

Air Quality Monitoring using IoT with Sensors

Project Definition:

IOT Based Air Pollution Monitoring System monitors the Air quality over a web server using Internet and will trigger an alarm when the air quality goes down beyond a certain threshold level, means when there are sufficient amount of harmful gases present in the air like CO₂, smoke, alcohol, benzene, NH₃, LPG and NO_x.

Hardware Selection

Select the appropriate hardware components for your IoT air quality monitoring system. These may include:

1. Microcontroller or Single-Board Computer (SBC).
2. Air quality sensors such as particulate matter sensors (PM_{2.5} and PM₁₀), gas sensors for detecting carbon dioxide (CO₂), carbon monoxide (CO), and other pollutants.
3. Supporting components like breadboards, jumper wires, resistors, and connectors.

Sensor Integration

Integrate the sensors with the microcontroller or SBC. Follow the datasheets and integration guides provided by the sensor manufacturers. Ensure you have the necessary libraries and drivers installed for the sensors to communicate with the microcontroller.

IoT Device Programming

Write a script or program to collect data from the sensors and transmit it to a data-sharing platform. Use a programming language like Python or C/C++ depending on the capabilities of your microcontroller. Here's an example using Python:

```
def read_sensor_data():  
    while True:  
        sensor_data = read_sensor_data()  
        send_to_data_platform(sensor_data)
```

Data Sharing Platform

Set up a data-sharing platform or cloud service where the collected air quality data can be stored and analyzed. You can use platforms like AWS IoT, Google Cloud IoT, Azure IoT, or custom solutions like MQTT brokers, databases, or web servers.

Ensure that the data-sharing platform has appropriate security measures to protect the transmitted data.

Deployment and Testing

Deploy the IoT air quality monitoring system in the desired location and conduct thorough testing to ensure accurate data collection and transmission. Monitor the system for any issues and make necessary adjustments as required.

To configure IoT devices for measuring air quality parameters, you need to choose appropriate sensors for pollution levels and particulate matter. Here's a general guide to help you configure the IoT devices:

Step 1: Sensor Selection:

1. Particulate Matter Sensors:

These sensors can detect and measure the concentration of particulate matter in the air, categorized as PM2.5 (fine particles) and PM10 (coarse particles). Examples include SDS011, PMS5003, and Nova PM Sensor SDS011.

2. Gas Sensors:

These sensors can detect various gases like carbon dioxide (CO₂), carbon monoxide (CO), nitrogen dioxide (NO₂), sulfur dioxide (SO₂), and ozone (O₃). Some popular gas sensors are the MQ series gas sensors and the Alphasense series.

Step 2: Hardware Setup

Follow the datasheets and guidelines provided by the sensor manufacturers to correctly set up the hardware components. Ensure that the sensors are appropriately connected to the IoT device or microcontroller, and that you have the necessary power supply and connections in place.

Step 3: Software Configuration

Utilize appropriate software libraries and development tools to configure the sensors to communicate with your IoT device. Install the necessary libraries and dependencies to enable the sensors to provide accurate data readings.

Step 4: Calibration

Calibrate the sensors to ensure accurate readings. Follow the calibration procedures provided by the sensor manufacturer and perform regular checks to maintain the accuracy of the sensor data.

Step 5: Data Interpretation

Understand the sensor outputs and their corresponding units of measurement. Interpret the data provided by the sensors to ensure that you accurately capture the air quality parameters you are interested in monitoring.

Step 6: Integration

Integrate the sensor data with the IoT device's data processing capabilities. Ensure that the data is processed and stored in a format that can be easily transmitted to the data-sharing platform or any other monitoring system that you have set up.

Python Program :

```
import requests

import json

import random

import time

# Replace these placeholders with actual values

DATA_ENDPOINT = 'YOUR_DATA_ENDPOINT'

API_KEY = 'YOUR_API_KEY'

# Function to read sensor data (Replace this with your actual sensor reading code)

def read_sensor_data():

    # Example data generation (Replace this with actual sensor data)

    temperature = round(random.uniform(20, 35), 2)

    humidity = round(random.uniform(40, 60), 2)

    return {'temperature': temperature, 'humidity': humidity}

# Function to send data to the data-sharing platform

def send_data_to_platform(data):

    headers = {
```

```

    'Content-Type': 'application/json',
    'Authorization': f'Bearer {API_KEY}'
}

response = requests.post(DATA_ENDPOINT, data=json.dumps(data),
headers=headers)

if response.status_code == 200:
    print("Data sent successfully!")
else:
    print(f"Failed to send data. Status code: {response.status_code}")

# Main execution loop
if __name__ == '__main__':
    while True:
        sensor_data = read_sensor_data()
        send_data_to_platform(sensor_data)
        time.sleep(5) # Adjust the time interval as per your requirement

```

Make sure you have the `requests` library installed to run the script. You can install it using pip:

```

bash

pip install requests

```