Lab 3. Switch and LED Interfacing

Preparation

1. Read Sections 2.7, 3.3.8, and 4.2
2. Download the following:

EE319K_TM4C123_artist.sch	PCB artist example drawing with all EE319K components
Lab3_EE319K_asm	Starter project with links to grading engine
PCBArtist	http://www.4pcb.com/ Schematic Software

Purpose

Lab 3 is the continuation of Lab 2; essentially, you will implement in hardware what you developed in simulation back in Lab 2. The purpose of this lab is to learn how to interface a switch and an LED. You will perform explicit measurements on the circuits in order to verify they are operational and to improve your understanding of how they work.

You will learn to use a free design tool called PCBArtist from Advanced Circuits for drawing electrical circuits. This tool will be used in EE319K, EE445L and EE445M.

System Requirements

Figure 4.13 in the book shows a switch interface that implements positive logic. In this lab you will connect it to PE0. The right side of Figure 4.14 shows an LED interface that implements positive logic. You will attach this switch and LED to your protoboard (the white piece with all the holes), and interface them to your TM4C123. Overall functionality of this system is the similar to Lab 2, with six changes:

1. The pin to which we connect the switch is moved to PE1
2. You will have to remove the PUR initialization because the pullup resistor is no longer needed
3. The pin to which we connect the LED is moved to PE0
4. The switch is changed from negative to positive logic
5. You should increase the delay so that the LED flashes (on then off) at about 8 Hz
6. The LED should be on when the switch is not pressed

Therefore, your embedded system will:

1. Make **PE0** an output and make **PE1** an input.

2. The start with the LED on (make **PE0** =1).
3. Wait about 62 ms (1000ms / 16 toggles per second)
4. If the switch is pressed (**PE1** is 1), then toggle the LED once, else turn the LED on.
5. Steps 3 and 4 are repeated over and over

Procedure

Back in Lab 2, you developed and debugged your system using the simulator, and then used a built-in switch and LED to run on the real board. During this lab you will build and test your hardware. The final demonstration will be run stand-alone without connection to the PC. This stand-alone operation illustrates how embedded systems operate.

To run the Lab 3 simulator, you must do two things. First, execute Project->Options and select the Debug tab. The debug parameter field must include **-dEE319KLab3**. Second, the **EE319KLab3.dll** file must be added to your Keil\ARM\BIN folder. The **Grading Controls** are for the MOOC and are not used as actual grades in EE319K this semester.

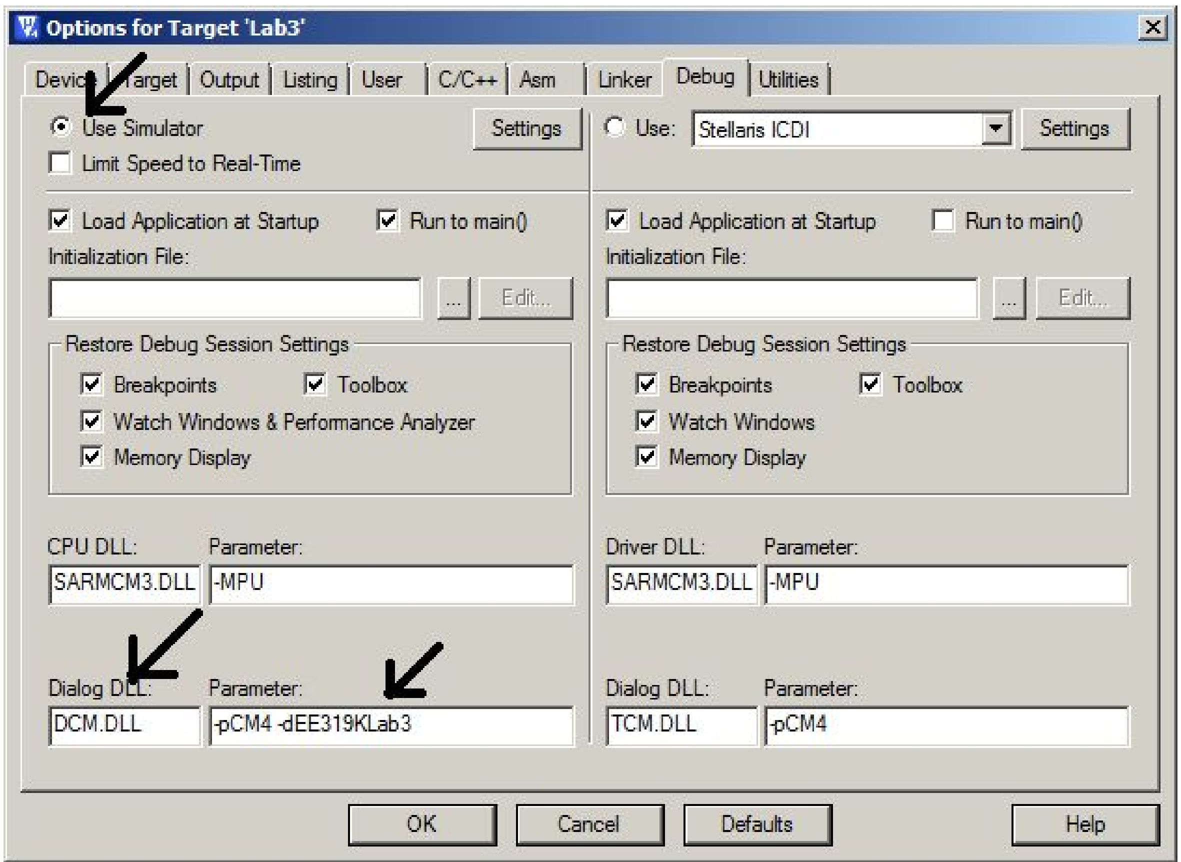


Figure 3.1a. Using TExaS to debug your software in simulation mode (DCM.DLL -pCM4 -dEE319KLab3).

