ANIMATRONICS FACE: PROFESSOR BRENDY RINCON TROCONIS

Ryan Beckmann
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Zbb122@my.utsa.edu

Bryan Ottinger
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
xjr823@my.utsa.edu

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

TABLE OF CONTENTS

ABSTRACT

The project in synopsis oversaw the creation of an animatronic face of a UTSA professor that responds to two individual input interactions. The overall project was within a strict and tight budget and so the face is a flat image of the professor (Professor Brendy Rincon Troconis) displayed on cardboard and attached to the front of a wooden box. The first interaction is caused by any motion, particularly by hand, that takes place in front of the ultrasonic sensor which triggers a servo motor output response making the mouth of the animatronic face open and close as the distance of the object motioning gets closer or further away from the sensor. In simple the amount the servo motor rotates and thus opens the mouth depends on the distance from the sensor the object is. The second interaction is controlled through the amount of force applied to the force sensor which controls the servo motors that control both eyes equally. As the force sensor is pressed on, bent and flexed the resistance in the sensor is changed and the amount that the eyes moved by the angle of the servos being turned is adjusted as such. In addition, a prerecorded greeting from the professor would play back from a sound shield when the first sensor interaction is triggered to align with the mouth response also triggered by that same sensor to mimic that the face is speaking to the person that is interacting with it.

Section 1: Literature Review

Robot Head 2

This robot was made primarily of wood. The head, jaw, neck, and box it was mounted to were all constructed out of plywood. The connections were made using various screws, bolts, and nuts. The pushrods used for manipulating the head were 1/8” brass rods. The head is a silhouette in the X-Y plane, as well as the jaw, and the eyes, lips, and eyebrows are in Z. The face talks through two computer speakers located in the box beneath the face. Servo motors attached to pushrods were used to control the face using a MiniSSCII controller. The actions these motors control are as follows: the eyebrows move up and down, the eyes move up and down, the head moves up and down, the neck rotates left and right, and the jaw moves up and down. No sensors were used to actuate any of the motors. The motors were all controlled through written code. The creator programmed multiple routines for different audiences. Robot Head 2 can be seen in action in video reference [1]. The creator listed all materials and tools used, as well as a description of the process of building and programming the animatronic on Instructables.com [2].

Figure 1: Robot Head 2

MiRAE

This robotic face is displayed in a minimalistic robotic skeleton form with all of its electrical components exposed. The face reciprocates human-like expressions towards the user when interacted with. The data used to make facial expressions was compiled over several years to make accurate expressions in response to user interaction. This is a more advanced approach that will be non-feasible to attain during this lab semester, but the mechanical design is something that can still be used for influence over a possible project choice. Focusing on the mechanical approach a main focus of this projects design was based around the structural support of the face. Erector sets were used to create a basic rectangular frame for the face. The mouth and eyebrows are made up of 2 pipe cleaners. The mouth is controlled with a servo motor at each corner to apply tension and relaxation to the pipe cleaners. The eyebrows and eyes display simplistic tilt up and down responses using additional servo motors. An Arduino and prototyping board are used as the digital input and output interpreter. A camera sensor would be used in this application to trigger a response by the animatronic face. The equipment used for this project consists of; 1 Arduino Uno, 1 Arduino Uno Prototyping Shield, Breakaway Headers (for the prototyping shield), 10 sub-micro servos (we use Hitech HS-55), 4 micro multipurpose Servo Brackets, 2 Tamiya Universal Plate L (210x160mm), 1 Tamiya Long Universal Arm set (longer option), 28 Tamiya Universal Metal Joints, jumper wires, pipe cleaners, 1 roll of Steel Gauge Wire (22-Gauge). All equipment used as well as a guide to constructing this machine can be found at [3].

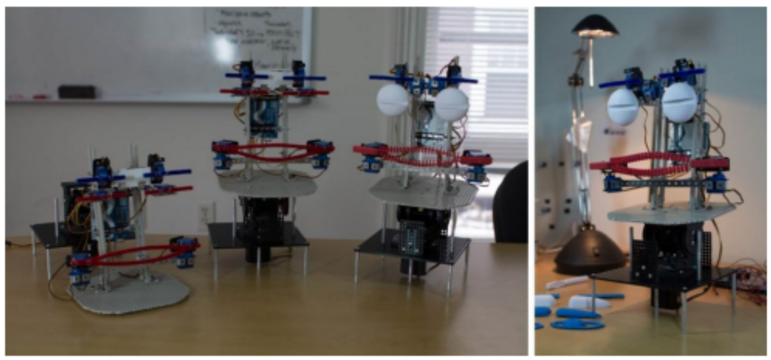

Figure 2: MiRAE

Kosmo's Face

This robot is an animatronic face integrated into a synthesizer. The purpose of the animatronic was to add to the visual and audio effect of the synthesizer. This project used scrap wood (multilayered plywood) for the housing unit to support all the features of the face and circuit board. The face has 3 sets of moving facial features; the pair of eyes, the eyebrows/eyelids, and the mouth. The method used for the face was primarily based around the use of several servo motors to exhibit basic push/pull control over each of the face's moving components. The eyes used two servo motors each to control up and down movement along with side to side movement within the gyroscope. A single gyroscope each controls the up and down movement of the mouth and eyebrows. The face is controlled using a Teensy 3.2 microcontroller which uses turn dials as the input control for the facial movements. The face also appears to use a motion sensor which triggers an audio and visual response that wasn't specified in full detail. The equipment used for this project consists of; 8 servo motors with zip tie connections, a Teensy 3.2 microcontroller, gyroscope arrayed metal frames for eyes, two ping pong balls for eyes, multilayered plywood for the frame, metal frame for face cover, chattering teeth for mouth, bread board, wiring, turn dials for control, audio speaker. (The face has lights behind the eyes and a possible motion sensor, but this was never mention so, led lights are an assumed additional used equipment along with possibly an ultrasonic sensor.) The equipment as well as the process used and videos of the robot can be found on Instructables.com [4].

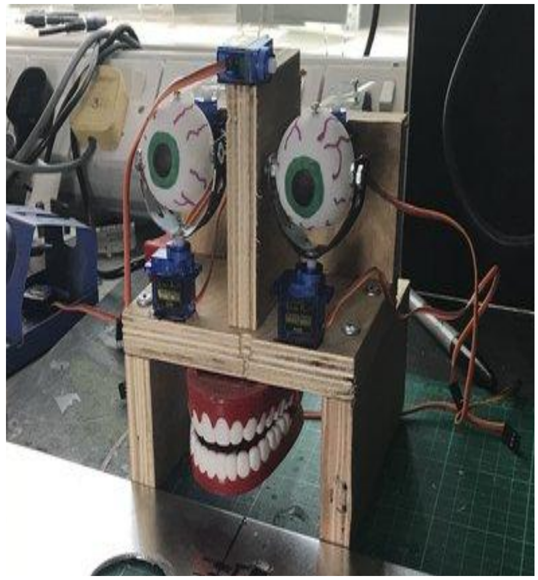

Figure 3: Kosmo's Face

Fritz

Fritz is another animatronic face that can be seen in [5]. It was constructed mainly of ¼" plywood sheets along with a few 3-D printed parts for the eyes. The head as well as supporting structure were laser cut pieces of a thin plywood. Most pieces of this animatronic were locked together using laser-cut sections that had been designed using a CAD program and hot glued. Nuts and bolts were another method of securing pieces. Eleven mini servo motors were used to actuate the eyes, eyelids, eyebrows, and mouth. Two bigger servos were used to move the head and neck. The servos were attached to the wood with screws, and to the body part to be moved using wire. An Arduino was also attached to the inside of the head. Wooden gears that had been laser-cut were used to assist in raising and lowering the head. This animatronic uses an ultrasonic sensor not to actuate individual motors, but to initiate routines that have already been programmed. All information regarding the construction of this animatronic were found on Instructables.com [6].

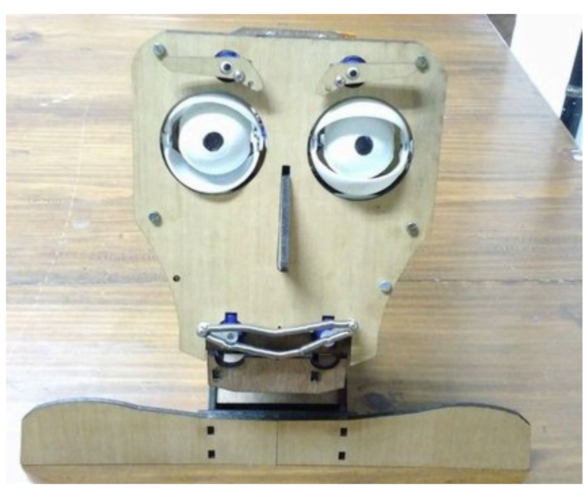

Figure 4: Fritz

FNaF

A third animatronic face that was studied was a character from a video game. This face was very simple and had only eyes and a mouth that moved. This can be seen in [7]. The face itself was a printed picture of the character glued to a Styrofoam sheet. The eyes were made from ping-pong balls. The face was mounted to a wooden frame using popsicle sticks. The servos were also held in place using popsicle sticks. Three servos were used to move the eyes and one for the mouth. The motors were controlled with an Arduino UNO. The instructions for this project can be found on Instructables.com [8].

Figure 5: FNaF

OHBOT

This robotic face is a commercial product pre-made and available to purchase for the purpose of teaching the basics of building such projects. Following the directions, the method used for this project uses similar techniques as the other project in order to control the eyes using 2 servo motors for the lateral and vertical control of each eye. In this project however, the eyes remain dependent on the same tilt and turn servo motors and remain synced for each movement. The mouth control in this project uses a servo motor on each side to control what would be the upper and lower lip. An additional servo motor controls the eyelid motion which is one continuous piece for the eyelids. The last small servo motor is on the side of the face to control the head tilt. The large servo motor is located at the neck base to control the head turn. All motors are controlled through the circuit board housed in the neck base of the face. An input power/USB chord is used for the input control of the device's movement. The equipment used for this project consists of; 6 small servo motors, 1 large servo motor, 2 premade eyes, 1 base mount, 1 eye mount plate, 1 mouth plate, 2 side of face plates, 2 mouth bars, 2 zip ties, electrical wiring, H-brain circuit board, power/USB cord, metal bent rods, screws. References [9] and [10] describe the process of constructing the OHBOT.

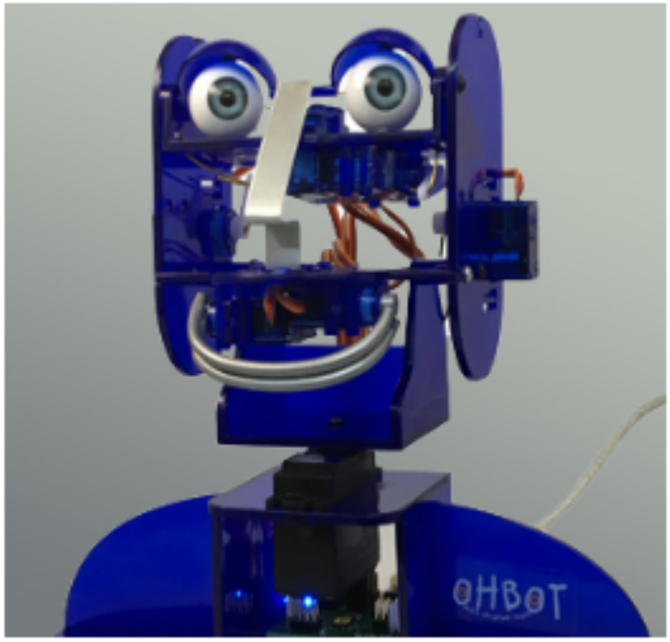

Figure 6: OHBOT

Section 2: Brainstorming (initial planning)

