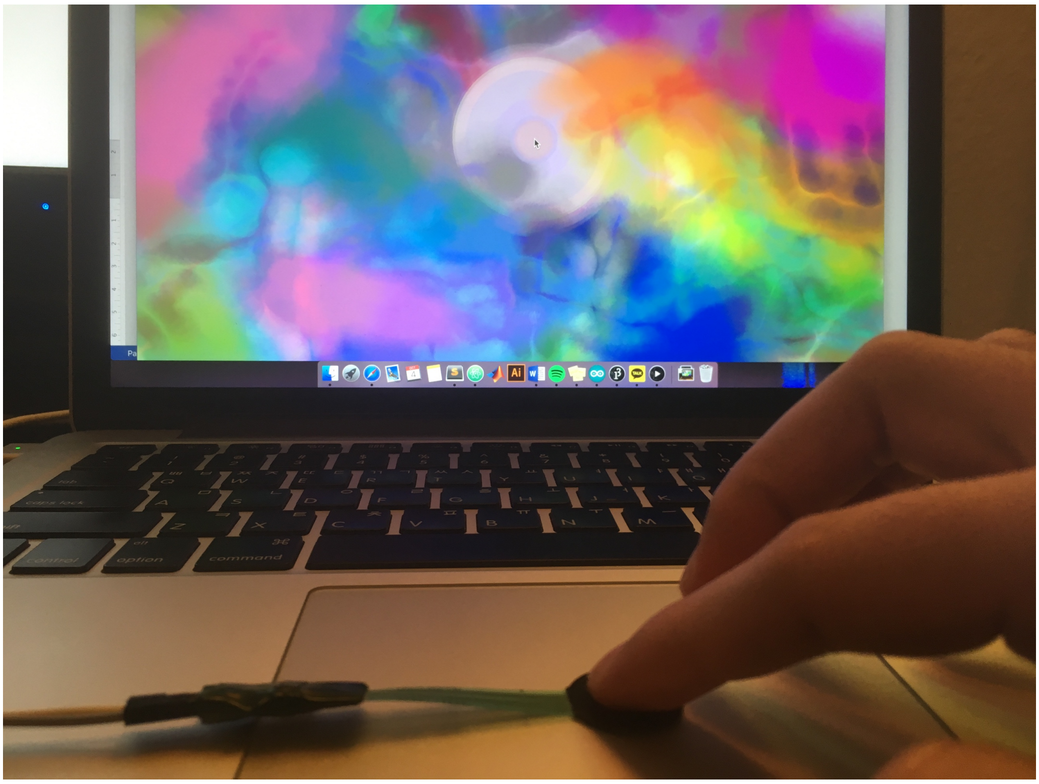
Jin Jeon
Info C262

Lab 5: Serial Communication with Firmata, Arduino, & Processing

Description

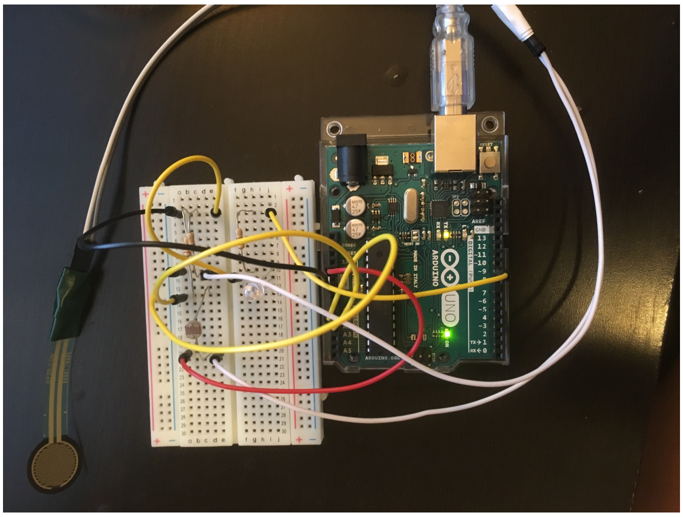
I used Firmata to have Arduino and Processing communicate. I connected FSR and photocell to create graphical output in Processing. In Processing, while there is a calm piano sound playing, there is an automatic background display of calm watercolor spreading. I used the existing open project from Openprocessing.org. On top of the graphics, I had custom shaped ellipses follow the mouse as it glides over the sketch. Depending on the FSR serial input, the size of the ellipses changes. The greater the force put into FSR, the bigger the circle will be. While the circles that follow the mouse cursor is opaque to fit the background image, the color of the circles change depending on the values communicated with the photocell. The brighter the surrounding, the brighter circle outputs will be. I placed the FSR to overlap with the finger manipulating the touchpad so that the FSR can naturally capture the force as the finger glides over the touchpad.

