DEVELOPMENT PART 1

Air quality monitoring:

Developing an IOT enabled Air quality monitoring requires both hardware and software components.

Hardware Development:

•Sensor:

There are the core components that measure various air quality parameters. Common sensors include those for Particulate matter (PM2.5 and PM10), Volatile organic compounds (VOCs).

particulate matter sensor

• Microcontroller:

Microcontrollers like Arduino or Raspberry Pi are often used for this purpose.

•Communication Module:

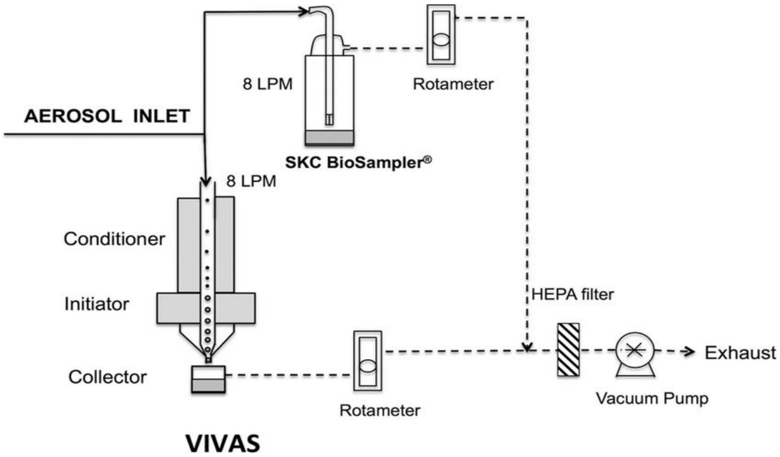
To transmit data to a central server or display it in real time, a communication module is used. This can be Wi-Fi, cellular or Ethernet connectivity.

•Power supply:

Depending on the location and purpose of the system, power can be supplied through mains , batteries, or even solar panels for remote installations.

•Airflow and sampling system:

For accurate measurements, the system may include a sampling mechanism to draw in ambient air. This involve pumps, fans, or passive diffusion methods.



•Central server or cloud connection:

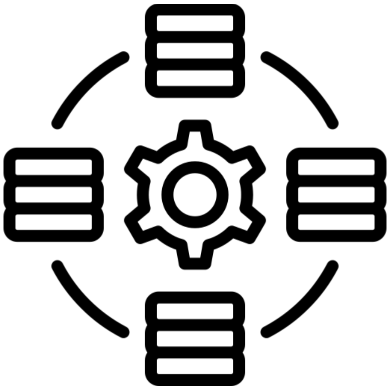
Air quality data is often sent to a central server or cloud-based platform for storage, analysis, and dissemination.

Software development:

•Data processing:

* Algorithms and Models: Process raw data to calculate pollutant concentrations, air quality indices, and other relevant metrics.

* Statistical Analysis Tools: Identify trends, patterns, and correlations in air quality data.



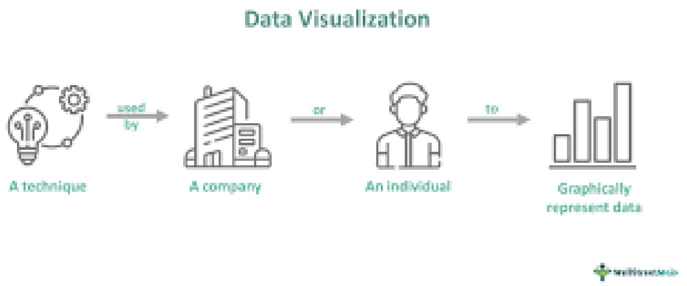
• Communication Software:

*Communication Protocols: Enable data transfer between monitoring stations and central databases.

*Web APIs: Facilitate integration with external systems or applications.

• Visualization and Reporting Tools:

* Graphical User Interface (GUI): Provides a user-friendly interface for operators to visualize real-time and historical data.



* Geographical Information System (GIS) Software: Maps air quality data to geographical locations for spatial analysis.

* Dashboard Software: Presents key metrics and trends in a visually appealing format.

* Reporting Tools: Generate automated reports for regulatory compliance or public dissemination.

•Remote Monitoring and Control Software:

* Remote Monitoring Tools: Allow operators to monitor and control the status of monitoring stations remotely.

* Telemetry Systems: Transmit real-time data from remote locations to a central monitoring.

- Display and User Interface:

User Interface: Displays real-time and historical data for operators and the public. This can be a physical display on the monitoring station or a web-based interface.

PROGRAM:

(python script)

```
import random
import time

def measure_air_quality():
    # Simulating sensor data for demonstration purposes
    pm_value = random.uniform(0, 100)
    co_value = random.uniform(0, 10)
    so2_value = random.uniform(0, 5)

    return pm_value, co_value, so2_value

def display air_quality(pm, co, so2):
    print(f"Air Quality Measurement:")
    print(f"PM: {pm} µg/m³")
    print(f"CO: {co} ppm")
    print(f"SO2: {so2} ppm")
    print()

def air_quality_monitoring_system():
    while True:
        pm, co, so2 = measure_air_quality()
        display_air_quality(pm, co, so2)
        time.sleep(5)  # Adjust the sleep duration as needed

if __name__ == "__main__":
    air_quality_monitoring_system()
```