Drexel University

To: Dr. Peters

From: Tyler Ostinato

Date: October 11, 2021

Re: ECE 303-Lab 3-Pulse Width Modulation

Purpose

Learn how to dim an LED using pulse width modulation on the Arduino Mega 2560. Be able to Change resistivity of a photocell using an LED and calculate its associated resistivity at varying intensities of light.

Discussion

Digital Pin Operation | 1.3

In this lab we will be making two circuits. Circuit one will be a simple LED circuit wired to a 1kΩ resistor. The second circuit will use a photocell in series with a 10kΩ resistor. The goal will be to change the resistivity of the photocell by using PWM to control intensity of an LED (Figure 1).

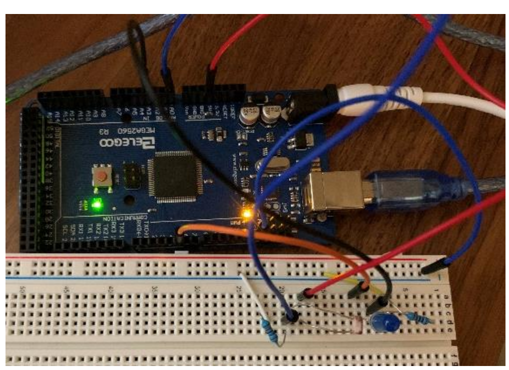


Figure 1: Set up of Circuits 1 and 2.

Digital Pin Operation | 1.4

In this section of the lab I used PWM to control the gradual intensity of the LED in the first circuit. PWM allows us to use the timers on the Arduino to control how long a signal is HIGH or LOW for a given pulse, this is referred to as the "duty cycle" (Figure 2). By doing this the LED will blink on and off at different rates, this has the affect of changing the intensity of the LED to the human eye.

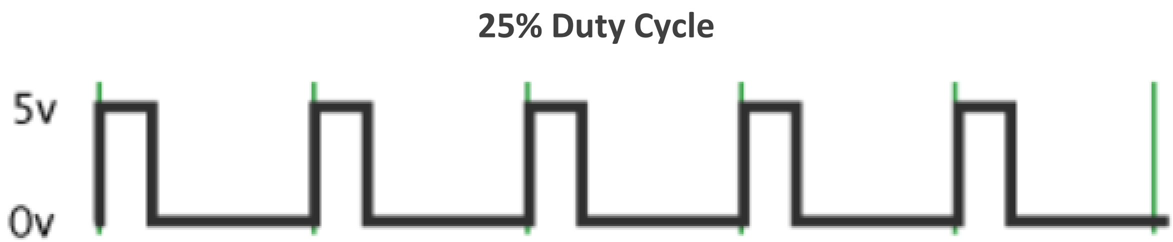


Figure 2: Example of a duty cycle

To increase the LED brightness in increments of 5% we can use Duty Cycle = OCR4A / ICR4. This allows the user to configure how long the pulse is high (using OCR4A) and how long the entire pulse is (using ICR4) (Figure 3).

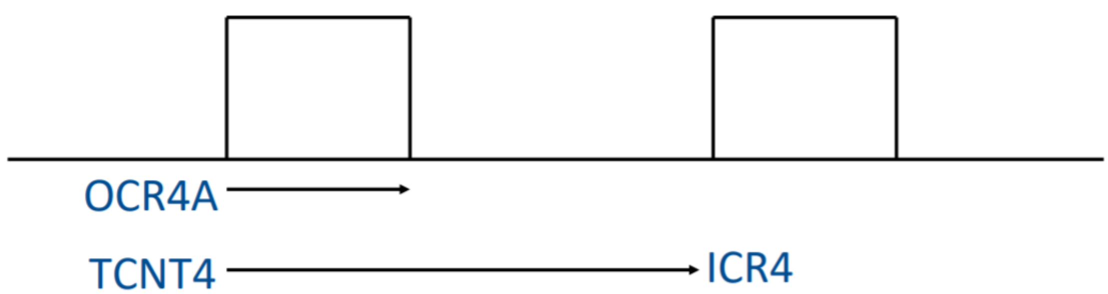


Figure 3: Using OCR4A and ICR4 to control the duty cycle of Arduino Mega 2560.

