

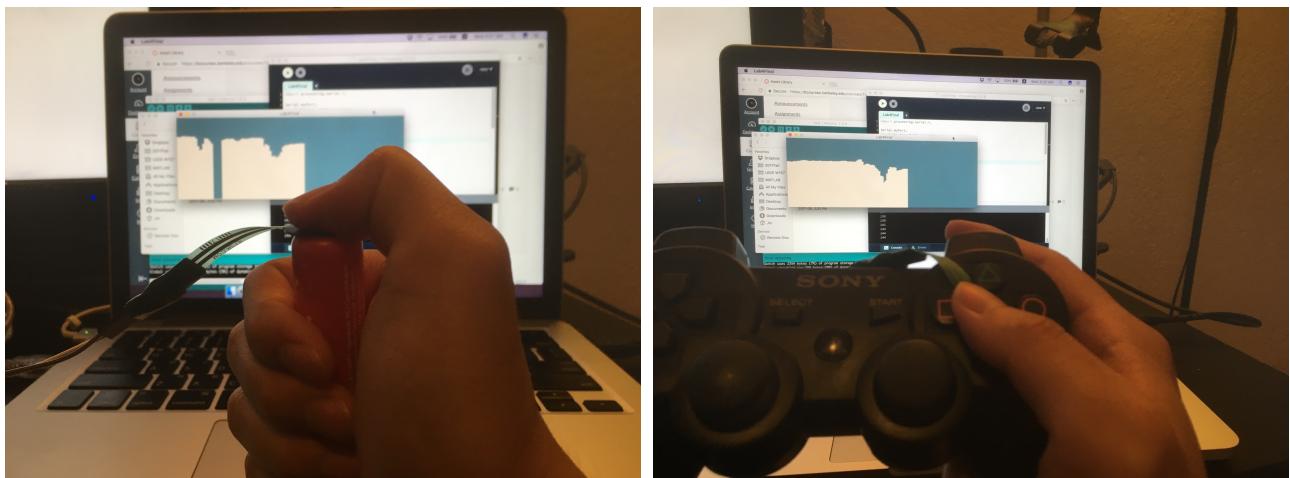
Jin Jeon
Info C262

Lab 4: FSR and Serial Communication with Processing

Description

In this lab, I used Processing to visualize the serial input converted from Force Sensitive Resistor (FSR) to Arduino. The serial input is constantly communicated from Arduino to Processing, which uses the values to graph them visually. The stronger the button is pressed, the greater the graph will rise in the Processing visualization. If there is no input, the graph will show no activity. On the other hand, if the force input is weak, the red LED will turn bright.

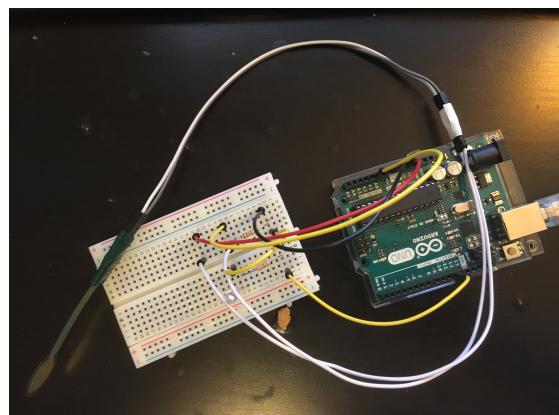
The inspiration behind this idea was an app game called [Flappy Bird](#). I wanted to apply such gamification elements into the Processing visual representation. By having the user control the amount of force input, the user can control the height or spiking of the graph and safely avoid peaking too high or low. The red LED light will spark bright if the user is “flying” too low. While I tried to randomly assign lines and curves that the user needs to avoid, manually inputting the coordinates required too much simple trials and failures. If given more time, I would love to apply more game elements and develop it into a short working-game.

