Raquel Sarabia Soto
netID: rs992

ROCK, PAPER, SCISSORS FRIEND

1. Overview

My device is an articulated robot hand that plays rock, paper & scissors. When it detects a hand, a countdown starts and when it finishes it makes one of the three possible movements. It counts with a scoreboard that shows the actual score of each player.

Controlled by an Arduino Uno, it counts with an ultrasonic sensor that detects if there is a hand at a close distance (from 5 to 20cm). The fingers are controlled by two positional servomotors, one for the thumb, ring finger and little finger and the other for the index finger and the middle finger. The device also counts with several LED lights; two of them are the ones that indicate when to play and six of them are the ones that build the scoreboard (so that the first that get 3 points wins the match). It also counts with three buttons, which are the way of entering the result of each game and they are one button for adding one point to Player 1 (the human), one button to add one point to Player two (the device) and a third button if the game was a tie (so no points are added).

The hand is built in cardboard and the fingers are connected to the servomotors with a thin thread.

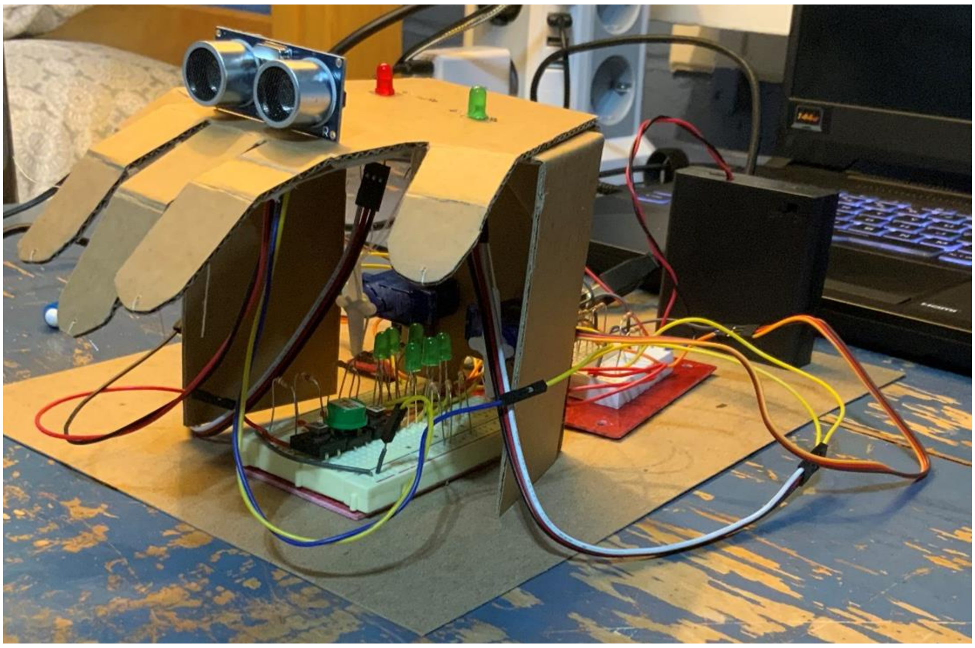


Figure 0. Picture of the completed device.

2. Design considerations

For a better result maybe, another material should have been used for building the hand in order to assure two things: on first place that the motors have strength enough in order to close the fingers and on second place that the fingers go completely back to their natural position (extended) when the motor is not working.

If cardboard is still used, I will say that what needs to be added would be a mechanism to extend the fingers and maybe avoiding marking a lot the fold of the fingers to avoid them to remain fold.

With this same budget it could be studied if a cheap speaker could replace the LED lights that indicate when to start.

Also, if more money was allowed, I would like to install a camera that is able to detect the movement that the human has made and use that instead of the buttons to know the result. If more time was allowed as well, I would have spent more time on investigating a way of helping the fingers go back to their natural position.

3. Assembly instructions

The support of the hand is built by three pieces of cardboard that are glued to the mand by folding the upper part of the cardboard. Also, to reinforce that it has triangled shaped supports that assure a 90º angle between the walls. It also has pieces of plastic straws that guide the thread to the servomotors.

Steps for building the mechanical structure:

1. Cut the cardboard into the figures 1 (hand), 2 (right wall), 3 (back wall), 4 (left wall) and six pieces of figure 5 (triangle supports).
2. Make holes in the hand for passing the ultrasonic sensor wires and the LEDs.
3. Fold the fingers where you want them to get folded.
4. Stick the pieces of the straw to guide the thread. (It is really important to assure that they are well attached as they will have to face the strength of the motors when they tight the thread.)
5. Fold the upper part of the “walls” and stick the triangles.

6. Stick the “walls” to the hand.
7. Pass the thread trough the straws.
8. Make tiny holes in the end of the fingers and pass a second thread through there and again through the straws.
9. Glue both servomotors to the back wall with the wires going out of the wall and reinforce them with a cardboard triangle on each of them.
10. Attach the thread to each motor.
11. Glue the red and green LED lights to the hand.
12. Attach the servomotor by just letting it lay over the cardboard and connect the wires through the whole.
13. On the back wall measure and cut a hole to let the wires from the breadboard pass through there.
14. Tips for assembling the electronic circuit:
    a. let most of the wires that are connected to the Arduino go through the center of the breadboard and tight them by passing a small wire over them (make sure when you connect the extra wire that you do not use any rows that are connected to other parts of the circuit).
    b. Use the mini breadboard that comes with the Arduino to connect the resistors of the LEDs on the top of the hand.
    c. Be careful with the position of the servomotors and the elements of the circuit (such as resistors) as they could collide when the servomotors turn.
    d.

4. Operation instructions.

1. Connect the 6V battery, the USB to the computer and the power supply to the Arduino.
2. It is recommended to upload the code every time you want to use the device as it goes back to its default position.

