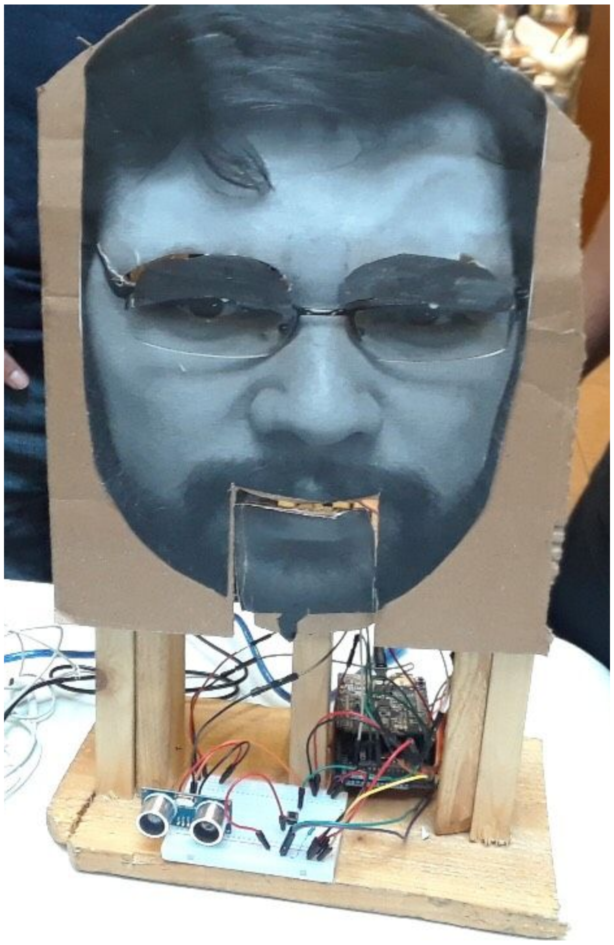
ANIMATRONICS FACE: ASSISTANT PROFESSOR, DR. LYLE HOOD



Nicholas Lopez
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Zky001@my.utsa.edu

Jesus Madrigal
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Qpi967@my.utsa.edu

Joseph Turrubiates
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Bae514@my.utsa.edu

Submitted as part of the project for ME4543 Mechatronics, Fall 2018

TABLE OF CONTENTS

ABSTRACT

Professor Lyle Hood was the assigned professor this group had to portray onto an animatronic face. The project requirements were as follow: 1) the face would need it’s jaw to move with an audio (no less than five words) from assigned professor being played simultaneously, 2) another facial joint was mandatory to show some type of motion, and 3) a sensor would trigger (1), while another would prompt (2), totaling up to a minimum of two sensors being needed, and all must be programmed onto one Arduino. Besides the required jaw movement, the eyebrows were the other joint motion this project was chosen to be displayed. The driving forces that led to each gesture’s action were servo motors. These motors were triggered by an ultrasonic sensor (HC-SR04) and button device. The button was not in the initial design; a motion sensor, followed by the touch sensor, were the first devices experimented with but were eventually concluded by the group that both module’s flaws proved to be un-credible towards the project's goal. Nonetheless, the ultrasonic sensor and button were already pre-owned which added another reason to implement them into the design. The structure keeping everything intact was kept as simple as possible to allow for flexibility with the adjustments throughout the design process more doable. It was constructed out of wood, with nails either being bolted or hammered to keep everything in place. Overall, it was a fascinating project that involved much coding and team collaboration that made this assignment a success.

Section 1: Literature review

First Animatronic Face Creation: Kismet

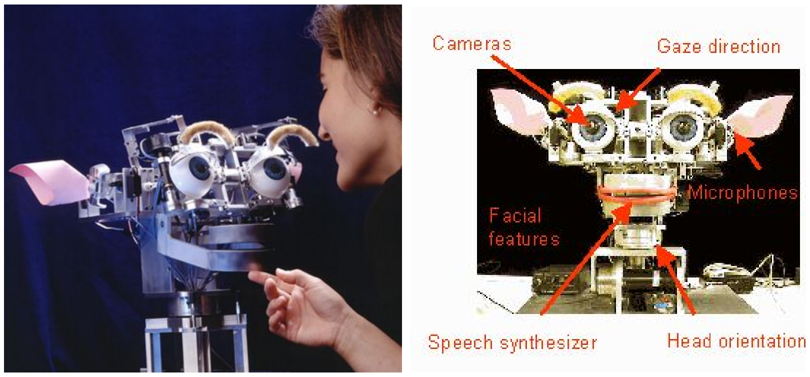


Figure 1: Kismet Robot

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 fields 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: RomoBot

