|  |
| --- |
| Difference Between NAAC & NBA Accreditation - Haq Se EngineerPREC LONIJai Shriram Engineering College (@JSREC09) / Twitter**JAI SHRIRAM ENGINEERING COLLEGE**  **TIRUPPUR – 638 660**  Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai  Recognized by UGC & Accredited by NAACandNBA (CSE and ECE) |

**DEPARTMENT OF**

**ELECTRONICS AND COMMUNICATION ENGINEERING**

**IBM - Naan Mudhalvan**

**Internet of Things Group 3**

**Phase 5 - Project Submission**

**NAME : E.MONISHA**

**NM ID :AU711221106017**

**YEAR : III**

**TRAFFIC MANAGEMENT SYSTEM USING INTERNET OF THINGS(IOT)**

**ABSTRACT:**

Over the years, there has been a sudden increase in the number of vehicles on the road. Traffic congestion is a growing problem everyone faces in their daily life. Manual control of traffic by traffic police has not proved to be efficient. Also the predefined set time for the signal at all circumstances (low and high traffic density has not solved this problem. A model to effectively solve the above mentioned problems by using Internet of Things (IOT) is proposed. We use cloud for internet based computing, where different services such as server, storage and application are delivered for traffic management. A network of sensors is used to track the number of vehicles and the traffic congestion at the intersections on a road and rerouting will be done on the basis of the traffic density on the lanes of a road.

Keywords: IOT, Sensors, Microcontroller.

**INTRODUCTION:**

**HYPOTHESIS:**

A smart traffic management system utilizing sensor data, communication and auto-mated algorithms is to be developed to keep traffic flowing more smoothly. The aim is to optimally control the duration of green or red light for a specific traffic light at an intersection. The traffic signals should not flash the same stretch of green or red all the time, but should depend on the number of cars present. When traffics heavy in one direction, the green lights should stay on longer; less traffic should mean the red lights should be on for longer time interval. This solution is expected to eliminate inefficiencies at intersections and minimize the cost of commuting and pollution.

**PROJECT OBJECTIVES:**

The objectives of a traffic management project typically include:

1. Improve Traffic Flow: Enhance the overall flow of vehicular and pedestrian traffic to reduce congestion and minimize delays.
2. Increase Safety: Implement measures to reduce accidents and enhance road safety for all road users.
3. Reduce Environmental Impact: Minimize emissions and environmental impact through efficient traffic management strategies.
4. Optimize Infrastructure: Ensure efficient use of road infrastructure by implementing measures such as signal synchronization, lane management, and intelligent transportation systems.
5. Enhance Public Transportation: Promote the use of public transportation and develop infrastructure for buses, trams, and other mass transit options.
6. Provide Real-Time Information: Offer real-time traffic updates and information to drivers and commuters to help them make informed decisions.
7. Support Emergency Services: Ensure quick and efficient access for emergency services in the event of accidents or disasters.
8. Promote Sustainable Transportation: Encourage walking, cycling, and other eco-friendly modes of transportation.
9. Manage Special Events: Handle traffic management for special events, construction, and road closures effectively.
10. Public Engagement and Education: Educate the public about safe driving practices and the benefits of the traffic management initiatives.
11. Data Collection and Analysis: Continuously collect and analyze traffic data to adapt and improve traffic management strategies.

These objectives can vary depending on the specific goals and challenges of a given traffic management project.

