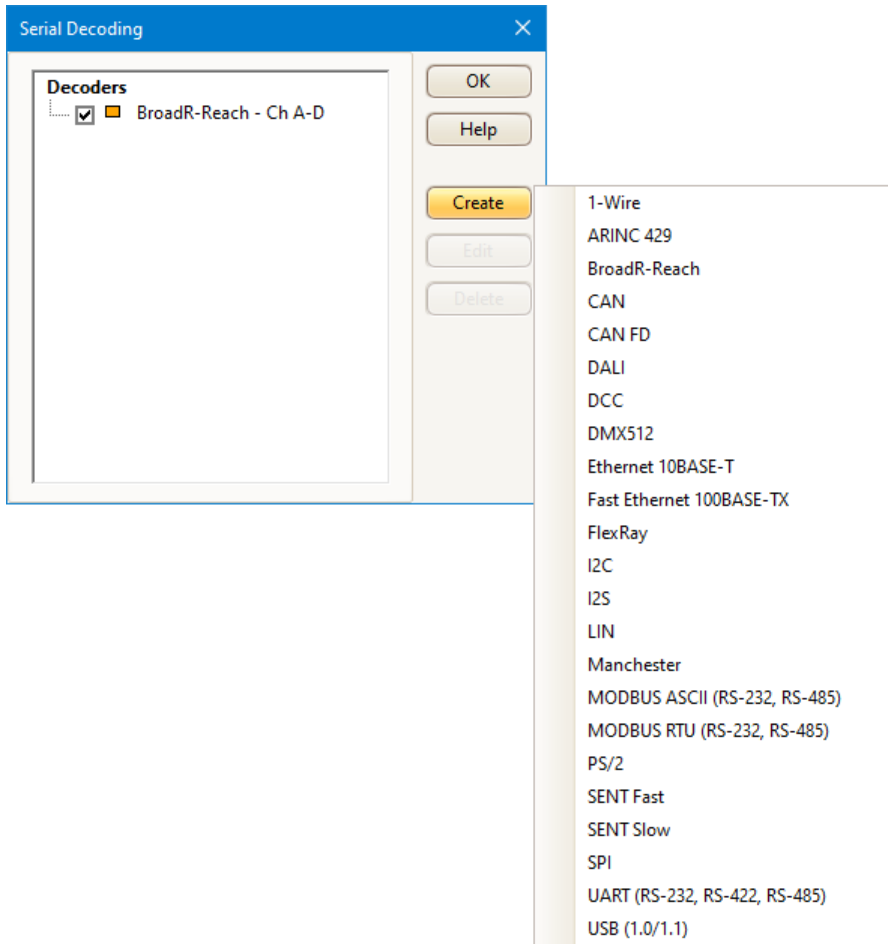


- Name:** PicoScope initially names the waveform after the input channel used as its source, but you can edit it to anything you like. Here we have named it *noise*. The name will appear in the waveform list in the [Reference Waveforms dialog](#).
- Color:** You can set the color of the trace to one of the standard colors in the drop-down-list, or click **Custom** to choose any color allowed by Windows.

## 5.5.4 Serial Decoding dialog

Location: [Tools](#) > **Serial Decoding**

Purpose: lets you set up which channels and [protocols](#) to use for [serial decoding](#)

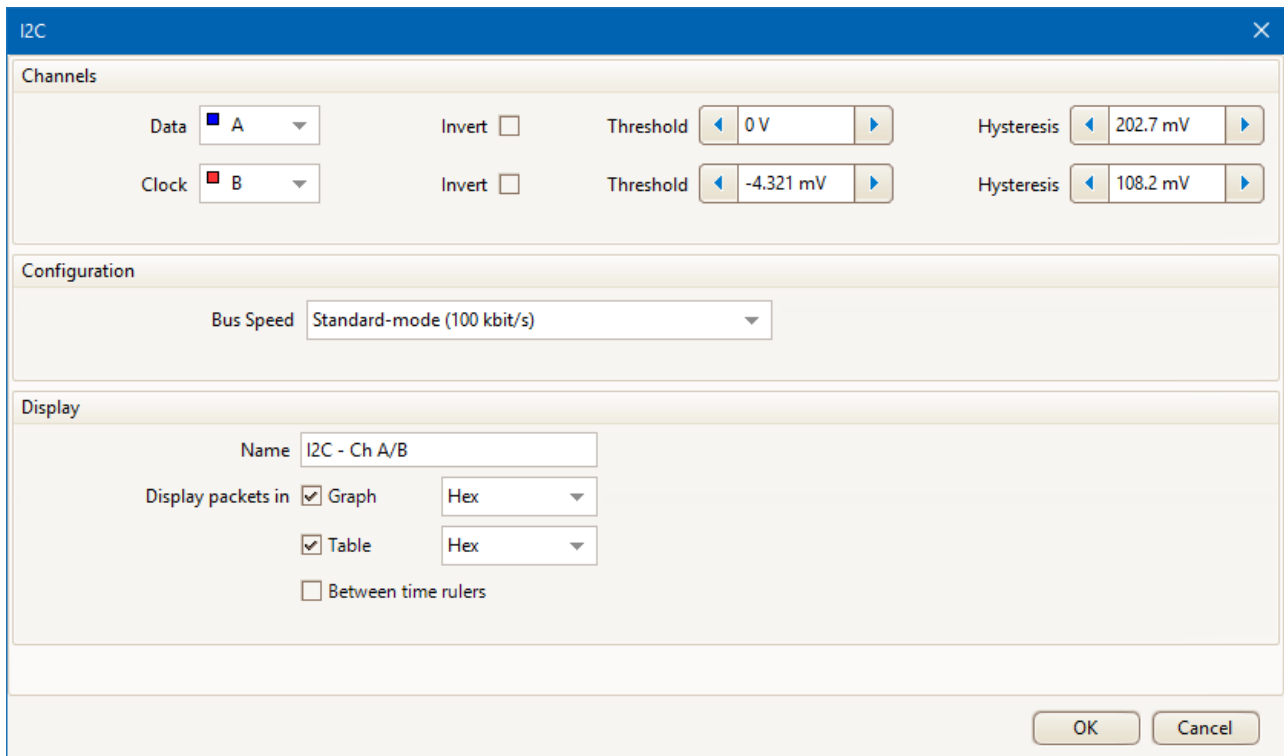


### Adding a new serial decoder

1. Make sure the channel you want to decode is enabled, using the [Channels toolbar](#). For duplex protocols such as UART, you must decode each direction (for example Tx and Rx) using a separate channel.
2. Click the **Create** button, as shown above, and select the serial decoding protocol you would like to use from the list.
3. This opens the [Decoder Settings](#) dialog: use this to select which channel to decode and how you would like to view the data, and to adjust other settings. Each protocol has different settings to adjust.
4. Click **OK**. Your selected protocols will appear in the **Serial decoders list**, as shown above, where you can edit or delete them.

### 5.5.4.1 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.

#### Channels

##### Signals

##### Data

Data channel for various protocols.

Note: Bidirectional SPI can be decoded as two separate buses, one with Data In (DI) as the I/O signal and the other with Data Out (DO) as the I/O signal. The Clock (and Chip Select if present) can be shared by both buses.

##### 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

##### Clock

Clock for various protocols

##### Word Select

For the I<sup>2</sup>S 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 and DataMinus

For the USB (1.0/1.1) protocol.

**Controls**

<b>Invert</b>	Reverses the polarity of the signal
<b>Threshold</b>	The voltage that defines the transition in either direction between high and low logic states
<b>Hysteresis</b>	Eliminates the effect of noise on the <b>threshold</b> , in a similar way to <a href="#">trigger hysteresis</a> . The value you enter here is a range, to be divided equally either side of the threshold.

**Configuration**

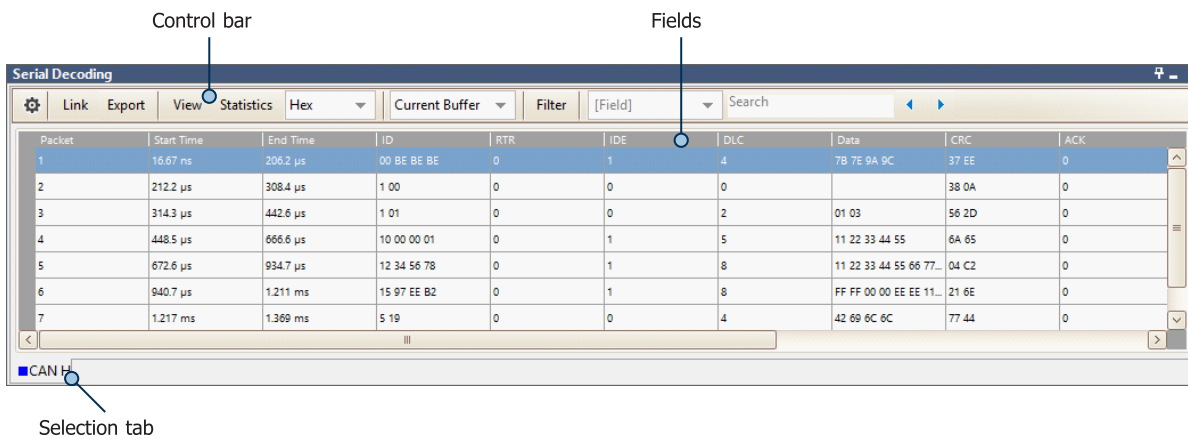
<b>Baud Rate</b>	The symbol frequency in baud. PicoScope will automatically detect the baud rate and select a value for you. Alternatively, you can choose one of the standard values from the drop-down list or enter an arbitrary value.
<b>Bit Order</b>	Whether the most or least significant bit occurs first.  For protocols with application-specific bit order, such as UART, you must set this to match to the data format.
<b>Bus Speed</b>	The maximum data rate of the I <sup>2</sup> C serial bus
<b>Checksum Length</b>	The number of words following the data field to decode as the checksum.
<b>Chip Select State</b>	The polarity of the Chip Select (CS) signal
<b>Clock Edge</b>	Whether to sample the clock (SCK) on the falling or rising edge
<b>Data Bits</b>	The number of bits in the data payload of each packet
<b>Data Bit Rate</b>	The bit rate (in baud) of the data phase of the signal, independent of bit encoding and decoding
<b>Data Encoding</b>	Encoding of the ARINC 429 data field: binary coded decimal (BCD), binary number representation (BNR), or discrete data representation
<b>Data Length</b>	The number of words decoded as the data field following the header.
<b>Data Length Type</b>	<b>Fixed:</b> The data field is a fixed length as specified. Used in protocols such as MIL-STD-1553. <b>Remaining:</b> 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.
<b>Display 4B/5B Code Groups</b>	Show the 4B/5B block coding scheme output (rather than Ethernet packets) for the Fast Ethernet protocol
<b>End Of Packet Byte</b>	For the Modbus ASCII protocol. Byte value of the line feed (LF) byte.
<b>First Slot</b>	The first DMX channel to display
<b>Full-speed (12 Mbit/s)</b>	Full speed USB 1.0 or 1.1 protocol
<b>Header Length</b>	The number of words decoded as the packet header after the preamble.
<b>High or Low</b>	The signal polarity of the CAN or CAN FD protocol

<b>High Threshold</b>	The voltage level of the high signal state
<b>Id Bytes</b>	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.
<b>Last Slot</b>	The last DMX channel to display
<b>Low-speed (1.5 Mbit/s)</b>	Low speed USB 1.0 or 1.1 protocol
<b>Low Threshold</b>	The voltage level of the low signal state
<b>Master</b>	Check to decode the master signals on a BroadR-Reach bus. Uncheck to decode the slave signals.
<b>Parity</b>	The type of error-correction bit added to each packet
<b>Packet Split Interval</b>	The time, in bit periods, between packets.
<b>Preamble Length</b>	The preamble is a fixed bit pattern that identifies the start of a frame.
<b>Sample time from falling edge</b>	The time since the last falling edge at which to sample the 1-Wire bus
<b>Sensor Type</b>	The type of SENT sensor attached. Defines the contents of the packet data nibbles.
<b>Start Bits</b>	The number of fixed bits (1s) at the start of each packet
<b>Stop Bits</b>	The number of fixed bits at the end of each packet
<b>Tick Time</b>	SENT transmitter-specific nominal clock period (tick)
<b>Word Length</b>	For I <sup>2</sup> S and Manchester encoding. Word length used in the packet.
<b>Display</b>	
<b>Name</b>	An arbitrary name for each protocol setup, which is displayed in the scope view.
<b>Display packets in</b>	<p><b>Graph:</b> shows the data in logic-analyzer style, on the same time axis as the analog waveform.</p> <ul style="list-style-type: none"> <li>• Hover the mouse pointer over any decoded packet to display its contents.</li> <li>• Click-and-drag the decoded data up or down the scope view.</li> <li>• If the Table display is visible, double-click on any packet to highlight it in the table.</li> </ul> <p><b>Table:</b> displays the decoded data in a table in the <a href="#">serial data window</a>, with advanced search and filtering functions.</p> <p><b>Between time rulers:</b> Only decodes the section of waveform shown between the <a href="#">time rulers</a>.</p> <p>You can also choose a format for the decoded data: <b>Hex, Binary, Decimal, or ASCII.</b></p>

### 5.5.4.2 Serial Decoding data window

Select **Table** in the [Decoder Settings](#) dialog to view the decoded serial data in an alphanumeric format below your waveform, as shown below. This format allows advanced filtering and searching of the data. You can also sort the data by any of the fields.

When **Graph** is also selected in the [Decoder Settings](#) dialog, the data will appear in graphical form on the scope view too. You can click on one of the data packets in the scope view to go to the corresponding row in the table, or you can double-click a row in the table to zoom to the corresponding frame in the scope view.



#### Control bar



Open the [Decoder Settings](#) dialog

#### Link

**Open:** Open a previously created link file, which translates numerical decoded data to text

**Create Template:** Create a link file with the correct field headers, but without any populated data

**Clear:** Clear the loaded link file

**Recent Files:** Files which have recently been saved or opened

#### Export

Save the decoded data in a link file, as an OpenDocument Spreadsheet (ODS) or Comma-Separated Values (CSV) text file

#### View

**Fields:** Choose which data fields are displayed

**Grid Font Size:** Change the size of the characters in the table

#### Statistics

View or hide the statistics columns, which list the signal voltages and packet time for each packet

#### Change numeric display format

Choose the format your data is displayed in: **Hex**, **Binary**, **Decimal** or **ASCII**

#### Buffer

Choose whether to display packets from the **Current Buffer** or **All Buffers**

**Filter** Allows you to reduce the size of the table by selecting which data to show. When you click the **Filter** button, text entry boxes appear above each column in the table. Choose a field to filter by and enter the value you are looking for. Only the packets which match your filter term will be displayed.

For example, enter 6C7 in the filter box at the top of the ID column, and PicoScope will only display the packets where the ID contains the value 6C7.

**[Field] and Search** Select the field you wish to search using the drop-down list, and then enter a search term in the search box. Use the arrows to scroll through matching data packets.

### Fields

The fields shown depend on which protocol has been selected. For more information on the fields, see the individual sections in [Serial protocols](#).

Click and drag the edges of each column to change the width, and make viewing the field data clearer.

Click on the field header to sort the table by that field, in ascending order. Click again to sort in descending order. To restore the default order, sort by **Packet**.

PicoScope sorts by value rather than by string. This is also the case if you are using a [link file](#) and have converted numerical codes into plain text: PicoScope will sort by the underlying value.

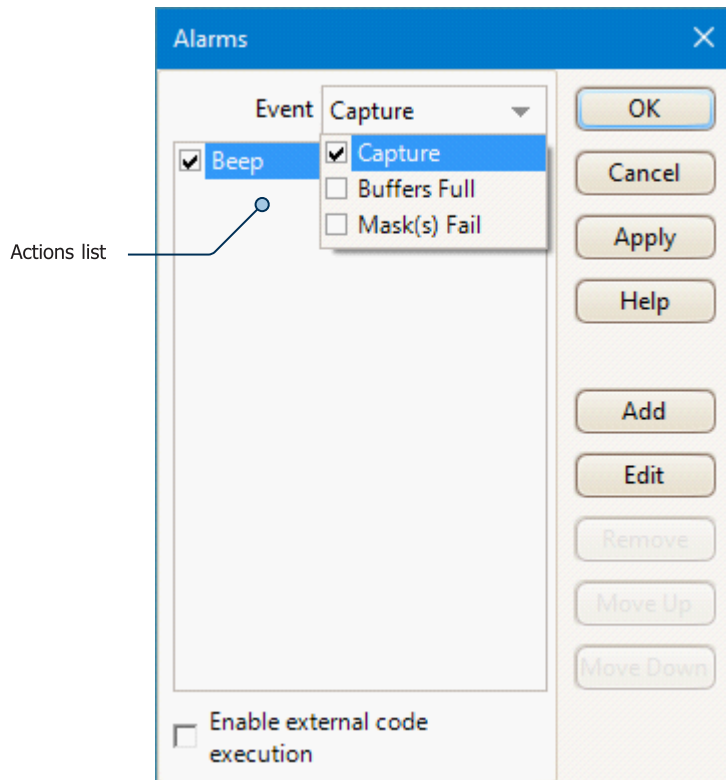
### Selection tabs

A separate table will be displayed for each serial decoding setup created, with different channels and protocols. Use the tabs to switch between them.

## 5.5.5 Alarms dialog

Location: [Tools](#) > **Alarms**

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



**Event:** Select the event(s) that will trigger the alarm:

**Capture:** when a waveform is captured. If [triggering](#) is enabled, this option corresponds to a trigger event. You can therefore use this function to save a file on each trigger event.

**Buffers Full:** when the number of waveforms in the [waveform buffer](#) reaches the [maximum waveform](#) count.

**Mask(s) Fail:** when any channel fails a [mask test](#).

**Note:** for an event to be active, its check box must be selected.

**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. For an action to execute, its check box must be selected.

**Apply**

Set up the scope according to the settings in this dialog.

**Add**

Add an event to the **Actions list**. Possible events are:

**Beep:** activate the computer's built-in sounder. 64-bit PCs redirect this sound to the headphone output.



**Play Sound:** specify the name of a .wav sound file to play.

**Stop Capture:** equivalent to pressing the red [Stop button](#).

**Restart Capture:** equivalent to pressing the green [Go button](#). Use this action only if you have used the **Stop Capture** action earlier in the list.

**Run Executable:** run the specified .exe, .com or .bat 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.

**Save Current Buffer:** save the current waveform from the buffer as a .psdata or.pssettings 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.

**Save All Buffers:** save the entire waveform buffer as a file.

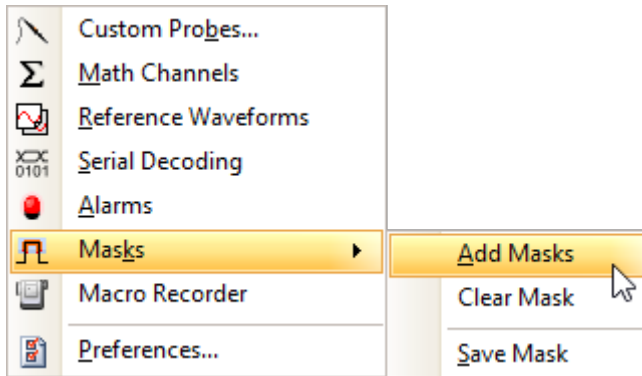
**Trigger Signal Generator:** if the scope device has a [triggerable signal generator](#), start generating a signal. The signal generator's **Trigger Source** option must be set to **Manual** for this operation to work.

**Enable external code execution** As a security feature, to enable **Run Executable** you must also select the **Enable external code execution** option.

## 5.5.6 Masks menu

Location: [Tools](#) > **Masks**

Purpose: gives control over [mask limit testing](#)



**Add Masks** Add a mask to the display using the [Mask Library dialog](#)

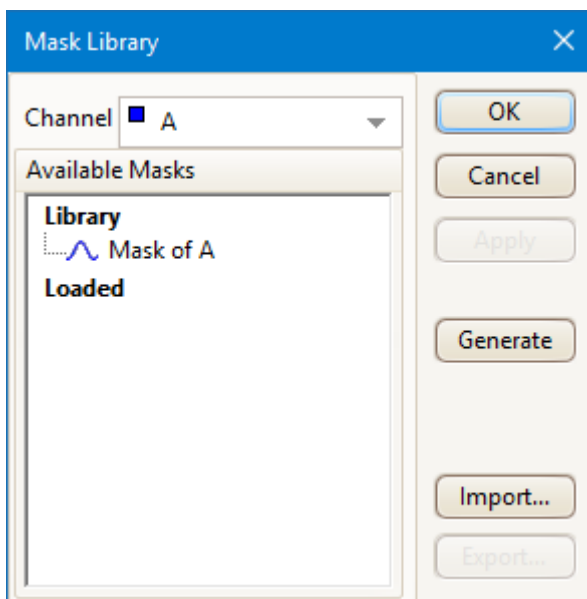
**Clear Mask** Remove the mask from the display

**Save Mask** Save the displayed mask to disk as a .mask file

### 5.5.6.1 Mask Library dialog

Location: [Tools](#) > **Masks** > **Add Masks**

Purpose: allows you to create, export and import masks for [mask limit testing](#)



**Channel** Select the channel to which you want to apply the mask

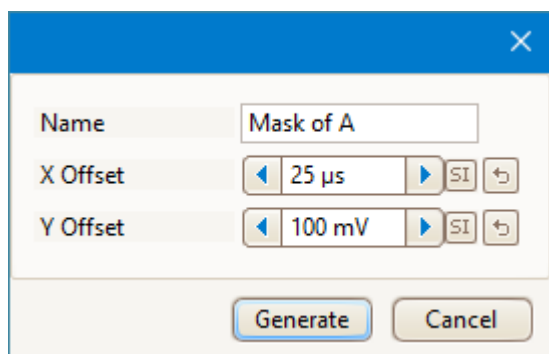
**Available masks** The **Library** section shows all the masks that you have saved in the past and not deleted

<b>Generate</b>	Create a new mask based on the last waveform captured from the selected channel. Opens the <a href="#">Generate Mask dialog</a> .
<b>Import</b>	Load a mask that has previously been saved as a .mask file
<b>Export</b>	Save a mask as a .mask file for future import
<b>Apply</b>	Use the selected mask on the selected channel but remain in the Mask Library dialog
<b>OK</b>	Use the selected mask on the selected channel and return to the <a href="#">scope view</a>

### 5.5.6.2 Generate Mask dialog

Location: [Mask Library dialog](#) > **Generate**

Purpose: allows you to set parameters for the automatically generated mask. PicoScope will then create a new mask based on the last captured waveform.



**Name:** PicoScope automatically chooses a name for the new mask. You can edit the name in this box.

**X Offset:** The horizontal distance between the waveform and the mask.

**Y Offset:** The vertical distance between the waveform and the mask.



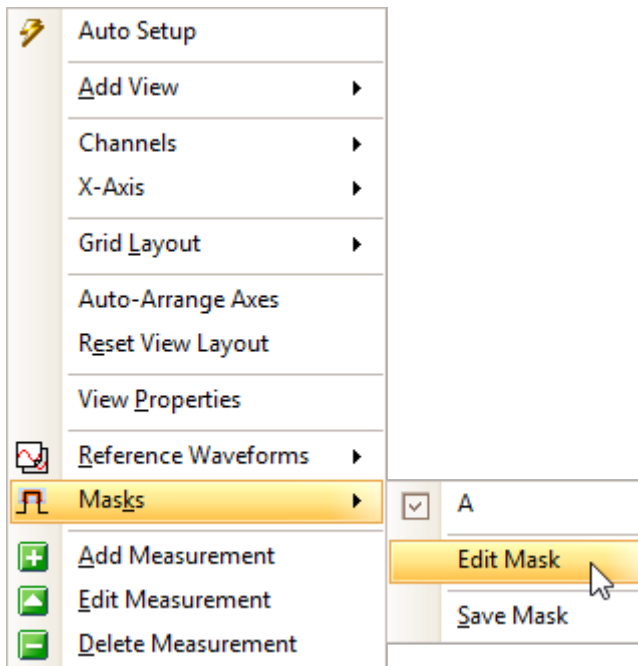
This button toggles the offset value between absolute units (SI) and relative units (% of full scale).



This button resets the offset value to its default.

### 5.5.6.3 Editing a mask

To edit a mask in [mask limit testing](#) mode, right-click on the [scope view](#) and select **Masks > Edit Mask**:



A mask is made up of one or more shapes called **polygons**. Click on the polygon you wish to edit. PicoScope will then draw edit handles on the selected mask polygon and display the mask edit box. If you drag any of the handles to edit the polygon, the statistical results will be updated immediately.






The mask edit box looks like this:

X	Y
500 $\mu$ s	2 V
0 s	2 V
0 s	1.185 V
44.5 $\mu$ s	1.183 V
45 $\mu$ s	1.167 V
45.75 $\mu$ s	1.167 V
47.75 $\mu$ s	1.131 V
48.25 $\mu$ s	1.115 V
49.25 $\mu$ s	1.114 V
50.25 $\mu$ s	1.078 V
51.75 $\mu$ s	1.056 V
52.5 $\mu$ s	1.028 V
53.25 $\mu$ s	1.018 V
58.25 $\mu$ s	849.5 mV

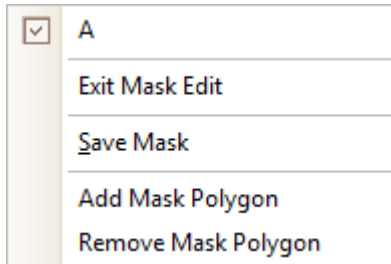
Normal view



Minimized view

If the edit box is not immediately visible, it may have been minimized; if this is the case, click the restore button . If you edit the coordinates of a vertex, the statistical results will be updated immediately. You can also export the mask to a .mask file with the export button . Use the  and  buttons to add or remove vertices. The minimize button has its usual function. To leave mask editing mode, close the mask edit box with the close button .

To add or remove an entire polygon, right-click the scope view and select either **Add Mask Polygon** or **Remove Mask Polygon**:

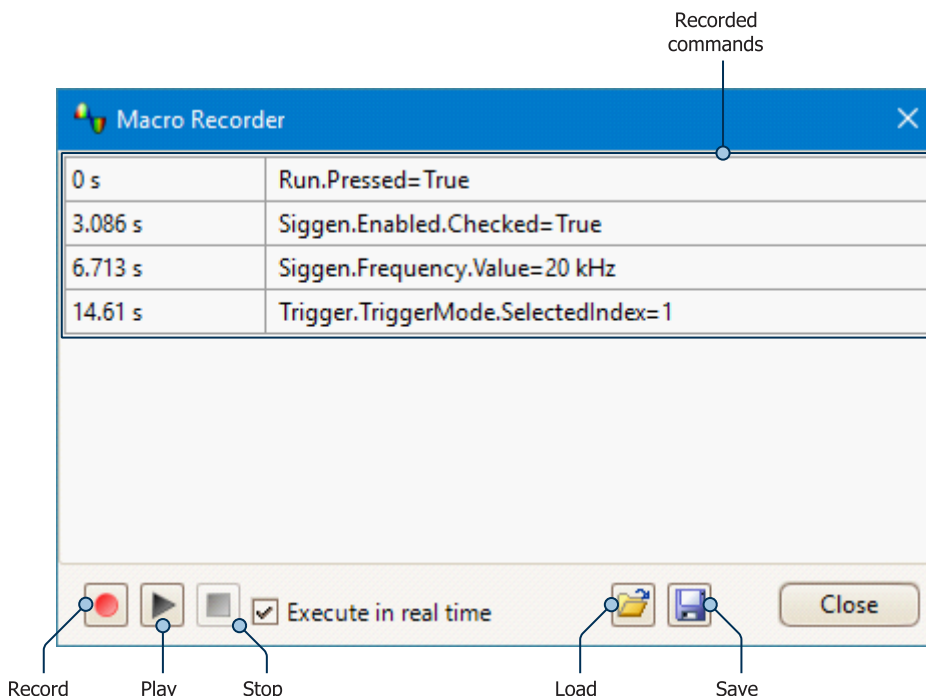


## 5.5.7 Macro Recorder

Location: [Tools](#) > **Macro Recorder**

Purpose: records a sequence of commands to be replayed later

The **Macro Recorder** is useful when you want to execute a series of commands repeatedly. It saves all commands to a .psmacro file, which you can modify using an XML editor.



**Execute in real time** Play back the macro at the same speed as it was recorded. When this box is not checked, playback will be as fast as possible.

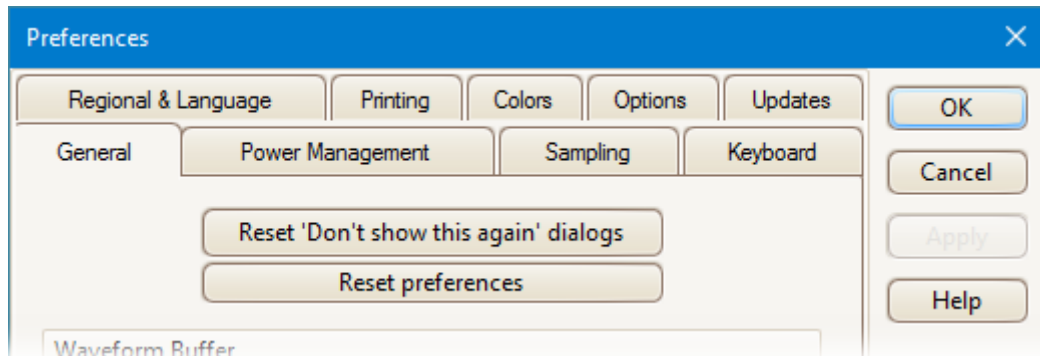
**Load** Load a macro from a file

You can also play back .psmacro files from the PicoScope command line. See [Command-line syntax](#) for information on this and other macro commands.

## 5.5.8 Preferences dialog

Location: [Tools](#) > **Preferences**

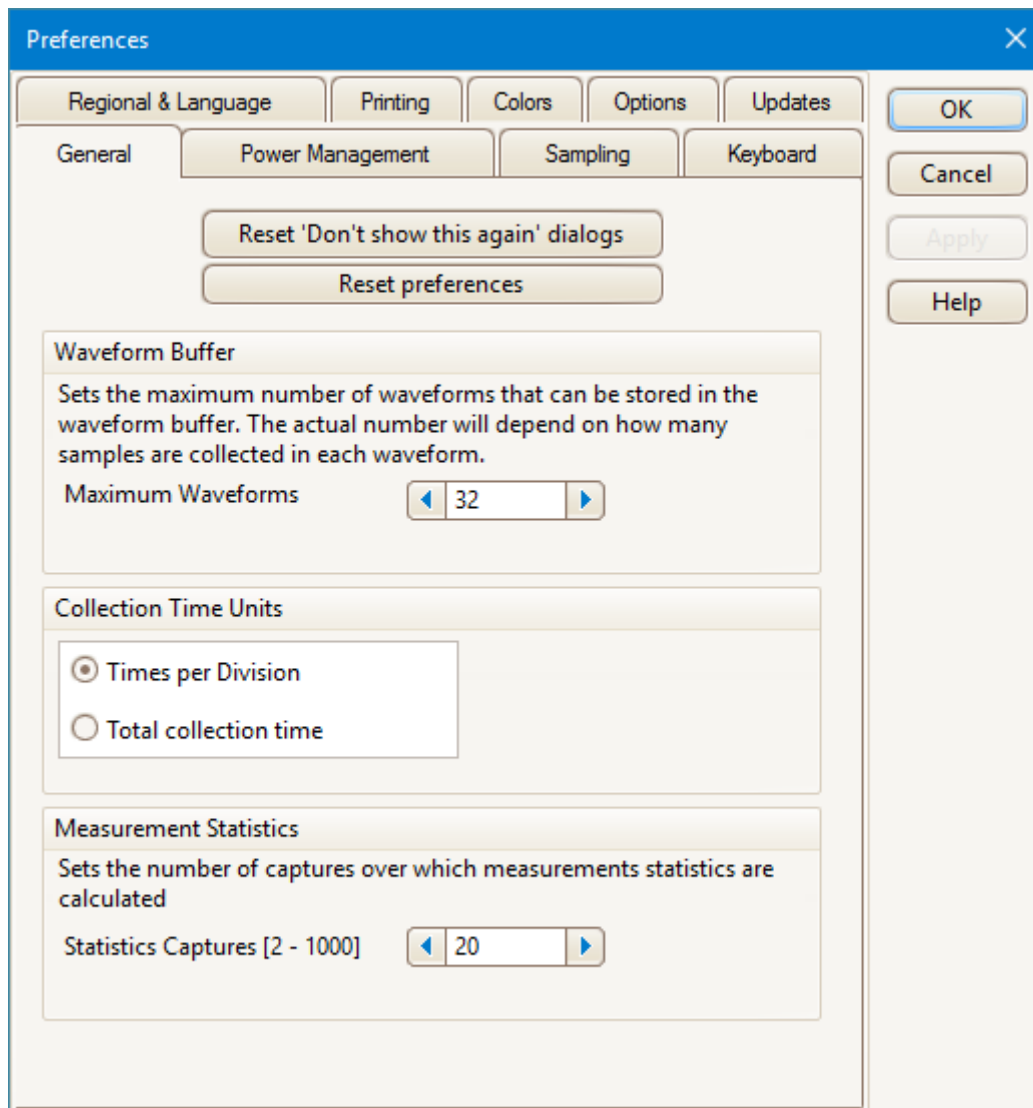
Purpose: Allows you to set options for the PicoScope software. Click one of the tabs in the image below to learn more.



### 5.5.8.1 General page

Location: [Tools](#) > [Preferences](#) > **General**

Purpose: contains general controls for PicoScope



**Reset "Don't show this again" dialogs** Restore any missing dialogs that you asked PicoScope not to show again.

**Reset preferences** Set all preferences back to their default values.

**Waveform Buffer** **Maximum Waveforms:** the maximum number of waveforms that PicoScope will store in the [waveform buffer](#). You can select any number from 1 to the maximum allowed by the oscilloscope that is connected: see the scope specifications for details. The actual number of waveforms stored depends on the available memory and the number of samples in each waveform.

**Collection Time Units** Choose how the [collection time](#) is displayed in the [Capture Setup toolbar](#).

**Times per division:** collection time is expressed in units of time per division, for example 5 ns/div.

**Total collection time:** the collection time shows the length of time represented by the entire width of the scope view, for example 50 ns.

The horizontal axis of the PicoScope display has ten divisions, with the time marked at each, so 5 ns/div and 50 ns represent exactly the same horizontal scale.

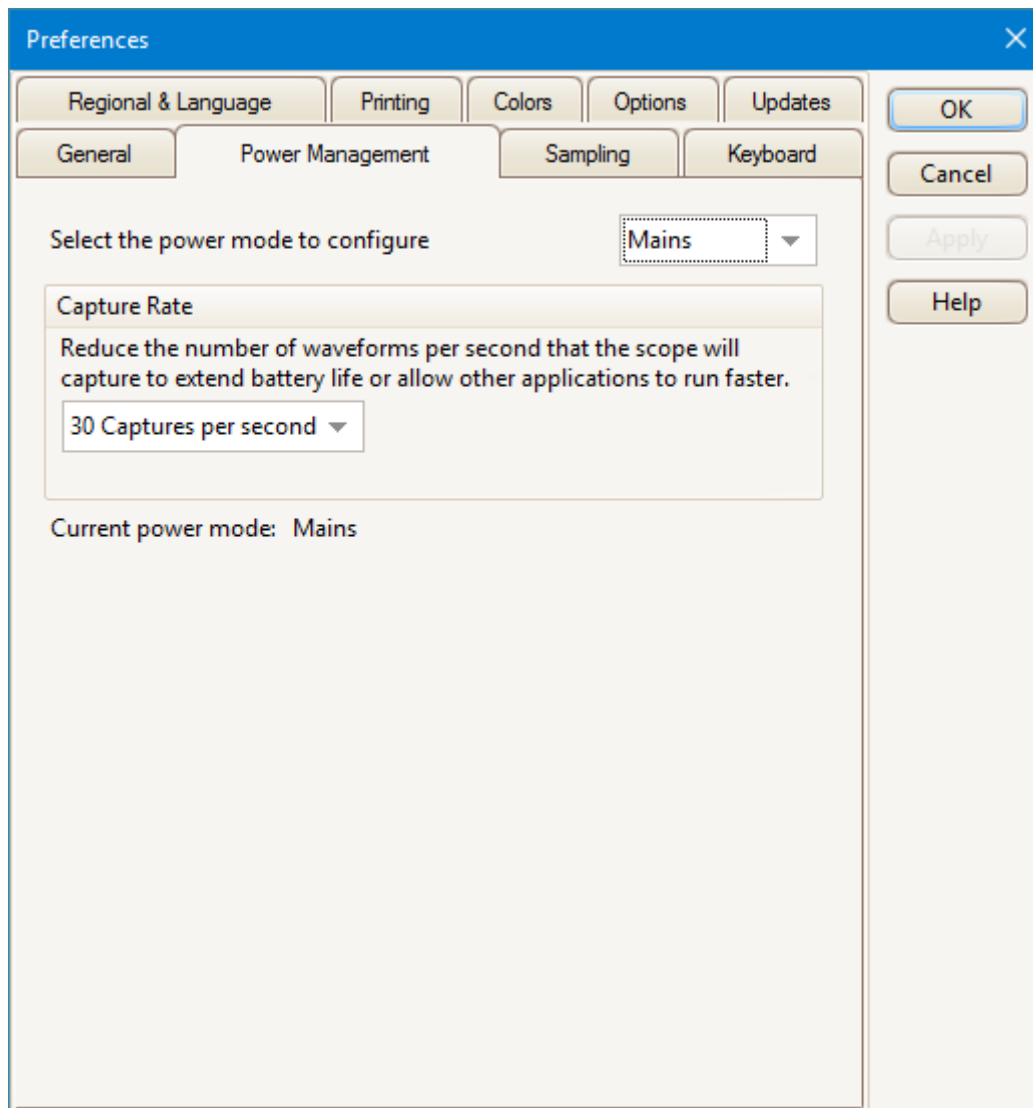
**Measurement Statistics Capture Size:** the number of successive captures that PicoScope uses to calculate the statistics in the [Measurements Table](#). A larger number produces more accurate statistics but causes them to be updated less frequently.



### 5.5.8.2 Power Management page

Location: [Tools](#) > [Preferences](#) > **Power Management**

Purpose: controls features of the oscilloscope that affect power consumption



#### Capture Rate

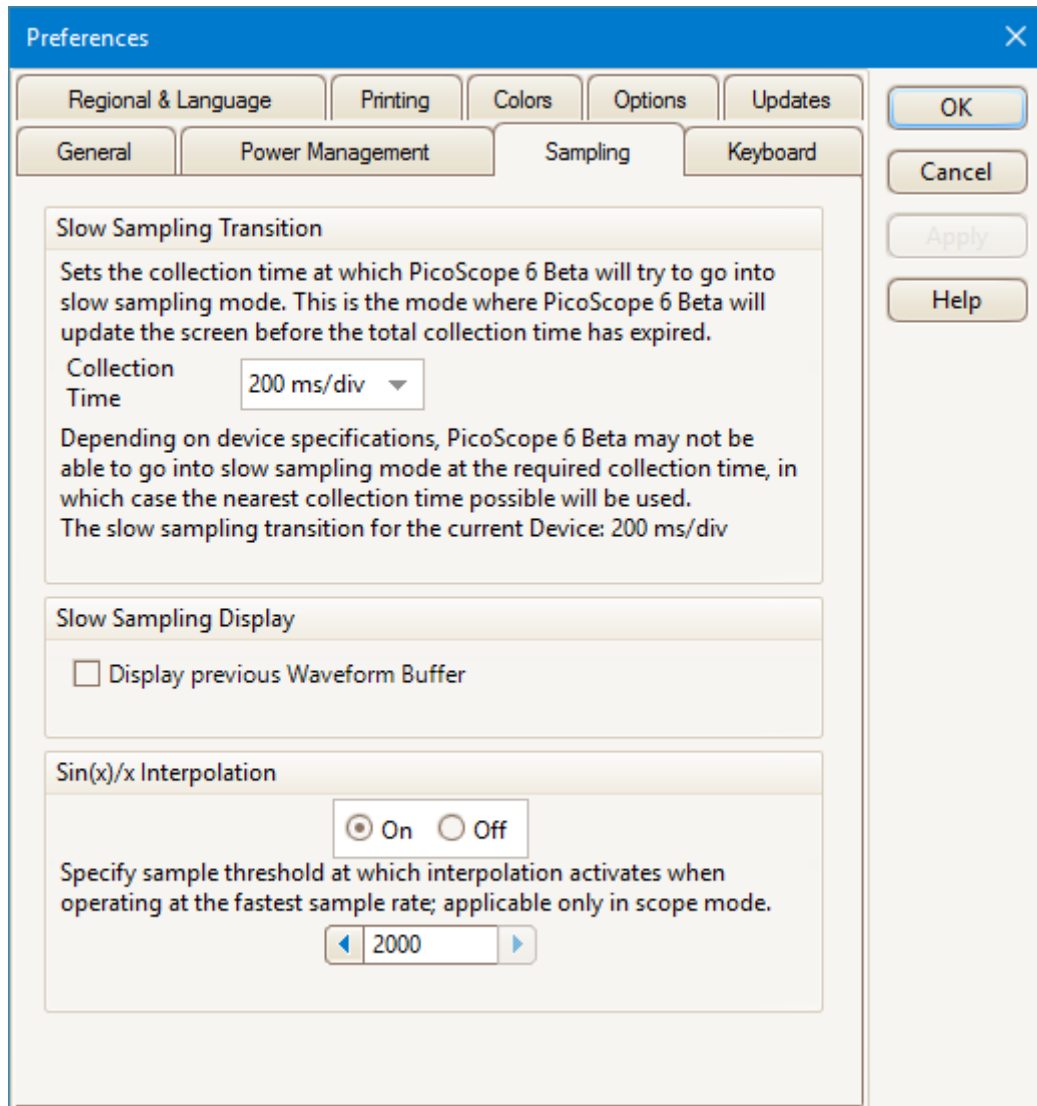
This control limits the speed at which PicoScope captures data from the [scope device](#). The other PicoScope settings, the type of scope device and the speed of the computer will all affect whether this limit can actually be reached. PicoScope automatically selects the appropriate limit according to whether your computer is running on batteries or on mains (line) power.

