IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⩫ Peer-reviewed & Refereed journal ⩫ Vol. 13, Issue 8, August 2025

DOI: 10.17148/IJIREEICE.2025.13806

IoT BASED VEHICLE EMISSION MONITORING SYSTEM

Ali I K[1], Anjali N R[2], Anusha V S [3], Bhagyashree V G [4], Spoorthi T S[5]

Assistant Professor, E&CE, BIET, Davangere, Karnataka India[1]

Student, UG, E&CE, BIET, Davangere, Karnataka India[2-5]

Abstract: The rapid increase in vehicular traffic has led to significant environmental concerns, particularly due to the emission of harmful gases such as carbon monoxide (CO), nitrogen oxides (NOx), and hydrocarbons (HC). Traditional methods of emission monitoring are often periodic and fail to provide real-time data, limiting their effectiveness in controlling pollution. This project presents an IoT-based Vehicle Emission Monitoring System designed to continuously track and report emission levels from individual vehicles. Using gas sensors integrated with a microcontroller and IoT module, the system detects pollutant levels and transmits the data to a cloud platform for real-time analysis and storage. Users and authorities can access this information via a web or mobile application, enabling immediate identification of vehicles exceeding emission norms. The proposed system aims to promote environmental awareness, support regulatory compliance, and contribute to cleaner air by facilitating timely maintenance and stricter enforcement of emission standards.

Keywords: harmful gases, individual vehicles, emission standards

I. INTRODUCTION

Air pollution, particularly in urban areas, is a major global concern, with vehicular emissions being a major contributor. Traditional emission checks often miss real-time pollution spikes, making continuous monitoring essential. This project uses the Internet of Things (IoT) to develop a real-time emission monitoring and alert system. The system tracks exhaust gas levels, issues instant alerts when thresholds are exceeded, and stores data for analysis and regulatory use. The system integrates an ESP32 microcontroller for Wi-Fi and processing, an MQ-7 sensor for CO detection, and an MQ-135 sensor for general air quality monitoring. Alerts are sent via the Blynk mobile app, while emission data and user complaints are managed through a PHP-MySQL web dashboard with a Bootstrap front end.

II. OBJECTIVES

The objectives are: To design and develop a functional prototype for vehicle emission monitoring system.
To update the real time data of the emission by vehicle through the sensor using IoT.

III. METHODOLOGY

The methodology adopted for this project integrates hardware design, firmware development, cloud platform utilization, and web application development to create a comprehensive emission monitoring and alert system. The approach can be broken down into distinct phases, focusing on component selection, system architecture design, software implementation, and testing.

A. BLOCK DIAGRAM

The overall architecture of the "IoT-Based Vehicle Emission Monitoring System" is depicted in Figure 1. It illustrates the utilization of a cloud platform and the development of a web application to create a comprehensive emission monitoring and alert system. The approach can be broken down into distinct phases, focusing on component selection, system architecture design, software implementation, and testing.

The system starts with the sensing unit near the vehicle's exhaust, where the MQ-7 detects CO levels and the MQ-135 monitors overall air quality. These analog readings are sent to the ESP32 microcontroller, which processes the data into meaningful values like CO in PPM. Using its built-in Wi-Fi, the ESP32 connects to the internet to perform two main communication tasks.

