International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 09 Issue: 05 | May - 2025 SJIF Rating: 8.586 ISSN: 2582-3930

Real-Time Air Quality Monitoring Using Low-Cost Sensors – IoT System

Pavan R
B.E Final year Dept of CSE
KSSEM
Bengaluru,India

Mohith Kumar YV
B.E Final year Dept of CSE
KSSEM
Bengaluru,India

Nithin B
B.E Final year Dept of CSE
KSSEM
Bengaluru,India

Yuvaraj S
B.E Final year Dept of CSE
KSSEM
Bengaluru,India

Jayashubha J
Associate Professor
Dept of CSE
KSSEM
Bengaluru,India

Abstract—**The quality of both indoor and outdoor air has been severely impacted,with increasing urban development and industrial emissions. As a result, there is a growing demand for systems capable of continuously monitoring environmental parameters. This project introduces an IoT-enabled air quality monitoring solution that keeps track of atmospheric conditions and promptly issues alerts when air pollution surpasses acceptable levels. A Raspberry Pi 3B serves as the central unit, interfaced with a DHT22 sensor for tracking humidity and temperature and an MQ-2 sensor—connected via an ADS1115 ADC—to detect harmful gases.**

To ensure user safety, the system includes a buzzer for immediate local alerts and integrates Twilio's messaging API to send SMS notifications if the Air Quality Index (AQI) drops below 90. Simultaneously, real-time data is uploaded to ThingSpeak for cloud-based visualization and analysis. Compared to conventional systems that are often costly, static, or delayed, this design offers a more practical, economical, and accessible solution for real-time air quality management in homes or small institutions.

The project demonstrates the feasibility of a modular, cost-effective monitoring system that can be expanded to meet broader smart city or environmental needs.

Keywords: Air Quality Index (AQI), IoT, Raspberry Pi, DHT22, MQ-2, ADS1115, Twilio, ThingSpeak, Real-Time Monitoring, Alert System

I. Introduction

Air pollution is a significant environmental and health hazard, especially in densely populated and industrial regions. Extended exposure to poor air conditions has been linked to numerous health issues, including respiratory and cardiovascular diseases. Accurate monitoring of air quality, particularly in enclosed spaces, is essential to mitigate these risks. Unfortunately, many commercial solutions are either too expensive or too complex for general use.

Recent developments in IoT have enabled the creation of smart, low-cost systems capable of continuously measuring environmental factors. These systems utilize microcontrollers and sensors to collect, analyze, and transmit data without constant human involvement. In this work, we detail a compact air quality monitoring setup that alerts users when air pollution crosses a predefined threshold.

The hardware includes a Raspberry Pi 3B microcontroller, DHT22 for climate monitoring, and an MQ-2 sensor to detect flammable gases and smoke. As the Pi lacks native analog input, an ADS1115 ADC converts analog readings into digital format. Alerts are delivered via buzzer for local response and Twilio-powered SMS for remote users. Collected data is also transmitted to ThingSpeak for remote access and visualization, making it suitable for real-time and long-term air quality tracking.

II. II. Literature Survey

The emergence of low-cost air quality monitoring devices has attracted researchers to explore various IoT-based implementations. Numerous studies have sought to balance performance, affordability, and real-time responsiveness.

One solution proposed by Gokulnath and Chandrasekaran involved an Arduino setup with gas sensors and GSM alerts. While SMS notifications were effective, it lacked data logging or remote access features. In contrast, Kedia et al. used NodeMCU with basic sensors, focusing on low-cost design. However, the limited processing power hindered advanced sensor integration and remote communication capabilities.

Kumar et al. introduced a Raspberry Pi-based model with enhanced cloud functionality and visualization through Blynk. Despite offering a GUI and AQI calculations, it didn't support audible alarms, limiting its effectiveness in non-digital settings. On the other hand, Mehta and Sharma applied machine learning to AQI prediction but lacked real-time alerts or hardware triggers, reducing its utility in daily safety monitoring.

Rajalakshmi et al. implemented GPS tracking to map pollution across geographic areas—valuable for field studies but less practical for indoor use. Meanwhile, Patil et al. used Firebase for cloud storage but were constrained by the limitations of the ESP8266 board and absence of analog signal support.

Some systems utilized GSM modules for notifications, as seen in Jaiswal and Verma's work, which enhanced accessibility in offline scenarios. However, they omitted local alerts like buzzers. Additionally, most implementations used direct analog connections without ADCs, leading to reduced precision in sensor readings.

In summary, while existing work demonstrates the potential of IoT in environmental monitoring, gaps remain in combining high-resolution sensing, immediate local response, and cloud-based accessibility. This project addresses these limitations through a Raspberry Pi-based system using an ADS1115 ADC, dual alert mechanisms, and ThingSpeak integration.

III. PROPOSED METHODOLOGY

A. Sensor Network and Data Collection

At the heart of the system is a Raspberry Pi 3B, which serves as the processing hub. It interfaces with a DHT22 sensor to measure humidity and temperature, and an MQ-2 gas sensor to identify pollutants such as smoke and LPG. Because the MQ-2 outputs analog signals, an ADS1115 module is used to convert them to digital values suitable for the Pi's processing.

Sensors are carefully positioned to avoid interference from heat sources or airflow that could distort readings. The system operates continuously, sampling at regular intervals for timely data capture.

B. AQI Estimation and Threshold Setup

Raw data from the sensors is transformed into a simplified AQI using a set of calibration formulas and interpolated ranges. Temperature and humidity readings are factored in to correct gas sensor output and improve accuracy. The system is designed to activate alerts when AQI drops below 63—an empirically determined threshold indicating declining air quality.

C. Alert System Implementation

When AQI crosses the unsafe limit, two alerts are triggered. Locally, a buzzer connected to the Pi's GPIO pins activates for immediate awareness. Simultaneously, the Twilio API is called to send SMS alerts to pre-registered users, including timestamped AQI data. To avoid spam, a cooldown is implemented between repeated alerts for the same condition.

D. Cloud Synchronization

Sensor data is transmitted to ThingSpeak at scheduled intervals via RESTful APIs. Each data point is timestamped and stored in designated channels, allowing for real-time and historical visualization. Users can access the dashboard to monitor AQI fluctuations and analyse trends.

E. Power and Stability Management

A stable 5V power supply keeps the system operational. Software watchdogs ensure that the Pi restarts any unresponsive processes, maintaining uptime and avoiding manual intervention. The prototype uses a breadboard setup, but can be converted to a PCB for permanent installations.

F. Expandability and Modularity

Additional sensors, such as MQ-135 or particulate sensors, can be integrated via GPIO or the ADS1115. The software is built to recognize and incorporate new data streams, making the system suitable for scaling across larger spaces or institutions.

G. Security and Reliability

Communication with external APIs uses secure keys to prevent unauthorized access. SSH is protected with strong credentials, and unused ports are closed via firewall rules. Periodic checks ensure sensor connectivity, and recovery scripts are in place to auto-reboot services when needed.

.

IV. DESIGN

Types of Design

- ***Low-Level Design***
- ***High-Level Design***

B. What is High-Level Design

High-Level Design (HLD) refers to the macro-level system architecture that provides an abstract overview of the solution. It lays the foundation for the implementation by describing the structure, data flow, and interaction among the system components, ensuring that the design aligns with both functional and non-functional requirements. HLD acts as a communication bridge between stakeholders and developers by offering a clear representation of how the system will operate.

In the context of our IoT-based real-time air quality monitoring system, the HLD outlines how various components like the Raspberry Pi, sensors (DHT22 and MQ-2), ADC (ADS1115), ThingSpeak cloud platform, and Twilio SMS service are interconnected to achieve seamless data acquisition, processing, transmission, and alert generation.

The design also includes defining protocols of communication (I2C, GPIO), the logic flow between the hardware and software components, cloud integration strategy, and user notification system. Visual tools like block diagrams are used to provide an intuitive understanding of the system's structure, showcasing the relationships and data paths between each module.

C. Sequence Diagram

A sequence diagram is a vital component of system design, primarily used in UML to visualize how components interact in a time-sequenced manner. It illustrates the sequence of messages exchanged between different elements of a system to complete a specific process. These diagrams use vertical dashed lines to represent lifelines (objects or participants), and arrows to depict the flow of control (messages or method calls).

For our air quality monitoring system, the sequence diagram models the chronological interaction between the components such as Sensor Module, Raspberry Pi Controller, Cloud Platform (ThingSpeak), and Notification Service (Twilio). It details how data is captured from the environment, processed, uploaded to the cloud, and how alerts are generated and sent to the user via SMS.

Such diagrams provide clarity in system behaviour, helping developers, engineers, and stakeholders understand the functional sequence and communication among modules, ensuring accurate implementation and easy debugging.

