Grenze International Journal of Engineering and Technology, June Issue

Real-Time IoT-based Air Quality Monitoring System using ESP32 and Cloud Integration

Ashwini Navghane[1], Tapsvi Solanke[2], Shravani Shalrushi[3], Sarthak Barhate[4], Deep Shirbhate[5] and Shiwansh Kukde[6]
[1]Department of Engineering Sciences and Humanities, Vishwakarma Institute of Technology, Pune, India
Email: ashwini.navghane@vit.edu
[2-6]Department of Electronics and Telecommunication Engineering, Vishwakarma Institute of Technology, Pune, India
Email: tapsvi.solanke24@vit.edu, shravani.shalrushi24@vit.edu, sarthak.barhate24@vit.edu, deep.shirbhate24@vit.edu, shiwansh.kukde24@vit.edu

***Abstract*— Atmospheric pollution in cities is a public health hazard that necessitates innovative monitoring technologies beyond standard station systems. In this study, we present an inexpensive air quality monitoring system based on the IoT approach, utilizing the ESP32 microcontroller. The system comprises MQ-135 gas sensors, MQ-9 carbon monoxide sensors, GP2Y1010AU0F optical dust sensor for PM2.5 measurement, and DHT22 temperature/humidity sensor. Using the dual-core ESP32 microcon-troller, we perform efficient data processing and real-time cloud transmission. A mobile application provides users with up-to-date AQI readings. Experimental results prove the system functions without significant delay.**

***Index Terms*— Air Quality Monitoring System, IoT, ESP32, Air Pollution, PM2.5, Carbon Monoxide, Gas Sensors, Data Acquisition, AQI, Real-Time Monitoring.**

I. INTRODUCTION

The swift pace of global industrialization growth has triggered an unprecedented drop in the quality of the atmosphere. Air pollution has outgrown its role as an environmental peripheral issue to realize its status as a central public health imperative. Despite the severity of the aforementioned risks, modern environmental monitoring is largely dependent on government-operated stations. Although the equipment used in the respective stations is highly precise, the cost associated with the same is rather high, ranging between tens of thousands of dollars per station. This has resulted in low spatial densities, where many areas are considered "data deserts," i.e., the pollution level on a particular street corner is unquantified Given this variability in concentration levels of pollutants within a short distance of hundreds of meters because of urban canyons and other factors, these stations fail to provide the hyperlocal telemetry data necessary for the effective management of personal health.
The arrival of the Internet of Things (IoT) revolutionized the concept of spatial resolution, where it is now possible to deploy dense, distributed sensor networks that offer rich, real-time environmental intelligence. Although early attempts were based on IoT, they were based on 8-bit microcontrollers, such as Arduino, where computational bottlenecks were common in a scenario where the platform had to perform many tasks concurrently, such as sensor sampling, signal processing, and wireless communication.

Grenze ID: 01.GIJET.12.2.133_26

Many IoT-based platforms have concentrated on the monitoring of gaseous pollutants, whereas the effects of airborne dust and fine particulates are significant in many environments. The main contribution of this research is the proposal of a comprehensive, high-performance-based monitoring solution, including the ESP32-based microcontroller, as shown in the below figure. The ESP32 has been chosen as it is a dual-core, 32-bit processor, where the two cores can perform independent functions, such as one handling high-frequency sensor sampling and the other handling the Wi-Fi stack and cloud synchronization.

The proposed system herein is comprised of a heterogeneous set of sensors that is capable of monitoring the most important pollutants that are urban-related. Designations of the incorporated sensors include the following: the MQ-135 for hazardous gases, including NH3 and Benzene; the MQ-9 for Carbon Monoxide and combustible gases; and the GP2Y1010 for quantification of PM2.5, as well as dust. Unlike other standard systems available, our system incorporates a digital temperature and humidity sensor that incorporates various compensation factors on the chemical detection.

The telemetry pipeline is achieved via the cloud-based ThingSpeak platform, enabling real-time data storage, visualization, and historical trend analysis. The custom-engineered application plays the role of the primary user interface and converts raw, non-linear voltages into a standardized air quality index (AQI). This way, users can be alerted instantly via their personal mobile phones by receiving notifications.

The key contribution of this research can be attributed to the design and implementation of an inexpensive and scalable IoT air quality monitoring solution based on the ESP32 controller. In order to improve the performance of the system, the dual-core architecture of ESP32 has been exploited, where one of the cores performs data acquisition from the connected sensors and the other takes care of cloud connectivity, thus allowing for efficient parallel processing. In addition, multiple types of sensors have been employed to give a comprehensive environmental picture by integrating gas sensors and PM sensors. Besides, the measurement of temperature and humidity allows for performing environmental compensation, which enhances the performance of the gas sensors.

The proposed work presents a low-cost real-time air quality monitoring architecture using the ESP32 dual-core microcontroller to efficiently perform parallel processing of sensing and communication tasks. The multi-sensor fusion approach is implemented with the MQ-135, MQ-9, GP2Y1010AU0F, and DHT22 sensors for a comprehensive environmental monitoring. The system includes a light-weight cloud-based monitoring framework based on the ThingSpeak platform for real-time data visualization and remote access. In addition, environmental compensation by means of temperature and humidity sensing has been used to enhance the stability and reliability of the gas sensor. A mobile based alert mechanism has also been developed to provide real time AQI notifications and to raise awareness on pollution. Experimental evaluation performed during 20 days shows stable operation with low cloud transmission latency and reliable monitoring performance.

II. Literature Review

The quick growth of urban areas and industries has resulted in rising pollution levels globally. Polluted air and water impact human well-being, weather changes, and the standard of living. In order to combat these problems, IoT-based environmental monitoring systems with the use of sensors, wireless connectivity, and cloud computing are being studied to perform an analysis of the environment in real time [1].

Many IoT air quality monitoring systems for indoors and outdoors were suggested by researchers using sensors to measure toxic gases, humidity, temperature, and particulates, while remote real-time visualizations can be carried out via a cloud computing platform [2]. A low-cost microcontroller such as ESP32 or Arduino is frequently employed in IoT environmental monitoring systems because of its low power requirements, inexpensive cost, and ability to connect with many sensors at once [3].

Wireless protocols such as Wi-Fi and MQTT provide an essential means of transmitting data in real-time and remote monitoring of IoT-based systems. Web dashboards and alerts have been designed by researchers to ensure that the user gets notified whenever the pollution threshold exceeds [4]. The monitoring systems based on IoT are further extended for applications in water quality and weather monitoring [5].

New advances in machine learning and edge computing technologies have enhanced the performance of the environmental monitoring systems.

Machine learning helps in forecasting the pollution levels, whereas edge computing minimizes the communication delay due to local processing of the collected data [6]. Even after these advancements, many challenges arise within IoT-based environmental monitoring systems in terms of sensor miscalibration, accuracy deterioration, power requirements, security problems, and scalability [7].

III. METHODOLOGY

A. System Architecture

Table I presents the components used in the proposed system along with their respective functions.

TABLE I: COMPONENTS USED AND THEIR PURPOSES

Components	Purpose
ESP32	Main controller and Wi-Fi gateway
GP2Y1010AU0F	Dust/particulate matter detection
MQ-135	Air quality and hazardous gas monitoring
MQ-9	CO and combustible gas detection
DHT22	Environmental compensation
Cloud platform	Data storage and remote monitoring
Mobile application	Real-time monitoring and user alerts

The proposed system outlines a low-cost IoT-based air quality monitoring system architecture that can effectively measure various environmental parameters and transfer the measured data to a cloud for real-time analysis.

