Home Lighting Controller with Wi-Fi Connectivity

Aseka Attanayake

2024/AE/032

Table of Contents

Table of Figures

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

02.1 Introduction

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 Wi-Fi 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.

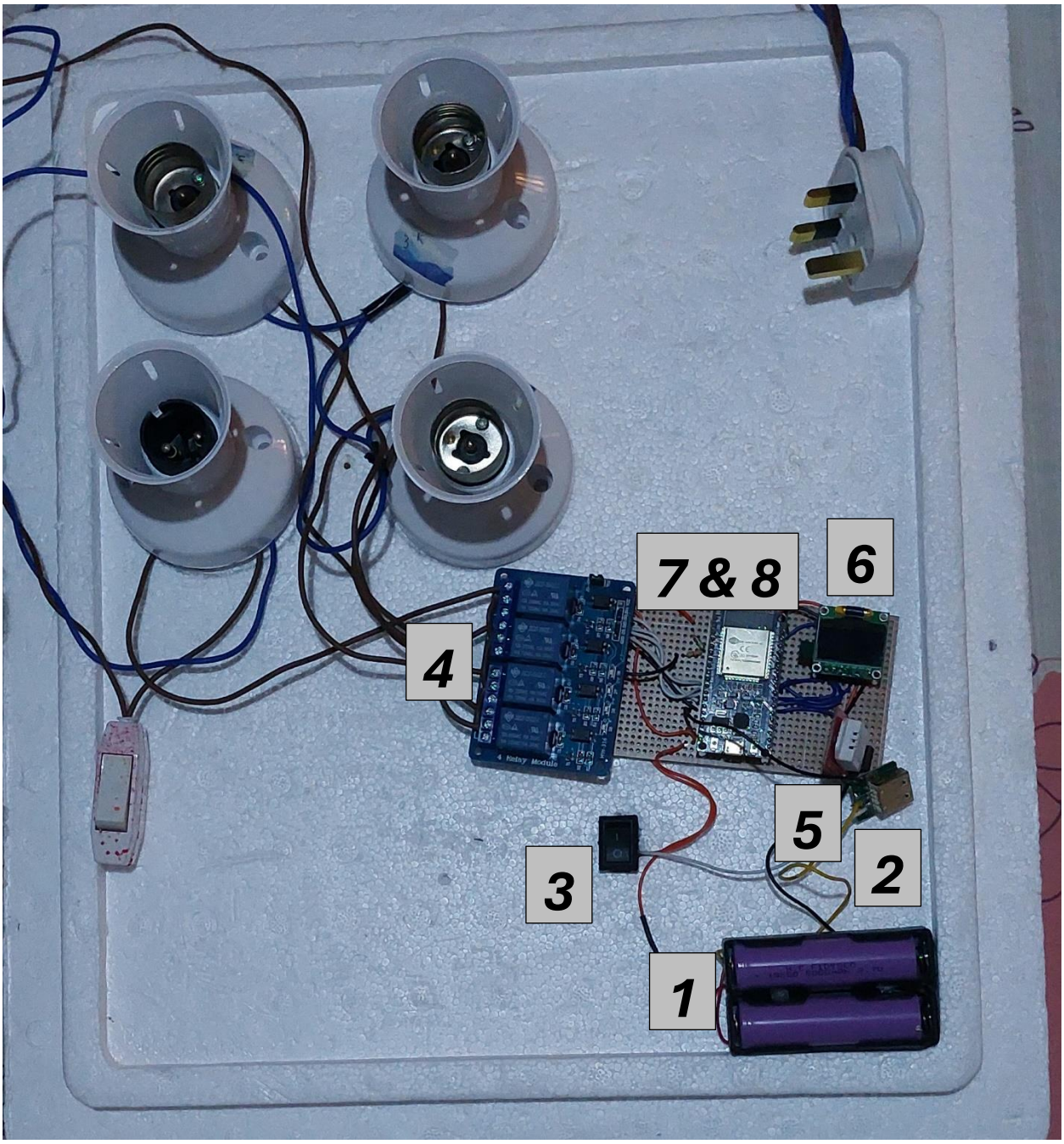

Figure 1: Overlook of the System

02.2 18650 Li-ion Batteries with 5V Power Bank Module and Switch

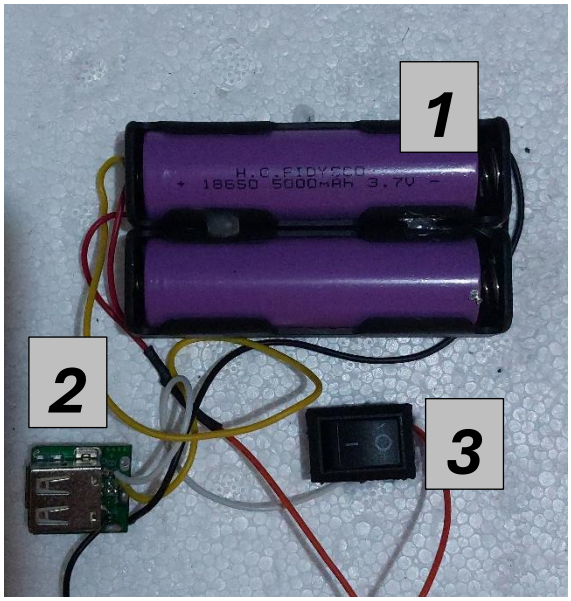

Figure 2: 18650 Li-ion Batteries with 5V Power Bank Module and Switch

The device is powered by two 18650 Li-ion batteries (1) connected in parallel, providing a stable 3.7V nominal voltage with increased capacity for longer operation. This battery setup is connected to a 5V power bank boost converter module (2), which steps up the voltage to provide a regulated 5V output suitable for powering the ESP32 and all peripheral components. An external switch (3) is installed between the battery and the boost converter, allowing the user to manually turn the device on or off without disconnecting the batteries. This setup ensures portability, rechargeability, and efficient power management for low-power IoT operation.

