

Design and Development of an Onboard Emissions Detection System

*A report submitted
in partial fulfillment of the requirements
for the degree of*

Bachelor of Technology

In

Mechanical Engineering

by

1. Arjun Taneja

2. Shashank Sharma

R113221003

R113221013



Mechanical Cluster

School of Advanced Engineering (SoAE)

UPES, Dehradun, India-248007

May 2025



Department of Mechanical Engineering

UPES

Certificate

It is certified that the work contained in the project titled “**Design and Development of an Onboard Emissions Detection System**” by following students has been carried out under my/our supervision and that this work has not been submitted elsewhere for a degree.

<i>Student Name</i>	<i>Roll Number</i>	<i>Signature</i>
1. Arjun Taneja	R113221003	
2. Shashank Sharma	R113221013	

Signature
Dr. Prashant Shukla
(Project Supervisor)
Mechanical Cluster
School of Engineering
Dehradun, Uttarakhand
India-248007

Signature
Dr. Ram Kunwer
(Activity Coordinator)
Mechanical Cluster
School of Engineering
Dehradun, Uttarakhand
India- 248007

Abstract

The sustainability of the environment and public health are seriously threatened by the growing problems with vehicle emissions in metropolitan areas. The development of sophisticated, continuous monitoring solutions is required since traditional emission detection systems, which frequently depend on recurring inspections, are insufficient in reducing real-time breaches. The novel Onboard Emissions Detection System (OEDS), a comprehensive system intended to track, analyze, and report vehicle emissions in real-time, is presented in this paper. The system seeks to improve adherence to regulations and support international initiatives for sustainable mobility.

An integrated assembly of high-precision gas sensors, positioned to capture exhaust gas compositions, is the basis of the proposed OEDS. The microcontroller is programmed with adaptive algorithms capable of analyzing sensor data and identifying patterns indicative of threshold breaches. This system not only ensures high detection accuracy but also minimizes false positives through dynamic calibration techniques.

One of the distinctive characteristics of the OEDS is its ability to wirelessly transmit real-time flagged emission data to Regional Transport Office (RTO) computers. The Internet of Things (IoT) architecture and secure communication protocols are used by the system to ensure seamless data flow and robust system interoperability. An emission violation triggers the creation of alerts that are immediately forwarded to regulatory agencies, enabling prompt action. The addition of geographic tagging provides precise location data for emissions hotspots, which enhances the system's use and enables targeted enforcement strategies.

The OEDS's design architecture prioritizes scalability and adaptability to ensure that it may be used to a wide range of vehicle types and regulatory environments. Widespread acceptance is encouraged by the hardware components' cost and energy efficiency

optimization, which includes the sensor assembly and microprocessor. In order to prevent any infractions, the system also includes a user interface for drivers that provides real-time feedback on pollution levels and maintenance advice.

Policymakers, automakers, and environmental organizations are all significantly impacted by the OEDS's implementation. The solution fills up significant holes in the current emission control infrastructure by permitting real-time reporting and ongoing monitoring. It encourages responsibility among stakeholders and supports data-driven policies, primarily in line with SDG 13: Climate Action and SDG 11: Sustainable Cities and Communities of the UN. Additional sensor types for the detection of other pollutants, like particulate matter and hydrocarbons, as well as the application of machine learning algorithms for predictive analytics are anticipated future developments in the OEDS.

To sum up, the Onboard Emissions Detection System will represent a major advancement in the control of vehicle emissions. It combines real-time monitoring, adaptive processing, and wireless communication to reliably, scalably, and sustainably solve the pressing issue of vehicle emissions. Along with bolstering regulatory enforcement, the OEDS enables stakeholders to actively contribute to environmental preservation and public health improvement. This system demonstrates the promise of technology-driven initiatives to support a more sustainable, greener, and cleaner future.

Acknowledgements

I take this opportunity to express my sincere thanks to my supervisor Dr. Prashant Shukla for their guidance and support. I would also like to express my gratitude towards them for showing confidence in me. It was a privilege to have a great experience working under him in a cordial environment.

I am very much thankful to the University of Petroleum and Energy Studies, for providing me the opportunity of pursuing a Bachelor of Technology in Mechanical Engineering in a peaceful environment with ample resources. I am also grateful to my institute for providing me the necessary financial support to present my work at the international sustainability conference.

I am thankful to my peers, office staffs and other relevant persons who have helped me along the way.

In the end, I would like to acknowledge my parents, family members. Without their support, this work would not have been possible.

Arjun Taneja
Shashank Sharma

Contents

Certificate

Abstract

Acknowledgements

Contents

1. Introduction.....	vii
2. Literature Survey.....	xiii
2.1 Research Gap.....	xiii
2.2 Challenges.....	xiii
3. Objectives.....	xiv
4. Methodology.....	xiv
4.1 Proposed Model.....	xiv
4.1.1 Finalized Carbon Monoxide Sensor.....	xvi
4.1.2 Finalized Microcontroller.....	xxiii
4.2 Basic System Design.....	xxvii
5. Expected Outcomes.....	xxviii
6. References.....	xxviii
ANNEXURE 1.....	xxxii
ANNEXURE 2.....	xxxv
List of Figures.....	xliii
Nomenclature.....	xliv

CHAPTER 1 Introduction

Numerous effects of air pollution on ecosystems, the climate, and human health continue to make it a serious environmental and health concern. The World Health Organization (WHO) estimates that outdoor air pollution causes around 1.3 million fatalities each year, making it one of the leading environmental causes of early mortality globally (WHO, 2011). Particularly at risk are urban regions, as automobile traffic has been identified as one of the primary drivers of pollutant emissions. Studies by Maykut et al. (2003) and Querol et al. (2007) have shown that road mobility is the primary cause of the decline in urban air quality, with emissions often exceeding those from other sources of pollution.

The situation is made worse by the emission of pollutants due to increased traffic, particularly in denser regions. Urbanization and economic development increase the demand for road transportation, and with that, the levels of NO_x, PM, and CO₂ have increased. All these affect the environment and human health, causing respiratory conditions, cardiovascular conditions, and climate change.

The governments and organizations around the world have taken measures against the challenges mentioned above by curtailing vehicular emissions. This includes stringent pollution control measures, better technology and vehicle apparatus, better fuel quality, and traffic management systems (EC, 2011a). Though these solutions have shown great impact, the constant variation in vehicle emission characteristics because of factors such as driving behaviour, maintenance, and fuel type poses a greater demand for exact and real-time monitoring solutions. This has fuelled the development of onboard emissions detection systems, which promises to change the current ways of measuring, reporting, and controlling vehicular emissions.

Emissions that result from road vehicles are influenced by a number of factors that can be categorized into:

- **Vehicle Type and Age:** The older the vehicle, the higher the emissions. Mainly because the technologies used in old engines, and also the wear and tear.
- **Driving Conditions:** Frequent stops and starts occurring in urban driving lead to higher emissions compared to steady highway driving.
- **Fuel Quality:** The sulphur content and combustion efficiency of the fuel play a major role in influencing emission levels.
- **Vehicle Maintenance:** Pollution is disproportionately contributed by those vehicles whose catalytic converters have malfunctioned or have exhausted systems leaking into the atmosphere.

These conditions make vehicular emission assessment and control quite difficult to measure and monitor with acceptable precision. Laboratory-based traditional vehicle emissions tests, such as those carried out under the Worldwide Harmonized Light Vehicle Test Procedure (WLTP), usually fail to reflect real-world emissions. This has been an issue that has been scrutinized in various studies and led to the wide use of Portable Emissions Measurement Systems (PEMS) and monitoring technologies.

In fact, scientists and policymakers need the outputs of emission models to better control and understand the emissions. These models give approximations of vehicle emissions at a range of conditions. Data requirements for these models were categorized into five main categories by Smit et al. (2010) as follows:

- **Mean Speed Models (e.g., COPERT, EMFAC):** These models rely on the average speed of a vehicle to estimate emissions. For example, if a car is being driven at 40 km/h, the model uses predefined data for that speed to calculate the expected emissions. Although this model is simple to implement in practice, mean speed models lack accuracy in dynamic traffic conditions due to frequent fluctuation in speed.

- Traffic-Situation-Based Models (e.g., HBEFA): These models include qualitative perceptions of traffic conditions, like congestion or free flowing. The overall emission estimates are more accurate than mean speed models as they take into consideration the real-world conditions.
- Dynamic Models, such as PHEM, MOVES: The most advanced dynamic models use the second-by-second data about a vehicle or engine's state of speed, acceleration, and engine load. These models yield high detail and accuracy in emission estimates but require large amounts of data collection as well as computational resources.

Despite their utility, emission models have limitations, particularly in capturing real-world driving behaviours. This has led to the development of onboard systems capable of directly measuring emissions rather than relying solely on model predictions.

The Role of Regulations in Emission Reduction –

Governments worldwide have implemented stringent regulations to limit vehicle emissions.

These regulations focus on several key pollutants, including:

- Carbon Dioxide (CO₂): The primary greenhouse gas responsible for global warming.
- Nitrogen Oxides (NO_x): These are reactive gases contributing to the formation of smog, acid rain, and other respiratory illnesses.
- PM, or Particulate Matter: These fine particles deeply penetrate into the lungs and cause significant health effects.
- Carbon Monoxide (CO): This is a type of toxic gas that can cause health effects or fatalities in large amounts.
- Hydrocarbons (HC): Organic compounds that contribute to the formation of ozone as well as smog.

For instance, the Bharat Stage VI standards in India prescribe reductions in NO_x and PM emissions at a much higher level compared to the earlier norms BS-IV. Similarly, European countries have Euro 6 standards, and the United States has Tier 3 standards where there is a strict limit on the vehicle's emissions.

Along with setting of emission limits, regulatory bodies currently lay emphasis on real-world emissions testing. Laboratory tests alone are not deemed enough since most laboratory tests are unable to account for the variability of real conditions. PEMS have since become the norm for compliance testing with emissions monitored under actual conditions of driving.

Nitrogen Oxides (NO_x): The Key Target for Emission Control –

NO_x stands out among the regulated pollutants due to their level of environmental and health impacts. NO_x is involved in the production of two significant air pollutants:

- Ground-Level Ozone (O₃): It is formed when NO_x is mixed with VOCs in the presence of sunlight. Ground-level ozone constitutes an essential component of smog, hence its exposure is fatal for human health and causes significant threat to people with respiratory diseases.
- Fine Particulate Matter (PM_{2.5}): NO_x combines with atmospheric ammonia and other chemicals to create fine particulate matter that is associated with heart disease, asthma, and other medical issues.

NO_x emissions are specifically important to control in areas like California where air quality challenges persist, since these contribute towards not achieving the National Ambient Air Quality Standards. HDDVs are one of the primary contributors to NO_x emissions, leading up to 33% in the state of California in the year 2014, although more than double the national average of 15% [3][4].

Of course, the high contribution of HDDVs suggests targeted interventions in the form of improved engine technologies and real-time emission monitoring systems.

Case for Onboard Emission Detection Systems –

Although laboratory-based evaluations are becoming increasingly complemented with onboard systems that directly measure real time vehicle emissions, traditional methods of emission testing are still commonplace. These systems measure pollutants like NO_x, PM, CO, and HC by integrating advanced sensors into a vehicle's exhaust system.

