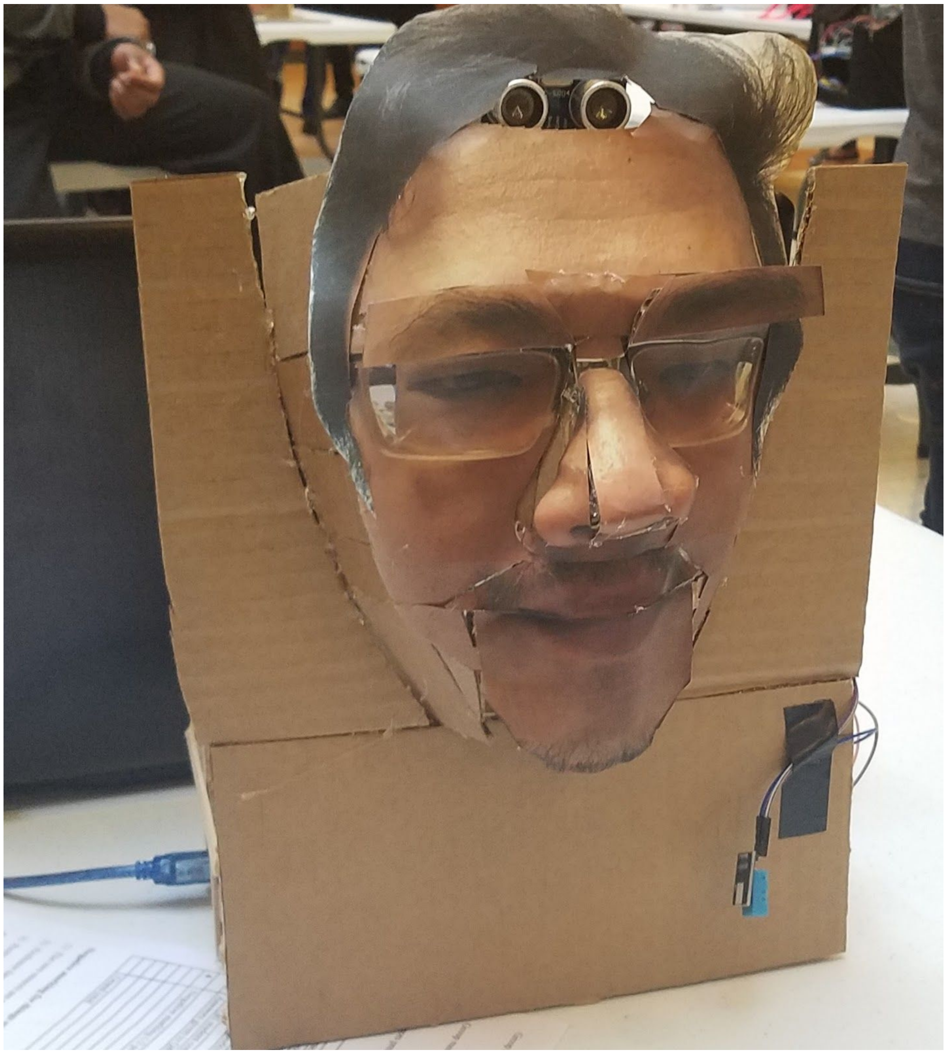
ANIMATRONICS FACE: PROFESSOR DONG



Youtube Link for Demonstration!

Nicholas Martinez
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
dfc898@my.utsa.edu

Ashford Ashiofu
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
ekc006@my.utsa.edu

Submitted as part of project for ME 4543 Mechatronics, Fall 2018

TABLE OF CONTENTS

ABSTRACT

The purpose of this project was to apply Mechatronics fundamentals to design, build, and program an animatronic face of one of the Department of Mechanical Engineering faculty members at UTSA. The project involved utilizing an Arduino microcontroller and then program parameters for two different sensors. The sensors were programmed to read physical inputs and then to actuate movement in a variety of motors such as servo or stepper motors for the facial expressions. A vocal recording was played to synchronize with the movements of the face to simulate the professor talking.

