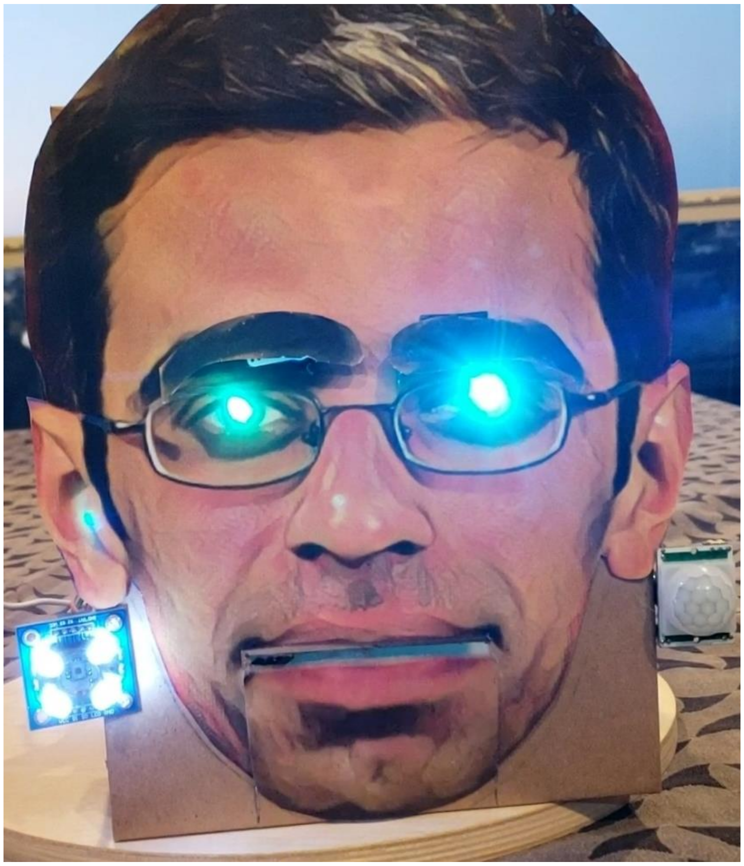
BNS-L HUMAN CYBORG RELATIONS PROTOCOL DROID DR. BHONSULE ANIMATRONIC



Jonathan Mayberry
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Anu941@my.utsa.edu

J. Ramon Vazquez C.
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Xnr451@my.utsa.edu

Toechukwu Udegbue
Dept. of Mechanical Engineering
San Antonio, TX, USA 78249
Toe.udegbue@gmail.com

Submitted as part of project for ME4543 Mechatronics, Fall 2018

TABLE OF CONTENTS

ABSTRACT

The goal of this project was to explore a minimalistic approach to design and develop a functional and interactive animatronic face of a professor from UTSA's Mechanical Engineering Department, as well as to build the foundation in the engineering of a mechatronics system. The broader aim was to create a simple, inexpensive, and replicable robotic face that can look at colors and tell you precisely what colors they are as well as ton help further the "science" of robots and make the robotics platform more accessible to the public in general.

SECTION 1: LITERATURE REVIEW

MiRAE (Robot-Face) Construction Manual

MiRAE is an inexpensive, replicable, and minimalistic face animatronic. It was designed y Dr. Casey Bennett an artificial intelligence researcher from the School of Informatics and Computing at Indiana University as it is referenced in [1]. MiRAE was constructed using; An Arduino, with a prototyping shield for controlling the robot's movements and actions. An assortment of servo motors, this includes micro servos, sub-micro servos, and high torque servos. These were used to control the eyes, eyebrows, mouth, and neck movements. Tamiya universal plates, arms, metal joins, nuts and bolts to do the armature structure. The purpose of this robot is to explore minimalistic features necessary for a robotic face to engage in meaningful affective social interaction, as well to study questions related to the temporal dynamics of social cognition in humans and human-robot interaction [1]. Figure 1 shows different variations of MiRAE.

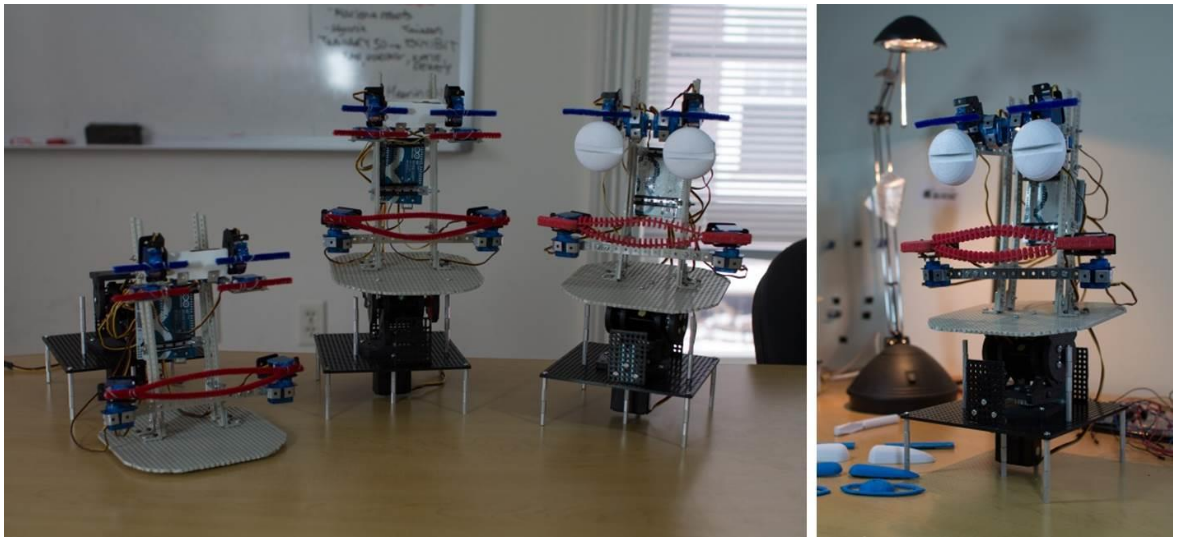


Figure 1: MiRAE Robot

Mecha Hitsu Gen2

The Mecha Hitsu Gen2 is fabricated to look like a skull. It is constructed of composite skull, steel, and various hardware. To operate the move the face features, it has 3 HS-645MG ultra-high torque servos. The motions include eye motion, jaw open and close, and proportional eye open and close. The Mecha Hitsu Gen 2 is controlled by the PWM servo controller, microcontroller, or simple RC system. It is used in haunted houses or scare houses to simulate zombies. [2] Figure 2 shows a front image of the Mecha Hitsu Gen 2 Robot.

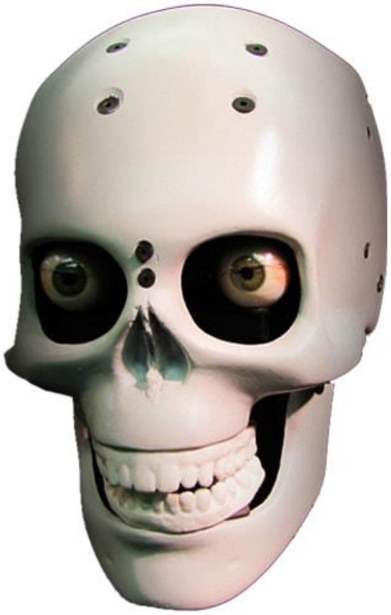


Figure 2: Mecha Hitsu Gen2

Kismet

Kismet is an animatronic created by undergraduate and graduate students from MIT. Kismet is an expressive robotic creature with perceptual and motor modalities tailored to natural human communication channels. To facilitate a natural infant-caretaker interaction, the robot is equipped with visual, auditory, and proprioceptive sensory inputs. The motor outputs include vocalizations, facial expressions, and motor capabilities to adjust the gaze direction of the eyes and the orientation of the head. Note that these motor systems serve to steer the visual and auditory sensors to the source of the stimulus and can also be used to display communicative cues. [3] Figure 3 shows Kismet's expressions to different stimulus.