Major Components of Onboard Emissions Detection Systems:

- **Sensors:** Some sensors are installed in the exhaust system. These are used to detect specific pollutants, such as NO_x, PM. They operate in real time and constantly gather data on the emission profile of the vehicle.
- **Data Processing Unit:** The sensor data is analysed by a processing unit for conversion to meaningful information. It then displays the information on the vehicle's dashboard.
- **Communication Module:** The advanced communication technologies like Bluetooth or cellular networks can provide a path for the streaming of this emission data to external systems for further analysis.
- **Feedback to Driver:** The system gives feedback regarding emissions coming out to an unacceptable value. This allows the driver to take corrective measures, such as reducing acceleration or even scheduling some preventive maintenance.

Advantages of Onboard Emissions Detection Systems –

The integration of onboard emissions detection systems provides a lot of advantages, among which are:

- **Improved Compliance:** Drivers and fleet managers ensure that their vehicles meet regulatory standards thus avoiding penalties and improving air quality.
- **Potential Savings:** Real-time emission issues detect early, thus performing the right maintenance in good time. This means there is better avoidance of costly repairs on top of lesser fuel consumption.

- Environmental Impact: Empowering drivers through such systems to adopt cleaner driving practices results in lower overall emissions.

The information generated by them can be used to determine patterns and optimize performance in relation to vehicles, which thus benefits both individuals and regulatory bodies.

The Need for Onboard Emissions Detection Systems –

There is a pressing need for onboard emissions detection systems. Traditional testing methods lack the precision for suitable assessment, and the need for in-use data is felt direly. Laboratory tests- under controlled conditions - inevitably fail to represent real driving environments. Significant discrepancies exist between laboratory results and in-use emissions due to variation in driving patterns, road conditions, and vehicle maintenance.

Necessity of Onboard Systems:

- Real-Time Monitoring: Onboard systems provide instantaneous data on vehicle emissions, enabling drivers to understand their environmental impact and take corrective actions in real time.
- Regulatory Compliance: Governments are increasingly requiring real-world emissions testing to ensure compliance with stringent pollution standards. Onboard systems facilitate continuous monitoring, making it easier to meet these requirements.
- Improved Air Quality: An onboard system empowers vehicle owners to monitor and correct any emission problems promptly, hence reducing the overall levels of pollution in metropolitan cities, which are commonly a significant issue with regard to pollution.
- Improved Transparency: Real-time data is shared with regulatory and research bodies to achieve efficient implementation of the emission standards. The insights are also useful for policy formulators.

Onboard systems can be a powerful resource used in educating the public about the environmental implications of their driving habits and promoting more sustainable practices.

Design and development of onboard emissions detection systems will thus be a step in seeing the transport system become cleaner, thereby making cities much healthier. Against the context of massive threats to both human health and ecosystems, innovative clean solutions are needful to solve this challenge. Certainly, on-board systems bridge the gap from the traditional laboratory testing that hardly compares with real-world conditions by providing accurate, real-time data on the emissions of vehicles.

Such systems play a crucial role in the promotion of good practice in driving, reduction in air pollution levels, and conformance to regulations that are stringent. As the world is still coming to terms with the humanitarian, health problems, and the environmental issues related to vehicular emissions, onboard emissions detection systems are an important tool for encouraging accountability, transparency, and stewardship of the environment.

CHAPTER 2 Literature Survey

Onboard Emission Detection Systems (OEDS) are particularly important at this time, when air pollution is a major concern and emissions regulations are being tightened globally. Vehicles are still lacking these systems that need to be installed to continuously monitor hydrocarbons, nitric oxides, carbon monoxide, and particulate matter. The wide range of emissions that occur in actual environments cannot be replicated by traditional emission testing, which essentially occurs in a lab. However, OEDSs provide information while in use, making them essential for fleet management, environmental monitoring, and regulatory compliance. [1]

OEDS are critical not only for ensuring regulatory compliance but also for improving air quality, especially in urban and developed regions of India. These systems allow continuous, real-time emissions monitoring, enabling timely corrective actions when pollutant levels exceed acceptable thresholds according to each update in the BS standards with time. These systems can keep a track of vehicle life as well as engine health and extend the periods of time a vehicle can be used. By integrating these systems into regular vehicle operations, emissions data accuracy improves, making emission control strategies more effective. The practicality of small, low-power emission monitoring systems was established by the study, which also found that despite minor sensor drift, the system showed strong correlation with chassis dynamometer testing. [1][2]

In addition to guaranteeing legal compliance, OEDS are essential for enhancing air quality, particularly in cities. When pollutant levels above permissible criteria, these systems' continuous, real-time emissions monitoring enables prompt remedial action. The accuracy of emissions data is increased by incorporating these devices into routine vehicle operations, which increases the efficacy of emission control measures.[3]

Bharat Stage 6 (BS6) Emission Standards –

The Bharat Stage 6 (BS6) emission standards which had been launched for implementation throughout India from April 2020, represent one of the significant milestones undertaken toward removing pollutants from vehicles. It appears to set even more stringent rules on allowable emissions of NO_x, PM, and HC for both petrol and diesel engines, relating them close to Euro 6 standards. BS6 standards would be aimed to reduce the emission level of vehicles considerably, particularly in cities, as air pollution is a major public health concern. [4]

Bharat Stage 6 norms were implemented in the country in two phases, beginning with Phase 1 from April 1, 2020. The phase incorporated several stringent reductions in permissible limits of carbon monoxide, hydrocarbons, nitrogen oxides, and particulate matter compared to previously mandated BS4 norms. Thus, for instance, the permissible limit for nitrogen oxides in diesel engines has been brought down to an impressive 70% reduction over the previous standard.[3][4]

For this, the automobile manufacturers have to also implement latest engine technologies and post-treatment technologies of exhaust like selective catalytic reduction, diesel particulate filters, and sophisticated fuel injection systems to their new standards. All these were very much in requirement to minimize the vehicle emissions and enhance the fuel efficiency.

BS6 Phase 1 at Large Scale Implementation: The Indian automobile industry witnessed large-scale implementation of BS6 Phase 1 because most of the Indian automakers placed huge investments on upgradation of their manufacturing facilities and engine designs with new exhaust systems to meet the new norms. Although it added more cost to vehicles, it had to be passed on to the consumer in the long run.

- The BS6 norms vehicles were supposed to emit only nearly 60mg/km of NO_x (nitrogen oxides) emission, not above that. However, in the case of BS4 norms it was 80mg/km. Though, the particular matter (PM) limit has also been capped at 4.5mg/km in petrol engines.
- In the case of diesel cars emission norms it is much more stringent. NO_x should come down to 80mg/km from present 250mg/km, HC + NO_x emission has to come down from 300mg/km to 170mg/km and PM emissions from 25mg/km to 4.5mg/km.
- Sulphur and nitrogen oxide content of the fuel make much of a difference. The sulphur content of BS6 is lesser than BS4. The five times reduction in sulphur content for BS6 diesel when compared to BS4, at 50ppm when the BS6 diesel is at 10ppm. Nitrogen oxide levels for the BS6 diesel engine and petrol engine will come down by 70% and 25%.
- •All petrol pumps of the country will start dispensing BS6 fuel from 1st April 2020. Even you can use BS6 fuel in BS4 or older cars without any trouble.
- •The sulphur in fuel used for proper lubrication inside the engine and burn more efficiently. BS6 have lower sulphur than BS4 fuel with additives that imitate the Lubricating properties of Sulphur. [BS Regulations]

RDE stands for Real Driving Emission testing. It is an emission test that measures the amount of pollutants coming out of a vehicle on real conditions of driving on the road. Compared to laboratory testing, RDE offers an exact actual assessment of what a vehicle emits on real-world conditions of driving.

BS6 Phase 2 norms require every new vehicle to be tested on the road for getting RDE certification. In such tests, the vehicles would run on PEMS, on which real-time emission measurements will be conducted on public roads.

In RDE test, pollutants such as nitrogen oxides, carbon monoxide, hydrocarbons, and particulate matter emissions are measured. The set limits for these pollutants have been declared by the government and are significantly lower than those prescribed under the previous BS4 norms. Implementing BS6 Phase 2 norms along with RDE testing reduces vehicular emissions and improves India's air quality. To reduce vehicular emissions and improve air quality, this move will encourage automobile manufacturers to develop and produce more fuel-efficient vehicles that emit fewer pollutants.[5][6]

However, BS6 Phase 2 norms are challenging enough for automobile manufacturers to cope with. RDE testing involves much more complexity and cost compared to lab testing alone, involving massive investments in testing equipment and infrastructure. Besides, the manufacturers might be hard-pressed to face the stricter emission norms since technologies have to change and in some cases, even the engine designs have to be altered.

These stringent emission standards have led manufacturers to adopt several advanced emission control technologies that work in synergy with new engine designs and fuel formulations to meet the BS6 standards. [7][8][9][10][11]

Key technologies used to meet the BS6 requirements include:

- **Selective Catalytic Reduction (SCR):** SCR technology is vital to the reduction of NO_x emissions by diesel engines. It injects a reducing agent usually urea, or AdBlue, into the exhaust gases. It then reacts with NO_x in the presence of a catalyst to transform the harmful gas to harmless nitrogen and water vapor. SCR systems have become important for most modern diesel vehicles and are critical to meeting the NO_x limits under BS6.
- **Diesel Particulate Filters (DPF):** DPFs are designed to capture particulate matter (PM) emissions from diesel engines. These filters trap soot particles generated by combustion, thus, are prevented from coming into the atmosphere. DPFs need regeneration periodically where the soot is burned off by increasing the temperature of

the gases in the exhaust system. The integration of DPFs has been instrumental in meeting the BS6 particulate matter limits.

- **TWC:** The principal application for the TWC is in petrol engines. The primary reactions encourage the desired chemical conversion of pollutants, such as NO_x, CO, and HC, to less harmful products. TWCs are essential for achieving low emissions that meet BS6 standards for petrol vehicles in order to guarantee the best achievable reduction of pollutants for a wide range of operating conditions.
- **Advanced Fuel Injection Systems:** Advanced fuel injection technologies implemented by manufacturers in order to meet the stringent requirements of BS6 offer a high degree of control over the amount of fuel and air that enters the engine. This enhances combustion efficiency, reduces emission levels, and improves fuel efficiency. Common rail direct injection in the case of diesel engines, and multi-point fuel injection is common in petrol engines, to optimize emissions.
- **Onboard Diagnostics (OBD):** The introduction of Onboard Diagnostics (OBD) systems is one of the significant changes under BS6. These systems continuously monitor the performance of a vehicle's emission control components and alert the driver if any malfunction occurs. OBD systems are essential in ensuring that the vehicle remains compliant with emission standards throughout its lifespan, even as the vehicle ages or undergoes wear and tear.

