16-681: MRSD Project Course

Homework 2: Introduction to programming, aka the I/O, serial, interrupt-aganza

Due Date:

Tuesday, September 17 11:59pm. Submit code on Blackboard.

Show your working prototype to the TAs during office hours.

Neil: 6-9pm Monday, Ben: 6-9pm Tuesday

Eratta:

In the previous version of this assignment, the due date was incorrect (Monday, 16th). It has been fixed.

The wires and resistors for the Red and Green LEDs were reversed. This document has the correct wiring.

In the State 0 text, Button 1 was erroneously called Button 0 twice. This has been corrected.

Purpose:

This assignment is to provide an introduction to programming, state machines, and basic Arduino libraries.

Teamwork:

This assignment is to be completed individually.

Materials:

The following materials will be provided to you.

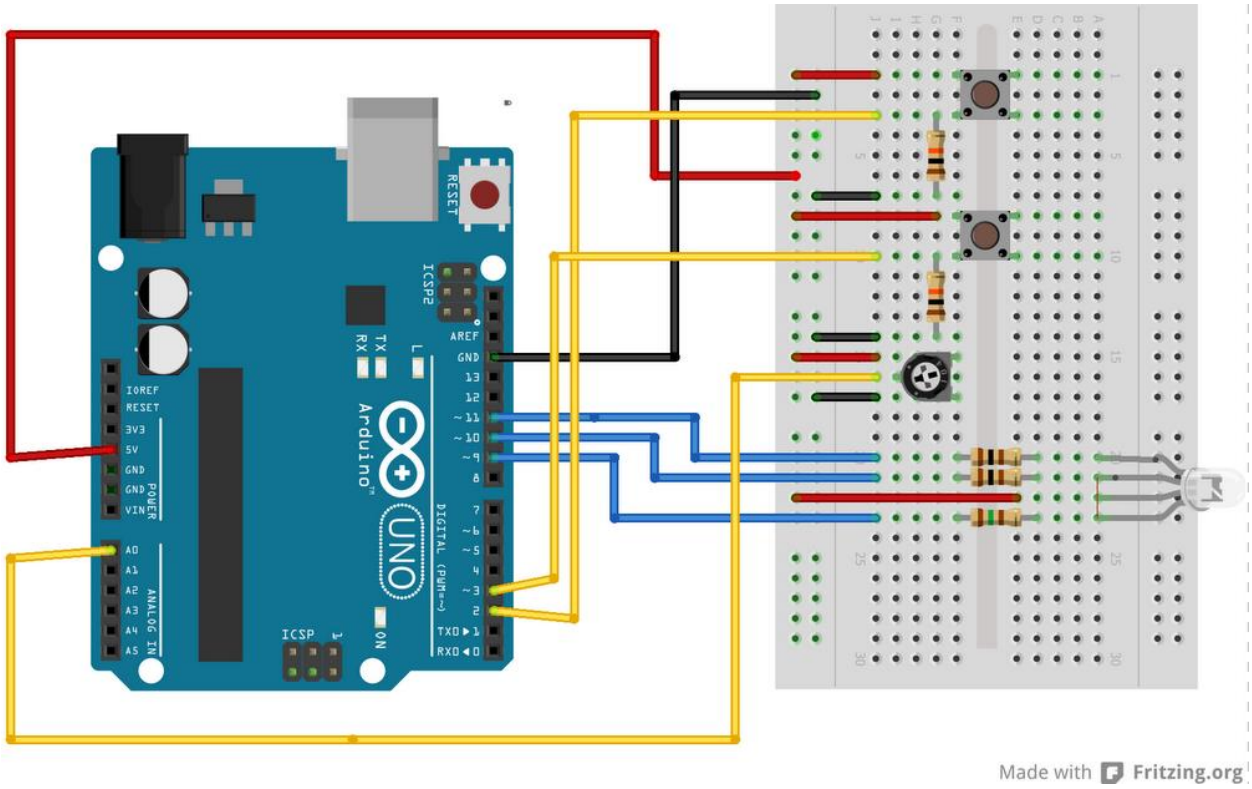
- Arduino Uno
- USB A-B Cable
- Breadboard
- Button (2)
- RGB LED
- 10kΩ Rotary Potentiometer
- 150Ω Resistor
- 100Ω Resistor (2)
- 10kΩ Resistor (2)
- Wire