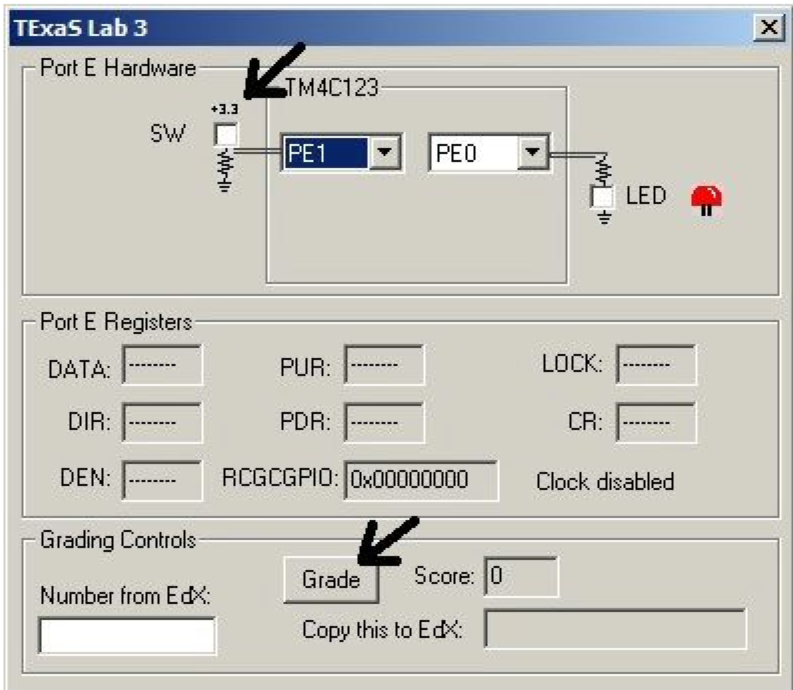
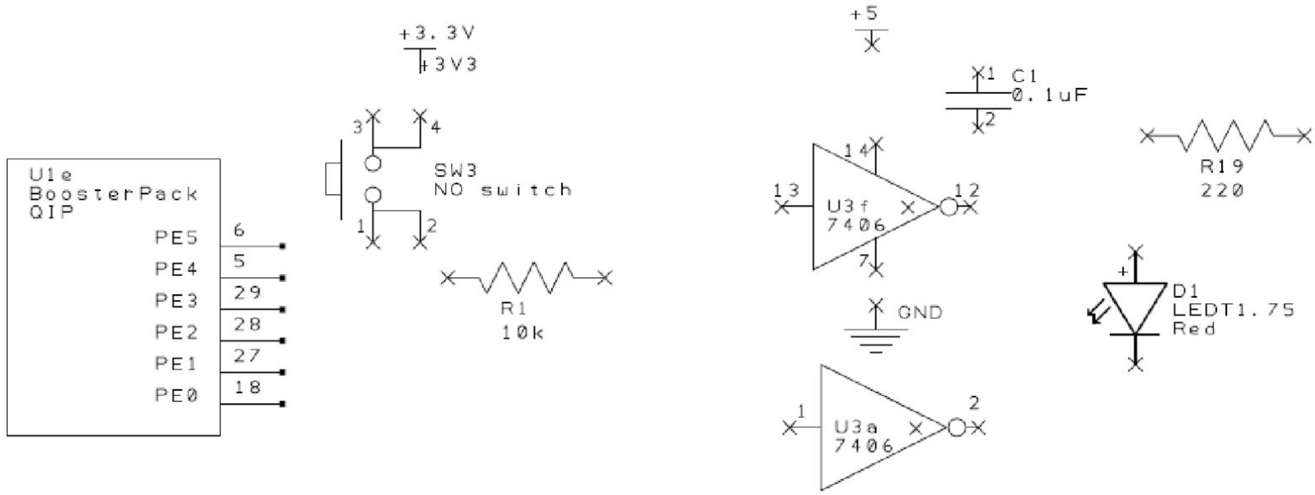


Figure 3.1b. Click in the white box to affect the state of the switch .

Part a - Draw Schematic

Download PCB Artist. Go to http://www.4pcb.com/. When downloading this free application from the internet, be careful NOT to accept any of the special offers (when asked yes or no, say NO.) When installed, there will be a drawing application called PCB Artist, which we will use to draw electrical circuits. Open the starter file **EE319K_TM4C123_artist.sch** using PCB Artist. We will also use PCB Artist to design PCB boards in EE445L.

