

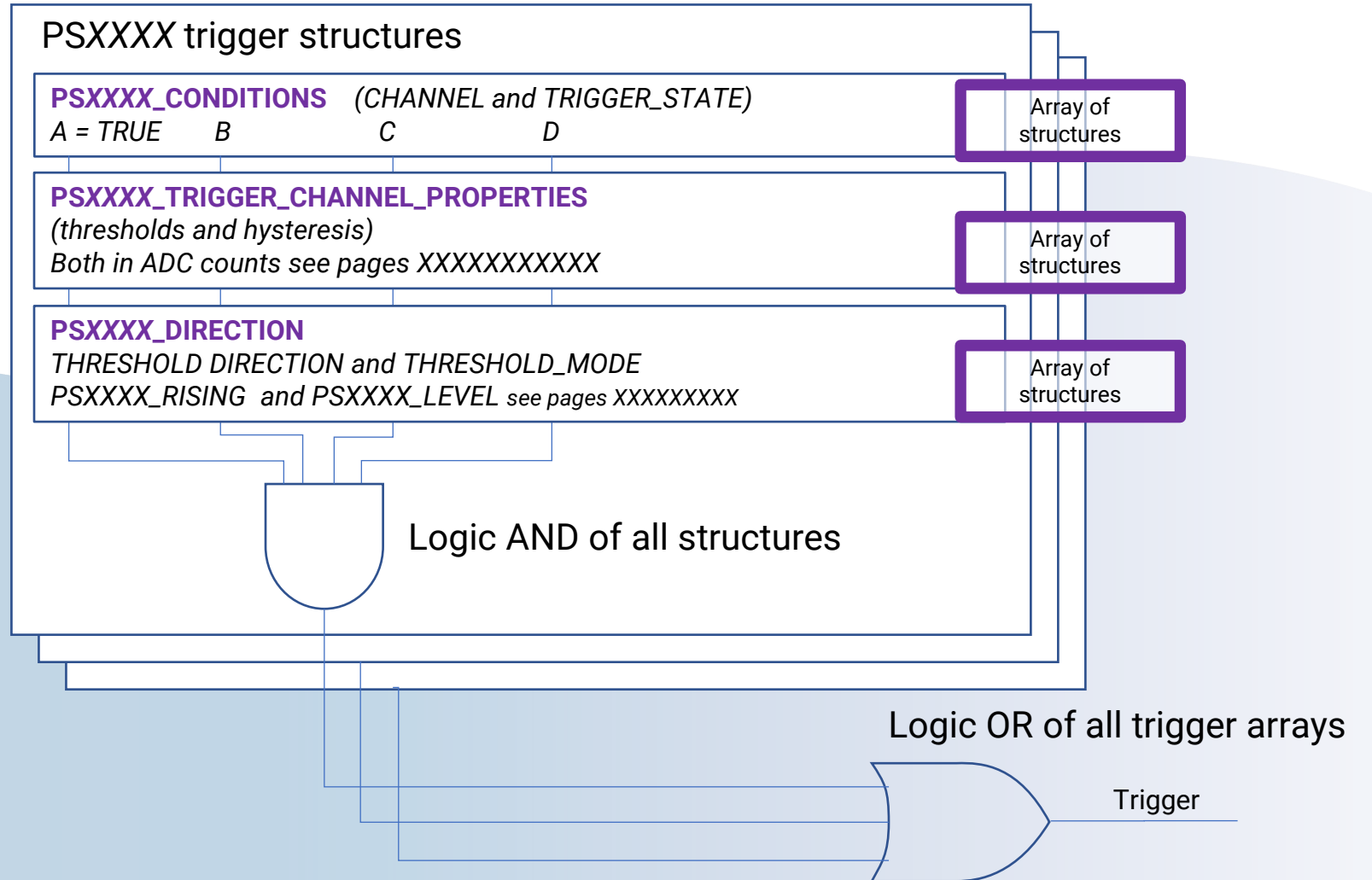
# SDK Advanced Triggers

## Advanced Triggers – Setup and Logic

A trigger event can occur when one of the signal or trigger input channels crosses a threshold voltage on either a rising or a falling edge, or when a more complex time-qualified condition occurs. It is also possible to combine multiple analog and digital inputs and time-qualified conditions using the logic trigger function.