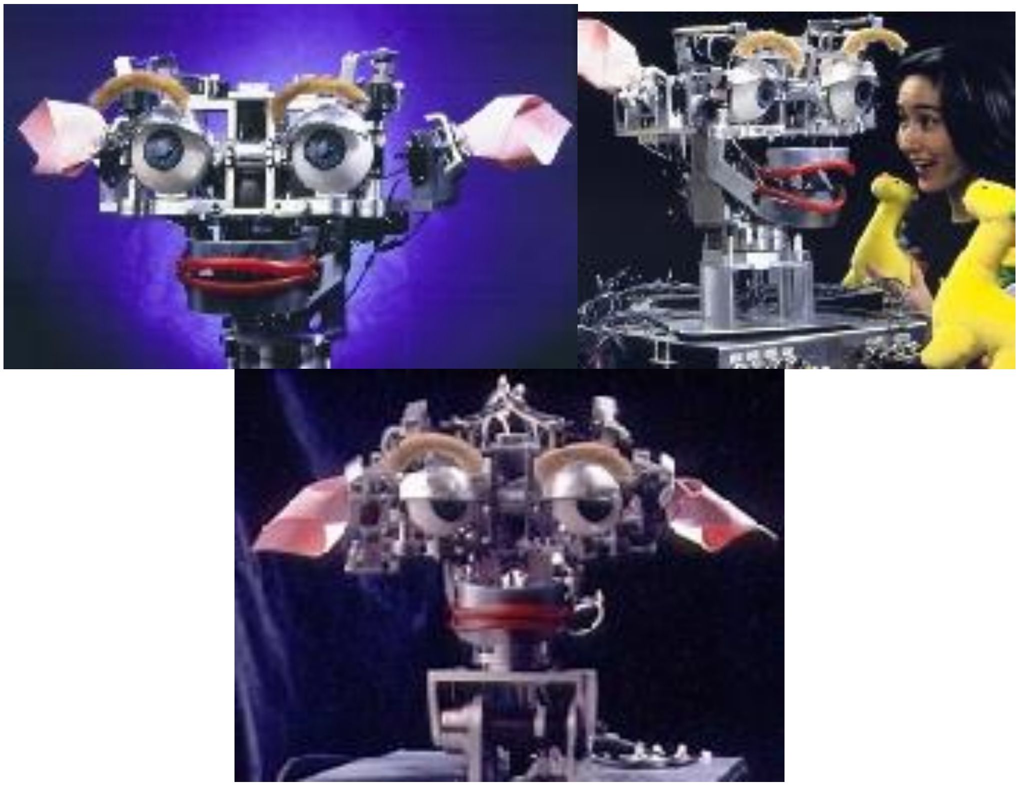


Figure 3: Kismet Animatronic

SECTION 2: BRAINSTORMING (INITIAL PLANNING)

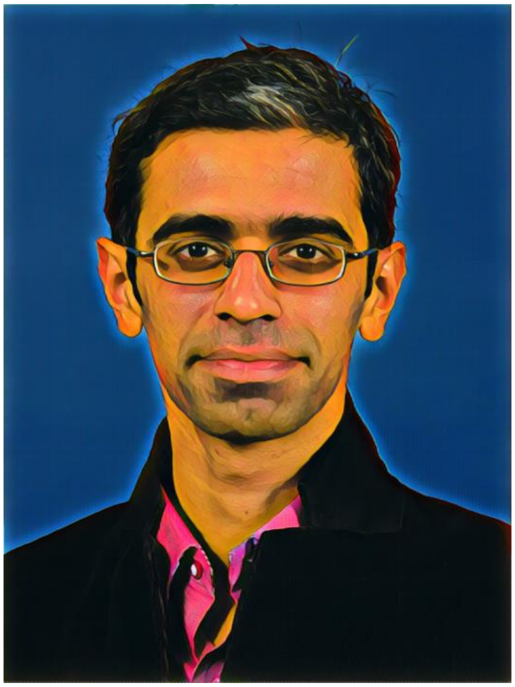


Figure 4: Dr. Pranav Bhounsule

This animatronic was pursued with the objective that it would be intended to have a purpose in life farther than only entertain, and making some people laugh or cringe. It was decided that this robot would be able to detect a color and replicate it and/or say which color it is. BNS-L was the name denoted to the animatronic in honor of its human inspiration, Dr. Bhounsule. The animatronic would be a protocol droid specialized in human and cyborg relations and it would be able to look at a color card and tell the user which color it was looking at.

This animatronic has three main moving parts; the mouth and both eyebrows. The movement mechanism used would be micro servos for each. The stimulus of the animatronic would be triggered by two different sensors; a Passive Infrared Sensor (PID sensor) which would be activated by motion, and a color sensor which would be able to read the values of Red Green and Blue light of a color presented to it and replicate them with RGB LEDs.

Since some of the main design objectives for this project were minimalism, and accessibility, the structure and chassis would be made from cardboard boxes, nuts, bolts screws, and wood. The face of the professor was obtained from UTSA's Mechanical engineering staff site. This image was enlarged and digitally edited to have a more artistic look. By positioning the printed image of the professor's face, and making cuts at corresponding locations on the boxes, the integration of the servos, structure and face of the professor was achieved.

For the sound it was decided that an Adafruit Music Maker MP3 Shied would be used and installed on top of the Arduino Mega. Introductory and conclusive phrases, as well as an aural test of the 6 principal colors would be recorded from the professor. In order to synchronize the audio tracks with the movement of the animatronic, Audacity, a digital audio editor and recording application software, was used. With that program, the audio signal waves can be analyzed, and the vowels and silences of a certain phrase or word identified. Once the time lengths of those were measured, the servo delays in the programing could be matched up with the phrases and thus the synchronization of the audio and the movement completed.

It is intended for this animatronic to work under almost any power condition, therefore it could be powered by connecting it to the computer, a DC socket, or a battery. Regardless, the Arduino would supply the necessary power for the three servos, the two RGB LEDs, the color and PIR Sensor as well as the MP3 Shield. Figures 1-3 show the animatronic reading and displaying the corresponding colors.

