# PoE Lab 1: Building a Bike Light

Liani Lye, Abigail Rodrigues

Due 09.16.14

## 1 Introduction

The objective of this lab was to construct a "bike light" consisting of three LEDs that, upon a button push, switched between four states: all on, all off, all flashing, and bouncing. We then added a potentiometer to vary the speed at which the LEDs would flash and bounce.

## 2 Circuit Analysis

### 2.1 Known Values

From the LED datasheet, the following attributes were defined:

$$\begin{aligned} Voltage_{max} &= 2.5V \\ Power_{dissipated} &= 75mW \end{aligned}$$

The Arduino sets $Voltage_{source}$ as 5V. To stay within LED's voltage rating, we set the voltage drops to be 2V and 3V across the LED and resistor, respectively.

Since all the LED/resistor pairs are in parallel (see Fig. 1), the voltage through each pair is the same. Hence, we isolated one of the pairs to calculate the resistance required for each LED/resistor pair.

### 2.2 Calculations

Using $Power_{dissipated}$ and the intended LED voltage drop, we calculated the system current:

$$\begin{aligned} P &= IV \\ 75 \times 10^{-3} &= I(2V) \\ I &= 37.5 \times 10^{-3}A \end{aligned}$$

With the calculated system current and intended resistor voltage drop, we calculated the appropriate resistance:

$$\begin{aligned} V &= IR \\ R &= \frac{V}{I} \\ &= \frac{3V}{37.5 \times 10^{-3}} \\ &= 80\Omega \end{aligned}$$

## 2.3 Component Values

Because voltage is proportional to resistance, we know that the resistance for each resistor must be a minimum of 80Ω. The closest resistance available was 100Ω, which we used for every LED/resistor pair.

# 3 Circuit Diagram

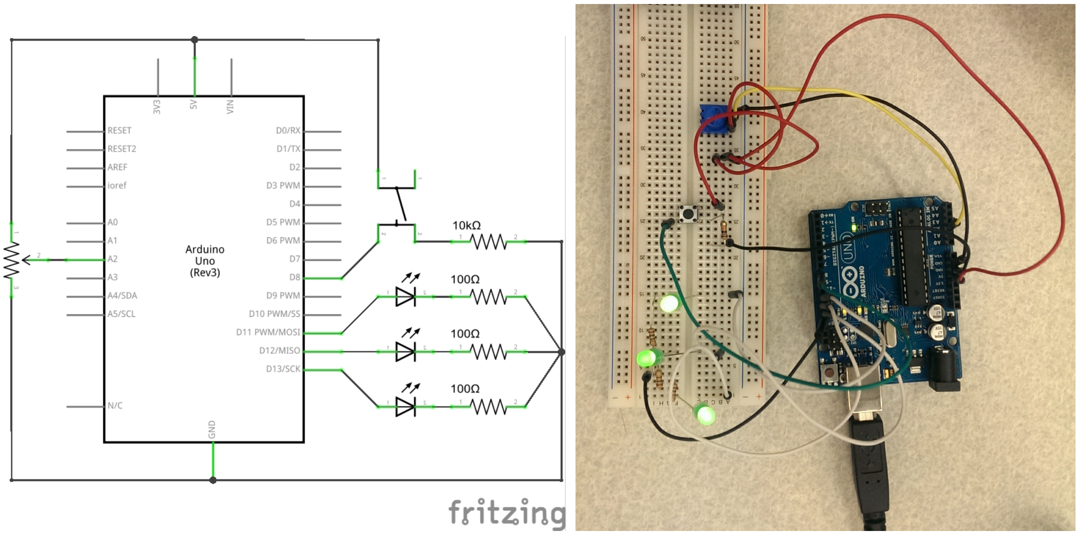
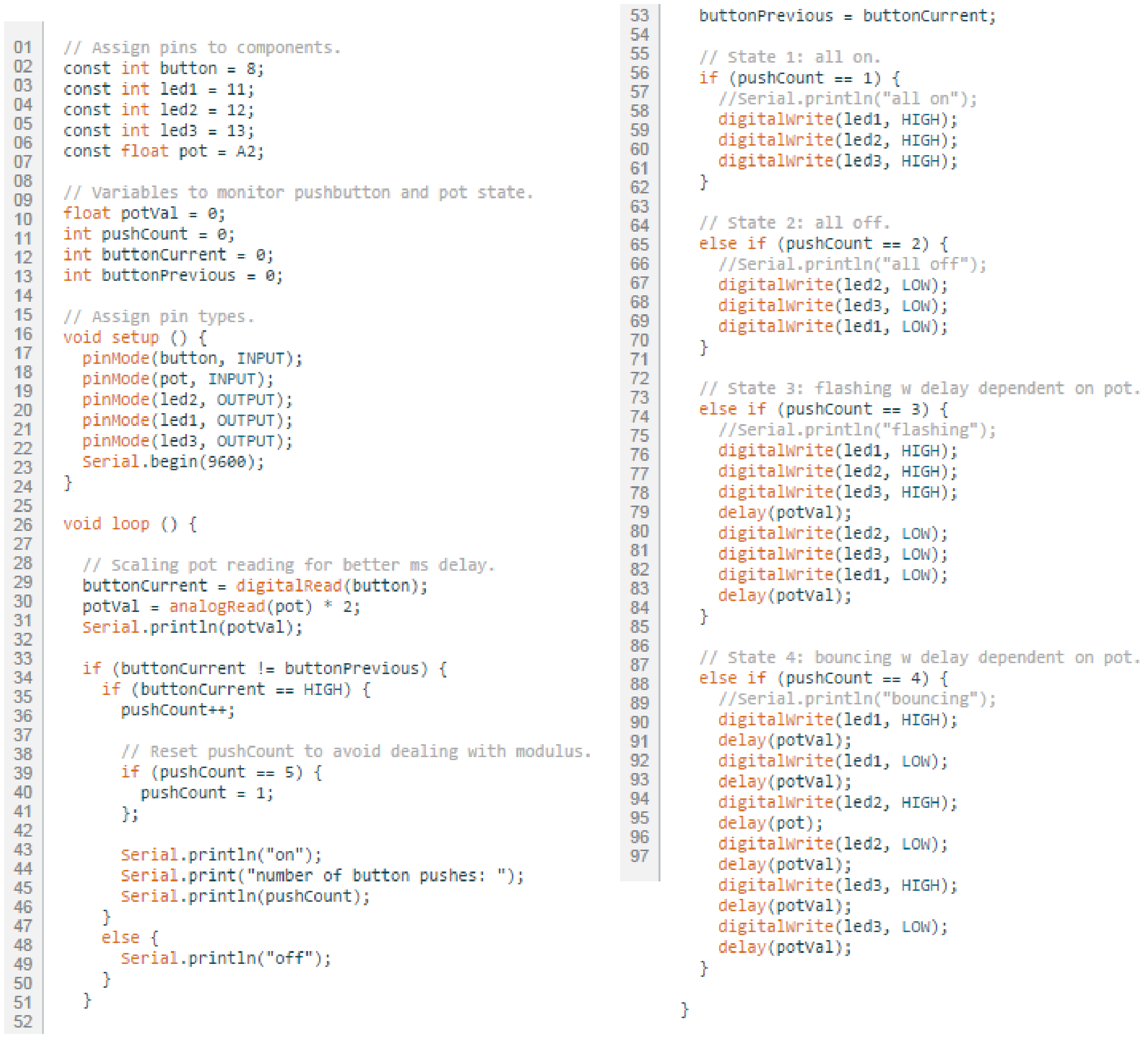


