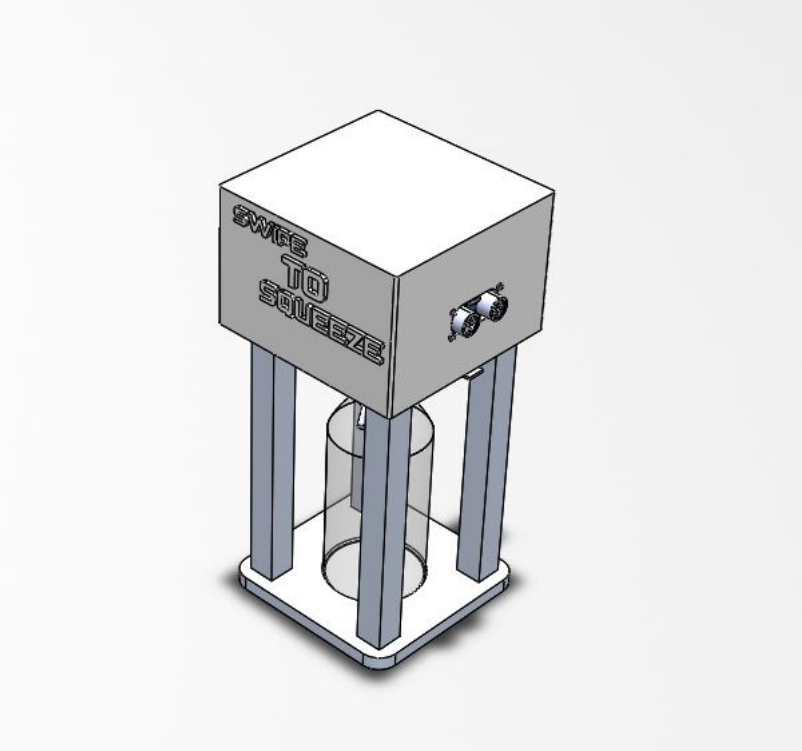

Swipe to Squeeze: No-contact Hand Sanitizer Dispenser

By Howard Yoon
ME 102B Fall 2020
Mechanical Engineering Department
University of California, Berkeley

Description of the product

On March 11, 2020, The World Health Organization has declared the coronavirus (COVID-19) outbreak a global pandemic. Since then, people in the world started to panic. This deadly virus has symptoms which are very similar to influenza or cold, so it's not easy to figure if one infected unless viral test is done individually. Worst of all, the virus is spread when people is exposed to respiratory droplets from those who are infected. COVID-19 forced people to stay away from each other, distancing at least 6 feet away. However, without contacts and interaction of individual, this world can't be in operation. In order to keep the wheels turning, Centers for Disease Control and Prevention (CDC) recommends that all individual is required to wash hands frequently to slow the spread. To practice it, washing hands with warm water and disinfecting soap would be the best case, but it is not available all the time. In such cases, hand sanitizers come in handy. Hand sanitizers that contain at least 60% alcohol are effective disinfecting hands and preventing further spread through physical contact. However, if one needs to use the hand sanitizer, unless it is in an automatic dispenser form, he or she needs to contact and hand-press the pump of the container which ironically can spread the virus. Although you can buy the automatic dispenser, I thought this would be a great opportunity to build a simple system that dispenses hand sanitizer without contact, so everyone who visits my place can safely disinfect their hands. I designed the system using an ultrasonic sensor and ESP32 from the micro lab kit and a servo motor I acquired from Amazon. Certain range of distance that the ultrasonic sensor reads is set. When the object (i.e. hand) is swiped within the range, it will send the signal to ESP32 and command the RC servo motor to rotate the arm to certain degree. Once the rotation is done, the arm rotates back to its original position.

