Programming for Internet of Things Boards

BCT3007

J COMPONENT FINAL REPORT

Submitted To

Prof. Narayanamoorthi M

TOPIC : **COVID-19 SOCIAL DISTANCING MONITORING IOT DEVICE**

Team Members :

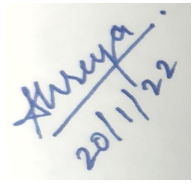
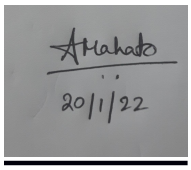
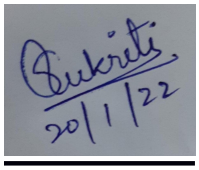
SL NO	Register No	Name	Slot	Email id	Signature with date
1	19BCT0133	SHREYA GODAWAT	C1	shreya.godawat2019@vitstudent.ac.in	
2	19BCT0173	ARKAPRAVA MAHATO	C1	arkaprava.mahato2019@vitstudent.ac.in	
3	19BCT0184	SUKRITI KUMAR	C1	sukriti.kumar2019@vitstudent.ac.in	

Table of Contents

ACKNOWLEDGEMENT

We wish to express our sincere thanks and deep sense of gratitude to our project guide,Narayanamoorthi M Sir for his consistent encouragement and valuable guidance offered to us in a pleasant manner throughout the course of the project work.

We also take this opportunity to thank all the faculty of the School of Computer Science & Engineering for their support and their wisdom imparted to us throughout the course. We thank our parents, family, and friends for bearing with us throughout the course of our project and for the opportunity they provided us in undergoing this course in such a prestigious institution.

And lastly and most importantly we would like to thank Vellore Institute of Technology, Vellore, Tamil Nadu for providing us with the opportunity and facilities to carry out this project.

ABSTRACT

In this current scenario of global outbreak, it is advised by WHO (World Health Organization) to maintain a distance among the people to avoid the spread of the SARS-CoV-2 or commonly known as the CoronaVirus. However, it is quite difficult to stay indoors completely as people need daily essentials to keep going.

The idea of this project is to make a device that warns people with the use of a Buzzer and LEDs when they come within a certain proximity. The way we can observe any infringement in social distancing is by observing the glowing red light whenever the distance is less than 6 feet.

The way we achieve this is we send data from arduino to thingspeak using ESP8266 (Wifi module) which communicates with thingspeak from where data can be collected regarding the distances and an output can be displayed. To display the result, we have built an app.

The device could be fitted in a bag or worn as a badge. Additionally, it could be placed in shop counters and pharmacies where a servo motor is fitted to ensure the sensor covers a total 180 degrees view to ensure full safety.

OBJECTIVE

Based on this approach, in the proposed work the wearable device is suggested. This device will help each individual to maintain at least social distancing. Due to this, the spread of the COVID-19 virus will minimize. In the suggested device, the ULTRASONIC sensor interfaced with Arduino which will detect the human presence. If human presence is there, then it will notify the individual through the audio message. Hence, the individual will be alerted and will maintain the social distance at the public place also. By wearing this device, any person will come to know the presence of a human being nearby him/her. Due to that, it will be helpful to keep social distance to avoid the spread of the COVID-19 virus.

LITERATURE SURVEY -

1. **Novel economical social distancing smart device for COVID19**

   Rahul Reddy Nadikattu, Sikender Mohsienuddin Mohammad and Pawan Whig, Novel Economical Social Distancing Smart Device for Covid19. International Journal of Electrical Engineering and Technology, 11(4), 2020

   In this paper they proposed a method to track humans' position in an outdoor environment based on sensors.With the help of artificial intelligence, this novel smart device is handy for maintaining a social distancing as well as detecting COVID 19 symptom patients and thereby safety.

2. **Social Distance Monitoring Approach Using Wearable Smart Tags**
   Alhmiedat, T.; Aborokbah, M. Social Distance Monitoring Approach Using Wearable Smart Tags. Electronics 2021, 10, 2435

   In this paper they proposed a smart monitoring system consists of a new smart wearable prototype of a compact and low-cost electronic device, based on human detection and proximity distance functions, to estimate the social distance between people and issue a notification when the social distance is less than a predefined threshold value.

3. **Social Distancing Sensor: Devices that Use Ultrasound and Radio Frequency Communication to Facilitate Social Distancing**

   Malik, Namya, "Social Distancing Sensor: Devices that Use Ultrasound and Radio Frequency Communication to Facilitate Social Distancing" (2020).

   Proposed a system that consists of multiple devices that communicate with each other through both ultrasonic (US) signals and radio frequency (RF) signals. It was not completely accurate, but their prototype shows successful proof-of-concept and the devices demonstrate successful bi-directional ultrasound and radio-frequency communication.

4. **The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission**

   Sun, C.; Zhai, Z. The efficacy of social distance and ventilation effectiveness in preventing COVID-19 transmission. Sustain. Cities Soc. 2020

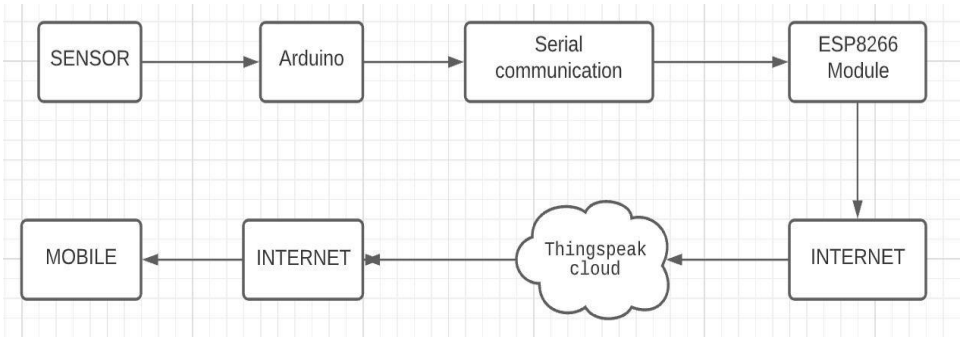
   This paper developed and introduces two critical indices – social distance probability Pd and ventilation effectiveness Ez – into the WellsRiley model to predict the infection probability of COVID-19. These two indices provide the quantitative evaluation of impacts of social distancing and ventilation effectiveness on respiratory disease infection risk.

