ELEC 2210 - EXPERIMENT 5
Switch Debounce Circuits

The objectives of this experiment:

The experiments in this laboratory exercise will provide an introduction to the problem of switch debouncing. You will use the ELVIS workstation to build and test circuits. The objectives of this experiment include:

- Review basic principles of latches from ELEC 2200
- Gain an understanding of mechanical switch bounce and its impact in systems
- Use the ELVIS workstation to build and test switch debounce circuits
- Learn to use the oscilloscope to view transient signals
- Continue to develop professional lab skills and written communication skills

I. Introduction

An important (but often overlooked) concern in the design of digital systems is the problem of *switch bounce*. The basic mechanisms involved are the mechanical design of the switch (where actual physical bounce of the contacts can occur) and the large electric fields that develop between switch contacts that are very close to each other (almost making contact with each other but not quite making contact). What results from this situation is arcing between the switch contacts until they finally make permanent contact with each other. This is illustrated in Figure 1. Unless mitigated, this arcing (or *bouncing* as it is more often referred to) can cause undesirable results. For example, one switch press can be counted as several.

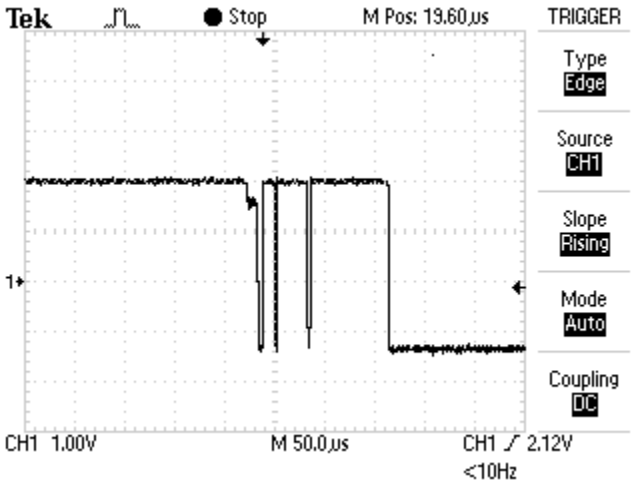


Figure 1. Oscilloscope image of switch bounce during a 1 to 0 transition.
(Source: http://www.labbookpages.co.uk/electronics/debounce.html)

As an example where we see the full effect of the switch bounce problem, consider the circuit illustrated in Figure 2. In this circuit, a simple push button is used to provide the clock input

signal to a binary counter. The counter is shown as a 74161, which is a 4-bit binary counter triggered on the rising edge of its clock signal input. Notice that a 10KΩ resistor is used to "pull-up" the clock input, CLK. This pulls the input to logic 1 when the switch is not activated. This is a good design practice to always pull-up (to a logic 1) or pull-down (to a logic 0) critical unused or unconnected inputs. The unwise alternative is to let the inputs signals "float" and to assume that they will always be at a valid logic level; this is generally a poor assumption.

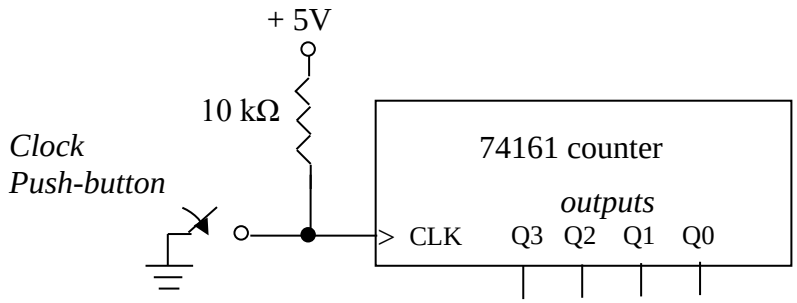


Figure 2. Example circuit with switch-driven clock.

The desired behavior of the circuit in Figure 2 is that each time we push the button, the CLK input is pulled low - nothing should happen at the counter outputs Q0 through Q3. When we release the button, CLK is pulled high and the resulting rising edge of the clock input to the counter should cause the count value at the outputs to increment by 1. In actuality, due to switch bounce we may see multiple increments of the counter, both when we push the Clock button and also when we release the button.

The reason for this undesirable behavior is illustrated in Figure 3 where the expected clock input signal to the counter is shown in Figure 3(a) along with the expected output response of the counter (assuming that we have just reset set the counter to the zero state) in Figure 3(b)

.

