Social Distancing Device for COVID-19

Modeling and Designing

Manthan Pawar
New York University,
Tandon School of Engineering
Brooklyn, New York
mvp321@nyu.edu

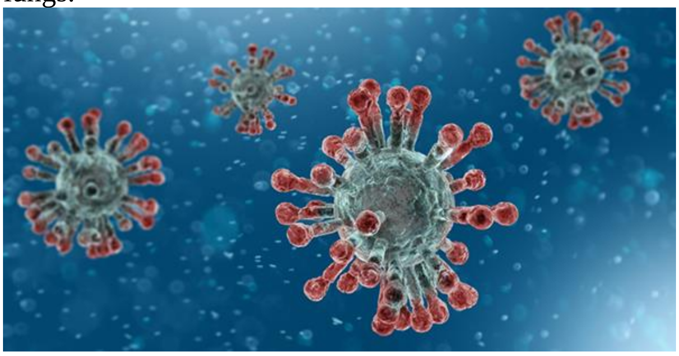
Zewen Wu
New York University
Tandon School of Engineering
Brooklyn, New York
zw2420@nyu.edu

***Abstract*— Novel Coronavirus disease 2019, known as COVID-19 hit the humanity om global scale and was announced as a global pandemic by World Health Organization (WHO). The coronavirus is thought to spread mainly from person to person. This can happen between people who are in close contact with one another. Droplets that are produced when an infected person coughs or sneezes may land in the mouths or noses of people who are nearby, or possibly be inhaled into their lungs. Understanding how the virus spreads reinforces the importance of social distancing and other health-promoting behaviors. Because COVID-19 spreads from person to person, reducing the ways people come in close contact with each other is essential. According to Centers for Disease Control and Prevention (CDC) guidelines, to practice social or physical distancing, you should stay at least 6 feet (2 meters) from other people. In these dark times, everyone needs wants to get out, and the key to start the economy back again is keeping those social distancing norms and work simultaneously. Therefore, we are proposing a prototype of a device which will help people keeping these social distancing norms. The device will notify the user if there is someone inside the 6ft radius from the person wearing it. That way the user can know that he/she needs to get distant from someone. The device is cap which can convert itself into a face shield, one of the most important Personal Protective Equipment (PPE) used against the fight of coronavirus, when a person is detected inside the radius of 6 ft from the person wearing it.**

Keywords*— COVID-19, Social Distancing, Centers for Disease Control and Prevention (CDC), Pandemic, Restart Economy, Personal Protective Equipment (PPE), Face Sheild, Cap.

I. Introduction

Coronaviruses are an extremely common cause of colds and other upper respiratory infections. COVID-19, short for "coronavirus disease 2019," is the official name given by the World Health Organization to the disease caused by this newly identified coronavirus. People of any age should take preventive health measures like frequent hand washing, physical distancing, and wearing a mask when going out in public, to help protect themselves and to reduce the chances of spreading the infection to others. Recently published research found that on average, the time from exposure to symptom onset (known as the incubation period) is about five to six days. However, studies have shown that symptoms could appear as soon as three days after exposure to as long as 13 days later. These findings continue to support the CDC recommendation of self-quarantine and monitoring of symptoms for 14 days post exposure. The coronavirus is thought to spread mainly from person to person. This can happen between people who are in close contact with one another. Droplets that are produced when an infected person coughs or sneezes may land in the mouths or noses of people who are nearby, or possibly be inhaled into their lungs.



A person infected with coronavirus — even one with no symptoms — may emit aerosols when they talk or breathe.

Aerosols are infectious viral particles that can float or drift around in the air for up to three hours. Another person can breathe in these aerosols and become infected with the coronavirus. Therefore, everyone should cover their nose and mouth when they go out in public.
Coronavirus can also spread from contact with infected surfaces or objects. For example, a person can get COVID-19 by touching a surface or object that has the virus on it and then touching their own mouth, nose, or possibly their eyes.

Need of Face Shield for everyone:

Study done by National Institute of Allergy and Infectious Diseases' Laboratory of Virology in the Division of Intramural Research in Hamilton; Montana helps to answer this question. The researchers used a nebulizer to blow coronaviruses into the air. They found that infectious viruses could remain in the air for up to three hours. The results of the study were published in the *New England Journal of Medicine* on March 17, 2020.

Face offer more effective protection against coronavirus than masks and should be worn by the public whenever they leave home, according to US physician and epidemiologist Michael Edmond.
Edmond, an infectious diseases physician and hospital epidemiologist in Iowa City, USA, has been advocating the use of face shields on his blog.