5. **Monitoring Social Distancing by Smart Phone App in the Effect of COVID-19**

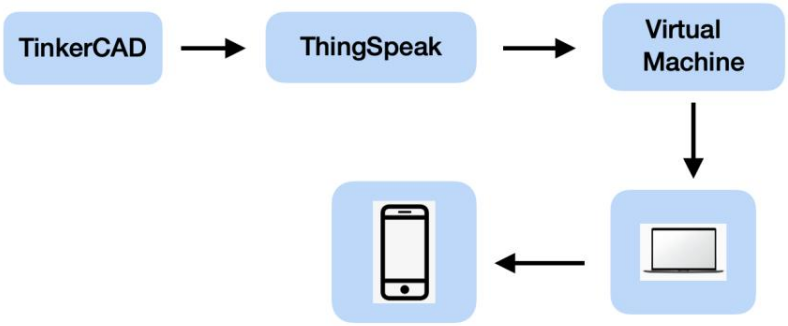
   Neelavathy Pari, S.; Vasu, B.; Geetha, A.V. Monitoring Social Distancing by Smart Phone App in the effect of COVID-19. Glob. J. Comput. Sci. Technol. 2020

   The paper presents two methods of monitoring social distancing for smartphone users. In camera surveillance method, pedestrian detection is implemented using deep learning algorithms and distance between two persons using euclidean formula. In Bluetooth distance calculation, calculated the real time distance by receiving its signal strength between two android devices.

PROPOSED WORK



ARCHITECTURE



HARDWARE AND SOFTWARE REQUIREMENTS

1. TinkerCAD (software used for simulation)
2. Arduino Uno
3. ESP8266 wifi module
4. Ultrasonic Sensor
5. Buzzer
6. LED
7. Mini Servo motor
8. ThingSpeak Cloud
9. Android studio
10. Flutter
11. Android x86

IMPORTANCE OF EACH COMPONENTS

Arduino UNO - Arduino boards are able to read inputs like light on a sensor, a finger on a button, or a Twitter message - and turn it into an output - activating a motor, turning on an LED, publishing something online,etc. You can tell your board what to do by sending a set of instructions to the microcontroller on the board. To do so you use the Arduino programming language (based on Wiring), and the Arduino Software (IDE), based on Processing

ESP8266 Wifi Module - The ESP8266 WiFi Module is a self contained SOC with integrated TCP/IP protocol stack that can give any microcontroller access to your WiFi network. The ESP8266 is capable of either hosting an application or offloading all Wi-Fi networking functions from another application processor.

Ultrasonic Sensors - An ultrasonic sensor is an electronic device that measures the distance of a target object by emitting ultrasonic sound waves, and converts the reflected sound into an electrical signal. Ultrasonic waves travel faster than the speed of audible sound (i.e. the sound that humans can hear).

Servo Motor – A mini servo is a motor that converts electrical signals into mechanical motion. It is a rotary or a linear actuator capable of turning to a predetermined position, exactly as commanded.

Buzzer- A device that makes beeping noises. Here, a piezoelectric buzzer is being used which generates a clicking noise when a certain voltage is applied.

SENSOR USED

The sensor used by us for the project was an HC-SR04 Ultrasonic sensor.

HC-SR04 Sensor Features

1. Operating voltage: +5V
2. Theoretical Measuring Distance: 2cm to 450cm
3. Practical Measuring Distance: 2cm to 80cm,Accuracy: 3mm
4. Measuring angle covered: <15°,Operating Current: <15mA
5. Operating Frequency: 40Hz

WORKING OF THE SENSOR

The Ultrasonic sensor works on the principle of Ultrasound.

Ultrasound is high-pitched sound waves with frequencies higher than the audible limit of human hearing. Humans have a hearing range of 20Hz to 20kHz, any sound higher than 20kHz is known as Ultrasound .Typically an ultrasonic sensor consists of a transducer that converts electrical signal to 40kHz and emits it in the form of pulses.

At its core, the Ultrasonic distance sensor consists of two ultrasonic transducers. The one acts as a transmitter which converts electrical signal into 40 KHz ultrasonic sound pulses.

The receiver listens for the transmitted pulses. If it receives them it produces an output pulse whose width can be used to determine the distance the pulse traveled.

A pulse of at least 10 µS (10 microseconds) in duration is applied to the Trigger pin. In response to that the sensor transmits a sonic burst of eight pulses at 40 KHz. This 8-pulse pattern makes the "ultrasonic signature" from the device unique, allowing the receiver to differentiate the transmitted pattern from the ambient ultrasonic noise.

The eight ultrasonic pulses travel through the air away from the transmitter. Meanwhile the Echo pin goes HIGH to start forming the beginning of the echo-back signal.

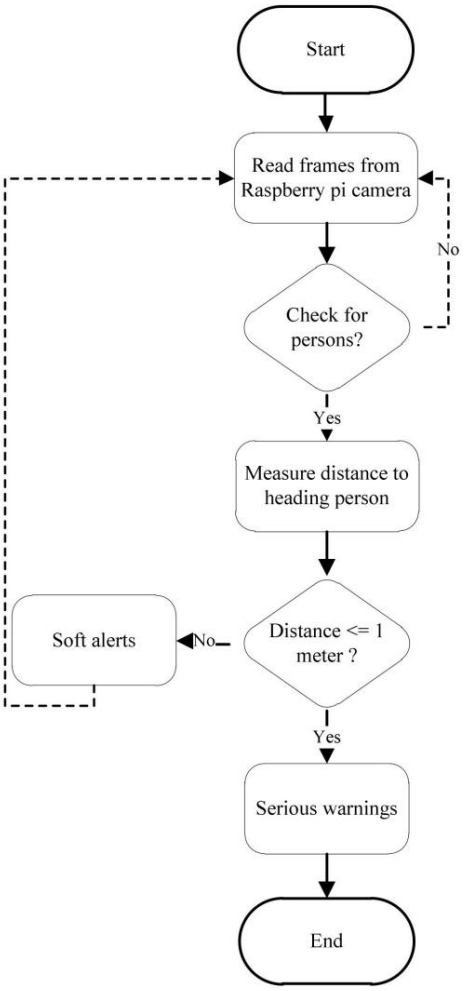
In case, If those pulses are not reflected back then the Echo signal will timeout after 38 mS (38 milliseconds) and return low. Thus a 38 mS pulse indicates no obstruction within the range of the sensor.

If those pulses are reflected back the Echo pin goes low as soon as the signal is received. This produces a pulse whose width varies between 150 µS to 25 mS, depending upon the time it took for the signal to be received.