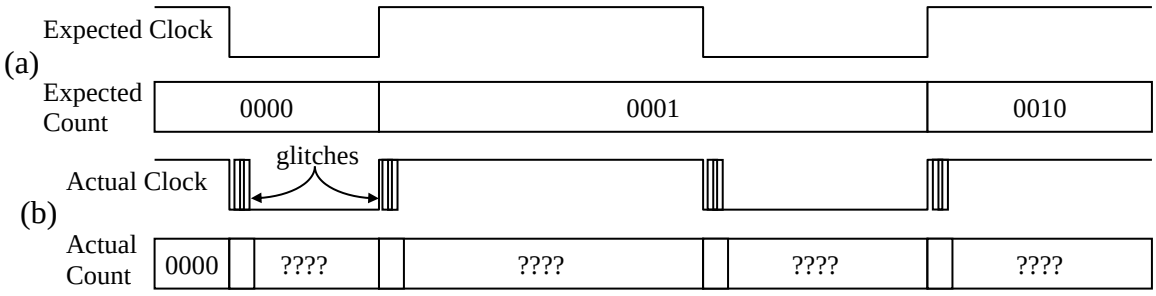
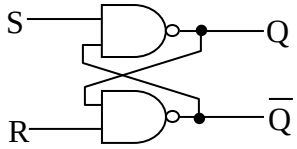


Figure 3. Clock signal and output response

As we push the switch and the contacts are near each other, the arcing begins with each arc pulling the clock signal to ground and the clock returning to a logic 1 (via the pull-up resistor) between arcs. As a result, the counter sees multiple rising edges of the clock input signal as shown in the figure, and responds to these rising edges (or "glitches" in this case) by incrementing by 1 for each glitch. Once contact is made, then the clock signal remains at

logic zero until we begin to release the switch, at which time the arcing begins again and continues until the switch contacts are far enough away from each other to prevent the arcing. Therefore we see multiple rising edges at both the pushing of the Clock button and the releasing of the Clock button and since we have no way of knowing the number of glitches that will occur on any particular push or release of the button, the count value seems to produce random numbers.

The solution to this problem is switch *de-bouncing*. Several alternative methods exist, but in this experiment we will study a very good and simple method which uses an RS latch constructed from cross-coupled NAND gates, as shown in Figure 4 with its associated state table.



S R	Q^+	Function
1 1	Q	storage (hold) state
1 0	0	reset state
0 1	1	set state
0 0	1	indeterminate

Figure 4. Cross-coupled NAND RS latch (R, S active low).

By inserting the RS latch into the Clock push button circuit, as shown in Figure 5, we can overcome the bouncing problem by making use of the Set, Reset, and Storage states of the RS latch. Notice that we have also changed the type of switch from a single-pole, single-throw (SPST) switch to a single-pole, double-throw (SPDT) switch (different switch types are illustrated in figure 6) and have added a pull-up resistor. The two pull-up resistors ensure that the S and R inputs to the RS latch are both at a logic 1 (which puts the RS latch in the storage state) when no contact is made by the switch – this is the case while we are in the process of pushing (or releasing) the switch from one contact to the other.

