

# Lab 1: Bike Light with Digital and Analog Inputs

Richard Gao and Theo Johnson

September 8, 2019

## 1 Description

In part 1 of this lab we were tasked to build a bike light with at least 3 LEDs and 5 LED display modes. In part 2 we were asked to add an analog input to our design from part 1 such that the input modifies the behaviour of the bike light. In our final design shown in Figure 1 we used 3 LEDs with 5 modes: all LEDs off, LED 1 on, LED 2 on, LED 3 on, and flashing LEDs. We also used a potentiometer as our analog input to control the speed of the flashing LEDs on the last mode.

## 2 Circuit Design

For part 1 of the lab, we built a system featuring an Arduino, three LEDs in parallel, and a push button based off of the schematic pictured in the lab description. What was left unknown to us was the resistor values that were in series with each LED. To figure out those resistor values, we used the equation

$$1/310\Omega = 3/R,$$

which we derived from the total resistance equation for parallel circuits, as the total resistance should equal 310 ohms. We originally determined this by looking at the LED datasheet which said the ideal voltage drop over an LED is 1.9v with a current of 10 mA, giving us the equation

$$(5v - 1.9v)/.01A = 310\Omega$$

Because the LEDs in the schematic is in parallel, each LED would drop the same amount: 1.9v. This resulted in the ideal resistance being 930 ohms which we reproduced using two 470 ohms resistors in series of each LED (940 ohms).

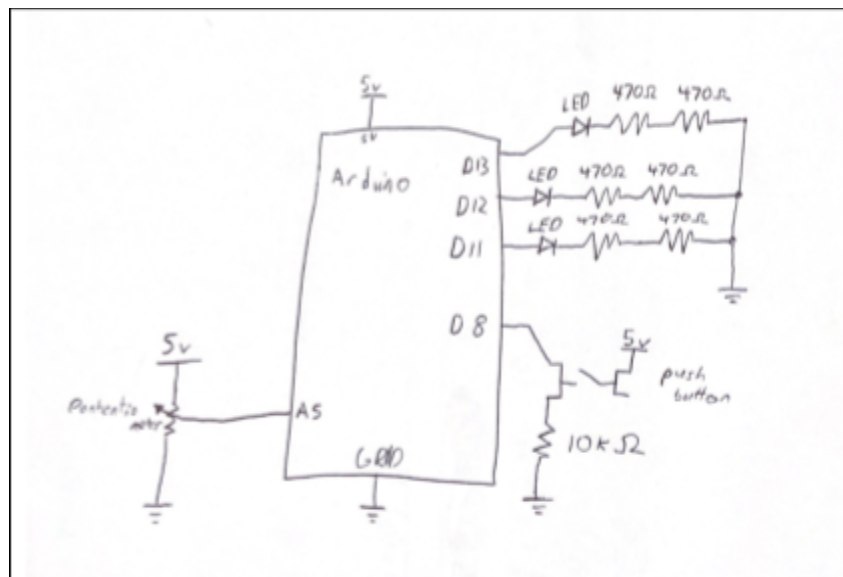


Figure 1: The final schematic

For part 2 of the lab, we added a potentiometer to analog port 5 and connected it to 5v and ground. This allowed us to modify our flashing LED mode to speed it up or slow it down depending on the value from the potentiometer.

### 3 Arduino Code

To add functionality to the circuit, we wrote code for the arduino so that LEDs can display different patterns and the button can switch between them. In our code we initialized input and output pins so we could read and write from and to the electronics on the board.

We programmed the LEDs by writing to each LED a high or low signal that corresponds with the LED being on or off. To flash the LEDs, we switched between writing them all on or all off with a time delay of minimum 100ms. We were able to change the speed of the LEDs by adding the potentiometer value read in by the A5 pin to the delay time.