Make a copy of the EE319K_TM4C123_artist.sch file and rename it with your EID and Lab3. Edit this file to show the interface for the switch and the LED. When you are done, print the circuit as a pdf file, and email this pdf file to your TA for approval. If you wish to build something different from Figure 3.3, please have a TA check your design. Your circuit should look similar to book Figure 4.12 and Figure 4.13. The first step is to click and drag components you will need so they are close to each other



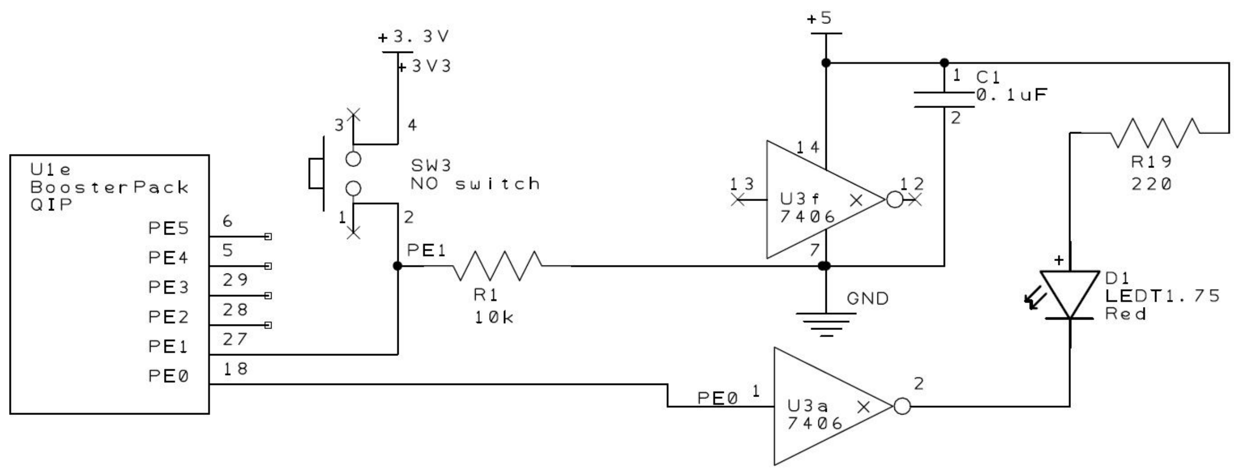


Figure 3.2. PCB Artist drawing showing Port E, 3.3V power, a 10kΩ resistor, a switch, one 7406 gate, an LED, a 220Ω resistor, a 0.1 μF capacitor, +5V power and ground.

The second step is to add traces (these will be actual wires when you build it). In general, we draw circuit diagrams so current flows from top to bottom on the page.

Figure 3.3. PCB Artist drawing one possible Lab 3 drawing. When building the circuit, the +3.3V comes from the +3.3V signal on the LaunchPad. The +5V comes from the VBUS signal on the LaunchPad. The LaunchPad ground must be connected to the ground signals on your external circuit.

In the left side of Figure 3.4, wire1 and wire2 cross in the circuit diagram, but are not electrically connected, because there is NO dot at the point of crossing. In the right side of Figure 3.4, wire1 and wire2 cross in the circuit diagram, and are electrically connected, because there IS a dot at the point of crossing.

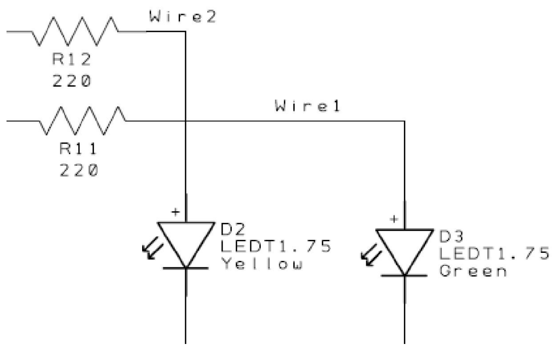


Figure 3.4. When traces cross without a dot they are not connected. When traces cross with a dot they are connected.

Part b - Update Software

Make the six changes listed in the requirements section to your Lab 2 system. The best approach to time delay will be to use a hardware timer, which we will learn to use in Lab 4. In this lab we do not expect you to use the hardware timer. Again, use the logic analyzer on the simulator to verify the software operates properly. In this lab the delay was increased so you will be able to see the LED flash with your eyes.

Whenever we call **TExaS_Init**, we activate the phase lock loop (PLL) and run at 80 MHz. %%%

Part c - Read Data Sheets

Engineers must be able to read datasheets during the design, implementation and debugging phases of their projects. During the design phase, datasheets allow us to evaluate alternatives, selecting devices that balance cost, package size, power, and performance. For example, we could have used other IC chips like the 7405, 74LS05, or 74HC05 to interface the LED to the TM4C123. In particular, we chose the 7406 or 74LS06 because it has a large output current (I_{OL} = 40 mA), 6 drivers, and is very inexpensive (59¢). During the implementation phase, the datasheet helps us identify which pins are which. During the debugging phase, the datasheet specifies input/output parameters that we can test. Download the 7406 and LED datasheets, 7406.pdf and LED_red.pdf, and find the two pictures as shown in Figure 3.5. Next, hold your actual 7406 chip and identify the location of pin 1. Find in the datasheet the specification that says the output low voltage (V_{OL}) will be 0.4V when the output low current (I_{OL}) is 16 mA (this is close to the operating point we will be using for the LED interface).