Section 1: Literature review

First Animatronic Face Creation: Kismet

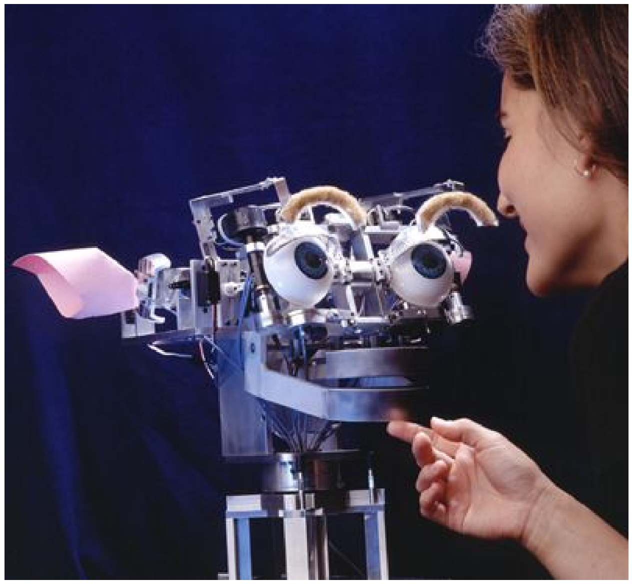


Figure 1: Kismet sociable humanoid robot

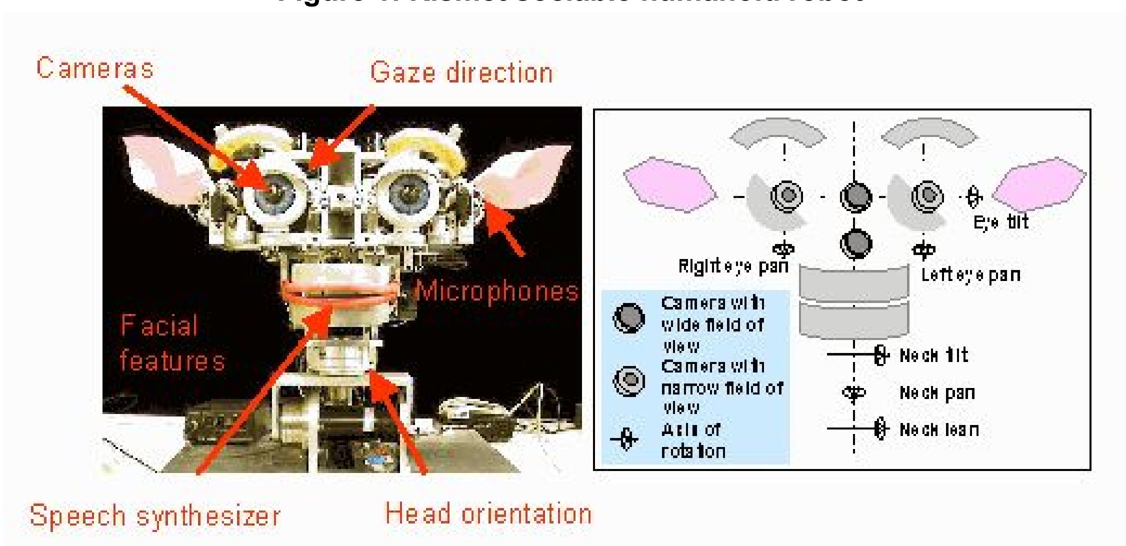


Figure 2: Anatomy and Controls of Kismet

Kismet is a robot with human communication characteristics. The design concept was to capture expressions as naturally as a human could through interaction. The robot has visual, auditory, and proprioceptive sensors and the motors control facial expressions and vocalizations. It responds to external stimuli to display communicative capabilities [1]. Kismet is equipped with a high level perception, motivation, behavior, and motor system. Four Motorola 68332 microprocessors running a proprietary Lisp code control the facial motor systems. Vision processing, eye and neck control are performed by nine PCs running QNX, a real time Unix based Operating System. Expressive speech synthesis runs on a PC using NT and speech recognition runs on a PC using Linux. The vision system uses four CCD cameras, two wide field of view cameras are mounted centrally. The expressive motor system allows for the face to have 15 Degrees of Freedom. Vocalizations are generated using an articulatory synthesizer based on the Klatt synthesizer which is particularly good at modeling the physiological characteristics of the human articulatory tract [2].

Second Animatronic Face Creation: From PyroElectro.com

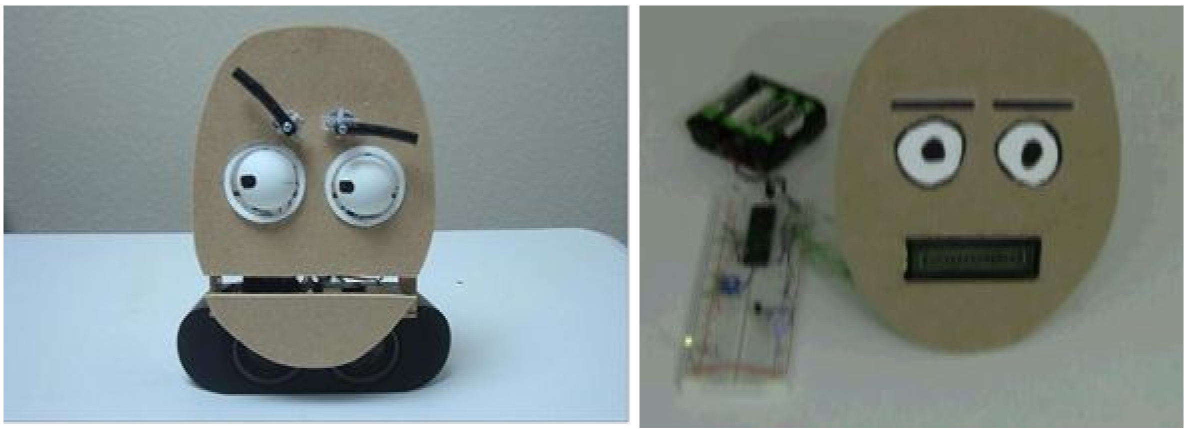


Figure 3: Animatronic Head from PyroElectro.com. Left is using the servo motor actuator, and the right is using an LCD configuration

This article, http://www.pyroelectro.com/tutorials/animatronic_mouths/, discusses two simple animatronic head designs that are both the same structure, but have different parts to achieve the end goal. Each design's purpose is focused around the mouth function, but the realism of expression is obtained from the other face features: eyebrows and eyes. This can be seen in the following link: https://youtu.be/KgCbQvBw-Bc. The first robotic mouth should be a mouth that moves up and down being articulated by a servo motor actuator.

The second design will consist of an LCD display with high power backlight that can be switched on and off. The other parts used for this project are outlined in Table _. Seven servo motors were used to actuate the movement seen in this animatronic face: 2 used for the eyebrows, 4 for the two eyes, and one to make the mouth go up and down. With the fact of only one axis mouth movement, not all mouth movements were attainable for this project.

A key highlight from this article is that human mouths don't move up and down, but rather in angular movements so taking this into account for my groups design will be beneficial. The mechanical and coding aspect for this design was enlightening. Seeing both a beginners design (LCD) and an intermediate one (servo motor actuator) gave two different perspectives that will benefit the design for this groups project.

Table 1: Parts for 2nd Animatronics Face Creation

Parts
HS-485HB
PIC 18F452
PICkit 2
7805 +5V Regulator
10kΩ Resistor
47uF Capacitor
20 MHz Crystal
Breadboard
Jumper Wire
SIPS
16x2 LCD Display
Wire Wrap
Wire Wrap Tool
4x0-80 Nuts and 3/4" Bolts
1/4" Acrylic Plastic or 1/4" MDF
Power Drill
E-Poxy
Glue

Third animatronic Face Creation: Humanoid Robot Head Roman

This article http://tcrn.ch/2uxbz8l is about the Humanoid Robot. It gives the behavior based emotional control architecture for the humanoid robot head ROMAN. The architecture is based on 3 main parts: emotions, drives and actions which interact with each other to realize the human like behavior of the robot. The mechanics of the humanoid head consist of basic unit of mounting plates which is fixed to the 4 DOF neck. These plates are the mounting points for the eyes, the servo motors and the cranial bone consisting of lower jaw, forehead and the back of the head. The skin is glued onto the cranial bone and can be moved with 8 metal plates , which are connected to 19 servos via wires. The neck of the of the face consist of 4 active degree of freedom. The electrical components will consist of a sensor vision camera, microphones and inertial systems. the actuator system of the robot consists of 21 different motors including electric, stepping and servo motors. All motors are connected to DSPs. The 6 stepping motors for the eyes (Faulhaber AM1524) are controlled by a single DSP. 2 additional DSP’s are needed to control the 4 electric motors (Faulhaber 2224 gear ratio 246:1) of the neck.

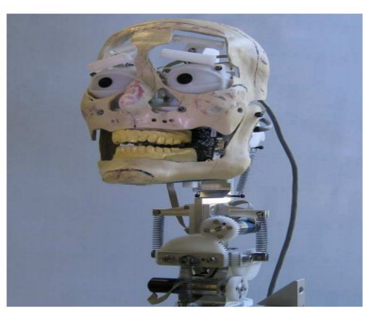


Figure 4: The humanoid robot head ”ROMAN

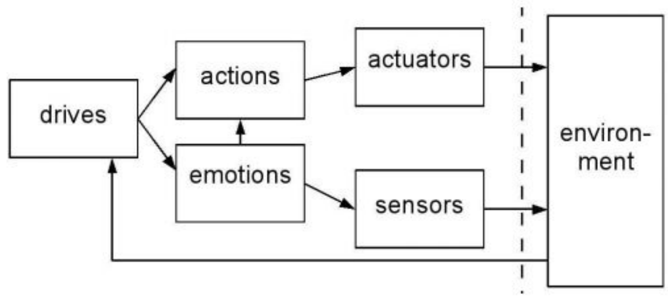


Figure 5: The design concept of the emotional architecture. The information flow of the main module groups (drives, emotions, actions, sensors, and actuators) is presented

Section 2: Brainstorming (initial planning)

The animatronics face should detect the proximity and motion of an object in front of it, such as a hand waving in front of it and then respond by moving its mouth and eyebrows. we will have a sensor placed above the eyes to help detect any movements. The joints will move via servo motors fixed to the frame. Data will be processed via an ultrasonic sensor for the proximity of motions for external movements.The sensor will be able to read no more than 3 inches away.

