Indian Institute of Technology (BHU), Varanasi

भारतीय प्रौद्योगिकी संस्थान काशी हिन्दू विश्वविद्यालय IIT INDIAN INSTITUTE OF TECHNOLOGY BANARAS HINDU UNIVERSITY

EXPLORATORY PROJECT

TOPIC-

DENSITY BASED TRAFFIC CONTROL USING ARDUINO & ULTRASONIC SENSORS

By

1.Abhishek Kumar, Roll no.-20095002

2.Ritesh Sahu, Roll no.-20095090

3.Dileep Gehlot, Roll no.-20095038

Under the Guidance of Professor

Dr Satyabrata Jit

Department of Electronics Engineering Indian Institute of Technology BHU Varanasi

Department of Electronics Engineering

Indian Institute of Technology (BHU)

CERTIFICATE

This is to certify that this project report **"To design Density based traffic control system using Arduino and Ultrasonic sensors"** is submitted by **Dileep Gahlot (20095038), Abhishek Kumar (20095002)** and **Ritesh Sahu (20095090)** who carried out the project work under the supervision of **Dr Satyabrata Jit.**

We approve this project for submission of the Exploratory Project, IIT(BHU) Varanasi.

Signature of Supervisor

Dr. Satyabrata Jit

Department of Electronics Engineering

IIT(BHU) Varanasi

ACKNOWLEDGEMENT

It is my great fortune that I have got opportunity to carry out this project work under the supervision of **Dr Satyabrata Jit** in the Department of Electronics Engineering, Institute of Information Technology VARANASI (BHU). I wish to convey my gratitude to Dr Satyabrata Jit for providing all kinds of infrastructural facility towards the project work.

STUDENT NAME:

Abhishek Kumar (20095002)

Ritesh Sahu (20095090)

Dileep Gehlot (20095038)

DATE: 02-05-2022

INDEX

ABSTRACT

The project is aimed at designing a density based dynamic traffic signal system where the timing of signal will change automatically on sensing the traffic density at any junction. Traffic congestion is a severe problem in most cities across the world and therefore it is time to shift more manual mode or fixed timer mode to an automated system with decision making capabilities.

Present day traffic signalling system is fixed time based which may render inefficient if one lane is operational than the others. To optimize this problem, we have made a framework for an intelligent traffic control system. Sometimes higher traffic density at one side of the junction demands longer green time as compared to standard allotted time We, therefore propose here a mechanism in which the time period of green light and red light is assigned on the basis of the density of the traffic present at that time. This is achieved by using PIR (proximity Infrared sensors). Once the density is calculated, the glowing time of green light is assigned by the help of the microcontroller (Arduino). The sensors which are present on sides of the road will detect the presence of the vehicles and sends the information to the microcontroller (Arduino) where it will decide how long a flank will be open or when to change over the signal lights. In subsequent sections, we have elaborated the procedure of this framework.

INTRODUCTION:

In today's high-speed life, traffic congestion becomes a serious issue in our day-to-day activities. It brings down the productivity of individual and thereby the society as lots of work hour is wasted in the signals. High volume of vehicles, the inadequate infrastructure and the irrational distribution of the signalling system are main reasons for these chaotic congestions. It indirectly also adds to the increase in pollution level as engines remain on in most cases, a huge volume of natural resources in forms of petrol and diesel is consumed without any fruitful outcome. Therefore, in order to get rid of these problems or at least reduce them to significant level, newer schemes need to be implemented by bringing in sensor-based automation technique in this field of traffic signalling system.

OBJECTIVE OF THE PROJECT:

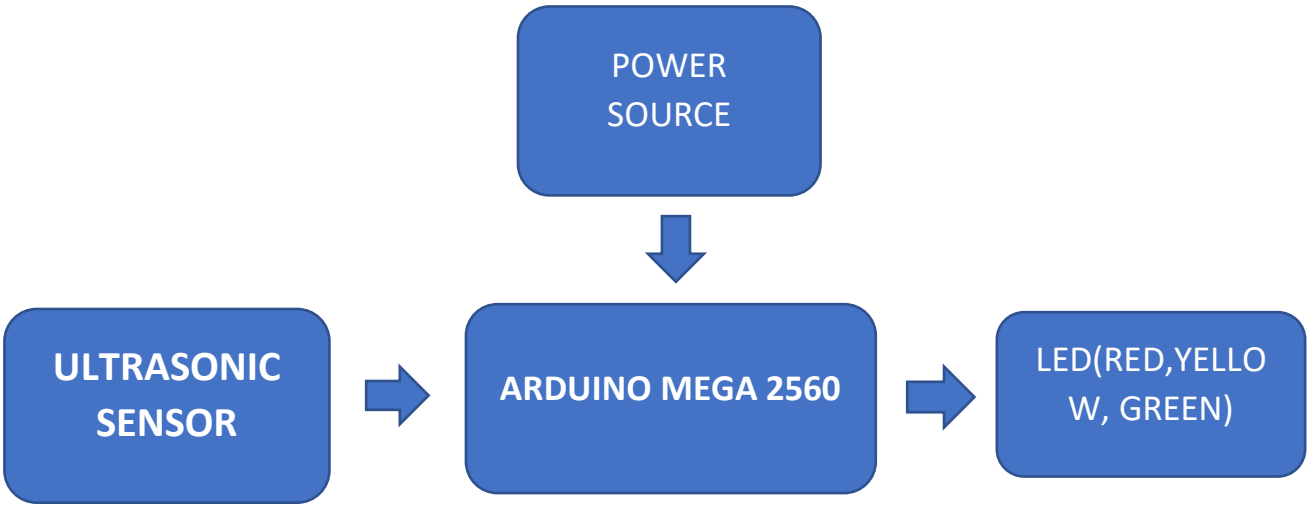
Our project aims at reducing traffic congestion and unwanted long-time delay during the traffic light switch overs especially when the traffic is very low. It is designed to be implemented in places nearing the junctions where the traffic signals are placed, in order to reduce the congestion in these junctions. It keeps a track of the vehicles in each road and accordingly adjusts the time for each traffic light signals. The higher the number of vehicles on the road the longer will be the time delay allotted for that corresponding traffic light signal. The main purpose of this project is, if there will be no traffic on the other signal, one shouldn't wait for that signal. The system will skip that signal and will move on the next one.

PRESENT TRAFFIC SIGNALING SYSTEM

Under present scenario, traffic control is achieved by the use of a system of hand signs by traffic police personnel, traffic signals, and markings. A comparable and matching education program is needed, through driver-licensing authorities, to assure that those who operate motor vehicles understand the rules of the road and the actions that they are required or advised to take when a particular control device is present. Each traffic control device is governed by standards of design and usage; for example, stop signs always have a red background and are octagonal in shape. Design standards allow the motorist to quickly and consistently perceive the sign in the visual field along the road. Standard use of colours and shape aids in this identification and in deciding on the appropriate course of action.

Under current circumstances, traffic lights are set on in the different directions with fixed time delay, following a particular cycle while switching from one signal to other creating unwanted and wasteful congestion on one lane while the other lanes remain vacant. The system we propose identify the density of traffic on individual lanes and thereby regulate the timing of the signals' timing. IR sensors count the obstructions and provide an idea about the traffic density on a particular lane and feed this response to a controller unit which will make the necessary decisions as and when required.

BLOCK DIAGRAM



LIST OF COMPONENTS

1.Arduino Mega 2560 --1

2.Ultrasonic sensor --1

3.Traffic Lights --8

LEDs-

i)Red

ii)green

iii)Yellow

4.Resistor (10k, 3w) --1

5.Pot-Hg --1 (high granularity interactive potentiometer)

6.Connecting wires

1.ARDUINO:

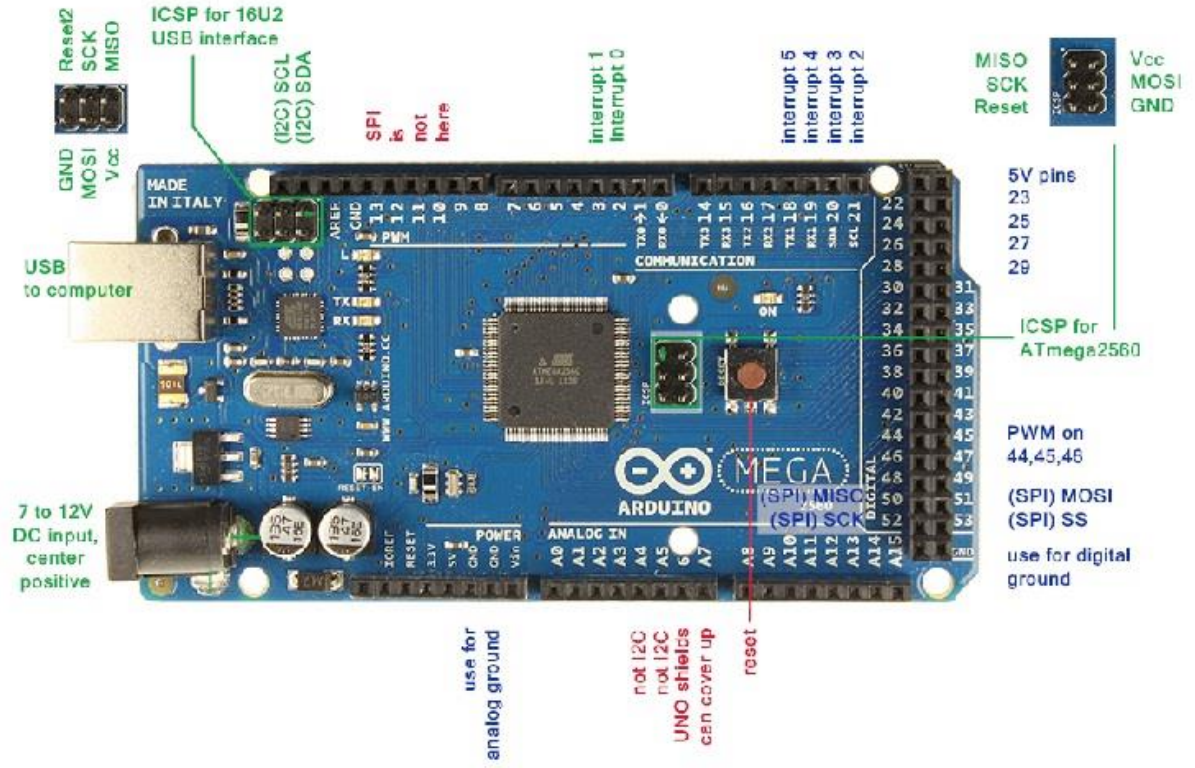
Arduino board is an open-source microcontroller board which is based on Atmega 2560 microcontroller. The microcontroller board like “Arduino Mega” depends on the ATmega2560 microcontroller. It includes digital input/output pins-54, where 16 pins are analog inputs, 14 are used like PWM outputs hardware serial ports (UARTs) – 4, a crystall osc-16 MHz, an ICSP header, a power jack, a USB connection, as well as an RST button. This board mainly includes everything which is essential for supporting the microcontroller. So, the power supply of this board can be done by connecting it to a PC using a USB cable, or battery or an AC-DC adapter. This board can be protected from the unexpected electrical discharge by placing a base plate.

General Pin functions:

- **LED**: There is a built-in LED driven by digital pin 13. When the pin is HIGH value, the LED is on, when the pin is LOW, it's off.
- **VIN**: The input voltage to the Arduino/Genuino board when it's using an external power source (as opposed to 5 volts from the USB connection or other regulated power source). You can supply voltage through this pin, or, if supplying voltage via the power jack, access it through this pin.
- **5V**: This pin outputs a regulated 5V from the regulator on the board. The board can be supplied with power either from the DC power jack (7 - 20V), the USB connector (5V), or the VIN pin of the board (7-20V). Supplying voltage via the 5V or 3.3V pins bypasses the regulator, and can damage the board.
- **3V3**: A 3.3 volt supply generated by the on-board regulator. Maximum current draw is 50 mA.
- **GND**: Ground pins.
- **IOREF**: This pin on the Arduino/Genuino board provides the voltage reference with which the microcontroller operates. A properly configured shield can read the IOREF pin voltage and select the appropriate power source or enable voltage translators on the outputs to work with the 5V or 3.3V.
- **Reset**: Typically used to add a reset button to shields which block the one on the board.

