

Triggering a PicoScope signal generator using the PicoScope API functions



1 Introduction

Most PicoScope models have either a built-in function generator or an arbitrary waveform generator (AWG). This article provides some guidance on how to call the API functions in order to trigger the signal generator output, which is useful in applications such as indicating that a particular event has occurred.

This article applies to PicoScope oscilloscopes that have signal generator outputs and use one of the following drivers:

- ps2000a
- ps3000a
- ps4000
- ps4000a
- ps5000
- ps5000a
- ps6000

This article will use the PicoScope 3000 Series A API functions as an example.

2 Setting parameters in order to trigger the signal generator

Below is the definition of the `ps3000aSetSigGenBuiltIn` function from the `ps3000aApi.h` header file:

```
PICO_STATUS ps3000aSetSigGenBuiltIn
(
    int16_t                handle,
    int32_t                offsetVoltage,
    uint32_t               pkToPk,
    int16_t                waveType,
    float                  startFrequency,
    float                  stopFrequency,
    float                  increment,
    float                  dwellTime,
    PS3000A_SWEEP_TYPE     sweepType,
    PS3000A_EXTRA_OPERATIONS operation,
    uint32_t               shots,
    uint32_t               sweeps,
    PS3000A_SIGGEN_TRIG_TYPE triggerType,
    PS3000A_SIGGEN_TRIG_SOURCE triggerSource,
    int16_t                extInThreshold
);
```

The following parameters will remain constant:

```
offsetVoltage = 0;
pkToPk = 2000000; // ±1 V output
waveType = PS3000A_SINE;
operation = PS3000A_ES_OFF;
```

When a trigger is specified, then either the number of shots or sweeps, but not both, must be non-zero.

The parameter settings discussed will also apply when calling the `ps3000aSetSigGenArbitrary` function.

Triggering a PicoScope signal generator using the PicoScope API functions



Note: The `ps3000aSetSigGenArbitrary`, `ps3000aSetSigGenBuiltIn`, or `ps3000aSetSigGenBuiltInV2` function must be called before the `ps3000aRunBlock` or `ps3000aRunStreaming` function as it is not possible to start the signal generator during data collection.

2.1 Shots

The signal generator can be triggered to output a specified number of cycles of a waveform. For example, setting:

```
shots = 1;
```

would result in one cycle of a sine wave being output by the signal generator.

Setting:

```
shots = PS3000A_SHOT_SWEEP_TRIGGER_CONTINUOUS_RUN; // 0xFFFFFFFF
```

would cause the signal generator to start and run continuously after the trigger event occurs.

2.2 Sweeps

The signal generator can be triggered to sweep between the start and stop frequencies. Before the trigger event occurs, the signal generator will output at the start frequency.

Setting the following parameters:

```
startFrequency = 100;  
stopFrequency = 1000;  
increment = 100;  
dwellTime = 1;  
sweepType = PS3000A_UP;  
sweeps = 1;
```

would cause the signal generator to sweep up from 100 Hz to 1 kHz with an increment of 100 Hz at 1 second intervals. The frequency would then remain at 1 kHz until the next trigger event.

Note: If the `sweepType` parameter is set to `PS3000A_UPDOWN` or `PS3000A_DOWNUP`, the number of sweeps must be set to an even number to ensure that the signal generator returns to the starting frequency when it reaches the stop frequency.

For example, if `sweepType` is set to `PS3000A_UPDOWN`, using the other parameters defined above with `sweeps` set to 1, the signal generator would only sweep up from 100 Hz to 1 kHz. If `sweeps` is set to 2, this will cause the signal generator to sweep up and then back down to the starting frequency when a trigger event occurs.

If the number of sweeps is set as:

```
sweeps = PS3000A_SHOT_SWEEP_TRIGGER_CONTINUOUS_RUN; // 0xFFFFFFFF
```

then the signal generator will start a sweep and continue sweeping between the start and stop frequencies after the trigger occurs.

Note: If no trigger is set for the signal generator then the number of shots and sweeps will be ignored.

Triggering a PicoScope signal generator using the PicoScope API functions



If a number greater than 0 is set for both shots and sweeps, then the signal generator will sweep between the frequencies as long as the parameters are set correctly.

2.3 Trigger type

The trigger type can be one of four enumeration values:

- PS3000A_SIGGEN_RISING - trigger on rising edge
- PS3000A_SIGGEN_FALLING - trigger on falling edge
- PS3000A_SIGGEN_GATE_HIGH - run while trigger is high
- PS3000A_SIGGEN_GATE_LOW - run while trigger is low

2.4 Trigger source

When using a trigger, the trigger source can be one of the following:

2.4.1 Scope trigger

```
triggerSource = PS3000A_SIGGEN_SCOPE_TRIG;
```

Note: If the Scope trigger is being used, the simple or advanced trigger API functions can be set before or after the appropriate signal generator function is called.

2.4.2 External trigger

Where a PicoScope oscilloscope has an external trigger input, this can be used to trigger the signal generator output.

```
triggerSource = PS3000A_SIGGEN_EXT_IN;
```

The `extInThreshold` parameter is then set to indicate the threshold level (in counts) at which the signal generator will be triggered.

