ESP32-S3 Web Server for AP and Station Mode

Introduction

This report details the development and implementation of an ESP32-S3 project that features a web server interface to control RGB LEDs, display DHT11 sensor data, and show messages on an OLED display. Additionally, the project includes an interactive web-based module that performs additional tasks, such as a real-time Wi-Fi network scanning.

Objectives

- Develop a web server that functions in both Access Point (AP) and Station modes.
- Control an RGB LED via a web interface by allowing users to input Red, Green, and Blue values.
- Display real-time temperature and humidity data from the DHT11 sensor on the web page.
- Show custom messages on an OLED display interfaced with the ESP32.
- Implement an additional feature such as a web-based calculator or interactive control panel.
- Ensure the web pages have a responsive and visually appealing design using CSS.

Hardware Requirements

- ESP32-S3 development board
- DHT11 temperature and humidity sensor
- OLED display (SSD1306)
- RGB LED (onboard or external)
- Power supply (USB or battery)
- Jumper wires and breadboard

Software Requirements

- MicroPython firmware for ESP32-S3
- Thonny with ESP32 plugins
- MicroPython libraries: machine, network, socket, time, dht, ssd1306, ujson
- HTML, CSS, and JavaScript for the web interface

Implementation

1. Setting Up the ESP32-S3 with MicroPython

- Install MicroPython firmware on the ESP32-S3.
- Connect the ESP32-S3 to a computer and program it using Thonny.
- Install necessary libraries and upload them to the ESP32 for handling peripherals.

2. Web Server Development for RGB LED Control

- The ESP32 hosts a web server with an interface where users can adjust LED colors by entering RGB values.
- Submitted values are processed and sent to the RGB LED for color adjustment.

3. Displaying DHT11 Sensor Data

- The ESP32 continuously reads temperature and humidity data from the DHT11 sensor.
- Data is updated on the web page every few seconds to provide real-time monitoring.

4. OLED Display Integration

- The OLED display shows system status, sensor readings, and custom messages sent via the web page.
- Ensures clear visualization for real-time monitoring without needing a web interface.

5. Additional Feature: Web-Based Wi-Fi Scanner.

- A Wi-Fi network scanning feature is included to list available networks and allow the user to select and connect to one.

Testing and Debugging

- Debugging was performed using serial output to monitor real-time data from the ESP32.
- The OLED display was tested for proper message updates.
- The web server was tested in both AP and Station modes to ensure stable connectivity.
- Various input scenarios for LED control and sensor readings were tested for accuracy and reliability.

Conclusion

This project successfully integrates multiple functionalities into the ESP32-S3, demonstrating its versatility as an IoT-enabled web server. The implementation of LED control, real-time sensor data display, OLED messaging, and additional web-based features highlights the ESP32's capability for responsive and interactive applications.