Changes in the original plan

In order to build the planned system, I acquired 20kg RC servo motor from the Amazon. I was able to design an Arduino code that meets all the requirements. However, I realized that pressing the pump of the bottle requires a lot of force. The servo's torque was more than enough to press it; however, the problem I faced was the reaction force. When the arm attached to the servo pressed the pump, the reaction force was so large that the mount I made with a cardboard box would break and fail. To produce structures durable enough to withstand the large reaction force, I would have to use either laser cutter or 3D printer. However, due to the current situation, I was not able to get access to those machines. Seeking for a bright solution, I decided to follow a unique route, a mix of Option 1 and Option 2, and make a rendering model instead of actual model. I will provide the code, the video of the system operation I originally planned out, and the CAD drawings. When I was designing the mechanism, I was inspired by the motion of camshaft and valves in the internal combustion engine: when the camshaft rotates, the point of the cam will press down the valves. However, since I decided to go mix of option 1 and option 2, I needed a new mechanism that involves moving parts. For the new mechanism, I was again inspired by the car, rack-and-pinion gears on the steering motor. The rack is placed vertically and meshed with the gear. Shaft of the gear rotates counter clock wise and meshed rack will move downward. This movement is applied to press down the pump of the bottle.

List of parts used

Manufacturer	Part Number	Part Name	Material	Quantity
N/A_Inhouse	STS0001	Base Platform	18-8 Stainless Steel	1
N/A_Inhouse	STS0002	Vertical Member	18-8 Stainless Steel	4
N/A_Inhouse	STS0003	Top Plate	18-8 Stainless Steel	1
McMaster-Carr	2664N12	Metal Gear	1045 Carbon Steel	1
McMaster-Carr	2485N217	Metal Gear Rack	Black-Oxide 1045 Carbon Steel	1
N/A_Inhouse	STS0004	Press Plate	Plastic	1
N/A_Inhouse	STS0005	Motor Mount	18-8 Stainless Steel	1
Pololu	#4753	37D Gear Motor 50-	N/A	1

		70 Encoder		
N/A_Inhouse	STS0006	Top Plate Guide	Aluminum	1
N/A_Inhouse	STS0007	Top Cover	Plastic	1
Elecfreaks	HC-SR04	Ultrasonic Ranging Module	N/A	1
McMaster-Carr	90480A002	Hex Nut	Zinc-Plated Steel	4
McMaster-Carr	91864A008	Socket Head Cap Screw	Black-Oxide Alloy Steel	4
McMaster-Carr	92000A338	M5 x 0.8mm Thread Philips Screw	Passivated 18-8 Stainless Steel	8
McMaster-Carr	90604A724	M3 x 0.5mm Phillips Slotted Screw	Passivated 18-8 Stainless Steel	6
McMaster-Carr	92000A327	M5 x 0.5mm Thread Phillips Screw	Passivated 18-8 Stainless Steel	2
McMaster-Carr	96194A202	Flange Locknut	Zinc-Plated Steel	2
McMaster-Carr	5682A27	Phillips Screwdriver	Steel	1
McMaster-Carr	5511A41	L-Key Sets	Steel	1
JB Weld	8265S	Twin Tube 2 Oz	N/A	1

Fabrication and assembly process

For visionary reference, please see rendered images in the Appendix.

1. Produce STS0001, STS0002, STS0003, STS0005, STS0006 according to their corresponding CAD drawings (CNC machine and Lathe are recommended).
2. Produce STS0004 and STS0006 (Injection molding or 3D Printing is recommended).
3. Place four vertical members into grooves of the base platform. Fasten them to the bottom of the platform using 92000A338 screws.
4. Mount ultrasonic ranging module onto the right plane of the top cover. Slide four 91864A008 screws from inside, through the holes of the module and plane. Hand tighten four hex nuts to the screws. Adequately tighten the screws using the L-Key sets.
5. Mix up contents of JB Weld Twin Tube and apply them into the groove of the press plate. Place one end of the smaller side of the metal gear rack into the groove of the press plate. Firmly hold them until they are bonded.
6. Using 92000A327 screws and 96194A202 locknuts, mount the top plate guide onto the middle of top plate.
7. Place the top plate onto the four vertical members. Situate the motor mount to the one side of the top plate. Align the holes of motor mount and top plate to the thread of the members. Bolt them with four 92000A338 screws.
8. Set the motor into the motor mount. Align six holes of the motor mount and six threaded holes of the motor. Fasten the motor to the mount with six 90604A724 screws.
9. Slide the gear rack assembly from step 5 through the guides of top plate and motor mount. Make sure the press plate is toward the base platform.
10. Slide the metal gear to the shaft of the motor. Make sure the set screw is located onto the tap of the shaft. Tighten the set screw. When installing the gear, cogs of the rack and the gear must mesh correctly.
11. Connect all the wires and gently slide the top cover to the top plate.