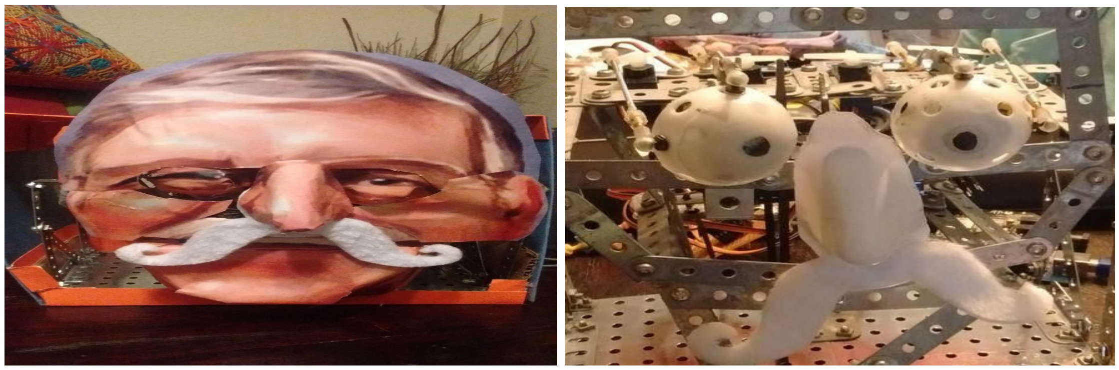


Figure 2: RomoBot

The Robot was made of UTSA ex-president as seen in figure 2. The reference [3] shows the link found of the RomoBot that provides steps of how the bot was initially built. Materials needed to complete the project was an erector set, clear acrylic board, servo motors, wooden shaft, Arduino mega, MP3 Arduino shield, push rods, plastic mask, Micro SD card, wires, and MDF board. The tools used was Allen wrench, hot glue gun, Drimmel tool, soldering iron, solder, wire strippers, butane torch, epoxy, and wood glue. All these materials and tools were needed to build and complete the RomoBot. One of the features wanted was having a moving mechanism on the eyes which was accomplished by using 4 servo motors 2 for each eye each for a different axis on the x-y. Face structure included the jaw, nose, eyes, and a mustache. The moving parts would be the mouth, nose, mustache, and the eyes. Which included audio using the MP3 shield to say actual words from the UTSA president.

Third Animatronic Face Creation:

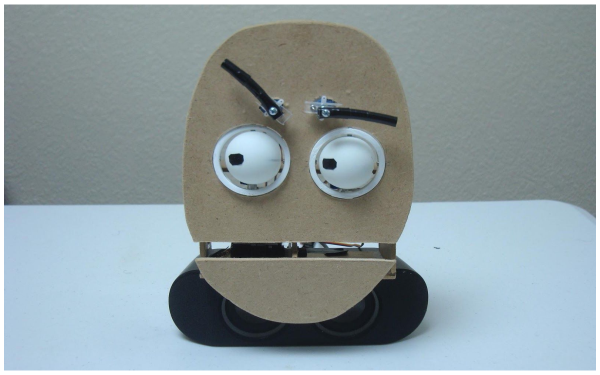


Figure 3: Animatronic Face

The first animatronic face featured moving eyes, eyebrows, a neck allowing freedom of movement along the φ and θ elements, and a mouth. Instructions provided by the creator [5] shows that the materials used to 0k9j8m the face were mainly MDF, the eyes were from table tennis balls. The wiring was done on a breadboard using jumper cables. The eye movements were done using a two-axis gimbal system and required two servo motors for each eye to accomplish. There was another two servo motors for the eyebrows, two for the neck, and a single motor with two mounts for the mouth for a total of seven.   The mounting was done partially with epoxy, and some were done using 4-40 bolts and nuts. A PICkit 2 microprocessor was used with a PIC 18F452 microchip to connect the servo motors independently. It relied on AA batteries.

Section 2: Brainstorming (initial planning)

This animatronic face design process first involved doing research on possible sensors that would be integrated into the project. Table 1 shows a list of sensors that were recorded after research was completed. After which, the following criteria were considered when deciding the two sensors to use for the project: environment project would be in, code availability online, and cost. The two sensors that aligned best with this gauge of judgment were the ultrasonic sensor (HC-SR04) and HC-SR501 PIR motion sensor. Confirmation to proceed with them was received by the professor. From this point, deliberation on the entire face project was the next topic of discussion. The facial features to move, not considering the mouth, that are currently being examined is the eyebrows or ears. Servo motors will be their mode for movement. The structure that will keep this project stabilized will be made of wood just because of its easy access and constructability.

Table 1: List of the first set of sensors after doing research online

First List of Sensors
Temperature Sensor
Proximity Sensor
Accelerometer
IR Sensor
Pressure Sensor
Light Sensor
Ultrasonic Sensor
Gas/alcohol Sensor
Touch Sensor
Color Sensor
Humidity Sensor
Tilt Sensor
Flow/Level Sensor
Magnetic Sensor
Knock Sensor

One of the resources that were deemed necessary for this project is an MP3 Shield. The time frame permits us to search for purchasing products online and having them delivered to us. However, there was much consideration on what criteria should be used to weigh the options against one another. It was deemed that our options would be decided upon by considering the compatibility with our Arduino model, the price, the acquisition time, the formats of the files that could be played, and options it would provide to produce the sounds. The reviews from other patrons would be considered to provide clear indicators of any shortcomings of the products being considered. Any resources the distributors would provide in the way of tutorials and instructions would also be a deciding factor, as it would make the programming and integration with the Arduino that much smoother.

It was deemed necessary that the MP3 shield has a capacity for several tracks, being able to play the audio through a speaker, be able to trigger each track without direct contact with the MP3 shield and to be able to either be powered by the Arduino board or have a portable supply of power. Ideally, we intend to record lines from our Professor that reflect the initial approach and movements of the individual that is interacting with the animatronics figure.

