Project Title: Traffic Management System

BY:BHARANIDHARAN.K

Project Definition: **The project involves using IoT devices and data analytics to monitor traffic flow and congestion in real-time, providing commuters with access to this information through a public platform or mobile apps. The objective is to help commuters make informed decisions about their routes and alleviate traffic congestion. This project includes defining objectives, designing the IoT traffic monitoring system, developing the traffic information platform, and integrating them using IoT technology and Python.**

DESCRIPTION OF PROJECT

● The aim is to reduce the manual interface to minimal.

● Smart traffic signals, AI to determine the flow of traffic, automated enforcement and communication to change the face of the traffic.

● **AI** coming in place, the signals would work according to the volume of **traffic** on each **road**.

IOT SYSTEM REQUIREMENTS

Traffic load is dependent on factors such as time, day, season, weather and unpredictable situations like accidents or construction activity or any special event.

❖The system can be divided into four main parts:

➢Hardware Model

➢Programming

➢Sensors

➢Arduino as PLC

The objective is to build a prototype that has the ability to collect information of the busy tracks by sensors and using a control unit to shift service to a given lane as per priority.

COMPONENTS OF PROJECT

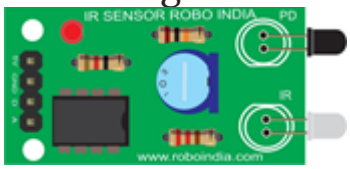
- **IR sensor:**

IR Sensor have three pins:

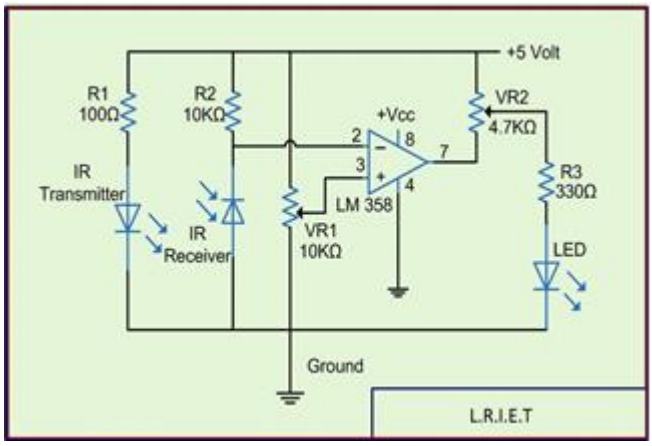
o VCC +5V

o GND

o D/A connects with any digital/Analog pin of Arduino when IR pair use as Digital Sensor.

