AIR QUALITY MONITORING – PHASE III

TEAM MEMBER : RAMESH T

(911721104078)

Phase 3 : 1 Submission Document
Project : Air Quality Monitoring
Phase 3 : Development part 1

INNOVATION :

Air pollution has become a common phenomenon everywhere. Specially in the urban areas, air pollution is a real-life problem. A lot of people get sick only due to air pollution. In the urban areas, the increased number of petrol and diesel vehicles and the presence of industrial areas at the outskirts of the major cities are the main causes of air pollution. The problem is seriously intensified in the metropolitan cities. Also, the climate change is now apparent. The governments all around the world are taking every measure in their capacity.

Select Appropriate Sensors: Choose sensors for detecting specific pollutants.

Calibration: Adjust sensor responses for accuracy.

Sensor Placement: Strategically position sensors near pollution sources.

Data Collection: Collect real-time air quality data. Data

Transmission: Send data to a central platform wirelessly.

Data Storage: Securely store collected data for analysis.

Data Analysis: Process and interpret sensor data.

Real-time Monitoring: Continuously track pollutant levels.

Pollutant Identification Algorithms: Develop algorithms to identify pollutants.

Data Fusion: Combine data from various sources for accuracy.

Alerts and Notifications: Notify authorities and the public of high pollutant levels.

Integration with GIS: Map and visualize pollution data geospatially.

Validation and Quality Assurance: Ensure data accuracy through validation.

Continuous Improvement: Refine algorithms and maintain sensors.

INTRODUCTION :

The quality of the air we breathe is a fundamental determinant of our well-being and the health of the planet. In an age of growing environmental awareness, the need for accurate and real-time air quality data has never been more significant. This is where the convergence of technology and environmental science, through the Internet of Things (IoT), takes center stage.

IoT, with its ability to interconnect devices, sensors, and systems, is revolutionizing the way we monitor and manage our environment. At the forefront of this technological revolution is the domain of air quality measurement through IoT—a powerful and dynamic field that brings together innovation, data, and environmental stewardship.

This project embarks on a journey into the heart of IoT-based air quality measurement, exploring the incredible potential and impact of this technology. We delve into the intricacies of creating a comprehensive air quality monitoring system that harnesses the power of IoT to provide real-time insights into the air we breathe.

Our mission is to equip you with the knowledge, tools, and practical skills needed to construct your own IoT-based air quality measurement solution. From understanding the nuances of air quality parameters to designing the hardware, implementing the software, and analyzing data in the cloud, this project will serve as your guide on a path toward a cleaner, healthier future.

In the chapters that follow, we will unravel the intricacies of air quality monitoring, dissect the components of IoT technology, and explore the practical steps to set up a robust and scalable air quality measurement system. With this knowledge, you'll be empowered to contribute to the ongoing effort to safeguard human health, protect the environment, and shape a more sustainable world.

So, let us embark on this enlightening journey into the world of IoT-driven air quality measurement, where data, technology, and environmental consciousness converge to create a brighter and cleaner future. --- Feel free to customize this introduction according to your project's specific objectives and audience.

Creating a full IoT project for air quality measurement involves several components and steps. Here's an overview of how you can build such a project:

Components: 1. Air Quality Sensors: You'll need sensors to measure air quality parameters. Common sensors include those for measuring particulate matter (PM2.5 and PM10), carbon dioxide (CO_2), carbon monoxide (CO), ozone (O_3), and nitrogen dioxide (NO_2).

2. Microcontroller: Use a microcontroller (e.g., Arduino, Raspberry Pi, or ESP8266/ESP32) to collect data from the sensors, process it, and transmit it to the cloud.

3. Connectivity: Choose a suitable connectivity method like Wi-Fi, Bluetooth, or LoRa for transmitting data to the cloud.

4. Cloud Platform: Set up a cloud platform (e.g., AWS, Azure, or Google Cloud) to receive and store the sensor data. You can also use IoT-specific platforms like AWS IoT or Azure IoT.

