**ANIMATRONICS FACE: DR. MANTEUFEL**



Link:<https://www.youtube.com/watch?v=RDPhasftaW4&feature=youtu.be>

John Carroll

Dept. of Mechanical Engineering

San Antonio, TX, USA 78249

kor929@my.utsa.edu

Andre Sirhan

Dept. of Mechanical Engineering

San Antonio, TX, USA 78249

andresir95@gmail.com

Rebeca Velasquez

Dept. of Mechanical Engineering

San Antonio, TX, USA 78249

lty225@my.utsa.edu

Submitted as part of project for ME4543 Mechatronics, Fall 2018

TABLE OF CONTENTS

|  |  |
| --- | --- |
| Abstract | 3 |
| Literature Review | 4 |
| Brainstorming | 7 |
| Supporting Structure | 8 |
| Joints and Motors | 10 |
| Sensors | 12 |
| Programming | 14 |
| Lesson learnt and suggestions | 15 |
| Personnel and Bill of Materials | 15 |
| References | 17 |
| Appendix A: Code | 18 |

**ABSTRACT**

    This project consists of creating an animatronic figure of Randall Manteufel, an engineering professor at the University of Texas at San Antonio. Animatronics is a multi-disciplinary field which integrates [anatomy](https://en.wikipedia.org/wiki/Anatomy), robots, [mechatronics](https://en.wikipedia.org/wiki/Mechatronics), and puppetry resulting in lifelike animation using cable pulled devices or motors to imitate a human.

    Our animatronic figure had to exhibit at least two moving parts, and the movements had to be triggered by two different sensors. We decided on using a photoresistor to trigger the eyelids of our figure to close. For the second sensor we chose an ultrasonic sensor, which would trigger the mouth to start moving synced with audio. This would make it appear that the animatronic was talking. These movements were created by servo motors. In addition to the required movements, our team added two other moving parts to our figure. One controlled by another servo motor, and one controlled by a stepper motor.

A supporting structure was created with joints for the moving parts, and the motors and sensors were attached to this structure. A program had to be written and implemented using an Arduino Mega 2560 microcontroller, in order to get the motors and sensors to behave as needed. An Adafruit soundboard and speaker were used to produce the soundbite. Once all the components were assembled in a circuit, we ran the program several times to adjust the code to get the movements right. Some troubleshooting had to be done, like changing the location of one of the motors, and adjusting the size of the structure to fit under a mask that was used to make the figure look like Professor Manteufel. Once completed we were able to demonstrate our ability to design and create a functional mechatronic system.

**Section 1: Literature review**

Animatronic face creation #1 (John Carroll)

    This review cover the the construction of the animatronic face called P.E.T.E.L.T. and can be seen at the link<http://www.youtube.com/watch?v=AhPnrvdMArM>

    The following is the instruction of the head construction, from the author knife141.  The head was built from 1/2 inch plywood. First a cardboard head and jaw were made, the design with experimented with to ensure the desired movement, ensure all the servos will have room to fit, and figure out the pivot points for the jaw and the head tilt.  This is where a laser engraver/cutter would be very beneficial. It would also be recommended to design in a CAD program, then laser the design onto the wood. But without one, trial & error. Provide sufficient documentation on the role of each servo, and provide how each servo connects to the component it controls. Once all the wooden parts of the head are cut out and test-fitted with servos, put a couple of coats of polyurethane varnish on the wooden parts. The heart of Robot Head 2's electronics is the servo controller and the RAPU.  The servo controller takes serial signals from the RAPU and translates them into servo movements. The RAPU takes programs written on a pc and stores both the movement commands and the audio portion. The RAPU is not entirely necessary. A pc could also be used for this task, but that would require laptop to be with Robot Head 2. A little single board computer was used for efficiency. Program each routine on a pc, then download it to a compact flash card on the RAPU and run it from there. All power supplies and the powered computer speakers are housed in the black wooden box beneath the head.  A fan in the box immediately beneath the head continually pulls cooling air through here to dissipate any heat buildup. Cut holes in the front of the power supply box to keep the speakers from sticking out too far, and fasten them in with hot glue. Mount a spotlight. The spotlight is more than just for illumination, its on/off position can be controlled via programming, and can also be used to augment a routine. For example, if you want to include the sound of thunder in a routine, you can also flash this light on and off to mimic lightening. The spotlight is made from a cheap clamp-on light which was torn apart and attached to a piece of  metal. At the other end an angle brace was attached, which allows for the mounting of the light. This light is controlled by a servo that opens/closes a micro switch that is in the box in the third photo. The switch box mounts on the back of the robot head's base and attaches to a servo control cable that sticks out of the back. On the back of the unit label some of the other connection/control points, including the serial connection (goes to the servo controller from the RAPU), the audio-in port (also connects to the RAPU), the master on/off switch, the main power input (goes to the power supplies), and the exhaust fan.

