POGOLITICAL

GIVE SOME HARD-HITTING SUPPORT TO YOUR FAVORITE CANDIDATE.

Pogolitical uses the force and direction of your pogo-stick bounces to give you a cathartic release for your political frustrations, handing you the power to deliver some justice to your most-hated politician. As you and your friends bounce, preferences are recorded as an innovative polling approach. Tweets from critics appear in real time, fueling your hatred and augmenting your experience. You'll be jumping at the chance to support your favorite candidate!

Corten Singer   Lydia Gilbert   Eugene Song   Erica Yin

POGOLITICAL

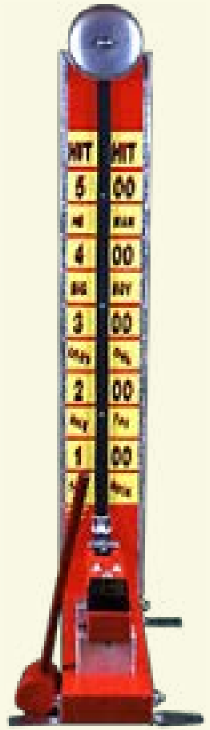
TRANSPORTATION TECHNOLOGY INVESTIGATION

Corten Singer Lydia Gilbert Eugene Song Erica Yin

EXPLORATION OF TRANSPORTATION METHODS

We wanted to choose an out-of-the-ordinary method of transportation with a visceral, physical impact.

Pogo sticks are not used in daily life, so we decided to give them a new context outside of a children’s toy. We wanted to capture and attach meaning to the intense, energetic, physical experience and very literal “impact” of pogo-ing.



INSPIRATION, MOTIVATION, AND INITIAL IDEA

With the current political situation in the US, many voters feel powerless- as though their impact is too small. We wanted to give them more physical power as an outlet for their political frustration.

We also drew inspiration from the arcade game shown above, in which the force of a hard hammer hit results in striking a bell at the top of the device.

In our installation, the force and "political lean" of a pogo jump translate to comically striking a divisive politician, capturing both the impassioned aggression of recent rallies, as well as giving cathartic release to the stifled anger of disillusioned voters.

Corten Singer Lydia Gilbert Eugene Song Erica Yin

OBSERVATION AND DESIGN PROCESS DOCUMENTATION

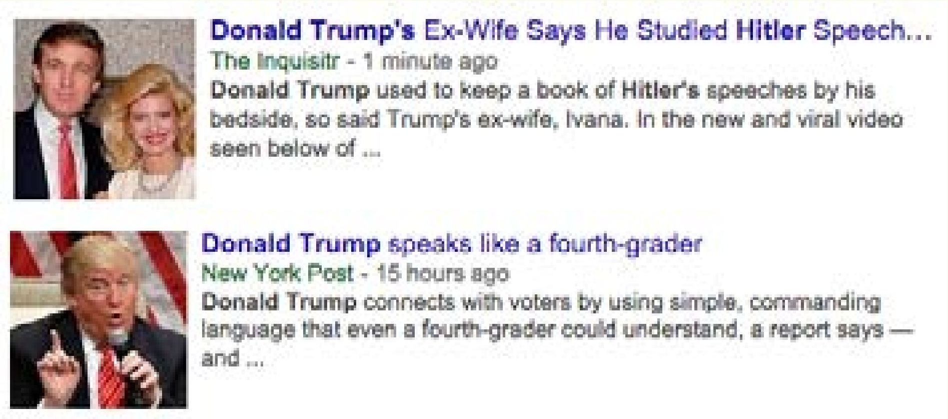
POGOLITICAL

OBSERVATIONS

We noticed that political rallies for the upcoming presidential election were becoming unusually violent, especially against Donald Trump. We wanted to make a product around that aggression and the idea of "moving" forward as a country. We considered how to give frustrated, scared voters an outlet for their emotions.

A pogo stick gave that energetic, visceral feeling of power and satisfaction. We decided to use an accelerometer to measure the "political lean" of the pogo stick and translate that into a meaningful, entertaining, and comedic release directed at your most hated politican.

We thought having a robot arm giving a much needed smack to Trump or Bernie would be a great solution to voter frustration and anger.



Donald Trump's Ex-Wife Says He Studied **Hitler** Speech...

The Inquisitr - 1 minute ago

Donald Trump used to keep a book of **Hitler's** speeches by his bedside, so said Trump's ex-wife, Ivana. In the new and viral video seen below of ...

Donald Trump speaks like a fourth-grader

New York Post - 15 hours ago

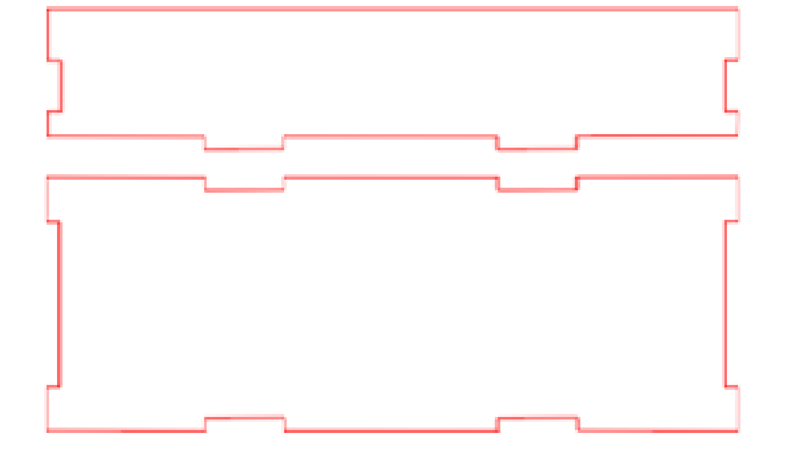
Donald Trump connects with voters by using simple, commanding language that even a fourth-grader could understand, a report says — and ...