**IOT SENSOR SETUP:**

A traffic management project using IOT sensor setup typically involves deploying various sensors and devices to monitor and manage traffic flow and congestion in real-time. Here's a general overview of such a project:

1. Sensor Deployment: IOT sensors, such as cameras, ultrasonic sensors, infrared sensors, and vehicle detectors, are strategically placed at key locations throughout the city or on specific roadways. These sensors can detect various parameters, including vehicle presence, speed, and traffic density.
2. Data Collection: The sensors continuously collect data and transmit it wirelessly to a central server or cloud platform. This data includes information about traffic conditions, such as the number of vehicles, vehicle types, and their speeds.
3. Data Processing: The collected data is processed and analyzed in real-time. Machine learning algorithms can be employed to predict traffic patterns, identify congestion points, and analyze traffic trends.
4. Traffic Management: Traffic authorities can access the processed data through a user-friendly interface, enabling them to make informed decisions. They can adjust traffic signals, update electronic road signs, or dispatch traffic personnel to address congestion and incidents promptly.
5. Communication and Alerts: IOT sensors can also be used to send alerts to drivers through variable message signs, mobile apps, or navigation systems, providing real-time information about traffic conditions and suggesting alternative routes.
6. Environmental Monitoring: Some IOT setups may include environmental sensors to monitor air quality and emissions, contributing to a holistic approach to traffic management and urban planning.
7. Integration with Smart Systems: In more advanced setups, the traffic management project can integrate with other smart city systems, such as smart street lighting, public transportation, and emergency services, to further optimize traffic flow.
8. Data Storage and Analytics: Collected data is often stored for historical analysis, helping authorities make long-term decisions related to infrastructure improvements, road expansions, and urban planning.
9. Cost and Efficiency Benefits: By reducing congestion, improving traffic flow, and decreasing fuel consumption and emissions, IOT-based traffic management projects can have significant economic and environmental benefits for a city.
10. Public Engagement: Citizen engagement can be enhanced through public access to real-time traffic data and participation in smart city initiatives, ultimately leading to more informed and satisfied communities.

IOT sensor setups for traffic management are a vital component of smart city initiatives, helping to alleviate traffic congestion, enhance safety, and create more efficient urban environments.

**MOBILE APP DEVELOPMENT:**

Developing a traffic management project mobile app involves creating a software application that assists in managing and optimizing traffic flow, safety, and information for users. Here are the key aspects involved

**User Interface (UI):** The app should have an intuitive and user-friendly interface for easy navigation. It should include maps, icons, and menus for various functions.

**Real-time Data:** The app needs to access and display real-time data, such as traffic conditions, accidents, road closures, and weather updates. This data can be sourced from various sensors, cameras, and traffic agencies.

**GPS Integration:** Utilize GPS to provide users with their current location, turn-by-turn navigation, and real-time traffic updates. Users should also be able to plan routes based on traffic conditions.

**Traffic Alerts:** Implement a notification system to alert users about accidents, road closures, or severe congestion on their routes, allowing them to choose alternatives.

**Live Cameras:** Include live traffic cameras at key locations to give users a visual perspective of current traffic conditions.

**Traffic Management Tools:** For authorities, include tools for monitoring and managing traffic. This could involve adjusting traffic signals, coordinating with law enforcement, or implementing emergency plans.

**User-Generated Content:** Allow users to report incidents, accidents, and road conditions. These reports can be shared with other users to increase awareness.

**Integration with Other Services:** Integrate the app with public transportation information, ride-sharing services, and parking availability to offer comprehensive commuting solutions.

**Emergency Services Integration:** Provide quick access to emergency services and first responders in case of accidents or emergencies.

**Analytics and Reporting:** For traffic agencies, incorporate data analytics and reporting features to track and analyze traffic patterns, enabling better decision-making.

**Security:** Ensure data security and privacy for users and traffic authorities, especially when collecting and sharing sensitive information.

**Scalability:** Design the app to handle a large user base and the potential for increased data traffic during peak times.

**Compatibility:** Develop versions of the app for various mobile platforms (iOS and Android) to reach a broader audience.

**Regulatory Compliance:** Ensure that the app complies with local traffic regulations and data protection laws.

**Testing and Quality Assurance:** Rigorously test the app for stability, performance, and accuracy of real-time data before its release.

**Feedback Mechanism:** Implement a feedback system to collect user opinions and suggestions for continuous improvement.

**Maintenance and Updates:** Regularly update the app to keep it current and improve its functionality based on user feedback and evolving traffic conditions.