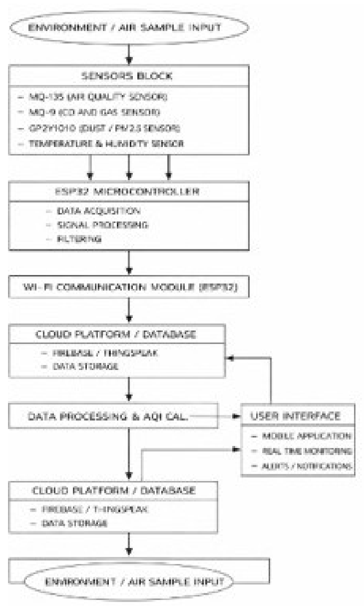

Figure 1: IoT based pollution monitoring system utilizing ESP32 and cloud integration block diagram

The system framework is illustrated in Fig.1. The overall workflow of the system is based on various steps related to environmental sensors, signal processing, wireless communication, cloud storage, and data visualization for users.

The pollution monitoring system is designed using an ESP32 microcontroller as a primary component for processing and communicating various data points. The ESP32 is chosen for its dual-core processing capabilities and integration of Wi-Fi functionality for IoT-based applications.

The system utilizes various environmental sensors to measure different parameters of air pollution. The sensors used are MQ-135 gas sensors for detecting hazardous gases, MQ-9 sensors for detecting CO and combustible gases, GP2Y1010 optical sensors for measuring PM2.5, and a temperature and humidity sensor DHT22.

The ESP32 collects the analog and digital signals sent by all the sensors, then it filters and preprocesses the data. In this regard, the core one processes all sensor data while the core two processes Wi-Fi and communication to the cloud.

The proposed air quality monitoring system based on IoT will be able to monitor the level of environmental pollution in real-time. Various types of sensors like MQ-135, MQ-9, GP2Y1010AU0F, and DHT22 will be employed for the detection of harmful gases, carbon monoxide, particulate matter, temperature, and humidity respectively. The ESP32 microcontroller will act as the central processing unit in which it gathers all the data collected by the attached sensors. After that, the processed data is further processed into useful air quality data using the AQI formula. Then the data collected by the ESP32 controller will be transmitted to the ThingSpeak cloud computing system via Wi-Fi transmission. The cloud computing system stores the environmental data and makes its graphical presentation for analysis. The pollution levels can be analyzed via a web dashboard and mobile application. Whenever the pollutants cross the set safety limit, an alert will be sent via the mobile application.

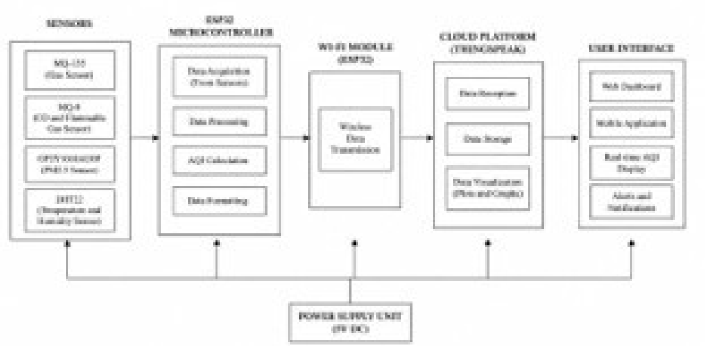

Figure 2: Proposed IoT-Based Air Quality Monitoring System Architecture Using ESP32 and Cloud Integration

B. Sensor Implementation

The multi-sensor technique has been utilized to ensure a comprehensive assessment of air quality. Each of the sensors has been designed to monitor various environmental parameters responsible for air pollution.
MQ-135 Gas Sensor has also been used for the detection of various harmful gases including ammonia, benzene, nitrogen oxide, and carbon dioxide. The MQ-135 sensor functions based on the sensing changes in resistance due to exposure to different types of gases in the atmosphere.
The MQ-9 sensor has been utilized to detect carbon monoxide and other combustible gases present in the environment. This sensor generates voltage proportional to the detected gases in the environment. For detecting the level of particulate matter present in the environment, the GP2Y1010 optical dust sensor has been integrated into the proposed system. This sensor has been utilized to detect dust particles in the environment and calculate the level of PM2.5 present in the environment.
Furthermore, a digital temperature and humidity sensor has been integrated into the proposed system to monitor various environmental parameters, including temperature and humidity, since these factors have an impact on gas sensors. The data from this sensor helps in enhancing the accuracy of gas detection.

C. Sensor Calibration

Before being deployed, all the sensors were calibrated to enhance the accuracy and stability of the proposed system. The MQ-135 and MQ-9 gas sensors were preheated at normal atmospheric conditions to have a stable baseline. The sensor outputs were calibrated using the resistance values and datasheet characteristics for the gas concentration estimation.
The GP2Y1010AU0F dust sensor was calibrated in different dust conditions . A moving average filter was used to remove the noise and fluctuations of the PM2.5 measurements. Moreover, the temperature and humidity sensor DHT22 was used in order to compensate the environment, since the performance of the gas sensor is affected by the local atmospheric conditions. The ESP32 microcontroller was used to average multiple sensor readings, which helped to increase measurement consistency and improve the overall system reliability during long-term monitoring.

D. Data Acquisition and Processing

The ESP32 microcontroller undertakes periodic data acquisition from all the sensors. The analog data from the MQ-135, MQ-9, and GP2Y1010 sensors are converted to digital form using the in-built Analog to Digital Converter of the ESP32 microcontroller.
The data from the sensors is processed using various filtering and calibration methods to obtain accurate results. The environmental temperature and humidity are also considered during this stage to compensate for fluctuations in the functioning of the gas sensors. This output data from the sensors is transformed to get the concentration of the pollutants. The AQI is determined using this concentration value.

Air Quality Index is a measure that determines the quality of the air. The AQI is calculated using a linear interpolation technique that is based on the concentration values of pollutants and is expressed as:

$$I = \frac{(I_{high} - I_{low})}{(C_{high} - C_{low})} \times (C - C_{low}) + I_{low} \quad (1)$$

Where C denotes the concentration of pollutants, C_{low} and C_{high} denote the breakpoint concentrations, while I_{low} and I_{high} refer to the index values of AQI.

The AQI formula transforms the pollutant concentration readings to a standard scale representing the amount of pollution and possible effects on human health. Normally, AQI is calculated independently for each pollutant and then the total AQI is derived based on the maximum sub-index.

In this study, the concentration reading C represents the level of specific pollutants like PM2.5 and CO measured using the corresponding sensors. The PM2.5 reading serves as an important metric because of the health risks associated with it.

The AQI formula is presented here for reference purposes and to provide context on interpreting the level of air quality, whereas the experimental results mainly include the sensor data.

Table II presents the standard AQI classification used to interpret air quality levels.

TABLE II: NATIONAL AIR QUALITY INDEX (NAQI) CLASSIFICATION

AQI value	Category
0-50	Good
51-100	Satisfactory
101-200	Moderate
201-300	Poor
301-400	Very Poor
401-500	Severe

E. Cloud Communication and Data Storage

The cloud communication offers a robust bridge between the sensor network and a centralized data repository. Utilizing the ThingSpeak IoT platform, a specialized Time-Series Database (TSDB) has been created to offer optimal functionality for high-frequency environmental logging. Data ingestion occurs through a Hypertext Transfer Protocol (HTTP) request using a RESTful API framework, where the microcontroller aggregates a GET request for multiple sensors within a single request. A deterministic 20-second sampling period has been incorporated to ensure system stability and cloud ingestion rates. Data packets are authenticated using a API Key.

On the backend side, a non-relational data storage model is used, where a timestamp in ISO 8601 format is automatically generated for each piece of data as it is received. This structured data is then mapped to distinct fields for the MQ-135, MQ-9, and GP2Y1010AU0F sensors, as well as the processed result for the Air Quality Index. The visualization engine on the cloud side uses a moving average filter on this data to reduce high-frequency noise from the electrochemical sensors. This allows for a constant temporal plot to be made from the data, necessary for longitudinal analysis of pollution.

F. Remote Alerting and Data Visualization

The proposed system will make use of a software layer that is specifically geared for data visualization and risk assessment. The remote interface will essentially act as a dedicated observation node that will monitor the quality of the atmosphere using an asynchronous polling mechanism. Instead of using a static display system, the system will periodically fetch serialized JSON data packets directly from the ThingSpeak API. This will ensure that the actual telemetry collected by the ESP32 Arduino hardware is displayed to the user.

To make the collected data easier for the user to interpret, the application will make use of a comparative algorithm that will essentially categorize the collected raw sensory inputs under different Air Quality Index classifications. Through the measurement of the concentrations of these pollutants such as CO, NH3, and PM2.5 relative to the NAQI index, the system will automatically categorize the quality of the environment into categories such as Good, Satisfactory, Moderate,Poor,Very poor or Severe. This will allow the user to monitor pollution levels within the city through remote means.

The system also features an event-based notification feature, which triggers notifications based on the crossing of a certain threshold of the pollutant If the concentration levels exceed a predefined threshold, the interface runs a high-priority interrupt to immediately send a push notification. This feature is useful because it notifies users of the adverse health impacts and provides a preventive measure against the adverse health impacts through immediate action. The system framework uses a multi-stage methodology that is useful in providing a robust system for environmental awareness through the use of data acquisition, cloud processing, and telemetry.

IV. Results and Discussions

The proposed IoT-based system for air quality monitoring was designed and implemented using the ESP32 microcontroller in combination with various environmental sensors. The designed system was tested in several environmental settings to check its sensing capabilities, cloud-based communication efficiency, monitoring effectiveness, and reliability.

This designed system performed the continuous monitoring of essential environmental factors such as concentration of hazardous gases, amount of carbon monoxide, particulate matter (PM2.5), temperature, and humidity. The sensed data from MQ-135, MQ-9, and GP2Y1010 sensors were transferred to the ThingSpeak cloud server using the Wi-Fi module included in ESP32 controller. Cloud based dashboard provided real-time viewing of sensed environmental data.

The results of experimentation show that the system under consideration worked flawlessly in terms of measuring environ-mental parameters for 20 consecutive days without significant interruptions. It was established that the average value of cloud transmission delay in experimental conditions was 1.9 seconds with the minimum value being 1.2 seconds and the maximum value being 2.8 seconds. ESP32 dual-core technology made it possible to acquire data from sensors and transmit this data to the cloud simultaneously without any delays.

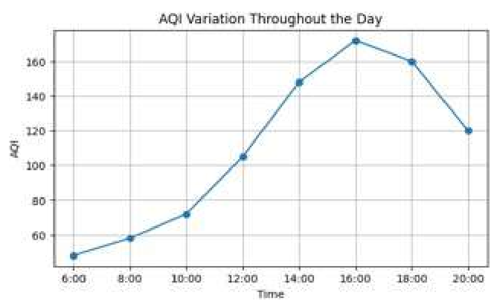

Figure 3: AQI variation throughout the day based on sensor measurements

Fig. 3 illustrates the fluctuations of AQI values during the whole process of observation. It was noted that AQI values fluctuated to an extent where the values showed a sharp rise in the afternoon and evening times owing to increased traffic and industry pollution. At low traffic intervals, AQI values were confined to the “Satisfactory” range; however, with high traffic levels, there was a marked increase in the AQI by 35–40

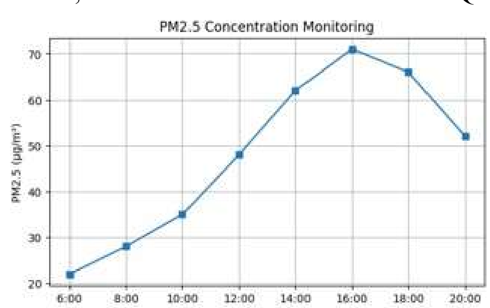

Figure 4: PM2.5 concentration monitoring using GP2Y1010 optical dust sensor

Fig. 4 represents the PM2.5 concentrations obtained through the use of the GP2Y1010AU0F optical dust sensor. It was noted from experiments that the PM2.5 concentrations were very high in areas where there were more dust particles and heavy traffic conditions. The observed range for the measured PM2.5 concentration was 35 μg/m³ to 92 μg/m³. There was no variation in the readings as the system employed the moving average filtering method.

