

# Practical Strategy for Efficient Crowd Management During Covid-19 Pandemic

## A Prototype Design

Prajjalak Chattopadhyay  
Dept. of High Energy Physics  
Tata Institute of Fundamental Research  
Mumbai, India

Sumit Sarkar  
Institute of Radio Physics & Electronics  
University of Calcutta  
Kolkata, India

Suprova Das  
Institute of Radio Physics & Electronics  
University of Calcutta  
Kolkata, India

**Abstract**— To prevent the spread of the Novel Coronavirus disease (COVID-19), we have implemented a circuit which will maintain covid safety protocols in crowded places like vaccination centers without manual influence. The model demonstrating protocol check will work in automation mode once power is given. This model will at first maintain the social distancing in the queue, followed by checking of body temperature. Only those who pass the temperature test will be allowed to sanitize their hands and proceed to enter the room. After that, the door will open when the person places his hand in front of the sensor. A similar arrangement is there for exit as well. It will also keep track of the number of people entering and exiting the place using the automatic door control system. The design and code are simple, and uses all off-the-shelf parts which are easily available in the market.

**Keywords**— Covid-19, Embedded Systems, Microcontrollers, ESP32, Sensors

### I. INTRODUCTION

Severe acute respiratory syndrome (SARS)-associated coronavirus (SARS-CoV) disease (COVID-19) [1] is an infectious disease. COVID-19 has affected about 300 million people and has claimed the lives of more than 50 million people from all over the World [2]. Since it is a droplet-borne disease, it is important to manage crowd and maintain covid protocol in public places. We report a design of self-implemented and standalone circuit using a ESP32 Dev Kit microcontroller board, along with several sensors and actuators, that keep track of social distancing protocol, and features completely non-contact operation of dispensing liquid sanitizer and opening doors.

### II. THEORY

The circuit diagram is shown in fig. (1). The brain of the circuit is a dual core high speed microcontroller ESP32 running FreeRTOS. We exploited the capabilities of FreeRTOS in this setup for parallelizing all the tasks.

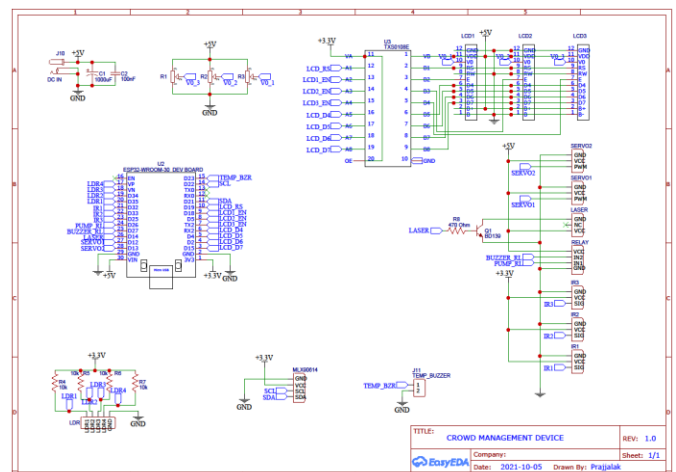


Figure 1: Schematic diagram of the circuit.

The first task is for social distancing. It is done using 4 diode lasers and corresponding LDRs. The Idea is simple -when someone stands in the correct spot, the light from laser will be blocked, and the resistance of the corresponding will increase. A voltage divider is made using 10k resistors and LDRs, and the junction point is measured for voltage using four channels of the inbuilt 12-bit ADCs of ESP32. A recursive filter (see Appendix) is applied on the data with 75% weight value. A sample ADC data and filtering characteristics is shown in fig. (2). If the value is lower than a predefined threshold, an alarm will be triggered, and a 16x2 character LCD will display the corresponding slot number.

Next task is the temperature check. A Melexis MLX90614ESF non-contact infrared temperature sensor [3] measures the temperature. The emissivity is set to 0.97, according to Warmouth and Oliver [4]. The emissivity is defined in eqn. (1). Since the sensor is factory calibrated, we don't need to perform any calibration. However, we verified the calibration of the device using a water bath.

$$E = \frac{T_{object,reading}^4 - T_{ambient,reading}^4}{T_{object,true}^4 - T_{ambient,true}^4} \quad (1)$$

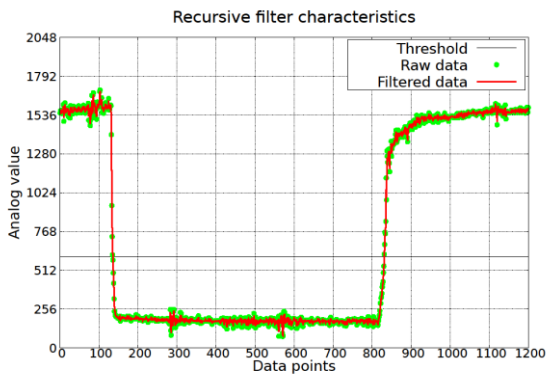


Figure 2: Raw data points from ADC and recursively filtered curve. The threshold is set at 600.