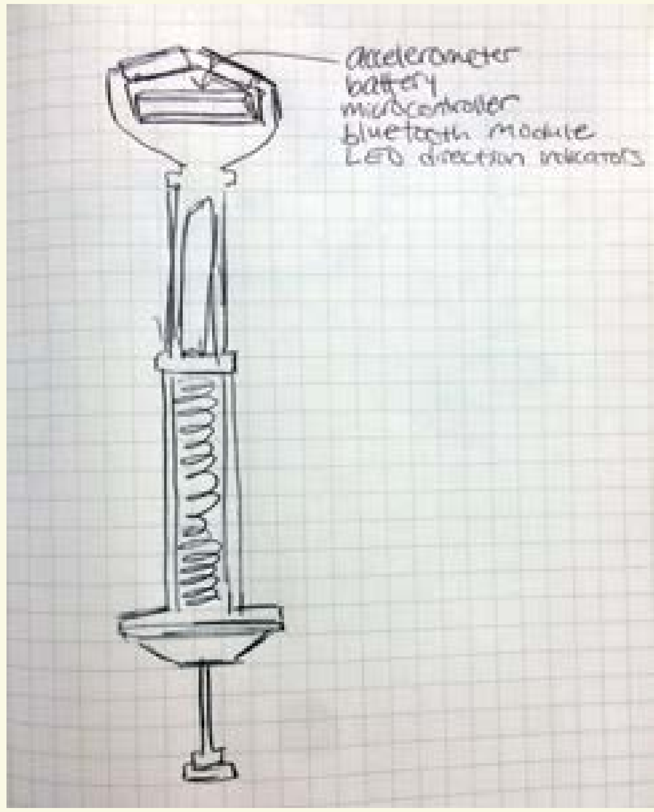
INITIAL IDEATION AND SKETCHES

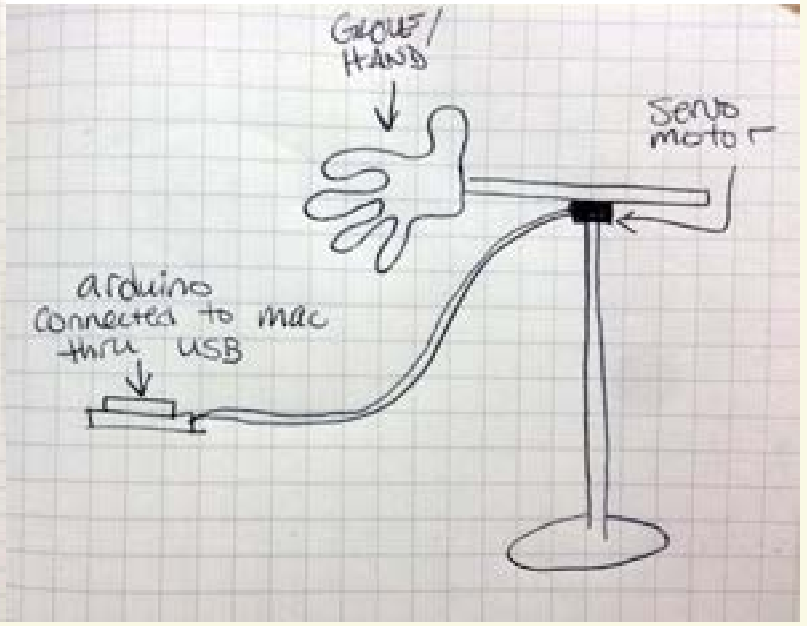
We decided that the fastest way to send accelerometer data to the robot arm was through bluetooth, and then to connect the arm to a Mac serially.

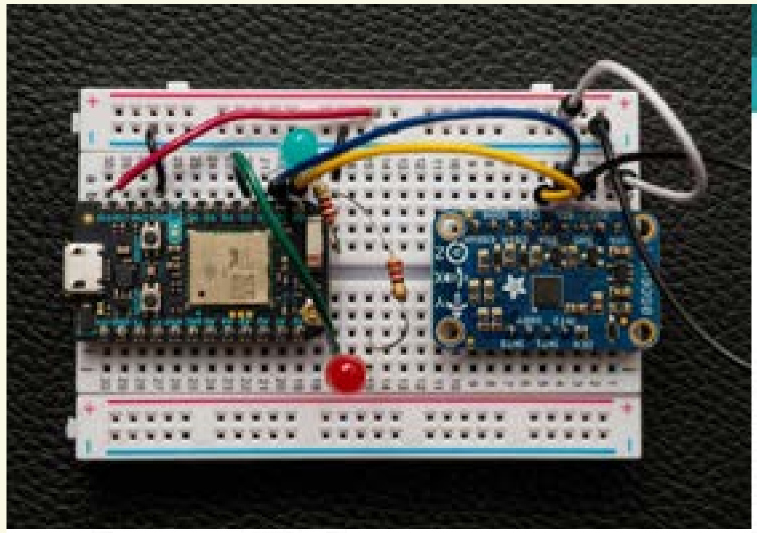
We also considered where on the pogo stick to mount the accelerometer, battery, bluetooth module, LEDs, and microcontroller. We decided to place the breadboard below the hanndlebar and cover it with acrylic to protect it.

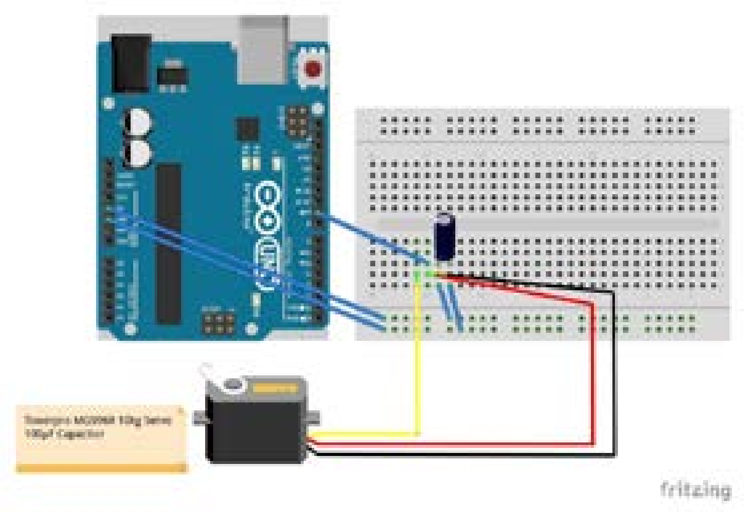
We used a servo motor mounted on a tripod for a stable stand for the robot arm to be at the correct height with the projector display.

