Density Based Traffic Control System

Control Systems (ECE2010)
Slot F1+TF1

Final Project Report

Made By:

Sree Lalitha P.V.S : 19BEC0530

Shreyas Neeraj Khandekar :19BEC0541

Ankita Mandal: 19BEC0543

Submitted to Faculty:

Bagubali A

Contents

Abstract:

With the increase in population and traffic in cities around the country, road traffic has become a nuisance. It is responsible for both causing huge traffic jams and secondly, it is a huge waste of time.

The traditional traffic control system allocates a fixed amount of time to each lane of traffic, hence failing in time management. Therefore, we propose a density-based traffic control system using a microcontroller, which provides a variable amount of time, depending upon the density of the traffic on each lane of the road. This system aims to drastically reduce the waiting time for drivers, making driving on roads a pleasant experience. Such a system will also reduce the risk of traffic congestion which is of good effect to people on the roads in the condition of an emergency.

Introduction:

Nowadays traffic congestion has become a major problem faced by people living in cities. Getting stranded in heavy traffic is not only a huge headache, but in fact, it also burns a hole in one's pocket. As per research conducted by IIT Madras, traffic congestion in New Delhi alone cost the city $9 Billion in 2017, and this figure will increase to $15 Billion by 2030 if no changes are made.

As of today, there are two ways of controlling traffic in Indian cities. The first, manually controlled by a traffic policeman. Such a method is dated and not possible at a major junction. The second method is using the standard traffic lights that are commonly found. These traffic lights use a simple concept, giving an equal timeslot to each lane of the road. Roads being unpredictable makes such an implementation obsolete.

Another such implementation that can be considered is the use of a Deep Learning Algorithm to classify motor vehicles and traffic, but such a system has 2 shortcomings, firstly a very high chance of false positives or false negatives. Secondly, it's very data-heavy and difficult to maintain in a server.

A density-based traffic control system is proposed to replace the current system. The proposed system is a very simple implementation that will cause a massive upshift in traffic management in the cities. The total system contains a few IR sensors, which will send a signal to a microcontroller, which will, in turn, control the traffic lights. Depending upon the activation of the IR sensors, the traffic lights can be effectively managed, by allotting a variable amount of time to each lane of traffic.

Problem Statement:

An effective method to counter the effect of rise in vehicle count and growth in traffic congestion. Analyse and improve the traffic systems previously implemented using various sensors by exploring their features and implementation.

Literature Review

AUTHOR	TITLE	YEAR OF PUBLICATION	INFERENCE
T.E Somefun, C.O.A. Awosope, A. Abdulkareem, E. Okpon, A.S. Alayande, C.T. Somefun	Design and Implementation of Density-Based Traffic Management System	2020	This paper uses Arduino Mega, infrared sensors and Bluetooth module to save 60% time to clear traffic and an interaction.
Rajeshwari Sundar, Santhoshs Hebbar, and Varaprasad Golla	Implementing Intelligent Traffic Control System for Congestion Control, Ambulance Clearance, and Stolen Vehicle Detection	2015	This paper uses RFID tags placed on vehicles to provide smooth passage for emergency vehicles and detect stolen vehicles at traffic junctions.

Giulia Piacentini, Paola Goatin, Antonella Ferrara	Traffic control via moving bottleneck of coordinated vehicles	2017	This paper introduces the use of a moving bottleneck as the control variable to rescue fuel consumption and also minimise travel time.
Pramod Sharma, Akansha Mishra, Kaumudi Singh	Density based Traffic control system using IR sensors	2015	This paper uses IR sensors and LED lights as traffic lights whose glowing time is decided by the microcontroller based on the density of the traffic.
Amit Kumar Bhakta, C Eng Faruk Bin Poyen, Durga manhar, Imran Ali,ArghyaSantra,	Density Based Traffic Control	2016	This paper uses proximity sensor and is based on the principle of changing delay based on the density of the traffic
Y.N. Udoakah, I.G. Okure	Design and implementation of a Density-based Traffic light control with surveillance system	2018	This paper uses arrays of IR sensors interfaced with microcontrollers and strategically positioned AND gates that determine the density of the traffic

Amanjot Kaur,Dr. Mohita Garag,Harpreet Kaur	Review of Traffic Management Control Technique	2017	In this paper, various traffic management techniques have been studied. The survey of various traffic management schemes concludes that different techniques having own advantages and disadvantages
Dipak Gade	ICT based Smart Traffic Management System “iSMART” for Smart Cities	2019	This paper discusses the implementation of a iSMART POC for Traffic Management.