3. When the code starts running there would be a sequence on the LED lights on the top of the hand that will go red, red, red, green. Every time this sequence is completed, the ultrasonic sensor will read a distance with the nearest object. If it detects an object (the hand of the human player) between 5 and 20cm the LED sequence will start again and when the green LED gets on, both players do the move (rock, paper and scissors). The player would have then to enter the result of that game trough the buttons (P1 if the player won, TIE if it was a tie and P2 if the device won) and the score will be printed on a set of three LEDs per player. The game would not continue until the result is introduced on the device. After this, the game would start again until one of the players reaches three points (the three LEDs on), where all the LEDs would turn off and a new match starts.
4. Take into account that if nothing happens it might be because the robot has had the same movements two times in a row (for example paper and paper), so the way to know that this is happening (and that the ultrasonic sensor has read you presence) would be that the LED light on the top of the hand will be off and that the device is waiting for a result.

Appendix A: bill of materials (BOM)

Part name	Vendor/source	Part number	Quantity	Price ($/unit)	Subtotal
Ultrasonic sensor	Amazon: Smraza	B01JG09DCK	1 (out of 5)	$2.32	$2.32
Servomotors	Amazon: DORHEA	B08FJ27Q1H	2 (out of 2)	$1.59	$3.18
Straws	Amazon: Items 4U	B06XSGZQCD	225*(1)	$3.79	$3.79
Thread	Amazon: Singer	B000PSBYBG	75 yd	$1.62/75yd	$1.62
5” x 7” x 1/16” Cardboard	Cornell Lab	-	5*(2)	$0.11	$0.55
8.5” x 11” 22Pt Cardstock	Cornell Lab	-	1	$0.15	$0.15
Wire*(3)	Cornell Lab	-	5 feet	$0.10/foot	$0.50
Push button	Cornell Lab	-	1	$0.35	$0.35
Glue (epoxy)	Cornell Lab	-	2	$1.75	$3.50
Arduino board	Digi-key	1050-1024-ND	1	$20.90	$20.90
4-wire harness	Digi-key	1568-1931-ND	1	$1.35	$1.35
Mini Breadborad	Newark	98AC7296	1	$1.05	$1.05
Tactile Switch Push Button	Jameco	155380	2	$0.35	$0.70
Wire kit	Amazon: Austor	B07PQKNQ22	1	$2.17	$2.17
Resistor 220ohm	Digi-key	220QBK-ND	8*(4)	$0.01	$0.08
Resistor 10kohm	Digi-key	10KQBK-ND	3	$0.01	$0.03
3-wire extension	Digi-key	1568-1930-ND	2	$1.35	$2.70
Red LED	Jameco	697602	1	$0.05	$0.05
Green LED	Jameco	334086	7	$0.08	$0.56
Power Supply	Jameco	133891	1	$4.95	$4.95
USB cable A to B	monoprice	39918	1	$1.09	$1.09
AA bateries	McMaster-Carr	71455K58	4	$0.40	$1.60
4-AA Battery Holder	Jameco	216187	1	$1.74	$1.75
				Total	$54.93

*:

(1): Only 5 needed (max) did not find someone to share with.

(2): I have only used 3 but it is not bad to have extra in case something needs to be repeated.

(3): I have not used any of it, I was able to connect all the circuit with the wire kit, a 4-wire harness and two 3-wire extensions (everything from the kit).

(4): Each LED needs a 220-ohm resistor. As in the kit I only had 5 I had to combine resistors from different values to form an equivalent resistor of approx. 220ohm. If someone is going to build the device, I would recommend try to find eight 220 ohm resistors (you will need less physical space).

Purchased items

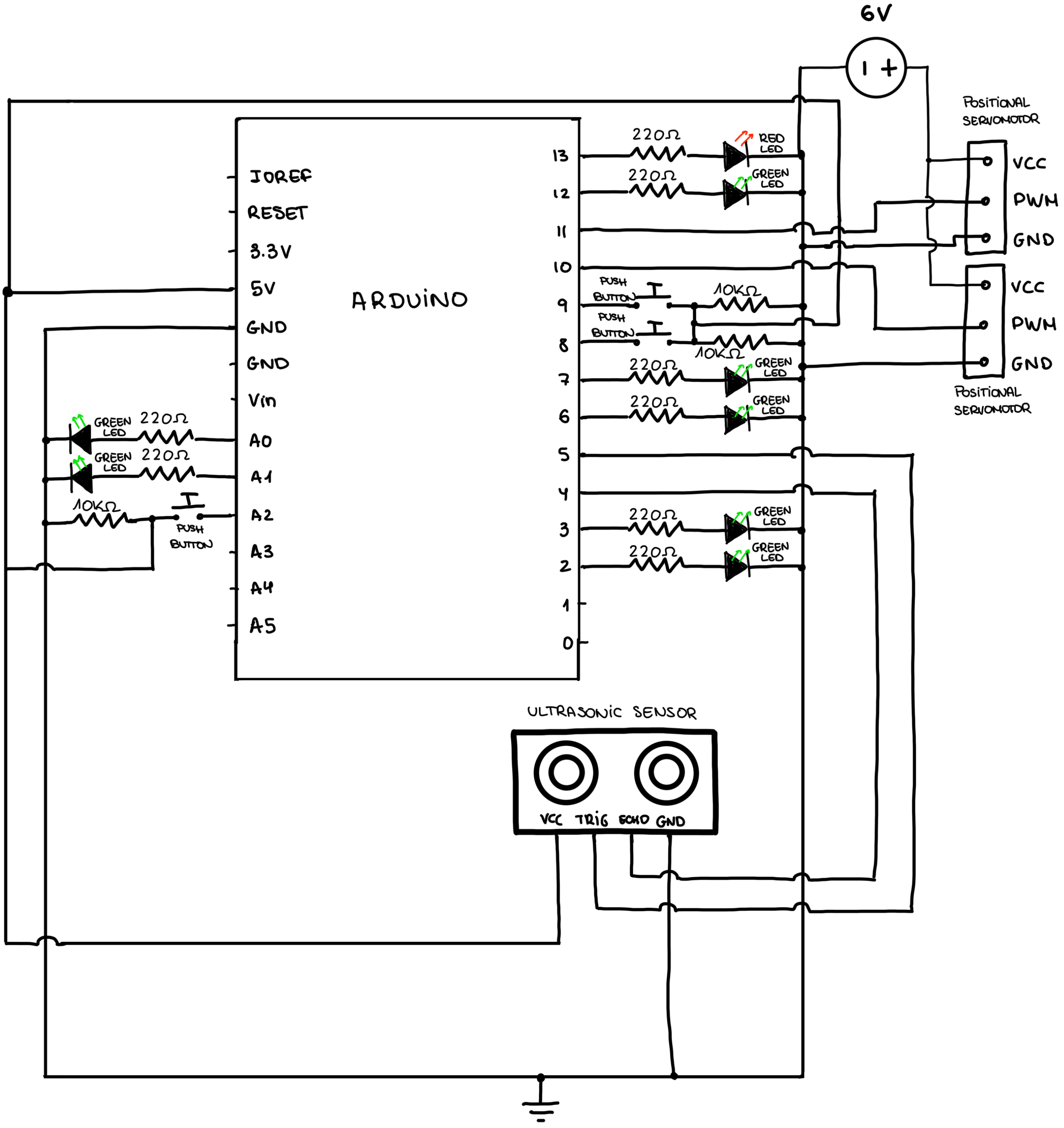
Part name	Vendor/source	Part number	Quantity	Price	Subtotal
Ultrasonic sensor	Amazon: Smraza	B01JG09DCK	1 (out of 5)	$2.32	$2.32
Servomotors	Amazon: DORHEA	B08FJ27Q1H	2 (out of 2)	$1.59	$3.18
Straws	Amazon: Items 4U	B06XSGZQCD	225	$3.79	$3.79
Thread	Amazon: Singer	B000PSBYBG	75 yd	$1.62/75yd	$1.62
5” x 7” x 1/16” Cardboard	Cornell Lab	-	5	$0.11	$0.55
8.5” x 11” 22Pt Cardstock	Cornell Lab	-	1	$0.15	$0.15
Wire	Cornell Lab	-	5 feet	$0.10/foot	$0.50
Push button	Cornell Lab	-	1	$0.35	$0.35
Glue (epoxy)	Cornell Lab	-	2	$1.75	$3.50
				Total	$15.96

