# ABSTRACT

We all have been affected by the current COVID-19 pandemic. However, the impact of the pandemic and its consequences are felt differently depending on our status as individuals and as members of society. Governments of various nations are taking all necessary steps to ensure that the people of their nation are prepared well to face the challenge and threat posed by the growing pandemic of COVID-19. The most important factor in preventing the spread of the Virus locally is to empower the citizens with the right information and taking precautions. One of the most common and effective way to prevent COVID-19 from spreading is SOCIAL DISTANCING. The risks of getting COVID-19 are higher in crowded and inadequately ventilated spaces where infected people spend long periods of time together in close proximity. Outbreaks have been reported in places where people have gather, often in crowded indoor settings and where they talk loudly, shout, or sing such as restaurants, fitness classes, offices and places of worship, etc.

The objective of our project is “ **Microcontroller based crowd detection system to prevent COVID-19** ”. Here we will use microcontroller Arduino UNO and infrared sensor modules and various eletronic devices to prevent gathering of huge crowds and hence will alert people to maintain social diastance. Using the mentioned devices we will try to implement an idea or a smart model to how we can prevent crowd gathering, with the help of only machines and minimum staffs(people) in the practical world.

We will be making a small scale version of our objective. Once we achieve success and satisfaction with the idea(model), then we will be able to observe the following results:

* Detection and prevention of crowd gathering.
* Limiting the number of people entering a room at a particular time to prevent COVID-19.
* Creating awareness and alerting people to promote social distancing through displays.

We will be using Arduino IDE and Python IDLE as the main software tool and Arduino UNO board as the microcontroller (hardware) along with various electronic devices for the implementation of our objective.

LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| **Fig. No.** | **Fig.Title** | **Page No.** |
| 3.2.a  3.2.b  3.2.c  3.2.d  3.2.f  3.2.g  3.2.h  3.2.i  3.2.j  3.3.a  3.3.b  3.3.c  3.3.d  3.3.e  3.3.f  3.3.g  3.3.h  3.3.i  3.3.j  3.3.k  3.4.a  3.4.b | pin configuration of the IR sensor module........................................  connection of components in an IR sensor module...........................  Light dependant Resistor..................................................................  Liquid Crystal Display......................................................................  Pins of a LCD...................................................................................  IC- PCF8574 of I2C module............................................................  Other components of I2C module.....................................................  Pins of I2C module...........................................................................  Interfacing of Arduino Uno with LCD- I2C module..........................  Setup of narrow model.....................................................................  different positions of two people within Tile A or Tile C..................  different positions of two people within Tile B or Tile D...................  cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S1........................................................  cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S2.......................................................  cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S3........................................................  cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S4.......................................................  cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S5........................................................  cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S6..............................................................  cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S7..............................................................  cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S8.............................................................  presssure sensor equivalent circuit.........................................................  presssure sensor equivalent circuit made using mentioned components | 7  7  9  9  11  12  12  13  13  14  15  16  17-18  19  19  20  21  22  23  24  24  25 |

|  |  |  |
| --- | --- | --- |
| 3.5.a  3.7.a  4.1.a  4.1.b  4.1.c  4.2.a  4.2.b  4.2.c | Circuit diagram of Narrow Model...........................................................  Circuit diagram of Wide Model..............................................................  Miniature model of Narrow model concept............................................  Serial monitor when no object on the Tiles.............................................  readings shown in Serial monitor according to the location of object on the narrow model................................................................................  wide model..............................................................................................  Screen view of wide model when no faces are registered.......................  Screen views of wide model when faces are detected............................ | 26  34  41  41  42-44  45  46  46-47 |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| **Table No.** | **Table Title** | **Page No.** |
| 3.2.e | Description of LCD pins................................................................... | 10-11 |

|  |  |  |
| --- | --- | --- |
|  | **CONTENTS** |  |
|  |  | Page No. |
| CANDIDATE’S DECLERATION | | i |
| ACKNOWLEDGEMENTS | | ii |
| ABSTRACT | | iii |
| LIST OF FIGURES | | iv- v |
| LIST OF TABLES | | v |
| CONTENTS | | vi- vii |
| **Chapter 1** | **INTRODUCTION** | **1-3** |
| 1.1 | Introduction............................................................................................. | 1-2 |
| 1.2 | Present day scenerio................................................................................ | 2 |
| 1.3 | Motivation............................................................................................... | 2 |
| 1.4 | Objectives............................................................................................... | 3 |
| 1.5 | Target Specification............................................................................... | 3 |
| 1.6 | Project work schedule............................................................................ | 3 |
| **Chapter 2** | **LITERATURE REVIEW** | **4-5** |
| 2.1 | A different approach............................................................................... | 4 |
| 2.2 | literature review...................................................................................... | 4-5 |
| 2.3 | Conclusion.............................................................................................. | 5 |
| **Chapter 3** | **METHODOLOGY** | **6-40** |
| 3.1 | Introduction to Narrow Model............................................................... | 6 |
| 3.2 | Components used in Narrow Model...................................................... | 6-13 |
| 3.3 | Methodology and algorithm used in Narrow Model............................. | 14-24 |
| 3.4 | Making an equivalent for presssure sensor........................................... | 24-25 |
| 3.5 | Execution of Narrow model.................................................................. | 25-32 |
| 3.6 | Introduction to the wide model............................................................. | 33 |
| 3.7 | Explanation of program and execution of wide model......................... | 34-40 |
| **Chapter 4** | **RESULT ANALYSIS** | **41- 48** |
| 4.1 | Practical demonstration results of narror model................................... | 41-45 |
| 4.2 | practical demonstration results of wide model.................................... | 45-48 |
| 4.3 | problems faced and the conclusion for both the models...................... |  |
| **Chapter 5** | **CONCLUSION AND FUTURE SCOPE** | **49- 50** |
| 5.1 | Summary of work................................................................................. | 49 |
| 5.2 | Conclusion............................................................................................ | 49 |
| 5.3 | Future scope of work............................................................................ | 49- 50 |

|  |  |
| --- | --- |
| **REFERENCES** | **11** |

# CHAPTER 1 INTRODUCTION

This chapter includes the introduction of the objective of our project. The chapter starts from section 1.1 that includes the need of our project in current scenerio. Section 1.2 includes about the current situation. Section 1.3 will includes the motivation to do this project. Section

1.4 includes the objectives based on which we will be making the project. Section 1.5 includes target specification and section 1.6 includes how we will be approaching our goal of the project.

* 1. : *Introduction*

COVID-19 (Corona Virus) is an infectious disease that has proved to be an pandemic, declared by the World Health Organization (WHO). COVID-19 being highly contagious, it is absolutely important to prevent and restrict the transmission mechanism. Though the actual propagation mechanism remain elusive, analysis has revealed the spread of respiratory droplets to be the primary reason behind transmission of infection from symptomatic people to asymptomatic ones. In order to limit transmission of infection, the outbreak of COVID-19 has given birth to a new concept of “social distancing” or physical distancing, which is defined as maintaining a distance of at least six feet (2 m) between people. The policy of the application of social distancing has been proven to be effective in reducing the incidence of the disease. So in today’s time, where we have a huge population and to prevent or decrease such diseases has become one of the biggest challenging task, in such case social distancing is one of the solutions to contain the spread of COVID-19. With the help of science and technology one can implement various technical methods or applications in social distancing process where it will give information in time and with great accuracy that will ultimately lead to the prevention of the disease. It will also help in limiting the social bubble. Due to which, it can be seen developers around the world have taken this as an opportunity to design and release applications that will help people maintain social distancing.

Being in a crowded community spaces such as schools, colleges, stadiums, subway stations or holy spots on Pilgrimage impacts not only the level of human convenience but above all the threat of human security. An abnormal crowd conduct can lead to push, mass panic, stampede, crowd-crush, and causing an overall control loss. The current work introduces a microcontroller based crowd detection system. Here, we have proposed two models: firstly,

the narrow model which consists of pressure sensing tiles on the floor where whenever a person steps on the tiles, their body pressure will activate the program and the system will compare their position with respect to a person either standing in front or behind to that person. If the distance between the two people is less than the provided safe distance limit, then an alarm with an alarming message will be displayed on the screen. The second model is wide model. From the name itself we come to know that it will monitor more number of people then the earlier model. It is based on monitoring of crowd in a room, hall, etc. If number of people at a time, in one frame is greater then the provided limit , then alarm will turn ‘ON’.

* 1. : *Present day scenerio*

Crowd Detection is a challenging task that impacts the functioning of public areas. When many people gather with a shared purpose or shared emotions, this is called a crowd. Analyzing and surveilling crowd is of high priority in current Covid scenario. Covid-19 bought a whole lot of restrictions to people lives and has made a huge impact on our lifestyle. Many countries after the initial lockdown lifted the restrictions without thorough planning and the second wave of the pandemic hit them and caused much more destruction.

Crowd Detection can make a huge impact on functioning of public areas. In the current scenario, there is a need for analyzing the crowd in almost everywhere to reduce the spread of the disease which may cause a lot of problems for the community.

* 1. : *Motivation*

Crowd detection is a challenging scientific topic directly impacting on the functioning of trafficked urban infrastructures such as train or metro stations, shopping malls, restaurants etc. Even more so, in time of Covid-19 pandemic, after an initial lock-down period, communities are still wondering how to resume a “new normal” life, while the virus is still circulating among the population. One of the key control measures has been to maintain a minimal physical distance (often also called “social distance”) between any two individuals. As there is a rather widespread suspicion that we may have to live with such requirements of physical distancing for months to come, it is therefore natural that this is becoming a design requirement for crowd detection for public infrastructures.

* 1. : *Objectives*
     + To reduce the posibilities of crowd and if crowd is detected, awareness among people can be created.
     + To design a system that can be understand and developed by common people.
     + To design a logic(algorithm) that requires less hardware and simple to understand by people.
     + To generate an idea that can be further reviewed and upgraded into a more efficient and effective way.
  2. : *Target Specification*

Actually, this project reveals need and importance for an efficient method for crowd dynamics. This system will have multiple applications. Since the project provides a system with reduced human involvement in crowd controlling, it will be highly beneficial. Until a proper treatment for the disease is found, the system will be able to help in to maintain social distancing and after the current scenario the system can be used in different applications that needs crowd behavior insights and trend analysis.

* 1. : *Project work schedule*

Selection of topic

Building the concepts

Rough Idea

Building and final testing of the objective in physical environment

Final outcome and conclusion

# CHAPTER 2 LITERATURE REVIEW

In this chapter we are going to briefly discuss about the theme of our project, literature review and the summarization of it. Section 2.1 includes the explanation and understanding of our objective. Section 2.2 includes the literature review and Section 2.3 contains the conclusion and outcome of the literature review.

* 1. : *A different approach*

The ongoing COVID-19 corona virus 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 has increased. 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 work is, therefore, to provide a microcontroller based crowd detection system for social distance tracking . But we belief that prevention is better than cure. So, in addition to detecting crowd, we will try to avoid crowd in the first place. Thus our approach will be to design such a system(model) that would allow the people to avoid crowd( by monitoring) and also if crowd is detected, the people who are knowingly or unknowingly violating the rule will get to know that that not following the norms and should follow them as soon as possible, for preventing the spread of covid19 to an extend.

* 1. : *literature review*

Some of the earlier works done by rechargers to control the spread of covid19 through social distancing are listed below:

1. In 2001, a very popular approach for object detection was proposed by Viola and Jones.

They used Haar features for features extraction and cascade classifiers with adaboost

learning algorithm for classification purposes.

1. Fu-Chun Hsu et al. proposed a hybrid approach to detect the head and shoulders by fusing

motion and visual characteristics.

1. Vijay and Shashikant proposed a real-time pedestrian detection for advanced driver

assistance. This system detects the pedestrian using Edgelet features to improve the

accuracy and a classifier based on the k-means clustering algorithm to lessen the system

complexity.

1. Seemanthini and Manjunath deployed the human detection technique for an action

recognition system.

1. Singh et al. proposed a human detection framework for extensive surveillance in the city

through CCTV cameras. They used the background subtraction technique to segment

moving objects, HOG descriptor to extract features and SVM for object classification.

1. YOLOv3 is an autonomous drone-based model for social distance monitoring. The data set is composed of frontal and side view images of limited people. The work is also extended for the monitoring offacial masks. The drone camera and theYOLOv3 algorithm help identify the social distance and monitor people from the side or frontal view in public wearingmasks.
2. Pouw, Toschi, van Schadewijk, and Corbetta (2020) suggested an efficient graph- based monitoring framework for physical distancing and crowdmanagement.
3. Sathyamoorthy, Patel, Savle, Paul, and Manocha (2020) performed human detection in a crowded situation. The model is designed for individuals who do not obey a social distancerestriction. They used a mobile robot with an RGB-D camera to make collision- free navigation in mass gatherings.
4. Until now researchers have done considerable work for detection ([Iqbal, Ahmad, Bin,](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0110)

 [Khan, & Rodrigues, 2020](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0110); [Patrick et al., 2020](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0140); [Yash Chaudhary & Mehta, 2020](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0230)), some provides an smart healthcare system for pandemic using Internet of Medical Things ([Chakraborty, 2021](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0070); [Chakraborty et al., 2021](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0075)). [Prem et al. (2020)](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7603992/#bib0150) studied the social distancing impacts on the spread of the COVID-19 outbreak.

The studies concluded that the early and immediate practice of social distancing could gradually reduce the peak of the virus attack.

* 1. : *Conclusion*

The literature review suggests that existing research findings on social distancing are generally based on or were done in the fields of image processing, computer vision, deep learning, machine learning. Our objective is to find a solution to detect crowd using microcontroller which is a new field to explore in this top

# CHAPTER 3

**METHODOLOGY**

In this chapter we will be discussing about the overall steps and methods we adopted to implement our objective. Section 3.1 will include introduction and practical idea of narrow model concept. Section 3.2 includes the explanation of the components required for building the narrow model. Section 3.3 includes the explanation of the algorithm, used to execute the narrow model. Section 3.4 describes an alternative to presssure sensor which we will use in narrow model. Section 3.5 overall execution and testing of narrow model. Section 3.6 includes introduction to the wide model . Section 3.7 includes the methodology and execution of wide model.

* 1. : *Introduction to Narrow Model*

We have divided our project into two parts : Narrow Model and Wide Model. First we would discuss about the Narrow model.

Narrow model is based on the measurement of linear distance between two people. As the name suggest, this concept can be applied wherever there is narrow space like corridor, or along a passage surrounded by walls on both sides. Practically, this model will be based on the concept that whenever and wherever along the passage, two or more people are standing close to each other with a distance less than the allowed distance, the staffs of the building as well as both the people who have violated with or without noticing will be alerted by a message that might be displayed on a screen along with an alarming sound. That way, people will come to know that they need to maintain some distance and hence social distancing can be followed and thus preventing the spread of covid.

In our miniaturised model we will replace the proxiity sensors with infrared sensors and the pressure sensing tiles with an equivalent logic using light dependent resistor(ldr)s and light emitting diodes. Also for displaying the message for alerting people we will be using a liquid crystal display(LCD) and a buzzer to produce sound. We will be writing our code in Arduino IDE and will be using microcontroller arduino Uno to execute the model based on the algorithm explained in the section 3.3.

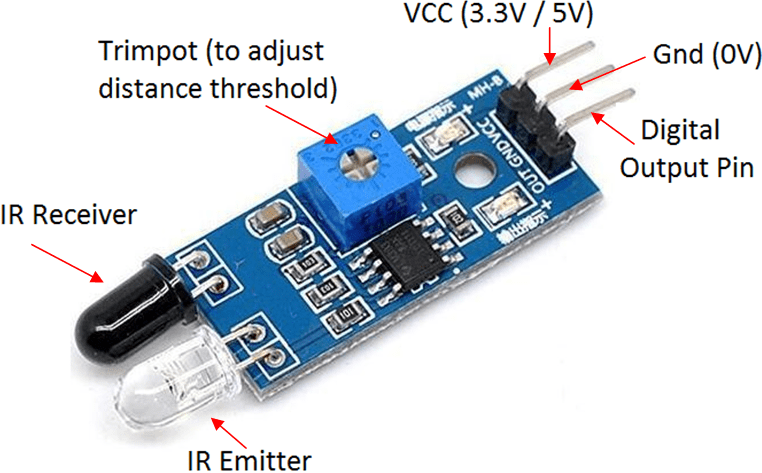
* 1. : *Components used in Narrow Model*

A brief introduction to the components we would use for narrow model:

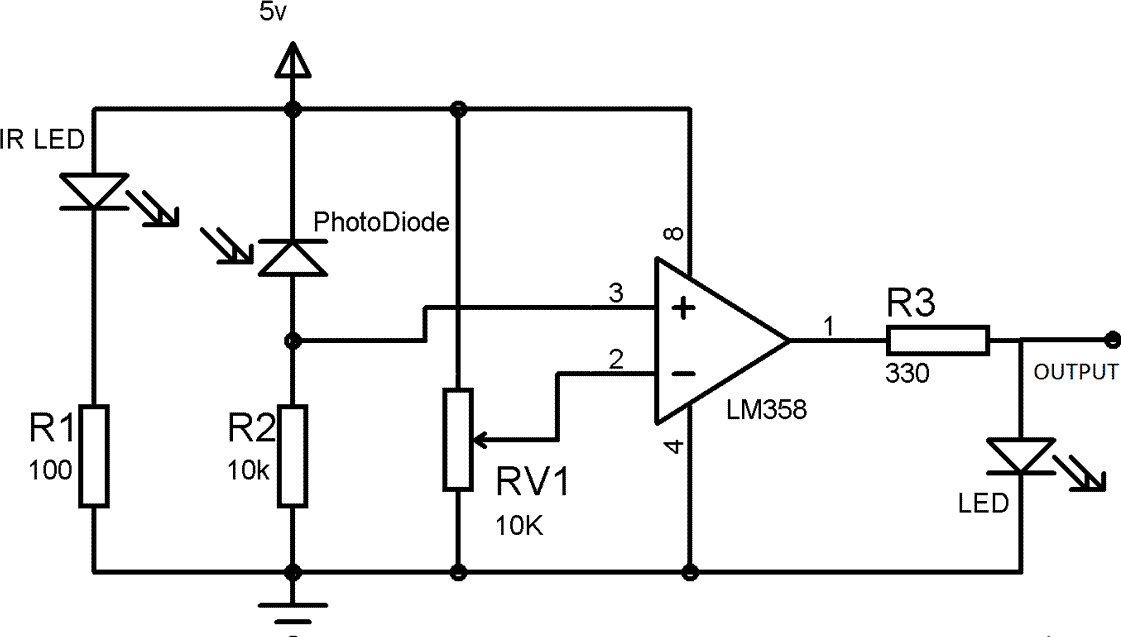
## Infrared Sensor Module

The IR sensor module includes five essential parts like IR LED Transmitter,

Photodiode Receiver, Operational amplifier( LM358), trimmer pot (variable resistor) & output LED where **VCC Pin** is power supply of input ,**GND Pin** is power supply ground, **OUT** is an active-high output. The pin configuration of the IR sensor module is discussed below.



fig(3.2.a): pin configuration of the IR sensor module



fig(3.2.b): connection of components in an IR sensor module

The main **specifications and features of the IR sensor** module include the following.