He believes the simple devices are more effective than masks at protecting the eyes, nose and mouth from Covid-19 infection and praised efforts by architects and designers to manufacture the devices as "one of the silver linings of this pandemic". According to him, every person should have the face shield.

II. Proposed Solution

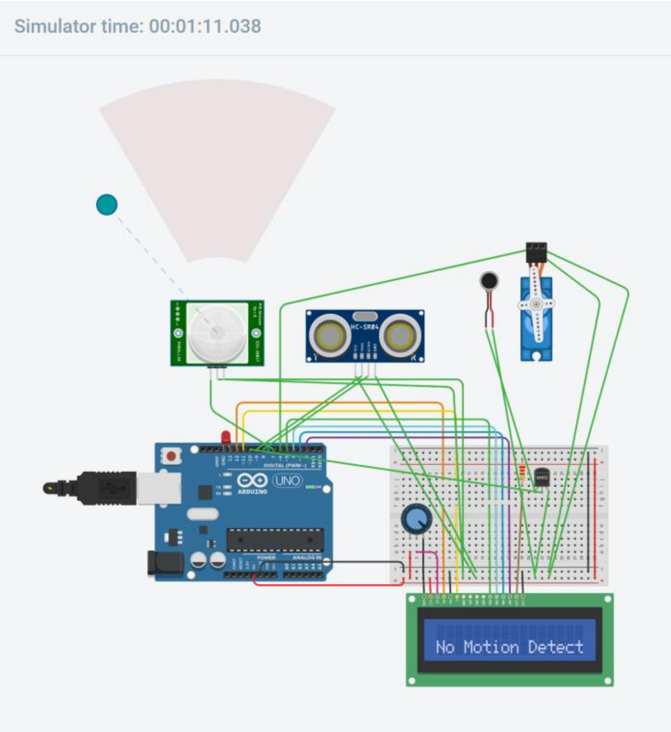
Our main idea is a sun cap whose shade converts into a face shield when it detects a presence of human within 6ft radius from the user and notifies the user.
Due to lack of hardware however, we are implementing the following prototype.
The device we are proposing is a cap which can turn into a face shield. The device detects if there is someone within 6ft behind the user. If there is someone within 6ft, the device flaps the shield down and warns the user about the presence of someone 6ft behind him/her. The device has a Passive Infrared Sensor (PIR Sensor) which detects the human motion. Once the motion is detected, Ultrasonic sensor detects the actual distance of the subject from the user. The vibration motor in the pocket and the LED on the cap tells the user that the distance is lower than 6 ft. Based on the intensity of vibration and LED brightness, criticality of the distance is represented. The LCD shows the information of motion detection and the distance. A servo motor is used to flap down the shield. Also, the user has a switch in his hand which he can use to directly trigger the shield. An encoder is used to Set up the user-defined rotation angle of servo motor of the shield. Currently we are implementing only the sensing of human from behind as lack of resources. Our actual proposal for the final project is that we will have a servo-actuated platform over which a camera will be mounted that continuously rotated and scans for motion in 360 degree. We are going to apply pinhole camera model and simple tracking algorithm, assisted with OpenCV, to find out if there is any person tends to be less than 6 feet away from the user.

Bill of Material of Major Components:

Component	Price
Propeller Activity Board	79
Passive Infrared Sensor (PIR Sensor)	9.95
Ultrasonic Sensor	3.95
Servo Motor	12.95
Rotary Encoder	4.95
Cap	9.99
Breadboard	4.95
Total	**125.74**

III. Simulation Prototype

The project is done on both hardware and simulation platform. The mechatronics for the device is tested in simulation and hardware simultaneously to make things remotely work during these works from home period. The simulation is done on Autodesk TinkerCad Arduino UNO platform. The simulation is used to create algorithms which will then be used to make a propeller hardware implementation. The online simulation can be accessed using the following link:
https://www.tinkercad.com/things/aAzLeJOregB-social-distancing-device/editel
Following components are used in the simulation:
1) Arduino UNO
2) LCD Display
3) Passive Infrared Sensor (PIR Sensor)
4) Ultrasonic Sensor
5) Vibration Motor
6) Servo Motor



The working of the simulation prototype:
The simulation runs when hit the button run. There are two section or experiment workspace infront of both PIR sensor and the unltasonic sensor. When the dot in PIR workspace is moved, it shows the information on display. It also blinks the LED. When the dot in ultrasonic sensor workspace is moved it shows the information in the serial monitor. When the distance is below 100cm, it triggers the servo motor flap and the vibration motor. The working code for the same is given below:

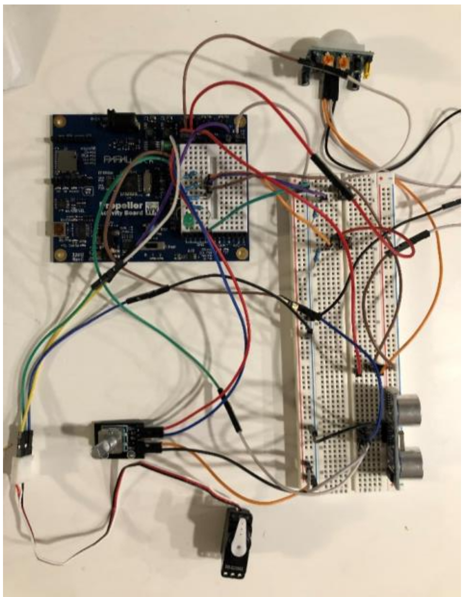
```
#include <LiquidCrystal.h>
#include <Servo.h>

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);
int led = 13;                // the pin that the LED is atteched
to
int sensor = 7;              // the pin that the sensor is atteched
to
int state = LOW;             // by default, no motion detected
int val = 0;                 // variable to store the sensor status
(value)

Servo servo1;
int trigPin = 9;
int echoPin = 8;
long distance;
long duration;



void setup() {

  lcd.begin(16, 2);
  pinMode(led, OUTPUT);      // initalize LED as an output
  pinMode(sensor, INPUT);    // initialize sensor as an
input
  Serial.begin(9600);        // initialize serial

  servo1.attach(6);
 pinMode(trigPin, OUTPUT);
 pinMode(echoPin, INPUT);// put your setup code here,
to run once:

 pinMode( 10 , OUTPUT);  // Must be a PWM pin
}



void loop() {

  // set the cursor to column 0, line 1
  // (note: line 1 is the second row, since counting begins
with 0):
  lcd.setCursor(0, 1);


  ultra();
  servo1.write(0);
  if(distance <= 100){
  servo1.write(90);
```

```
  }


  val = digitalRead(sensor);   // read sensor value
  if (val == HIGH) {           // check if the sensor is HIGH
     // Print a message to the LCD.
  lcd.print("Motion Detected!");
    digitalWrite(led, HIGH);   // turn LED ON

  analogWrite( 6 , 153 );  // 60% duty cycle
  delay(500);              // play for 0.5s
  analogWrite( 6 , 0 );    // 0% duty cycle (off)

    if (state == LOW) {
      Serial.println("Motion Detected!");
      state = HIGH;       // update variable state to HIGH
    }
  }
  else {
    // Print a message to the LCD.
  lcd.print("No Motion Detected");
      digitalWrite(led, LOW); // turn LED OFF


      if (state == HIGH){
        Serial.println("Motion stopped!");
        state = LOW;       // update variable state to LOW
    }
  }

}



void ultra(){
  digitalWrite(trigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(trigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigPin, LOW);
  duration = pulseIn(echoPin, HIGH);
  distance = duration*0.034/2;
  }
```

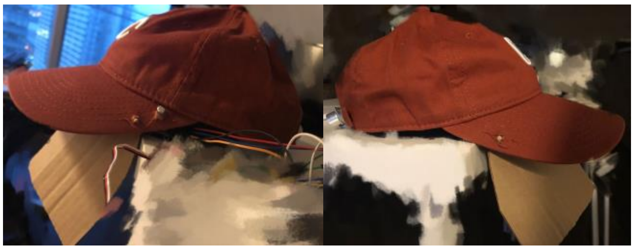
IV. HARWARE PROTOTYPE

Following components are used in the prototype:

1) Propeller Activity board
2) Passive Infrared Sensor (PIR Sensor)
3) Ultrasonic Sensor
4) Servo Motor
5) LED (On-board LEDs (Pin 26 and Pin 27) are used as well)
6) Rotary Encoder
7) Push Buttons



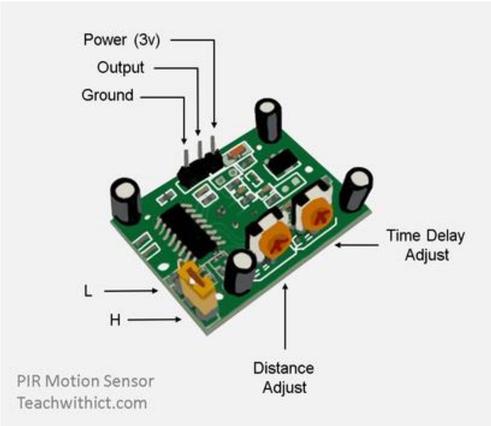
The hardware prototype is implemented on a cap. A cardboard piece is shown as a shield for a temporary prototype. The hardware is rather a mechatronic prototype. Due to the limitation of tools and parts, the prototype is far away from the final product. For the on and off of the mask, the design takes advantage of the gravity. If the mask is intended to be on the face, the servo motor will be commanded to rotate down to function as a limit switch. Also, the mask can be lifted up by the motor once it is required to.

