Airr quality monitoring system using an ESP32 microcontroller

Components needed:

ESP32 Microcontroller:

This is the heart of your system, providing the processing power and Wi-Fi capabilities.

Air Quality Sensors:

Choose appropriate sensors to measure parameters like particulate matter (PM2.5 and PM10), carbon dioxide (CO2), volatile organic compounds (VOCs), and more. Some commonly used sensors include the SDS011 for PM levels and the CCS811 for CO2 and VOCs.

Power Supply:

Ensure a stable power supply for your ESP32 and sensors. You can use a USB power source or a battery with a voltage regulator.

Display:

You may want to include an OLED or LED display to show real-time air quality readings.

Internet Connectivity:

ESP32 has built-in Wi-Fi capabilities. Use this to send data to a cloud platform or a local server.

Data Storage/Cloud Platform:

Store and analyze data in a database or a cloud platform like AWS, Google Cloud, or Azure.

Visualization Interface:

Create a web or mobile app to display air quality data in a user-friendly manner.

Steps to design the system:

Hardware Setup:

Connect the air quality sensors to the ESP32 according to their datasheets.

Set up the power supply for your components.

Software Development:

Write code for your ESP32 microcontroller using Arduino IDE or PlatformIO.

Use libraries for the specific sensors you're using.

Implement Wi-Fi connectivity to your local network.

Data Acquisition:

Read data from the sensors at regular intervals.

Store or send this data for further processing.

Data Transmission:

Send the air quality data to your chosen destination (local server or cloud platform) over Wi-Fi. You can use protocols like MQTT or HTTP.

Analayatic Data Storage and Analysis:

Store incoming data in a database or cloud storage.

Implement data analysis and anomaly detection if needed.

Visualization:

Create a user interface (web or mobile app) to display air quality data.

Use frameworks like React, Angular, or Flutter for app development.

Alerting:

Implement alerting mechanisms if air quality readings exceed predefined thresholds.Visualization

Power Management:

Optimize power consumption for longer battery life if you're using a battery.

Calibration and Maintenance:

Periodically calibrate the sensors and perform system maintenance.

Security:

Ensure data security and user privacy, especially if you're transmitting data over the internet.