Bubbles the Clown Animatronic

By Katelyn Gerhardt

1 Description of product:

Since childhood my family has always gone above and beyond for Halloween. From fake blood splattered on the front of the house to life sized clowns sitting on the front lawn, my house was known for putting on an extravagant and gory scene. This has always made me extremely interested in the entertainment field of animatronics. Thus, for my project I have taken an old life sized clown Halloween prop and turned it into Bubbles, the Clown Animatronic. This idea was chosen purely out of curiosity and getting a feel for the animatronic world with the intention of seeking a potential career in the animatronics world for major theme parks which has been a dream since childhood.

Figure 1: Before and after of Bubbles the Clown

2 Demonstration

https://drive.google.com/file/d/1wq3goDjkOHvRbNX7bdmdzc0dBOQO9pZJ/view?usp=sharing
https://youtu.be/dSgdo8eXig4

3 Electromechanical Details:

3.1 Life Size Clown

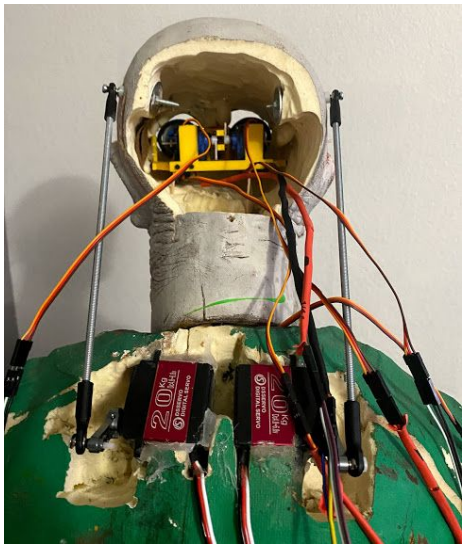
Upon starting the project, the life sized clown was entirely made of styrofoam with no prior electronics. I started from complete scratch and emptied out the entire head to fit in the electronics. After that, I used foam from a can to patch up the holes made from the assembly process. Finally, I used a dremel to shave down the excess foam to make him look more realistic.

3.2 Controls

Movement of Bubbles is controlled completely by a PlayStation controller using the PS4-ESP32 [1] and ESP32Servo [2] libraries. As pictured below in **Figure 2** the controls are as follows: the left joystick controls the X and Y movement of the eyes, the right joystick controls the Y movement of the head, the left trigger controls the blinking of the eyes, and the right trigger controls the movement of the jaw. Each trigger of the controller corresponds to a PWM signal ranging from 0 to 255, which in turn maps to the degrees of the servo arm from 0° to 180°. Similarly, the joysticks correspond to analog signals ranging from -127 to 128 which in turn maps to the degrees of the servo arm from 0° to 180°.

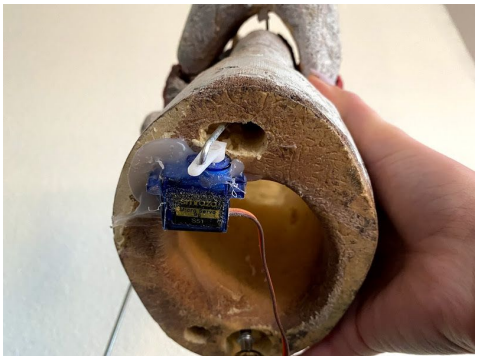
In order to get my microcontroller to communicate with the PS4 controller, I had to find the controller's MAC address which I was able to do using the program "SixAxis Pair Tool" that gave me the controllers MAC address.

Figure 2: PlayStation controls and events



3.3 Head & Shoulders

Originally, the head was mounted on a large spring attached to his torso. I completely removed the spring and replaced it with a ball and socket joint so the head can freely move. Next, I cut holes out of his back to fit two 20KG servo motors. The servos were then attached to a long rod with two servo ball-links on either side. This rod connects the servo motors to the side of his head to allow movement in the Y direction.