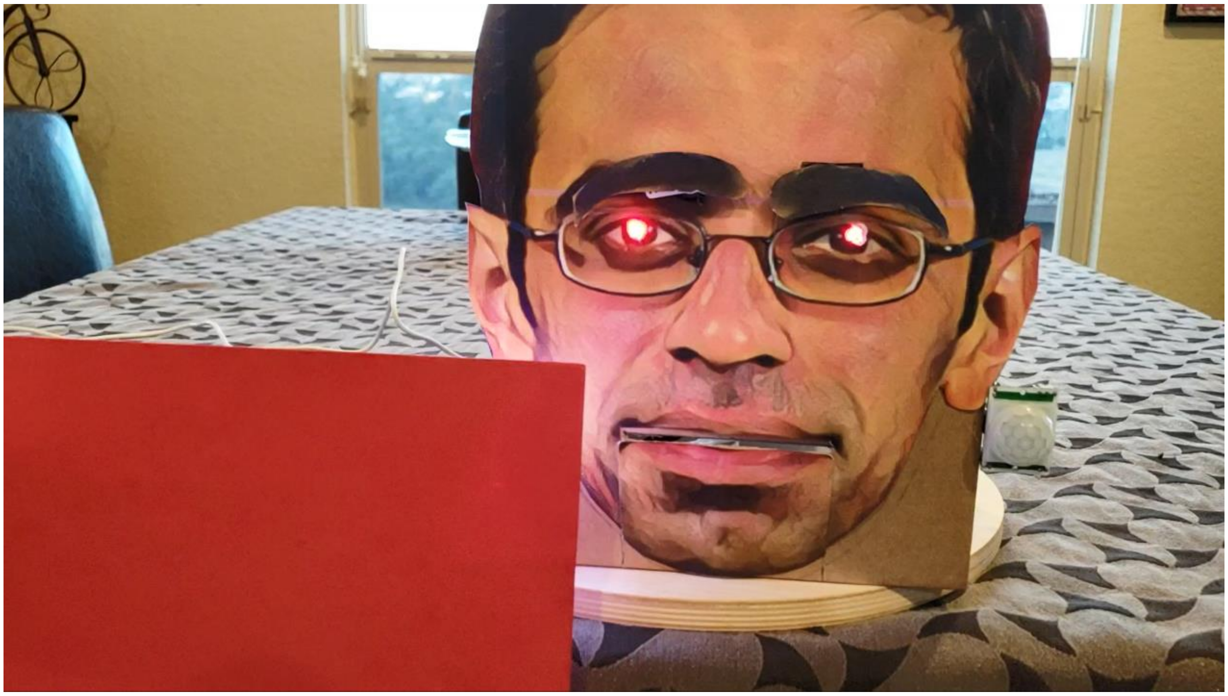


Figure 5: BNS-L Reading Red Color

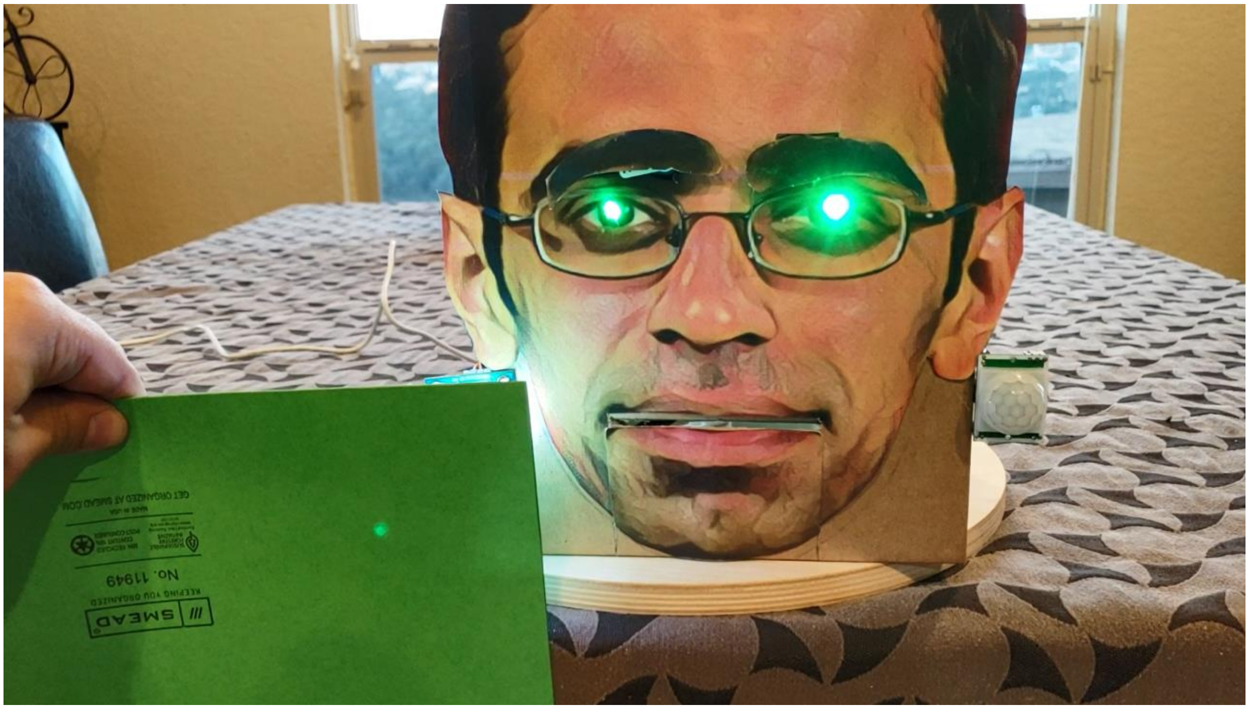


Figure 6: BNS-L Reading Green Color

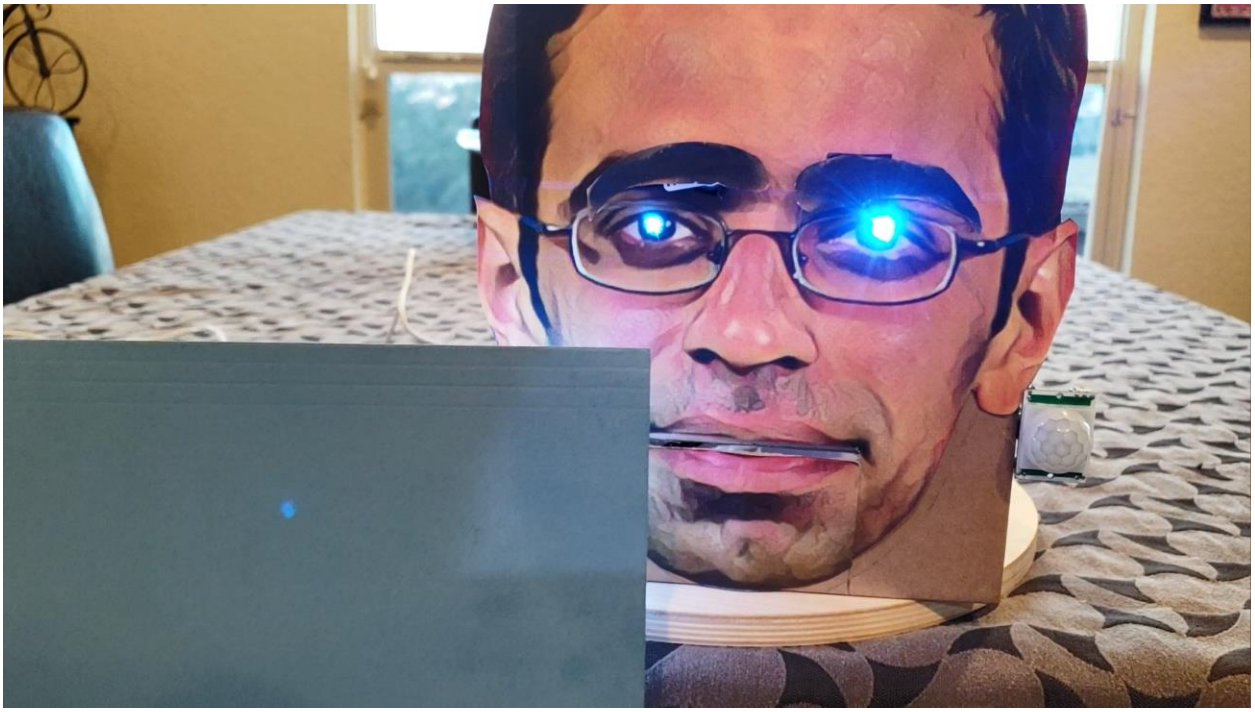


Figure 7: BNS-L Reading Blue Color

SECTION 3: SUPPORTING STRUCTURE

The supporting structure consisted of one wooden circular board and two cardboard boxes. Figure 8 and 9 show these materials.

Figure 8: 12-in Birch Circle

Figure 9: Framing Cardboard Box

The base was a 12” radius circular pine board and the boxes were regular 6” x 8” cardboard boxes. The professor’s image was digitally edited so that it could have an artistic aesthetic. The outline of the face was cut and positioned in the front face of one of the boxes. Key outlines of the face such as the eyes, eyebrows, nose, and mouth were marked so that appropriate cuts could be made. Once everything was dimensioned, cuts for the mouth and the eyebrows as well as perforations for the eyes were done on the markings. The lids for the boxes were cut ¾ of its length and were used for the top and bottom of the electronics enclosure. With bolts, all the parts that form the chassis, structure and face of the animatronic were assembled. On the back of the now formed box, holes were cut so that cable management was better when electronics were eventually introduced. Figures 10 - 14 show different components of the structure at different phases.

