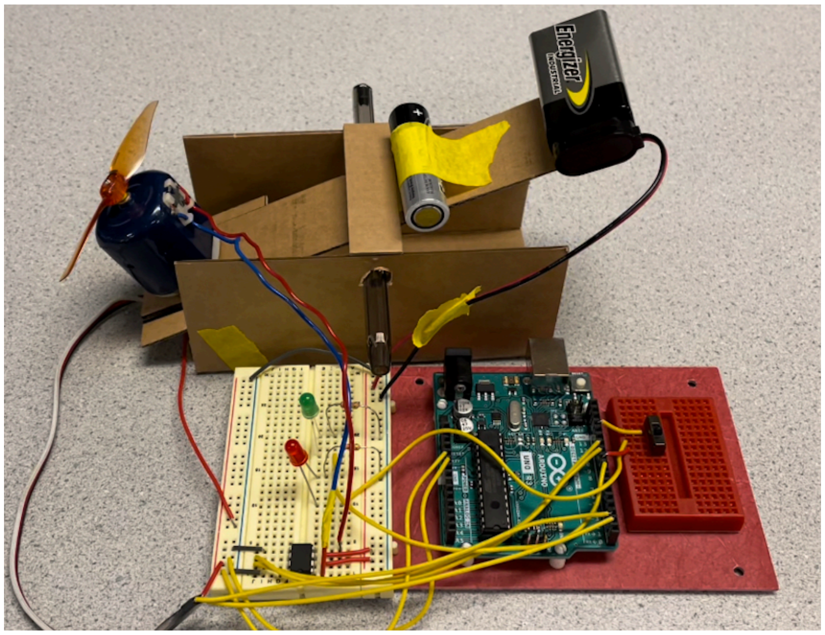
Govind Chari (gmc93)
MAE 3780: Mechatronics
November 2, 2021

Individual Project – Final Report

Overview

This project is an angle controllable seesaw that is intended to demonstrate feedback controls. There is a seesaw structure with a DC motor with a propeller on one end of the arm. This motor spins to generate a force on the seesaw arm to keep it horizonal. The angle of the arm is detected indirectly by using an ultrasonic sensor. This sensor measures the distance from the ground to the motor. If this distance is smaller than nominal for a horizonal arm, then the motor spins faster to lift the arm up more. If the distance is larger than nominal, then the motor decreases its speed. The motor's speed is computed by the Arduino using a PID controller. There are two status LEDs. The green LED turn on when the motor is spinning, and the red LED turn on when the motor is not spinning. There is also a switch to turn the red LED off. I found that the motor does not spin fast enough to generate large amounts of thrust, so for the motor to have the desired control authority, the seesaw should be reasonably close to balanced, which is why the 9V and AA batteries are on the other side of the arm. The 9V battery powers the motor, but the AA battery is just there for balance.



Design Considerations

This project had two main things that I would like to improve on. The first was that the motor did not provide enough thrust and the second is that the ultrasonic sensor does not work very well when measuring distances under three centimeters. Due to the kit motor's size, I thought it would be able to generate an appreciable force, but there seemed to be internal rubbing and other losses that reduced the motor speed.

If I had the same constraints, I would buy a better motor rather than using the kit motor and try to find cheaper propellers instead of spending half of my budget on a pack of propellers.

To solve the problem with the ultrasonic sensor, I would increase the height of the pivot point, so the motor would nominally be much higher than three centimeters off the table.

If I had a larger budget, would firstly buy a larger motor to solve the lack of thrust problem. To attack the ultrasonic sensor problem, I would also raise the pivot point higher, but I would also buy a low friction potentiometer to sense the arm angle. I could combine the potentiometer measurement and ultrasonic sensor measurement in a Kalman filter to get a better estimate of the true angle of the arm. The potentiometer would also allow for an angle measurement of the arm when the ultrasonic sensor is too close to the motor to make a measurement.

Assembly Instructions

1. Draw two lines to divide a 5"x7" piece of cardboard into three parts (Figure 1)
2. Fold along the lines into a U shape and stick a pen (or any other axle-like object) through the cardboard about half an inch from the top (Figure 2)
3. Mark and cut off a 1.5"x7" piece of cardboard to make the arm (Figure 3)
4. Glue another small rectangle of cardboard onto the arm to give the ultrasonic sensor more surface area to detect (Figure 4)
5. Bore a hole into the two layered pieces big enough to press fit the motor onto the arm (Figure 4)
6. Press the motor into this hole (Figure 5)
7. Glue the arm onto the pen (Figure 6)
8. Glue a cardboard stiffener onto the structure to keep it in a rigid box shape (Figure 7)
9. Tape the ultrasonic sensor underneath the motor side of the arm (Figure 8)
10. Tape on the 9V and AA battery onto the other side of the arm (Figure 9)

Operation Instructions

Operation of this seesaw is very straightforward since it is designed to be an autonomous system. Firstly, the 9V battery should be connected to the power rail of the breadboard and then the Arduino sketch should be loaded onto the board via the USB cable. After the program is loaded, it will immediately start to execute and the motor will spin up to keep the seesaw level. Since this is intended to be a demonstration, the setup was designed to be very straightforward.