Development teams often consist of UI/UX designers, mobile app developers, database specialists, and data analysts. The success of a traffic management project mobile app depends on effective collaboration between the relevant transportation authorities and the development team to ensure accurate data and seamless functionality.

**RASPBERRY PI INTEGRATION:**

Integrating Raspberry Pi into a traffic management project involves using the Raspberry Pi as a versatile and cost-effective controller for various traffic-related tasks. Here's a general overview of how it can be done:

**Hardware Setup:**

Raspberry Pi board: Select an appropriate Raspberry Pi model with the required processing power and connectivity options.

Camera Module: Attach a compatible camera module for image and video capture.

Sensors: Connect sensors (e.g., ultrasonic, infrared) to detect vehicles and pedestrians.

LED Displays: Interface LED displays or traffic lights for signaling.

**Software Development:**

Operating System: Install a Linux-based OS on the Raspberry Pi, such as Raspbian (now Raspberry Pi OS).

Programming: Develop software using programming languages like Python or C/C++.

Image Processing: Implement image processing algorithms to analyze traffic conditions.

Data Analysis: Collect and process data from sensors and cameras.

Communication: Establish communication protocols for sending/receiving data to/from central servers or other devices.

**Traffic Monitoring:**

Vehicle Detection: Use the camera and sensors to detect and count vehicles.

Traffic Flow Analysis: Analyze vehicle speed and density for traffic flow monitoring.

License Plate Recognition: Implement license plate recognition for security and traffic enforcement.

**Traffic Control:**

Traffic Signal Control: Adjust traffic signals based on real-time traffic conditions.

Dynamic Lane Management: Control lane direction or usage as needed.

Emergency Response: Automatically respond to emergencies by changing traffic patterns.

**Data Reporting:**

Data Visualization: Create graphical representations of traffic data.

Remote Access: Enable remote monitoring and control via web interfaces or apps.

Data Storage: Store historical data for analysis and decision-making.

**Integration with Other Systems:**

IOT Connectivity: Integrate with Internet of Things (IOT) platforms for seamless data sharing.

Cloud Integration: Upload data to the cloud for further analysis and long-term storage.

**Safety and Redundancy:**

Implement safety mechanisms to handle unexpected events or failures.

Ensure redundancy in critical components to maintain system operation.

**Testing and Calibration:**

Thoroughly test the system in various traffic scenarios.

Calibrate sensors and cameras for accuracy.

**Maintenance and Updates:**

Regularly update the software to fix bugs and improve performance.

Perform routine maintenance to ensure the system's reliability.

**Compliance and Regulations:**

Ensure the project adheres to local traffic regulations and safety standards.

Integrating a Raspberry Pi into a traffic management project offers flexibility, affordability, and the potential for real-time monitoring and control. However, it's important to carefully plan and implement the project to ensure its effectiveness and reliability.

**CODE IMPLEMENTATION:**

Implementing a traffic management project using Python involves various components, including data collection, analysis, and control. Here's a high-level overview of the steps and components you might consider:

1. **Data Collection**:
   * Use sensors, cameras, or APIs to collect real-time traffic data, such as vehicle counts, speed, and congestion.
2. **Data Processing**:
   * Process and store the collected data in a database, such as SQLite or MySQL, for analysis and decision-making.
3. **Data Analysis**:
   * Use Python libraries like Pandas and NumPy to analyze the traffic data to identify patterns and congestion points.
4. **Visualization**:
   * Utilize libraries like Matplotlib or Plot to create visualizations like traffic heat maps, real-time traffic flow maps, and congestion alerts.
5. **Machine Learning (Optional)**:
   * Implement machine learning models to predict traffic patterns or congestion, allowing for proactive management.
6. **Traffic Control**:
   * Depending on the scale and resources, you can control traffic lights or signals in real-time. Use libraries like RPi,GPIO if you're working with Raspberry Pi.
7. **User Interface**:
   * Create a user interface for monitoring and controlling traffic. You can use web frameworks like Flask or Django for this.
8. **Communication**:
   * Establish communication with traffic control systems using protocols such as MQTT or HTTP for remote management.
9. **Alerting System**:
   * Implement an alerting system to notify authorities or the public about traffic incidents or congestion.
10. **Testing and Simulation**:
    * Simulate the traffic management system to test its effectiveness and identify any issues.
11. **Deployment**:
    * Deploy the system in a real-world environment, ensuring it's robust and reliable.
12. **Maintenance and Monitoring**:
    * Regularly update and monitor the system for any issues or updates.