Distance is measured as,

$$\text{Distance} = (\text{Speed of sound in air} \times \text{Time})/2$$

FLOWCHART



DESIGN AND WORKING PRINCIPLE

1. Design :

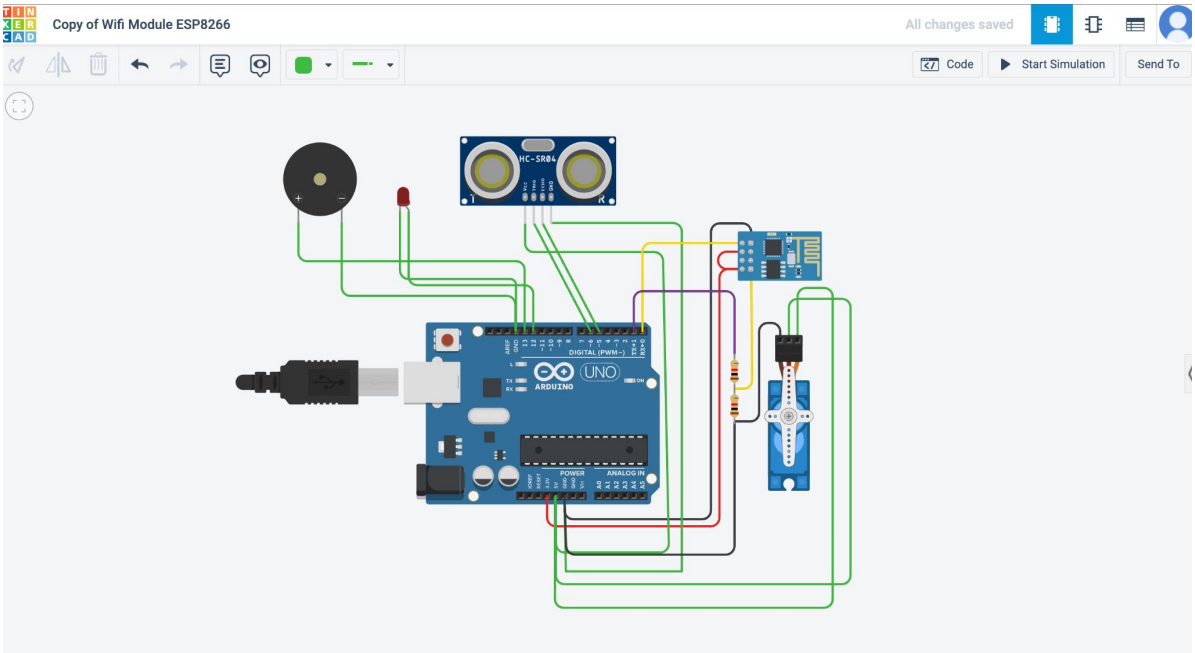
The design procedure is quite simple. The trigger pin of the ultrasonic sensor was connected to pin 6, echo pin was connected to 5, the buzzer was connected to pin 13 and the LED was connected to pin 12 of the Arduino board. The other parts were grounded while the Vcc pin of the ultrasonic sensor was connected to the 5V power supply pin of the board. A data cable was connected to the arduino which

was connected to the laptop to load the code into the microcontroller and give power supply to the system.

2. Working :

- The Ultrasonic emits and receives a signal to detect an object nearby every 1ms.
- If any object is present at distance < 1m
- A buzzer is triggered as well as an LED
- If not, the Buzzer and Led is set to low
- The whole process repeats after 60ms

Circuit diagram :



