

## Things to consider when designing with the Kemet SS-430 sensor module

Aug-19

Recently Kemet released the series of their proximity (sometimes called motion) sensors modules:

**SS-430** - without lens (DK p/n [399-20620-ND](#))

and 3 parts with the lens:

**SS-430L-BK** – with Black lens (DK p/n [399-20623-ND](#))

**SS-430L-N** – with Natural lens (DK p/n [399-20621-ND](#))

**SS-430L-W** – with White lens (DK p/n [399-20622-ND](#))



No Lens



Natural Lens



White Lens



Black Lens

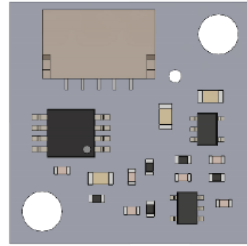
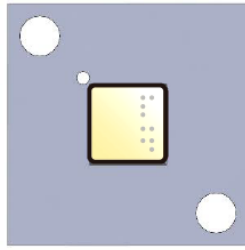
All these modules are based on the proximity sensor chip **PL-N823-01** (DKE p/n [399-17913-1-ND](#)) which is Pyroelectric Infrared Sensor (aka PIR - Passive Infrared or IR motion sensors) and exists in the market for some time already. The sensor is based on Tokin (a vendor acquired by Kemet in 2017) patent technology that does not require the lens for operation up to 2m and can sense up to 20cm with resin or glass without a hole for a sensor.

Using the lens as an option will increase the sensing distance up to 5 meters

The sensor chip **PL-N823-01** outputs analog voltage that should be amplified and connected to a MCU's A/D for filtering and processing and fits the high volume or price sensitive applications and involves some design activity from the customer side.

The sensor modules mentioned above, have all the necessary circuitry built-in in the module, they are still rather small – about a dime size (Figure 1). The exact dimensions are shown in a module data sheet on

[https://content.kemet.com/datasheets/KEM\\_SE0208\\_SS.pdf](https://content.kemet.com/datasheets/KEM_SE0208_SS.pdf)



Sensor mounting surface    Component mounting surface

**Figure 1**

I decided to test the sensor module **SS-430** (without the lens)

The sensor modules **SS-430** (and **SS-430-N/W/BK** with the lens) have just 3 pins to be used as per below table

Pin No.	Pin Name	Remarks
1	$V_{in}$	3.5 ~ 5.5 VDC
2	TX	NC
3	RX	NC
4	$V_{out}$	Comparator output
5	GND	Ground

The module is shipped with a cable to be connected to a JST connector on a Component mounted side of the module (JST p/n is **SM05B-SRSS-TB(LF)(SN)**)



The basic platform which was used to host the sensor, provide the necessary voltage and evaluate the sensor performance is **Espressif IoT evaluation board ESP32-DEVKITC-32D-F** (DK p/n [1965-1003-ND](#)) powered through the USB cable.



**Vin** and **GND** pins of the sensor module are connected to **5v** and **GND** in the ESP32 board respectively.  
**Vout** pin is connected to **I/O pin 16** in the ESP32 board a will trigger the LED anode, connected to **I/O pin 17** to turn ON/OFF the LED to better illustrate sensor operation.

