



# SENSOR-DRIVEN TRAFFIC SIGNAL OPTIMIZATION

<sup>1</sup>Anagha Rathkanthiwar, <sup>2</sup>Yashika Chaudhary, <sup>3</sup>Madhav Ambulkar

<sup>4</sup>Anurag Kokate, <sup>5</sup>Sagar Raut

<sup>1</sup>Associate Professor, <sup>2</sup>Student, <sup>3</sup>Student, <sup>4</sup>Student, <sup>5</sup>Student

Department of Electronics and Communication,  
Priyadarshini College Of Engineering, Nagpur, India

## ABSTRACT

This paper introduces an advanced Sensor-Driven Signal Optimization system that enhances traffic management and prioritizes emergency vehicle movement at road intersections. The system integrates two ultrasonic sensors and a microphone (mic) module connected to a Raspberry Pi, which processes real-time data to optimize traffic signal timings. The ultrasonic sensors are installed to measure the distance of vehicles from the signal post, determining traffic density in each direction. Simultaneously, the mic module listens for emergency sirens, such as those from ambulances or fire trucks, to detect the presence of emergency vehicles. Upon identification of an emergency vehicle, the system adjusts the traffic signal to prioritize the emergency vehicle's movement, ensuring minimal delays. The system dynamically adjusts the signal timings based on real-time traffic density and the need for emergency vehicle clearance. This novel approach improves traffic flow efficiency, reduces congestion, and enhances emergency response times by providing a responsive and adaptable signal control mechanism. The combination of ultrasonic sensors for vehicle detection and the mic module for emergency vehicle identification forms the basis for a flexible, efficient, and smart traffic management solution.

## INTRODUCTION

In urban areas, traffic congestion and emergency vehicle delays present significant challenges for city planners and traffic authorities. Traditional traffic signal systems operate on fixed schedules, often failing to account for fluctuating traffic densities and the urgent needs of emergency vehicles. This leads to inefficiencies in traffic flow and can delay emergency responses, posing risks to public safety. This paper addresses these issues by proposing an Sensor-Driven Signal Optimization system that uses ultrasonic sensors and a microphone module to dynamically adjust signal timings based on real-time traffic conditions and emergency vehicle detection.

The system utilizes two ultrasonic sensors to measure vehicle presence and distance, allowing it to estimate traffic density in each lane. By continuously monitoring this data, the system adjusts the signal phases to ensure that traffic flow is optimized. Additionally, a microphone module is employed to detect the sound frequencies associated with emergency vehicles. When an emergency vehicle is identified, the system gives priority to that vehicle by extending the green signal duration in the relevant direction, thus minimizing delays and allowing for quicker emergency response times.

The integration of these novel components—ultrasonic sensors for traffic density measurement and a mic module for emergency vehicle detection—creates a more responsive and efficient traffic management system. This research aims to improve the overall functionality of traffic signals by making them adaptable to both everyday traffic conditions and emergency situations. By offering an innovative solution to traffic congestion and emergency vehicle delays, this system presents a significant advancement in intelligent transportation systems (ITS), contributing to safer and more efficient urban mobility.

## **HARDWARE**

### **1. Raspberry Pi 5 Model B+**

The Raspberry Pi 5 Model B+ is a compact, powerful single-board computer that serves as the brain of the traffic control system. It handles real-time sensor data processing, traffic signal control, and decision-making based on vehicle detection and emergency vehicle recognition. With its powerful CPU, enhanced GPU, and generous RAM, it is capable of multitasking efficiently and operating several GPIO-based hardware interfaces simultaneously. Its GPIO pins are used to directly control LED-based traffic lights and receive inputs from the ultrasonic sensors and the microphone module, making it the central hub for managing the entire traffic management operation.

### **2. Ultrasonic Sensors (HC-SR04 or equivalent)**

The ultrasonic sensors are responsible for real-time vehicle detection at the intersection. Two sensors are installed facing different directions to measure the distance between vehicles and the traffic signal post. They work by emitting ultrasonic waves and measuring the time taken for the echo to return after hitting an obstacle (vehicle). These sensors allow the system to estimate the presence or absence of vehicles in each lane. By continuously monitoring vehicle density, they enable the dynamic adjustment of traffic light timings, thereby reducing unnecessary waiting times and optimizing traffic flow.