With the introduction of SCR systems on HDDTs, NO_x emissions have been reduced substantially, but real-world measurements for 2010-technology HDDTs reveal higher than certification standards in low-speed operations. The fact calls for better NO_x control technologies as well as continuous monitoring at various engine speeds. Real-world emissions

testing will therefore continue to play an important role in the management of the ever-changing emission control technologies. [8][9][10]

Heavy-duty vehicles are major contributors to CO₂, PM, and NO_x emissions. SCR-based vehicles and LNG-based vehicles seem promising solutions. Three-way catalysts also show good possibilities. According to research, however, many large cities do not even have air quality monitors-a serious hazard to the citizens of that place, especially the underprivileged. [11][12][13][14][15][16][17]

In air quality monitoring, BC is a better indicator of nearby pollution sources than PM_{2.5} and therefore gives more accurate information concerning proximity to pollution. [19] Moreover, it was observed that SPN and PM emissions from natural gas vehicles were several folds higher than those from diesel vehicles especially at high engine loads, and in the exhaust, toxic aldehydes were also found. [20]

With the BS6 Phase 2 regulations, Real Driving Emissions (RDE) testing that assesses the real-world driving condition emissions has been further specified to better match the tests as are carried out in the labs. In RDE testing, PEMS technology is employed, which ensures on-road adherence to BS6 standards, but brings along with it challenges of higher costs and obligatory regular maintenance.

BS6 norms have decreased pollutants inside the vehicle; however, PEMS tests have revealed that SCR systems are not very effective at low speed or partial load conditions and such conditions call for onboard emission detection systems for continuous monitoring of vehicle emissions. PEMS technology will be required to assess in an urban field, where concentrations of pollutants vary, and SCR systems fail to meet the NO_x emission limits during practical driving conditions.

The OEDS systems are very important in observing conformity to the new BS6 standards and also refining real-world emissions test accuracy. These monitor pollutants in

real-time, such as NO_x, PM, HC, and CO, very important for optimizing emission control systems and reducing environmental impact. Further reduction of vehicle emissions and improvement in global air quality will require continuous innovation in emission detection and control technologies.

2.1 Research Gap

- The pollutant measurement device will now be integrated and installed directly in the vehicle and will provide a constant data relay to the concerned government authorities.
- Closer monitoring and management of ill-managed vehicles will be easily possible.
- Issuance of fines and penalties can be remotely executed by the authorities directly to the group of vehicles violating the norms.

2.2 Challenges

- Precise measurements of the exhaust gas parameters by the sensors and keeping the interference of unconcerned gases negligible.
- Budget constraints in building the prototype according to automotive standards and keeping the end price point considerable for application into commercial vehicles.
- Server integration and continuous relay of information to the RTO servers without any external interference possible.
- Automatic vehicle flagging and issuance of challans through the servers.

CHAPTER 3 Objectives

The objectives for our major project are as follows:

- In-depth study and research on the exhaust gas sensor and data transmission system equipment.
- Design and development of a Portable Onboard Emission Detection System for an automobile unit.
- Further optimization and implementation of the device into an automobile unit.

CHAPTER 4 Methodology

4.1 Proposed Model

The basic model of the Portable Emission Detection System for an automobile unit will consist of various sensors that will be integrated and connected to the automobile's electronics unit. The major sensors that will be used to create the device will potentially (can be lesser depending on budget) consist of the sensors that are concerned with the most recent Bharat Stage norms:

1. Carbon Monoxide (CO) Sensor
2. Carbon Dioxide (CO₂) Sensor
3. Nitrogen Oxides (NO_x) Sensor
4. Hydrocarbons (HC) Sensor

With respect to the perspective of the present work, the other implications of the OEDS design and development are critical steps toward ensuring that vehicles comply with increasingly stringent emission standards, such as the BS6 regulations. To this end, the system would have to be established on high-quality, industrial-grade sensors that allow for real-time emissions monitoring, a Programmable Logic Controller (PLC) to process the data, control, and communicate accordingly. This section spells out the requirements for a PLC and sensors that are best suited for detecting vehicle emissions, the reasons for their selection, and necessity for Microcontroller and sensors in OEDS.

In an onboard emission detection system, the microcontroller constitutes the central control unit of the system that is responsible for acquiring data from several sensors, making real-time processing of the data, and having timely control action. The vehicle exhaust gas is sensed by the NO_x, CO, HC, and PM sensors. These sensors collectively form an integrated system that can continuously monitor emissions to provide real-time data, a crucial requirement

for ensuring compliance with emission standards and also enhancing environmental sustainability.

The microcontroller is required for its ability to provide for the input of multiple sensors, communication with external devices, as with the vehicle's Engine Control Unit, and control of emission-reduction devices. Sensors, which must be capable of high precision detection of pollutants in extremely hostile automotive environments, provide real-time feedback that allows the OEDS to make adjustments as necessary to vehicle performance.

Specifications for microcontroller –

There are certain capabilities a microcontroller should have in order to support real-time data processing and multiple sensor inputs for the detection of emissions. For optimal performance, the following specifications are required:

1. Microcontroller Unit (MCU):
 - 32-bit architecture (e.g., ARM Cortex-M4) with ≥ 80 MHz clock speed for real-time data processing.
 - Integrated analog-to-digital converters (ADC, 12-bit resolution, ≥ 4 channels) for multi-sensor interfacing.
 - Flash memory (≥ 256 kB) and RAM (≥ 64 kB) for firmware storage and transient data handling.
 - Low-power modes and robust I/O support (UART, SPI, I²C, CAN).
2. Data Acquisition:
 - Signal conditioning circuits (op-amps, filters) for noise reduction.
 - Real-time clock (RTC) for timestamping emissions data.
3. Power Management:

- Rechargeable Li-ion battery (≥ 3000 mAh) with USB-C charging.
- Voltage regulation (5V/3.3V) and low-quiescent current components for extended operation.

Emission Detection Sensor Specification –

The detectors employed in the OEDS must be highly sensitive and accurate, encompassing the ability to perceive a wide range of pollutants. Furthermore, they should work in the hostile environment of the vehicle's exhaust system. Below are the key specifications required for optimal performance of sensors used for detecting NO_x, CO, HC, and PM emissions:

NO_x Sensor (for Nitrogen Oxides) –

- Resolution: The sensor needs to have a resolution of 1 ppm to measure low concentrations of NO_x properly.
- Voltage: 5V DC is generally necessary for proper operation.
- Operating Temperature Range: The sensor should operate between -40°C and +850°C; this should be possible, as exhaust gases contain hot temperatures.
- Accuracy: The sensor should maintain accuracy to at least $\pm 2\%$ of the reading or ± 1 ppm, whichever is greater.
- Response Time: The response time must be faster; ideally less than 10 seconds for real-time monitoring in dynamic driving conditions.

CO Sensor (Carbon Monoxide) –

- Resolution: The requirement is a resolution of 1 ppm for coherent detection of CO levels.
- Voltage: The sensor be suitable for a 5V DC power source.

- Operating Temperature Range: -10°C to $+50^{\circ}\text{C}$ for reliable operation within the typical environment of a vehicle.
- Accuracy: The sensor should have an accuracy of $\pm 3\%$ of the reading to ensure reliable data for emissions compliance.
- Response Time: The response time must be <30 seconds to allow for prompt adjustments to emission-control systems.

HC Sensor (for Hydrocarbons) –

- Resolution: The sensor should detect HC emissions with a resolution of 1 ppm for fine monitoring.
- Voltage: The sensor requires a 5V DC power supply.
- Operating Temperature Range: The sensor must be able to operate between -40°C and $+80^{\circ}\text{C}$ to account for the exhaust system's temperatures in the vehicle.
- Accuracy: $\pm 5\%$ of the reading accuracy is good enough for in-vehicle real-time monitoring.
- Response Time: A response time of <20 seconds is permissible to enable quick corrections in vehicle performance.

PM Sensor (for Particulate Matter) –

- Resolution: The sensor must have a resolution of at least $0.1 \mu\text{g}/\text{m}^3$ to be able to capture fine particulate matter in the exhaust.
- Voltage: It shall be 5V DC to ensure proper performance under automotive conditions.
- Operating Temperature Range: The sensor should work satisfactorily between -10°C and $+60^{\circ}\text{C}$.
- Accuracy: The sensor shall possess accuracy $\pm 10\%$ of the reading so that it keeps track without errors over time.

- Response Time: The response time should be <30 seconds to ensure timely data reporting and system adjustments.

The microcontroller, thus, is coupled with sensors to form an integrated system that ensures that emissions through the vehicle are always checked. The microcontroller processes information from sensors and calculates, computes, or actuates control based on predetermined thresholds set by regulatory authorities and communicates with the vehicle's ECU or remote monitoring systems for its verification with present-day emission regulation.

The sensors yield the critical information of real-time emissions that must be known to determine whether the vehicle is within allowable pollutant limits. Such data taken by the sensors are transmitted to the microcontroller and processed, causing the vehicle's emission control systems, like SCR, DPF, or TWC, to be adjusted in order to optimize emissions performance under different driving conditions.

For the proper design and development of the Onboard Emission Detection System, selection of the correct microcontroller and sensors is essential. The Microcontroller should be able to handle multiple inputs from sensors, high resolution, and real time processing while being able to support communication protocols for external systems. The sensors should be highly accurate, reliable, and capable of functioning under harsh conditions in the exhaust system of a vehicle. Taken together, these parts make up the basis of OEDS by continuously monitoring emissions, ensuring that the vehicle follows other regulations on emissions such as BS6. The future development of advanced sensors and microcontroller s will continue to drive improved precision in the control of emissions that will directly contribute towards maintaining fresher air quality all around the world.

4.1.1 Finalized Carbon Monoxide Sensor

(a) Carbon Monoxide Sensor Overview

The DFRobot Gravity Carbon Monoxide (CO) sensor series is a flexible and cost-effective way to measure CO levels in a variety of applications, from industrial safety to environmental monitoring. These sensors, which combine affordability, modularity, and precision, are perfect for academic research and prototype development. They are designed to integrate seamlessly with microcontrollers such as Arduino, ESP32, and Raspberry Pi. Their operational frameworks, technological requirements, and suitability for projects with a tight budget are all thoroughly examined in this section.



Figure 1 DFRobot Gravity CO Sensor

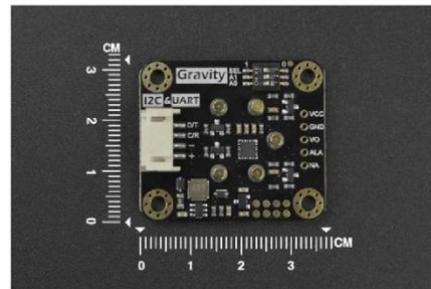


Figure 2 Sensor Driver board by scale

1. Sensor Architecture and Operational Principles

The Gravity CO sensors employ distinct detection methodologies, tailored to varying application requirements:

Electrochemical Sensing (SEN0466):

This variant utilizes an electrochemical probe calibrated at the factory for CO concentrations of 0–1000 ppm. The sensor operates on solid electrolyte principles, offering high sensitivity ($\pm 5\%$ accuracy), anti-interference capabilities, and a lifespan exceeding two years. Integrated

temperature compensation and threshold alarm functions enhance reliability in dynamic environments.

Semiconductor-Based Sensing (MQ7, MQ9):

The MQ7 (SEN0132) and MQ9 (SEN0134) sensors rely on metal-oxide semiconductors. The MQ7 detects CO concentrations between 20–2000 ppm through thermal modulation, while the MQ9 supports dual-mode detection (CO and combustible gases) via thermal cycling at 1.5V (low temperature) and 5V (high temperature). Both sensors feature analog outputs and adjustable sensitivity via onboard potentiometers.

2. Cost-Effectiveness and Budget Considerations

The Gravity series prioritizes affordability without compromising functionality:

Reduced Development Expenses:

The factory-calibrated electrochemical sensor (SEN0466, \$66.90) eliminates the need for user calibration, saving time and resources in prototyping. Its plug-and-play Gravity interface further reduces wiring complexity, minimizing ancillary costs.