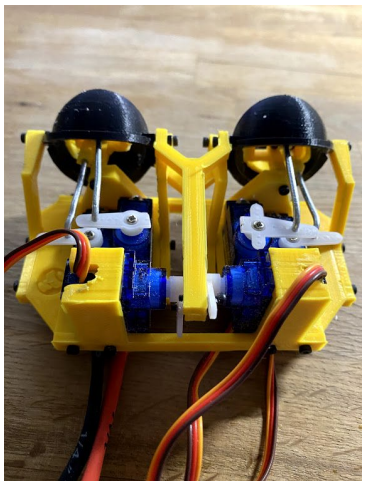


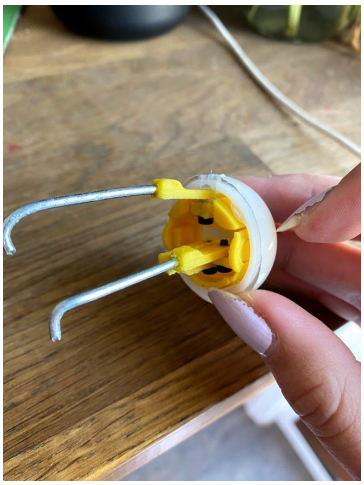
3.4 Jaw

The jaw is controlled by a SG90 Micro Servo attached to a long stiff wire through his neck to the back of his jaw. Originally, the clown's jaw was attached by small springs on either side which I left to allow for easy movement by the servo. The servo was then hot glued down to the bottom of his neck. A hole was extruded through his throat for a long stiff wire to connect to his jaw.

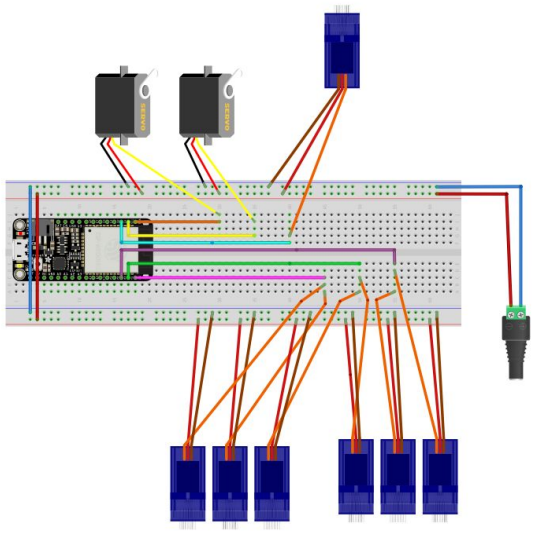
3.5 Eyes

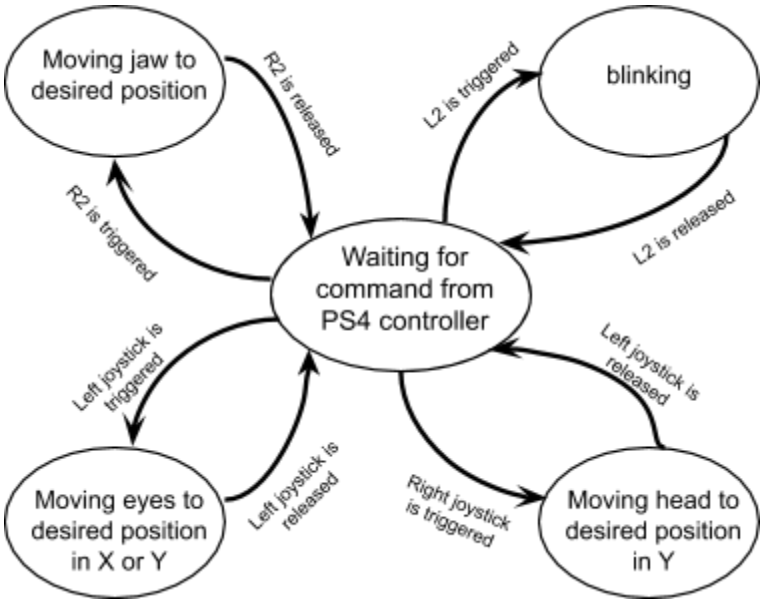
The eye mechanism was designed by Nilhem Electronics [3] and 3D printed at Jacobs Makerspace. This implementation was the most difficult and time consuming. The components of the eyes can be seen in the appendix. The eyes are controlled by six SG90 Micro Servos: two for the X direction, two for the Y direction and two for each eyelid. I had slightly tweaked Nilhem's design and used stiff wire to connect the servo horns to the eye mechanism. The eye mechanism is placed into the actual eye which sits inside the eyelids.