The frame will be made of popsicle sticks joined with hot glue. The face will be placed over a mask to give it more lifelike contours, a photo of the the professor's face will be cut up and pieces placed on the mask. The voice recordings will play when the sensors detect movement and the mouth should be synchronized to the voice lines. The Arduino, motors, and sensors will be powered via batteries or DC wall adapter. Several voice lines will be recorded from the professor, some catch phrases or humorous lines will be appropriate.The professor's voice will be replicated into our device and controlled by the sensor. we plan on collecting a soundtrack that deals with the professor teaching or instructing a class of students.

Section 3: Supporting structure

The bottom of the frame was made using wooden popsicle sticks and hot glue. One benefit of the wood is that it is non-conductive so the sensitive electronics won't need to be grounded. There are multiple platforms for which to mount the hardware such as the Arduino Microcontroller. The base will be flat and will most likely house the Arduino microcontroller

The bottom of the frame was made using a row of the popsicle sticks. It allows for a lot of surface area for the hardware to sit. The frames are made to be modular for quick additions for the wiring and the hardware to be mounted. This frame will give us a lot of flexibility and make it very easy to work with.

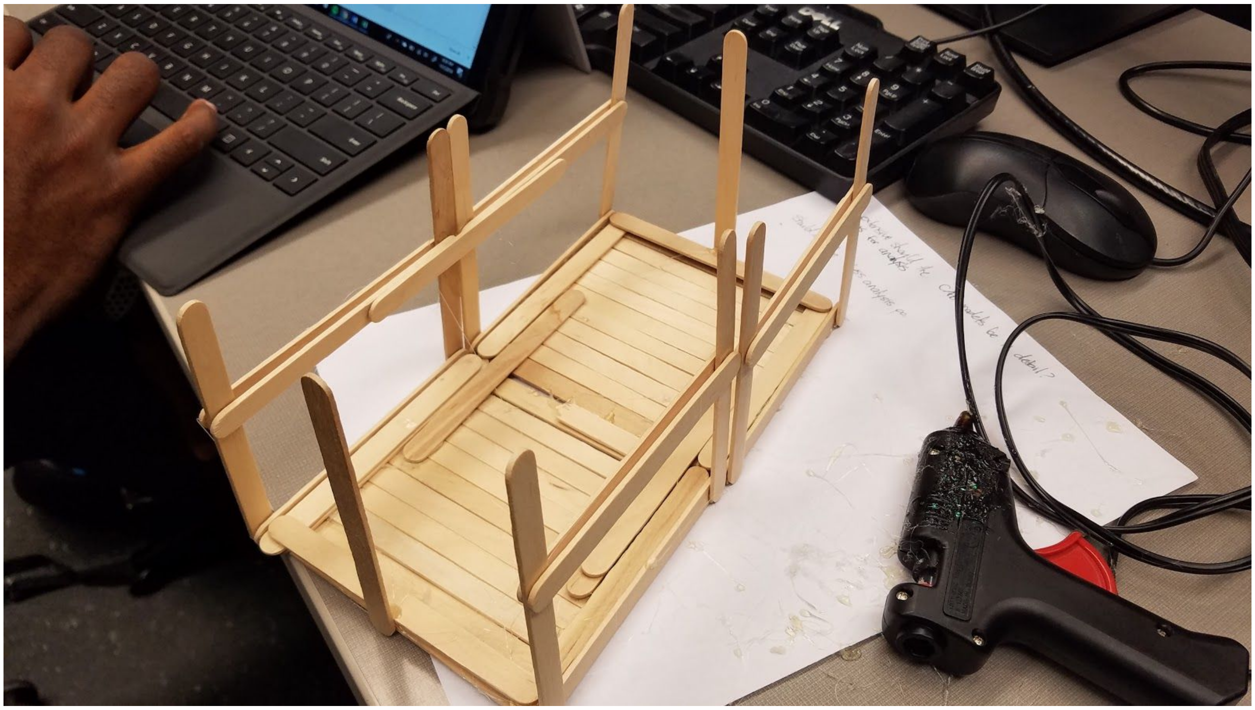


Figure 6: Base structure for frame with vertical supports

This final design for the frame will support the face that will eventually be attached and also support all of the following components needed to make the project successful.

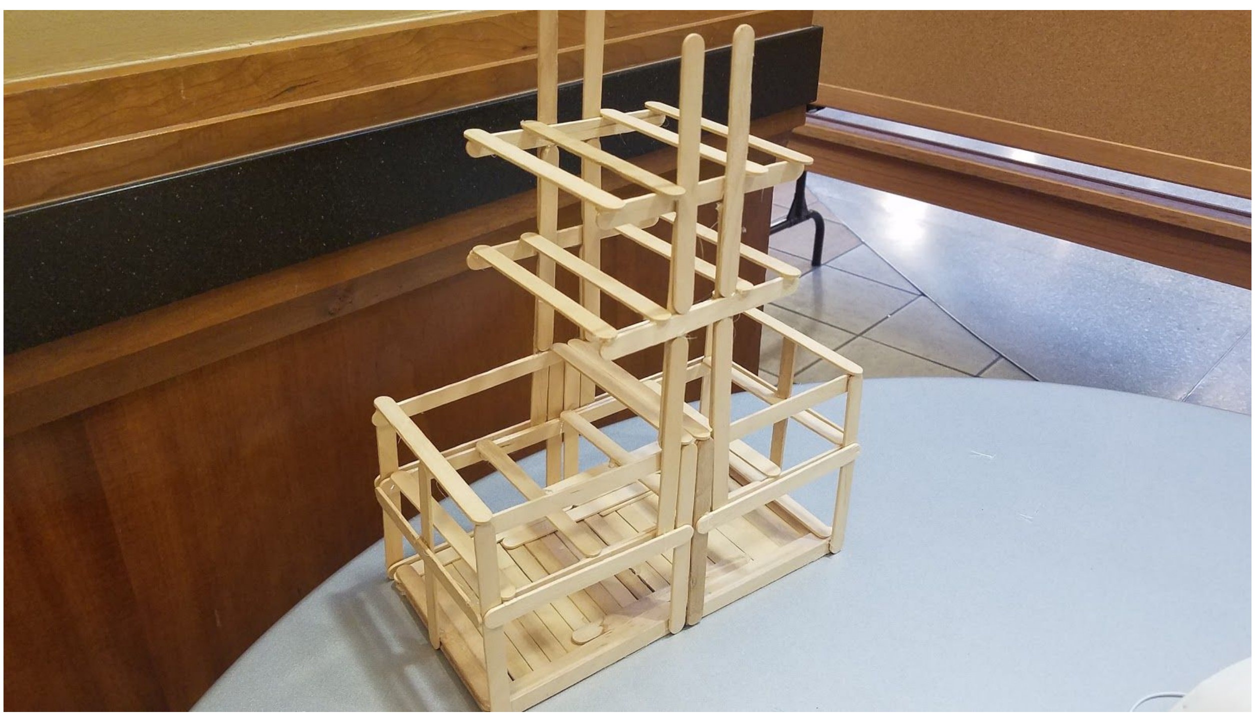


Figure 7: Base and Vertical support frame

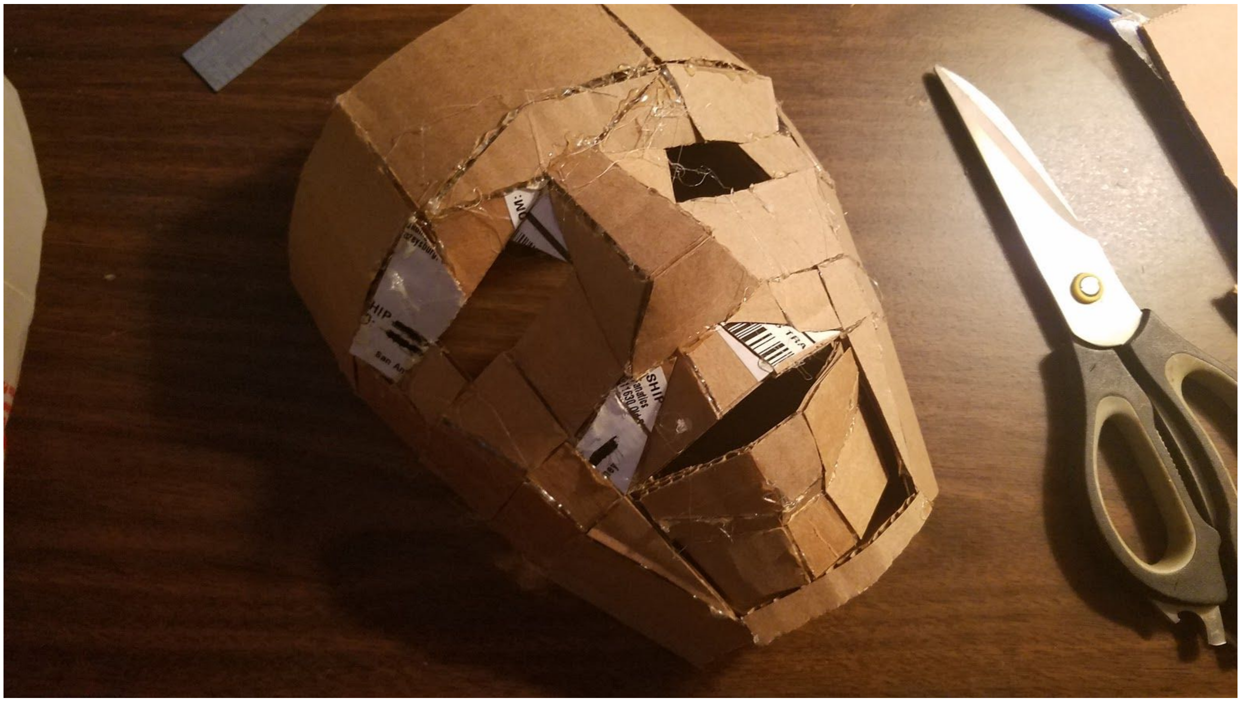


Figure 8: Cardboard mask made for 3D contours of face.

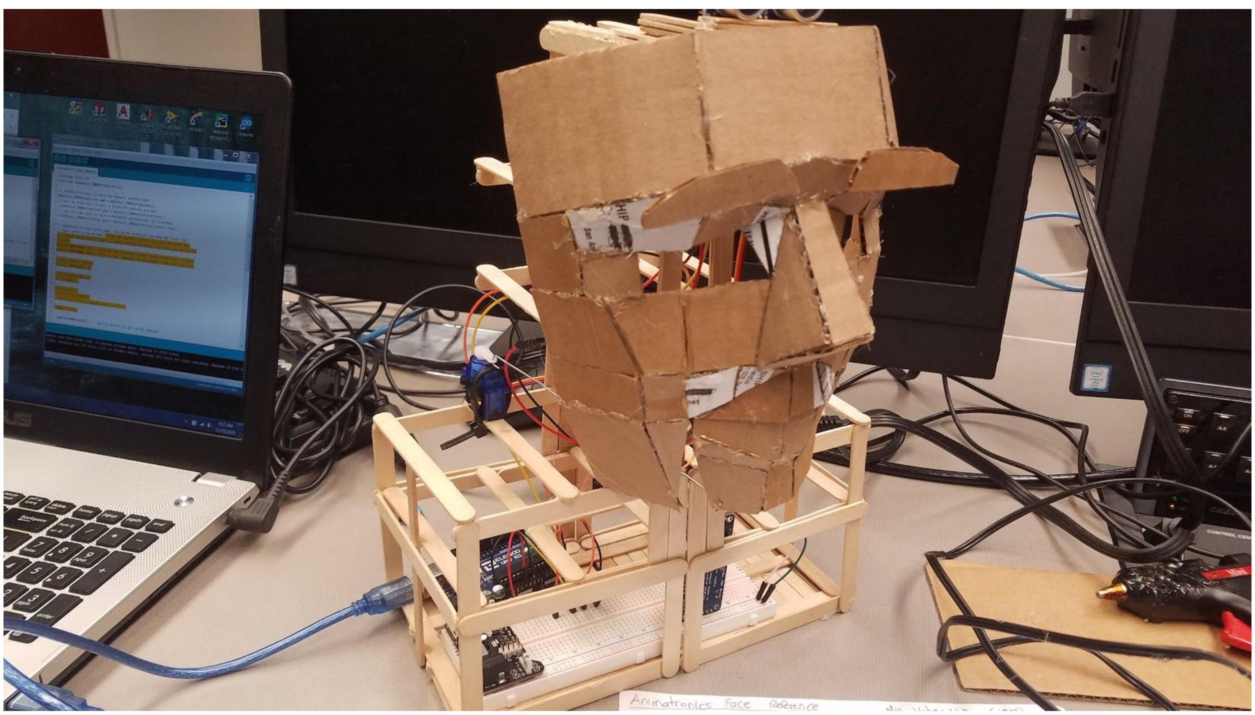


Figure 9: Mask mounted to frame with eyebrows

Section 4: Joints and motors

The joints of the mouth and eyebrows will be actuated using the servo motors. The mouth on the mask will be cut around the mouth to allow it to move. Fishing line or some other thin string will be attached to the servo motor and then to the mouth to pull on the mouth to give it movement. The eyebrows are attached with small pieces of a coat hanger that is glued to the servo motors to allow for rotation. Holes made in the front of the mask above the eyes were made to slot the eyebrows through and attached to the servo motors. This can be seen in **Figure 9** above.

The joints ended up being made with a paper clip attached to the rotating piece of a servo motor which controls the mouth as shown in **Figure 10** below. The eyebrow servos were placed onto the top platform from the main supporting structure. Duct tape was wrapped around rotating piece of the servo and a small piece of popsicle stick. Two small plastic pieces of a coat hanger were cut and then hot glued to the ends of the stick. They were poked out of the mask and pieces of cardboard shaped like eyebrows were hot glued to the ends.

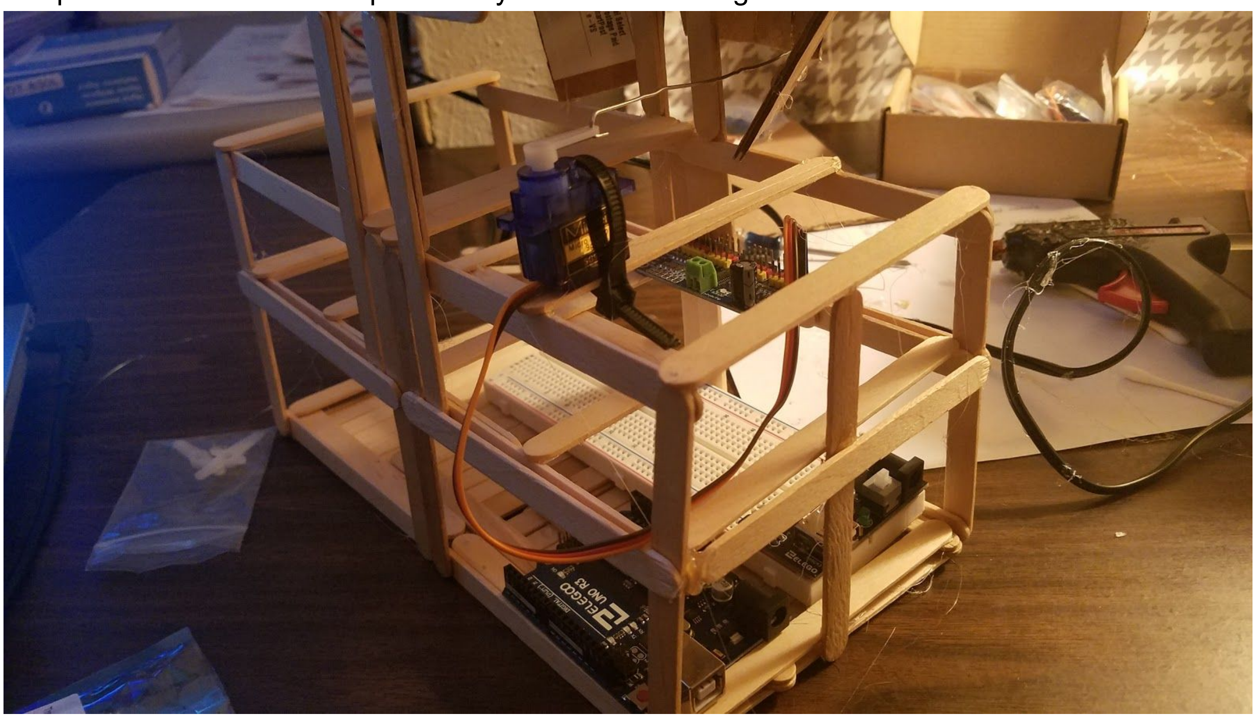


Figure 10: Rear of the frame to show servo motor for mouth movement and Arduino.

Section 5: Sensors

When choosing sensors, it was important to think of the simplest methods for user interaction possible. A good sensor in this regard is the proximity/ ultrasonic sensor which sends out a signal, bounces off of an object, and then returns to the sensor, giving a distance reading. When programmed with the Arduino, this allows for a conditional “if” statement to be made and depending on the distance, then actuates the movement for the servo motors, making the movement seem autonomous to a user who moves into the sensor’s range.

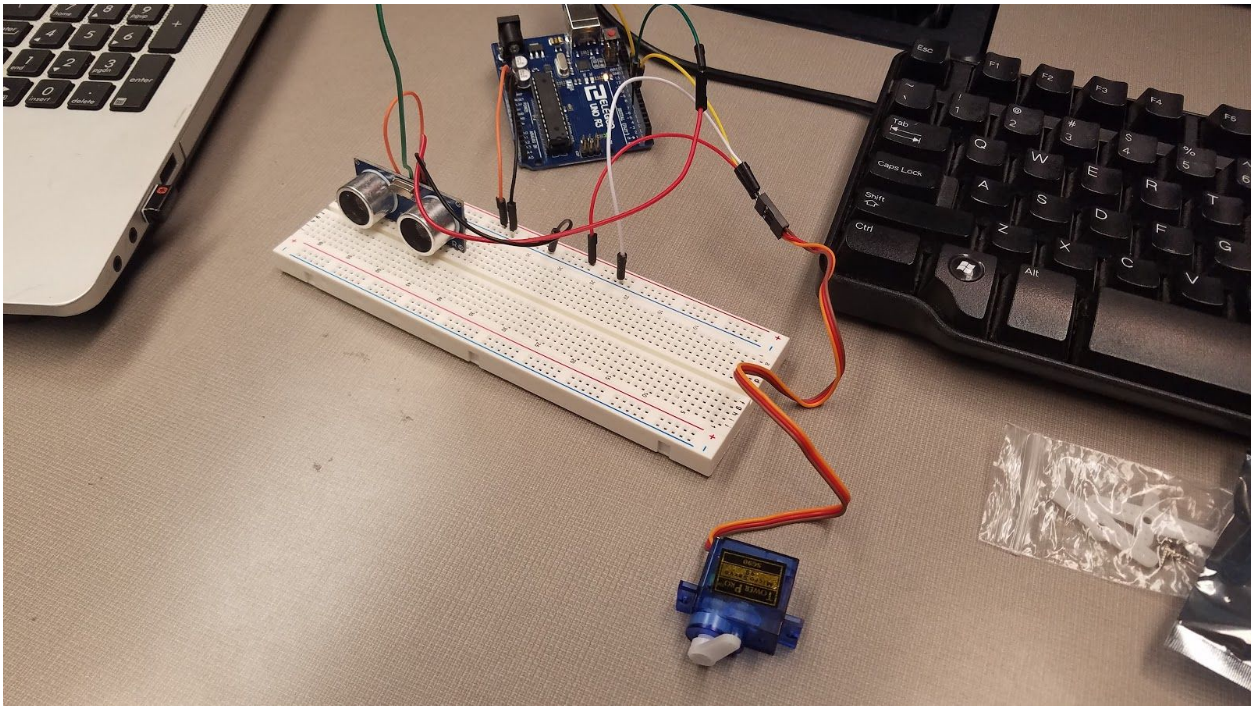


Figure 11: Initial testing of ultrasonic sensor controlling a servo motor.

The Ultrasonic sensor is good for this application as a person can wave or gesture their hand in front of the face and the face can react give the input. The temperature and humidity sensor will be calibrated to around room temperature and then with the increase of temperature, the servos can also be activated for movement and the audio queue.

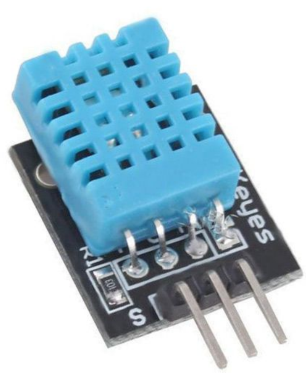


Figure 12: HC-SR04 Ultrasonic Sensor **Figure 13: DHT11 Temp. & Humidity Sensor**

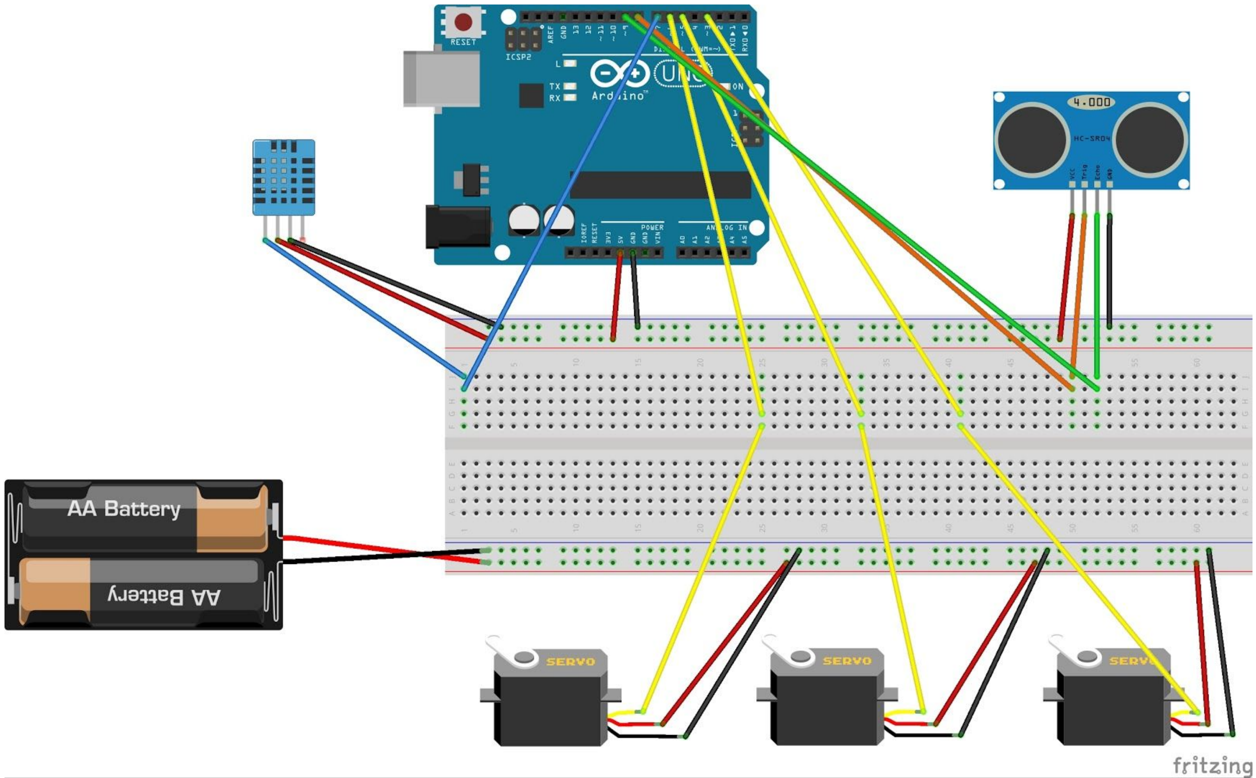


Figure 14: Wiring schematic for motors and sensors

Figure 14 above shows the wiring schematic for the project. An external voltage supply (we used a 9V battery for the demonstration) provided power to the servo motors. The servos are connected to the breadboard and then to the PWM outputs of the Arduino using jumper wires. The opposite power rail was connected the Arduino 5V and GND and provided power to the HC-SR04 ultrasonic sensor and the DHT11 temperature and humidity sensor.

