School of Electrical Engineering and Computer Science

CptS466/566: Embedded Systems

Fall 2018

Project 2 (P2)

Code & Report Due: 9/20/2018 @ 11:59pm (Blackboard)

Demo Due: 9/20/2018 @ 1:25pm (Classroom)

1. Preparation

You will need the LaunchPad, two switches, two 10 kΩ resistors, three LEDs, and three 470Ω resistors.

2. Project Description

In this project, you will learn how to interface switch and LED components with your Launch pad to build a simple traffic light control system. To capture the presence of a vehicle, we need some kind of sensor in real-life (e.g., cameras). In your prototype, you can potentially use a switch as input to your system. The switch indicates the presence of a vehicle at the intersection. You will also need to sense the presence of pedestrians at the intersection. You can potentially use a second switch to indicate the presence of a pedestrian (and that they would like to cross the intersection) in your prototype system.

You will need to build circuits (using switches, LEDs, resistors, etc.) on the breadboard and connect them to the LaunchPad.

Here are some additional requirements for this project:

- The traffic light is initially ‘Green’ (i.e., the green signal/lED is ON while yellow and red signals are OFF).
- Upon arrival of a pedestrian, the traffic light turns ‘Yellow’ and after a delay it turns ‘Red’.
- A ‘Red’ traffic light turns ‘Green’ when a vehicle arrives.

You are required to make any reasonable assumption for your system to emulate real-life situations. The goal is not to provide you with all design decision choices and system details. Instead, you need to think (starting with high-level requirements discussed above) about details of the system, other requirements, etc. For example, you will need to choose reasonable delay while in Yellow state. Also, it is clear that if a car is passing the intersection, arrival of a second car will not chance the already Green signal (pressing the switch for the second/third/… times will not change the green signal).

You need to follow guidelines in a system development process. Document your development phases (requirements, design, development, etc.) and write the software that satisfies the requirements for this lab.

Note: the best approach to time delay will be to use a hardware timer, which we will learn in next lab. In this lab, however, we do not expect you to use the hardware timer.

2. What to submit?

Submit a .zip file with the following content:

- Your well commented C source code implementing the application described above.
- A report that documents the development process, your observations, and a discussion of your design decisions.

Show a demo of your application in the class on the due date.

3. Report

In a separate written document (in Word or PDF), compile sections A through C as follows:

A: Requirements document. Treat this section as a requirements document and outline various requirements of the project in separate categories including (1) overview; (2) Function description; and (3) Deliverables. The overview section should contain, at least, information about objective, process, and interaction with existing systems. The function description section should contain, as a minimum, information about prototype, performance, and usability. The deliverables section should list or discuss any deliverables such as the built prototype, and report.

B: Design document. Draw a data flow graph showing schematic of the system and flow of the data from inputs to outputs. Your data flow graph should precisely shows the I/O pins, LEDs, and Switches, and the way they are used in conjunction with the microcontroller. Also, draw a flowchart showing the software design. The flowchart needs to include enough details about the hardware components used.

C: Discussions. In this section discuss how you implemented delay in your program. Discuss any specific observations that you had, any difficulties that you ran into while designing or testing the system. Discuss limitations of your developed system and potential ways that those limitations can be addressed in a future revision of your design.

