CS226 PROJECT REPORT

-Siddharth Sanskritayan(1901CS75)

Multifunctional Bi-Directional Visitor Counter

Introduction

A Bi-Directional person/object counter is a device which can be used to monitor the number of objects entering / exiting through it. If it is set up at the entrance of a shop, it displays the number of visitors currently inside the shop. To increase its functionality, an LED is used which glows when there is at least one person inside the shop. Also whenever the number of visitors becomes greater than some threshold value, a buzzer is turned on.

Components Required

- Arduino Uno
- Two Ultrasonic Sensors
- LED
- Resistors
- Breadboard
- Passive Buzzer
- Jumper wires
- Power Source

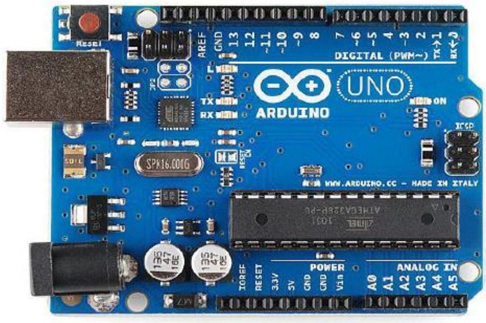
Hardware Description

Arduino UNO R3 : Arduino UNO is a microcontroller board based on the ATmega328P. It has 14 digital input/output pins in which 6 pin is used as PWM outputs, 6 as analog inputs, a 16MHz quartz crystal, a USB connection, a power jack, an ICSP header and a reset button. It can be connected to any AC-to-DC adapter to get started. Arduino comes with the IDE which is an open source platform which needs to be installed in Pc while programming and to be connected to USB port to upload the relevant code.

Ultrasonic sensor

An ultrasonic sensor is a device that can measure the distance to an object by using sound waves. It measures distance by sending out a sound wave at a specific frequency and listening for that sound wave to bounce back. For our project we are using HC-SR04 Ultrasonic sensor which emits an ultrasound at 40000Hz that travels through the air and bounce back if there is object or obstacle present in their path. Distance can be calculated considering the travel time and speed of the sound. The HC-SR04 Ultrasonic module has 4 pins, Ground, VCC, Trig, Echo. The Ground and VCC pins of the module needs to be connected to the Ground and 5V pins on the Arduino board respectively and the trig and echo pins to any digital I/O pin on Arduino Board.

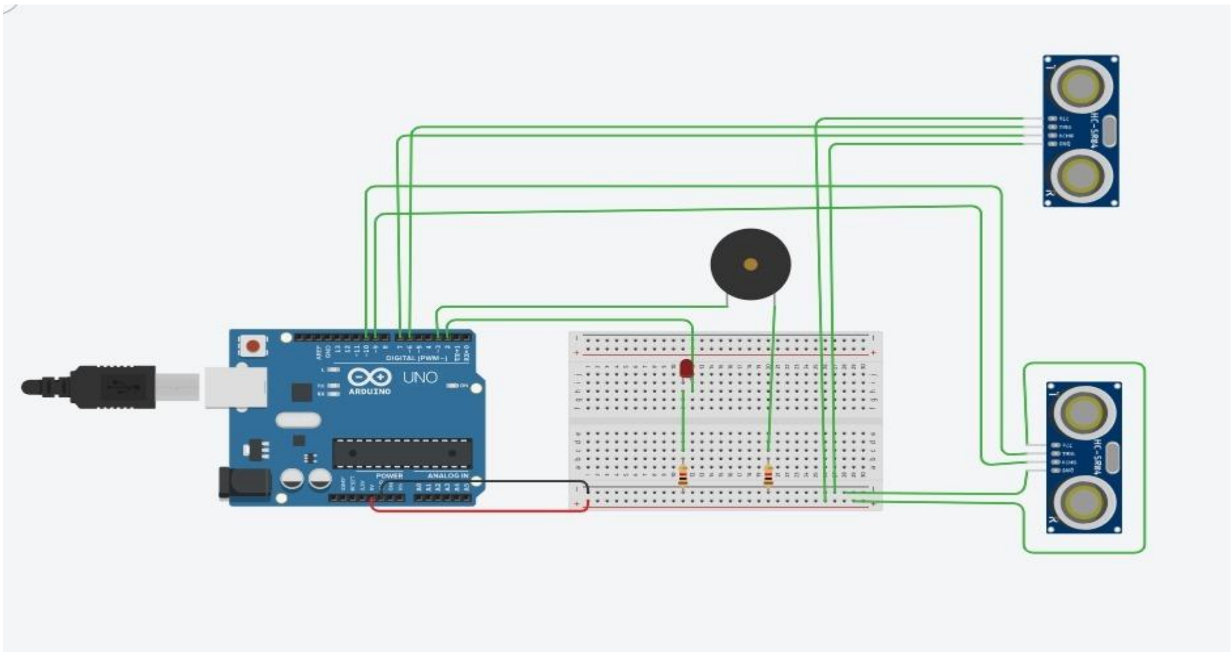
In order to generate the ultrasound, you need to set the Trig on a High state for 10us. It sends out an 8 cycle sonic burst and it will return back to Echo pin after reflecting from object/obstacles. The Echo pin output the time in microseconds the sound wave traveled which is further converted into distance the object is located from the sensor.