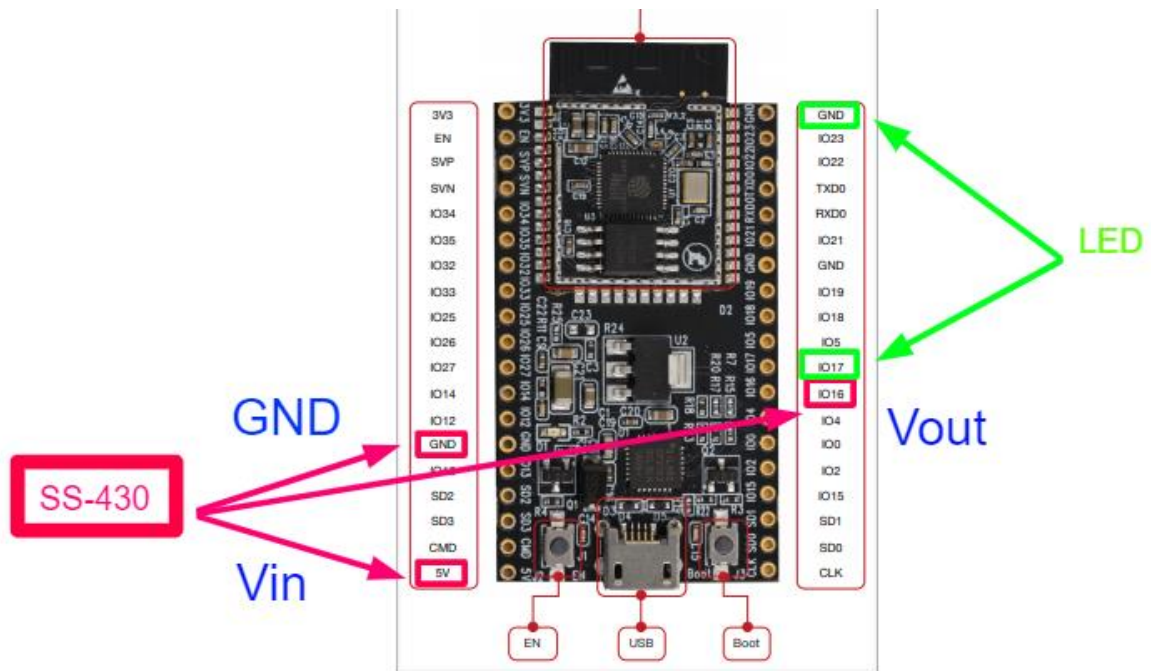


Figure 2

And finally I used **Analog Devices** Active Learning Module - **ADALM2000** (DK p/n [ADALM2000-ND](https://media.digikey.com/pdf/Data%20Sheets/Analog%20Devices%20PDFs/ADALM2000_Web.pdf)) and PC resident software interface **Scopy**  
 This **ADALM2000** is great low cost All-In-One lab tool suitable Students, hobbyists and self-Learners and is good for debugging the designs with low speed signals like in our case.  
[https://media.digikey.com/pdf/Data%20Sheets/Analog%20Devices%20PDFs/ADALM2000\\_Web.pdf](https://media.digikey.com/pdf/Data%20Sheets/Analog%20Devices%20PDFs/ADALM2000_Web.pdf)



First, I had to understand the sensor output behavior

**SS-430** Data Sheet shows that the output ( $V_{out}$ ) of the sensor is **3v, one 200msec width pulse on detection** after the stabilization period of **30 sec**

Part Number	Supply Voltage (VDC)	Output Voltage (VDC)	Consumption Current ( $\mu$ A) Maximum	Detection Output Voltage on Time (secs)	Warm up Time <sup>1</sup> (seconds) Maximum	Field of View ( $^{\circ}$ )	Lens Color	Weight (g)
SS-430	3.5 ~ 5.5	Detection : 3.0 Non-Detection : 0.0	800	0.2	30	Horizontal: $\pm 60^{\circ}$ Vertical : $\pm 60^{\circ}$	-	0.7

This specification is not precise. 2 important questions can be asked:

1. What happens when the object leaves the detection range?
2. What happens on non-movement or quick movements of the heat source inside the detection area?

I had made the following experiments to address these questions

1. The sensor is used without any resin
2. The sensor is used with glass resin

## 1. The sensor is used without any resin

Before that need to take into account the following limitations of PIR technology  
By definition of the Pyroelectric Infrared sensor, the sensor detects a heat source if there is more than 4°C temperature difference from ambient temperature.

However, because of the nature of the PIR sensor, it can detect the heat source regardless whether it is in the detection area, when the sensor is exposed to the following events or conditions:

Indirect and direct sun light, incandescent lamp, strong light beam, air flow from the air conditioner or chiller, frequent ambient temperature changes, strong electromagnetic fields and strong vibrations, etc, etc.

All PIR sensors manufacturers mention some or all of above limitations in the “Cautions for Use” at the end of the Data Sheet.