The animatronics figure shall be able to talk and move using the sensors and servo motors to meet the requirements for the project. The servo motors will be moving the joints chosen to move the moving parts once the sensor activates. The top picks for the moving parts for the animatronics project are the ears, nose, or the eyebrows. The mouth is a required item that needs to move because it would be talking due to the MP3 shield. Another way instead of the MP3 shield is synchronizing the laptop that will power the animatronic face to talk instead of using MP3 shield as recommended by the professor. The assigned professor will be Dr. Hood which an appointment has been made to record words of his preference and complies with the project objectives being 5 words.

Section 3: Supporting structure

It can be seen in Figure 4 the initial design was based to resemble, to some degree, a football goal. This simple model allowed the group substantial room for furthering the necessary design adjustments that were going to be needed to move the moving parts of the animatronic face. It is important to note that at the point the image was taken, it was still unsure as to how exactly the moving parts and sensor would be integrated into the system so the fact that this design was flexible made it that much more desirable.

Figure 5 was the next and final phase of the supporting structure. It was planned, at the point of constructing this model, that a fishing rod would be the mode for connecting the servo motors to the moving parts. The rightmost image illustrates how the middle picture is being stabilized. The leftmost image shows a screw which is what the rod was supposed to go around to pick up the jaw. Upon further analysis, it was understood by the group that swapping the screw feature with a servo motor (as can be seen in Figure 6) was the most efficient way to complete the jaw up and down as compared to relying on the screw joint. The entire structure was constructed through means of bolting/hammering nails into the wood, obtaining already owned wood, and sawing the wood as saw fit by the group.

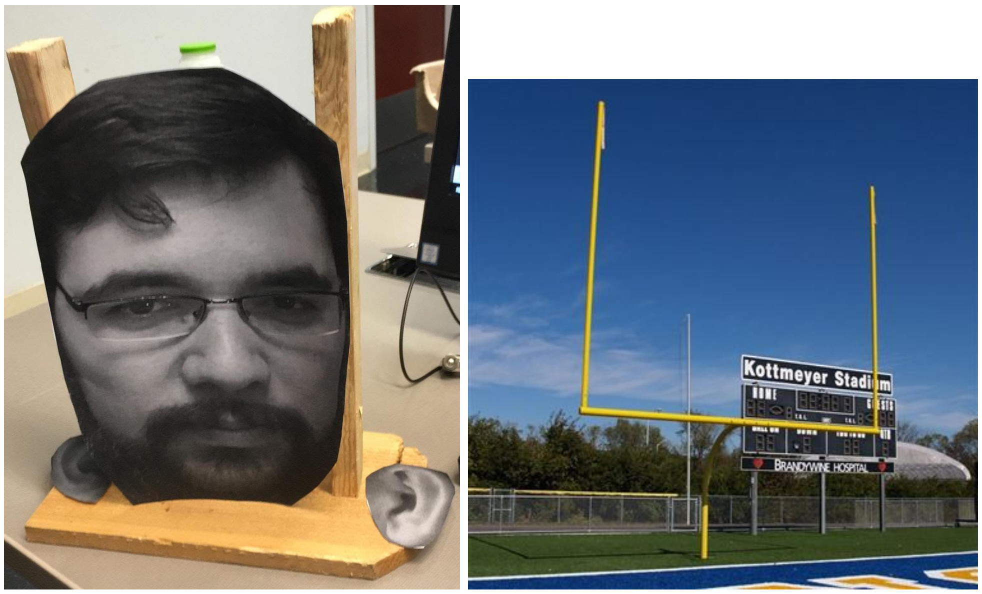


Figure 4: Initial Design of Supporting Structure

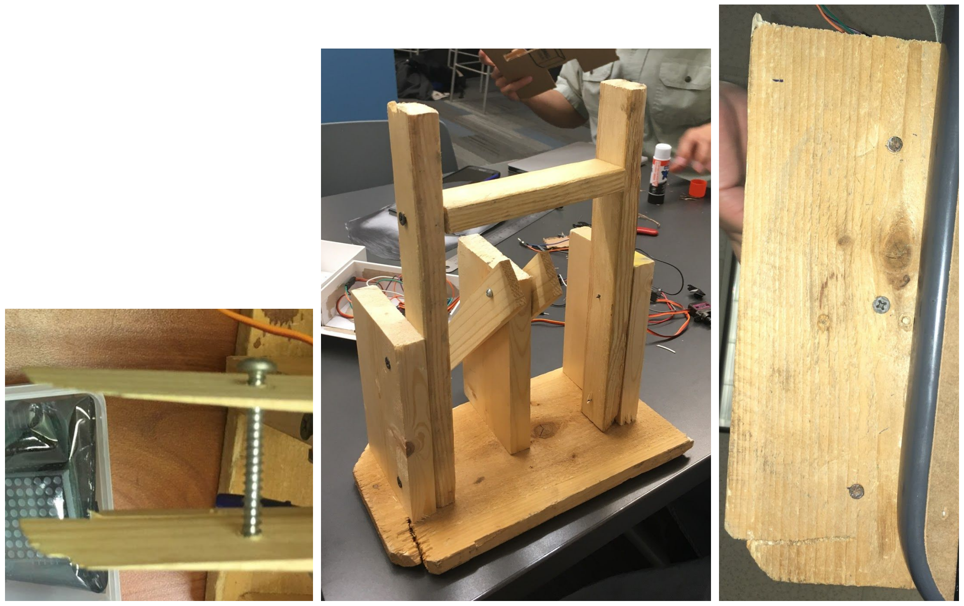


Figure 5: Next Phase of Supporting Structure

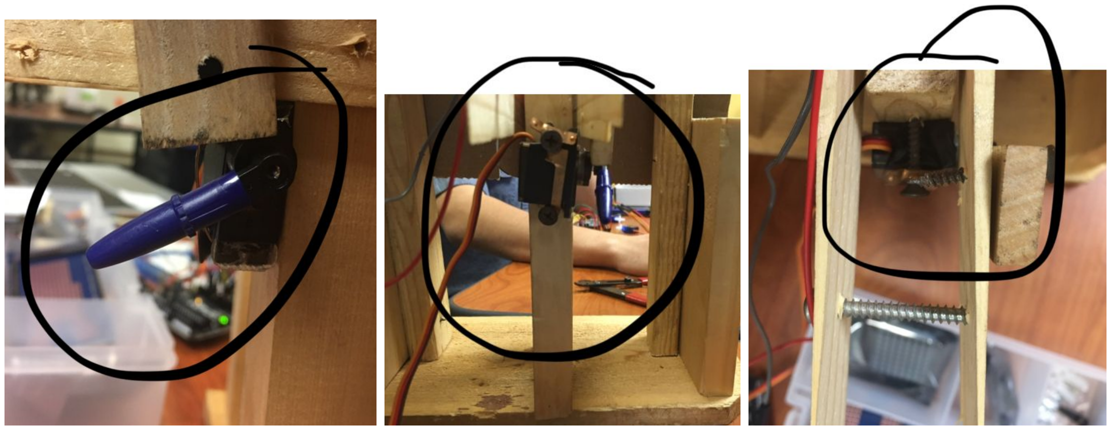


Figure 6: Minor Adjustment, different views

Section 4: Joints and motors

Servo motors (SG90) were chosen to create the movement for the following facial features: jaw and eyebrows. The servo motors were the best option considering they were already pre-owned and many codes online could be found for them. Fear of them not being able to withstand the pulling or pushing action required to get the joints to move was a worry the group had. To maximize full effectiveness in this category, metal servo motors were obtained from a fellow colleague who took mechatronics last semester. Figure 7 & 8 illustrates how the servo motor will create the jaw movement. As for the eyebrows, figure 9 exhibits how exactly the movement for them will be accomplished. The joints were constructed through means of bolting/hammering nails into the wood in the points as can be seen in Figures 7,8, and 9, obtaining already owned wood, sawing the wood as saw fit by the group, utilizing tape, ripping a piece of cardboard from a box, and printing out Professor Hood's picture taken. Next section discusses which sensor sets off which facial action.

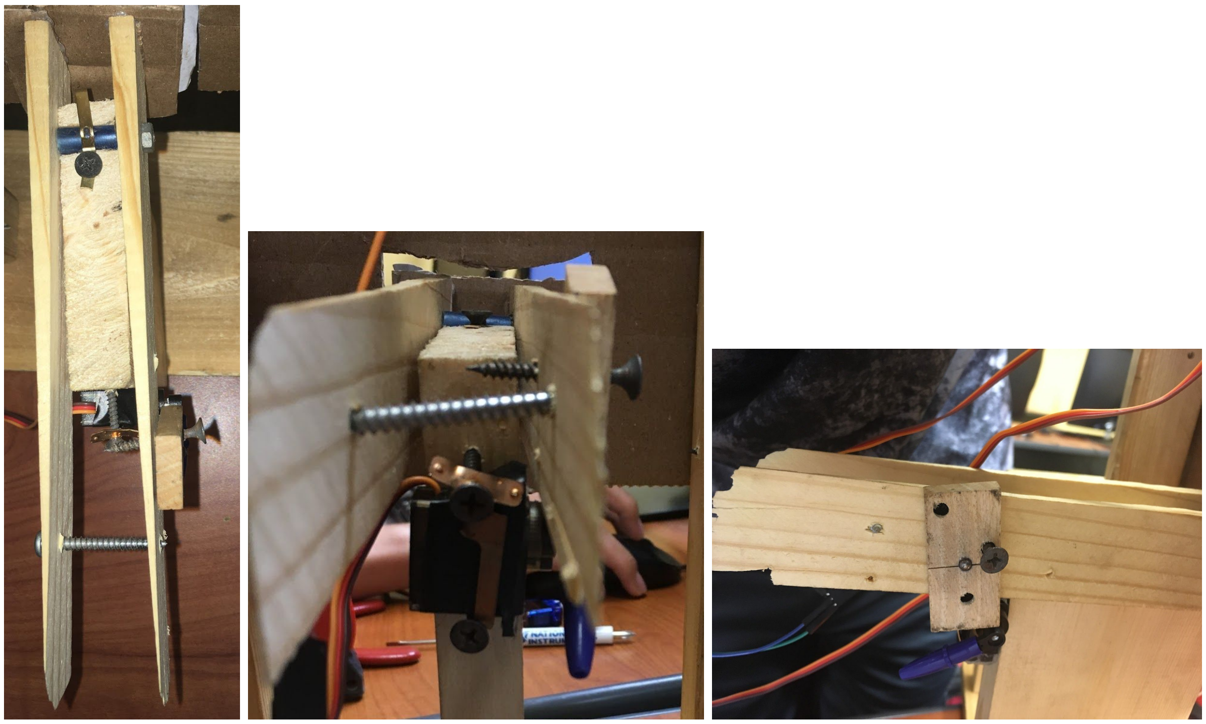


Figure 7: Detailed views on how the servo motor will create the jaw movement

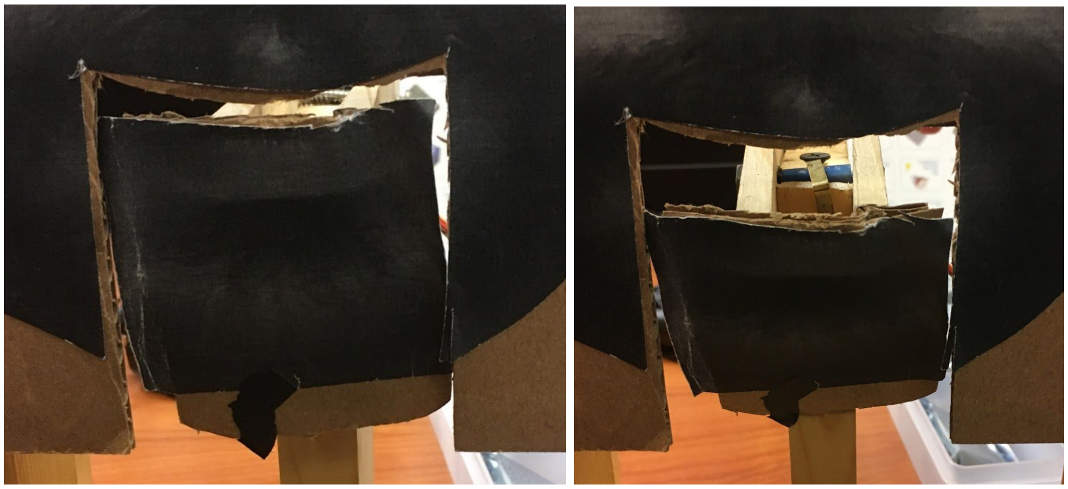


Figure 8: Left image is without the sensor triggering the servo motor, and the right image is when the servo motor has been triggered by the sensor

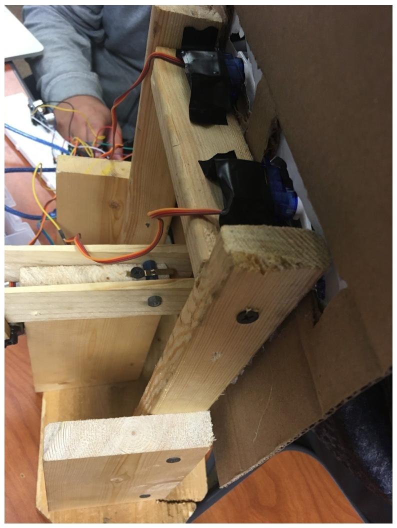
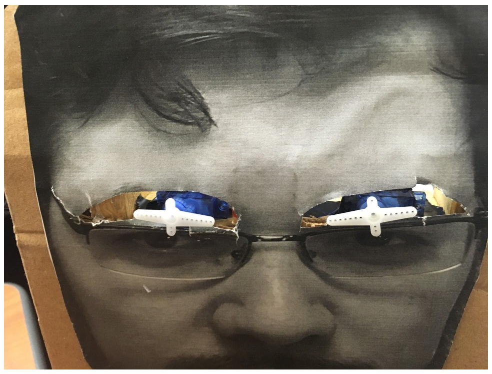



Figure 9: Left image is the servo motors mounted behind the face with also showing where the screws are located, and the right image is a visible illustration on the animatronic faces "eyebrows"

Section 5: Sensors

The jaw, eyebrows, and audio are sparked from the following triggers: ultrasonic sensor (HC-SR04) and a Push Button. These both can be seen in figure 10, and a closeup of them both on the Arduino board can be seen in figure 11. The ultrasonic sensor was utilized since it was already pre-owned. The way this sensor work is that it emits an ultrasound at 40,000 Hz which travels through the air and if there is an object or obstacle on its path, then it will bounce back on the module. The HC-SR04 has 4 pins: ground, VCC, Trig, and Echo. The ground and the VCC pins of the module needs to be connected to the ground and the 5V pins on the Arduino board respectively. The trig and echo pins are to be connected to any Digital I/O pin on the Arduino Board.

It provided no problems but the second sensor is what created a dilemma for the group. The button piece was not initially in the design. Before this piece, a motion and touch sensor were experimented with. Both were approved by Professor Bhounsule to use for the project and can be seen in figure 12. At first, a HC-SR501 PIR Motion Sensor was hypothesized to be an easy use sensor. Upon testing, it was concluded that this sensor took too long to calibrate and would recognize too much of its surroundings. Next up was a KY-036 Metal Touch Sensor. It was confirmed after testing it that there was an inconsistency in how the metal touch sensor was triggered. Not every contact with the sensor triggered the motor, so the group searched for an alternative that used the same principle as the metal touch sensor to remedy this discrepancy. As a result, a button that comes with the already owned Mega 2560 Project Starter Kit was chosen for the design. Reason? To save money and time taken for ordering another sensor. The button connects two points in a circuit when you press them. Once the button is pressed, the jaw and the eyebrows will move. The ultrasonic will prompt the servo motor for the jaw to move once it senses an object within a 27 cm radius, and this will also spark the audio to sound off.