The settings are in captures per second. By default, the capture rate is set to **30 Captures per second** when your computer is running on **Mains** (line) power, for maximum performance. If other applications on your PC run too slowly while PicoScope is capturing, reduce the capture rate. When your computer is running on **Battery** power, PicoScope imposes a performance limit to save the battery. You can increase this limit manually, but this will cause the battery to drain very quickly.

### 5.5.8.3 Sampling page

Location: [Tools](#) > [Preferences](#) > [Sampling](#)

Purpose: controls the sampling behavior of the oscilloscope



#### Slow Sampling Transition

In normal (fast) sampling mode, PicoScope collects enough data to fill the screen and then redraws the whole view at once. This method is suitable for short collection times, when the screen is redrawn many times each second, but with long collection times it can cause an unacceptable delay before the data appears on the screen. To avoid this delay, PicoScope automatically switches to slow sampling mode, in which the scope trace advances gradually across the screen as the scope captures data. The **Collection Time** control lets you choose the collection time at which PicoScope makes this switch.

#### Slow Sampling Display

When this box is checked, PicoScope displays the previous waveform in the buffer while gradually redrawing the new waveform over the top of it. Thus, at any time, the left-hand side of the view shows the beginning of the new waveform, while the right-hand side shows the end of the previous waveform. A vertical bar separates the two waveforms. This operation uses the Fast Streaming Mode capability of the PicoScope oscilloscope hardware.

**Sin(x)/x Interpolation**

When the number of pixels across the scope view is greater than the number of samples in the waveform buffer, PicoScope interpolates - that is, it fills the space between the samples with estimated data. It can either draw straight lines between the samples (linear interpolation) or connect them with smooth curves (sin(x)/x interpolation). Linear interpolation makes it easier to see where the samples are, which is useful for high-accuracy measurements, but results in a jagged waveform. Sin(x)/x interpolation gives a smoother waveform but disguises the true locations of the samples, so should be used with care when the number of samples on the screen is low.

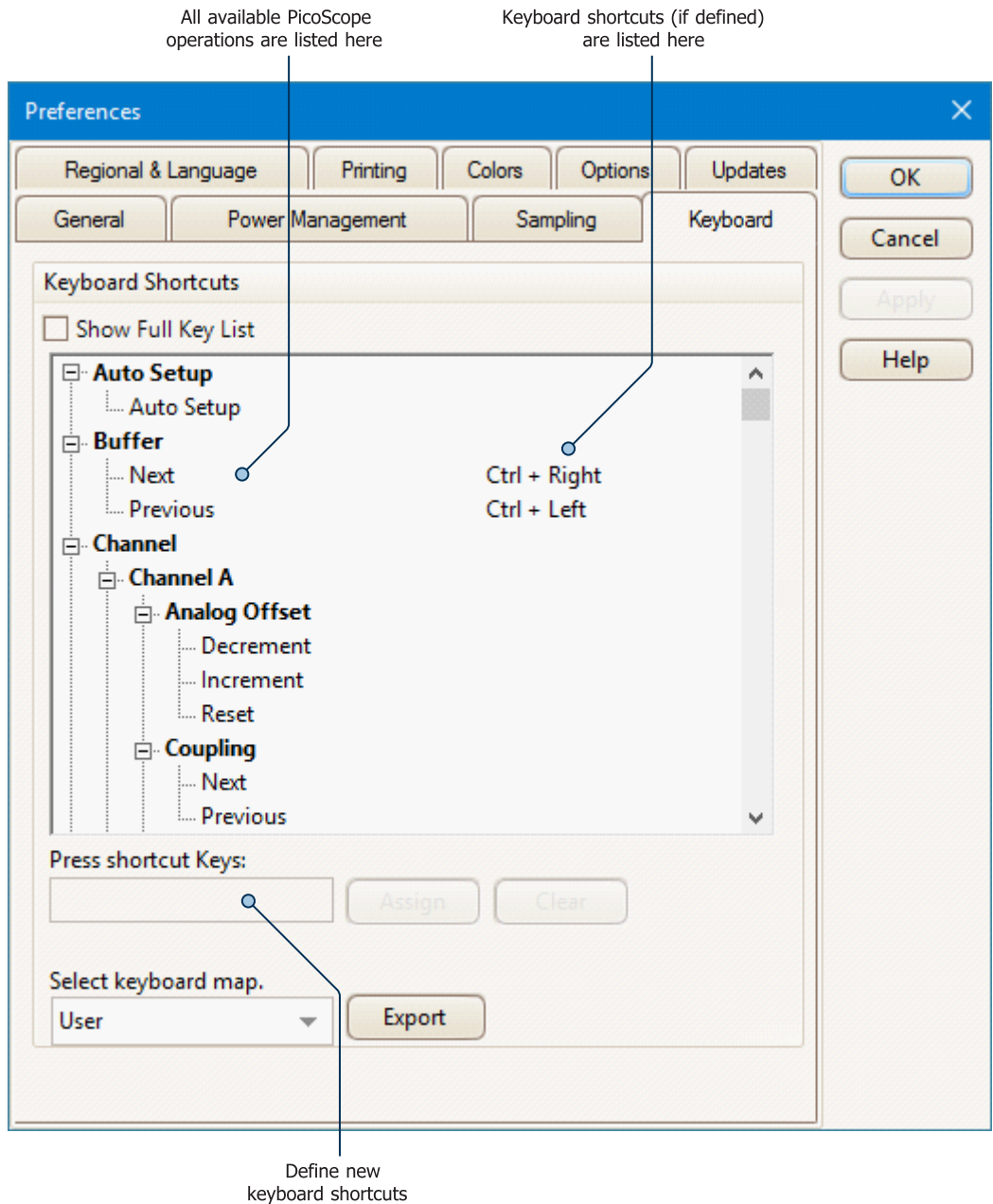
You can adjust the number of samples below which sin(x)/x interpolation is switched on. Sin(x)/x interpolation is used only on the scope's shortest collection time.

### 5.5.8.4 Keyboard page

Location: [Tools](#) > [Preferences](#) > [Keyboard](#)

Purpose: displays, and allows you to edit, keyboard shortcuts

A keyboard shortcut is a combination of keys that can be pressed on the keyboard to activate a PicoScope operation.



**Keyboard Shortcuts** This is a list of PicoScope operations and their associated keyboard shortcuts (if defined). The extent of the list depends on the **Show Full Key List** option (see below).

To edit or add a keyboard shortcut:

- Scroll through the list of PicoScope commands until the required operation is visible
- Select the required operation
- Click the **Press shortcut Keys:** box
- Press the required key combination on the keyboard
- Click **Assign**

**Show Full Key List** Check this box to show all available operations. By default, only the most common operations are listed, plus any other operations that have a keyboard shortcut assigned to them.

**Select keyboard map** A set of keyboard shortcuts is called a **map**. You can define multiple maps for different applications.

**Default:** This map cannot be edited. Use it to return to the factory-defined basic shortcuts.

**Advanced (QWERTY)** and **Advanced (DVORAK):** These are some more factory-defined maps that cannot be edited. They contains more comprehensive sets of shortcuts, for two different keyboard layouts.

**User:** This is the map that you most recently created or imported. It is preserved between PicoScope sessions.

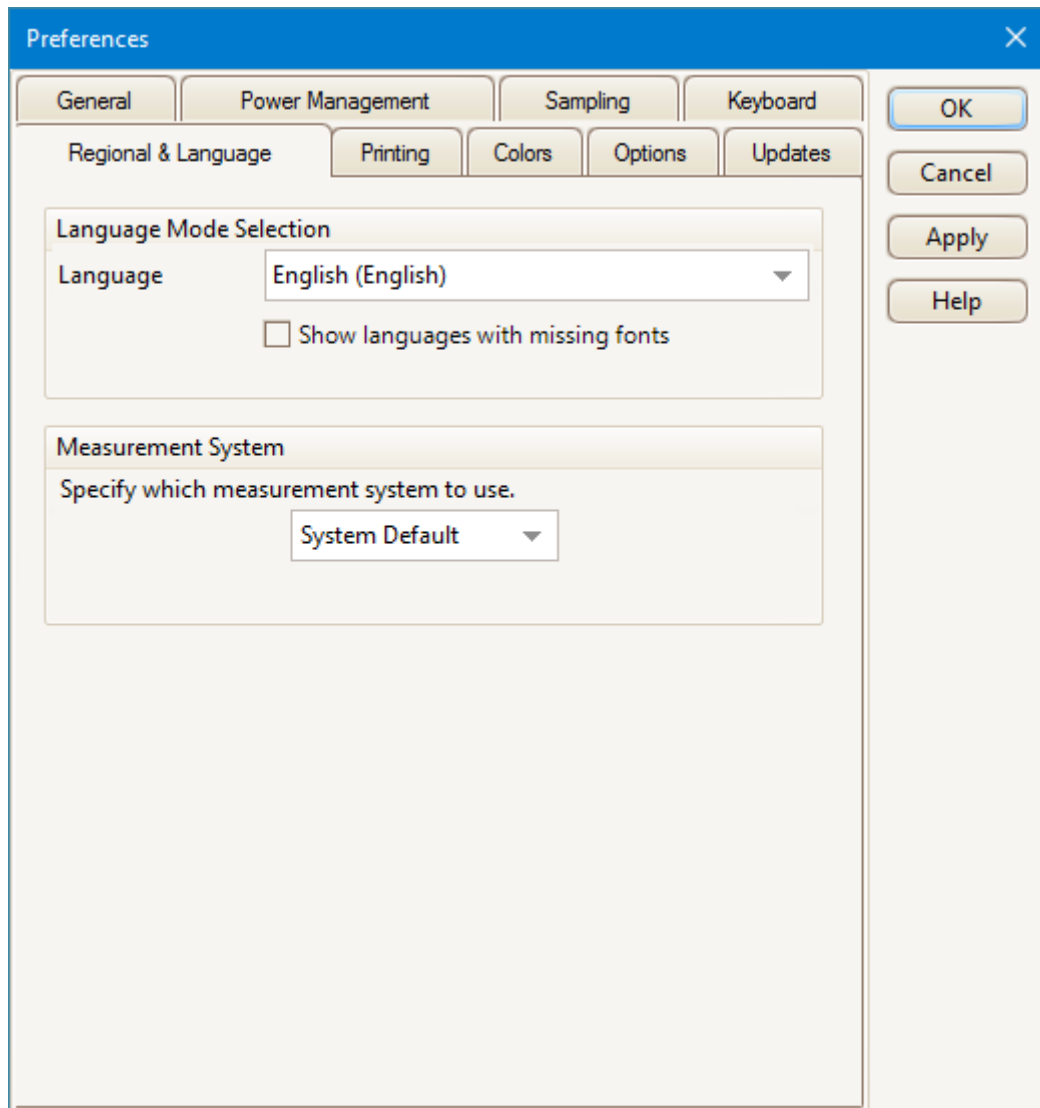
**Import:** Load a keyboard map from a `.pskeys` file

**Export** Save the current keyboard map to a `.pskeys` file

### 5.5.8.5 Regional & Language page

Location: [Tools](#) > [Preferences](#) > [Regional & Language](#)

Purpose: lets you select the language and other location-dependent settings for PicoScope's user interface



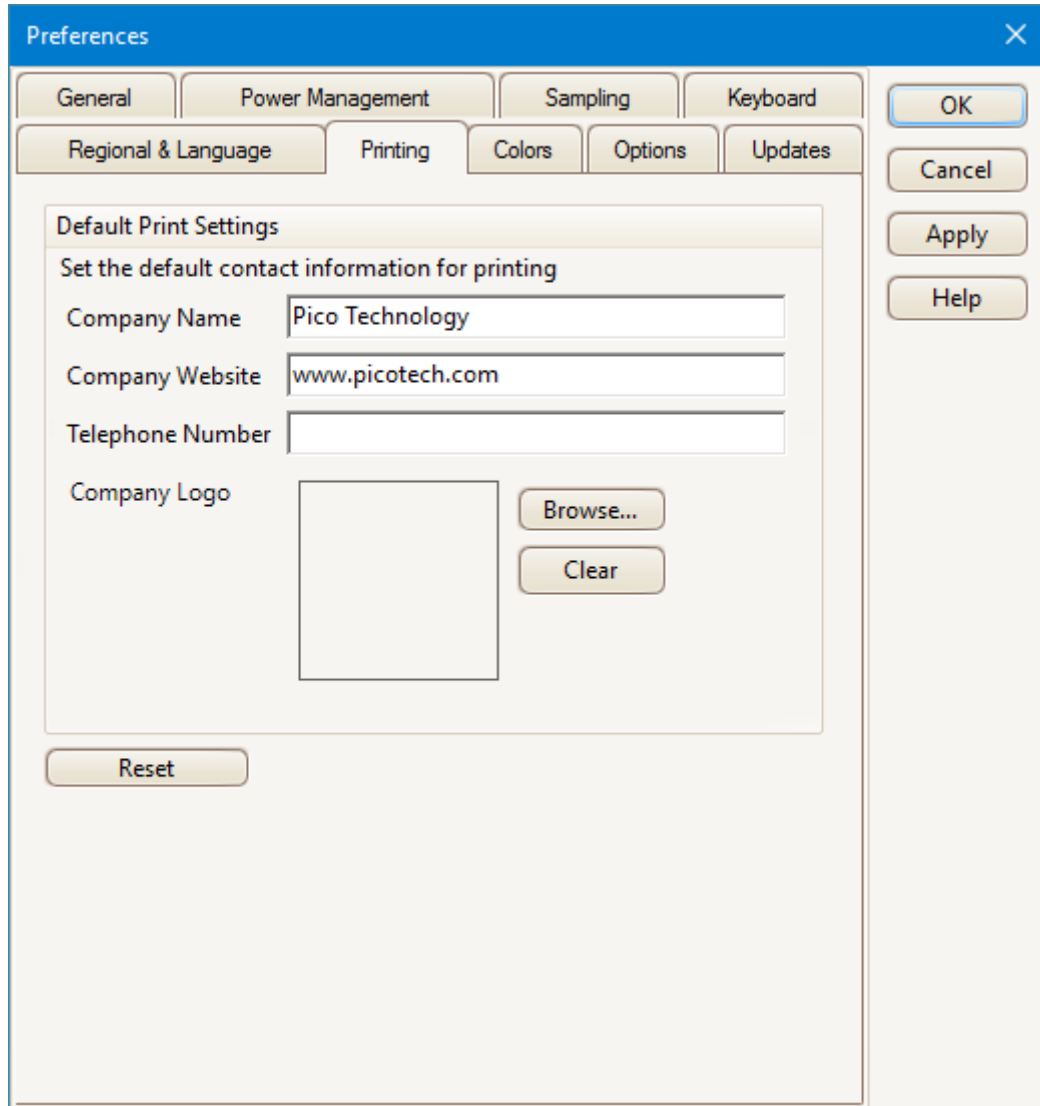
**Language** Select the language you wish to use for the PicoScope 6 user interface from the drop-down box. PicoScope will ask you to restart the program before switching to the new language.

**Measurement System** Select metric or U.S. units.

### 5.5.8.6 Printing page

Location: [Tools](#) > [Preferences](#) > [Printing](#)

Purpose: lets you enter the details that will appear at the bottom of any printed output

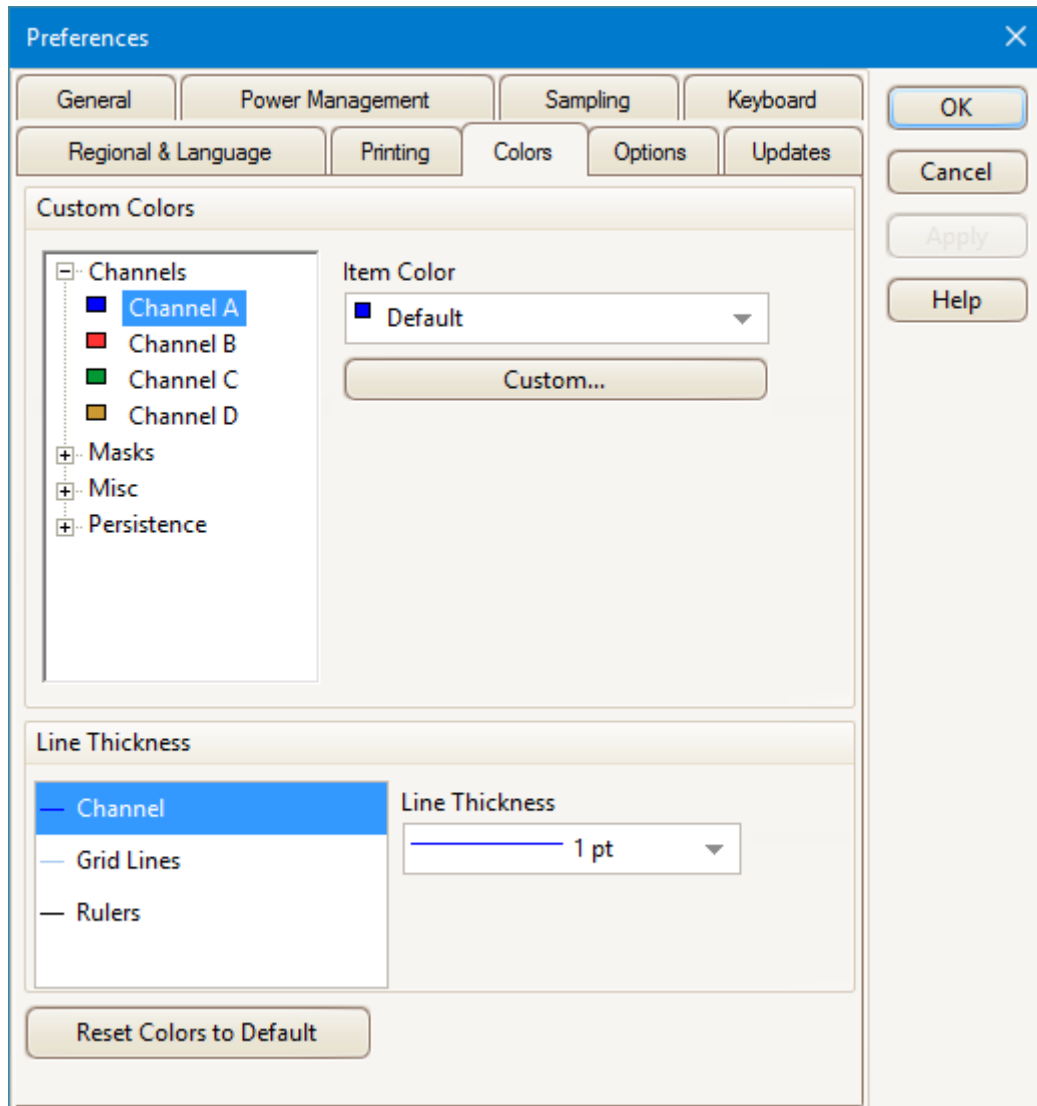


**Default Print Settings** When you print a view from the [File menu](#), these details will be added to the bottom of the page.

### 5.5.8.7 Colors page

Location: [Tools](#) > [Preferences](#) > [Colors](#)

Purpose: lets you set the colors for various parts of the user interface



#### Custom Colors

These controls let you specify the colors for various parts of the PicoScope screen:

- Channels**                    the trace color for each [scope channel](#)
- Digital Channels**        if you have a [mixed-signal oscilloscope \(MSO\)](#), the color of each channel can be set here
- Masks**                    the mask areas in [Mask Limit Testing](#)
- Misc**                      miscellaneous items on the display, including:
  - Grid lines**                the horizontal and vertical lines on the [graticule](#)



<b>Background</b>	the area behind the waveforms and graticule. (In <a href="#">persistence mode</a> , you can override this setting using the <a href="#">Persistence Options dialog</a> .)
<b>Live trigger</b>	the <a href="#">trigger marker</a> for the current trigger position
<b>Trigger</b>	secondary trigger marker (appears when the live trigger has moved since the last waveform capture)
<b>Horizontal axis</b>	the numbers across the bottom of each <a href="#">view</a> , which usually indicate time measurements
<b>Rulers</b>	the horizontal and vertical <a href="#">rulers</a> that you can drag into position to help measure features on the waveform

**Persistence** the three colors to use for each channel in digital color [persistence mode](#). The top color is used for the most frequently hit pixels, the middle and bottom colors for the less and least frequently hit pixels.

### Line Thickness

These controls let you specify the thickness of the lines drawn on the [scope](#) and [spectrum](#) views:

<b>Channel</b>	the waveforms and spectrum traces for all scope channels
<b>Grid Lines</b>	the horizontal and vertical lines on the <a href="#">graticule</a>
<b>Rulers</b>	the horizontal and vertical <a href="#">rulers</a> that you can drag into position to help measure features on the waveform

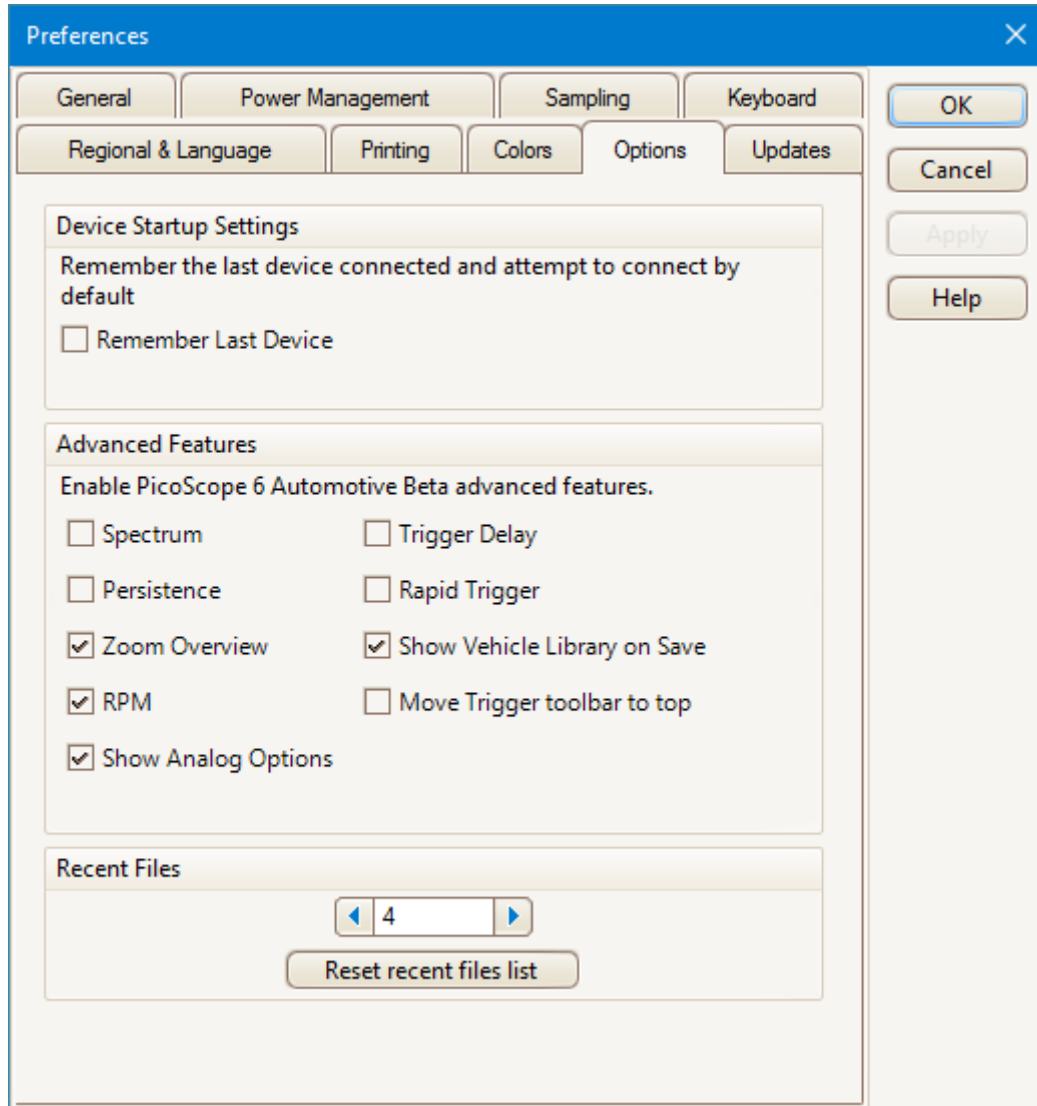
### Reset Colors to Default

Resets all of the color and line thickness settings to their default values.

### 5.5.8.8 Options page

Location: [Tools](#) > [Preferences](#) > [Options](#)

Purpose: lets you set various options that control the way that PicoScope 6 works



#### Device Startup Settings

**Remember Last Device.** This option is used when PicoScope finds more than one scope device. If the box is checked, PicoScope will attempt to use the same device that was used last time. Otherwise it will use the first device available.

#### Advanced Features

The advanced [capture modes](#) are enabled by default in PicoScope 6 and disabled by default in PicoScope 6 Automotive. Regardless of the version you have, you can enable or disable these features using the following options:

**Spectrum** [Spectrum view](#) features

**Persistence** Digital Color, Analog Intensity and custom [persistence display](#) modes

<b>Zoom Overview</b>	A window that appears when you <a href="#">zoom in</a> , to help you move around large waveforms with the minimum of mouse clicks
<b>RPM</b>	Revolutions per minute, displayed alongside hertz in the <a href="#">frequency legend</a>
<b>Show Analog Options</b>	Allows you to adjust settings such as <b>DC Offset</b> in the <a href="#">Channel Options menu</a>
<b>Trigger Delay</b>	The <a href="#">Time-delay</a> function in the <a href="#">Trigger controls</a>
<b>Rapid Trigger</b>	Enables the <b>Rapid</b> trigger mode in the <a href="#">Trigger controls</a>
<b>Show Vehicle Library on Save</b>	PicoScope 6 Automotive only. Opens the <a href="#">Details dialog</a> before the <b>Save As...</b> dialog when you save a waveform, enabling you to record details about the vehicle and the customer.
<b>Move Trigger toolbar to top</b>	The <a href="#">Trigger toolbar</a> , containing the <a href="#">Trigger controls</a> and the <a href="#">Stop and Go, Measurements</a> and <a href="#">other</a> buttons, is at the bottom of the PicoScope window by default. This option moves it to the top, immediately below the <a href="#">Channels toolbar</a> .
<b>Show Analog Options</b>	Enables the <a href="#">Analog Options</a> in the <a href="#">Channel Options menu</a>

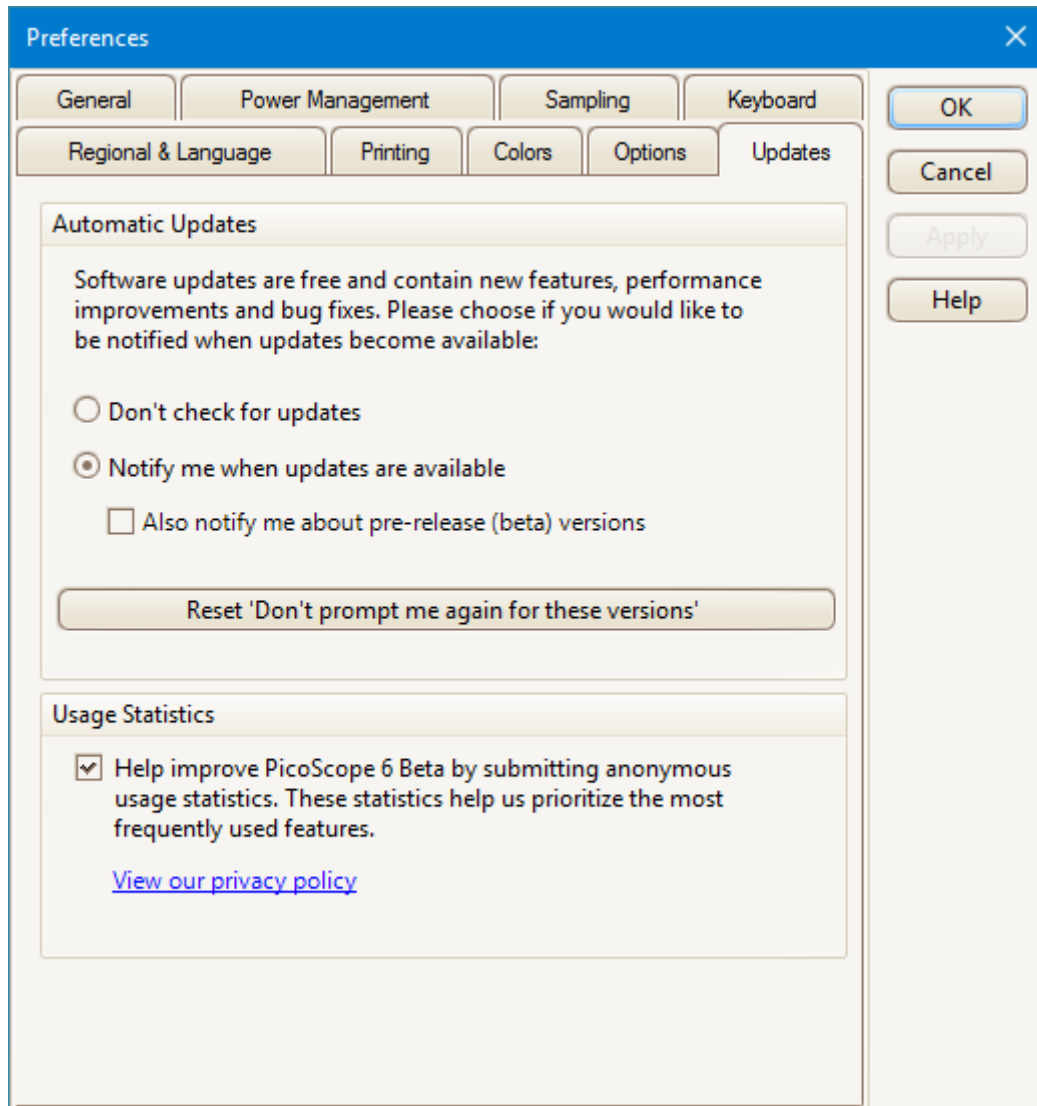
#### Recent Files

The maximum number of files listed in the [File](#) > [Recent Files](#) menu. Click the button to clear the list.

### 5.5.8.9 Updates page

Location: [Tools](#) > [Preferences](#) > [Updates](#)

Purpose: configures the automatic update checker and related services



#### Automatic Updates

##### Don't check for updates

If you prefer, you can check for updates yourself, either by clicking the **Check for updates** command in the [Help menu](#), or by visiting [www.picotech.com/downloads](http://www.picotech.com/downloads).

##### Notify me when updates are available

PicoScope will regularly check for updates to your software. Note: this option requires an internet connection.

##### Reset "Don't prompt me again for these versions"

If you have checked the box marked **Don't notify me about selected update(s) again** in the Update Available dialog, this button enables reminders again.

## Usage Statistics

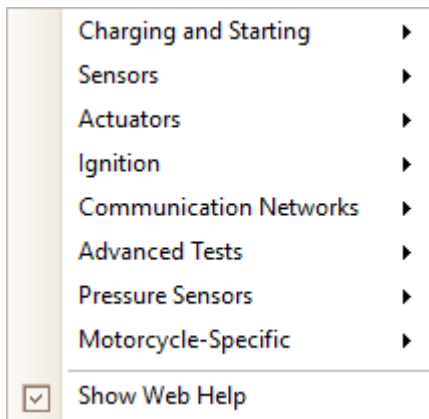
To help us to continue to improve PicoScope, the program periodically sends anonymous information back to our developers about which features you use the most. This information does not contain your name, email address or any other personal details, but does allow us to identify which country you are in. If you do not wish to send us this information, uncheck this box.

## 5.6 Automotive menu

Location: [Menu bar](#) > **Automotive**

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

Applicability: PicoScope 6 Automotive only



Show Web Help: when you select a test with this box checked, a web-based guidance page for that test opens, with full instructions for carrying out the test, along with some connection and technical information. If you clear this box, guided tests will be disabled, although PicoScope will still display an example waveform and adjust the settings for the preset test.

The Automotive menu contains over 150 preset guided tests ready for you to perform on vehicles.

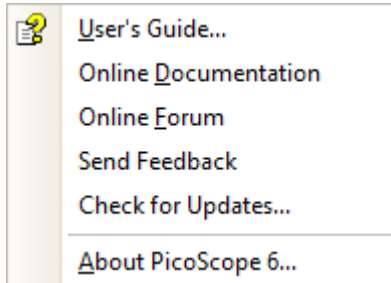
### To run a guided test

1. Select the test you wish to run.
2. If you have checked the **Show Web Help** box, PicoScope opens the relevant guided test page that explains how to set up the test for the scope you have connected, run the test and interpret the results (a few tests do not have an information page).
3. PicoScope displays an example waveform.
4. PicoScope configures itself with the necessary settings.
5. Start the oscilloscope running, either using the red and green [Stop and Go buttons](#), or (if you are using the default [keyboard map](#)) by pressing the space bar.

## 5.7 Help menu

Location: [Menu bar](#) > **Help**

Purpose: gives access to the PicoScope 6 User's Guide and related information



<b>User's Guide</b>	The document you are reading now. This is the most complete source of information on PicoScope.
<b>Online Documentation</b>	Find manuals, data sheets, training guides and more for Pico Technology products
<b>Online Forum</b>	Request technical support and discuss questions with fellow PicoScope users. New software features are sometimes announced here before they make it into the <b>User's Guide</b> .
<b>Send Feedback</b>	Opens the <a href="#">Send Feedback dialog</a> . You can use this to suggest improvements and report software bugs. You can also tell us about errors in documentation such as user's guides, and about any translation errors in the software or the documentation.
<b>Check for Updates</b>	Check online for later versions of PicoScope. If you have both stable and beta versions installed, this can check for updates to both of them. You can configure the settings for automatic update checking on the <a href="#">Updates page</a> of the <a href="#">Preferences dialog</a> .
<b>About PicoScope 6</b>	Displays useful information such as the model and serial number of your scope, and software and driver version numbers.

## 5.7.1 Send Feedback dialog

Location: [Help](#) > **Send Feedback**

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

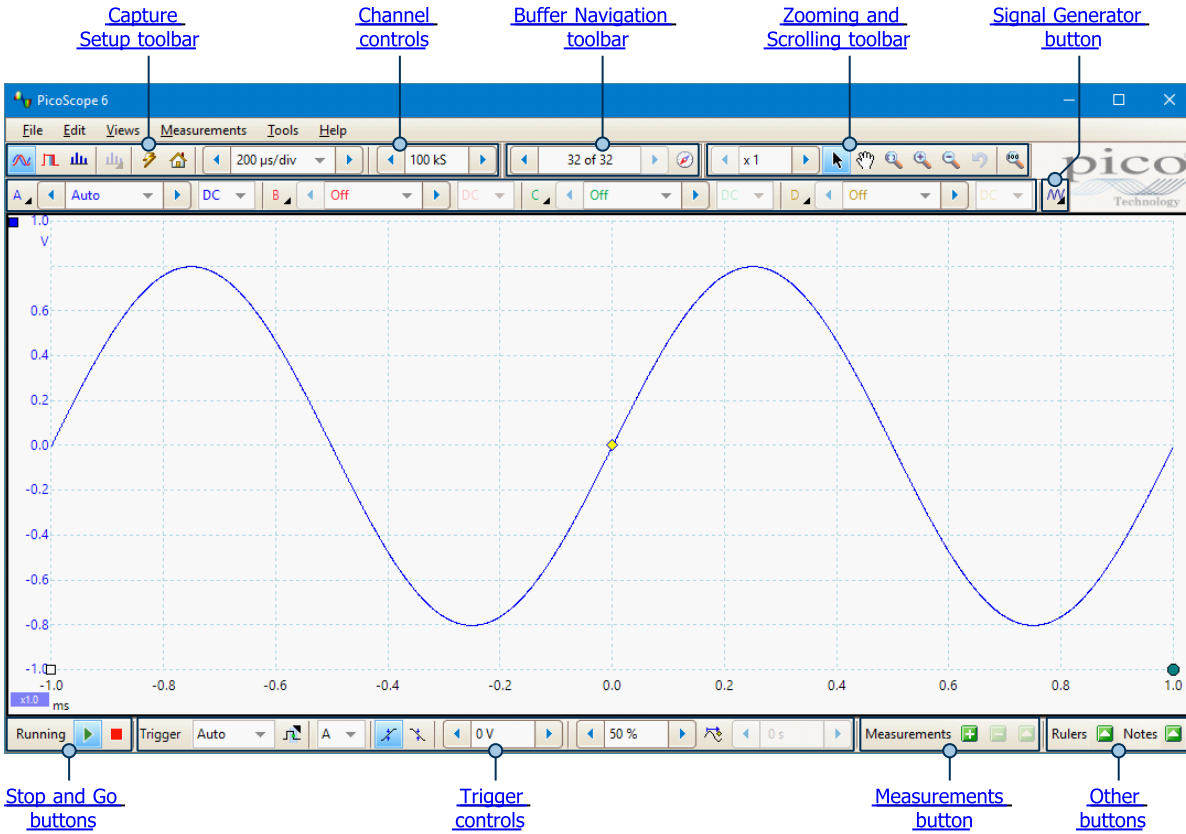
The **Send Feedback** command opens a web page (requires an internet connection) through which you can let us know about your experience using PicoScope 6.

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.

# 6 Toolbars and buttons

A toolbar is a collection of buttons and controls with related functions. PicoScope 6 includes the following toolbars and buttons:



## 6.1 Capture Setup toolbar

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

### Scope Mode

In [scope mode](#), the toolbar looks like this:



See below for different versions of the toolbar in [spectrum mode](#) and [persistence mode](#).



**Scope Mode.** Sets up PicoScope to operate as an [oscilloscope](#). Use the **Auto Setup button** to optimize the settings. If you wish, you can add a secondary [spectrum view](#) from the context menu (by right-clicking on the scope view).



**Persistence Mode.** Toggles [persistence mode](#), which allows old traces to remain on the screen in faded colors while new traces are drawn on top in brighter colors. The use of colors is controlled by the **Persistence Options** dialog. PicoScope will remember any views that were open, so that you can return to them by clicking the **Persistence Mode** button again.





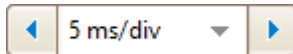
**Spectrum Mode.** Sets up PicoScope to operate in [spectrum analyzer mode](#). Use the **Auto Setup** button to optimize the settings. If you wish, you can add a secondary [scope view](#) from the context menu (by right-clicking on the scope view).



**Auto Setup.** Searches for a signal on one of the enabled input channels, then sets up the collection time and signal range to display the signal correctly.

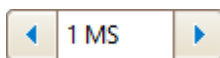


**Home.** Restores PicoScope to its default settings. Equivalent to the [File > Startup Settings > Load Startup Settings](#) command.



**Collection Time.** Sets the time represented by the horizontal axis when the [horizontal zoom control](#) is set to x1. The options available depend on the type of [scope device](#) you are using and, for some devices, on the [number and combination of channels enabled](#) and the selected [trigger mode](#). PicoScope can display the collection time as either times per division (as shown here) or a total collection time: you can choose between them using the **Collection Time Units** control on the [General page](#) of the [Preferences dialog](#).

Choosing a collection time equal to or longer than PicoScope's **Slow Sampling Transition** setting (200 ms/div by default) causes the scope to switch to a different mode of data transfer. The internal details of this are taken care of by PicoScope, but the slow mode limits the sampling rate to the 'streaming mode sampling rate' specified in the scope device's data sheet. The Slow Sampling Transition setting can be changed on the [Sampling page](#) of the [Preferences dialog](#).



**Number of Samples.** Sets the maximum number of samples that will be captured for each channel. If this is larger than the number of pixels across the scope view, you can zoom in to see more detail. The actual number of samples captured is displayed on the [Properties sheet](#), and may be different from the number requested here, depending on which collection time is selected and which scope device is in use. To capture one waveform that occupies the entire buffer memory, first set the [Trigger Mode](#) control to **Single**.

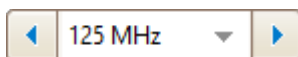
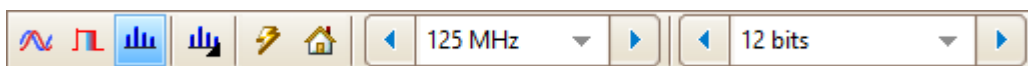


**Hardware Resolution** ([Flexible resolution oscilloscopes](#) only). Sets the number of hardware bits used for sampling. The range of options depends on the number of channels enabled and the sampling rate selected.

**Automatic Resolution** chooses the highest resolution compatible with the currently selected sampling rate and capture size. To find out what resolution is actually in use, view the **Effective Resolution** value in the [Properties Sheet](#). Resolution can be further increased by software filtering: see [resolution enhancement](#).

## Spectrum Mode

In [spectrum mode](#), the **Capture Setup toolbar** looks like this:



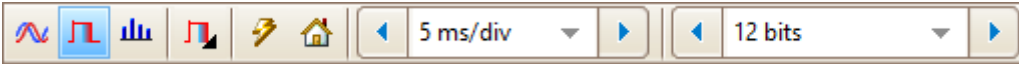
**Spectrum Range control.** Sets the frequency range across the horizontal axis of the spectrum analyzer when the **horizontal zoom control** is set to x1.



**Spectrum Options.** Appears if a [spectrum view](#) is open, regardless of whether [scope mode](#) or [spectrum mode](#) is selected. It opens the [Spectrum Options dialog](#).

### Persistence Mode

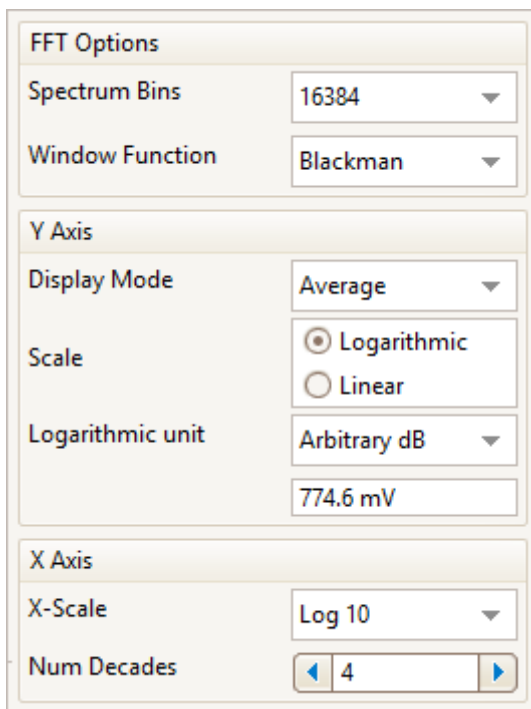
In [persistence mode](#), the **Capture Setup toolbar** looks like this:



**Persistence Options.** Opens the [Persistence Options dialog](#), which controls several parameters affecting the way that PicoScope represents old and new data in persistence mode.

