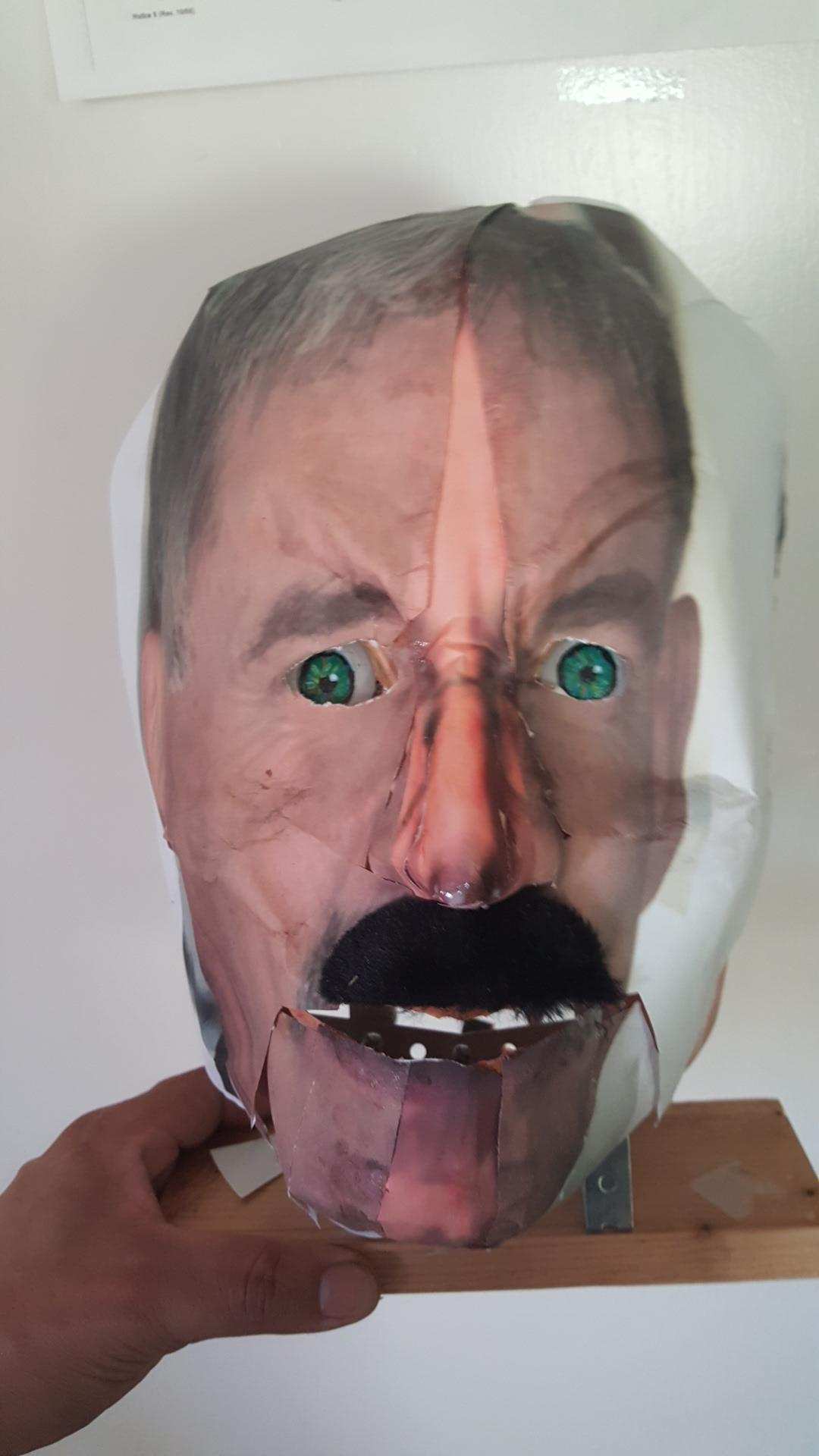
ANIMATRONICS FACE: PROfessor Manteufel



Dylan Martin

Dept. of Mechanical Engineering

San Antonio, TX, USA 78249

dylanjames1986@gmail.com

Martin Guerrero

Dept. of Mechanical Engineering

San Antonio, TX, USA 78249

Gmartin\_2006@hotmail.com

Nicholas Comeau

Dept. of Mechanical Engineering

San Antonio, TX, USA 78249

Nicomeau1@gmail.com

Submitted as part of project for ME4543 Mechatronics, Fall 2018

TABLE OF CONTENTS

|  |  |
| --- | --- |
| Abstract | 3 |
| Literature Review | 4 |
| Brainstorming | 6 |
| Supporting Structure | 6 |
| Joints and Motors | 7 |
| Sensors | 9 |
| Programming | 9 |
| Lesson learnt and suggestions | 9 |
| Personnel and Bill of Materials | 10 |
| References | 11 |
| Appendix A: Arduino Code | 12 |

**ABSTRACT**

The group designed and constructed an animatronic face based of the voice and features of UTSA professor, Dr. Randall Manteufel. The group used two motors and a supporting structure made of wood and metal brackets to construct the face. An Arduino was used to control the face, that is, make its mouth and eyes move in conjecture with the recorded voice of Dr. Manteufel. A motion sensor was used to trigger the mouth and speech, while a button sensor was used to cause the eyes of the face to move. The group used an existing mask and overlaid a picture of Dr. Manteufel’s face onto the mask in order to capture the resemblance and features of the professor.

**Section 1: Literature review**

**Cardboard Animatronic**

The animatronic face was predominately made out of cardboard.1 The face was held together with hot glue to stick the cardboard and the motor to the face, and screws for the moveable parts, such as the eyebrows and the jaws. Two DC motors were used in order to provide the motion for the face. One DC motor was used to move the jaw of the face, and a second one was used to move the eyebrows of the face. No sensors were used in this model, but sensors could be easily incorporated into the model. For example, an ultrasonic sensor could be used to trigger the mouth movements and speech upon hearing sound. The purpose of this face was for entertainment.

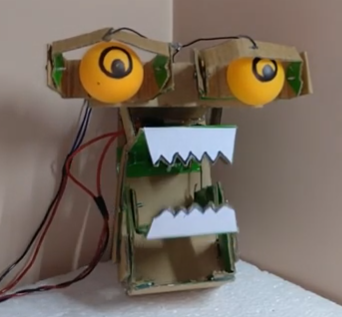


Figure 1 – Cardboard Animatronic Face

**Tyrannosaurus Rex from Jurassic Park (1993)**

When Jurassic Park first hit theatres in the year 1993, it was a ground-breaking film in more than one way. Not only was it a box office smash when released, it is still one of the top grossing films of all time, but it was a landmark case for the marriage of technology and film. The most spectacular aspect of the visual effects was not the way ahead of its time level of CGI realism, but rather the practical effects that visual effects icon Stan Winston and his team were able to produce. Although scenes involving a fully digital T-Rex were used throughout the movie, a life-size fully robotic king of lizards was constructed specifically for the film.2 The T-rex was initially designed by Disney Imagineer Bob Gurr, and its frame was constructed primarily of high-grade steel. The skin was constructed using film-quality latex and rubber, with polymer fittings and hand-painted to provide high details. The dynamic aspects of the Rex were powered using several hydraulic actuators and required four controllers to operate simultaneously. The controllers were able to move and rotate the head, and at the same time open and close the lower jaw, which provided that classic terrifying roar. The purpose of this animatronic was to provide a realism and bring a tangible sense of magnitude to the film, more so than a purely CGI model would.

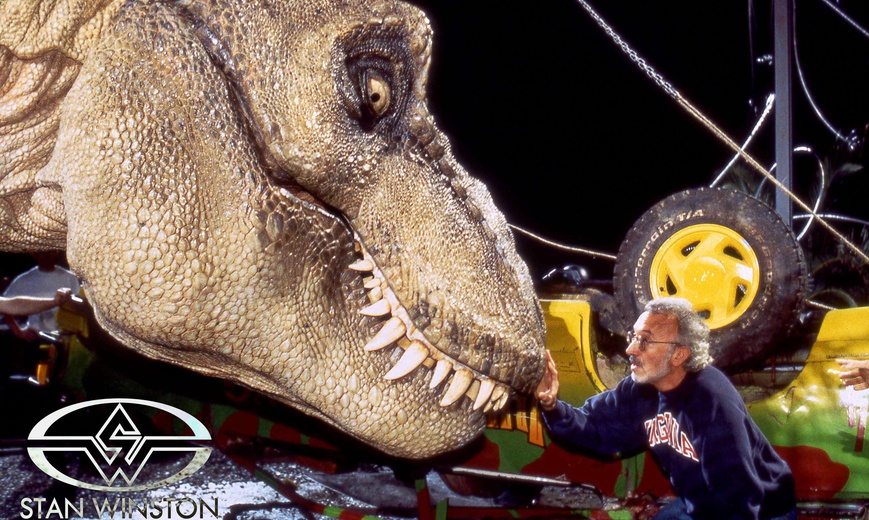


Figure 2 – Jurassic Park Animatronic Dinosaur

**Animatronics Puppet Head**

This puppet head was created to imitate mouth movements by being controlled by a person’s voice without having to use the hand input inside the puppets head.3 The mechanism used was very simple. It consisted of a pinball solenoid and a metal rod attached to the 3D printed jaw of the puppet. To trigger the solenoid, a microphone was used to take the voice inputs, 2 op-amps to increase the waveform of microphone and a 3rd op-amp to turn wave into digital signal. The creator of this head also used a multi-vibrator timer along with a 3 Mohm resistance and a 555 timer to help control the pulses and adjust the jaw’s power. A few more components were used to complete the electronics part of this project, like an AND gate and an IRL530 logic level MOSFET. All in all, the puppet will move its mouth with the sound of a person’s voice. A more detail explanation can be found in the link below.3



Figure 3 – Animatronic Puppet

**Section 2: Brainstorming (initial planning)**

The mechatronic face was meant to move its mouth and move its eyes. This will be accomplished by using two servos mounted to the supporting structure. The first servo will move the eyes via two wooden sticks placed in the eyes and attached to a popsicle stick whose center is placed on the motor. The mouth will be attached to another popsicle stick, which runs to the second motor. The structure for the face will consist of metal brackets to hold everything in place. The face of Dr. Manteufel will be printed out on paper and overlaid on a plastic mask in order to get the structure of the face. The voice of Dr. Manteufel will be downloaded from his existing YouTube videos and synced into the mouth movements of the mechatronic face. The Arduino and motors will be powered through the use of a wall outlet, if available, or via laptop.

**Section 3: Supporting structure**

The supporting structure was created using 8 8” and 8 6” metal brackets. The brackets were mounted to a plank of wood, in order to ensure a secure reference point. The brackets were held in place using nuts and screws of the appropriate size to fit in the holes.



Figure 4 – Basic Structure for Face

After the original frame was constructed, the supports required to hold the motors in place were fabricated. By adding the additional structure, the motors and bread board could be placed at the effective same height as the objects they needed to move, in order to ensure successful implementation of the motors.



Figure 5 – Finished Structure with Mask

**Section 4: Joints and motors**

The group used a mouth joint and eye joints to complete the project. The group accomplished this by using two 9g servo motors attached with popsicle sticks. The popsicle stick for the mouth was then attached to the chassis for the mouth, which allowed the mouth to move up and down.