Based on above “Cautions for Use”, looks like that Kemet SS-430 module is very sensitive to be tested in a office/lab standard environment on the table as the output is sending the pulses all the time even if the object is not in detection area as shown in Figure 3

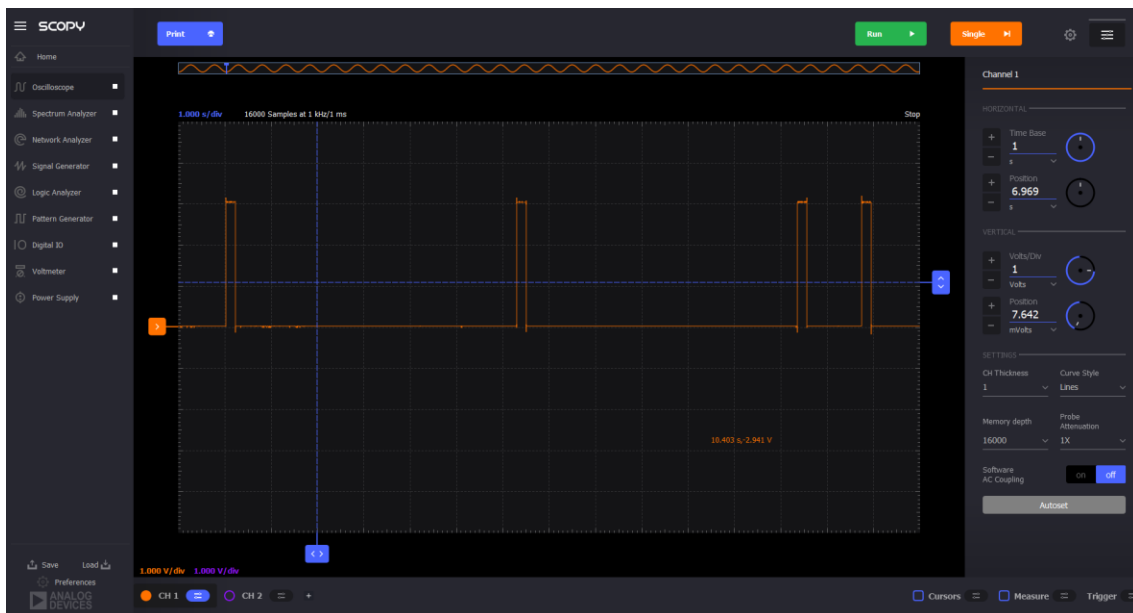
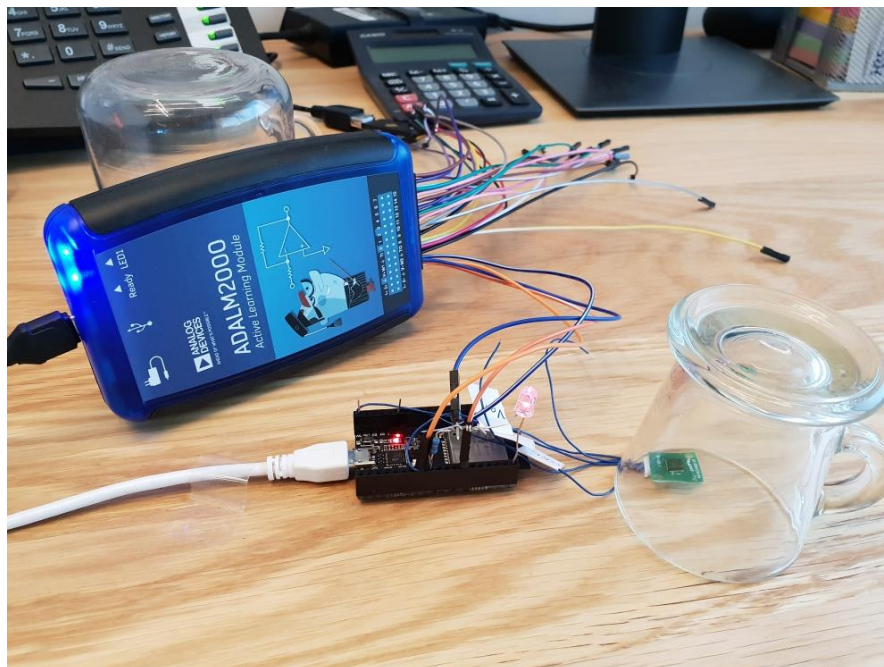
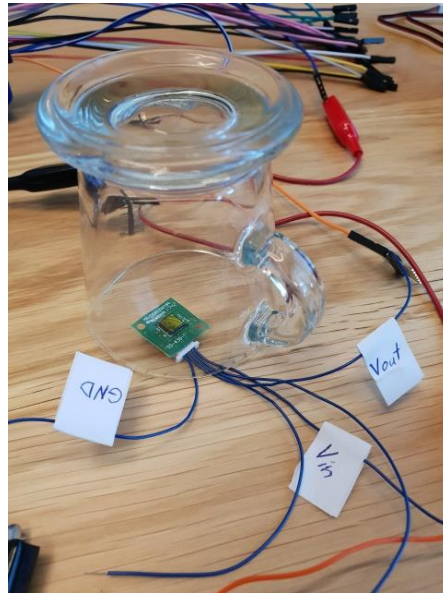