The format to configure an advanced trigger is detailed below. You first must form an array of each type of structure. The trigger logic is the logical AND of structures in the array. Then call the related function, passing in each array. For example, if a rising edge is set for both A and B in same array, they are logically ANDed and a rising edge must be seen on both A and B for the scope to trigger. For a logical OR, create an array for each ORed input and call the related function, passing each array. So for a 4-channel-input logic OR, create four sets of arrays for each structure type and pass the corresponding arrays related to each function.

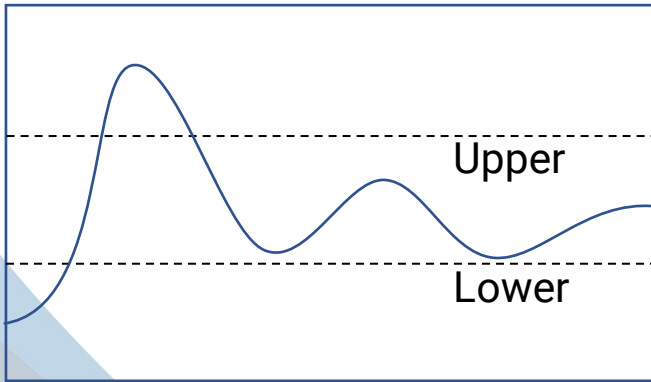
```
Form an array of structures:
PSXXX_XXXX array[] = {
PSXXX_XXXX structure1
PSXXX_XXXX structure2,
PSXXX_XXXX structureN
}
```



Note: Some driver/APIs have V2 trigger structures and functions. It is now recommended to use these for new examples.

This general overview of how the trigger APIs work. Please refer to the specific API programmer's guide for your model of PicoScope.

- TRIGGER\_CHANNEL\_PROPERTIES structure



In the following example waveforms both thresholds are in the same place expect the POSITIVE\_RUNT example.

- **Note: Some of our example SDK code like the C# examples use different names for the thresholds:**

- Upper is called Major
- Lower is called Minor

- **Note: Hysteresis values are specified in 'scaled ADC counts' (16-bit variables). So you should use values for your scope's actual hardware resolution, x per ADC step;**

- 8-bit scope - (scaled  $\pm 32\ 512$ , +7F00 hex), 254 per ADC step
- 12-bit scope - (scaled  $\pm 32\ 767$ , +7FFF hex), 16 per ADC step
- 14-bit scope - (scaled  $\pm 32\ 767$ , +7FFF hex), 4 per ADC step
- 15-bit scope - (scaled  $\pm 32\ 767$ , +7FFF hex), 2 per ADC step

- Upper and lower thresholds

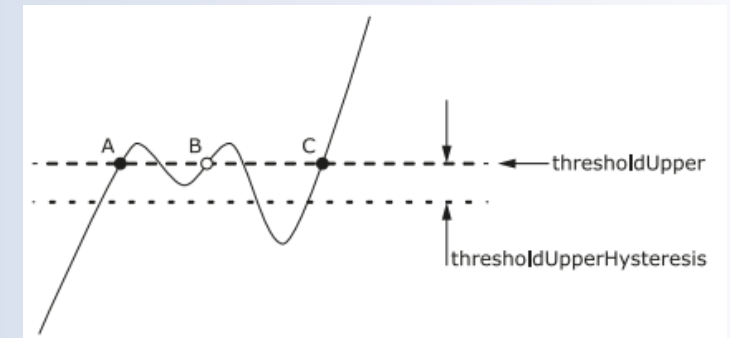
The digital triggering hardware in your PicoScope has two independent trigger thresholds called *upper* and *lower*. For some trigger types you can freely choose which threshold to use. See PSXXX\_THRESHOLD\_DIRECTION on the following pages for a list of trigger types and the thresholds that they support. Dual thresholds are used for pulse-width triggering, when one threshold applies to the level trigger and the other to the pulse-width qualifier; and for window triggering, when the two thresholds define the upper and lower limits of the window. Each threshold has its own trigger and hysteresis settings.