TECHNICAL SPECIFICATIONS:

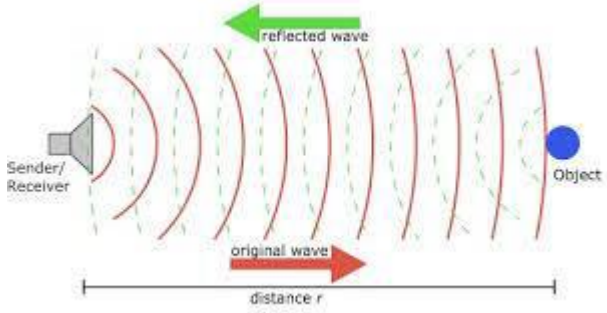
- Operating voltage: 5V
- Input voltage (recommended): 7-12V
- Input voltage (limits): 6-20V
- Digital I/O pins: 54 (of which 14 provide PWM output)
- Analog input pins: 16
- DC current per I/O pin: 40mA
- DC current for 3.3V pin: 50mA
- Flash Memory: 256 KB, 8KB used by bootloader
- SRAM: 8 KB
- EEPROM: 4 KB
- Clock Speed: 16 MHz



2.**ULSTRASONIC SENSORS:**

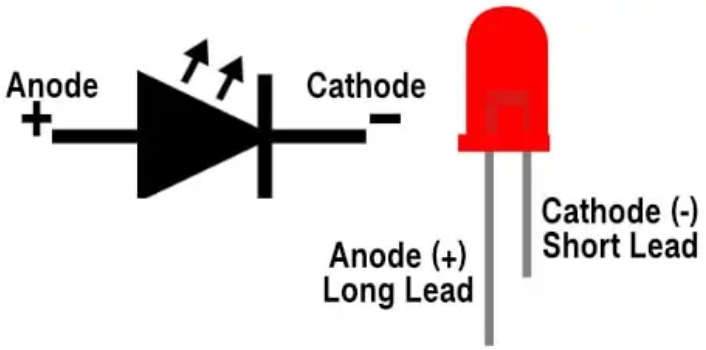
The HC-SR04 ultrasonic sensor uses SONAR to determine the distance of an object just like the bats do. It offers excellent non-contact range detection with high accuracy and stable readings in an easy-to-use package from 2 cm to 400 cm or 1” to 13 feet.

The operation is not affected by sunlight or black material, although acoustically, soft materials like cloth can be difficult to detect. It comes complete with ultrasonic transmitter and receiver module.



3.LED:

A light-emitting diode (LED) is a semiconductor device that emits light when an electric current flows through it. When current passes through an LED, the electrons recombine with holes emitting light in the process. LEDs allow the current to flow in the forward direction and blocks the current in the reverse direction.



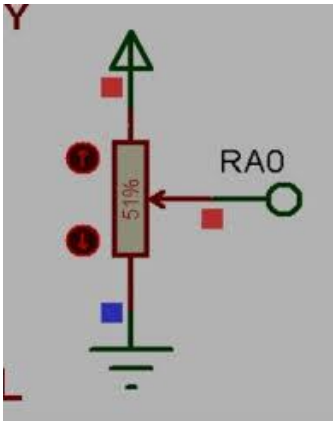
4.RESISTOR:

A resistor is a passive two-terminal electrical component that implements electrical resistance as a circuit element. In electronic circuits, resistors are used to reduce current flow, adjust signal levels, to divide voltages, bias active elements, and terminate transmission lines, among other uses.