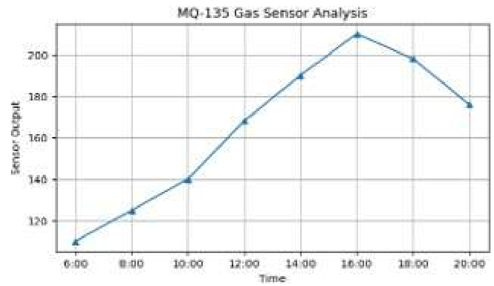

Figure 5: MQ-135 sensor response for hazardous gas concentration analysis

The reaction behavior of the MQ-135 gas sensor at varying atmospheric conditions is illustrated in Fig. 5. The output voltage level of the sensor was observed to rise significantly in areas where there was heavy vehicular and industrial pollution, showing that there were high levels of harmful gases like ammonia, nitrous oxides, and benzene compounds. From experimental analysis, an output level increment of around 30-45

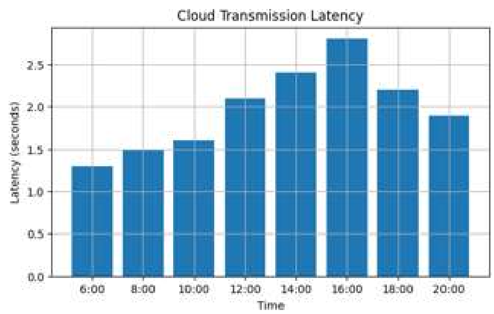

Figure 6: Cloud transmission latency during real-time environmental monitoring

Fig. 6 shows the latency period experienced by the cloud during the real-time monitoring process. The latency period for all

the tests was within the range of 1.2 to 2.8 seconds. The small latency period is an indication that the ESP32 microcontroller and the ThingSpeak Cloud can be used to implement effective real-time environmental monitoring applications.

Table III provides an overall performance evaluation of the proposed IoT-based air quality monitoring system in terms of sensing capability, communication efficiency, and operational reliability.

TABLE III. OVERALL PERFORMANCE EVALUATION OF PROPOSED IOT-BASED AIR QUALITY MONITORING SYSTEM

Parameter	Observation
PM2.5 Detection Accuracy	±10 µg/m³
MQ-135 Sensitivity	Effective detection of NH3, Benzene, NOx
MQ-9 Performance	Stable CO detection
Cloud Transmission Delay	1.2 – 2.8 seconds
Sampling Interval	20 seconds
System Reliability	Stable operation for 20 days
Wi-Fi Connectivity	Continuous without major interruption

The prototype was tested regularly for 20 days to study the stability of the system in terms of its functionality and sensors' accuracy over time. There were no complaints about any problem of the functioning of ESP32 board; it communicated properly and did not face any issues during the process of data transmission or processing. Dual-core processors of ESP32 board were able to process data from the sensors and transmit it to the cloud simultaneously without facing any processing delay.

The results obtained through experimentation show that the suggested system is an efficient and economical solution for real-time air pollution monitoring. The use of several sensors allowed us to monitor gas pollution, particulate pollution, air temperature, and humidity simultaneously. Environmental compensation provided by the use of DHT22 sensors made the gas sensor values stable under changing environmental conditions. Also, moving average filtering ensured that the data received from the sensors was not affected by any sudden changes or noise.

Figure 7: Mobile Application Interface for Real-Time Air Quality Monitoring and Alert System

Fig. 7 shows the mobile application interface developed. It is user-friendly and facilitates real-time monitoring of environmen-tal conditions. The parameters that will be shown include AQI value, PM2.5 concentration, carbon monoxide concentrations, temperature, and humidity recorded by the IoT sensors. Various screens have been developed for live monitoring, AQI value analysis, visualization of trends over time, and alerts. The dashboard screen will enable users to visually see the pollution level from the use of various graphical representations and also classification of AQI values. The history feature enables users to get trend analysis based on the graphical plot that will be done. The alert feature will notify the users immediately the level of pollutants exceeds safety threshold values.