After setting the duty cycle the LED will now increase intensity in increments of 5%. To see the effects this has on the circuit we need to do some calculations. To find the voltage across the resistor in circuit one, we can use analogRead() to take the input from circuit one and find the voltage drop. While doing this I ran into an issue where the analog input would return 0 most of the time. I believe this was due to the analog reading not getting the PWM signal while it was in HIGH mode. To fix this problem I took 100 samples of the analog input and took the average to get the resulting voltage drop across the resistors (Figure 4).

```
while (loop_count < 100) {
  int analog_led = analogRead(A3);
  int analog_photo = analogRead(A0);
  if(analog_led!=0){
    sum_led += analog_led;
    div_count_led++;
  }
  if(analog_photo!=0){
    sum_photo += analog_photo;
    div_count_photo++;
  }
  loop_count++;
}
```

Figure 4: Code to capture 100 samples of the analog reading to get an accurate guess for voltage drop across the resistors in circuits 1 and 2.

After getting the analog readings they need to be read into a human readable format to properly display the voltage across the respective resistor. The analogRead() function will take in a digital signal a return a number from 0-1023. To get this to a voltage reading we can use analog reading * target voltage / 1023 (Figure 5).

```
float voltage_of_led = (sum_led/div_count_led) * (5.0 / 1023.0);
Serial.print("Voltage of R in LED circuit: ");
Serial.print(voltage_of_led);
Serial.print("\n");
```

Figure 5: Code to turn analog reading into human readable voltage value.

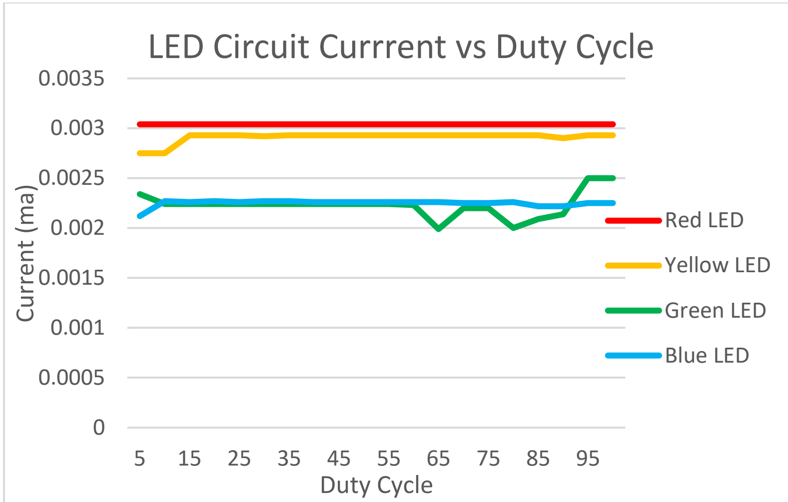
Digital Pin Operation | 1.5

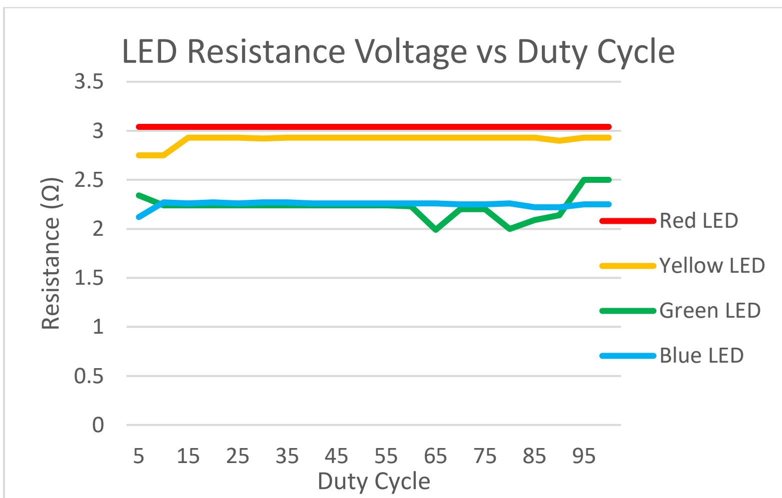
Using these values we will be able to calculate the resistance in the photocell at respective intensity values and differing LEDS. Differing LEDs will have an effective on the photoresistor due to their respective intensity as well as their forwarding voltage values.