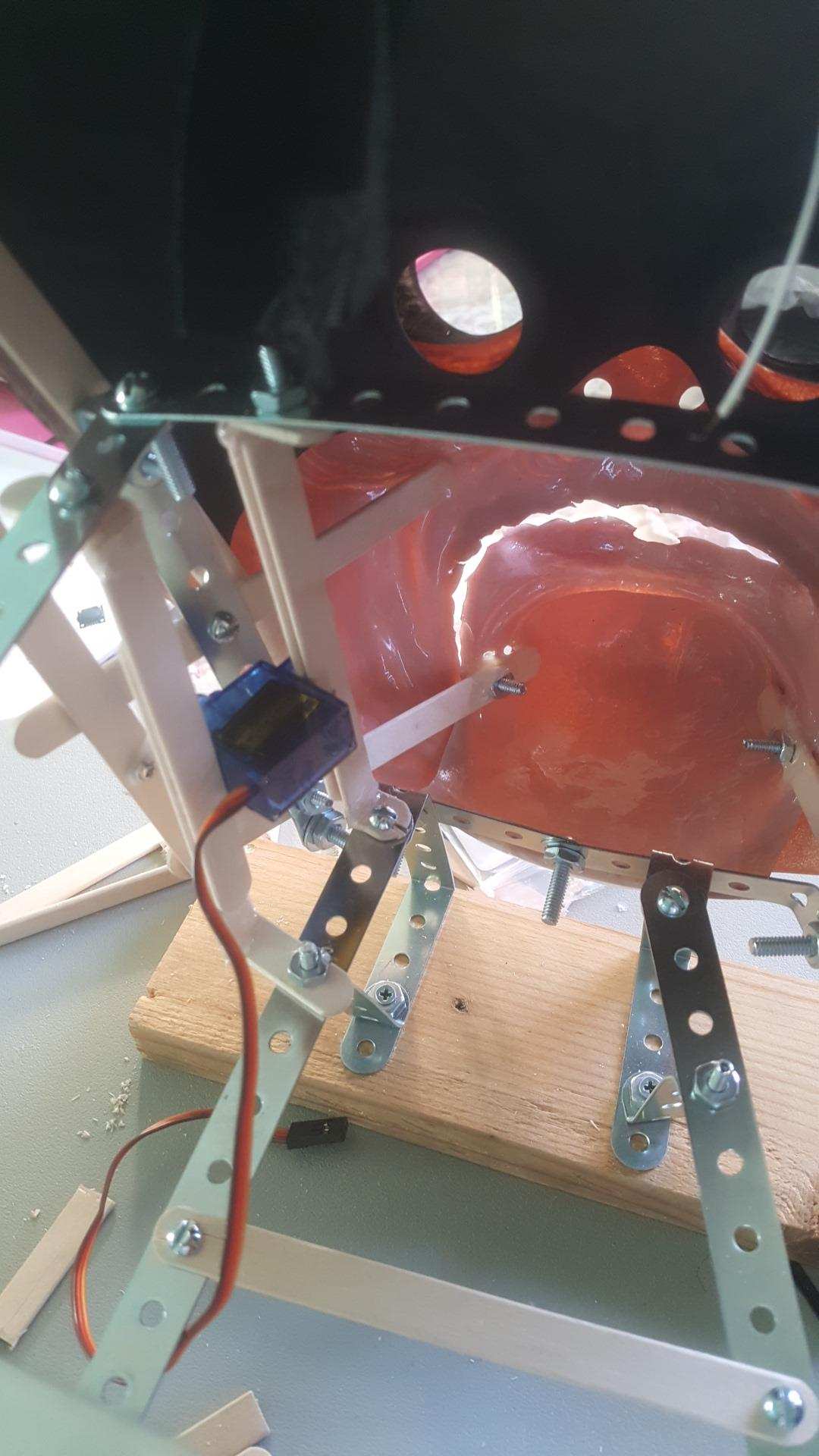


Figure 6 – Closeup of Mouth Servo

The popsicle stick was centered on the motor for symmetry. By placing the center of the stick on the motor, and the two thinner sticks equidistant on the sides, we could assure the eyes moved the same distance, simultaneously. Two ping pong balls were attached to the two thin sticks to serve as the eyes of the face.