* + The operating voltage is 5VDC
  + I/O pins – 3.3V & 5V
  + Mounting hole
  + The range is up to 20 centimeters
  + The supply current is 20mA
  + The range of sensing is adjustable

IR LED Transmitter

IR LED emits light, in the range of Infrared frequency. IR light is invisible to us as its wavelength (700nm – 1mm) is much higher than the visible light range. IR LEDs have light emitting angle of approx. 20-60 degree and range of approx. IR LED is white or transparent in colour, so it can give out amount of maximum light.

Photodiode Receiver

Photodiode acts as the IR receiver as its conducts when light falls on it. Photodiode is a semiconductor which has a P-N junction, operated in Reverse Bias, means it start conducting the current in reverse direction when Light falls on it, and the amount of current flow is proportional to the amount of Light. This property makes it useful for IR detection. Photodiode looks like a LED, with a black colour coating on its outer side, Black colour absorbs the highest amount of light.

LM358 Opamp

LM358 is an Operational Amplifier (Op-Amp) is used as voltage comparator in the IR sensor. the comparator will compare the threshold voltage set using the preset (pin2) and the photodiode’s series resistor voltage (pin3).

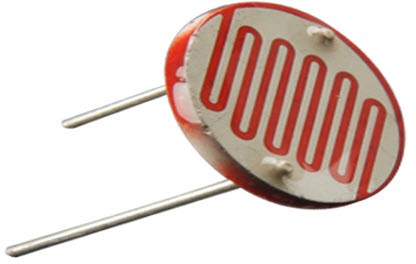
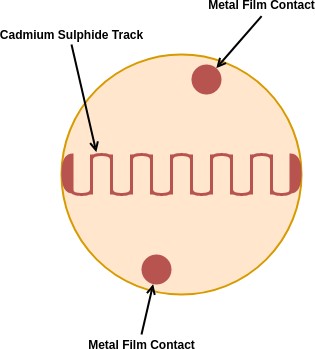
* Photodiode’s series resistor voltage drop > Threshold voltage = Opamp output is High
* Photodiode’s series resistor voltage drop < Threshold voltage = Opamp output is Low
* When Opamp's output is **high** the LED at the Opamp output terminal **turns ON** (Indicating the detection of Object).

Variable Resistor

The variable resistor used here is a preset. It is used to calibrate the distance range at which object should be detected.

## Light dependant Resistor

A photoresistor or light dependent resistor is an electronic component that is sensitive to light. When light falls upon it, then the resistance changes. Values of the resistance of the

LDR may change over many orders of magnitude the value of the resistance falling as the level of light increases. An LDR or photoresistor is made out of any semiconductor material with a high resistance. It has a high resistance because there are very few electrons that are free and able to move - the vast majority of the electrons are locked into the crystal lattice and unable to move. Therefore in this state there is a high LDR resistance. As light falls on the semiconductor, the light photons are absorbed by the semiconductor lattice and some of their energy is transferred to the electrons. The amount of energy transferred to the electrons gives some of them sufficient energy to break free from the crystal lattice so that they can then conduct electricity. This results in a lowering of the resistance of the semiconductor and hence the overall LDR resistance. The process is progressive, and as more light shines on the LDR semiconductor, so more electrons are released to conduct electricity and the resistance falls further.

fig(3.2.c): Light dependant Resistor

## Liquid Crystal Display with I2C module

The display units are very important in communication between the human world and the machine world.The 16×1 display unit has the 16 characters which present in one line and 16×2 display units have 32 characters which are present in the 2 line.



fig(3.2.d): Liquid Crystal Display

The liquid crystal display uses the property of light monitoring of liquid crystal and they do not emit the light directly. The Liquid crystal display is a flat panel display. They are all around us in laptop computers, digital clocks and watches, microwave ovens, CD players and many other electronic devices. LCDs are common because they offer some real advantages over other display technologies. They are thinner and lighter and draw much less power than LEDs.

Liquid Crystal Display of 16×2:

The 16×2 liquid crystal display contains two horizontal lines and they are used for compressing the space of 16 display characters. In inbuilt, the LCD has two registers which are described below:

* + Command Register
  + Data Register

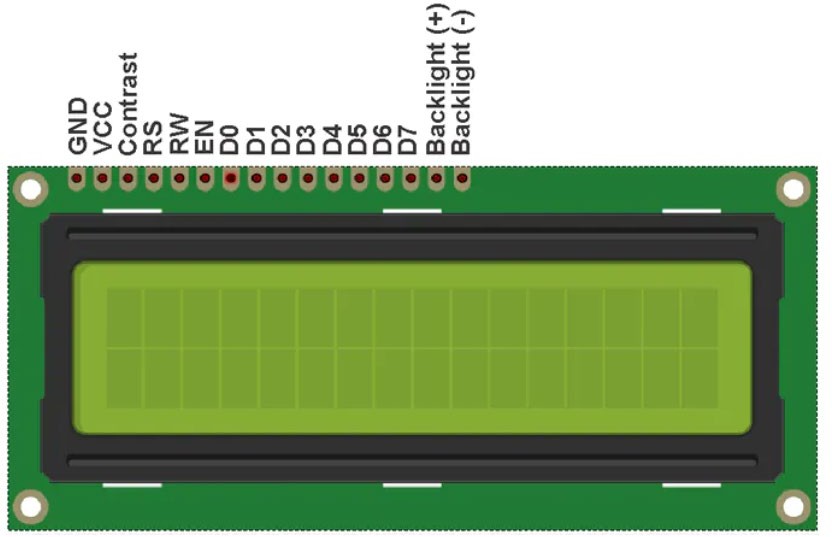
*Command Register:* This register is used to insert a special command in the LCD. The command is a special set of data and it is used to give the internal command to the liquid crystal display like clear screen, move to line 1 character 1, setting the curser and etc.

*Data Register:* The data registers are used to enter the line in the LCD.

fig(3.2.e): Description of LCD pins

|  |  |  |
| --- | --- | --- |
| Pin No | Pin Name | Pin Description |
| Pin 1 | GND | This pin is a ground pin and the LCD is connected to the  Ground |
| Pin 2 | VCC | The VCC pin is used to supply the power to the LCD |
| Pin 3 | VEE | This pin is used for adjusting the contrast of the LCD by  connecting the variable resistor in between the VCC & Ground. |
| Pin 4 | RS | The RS is known as register select and it selects the Command/Data register. To select the command register the RS should be equal to zero. To select the Data register the RS  should be equal to one. |

|  |  |  |
| --- | --- | --- |
| Pin 5 | R/W | This pin is used to select the operations of Read/Write. To perform the write operations the R/W should be equal to zero.  To perform the read operations the R/W should be equal to one. |
| Pin 6 | EN | This is an enable signal pin if the positive pulses are passing  through a pin, then the pin function as a read/write pin. |
| Pin 7 | DB0 to DB7 | The pin 7 contains total 8 pins which are used as a Data pin of  LCD. |
| Pin 15 | LED + | This pin is connected to VCC and it is used for the pin 16 to set  up the glow of backlight of LCD. |
| Pin 16 | LED – | This pin is connected to Ground and it is used for the pin 15 to  set up the glow of backlight of the LCD. |



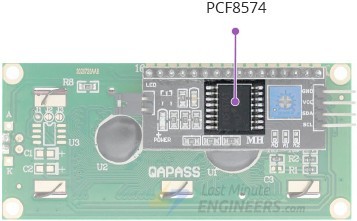
fig(3.2.f): Pins of a LCD

If we notice, then we can find that to connect an LCD display with an Arduino, a lot of pins on the Arduino is consumed. Even in 4-bit mode, the Arduino still requires a total of seven connections – which is half of Arduino’s available digital I/O pins.

The solution is to use an I2C LCD Display. It consumes only two I/O pins that are not even part of a set of digital I/O pins and can also be shared with other I2C devices.

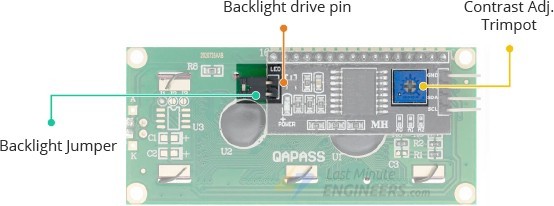
*I2C LCD Adapter*

At the heart of the adapter is an 8-Bit I/O Expander chip – PCF8574. This chip converts the I2C data from an Arduino into the parallel data required by the LCD display.



fig(3.2.g): IC- PCF8574 of I2C module

The board also comes with a small trimpot to make fine adjustments to the contrast of the display.



fig(3.2.h): Other components of I2C module

In addition, there is a jumper on the board that supplies power to the backlight. To control the intensity of the backlight, you can remove the jumper and apply an external voltage to the header pin that is marked as ‘LED’.

I2C LCD display Pinout

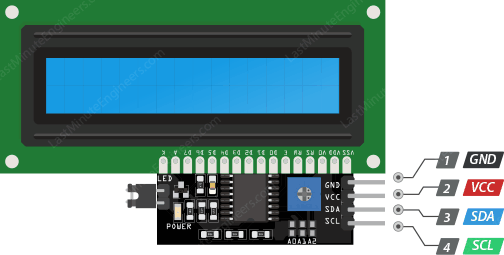
An I2C LCD has only 4 pins that interface it to the outside world. The connections are as follows:

‘GND’ is a ground pin and should be connected to the ground of Arduino.

‘VCC’ supplies power to the module and the LCD. Connect it to the 5V output of the Arduino or a separate power supply.

‘SDA’ is a Serial Data pin. This line is used for both transmit and receive. Connect to the SDA pin on the Arduino.

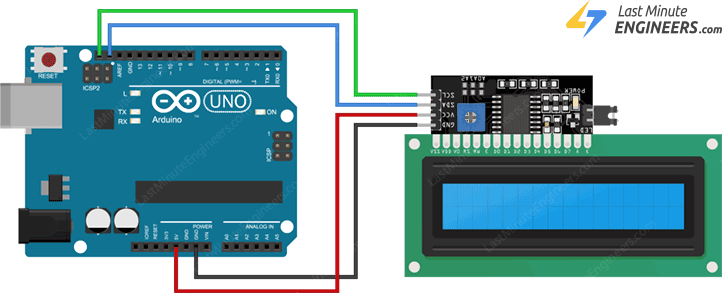
‘SCL’ is a Serial Clock pin. This is a timing signal supplied by the Bus Master device.



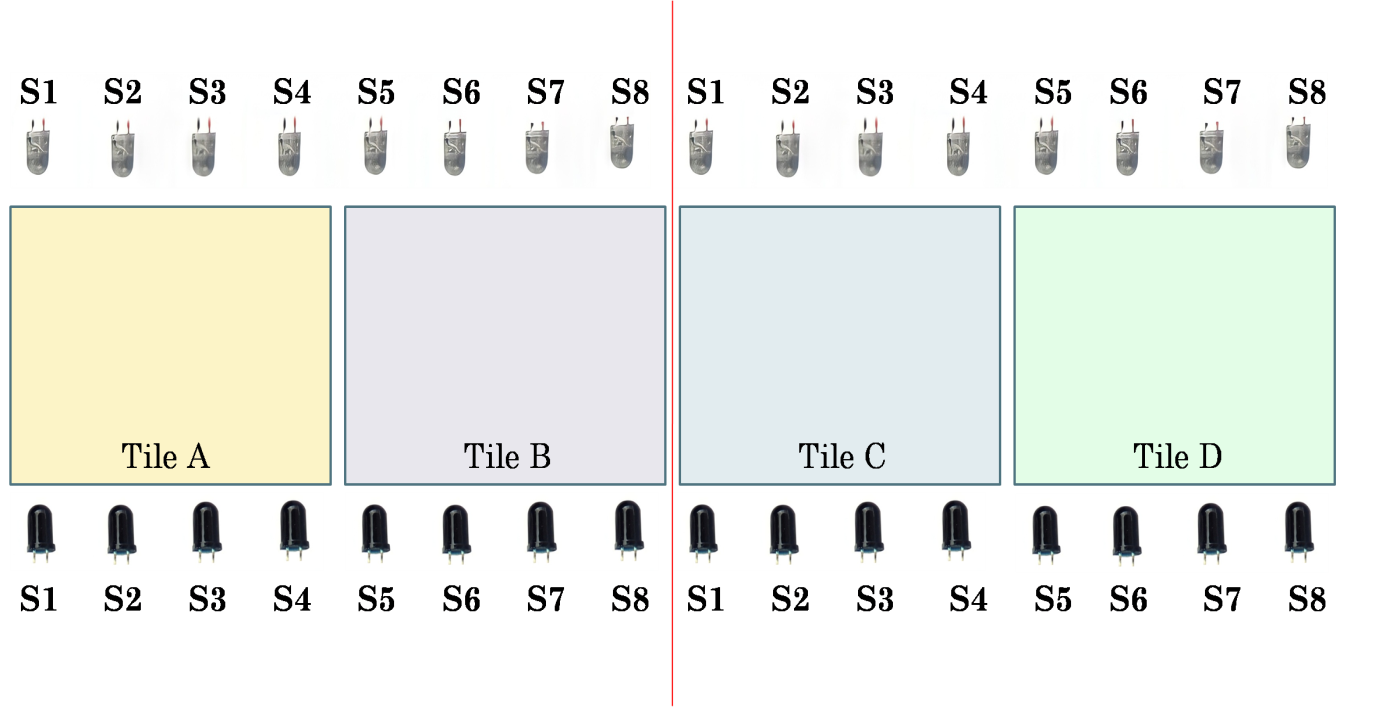
fig(3.2.i): Pins of I2C module

After we wire the LCD, we will need to adjust the contrast of the display. On the I2C module, we can find a potentiometer that can be turned using a small screwdriver.

Plug in the USB connector of the Arduino to power the LCD. We should see the backlight light up. Now we rotate the potentiometer until the first line of the rectangle appears.Once that is done, we can start programming the LCD.



fig(3.2.j): Interfacing of Arduino Uno with LCD- I2C module

* 1. : *Methodology and algorithm used in Narrow Model*

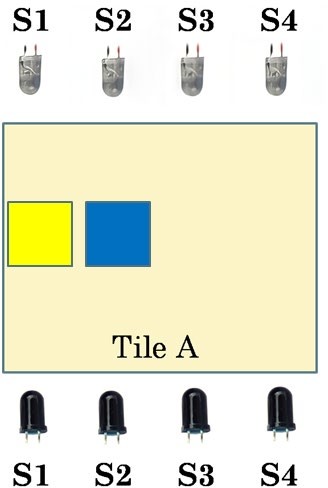
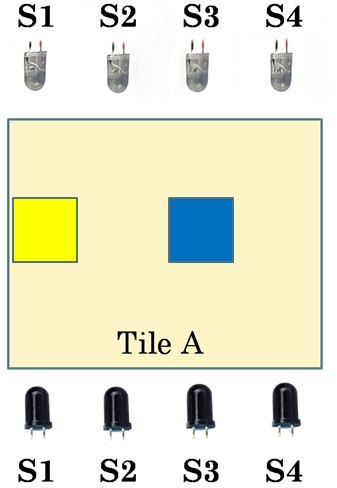
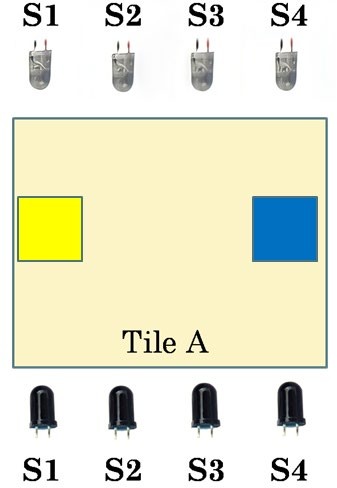
fig(3.3.a): Setup of narrow model

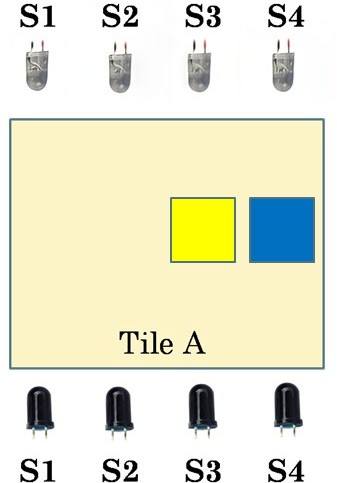
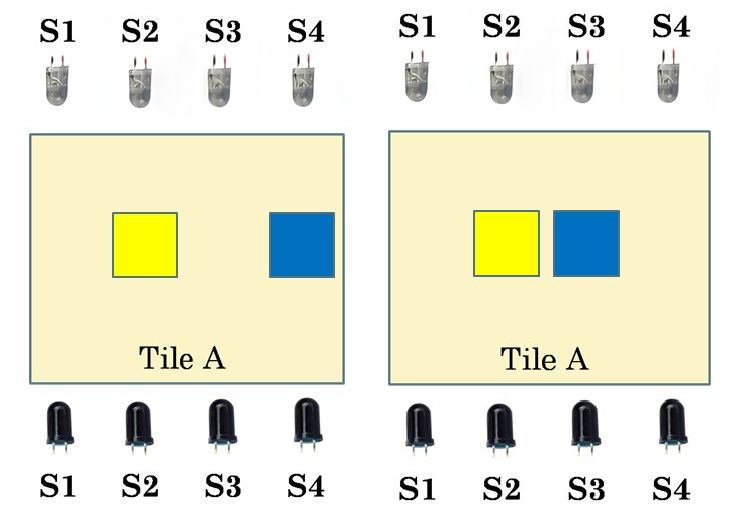
The above figure is the general setup of the narrow model. Here the white or transparent diodes are the infrared emitters or IR LEDs and the black diodes are the infrared receivers or photodiodes. Here we have used 8 pairs of IR LED and photodiode and each IR LED is aligned HEAD ON with a photodiode. Also we have four tiles namly Tile A, Tile B, Tile C and Tile D and in each of Tile A and Tile B we have four sets of IR transmitter and receiver that is TileA contains IR transmitter and receiver from one to four (namly S1, S2, S3, S4) and TileB contains IR transmitter and receiver from five to eight (namly S5, S6, S7, S8). The same goes to Tile C and Tile D where each tile has 4 pairs of IR transmitters and receivers.