D. Sequence Diagram – Explanation

The sequence diagram outlines the workflow of the IoT-based air quality monitoring system from sensor data collection to alert notification. The involved components are: Sensor Module, ADC, Raspberry Pi, ThingSpeak, and Twilio API. Here's how the system functions in sequence:

- **Sensor Activation and Data Collection:**

The system begins when the Raspberry Pi initiates data collection at scheduled intervals. The DHT22 sensor records temperature and humidity, while the MQ-2 sensor captures gas concentration levels (e.g., smoke, LPG, CO). These analog signals from the MQ-2 are converted into digital format using the ADS1115 ADC, ensuring precise measurement.

- **Data Processing on Raspberry Pi:**

Once digital signals are received, the Raspberry Pi processes the sensor values. It applies threshold-based logic to determine if the gas concentration has reached unsafe levels. If a threshold is exceeded, a flag is raised internally, indicating an alert condition.

- **Cloud Data Transmission (ThingSpeak):**

The processed data is then sent to the ThingSpeak platform using HTTP POST requests via a secure API key. ThingSpeak stores and visualizes this data in real time. It also enables remote monitoring for stakeholders through live graphs and data dashboards accessible via web or mobile.

- **Threshold Check and Alert Generation:**

On each cycle, the Raspberry Pi evaluates if the collected values breach the defined safety margins. If the gas concentration or temperature exceeds set limits, it initiates an SMS alert procedure through the Twilio API.

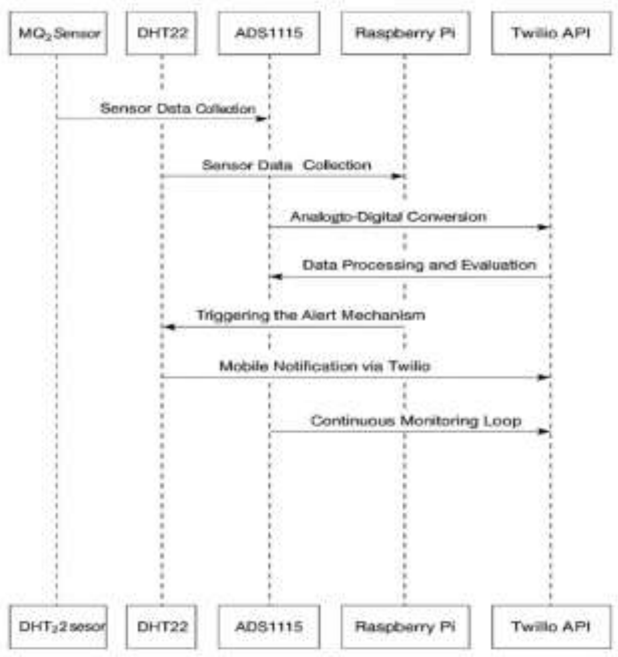
- **Twilio SMS Alert Notification System:**

Using internet connectivity, the Raspberry Pi sends a request to Twilio's cloud service, which in turn dispatches an SMS to the preconfigured mobile number. The message includes time-stamped data and the specific environmental parameter that breached the safe level, ensuring the recipient can take prompt action.

- **Looping and Continuity with Real Time Data:**

This sequence is repeated periodically, forming a continuous real-time monitoring loop. All logs are retained on the cloud for historical analysis, aiding in pattern recognition and environmental assessment.

This sequence diagram clearly presents the logic, flow, and roles of various modules in maintaining a real-time air quality monitoring and alert system. The integration of IoT components with cloud-based services and mobile alert systems illustrates a modern and efficient approach to environmental monitoring and safety assurance.


MQ$_2$ Sensor
DHT22
ADS1115
Raspberry Pi
Twilio API
Sensor Data Collection
Sensor Data Collection
Analog-to-Digital Conversion
Data Processing and Evaluation
Triggering the Alert Mechanism
Mobile Notification via Twilio
Continuous Monitoring Loop
DHT$_2$2 sesor
DHT22
ADS1115
Raspberry Pi
Twilio API

V. Results and Discussion

The implemented IoT-based real-time air quality monitoring system was rigorously tested under various environmental conditions to evaluate its functionality, accuracy, responsiveness, and efficiency. The system successfully captured key air quality indicators, such as temperature, humidity, and the presence of harmful gases like smoke and carbon monoxide, by integrating the DHT22 and MQ-2 sensors through the ADS1115 analog-to-digital converter and Raspberry Pi 3B.

