ESP32 Environmental Monitoring System

Project Overview

This project presents an IoT-based environmental monitoring system using the ESP32-WROOM-32 board. It collects environmental data such as temperature, humidity, pressure, and altitude using the BME280 sensor and displays it via an OLED screen and a web server. The collected data is also saved to an SD card in CSV format for further analysis using Excel.

Objectives

- Measure temperature, humidity, pressure, and altitude using BME280
- Display data on an OLED screen
- Create a local web server with ESP32 to show data in real time
- Save data in CSV format to SD card
- Analyze and plot the data using Excel

Applications

- Environmental monitoring in greenhouses, server rooms, and labs
- Remote monitoring through Wi-Fi access
- Long-term data collection for statistical analysis
- Education and research in IoT and microcontroller systems

Components Used

- ESP32-WROOM-32 board
- BME280 sensor (I2C)
- SSD1306 OLED display (I2C)
- SD card reader module (SPI)
- Jumper wires, breadboard

Wi-Fi Modes of ESP32

ESP32 supports multiple Wi-Fi modes:

- Station Mode (STA): ESP32 connects to an existing Wi-Fi network.
- Access Point (AP): ESP32 creates its own Wi-Fi hotspot.
- AP + STA Mode: Both modes enabled simultaneously.

Sensor Data Reading Code (BME280)

```
float temp = bme.readTemperature();
float pressure = bme.readPressure() / 100.0F;
float humidity = bme.readHumidity();
float altitude = bme.readAltitude(SEALEVELPRESSURE_HPA);
```

OLED Display with I2C

OLED screens use the SSD1306 driver with I2C communication. They offer high contrast, don't require backlight, and consume less power.

Libraries used:

```
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
```

Displaying Icons on OLED

Icons like thermometer symbols can be displayed using bitmaps.

Example (16x16 icon):

```
const unsigned char icon_temp[] PROGMEM = {...};
```

SD Card Storage

SD card modules use SPI communication. Data is saved as .csv format using Arduino's built-in SD library.

Example:

```
File dataFile = SD.open("/data.csv", FILE_APPEND);
...
```

SPI Communication Overview

- SCLK (Serial Clock): Synchronization clock
- MOSI (Master Out Slave In): Data from master to slave
- MISO (Master In Slave Out): Data from slave to master
- SS (Slave Select): Enables communication with specific slave

Data Logging in Proteus Simulation

Data collected from the BME280 sensor is logged to a file called 'data.csv'. This file can later be imported into Excel for analysis.

Ensure the SD card is formatted to FAT32.

Final Outputs

- Real-time display on OLED
- Web server with live sensor data
- CSV data file for long-term analysis
- Excel graphs plotted from stored data