4 Circuit Diagram and Finite State Machine Diagram





5 Description of all Parts

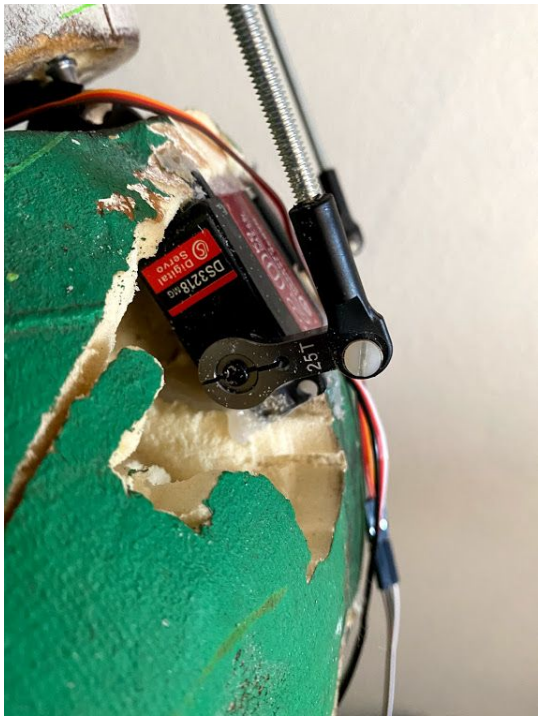
In addition to the parts used from the MicroKit, the following parts were bought (and some bought by my dad) for a total of ~$100 out of pocket.

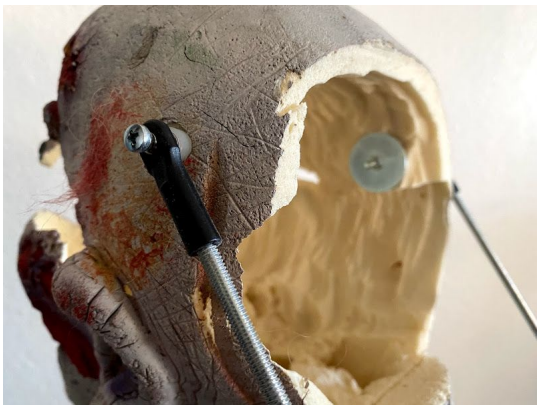
- 20KG Digital Servo x2 (~$16.99 each) Used for head movement.
- SG90 Micro Servo Motor x6 (~$18.59 for 10 pcs) Used for jaw and eye movement. Had to buy two orders for a total of 20 pcs because motors broke unexpectedly.
- Universal Ball and Socket Joint x1 (~$7.99) Used for universal movement of the head.
- Ball Joint Link x4 (~$20 for 5 from local hobby store) Used for smoother movement of servo joints.
- Threaded Rods x2 (~$?? From home depot) Used for attaching the servos in the shoulder to the head. Not sure of the exact price, my dad picked it up for me and forgot the price.
- Stiff Wire x1 (~$?? From home depot) Used to attach the servo to the jaw running through the neck. Also bought my dad from Home depot and forgot the price.
- M3 Screws x40(~$12 from local hobby store) Used to attach 3D printed parts
- PLA 170g (~$20 including shipping from Jacobs Maker Space) Used for printing eye components through the Jacobs Support Program.
- Expanding Foam (~$?? From home depot) Used for filling the holes made from the assembly process. Also bought my dad from Home depot and forgot the price.

