AIR QUALITY MONITORING

SYSTEM

IoT based Air Pollution Monitoring System

he IoT-based air pollution monitoring system provides several benefits over traditional air pollution monitoring systems. It can collect real-time data from multiple locations, which then analyzed to identify the sources of pollution. It helps to take necessary measures to reduce it.

The system can also alert the users if the air quality reaches a dangerous level, allowing them to take precautions to protect themselves.

Read More about Ambient Air Quality Testing

1.IoT Monitoring System components

IoT-based air pollution monitoring systems comprise several components that work together to collect and analyze air quality data. The components include:

Sensors，Microcontroller ,Cloud ServerPower Supply

2.Working:

The sensors measure air quality parameters, the microcontroller processes the data, the communication module sends the data to the cloud server, the cloud server stores and analyzes the data, and the power supply and enclosure provide power and protection to the system. By working together, these components enable the development of accurate and reliable air pollution monitoring systems.

3.Usage of Monitoring System

The IoT-based air pollution monitoring system can be used in various settings, including residential, industrial, and urban areas. It can also be integrated with existing air pollution monitoring systems to enhance their capabilities. The system can provide valuable data to government agencies, researchers, and the public to make informed decisions about air pollution

4.Advantages:

One of the significant advantages of an IoT-based air pollution monitoring system is its scalability. The system can be easily scaled up or down based on the needs of the users. It can be customized to meet the specific requirements of a particular location, making it a versatile solution for air pollution .

5.Working :

IoT (Internet of Things) plays a crucial role in reducing air pollution through its ability to collect real-time data and enable smart decision-making. IoT devices, such as air quality sensors, can monitor pollutant levels in various environments, including cities, industries, and homes.

This data can be analyzed to identify pollution sources, implement targeted mitigation strategies, and track the effectiveness of pollution control measures. IoT-enabled smart city solutions optimize transportation, waste management, and energy consumption, reducing emissions and improving air quality.

6.Implementation:

An IoT-based air and sound pollution monitoring system is implemented using a network of sensors, connectivity technologies, and data analytics platforms. Air quality sensors are deployed in strategic locations to measure pollutant levels such as particulate matter, gases, and volatile organic compounds (VOCs). Sound sensors capture noise levels and patterns in the environment.

These sensors are connected to a central data management system through wireless or wired communication protocols. The collected data is then processed and analyzed in real-time, leveraging cloud-based analytics platforms. Users can access the monitoring system through web or mobile applications, which provide visualizations, alerts, and historical data.

This allows authorities, environmental agencies, and individuals to monitor pollution levels, identify hotspots, and take necessary actions for pollution control and mitigation. The system can also integrate with existing infrastructure such as smart city platforms or industrial monitoring systems to provide a comprehensive view of environmental conditions and enable effective decision.

SENSOR CONNECTIONS:

Connect the sensor's pins to the corresponding pins on your Raspberry Pi or Arduino.

Sensor VCC to Raspberry Pi 3.3V (or Arduino 3.3V)

Sensor GND to Raspberry Pi GND (or Arduino GND)

Sensor SDA to Raspberry Pi SDA (or Arduino A4 on an Arduino Uno)

Sensor SCL to Raspberry Pi SCL (or Arduino A5 on an Arduino Uno)

CLOUD OR MOBILE APP INTEGRATION:

On your cloud server or mobile app, set up an endpoint to receive data from the Raspberry Pi. Process the incoming data and update the parking space occupancy status accordingly.

Ensure that your server has proper security measures, like HTTPS and authentication, to protect the data transmission.

TESTING:

Run the Python script on your Raspberry Pi by executing python example.py in the terminal.

Monitor the output to ensure the correct distance measurements are being obtained.

Verify that data is being sent to your cloud server or mobile app and that the occupancy status is being updated correctly.

Depending on our project requirements, we might need to add more sensors, implement error handling, or optimize the code for better performance andreliability. Additionally, security considerations are crucial, especially if we are dealing with sensitive data

SOURCE CODE:  
program

int analogPin = A0;  // Define the analog pin for the gas sensor

int threshold = 400; // Set a threshold value to trigger air quality warnings

void setup() {

**Serial**.begin(9600);

}

void loop() {

  int sensorValue = analogRead(analogPin);

**Serial**.println(sensorValue);

  if (sensorValue > threshold) {

**Serial**.println("Air quality is poor! Open windows and ventilate.");

  } else {

**Serial**.println("Air quality is good.");

  }

  delay(10000); // Delay for 10 seconds before the next reading

}

Sample output

165

Air quality is good.

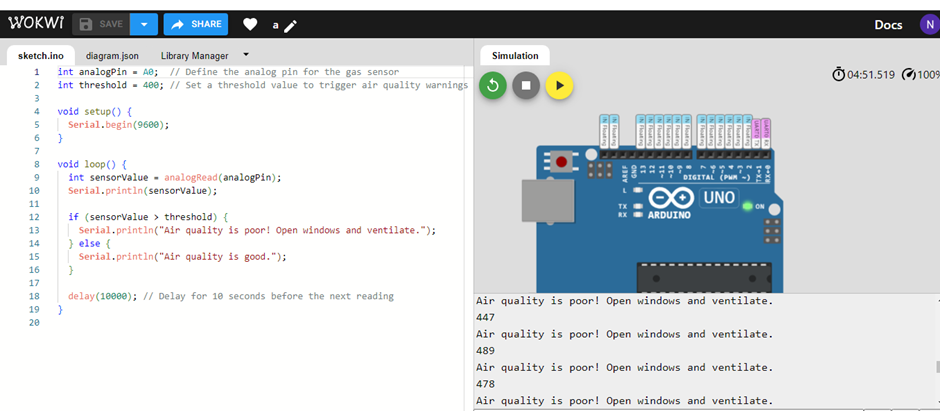
205

Air quality is good.

WEBSITE LINK:

<https://project66.mydurable.com/>

SIMULATION SITE:



WEBSITE DEPICTION:

