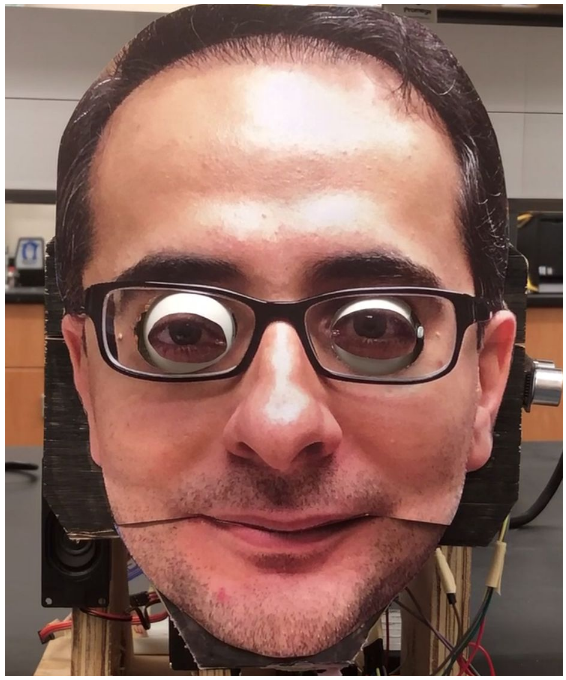
ANIMATRONICS FACE: PROF ALAEDDINI



Video Link:
https://www.youtube.com/watch?v=anDZXpLLTjs

Juan C Garduno-Martinez
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
j.c.garduno96@gmail.com

Gregory C. Williams
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
qhy288@my.utsa.edu

Joel L. Gomez
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
joelgomez190@gmail.com

Submitted as part of project for ME4543 Mechatronics, Fall 2018

TABLE OF CONTENTS

Abstract	3
Literature Review	4-7
Brainstorming	7-9
Supporting Structure	9
Joints and Motors	9-11
Sensors	11-12
Programming	12-13
Lesson learnt and suggestions	13-14
Personnel and Bill of Materials	14-16
Acknowledgments	16
References	16
Appendix A: Code	17-24
Appendix B: Arduino wiring	25
Appendix C: Figures	26-27

ABSTRACT

The purpose of this project is to build an animatronic face having two different types of sensors with three different human like movements using the skills and knowledge learned from UTSA's Mechatronics Laboratory. The ability to incorporate the right materials, being sensors, mechanical joints, and actuators with code to make a working robot will show in detail the skills gained during the lab. This project will utilize servo motors, Arduino Mega, a mounting surface and much more hardware to actively demonstrate human facial characteristics of the animatronic face of a UTSA Mechanical Engineering Department faculty member. After much work the final product was a recreation of Dr. Alaeddini who waits for a person to interact with him, then goes ahead and asks the user a question related to data mining. Depending on if the user responds correctly or wrongly Animatronic Alaeddini will respond differently.

Section 1: Literature review

Talking Animatronic Robot Head

The talking robotic head was an animatronic project that an, unnamed, gentleman had plenty of free hobby time to perform. The purpose of his project was his interest as a hobby and also he shared it with schools in hopes to develop young minds in the interest of robotic engineering. He is able to take his robotic head to elementary, junior high and high schools and have the head tell jokes as well as make funny comments mostly about technology and how it "personally" affects him. With all of the developing minds sitting on the edge of their seats, this talking robot sparks the technological interest of many students.

Figure 1: Talking animatronic robot head

Using a lot of equipment, he was able to complete this hobby project in less than 6 months. The supplies are as follows but not limited to: plywood, wooden knobs, doll eyes, screws, nuts, bolts, 1/8th inch brass rods, sheets of metal, latex, various servos, wires, servo controller, powered computer speakers, electrical connectors, large trunk, wooden box, a clip, power strip, cooling fan, extension cords, sunglasses, single-board computer, compact flash card, micro switch, plastic box, hot glue, heat shrink tubing and varnish. The tools are as follows but not limited to: drills, drill bits, band saw, scroll saw, wire stripers, soldering gun, heat gun, hot glue gun, hammer, screw driver, and pliers. Using all of these tools and equipment the gentleman was able to put together the animatronic face and encourage kids all over his town to follow their dreams of becoming a robotic engineer.

For more references and detailed information please visit reference [1].