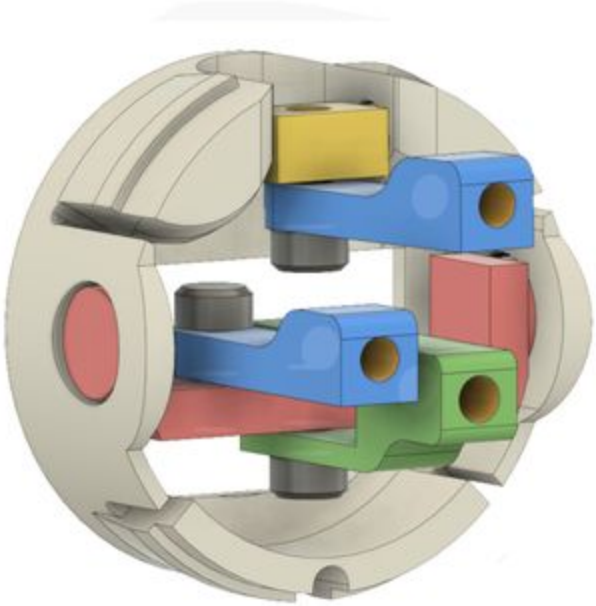
References

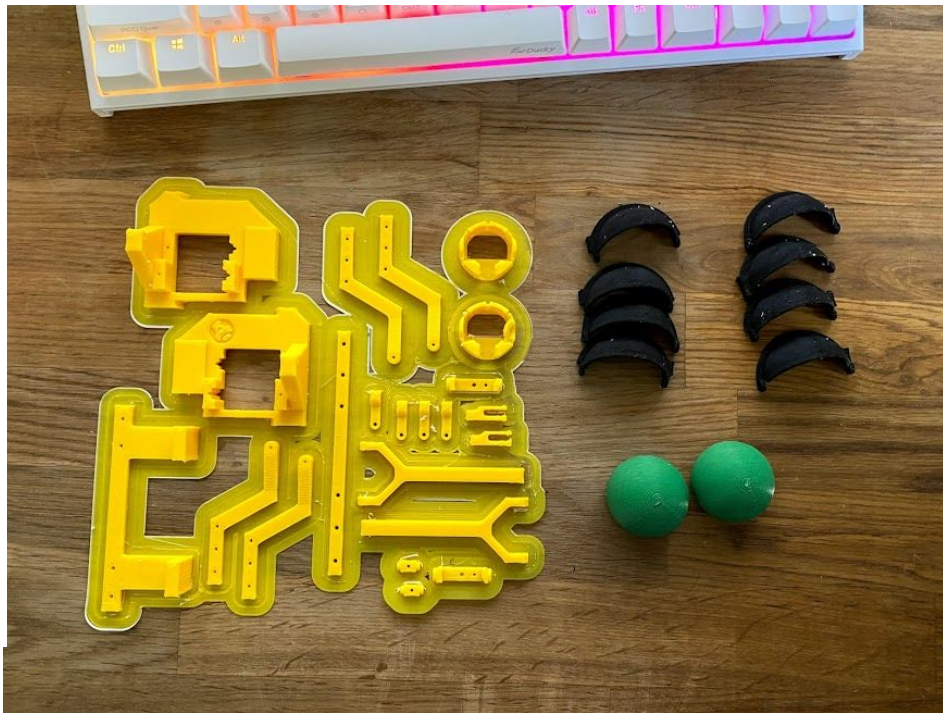
1. https://github.com/aed3/PS4-esp32
2. https://www.arduinolibraries.info/libraries/esp32-servo
3. Cogley, Will. “Compact and Robust 3D Printed Animatronic Eye Mechanism.” *Nilheim Mechatronics*, www.nilheim.co.uk/latest-projects-and-blog/compact-and-robust-3d-printed-animatronic-eye-mechanism

