COVID ’19 AUTOMATED SCREENING AND SANITISATION MACHINE

Project Done by:

Name: Sagnik Nayak

Contact Number: +91 7980027546

Email ID: sagniknayak2608@gmail.com

Institution: University of Calcutta

Semester: III

Address: Opposite to Mangal Chandi Mandir, South Subhash Palli, Dankuni, Hooghly, West Bengal, India - 712311

*This Project was made as the final project for a 7-day online program on Arduino IDE organised by Ministry of Biotechnology, Govt. of India in association with Deen Dayal Upadhyaya College, University of Delhi

CONTENTS

Sl. No	Topic	Page No.
1	The Problem Statement	3
2	COVID ’19 Automated Thermal Screening and Sanitisation	4-15
2.1	Abstract	4
2.2	Introduction	4
2.3	Hardware Components required	5
2.4	Circuit Diagram with labelling	6
2.5	Circuit Diagram without labelling	6
2.6	Functionality	7
2.7	Source Code	7
2.8	Innovative points about the project	14
2.9	Links	14
2.10	Scopes for improvement	14

THE PROBLEM STATEMENT

Title: COVID '19 Screening Machine

Aim: To automate the Temperature based Screening Task for COVID '19 Screening.

Problem Statement: You have to make a Prototype Simulation Model (in TinkerCAD) of Temperature based screening machine for COVID '19. This machine is useful at entry gates where manual screening is done by persons using Digital Thermometers. We want you to automate that particular task only in an effective way.

Features:

- Detect and alerts whenever a person is in between the range of 10cm to 20 cm in front of the machine.
- Measures his/her temperature.
- It should measure temperature only when someone is detected and not all time.
- If the temperature is not normal then the buzzer & RED LED should be turned on to alert Gatekeeper.
- If the temperature is normal then it should turn on the green LED.
- It should spray sanitiser whenever hands are placed below a knob/a fixed point.
- Display all relevant things over the LCD & Serial Monitor simultaneously.
- Count of the number of persons currently inside should be maintained and displayed properly.

COVID ’19 Automated Thermal Screening and Sanitisation

3.1 Abstract:

The objective of this project is to automate the process of thermal screening and sanitisation before entering public places, which has been done recently due to COVID ’19. This project would limit human-to-human interaction during the screening and sanitisation which could be a reason for COVID ’19 spread, minimalizing the effect.

Arduino Uno R3 (ATmega328P) and peripheral components like sensors and actuators have been used to accomplish this task. The project has utilised component that are available at a very low cost and can be easily set up, making it affordable and deployable at all scales, from small businesses to large companies and at public places like cinema halls, malls and airports. In this report, we provide the information regarding functionality, features and simulation model (using TinkerCAD) of the project.

3.2 Introduction:

The entire world has been affected by SARS-CoV-2 which had resulted in a lockdown in many countries due to it nature of spreading through droplets. As situations normalise and as lockdowns are lifted gradually with passage of time, various public places are also made open but to fight the spread of COVID, thermal screening and sanitisation has been made mandatory in all places across countries, to reduce the risk of infection as much as possible. This screening and sanitisation are done by people as they screen every person using a thermal screening device and providing sanitiser. This puts the ones who are checking to be at a high risk of being infected by the one being checked and vice versa.

