ECE 303 Lab Technical Memos

Nicholas Sica

October 2, 2020

Contents

1 Lab 2

1.1 Discussion

The point of this lab was to introduce us to the Arduino toolkit and allow anyone who is new to the platform time to adjust and get familiar with the tools presented to them. The lab was straightforward, connect an LED and resistor to a pin on the arduino and writing simplistic code to change the pulse width modulation values. The output is as expected, with the intensity of the light changing based off the number put into the serial monitor, zero being off and 255 being full brightness.

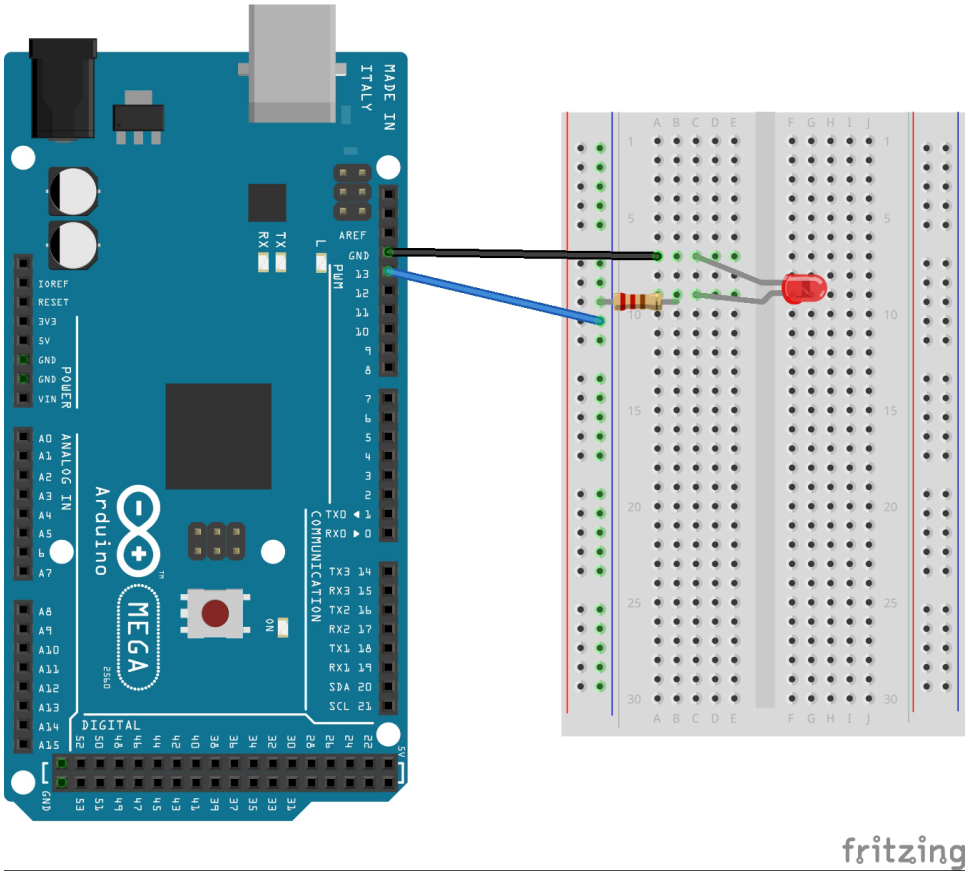

Figure 1: Basic LED Circuit Setup

2 Lab 3

2.1 Discussion

This lab was used to get us acquainted with timers and learn not only how to set up the correct bit values, but also how to use them efficiently. A lot of the time was spent debugging the bit values that the different masks are initialized with as well as learning how to turn the interrupts off efficiently. When run, every LED blinks at a slow pace and each guess for the code either causes it to blink faster, if wrong, or turn off, if correct. When the user is out of attempts, the remaining LEDs stay permanently on.

$$OCR3A = \frac{16 \times 10^6}{p \times f} - 1 \tag{1}$$

The value of the register was found using equation 1, where p is the prescalar, in this case 1024, and f is the target frequency, in this case around 0.5 hertz or a 2 second period. Every subsequent wrong guess, it was divided by two to make it go faster.

