TRAFFIC MANAGEMENT SYSTEM USING IOT

TEAM MEMNBER

111421106053

U.Bhanu Prasad

PHASE 4 : <u>DEVELOPMENT PART 2</u>

Abstract:

Traffic management system is one of the major proportions of a smart city. With the rapid growth of population and rapid increase of vehicles across the whole country which further leads to the traffic Congestion which is usually seen on roads. Nowadays traffic congestion is a difficult issue to deal with as number of vehicles is increasing day by day. To tackle various issues of traffic on roads and to help authorities in proper planning, a smart traffic management system using the Internet of Things (IoT) is proposed in this paper. A simple, effective and less costly method is used to optimize traffic flow on roads and an algorithm is devised to manage various traffic situations efficiently and automatically. For this purpose, the system takes traffic density as input from 8 different sensors which are there in 4 lanes which manages traffic signals. Besides this manual control of this system using Wi-Fi is also used to prioritize the emergency vehicles such as ambulances and fire brigade vehicles during a traffic jam, so that we can open the specific lane with the remote using Wi-Fi. To show the effectiveness of this proposed traffic management system, a prototype is developed which optimizes the traffic having lesser cost and is very effective. And the real time data will also be visible in the mobile phone through application

SYSTEM DESIGN AND ARCHITECHTURE

ARDUINO NANO

An 8 bit Microchip AVR which is small, complete and bread board friendly board based on the Atmega328. It is the main CPU of our Project, in which we all the program will run.

POWER SUPPLY MODULE

A power supply is a hardware component that provides power to any electrical device.

IR SENSOR

These sensors are used to detect the object through infrared rays. The rays which are thrown from the sensors are reflected back by the object by which it encountered and then after captured by these infrared sensors which further gets converted into electric signals. These sensors are put sideways for giving us the density of vehicles in the specific lane. Infrared sensors are used for signal control, detection of pedestrians in crosswalks and transmission of traffic information [8]. The basic disadvantages of infrared sensors are that the operation of the system may be affected due to fog; also installation and maintenance of the system is tedious

WIFI MODULE

It is used to give microcontroller access to your wife network.

LED

Light bulbs are used for output and instruction for this system.

BLYNK APP

It is a mobile application for output and verification for real time data collected.

Flow Chart with Algorithm

Case1

 1. Start

 2. Check the vehicle Density

3. Vehicle density++

 4. Is vehicle density <Threshold

5. Yes

 6. Normal Traffic

 7. Give green signal to each Lane in a sequential manner.

Case2

 1. Start

 2. Check the vehicle Density

3. Vehicle density++

4. Is vehicle density <Threshold

5. No

6. Status =congestion

7. Compare the number of density in each lane

 8. Open the lane with highest number of density

9. Remove current Lane from Comparison

10. Then start once again

CODE – for density of vehicals:

```
// Simulate traffic density data from sensors
const sensors = [
    { lane: 1, data: generateRandomDensity() },
    { lane: 2, data: generateRandomDensity() },
    { lane: 3, data: generateRandomDensity() },
    { lane: 4, data: generateRandomDensity() }
];
function generateRandomDensity() {
    // Simulate random traffic density values
    return Math.floor(Math.random() * 100); // Adjust range as needed
}
// Periodically update sensor data
```
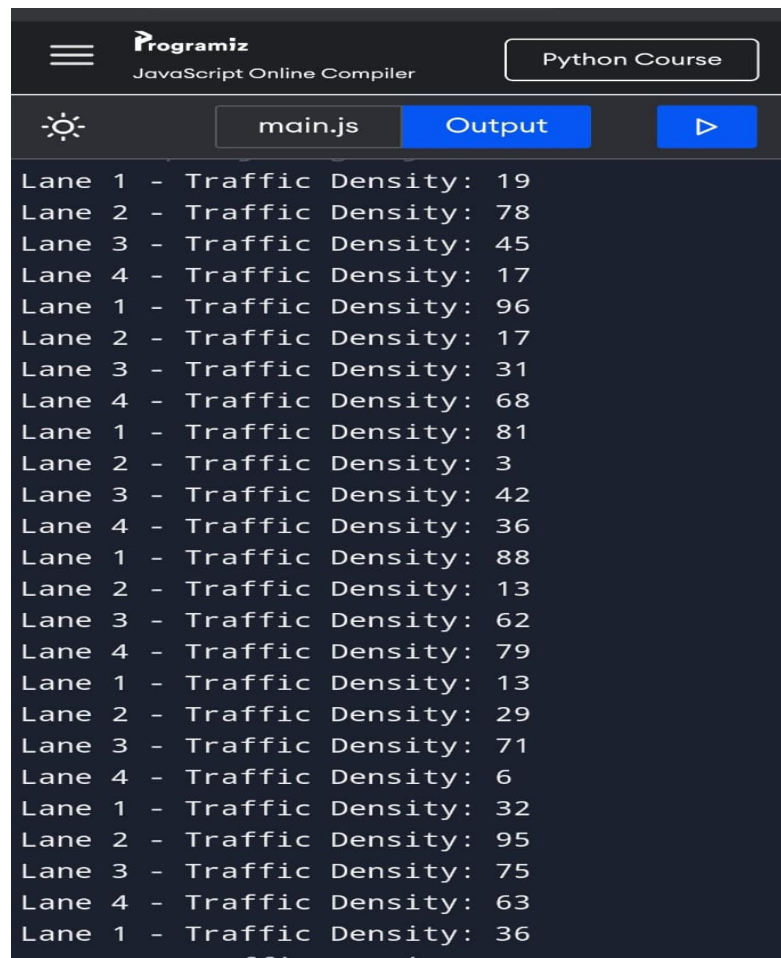
```javascript
setInterval(() => {

    sensors.forEach(sensor => {

        sensor.data =
generateRandomDensity();

        console.log(Lane ${sensor.lane} -
Traffic Density: ${sensor.data});

    });

    // Send sensor data to the central system
for analysis and contro, This part would
involve more complex IoT communication,
not shown here

}, 5000); // Update every 5 seconds
(adjust as needed)/
```