DOI: 10.17148/IJIREEICE.2025.13806

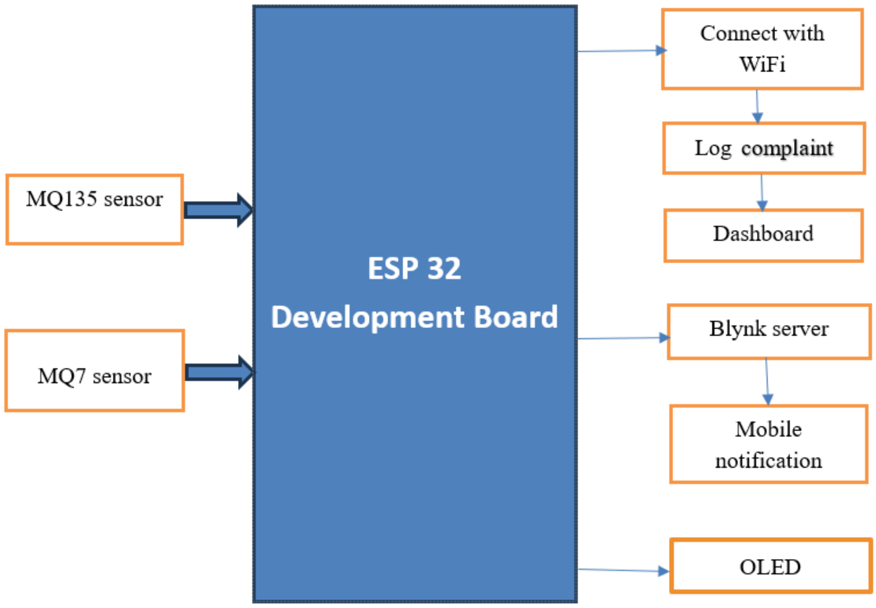

Figure 1 Block diagram

1. It sends processed sensor data to the Blynk Cloud Server. If the CO level exceeds a predefined threshold, the Blynk server triggers an alert notification, which is instantly pushed to the Blynk Mobile Application on the user's smartphone. The app also displays real-time sensor readings.
2. The ESP32 sends sensor data to a web server running PHP scripts, which process and store the data in a MySQL database along with timestamps and vehicle IDs. The server also hosts a web dashboard where users can view emission data, analyze trends, and submit complaints about polluting vehicles. These complaints are also handled by PHP scripts and stored in the database.

This architecture ensures real-time data acquisition, immediate alerting for critical emission events, and long-term data storage and visualization for comprehensive monitoring and analysis.

B. CIRCUIT DIAGRAM

The circuit diagram shows how all components are connected, with the ESP32 as the central controller. MQ-7 and MQ-135 sensors connect their VCC to 5V and GND to ground, with their analog outputs linked to the ESP32's ADC pins. The power supply provides 5V for sensors and either 5V to VIN or 3.3V directly to the ESP32. Any necessary pull-up or pull-down resistors are included, though many are built into the sensor modules. The use of standard modules simplifies the design, and the diagram clearly illustrates all power and signal connections for easy hardware setup.

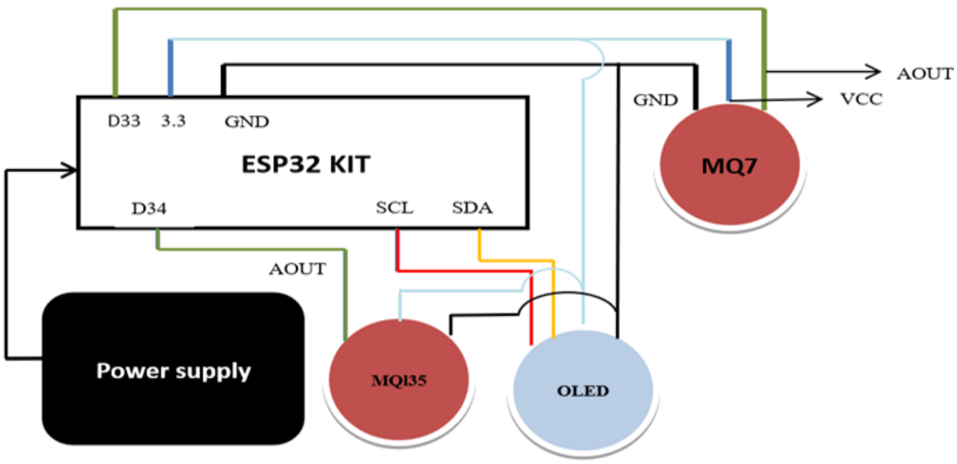

Figure 2 Circuit Diagram

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2025
DOI: 10.17148/IJIREEICE.2025.13806

IV. FLOW CHART

The operational logic of the system, particularly the firmware on the ESP32 and the core web interactions, can be represented as follows.