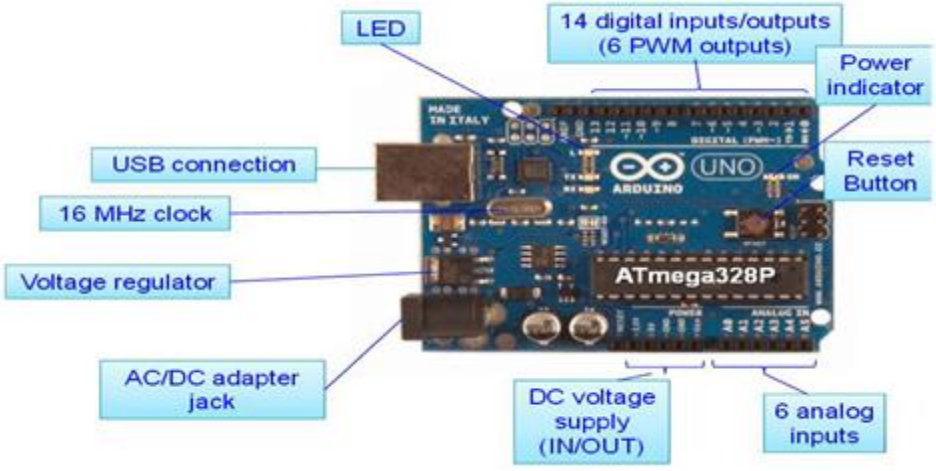


IR Sensor

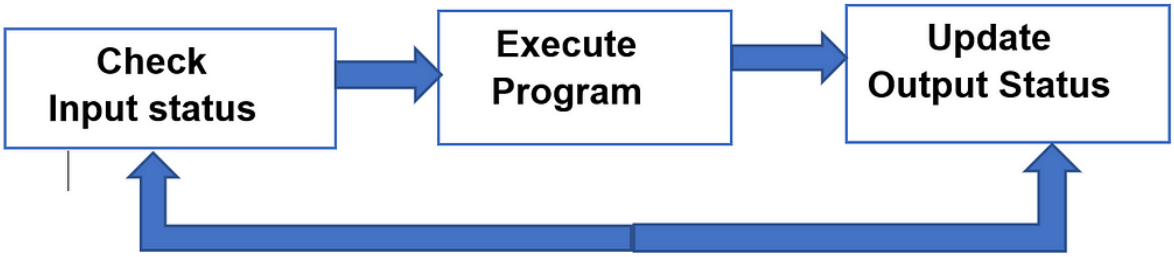


Circuit Diagram of IR Sensor

- **Arduino-UNO:** The Arduino UNO is an open-source micro-controller board based on the Microchip ATmega328P micro-controller and developed by Arduino.cc.The board is equipped with sets of digital and analog input/output (I/O) pins that may be interfaced to various expansion boards (shields) and other circuits. The board has 14 Digital pins, 6 Analog pins, and programmable with the Arduino IDE (Integrated Development Environment) via a type B USB cable.