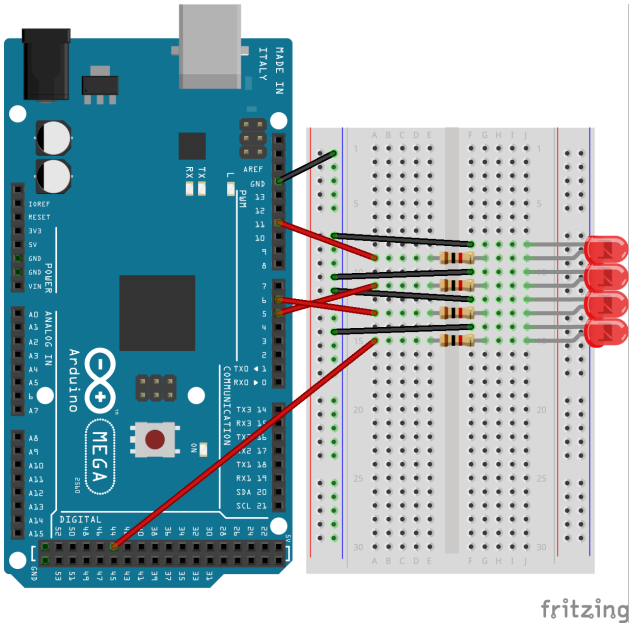

Figure 2: Codebreaker Circuit Setup

2.2 Hardware

Figure 2 shows the setup of the circuit. The setup of the hardware was very similar to the previous lab, but this time there are four LEDs and care was taken to connect to them to the correct pin corresponding to the timer we want to use for it.

3 Lab 4

3.1 Discussion

This lab was used to get us to get more comfortable with timers in a different way as well as introduce us to a photocell. This time around not a lot of time was spent getting the correct timer values, we just had to make sure that they represented the correct duty cycle. The results for this lab were a bit worse due to lights coming from my computer. I covered it with a box, but there was bound to be some leakage. It would have been better if I could isolate my circuit in a container and have my photocell the perfect distance from the LED. After copying all the data to excel the voltage divider equation shown in Equation 2 was used to get the resistances of the photocell and led. V_{DD} is the source voltage, in this case 5V, R_{gnd} is the resistor leading to ground, 10k in the photocell circuit and 1k in the led circuit, V_{out} is the voltage read from pin A0 for the led and pin A1 for the photocell.

$$R = \frac{V_{DD} \times R_{gnd}}{V_{out}} - R_{gnd} \tag{2}$$

The voltage of the LED and photocell were found by using Kirchoff's Volt-

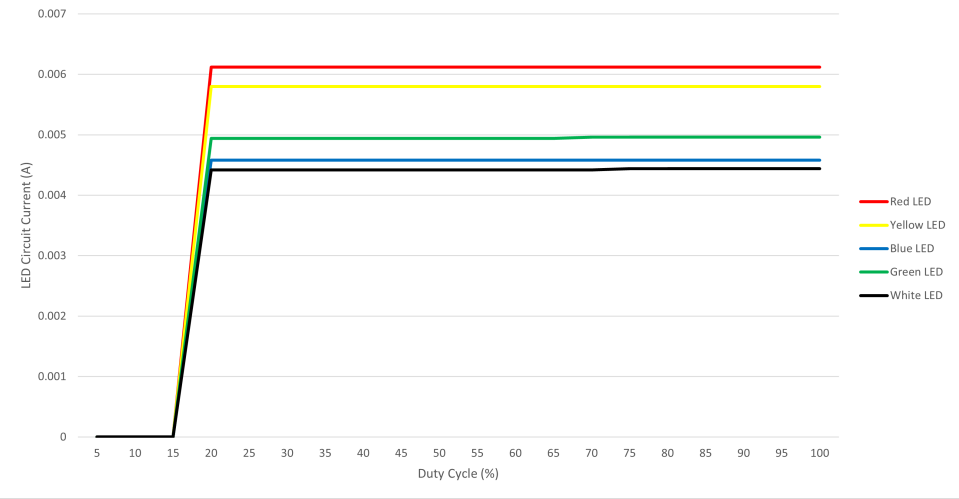

Figure 3: Duty Cycle Versus LED Circuit Current

age law and subtracting the voltage across the respective resistor from 5V.

