

PicoScope 3000 series

User guide

Copyright 2004 Pico Technology Limited. All rights reserved.

Table of Contents

Part I Introduction	3
1 Overview	3
2 Installation instructions	3
3 Safety symbols	4
4 Safety warning	5
5 FCC notice	6
6 CE notice	6
7 Company details	7
Part II Product information	8
1 Specifications	8
2 Minimum system requirements	9
Part III Advanced features	10
1 Sampling modes	10
2 More on block mode	10
3 More on streaming mode	10
4 Triggering	11
5 ETS (Equivalent Time Sampling)	11
6 Voltage ranges	11
7 AC/DC operation	11
8 Oversampling	11
9 Scaling	11
10 Signal generator	12
11 Combining oscilloscopes	12
Part IV Using different modes	13
1 Introduction	13
2 Using block mode	13
3 Using streaming mode	13
4 Using ETS mode	13
Part V Technical reference	15
1 Driver	15
2 Functions	15
3 Programming	25
4 Glossary	27
Part VI Troubleshooting	30

1 FAQ	30
2 Error codes	30
Index	32

1 Introduction

1.1 Overview

The PicoScope 3000 series is a range of high speed PC oscilloscopes, fully USB 2.0 capable and backwards compatible with USB 1.1. There is no need for an additional power supply, as power input is from the USB port: these oscilloscopes are highly portable.

With the PicoScope software, oscilloscopes from the PicoScope 3000 series can be used as PC oscilloscopes and spectrum analyzers; with the PicoLog software, oscilloscopes from the PicoScope 3000 series can be used as data loggers. Alternatively, you might want to use some of the API functions to develop your own programs to collect and analyze data from the oscilloscope.

A typical PicoScope 3000 oscilloscope is supplied with the following items:

- USB cable, for use with both types of USB port
- Software CD
- Installation guide

1.2 Installation instructions

Warning: Do not connect an oscilloscope from the PicoScope 3000 series to the PC until the software has been installed. Install the software by following the steps in the installation guide supplied with your oscilloscope. Your oscilloscope is connected to the PC with the USB cable supplied. There is no need for an additional power supply as power is drawn from the USB port.

Checking the installation

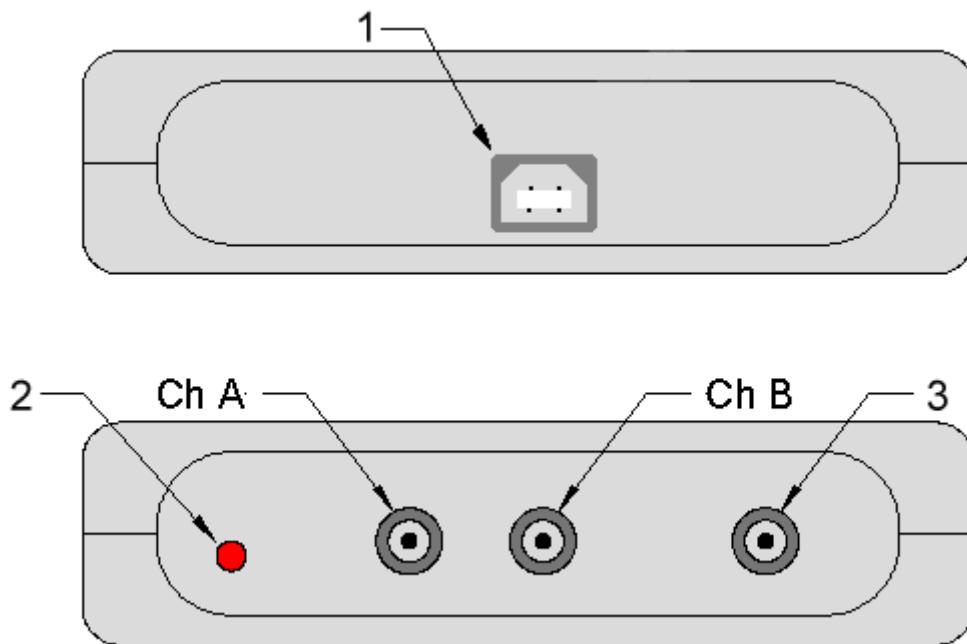
Once the software has been installed, ensure that the oscilloscope is connected to the PC and start up the PicoScope or PicoLog software. The software should now display the voltage that you have connected. If you are using a scope probe and PicoScope, you should see a small 50Hz or 60Hz mains signal in the oscilloscope window when you touch the scope probe tip with your finger.

Standard oscilloscope connectors

PicoScope 3000 series oscilloscopes have standard oscilloscope connectors. The input impedance is also standard, so the x10 function on scope probes works correctly.

The BNC connector, labelled **3** on the diagram below, 'E' on the oscilloscope, has two functions. In normal use it is the external trigger input with an adjustable threshold. Alternatively, on some oscilloscopes, this connector can also be used to output sine, square and triangle waveforms which can be swept back and forth at a user-defined frequency. The integrated signal generator can be controlled via the PicoScope software or by API calls. The signal generator can also be used to compensate x10 scope probes when set to output a square wave.

Connector diagram



1 USB port connector

2 LED. When lit, indicates the PicoScope 3000 series oscilloscope is sampling data

3 External trigger / Signal generator

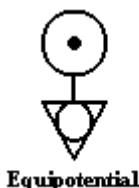
1.3 Safety symbols

Symbol 1: Warning Triangle



This symbol indicates that a safety hazard exists on the indicated connections if correct precautions are not taken. Ensure that all safety documentation associated with the product is read in detail before a product is used.

Symbol 2: Equipotential



This symbol indicates that the outer shell of the indicated BNC connectors are all at the same potential (i.e. are shorted together). The user must therefore take necessary precautions to avoid a potential being applied across the return connections of the indicated BNC terminals as this may result in a large current flow causing damage to the product and/or connected equipment.

1.4 Safety warning

We strongly recommend that you read the general safety information below before using your oscilloscope for the first time. Safety protection built in to equipment may cease to function if the equipment is used incorrectly. This could cause damage to your computer, or lead to injury to yourself and others.

Maximum input range

Oscilloscopes from the PicoScope 3000 range are designed to measure voltages in the range -20V to +20V. Any voltages in excess of $\pm 100V$ may cause physical damage.

Mains voltages

Pico Technology products are not designed for use with mains voltages. To measure mains, use a differential isolating probe specifically designed for a high source voltage.

Safety grounding

PicoScope 3000 series oscilloscopes connect direct to the ground of a computer via the interconnecting cable provided. This method minimises interference.