The animatronic face should have multiple features that actuate in response to a certain stimulus. Those features may include the eyes, eyebrows, nose, mouth, and possibly the face as a whole. The eyes could move up and down, left and right, or a combination thereof. A combination of movements may not be ascertainable during the given timeframe; therefore, movement may be limited to up and down or left and right, moving in coordination with the face as it talks. It may even be required that the eyes move in unison rather than independently depending on time and supplies. The eyebrows might also need to move in unison due to the time constraints and they can raise and lower to show further expression. The nose can rotate in a limited arc of no more than 20° to either side of the center line to prevent an unnatural appearance. The movement of the mouth can be restricted to vertical motion, allowing the face to open and close its mouth to talk and say an expression. The whole face/head can move side to side or rotate back and forth around a pivot point depending upon how the face is attached to the frame.

These moving features would all be controlled by servo motors similar in the approach to what was researched in section 1. The eyes can be placed in gyroscopic half circle metal brackets and can be moved vertically and horizontally using a servo motor for each of the directions. The motors can be programmed to move in conjunction with one another to produce angular movement. The mouth can move vertically open and closed using one servo motor, with the lower jaw being the portion of the mouth that moves. The eyebrows can raise and lower vertically using one to two servo motors. Each of the servo motors can be connected to each of the moving facial components using either small metal or wooden rods or plastic zip ties. The structure for housing the device can be made up of wood; something cheap such as plywood or scrap wood, or constructed from pieces of a metal erector set. The structure can be similar to that of the design for Kosmo’s face that was researched in section 1. Small pieces of square wood will be placed in a post and lintel style structure and the mouth can be attached underneath the lintel portion. The eyes could either be secured to the upper surface of the lintel portion of the wood with another rectangular wooden block being placed between the eyes to create the nose bridge allowing the face to be placed on the front surface of the wood, or they could be affixed to cages made from pieces of the erector set if that method is chosen. A flat surface mask with an extended nose would be the preferred face structure to use as it is flat and will adhere to the wood easily using Velcro. The eyebrows will be on the exposed face the control rods will be fed through the mask.

Two sensor stimulus input triggers will be used for the face, a simple button and an ultrasonic motion detecting sensor. Other options for sensors include a PIR sensor, an accelerometer, or a piezo disk vibration sensor. A prerecorded expression by the predetermined professor saying a short greeting will be used to playback in audio when the motion sensor is triggered at a set distance, or when it is moved. The motion sensor will command both the facial movements and the audio to play at the same time allowing those actions to be synchronized. An MP3 shield will need to be acquired for the Arduino to play the audio file. The sound track will be a recording of something related to the professor's interests. All the sensors being used will be connected through an Arduino board that will be powered by a 9V battery. The speaker will be separately powered by its own battery.

Section 3: Supporting structure

For this part of the project, 1/8" sheets of plywood and scrap pieces of wood were used to construct the supporting structure. The sheets of plywood were cut to 11-inch squares using a table saw. The scrap wood was cut to a one-inch square cross section and the lengths were cut according to their location on the frame, 11 inches for the side pieces and 9 inches for the back pieces. The locations of the rectangular sections were measured and marked with a pencil corresponding with the distance from the top of the head to the eyes and from the eyes to the mouth of an average person. The rectangular pieces were then lined up on the plywood and attached using heavy-duty staples. Once all five sections were complete, they were assembled together with staples. The two shelves were slid into place and rested on the supports. The completed supporting structure can be seen in Figure 7. Pieces of Velcro were attached to the frame for affixing the cardboard face to the structure. This can also be seen in Figure 7.

