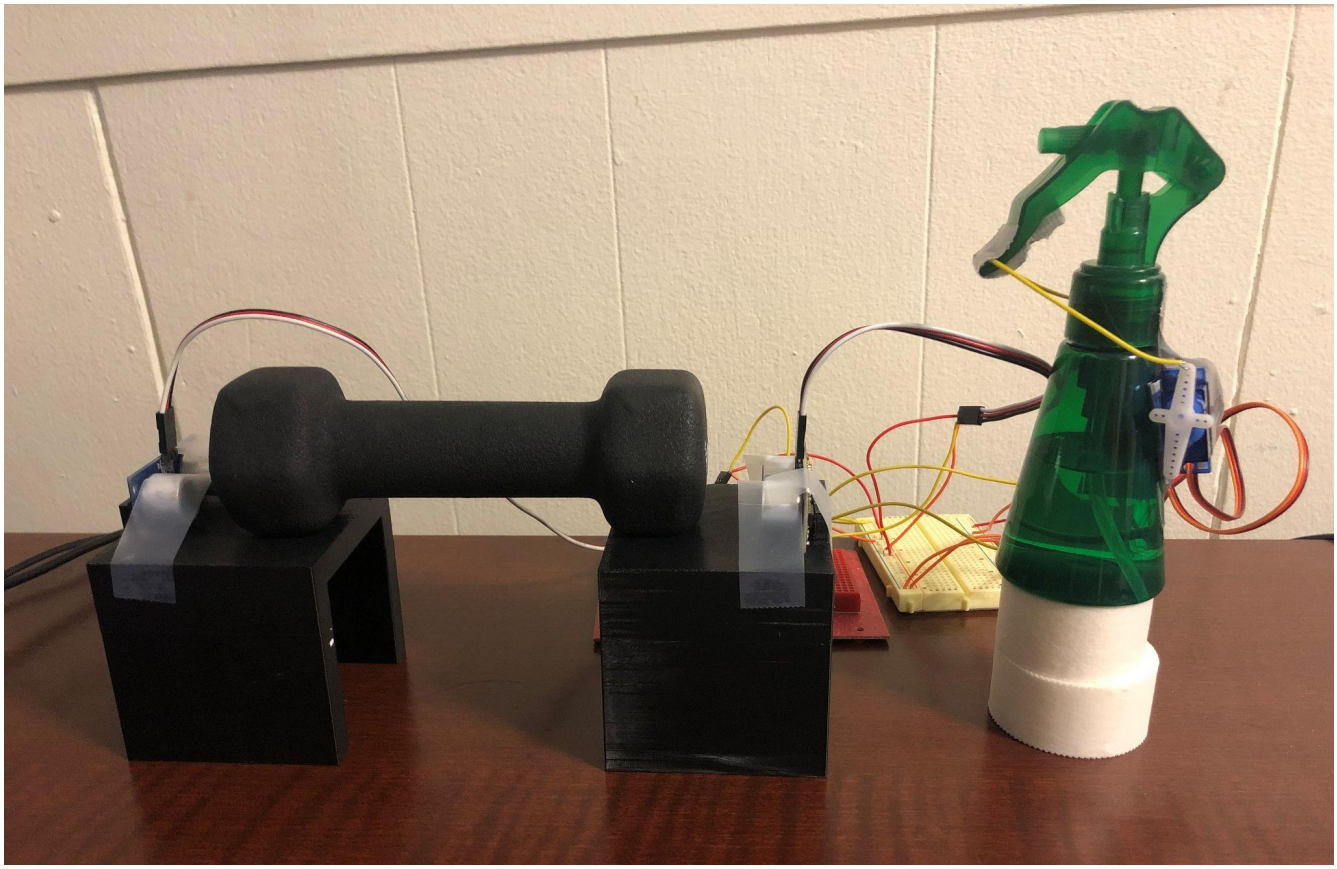
Mathias Kohler
Mk2227
MAE 3780 Individual Project Final Report, Fall 2021