System model:

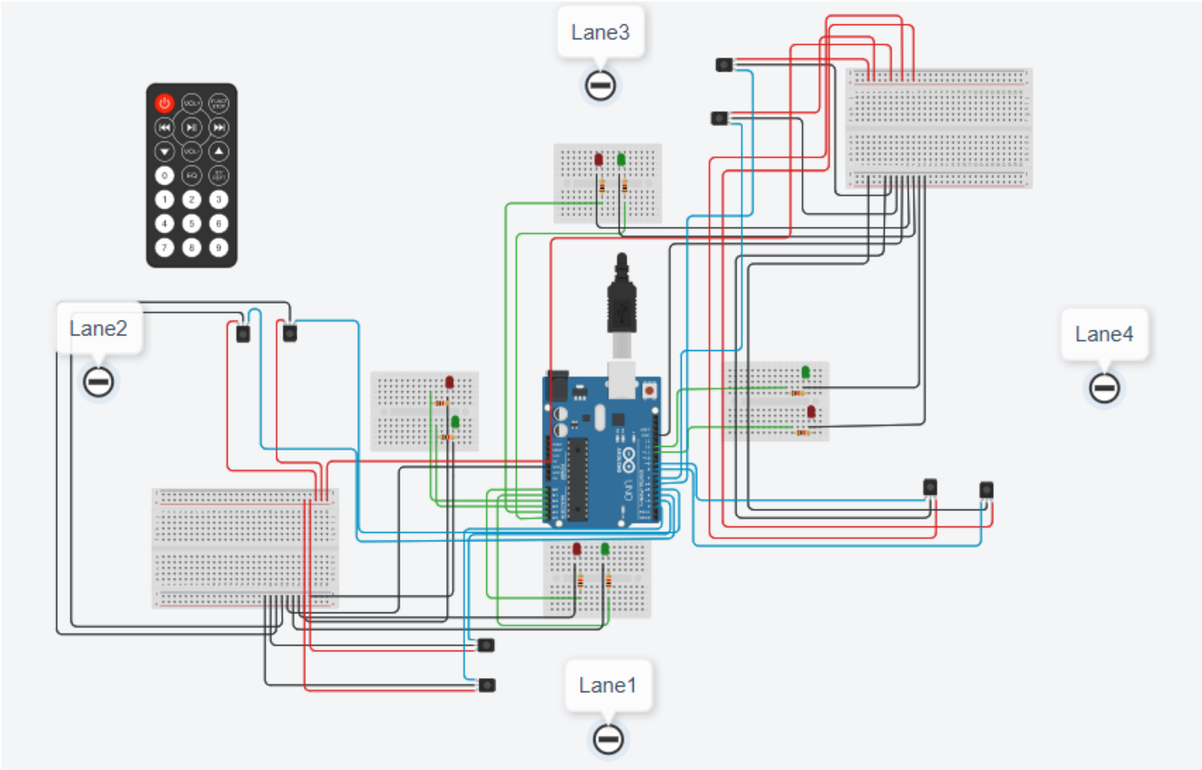

Fig. 1: Computerized representation of the circuit of the prototype

Components:

- Arduino UNO: An open-source microcontroller board based on the Microchip ATmega328P microcontroller. The board is equipped with sets of digital and analogue input/output (I/O) pins.
- BreadBoard: A solder less device for a temporary prototype with electronics and test circuit designs.
- IR sensor: IR sensor has 3 pins - VCC, PWR, GND. The connections on the project are made appropriately. The OUT pin is connected to the
digital slots of the microcontroller, The VCC and GND pins are connected to the breadboard, where the pins are connected to the output power and ground slots of the microcontroller board respectively.
- LED- Standard RED and Green LEDs are used to show traffic lights, the anode of the LEDs are connected to analog or digital pins of the microcontroller to operate appropriately to the signal, and the cathode is connected to the ground using a breadboard.