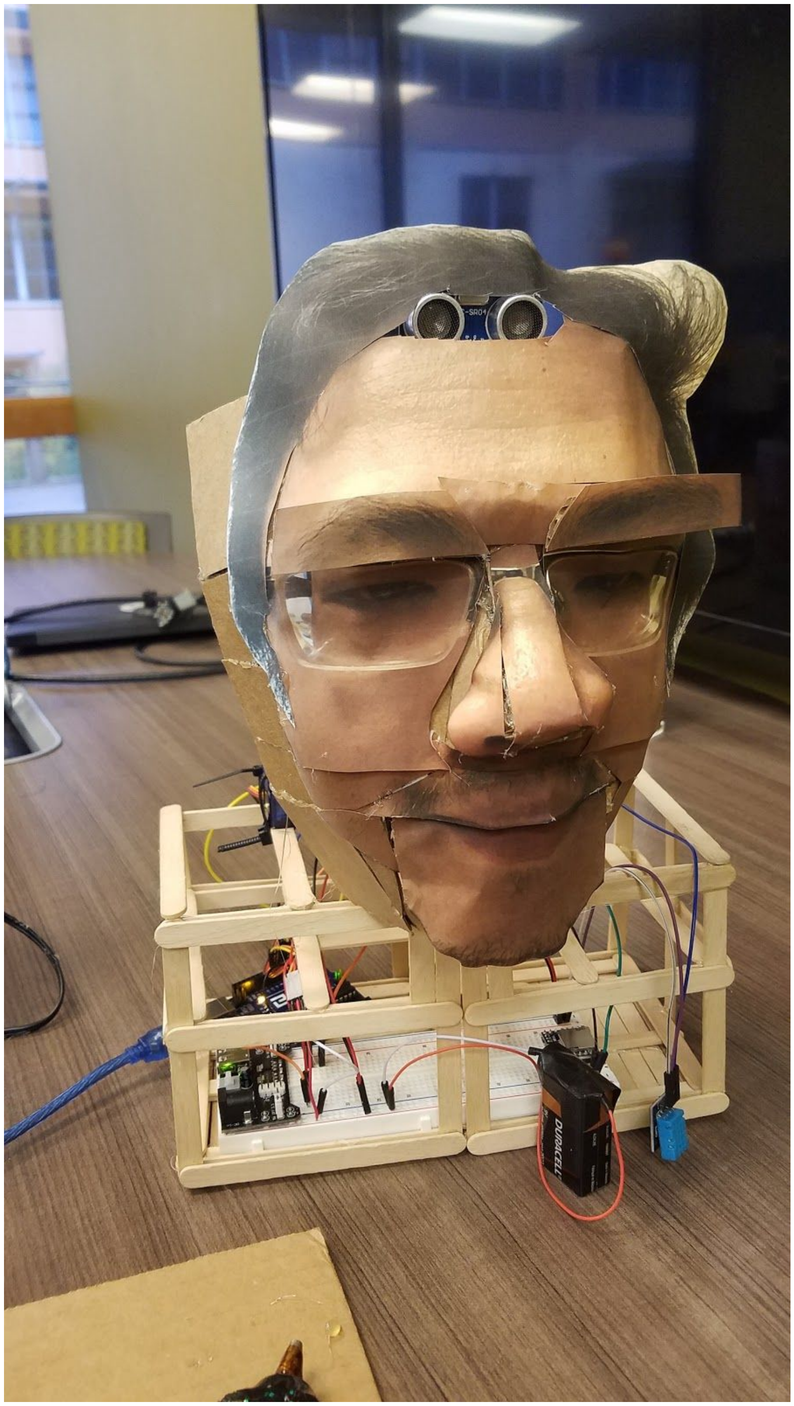
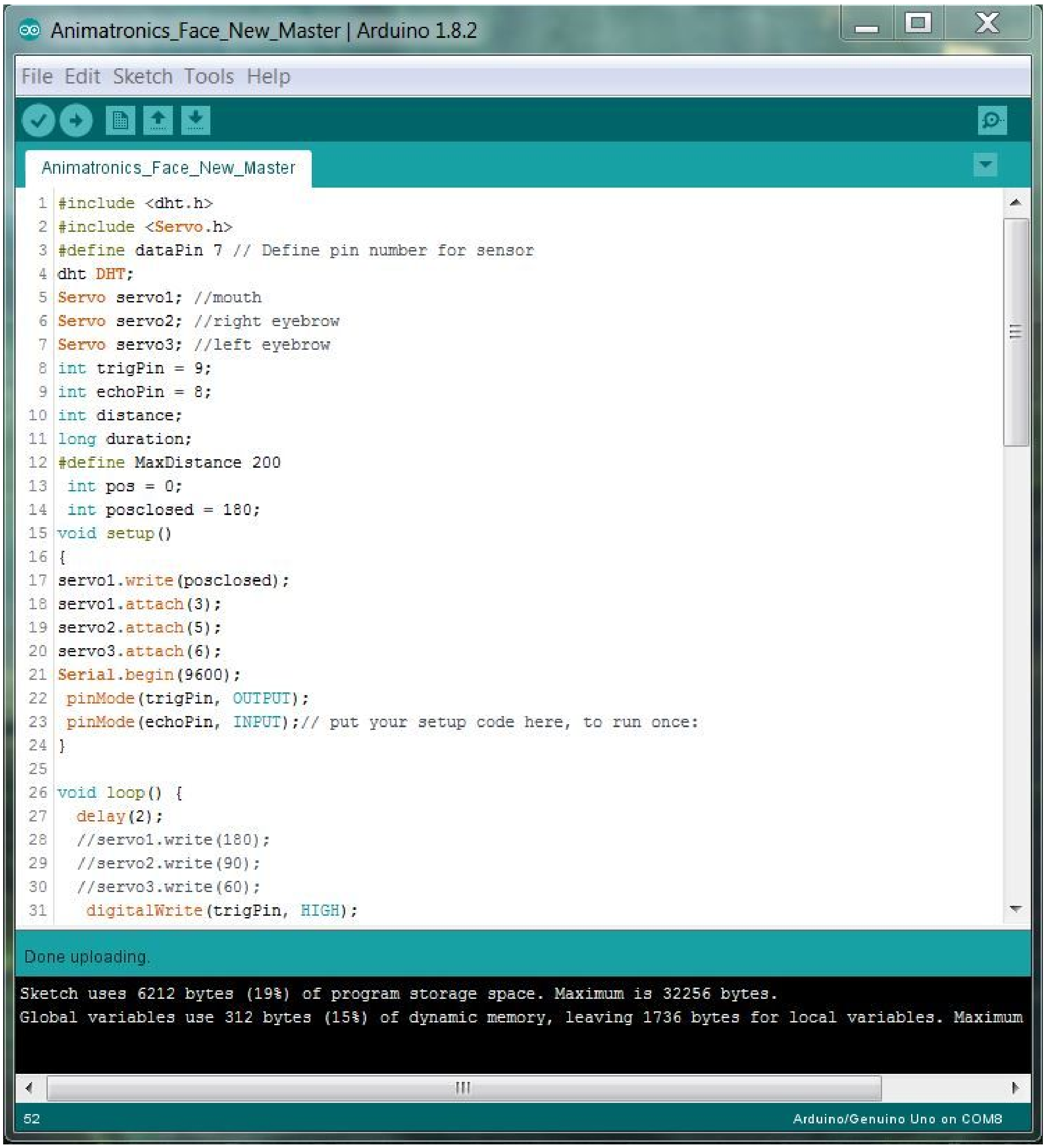


Figure 15: The animatronic face

Section 6: Programming for interaction

Programming the Arduino involved referencing the dht.h library for the DHT11 sensor, and Servo.h for using servo motors. The servos were defined and pins attached to the ultrasonic and temperature/ humidity sensors. Within the void loop, the ultrasonic sensor was referenced in order to then make an “if” statement where the condition that when the value was greater than or equal to some distance, it would actuate the servos to a specific angle and then delay. This was repeated several times along the code to give more variation to the movements made. An “else” statement was defined to return the positions of the mouth and eyebrows to their starting positions if there was no object within the range of the ultrasonic sensor.

The DHT11 temperature/ humidity sensor had a separate “if” statement to actuate the servo motors when the value for the humidity was greater than or equal to some humidity reading. This meant that if a person held onto the sensor, the humidity would increase, thereby actuating the motors to create the movement. The first part of the code is shown in **Figure 16** and the rest in Appendix A.




```
#include <dht.h>
#include <Servo.h>
#define dataPin 7 // Define pin number for sensor
dht DHT;
Servo servo1; //mouth
Servo servo2; //right eyebrow
Servo servo3; //left eyebrow
int trigPin = 9;
int echoPin = 8;
int distance;
long duration;
#define MaxDistance 200
 int pos = 0;
 int posclosed = 180;
void setup()
{
servo1.write(posclosed);
servo1.attach(3);
servo2.attach(5);
servo3.attach(6);
Serial.begin(9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);// put your setup code here, to run once:
}

void loop() {
  delay(2);
  //servo1.write(180);
  //servo2.write(90);
  //servo3.write(60);
   digitalWrite(trigPin, HIGH);
```


Figure 16: Arduino code

Section 7: Lessons learnt and suggestions

This project was a good exercise in learning and applying mechatronic principles in a fun way. The materials were not too expensive and it was encouraged to be creative with our designs. Something that was a struggle was properly adding an auditory component to the face. The initial plan was to have a DFPlayer Mini MP3 module attached to the breadboard and then into a small amplifier which were to drive 3 Watt speakers to provide audio. A code would be added to synchronize the audio with the movements of the mouth. However, this proved to be difficult just to physically implement the MP3 module as ours did not seem to function when testing with the amplifier and speaker. This was most likely due to incorrect wiring. An alternative is to get an all in one MP3 shield to mount on top of the Arduino pins and is fitted with a 3.5 mm audio out jack for easy connection to any speakers.