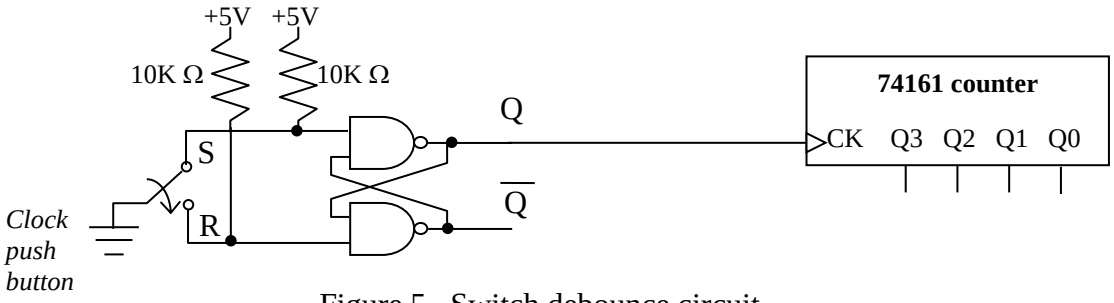


Figure 5. Switch debounce circuit

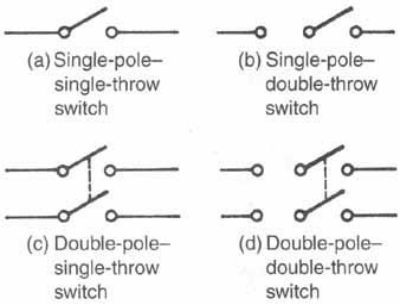


Figure 6. Basic switch types

Comparing the action of the switch to the state table for the RS latch we can see how the switch debouncing circuit works to prevent glitches in our clock signal. Assume that the switch is normally connected to the S (set) input to the RS latch as illustrated in Figure 5. As a result, the S input is pulled to a logic 0 and the Q output is a logic 1 since the RS latch is in the set state. Now, referring to the timing diagram in Figure 7, when we push the button, the break in contact takes the RS latch inputs from S=0 and R=1 to S=1 and R=1 which corresponds to the storage state where the Q output continues to be a logic 1. Glitches at the S input to the RS latch will take the latch between the set state and the storage state (keeping the Q output at a logic 1). As the switch moves towards the contact that connects to the R input to the RS latch we remain in the storage state with Q=1 until we encounter the first arc between the switch and the contact connected to the R input. With this first arc, the inputs to the latch change from S=1 and R=1 to S=1 and R=0 which corresponds to the reset state, producing a logic 0 at the Q output. Subsequent glitches take the RS latch between the reset state and the storage state which keeps Q=0. As a result, we see a clean falling edge of the clock input to the counter as illustrated in Figure 7 and the Expected Clock in Figure 3(a).

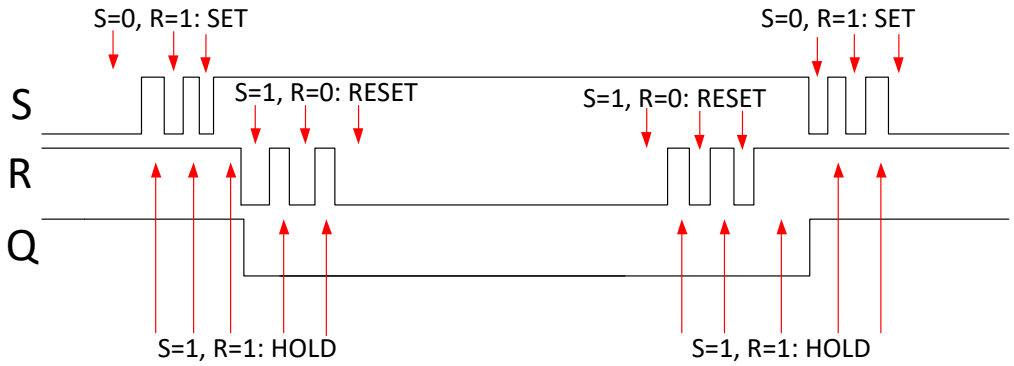


Figure 7. Latch in Figure 5 filters out "bouncing."

When we release the button, we once again go between the reset state and the storage state with the glitches due to the arcing of the switch contacts until the contacts are of sufficient distance that the arcing stops. When the switch nears the contact connected to the S input of the RS latch and arcing begins, the latch will go between the set state and the storage state with the first glitch producing logic 1 at the Q output. This transition in Q produces a rising edge of the clock signal to the counter and our counter is increment one time (and only one time) until the next press and release of the clock push button.

Note that the two pull-up resistors at the inputs to the RS latch also prevent the latch from entering the indeterminate state where both R=0 and S=0, forcing both Q and Qbar outputs of the latch to be logic 1's. Since Qbar should be the opposite logic value of the Q output, this state could cause undesirable behavior in some logic circuits. Since Qbar is unconnected in our example circuit in Figure 5 then there is no problem, but this state should be avoided for good design practice.

II. Pre-Lab

1. In *Multisim*, draw the package-level schematic for the switch debounce circuit experiment shown in Figure 5, with the counter outputs connected to LEDs and to an ELVIS Digital Reader. Simulate the operation of this circuit to verify that changing the switch from the R to the S position increments the counter. (You will not be able to produce "switch bounces" in the simulator, but you can verify that the latch and counter work properly.)

2. On the attached sheet, draw the wiring diagram to be used to implement the circuit on the ELVIS workstation. Note that outputs will be displayed both on LEDs on the ELVIS workstation, and in the NI ELVISmx Digital Reader on your PC.

