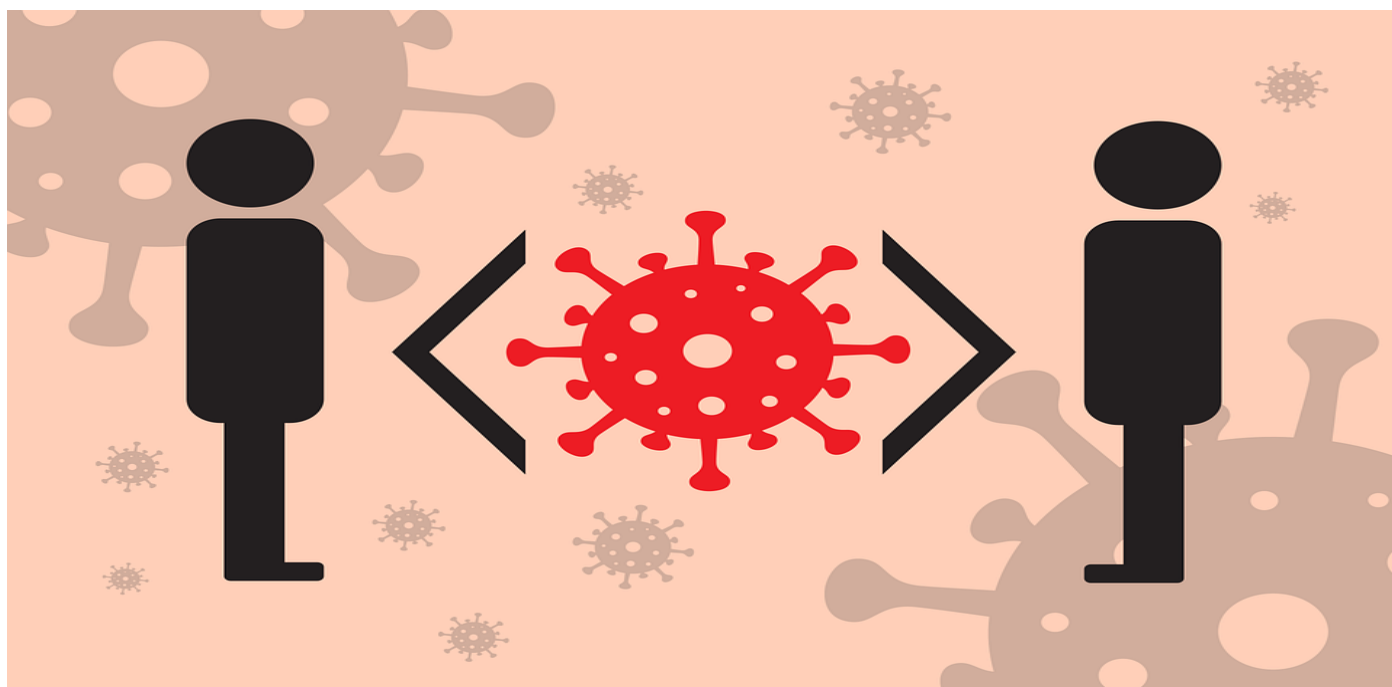


Computer Organization and Architecture Lab
[15B17CI373]