Appendix A (Bill of Materials)

Part	Source	Part Number	Cost per Unit	Quantity	Total Cost
Propeller	Amazon	B095C61CGN	$3.95	1	$3.95
Ultrasonic Sensor	Digikey	1528-2711-ND	$9.99	1	$9.99
Cardboard	Lab	N/A	$0.11	4	$0.44
Pen	Scavenged	N/A	$0.19	1	$0.19
9V Battery	Circuit Kit	N/A	$4.00	1	$4.00
DC Motor	Circuit Kit	N/A*	$1.17	1	$1.17
H Bridge	Circuit Kit	L9110	$1.50	1	$1.50
AA Battery	Circuit Kit	N/A	$1.60	1	$1.60

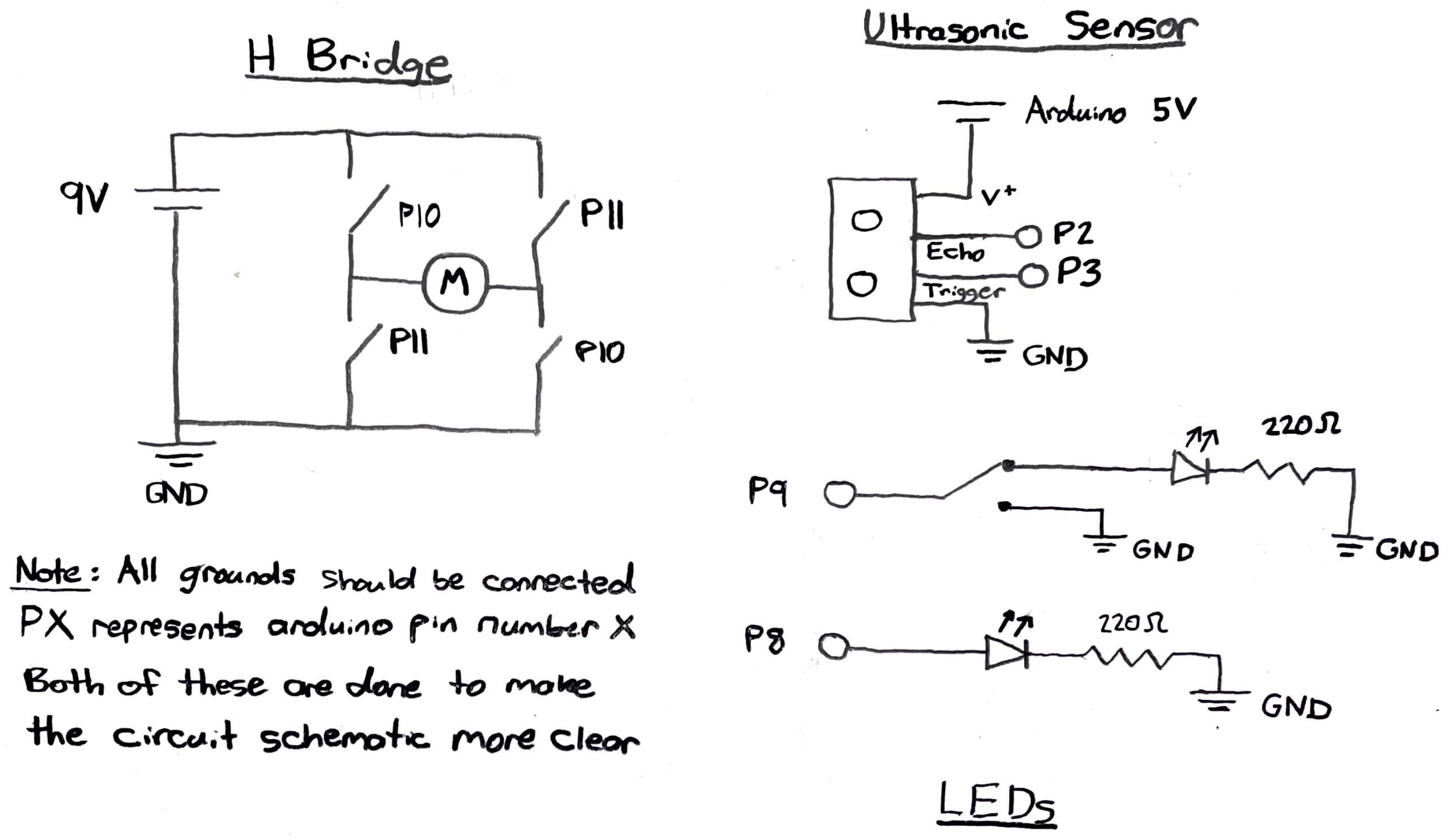
Resistors	Circuit Kit	220 Ohm	$0.05	2	$0.10
Masking Tape	Scavenged	N/A	$0.05	1 foot	$0.05
SPDT Switch	Circuit Kit	N/A	$0.95	1	$0.95
Arduino Uno	Circuit Kit	N/A	$20.00	1	$20.00
Breadboard	Circuit Kit	N/A	$2.00	1	$2.00
LED	Circuit Kit	N/A	$0.40	2	$0.80
Jumper wire	Circuit Kit	N/A	$0.18	2 feet	$0.18