For this design, the creator used a single piece of wood as the rigid structure for the entire assembly, this seems to have provided a good amount of rigidity for the operation of the head. The mouth functions from a jawbone like joint attached to the main piece of wood, which is moved by a servo motor. The eyeballs are mounted onto a wooden piece that rotates to make the eyeballs look up or down, the rotation is actuated by a servo motor. Additionally the creator made the entire head rotate about the x and y axis, using an additional two servo motors. A single board was used to control the entire assembly of motors and speakers, and the purpose of the purpose of the creation was to provide entertainment to kids.

Animatronic face creation #2 (Andre Sirhan)

    One example found of an animatronics face was of a male actor, Greg Townley, shown in Figures 1 and 2 below. The project was made for a film called “The World's End” and can be seen at the link <https://www.youtube.com/watch?v=26fLK9m4eNE>. “

    The link previously shown is test footage, images, and a technical description of the animatronic built for Edgar Writes, from the author Matt Denton. “The build process starts with a lifecast of the performer's head which is cast out in wax clay, the eyes opened and generally cleaned up. A fibreglass mould is taken of this and a fibreglass core produced from it, which will be the basis for the mechanical understructure on which the skin will sit. Softened silicone, pigmented to the base colour of the performer's skin is poured into the mould and when it has set forms the skin for the head.”  The fiberglass core consists of 12 servos in the head, 10 for the eye and brows and 2 for the jaw/jaw slew, for it to be mechanised. The eye mixing and jitter was programmed by a iPuP-Ext unit, with dual 3 axis control sticks for puppeteering. The head was self contained utilising a magnetic latching battery compartment in the base of the neck, and a magnetic on.off switch buried below the skin. This was done for ease of power saving without battery removal.

    The purpose of this project was for it to be used in a scene in the film “The World’s End’ to replicate an actor whose head was knocked off against a urinal. The head was only required to perform a limited set of movements, therefore the functionality was mainly in the eyes, jaws, and brows.



Figure 1: Animatronic Head

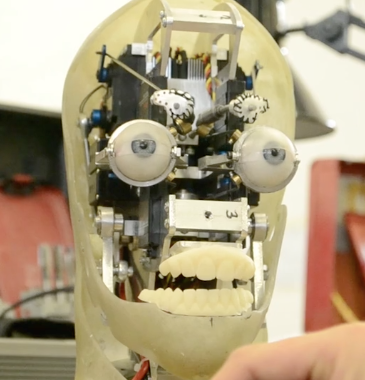


Figure 2: Animatronic Head without Skin

Animatronic face creation #3 (Rebeca Velasquez)

ToMoMi Gen 1 is an animatronic head kit which can be used for animatronics research. It can be seen in action at https://www.youtube.com/watch?v=npIW3sbjb4s. The structure is built with a composite underskull, polysiloxane face, CNC machined anodized aircraft aluminum sub-frames. It can be controlled by any PWM servo controller, microcontroller, or an RC system. The power supply recommended for use with the ToMoMi Gen 1 is a 6VDC power supply rated for 20 amps. The face has many motions it can display including moving the jaw in six directions, a range of motions with the mouth, opening and closing the eye lids and moving the eye brows. It uses Color Super Micro Video Camera PC207XP in each eye to allow for stereoscopic vision research. The motors used for the movements of the robot include one JR z9100s servo, two JR DS-378 low profile servos, and nineteen HS-82MG Hitec servos.