As with most oscilloscopes, avoid connecting the ground input to any source other than ground. If in doubt, use a meter to check that there is no significant AC or DC voltage. Failure to check may cause damage to your computer, or lead to injury to yourself and others.

You should assume that the product does not have a protective safety earth.

Repairs

The oscilloscope contains no serviceable parts: repair or calibration of the oscilloscope requires specialised test equipment and must be performed by Pico Technology.

1.5 FCC notice

This device complies with part 15 of the FCC Rules. Operation is subject to the following two conditions:

- (1) This device may not cause harmful interference**
- (2) This device must accept any interference received, including interference that may cause undesired operation**

This equipment has been tested and found to comply with the limits for a Class B digital device, pursuant to part 15 of the FCC Rules. These limits are designed to provide reasonable protection against harmful interference in a residential installation. This equipment generates, uses and can radiate radio frequency energy and, if not installed and used in accordance with the instructions, may cause harmful interference to radio communications. However, there is no guarantee that interference will not occur in a particular installation. If this equipment does cause harmful interference to radio or television reception, which can be determined by turning the equipment off and on, the user is encouraged to try to correct the interference by one or more of the following measures:

- Reorient or relocate the receiving antenna
- Increase the separation between the equipment and receiver
- Connect the equipment into an outlet on a circuit different from that to which the receiver is connected
- Consult the dealer or an experienced radio/TV technician for help

For safety and maintenance information see the [safety warning](#).

1.6 CE notice

The PicoScope 3000 series of PC oscilloscopes meets the intent of the EMC directive 89/336/EEC and the following standard has been applied: EN61326-1 (1997) Class A Emissions and Immunity.

Products from the PicoScope 3000 range also meet the intent of the Low Voltage Directive and the following standard has been applied: BS EN 61010-1:2001 IEC 61010-1:2001 (safety requirements for electrical equipment, control, and laboratory use)

1.7 Company details

Address:

Pico Technology Limited
The Mill House
Cambridge Street
St Neots
Cambridgeshire
PE19 1QB
United Kingdom

Phone: +44 (0)1480 396395
Fax: +44 (0)1480 396296

Email:

Technical Support
support@picotech.com

Sales
sales@picotech.com

Web site:

www.picotech.com

2 Product information

2.1 Specifications

	PicoScope 3204	PicoScope 3205	PicoScope 3206
VERTICAL RESOLUTION	8 bit	8 bit	8 bit
ANALOG BANDWIDTH	50 MHz	100 MHz	200 MHz
MAXIMUM SAMPLING RATE			
Single channel	50 MS/s	100 MS/s	200 MS/s
Dual channel	50 MS/s	100 MS/s	100 MS/s
Repetitive signals	2.5 GS/s	5 GS/s	10 GS/s
TRIGGER BANDWIDTH	50 MHz	100 MHz	150 MHz
BUFFER SIZE			
Single channel	256 k	512 k	1 M
Dual channel	128 k	256 k	512 k
INPUTS	2 channels via BNC 1 MΩ impedance AC/DC coupling 20 pF input capacitance		
OUTPUTS			
Signal generator	1 BNC shared with external trigger Fixed frequency 1 kHz 5 V square wave 600 Ω output impedance	1 BNC shared with signal generator Variable frequency 1 Hz to 1 MHz 5 V square wave 1 V sine wave and triangle functions Repeat sweep function Dual slope function 600 Ω output impedance	
External trigger	1 BNC output shared with signal generator Variable trigger threshold ± 20 V rising/falling 12.2 mV resolution 1 MΩ impedance		
VOLTAGE RANGES	± 100 mV to ± 20 V ranges in 1, 2, 5 steps		
ACCURACY	3 % voltage 100 ppm time		
OPERATING ENVIRONMENT			
Temperature range	0°C to 70°C (25°C for quoted accuracy)		
Humidity	25% to 75% RH		
OVERLOAD PROTECTION			
Channels	± 50 V		
External trigger	± 30 V		
PC CONNECTION	USB 2.0 Compatible with USB 1.1		
POWER SUPPLY	From USB port: 4.6 to 5.25 V; 500 mA External power supply is not required		
DIMENSIONS	140 mm x 190 mm x 45 mm		
COMPLIANCE	CE standard ; FCC standard		

2.2 Minimum system requirements

For oscilloscopes from the PicoScope 3000 series to operate, a computer with the minimum system requirements to run Windows or the following (whichever is the higher specification) is required:

Processor	Pentium class processor or equivalent minimum.
Memory	32 MB minimum.
Disk space	10 MB minimum.
Operating system	Microsoft Windows 98 SE, ME, Microsoft Windows 2000, XP or later.
Ports	USB 1.1 compliant port minimum. USB 2.0 compliant port recommended. Must be connected direct to the port or a powered USB hub. Will not work on a passive hub.

3 Advanced features

3.1 Sampling modes

PicoScope 3000 series oscilloscopes run in various sampling modes. At high sampling rates, the PicoScope 3000 series oscilloscope will collect data much faster than a PC can read it. To compensate for this, the oscilloscope stores a block of data in an internal memory buffer, delaying transfer to the PC until the required number of data points have been sampled. This is called **block mode**. At very low sampling rates, you may want to switch to **streaming mode**. This allows data to be transferred back to the PC as it is collected.

3.2 More on block mode

In block mode, the computer prompts an oscilloscope from the PicoScope 3000 series to collect a block of data into its internal memory. When the PicoScope 3000 series oscilloscope has collected the whole block, it will signal it is ready, and transfer the whole block into computer memory via the USB port.

The maximum number of values depends upon the size of the oscilloscope's memory. A PicoScope 3000 series oscilloscope can sample at a number of different rates. These rates correspond to the maximum clock frequency multiplied by 0.5, 0.25, 0.125, and so on. There are 14 sampling rates.

There is a separate memory buffer for each channel. When a channel is unused, its memory can be utilised by the enabled channels. On the faster models, one input can be routed to two circuits in the oscilloscope, thus doubling the effective sampling rate of a single channel.

The driver for a PicoScope 3000 series oscilloscope normally performs a number of setup operations before collecting each block of data. This can take up to 50 milliseconds. If it is necessary to collect data with the minimum time interval between blocks, avoid calling setup functions between calls to [ps3000_run_block\(\)](#), [ps3000_ready\(\)](#), [ps3000_stop\(\)](#) and [ps3000_get_values\(\)](#).