The cost for all items but the Amazon/Digikey and Lab purchased supplies are determined by a quick google search. Additionally, if a pack of ten pens is $1.90, the cost is cited as $0.19.

*If someone were to recreate the project, you can probably buy almost any DC motor and it would work better than in my project. My DC motor had a lot of internal rubbing and friction likely due to its years of abuse. A fresh motor would spin fast enough to generate substantial thrust.

Total Cost (buying everything from scratch): **$46.92**
Total Cost (not accounting for kit parts): **$14.62**

Appendix B: Circuit Diagram



Note: All grounds should be connected
PX represents arduino pin number X
Both of these are done to make the circuit schematic more clear

Appendix C: Assembly Pictures/Mechanical Drawings

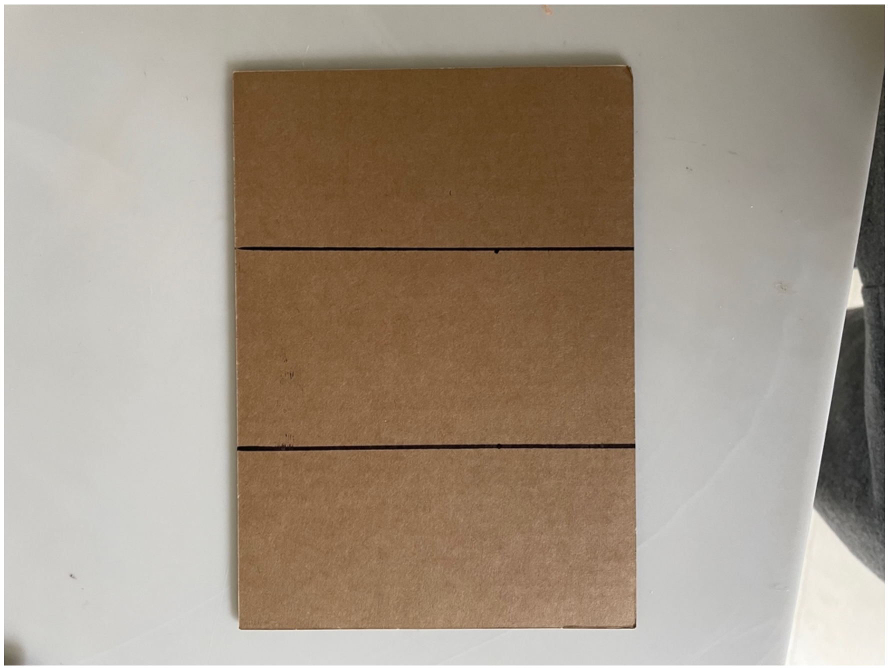


Figure 1: Step 1

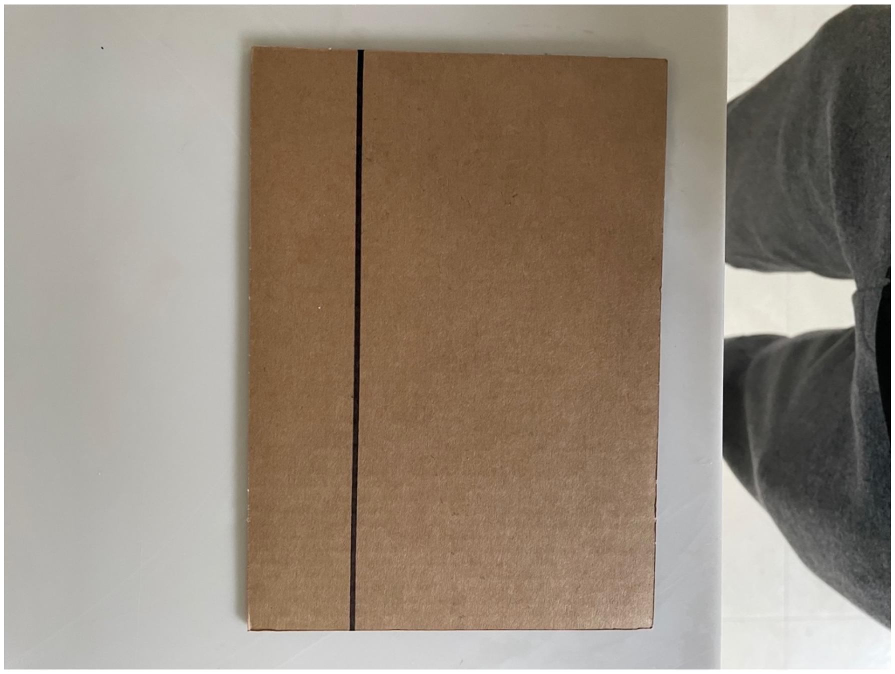


Figure 2: Step 2

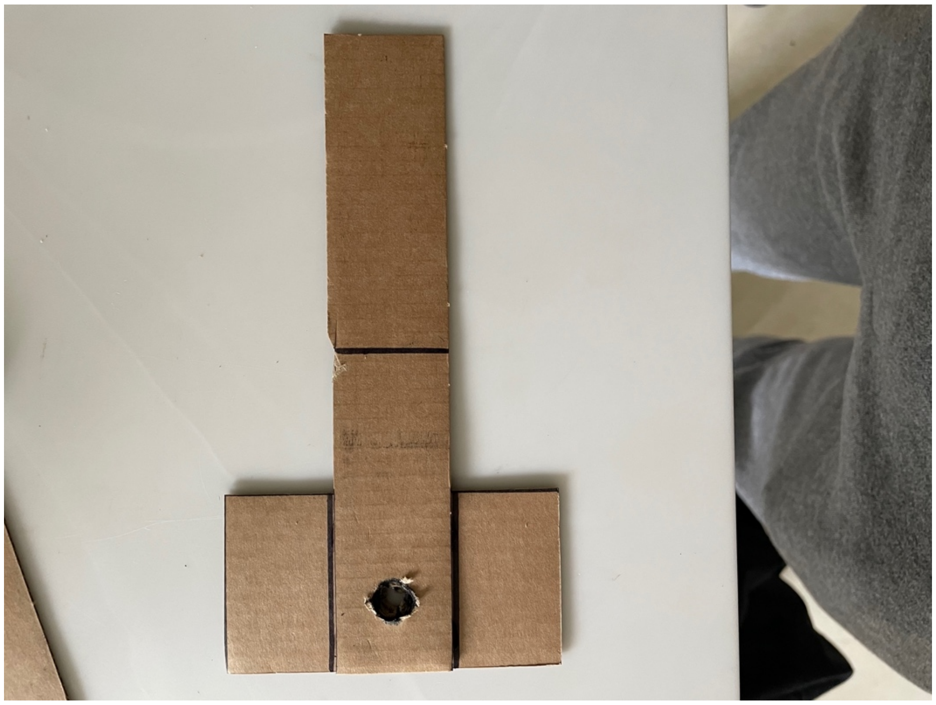


Figure 3: Step 3

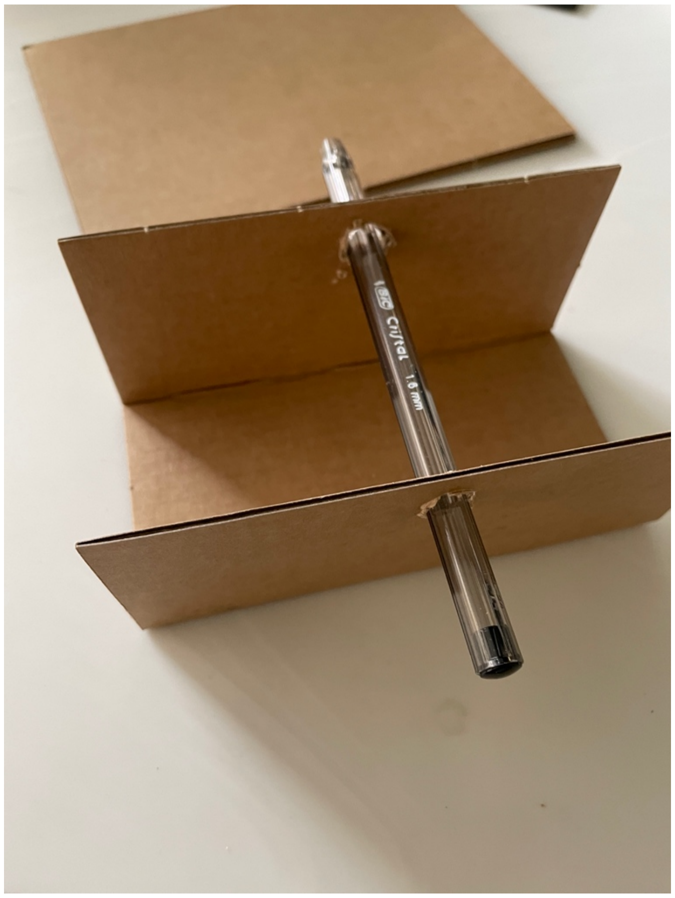


Figure 4: Steps 4 and 5

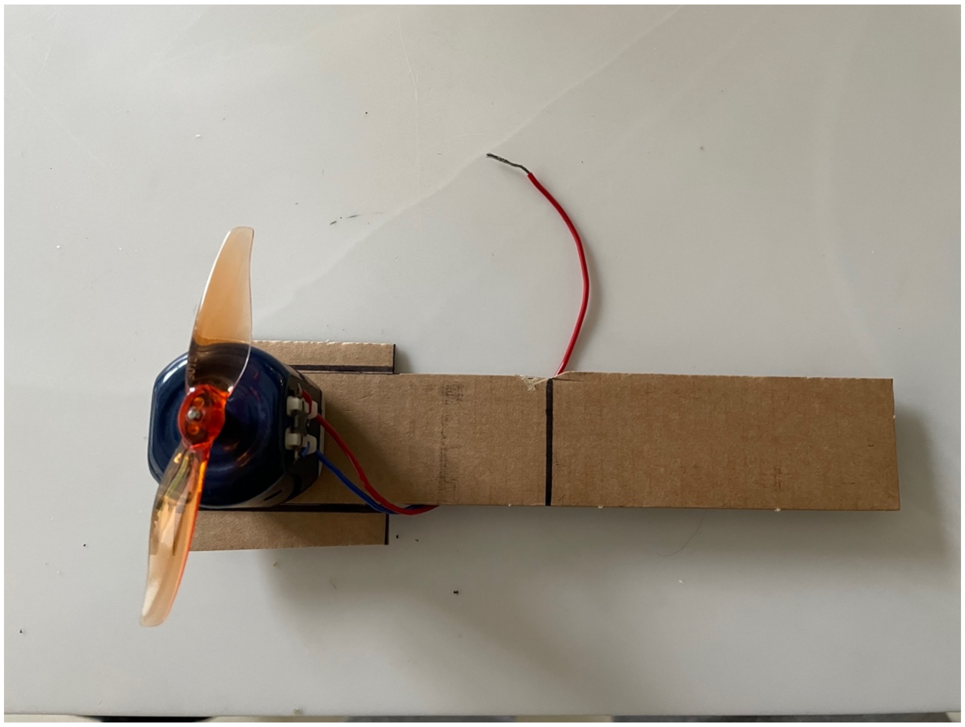


Figure 5: Step 6

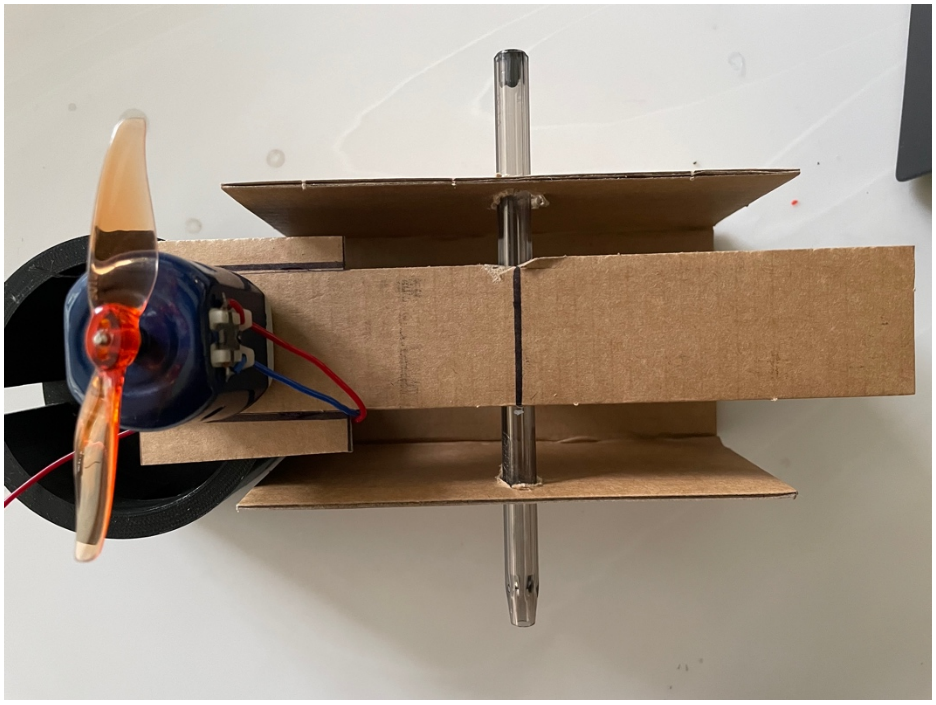


Figure 6: Step 7

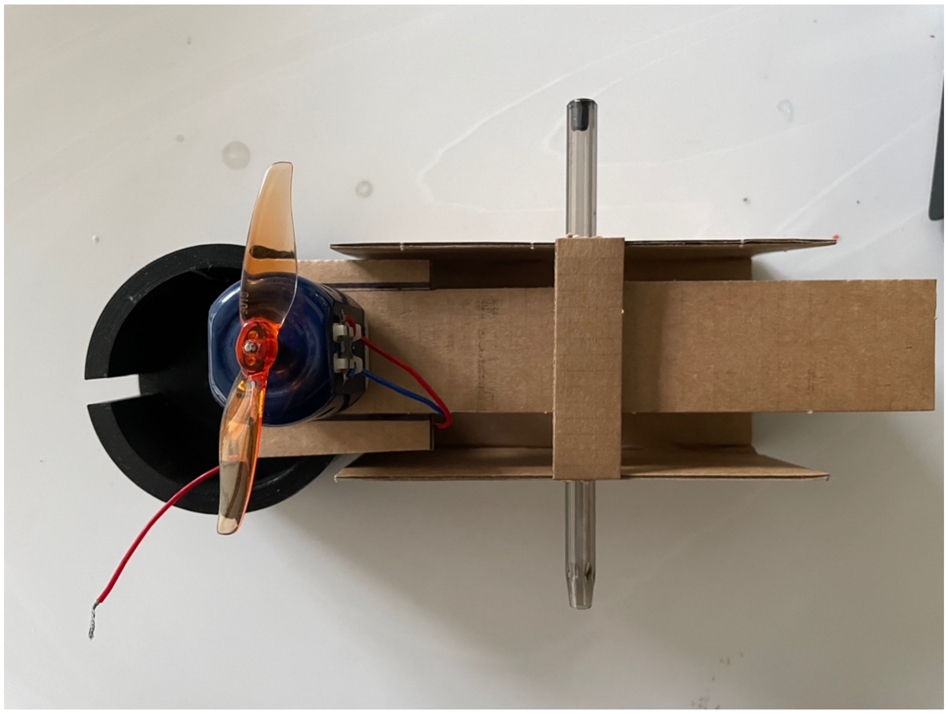


Figure 7: Step 8

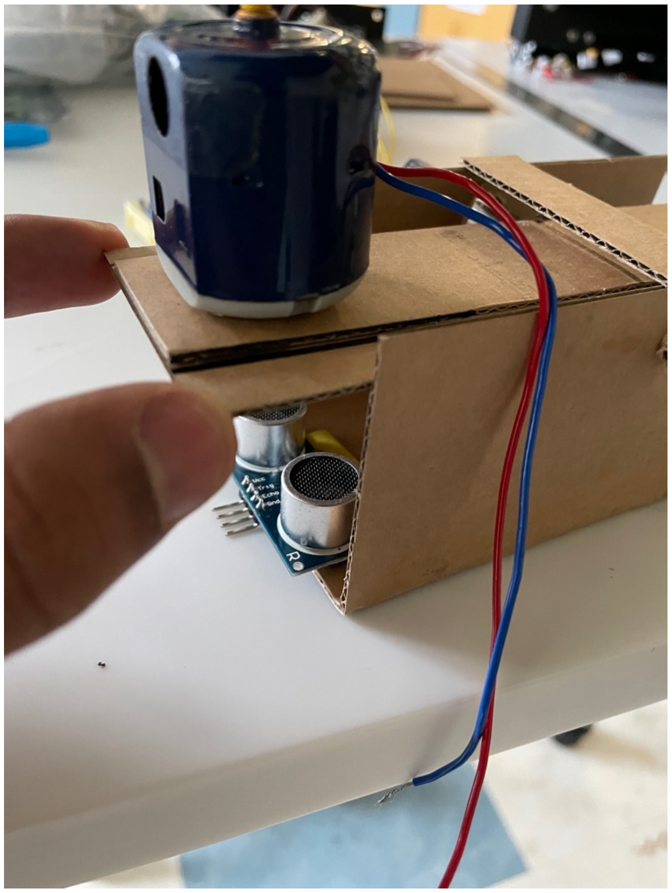


Figure 8: Step 9

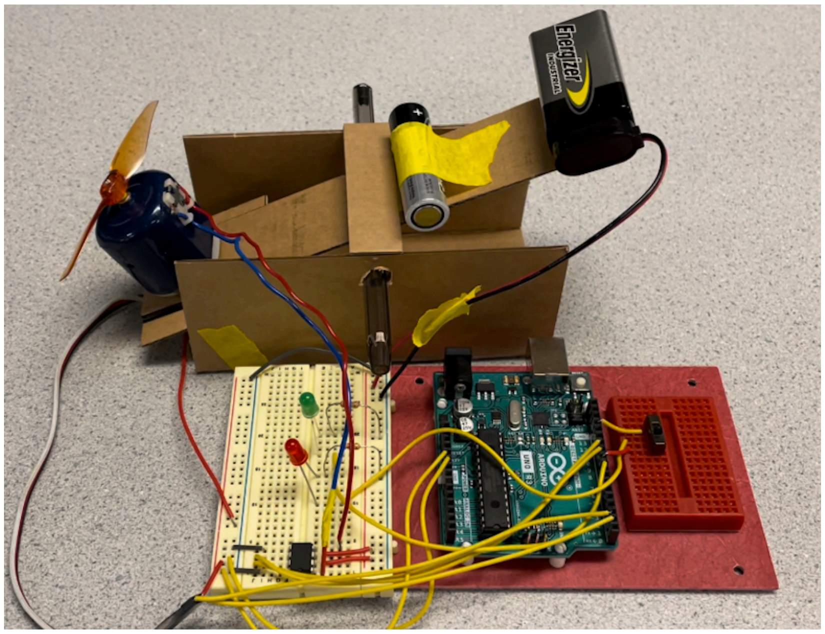


Figure 9: Step 10

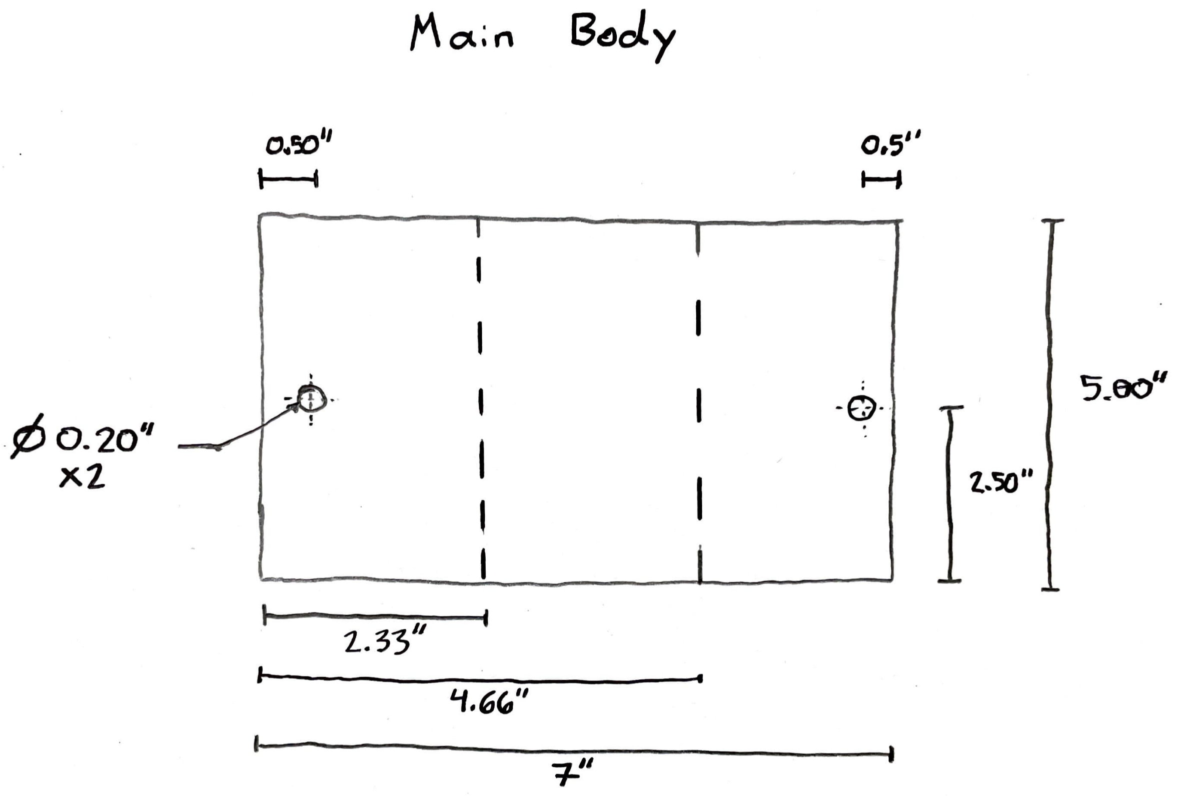


Figure 10: Main Body Drawing

