Joshua Prier

# Lab Report 13: Free Lab

## Lab Report Rubric

| Category | Student Score | Grader Score |
|---|---|---|
| **Organization** | | |
| Appearance and formatting | 1/1 | /1 |
| Spelling, grammar, sentence structure | 1/1 | /1 |
| **Work** | | |
| Experimental procedure | 3/3 | /3 |
| Results (data, code, figure, graph, tables, etc.) | 3/3 | /3 |
| Conclusion | 1/2 | /2 |
| | | |
| **Total** | **9/10** | **/10** |

## Introduction

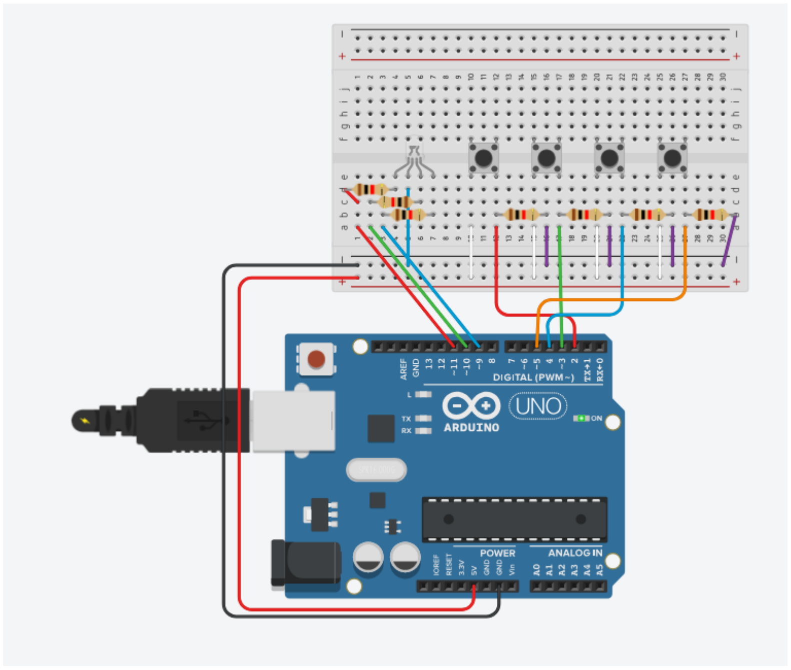
The objective of this lab was to create a simple interface for controlling the color output of an RGB LED using four pushbuttons. By incrementing the intensity of red, green, and blue channels and resetting the color values to zero with individual buttons.

## Procedure

1. To begin I did research on how to program and wire an RGB LED where I learned the **void setColor** command to be able to set up a function that helps to easily control the RGB LED
2. I connected the RGB LED to pins 13, 12, and  (red, green, and blue, respectively).
3. Then I connected four pushbuttons to pins 2, 3, 4, and 5 to control the values of red, green, blue and to clear values.
4. Next I defined pin configurations and variables to track the RGB values and button states. (e.g. **redButton = 2, redValue = 0, buttonstate1 = 0,  lastButtonState = 0)** some of which was copied from a previous lab (lab 7)
5. After that I added the **void setColor** function.
6. In the **void setup()** I configured the pins for the RGB LED as outputs and the pushbuttons as well as setting up the serial monitor
7. Implementing some code from a lab 7  I set **buttonState1 = digitalRead(redButton);** and did similar to the other button inputs.
8. Then using if statements I check to see if the current **buttonState** is High and that the **lastButtonState** is low of each individual button to determine when it is pressed. When this is true the value of the color will change by 17 according to its respective button. (**redValue =** (**redValue + 17) % 255**) or **redValue = 0; greenValue = 0; blueValue = 0; setColor(redValue, greenValue, blueValue);** for clear button
9. Then I made it so that the current color will be displayed after the button is pressed **Serial.print("Red Value: "); Serial.println(redValue);**
10. lastly I added the lines **lastButtonState1 = buttonState1;** from lab 7 so that the check for the if statements would function properly.
11. When testing the program, I had some issues with the colors from green and red values where colors would not appear until a value of 130 or greater was present, at first I thought it was a wiring issue so I moved the LED, resistors, and wires to different positions on the breadboard but that didn't have any effect on my results. Then I decided to check to see if the red and green color would work if connected to the blue output pin (11). Which did work. After looking at the output on the Arduino board I noticed that pin 11 had a ~ on it while pins 12, and 13 did not.
12. After this realization I changed the output in the program to use pins 9, 10, and 11.

