

Learning Goals:

- Learn how to create a circuit that sends input from a connected component to the Arduino.
- Get practice with slightly more complicated Arduino programming.
- Refresh your memory on binary conversion.

Prelab:

Set up a push button and LED with the Arduino.

When the push button is pressed the Arduino should read in the value from the push button, and then toggle the LED.

(if the LED is on, it goes off; if the LED is off it goes on).

A very simple push button example is here: <https://www.arduino.cc/en/Tutorial/Button>

A helpful resource is Arduino lesson 5: <http://www.ladyada.net/learn/arduino/lesson5.html>

Using the serial monitor to read print statements from your Arduino can be very helpful here.

Lab 2 - Due: Wednesday 2/6/2019

Use two push buttons and three LEDs for this circuit. The push buttons will change an internal “3-bit count”. The LED’s will display the value of this internal “3-bit count”.

- One push button will increment the 3-bit count. This is the “up button”.
- The other push button should decrement the 3-bit count. This is the “down button”.
- The 3-bit count value will be displayed on the three LEDs in binary
 - one LED per binary digit
 - a binary digit of 1 is indicated when the LED is on

You should write code to keep track of the number of button presses and algorithmically convert that number to a binary output. Your code should NOT just be implementing a giant lookup table to translate to binary, and solutions which do so will not receive credit.

Your lights should increment (or decrement) by one each time a push button is pressed. If we continually depress the push button, the lights should not run up (or down) to 7 (or 0) (i.e. it should only change the value once per button press).

Also document what happens when you reach 7 and press the “up button” (or reach 0 and press the “down button”).

What should I include with my .ino Code File?

As with any code file, it should be written in Good Coding Style: in a manner that will help other people read and understand the intent, purpose, operation of the code. So your code must include:

- Name the .ino file with your NetId and Lab Number – 3 points
 - I.E. something like: ptroy4Lab2.ino
- Header Comments (including the following) - 12 points
 - // FirstName LastName, UIN and NetID
 - // Lab x - Title

- // Description - what is this code supposed to do?
- // Include any assumptions you may have made, what do you expect from the hardware, pinouts, particular arduino versions, etc.
- // References - where did you find code snippets, ideas, inspirations? if no references used say: "no references used"
- Code is well documented/formatted with comments, indentations, and descriptive variable names - 10 points
- Actual code - the functions in the cpp/ino file - 25 points

Total: 50 points per lab based on the code file submitted.

So, if you just submit working code, you will only get half of the total allocated points.

Academic Integrity Guidelines: You may use any resources linked from this lab, or posted by the professor or TAs on piazza/class web page/etc. You should not look at any other internet resources for this. This is an individual assignment, and should be completed on your own. You should not show anyone your code, or look at anyone else's code. You are responsible for writing the code on your own that converts the input to binary which lights up the proper LEDs.

To be considered completed "on time", Lab needs to be demonstrated by end of Lab on Wednesday 2/6/2019, and the code submitted to Gradescope, before 11:59pm on 2/6/2019.

Late Policy

- Late Submission - Submitted and/or demonstrated later that week (before Friday 2/8/19 11:59pm) 25% Penalty
- Late submission – Submitted and/or demonstrated during the following week (before Friday 2/15/19 11:59pm) 50% Penalty