1. ESP32 Firmware Main Loop Flow:
The program initializes the ESP32, sets up Wi-Fi, Blynk, and sensor warm-up. After connecting to Wi-Fi and Blynk, it enters a loop where it reads data from MQ-7 and MQ-135 sensors, converts it into CO and air quality values, and sends it to the Blynk app and a web server in JSON format. It includes error handling and triggers alerts if CO exceeds safe levels, repeating the process every few seconds for real-time monitoring.

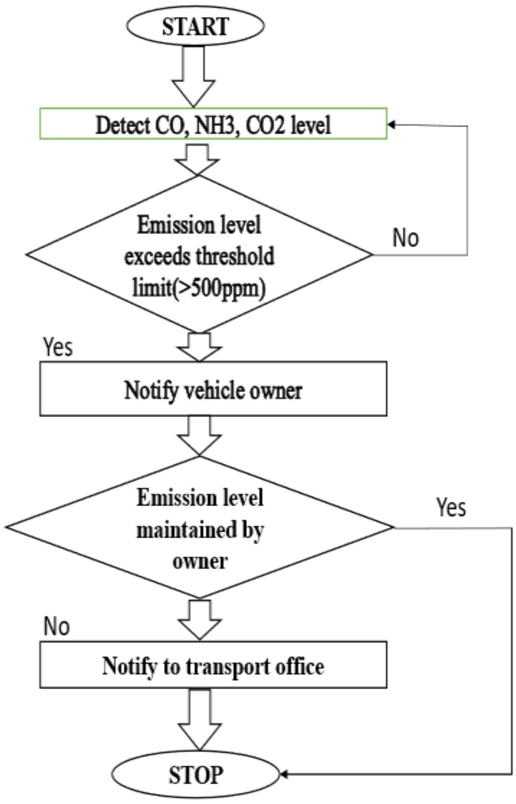

Figure 3 Flow chart

2. Web Server Data Logging (Simplified PHP Script Logic):
The process starts with a PHP script triggered by an HTTP request from the ESP32, sending data like CO PPM, AQI, timestamp, and device ID. The script validates and sanitizes the data, then connects to the MySQL database. If the connection succeeds, it inserts the data into the emissions_log table. A success or error response is sent based on the result, and the database connection is closed to complete the process.
These flowcharts outline the core processes ensuring continuous monitoring, data transmission, alerting, and logging within the system.

V. HARDWARE DESCRIPTION

The hardware components form the backbone of the data acquisition and initial processing for the "IoT Based Vehicle Emission Monitoring and Alert System." This chapter details the individual components, their interfacing, and the overall circuit design.

1. *ESP32 Microcontroller*
The ESP32 is a powerful, low-cost, low-power system-on-a-chip (SoC) microcontroller with integrated Wi-Fi and dual-mode Bluetooth capabilities. It is developed by Espressif Systems and is widely used in IoT projects due to its versatility and extensive community support.

ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2025
DOI: 10.17148/IJIREEICE.2025.13806

The ESP32 serves as the central processing unit. It reads analog data from the MQ-7 and MQ-135 sensors via its ADC channels, processes this data to determine gas concentrations, connects to the internet via Wi-Fi to send data to the Blynk cloud and the custom web server, and manages the overall operation of the onboard sensing unit.

Figure 4. ESP8266 WiFi MODULE

2. MQ-7 Gas Sensor (Carbon Monoxide Sensor)

The MQ-7 is a gas sensor designed to detect Carbon Monoxide (CO) using a Tin Dioxide (SnO2) sensing element. It operates in high-low temperature cycles to clean and detect CO effectively. In this system, it monitors CO levels in vehicle exhaust, sending analog output to the ESP32 for calibration and alert generation if levels are hazardous.

Figure 5 MQ7 sensor

3. MQ-135 Gas Sensor (Air Quality Sensor)

The MQ-135 is a semiconductor gas sensor primarily used for general air quality monitoring. It is sensitive to a range of gases. The MQ-135 complements the MQ-7 by providing a general indication of the air quality around the exhaust, potentially indicating the presence of other pollutants besides CO. Its reading contributes to a broader assessment of the vehicle's emission profile.