Kit items

Part name	Vendor	Part number	Quantity	Price	Subtotal
Arduino board	Digi-key	1050-1024-ND	1	$20.90	$20.90
4-wire harness	Digi-key	1568-1931-ND	1	$1.35	$1.35
Mini Breadborad	Newark	98AC7296	1	$1.05	$1.05
Tactile Switch Push Button	Jameco	155380	2	$0.35	$0.70

Wire kit	Amazon: Austor	B07PQKNQ22	1	$2.17	$2.17
Resistor 220ohm	Digi-key	220QBK-ND	8	$0.01	$0.08
Resistor 10kohm	Digi-key	10KQBK-ND	3	$0.01	$0.03
3-wire extension	Digi-key	1568-1930-ND	2	$1.35	$2.70
Red LED	Jameco	697602	1	$0.05	$0.05
Green LED	Jameco	334086	7	$0.08	$0.56
Power Supply	Jameco	133891	1	$4.95	$4.95
USB cable A to B	monoprice	39918	1	$1.09	$1.09
AA bateries	McMaster-Carr	71455K58	4	$0.40	$1.60
4-AA Battery Holder	Jameco	216187	1	$1.74	$1.75
				Total	$38.98

Appendix B: circuit diagram



Appendix C: CAD files

*All measures are in mm.

Contents:

-Figure 1. Hand

> -Considerations: it does not have many measures because the idea is that each person draws its own hand and the position of the wholes for the ultrasonic sensor and LED lights depends on the position of the rest of the elements.

