

Internet Of Things - Group 1

AQM

PHASE 3 – DEVELOPMENT PART 1

Team members:

Saraniya.S

Sushma.K

Divya Geetha.N

Kavya Aarthi.G

Project Title: AQM

Project Definition: The project involves setting up IoT devices to measure air quality parameters and make the data publicly available for raising awareness about air quality and its impact on public health. The objective is to create a platform that provides real-time air quality information to the public. This project includes defining objectives, designing the IoT monitoring system, developing the data-sharing platform, and integrating them using IoT technology and Python.

Phase 3: Development Part 1

Configuration of sensors

PM2.5 Sensor (Particulate Matter with a diameter of 2.5 micrometers or smaller):

- Measures fine particulate matter in the air.
- Detects tiny particles that can penetrate deep into the respiratory system.
- Used for monitoring air pollution, indoor air quality, and assessing health risks.
- Common technologies include optical scattering and laser-based methods.
- Provides data in micrograms per cubic meter ($\mu\text{g}/\text{m}^3$).
- Crucial for tracking health-related air quality concerns.

PM10 Sensor (Particulate Matter with a diameter of 10 micrometers or smaller):

- Measures larger particulate matter than PM2.5 sensors.

- Includes particles that are still inhalable but can impact respiratory health.
- Used for assessing outdoor air quality, construction site monitoring, and dust control.
- Typically employs the same detection technologies as PM2.5 sensors.
- Provides data in micrograms per cubic meter ($\mu\text{g}/\text{m}^3$).
- Valuable for evaluating air quality in industrial and construction settings.

CO2 Sensor (Carbon Dioxide):

- Measures the concentration of carbon dioxide in the air.
- Used to assess indoor air quality, ventilation, and energy efficiency.
- Common technologies include non-dispersive infrared (NDIR) and chemical sensors.
- Provides data in parts per million (ppm).
- Important for optimizing HVAC systems and ensuring occupant comfort in buildings.

CO Sensor (Carbon Monoxide):

- Measures the concentration of carbon monoxide gas in the air.
- Primarily used for detecting potentially lethal levels of CO, often produced by incomplete combustion.
- Technologies include electrochemical and metal oxide sensors.
- Provides data in parts per million (ppm).
- Crucial for residential and industrial safety, especially in spaces with combustion appliances.

NO2 Sensor (Nitrogen Dioxide):

- Measures the concentration of nitrogen dioxide gas in the air.
- A significant component of air pollution, primarily from combustion processes.
- Technologies include electrochemical sensors.
- Provides data in parts per billion (ppb).
- Used to monitor urban air quality and assess the impact of vehicle emissions.
- Important for public health and environmental regulatory compliance.

PROGRAMMING:

```
Import paho.mqtt.client as mqtt
```

```
Import time
```

```
Import random # Replace with actual sensor data collection
```

```
# Replace these with your actual IoT device information
```

```
Broker_address = "mqtt.example.com" # MQTT broker address
```

```
Port = 1883 # Default MQTT port
```

```
Username = "your_username" # MQTT broker username
```

```
Password = "your_password" # MQTT broker password
```

```
Topic = "air_quality_data" # MQTT topic where data will be published
```

```
# Simulated data for testing; replace this with your actual data collection code
```

```
Def collect_air_quality_data():
```

```
    # Replace with code to collect air quality data
```

```
Pm25 = random.uniform(0, 100)
```

```
Pm10 = random.uniform(0, 100)
```

```
Co2 = random.uniform(300, 1000)
```

```
Return {"pm25": pm25, "pm10": pm10, "co2": co2}
```

```
# Callback when the client connects to the MQTT broker
```

```
Def on_connect(client, userdata, flags, rc):
```

```
    Print("Connected to MQTT broker with result code " + str(rc))
```

```
# Create an MQTT client
```

```
Client = mqtt.Client()
```

```
Client.username_pw_set(username, password)
```

```
# Set up the on_connect callback
```

```
Client.on_connect = on_connect
```

```
# Connect to the MQTT broker
```

```
Client.connect(broker_address, port, keepalive=60)
```

```
Try:
```

```
    While True:
```

```
        Air_quality_data = collect_air_quality_data()
```

```
        # Convert data to JSON format and publish to the MQTT topic
```

```
        Client.publish(topic, str(air_quality_data))
```

```
        Print("Published: " + str(air_quality_data))
```

```
        Time.sleep(60) # Adjust the interval as needed
except KeyboardInterrupt:
    Client.disconnect()
    Print("Disconnected from MQTT broker")

# Keep the script running to continuously collect and send data
```