Figure 3: ToMoMi Gen 1

**Section 2: Brainstorming**

As per the project requirements, the mouth will open and close when unit is speaking. In addition we’d like to have eyelids that will open and close over the exposed eyeballs when desired. We’d also like to have a pair of sunglasses that would cover the eyes when desired, and a cigar that would protrude from the mouth opening when desired. The eyelids will be controlled by a single servo motor that will open and close them, the input for the eyelid servo motor will be controlled by the output of a photoresistor that will activate the motor when the light reaches a certain intensity. The mouth will be controlled by a single stepper motor that will rotate the jawbone around its axis. The stepper motor input will be controlled by the output of the audio file. The sunglasses will be controlled by a stepper motor that will be place at the axis of rotation around head, and the cigar will be controlled by a stepper motor that will linearly actuate it out of the head. Both the sunglasses and cigar’s input will be activated by the output of a ultrasonic sensor. The structure will be made out of a single vertical piece of wood that will be secured to a perpendicular wooden platform, this will provide a rigid platform for the components to be mounted to. For the face, we are going to put a mask over the wooden structure. We will secure the mask to the wood with fasteners. The eyeballs will be attached to the structure and the eyelids will be made with a cut ping pong ball.  The entire system, arduino, motors and sensors will all be powered by a 12 volt battery pack. The movement of the figure will be based on the soundtrack created for the project. The movements of the jaw will be timed closely to that of the words spoken in the sound track. We’d like the animatronics figure to speak audio sound bites that are commonly stated by Professor Randall Manteufel in his lectures, preferably ones that mention thermodynamics and heat transfer. An mp3 file, mini sound board, and speaker will be used to implement the soundtrack for the project.

**Section 3: Supporting structure**

The supporting structure and face was created out of scrap wood found in the garage. First, a square base was cut out of 1 inch thick plywood. The predicted weight distribution of the finalized structure, including the mounting placements for sensors and electronics, was considered so that the structure would remain stable without tipping over. With this consideration, a  4x4 inch square was chosen to give a solid platform. Next a side view of a human face was sketched and cut out of a piece of paper and then traced on plywood to be cut. The face was cut out using a hacksaw. A cut was made horizontally from the center of the lips towards the rear of the head to create a jaw. The cutout of the bottom jaw was traced on plywood and modified such that the rear of the jaw would overlap the original cut face up to the ears. This was then cut and placed on top of the face to find proper placement to mount jaw to symbolize natural jaw movement. Once desired mounting position was found a hole was drilled through the jaw and face simultaneously. Next, two rectangular pieces of wood were cut to be mounted vertically upward of bottom base. A hole was drilled through the blocks one inch from the end of length and cross center of width.  The face and jaw was placed in between the two rectangular blocks aligning the holes drilled on all parts. A wooden circular dowel was placed in the holes of the parts holding alignment. Figure (4) below shows the alignment and order of the parts arranged together.



Figure 4: Structure Assembly

This would both act as an axis for the jaw to move on the face and create a mounting position to the base. The bottom of the rectangular blocks were mounted onto base using wood screws leaving room for the jaw to move freely. Metal rods found in the chassis of radio controlled cars and trucks were used to support the rear of the face to base shown in Figure(5)



Figure 5: Rod used to support rear of face

**Section 4: Joints and motors**

For the mouth joint we drilled a hole through two pieces of wood, and then placed a wooden dowel through them to act as a pin. The dowel allowed the piece of wood acting as the lower jaw to rotate freely. One end of a rod was then attached to the piece of wood acting as the jaw, this can be seen in the figure below,



Figure 6. Rod used to rotate the mouthpiece around joint axis.

The other end attached to a servo motor, this would allow us to control the opening and closing of the mouth by controlling the servo motors position and speed.

For the eyelids we cut a ping-pong ball in half, both pieces of the ping-pong ball were then connected to a rod after running the rod through our head frame piece. This would allow us to rotate both “eyelids” by rotating the rod. A pulley was attached to the rod with the eyelids, and a servo motor, this can be seen in the figure below,



Figure 7. Pulley system used to rotate the eyelids around the eyeballs.

This would allow us to control the rotation position and speed of the eyelids about the eyeballs.

