Visionair: An IOT-based Air Quality Monitoring System Using SMA and Hysterisis Algorithm

Alexander Roland A. Canarias
Department of College of Computer and Information Sciences
Mapua Malayan Colleges Mindanao
Davao City, Philippines
arCanarias@mcm.edu.ph

Daniel John Henrick D. Sanchez
Department of College of Computer and Information Sciences
Mapua Malayan Colleges Mindanao
Davao City, Philippines
djhSanchez@mcm.edu.ph

Jomer R. Mandap
Department of College of Computer and Information Sciences
Mapua Malayan Colleges Mindanao
Davao City, Philippines
jrMandap@mcm.edu.ph

I. INTRODUCTION

In today's rapidly urbanizing world, air pollution has emerged as a critical public health concern. According to the World Health Organization (WHO), 99% of the global population breathes air that exceeds pollution limits, leading to severe health risks including respiratory diseases and cardiovascular problems [15]. The increasing levels of pollutants, driven by industrial activities, vehicular emissions, and urban sprawl, necessitate innovative solutions for effective monitoring and management of air quality. Our project, an IoT Air Pollution Monitoring System, seeks to address this pressing issue by leveraging microcontroller-based technology to provide real-time data on air quality. This system aims to offer accurate, timely information to authorities and the public, facilitating informed decision-making and promoting healthier living environments.

The nature of the problem lies in the inadequate monitoring and management of air quality in many urban areas. Traditional air monitoring systems are often expensive and limited in coverage, failing to provide comprehensive data essential for addressing pollution hotspots. For instance, a study by the Environmental Protection Agency (EPA) highlights that many urban areas lack sufficient monitoring stations to accurately reflect localized air pollution levels [16]. By employing IoT-based technologies, our project offers a cost-effective, scalable solution that enhances monitoring capabilities across diverse locations. IoT devices can be strategically placed throughout urban environments to collect granular air quality data, providing a more detailed and accurate picture of pollution levels.

The concept of our project centers on addressing a critical environmental and public health issue: the pervasive and often invisible threat of air pollution. By utilizing microcontroller-based technology, our system ensures continuous, real-time monitoring of air quality parameters such as carbon dioxide (CO2), ammonia (NH3), and other hazardous gases. Additionally, the system measures temperature and humidity, which are essential for understanding the context of air quality data. The European Environment Agency (EEA) emphasizes the importance of real-time air quality monitoring in mitigating pollution-related health risks [17].

The primary objectives of this project are to develop an accessible and reliable air pollution monitoring system, raise public awareness about air quality issues, and support policymakers in implementing effective pollution control measures. The significance of this initiative is underscored by its potential to improve public health, reduce healthcare costs, and contribute to the overall well-being of communities. Research from the European Public Health Alliance (EPHA) indicates that improved air quality can significantly reduce healthcare expenses and enhance quality of life [18]. Embracing IoT and microcontroller-based solutions not only enhances the precision of pollution monitoring but also paves the way for smarter, more responsive environmental management. The World Economic Forum (WEF) has highlighted the transformative potential of IoT technologies in creating sustainable urban environments [19].

In summary, our IoT Air Pollution Monitoring System addresses a vital environmental challenge with a technologically advanced, scalable, and cost-effective solution. By providing real-time, accurate data on air quality, it empowers communities and authorities to take proactive measures in combating air pollution and safeguarding public health.

II. RELATED WORKS

Air pollution is a major challenge, affecting not only humans but also animals, plants, and buildings. It is a serious environmental issue causing significant health problems worldwide, with air quality indices reaching critical levels in some areas [5]. Predicting air contaminants is crucial for addressing this issue, and this chapter explores the use of artificial intelligence and machine learning for this purpose [7]. The World Health Organization reported in 2019 that poor air quality affects about 91% of the global population, leading to around 7 million deaths annually and environmental damage. Despite numerous strategies proposed to reduce air pollution, essential activities such as manufacturing and coal combustion continue to contribute to the problem. Advanced data-centric systems and artificial intelligence are needed to forecast air pollution by analyzing sources, predicting levels, assessing impacts, and providing data-driven solutions [6].

Air pollution has become major cause of global concern in recent times. Global warming, climate changes and many major hazardous diseases are all reasons of air pollution. Rapid industrialization, urbanization, deforestation have all accelerated the air pollution level. Air is getting polluted due to release of poisonous gases by industrialization, vehicular emissions, unplanned construction activities. Everyone is at an alarming risk whether they are children or elder [10]. In response, the 1955 Air Pollution Control Act laid the groundwork for subsequent reforms. The 1963 Clean Air Act established a federal air pollution control program, and the 1967 Air Quality Act expanded federal efforts, including extensive air monitoring and inspections. Public awareness surged in the 1960s and 1970s, leading to the 1970 Clean Air Act, which replaced the 1967 Act, created the EPA, and mandated the development of National Ambient Air Quality Standards (NAAQS) for six criteria pollutants to protect public health and welfare [8].

The world is getting polluted because of emission of dangerous gases into air such as CO_2, SO_2, NO_2, and CO. These toxic gases are dissolved in air and cannot be predicted. Hence a tool is required to check the air quality. The air pollution can be monitor by using internet based devices like IoT. Internet of thing (IoT) devices can collect the data and based on data can analysis for prediction i.e. quality of air is good or not. Thus, the air quality of a particular area can be monitored using IOT based devices and sensors using Arduino/Raspberry Pi [2]. The significance of the air quality statistics makes the highly accurate real-time monitoring systems vital. The partial data access, high cost and the non-scalability of conventional air monitoring system enforce the researchers to develop future air pollution monitoring system employing advance technologies such as internet of things (IoT), wireless sensor network (WSN) and the low-cost ambient sensors [11]. Wireless sensor networks (WSNs) have advanced the monitoring of various environmental phenomena, including air pollution, as discussed in this paper. A WSN-based microcontroller equipped with gas sensors was actively utilized for air quality monitoring. The design comprised several components: an Arduino microcontroller, MQ-2 gas sensors, and a current regulator circuit. Results showed significant differences in the levels of LPG and CO gases compared to normal clean air levels, based on multiple tests and circuit runs. However, air quality measurements inside KFU buildings indicated no hazardous conditions requiring further action. This study applies electrical and environmental engineering techniques to use WSNs for measuring air quality parameters like carbon monoxide (CO) and liquid petroleum gas (LPG) [4].

