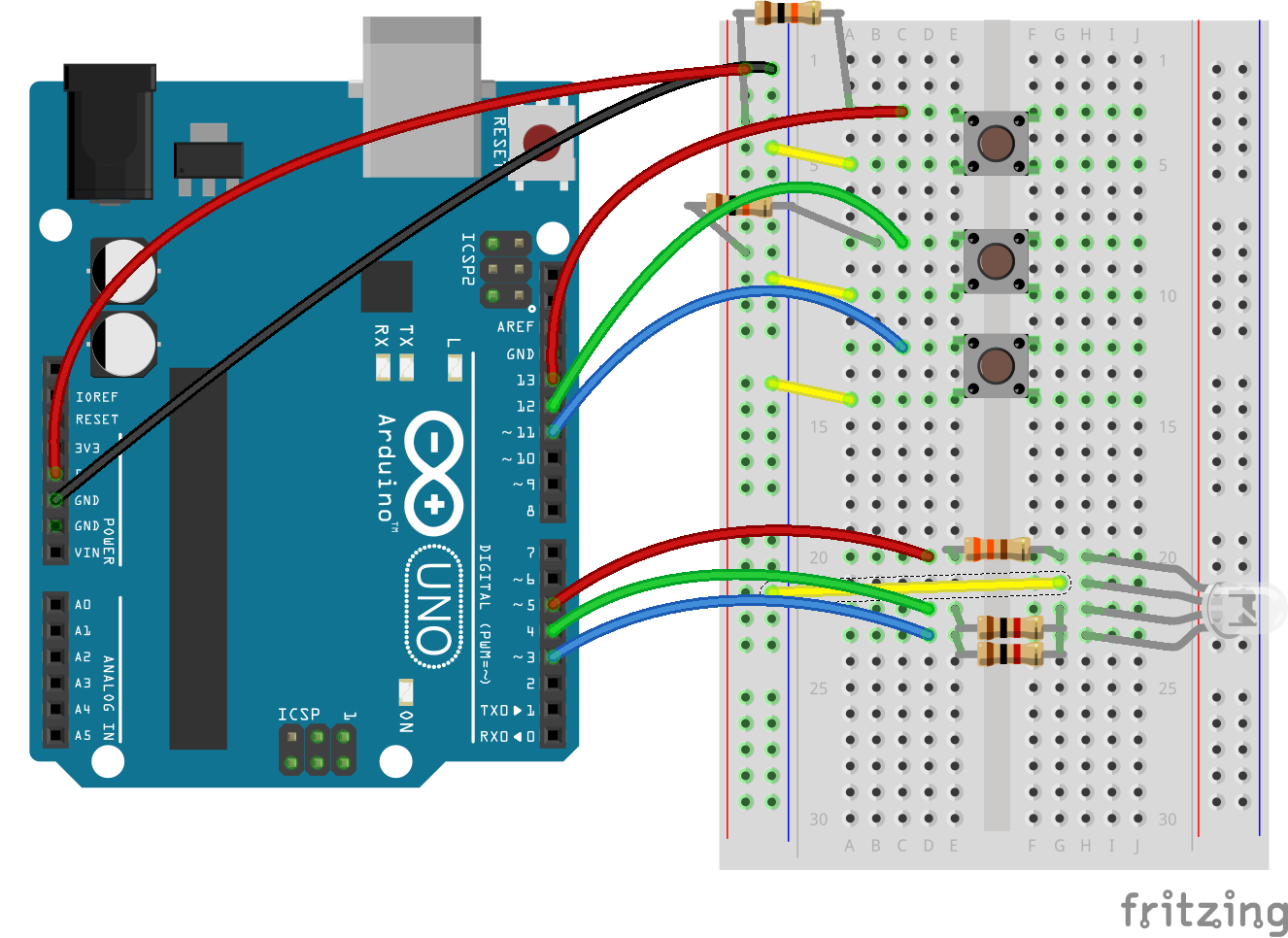
**Manual RGB LED Color Mixing**



Gusti S. Halima

ENGR 102 Intro to Electronics

Shoreline Community College

October 2, 2017

**Description of the problem and reason for building the circuit**

This circuit is based from the 3rd circuit of SparkFun’s SIK Guide. I chose this circuit because it would be interesting to see how the basic colors of an RGB LED will look like by manually mixing the colors.