### **3. Microphone Sound Sensor Module**

The microphone module is used to detect emergency vehicle sirens in real time. Installed near the intersection, it constantly listens to the ambient sound environment. When a siren (characterized by a high-pitched sound) is detected, the mic module sends a logic signal to the Raspberry Pi. Upon detection, the system immediately prioritizes the green signal for the direction of the emergency, allowing ambulances, fire trucks, or police vehicles to pass through without delay. This ensures faster clearance for emergency services and minimizes their waiting time at traffic lights.

### **4. Traffic Light Module / LED Signal System**

LED-based traffic light modules simulate the functioning of real-world traffic signals. Each module consists of three LEDs (Red, Yellow, and Green) which are controlled through the Raspberry Pi's GPIO pins. These lights switch dynamically according to the traffic density detected by the ultrasonic sensors and emergency signals detected by the microphone. Real-time control of these LEDs enables the system to manage traffic flow intelligently, minimize idle times, and respond quickly to varying traffic and emergency situations.

### **5. MicroSD Card (32GB or Higher)**

The MicroSD card is used as the primary storage medium for the Raspberry Pi. It contains the Raspberry Pi OS, Python scripts, sensor libraries, and configuration files necessary to run the traffic control system. A high-speed and high-capacity card is essential to ensure smooth operation, faster boot times, and reliable performance of the real-time application, especially when processing continuous sensor inputs without any delay.

## **SOFTWARE**

### **1. Raspberry Pi OS (Operating System)**

Raspberry Pi OS (formerly known as Raspbian) is a lightweight Debian-based operating system that provides a stable platform to run Python scripts and manage hardware interfaces. It supports the necessary libraries for

GPIO control, sensor interfacing, and remote access. Raspberry Pi OS ensures efficient and stable performance during the continuous operation of the traffic management system.

## 2. Python Programming Language

Python serves as the core programming language for developing the logic and control algorithms of the system. Its simplicity, vast library support, and seamless GPIO interfacing capabilities make it ideal for real-time traffic management applications. Python scripts coordinate the reading of sensor data, process vehicle and siren detections, and dynamically control the traffic signal LEDs based on real-time inputs.

## 3. RPi.GPIO Library

The RPi.GPIO library is used in Python to control the Raspberry Pi's GPIO pins. It enables the Raspberry Pi to read inputs from the microphone and ultrasonic sensors and to control outputs such as turning LEDs ON and OFF. The library provides an easy-to-use interface for managing multiple GPIO operations simultaneously, which is crucial for real-time traffic signal control.

## 4. gpiozero Library

The gpiozero library simplifies interaction with components like ultrasonic sensors. It provides high-level interfaces that allow the system to easily measure distances and react to changing traffic conditions. The DistanceSensor class from gpiozero is used to monitor the distance between the sensor and vehicles continuously, which is essential for making accurate traffic light timing decisions.

## 5. Time Library

The built-in time library in Python is used for introducing delays between operations, such as controlling the duration of green, yellow, and red signals. It ensures that the traffic lights remain ON for the required time intervals and helps synchronize the switching of traffic signals according to real-time traffic density and emergency vehicle detection.

## BLOCK DIAGRAM

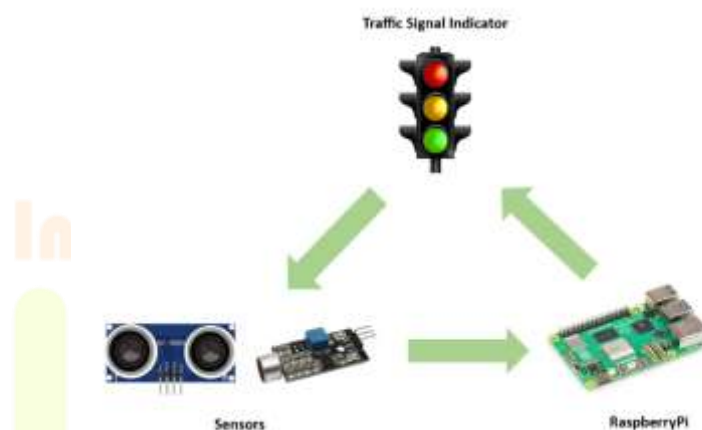


Fig.1 The system setup is illustrated through the block diagram, where sensors detect vehicles and emergency sirens, sending inputs to a Raspberry Pi. The Raspberry Pi processes this data and adjusts the traffic signal timings to improve traffic flow and prioritize emergency vehicles