-Figure 2. Right wall

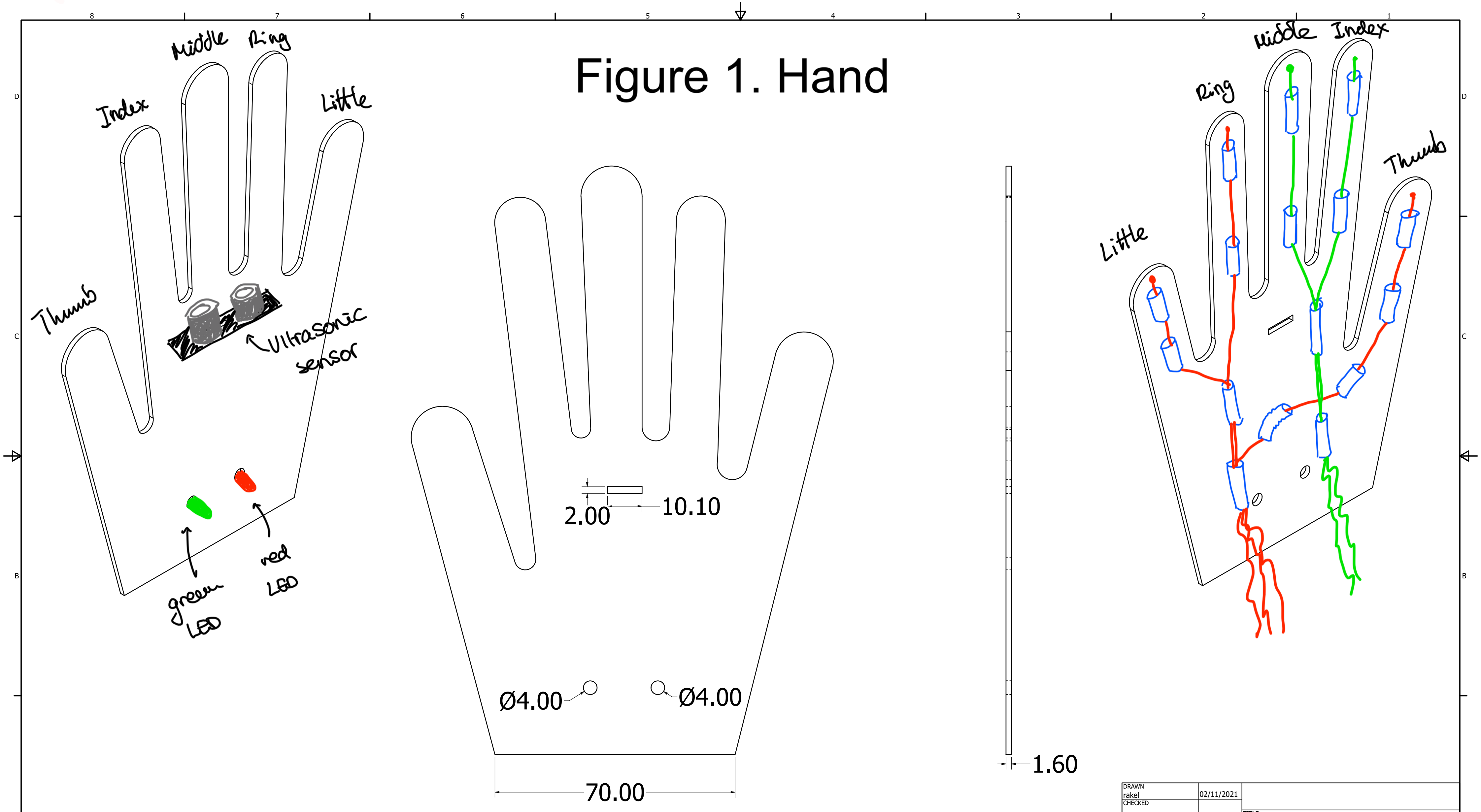
-Figure 3. Back wall

-Figure 4. Left wall

-Figure 5. Triangle

-Figure 6. Assembly

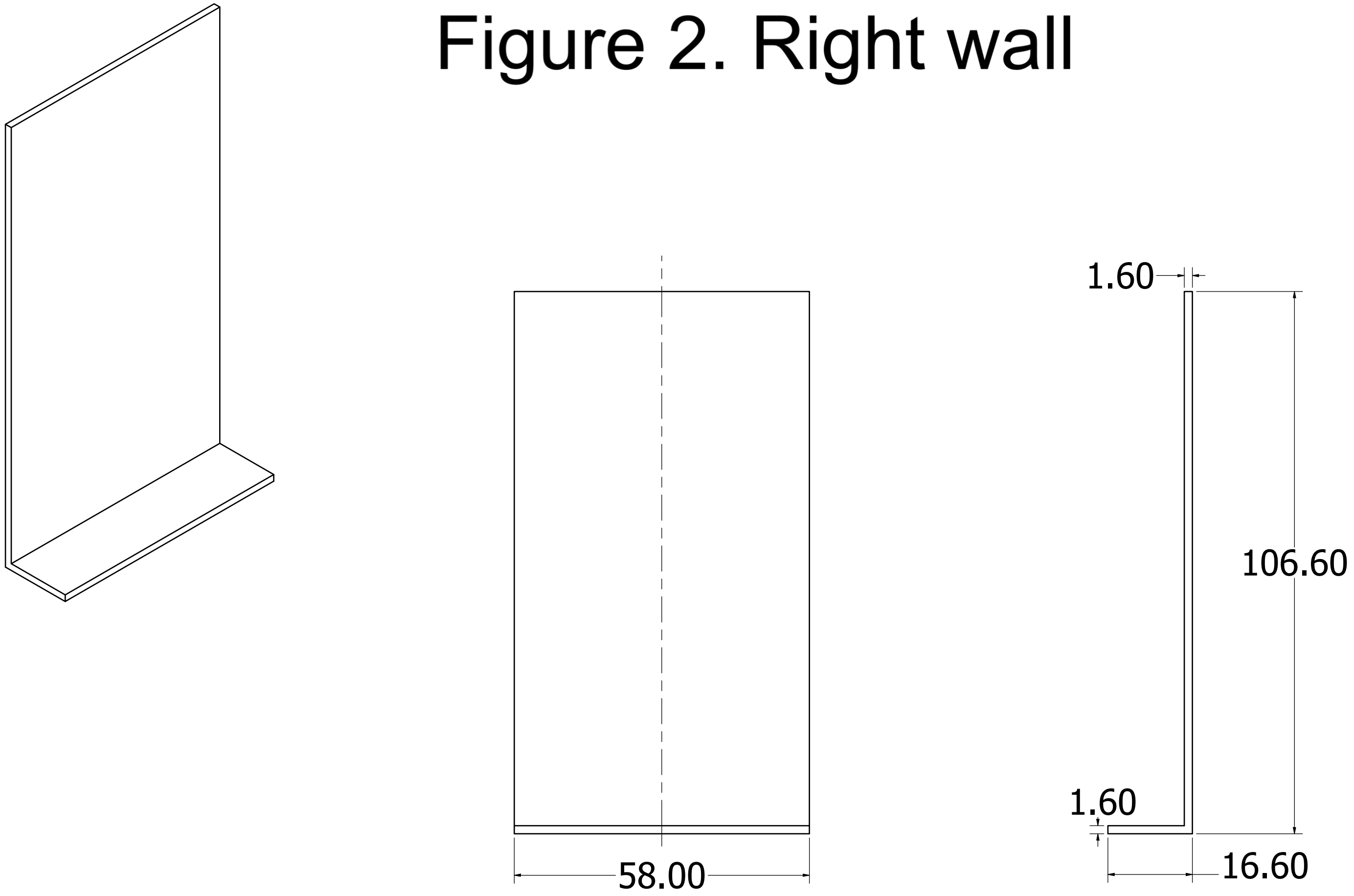
Figure 1. Hand



DRAWN rakel	02/11/2021	
CHECKED		TITLE
QA		
MFG		
APPROVED		
		SIZE D