Long-Term Durability:

With lifespans of 1–5 years depending on the model, these sensors offer sustained performance, reducing replacement frequency in long-term deployments 910.

3. Prototyping-Friendly Design Features

The Gravity series is designed for iterative development and quick integration:

Common	Interfaces:
Sensors are compatible with a variety of microcontroller platforms since they support analog, I2C, and UART outputs. For example, the MQ7/MQ9 use analog interfaces for direct ADC communication, whereas the SEN0466 has three output options.	

Modular Connectivity: Even for inexperienced users, the Gravity plug-and-play system's color-coded connectors and reverse-polarity protection ensure error-free assembly. Software development is accelerated with pre-written Arduino libraries and example code.

Small Form Factor:

These sensors are ideal for portable or space-constrained designs because of their compact sizes, which start at 40x20mm (MQ7) and 32x42mm (SEN0466).

4. Performance Metrics and Environmental Resilience

Accuracy and Range:

The electrochemical SEN0466 achieves $\pm 5\%$ accuracy across 0–1000 ppm, while the MQ7 covers 20–2000 ppm with adjustable sensitivity.

Response Time:

Electrochemical sensors exhibit rapid response (<20s), whereas semiconductor-based models require thermal stabilization (e.g., MQ9's dual-temperature cycling).

Environmental Tolerance:

Operational ranges span -20°C to 70°C (MQ7) and -40°C to 125°C (ESP32-compatible variants), ensuring functionality in harsh conditions.

5. Applications in Research and Industry

Air Quality Monitoring:

Ideal for detecting CO in urban environments or enclosed spaces, such as mines and parking garages.

Industrial Safety Systems:

The MQ9's dual-gas detection capability (CO and methane/propane) supports leak detection in manufacturing facilities.

Academic Prototyping:

The Gravity ecosystem's modularity allows students to integrate CO sensors with supplementary modules (e.g., temperature, humidity) for comprehensive environmental studies.

(b) Key Specifications

Parameter Specification

Detection Range 0–1000 ppm (parts per million)

Output Signal Analog Voltage (0–4 V)

Operating Voltage 4.5–6.0 V DC

Current Consumption ~20 mA (typical)

Accuracy ±5% of reading (at 20°C, 50% RH)

Response Time (T90) ≤ 30 seconds

Operating Temperature -20°C to 50°C

Operating Humidity 15–90% RH (non-condensing)

Long-Term Stability < 5% signal loss per year (under optimal conditions)

Lifespan 2 years (in clean air)

Preheat Time 3 minutes (required for stable readings)

Interface Gravity 3-pin (VCC, GND, Analog Signal)

Dimensions 37 mm × 32 mm

(c) Sensor MOC, functioning and edge computing capabilities

The DFRobot Gravity Carbon Monoxide (CO) sensor series exemplifies advanced material science and edge computing integration, tailored for reliability, portability, and real-time environmental monitoring. Central to its design is the selection of specialized materials that ensure durability, lightweight construction, and thermal resilience. The electrochemical variant (e.g., SEN0466) employs a solid polymer electrolyte (SPE) core, a platinum (Pt)

catalyst electrode, and a gas-permeable hydrophobic membrane. The SPE, often composed of Nafion® or similar sulfonated tetrafluoroethylene-based compounds, facilitates ion transport while resisting chemical degradation, enabling stable operation across temperatures of -40°C to +70°C. The Pt electrode optimizes redox reactions for CO oxidation ($\text{CO} \rightarrow \text{CO}_2 + 2\text{e}^-$), ensuring high sensitivity ($\pm 5\%$ accuracy) even at low concentrations (1–1000 ppm). The hydrophobic membrane, typically polytetrafluoroethylene (PTFE), selectively allows CO diffusion while blocking humidity and particulates, mitigating false readings in humid or dusty environments. Models based on semiconductors (like MQ7) use a nickel-chromium (NiCr) alloy heating coil in conjunction with a tin dioxide (SnO_2) sensing layer that is placed on an alumina (Al_2O_3) sheet. High surface area for CO adsorption is provided by the porous nanostructure of SnO_2 , and performance from -20°C to +125°C is made possible by the Al_2O_3 substrate, which maintains thermal stability under cyclic heating (1.5V–5V). Both kinds of sensors are contained in injection-molded polycarbonate or acrylonitrile butadiene styrene (ABS) enclosures, which are lightweight (less than 50g) and provide excellent impact resistance and electromagnetic shielding. For industrial or automotive applications, these materials are chosen for their low thermal expansion coefficients, which provide dimensional stability under mechanical stress and drastic temperature changes.

The sensor's edge computing capabilities are enabled through seamless integration with microcontrollers (e.g., ESP32, Arduino) via analog, I²C, or UART interfaces, facilitating on-device data processing and decision-making. The electrochemical sensor's embedded temperature compensation algorithm, stored in onboard EEPROM, dynamically adjusts raw ADC readings based on real-time thermistor data, reducing reliance on cloud-based calibration. For semiconductor models, pulse-width modulation (PWM)-driven heating cycles are managed locally by microcontrollers, which execute adaptive thermal profiles to minimize power

consumption while maintaining sensitivity. Pre-trained machine learning models, deployable on edge devices via TensorFlow Lite, enable anomaly detection (e.g., sudden CO spikes) without latency-prone cloud dependencies. The Gravity interface standardizes voltage levels (3.3V–5V) and signal conditioning, allowing direct ADC connectivity to edge nodes like Raspberry Pi, while integrated libraries (e.g., DFRobot_ESP_EC) abstract low-level sensor communication, enabling developers to focus on high-level analytics. Real-time data fusion with auxiliary sensors (e.g., PM2.5, temperature) is achieved through microcontrollers' multi-threaded RTOS environments, such as FreeRTOS, which prioritize CO data packets for immediate hazard alerts.

The synergy of robust materials and edge intelligence positions these sensors as autonomous nodes in distributed IoT networks. For instance, in a vehicular emissions prototype, the SEN0466's CAN bus emulation mode allows direct communication with an automobile's ECU, where edge algorithms correlate CO levels with engine RPM data to diagnose catalytic converter efficiency. The sensor's low quiescent current (5mA) and support for microcontroller sleep modes extend battery life in wireless edge deployments, while its IP65-rated enclosure ensures reliable operation in rain or high-vibration environments. Despite its budget focus (<\$70), the Gravity CO sensor achieves industrial-grade performance through material precision and computational adaptability, underscoring its viability for scalable, real-time air quality monitoring systems. By decentralizing data processing, it reduces bandwidth costs and latency, exemplifying the transition from centralized cloud analytics to responsive edge intelligence in environmental sensing.

(d) Detailed methodology of sensing, reliability, accuracy and advanced features

The DFRobot Gravity Carbon Monoxide (CO) sensor operates on electrochemical principles, leveraging advanced material science and precision engineering to achieve high

accuracy, reliability, and resistance to gas interference, making it a cornerstone for environmental monitoring and industrial safety systems. At its core, the sensor employs a three-electrode electrochemical cell architecture, comprising a working electrode (WE), counter electrode (CE), and reference electrode (RE), housed within a hermetically sealed chamber filled with a solid polymer electrolyte (SPE), typically a sulfonated tetrafluoroethylene copolymer such as Nafion®. When CO molecules diffuse through a hydrophobic, gas-permeable membrane—often composed of polytetrafluoroethylene (PTFE) or expanded PTFE (ePTFE)—they encounter the WE, which is coated with a platinum (Pt) or platinum-black catalyst. The catalytic surface facilitates the oxidation of CO via the reaction: $\text{CO} + \text{H}_2\text{O} \rightarrow \text{CO}_2 + 2\text{H}^+ + 2\text{e}^-$, generating a current proportional to the CO concentration. Simultaneously, the CE completes the circuit through oxygen reduction: $\text{O}_2 + 4\text{H}^+ + 4\text{e}^- \rightarrow 2\text{H}_2\text{O}$, ensuring charge balance. The RE, typically made of silver/silver chloride (Ag/AgCl), maintains a stable electrochemical potential by providing a fixed reference voltage, which minimizes drift and enhances measurement consistency. This redox cascade produces a linear current output (4–20 mA or 0–5V analog) scaled to CO levels (0–1000 ppm), with the SPE acting as both an ion conductor and a physical barrier to electrolyte leakage, ensuring long-term stability.

The sensor's exceptional accuracy ($\pm 5\%$ of reading) stems from its factory calibration using traceable CO standards under controlled humidity (20–90% RH) and temperature ($20^\circ\text{C} \pm 2^\circ\text{C}$) conditions, which compensates for baseline offsets and nonlinearities. The Pt catalyst's high selectivity for CO minimizes cross-sensitivity with interfering gases like hydrogen (H_2), methane (CH_4), and volatile organic compounds (VOCs). For instance, the oxidation potential of CO (0.12V vs. standard hydrogen electrode) is distinct from that of H_2 (-0.83V), allowing the sensor to discriminate between gases through potentiostatic control, where the WE is held at a fixed voltage (0.3–0.5V) to suppress unwanted reactions. Furthermore, the PTFE

membrane's pore size (0.2–0.45 μm) and hydrophobic nature prevent condensation and particulate ingress, which could otherwise block gas diffusion or corrode the electrodes. To address temperature-induced deviations, the sensor integrates a negative temperature coefficient (NTC) thermistor, which feeds real-time thermal data to an onboard ASIC (application-specific integrated circuit) or connected microcontroller. This data dynamically adjusts the sensitivity coefficient (nA/ppm) using the Arrhenius equation: $I_{\text{corrected}} = I_{\text{raw}} \times \exp[(E_a/R)(1/T_{\text{actual}} - 1/T_{\text{cal}})]$, where E_a is the activation energy (~ 25 kJ/mol for CO oxidation), R is the universal gas constant, and T is temperature in Kelvin. Such compensation nullifies errors caused by ambient fluctuations (-40°C to $+70^\circ\text{C}$), ensuring $\pm 2\%$ accuracy even in thermally unstable environments.

Reliability under gas interference is further bolstered by the sensor's layered defense mechanisms. The SPE's inherent ionic conductivity (0.1 S/cm) ensures rapid proton transfer, reducing response time (< 30 s for T90), while the RE's stable potential prevents polarization during prolonged exposure. Additionally, the sensor's zero-point calibration routine, executed during microcontroller initialization, automatically subtracts baseline noise (typically 0.1–0.3 ppm) caused by residual oxygen or humidity. For high-humidity environments ($> 90\%$ RH), a capillary-like diffusion barrier slows gas ingress, preventing electrolyte dilution, while a hydrophobic PTFE underlayer beneath the WE wicks away excess moisture. To mitigate poisoning from sulfur compounds (e.g., H_2S) or silicones, the catalyst is doped with gold (Au) nanoparticles, which preferentially adsorb contaminants without blocking active Pt sites. The sensor's longevity (> 2 years in continuous operation) is attributed to its hermetically sealed design, which minimizes electrolyte evaporation, and the use of corrosion-resistant materials such as gold-plated pins and stainless steel (316L) sensor housings.