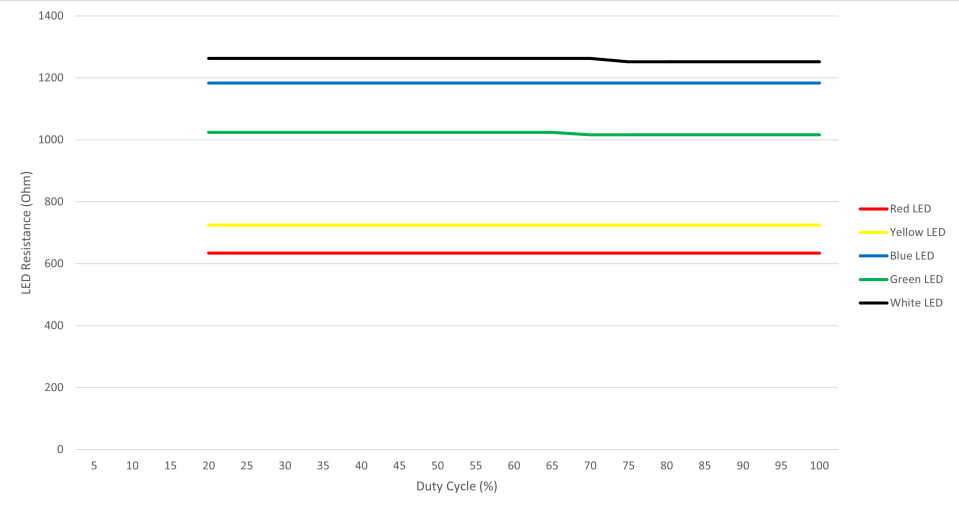

Figure 4: Duty Cycle Versus LED Resistance

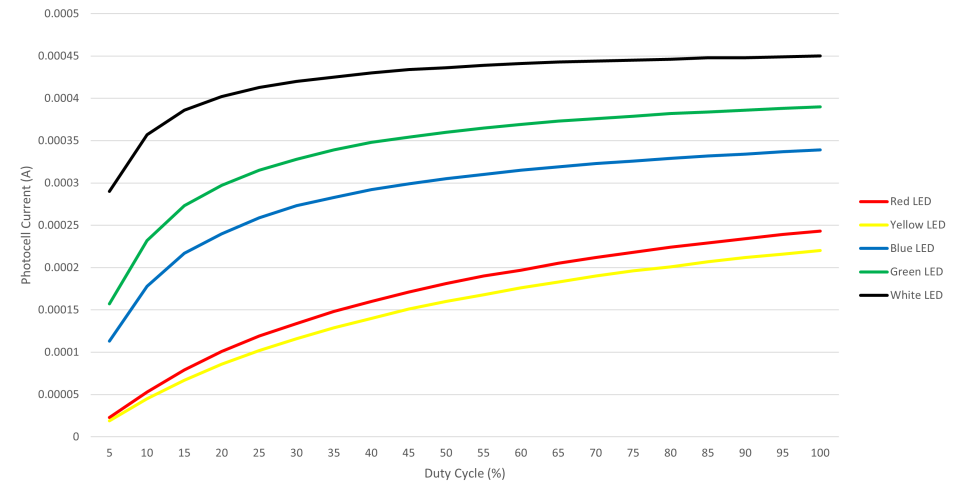

Figure 5: Duty Cycle Versus Photocell Current

Lastly, the currents were found using the previous values found for voltage and resistance and applying Ohm's Law. An error in the values read from pin A0 was introduced due to the fact that we were reading it from a PWM source. Figure 3 shows the duty cycle versus LED circuit current which is what you'd expect, a constant value after it gets up and running. Figure 4