Arduino Uno

Ultrasonic Sensor

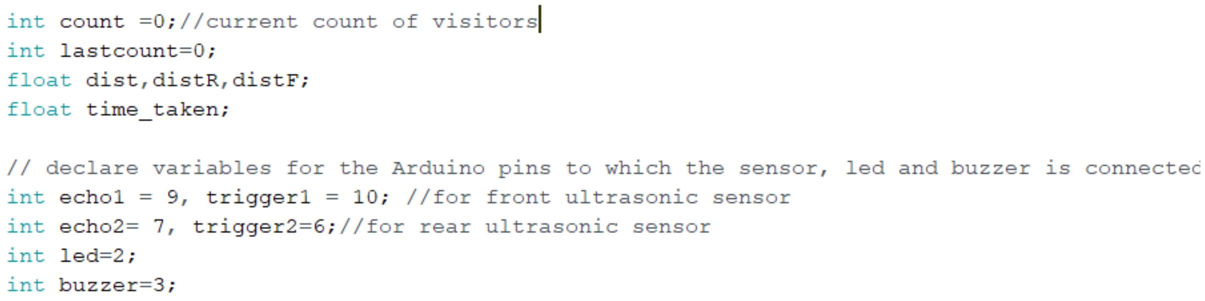
Circuit Diagram:



Working:

A counter variable initialized to 0 stores the current number of visitors present inside the shop. When a person enters the front ultrasonic sensor senses the object first followed by the rear sensor and the counter is increased by 1 whereas when a person exits , the rear ultrasonic sensor senses the object first followed by the front sensor and the counter is decreased by 1. The value of count along with a greeting message is displayed on the Serial Monitor. Whenever count>0 , the LED glows automatically. Whenever count>5 (any threshold value may be taken), the buzzer is turned on automatically.

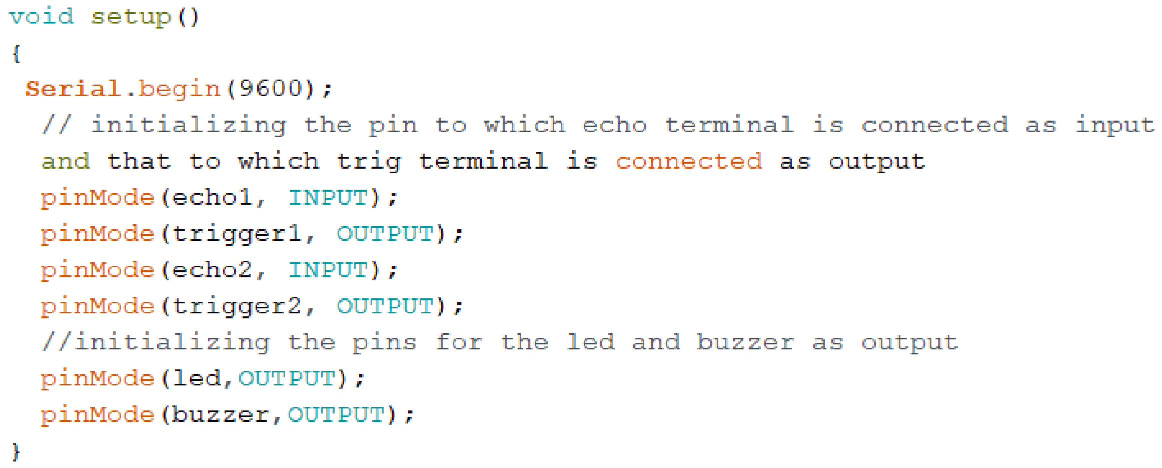
Code:



```
int count =0;//current count of visitors
int lastcount=0;
float dist,distR,distF;
float time_taken;

// declare variables for the Arduino pins to which the sensor, led and buzzer is connectec
int echo1 = 9, trigger1 = 10; //for front ultrasonic sensor
int echo2= 7, trigger2=6;//for rear ultrasonic sensor
int led=2;
int buzzer=3;
```

Here all the Arduino pins that are used are declared with appropriate variable names.

