YHU 2264 Final Project Report

Dancing With The Stars

By Jasmine Tan (Solo Project)

Project Synopsis

Dancing With The Stars is a project that re-envisions dance as an art that can be enhanced by visual effects calibrated by the dancer's movements. Specifically, using self-made velostat sensors in shoe soles, the real-time motion of a dancer's feet is translated into visual output, in the form of light.

Motivation of My Work

Being someone who loves to dance, I was personally motivated to think of dance as a time-based art form in this class. Much of my time was spent envisioning how the movements of a dancer can be transformed into some sort of impactful output for the audience. Then, I was further inspired by "Pixel", an interactive dance performance created by Adrien M / Claire B Company, where dancers' actions generated images that were animated and projected live on stage. The idea of having an interactive and responsive dance performance that took real-time input of dancers' movements and produced an output of stunning visuals for the audience fascinated me. Eventually, I decided to make a pair of shoes that light up based on pressure input from the soles. This way, a dancer's movements on the feet can be translated into visual output, in the form of different colours and patterns of LED lights (as determined by the amount of pressure exerted on the front portion of the shoe sole).

Hardware Setup

For an overview of the setup, I refer to the following circuit diagram.

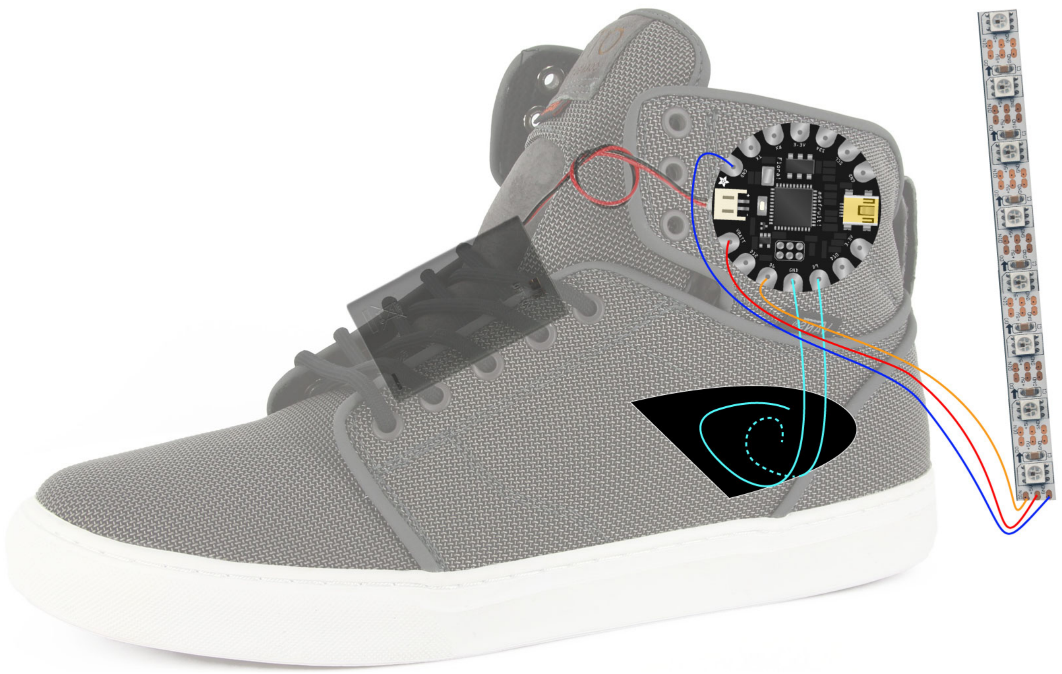

Figure 1: Circuit Diagram of Shoe
From: https://learn.adafruit.com/assets/10598
Diagrams made with Fritzing v0.7.5 library.

As seen, the circuit consists of a FLORA microcontroller is connected to an LED light strip, a pressure sensor (on the sole of the shoe) and a battery pack. The FLORA is a wearable microcontroller that can be sewn onto my shoes, since it is small and has pins available for threads to go through.

For this project, I made a pressure sensor using velostat, which is a material that has decreased resistance when more pressure is applied on it, hence allowing more electricity to flow through. The velostat pressure sensor is connected to GND (ground) and pin D9 on the FLORA microcontroller via conductive thread which goes through the side of the shoe. To test my velostat pressure sensor, I set an arbitrary pressure level to trigger the LED lights when pressure is applied to the sensor, as seen in Figure 2 (video file attached as Testing_of_Velostat_Pressure_Sensor.MOV).