Here's a simple example of Python code to get you started with traffic data analysis using Pandas:

**PYTHON CODE:**

import pandas as pd

# Load traffic data from a CSV file

traffic\_data = pd.read\_csv('traffic\_data.csv')

# Analyze the data, e.g., calculate average speed

average\_speed = traffic\_data['speed'].mean()

import matplotlib.pyplot as

plt

traffic\_data. Plotx='timestamp', y='speed')

plt.xlabel('Time')

plt.ylabel('Speed')

plt.title('Traffic Speed Over Time')

plt.show()

Remember, the implementation can be much more complex based on the scale and requirements of your project. You may also need to integrate with hardware and external systems for traffic control.

**DIAGRAMS,SCHEMATICS AND SCREENSHOTS OF IOT SENSOR:**

**REQUIREMENTS**

**Hardware Components**

Microcontroller (Raspberry pi Mega 2560):

The Raspberry pi Mega 2560 is a micro- controller board based on the At mega 2560. It has 54 digital input/output pins (of which 15 can be used as PWM outputs), 16 analog inputs, 4 UARTs(hardware serial ports), a 16 MHz crystal oscillator, a USB connection, a power jack, an ICSP header, and a reset button. It contains everything needed to support the microcontroller; simply connect it to a computer with a USB cable or power it with a AC-to-DC adapter or battery to get started. The Mega 2560 board is compatible with most shields designed for the Extension board and the former boards.

**Microcontroller(Raspberrypi)**:

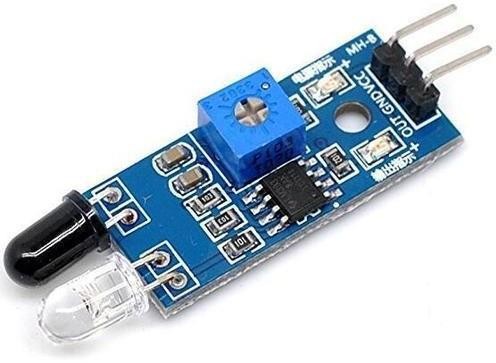
TheRaspberrypiisanopensourcemicrocontrollerboardbasedontheMicrochipATmega328PmicrocontrolleranddevelopedbyArduino.cc.Theboardisequippedwithsetsofdigitalandanaloginput/output(I/O)pinsthatmaybeinterfacedtovariousexpansionboards(shields)andothercircuits.The board has 14 Digital pins, 6 Analog pins, and programmable with the RaspberrypiIDE(IntegratedDevelopmentEnvironment)viaatypeBUSBcable. 

**LEDs**:

LEDsareusedforthepurposeofsignalingaccordingtothetrafficcondition.

LED for Traffic Lights.

**IR Sensor**:

IRS ensures used to count they vehicles on the road. from the Wiring project,which provides many common input and output procedures

IR Sensors.

**Jumper Wires**:

It is used to connect the components to each other.

### REQUIREMENTS

### Software Requirement

### Raspberry piIDE:

The Raspberry pi integrated development environment(IDE)is a cross-platform application (for Windows, MacOS, Linux) that is written intheprogramminglanguageJava.ItisusedtowriteanduploadprogramstoRaspberry piboard.