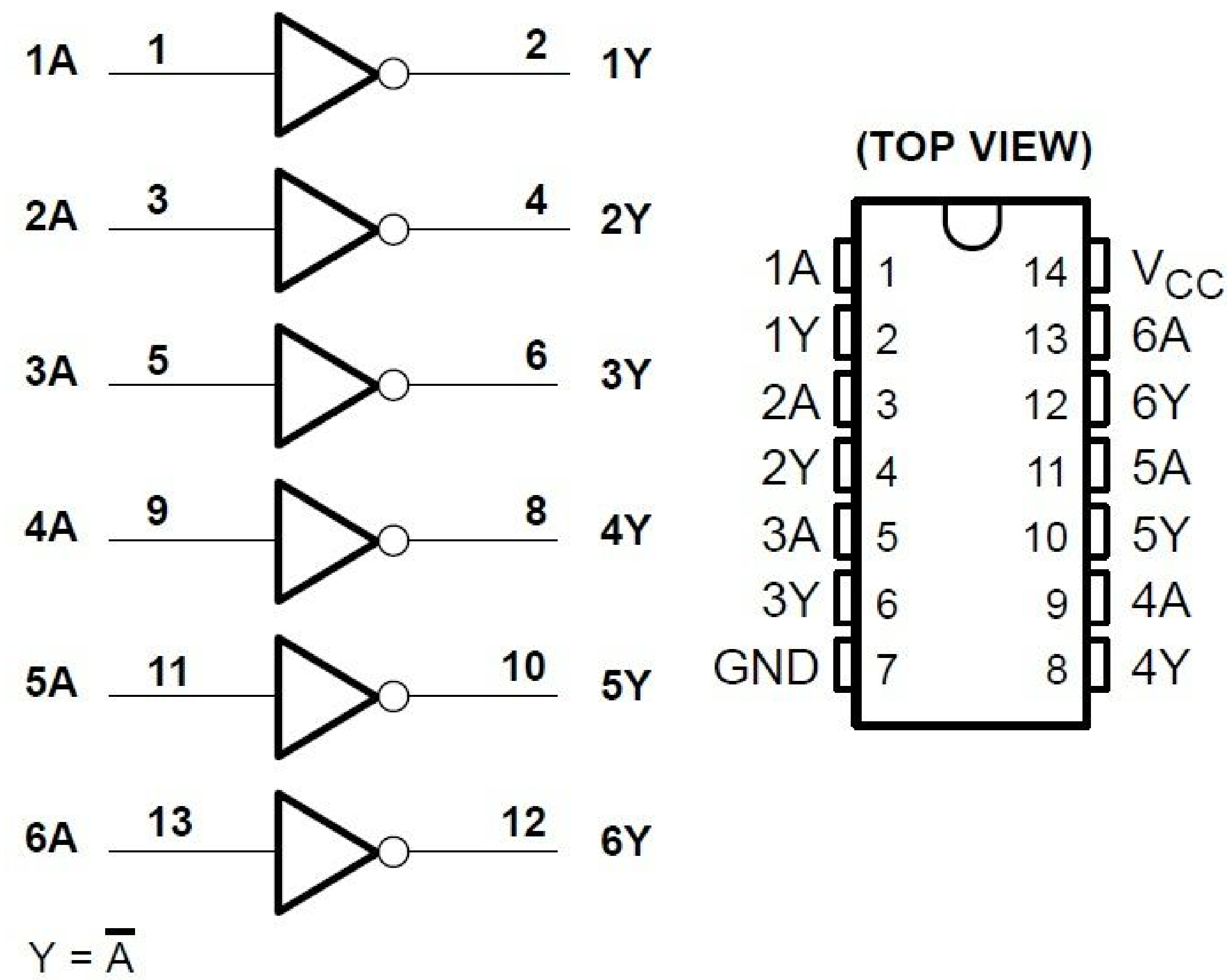


Figure 3.5. Connection diagram and physical package diagram for the 7406.

Using the data sheet, hold an LED and identify which pin is the anode and which is the cathode.

Sometimes we are asked to interface a device without a data sheet. Notice the switch has 4 pins in a rectangular shape, as shown in Figure 3.6. Each button is a single-pole single-throw normally-open switch. All four pins are connected to the switch. Using your ohmmeter determine which pairs of pins are internally connected (having a very small resistance), and across which pair of pins is the switch itself. In particular, draw the internal connections of the switch, started in Figure 3.6, showing how the four pins are connected to the switch.

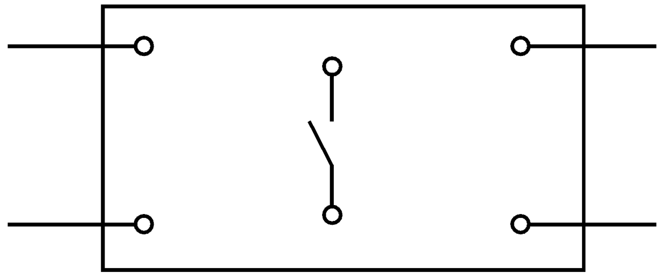


Figure 3.6. Connection diagram for the normally open switch.