3.3 More on streaming mode

In streaming mode, the computer prompts the PicoScope 3000 series oscilloscope to start collecting data. The data is then transferred back to the PC without being stored in oscilloscope memory. Data can be sampled with a period of between 1 ms and 60 s.

Data can be transferred by the oscilloscope's driver to a computer program either in normal or windowed mode. In normal mode, any data collected since the last data transfer operation is returned in its entirety. In windowed mode, a fixed number (n) of samples is returned, where the oldest samples may have already been returned before. Normal mode is useful if the computer program requires fresh data on every transfer. Windowed mode is useful when the program requires a constant time frame of data.

Once the oscilloscope is collecting data in streaming mode, any setup changes (for example, changing a channel range or AC/DC setting in the PicoScope software application) will cause a restart of the data stream. The driver can buffer up to 32K samples of data per channel, but the user must ensure that the [ps3000_get_values\(\)](#) function is called frequently enough to avoid buffer overrun.

The [ps3000_get_times_and_values\(\)](#) function will always return FALSE (0) in streaming mode.

3.4 Triggering

The PicoScope 3000 series oscilloscope can either start collecting data immediately, or it can be programmed to wait for a trigger event to occur. In either case, you need to use the [ps3000_set_trigger\(\)](#) function. A trigger event can occur when the channel A or B input crosses a threshold voltage, or when an external trigger input crosses a threshold voltage. The trigger event can be either a rising or a falling edge.

The external trigger input uses the same physical connection as the signal generator output, so these two functions cannot be used at the same time. It is possible, however, to use the output from the signal generator as a trigger. Triggering is available in block mode only. Any calls to the [ps3000_set_trigger\(\)](#) function has no effect in streaming mode.

3.5 ETS (Equivalent Time Sampling)

ETS is a way of increasing the effective sample rate when working with repetitive signals. It is not possible to use ETS with one-shot signals. ETS is controlled via use of the [ps3000_set_trigger\(\)](#) and [ps3000_set_ets\(\)](#) functions. ETS is available in block mode only. Calls to the [ps3000_set_trigger\(\)](#) function have no effect in streaming mode. As ETS will return random time intervals, the [ps3000_get_times_and_values\(\)](#) function must be used. The [ps3000_get_values\(\)](#) function will return FALSE (0).

3.6 Voltage ranges

It is possible to set the gain for each channel with the [ps3000_set_channel\(\)](#) function. This will give an input voltage range of between 100mV and 20V. The external trigger has a fixed input range of 20V.

3.7 AC/DC operation

Using the [ps3000_set_channel](#) function, each channel can be set to either AC or DC coupling. When AC coupling is used, any DC component of the signal is filtered out.

3.8 Oversampling

When the oscilloscope is operating in block mode at speeds that are less than maximum, it is possible to oversample. Oversampling is taking more than one measurement during a time interval and returning an average. This reduces the effects of noise, and increases the resolution of the oscilloscope.

3.9 Scaling

Oscilloscopes from the PicoScope 3000 series are 8-bit. To take advantage of noise reduction due to oversampling (if enabled), the oscilloscope driver returns 16-bit readings. The following table shows the relationship between the reading from the driver and the voltage of the signal.

Reading	Voltage
-32767	Minimum
0	Zero volts
32767	Maximum

3.10 Signal generator

PicoScope 3000 series oscilloscopes have a built-in signal generator which is set using [ps3000_set_siggen\(\)](#). It produces a selection of accurate frequencies from 1Hz to 1MHz. The waveform can be set to sine, square or triangle and can be swept back and forth in frequency. These are selected under software control.

The signal generator output is physically the same as the external trigger input, so these two functions cannot be used at the same time. It is possible, however, to use the output from the signal generator as a trigger. Note: The signal generator function is not available on all PicoScope 3000 series oscilloscopes.

3.11 Combining oscilloscopes

With PicoLog or your own program it is possible to collect data using up to four oscilloscopes from the PicoScope 3000 series at the same time. Each oscilloscope must be connected to a separate USB port. If a USB hub is used it must be a powered hub. The [ps3000_open_unit\(\)](#) function returns a handle to an oscilloscope. All of the other functions require this handle for oscilloscope identification. For example, to collect data from two oscilloscopes at the same time:

```
handle1 = ps3000_open ()
handle2 = ps3000_open ()

ps3000_set_channel (handle1)
... set up unit 1
ps3000_run(handle1)

ps3000_set_channel (handle2)
... set up unit 2
ps3000_run(handle2)

ready = FALSE
while not ready
    ready = ps3000_ready (handle1)
    ready &= ps3000_ready (handle2)

ps3000_get_values(handle1)
ps3000_get_values(handle2)
```

Note: It is not possible to synchronise the collection of data between oscilloscopes when you are using them in combination.

4 Using different modes

4.1 Introduction

The previous section on advanced features aimed to supply the programmer with extended information on PicoScope 3000 series oscilloscopes. The C sample program, `ps3000con.c`, demonstrates how to use the functions of the driver software, and includes examples showing how to use each of the modes available.

4.2 Using block mode

This is the general procedure for reading and displaying data in block mode:

- 1 Open the oscilloscope using `ps3000_open_unit`
- 2 Select channel ranges and AC/DC switches using `ps3000_set_channel`
- 3 Using `ps3000_set_trigger`, set the trigger if required
- 4 Using `ps3000_get_timebase`, select timebases until the required ns per sample is located
- 5 If required, set the signal generator frequency using `ps3000_set_siggen`
- 6 Start the oscilloscope running using `ps3000_run_block`
- 7 Wait until the oscilloscope says it is ready using `ps3000_ready`
- 8 Transfer the block of data from the oscilloscope using `ps3000_get_values` or `ps3000_get_times_and_values`
- 9 Display the data
- 10 Repeat steps 6 to 9
- 11 Stop the oscilloscope using `ps3000_stop`

4.3 Using streaming mode

This is the general procedure for reading and displaying data in streaming mode:

- 1 Open the oscilloscope using `ps3000_open_unit`
- 2 Select channel ranges and AC/DC switches using `ps3000_set_channel`
- 3 Start the oscilloscope running using `ps3000_run_streaming`
- 4 Transfer the block of data from the oscilloscope using `ps3000_get_values`
- 5 Display the data
- 6 Repeat steps 3 to 5 as necessary
- 7 Stop the oscilloscope using `ps3000_stop`

4.4 Using ETS mode

This is the general procedure for reading and displaying data in ETS mode:

- 1 Open the oscilloscope using `ps3000_open_unit`
- 2 Select channel ranges and AC/DC switches using `ps3000_set_channel`
- 3 Using `ps3000_set_trigger`, set the trigger if required
- 4 Set ETS mode using `ps3000_set_ets`
- 5 Start the oscilloscope running using `ps3000_run_block`
- 6 Wait until the oscilloscope says it is ready using `ps3000_ready`
- 7 Transfer the block of data from the oscilloscope using `ps3000_get_times_and_values`
- 8 Display the data

- 9 Repeat steps 5 to 8 as necessary
- 10 Stop the oscilloscope using ps3000_stop

5 Technical reference

5.1 Driver

The Windows 98SE/ME/2K/XP/2003 32-bit driver, `picopp.sys`, is installed in Windows. It is loaded using an inf file, `picopp.inf`.