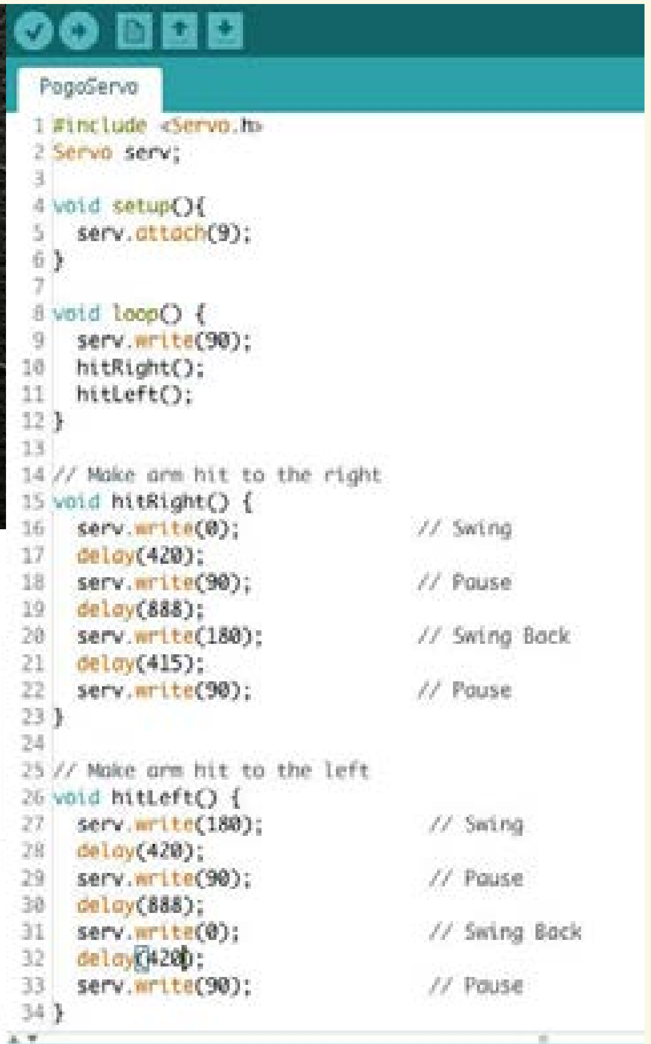




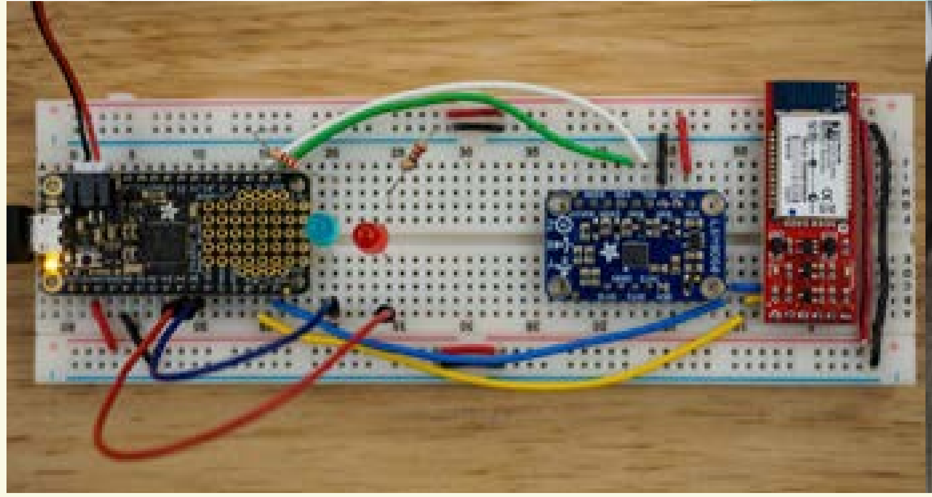


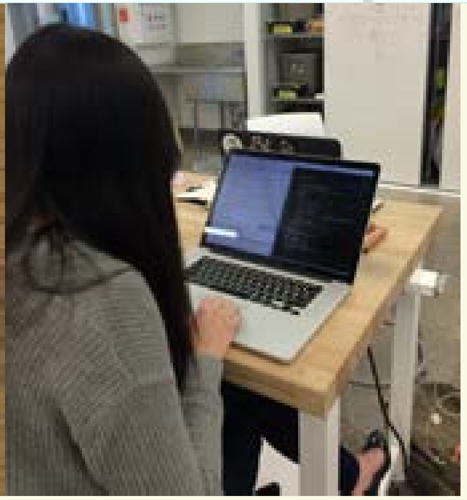




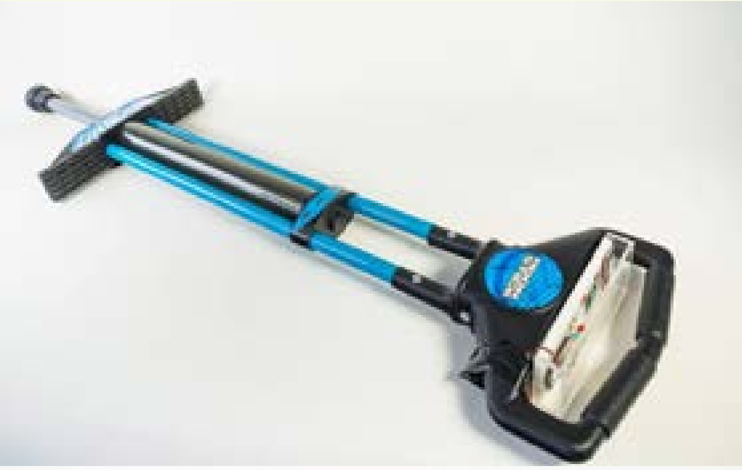


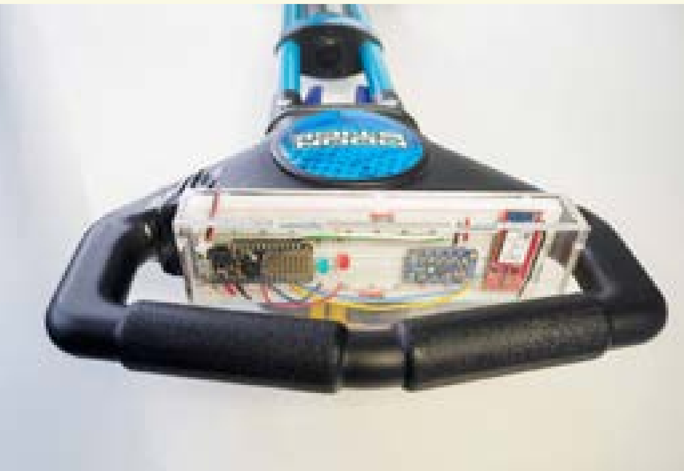
```
#include <Servo.h>
Servo serv;

void setup(){
  serv.attach(9);
}

void loop() {
  serv.write(90);
  hitRight();
  hitLeft();
}

// Make arm hit to the right
void hitRight() {
  serv.write(0);                // Swing
  delay(420);
  serv.write(90);               // Pause
  delay(888);
  serv.write(180);              // Swing Back
  delay(415);
  serv.write(90);               // Pause
}

// Make arm hit to the left
void hitLeft() {
  serv.write(180);               // Swing
  delay(420);
  serv.write(90);                // Pause
  delay(888);
  serv.write(0);                 // Swing Back
  delay(420);
  serv.write(90);                // Pause
}
```