5. Database: Create a database to store the collected data for historical analysis. 6. Web or Mobile App: Develop a web or mobile application to visualize the air quality data. Users can access this app to view real-time and historical data.

7. User Interface: Design a user-friendly interface to display air quality information using charts, graphs, and alerts.

Steps:

1. Sensor Integration: Connect the air quality sensors to your microcontroller. Write code to collect data from these sensors.

2. Data Processing: Process the sensor data to ensure it's accurate and meaningful. You may need to convert sensor readings into standard air quality indices.

3. Connectivity Setup: Configure the microcontroller to connect to the internet via Wi-Fi, Ethernet, or any other suitable means.

4. Cloud Integration: Set up your cloud platform and establish a connection between the microcontroller and the cloud. Use secure protocols for data transmission.

5. Data Storage: Store the data in a database on the cloud platform. Ensure data security and integrity.

6. Application Development: Create a web or mobile app for users to access air quality data. Use APIs to retrieve data from your cloud platform.

7. Visualization: Display air quality data using charts and graphs in your application. Implement features like historical data access and real-time alerts.

8. User Alerts: Implement notifications and alerts in the app to inform users when air quality reaches specific thresholds.

9. Testing: Test the entire system for accuracy, reliability, and security.

<u>10. Deployment:</u> **Deploy the IoT devices in the locations where you want to monitor air quality.**

<u>11. Maintenance:</u> **Regularly maintain the system, including sensor calibration and software updates.**

Remember that IoT projects can vary in complexity depending on your requirements. This is a high-level overview, and you may need to dive deeper into each step and use specific hardware and software technologies based on your project's scope and goals.

PROGRAM :

```
#include <Wire.h>
#include <Adafruit_Sensor.h>
 #include <Adafruit_BME680.h>
#define MQ135_PIN A0 // Analog pin for MQ-135 sensor
#define READ_INTERVAL 5000 // Read sensor data every 5 seconds
#define MQ_SAMPLE_TIMES 5
 #define MQ_SAMPLE_INTERVAL 50
 Adafruit_BME680 bme; // Create a BME680 sensor instance
 void setup() { Serial.begin(9600);
 if (!bme.begin(0x76))
 { Serial.println("Could not find a valid BME680 sensor, check wiring!");
 while (1); } } void loop()
 {
 float mq135Value = readMQ135();
float temperature = bme.readTemperature();
 float humidity = bme.readHumidity();
 float pressure = bme.readPressure() / 100.0F;
 Serial.print("MQ-135: ");
 Serial.print(mq135Value);
 Serial.println(" ppm");
Serial.print("Temperature: ");
Serial.print(temperature);
 Serial.println(" *C");
Serial.print("Humidity: ");
Serial.print(humidity);
 Serial.println(" %");
 Serial.print("Pressure: ");
 Serial.print(pressure);
 Serial.println(" hPa");
// Here you can send the data to a cloud platform or display it as needed
delay(READ_INTERVAL);
}
float readMQ135()
{
```

```
float rs = 0;
for (int i = 0; i < MQ_SAMPLE_TIMES; i++)
{
rs += (1023.0 / analogRead(MQ135_PIN) - 1.0);
delay(MQ_SAMPLE_INTERVAL);
} rs = rs / MQ_SAMPLE_TIMES; float ratio = rs / (1023.0 - rs);
float mq135Value = 116.6020682 * pow(ratio, -2.769034857);
return mq135Value;
}
```

OUTPUT :

```
MQ-135: XX ppm
Temperature: XX.XX *C
Humidity: XX.XX %
Pressure: XXXX.XX hPa
```

Certainly, here are some additional details about an IoT air quality measurement project:

Calibration: -
Calibration is a critical aspect of air quality measurement. Sensors may drift over time, and environmental conditions can impact their accuracy. Regular calibration and maintenance are essential to ensure reliable data.

Power Management: -
Depending on the deployment location, you may need to consider power management. Battery-powered IoT devices need to optimize power consumption for extended operation, while mains-powered devices should have backup solutions in case of power outages.