The main thing we can observe here is that Tile A and Tile C both share the same sensor logic, which means though overall there are 8 pairs of IR transmitters and receivers but by the same sensor logic we mean that sensors of Tile A and Tile C are internally shorted, thus one signal from the microcontroller will serve both the sensor pair or IR module. Similarly we can observe that Tile B and Tile C share the same sensor logic with S5, S6, S7, S8. For an easy understanding of the algorithm, which we used to implement the narrow model, we will be explaining the algorithm by taking cases as follows:

CASE1(a): Crowd Detection on a single tile (Tile A or Tile C)

Here we will consider what will happened if two or more people stand on either Tile A or Tile C only. Both the tiles share the same sensor logic as explained before.

* + 1. b) c)

d) e) f) fig(3.3.b): different positions of two people within Tile A or Tile C

The distance between each IR sensor is 1.5 cm (used in our miniature model). Let us consider two people and indicate their positions on the tiles by yellow for one person and by blue colour for the other. Here, both the person are standing on the same Tile A. If a distance less then 4cm is detected then an alarm along with a message will be displayed on the screen to aware the people to maintain distance. In all the cases we will move the position of the person marked by blue colour with respect to the person marked by yellow colour. Suppose a person is detected by the sensor S1 and at the same time another person is detected by the sensor S2 then as already mentioned distance between two sensors is 1.5cm (for the model) which is less than 4cm, hence there will be an awareness alarm alerting both the person and the authorities. Thus if a person is at a position detected by sensor S1 and simultaneausly another person is at a position detected by sensor S2, then there will be an alert. Similarly in cases of fig b), fig c), fig d), fig e), fig f), due to having a distance less than 4cm between the two

people, an alert signal would be generated to promote awareness.Thus algorithm for the first case that is Crowd Detection on a single tile (Tile A or Tile C) is as follows:

If Sensor 1 (S1) = 1 and Sensor 2 (S2) = 1

If Sensor 1 (S1) = 1 and Sensor 3 (S3) = 1

If Sensor 1 (S1) = 1 and Sensor 4 (S4) = 1

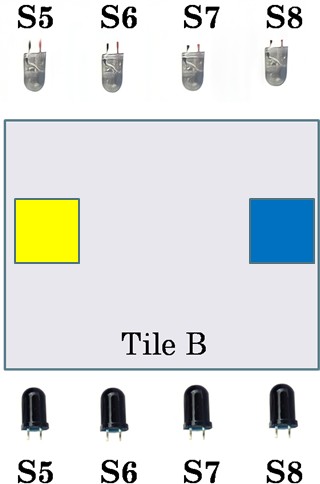
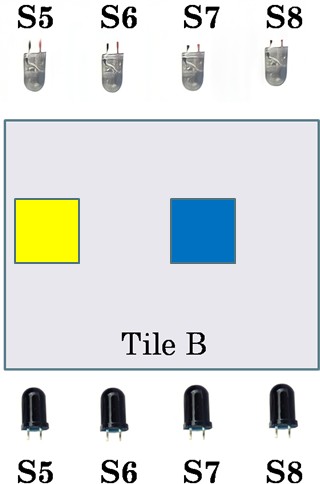
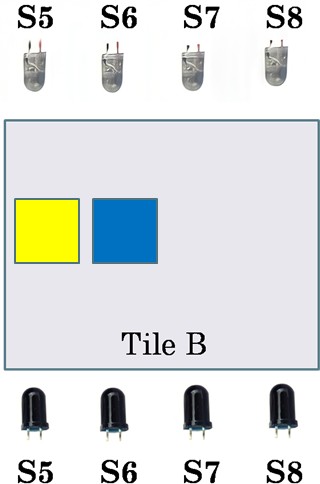
If Sensor 2 (S2) = 1 and Sensor 3 (S3) = 1

If Sensor 2 (S2) = 1 and Sensor 4 (S4) = 1

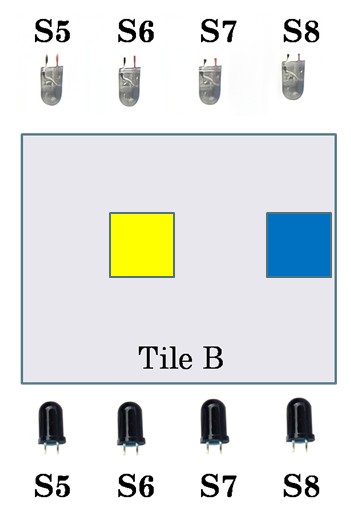
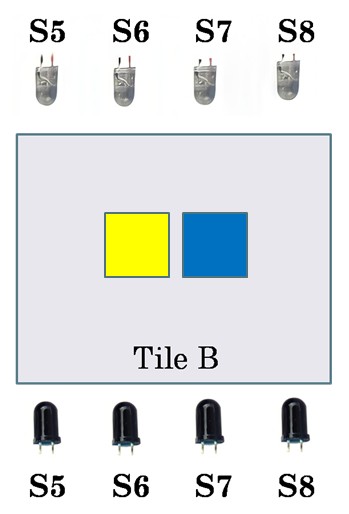
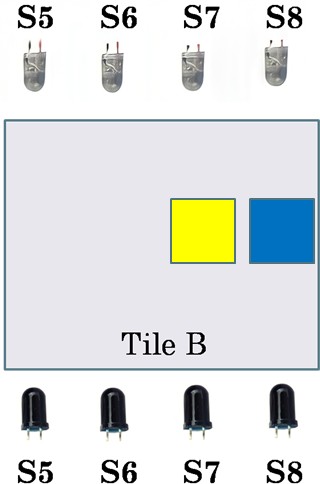
If Sensor 3 (S3) = 1 and Sensor 4 (S4) = 1

Then show “Crowd detected. Please maintain social distancing.” Else No Alert.

CASE 1(b): Crowd Detection on a single tile (Tile B or Tile D)

Similarly, here we will consider what will happened if two or more people stand on either Tile B or Tile D only. Also both the tiles share the same sensor logic as explained before.

a) b) c)

d) e) f) fig(3.3.c): different positions of two people within Tile B or Tile D

Everything will remain the same as the previous case1(a). Only difference will lie here is the sensors, that is earlier we had sensor S1, S2, S3, S4 and now we will have sensors S5, S6, S7, S8. So the algorithm for this case that is ‘Crowd Detection on a single tile (Tile B or Tile D)’ is as follows:

If Sensor 5 (S5) = 1 and Sensor 6 (S6) = 1 (fig a) If Sensor 5 (S5) = 1 and Sensor 7 (S7) = 1 (fig b) If Sensor 5 (S5) = 1 and Sensor 8 (S8) = 1 (fig c) If Sensor 6 (S6) = 1 and Sensor 7 (S7) = 1 (fig d) If Sensor 6 (S6) = 1 and Sensor 8 (S8) = 1 (fig e) If Sensor 7 (S7) = 1 and Sensor 8 (S8) = 1 (fig f)

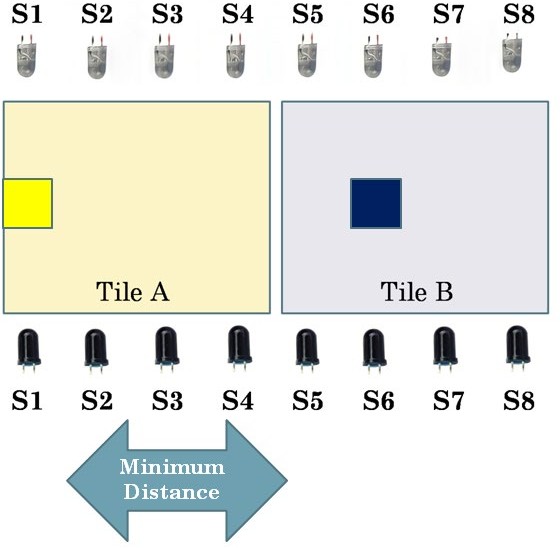
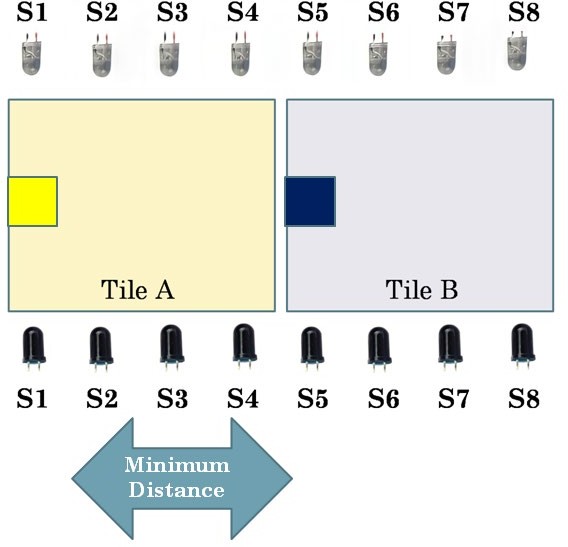
Then show “Crowd detected. Please maintain social distancing.” Else No Alert.

The above two cases describes the situation when crowd is detected on a single tile. Now what will happen if one person is standing on one Tile and another person standing on a different Tile? How will the algorithm work in such type of situation? So, now we move on to the next case.

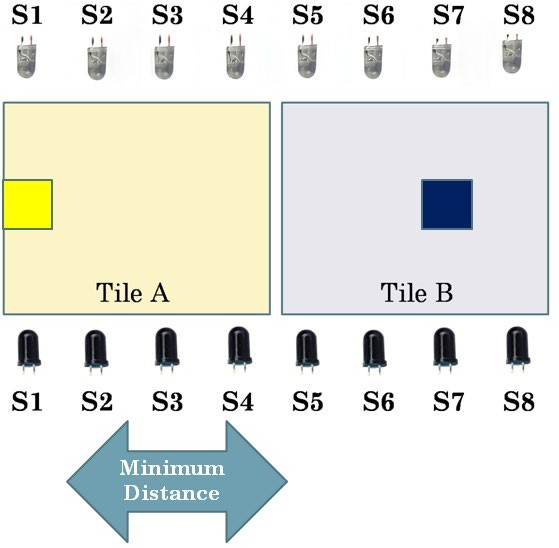
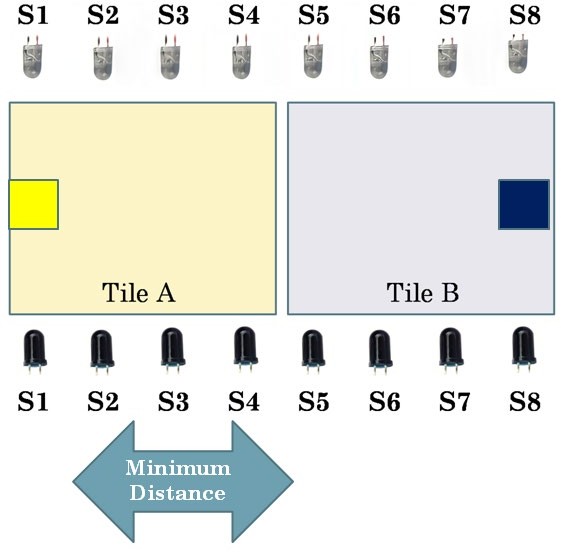
CASE 2(a): Crowd Detection between two tiles (Tile A and Tile B) or (Tile C and Tile D)

As algorithm for Tile A and Tile B is similiar to that of Tile C and Tile D respectively, therefore this case can be explained either taking Tile A and Tile B or taking Tile C and Tile

D. We will be explaining the algorithm using Tile A and Tile B.



1. b)

c) d)

fig(3.3.d): cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S1.

Here, we will be changing the position of the person standing on Tile B (marked by blue) with respect to the person standing on Tile A(marked by yellow).

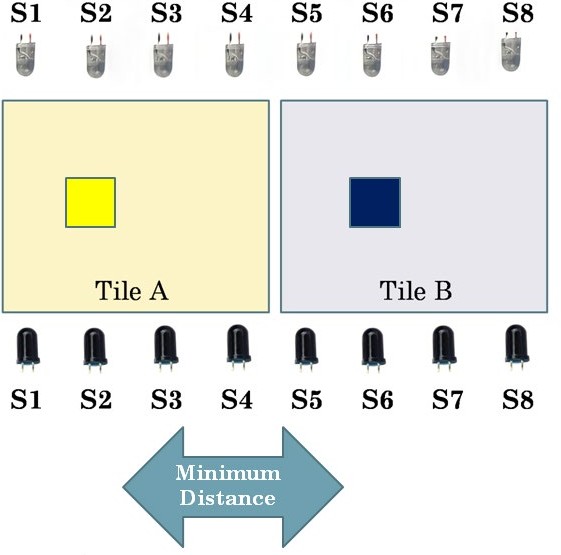
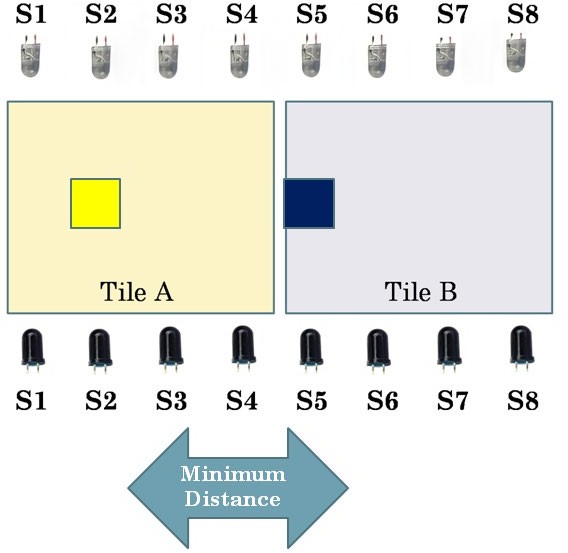
From the fig(3.3.d) we observe that one person is standing on Tile A and is detected by sensor S1. From earlier cases we already know that if the distance between two or more people is less than 4cm(for our model), then alarm will turn on. So here also the logic of 4cm distance will continue. Also we know the distance between two sensors (can be between two IR LEDs or between two photodiodes) is 1.5 cm. Now we assume that a person is on Tile A and is been detected by sensor S1 and another person is on Tile B detected by sensor S5, as the distance between S1 and S5 is more than 4cm therefore no alarm will be turned on. Similarly if the person’s position marked by blue colour moves to position S6,S7 or S8 the result will be the same.

In short: If **S1=1 and S5=1** or If **S1=1 and S6=1** or If **S1=1 and S7=1** or If **S1=1 and S8=1**, There won’t be any alarm.

Suppose the Person at S1(fig(3.3.e)) moves to sensor S2 and is detected by S2 now and the other person is on Tile B at a position detected by sensor S5. As the distance between S2 and S5 is smaller then 4cm, therefore there will be an alarm. Now the other person moves from S5 to S6. The distance between S2 and S6 is greater than 4cm ,hence no alarm will be activated.Similarly the distance between S2 and S7 and S2 and S8 will have distance greater than 4cm , so alarm will be off for these cases.

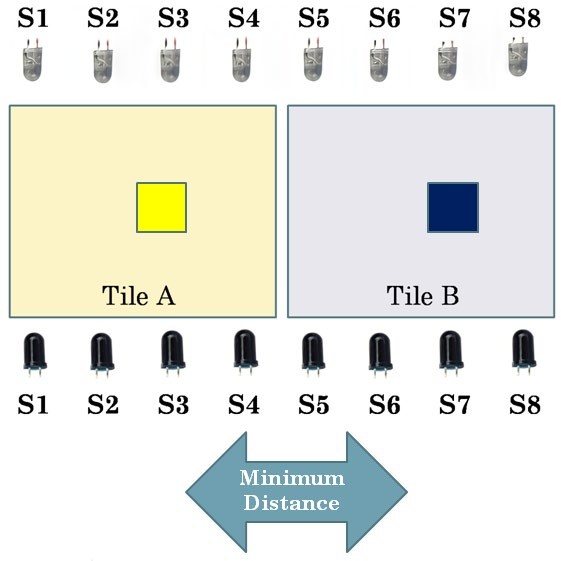
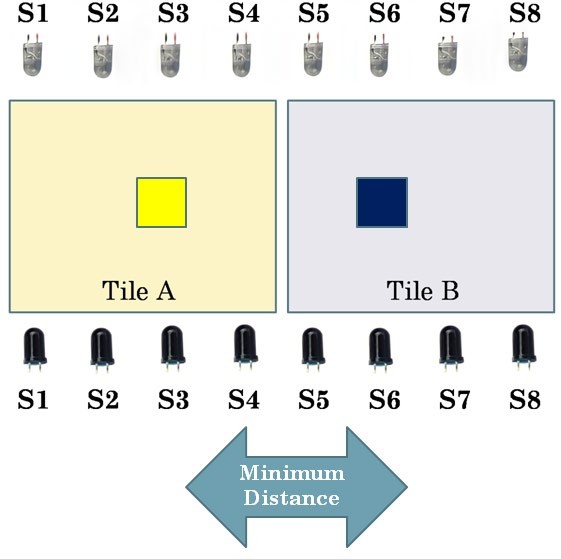
In short: If **S2=1 and S5=1** , then show Crowd Detected

Else Alarm will be off



* 1. b)

fig(3.3.e): cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S2.



