Home Lighting Controller with Wi-Fi Connectivity

Aseka Attanayake

2024/AE/032

01. Introduction

This project presents a battery-powered smart monitoring and control device based on the ESP32 microcontroller. The primary objective of the system is to measure environmental parameters such as temperature and humidity using a DHT22 sensor and to control four relays which can be connected to lights, in this project for Living Room, Bedroom, Kitchen, and Garden). The system is designed with energy efficiency and reliable wireless connectivity, enabling reliable data collection and remote control even under power or network constraints.

The device is powered by two 18650 Li-ion batteries and includes power monitoring functionality to detect low-voltage conditions and enter deep sleep mode to conserve energy. It connects to the internet via Wi-Fi and synchronizes time from an NTP server. Sensor data is transmitted using MQTT, HTTP, and UDP protocols, allowing flexibility in communication with different server and client types. A web-based interface is provided for real-time monitoring, relay control, and schedule configuration. The circuit supports offline operation with wake-up functionality and offers an OLED display for visualization of sensor data, time, battery level, and relay status.

02. User Instructions

To use the device, power it using the built-in battery system or charge the batteries using the buck booster module via USB. Upon startup, the device automatically connects to the configured WiFi network. Once connected, it retrieves the current time from an NTP server and begins reading temperature and humidity values from the DHT22 sensor.

Users can monitor readings and control relays in multiple ways:

- By visiting the device's IP address in a web browser to access the HTTP interface.
- By subscribing to relevant MQTT topics to both receive data and publish control commands.
- By sending control commands via UDP from a terminal or application.

Users can schedule the on/off times of the garden relay through the web interface. All relay states and the schedule persist during normal operation. If the battery voltage falls below a predefined threshold (3.0V), the system displays a warning and enters deep sleep mode to prevent damage and preserve power. The device wakes periodically exactly in 1 minute time to check voltage and resumes operation when the battery is sufficiently charged.

Configuration parameters such as relay schedules, Wi-Fi credentials, and MQTT settings can be changed by modifying the source code or web interface (for schedule). Sensor readings are displayed on the OLED screen and can also be viewed via HTTP and MQTT communication.

03. Principle of Operation

The ESP32 serves as the central controller, interfacing with the DHT22 sensor for environmental data, the OLED display for user feedback, and the relay module for controlling connected loads. Upon boot-up, the ESP32 connects to the configured Wi-Fi network and synchronizes time using the NTP protocol. The system uses asynchronous event-driven communication to handle HTTP, MQTT, and UDP traffic.

Sensor readings are taken every 10 seconds and published to MQTT topics (home/sensor/temperature and home/sensor/humidity). The web interface is updated using JavaScript and displays the latest data in real time. Users can toggle relays via web buttons or by publishing to MQTT topics.

To minimize power consumption, the following strategies have been implemented:

- The system enters deep sleep mode when the battery voltage drops below 3.0V.
- Time-based wake-up is configured during deep sleep to periodically check voltage and resume operation if safe.
- OLED updates and sensor reads are rate-limited to reduce processing time and conserve power.

The entire device is mounted on a compact vero board with minimal active current consumption and features such as battery level indication and visual feedback on the OLED display. The buck booster allows for safe charging of the 18650 batteries without removal.

Following is the diagram of the whole system for the smart home lighting system.