The coding is in the language of java script. And the BLYNK
APPILICATION is used to console the code.

Output

- import java.awt.*;

```
Lane 1 - Traffic Density: 19
Lane 2 - Traffic Density: 78
Lane 3 - Traffic Density: 45
Lane 4 - Traffic Density: 17
Lane 1 - Traffic Density: 96
Lane 2 - Traffic Density: 17
Lane 3 - Traffic Density: 31
Lane 4 - Traffic Density: 68
Lane 1 - Traffic Density: 81
Lane 2 - Traffic Density: 3
Lane 3 - Traffic Density: 42
Lane 4 - Traffic Density: 36
Lane 1 - Traffic Density: 88
Lane 2 - Traffic Density: 13
Lane 3 - Traffic Density: 62
Lane 4 - Traffic Density: 79
Lane 1 - Traffic Density: 13
Lane 2 - Traffic Density: 29
Lane 3 - Traffic Density: 71
Lane 4 - Traffic Density: 6
Lane 1 - Traffic Density: 32
Lane 2 - Traffic Density: 95
Lane 3 - Traffic Density: 75
Lane 4 - Traffic Density: 63
Lane 1 - Traffic Density: 36
```

Code – for signal indication:

```java
import java.awt.*;

import javax.swing.*;

import java.awt.event.*;

import javax.swing.border.*;

public class TrafficLight extends JFrame implements ActionListener {

    JButton b1, b2, b3;

    Signal green = new Signal(Color.green);

    Signal yellow = new Signal(Color.yellow);

    Signal red = new Signal(Color.red);
```

```java
public TrafficLight(){
    super("Traffic Light");
    getContentPane().setLayout(new GridLayout(2, 1));
    b1 = new JButton("Red");
    b2 = new JButton("Yellow");
    b3 = new JButton("Green");
    b1.addActionListener(this);
    b2.addActionListener(this);
    b3.addActionListener(this);
    green.turnOn(false);
    yellow.turnOn(false);
    red.turnOn(true);
    JPanel p1 = new JPanel(new GridLayout(3,1));
    p1.add(red);
    p1.add(yellow);
    p1.add(green);
    JPanel p2 = new JPanel(new FlowLayout());
    p2.add(b1);
    p2.add(b2);
    p2.add(b3);
    getContentPane().add(p1);
    getContentPane().add(p2);
    pack();
}
public static void main(String[] args){
```

```java
        TrafficLight tl = new TrafficLight();
        tl.setVisible(true);
    }
public void actionPerformed(ActionEvent e){
    if (e.getSource() == b1){
        green.turnOn(false);
        yellow.turnOn(false);
        red.turnOn(true);
    } else if (e.getSource() == b2){
        yellow.turnOn(true);
        green.turnOn(false);
        red.turnOn(false);
    } else if (e.getSource() == b3){
        red.turnOn(false);
        yellow.turnOn(false);
        green.turnOn(true);
    }
    }
}
class Signal extends JPanel{
    Color on;
    int radius = 40;
    int border = 10;
    boolean change;
Signal(Color color){
```

```java
        on = color;
        change = true;
    }
public void turnOn(boolean a){
        change = a;
        repaint();
    }
public Dimension getPreferredSize(){
        int size = (radius+border)*2;
        return new Dimension( size, size );
    }
public void paintComponent(Graphics g){
        g.setColor( Color.black );
        g.fillRect(0,0,getWidth(),getHeight());
        if (change){
            g.setColor( on );
        } else {
            g.setColor( on.darker().darker().darker() );
        }
        g.fillOval( border,border,2*radius,2*radius );
    }
}
```

OUTPUT :

The code is executed successfully, development of the app and web services is successfully implemented.