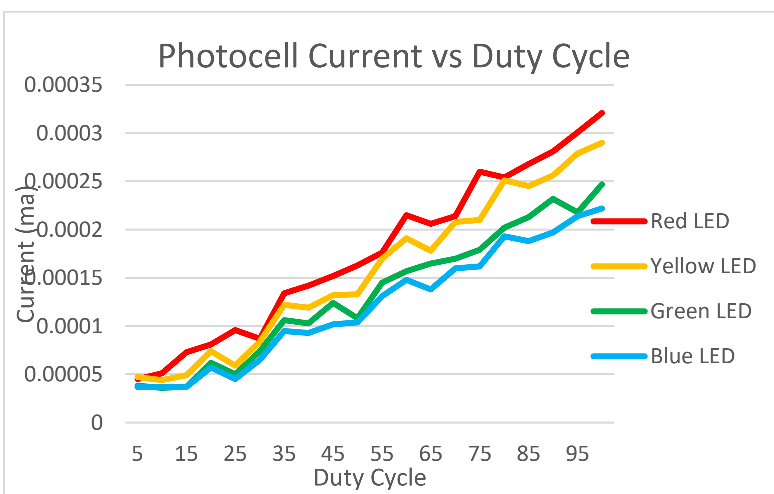
Conclusion

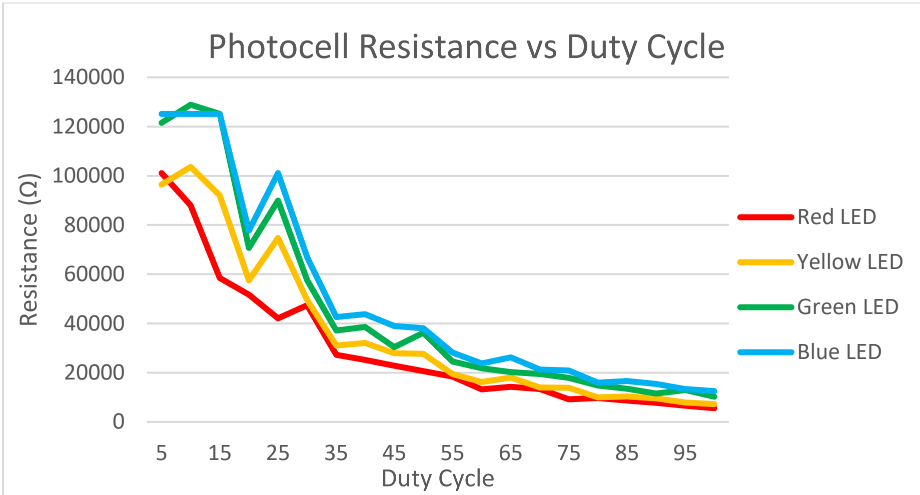
In this lab we have learned how to use PWM to manipulate the intensity of an LED and how that LED can change the resistivity in a photocell. We also learned how to use analogRead() to take in a digital value and turn it into an analog value to measure voltage across a resistor.