System connections:

- Lane 1-
    - Red Led - A0 pin
    - Green Led - A1 pin
    - IR sensor 1 - D2
    - IR sensor 2 - D3
- Lane 2-
    - Red Led- A2 pin
    - Green Led - A3 pin
    - IR sensor 1- D4
    - IR sensor 2- D5
- Lane 3- ○ Red Led - A4 pin
    - Green Led A5 pin
    - IR sensor 1 - D6
    - IR sensor 2- D7
- Lane 4- ○ Red Led - D11 pin
    - Green Led - D12 pin
    - IR Sensor 1- D8
    - IR sensor 2 - D9

Methodology:

The prototype has been programmed using Arduino IDE in C++ language. It works completely independently as it is pre-coded and doesn't require any additional inputs. We can change the amount of time it stays on by changing the values in the code. But apart from that, it does not take any other input. It will only require a power supply to power the microcontroller.

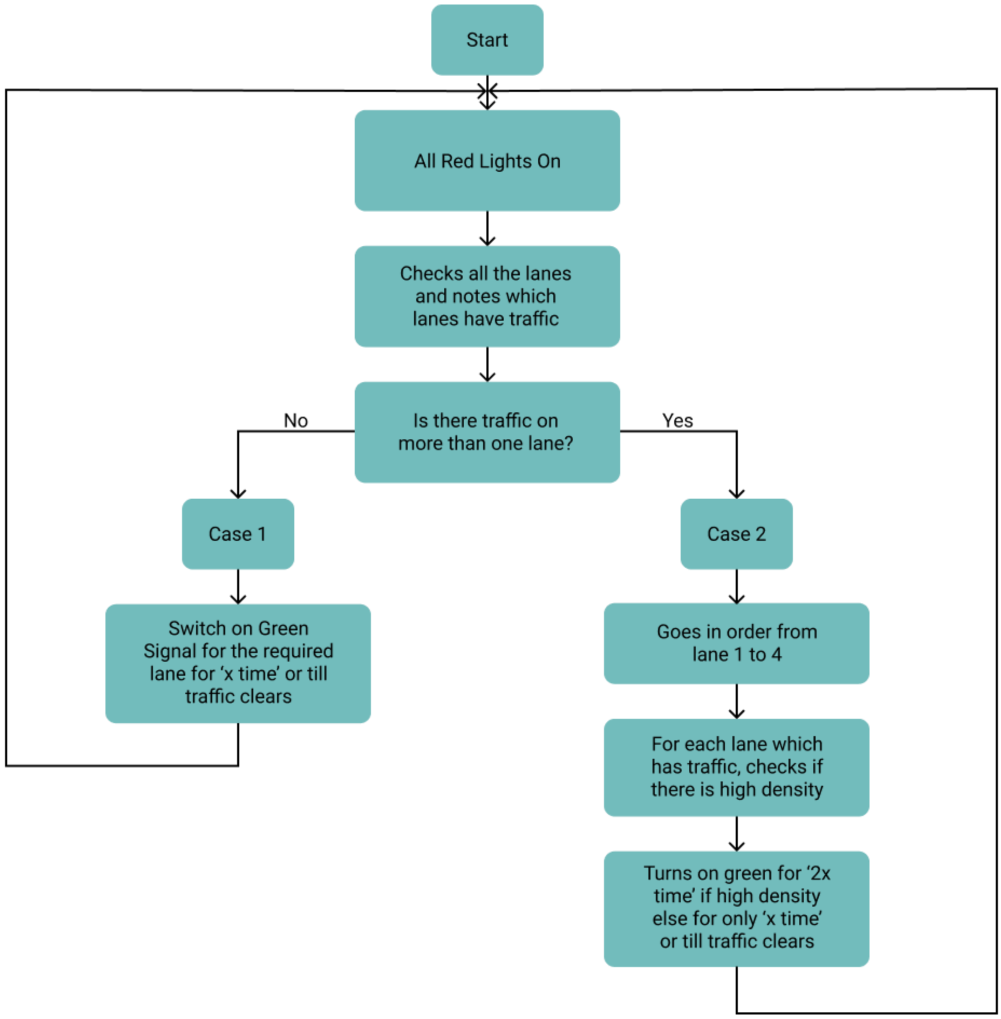

Fig. 2: Flow chart explaining the working

The basic working of the code has been explained in Fig. 2. To elaborate on that, once the system is initiated, the code turns on the red light for all 4 lanes. Once it enters the main loop, all the lanes from 1 to 4 are checked if they have traffic. If they do, then the lane number is noted. We also keep a counter variable which keeps a count of how many lanes have traffic. Now we use a switch case with the counter variable. If there is just one lane with traffic, we call the Case1 function. However, if there is more than one lane with traffic, we call the Case2 function.

In the Case1 function, we just switch on the green signal for the required lane. This green signal stays on for 'x time'. In this case, we have set the time to 15secs as this is just a prototype and we wanted to show how it works. For real life scenarios, we would need a much longer delay. There is also a provision in the system such that if the traffic in that particular lane clears out, the signal turns red again even if the given time condition hasn't been fulfilled. After switching to red, the system goes back to the main loop and begins the process from the start.

In the Case2 function, we cycle through from lane 1 to 4. For each lane that has traffic, we switch on the green signal and that lane. Then we check if that lane has a high density of traffic. If it does, then the time condition for that signal is changed from ‘x time’ to ‘2x time’. So in our prototype, we change it from 15secs to 30secs. This also has the provision wherein the signal turns back to red whenever the traffic is cleared before the time condition is satisfied. If there is light traffic in any lane, similar to Case1, it switches the signal to green for 15secs or till the traffic clears and then moves on to the next lane. After cycling through all the lanes the system again passes back to the main loop and starts checking for traffic all over again.

The important thing to note here is that it never switches on the green light for a lane that does not have traffic. This saves time in a lot of cases as is evident in our comparison in the next section.

Other Algorithms

One of the other algorithms [12]controls traffic lights using image processing techniques which can be implemented in real time. A web camera is used at each stage of the traffic light to record images of the traffic lane we have selected. These captured images are then compared to a reference image of the empty road by image matching process. Canny edge detection method is used to determine the boundaries of the images. Here, the traffic is controlled depending on the percentage of match between the images.

Another system uses a video camera to capture images of the road and make use of that data to control traffic signals. Instead of finding the total number of vehicles, they determine the traffic density corresponding to the total area occupied by vehicles on the road with respect to the total amount of pixels in one video frame.

Simulation Implementation:

In the below case we have simulated the code on tinker cad and for simulation purposes we have used an IR sensor remote to indicate the presence of car in a lane.
The number 1,2,3,4 on the IR remote represent lanes 1,2,3,4 respectively.