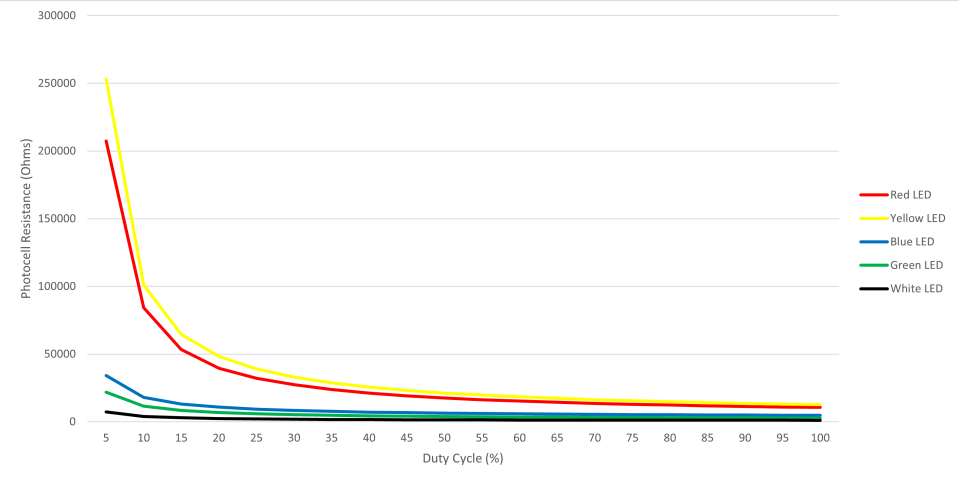

Figure 6: Duty Cycle Versus Photocell Resistance

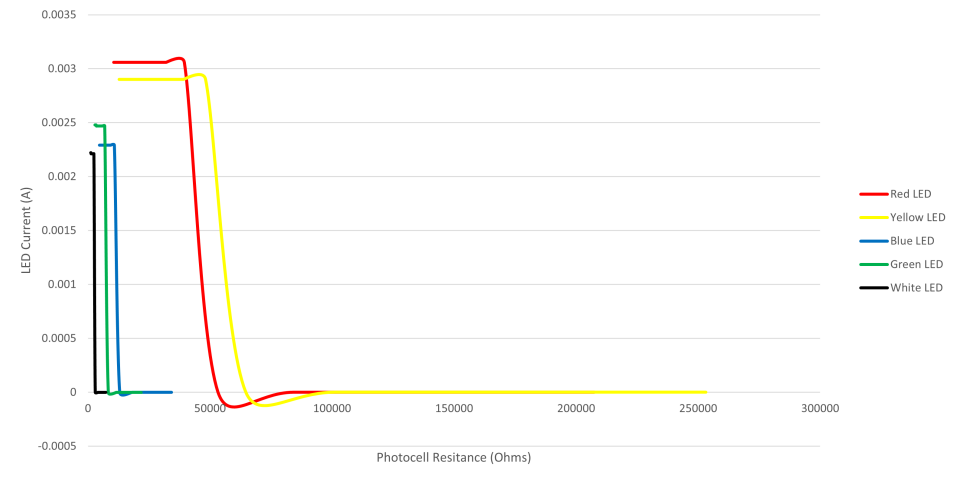

Figure 7: Photocell Resistance Versus LED Current

shows the duty cycle versus the LED resistance which is a constant value due to the voltage applied not changing. Figure 5 shows duty cycle versus photocell current which is increasing due to the resistance in the photocell decreasing as the LED strength increases. Figure 6 shows duty cycle versus photocell resistance which is decreasing due to the LED growing in strength. Finally, Figure 7 which shows photocell resistance versus LED current which is the most interesting because it is fairly constant until it suddenly dips.

3.2 Hardware

Figure 8 shows the setup of the circuit. The setup of the hardware was very simple, but could've been improved with a housing that blocks out light. You could also get different results depending on the distance the photocell was from the LED, which could lead to skewed results when swapping the LED out for another color.