Once you have installed the PicoScope and PicoLog software, and the PicoScope 3000 oscilloscope is plugged in for the first time, Windows will automatically install the driver.

5.2 Functions

5.2.1 `ps3000_open_unit`

```
short ps3000_open_unit ( void )
```

This function opens an oscilloscope from the PicoScope 3000 series. The API driver can support up to four oscilloscopes.

Arguments	None.
Returns	<code>-1</code> if the oscilloscope fails to open, <code>0</code> if no oscilloscope is found or a positive (<code>>0</code>) handle to the device opened.

5.2.2 `ps3000_get_unit_info`

```
short ps3000_get_unit_info ( short handle,
                             char * string,
                             short string_length,
                             short info )
```

This function writes oscilloscope information to a character string. If the oscilloscope fails to open, only infos 0 and 6 are available to explain why the last open unit call failed.

Arguments	<p><code>handle</code>, the handle to the device from which info is required. If an invalid handle is passed, the error code from the last unit that failed to open is returned from line 2.</p> <p><code>* string</code>, a pointer to the character string buffer in the calling function where the unit information string (selected with line) will be stored. If a null pointer is passed, no information will be written.</p> <p><code>string_length</code>, the length of the character string buffer. If the string is not long enough to accept all of the information, only the first <code>string_length</code> characters are returned.</p> <p><code>info</code>, is an enumerated type specifying what information is required from the driver.</p>
Returns	The length of the string written to the character string buffer, string, by the function. If one of the parameters are out of range, or a null pointer is passed for string, zero will be returned.

<code>info</code>	Description	Example
PS3000_DRIVER_VERSION (0)	Returns the version number of the DLL used by the oscilloscope driver.	"1, 0, 0, 2"
PS3000_USB_VERSION (1)	Returns the type of USB connection that is being used to connect the oscilloscope to the computer.	"1.1" or "2.0"
PS3000_HARDWARE_VERSION (2)	Returns information about what is the hardware version of the attached oscilloscope.	"1"
PS3000_VARIANT_INFO (3)	Returns information about what model of PicoScope 3000 series oscilloscope is attached to the computer.	"3206"
PS3000_BATCH_AND_SERIAL (4)	Returns the batch and serial number of the oscilloscope.	"CMY66/052"
PS3000_CAL_DATE (5)	Returns the calibration date of the oscilloscope.	"21Oct03"
PS3000_ERROR_CODE (6)	Returns one of the Error codes .	"4"

5.2.3 ps3000_flash_led

```
short ps3000_flash_led ( short handle )
```

Flashes the LED on the front of the oscilloscope three times and returns within one second.

Arguments	<code>handle</code> , the handle of the PicoScope 3000 series oscilloscope.
Returns	<code>1</code> if a valid handle is passed, <code>0</code> if not.

5.2.4 ps3000_close_unit

```
short ps3000_close_unit ( short handle )
```

Shuts down a PicoScope 3000 series oscilloscope.

Arguments	<code>handle</code> , the handle, returned by ps3000_open_unit , of the oscilloscope being closed.
Returns	<code>1</code> if a valid handle is passed, <code>0</code> if not.

5.2.5 ps3000_set_channel

```
short ps3000_set_channel ( short handle,
                           short channel,
                           short enabled,
                           short dc,
                           short range )
```

Specifies if a is to be enabled, the position of the AC\DC switch and the input range.

Arguments	<code>handle</code> , the handle to the required device. <code>channel</code> , an enumerated type, use PS3000_CHANNEL_A (0), PS3000_CHANNEL_B (1), PS3000_CHANNEL_C (2) or PS3000_CHANNEL_D (3). <i>Channels C and D are not available on all models.</i> <code>enabled</code> , specify if the channel is active: TRUE=active, FALSE=inactive. <code>dc</code> , specifies the position of the AC/DC switch: TRUE=DC, FALSE=AC. <code>range</code> , a code between 0 and 12 see the table below.
Returns	<code>0</code> if unsuccessful, or if one or more of the arguments are out of range. <code>1</code> if successful.

Code	Enumeration	Range	
0	PS3000_10MV	$\pm 10\text{mV}$	<i>Not available on all models.</i>
1	PS3000_20MV	$\pm 20\text{mV}$	<i>Not available on all models..</i>
2	PS3000_50MV	$\pm 50\text{mV}$	<i>Not available on all models.</i>
3	PS3000_100MV	$\pm 100\text{mV}$	
4	PS3000_200MV	$\pm 200\text{mV}$	
5	PS3000_500MV	$\pm 500\text{mV}$	
6	PS3000_1V	$\pm 1\text{V}$	
7	PS3000_2V	$\pm 2\text{V}$	
8	PS3000_5V	$\pm 5\text{V}$	
9	PS3000_10V	$\pm 10\text{V}$	
10	PS3000_20V	$\pm 20\text{V}$	
11	PS3000_50V	$\pm 50\text{V}$	<i>Not available on all units.</i>

5.2.6 ps3000_get_timebase

```
short ps3000_get_timebase ( short handle,
                            short timebase,
                            long no_of_samples,
                            long * time_interval_ns,
                            short * time_units,
                            short oversample,
                            long * max_samples )
```