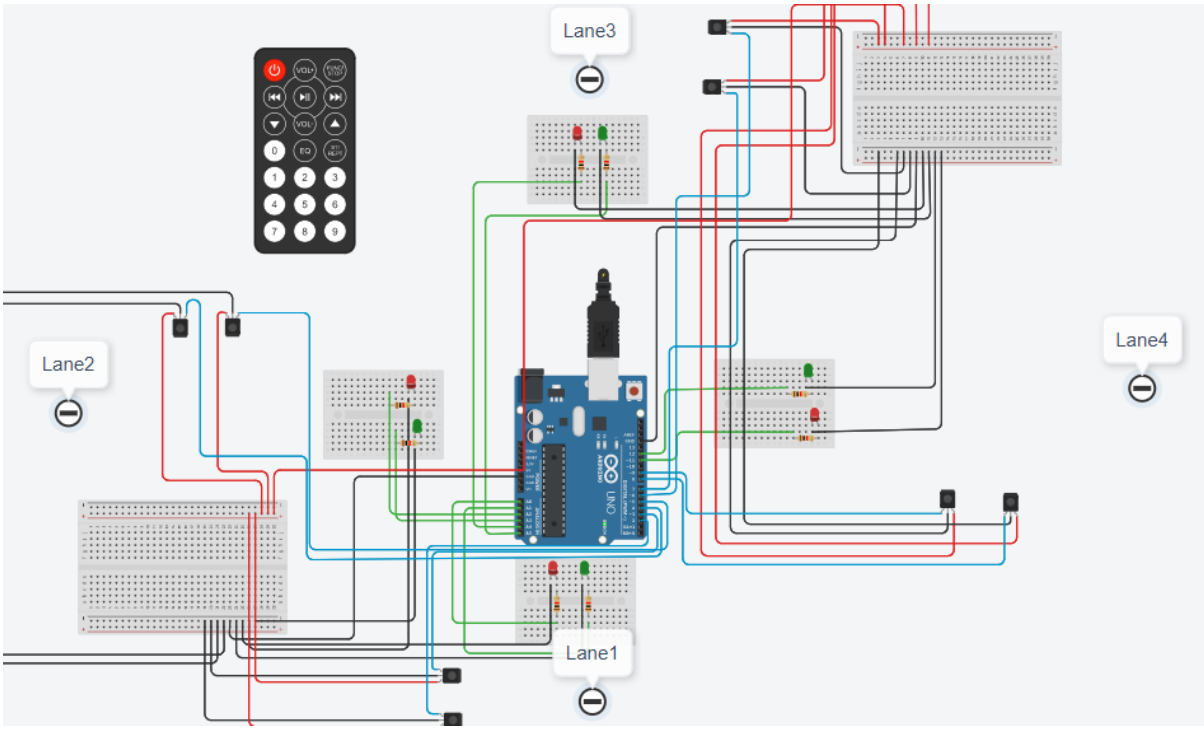

Fig. 3: Initial condition of the system

Here Initially all the lights are red for the initial condition as no lane has cars.[Fig.3]