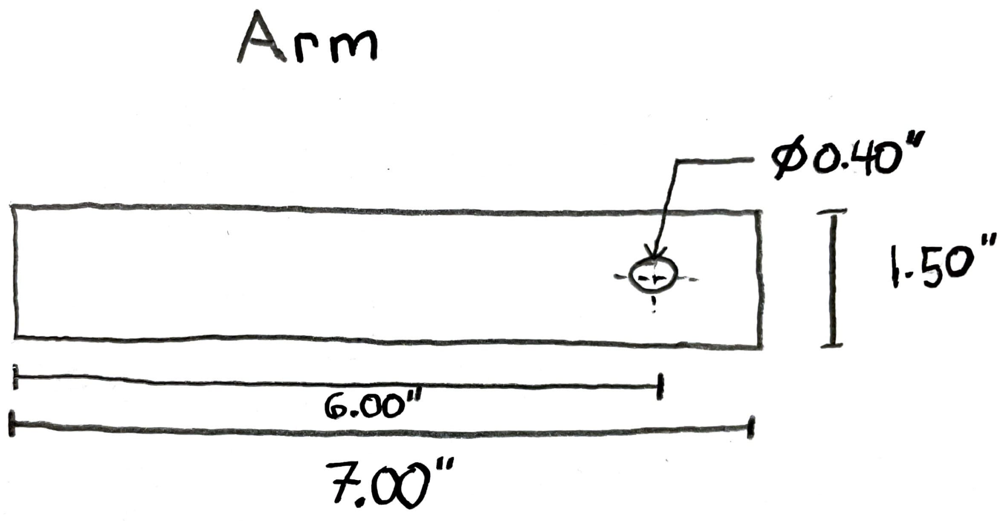


Figure 11: Arm Drawing

Appendix D: Code

```
/*
Govind Chari (gmc93) MAE 3780 Individual Project

The general structure of the code is Read, Control, Actuate.

If the color red is detected, then the motor shuts off and the
red LED is lit. Otherwise, the green LED is lit and firstly,
the ultrasonic sensor detects the position of the seesaw. This value
is then compared to a reference position to compute an error term.
This error term is continously integrated and differentiated then these
terms are taken in weighted summation to yield a motor speed, which is
then written to the motor.
*/

// Pins
int pin_fwd = 11;
int pin_rev = 10;
int pin_ping = 8;
int red_led = 9;
int green_led = 8;