We used Adafruit’s MLX90614 Arduino library [5] in our code. Another 8×2 LCD display will show the temperature reading. If the temperature is higher than a predefined threshold, an alarm will trigger, and further tasks will halt for a predefined time. This is to prevent any person with fever entering the place.

The next task is hand sanitizer dispensation. An IR proximity sensor is used to trigger a peristaltic pump (using a relay, via an optocoupler) for a predefined time, given the temperature check has been passed before. An automatic door control system comes next, where another IR proximity sensor triggers a servo motor for a predefined time. The servo opens a door, through which the person can enter the place. Similar automatic door control system is also in place for exit door. The last task in our code keeps track of number of people entering and exiting the place, alongwith no. of people currently inside. All these stats are displayed in another 16×2 LCD display. All 3 LCD screens share a common bus with different enable lines. All the 3 IR sensors in this circuit are debounced in software.

### III. OPERATION

The people are supposed to stand in correct slots maintaining social distancing. Any violation of that will trigger an audible alarm and will also display the slot number(s) that violated the rule.

The person in front of the queue can then check his temperature. If passed the check, can sanitize his/her hands, and can open the entrance door without touching anything. The person is supposed to place his/her hand in front of the IR proximity sensors.

While exiting, it's again the same non-contact procedure. The number of people entering, exiting and currently inside are also readily available in the corresponding display.

### IV. RESULTS

Initially, a breadboard prototype was made to test the circuit. Several design changes had to be done in the process. The breadboard prototype is shown in fig. (3).

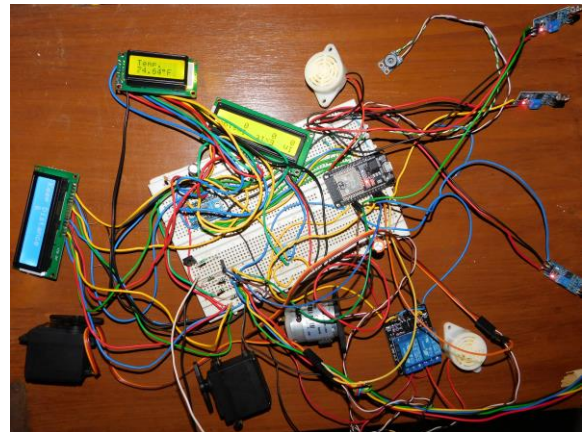


Figure 3: The breadboard prototype. Arduino replaced by simple push button for testing.

A customized PCB followed later, which was designed by us, and was produced by a professional manufacturer. A 3D model of the PCB is shown in fig. (4). The fully assembled PCB is shown in fig.\*.

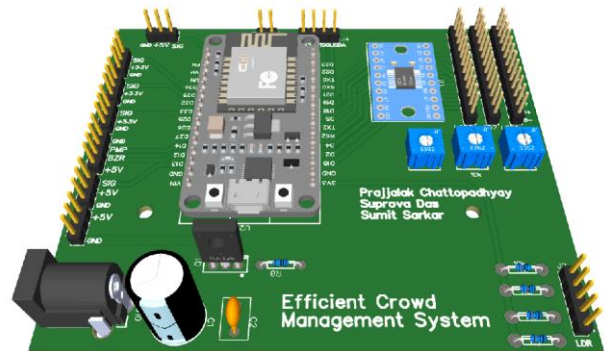


Figure 4: 3D model of PCB, generated by free circuit design software EasyEDA.

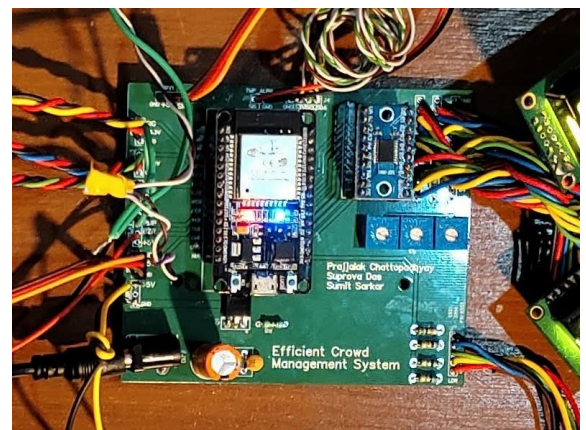
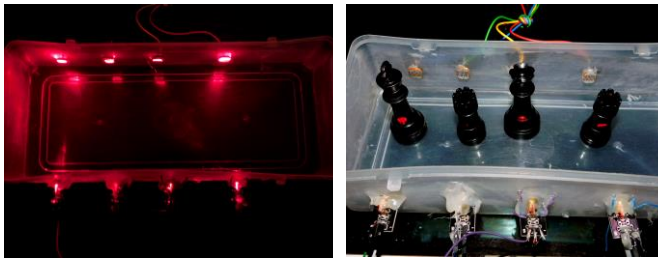


