Overview

- The device is an automated phone stand which tracks the user's head intended for environments such as sitting at your desk, or any scenario where your phone would normally be laying on a desk/table. The device consists of three ultrasonic sensors (HC-SR04) set at about 3 inches apart from one another, both a positional and continuous servo motor, and a slide switch. The slide switch just works as a basic on/off switch. The device works by combining basic human anatomy (the general shape of the upper body and head) with the three sensors. The goal of the device is to ultimately end up with the middle sensor hitting a target less than 2ft (60cm) while having the side sensors hit a target greater than 2ft. It is able to do this by having the 3 sensors mount onto the phone mount which can rotate 360 degrees, and it will rotate until all three sensors are hitting a target less than 2ft which represents hitting the chest area. Once this happens the phone mount is also attached to the positional motor so the phone stand will gradually increase its angle up to 90 degrees, until either of the side sensors obtains a distance greater than 2ft. When this happens, the positional motor will stop increasing the angle and the continuous motor will return to rotating in the same direction. The device will continue to swap between using the positional and continuous motor until only the center sensor is reading a distance less than 2ft, so the device has successfully completed its task.

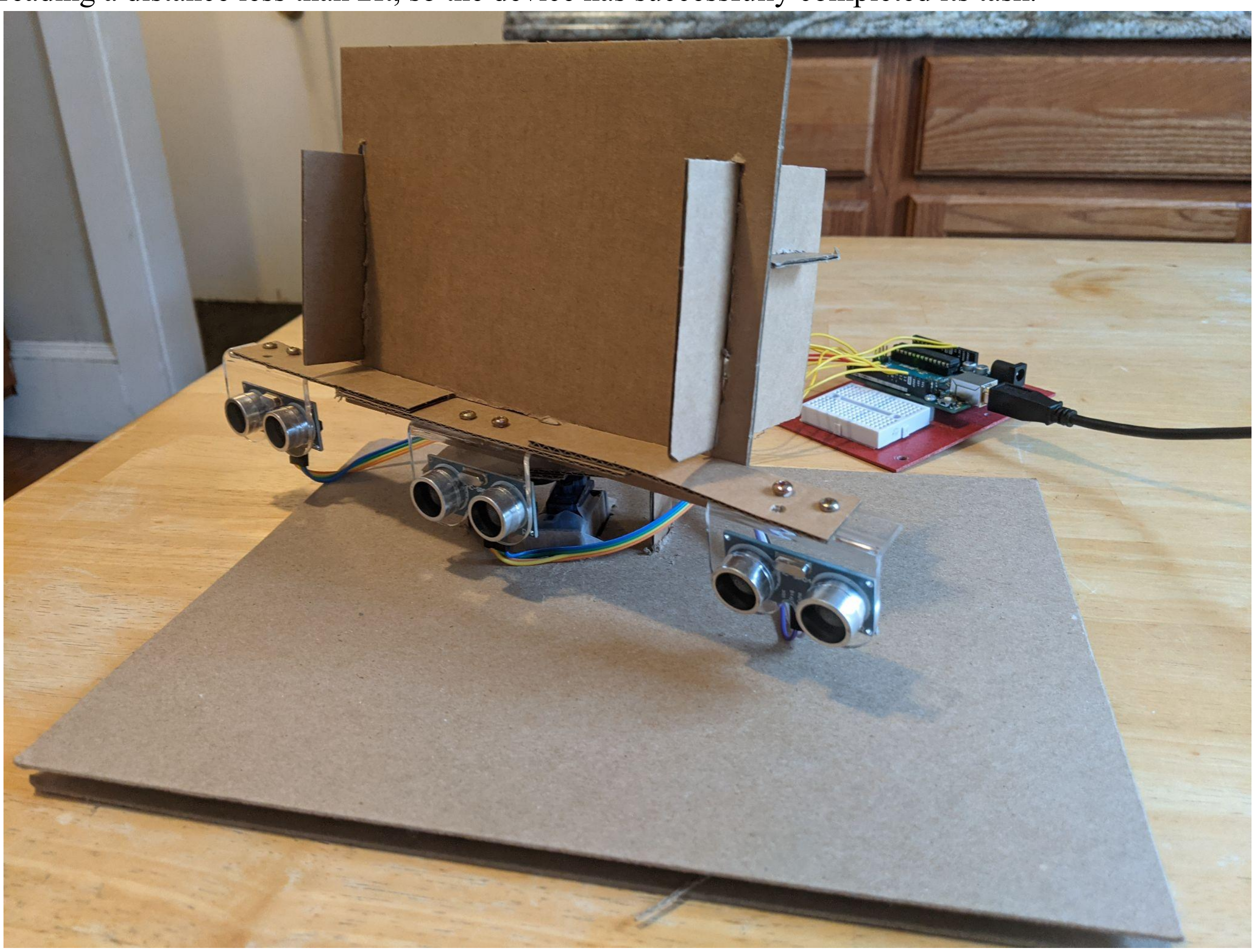


Figure 1: Front Picture of Final Design

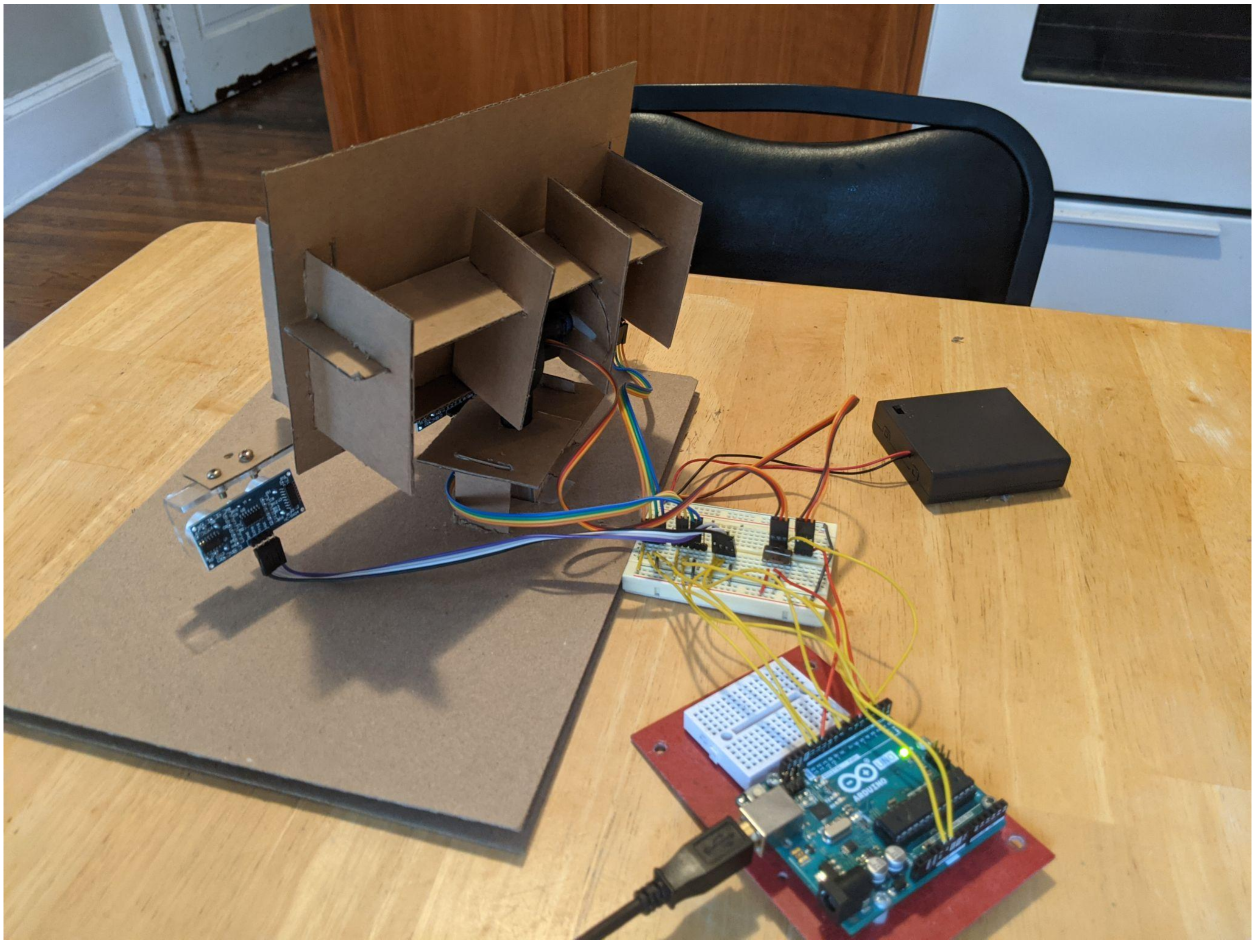


Figure 2: Back Picture of Final Design

Design Considerations

- Given the same constraints:
    - I would redesign the center mount, which was 3D printed, to have its center of mass closer to the center of rotation of the continuous motor.
    - Put a counterweight on the opposite side of the phone stand from the phone to allow for more stability
- Given similar constraints but more time and money:
    - I would get stronger motors, since the continuous motor seemed to struggle to rotate the system at times.
    - I would invest in better ultrasonic sensors, since these sensors would only work properly for a flat surface, which is why the video uses a cardboard cutout of a person instead of an actual person.
    - I would also use more 3D printed parts to allow for a better interface for the motor arm to rotational mount mounting. Using more 3D printed parts would also allow for more stability.

Assembly Instructions

1. Obtain 4 pieces of 5”x7” cardboard and 2 pieces of 8 ½” x 11”