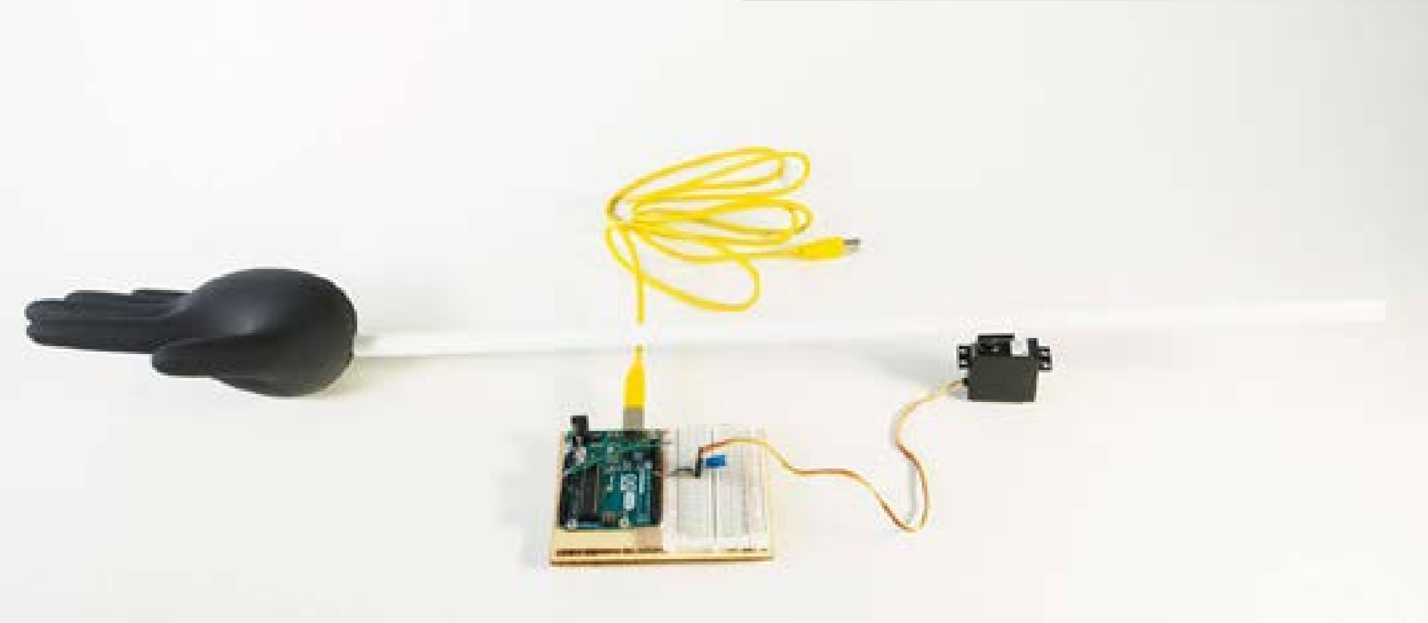




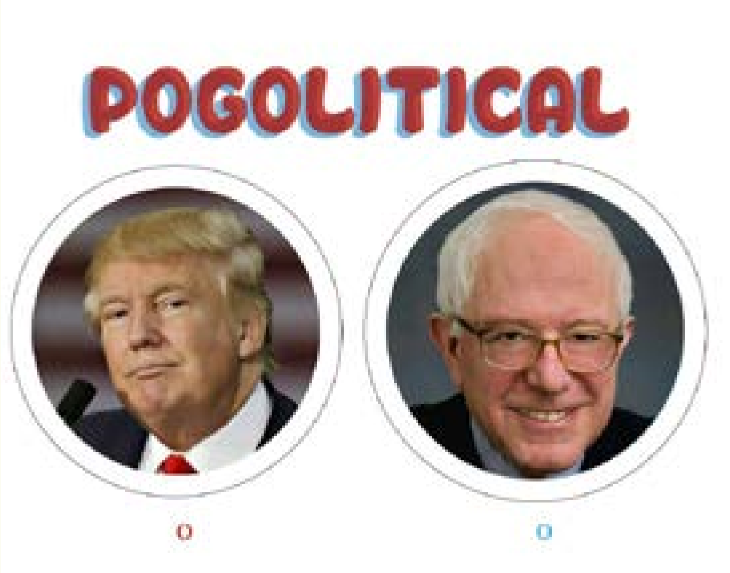
BUILDING, ASSEMBLY, AND CREATION





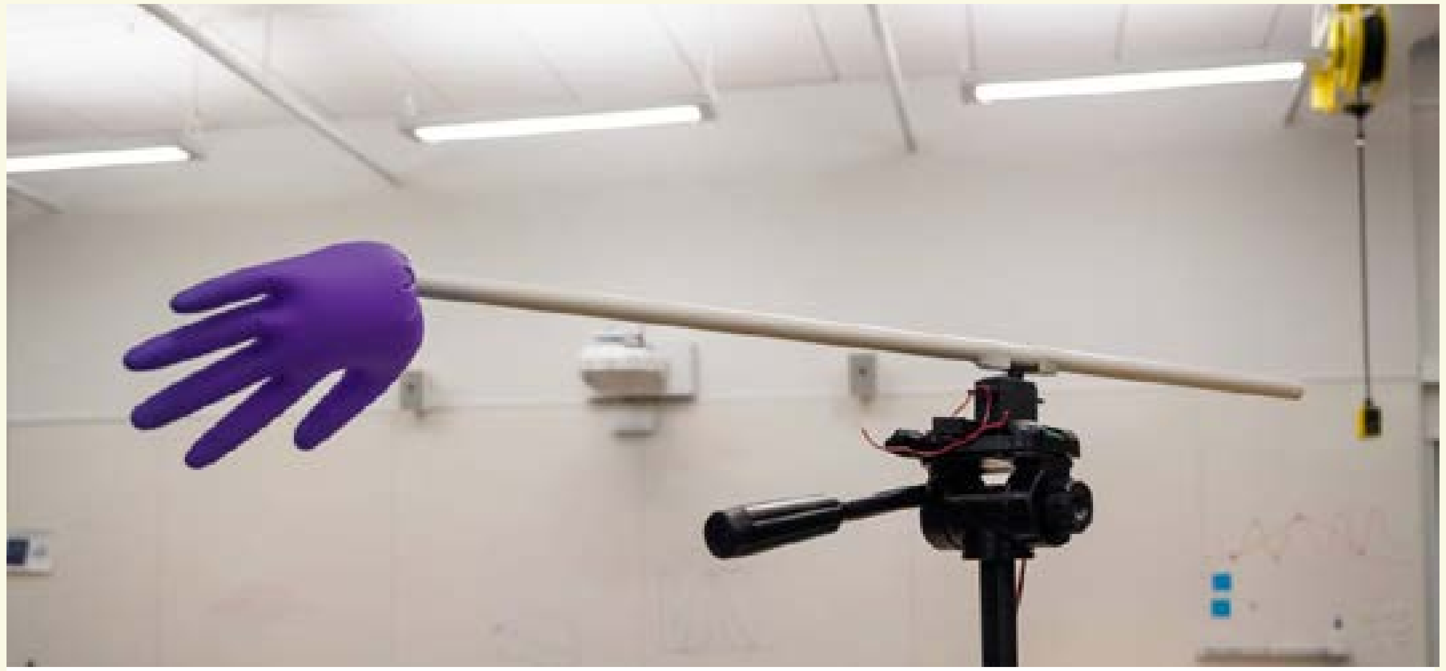


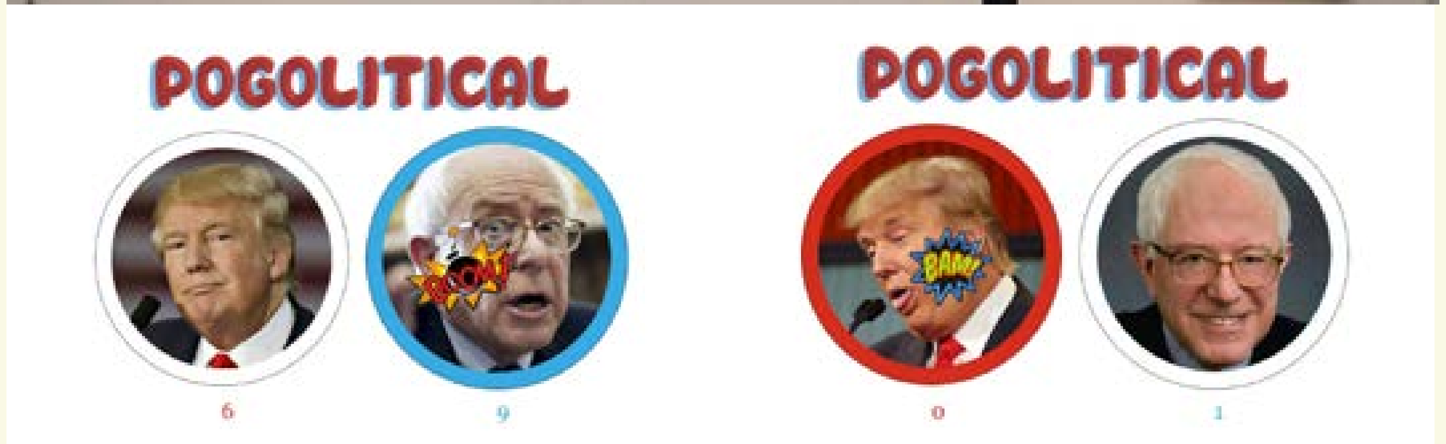
This is the final, functioning control mounted on the pogo stick, and the robot arm using the lightest "hand" (a plastic glove) possible. On the right is the initial state of our interface.