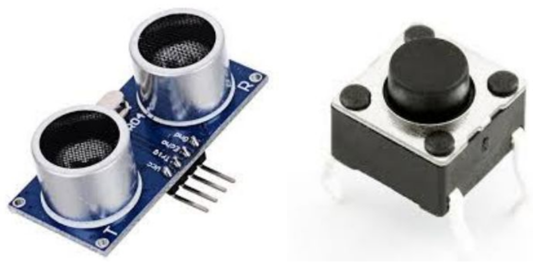


Figure 10: Ultrasonic sensor on left, and button on the right

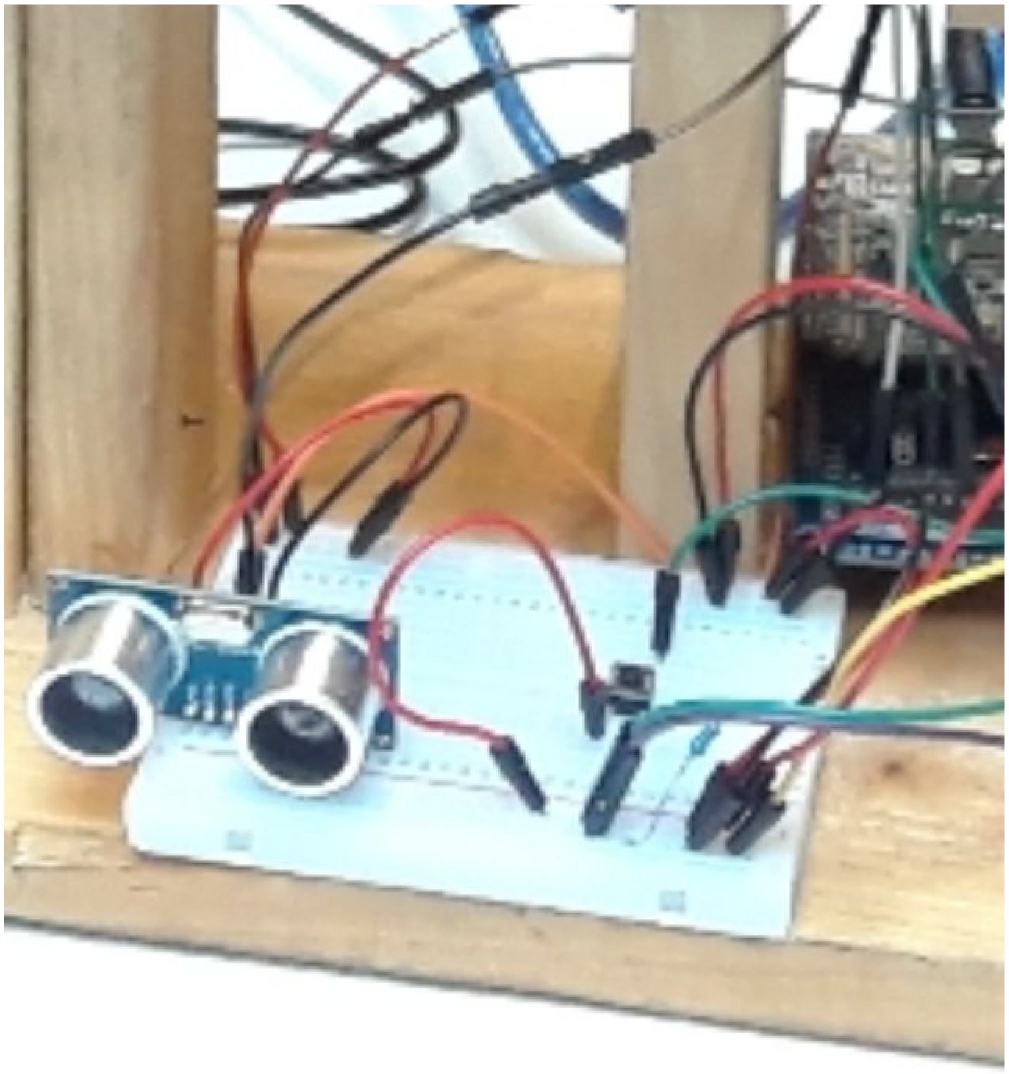


Figure 11: Ultrasonic sensor and button close up on Arduino board

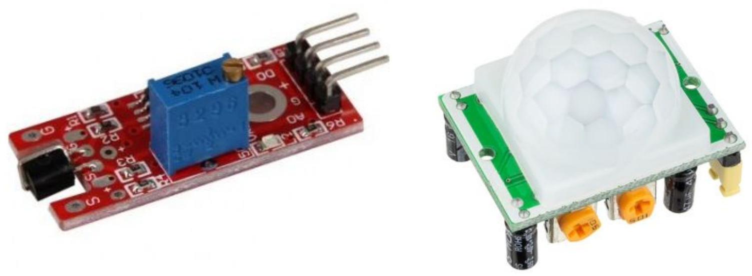


Figure 12: Touch Sensor on left, and motion sensor on the right

Section 6: Programming for interaction

The first initial steps of initiating the interactions for the animatronics face was a learning curve of how the servo motors work alone which went from 0 to 180 degrees. the servo motors code was simple which was learned basically by looking at examples and tutorials online. Next step was using one of the sensors that were going to be used for the interaction of the project and coding in it with the servo motors which was a bit challenging at first working with the “if” and “else” statements to make them begin movement when the ultrasonic detected an object from 20 cm it used the if statement for interacting it to move the servo about 90 degrees which opened the mouth when detected. when it was greater than 20 cm then the “else” statement came in to say to remain at 0 degrees. Furthermore, the following step was interacting push-button inside the ultrasonic code with the servo motors. The button would be moving the eyebrows at 45 degrees when pushed and the mouth at 90 degrees. The button would be coded with the MP3 shield which would talk when the button is pushed and would say “I need my personal space”. The interaction with the motors and sensors was successful when we presented the animatronic project which can be seen in the link below and the code can be seen in Appendix A.