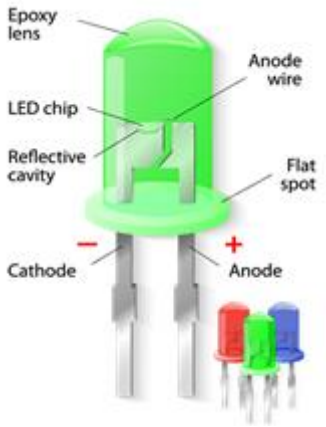


Arduino-UNO



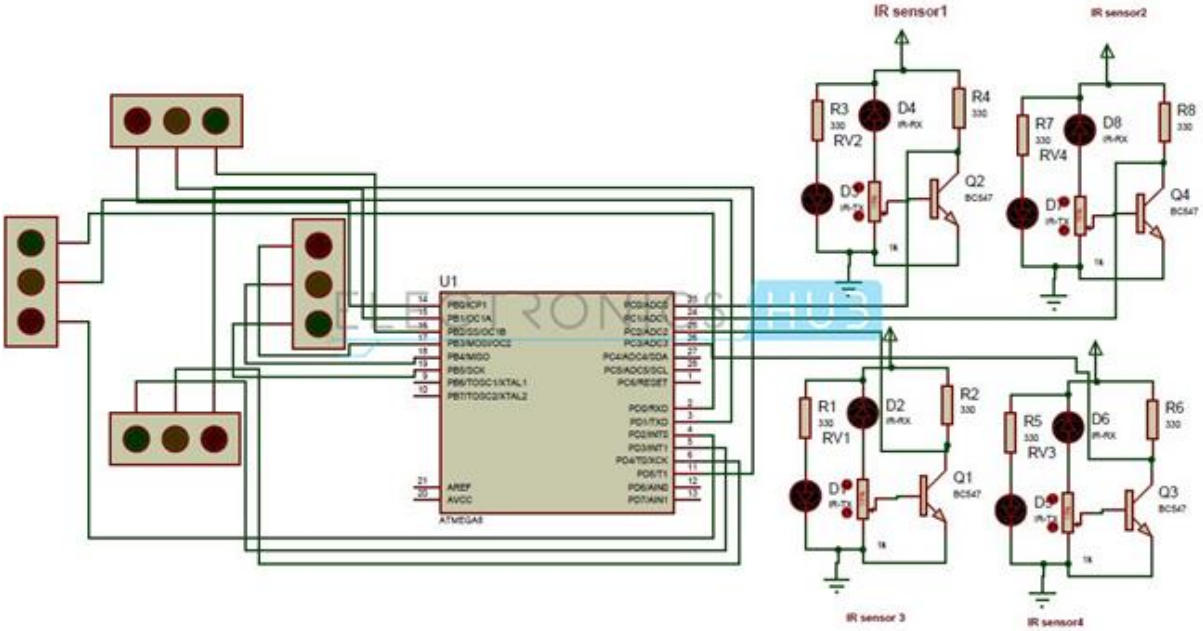
Arduino-UNO Working Process

- **LED s:**

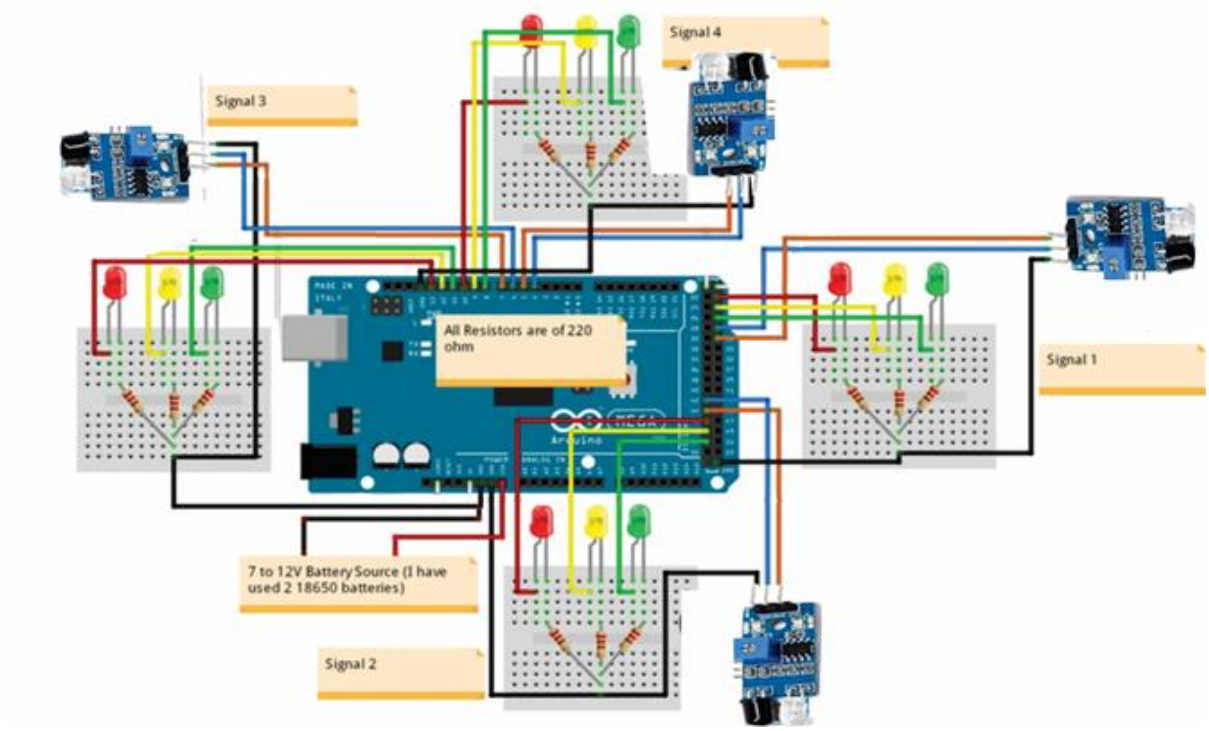


LED light diagram

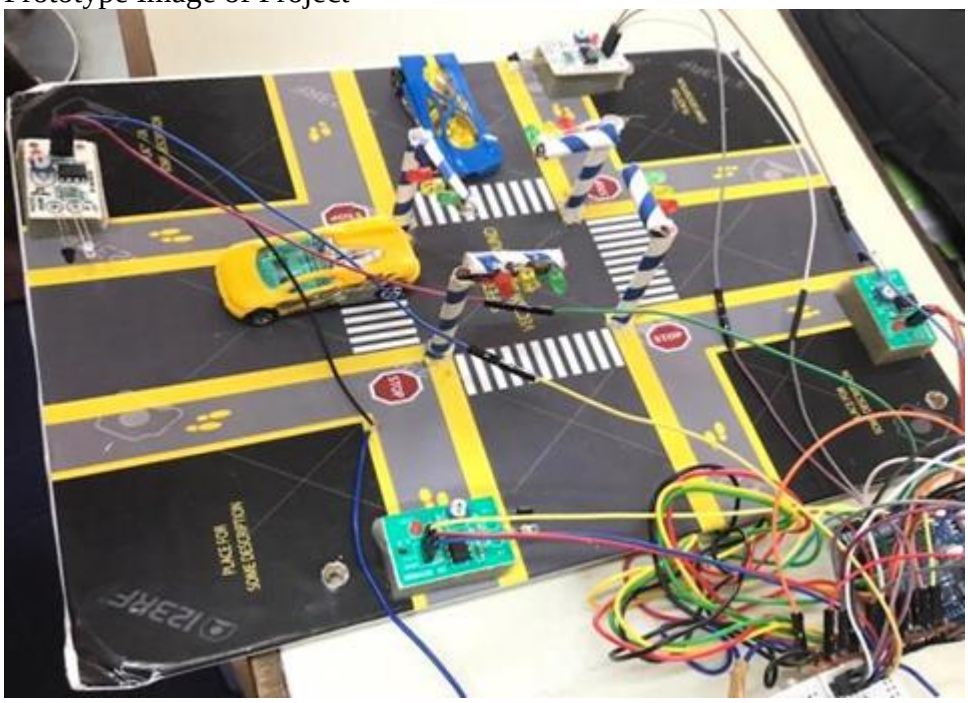
PROTOTYPE OF PROJECT



Circuit Diagram of Project



Prototype Image of Project


Working Model of Project

OPERATION OF MODEL

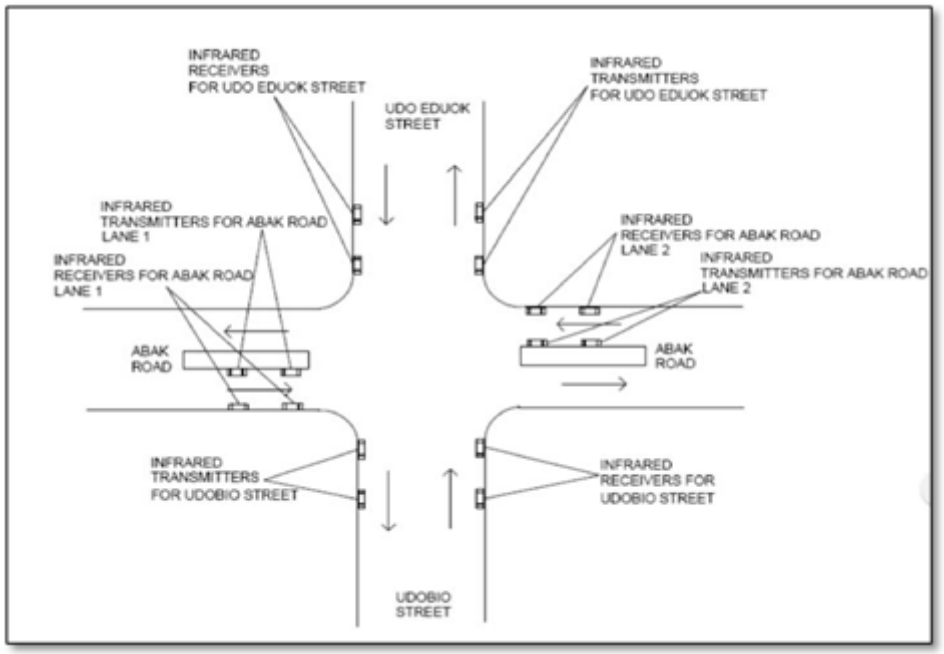
The model works on the principle of changing delay of Traffic signals based on the number of cars passing through an assigned section of the road. There are four sensors placed at four sides of a

four-way road which counts the number of cars passing by the area covered by the sensors.

Here we are using IR sensors replacing system to design a motion-based traffic control system. IR sensor contains IR transmitter IR receiver (photo-diode) in itself. These IR transmitter and IR receiver will be mounted on same sides of the road at a particular distance. As the vehicle passes through these IR sensors, the IR sensor will detect the vehicle & will send the information to the micro-controller.

The micro-controller will count the number of vehicles, and program allowing time to LED according to the density of vehicles. If the density is higher, LED will glow for higher time than average or vice versa.

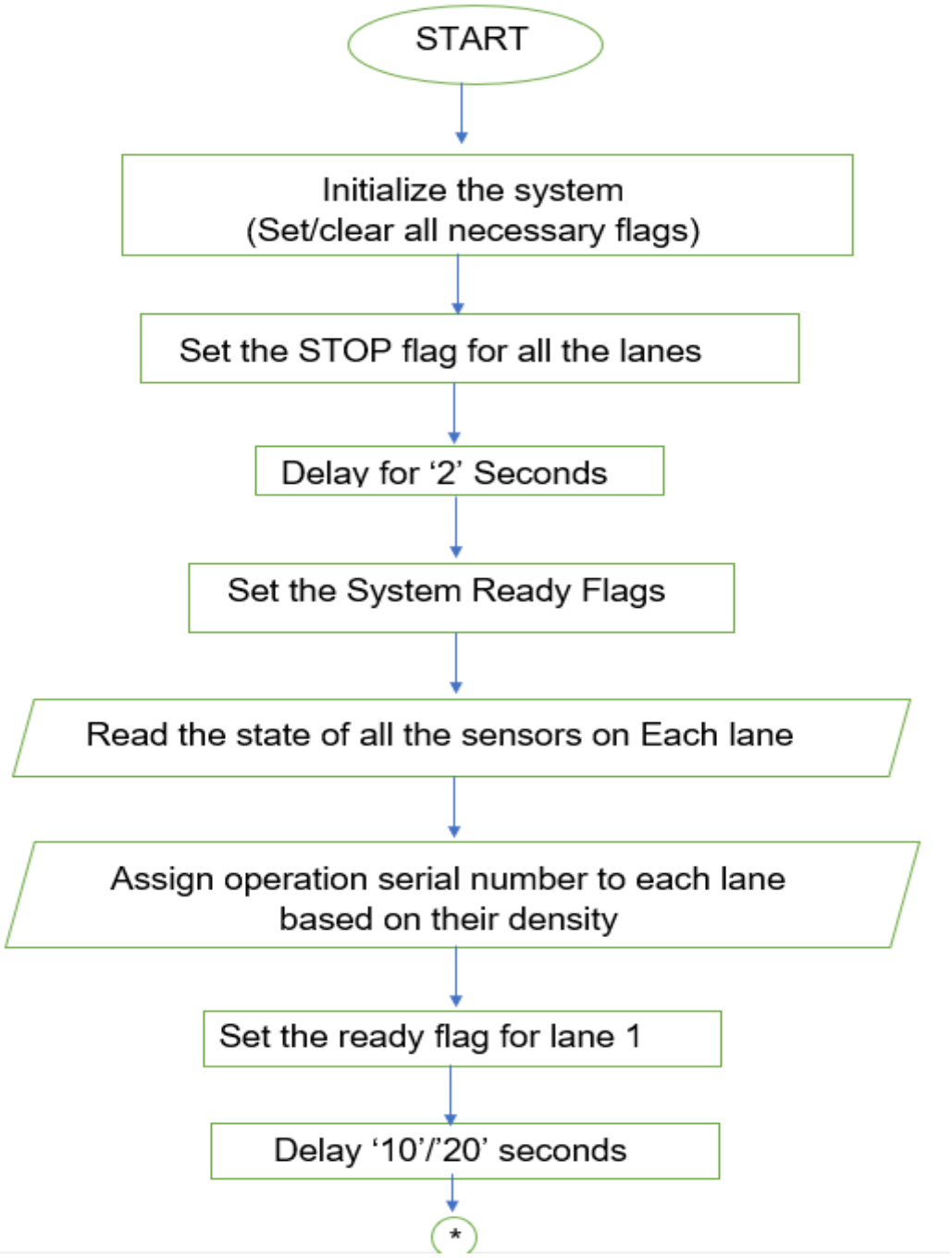
Infrared Sensors Arrangement and Implementation



Real implementation prototype

Since the design is focused on sensing the traffic level on each of the lanes of the road depending on the density of each lane using infrared sensors, the arrangement of the infrared sensor on the road layout was positioned to perform this function.

FLOWCHART OF THE SYSTEM



Flow Chart of the System

MODE OF OPERATION

Once the traffic control commences operation, the states of all the sensor arrays on each lane of traffic is read and given as input to the micro-controller for logical operations. The system assigns serial number to each lane based on their density, where the lane with the most density is assigned lane one. Accordingly, the system sets the ready flag for lane one where the YELLOW light shows; in preparation for the passing of traffic in that lane and delays for a certain time before giving the go signal with the decided time.