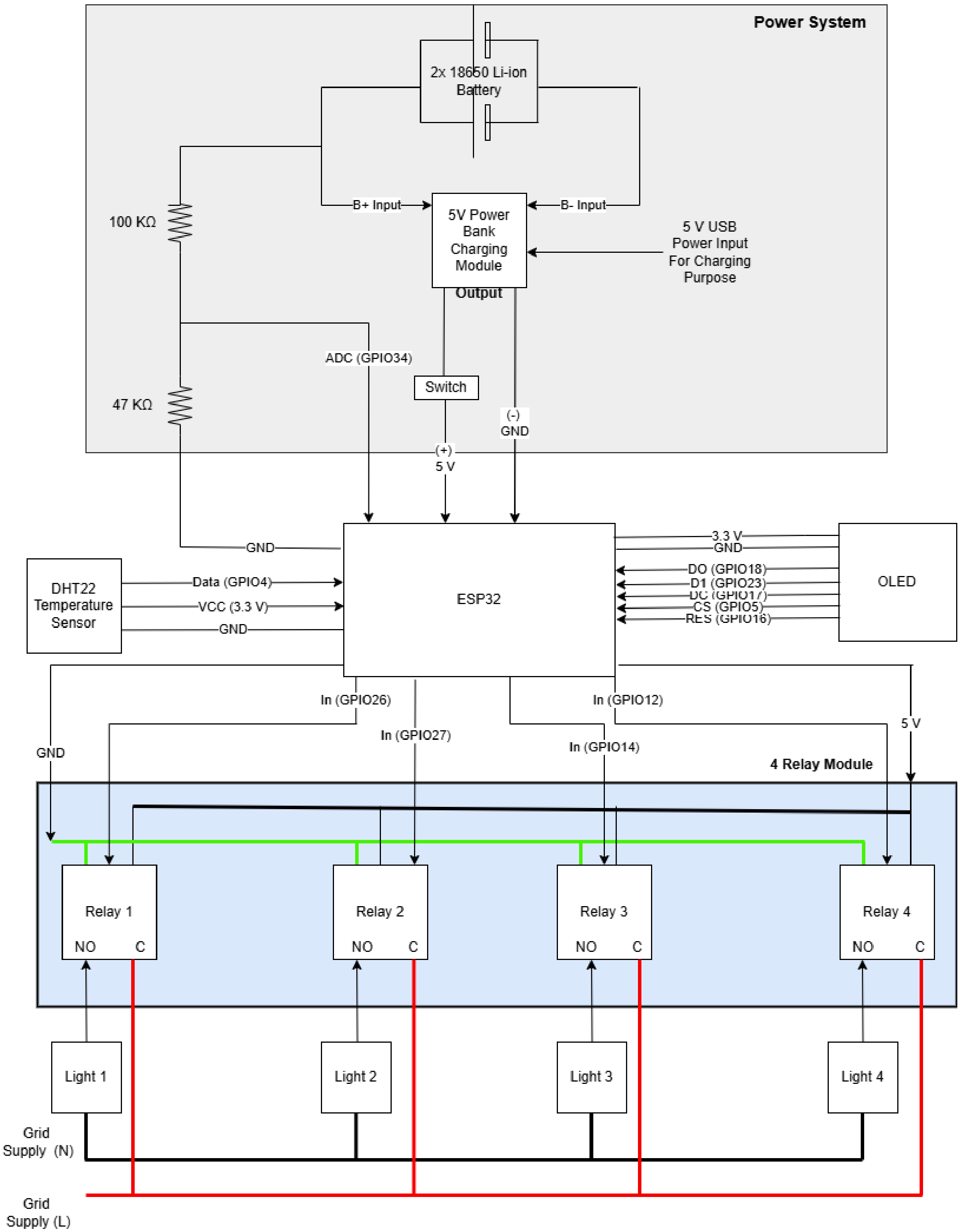

Figure 1: Block Diagram of the Smart Home Lighting System

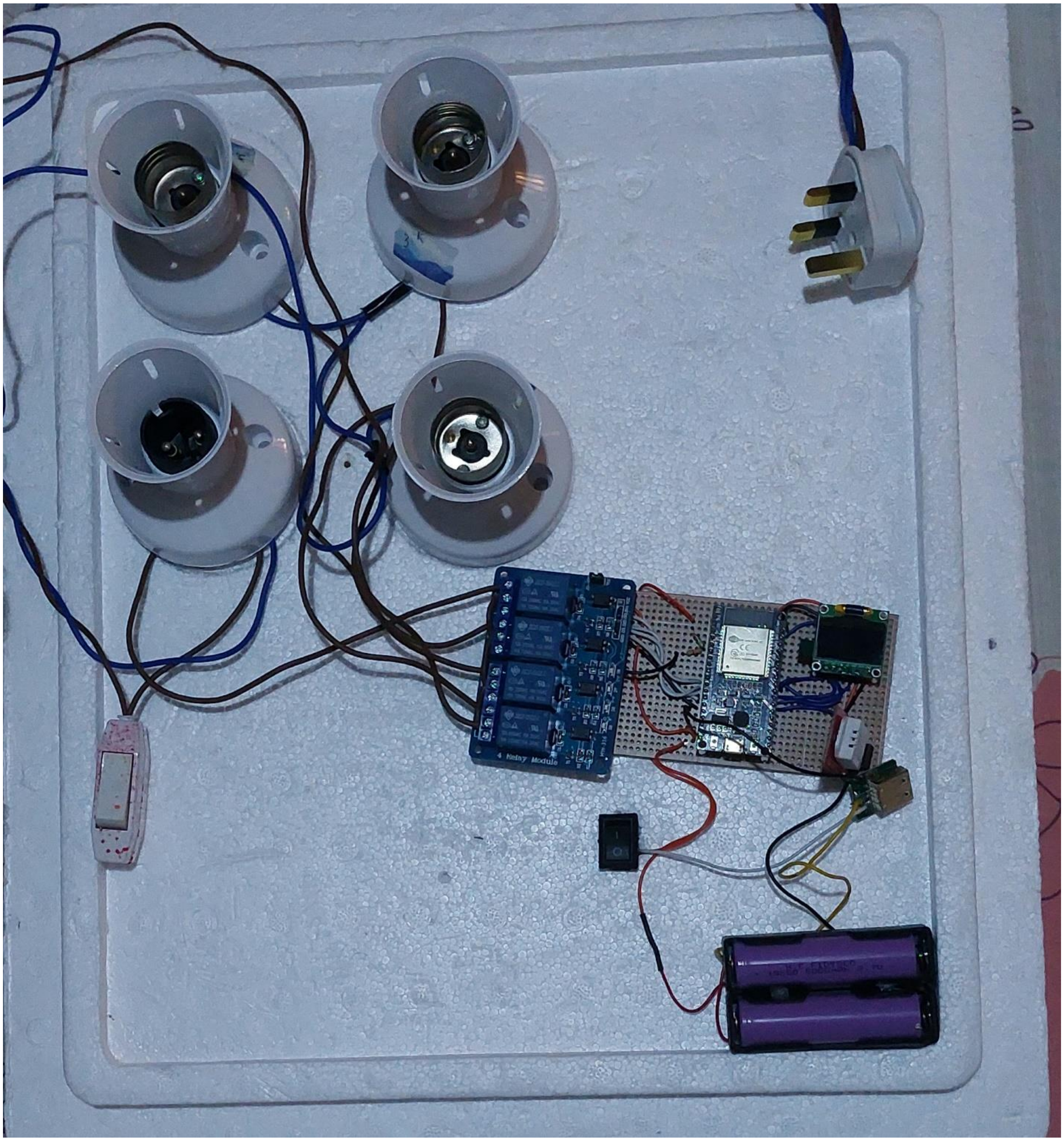