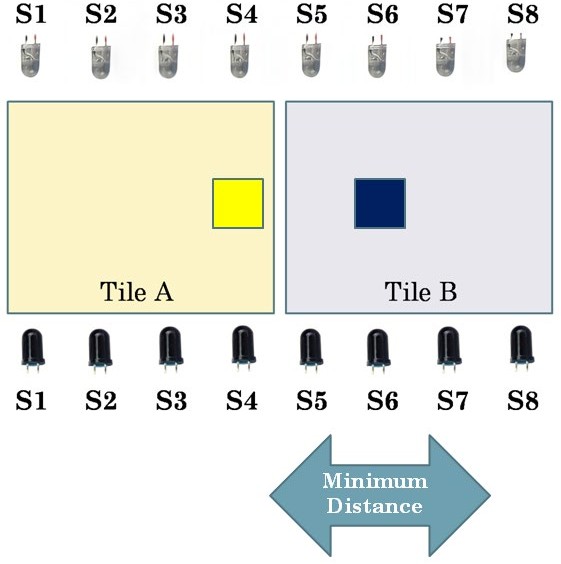
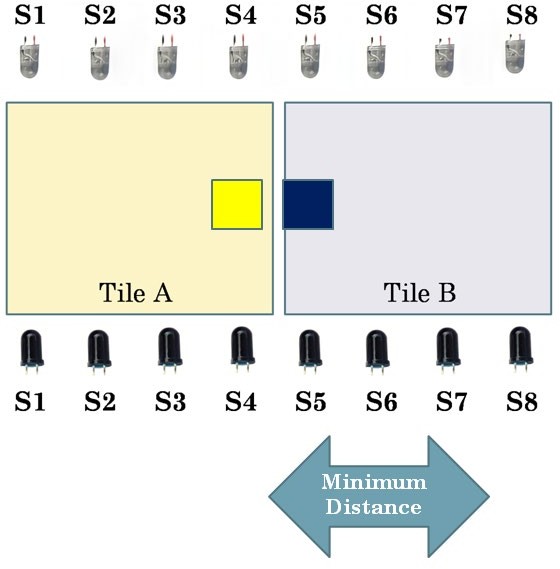
* + 1. b)

fig(3.3.f): cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S3.

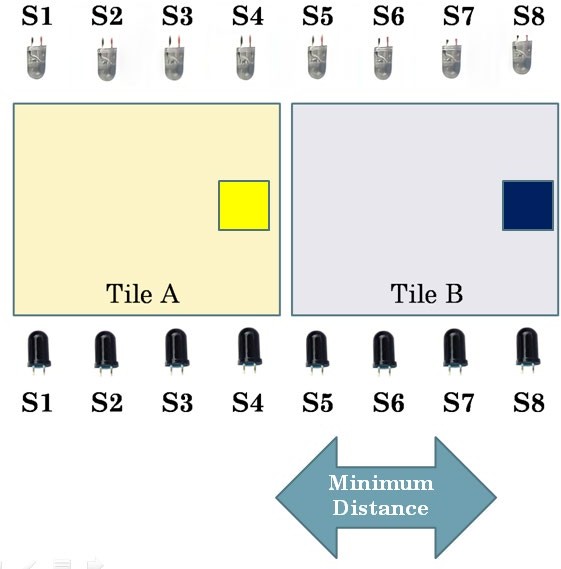
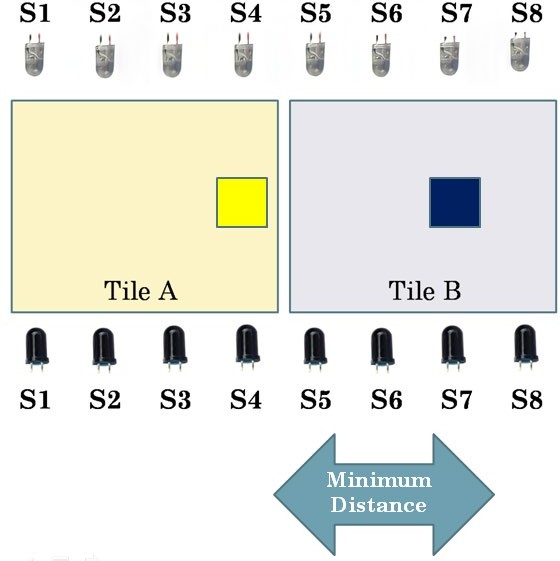
Suppose now the person standing on Tile A moves to position S3 from S2 and another person is at the position detected by sensor S5, clearly the distance will be less than 4cm, hence alarm will be turned on. Similarly when the other person moves from S5 to S6 position, alarm will still turn ‘ON’. Now when the person moves to position S7 from S6 , due to greater than

4cm distance between , alarm will not turn ‘ON’ and same can be explained for distance S3 and S8. In short: If **S3=1 and S5=1** or If **S3=1 and S6=1** , then show Crowd Detected.

Else No alarm.



* + - 1. b)



1. d)

fig(3.3.g): cases for a person moving to different positions on Tile B w.r.t person on Tile A detected by sensor S4.

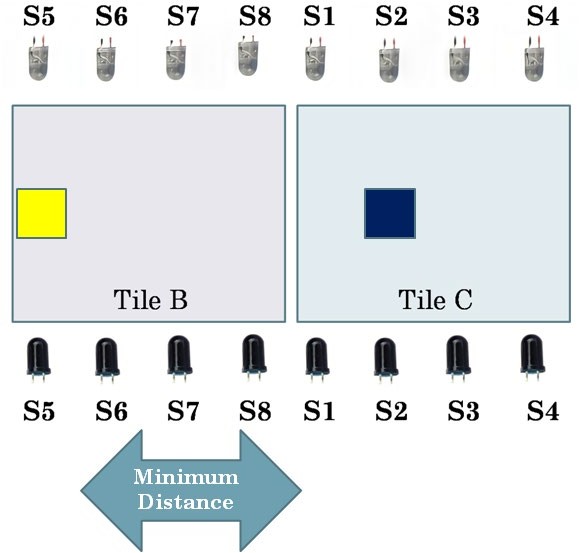
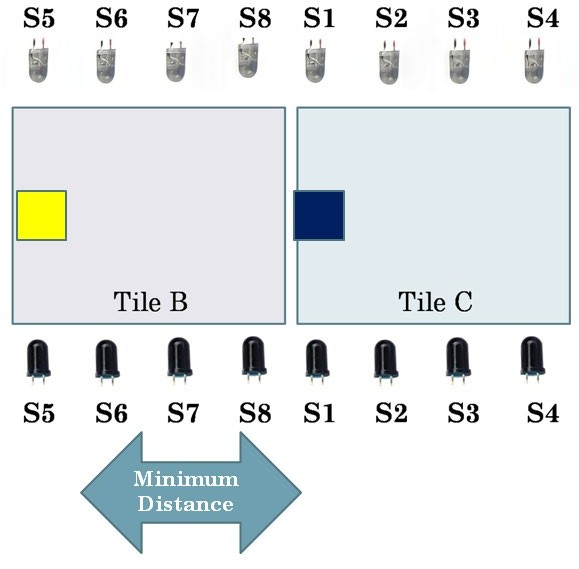
Similarly, we can explained that if one person is at a position detected by sensor S4(fig(3.3.g)) and other is at S5 or S6 or S7, then due to having a distance less than 4cm between S4 and S5, S4 and S6, S4 and S7, alarm will be turned ‘On’ alerting the people

whereas if the other person moves to S8, there won’t be any alarm as now distance between S4 and S8 is greater than 4cm.

In short: If **S4=1 and S5=1** or If **S4=1 and S5=1** or If **S4=1 and S5=1** , then show Crowd Detected. Else No alarm.

CASE 2(b): Crowd Detection between two tiles (Tile B and Tile C)

Here we will understand the algorithm when one person is on Tile B and another person is standing on Tile C. We will be changing the position of the person standing on Tile C with respect to the person on Tile B.



* 1. b)

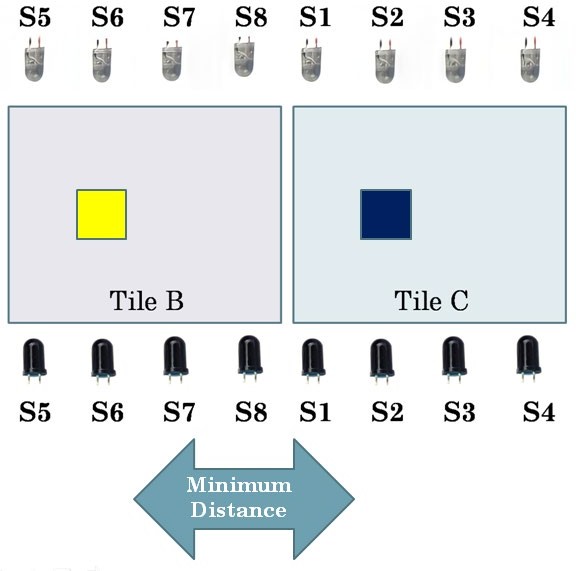
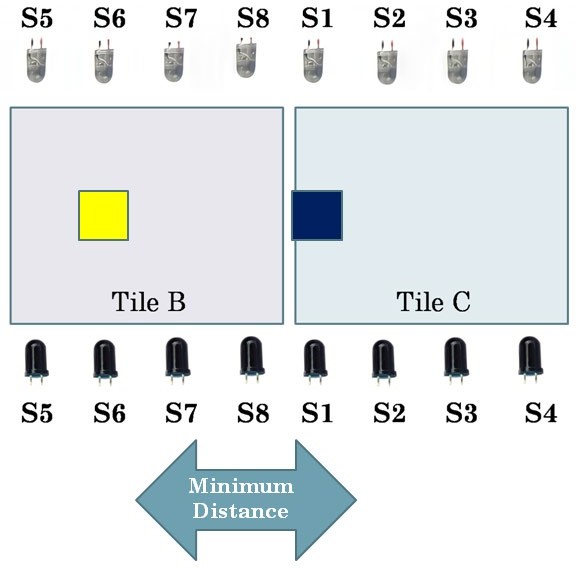
fig(3.3.h): cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S5.

From the above figure we can observe that if a person is at position S5 of Tile B and another person is at position S1 of Tile C , then due to having a distance larger than 4cm between sensor S5 and S1, alarm will be ‘OFF’. Similarly,for position of the other person detected by sensor S2, S3, S4 the alarm will be in OFF state.

In short: If **S5=1 and S1=1** or If **S5=1 and S2=1** or If **S5=1 and S3=1** or If **S5=1 and S4=1**, There won’t be any alarm.

Suppose the Person at S5(fig(3.3.e)) moves to sensor S6 and is detected by S6 now and the other person is on Tile C is at a position detected by sensor S1. As the distance between S6 and S1 is smaller then 4cm, therefore there will be an alarm. Now the other person moves

from S1 to S2. The distance between S6 and S2 is greater than 4cm ,hence no alarm will be activated.Similarly the distance between S6 and S3 and S6 and S4 will have distance greater than 4cm , so alarm will be off for these cases.

In short: If **S6=1 and S1=1** , then show Crowd Detected Else Alarm will be off

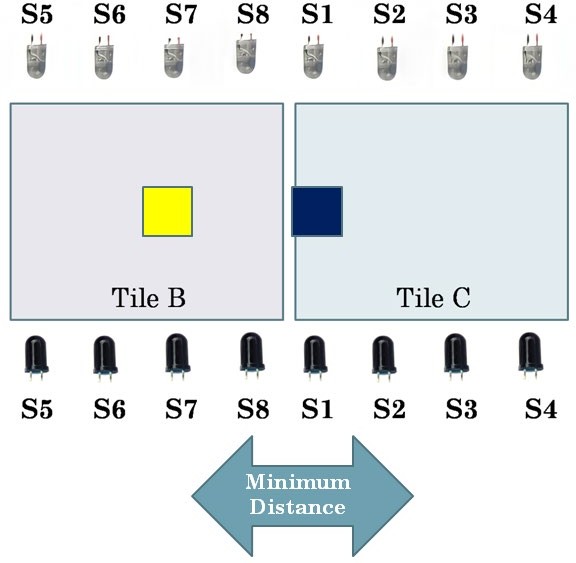
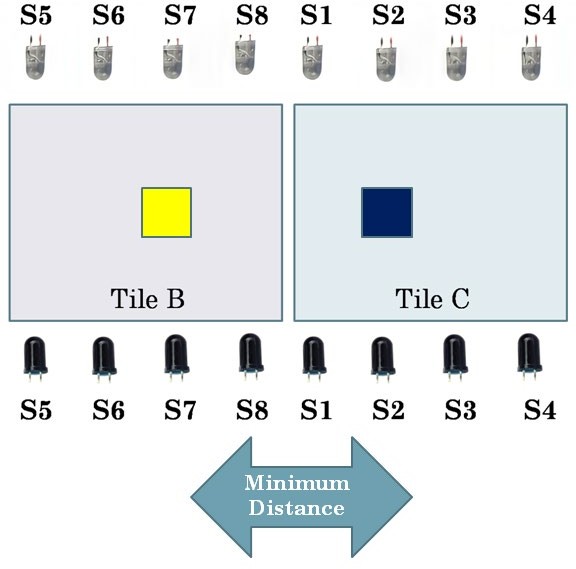
a) b)

fig(3.3.i): cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S6.

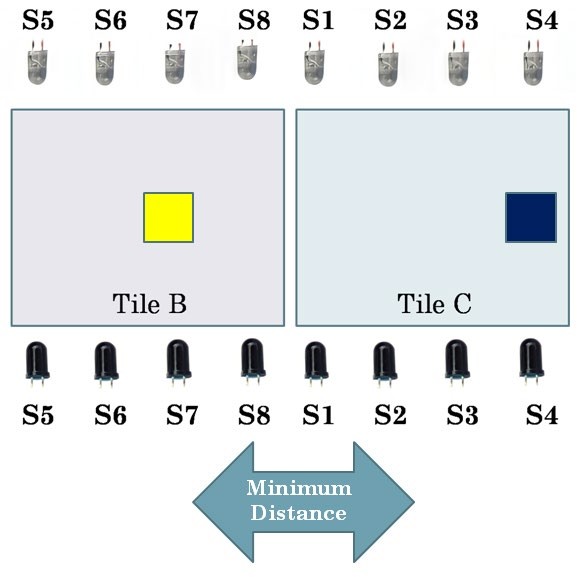
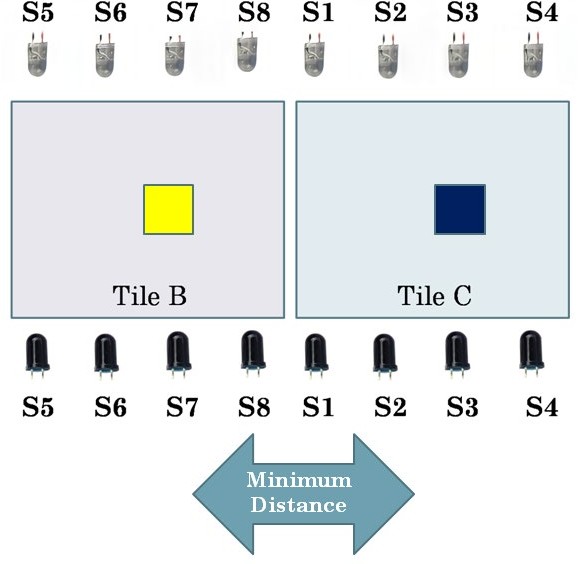
Suppose now the person standing on Tile B(fig(3.3.j)) moves to position S7 from S6 and another person is at the position detected by sensor S1, clearly the distance will be less than 4cm, hence alarm will be turned on. Similarly when the other person moves from S1 to S2 position, alarm will still turn ‘ON’. Now when the person moves to position S3 from S2 , due to greater than 4cm distance between S7 and S3, alarm will not turn ‘ON’ and same can be explained for distance S7 and S4.

In short: If **S7=1 and S1=1** or If **S7=1 and S2=1** , then show Crowd Detected.

Else No alarm.

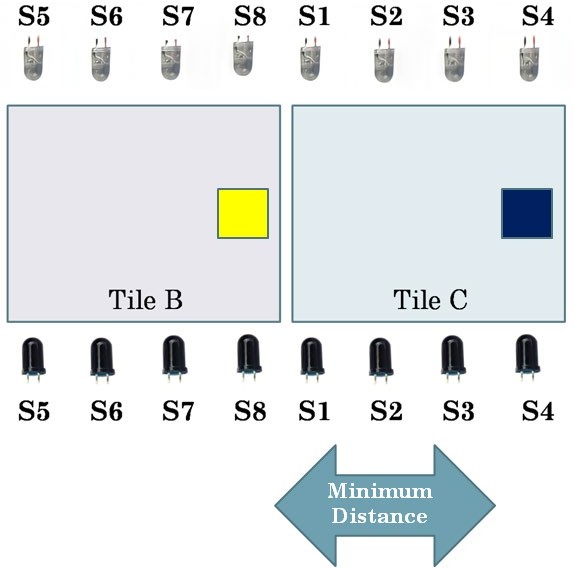
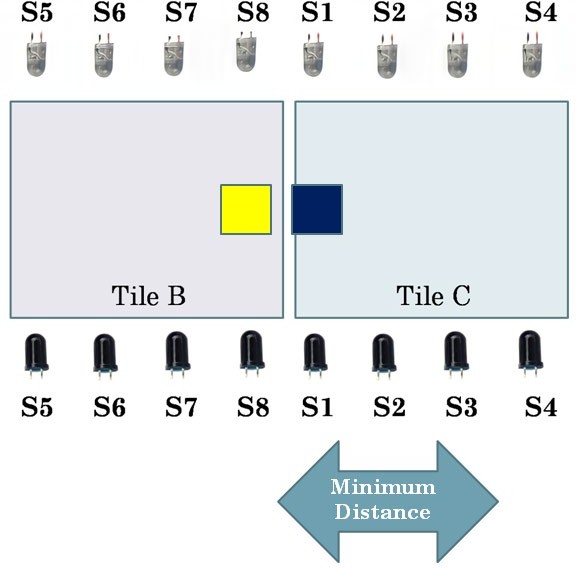
1. b)



* 1. d)

fig(3.3.j): cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S7.