02.3 4 Relay Module

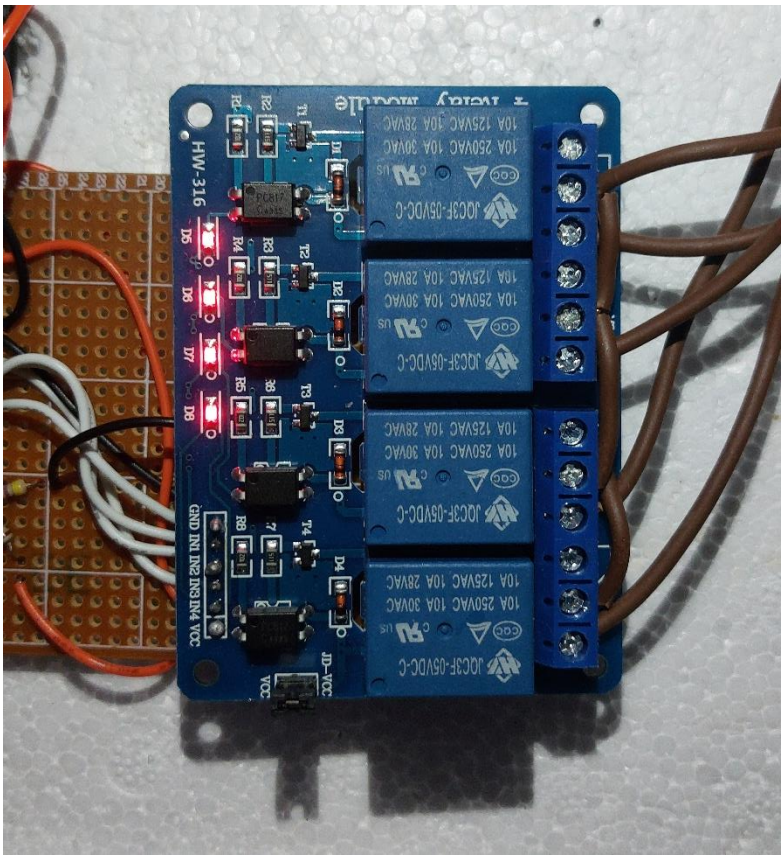

Figure 3: 4 Relay Module

The system features a 4-channel relay module that enables the ESP32 to control four individual lights: Living Room, Bedroom, Kitchen, and Garden. Each relay is controlled by a dedicated GPIO pin and can be operated remotely through HTTP web interface, MQTT messaging, or UDP commands. The Garden light relay supports user defined ON/OFF scheduling through the web interface. The relay module operates at 5V and includes built-in opto-isolators for safe switching and electrical isolation between the ESP32 and high voltage components. As for the future development on this part of this system, can be connected to the relay that user wants to control via Wi-Fi. Also even when the whole system is control through Wi-Fi, Only the garden light can be turned on using external switch which is connected to the grid supply directly. As for the future developments, can be connect all 4 lights to the grid supply for more accessibility.

02.4 DHT22 Temperature and Humidity Sensor

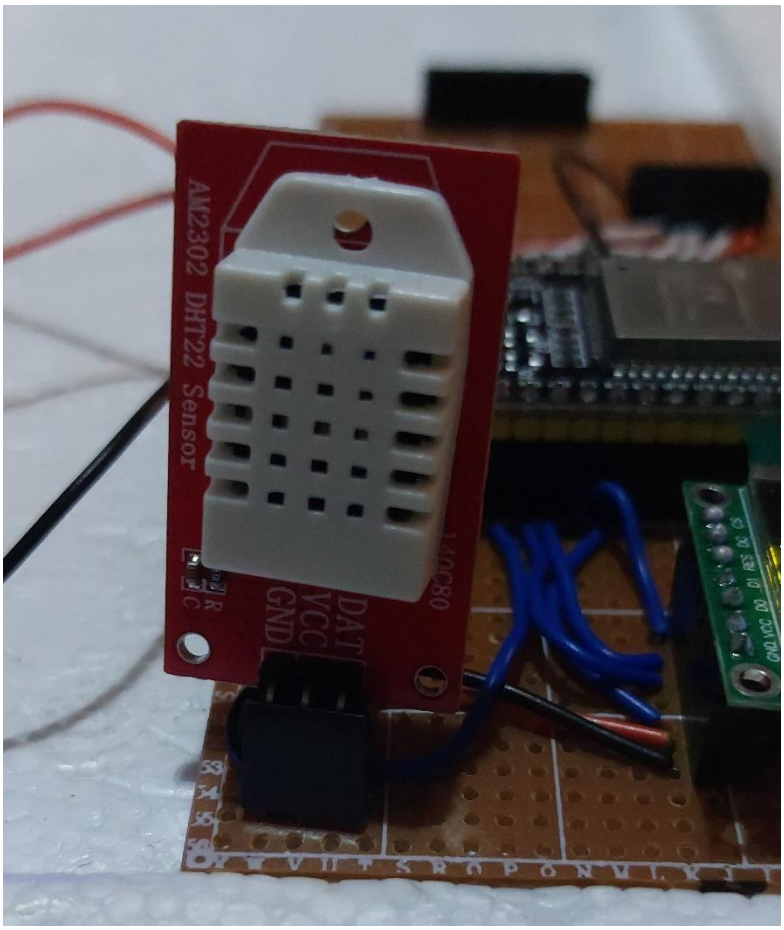

Figure 4: DHT22 Temperature and Humidity Sensor

A DHT22 digital sensor is used to continuously monitor the surrounding temperature and humidity levels. It is connected to GPIO 4 of the ESP32 and provides reliable readings at regular intervals. The ESP32 collects and processes this data, displaying it on the OLED screen, and also transmits it every 2 minutes to external servers using MQTT and HTTP protocols. This sensor plays a central role in long-term data collections on environmental condition monitoring.

