Objectives:

Monitoring air quality using IoT (Internet of Things) technology can be a crucial initiative to ensure a healthier environment. The objectives of air quality monitoring using IoT

Iot setup devices:

1) sensors

2) Microcontroller (such as Arduino or Raspberry Pi)

3) Communication module (WIFI and GSM)

Hardware Components:

1. **Air Quality Sensors:**

- **Particulate Matter (PM) Sensors:** Measure PM2.5 and PM10 levels in the air.

- **Gas Sensors:** Measure specific gases like CO2, CO, SO2, NO2, and O3.

- **Temperature and Humidity Sensors:** Measure environmental conditions.

2. **Microcontroller:**

- **Arduino:** Arduino boards (such as Arduino Uno or Arduino Nano) can be used for data processing and interfacing with sensors.

- **Raspberry Pi:** Raspberry Pi can be used for more complex applications, data storage, and remote monitoring capabilities.

3. **Communication Module:**

- **Wi-Fi Module:** IoT devices can connect to the internet using Wi-Fi modules (like ESP8266 or ESP32), enabling data transmission to the cloud or a local server.

- **GSM/3G/4G Module:** In areas without stable Wi-Fi connectivity, GSM, 3G, or 4G modules can be used for cellular data transmission.

4. **Power Supply:**

- **Battery or External Power Supply:** Depending on the deployment location, the system can be powered by batteries or connected to an external power source.

Software specifications:

1. **Data Acquisition:*

- Specify the types of air pollutants to be monitored (e.g., particulate matter, carbon dioxide, ozone).

- Define the sampling frequency and accuracy requirements for each pollutant.

- Detail the sensors and data acquisition devices to be used.

2. **Data Transmission:**

- Determine the communication protocols (e.g., MQTT, HTTP) for transmitting data from sensors to the central server.

- Specify the frequency of data transmission and the method of error handling.

3. **Data Processing and Storage:**

- Describe how raw sensor data will be processed, filtered, and converted into meaningful air quality metrics.

- Specify the database system and storage requirements for storing the processed data securely.

4. **User Interface:**

- Define the design and functionality of the user interface (web application, mobile app) for real-time monitoring.

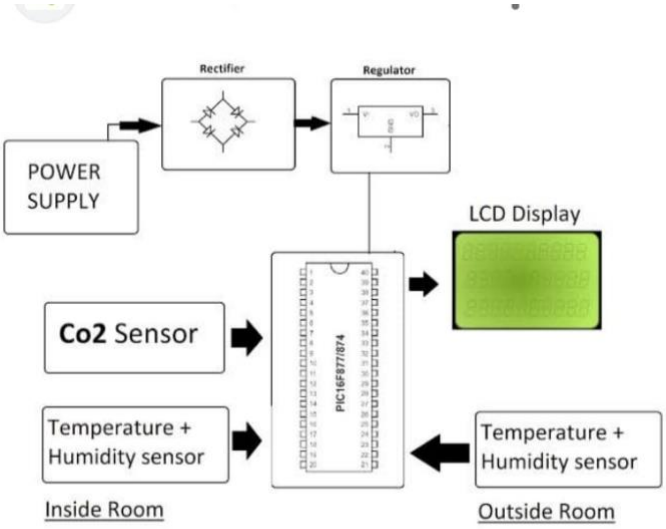
- Specify the features such as historical data analysis, visualization tools, and user alerts.

Hardware Setup :

- Raspberry Pi (or any other microcontroller platform)