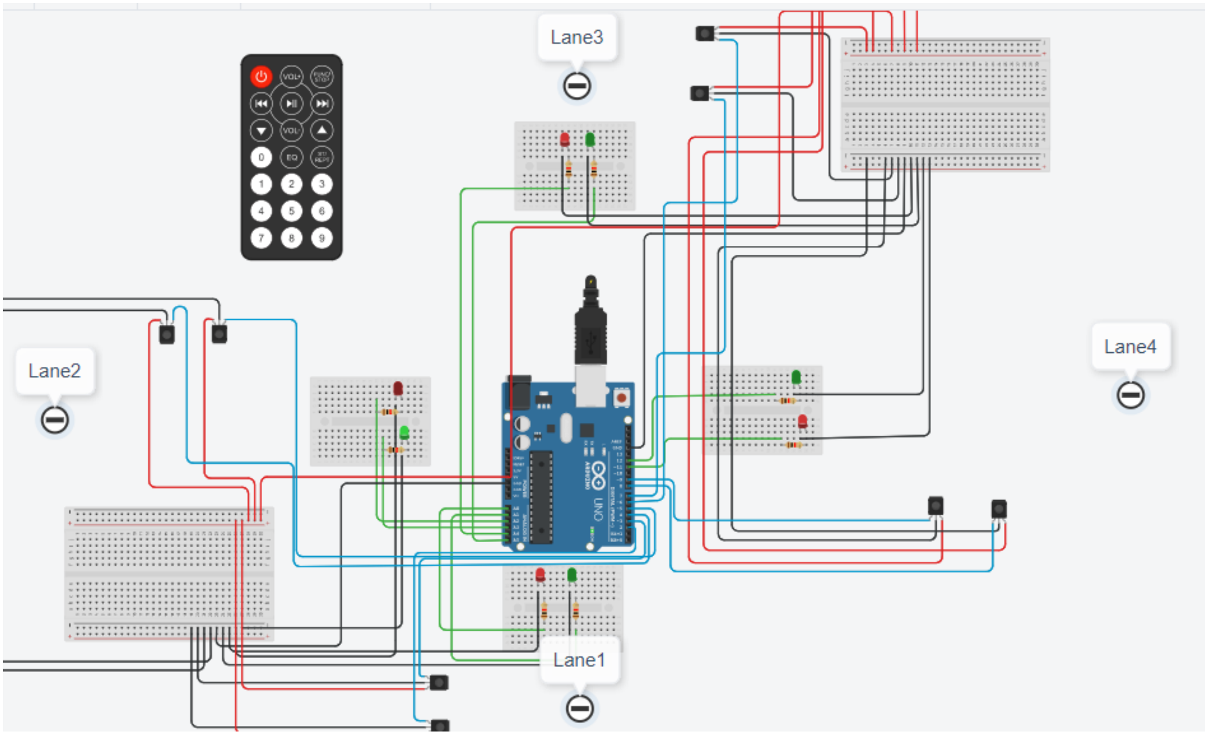

Fig. 4:Ccondition of the system when there is traffic on Lane 2

Here As we press button 2 on IR remote it indicates the presence of car in lane 2 so the Light for lane 2 becomes green.