- Hysteresis

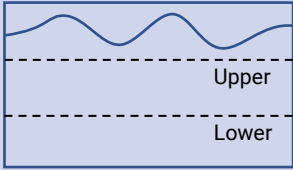
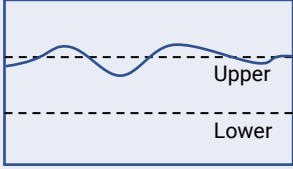
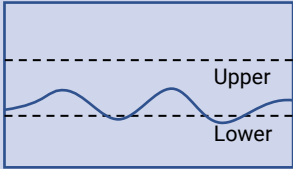
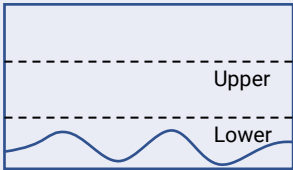
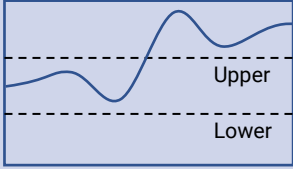
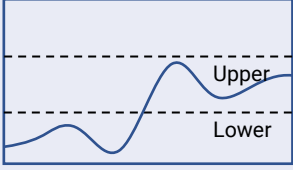
Each trigger threshold (upper and lower) has an accompanying parameter called hysteresis. This defines a second threshold at a small offset from the main threshold. The trigger fires when the signal crosses the trigger threshold, but will not fire again until the signal has crossed the hysteresis threshold and then returned to cross the trigger threshold. The double-threshold mechanism prevents noise on the signal from causing unwanted trigger events.

For a rising-edge trigger the hysteresis threshold is below the trigger threshold. After one trigger event, the signal must fall below the hysteresis threshold before the trigger is enabled for the next event. Conversely, for a falling-edge trigger, the hysteresis threshold is always above the trigger threshold. After a trigger event, the signal must rise above the hysteresis threshold before the trigger is enabled for the next event.

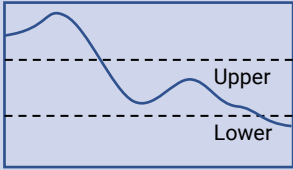
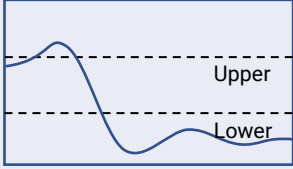
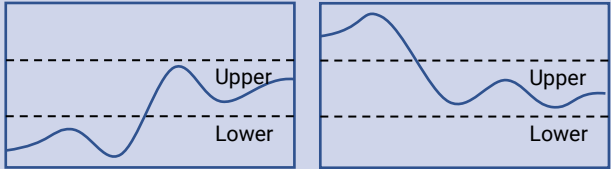
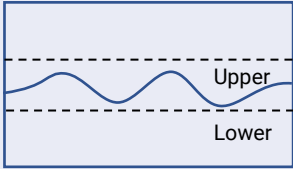
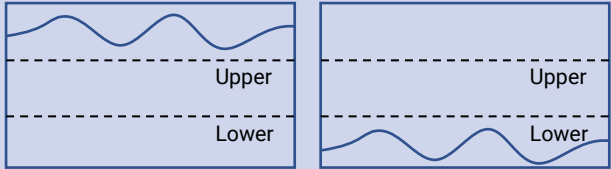
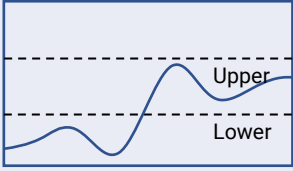
The trigger fires at **A** as the signal rises past the trigger threshold. It does not fire at **B** because the signal has not yet dipped below the hysteresis threshold. The trigger fires again at **C** after the signal has dipped below the hysteresis threshold and risen again past the trigger threshold.