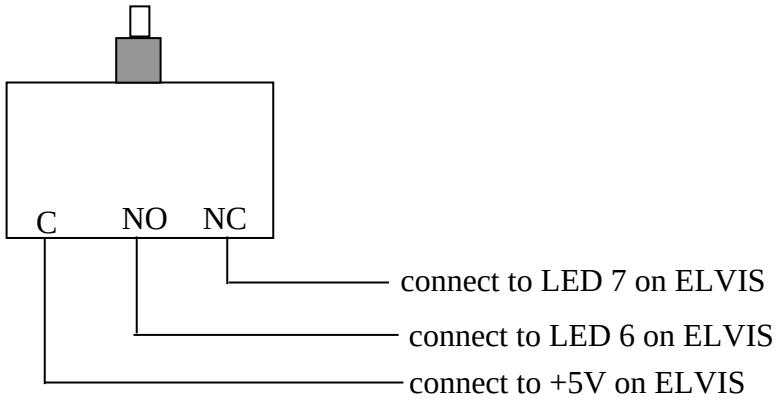
III. Lab Exercise

Obtain the required components. You will use the following:

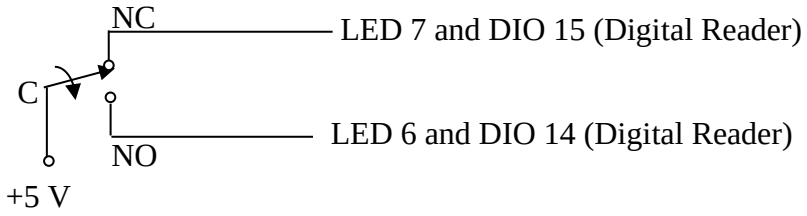
Qty	Part # or value	Description
1	SPDT	Single-pole, double-throw pushbutton switch
1	74161	4-bit binary counter, TTL, 16-pin IC
1	7400	Quad, 2-input NAND gate
2	10K ohm	Resistor

STEP 1.

You will be given a single-pole, double-throw (SPDT) pushbutton switch in the lab. Test your switch to understand how it works, and to verify that it is functioning correctly. Connect as shown below:



The corresponding circuit schematic is shown here:



The lead marked "C" is the center, or "throw", the lead marked "NC" is the "normally closed" position, and the lead marked “NO” is the “normally open” position. In other words, when the button is not pushed, C is connected to NC, and NO is not connected. When the button is pushed, C is connected to NO, and NC is not connected.

If your switch is working properly, LED 7 should be on normally, and LED 6 should be off. When the button is pressed, the LED's should reverse (7 off, 6 on). Note that you should connect the NC output to both LED 7 and DIO 15, and the NO output to both LED 6 and to DIO14, to display the values both on LEDs and in the Digital Reader on your PC.

Have your GTA confirm this by initialing your checklist.

STEP 2.

Connect up the RS latch shown in Figure 5. For testing, also connect the two latch outputs (Q and Qbar) to LEDs 5 and 4, and to DIO 13 and 12. Use the pin assignments and LED connections from your pre-lab wiring diagram. At this step, do not connect the pushbutton switch to the RS latch inputs or connect the latch’s output to the adder. Instead, connect DIO 1 and 0 on the ELVIS to the S and R latch inputs, so that they can be controlled with a Digital Writer. Use the Digital Writer to change the RS latch inputs, and LEDs on the latch outputs, confirm that the latch is working as expected, according to the truth table given in Figure 4.

Have the GTA initial this on your checklist.

STEP 3.

Disconnect the wires from DIO 1 and 0. Also, disconnect the wire from the C lead of the pushbutton switch, connect the C lead to GROUND (0 V), and then connect your SPDT pushbutton switch to the RS latch as shown in Figure 5, including the pull-up resistors. At this step, do not connect the latch output Q to the counter. Connect the NC position to S and the NO position to R. Connect and set up the oscilloscope as indicated below. Then, as illustrated in Figure 9, observe the raw switch signal at the R latch input, and compare with the debounced output at the Q output of the latch. Press the button 10 times to find the case with the worst bounce. You should observe the signal both when the button is pressed and when it is released. When you find an image with significant bouncing, capture the oscilloscope window and paste it into your report.