Project Report

Social Distancing Alert



Members

Pranav Gupta 19803021

Ritawari Pareek 19803009

Kanishka Khullar 19803011

Submitted to

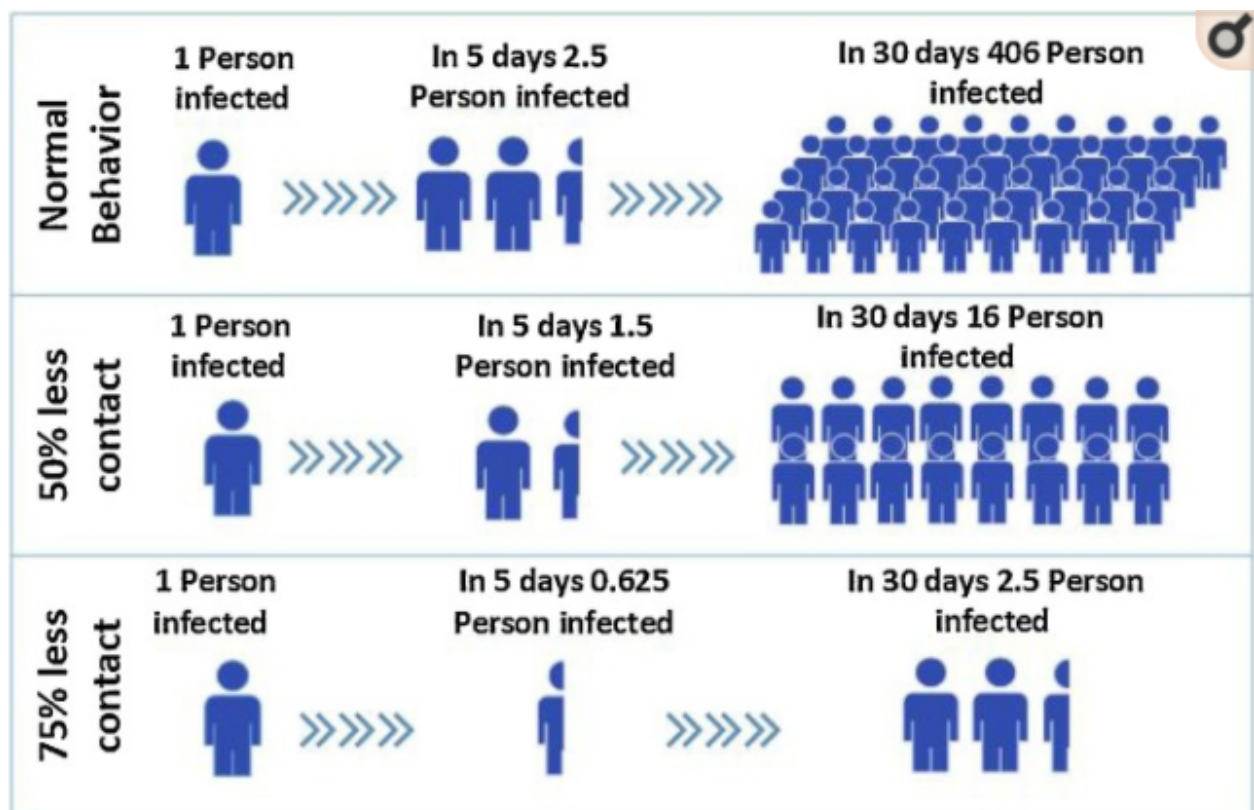
Mrs Amarjeet Kaur

ABSTRACT

The ongoing COVID-19 coronavirus outbreak has caused a global disaster with its deadly spreading. Due to the absence of effective remedial agents and the shortage of immunizations against the virus, population vulnerability increases. In the current situation, as there are no antidotes available; therefore, social distancing is thought to be an adequate precaution (norm) against the spread of the pandemic virus. The risks of virus spread can be minimized by avoiding physical contact among people. The purpose of this project is to alert the person if he is in close contact with another person.

Further, this could be installed within the mobile phones, and using the AAROGYA SETU app the person can be alerted if he/she came in close contact with an infected person.

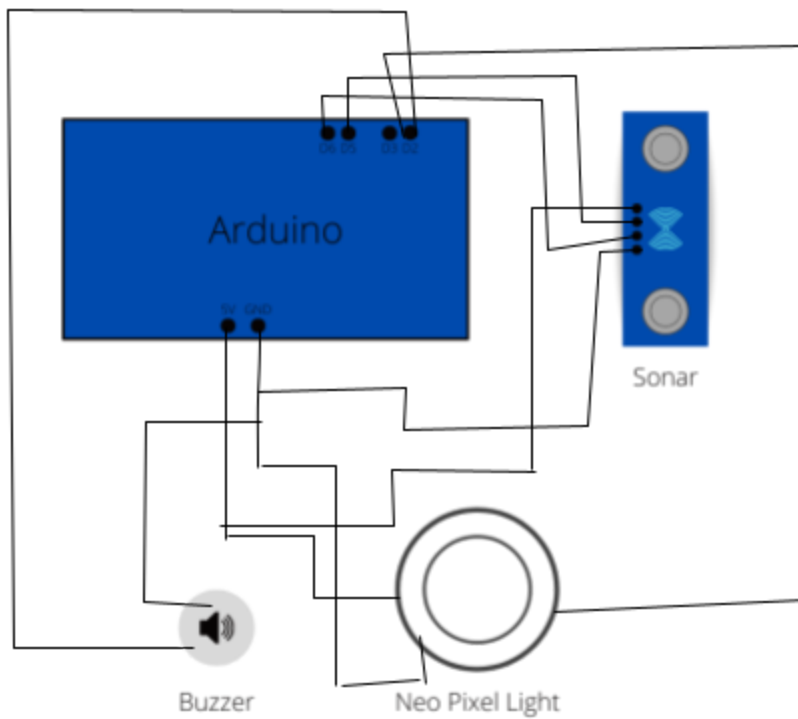
Also as everyone is aware of the OMICRON variant, this project could be very helpful in preventing the rapid spread of the virus.