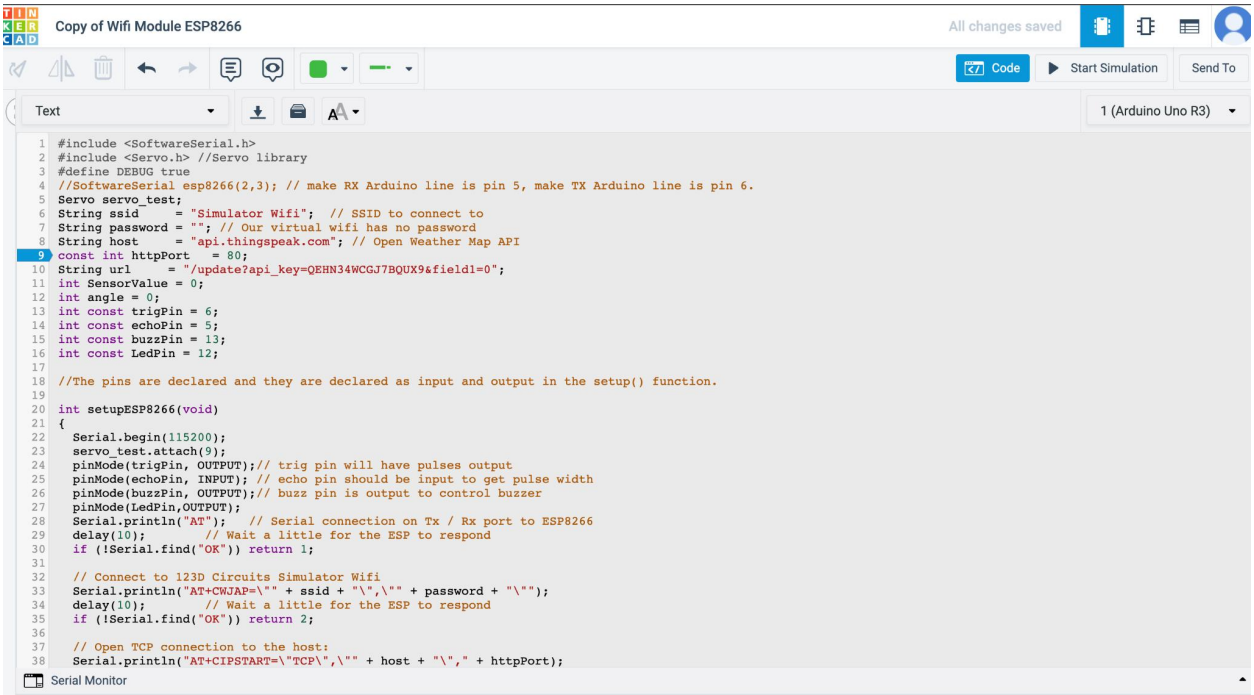
IMPLEMENTATION

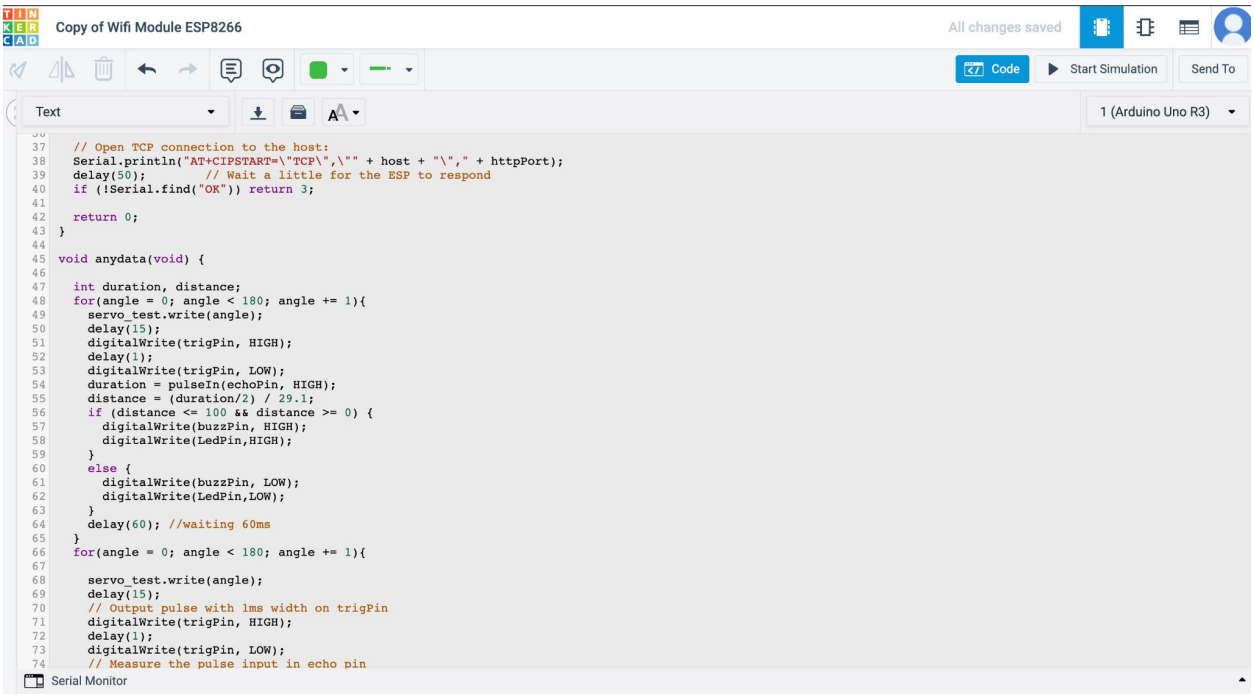
Brief explanation of the code :

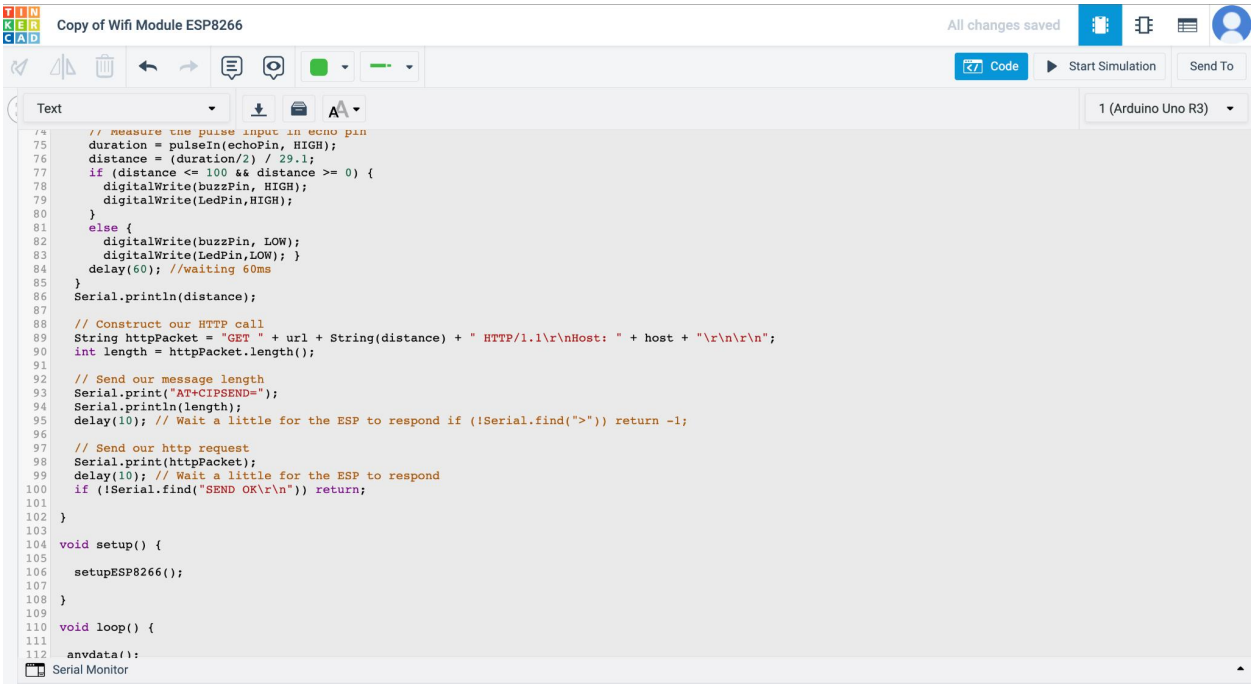
- The pins are declared and they are declared as input and output in the setup() function.
- The echoPin is input because it measures the Ultrasonic waves from the surroundings and the
- Buzzer,Led and servo are declared as output ports.
- The loop function runs continuously and the servo motor rotates from 0 to 180 degrees.
- If the distance measured by the Ultrasonic sensor is less than 1m, the buzzer and LED is switched to HIGH voltage or switched ON.
- If the distance is more than 1m, the buzzer and LEDs are switched OFF or set to LOW voltage.
- The same process repeats from 180 degrees to 0 and thus the entire field is covered.