To build circuits, we'll use a solderless breadboard, also referred to as a protoboard. The holes in the protoboard are internally connected in a systematic manner, as shown in Figure 3.7. The long rows of of 50 holes along the outer sides of the protoboard are electrically connected. Some protoboards like the one in Figure 3.7 have four long rows (two on each side), while others have just two long rows (one on each side). We refer to the long rows as power buses. If your protoboard has only two long rows (one on each side), we will connect one row to +3.3V and another row to ground. If your protoboard has two long rows on each side, then two rows will be ground, one row will be +3.3V and the last row will be +5V (from VBUS). Use a black marker and label the voltage on each row. In the middle of the protoboard, you'll find two groups of holes placed in a 0.1 inch grid. Each adjacent row of five pins is electrically connected. We usually insert components into these holes. IC chips are placed on the protoboard, such that the two rows of pins straddle the center valley.

To make connections to the TM4C123 we can run male-male solid wire from the bottom of the microcontroller board to the protoboard. For example, assume we wish to connect TM4C123 PE1 output to the 7406 input pin 1. First, cut a 24 gauge solid wire long enough to reach from PE1 and pin 1 of the 7406. Next, strip about 0.25 inch off each end. Place one end of the wire in the hole for the PE1 and the other end in one of the four remaining of the 7406 pin 1.

If you are unsure about the wiring, please show it to your TA before plugging in the USB cable. I like to place my voltmeter on the +3.3V power when I first power up a new circuit. If the +3.3V line doesn't immediately jump to +3.3V, I quickly disconnect the USB cable. Alternatively, you can monitor the +3.3V line with the green led in the upper portion of the launchpad near the usb connectors. If this led does not turn on when you think you are applying power, you might have a short circuit.

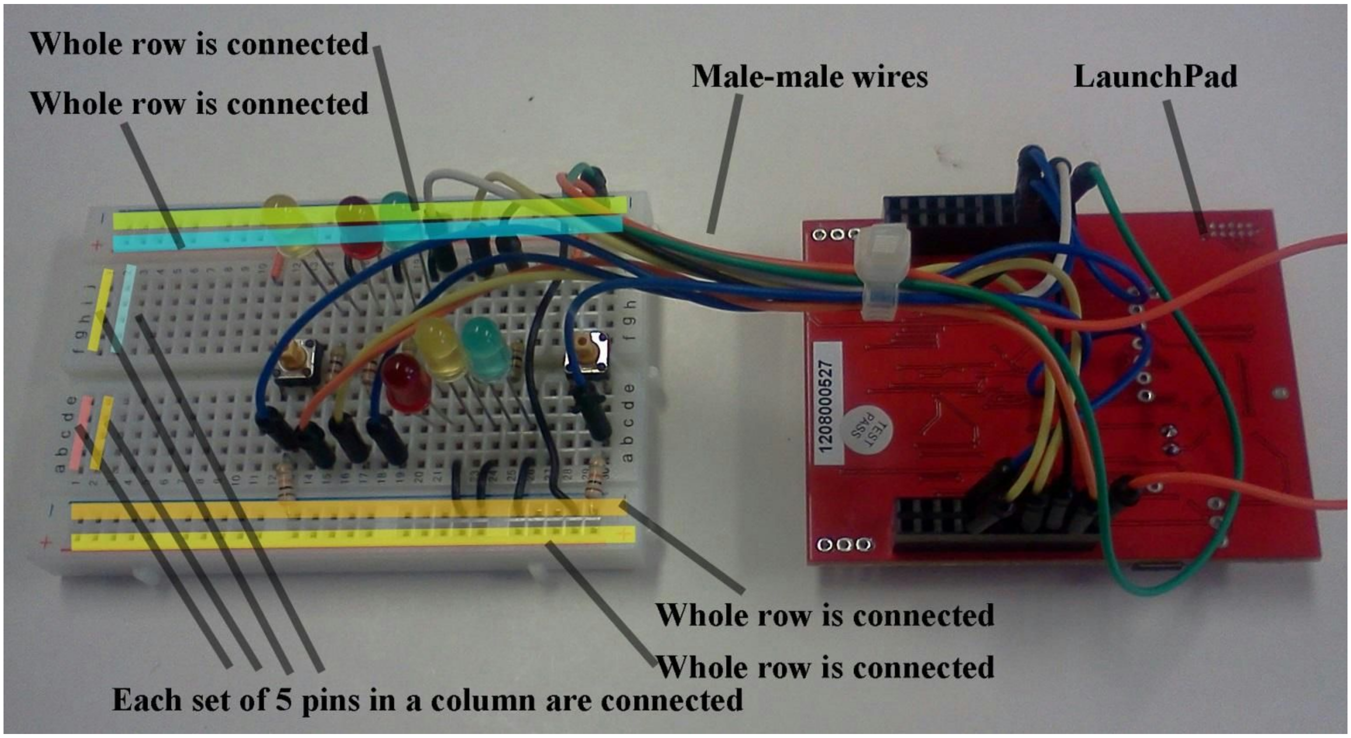


Figure 3.7. All the pins on each of the four long rows are connected. The 5 pins in each short column are connected. Use male-male wires to connect signals on the LaunchPad to devices on the protoboard. Make sure ground wire connected between the LaunchPad and your circuit. The +3.3V and VBUS (+5V) power can be wired from the LaunchPad to your circuit. I like to connect the two dark blue rows to ground, one red row to +3.3V and the other red row to VBUS(+5V).

