Find Glitches Faster with PicoScope

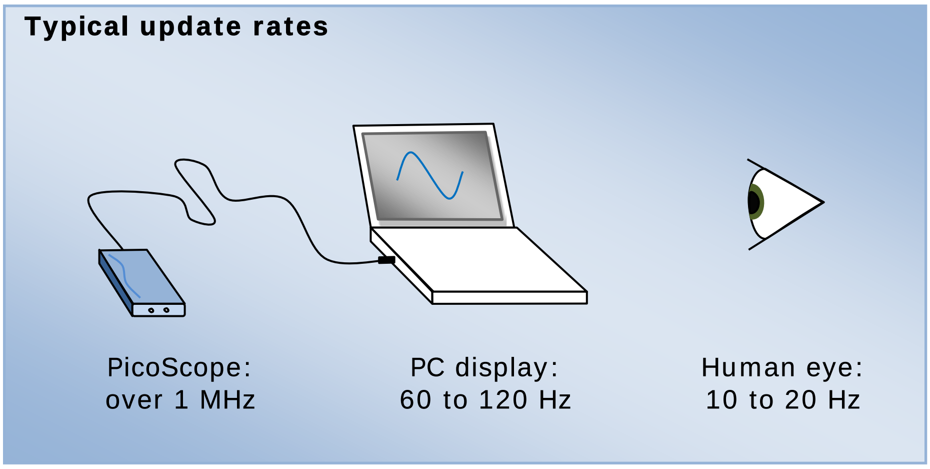
1. Introduction

Using an oscilloscope to search for a rare event can be a hit-or-miss experience. We recently compared several capture methods offered by PicoScope to find a 1-in-1-million glitch, with different waiting times in each case. Recent updates to PicoScope have increased the capture rate, which along with other advanced features helps to capture and find intermittent events like glitches.

2. Capture rates of PicoScope oscilloscopes

At the last count there were 50 PicoScope real-time oscilloscopes, all with different speeds and other varying characteristics. What unites them, and all other digital oscilloscopes, is that the data acquisition hardware that converts the analog signals to digital data is much faster than the display hardware that draws waveforms on the screen.

The update rate required for an oscilloscope display is usually assumed to be between 10 and 20 times per second. This is the frame rate that the human eye finds comfortable when watching video. The upper limit is determined by the refresh rate of the screen, which is normally between 60 and 120 times per second.



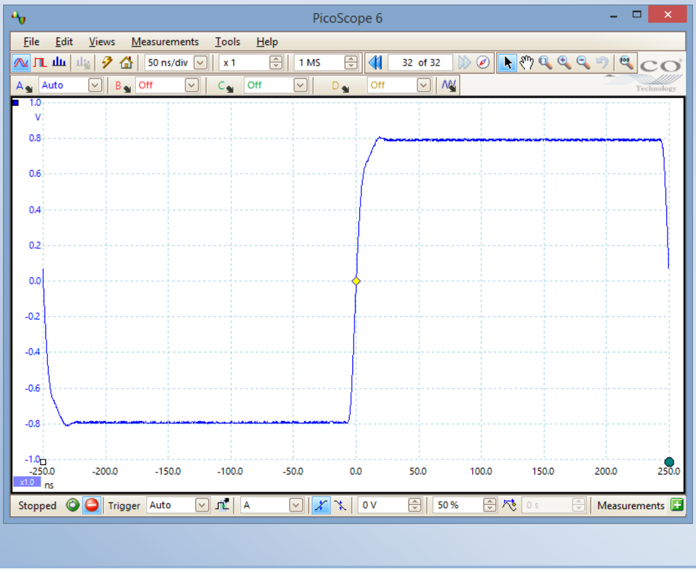
The data acquisition hardware can often capture millions of waveforms per second. Clearly, the digital oscilloscope needs to reduce the amount of captured data before displaying it, but how can it do this without throwing away important information? The answer is: it's up to you, the user. If you understand how to use the scope to filter out unwanted data, you will be able to maximize your chances of finding intermittent events such as glitches.

3. Real-time display mode

The most straightforward way to capture a glitch is to use real-time display mode. The scope's trigger is armed to capture one waveform, the waveform is displayed, and then the trigger is rearmed ready to capture another waveform. PicoScope operates in this way when you select Auto, Repeat or Single triggering mode. The drawback to this approach is that the scope spends only a small fraction of its time acquiring the signal, the rest being wasted while the scope updates the display. If an important event occurs in the signal during the display update, it will be missed.