Figure 3

I assume that in order to test the sensor’s functionality for 1 – 2m range without a resin or glass , the sensor should be used in dark environment or properly housed to prevent the external disturbances as listed above

## 2. The sensor is used with a glass resin

In this environment, the sensor is “less sensitive” and can be tested in the standard lab during a day time. The detecting range will be several centimeters but it is enough for testing purposes. So I put a glass cup on the sensor. The bottom of the glass is rather thick, so only actually putting the hand on it causes the sensor to function. In real application using a thinner glass will increase the distance to specified 10 – 20 cm



The test environment was set up as following

1. The sensor is connected to Espressif IoT evaluation board **ESP32-DEVKITC** as described in Figure 2
2. Analog Devices **ADALM2000** is used in oscilloscope mode and the orange channel 1 (out of 2 available) is connected to **Vout** pin of the sensor **SS-430** (via I/O pin 16 in the ESP32 board)
3. The trigger of the scope is set for the rising edge of this channel 1 (Vout) waiting for the sensor to output the pulse
4. The hand is touching the bottom of the glass and is removed with different time periods between touching and removing

After many trials of touching and removing the hand from the bottom of the glass, the following were observed:

1. The output voltage is stable on zero volts if the hand does not touch the glass
2. The sensor outputs the pulse/pulses both on detection of the heat source (the hand) and **also when the heat source (the hand) leaves the detection range**
3. **One or Two pulses signals are outputted** from the sensor on both detection and leaving the detection range, depending on how fast the hand is moving. The distance between the multiple pulses is not constant, but they are close to each other. Figures 4, 5, 6 show the scope screen for this behavior. The first pulse/pulses are on touching the glass, the next series of the pulses are on removing the hand.
4. The sensor outputs One or Two pulses on detection and **does not output any additional pulses as long as the heat source (the hand) is in detection range**
5. The output signals are 200 msec width as specified and are rather clean

