# Using PicoScope PC Oscilloscopes for Fast USB Data Acquisition

The PicoScope 6 software and PicoScope drivers, supplied with all PicoScope oscilloscopes, have recently been improved to give faster data transfer over the USB interface. The improvements have focused on 'streaming mode', one of the two data transfer modes used by PicoScope devices (the other is 'block mode'). When you use PicoScope 6, you rarely need to know the difference between the two modes as the software switches automatically between them. If, on the other hand, you are writing your own software using the PicoScope Application Programming Interface (API), it helps to know that you can capture large volumes of data at high speed by selecting streaming mode. This note explains how streaming mode works, and what scope performance you can expect when using PicoScope 6 or the PicoScope API.

> " The PicoScope 6 software and PicoScope drivers are supplied with all PicoScope oscilloscopes. "
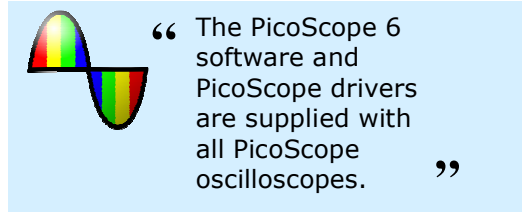
CONTENTS

1. Fast USB streaming in PicoScope 6

In streaming mode, PicoScope 6 tells the scope to capture data continuously and, at the same time, to transfer each sample to the PC with minimum delay. The amount of streamed data that PicoScope 6 can store is limited by design, to prevent the program from consuming too much of the PC's resources. For example, with a 4000 Series scope, PicoScope 6 can create a streamed data file of up to 20 M samples per channel.

PicoScope 6 limits the sampling rate to 1 MS/s in streaming mode, again to conserve the PC's resources. If you request a faster sampling rate, then the program switches to block mode. PicoScope 6 does this mode switching automatically, but you can change the collection time at which the transition happens by going to the *Preferences* menu (Figure 1).
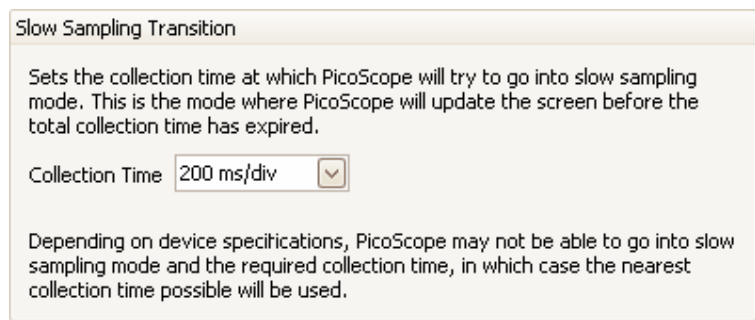
Figure 1 – PicoScope 6 *Preferences / Sampling* dialog

2. PicoScope 6 streaming specifications

As the following table shows, some scopes can sample faster than others in streaming mode. PicoScope 6 limits the sampling rate to avoid overloading the USB interface or slowing down the PC, whilst obtaining enough data to display a clear scope trace at an acceptable update rate.

Oscilloscope series	Maximum specifications	
	Record length	Sampling rate
PicoScope 2000 2-channel	2 MS	1 MS/s
PicoScope 2000 handheld	65 kS	1 kS/s
PicoScope 3000 8-bit	65 kS	1 kS/s
PicoScope 3224	2 MS	1 MS/s
PicoScope 3424	2 MS	1 MS/s
PicoScope 4000	20 MS	1 MS/s
PicoScope 5000	20 MS	1 MS/s

3. Fast USB streaming in the PicoScope API

When you write your own application using the PicoScope API, you can choose whether to use block mode or streaming mode. If the amount of data that you need to capture will fit in the scope's buffer memory, then you should use block mode as this will give you the widest choice of sampling rates. If record length is more important to you than sampling rate, then you should choose streaming mode.

> " Once the application has its own copy of the data, it can create a record whose length is limited only by the resources of the PC. "

In streaming mode, just like PicoScope 6, the PicoScope driver tells the scope to capture data continuously and then to transfer each sample or small group of samples to the PC with minimum delay. The driver stores the data in its own buffer and then sends a signal to the application. The application must then copy the data before the driver overwrites the buffer with new data. Once the application has its own copy of the data, it can create a record whose length is limited only by the resources of the PC.