This function discovers which timebases are available on the oscilloscope. This function should be called after channel and ETS options have been set.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>timebase</code>, a code between 0 and the maximum timebase (variant dependant). Timebase 0 is the fastest timebase, timebase 1 is twice the time per sample as timebase 0, timebase 2 is four times, etc.</p> <p><code>no_of_samples</code>, the number of samples required. This value is used to calculate the most suitable time unit to use.</p> <p><code>time_interval_ns</code>, a pointer to the time interval, in ns, between readings at the selected timebase. If a null pointer is passed, nothing will be written here.</p> <p><code>time_units</code>, a pointer to the most suitable time units to return data in, when calling <code>ps3000_get_times_and_values</code>. If a null pointer is passed, nothing will be written here.</p> <p><code>oversample</code>, the amount of oversample required. An oversample of 4 would quadruple the time interval and quarter the maximum samples but would increase the effective resolution by one bit. the increase in resolution given by oversampling can be calculated from the equation below:</p> $\text{Increase in resolution (bits)} = (\log(\text{oversample})) / (2 \log(2))$ <p><code>max_samples</code>, A pointer to the maximum samples available. The maximum samples may vary depending on the number of channels enabled, the timebase chosen and the oversample selected. If this pointer is null, nothing would be written here.</p>
Returns	<p><code>1</code> if all parameters are in range, otherwise <code>0</code>.</p>

5.2.7 `ps3000_set_siggen`

```
long ps3000_set_siggen (    short handle,
                            short wave_type,
                            long start_frequency,
                            long stop_frequency,
                            short increment,
                            short dwell_time,
                            short repeat,
                            short dual_slope)
```

This function is used to enable or disable the [signal generator](#) and sweep functions. Sweep functions are not available if the oscilloscope is in [streaming mode](#).

The signal generator is not available on the PicoScope 3204 oscilloscope model. A 1kHz square wave output is made available and can be set with this function. See remarks and [specifications](#) for more information.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>type</code>, the type of wave, choose <code>PS3000_SINE</code> (0), <code>PS3000_SQUARE</code> (1) or <code>PS3000_TRIANGLE</code> (2). <i>This argument has no effect if used with the PicoScope 3204 variant.</i></p> <p><code>start_frequency</code>, the required frequency, in the range $0 < \text{freq} < 1\text{MHz}$, to start the sweep or the frequency generated in a non-sweep mode. <code>0</code> switches the signal generator off.</p> <p><code>stop_frequency</code>, the required stop frequency of the sweep, in the range $0 < \text{freq} < 1\text{MHz}$ but not necessarily greater than <code>start_frequency</code>. If the start and stop frequencies are the same, the signal generator will be run with a constant frequency. <i>This argument has no effect if used with the PicoScope 3204 variant..</i></p> <p><code>increment</code>, The size of the steps to increment or decrement the frequency by in a sweep mode. This must always be positive, the start and stop frequencies will determine whether to increment or decrement. This must be a frequency in the range $0.1\text{Hz} < \text{increment} < \text{stop_frequency} - \text{start_frequency}$. This is not used in a non-sweep mode. <i>This argument has no effect if used with the PicoScope 3204 variant..</i></p> <p><code>dwell_time</code>, This is the time, in ms, to wait before increasing the frequency by <code>increment</code> in a sweep mode. This is unused in a non-sweep mode. <i>This argument has no effect if used with the PicoScope 3204 variant..</i></p> <p><code>repeat</code>, <code>TRUE</code> restarts the sweep when the <code>stop_frequency</code> is reached, <code>FALSE</code> continues indefinitely at <code>stop_frequency</code> when it is reached. <i>This argument has no effect if used with the PicoScope 3204 variant..</i></p> <p><code>dual_slope</code>, if <code>repeat</code> is <code>TRUE</code> this specifies what to do at the <code>stop_frequency</code>. <code>TRUE</code> will sweep back towards the <code>start_frequency</code>, <code>FALSE</code> will restart the sweep from <code>start_frequency</code>. <i>This argument has no effect if used with the PicoScope 3204 variant..</i></p>
Returns	The actual frequency, in Hz, or start frequency, in Hz, that is generated. Zero if one of the parameters are not in range.

Remarks

The PicoScope 3204 oscilloscope model has a simple, 1kHz square wave, signal generator for scope probe calibration. Therefore only two arguments of this function have any effect:

To switch the square wave on, use a valid `handle` and set `start_frequency` to a non-zero value.
To switch the square wave off, use a valid `handle` and set `start_frequency` to 0.

5.2.8 `ps3000_set_ets`

```
long ps3000_set_ets ( short handle,
                      short mode,
                      short ets_cycles,
                      short ets_interleave)
```

This function is used to enable or disable ETS and to set the ETS parameters.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>mode</code>, <code>PS3000_ETS_OFF (0)</code> - disables ETS, <code>PS3000_ETS_FAST(1)</code> - enable ETS and provides ets_cycles of data, which may contain data from previously returned cycles, <code>PS3000_ETS_SLOW (2)</code> - enable ETS and provide fresh data every ets_cycles cycles. <code>PS3000_ETS_SLOW</code> takes longer to provide each data set, but the data sets are more stable and unique.</p> <p><code>ets_cycles</code>, Specifies the number of cycles to store: the computer can then select ets_interleave cycles to give the most uniform spread of samples. <code>ets_cycles</code> should be between two and five times the value of <code>ets_interleave</code>.</p> <p><code>ets_interleave</code>, Specifies the number of ETS interleaves to use. If the sample time is 20ns and the interleave 10, the approximate time per sample will be 2 ns.</p>
Returns	<p>If ETS is enabled, the effective sample time will be returned. Zero if ETS is disabled or one of the parameters is out of range.</p>

5.2.9 `ps3000_set_trigger`

```
short ps3000_set_trigger ( short handle,
                           short source,
                           short threshold,
                           short direction,
                           short delay,
                           short auto_trigger_ms)
```

This function is used to enable or disable triggering and its parameters. Triggering is not available in streaming mode.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>source</code>, specifies which to look for a trigger on, use <code>PS3000_CHANNEL_A (0)</code>, <code>PS3000_CHANNEL_B (1)</code>, <code>PS3000_CHANNEL_C (2)</code>, <code>PS3000_CHANNEL_D (3)</code>, <code>PS3000_EXTERNAL(4)</code> or <code>PS3000_NONE(5)</code>. <i>Channels C and D are not available on all models.</i></p> <p><code>threshold</code>, the threshold that the trigger event is to take place. This is scaled in 16bit ADC counts at the currently selected range. If an external trigger is enabled the range is fixed at +/-20V.</p> <p><code>direction</code>, use <code>PS3000_RISING(0)</code> or <code>PS3000_FALLING (1)</code>.</p> <p><code>delay</code>, This specifies the delay, as a percentage of the requested number of data points, between the trigger event and the start of the block. It should be in the range -100% to +100%. Thus, 0% means that the first data value in the block, and -50% means that the trigger event is in the middle of the block.</p> <p><code>auto_trigger_ms</code>, the delay in ms after which the oscilloscope will collect samples if no trigger event occurs. If this is set to zero the oscilloscope will wait for a trigger indefinitely.</p>
Returns	<p>0 if one of the parameters are out of range, otherwise 1.</p>

5.2.10 ps3000_run_block

```
short ps3000_run_block (    short handle,
                           long no_of_samples,
                           short timebase,
                           short oversample,
                           long * time_indisposed_ms)
```

This function tells the oscilloscope to start collecting data in [block mode](#).