2. Cut the boards appropriately according to the cut drawings (Figures 3-7)

3. Glue the blue shaded components together as seen in Figure 8.
    a. Make sure that the motor is inside the box while you're gluing it together so that the wires can fit through the designated slot.
4. For the gray shaded components, put part B into the horizontal cut in part D, leaving about 1" sticking out the front. Put both Part F pieces through the vertical slots in part D, while leaving the slots in part F behind the front of part D. Put part E through the slots in F. Trace the motor level arm onto the center of component A, and then cut out the trace. Glue part A onto the bottom right corner of H. Put both part H components onto part E, as shown in Figure 9. Adjust the location of both H components to be approximately 2 inches apart from each other, while also having about 2 inches from each F component. Glue parts G onto the portion of part B that sticks out the front of D. Have about 1.5" of G glue onto part B.
    a. Create three sets of two holes with a separation distance of ⅝" at the center of B, and at the edges of E, as shown in Figures 1 and 2.
    b. Place the sensor mounts underneath components B and G, shown in Figures 1 and 2, and use the screw and hex nut to hold the mounts in place.
5. For the red shaded components, wrap each part A around the 3D printed component stem, as shown in Figure 2 and Figure 12. Then glue the two part A components together once already surrounding the 3D printed component. Stick the B components through the slots in A.
6. Cut up excess cardboard in relatively small rectangles. Stack 4 of these rectangles on top of one another to obtain a thickness 4 times the original thickness. Glue the stacks of small cardboard together, and do this for about until you have about 5 stacks. Glue the stacks into each corner and one into the center of the uncut cardstock. Glue the cut cardstock directly on top of the stacks, while being in line with the cardstock underneath.
7. Refer to Figure 1 and 2 for full assembly construction
    a. Place the motor holder component into the corresponding slot in the cardstock. Place the vertical mount stabilizer into the designated slots on the cardstock, while also trying to align the vertical mount to the motor interface so that the continuous motor lever arm fits in the interface.
    b. Place the vertical 3D printed mount in between the two center supports across the back of the phone stand, while also aligning the motor lever arm to the hole in the circular component which was glued onto the separators.
    c. Place the sensors into the appropriate mounts (left, right, or center).