Part d - Construct Circuit

After the software has been debugged on the simulator, you will build the hardware on the real board. Please get your design checked by the TA before you apply power (plug in the USB cable). *Do not place or remove wires on the protoboard while the power is on.*

Before connecting the switch to the microcontroller, please take the measurements in Table 3.1 using your digital multimeter. The input voltage (V_{PE0}) is the signal that will eventually be connected to **PE1**. With a positive logic switch interface, the resistor current will be V_{PE10}/10kΩ. The voltages should be near +3.3 V or near 0 V and the currents will be less than 1 mA. The goal is to verify the V_{PE1} voltage is low when the switch is not pressed and high when the switch is pressed.

The circuit in Figure 3.2 used R1 as the 10kΩ resistor. There are other 10k resistors in the PCB artist starter file; any one of which could have been used.

Parameter	Value	Units	Conditions
Resistance of the 10kΩ resistor, R1		ohms	with power off and disconnected from circuit (measured with ohmmeter)
Supply Voltage, $V_{+3.3}$		volts	Powered (measured with voltmeter)
Input Voltage, V_{PE1}		volts	Powered, but with switch not pressed (measured with voltmeter)
Resistor current		mA	Powered, but switch not pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)
Input Voltage, V_{PE1}		volts	Powered and with switch pressed (measured with voltmeter)
Resistor current		mA	Powered and switch pressed $I=V_{PE1}/R1$ (calculated and measured with an ammeter)

Table 3.1. Switch measurements.

Next, you can connect the input voltage to **PE1** and use the debugger to observe the input pin to verify the proper operation of the switch interface. You will have to single step through the code that initializes Port E, and PE1. You then execute the **Peripherals->TEXaS Port E** command. As you single step you should see the actual input as controlled by the switch you have interfaced, see Figure 3.1.

The next step is to build the LED output circuit. LEDs emit light when an electric current passes through them, as shown in Figure 3.8. LEDs have polarity, meaning current must pass from anode to cathode to activate. The anode is labeled **a** or **+** , and cathode is labeled **k** or **-**. The cathode is the short lead and there may be a slight flat spot on the body of round LEDs. Thus, the anode is the longer lead. LEDs are not usually damaged by heat when soldering. Furthermore, LEDs will not be damaged if you plug it in backwards. However, LEDs won't work plugged in backwards. Look up the pin assignments in the 7406 data sheet. Be sure to connect +5V power to pin 14 and ground to pin 7. The 0.1 µF capacitor from +5V to ground filters the power line. Every digital chip (e.g., 7406) should have a filter capacitor from its power line (i.e., pin 14 V_{CC}) to ground. The capacitor in your kit is ceramic, which is not polarized, meaning it can be connected in either direction.

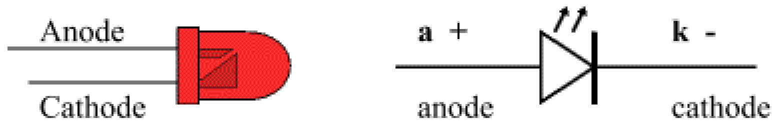


Figure 3.8. Left: a side view of an LED with leads labeled; Right: the corresponding circuit diagram

The circuit in Figures 3.2 and 3.3 used R10 as the 220Ω resistor. There are six 220Ω resistors R10 – R15 in the PCB artist starter file, any one of which could have been used.

Take the measurements as described in Table 3.2. The R10 measurement occurs before R10 is inserted into the circuit. Single step your software to make **PE0** to output. Initially **PE0** will be low. So take four measurements with **PE0** low, rows 2,3,4,5 in Table 3.2. Then, single step some more until **PE0** is high and measure the three voltages (rows 8,9,10 in Table 3.2). When active, the LED voltage should be about 2 V, and the LED current should be about 10 mA. The remaining rows are calculated values, based on these 8 measurements. The LED current (row 12) can be determined by calculation or by direct measurement using the ammeter function. You should perform both ways to get LED current.

Warning: NEVER INSERT/REMOVE WIRES/CHIPS WHEN THE POWER IS ON.

Row	Parameter	Value	Units	Conditions
1	Resistance of the 220Ω resistor, R10		ohms	with power off and disconnected from circuit (measured with ohmmeter)
2	+5 V power supply V_{+5}		volts	(measured with voltmeter, *notice that the +5V power is not exactly +5 volts*)
3	TM4C123 Output, V_{PE1} input to 7406		volts	with **PE0** = 0 (measured with voltmeter)
4	7406 Output, V_{k-} LED k-		volts	with **PE0** = 0 (measured with voltmeter)
5	LED a+, V_{a+} Bottom side of R10		volts	with **PE0** = 0 (measured with voltmeter)
6	LED voltage		volts	calculated as $V_{a+} - V_{k-}$
				calculated as $(V_{+5} - V_{a+})$/R10

7	LED current		mA	and measured with an ammeter
8	TM4C123 Output, V_{PE1} input to 7406		volts	with **PE0** = 1 (measured with voltmeter)
9	7406 Output, V_{k-} LED k-		volts	with **PE0** = 1 (measured with voltmeter)
10	LED a+, V_{a+} Bottom side of R10		volts	with **PE0** = 1 (measured with voltmeter)
11	LED voltage		volts	calculated as $V_{a+} - V_{k-}$
12	LED current		mA	calculated as $(V_{+5} - V_{a+})/\mathrm{R10}$ and measured with an ammeter

Table 3.2. LED measurements (assuming the 220 Ω resistor is labeled R10).