Code:

```
#include <IRremote.h>

//declaring leds
const int s1r = A0;
const int s1g = A1;
const int s2r = A2;
const int s2g = A3;
const int s3r = A4;
const int s3g = A5;
const int s4r = 11;
const int s4g = 12;
//declaring ir sensors
const int s1ir1 = 2;
const int s1ir2 = 3;
const int s2ir1 = 4;
const int s2ir2 = 5;
const int s3ir1 = 6;
const int s3ir2 = 7;
const int s4ir1 = 8;
const int s4ir2 = 9;
//declaring other variables
int ctr;
int i;
```

```
int lanes[4]; //4 lane traffic signal
int RECV_PIN = 2;
IRrecv irrecv(RECV_PIN);
decode_results results;

void setup() {
 //setting up leds
 pinMode(s1r,OUTPUT);
 pinMode(s1g,OUTPUT);
 pinMode(s2r,OUTPUT);
 pinMode(s2g,OUTPUT);
 pinMode(s3r,OUTPUT);
 pinMode(s3g,OUTPUT);
 pinMode(s4r,OUTPUT);
 pinMode(s4g,OUTPUT);

 //setting up ir sensors
 pinMode(s1ir1,INPUT);
 pinMode(s1ir2,INPUT);
 pinMode(s2ir1,INPUT);
 pinMode(s2ir2,INPUT);
 pinMode(s3ir1,INPUT);
 pinMode(s3ir2,INPUT);
 pinMode(s4ir1,INPUT);
 pinMode(s4ir2,INPUT);
 //initial setup of all red on green off
 analogWrite(s1r,255);
 analogWrite(s2r,255);
 analogWrite(s3r,255);
 digitalWrite(s4r,HIGH);
 analogWrite(s1g,0);
 analogWrite(s2g,0);
 analogWrite(s3g,0);
 digitalWrite(s4g,LOW);

 Serial.begin(9600);
 irrecv.enableIRIn();
 irrecv.resume(); // Receive the next value
}

//main loop
void loop() {
 ctr=0;
 lanes[0]=0;
 lanes[1]=0;
 lanes[2]=0;
 lanes[3]=0;
 i=0;
 //to check which lanes are active
 if (irrecv.decode(&results)){
   Serial.println(results.value, HEX);

   switch(results.value){ //IR remote input to signify presence of car
     case 0xFD08F7:
```

```
      ctr++;
        lanes[0]=1;
      break;
    case 0xFD8877:
      ctr++;
        lanes[1]=1;
      break;
    case 0xFD48B7:
      ctr++;
        lanes[2]=1;
      break;
    case 0xFD28D7:
      ctr++;
        lanes[3]=1;
      break;
    case 0xFD58A7:  //button 9 for completing the input
      i=1;
   }
  irrecv.resume();

 //to activate appropriate function based on how many lanes are active
 switch(ctr)
 {
 case 1: Case1(); break;
 case 2: Case2(); break;
 case 3: Case2(); break;
 case 4: Case2(); break;
 default: analogWrite(s1r,255);
 analogWrite(s2r,255);
 analogWrite(s3r,255);
 digitalWrite(s4r,HIGH);
 analogWrite(s1g,0);
 analogWrite(s2g,0);
 analogWrite(s3g,0);
 digitalWrite(s4g,LOW);
 break;
  }
 }
}

//case for single lane active
void Case1() {
 delay(300); //common delay
 if(lanes[0]==1){
 analogWrite(s1r,0);
 analogWrite(s1g,255); // green for the car in lane 1
 delay(1000);
 lanes[0]=0;
 analogWrite(s1g,0);
 analogWrite(s1r,255); // after the car is gone, red light
 if(digitalRead(s1ir1)==HIGH){ // if lane is cleared before the delay is
over
 lanes[0]=0;
 analogWrite(s1g,0);