Real-time display mode example (time to spot a 1-in-1-million event)

Input signal:	2 MHz square wave
Timebase:	50 ns/div, 500 ns sweep time
Capture rate:	20 Hz
Display update rate:	20 Hz
Proportion of events displayed:	20 Hz / 2 MHz = 0.001%
Glitch repetition period:	10^6 / 2 MHz = 500 ms
Mean time to see glitch:	500 ms / 0.001% = **50,000 s (~ 14 hours)**



The above example shows that, even with PicoScope's optimized 20 Hz display update rate, only a tiny proportion of the input signal ever appears on the display. Even if the glitch does appear, you might not see it after waiting for 14 hours.

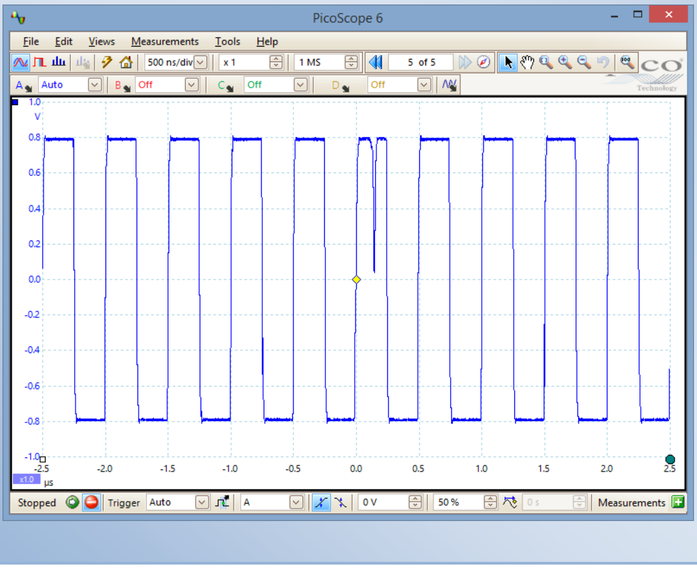
In some cases you can set up an advanced trigger to isolate the glitch, but this is possible only if you know what kind of signal you are looking for.

Tip: displaying multiple cycles per capture

With deep-memory scopes, you can reduce the mean time to see the glitch by slowing down the timebase to display multiple cycles. In the example above, we could slow down the timebase by a factor of ten to 500 ns/div. PicoScope's large, high-resolution display has a major advantage over the small screens of most benchtop instruments as it allows you to view multiple cycles in fine detail.

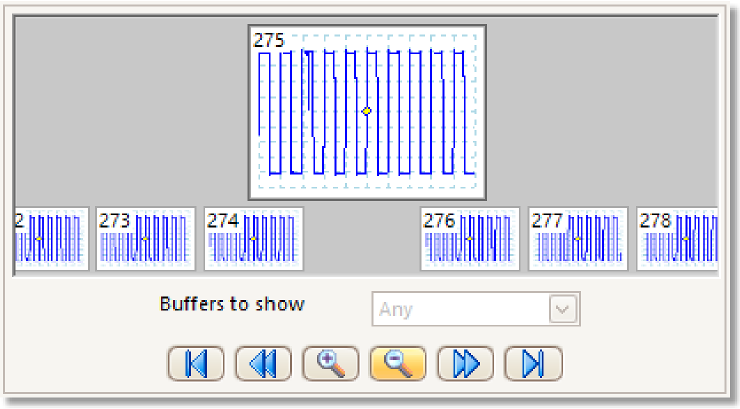
Real-time display mode example (time to spot a 1-in-1-million event)
Variation with longer timebase

Input signal:	2 MHz square wave
Timebase:	500 ns/div, 5 µs sweep time
Capture rate:	20 Hz
Display update rate:	20 Hz
Proportion of events displayed:	10 × 20 Hz / 2 MHz = 0.01%
Glitch repetition period:	10^6 / 2 MHz = 500 ms
Mean time to see glitch:	500 ms / 0.01% = **5000 s (~ 1.4 hours)**



As expected, we have reduced the mean time to see the glitch by a factor of ten.

If you view the screen for an hour you are likely to see the glitch flash past on the screen, but before you can stop the scope, the glitch has gone. Fortunately this is less of a problem with a PicoScope, as waveforms are stored in a circular buffer so that you can "rewind time" over hundreds of stored waveforms. If you stop the scope quickly enough, the glitch will still be in the buffer:

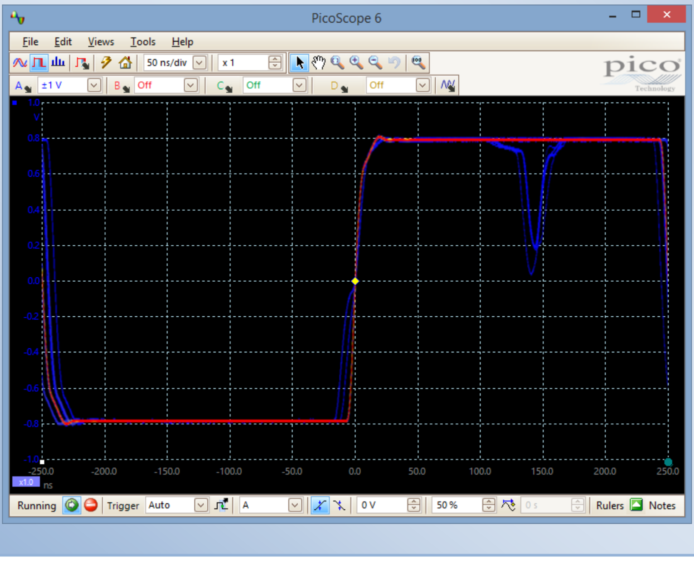


4. Persistence display mode

Persistence display mode overlays a large number of waveforms on the display, using various methods of color-coding or shading to distinguish frequent and infrequent data. You can arrange for old data to fade away after a specified time or to remain on the display until you erase it. Persistence mode has two advantages: it allows the scope to capture waveforms faster than it can update the display, and it makes it easier for you to spot a transient event.

Persistence display mode example (time to spot a 1-in-1-million event)

Input signal:	2 MHz square wave
Timebase:	50 ns/div, 500 ns sweep time
Capture rate:	10,000/s (PicoScope 5444B)
Display update rate:	10 Hz
Proportion of events displayed:	10 kHz / 2 MHz = 0.5%
Glitch repetition period:	10^6 / 2 MHz = 500 ms
Mean time to see glitch:	500 ms / 0.5% = **100 s (~1.7 minutes)**



The hottest colors (such as red) indicate the densest areas of data where the majority of the waveforms are located. The coolest colors (such as blue) indicate transient events such as glitches and jitter.

The capture rate depends on the scope settings but is usually many thousands of waveforms per second. (For even faster capture rates, see *Rapid triggering* below.) The display rate is typically 10 to 20 updates per second in this mode, although this is of little importance because each waveform persists for a long time on the display. In fact, with the persistence time set to infinity, you can leave PicoScope running overnight and still be sure of catching glitches.

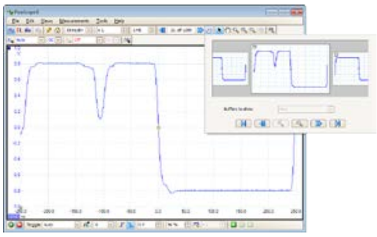
Persistence mode has allowed us to see a glitch that was practically invisible in real-time mode, and gives us a qualitative display of its rate of recurrence, but it cannot tell us exactly how often it occurs. Also, since the display is a composite of many waveforms, it is difficult to measure a single instance accurately. You can, however, work around this by estimating the shape of the glitch and then setting up an advanced trigger to capture a single occurrence.

5. Additional techniques

Here are some of PicoScope's more advanced features that can be used to look for glitches in special situations. Videos on these and other features can be found at:

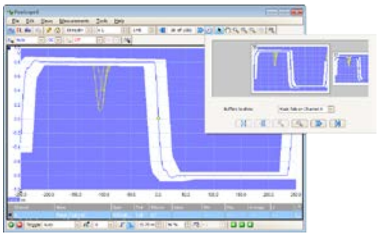
www.youtube.com/user/picotech

Rapid triggering. The scope captures a rapid sequence of events with minimal rearm time between waveforms, so there's less chance of missing a glitch. The captures are stored in the waveform buffer for you to review at leisure. With the fastest PicoScopes you can capture as many as 10,000 waveforms in a 10 ms burst (that's an average rate of 1 million per second).



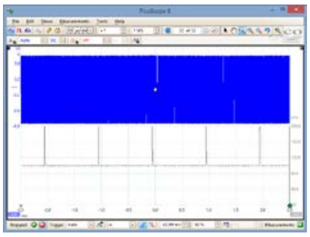
Video: Deep Memory PicoScopes

Mask limit testing. Draw a test mask (either manually or automatically) and PicoScope highlights the waveforms that stray into the mask area. Instead of manually searching the waveform buffer for failures, let PicoScope find them automatically.



Video: Mask Limit Testing

Math channels. Searching for a glitch in a capture of hundreds of megasamples is a time-consuming task. PicoScope can apply math operators to live or stored data to make glitches stand out. For example, the frequency operator *freq()* makes missing or double pulses stand out as spikes in the frequency plot. You can then zoom in to see the abnormal signal.



Video: Using the Maths Channel Feature

6. Conclusion

We recommend that you familiarize yourself with PicoScope's Persistence Display mode, Rapid Trigger mode, Mask Limit Testing and Math Channels. They can all be used to improve your chances of finding an infrequent glitch, particularly when you don't know its shape or frequency of occurrence.

7. Further information

Download oscilloscope Data Sheets, hardware and software User's Guides, Programmer's Guides and Software Development Kits from:

www.picotech.com