Overview

My device is called the “Power Cleaner” and it sprays disinfectant on dumbbells at the gym by letting ultrasonic sensors and servo motors do the mechanical work so that gym-goers can focus on their workout, especially during COVID. The Power Cleaner works by having two Parallax ping ultrasonic distance sensors as inputs which will detect when the dumbbell is placed back on the rack. It will accomplish this by having each sensor placed on opposite ends of the dumbbell so when the sensors detect a change in distance to less than 8 cm they will send a signal to the output. The output consists of two servo motors, one positional and one continuous, attached to a small spray bottle with a small piece of rope connecting the two servo horns over the handle of the spray bottle. As the servo motors get the signal, they will rotate backward causing the rope to tense up and pull back the handle of the spray bottle leading to disinfectant spray being sprayed onto the dumbbell handle.



Design Considerations

The advice I would give to someone attempting to replicate the “Power Cleaner” would be to focus less on how the dumbbell would be placed on top of the ultrasonic distance

sensors without breaking them and focus more on how you would securely attach the servos to the spray bottle and the rope to the servo horns. I believe that I spent too much time trying to create a stand for the dumbbell to lay upon when in the real-life application the dumbbell would just lay on a dumbbell rack and what would really matter is how the circuit and spray bottle would fit into the rack. In the end I realized I didnt even use the stands and only needed the top part to hold the ultrasonic sensors in place which is why in my budget I accounted for only the ABS filament I used since I could have used a box for height. Lastly, I ran into issues with having the rope come apart from the servo horn multiple items leading to poor results which is why I recommend focusing on this.

If I could do the entire project over with the same constraints, I would still purchase two ultrasonic distance sensors but I would try to purchase a second positional servo and design a 3D printed ring jig for the spray bottle. The jig would allow both servos to be held firmly and allow for a stronger force when pulling back the handle compared to being held with duct tape. I would then spend any remaining budget on cardboard or any other cheap material to hold the ultrasonic distance sensors horizontally apart at the distance the length of the dumbbell handle. I think that these adjustments would allow for a better demonstration of how the final product would work as the positional servos are much easier to control and would reduce the cost of the 3D printing significantly. I would also spend more time thinking about how you could attach the rope to the two servos better so that it does not come apart every time you test it.
If I could do the entire project over with more time and money, I would spend time testing having the spray bottle in different positions and at different angles to see which one results in the greatest amount of surface area being disinfected. Then I would incorporate all of the features I mentioned in my answer above regarding the spray bottle jig and using a second position servo. I would also spend the extra money on building a makeshift “rack” for my dumbbell to lay upon possibly out of acrylic so that I can get a better idea of where I could fit all of the circuitry and components in an actual dumbbell rack. Lastly, I would try to think of a way to cover the handle of the dumbbell as it is being sprayed so that a person won’t try to reach for it right away and get sprayed. I think that these improvements would help me reduce costs by reducing the size of the 3D prints and create a seamless end product for gyms to implement.

<u>Assembly instructions</u>

1. Construct the circuit based on the circuit diagram in Figure B.1
2. Create the stand shown in Figure C.1 in a CAD software program such as Fusion360 to the dimensions given in the image and 3D print two of these using the stronger ABS filament
3. Once the two stands are 3D printed, attach the ultrasonic distance sensors to the side wall of the stand, as shown in Figure C.2, using tape or a zip tie
4. Use the dumbbell to set the two stands and sensors at the necessary distance for the head of the dumbbell to be the only thing on the stand as shown in Figure C.4
5. Place the spray bottle next to the dumbbell and the stands as seen in Figure C.5, a distance of 15 cm works best usually but you should test it beforehand

6. Once you figure out the distance the spray bottle should be at, attach the two servo motors to the sides of the spray bottle at a slight angle like in Figure C.6 and C.7 using duct tape
7. Then using a small piece of rope or a wire, connect the servo horns over the handle of the spray bottle under tension so that the rotation of the servos can pull back the handle with enough force to have spray come out using knots at the horns and tape at the handle
8. Your final product should look similar to Figure C.8 just missing the circuit and the rope