Similarly, we can explained that if one person is at a position detected by sensor S8(fig(3.3.k)) and other is at S1 or S2 or S3, then due to having a distance less than 4cm between S8 and S1, S8 and S2, S8 and S3, alarm will be turned ‘On’ alerting the people whereas if the other person moves to S4, there won’t be any alarm as now distance between S8 and S4 is greater than 4cm.

In short: If **S8=1 and S1=1** or If **S8=1 and S2=1** or If **S8=1 and S3=1** , then show Crowd Detected. Else No alarm.

* + 1. b)

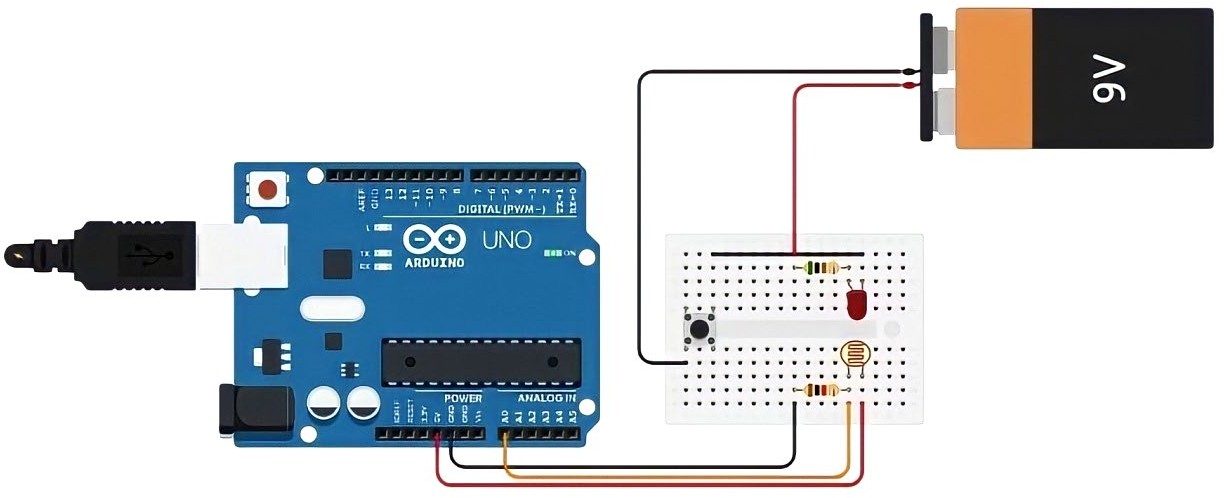
fig(3.3.k): cases for a person moving to different positions on Tile C w.r.t person on Tile B detected by sensor S8.

That was all about the algorithm part.

* 1. : *Making an equivalent for presssure sensor.*

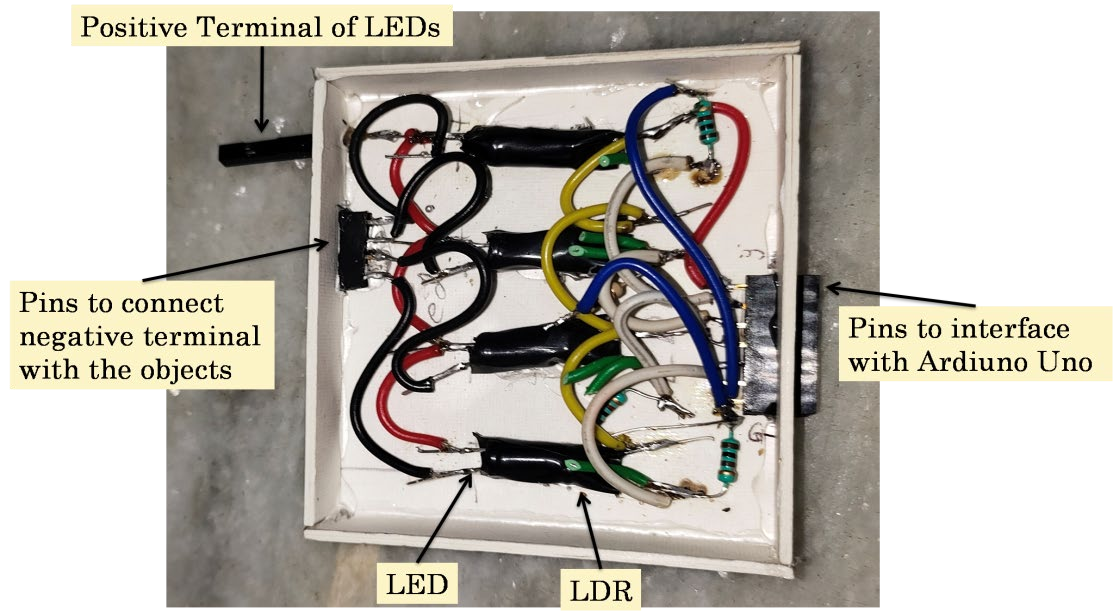
Rather then using a pressure sensor, we will be using an equivalent logic that a pressure sensor will do. There are two major reasons why we will not use pressure sensor directly in our miniaturized model:

1. A high amount of pressure is required to indicate an action which may be difficult to achieve in our miniature model.
2. Cost of pressure sensor is relatively high.



fig(3.4.a): presssure sensor equivalent circuit

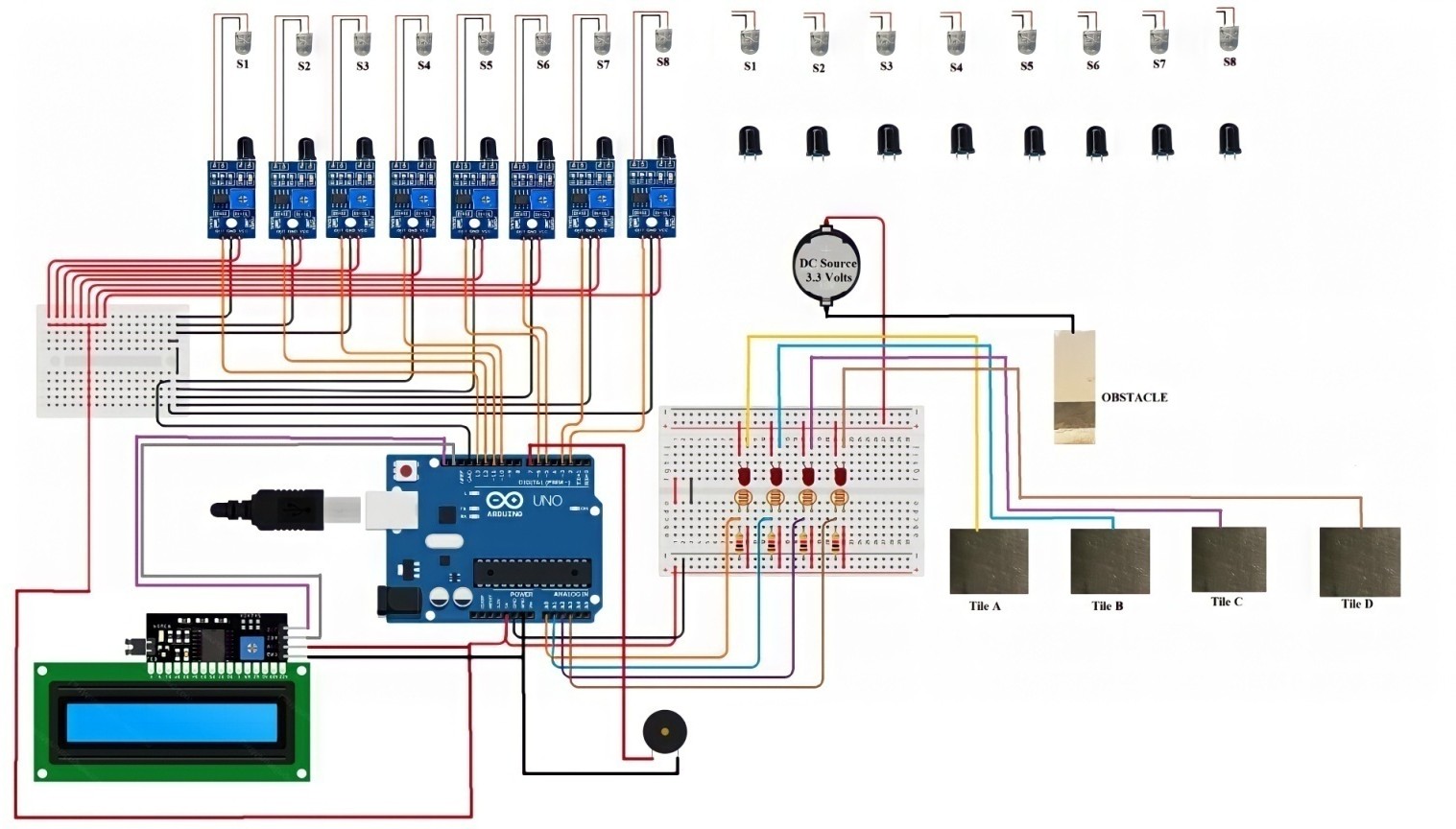
From fig(3.4.a) we can observe that an LDR is placed head on to an LED. So whenever light from LED falls onto the LDR, the value of LDR increases and when there is no light , the LDR value decreases. So, using this concept, we made a simple circuit where one terminal of power supply will be linked to an object that will serve as a person in our demonstration of narrow model and another terminal will be linked to a LED. In total we would have 4 sets of LED-LDR pairs to represent each of the four Tiles. At the same time 4 LDRs are connected to analog pins A0, A1, A2, A3 of arduino Uno board. Thus whenever an object will come in contact with the tile, the circuit of LED will be completed and so LED will glow and due to this the corresponding LDR connected to that LED will have a higher value and when there is no contact between the object and any of the Tiles, circuit will remain open, so no LED will light up and so LDR will have lower values.



fig(3.4.b): presssure sensor equivalent circuit made using mentioned components

* 1. : *Execution of Narrow model*

From fig(3.5.a) we have the overall circuit diagram of narrow model. Here the arduino Uno is controlling the entire circuit. We have used 8 IR sensor modules where each IR sensor module is having 2 pairs of IR leds(emitter) and photodiodes(receiver). The IR sensor modules are connected to pin 2, 3, 5, 6, 10, 11, 12, 13 of arduino uno. LCD with I2C has four terminals namely SDA, SCL, GND, VCC. SDA of I2C is connected to SDA of arduino. SCLof I2C is connected to SCL of arduino. GND and VCC of I2C is connected to GND and 5v of arduino respectively. A buzzer is connected to arduino’s pin number 7. As already mentioned in section 3.4 about the interfacing of pressure equivalent logic so therefore we won’t repeat that here again.



fig(3.5.a): Circuit diagram of Narrow Model Arduino Program for Narrow Model

// initialization of 4 ir sensors 8 i/o pins

const int ir1 = 13; //A1 const int ir2 = 12; //A2 const int ir3 = 11; //A3 const int ir4 = 10; //A0 const int ir5 = 6; //6 const int ir6 = 5; //5 const int ir7 = 3; //3 const int ir8 = 2; //2

//

// initialisation of 4 ldr readings 5 i/o pins

int X = 150; //threshold value for LDRs

int valA; int valB; int valC; int valD;

//

// initialization of LCD and I2C module 2 pins SAD & SCL

#include <Wire.h>

#include <LiquidCrystal\_I2C.h>

LiquidCrystal\_I2C lcd(0x27, 20, 4); // setting of the LCD address to 0x27 for a 16 chars and 2 line display

//=================================================================

void setup() { // setup declaration of input/output pins pinMode(ir1, INPUT);

pinMode(ir2, INPUT); pinMode(ir3, INPUT); pinMode(ir4, INPUT); pinMode(ir5, INPUT); pinMode(ir6, INPUT); pinMode(ir7, INPUT); pinMode(ir8, INPUT);

lcd.init(); // initialize the lcd lcd.init();

// Print a message to the LCD. lcd.backlight();

Serial.begin(9600);

pinMode(7, OUTPUT); // Buzzer to alert people for maintaining distance

}

//=================================================================

//FUNCTION SETUPS

void LCDshow() //alert the people for maintaining social distance

{

lcd.setCursor(0, 0); lcd.print("CROWD DETECTED!!");

digitalWrite(7, HIGH); delay(600);

lcd.clear();

lcd.setCursor(0, 0); lcd.print("PLEASE MAINTAIN");

lcd.setCursor(0, 1); lcd.print("SOCIAL DISTANCE");

delay(600); lcd.clear();

}

void check1() //to check social distancing within a tile(A & C)

{

if (digitalRead(13) == 1 && digitalRead(11) == 1 || digitalRead(13) == 1 && digitalRead(10) == 1 || digitalRead(12) == 1 && digitalRead(10) == 1)

{

LCDshow();

}

else

{

digitalWrite(7, LOW); lcd.clear();

}

delayMicroseconds(1);

}

//

void check2() //to check social distancing within a tile(B & D)

{

if (digitalRead(6) == 1 && digitalRead(3) == 1 || digitalRead(6) == 1 && digitalRead(2) == 1 || digitalRead(5) == 1 && digitalRead(2) == 1)

{

LCDshow();

}

else

{

digitalWrite(7, LOW); lcd.clear();

}

delayMicroseconds(1);

}

//

void select1() //function for crowd detection on multiple tiles(A & B or C & D)

{

if (digitalRead(12) == 1 && digitalRead(6) == 1 || digitalRead(11) == 1 && digitalRead(6)

== 1 || digitalRead(11) == 1 && digitalRead(5) == 1 || digitalRead(10) == 1 && digitalRead(5) == 1 || digitalRead(10) == 1 && digitalRead(3) == 1)

{

LCDshow();

}

else

{

digitalWrite(7, LOW); lcd.clear();

}

delayMicroseconds(1);

}

//

void select2() //function for crowd detection on multiple tiles(B & C)

{

if (digitalRead(5) == 1 && digitalRead(13) == 1 || digitalRead(3) == 1 && digitalRead(13)

== 1 || digitalRead(3) == 1 && digitalRead(12) == 1 || digitalRead(2) == 1 && digitalRead(12) == 1 || digitalRead(2) == 1 && digitalRead(11) == 1)

{

LCDshow();

}

else

{

digitalWrite(7, LOW); lcd.clear();

}

delayMicroseconds(1);

}

//

void check\_all1() // function for smooth working between Tile A and Tile C due to sharing

// of common sensors

{

if (digitalRead(10) == 1 && digitalRead(13) == 1 || digitalRead(13) == 1 &&

digitalRead(12) == 1 || digitalRead(13) == 1 && digitalRead(11) == 1 || digitalRead(12) == 1 && digitalRead(11) == 1 || digitalRead(12) == 1 && digitalRead(10) == 1 || digitalRead(11)

== 1 && digitalRead(10) == 1)

{

digitalWrite(7, LOW); lcd.clear();

}

}

//

void check\_all2() // function for smooth working between Tile B and Tile D due to sharing

// of common sensors

{

if (digitalRead(6) == 1 && digitalRead(5) == 1 || digitalRead(6) == 1 && digitalRead(3) == 1 || digitalRead(6) == 1 && digitalRead(2) == 1 || digitalRead(5) == 1 && digitalRead(3) == 1 || digitalRead(5) == 1 && digitalRead(2) == 1 || digitalRead(3) == 1 && digitalRead(2) == 1)

{

digitalWrite(7, LOW); lcd.clear();

}

}

//

void main\_program() // main program

{

if (valA >= X ) //condition for Tile A

{

if (valB >= X )

{

select1();

}

else if (valB >= X && valC >= X)

{

select1(); select2(); check\_all1();

}

else

{

check1();

}

}

if (valB >= X ) //condition for Tile B

{

if (valC >= X )

{

select2();

}

else if (valD >= X )

{

check\_all2();

}

else {

check2();

}

}

if (valC >= X ) //condition for Tile C

{

if (valD >= X )

{

select1();

}

else if (valA >= X)

{

check\_all1();

}

else

{

check1();

}

}

if (valD >= X ) //condition for Tile D

{

if (valB >= X)

{

check\_all2();

}

else

{

check2();

}

}

}

//

void display\_serial() // function for serial display to show "Tiles" and "Sensors" status

{

valA = analogRead(A0); Serial.print("Tile A: "); Serial.println(valA);

valB = analogRead(A1); Serial.print("Tile B: "); Serial.println(valB);

valC = analogRead(A2); Serial.print("Tile C: ");

Serial.println(valC);

valD = analogRead(A3); Serial.print("Tile D: "); Serial.println(valD);

Serial.print("IRSensor:1 "); Serial.println(digitalRead(13)); Serial.print("IRSensor:2 "); Serial.println(digitalRead(12)); Serial.print("IRSensor:3 "); Serial.println(digitalRead(11)); Serial.print("IRSensor:4 "); Serial.println(digitalRead(10)); Serial.print("IRSensor:5 "); Serial.println(digitalRead(6)); Serial.print("IRSensor:6 "); Serial.println(digitalRead(5)); Serial.print("IRSensor:7 "); Serial.println(digitalRead(3)); Serial.print("IRSensor:8 "); Serial.println(digitalRead(2));

}

//=================================================================

void loop() // program to reapeat

{

display\_serial(); main\_program();

if (valA >= X && valB >= X && valC >= X && valD >= X)

{

main\_program();

}

else

{

digitalWrite(7, LOW); lcd.clear();

}

}

* 1. : *Introduction to the wide model*

The major problem with narrow model is that the model can be applied only when crowd in the region is very low. It is applicable only inside houses, buildings enclosed by walls, floor and ceiling and can be used for crowd detection only in linear way (unidirectional)(hence the name).