The count range is usually specified in the header file:

```
#define PS3000A_EXT_MAX_VALUE 32767  
#define PS3000A_EXT_MIN_VALUE -32767
```

These values will correspond to the maximum and minimum voltage level for the external trigger input (± 5 V in the case of the PicoScope 3000 series).

In the function call below, the threshold level of 6553 corresponds to approximately 1 V using the following formula:

```
thresholdLevel = (voltageLevel / maxExternalVoltage) *  
PS3000A_EXT_MAX_VALUE
```

Triggering a PicoScope signal generator using the PicoScope API functions



```
status = ps3000aSetSigGenBuiltIn(handle,
    offsetVoltage,
    pkToPk,
    waveType,
    startFrequency,
    stopFrequency,
    0,
    0,
    (PS3000A_SWEEP_TYPE) 0,
    (PS3000A_EXTRA_OPERATIONS) 0,
    1,
    0,
    PS3000A_SIGGEN_RISING,
    PS3000A_SIGGEN_EXT_IN,
    6553);
```

Gated triggers using the external trigger

Using the external trigger input, the signal generator can be triggered to output when the voltage level is above or below a defined threshold. When the signal passes through the threshold in the opposite direction after the signal generator has been activated, the voltage level remains at the previous state.

Calling the `ps3000aSetSigGenBuiltIn` with the following parameters, the signal generator will output a sine wave and sweep up in frequency as long as the input voltage on the external trigger input is above 500 mV:

```
startFrequency = 100.0;
stopFrequency = 1000.0;
increment = 100;
dwellTime = 1;
sweepType = PS3000A_UP;
shots = 0;
sweeps = 0;
triggerType = PS3000A_SIGGEN_GATE_HIGH;
triggerSource = PS3000A_SIGGEN_EXT_IN;
extInThreshold = 3277; // 500 mV
```

Note: Using the example above, if the external trigger input were to drop below and then rise above the threshold, the signal generator would continue to output from the last frequency before the change in event.

2.4.3 Software trigger

The signal generator can also be triggered via software.

Triggering a PicoScope signal generator using the PicoScope API functions



```
triggerSource = PS3000A_SIGGEN_SOFT_TRIG;
```

In this case the `ps3000aSigGenSoftwareControl` function is called to provide a software trigger and cause the signal generator to output 5 cycles of a sine wave:

```
status = ps3000aSetSigGenBuiltIn(handle,
    offsetVoltage,
    pkToPk,
    waveType,
    startFrequency,
    stopFrequency,
    0,
    0,
    (PS3000A_SWEEP_TYPE) 0,
    (PS3000A_EXTRA_OPERATIONS) 0,
    5,
    0,
    PS3000A_SIGGEN_RISING,
    PS3000A_SIGGEN_SOFT_TRIG,
    0);

status = ps3000aSigGenSoftwareControl(handle, 1);
```

Gated triggers using the software trigger

The `ps3000aSigGenSoftwareControl` function can also be called to provide a gated trigger for the signal generator.

Set the `triggerType` parameter to `PS3000A_SIGGEN_GATE_HIGH` or `PS3000A_SIGGEN_GATE_LOW` then call the function with the `state` parameter set to a non-zero value or 0 to activate the signal generator as shown below:

```
triggerType = PS3000A_SIGGEN_GATE_HIGH;
```

To trigger the signal generator, call:

```
status = ps3000aSigGenSoftwareControl(handle, 1);
```

To stop the output of the waveform:

```
status = ps3000aSigGenSoftwareControl(handle, 0);

triggerType = PS3000A_SIGGEN_GATE_LOW;
```

To trigger the signal generator, call:

```
status = ps3000aSigGenSoftwareControl(handle, 0);
```

To stop the output of the waveform:

```
status = ps3000aSigGenSoftwareControl(handle, 1);
```

As with the external trigger input, the voltage level will remain at the state it was at when the `ps3000aSigGenSoftwareControl` function was called in order to stop the output of the waveform.

Triggering a PicoScope signal generator using the PicoScope API functions



3 Other information

3.1 Changing signal generator properties while waiting for a trigger

While waiting for a trigger, it is possible to change certain properties such as the start and stop frequency/delta phase using the `ps3000aSetSigGenPropertiesArbitrary` or `ps3000aSetSigGenPropertiesBuiltIn` functions (depending on the signal generator function called).

It is not possible to change the offset or peak to peak voltages, or the waveform.

3.2 Triggering the signal generator to output a DC voltage

It is not possible to trigger the function generator in order to output a DC voltage. Instead define a waveform with a single value corresponding to another voltage level, e.g. 0 V, and set the rest of the values to correspond to the maximum amplitude (in this case 32767).

Set the peak to peak value such that the maximum amplitude values will be output at the correct voltage level, e.g. use 2000000 for a 1 V 'DC voltage'.

When the arbitrary waveform function is called, the output will correspond to the initial value of the waveform. When the trigger occurs, the first sample will be output, followed by the samples at the maximum amplitude value.