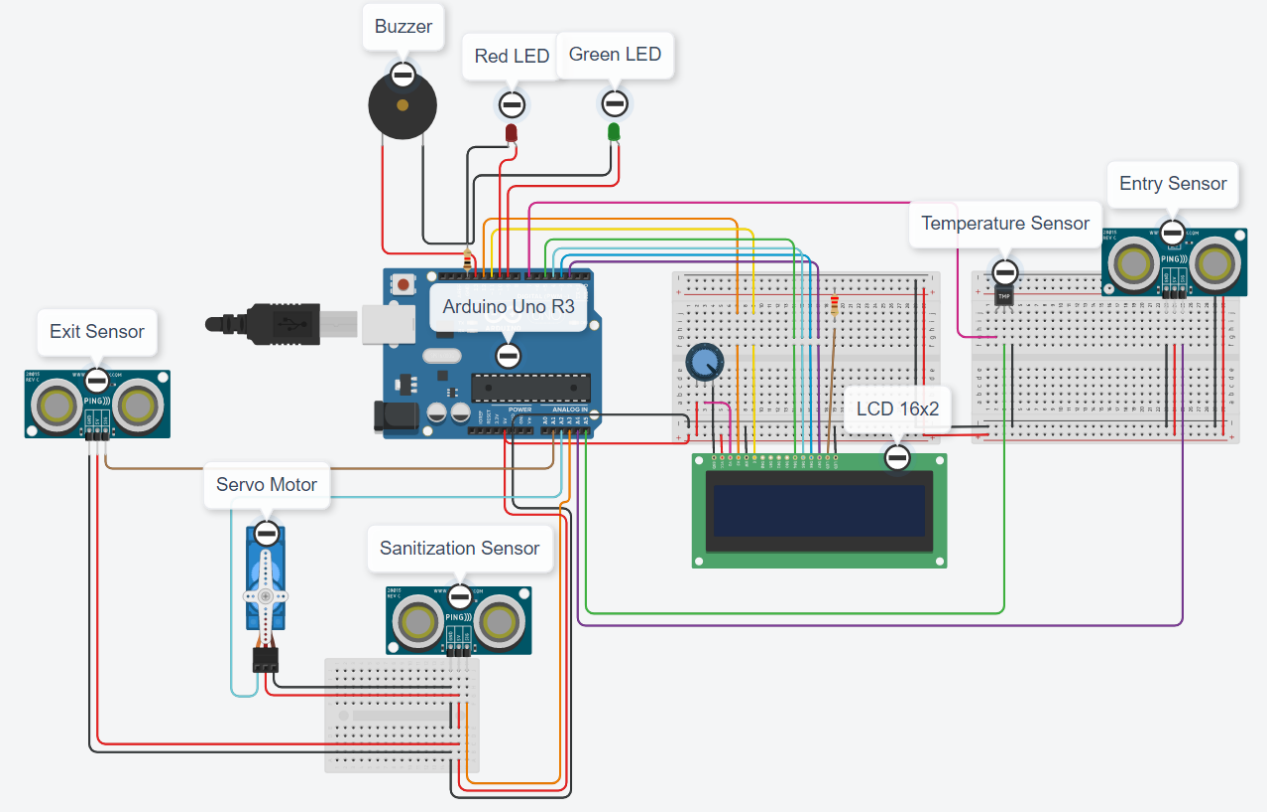
This is where our project could make a difference, by automating the process of thermal screening and sanitisation, thus reduces the risk of infecting humans.

In this detailed report, we are providing the simulated prototype (on TinkerCAD) along with the name of the hardware components required. It is made with inexpensive peripheral devices like sensors and actuators and is reliable to work at any scale. Easy to set-up and customisable according to needs as the Source Code is well segmented and thus can be easily modified.