Going further, to implement the primary loop of the flow chart the system increments the counter where the amount of time allotted for lane one is accounted for; and takes a decision if the go time for lane one has elapsed. If it has not, the system takes a decision on the density of traffic on lane one by checking if the density flag has been cleared; If no, the system increments the counter again. After the density flag has been cleared and time has elapsed, the system makes a decision further to verify that the density flag has been cleared, if not, additional time is added and the primary loop is performed again. If the density flag for lane one is cleared, it makes a decision by checking to see if either lane one or any of the other three lanes are active, if not, the program stops. If lane one is the active lane, the ready flag for that lane is set again and the YELLOW light shows in preparation to stop traffic on that lane while simultaneously setting the ready flag for lane two. Once lane one has been stopped, the YELLOW light at lane two shows indicating readiness and eventually the lane is passed and the primary loop of the system could be performed RED light. again, this process is

repeated for all the other lanes depending on which one is the active lane.

Project Coding:

Project Code 1: This is the ideal code for the project but as Arduino-UNO does't support multiprocessing this code will not for our working model, but if we perform the same task with other micro-controller which helps multi-Processing this code can perform the task efficiently.


```
int obstacle1= 2;
int  r[]={13,4,7,10};
int g[]={3,6,9,12};
int y[]={5,8,11}
int arr[]={9,9,9,9};
int t1,t2;
int isobstacle=LOW;
int counter=0;
int greenTime=0;
int green=9; //variable green
int nextRED=1; //variable for next red
int currentRED=0; //current red variable
int currentGREEN=0; //current green variable
int waitTime=9;

void setup() {

  for(int i=0;i<4;i++)
  {
  pinMode (r[i], OUTPUT);
  }

  for(int i=0;i<4;i++)
  {
  pinMode (g[i], OUTPUT);
  }
```

I/O variable Declaration

File Edit Sketch Tools Help

MT_TRF_IDEAL_implementaion §

```
pinMode(obstacle1,INPUT);

digitalWrite(g[1],HIGH);
digitalWrite(r[2],HIGH);
digitalWrite(r[3],HIGH);
digitalWrite(r[4],HIGH);
}


void loop()
 {  if(waitTime>0)
      {
fun(waitTime);   // count next green Time
waitTime=0;
}


      else
      {
        waitTime=green;   // pass calculated waiting time for GREEN signal
      }

}

void fun(int wait_time)
{
 green=9;
 isobstacle=digitalRead(obstacle1);
   if(isobstacle==HIGH) //if obstacle is high
```

Done Saving.

I/O Port declaration

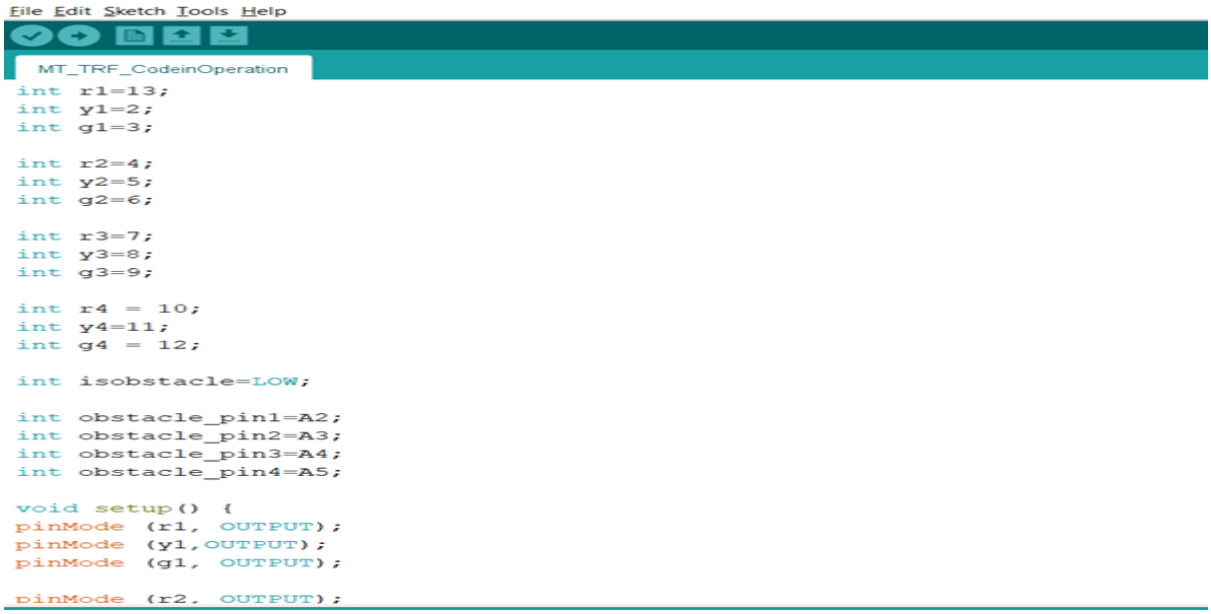
MT_TRF_IDEAL_implementaion §

```
      counter=counter+1; //counting the counter
  }

  Serial.println(counter);//printing the counter

  greenTime=(counter*10)/27; //greenTime calculation

green=green+greenTime; //originall green time + calcaulated green time

  Serial.print("\t Next signal will be Green for:"); //Just message priting
  Serial.print(green);
  for(int i=0;i<4;i++)
  {
if( i!=nextRED )    // increase RED timer (exclude next red signal)
{
arr[i]=green;
       }
  }
t1=currentRED;
t2=currentGREEN;


  if(currentRED+1 < 4)    // move to next RED and GREEN
    {
currentRED++; currentGREEN++;
    }
 else
    {
currentRED=0; currentGREEN=0;
    }
```