- MQ-135 Air Quality Sensor

- Jumper wires and breadboar

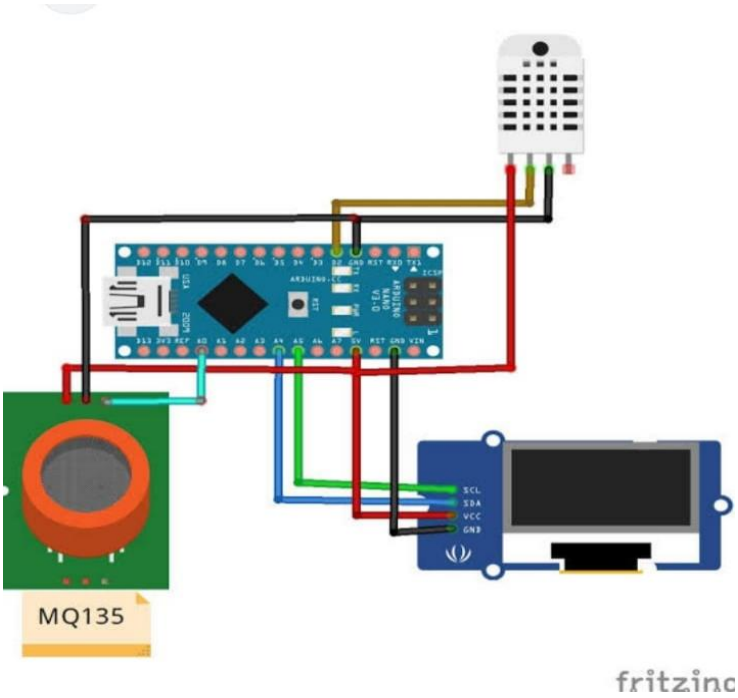


Sensor Deployment these Various sensors are placed strategically in different locations to measure pollutants in the air. These sensors can detect substances like particulate matter (PM2.5 and PM10), ozone (O3), nitrogen dioxide (NO2), carbon monoxide (CO), sulfur dioxide (SO2), and volatile organic compounds (VOCs).

Data Collection is Sensors collect real-time data on the concentrations of these pollutants in the atmosphere. This data is usually collected at regular intervals, providing a continuous stream of information.

Data Transmission is Collected data is transmitted to a central server or database using wired or wireless communication methods. This allows for remote monitoring and analysis.Data Processing is The received data is processed and analyzed to interpret pollutant levels. Statistical methods and machine learning algorithms are often used to identify patterns, trends, and anomalies in the data.

The data virturalization is Processed data is presented in a user-friendly format through web interfaces or mobile apps. Visualization tools like charts, graphs, and maps help stakeholders understand the air quality status easily.




```

Python Source Code:

```python
import time
import Adafruit_MQ
import Adafruit_MQ.MQ135 as MQ135

# Adafruit IO credentials and feeds
ADAFRUIT_IO_KEY = 'YOUR_AIO_KEY'
ADAFRUIT_IO_USERNAME = 'YOUR_AIO_USERNAME'
AIO_FEED = 'air-quality'

# Set up Adafruit IO MQTT client
aio = Client(ADAFRUIT_IO_USERNAME, ADAFRUIT_IO_KEY)
```
```

```
# Set up MQ-135 sensor
MQ135_PIN = 17  # Pin connected to the MQ-135 sensor
mq135 = MQ135.MQ135(MQ135_PIN)

while True:
    air_quality = mq135.read_adc()
    print('Air Quality: {}'.format(air_quality))

    # Publish data to Adafruit IO feed
    aio.send_data(AIO_FEED, air_quality)
    time.sleep(10)  # Delay for 10 seconds before the next reading
```

Replace `'YOUR_AIO_KEY'` and `'YOUR_AIO_USERNAME'` with your Adafruit IO credentials. Ensure you have created a feed named `'air-quality'` on Adafruit IO to store the air quality data.

Running the Code:

Run the Python script on your Raspberry Pi. It reads air quality data from the MQ-135 sensor and publishes it to the specified Adafruit IO feed

Conclusion:

In implementing air quality monitoring using IoT (Internet of Things) technology offers significant advantages in understanding and addressing environmental concerns. By leveraging sensors and connectivity, IoT enables real-time data collection, analysis, and reporting, allowing for informed decision-making and proactive measures to improve air quality. This technology empowers governments, organizations, and communities to track pollution levels, identify sources, and implement effective policies to safeguard public health and the environment. As IoT continues to advance, its integration in air quality monitoring will play a pivotal role in creating cleaner and healthier environments for future generations.