Operation Instructions

1. Set the device at a distance less than 2ft, and try not to set the device within 2 ft of a wall.
2. Turn the on button.
    a. The device will do all of the work, you do not need to do anything else once the switch is turned on since the device will track your head.
3. Turn off the device when done.

Appendix A: Bill of Materials (BOM)

Part Description	Vendor	Part Number	Unit Cost	Quantity	Total Cost (Purchased	Total Cost (Purchases &

					Parts)	**Kit)**
8.5” x 11” 22Pt Cardstock	N/A	N/A	0.15	3	$0.45	$0.45
5" x 7" x 1/16" Cardboard	N/A	N/A	0.11	5	$0.55	$0.55
Tongue Depressor	N/A	N/A	$0.03	15	$0.45	$0.45
Ultrasonic Distance Sensors	Amazon	B07L68X65N	(1/5)*$9.59	3	$5.75	$5.75
3D Printed Mount (PLA)	RPL	N/A	$1 + $0.4 per g	26.56g	$11.62	$11.62
Micro Servo Continuous Motor	DFRobot	SER-0043	$3.90	1		$3.90
Micro Servo Positional	DFRobot	SER0006	$3.30	1		$3.30
4 AA Battery Holder	Jameco	216187	$1.75	1		$1.75
AA Batteries	McMaster-Carr	71455K58	$0.40	4		$1.60
Wire Kit	Amazon	B07PQKNQ22	$2.17	1		$2.17
Arduino Board	Digi-Key	1050-1024-ND	$20.90	1		$20.90
Mini Power Switch	Jameco	2258831	$0.49	1		$0.49
				Total Cost	$18.82	$52.93

