CS 362 Lab 9 Spring 2024

Learning Goal:

- Wire and use the RGB LED
- Understanding Pulse Width Modulation
- Using multiple analog inputs with the two-axis joystick and potentiometer

Lab: Due Monday 4/1/2024 at your registered lab time.

Materials

- 1x Arduino
- 1x Breadboard
- 1x Two-axis joystick
- 1x Potentiometer
- 6x 220 Ohm resistor (or others as needed)
- 1x RGB LED
- 1x Red LED
- 1x Green LED
- 1x Blue LED
- Wires

Wire your Arduino so that it has 3 Analog inputs. Ideally, two inputs are generated from using the two-axis joystick and the third input is generated using the potentiometer. These three inputs are to be used to control the color of the RGB LED. The first input (x-axis on the joystick) is to control the Red value. The second input (y-axis on the joystick) is to control the Green value. The third input (the potentiometer) is to control the Blue value. The following web pages discuss using the two-axis joystick and potentiometer:

- https://arduinogetstarted.com/tutorials/arduino-joystick
- https://docs.arduino.cc/built-in-examples/usb/JoystickMouseControl/
- https://www.instructables.com/How-to-use-Potentiometer-Arduino-Tutorial/
- https://www.arduino.cc/en/tutorial/potentiometer

If you do not have a two-axis joystick (most kits do include them), you may use multiple potentiometers, or wire some buttons/switch along with a single potentiometer to create the different inputs for the Red, Green and Blue values. Also note that the two-axis joystick contains a button that is not used with this lab.

The RGB LED uses Pulse Width Modulation to drive the amount of Red, Green, and Blue produced by the LED. The PWM pins are the digital pins on the Arduino UNO that are marked with the tilde symbol ~. On the Arduino Uno, the only official supported PWM pins are pins 3, 5, 6, 9, 10, and 11. Note that some manufacturers may include other pins as PWM pins, but we will restrict usage in this lab to the official pins only! The following webpages talk about using PWM pins and the RGB LED. Note that not all restrict usage to official PWM pins:

- https://support.arduino.cc/hc/en-us/articles/9350537961500-Use-PWM-output-with-Arduino
- https://docs.arduino.cc/tutorials/generic/secrets-of-arduino-pwm/
- https://projecthub.arduino.cc/semsemharaz/interfacing-rgb-led-with-arduino-b59902
- https://howtomechatronics.com/tutorials/arduino/how-to-use-a-rgb-led-with-arduino/

- https://arduinogetstarted.com/tutorials/arduino-rgb-led

You must also show the individual Red, Green and Blue values separate from the RGB LED. Wire a Red LED, a Green LED, and a Blue LED to show each separate color. Since there are 6 PWM pins, you could use a second PWM pin for each of these single color LEDs. Or you can use the same PWM pins as the corresponding RGB value and set up the single color LED in parallel or in series with the RGB LED. Be sure to properly set up the resistors properly so you don't "blow up" any LEDs. These single color LEDs should change from being "completely on" to "dim" to "off" as the corresponding input values change from maximum-on to middle-amount to no-input.

Pulse Width Modulation is basically turning the pin on and off very quickly. The higher the value for the PWM output, the longer the pin is on. The length of the PWM signal is about 500Hz or 2 milliseconds (based on readings from the above web pages and other articles found). Thus, if you allow at least 10 milliseconds between sending values to a PWM pin, they should work properly. There are reported issues if you don't wait long enough before sending another value to a PWM pin. This wait is to be done using the millis() function and NOT using the delay() function!

Your code must be submitted to Gradescope BEFORE you demo your 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
- Late Submission 1
    - Demonstrated before 11:59pm Thursday after Lab Due Date
    - 25% Penalty
- Late submission 2
    - Demonstrated between Friday and Monday after Lab Due Date
    - 50% Penalty

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
    - I.E. something like: ptroy4Lab2.ino
- Header Comments (including the following)
    - // 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
- Actual code - the functions in the cpp/ino file

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.