This study discusses the development of a gas sensor system designed to function as a sensing node within a dense, real-time environmental monitoring network. A novel auto-calibration method is introduced to ensure the sensor network operates without the need for maintenance. Network connectivity facilitates not only data collection but also the calibration and diagnosis of sensors by comparing measured pollutant concentrations with nearby sensors and governmental monitoring stations. A case study on local NO2 distribution revealed that under certain conditions, pollutant concentrations can become low and uniform in a specific area, allowing baseline adjustment using data from neighboring stations. Experimental results demonstrate that by incorporating temperature and humidity compensation, NO2 concentrations can be measured accurately, and the effectiveness of auto-calibration in maintaining long-term measurement accuracy is confirmed [3].

Air quality monitoring is crucial in our daily lives, significantly impacting human health. It aids in understanding the sources, effects, and levels of pollutants in the air, which helps develop effective pollution control measures. This work aims to develop and test, a high-resolution multi-scale air pollution modelling system by integrating a set of adequate tools. The Air Quality Index (AQI) is a tool used to gauge pollution levels. This study reviews advanced methods like SVM, RF, ANN, RNN, and FL for predicting air quality with machine learning, showing promising results that can be implemented in cost-effective hardware for both household and commercial use.

The proposed model uses eight parameters—NO2, CO, O3, PM2.5, PM10, SO2, temperature, pressure, dew point, rainfall, wind direction, and wind speed from the Beijing dataset—to predict AQI and has been tested with regional data [1]. Various algorithms were used like Linear Regression, Lasso Regression, Random Forest and Decision Tree to predict the future AQI with past AQI and found that Decision Tree algorithm gives best prediction with limited number of data entries and Random Forest gives best result with larger dataset [10]. This system is able to provide detailed air pollutant concentrations in areas and support air quality management strategies through a better identification of different atmospheric processes. It also allows furthering the design and assessment of air pollution control for a specific area [9].

The Arduino-based Air Quality Monitoring System uses various sensors to detect environmental parameters such as gas levels, temperature, and humidity. The system utilizes an MQ135 gas sensor to measure air quality, an Arduino UNO board for processing, and a DHT11 sensor to capture temperature and humidity data. The analog readings from the MQ135 sensor are input into the Arduino's analog pin, where they are mapped to predefined thresholds to categorize air quality as "Good," "Poor," "Very Bad," or "Toxic." These categories are based on the Air Quality Index (AQI) standards, which convert raw sensor data (e.g., PM2.5, PM10, CO, NO2, O3 levels) into standardized AQI values using predefined formulas and breakpoints set by environmental agencies. The system displays real-time data on an OLED screen using Adafruit libraries, giving users a snapshot of air quality conditions. This continuous monitoring enables timely assessment and response to environmental changes, ensuring better air quality management and awareness. According to research by IEEE Xplore [1], the effectiveness of such systems is enhanced by integrating reliable sensors and efficient data processing units like the Arduino, making them suitable for various applications, including home monitoring and environmental assessments.

III. MATERIALS AND METHODS

3.1 List of Materials

Name	Purpose
ESP32	Serves as the central processing unit, executing the code to control the air pollution monitoring system.
MQ-135 Gas Sensor	Detect various harmful gases like ammonia, nitrogen oxides, alcohol, benzene, smoke, and carbon dioxide in the air.
MQ4 Methane Sensor	Detects Methane (CH4), and can also sense other gases such as Liquefied Petroleum Gas (LPG) and natural gas.
MQ6 LPG Gas Sensor	Detects Liquefied Petroleum Gas (LPG), isobutane, propane, and can also sense other gases.
MQ7 Carbon Monoxide Sensor	Detects Carbon Monoxide (CO) and can also sense hydrogen gas.
MQ9 Gas Sensor	Detects Carbon Monoxide (CO) and Methane (CH4), and can also sense other gases like LPG and coal gas.
DHT11 Temperature and Humidity Sensor	Measures the ambient temperature and humidity levels, providing additional environmental data for air quality assessment.
LCD Display	Shows real-time data from the sensors, allowing users to visually monitor air quality metrics.
Breadboard	A platform that allows for the temporary setup and connection of electronic circuits without the need for soldering.
Jump Wires	Electrical wires used to connect different components on the breadboard and establish connections with the Arduino board.
Enclosure	A three dimensional enclosure for the system.

Table 1 – List of Materials

The table above offers a comprehensive list of vital materials required for successfully implementing the Pathfinder project. It encompasses a diverse range of essential items, each providing a pivotal role in ensuring the effective execution of the project. The table above outlines the necessary materials for the Pathfinder project, covering a diverse range of crucial items essential for the project's successful execution.

3.2 Block Diagram

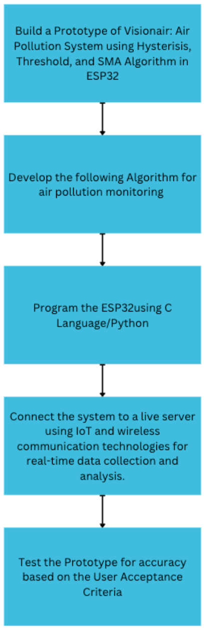

Figure 1: System Flow Chart

The figure above illustrates the implementation of Visionair, an air pollution monitoring system that leverages hysteresis, threshold, and SMA algorithms on an ESP32 platform. The process begins with building the prototype, which involves assembling the necessary hardware components such as sensors and the ESP32 board. Subsequently, specialized algorithms for air pollution monitoring are developed to assess air quality levels accurately. These algorithms are programmed into the ESP32 using C language or Python, enabling the system to process sensor data and compute the Air Quality Index. The system is then connected to a live server using IoT and wireless communication technologies, facilitating real-time data collection and analysis.