Part e - Debug Hardware + Software

Debug your combined hardware and software system.

Demonstration

(both partners must be present, and demonstration grades for partners may be different)

You will show the TA your program operation on the actual TM4C123 board. The TA may look at your data and expect you to understand how the data was collected and how the switch and LEDs work. Also be prepared to explain how your software works and to discuss other ways the problem could have been solved. Why the 7406 was used to interface the LED? I.e., why did we not connect the LED directly to the TM4C123. Why do you think you need the capacitor for 7406 chip? Why was the delay increased from 1 to 62 ms? How would you modify the software to change the rate at which LED flickers? What operating point (voltage, current) exists when the LED is on? Sketch the approximate current versus voltage curve of the LED. Explain how you use the resistor value to select the operating point. What is the difference between a positive logic and negative logic interface for the switch or the LED? We may test to see if you can measure voltage, current and/or resistance with your meter (so bring your meter to the demonstration).

Deliverables

(Items 2, 3, 4, 5, and 6 are one pdf file uploaded to Canvas, have this file open during demo.)

1. Lab 3 grading sheet. You can print it yourself or pick up a copy in lab. You fill out the information at the top.
2. Circuit diagram, using PCBArtist, or hand drawn
3. Screenshot like Figure 3.9 showing your debugging in the simulator
4. Switch measurements (Table 3.1)
5. LED measurements (Table 3.2)
6. Assembly source code of your final program
7. Optional Feedback : http://goo.gl/forms/rBsP9NTxSy

Precautions to avoid damaging your system

1. Do not attach or remove wires on the protoboard when power is on. Always shut power off when making hardware changes to the system.
2. Touch a grounded object before handling CMOS electronics. Try not to touch any exposed wires.
3. Do not plug or unplug the modules into the LaunchPad while the system is powered.
4. Do not use the TM4C123 with any external power sources, other than the USB cable. In particular, avoid connecting signals to the TM4C123 that are not within the 0 to +5V range. Voltages less than 0V or greater than +5V will damage the microcontroller.
5. Do not use PC3,PC2,PC1,PC0. These are needed for the debugger.
6. You can't use PA1 PA0 PD4 and PD5. These are connected to the serial port and USB.
7. See Figure 4.8 in the textbook. If you use both PD0 and PB6, remove R9 from your board. If you use both PD1 and PB7, remove R10 from your board.

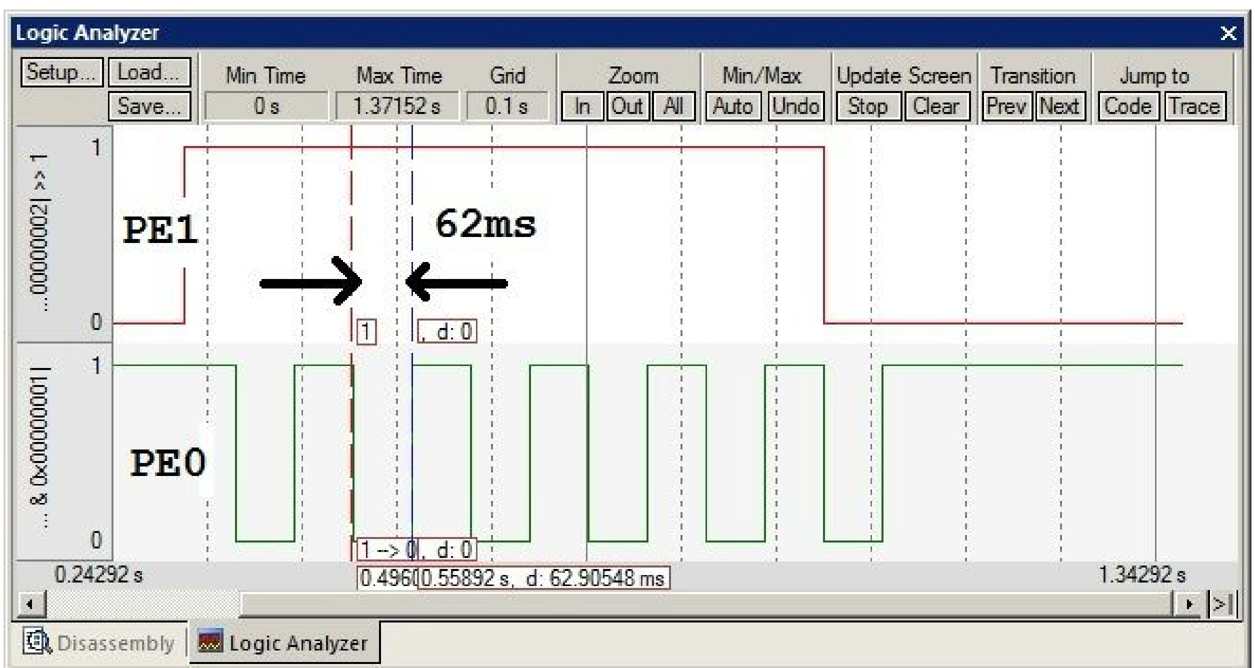


Figure 3.9. Simulation of Lab 3, showing PE1 high, and the PE0 output toggling at 8 Hz.