## METHODOLOGY

### A. Existing System

Most traditional traffic management systems still rely on fixed-time traffic signals or manual control by traffic police officers. These systems are designed with static timings that do not adapt according to real-time traffic conditions, leading to inefficiencies such as unnecessary waiting, traffic congestion, and increased travel time. Moreover, during emergencies, such as ambulances or fire trucks approaching an intersection, these systems lack the ability to prioritize the emergency vehicles automatically, further delaying critical response times.

### B. Issues in Existing System

The major limitation of the existing traffic systems is their **inability to adjust traffic signals dynamically** based on real-time traffic density or emergency needs.

In fixed-timer systems, signals continue to operate regardless of whether a lane is crowded or empty, causing unnecessary delays and increasing fuel consumption.

Additionally, the absence of emergency detection mechanisms results in emergency vehicles getting stuck in traffic, delaying critical medical or safety responses. Manual monitoring is not only inefficient but also impractical in managing traffic in large, high-density urban areas, where quick real-time decision-making is crucial.

### C. Proposed Methodology

To overcome these challenges, the proposed system introduces a **Sensor-Driven Signal Optimization** solution using **ultrasonic sensors** and a **mic sound sensor module**. Two ultrasonic sensors are placed at strategic positions to continuously monitor vehicle presence and traffic density in two different directions at the intersection. These sensors measure the distance of approaching vehicles and send real-time data to a **Raspberry Pi**, which processes the information to determine traffic flow.

In addition to traffic density detection, a mic sound sensor is integrated into the system to detect emergency vehicle sirens.

The mic sensor continuously monitors ambient sound levels, and upon detecting a siren sound, the system triggers an emergency response by prioritizing the lane where the siren is detected.

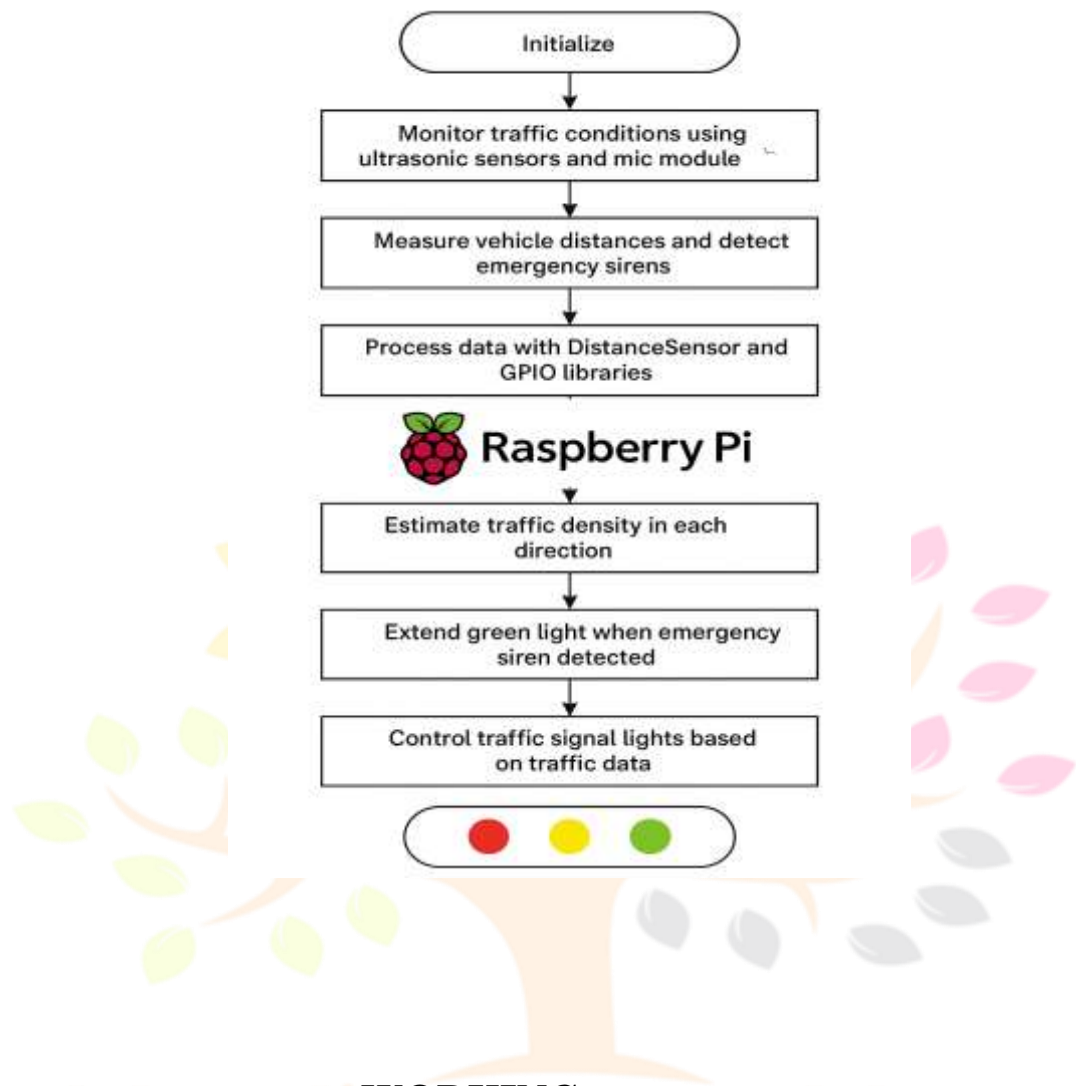
The Raspberry Pi acts as the processing and control unit. It uses Python-based logic to dynamically adjust traffic light durations based on the number of detected vehicles or the presence of an emergency vehicle. If normal traffic is detected, the system switches between lanes based on density and preset timers.