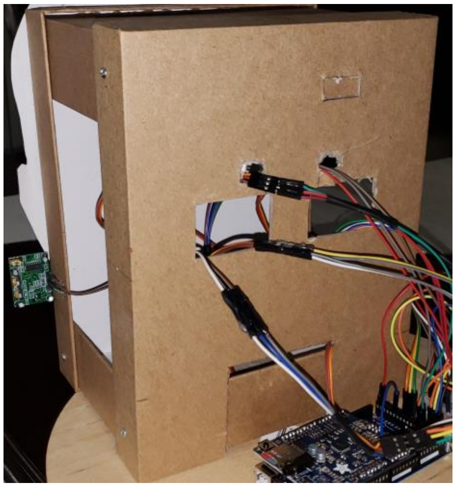


Figure 10: Back view of Animatronic Structure

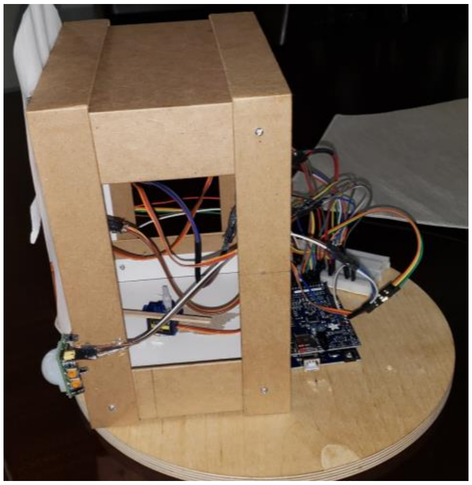


Figure 11: Side view of Animatronic Structure

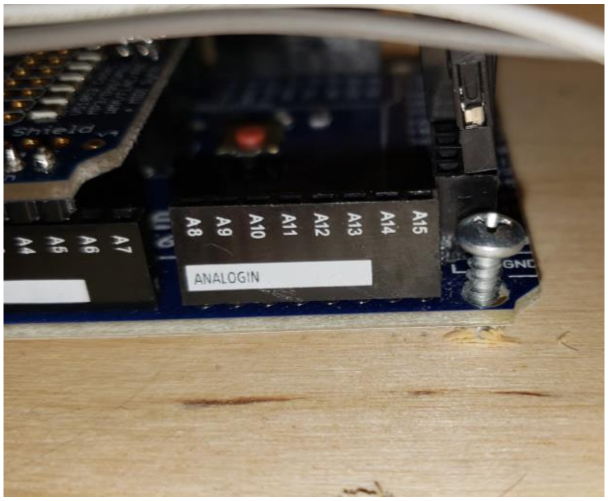


Figure 12: Bolted Arduino Mega

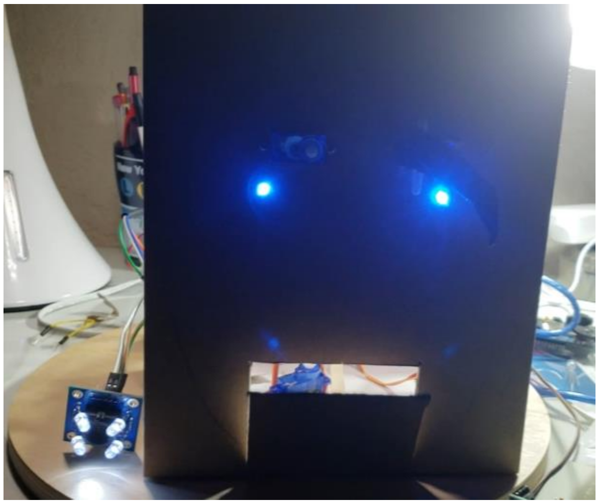


Figure 13: Preliminary Presentation for the Structure

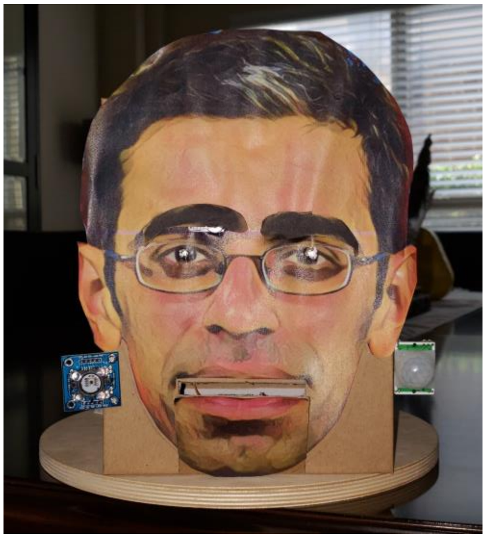


Figure 14: Finalized Structure Arrangement

SECTION 4: JOINTS AND MOTORS

As previously mentioned, this animatronic consists of three joints; two for right and left eyebrows, and one for the mouth. Each joint utilizes an SG90 Micro Servo seen in Figure 15.

Figure 15: SG90 Micro Servo

Eyebrows

The joints for the eyebrows consisted of mounting the servo motors where the eyebrows would be. For this, the outline of the servos was traced enclosed by the area of the eyebrows on the face. This was done carefully to make sure the cardboard wouldn't get damaged and the servos could sit securely. Since the micro servo motors have two screw holes, these were marked and perforated onto the cardboard box to tightly secure them in place. Figures 16 – 19 show the positioning and the mounting of the eyebrow servo motors.

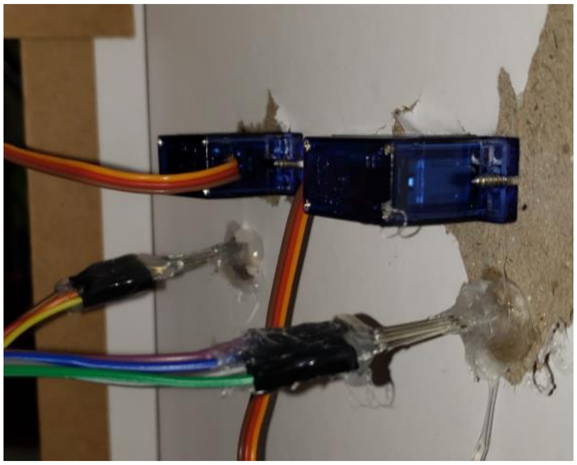


Figure 16: Inside View of Eyebrow Servo Mounting

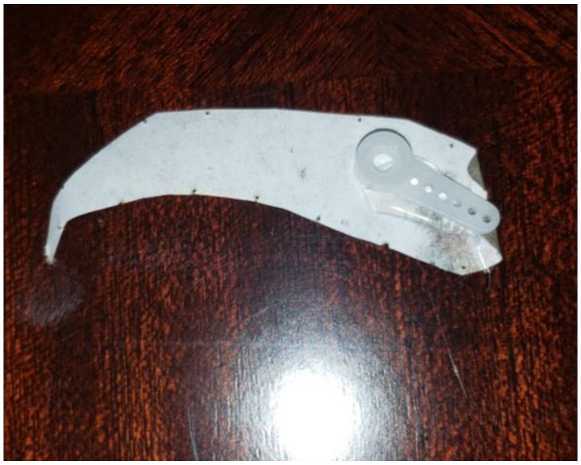


Figure 17: Servo Arm glued to Eyebrow Paper

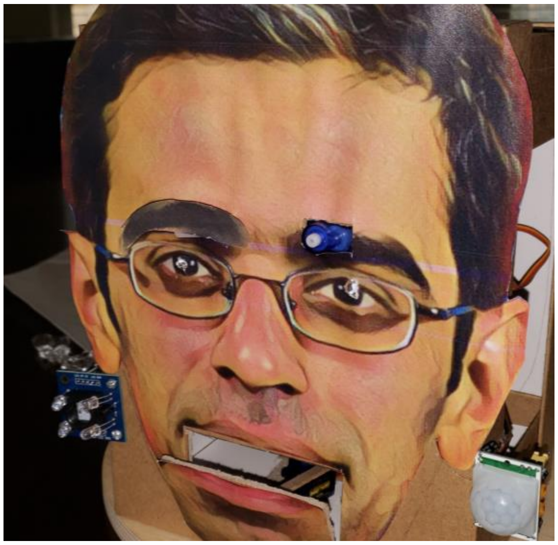


Figure 18: Eyebrow Servo Positioning in Structure with relation to face

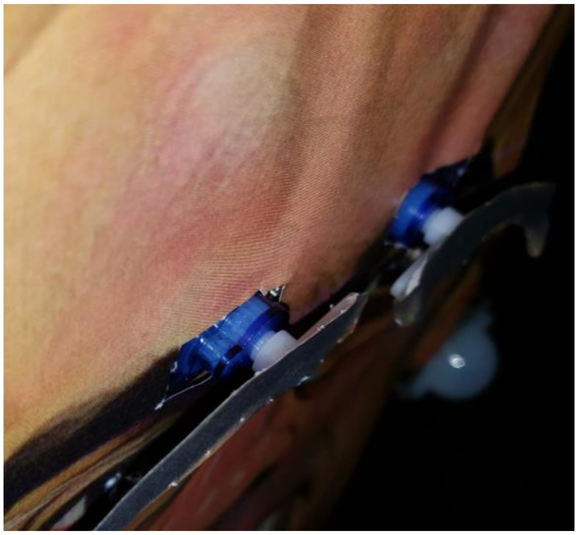


Figure 19: Top view of Servos mounted onto structure with servo arms and Eyebrows

Mouth

For the mouth joint, the mouth cut was performed in such a way that it would act as hinge, having the movement of the mouth to be outward and inward instead of upward and downward. A little wooden stick was attached to the servo arm using a small wire. The servo itself was glued to the bottom of the face box while the tip of the wooden stick was glued to the wall of the mouth. Note that one must make sure that the servo and the wooden stick are at corresponding distances so that they can move freely as seen on Figures 20 and 21.