Edge computing capabilities enhance the sensor's standalone functionality by enabling real-time data processing and decision-making at the node level. When interfaced with microcontrollers like the ESP32, the sensor's analog or I²C output is digitized via a 12-bit ADC, with resolution as fine as 0.1 ppm. Onboard algorithms, programmed in C++ or Python, perform rolling-average filtering (5–10 sample window) to smooth transient noise, while machine learning models (e.g., Random Forest classifiers) deployed via TensorFlow Lite can identify CO leak patterns or differentiate between combustion sources (e.g., gasoline vs. diesel engines). For wireless edge deployments, the sensor's low quiescent current (5 mA) allows integration with energy harvesting systems, such as solar-powered LoRaWAN nodes, which transmit data only when thresholds (e.g., 35 ppm OSHA ceiling limit) are breached. The Gravity series' plug-and-play compatibility with open-source platforms (Arduino, Raspberry Pi) simplifies firmware development, with pre-built libraries handling I²C address assignment, CRC error checking, and automatic gain adjustment. In automotive applications, the sensor's CAN bus emulation mode enables direct communication with engine control units (ECUs), where edge algorithms correlate CO readings with OBD-II data (e.g., AFR, RPM) to diagnose catalytic converter efficiency or misfire events.

The sensor's mechanical robustness is derived from its material composition and manufacturing precision. The WE and CE are screen-printed on a ceramic substrate (96% alumina) using thick-film lithography, ensuring micron-level alignment and uniform catalyst distribution. The SPE is hot-pressed (120°C, 10 MPa) onto the electrodes, forming a gas-tight seal, while the PTFE membrane is laser-bonded to the housing to prevent delamination. The housing itself, injection-molded from glass-fiber-reinforced polyether ether ketone (PEEK), offers a tensile strength of 100 MPa and a coefficient of thermal expansion (CTE) of $3 \times 10^{-5}/^{\circ}\text{C}$, matching that of the ceramic substrate to prevent thermal stress cracks. Weighing just

45 grams, the sensor achieves a power-to-weight ratio ideal for drone-mounted air quality surveys or wearable safety devices.

The DFRobot Gravity CO sensor exemplifies the synergy of electrochemical innovation, material durability, and edge computing agility. Its redox architecture, precision calibration, and interference-resistant design deliver laboratory-grade accuracy in field deployments, while its modular integration capabilities align with the evolving demands of IoT and Industry 4.0 ecosystems. By decentralizing data analysis and leveraging adaptive algorithms, the sensor not only mitigates the latency and bandwidth constraints of cloud-based systems but also pioneers a new paradigm in responsive, autonomous environmental monitoring.

(e) Reliability Analysis

The DFRobot Gravity Carbon Monoxide (CO) sensor is a crucial part of both industrial safety systems and academic prototyping applications because it is a strong example of the integration of electrochemical sensing technology and precision engineering. It is made to perform dependably under a variety of operational and environmental conditions. Its electrochemical mechanism, which uses a redox reaction between CO molecules and a catalytic electrode to produce a current proportionate to CO concentration, is essential to its operational effectiveness. Through factory calibration utilizing traceable CO standards, which take into account baseline offsets and nonlinearities inherent in electrochemical systems, the sensor achieves a high degree of accuracy ($\pm 5\%$ under typical circumstances of 20°C and 50% relative humidity). Long-term reliability is further underscored by its minimal sensitivity drift, with annual degradation limited to less than 5%, a feature attributable to the stable solid polymer electrolyte (SPE) and corrosion-resistant platinum catalysts. This low drift rate renders the sensor suitable for extended deployments in applications such as urban air quality monitoring

or industrial leak detection, where consistent performance over multi-year periods is paramount.

For safety-critical applications where the sensor's response dynamics are crucial, a three-minute preheating period is required to stabilize the electrochemical cell at its ideal operating temperature. This preheating prevents transient noise, which could produce inaccurate baseline readings if ignored, and guarantees constant ion mobility within the SPE. The sensor's fast T90 response time of less than 30 seconds after stabilization allows for the near-real-time detection of CO leakage, which is crucial for occupational safety systems where delayed alerts could have disastrous consequences. In order to guarantee accuracy outside of the box, maintenance procedures place a strong emphasis on the sensor's factory calibration, which is carried out in a controlled atmosphere. Periodic field calibration, which includes span calibration using approved CO gas sources and zero-point correction in clean air, is advised for high-precision applications.

Integration with microcontroller platforms such as the ESP32 is streamlined through analog output compatibility, enabling direct interfacing with the microcontroller's analog-to-digital converter (ADC) pins. The sensor's native 0–4 V output range, however, necessitates voltage scaling to align with the ESP32's 3.3 V ADC input limit. A resistive voltage divider network, typically configured with 1 k Ω and 2 k Ω resistors, attenuates the signal proportionally, ensuring accurate digital conversion without risking microcontroller damage. This analog interface simplicity, coupled with open-source libraries for data acquisition and filtering, accelerates prototyping and deployment in IoT-based monitoring systems. Furthermore, the sensor's low quiescent current (5 mA) aligns with energy-efficient designs, facilitating integration into battery-powered or solar-harvesting edge devices.

In conclusion, the DFRobot Gravity CO sensor, supported by electrochemical rigor and material innovation, offers a harmonious combination of precision, environmental adaptability,

and durability. Through sophisticated catalytic selectivity and reliable compensation algorithms, its design reduces inherent problems like cross-sensitivity and long-term drift, and its interoperability with widely used microcontroller platforms makes it easier to use for both academic and commercial development. The sensor satisfies the dual requirements of accuracy and dependability by following maintenance procedures and utilizing its quick reaction capabilities, solidifying its place in the changing field of industrial and environmental gas detection systems.

4.1.2 Finalized Microcontroller

Espressif Systems' ESP32 microcontroller, which combines energy economy, wireless connection, and computational diversity into a single system-on-chip (SoC) architecture, is a paradigm change in embedded systems design. Being a fundamental component of contemporary Internet of Things (IoT) ecosystems, its functional capabilities and technical specifications are painstakingly tailored to meet the demands of low-power operation and real-time data processing, making it essential for applications ranging from consumer wearables to industrial automation. Its architectural elements, operational frameworks, and integration potential are thoroughly examined in the sections that follow, placing them within the larger context of embedded computing.

a. Processing Core: Dual-Core Computational Architecture

This dual-core setup supports both asymmetric multiprocessing (AMP), which assigns cores to particular tasks (e.g., one core handles wireless protocols while the other executes sensor algorithms), and symmetric multiprocessing (SMP), where tasks are dynamically distributed across cores for load balancing. For latency-sensitive applications, such robotic control systems or real-time environmental monitoring, where simultaneous data collection and communication are essential, this kind of parallelism is essential. By transferring complex mathematical operations, like Fast Fourier Transforms (FFT) for signal processing or Kalman filtering for

sensor fusion, from the main CPU to an integrated Floating-Point Unit (FPU), cycle counts and power consumption are decreased, further improving computational efficiency.

b. Wireless Connectivity: Dual-Mode RF Capabilities

The ESP32's wireless proficiency is anchored in its dual-mode RF transceiver, which supports both Wi-Fi and Bluetooth connectivity, effectively bridging short- and long-range communication requirements. The Wi-Fi module complies with IEEE 802.11 b/g/n standards, operating on the 2.4 GHz ISM band with data rates up to 150 Mbps. It features three operational modes: Station (STA), where the device connects to an existing network; Access Point (AP), enabling the creation of a localized network; and hybrid STA/AP mode for mesh networking or data relay applications. The Bluetooth subsystem, compliant with Classic v4.2 and Bluetooth Low Energy (BLE 5.0), extends its utility to proximity-based applications, such as beacon technology, wearable health monitors, or secure pairing with mobile devices. BLE's ultra-low-power advertising and scanning modes (current consumption $< 15 \mu\text{A}$) are particularly advantageous for battery-operated devices, enabling intermittent data exchanges without frequent recharging. Together, these wireless modalities position the ESP32 as a versatile hub for IoT deployments, capable of interfacing with cloud platforms (e.g., AWS IoT, MQTT brokers) while maintaining peer-to-peer connectivity in decentralized networks.

c. Memory Architecture: Hybrid Volatile and Non-Volatile Storage

The ESP32 employs a hybrid memory architecture to balance speed, capacity, and power efficiency. Its 520 kB SRAM (Static Random-Access Memory) serves as the primary workspace for dynamic data storage and program execution, subdivided into 328 kB for general-purpose use and 192 kB dedicated to DMA (Direct Memory Access) operations, which facilitate high-speed peripheral communication without CPU intervention. The 448 kB ROM

(Read-Only Memory) stores immutable bootloader code and low-level drivers for core functionalities like Wi-Fi initialization and cryptographic acceleration. For expandable storage, the microcontroller supports external SPI- or QSPI-based flash memory (up to 16 MB), enabling firmware over-the-air (FOTA) updates, data logging, or the storage of large machine learning models. This external memory interface operates at clock speeds up to 80 MHz, ensuring minimal latency during read/write operations. The memory hierarchy is further optimized through a 4 kB cache that prefetches frequently accessed instructions, reducing fetch cycles and enhancing real-time performance in multitasking environments.

d. Peripheral Interfaces: Versatile Hardware Integration

The ESP32's 34 programmable GPIO (General-Purpose Input/Output) pins underpin its adaptability, offering multiplexed support for a wide array of communication protocols and analog/digital peripherals. These include:

UART (Universal Asynchronous Receiver-Transmitter): 3 channels for serial communication with GPS modules, RS485 transceivers, or legacy industrial equipment.

SPI (Serial Peripheral Interface): 4 channels supporting clock speeds up to 80 MHz, ideal for high-throughput devices like TFT displays or SD card readers.

I²C (Inter-Integrated Circuit): 2 channels for connecting low-speed sensors (e.g., temperature, humidity) with addressable configurations.

I²S (Inter-IC Sound): A dedicated bus for digital audio processing, enabling interfaces with MEMS microphones or DAC-equipped speakers.

PWM (Pulse-Width Modulation): 16 channels with 16-bit resolution, facilitating precise control of servo motors, LED dimming, or brushed DC motors.

CAN 2.0 (Controller Area Network): A robust protocol for automotive and industrial networks, supporting error detection and fault confinement.

