AIR POLLUTION MONITORING

IOT PHASE-3

HARDWARD COMPONENTS

- **AIR QUALITY SENSORS:**

Air quality sensors are the core components for measuring different air pollutants common sensors include:

- **MQ series sensor (e.g., MQ-7 for CO, MQ-135 for CO2)**
- **Particulate matter (PM) sensors (e.g., SDS011, PMS5003)**
- **Ozone(O3) Sensors**
- **Nitrogen dioxide (NO2) sensors**

- **MICROCONTROLLER:**

NodeMCU (ESP8266) or Arduino boards (e.g., Arduino Uno, Arduino Nano) to interface with sensors, process data and send it to a central server cloud.

- **COMMUNICATION MODULE:**

Wi-Fi module (e.g., ESP8266, ESP32) to connect the microcontroller to the internet, enabling data transmission to the server or cloud.

- **POWER SUPPLY:**

Power source (e.g., battery, power adapter) to power the sensor, microcontroller and communication modules.

- **BREADBOARD AND JUMPER WIRES:**

Breadboards for prototyping and jumper wires to connect components on the breadboard.

- **LCD DISPLAY:**

LCD module (e.g.,16x2 character LCD) to display real time air quality data.

- **ENCLOSURE:**

A protective housing to house and components and protect them from environmental condition (optional, especially for outdoor installations).

OTHER COMPONENTS

- **Resistor:**

Air Quality Monitoring Networks allow the measurement, operation and predictive analysis of the evolution of air pollution in different areas (urban areas, industrial areas, special nature conservation areas, etc.) Some stations are equipped with meteorological sensors and/or noise level meters to measure noise levels.

- **Capacitors:**

Variable air capacitors are used in circumstances where the capacitance needs to be varied. They are sometimes used in resonant circuits, such as radio tuners, frequency mixers or antenna impedance matching applications. Another use for variable capacitors is while prototyping an electronic circuit design.

- **Software code:**

```
Import time

From datatime import datatime

From sds011 import SDS011

 Import matplotlip .pyplot as plt

#initialize thePMsensor

  sds=SDS011("\dev\ttyUSB0")
```

```
        use_query_mode=true)
        def read _pm_sensor():
        “””””Read PM sensor data.”””””
        try:
        pm_data =sds.query():
        pm25,pm10=pm_data
         except Exception as e:
              print(f"Error reading PM sensor: {e}")
              return None

# Data storage (in-memory for this example)
        pm25_data = []
        pm10_data = [] import time
        from datetime import datetime
        from sds011 import SDS011
        import matplotlib.pyplot as plt

# Initialize the PM sensor
        sds = SDS011("/dev/ttyUSB0", use_query_mode=True)

        def read_pm_sensor():
           """Read PM sensor data."""
           try:
              pm_data = sds.query()
              pm25, pm10 = pm_data
```

```
def record_pm_data():
    """Record PM sensor data."""
    global pm25_data, pm10_data
    pm_values = read_pm_sensor()
    if pm_values:
        pm25, pm10 = pm_values
        timestamp = datetime.now()
        pm25_data. append((timestamp, pm25))
        pm10_data. append((timestamp, pm10))

# Data visualization
def plot_pm_data():
    """Plot PM sensor data."""
    timestamps, pm25_values = zip(*pm25_data)
    _, pm10_values = zip(*pm10_data)

    plt.figure(figsize=(10, 6))
    plt.plot(timestamps, pm25_values, label="PM2.5 (µg/m³)")
    plt.plot(timestamps, pm10_values, label="PM10 (µg/m³)")
    plt.xlabel("Timestamp")
    plt.ylabel("Concentration")
    plt.title("Air Pollution Monitoring")
    plt.legend()
    plt.grid(True)
    plt.show()
```

```
# Main loop for data collection
try:
    while True:
        record_pm_data()
        time.sleep(300)  # Record every 5 minutes (adjust as needed)
except KeyboardInterrupt:
    # Plot data on KeyboardInterrupt (Ctrl+C)
    plot_pm_data()
```

- **HARDWARE DESING**