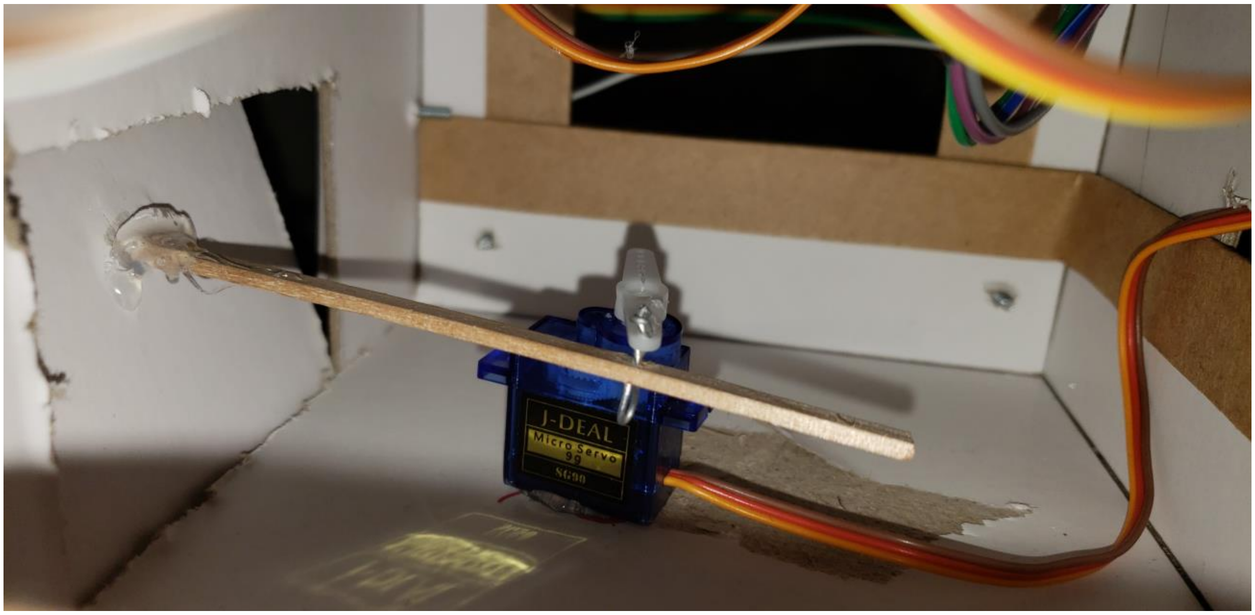


Figure 20: Mouth servo with wooden Rod attached to structure and joint

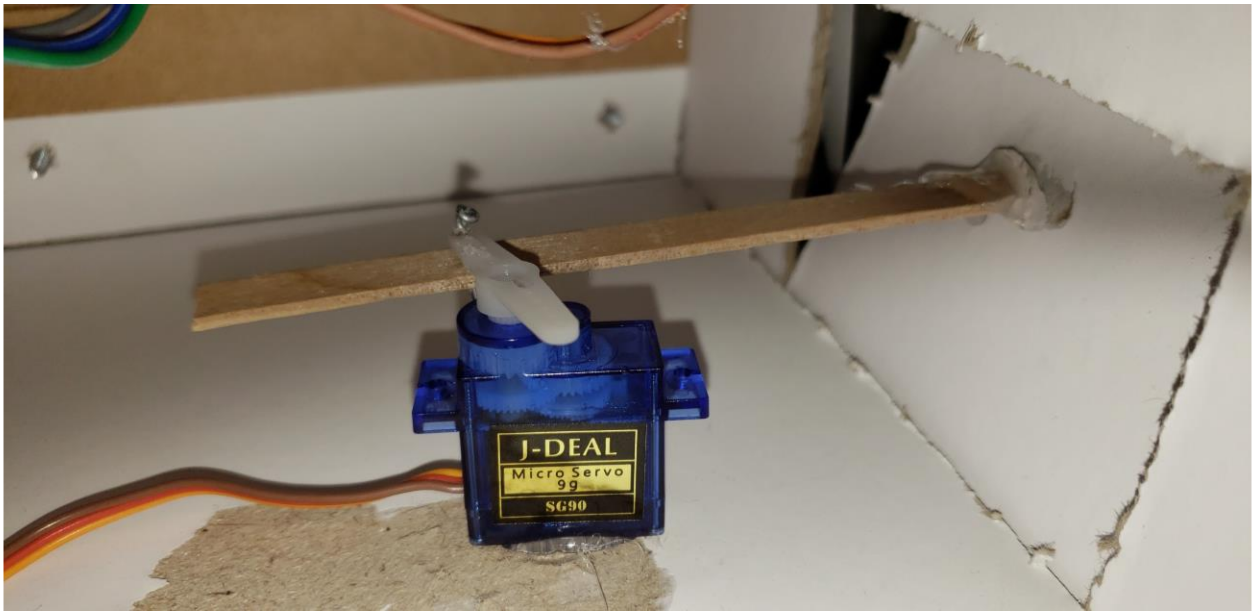


Figure 21: Servo mounted for mouth joint movement.

SECTION 5: SENSORS

This animatronic use two different sensors; An HC-SR501 Passive Infrared Sensor (PIR) and a TCS230 color sensor.

PIR Sensor

The working principle of the PIR module consists of a pyroelectric sensor which generates energy when exposed to heat. When a human or animal body gets in the range, the sensor will detect the heat energy in the form of infrared radiation emitted by the human or dog. The term passive means that the sensor is not using any energy for detecting purposes, it just works by detecting the energy given off by the other objects. The module also consists of a Fresnel Lens, which focuses the infrared signals on to the pyroelectric sensor. Figure 22 and 23 Show the PIR Sensor and a visual representation of how it works respectively. While Figure 24 shows the mounting of the Sensor onto the structure.

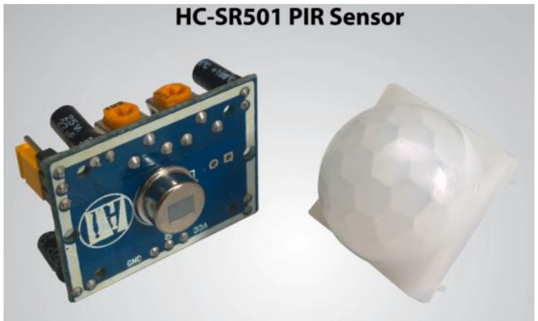


Figure 22: HC-SR502 PIR Sensor

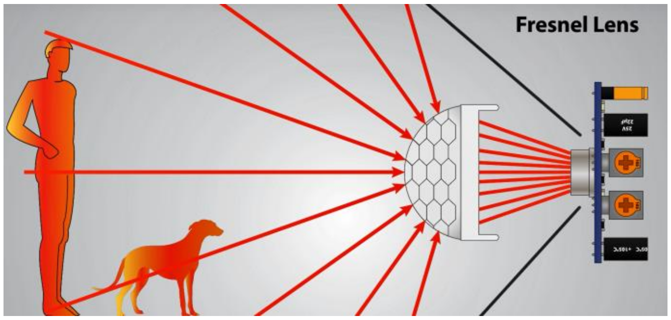


Figure 23: Functioning Principle of PIR Sensor and Fresnel Lens in Sensor Module

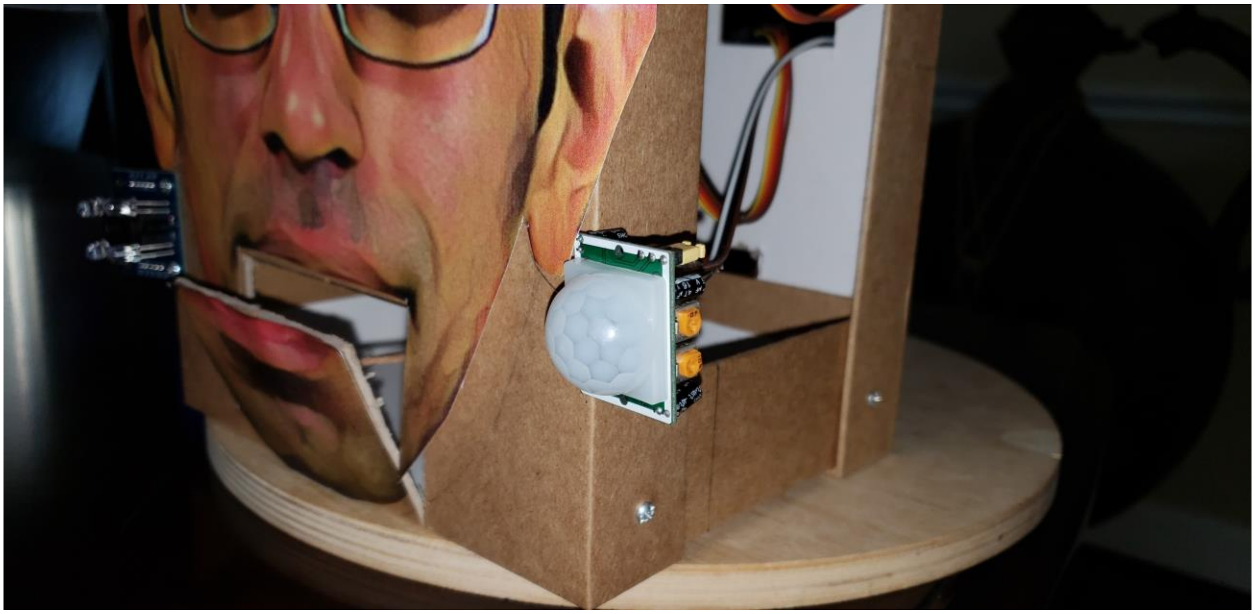


Figure 24: PIR Sensor Mounted on Structure

Color Sensor

The he DCS230 senses the color of the light utilizing an 8 x 8 arrangement of photodiodes. Then, using a current or frequency converter the readings from the photodiodes are converted into a square wave with a frequency directly proportional to the light intensity. Finally, using the Arduino board we can read the square we found and get the results for the color. The photodiodes, shown in Figure have 3 different color filters; sixteen of them have red filters, sixteen have green filters, sixteen more have blue filters, and the other sixteen are clear filters. Each color that the photodiode reads, depends on the voltage written to the two control pins; S2 and S3. This can be seen in Table 1. [4] Figures 25 and 26 show an image of the color sensor and a zoomed in representation of the photodiodes while Figure 27 shows the color sensor mounted onto the animatronic structure.