Figure 6 MQ 135 sensor

4. OLED

OLED displays are energy-efficient, high-contrast modules that emit their own light, eliminating the need for a backlight. Known for thin design and deep blacks, they are ideal for embedded systems to display text, graphics, and sensor data. Easily integrated with microcontrollers like the ESP32 using I2C or SPI protocols, they are popular in compact electronic projects.

ISSN (O) 2321-2004, ISSN (P) 2321-5526
International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering
Impact Factor 8.414 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 13, Issue 8, August 2025
DOI: 10.17148/IJIREEICE.2025.13806

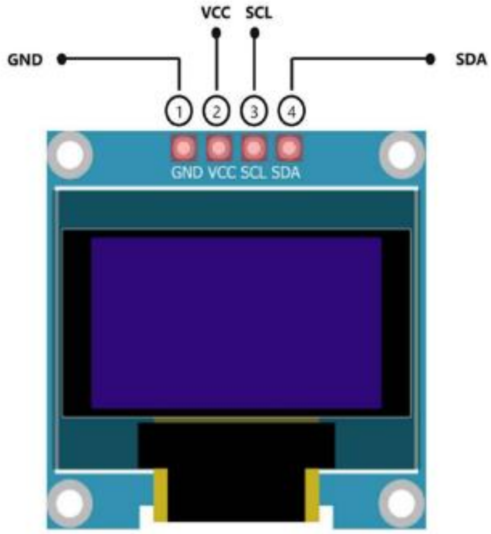

Figure 7 OLED

VI. SOFTWARE DESCRIPTION

The software components of the "IoT Based Vehicle Emission Monitoring and Alert System" are integral to its functionality, enabling data acquisition, processing, communication, user alerts, and data management. The software ecosystem comprises firmware for the ESP32 microcontroller and a web application for the dashboard.

6.1 ESP32 Firmware (developed using Arduino IDE):
This project uses the Arduino IDE with ESP32 board support for C/C++ programming. The ESP32 reads analog data from MQ-7 and MQ-135 sensors, processes it with calibration factors, and connects to the internet using the WiFi.h library. The BlynkSimpleEsp32.h library enables communication with the Blynk app for data display and CO alerts. The HTTPClient.h library is used to send data to a web server via a PHP script. The main loop() function handles sensor reading, data processing, communication, and alert logic in a continuous cycle.

6.2 Blynk Mobile Application
Blynk is an IoT platform used to create mobile interfaces for monitoring hardware projects. In this setup, a project is created in the Blynk app with an auth token linked to the ESP32. Widgets like gauges, notifications, and optional charts display real-time CO and air quality data. It allows users to view emissions and receive alerts directly on their smartphones.

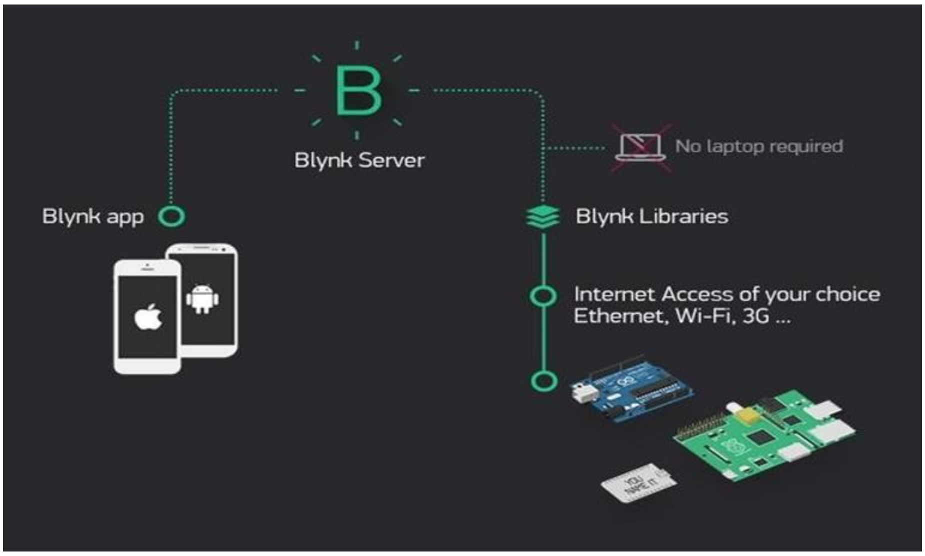

Figure 8 Figure 8 Blynk IoT

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⪤ Peer-reviewed & Refereed journal ⪤ Vol. 13, Issue 8, August 2025

