Part A. Revisiting Blink

1. Blinking LEDs with Arduino

a. What line(s) of code do you need to change to make the LED blink (like, at all)?

No line needs to be changed if we want to make the onboard LED blink.

If we want to make an external LED blink, we need to change LED_BUILTIN into pin 9.

```
const int ledPin =  9;      // the number of the LED pin

// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(ledPin, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(ledPin, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(ledPin, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

b. What line(s) of code do you need to change to change the rate of blinking?

The second delay( ) (the highlighted line) needs to be changed to change the rate of blinking. We could also change the other delay() so that the LED is on and off for the same amount of time.

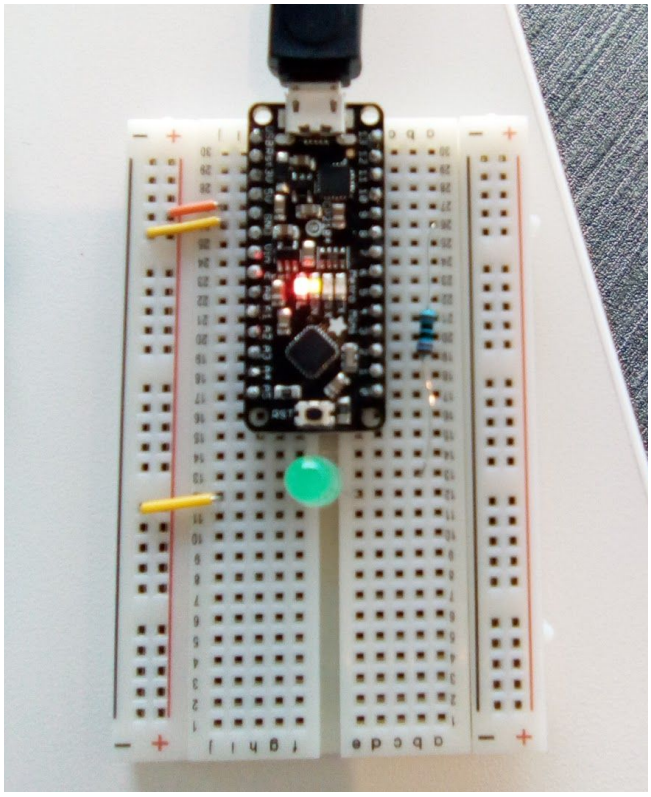
```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);   // turn the LED on (HIGH is the voltage level)
  delay(1000);                       // wait for a second
  digitalWrite(LED_BUILTIN, LOW);    // turn the LED off by making the voltage LOW
  delay(1000);                       // wait for a second
}
```

Quick blinking with shorter time between blinks and shorter length of blink:

```
// the setup routine runs once when you press reset:
void setup() {
  // initialize the digital pin as an output.
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);   // turn the LED on (HIGH is the voltage
  delay(100);                // wait for a second
  digitalWrite(led, LOW);    // turn the LED off by making the volta
  delay(50);                 // wait for a second
}
```

c. What circuit element would you want to add to protect the board and external LED?

We would want to add a resistor to protect the board and external LED.This prevents a voltage higher than LED voltage drop being applied to LED.



2. Digitally toggle LEDs on and off using the Arduino

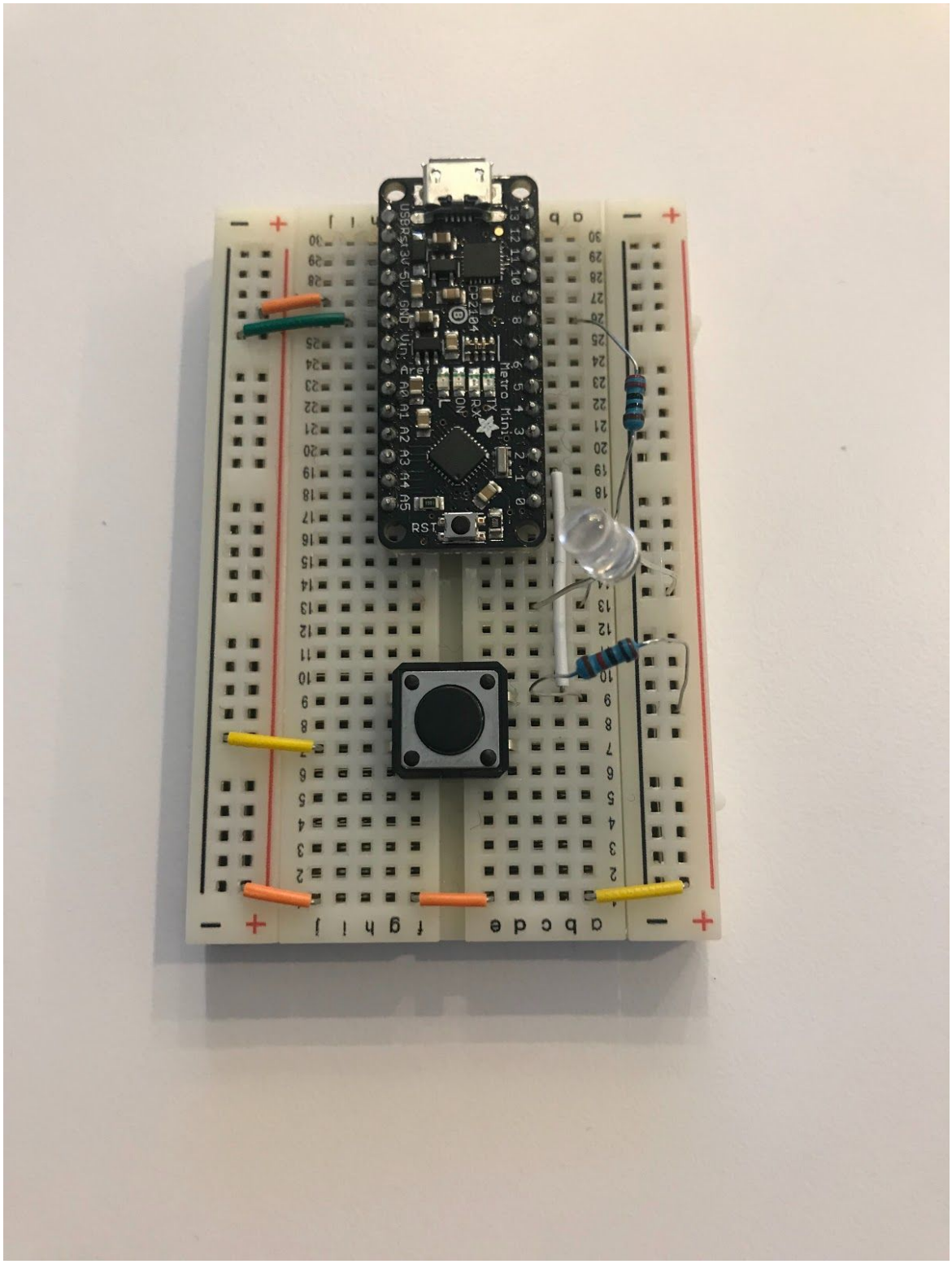
a. Which lines do you need to modify to correspond with your button and LED pins?

We need to change the ledPin from 13 to 9.

```
  This example code is in the public domain.

  http://www.arduino.cc/en/Tutorial/Button
*/

// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  9;       // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is HIGH:
  if (buttonState == HIGH) {
    // turn LED on:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED off:
    digitalWrite(ledPin, LOW);
  }
}
```

b. Modify the code or the circuit so that the LED lights only while the button is depressed. Include your code in your lab write-up.

We don’t really need to change the code itself if we were to use the alternate circuit provided as shown below.



The only things need to be changed are comments. When the button is pressed, the IO2, which is the buttonPin is grounded. Therefore, the buttonState is LOW instead of HIGH. When the buttonState is HIGH, which means button is not pressed, we want the LED off. So, we need the ledPin HIGH as it is. Thus, no code needs to be changed but comments.