The mobile application created by was quite efficient in displaying the values of air quality index in real time and creating alerts whenever the concentration of pollutants exceeded a certain value. Moreover, mobile application proved to be helpful in making people aware of pollution and keeping them away from inhaling harmful pollutants in the air.

The results prove that the designed IoT-based pollution monitoring system is efficient, scalable, and economical.

The projected hardware cost for developing the prototype is around 1500 to 2000, making it considerably cheaper than regular air quality monitoring stations.

V. Conclusion

In this paper, the design and development of the IoT-based Pollution Monitoring System with Cloud Integration were presented. This system employs an ESP32 microcontroller with high-accuracy gas and particulate sensors, a cloud-based database, and an application for the monitoring of pollution exposure risk to individuals. The designed and implemented system proved its reliable functioning for detecting pollutants, maintaining the integrity of information in the cloud and notifying users. This project presents a cost-effective way for monitoring localized air quality compared to expensive static monitoring systems used by governments. Based on a reliable IoT-based air quality monitoring system, the device recorded air quality through various means. The device was successful in collecting and sending all of the sensor data from the sensors to a cloud-based service with a low latency of 1.2 - 2.8 seconds.

The device utilized an ESP32 microcontroller to allow monitoring of various parameters that determine air quality such as the air quality index (AQI), PM2.5, CO, temperature and humidity. As a result, the system provided reliable and accurate results. The GP2Y1010AU0F was effective at being able to measure PM2.5, while the MQ-135 and MQ-9 provided very good data regarding the presence of harmful gases, CO.

Temperature and humidity compensated for sensor readings and ensured that sensors provided stable data. To compensate for sensor noise and fluctuations, sensor data went through a moving average filter, giving the sensor data accuracy. The complete system could run for 20 days without any interruption.

The future improvements of this system include adding modules for solar energy harvesting in order to guarantee power supply for the operation of the system. Another improvement that could be made in the future includes the addition of other sensors to measure noise and radiation exposure levels.

Acknowledgment

The authors would like to express their sincere gratitude to Prof. Ashwini Navghane, Department of Engineering Sciences and Humanities, Vishwakarma Institute of Technology, Pune, for her valuable guidance, continuous support, and encouragement throughout the development of this research work.

References

[1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.

[2] M. A. Al Mamun, M. Hannan, A. Hussain, and H. Basri, "Integrated sensing systems and algorithms for air quality monitoring using IoT," IEEE Sens. J., vol. 19, no. 11, 2019.

[3] S. R. Prabhu and V. K. Shanmuganathan, "IoT and machine learning based smart environment monitoring system," Int.

[4] J. Adv. Comput. Sci. Appl., vol. 10, no. 6, 2019.

[5] A. Kumar and G. P. Hancke, "Energy efficient environment monitoring system based on IoT using ESP32," IEEE Internet Things J., vol. 6, no. 5, 2019.

[6] R. S. Bansode and P. P. Patil, "IoT based water flow monitoring system using ESP32," Int. J. Eng. Res. Technol. (IJERT), vol. 9, no. 6, 2020.

[7] M. S. Hossain and G. Muhammad, "Cloud-assisted industrial Internet of Things (IIoT) enabled framework for health monitoring," IEEE Internet Things J., vol. 5, no. 4, pp. 2946–2955, 2018.

[8] J. Kang, Y. Hwang, and H. Park, "IoT-based indoor and outdoor air quality monitoring system," in Proc. Int. Conf. Inf. Commun. Technol. Convergence (ICTC), 2018.

[9] N. K. Suryadevara and S. C. Mukhopadhyay, "Wireless sensor network based home monitoring system for wellness determination," IEEE Sens. J., vol. 12, no. 6, pp. 1965–1972, 2012.

[10] L. Liu, Q. Zhang, and Y. Peng, "Smart air quality monitoring system based on IoT and big data analytics," IEEE Access, vol. 7, pp. 123–135, 2019.

[11] T. Perumal, M. N. Sulaiman, and C. Y. Leong, "Internet of Things (IoT) enabled water monitoring system," in Proc. IEEE 4th Global Conf. Consumer Electronics (GCCE), 2015.