Table 1: Control Pin Color Readings

S2	S3	Color Read
LOW	LOW	Red
LOW	HIGH	Blue
HIGH	HIGH	Green

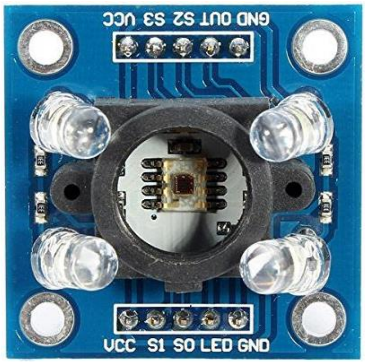


Figure 22: HC-SR502 PIR Sensor

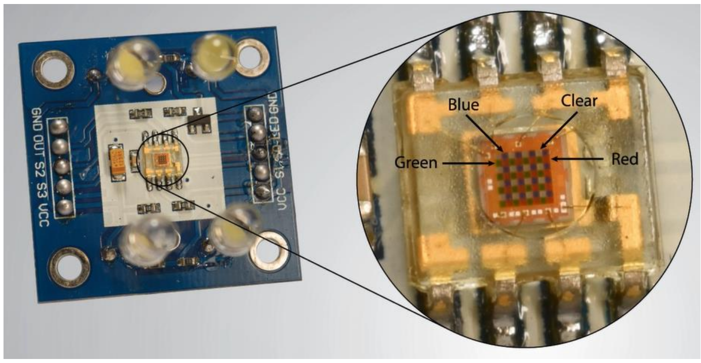


Figure 22: HC-SR502 PIR Sensor

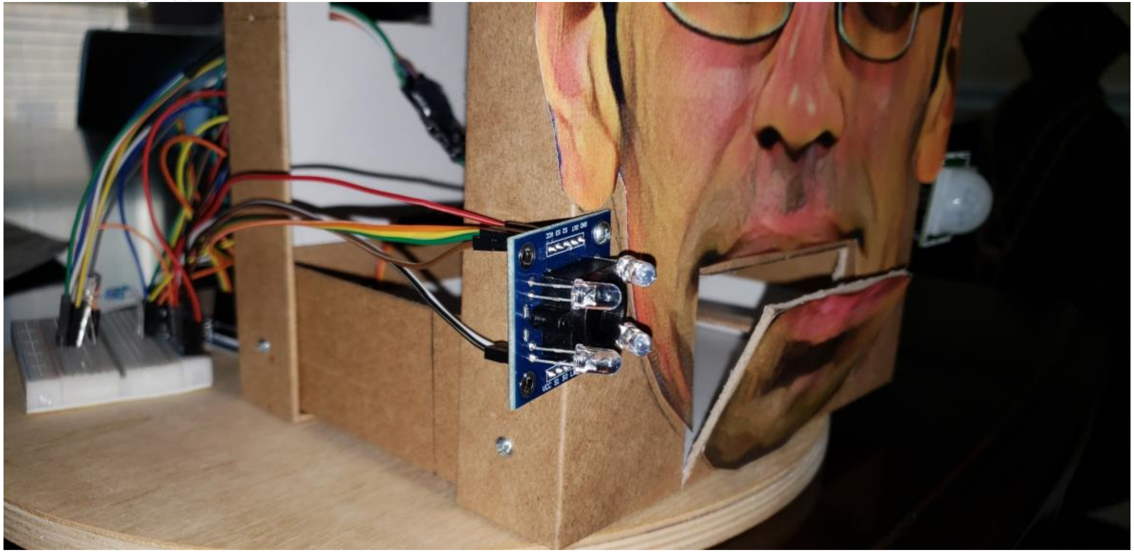


Figure 23: Color Sensor Attached to the Facial Structure

SECTION 6: PROGRAMMING FOR INTERACTION

This animatronic would be initialized from the moment power were to be supplied to it. The logic behind the programming would consists in the following manner:

1. Declaration of all coding libraries and global variables for all the components to work. I.e.; Servos, MP3 Shield, Color Sensor, and RGB LEDs.
2. Setup loop would turn on the serial monitor, attach the servos to desired pins, initialize the music player, declare input and output pins, and play the introductory welcome message.
3. The void loop is dictated by one trigger, the reading of the PIR Sensor. If the sensor is HIGH, meaning it detects movement, it will move to the next part, if not it will simply wait idly until it does.
4. Once the PIR Sensor is high then the music player will play the instruction message asking the user for input using an if statement.
5. Another if statement will initialize the color sensor read values whenever the instruction track is played. Then by a series of other if statements will determine which value of the RGB light is more predominant and output that response

For the full code and wiring refer to the appropriate sections of the appendices.
For a demonstration of the animatronic please refer to THIS VIDEO.

SECTION 7: LESSONS LEARNT AND SUGGESTIONS

Lessons Learned

1. Some libraries have conflicting variables within them, making the program sometimes impossible of working. For example, at first, an Adafruit Wave Shield was going to be used to play the music but the library for that component conflicted with the Arduino Servo library. The only solution was to use another MP3 Shield.
2. One of the main difficulties of this project was to get the audio to sync with the joint movement. Thanks to some research it was learned that by analyzing the sound waves of the recording and syncing the vowels and silences with the delays of the servos, a successful synchronization was achieved.
3. It was also learned that sometimes when it comes to robotics, simplicity is the key to have a successful product. At first the robot was thought of having 10 joints including the neck and 2 axis eye movement it was soon found that having a simple but efficient animatronic would be better
4. Lastly, it was learned that sometimes the Arduino does not output enough voltage to power up all components. The RGB replication of the color read through the sensor was not fully functional when the MP3 shield was active. The solution to this problem was implement two modes of the animatronic, one with fully functional LED replication and another one without it.

Suggestions

1. Implement a relay that would be able to output more voltage so that the RGB LEDs could replicate the color being read and spoken by the animatronic.
2. Another suggestion or area of improvement would be to calculate if statements

for the animatronic to read and identify the secondary and tertiary colors of the color wheel.

SECTION 8: PERSONNEL AND BILL OF MATERIALS

(a) Personnel

Task	Main Personnel	Secondary personnel
Structure/Chassis	Ramon	-
Joints and Motor Interfacing	Ramon	-
Sensor Interface	Ramon	-
Programming and Overall integration	Ramon	John
Report Writing	Ramon	Brainstorming: John, Toe

(b) Bill of materials

No.	Description	Website/comment	Qty.	Unit $	Total $
1	Arduino MEGA 2560	Provided	1	0	**0**
2	12-in Birch Circle	https://www.lowes.com/pd/12-in-Birch-Circle/1000458197	1	7.98	**7.98**
3	Kraft Frame Box by Celebrate It	https://www.michaels.com/s/MichaelsUS/default/10406318.html?productsource=PDPZ1	2	1.99	**11.96**
4	Hillman #4 x 1/2-in Phillips-Drive Sheet Metal Screws (100-Count)	https://www.lowes.com/pd/Hillman-4-x-1-2-in-Phillips-Drive-Sheet-Metal-Screws-100-Count/3037163?cm_mmc=SCE_PLA-_-ToolsAndHardware-_-Fasteners-_-3037163:Hillman&CAWELAID=&kpid=3037163&CAGPSPN=pla&store_code=2480&k_clickID=go_625706834_34613750110_111132558790_aud-449333924337:pla-78266456166_c_9027973&gclid=Cj0KCQiAxZPgBRCmARIsAOrTHSZd8ri_SE5yPYK31Po0hl8_e3Lxid8hzLMCsmh7JPwgofI2fnNu30QaAtVPEALw_wcB	1	4.74	**16.7**
5	Micro servo SG92R	https://www.adafruit.com/product/169?gclid=Cj0KCQiAoo7gBRDuARIsANeJKUaFpOdhcJ0wu2_v1igKvGvVedf2YBOZGkwoiKdFIoRHYhYgktHRlQ8aAsGcEALw_wcB	3	5.95	**34.55**
6	TCS230 Color Sensor	https://www.amazon.com/gp/product/B06XHL79SQ/ref=oh_aui_detailpage_o04_s00?ie=UTF8&psc=1	1	9.99	**44.54**