Figure 7: Supporting Structure

Section 4: Joints and motors

The joints used in this project were very simple. The eyes were simply attached directly to the servos using a small piece of wire that was stuck inside the rotating portion of the servo as well as into the ping pong balls that were used for the eyes. The joints were then secured using hot glue. The eyes were attached to individual servos and then tied to the same pin on the Arduino so that they moved together. The mouth was controlled using a servo. The mouth was attached to the structure by rolling up two strips of paper into tubes, as seen in Figure 8, taping the paper tubes to the back of the mouth, and then routing the plastic straws that were taped to the frame through the tubes. This restricted the motion of the mouth to only the vertical direction which was needed because the motion of the servo that was used to move the mouth was angular. The straws can be seen in Figure 7 and were attached to the frame by folding a portion of the straws over the shelf in the supporting structure as well as the bottom of the structure and taping them in place. Duct tape was used to restrain the straws on the bottom of the structure and Scotch tape was used to fasten the straws to the shelf. The mouth was attached to the servo, as well as positioned, by taping a piece of fishing line to the back of the mouth and then tying it to the plastic piece that comes with the servo.

Figure 8: Mouth Joints

Figure 9: Motors

Section 5: Sensors

The two sensors used in this project were an HC-SR04 ultrasonic sensor and a force sensitive resistor (FSR) from Adafruit. The ultrasonic sensor was used because it has the ability to actuate a motor after being triggered at a certain distance. It is not dependent upon light or sound which may be difficult to control in the presentation area. The FSR was chosen for its simplicity. As with the ultrasonic sensor, the FSR is not dependent upon light or sound and was thus a good candidate for use in this application. Also, during the brainstorming portion of this project, an external hand was considered for activation of the motors. This could have been activated using an FSR but this idea was scrapped for a simpler design. The ultrasonic sensor works by sending out a sound wave and tracking the amount of time it takes for the wave to be reflected back. The time is converted to a distance using the speed of that sound wave. The microcontroller is programmed to actuate the motors in response to the sensor being triggered at a certain distance. The FSR works by using a conductive film, spacer, and conductive print on a substrate. When more force is applied, more of the conductive print comes into contact with the conductive film, thus lowering the resistance of the material. This change in resistance also affects the voltage that is relayed to the microcontroller. The microcontroller was programmed to activate a certain motor when it receives a certain voltage from the sensor.