However, if an emergency siren is detected, the system overrides the normal sequence, extending the green light duration for the emergency lane to allow quick and uninterrupted passage.

This adaptive approach ensures better traffic management by reducing congestion, minimizing idle time, and providing an efficient mechanism for emergency vehicle prioritization, ultimately enhancing road safety and mobility.



## FLOWCHART



## WORKING

The traffic flow management system operates using two ultrasonic sensors and a mic module connected to a Raspberry Pi, continuously monitoring traffic at an intersection. The ultrasonic sensors face each incoming lane and measure real-time vehicle distances, while the mic module detects ambient sounds for emergency sirens.

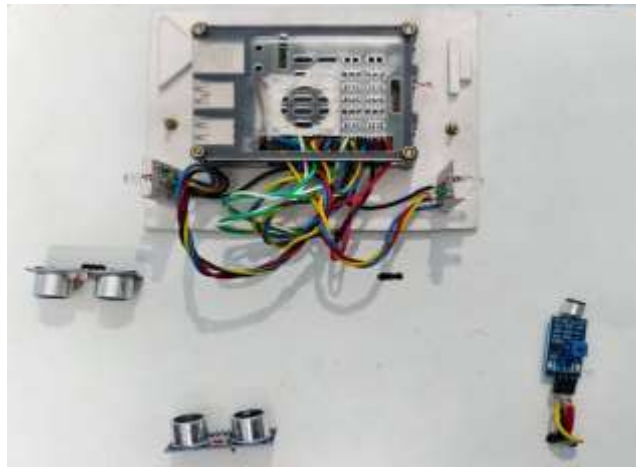
The Raspberry Pi, running a Python program with DistanceSensor and GPIO libraries, processes sensor data. When a vehicle is detected within a threshold distance (less than 6 cm), the system estimates traffic density for each direction. Simultaneously, the mic module monitors for emergency sirens; upon detection (logic low signal), the system prioritizes that lane by extending the green light duration.

Based on vehicle counts and siren detection, the Raspberry Pi dynamically controls LED-based traffic signals. Under normal conditions, green lights are allocated based on traffic density, switching every 10 seconds with a brief yellow "Wait" phase. In emergencies, the green light is immediately extended up to 40 seconds for the relevant direction.

By constantly adapting to real-time traffic and emergency conditions, the system ensures smoother traffic flow, reduces congestion, and enables faster clearance for emergency vehicles at intersections.