## 6.1.1 Spectrum Options dialog

This dialog appears when you click the **Spectrum Options** button in the [Capture Setup toolbar](#). It is available only when a [spectrum view](#) is open. It contains controls that determine how PicoScope converts the source waveform in the current scope view to a spectrum view.



### FFT Options

#### Spectrum Bins

The number of frequency bins into which the spectrum is divided. This control sets the maximum number of frequency bins, which the software may or may not be able to provide depending on other settings. The main constraint is that the number of bins cannot greatly exceed half the number of samples in the source waveform.

If the source waveform contains fewer samples than required (that is, fewer than twice the number of frequency bins), then PicoScope zero-pads the waveform up to the next power of two. For example, if the scope view contains 10 000 samples, and you set Spectrum Bins to 16 384, then PicoScope zero-pads the waveform to

16 384 samples, which is the nearest power of two above 10 000. It then uses these 16 384 samples to provide 8192 frequency bins, not the 16 384 requested.

If the source waveform contains more samples than required, then PicoScope uses as many samples as necessary, starting from the beginning of the waveform buffer. For example, if the source waveform contains 100 000 samples and you request 16 384 frequency bins, PicoScope needs only  $2 \times 16\,384 = 32\,768$  samples, so it uses the first 32 768 samples from the waveform buffer and ignores the rest. The amount of data actually used is displayed as the **Time Gate** setting in the [Properties sheet](#).

#### Window Function

Allows you to choose one of the standard window functions to reduce the effect of operating on a time-limited waveform. See [Spectrum window functions](#) for a list of available functions.

#### Y Axis

##### Display Mode

You can choose **Magnitude**, **Average** or **Peak Hold**.

**Magnitude:** the spectrum view shows the frequency spectrum of the last waveform captured, whether live or stored in the [waveform buffer](#).

**Average:** the spectrum view shows a rolling average of spectra calculated from all the waveforms in the [waveform buffer](#). This has the effect of reducing the noise visible in the spectrum view. To clear the averaged data, click [Stop](#) and then [Go](#), or change from **Average** mode to **Magnitude** mode.

**Peak Hold:** the spectrum view shows a rolling maximum of the spectra calculated from all the waveforms in the buffer. In this mode, the amplitude of any frequency band in the spectrum view will either stay the same or increase, but never decrease, over time. To clear the peak hold data, click [Stop](#) and then [Go](#), or change from **Peak Hold** mode to **Magnitude** mode.

Note: when you switch to **Average** or **Peak Hold** mode, there may be a noticeable delay while PicoScope processes the entire contents of the waveform buffer, which may contain many waveforms, to build up the initial display.

##### Scale

Specifies the labeling and scaling of the vertical (signal) axis. This can be one of the following:

**Linear:** The vertical axis is scaled in volts RMS

**Logarithmic:** The vertical axis is scaled in decibels, with reference quantities selectable in the **Logarithmic unit** control below.

##### Logarithmic unit

Specifies the reference quantity used on the logarithmic scale. This option is only available when you have already selected **Logarithmic** in the **Scale** control.

**dBV:** Reference level is 1 V

**dBu:** Reference level is 1 mW with a load resistance of 600  $\Omega$ . This corresponds to a voltage of about 775 mV.

**dBm:** Reference level is 1 mW into the specified load impedance. You can enter the load impedance in the box below the **Logarithmic unit** control.

**Arbitrary dB:** Reference level is an arbitrary voltage, which you can specify in the box below the **Logarithmic unit** control.

**X Axis****X Scale**

Specifies the scale of the frequency axis. You can choose between **Linear** and **Log 10**.


**Linear:** The axis is scaled in equal intervals from DC to the frequency set by the [Spectrum Range](#) control.

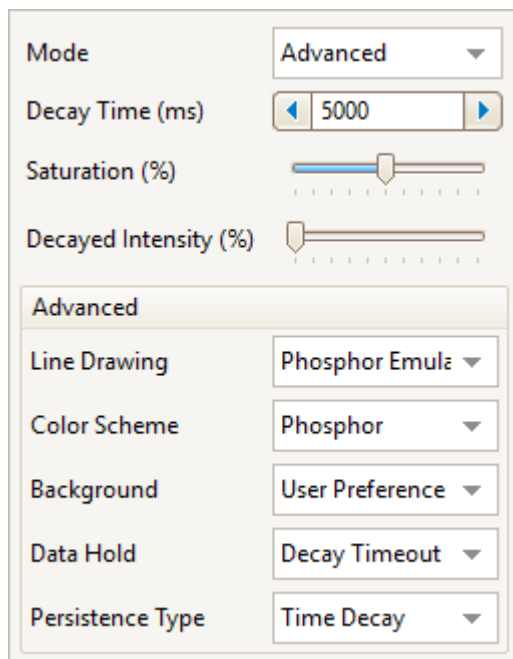
**Log 10:** The axis is scaled in decades ending at the specified **Spectrum Range** frequency, and beginning at a number of decades below that as specified by the **Number of Decades** control.

**Num Decades**

Specifies the number of decades into which the frequency axis is divided when **X Scale** is set to **Log 10**.

## 6.1.2 Persistence Options dialog

This dialog appears when you click the **Persistence Options** button  in the [Capture Setup toolbar](#). It is available only when [persistence mode](#) is selected. It controls the colors and fading algorithm used to distinguish new or frequent data from old or intermittent data in the persistence view.

**Mode**

**Digital Color:** This mode uses a range of colors to indicate the frequency of waveform data. Red is used for the most frequent data, and less frequent data is represented successively by yellow and blue.

**Analog Intensity:** This mode uses color intensity to indicate the age of waveform data. The latest data is drawn at full intensity in the selected color for that channel, with older data being represented by paler shades of the same color.

**Fast:** A mode with simplified display options to enable the fastest possible update rate. Only available on scope devices that support [rapid triggering](#).

**Advanced:** This mode activates the **Advanced** options at the bottom of the dialog, allowing you to customize the persistence mode display.

<b>Decay Time</b>	The time, in milliseconds, taken for waveform data to fade from red to blue or from maximum to minimum analog intensity. The longer the decay time, the longer the older waveforms will remain on the screen.
<b>Saturation</b>	The intensity or color with which new waveforms are drawn.
<b>Decayed Intensity</b>	The intensity or color to which the oldest waveforms decay when the decay time expires. If the decayed intensity is zero, then older waveforms will be completely erased from the display after the decay time. For non-zero values of decayed intensity, old waveforms will remain indefinitely on the screen at that intensity unless overwritten by new ones.

### Advanced

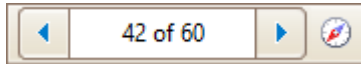
These options are available when **Mode** is set to **Advanced**.

<b>Line Drawing</b>	<p>The type of line drawn between samples that are adjacent in time.</p> <p><b>Phosphor Emulation:</b> Joins each pair of sample points with a line whose intensity varies inversely with the slew rate</p> <p><b>Constant Density:</b> Joins each pair of sample points with a line of uniform color</p> <p><b>Scatter:</b> Draws sample points as unconnected dots</p>
<b>Color Scheme</b>	<p><b>Phosphor:</b> Uses a single hue for each channel, with varying intensity.</p> <p><b>Color:</b> Uses a color from red to blue to represent the age of each waveform.</p>
<b>Background</b>	<p><b>Black:</b> Overrides the <a href="#">Colors page</a> of the <a href="#">Preferences dialog</a>.</p> <p><b>White:</b> Overrides the <a href="#">Colors page</a> of the <a href="#">Preferences dialog</a>. This is the default in Analog Intensity and Digital Color modes.</p> <p><b>User Preference:</b> Sets the background color to the preference set on the <a href="#">Colors page</a> of the <a href="#">Preferences dialog</a>.</p>
<b>Data Hold</b>	<p>This option is enabled only when <b>Persistence Type</b> (see below) is set to <b>Time Decay</b>.</p> <p><b>Decay Timeout:</b> Old waveforms fade until they reach <b>Decayed Intensity</b> and then disappear.</p> <p><b>Infinite:</b> Old waveforms fade until they reach <b>Decayed Intensity</b> and then remain indefinitely unless overwritten by new waveforms.</p>
<b>Persistence Type</b>	<p><b>Fast:</b> The <b>Line Drawing</b>, <b>Decay Time</b>, <b>Saturation</b> and <b>Decayed Intensity</b> options are disabled to maximize the update rate. In addition, <a href="#">resolution enhancement</a>, <a href="#">lowpass filtering</a>, <a href="#">sin(x)/x interpolation</a> and nonlinear <a href="#">custom probes</a> are unavailable in this mode. This mode requires a scope that supports <a href="#">rapid triggering</a> (see <a href="#">Device feature table</a>).</p> <p><b>Time Decay:</b> Points on the display are drawn at full intensity when hit by a waveform, and are then allowed to decay to <b>Decayed Intensity</b>. The behavior after this depends on the <b>Data Hold</b> setting (see above).</p>

**Frequency:** Points on the display are drawn with a color or intensity that depends on the frequency with which they are hit by waveforms.

## 6.2 Buffer Navigation toolbar

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



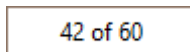
### What is the waveform buffer?

Depending on the settings you have chosen, PicoScope can store multiple waveforms in its waveform buffer. When you click the [Go](#) button or change a [capture setting](#), PicoScope clears the buffer and then adds a new waveform to it each time the scope device captures data. This continues until the buffer is full or you click the [Stop](#) button. You can set the maximum number of waveforms in the buffer to any number between 1 and 10 000 using the [General page](#) of the [Preferences dialog](#).

You can review the waveforms stored in the buffer using these buttons:



**Previous** button. Displays the previous waveform in the buffer.



**Waveform buffer index.** Shows which waveform is currently being displayed and how many waveforms are in the buffer. Edit the number in the box and press **Enter** to make PicoScope jump to the specified waveform.



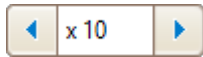
**Next** button. Displays the next waveform in the buffer.



**Buffer Overview** button. Opens the [Buffer Overview](#) window, for quick selection of buffer waveforms.

## 6.3 Zooming and Scrolling toolbar

The **Zooming and Scrolling toolbar** allows you to examine a [scope view](#) or [spectrum view](#) in greater detail and move about the zoomed view. Each button has a keyboard shortcut, as listed below.



**Horizontal zoom control.** Zooms the view, in the horizontal direction only, by the specified amount. Click the and buttons to adjust the zoom factor, or the button to reset.



**Ctrl+S**  
or  
**Esc**

**Normal Selection tool.** Restores the mouse pointer to its normal appearance. You can use this pointer to click buttons, drag [rulers](#) and operate any other controls in the PicoScope window.



**Ctrl+D**

**Hand tool.** Turns the mouse pointer into a hand tool () that you can use to click and drag the view to pan it vertically and horizontally when you are zoomed in. You can also pan using the scroll bars. Press the **Esc** key to return to the **Normal Selection tool**.



**Ctrl+M**

**Marquee Zoom tool.** This button turns the mouse pointer into a marquee zoom tool: . Use it to draw a box (called a marquee) on the view and PicoScope will magnify that box to fill the view. Scroll bars will appear, which you can drag to pan around in the view, or you can pan by using the **Hand tool** (see above). Zooming in also opens the [Zoom Overview](#) window. Press the **Esc** key to return to the **Normal Selection tool**.

If you point to the time axis, the pointer changes into the horizontal marquee zoom tool () which restricts zooming to the horizontal axis. This lets you zoom in by an arbitrary amount without disturbing the vertical zoom factor.

Holding down the **Ctrl** key while dragging also restricts zooming to the horizontal axis.



**Ctrl+I**

**Zoom-in tool.** Turns the mouse pointer into a zoom-in tool: . Click on the view with this tool to zoom in to the specified location. Zooming in also opens the [Zoom Overview](#) window.

If you point to the time axis, the pointer changes into the horizontal zoom-in tool () which restricts zooming to the horizontal axis. This lets you zoom in without disturbing the vertical zoom factor.


Holding down the **Ctrl** key while dragging also restricts zooming to the horizontal axis.

Holding down the **Shift** key while dragging changes the zoom mode to 'zoom out'.



**Ctrl+O**

**Zoom-out tool.** Turns the mouse pointer into a zoom-out tool: . Click on the view with this tool to zoom out around the specified location.

If you point to the time axis, the pointer changes into the horizontal zoom-out tool () , which restricts zooming to the horizontal axis. This lets you zoom out without disturbing the vertical zoom factor.

Holding down the **Ctrl** key while dragging also restricts zooming to the horizontal axis.

Holding down the **Shift** key while dragging changes the zoom mode to 'zoom in'.



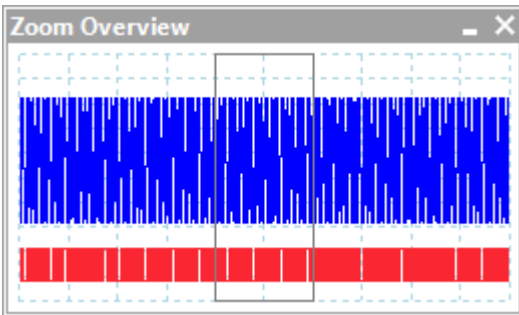
**Undo zoom.** Returns the view to the previous zoom and pan settings.



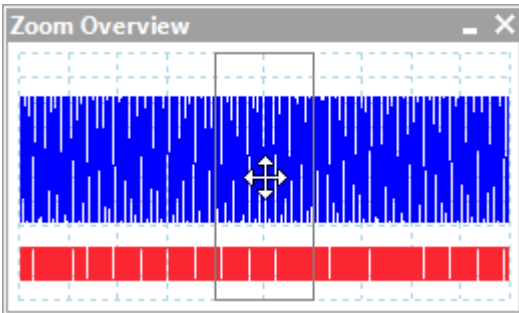
**Ctrl+U** **Zoom to full view.** Resets the view to normal size. The view will no longer have scroll bars, and panning will no longer be possible.

### 6.3.1 Zoom Overview window

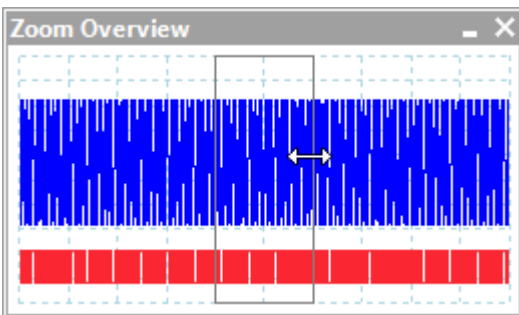
Whenever you zoom in using the [Zooming and Scrolling toolbar](#), the **Zoom Overview** window should appear. If the Zoom Overview window does not appear, you may have switched it off: make sure you have checked the **Zoom Overview** box on the [Options page](#) of the [Preferences dialog](#).



The **Zoom Overview** shows the full waveforms on all enabled channels. The rectangle indicates the area that is visible in the current view.



You can move around the waveform by dragging the rectangle.



You can also adjust the zoom factor by dragging the edges of the rectangle to resize it.



**Minimize** button: reduce the **Zoom Overview** window in size without affecting the zoom settings.





**Close button:** close the **Zoom Overview** window and return the zoom factor to 100%.

## 6.4 Channels toolbar (PicoScope devices)

The Channels toolbar contains all the [Channel controls](#) and the [Signal Generator button](#).

If you are using PicoScope 6 with a data logger, please refer to [PicoLog 1000 Series Channels toolbar](#) and [USB DrDAQ Channels toolbar](#).

### 6.4.1 Channel controls (PicoScope devices)

The **Channel controls** allow you to adjust the settings for each input [channel](#). The screenshot below shows the toolbar for a two-channel [MSO scope device](#), but different scope devices may have different numbers of channels. Demo device settings are also displayed here.



If you are using PicoScope 6 with a data logger, please refer to [PicoLog 1000 Series Channels toolbar](#) and [USB DrDAQ Channels toolbar](#).


Each channel has its own set of buttons:



**Channel Options button.** Opens the [Channel Options menu](#) with options for [probes](#), [resolution enhancement](#), [scaling](#) and filtering.



**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. If you select **Auto**, PicoScope will continually adjust the vertical scale so that the height of the waveform fills as much of the view as possible.



**Coupling control.** Sets up the input circuitry.

**AC coupling:** rejects frequencies below about 1 Hz.

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

**50Ω DC:** low-impedance option (see [Device feature table](#)).

**Accelerometer:** switches on the current source output for [IEPE](#)-enabled scopes such as the PicoScope 4224 IEPE. The User's Guide for the oscilloscope has details of the [IEPE](#) channel specifications.

**Frequency:** enable the built-in [frequency counter](#). Only one channel at a time can be operated in this mode. Available only if your scope device has hardware support for this feature: see [Device feature table](#). Not available in [Demo mode](#).




**ConnectDetect® button.** Automotive oscilloscopes only.





**Digital Channels & Groups** button. [MSOs](#) only.

### 6.4.1.1 Channel Options menu

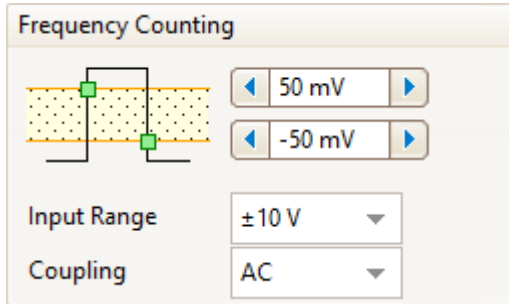
The **Channel Options** menu appears when you click the

**Channel Options** button (for example: ) on the [Channels toolbar](#). There is one Channel Options menu for each channel.



**Probe list.** Indicates the probe currently in use and allows you to select a different one. Use it to tell PicoScope what type of probe is connected to the channel. By default, the probe is assumed to be x1, which means that a 1 V signal at the input to the probe will appear as 1 V on the display. Click  to select a probe from the list or click  to open the [Custom Probes dialog](#) and edit your library of custom probes.

Note: When selecting a Flow Meter probe, you will have to enter a K Factor calibration value upon first use. You can usually find this value on the flow meter itself.

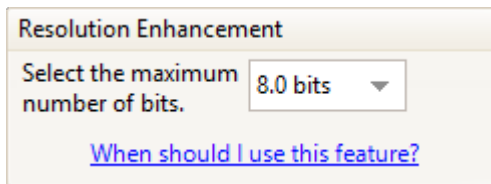


**Frequency counting.** Graphs the frequency of a user-defined pulse. For availability of the built-in frequency counter, see the [Device feature table](#).

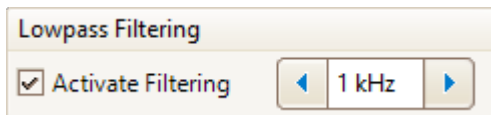
**Upper and lower thresholds:** Voltage levels that the signal must fall and rise through respectively to register as a valid pulse. The two thresholds provide hysteresis and eliminate noise. If this is not a concern, you can set both thresholds to the same value.

**Input range:** see [Input Range control](#)

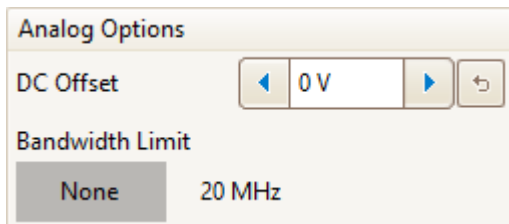
**Coupling:** see [Coupling control](#)



**Resolution enhancement.** Allows you to increase the effective resolution of your scope device using [Resolution enhancement](#). The number in this box is a target value that the software will attempt to use whenever possible.



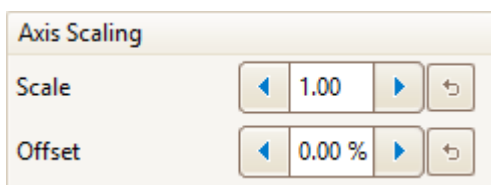
**Lowpass Filtering.** An independent digital [lowpass filter](#) for each input channel, with programmable cut-off frequency. This can be useful for removing noise from your signal to make more accurate measurements. For availability, see the [Device feature table](#).



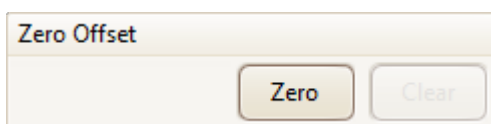
**Analog Options.** Options that can be applied to the oscilloscope input hardware, if the oscilloscope hardware supports them.

**DC Offset:** an offset voltage added to the analog input before digitization. For availability, see the [Device feature table](#).

**Bandwidth Limit:** a fixed-frequency single-pole analog filter. This can be useful for rejecting noise and harmonics that would otherwise cause aliasing. This advanced feature needs to be enabled on the [Options page](#) of the [Preferences dialog](#) before use. For availability, see the [Device feature table](#).



**Axis Scaling.** These are the [axis scaling controls](#) that allow you to set the scale and offset for each vertical axis individually.



**Zero offset.** Digitally removes any offset from the input channel. Before starting this operation, remove any input signal from the selected channel and short-circuit the input. Click **Zero** to begin the adjustment. Click **Clear** to restore the input to its uncorrected state.

### 6.4.1.1.1 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 the **Channel Options** button  in the [Channel controls](#) to open the [Channel Options menu](#).
- 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 5204 (resolution = 8 bits). You have selected an effective resolution of 9.5 bits. The resolution enhancement is therefore:

$$e = 9.5 - 8.0 = 1.5 \text{ bits}$$

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

$$n = 8 \text{ samples}$$

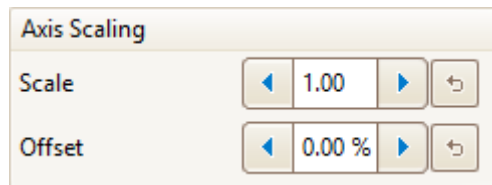
This number gives a guide to what sort of filtering effect the resolution enhancement will have on the signal. The best way of seeing the actual lowpass filter effect is to add a spectrum view and look at the shape of the noise floor (try dragging the y-axis upwards to see the noise more clearly).

### Related topics

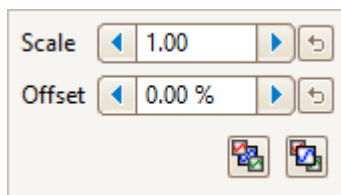
See [Hardware resolution](#) (applies to flexible resolution oscilloscopes only).

#### 6.4.1.1.2 Axis scaling controls

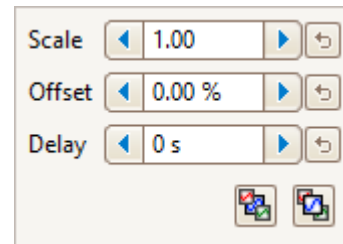
The **axis scaling controls** let you change the scale and offset of each vertical axis individually. If the axis belongs to a [reference waveform](#), you can also adjust its delay relative to the live waveforms. You can use the axis scaling controls in any [capture mode](#).



Controls for a live waveform (Channel Options menu)




Controls for a live waveform (scaling button)




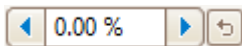
Controls for a reference waveform (scaling button)


There are two ways to open the axis scaling control:

- For any channel displayed in a [view](#): click the colored scaling button  at the bottom of the vertical axis
- For any input channel: click the [Channel Options button](#) in the [Channels toolbar](#)




**Scale control.** 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. Click the reset button () to return to a scale of 1.0. The scaling button always shows the selected scale.



**Offset control.** 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. Click the reset button () to return to an offset of 0.00%. You can also adjust the offset by dragging the vertical axis up or down the display.



**Delay control (for reference waveforms only).** Increase to move the waveform to the left relative to the timing reference point, decrease to move it to the right. Click the reset button () to return to a delay of 0 s.

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.




**Send to back.** Draws the channel behind all others. Use if the channel is obscuring another channel of interest.

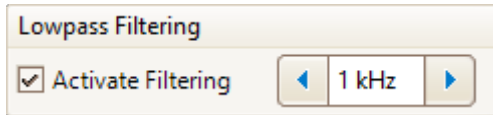


**Bring to front.** Draws the channel in front of all others. Use if the channel is hidden behind another.

### 6.4.1.1.3 Lowpass filtering

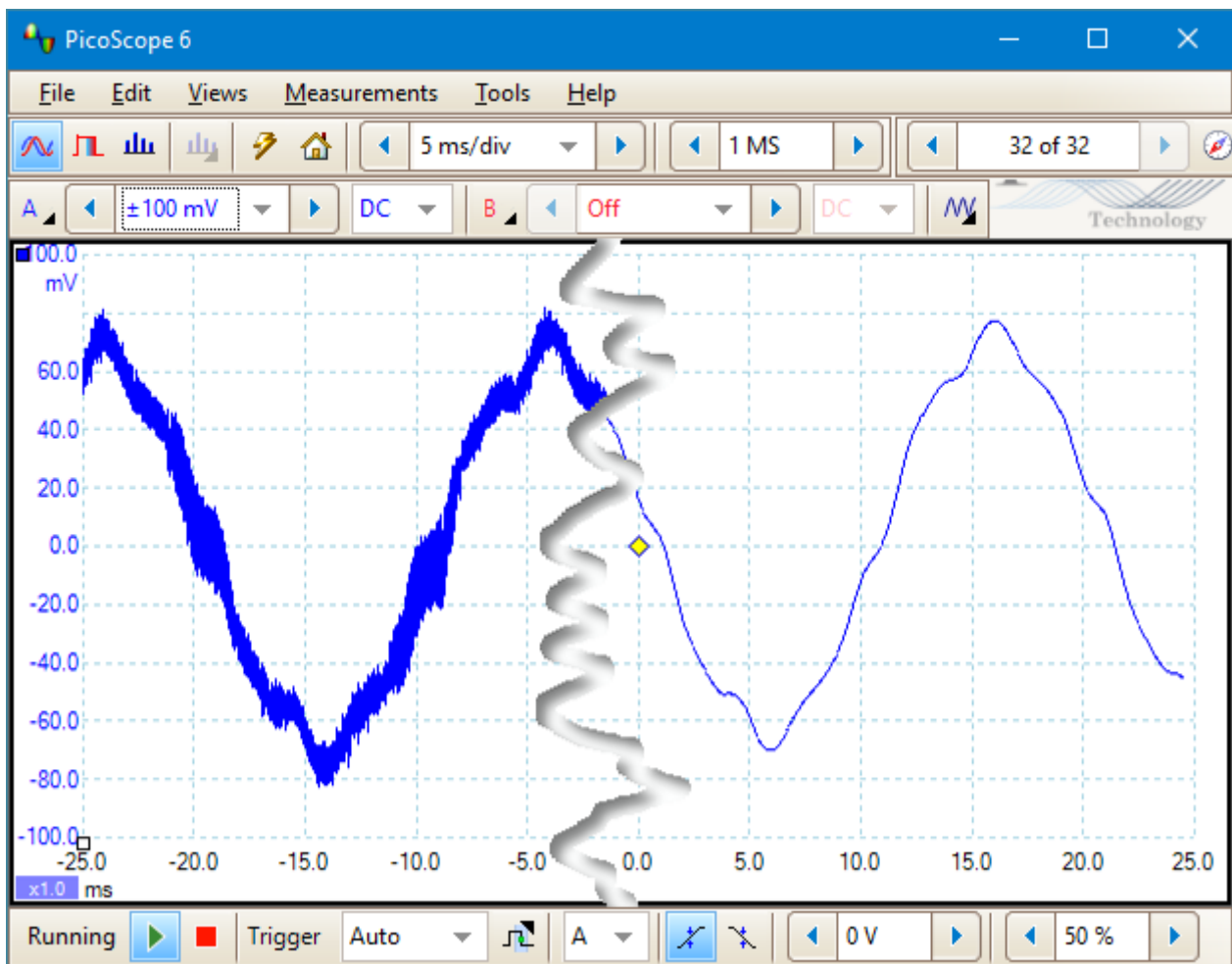
The **lowpass filtering** feature can reject high frequencies from any selected input channel. The filtering control is found in the [Advanced Channel Options dialog](#), which is opened by clicking the **Channel Options**

button (  ) for the relevant channel on the [Channels toolbar](#). The control determines the cut-off frequency of the filter, which must be below half the sampling rate shown in the [Properties sheet](#).



For availability, see the [Device feature table](#).

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.

### Filter details

The lowpass filtering algorithm is chosen according to the ratio of the selected cut-off frequency ( $f_c$ ) to the sampling rate ( $f_s$ ), as follows:

$f_c \div f_s$	Filter type	Description
0.0 to 0.1	Moving average	A moving average filter is used for low cut-off frequencies. The length of the filter is adjusted to achieve the selected cut-off frequency, which is defined as the first minimum in the frequency response. There is significant signal leakage above the cut-off frequency. This filter changes a vertical edge into a linear slope.
0.1 to < 0.5	FIR	A finite impulse response filter is used for medium to high cut-off frequencies. This has a monotonic roll-off above the cut-off frequency and therefore suffers from less leakage than the moving average filter.

You can force PicoScope to use one or other of the filter types by adjusting the **Samples** control in the [Capture Setup toolbar](#) to make the ratio  $f_c/f_s$  fall into one of the two ranges shown in the table. As the table shows, the cut-off frequency must be below half the sampling frequency.

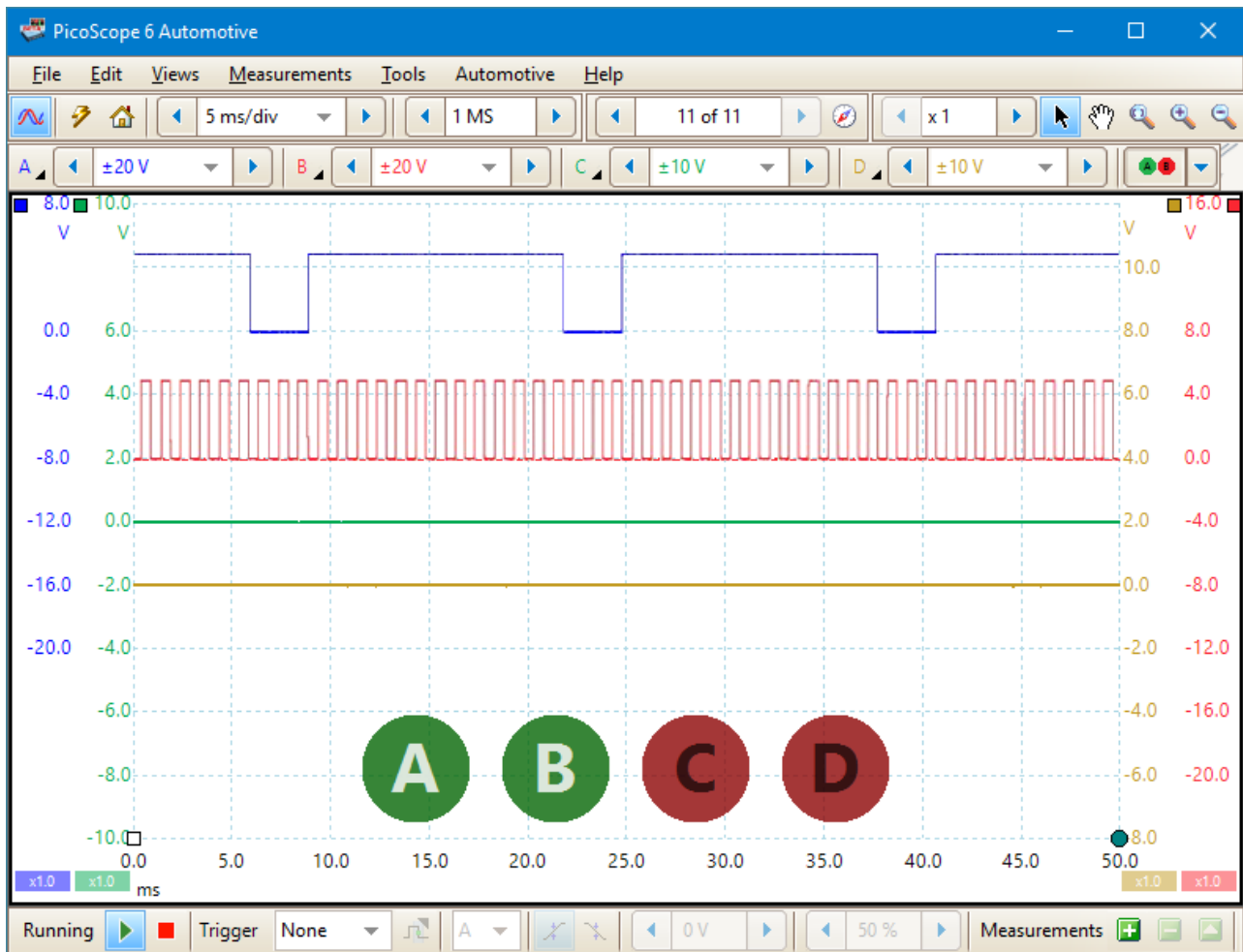


### 6.4.1.2 ConnectDetect®


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

**Location:** [Channels toolbar](#) (automotive oscilloscopes only)

**Applicability:** PicoScope 4225 and 4425 automotive scopes only  
DC coupling mode only



Every channel input on the PicoScope 4225 and 4425 scopes has an LED next to it. When ConnectDetect is activated for a channel, the LED for that channel is either green, to indicate that the test probe is directly connected across a component, or red to indicate that it is not. You can select which channels

ConnectDetect is activated for by clicking the arrow on the ConnectDetect button: 

An icon representing the LED is also shown on the PicoScope screen, as in the example above, where ConnectDetect has been activated on all four channels, but only Channels A and B are properly connected.

**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.

### 6.4.1.3 Digital Channels & Groups button

Location: [Channels toolbar](#) (MSOs only)

Purpose: controls the settings for the digital inputs of a mixed signal oscilloscope ([MSO](#))

Digital on/off  Digital setup 



Switches the [digital view](#) on or off. If you have not yet set up any digital inputs, the [Select Digital Channels/Groups dialog](#) will open.

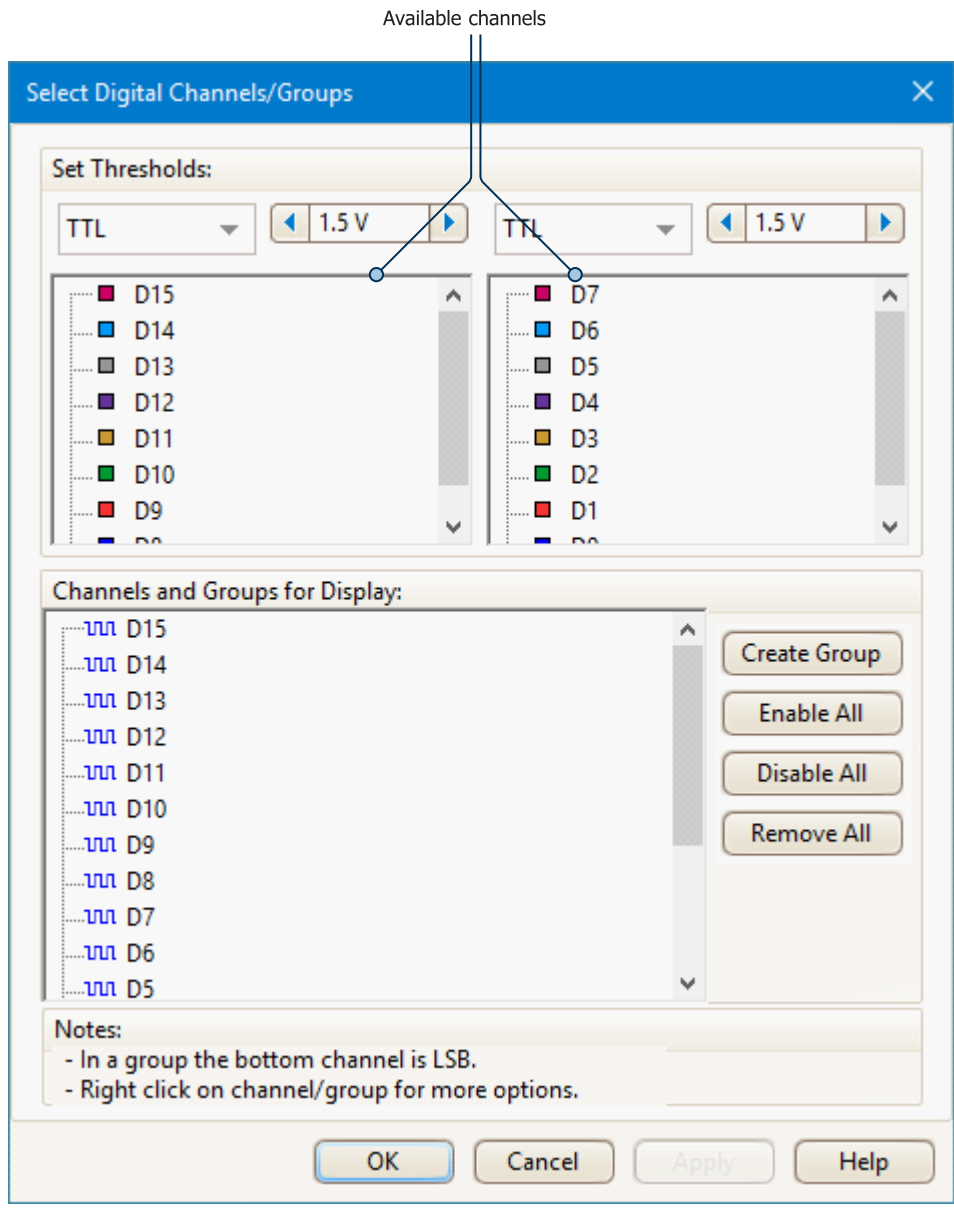


Opens the [Select Digital Channels/Groups dialog](#) for channel selection and options. Any digital inputs you activate here will remain active even when you switch the digital view off.

6.4.1.3.1 Select Digital Channels/Groups dialog

Location: [Digital Channels & Groups button](#)

Purpose: controls the digital inputs of an [MSO](#)



**Set Thresholds**

Choose the digital threshold voltage from the drop-down list, or select the **Custom** threshold and set your own voltage using the numeric entry control. The preset thresholds are:

TTL	1.5 V	LVC MOS 1.8 V	0.9 V
CMOS	2.5 V	LVC MOS 2.5 V	1.25 V
ECL	-1.3 V	LVC MOS 3.3 V	1.65 V
PECL	3.7 V	LVDS	100 mV
LVPECL	2 V	0V Differential	0 V
LVC MOS 1.5 V	750 mV		


Each port has its own independent threshold. Port 0 contains channels D7...D0 and Port 1 contains channels D15...D8.


### Available Channels

This section lists the available digital input channels. They will not be displayed unless you add them to the **Channels and Groups for Display** section of the dialog. Click and drag individual channels into the Channels and Groups for Display section, select a range of channels and drag them all at once, or double-click a channel to add it directly.

### Channels and Groups for Display

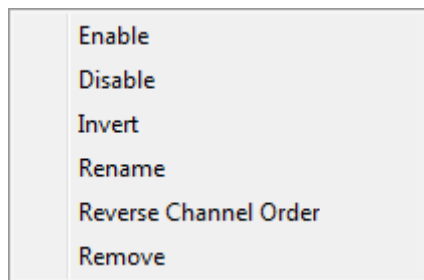
This section lists the digital channels that have been selected for display. Any groups of channels that you have defined will also be listed here.

 indicates a digital channel.

 indicates a group of digital channels. By default, channels added to a group are placed with the most significant bit at the top of the list.


Click the **Create Group** button to add a new group. This adds a new group to the Channels and Groups for Display section, and you can add channels to it by dragging them on top of it: you can do this from the available channels or from the Channels and Groups for Display section.

To rename a channel or group, click the name and type. For other operations, right-click the channel or group for a menu of actions:



<b>Enable</b>	Show the channel. All channels in the list are enabled by default.
<b>Disable</b>	Hide the channel from the display
<b>Invert</b>	Reverse the polarity of the channel. Useful for active-low signals.
<b>Rename</b>	Enter a new name for the channel
<b>Reverse Channel Order</b>	Groups only. Reverse the order of the channels in the group.
<b>Remove</b>	Remove the channel from the list

## 6.4.2 Signal Generator button

The **Signal Generator** button  allows you to set up your scope's test signal generator, if it has one, or the demo signal settings if PicoScope is in [demo mode](#).

If your scope has a built-in signal generator, clicking the Signal Generator button opens the [Signal Generator dialog](#).

If PicoScope is in [demo mode](#) then clicking the Signal Generator button opens the [Demo Signals menu](#).

### 6.4.2.1 Signal Generator dialog (PicoScope devices)

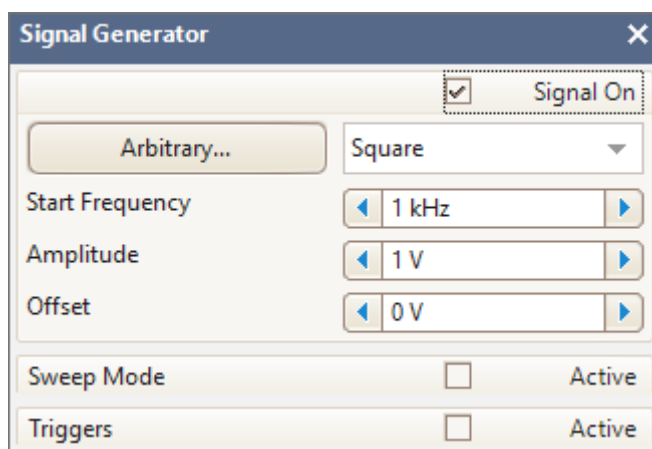
Location: [Channels toolbar](#) > [Signal Generator button](#)

Purpose: controls the [scope device's](#) built-in signal generator

Not all scope devices have a signal generator, and those that do have a varying range of controls in the signal generator dialog. See the [device feature table](#) for details.