DOI: 10.17148/IJIREEICE.2025.13806

6.3 Web Application (Backend and Frontend)

1. Backend (PHP & MySQL): PHP (Hypertext Preprocessor) - A server-side scripting language used for Three PHP scripts manage interactions with a MySQL database for environmental monitoring. data_logger.php stores sanitized sensor data from the ESP32, submit_complaint.php handles and logs user complaints from the dashboard, and dashboard_data.php retrieves data for display, often as JSON. The MySQL database stores emission data in an emission_logs table and complaints in a complaints table.

2. Frontend (HTML, CSS, Bootstrap, JavaScript) - HTML provides structure, CSS styles appearance, Bootstrap offers pre-built components for responsive layouts. JavaScript enables form validation, AJAX requests to update pages without reloading, and interactive chart creation. These technologies are essential for creating visually appealing and functional web dashboards.

Web Server: Software like Apache or Nginx running on a hosting platform (or XAMPP/WAMP for local development) to serve the PHP pages and manage requests.

VII. RESULTS

This chapter presents the expected and observed results from the testing and validation of the "IoT Based Vehicle Emission Monitoring and Alert System." The system was tested module-wise and then as an integrated unit to verify its functionality against the project objectives.

Figure 9 Demonstration

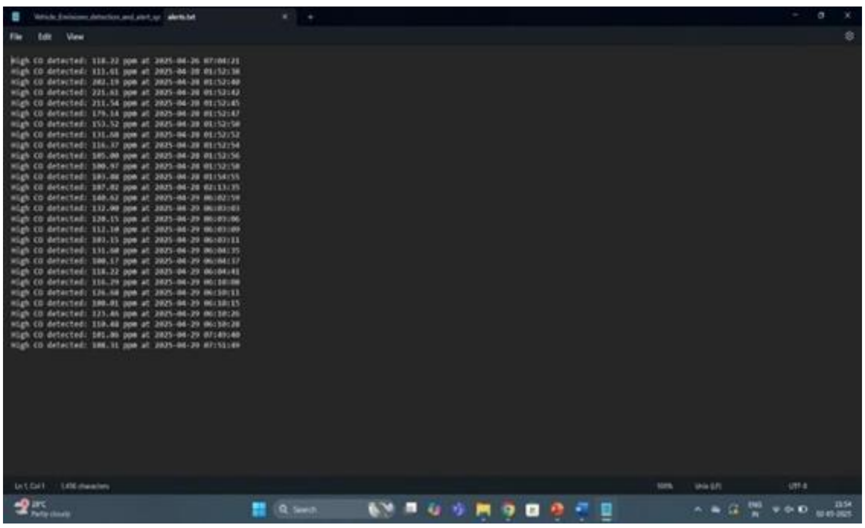

Figure 10 Data stored in WampServer

IJIREEICE ISSN (O) 2321-2004, ISSN (P) 2321-5526

International Journal of Innovative Research in Electrical, Electronics, Instrumentation and Control Engineering

Impact Factor 8.414 ⁒ Peer-reviewed & Refereed journal ⁒ Vol. 13, Issue 8, August 2025

DOI: 10.17148/IJIREEICE.2025.13806

Figures demonstrate the working of the IoT-based vehicle emission monitoring system. Figure 9 shows the live setup where emission data is collected using sensors and the ESP32 board. Figure 10 displays the emission data successfully stored and viewed on WampServer.