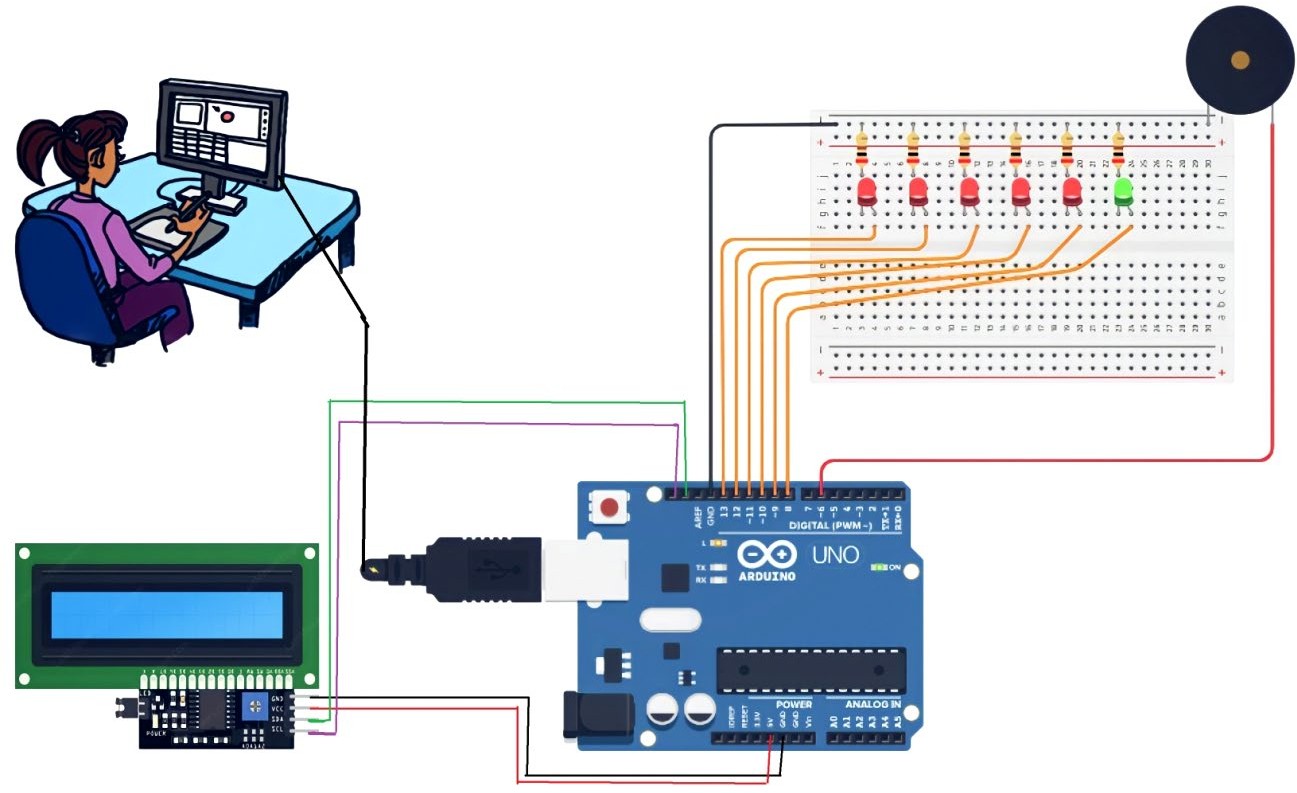
Making a survallaince system using microcontroller alone is a tough job. Microcontroller is a very primitive idea and the projects related to microcontroller are simple and are used, where number of executions is limited. Here in wide model our objective is to remove the disadvantages that were found in the narrow model. Here we will be observing crowd through cameras and when crowd is detected, the system will be alerted and will send a signal to the microcontroller board, which inturn will raise the alarm. Interfacing a camera with a microcontroller is a tedious process and also programming the camera in such a way that it can detect crowd is a more difficult process, thus we will be using OpenCV ( Open Source Computer Vision Library) in python along with our microcontroller to execute wide model.

OpenCV is an open source software library for computer vision and machine learning. OpenCV was created to provide a shared infrastructure for applications for computer vision and to speed up the use of machine perception in consumer products. Using OpenCV it becomes easy to do complex tasks such as identify and recognise faces, identify objects, classify human actions in videos, track camera movements, track moving objects, extract 3D object models and many more. Using OpenCV, complex tasks such as face detection becomes easy to implement and since pre-trained models that are capable of detecting faces, noses, and eyes are included in the OpenCV package, we don’t need to train any classifier.

Flow of wide model:

The idea of wide model is to use a survaillaince camera which will monitor the number of faces of people. The system will be programmed in such a way that when more than the required number of faces appear infront of the camera, the system will send a signal to the microcontroller , which in turn will set the alarm “ON”. Also this model will tell the density of crowd, that is the number of people that the camera has detected and this number will be displayed on the Screen to led the people know for maintaining a safe distance from each other.

* 1. : *Explanation of program and execution of wide model*

For this model we have two codes (programs), one for the arduino board and another written in python language by importing open cv module for the system(computer). We will be using the inbuild webcam of our computer. The arduino code will be based on the logic that whenever it receives a specific character from the computer , the code will tell the arduino to execute a task (task can be display meassage on LCD , Turn ‘ON’ the buzzer, turn on the number of LEDs based on crowd density) which is specifically made for doing that task. Now how will the system know which character it has to pass? Thus for this, we have the python program, through the python program the webcam will turn on , and through the webcam number of faces will be monitored. For a range of number of faces we have provided specific characters and these characters will be send to the arduino board , which in turn will do the task based on the character received by the microcontroller.

fig(3.7.a): Circuit diagram of Wide Model Arduino Program for Wide model

#include <Wire.h>

#include <LiquidCrystal\_I2C.h> // initialized the I2C-LCD

LiquidCrystal\_I2C lcd(0x27, 16, 2); // 16 x 2

int buzzer = 6; // initialized buzzer

int ledG = 13; // green LED for No Crowd

int led1 = 12; // Red LED for Crowd detected

int led2 = 11; int led3 = 10; int led4 = 9; int led5 = 8;

unsigned long newMillis; unsigned long currentMillis;

void setup() { // declaration of pins of all the

pinMode(buzzer, OUTPUT); // components pinMode(ledG, OUTPUT);

pinMode(led1, OUTPUT); pinMode(led2, OUTPUT); pinMode(led3, OUTPUT); pinMode(led4, OUTPUT); pinMode(led5, OUTPUT);

lcd.init(); // initialize the lcd lcd.init();

// Print a message to the LCD. lcd.backlight(); Serial.begin(9600);

delay(2000); newMillis = millis();

}

void LCD\_show() // function declaration for LCD

{

lcd.clear(); lcd.setCursor(0, 0);

lcd.print("Crowd Detected"); delay(1000);

lcd.clear();

}

void loop() {

if (Serial.available() > 0)

{

char ser = Serial.read();

currentMillis = millis();

if ((currentMillis - newMillis > 150) && (ser == 's')) // if “s” is

{ // recieved

digitalWrite(buzzer, LOW); digitalWrite(ledG, HIGH);

digitalWrite(led1, LOW); digitalWrite(led2, LOW); digitalWrite(led3, LOW); digitalWrite(led4, LOW); digitalWrite(led5, LOW); lcd.clear();

newMillis = millis();

}

else if ((currentMillis - newMillis > 150) && (ser == 'c')) // if “c” is

{ // recieved

digitalWrite(buzzer, HIGH); digitalWrite(ledG, LOW); digitalWrite(led1, HIGH); digitalWrite(led2, LOW); digitalWrite(led3, LOW); digitalWrite(led4, LOW); digitalWrite(led5, LOW); LCD\_show(); lcd.setCursor(1, 0); lcd.print("No. of People"); lcd.setCursor(1, 1);

lcd.print("Detected: 5-9"); delay(1000);

newMillis = millis();

}

else if ((currentMillis - newMillis > 150) && (ser == 'm')) // if “m” is

{ // recieved

digitalWrite(buzzer, HIGH); digitalWrite(ledG, LOW); digitalWrite(led1, HIGH); digitalWrite(led2, HIGH); digitalWrite(led3, LOW); digitalWrite(led4, LOW); digitalWrite(led5, LOW); LCD\_show(); lcd.setCursor(1, 0); lcd.print("No. of People"); lcd.setCursor(1, 1);

lcd.print("Detected: 10-14"); delay(1000);

newMillis = millis();

}

else if ((currentMillis - newMillis > 150) && (ser == 'n')) // if “n” is

{ // recieved

digitalWrite(buzzer, HIGH); digitalWrite(ledG, LOW); digitalWrite(led1, HIGH); digitalWrite(led2, HIGH); digitalWrite(led3, HIGH); digitalWrite(led4, LOW); digitalWrite(led5, LOW); LCD\_show(); lcd.setCursor(1, 0); lcd.print("No. of People"); lcd.setCursor(1, 1);

lcd.print("Detected: 15-19"); delay(1000);

newMillis = millis();

}

// if “o” is

else if ((currentMillis - newMillis > 150) && (ser == 'o')) // recieved

{

digitalWrite(buzzer, HIGH); digitalWrite(ledG, LOW); digitalWrite(led1, HIGH); digitalWrite(led2, HIGH); digitalWrite(led3, HIGH); digitalWrite(led4, HIGH); digitalWrite(led5, LOW); LCD\_show(); lcd.setCursor(1, 0); lcd.print("No. of People"); lcd.setCursor(1, 1);

lcd.print("Detected: 20-24"); delay(1000);

newMillis = millis();

}

else if ((currentMillis - newMillis > 150) && (ser == 'p')) // if “p” is

{ // recieved

digitalWrite(buzzer, HIGH); digitalWrite(ledG, LOW); digitalWrite(led1, HIGH); digitalWrite(led2, HIGH); digitalWrite(led3, HIGH); digitalWrite(led4, HIGH); digitalWrite(led5, HIGH); LCD\_show(); lcd.setCursor(1, 0); lcd.print("No. of People"); lcd.setCursor(1, 1);

lcd.print("Detected: 25+ "); delay(1000);

newMillis = millis();

}

}}

Python Program for Wide model

import cv2 # import open CV

import time

import serial # serial communication

# load face detection model

face\_cascade = cv2.CascadeClassifier('haarcascade\_frontalface\_default.xml') # open camera

cap = cv2.VideoCapture(0)

crowdDef = 5 # or more , kept the crowd limit to 5 people crowdCount = 0

port = serial.Serial("COM14", 9600, timeout=1) #port.open()

time.sleep(2)

def arduino(out): port.write(out.encode())

while 1:

# reset count crowdCount = 0

# read camera frame ret, img = cap.read()

img = cv2.flip(img, 1) img2 = img.copy()

gray = cv2.cvtColor(img, cv2.COLOR\_BGR2GRAY) # detect face/people

faces = face\_cascade.detectMultiScale(gray, 1.3, 5)

# read faces

for (x,y,w,h) in faces:

# mark faces identified

cv2.rectangle(img,(x,y),(x+w,y+h),(0,0,255),3) #B,G,R , width of frame # increment faces/people count

crowdCount = crowdCount + 1

print("Crowd Count") print(crowdCount)

# if people count greater than crowd defination alert Crowd Detected if (1\*crowdDef)<= crowdCount <= (2\*crowdDef):

print("Crowd recorded") print("Crowd alert")

arduino('c') # if no. of people 5 to 10

elif (2\*crowdDef) < crowdCount <= (3\*crowdDef): print("Crowd recorded")

print("Crowd alert")

arduino('m') # if no. of people 10 to 15

elif (3\*crowdDef) < crowdCount <= (4\*crowdDef): print("Crowd recorded")

print("Crowd alert")

arduino('n') # if no. of people 15 to 20

elif (4\*crowdDef)< crowdCount <= (5\*crowdDef): print("Crowd recorded")

print("Crowd alert")

arduino('o') # if no. of people 20 to 25

elif (5\*crowdDef)< crowdCount <= (6\*crowdDef): print("Crowd recorded")

print("Crowd alert")

arduino('p') # if no. of people is more than 25

elif crowdCount < crowdDef: print("no crowd")

arduino('s') # if no crowd detected cv2.imshow('img',img)

# check key press, if q key preessed exit application if cv2.waitKey(1) & 0xFF == ord('q'):

arduino('o') port.close() break

#time.sleep(0.5)

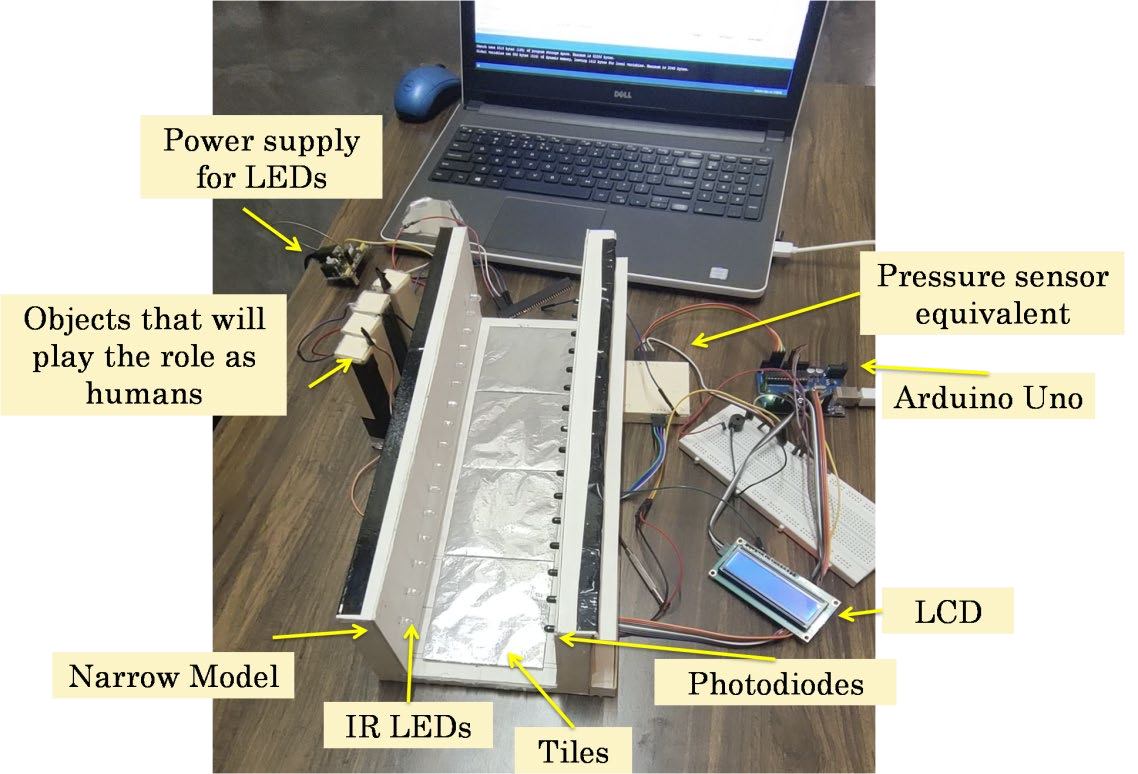
# release object created cap.release() cv2.destroyAllWindows()

In the pthon program we are using an already available classifier namely 'haarcascade\_front alface\_default.xml' to identify faces of people through the webcam. In the python program we have set a threshold value of 5 which means that if the webcam(or camera) detects more than five faces, then character ‘c’ will be transmitted to the arduino board via the computer which is communication with the microcontroller at a baud rate 9600. Once character ‘c’ is received by the arduino board , the necessary function related to character ‘c’ will be executed by the board. Now if the webcam detects 11 people(say) then the system will send character ‘m’ to the arduino and once ‘m’ is received , the necessary function related to character ‘m’ will be executed by the board. Thus according to the crowd density different characters will be send.

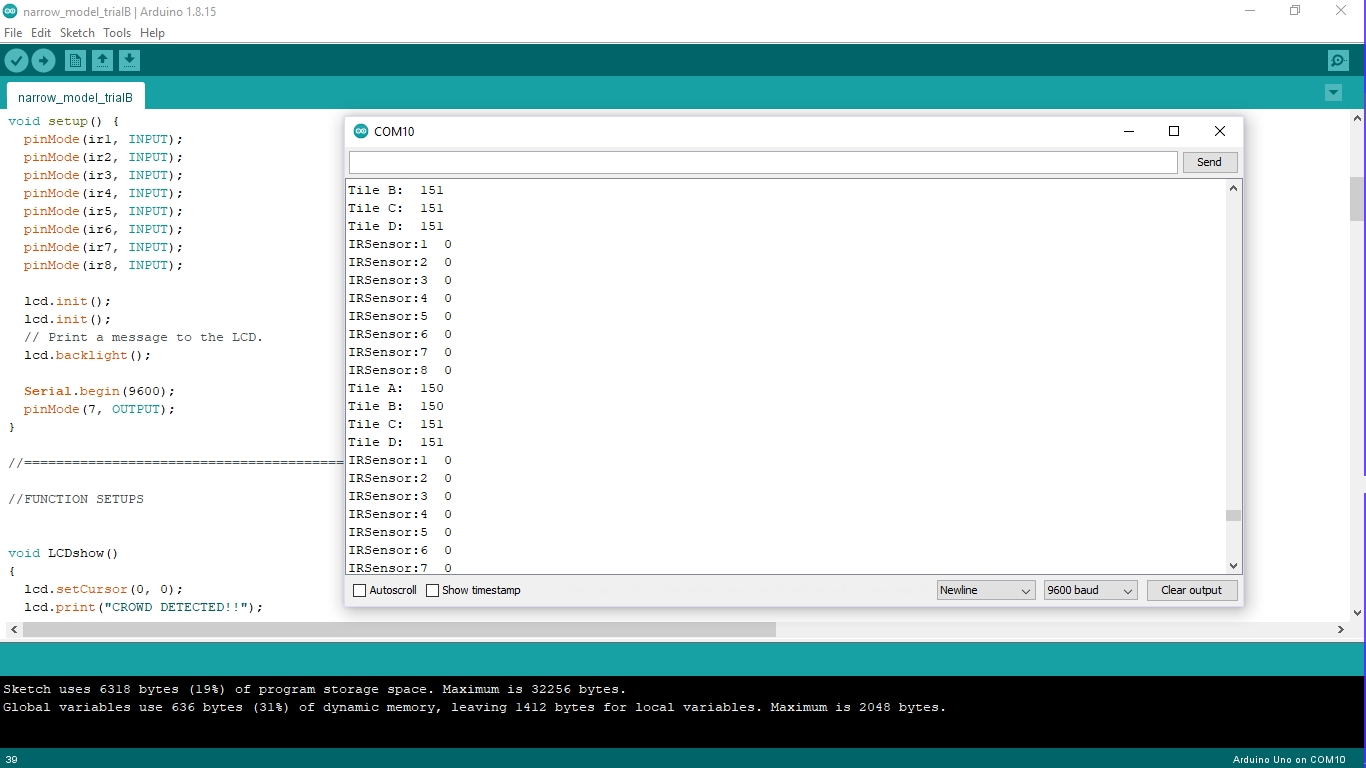
# CHAPTER 4 RESULT ANALYSIS

In this part of the report we are going to discuss about the conclusions and significant results of our objective. Section 4.1 will include the practical demonstration results of narror model. Section 4.2 will include the practical demonstration results of wide model. Section 4.3 will discuss about the problems we faced and the conclusion of both the models.