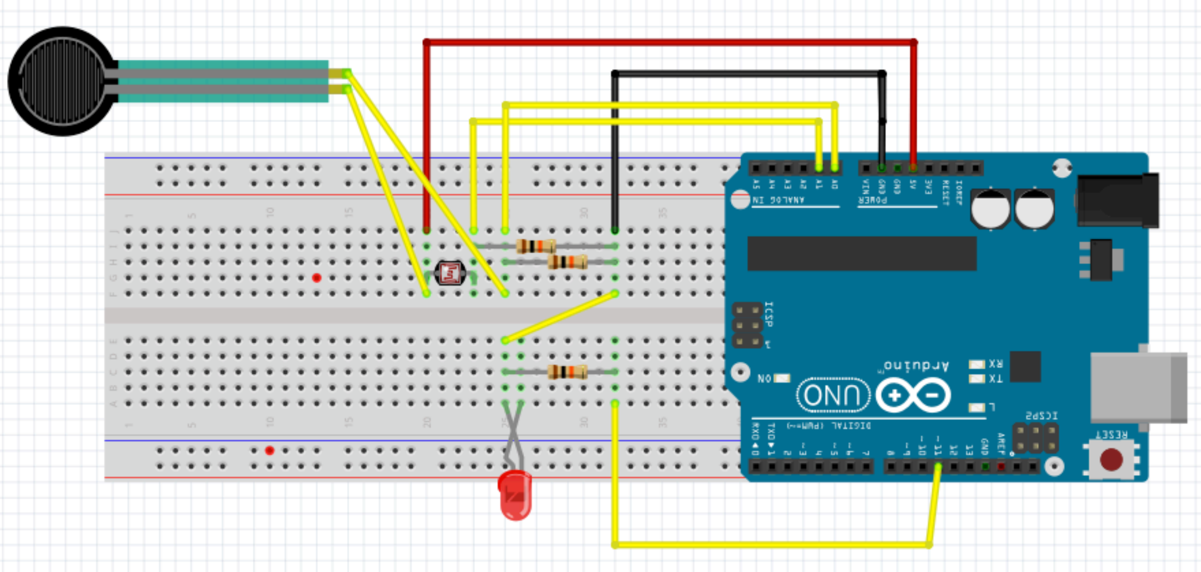


The two methods, custom serial and Firmata, both are similar in that they directly use serial communication values to Processing for implementations. I find using Firmata to be useful and efficient because there is less code pages to worry about. If I were to use custom serial communication, I would usually open both Arduino and Processing, and make changes accordingly in one and the other, but having all codes organized in Processing made the task easier and efficient. Even though I'd prefer using Firmata, using custom serial may have the advantage of having each variable organized separately, while Arduino code has to be set prior in order to use Firmata.

Components

1 Arduino
1 LED (red)
3 Resistor (10K resistor)
1 Breadboard
1 FSR
1 Photocell
6 Jumper wires





Arduino Code

```
Given code from Lab5 page (Skipped because it’s too long)
```

Processing Code

```
//Jin Jeon
//Info C262 Lab 5
//Inspiration and background graphics by Levente Sandor's Watercolor

import processing.serial.*;
import cc.arduino.*;
import processing.sound.*;

Arduino arduino;
int clr2;
int ledBrightness = 0;
int sensorFSR = 0; // pin A0 FSR
int sensorPhoto = 1; // pin A1 photocell
int FSRVal = 10;
int PhotoVal = 10;
SoundFile file;
```

```
ArrayList<Brush> brushes;
float x;
float y;

void setup() {
  println(Arduino.list());
  arduino = new Arduino(this, Arduino.list()[1], 57600);
  arduino.pinMode(sensorFSR, Arduino.INPUT);
  arduino.pinMode(sensorPhoto, Arduino.INPUT);

  size(1200, 700);
  background(255);
  frameRate(50);
  brushes = new ArrayList<Brush>();
  for (int i = 0; i < 50; i++) {
    brushes.add(new Brush());
  }
  noStroke();

  //For sound play
  file = new SoundFile(this,
"/Users/Jin/dropbox/2017Fall/InfoC262/Lab5/shadydave__my-love-piano-
loop.mp3");
  file.play();
  //file.stop();
  file.loop();
}

void draw() {
  FSRVal = arduino.analogRead(sensorFSR);
  if (FSRVal < 10) {
    FSRVal = 25;
  }
  PhotoVal = arduino.analogRead(sensorPhoto) + 30;
  if (PhotoVal < 50) {
    PhotoVal = 75;
  }
  if (PhotoVal >= 255) {
    PhotoVal = PhotoVal - 255;
  }
  println("PhotoVal: " + PhotoVal);

  for (Brush brush : brushes) {
    brush.paint();
  }

  x = lerp(x, mouseX, 0.08);
  y = lerp(y, mouseY, 0.08);
  if (PhotoVal > 120) {
    clr2 = color(random(125,255),random(125,255),random(125,255));
  } else {
    clr2 = color(random(0,124),random(0,124),random(0,124));
  }

  //ellipseMode(CORNER);
  fill(clr2, 40);
  println("FSRValue: " + FSRVal);
  ellipse(x, y, FSRVal, FSRVal);
}

void mouseClicked() {
  setup();
}

class Brush {
  float angle;
  int components[];
  float x, y;
  color clr;

  Brush() {
    PhotoVal = PhotoVal * 4;
    //println(PhotoVal);
    angle = random(TWO_PI);
    x = random(width);
    y = random(height);
    clr = color(random(255), random(255), random(255), 5);
    components = new int[2];
    for (int i = 0; i < 2; i++) {
      components[i] = int(random(1, 5));
    }
  }

  void paint() {
    float a = 0;
    float r = 0;
    float x1 = x;
    float y1 = y;
    float u = random(0.5, 1);

    fill(clr);
    noStroke();

    beginShape();
    while (a < TWO_PI) {
      vertex(x1, y1);
      float v = random(0.85, 1);
      x1 = x + r * cos(angle + a) * u * v;
      y1 = y + r * sin(angle + a) * u * v;
      a += PI / 180;
      for (int i = 0; i < 2; i++) {
        r += sin(a * components[i]);
      }
    }
    endShape(CLOSE);

    if (x < 0 || x > width ||y < 0 || y > height) {
      angle += HALF_PI;
    }

    x += 2 * cos(angle);
    y += 2 * sin(angle);
    angle += random(-0.15, 0.15);
  }
}
```