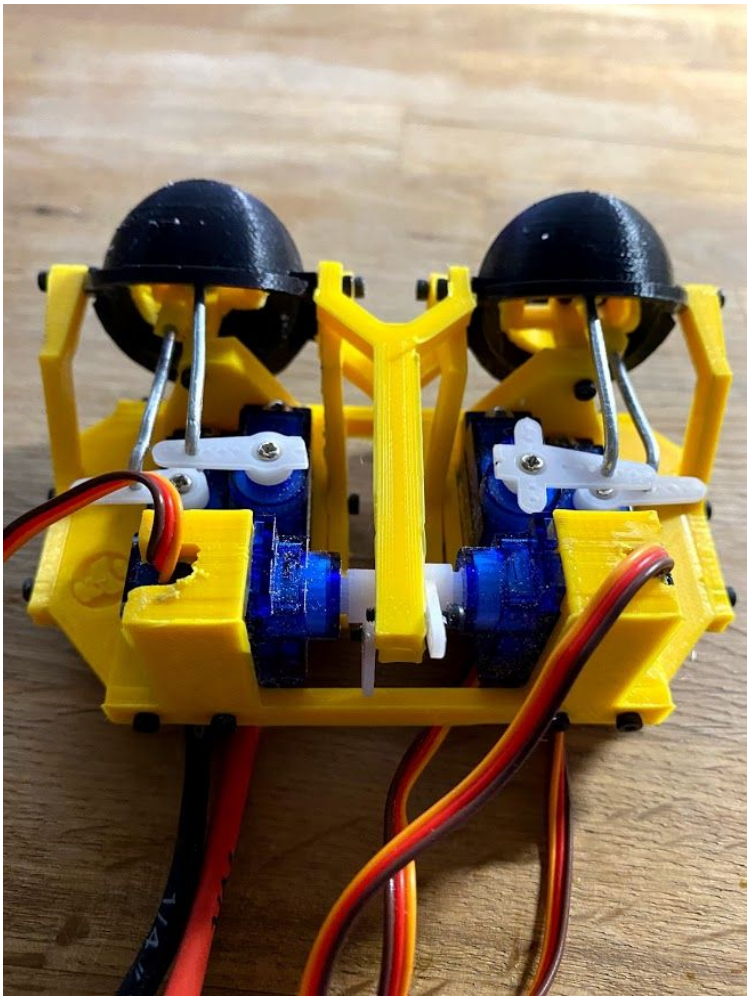
Appendix

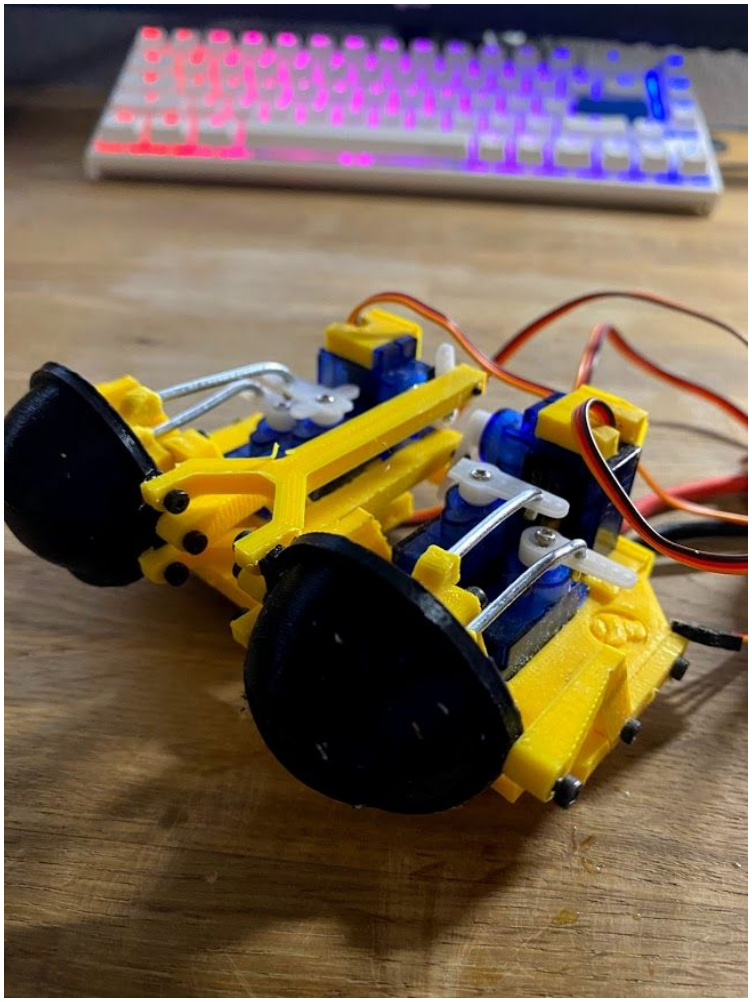












Arduino Code

Katelyn Gerhardt

```
#include <ESP32Servo.h>
#include <PS4Controller.h>
#include <Wire.h>


// Shoulder and Jaw Servos
Servo rightShoulder; //controlled by right joystick
Servo leftShoulder; //controlled by right joystick
Servo jaw; //controlled by R2
int rightShoulderPIN = 23;
int leftShoulderPIN = 22;
int jawPIN = 14;
int shoulderDesY;
int shoulderDesX;
int shoulderCurrY;
int shoulderCurrX;
int R2pos;

// Eye Servos
Servo eyeX; //controlled by left joystick X
Servo eyeYleft; //controlled by left joystick Y
Servo eyeYright;
Servo eyeLids; //controlled by L2
int eyeXPIN = 21;
int eyeYPINright = 17;
int eyeYPINleft = 16;
int eyeLidsPIN = 19;
int L2pos;
int eyeDesX;
int eyeDesYright;
int eyeDesYleft;
int delayy = 1000;

#define SERVO_SPEED 45 //higher value, slower the speed

static unsigned long servo_time;

void eyeloop() {
  while  (!PS4.data.button.circle) {

    positionA();
    blinkEyes();


    positionB();
    positionC();
    blinkEyes();
    positionA();

    positionD();
    positionE();
    positionA();
    blinkEyes();

    positionF();
    positionG();
    blinkEyes();
```

```
58
59    }
60  }
61
62  void positionA() {
63    eyeYright.write(101);
64    eyeYleft.write(102);
65    eyeX.write(88);
66    delay(delayy);
67  }
68
69  void positionB() {
70    eyeYright.write(101);
71    eyeYleft.write(102);
72    eyeX.write(88);
73    delay(delayy);
74  }
75
76  void positionC() {
77    eyeYright.write(100);
78    eyeYleft.write(103);
79    eyeX.write(149);
80    delay(delayy);
81  }
82
83  void positionD() {
84    eyeYright.write(108);
85    eyeYleft.write(95);
86    eyeX.write(64);
87    delay(delayy);
88  }
89
90  void positionE() {
91    eyeYright.write(119);
92    eyeYleft.write(84);
93    eyeX.write(124);
94    delay(delayy);
95  }
96
97  void positionF() {
98    eyeYright.write(90);
99    eyeYleft.write(114);
100   eyeX.write(66);
101   delay(delayy);
102 }
103
104 void positionG() {
105   eyeYright.write(90);
106   eyeYleft.write(113);
107   eyeX.write(140);
108   delay(delayy);
109 }
110
111 void blinkEyes() {
112   eyeLids.write(10);
113   eyeLids.write(90);
114   eyeLids.write(10);
115 }
116
117 void setup() {
118
119   // Servo Set up
120   //Jaw
121   jaw.setPeriodHertz(50);
122   jaw.attach(jawPIN);
```

```
123
124     //Shoulders
125     rightShoulder.setPeriodHertz(50);
126     rightShoulder.attach(rightShoulderPIN);
127     leftShoulder.setPeriodHertz(50);
128     leftShoulder.attach(leftShoulderPIN);
129
130     // Servos for Eyes
131     eyeX.setPeriodHertz(50);
132     eyeX.attach(eyeXPIN);
133     eyeYleft.setPeriodHertz(50);