Kosmo’s 2.0 Face (Synthesizer with a face)

Kosmo 2.0 is a sound synthesizer, with a face, that reacts to the music being played. The creator, Sam Battle, is a music artist that goes by the alias “Look mum no computer”. He decided to put a face on his music synthesizer because he felt that the synthesizer is his “buddy” and he deserves a face. The face has the capability of moving two eyes independently up and down, a moving mouth, and eyelids. The face has two ways to control it. The first way is by using knobs, each will move an individual servo, which will then move a part of the face. The second way to control the face is by the input signals that create a melody, when a melody is played, the eyes and mouth move. If a low note is played the eyes move one way and if a high note is played the eyes move to the opposite way. Furthermore, the eyebrows react to the kick drum. On the construction of the face, a piece of three-ply was used for the frame of the face, where all the components would be mounted. Ping pong balls were used for the eyes and metal scrap was used to construct the frame for the eyes. Then a screw was used as a shaft for the eyes to move up or down, to achieve the movement, the eye was connected to a servo motor using a zip tie. To move the eyes left or right the frame was simply mounted to another servo motor. The mouth and eyebrows were constructed using the same principle, connecting the components to a servo motor and mounting them to the wood frame. Unfortunately the code for the movement of the face was not included. The face was controlled through a Teensy 3.2 because the creator needed more PWM pins than the Arduino could provide. Please refer to reference [2] and [3] for more information.

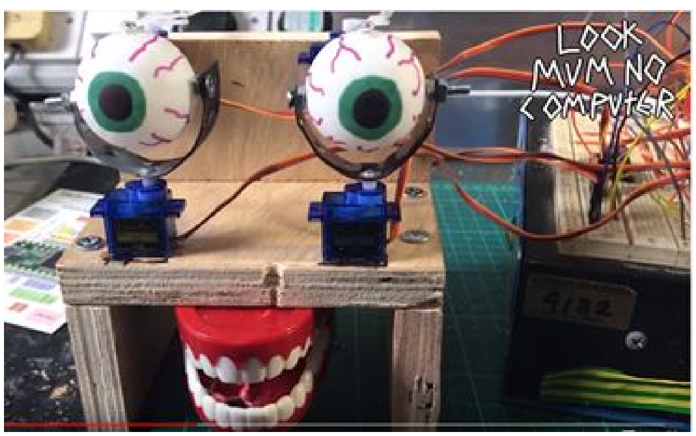


Figure 2: Kosmo 2.0’s frame

Figure 3: Kosmo 2.0

Peter Penguin

Penguin animatronic was created to be an animated holiday display to help raise money for a local arts program. The penguin was able to dance and make facial expressions along with the movements. The facial expressions were created through eyes which move side to side and eyelids along with a moving mouth or beak in this case.
The eyes were constructed using model plane plywood for the frame, 2 mini RC servos, 1 standard RC servo, styrene, epoxy, glue, paper clips, and small screws. The eyelids are made from the round end of Easter plastic eggs as shown below in Figure 4.

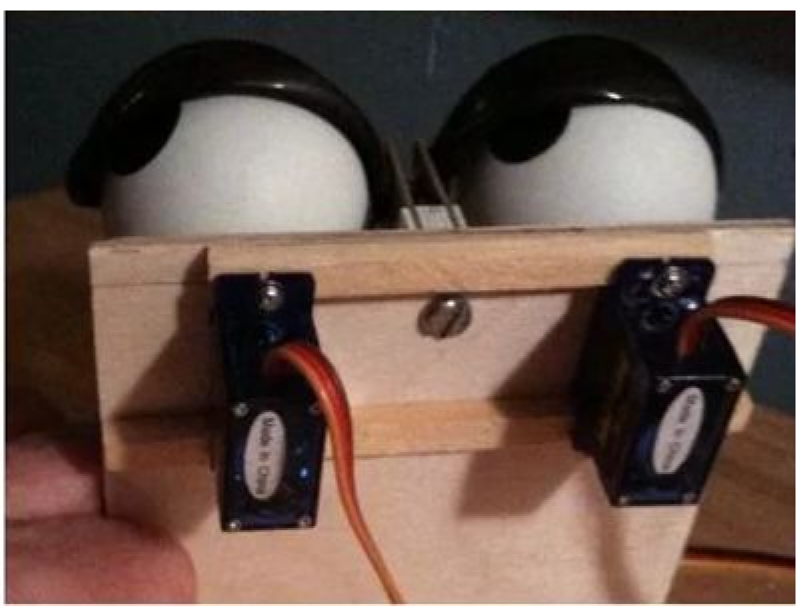


Figure 4: Servos mounted to plywood base and connected to eyes

As shown in the figure above the step motors are mounted below the eyes to control the side to side movement of the eyes.

Figure 5: Final assembly of animatronic penguin face

In the Figure 5 above the black servo atop the plywood base is used to control the movement of the eyebrows.

In terms of hardware for controlling the servo motors, the creator used a SSC-32 (serial servo controller - 32 channel) from Lynxmotion, which comes with a USB to serial converter. Additionally, Lynxmotion provides a terminal program called LynxTerm that may be used when aligning the servos to set up the initial positions for the eyes and eyelids.

The code is based on the Lynxmotion SSC-32 controller. The creator used a DOS .CMD file to send commands to the servo controller. They used an Excel spreadsheet to map action descriptions to the SSC-32 commands, then, in another sheet, used these descriptions to "script" and look up the appropriate "codes" to construct the DOS command. The results were cut and pasted into a .CMD file which was then run to produce the resulting motion. The values for the servos were determined via experimentation with the LynxTerm program.

For more information please visit references [4] and [5].

Section 2: Brainstorming (initial planning)

Structure:

For the animatronic face structure wood will be the best option due to durability, weight, and the ability to simply modify. This will be a trial and error build, so the ability to modify the structure is very important. Assume all wood cutouts are from ¾" plywood. We will first start with a base using an 18" x 18" cutout. This will give us the base structure to attach our electronics to. Starting from the bottom of the face we need to account for the chin movement which will move in a vertical fashion, perpendicular to the face. Therefore, when we attach our servo motor for the chin it will need to be installed where the arm of the servo can rotate vertically, so we will vertically install a 3” high x 2” wide cutout directly in the middle of the base perpendicular to the face. Next we need to consider the eyes. The eyes will be located 4” apart from each other, centered over the chin and 7” from the base. Since they will have their own servo motors running perpendicular to the face we will need to install a T shaped piece of wood that is 6 ¼” tall (3/4” accounting for the servo height). Here is a possible modification where we could attach the chin servo to the bottom of the T structure. Lastly the eyebrows are our next challenge. The eyebrows will need to rotate from the inside of the eyebrow giving a “brow raising” effect. For this we will need to attach two servos that rotate parallel with the face and need to be located ½” from the top of the eyes. We can erect an 8” bridge from the base that is 10” wide, providing enough room for eyeball clearance. Center the bridge parallel to the face and attach the two servo motors at the top face of the bridge to rotate parallel to the face approximately 1” the right center and 1” on the left center of the base. For the face structure itself we will need to erect a face profiled piece of wood with cutouts for the chin and eyes and also 1/8” holes for the eyebrows. I would suggest making the chin from Styrofoam board to lessen the strain taken on by the chin servo motor.

Electronics:

Arduino MEGA powered by 9V battery will be used to control all motors and sensors in this project. Circuit will be powered through use of 6 AA batteries. A total of 3 servo motors will be utilized to provide the motion, for the eyes vertically and horizontally, and mouth. To transmit the motion from the servos to the animatronic body parts, particularly the eyes, zip ties and popsicle sticks will be cut and used as connectors to transmit the motion to the eyes. Eyes will be made from ping pong balls, and the mouth will come from a plastic toy.

Generic components such as wires and resistors, will come from Arduino kits which the group already has. In order to have someone interact with the animatronic an ultrasonic sensor will be used to control one of the motion modes, while a button will be used to control the other. For the audio output an DFPlayer Mini Mp3 player will be purchased as an add-on for the Arduino along with a small speaker. Breadboard will be utilized for all wire connections, but if wire length is a limitation then longer wires are available for us to use, left over from previous projects. Multiple breadboards may be needed to accommodate for buttons to be located in easy to access location for persons to interact.

The face will be interacted with by use of an ultrasonic HC-SR04 sensor, and the buttons. The ultrasonic sensor will detect when someone is close to the face. At this point the face will move and ask a question. After that the face will wait for some time until the user presses one of two button, representing a ‘wrong’ and ‘right’ answer. The face will then react differently based on the response to the question. This is inspired by the iClicker style questions posed by professor in lecture

When meeting with Dr. Alaeddini, he will be recorded introducing himself, asking a question related to his field, data mining, provide two choices for the answer to the question, and a response for the right and wrong answer each. These recordings will then be converted to mp3 and saved to small microSD card.

Section 3: Supporting structure

For the animatronic face structure wood was the best option due to durability, weight, and the ability to simply modify. This was a trial and error build, so the ability to modify the structure was very important. All wood was cut from a ¾"x4'x4' piece of plywood purchased from Home Depot. We first started with a base using an 18" x 18" cutout. This gave us the base structure to attach our electronics to. Starting from the bottom of the face we need to account for the chin movement which moves in a vertical fashion, perpendicular to the face. Therefore, when we attached our servo motor for the chin it needed to be installed where the arm of the servo can rotate vertically, so we vertically install a 3" high x 2" wide cutout directly in the middle of the base perpendicular to the face(Figure 14). Next we considered the eyes. The eyes were located 4" apart from each other, centered over the chin and 7" from the base. Since they shared a servo motor running perpendicular to the face we installed a T shaped piece of wood that is 6 ¼" tall (3/4" accounting for the servo height). Here we modified where we could attach the chin servo to the bottom of the T structure to get an accurate vertical movement of the chin(Figure 13). Lastly the vertical eye movement was another challenge. For this we needed to attach one servo that rotated vertically behind and approximately ⅛" above the eyeballs. We erected an 8" bridge from the base that is 10" wide, providing enough room for the eyeball clearance(Figure 13). Center the bridge parallel to the face and attach the servo motor at the top face of the bridge to rotate vertically and perpendicularly to the face. For the face structure itself we erected a face profiled piece of wood with cutouts for the chin and eyes(Figure 12). Instead of using the existing ¾" plywood chin cutout we replaced it with styrofoam board to minimize the stress put on the chin servo motor.

Section 4: Joints and motors

The animatronics face had three moving parts, the chin, and two eyes. The chin had a direct connection to one servo as shown in Figure 6 below. It was made out of 3 popsicle sticks glued together to have a stiff connection. It was hot glued to the chin and connected to the servo using two small screws. The result was a rigid connection that would give the chin the same angular velocity as the servo had. This was the simplest connection.

Figure 6: Chin Joint

First, the eyes were mounted in a Y frame. This would allow the eyes to on two axis, up and down, left and right. This was achieved using only two servos, one for the up and down movement, and the second servo for the left and right movement. The two Y frames were connected from the bottom using popsicle sticks, it would later be connected to the driver servo to rotate the eyes left and right.

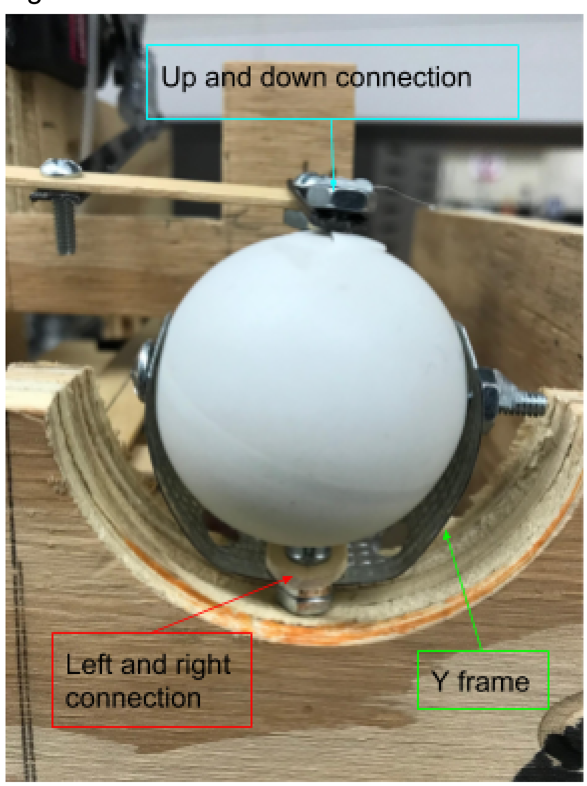


Figure 7: Y frame and Eye Connections

Then, the eyes were connected from to top by a popsicle stick using zip ties to have flexibility on the joints and allow the eyes to rotate on the left and right direction while having a up and down movement at the same time. The popsicle connection would later be connected to the driver servo also through a zip tie.

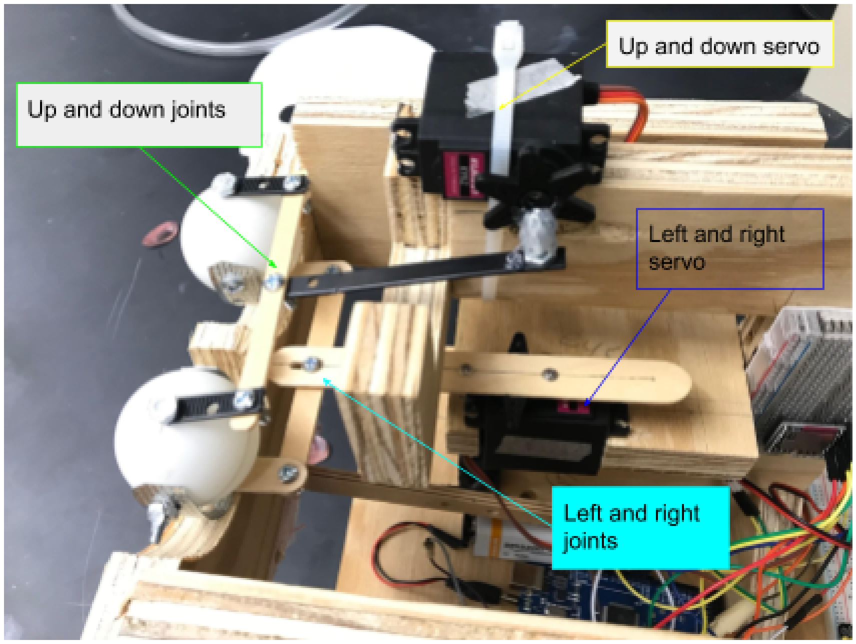


Figure 8: Eye Joints

All links were connected using small screws to allow for a free rotation. The servo and joint configuration reduced the coding and wiring requirements by reducing the number of servos to the minimum and assured that the eyes would move at the same time and velocity. The servos choosed to move the joints were Kuman KY62 because they provided a good torque and were sold at a good price to one of our team members.