**Things to do with the circuit**

There are three buttons on the circuit which individually represent the colors red, green, and blue from the RGB led. Pressing a single button will light a specific color but it is also possible to mix between two colors or all at once.

**Prior knowledge needed to create the circuit**

A basic understanding to create a circuit and write code for individual LEDs and assigning them to their own buttons are needed. It is also recommended to understand how different resistors will affect the output of a color from the RGB LED.

**Parts Required:**

1. Arduino Board
2. Breadboard
3. 6 long jumper wires
4. 4 small jumper wires
5. 3 buttons
6. 1 RGB LED
7. 2 1kΩ resistor
8. 1 330Ω resistor
9. 3 10kΩ resistor

**Problem-Solving Approach**

This circuit was simple to design which required no complicated planning. I created the circuit first then wrote the code.

**Problems:**

I had initially used the same 330Ω resistors for all three colors of the RGB led which resulted significantly brighter blue and green LEDs compared to the red LED. This problem was solved by changing the blue and green resistors to a 1kΩ resistor. However, it is also possible to use the same resistors for all three LEDs by setting the range values according to the RGB LED’s value spectrum which can be found online.

**Schematic**

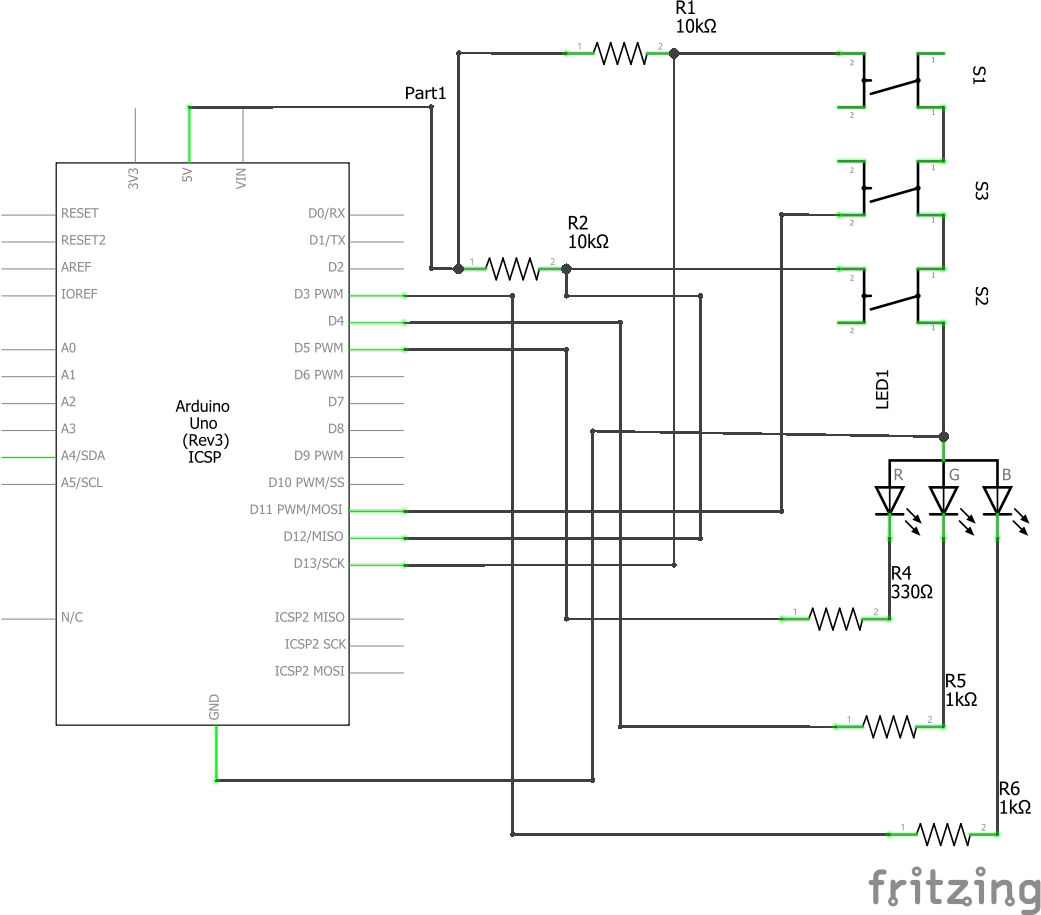
****

Figure . schematic of the circuit

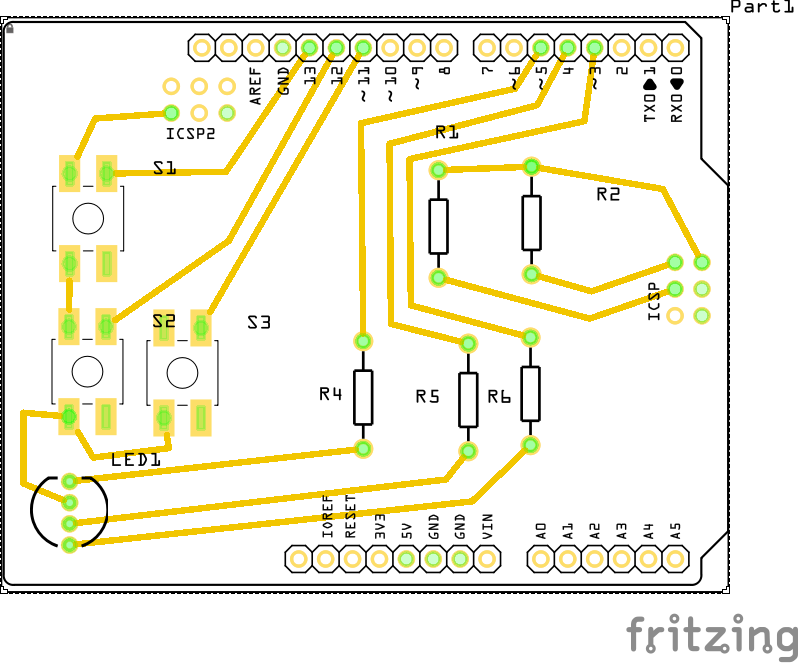
****

Figure . PCB of the circuit

**Printed Circuit Board**

Output Examples

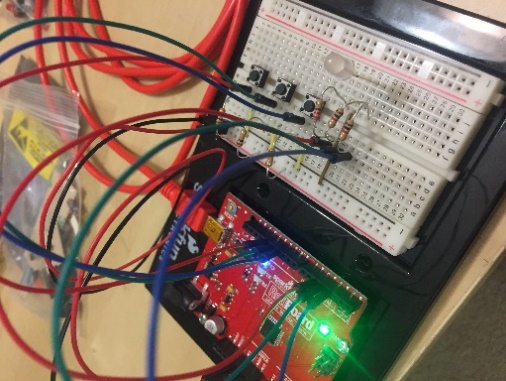
****

Figure . RGB LED turns off by default

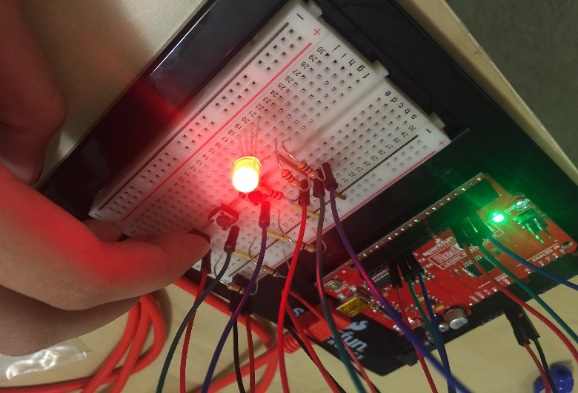
****

Figure . pressing the top button lights a red LED

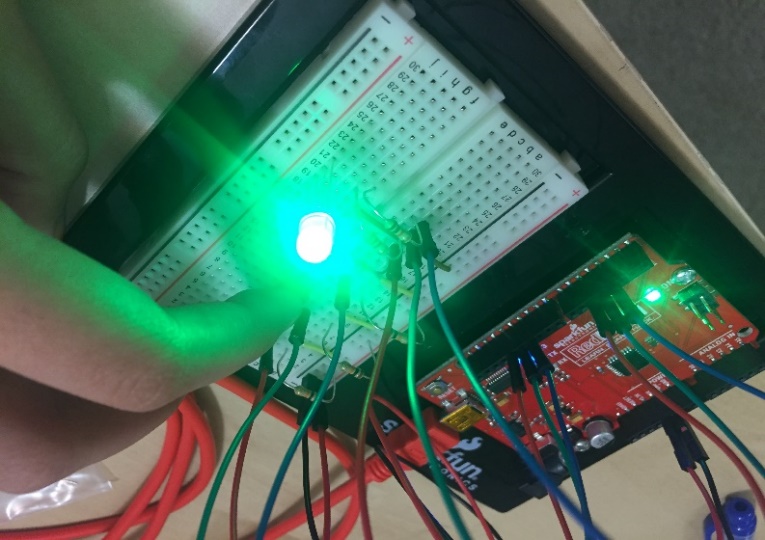
**­**

Figure . pressing the middle button lights the green LED

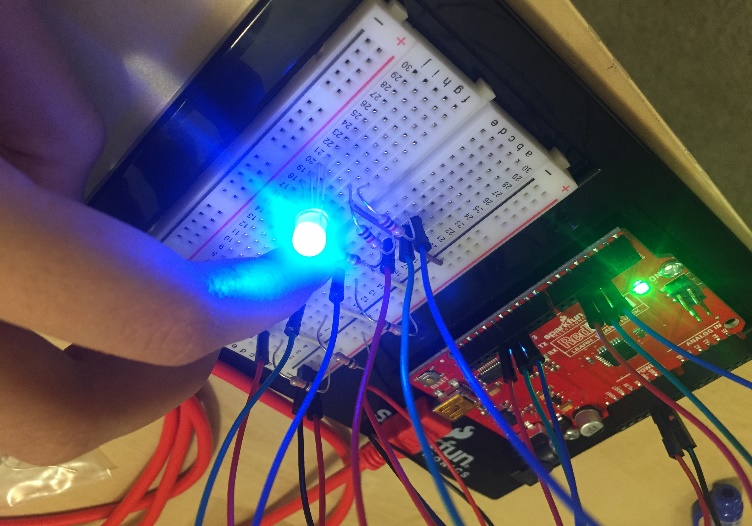
****

Figure . pressing the bottom button lights a blue LED

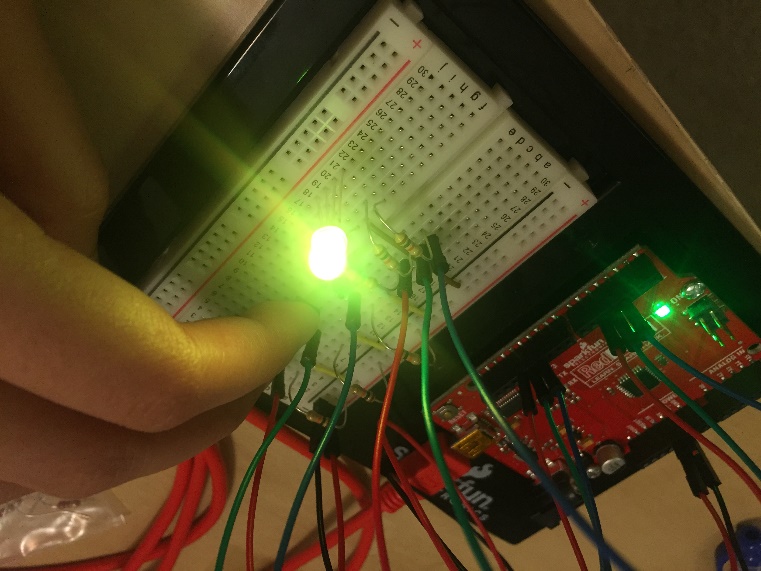


Figure . example of mixing two colors; red and green LEDs

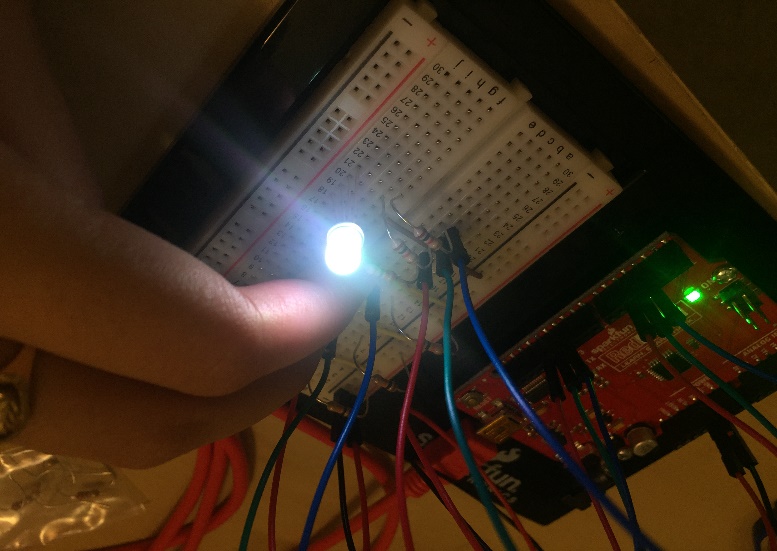
****

Figure . result of pressing all buttons simultaneously

**Code**

1. /\*
2. \* -Name: Gusti S. Halima
3. \* -class: ENGR 102, Fall 2017
4. \* -Project #1: Manual RGB LED Color Mixing
5. \* -What it does: The RGB LED main color LEDs (red, green, blue) are seperated
6. \* by individual buttons. Pressing the buttons simultaneously will mix the colors
7. \*/
9. //Creating arrays to store buttonPins and led pin values
10. **int** buttonPins[] = {11, 12, 13}; //blue, green, red
11. **int** led[] ={3, 4, 5}; //blue, green, red
13. //variables to check the state of the buttons assigned for each led
14. **int** redState, greenState, blueState;
16. **void** setup() {
17. //individually assigned pins
18. pinMode(buttonPins[0], INPUT);
19. pinMode(buttonPins[1], INPUT);
20. pinMode(buttonPins[2], INPUT);
21. pinMode(led[0], OUTPUT);
22. pinMode(led[1], OUTPUT);
23. pinMode(led[2], OUTPUT);
25. /\*
26. \* Creating a loop to assign the LED outputs
27. \* resulted a dim red LED, problem solved by
28. \* individually assigning the LEDs
29. \*/
30. }
32. **void** loop() {
33. blueState = digitalRead(buttonPins[0]);
34. greenState = digitalRead(buttonPins[1]);
35. redState = digitalRead(buttonPins[2]);
37. /\*
38. \* note that when a button's state is HIGH / 1 it actually turns
39. \* off the LED (buttons are inverted by default, search web for more
40. \* info). Either remeber this or change the buttons' pinMode from
41. \* output to INPUT\_PULLUP will turn LED on if buttons are HIGH / 1.
42. \*/
44. //if-else statement for default colors
45. **if** ((blueState == 0) && (greenState == 1) && (redState == 1)) {
46. lightLED(HIGH, LOW, LOW);
47. }
49. **else** **if** ((blueState == 1) && (greenState == 0) && (redState == 1)) {
50. lightLED(LOW, HIGH, LOW);
51. }
53. **else** **if** ((blueState == 1) && (greenState == 1) && (redState == 0)) {
54. lightLED(LOW, LOW, HIGH);
55. }