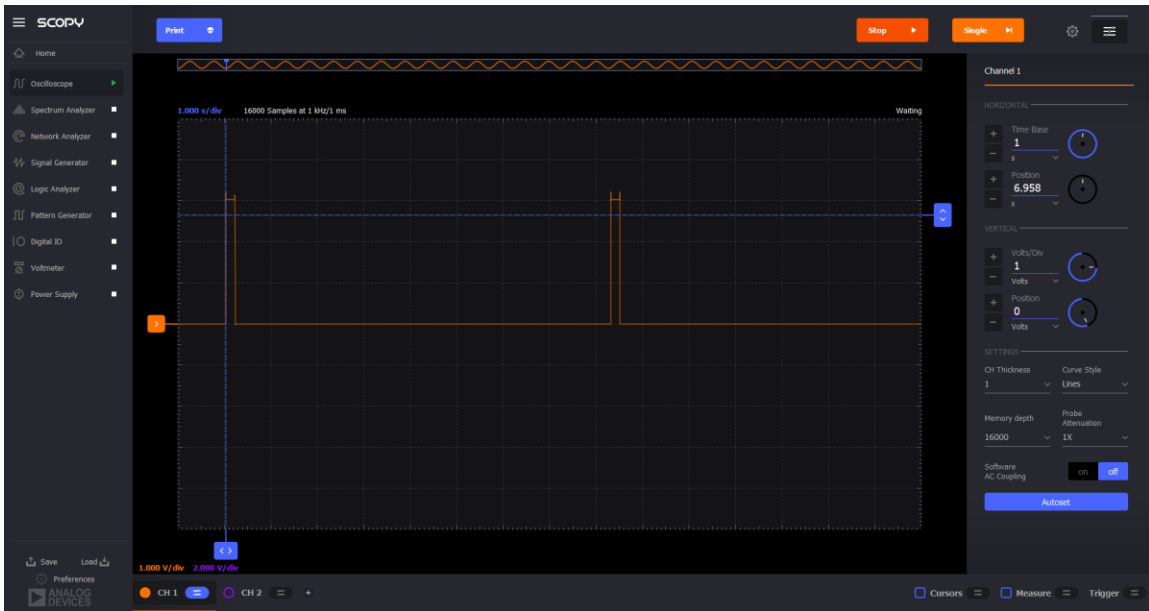


Figure 4

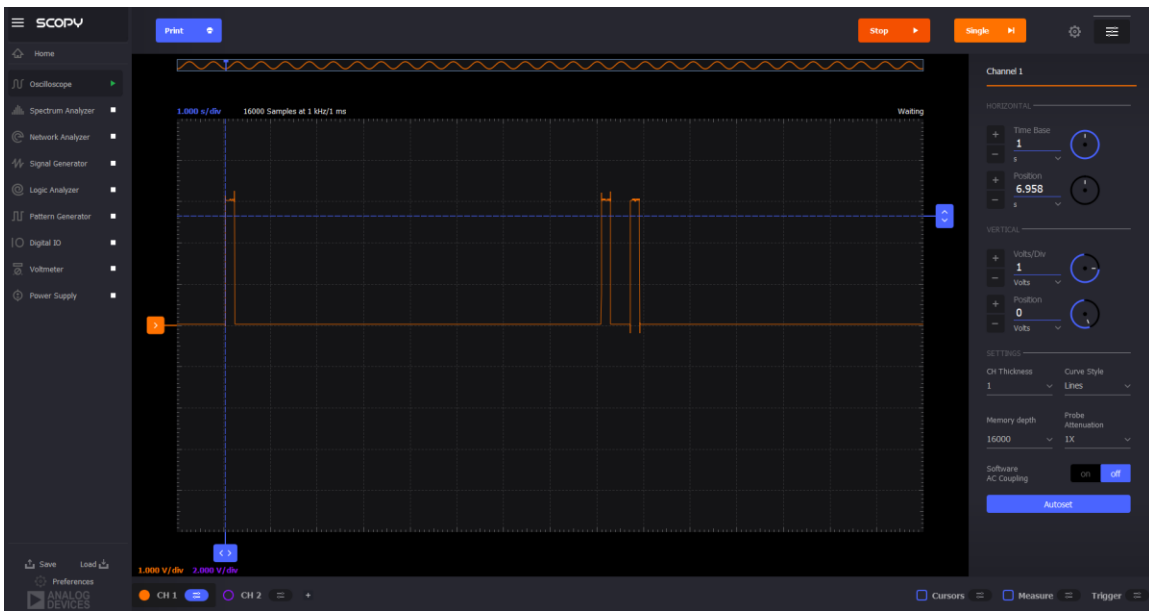


Figure 5

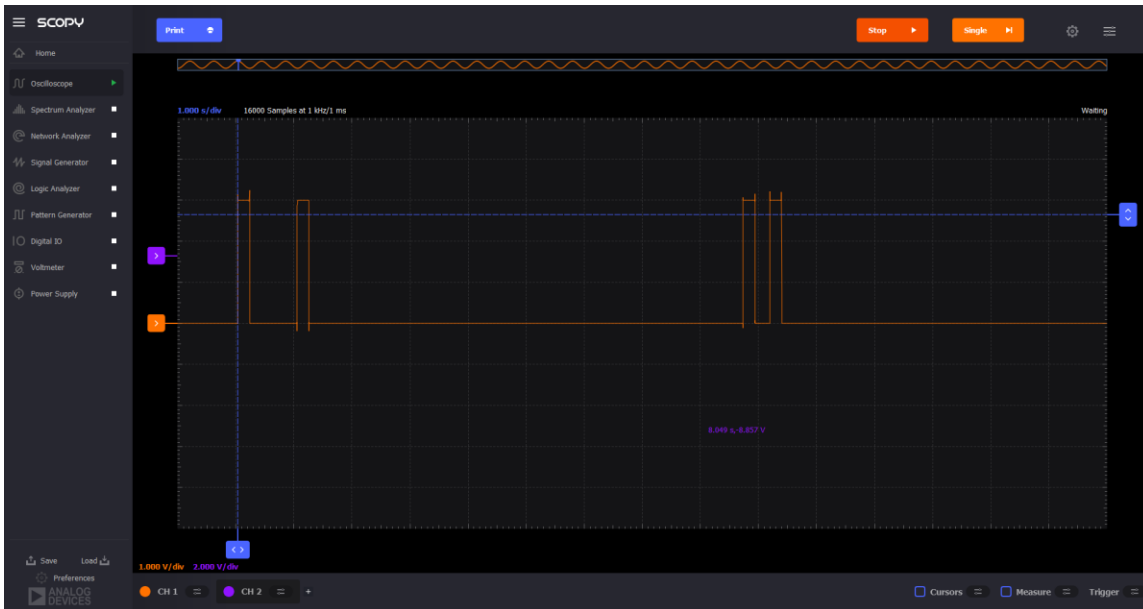


Figure 6

To illustrate the sensor behavior, I added a LED as shown in Figure 2. I decided to use [Digi-Key IoT Studio](#) which is the simplest and fastest way to create one sequential pulse between touching the glass and removing the hand from the glass to turn ON/OFF the LED without writing any code. It is not the most sophisticated way, but serves right the illustration purpose