COMPONENT USED ([Tinkercad](#))

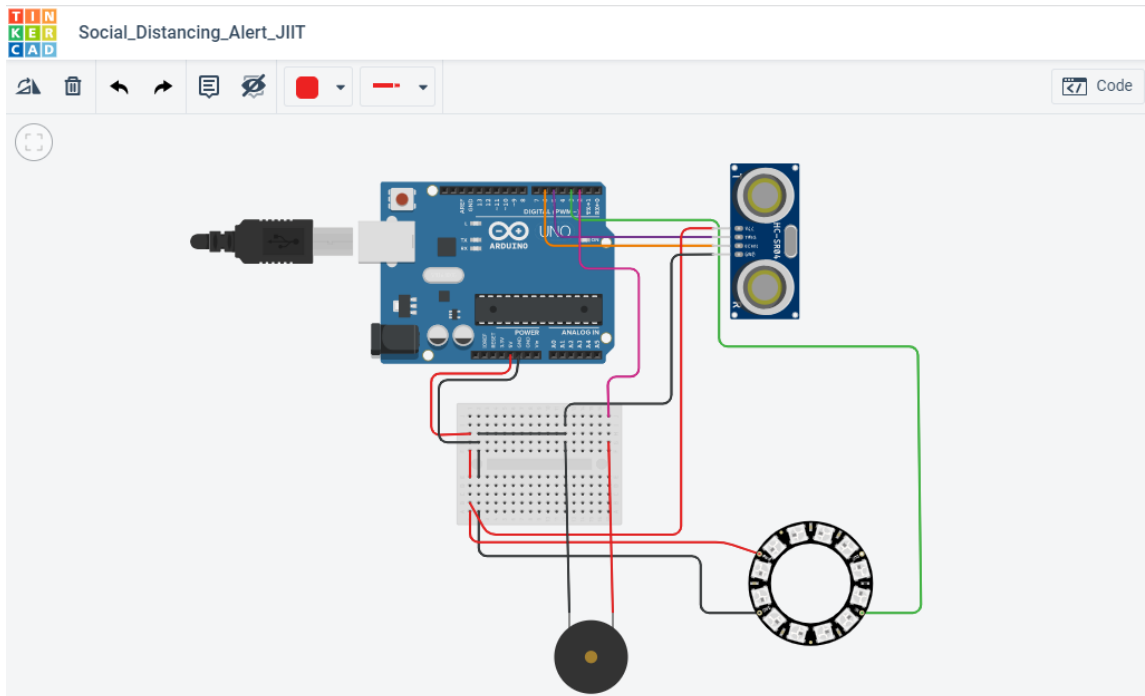
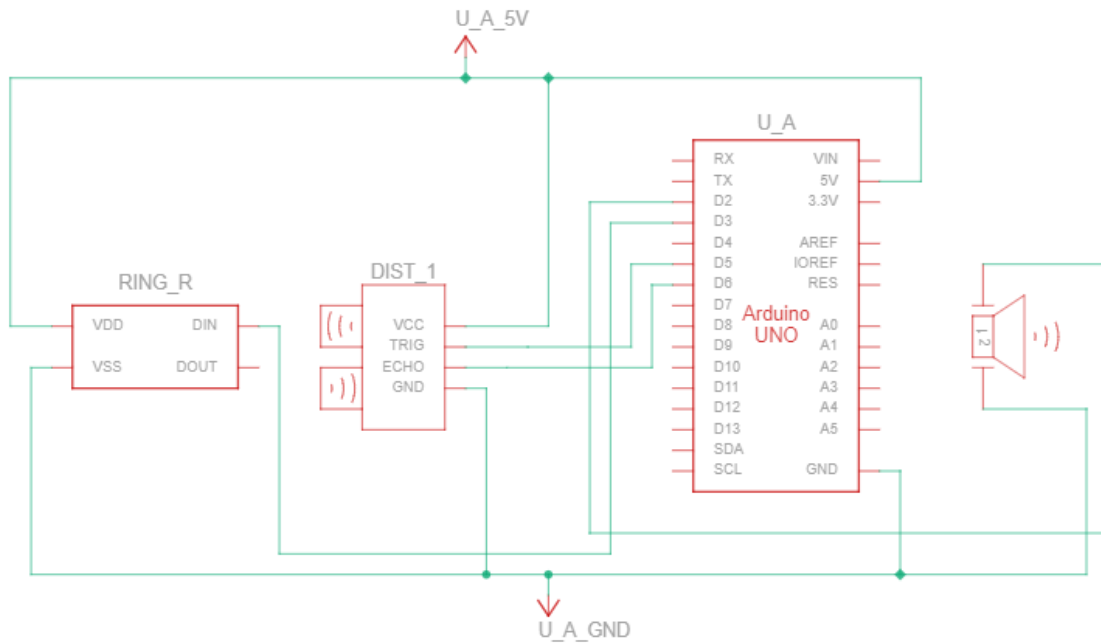
S.no	Component	Quantity
1.	Arduino UNO R3	01
2.	Ultrasonic Distance Sensor	01
3.	NeoPixel Ring 12	01
4.	Piezo	01
5.	Breadboard	01
6.	Connecting Wires	As per need