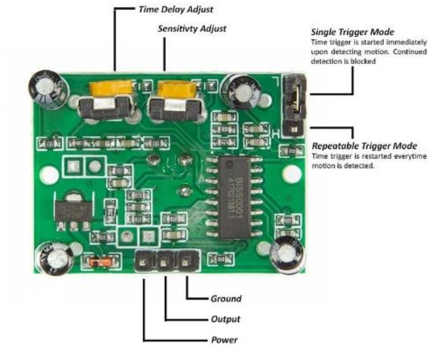


About PIR sensor:

A passive infrared sensor (PIR sensor) is an electronic sensor that measures infrared (IR) light radiating from objects in its field of view. They are most often used in PIR-based motion detectors. PIR sensors are commonly used in security alarms and automatic lighting applications.

PIR sensors detect general movement, but do not give information on who or what moved. For that purpose, an active IR sensor is required.

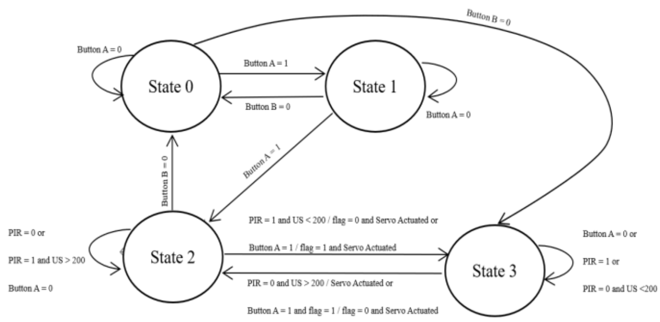




V. User Interface and Control

At the current stage, there is not a communication constructed between the prototype and mobile device, which means all components have to be assembled on the cap. In this case, to optimize the usability, User Interface has to be designed as simple as possible. Therefore, only two buttons and a rotary encoder has been designed for users to activate the system and adjust the setting. The overall procedure of user interface can be specified as: a) turn on the power; b) press button A (green LED will be on, which indicates that the user can start set the rotation angle of the mask based on personal preferences); c) rotate the rotary encoder to adjust the setting (the default rotation angle is 90 degree); d) press button A to activate the system; e) press button B so that the system will return to its initial state (Green LED will be off and button B can be pressed anytime during the progress). Here, referring to the first figure in Section IV, button A is the push button on the top while button B is the push button at the bottom.

Other than UI designed for the product, a closed-loop control system has been designed for the prototype. When the system is in progress, it receives signals from button A, ultrasonic sensor and PIR sensor continuously. The bus contains these three signals will be pushed to the controller which is designed to generate the input signal for the servo motor. The position of the servo motor will be integrated to the bus which is the input for the controller. The control system, while it was being implemented can be considered as a finite state machine, referring to the design of digital logic circuit.
To be specific, except procedures that has been introduced above, if button A is either be pushed or a motion within 6 feet has been detected, the mask is expected to be actuated down. Otherwise, the system will remain in the current state. Similar strategy has been applied to state 3. One thing should be noted that, if the state switch is caused by the state of button A, moving back to state 2 can only be triggered by the state of button A, which has been realized by introducing a flag, which is either an input or an output in the state machine.