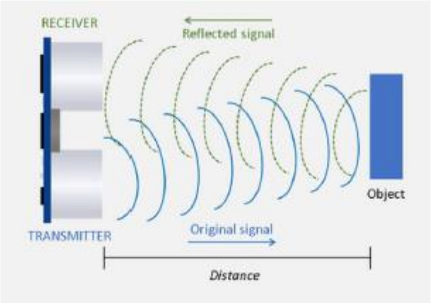


```
void setup()
{
 Serial.begin(9600);
  // initializing the pin to which echo terminal is connected as input
  and that to which trig terminal is connected as output
  pinMode(echo1, INPUT);
  pinMode(trigger1, OUTPUT);
  pinMode(echo2, INPUT);
  pinMode(trigger2, OUTPUT);
  //initializing the pins for the led and buzzer as output
  pinMode(led,OUTPUT);
  pinMode(buzzer,OUTPUT);
}
```

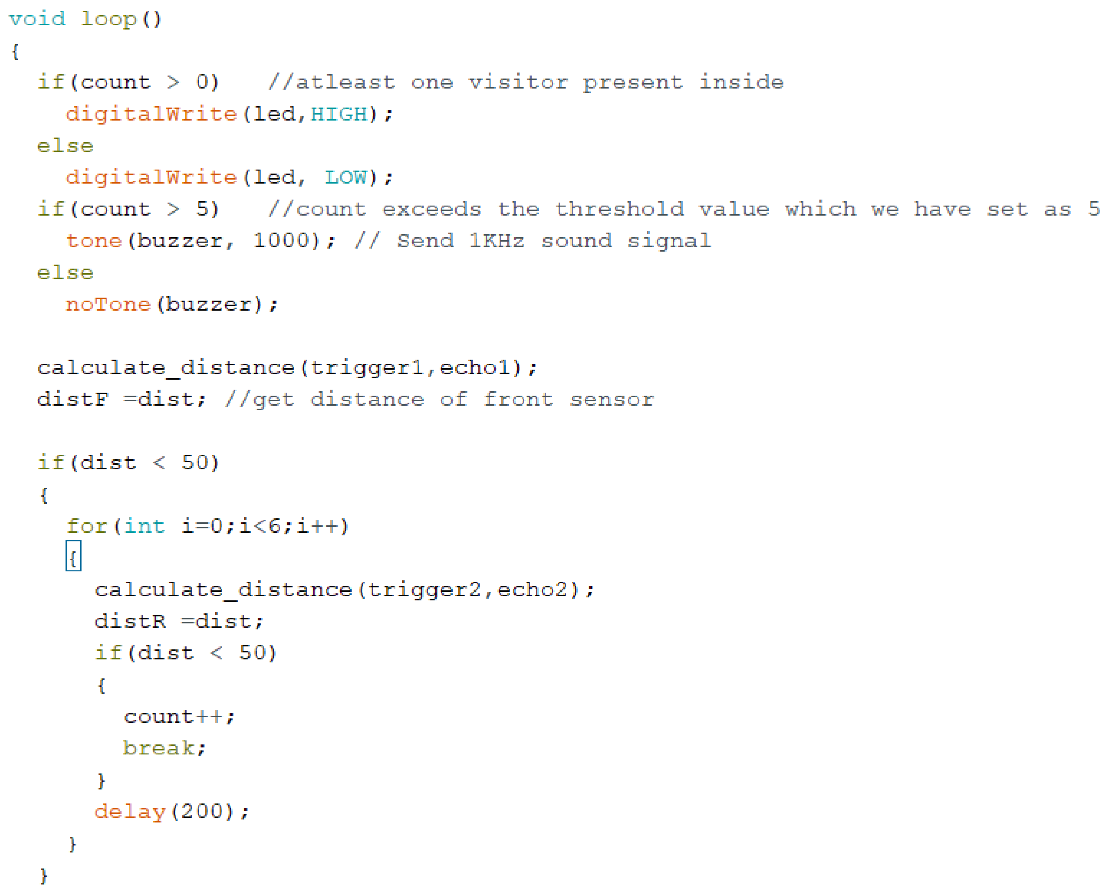
All the pins are initialized as either input or output.

```
//function to calculate the distance of an object placed in front of the ultrasonic sensor
void calculate_distance(int trigger, int echo)
{
  digitalWrite(trigger, LOW);
  delayMicroseconds(10);
  digitalWrite(trigger, HIGH);
  delayMicroseconds(10);
  digitalWrite(trigger, LOW);

  time_taken = pulseIn(echo, HIGH);
  dist= time_taken*0.034/2;
  if (dist>50)
    dist = 50;
}
```