Appendix B: Circuit Diagram

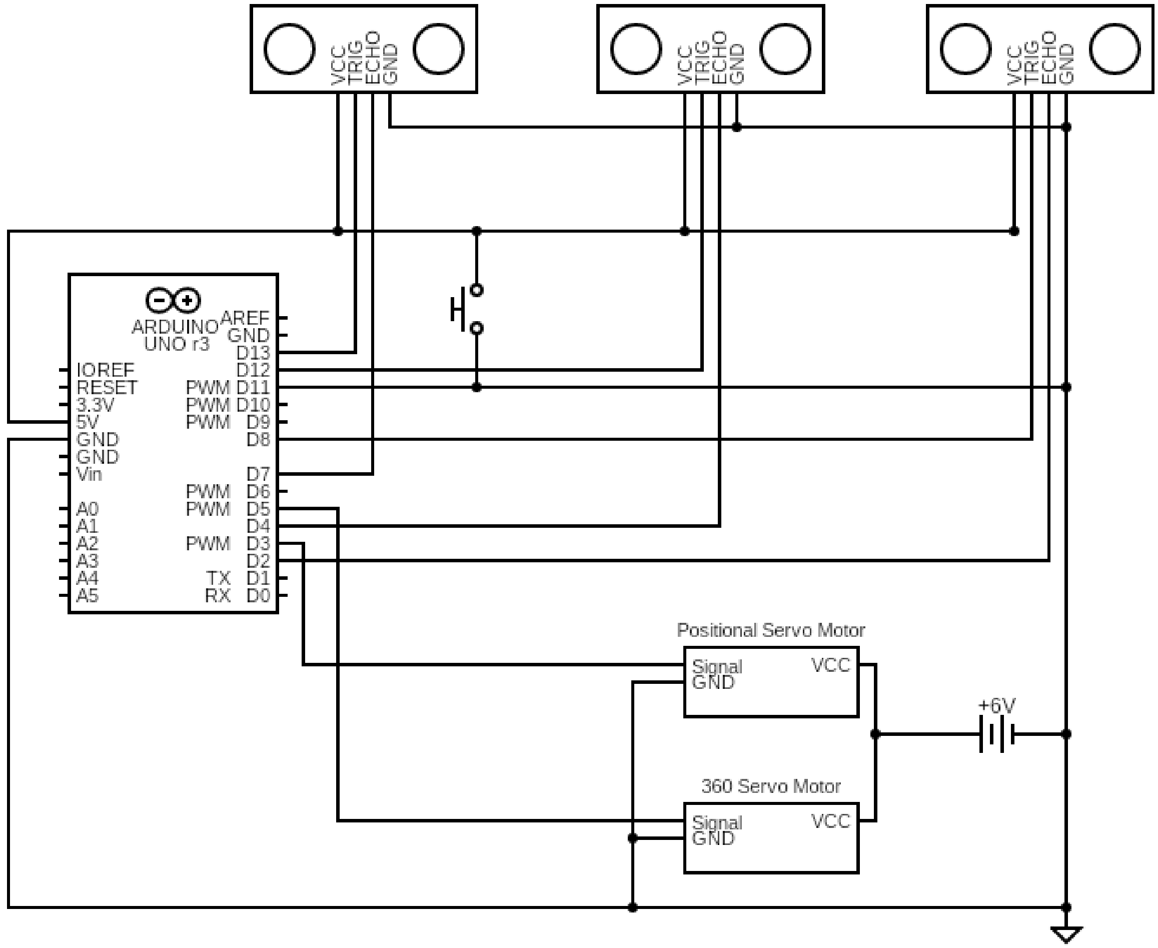


Figure 3: Circuit Diagram

Appendix C: CAD Files & Drawings

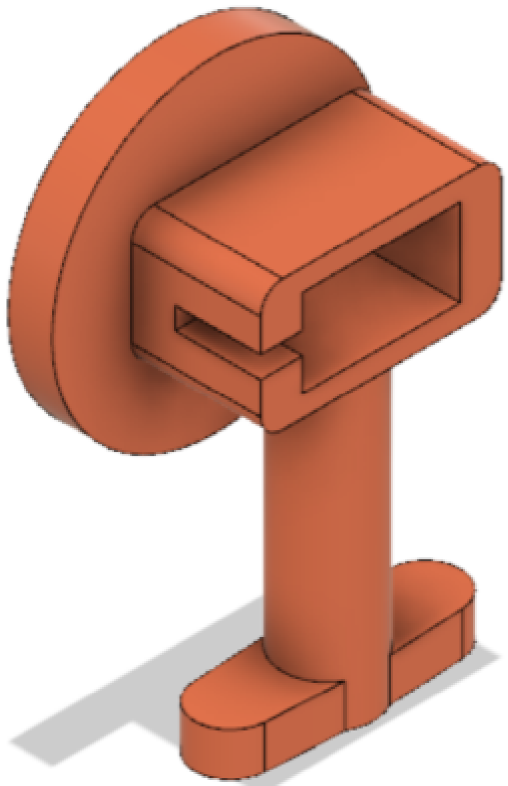


Figure 4: 3D Printed Vertical, Rotating Mount