Renderings & Drawings

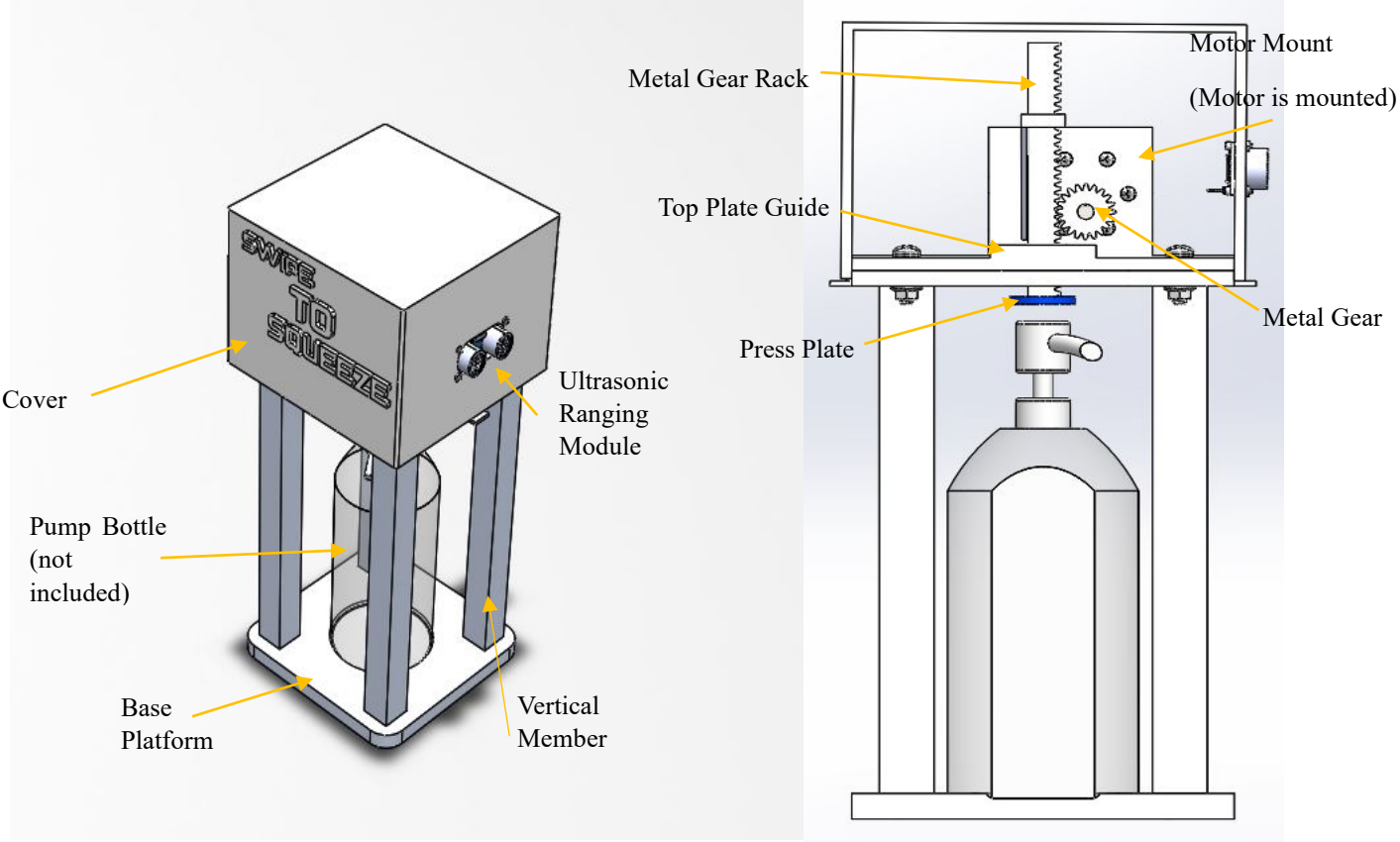

Figure 1&2. Isometric View (L) and Section View (R)