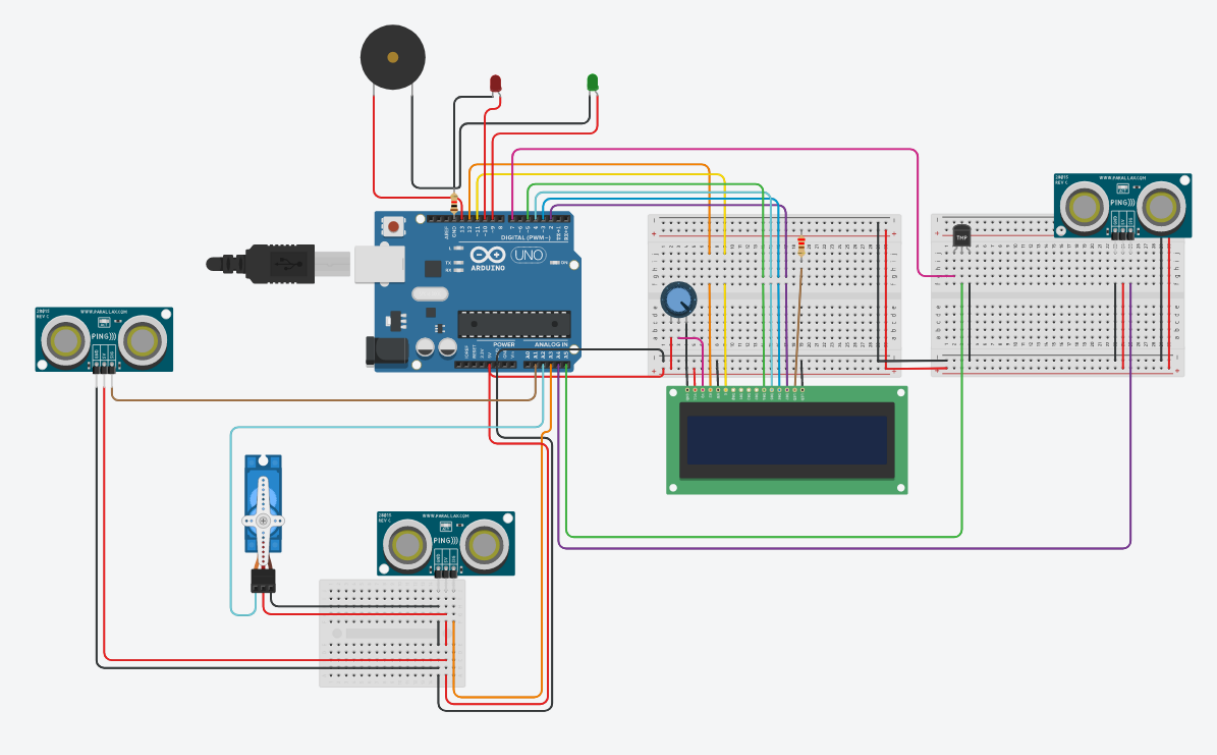
3.3 Hardware Components required:

Components required	Quantity
Arduino Uno R3 (ATmega328P)	1
LCD 16x2	1
250 kΩ Potentiometer	1
220 Ω Resistor	1
Temperature Sensor [TMP36]	1
Ultrasonic Distance Sensor	3
Green LED	1
Red LED	1
Piezo(buzzer)	1
1 kΩ Resistor	1
Red LED	1
Positional Micro Servo	1
Breadboard (Small)	2
Breadboard (Mini)	1
Connecting wires	-

3.4 Circuit Diagram with labelling:



3.5 Circuit Diagram without labelling:



3.6 Functionality:

1. Detects and alerts (through the Buzzer) whenever a person comes near the Entry Sensor, within 10cm to 20cm.
   The attendance and instruction to stand within 10cm to 20cm will be displayed on the LCD.
2. Measures his/her temperature using the Temperature Sensor only when it has already detected the person using Entry Sensor as mentioned in functionality point 1.
3. If the temperature is measured to be above or equal to 37 degree Celsius, the "Buzzer" and the Red LED will turn on to indicate risk. The Buzzer and Red LED will remain active till the person moves away or is moved away to at least 200cm away (or 2 metres).
   If the temperature measured is below 37 degrees Celsius, it will turn on the Green LED and continue to further stages of procedure, like sanitisation.
   The temperature and further instructions will be displayed on the LCD.
4. It will spray sanitiser, simulated by the movement of the Servo Motor whenever the hand is detected at less that 10cm from the Sanitization Sensor.
5. All relevant information will be displayed on the Serial Monitor and LCD.
6. The entry counting and exit counting will be done independently by sensors Entry Sensor and Exit Sensor respectively to ensure more efficient attendance logging.