Section 5: Sensors

For this project three interactable sensors were used, one Ultrasonic sensor and two push buttons. Ultrasonic HC-SR04 sensor spec sheet can be found in reference [6]. This sensor sends out an ultrasonic pulse which hits an object and bounces back to be read by the sensor, allowing for the sensor to determine the distance between it and an object, much like the echolocation of a bat. The sensor measures the amount of time for the pulse to leave the sensor and then return to the receiver after bouncing off an object. Based on the speed of sound in air the distance to the object can be determined using the measured time by the following equation: distance= duration*0.034/2. The sensor has 4 pins as shown in figure 9 below. Vcc connects to 5V power supply, trigger sends out the output pulse signal, echo receives the returning pulse signal from an object, and the last one connects to ground. This sensor was chosen due to it's minimal setup(no extra circuitry requires like resistors) which allowed it to be plug and play, and also due to one of the members experience using it in a past Measurements and Instrumentation project.

Figure 9: HC-SR04 Ultrasonic Sensor

The second interactable set of components used in the animatronic were two push buttons like the ones in figure 10 below. The button operates as a switch in the circuit. One of the legs is connected to power (5V) and the other is connected by a pull down resistor to ground. One of the top pins can then be connected to the Arduino so that when the button is pushed, completing the circuit and connecting power to ground, the microcontroller reads a HIGH value. Otherwise the microcontroller receives a LOW signal from the button. The main reason for selecting push buttons for the project was their similarity to the buttons on the iClicker used by many professors at UTSA, which was the inspiration for the interaction the animatronic would have by posing a question and then having the user poll their response.

Figure 10: Push Button

Section 6: Programming for interaction

The code, which can be found in Appendix A begins by loading the necessary libraries for the sensors and mp3 module as well as initializing all connections and setting the speaker volume to 30, being the highest setting. The servos are attached to three different pins with names indicating the body part or motion they control.

An if statement is established for the ultrasonic sensor so that when a reading of less than 5 and more than 1 cm is detected, the animatronic will run through a sequence of motions while playing through the first four lines of audio recorded by Dr. Alaeddini. The sequence of motions is done by setting or writing the servo position it should move to along with a delay before the next motion. The determination of delay times is further discussed later in this section of the report.

After the user has set off and listened to the animatronic pose the question, the system waits for the user to respond by pushing the buttons corresponding to answer choice ‘A’ and ‘B’. This is controlled by two if conditions, one for button ‘A’ pushed, leading to a correct answer audio to play along with a set of motions, and the other for button ‘B’ being pushed leading to a wrong answer audio response to play with a different set of motions.

At the end of the code an idle mode ‘else’ condition is set so that if the sensor is not detecting an object within 5 cm or neither button ‘A’ or ‘B’ are pressed then the face will run through a set of motions mimicking a human looking around the room.

The microSD card containing the audio files is placed into the DFPlayer Mini. The files must be named using 4 digit number (ie 0001, 0002, 0010) followed by any name you would like. The specific audio file can then be called using the numeric value. A spec sheet with sample codes and wiring diagram for the DFPlayer Mini mp3 module can be found at reference [7].

Delays for synchronized motions were determined by using Audacity software with mp3 files to figure out the length of time for Professor Alaeddini to speak each word in the sentences and also the time for gaps between words.

More detail for the code can be found by reading the comments in the code found in Appendix B.

Section 7: Lessons learnt and suggestions

Lessons Learnt

1. Servos draw power suddenly. Ultrasonic sensor when connected to same power source as servos would experience power drop and give off false readings of objects at 0cm. To fix issue sensor was connected to power from Arduino MEGA board separate from servo motor power supply.
2. Buttons would not work if one leg was connected directly to ground. 1kΩ pull down resistors needed to be connected from button to ground for buttons to function.
3. Servos lose track of absolute position after the code is re uploaded or the board is reseted
4. Speakers need a different power supply than the motors to work properly.
5. Debugging and getting the wiring right is painful and frustrating.

Improvements

1. Servos lose track of initial position over time, and when turned on and off. A line of code should be included so that the servos do not lose track of position.

2. Servos twitch between changes in position, the wiring should be checked again to solve issue .
3. There is a significant delay between user pressing answer choice button and animatronic faces reaction to answer choice, the face should react as soon as the response is submitted.
4. A command should be added to allow user to submit a second response if they have answered wrong the first time.
5. As of now, the ultrasonic sensor checks for proximity only after a set of movements by the eyes is completed. The sensor should be allowed to initialize the conversation as soon as it reads a distance less than 5 cm.
6. Servo motors were quite loud and at times made hearing the audio difficult.
7. Higher volume speakers should be used when displaying in crowded environment.