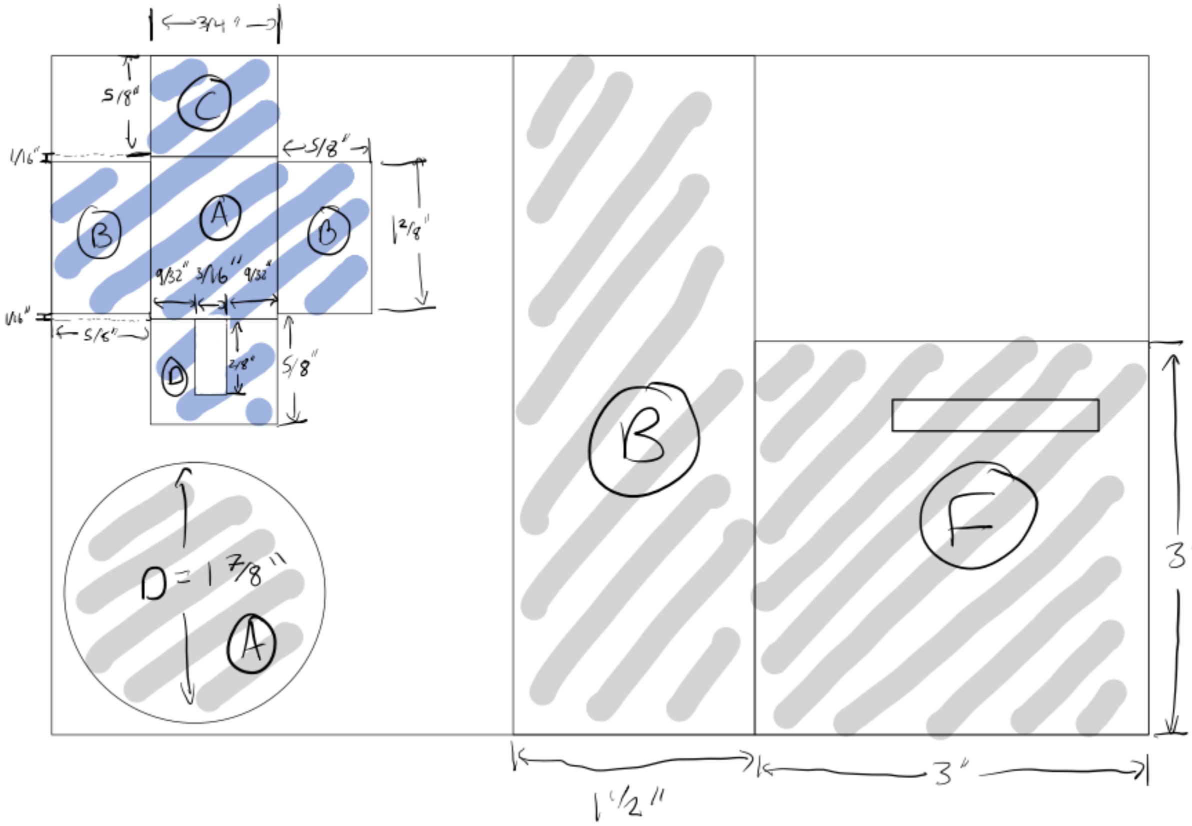


Figure 5: Cardboard 1 Cut Layout

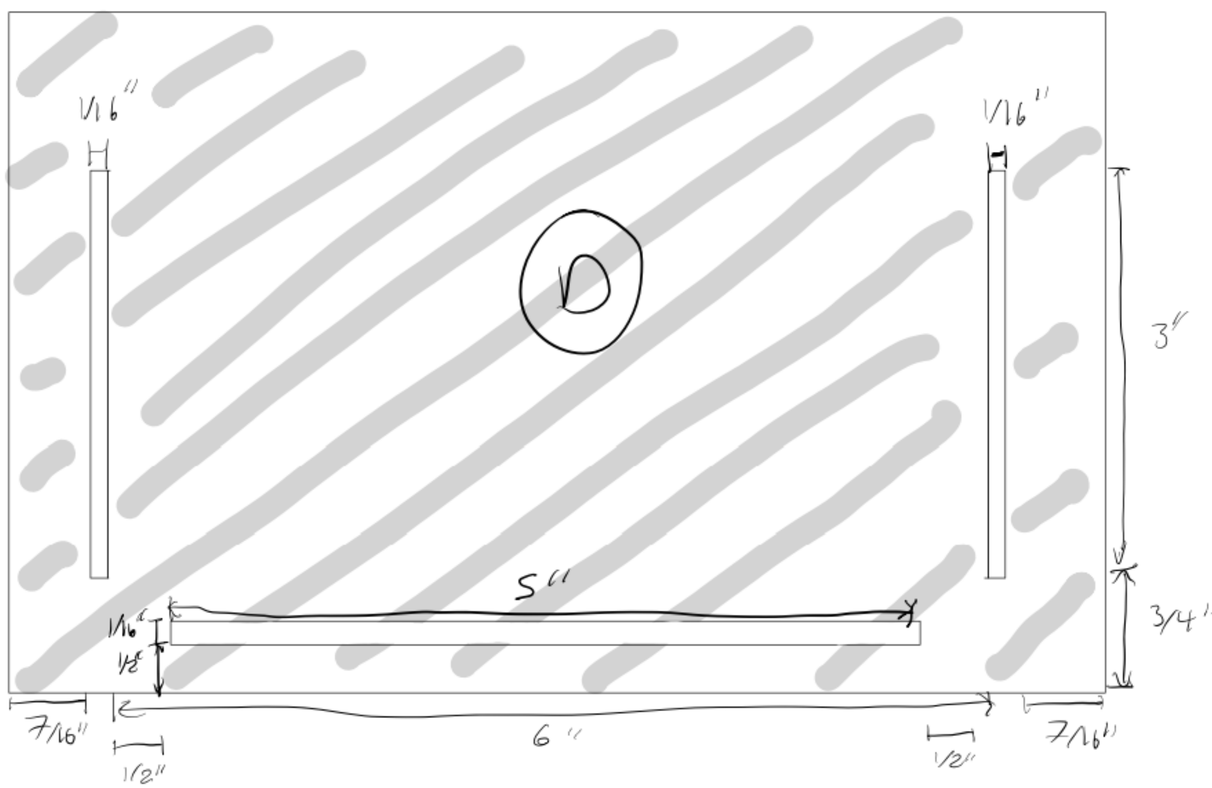


Figure 6: Cardboard 2 Cut Layout

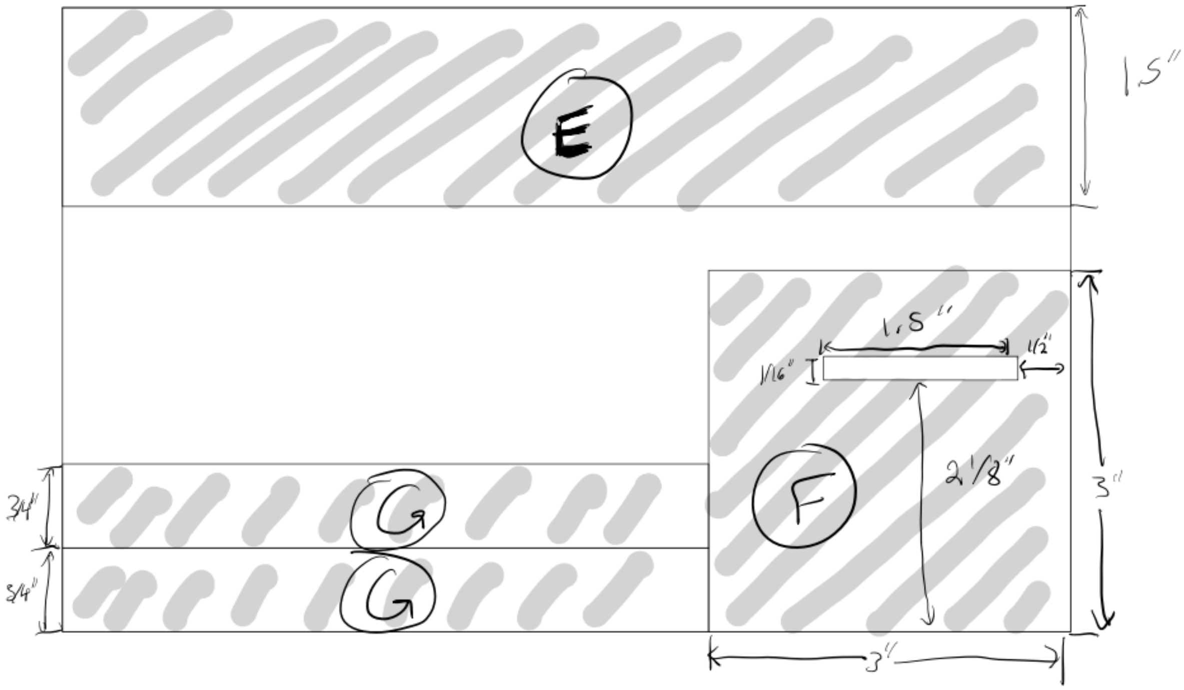