CODE

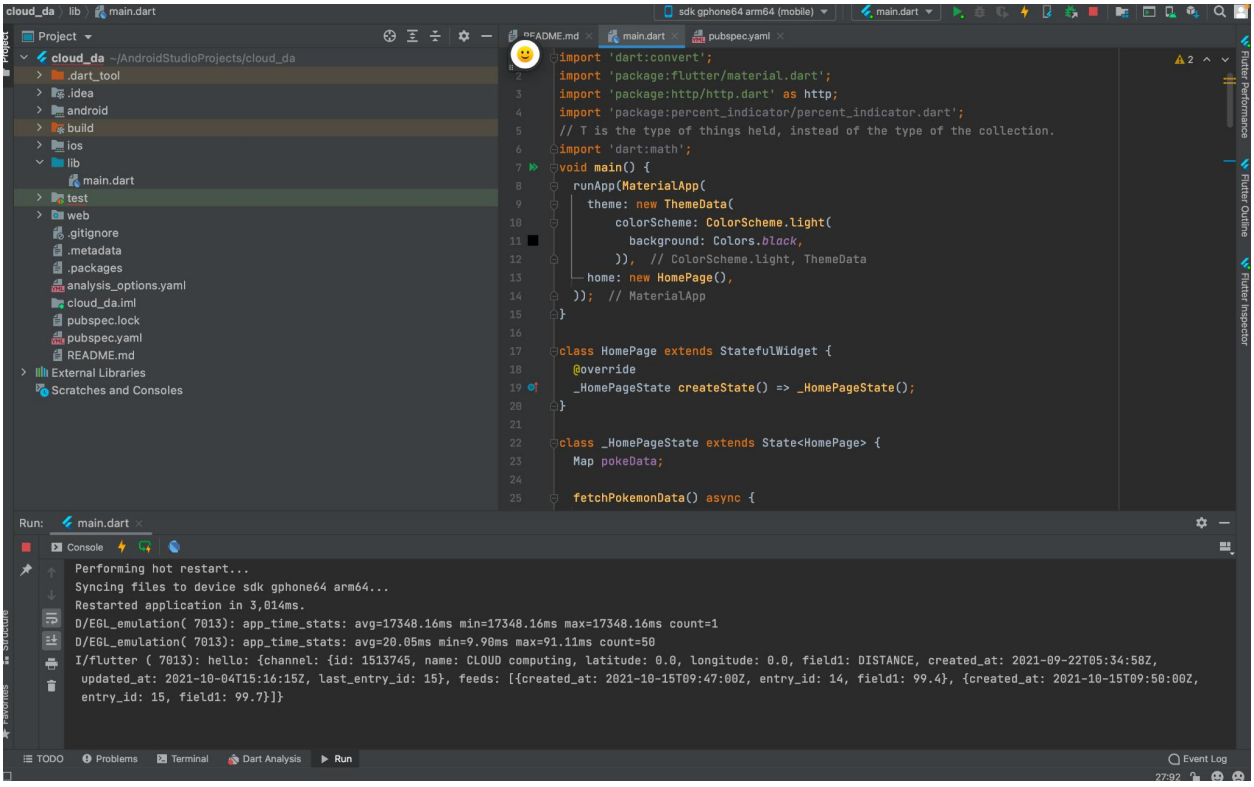
TINKERCAD -

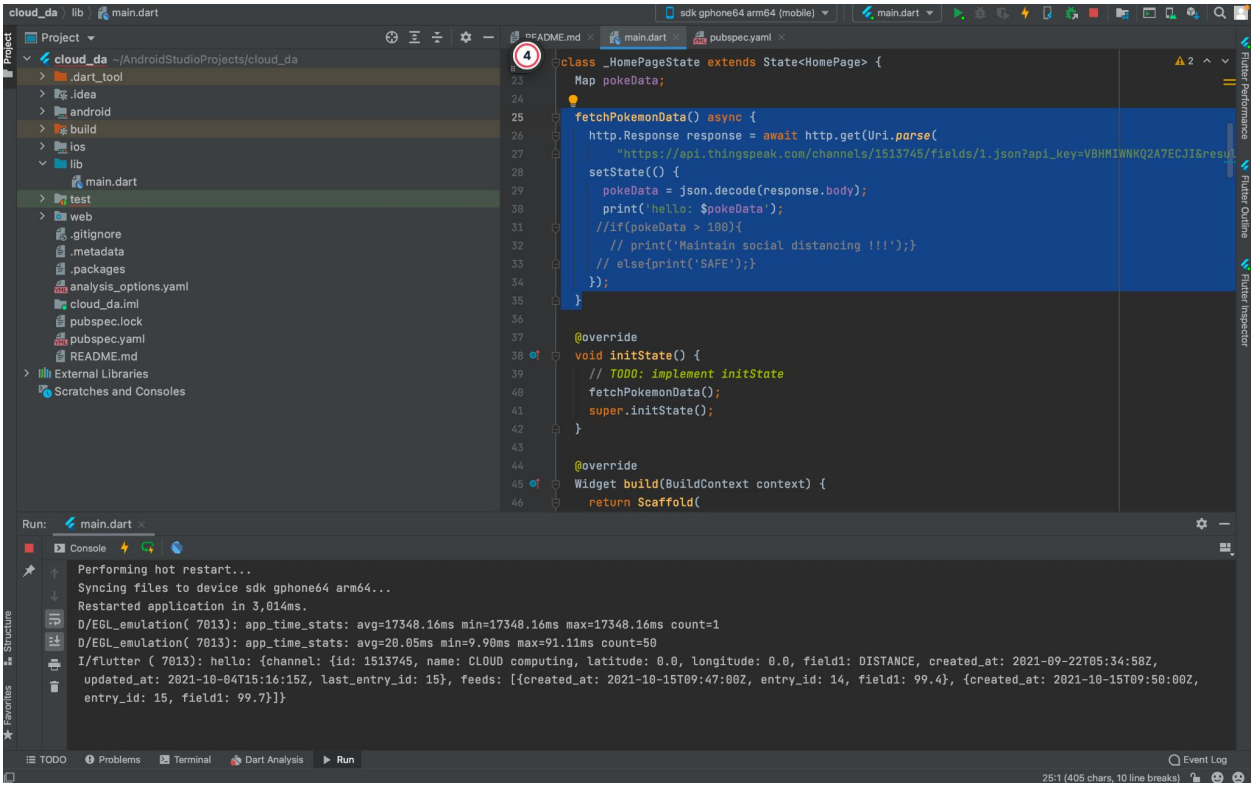


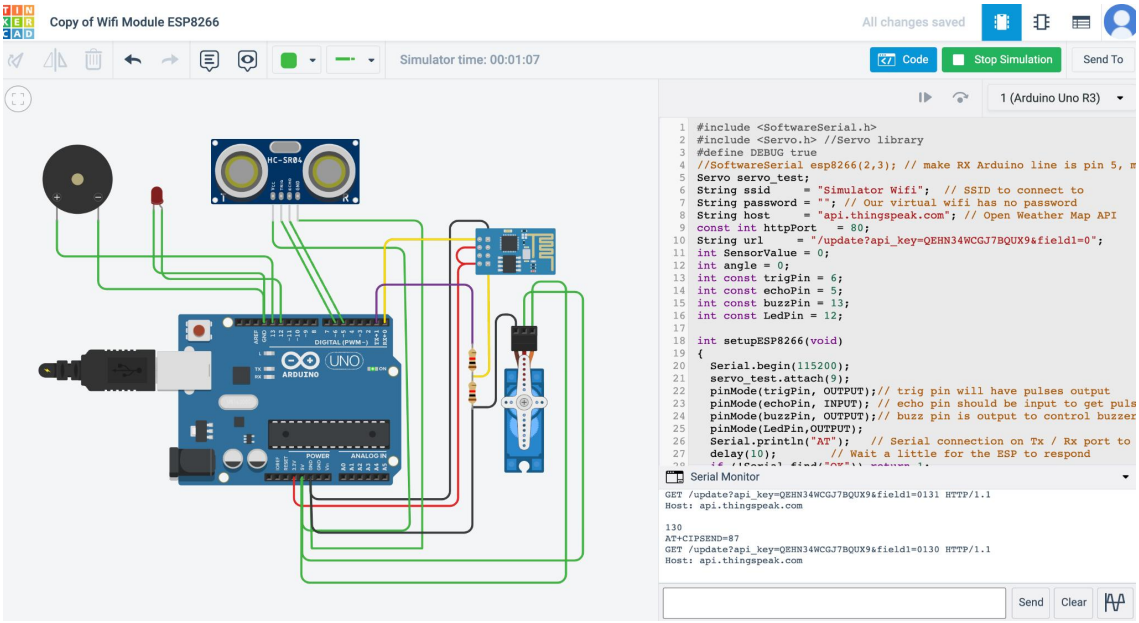

Copy of Wifi Module ESP8266
All changes saved
Code
Start Simulation
Send To
Text
1 (Arduino Uno R3)
// Open TCP connection to the host:
Serial.println("AT+CIPSTART=\"TCP\",\"" + host + "\"," + httpPort);
delay(50); // Wait a little for the ESP to respond
if (!Serial.find("OK")) return 3;
return 0;
}
void anydata(void) {
int duration, distance;
for(angle = 0; angle < 180; angle += 1){
servo_test.write(angle);
delay(15);
digitalWrite(trigPin, HIGH);
delay(1);
digitalWrite(trigPin, LOW);
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance <= 100 && distance >= 0) {
digitalWrite(buzzPin, HIGH);
digitalWrite(LedPin,HIGH);
}
else {
digitalWrite(buzzPin, LOW);
digitalWrite(LedPin,LOW);
}
delay(60); //waiting 60ms
}
for(angle = 0; angle < 180; angle += 1){
servo_test.write(angle);
delay(15);
// Output pulse with 1ms width on trigPin
digitalWrite(trigPin, HIGH);
delay(1);
digitalWrite(trigPin, LOW);
// Measure the pulse input in echo pin
Serial Monitor


Copy of Wifi Module ESP8266
All changes saved
Code
Start Simulation
Send To
Text
1 (Arduino Uno R3)
duration = pulseIn(echoPin, HIGH);
distance = (duration/2) / 29.1;
if (distance <= 100 && distance >= 0) {
digitalWrite(buzzPin, HIGH);
digitalWrite(LedPin,HIGH);
}
else {
digitalWrite(buzzPin, LOW);
digitalWrite(LedPin,LOW); }
delay(60); //waiting 60ms
}
Serial.println(distance);
// Construct our HTTP call
String httpPacket = "GET " + url + String(distance) + " HTTP/1.1\r\nHost: " + host + "\r\n\r\n";
int length = httpPacket.length();
// Send our message length
Serial.print("AT+CIPSEND=");
Serial.println(length);
delay(10); // Wait a little for the ESP to respond if (!Serial.find(">")) return -1;
// Send our http request
Serial.print(httpPacket);
delay(10); // Wait a little for the ESP to respond
if (!Serial.find("SEND OK\r\n")) return;
}
void setup() {
setupESP8266();
}
void loop() {
anydata();
Serial Monitor