int echoPin = 2; // attach pin D2 Arduino to pin Echo of HC-SR04
int trigPin = 3; //attach pin D3 Arduino to pin Trig of HC-SR04

// PID and Filter Gains
double kp = 10.0;
double ki = 0.0;
double kd = 1.0;
double a = 0.8;
double dt = 0.01;
```

```
// Errors, reference, and distance
double e, ed, ei, ed_filt, dist;
double ref = 5.0;


// Filter/Error Initialization
double eprev = 0.0;
double ed_filt_prev = 0.0;

// Motor Speed
int speed;
long duration;
int distance;


void setup()
{
  Serial.begin(9600);
  pinMode(pin_fwd, OUTPUT);
  pinMode(pin_rev, OUTPUT);
  pinMode(red_led, OUTPUT);
  pinMode(green_led, OUTPUT);
  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT
  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT
}

void loop()
{

    // Pull distance from sensor

    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
    // Sets the trigPin HIGH (ACTIVE) for 10 microseconds
    digitalWrite(trigPin, HIGH);
    delayMicroseconds(10);
    digitalWrite(trigPin, LOW);
    // Reads the echoPin, returns the sound wave travel time in microseconds
    duration = pulseIn(echoPin, HIGH);
    // Calculating the distance
    distance = duration * 0.034 / 2; // Speed of sound wave divided by 2 (go and back)

    // Compute error, integrated error, and derivative of error
    e = -(distance-ref);
    ei = ei + e*dt;
    ed = (e-eprev)/dt;

    // Put derivative through digital lowpass filter
    ed_filt = a*ed + (1.0-a)*ed_filt_prev;

    // Compute final motor speed
    speed = kp*e + ki*ei + kd*ed;


    // Saturation block – Restricts speed to be between 0 and 255
    if (speed <= 0){
      speed = 0;
      digitalWrite(red_led,HIGH); // Turn on red LED
      digitalWrite(green_led,LOW); // Turn off green LED
    }
    else if (speed > 255){
      speed = 255;
      digitalWrite(green_led,HIGH); // Turn on green LED
      digitalWrite(red_led,LOW);  // Turn off red LED

    } else {
      digitalWrite(green_led,HIGH); // Turn on green LED
      digitalWrite(red_led,LOW);  // Turn off red LED

      }

    // Actuate – Write speed to motor using PWM
    Serial.println(speed);
    analogWrite(pin_fwd, speed);
}
```