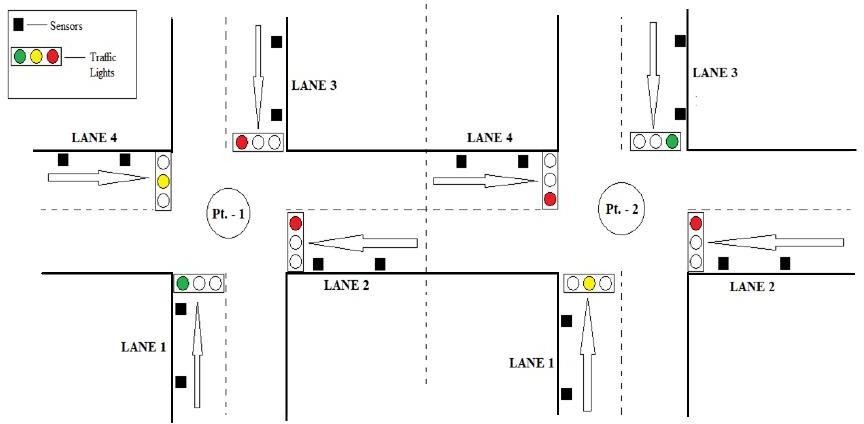
The source code for the IDE is released under the GNU General Public License ,version 2. The Raspberry piIDE supports the languages C and C++ using special rules of code structuring. The Raspberry piIDE supplies a software library

**Proteus Design Suite**:

The Proteus Design Suite is a proprietary software tool suite used primarily for electronic design automation. The software is used mainly by electronic design engineers and technicians to create schematics and electronic prints for manufacturing printed circuit boards.

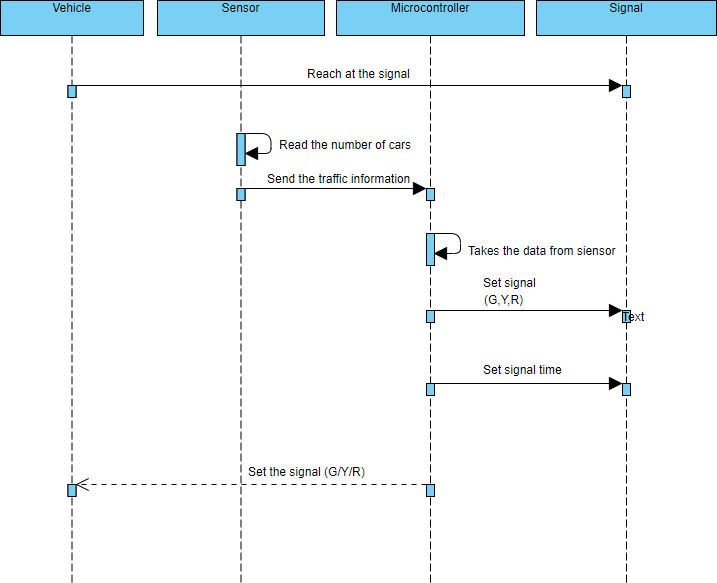
**METHOD:**In this proposed system, the traffic lights are LEDs and the car counting sensor is an ultrasonic sensor. Both blocks are connected to a Microcontroller using physical wires. The Microcontroller is the traffic light controller which receives the collected sensor data and manages the traffic lights by switching between green, yellow and red. The Microcontroller computes the number of cars in the street of the inter-section it is monitoring based on the distances measured by the ultrasonic sensor and the timing between those measurements. The Microcontroller then sends the number of cars every minute to the local server. This communication is done using the Microcontroller serial port. The local server exchanges the data received with the cloud server in order to better predict the changes in timings of the traffic light .This communication is done using Wi-Fi. More specifically, the cloud server uses anequationthattakesthedatareceived(numberofcars)asinputthendeterminesthetime interval of LEDs needed for a smooth traffic flow. This calculated time is thencomparedtothecurrentactualtimeoftheLEDs(thisdataissavedinadatabaseonthe cloud server). The server then comes up with a decision. If the current actual green time is less than the calculated time, the decision is to increase the green time, else to decrease the green time.