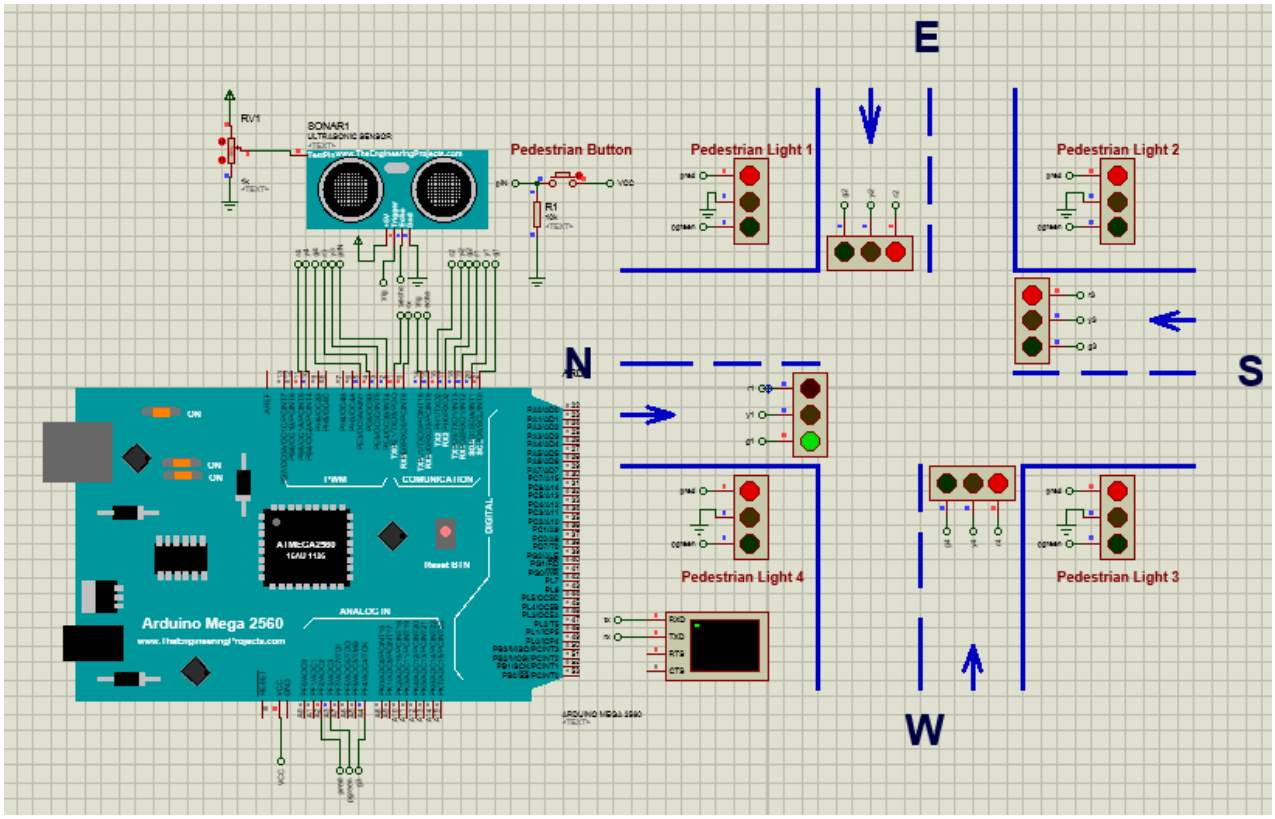
5.POTENTIOMETER HG:

Potentiometer hg is an active variable resistor that allow to change the resistance during simulation run time.

Here HG refers to High granularity which defines data to the most precision.



CIRCUIRT DIAGRAM



WORKING:

The model works on the principle of changing of Traffic signals based on the density through an assigned section of the road. There is one Ultrasonic sensor placed at the centre of crossroad which rotates 360 degree and checks the density of the 4 lanes area covered by the sensors. Here we are using Ultrasonic sensor to design an intelligent traffic control system. In order to measure the density of traffic on each side, Ultrasonic sensor will be kept at the centre of crossroad where it rotates 360 degrees. ULTRASONIC sensor consists of an Ultrasonic transmitter and an Ultrasonic receiver. Just as the name suggests, the Ultrasonic transmitter transmits the ultrasonic sound and the receiver is responsible to receive the ultrasonic sound. The whole system is controlled by the microcontroller which is the Arduino. Arduino is interfaced with ULTRASONIC sensor. As the vehicle passes through these Ultrasonic sensor will detect the vehicle & will send the information to the Arduino. The total no of Ultrasonic sensors required are 1 and Led's 12.

Three sets of LEDs via Green, Yellow and Red are used to indicate the GO state, Ready to Go state and WAIT state. The LEDs G (green), Y (yellow) and R (red) glow in following sequence.

- G1-R2-R3-R4
- Y1-R2-R3-R4
- R1-G2-R3-R4
- R1-Y2-R3-R4.
- R1-R2-G3-R4
- R1-R2-Y3-R4
- R1-R2-R3-G4
- R1-R2-R3-Y4

i.e., timing-based traffic signal will be automatically implemented when all the signals having same condition.

When Ultrasonic sensor start receiving signal the delay time for green signal of each lane will change accordingly. For distance less than 200 inch delay time is

2 seconds and for distance greater than 200 inch and less than 300 inch delay time is 4 seconds and for greater than 300 it is 6 seconds.

Let green light is On in the fourth traffic signal for 6 seconds and during that time ultrasonic sensor is receiving data of first lane then after green light it will take few seconds delay for yellow light and then first traffic signal green light will be on for the time calculated on the basis of data received from ultrasonic sensor

- R1-R2-R3-G4
- R1-R2-R3-Y4
- G1-R2-R3-R4

Whole time ultrasonic sensor rotates 360 degree and receives data of each lane and Arduino uses that data to change the delay time of each traffic signal accordingly

In between we can turn pedestrian light green and all other red so that people can cross the road.