There are several factors that limit the performance of the scope in streaming mode. One is that the scope's buffer memory is simultaneously receiving sampled data and transmitting it to the USB interface, so the maximum sampling rate for some scope types may be less than in block mode. The other limitations are the transfer speeds of the USB 2.0 interface and the driver. However,

Figure 2 – The main data bottlenecks: the scope buffer, the USB interface and the driver

development work on the PicoScope hardware and drivers has been steadily pushing these speeds upwards. PicoScope oscilloscopes can transfer data at a typical rate of 20 MS/s in streaming mode, with the exact rate depending on the capability of the PC and the drivers.
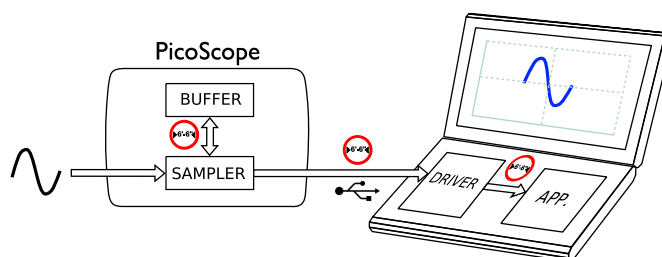
Using PicoScope PC Oscilloscopes for Fast USB Data Acquisition

4. PicoScope API streaming specifications

The theoretical maximum speed of the USB interface for streaming data, after deducting all overheads, is about 40 MB/s. Under optimum conditions, with no unnecessary processes running, our tests have achieved speeds of about 30 MB/s. This is equivalent to 30 MS/s for 8-bit scopes or 15 MS/s for 12-bit scopes. As the following table shows, some scopes can sample faster than this in streaming mode. To avoid overflowing their internal buffer, you can use data aggregation to compress the sampled data to a rate that can be transmitted on the bus. (Data aggregation is explained below.) In some cases it may still not be possible to sample for sustained periods at the highest sampling rates listed here.

Oscilloscope series	Maximum specifications	
	Buffer size [1]	Sampling rate
PicoScope 2000 2-channel	2 MS	6.6 MS/s [2]
PicoScope 2000 handheld	65 kS	1 kS/s
PicoScope 3000 8-bit	65 kS	1 kS/s
PicoScope 3224	500 kS	6.6 MS/s [2]
PicoScope 3424	1 MS	6.6 MS/s [2]
PicoScope 4000	4.2 GS	80 MS/s [2]
PicoScope 5000	4.2 GS	125 MS/s [2]

Notes:

[1] This is the longest record that the driver can store in its own buffer. User applications can create records of unlimited length.
[2] Sampling rate is shared between channels.

5. Aggregation

There are some situations in which we might want to reduce the number of samples after capturing them in streaming mode. This can occur in PicoScope 6 when the scope has captured more samples than the number of pixels across the display. It can also occur with the PicoScope API, for reasons that will be unique to each application. All of these cases call for some method of data reduction.

Instead of just throwing away the excess samples (which is called *decimation:* see Figure 3b), it is better to use signal processing to preserve as much of the original information as possible. In general, this type of processing is called *aggregation*. PicoScope does this by recording the minimum and maximum values of the missing samples, a process called *peak detection* (see Figure 3c). Whereas decimated data can give a false impression of smoothness and sometimes the added problem of aliasing (the creation of spurious frequencies that were not in the original signal), peak-detected data always shows the full range of values. Some PicoScope products do the peak-detection in the oscilloscope hardware, while others do it in the Pico driver software. Whichever method your scope uses, the PicoScope hardware and software are designed to give you optimal performance without throwing away vital information.
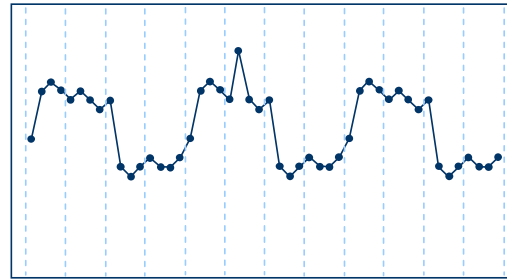
6. References

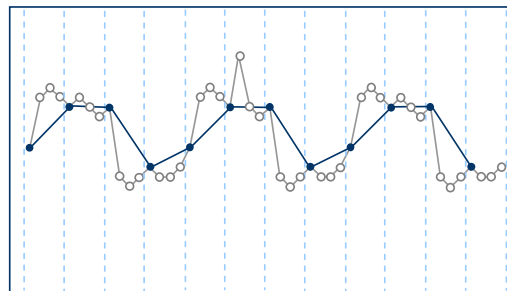*Universal Serial Bus Specification Version 2.0*, April 27, 2000

7. Downloads

Download oscilloscope data sheets, Programmer's Guides, drivers and example source code from:
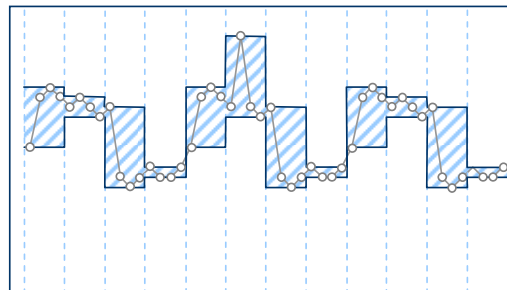
www.picotech.com

a) Original sampled data. Note the glitch in the second pulse.

b) Data decimated by 3 in 4. The glitch has been lost in the decimation process.

c) Peak-detected data (method used by PicoScope). Although the shape of the glitch is lost, its peak value is still visible.

Figure 3 – Two different data aggregation methods

Terminology
API Application Programming Interface
GS   gigasamples
MB   megabytes
MS   megasamples