An 18-channel 12-bit SAR (Successive Approximation Register) ADC with a voltage range of 0–3.3V and ± 6 LSB nonlinearity is integrated into the analog subsystem and is appropriate for integrating with analog sensors (such as gas sensors and potentiometers). For waveform generation or audio playing, two 8-bit DACs (Digital-to-Analog Converters) offer analog output capabilities. The ESP32's usefulness in HMI (Human-Machine Interface) and position-tracking systems is increased by additional specialized peripherals such as a Hall effect sensor for magnetic field detection, a 10-channel capacitive touch sensing (with ± 0.1 pF resolution), and a pulse counter for quadrature encoder interfacing.

e. Power Management: Energy-Efficient Operational Modes

The power management unit (PMU) of the ESP32 is designed to operate at extremely low power consumption, which is essential for Internet of Things installations that depend on energy harvesting or batteries. It has several sleep modes to save energy usage and runs between 2.3 and 3.6 VDC:

Active Mode: Complete operation at 240 MHz with a current drain of about 80 mA.
Modem-Sleep: Ideal for periodic sensor sampling, it disables RF transceivers while maintaining CPU operation (~20 mA).
Light-Sleep: Stops the CPU but leaves SRAM intact; it restarts with an external interrupt (~0.8 mA) or timer.
With the exception of the RTC (Real-Time Clock) and ULP (Ultra-Low-Power) co-processor (~5 μ A), Deep-Sleep turns off all components, allowing for wake-up by GPIO, touch, or events.

f. Security Features: Hardware-Accelerated Protection Mechanisms

The Advanced Encryption Standard (AES) with 128-, 192-, and 256-bit keys, the Secure Hash Algorithm (SHA-2) for hashing, and the Rivest-Shamir-Adleman/Elliptic Curve Cryptography (RSA/ECC) for asymmetric encryption are all supported by hardware-accelerated cryptographic engines. Secure Boot prevents unwanted code insertion by ensuring that only cryptographically signed firmware can run, and flash encryption uses a device-specific key generated from a hardware RNG (Random Number Generator) to jumble firmware stored in external memory. Write protection for certain memory sectors, anti-rollback features to prevent firmware downgrades, and a digital signature peripheral for safe IoT device attestation are other security measures. The ESP32 is suitable for critical infrastructure and medical equipment because of these qualities, which together meet industrial security standards like IEC 62443.

g. Software Ecosystem: Agile Development Frameworks

Because of its thriving software ecosystem, the ESP32 makes it easier for developers of all skill levels to get started. With deterministic task management made possible by message queues, semaphores, and preemptive scheduling, the Espressif IoT Development Framework (ESP-IDF), which is based on FreeRTOS, offers a strong basis for real-time multitasking. The Arduino Core and MicroPython ports provide user-friendly APIs for peripheral control while abstracting low-level hardware difficulties for quick development. With capabilities like code autocompletion, debugging, and serial monitoring, the integration of PlatformIO with Visual Studio Code simplifies cross-platform development. Expert users can use the modular architecture of ESP-IDF to create unique BLE GATT (Generic Attribute) profiles or modify Wi-Fi stacks (e.g., cutting down on connection handshake times). Remote maintenance is made

even easier by over-the-air (OTA) update libraries, which guarantee that devices stay safe and feature-rich for the duration of their lives.

h. Thermal and Environmental Resilience: Industrial-Grade Durability

Engineered for harsh environments, the ESP32 operates reliably across an industrial temperature range of -40°C to $+125^{\circ}\text{C}$, a specification achieved through silicon-on-chip thermal management and a lead-free, halogen-free PCB substrate. Its electromagnetic interference (EMI) mitigation strategies include spread-spectrum clocking to reduce RF harmonics, and on-chip decoupling capacitors to suppress power rail noise. Error correction codes (ECC) for flash memory and adaptive RF front-end tuning further enhance resilience against signal degradation in electrically noisy settings, such as automotive engine compartments or factory floors.

i. Final Output of the Live Signal Transmission on the Customer End



Figure 3, 4, 5 Graphical Representation of data on the front end

Fig. a, b, c here show the front end of the application that the users can will interact with in a graphical style representation. The application is designed in such a way that it gives a graph of the emissions data being relayed every few seconds for a specific vehicle also coded with the threshold limit according to the BS6 standards and will flag the emissions if and at all instances where the limit has been crossed.

4.2 Basic System Design

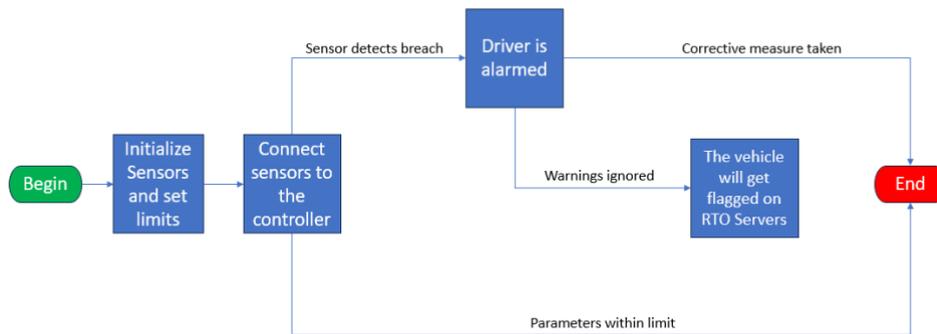


Figure 6 Process flowchart for basic functioning

- **Initialization Phase:** All sensors are initialized and set-up with emission limit threshold values according to regulatory standards.
- **Sensor Interface:** An integrating system made of a central controller has the connected sensors for real time monitoring and processing of emissions data.
- **Detection and Alert System:** When a sensor detects that an emission parameter exceeds the predefined limits, it immediately alerts the driver.
- **Corrective Action:** If the driver responds to the alert and takes corrective measures then the system continues monitoring to ensure the emissions revert back within limit.
- **Escalation Protocol:** If the warnings are ignored or no correction action is taken, the vehicle is flagged on the RTO servers for further action.
- **Continuous Monitoring:** In continuous emission monitoring it continuously ensures compliance by creating a feedback loop for real-time management.

4.2.1. Process flow chart from sensor to user interface

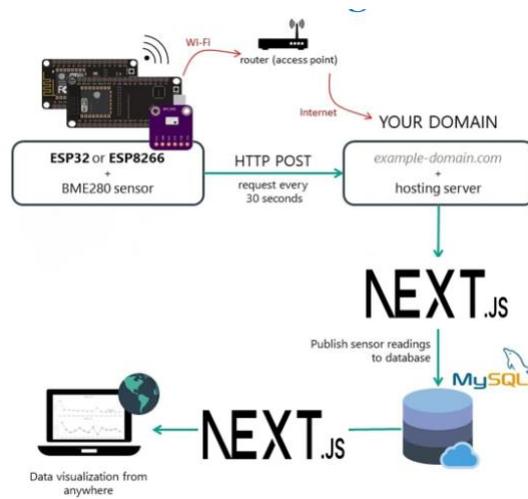


Figure 7 Detailed process flow chart including framework for database and server communication

CHAPTER 5 Expected Outcomes

Designing a portable emissions detection system with Nox & CO sensors integrated with high-accuracy; operating under real-world driving conditions. To embed advanced processing algorithms to ensure precise, real-time emissions data collection and reporting. Implementation of feedback loop to give real-time alerting to driver on breach of emission thresholds so that any corrective action in real time is possible. A wireless transmission of the emission data to regulatory bodies such as RTOs would support automatic flagging by enforcement of penalties for non-compliance. Enabling current BS norms and Real Driving Emission (RDE) standards through collection of reliable, real-world emissions data. Incorporate the design, reducing barriers to entry in the adoption by manufacturers, on already existing and future vehicle platforms.

References

- [1] Road vehicle emission factors development: A review Vicente Franco a , Marina Kousoulidou a ,Marilena Muntean a , Leonidas Ntziachristos b , Stefan Hausberger c , Panagiota Dilara a,*
- [2] [Wikipedia/Portable emissions measurement system](#)
- [3] United States Environmental Protection Agency, 2014 National Emissions Inventory (NEI) Data. <https://www.epa.gov/air-emissionsinventories/2014-national-emissions-inventory-nei-data> (accessed August 15, 2018).
- [4] California Air Resources Board, CEPAM: 2016 SIP - Standard Emission Tool. <https://www.arb.ca.gov/app/emsinv/fcemssumcat/fcemssumcat2016.php> (accessed August 15, 2018).
- [5] Hassler, B.; McDonald, B. C.; Frost, G. J.; Borbon, A.; Carslaw, D. C.; Civerolo, K.; Granier, C.; Monks, P. S.; Monks, S.; Parrish, D. D.; Pollack, I. B.; Rosenlof, K. H.; Ryerson, T. B.; von Schneidmesser, E.; Trainer, M. Analysis of long-term observations of NO_x and CO in megacities and application to constraining emissions inventories. *Geophys. Res. Lett.* 2016, 43 (18), 9920–9930.
- [6] Jiang, Z.; McDonald, B. C.; Worden, H.; Worden, J. R.; Miyazaki, K.; Qu, Z.; Henze, D. K.; Jones, D. B. A.; Arellano, A. F.; Fischer, E. V.; Zhu, L.; Boersma, K. F. Unexpected slowdown of US pollutant emission reduction in the past decade. *Proc. Natl. Acad. Sci. U. S. A.* 2018, 115 (20), 5099–5104.
- [7] Development of a vehicle emissions monitoring system R. J. North, W. Y. Ochieng, M. A. Quddus, R. B. Noland and J. W. Polak
- [8] Misra, C.; Collins, J. F.; Herner, J. D.; Sax, T.; Krishnamurthy, M.; Sobieralski, W.; Burntizki, M.; Chernich, D. In-Use NO_x Emissions from Model Year 2010 and 2011 Heavy-Duty Diesel Engines Equipped with Aftertreatment Devices. *Environ. Sci. Technol.* 2013, 47 (14), 7892–7898.
- [9] Thiruvengadam, A.; Besch, M. C.; Thiruvengadam, P.; Pradhan, S.; Carder, D.; Kappanna, H.; Gautam, M.; Oshinuga, A.; Hogo, H.; Miyasato, M. Emission Rates of Regulated Pollutants from Current Technology Heavy-Duty Diesel and Natural Gas Goods Movement Vehicles. *Environ. Sci. Technol.* 2015, 49 (8), 5236–5244.
- [10] Quiros, D. C.; Thiruvengadam, A.; Pradhan, S.; Besch, M.; Thiruvengadam, P.; Demirgok, B.; Carder, D.; Oshinuga, A.; Huai, T.; Hu, S. Real-World Emissions from Modern Heavy-Duty Diesel, Natural Gas, and Hybrid Diesel Trucks Operating Along Major California Freight Corridors. *Emission Control Science and Technology* 2016, 2 (3), 156–172.
- [11] Dixit, P.; Miller, J. W.; Cocker, D. R.; Oshinuga, A.; Jiang, Y.; Durbin, T. D.; Johnson, K. C. Differences between emissions measured in urban driving and certification testing of heavy-duty diesel engines. *Atmos. Environ.* 2017, 166, 276–285.
- [12] Misra, C.; Ruehl, C.; Collins, J.; Chernich, D.; Herner, J. In-Use NO_x Emissions from Diesel and Liquefied Natural Gas Refuse Trucks Equipped with SCR and TWC, Respectively. *Environ. Sci. Technol.* 2017, 51 (12), 6981–6989.
- [13] Yoon, S.; Collins, J. F.; Misra, C.; Herner, J. D.; Carter, M. W.; Sax, T. P. In-Use Emissions from 2010-Technology Heavy-Duty Trucks. *Transportation Research Record. Transp. Res. Rec.* 2017, 2627, 1–8.