In the final phase, the prototype undergoes rigorous testing to ensure its accuracy, reliability, and overall performance, based on established User Acceptance Criteria. These tests involve comparing the system's readings with calibrated reference devices, assessing its responsiveness to environmental changes, and verifying stable data transmission to the server. The integration of real-time monitoring capabilities and AI-driven analysis makes Visionair a robust tool for air quality management. This comprehensive approach not only enhances the precision of air quality assessments but also supports the development of effective pollution control measures. By addressing these critical aspects, the Visionair system aims to contribute significantly to environmental monitoring and public health protection.

3.3.1 MQ135 Gas Sensor

The Gas Sensor is commonly used to detect a variety of gases, such as ammonia (NH3), nitrogen oxides (NOx), alcohol, benzene, smoke, and carbon dioxide (CO2). These sensors are beneficial for monitoring air quality due to their sensitivity to various gases commonly found in polluted environments. The primary function of the Gas Sensor is to detect harmful gases and provide an analog output signal that the Arduino can read. The hardware allows the system to measure the concentration of various gases in the air. By converting the analog signal into digital values, the Arduino can categorize the air quality into predefined levels such as "Good," "Poor," "Very Bad," or "Toxic." This categorization helps to provide a clear and understandable assessment of air quality conditions, which is crucial for ensuring public health and safety. The versatility and reliability of the Gas Sensor make it an ideal choice for various applications, including indoor air quality monitoring, industrial safety, and environmental studies. According to Majumder et al. [13], the Gas Sensor's ability to detect multiple gases makes it valuable in comprehensive air quality monitoring systems.

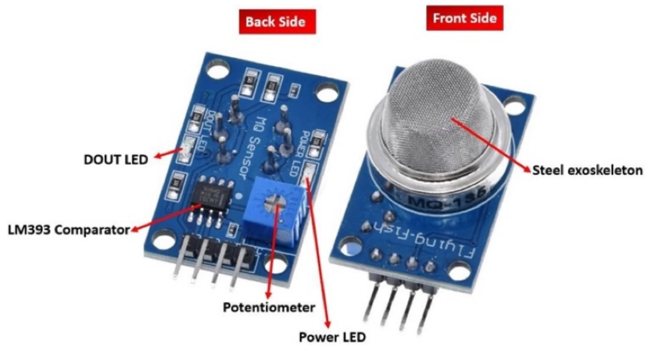

Figure 2: MQ135 Gas Sensor Diagram

3.3.2 DHT 11 Temperature and Humidity Sensor

The DHT11 Temperature and Humidity Sensor is an economical digital sensor that measures ambient temperature and humidity. These sensors offer calibrated digital output, simplifying the connection to the Arduino. The DHT11 provides ambient temperature and humidity measurements, adding valuable context to air quality data. These factors are crucial in influencing pollutant behavior and overall air quality. By incorporating temperature and humidity data, the system can offer a more comprehensive evaluation of environmental conditions, enhancing the accuracy of AQI calculations. This integration is essential for assessing air quality, as temperature and humidity can significantly influence the dispersion and concentration of pollutants in the air. As noted by Zhang et al. [14], the DHT11 sensor has been successfully utilized in various environmental monitoring applications, underscoring its reliability and compatibility with microcontroller-based systems.

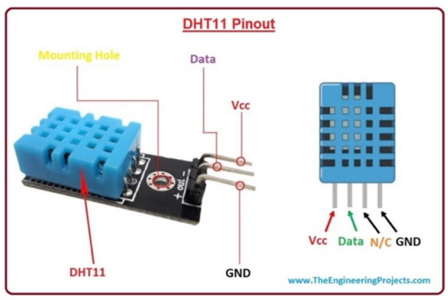

Figure 2: DHT11 Temperature and Humidity Sensor Diagram

3.3.3 MQ4 Methane Sensor

The MQ4 methane gas sensor operates based on the electrical conductivity of a metal oxide semiconductor (MOS) material. When exposed to methane gas (CH_4), the resistivity of the detecting element (usually a ceramic coated with tin dioxide) changes. This change in resistivity is measured to determine the concentration of methane gas. The sensor has three pins: H Pins (supply voltage and ground), A Pins (interchangeable, connected to the supply voltage), and B Pins (interchangeable, one acts as the output pin, and the other is grounded). It provides both analog output (AO) with a voltage range of 0V to 5V and digital output (DO) based on a threshold set using a potentiometer. Applications include detecting natural gas leaks (methane) in homes and industries, as well as CNG (compressed natural gas) detection. Sensitivity can be adjusted via the potentiometer. [20] The MQ-4 methane gas sensor can significantly enhance an Arduino-based air quality monitoring system by providing reliable methane detection capabilities. It is designed to detect methane (CH_4) gas, making it ideal for monitoring environments where methane presence could indicate potential leaks or hazardous conditions, such as in homes, industrial sites, and areas utilizing natural gas or compressed natural gas (CNG). The sensor offers both analog and digital outputs, which allow it to integrate seamlessly with Arduino. The analog output provides a continuous voltage signal proportional to the methane concentration, while the digital output can be used to trigger alarms or other actions when methane levels exceed a set threshold. This versatility makes the MQ-4 sensor an essential component for ensuring safety and monitoring air quality effectively.