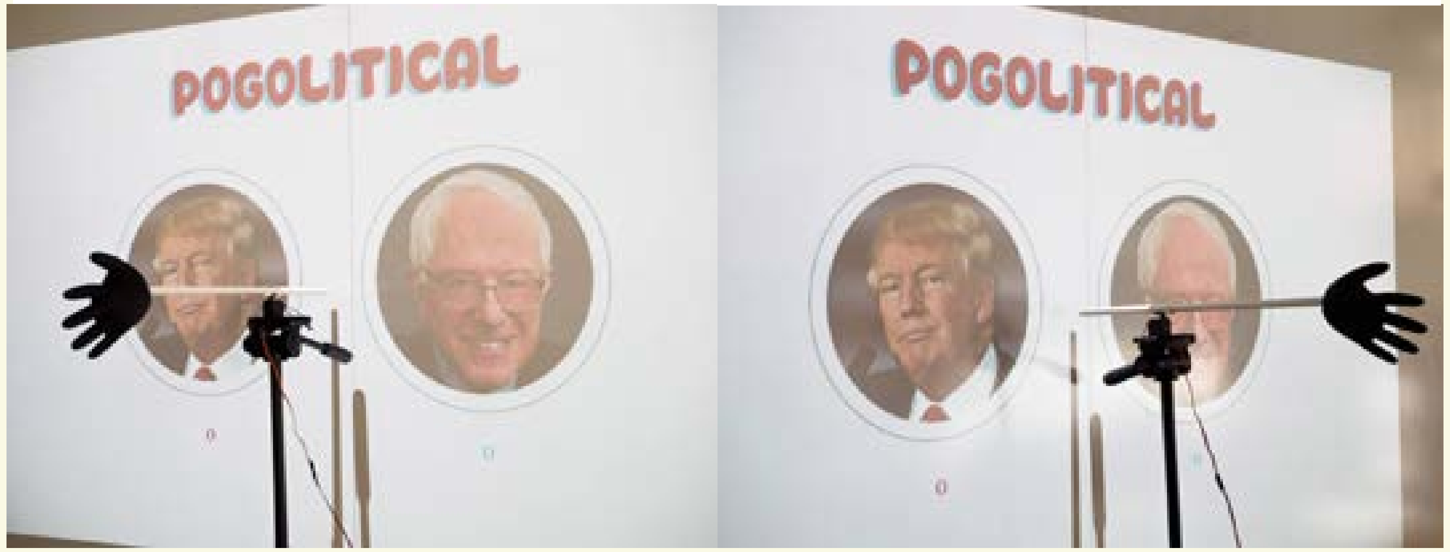


PROTOTYPE

IN ACTION







POGOLITICAL

STEP-BY-STEP PROCESS

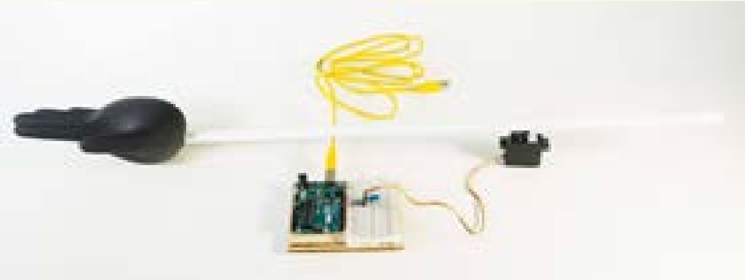
Corten Singer   Lydia Gilbert   Eugene Song   Erica Yin

SUPPLIES

- Pogostick
- 1 Fullsized Breadboard
- 1 Halfsized Breadboard
- Arduino Uno
- Adafruit Feather M0 Basic Proto
- Adafruit 9-DOF Accel/Mag/Gyro Breakout Board
- SparkFun Bluetooth Mate Silver
- Towerpro MG996R 10g Servo Motor
- 100µF Capacitor
- Jumper Wires (female/male)
- A-B USB cable
- LiPo battery
- LED lights & some 220 Ω resistors (optional)
- Camera Tripod (or other stand)
- Software:
  - Arduino (https://www.arduino.cc/en/Main/Software)
  - Processing (https://processing.org/download)
- Projector

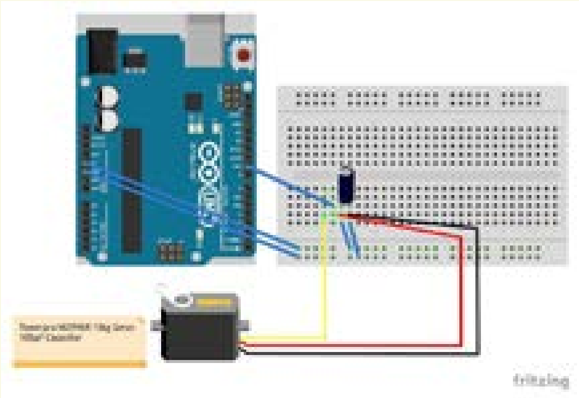
STEP 0 - BUILD AN ARM

- You want to select something sturdy and lightweight
- Cardboard works very well for prototypes, or even final models
- We used a thin plastic tubing for its sturdiness
- We used a Nitrile glove as an inflatable, lightweight boxing glove
- Secure the base of your stick to the motor attachment
- A counterweight can help maintain the arm over time