02.5 OLED

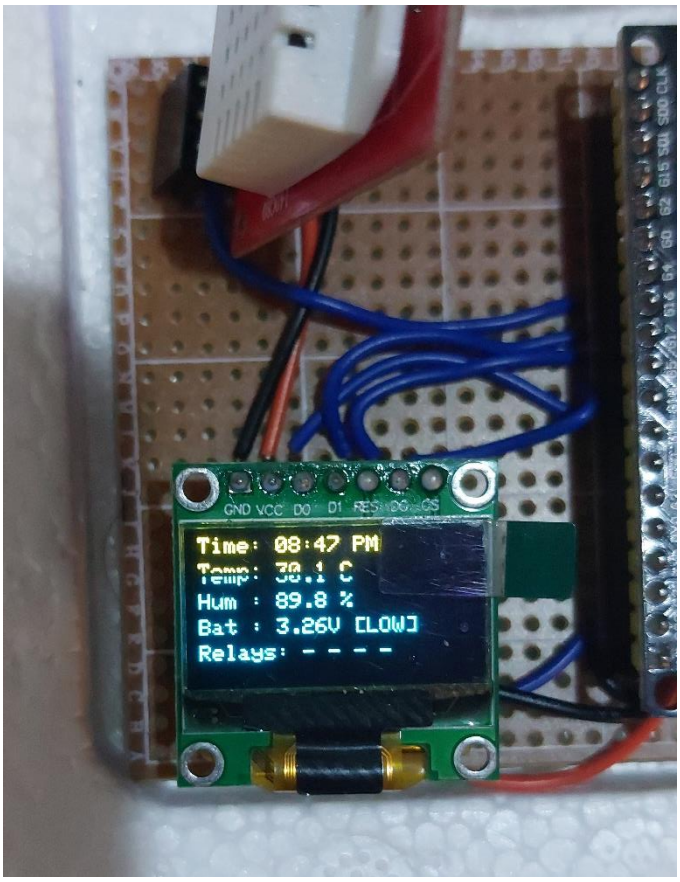

Figure 5: 0.96-inch 128x64 SPI OLED display with 7 pins

The device includes a 0.96-inch 128x64 SPI OLED display to provide a live visual interface. It shows key information such as the current time in 12-hour format with seconds, temperature in °C, humidity in %, battery voltage level, and relay status. The OLED also shows the status of each relay (ON/OFF) using initial letters “L, B, K, G” for the living room, bed room, kitchen and garden. This visual feedback allows users to monitor the system status in real-time without accessing the web interface, making it ideal for standalone and low-interaction scenarios.

02.6 ESP32

At the heart of the system is the ESP32 (7) microcontroller, which handles all major functionalities. It connects to Wi-Fi to sync time from an NTP server, hosts a local HTTP server for configuration and control, publishes sensor data and receives relay commands via MQTT, and supports UDP-based control as well. The ESP32 also monitors battery voltage through an ADC input and enters deep sleep mode when the voltage drops below a critical threshold 3.0V, conserving energy and preventing malfunction. It wakes periodically for data collection and communication, making the device efficient for continuous long-term operation.

02.7 Voltage Divider

To safely monitor battery voltage, a voltage divider circuit (8) is built using 100kΩ and 47kΩ resistors, connected to the ESP32's ADC input GPIO 34. This circuit scales down the actual battery voltage to within the ESP32's ADC range (0–3.3V), allowing accurate monitoring without damaging the microcontroller. The firmware reads the voltage regularly, and when it drops below 3.0V, the system displays a low battery warning and enters deep sleep mode after a short delay. This ensures both safe operation and longer battery life by avoiding deep discharge of the Li-ion batteries.