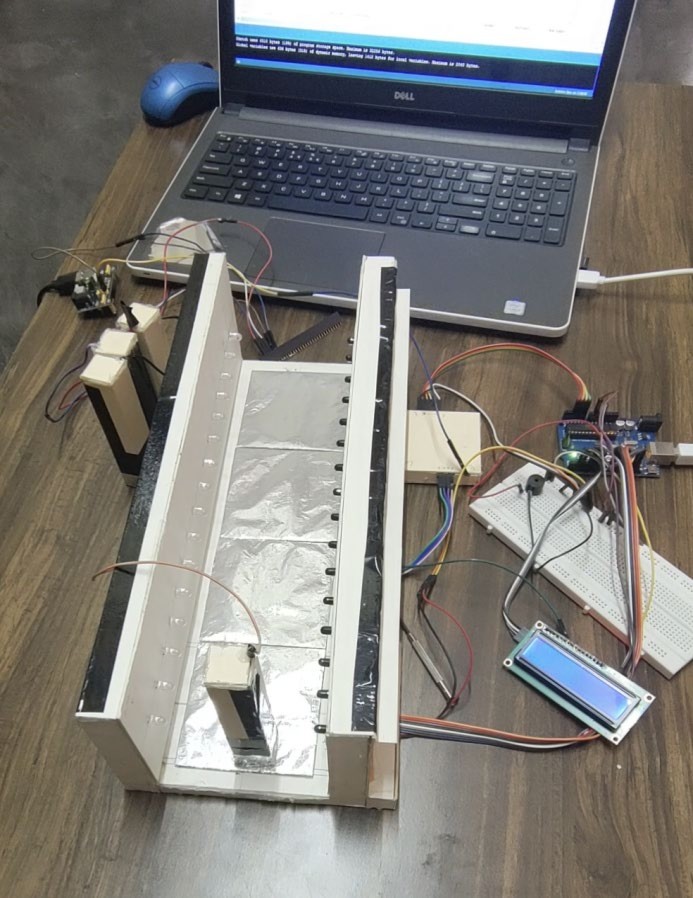
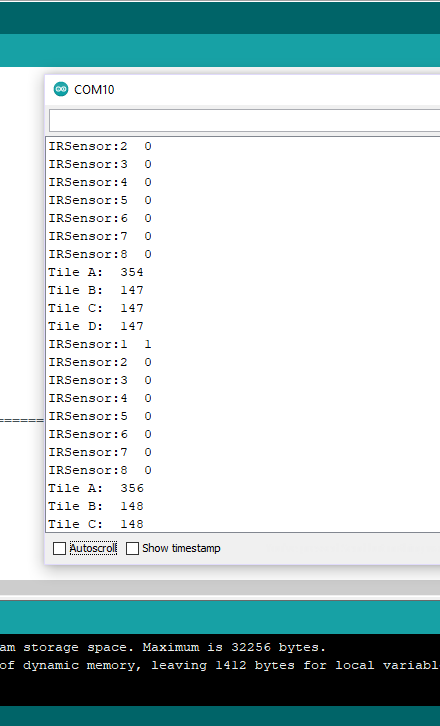
* 1. : *Practical demonstration results of narror model*



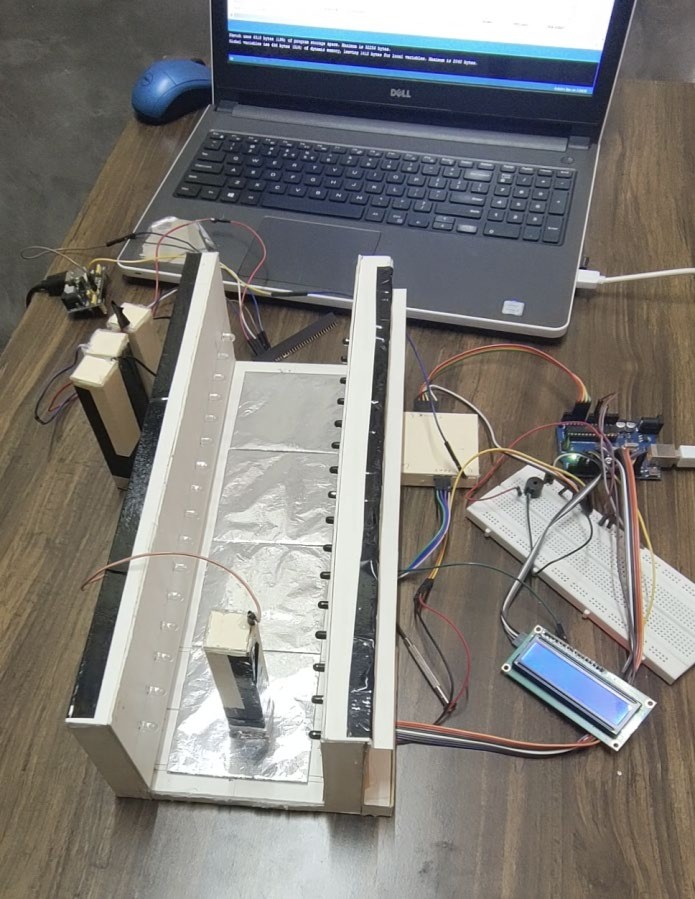
fig(4.1.a): Miniature model of Narrow model concept



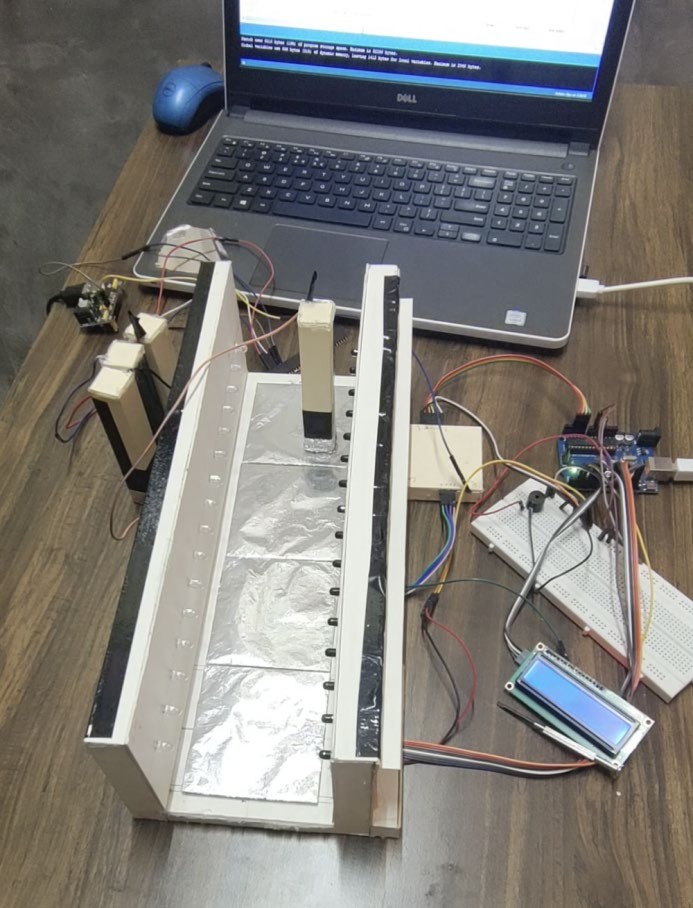
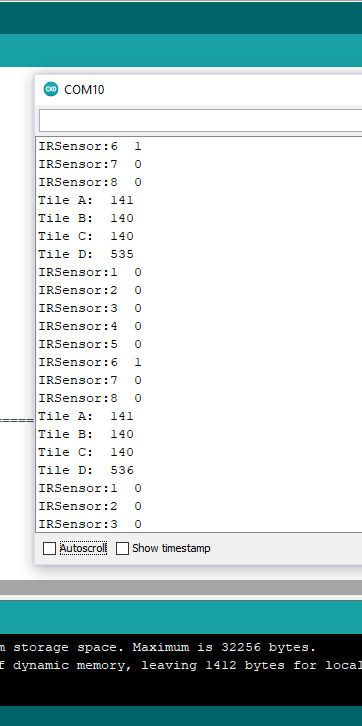
fig(4.1.b): Serial monitor when no object on the Tiles

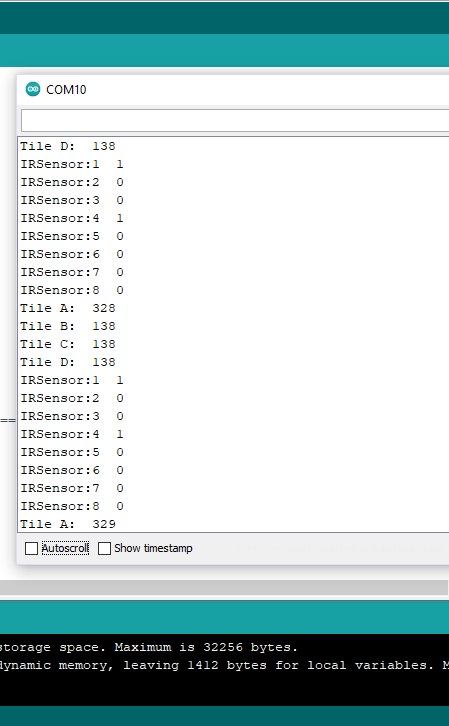
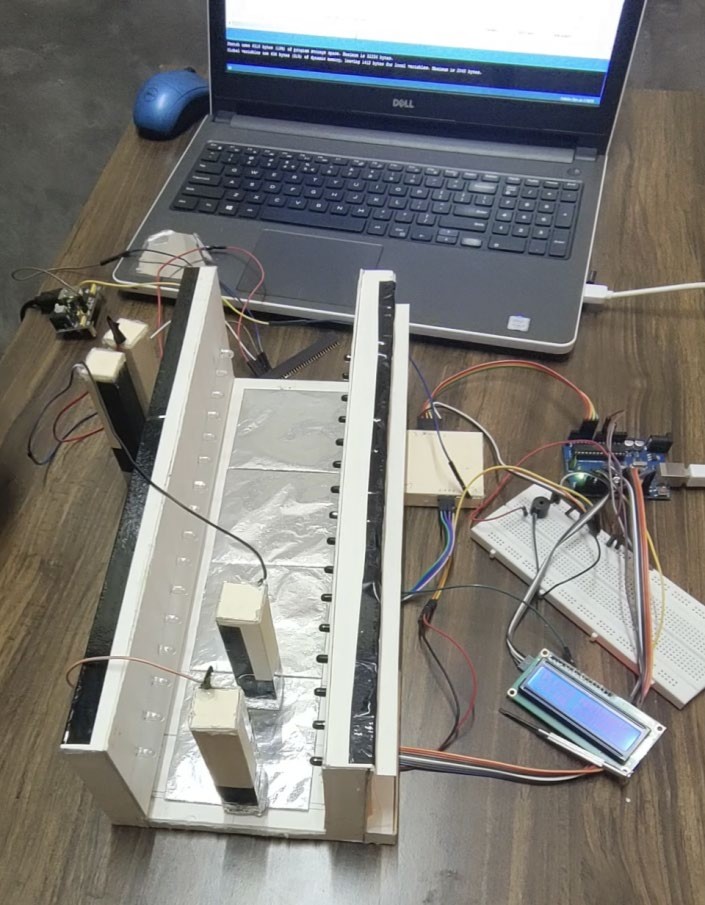
* + 1. b)



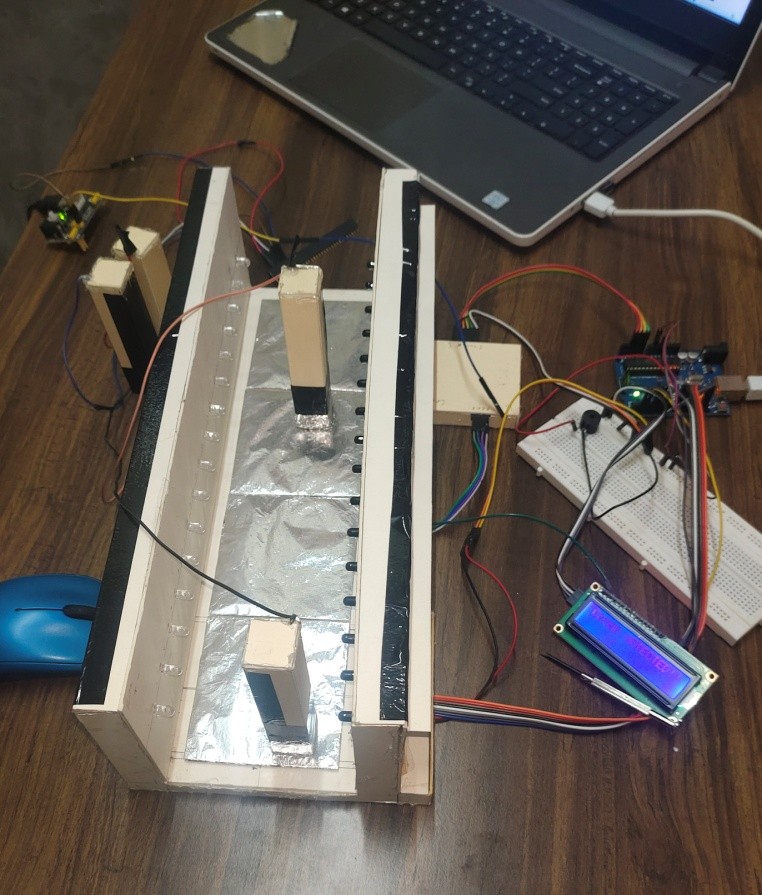
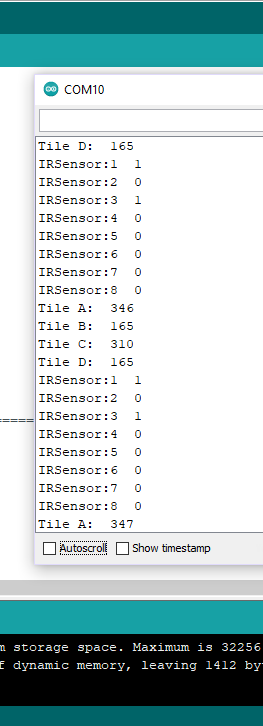
c) d)

e) f)



1. h)

* 1. j)

fig(4.1.c): readings shown in Serial monitor according to the location of object on the narrow model

From fig(4.1.a) , we can observe that when there is no object on the narrow model(on the Tiles) ,the reading of all the IR sensor modules(1-8) are ‘0’ and also all the tiles are giving a minimum value of 150-151 (from fig(4.1.b)) which is due to the fact that the LEDs of the pressure equivalent circuit are ‘OFF’ and so the LDRs are returning minimum values to the arduino board. From fig(4.1.c) case a) and b) , When an object is placed at a position that can be detected by sensor S1 on Tile A, we see that on the arduino IDE serial monitor the value of IR sensor:1 has change to ‘1’ and also Tile A value has increased due to the fact that the LED of the pressure equivalent circuit connected to Tile A is ‘ON’ and so the LDR 1 is returning values greater than the minimum value to the arduino board.

Now we move the object to a position detected by sensor S2 (fig(4.1.c) c) d))) on Tile A, we see that on the arduino IDE serial monitor the value of IR sensor:2 has change to ‘1’ and value of IR sensor: 1 has went back to ‘0’ and since the object is still on Tile A, therefore there is no change in the value of Tile A.

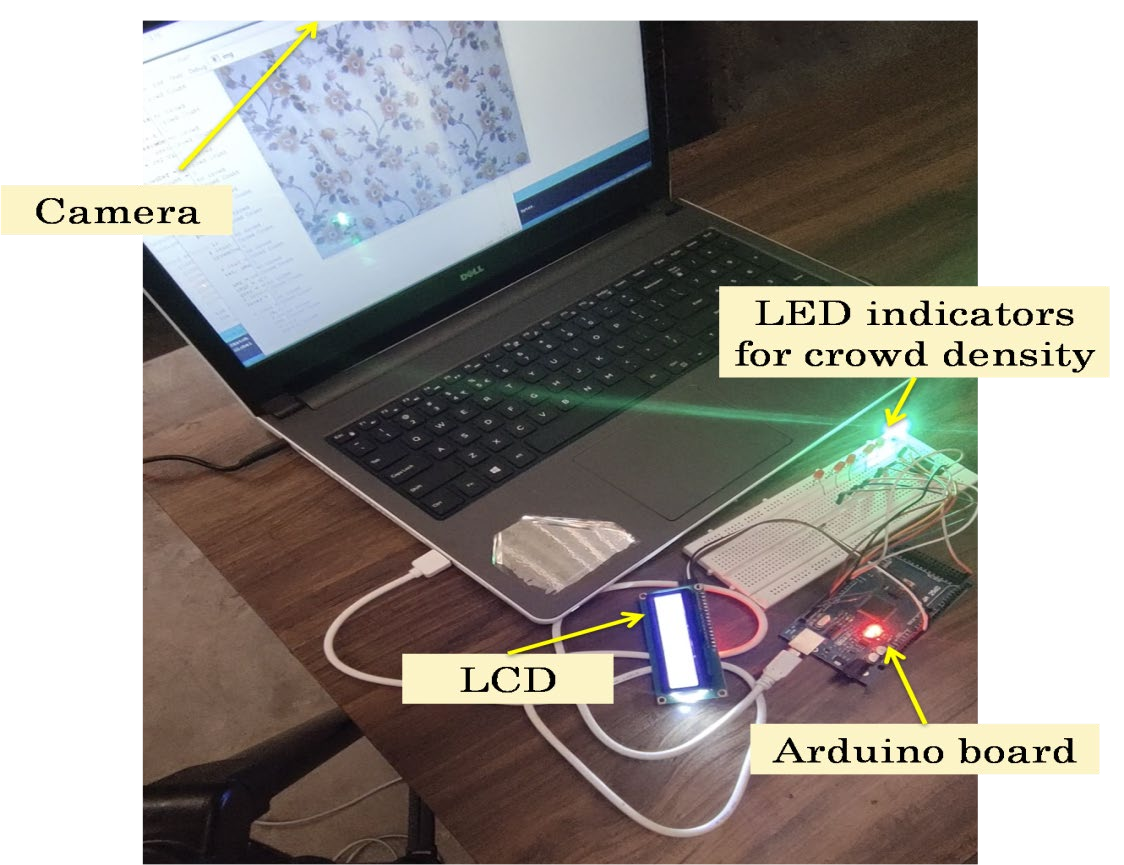
Similarly if we move the object to a position detected by sensor S6(fig(4.1.c) e) f))) on Tile D, we see that on the arduino IDE serial monitor, the value of IR sensor:6 has change to ‘1’ where rest IR sensors are still ‘0’ and since the object is on Tile D, therfore value of Tile D will be more than rest of the Tiles.