Figure 7: Cardboard 3 Cut Layout

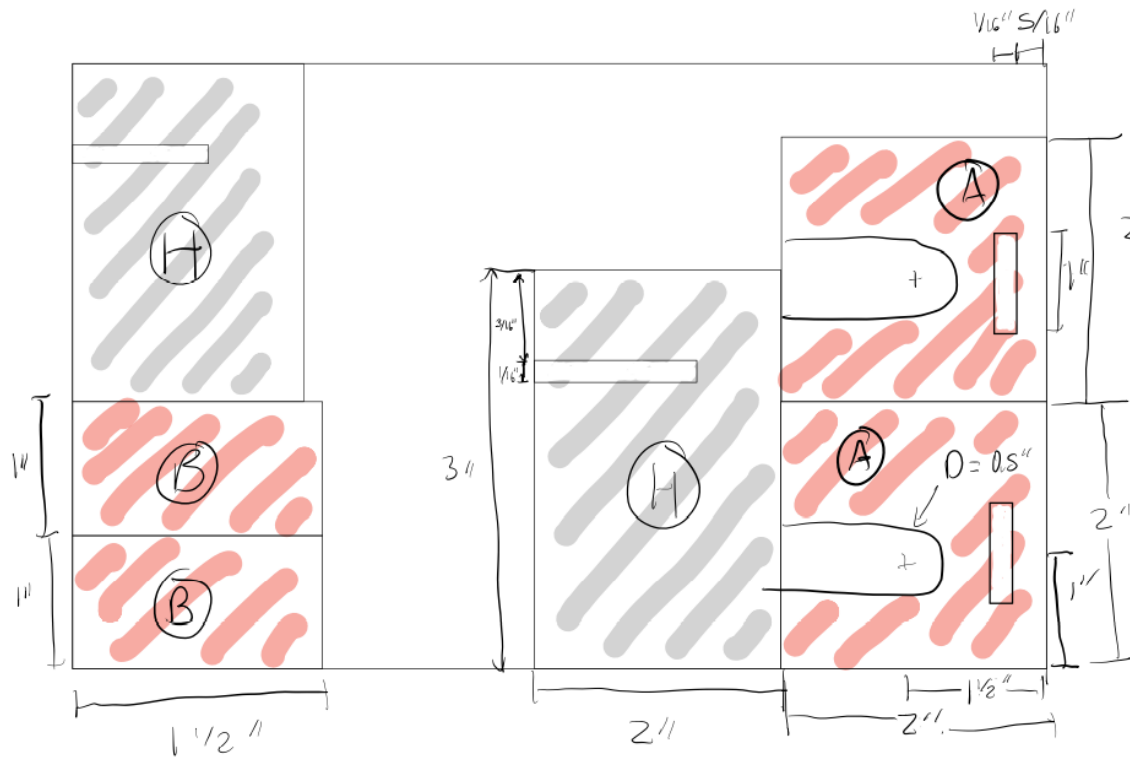


Figure 8: Cardboard 4 Cut Layout

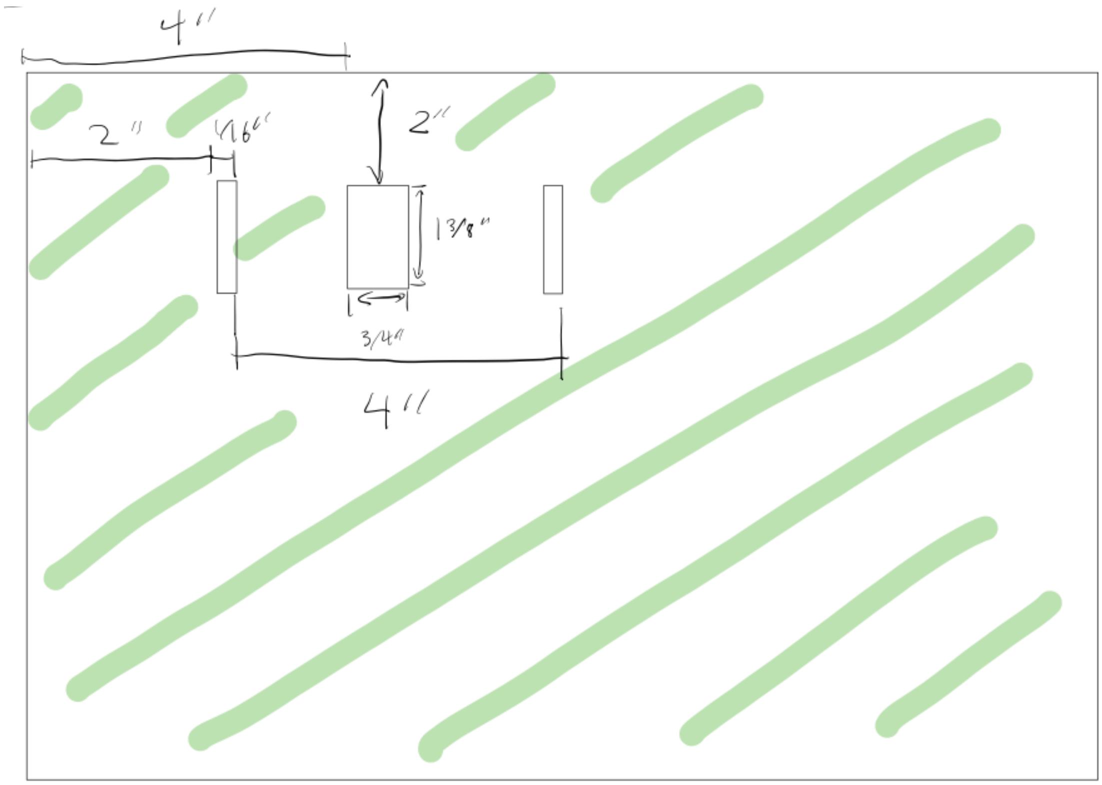


Figure 9: Cardstock Cut Layout

Component 1 (Blue Shaded)