The Signal Generator dialog has three sections, the [Basic controls](#), the [Sweep controls](#) and the [Signal generator triggers](#).

#### 6.4.2.1.1 Basic controls



**Signal On** Check this box to enable the signal generator

**Wave 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.

**Start 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.
- Sweep Mode** Check this box to activate the [Sweep controls](#)
- Triggers** Check this box to enable [Signal generator triggers](#)

#### 6.4.2.1.2 Sweep controls

The signal generator normally generates a fixed frequency set by the **Start Frequency** control. In sweep mode, it generates a frequency that varies between two specified limits.

- Active** Check this box to enable sweep mode
- Sweep Type** Specifies the direction in which the frequency is swept
- Stop Frequency** Specifies the frequency at which the generator stops increasing the frequency
- Frequency Increment** Specifies the amount by which the generator increases or decreases the frequency every **Increment Time Interval**
- Increment Time Interval** Sets the time interval to elapse between **Frequency Increments**

#### 6.4.2.1.3 Signal generator triggers

Usually, the signal generator of your oscilloscope runs continually. However, [some PicoScope PC oscilloscopes](#) allow you to trigger the signal generator, so that it waits for a specified event before generating an output, and will only generate a predetermined number of cycles of the signal.

- Active** Check this box to enable signal generator triggering

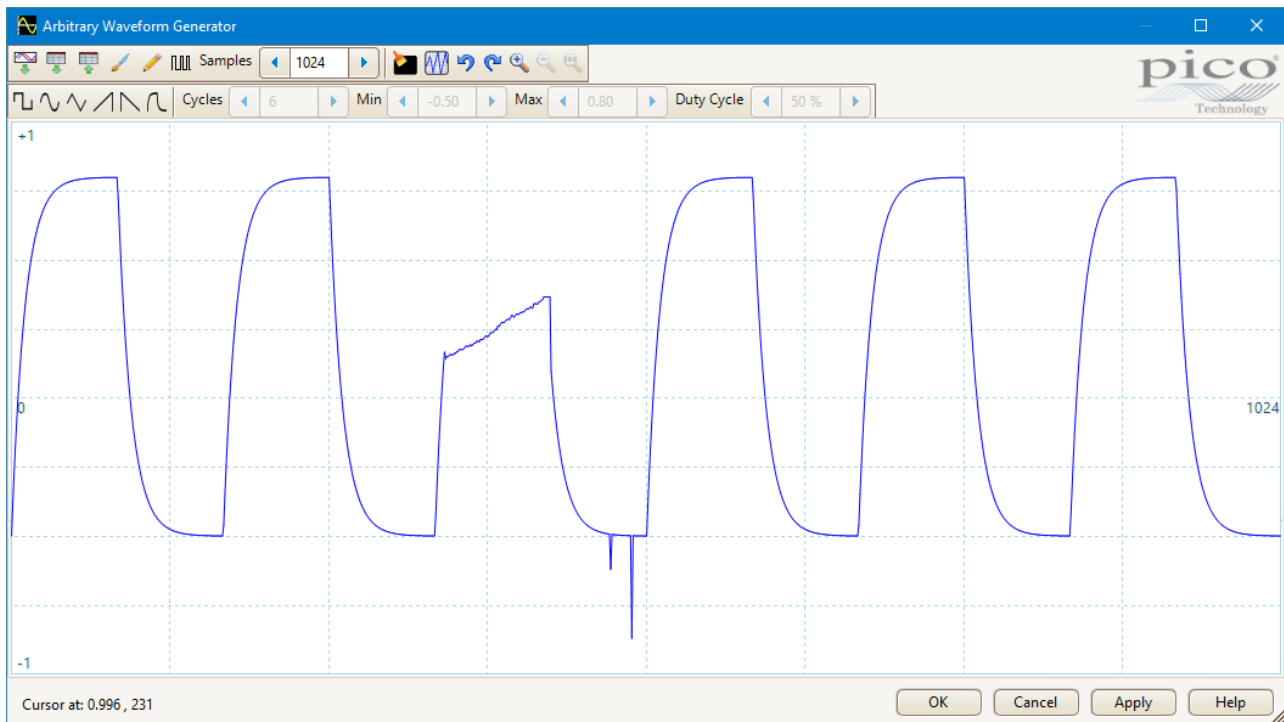
<b>Trigger Source</b>	<p>Allows you to specify what signal will trigger the signal generator.</p> <p><b>Scope:</b> The same condition that triggers the scope. This allows you to start the signal generator at a specific point on the PicoScope display, and to use an advanced trigger.</p> <p>Note: if you set the oscilloscope to trigger on the generated signal, then the signal generator may never trigger.</p> <p><b>Manual:</b> The <b>Trigger Now</b> button or an <a href="#">alarm action</a></p> <p><b>Ext Input:</b> The trigger input marked <b>EXT</b> (<a href="#">if fitted</a>) on the scope device</p> <p><b>Aux Input:</b> The trigger input marked <b>AUX</b> (<a href="#">if fitted</a>) on the scope device</p>
<b>Type</b>	<p>If the <b>Trigger Source</b> is <b>Ext</b> or <b>Aux</b>, you can select the type of trigger that the signal generator will use.</p> <p><b>Rising Edge:</b> The signal generator starts running when the trigger signal changes from low to high.</p> <p><b>Falling Edge:</b> The signal generator starts running when the trigger signal changes from high to low.</p> <p><b>Gate High:</b> The signal generator runs whenever the trigger signal is high.</p> <p><b>Gate Low:</b> The signal generator runs whenever the trigger signal is low.</p>
<b>Cycles Per Trigger</b>	<p>The number of cycles of the specified waveform to output each time the signal generator is triggered. If the trigger <b>Type</b> is <b>Gate High</b> or <b>Gate Low</b>, the generator will stop when the gate signal becomes inactive, even if the requested number of cycles has not been generated.</p>
<b>Threshold</b>	<p>Available only when the <b>Trigger Source</b> is <b>Ext Input</b>. Sets the voltage level used to distinguish between the high and low states of the trigger signal.</p>
<b>Manual Trigger</b>	<p>Available only when the <b>Trigger Source</b> is <b>Manual</b>. Click the <b>Trigger Now</b> button to trigger the signal generator to produce the specified number of cycles (or sweeps, if the signal generator is in <a href="#">sweep mode</a>).</p>

## 6.4.2.2 Arbitrary Waveform Generator window

Location: [Signal Generator dialog](#) > **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 dialog](#). 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.



Once the desired waveform appears in the window, click **OK** or **Apply** to start using it.

### Toolbar buttons



**Import from a channel.** Opens the [Import from Channel dialog](#), which lets you copy a waveform from the scope into the arbitrary waveform window.



**Import from CSV.** Displays an **Open** dialog to allow you to [import an arbitrary waveform from a CSV file](#).



**Export as CSV.** Displays a **Save As...** dialog to allow you to [export the arbitrary waveform as a CSV file](#).



**Smooth Drawing mode.** Allows you to draw or edit a waveform shape freehand, using the mouse.





**Line Drawing mode.** Enters straight line mode, in which you can click on the waveform to draw a straight line from the previous point. To start a new series of lines, click the button again.



**Bit Stream mode.** Draws a sequence of bits according to binary or hex data that you specify. The logic high and low levels are adjustable.



**Samples.** The number of samples in the arbitrary waveform. Each sample represents the signal value at a given instant in time, and the samples are equally spaced in time. For example, if there are 1024 samples and the [arbitrary waveform generator](#) is set to play back at 1 kHz, then each sample represents  $(1/1 \text{ kHz} \div 1024)$  or about 0.98 microseconds.



**Clear All.** Deletes the arbitrary waveform.



**Normalize.** Expands the waveform vertically about 0 V, until the highest peak value reaches either -1 or +1.



**Undo and Redo.** The **Undo button** reverses the last change made to the arbitrary waveform. The **Redo button** reverses the last action of the **Undo button**.



**Zoom tools.** To zoom the time axis in or out, click the + or - zoom button and then click on the waveform area. Click the **100%** button to restore the time axis to its original scale.

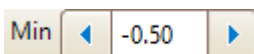
### Waveform settings



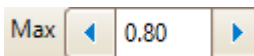
**Standard waveform shapes.** Draw a standard waveform with the settings specified in the numerical controls below the toolbar. The current waveform will be erased.



**Cycles.** The number of cycles to draw. This control is used in conjunction with the **Standard waveform shapes buttons**. Select one of the standard waveform shapes and then enter the number of cycles, and PicoScope will draw the requested number of cycles of the waveform.



**Minimum.** When one of the **Standard waveform shapes buttons** is pressed, this control sets the minimum signal level.



**Maximum.** When one of the **Standard waveform shapes buttons** is pressed, this control sets the maximum signal level.



**Duty cycle.** When a square, triangular or ramp waveform is selected using one of the **Standard waveform shapes buttons**, this control sets the duty cycle of the signal. A symmetrical square or triangular wave has a duty cycle of 50%. Reducing the duty cycle shortens the positive part of the cycle and lengthens the negative part, and increasing the duty cycle does the opposite.

### Other buttons

#### OK

Copies the waveform from the graphical editor into the arbitrary waveform generator and returns to the main [PicoScope](#) window.

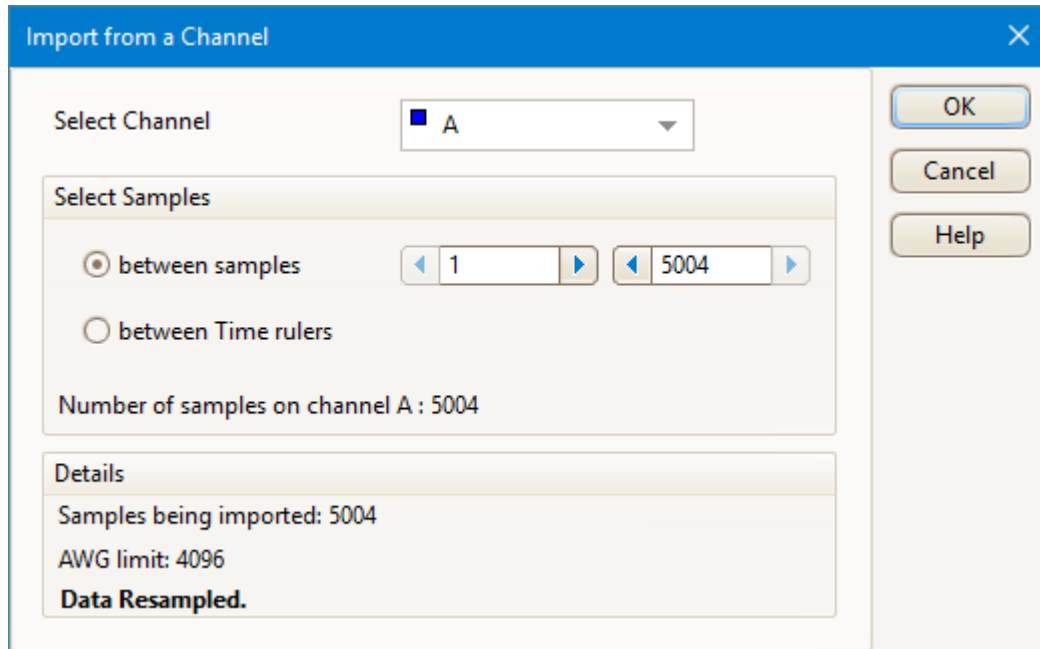
#### Apply

Copies the waveform from the graphical editor into the arbitrary waveform generator and remains in the Arbitrary Waveform Generator window.

### 6.4.2.2.1 Import from a Channel dialog

Location: [Arbitrary Waveform Generator window](#) > [Import from a Channel button](#)

Purpose: allows you to copy captured data from a scope channel to the [Arbitrary Waveform Generator window](#)



**Select Channel** You can import the latest waveform from any available channel

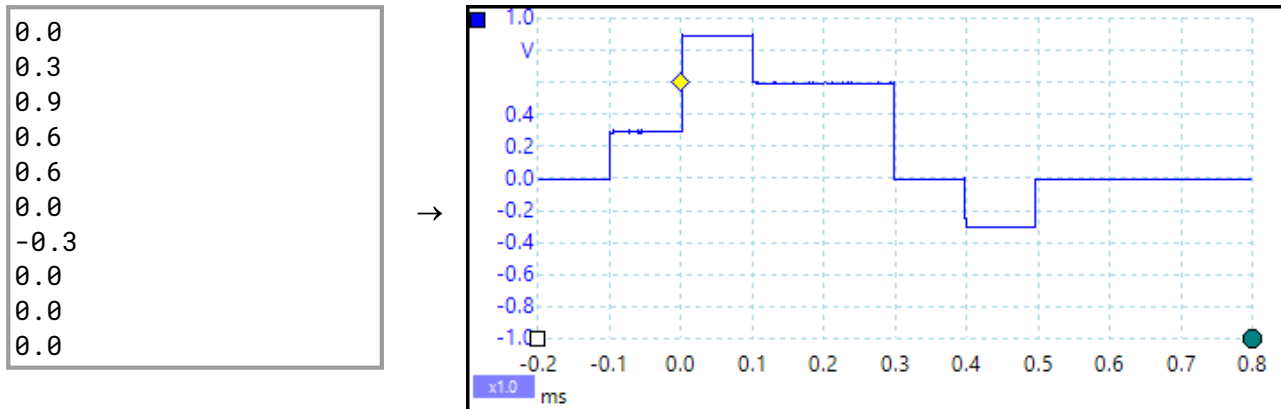
**Select Samples** By default, the entire capture is imported. This control allows you to specify a subset of the capture, either between specified sample numbers or between [time rulers](#). The subset will be scaled to fit the number of samples specified in the **Samples** control in the [Arbitrary Waveform window](#).

### 6.4.2.2.2 Import from CSV

Location: [Arbitrary Waveform Generator window](#) > [Import from CSV 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 6 as a [CSV](#) file, which must be laid out as a single column of decimal floating-point values, as in this example:



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 dialog](#) 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 dialog](#). 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 from CSV** 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.

### 6.4.2.2.3 Export as CSV

Location: [Arbitrary Waveform Generator window](#) > [Export as CSV button](#)

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

PicoScope 6 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 as CSV** 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.

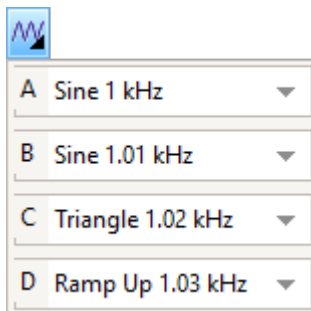
The file will contain between 10 and 8192 values between  $-1.0$  and  $+1.0$ , evenly spaced in time. The output is scaled to the amplitude selected in the [Signal Generator dialog](#) and the selected offset is added if necessary.

### 6.4.2.3 Demo Signals menu

Location: start PicoScope with no scope device plugged in, select [Demo mode](#), then click the [Signal Generator button](#)

Purpose: allows you to set up test signals so that you can experiment with PicoScope when no scope device is connected

The **Demo Signals** menu looks like this:

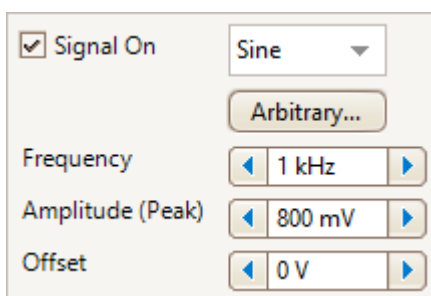


Click one of the channels to open the [Demo Signals dialog](#), which will allow you to set up a signal from that channel.

#### 6.4.2.3.1 Demo Signals dialog

Location: [Demo Signals menu](#) > select a channel

Purpose: controls one channel of the demo signal source, a feature of PicoScope that creates a variety of test signals to simulate a scope device.



**Signal On** Check this box to enable the demo signal source.

**Sine** **Signal type.** Select from a list of standard signal types.

**Arbitrary...** **Create Arbitrary Waveform.** Opens the [Arbitrary Waveform Generator window](#).

**1 kHz** **Frequency.** Type the desired frequency in hertz, or use the spin buttons.

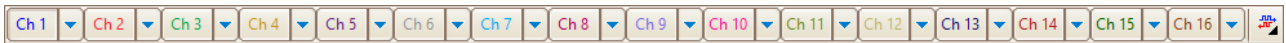
**800 mV** **Amplitude (Peak).** Type the desired amplitude in volts, or use the spin buttons.



**Offset.** Enter a number to add a DC offset to the demo signal. By default, the demo signals have a mean value of zero volts.

## 6.5 PicoLog 1000 Series Channels toolbar

The **Channels toolbar** controls the settings for each vertical input [channel](#). The toolbar has a different appearance for PicoLog 1000 Series Data Loggers than for PicoScope oscilloscopes. See [Channels toolbar \(PicoScope devices\)](#) for the standard version.



**Channel control.** This control contains two buttons in one rectangular outline. Click the small triangle on the right to open a simplified version of the [Channel Options menu](#), with options for [probes](#), [scaling](#) and zero offset. Click the channel name to toggle the channel on or off.

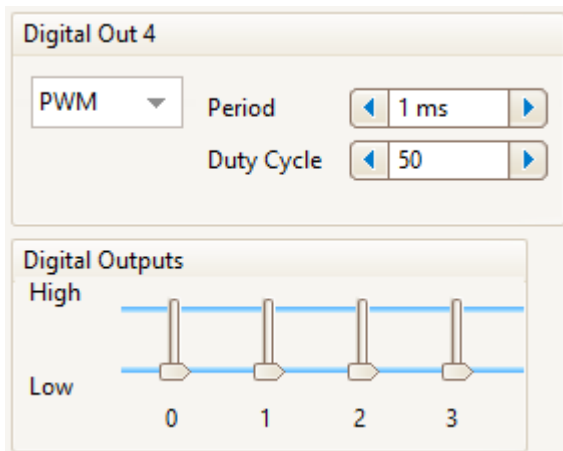


**Digital Outputs button.** For controlling the 2 or 4 digital outputs of the PicoLog 1000 Series device. Opens the [Digital Outputs dialog](#).

### 6.5.1 PicoLog 1000 Series Digital Outputs dialog

Location: [PicoLog 1000 Series Channels toolbar](#) > [Digital Outputs button](#)

Purpose: controls the [data logger's](#) built-in signal generator

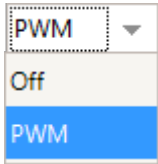


Digital Outputs dialog for the PicoLog 1216

The range of controls available depends on which model of data logger you have.

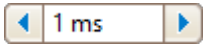
#### Digital Out 4

This set of controls allows you to set up a pulse-width modulated (PWM) waveform. This is a logic signal that is toggled with a specified period and duty cycle. The average value of the signal is proportional to its duty cycle, so it can be processed by an external lowpass filter to produce a signal that is proportional to the duty cycle.



**Off:** Disable the PWM output.

**PWM:** Enable the PWM output with the specified controllable **Period** and **Duty Cycle**.



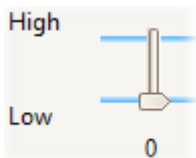
**Period.** Select the duration of one cycle of the PWM output.



**Duty Cycle.** The percentage of the PWM signal period that the signal spends at the logic high level. For example, if the period is 1 ms and the duty cycle 25%, then the signal will spend 25% of 1 ms = 250  $\mu$ s of each cycle at the logic high level, and the remaining 750  $\mu$ s at the logic low level. The voltages of the logic high and low levels are specified in the User's Guide for the data logger, but are typically 0 volts (low) and 3.3 volts (high). Using our example figures, the average value of the PWM output will be 25% x 3.3 volts = 0.825 volts.

## Digital Outputs

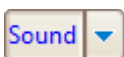
PicoLog PC Data Loggers have one or more digital outputs capable of driving low-current loads.



Each output can be set to a high or low logic level by moving the slider.

## 6.6 USB DrDAQ Channels toolbar

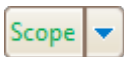
The **Channels toolbar for the USB DrDAQ** controls the settings for each input and output [channel](#). Click the channel name to toggle it on or off.



**Sound waveform sensor** control. The small arrow sets options for the sound waveform input (measured in uncalibrated amplitude units) using the on-board microphone.



**Sound level sensor** control. The small arrow sets options for the sound level input (measured in decibels) using the on-board microphone.



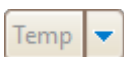
**Scope input** control. The small arrow sets options for the oscilloscope input (the BNC socket marked **Scope**), with options for [probes](#) and [scaling](#).



**Resistance input** control. The small arrow sets options for the 0 to 1 M $\Omega$  resistance-measuring input on the screw-terminal block.



**pH input** control. The small arrow sets options for the pH and redox/ORP (oxidation/reduction potential) measuring input.



**Temperature sensor** control. The small arrow sets options for the on-board temperature sensor.



**Light sensor** control. The small arrow sets options for the on-board light level sensor.



**External sensor** controls. The small arrows set options for external sensor inputs 1 to 3.



**Signal Generator** button. Opens the [Signal Generator dialog](#), which allows you to set the characteristics of the signal generator output.



**RGB LED** button. Opens the [RGB LED control dialog](#), which allows you to set the color of the on-board LED.

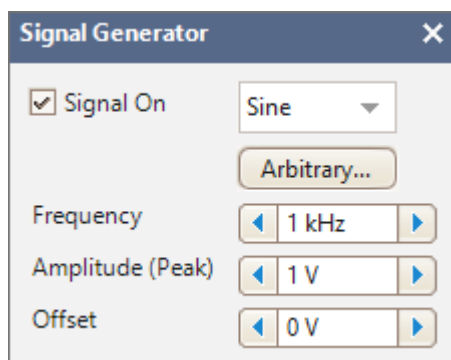


**Digital Outputs** button. Opens the [Digital Outputs dialog](#), which allows you to control the states of the four digital outputs.

## 6.6.1 USB DrDAQ Signal Generator dialog

Location: [USB DrDAQ Channels toolbar](#) > [Signal Generator button](#)

Purpose: controls the USB DrDAQ's built-in signal generator

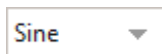


Signal generator dialog for the USB DrDAQ

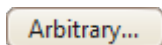
### Controls



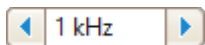
**Signal On.** Check this box to enable the signal generator.



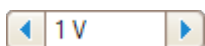
**Wave Type.** Select the shape of the waveform to be generated.



**Create Arbitrary Waveform.** Opens the [Arbitrary Waveform Generator window](#), allowing you to define your own waveform shape.



**Frequency.** Type in this box or use the spin buttons to select the frequency of the output waveform.



**Amplitude.** The amplitude of the waveform measured from peak to peak. For example, if **Amplitude** is 1 V and **Offset** is 0 V, the output will have a negative peak of -0.5 V and a positive peak of +0.5 V.

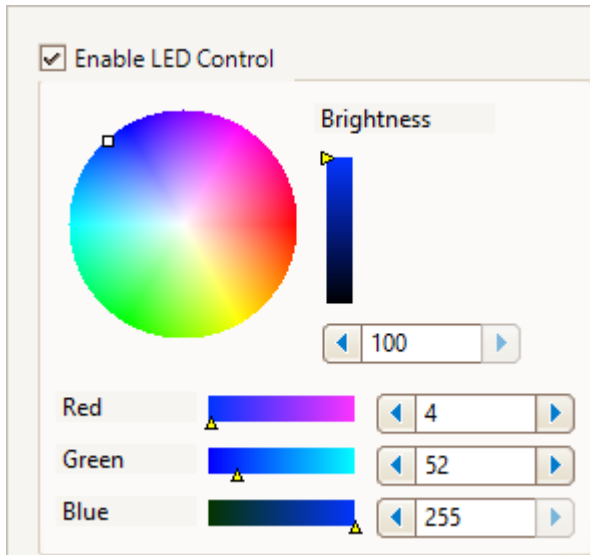


**Offset.** The mean value of the signal. For example, when **Offset** is 0 V, a sine or square wave will have equal positive and negative peak voltages.

## 6.6.2 USB DrDAQ RGB LED control

Location: [USB DrDAQ Channels toolbar](#) > [RGB LED button](#)

Purpose: allows you to set the color of the on-board LED to any one of 16.7 million colors



### Enable LED Control

**Box checked:** you can set the on-board RGB LED to any color

**Box clear:** the LED has its normal function of flashing red to indicate data capture on the input channels

### Other controls

Experiment with these to see what they do!

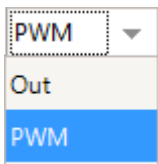
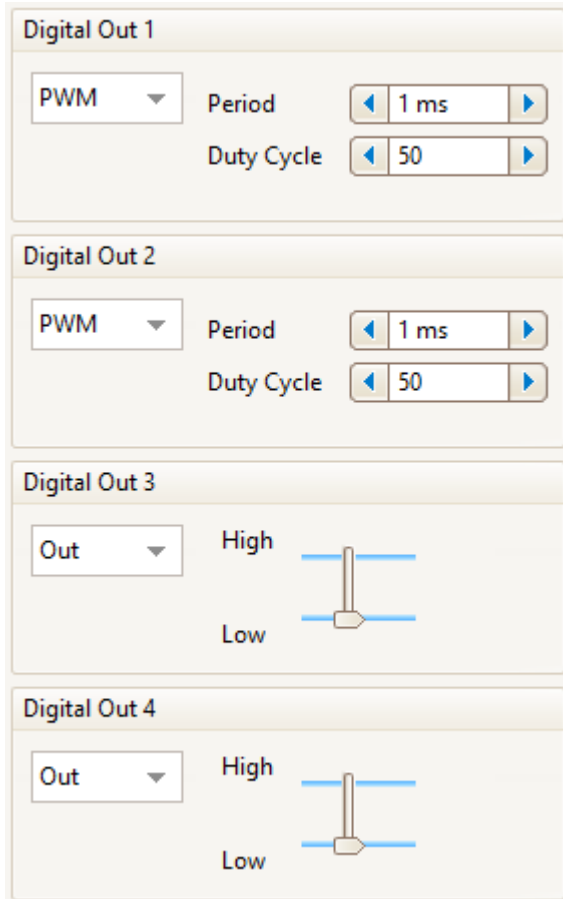


### 6.6.3 USB DrDAQ Digital Outputs dialog

Location: [USB DrDAQ Channels toolbar](#) > [Digital Outputs button](#)

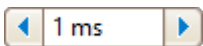
Purpose: allows you to set the characteristics of the four digital outputs on the screw-terminal block.

Each output has its own set of controls:



**Out:** you can set the output to either a fixed logic low (near 0 V) or a fixed logic high (near 3.3 V)

**PWM:** the output is a two-level pulse-width modulated (PWM) waveform (alternating between 0 V and 3.3 V) with variable **Period** and **Duty Cycle**. The signal can be filtered to produce a DC level proportional to the duty cycle.



**Period.** The time between successive pulses on the output

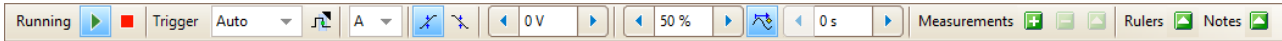


**Duty Cycle.** The percentage of the **Period** for which the output is high

## 6.7 Trigger toolbar

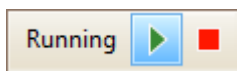
The Trigger toolbar is a general name for the collection of toolbars found at the bottom of the PicoScope 6 window by default. You can move the Trigger toolbar to the top of the window, directly below the [Channels toolbar](#), by checking the box on the [Options page](#) of the [Preferences dialog](#).

Click a button below for more information:



### 6.7.1 Stop and Go buttons

The **Stop** and **Go** buttons allow you to start and stop the scope device. Click anywhere on this part of the toolbar, or use a keyboard shortcut (by default, the space bar), to start or stop sampling.



**Go.** Highlighted if the oscilloscope is sampling.

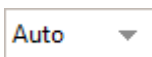
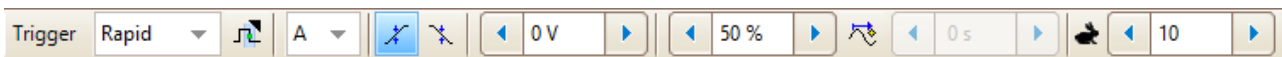


**Stop.** Highlighted if the oscilloscope is stopped.

These buttons are usually at the bottom of the PicoScope 6 window, but you can move them to the top by selecting **Move Trigger toolbar to top** on the [Options page](#) of the [Preferences dialog](#).

### 6.7.2 Trigger controls

The **Trigger controls** tell the scope device when to start capturing data. See also [Trigger](#).



**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 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](#) and the **Advanced Triggers** button will be disabled.



**Advanced Triggers.** Click to open the [Advanced Triggers dialog](#), which gives you extra trigger types beyond the simple edge trigger. If this button is disabled, it is either because the Trigger Mode has been set to **None** or **ETS**, or because your scope device does not support advanced triggers. To enable the **Advanced Triggers** button, set the **Trigger Mode** to **Auto**, **Repeat**, **Single** or **Rapid**.



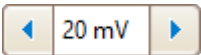
**Trigger Channel.** This is the channel that PicoScope monitors for the trigger condition. You can set it to any of the scope channels, or to **Ext** or **AuxIO**.



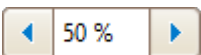
**Rising Edge.** Click to trigger on the rising edge of the waveform.



**Falling Edge.** Click to trigger on the falling edge of the waveform.



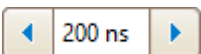
**Threshold.** Sets the voltage threshold for the trigger. You can also set the threshold by dragging the [trigger marker](#) up or down the screen.



**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.



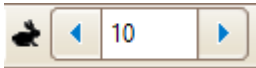
**Time-delay Enable.** Click this button to toggle the [Time-delay](#) control. You can use this button to switch the [scope view](#) between the trigger event and the trigger reference point.



**Time-delay.** The time-delay function allows you to view and analyze data from a certain time after the trigger event, by introducing an offset between the trigger event and the trigger marker. For this control to take effect, make sure that you have clicked the **Time-delay Enable** button and that it is highlighted in blue.

You can reposition the trigger marker, while preserving the time delay you have set, by dragging it with the mouse. As you drag the marker, the [time-delay arrow](#) appears on the screen: it is displayed until you release the mouse button.

See [Trigger Timing](#) for information on how the **Pre-trigger** and **Time-delay** controls interact.



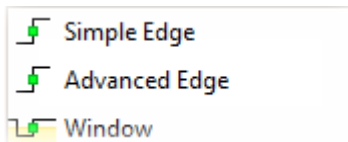
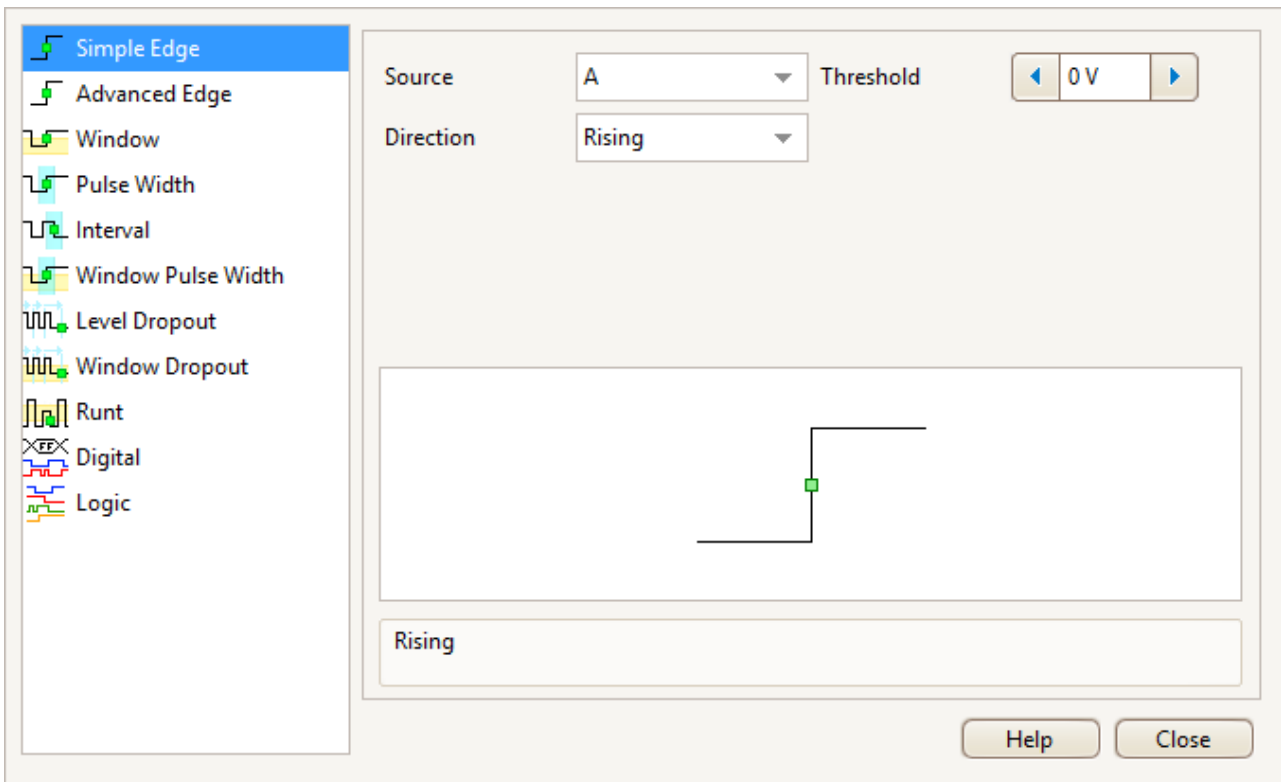
**Rapid Captures.** In [Rapid](#) trigger mode, this is the number of waveforms to capture in a sequence. They will be captured with the minimum possible [dead time](#) between them.

These controls are usually at the bottom of the PicoScope 6 window, but you can move them to the top by selecting **Move Trigger toolbar to top** on the [Options page](#) of the [Preferences dialog](#).

### 6.7.2.1 Advanced Triggers dialog

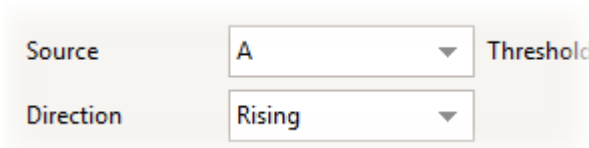
Location: [Trigger controls](#) > [Advanced Triggers button](#)

Purpose: allows you to set up more complex trigger types



**List of advanced trigger types.** This control lists all the available [advanced trigger types](#). Click on the condition you require, and a diagram and description will appear alongside the advanced trigger options on the right.

If [ETS triggering](#) is enabled in the [Trigger controls](#), the trigger type will be set to **Simple Edge**.



**Advanced trigger options.** The options available depend on the trigger type selected: see [Advanced trigger types](#) for a list.

### 6.7.2.1.1 Advanced trigger types

The **advanced trigger types** can be switched on in the [Advanced Triggers dialog](#).

For all trigger types except [Digital](#), the first step is to select which signal the scope should use as the trigger; so set **Trigger Channel** to the desired channel, or to **Ext** or **AuxIO**. These names correspond to the BNC input connectors on the scope device. Then choose one of the trigger types below.



**Simple Edge.** This type provides the same **Rising** and **Falling** edge triggers that are available from the [Trigger controls](#). It is included in the [Advanced Triggers dialog](#) as an alternative setup method.

You can set the trigger [threshold](#) in this dialog, or you can drag the [trigger marker](#) on the scope view.

This is the only trigger type that is compatible with [ETS](#) mode.



**Advanced Edge.** This trigger type is similar to the simple edge trigger, but with two additional options. The **Rising or Falling** option triggers on both edges of a waveform, and is useful for monitoring pulses of both polarities at once (dual-edge triggering). [Hysteresis](#) can be set to reduce the number of false triggers on a noisy signal, and is described in a separate topic.



**Window.** This trigger type detects when the signal enters or leaves a specified voltage window. The **Direction** control specifies whether the trigger should detect the signal entering the window, leaving it, or both. **Threshold 1** and **Threshold 2** are the upper and lower voltage limits of the window. The order in which you specify the two voltages does not matter. [Hysteresis](#) is described in a separate topic.



**Pulse Width.** This trigger type detects pulses of a specified width.

First set the **Pulse Direction** to either **Positive** or **Negative** according to the polarity of the pulse you are interested in.

Next, set one of the four **Condition** options:

**Greater than** triggers on pulses wider than the specified time.

**Less than** triggers on pulses that are narrower (useful for finding glitches).

**Inside time range** triggers on pulses that are wider than **Time 1** but no wider than **Time 2** (useful for finding pulses that meet a specification).

**Outside time range** does the opposite: it triggers on pulses that are either narrower than **Time 1** or wider than **Time 2** (useful for finding pulses that violate a specification).

Now set the trigger **Threshold** in volts or other units, or drag the [trigger marker](#) on the scope view.

Finally, set up **Time 1** (and **Time 2** if present) to define the pulse width.



**Interval.** This type lets you search for two successive edges of the same polarity that are separated by a specified interval of time.

First, set the **Starting edge** to either **Rising** or **Falling** according to the polarity of the edges you are interested in.

Next, select one of the four **Condition** options:

**Greater than** triggers when the second edge occurs later than **Time 1** after the first edge (useful for detecting missing events).

**Less than** triggers when the second edge occurs earlier than **Time 1** after the first edge (useful for detecting timing violations and spurious edges).

**Inside time range** triggers when the second edge is later than **Time 1** after the first edge and earlier than **Time 2** (useful for finding valid edges).

**Outside time range** triggers when the second edge is earlier than **Time 1** after the first edge or later than **Time 2** (useful for finding spurious edges).

Finally, set up **Time 1** (and **Time 2** if present) to define the time interval.



**Window pulse width.** This is a combination of the window trigger and the pulse width trigger. It detects when the signal enters or leaves an input range for a specified period of time.



**Level dropout.** This detects an edge followed by a specified time with no edges. It is useful for triggering on the end of a pulse train.



**Window dropout.** This is a combination of the window trigger and the dropout trigger. It detects when the signal enters a specified input range and stays there for a specified time. This is useful for detecting when a signal gets stuck at a particular level.



**Runt.** Detects a pulse that crosses one threshold and then falls below the same threshold, without crossing the second threshold. This is typically used for finding pulses that fail to reach a valid logic level.



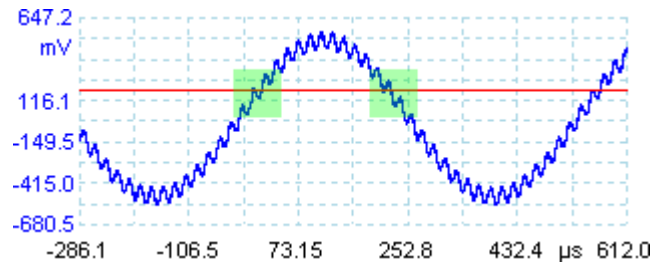
**Digital** ([MSO](#) devices only). Triggers on a combination of the state of the digital inputs and a transition (edge) on one digital input. See [Digital trigger](#).



**Logic.** This detects a logical combination of the scope's inputs. The conditions that can be applied to each input vary: analog inputs can be edge-, level- or window-qualified; EXT and D15...D0 (if present) are level-qualified with a variable threshold; and AUX is level-qualified with a fixed TTL threshold. See [Logic trigger](#).

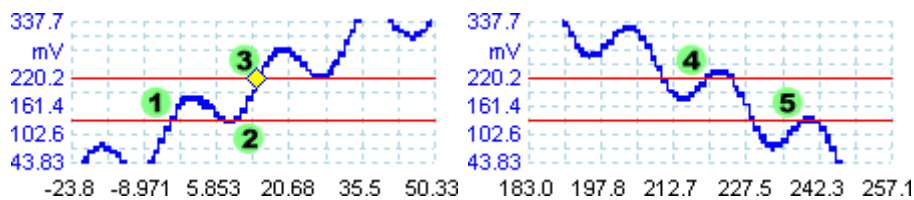
## 6.7.2.1.1.1 Hysteresis

**Hysteresis** is a feature of the [advanced trigger types](#) in PicoScope 6 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 lower red line. The upper 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.

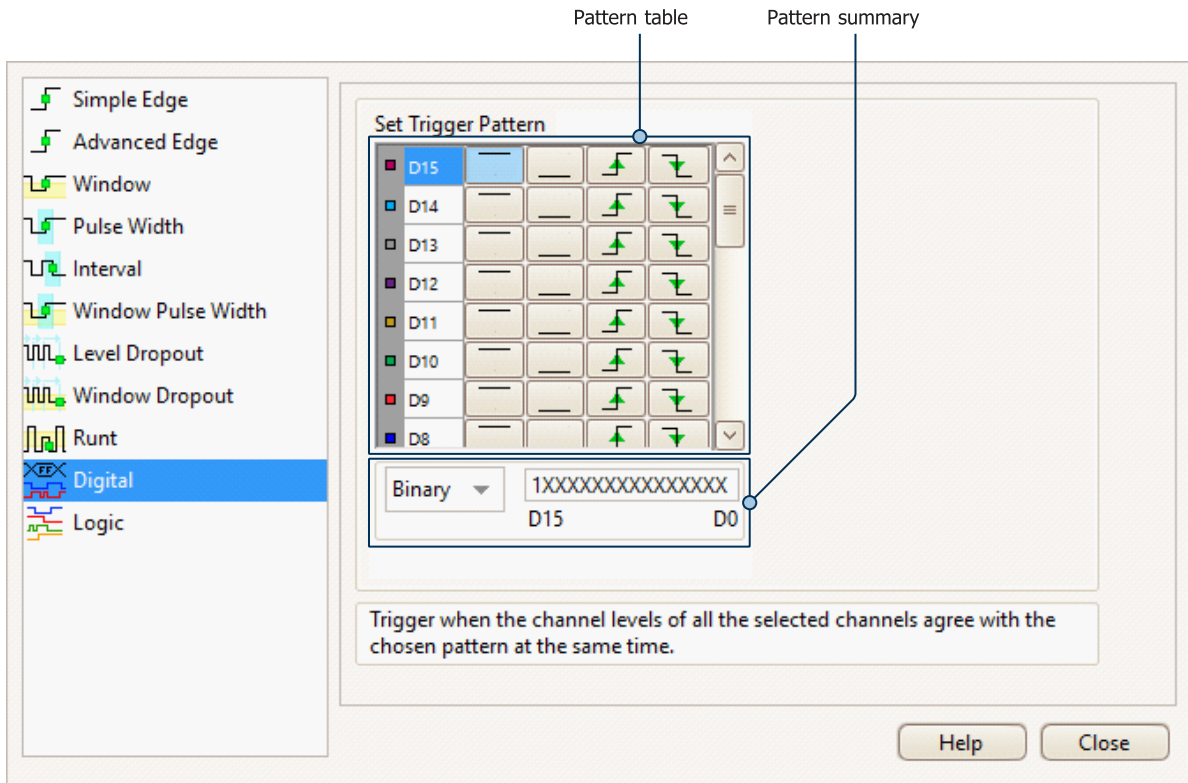
Hysteresis is enabled by default for all the advanced trigger types. The **Hysteresis** controls in the [Advanced triggers dialog](#) let you change the hysteresis voltage as a percentage of full scale. The trigger marker  shows the size of the hysteresis window.