Figure 2: Overlook of the Smart Home Lighting System

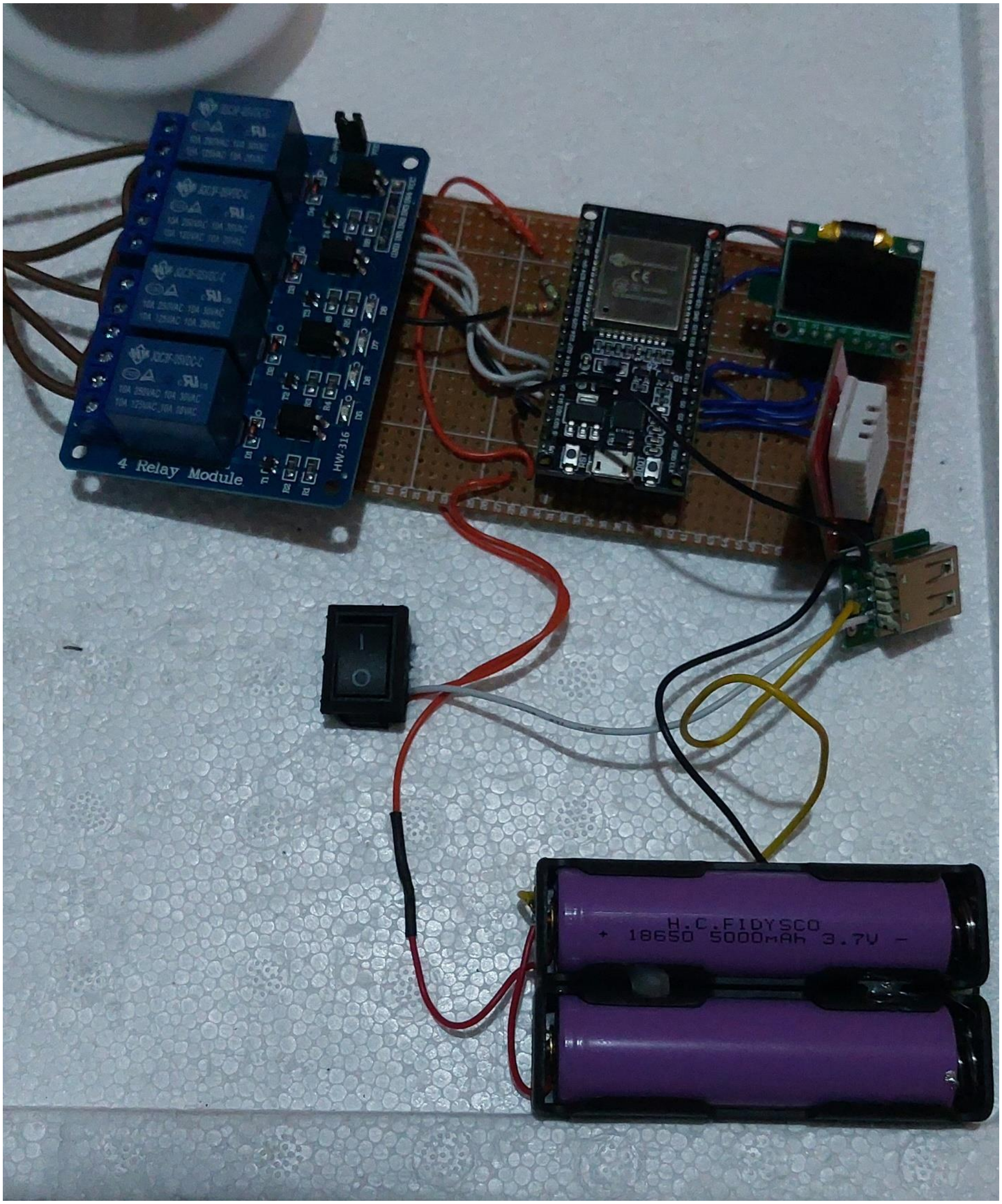

Figure 3: Closer look of the Operating System of the Home Lighting System