Oscilloscope Connection and Setup:

On the left side of the ELVIS base, make sure the cables connecting CH 0 (Channel 0) to BNC 1, and CH 1 (Channel 1) to BNC 2 are in place. Then connect the oscilloscope to your circuit as follows:

1. Use a wire to connect BNC 1 + (hole 42 on the terminal strip on the left side of the project board) to the R input of the latch, and another to connect BNC 1 - (terminal 43) to GROUND. This connects the R input to oscilloscope channel 0.
2. Use a wire to connect BNC 2 + (44 on the terminal strip) to the Q output of the latch, and another to connect BNC 2 - (terminal 45) to GROUND. This connects the Q output to oscilloscope channel 1.

(Note that an oscilloscope probe measures the potential difference between two points. In this case, we are measuring the voltage on the R latch input, with respect to ground, and on the Q output, with respect to ground. Hence, each probe is connected to both the signal to be captured and ground.)

Open the oscilloscope (Scope) instrument from the NI ELVISmx Instrument Launcher and make the following settings, as illustrated in Figure 9.

1. Channel 0 Source: SCOPE CH 0.
2. Channel 1 Source: SCOPE CH 1.
3. Both channels: "Probe" = 1x, "Coupling" = DC, and "Scale" = 2 V per division.
4. Channel 0 "Vertical Position" = 0; Channel 1 "Vertical Position" = -4
   (This separates the two images on the screen.)
5. Timebase = 50 µs (try different values during the experiment to see finer detail or to see more events.)
6. Trigger: "Type" = Edge; "Slope" = Rising; "Level" = 1 V; "Source" = Chan 0 Source; "Horizontal Position" = 20%,
7. Acquisition Mode: Continuous

Have the GTA initial this step on your checklist.

STEP 4.

Wire up the 74161 as in your pre-lab wiring diagram, connecting the latch output **Q** to the 74161 clock input, and the counter outputs QD-QA to LEDs 3-0 and to DIO 11-8. Push the button a number of times, while observing the oscilloscope display and the counter output. When / if your switch bounces, note what happens to the counter output (it should skip one or more counts). Estimate the average number of bounce events for the de-bounced switch (how many times did the counter increment for a given button press).

Repeat using the raw (non-debounced) pushbutton output connected directly to the counter. Bypass the de-bounce latch circuit by (a) disconnecting latch output Q from the counter input, and then (b) connecting a wire directly from node R to the counter.

Demonstrate to your GTA, and have him/her initial this step on your checklist.

STEP 5. Cleanup

DO NOT PUT RESISTORS, CAPACITORS, CHIPS, OR ANY OTHER COMPONENTS IN THE WIRE TUBS.

(a) Turn off the power to the ELVIS base and board.
(b) Disassemble your circuit and place all wires back in the wire tub.
(c) Put all chips and other components back in the proper bins.
(d) Clean up your workstation and discard any trash.

Have your GTA inspect your workstation and check off on your checklist if acceptable.