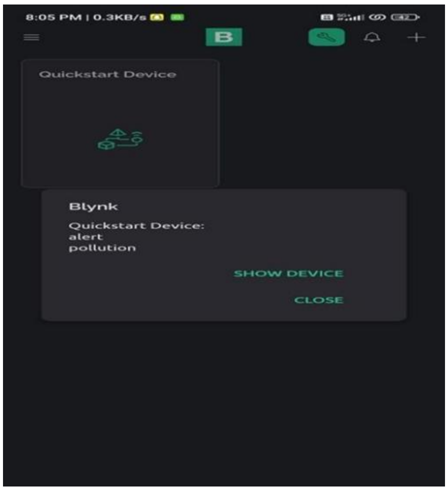

Figure 11 Alert message sending to the vehicle owner through blynk app

Figure 11 shows an alert message sent to the vehicle owner via the Blynk app when the emission exceeds the threshold level. Figure 12 displays the real-time sensor readings and output on the serial monitor, confirming accurate data transmission.

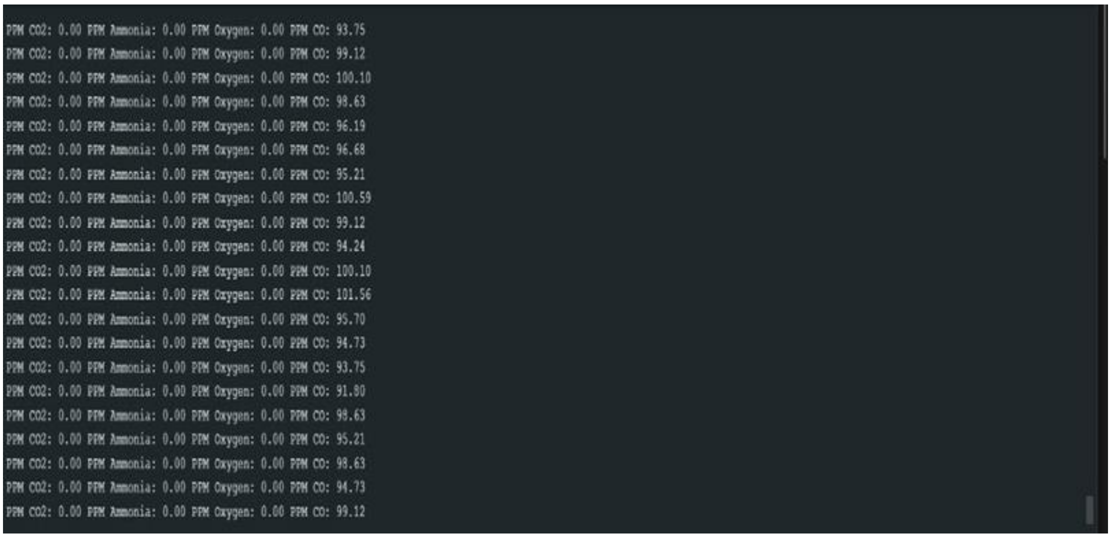

Figure 12 Shows the output in serial monitor

Figure 13 Emission monitoring dashboard

DOI: 10.17148/IJIREEICE.2025.13806

Figure 13 shows the emission monitoring dashboard used to display real-time vehicle data. It includes details such as date, vehicle number, and corresponding CO emission levels. This interface helps users track and review emission records efficiently through a web-based platform.

Result comparison:
The results show that the prototype system satisfies the essential functional requirements. The MQ sensors provide enough responsiveness for a proof-of-concept alert system, although not being analytical tools of professional caliber. The ESP32 demonstrated its ability to manage dual-channel communication (Blynk and HTTP), computing, and sensing. While the web dashboard offered a way to analyze and manage data over a longer period of time, the Blynk app offered an efficient and instantaneous user experience for notifications. More thorough sensor calibration against approved instruments would be necessary for further improvement in order to increase PPM reporting accuracy.

Vehicle No.	CO Tested in ETC (in ppm)	CO Tested in Project model (in ppm)
KA17EZ6126	2300	510
KA17HD4527	7040	520

VIII. APPLICATIONS

1. Individual Vehicle Monitoring: Car owners can install the system to get real-time feedback on their vehicle's emissions, encouraging timely maintenance if pollution levels rise, potentially avoiding fines and contributing to cleaner air.
2. Fleet Management: Transport companies (e.g., logistics, public transport, taxi services) can equip their fleet with this system to: Monitor emission compliance across all vehicles. Identify vehicles requiring urgent maintenance. Optimize routes or driving habits to reduce emissions. Demonstrate environmental responsibility.
3. Regulatory Enforcement Support: Government environmental agencies and transport authorities can utilize data from such systems (especially if made mandatory or subsidized) to: Identify highly polluting vehicles on a continuous basis, complementing periodic checks. Target enforcement actions more effectively. Gather data for policy-making and urban planning.