		SCALE 2 : 1

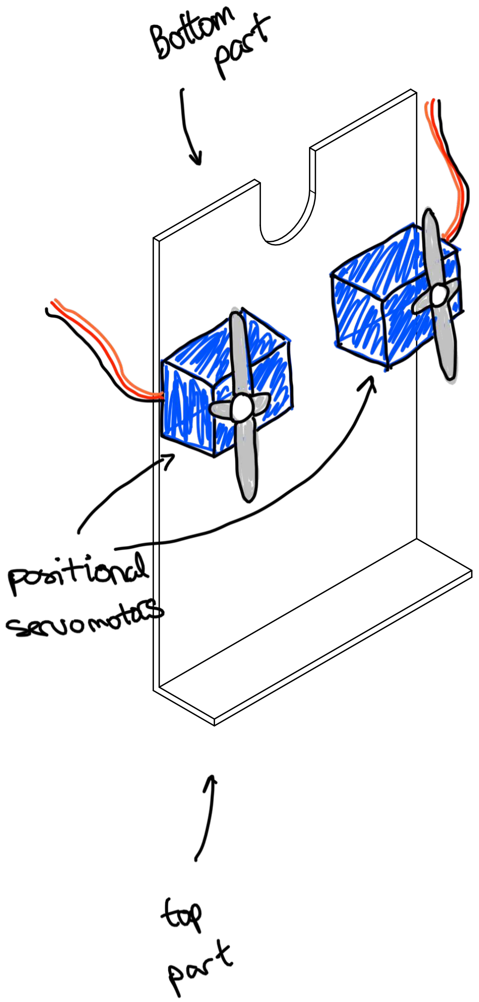
DWG NO Hand		REV
	SHEET 1 OF 1	

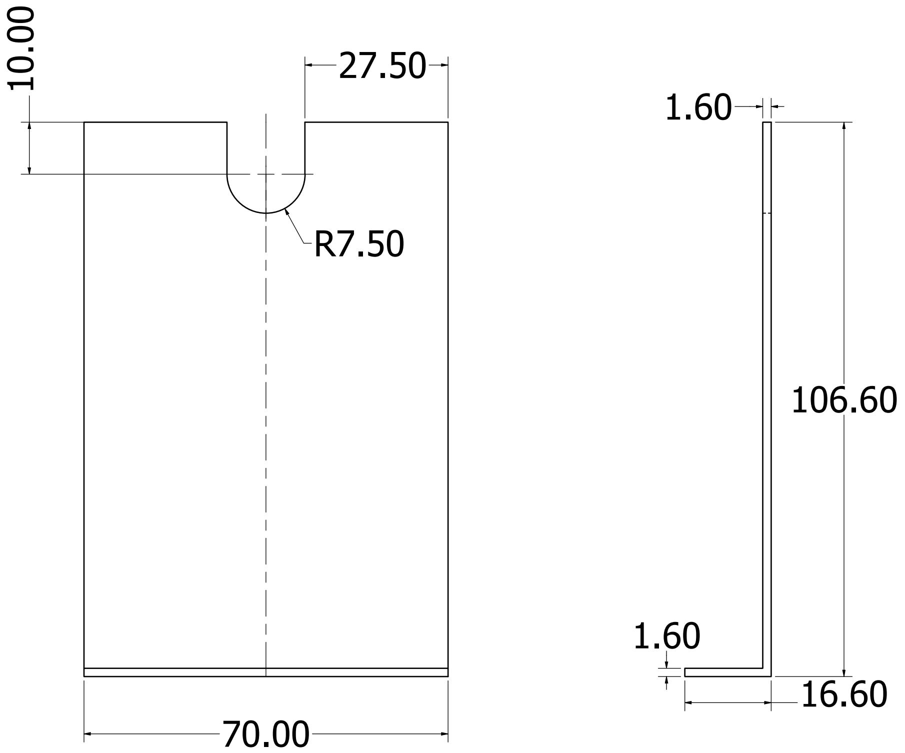
Figure 2. Right wall



DRAWN rakel	02/11/2021				
CHECKED		TITLE			
QA					
MFG					
APPROVED		SIZE D	DWG NO right wall		REV
		SCALE	2 : 1	SHEET 1 OF 1	