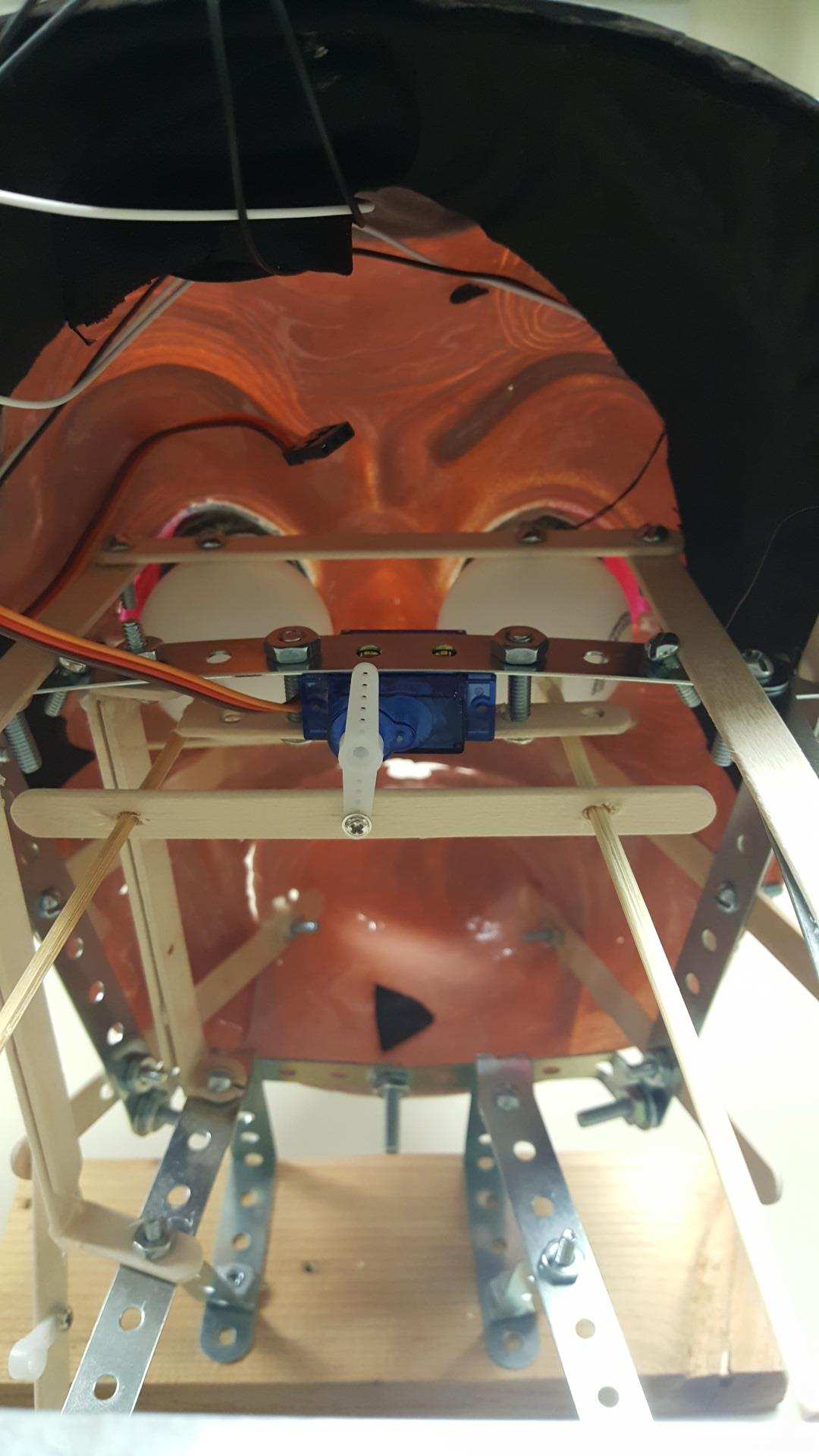


Figure 7 – Closeup of Eye Servo

These motors were chosen due to their inexpensive price, availability, and the fact that they can perform the tasks required for the project.

**Section 5: Sensors**

The two sensors chosen for this project were an infrared motion sensor and a button sensor. The group decided on a motion sensor to activate the mouth. The purpose of this decision was to have the face say something to whomever approached it. The sensor was mounted on the back of the supporting structure in order to allow the most possible visibility to detect the user.