VI. Program

Final program for the hardware prototype is as follows:

```
#include "simpletools.h" // Include simple tools
#include "servo.h" // Include servo library

#define ButtonPin 6
#define ButtonPinB 7
#define TrigPin 10
#define EchoPin 11
#define PIRPin 5
#define ServoPin 15
#define PinA 14
#define PinB 13
#define Safedistance 200
#define LEDPin 1

int main()
{
```

```
set_directions(14, 13, 0b00); // ensure pin 13 and 14 are
all input pins
set_direction(PIRPin, 0b0);
set_directions(ButtonPinB, ButtonPin, 0b00);
int pin_last = input(PinA);
long duration;
int distance;
int flag = 0;
int state = 0; // initial state
servo_angle(ServoPin, 0); // servo initial pos
int last_state = 0;
int last_state_B = 0;
int button = 0;
int count = 0;
int set_degree = 0;
int re_counter = 0;
int US_last = 0;
while (1)
{
int PIR = input(PIRPin);
int button_state = input(ButtonPin);
int button_state_B = input(ButtonPinB);

if (last_state_B == 0 && button_state_B == 1)
{
state = 0;
low (LEDPin);
}
last_state_B = button_state_B;
if (last_state == 0 && button_state == 1)
{
count = count + 1;
button = 1;
printf("Button state changed, button = %d\n", button);
printf("count = %d\n", count);
}
else
{
button = 0;
}
last_state = button_state;
duration = ping(TrigPin,EchoPin);
distance = duration/29/2;
if (distance > 2000)
{
distance = US_last;
}
if (state != 0)
{
if (distance - US_last < 20)
{
PIR = 0;
}
}
print("PIR = %d\n", PIR);
US_last = distance;
printf("%d\n", distance);
switch(state)
{
case 0:
printf("initial state\n");
if (count == 1)
{
printf("set the position of the mask\n");
state = 1;
high(LEDPin);
}
break;
case 1:
if (count == 1)
{
int aVal = input(PinA);
if (aVal != pin_last)
{
printf("Updated!\n");
if (aVal != input(PinB))
{
if (set_degree < 90)
{
re_counter = re_counter + 1;
//printf("++%d\n", re_counter);
set_degree = set_degree + 10;
}
}
else
{
if (set_degree > 0)
{
re_counter = re_counter - 1;
set_degree = set_degree - 10;
}
}
printf("%d\n", re_counter);
servo_angle(ServoPin,10*(set_degree));
}
pin_last = aVal;
}
else{
count = 0;
state = 2;
servo_angle(ServoPin, 0);
printf("state 0\n");
//high(1);
high(26);
```

```
pause(3000);
servo_disable(ServoPin);
}
break;
case 2: //initial state
if (button == 1 && PIR == 0)
{
printf("Button pushed\n");
servo_angle(ServoPin, set_degree*10);
pause(1000);
servo_disable(ServoPin);
flag = 1;
state = 3;
PIR = 0;
printf("Mask On\n");
low(26);
high(27);
pause(5000);
}
else if (button == 0 && PIR == 1)
{
if (distance < Safedistance)
{
printf("Person Behind Detected, distance = %d\n", distance);
servo_angle(ServoPin, set_degree*10);
pause(1000);
servo_disable(ServoPin);
state = 3;
printf("Mask On\n");
low(26);
high(27);
pause(5000);
}
}
printf("%d\n", count);
break;
case 3: // Mask off
if (PIR == 0 && button == 1)
{
printf("Button pressed to lift the mask\n");
servo_angle(ServoPin, 0);
pause(1000);
servo_disable(ServoPin);
state = 2;
PIR = 0;
printf("Mask Off\n");
flag = 0;
low(27);
high(26);
}
else if(PIR == 1 && distance > Safedistance && flag == 0)
{
servo_angle(ServoPin, 0);
pause(1000);
servo_disable(ServoPin);
state = 2;
printf("Mask Off\n");
low(27);
high(26);
}
printf("%d\n", count);;
break;
}
}
}

int ping(int trig, int echo)//trig is trigger pin, echo is echo pin
{
low(trig);//set trig low for start pulse
low(echo);//set echo low to be safe
pulse_out(trig, 10);//send the minimum 10 ms pulse on trig to start a ping
return pulse_in(echo, 1);//get the pulse duration back on echo pin
}
```

ACKNOWLEDGMENT

We thank professor Kapila and the class of Advanced Mechatronics, NYU Tandon school of Engineering for their guidance and suggestions for the project.

FUTURE SCOPE

For the next project we will have a servo-actuated platform over which a camera will be mounted that continuously rotates and scans for motion in 360 degree. We are going to apply pinhole camera model and simple tracking algorithm, assisted with OpenCV, to find out if there is any person tends to be less than 6 feet away from the user. By the next prototype, we hope to get our hands on some more hardware resources. We propose to have a better hardware prototype and robust algorithm based on image processing to scan the human motion and assist social distancing.

REFERENCES

[1] https://www.health.harvard.edu/diseases-and-conditions/covid-19-basics

[2] https://www.cdc.gov/coronavirus/2019-ncov/prevent-getting-sick/social-distancing.html
[3] https://www.who.int/health-topics/coronavirus#tab=tab_1
[4] https://coronavirus.jhu.edu/map.html