6.7.2.1.1.2 Digital trigger dialog

Location: [Advanced Triggers dialog](#) > **Digital**

Purpose: sets up triggering on digital inputs

Applicability: [MSO devices](#) only



**Pattern table**

Lists all available inputs as selected in the [Select Digital Channels/Groups dialog](#). Each one can be monitored for a low or high level or a rising or falling edge, or ignored. Any number of levels can be specified, but no more than one transition (edge).

- D11     D11 = X (don't care)
- D11     D11 = 0 (low level)
- D11     D11 = 1 (high level)
- D11     D11 = R (rising edge)
- D11     D11 = F (falling edge)



**Pattern summary**

This section contains the same settings as the **pattern table** but in a more concise format.

Binary ▾

The numerical format to use for this section: **Binary** or **Hex**.

RXXXXXXXX1010XXXX  
D15 D0

The complete triggering pattern and transition. In **Binary** mode the bits are labeled as follows:

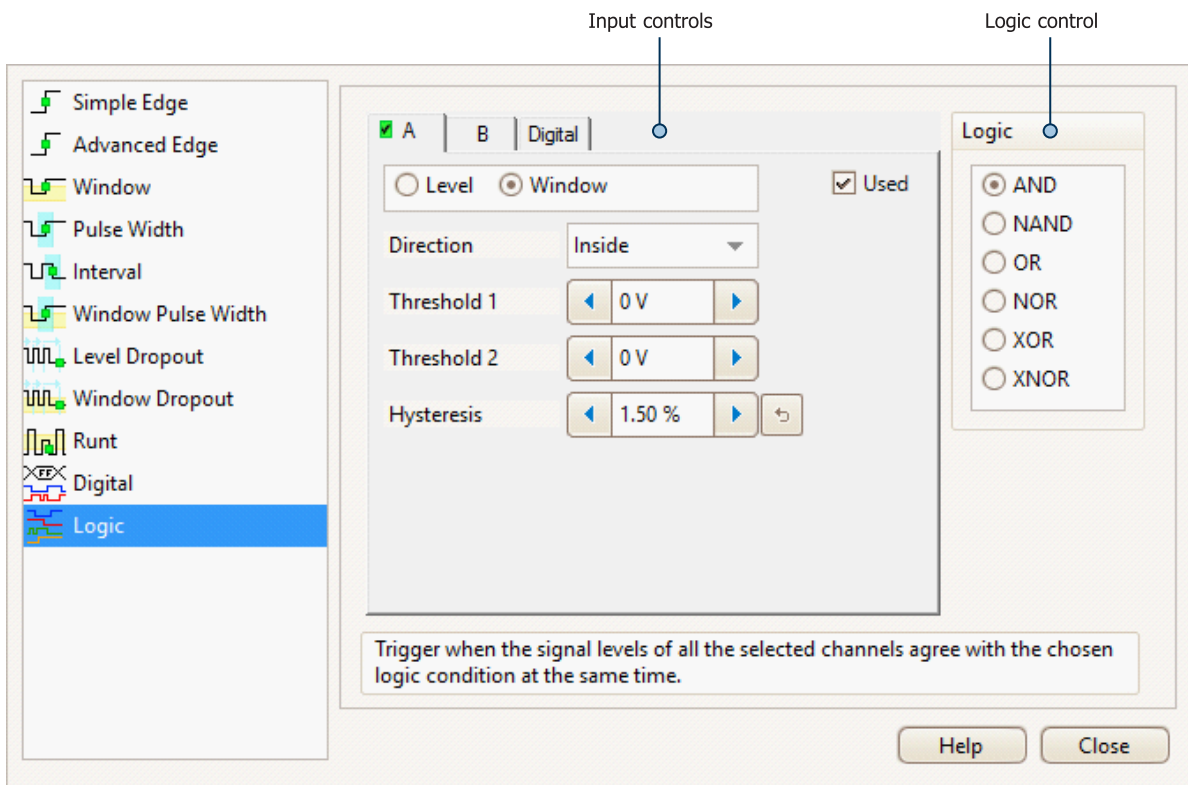
- X** = don't care
- 0** = binary 0
- 1** = binary 1
- R** = rising edge
- F** = falling edge

6.7.2.1.1.3 Logic trigger dialog

Location: [Advanced Triggers dialog](#) > [Logic](#)

Purpose: sets up triggering on a combination of inputs

Applicability: all devices with more than one active input



**Input controls**

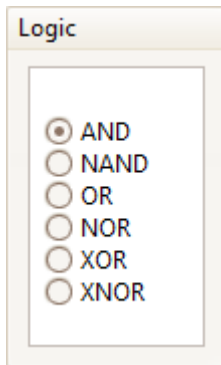
There is a set of controls for each active input of the oscilloscope. The selection of inputs depends on the model of oscilloscope in use. The selection of controls (thresholds, hysteresis, window mode and so on) for each input also depends on the hardware capabilities of the oscilloscope.

A

Channel A

<input type="checkbox"/> B	Channel B
<input type="checkbox"/> C	Channel C
<input type="checkbox"/> D	Channel D
<input type="checkbox"/> Ext	EXT input ( <a href="#">if present</a> )
<input type="checkbox"/> AuxIO	AUX input ( <a href="#">if present</a> )
<input type="checkbox"/> Digital	Digital inputs ( <a href="#">mixed-signal oscilloscopes</a> only). The controls here are the same as those in the <a href="#">Digital Trigger dialog</a> .
<input checked="" type="checkbox"/> Used	Check this box to include the relevant input in the Logic Trigger condition. If the box is not checked, the input will be ignored by the Logic Trigger.

### Logic control

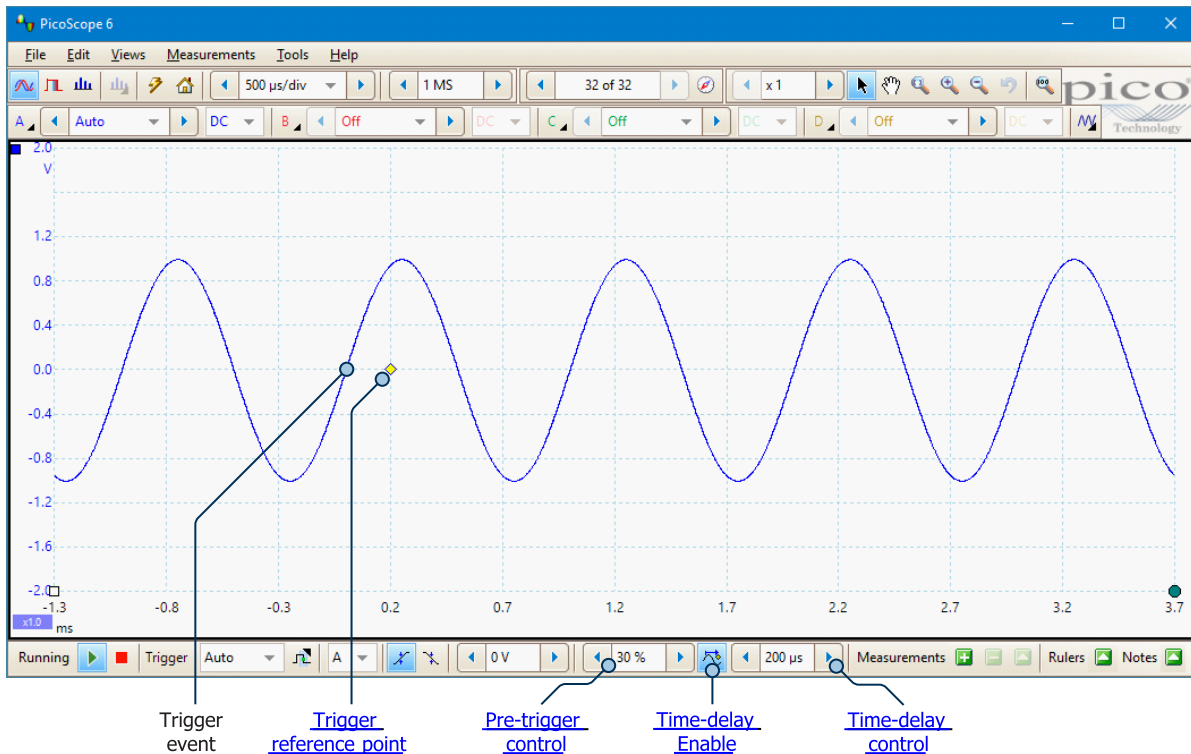


Specifies the Boolean operation used to combine the input trigger conditions. Only inputs with the **Used** box checked (see above) are included in the trigger logic.

AND	All of the input trigger conditions must be met
NAND	None of the input trigger conditions must be met
OR	One or more of the input trigger conditions must be met
NOR	None of the input trigger conditions must be met
XOR	An odd number of input trigger conditions must be met
XNOR	An even number of the input trigger conditions must be met

### 6.7.2.2 Trigger timing

The **pre-trigger** and **time-delay** functions are both described individually under [Trigger controls](#), but it is also important to understand how they interact with one another. Here is a screenshot of a [scope view](#) with time delay enabled:



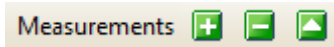
The **pre-trigger** control positions the scope view in relation to the **trigger reference point**, allowing you to decide how much of the waveform should appear before the reference point, and how much after it. In this example, it is set to 30%, so the trigger reference point appears 30% of the way across the display from the left-hand edge.

The trigger reference point (◊) does not lie on the waveform. This is because the time-delay control is set to 200 μs, which means that the trigger event occurred 200 μs before the reference point, closer to the left-hand edge of the display. The time axis is aligned so that the trigger reference point is at 200 μs. If the time delay is greater than the pre-trigger, the trigger event itself will be somewhere beyond the left-hand side of the display.




Note: The number of sampling intervals that can elapse over the course of the time delay is limited by the scope device's capture memory. The trigger event and trigger reference must both occur within the time taken to fill the waveform buffer. If you set too long a time delay, the PicoScope software will adjust the pre-trigger to prevent the total from exceeding this limit. Capture memory varies between oscilloscopes, so refer to the data sheet for the exact specification.

### 6.7.3 Measurements buttons

The **Measurements** buttons control the [Measurements table](#).



The buttons are as follows:

-  **Add Measurement.** Opens the [Add Measurement dialog](#), allowing you to add a measurement of a waveform to the selected view.
-  **Delete Measurement.** Deletes the currently selected row from the [Measurements table](#).
-  **Edit Measurement.** Opens the [Edit Measurement dialog](#) for the currently selected measurement. You can also edit a measurement by double-clicking on a row of the [Measurements table](#).

These buttons are usually at the bottom of the PicoScope 6 window, but you can move them to the top by selecting **Move Trigger toolbar to top** on the [Options page](#) of the [Preferences dialog](#).

### 6.7.4 Other buttons

The buttons in the bottom right corner of the PicoScope window control the **Phase (Rotation) Rulers, Notes** and **Channel Labels** (PicoScope 6 Automotive only).



The buttons are as follows:

- |                       |   |
|-----------------------|---|
| <b>Rulers</b>         | Opens the <a href="#">Ruler Settings dialog</a> that controls the <b>Phase rulers</b> (called <b>Rotation rulers</b> in PicoScope 6 Automotive) |
| <b>Notes</b>          | Displays the <a href="#">Notes</a> at the bottom of the window  |
| <b>Channel Labels</b> | PicoScope 6 Automotive only. Displays the <a href="#">Channel Labels</a> at the bottom of the window.   |

These buttons are usually at the bottom of the PicoScope 6 window, but you can move them to the top by selecting **Move Trigger toolbar to top** on the [Options page](#) of the [Preferences dialog](#).

## 7 How to...

This chapter explains how to perform some common tasks.

### 7.1 How to change to a different device

1. Unplug the old [device](#).
2. Cancel the **Check USB cable...** dialog.
3. Plug in the new device.
4. PicoScope will detect the new device and begin to use it.

Alternatively, close and reopen the PicoScope software after plugging in the new device.

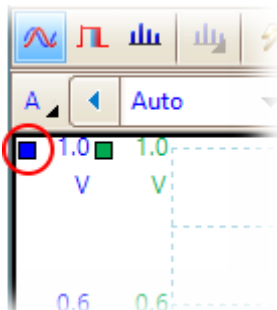
### 7.2 How to use rulers to measure a signal

#### Using a single ruler for signal-to-ground measurements

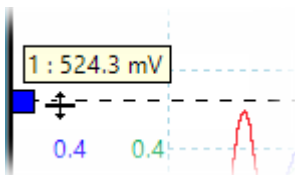
1. Look at the [Channel controls](#) to find the color code for the [channel](#) you wish to measure. The default colors are shown in the image below. In this example, we will measure the signal on Channel A.



2. Find the ruler handle (the small colored square in the top left or top right corner of the [scope view](#) or [spectrum view](#)) in this color:



3. Drag the ruler handle downwards. A [signal ruler](#) (horizontal broken line) will appear across the view. Release the ruler handle when the ruler is where you want it.



4. Look at the [ruler legend](#) (the small table that appears on the view). It should have a row marked by a small square the same color as your ruler handle. The first column shows the signal level of the ruler.

1	2	Δ	-	×
■ 524.3 mV	--.--	--.--		

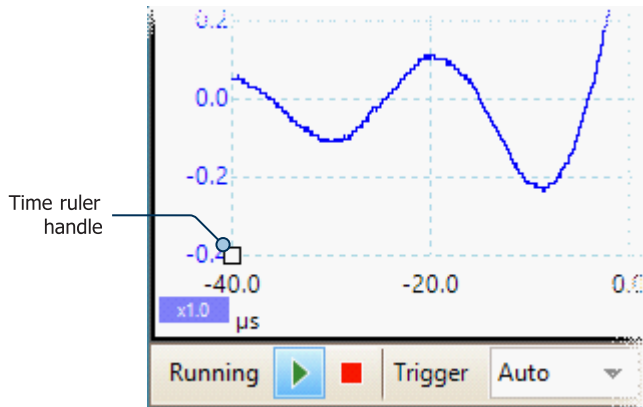
**Using two rulers for differential measurements**

1. Follow the steps above for using a single ruler.
2. Drag the second ruler handle of the same color downwards until its ruler is at the signal level to be measured.
3. Look at the [ruler legend](#) again. The second column now shows the signal level of the second ruler, and the third column shows the difference between the two rulers.

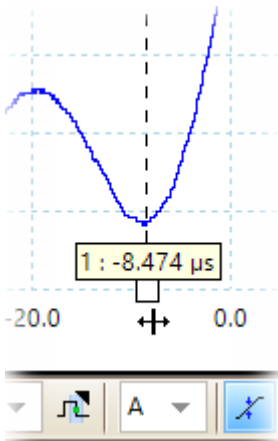
1	2	$\Delta$
524.3 mV	-529.4 mV	1.054 V

### 7.3 How to measure a time difference

1. Find the time ruler handle (the small white square in the bottom left corner of the [scope view](#)).



2. Drag the time ruler handle to the right. A [time ruler](#) (vertical broken line) will appear on the scope view. Release the ruler handle when the ruler is at the time you wish to use as the reference.



3. Drag the second time ruler handle to the right until its ruler is at the time to be measured.
4. Look at the [ruler legend](#) (the small table that appears on the scope view). It should have a row marked by a small white square. The first two columns show the times of the two rulers, and the third column shows the time difference.

1	2	$\Delta$
-8.474 $\mu$ s	21.26 $\mu$ s	29.74 $\mu$ s

5. The [frequency legend](#) shows  $1/\Delta$ , where  $\Delta$  is the time difference.

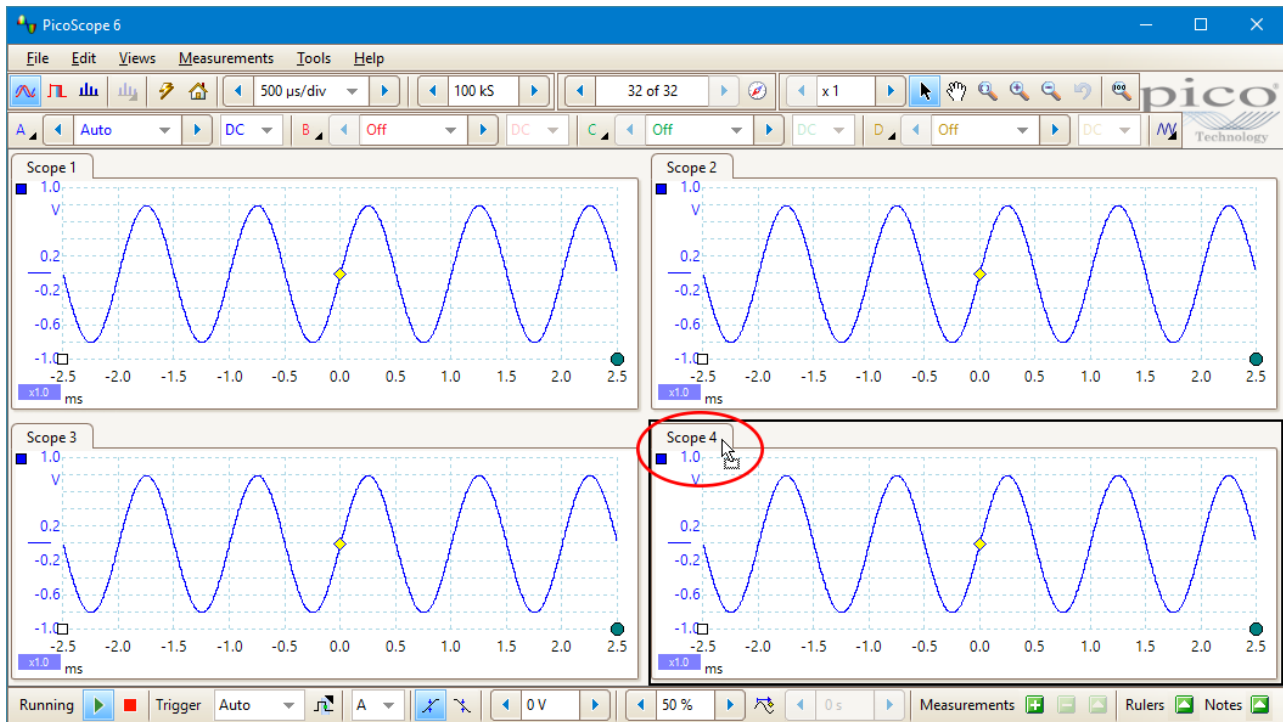
$1/\Delta$  1.331 kHz, 79843 RPM

You can use a similar method to measure a frequency difference on a [spectrum view](#).

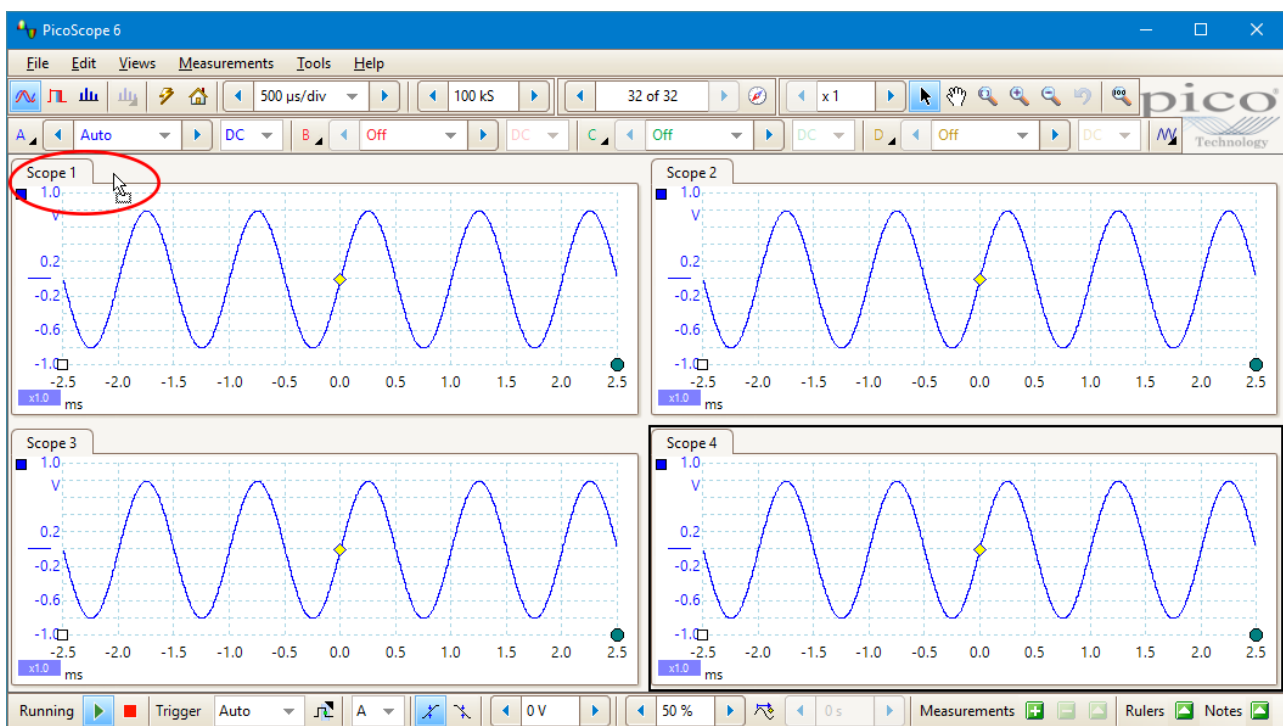
## 7.4 How to move a view

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.

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

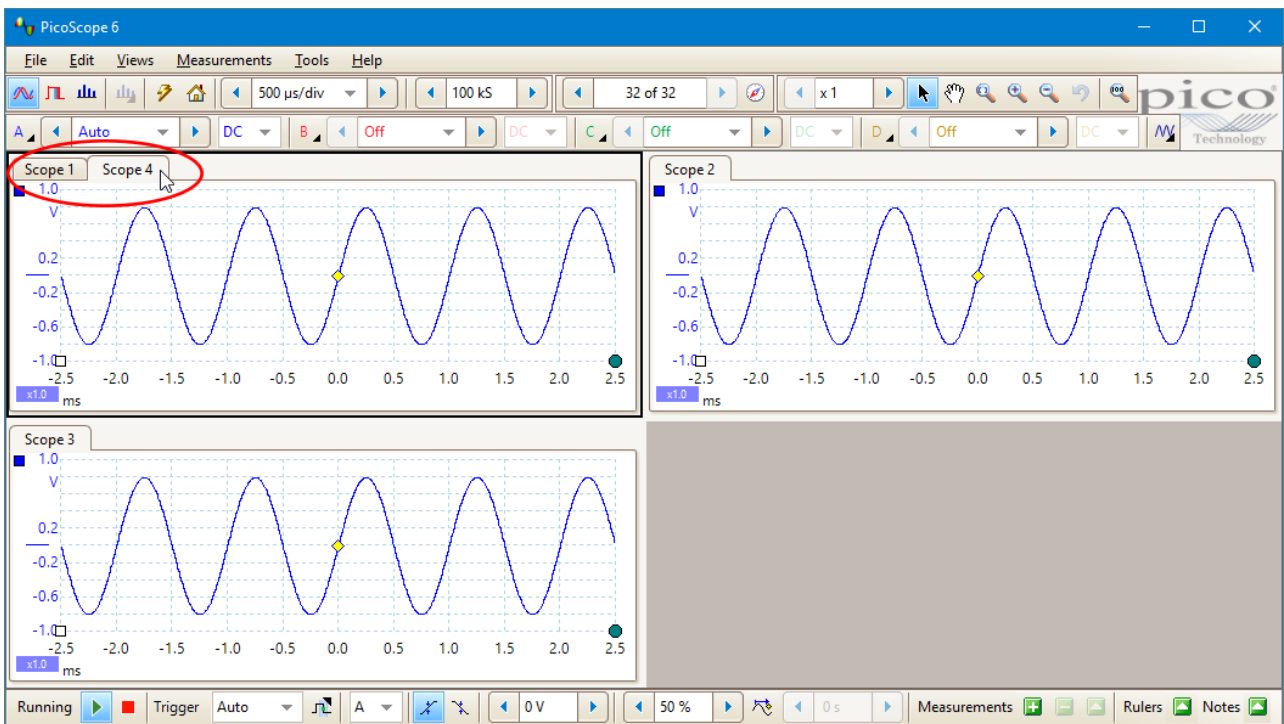


2. Drag the mouse pointer to the new location, next to the name tab of the Scope 1 view.





3. Release the mouse button. The Scope 4 view will move to its new location.



## 7.5 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 (see [Device feature table](#)).

### Global zooming and scrolling

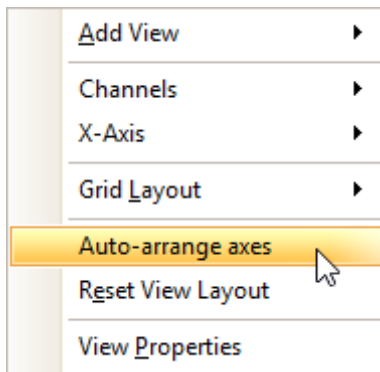
This is usually the quickest way to get a closer look at the fine detail on your signals. The global zooming and scrolling tools move all the signals at once and are found on the [zooming and scrolling toolbar](#).



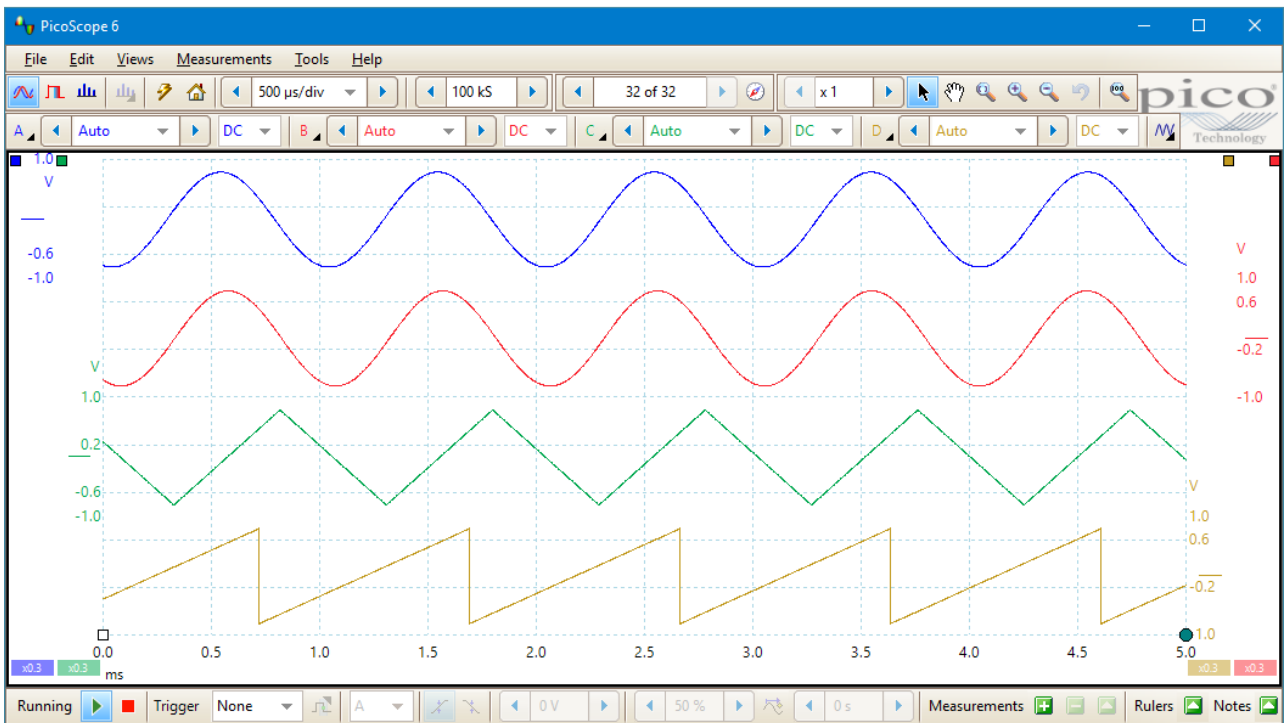
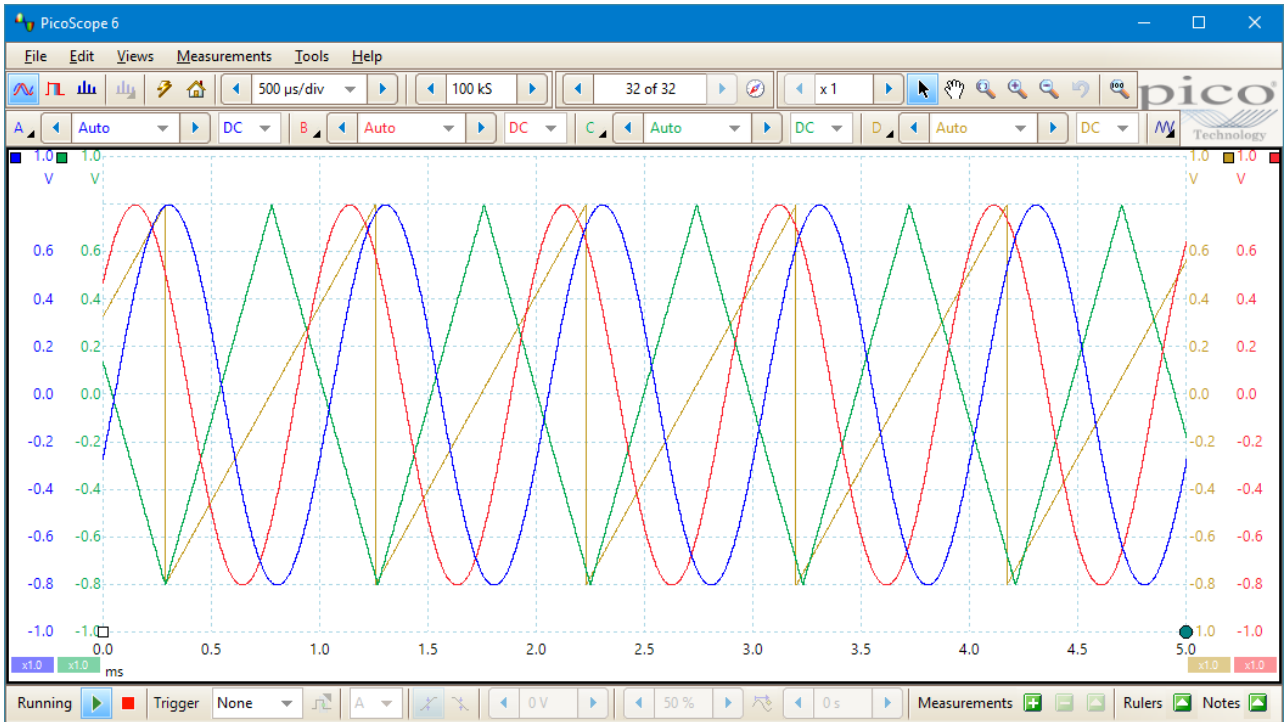
When a view is zoomed in, it has vertical and horizontal scroll bars that let you move the signals around as a group. You can also use the hand tool to scroll around the graph.

### Auto-arrange axes

Right-click the scope or spectrum view, or open the [Views menu](#), and select [Auto-arrange axes](#):




PicoScope automatically scales and offsets the channels to make them fit on the view without overlapping. This is the quickest way to tidy up the scope view:



### Axis scaling and offset

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).

Click the scaling button  at the bottom of the axis you wish to modify, and the [axis scaling controls](#) will appear. To adjust the offset without using the axis scaling controls, click on the vertical axis and drag it up or down.

### 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.

- Custom Probe scaling is a permanent transformation. The scaling is applied when the waveform is captured and cannot be changed later.
- 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.

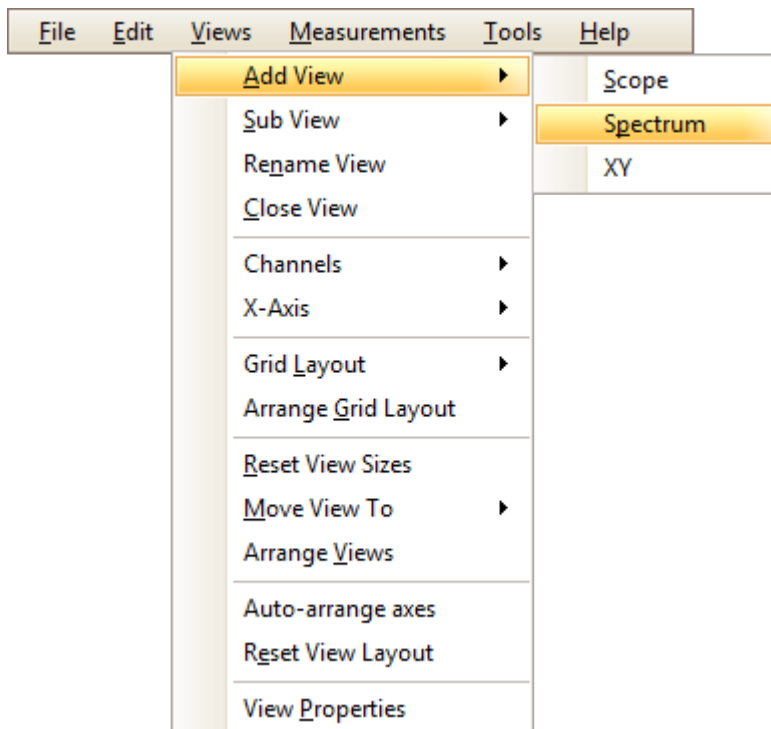
## 7.6 How to set up the spectrum view

### Creating a spectrum view

First, make sure that the [trigger mode](#) is not set to [ETS](#), as it is not possible to open a spectrum view in ETS trigger mode.

There are three ways to open a [spectrum view](#):

- Click the **Spectrum Mode button** in the [Capture Setup toolbar](#). We recommend using this method to get the best spectrum analysis performance from your scope. Once in Spectrum Mode, you can still open a scope view to see your data in the time domain, but PicoScope optimizes the settings for the spectrum view.
- Go to the [Views menu](#) select **Add view**, then select **Spectrum**.



This method opens a spectrum view in the currently selected mode, whether this is Scope Mode or Spectrum Mode. For best results, we recommend that you switch to Spectrum Mode, as described in the method immediately above.

- Right-click on any [view](#), select **Add view**, then select **Spectrum**. The menu is similar to the [Views menu](#) shown above.

### Configuring the spectrum view

See [Spectrum Settings dialog](#).

### Selecting the source data


PicoScope can produce a [spectrum view](#) based on either live or stored data. If PicoScope is running (the [Go button](#) is highlighted in blue), the spectrum view represents live data. Otherwise, with PicoScope stopped (the [Stop button](#) is highlighted in blue), the view represents data stored in the currently selected page of the waveform buffer. When PicoScope is stopped, you can use the [buffer controls](#) to scroll through the buffer and the spectrum view will be recalculated from the waveform currently selected.

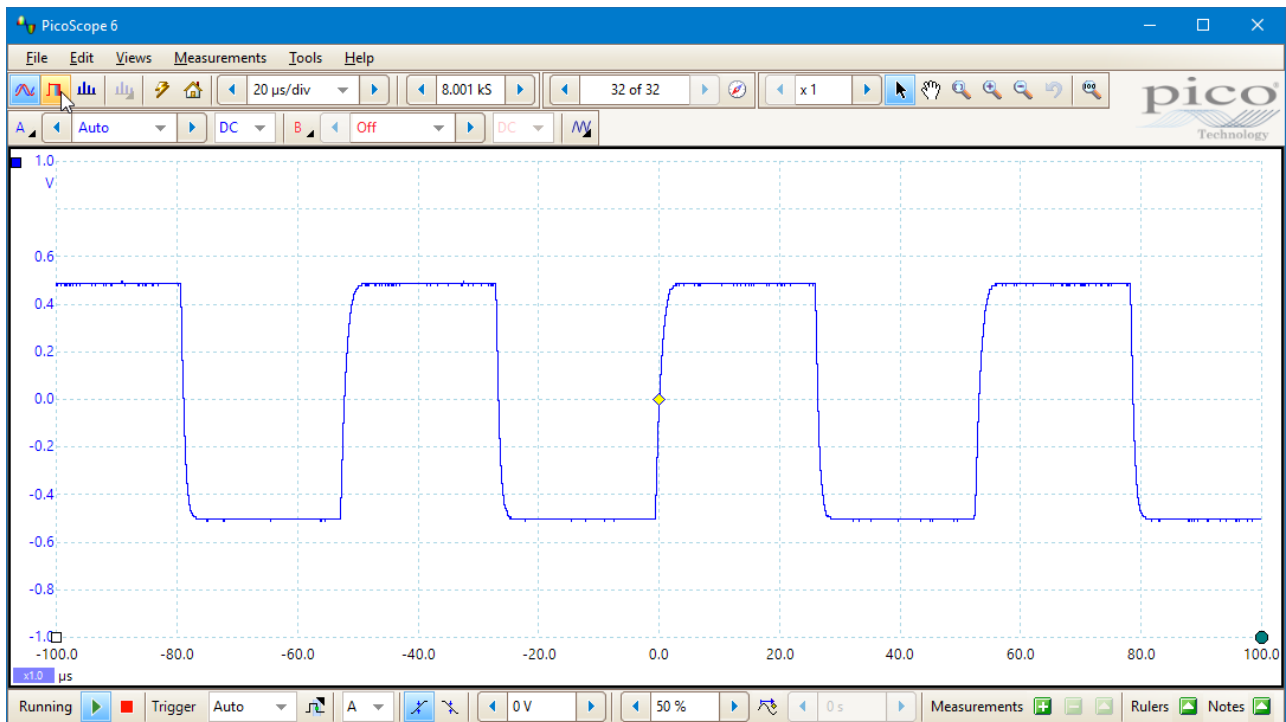
## 7.7 How to find a glitch using persistence mode

[Persistence mode](#) helps you find rare events hidden in otherwise repetitive waveforms. In normal [scope mode](#), such an event may appear on the display for just a fraction of a second, too quickly for you to stop the capture and freeze it on the screen. Persistence mode keeps the event on the display for a predetermined time, allowing you to set up the trigger options to capture it more reliably.

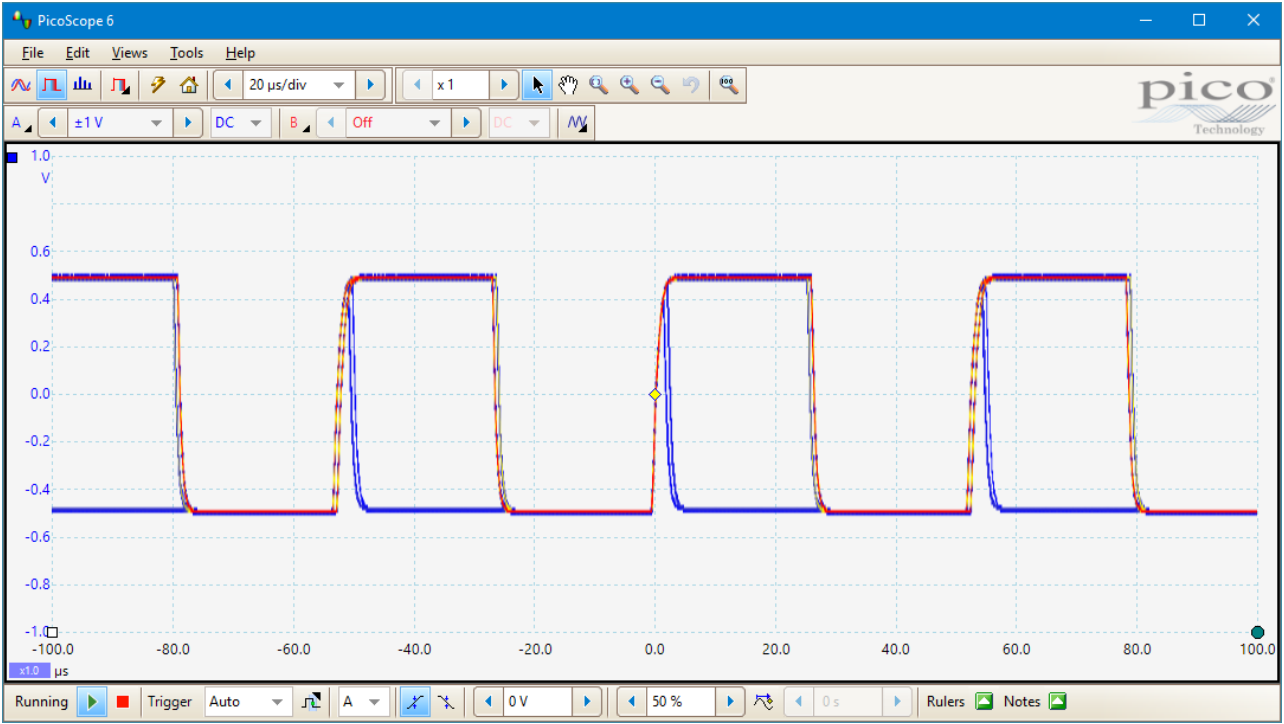
In the example below, we will look at a waveform with an occasional glitch, identify it in persistence mode and set up a trigger in scope mode so we can analyze it in more detail.