Now coming to the crowd detection, suppose now we have two people(fig(4.1.c) g) h))), both are on Tile A, one of them is detected by S1 and other by S4, due to the activation of S1 and S4 at the same time, there will be a “crowd detected” alarm. This situation belongs to case1(a) of narrow model algorithm that is crowd detection on Tile A. Similarly, the mentioned algorithm will be followed , as the program is based on the algorithm.

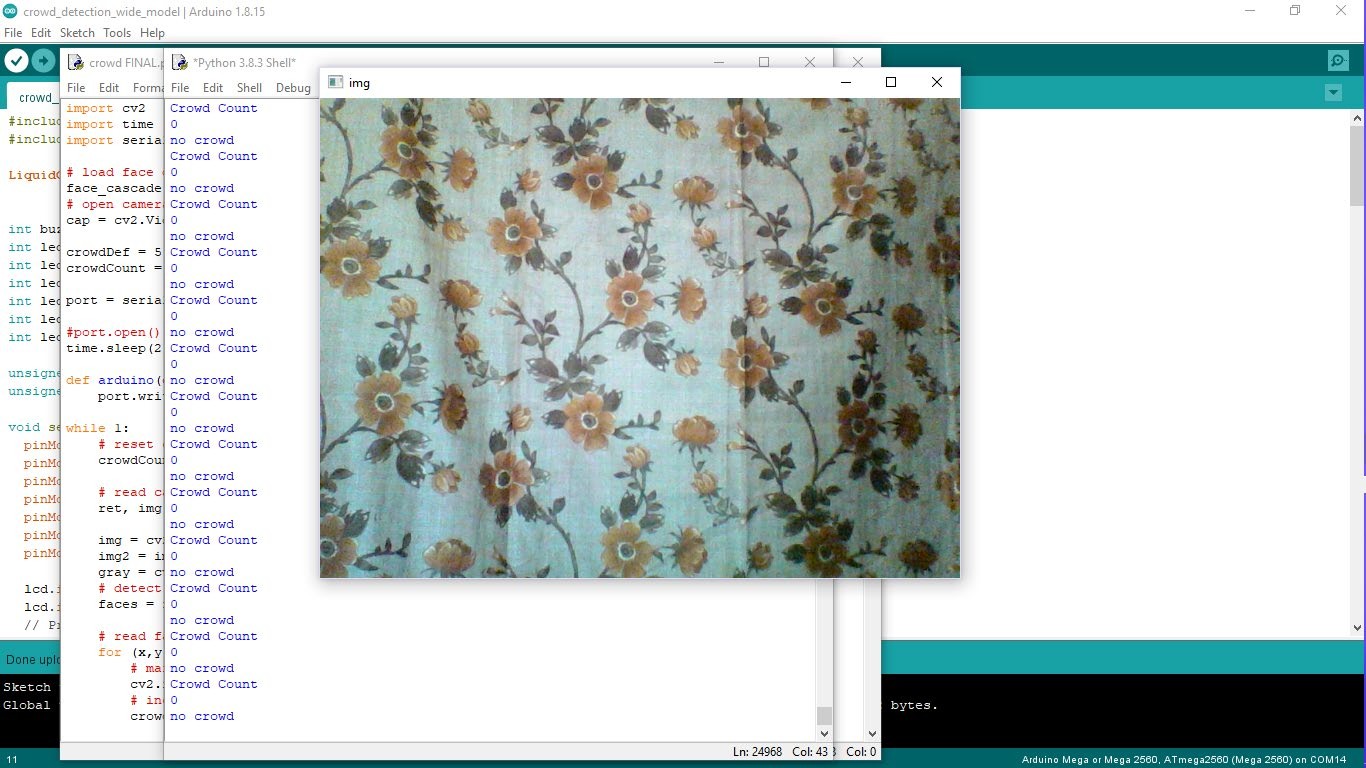
Once the alarm rings, either of the two persons have to move away from each other, thus they have to follow social distancing.

Thus we can say the narrow model is working properly according to the algorithm and program . but there are some problems that we have faced that we will mention in section 4.3.

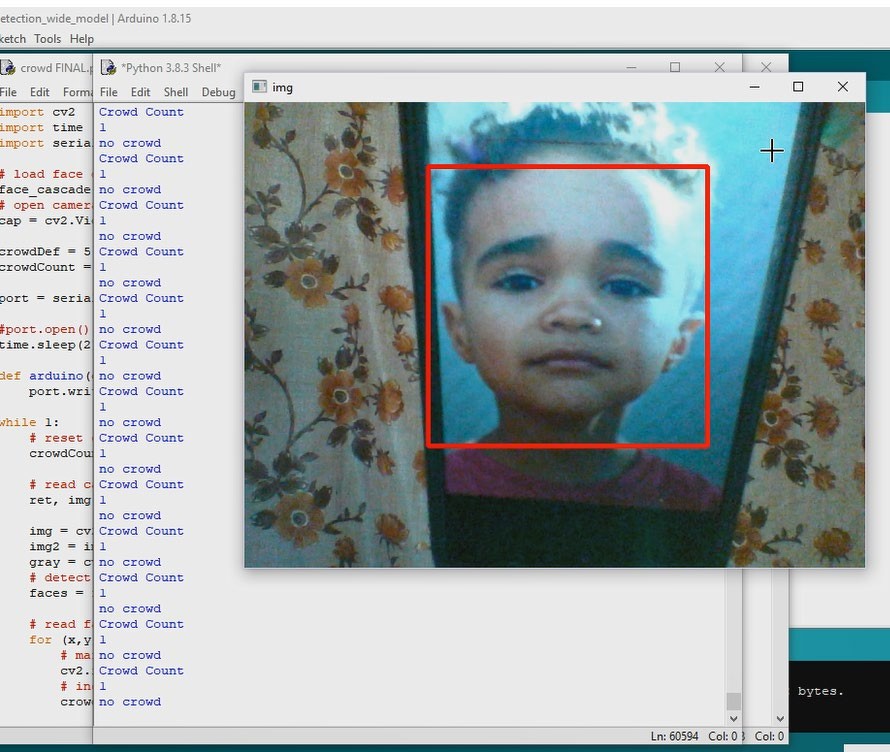
* 1. : *practical demonstration results of wide model*



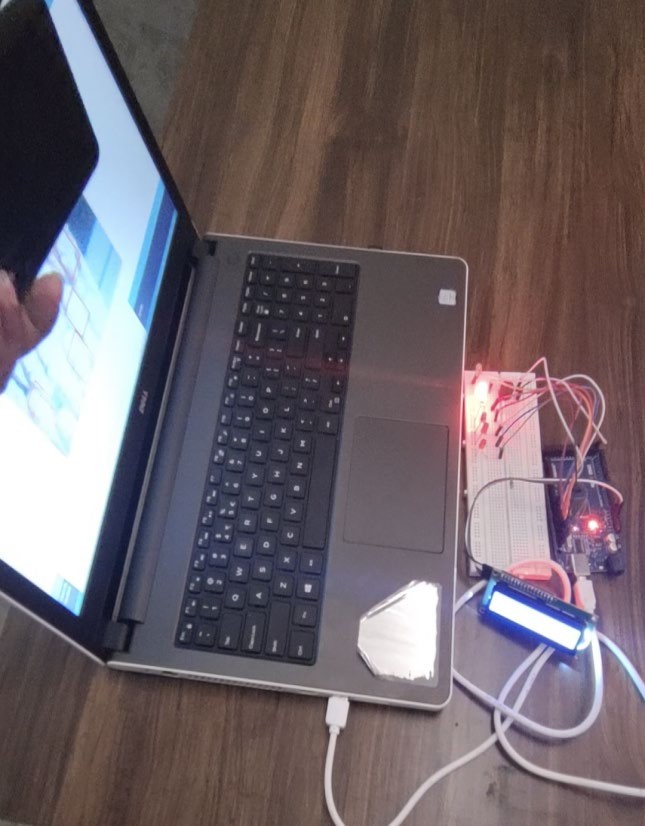
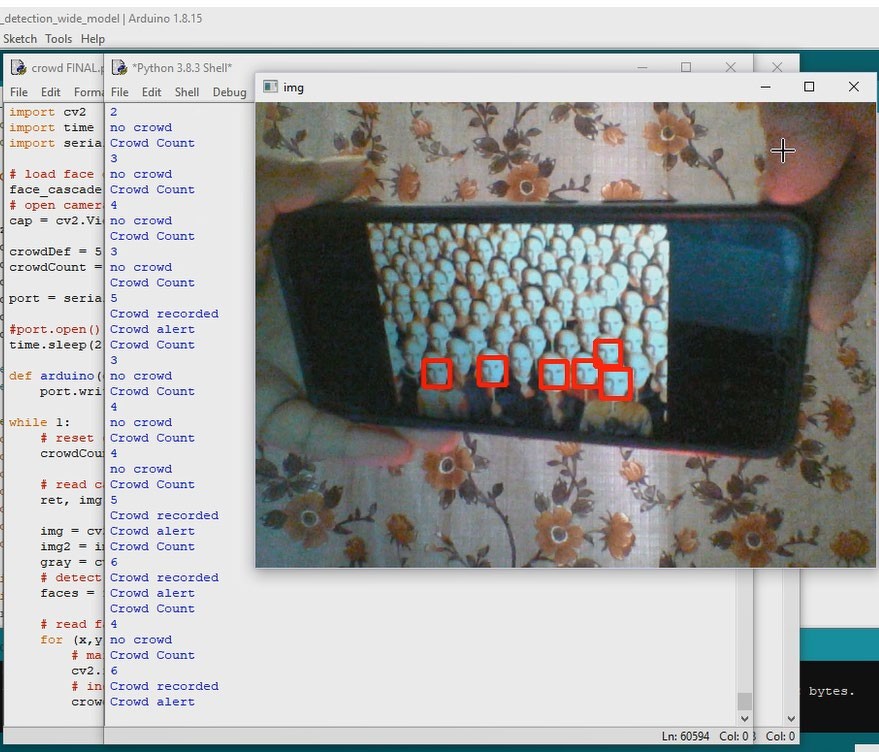
fig(4.2.a): wide model



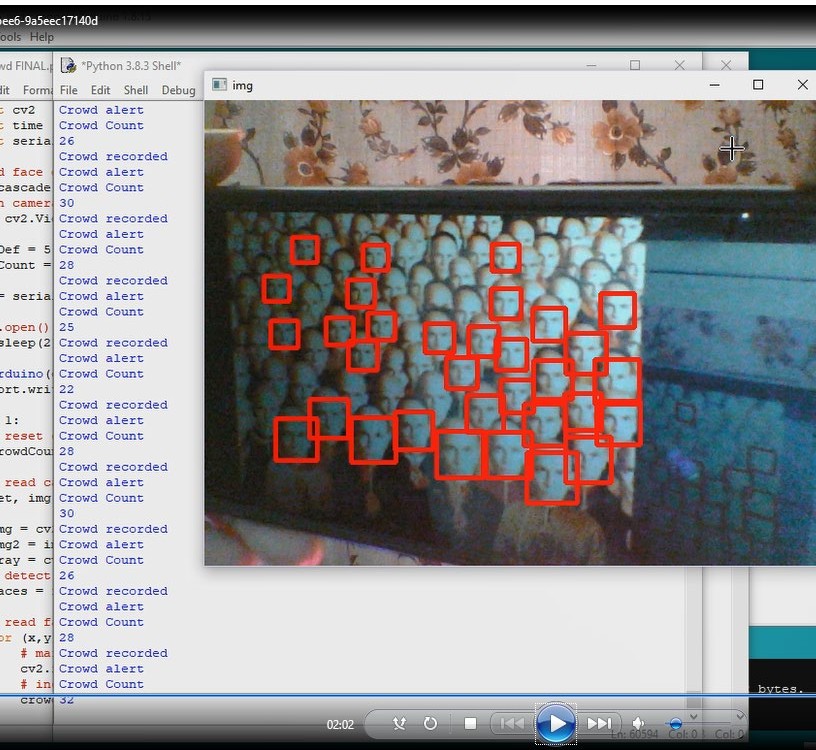
fig(4.2.b): Screen view of wide model when no faces are registered



* + 1. b)

c) d)



e) f)

fig(4.2.c): Screen views of wide model when faces are detected

The wide model uses a camera (here we are using webcam) for detecting faces and according to the number of faces we will get different outputs. From fig(4.2.b) we can observe that when no face is detected, the message ‘no crowd’ appears and on the LCD no message is shown. Also the green light indicates “NO CROWD”. If the camera detects one face (fig(4.2.c) a) b)), then the message ‘no crowd’ and crowd count =1 appears and on the LCD no message is shown and green light keeps on glowing. But when the number of faces is greater than the limit( here the minimum face required for crowd detection is set to 5), then red LEDs will glow (green LED will turn ‘OFF’), along with an alarm and crowd detection message on LCD. Additionally here we will also get the range of faces(people) detected by the camera which will be displayed both on the python cammand window as well as on the LCD.The more the number of people detected , the more the number of red LEDs will glow (fig(4.2.c)). Thus we can say the wide model is working fine according to the program .

* 1. : *problems faced and the conclusion for both the models*

Problems found in the narrow model:

1. The IR sensor modules are very sensitive, thus a lot of time is wasted on tuning their sensitivity levels. Due to this many a times, Sensors may become or turn to value ‘1’ even if there is no object in between the IR LED and photodiode.

As a conclusion to this chapter, we can say that that we faced some problems with narrow model but it can be applied practically where the pressure sensor equivalent circuit can be replaced by pressure sensors and the infrared sensors can be replaced by passive infrared sensor(PIR).

The main problem with IR sensors is that it supports shorter range and hence it’s performance degrades with longer distances. Infrared frequencies are affected by hard objects (e.g. walls, doors) , smoke, intensity of light etc. IR sensors can detect not only animal, human but also opaque objects. Whereas PIR sensor is more suitable to use practically as it detects motions accurately at any time, consumes a very less amount of energy between 0.8 W to 1.0 W and has a wide range detection (up to 10 meter).

To our knowledge in wide model, we haven’t face any problem. The model is working according to our expectations. But, this model has its disadvantages, which we will discuss in the next chapter.

# CHAPTER 5 CONCLUSIONAND FUTURE SCOPE OF WORK

* 1. : *Summary of work*

Our project title is “ Microcontroller based crowd detection system to prevent Covid-19 ”. During the making of the project we faced many issues, problems but with every problem we faced we gain a new experience. We divided our project into two parts namely narrow model and wide model. In narrow model we basically focus on the concept of linear distance that is the distance between a person and another person who is either in front or behind to the previous person. We approach the model with an algorithm and tried our best to make the algorithm more efficient and at the same time simple to understand. The major problem with the narrow model was that it can detect crowd only in a linear or one way fashion, thus it may not be so applicable in large gathering places. Thus to remove this problem we have came up with the idea of wide model. Here we will be having a camera that will monitor the number of faces and if the number of faces detected more than the allowed number, then alarm will turn ‘ON’.

* 1. : *Conclusion*

As a conclusion we can say that to some extend we have successfully achieved our goal. The narrow model is/can be applicable where more strictness is required like at the entrance gate of shopping mall, cinema complex, stores,etc . Thus, narrow can work fine, in the mentioned places. On the other hand wide model can be used for monitoring both small and large gatherings.

* 1. : *Future scope of work*

As told earlier, Microcontroller is a very limited domain. Projects related to microcontroller are simple and are used, where number of executions is limited. Thus, making a project that includes so many complexities is difficult using only microcontroller. Therefore, we took help from other domains like ‘image processing’, ‘computer vision’,etc to accomplise our objective.

We would like to work more on the topic but due to lack of time, we were unable to accomplised that goal. To the people who liked our project and wants to do or upgrade our objective can try some ideas listed below :

1. In our wide model, we made the use of 'haarcascade\_frontalface\_default.xml' classifier. This classifier is a pre-trained model that is capable of detecting faces, noses, and eyes of people. Using this classifier we wrote a python program to count the number of faces available or seen by the camera. The limitation here is that, there might be chances that although people are maintaining a safe distance and at the same time the number of faces detected by the camera is greater than the limit provided, then in such situation there may be possibilies of bugs and errors, due to which the alarm will turn “ON”. We can remove this problem by bringing an additional “distance” parameter to our python program. Just like face detection, we can also trained a model or can find an available classifier that can detect or tell the admin about the distance between each person from one another. That way with this additional condition crowd detection can be made more efficient and smoother.
2. A more innovative approach towards social distancing indicator would be to include AR (Augmented Reality) libraries from openCV and use them. Due to time restrictions, we couldn't do it but it is can be done with openCV.
3. We can also use thermal cameras for detecting the temperature of people through camera itself. With the help of theral cameras, we can create awareness among people who might be suffering from covid and prevent covid from spreading to an extend.

# REFERENCES

*Web*

[1] Covid19, <https://link.springer.com/article/10.1007/s41870-021-00658-2#Sec4>

* 1. openCV, <https://www.mygreatlearning.com/blog/opencv-tutorial-in-python/#sh20>
  2. openCV, <https://github.com/Lucifergene/Social-Distancing-Project-OpenCV>