**Section 5: Sensors**

Two sensors were used to initiate movement of the figure. An HC-SR04 ultrasonic sensor was used to initiate the program of speech and mouth movement, this was placed on the structure platform as seen in the figure below,

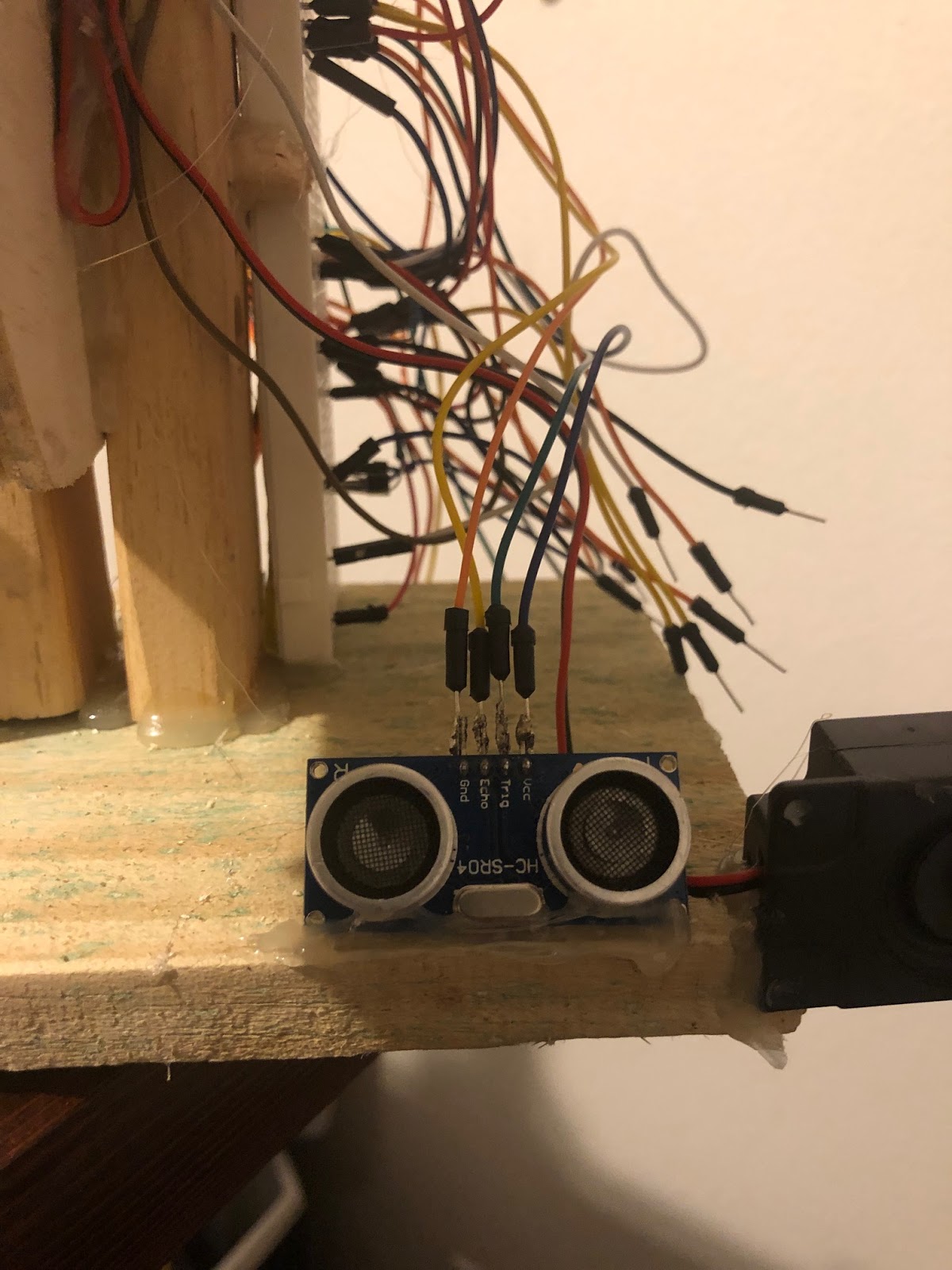


Figure 8. Ultrasonic sensor used to initiate actions and sound.

The ultrasonic sensor emits high frequencies and can measure the distance of objects in front of it by the time it takes to those waves to get back to its reciever.

A Photoresistor was used to initiate eyelid motion. The photoresistor was placed inside the eyeball, as seen in the figure below,



Figure 9. Photoresistor placed inside eyeball that initiates eyelid motion.