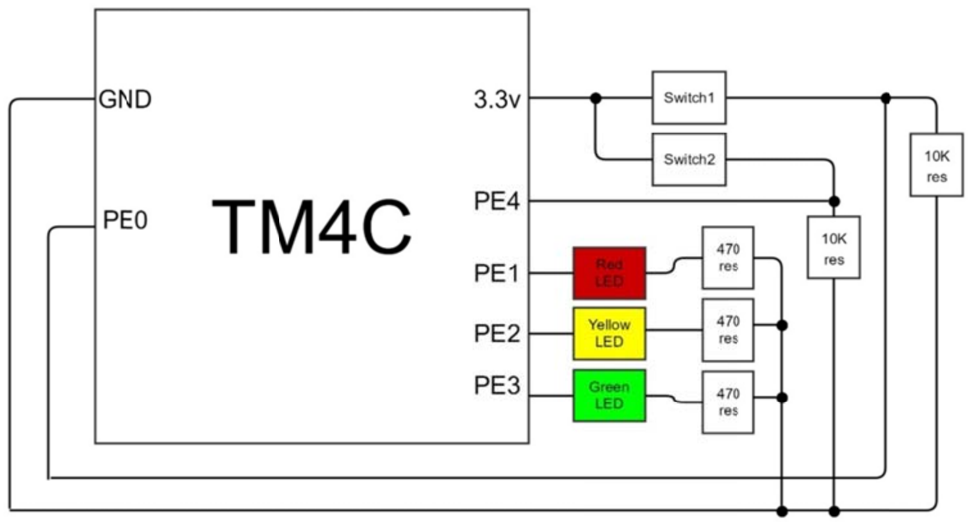
4. Grading

Assume that the whole assignment is worth 100 points.

- 35 pts for well commented and correct C source code satisfying the project requirements.
- 35 pts for report.
- 30 pts for demo.

5. Appendix A – Example System Schematic

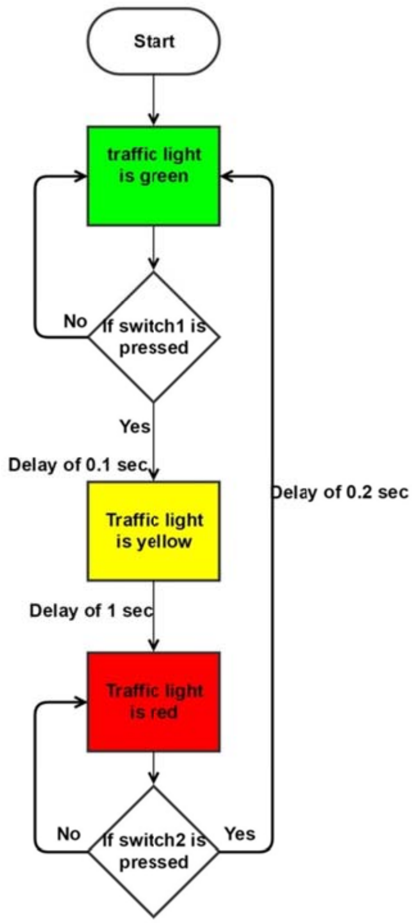
A potential schematic of the system is shown below. This could be used as a basis for your data flow graph design. The figure below shows a circuit diagram connecting the switch and LED components to the microcontroller.



6. Appendix B – Example Flowchart

Figure below illustrates a potential flowchart diagram which provides an overview of the system functionality. This diagram is not meant to be comprehensive and the numbers (e.g., delay values) are only examples. You will need to complete and redraw the flowchart based on your own design parameters.

Note that the example flowchart here assumes that the switch interface implements a positive logic and that the LED interface implements a positive logic as well. The assumption here is that you attach switches and LEDs to your protoboard (the white piece), and interface them with your TM4C123 microcontroller.



7. Appendix C – Reading Microcontroller Datasheet

In this section, we discuss how can program the Launchpad with the provided demo code.

Let us look at few pages in the TM4C123 microcontroller datasheet. The datasheet is provided with this project. Alternatively, you can download the data sheet for the TM4C123 microcontroller here:

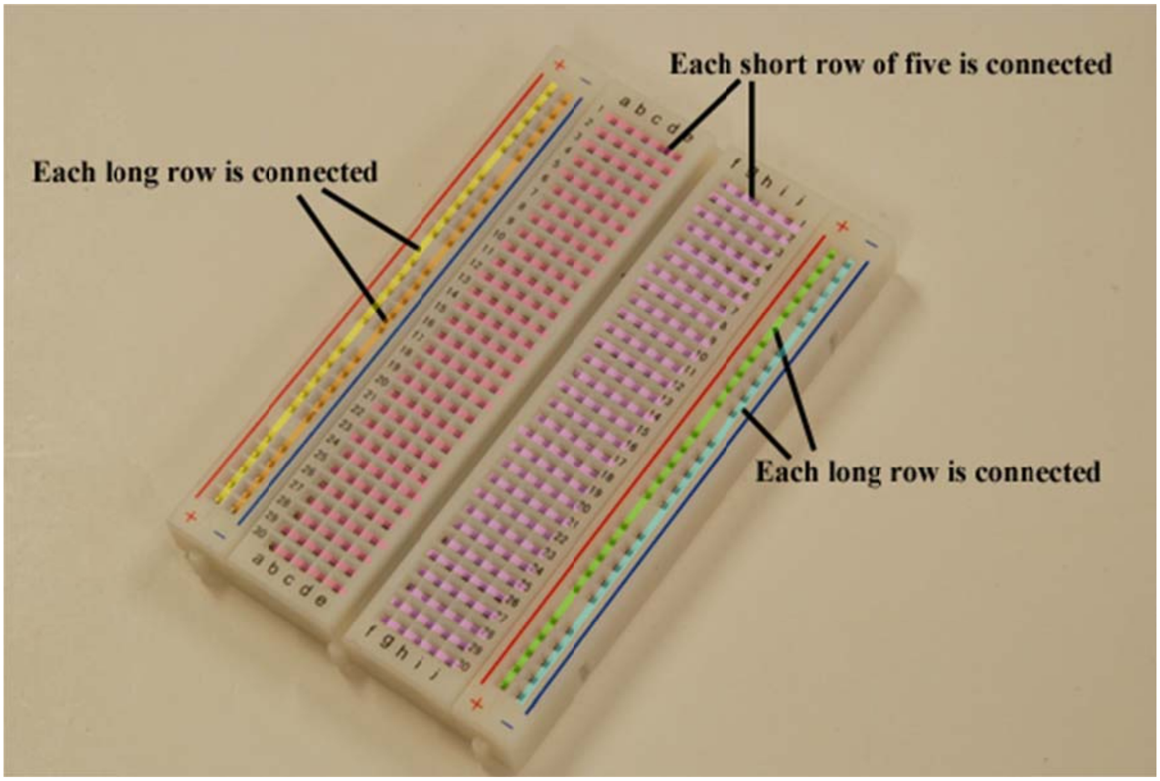
https://prod-edxapp.edx-cdn.org/assets/courseware/v1/fa65824471e6a00ada3317b79943c1ae/asset-v1:UTAustinX+UT.6.10x+2T2018+type@asset+block/tm4c123gh6pm.pdf

- Look at the block diagram on page 46 to see the amounts of RAM and ROM memories that this microcontroller has.
- Look at page 649 to see how many I/O pins exist.

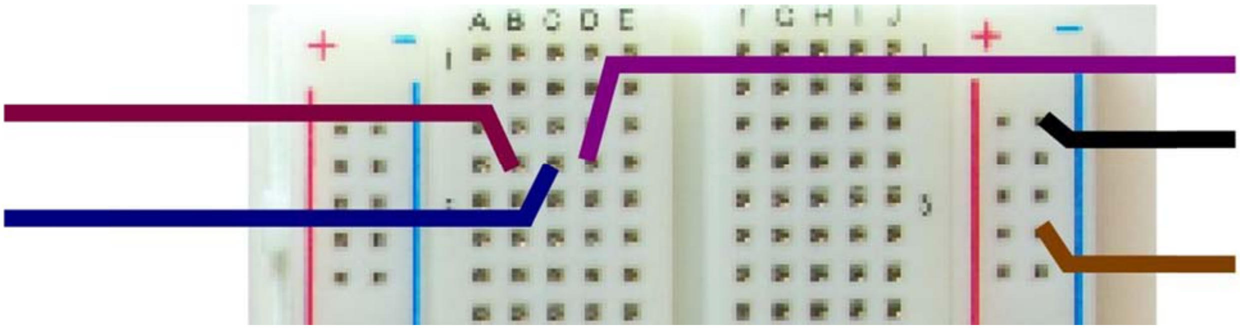
8. Appendix D – Interfacing Switches and LEDs

Solderless breadboard

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 8.1. The long rows of holes along the outer sides of the protoboard are electrically connected. Some protoboards like the one in Figure 8.1 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. 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 holes is electrically connected. We usually insert components into these holes. If integrated circuits (IC) are to be placed on the protoboard, it is done such that the two rows of pins straddle the center valley.