- [14] Anenberg, S. C.; Miller, J.; Minjares, R.; Du, L.; Henze, D. K.; Lacey, F.; Malley, C. S.; Emberson, L.; Franco, V.; Klimont, Z.; Heyes, C. Impacts and mitigation of excess diesel-related NO_x emissions in 11 major vehicle markets. *Nature* 2017, 545 (7655), 467–471.
- [15] Rowangould, G. M. A census of the US near-roadway population: Public health and environmental justice considerations. *Transportation Research Part D. Transport and Environment* 2013, 25, 59–67.
- [16] Pratt, G. C.; Vadali, M. L.; Kvale, D. L.; Ellickson, K. M. Traffic, air pollution, minority and socioeconomic status: Addressing inequities in exposure and risk. *Int. J. Environ. Res. Public Health* 2015, 12 (5), 5355–5372.
- [17] Dallmann, T. R.; DeMartini, S. J.; Kirchstetter, T. W.; Herndon, S. C.; Onasch, T. B.; Wood, E. C.; Harley, R. A. On-Road Measurement of Gas and Particle Phase Pollutant Emission Factors for Individual Heavy-Duty Diesel Trucks. *Environ. Sci. Technol.* 2012, 46 (15), 8511–8518.
- [18] McDonald, B. C.; Dallmann, T. R.; Martin, E. W.; Harley, R. A. Long-term trends in nitrogen oxide emissions from motor vehicles at national, state, and air basin scales. *J. Geophys. Res.: Atmos.* 2012, DOI: 10.1029/2012JD018304.
- [19] Preble, C. V.; Dallmann, T. R.; Kreisberg, N. M.; Hering, S. V.; Harley, R. A.; Kirchstetter, T. W. Effects of Particle Filters and Selective Catalytic Reduction on Heavy-Duty Diesel Drayage Truck Emissions at the Port of Oakland. *Environ. Sci. Technol.* 2015, 49 (14), 8864–8871.
- [20] Haugen, M. J.; Bishop, G. A. Repeat Fuel Specific Emission Measurements on Two California Heavy-Duty Truck Fleets. *Environ. Sci. Technol.* 2017, 51 (7), 4100–4107.
- [21] Haugen, M. J.; Bishop, G. A. Long-Term Fuel-Specific NO_x and Particle Emission trends for In-Use Heavy-Duty Vehicles in California. *Environ. Sci. Technol.* 2018, 52 (10), 6070–6076.
- [22] DFRobot. (n.d.). Gravity: Analog Carbon Monoxide Sensor (MQ7) (SKU: SEN0132). Product Datasheet and User Manual. Retrieved from DFRobot Official Website.
- [23] DFRobot. (n.d.). Gravity: Electrochemical Carbon Monoxide Sensor (SEN0466). Technical Specifications and Application Notes. Retrieved from DFRobot Documentation Hub.
- [24] Espressif Systems. (2023). ESP32 Technical Reference Manual. Version 4.7. Espressif Systems.
- [25] International Organization for Standardization. (2020). *ISO 15031-5: Road Vehicles—Communication Between Vehicle and External Equipment for Emissions-Related Diagnostics*. ISO.
- [26] European Union. (2015). *Commission Regulation (EU) 2016/646: Euro 6 Emission Standards for Light Passenger and Commercial Vehicles*. Official Journal of the European Union.
- [27] U.S. Environmental Protection Agency. (2021). EPA Tier 3 Vehicle Emissions and Fuel Standards Program. EPA Regulatory Guidelines.
- [28] International Electrotechnical Commission. (2016). CISPR 25: Vehicles, Boats, and Internal Combustion Engines—Radio Disturbance Characteristics for the Protection of Onboard Receivers. IEC.

- [29] FreeRTOS. (2023). FreeRTOS Real-Time Kernel Configuration Guide. Amazon Web Services Documentation.
- [30] Espressif Systems. (2023). ESP-IDF Programming Guide. Espressif IoT Development Framework Documentation.
- [31] DFRobot. (n.d.). *Gravity: I2C ADS1115 16-Bit ADC Module (SKU: DFR0553)*. Integration Guide for Analog Sensors. Retrieved from DFRobot Wiki.
- [32] Texas Instruments. (2018). *ADS1115 Ultra-Small, Low-Power, 16-Bit Analog-to-Digital Converter Datasheet*. TI Literature Number: SBAS444D.
- [33] NXP Semiconductors. (2019). Introduction to CAN 2.0 Protocol and Controller Area Network Applications. AN00022 Application Note.
- [34] International Electrotechnical Commission. (2018). IEC 62443: Industrial Communication Networks—Network and System Security. IEC.

Annexure I

```
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <HTTPClient.h>

// Network credentials
const char* ssid = "Galaxy S20 FE 5G 92B3";
const char* password = "avpk9971";

// Server configuration
const char* serverName = "https://emission-monitor.vercel.app/api/metrics";
String apiKeyValue = "tPmAT5Ab3j7F9"; // Changed to const char*

// CO Sensor Configuration - Using GPIO3 (ADC1_CH4 on ESP32-C3)
const int CO_SENSOR_PIN = 3; // Safe analog input pin on C3
float baseline_voltage = 0.61;
float max_voltage = 3.0;

// WiFi settings
#define WIFI_TIMEOUT 15000 // 15 seconds connection timeout
#define RETRY_INTERVAL 30000 // 30 seconds between attempts

WiFiClientSecure client;
HTTPClient https;

void setup() {
  Serial.begin(115200);
  delay(500); // Stabilization delay

  // Configure ADC attenuation for 3.3V range
  analogSetAttenuation(ADC_11db);

  connectToWiFi();
}

void connectToWiFi() {
  if(WiFi.status() == WL_CONNECTED) return;

  Serial.print("Connecting to WiFi");
  WiFi.disconnect(true);
  WiFi.begin(ssid, password);

  unsigned long start = millis();
  while(WiFi.status() != WL_CONNECTED && millis() - start < WIFI_TIMEOUT) {
    delay(250);
    Serial.print(".");
  }
}
```

```

if(WiFi.status() == WL_CONNECTED) {
  Serial.println("\nConnected! IP: " + WiFi.localIP().toString());
} else {
  Serial.println("\nConnection failed");
  delay(RETRY_INTERVAL);
  ESP.restart();
}
}

float readCO() {
  // ESP32-C3 has single ADC peripheral, no need for WiFi disable
  int rawValue = analogRead(CO_SENSOR_PIN);
  float voltage = rawValue * (3.3 / 4095.0);
  return fmaxf((voltage - baseline_voltage) * (1000.0 / (max_voltage - baseline_voltage)),
0.0);
}

void sendData(float co_ppm) {

  client.setInsecure(); // Remove for production

  // Build POST data safely
  String httpRequestData = "api_key=" + apiKeyValue + "&value=" + String(co_ppm);
  Serial.println("httpRequestData: " + httpRequestData);

  if(https.begin(client, serverName)) {
    https.addHeader("Content-Type", "application/x-www-form-urlencoded");

    int httpCode = https.POST(httpRequestData);
    if(httpCode > 0) {
      Serial.printf("HTTP Code: %d\n", httpCode);
    } else {
      Serial.printf("Error: %s\n", https.errorToString(httpCode).c_str());
    }
  }

  // https.end();
}
//client.stop();
}
static unsigned long lastSend = 0;

// Define variables for timing
unsigned long previousMillis = 0;
const long interval = 1000; // Interval in milliseconds (1 second)

// Array to store high CO values
const int maxReadings = 100; // Maximum number of readings to store
float coValues[maxReadings];
float threshold = 1000.0; // Threshold for CO values
int coCount = 0; // Counter for stored values

```

```

void loop() {
  unsigned long currentMillis = millis();

  // Check if 1 second has passed
  if (currentMillis - previousMillis >= interval) {
    previousMillis = currentMillis;

    float co_ppm = readCO();

    if (co_ppm > threshold) {
      if (coCount < maxReadings) {
        coValues[coCount] = co_ppm;
        // Serial.print("Added value:");
        // Serial.println(coValues[coCount]);
        coCount++;
      } else {
        Serial.println("Array full! Cannot store more values. Miliseconds: " +
String(currentMillis));
      }
    }
    // Check if 30 second has passed
    if(currentMillis - lastSend >=30000){
      lastSend = currentMillis;
      if(coCount > 0){
        Serial.println("Sending data to server... Miliseconds: " + String(currentMillis));
        Serial.println("Sending "+ String(coCount) +" values");

        for(int i = 0; i < coCount; i++){
          // sendData(coValues[i]);
          Serial.println("Sent value: " + String(coValues[i]));
        }
        coCount = 0;
        // Reset the array

      }
      else{
        Serial.println("No values above threshold (1000 ppm) were recorded in last 30
seconds. Miliseconds: " + String(currentMillis));
      }
    }
  }
}

void loop() {

  if(millis() - lastSend >= 10000) {

```

```
if(WiFi.status() != WL_CONNECTED) {
  connectToWiFi();
}

if(WiFi.status() == WL_CONNECTED) {
  float co_ppm = readCO();
  Serial.println(co_ppm);
  sendData(co_ppm);
  lastSend = millis();
}
}
}
```

Annexure II

```
6. import { NextResponse } from "next/server";
7. import mysql from "mysql2/promise";
8.
9. // Shared connection configuration
10. const dbConfig = {
11.   host: process.env.DB_HOST,
12.   port: process.env.DB_PORT,
13.   user: process.env.DB_USER,
14.   password: process.env.DB_PASSWORD,
15.   database: process.env.DB_NAME,
16.   ssl: {
17.     rejectUnauthorized: true,
18.     ca: `-----BEGIN CERTIFICATE-----
19.
20.     MIIEQTCCAqmgAwIBAgIUFP4vR7EXMur/3tnlCtX1NiOLAtUwDQYJKoZIhvcN
21.     AQEM
22.     BQAwojE4MDYGA1UEAwvZDQ0NmVIZGltOTVjMC00N2ExLTg3OGQtY2Iy
23.     NzNIYjEy
24.     NDU0IFByb2plY3QgQ0EwHhcNMjQxMjEwMDYzMDYzMDYzMDYzMDYzMDYz
25.     MzQ2WjA6
26.     MTgwNgYDVQQDDC9kNDQ2ZWVhYi05NWMwLTQ3YTEtODc4ZC1jYjI3M2V
27.     iMTI0NTQg
28.     UHJvamVjdCBDQTCCAAIwDQYJKoZIhvcNAQEBBQADggGPADCCAYoCggGB
29.     AMx1QoF3
30.     ZzRDwOahTuEqR1+OwBmp8zxPOiaDThjJXCMkQjqdeRiAaCbhCZU4wv9uppA2I
31.     9mj
32.     Ksa/qVBad+pR4X9ZXuw5KcPxLSpY9ZPhVN0qb9U5FNbLIs3C329CtqFx6m84Lq
33.     WY
34.     pqbpRyI/GYA4a3KWOEM7AGok52wWEyJM67pzyHra520Cq+N0NFuuhLkeD6C5
35.     caSz
36.     jG6FLMmQ4tD8rwu/nrFrcX+PKkeDDPpizfh1JZpIdp393tKFFTEqsNBbh60mnrbv
37.     j2GeU05TG/G5Uliqt9kTjT2HHhSy/7/+tkyOBRBF9w2JkMi0+TcecVFzjfKAsq2o
38.     29.
39.     p8530ae/eFKcDBdh/oNpPVepai3AFX7PCxIFQ5xo/Fqn0hu+NG9UD4jeDb8MAOw
40.     6
41.     MtQwiaQvLDV0iViiCodDTNQ1IErW/ETuN+r+s5BM4ne3JmKi+ojtat+ogyJ0QHfT
42.     31.
43.     9PaV9utyItmS281/iB76y11o8qJqmz3SMRf1OwYqciPINUPBoHuwZqF7YQIDAQA
44.     B
```

```

32.      oz8wPTAdBgNVHQ4EFgQU45cMof27kFrK28s1D2MdUqFzZTAwDwYDVR0TB
AgwBgEB
33.      /wIBADALBgNVHQ8EBAMCAQYwDQYJKoZIhvcNAQEMBAQADggGBAK8ON
1/EotSxh0p9
34.      ch2zscbtD/e26eECMqNfSEbQroYE2BHxlZahOcf0wkJerCCtrcWRUTJfClz
35.      PZSaGjH3Ql+4akHYg8IWfLI/XJzN5EMDHGb12OYVeSnqEytOYH7gXDUQH1Jt
T9px
36.      YYk1sApdEbuurfoYBtFtU0/BNPTA+qnpnC84gBBrn7vD2x2cTFe+tsVebjsCh/IV
37.      xVRViOyZl3lX2aZHTEq/XuAteb8RILsqeDCqhcK04oH7ntj3bYe3P1EAyA2CYWP
A
38.      CGDnKbpbADW7RAREPrOOPSDOBFw4GX1sOrjHxWou9B+5mr5I5Ygt9YIA5la
MVI8x
39.      +bW9BfE27OCfAKdD2vAAAp6l7I7Ilz0w7LzLu2q0Xs/sIoL7yCgS3onjvX0EoWAp
40.      vbdXqzmWXoO3ZgnxUyk7lxqlb7DNGKIS6qy9JcJY7q5DzT6b0BkETL8IVI0eJYA
U
41.      Cq1EttvDL29pSNvI5VSgyaGMTLZE6SL+NU+66AwTzIEN4YSLmA==
42.      -----END CERTIFICATE-----`,
43.      },
44.      connectTimeout: 10000, // 10 seconds timeout
45.      };
46.
47.      export async function POST(request) {
48.      let connection;
49.      try {
50.      // Parse URL-encoded form data instead of JSON
51.      const formData = await request.formData();
52.      const data = {
53.      api_key: formData.get("api_key"),
54.      value: parseFloat(formData.get("value")),
55.      };
56.
57.      // Validate API Key
58.      if (data.api_key !== process.env.API_KEY) {
59.      return NextResponse.json({ error: "Unauthorized" }, { status: 401 });
60.      }
61.
62.      // Validate value
63.      if (isNaN(data.value)) {
64.      return NextResponse.json(
65.      { error: "Invalid value - must be a number" },
66.      { status: 400 }
67.      );
68.      }
69.