The project has increased the use of Arduino and overall knowledge of electronic components and integrating them into a mechanical system. Programming came from trial and error but if it would be good have a calibration for each sensor and the degrees of movement of each motor for more precise control of specific increments. Another thing that would have been interesting to have added were eyes that could move and rotate along with the eyebrows to give it a more lifelike appearance and realistic movements. We learned that completing weekly milestones much earlier than the due date was very helpful as the amount of work to conceptualize, build, test, and fully program the animatronics face was considerably high when factoring in work from other courses at the same time.

Some suggestions could include beginning preparation for the project earlier on in the semester and perhaps involving some of the labs to learn how to implement servo motors and sensors to obtain readings. This way, it can lead into the project with much of the skills already covered during the regular lab times.

Overall, a very fun project. Enjoyed presenting it and learning how to apply skills from the Mechatronics course and labs.

Section 8: Personnel and bill of materials
(a) Personnel

Table 2:

Task	Main Personnel
Build Frame	Nicholas Martinez, Ashford Ashiofu
Test sensors with motors	Nicholas Martinez, Ashford Ashiofu
Idea generation, brainstorming	Ashford Ashiofu, Nicholas Martinez
Ordered parts	Nicholas Martinez, Ashford Ashiofu
Constructed cardboard mask	Nicholas Martinez
Visit Professor Dong, record voice, take picture	Nicholas Martinez, Ashford Ashiofu
Assembled arduino, wired components	Nicholas Martinez
Cut paper face, attach to cardboard mask	Nicholas Martinez
Create, test, and run code programs	Nicholas Martinez
Present finished animatronic face	Nicholas Martinez, Ashford Ashiofu
Write final report	Nicholas Martinez, Ashford Ashiofu

(b) Bill of materials

I owned this kit already. Many of the components came with an Arduino Uno kit purchased from Amazon here. Components from the kit are designated with the comment "Student supplied."

https://www.amazon.com/Elegoo-EL-KIT-001-Project-Complete-Tutorial/dp/B01CZTLHGE/ref=sr_1_7?ie=UTF8&qid=1543981166&sr=8-7&keywords=arduino+uno

Table 3:

No.	Description	Website/comment	Qty.	Unit $	Total $
1	Elegoo Arduino Kit	Includes everything "student supplied"	1	53.99	53.99
2	Arduino Uno Rev 3	Student supplied	1	N/A	N/A
3	HC-SR04 Sensor	Student supplied	1	N/A	N/A
4	DHT11 Sensor	Student supplied	1	N/A	N/A
5	Power Supply Module	Student supplied	1	N/A	N/A
6	Breadboard	Student supplied	1	N/A	N/A
7	Jumper Wires	Student supplied		N/A	N/A
8	Servo Motor SG90	Bought lot of 10	3	5.39	17.99
9	Zip Ties	100 pack	4	0.99	3.97
10	9V Battery	Pack of 2	1	3.90	7.80
11	Popsicle Sticks	1000 count	1	6.07	6.07
12	Hot Glue	100 stick pack	1	2.22	2.22
13	Paper Clip	100 pack	1	0.02	1.37
14	Plastic Coat Hanger Pieces	Cut 2 pieces for eyebrows	2	N/A	N/A
15	Duct Tape	1 roll	1	3.37	3.37
16	Cardboard	Cut up boxes	N/A	N/A	N/A

Acknowledgements

A special thank you to Professor Bing Dong for allowing us to use his face and voice for the project. Thank you to the UTSA College of Engineering.

References:

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

[] ROMOBOT - ANIMATRONIC FACE ROBOT
https://www.instructables.com/id/RomoBOT-Animatronic-Face-Robot/

[1] Kismet
http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html
[2] Kismet, the robot The Hardware Design
http://www.ai.mit.edu/projects/sociable/baby-bits.html

[] DHT11 & DHT22 Sensors Temperature and Humidity Tutorial using Arduino
https://howtomechatronics.com/tutorials/arduino/dht11-dht22-sensors-temperature-and-humidity-tutorial-using-arduino/

Appendix A: Code

Animatronics_Face_New_Master | Arduino 1.8.2

File Edit Sketch Tools Help

Animatronics_Face_New_Master

```
#include <dht.h>
#include <Servo.h>
#define dataPin 7 // Define pin number for sensor
dht DHT;
Servo servo1; //mouth
Servo servo2; //right eyebrow
Servo servo3; //left eyebrow
int trigPin = 9;
int echoPin = 8;
int distance;
long duration;
#define MaxDistance 200
 int pos = 0;
 int posclosed = 180;
void setup()
{
servo1.write(posclosed);
servo1.attach(3);
servo2.attach(5);
servo3.attach(6);
Serial.begin(9600);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);// put your setup code here, to run once:
}

void loop() {
  delay(2);
  //servo1.write(180);
  //servo2.write(90);
  //servo3.write(60);
   digitalWrite(trigPin, HIGH);
```

Done uploading.

```
Sketch uses 6212 bytes (19%) of program storage space. Maximum is 32256 bytes.
Global variables use 312 bytes (15%) of dynamic memory, leaving 1736 bytes for local variables. Maximum
```

52 Arduino/Genuino Uno on COM8


Animatronics_Face_New_Master | Arduino 1.8.2
File Edit Sketch Tools Help
Animatronics_Face_New_Master
_delay_ms(500);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2)/29.1;
int readData = DHT.read11(dataPin);
float t = DHT.temperature;
float h = DHT.humidity;
Serial.print("Temperature = ");
Serial.print(t);
Serial.print(" *C ");
Serial.print(" Humidity = ");
Serial.print(h);
Serial.println(" % ");
if(distance <= 30){
//My
servo1.write(0);
delay(150);
servo2.write(55);
delay(75);
servo3.write(77);
delay(200);
servo1.write(180);
delay(200);
// Research
servo1.write(0);
delay(150);
servo2.write(90);
delay(50);
Done uploading.
Sketch uses 6212 bytes (19%) of program storage space. Maximum is 32256 bytes.
Global variables use 312 bytes (15%) of dynamic memory, leaving 1736 bytes for local variables. Maximum
52
Arduino/Genuino Uno on COM8


Animatronics_Face_New_Master | Arduino 1.8.2
File Edit Sketch Tools Help
Animatronics_Face_New_Master
  servo2.write(90);
  delay(50);
  servo3.write(60);
  delay(50);
  servo1.write(180);
  delay(200);

  // And
  servo1.write(0);
  delay(250);
  }
  else{
    servo1.write(180);
    servo2.write(90);
    servo3.write(60);
  }
if(h >= 65){
   servo2.write(55);
  delay(300);
  servo3.write(77);
  delay(300);
  servo2.write(100);
  delay(100);
  servo3.write(70);
  delay(100);
}
 Serial.print("Distance: ");
Serial.println(distance);

}
Done uploading.
Sketch uses 6212 bytes (19%) of program storage space. Maximum is 32256 bytes.
Global variables use 312 bytes (15%) of dynamic memory, leaving 1736 bytes for local variables. Maximum
52
Arduino/Genuino Uno on COM8

Appendix B: Arduino wiring