TRAFFIC MANAGEMENT USING IoT

# NAME:  AVIKA M

COLLEGE CODE:9530

NAN MUDHALVAN ID:  au953021106011

PHASE-3 PROJECT SUBMISSION

PROJECT TITLE:  TRAFFIC MANAGEMENT SYSTEM

PHASE 3:  DEVELOPEMENT PART 1

# INTRODUCTION:

PROBLEM STATEMENT:

Traffic congestion consists of incremental delay, consumption, pollution emission and stress that result from interference vehicles in traffic stream, particularly as traffic volumes approaches a road’s capacity.

DESCRIPTION:

Traffic management concerns the control, planning, and purchasing of transport services needed to physically move road vehicles.

COMPONENTS REQUIRED:

* ARDUINO UNO
* TRAFFIC SENSOR
* JUMPER WIRES
* BREAD BOARD
* LED

### SOFTWARE REQUIRED:

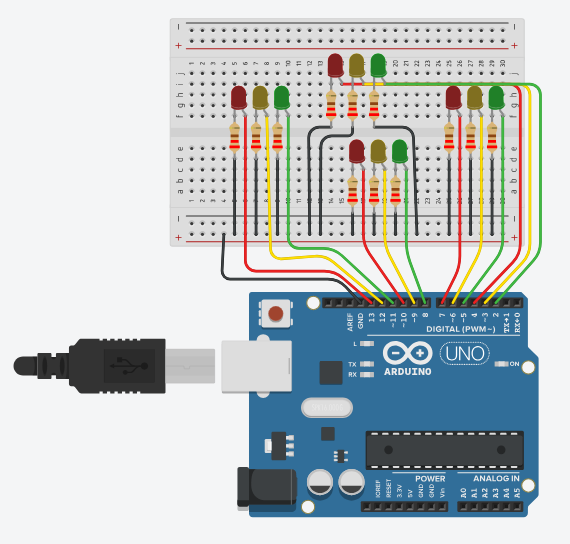
* Arduino IDE:

Used to connect Arduino UNO and upload the file into them for further processing.

* TINKERCAD:

Used for creating digital circuit and to simulate that.

# **CIRCUIT DIAGRAM:**



# SOURCE CODE TO SIMULATE ARDUINO:

int red\_1=13;  
int orange\_1=12;  
int green\_1=11;  
int red\_2=10;  
int orange\_2=9;  
int green\_2=8;  
int red\_3=7;  
int orange\_3=6;  
int green\_3=5;  
int red\_4=4;  
int orange\_4=3;  
int green\_4=2;  
  
void direction\_1\_green(void)  
{  
    digitalWrite(red\_1,LOW);  
    digitalWrite(orange\_1,LOW);  
    digitalWrite(green\_1,HIGH);  
    digitalWrite(red\_2,HIGH);  
    digitalWrite(orange\_2,LOW);  
    digitalWrite(green\_2,LOW);  
    digitalWrite(red\_3,HIGH);  
    digitalWrite(orange\_3,LOW);  
    digitalWrite(green\_3,LOW);  
    digitalWrite(red\_4,HIGH);  
    digitalWrite(orange\_4,LOW);  
    digitalWrite(green\_4,LOW); }  
  
void direction\_2\_orange(void)  
{  
    digitalWrite(red\_1,HIGH);  
    digitalWrite(orange\_1,LOW);  
    digitalWrite(green\_1,LOW);  
    digitalWrite(red\_2,LOW);  
    digitalWrite(orange\_2,HIGH);  
    digitalWrite(green\_2,LOW);  
    digitalWrite(red\_3,HIGH);  
    digitalWrite(orange\_3,LOW);  
    digitalWrite(green\_3,LOW);  
    digitalWrite(red\_4,HIGH);  
    digitalWrite(orange\_4,LOW);  
    digitalWrite(green\_4,LOW);  
     
}  
  
void direction\_2\_green(void)  
{  
    digitalWrite(red\_1,HIGH);  
    digitalWrite(orange\_1,LOW);  
    digitalWrite(green\_1,LOW);  
    digitalWrite(red\_2,LOW);  
    digitalWrite(orange\_2,LOW);  
    digital Write(green\_2,HIGH);  
    digitalWrite(red\_3,HIGH);  
    digitalWrite(orange\_3,LOW);  
    digitalWrite(green\_3,LOW);  
    digitalWrite(red\_4,HIGH);  
    digitalWrite(orange\_4,LOW);  
    digitalWrite(green\_4,LOW);  
     
}  
  
void direction\_3\_orange(void)  
  
{  
    digitalWrite(red\_1,HIGH);  
    digitalWrite(orange\_1,LOW);  
    digitalWrite(green\_1,LOW);  
    digitalWrite(red\_2,HIGH);  
    digitalWrite(orange\_2,LOW);  
    digitalWrite(green\_2,LOW);  
    digitalWrite(red\_3,LOW);  
    digitalWrite(orange\_3,HIGH);  
    digitalWrite(green\_3,LOW);  
    digitalWrite(red\_4,HIGH);  
    digitalWrite(orange\_4,LOW);  
    digitalWrite(green\_4,LOW);  
    }  
  