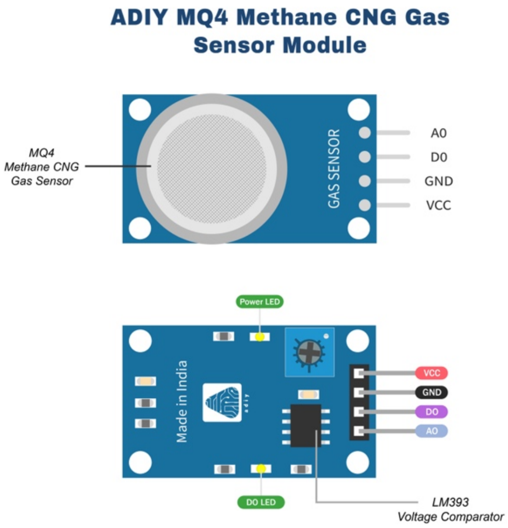

Figure 3: MQ4 Methane Sensor Module Diagram

3.3.4 MQ6 LPG Gas Sensor

The MQ6 gas sensor is a small and cost-effective device used to detect specific gases, including methane, propane, and particularly LPG (liquefied petroleum gas). It operates by measuring changes in resistance caused by the presence of a gas. When exposed to gases like methane, propane, or LPG, the sensor's detecting element (usually a ceramic coated with tin dioxide) undergoes a change in resistivity. This change is then used to determine the gas concentration. The MQ6 sensor typically has three pins: H Pins (supply voltage and ground), A Pins (interchangeable, connected to the supply voltage), and B Pins (interchangeable, one acts as the output pin, and the other is pulled to the ground terminal). Applications include detecting gas leaks at home or in industries, suitable for LPG gas detection, and it can also detect other combustible gases. [21] Adding the MQ-6 sensor to an Arduino-based air quality monitoring system can significantly enhance its capability to detect a range of combustible gases, ensuring safety in environments where such gases are present.

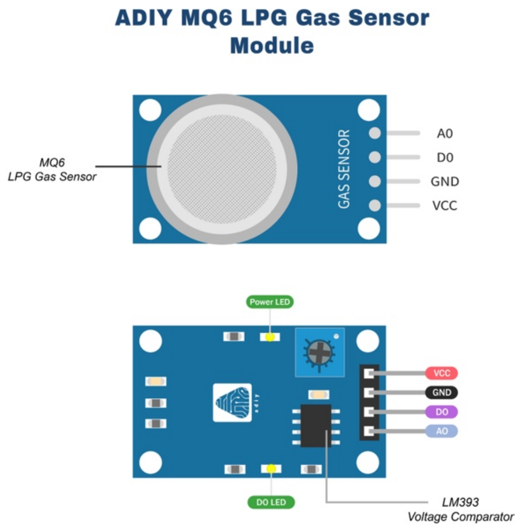

Figure 4: MQ6 LPG Gas Sensor Module Diagram

3.3.5 MQ7 Carbon Monoxide Gas Sensor

The MQ7 gas sensor is a semiconductor-based device designed to detect carbon monoxide (CO) in the air. It operates on the principle of chemoresistance—when exposed to CO gas, the electrical resistance of the sensor changes. The MQ7 sensor specifically targets carbon monoxide, utilizing a detecting element (usually tin oxide, SnO_2) that responds to CO. As the concentration of CO increases, the sensor's resistance changes, allowing for reliable CO detection. Its applications include detecting carbon monoxide leaks in homes, vehicles, and industrial settings, with safety implications to prevent CO poisoning. Note that other sensors in the MQ series are optimized for different gases. [22] Incorporating the MQ-7 sensor into an Arduino-based air quality monitoring system can provide vital carbon monoxide detection, enhancing the system's ability to ensure safety in environments where CO exposure is a risk. The sensor's ability to provide real-time monitoring and alert systems can help in early detection of hazardous CO levels, allowing for prompt action to prevent health hazards. Additionally, the sensor's affordability and ease of use make it an ideal choice for DIY projects and professional applications alike, ensuring that air quality monitoring is both effective and accessible.

To integrate the MQ-7 sensor with an Arduino, one can connect the sensor's analog output to one of the analog input pins on the Arduino. This setup allows for reading the varying voltage levels corresponding to the CO concentration. The digital output can be connected to a digital pin on the Arduino, enabling the system to trigger alarms or notifications when CO levels exceed a predefined threshold. By utilizing the Arduino's programming capabilities, it is possible to create a comprehensive air quality monitoring system that logs CO levels, displays real-time data on an LCD screen, and sends alerts via SMS or email. This integration makes the MQ-7 sensor an essential component for developing a robust and responsive air quality monitoring system, providing peace of mind and enhancing safety measures in various settings.