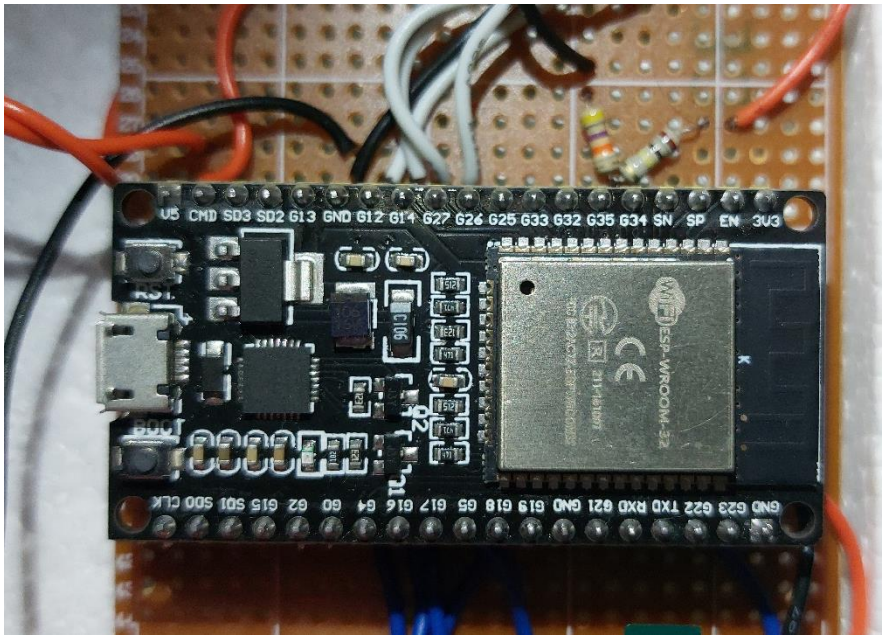

Figure 6: ESP32 with Voltage Divider Circuit

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 safe charging of the 18650 batteries without removal.