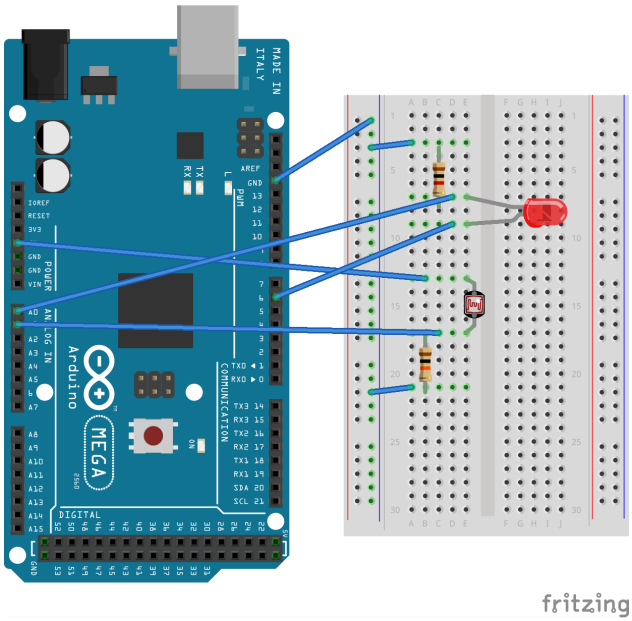

Figure 8: Photocell and LED Circuit Setup

Appendices

A Program Code

A.1 Lab 2 Code

```
1  #include <Arduino.h>
2
3  int led = 13;
4  int intensity = 0;
5
6  void setup()
7  {
8      pinMode(led, OUTPUT);
9      analogWrite(led, LOW);
10     Serial.begin(9600);
11     Serial.print("Please enter a number from 0 to 255: ");
12 }
13
14 void loop()
15 {
16     if(Serial.available() > 0)
17     {
18         intensity = Serial.parseInt();
19         Serial.print("\nGot number: ");
20         Serial.println(intensity, DEC);
21         analogWrite(led, intensity);
22         Serial.print("Please enter a number from 0 to 255: ")
               ;
23     }
24 }
```

Listing 1: Lab 2 Code

A.2 Lab 3 Code

```
1  #include <Arduino.h>
2
3  const int leds[] = {11, 5, 6, 44};
4  long password;
5  uint8_t correct_nums = 0b0000;
6  uint8_t num_tries = 0;
```

```
7  // = (16 * 10^6) / (1024 * 0.5) - 1
8  uint16_t starting_freq = 31248;
9  bool locked_out = false;
10
11 void setup()
12 {
13     noInterrupts();
14     for(int i = 0; i < 4; i++)
15     {
16         pinMode(leds[i], OUTPUT);
17         digitalWrite(leds[i], LOW);
18     }
19
20     // Setup timer 1 pin 11 channel A
21     TCCR1A = 0;
22     TCCR1B = 0;
23     TIMSK1 = 0;
24     TCNT1  = 0;
25     OCR1A  = starting_freq;
26     TCCR1B |= (1 << WGM12);
27     // 1024 prescalar
28     TCCR1B |= (1 << CS12) | (0 << CS11) | (1 << CS10);
29     TIMSK1 |= (1 << OCIE1A);
30
31     // Setup timer 3 pin 5 channel A
32     TCCR3A = 0;
33     TCCR3B = 0;
34     TIMSK3 = 0;
35     TCNT3  = 0;
36     OCR3A  = starting_freq;
37     TCCR3B |= (1 << WGM32);
38     TCCR3B |= (1 << CS32) | (0 << CS31) | (1 << CS30);
39     TIMSK3 |= (1 << OCIE3A);
40
41     // Setup timer 4 pin 6 channel A
42     TCCR4A = 0;
43     TCCR4B = 0;
44     TIMSK4 = 0;
45     TCNT4  = 0;
46     OCR4A  = starting_freq;
47     TCCR4B |= (1 << WGM42);
48     TCCR4B |= (1 << CS42) | (0 << CS41) | (1 << CS40);
49     TIMSK4 |= (1 << OCIE4A);
50
51     // Setup timer 5 pin 44 channel A
```

```
52      TCCR5A  = 0;
53      TCCR5B  = 0;
54      TIMSK5  = 0;
55      TCNT5   = 0;
56      OCR5A   = starting_freq;
57      TCCR5B |= (1 << WGM52);
58      TCCR5B |= (1 << CS52) | (0 << CS51) | (1 << CS50);
59      TIMSK5 |= (1 << OCIE5A);
60
61      Serial.begin(9600);
62      randomSeed(analogRead(0));
63      password = random(10000);
64      Serial.print("Password is ");
65      Serial.println(password, DEC);
66      Serial.println("Please enter guess:");
67      interrupts();
68  }
69
70  void loop()
71  {
72      if(num_tries < 5)
73      {
74          if(Serial.available() > 0)
75          {
76              int guess = Serial.parseInt();
77              Serial.print("Guess is ");
78              Serial.println(guess, DEC);
79              int temp_password = password;
80              for(int i = 0; i <= 3; ++i)
81              {
82                  if(guess % 10 == temp_password % 10)
83                  {
84                      correct_nums |= 1 << i;
85                      switch(i)
86                      {
87                          case 0: TIMSK1 = 0;
88                          case 1: TIMSK3 = 0;
89                          case 2: TIMSK4 = 0;
90                          case 3: TIMSK5 = 0;
91                      }
92                      digitalWrite(leds[i], LOW);
93                  }
94                  guess = guess / 10;
95                  temp_password = temp_password / 10;
96              }
```

```
97
98              TCNT1 = 0;
99              TCNT3 = 0;
100             TCNT4 = 0;
101             TCNT5 = 0;
102             OCR1A = OCR1A / 2;
103             OCR3A = OCR3A / 2;
104             OCR4A = OCR4A / 2;
105             OCR5A = OCR5A / 2;
106             if(num_tries < 4)
107                 Serial.println("Please enter guess: ");
108             num_tries++;
109         }
110     }
111     else if(!locked_out)
112     {
113         TIMSK1 = 0;
114         TIMSK3 = 0;
115         TIMSK4 = 0;
116         TIMSK5 = 0;
117         Serial.println("Out of tries!");
118         for(int i = 0; i < 4; i++)
119         {
120             if((correct_nums & (1 << i)) == 0b0000)
121                 digitalWrite(leds[i], HIGH);
122         }
123             locked_out = true;
124     }
125
126 }
127
128 ISR(TIMER1_COMPA_vect)
129 {
130     digitalWrite(leds[0], !digitalRead(leds[0]));
131 }
132
133 ISR(TIMER3_COMPA_vect)
134 {
135     digitalWrite(leds[1], !digitalRead(leds[1]));
136 }
137
138 ISR(TIMER4_COMPA_vect)
139 {
140     digitalWrite(leds[2], !digitalRead(leds[2]));
141 }
```

```
142
143 ISR(TIMER5_COMPA_vect)
144 {
145     digitalWrite(leds[3], !digitalRead(leds[3]));
146 }
```