void direction\_3\_green(void)  
{  
    digitalWrite(red\_1,HIGH);  
    digitalWrite(orange\_1,LOW);  
    digitalWrite(green\_1,LOW);  
    digitalWrite(red\_2,HIGH);  
    digitalWrite(orange\_2,LOW);  
    digitalWrite(green\_2,LOW);  
    digitalWrite(red\_3,LOW);  
    digitalWrite(orange\_3,LOW);  
    digitalWrite(green\_3,HIGH);  
    digitalWrite(red\_4,HIGH);  
    digitalWrite(orange\_4,LOW);  
    digitalWrite(green\_4,LOW);  
    }  
    void direction\_4\_orange(void)  
{  
    digitalWrite(red\_1,HIGH);  
    digitalWrite(orange\_1,LOW);  
    digitalWrite(green\_1,LOW);  
    digitalWrite(red\_2,HIGH);  
    digitalWrite(orange\_2,LOW);  
    digitalWrite(green\_2,LOW);  
    digitalWrite(red\_3,HIGH);  
    digitalWrite(orange\_3,LOW);  
    digitalWrite(green\_3,LOW);  
    digitalWrite(red\_4,LOW);  
    digitalWrite(orange\_4,HIGH);  
    digitalWrite(green\_4,LOW);  
    }  
    void direction\_4\_green(void)  
    //green LED of direction 4 will turn ON  
{  
     digitalWrite(red\_1,HIGH);  
     digitalWrite(orange\_1,LOW);  
     digitalWrite(green\_1,LOW);  
     digitalWrite(red\_2,HIGH);  
     digitalWrite(orange\_2,LOW);  
     digitalWrite(green\_2,LOW);  
     digitalWrite(red\_3,HIGH);  
     digitalWrite(orange\_3,LOW);  
     digitalWrite(green\_3,LOW);  
     digitalWrite(red\_4,LOW);  
     digitalWrite(orange\_4,LOW);  
     digitalWrite(green\_4,HIGH);  
     }  
     void direction\_1\_orange(void)  
     //orange LED of direction 1 will turn ON  
     {  
         
         digitalWrite(red\_1,LOW);  
          
        digitalWrite(orange\_1,HIGH);  
        digitalWrite(green\_1,LOW);  
        digitalWrite(red\_2,HIGH);  
        digitalWrite(orange\_2,LOW);  
        digitalWrite(green\_2,LOW);  
        digitalWrite(red\_3,HIGH);  
        digitalWrite(orange\_3,LOW);  
        digitalWrite(green\_3,LOW);  
        digitalWrite(red\_4,HIGH);  
        digitalWrite(orange\_4,LOW);  
        digitalWrite(green\_4,LOW); }  
        void setup()  
        {  
            // Declaring all the LED's as output  
  
for(int I=2;i<=13;i++)  
pinMode(I, OUTPUT);  
  
} void loop()  
//In the loop function, we controlled the signal one // by one to control the flow of traffic.  
{  
    direction\_1\_green();  
    delay([5000](tel:5000));  
    direction\_2\_orange(); delay([3000](tel:3000));  
    direction\_2\_green(); delay([5000](tel:5000));  
    direction\_3\_orange(); delay([3000](tel:3000));  
    direction\_3\_green(); delay([5000](tel:5000));  
    direction\_4\_orange(); delay([3000](tel:3000));  
    direction\_4\_green(); delay([5000](tel:5000));  
    direction\_1\_orange(); delay([3000](tel:3000)); }

# UPLOAD THE CODE TO YOUR ARDUINO UNO:

* Open the ARDUINO IDE on your computer.
* connect your Arduino UNO to your computer via USB.
* Select the correct board and plot under the “tools “menu.
* copy and paste the above code into the Arduino IDE.
* click the “upload” button to upload the code to your Arduino Uno.
* simulate the code to get the output.

# TRAFFIC SYSTEM:

# 

# SOURCE  CODE IN PYTHON:

import time  
import random  
  
class Vehicle:  
    def \_\_init\_\_(self, vehicle\_id, speed):  
        self.vehicle\_id = vehicle\_id  
        [self.speed](http://self.speed) = speed  
  
class TrafficLight:  
    def \_\_init\_\_(self):  
        [self.state](http://self.state) = 'red'  
  
    def switch\_state(self):  
        [self.state](http://self.state) = 'green' if [self.state](http://self.state) == 'red' else 'red'  
  
class TrafficManager:  
    def \_\_init\_\_(self, num\_vehicles, num\_iterations):  
        [self.vehicles](http://self.vehicles) = [Vehicle(i, [random.randint](http://random.randint)(30, 70)) for i in range(num\_vehicles)]  
        self.traffic\_light = TrafficLight()  
        self.num\_iterations = num\_iterations  
  
    def run\_simulation(self):  
        for iteration in range(self.num\_iterations):  
            print(f"Iteration {iteration + 1}: Traffic Light is {[self.traffic\_light.state](http://self.traffic_light.state)}")  
            for vehicle in [self.vehicles](http://self.vehicles):  
                if [self.traffic\_light.state](http://self.traffic_light.state) == 'green':  
                    distance = [vehicle.speed](http://vehicle.speed) \* 2  
                    print(f"Vehicle {vehicle.vehicle\_id} moves {distance} meters.")  
                else:  
                    print(f"Vehicle {vehicle.vehicle\_id} stops at the red light.")  
            self.traffic\_light.switch\_state()  
            [time. Sleep](http://time.sleep)(1)  
  
def main():  
    num\_vehicles = 5  
    num\_iterations = 5  
    traffic\_manager = TrafficManager(num\_vehicles, num\_iterations)  
    traffic\_manager.run\_simulation()  
  
if \_\_name\_\_ == "\_\_main\_\_":  
    main()

**CONCLUSION:**

To summarize, the project involves the use of IoT devices such as traffic cameras, sensors, and traffic lights, along with a centralized platform for data collection, analysis, and visualization. The system aims to optimize traffic flow, enhance road safety, and promote environmental sustainability through real-time monitoring and predictive algorithms. By utilizing advanced data analytics and visualization techniques, the project facilitates informed decision-making for traffic authorities and enhances the overall efficiency of the transportation system.