The hardest part of this lab was getting the push button to work as intended (trigger only when pushed down). To accomplish this, we introduced debouncing to our code which allowed us to limit when our code checks for a button press to within a certain time specified by our variable *debounceDelay* and only when pressed down as shown in Figure 2.

```
int reading = digitalRead(buttonPin);

// Resets lastDebounceTime when button is pressed or released
if (reading != lastButtonState){
  lastDebounceTime = millis();
}

// Checks to change modeLED if time greater than debounceDelay has elapsed
if ((millis() - lastDebounceTime) > debounceDelay){
  // Prevents button being held from changing modeLED
  if (reading != buttonState){
    buttonState = reading;
    if (buttonState == HIGH){
      modeLED++;
      Serial.println(modeLED);
    }
  }
}
```

Figure 2: Debouncing code

### 4 Reflection

In this lab, we were reintroduced with concepts from ISIM, specifically circuits and circuit calculations. We felt comfortable again with this material fairly quickly. We were also introduced to Arduinos and the coding for them. Because we weren't very familiar with Arduinos, this challenged us more as we had to learn how to set them up in both hardware and software. We

also had to learn how to code them. Arduino's website was very helpful with learning the syntax. We also learned new concepts in coding, namely using millis vs delays. Ultimately, we felt this lab was a good combination of refreshing on old content while introducing us to the coming content of this course.

```
/*
 * POE Lab 1 - Build Your Own Bike Light
 * Richard Gao & Theo Johnson
 */

// Declare I/O pins
const int buttonPin = 8;
const int led1 = 11;
const int led2 = 12;
const int led3 = 13;
const int potPin = A5;

// modeLED keeps track of the current LED display mode
int modeLED = 0;

// Initialize debouncing variables
unsigned long lastDebounceTime = 0;
unsigned long debounceDelay = 50;
int buttonState = 0;
int lastButtonState = LOW;

// Initialize flag and time variables for flashing LED mode
bool flash = true;
unsigned long flashTime = 0;

void setup() {
  // Initialize inputs
  pinMode(buttonPin, INPUT);
  pinMode(buttonPin, INPUT_PULLUP);

  // Initialize output
  pinMode(led1, OUTPUT);
  pinMode(led2, OUTPUT);
  pinMode(led3, OUTPUT);

  Serial.begin(9600);
}
```

```
void loop() {
  int reading = digitalRead(buttonPin);

  // Resets lastDebounceTime when button is pressed or released
  if (reading != lastButtonState){
    lastDebounceTime = millis();
  }

  // Checks to change modeLED if time greater than debounceDelay
  has elapsed
  if ((millis() - lastDebounceTime) > debounceDelay){
    // Prevents button being held from changing modeLED
    if (reading != buttonState){
      buttonState = reading;
      if (buttonState == HIGH){
        modeLED++;
        Serial.println(modeLED);
      }
    }
  }
}

int potValue = analogRead(potPin);

// Defines the different LED modes
switch (modeLED % 5) {
  // All off
  case 0:
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);
    break;
  // LED1 on
  case 1:
    digitalWrite(led1, HIGH);
    digitalWrite(led2, LOW);
    digitalWrite(led3, LOW);

```

```

    break;
// LED2 on
case 2:
    digitalWrite(led1, LOW);
    digitalWrite(led2, HIGH);
    digitalWrite(led3, LOW);
    break;
// LED3 on
case 3:
    digitalWrite(led1, LOW);
    digitalWrite(led2, LOW);
    digitalWrite(led3, HIGH);
    break;
// Flashing LEDs
case 4:
    // Switch LED states if 100ms + variable potValue time has
passed
    if (millis() - flashTime > 100 + potValue) {
        flash = !flash;
        digitalWrite(led1, flash);
        digitalWrite(led2, flash);
        digitalWrite(led3, flash);
        flashTime = millis();
    }
    break;
default:
    Serial.println("Something went wrong. Skipped all modes.");
    break;
}

// Remember the last button reading
lastButtonState = reading;
}

```