A key part of the MP3 Player’s integration with the project was including a custom library from Adafruit, the MP3 shield manufacturer, into our installation of Arduino IDE for additional commands to be recognized and programmed into the Arduino Mega. Adafruit provided their own tutorial on how to utilize the custom library and commands associated with them.

Link to the project operating is in the following link:
https://youtu.be/qT1WzgDSaUM

Section 7: Lessons learned and suggestions (1 page)

Lessons Learned:

1. Research pales in comparison to the experience
2. Hardware has its own demands

A key problem that delayed the design process was the lack of the group's knowledge of sensors. As already mentioned in the sensor's section; a total of 2 sensors were experimented with before finalizing our second and final trigger (the button). Research can only do so much, and it proved to not be enough when operating the first two sensors. It all came down to experience, which the group grew in thanks to this project. It was troubleshooting commands and inputs which taught the most, not reading a list of commands from Arduino tutorials. Future projects within the same area will now be completed with more experience, the team having a stronger grasp of the language and approach.

We also encountered one of the problems with the project was having the MP3 shield play a track while simultaneously using the ultrasound sensor as a trigger for the servo motors. After much analysis and troubleshooting, the cause was determined to be something none of us had foreseen. The MP3 shield demanded so much of the Arduino's resources that it simply crashed when the demand, on top of the Ultrasound sensor and servo motors, grew too great for the Arduino Mega to reasonably handle. The only way to resolve the problem, at the time, was to simply remove the recording from the first set of movements altogether. A sign of this issue can be seen in the provided video in the prior section, where the mouth servo becomes far less responsive when operating while also playing a recording, triggered upon a button press. However, the lesson learned here was to consider the demands and available processing power of your hardware in designing your project.

Suggested Improvements:

1. A less resource intensive MP3 Shield or Sensor
2. Connected speakers
3. Independent Power Source

For an improvement to the design and as a better solution to the aforestated problem, it would be ideal to simply find either a less resource intensive MP3 shield or sensor so that playing a recorded speech track would not kill the operation.

Another way to improve would have been to use a different version of the MP3 Shield which came with Stereo Amps which could be connected to 3 W 8 Ohm speakers, removing the need for having to carry a speaker independent of the construct. This would have made the final project completely independent save for requiring a power source.

The third potential improvement would be to make the entire project completely independent by providing a battery to power the Arduino Mega and using an adapter for it to act as the external power supply. This would make it unnecessary to connect it to anything else and have it be a completely standalone assembly, requiring only a battery and connection to start operation.

Section 8: Personnel and bill of materials

(a) Personnel

Task	Main Personnel	Secondary personnel
Jesus Madrigal	Creating joints and motor interfacing/programming	we all contribute together when needed
Nicholas Lopez	structure/chassis design including the face	we all contribute together when needed
Joseph Turrubiates	programming and integration	we all contribute together when needed

(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			
4	Nails	Pre-owned			
5	Cardboard	Pre-owned			
6	Picture of Professor's Face	Printed From UTSA			
7	KY-036 Metal Touch Sensor	https://www.newegg.com/Info/TrackOrder.aspx?Trackingnumber=LN678490918CN	1	5.59	5.59
8	Adafruit Music Maker MP3 Shield	https://www.amazon.com/gp/product/B00SDTE380/ref=oh_aui_detailpage_o01_s00?ie=UTF8&psc=1	1	40	40
	Total				52.54

Acknowledgments
The group wants to take this time to acknowledge Professor Lyle Hood for taking time out of his schedule to meet for the recording and face picture.

References:

[1] Cynthia, n.d., “Kismet,” from
http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html

[2] Cynthia, n.d., “Kisme, the robot,” from
http://www.ai.mit.edu/projects/sociable/baby-bits.html

[3] gtooms, n.d., “RomoBOT-Animatronics Face Robot,” from
https://www.instructables.com/id/RomoBOT-Animatronic-Face-Robot/

[4] PyroElectro, August 25th, 2011, “Animatronic Face Completed-Testing Movements,” from
https://www.youtube.com/watch?v=YmK1c3W1axw&feature=youtu.be

[5] PyroElectro, n.d., “Animatronics How to DYI,” from
http://www.pyroelectro.com/animatronics-how-to/