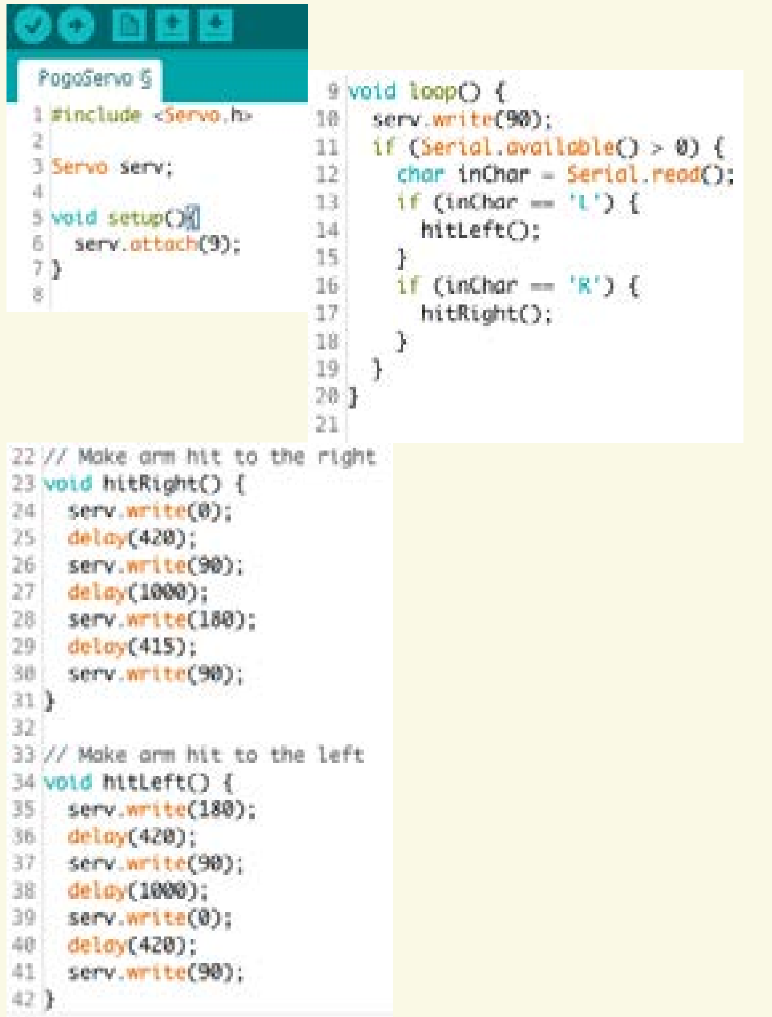
STEP 1 - SET UP CIRCUIT

- Hook up a circuit with your Arduino Uno and a Half-Sized breadboard
- Use a wire connecting the ~9 as an output
- Hook up the yellow wire to this ~9 output
- Connect the red wire to a voltage source of 4.8-6V (Arduino provides 5V output)
- Connect the black to ground



STEP 2 - ARDUINO CODE

- In the arduino IDE, setup a new sketch with the Servo.h header file and Servo object and attach it to pin 9
- Use helper functions in line 22-42 to control your punching arm
- You will need to tune the delay times to the weight and dimensions of your arm (this varies a lot)
- Repeatedly loop and check if there is incoming serial data from the Processing program
- Receiving the character 'R' indicates a Right hit, 'L' indicates a Left hit



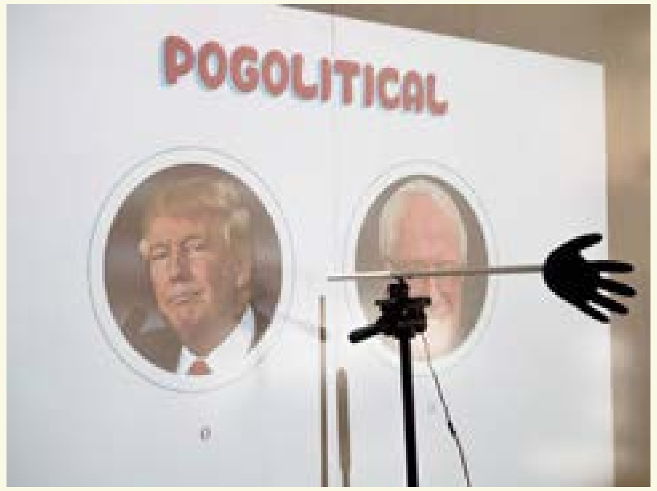
```
1 #include <Servo.h>
2
3 Servo serv;
4
5 void setup(){
6   serv.attach(9);
7 }
8
```

```
9 void loop() {
10   serv.write(90);
11   if (Serial.available() > 0) {
12     char inChar = Serial.read();
13     if (inChar == 'L') {
14       hitLeft();
15     }
16     if (inChar == 'R') {
17       hitRight();
18     }
19   }
20 }
21
```

```
22 // Make arm hit to the right
23 void hitRight() {
24   serv.write(0);
25   delay(420);
26   serv.write(90);
27   delay(1000);
28   serv.write(180);
29   delay(415);
30   serv.write(90);
31 }
32
33 // Make arm hit to the left
34 void hitLeft() {
35   serv.write(180);
36   delay(420);
37   serv.write(90);
38   delay(1000);
39   serv.write(0);
40   delay(420);
41   serv.write(90);
42 }
```

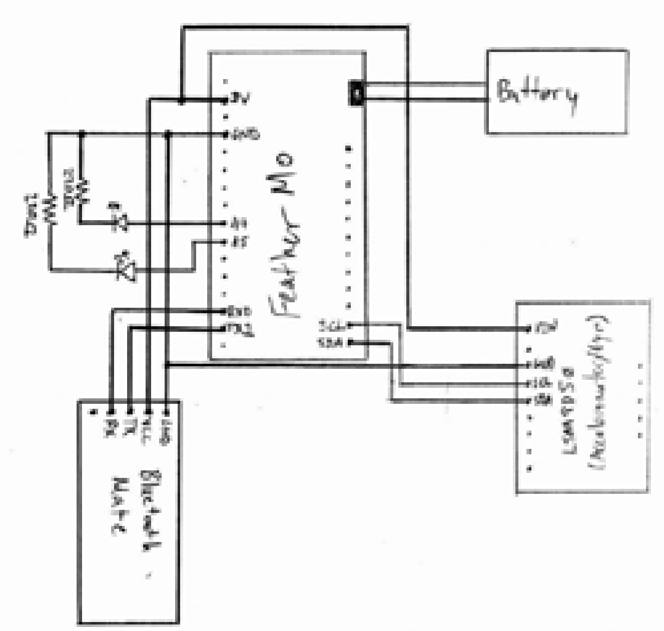
STEP 3 - SET UP STAND

- We used a Camera Tripod that clipped the motor in place and allowed the arm to swing securely
- We recommend something very sturdy and heavier than the motor/arm combo to prevent sway
- Using jumper wires, connect the wires of the motor to the Arduino circuit from above



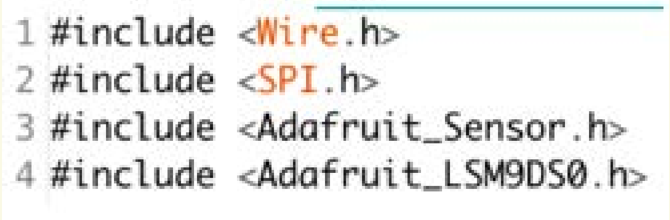
STEP 4 - POGO HARDWARE

- Hook up the Feather M0, accelerometer, and Bluetooth module together with some LEDs.
- Power the circuit through the Feather's 3V and GND pins, i.e. connect all of the GND pins from the three boards together, and connect the accelerometer VIN and Bluetooth module VCC to 3V.
- Connect the SCL and SDA pins from the accelerometer to the SCL and SDA pins of the Feather, respectively.
- Connect the TX and RX pins from the Bluetooth module to the RX0 and TX1 pins on the Feather, respectively.
- Connect an LED to A4 on the Feather at one end, and a 220 ohm resistor at the other. Ground the other end of the resistor.
- Do the same with an LED connected to A5 on the Feather.