```

```

70.   const timestamp = data.timestamp || new Date().toISOString();
71.   console.log("Timestamp:", timestamp);
72.   connection = await mysql.createConnection(dbConfig);
73.
74.   const [result] = await connection.execute(
75.     "INSERT INTO metrics (value) VALUES (?)",
76.     [data.value]
77.   );
78.
79.   return NextResponse.json(
80.     {
81.       id: result.insertId,
82.       message: "Metric added successfully",
83.     },
84.     { status: 201 }
85.   );
86.   } catch (error) {
87.     console.error("POST error:", error);
88.     return NextResponse.json(
89.       { error: "Failed to create metric" },
90.       { status: 500 }
91.     );
92.   } finally {
93.     if (connection) await connection.end();
94.   }
95. }
96.
97. export async function GET(request) {
98.   let connection;
99.   try {
100.    connection = await mysql.createConnection(dbConfig);
101.
102.    // Extract the range query parameter
103.    const { searchParams } = new URL(request.url);
104.    const range = searchParams.get("range");
105.
106.    let startDate;
107.    const now = new Date(); // Current time in UTC
108.
109.    // Calculate start date based on the selected range
110.    switch (range) {
111.      case "1h":
112.        startDate = new Date(now.getTime() - 60 * 60 * 1000); // 1 hour ago
113.        break;
114.      case "5h":
115.        startDate = new Date(now.getTime() - 5 * 60 * 60 * 1000); // 5 hours ago
116.        break;
117.      case "24h":
118.        startDate = new Date(now.getTime() - 24 * 60 * 60 * 1000); // 24 hours ago
119.        break;

```

```

120. default:
121. startDate = null; // No range selected
122. }
123.
124. // Build the SQL query based on the presence of a range
125. let query = "SELECT id, value, timestamp FROM metrics";
126. const params = [];
127. if (startDate) {
128. query += " WHERE timestamp >= ?";
129. params.push(startDate);
130. }
131. query += " ORDER BY timestamp ASC";
132.
133. // Execute the query with parameters
134. const [rows] = await connection.execute(query, params);
135.
136. // Format the timestamp to ISO string
137. const formattedData = rows.map((row) => ({
138. ...row,
139. timestamp: new Date(row.timestamp).toISOString(),
140. }));
141.
142. return NextResponse.json(formattedData);
143. } catch (error) {
144. console.error("Database error:", error);
145. return NextResponse.json(
146. { error: "Failed to fetch metrics" },
147. { status: 500 }
148. );
149. } finally {
150. if (connection) await connection.end();
151. }
152. }
153.
154.
155. Components/MetricChart.jsx:
156. "use client";
157.
158. import { useState, useEffect } from "react";
159. import {
160. LineChart,
161. Line,
162. XAxis,
163. YAxis,
164. CartesianGrid,
165. Tooltip,
166. ResponsiveContainer,
167. } from "recharts";
168.
169. export default function MetricsChart() {

```

```

170. const [metrics, setMetrics] = useState([]);
171. const [loading, setLoading] = useState(true);
172. const [error, setError] = useState(null);
173. const [selectedRange, setSelectedRange] = useState("1h"); // State for selected time
range
174. const [refresh, setRefresh] = useState(1); // State for selected time range
175.
176. useEffect(() => {
177.   const fetchMetrics = async () => {
178.     try {
179.       const url = `/api/metrics${
180.         selectedRange ? `?range=${selectedRange}` : ""
181.       }`;
182.       const response = await fetch(url);
183.       if (!response.ok)
184.         throw new Error(`HTTP error! status: ${response.status}`);
185.       const data = await response.json();
186.       setMetrics(data);
187.     } catch (err) {
188.       setError(err.message);
189.     } finally {
190.       setLoading(false);
191.     }
192.   };
193.
194.   fetchMetrics();
195. }, [selectedRange, refresh]);
196.
197. if (loading) {
198.   return <div className="p-4 text-gray-600">Loading metrics...</div>;
199. }
200.
201. if (error) {
202.   return <div className="p-4 text-red-500">Error: {error}</div>;
203. }
204.
205. return (
206.   <div className="py-4 px-2">
207.     <h2 className="text-xl font-bold mb-4">Emission Metrics</h2>
208.     <div className="h-96 w-full">
209.       {metrics.length === 0 ? (
210.         <div className="p-4 text-center text-gray-500">No metrics found</div>
211.       ) : (
212.         <ResponsiveContainer width="100%" height="100%">
213.           <LineChart
214.             data={metrics}
215.             margin={{ top: 20, right: 0, left: 0, bottom: 20 }}
216.           >
217.             <CartesianGrid strokeDasharray="3 3" />
218.             <XAxis

```

```

219.   dataKey="timestamp"
220.   tickFormatter={(timeStr) =>
221.     new Date(timeStr).toLocaleDateString()
222.   }
223.   label={{
224.     value: "Timestamp",
225.     position: "bottom",
226.     offset: 5,
227.   }}
228. />
229. <YAxis
230.   label={{
231.     value: "Emission (ppm)",
232.     angle: -90,
233.     position: "left",
234.     offset: -6,
235.   }}
236.   tickFormatter={(value) => `${value.toFixed(0)} `}
237. />
238. <Tooltip
239.   contentStyle={{
240.     backgroundColor: "#fff",
241.     border: "1px solid #ddd",
242.     borderRadius: "4px",
243.     color: "#2865eb",
244.   }}
245.   formatter={(value) => [`${value.toFixed(2)} ppm`, "Value"]}
246.   labelFormatter={(label) => new Date(label).toLocaleString()}
247. />
248. <Line
249.   type="monotone"
250.   dataKey="value"
251.   stroke="#2563eb"
252.   strokeWidth={2}
253.   dot={{ r: 4 }}
254.   activeDot={{ r: 8 }}
255. />
256. </LineChart>
257. </ResponsiveContainer>
258. )}
259. </div>
260. <hr />
261. <div>
262.   { /* Add buttons to select time range */}
263.   <div className="w-full flex justify-center items-center gap-5 my-4">
264.     <button
265.       className={` ${
266.         selectedRange == "1h" ? "bg-slate-500" : "bg-slate-800"
267.       } cursor-pointer hover:bg-slate-700 p-2 border-1 rounded-xl`}
268.       onClick={() => setSelectedRange("1h")}

```

```

269. >
270. 1 Hour
271. </button>
272. <button
273.   className={` ${
274.     selectedRange == "5h" ? "bg-slate-500" : "bg-slate-800"
275.   } cursor-pointer hover:bg-slate-700 p-2 border-1 rounded-xl` }
276.   onClick={() => setSelectedRange("5h")}
277. >
278. 5 Hours
279. </button>
280. <button
281.   className={` ${
282.     selectedRange == "24h" ? "bg-slate-500" : "bg-slate-800"
283.   } cursor-pointer hover:bg-slate-700 p-2 border-1 rounded-xl` }
284.   onClick={() => setSelectedRange("24h")}
285. >
286. 1 Day
287. </button>
288. <button
289.   className={` ${
290.     selectedRange == "" ? "bg-slate-500" : "bg-slate-800"
291.   } cursor-pointer hover:bg-slate-700 p-2 border-1 rounded-xl` }
292.   onClick={() => setSelectedRange("")}
293. >
294. All Time
295. </button>
296. </div>
297. <div className="w-full flex justify-center items-center gap-5 my-4">
298. <button
299.   className={`bg-blue-900 cursor-pointer hover:bg-slate-700 px-5 py-7 border-1
rounded-full` }
300.   onClick={() => setRefresh((prev) => prev + 1)}
301. >
302. Reset
303. </button>
304. </div>
305. { /* Rest of your chart rendering logic */ }
306. </div>
307. </div>
308. );
309. }
310.
311.
312. App/page.js (home)
313. import Image from "next/image";
314. import Metrics from "@components/Metrics";
315. import MetricForm from "@components/MetricForm";
316. import MetricsChart from "@components/MetricChart";
317.

```

```
318. export default function Home() {
319.   return (
320.     <div>
321.       {/* Display Chart component here*/}
322.       <MetricsChart />
323.     </div>
324.   );
325. }
```

List of Figures

S. No.	Table Captions	Page
4.5 (a)	Graphical interface for the front end of the application (user interface)	47
4.5 (b)	Graphical interface for the front end of the application (user interface)	47
4.5 (c)	Graphical interface for the front end of the application (user interface)	47

Nomenclature

Roman Symbols

A_o	Cross-sectional area of orifice <Times new roman 12 regular>
B	Breadth of orifice
C_1	Shape constant of orifice
D	Diameter of orifice
dA	Elemental area
d_a	Particle aerodynamic diameter
f_B	Braggs frequency
f_D	Doppler frequency
f_R	Resultant frequency
F_D	Focal length of lens
H	Height of synthetic jet cavity
H	Thickness of orifice
H_B	Separation distance between the counter rotating vortices in LIF images in major axis plane of orifice
H_D	Separation distance between the counter rotating vortices in LIF images
H_W	Separation distance between the counter rotating vortices in LIF images in minor axis plane of orifice
I	Ampere given to hot-film
l_m	Length of oval shape measuring volume
L	Equivalent slug length of ejected fluid in expulsion stroke of diaphragm