

Lab 4 – Digital Timer

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?

I used digitalWrite to alternate between high and low values with a delay to program a blink.

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

To change the rate of blinking, the delays between the high and low states can be set to different values.

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

A resistor of suitable value should be used to protect the board and the external LED. The resistor is used to limit the current so as to not damage any components.

Code

```
int led = 9;

void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(led, OUTPUT);
}

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

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?

To correspond with my button and LED pins, the initializations for both should be set to pins 2 and 9 respectively.

Code

```
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.

The conditions in the last if statement were reversed for this case.

Code

```
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, LOW);
  } else {
    // turn LED off:
    digitalWrite(ledPin, HIGH);
  }
}
```

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?

To correspond with my LED pin, the initialization should be set to pin 9. Additionally, to control the LED to fade, analogWrite is to be used instead of digitalWrite.

Code

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

void setup() {
}

void loop() {
  for (int fadeValue = 0 ; fadeValue <= 255; fadeValue += 5) {
    analogWrite(ledPin, fadeValue);
    delay(30);
  }
  for (int fadeValue = 255 ; fadeValue >= 0; fadeValue -= 5) {
    analogWrite(ledPin, fadeValue);
    delay(30);
  }
}
```

b) How would you change the rate of fading?

To change the rate of fading, the fadeValue and delays should be changed.

c) 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?

Increase delay for smaller voltages and decrease delay at higher voltages.

Code

```
int ledPin = 9; // set LED pin to pin 9
int ledOutput = 0; // initialize output to LED pin
int fadeRate = 10; // amount of change for each loop

void setup() {
  pinMode(ledPin, OUTPUT); // initialize LED pin as an output.
}

void loop() {
  for (int ledOutput = 0; ledOutput <= 255; ledOutput += fadeRate) {
    analogWrite(ledPin, ledOutput);
    delay(255 - ledOutput);
  }
  for (int ledOutput = 255; ledOutput >= 0; ledOutput -= fadeRate) {
    analogWrite(ledPin, ledOutput);
    delay(ledOutput);
  }
}
```

Part B – Advanced Inputs

1. Potentiometer

a) Post a copy of your new code in your lab write-up.

Code

```
int potPin= A0; //Declare potPin to be analog pin A0
int LEDPin= 9; // Declare LEDPin to be arduino pin 9
int readValue; // Use this variable to read Potentiometer
int writeValue; // Use this variable for writing to LED

void setup() {
  pinMode(potPin, INPUT); //set potPin to be an input
  pinMode(LEDPin, OUTPUT); //set LEDPin to be an OUTPUT
}

void loop() {

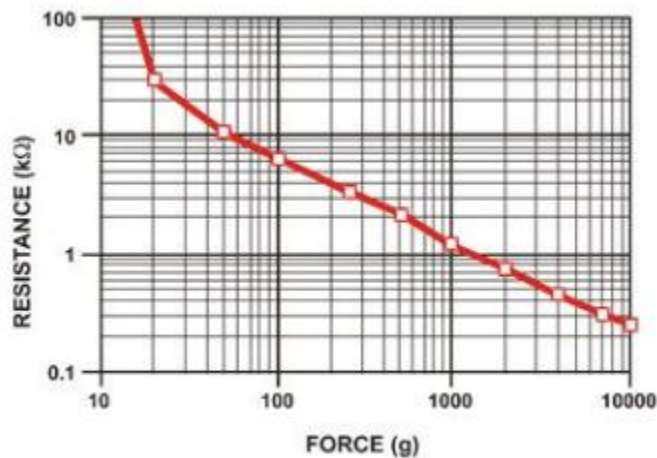
  readValue = analogRead(potPin); //Read the voltage on the Potentiometer
  writeValue = (255./1023.) * readValue; //Calculate Write Value for LED
  analogWrite(LEDPin, writeValue); //Write to the LED
}
```

2. Force Sensitive Resistor

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

The FSR has a resistance value of infinity and acts like an open circuit when no pressure is applied. When light pressure is applied, the resistance value is close to 100k Ω and when maximum pressure is applied, the resistance value is 200 Ω .

b) What kind of relationship does the resistance have as a function of the force applied?



The FSR response approximately follows an inverse power-law characteristic (roughly $1/R$).

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?

To get the full range of output voltages, the writeValue sent to the LED should be $(255/1023) \times \text{readValue}$ from the FSR.

Code

```
int FSRPin= A0; //Declare potPin to be analog pin A0
int LEDPin= 9; // Declare LEDPin to be arduino pin 9
int readValue; // Use this variable to read Potentiometer
int writeValue; // Use this variable for writing to LED

void setup() {
  pinMode(FSRPin, INPUT); //set potPin to be an input
  pinMode(LEDPin, OUTPUT); //set LEDPin to be an OUTPUT
}

void loop() {

  readValue = analogRead(FSRPin); //Read the voltage on the Potentiometer
  writeValue = (255./1023.) * readValue; //Calculate Write Value for LED
  analogWrite(LEDPin, writeValue); //Write to the LED
}
```

Part C – Writing to the LCD

a) What voltage level do you need to power your display? What voltage level do you need to power the display backlight?

5V is required to power the display and 3.3V is required to power the backlight. However, both work when connected to 5V.

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

I had no issues wiring up the display.

c) What line of code do you need to change to make it flash your name instead of “Hello World”?

Change the code `lcd.print("Hello World")` to `lcd.print("Rohan")`.

d) Include a copy of your Lowly Multimeter code.

Code

```
#include <LiquidCrystal.h>

int potPin= A0; //Declare potPin to be analog pin A0
int LEDPin= 9; // Declare LEDPin to be arduino pin 9
int readValue; // Use this variable to read Potentiometer
int writeValue; // Use this variable for writing to LED
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  pinMode(potPin, INPUT); //set potPin to be an input
  pinMode(LEDPin, OUTPUT); //set LEDPin to be an OUTPUT
  lcd.begin(16, 2);
  lcd.print("Analog Value");
}

void loop() {

  readValue = analogRead(potPin); //Read the voltage on the Potentiometer
  writeValue = (255./1023.) * readValue; //Calculate Write Value for LED
  analogWrite(LEDPin, writeValue); //Write to the LED
  lcd.setCursor(0, 1);
  lcd.print(writeValue);
}
```


e) Include a copy of your FSR thumb wrestling code.**Code**

```
#include <LiquidCrystal.h>

int FSRPin= A0;
int FSRPin2= A1;
int readValue;
int readValue2;
int writeValue;
int writeValue2;
const int rs = 12, en = 11, d4 = 5, d5 = 4, d6 = 3, d7 = 2;
LiquidCrystal lcd(rs, en, d4, d5, d6, d7);

void setup() {
  pinMode(FSRPin, INPUT);
  pinMode(FSRPin2, INPUT);
  lcd.begin(16, 2);
}

void loop() {

  readValue = analogRead(FSRPin);
  readValue2 = analogRead(FSRPin2);
  writeValue = (255./1023.) * readValue;
  writeValue2 = (255./1023.) * readValue2;

  lcd.setCursor(0, 0);
  lcd.print("Player 1:");
  lcd.setCursor(0, 1);
  lcd.print("Player 2:");
  lcd.setCursor(11, 0);
  lcd.print(writeValue);
  lcd.setCursor(11, 1);
  lcd.print(writeValue2);
}
```

Part D - Timer

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

Link: <https://youtu.be/z1bDnXbD2qo>

Code

Link: <https://qoo.gl/q634xB>

Part E - Tone

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

Change the value of noteDuration.

b) What song is playing?

The Star Wars theme!