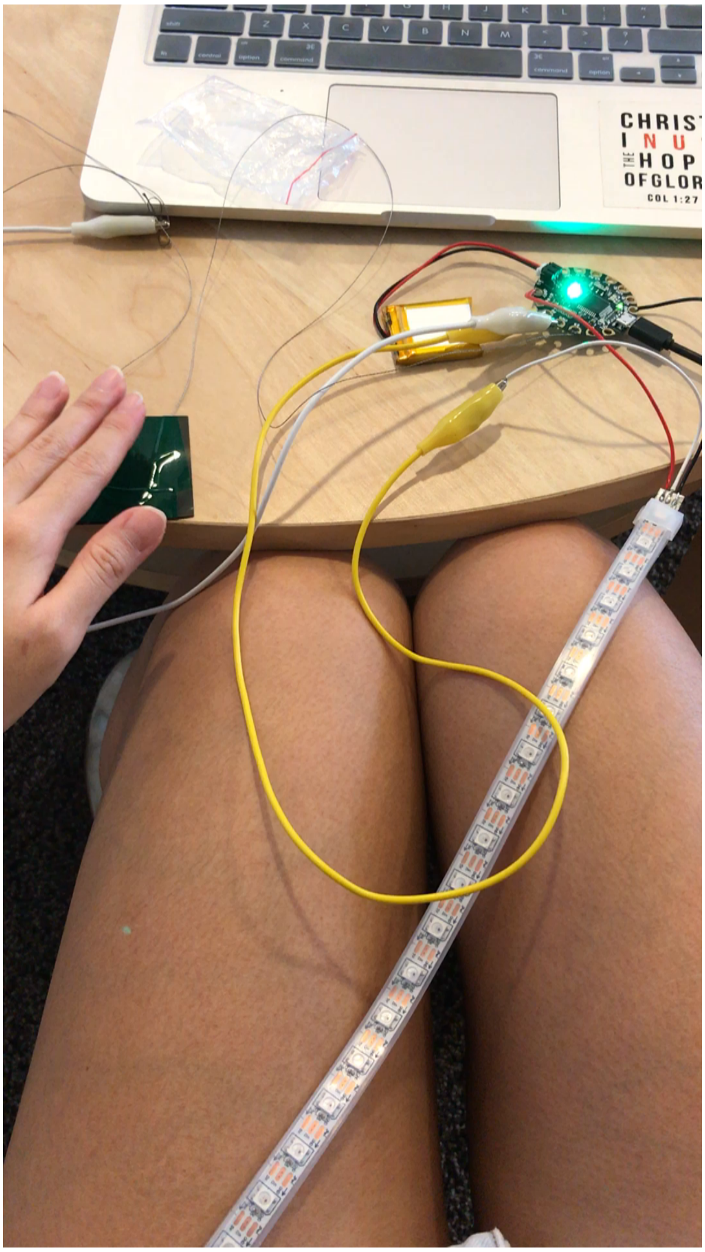

Figure 2: Testing of velostat pressure sensor

The velostat sensor is made by taping two strands of conductive thread on opposite sides of a piece of velostat. The conductive thread is looped many times on each side to increase the number of intersections it has with the other thread on the opposite side of the velostat. It is at these intersections that electricity passes through when resistance decreases (due to pressure applied).

Instead of placing the velostat pressure sensor at the heel of the shoe sole, as shown in the circuit diagram, I placed my sensor in the front portion of my shoe, as I wanted to control the lights using the movement on the balls of my feet and my toes. For instance, I only wanted the LED lights to go off when high pressure was exerted while I was jumping on the balls of my feet or when I was on tiptoe. Placing the sensor at the heel of my feet would cause it to go off every step I took, which was not what I wanted.

Next, I marked out the areas on my shoe which bent with my foot movements, in order to replace these bendy parts with individual LED pixels, instead of the LED light strip. This is to allow the LED light strip to be flexible and not break due to vigorous movement. I soldered the individual pixels together with extra-long portions of silicon-stranded wire to provide ample slack for lateral bending (refer to figure below).

Figure 3: Individually soldered LED pixels at the bend of my shoe

The LED strip is then connected to the FLORA microcontroller with three wires in this way:
NeoPixel strip DIN (data) -> FLORA pin D6
NeoPixel strip +5V (power) -> FLORA VBATT
NeoPixel strip GND (ground) -> FLORA GND

After testing the circuit and ensuring that it works, I inserted the LED strip into its rubber casing and glued it to the bottom of my shoe.

Finally, the lithium polymer (LiPo) battery, used to power the circuit, is plugged into FLORA's JST connecter and tucked under my shoelaces.