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>number of samples</code>, the number of samples to return.</p> <p><code>timebase</code>, a code between 0 and 14. Timebase 0 is the fastest timebase, timebase 1 is twice the time per sample as timebase 0, timebase 2 is four times, and so on. For the maximum sampling rate, see the specifications. Note: This parameter is ignored during operation in ETS mode.</p> <p><code>oversample</code>, the oversample factor, a number between 1 and 256.</p> <p><code>time_indisposed_ms</code>, a pointer to the <code>time_indisposed_ms</code>. This is the approximate time, in ms, over which the ADC will collect data. If a trigger is set, it is the amount of time the ADC takes, in ms, to collect a block of data after a trigger event, calculated as sample interval x number of points required. Note: The actual time may differ from computer to computer, depending on how fast the computer can respond to IO requests.</p>
Returns	<code>0</code> if one of the parameters are out of range, otherwise <code>1</code> .

5.2.11 ps3000_run_streaming

```
short ps3000_run_streaming (    short handle,
                               short time_interval_ms,
                               long max_samples,
                               short windowed)
```

This function tells the oscilloscope to start collecting data in [streaming mode](#). If this function is called when a trigger has been enabled, the trigger settings will be ignored.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>time_interval_ms</code>, the time interval, in ms, between data points. This can be no shorter than 1ms.</p> <p><code>max_samples</code>, the maximum number of samples that the driver is to store. This can be no greater than 60000. It is the caller's responsibility to retrieve data before the oldest values are overwritten.</p> <p><code>windowed</code>, if this is <code>0</code>, only the values taken since the last call to get values are returned. If this is <code>1</code>, the number of values requested by <code>get_values</code> are returned, even if they have already been read by ps_get_values.</p>
Returns	<code>1</code> if streaming has been enabled correctly, otherwise <code>0</code> if a problem occurred or a value was out of range.

5.2.12 ps3000_ready

```
short ps3000_ready ( short handle)
```

This function checks to see if the oscilloscope has finished the last data collection operation. This function does nothing if the oscilloscope is in streaming mode.

Arguments	<code>handle</code> , the handle to the required device.
Returns	<p><code>1</code> (meaning 'ready') is returned when the oscilloscope has collected a complete block of data or the auto trigger timeout has been reached. If an invalid handle is passed or if the oscilloscope is in streaming mode it returns <code>0</code> (meaning 'not ready').</p> <p><code>-1</code> (meaning 'device not attached') is returned if the endpoint transfer fails indicating that the unit may well have been unplugged.</p>

5.2.13 ps3000_stop

```
void ps3000_stop ( short handle)
```

Call this function to stop the oscilloscope from sampling data. If this function is called before a trigger event occurs, the oscilloscope may not contain valid data.

Arguments	<code>handle</code> , the handle to the required device.
Returns	<code>0</code> if an invalid handle is passed, otherwise <code>1</code> .

5.2.14 ps3000_get_values

```
long ps3000_get_values( short handle
                        short * buffer_a,
                        short * buffer_b,
                        short * buffer_c,
                        short * buffer_d,
                        short * overflow,
                        long no_of_values)
```

This function is used to get values. This function does nothing if ETS triggering is enabled.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>buffer_a</code>, this is a pointer to the buffer to which data from A will go. It is unused if the oscilloscope is collecting only from channel B. If the pointer is <code>NULL</code>, nothing will be written here.</p> <p><code>buffer_b</code>, this is a pointer to the buffer to which data from B will go. It is unused if the oscilloscope is collecting only from channel A. If the pointer is <code>NULL</code>, nothing will be written here.</p> <p><code>buffer_c</code>, for future use -set to <code>NULL</code>.</p> <p><code>buffer_d</code>, for future use -set to <code>NULL</code>.</p> <p><code>overflow</code>, a bit pattern indicating whether an overflow has occurred on a channel, bit 0 overflow on channel A, bit 1 is an overflow on channel B, bit 2 is an overflow on channel C and bit 3 is an overflow on channel D. Where bit0 is the LSB.</p> <p><code>no_of_values</code>. The number of data points to return. In streaming mode, this is the maximum no of values to return.</p>
Returns	<p>The actual number of data values per channel returned which may be less than the <code>no_of_values</code> if streaming. <code>FALSE</code> is returned if one of the parameters are out of range.</p>

5.2.15 ps3000_get_times_and_values

```
long ps3000_get_times_and_values( short handle
                                  long * times,
                                  short * buffer_a,
                                  short * buffer_b,
                                  short * buffer_c,
                                  short * buffer_d,
                                  short * overflow,
                                  short time_units,
                                  long no_of_values)
```

This function is used to get values and times. This function will not return any valid times if the oscilloscope is in [streaming mode](#). This function is essential for ETS operation.

Arguments	<p><code>handle</code>, the handle to the required device.</p> <p><code>times</code>, this is a pointer to the buffer for the times. Each time is the interval between the trigger event and the corresponding sample. Times before the trigger event are negative, and times after the trigger event are positive.</p> <p><code>buffer_a</code>, this is a pointer to the buffer to put data from channel A into. It is unused if the oscilloscope is collecting only from channel B. If the pointer is <code>NULL</code>, nothing will be written here.</p> <p><code>buffer_b</code>, this is a pointer to the buffer to put data from channel B into, It is unused if the oscilloscope is collecting only from channel A. If the pointer is <code>NULL</code>, nothing will be written here.</p> <p><code>buffer_c</code>, for future use -set to <code>NULL</code>.</p> <p><code>buffer_d</code>, for future use -set to <code>NULL</code>.</p> <p><code>overflow</code>, a bit pattern indicating whether an overflow has occurred on a channel, bit 0 overflow on channel A, bit 1 is an overflow on channel B, bit 2 is an overflow on channel C and bit 3 is an overflow on channel D. Where bit0 is the LSB.</p> <p><code>time_units</code>, which can be one of: <code>PS3000_FS (0)</code>, <code>PS3000_PS (1)</code>, <code>PS3000_NS (2)</code>, <code>PS3000_US (3)</code>, <code>PS3000_MS (4)</code> or <code>PS3000_S (5)</code> which are femto seconds, pico seconds, nano seconds (default), micro seconds, milli seconds and seconds respectively.</p> <p><code>no_of_values</code>. The number of data points to return. In streaming mode, this is the maximum no of values to return.</p>
Returns	<p>The actual number of data values per channel returned which may be less than the <code>no_of_values</code> if streaming.</p> <p><code>0</code> is returned if one or more of the parameters are out of range or if the times will overflow with the <code>time_units</code> requested. Use <code>ps3000_get_timebase</code> in order to acquire the most suitable <code>time_units</code>.</p>

5.3 Programming

5.3.1 C

There are two C example programs: one is a simple GUI application, and the other is a more comprehensive console mode program that demonstrates all of the facilities of the driver.