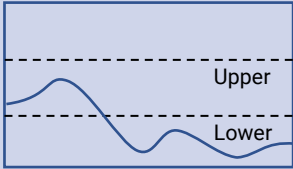
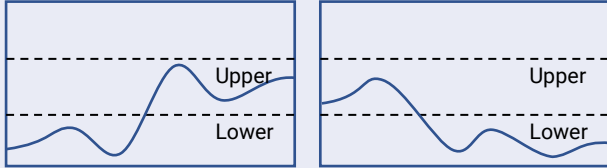
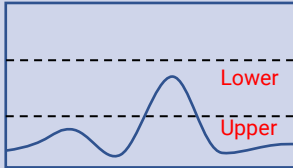
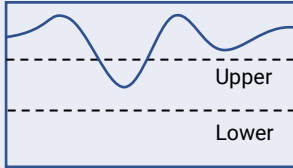
- THRESHOLD\_DIRECTION constants

Constant	Trigger Type	Threshold	Polarity	Example Waveform
PSXXXX_ABOVE	Gated	Upper	Above	
PSXXXX_ABOVE_LOWER	Gated	Lower	Above	
PSXXXX_BELOW	Gated	Upper	Below	
PSXXXX_BELOW_LOWER	Gated	Lower	Below	
PSXXXX_RISING	Threshold	Upper	Rising	
PSXXXX_RISING_LOWER	Threshold	Lower	Rising	

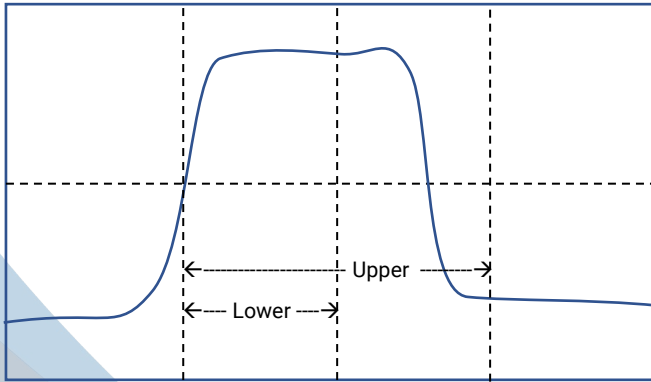
- THRESHOLD\_DIRECTION constants

Constant	Trigger Type	Threshold	Polarity	Example Waveform(s)
PSXXXX_FALLING	Threshold	Upper	Falling	
PSXXXX_FALLING_LOWER	Threshold	Lower	Falling	
PSXXXX_RISING_OR_FALLING	Threshold	Lower -> Upper ->	For Rising For Falling	
PSXXXX_INSIDE	Window-qualified	Both	Inside	
PSXXXX_OUTSIDE	Window-qualified	Both	Outside	
PSXXXX_ENTER	Window	Both	Entering	

- THRESHOLD\_DIRECTION constants

Constant	Trigger Type	Threshold	Polarity	Example Waveform(s)
PSXXXX_EXIT	Window	Both	Falling	
PSXXXX_ENTER_OR_EXIT	Window	Both	Falling	
PSXXXX_POSITIVE_RUNT	Window-qualified	Both	Entering from below the window	 <p>A portion of a signal that crosses <i>Upper</i> threshold and then subsequently crosses <i>Upper</i> threshold again without first crossing <i>Lower</i> threshold.</p>
PSXXXX_NEGATIVE_RUNT	Window-qualified	Both	Entering from above the window	 <p>Same as above</p>
PSXXXX_NONE	None	None	None	