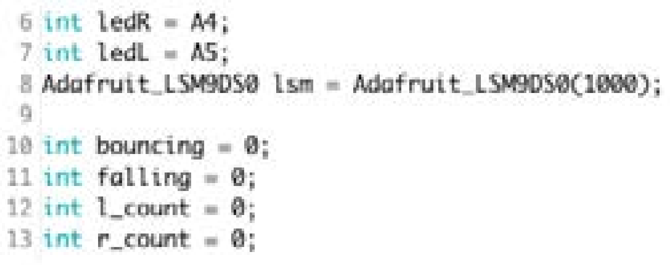


STEP 5 - POGO CODE

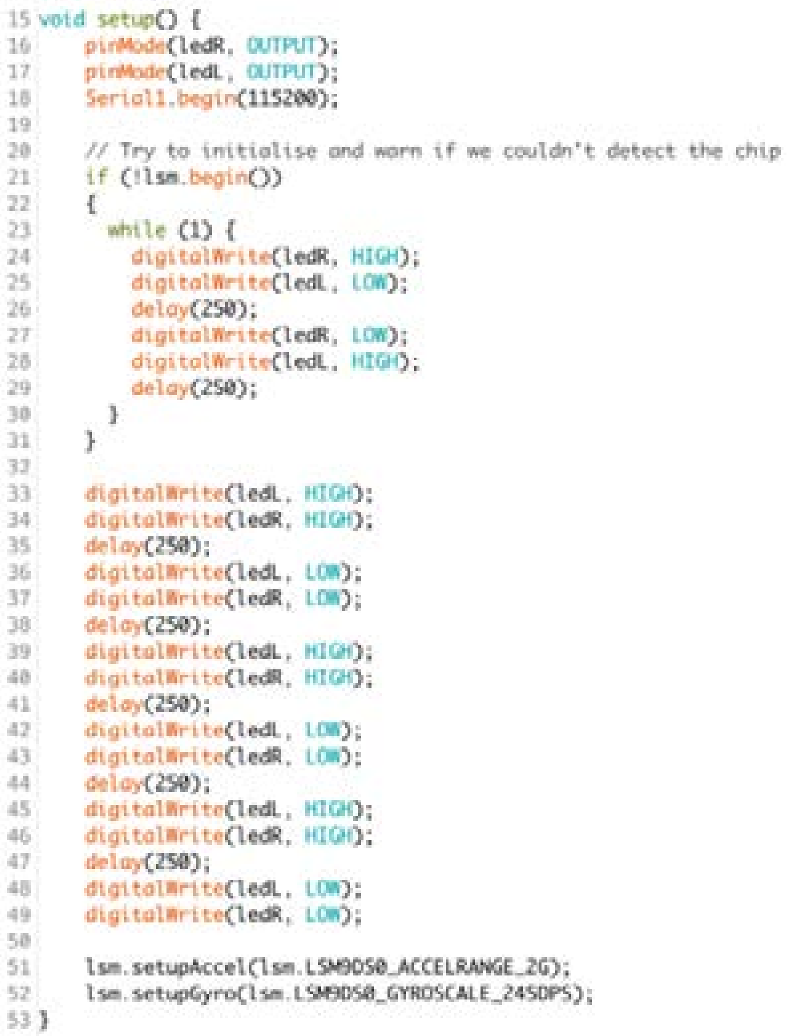
- Open the Arduino IDE.
- Use these include statements for easy communication with the accelerometer (lines 1-4).
- Initialize these variables (lines 6-13).
- Use this setup() routine (lines 15-53). It initializes communication with the accelerometer and serial communication with the Bluetooth module. It will blink the LEDs to indicate success or failure (alternating blinks are failure, 3 simultaneous blinks indicate success).
- Use this loop() routine (lines 55-121). It will sense bounces and bounce direction, then send the direction over serial to the Bluetooth module. Bounces are detected by alternations between near-zero and near-gravitational vertical accelerations. Directions are detected using the gyroscope and rotation along the Y-axis.



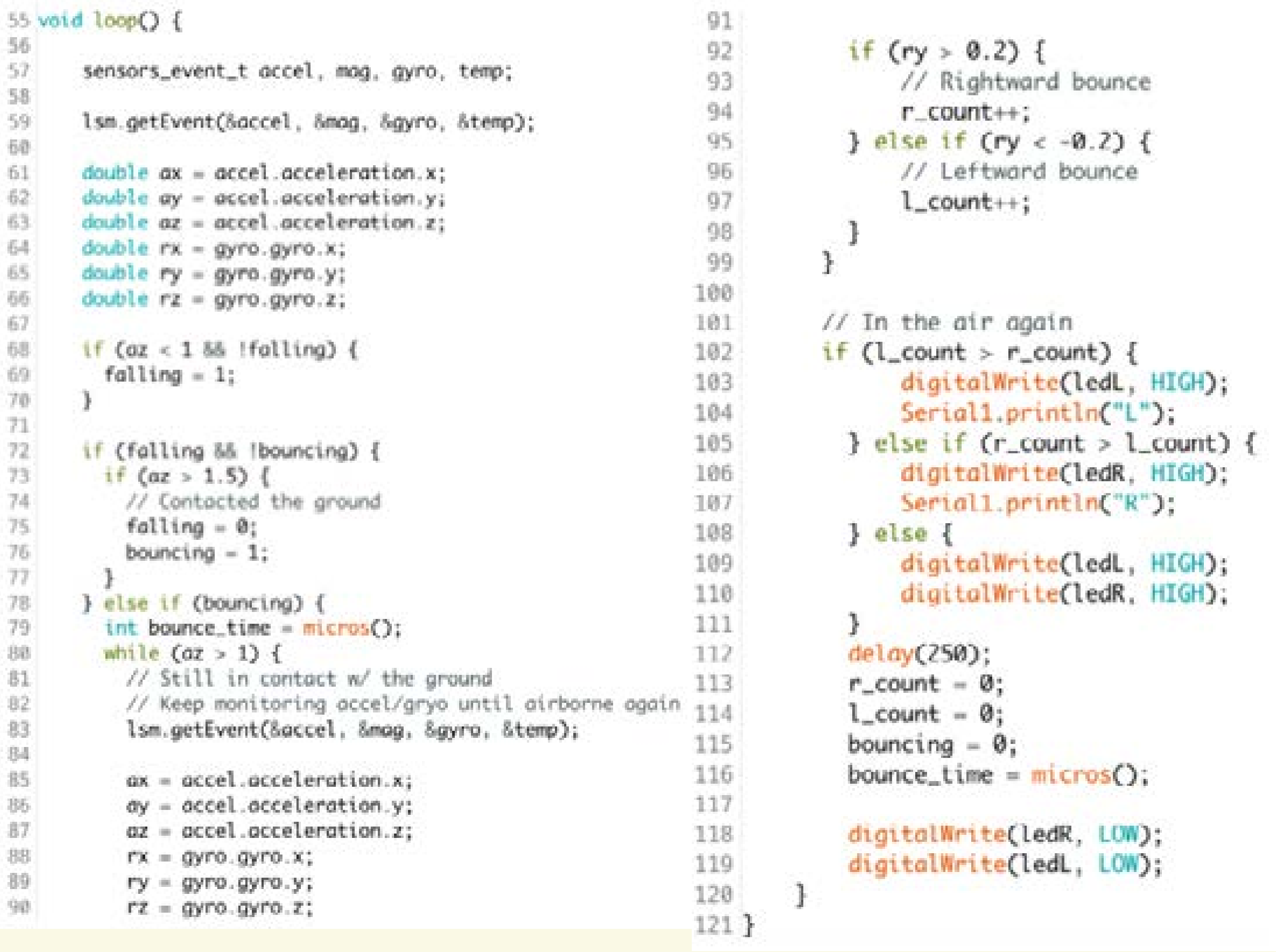
```
1 #include <Wire.h>
2 #include <SPI.h>
3 #include <Adafruit_Sensor.h>
4 #include <Adafruit_LSM9DS0.h>
```



```
6 int ledR = A4;
7 int ledL = A5;
8 Adafruit_LSM9DS0 lsm = Adafruit_LSM9DS0(1000);
9
10 int bouncing = 0;
11 int falling = 0;
12 int l_count = 0;
13 int r_count = 0;
```