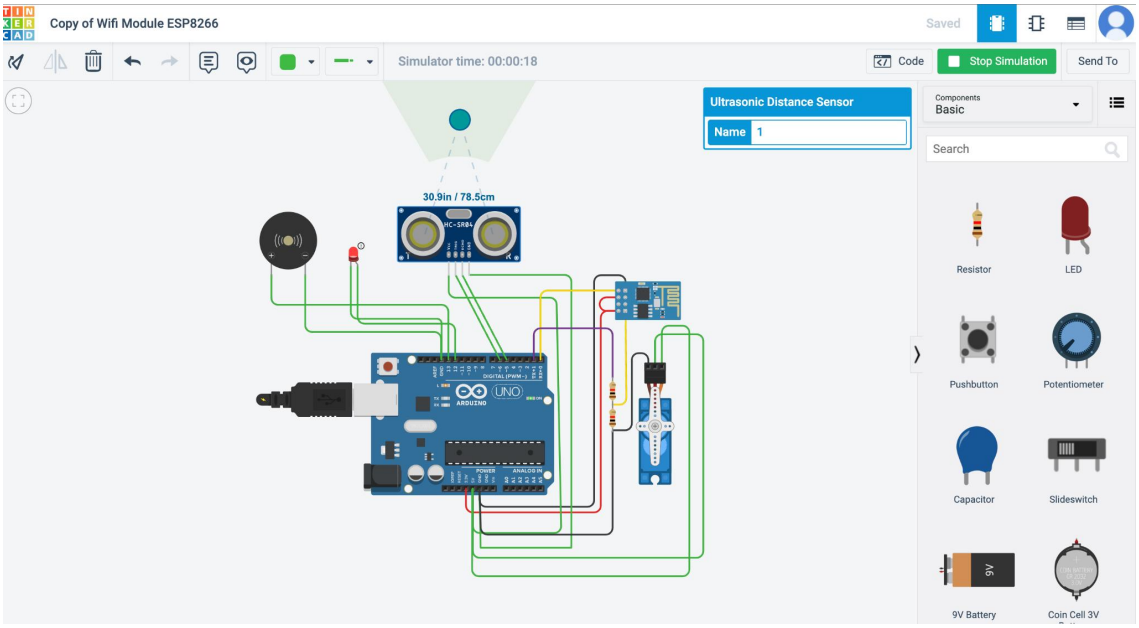
ANDROID STUDIO -

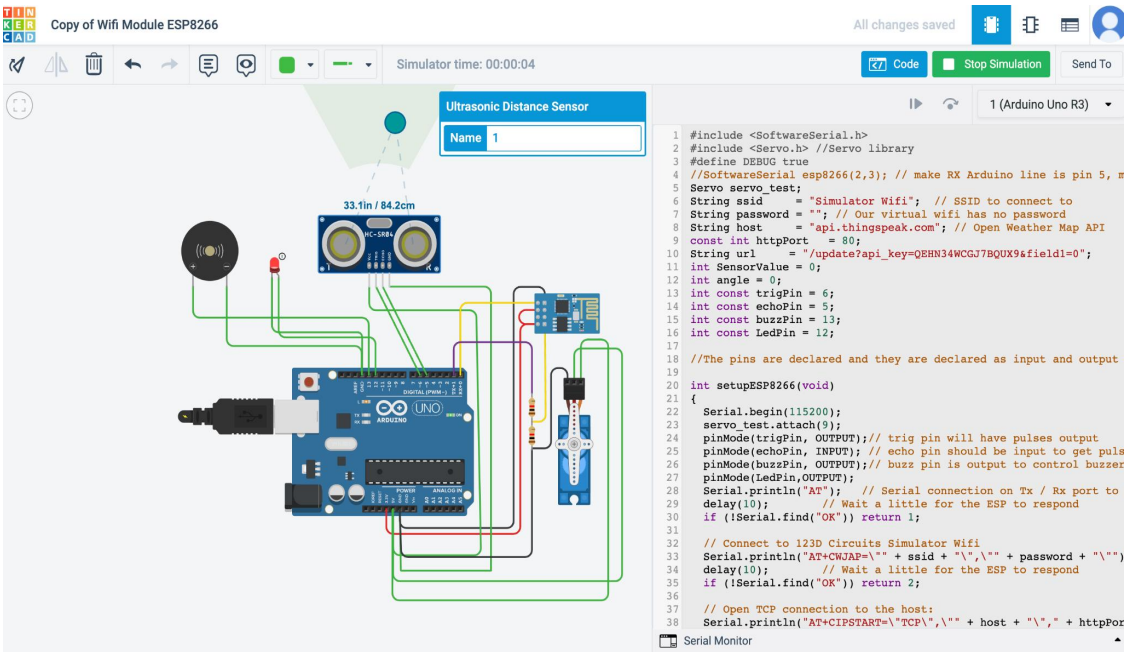




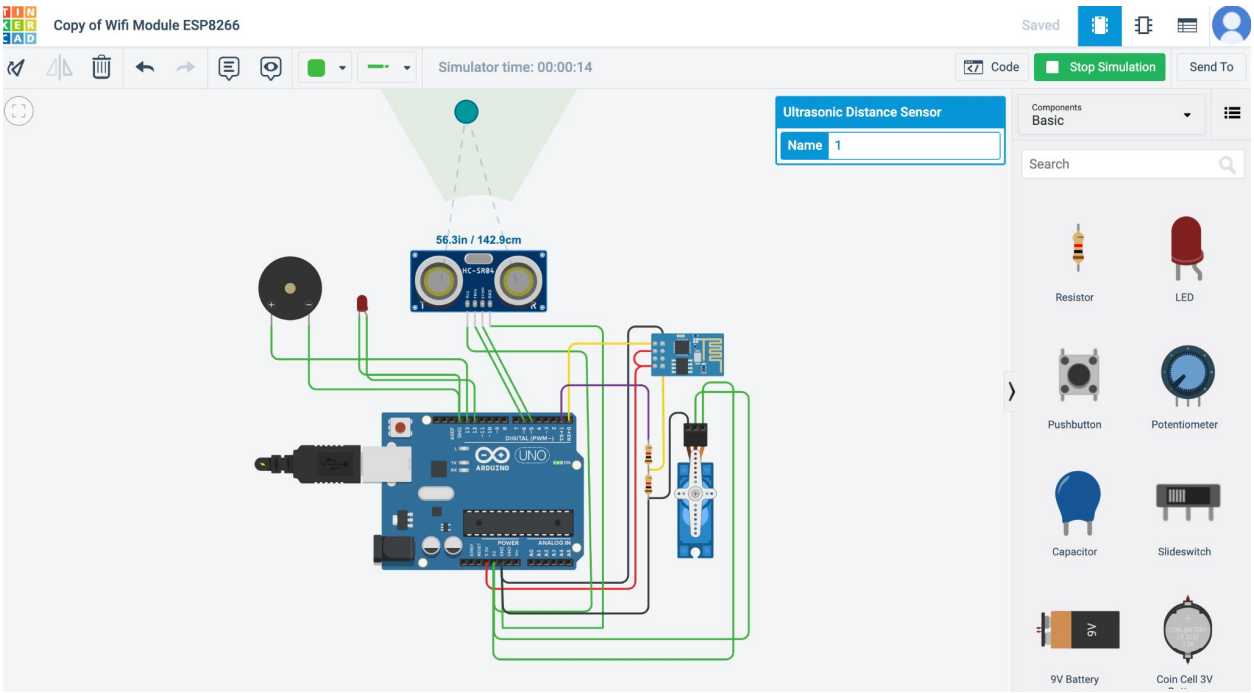


The distance is less than 1m therefore the buzzer and LEDs are switched ON -

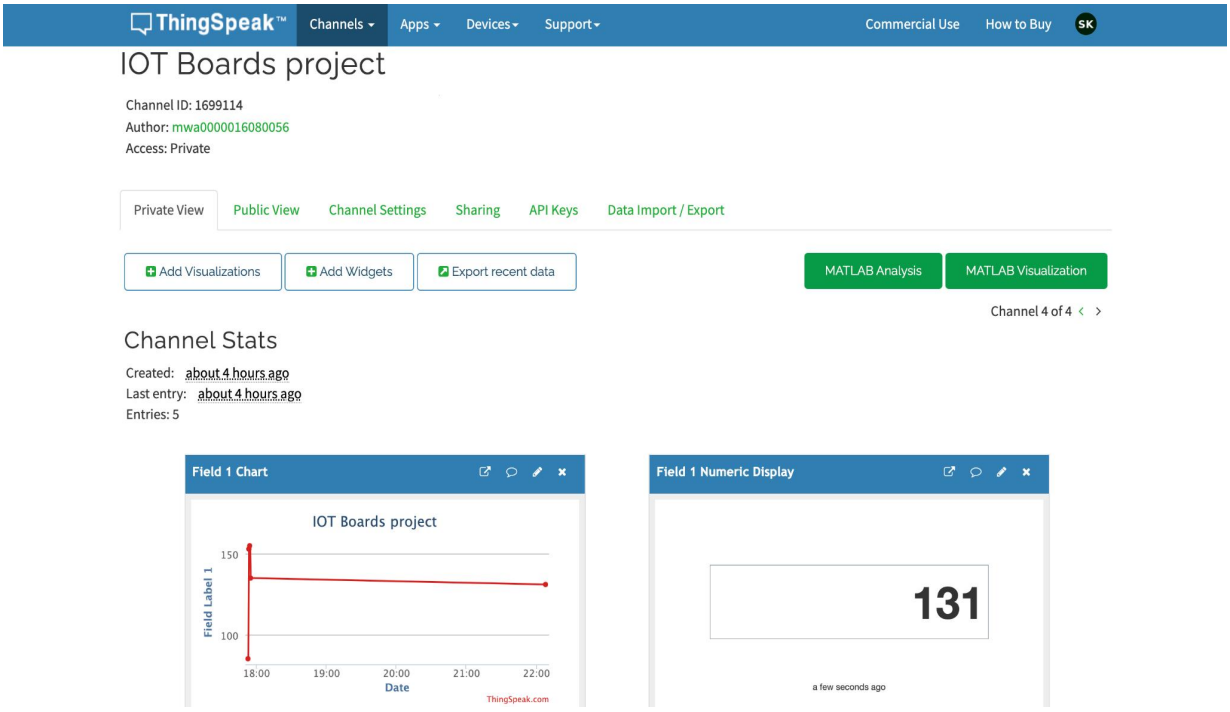


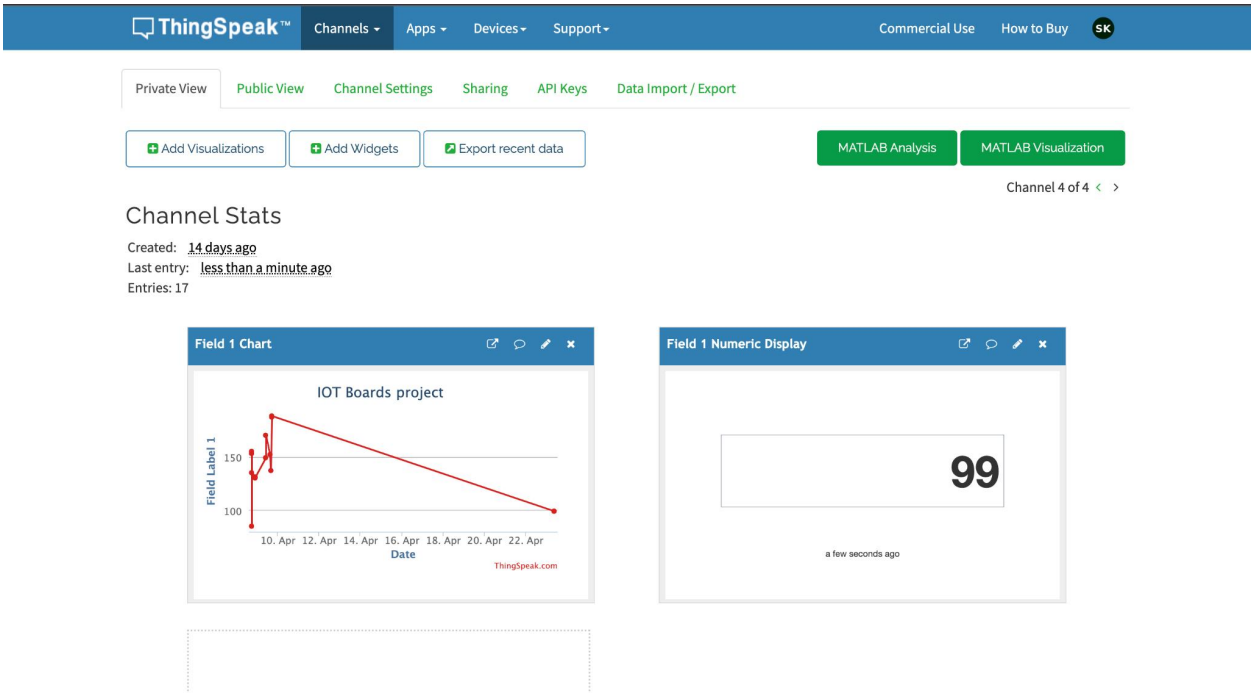


The distance is more than 1m therefore the buzzer and LEDs are switched OFF -



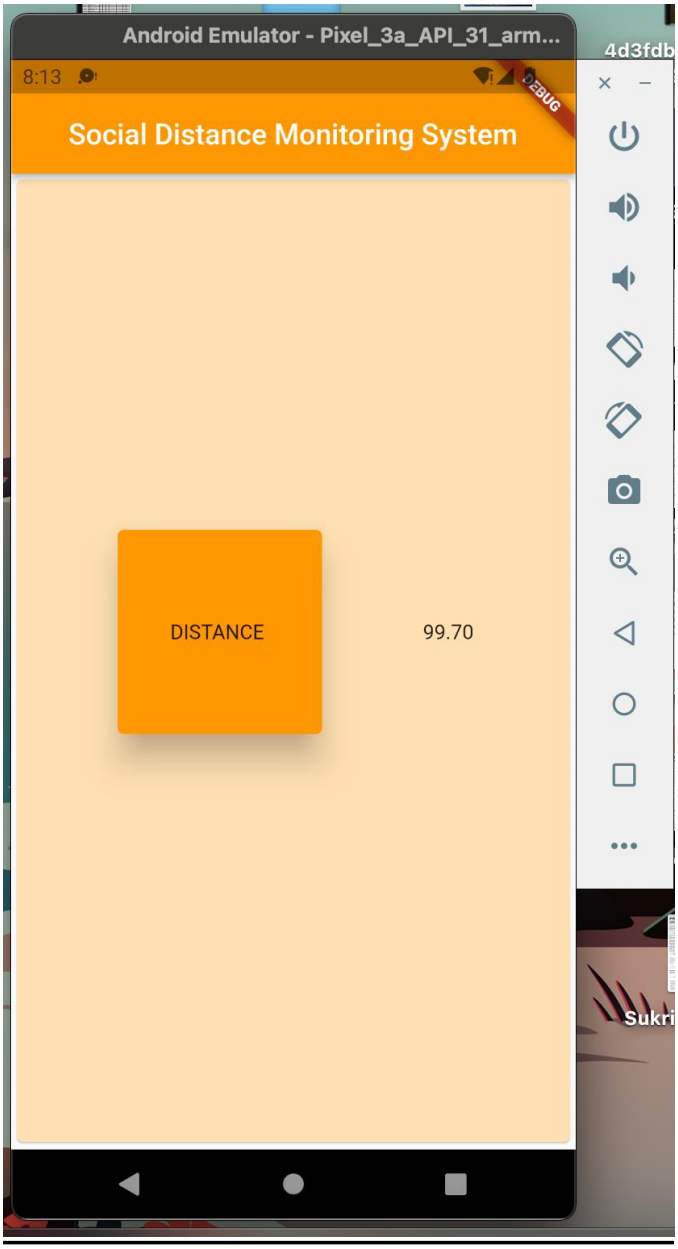
THINGSPEAK CLOUD (DATA STORAGE) -





RESULTS

Alert notification in APP



CONCLUSION

Given that the main purpose of a social distancing device is to detect the distance from people, the perceived severity of and vulnerability to SARS-CoV-2 infection may also be positively related to the motivation for using such an application by organizations in view of public safety.

REFERENCES

- https://jesit.springeropen.com/articles/10.1186/s43067-021-00040-z
- https://osf.io/s3pfx/
- https://www.mdpi.com/2079-9292/10/19/2435/htm
- https://www.allaboutcircuits.com/projects/servo-motor-control-with-an-arduino/
- https://www.arduino.cc/en/Guide/introduction
- https://docs.arduino.cc/learn/electronics/servo-motors