BLOCK DIAGRAM



CIRCUIT DIAGRAM

[View Project here](#)



Working

This Project uses Arduino UNO R3. This works using the SONAR which detects how far the object is from the source. Accordingly, Arduino lights up more lights in the NeoPixel ring using the logic from the code fitted inside the Arduino. Ultrasonic sensors send waves. These waves are absolutely invisible and come back after hitting an optical.

Input to NeoPixel Ring is provided by the D3 terminal of the Arduino and the other two terminals are connected to power(5V) and the ground terminal.

Trigger and Echo Terminal of SONAR are connected to D5 and D6 respectively, and the power and grounding are provided by Arduino

The D2 pin is connected to the piezo(buzzer) and the other terminal of the buzzer is connected to the ground (Power Section of Arduino).

IMPLEMENTATION

```
// Social Distancing Alert JIIT
#include <Adafruit_NeoPixel.h>
// Which pin on the Arduino is connected to the NeoPixels?
int ledPin= 3;
// How many NeoPixels are attached to the Arduino?
int ledNo= 12;
// Declare our NeoPixel strip object:
// When we set up the NeoPixel library, we tell it how many pixels, and which
pin to use to send signals.
Adafruit_NeoPixel strip = Adafruit_NeoPixel(ledNo,ledPin,NEO_RGB+NEO_KHZ800);
// which pin on the Arduino is connected to buzzer
int buzzerPin= 2;
//which pin on the Arduino is connected to echoPin
int echoPin= 6;
//which pin on the Arduino is connected to trigPin
int trigPin= 5;
//setting the minimum and maximum distance
```

```
int minDistance = 100;
int maxDistance = 300;
```

LED No. is the total number of LEDs to be lit up, and D3 pin is the led pin. The buzzer pin is represented by D2. The trigger and echo pins are D5 and D6, respectively. An input pin is a trigger pin. This pin must be held high for 10 microseconds in order to begin measuring by transmitting an Ultrasonic wave. An output pin is an echo pin. This pin goes high for the amount of time it takes the Ultrasonic wave to return to the sensor.