Placing the photoresistor in the eyeball allowed us to have the eyelid close when light was directed onto the eyeball.  A photoresistor is a variable resistor that changes its resistive value when it is exposed to light or removed from light.

**Section 6: Programming for interaction**

The Arduino programming language was used to control the movements of the figure. The language consists of three main parts: functions, for controlling the Arduino board and performing computations, variables, which are data types and constants, and structure, which are elements of C++ code [4]. First the variables had to be set, and the stepper and servo motor libraries had to be added to the program using the #include command. Several elements of the structure components were used to control when certain movements would take place. The setup() function was first used to initialize the pinMode() function of the components connected to the Arduino. This sets the pins as either inputs or outputs.

The loop() function was then used to continuously run the part of the program that was dealing with the reading of the sensors and movements of the motors. To take the readings from the photoresistor, the function analogRead(), was used to take the light values from the pin A0 on the Arduino and the map() function converted the light values into angles produced by the servo motor. The code to control the ultrasonic sensor was run next. It used digitalWrite() to send out the signal from the trig pin and utilized the pulseIn() function to time the signals received by the sensor and sent to the echo pin.

Using an if statement, we set the mouth to start moving, and the audio to start playing as soon as the ultrasonic sensor detected an object less than 10 centimeters away. The audio was triggered by using the digitalWrite() function to turn on the voltage to pin 2, which was used to connect the soundboard to the Arduino. For statements were used to control the speed and angles the mouth would move. It was to keep moving while the voice of Dr. Manteufel played. The cigar and the sunglasses were set to run after the mouth stopped moving and the voice stopped speaking. The audio continued to play music as these movements occurred. The servo for the sunglasses was controlled by different for statements. One to make them come down, and one to make them go back up. Like the mouth, these statements controlled the angle and speed which the motor moved. To control the stepper motor, we put in the stepSpeed() function to control the RPMs and, step() function to tell it the number of steps to take, the direction it would turn was determined by a positive or negative sign of the steps.  Once the music was done, the cigar was meant to go back into the mouth, and the sunglasses were meant to come back up. To stop the audio from playing continuously, digitalWrite() had to be used again to turn off voltage to pin 2. This was the end of the if statement for the distance of less than 10 centimeters.

The program also recorded the number of centimeters away an object was from the sensor, when it was between zero and sixty centimeters. Or it would simply state that it was more than sixty. These were the last commands inside the loop(). They were not related to any movements of the figure.

**Section 7: Lessons learnt and suggestions**

1. Jaw movement was weak due to the structured design. Movement of the jaw servo to make the jaw rod vertical would make more precise and stronger jaw movement. A higher torque servo for jaw would also help.
2. Sunglasses did not fit as intended around mask. The overall size of the face with the mask mounted was not taken into consideration. The sunglasses would rub against the side of the mask preventing movement. Use of wider sunglasses would fix the issue or support from inside of mask near the ears to mount mask smaller in width to fit sunglasses would have benefitted.
3. Motor/servo movements were different every time code was initiated or loaded from arduino. Creating seperate loops within the programming code for each motor may fix this issue. Initiating start and end points for servos and motors throughout the code may help keep things in sync also.
4. Cigar was hitting the bottom of the mouth when it was supposed to come out. The problem was fixed by moving the cigar, and motor controlling it, up higher. This could have been avoided if we had gotten the mask sooner and used it as a reference to see where the motor should have been attached.

**Section 8: Personnel and bill of materials**

**(a) Personnel**

|  |  |  |
| --- | --- | --- |
| Task | Main Personnel | Secondary personnel |
| Structure/chassis design including the face | Andre Sirhan | Rebeca Velasquez, John Carroll |
| Creating joints and motor interfacing | John Carroll | Rebeca Velasquez, Andre Sirhan |
| Integrating sensors and interfacing sensors | Rebeca Velasquez, John Carroll, Andre Sirhan |  |
| Overall programming and integration | Rebeca Velasquez | John Carroll, Andre Sirhan |