### Step-by-step guide

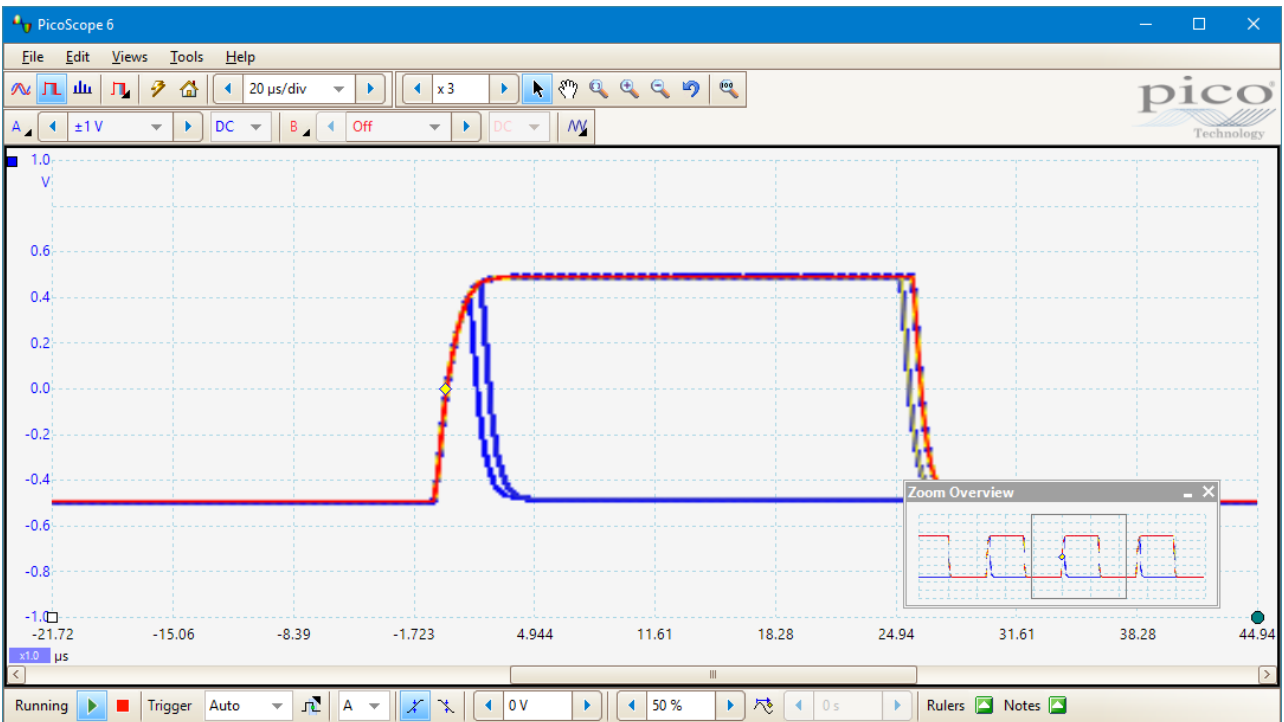
- Set up the scope to display your waveform, then click the **Persistence Mode** button . It is important that the waveform display is stable. We have set a simple rising-edge trigger.





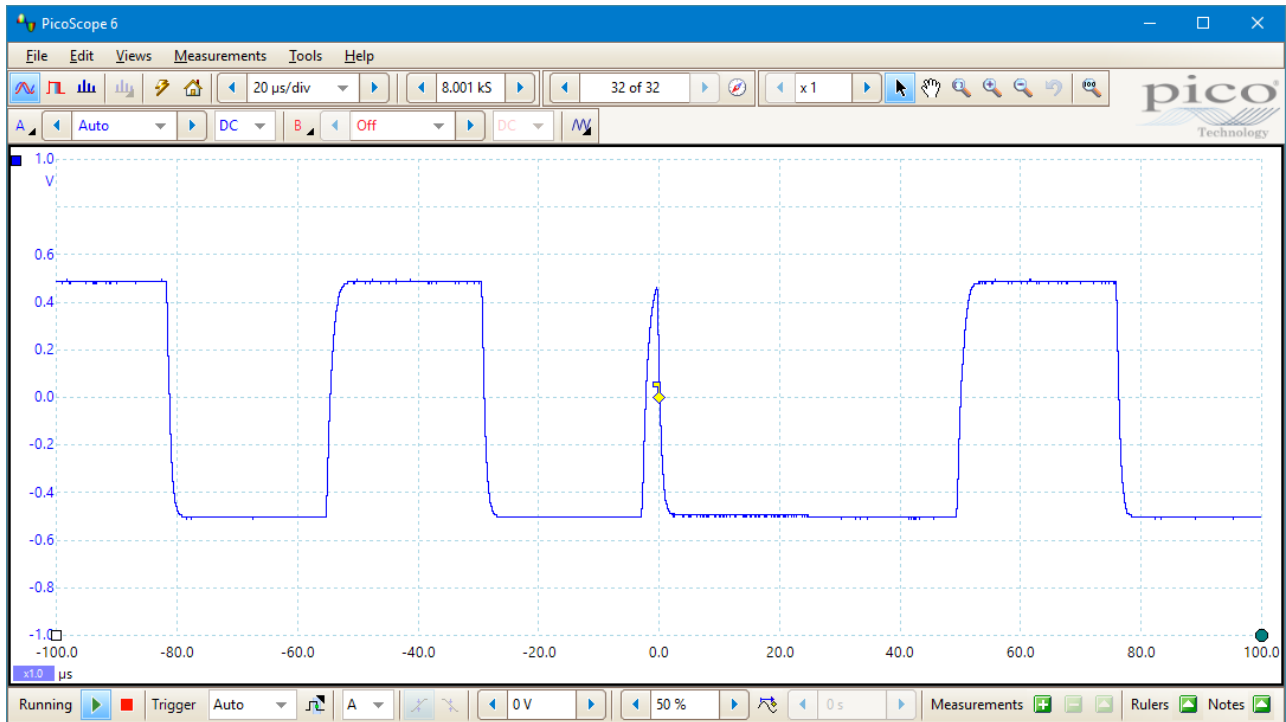
- Our original scope view is replaced by a persistence view, as shown below. At this point, we have left the **Saturation** control in the [Persistence Options dialog](#) on its default setting of 50%, so the most common samples are shown in red and the least common in blue: the patches of yellow fall somewhere between the two. We can see that although there is a clear and regular cycle shown in red and yellow, there is also a less regular and less frequent pattern shown in blue.



- Looking more closely, we can also see that there appears to be a much narrower pulse than usual immediately after the trigger marker. It is also visible approximately 50 μs before the trigger marker and about 55 μs later. Below is a close-up of one of these spikes:



- Persistence mode has done its job: we have found our glitch. In order to examine our waveform in more detail and carry out automatic measurements, we will need to click the **Scope Mode** button  and return to [Scope mode](#).
- Now we know what the glitch looks like, we can click the **Advanced Triggers** button  and set up a pulse-width trigger to look for a positive pulse narrower than  $10\ \mu\text{s}$ . PicoScope finds it immediately:



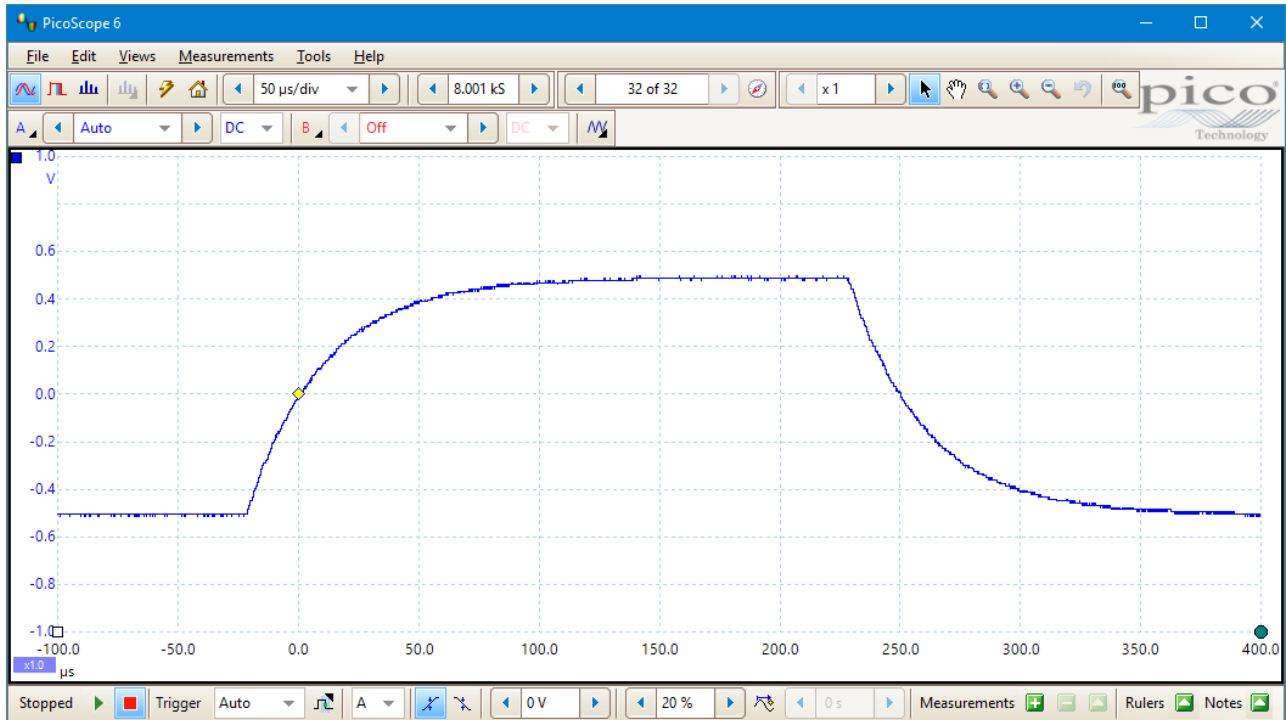
- We can now add automatic measurements or drag the rulers into place to analyze the spike in detail.



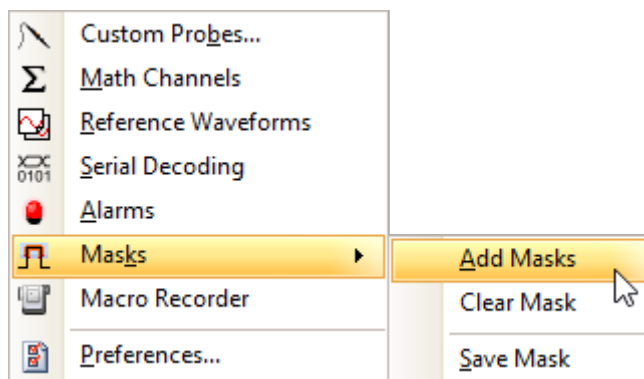
## 7.8 How to set up a mask limit test

You can carry out a [mask limit test](#) in a [Scope](#), [Spectrum](#) or [XY](#) view. In this example, we will use a scope view.

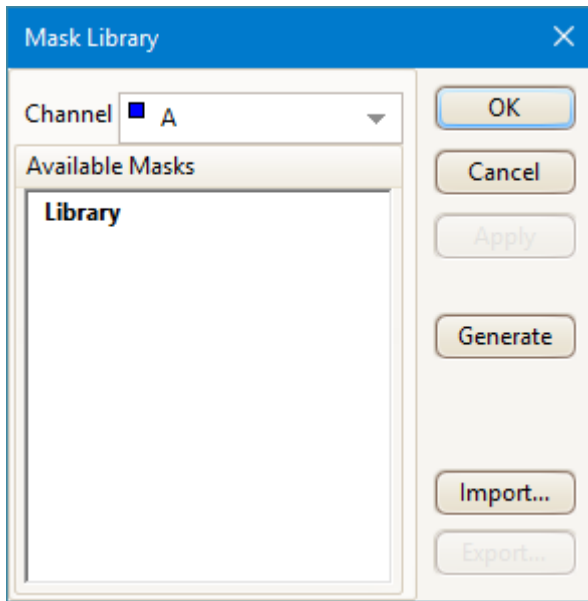
1. Set up the display to show a stable waveform. Adjust the input range and collection time so that the feature of interest fills most of the view. In this example, we are viewing a repetitive pulse, such as you might find on a data bus.



2. Select the **Tools > Masks > Add Masks** command.

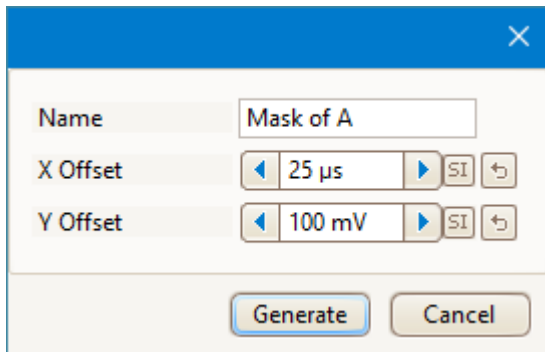


- The [Mask Library dialog](#) will open:

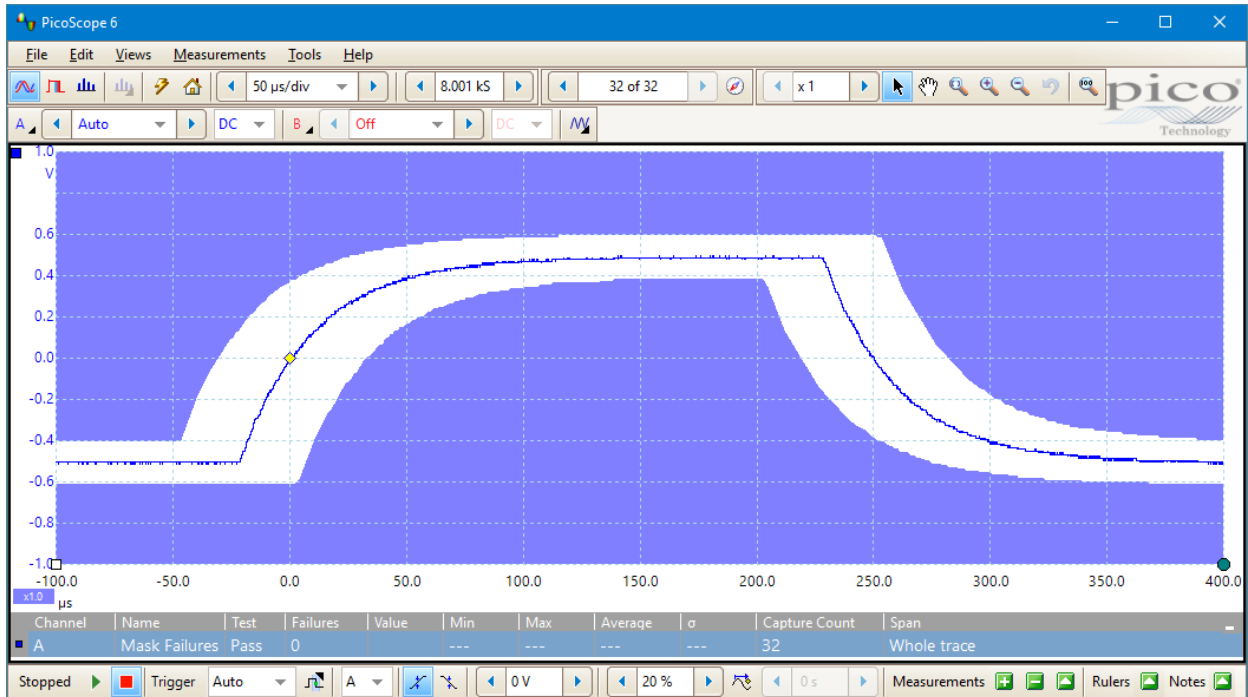


Channel A is selected by default. You can change this to any active channel.

- Click the **Generate** button to open the [Generate Mask dialog](#):

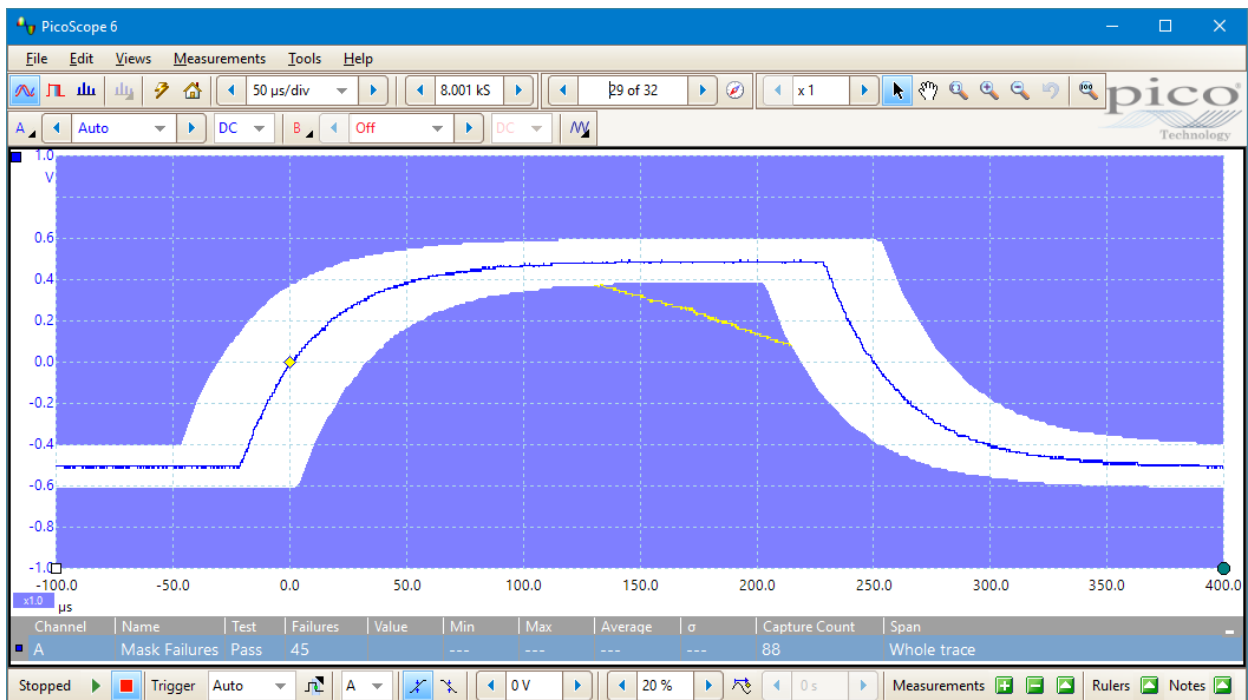


5. For now, accept the default settings and click **Generate**. Make sure your mask is highlighted in the [Mask Library dialog](#), click **OK** and return to the scope view:



You now have a mask drawn around the original waveform.

6. PicoScope stops capturing when you enter the [Mask Library dialog](#), so click the [Go button](#) to restart. If any captured waveform fails to fit inside the mask, the offending parts are drawn in a contrasting color. The [Measurements table](#) shows the number of failures:

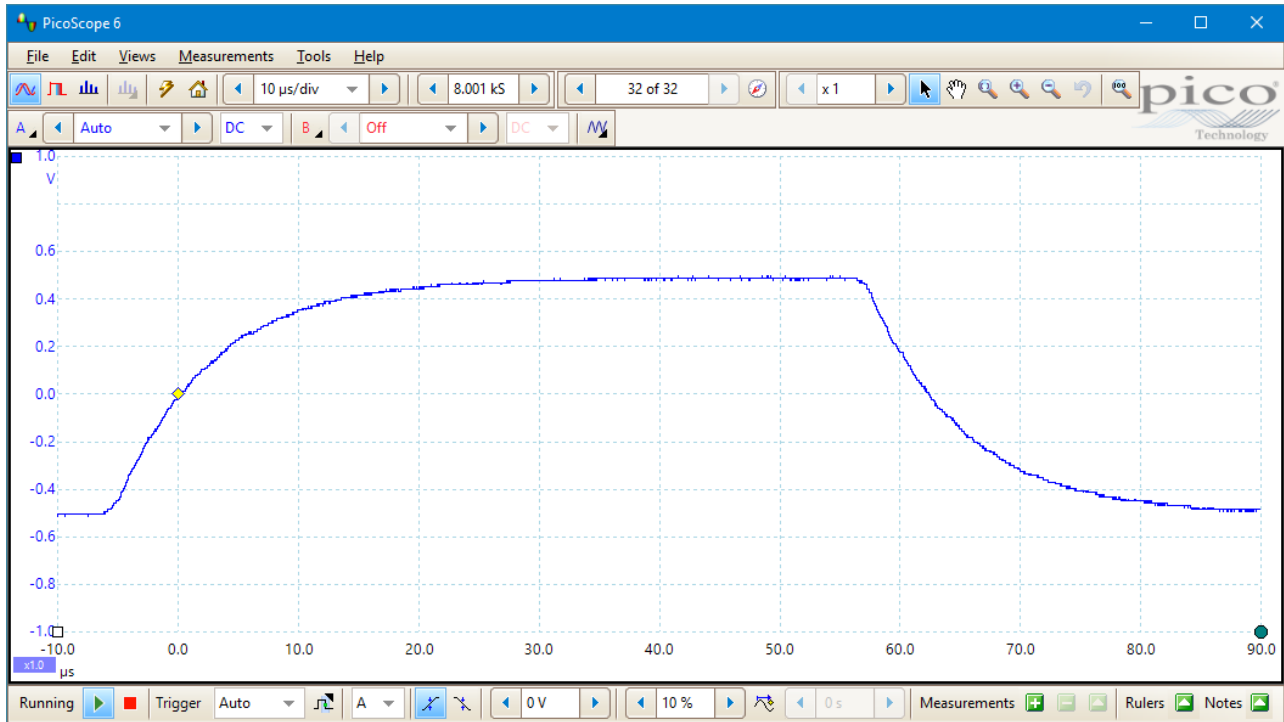


7. You now have a functioning mask limit test. Please refer to [Mask limit testing](#) for information on editing, importing and exporting masks. It is also possible to set up a mask limit test on a [spectrum](#) or [XY](#) view.

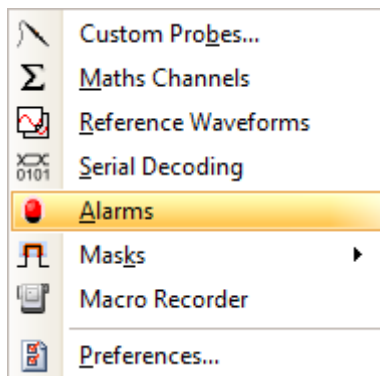
## 7.9 How to save on trigger

**Save-on-trigger** is just one of a number of functions that are possible with the [Alarms](#) feature.

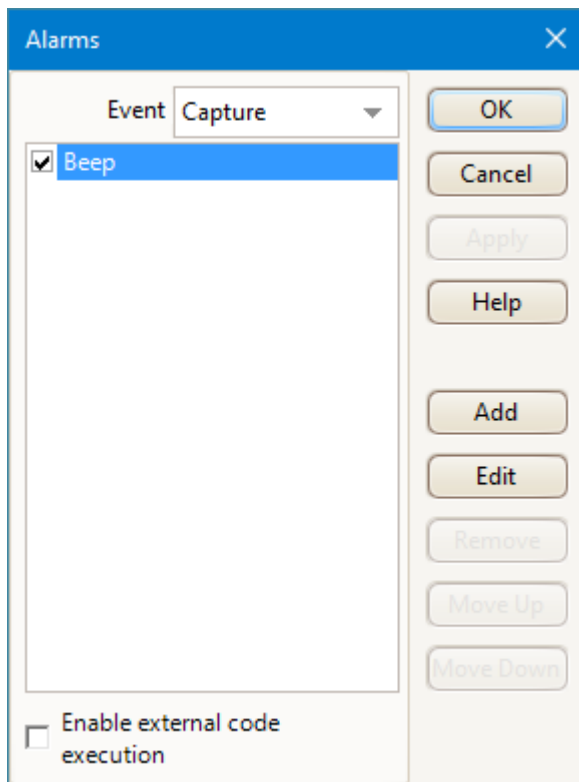
1. Set up PicoScope to display your waveform, and enable triggering:



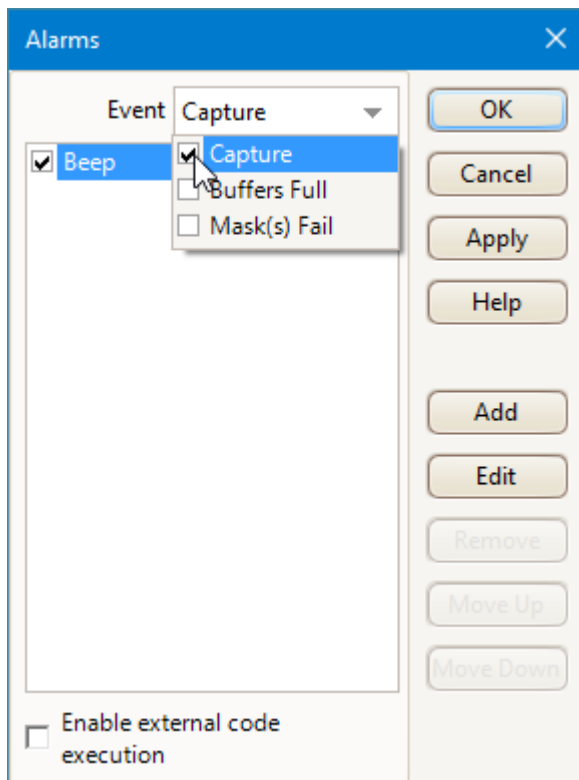
2. Select the [Tools](#) > [Alarms](#) command:



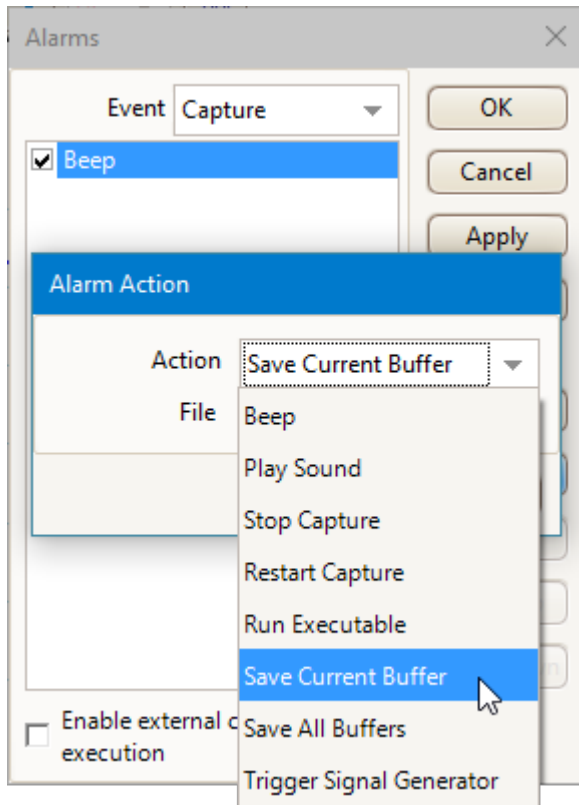
3. The [Alarms dialog](#) will open:




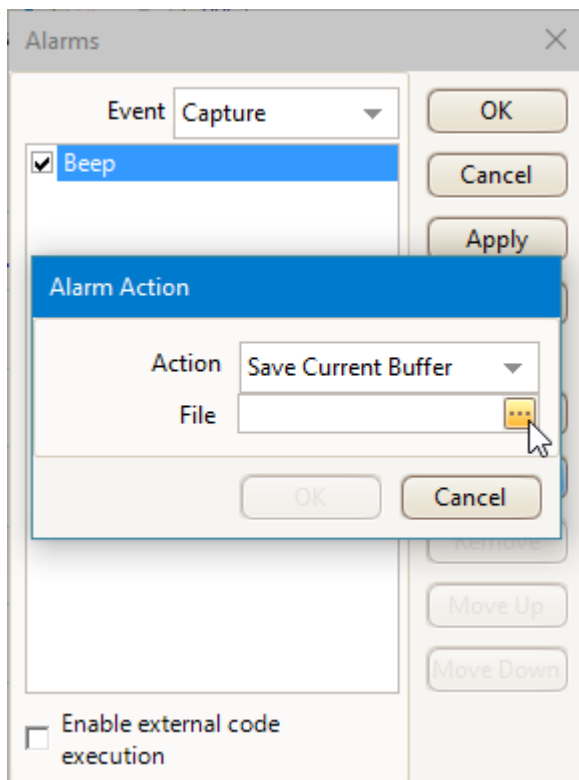
4. Set **Event** to **Capture**, making sure that you have checked the box:



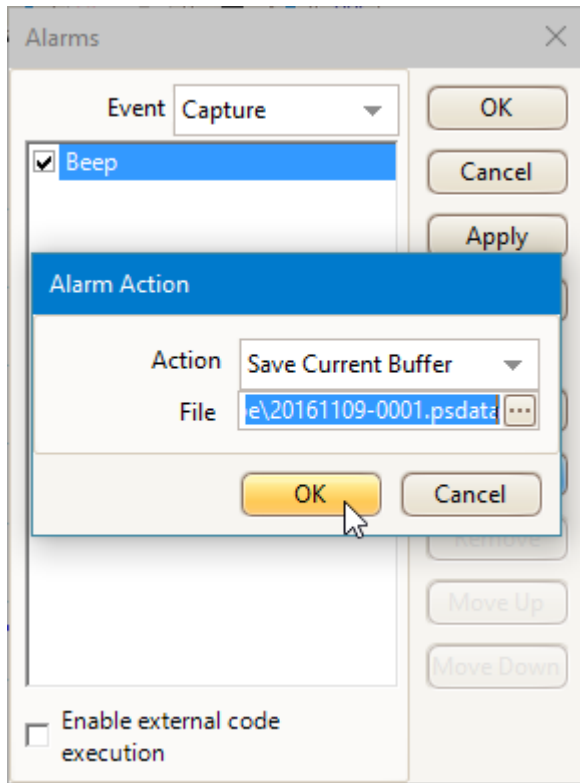
5. Select the first item in the **Actions** list, click **Edit**, and change **Action** to **Save Current Buffer**:



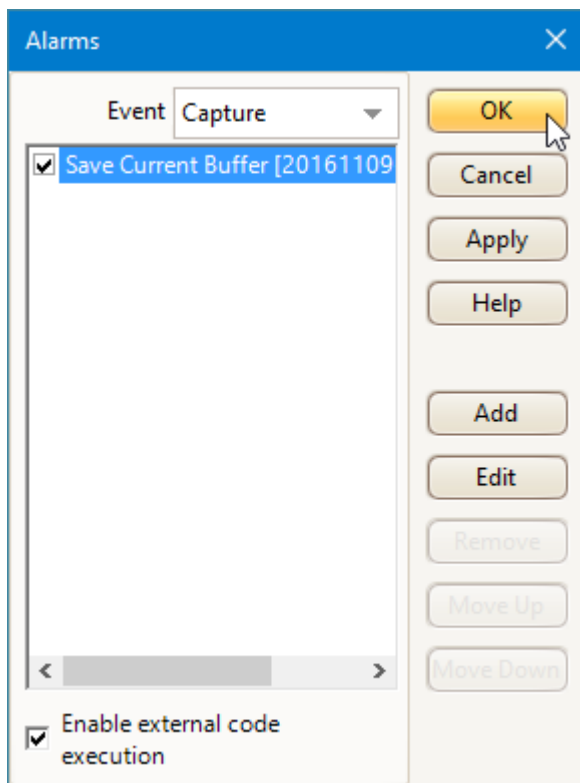
6. Click the  button to the right of the **File** box, browse to the location you wish to save to, enter the file name and click **Save**.



- 7. Click **OK** in the **Alarm Action** dialog:



- 8. Make sure that both the **Save Current Buffer** check box and the **Enable external code execution** check boxes are set:



9. Click **OK**. PicoScope will now save a file on every trigger event.
10. Switch off the alarm when you have finished using it, to avoid creating unwanted files.

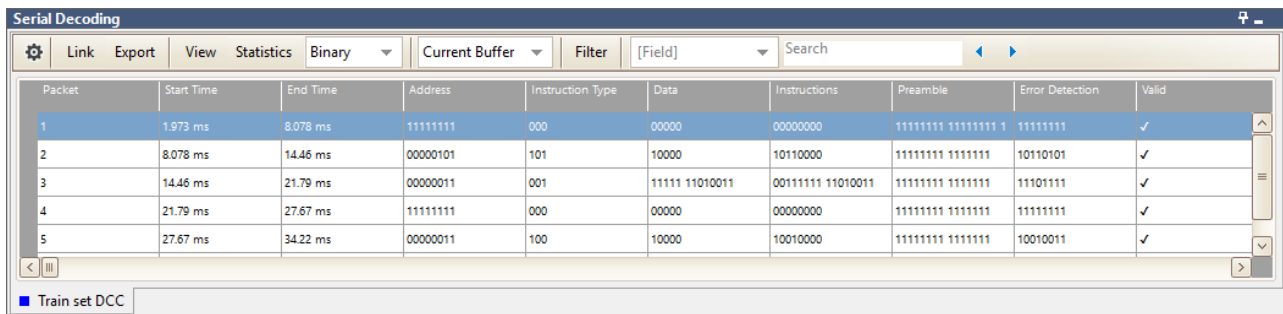


## 7.10 How to create a link file

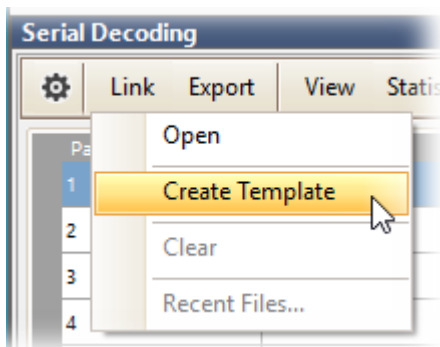
When you perform [Serial decoding](#), the [Table view](#) uses a **link file** to convert numeric data to text. This is useful for displaying decoded packets in a human-readable format. The example below shows the creation of a link file for the [DCC bus](#) used by a digital train set, but although different protocols have different fields, these steps will work for any serial protocol supported by the PicoScope software.

The [Serial protocols](#) section lists which data fields each serial protocol uses.

1. Use the [Serial Decoding dialog](#) to select the required serial protocol and create a serial decoder. Make sure the **Table** view is enabled.
2. PicoScope will show the decoded data in numeric format:



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



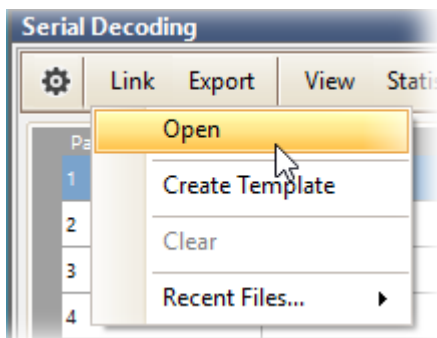
4. A **Save As** dialog will appear. Choose the file type from the **Save As Type** box (.ods for OpenDocument Spreadsheet format, or .csv for comma-separated text format) and save the link file to a convenient location.
5. Open the link file with an appropriate text or spreadsheet editor, and add number/string pairs appropriate to the target system.

G	H	I	J	K	L	M	N
Address	Address Description	Instruction Type	Instruction Type Description	Data	Data Description	Instructions	Instructions Description
00000011	Ivor	000	Decoder and Consist Control Instruction			10110000	no function activated
		001	Advanced Operation Instructions			10110001	F5 on
		010	Speed and Direction Instruction for reverse operation			10110010	F6 on
		011	Speed and Direction Instruction for forward operation			10110011	F5, F6 on
		100	Function Group One Instruction			10110100	F7 on
		101	Function Group Two Instruction			10110101	F5, F7 on
		110	Future Expansion			10110110	F6, F7 on
		111	Configuration Variable Access Instruction			10110111	F5, F6, F7 on

In this example we have specified that the string "Ivor" will appear in place of the value "0000011" in the **Address** field, and so on. The numeric values are treated literally, so if the **Change numeric display format** control (next to the **Statistics** button) in the data window is set to **Hex**, the link file must contain the string "0D" to match the hexadecimal value 0D. If you change the numeric display format to **Decimal**, for example, you will need to create a new link file containing decimal values.

Note: spreadsheet editors such as Microsoft Excel will often delete any zeroes before a number, so that "0000011" becomes "11". Make sure you format all numbers in the link file as text and preserve all preceding zeroes.

4. Save the link file.
5. Click the **Link** button again and select **Open**:



6. An **Open** dialog will appear. Select the link file and click **OK**.
7. PicoScope will now replace the numeric data with the text strings that you have defined:

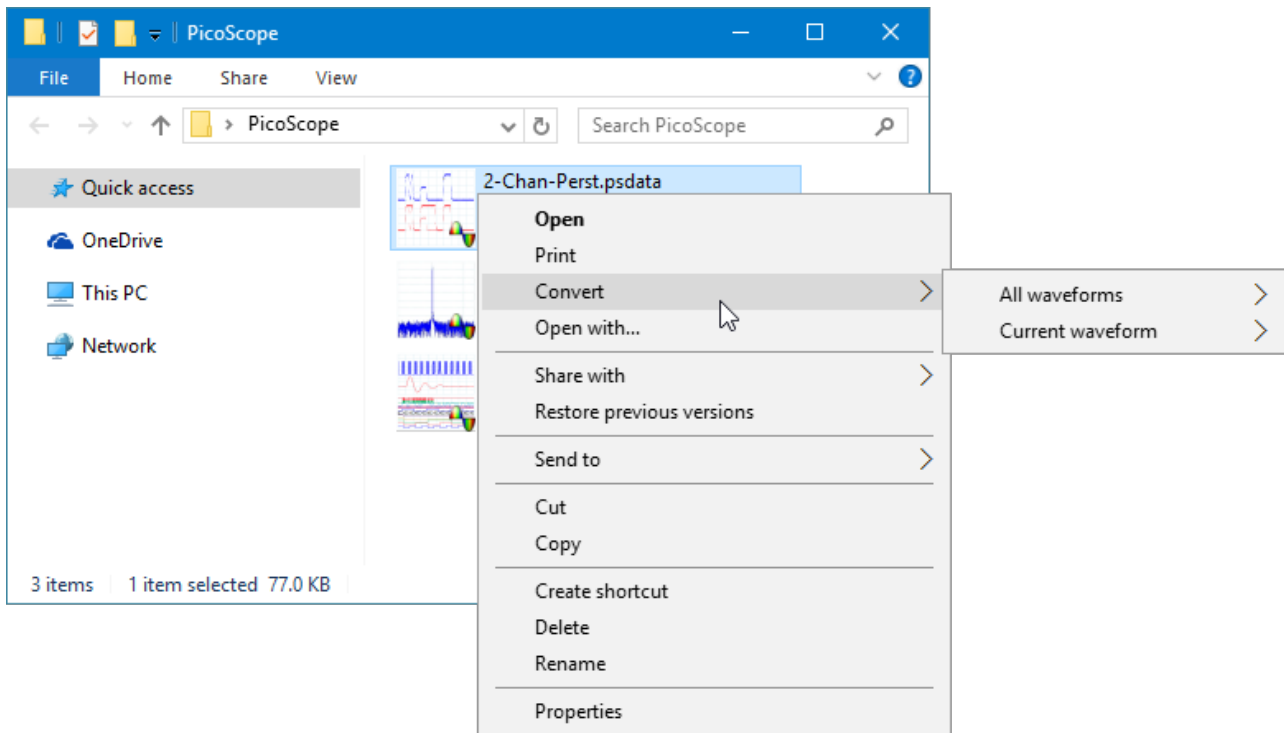
 A screenshot of the 'Serial Decoding' software interface showing the updated data table. The table has columns: Packet, Start Time, End Time, Address, Instruction Type, Data, Instructions, Preamble, Error Detection, and Valid. The data is as follows:
 

Packet	Start Time	End Time	Address	Instruction Type	Data	Instructions	Preamble	Error Detection	Valid
1	1.973 ms	8.078 ms	11111111	Decoder and Consis...	00000	00000000	11111111 11111111 1	11111111	✓
2	8.078 ms	14.46 ms	00000101	Function Group Tw...	10000	no function activated	11111111 11111111	10110101	✓
3	14.46 ms	21.79 ms	Ivor	Advanced Operatio...	11111 11010011	00111111 11010011	11111111 11111111	11101111	✓
4	21.79 ms	27.67 ms	11111111	Decoder and Consis...	00000	00000000	11111111 11111111	11111111	✓
5	27.67 ms	34.22 ms	Ivor	Function Group On...	10000	10010000	11111111 11111111	10010011	✓

## 7.11 How to convert files in Windows File Explorer

You can convert PicoScope data files to other formats for use in other applications, or to different forms of data for use with PicoScope.


The easiest way to do this conversion is through the context menu in **Windows File Explorer**. The context menu is the menu that pops up when you right-click the file icon with the mouse or activate it with the **menu** button on a Windows keyboard. When you install PicoScope, a **Convert** entry is added to the context menu to enable you to convert PicoScope data files.



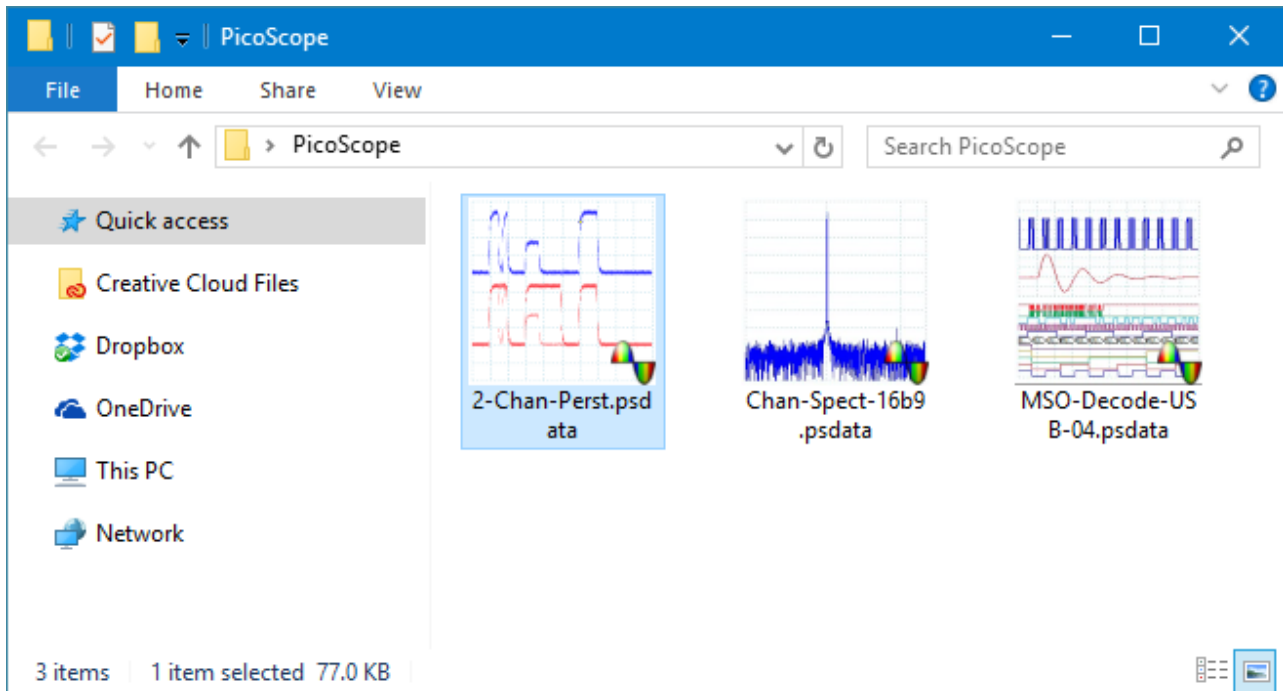
The PicoScope context menu in Windows Explorer

### Converting to PicoScope 6.2.4 format

The example above shows four pre-existing PicoScope data files represented by standard PicoScope icons. PicoScope 6.2.4 introduced a new feature that allows PicoScope data files to appear as waveforms instead of icons. To enable this feature for old data files, you need to convert them to the new format using the context menu of Windows Explorer.