The falling edge of the sensor output pulse on I/O pin 16, triggers the counter, which counts 2 pulses as modulo 2 counter and on reset toggles the I/O pin 17 – turning the LED ON/OFF as shown on Figure 7.

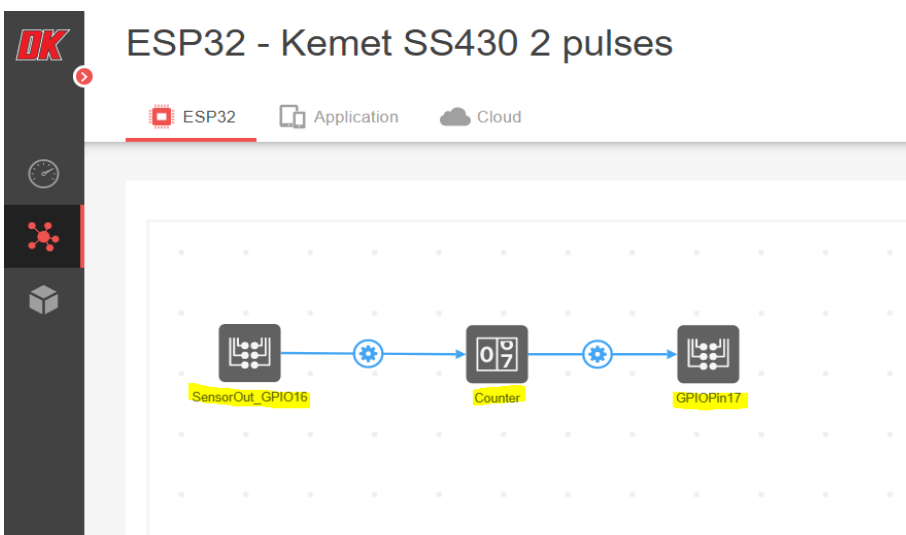


Figure 7

Figure 8 shows the output of the sensor (orange channel) with 2 pulses in touch-remove-touch sequence and purple channel shows the pulse generated by DK IoT studio that is driving the LED