This function is used to calculate the distance of an object placed in front of the ultrasonic sensor. Here distance is calculated as time*speed/2 because the total distance travelled by the ultrasonic beam is twice the original distance between the sensor and the object. If no object detected or object is far away, distance is assigned 50 cm.



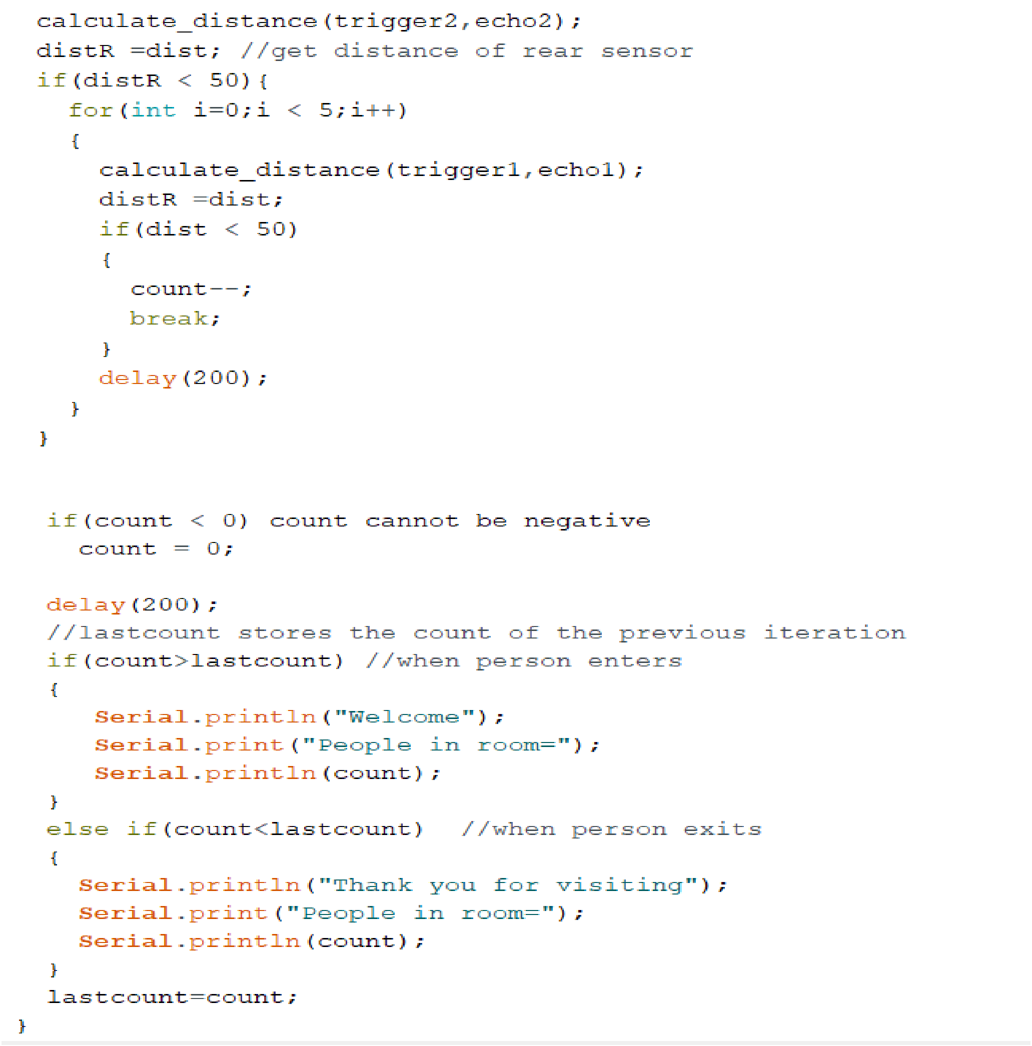
```
void loop()
{
  if(count > 0)   //atleast one visitor present inside
    digitalWrite(led,HIGH);
  else
    digitalWrite(led, LOW);
  if(count > 5)   //count exceeds the threshold value which we have set as 5
    tone(buzzer, 1000); // Send 1KHz sound signal
  else
    noTone(buzzer);

  calculate_distance(trigger1,echo1);
  distF =dist; //get distance of front sensor

  if(dist < 50)
  {
    for(int i=0;i<6;i++)
    {
      calculate_distance(trigger2,echo2);
      distR =dist;
      if(dist < 50)
      {
        count++;
        break;
      }
      delay(200);
    }
  }
```