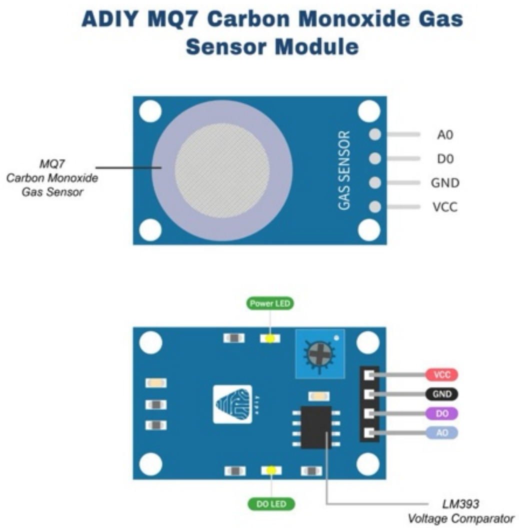

Figure 5: MQ7 Carbon Monoxide Gas Sensor Module Diagram

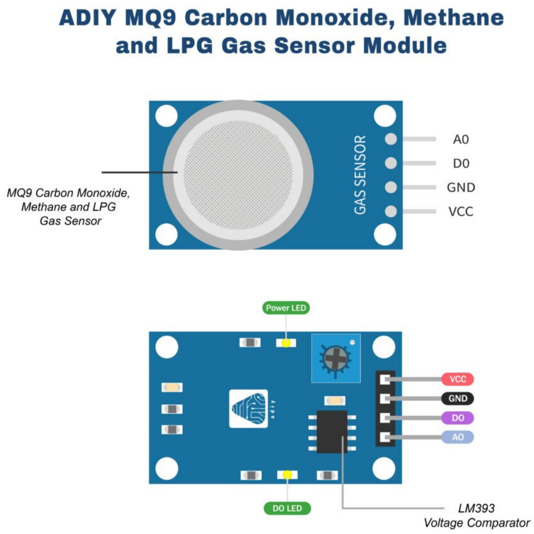

Figure 6: MQ9 Carbon Monoxide, Methane, and LPG Gas Sensor Module Diagram

3.3.6 MQ9 Carbon Monoxide, Methane, and LPG Gas Sensor

The MQ-9 gas sensor belongs to the MQ series, which includes specialized sensors for specific gases. In the case of the MQ-9, it targets gases like carbon monoxide (CO), methane, and propane—all of which can be life-threatening if leaked. The sensor's core components include a ceramic sensing substrate (usually alumina, Al_2O_3), a tin oxide (SnO_2) sensing element, a heater circuit (nichrome wire), and metallic electrodes. Operating on the chemiresistor principle, the sensor's resistance changes as it absorbs target gases. Applications include detecting carbon monoxide leaks in homes, vehicles, and industrial settings, with safety implications to prevent CO poisoning. Note that other sensors in the MQ series are optimized for different gases. [23] Incorporating the MQ-9 sensor into an Arduino-based air quality monitoring system enhances its capability to detect multiple hazardous gases. The sensor's versatility in detecting CO, methane, and propane makes it ideal for environments where the presence of these gases could pose significant safety risks. By connecting the MQ-9 sensor's analog output to an Arduino's analog input pin, the system can monitor real-time gas concentrations and provide immediate feedback. The digital output can be used to activate alarms or other safety mechanisms when gas levels exceed safe thresholds.

Using the Arduino's programming features, one can develop a comprehensive air quality monitoring system that logs gas levels, displays data on an LCD screen, and sends alerts via SMS or email. This integration ensures a robust and responsive system capable of early detection and prevention of hazardous conditions, thus safeguarding lives and property in various settings. The affordability and reliability of the MQ-9 sensor make it a practical choice for both DIY projects and professional applications.

3.4 System Implementation

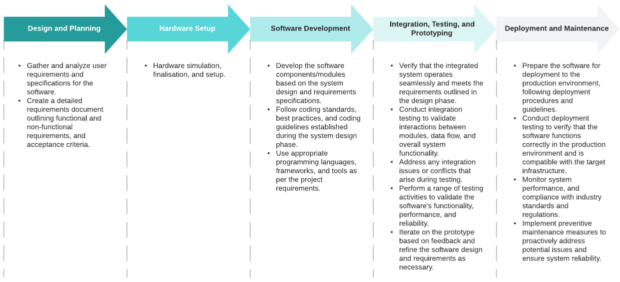

Figure 3.4.1 Sequence of the System Implementation

The provided diagram outlines a comprehensive process for software development, divided into several key phases. Initially, the "Design and Planning" phase involves gathering and analyzing user requirements and specifications to create a detailed requirements document that includes both functional and non-functional requirements, as well as acceptance criteria. This is followed by the "Hardware Setup" phase, where hardware simulation, finalization, and setup are carried out to ensure the hardware components are ready for integration.

In the "Software Development" phase, the software components and modules are developed based on the system design and requirements specifications. This phase emphasizes adhering to coding standards, best practices, and guidelines established during the system design phase, using appropriate programming languages, frameworks, and tools as per the project requirements.

The next phase, "Integration, Testing, and Prototyping," involves verifying that the integrated system operates seamlessly and meets the requirements outlined in the design phase. Integration testing is conducted to validate interactions between modules, data flow, and overall system functionality, addressing any integration issues or conflicts that arise. Various testing activities are performed to validate the software's functionality, performance, and reliability, with iterations on the prototype based on feedback to refine the software design and requirements as necessary.

Finally, in the "Deployment and Maintenance" phase, the software is prepared for deployment to the production environment following deployment procedures and guidelines. Deployment testing is conducted to verify that the software functions correctly in the production environment and is compatible with the target infrastructure. Post-deployment, system performance is monitored to ensure compliance with industry standards and regulations, and preventive maintenance measures are implemented to proactively address potential issues and ensure system reliability.

3.4 Operations

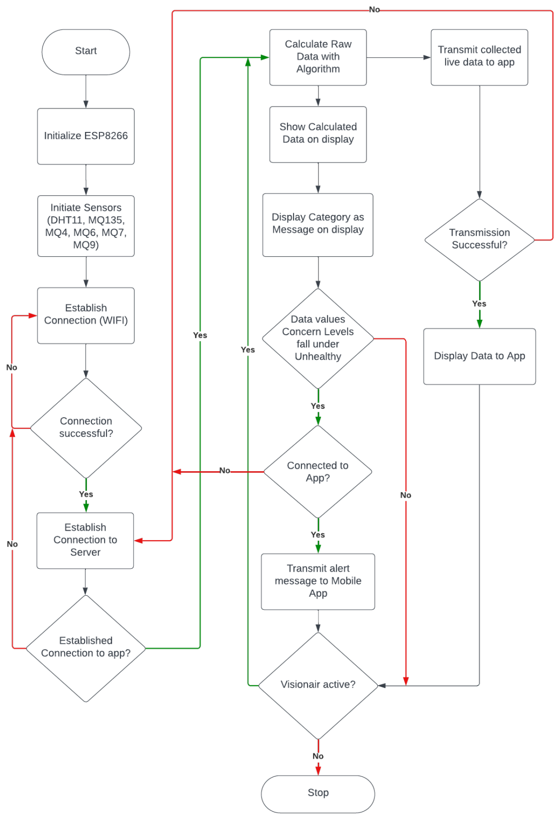

Figure 5: System Operation Chart

The flowchart illustrates the process of implementing an air quality monitoring system using an ESP32 microcontroller, sensors (DHT11 for temperature and humidity, MQ135 for gas detection), and the Blynk platform for remote monitoring. The process begins with initializing the ESP8266, followed by initiating the DHT11 and MQ135 sensors. Once the sensors are active, the system attempts to establish a WiFi connection. If the connection is unsuccessful, the system will continue to retry until a successful connection is established. Upon successful WiFi connection, the system then attempts to connect to the Blynk server. If this connection fails, the system retries until it successfully establishes a connection to the Blynk server.

Once connected to Blynk, the system calculates the raw data obtained from the sensors using an algorithm and displays the calculated data on an OLED display. If the data values indicate unhealthy air quality levels, the system displays a warning message on the OLED display. Additionally, the system checks if it is connected to the Blynk mobile app. If connected, it transmits an alert message to the Blynk app. The system then transmits the collected live data to Blynk. If the data transmission is successful, it displays the data on the app; otherwise, it continues to retry the transmission. Finally, the system checks if VisionAir (a hypothetical module or feature) is active. If VisionAir is not active, the system stops; otherwise, it continues to monitor and transmit data, ensuring real-time updates on air quality.