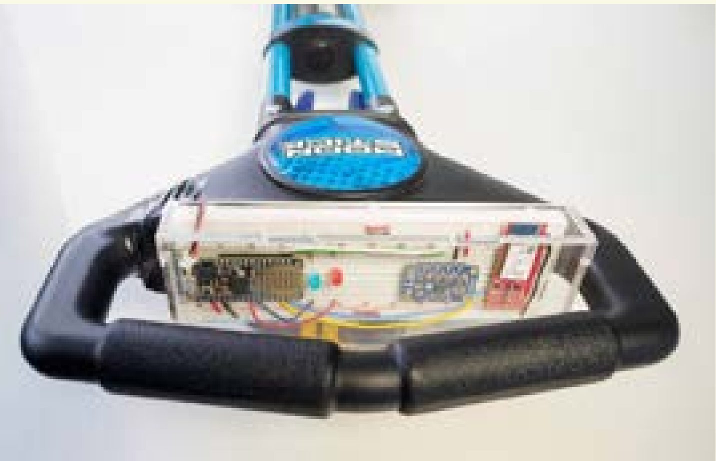
```
15 void setup() {
16     pinMode(ledR, OUTPUT);
17     pinMode(ledL, OUTPUT);
18     Serial1.begin(115200);
19
20     // Try to initialise and warn if we couldn't detect the chip
21     if (!lsm.begin())
22     {
23       while (1) {
24         digitalWrite(ledR, HIGH);
25         digitalWrite(ledL, LOW);
26         delay(250);
27         digitalWrite(ledR, LOW);
28         digitalWrite(ledL, HIGH);
29         delay(250);
30       }
31     }
32
33     digitalWrite(ledL, HIGH);
34     digitalWrite(ledR, HIGH);
35     delay(250);
36     digitalWrite(ledL, LOW);
37     digitalWrite(ledR, LOW);
38     delay(250);
39     digitalWrite(ledL, HIGH);
40     digitalWrite(ledR, HIGH);
41     delay(250);
42     digitalWrite(ledL, LOW);
43     digitalWrite(ledR, LOW);
44     delay(250);
45     digitalWrite(ledL, HIGH);
46     digitalWrite(ledR, HIGH);
47     delay(250);
48     digitalWrite(ledL, LOW);
49     digitalWrite(ledR, LOW);
50
51     lsm.setupAccel(lsm.LSM9DS0_ACCELRANGE_2G);
52     lsm.setupGyro(lsm.LSM9DS0_GYROSCALE_245DPS);
53 }
```



```
void loop() {

    sensors_event_t accel, mag, gyro, temp;

    lsm.getEvent(&accel, &mag, &gyro, &temp);

    double ax = accel.acceleration.x;
    double ay = accel.acceleration.y;
    double az = accel.acceleration.z;
    double rx = gyro.gyro.x;
    double ry = gyro.gyro.y;
    double rz = gyro.gyro.z;

    if (az < 1 && !falling) {
      falling = 1;
    }

    if (falling && !bouncing) {
      if (az > 1.5) {
        // Contacted the ground
        falling = 0;
        bouncing = 1;
      }
    } else if (bouncing) {
      int bounce_time = micros();
      while (az > 1) {
        // Still in contact w/ the ground
        // Keep monitoring accel/gryo until airborne again
        lsm.getEvent(&accel, &mag, &gyro, &temp);

        ax = accel.acceleration.x;
        ay = accel.acceleration.y;
        az = accel.acceleration.z;
        rx = gyro.gyro.x;
        ry = gyro.gyro.y;
        rz = gyro.gyro.z;

        if (ry > 0.2) {
            // Rightward bounce
            r_count++;
        } else if (ry < -0.2) {
            // Leftward bounce
            l_count++;
        }
      }

      // In the air again
      if (l_count > r_count) {
            digitalWrite(ledL, HIGH);
            Serial1.println("L");
        } else if (r_count > l_count) {
            digitalWrite(ledR, HIGH);
            Serial1.println("R");
        } else {
            digitalWrite(ledL, HIGH);
            digitalWrite(ledR, HIGH);
        }
        delay(250);
        r_count = 0;
        l_count = 0;
        bouncing = 0;
        bounce_time = micros();

        digitalWrite(ledR, LOW);
        digitalWrite(ledL, LOW);
    }
}
```

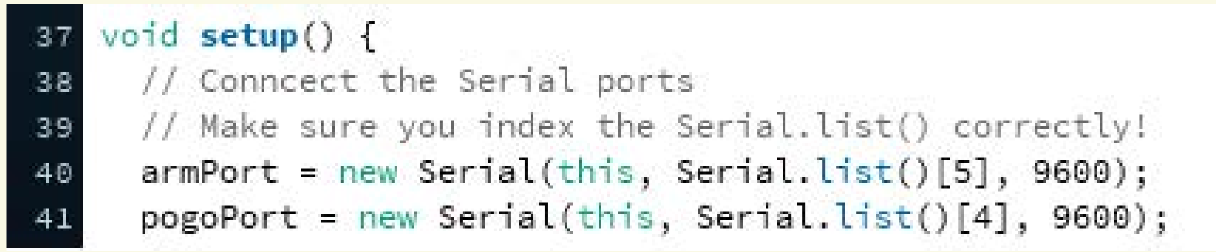
STEP 6 - MOUNTING DEVICE

- We used a laser cutter to cut an acrylic rectangular prism casing, but any protective material could suffice
- Once the case is on, remove the adhesive strip from the breadboard
- Place the breadboard on a safe location of the pogostick

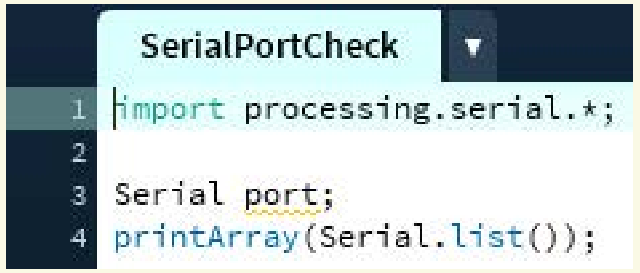


STEP 7 - PROCESSING

- Set up 2 serial port objects (lines 37-41)
    - One port receiving the incoming Bluetooth signals from the PogoStick
    - The other port connected to the Arduino Uno
    - When receiving a jump signal, relay that signal to the motor
- Make sure to run the following serial check before running the whole code (lines 1-4)
    - You MUST see which ports to connect
- After this, you can just upload and run the attached Processing code in the Processing software.



```
37 void setup() {
38   // Conncect the Serial ports
39   // Make sure you index the Serial.list() correctly!
40   armPort = new Serial(this, Serial.list()[5], 9600);
41   pogoPort = new Serial(this, Serial.list()[4], 9600);
```



```
1 import processing.serial.*;
2
3 Serial port;
4 printArray(Serial.list());
```