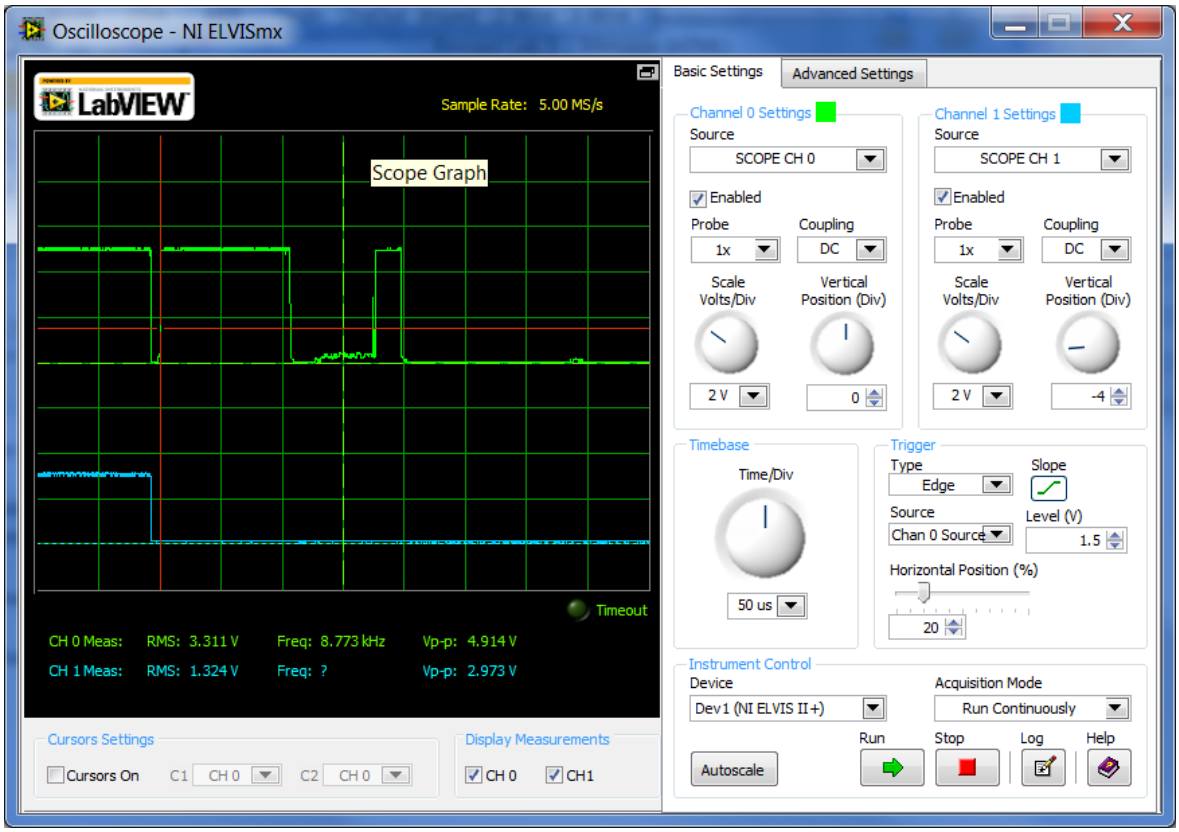
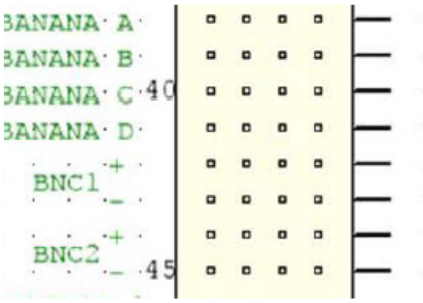
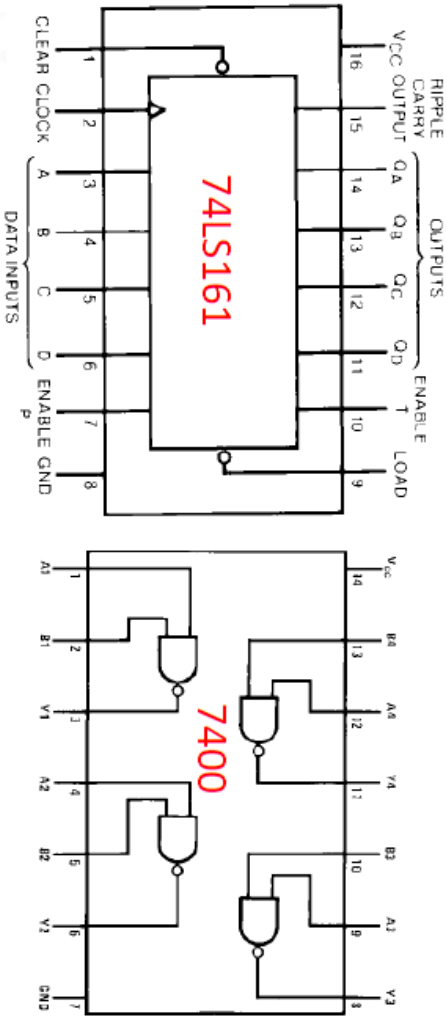
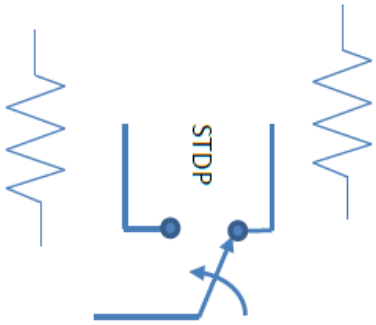


Figure 9. ELVIS oscilloscope – bouncing signal in green, debounced signal in blue.







Digital Writer

- DIO 0
- DIO 1
- DIO 2
- DIO 3
- DIO 4
- DIO 5
- DIO 6
- DIO 7

Digital Reader

- DIO 8
- DIO 9
- DIO 10
- DIO 11
- DIO 12
- DIO 13
- DIO 14
- DIO 15

- LED 0
- LED 1
- LED 2
- LED 3
- LED 4
- LED 5
- LED 6
- LED 7

- +5V
- GROUND