Figure 5: The fully assembled PCB

The social distancing mechanism is shown in fig. (6).



(a) No person standing, The lasers shine light on the LDRs.  
(b) People (dummy) standing in the places. The lights get blocked, and LDRs doesn't receive light.

Figure 6: Social distancing mechanism

Each person in slots whether maintaining social distancing or not is displayed in the LCD as shown in the fig. (7). Following the process sequentially, the measured temperature is displayed in another LCD display as shown in fig. (8). Upon placing user's hand near the first proximity sensor, liquid sanitizer is dispensed.



Figure 7: Social distancing display. It's a 16x2 character LCD with blue backlight.



Figure 8: Temperature display. It's an 8x2 character LCD with green backlight.



Figure 9: No. of persons count display. It's a 16x2 character LCD with green backlight.

Following the sequence described above, entry and exit gates operate with their corresponding proximity sensors. No. of persons is shown on another LCD display, shown in fig. (9).

## V. DISCUSSIONS

The primary design philosophy was to design a system that is cost effective and usable. In dry test our device performed well. But the social distancing monitoring system is not perfect, and creates false trigger sometimes, especially when in well lit area. Also, the number of slots can be increased by using analog multiplexers.

The choice of microcontroller was focused on cost and RTOS support through Arduino IDE. FreeRTOS on ESP32 enables the usage of multitasking. Added, the two cores of ESP32 helps to distribute the computational load.

Finally, Wi-Fi and Bluetooth are very useful features on ESP32 platform. It may be exploited to send these data to some PC or server using MQTT or other similar techniques.

## VI. CONCLUSIONS

Our prototype device ran well in small scale (dummy) testing. This device may be implemented on a large scale and can perfectly be used in the field of real time applications.

## VII. APPENDIX

Raw analog data from a sensor can fluctuate a lot, creating false triggers. Data filtering is required to mitigate the problem. There are many different ways to filter data, like moving average method, and recursive method.

The moving average method requires more memory, and requires more computation, and hence much slower responding to sudden changes. But it is a strong filter.

The recursive filter is much simpler, requires less memory, and less computation. It is fast acting, and responds to sudden changes. Also, the amount of filtering is set by one single variable, called the 'weight factor',  $w$ . Hence it is more suitable for this circuit.

If  $x_n$  be the raw data at  $n^{\text{th}}$  state, and  $y_{n-1}$  is the filtered data at  $(n-1)^{\text{th}}$  state, then filtered data at  $n^{\text{th}}$  state is given by:

$$y_n = wx_n + (1 - w)y_{n-1}$$

The weight factor is between 0-1 (i.e., 0-100%). High value of  $w$  will prefer current raw data over past filtered data, and the filtration will be lower. But the response will be fast.

In the other hand, low  $w$  will prefer last filtered data over current data, and will be a strong filter, but will be slow acting.

We have chosen a weight of 0.75 (75%) as we need fairly fast response time, with little bit of filtering.

## VIII. ACKNOWLEDGMENTS

This project was started as an entry in CoVigilance (A pan India student contest) organized by IEEE MTT & APS Society.

We would like to thank Ms. Subhosri Basu for her help in initiating the project.

## IX. REFERENCES

- [1] "WHO Covid-19 Webpage," [Online]. Available: <https://www.who.int/emergencies/diseases/novel-coronavirus-2019>.
- [2] "World Covid-19 Records," [Online]. Available: <https://www.worldometers.info/coronavirus/>.
- [3] "MLX90614 Datasheet," [Online]. Available: <https://www.melexis.com/en/product/MLX90614/Digital-Plug-Play-Infrared-Thermometer-TO-Can>.
- [4] D. J. WATMOUGH and R. OLIVER, "Emissivity of Human Skin in the Waveband between  $2\mu$  and  $6\mu$ ," *Nature*, vol. 219, no. 5154, pp. 622-624, 01 Aug 1968.
- [5] "Adafruit MLX90614 Library," [Online]. Available: <https://github.com/adafruit/Adafruit-MLX90614-Library>.