When visitor count is greater than zero, LED is turned on and when the count exceeds the threshold value buzzer is turned on.

Distance of object from the front sensor is calculated and if it is less than 50 cm (object detected), then we wait for some time and keep on calculating the distance from rear sensor. If within this time the object is detected in front of rear sensor (distance<50 cm) then count is increased by 1.



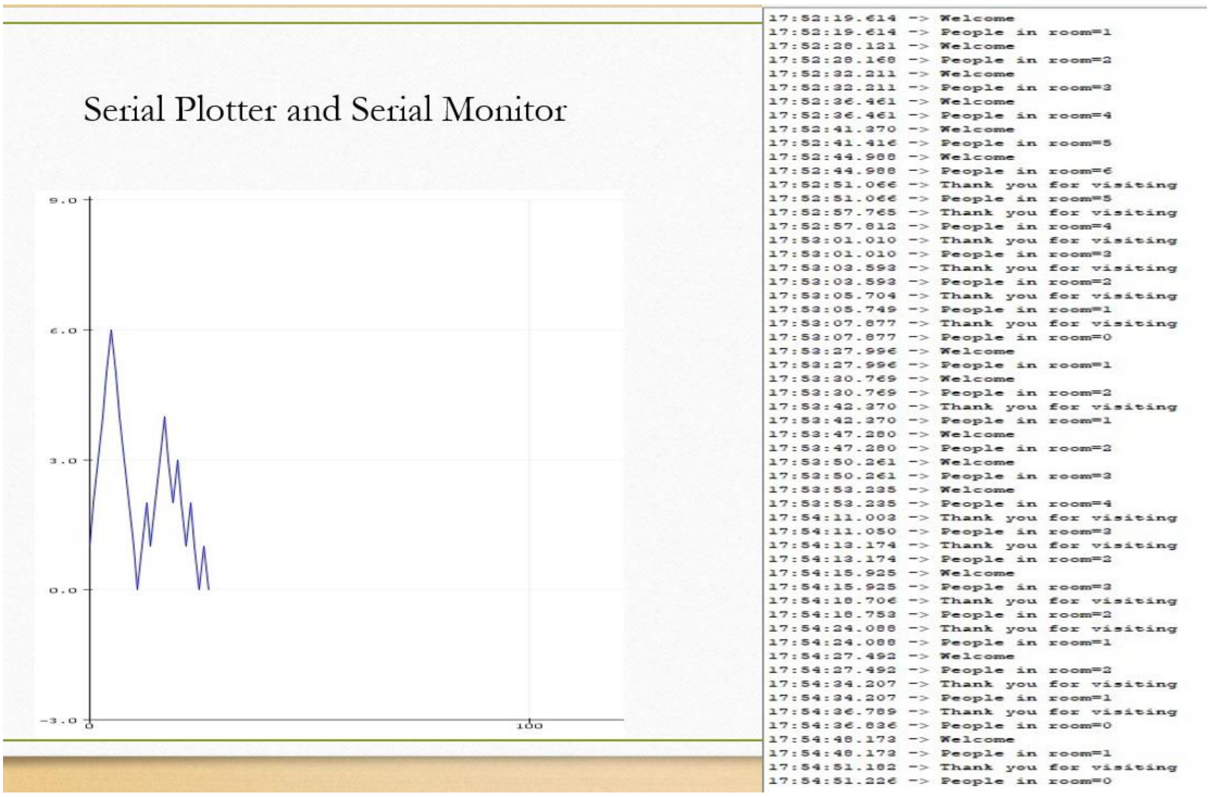
```
  calculate_distance(trigger2,echo2);
  distR =dist; //get distance of rear sensor
  if(distR < 50){
    for(int i=0;i < 5;i++)
    {
      calculate_distance(trigger1,echo1);
      distR =dist;
      if(dist < 50)
      {
        count--;
        break;
      }
      delay(200);
    }
  }


   if(count < 0) count cannot be negative
     count = 0;

   delay(200);
   //lastcount stores the count of the previous iteration
   if(count>lastcount) //when person enters
   {
      Serial.println("Welcome");
      Serial.print("People in room=");
      Serial.println(count);
   }
   else if(count<lastcount)  //when person exits
   {
     Serial.println("Thank you for visiting");
     Serial.print("People in room=");
     Serial.println(count);
   }
   lastcount=count;
 }
```

