**ESP32 Project Report**

**1. Introduction**

This report details the development and implementation of an ESP32-S3 project with a web server interface to control RGB LEDs, display DHT11 sensor data, and show messages on an OLED display. Additionally, the project includes a web-based calculator hosted on the ESP32-S3, which performs calculations based on user input and displays the results on both the web server and the OLED display.

**2. Objectives**

* Control the onboard RGB LED of the ESP32-S3 using a web interface.
* Display temperature and humidity data from the DHT11 sensor on a web server.
* Show custom messages on an OLED display.
* Develop a web-based calculator that takes user input, processes calculations on the ESP32, and displays results on both the web server and OLED.

**3. 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

**4. 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

**5. Implementation**

**5.1 Setting Up the ESP32-S3 with MicroPython**

1. Install MicroPython on the ESP32-S3.
2. Connect the ESP32-S3 to a computer and use Thonny to program it.
3. Install necessary libraries using mpy-cross and upload them to the ESP32.

**5.2 Controlling RGB LED via Web Server**

* The ESP32 hosts a web server with buttons to change LED colors.
* The color selection is handled via HTML, and the ESP32 executes the corresponding LED control commands.

**5.3 Displaying DHT11 Sensor Data**

* The ESP32 reads temperature and humidity data and updates the web server every few seconds.

**5.4 OLED Display Integration**

* The OLED is used to display temperature, humidity, and custom messages received

**5.5 Web-Based Calculator on ESP32**

* The ESP32 hosts a web interface where users enter values and an operation.
* The ESP32 processes the input, computes the result, and displays it on both the web page and the OLED.

**6. Testing and Debugging**

* Debugging was done using serial output to monitor sensor readings and web server responses.
* The OLED display was checked for real-time updates.
* Various input scenarios were tested for the calculator.

**7. Conclusion**

This project successfully integrates multiple features into the ESP32-S3, including LED control, sensor data display, and a functional web-based calculator. The implementation demonstrates the ESP32's capability as a lightweight IoT server handling real-time interactions.