3.7 Source Code:

```
#include <Servo.h>

#include <LiquidCrystal.h>


// initializing the library of LCD with the numbers of the interface pins

LiquidCrystal lcd(12, 11, 5, 4, 3, 2);


// initializing the library of servo

Servo servo;


// initializing sensor pins

int entry_sensor = A4;

int temp_Vin = 7;

int temp_Vout = A5;

int sanit_sensor = A3;

int servo_pin = A2;

int exit_sensor = A1;
```

```
int green_led = 9;
int red_led = 10;
int buzzer = 13;

// initializing maximum capacity
int max_cap = 10;

// initializing some variables used within the program
String mode = "step closer";
boolean flagPrint = true;
boolean left = false;
int skip = 0;
int attend = 0;
double val;
long long timer = 0;

void setup()
{
  servo.attach(servo_pin);      // attaching servo object to pin
  servo.write(0);
  lcd.begin(16, 2);          // starting functioning of LCD
  Serial.begin(9600);          // starting Serial
}

void loop()
{

  // segment for exiting attendance count
  if (attend > 0)
  {
    double exit = 0.01723 * readUltrasonicDistance(exit_sensor, exit_sensor);
    if (exit < 20)
    {
      attend--;          // decreasing attendance
      Serial.println("Attendance decreased");
      delay(2000);
    }
```

```
}

// segment when attendance is less than maximum capacity
if (attend < max_cap)
{

  // mode "step closer"
  if (mode == "step closer")
  {
    digitalWrite(green_led, LOW);
    digitalWrite(red_led, LOW);
    char c[] = "WELCOME! PLEASE STAND WITHIN 10 TO 20cm ";
    String s = "";
    for (int I = 0; I < 40; i++)
      s = s + c[(skip + i) % 40];
    lcd.setCursor(0, 0);
    lcd.print(s);
    lcd.setCursor(0, 1);
    lcd.print("Attendance: ");
    lcd.print(attend);
    lcd.print("  ");
    skip++;
    val = 0.01723 * readUltrasonicDistance(entry_sensor, entry_sensor);

    // if person is within 10cm to 20cm
    if (val > 10 && val < 20)
    {
      Serial.println("Person detected near entrance");
      timer = millis();
      left = true;
      tone(buzzer, 200);
      delay(50);
      tone(buzzer, 300);
      delay(50);
      noTone(buzzer);
      mode = "temp_check";    // changing mode to "temp check"
    }
```

```
}

// mode "temp check"
else if (mode == "temp_check")
{
  lcd.clear();
  lcd.setCursor(0, 0);
  lcd.print("CHECKING TEMP.");
  lcd.setCursor(0, 1);
  lcd.print("PLEASE WAIT ");
  Serial.println("Checking Temperature");
  digitalWrite(temp_Vin, HIGH);
  delay(2000);
  int reading = analogRead(temp_Vout);
  float voltage = reading * 5.0;
  voltage /= 1024.0;
  float temperature = (voltage - 0.5) * 100 ;
  lcd.clear();
  lcd.print("TEMPERATURE: ");
  lcd.print(temperature);
  lcd.print(" C");

  // when temperature is normal
  if (temperature < 37.0)
  {
    Serial.println("Temperature within limit");
    digitalWrite(green_led, HIGH);      // turning on Green LED
    lcd.setCursor(0, 1);
    lcd.print("PLEASE SANITISE YOUR HANDS");
    for (int I = 1; I <= 40; i++)
    {
      delay(200);
      lcd.scrollDisplayLeft();
    }
    mode = "sanitise";
    lcd.clear();
  }
```

```
  // when temperature is high
  else
  {
    Serial.println("Temperature above limit");
    digitalWrite(red_led, HIGH);        // turning on Red LED
    lcd.setCursor(0, 1);
    lcd.print("PLEASE CONSULT A DOCTOR!");
    double sens_val = 0.0;
    while (sens_val < 200)              // using buzzer to alert
    {
      tone(buzzer, 500);
      delay(200);
      tone(buzzer, 250);
      delay(200);
      lcd.scrollDisplayLeft();
      sens_val = 0.01723 * readUltrasonicDistance(entry_sensor, entry_sensor);
    }
    noTone(buzzer);
    mode = "step closer";        // changing mode to "step closer"
    lcd.clear();
  }
}

// mode "sanitise"
else if (mode == "sanitise")
{
  lcd.setCursor(0, 0);
  lcd.print("BRING YOUR HANDS");
  lcd.setCursor(0, 1);
  lcd.print("NEAR TO SANITISE");
  double sanit = 0.01723 * readUltrasonicDistance(sanit_sensor, sanit_sensor);

  // when hand is close to the sanitiser/sensor
  if (sanit < 10)
  {
```

```
    Serial.println("Hand detected near Sanitiser");
    mode = "pour sanit";      // changing mode to "pour sanit"
    lcd.clear();
  }
}

// mode "pour sanit"
else if (mode == "pour sanit")
{
  Serial.println("Pouring Sanitiser");
  lcd.setCursor(0, 0);
  lcd.print("   SANITISING   ");
  servo.write(120);
  delay(2000);
  servo.write(0);
  delay(1000);
  lcd.clear();
  mode = "enter";     // changing mode to "enter"
}

// mode "enter"
else if (mode == "enter")
{
  lcd.setCursor(0, 0);
  lcd.print("  PLEASE ENTER  ");
  lcd.setCursor(0, 1);
  lcd.print("    THE GATE    ");
  double entry = 0.01723 * readUltrasonicDistance(entry_sensor, entry_sensor);
  if (entry > 200)
  {
    Serial.println("Person entered the gate");
    Serial.println("Attendance increased");
    lcd.clear();
    attend++;       // increasing attendance
    mode = "step closer";   // changing mode to "step closer"
  }
}
```

```
    // segment to handle timeout
    int tim_req = (millis() - timer) / 1000;
    Serial.println(tim_req);
    if (tim_req > 20 && left)
    {
      lcd.clear();
      Serial.println(tim_req);
      left = false;
      mode = "step closer";
    }
  }

  // segment when maximum capacity has reached
  else
  {
    Serial.println("Maximum Capacity reached");
    char c[] = "MAXIMUM CAPACITY REACHED    ";
    String s = "";
    for (int I = 0; I < 28; i++)
      s = s + c[(skip + i) % 28];
    skip++;
    lcd.setCursor(0, 0);
    lcd.print(s);
  }
  skip = skip % 50;
}

// function to use Ultrasonic Distance Sensor
long readUltrasonicDistance(int triggerPin, int echoPin)
{
  pinMode(triggerPin, OUTPUT);
  digitalWrite(triggerPin, LOW);
  delayMicroseconds(2);
  digitalWrite(triggerPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(triggerPin, LOW);
```

```
  pinMode(echoPin, INPUT);
  return pulseIn(echoPin, HIGH);
}
```