```
// constants won't change. They're used here to set pin numbers:
const int buttonPin = 2;     // the number of the pushbutton pin
const int ledPin =  9;      // the number of the LED pin

// variables will change:
int buttonState = 0;         // variable for reading the pushbutton status

void setup() {
  // initialize the LED pin as an output:
  pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(buttonPin, INPUT);
}

void loop() {
  // read the state of the pushbutton value:
  buttonState = digitalRead(buttonPin);

  // check if the pushbutton is pressed. If it is, the buttonState is LOW:
  if (buttonState == HIGH) {
    // turn LED off:
    digitalWrite(ledPin, HIGH);
  } else {
    // turn LED on:
    digitalWrite(ledPin, LOW);
  }
}
```

---

3. Fading LEDs on and off using Arduino

a) Which line(s) of code do you need to modify to correspond with your LED pin?

No line needs to be modified since the ledPin is already set to 9.

b) How would you change the rate of fading?

We can either change delay() (highlighted line) or fadeValue. By increasing the delay time and/or decreasing the fadeValue, we can extend the length of each cycle.

```
int ledPin = 9;    // LED connected to digital pin 9

void setup() {
  // nothing happens in setup
}

void loop() {
  // fade in from min to max in increments of 5 points:
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    // sets the value (range from 0 to 255):
    analogWrite(ledPin, fadeValue);
    // wait for 300 milliseconds to see the dimming effect
    delay(300);
  }
```

c) (Extra) Since the human eye doesn't see increases in brightness linearly and the diode brightness is also nonlinear with voltage, how could you change the code to make the light appear to fade linearly?

Part B. Advanced Inputs

1. Potentiometer

2. Force Sensitive Sensor

a. What resistance values do you see from your force sensor?

When no pressure applied, the force sensor seems to have an infinite resistance as it act like an open circuit.

When moderate force applied, the force sensor has a resistance of around 1000 Ohms.

When the maximum force my fingers can apply applied, the force sensor has a resistance of roughly 340 Ohms.

b. What kind of relationship does the resistance have as a function of the force applied? (e.g., linear?)

The relationship is nonlinear. Close to inverse proportional.

c. Can you change the LED fading code values so that you get the full range of output voltages from the LED when using your FSR?

Yes. Since the maximum value sensorPin A0 can register is 1007 instead of 1023, and we wanted to get the full range, which is 0 to 255, we decided to let the outputValue saturate when the value A0 registers is greater than 1000.

```
int sensorPin = A0;    // select the input pin for the potentiometer
int ledPin = 9;      // select the pin for the LED
int sensorValue = 0;  // variable to store the value coming from the sensor
int outputValue = 0;

void setup() {
  // declare the ledPin as an OUTPUT:
  pinMode(ledPin, OUTPUT);
  // open the serial port at 9600 bps:
  Serial.begin(9600);
}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  // calculate the output value
  outputValue = min(double(sensorValue)/1000*255, 255);
  // set the output value of LED
  analogWrite(ledPin, outputValue);
  //Serial.print(outputValue);
  //Serial.print('\n');
  //delay(100);
}
```

Part C. Writing to the LCD

a. What voltage level do you need to power your display?

5V. (from the circuit diagram)

b. What voltage level do you need to power the display backlight?

3.3v. (from the circuit diagram)

b. What was one mistake you made when wiring up the display? How did you fix it?

We didn't make any mistake when wiring up the display.

c. What line of code do you need to change to make it flash your name instead of "Hello World"?