7	SR501 Pir Motion Sensor	https://www.amazon.com/DIYmall-HC-SR501-Motion-Infrared-Arduino/dp/B012ZZ4LPM/ref=sr_1_3?ie=UTF8&qid=1543818041&sr=8-3&keywords=pir+sensor+arduino	1	8.89	**53.43**
8	210 Ω Resistor	https://www.amazon.com/Baoblaze-50pcs-Resistors-220ohm-Resistance/dp/B07B9PY36T/ref=sr_1_5?s=electronics&ie=UTF8&qid=1543818081&sr=1-5&keywords=220+ohm+resistor	1	3.53	**56.96**
9	Breadboard and Jumper Wires	https://www.amazon.com/gp/product/B0727X6N9D/ref=oh_aui_detailpage_o08_s00?ie=UTF8&psc=1	1	12	**68.95**
10	RGB LED Diodes	https://www.amazon.com/gp/product/B01C19ENDM/ref=oh_aui_detailpage_o04_s01?ie=UTF8&psc=1	1	8.96	**77.91**
11	X-Mini Speaker	https://www.walmart.com/ip/X-Mini-Bluetooth-Portable-Capsule-Wireless-Speaker-KAI-XAM11-B/26819823?wmlspartner=wlpa&selectedSellerId=638&adid=22222222227018457233&wl0=&wl1=g&wl2=c&wl3=75182716151&wl4=pla-131801149031&wl5=9027973&wl6=&wl7=&wl8=&wl9=pla&wl10=112343685&wl11=online&wl12=26819823&wl13=&veh=sem&gclid=Cj0KCQiAoo7gBRDuARIsANeJKUbgyEBdqYZFC4wlDyvlnXf7hNuSd_KZ_4lVYYNln07PUYSKqBMsTTEaAl1iEALw_wcB	1	15	**92.9**
12	Adafruit Music Maker MP3 Shield	https://www.amazon.com/gp/product/B00SDTE380/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1	1	39.9	**132.8**

The total price for this project excluding the Arduino MEGA was **$132.80**

ACKNOWLEDGEMENTS

I would like to thank Dr. Bhounsule that collaborated with his voice and contacts for the presentation. I would also like to acknowledge the help of the Teacher assistants.

REFERENCES:

[1] Casey Bennet MiRAE (Robot Face) - Project Page, http://www.caseybennett.com/research.html#

[2] Mecha Hitsu Gen2 Product Description, http://animatronicrobotics.com/product/mecha-hitsu-gen2/

[3] Kismet Research Page, http://www.ai.mit.edu/projects/humanoid-robotics-group/kismet/kismet.html

[4] Top Tech Boy Lesson 15 http://www.toptechboy.com/arduino/lesson-15-super-cool-arduino-color-sensor-project/

APPENDIX A: CODE

```
// Welcome I am BSNL-L. Human cyborg Relations. I was created in Dr. Bhonsule's image.
// I can tell you which color you are showing me, I can also make my eyes match that color.
// If you are colorblind I can tell you which color you are showing me. If you are not, I can give you reassurance 😉

//ADAFRUIT MUSIC MAKER SHIELD
#include <SPI.h>
#include <Adafruit_VS1053.h>
#include <SD.h>
// These are the pins used for the breakout example
#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);


//SERVOS
#include <Servo.h>    // This library was included because of conflicting directories with WaveHC.h
Servo mouth;          // create servo object to control a servo
Servo rightEyebrow;   // create servo object to control a servo
Servo leftEyebrow;    // create servo object to control a servo
long randNumEyebrowR;
long randNumEyebrowL;
long randNumMouth;
int pos;
```

```
//IR SENSOR
int pirSensor = 30;
int pirState = LOW;
int val = LOW;

//COLOR SENSOR
int redPin = 49;     //red pin
int greenPin = 51;   //green pin
int bluePin = 53;    //blue pin
int S2 = 37;         //Color Sensor Pin S2 to Arduino Pin 7
int S3 = 39;         //Color Sensor pin S3 to Arduino Pin 8
int outPin = 35;     //Color Sensor OUT to Arduino pin 4
int rColorStrength;
int gColorStrength;
int bColorStrength;
unsigned int pulseWidth;

//MISC
String welcomeMsg = "Welcome to BSN-L! Human Cyborg
Relations";                          //Welcome Message
String descriptionMsg = "Show me a color and I will tell you which color it
is!";            //Description Message



void setup() {
  Serial.begin(9600);                    // Turn on Serial Port

  //MISC
  Serial.println(welcomeMsg);            //Print Program Instructions
  Serial.println(descriptionMsg);        //Print Program Instructions


  //SERVO
  mouth.attach(44);              //attaches servo to pin 4
  rightEyebrow.attach(45);  //attaches servo to pin 5
  leftEyebrow.attach(46);    //attaches servo to pin 6


  //ADAFRUIT MUSIC MAKER SHIELD
  // initialise the music player
  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"));
```

```
musicPlayer.sineTest(0x44, 500);    // Make a tone to indicate VS1053 is working

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

Serial.println("SD OK!");
Serial.println("");
// Set volume for left, right channels. lower numbers == louder volume!
musicPlayer.setVolume(1, 1);

if (! musicPlayer.useInterrupt(VS1053_FILEPLAYER_PIN_INT))
  Serial.println(F("DREQ pin is not an interrupt pin"));
Serial.println("");

musicPlayer.playFullFile("welco001.ogg");
if (! musicPlayer.startPlayingFile("/welco001.mp3")) {
  Serial.println("Could not open file welco001.mp3");
  Serial.println("");
  while (1);
}

if (musicPlayer.startPlayingFile("/welco001.mp3")) {
  Serial.println("");
  Serial.println("Hello. I am Dr Bhonsule. Human Cyborg Relations");
  //Hello
  rightEyebrow.write(45);
  leftEyebrow.write(45);
  delay(221);
  mouth.write(0);
  delay(99);
  mouth.write(120);
  delay(24);
  mouth.write(0);
  delay(300);
  mouth.write(120);
  delay(700);
  // I'm
  mouth.write(0);
  delay(210);
  mouth.write(120);
  delay(30);
  // Doctor
  mouth.write(0);
  delay(150);
```