Innovative points about the project:

- The sensor to count the people exiting is independent to the thermal screening and sanitisation process and can thus be placed on a different exit door, so it can count people exiting even if a person if going through the screening and sanitisation process at the entry point.
- Going through the sanitisation stage is compulsory if a person wants to enter through the entry point, which should be the only path to enter.
- A person must stand at least 200cm away (2m away, which is the norm) from the person currently going through the screening or sanitisation process, or else he won't be checked and thus not given entry.
- The project has a timeout feature. If the person leaves, after standing and measuring their temperature, or does not sanitise within 10 more second, then he/she must go through the entire process again or the next person will be checked after at least 10 seconds after the previous person's temperature was measured. So, if a person leaves in the middle of the entire screening and sanitisation process, the system will go back to normal activity after sometime time automatically, hence, timeout.
- LCD displays instructions clearly, so a person should understand what to do if he/she feels confused.

Links:

TinkerCAD Project Link: https://www.tinkercad.com/things/fSV1rkQ0w65

Simulation Video Link: https://drive.google.com/file/d/1ZoDRI-b_58xn7UNqeSQ-Vc3ylz5riV_m/view?usp=sharing

Scopes for improvement:

- There is always a scope of improvement for sensors and thus, making a more efficient code could be possible. However, as the code is segmented properly, due to increased readability, improving the code will not be an issue.
- A mechanism attached to the door can be integrated that will order the door to open and close accordingly and will also help counting attendance more effectively.
- The temperature sensor available on TinkerCAD [TMP36] only measures its ambient temperature and thus body temperature of a person cannot be measured properly. So, a different sensor should be used which can measure temperature over a distance.

- A camera and a pre-trained Machine Learning model can be implemented to detect if the person is wearing a mask or not and thus act accordingly. However, handling a camera and sending/receiving data to/from a Machine Learning model requires more computational power and connection to the internet and thus Arduino Uno R3 can't be used. For such purposes, development boards like Raspberry Pi or Orange Pi, integrated with Pi Camera might be used, that would increase the cost of device but will provide significant performance boost and many more features could be integrated.