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

Task	Main Personnel	Secondary personnel
Structural Frame	Greg	Juan
Joints and Motor Interface	Greg	Juan
Integrating Sensor	Joel	Juan
Wiring	Joel	Juan
Servo Coding	Juan	Greg
Audio Coding	Juan	Joel
Voice-Motor Sync	Juan	Greg

(b) Bill of materials

No.	Description	Website/comment	Qty.	Unit $	Total $
1	Arduino MEGA 2560	Provided	1	-	-
2	Servo Motors	amazon.com	4	4.30	17.20
3	DFPlayer Mini	https://www.amazon.com/gp/product/B06WP9RQ9V/ref=oh_aui_detailpage_o	1	4.69	4.69

		03_s00?ie=UTF8&psc=1			
4	3W 8Ω Speaker	https://www.amazon.com/CQRobot-JST-PH2-0-interface-Electronic-Projects/dp/B0738NLFTG/ref=mp_s_a_1_1?ie=UTF8&qid=1543721339&sr=8-1&pi=AC_SX236_SY340_QL65&keywords=cqrobot&dpPl=1&dpID=51tCpWBH2ZL&ref=plSrch	1	7.87	7.87
5	M-M, M-F, F-F Jumper Wires	https://www.amazon.com/gp/product/B077X7MKHN/ref=od_aui_detailpages00?ie=UTF8&psc=1	120	0.05	5.99
6	64 MB MicroSD Card	From home	1	-	-
7	1kΩ Resistor	From home	3	-	-
8	9V Battery	From home	1	-	-
9	AA Batteries	From home	6	-	-
10	Breadboard Power Supply Module	From home	1	-	-
11	Full Breadboard	From home	1	-	-
12	Half Breadboard	From home	1	-	-
13	Mini Breadboard	From home	1	-	-
14	Pushbuttons	From home	2	-	-
15	HC-SR04 Ultrasonic Sensor	From home	1	-	-
16	¾"x4'x4' Plywood	Home Depot	1	17.26	17.26
17	Wood Screws(box)	Home Depot	1	4.28	4.28
18	Drywall Corner Bead (eye mounts)	Home Depot	1	2.72	2.72

19	Bolts/Nuts (combination pack)	Home Depot	2	3.26	6.52
20	Ping Pong Balls	Student Rec Center	2	$0.25	$0.50

The total price for this project excluding the Arduino MEGA and parts from home was **$67.03**

Acknowledgements

Dr. Adel Alaeddini

For providing his voice and likeness to the animatronic.

Dr. Pranav Bhounsule

For presenting such a unique project and offering his mentorship to all students who come into his path.

Jose Moreno

For recommending the speaker, and providing help with coding for DFPlayer Mini.

Tony Amaro

For recommending DFPlayer Mini Mp3 module to play audio.

Salvador Echeveste

For providing a hot glue gun that one day, and also helping the lab run smoothly as we learned more about electronic components leading into the project.

References:

[1]Instructables. “Talking Animatronic Robot Head.” *Instructables.com*, Instructables, 3 Nov. 2017, www.instructables.com/id/Talking-Animatronic-Robot-Head/.

[2] www.youtube.com/watch?v=mDe6q6RMUtY.

[3] https://www.youtube.com/watch?v=U5qHMgZJ2w4

[4] Instructables. “Animatronic Eyes.” *Instructables.com*, Instructables, 3 Nov. 2017, www.instructables.co m/id/Eyes/?utm_source=rb-community&utm_medium= forum&ut%2B%2Bm%2B_campaign=peter-penguin.

[5]“DIY Animatronic Penguin Shakes and Grooves.” *Hackaday*, 25 Feb. 2011, hackaday.com/2011/02/25/diy-animatronic-penguin-shakes-and-grooves/.