Operation instructions

Once the “Power Cleaner” is fully assembled as seen in Figure C.8, first take off the dumbbell to prevent any unwanted activations by your circuit. All you have to do is download the code from Appendix D to the Arduino board and power the circuit using a 4-AA battery pack or an equivalent voltage battery hooked up to the Arduino V_{in} and ground connections as seen in Figure B.1. You can then place the dumbbell on top of the two 3D printed stands and the ultrasonic distance sensors will automatically detect the dumbbell and wait 3 seconds before rotating backward, giving time to the user to remove their hand. It will then release the spray and wait 10 seconds before returning to its original position. In these 10 seconds, you will need to remove the dumbbell or remove the battery from the Arduino so that the circuit does not run in an infinite loop thus draining the battery.

Appendix A: Bill of Materials

Part name	Vendor/Source	Part #	Quantity	Price(ea/per unit)	Subtotal ($)
Ultrasonic Distance Sensor	Amazon - Smraza	HC-SR04	2	2.32	4.46
3-wire extension	Digi-Key	1568-1930-ND	2	1.35	2.70
Spray bottle (8 oz)	Amazon - SuperMMarK	ASIN: B082GPG1HG	1	2.25	2.25
ABS filament for 3D prints	RPL	N/A	2 (4.4055 g/ea)	4.40 ($0.5/g + $1.5 per job)	10.31
Arduino Uno	Digi-Key	1050-1024-ND	1	20.90	20.90
Breadboard	Newark	79X3922	1	2.71	2.71
Micro servo positional	DFRobot	SER0006	1	3.30	3.30
Micro servo continuous	DFRobot	SER-0043	1	3.90	3.90
Wire Kit	Amazon: Austor	ASIN : B07PQKNQ22	1	2.17	2.17
4 AA Battery Holder	Jameco	216187	1	1.75	1.75
AA batteries	McMasterr Carr	71455K58	4	0.40	1.60
USB Cable A/B	Monoprice	39918	1	1.09	1.09
				Total cost, purchased (grey rows)	19.72
				Total cost, scratch (all rows)	57.14