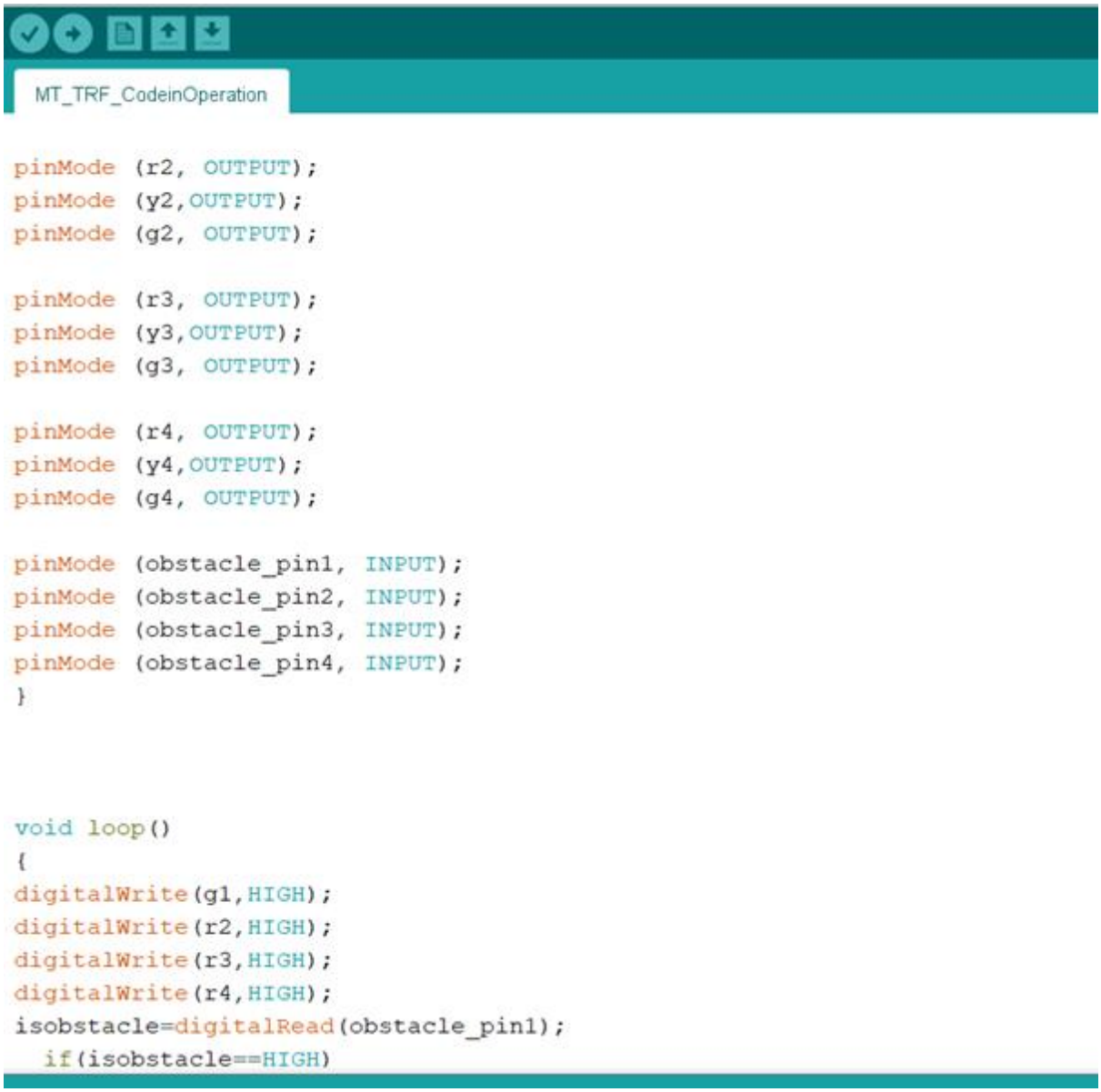
Working Procedure

Working Procedure

Project code 2 (Code in Operation): This is the code part which is implemented in the Working project and working fine with the conditions in which it don't need to implement multi-processing.



I/O Variable Declaration



Setting ports for I/O

File Edit Sketch Tools Help

MT_TRF_CodeinOperation

```
isobstacle=digitalRead(obstacle_pin1);
  if(isobstacle==HIGH)
  {
  Serial.println("1");
  delay(20000);

  }
  else
  {
    Serial.println("0");
  delay(10000);
  }

digitalWrite(y1,HIGH);
digitalWrite(g1,LOW);
delay(2000);
digitalWrite(y1,LOW);
digitalWrite(r1,HIGH);
digitalWrite(g2,HIGH);
digitalWrite(r2,LOW);

isobstacle=digitalRead(obstacle_pin2);
  if(isobstacle==HIGH)
  {
    Serial.println("1");
  delay(20000);

  }
  else
```

Working Procedure

File Edit Sketch Tools Help

MT_TRF_CodeinOperation

```
//delay(10000);
digitalWrite(y2,HIGH);
digitalWrite(g2,LOW);
delay(2000);
digitalWrite(y2,LOW);
digitalWrite(r2,HIGH);
digitalWrite(g3,HIGH);
digitalWrite(r3,LOW);

isobstacle=digitalRead(obstacle_pin3);
  if(isobstacle==HIGH)
  {
    Serial.println("1");
  delay(20000);

  }
  else
  {
    Serial.println("0");
  delay(10000);
  }

//delay(10000);
digitalWrite(y3,HIGH);
digitalWrite(g3,LOW);
delay(2000);
digitalWrite(y3,LOW);
digitalWrite(r3,HIGH);
digitalWrite(g4,HIGH);
```

Working Procedure

```
digitalWrite(r4,LOW);
 isobstacle=digitalRead(obstacle_pin4);
  if(isobstacle==HIGH)
  {
  Serial.println("1");
  delay(20000);

  }
  else
  {
    Serial.println("0");
  delay(10000);
  }
//delay(10000);
digitalWrite(y4,HIGH);
digitalWrite(g4,LOW);
delay(2000);