IV. IMPLEMENTATION AND ANALYSIS

4.1 Hardware Setup

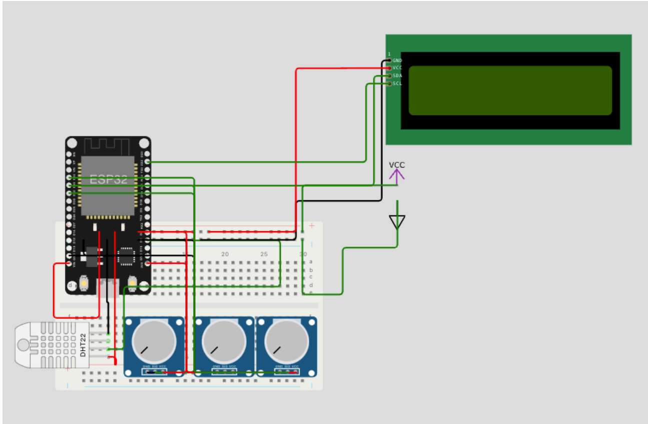

Figure 6: Simulation Model

The circuit diagram involves an ESP32 microcontroller connected to various components to monitor environmental conditions. The MQ series gas sensors is used to detect air quality and various gases, with its analog output connected to one of the ESP32's analog input pins. The DHT11 sensor measures temperature and humidity, with its data pin connected to a digital input pin of the ESP32. An OLED display, connected via I2C protocol (with SCL and SDA lines connected to GPIO 22 and GPIO 21 respectively), shows the readings from both sensors. Additionally, an LED is connected to a digital output pin of the ESP32 through a current-limiting resistor, serving as an indicator. All components are mounted on a breadboard, with power and ground connections distributed from the ESP32. This setup allows real-time monitoring and display of air quality, temperature, and humidity data.

4.2 Software Development

The software underpinning the VisionAir air pollution monitoring system is pivotal in managing the seamless integration of real-time data collection, sophisticated processing, and user interaction. With a commitment to robustness and user accessibility, the development strategy emphasizes the utilization of the Blynk Cloud UI for the user interface and applies a combination of the Hysteresis Algorithm, Simple Moving Average (SMA), and thresholding techniques for data processing and analysis.

Data Acquisition Module Enhancement

Utilizing secure and efficient communication protocols, this module ensures the accuracy and real-time capture of data from IoT sensors. Advanced encryption techniques safeguard data during transmission, emphasizing speed and integrity.

Processing and Analysis Unit with Specific Algorithms

This unit is the heart of VisionAir's analytical capabilities, employing

1. ***Hysteresis Algorithm***
    - It effectively filters data to differentiate between minor fluctuations and significant environmental changes, reducing false alarms and enhancing the relevancy of alerts.

2. ***Simple Moving Average (SMA)***
   - By smoothing out data series, the SMA technique offers clearer insights into air quality trends, making the system's forecasts more reliable and understandable.

3. ***Threshold Techniques***
   - These are used to trigger alerts when pollutant levels cross predefined safe limits, ensuring timely notifications and actions.

Integration of these algorithms provides a robust framework for analyzing air quality data, enabling predictive insights and actionable intelligence.

User Interface via Blynk Cloud UI

The choice of Blynk Cloud UI for the user interface design stands out for its simplicity, efficiency, and flexibility. This platform allows for the creation of a highly intuitive, engaging user interface that caters to both technical experts and the public. Features include:

1. Easy navigation through real-time and historical data.
2. Customizable dashboards tailored to user preferences.
3. Accessibility enhancements for wider user inclusivity.

The Blynk Cloud UI not only simplifies the user interaction with complex data but also promotes an inclusive environment where more users can actively participate in monitoring air quality.

Notification and Alerts System

The refined system offers personalized settings for alerts based on specific user-defined thresholds over the Blynk Cloud UI, ensuring that users receive relevant notifications that cater to their health and environmental concerns.

Scalability and Security Through Cloud-Based Integration

Leveraging cloud technologies ensures that VisionAir's infrastructure is scalable, secure, and capable of integrating with third-party services. This broadens the system's capability to include additional environmental data, offering a comprehensive view of factors affecting air quality.

Through these focused enhancements, the VisionAir project aims to provide an accessible, reliable, and comprehensive air quality monitoring solution, empowering communities to make informed decisions for a healthier environment.

4.3 Data Visualisation

The Data Visualization component of the VisionAir air pollution monitoring system represents a crucial interface for users, enabling them to intuitively understand and interpret the air quality data collected by the IoT sensors. Central to this system is an interactive dashboard that presents data in an engaging and informative manner, ensuring that users can easily grasp the current air quality status and trends over time.

Interactive Dashboard Overview

The core of VisionAir's data visualization is the interactive dashboard, designed with the user's experience in mind. This dashboard integrates graphs for each sensor reading, presenting a comprehensive view of air quality metrics such as CO2 levels, particulate matter (PM2.5 and PM10), nitrogen dioxide (NO2), and others. Users can access real-time data and historical data analysis, allowing for a comparative view of air quality changes.

Features of the Dashboard

1. ***Real-Time Graphs***
   - Display live data feeds from each sensor, dynamically updating to ensure the most current information is always available. This instant access to data enables users to monitor air quality changes as they happen.

2. ***Historical Trend Analysis***
   - Users can select specific time frames to analyze historical air quality trends. This feature is invaluable for identifying patterns or determining whether the air quality in a particular area is improving or deteriorating over time.

3. ***Customizable Views***
   - The dashboard allows users to customize their view by selecting which sensor readings they want to display or highlight. This customization ensures that users can focus on the pollutants of most concern to them.

4. ***Alert Threshold Visualization***
   - For ease of understanding, the dashboard visually represents threshold levels for each pollutant. When readings exceed safe levels, the corresponding graphs change color or trigger visual alerts, making it immediately apparent when air quality poses a health risk.

5. ***Comparative Analysis Tools***
   - Users can compare air quality across different locations or times, offering insights into how environmental factors or policy changes impact air quality.

