Assignment 1

Geosc 597-003
Techniques of Geophysical Experimentation

February 9, 2021

Objective(s): *Build on in-class blinking an LED (diode).*

Activity 1 - Blinky 2

Much like the in-class activity, we will control the blinking of ***2 external LEDs*** on a breadboard.

Materials

- Arduino UNO
- USB Cable
- Breadboard
- LEDs (2 of your favorite colors)

- 330 Ω resistors (Orange-Orange-Brown)
- M/M Jumper wires
- Computer (Mac, Linux, Windows)

Procedure

- Hook up the LEDs, resistors, and Arduino using the jumper wires and breadboard, similar to the in-class activity.

- LEDs are *polarized* components meaning they have a certain way they need to be in the circuit. On LEDs the short leg next to the flat edge is the ground (-) connection.

- You will need to bend the legs of resistors to use them on the breadboard, you can do this with your hands or small pliers.

- Based on the code we used in the in-class Blinky activity, write a new program that blinks the LEDs in an alternating pattern.

- Get creative. What patterns can you make? Can you add more LEDs or change the pin assignment for each LED?

<div align="center">

Grading Rubric

Objective	Points
Code compiles	10
LEDs blink in patterns	15
Total	25

</div>

Activity 2 - Stoplight

In this activity you will make a simple single stoplight controller with an Arduino UNO and some LEDs. You will become familiar with using the Arduino programming environment and learn how to use the General Purpose Input/Output (GPIO) pins on the microcontroller. You will also practice using good software design technique by implementing well known design patterns and making maintainable code.

Materials

- Arduino UNO
- USB Cable
- Breadboard
- LEDs (Red, Yellow, Green)
- 330 Ω resistors (Orange-Orange-Brown)

- 10k Ω resistors (Brown-Black-Orange)
- Push button (momentary-on type)
- M/M Jumper wires
- Computer (Mac, Linux, Windows)

Procedure

- Connect the button, stop, caution, go, and left turn LEDs as shown in diagram.

- Start the Arduino IDE. Open the Blink example from: **File → Examples → 01.Basics → Blink**. Read the comments and make sure you understand how it works.

- Connect your Arduino and hit the upload button. If it fails, check the board and port settings (in the **Tools** menu). Make sure the on-board LED is blinking to show a successful program upload.

- Change the pin number in the blink example to that of one of your LEDs. Make sure that the LED on the breadboard blinks, if not, you need to check the connections. Do this for each of the 4 LEDs.

- Draw a state machine diagram to meet the specifications of the attached requirements. Turn this in with the assignment! https://creately.com/lp/state-machine-diagram-tool/

- Build the state machine in the Arduino IDE and test it on your stoplight. Your final code should be commented, compile and run, and meet the specifications. Be sure to use good coding practices! Your code will be tested/graded by an identical Arduino setup.

Requirements

- Begins in the red light state.

- Red light cycle lasts for 3 seconds.

- Yellow light cycle lasts for 1.5 seconds.

- Green light cycle lasts for 3 seconds.

- Works like a normal stoplight would, only one light on at a time and in the normal order (Red - Green - Yellow - Red).

- If a car was present in the left turn lane (simulated by holding down the push button) **before** the green light state, add a green left turn light for 2 seconds. If no car is present, repeat the cycle.

- Uses the state machine implementation with functionalized code. No interrupts allowed!

Grading Rubric	
Objective	*Points*
Code compiles	30
Meets project requirements	45
Total	75

Example: https://www.youtube.com/embed/ltXPpmL2szE

Note: The green button seen in video is different than the one from the Elegoo kit.

What to upload to Canvas

You should upload the following with *consistent file names*:

- Your Arduino codes. Make sure it works before you upload.

    - **Fully comment your code!** The blinky example from the Arduino IDE is commented so that others can read the code and understands what is executed.
    - **Naming convention:** username_blinky2.ino and username_stoplight.ino
      Example: cew52_blinky2.ino, cew52_stoplight.ino.

- A photograph of your board or a movie file showing the board and your blinking lights.
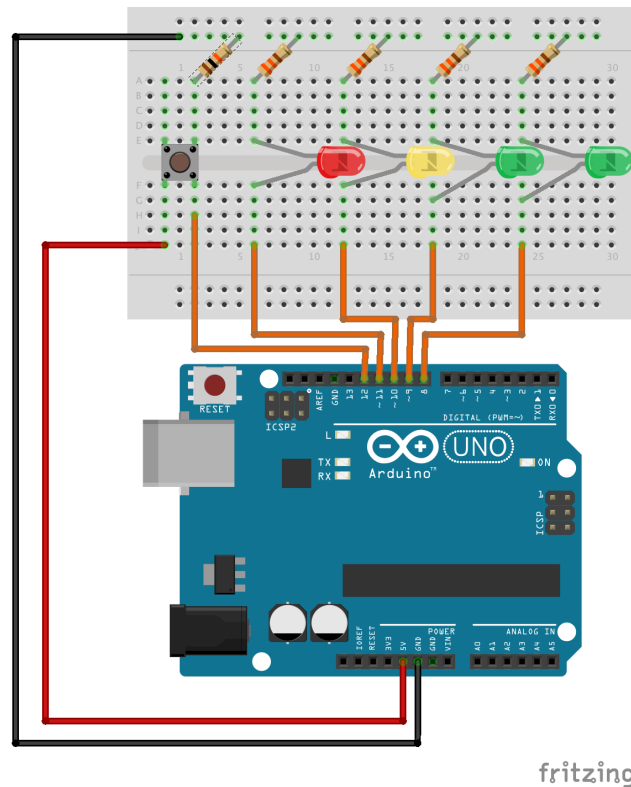
    - **Naming convention:** username_stoplight.mov

Figure 1: Wiring diagram for stoplight activity