RGB LED info

## Results



```
// Pins for the RGB LED
int redPin = 11;
int greenPin = 10;
int bluePin = 9;

// Button pins
int redButton = 7;
int greenButton = 3;
int blueButton = 4;
int clearButton = 5;

// Variables to track current color values
int redValue = 0;
int greenValue = 0;
int blueValue = 0;

int buttonState1 = 0;
int buttonState2 = 0;
int buttonState3 = 0;
int buttonState4 = 0;
int lastButtonState1 = 0;
int lastButtonState2 = 0;
int lastButtonState3 = 0;
int lastButtonState4 = 0;

// Function to set RGB color
void setColor(int redValue, int greenValue, int blueValue) {
  analogWrite(redPin, redValue);
  analogWrite(greenPin, greenValue);
  analogWrite(bluePin, blueValue);
}

void setup() {
  Serial.begin(9600);

  // Set LED pins as outputs
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);

  // Set button pins as inputs with pullup resistors
  pinMode(redButton, INPUT);
  pinMode(greenButton, INPUT);
  pinMode(blueButton, INPUT);
  pinMode(clearButton, INPUT);
}

void loop() {
  buttonState1 = digitalRead(redButton);
  buttonState2 = digitalRead(greenButton);
  buttonState3 = digitalRead(blueButton);
  buttonState4 = digitalRead(clearButton);
  // Increment red value
  if (buttonState1 == HIGH && lastButtonState1 == LOW) {
    redValue = (redValue + 17) % 255; // Increment by 17, loop back at 255
    setColor(redValue, greenValue, blueValue);
    Serial.print("Red Value: ");
    Serial.println(redValue);

  }

  // Increment green value
  if (buttonState2 == HIGH && lastButtonState2 == LOW) {
    greenValue = (greenValue + 17) % 255; // Increment by 17, loop back at 255
    setColor(redValue, greenValue, blueValue);
    Serial.print("Green Value: ");
    Serial.println(greenValue);

  }

  // Increment blue value
  if (buttonState3 == HIGH && lastButtonState3 == LOW) {
    blueValue = (blueValue + 17) % 255; // Increment by 17, loop back at 255
    setColor(redValue, greenValue, blueValue);
    Serial.print("Blue Value: ");
    Serial.println(blueValue);

  }
  // Clear all values
  if (buttonState4 == HIGH && lastButtonState4 == LOW) {
    redValue = 0;
    greenValue = 0;
    blueValue = 0;
    setColor(redValue, greenValue, blueValue);
    Serial.println("Colors Cleared");

  }
  lastButtonState1 = buttonState1;
  lastButtonState2 = buttonState2;
  lastButtonState3 = buttonState3;
  lastButtonState4 = buttonState4;
}
```

## Conclusions and Reflection

In conclusion, this lab helped me learn how to control an RGB LED using pushbuttons and how to solve problems along the way. I programmed the buttons to adjust the red, green, and blue values or reset them, and I used the setColor function to make it easier to control the LED. When I had issues with the red and green colors not working properly, I tested the wiring and figured out that the problem was with the output pins I chose. By switching to pins 9, 10, and 11, which support PWM, I was able to fix the issue and get everything working. This lab showed me how important it is to test and adjust both the hardware and the code when building a project.