Appendix B: Circuit Diagram

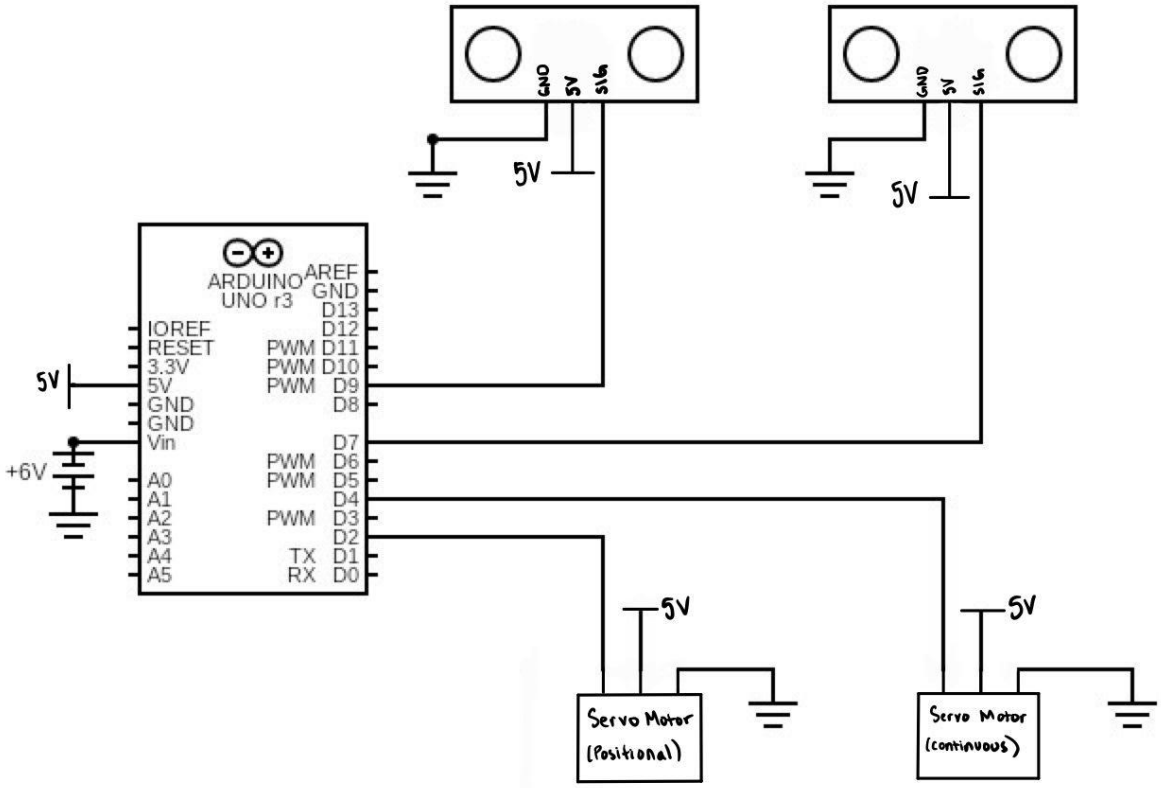


Figure B.1: Circuit Diagram

Appendix C: CAD files and drawings

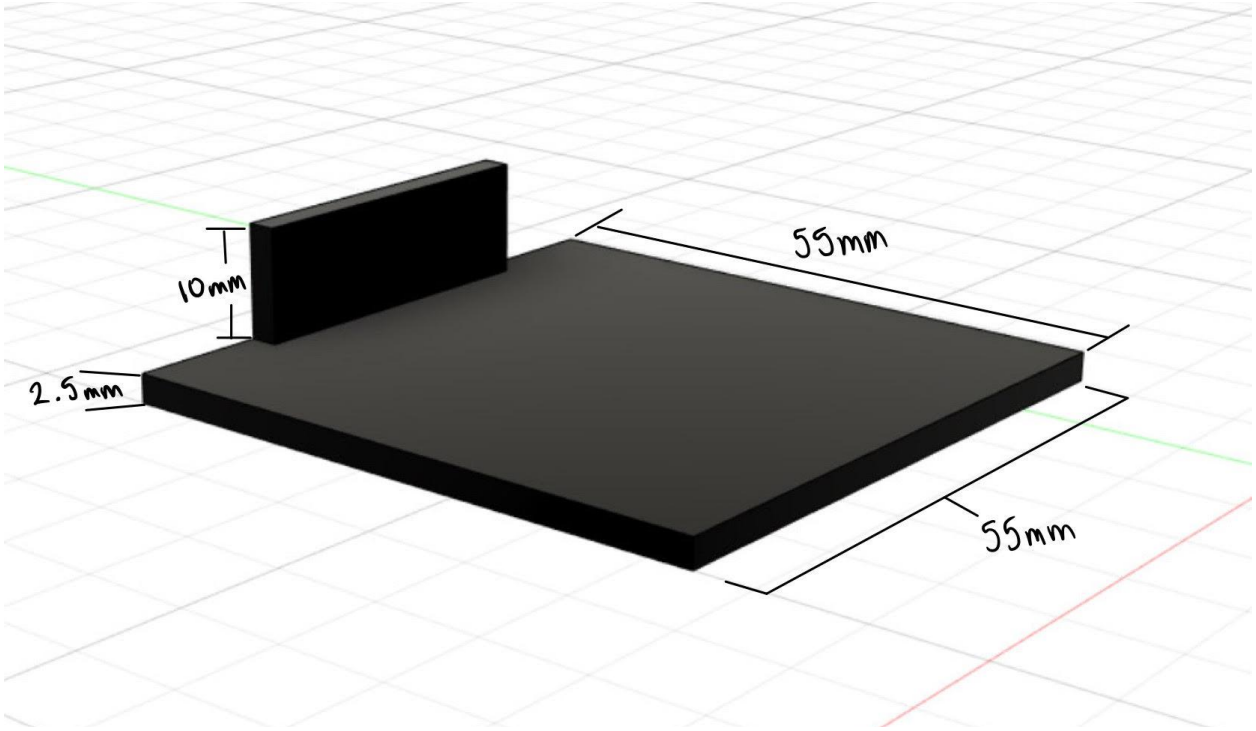


Figure C.1: Ultrasonic Distance Sensor Holder and Dumbbell Stand w/ measurements

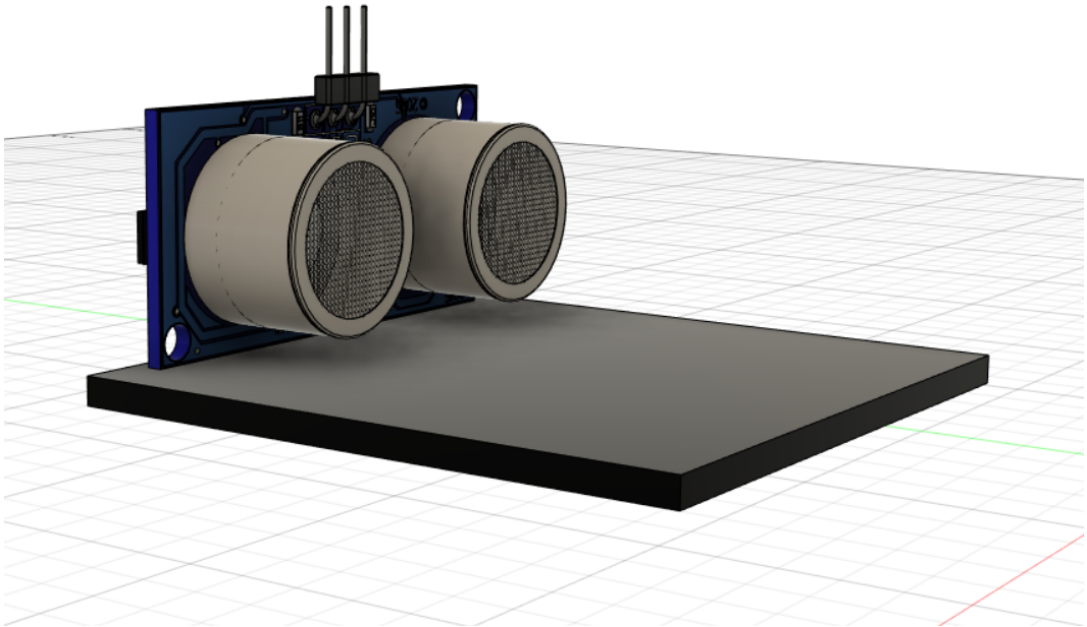


Figure C.2: Ultrasonic Distance Sensor attached to the dumbbell stand

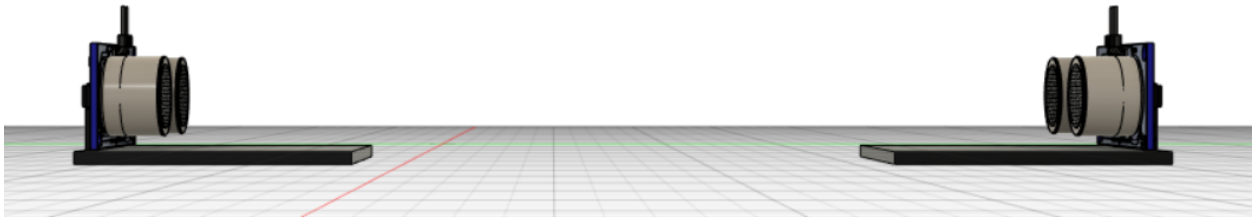


Figure C.3: Both dumbbell stands setup with their ultrasonic sensors attached

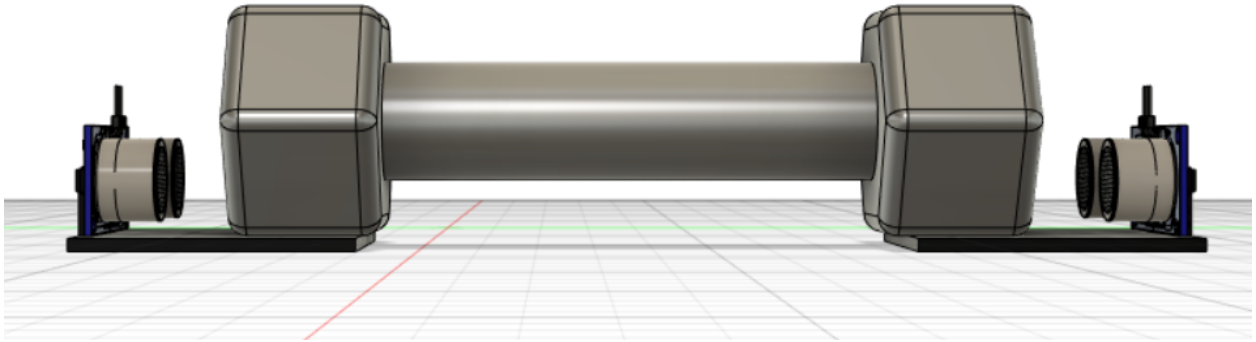


Figure C.4: Use dumbbell to determine distance apart of stands

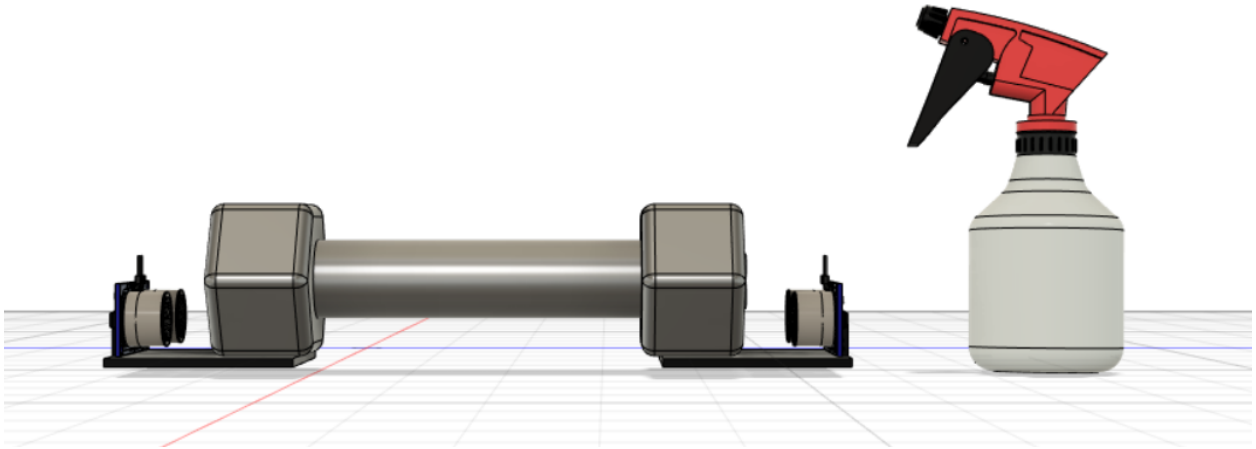


Figure C.5: Spray bottle placed next to dumbbell that needs spraying

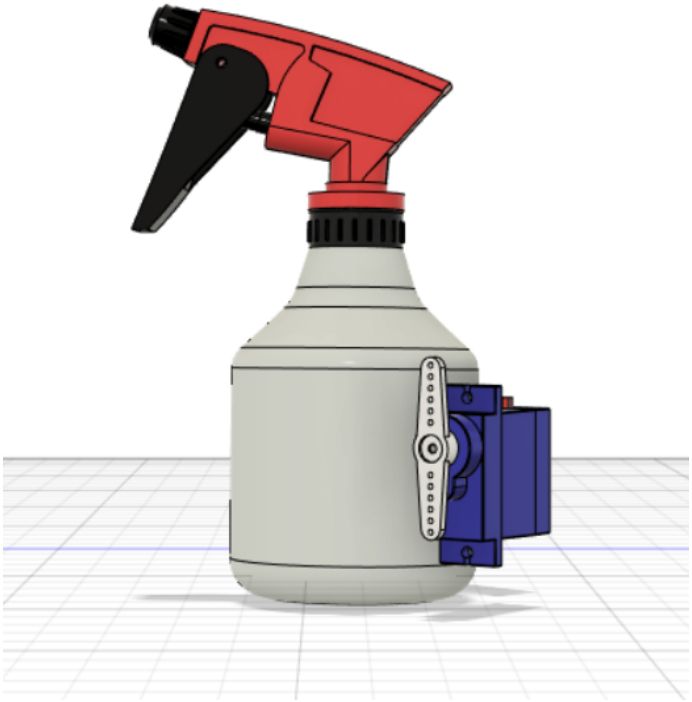