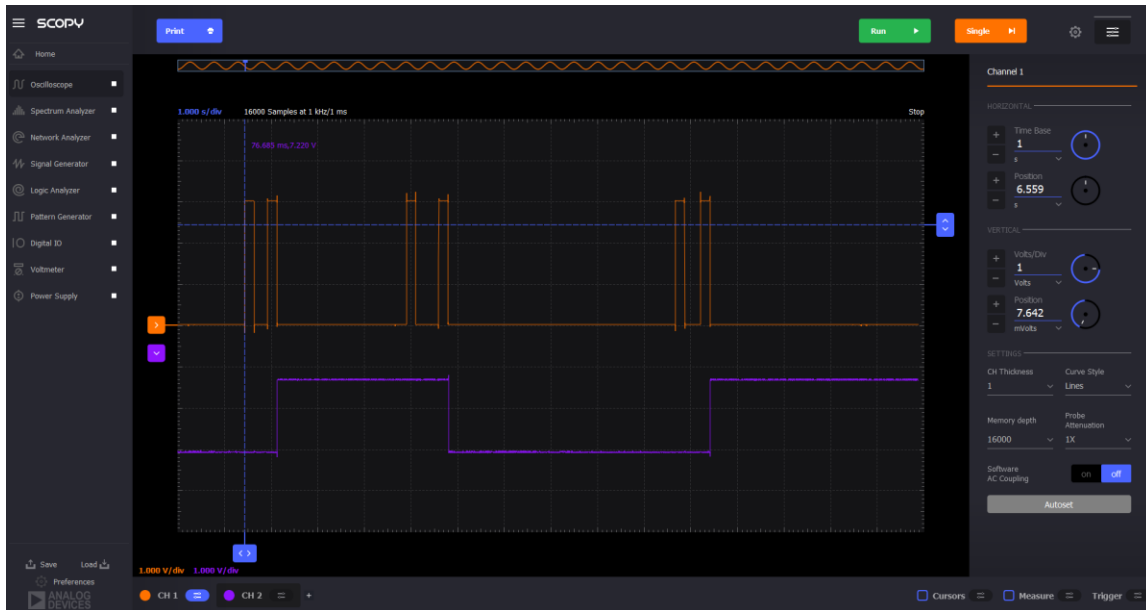


Figure 8

The **video 1** illustrates touching the glass that triggers the scope (the trigger of the scope is set for the rising edge of the channel 1 (Vout) waiting for the sensor to output the pulse) and turning the LED ON in parallel and then removing the hand from the glass is triggering the scope again and is turning the LED OFF

The **video 2** shows the sequence of 3 times touching/removing the hand from the glass that turns ON/OFF the LED. This video demonstrates that there are no additional pulses outputted from the sensor while the heat object is inside the detection range (the hand is on the glass)

To summarize:

1. The sensor SS-430 is very small and low profile module that well fits space critical applications and products where the sensor should be Inconspicuous
2. The module can be operated without the lens
3. To use the sensor without the resin or glass requires mechanical housing design to eliminate the environmental conditions that affect the sensor behavior
4. The sensor can output One or Two pulses both on detecting the heat object and when the heat object passes away from the detection range

## The conclusion

To create a robust design that deals with multiple pulses output and creates the ON/OFF switch to drive another circuitry on detection, the use of a low cost MCU or Programmable Logic is needed to build a pulse-count circuit that counts the pulse/pulses appearance within 2-3 seconds time slot. That delay is tolerable by many applications using PIR proximity sensors

This will make Kemet sensor module SS-430 a real "Plug&Sense" module

Another option is to turn to Kemet and implement this logic in Microchip MCU **PIC12F617T-I/MS** which resides already in the module and is taking care of the A/D sampling, filtering and amplifying.

But I leave this option for the next revision of the module