[6] https://www.mouser.com/ds/2/813/HCSR04-1022824.pdf

[7] https://www.dfrobot.com/wiki/index.php/DFPlayer_Mini_SKU:DFR0299

[8] https://www.youtube.com/watch?v=anDZXpLLTjs

Appendix A: Code

```
// defines pins numbers for ultrasonic sensor
const int trigPin = 2;
const int echoPin = 4;

// defines variables for ultrasonic sensor
long duration;
int distance;

//library
#include <DFRobotDFPlayerMini.h>
#include <SoftwareSerial.h>
#include <Servo.h>

//servos
Servo MOUTH;
Servo eyeside;
Servo eyeup;

//buttons
int button1 = 12;
int button2 = 13;
int answera;
int answerb;

int buttonState=0;

//audio
SoftwareSerial mySoftwareSerial(10, 11); // RX,TX connections
DFRobotDFPlayerMini myDFPlayer;
void printDetail(uint8_t type, int value);

void setup() {
//ultrasonic
pinMode(trigPin, OUTPUT); // Sets the trigPin as an Output
pinMode(echoPin, INPUT); // Sets the echoPin as an Input
Serial.begin(9600); // Starts the serial communication

//servos
MOUTH.attach(3);
eyeside.attach(9);
eyeup.attach(6);
```

```
//audio
mySoftwareSerial.begin(9600);
Serial.begin(9600);

if (!myDFPlayer.begin(mySoftwareSerial)) { //Use softwareSerial to communicate with mp3.
while(true){
delay(1000); // Code to compatible with ESP8266 watch dog.
}
}
myDFPlayer.volume(30); //Set volume value. From 0 to 30

}

void loop() {
//////////////////////echo pin
// Clears the trigPin
digitalWrite(trigPin, LOW);
delayMicroseconds(2);

// Sets the trigPin on HIGH state for 10 micro seconds
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;

// Prints the distance on the Serial Monitor
Serial.print("Distance: ");
Serial.println(distance);
//////////////////////

//Initialize servo positions
eyeup.write(90);
eyeside.write(90);
MOUTH.write(65);
//////////////////

/////////
//Alaeddini poses question when ultrasonic detects something less than 5cm
```

```
if(distance <=5 && distance >=1 )
{
eyeside.write(90);
myDFPlayer.play(1); //Play the first mp3

//HI
MOUTH.write(90);
delay(500);
MOUTH.write(65);
delay(500);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(100);
MOUTH.write(65);
delay(1000);

//Can I ask you a question?
myDFPlayer.play(2);
delay(1514);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(65);
delay(2500);
```

```
//Question
myDFPlayer.play(3);
delay(1229);
MOUTH.write(90);
delay(200);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(65);
delay(2500);

//ANSWER A
myDFPlayer.play(4);
//voice movement
delay(1316);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(300);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
```

```
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(100);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(65);
delay(1000);

//ANSWER B
myDFPlayer.play(5);
//voice movement
delay(665);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(300);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
```

```
delay(150);
MOUTH.write(65);
delay(10000);

//////// response
answera = digitalRead (button1);
answerb = digitalRead (button2);

//// response a: Right Answer
if(answera == LOW && answerb == HIGH)
{ myDFPlayer.play(6);
delay(200);
MOUTH.write(100);
delay(200);
MOUTH.write(75);
delay(150);
MOUTH.write(100);
delay(150);
MOUTH.write(75);
delay(300);
MOUTH.write(100);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(100);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(100);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(100);
delay(150);
MOUTH.write(75);
delay(800);
}

//// response b: Wrong answer
if(answera == HIGH && answerb == LOW)
{ eyeside.write(85);
delay(400);
eyeside.write(95);
```

```
delay(400);
eyeside.write(85);
delay(400);
eyeside.write(95);
delay(400);
eyeside.write(90);
myDFPlayer.play(7);
delay(100);
MOUTH.write(90);
delay(200);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(300);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(212);
MOUTH.write(90);
delay(150);
MOUTH.write(75);
delay(150);
delay(1000);
}
}

//If nothing less than 5cm from ultrasonic, remain idle and look around
else{
eyeside.write(85);
delay(1000);
eyeup.write(105);
delay(1000);
eyeside.write(100);
```

```
delay(1000);
eyeup.write(75);
delay(1000);

}
}
```