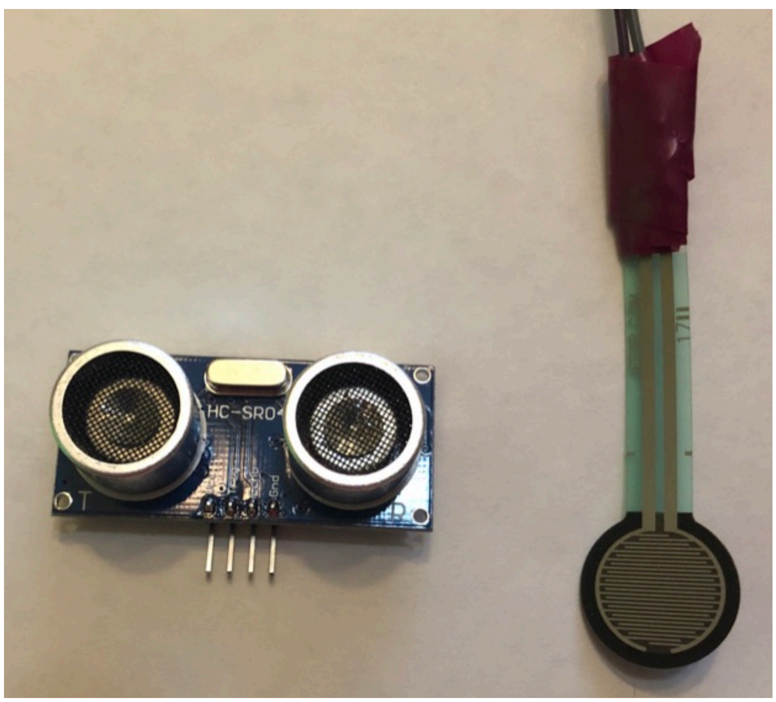

Figure 10: Sensors

Section 6: Programming for Interaction

Programming this animatronic face for interaction included programming each sensor and motor individually and then merging the two programs and modifying the code from there. At first, each sensor, the ultrasonic and the FSR, were programmed for functionality. For the most part the general code was available online but, in some cases, it had to be generated by the team. Once the functionality of each sensor was verified, it was paired with a motor or set of motors. The ultrasonic sensor was paired with the two servos that controlled the eyes. They were attached to the same pin so that they moved in coordination with each other. The motors were programmed to respond to the ultrasonic sensor in that they would move to one position when the distance read by the sensor was between 7 and 14 cm, and move to another position when the distance was less than 7 cm. This movement was restricted so that the eyes had a more natural appearance rather than just spinning in circles. The mouth was programmed to respond to the FSR. It was set to move a scaled distance of the voltage read by the FSR. This was scaled to show a more realistic movement of the mouth. This project can be viewed at https://youtu.be/bhQWp0692Jg. When the video was taken, the project did not want to work as fluidly as it had before so additional videos showing intermediate steps where everything was working properly can be found at the following links. Eyes: https://youtu.be/YGVtutEqXG0. Mouth: https://youtu.be/VWnRvixCM14. Eyes and mouth moving in conjunction with one another: https://youtu.be/hCVYgemTZ3Q.

Section 7: Lessons learned and suggestions (1 page)

1. This project helped in shaping a better understanding of integrating circuit boards and electrical components and being able to access and control them through the use of coding in Arduino in order to create an interactive and responsive system.

2. Along with the understandings in the electrical and coding domain the project also gave some insight into how to go about in erecting physical and mechanical components such as the motors, joints and supports used in the project to make sure that it functioned as it was desired to.

3. This project was very helpful in understanding the importance of troubleshooting when a situation occurs as many problems arose throughout the duration of the project involving both mechanical and electrical components. These problems were often times very specific and required a specific solution to meet the needs of what was needing tending to.

4. One of the first issues that arose during the project was in the issue of finding out a way to allow the mouth to move specifically in and up and down motion as the project required without any side to side or forward and back movement. This problem was mechanical in nature and as so needed a mechanical solution which was found by having the mouth be guided along two vertical posts (straws) in parallel to one another. The mouth would be guided along its desired vertical up and down path using tubes fixed to its back side with the vertical posts running through them. When pulled by fishing line attached to the mouth at one end and to the servo motor at the other end feed through a small pulley system the mouth piece would rise up and down as originally wanted.

5. Among some of the things that went wrong during the project was getting the second sensor to work properly in controlling the output servo motors. For the project an ultrasonic sensor was used as the primary sensor and a force sensitive resistor or “pressure sensor” was used as the secondary sensor to control the eye movements. When using the pressure sensor there wasn’t always a response from the motors that would occur when the sensor was bent and pressed on as it requires for the change in resistance causing the response of a change in position in the motors. This was on ongoing situation during the sensors use for the project and created some difficulties in the presentation of the project. It was noted that by flexing the sensors instead of pressing on it, a better response by the motors was observed and so that technique was used when the project was tested.

6. Another issue that arose during the project was trying to find a way to get the code to run the audio using the Adafruit music maker shield in conjunction with the motor response from the same sensory input. For this project the ultrasonic senor was used as the input sensor that when triggered by motion within its range would cause both the mouth to move by control of the servo motor and the audio to play from the Adafruit shield. However, the response would only trigger the mouth to move instead of having both be triggered simultaneously. To accommodate for this during the presentation a separate code to trigger the audio was written in Arduino and run when the input sensor was triggered in one demonstration and then the original code with the servo response was demonstrated in another.