```

```
 analogWrite(s1r,255);
 }
 }
  if(lanes[1]==1){
 analogWrite(s2r,0);
 analogWrite(s2g,255); // green for the car in lane 2
 delay(1000);
 lanes[1]=0;
 analogWrite(s2g,0);
 analogWrite(s2r,255); // after the car is gone, red light
    if(digitalRead(s2ir1)==HIGH){ // if lane is cleared before the delay
is over
 lanes[1]=0;
 analogWrite(s2g,0);
 analogWrite(s2r,255);
 }
 }
  if(lanes[2]==1){
 analogWrite(s3r,0);
 analogWrite(s3g,255); // green for the car in lane 3
 delay(1000);
 lanes[2]=0;
 analogWrite(s3g,0);
 analogWrite(s3r,255); // after the car is gone, red light
    if(digitalRead(s3ir1)==HIGH){ // if lane is cleared before the delay
is over
 lanes[2]=0;
 analogWrite(s3g,0);
 analogWrite(s3r,255);
 }
 }
  if(lanes[3]==1){
 digitalWrite(s4r,0);
 digitalWrite(s4g,1); // green for the car in lane 4
 delay(1000);
 lanes[3]=0;
 digitalWrite(s4g,0);
 digitalWrite(s4r,1); // after the car is gone, red light
    if(digitalRead(s4ir1)==HIGH){ // if lane is cleared before the delay
is over
 lanes[3]=0;
 analogWrite(s4g,0);
 analogWrite(s4r,255);
 }
 }
 }

// for multiple lanes with cars
void Case2(){
  delay (300);

 if(lanes[0]>=1){
 analogWrite(s1r,0);
 analogWrite(s1g,255); // green for the car in lane 1
```

```
 delay(2000*2); //delay given based on density of that lane
 lanes[0]=0;
 analogWrite(s1g,0);
 analogWrite(s1r,255); // after the car is gone, red light
 if(digitalRead(s1ir1)==HIGH){ // if lane is cleared before the delay is
over
 lanes[0]=0;
 analogWrite(s1g,0);
 analogWrite(s1r,255);
 }
 }
  if(lanes[1]>=1){
 analogWrite(s2r,0);
 analogWrite(s2g,255); // green for the car in lane 2
 delay(2000*2);
 lanes[1]=0;
 analogWrite(s2g,0);
 analogWrite(s2r,255); // after the car is gone, red light
 if(digitalRead(s2ir1)==HIGH){ // if lane is cleared before the delay is
over
 lanes[1]=0;
 analogWrite(s2g,0);
 analogWrite(s2r,255);
 }
 }
  if(lanes[2]>=1){
 analogWrite(s3r,0);
 analogWrite(s3g,255); // green for the car in lane 3
 delay(2000*2);
 lanes[2]=0;
 analogWrite(s3g,0);
 analogWrite(s3r,255); // after the car is gone, red light
 if(digitalRead(s3ir1)==HIGH){ // if lane is cleared before the delay is
over
 lanes[2]=0;
 analogWrite(s3g,0);
 analogWrite(s3r,255);
 }
 }
  if(lanes[3]>=1){
 digitalWrite(s4r,0);
 digitalWrite(s4g,1); // green for the car in lane 4
 delay(2000*2);
 lanes[3]=0;
 digitalWrite(s4g,0);
 digitalWrite(s4r,1); // after the car is gone, red light
 if(digitalRead(s4ir1)==HIGH){ // if lane is cleared before the delay is
over
 lanes[3]=0;
 analogWrite(s4g,0);
 analogWrite(s4r,255);
 }
 }
}
```

Results and discussion:

Each lane gets **20 seconds** of green time In general Model. Taking a switching time of **2 seconds**.