Data Security: -
Protecting air quality data is vital. Ensure data encryption during transmission, access control, and data integrity in storage. Compliance with data privacy regulations may also be required, depending on the project's location.

Alerts and Notifications: -
Implementing alert mechanisms is crucial. When air quality levels exceed predefined thresholds, the system should trigger alerts through SMS, email, or push notifications to inform users or authorities.

Geolocation: -
In some cases, it's essential to associate air quality data with precise geographic locations. GPS modules can be added to your devices for geolocation tagging.

Scalability: - Consider scalability when designing your system. As more sensors are deployed, your IoT infrastructure should be capable of handling an increasing amount of data and devices.

Data Analysis: - The data collected can be analyzed for insights. This could involve identifying air quality trends, pollution sources, and their impact over time. Machine learning models can assist in predictive analysis.

Community Engagement: - Involve the community in your project. Share air quality data with the public and encourage participation. Crowdsourced data can enhance the project's effectiveness.

Regulatory Compliance: - Understand the regulatory requirements for air quality monitoring in your region. Compliance with local and national standards may be necessary, and your project may need to meet specific reporting requirements.

Data Visualization: - User-friendly data visualization is essential for conveying air quality information effectively. Graphs, maps, and historical data comparisons can make the data more understandable.

Cost Considerations: - Be mindful of costs, especially if you plan to deploy multiple sensors. The choice of sensors, microcontrollers, and cloud platforms can significantly impact your project's budget. Real-time Data: - For real-time applications, ensure that data is collected and transmitted with minimal latency. Users should be able to access current air quality data as quickly as possible.

Weather and Environmental Data: - Consider integrating weather data into your project. Weather conditions can impact air quality, so having access to meteorological data can provide a more comprehensive view of environmental conditions.

Public Awareness and Education: - Use your project to raise public awareness about air quality issues. Educate the community about the importance of clean air and how individuals can contribute to better air quality.

Remember, the complexity and features of your IoT air quality measurement project will depend on your specific goals, budget, and the environmental conditions of your deployment location. It's crucial to plan carefully, iterate, and continually improve the project to meet your objectives.

CIRCUIT WORKS AS:

The device developed in this project can be installed near any Wi-Fi hotspot in a populated urban area.

As the device is powered, the Arduino board loads the required libraries, flashes some initial messages on the LCD screen and start sensing data from the MQ-135 sensor.

The sensitivity curve of the sensor for different combustible gases is already mentioned above.

The sensor can be calibrated so that its analog output voltage is proportional to the concentration of polluting gases in PPM.

The analog voltage sensed at the pin A0 of the Arduino is converted to a digital value by using the in-built ADC channel of the Arduino.

The Arduino board has 10-bit ADC channels, so the digitized value ranges from 0 to 1023.

The digitized value can be assumed proportional to the concentration of gases in PPM.

The read value is first displayed on LCD screen and passed to the ESP8266 module wrapped in proper string through virtual serial function.

The Wi-Fi module is configured to connect with the ThingSpeak IOT platform.

ThingSpeak is an IOT analytics platform service that allows to aggregate, visualize and analyze live data streams in the cloud.

ThingSpeak provides instant visualizations of data posted by the IOT devices to ThingSpeak server. With the ability to execute MATLAB code in ThingSpeak one can perform online analysis and processing of the data as it comes in.

CONCLUSION:

In conclusion, the innovation of an IoT-based air quality monitoring device represents a significant step forward in addressing environmental and health concerns. These devices offer real-time data collection and analysis, providing valuable insights into air pollution levels and enabling informed decision-making.

With their ability to connect to the internet and share data seamlessly, they enhance our ability to monitor air quality on a large scale, potentially leading to improved public health outcomes, reduced environmental impact, and more sustainable urban planning. However, it's important to continue refining and expanding these devices to ensure their accuracy, accessibility, and affordability, as well as to foster collaboration between stakeholders to effectively tackle air quality challenges globally. The future of air quality monitoring holds great promise, driven by ongoing innovations in Iot technology and data analytics.