### VIEW OF SIGNALS AT DIFFERENT LANES:

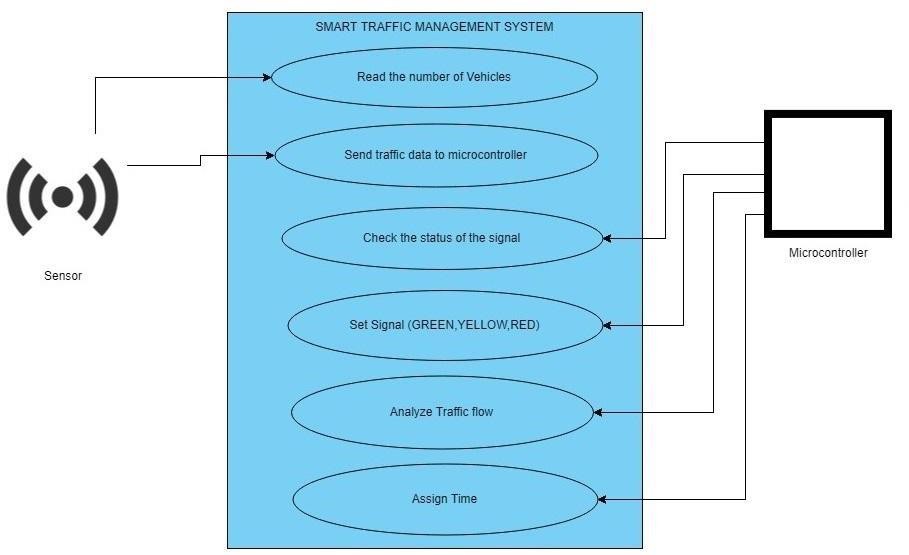
****

**CONTROL OF PREVIOUS INTERSECTION:**

In the above figure, in Pt. - 1, LANE 1 is currently open with green signal and LANE4 is ready with yellow signal but LANE 2 and LANE 3 are blocked. InLANE 3,vehiclecountisalreadygreaterthanthethresholdvalue,thereforetheroadcomingto LANE 2 of Pt. - 1 is blocked in the Pt.- 2 itself. Thus re-routing them through another lanes. (Assuming that Pt. - 1 is the current intersection and Pt.- 2 is the previous intersection.)

**SEQUENCE DIAGRAM:**

**CASE DIAGRAM:**



### COUNTER ALGORITHM:

AssumingtheobjectsdetectedbytheIRSensorstobevehicles,intcounter=0;

inthitObject=false;

int val;

step 1;Read value from sensor.  
**Step2:**Ifval==0hitObject=falsethenincrementthecounterandsethit object

=true.  
elseifval==1hit Object=truethensethitObject=false.

**Step3:**Gotostep1

#### **TRAFFIC CONTROL ALGORITHM**

No.ofsensors=8andaredenotedbyS1,S2,S3,S4,S5,S6,S7,S8No.ofcarsinLane1(N1)=S1–S2

No. of cars in Lane 2 (N2) = S3 – S4No. of cars in Lane 3 (N3) = S5 – S6No.ofcarsinLane4(N4)=S7–S8

Li=(L1,L2,L3,L4),Ni=(N1,N2,N3,N4),Ti=(T1,T2,T3,T4)

**Step1:**Start

**Step2:**Sensorswillreadtheno.ofvehiclesoneachlane(i.e.L1,L2,L3,L4)

**Step3:** if(VehicleCount*<*Threshold)

Then status = Normal traffic. Turn on the green signal for all the lanes one afteranotherinasequentialmanner(L1-L2-L3-L4).Whensignalisgreenforonelane,theotherswillremainred.