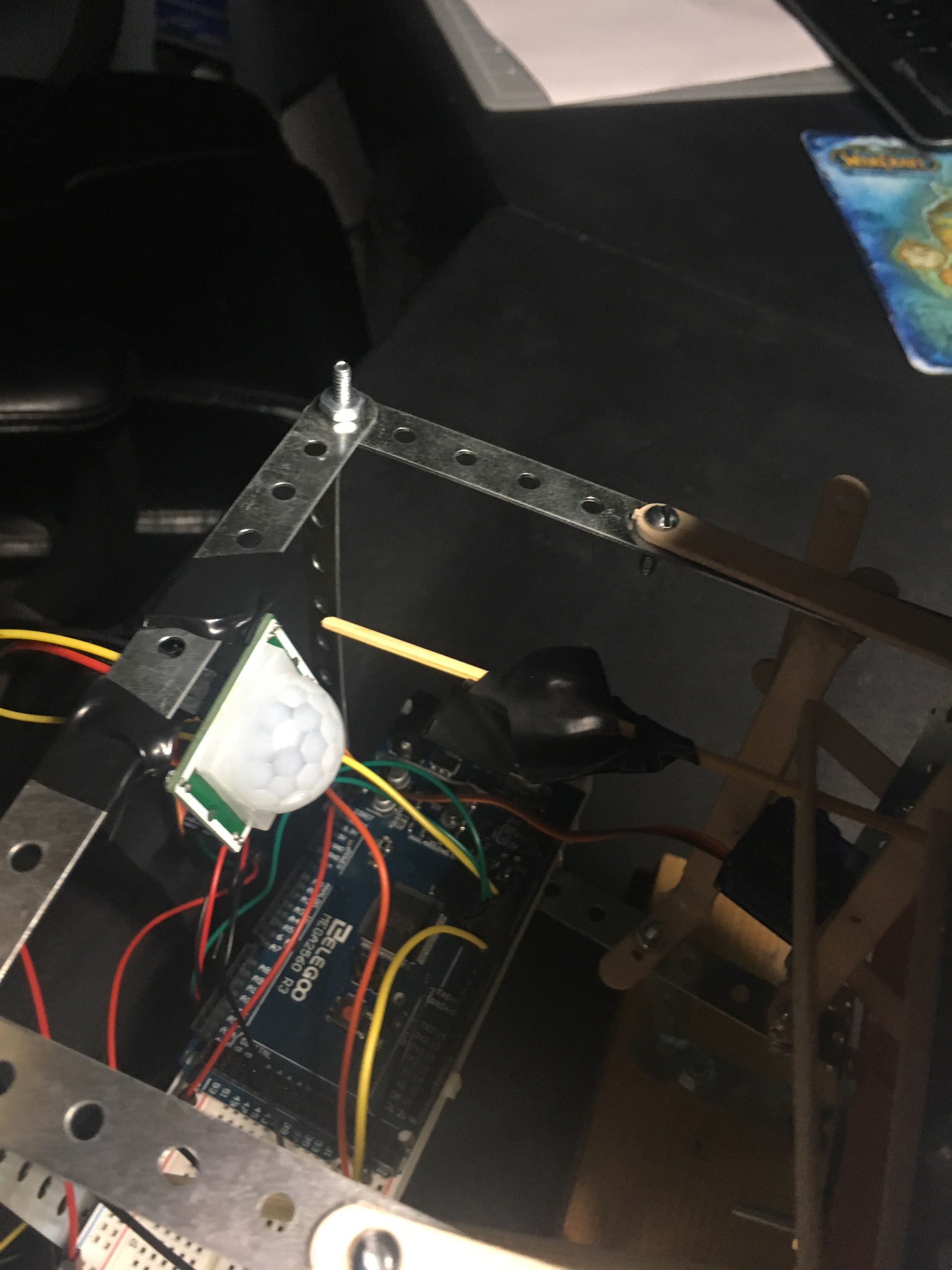


Figure 8 – Closeup of Motion Sensor

A button sensor was attached to the Arduino and placed in front of the display in order to allow the user easy interaction with it. The button triggered the eye movement of the face. The group chose to use a button sensor because it is easy for a passerby to interact with it.