Section 8: Personnel and bill of materials

(a) Personnel

Task	**Main Personnel**	**Secondary Personnel**
Structure Construction	Beckmann	Ottinger
Joints and Motor Interfacing	Ottinger	Beckmann
Integrating and Interfacing Sensors	Beckmann, Ottinger	
Programming	Beckmann	Ottinger
Abstract	Ottinger	
Literature Review	Beckmann, Ottinger	
Brainstorming	Beckmann, Ottinger	
Supporting Structure	Beckmann	
Joints and Motors	Beckmann	
Sensors	Beckmann	
Programming	Beckmann	
Lessons Learned	Ottinger	

(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	3	6.95	20.85
3	Wood	Scrap	-	-	-
4	HR-SC04	https://www.amazon.com/	1	6.89	6.89
5	FSR	https://www.amazon.com/	1	9.01	9.01
6	1/8” Sheet Plywood	https://www.lowes.com/	1	14.98	14.98
7	Ping Pong Balls	https://www.walmart.com/	1	6.48	6.48
8	Cardboard	On Hand	-	-	-
9	Jumper Wires	On Hand	-	-	-
10	MP3 Shield	https://www.amazon.com/	1	32.49	32.49
	Total	-	-	-	90.7

References:

[1] Robot Head 2

https://youtu.be/AhPnrvdMArM

[2] Talking Animatronic Robot Head

https://www.instructables.com/id/Talking-Animatronic-Robot-Head/

[3] MiRAE Robotic Face

http://r-house.sice.indiana.edu/mirae/MiRAE_Construction_Manual.pdf

[4] Kosmo's Face tutorial

https://www.lookmumnocomputer.com/projects/#/kosmos-face/

[5] Fritz

https://youtu.be/1A_sELjnYPs

[6] FRITZ – ANIMATRONIC ROBOTIC HEAD

https://www.instructables.com/id/FRITZ-ANIMATRONIC-ROBOTIC-HEAD/

[7] DIY Animatronic Robot

https://youtu.be/ZhNy46JhXyc

[8] Animatronic Robot: Five Nights at Freddy's

https://www.instructables.com/id/Animatronic-Robot-Five-Nights-at-Freddys/

[9] OHBOT

https://www.ohbot.co.uk/about.html

[10] OHBOT Instructions

https://www.ohbot.co.uk/uploads/2/1/8/3/21834178/ohbot_instructions_v21aardvark_v4a.pdf

[11] Additional source for robotic faces

http://mindtrans.narod.ru/heads/heads.htm

Appendix A: Code

```
#include <SR04.h>
#include <NewPing.h>
#include <Servo.h>
#include <SPI.h>
#include <Adafruit_VS1053.h>
#include <SD.h>
#define SHIELD_RESET  -1
#define SHIELD_CS     7      // chip select pin (output)
#define SHIELD_DCS    6      // Data/command select pin (output)
#define CARDCS 4     // Card chip select pin
#define DREQ 3       // Data request
Adafruit_VS1053_FilePlayer musicPlayer =
  Adafruit_VS1053_FilePlayer(SHIELD_RESET, SHIELD_CS, SHIELD_DCS, DREQ,
CARDCS);
Servo myservo;
Servo Servo1;
const int trigPin = 2;
const int echoPin = 4;
int vib = A0;
int val;
void setup() {
Serial.begin(9600);
myservo.attach(12);
Servo1.attach(13);
Serial.println("Adafruit VS1053 Simple Test");
  if (! musicPlayer.begin()) {
     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);
  }
  printDirectory(SD.open("/"), 0);
  musicPlayer.setVolume(20,20);
  Serial.println(F("Playing track 001"));
  musicPlayer.playFullFile("/Track001.mp3");
}
void loop() {
val = analogRead(0);
val = map(val, 0, 1023, 20, 40);
Servo1.write(val);
long duration, cm;
pinMode(trigPin, OUTPUT);
digitalWrite(trigPin, LOW);
delayMicroseconds(2);
digitalWrite(trigPin, HIGH);
delayMicroseconds(20);
```

```
digitalWrite(trigPin, LOW);
pinMode(echoPin, INPUT);
duration = pulseIn(echoPin, HIGH);
cm = microsecondsToCentimeters(duration);
if ( cm > 7 && cm < 14)
{
myservo.write(100);
delay(400);
}
else if ( cm < 7)
{
myservo.write(40);
delay(100);
}
else
{
myservo.write(40);
delay(100);
}
Serial.print(cm);
Serial.print("cm");
Serial.println();
delay(100);
}
long microsecondsToCentimeters(long microseconds) {
return microseconds / 29 / 2;
 if (musicPlayer.stopped()) {
    Serial.println("Done playing music");
    while (1) {
      delay(10);  // we're done! do nothing...
    }
  }
  if (Serial.available()) {
    char c = Serial.read();
    if (c == 's') {
      musicPlayer.stopPlaying();
    }
    if (c == 'p') {
      if (! musicPlayer.paused()) {
        Serial.println("Paused");
        musicPlayer.pausePlaying(true);
      } else {
        Serial.println("Resumed");
        musicPlayer.pausePlaying(false);
      }
    }
  }

  delay(100);
}
void printDirectory(File dir, int numTabs) {
```

```
  while(true) {
    File entry =  dir.openNextFile();
    if (! entry) {
      Serial.println("**nomorefiles**");
      break;
}
    for (uint8_t i=0; i<numTabs; i++) {
      Serial.print('\t');
 }
    Serial.print(entry.name());
    if (entry.isDirectory()) {
      Serial.println("/");
      printDirectory(entry, numTabs+1);
    } else {
      Serial.print("\t\t");
      Serial.println(entry.size(), DEC);
    }
    entry.close();
  }
}
```