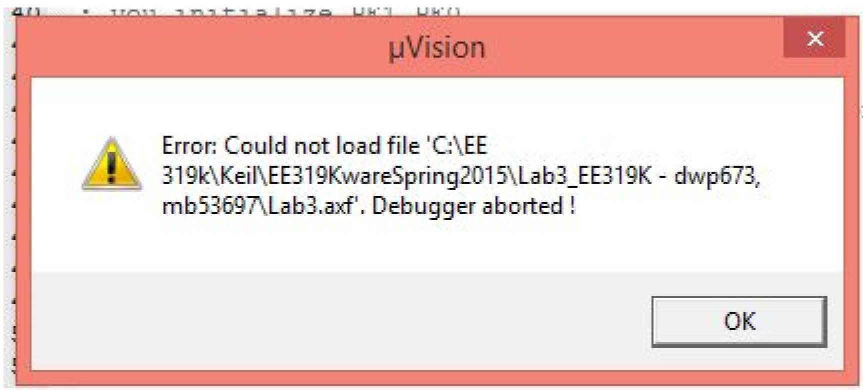
FAQ

The list of FAQ below are populated from Piazza over the semesters (thanks to the contributions of all past TAs and students). More questions may be posted so please check back regularly.

1. For my 62 ms delay, I'm having trouble getting the delay up to the right number of ms. I've tried combining multiple delay subroutines to increase the delay, but i'm still quite a ways from 62. Can anyone share a smart way of increasing the delay?

   Notice that the clock for this lab is running at 80MHz instead. If you understand how the delay operation in Lab2 works, you should be able to simply modify one number to get the delay to 62ms.

2. I get this error when I try to run the debugger in lab 3. How do i fix it?

   This typically means there is an error within your code that prevented keil from successfully compiling your code. When you hit 'build' or 'rebuild' the command window at the bottom will tell you if there is an error. It will also give you some information as to the cause of the error.

3. Is the PCB artist file downloaded from Valvano's page supposed to be an empty page when opened in PCB Artist? If so, how do we get the proper components on the page?

   Try redownloading the EE319K_TM4C123_artist.sch file on the downloads page. There should be components already drawn within this file, so try looking and scrolling all around the workspace in PCB to find them.

4. How are we supposed to determine which pair of pins has the switch across them using the ohmmeter? Each pair has the exact same resistance between them when I measure it.

   Were you measuring the resistance with the switch pressed? If so, all of the pins will be connected together and have the same, albiet small, resistance. Otherwise, if you were measuring with the switch not pressed, you should be measuring different resistances between different pairs. If you are absolutely sure this is not the case, it is possible that you could have a broken switch.

   The size of the resistance might also provide a hint as to what is going on (think about when the resistance should be very large and when it should be very small, assuming the switch is working correctly).

5. How do we measure current across a resistor?

   You put the multimeter in series with the resistor. Or you can kind of "cheat" and measure the voltage, then calculate the current. You should understand how to do both ways though.

6. What is the bottom side of a resistor?

   The lab manual is sort of confusing when it says 'bottom side of R10', but what it wants is the voltage of the LED anode side. It refers to the part of the circuit 'below' the 220 resistor and 'above' the LED.

7. How do we measure stepwise data for rows 8 -10 on the table? I'm reading fluctuating data on the multimeter.

   With Keil you can debug your circuit in real time. If you click under Projects --> Options for Target --> Debug and use the stellaris ICDI instead of the simulator you can the press debug and actually step through your code and watch how it affects your circuit. Think about when in your program should PE1 be set to zero and jump to it. You can now measure these elements knowing that PE1 is supplying no voltage. The values can be expected to vary slightly.

8. How can I save PCBArtist file as a PDF?

   (1) Take a screen shot and paste it into wherever you are putting your lab deliverables (i.e., Word or GoogleDocs.
   (2) Use something like CutePDF to "print" things to a PDF.

9. After getting my hardware wiring checked by a TA, I tried implementing it with the software and couldn't even get the LED to turn on. Are there any possible explanations for why this could be happening?

   Before moving on to circuits, make sure your program work first on the simulator. Once that is checked, make sure everything is connected securely. Circuit problems come from crappy wires most of the time.. are you using your own wires or wires from the check out desks? A multimeter will be your best friend when it comes to debugging a circuit.

10. Do we actually need to add a capacitor as shown in the PCB artist example? What impact will it have to our system?

    The capacitor acts as a filter for the system's 5V power line. We are using the 5V line to power the 7406 buffer, which is being used as the current sink for our LED. When we are toggling it, and due to a ton of other random factors, we will see current and voltage fluctuations (typically pretty small in this type of system). The capacitor across the power line and ground will act as a sort of dampener/filter, resisting changes in voltage. This is to keep the 5V as close to 5V as possible amid voltage fluctuations. You circuit will likely work fine most of the time, but as your circuits become larger, more complex, and more dynamic, you may see certain systems drop below their necessary minimum Vcc and not function properly. You can think of the capacitor as a large water reservoir. Excess voltage will fill the tank, while a deficit will be countered by the excess "water" in the tank flowing out.

11. My logic analyzer no longer is showing the ports on the side. When I run it the analyzer is completely blank. Any idea on how to get the values of the ports back on the logic analyzer?

    See FAQ 2 in Lab2 descriptor.