- Pulse-width qualifier



*In the following examples waveforms are all using a rising edge trigger.*

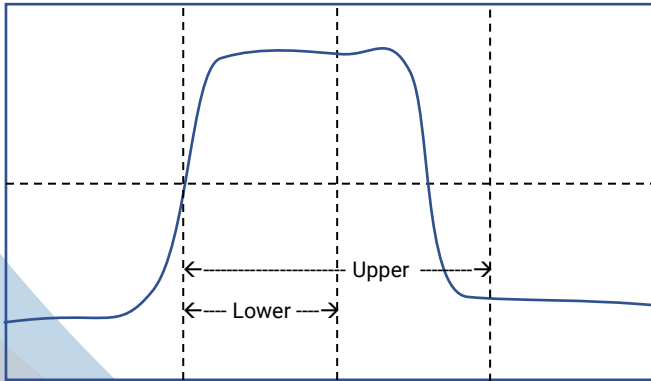
- **Direction Argument**

The direction of the signal required for the pulse width trigger to fire. See `PXXXX_THRESHOLD_DIRECTION` constants for the list of possible values.

Each channel of the oscilloscope (except the EXT input) has two thresholds for each direction—for example, `PSXXXX_RISING` and `PSXXXX_RISING_LOWER`—so that one can be used for the pulse-width qualifier and the other for the level trigger.

The driver will not let you use the same threshold for both triggers; so, for example, you cannot use `PSXXXX_RISING` as the direction argument for both `psXXXXSetTriggerConditions` and `psXXXXSetPulseWidthQualifier` at the same time. There is no such restriction when using window triggers.

- Pulse-width qualifier



*In the following examples waveforms are all using a rising edge trigger.*

- PWQ - Upper and lower thresholds

The pulse width, delay and drop-out triggering methods additionally require the use of the pulse-width qualifier functions:

**psXXXXSetPulseWidthQualifierProperties**

**psXXXXSetPulseWidthQualifierConditions**

**psXXXXSetPulseWidthQualifierDirections**

- Function Arguments

**psXXXXSetPulseWidthQualifierProperties**

*lower*, the lower limit of the pulse-width counter, in samples. This argument is required for all pulse width types.

*upper*, the upper limit of the pulse-width counter, in samples. This argument is used only when the type is PSXXXX\_PW\_TYPE\_IN\_RANGE or PSXXXX\_PW\_TYPE\_OUT\_OF\_RANGE.

*type*, the type of pulse width trigger. See PSXXXX\_PULSE\_WIDTH\_TYPE.

**psXXXXSetPulseWidthQualifierConditions**

*conditions*, a list of PSXXXX\_CONDITION structures

*nConditions*, the number of values in the conditions list

*info*, whether to add this condition to the existing definition or clear the definition and start a new one. See PSXXXX\_CONDITIONS\_INFO.

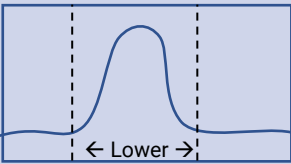
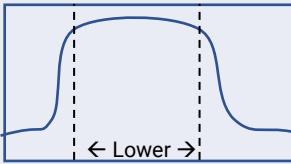
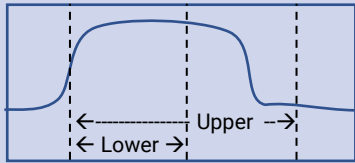
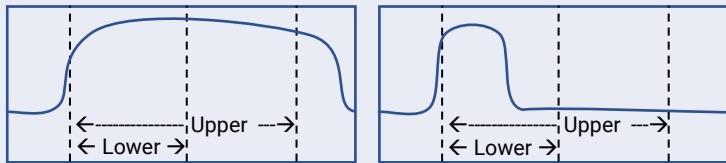
**psXXXXSetPulseWidthQualifierDirections**

*directions*, a list of PSXXXX\_DIRECTION structures specifying which direction to apply to each trigger source

*nDirections*, the number of items in the directions list.



- PSXXXX\_PULSE\_WIDTH\_TYPE constants

Constant	Time Threshold(s)	Range	Example Waveform(s)
PSXXXX_PW_TYPE_LESS_THAN	Lower	Less Than	
PSXXXX_PW_TYPE_GREATER_THAN	Lower	Greater Than	
PSXXXX_PW_TYPE_IN_RANGE	Both	Inside both	
PSXXXX_PW_TYPE_OUT_OF_RANGE	Both	Outside both	
PSXXXX_PW_TYPE_NONE	None	None	