Appendix B: Arduino wiring

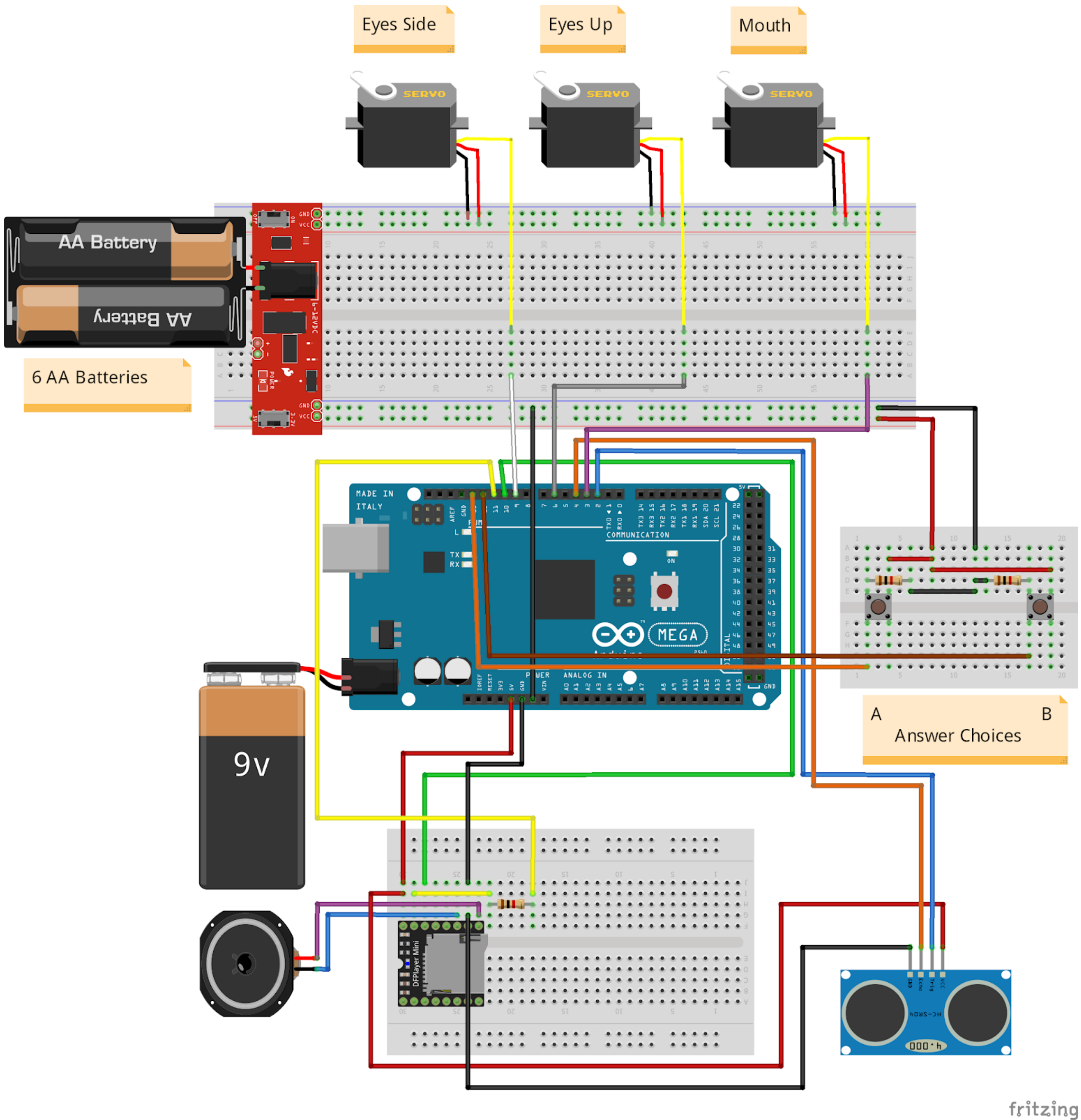


Figure 11: Arduino Wiring

Appendix C: Figures

Figure 12: Front construction

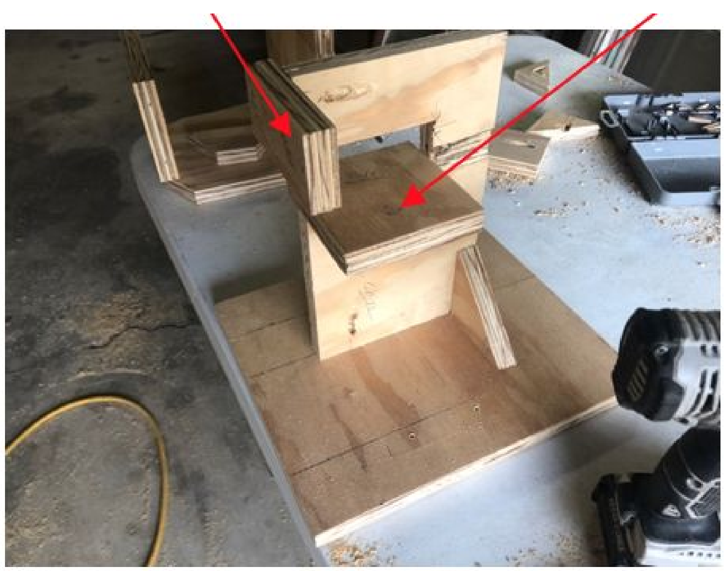


Figure 13: Mounting surface for servos

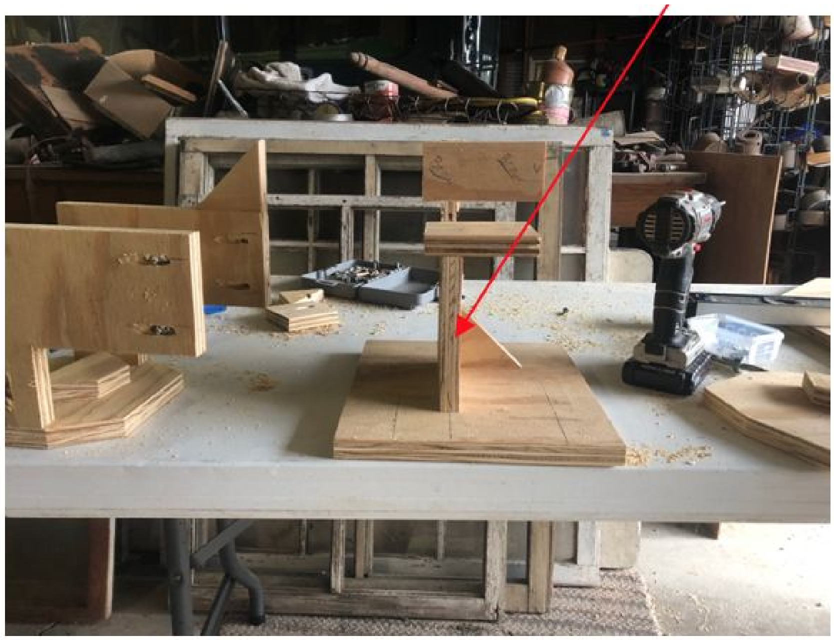


Figure 14: Mounting surface for chin servo