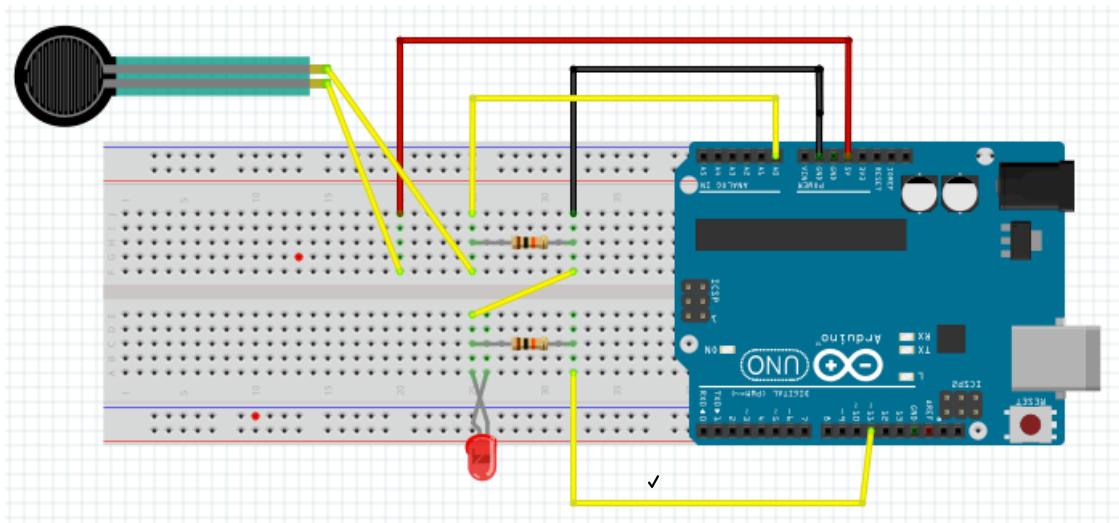


In order to give sense of gaming, I thought of using a joystick. Since it is a single button, I used a lip balm to place the FSR on top and work as a joystick. I also tried placing the FSR on top of PlayStation joystick.

Components

- 1 Arduino
- 1 LED (red)
- 2 Resistor (10K resistor)
- 1 Breadboard
- 1 FSR
- 5 Jumper wires





Arduino Code

```

int fsrPin = 0; // the cell and 10K pulldown are connected to a0
int fsrReading; // the analog reading from the sensor divider
int LEDpin = 11; // connect Red LED to pin 11 (PWM pin)
int LEDbrightness; //

void setup() {
  Serial.begin(9600);
}

void loop() {
  int analogValue = analogRead(A0)/4; // read the sensor value
  Serial.write(analogValue); // send the value serially
  // as a binary value
  fsrReading = analogRead(fsrPin);

  // LED lights red when sensor is weak = meaning danger
  // that means we have to -invert- the reading from 0-1023 back to 1023-0
  fsrReading = 1023 - fsrReading;
  //now we have to map 0-1023 to 0-255 since that's the range analogWrite uses
  LEDbrightness = map(fsrReading, 0, 1023, 0, 255);
  analogWrite(LEDpin, LEDbrightness);

  delay(100);
}

```

Processing Code

```

import processing.serial.*;

Serial myPort;
// at the top of the program:
float xPos = 0; // horizontal position of the graph
float yPos = 500; // vertical position of the graph
void setup () {
  size(700, 250); // window size

  // Lists all the available serial ports
  println(Serial.list());
}

```

```
String portName = Serial.list()[1];
myPort = new Serial(this, portName, 9600);

background(#597D87);

strokeWeight(3);

}

void serialEvent (Serial myPort) {
    // get serial input
    int inByte = myPort.read();
    // print
    println(inByte);

    yPos = height - inByte;
}

void draw () {
    // drawing lines based on FSR input

    stroke(#F2E3D5);
    line(xPos, height, xPos, yPos);

    // Clearing the screen
    if (xPos >= width) {
        xPos = 0;
        // clear screen once width is filled;
        background(#597D87);
    } else {
        // increment x value for next reading;
        xPos++;
    }
}
```