This concludes the hardware setup. The colour and speed of the LED lights are determined with the Arduino code, which is programmed into the FLORA microcontroller. All code is attached in the appendix for reference.

Interaction With The Work

For my project, one person is able to wear the shoes at one point of time. The movement of the person's feet will determine the color and speed of the LED light displayed on the shoes. I created the shoes with dancers in mind as I felt that it would be great to be able to transform dancers' motions and subtle movements of their feet into a visual output of LED display that can be seen by the audience. Oftentimes, dancers' feet have many small variations of movement that cannot be explicitly seen by the audience. My project allows the pressure exerted on the soles of a dancer's foot to be expressed visually in the form of LED light patterns and color.

The LED lights are programmed to give out different colors when different pressure ranges are hit. The lightest pressure measured on the velostat sensor will produce white colored LED light. The right shoe is programmed to have a slower appearance (and disappearance) of white light than the left shoe.

Based on the music played, I danced and pressed the balls of my feet at different intensities and varied my exertion of weight, in order to produce a range of colors on my shoes. The audience would then be able to see the lights vary according to the music and dance steps. Short videos are attached to show how the shoes work (video files attached as Showcase1.MOV and Showcase2.MP4).

Future Work

There are many ideas and avenues to explore with what I have created. Firstly, I would like to further enhance the visual output that my shoes produce. Since the pixels in the shoe can be individually controlled and programmed, I could perhaps program individual pixels to be on or off, as well as program different colours for each pixel. Furthermore, the visual appeal of LED lights could be made even more attractive in the dark. A choreographed performance could e held in a dark space, which would showcase the LED lights even more beautifully.

Secondly, I could place more velostat pressure sensors in my shoe, perhaps one at the front of the sole and one at the back. This would allow movements on different parts of my feet to be captured, producing more types of output and allowing the shoes to have greater interaction with the dancer.

Thirdly, other types of sensors could also be used on these shoes. For instance, my initial idea included attaching an accelerometer to the shoes as well. As the dancer moved faster or goes upside down, I could program for different LED light displays. Unfortunately, it was too expensive to include the accelerometer as well so I stuck with the velostat pressure sensors for this project. However, for future work, this could very well be incorporated.

In addition, there are a multitude of opportunities to be tried out with regard to using sensors to enhance the art form of dance. In the process of brainstorming for ideas, I thought of how flex sensors could be placed in a dancer's costume (at the elbows or shoulders etc) to transform their bodily movements into some form of output, whether visual, audio or sensory. Furthermore, an entire dance crew could wear such shoes and choreograph a synced routine for a dynamic performance. The possibilities are indeed endless.

Ultimately, dance is meant to be expressive. By transforming the movements of dancers into unconventional kinds of outputs, we could very well redefine the art of dance.

Appendix

References

- https://learn.adafruit.com/firewalker-led-sneakers/overview
- https://learn.adafruit.com/gemma-led-sneakers/overview
- https://vimeo.com/114767889
- https://www.wired.com/2014/12/high-tech-dance-performance-melds-human-bodies-code/
- http://www.thisiscolossal.com/2015/01/pixel-a-mesmerizing-dance-performance-incorporating-digital-projection/
- https://blog.adafruit.com/2014/03/19/flora-vs-gemma-which-board-should-i-use-wearablewednesday/

Code

<u>Left Shoe</u>

```
#include <Adafruit_NeoPixel.h>

const int analogInPin = A9;  // Analog input pin that the potentiometer is
attached to
Adafruit_NeoPixel strip = Adafruit_NeoPixel(33, 6, NEO_GRB +
NEO_KHZ800);
int sensorValue = 0;        // value read from the pot

const int numReadings = 20;

int readings[numReadings];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
int total = 0;                  // the running total
int average = 0;                // the average

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  pinMode(9, INPUT_PULLUP);
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }

}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // print the results to the serial monitor:
  Serial.print("sensor = " );
```

```
Serial.println(sensorValue);

// subtract the last reading:
total = total - readings[readIndex];
// read from the sensor:
readings[readIndex] = sensorValue;
// add the reading to the total:
total = total + readings[readIndex];
// advance to the next position in the array:
readIndex = readIndex + 1;

// if we're at the end of the array...
if (readIndex >= numReadings) {
  // ...wrap around to the beginning:
  readIndex = 0;
}

// calculate the average:
average = total / numReadings;
/*Serial.print("average = " );
Serial.println(average);
int redValue = map(average,127,40,0,255);
Serial.print("scaled = " );
Serial.println(redValue);

for (uint16_t i = 0; i < strip.numPixels(); i++) {
  strip.setPixelColor(i, strip.Color(0,0,0));
  strip.show();
}
delay(10);*/
if (sensorValue < 230 && sensorValue > 200) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(255, 200, 50), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 200 && sensorValue > 190) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(0, 255, 0), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 190 && sensorValue > 180) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(0, 255, 40), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 180 && sensorValue > 170) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(204, 204, 255), 5);
```

```
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 170 && sensorValue > 160) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(50, 0, 13), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 160 && sensorValue > 150) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(0, 255, 30), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 150 && sensorValue > 140) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(255, 0, 255), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 140 && sensorValue > 130) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(0, 102, 102), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 130 && sensorValue > 120) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(51, 255, 255), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 120) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(0, 0, 255), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
}

void colorWipe(uint32_t c, uint8_t wait) {
  for (uint16_t i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}
```