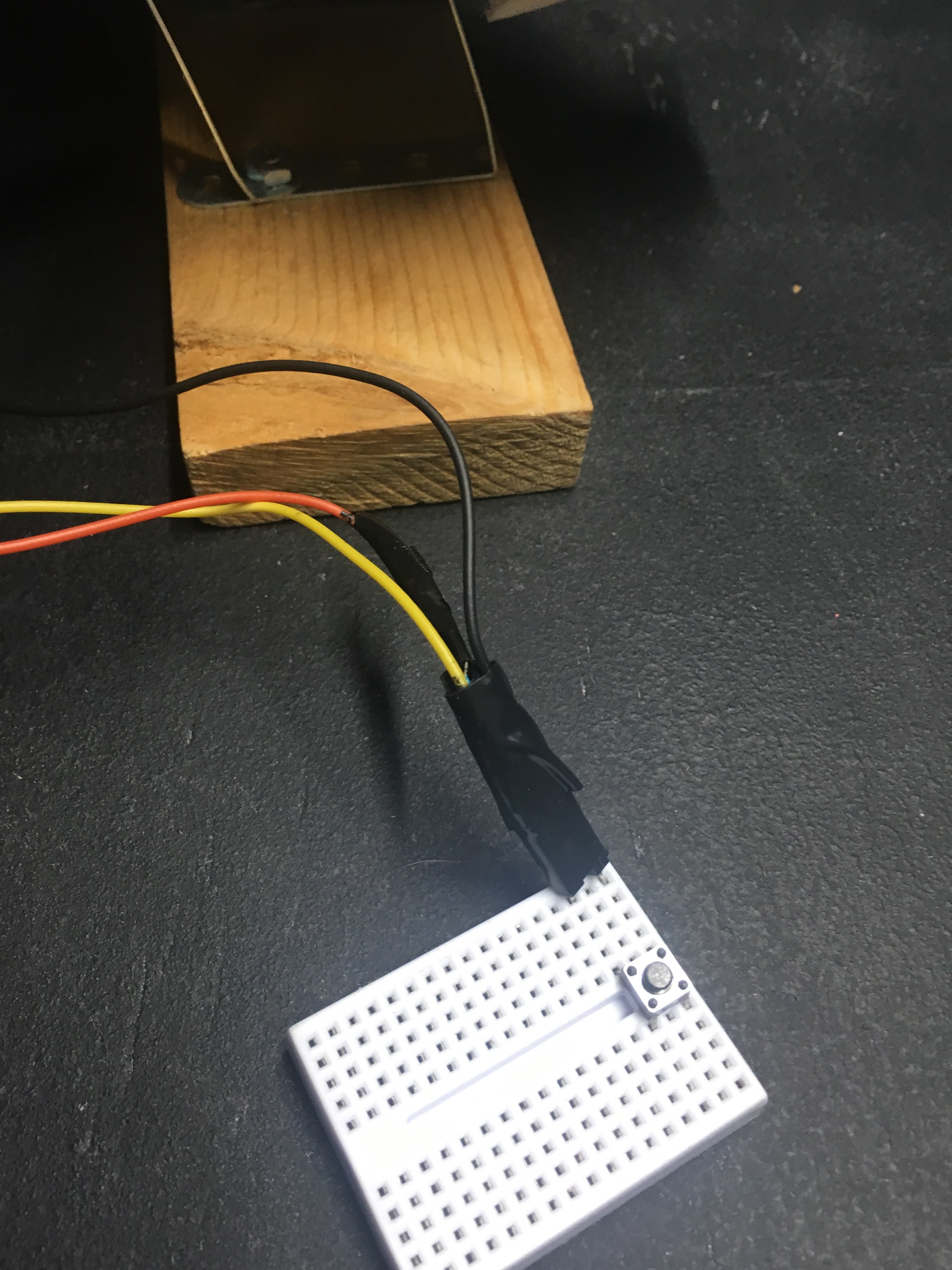


Figure 9 – Closeup of Button Sensor

**Section 6: Programming for interaction**

The program was intended to grab the user’s attention with the motion sensor. The user would walk by, and notice the mouth started to move due to their motion. The user could then come up to the table and interact with the face more by triggering the motion sensor again, or by pushing a button to get the eyes to look at them. The exact program and the logic behind each individual command can be found in the appendix. The video for the face in its completion can be found in the references as well.4

**Section 7: Lessons learned and suggestions**

Lessons

1. Arduino programming can be more time consuming and challenging than originally anticipated. But Dylan put forth the effort and made an exquisite program for the project.
2. Material components do not always behave as originally designed. Martin had to make various adjustments when the joints did not act as anticipated in conjunction with the motor, but he succeeded in his efforts to correct the issue.

Changes

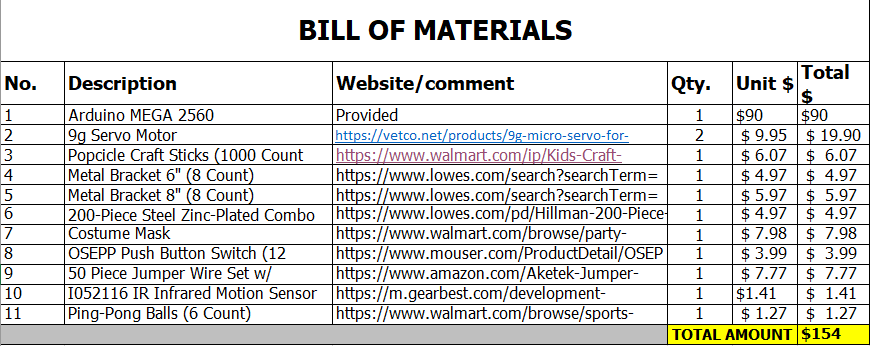
1. The group would have like to include additional eye movements in the interaction. The one interaction was fun, but mixing it up a bit would have been more entertaining for the casual observer.
2. The group would also have liked to sync the sound and the mouth movements a little better. Having the mouth open slightly longer for longer sounds. And mimicking the distance the mouth opens for different vowels and consonants.
3. I higher quality of picture or overlay job for the print out of the face onto the mask would also have been nice.