**Step4:**elsestatus=congestion.

**Step 5:** COMPARE (N1, N2 , N3, N4), Select the highest of the four (say Ni),turnongreensignalforthatlane(sayLi)fortime(Ti).WhentimeTiends,turnontheredsignal.

**Step 6:** COMPARE (N2, N3, N4), Select the highest of the three (say Ni), turn on green signal for that lane (say Li) for time (Ti). When time Ti ends, turn on the red signal.

**Step7:**COMPARE(N3,N4),Selectthehighestofthetwo(sayNi),turnongreensignal for that lane (say Li) for time (Ti). When time Ti ends, turn on the red signal.

**Step 8:** The last remaining lane automatically gets selected and it is given the green signal for time Ti.

**Step9:**JumptoStep3.

**REAL TIME TRAFFIC MONITORING SYSTEM CAN ASSIST COMMUTERS IN MAKING OPTIMAL ROUTE DECISIONS AND IMPROVING TRAFFIC FLOW:**

**Real-time Data Updates:** These systems provide up-to-the-minute traffic information, including accidents, road closures, and congestion. Commuters can access this data through mobile apps or in-vehicle navigation systems to make informed decisions.

**Route Recommendations:** The system can suggest alternate routes to avoid heavy traffic, accidents, or road closures, helping commuters find the fastest and least congested path to their destination.

**Traffic Predictions:** By analyzing historical and real-time traffic data, the system can predict traffic patterns, allowing commuters to plan their trips during off-peak hours to reduce congestion and travel time.

**Reducing Bottlenecks:** Commuters who receive real-time traffic information can spread out on various routes, reducing congestion on specific roadways and preventing traffic bottlenecks.

**Public Transit Integration:** Some systems also integrate with public transportation options, allowing commuters to switch to buses or trains when traffic is particularly heavy, further reducing road congestion.

**Improved Traffic Flow:** By distributing traffic across multiple routes, the system helps to improve overall traffic flow, reducing stop-and-go conditions and preventing gridlock.

**Emergency Response:** In case of accidents or emergencies, the system can alert emergency services and reroute traffic, facilitating quicker responses and minimizing the impact on overall traffic flow.

**Environmental Impact:** Reduced congestion and more efficient routing can lead to lower fuel consumption and emissions, contributing to a more environmentally friendly transportation system.

**Data-Driven Planning:** Traffic monitoring systems generate valuable data for city planners and transportation authorities, helping them make informed decisions about infrastructure improvements and traffic management.  
In summary, a real-time traffic monitoring system assists commuters in making optimal route decisions by providing current traffic information and route recommendations, ultimately leading to improved traffic flow, reduced travel time, and a more efficient transportation network.

**CONCLUSION:**

Smart Traffic Management System has been developed by using multiple features of hardware components in IOT. Traffic optimization is achieved using IOT platform for efficient utilizing allocating varying time to all traffic signal according to available vehicles count in road path. Smart Traffic Management System is implemented to deal efficiently with problem of congestion and perform re-routing at intersections ona road .This research presents an effective solution for rapid growth of traffic flow particularly in big cities which is increasing day by day and traditional systems have Some limitations as they fail to manage current traffic effectively. Keeping in view the state of the art approach for traffic management systems, a smart traffic man-agreement system is proposed to control road traffic situations more efficiently and effectively. It changes the signal timing intelligently according to traffic density on the particular roadside and regulate  
traffic flow by communicating with local server more effectively than ever before. The decentralized approach makes it optimized and effective as the system works even if a local server or centralized server has crashed. The system also provides useful information to higher authorities that can be used in road planning which helps in optimal usage of resources.

**RESULTS AND ANALYSIS**

The proposed system helps in better time based monitoring and thus has certain advantages over the existing system like minimizing number of accidents, reducing fuel cost and is remotely controllable etc.The proposed system is designed in such a way that it will be able to control the traffic congestion as well as track the number of vehicles. The administrator of the system can access local server in order to maintain the system.