IX. ADVANTAGES

1. Real-Time Continuous Monitoring: Unlike periodic checks, the system provides continuous data on vehicle emissions during actual operation, offering a more accurate and dynamic understanding of a vehicle's pollution profile.
2. Immediate Alerting: The integration with Blynk ensures instant notifications to users or designated authorities when emission levels exceed predefined thresholds, enabling prompt corrective action.
3. Proactive Maintenance: By alerting vehicle owners to rising emission levels, the system encourages proactive maintenance, potentially preventing more severe engine problems and reducing long-term pollution.

X. LIMITATIONS

1. Sensor Accuracy and Calibration: MQ-series sensors are low-cost and provide good qualitative indications but may lack the precision and long-term stability of professional-grade analytical instruments. Their accuracy can be affected by ambient temperature, humidity, and cross-sensitivity to other gases. Frequent and proper calibration is essential for reliable quantitative measurements, which can be challenging to implement for widespread, low-maintenance deployment.
2. Dependence on Wi-Fi Connectivity: Real-time data transmission to Blynk and the web server relies on a stable Wi-Fi connection. In areas with poor or no Wi-Fi coverage, data logging and alerts would be delayed or missed unless onboard data buffering and store- and-forward mechanisms are implemented.
3. Security and Privacy: Data transmitted over Wi-Fi and stored on the web server could be vulnerable to security breaches if not adequately protected with encryption and secure protocols. Privacy concerns regarding the collection and use of vehicle emission data and potentially location data (if GPS is added) need to be addressed.

DOI: 10.17148/IJIREEICE.2025.13806

XI. FUTURE SCOPE

The project can be improved further by implementing fallowing details

1. It can be improved further by adding more sensors to existing system like dust particles sensors and etc. Interface GPS module to screen the contamination at precise area and transfer on the website page for the netizens.
2. Integration of Advanced Sensors: Incorporate more accurate, selective, and durable sensors for specific pollutants like NOx, SOx, PM2.5, and Hydrocarbons (HC). This could involve exploring electrochemical sensors or NDIR (Non-Dispersive Infrared) sensors for CO2.
3. GPS Module Integration: Add a GPS module to enable real-time location tagging of emission data. This would allow for spatio-temporal pollution mapping, identification of pollution hotspots, and tracking of offending vehicles.

XII. CONCLUSION

This project has been designed and demonstrated using ESP32, MQ-7, and MQ-135 sensors to capture real-time emission data. The system transmits data wirelessly, sends mobile alerts when CO thresholds are exceeded, and logs data on a web dashboard for and complaint management. The prototype validates the concept, providing immediate feedback on hazardous emission levels and empowering individuals and authorities with actionable insights to promote sustainable transportation practices.

REFERENCES

[1]. Chen, H.," Energy-Efficient Data Transmission Protocols for Vehicular IoT Sensor Networks", IEEE Internet of Things Journal, 11(W),(2024)
[2]. Ahmed, S.," Cloud-Integrated Smart System for Vehicular Air Pollution Detection and Geolocation Tagging", In Proceedings of the IEEE International Conference on Smart Computing (SMARTCOMP). City, Country (2023).
[3]. Zhang, Q., & Liu, F, "A Secure and Privacy-Preserving Framework for IoT-Based Vehicular Emission Data Collection", in Proceedings of the IEEE Conference on Communications and Network Security (CNS) (pp. EE-FF). City, Country. (2023)
[4]. Kumar, P., & Singh, A.," An IoT-Enabled Real-Time Vehicle Emission Monitoring System Using Low-Cost Electrochemical Sensors", IEEE Sensors Journal, 22(X), (2022).
[5]. Lee, J., & Kim, Y," Development of a Machine Learning Model for Predicting Vehicle Emission Levels based on IoT Sensor Data", IEEE Transactions on Intelligent Transportation Systems, 22(Z) (2021).