digitalWrite(y4,LOW);
digitalWrite(r4,HIGH);
digitalWrite(r1,LOW);
}
```

Done Saving.

Working Procedure

RESULT:

The program was burned and then the model was given power supply. By setting the IR sensor LOW initially density of all lanes were kept low. As a result, the traffic system operates in a sequential order servicing one lane after the other. The status of the LEDS give the status of any given lane.

Next when we increased the density of lane 3 via IR Sensor G1 or the green light in lane 1 gets on and that lane gets service irrespective of its sequence. After a fixed time, interval however the control shifts to the next lane in sequence. As a result, all lanes get service but the lane with higher density gets higher green time.

FUTURE WORK

● We will implement this system for traffic controlling in a 4-lane junction.

● We will update this system with when a pedestrian (a person walking rather than travelling in a vehicle.) try to cross the road during green signal it will turn on an alarm and warn the pedestrian and traffic police.

● We will update this system with when a vehicle tries to move even during red signal it will turn on an alarm to warn the driver of the vehicle and the traffic

CONCLUSION

The project may be very well used in where the traffic signals is kept and in many other places where we need to full fill the need of the automation. In this project we have studied the optimization of traffic light controller in a City using IR sensors and Arduino. By using this system configuration, we try to reduce the possibilities of traffic jams, caused by traffic lights, to an extent and we have successfully got the results.