Figure 1: Schematic and photo of the Part II circuit.

The Part I circuit is almost identical to the Part II circuit, the only difference being the omission of the potentiometer. To use the pot to vary bounce and flash speed, we connected the pot's output to an analog pin (as digital pins only register "HIGH" or "LOW" ranges rather than distinct values). We then, as seen in Lines 28-31 of the code below, multiplied the output value by two to exaggerate the delay.

# 4 Source Code

The following code applies to both Parts 1 and 2.



```
// Assign pins to components.
const int button = 8;
const int led1 = 11;
const int led2 = 12;
const int led3 = 13;
const float pot = A2;

// Variables to monitor pushbutton and pot state.
float potVal = 0;
int pushCount = 0;
int buttonCurrent = 0;
int buttonPrevious = 0;

// Assign pin types.
void setup () {
  pinMode(button, INPUT);
  pinMode(pot, INPUT);
  pinMode(led2, OUTPUT);
  pinMode(led1, OUTPUT);
  pinMode(led3, OUTPUT);
  Serial.begin(9600);
}

void loop () {

  // Scaling pot reading for better ms delay.
  buttonCurrent = digitalRead(button);
  potVal = analogRead(pot) * 2;
  Serial.println(potVal);

  if (buttonCurrent != buttonPrevious) {
    if (buttonCurrent == HIGH) {
      pushCount++;

      // Reset pushCount to avoid dealing with modulus.
      if (pushCount == 5) {
        pushCount = 1;
      };

      Serial.println("on");
      Serial.print("number of button pushes: ");
      Serial.println(pushCount);
    }
    else {
      Serial.println("off");
    }
  }

  buttonPrevious = buttonCurrent;

  // State 1: all on.
  if (pushCount == 1) {
    //Serial.println("all on");
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
  }

  // State 2: all off.
  else if (pushCount == 2) {
    //Serial.println("all off");
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led1, LOW);
  }

  // State 3: flashing w delay dependent on pot.
  else if (pushCount == 3) {
    //Serial.println("flashing");
    digitalWrite(led1, HIGH);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, HIGH);
    delay(potVal);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    digitalWrite(led1, LOW);
    delay(potVal);
  }

  // State 4: bouncing w delay dependent on pot.
  else if (pushCount == 4) {
    //Serial.println("bouncing");
    digitalWrite(led1, HIGH);
    delay(potVal);
    digitalWrite(led1, LOW);
    delay(potVal);
    digitalWrite(led2, HIGH);
    delay(pot);
    digitalWrite(led2, LOW);
    delay(potVal);
    digitalWrite(led3, HIGH);
    delay(potVal);
    digitalWrite(led3, LOW);
    delay(potVal);
  }

}
```

# 5 Reflection

Liani: I've never seriously coded anything, and it's been a while since I've programmed an Arduino (the most recent time being the middle of last semester for Real-World Measurements). Fortunately, my experiences over the summer have made me more comfortable with coding, so the challenge this time around was figuring out C syntax. The largest hurdle in both the lab and pre-lab was figuring out how the push-button operated.

Abi: I think the first lab for PoE was a good introduction to the class, because even though it wasn't too difficult to tackle the task given, it allowed us to explore Arduino coding. Considering that I haven't had much experience coding, this gave me an opportunity to learn about Arduino syntax and structure. In addition, Liani and I both used GitHub for our code, which was a new experience for me. Even though we didn't really have to work outside of class, I definitely enjoyed the experience of learning how to code in Arduino and how to use GitHub.