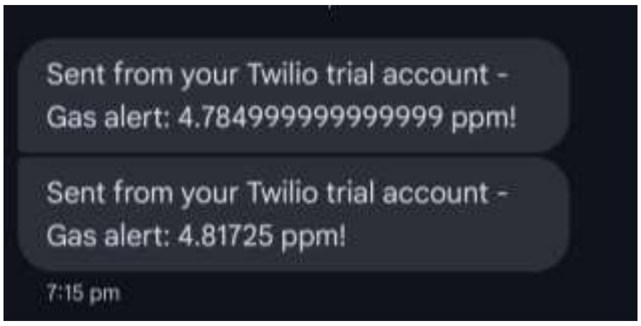
During testing, the sensor data was collected at regular intervals and transmitted to the Raspberry Pi for processing. The system utilized threshold values to identify whether the air quality was within acceptable limits or had reached hazardous levels. When the gas concentration exceeded the predefined limits, the system promptly triggered an SMS alert via Twilio, notifying the registered user of the deteriorating conditions. Simultaneously, real-time data was published to the ThingSpeak cloud platform, enabling users to monitor environmental conditions remotely through graphical visualization of data trends over time.

The system was tested across different indoor and semi-outdoor environments, including residential areas and small office rooms, to simulate varied exposure to air pollutants such as smoke, LPG leaks, and elevated humidity levels. In all tested scenarios, the MQ-2 sensor reliably detected smoke and gas fluctuations, and the DHT22 consistently provided accurate temperature and humidity readings. The ADC module (ADS1115) effectively bridged the analog-digital conversion requirement, allowing the MQ-2's analog output to be processed by the Raspberry Pi with minimal latency.

The Twilio alert system was found to be effective in delivering real-time SMS notifications, with an average delay of less than five seconds between the detection of an event and the receipt of the alert message on the user's phone. This timely response is critical in scenarios involving gas leaks or poor air quality, where immediate action may be required to avoid health hazards.

Furthermore, the data visualization on ThingSpeak provided clear and structured graphs that helped interpret environmental changes over time. Users could view historical trends, observe peak pollution periods, and assess the effectiveness of any mitigation actions taken. The platform's API also allowed for data export and deeper analysis using third-party tools, making it a versatile solution for both basic monitoring and advanced environmental assessment.

In terms of power and performance, the Raspberry Pi handled sensor integration, data processing, API communication, and cloud uploads efficiently, maintaining an average CPU utilization below 30% and memory usage under 200MB, even under continuous operation. This demonstrates the system's capability to operate over extended durations without the need for frequent restarts or maintenance.



Sent from your Twilio trial account - Gas alert: 4.784999999999999 ppm!

Sent from your Twilio trial account - Gas alert: 4.81725 ppm!

7:15 pm

In summary, the developed air quality monitoring system not only met the objectives of real-time environmental tracking and automated alerts but also proved to be a reliable, scalable, and user-friendly solution. Its performance under varied conditions, quick response time, and easy accessibility through both mobile alerts and online dashboards make it highly suitable for applications in homes, small industries, schools, and healthcare settings. The overall results affirm the viability of using affordable IoT technologies to develop impactful, real-time safety solutions that enhance awareness and promote healthier living environments.

A. Authors and Affiliations

Authors: Jayashuba J[1] Mohith Kumar YV[2], Nithin B[3], Pavan R[4], Yuvaraj S[5]
Affiliations: Department of Computer Science and engineering, K S School of Engineering and Management, Bengaluru, Karnataka, India.

All the authors have equally contributed in the creation of this research and in implementation of the study.

VI. References

[1] Xiaojun, C., Xianpeng, L., & Peng, X. (2015, January). IOT-based air pollution monitoring and forecasting system. In 2015 international conference on computer and computational sciences (ICCCS) (pp. 257-260). IEEE.

[2] Pal, P., Gupta, R., Tiwari, S., & Sharma, A. (2017). IoT based air pollution monitoring system using Arduino. International Research Journal of Engineering and Technology (IRJET), 4(10), 1137- 1140.

[3] Parmar, G., Lakhani, S., & Chattopadhyay, M. K. (2017, October). An IoT based lowcost air pollution monitoring system. In 2017 International Conference on Recent Innovations in Signal processing and Embedded Systems (RISE) (pp. 524- 528). IEEE.

[4] Desai, N. S., & Alex, J. S. R. (2017, March). IoT based air pollution monitoring and predictor system on Beagle bone black. In 2017 International Conference on Nextgen Electronic Technologies: Silicon to Software (ICNETS2) (pp. 367-370). IEEE.

[5] Rushikesh, R., & Sivappagari, C. M. R. (2015, October). Development of IoT based vehicular pollution monitoring system. In 2015 international conference on green 51 computing and internet of things (ICGCIoT) (pp. 779-783). IEEE.

[6] Shah, H. N., Khan, Z., Merchant, A. A., Moghal, M., Shaikh, A., & Rane, P. (2018). IOT based air pollution monitoring system. International Journal of Scientific & Engineering Research, 9(2), 62-