134     eyeYleft.attach(eyeYPINleft);
135     eyeYright.setPeriodHertz(50);
136     eyeYright.attach(eyeYPINright);
137     eyeLids.setPeriodHertz(50);
138     eyeLids.attach(eyeLidsPIN);
139
140     // PS4 Controller set up
141     Serial.begin(9600);
142     PS4.begin("40:49:0f:d5:b2:80");
143     Serial.println("Ready.");
144     if (PS4.isConnected()) {
145       Serial.println("Connected.");
146     }
147   }
148
149   void loop() {//MAIN LOOP%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
150     if (PS4.isConnected()) {
151
152       R2pos = map(PS4.data.analog.button.r2, 0, 255, 180, 0);
153       shoulderDesY = map(PS4.data.analog.stick.ry, -128, 127, 180, 0);
154       shoulderDesX = map(PS4.data.analog.stick.rx, -128, 120, 180, 0);
155       shoulderCurrX = rightShoulder.read();
156       shoulderCurrY = rightShoulder.read();
157       int shoulderCurrYright = rightShoulder.read();
158       int shoulderCurrYleft = leftShoulder.read();
159
160       eyeDesX = map(PS4.data.analog.stick.lx, -128, 127, 150, 60);
161       eyeDesYright = map(PS4.data.analog.stick.ly, -128, 127, 120, 85);
162       eyeDesYleft = map(PS4.data.analog.stick.ly, -128, 127, 85, 120);
163       L2pos = map(PS4.data.analog.button.l2, 0, 255, 0, 90);
164
165       if ((millis() - servo_time) >= SERVO_SPEED) {
166         servo_time = millis();
167
168         //head control for right joystick in X %%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
169         if (shoulderDesX != shoulderCurrX) {
170           if ( shoulderDesX > shoulderCurrX ) {
171             rightShoulder.write(rightShoulder.read() + 3);//++
172             leftShoulder.write(leftShoulder.read() + 3);
173           }
174           else if (shoulderDesX < shoulderCurrX) {
175             rightShoulder.write(rightShoulder.read() - 1);//--
176             leftShoulder.write(leftShoulder.read() - 1);
177           }
178         }
179
180         //      eye control for left joystick
181         if (PS4.event.analog_move.stick.ly ) {
182           eyeYleft.write(eyeDesYleft);
183           eyeYright.write(eyeDesYright);
184         }
185         if (PS4.event.analog_move.stick.lx ) {
186           eyeX.write(eyeDesX);
187         }
```

```
188
189         //      blink controlled by L2
190         if ( PS4.data.button.l2 ) {
191           eyeLids.write(L2pos);
192         }
193
194         //       jaw controlled by R2
195         if ( PS4.data.button.r2 ) {
196           jaw.write(R2pos);
197           if (PS4.data.analog.button.r2 <= 20) {
198             jaw.write(150);
199           }
200         }
201
202         if ( PS4.data.button.triangle ) {
203           eyeloop();
204         }
205       }
206     }
207   }
```