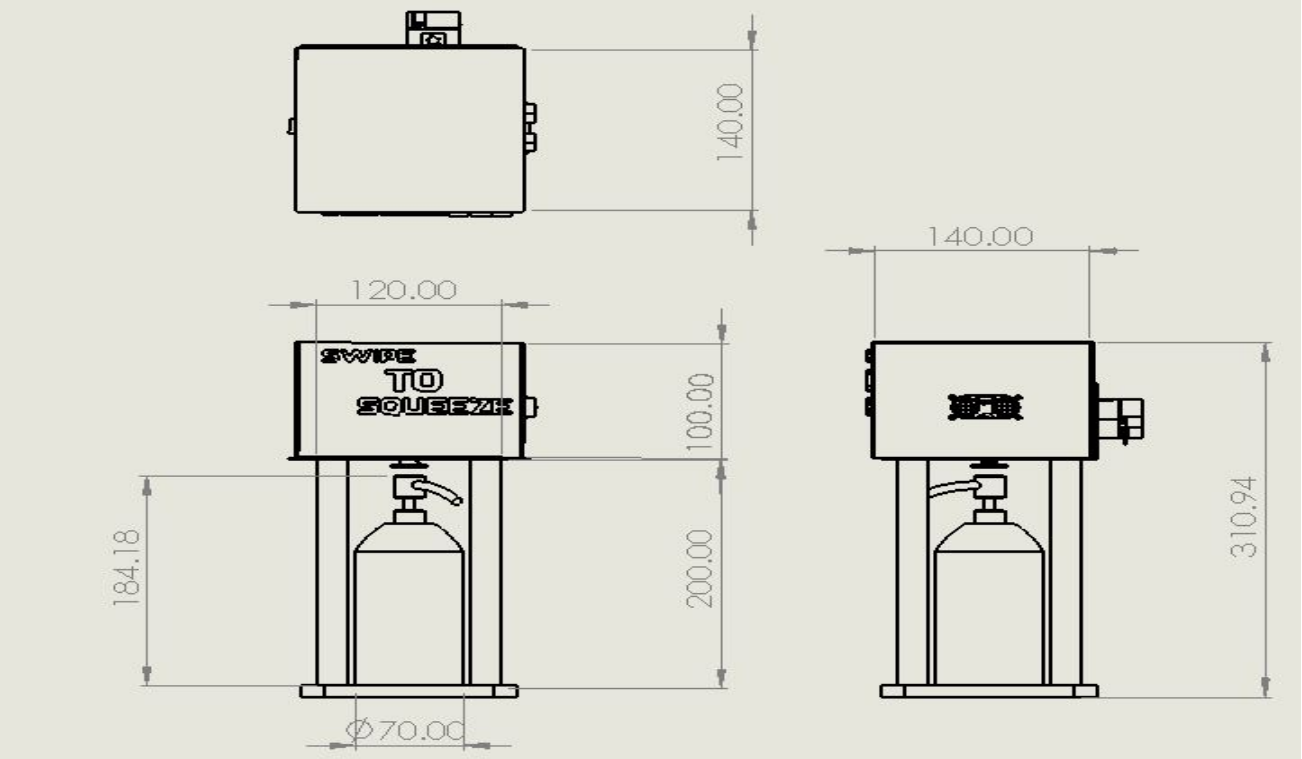

Figure 3. 3rd Angle Projection with Key Dimensions

Appendix

A.1 Arduino Code Based on Primary Design

```
#include <Servo.h>

#define echoPin 14
#define trigPin 15

static const int servoPin = 4;

Servo servo1;

long duration; // variable for the duration of sound wave travel
int distance; // variable for the distance measurement

void setup() {
  pinMode(trigPin, OUTPUT);
  pinMode(echoPin ,INPUT);
  Serial.begin(115200);
  servo1.attach(servoPin);
  //servo1.write(); // initial position(degree) of the servo motor

}

void loop() {
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10); // generate the ultrasound, send out an 8 cycle sonic burst which will be received in the Echo Pin
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH); //output time in microseconds the sound wave traveled
  distance = duration*0.034/2; //speed of the sound:0.034cm/microsececond, the sound wave travels forward and bounce backward
  Serial.println(distance);
  //In order to compensate that, we divide the duration in half
  if (distance >= 7 && distance <= 12) {
    for(int posDegrees = 40; posDegrees < 120; posDegrees++) {
      servo1.write(posDegrees);
      delay(20);
    }
   servo1.write(40);
  }
}
```

A.2. Wiring Diagram of Primary Design