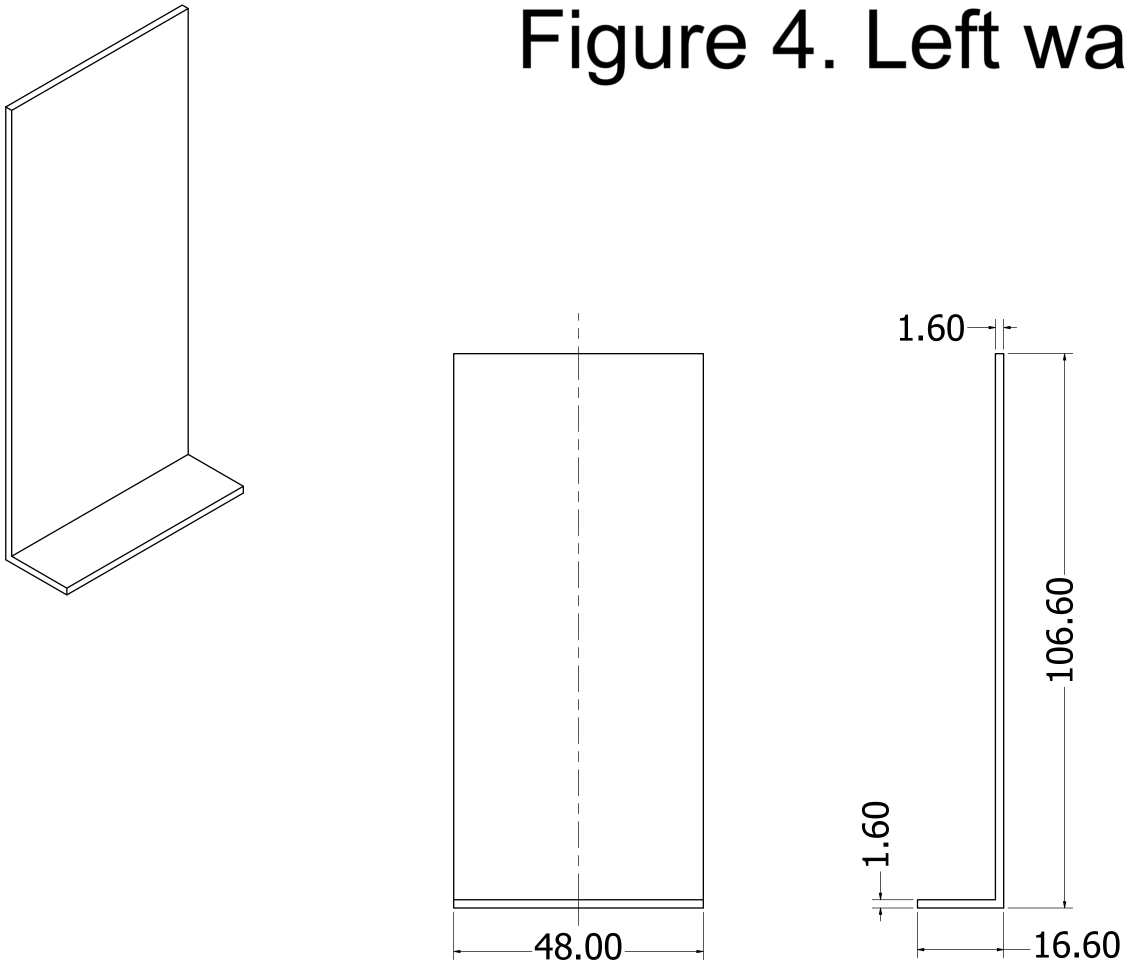
Figure 3. Back wall





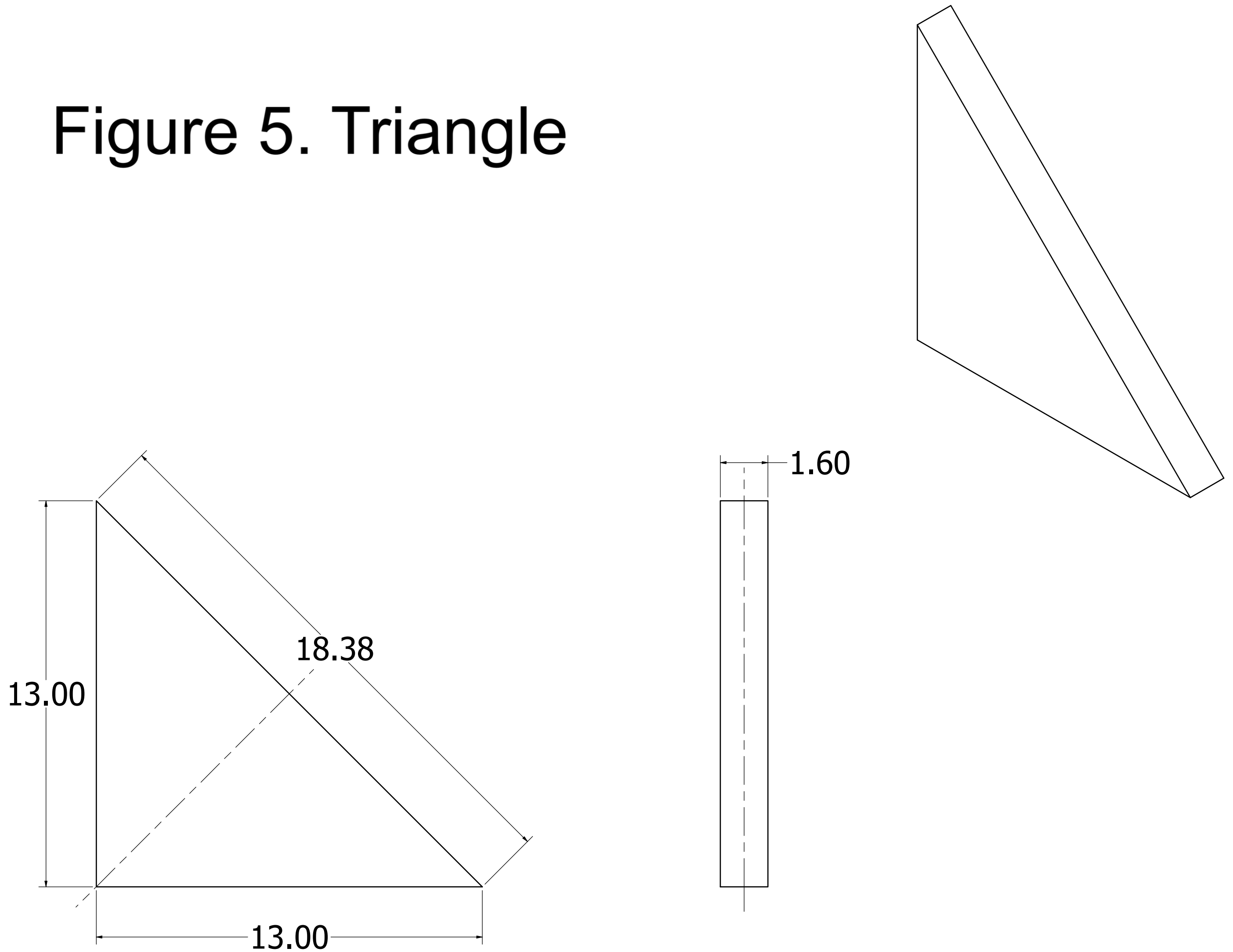
DRAWN rakel	02/11/2021			
CHECKED		TITLE		
QA				
MFG				
APPROVED				
		SIZE D	DWG NO back wall	REV
		SCALE 2 : 1		SHEET 1 OF 1

Figure 4. Left wall



DRAWN rakel	02/11/2021		
CHECKED		TITLE	
QA			
MFG			
APPROVED			
		SIZE D	DWG NO left wall	REV
		SCALE 2 : 1	SHEET 1 OF 1