**Section 8: Personnel and bill of materials**

**(a) Personnel**

|  |  |  |
| --- | --- | --- |
| Task | Main Personnel | Secondary personnel |
| Structure/ Chassis Design | Martin | Dylan |
| Creating Joints and Motor Interface | Martin | Dylan and Nicholas |
| Integrating Sensors | Dylan | Martin and Nicholas |
| Overall Programming and Integration | Dylan | Martin and Nicholas |
| Report Writing and Assembly | Nicholas | Martin and Dylan |

**(b) Bill of materials**



**References:**

[1] 2018, “How to make a animatronics robot face", YouTube [Online]. Available: https://www.youtube.com/watch?v=ycE\_IDdh5NA. [Accessed: 21- Oct- 2018].

[2] Stan Winston School of Character Arts [Online]. Available: https://www.stanwinstonschool.com/blog/jurassic-park-t-rex-robot-almost-eats-crewmember. [Accessed: 21-Oct-2018].

[3] thebenheckshow, 2015, “Ben Heck's Animatronic Puppet Head Part 2,” YouTube [Online]. Available: https://www.youtube.com/watch?v=NeCQA9zGOwA. [Accessed: 21-Oct-2018].

[4] Martin, D., 2018, "UTSA Mechatronics Fall 2018", YouTube [Online]. Available: https://youtu.be/TGnB-Dgxkx8. [Accessed: 05- Dec- 2018].

**Appendix A: Arduino Code**

#include <Servo.h>

Servo myservo1; //creates the first servo object

int pos = 0; //initial servo variable

int calibrationTime = 10; //amount of time we give the sensor to calibrate(10-60 secs according to the datasheet)

boolean lockLow = true;

boolean takeLowTime;

int pirPin = 7; //digital pin connected to the PIR's output

int pirPos = 13; //connects to the PIR's 5V pin

const int buttonPin = 4; // pushbutton pin

const int servoPin = 6; //Servo Pin

Servo servo; //creates 2nd servo motor

int counter = 0; //initial servo posistion

void setup(){

myservo1.attach(5); //attaches servo to pin 4

Serial.begin(9600); //begins serial communication

pinMode(pirPin, INPUT);

pinMode(pirPos, OUTPUT);

digitalWrite(pirPos, HIGH);

Serial.println("calibrating sensor ");

for(int i = 0; i < calibrationTime; i++){

Serial.print(calibrationTime - i);

Serial.print("-");

delay(1000);

servo.attach (servoPin);

pinMode(buttonPin, INPUT);

Serial.println();

Serial.println("done");

while (digitalRead(pirPin) == HIGH) {

delay(500);

Serial.print(".");

}

Serial.print("SENSOR ACTIVE");

}

void loop(){

if(digitalRead(pirPin) == HIGH){ //if the PIR output is HIGH, turn servo

for(pos = 0; pos < 30; pos += 1) //goes from 0 to 180 degrees

{ //in steps of one degree

myservo1.write(pos); //tells servo to go to position in variable "pos"

delay(5); //waits for the servo to reach the position

}

for(pos = 30; pos>=1; pos-=1) //goes from 180 to 0 degrees

{

myservo1.write(pos); //to make the servo go faster, decrease the time in delays for

delay(5); //to make it go slower, increase the number.

}

if(lockLow){

//makes sure we wait for a transition to LOW before further output is made

lockLow = false;

Serial.println("---");

Serial.print("motion detected at ");

Serial.print(millis()/1000);

Serial.println(" sec");

delay(50);

}

takeLowTime = true;

}

if(digitalRead(pirPin) == LOW){

if(takeLowTime){

lowIn = millis(); //save the time of the transition from HIGH to LOW

takeLowTime = false; //make sure this is only done at the start of a LOW phase

}

//if the sensor is low for more than the given pause,

//we can assume the motion has stopped

if(!lockLow && millis() - lowIn > pause){

//makes sure this block of code is only executed again after

//a new motion sequence has been detected

lockLow = true;

Serial.print("motion ended at "); //output

Serial.print((millis() - pause)/1000);

Serial.println(" sec");

delay(50);

}

}

// local variable to hold the pushbutton states

int buttonState;

//read the digital state of buttonPin with digitalRead() function and store the //value in buttonState variable

buttonState = digitalRead(buttonPin);

//if the button is pressed increment counter and wait a tiny bit to give us some //time to release the button

if (buttonState == LOW) // light the LED

{

counter++;

delay(150);

}

if(counter == 0)

servo.write (20); // zero degrees

else if(counter == 1)

servo.write(90);

//else reset the counter to 0 which resets thr servo to 0 degrees

else

counter = 0;

}