Appendix A: Code

```
// include SPI, MP3 and SD libraries
#include <SPI.h>
#include <Adafruit_VS1053.h>
#include <SD.h>
// define the pins used
//#define CLK 13       // SPI Clock, shared with SD card
//#define MISO 12      // Input data, from VS1053/SD card
//#define MOSI 11      // Output data, to VS1053/SD card
// Connect CLK, MISO and MOSI to hardware SPI pins.
// See http://arduino.cc/en/Reference/SPI "Connections"

#define BREAKOUT_RESET  9      // VS1053 reset pin (output)
#define BREAKOUT_CS     10     // VS1053 chip select pin (output)
#define BREAKOUT_DCS    8      // VS1053 Data/command select pin (output)
// These are the pins used for the music maker shield
#define SHIELD_RESET  -1      // VS1053 reset pin (unused!)
#define SHIELD_CS     7      // VS1053 chip select pin (output)
#define SHIELD_DCS    6      // VS1053 Data/command select pin (output)

// These are common pins between breakout and shield
#define CARDCS 4     // Card chip select pin
// DREQ should be an Int pin, see http://arduino.cc/en/Reference/attachInterrupt
#define DREQ 3       // VS1053 Data request, ideally an Interrupt pin

  Adafruit_VS1053_FilePlayer musicPlayer =
    // create breakout-example object!
    //Adafruit_VS1053_FilePlayer(BREAKOUT_RESET, BREAKOUT_CS,
BREAKOUT_DCS, DREQ, CARDCS);
    // create shield-example object!
    Adafruit_VS1053_FilePlayer(SHIELD_RESET, SHIELD_CS, SHIELD_DCS, DREQ,
CARDCS);

#include <Servo.h>
#define trigpin 47//set trigpin
#define echopin 43//set echopin
Servo myservo;// declare servo name type servo
 int duration, distance;//declare variable for ultrasonic sensor
int button1 = 53; //button pin, connect to ground to move servo
int press1 = 0;
Servo servo1;  //left eyebrow
Servo servo2; //right eyebrow
void setup() {
  Serial.begin(9600);
  // MP3 Shield Code
   Serial.println("Adafruit VS1053 Simple Test");

  if (! musicPlayer.begin()) { // initialise the music player
     Serial.println(F("Couldn't find VS1053, do you have the right pins defined?"));
     while (1);
  }
  Serial.println(F("VS1053 found"));

   if (!SD.begin(CARDCS)) {
    Serial.println(F("SD failed, or not present"));
    while (1);  // don't do anything more
  }

  // Set volume for left, right channels. lower numbers == louder volume!
  musicPlayer.setVolume(20,20);

  // Timer interrupts are not suggested, better to use DREQ interrupt!
  //musicPlayer.useInterrupt(VS1053_FILEPLAYER_TIMER0_INT); // timer int

  // If DREQ is on an interrupt pin (on uno, #2 or #3) we can do background
  // audio playing
  musicPlayer.useInterrupt(VS1053_FILEPLAYER_PIN_INT);  // DREQ int
  pinMode(trigpin, OUTPUT);

  pinMode(echopin, INPUT);
  myservo.attach(22);// attach your servo
myservo.writeMicroseconds(1500);
  // put your setup code here, to run once:

  pinMode(button1, INPUT);
  servo1.attach(24);
  servo2.attach(23);
  digitalWrite(4, HIGH); //enable pullups to make pin high
}
void loop() {
  //MP3 Shield Code
if (Serial.available()) {
    char c = Serial.read();

    // if we get an 's' on the serial console, stop!
    if (c == 's') {
      musicPlayer.stopPlaying();
    }

    // if we get an 'p' on the serial console, pause/unpause!
    if (c == 'p') {
      if (! musicPlayer.paused()) {
        Serial.println("Paused");
        musicPlayer.pausePlaying(true);
      } else {
        Serial.println("Resumed");
        musicPlayer.pausePlaying(false);
      }
    }
  }
  //End MP3 Shield Code
 myservo.write(90);// always set servo to 90 to position it to the middle
//ultrasonic code
 digitalWrite(trigpin,HIGH);
  _delay_ms(500);
  digitalWrite(trigpin, LOW);

  duration=pulseIn(echopin,HIGH);
  distance=(duration/2)/29.1;
 if(distance <=20)// if ultrasonic sensor detects an obstacle less than 20cm in 90 degree angle.
  {
  myservo.write(0); //servo rotates at full speed to the right
 delay(200);
 myservo.write(90);//close
 delay(150);
 myservo.write(0);
 delay(150);
 myservo.write(90);
 delay(150);
 myservo.write(0);
 delay(150);
 myservo.write(90);
 delay(150);
 myservo.write(0);
 delay(150);
 myservo.write(90);
 delay(150);
 myservo.write(0);
 delay(150);
 myservo.write(90);
 delay(500);

   delay(150);
 return;
}
else
{
  myservo.write(90);// else servo stays at 90 degree angle.
  delay(400);
}
  Serial.print("cm"); //print distance unit cm
Serial.println(distance);//distance
  // put your main code here, to run repeatedly:
press1 = digitalRead(button1);
  if  (press1 == HIGH)
  {
    Serial.println(F("playing track 004"));
    musicPlayer.startPlayingFile("/track004.mp3");
    delay(300);
    servo1.write(0);
    servo2.write(180);
    myservo.write(90);
    delay(200);
    servo1.write(90);
    myservo.write(0);
    servo2.write(90);
    delay(150);
    servo1.write(0);
    myservo.write(90);
    servo2.write(180);
    delay(150);
    servo1.write(90);
    myservo.write(0);
    servo2.write(90);
    delay(150);
    servo1.write(0);
    myservo.write(90);
    servo2.write(180);
    delay(200);
    servo1.write(90);
    myservo.write(0);
    servo2.write(90);
    delay(100);
    servo1.write(0);
    myservo.write(90);
    servo2.write(180);
    delay(200);
    servo1.write(90);
    myservo.write(0);
    servo2.write(90);
    delay(100);
    servo1.write(0);
    myservo.write(90);
    servo2.write(180);
    delay(150);
  }
  else {
    servo1.write(90);
  }
  myservo.write(90); //close
  servo1.write(90);
  servo2.write(90);
  }
```