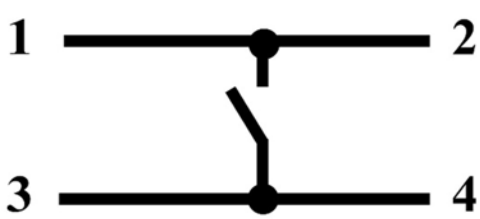


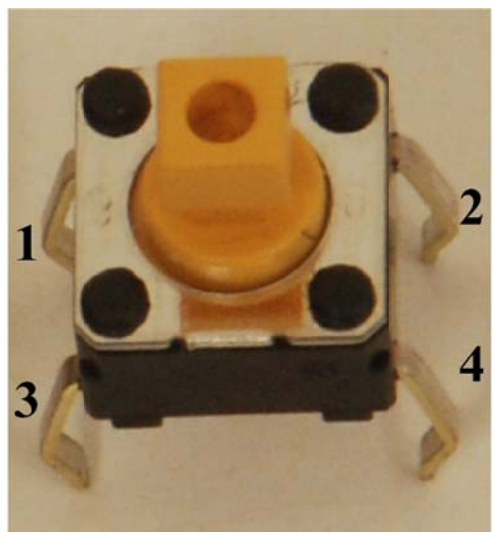
For example In the below figure the blue, red, and purple wires are connected. Similarly, the black and brown wires are connected.



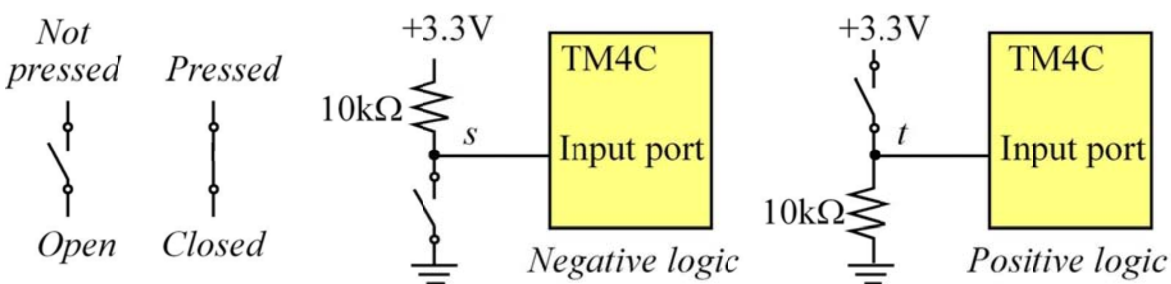
Interfacing a Switch

Figure below is a B3F tactile switch. For this switch, pins 1 and 2 are always connected. Pins 3 and 4 are also always connected. If the switch is pressed, pins 1 and 3 are shorted together. If the switch is not pressed, pins 1 and 3 are not connected, open circuit.