The GUI example program is a generic Windows application - ie it does not use Borland AppExpert or Microsoft AppWizard. To compile the program, create a new project for an Application containing the following files:

`ps3000.c`;
`ps3000.rc`; and

`ps3000bc.lib` (Borland 32-bit applications); or
`ps3000.lib` (Microsoft Visual C 32-bit applications)

The following files must be in the compilation directory:

```
ps3000.rch;  
ps3000.h;
```

and the following file must be in the same directory as the executable.

```
ps3000.dll
```

The console example program is a generic windows application - ie it does not use Borland AppExpert or Microsoft AppWizard. To compile the program, create a new project for an Application containing the following files:

```
ps3000con.c; and
```

```
ps3000bc.lib (Borland 32-bit applications); or  
ps3000.lib (Microsoft Visual C 32-bit applications).
```

The following files must be in the compilation directory:

```
ps3000.h;
```

and the following file must be in the same directory as the executable.

```
ps3000.dll
```

5.3.2 Visual Basic

The Win32 sub-directory contains the following files:

```
ps3000.vbp - project file  
ps3000.bas - procedure prototypes  
ps3000 frm - form and program
```

Note: The functions which return a TRUE/FALSE value, return 0 for FALSE and 1 for TRUE, whereas Visual basic expects 65535 for TRUE. Check for > 0 rather than =TRUE.

5.3.3 Delphi

The program `ps3000.dpr` demonstrates how to operate oscilloscopes from the PicoScope 3000 series. The file `ps3000.inc` contains procedure prototypes that you can include in your own programs. Other required files include `ps300fm.res`, `ps300fm.dfm` and `ps300fm.pas`. This has been tested with Delphi versions 3.

5.3.4 Excel

- 1 Load the spreadsheet `ps3000.xls`
- 2 Select **Tools | Macro**
- 3 Select **GetData**
- 4 Select **Run**

Note: The Excel Macro language is similar to Visual Basic. The functions which return a TRUE/FALSE value, return 0 for FALSE and 1 for TRUE, whereas Visual Basic expects 65535 for TRUE. Check for > 0 rather than =TRUE.

5.3.5 HP-Vee

The example function `ps3000.vee` is in the drivers sub-directory. It uses procedures that are defined in `ps3000.vh`. It was tested using HP-Vee version 5.

5.4 Glossary

AC/DC switch

To switch from alternating current to direct current, or vice versa, select AC or DC from the control on the oscilloscope toolbar of the PicoScope software application. The setting should be adjusted to suit the characteristics of the input signal.

Analog bandwidth

The input frequency at which the signal amplitude has fallen by 3 dB from its nominal value.

Block mode

A sampling mode in which the computer prompts the oscilloscope to collect a block of data into its internal memory before stopping the oscilloscope and transferring the whole block into computer memory. This mode of operation is effective when the input signal being sampled is high frequency. Note: To avoid aliasing effects, the maximum input frequency must be less than half the sampling rate.

Buffer size

The size of the oscilloscope buffer memory. The buffer memory is used by the oscilloscope to temporarily store data. This helps to compensate for the differences in data transfer rate from one device to another.

Device Manager

Device Manager is a Windows applet that displays the current hardware configuration of your computer. On Windows 98 or Windows ME, right click on 'My Computer' and choose the 'Device Manager' tab. On Windows 2000 or Windows XP, right click on 'My Computer,' choose the 'Hardware' tab and select 'Device Manager'.

Driver

A software application that controls a piece of hardware. The driver for the PicoScope 3000 series of oscilloscopes is supplied in the form of a 32 bit Windows DLL. This is used by the PicoScope and PicoLog software and also by custom programs to control oscilloscopes from the PicoScope 3000 series.

ETS

ETS stands for Equivalent Time Sampling. ETS constructs a picture of a repetitive signal by accumulating information over many wave cycles. This means the oscilloscope can capture fast repeating signals that have a higher frequency than the maximum sampling rate. Note: ETS should not be used for one-shot or non-repetitive signals.

External trigger

This is the BNC socket marked **Ext** on a PicoScope 3000 series oscilloscope. This connection point is shared with the signal generator output. It can be used to start a data collection run but cannot be used to record data. The external trigger input is physically the same as the signal generator output, so these two functions cannot be used at the same time. It is possible, however, to use the output from the signal generator as a trigger.

Maximum sampling rate

A figure indicating the maximum number of samples the oscilloscope is capable of acquiring per second. Maximum sample rates are usually given in MS/s (mega samples per second) or GS/s (giga samples per second.) The higher the sampling capability of the oscilloscope, the more accurate the representation of the fine details in a fast signal.

PC oscilloscope

A generic term used to describe a PicoScope 3000 series oscilloscope and the PicoScope software application.

PicoLog software

This is a software product that accompanies all our oscilloscopes. It turns your PC into a data logger and chart recorder.

PicoScope 3000 series

An oscilloscope range comprising the PicoScope 3204, PicoScope 3205, and PicoScope 3206 oscilloscopes.

PicoScope software

This is a software product that accompanies all our oscilloscopes. It turns your PC into an oscilloscope, spectrum analyzer, and meter display.

Range

This allows you to specify an input voltage range.

Signal generator

This is a feature on an oscilloscope which allows a signal to be generated without an external input device being present. The signal generator output is the BNC socket marked **Ext** on the oscilloscope. If you connect a BNC cable between this, and one of the channel inputs, you can send a signal down one of the channels. On some units, the signal generator can generate a simple TTL square wave, on others it can generate a sine, square or triangle wave that can be swept back and forth. Consult the [specifications](#) for further details.