User-Centric Design

The dashboard's design adheres to best practices in user interface (UI) design, ensuring it is not only functional but also aesthetically pleasing. Accessibility is a key consideration, with design choices that accommodate a broad spectrum of users, including those with visual impairments. Interactive elements such as tooltips and zoom-in features allow users to delve deeper into the data for a better understanding of air quality metrics.

Integration with Educational Resources

Beyond presenting data, the dashboard serves as an educational tool. Links to resources about air pollution and its effects, tips for reducing exposure, and explanations of the data and metrics are integrated into the dashboard. This approach empowers users with knowledge, making the dashboard not just a tool for monitoring but also for learning and action.

The Data Visualization module of the VisionAir system, epitomized by its interactive dashboard, stands as a testament to the power of clear, engaging, and actionable information presentation. By making complex data accessible and understandable, VisionAir plays a pivotal role in fostering community engagement, raising awareness, and driving collective action towards cleaner air.

4.4 Algorithm Integration

Threshold-Base Algorithm

The Threshold algorithm serves as a foundational element in our air quality monitoring project. This algorithm sets specific limits or thresholds for key parameters such as CO2 levels, temperature, and humidity, based on established standards or user-defined criteria. When sensor readings cross these predetermined thresholds, the system initiates predefined actions, such as activating ventilation systems, adjusting HVAC settings, or sending notifications to alert users. By implementing the Threshold algorithm, we ensure that our monitoring system can promptly identify and respond to deviations from acceptable air quality conditions, contributing to a healthier and more comfortable indoor environment.

```
initialise variable threshold_value to predefined limit
initialise array alert_messages

for each data_point in sensor_data:
    if data_point > threshold_value:
        append "High pollution alert!" to alert_messages
    else:
        append "Pollution levels normal" to alert_messages

return alert_messages
```

Figure 4.4.1: Threshold-Based Algorithm Pseudocode

Simple Moving Average Algorithm

In conjunction with the Threshold algorithm, the Simple Moving Average (SMA) algorithm plays a vital role in our project's data processing. SMA works by computing the average of recent sensor readings over a specified time window. This averaging process helps to smooth out short-term fluctuations and noise in the data, providing a clearer and more stable representation of air quality trends. By utilizing SMA, we can effectively observe patterns and changes in CO2 levels, temperature, and humidity, enabling us to make informed decisions and take appropriate actions to maintain optimal indoor air quality.

$$SMA = \frac{\sum_{i=1}^{n} X_i}{n}$$

Figure 4.4.2: SMA Algorithm Mathematical Equation

```
initialise array data_window
initialise variable window_size to desired window length
initialise variable sum to 0
initialise array sma_values

for each new data_point in sensor_data:
    if data_window has less than window_size elements:
        append data_point to data_window
        add data_point to sum
        sma = sum / length of data_window
    else:
        remove the oldest data point from data_window
        subtract the oldest data point from sum
        append data_point to data_window
        add data_point to sum
        sma = sum / window_size

    append sma to sma_values

return sma_values
```

Figure 4.4.3: SMA Algorithm Pseudocode

Hysteresis Algorithm

Additionally, the Hysteresis algorithm complements our monitoring system by introducing buffer zones around the predefined thresholds established by the Threshold algorithm. This buffer zone prevents rapid toggling of actions in response to minor fluctuations in sensor readings (e.g. temperature and gas levels). By requiring a significant deviation from the thresholds before triggering actions, the Hysteresis algorithm ensures more stable and reliable responses, reducing unnecessary alerts and system disruptions. The combination of the Hysteresis algorithm with the Threshold and SMA algorithms enhances the overall robustness and effectiveness of our air quality monitoring system, ensuring continuous monitoring and management of indoor air quality parameters.

```
initialize variable upper_threshold to high limit
initialize variable lower_threshold to low limit
initialize variable alert_status to False
initialize array hysteresis_alerts

for each data_point in sensor_data:
    if data_point > upper_threshold:
        if alert_status is False:
            append "High pollution alert!" to hysteresis_alerts
            set alert_status to True
    elif data_point < lower_threshold:
        if alert_status is True:
            append "Pollution levels normal" to hysteresis_alerts
            set alert_status to False

return hysteresis_alerts
```

Figure 4.4.4: Hysteresis Algorithm Pseudocode

4.5 Impact Assessment

The VisionAir air pollution monitoring system significantly impacts beyond public health, particularly in areas of environmental awareness and community engagement. Through its deployment, it fosters a deeper understanding of pollution, its sources, and its effects among students and the public. Educational programs utilizing VisionAir's real-time air quality data can encourage a culture of environmental stewardship and responsibility, leading to behavioral changes that positively affect the environment. Additionally, by providing accessible air quality data to the public, VisionAir empowers residents to actively participate in advocating for and implementing cleaner air initiatives, ultimately strengthening grassroots movements dedicated to enhancing environmental practices.

Analysis Method

For assessing the health impacts of air pollution detected by our system, we adopt a simpler, yet effective method. Instead of complex epidemiological models, we use a direct comparison approach. We gather air quality data from our sensors and directly correlate this with health incident reports from local healthcare facilities over the same time frame. By observing increases in health incidents aligned with poor air quality days, we establish a straightforward link between air pollution levels and health impacts on the community. This method allows us to clearly communicate with local authorities and community members about the immediate health risks associated with air pollution spikes. Although less detailed than advanced models, this approach effectively highlights critical issues, providing a clear basis for action and awareness campaigns.

Policy Recommendations and Community Awareness

Based on our findings, we make practical recommendations to local policymakers aimed at reducing harmful emissions and promoting public health initiatives, such as increasing green spaces or improving public transportation. We also aim to raise awareness within the community about the health risks linked to air pollution through easy-to-understand reports and alerts via the Blynk app. Informing people about the air quality in real-time encourages proactive behavior changes to minimize exposure to pollution, such as choosing indoor activities on high-pollution days.

By simplifying our health impact assessment method, we maintain a focus on clear, actionable data that can spur both community and governmental actions towards healthier environments.