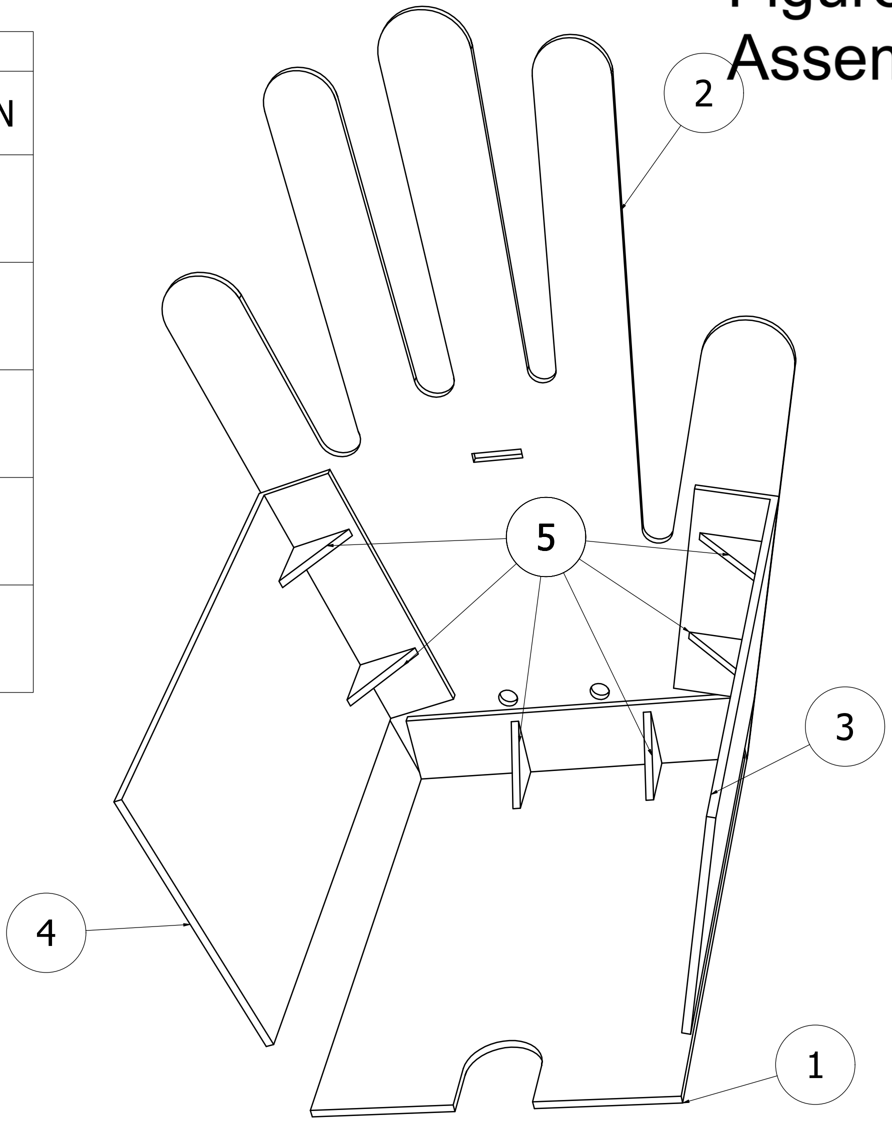
Figure 5. Triangle



DRAWN rakel	02/11/2021		
CHECKED		TITLE	
QA			
MFG			
APPROVED			
		SIZE D	DWG NO Triangle	REV
		SCALE 13 : 1	SHEET 1 OF 1

Figure 6. Assembly

PARTS LIST			
ITEM	QTY	PART NUMBER	DESCRIPTION
1	1	back wall	
2	1	Hand	
3	1	left wall	
4	1	right wall	
5	6	Triangle	



DRAWN rakel	03/11/2021			
CHECKED		TITLE		
QA				
MFG				
APPROVED				
		SIZE D	DWG NO Assembly1	REV
		SCALE 2 : 1	SHEET 1	OF 1