- If PicoScope is running, close it.
- In Windows Explorer, right-click a PicoScope data file.
- Select **Convert > All waveforms > .psdata**. A PicoScope icon  will appear in the Windows notification area while the conversion is in progress.
- PicoScope will ask you to confirm that you wish to overwrite the .psdata file with a new version. Click **Yes**.
- Wait for Windows Explorer to update the display.
- Repeat for all .psdata files.

The .psdata files should now appear as in this picture:



### Converting to other formats

For all of these conversions, you can choose either **All waveforms** or **Current waveform**. A .psdata file can contain either a single waveform or the entire contents of the waveform buffer, which can hold a number of waveforms from successive trigger events. If the .psdata file contains more than one waveform, then you can choose to convert all of them or just the one that was last viewed in PicoScope.

- Right-click a PicoScope data file.
- To convert all waveforms in the file, select **Convert > All waveforms** or **Convert > Current waveform** and then the file format that you require. A PicoScope icon  will appear in the Windows notification area while the conversion is in progress.

### Complex operations

For more complex operations, such as converting all files in a directory, you can run PicoScope in a command window (see [Command-line syntax](#)).

## 8 Reference

This is where you can find detailed information on the operation of PicoScope. The topics covered are:

- [Command-line syntax](#)
- [Flexible power](#)
- [Hydraulic waveforms: Examples and analysis](#)
- [Measurement types](#)
- [Serial protocols](#)
- [Signal generator waveform types](#)
- [Spectrum window functions](#)

This section is also home to the [Device feature table](#) and the [Glossary](#).

### 8.1 Command-line syntax

PicoScope can be run from the Windows command line, allowing you to run tasks manually or under the control of a batch file or other program.

#### To display the GUI

```
PicoScope
```

#### To open a file

```
PicoScope <filename>
```

<filename>                      Specifies a single .psdata or .pssettings file.

Example: PicoScope C:\Temp\source.psdata

#### To display help

```
PicoScope /?
```

Shows help on all command-line options.

#### To convert a psdata file

```
PicoScope /C,/c
```

Converts a psdata file from one format to another. Cannot be used with /p[rint].

Syntax:

```
PicoScope /c[onvert] <names> [/d <names>] /f <format> [/q]
[/b [<n>[:<m>]] | [all]] [/v <viewportname>]
```

<names>                      Specifies a list of one or more directories or psdata files. Wildcards may be used to specify multiple files. If a directory is specified, all psdata files within that directory will be specified. This is a mandatory argument.

/d <names>                      Destination. Default is input filename with new extension.

/f <format>                      Destination format: .csv, .txt, .png, .bmp, .gif, .agif (animated GIF), .psdata, .pssettings, .mat (MATLAB). This is a mandatory argument.

<code>/q</code>	Quiet mode. Do not ask before overwriting files. The default is to prompt.
<code>/b [&lt;n&gt;[:&lt;m&gt;]] all</code>	Waveform number n, waveform range n to m or all waveforms. The default is the current waveform.
<code>/v &lt;viewportname&gt;</code>	View to convert. Default is current view.

Example:

```
PicoScope /c C:\Temp\source.psddata /f png /b 5:9 /v Scope2
```

### To print a view

```
PicoScope /P,/p
```

Prints a view in the psdata file. Cannot be used with `/c[onvert]`.

Syntax:

```
PicoScope /p[rint] <names> [/b [<n>[:<m>]] | all] [/v <viewportname>]
```

`<names>` Specifies a list of one or more directories or .psdata files. Wildcards may be used to specify multiple files. If a directory is specified, all .psdata files within that directory will be specified. This is a mandatory argument.

`/b [<n>[:<m>]]|all` Waveform number n, waveform range n to m or all buffers. Default is current waveform.

`/v <viewportname>` View to convert. Default is current view.

Example:

```
PicoScope /p C:\Temp\source.psddata /b 5:9 /v Scope2
```

### To import notes

```
PicoScope /N,/n
```

Copies text from a specified file to the [Notes area](#).

Syntax:

```
PicoScope /n[otes] <notes filename> <filename>
```

`<notes filename>` Specifies a single text file.

`<filename>` Specifies a single psdata or pssettings file.

Example:

```
PicoScope /n C:\Temp\source.txt C:\Temp\source.psddata
```

**To run an automation command**

```
PicoScope /A, /a
```

Run automation command or macro on an existing instance of PicoScope 6.

Syntax:

```
PicoScope /a[utomation] <command> | <macro>
```

<command>                      Automation command

<macro>                        Path to .psmacro file containing a [macro](#)

Examples:

```
PicoScope /a Run.Pressed=True
```

```
PicoScope /a MyMacro.psmacro
```

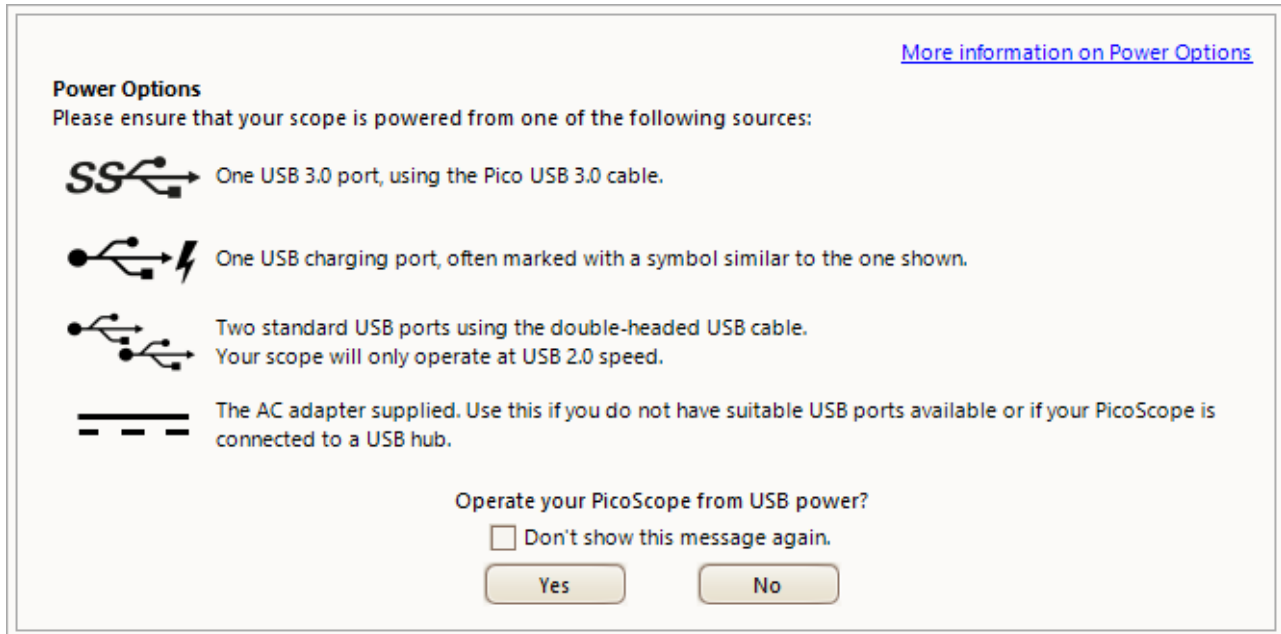
```
PicoScope /a ? (prints a list of automation commands to the console)
```

```
PicoScope /a Measurements? (prints information on the Measurements command)
```


An instance of PicoScope must be running before any automation command is issued.


## 8.2 Flexible power

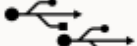
The flexible power system for PicoScope devices provides a choice of power sources. In most cases a straightforward USB connection will be enough to power the scope. If PicoScope needs you to switch to a different powering method, it will display a dialog similar to this:




### The power icons explained

 **USB 3.0 port.** If this symbol appears, you may power the scope from any USB 3.0 port using the USB 3.0 cable provided with your scope.

 **USB charging port.** If this symbol appears, you may power the scope from any USB port capable of supplying 1200 mA of current.

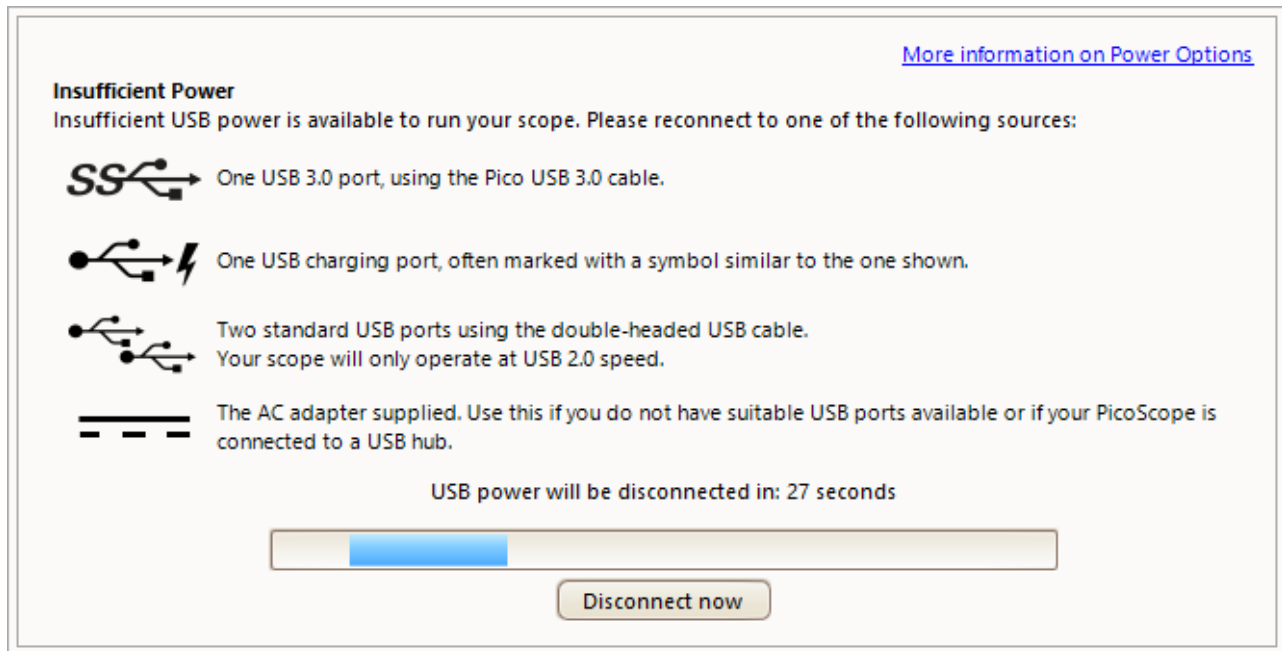
 **Dual USB port power.** If this symbol appears, you may power the scope from any two powered USB ports using the dual-headed USB 2.0 cable supplied with your scope.

 **AC adapter.** Use the adapter supplied and plug it into the **DC IN** socket on your scope. If you power the scope in this way, the **Power Options** dialog will automatically close and the scope will start working immediately.



### Insufficient USB power

If insufficient USB power is available, PicoScope displays this dialog:



Connect one of the power sources listed and try again. If you plug in the AC adapter, the dialog will automatically close and you can continue working.

## 8.3 Hydraulic waveforms: Examples and analysis

Pico's Engine and hydraulics kit allows you to diagnose problems on your engine and hydraulics equipment. You can use it for simple tests of purely electric signals, such as extending and retracting a hydraulic ram, and for full system tests.

The settings and connections you have to use depend on the test you want to perform and the vehicle you wish to test.

On the following pages we have compiled some examples to illustrate what possible waveforms you may see on the screen, and what to look for during the diagnostic process.

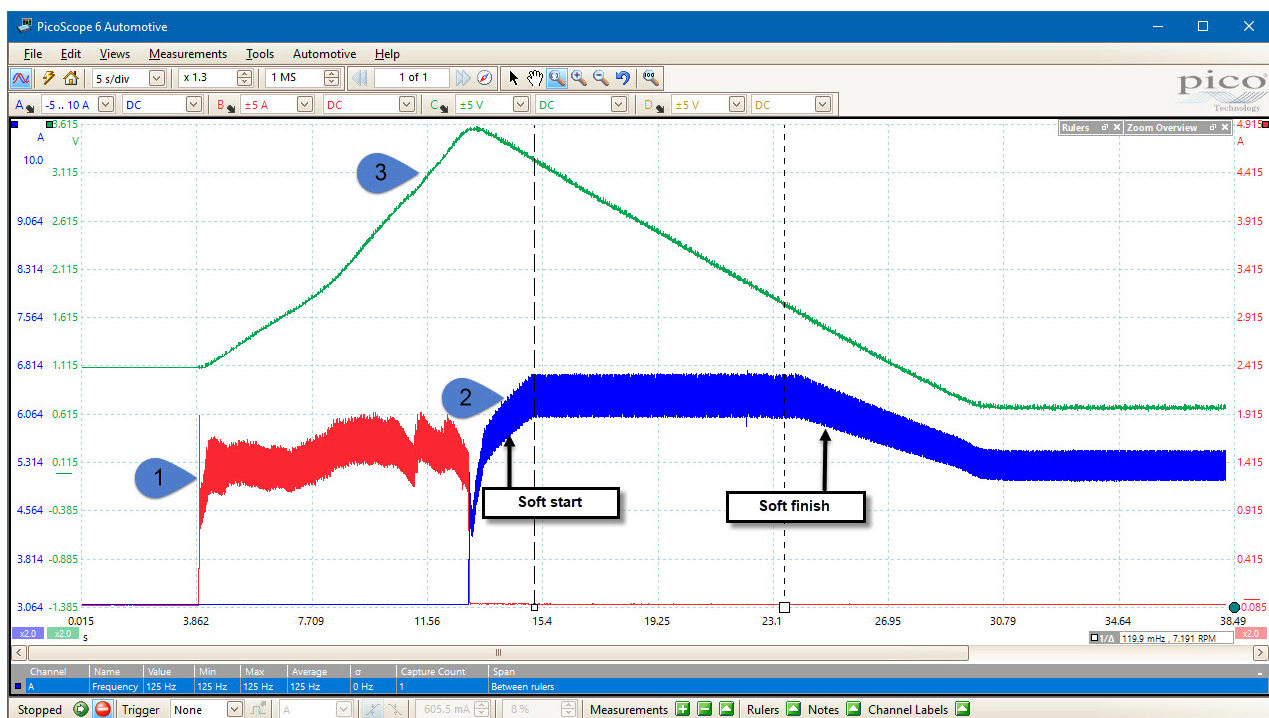
- [Example 1](#) – Hydraulic ram on a tractor's three-point hitch
- [Example 2](#) – Hydraulic system on telehandler
- [Example 3](#) – Hydraulic clutch pack

### 8.3.1 Hydraulic ram on a tractor's three-point hitch

Test: Extending and lowering the hydraulic ram on a tractor's three-point hitch.

#### Example waveform:

- Channel A: Current on lowering solenoid
- Channel B: Current on lifting solenoid
- Channel C: Position sensor



#### Waveform key:

1. Lifting operation
2. Lowering operation
3. Position sensor voltage during lifting and lowering

**Description:**

This example shows you the difference in behavior between lifting and lowering. The lift was under manual control, and you can see that the waveform during this period is quite uneven. Lowering was under the control of the ECU, and here you can see that the waveform displays a soft start and finish.

The control valves used in this example are both pulse width modulated (PWM) dither valves. This means that the average current through them is proportional to the mark-space ratio of the controlling voltage signal, the midpoint current at any instant indicating the degree to which the valve is open or closed. The ripple on the PWM current shows the underlying frequency of the PWM signal (in this case 125 Hz).

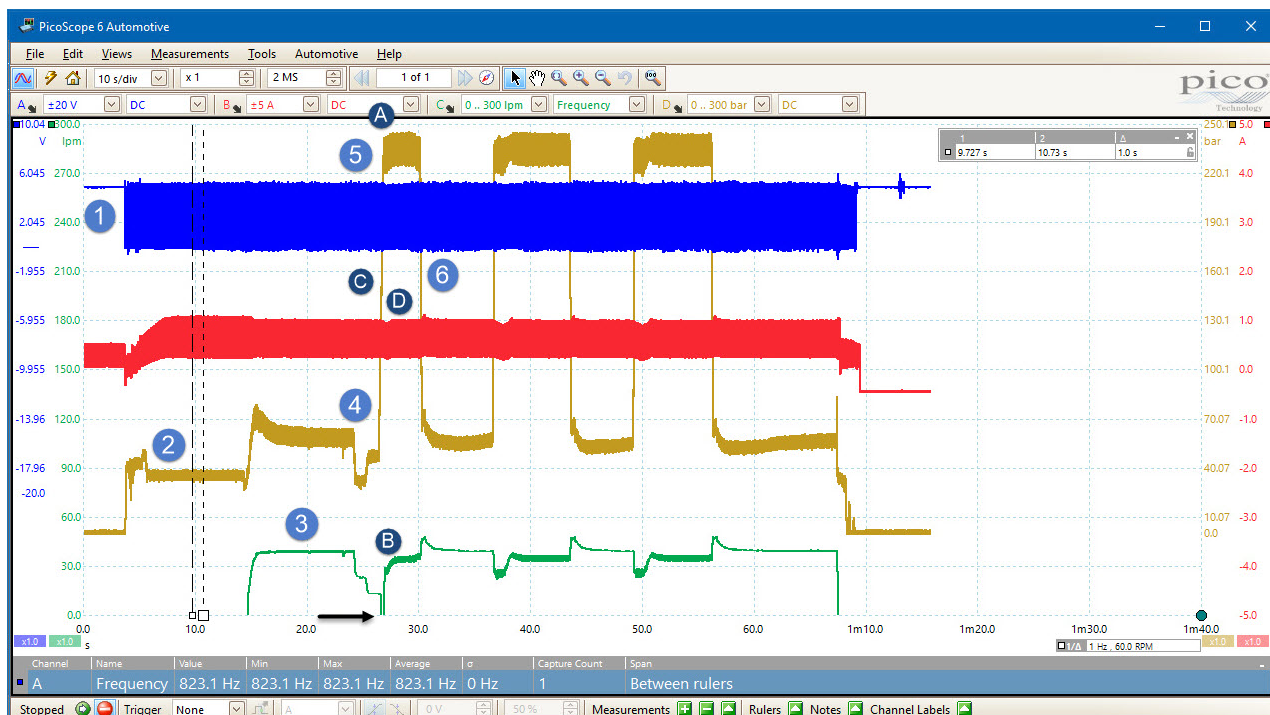
## 8.3.2 Hydraulic system on telehandler

With more advanced system testing, you can explore the operation of the whole system with one capture. In this example, we look at the performance of the hydraulic system on a telehandler using the TA377 300 lpm Webtec flowmeter (not included in the kit).

There is a huge amount of diagnostic information in this capture, showing both the basic operation of the system and some of the feedback controls built into this vehicle. Engine speed can be inferred from the crank speed sensor by using a measurement showing the frequency of the crank signal between two closely spaced rulers. As the crank sensor on this engine is a 60-tooth wheel with two teeth missing, engine RPM is almost the same as the crank sensor frequency (RPM is 3.5% higher). The vehicle was sitting with the boom raised and the forks pointing down.

**Example waveform:**

- Channel A: Crank speed sensor
- Channel B: Main control valve solenoid
- Channel C: Flow through aux (auxiliary) line
- Channel D: Hydraulic pressure at control valve output



**Waveform Key:**

1. The engine is started – the crank sensor speeds up to its tick-over value and the hydraulic system pressurizes to its standby pressure. The dither control on the main valve starts at a minimum value and builds slowly to avoid stalling the engine.
2. The waveform shows a steady state, with the engine idling and only the standby flow in the internal hydraulic system to provide lubrication for the main hydraulic components.  
Note: at this time there is no significant hydraulic flow in the aux line.
3. The aux system is turned on with the full flow rate of 30 lpm going to tank. The engine speed remains constant. The hydraulic pressure at the pump has increased to cope with the flow through the aux line.
4. The forks are lifted to apply some load to the system. As you can see, the flow to the ram has priority over the aux line and the flow in the aux line drops to near zero (the flow meter used had a minimum flow reading of around 15 lpm – below this value the scope trace will disappear off the bottom of the screen). As the forks start to lift, you can see the line pressure drop to near standby, as there is no real load on the forks and the standby pressure is enough to overcome gravity. As the ram reaches its end stop, you can see a small pressure step as the soft-finish function of the ram cuts in.
5. With the forks fully up, the operator applies and maintains full pressure to the system to generate maximum load. As this occurs, several things happen:
  - A. First, the pressure goes to system maximum.
  - B. The flow is switched back to the aux line as the forks cylinder no longer requires a significant flow.
  - C. The engine slows down as the load comes on due to the increased power demand. If you calculate the engine speed as the system goes to max load you will see the approximate engine speed drop from around 820 RPM to 720 RPM. Once the revs have dropped by the preset amount, the engine's anti-stall system cuts in to keep it running.
  - D. The pulse width modulated (PWM) main control valve is backed off a small amount to unload the engine. You can see this in the way the max current (Channel B) dips slightly. This unloading, along with the engine's electronic controls, allows the ECU to return the engine to the correct number of revs. An additional effect of the pressure backing off is the rounded nature of the initial corner as the pump goes to maximum pressure.
6. The operator stops applying pressure to the forks cylinder. Although the forks do not actually move, you can see the pressure decay. When this happens, the load on the engine is removed and the engine starts to speed up. The engine ECU uses the hydraulic system as a brake and opens the main control valve slightly to apply more load to the engine, bringing its speed back to the correct value as quickly as possible.

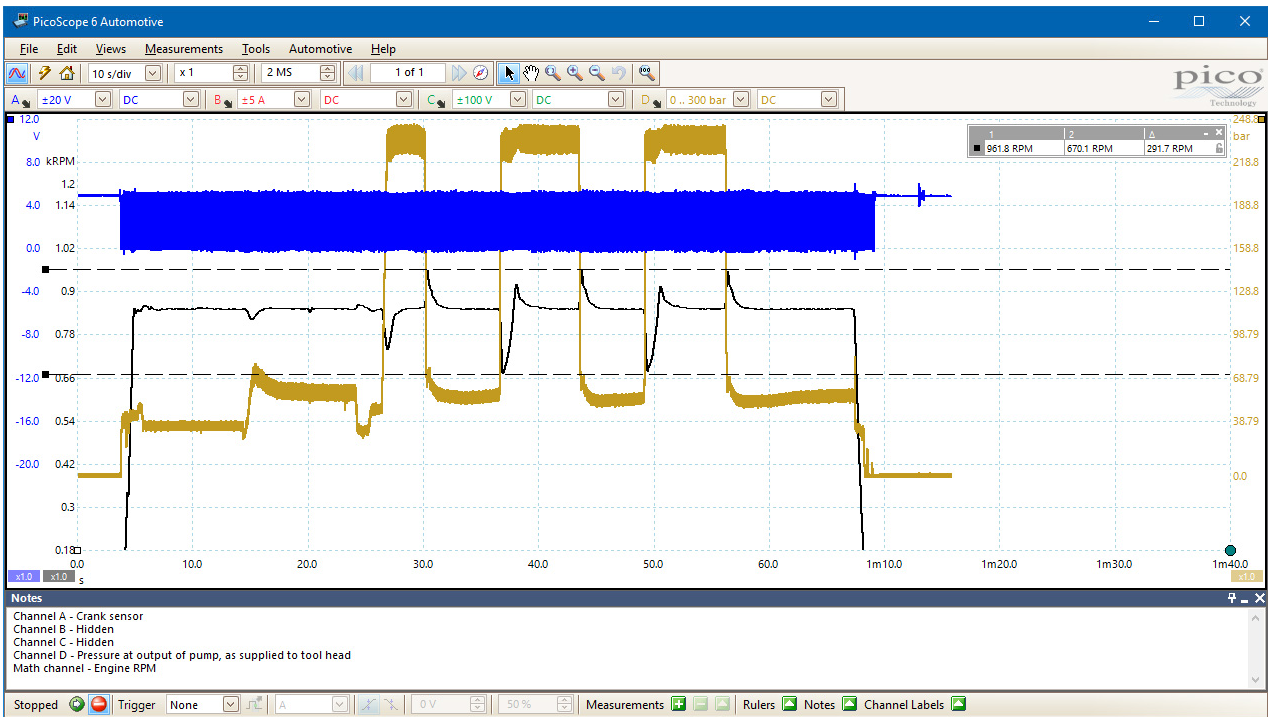
**Description:**

From this simple test, we can see how the system's electronic control is applied to manage the operation of the complete system.

It is important to understand that a simple-sounding fault such as "the engine stalls when a heavy hydraulic load is applied" can be down to problems with either the engine or the hydraulic system. Having the ability to directly relate the hydraulic system's performance to engine signals such as injectors or the crank sensor can show the technician not only what is happening, but also what the ECU expects to be happening, hence highlighting an issue.

In PicoScope 6 you can set up a math channel to plot the engine RPM as it varies. This gives you a graphical view of the behavior of the engine anti-stall system. In the example below we are using a lowpass filter with a 2 Hz cut-off in order to remove the peaks that come from the missing teeth in the crank sensor reluctor wheel.

Example math channel formula:  $1.034 * \text{LowPass}(\text{freq}(A), 2)$ , with **Units** named as RPM and **Max Range** set at 1200. 1.034 is the correction factor to convert from 58 pulses per second from the crank sensor to RPM.

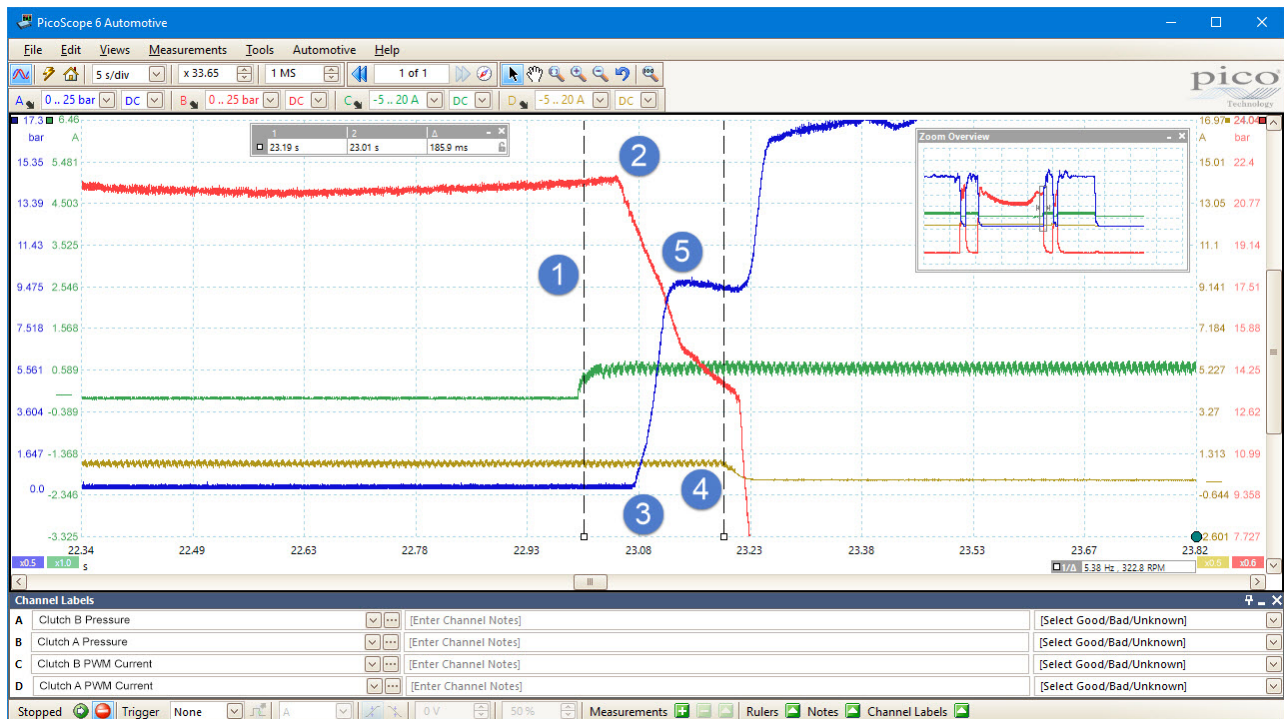


### 8.3.3 Hydraulic clutch pack

This example is from a hydraulic clutch pack changing from Clutch A to Clutch B. It illustrates the real-world delays caused by the time it takes for the valve spools to move, oil to flow, and pressure to decay. Note that the entire changeover only takes 250 ms, and there is less than 200 ms between turning one solenoid on and turning the second off.

#### Example waveform:

- Channel A: Clutch B pressure
- Channel B: Clutch A pressure
- Channel C: Clutch B current
- Channel D: Clutch A current



#### Waveform key:

1. The solenoid in Clutch B is activated
2. The pressure in Clutch A starts to decrease
3. The pressure in Clutch B starts to increase
4. The solenoid in Clutch A is deactivated
5. The pressure in Clutch B plateaus until the clutch is deactivated. The pressure then rises to its full normal operating pressure

#### Description:

We can see that Clutch A is already engaged and that the pulse width modulated (PWM) signal to its control solenoid is active. Note that the pressure traces for the two clutches have different vertical scales.

When the clutch change is activated, first the power is turned on to Clutch B. For a short period of time nothing appears to happen. In fact, the spool moves within the directional control valve and oil starts to flow. Once the valve spool has moved far enough, we can see the pressure in Clutch A begin to decrease. With this clutch pack design, there is an internal gallery between the two clutches to speed operation. Initially the pressure decay is caused by oil flowing through the internal gallery and directly into Clutch B. As the hydraulic pump on this system (like many others) cannot supply the volume of oil required to fill the second clutch while already supplying oil to the first one, the system is designed to start filling the second clutch by using the oil draining from the first one.

When the second clutch comes on load (marker 5), the pressure stabilizes while the clutch plates move. During this period the pump is still supplying oil to both clutches, but the internal gallery is making sure the pressure in the two units equalizes as quickly as possible and that the pressure in Clutch A decreases as pressure builds in Clutch B. Note that the system is configured to make this crossover time, with both clutches driving, as small as possible to minimize clutch slippage and wear. The behavior of the system during this crossover period is critical to the smoothness of the changeover. If Clutch A fully disengages before Clutch B is ready, the vehicle will decelerate suddenly with a noticeable jerk. If the clutch overlap is too long, you will get accelerated wear of the clutch plates and excessive heating of the hydraulic oil.

When the solenoid in Clutch A is deactivated we see a small time delay while the valve spool moves to cut off flow to Clutch A. Once this is completed you can see that the pressure in Clutch A drops rapidly to tank pressure and that the pressure in Clutch B builds to normal operating pressure. The system is now operating normally with Clutch B fully engaged and Clutch A off.

## 8.4 Measurement types

The [Add/Edit Measurement dialog](#) presents a choice of measurements that PicoScope can calculate for the selected view. This list varies depending on whether you are using a [scope view](#) or a [spectrum view](#), as some are based on the time domain and others are based on the frequency domain.

The topics in this section explain the different choices available for [Scope measurements](#) and [Spectrum measurements](#).

### 8.4.1 Scope measurements

The measurements available in a [scope view](#) are:

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

**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.

**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.

**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. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

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

**Fall Time.** The time the signal takes to fall from the upper threshold to the lower threshold. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**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.

**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.

**Maximum.** The highest level reached by the signal.

**Minimum.** The lowest level reached by the signal.

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

**Rise Time.** The time the signal takes to rise from the lower threshold to the upper threshold. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**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. Click the **Advanced** button in the **Add Measurement** or **Edit Measurement** dialog to specify the signal level thresholds for the measurement.

**True RMS.** The root mean square (RMS) value of the waveform, including the DC component.

**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.

## 8.4.2 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  $f_1$  and  $f_2$ , the two second-order distortion signals will be found at frequencies:  $f_3 = (f_1 + f_2)$  and  $f_4 = (f_1 - f_2)$ .

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

$F_3$  and  $F_4$  are the amplitudes of the two second-order distortion terms (at frequencies  $f_3$  and  $f_4$  defined above)

and

$F_1$  and  $F_2$  are the amplitudes of the input tones (at frequencies  $f_1$  and  $f_2$ , as marked by the [frequency rulers](#) in the spectrum window).

For reference, the third-order terms are at frequencies  $(2F_1 + F_2)$ ,  $(2F_1 - F_2)$ ,  $(F_1 + 2F_2)$  and  $(F_1 - 2F_2)$ .

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.

## 8.5 Serial protocols

The [serial decoding](#) feature of PicoScope understands a range of serial protocols. Each one will show different sets of data, as detailed in the following sections. To see all of these data fields, make sure **Table** is selected in the [Decoder settings](#). You can decode the following serial protocols in PicoScope 6:

[1-Wire](#)

[ARINC 429](#)

[BroadR-Reach](#)

[CAN](#)

[CAN FD](#)

[DALI](#)

[DCC](#)

[DMX512](#)

[Ethernet 10BASE-T](#)

[Fast Ethernet 100BASE-TX](#)

[FlexRay](#)

[I<sup>2</sup>C](#)

[I<sup>2</sup>S](#)

[LIN](#)

[Manchester](#)

[Modbus ASCII](#)

[Modbus RTU](#)

[PS/2](#)

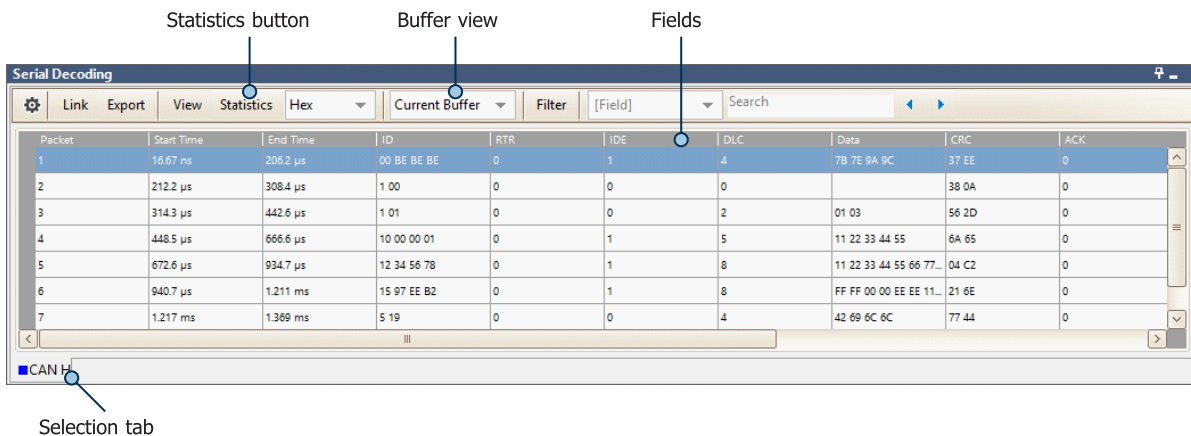
[SENT Fast](#)

[SENT Slow](#)

[SPI](#)

[UART \(RS-232, RS-422, RS-485\)](#)

[USB](#)



There are some fields that every protocol will display:

**Packet** Serial number of the frame (decimal).

If **All Buffers** is selected, this is counted from the start of the selected waveform.  
If **Current Buffer** is selected, this is counted from the start of the first waveform in the current waveform buffer.

**Start Time** Value of the PicoScope collection time at the start of the frame

**End Time** Value of the PicoScope collection time at the end of the frame

When you click the **Statistics** button, a further four columns appear:

**Min Voltage** Minimum voltage

**Max Voltage** Maximum voltage

**Voltage Pk-Pk** Voltage range of the signal, from **Max Voltage** to **Min Voltage**

**Packet Time** The duration of the packet, from **Start Time** to **End Time**

## 8.5.1 1-Wire protocol

1-Wire is a serial protocol originally developed by the Dallas Semiconductor Corporation. It provides low-speed data and power to the target device over the same wire.

For more information on the 1-Wire protocol, see [PicoScope from A to Z](#).

<b>Reset Pulse</b>	Duration of the reset pulse generated by the master.
<b>Presence Pulse</b>	Duration of the presence pulse generated by the slave.
<b>Command</b>	The numerical value of the command.
<b>Command Code</b>	The name of the command code.
<b>Data</b>	The data content of the packet.

## 8.5.2 ARINC 429 protocol

ARINC 429, also known as the Mark 33 DITS specification, was developed to provide interchangeability and interoperability of line replaceable units (LRUs) in commercial aircraft. ARINC 429 defines the physical and electrical interfaces of a two-wire data bus and the data protocol to support an aircraft's avionics local area network.

For more information on ARINC 429, see [PicoScope from A to Z](#).

<b>BNR Data</b>	Binary Data. Formatted according to the SSM bits.
<b>BNR STATUS MATRIX</b>	The meaning of the SSM field decoded according to the Label.
<b>Label</b>	An 8-bit value indicating the data type.
<b>SDI</b>	Source/Destination Identifier.
<b>Data</b>	The data content of the word.
<b>SSM</b>	Sign/Status Matrix. The meaning of this 2-bit field depends on the Label.
<b>Parity Bit</b>	The error-correction bit at the end of the word.
<b>Parity Valid</b>	Parity result calculated by PicoScope.
<b>Valid</b>	Confirms whether the packet is valid.

## 8.5.3 BroadR-Reach

BroadR-Reach (IEEE 802.3 100BASE-T1) is an Ethernet physical layer standard designed for use in automotive networking applications. It allows multiple in-vehicle systems to simultaneously access information over low-cost unshielded single twisted pair cable. The protocol delivers relatively high-speed data, 100 Mbps, with low connectivity costs and cabling weight.

[BroadR-Reach info on picotech.com](#)

<b>Preamble</b>	Synchronization pattern to indicate start of transmission.
<b>MAC Destination</b>	Address of receiving device.
<b>MAC Source</b>	Address of transmitting device.
<b>Ether Type</b>	Protocol identifier (TPID).
<b>TCI</b>	Control information. Consists of Priority (PCP), Drop Eligible or Canonical Form Indicator (DEI or CFI), and Identifier (VID).
<b>Payload</b>	Payload data. 42 to 1500 bytes.
<b>CRC</b>	Checksum.
<b>CRC Pass</b>	Result of checksum comparison.

BroadR-Reach decoding requires PicoScope release 6.14.10 or later.

## 8.5.4 CAN protocol

CAN (Controller Area Network) is a serial protocol used in automotive and industrial machinery to allow microcontrollers to communicate with each other. The standard was originally developed in 1983 by Robert Bosch GmbH. It typically uses differential signaling (with signals named CAN H and CAN L) to increase noise immunity.

For more information on the CAN protocol, see [PicoScope from A to Z](#).

<b>ID</b>	Identifier (hexadecimal). This should be unique for a specified data type. Base frames have an 11-bit identifier, and extended frames have a 29-bit identifier.
<b>RTR</b>	Remote transmission request.
<b>IDE</b>	ID Extension bit. 0 for standard format packet, 1 for extended format.
<b>DLC</b>	Data length code. Indicates the number of bytes of data.
<b>Data</b>	The data content of the packet.
<b>CRC</b>	A cyclic redundancy check of the data.
<b>ACK</b>	Node asserts this bit to acknowledge receipt.
<b>CRC Valid</b>	The result of PicoScope's CRC.

## 8.5.5 CAN FD protocol

The CAN FD (Controller Area Network with Flexible Data Rate) serial protocol is an enhancement of the Classical [CAN protocol](#). Like the CAN protocol, it is used in automotive and industrial machinery to allow microcontrollers to communicate with each other, and typically uses differential signaling (with signals named CAN H and CAN L) to increase noise immunity.

ISO 11898-1:2015 specifies both the Classical CAN and CAN FD protocols. The Classical CAN protocol allows bit rates up to 1 Mb/s and payloads up to 8 bytes per frame. The CAN FD protocol allows bit rates higher than 1 Mb/s and payloads longer than 8 bytes per frame.

<b>BRS</b>	Bit rate switch. Indicates whether or not the bit rate is switched inside the FD frame.
<b>ESI</b>	Error state indicator.
<b>Stuff Count</b>	The number of stuff bits in a frame before the CRC field, not including fixed stuff bits.
<b>Parity</b>	Stuff count parity check. The receiver checks whether the stuff count field matches its own count.
<b>ID</b>	Identifier (hexadecimal). This should be unique for a specified data type. Base frames have an 11-bit identifier, and extended frames have a 29-bit identifier.
<b>RTR</b>	Remote transmission request.
<b>IDE</b>	ID Extension bit. 0 for standard format packet, 1 for extended format.
<b>DLC</b>	Data length code. Indicates the number of bytes of data.
<b>Data</b>	The data content of the packet.
<b>CRC</b>	A cyclic redundancy check of the data.
<b>ACK</b>	Node asserts this bit to acknowledge receipt.
<b>CRC Valid</b>	The result of PicoScope's CRC.