Figure C.6: Spray bottle with servo attached to one side

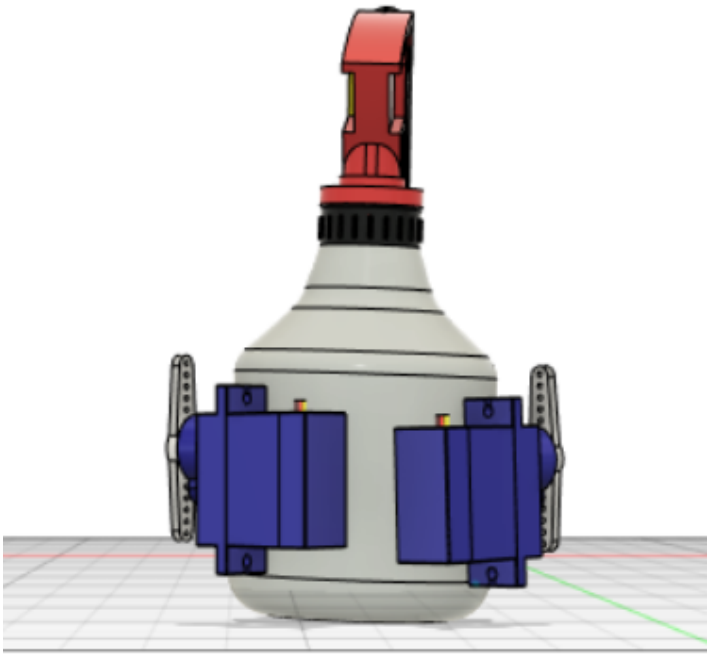


Figure C.7: Spray bottle with two servo motors attached

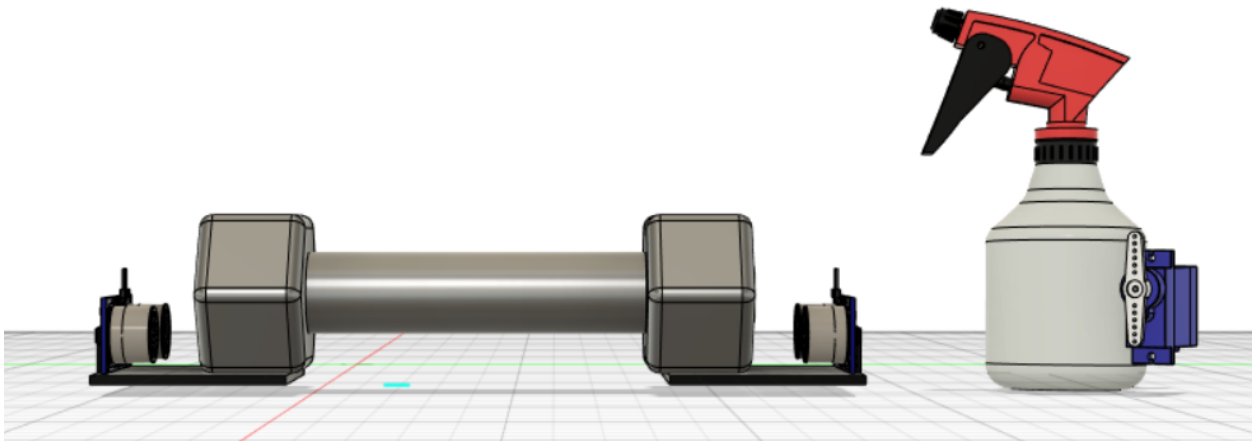


Figure C.8: Final assembly of the mechanical structure

Appendix D: Arduino Code

```
// C++ code
//This program begins to receive input from the ultrasonic distance sensor
//once the distance is less than 8 cm
//and then based on the distance it reads it will move the servors backwards
//to a 90 degree position

#include <Servo.h>

Servo servo_2; //Name both servos
Servo servo_4;
const int pingPin1 = 7;  //Initialize I/O pins
const int pingPin2 = 9;  //Initialize I/O pins
long duration1;  //Since PulseIn return an unsigned Long
int distance1;  //To save the distance