Appendix B: Arduino Wiring

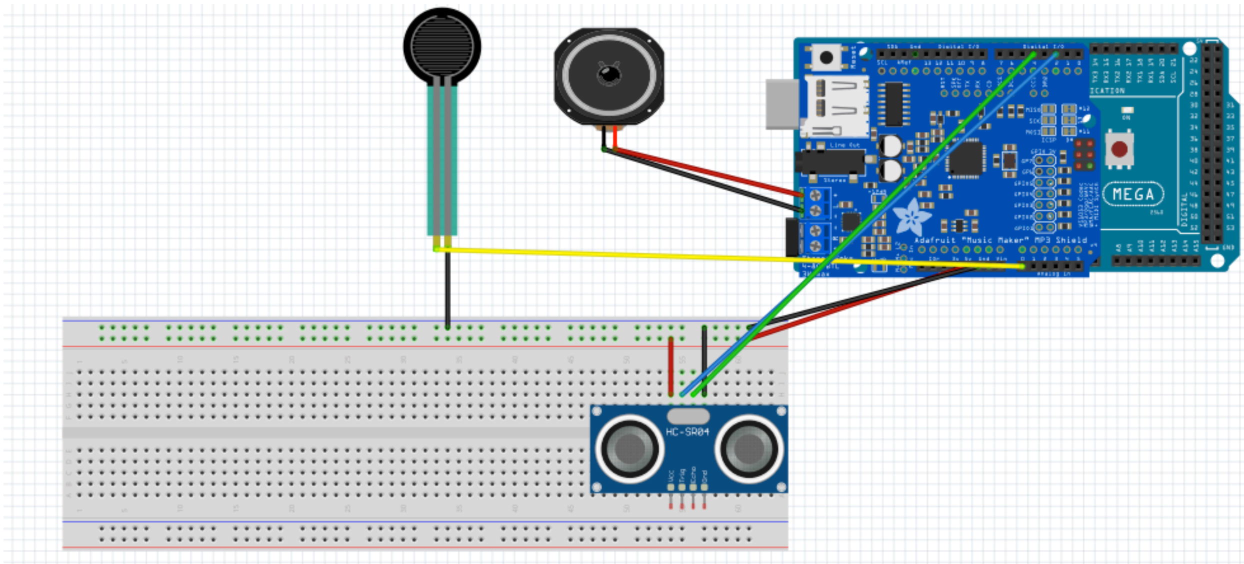

Figure 11: Fritzing Sketch of the Setup