Right Shoe

```
#include <Adafruit_NeoPixel.h>

const int analogInPin = A9;  // Analog input pin that the potentiometer is
attached to
```

```
Adafruit_NeoPixel strip = Adafruit_NeoPixel(33, 6, NEO_GRB +
NEO_KHZ800);
int sensorValue = 0;        // value read from the pot

const int numReadings = 20;

int readings[numReadings];      // the readings from the analog input
int readIndex = 0;              // the index of the current reading
int total = 0;                  // the running total
int average = 0;                // the average

void setup() {
  // initialize serial communications at 9600 bps:
  Serial.begin(9600);
  pinMode(9, INPUT_PULLUP);
  strip.begin();
  strip.show(); // Initialize all pixels to 'off'

  for (int thisReading = 0; thisReading < numReadings; thisReading++) {
    readings[thisReading] = 0;
  }

}

void loop() {
  // read the analog in value:
  sensorValue = analogRead(analogInPin);
  // print the results to the serial monitor:
  Serial.print("sensor = " );
  Serial.println(sensorValue);

  // subtract the last reading:
  total = total - readings[readIndex];
  // read from the sensor:
  readings[readIndex] = sensorValue;
  // add the reading to the total:
  total = total + readings[readIndex];
  // advance to the next position in the array:
  readIndex = readIndex + 1;

  // if we're at the end of the array...
  if (readIndex >= numReadings) {
    // ...wrap around to the beginning:
    readIndex = 0;
  }

  // calculate the average:
  average = total / numReadings;
```

```
/*Serial.print("average = " );
Serial.println(average);
int redValue = map(average,127,40,0,255);
Serial.print("scaled = " );
Serial.println(redValue);

for (uint16_t i = 0; i < strip.numPixels(); i++) {
  strip.setPixelColor(i, strip.Color(0,0,0));
  strip.show();
}
delay(10);*/
if (sensorValue < 290 && sensorValue > 260) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(255, 200, 50), 10);
  colorWipe(strip.Color(0, 0, 0), 10);
  }
  else if (sensorValue < 260 && sensorValue > 250) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(255, 51, 153), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 250 && sensorValue > 240) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(255, 0, 127), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 240 && sensorValue > 230) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(102, 255, 255), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 230 && sensorValue > 220) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(0, 76, 153), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 220 && sensorValue > 210) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(255, 178, 102), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 210 && sensorValue > 200) {
  Serial.println("leds triggered");
  colorWipe(strip.Color(255, 153, 153), 5);
  colorWipe(strip.Color(0, 0, 0), 3);
  }
  else if (sensorValue < 200 && sensorValue > 190) {
  Serial.println("leds triggered");
```

```
    colorWipe(strip.Color(153, 0, 76), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 190 && sensorValue > 180) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(255, 128, 0), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 180 && sensorValue > 170) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(0, 102, 102), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 170 && sensorValue > 160) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(250, 20, 147), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
    else if (sensorValue < 160) {
    Serial.println("leds triggered");
    colorWipe(strip.Color(255, 0, 0), 5);
    colorWipe(strip.Color(0, 0, 0), 3);
    }
}

void colorWipe(uint32_t c, uint8_t wait) {
  for (uint16_t i = 0; i < strip.numPixels(); i++) {
    strip.setPixelColor(i, c);
    strip.show();
    delay(wait);
  }
}
```

Parts List

- 2 FLORA main boards
- Velostat
- 2 meters NeoPixel 60-LED strip in white
- 2 150mAh LiPo battery
- Conductive thread
- Standard thread
- Silicone coated stranded wires
- Tape
- Super strong glue
- Shoes