Now distance of object from the rear sensor is calculated and if it is less than 50 cm (object detected), then we wait for some time and keep on calculating the distance from front sensor. If within this time the object is detected in front of front sensor (distance<50 cm) then count is decreased by 1.

Finally if the count changes (increased or decreased by 1), we print the count along with a greeting message.



Serial Plotter and Serial Monitor

```
17:52:19.614 -> Welcome
17:52:19.614 -> People in room=1
17:52:28.121 -> Welcome
17:52:28.168 -> People in room=2
17:52:32.211 -> Welcome
17:52:32.211 -> People in room=3
17:52:36.461 -> Welcome
17:52:36.461 -> People in room=4
17:52:41.370 -> Welcome
17:52:41.416 -> People in room=5
17:52:44.988 -> Welcome
17:52:44.988 -> People in room=6
17:52:51.066 -> Thank you for visiting
17:52:51.066 -> People in room=5
17:52:57.765 -> Thank you for visiting
17:52:57.812 -> People in room=4
17:53:01.010 -> Thank you for visiting
17:53:01.010 -> People in room=3
17:53:03.593 -> Thank you for visiting
17:53:03.593 -> People in room=2
17:53:05.704 -> Thank you for visiting
17:53:05.749 -> People in room=1
17:53:07.877 -> Thank you for visiting
17:53:07.877 -> People in room=0
17:53:27.996 -> Welcome
17:53:27.996 -> People in room=1
17:53:30.769 -> Welcome
17:53:30.769 -> People in room=2
17:53:42.370 -> Thank you for visiting
17:53:42.370 -> People in room=1
17:53:47.280 -> Welcome
17:53:47.280 -> People in room=2
17:53:50.261 -> Welcome
17:53:50.261 -> People in room=3
17:53:53.235 -> Welcome
17:53:53.235 -> People in room=4
17:54:11.003 -> Thank you for visiting
17:54:11.050 -> People in room=3
17:54:13.174 -> Thank you for visiting
17:54:13.174 -> People in room=2
17:54:15.925 -> Welcome
17:54:15.925 -> People in room=3
17:54:18.706 -> Thank you for visiting
17:54:18.753 -> People in room=2
17:54:24.088 -> Thank you for visiting
17:54:24.088 -> People in room=1
17:54:27.492 -> Welcome
17:54:27.492 -> People in room=2
17:54:34.207 -> Thank you for visiting
17:54:34.207 -> People in room=1
17:54:36.789 -> Thank you for visiting
17:54:36.836 -> People in room=0
17:54:48.173 -> Welcome
17:54:48.173 -> People in room=1
17:54:51.182 -> Thank you for visiting
17:54:51.226 -> People in room=0
```

Conclusion:

This Arduino based project can have various real life applications such as:

In commercial sector: It can be used in malls and other retail shop to continuously monitor the current number of visitors and find out the number of visitors as a function of time (using Serial Plotter) to know when is the maximum customer count and manage service accordingly.

In medical Industry: Use in hospital to regulate the appointment of people waiting outside the doctor’s cabin. It can be used to restrict the overcrowding especially during covid times.

In concert and hall: Crowd of the people can be controlled at party or function (buzzer may be used which automatically turns on when there is overcrowding).

In home automation: Fan or bulb can be switched on or off automatically according to the count of the visitors. When there is no one inside, all appliances may be automatically switched off to save electricity.