## 8.5.6 DALI protocol

DALI (Digital Addressable Lighting Interface) is a serial protocol used for the control of lighting fixtures in buildings. The signaling system is a form of Manchester encoding.

For more information on DALI, see [PicoScope from A to Z](#).

<b>Frame</b>	The direction of the frame: <b>Forward:</b> from Control Unit to Ballast. <b>Backward:</b> from Ballast to Control Unit.
<b>Start Bit</b>	A single <b>1</b> to start the frame.
<b>Y</b>	<b>0:</b> short address <b>1:</b> group address or broadcast
<b>Address</b>	An 8-bit value. Only in <b>Forward</b> frames.
<b>Selector Bit</b>	The last bit of the <b>Address</b> . Indicates the meaning of the following byte. <b>0:</b> following byte is direct arc power level <b>1:</b> following byte is command
<b>Data</b>	An 8-bit value. Meaning depends on <b>Selector Bit</b> .
<b>Address Type</b>	Decoded from the <b>Address</b> : <b>Short</b> if first bit of <b>Address</b> is <b>0</b> <b>Group</b> if first three bits of <b>Address</b> are <b>100</b> <b>Broadcast</b> if first seven bits of <b>Address</b> are <b>1111111</b> <b>Special Command</b> if <b>Selector Bit</b> is <b>1</b>

<b>Control Command</b>	Decoded only if <b>Selector Bit</b> is <b>0</b> .
<b>Special Command</b>	Decoded only if <b>Selector Bit</b> is <b>1</b> .
<b>Valid</b>	<b>Check mark:</b> valid frame <b>X:</b> invalid frame

## 8.5.7 DCC protocol

DCC (Digital Command Control) is a serial protocol used for the control of digital model railway systems. It allows users to control multiple locomotives and accessories independently.

For more information on DCC, see [PicoScope from A to Z](#).

<b>Address</b>	Shown for address packets.
<b>Instructions</b>	Commands sent to multi-function digital decoders. The First three bits determine an instruction type, and the rest describe the actions within this type.
<b>Preamble</b>	String of ones indicating start of packet.
<b>Error Detection</b>	Error correction byte.
<b>Valid</b>	The results of the error check.
<b>Address Partition</b>	Assigned based on the type of decoder. Specified in the DCC standard.
<b>Instruction Type</b>	The first three bits of the instruction bytes determining the instruction type, as specified in the DCC standard.
<b>Data</b>	The rest of the instruction bits determining a specific action for a corresponding instruction type (given in Instruction Type Field).

## 8.5.8 DMX512 protocol

DMX512 (Digital Multiplex) is a serial protocol designed for the control of stage lighting and effects.

<b>Slot Count</b>	Number of channels.
<b>Slot 0 Value</b>	Also known as the start code.
<b>Slot Range</b>	Numbers of the first and last slots in this packet.
<b>Slot Values</b>	An 8-bit value for each slot.
<b>Active Channel Count</b>	The number of channels which are above 0.

## 8.5.9 Ethernet 10BASE-T protocol

Ethernet 10BASE-T is a serial protocol supporting Ethernet speeds of up to 10 Mbit/s.

<b>Preamble</b>	Ethernet preamble (PRE).
<b>Start Frame Delimiter</b>	A fixed bit pattern of 10101011.
<b>Destination MAC</b>	Destination address.
<b>Source MAC</b>	Source address.
<b>Type</b>	Length/Type. If $\leq 1500$ , the number of LLC bytes. If $> 1536$ , the packet type.
<b>Data</b>	The data content of the packet.
<b>Frame Check Sequence</b>	CRC value used for error detection.
<b>Ethertype</b>	Meaning of the Length/Type field if it represents a Type.
<b>CRC Pass</b>	Result of the cyclic redundancy check.

## 8.5.10 Fast Ethernet 100BASE-TX protocol

Ethernet 100BASE-TX is a serial protocol supporting Ethernet speeds of up to 100 Mbit/s.

<b>Preamble</b>	Ethernet preamble (PRE).
<b>Start Frame Delimiter</b>	A fixed bit pattern of 10101011.
<b>Destination MAC</b>	Destination address.
<b>Source MAC</b>	Source address.
<b>Type</b>	Length/Type. If $\leq 1500$ , the number of LLC bytes. If $> 1536$ , the packet type.
<b>Data</b>	The data content of the packet.
<b>Frame Check Sequence</b>	CRC value used for error detection.
<b>Ethertype</b>	Meaning of the Length/Type field if it represents a Type.
<b>CRC Pass</b>	Result of the cyclic redundancy check.



## 8.5.11 FlexRay protocol

FlexRay is an automotive network communications protocol for high-speed data. It is designed to be fault-tolerant.

<b>Header</b>	Header segment.
<b>Payload</b>	Payload segment.
<b>Trailer</b>	Trailer segment.
<b>Reserved Bit</b>	Undefined.
<b>Payload Preamble Indicator</b>	Indicates the presence of a network management vector or message ID.
<b>Null Frame Indicator</b>	Indicates that the payload contains no useful payload data.
<b>Sync Frame Indicator</b>	Indicates whether the frame is a synchronization frame.
<b>Startup Frame Indicator</b>	Indicates that this is a clock startup frame.
<b>Frame ID</b>	Defines the slot in which the frame is transmitted.
<b>Payload Length</b>	The number of payload bytes divided by 2.
<b>Header CRC</b>	Cyclic redundancy check of data from the sync frame indicator to the payload length.
<b>Cycle Count</b>	The value of the cycle counter.
<b>Header CRC Pass</b>	Result of Header cyclic redundancy check test.
<b>Frame CRC A Pass</b>	FlexRay frame CRC calculation algorithm - Channel A.
<b>Frame CRC B Pass</b>	FlexRay frame CRC calculation algorithm - Channel B.

## 8.5.12 I<sup>2</sup>C protocol

I<sup>2</sup>C (Inter-Integrated Circuit) was developed by Philips in the 1980s as a low-cost way to connect peripheral chips in consumer electronics products, and is widely adopted in embedded systems. Many devices, including temperature sensors, control panels, microcontrollers, and single board prototyping systems incorporate multiple I<sup>2</sup>C interfaces for easy chip-to-chip and inter-module communications. It uses two signals: clock (SCL) and data (SDA).

For more information on I<sup>2</sup>C, see [PicoScope from A to Z](#).

<b>Address Bits</b>	Address size.
<b>Address</b>	Shown for address packets.
<b>Address + R/W</b>	Address including read/write flag.
<b>R/W</b>	Polarity of the read/write flag.

<b>Address Bits</b>	Address size.
<b>Address ACK</b>	Address acknowledgment.
<b>Data</b>	The data content of the packet (hexadecimal).
<b>Data ACK</b>	Data acknowledgment.

### 8.5.13 I<sup>2</sup>S protocol

I<sup>2</sup>S (Inter-IC Sound) is a serial protocol used in digital audio devices for communications between circuits such as CD transports and audio DACs. The standard was originally developed by Philips Semiconductors. It uses three signals: clock (SCK), word select (WS) and data (SD).

<b>Channel</b>	0 = channel 1 (left). 1 = channel 2 (right).
<b>Data</b>	Raw data.
<b>Value</b>	Data decoded as a two's complement value.

### 8.5.14 LIN protocol

LIN (Local Interconnect Network) is a serial protocol used in automotive electronics to allow microcontrollers to communicate with low-speed peripherals. The standard was defined by the LIN Consortium, a group of five vehicle manufacturers. It uses a single wire pair with a master-slave topology.

<b>Sync Byte</b>	A fixed value (0x55) used for automatic baud rate detection.
<b>Protected Identifier</b>	Message address plus two parity bits.
<b>Frame Identifier</b>	Message address.
<b>Parity OK</b>	Result of Protected Identifier parity check.
<b>Data</b>	The data content of the packet.
<b>Checksum</b>	A cyclic redundancy check of the data.
<b>Classic Checksum</b>	Validates the data bytes.
<b>Enhanced Checksum</b>	Validates the data bytes and identifier.

## 8.5.15 Manchester encoding

Manchester encoding is a serial data signaling scheme once used in magnetic storage devices, but now more common in networks, infrared devices and lighting controls.

For more information on Manchester encoding, see [PicoScope from A to Z](#).

Manchester encoding does not have a specified frame structure. PicoScope's decoder is therefore highly flexible, allowing you to specify the frame configuration used in your application.

<b>Preamble</b>	A bit pattern consisting of alternating <b>1</b> s and <b>0</b> s indicating the start of a frame.
<b>StartBit</b>	The specified number of bits that indicate the start of data.
<b>Header</b>	The specified number of header words.
<b>Data</b>	Either fixed or variable-length field depending on settings.
<b>Checksum</b>	As received by the decoder.
<b>Valid Length</b>	<b>Check mark:</b> Frame is of the specified length <b>X:</b> Frame is of the wrong length

## 8.5.16 Modbus ASCII protocol

Modbus is a low-speed serial data protocol commonly used in industrial applications where a supervisory computer (master) controls or monitors multiple remote devices (slaves). The standard was originally published in 1979 by Modicon (now Schneider Electric) for use with its programmable logic controllers.

Modbus ASCII uses ASCII characters, making it more readable than Modbus RTU but less efficient. Modbus ASCII messages require twice as many bytes to transmit the same content as Modbus RTU messages.

<b>Master</b>	Identifies whether the communicating device is the master or a slave. <ul style="list-style-type: none"><li>✓ represents the master</li><li>✗ represents the slave</li></ul>
<b>Start byte</b>	Indicates the start of the packet.
<b>Slave ID</b>	When the packet is transmitted by the master device, this represents the slave being addressed.  When the packet is transmitted by a slave device, this represents the slave's own ID.
<b>Function</b>	The requested function, as defined in the Modbus library.
<b>Exception</b>	Any errors. If this field is blank, there are no errors. Only a slave device can send an exception.
<b>Data</b>	Data content of the packet.
<b>LRC</b>	Longitudinal redundancy check. An error check of the <b>Data</b> field.
<b>Parity Check</b>	The result of PicoScope's parity check.
<b>End Byte</b>	Indicates the end of the packet.
<b>ASCII Data</b>	The result of PicoScope's ASCII data check.
<b>Length OK</b>	The result of PicoScope's length check.
<b>Valid LRC</b>	The result of PicoScope's LRC.

## 8.5.17 Modbus RTU protocol

Modbus is a low-speed serial data protocol commonly used in industrial applications where a supervisory computer (master) controls or monitors multiple remote devices (slaves). The standard was originally published in 1979 by Modicon (now Schneider Electric) for use with its programmable logic controllers.

Modbus RTU (remote terminal unit) uses binary coding and CRC error checking. It is more efficient than Modbus ASCII, but harder to read without a [link file](#). Modbus RTU messages require half as many bytes to transmit the same content as Modbus ASCII messages.

<b>Master</b>	Identifies whether the communicating device is the master or a slave. <ul style="list-style-type: none"><li>✓ represents the master</li><li>✗ represents the slave</li></ul>
<b>Slave ID</b>	When the packet is transmitted by the master device, this represents the slave being addressed.  When the packet is transmitted by a slave device, this represents the slave's own ID.
<b>Function</b>	The requested function, as defined in the Modbus library.
<b>Exception</b>	Any errors. If this field is blank, there are no errors. Only a slave device can send an exception.
<b>Data</b>	Data content of the packet.
<b>CRC</b>	Cyclic redundancy check. An error check of the <b>Data</b> field.
<b>Parity Check</b>	The result of PicoScope's parity check.
<b>Valid Start</b>	
<b>Valid End</b>	Validity checks carried out by the software on the data format, packet length and the signal itself.
<b>Valid CRC</b>	
<b>Valid Gap</b>	
<b>Valid</b>	Whether or not the packet as a whole is valid.

## 8.5.18 PS/2 protocol

PS/2 (Personal System/2) is the serial protocol used with PS/2 DIN connectors, designed to connect keyboards and mice to a PC. The system was developed by IBM in the late 1980s.

<b>Acknowledge Bit</b>	Device pulls 0 to acknowledge host.
<b>Data</b>	The data content of the packet.
<b>Parity</b>	Error-checking bit. Parity is odd.
<b>Parity Valid</b>	Result of parity check.
<b>Data Direction</b>	Device to Host or Host to Device.

## 8.5.19 SENT Fast protocol

SENT (Single Edge Nibble Transmission) is a one way, point-to-point protocol for transmission of signal values from sensors to a controller. It is used mainly in automotive applications and enables reliable transmission of high-resolution data with lower cost than standards such as CAN, and higher bit rate than LIN.

This decoder extracts only the fast channel data. Use the SENT Slow decoder to extract the other channel.

For more information on the SENT Fast protocol, visit the [PicoScope A to Z](#).

<b>Ch1</b>	Channel 1 data.
<b>Ch2</b>	Channel 2 data.
<b>Calibration Pulse</b>	Duration of pulse, equal to 56 clock ticks.
<b>Clock Tick</b>	Clock period, calculated as 1/56 of calibration pulse.
<b>Status</b>	Status and communication bits.
<b>Data</b>	Variable number of nibbles (4-bit values) depending on the application.
<b>Checksum</b>	Error-checking value.
<b>Reserved 1</b>	Reserved for specific application.
<b>Reserved 2</b>	Reserved for specific application.
<b>Serial Data</b>	Serial data message bits.
<b>Message Start</b>	Message start = 1. Otherwise = 0.
<b>Legacy Checksum Pass</b>	Result of CRC test to February 2008 standard.
<b>Recommended Checksum Pass</b>	Result of CRC test to January 2010 standard.

## 8.5.20 SENT Slow protocol

SENT (Single Edge Nibble Transmission) is a one way, point-to-point protocol for transmission of signal values from sensors to a controller. It is used mainly in automotive applications and enables reliable transmission of high-resolution data with lower cost than standards such as CAN, and higher bit rate than LIN.

This decoder extracts only the slow channel data. Use the SENT Fast decoder to extract the other channel.

For more information on the SENT Fast protocol, visit the [PicoScope A to Z](#).

<b>8Bit Message ID</b>	Definition of the message ID.
<b>Enhanced Serial Message</b>	Definition of the data value.
<b>CRC</b>	The cyclic redundancy check value within the packet.
<b>Configuration bit</b>	Packet is 12-bit data and 8-bit message ID, or 16-bit data and 4-bit message ID.
<b>Message ID</b>	Message value.
<b>Data</b>	The data content of the packet.
<b>Data Value</b>	Integer value of the data field.
<b>CRC Pass</b>	Serial data message bits computed CRC matches the packet CRC.

## 8.5.21 SPI protocol

SPI (Serial Peripheral Interface) bus was originally developed by Motorola for use with their microcontrollers. Due to the simplicity of the bus, other manufacturers adopted it and it has become widely available in components used in embedded system designs. It is commonly used for chip-to-chip communications between a CPU and a peripheral device such as keyboard, display, ADCs and DACs, real-time clocks, EEPROM, SD and other memory devices.

SPI is a bidirectional protocol. To decode both directions (SDI and SDO) set up two PicoScope SPI decoders, one for each direction.

For more information on the SPI protocol, visit the [PicoScope A to Z](#).

<b>Data</b>	The data content of the packet (hexadecimal).
-------------	---

## 8.5.22 UART protocol

A UART (Universal Asynchronous Receiver/Transmitter) is the communication interface found in the serial or COM ports once commonly found on computers. UARTs were developed in the 1960s for connecting modems to terminals. Each channel of data is transmitted on a single wire pair using an electrical standard such as RS-232, RS-422, or RS-485.

A typical full-duplex UART link consists of a Tx (transmit) and an Rx (receive) channel. To decode both channels, set up two PicoScope UART decoders, one for each channel.

For more information, see [PicoScope from A to Z](#).

<b>Data</b>	The data content of the packet.
<b>Parity</b>	Error-checking bit (if parity is enabled).
<b>Parity Check</b>	Result of parity check (if parity is enabled).

## 8.5.23 USB (1.0/1.1) protocol

USB (Universal Serial Bus) is widely adopted in today's personal computers and tablets for communication and power to printers, scanners, keyboard, mice, memory devices and many other peripherals. It is also used for chip-to-chip and module-to-module communication in many other types of electronic devices and embedded systems. The USB specification has evolved several times since its launch in 1996.

There are four types of USB packet: *Handshake*, *Data*, *Token* and *Special*. *Handshake* is used to initialize communication between two end points. *Data* is used to send actual information. *Token* and *Special* (a specialized version of *Token*) are used to keep both devices informed concurrently. The data fields shown will vary depending on the USB packet type.

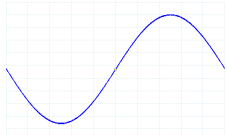
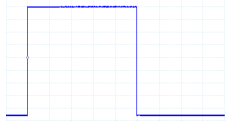





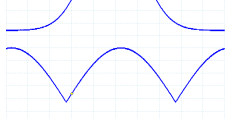

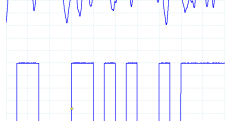
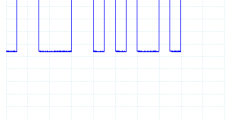
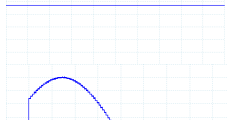
For more information on the USB (1.0/1.1) protocol, visit the [PicoScope A to Z](#).

<b>Data</b>	The data content of the packet.
<b>CRC5</b>	Cyclic Redundancy Check, 5 bits.
<b>CRC16</b>	Cyclic Redundancy Check, 16 bits.
<b>Pid</b>	Packet Identifier. Determines the packet type and structure.
<b>Address</b>	For Token-specific USB packets.
<b>Endpoint</b>	For Token-specific USB packets.
<b>Error Message</b>	Set if PicoScope detects an error.



## 8.6 Signal generator waveform types

The list of waveform types available in the [Signal Generator dialog](#) 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 <b>Offset</b> control
Arbitrary		Any waveform created by the arbitrary waveform editor

## 8.7 Spectrum window functions

To create a [spectrum view](#), PicoScope captures a block of sampled data over a finite time interval and then uses a fast Fourier transform (FFT) to compute its spectrum. The algorithm assumes a signal level of zero at all times outside the captured time interval. Typically, this assumption causes sharp transitions to zero at either end of the data, and these transitions have an effect on the computed spectrum, creating unwanted artifacts such as ripple and gain errors. To reduce these artifacts, the signal can be faded in and out at the start and end of the block. There are several commonly used **window functions** that can be convolved with the data to effect this fading, and which are chosen according to the type of signal and the purpose of the measurement.

The **Window Functions control** in the [Spectrum Options dialog](#) lets you select one of the standard window functions for spectrum analysis. The following table shows some of the figures of merit used to compare the functions.

Window	Main peak width (bins @ -3 dB)	Highest side lobe (dB)	Side lobe roll-off (dB/octave)	Remarks
<b>Blackman</b>	1.68	-58	18	Often used for audio work
<b>Gaussian</b>	1.33 to 1.79	-42 to -69	6	Gives minimal time and frequency errors
<b>Triangular</b>	1.28	-27	12	Also called Bartlett window
<b>Hamming</b>	1.30	-41.9	6	Also called raised sine-squared; used in speech analysis
<b>Hann</b>	1.20 to 1.86	-23 to -47	12 to 30	Also called sine-squared; used for audio and vibration work
<b>Blackman-Harris</b>	1.90	-92	6	General-purpose
<b>Flat-top</b>	2.94	-44	6	Negligible pass-band ripple; used mainly for calibration
<b>Rectangular</b>	0.89	-13.2	6	No fading; maximal sharpness; used for short transients

## 8.8 Device feature table

Some PicoScope 6 features require specialized hardware and are therefore not available on all devices. Feature availability is indicated in the table below. (Detailed specifications for each feature may vary). Please see the relevant device data sheet for more details.

Series / Model	50	AT	AW	AX	BL	CD	DC	DI	DM	EX	FC	FR	LP	RA	RU	SG	SM	ST	SW	TS
<a href="#">USB DrDAQ</a>			✓													✓	✓			
PicoLog 1000																	✓			
PicoScope 2104-2105																	✓			
PicoScope 2202																	✓			
PicoScope 2203			✓														✓	✓		✓
PicoScope 2204-2205		✓	✓														✓	✓		✓
PicoScope 2206-2208		✓	✓				✓			✓				✓			✓	✓	✓	✓
PicoScope 2204A																				
PicoScope 2205A		✓	✓														✓	✓		✓
PicoScope 2206A																				
PicoScope 2207A		✓	✓					✓									✓	✓	✓	✓
PicoScope 2208A																	✓	✓	✓	✓
PicoScope 2205 <a href="#">MSO</a>		✓	✓						✓						✓	✓	✓	✓	✓	✓
PicoScope 2205A <a href="#">MSO</a>		✓	✓				✓	✓							✓	✓	✓	✓	✓	✓
PicoScope 2000B		✓	✓				✓						✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 2000B <a href="#">MSO</a>		✓	✓				✓	✓					✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 2405A		✓	✓				✓							✓	✓	✓	✓	✓	✓	✓
PicoScope 3204									✓	✓			✓				✓			
PicoScope 3205-3206									✓	✓			✓				✓			✓
PicoScope 3223/3423									✓									✓		
PicoScope 3224/3424		✓							✓									✓		
PicoScope 3425		✓							✓									✓		
PicoScope 3200A		✓					✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 3400A		✓			✓		✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 3200B		✓	✓				✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 3400B		✓	✓		✓		✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 3000D		✓	✓		✓		✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 3000 <a href="#">MSO</a>		✓	✓				✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 3000D <a href="#">MSO</a>		✓	✓		✓		✓	✓	✓				✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 4223-4224									✓		✓			✓	✓		✓			
PicoScope 4423-4424		✓							✓					✓	✓		✓			
PicoScope 4225/4425		✓			✓	✓			✓		✓			✓	✓		✓			
PicoScope 4226-4227		✓	✓						✓	✓				✓	✓	✓	✓	✓	✓	✓
PicoScope 4262		✓	✓		✓				✓	✓				✓	✓	✓	✓	✓	✓	✓
PicoScope 4444		✓			✓		✓		✓			✓	✓	✓	✓		✓			
PicoScope 4824		✓	✓				✓		✓				✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 5203-5204		✓	✓	✓					✓	✓				✓	✓		✓	✓	✓	✓
PicoScope 5000A		✓			✓		✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 5000B		✓	✓		✓		✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 5000D		✓	✓		✓		✓		✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 5000D <a href="#">MSO</a>		✓	✓		✓		✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 6000	✓	✓	✓	✓	✓		✓		✓				✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 6000A/C	✓	✓		✓	✓		✓		✓				✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 6000B/D	✓	✓	✓	✓	✓		✓		✓				✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 6000E	✓	✓	✓	✓	✓		✓	✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓
PicoScope 6407	✓	✓	✓	✓					✓				✓	✓	✓	✓	✓	✓	✓	✓

50 [50 Q inputs](#)  
 AT [Advanced triggers](#)

FC [Frequency counter](#)  
 FR [Flexible resolution](#)

AW	<a href="#">Arbitrary waveform generator</a>	LP	<a href="#">Lowpass filtering</a>
AX	<a href="#">Auxiliary input/output</a>	RA	<a href="#">Rapid triggering</a>
BL	<a href="#">Switchable bandwidth limiter</a>	RU	<a href="#">Runt pulse trigger</a>
CD	<a href="#">ConnectDetect®</a>	SG	<a href="#">Signal generator</a>
DC	<a href="#">DC offset adjustment</a>	SM	<a href="#">Streaming mode</a>
DI	<a href="#">Digital inputs</a>	ST	<a href="#">Signal generator triggers</a>
DM	<a href="#">DeepMeasure™</a>	SW	<a href="#">Signal generator sweep modes</a>
EX	<a href="#">External trigger input</a>	TS	<a href="#">Time-stamping (high-resolution)</a>

## 8.9 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  $\times 1$ . On traditional oscilloscopes, this function is often known as the timebase. [Click here](#) for more information.

**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.

**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.

**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.

**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.

**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.

**Grid.** The arrangement of [viewports](#). The number of grid rows and the number of grid columns can each be either 1, 2, 3 or 4.

**IEPE.** Integrated circuit piezoelectric. A type of sensor, usually built to detect acceleration, vibration or sound, and with a built-in amplifier. IEPE sensors can only be used with special PicoScope oscilloscopes that have IEPE-compatible inputs.

**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.

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

**PC Data Logger.** A measuring instrument consisting of a hardware interface and the PicoLog software running on a PC. You can also use selected data loggers with the PicoScope software to create a multichannel voltage input oscilloscope.

**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. You can improve its performance by upgrading the PC using standard parts from any computer shop, or by purchasing a new scope device, and you can upgrade the software by downloading an update from Pico Technology.

**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.

**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.

**Slow sampling 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 page](#) of the [Preferences dialog](#)), it switches to slow sampling 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  $y_0 \cdots y_{n-1}$  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.

**Tool tip.** A label that appears when you move the mouse pointer over some parts of the PicoScope screen, such as buttons, controls and rulers.

**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.

# Index

## %

- %buffer% variable 99
- %file% variable 99
- %time% variable 99

## A

- AC adapter 194
- AC coupling 222
- AC power 107
- Accelerometer inputs 131
- Access 4
- Add Measurement dialog 55
- Adding a measurement 55
- Advance function 86
- Advanced trigger
  - types 159
- Advanced triggers 157, 158, 221
- Alarms 34, 61, 98
  - save on trigger 182
- Analog intensity 126
- Analog offset 133
- AND logical operator 164
- Animated GIF 42
- Arbitrary waveform generator 143, 221, 222
  - editing window 146
  - exporting as CSV file 149
  - importing from channel 146, 148
  - importing from CSV file 149
- Automotive menu 119
- Auxiliary (AUX) I/O 157, 159, 221
- Average (math channel) 85
- Average (statistics) 20
- AWG 222
- Axis 13, 16, 18, 222
  - auto-arrange 53
  - horizontal 13, 16, 18
  - offset 174
  - scaling 133, 135, 174
  - vertical 13, 16, 18

## B

- Bandwidth limiter 133, 221
- Battery power 107
- Beep 98
- Binary files, exporting 44
- Bit stream 147

- bmp files 42
- Bring channel to front 135
- Buffer Navigation toolbar 128
- Buffer Overview 34
- Bugs
  - reporting 121
- Buttons 122
  - Measurements 166
  - Notes 166
  - Rulers 166
  - Signal Generator 143
  - Stop and Go 156

## C

- Capture count 20
- Capture modes 10
- Capture rate 28
  - preference 107
- Capture Setup toolbar 122
- Capture size preference 106
- Changing device 167
- Channel 222
  - selecting in a view 52
- Channel controls 131
- Channel Labels 50, 51, 166
- Channel Options
  - button 131
  - menu 132
- Channel ordering 135
- Channel settings
  - in Properties sheet 28
- Channels toolbar
  - PicoLog 1000 Series 151
  - standard 131
  - USB DrDAQ 152
- Clipboard 48
- Collection Time 222
  - control 123
- Collection Time control 122
- Collection time units 105
- Color preferences 114
- Column Auto-width 54
- Command-line syntax 191
- Connect Device dialog 36, 38
- ConnectDetect 131, 139
- Controls 122
  - Channel 131
  - Trigger 156
- Converting data files 189, 191
- Copy
  - as image 48

- Copy
  - as text 48
- Copyright 4
- Coupling control 131
- Create New Custom Probe dialog 66
- Create New Math Channel dialog 81
- CSV files 41, 222
  - exporting 43
- Custom grid layout dialog 53
- Custom Probe wizard 64
  - Create New Custom Probe dialog 66
  - Edit Existing Custom Probe dialog 67
  - Edit Range dialog 74
  - Filter Method dialog 76
  - Lookup-table Scaling dialog 70
  - Manual Ranges Setup dialog 72
  - Probe Identification dialog 77
  - Probe Output Units dialog 68
  - Range Management dialog 71
  - Scaling Method dialog 69
- Custom probes 29, 61, 174
  - dialog 63
  - saving 36

## D

- Data files
  - converting 189
- DC coupling 222
- DC IN power 194
- DC inputs, 50  $\Omega$  131, 221
- DC offset 133, 221
- Dead time 222
- Decoding tab 32
- DeepMeasure 21, 54, 58, 221
  - data window 60
  - DeepMeasure settings dialog 59
  - dialog 58
  - overlay 60
- Default print settings 113
- Delay function 86
- Demo device 39, 150
- Demo mode 39, 150, 222
- Demo Signals
  - dialog 150
  - menu 150
- Details dialog
  - channel labels 50
  - customer details 50
  - vehicle details 50
- Device
  - how to change 167

- Device feature table 221
- Digital Channels & Groups button 140
- Digital color 126
- Digital inputs 141, 221
- Digital Inputs button 131
- Digital outputs
  - PicoLog 1000 Series 151
  - USB DrDAQ 155
- Digital trigger 160
  - dialog 162
- Digital view 15
  - context menu 15
- DrDAQ 152
  - LED control 154
- Dropout trigger 160

## E

- Edge trigger 159
- Edit Existing Custom Probe dialog 67
- Edit Measurement dialog 55
- Edit menu 48
- Edit Range dialog 74
  - advanced options 75
- Edit Reference Waveform dialog 91
- Editing a measurement 55
- Effective resolution 28, 133, 134
- Equivalent time sampling 157
- ETS 157, 222
  - and advanced triggers 158
- Exit 37
- Exporting data 42
  - binary format 44
  - text format 43
- External (EXT) trigger 157, 159, 221

## F

- Fall time
  - threshold 56
- Falling Edge button 157
- Feedback 120, 121
- File conversion 189
- File menu 36
- Filter Method dialog 76
- Filtering 133
  - channels 137
  - measurements 20
- Fitness for purpose 4
- Flexible power 194
- Flexible Resolution 221
- Flow meter 132



- Focus 223
- Frequency counter 221
- Frequency counting 131, 133
- Frequency difference
  - measuring 169
- Frequency legend 23, 27
- Frequency rulers 18, 23
- Functions, mathematical 30, 82

## G

- General preferences 105
- Generate Mask
  - button 100
  - dialog 101
- gif files 42
- Glitches, finding 159
- Glossary 222
- Graticule 16, 18, 223
- Grid 223
  - layout 53
- Groups, digital input 142
- Guided tests 119

## H

- Hand tool 129
- Hardware resolution 123
- Harmonic control for measurements 57
- Help menu 120
- Horizontal axis 13, 16, 18
- House current 107
- How to... 167
  - change device 167
  - convert files in Windows File Explorer 189
  - create a link file 187
  - find a glitch using persistence mode 176
  - measure frequency difference 169
  - measure time difference 169
  - move a view 170
  - save on trigger 182
  - scale and offset a signal 172
  - set up a Mask Limit Test 179
  - set up the spectrum view 175
  - use rulers to measure a signal 167
- Hydraulics 196, 197, 200
  - ram 196
  - telehandler 197, 200
- Hysteresis 161

## I

- IEPE 223

- inputs 131
- Image, saving as 41
- Import from a Channel dialog 148
- Input impedance 131
- Interpolation
  - linear 109
  - sin(x)/x 109
- Interval trigger 159

## K

- K Factor 132
- K Factor calibration 132
- Keyboard maps 111
- Keyboard shortcuts 110, 129

## L

- Language preferences 112
- LED
  - on USB DrDAQ 154
- Legal statement 4
- Liability 4
- Line thickness 115
- Link file 96
  - creating 187
- Lissajous figures 16
- Logic trigger 160
  - dialog 163
- Lookup-table Scaling dialog 70
- Lowpass filtering 76, 133, 137, 221

## M

- Macro
  - recorder 61, 103
  - running from command line 193
- Mains power 107
- Manual Ranges Setup dialog 72
- Marquee Zoom tool 129
- Mask limit testing 33
  - Setup 179
- Masks 61
  - colors 33, 114
  - displaying 100
  - editing 102
  - exporting 100
  - generating 100, 101
  - importing 100
  - in Buffer Overview 34
  - mask files 100
  - Mask Library dialog 100
  - menu 100

- Masks 61
  - polygons 102
- Math Channel wizard 79
  - Create new math channel 81
  - Equation dialog 82
  - Finished dialog 89
  - Name and Color dialog 87
  - Units and Range dialog 88
- Math channels 61, 78
  - Built-in 78
  - dialog 78
  - Library 78
  - Loaded 78
  - operands 83
  - overview 30
  - saving 36
  - warning symbol 30
  - wizard 79
- MATLAB files
  - exporting 44, 91
  - saving 42
- Max (math channel) 85
- Max (statistics) 20
- Maximum Waveforms setting 105
- Measurement system
  - selecting 112
- Measurements
  - adding 20, 54, 55
  - buttons 166
  - capture size 106
  - DeepMeasure 54
  - deleting 20, 54
  - editing 20, 54, 55
  - grid font size 54
  - menu 54
  - scope 56, 202
  - spectrum 57, 203
  - statistics 20
  - table 20
  - types 202
- Menus 36
  - Automotive 119
  - Edit 48
  - File 36
  - Help 120
  - Measurements 54
  - Tools 61
  - View 52
- Metric measurements 112
- Min (math channel) 85
- Min (statistics) 20
- Missing events, finding 159

- Mission-critical applications 4
- MSO 223
  - setup 141
  - view 14

## N

- NAND logical operator 164
- No. samples 28
- NOR logical operator 164
- Normal Selection tool 129
- Notes 48, 49, 166
  - area 49
  - importing from command line 192

## O

- Offset 172
  - analog 133
- Open
  - dialog 40
- Open file 36, 40
- OR logical operator 164
- Oscilloscope 7
- Other buttons 166
- Overrange
  - BNC shell to chassis 13
  - indicator 13, 131
  - normal measurement range 13

## P

- Panning 129, 130
- Pattern trigger 162
- PC Data Logger 223
- PC Oscilloscope 8, 223
- Persistence mode 10, 19
  - button 122
  - enabling and disabling 116
  - finding a glitch 176
  - options 126
- Phase rulers 24
  - partition 26
  - units 26
  - wrap 26
- PicoLog 1000 Series
  - Channels toolbar 151
  - Digital Outputs control 151
- PicoScope 6 1
  - how to use 1, 6, 7, 9
  - main window 12
- png files 42
- Pointer tool tip 22

- Polygon 102
- Power Management preferences 107
- Power supply 194
- Preferences 61
  - capture rate 107
  - colors 114
  - Default Print Settings 113
  - device selection 116
  - dialog 104
  - general 105
  - keyboard 110
  - language 112
  - persistence modes 116
  - power management 107
  - sampling 108
  - spectrum modes 116
  - updates 118
- Pre-trigger 165
  - control 157, 165
- Printing 37
  - from command line 192
  - from menu 37
  - preferences 113
  - preview 37
- Probe 223
  - custom 29, 61, 63, 174
  - Identification dialog 77
  - Output Units dialog 68
- Properties sheet 28
  - displaying 53
- psdata files
  - converting 189, 191
  - opening 40
  - saving 41
- pskeys files 111
- psmaths files 79
  - opening 40
- psreference files 91
  - opening 40
- pssettings files 41
  - opening 40
- Pulse width trigger 159
- PWM output
  - PicoLog 1000 Series 151
  - USB DrDAQ 155

## R

- Range control 131
- Range Management dialog 71
- Rapid trigger 157
- Rapid triggering 221

- Reference 191
- Reference waveforms 31, 61
  - delay 135
  - dialog 90
  - edit 91
  - Library 90
  - Loaded 90
  - using in equations 83
- Reset "Don't show this again" dialogs 105
- Resolution enhancement 28, 133, 134, 223
- RGB LED on USB DrDAQ 154
- Rise time
  - threshold 56
- Rising Edge button 157
- Rotation rulers 24
  - partition 26
  - units 26
  - wrap 26
- RPM 27, 117
- Rulers 13, 18, 166
  - definition 223
  - deleting 22, 23
  - frequency 18, 23, 169
  - handles 13, 22, 23, 24
  - legend 27
  - lock button 27
  - phase 24
  - rotation 24
  - settings 26
  - signal 13, 18, 22, 167
  - time 13, 23, 169
  - using to measure a signal 167
- Runt pulse trigger 160, 221

## S

- Sampling preferences 108
- Sampling rate 123
- Save As 36
  - dialog 41
- Save file 36
- Save on trigger 99, 182
- Scaling 135, 172
- Scaling Method dialog 69
- Scope device 223
- Scope measurements
  - AC volts 202
  - Cycle time 202
  - DC average 202
  - Duty cycle 202
  - Fall time 202
  - Falling rate 202

- Scope measurements
  - Frequency 202
  - High pulse width 202
  - Low pulse width 202
  - Mask failures 202
  - Maximum 202
  - Minimum 202
  - options 56
  - Peak to peak 202
  - Rise time 202
  - Rising rate 202
  - True RMS 202
- Scope mode 10
  - button 122
- Scope view 10, 13
- Scrolling 172
- Select Digital Channels/Groups dialog 141
- Selection tool, normal 129
- Send channel to back 135
- Serial decoding 32, 61, 92
  - 1-Wire protocol 206
  - ARINC 429 protocol 206
  - BroadR-Reach protocol 206
  - CAN FD protocol 207
  - CAN protocol 207
  - DALI protocol 208
  - data window 96
  - DCC protocol 209
  - decoder settings dialog 93
  - dialog 92
  - DMX512 protocol 209
  - Ethernet 10BASE-T protocol 210
  - Fast Ethernet 100BASE-TX protocol 210
  - FlexRay protocol 211
  - I<sup>2</sup>C protocol 211
  - I<sup>2</sup>S protocol 212
  - LIN protocol 212
  - link file 96
  - LPS/2 protocol 216
  - Manchester encoding 213
  - Modbus ASCII protocol 214
  - Modbus RTU protocol 215
  - protocols 205
  - SENT Fast protocol 216
  - SENT Slow protocol 217
  - SPI protocol 217
  - UART protocol 218
  - USB (1.0/1.1) protocol 218
- Serial number
  - of oscilloscope 120
- Settings
  - saving 36
- Signal difference
  - how to measure 167
- Signal enerator
  - button 143
- Signal generator
  - basic mode 143
  - dialog 143
  - dialog, USB DrDAQ 153
  - sweep mode 144, 221
  - triggers 144, 221
  - waveform types 219
- Signal rulers 13, 18, 22
- Sin(x)/x preferences 109
- Slow sampling mode 223
- Slow sampling transition 108
- Smoothing 109
- Software version 1
- Sound file 99
- Space bar 156
- Spectrum measurements
  - Amplitude at peak 203
  - Average amplitude at peak 203
  - Frequency at peak 203
  - Intermodulation Distortion (IMD) 203, 204
  - Mask failures 203
  - options 57
  - Signal to noise and distortion ratio (SINAD) 203
  - Signal to noise ratio (SNR) 203
  - Spurious-free dynamic range (SFDR) 203
  - Total harmonic distortion (THD) 203
  - Total harmonic distortion plus noise (THD+N) 203
  - Total power 203
- Spectrum mode 10
  - button 122
  - enabling and disabling 116
- Spectrum options
  - bins 124
  - dialog 124
  - display mode 125
  - scale 125
  - window functions 125, 220
- Spectrum view 10, 18
  - how to set up 175
- Spreadsheet, exporting to 41
- Spurious edges, finding 160
- Standard deviation 20, 223
- Start-up Settings menu 46
- Statistics 20
- Stop and Go buttons 156
- Suggestions 121
- Support 4
- Sweep mode 144, 221

## Symbols

- red warning 13, 131
- yellow warning 13

System requirements 5

**T**

Text files, exporting 42, 43

Threshold for measurements 56

Thresholds, digital input 141

Time difference

- how to measure 169

Time gate 28

Time rulers 13, 23

Timebase 222

Time-delay 165

- arrow 17
- control 157, 165
- Enable button 157

Time-stamping 221

Tool tip 224

Toolbars 122

- Buffer Navigation 128
- Capture Setup 122
- Channels 131, 151, 152
- move bottom toolbar 117
- Trigger 156
- Zooming and Scrolling 129

Tools menu 61

Trace 7

Tracking rulers 27

Trade marks 4

Translation 120, 121

Trigger 156, 165, 224

- advanced 156, 158
- controls 156
- digital 160, 162
- dropout 160
- dual-edge 159
- edge 159
- glitches 159
- hysteresis 161
- interval 159
- logic 160, 163
- marker 17
- missing events 159
- mode control 156
- pulse width 159
- reference point 165
- runt pulse 160
- signal generator 144
- spurious edges 160

threshold 157

timing 165

toolbar 156

valid edges 160

window 159

txt files 42

Types of measurement 202

**U**

U.S. measurements 112

Undo zoom 130

Updates

- checking for 120
- preferences 118

Upgrades 4

Usage 4

Usage statistics 118

USB DrDAQ 152

- digital outputs 155

LED control 154

PWM output 155

USB power 194

User's Guide 120

**V**

Valid edges, finding 160

Version number

- hardware 120
- software 1, 120

Vertical axis 13, 16, 18

Vertical resolution 224

View 224

- enabling sub views 52
- how to move 170
- menu 52
- scope 13
- selecting channels 52
- spectrum 18
- XY 16

Viewport 224

Viruses 4

**W**

Warning symbol

- red 13, 131
- yellow 13

Waveform 7, 13

- saving 36

Waveform buffers

- maximum 105

Waveform Library 47  
Window functions 28, 124, 125, 220  
Window trigger 159  
WPS600C 196, 197, 200

## X

X axis, configuring 52  
X-Axis command 16  
XNOR logical operator 164  
XOR logical operator 164  
XY view 16

## Z

Zero offset 133  
Zooming 172  
    undo 130  
    Zoom Overview window 130  
    Zooming and Scrolling toolbar 129

**United Kingdom  
headquarters**

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

Tel: +44 (0) 1480 396 395

[sales@picotech.com](mailto:sales@picotech.com)  
[support@picotech.com](mailto:support@picotech.com)

[www.picotech.com](http://www.picotech.com)

**North America regional  
office**

Pico Technology  
320 N Glenwood Blvd  
Tyler  
Texas 75702  
United States

Tel: +1 800 591 2796

[sales@picotech.com](mailto:sales@picotech.com)  
[support@picotech.com](mailto:support@picotech.com)

**Asia-Pacific regional office**

Pico Technology  
Room 2252, 22/F, Centro  
568 Hengfeng Road  
Zhabei District  
Shanghai 200070  
PR China

Tel: +86 21 2226-5152

[pico.china@picotech.com](mailto:pico.china@picotech.com)