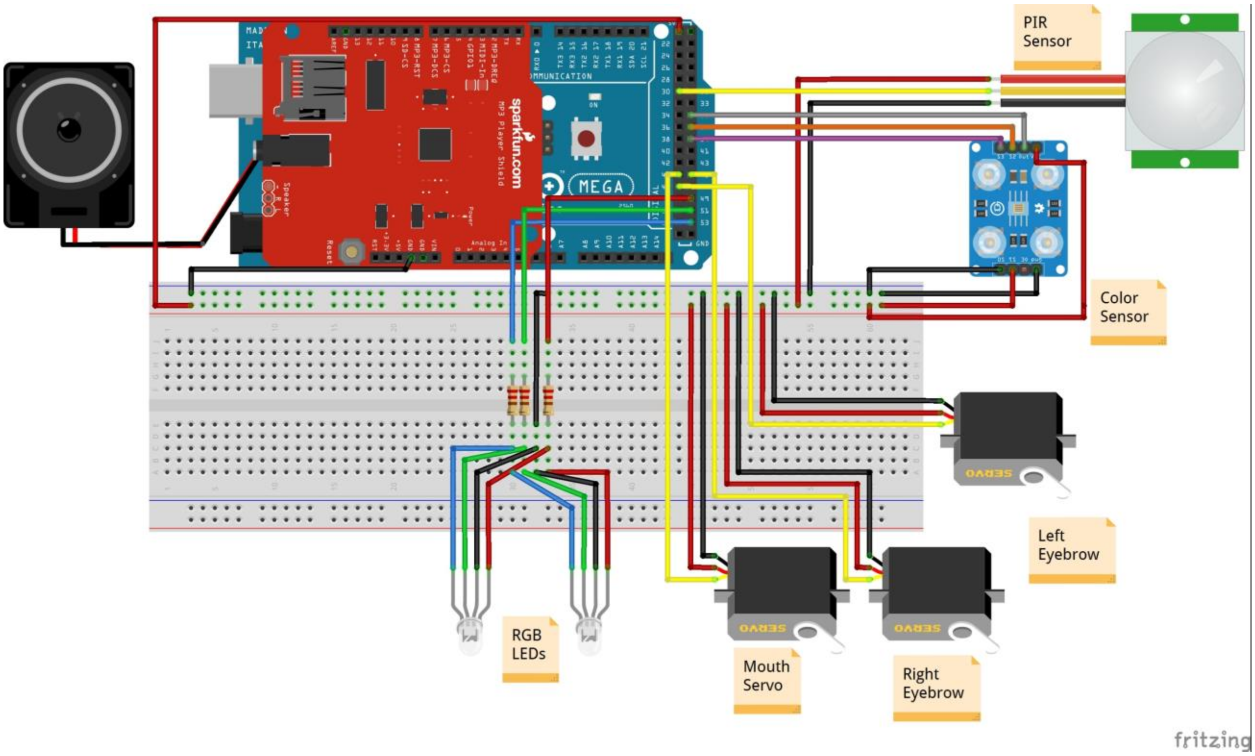
```
mouth.write(120);
delay(40);
mouth.write(0);
delay(111);
mouth.write(120);
delay(40);
// Bhonsule.
mouth.write(0);
delay(100);
mouth.write(120);
delay(100);
mouth.write(0);
delay(274);
mouth.write(120);
delay(100);
// Human
mouth.write(0);
delay(153);
mouth.write(120);
delay(76);
mouth.write(0);
delay(153);
mouth.write(120);
delay(170);
// Cyborg
mouth.write(0);
delay(123);
mouth.write(120);
delay(76);
mouth.write(0);
delay(123);
mouth.write(120);
delay(76);
// Relations
mouth.write(0);
delay(82);
mouth.write(120);
delay(82);
mouth.write(0);
delay(82);
mouth.write(120);
delay(82);
mouth.write(0);
delay(82);
mouth.write(120);
delay(82);
```

```
    delay(3000);
  }

  //IR SENSOR
  pinMode(pirSensor, INPUT);

  //COLOR SENSOR
  pinMode(redPin, OUTPUT);
  pinMode(greenPin, OUTPUT);
  pinMode(bluePin, OUTPUT);

  pinMode(S2, OUTPUT);
  pinMode(S3, OUTPUT);
  pinMode(outPin, INPUT);

}


void loop() {                    // put your main code here, to run repeatedly:

  //IR SENSOR
  val = digitalRead(pirSensor);

  if (val == HIGH) {
    Serial.println(val);
    musicPlayer.playFullFile("show001.ogg");
    if (! musicPlayer.startPlayingFile("/show001.mp3")) {
      Serial.println("Could not open file show001.mp3");
      while (1);
    }
    Serial.println(F("Started playing"));
  }

  if (musicPlayer.startPlayingFile("/show001.mp3")) {
    Serial.println("");
    Serial.println("Hello. Show me a color!");
    Serial.println("moving mouth");
    Serial.println("moving eyebrows");
    //Hello
    rightEyebrow.write(90);
    leftEyebrow.write(0);
    delay(212);
    mouth.write(0);
    delay(64);
    mouth.write(120);
    delay(24);
  mouth.write(0);
  delay(212);
  mouth.write(120);
  delay(512);
  //Show
  mouth.write(0);
  delay(177);
  mouth.write(120);
  //me
  mouth.write(0);
  delay(112);
  mouth.write(120);
  //A
  mouth.write(0);
  delay(141);
  mouth.write(120);
  delay(24);
  //Color
  mouth.write(0);
  delay(115);
  mouth.write(120);
  mouth.write(0);
  delay(224);
  mouth.write(120);
  rightEyebrow.write(45);
  leftEyebrow.write(45);
  delay(305);
  delay(3000);
}

//Color Sensor
if (val == HIGH) {
  // Reading Red component of color
  //S2 and S3 should be set to LOW
  digitalWrite(S2, LOW);
  digitalWrite(S3, LOW);
  pulseWidth = pulseIn(outPin, LOW);
  rColorStrength = pulseWidth / 400. - 1;
  rColorStrength = (255 - rColorStrength);

  // Reading Green component of color
  //S2 and S3 should be set to HIGH
  digitalWrite(S2, HIGH);
  digitalWrite(S3, HIGH);
  pulseWidth = pulseIn(outPin, LOW);
  gColorStrength = pulseWidth / 400. - 1;
gColorStrength = (255 - gColorStrength);

// Reading Blue component of color
//S2 and S3 should be set to LOW and HIGH respectively
digitalWrite(S2, LOW);
digitalWrite(S3, HIGH);
pulseWidth = pulseIn(outPin, LOW);
bColorStrength = pulseWidth / 400. - 1;
bColorStrength = (255 - bColorStrength);

Serial.println("Test");
Serial.print(rColorStrength);
Serial.print(", ");
Serial.print(gColorStrength);
Serial.print(", ");
Serial.println(bColorStrength);

//RGB & RBG
if (rColorStrength > gColorStrength && gColorStrength > bColorStrength) {
  rColorStrength = 255;
  gColorStrength = 0;
  bColorStrength = 0;
  Serial.println("RED");
  musicPlayer.playFullFile("red001.ogg");
  if (! musicPlayer.startPlayingFile("/red001.mp3")) {
    Serial.println("Could not open file red001.mp3");
    while (1);
  }
  Serial.println(F("Started playing RED"));
  mouth.write(0);
  delay(250);
  mouth.write(120);
  delay(250);
}
if (rColorStrength > bColorStrength && bColorStrength > gColorStrength) {
  rColorStrength = 255;
  gColorStrength = 0;
  bColorStrength = 0;
  Serial.println("RED");
  musicPlayer.playFullFile("red001.ogg");
  if (! musicPlayer.startPlayingFile("/red001.mp3")) {
    Serial.println("Could not open file red001.mp3");
    while (1);
  }
  Serial.println(F("Started playing RED"));
  mouth.write(0);
```

```
    delay(250);
    mouth.write(120);
    delay(250);
  }

  //GRB & GBR
  if (gColorStrength > rColorStrength && rColorStrength > bColorStrength) {
    rColorStrength = 0;
    gColorStrength = 255;
    bColorStrength = 0;
    Serial.println("GREEN");
    musicPlayer.playFullFile("green001.ogg");
    if (! musicPlayer.startPlayingFile("/green001.mp3")) {
      Serial.println("Could not open file red001.mp3");
      while (1);
    }
    Serial.println(F("Started playing GREEN"));
    mouth.write(0);
    delay(250);
    mouth.write(120);
    delay(250);
  }
  if (gColorStrength > bColorStrength && bColorStrength > rColorStrength) {
    rColorStrength = 0;
    gColorStrength = 255;
    bColorStrength = 0;
    Serial.println("GREEN");
    musicPlayer.playFullFile("green001.ogg");
    if (! musicPlayer.startPlayingFile("/green001.mp3")) {
      Serial.println("Could not open file red001.mp3");
      while (1);
    }
    Serial.println(F("Started playing GREEN"));
    mouth.write(0);
    delay(250);
    mouth.write(120);
    delay(250);
  }

  //BRG & BGR
  if (bColorStrength > rColorStrength && rColorStrength > gColorStrength) {
    rColorStrength = 0;
    gColorStrength = 0;
    bColorStrength = 255;
    Serial.println("BLUE");
    musicPlayer.playFullFile("blue001.ogg");
```

```
      if (! musicPlayer.startPlayingFile("/blue001.mp3")) {
        Serial.println("Could not open file blue001.mp3");
        while (1);
      }
      Serial.println(F("Started playing BLUE"));
      mouth.write(0);
      delay(250);
      mouth.write(120);
      delay(250);
    }
    if (bColorStrength > gColorStrength && gColorStrength > rColorStrength) {
      rColorStrength = 0;
      gColorStrength = 0;
      bColorStrength = 255;
      Serial.println("BLUE");
      musicPlayer.playFullFile("blue001.ogg");
      if (! musicPlayer.startPlayingFile("/blue001.mp3")) {
        Serial.println("Could not open file blue001.mp3");
        while (1);
      }
      Serial.println(F("Started playing BLUE"));
      mouth.write(0);
      delay(250);
      mouth.write(120);
      delay(250);
    }

    Serial.println("Augmented value");
    Serial.print(rColorStrength);
    Serial.print(" , ");
    Serial.print(gColorStrength);
    Serial.print(" , ");
    Serial.print(bColorStrength);
    Serial.println("");
    Serial.println("");
    analogWrite(redPin, rColorStrength);
    analogWrite(greenPin, gColorStrength);
    analogWrite(bluePin, bColorStrength);
    delay(1000);
  }
  if (val == LOW) {
    Serial.println("Make arduino say OK. Goodbye!");
    //    vcc= LOW;

  }
}
```

APPENDIX B: ARDUINO WIRING


PIR Sensor
Color Sensor
Left Eyebrow
RGB LEDs
Mouth Servo
Right Eyebrow
MEGA
sparkfun.com
MP3 Player Shield
SERVO
fritzing