## RESULT

The developed traffic flow management system was successfully implemented using two ultrasonic sensors, a mic module, and a Raspberry Pi. The system accurately detected the presence of vehicles by continuously measuring distances in each approaching lane. Whenever a vehicle was detected within the 6 cm threshold, the corresponding traffic density was calculated in real-time. The mic module consistently monitored ambient sound levels and effectively identified emergency sirens by detecting a logic low signal.



**Fig.2 Sensor-Driven Traffic Signal Optimization System**

```

28 direction=0
29 state1=0
30 state2=0
31 dist1=0
32 dist2=0
33 event1=0
34 extendedTime=0
35
36 def aux1():
37     event1=0
38     print ("DIR-1 ON",count1,state1,state2)
39     GPIO.output(9,GPIO.HIGH)#G1
40     GPIO.output(10,GPIO.LOW)#G2
41     GPIO.output(24,GPIO.LOW)#G3

```

**Fig.3 The Python program on the Raspberry Pi successfully activated the correct traffic direction control (DIR-1 and DIR-2) based on the sensor inputs, with real-time vehicle count and emergency detection data printed in the Shell window, confirming proper switching and logic operation.**

The Raspberry Pi processed the sensor inputs and dynamically controlled the LED-based traffic signals based on current conditions. Under normal traffic flow, the green light duration was assigned fairly according to the vehicle density detected in each lane, maintaining smooth transitions with approximately 10 seconds of green signal followed by a yellow "Wait" phase for safe switching. In the case of an emergency siren detection, the system promptly extended the green signal time up to 40 seconds for the direction of the emergency vehicle, allowing quick and unobstructed passage through the intersection.

Throughout testing under simulated traffic conditions, the system consistently adapted to changes in vehicle density and emergency occurrences without manual intervention. The sensors showed stable performance with minimal false readings, and the mic module reliably distinguished emergency sirens from regular ambient noise. As a result, the system effectively reduced waiting time at signals, improved overall traffic flow, and demonstrated its ability to prioritize emergency vehicles, fulfilling all intended objectives of the project.

## CONCLUSION

The designed traffic flow management system based on Raspberry Pi, ultrasonic sensors, and a mic module proves to be an efficient solution for intelligent traffic control. By adapting signal timings dynamically according to vehicle density and emergency requirements, the system improves road traffic conditions and minimizes waiting time at intersections. The integration of sound-based emergency detection enhances the responsiveness of the system toward critical situations, allowing emergency vehicles to clear intersections faster. Thus, the project demonstrates the potential of simple sensor-based automation in developing smart, real-time traffic management solutions that are scalable and practical for urban applications.

## FUTURE SCOPE

The traffic flow management system developed in this project can be further enhanced by integrating more advanced technologies. In the future, the use of machine learning algorithms can enable more accurate prediction of traffic patterns and dynamic signal optimization based on historical data. High-resolution cameras and computer vision techniques could be added to detect vehicle types and lane-specific congestion for even finer control.

Additionally, connecting the system to a central traffic monitoring network would allow real-time updates and remote management across multiple intersections.

Integration with IoT platforms and mobile applications could provide drivers with real-time traffic signal timings and suggest alternative routes to reduce congestion.

Furthermore, by using GPS-based emergency vehicle tracking instead of only audio detection, the system can respond even faster and more precisely to emergency situations. Solar-powered Raspberry Pi setups could also be explored to make the system energy-efficient and environmentally friendly. Overall, these enhancements would contribute to building a fully automated, smart traffic ecosystem suitable for modern smart cities.

## REFERENCES

- [1] A. Bhosale and S. Pathan, "Smart Traffic Light Control System using Arduino and Ultrasonic Sensors," *International Journal of Engineering Research & Technology (IJERT)*, vol. 10, no. 4, 2021.
- [2] P. Rathod and D. Kalbande, "Arduino Based Adaptive Traffic Control System," *International Journal of Computer Applications*, vol. 175, no. 28, pp. 6–9, 2020.
- [3] P. Raut, R. Chandak, and M. Khedkar, "Smart Traffic Signal Using Ultrasonic Sensor," 2013.
- [4] M. Riyazhussain, P. Maheshwari, and S. Tiwari, "Raspberry Pi Controlled Traffic Density Monitoring System," *Semantic Scholar*, 2022.
- [5] R. Khan and S. Raza, "A Review on Smart Traffic Management System using Sensors and Image Processing," *Journal of Traffic and Transportation Engineering*, vol. 6, no. 4, 2019.