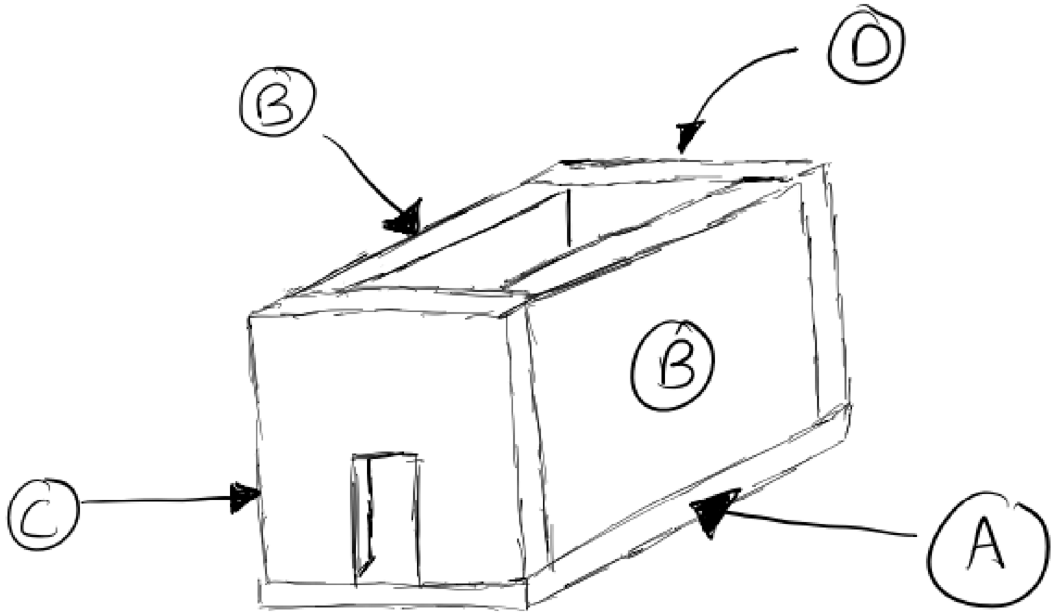
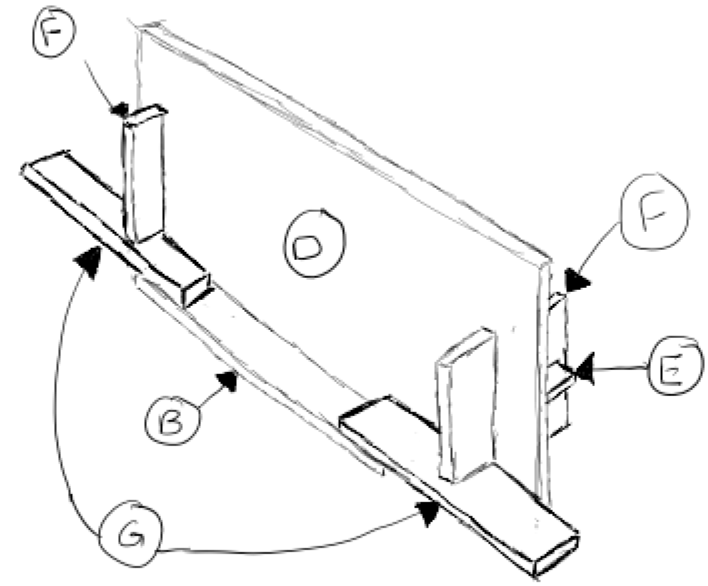


Figure 10: Continuous Motor Holder Assembly

Component 2 (Gray Shaded)

FRONT SIDE ISOMETRIC



BACK SIDE ISOMETRIC

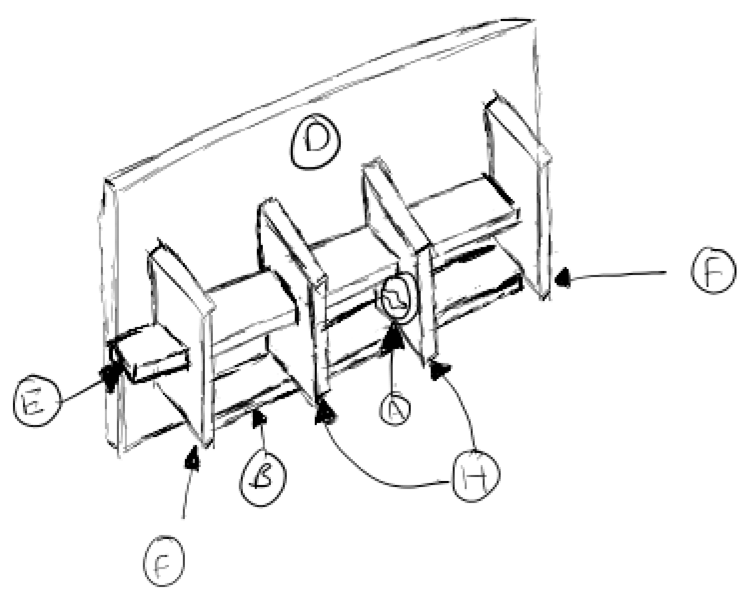


Figure 11: Phone Mount

Component 3 (Red Shaded)

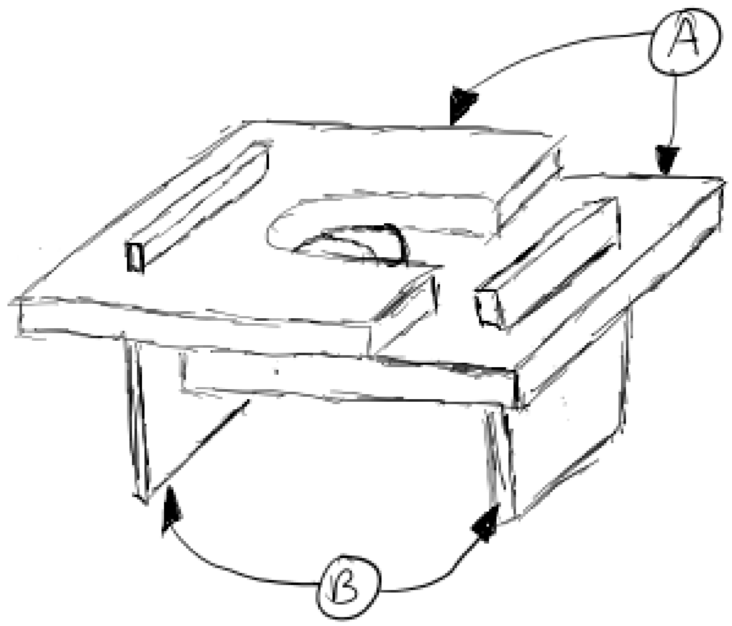


Figure 12: Vertical Mount Stabilizer

Component 4 (Green Shaded)