On an average every car takes **4 seconds** to cross the intersection

Now considering 2 lanes:

No.of cars in **lane 1** =**5**

No.of cars in **lane 2** =**3**

Generic Model time:

For 2 lanes= 20*2+2*2 = **44 seconds**

For density based:

Time for lane 1= No.of cars* clear time = 5*4 = **20 seconds**

Time for lane 2 = No.of cars* clear time = 3*4 = **12 seconds**

Total time= Time for lane 1+ Time for lane 2+ No.of lanes*switching time

= 20+12+2*2

= 36 sec **Time saved = 44-36 = 8 seconds.**

As evidenced by the experiment, the density-based system is vastly better than the traditional time-based system we are using nowadays. On average, the proposed system was **more efficient** than the traditional time-based system.

Conclusion:

From the above theory and the reports of the experiment and analysis, we can easily conclude the effectiveness of a density-based traffic control system. Such a system can clear traffic with improved efficiency, causing much lesser no of congestions in traffic, saving time and money.

The equipment used and a wide implementation are relatively simple, and the changes can be easily made in already existing traffic signals without a lot of effort. The addition of a microcontroller to this system also helps in

future-proofing the system, making upgrades or expansion of a traffic signal much easier.
Furthermore, such a system is a lot more cost-effective for the city to maintain.

References:

[1] Somefun, T. E., Awosope, C. O. A., Abdulkareem, A., Okpon, E., Alayande, A. S., & Somefun, C. T. (2020). Design and implementation of density-based traffic management system. *International Journal of Engineering Research and Technology*, *13*, 2157-2164.

[2] Mohammed Fayaz, Pooja K, Pranitha P Reddy, Swathi T," Density Based Traffic Control System with Ambulance Detection" *International Journal of Engineering Research and Technology* (2019)

[3] Md. Shamsul Arifin, Sheikh Alimur Razi, Akramul Haque, Nur Mohammad. "A Microcontroller Based Intelligent Traffic Control System." *American Journal of Embedded Systems and Applications*. Vol. 7, No. 1, 2019, pp. 21-25.

[4] Soma Bhattacharyya, Somyadeep Ghosh, Abhra Deep Ghosh, Oindrila Ghosh, " Density Based Traffic Control System", *International Journal of Engineering Research and Technology:* (2019)

[5] Piacentini, G., Goatin, P., & Ferrara, A. (2018). Traffic control via moving bottleneck of coordinated vehicles. *IFAC-PapersOnLine*, *51*(9), 13-18.

[6] Udoakah, Y. N., & Okure, I. G. (2017). Design and implementation of a density-based traffic light control with surveillance system. *Nigerian Journal of Technology*, *36*(4), 1239-1248.

[7] Y. N. Udoakah, I.G. Okure, "Design and Implementation of Density Based Traffic Light Control with Surveillance System", *Nigerian Journal of Technology*(2017)

[8] Poyen, E. F. B., Bhakta, A. K., Manohar, B. D., Ali, I., ArghyaSantra, A., & Rao, A. (2016). Density based traffic control. *International Journal of Advanced Engineering, Management and Science*, *2*(8), 239611.

[9] B Durga Manohar, C Eng Faruk Poyer, Amit Kumar, Bhakta, Imran Ali, "Density Based Traffic control" *International Journal of Advanced Engineering, Management and Science* (2016)

[10] Siddharth Srivastav, Subhadeep Chakraborty, Raj Kamal, Rahil Monocha, "Adaptive Traffic Light Control". *Nerd IIT Kanpur* (2016)

[11] M Tubaishat, Y Shang, H Shi, "Adaptive Traffic Light Control with Wireless Networks", *4th IEEE Consumer Communications and Network Conference* (2007)

[12] Chandrasekhar. M, Saikrishna. C, Chakradhar. B, Phaneendra Kumar. P & Sasanka. C. “Traffic Control Using Digital Image Processing”, IJAEEE, ISSN (Print): 2278-8948, Vol-2, Issue-5, 2013

[13] Md. Munir Hasan, Aminul Hoque, Gobinda Saha, and Md. Badruddoja Majumder. “Smart Traffic Control System with Application of Image Processing Techniques”, 3rd International Conference on Informatics, Electronics & Vision 2014