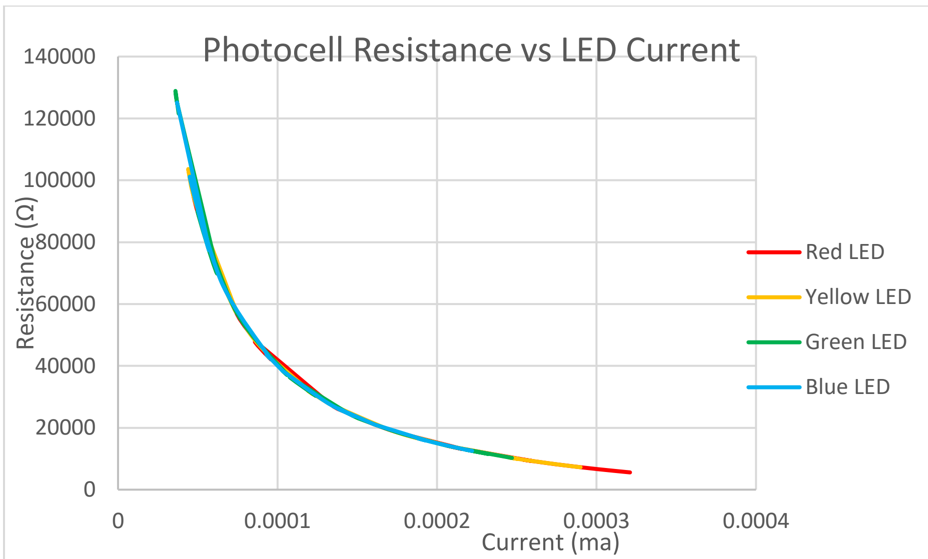
Appendix

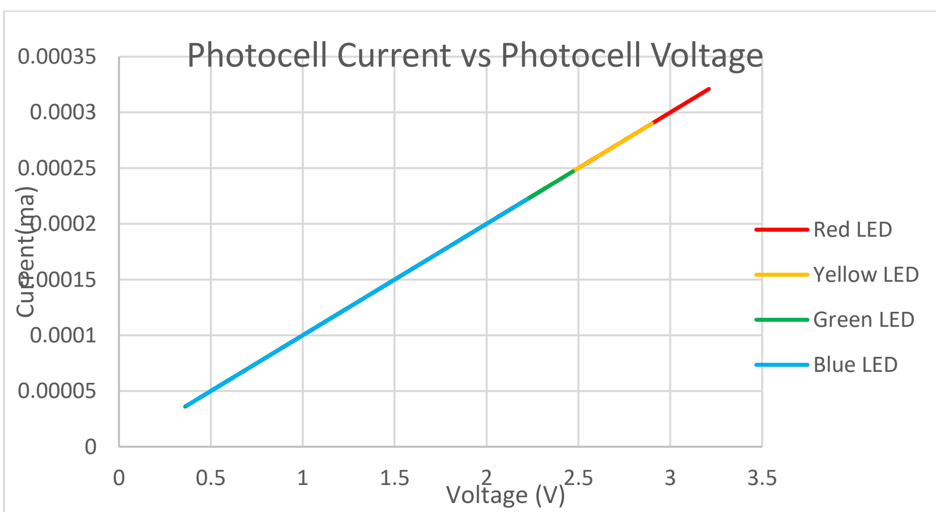

LED Circuit Currrent vs Duty Cycle
0.0035
0.003
0.0025
0.002
0.0015
0.001
0.0005
0
Current (ma)
5
15
25
35
45
55
65
75
85
95
Duty Cycle
Red LED
Yellow LED
Green LED
Blue LED


LED Resistance Voltage vs Duty Cycle
3.5
3
2.5
2
1.5
1
0.5
0
Resistance (Ω)
5
15
25
35
45
55
65
75
85
95
Duty Cycle
Red LED
Yellow LED
Green LED
Blue LED


Photocell Current vs Duty Cycle
0.00035
0.0003
0.00025
0.0002
0.00015
0.0001
0.00005
0
Current (ma)
5
15
25
35
45
55
65
75
85
95
Duty Cycle
Red LED
Yellow LED
Green LED
Blue LED


Photocell Resistance vs Duty Cycle
140000
120000
100000
80000
60000
40000
20000
0
Resistance (Ω)
5
15
25
35
45
55
65
75
85
95
Duty Cycle
Red LED
Yellow LED
Green LED
Blue LED


Photocell Resistance vs LED Current
140000
120000
100000
80000
60000
40000
20000
0
Resistance (Ω)
0
0.0001
0.0002
0.0003
0.0004
Current (ma)
Red LED
Yellow LED
Green LED
Blue LED


Photocell Current vs Photocell Voltage
0.00035
0.0003
0.00025
0.0002
0.00015
0.0001
0.00005
0
Current (ma)
0
0.5
1
1.5
2
2.5
3
3.5
Voltage (V)
Red LED
Yellow LED
Green LED
Blue LED

```
const int VOL_PIN = A0;

void setup() {
  // put your setup code here, to run once:
  TCCR3A=0b10000010;
  TCCR3B=0b00010001; // Pre-scaler of 1
  ICR3=12500;
  OCR3A=625; // Duty cycle is 5%
  TCNT3=0;
  pinMode(5,OUTPUT);
  Serial.begin(9600);
}


void loop() {

    delay(1000);

    for(int i=1;i<=20;i++){

      // Increase duty cycle by 5%
      Serial.print("Duty cycle -> ");
      Serial.print(5*i);
      Serial.print("%");
      Serial.print("\n");

      long int sum_led=0;
      long int sum_photo=0;
      int loop_count=0;
      int div_count_led=0;
      int div_count_photo=0;

      while (loop_count < 100) {
```

```
      int analog_led = analogRead(A3);
      int analog_photo = analogRead(A0);
      if(analog_led!=0){
        sum_led += analog_led;
        div_count_led++;
      }
      if(analog_photo!=0){
        sum_photo += analog_photo;
        div_count_photo++;
      }
      loop_count++;
    }

    float voltage_of_led = (sum_led/div_count_led) * (5.0 / 1023.0);
    Serial.print("Voltage of R in LED circuit: ");
    Serial.print(voltage_of_led);
    Serial.print("\n");

    float voltage_of_photo = (sum_photo/div_count_photo) * (5.0 / 1023.0);
    Serial.print("Voltage of R in sensor circuit: ");
    Serial.print(voltage_of_photo);
    Serial.print("\n");

    delay(2000);

    OCR3A+=625; // Increase duty cycle by 5%
  }
  Serial.print("\n");

  Serial.print("\nDone.");
  delay(1000);
  exit(0);
}
```