Appendix D: commented Arduino code

```
// Rock, paper, scissors friend
// Designed by Raquel Sarabia
// Date: 07 October 2021
// ------------------------------------------------------------------ //
// Description:
// This program plays the game Rock, Paper, Scissors
// If the ultrasonic sensor detects a hand at a close distance it
generates
// a random movement (rock, paper or scissors) and asks the user what
has
// been the result. The result is introduced by pushing the respective
button:
// P1 if player one won, P2 if player two won or tie if both were
equal.
// After this when the ultrasonic sensor detects the player again
round 2 starts.
// The game finish when one of the player reaches 3 points and it
starts again.
// ------------------------------------------------------------------ //


#define PIN_A0   (14)//Use analog pin as digital
#define PIN_A1   (15)//Use analog pin as digital
#define PIN_A2   (16)//Use analog pin as digital
#define echoPin 4 // attach pin D2 Arduino to pin Echo of HC-SR04
#define trigPin 5 //attach pin D3 Arduino to pin Trig of HC-SR04
#include <Servo.h> // load servo library
Servo myservo1; // create servo object
Servo myservo2; // create servo object


// defines variables
long duration; // variable for the duration of sound wave travel
int distance;// variable for the distance measurement
long aleatorio;//random varible that will generate a new move
long paleat;// variable to save previous rand number
int angle1; // declare variable for myservo1 angle1
int angle2; // declare variable for myservo2 angle2

int k = 1; // variable to initialice

int p1 = 0; // variable to define player's 1 result

int p2 = 0; // variable to define player's 2 result


void setup() {

  pinMode(trigPin, OUTPUT); // Sets the trigPin as an OUTPUT

  pinMode(echoPin, INPUT); // Sets the echoPin as an INPUT

  Serial.begin(9600); // // Serial Communication is starting with 9600
of baudrate speed

  myservo1.attach(11); // tell the Arduino which pin will drive the
servo

  myservo2.attach(10); // tell the Arduino which pin will drive the
servo

  pinMode(7, OUTPUT); // Green LED P1 (1st point)

  pinMode(6, OUTPUT); // Green LED P1 (2nd point)

  pinMode(3, OUTPUT); // Green LED P1 (3rd point)

  pinMode(14, OUTPUT); // Green LED P2 (1st point)

  pinMode(15, OUTPUT); // Green LED P1 (2nd point)

  pinMode(2, OUTPUT); // Green LED P2 (3rd point)

  pinMode(9, INPUT); // Button P1

  pinMode(8, INPUT); // Button P2

  pinMode(16, INPUT); // Button tie

  pinMode(12, OUTPUT); // Green LED start signal

  pinMode(13, OUTPUT); // Red LED ready to start signal

  randomSeed(analogRead(A3));//Instruction I've read that avoids
random command from being pseudo-random

}


void loop() {

  if (k == 1){ // set the program to a initial point (servos on a
specific position and LEDs off)

    myservo1.write(10); // Initial position for myservo1

    myservo2.write(10); // Initial position for myservo1

    //Turn off every LED that shows the results

    digitalWrite(14,LOW);
    digitalWrite(15,LOW);
    digitalWrite(2,LOW);
    digitalWrite(7,LOW);
    digitalWrite(6,LOW);
    digitalWrite(3,LOW);
    paleat = 2; // The initial position will be 'paper'
    k = k+1; // We avoid reading this in the following rounds
  }


  // Clears the trigPin condition
  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)
  // Displays the distance on the Serial Monitor
  Serial.print("Distance: ");
  Serial.print(distance);
  Serial.println(" cm");


  //Ready, set, go signal (green and red LEDs)
  digitalWrite(13, HIGH); //Turns on red
  delay(300);
  digitalWrite(13,LOW); //Turns off red
  delay(300);
  digitalWrite(13, HIGH); //Turns on red
  delay(300);
  digitalWrite(13,LOW);  //Turns off red
  delay(300);
  digitalWrite(13, HIGH); //Turns on red
  delay(300);
  digitalWrite(13,LOW); //Turns off red
  delay(300);
  digitalWrite(12, HIGH); //Turns on green
  delay(700);
  digitalWrite(12,LOW); //Turns off green


  //Here is where the game really starts
  if (4 <= distance && distance <=25){ //If the sensor detects the 
hand...
    aleatorio=random(1,4); // A random number is generated
    Serial.print(aleatorio);
    if (aleatorio == 1){
      Serial.println(" (ROCK)"); // Number 1 means ROCK
    }
    else if  (aleatorio == 2){
      Serial.println(" (PAPER)"); // Number 2 means paper
    }
    else{
      Serial.println(" (SCISSORS)"); // Number 3 means paper
    }


    //Code to do ROCK
    if (aleatorio == 1 && paleat == 1){ // If previous position was 
ROCK it stays the same
    }
    else if (aleatorio == 1 && paleat == 2){ // If previous was 
PAPER...
        for(int angle1 = 10; angle1 < 120; angle1++){ // Thumb, ring & 
little fingers close   CERRAR ++
        myservo1.write(angle1);  //servo spins
        delay(2);
        }
        for(int angle2 = 10; angle2 < 120; angle2++){ // Index & 
middle fingers close  CERRAR ++
        myservo2.write(angle2); //servo spins
        delay(2);
      }
    }
    else if (aleatorio == 1 && paleat == 3){ // If previous was 
SCISSORS...
        for(int angle2 = 10; angle2 < 120; angle2++){ // Index & 
middle fingers close    CERRAR ++
        myservo2.write(angle2); //servo spins
        delay(2);
      }
    }

    //Code to do PAPER
    else if (aleatorio == 2 && paleat == 2){ // If previous position 
was PAPER it stays the same
    }
    else if (aleatorio == 2 && paleat == 1){ // If previous was 
ROCK...
        for(int angle1 = 180; angle1 > 10; angle1--){ // Thumb, ring & 
little fingers open      ABRIR --
        myservo1.write(angle1); //servo spins
        delay(2);
      }
        for(int angle2 = 180; angle2 > 10; angle2--){ // Index & 
middle fingers open       ABRIR --
        myservo2.write(angle2); //servo spins
        delay(2);
      }
    }
    else if (aleatorio == 2 && paleat == 3){ // If previous was 
SCISSORS...
        for(int angle1 = 180; angle1 > 10; angle1--){ // Thumb, ring & 
little fingers open       ABRIR --
        myservo1.write(angle1); //servo spins
        delay(2);
      }
    }

    //Code to do SCISSORS
    else if (aleatorio == 3 && paleat == 3){// If previous position
was SCISSORS it stays the same
    }
    else if (aleatorio == 3 && paleat == 1){// If previous was ROCK...
        for(int angle2 = 180; angle2 > 10; angle2--){ // Index &
middle fingers open        ABRIR --
        myservo2.write(angle2); //servo spins
        delay(2);
      }
    }
    else if (aleatorio == 3 && paleat == 2){// If previous was
PAPER...
        for(int angle1 = 10; angle1 < 120; angle1++){ // Thumb, ring &
little fingers close      CERRAR ++
        myservo1.write(angle1); //servo spins
        delay(2);
      }
    }

    Serial.println("Write the result");//Here you have to say what was
the result to continue playing

    while (digitalRead(9) == LOW && digitalRead(8) == LOW &&
digitalRead(16) == LOW){//The codes blocks until you answer
    }
    if (digitalRead(9) == HIGH){//If P1 button is pulsed
      p1 = p1 + 1;//This adds a point to player one
      if (p1 == 1){
        digitalWrite(7, HIGH);// P1 1st point (LED)
      }
      else if (p1 == 2){
        digitalWrite(6, HIGH);// P1 2nd point (LED)
      }
      else if (p1 == 3){
        digitalWrite(3, HIGH);// P1 2nd point (LED)
        delay(1000);
        //Restart the scores of both players when one of them wins
        digitalWrite(7,LOW);
        digitalWrite(6,LOW);
        digitalWrite(3,LOW);
        digitalWrite(14,LOW);
        digitalWrite(15,LOW);
        digitalWrite(2,LOW);
        p1 = 0;
        p2 = 0;
      }
    }
    if (digitalRead(8) == HIGH){//If P2 button is pulsed
      p2 = p2 + 1;//This adds a point to player two
      if (p2 == 1){
        digitalWrite(14, HIGH);// P2 1st point (LED)
      }
      else if (p2 == 2){
        digitalWrite(15, HIGH);// P2 2nd point (LED)
      }
      else if (p2 == 3){
        digitalWrite(2, HIGH);// P2 3rd point (LED)
        delay(1000);
        //Restart the scores of both players when one of them wins
        digitalWrite(7,LOW);
        digitalWrite(6,LOW);
        digitalWrite(3,LOW);
        digitalWrite(14,LOW);
        digitalWrite(15,LOW);
        digitalWrite(2,LOW);
        p1 = 0;
        p2 = 0;
      }
    }
    if (digitalRead(16) == HIGH){//If tie button is pulsed
      Serial.println("tie");//Game continues
    }
    delay(2500);


    paleat=aleatorio;//paleat remembers the previous figure for the
next round
    }
}
```