```
  // Print a message to the LCD.
  lcd.print("hello, world!");
```

```
void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Print a message to the LCD.
  lcd.print("Sarah & Hao");
}
```

d. Include a copy of your Lowly Multimeter code in your lab write-up.
We used the lcd.setCursor() from the Arduino LiquidCrystal Library.
https://www.arduino.cc/en/Tutorial/LiquidCrystalSetCursor

LowlyMultimeter §

```
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int sensorPin = A0; // select the input pin for the potentiometer
int sensorValue = 0; // variable to store the value coming from the sensor

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  // Serial.begin(9600);

}

void loop() {
  // read the value from the sensor:
  sensorValue = analogRead(sensorPin);
  lcd.setCursor(12,0);

  if (sensorValue <10) {
    lcd.print("   ");
  } else if (sensorValue < 100) {
    lcd.print("  ");
  } else if (sensorValue < 1000) {
    lcd.print(" ");
  }
  lcd.print(sensorValue);
  // Serial.print(sensorValue);
  // Serial.print('\n');
}
```

e. Include a copy of your FSR thumb wrestling code in your lab write-up.

```
/*
  LiquidCrystal Library - display() and noDisplay()

 Demonstrates the use a 16x2 LCD display.  The LiquidCrystal
 library works with all LCD displays that are compatible with the
 Hitachi HD44780 driver. There are many of them out there, and you
 can usually tell them by the 16-pin interface.

 This sketch prints "Hello World!" to the LCD and uses the
 display() and noDisplay() functions to turn on and off
 the display.

 The circuit:
 * LCD RS pin to digital pin 12
 * LCD Enable pin to digital pin 11
 * LCD D4 pin to digital pin 5
 * LCD D5 pin to digital pin 4
 * LCD D6 pin to digital pin 3
 * LCD D7 pin to digital pin 2
 * LCD R/W pin to ground
 * 10K resistor:
 * ends to +5V and ground
 * wiper to LCD VO pin (pin 3)

 Library originally added 18 Apr 2008
 by David A. Mellis
 library modified 5 Jul 2009
 by Limor Fried (http://www.ladyada.net)
 example added 9 Jul 2009
 by Tom Igoe
 modified 22 Nov 2010
 by Tom Igoe
 modified 7 Nov 2016
 by Arturo Guadalupi

 This example code is in the public domain.

 http://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay

*/

// include the library code:
#include <LiquidCrystal.h>

// initialize the library by associating any needed LCD interface pin
```

```
// with the arduino pin number it is connected to
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

int haoPin = A2; // select the input pin for the potentiometer
int sarahPin = A1;
int haoValue = 0; // variable to store the value coming from the sensor
int sarahValue = 0;

void setup() {
  // set up the LCD's number of columns and rows:
  lcd.begin(16, 2);
  //Serial.begin(9600);
}

void loop() {
  // read the value from the sensor:
  haoValue = analogRead(haoPin);
  sarahValue = analogRead(sarahPin);
  lcd.clear();

  int dif = haoValue - sarahValue;
  dif = abs(dif);

  if (dif < 3) {
    lcd.print("It's a tie!");
  } else {
    if (haoValue > sarahValue) {
      lcd.print("Hao's winning!");
    } else {
      lcd.print("Sarah's winning!");
    }
  }

  Serial.print(haoValue);
  Serial.print(" ");
  Serial.print(sarahValue);
  Serial.print('\n');
  delay(200);
}
```

Part D. Timer

a. Make a short video showing how your timer works, and what happens when time is up!

b. Post a link to the completed lab report to the class Slack.

Part E. Tone

a. How would you change the code to make the song play twice as fast?

We can either double the all of the noteDurations

```
int noteDurations[] = {
  4, 8, 8, 4, 4, 4, 4, 4
};
```

Or divide 1000 (1 minute) by 2

```
// to calculate the note duration, take one second divided by the note type.
//e.g. quarter note = 1000 / 4, eighth note = 1000/8, etc.
int noteDuration = 500 / noteDurations[thisNote];
tone(8, melody[thisNote], noteDuration);
```

b. What song is playing?

Star Wars Theme Song