ARDUINO CODE:

```
const int TrigPin = 14; // Trigger Pin of Ultrasonic Sensor
const int echoPin = 15; // Echo Pin of Ultrasonic Sensor
const int g1 = 21;
const int y1 = 20;
const int r1 = 19;
const int g2 = 18;
const int y2 = 17;
const int r2 = 16;
const int g3 = A4;
const int y3 = 3;
const int r3 = 4;
const int g4 = 5;
const int y4 = 10;
const int r4 = 11;
const int pRed = A2;
const int pgreen = A3;
const int pIN = 2;
long microsecondsToInches(long microseconds)
{
 return microseconds / 74 / 2;
}
long microsecondsToCentimeters(long microseconds)
{
   return microseconds / 29 / 2;
 }
void setup()
{
pinMode (r1, OUTPUT);
pinMode (y1, OUTPUT);
```

```
pinMode (g1, OUTPUT);
pinMode (r2, OUTPUT);
pinMode (y2, OUTPUT);
pinMode (g2, OUTPUT);
pinMode (r3, OUTPUT);
pinMode (y3, OUTPUT);
pinMode (g3, OUTPUT);
pinMode (r4, OUTPUT);
pinMode (y4, OUTPUT);
pinMode (g4, OUTPUT);
pinMode (pRed, OUTPUT);
pinMode (pgreen, OUTPUT);
pinMode (pIN, INPUT);
digitalWrite (r1, HIGH);
digitalWrite (r2, HIGH);
digitalWrite (r3, HIGH);
digitalWrite (r4, HIGH);
digitalWrite (pRed, HIGH);
Serial.begin(9600);
}
void loop() {
 long duration, inches, cm;
  pinMode(TrigPin, OUTPUT);
  digitalWrite(TrigPin, LOW);
  delayMicroseconds(2);
  digitalWrite(TrigPin, HIGH);
  delayMicroseconds(10);
  digitalWrite(TrigPin, LOW);
  pinMode(echoPin, INPUT);
  duration = pulseIn(echoPin, HIGH);
  inches = microsecondsToInches(duration);
```

```
  cm = microsecondsToCentimeters(duration);
  Serial.print("Distance = ");
  Serial.print(inches);
  Serial.print("in, ");
  Serial.print(cm);
  Serial.print("cm");
  Serial.println();
  if(inches >= 200)
  {
  int crossIn = digitalRead (pIN);
if (crossIn == HIGH){
  WalkCycle();}
else
{
delay(100);
digitalWrite (r1, LOW);
digitalWrite(g1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(4000);
}
int crossIn2 = digitalRead (pIN);
if (crossIn2 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g1,LOW);
digitalWrite(y1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
```

```
digitalWrite(r4,HIGH);
delay(2000);
}
int crossIn3 = digitalRead (pIN);
if (crossIn3 == HIGH){
  WalkCycle();}
else
{
digitalWrite(y1,LOW);
digitalWrite(r2,LOW);
digitalWrite(r1,HIGH);
digitalWrite(g2,HIGH); //2nd light
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(4000);
}
int crossIn4 = digitalRead (pIN);
if (crossIn4 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g2,LOW);
digitalWrite(y2,HIGH);
digitalWrite(r1,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(2000);
}
int crossIn5 = digitalRead (pIN);
if (crossIn5 == HIGH){
  WalkCycle();}
```

```
else
{
digitalWrite(y2,LOW);
digitalWrite(r3,LOW);
digitalWrite(g3,HIGH);//3rd light
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r4,HIGH);
delay(4000);
}
int crossIn6 = digitalRead (pIN);
if (crossIn6 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g3,LOW);
digitalWrite(y3,HIGH);
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r4,HIGH);
delay(2000);
}
int crossIn7 = digitalRead (pIN);
if (crossIn7 == HIGH){
  WalkCycle();}
else
{
digitalWrite(y3,LOW);
digitalWrite(r3,LOW);
digitalWrite(r4,LOW);
digitalWrite(g4,HIGH);//4th light
```

```
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
delay(4000);
}
int crossIn8 = digitalRead (pIN);
if (crossIn8 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g4,LOW);
digitalWrite(r4,LOW);
digitalWrite(y4,HIGH);
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
delay(2000);
}
digitalWrite(y4,LOW);
digitalWrite(r4,LOW);
digitalWrite(r1,LOW);
  }
  else if(inches >= 300 )
  {
    int crossIn = digitalRead (pIN);
if (crossIn == HIGH){
  WalkCycle();}
else
{
delay(100);
digitalWrite (r1, LOW);
```

```
digitalWrite(g1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(4000);