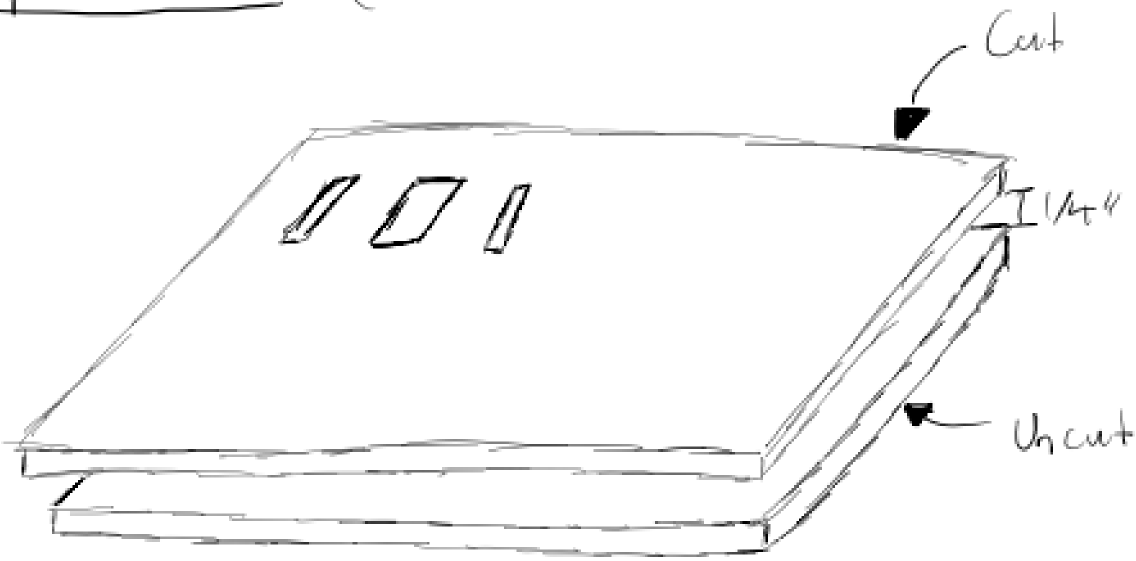


FIgure 13: Base Layout

Appendix D: Commented Arduino Code

```
// C++ code
//
#include <Servo.h>

//The code uses the sensor's results to determine which motor to turn in order to get the center
sensor aligned with the center of the users face.

Servo positional; // 360 servo motor setup
Servo continuous; // 180 servo motor setup

//assigning names to the pins
#define trigPin1 13
#define echoPin1 7
#define trigPin2 12
#define echoPin2 4
#define trigPin3 8
#define echoPin3 2

#define slideswitch   11

//set long variables to be used in upcoming code
long duration, distance, Face_left, Face_center, Face_right;

int angle = 30;

void setup()
{
Serial.begin (9600);

//setting up inputs and outputs for all used ports on the arduino
pinMode(trigPin1, OUTPUT);
pinMode(echoPin1, INPUT);
pinMode(trigPin2, OUTPUT);
pinMode(echoPin2, INPUT);
pinMode(trigPin3, OUTPUT);
pinMode(echoPin3, INPUT);

pinMode(slideswitch, INPUT);


//assigning the motors to a pin
continuous.attach(5);
positional.attach(3);
```

```
//set the 180 servo motor to a base angle.
positional.write(30);
}


void loop() {

//Next 10 lines were copied/altered naming only
//Uses SonarSenor function to gather the distance and assign the distance to a variable to be used for the motors
SonarSensor(trigPin1, echoPin1);
Face_left = distance;
SonarSensor(trigPin2, echoPin2);
Face_center = distance;
SonarSensor(trigPin3, echoPin3);
Face_right = distance;

//printout of the distances
Serial.print(Face_left);
Serial.print(" - ");
Serial.print(Face_center);
Serial.print(" - ");
Serial.println(Face_right);


//if the switch is flipped one way...
if (digitalRead(11)){
 delay(500);
 continuous.write(90); //Set 360 motor to about 0 speed
 positional.write(30); //set 180 motor to start angle of 30 degrees
}

//if the switch is flipped the other way
else{

  //Following if and else if statements cover the possible scenarios needed navigate towards the center of a face, assuming there is one
  if (Face_left > 60 && Face_right > 60 && Face_center > 60){
     continuous.write(100); //set motor spinning at low speed in one direction
      delay(200);
  }

   else if (Face_left < 60 && Face_right < 60 && Face_center < 60){ //ideally represents hitting a person's chest, so the next logical step would be to rotate the sensors up.
       continuous.write(90);
```

```
 delay(200);

 for (angle; angle <= 90; angle++){  //for loop to gradually increase the angle until either one or both side sensors hit a value greater than 2ft

   positional.write(angle);//sets the angle for the positional motor
   delay(100);
   //same copied code from before now included in the for loop
   SonarSensor(trigPin1, echoPin1);
   Face_left = distance;
   SonarSensor(trigPin2, echoPin2);
   Face_center = distance;
   SonarSensor(trigPin3, echoPin3);
   Face_right = distance;

   Serial.print(Face_left);
   Serial.print(" - ");
   Serial.print(Face_center);
   Serial.print(" - ");
   Serial.println(Face_right);

   if (Face_left > 60 && Face_right > 60 && Face_center < 60){
    delay(100);
     exit(0); //found the face
   }
    else if(Face_left > 60 || Face_right > 60){
    break; //return back to spinning in 360 degrees
   }

  }
 }
  else if (Face_left > 60 && Face_right > 60 && Face_center < 60){
    exit(0);  // found the face
   }
  else if(Face_right < 60 && Face_center < 60 && Face_left > 60){
    continuous.write(100);     //     rotate until all 3 sensors get a value less than 60cm or the center of the face is found
   }
  else if(Face_left < 60 && Face_center < 60 && Face_right > 60){
   continuous.write(100);     //     rotate until all 3 sensors get a value less than 60cm or the center of the face is found

  }


}
```

```
}


//This portion of the code was copied, it just converts the signal from the sensors to a numerical value in cm
void SonarSensor(int trigPin,int echoPin)
{
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;

}
```