long duration2;  //Since PulseIn return an unsigned Long
int distance2;  //To save the distance

void setup()
{
  servo_2.attach(2);//attach servos to pins
  servo_4.attach(4);
 Serial.begin(9600);   //Initialize Serial communication
}


void loop(){
  distance1 = 0; //initialize distance variables
  distance2=0;
 pinMode(pingPin1, OUTPUT);   //Trigger pin as Output
  digitalWrite(pingPin1, LOW); //Make Trigger pin Low at start
  delayMicroseconds(3);
  digitalWrite(pingPin1, HIGH);
  delayMicroseconds(5);  //Make Trigger pin High for 5 uS to start sending the pulse
  digitalWrite(pingPin1, LOW);
  pinMode(pingPin1, INPUT);
   duration1 = pulseIn(pingPin1, HIGH);  //Save the time it took ultrasonic wave to come
back

  pinMode(pingPin2, OUTPUT);   //Trigger pin as Output
  digitalWrite(pingPin2, LOW); //Make Trigger pin Low at start
  delayMicroseconds(3);
  digitalWrite(pingPin2, HIGH);
  delayMicroseconds(5);  //Make Trigger pin High for 5 uS to start sending the pulse
  digitalWrite(pingPin2, LOW);
  pinMode(pingPin2, INPUT);
```

```
  duration2 = pulseIn(pingPin2, HIGH);  //Save the time it took ultrasonic wave to come back

    distance1 = microsecondsToCentimeters(duration1); //convert from duration to distance in cm
  distance2 = microsecondsToCentimeters(duration2);
  Serial.println(distance1); //print distances
  Serial.println(distance2);
  follow(); //Initializes the follow function
}

long microsecondsToCentimeters(long microseconds) {
  // The speed of sound is 340 m/s or 29 microseconds per centimeter.
  // The ping travels out and back, so to find the distance of the object we
  // take half of the distance travelled.
  return microseconds / 29 / 2;
}

void follow()
{
   if ((distance1<8)&&(distance2<8)){ //ensure both distances are met before beginning
    delay(3000); // 3 second delay to remove hand
    servo_4.write(90);  // tell servo 4 to go to position 90 degrees
    servo_2.write(45); // tell servo 9 to rotate backwards at 45 speed
    delay(450); //0.45 second delay at 45 speed
    servo_2.write(90); //tell servo 9 to stop
    delay(10000); //delay 10 seconds before returning to initial position
   } else {
    servo_4.write(0); //leave servo 4 at initial zero position
    servo_2.write(90); //pause servo 9
    delay(15); // Wait for 15 millisecond(s)
   }
}
```