}
int crossIn2 = digitalRead (pIN);
if (crossIn2 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g1,LOW);
digitalWrite(y1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(2000);
}
int crossIn3 = digitalRead (pIN);
if (crossIn3 == HIGH){
  WalkCycle();}
else
{
digitalWrite(y1,LOW);
digitalWrite(r2,LOW);
digitalWrite(r1,HIGH);
digitalWrite(g2,HIGH); //2nd light
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(4000);
}
```

```
int crossIn4 = digitalRead (pIN);
if (crossIn4 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g2,LOW);
digitalWrite(y2,HIGH);
digitalWrite(r1,HIGH);
digitalWrite(r3,HIGH);
digitalWrite(r4,HIGH);
delay(2000);
}
int crossIn5 = digitalRead (pIN);
if (crossIn5 == HIGH){
  WalkCycle();}
else
{
digitalWrite(y2,LOW);
digitalWrite(r3,LOW);
digitalWrite(g3,HIGH);//3rd light
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r4,HIGH);
delay(4000);
}
int crossIn6 = digitalRead (pIN);
if (crossIn6 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g3,LOW);
```

```
digitalWrite(y3,HIGH);
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r4,HIGH);
delay(2000);
}
int crossIn7 = digitalRead (pIN);
if (crossIn7 == HIGH){
  WalkCycle();}
else
{
digitalWrite(y3,LOW);
digitalWrite(r3,LOW);
digitalWrite(r4,LOW);
digitalWrite(g4,HIGH);//4th light
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
delay(4000);
}
int crossIn8 = digitalRead (pIN);
if (crossIn8 == HIGH){
  WalkCycle();}
else
{
digitalWrite(g4,LOW);
digitalWrite(r4,LOW);
digitalWrite(y4,HIGH);
digitalWrite(r1,HIGH);
digitalWrite(r2,HIGH);
digitalWrite(r3,HIGH);
```

```
delay(2000);
}
digitalWrite(y4,LOW);
digitalWrite(r4,LOW);
digitalWrite(r1,LOW);
 }
 void WalkCycle() {
 delay(350);
digitalWrite (g1, LOW);digitalWrite (g2, LOW);digitalWrite (g3, LOW);digitalWrite (g4, LOW);
digitalWrite (y1, LOW);digitalWrite (y2, LOW);digitalWrite (y3, LOW);digitalWrite (y4, LOW);
digitalWrite (r1, HIGH);digitalWrite (r2, HIGH);digitalWrite (r3, HIGH);digitalWrite (r4, HIGH);
digitalWrite (pgreen, HIGH);
 digitalWrite (pRed, LOW);
 delay (3000);
 digitalWrite (pgreen, LOW);
 //digitalWrite(pIN, LOW);  // Turn off green Pedestrian Light
 delay(250);
 for (int x = 0; x < 5; x++) {  // Flash green Ped LED 5X
   digitalWrite(pgreen, HIGH);
   delay(250);
   digitalWrite(pgreen, LOW);
   delay(250);
 }
 digitalWrite(pRed, HIGH);
}
```

RESULTS AND DISCUSSIONS:

From the series of experiments we have conducted the following results were

obtained:

- Traffic can be cleared without any irregularities
- Time can be shared evenly for all intersections
- Effective time management

CONCLUSION :

There is exigent need of efficient traffic management system in our country, as India meets with 384 road accidents every day. To reduce this congestion and unwanted time delay in traffic an advanced system is designed here in this project. With field application of this technology, the maddening chaos of traffic can be effectively channelized by distributing the time slots based on the merit of the vehicle load in certain lanes of multi junction crossing. We have successfully implemented the prototype at laboratory scale with remarkable outcome. The next step forward is to implement this schema is real life scenario for first hand results, before implementing it on the largest scale. We believe that this may bring a revolutionary change in traffic management system on its application in actual field environment.

REFERENCE:

- Understanding How Ultrasonic Sensors Work | MaxBotix Inc.
- https://www.arduino.cc/en/Guide/Introduction
- https://en.wikipedia.org/wiki/Potentiometer
- https://www.go2kennewick.com/354/Pedestrian-Signals
- https://www.electronicshub.org/density-based-traffic-signal-system-using-microcontroller