This function setup() performs the initialization and turns off the led. The loop function calculates the object's distance from the source and then maps it into 12 components, which determines how many LEDs should be lit up to show the severity of danger at a certain distance. The LEDs are then assigned the proper colour, and the other LEDs are turned off.

```
void setup()
{
  pinMode(buzzerPin, OUTPUT); //sets the buzzerPin as output
  pinMode(trigPin, OUTPUT); //sets the trigPin as output
  pinMode(echoPin, INPUT); //sets the buzzerPin as output
  Serial.begin(9600); // starts the serial communication
  strip.begin(); // prepares data pin for Neopixel output
  for(int i = 0; i < ledNo; i++)
  {
    // setting the led colour black
    strip.setPixelColor(i, strip.Color(0,0,0));
  }
  strip.show(); // initialise all pixels to "off"
}

void loop()
{
  int distance = calcDistance();
  Serial.println(distance);
  int ledsToGlow = map(distance, minDistance, maxDistance, ledNo, 1);
  Serial.println(ledsToGlow);
  if(ledsToGlow == 12)
  {
```

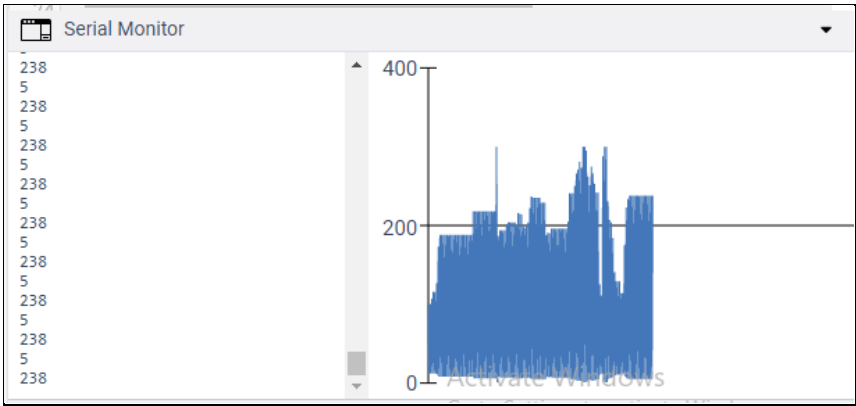
```
digitalWrite(buzzerPin, HIGH);
}
else
{
    digitalWrite(buzzerPin, LOW);
}
for(int i = 0; i < ledsToGlow; i++)
{
    if(i < 4)
    {
        strip.setPixelColor(i, strip.Color(50,0,0)); //green, red, blue
    }
    else if(i >= 4 && i < 8)
    {
        strip.setPixelColor(i, strip.Color(50,50,0)); //green, red, blue
    }
    else if(i >= 8 && i < 12)
    {
        strip.setPixelColor(i, strip.Color(0,50,0)); //green, red, blue
    }
}
for(int i = ledsToGlow; i < ledNo; i++)
{
    strip.setPixelColor(i, strip.Color(0,0,0));
}
strip.show();
delay(50);
}

int calcDistance()
{
    long distance, duration;
    digitalWrite(trigPin, LOW);
    delayMicroseconds(2);
```

```
digitalWrite(trigPin, HIGH);
delayMicroseconds(10);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = duration/29/2;
if(distance >= maxDistance)
{
    distance = maxDistance;
}
if(distance <= minDistance)
{
    distance = minDistance;
}
return distance;
}
```

This function returns the distance of the object from the source. The trigger sends a ping. The echo listens for an echo that comes back to calculate the distance. This is measured in microseconds. Sound travels at 343 metres per second, which means it needs 29.155 microseconds per centimetre. So, we have to divide the duration by 29 and then by 2, because the sound has to travel the distance twice. It travels to the object and then back to the sensor. Then finally it returns the distance calculated but if it lies beyond the range it returns the same minimum and maximum distance respectively

Fig: Serial Monitor



RESULT

The distance of the object from the source is varied and at various regions, NeoPixel light gives varying outputs.

The NeoPixel Ring shows the intensity of closeness from another person and if the distance is approx 100units, the buzzer alerts by producing sound.

REFERENCES

1. [NeoPixel: How to Control WS2812 RGB LED w/ Arduino](#)
2. [Tinkercad - Arduino](#)
3. [Ultrasonic Distance Sensor in Arduino With Tinkercad : 6 Steps \(with Pictures\) - Instructables](#)
4. <https://all3dp.com/2/tinkercad-arduino-how-to-design-simulate-circuits/>

