ECE296 Lab 7 - Arduino Color Detector

Chase A. Lotito, *SIUC Undergraduate*

I. Introduction

This experiment is the next step after the night light lab. Now, we replace the monochromatic LED with a RGB LED (Red-Green-Blue). The RGB LED will flash its colors against an object with a color that we wish to detect, and the photoresistor will see how much red, green, and blue the object reflects. Based on the intensities of reflection, we can set ranges for various colors–which let us tune the color detector accordingly.

II. Assessment of Design

Figure 1 shows the physical implementation of the color detector.

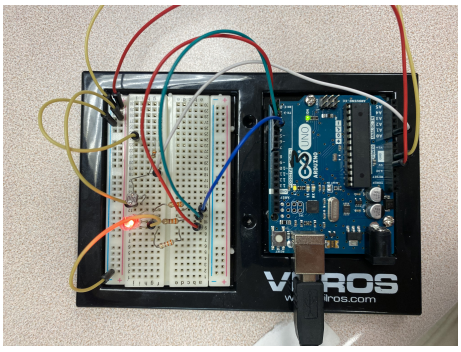

Fig. 1. Color Detector Design

The circuit itself was difficult to tune; the sample colors were shown off of a phone screen, which caused volatility in brightness. However, the circuit was able to detect all 12 colors.

The Arduino code flashes the LED red, then green, then blue, repeatedly in 100ms increments. Each individual pulse, we read the voltage between the photoresistor and $10k\Omega$-resistor, which via voltage-division should be between 0 and 5V. The *analogRead()* function will translate this as an integer value between 0 and 1023.

I personally kept this range intact, instead of using *map()*, and this gave me a larger range to work with. The RGB values between successive colors tended to be small, so having a large range allowed more room for tolerances.

Finally, we collect the current RGB values detected from the photoresistor and send them through the *if-else* conditional logic to find the correct color. The ranges for these can be referenced in the code in *Appendix B*. The color detection is displayed on the Serial Monitor, and those can be seen in Figure 2.

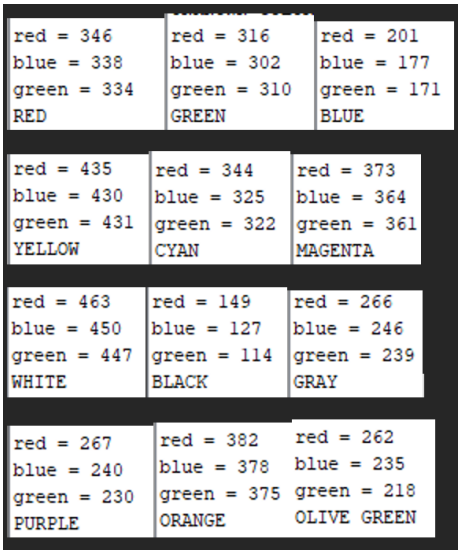

Fig. 2. All colors detected by circuit

III. Conclusion

Overall, this lab displayed the power of using solid-state devices with the Arduino to create useful sensors. Color detection seems like an extremely difficult task that would require AI image models, but all you need are simple passive components.

APPENDIX A: HARDWARE SCHEMATIC

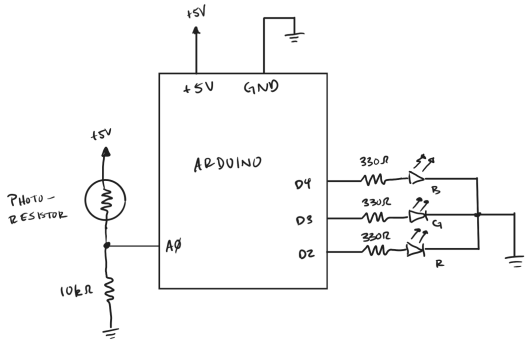

Fig. 3. Color Detector circuit diagram

APPENDIX B: CODE FOR THE SOFTWARE DEVELOPED

```
int red, blue, green;
int delay_time = 100;
int sensor;

void setup() {
    pinMode(2, OUTPUT); // Red
    pinMode(3, OUTPUT); // Blue
    pinMode(4, OUTPUT); // Green
    Serial.begin(9600);
}

void loop() {
    digitalWrite(2,HIGH);
    digitalWrite(3,LOW);
    digitalWrite(4,LOW);
    delay(delay_time);
    red = analogRead(A0);
    red = map(red, 230, 670, 0, 255);

    digitalWrite(2,LOW);
    digitalWrite(3,HIGH);
    digitalWrite(4,LOW);
    delay(delay_time);
    blue = analogRead(A0);
    blue = map(blue, 340, 675, 0, 255)
        ;

    digitalWrite(2,LOW);
    digitalWrite(3,LOW);
    digitalWrite(4,HIGH);
    delay(delay_time);
    green = analogRead(A0);
    green = map(green, 240, 666, 0,
        255);

    Serial.print("red = ");
    Serial.println(red);
    Serial.print("blue = ");
    Serial.println(blue);
    Serial.print("green = ");
    Serial.println(green);

    if (red > 175 && blue < 176 &&
        green > 168){
        Serial.println("RED");
    }
    else if (red < 183 && blue > 170
        && green < 180){
        Serial.println("GREEN");
    }
    else if (red > 180 && blue > 205
        && green < 182){
        Serial.println("CYAN");
    }
    else if (red < 200 && blue < 186
        && green > 182){
        Serial.println("MAGENTA");
    }
    else if (red < 85 && blue < 85 &&
        green > 35){
        Serial.println("BLUE");
    }
    else if (red < 245 && blue < 240
        && green > 229) {
        Serial.println("YELLOW");
    }
    else if (red < 30 && blue < 2 &&
        green < 0) {
        Serial.println("BLACK");
    }
    else if (red > 240 && blue > 240
        && green < 255) {
        Serial.println("WHITE");
    }
    else {
        Serial.println("UNKNOWN COLOR
            ");
    }
}
```