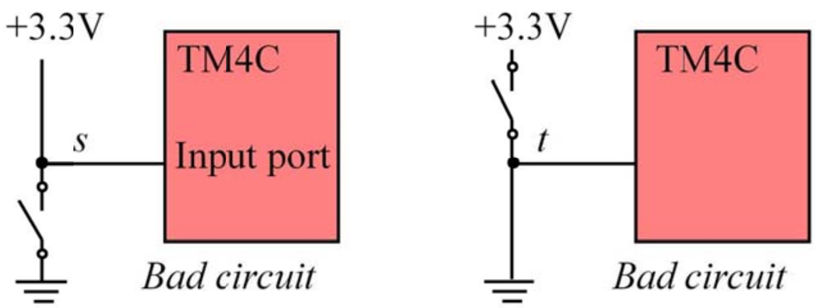




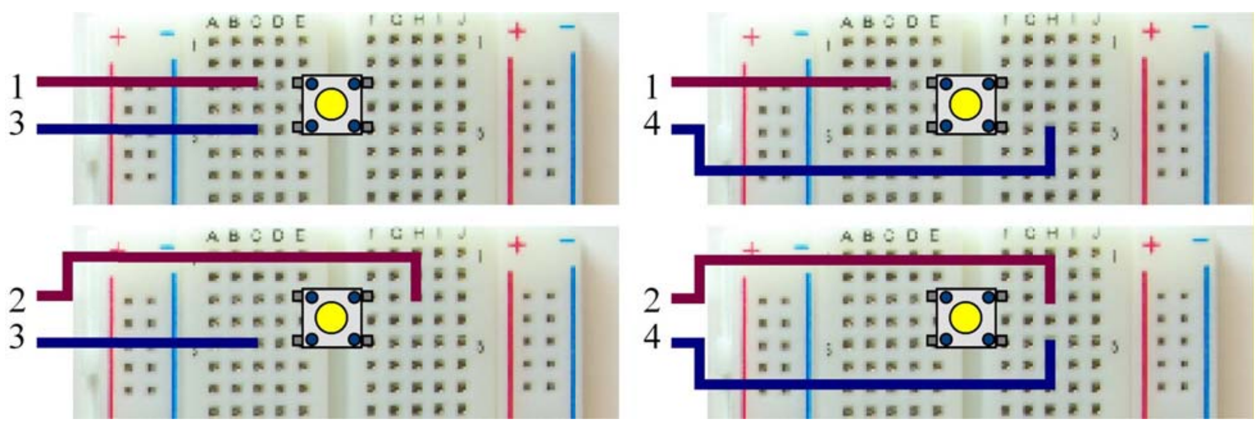
The switches are shown as little open circles in next figure. In a normally open switch (NO), the resistance between the connections is infinite (over 100 MΩ on the B3F tactile switch) if the switch is not pressed. The resistance is zero (under 0.1 Ω on the B3F tactile switch) if the switch is pressed. To convert the infinite/zero resistance into a digital signal, we can use a pull-down resistor to ground or a pull-up resistor to +3.3V as shown in next figure notice that 10 kΩ is 100,000 times larger than the on-resistance of the switch and 10,000 times smaller than its off-resistance. Another way to choose the pull-down or pull-up resistor is to consider the input current of the microcontroller input pin. The current into the microcontroller will be less than 2µA (shown as IIL and IIH in the data sheet). So, if the current into microcontroller is 2µA, then the voltage drop across the 10 kΩ resistor will be 0.02 V, which is negligibly small. With a pull-down resistor, the digital signal will be low if the switch is not pressed and high if the switch is pressed (right figure). This is defined as positive logic because the asserted state is a logic high. Conversely, with a pull-up resistor, the digital signal will be high if the switch is not pressed and low if the switch is pressed (middle figure). This is defined as negative logic because the asserted state is a logic low.



Do not replace the 10k resistor with a zero-ohm wire. When you press the switch there will be a short circuit from +3.3V to ground causing sparks and damage to the switch and the microcontroller board.

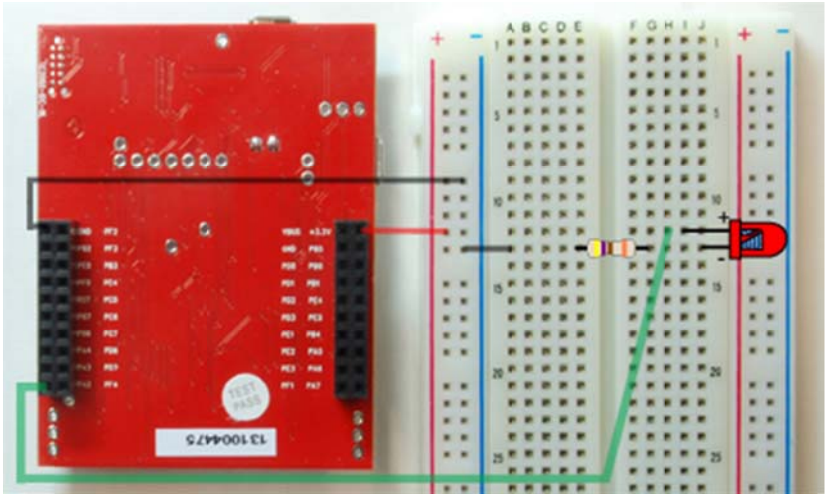


There are four possible connections that will work. Looking at below figure, you see the switch circuit can be built across pins {1,3} {1,4} {2,3} or {2,4}. My favorite scheme is to use pins that are across the diagonal. Another scheme for figuring it out is to use an ohmmeter to test across which two pins the switch exists.



LED Interfaces

A light emitting diode (LED) emits light when an electric current passes through it. LEDs have polarity, meaning current must pass from anode to cathode to activate. The anode is labelled a or + , and cathode is labelled 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.



Construction of the interface of an LED to a microcomputer output (Above Figure). The yellow-purple-brown resistor is 470Ω. It doesn't matter what color the wires are, but in this figure the wires are black, red and green. The two black wires are ground, the red wire is +3.3V, and the green wire is the signal Out, which connects PA2 of the microcontroller to the positive side of the LED.