57. //if- else conditionals to mix the colors
58. /\*
59. \* Initially couldn't mix the colors inside the RGB LED from using a single 220 ohm
60. \* resistor connected to the GND. Fixed by giving each LED their own reistor.
61. \*/
62. **else** **if** (blueState == 0 && redState == 0 && greenState == 1) { //red + blue
63. lightLED(1, 0, 1);
64. }
66. **else** **if** (blueState == 1 && redState == 0 && greenState == 0) { //red + green
67. lightLED(0, 1, 1);
68. }
70. **else** **if** (blueState == 0 && redState == 1 && greenState == 0) { //green + blue
71. lightLED(1, 0, 1);
72. }
74. **else** **if** (blueState == 0 && redState == 0 && greenState == 0) { //all
75. lightLED(1, 1, 1);
76. }
78. //When all buttons are high (not pressed), the RGB LED is turned off by default
79. **else** {
80. lightLED(0, 0, 0);
81. }
82. }
84. /\*
85. the following method reduces the redundancy
86. of typing digitalWrite() for each combinations.
87. \*/
88. **void** lightLED(**int** b, **int** g, **int** r)
89. digitalWrite(led[0], b);
90. digitalWrite(led[1], g);
91. digitalWrite(led[2], r);
92. }

Video:<https://www.youtube.com/watch?v=Sz1xbOAvX4w>

Resource:<https://cdn.sparkfun.com/datasheets/Kits/SFE03-0012-SIK.Guide-300dpi-01.pdf>

Syntax Highlighter used:<http://www.planetb.ca/syntax-highlight-word>

Schematic created using Fritzing