Environmental Awareness and Education

The VisionAir system plays a crucial role in environmental education by providing real-time data on air quality. Schools and community groups can use this information to teach students and the public about pollution, its sources, and its effects. Interactive workshops and educational programs can leverage the data collected by VisionAir to foster a culture of environmental stewardship and responsibility. This can lead to long-term behavioral changes that benefit the environment.

Community Engagement and Empowerment

Community involvement is integral to the VisionAir project. By providing accessible air quality data, the system empowers residents to act in advocating for cleaner air initiatives. Community-driven efforts, supported by accurate data, can lead to grassroots movements demanding better environmental practices and policies. This democratization of data fosters a sense of ownership and accountability among community members.

References

[1] Livingston, S. J., Kanmani, S. D., Ebenezer, A. S., Sam, D., & Joshi, A. (2023). An ensembled method for air quality monitoring and control using machine learning. *Measurement. Sensors*, *30*, 100914. https://doi.org/10.1016/j.measen.2023.100914

[2] Malleswari, S., & Mohana, T. K. (2022). Air pollution monitoring system using IoT devices: Review. *Materials Today: Proceedings*, *51*, 1147–1150. https://doi.org/10.1016/j.matpr.2021.07.114

[3] Tsujita, W., Yoshino, A., Ishida, H., & Moriizumi, T. (2005). Gas sensor network for air-pollution monitoring. *Sensors and Actuators. B, Chemical*, *110*(2), 304–311. https://doi.org/10.1016/j.snb.2005.02.008

[4] Al-Haija, Q. A., Al-Qadeeb, H., & Al-Lwaimi, A. (2013). Case study: Monitoring of AIR quality in King Faisal University using a microcontroller and WSN. *Procedia Computer Science*, *21*, 517–521. https://doi.org/10.1016/j.procs.2013.09.072

[5] Vashishtha, P., & Choudhury, T. (2020). DMAPS: an effective and efficient way for the air purification of the outdoors. *Procedia Computer Science*, *167*, 334–343. https://doi.org/10.1016/j.procs.2020.03.233

[6] Ho, S. (2014). Other applications. In *Elsevier eBooks* (pp. 261–271). https://doi.org/10.1016/b978-0-12-398537-8.00013-4

[7] Sohani, A., Moradi, M. H., Rajski, K., Akhlaghi, Y. G., Zabihigivi, M., Wagner, U., & Koch, T. (2022). Intelligent systems in air pollution research: a review. *Elsevier eBooks*, 59–82. https://doi.org/10.1016/b978-0-323-85597-6.00002-1

[8] Vallero, D. A. (2008). Air quality. In *Elsevier eBooks* (pp. 263–295). https://doi.org/10.1016/b978-012373615-4/50009-1

[9] Lopes, D., Ferreira, J., Rafael, S., Hoi, K. I., Li, X., Liu, Y., Yuen, K., Mok, K. M., & Miranda, A. I. (2024). High-resolution multi-scale air pollution system: Evaluation of modelling performance and emission control strategies. *Journal of Environmental Sciences/Journal of Environmental Sciences*, *137*, 65–81. https://doi.org/10.1016/j.jes.2023.02.046

[10] Sudhanshu, P., Debanshi, & P., Ena, J. (2022) *Effectiveness of machine learning Algorithms in forecasting AQI*. (2021, November 10). IEEE Conference Publication | IEEE Xplore. https://ieeexplore.ieee.org/document/9673385

[11] Idrees, Z., & Zheng, L. (2020). Low cost air pollution monitoring systems: A review of protocols and enabling technologies. *Journal of Industrial Information Integration*, *17*, 100123. https://doi.org/10.1016/j.jii.2019.100123

[12] S. Vishwas *et al.*, “Design and Development of Arduino Based Portable Air Quality Monitoring Systems,” *2021 1st Odisha International Conference on Electrical Power Engineering, Communication and Computing Technology(ODICON)*, Jan. 2021, doi: https://doi.org/10.1109/odicon50556.2021.9428954.

[13] “Implementation of Ammonia Gas Leakage Detection & Monitoring System using Internet of Things,” *IEEE Conference Publication | IEEE Xplore*, Dec. 01, 2019. https://ieeexplore.ieee.org/document/9032577

[14] “Design of temperature and humidity measurement system based on single chip microcomputer,” *IEEE Conference Publication | IEEE Xplore*, Dec. 01, 2021. https://ieeexplore.ieee.org/document/9723718

[15] World Health Organization. "Ambient (outdoor) air pollution." Available: https://www.who.int/news-room/fact-sheets/detail/ambient-(outdoor)-air-quality-and-health

[16] Environmental Protection Agency. "Air Quality Monitoring: A Necessity for Clean Air." Available: https://www.epa.gov/air-quality-management

[17] European Environment Agency. "Air Pollution: Real-time Monitoring and Management." Available: https://www.eea.europa.eu/themes/air/air-quality-management

[18] European Public Health Alliance. "The Impact of Air Quality on Health and Healthcare Costs." Available: https://epha.org/air-quality-healthcare-costs/

[19] World Economic Forum. "Harnessing IoT for Urban Sustainability." Available: https://www.weforum.org/reports/harnessing-iot-for-urban-sustainability

[20] T. Agarwal, “MQ4 Methane Gas Sensor Datasheet : Working & Its Applications,” ElProCus - Electronic Projects for Engineering Students, Nov. 03, 2021. https://www.elprocus.com/mq4-methane-gas-sensor/

[21] “MQ-6 Gas Sensor,” Components101. https://components101.com/sensors/mq-6-gas-sensor-pinout-equivalent-datasheet

[22] Xeohacker, “MQ-7 Carbon Monoxide Sensor: Datasheet, Pinout & Working,” The Engineering Projects, Mar. 02, 2024. https://www.theengineeringprojects.com/2024/02/mq-7-carbon-monoxide-sensor-datasheet-pinout-working.html

[23] Xeohacker, “MQ-9 Gas Sensor: Datasheet, Pinout & Working,” The Engineering Projects, Mar. 19, 2024. https://www.theengineeringprojects.com/2024/03/mq-9-gas-sensor.html