Listing 2: Lab 3 Code

A.3 Lab 4 Code

```
1  #include <Arduino.h>
2
3  int led = 6;
4
5  void setup()
6  {
7      pinMode(led, OUTPUT);
8      pinMode(A0, INPUT);
9      pinMode(A1, INPUT);
10
11     noInterrupts();
12     // Setup fast PWM timer 4 channel A pin 6
13     TCCR4A = 0;
14     TCCR4B = 0;
15     TIMSK4 = 0;
16     TCNT4  = 0;
17     ICR4   = 12500;
18     OCR4A  = 625;
19     TCCR4A |= (1 << WGM41);
20     TCCR4A |= (1 << COM4A1);
21     TCCR4B |= (1 << WGM43);
22     TCCR4B |= (0 << CS41) | (1 << CS40);
23
24     Serial.begin(9600);
25     interrupts();
26 }
27
28 void loop()
29 {
30     if(OCR4A <= 12500)
31     {
32         delay(2000);
33         Serial.println("Duty Cycle, LED Resistor Voltage,
               Photocell Resistor Voltage");
34
```

```
35          int duty_cycle = (int)((float)OCR4A / (float)ICR4 *
                100);
36          Serial.println(duty_cycle);
37
38          int led_resistor_value = analogRead(A0);
39          float led_resistor_voltage = (float)
                led_resistor_value * (5.0 / 1023.0);
40          Serial.println(led_resistor_voltage);
41
42          int photo_resistor_value = analogRead(A1);
43          float photo_resistor_voltage = (float)
                photo_resistor_value * (5.0 / 1023.0);
44          Serial.println(photo_resistor_voltage);
45          Serial.println();
46          OCR4A += 625;
47      }
48  }
```

Listing 3: Lab 4 Code