**Prototype Code 1: Average method**

void setup() {

// initialize serial communication at 9600 bits per second:

Serial.begin(9600);

myservo.attach(9);

}

int Sensor1Max=0, Sensor2Max=0, count=0,servoVal=90;

float ratio=0.0;

// the loop routine runs over and over again forever:

void loop() {

// read the input on analog pin 0:

int sensorValue1 = analogRead(A0);

Serial.println(sensorValue1);

if(sensorValue1>Sensor1Max){

Sensor1Max=sensorValue1;

}

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage1 = sensorValue1 \* (5.0 / 1023.0);

Serial.println(voltage1);

int sensorValue2 = analogRead(A1);

Serial.println(sensorValue2);

if(sensorValue2>Sensor2Max){

Sensor2Max=sensorValue2;

}

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float voltage2 = sensorValue2 \* (5.0 / 1023.0);

// print out the value you read:

if(count>10){

ratio = Sensor1Max/Sensor2Max;

if(ratio>1){

servoVal=179;

}

elseif(ratio<1){

servoVal=0;

}

} //wait for 10 seconds to read the Max input from both players

Serial.println(ratio);

myservo.write(val);

delay(500); //delay for 0.5 seconds

count++;

}

**Prototype Code 2: Continuous Method**

#include <Servo.h>

Servo myservo;

const int Motor1Control = 13;

const int Motor2Control = 12;

void setup() {

Serial.begin(9600);

myservo.attach(Motor1Control);

myservo.attach(Motor2Control);

}

float increment=0.0;

void loop() {

int sensorValue1 = analogRead(A0);

//sensorValue1=map(sensorValue1, 0, 1023, 0, 179);

//analogWrite(Motor1Control, sensorValue1);// Write the value to motor control (real time)

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

int sensorValue2 = analogRead(A1);

//sensorValue2=map(sensorValue2, 0, 1023, 0, 179);

//analogWrite(Motor2Control, sensorValue2);// Write the value to motor control (real time)

// Convert the analog reading (which goes from 0 - 1023) to a voltage (0 - 5V):

float ratio=sensorValue2/sensorValue1;

if(ratio > 5)

ratio=5;

if(ratio<0.2)

ratio=0.2;

if(ratio>1)

increment=16\*(ratio-1);

else if(ratio<=1)

increment=16\*(1/ratio-1);

analogWrite(Motor1Control, int(90+increment));

// Algorithm to check if one of the players won

//when one of the arms reached horizontal level and both arms were in contact

}

State Machine:

Calibration:

need to do:

read in the signals from sensors on both players’ arms

maybe voltage is better on the analog input of Arduino

design an algorithm that put both player’s max input on the same level.

maybe amplify the weaker input

calibration (led on)

...after calibration OK (10 seconds), jump into the [Game play] state.

Game play:

need to do:

read in the signals from sensors on both players’ arms

Raw data: sensorArm1, sensorArm2

apply manipulated current or voltage to motor

...when sensorWin1 or sensorWin2 is activated, jump into state [Win]

Win:

see which sensor is activated

usually we supply a sensor with a constant voltage, and get the current as an output

but since

Prototype code 2: