Three Blinking Lights

Academic Integrity Guidelines
This is an individual assignment, and all code you write should be your own. You should not use any resources other than those linked to in the assignment, or provided by the professor or TAs. You should not look at anyone else's code, or show anyone your code.

Learning Goals:
- Get familiar with the Arduino programming environment.
- Learn how to build a circuit to connect external components to the Arduino.
- Learn how to program the Arduino to control external components.
- Bonus: Pretty blinking lights.

As working with microcontrollers and electronics is new to most of you, don't worry if some of this seems weird or confusing. Please attend the lab during your scheduled lab time, or any of the office hours, in order to get help setting everything up.

Prelab: (Things to do on your own to understand how to do the actual lab!)
Get a blinking LED on Pin 13. (Note: Pin 13 controls both the onboard LED, and the external pin 13. If you do this correctly, your onboard and external LED should blink at the same time.)

You will first need to get the Arduino IDE set up on your machine. You can download it here:
- https://www.arduino.cc/en/software.

There are plenty of tutorials for setting up Arduinos on the Arduino website.

You can follow one of the many online tutorials on how to do this, such as:
- http://arduino.cc/en/Tutorial/Blink.

Notes:
- A good way to approach this is to get your onboard LED blinking first, and then connect an external LED.
- The Arduino IDE comes with sample code to get your LED blinking.

Lab 1: Due during your scheduled lab time on Week 3 : Monday 1/22/2024

Materials
- 1x Arduino
- 1x Breadboard
- 1x Red LED
- 1x Green LED
- 2x 220 Ohm resistors
- Many wires

Use the onboard LED on pin 13, a red LED on a second digital pin of your choice and a green LED on a third digital pin of your choice.

Have these three LED's cycle so only one LED is on at any given point in time. So your onboard LED should turn on, then it should turn off and your red LED should turn on, then it should turn off and your

green LED should turn on, then onboard, then red, then green, etc . . . The digital pins should be labeled 0-13 on your board.

You will NOT receive full credit if you use delay() anywhere in your code. This is a 300-level Computer Science course, and a certain level of programming maturity is expected from the students.

- The millis() function (https://www.arduino.cc/reference/en/language/functions/time/millis/) might be helpful for this.
- You can also check this article Why Use millis() Instead of delay()? (https://www.norwegiancreations.com/2017/09/arduino-tutorial-using-millis-instead-of-delay/).
- https://docs.arduino.cc/built-in-examples/digital/BlinkWithoutDelay/
- https://forum.arduino.cc/t/using-millis-for-timing-a-beginners-guide/483573
- Note: putting a millis() statement inside a loop statement is the same as using a delay()!

Each off-board LED should be connected to a 220 Ohm resistor. Note that this will mean you have to connect two separate LEDs to the ground pin. We recommend using a breadboard for this.

Note that this will mean you have to connect two separate LEDs to the ground pin. I recommend using a breadboard for this.

If you wish, you may wish to connect a third off-board LED to the Digital Pin 13. If so, pick a different color than Red or Green for the LED. Using a third off-board LED is not a requirement for the lab (but it is much easier to see a third off-board LED blinking than the on-board LED).

Due by Lab Time Week 3: Have a TA check out your board by the end of your scheduled lab on either Monday 1/22/24, **AND** submit the code via Gradescope prior to the demonstration of your lab.

Import notes concerning the submission of your Arduino Code for the Lab:

1. Your code must be written as an .ino code file using C/C++ as the base language. Other programming languages are not accepted for CS 362.

2. If you fail to submit your code to Gradescope before you demonstrate your project to the TA, the TA will not have an entry in Gradescope to record your results. Thus, your demo would be pointless and would require you to demonstrate your lab again.

3. Updating your Gradescope submission after you demonstrate your project to TA will cause another entry in the Gradescope system which will cause any points from your demonstration to be lost. Thus requiring you to demonstrate your lab again.

4. Failure to follow these points will result in a large deduction of the points for your lab.

5. If possible, just prior to your demonstration add a comment at the end of your Header Comments with the Date, Time and Name of the TA to whom you are demonstrating. If a question comes up regarding the grading or demonstration of your lab, having that information can mean the difference between successfully resolving the question or not getting it resolved.

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: ptroy4Lab1.ino
  - where "ptroy4" is replaced with your own NetID
  - your NetID is the part of your UIC email address before the "@uic.edu"
- 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 without doing a demo, you will only get half of the total allocated points.

Academic Integrity Guidelines for this lab: You are allowed to consult resources linked from this lab write up, and anything provided by the professor or TAs via blackboard, as well as any resources you need in order to get the Arduino software and hardware working with your computer. You should NOT look up any of the code you need to get the LEDs to turn on and off as described, you should be able to figure this out yourself from the materials provided. This is an individual assignment, and you should not look at anyone else's code, or show anyone your code.

Lab Late Policy

- Lateness is determined by the time the lab is demonstrated, not when the .ino file is submitted.
- Labs that are not demonstrated get a score of 0.
  - -50% for no demonstration
  - -50% for being late
- Any lab demonstrated after 4:50pm on the day on the day when the lab is due is considered late.
- Late Deadline 1: within 3 days late
  - Demonstrated after the end of Lab Time on 1/22, but before 11:59pm on Thursday 1/25
  - 25% Penalty
- Late Deadline 2: up to 1 week late
  - Demonstrated between Friday 1/26 but before 11:59pm Monday 1/29/2024.
  - 50% Penalty
- Labs Demonstrated after 1 week late receive a score of 0 (as if they were never demonstrated).