Note: The signal generator output is physically the same as the external trigger input, so these two functions cannot be used at the same time. It is possible, however, to use the output from the signal generator as a trigger.

Streaming mode

A sampling mode in which the oscilloscope samples data and returns it to the computer in an unbroken stream. This mode of operation is effective when the input signal being sampled is low frequency.

Timebase

The timebase controls the time interval across the scope display. If you select **Timebase is time per division** in the preferences dialog box, this works like a traditional bench top scope. There are ten divisions across the screen, so the total time interval is ten times the timebase.

Trigger bandwidth

The maximum frequency at which the trigger circuit will reliably generate a trigger event.

USB 1.1

USB is the abbreviation for Universal Serial Bus. This is a standard port that enables you to connect external devices to PCs. A typical USB 1.1 port supports a data transfer rate of 12 Mbps (12 mega bits per second), and is much faster than a serial port.

USB 2.0

USB is the abbreviation for Universal Serial Bus. This is a standard port that enables you to connect external devices to PCs. A typical USB 2.0 port supports a data transfer rate that is 40 times faster than that supported by USB 1.1. An additional feature of USB 2.0 is that it is backwards compatible with USB 1.1.

Vertical resolution

A value, in bits, indicating the degree of precision with which the oscilloscope can turn input voltages into digital values. Calculation techniques can improve the effective resolution.

6 Troubleshooting

6.1 FAQ

Consult this section if you are a PicoScope or PicoLog user. If you are writing your own program, refer to the [error codes](#) section.

PicoLog reports error code 1.

This error is reported when more than 4 oscilloscopes are opened on one machine. It is not possible to use more than 4 oscilloscopes with PicoLog.

PicoScope or PicoLog reports error code 2.

This error is reported when the driver cannot allocate enough of the computer's memory to operate the oscilloscope. Consult the [system requirements](#) section for more information.

PicoScope or PicoLog reports error code 3.

This error indicates that an oscilloscope from the PicoScope 3000 series could not be found on your machine. Make sure the software is installed before the oscilloscope is plugged into the USB socket and restart your computer.

Ensure that mention of the oscilloscope can be found in Device Manager under the USB section of this facility. If the oscilloscope is not mentioned here, consult Pico Technical Support for further advice.

PicoScope or PicoLog reports error code 4, 5 or 6.

This error is reported when there is problem with the oscilloscope itself, these problems could arise from configuration settings being corrupted, or a firmware or hardware error.

Unplug the oscilloscope, wait a few seconds, and reconnect it to the USB port. If the error is still reported, consult Pico Technical Support for further advice.

PicoScope or PicoLog reports error code 7.

This error is reported if the operating system is not recent enough to support the PicoScope 3000 series oscilloscope. Consult the [system requirements](#) section for more information.

6.2 Error codes

This section is aimed at those people who intend to write their own programs for use with the driver. A description of the driver error codes is given below. If the PicoScope or PicoLog software reports an error, refer to the [FAQ](#).

Code	Enumeration	Description
0	PS3000_OK	The oscilloscope is functioning correctly.
1	PS3000_MAX_UNITS_OPENED	Attempts have been made to open more than PS3000_MAX_UNITS.
2	PS3000_MEM_FAIL	Not enough memory could be allocated on the host machine.
3	PS3000_NOT_FOUND	An oscilloscope could not be found.
4	PS3000_FW_FAIL	Unable to download firmware.
5	PS3000_NOT RESPONDING	The oscilloscope is not responding to commands from the PC.
6	PS3000_CONFIG_FAIL	The configuration information in the oscilloscope has become corrupt or is missing.
7	PS3000_OS_NOT_SUPPORTED	Need to use Windows 98 (or later) or Windows 2000 (or later)

Index

- A -

AC coupling 11
AC/DC switch 10, 13, 17
ADC 10, 11
Aliasing 11
API 3, 15, 17, 18, 19, 20, 21, 22, 23, 24
API functions 3

- B -

Block mode 10, 11, 13, 22
BNC connector 3
Buffer 10, 11

- C -

C 13
C programming 25
Calibration 5
Channel 10, 11, 13, 17, 18, 21, 23
Clock frequency 10
Contact details 7

- D -

Data logger 3
DC coupling 11
Delphi programming 26
Device Manager 30
Driver 10, 11, 13, 15, 17, 18, 19, 20, 21, 22, 23, 24, 25, 27, 30

- E -

Error code 1 30
Error code 2 30
Error code 3 30
Error code 4 30
Error code 5 30
Error code 6 30
Error code 7 30

ETS 11, 13, 18, 20, 23, 24
ETS mode 13
Excel macros 26
External trigger 3, 11, 12, 21

- F -

FAQ 30
Functions 10, 11, 12, 13, 15, 17, 18, 19, 20, 21, 22, 23, 24

- G -

Gain 11

- H -

High speed 3, 10
HP-Vee 27

- M -

Macros in Excel 26
Maximum input range 5
Memory buffer 10, 11
Multi-unit operation 12

- N -

Normal mode 10

- O -

One-shot signal 11
Oscilloscope probe 3
Oversampling 11

- P -

PC oscilloscopes 3, 6
Pico Technical Support 30
PicoLog software 3, 15, 30
picopp.inf 15
picopp.sys 15
PicoScope 3000 series 3, 5, 6, 9, 10, 11, 12, 13, 15, 17, 19, 21, 22, 23, 26, 30

PicoScope software 3, 15, 30
Pre-trigger 11
Programming in C 13, 25
Programming in Delphi 26
Programming in Visual Basic 26

- R -

Repair 5
Resolution 11

- S -

Safety warning 6
Sample rate 11
Samples 11
Sampling 11
Sampling clock 11
Scope probe 3
Signal generator 3, 11, 12, 13, 19
Software control 12
Spectrum analyzer 3
Square wave 3
Streaming mode 10, 13
Sweep 12
System requirements 9

- T -

Technical Support 30
Test equipment 5
Threshold voltage 11
Time interval 11
Time interval 11
Timebase 18, 22
Trigger 11
Trigger event 11
Triggering 11

- U -

USB 3
USB 9
USB 1.1 3, 9
USB 2.0 3, 9
USB hub 12
USB port 30

- V -

Visual Basic programming 26

- W -

Windowed mode 10

Pico Technology Ltd

The Mill House
Cambridge Street
St Neots PE19 1QB
United Kingdom
Tel: +44 (0) 1480 396 395
Fax: +44 (0) 1480 396 296
Web: www.picotech.com