**(b) Bill of materials**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| No. | Description | Website/comment | Qty. | Unit $ | Total $ |
| 1 | Arduino MEGA 2560 | Provided | 1 |  |  |
| 2 | Small stepper motor | <https://www.sparkfun.com/products/10551> | 1 | 6.95 | 6.95 |
| 3 | Wood | Scrap | 1 | 8.00 | 8.00 |
| 4 | Servo kit | Amazon | 1 | 15.00 | 15.00 |
| 5 | Sensor kit | Amazon | 1 | 28.00 | 28.00 |
| 6 | Mask | <https://www.amazon.com/Fun-World-Putin-Political-Standard-x/dp/B072C52M15/ref=sr_1_5?ie=UTF8&qid=1544020371&sr=8-5&keywords=putin+mask> | 1 | 10.99 | 10.99 |
| 7 | Wig | <https://www.partycity.com/silver-fox-wig-482138.html> | 1 | 16.99 | 16.99 |
| 8 | Adafruit Soundboard | Amazon | 1 | 25.00 | 25.00 |
| 9 | Adafruit Amplifier/Speaker | Amazon | 1 | 5.00 | 5.00 |
| 10 | Duracell 9V Batteries | Lowes | 4 | 3.75 | 15.00 |
| 11 | stytofoam balls | Walmart | 2 | 0.50 | 1.00 |

The total price for this project excluding the Arduino MEGA was **$131.93.**

**References:**

# [1] Animatronics Face of UTSA's President, Dr. Ricardo Romo, <https://youtu.be/xkze1_hnam0>

# 

# [2] ROMOBOT - ANIMATRONIC FACE ROBOT

# <https://www.instructables.com/id/RomoBOT-Animatronic-Face-Robot/>

[3]Animatronics, <https://en.wikipedia.org/wiki/Animatronics>

[4] Arduino programming language, <https://www.arduino.cc/reference/en/>

**Appendix A: Code**

#include <Stepper.h>

#include <Servo.h>

#define trigPin 7

#define echoPin 6

Servo servo;

Servo servo1;

Servo servo2;

int sound = 250;

int pos1 = 90;

int pos2 = 180;

const float STEPS\_PER\_REV = 32;

const float GEAR\_RED = 64;

const float STEPS\_PER\_OUT\_REV = STEPS\_PER\_REV \* GEAR\_RED;

int StepsRequired;

Stepper steppermotor(STEPS\_PER\_REV, 10, 12, 11, 13);

void setup() {

servo.attach(8);

servo1.attach(9);

servo2.attach(5);

pinMode(trigPin, OUTPUT);

pinMode(echoPin, INPUT);

pinMode(2, OUTPUT);

Serial.begin (9600);

}

void loop() {

int lightValue = analogRead(A0);

lightValue = map (lightValue, 900, 1023, 0, 180);

delay(500);

servo1.write (lightValue);

long duration, distance;

digitalWrite(trigPin, LOW);

delayMicroseconds(2);

digitalWrite(trigPin, HIGH);

delayMicroseconds(10);

digitalWrite(trigPin, LOW);

duration = pulseIn(echoPin, HIGH);

distance = (duration/2) / 29.1;

if(distance < 10 ) {

Serial.println("the distance is less than 10");

digitalWrite(2, HIGH);

delay(500);

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

 }

 for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

 }

 for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

 }

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

for (pos1 = 90; pos1 <= 140; pos1 += 10) {

    servo.write(pos1);

     delay(50);

 }

for (pos1 = 140; pos1 >= 90; pos1 -= 10) {

    servo.write(pos1);

    delay(50);

}

delay (500);

for (pos2 = 180; pos2 >= 90; pos2 -= 5) {

    servo2.write(pos2);

    delay(50);

 }

 StepsRequired  = 2.25\*STEPS\_PER\_OUT\_REV ;

 steppermotor.setSpeed(900);

 steppermotor.step(StepsRequired);

 delay(800);

digitalWrite(2,LOW);

 StepsRequired  = -2.25\*STEPS\_PER\_OUT\_REV;

 steppermotor.setSpeed(900);

 steppermotor.step(StepsRequired);

  for (pos2 = 90; pos2 <= 180; pos2 += 5) {

    servo2.write(pos2);

    delay(50);

 }

}

if (distance > 60 || distance <= 0){

Serial.println("The distance is more than 60");

}

else {

Serial.print(distance);

Serial.println(" cm");

}

}