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

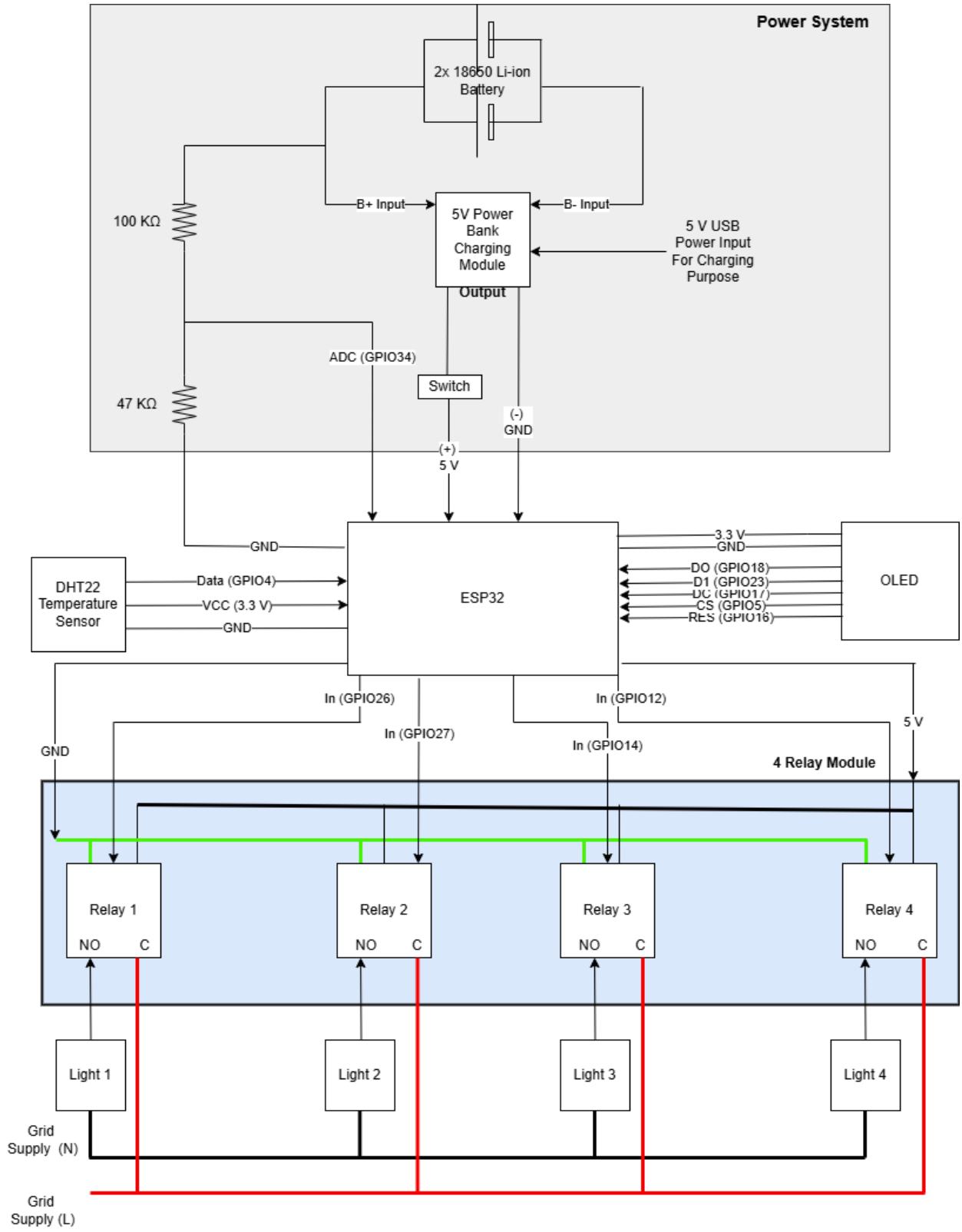

Figure 7: Block Diagram of the System

04. Example Data

To demonstrate the functionality of the system, temperature and humidity data were collected continuously over a 24-hour period using the DHT22 sensor. Data was transmitted at regular **2-minute intervals**, adhering to the project requirements for interval-based sampling. This data was sent to an external server using the HTTP protocol and stored for analysis. The graphs presented below illustrate the variations of environmental conditions throughout the day. The **"Time vs Temperature"** graph shows how ambient temperature fluctuates during daylight and nighttime hours, while the **"Time vs Humidity"** graph captures relative humidity changes in response to environmental factors. These results validate the system's capability to collect, transmit, and log sensor data reliably over long durations while operating on battery power.