Overview:

Write a program for the Arduino to control the brightness and hue of a tri-color LED. This control will be implemented using a number of inputs: digital, analog, and serial. A simple finite state machine will be used to decide the source of control at any time.

Wiring:

To enable you to focus on the software instead of the hardware, the circuit has already been designed. **Be sure to follow this circuit**, because your code will be evaluated by running it on a setup wired in this configuration.



Of note, the sensors/LEDs will be connected to the following pins:

2: Button 0

3: Button 1

9: Red LED

10: Green LED

11: Blue LED

A0: Potentiometer

Tip: The buttons are not intended for breadboards. To insert buttons, try straightening the pins and giving them a slight axial twist.

State control:

Switching between states shall happen every time Button 0 is pressed. The states shall transition in order (0, 1, 2, 0, 1, 2, …). Every time Button 0 is pressed, one and only one state shall advance. Holding or releasing Button 0 shall have no effect. This state machine switching shall be implemented using an interrupt. Button 0 has been wired to an interrupt pin so this is possible.

States:

Each state must behave in the following way. Each state also has a "bonus challenge" in case you want an additional task for more practice. For each state, either behavior is acceptable and will not lose or gain points (assuming it works).

In each state, you shall print to Serial whenever the state changes (including the current state value). It is also helpful to print the values being read from the sensors and being written to the LEDs to help with debugging.

Behavior during each state is defined as follows:

0. Each time Button 1 is pressed, the color of the LED shall toggle once between on and off (if the LED is off, it should turn on; if the LED is on, it should turn off). Holding or releasing Button 1 shall have no effect. In this state, the LED can be treated as a binary switch, where the controlling pin (for each color) can be set LOW to turn it on and set HIGH to turn it off. The key here is that you are reading a digital input and writing to a digital output each time Button 1 is pressed.
    a. Bonus: If you desire, you may instead have the LED switch between colors by changing which of the colors are on or off each time the button is pressed. For example, cycling through red-green-blue, or red-yellow-green-cyan-blue-magenta.
1. The brightness of the LED is controlled by turning the potentiometer. As the potentiometer is turned to the right, the LED will become brighter, and as it is turned to the left, the LED will become dimmer. The key here is you are reading an analog input and writing to an analog output.
    a. Bonus: If you the desire, you may instead change the hue of the LED (by dimming the red, green, and blue together, but at individual values, such as out of phase sine waves).
2. The color of the LED is controlled by sending serial commands. The commands will be structured in the following way: a letter followed by a number. The letter will be 'r', 'g', or 'b', and the number will be an integer in range [0, 255]. The letters indicate which channel of the LED is changing (red, green, and blue respectively), and the number indicates the desired brightness of that channel. For example: "r255" means that the red LED channel should be set to full brightness; "g0" means the green LED channel should be set to off, and "b128" means the blue LED channel should be set to half brightness. Don't worry about non-linearity in the LED brightness levels. A newline will be present after each command. If a "garbage" command is received (something other than the above specified format), just discard the line.
    a. Bonus: Accept more than one command on a line (separated by spaces). For example, to set all the LEDs off at the same time, the command "r0 g0 b0" can be sent.
    b. Bonus: Create other possible commands, such as an automated sequence fading/flashing different colors. Document these in your code.

References:

Arduino Tutorials: http://arduino.cc/en/Tutorial/HomePage

Arduino language reference: http://arduino.cc/en/Reference/HomePage

Finite State Machine tutorial: http://www.mathertel.de/Arduino/FiniteStateMachine.aspx