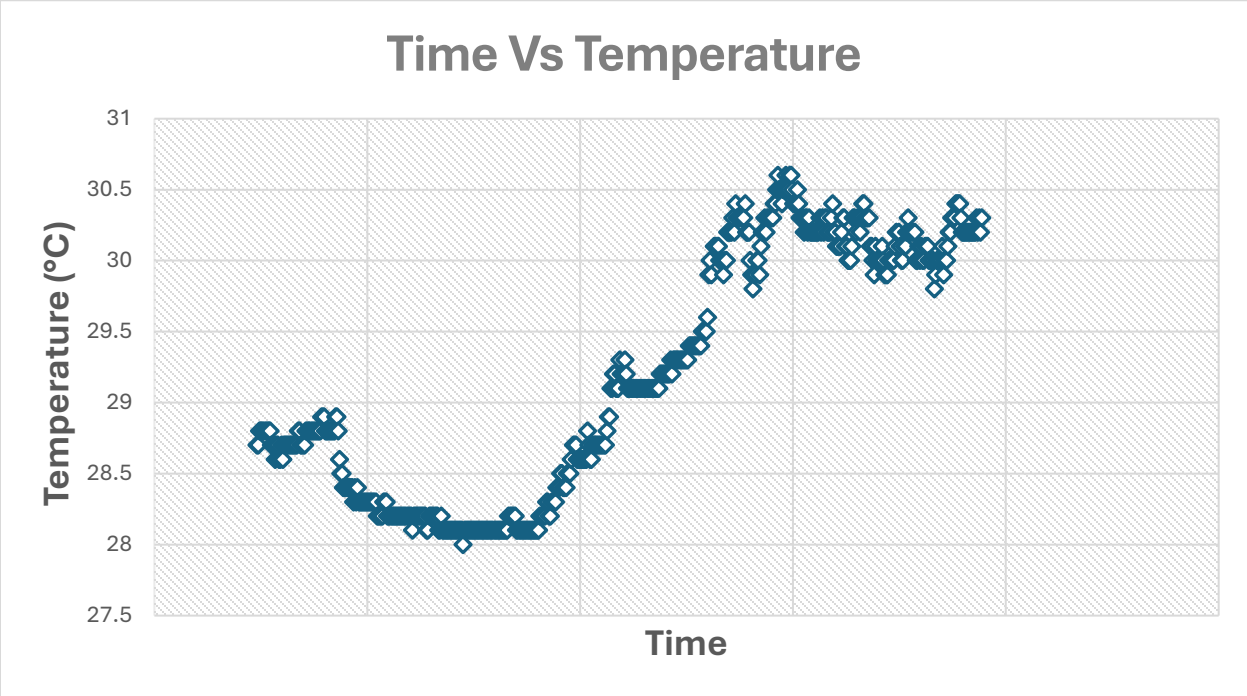

Figure 8: Graph of Time vs Temperature for 24hrs within 2 minutes interval

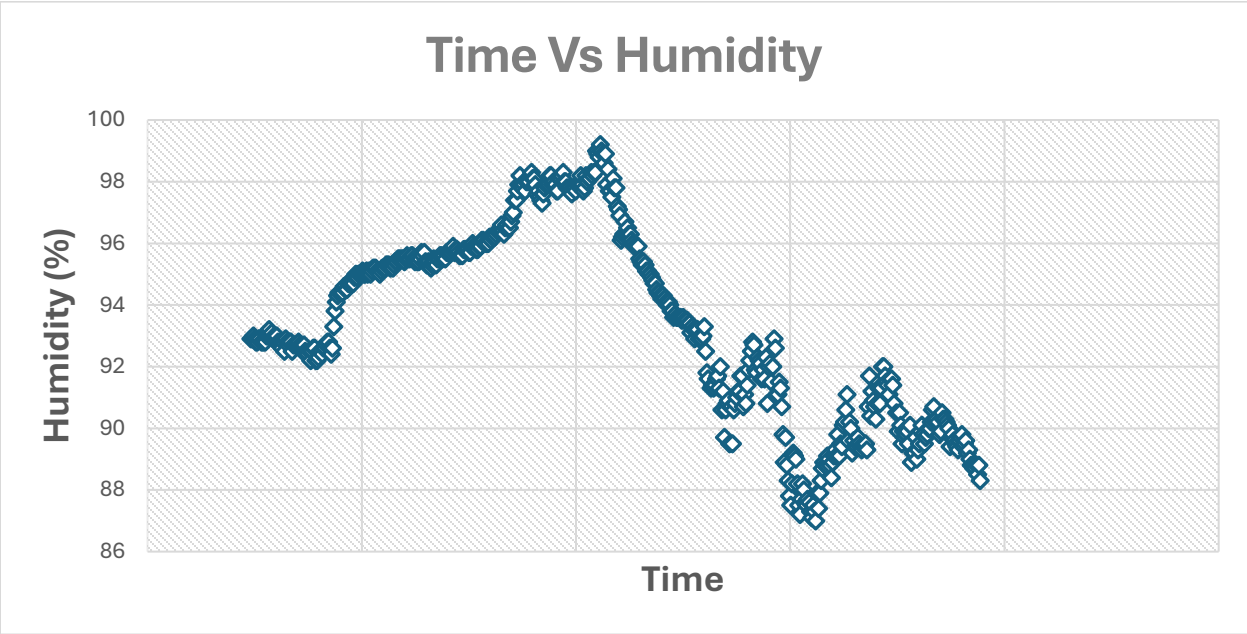

Figure 9: Graph of Time vs Humidity for 24hrs within 2 minutes interval

To access Full code, User Manual, Summery of Tools used, Pictures, Data and Graphs visit:
https://github.com/AsekaAttanayake/Home-Lighting-Controller-with-Wi-Fi-Connectivity

USER MANUAL

Home Lighting Controller with Wi-Fi Connectivity

Aseka Attanayake

2024/AE/032

Table of Contents

01. User Instructions

01.1 Introduction – Getting Started

- Power the device using the built-in **18650 Li-ion battery system** with the **5V boost converter** and switch.
- If charging is needed, connect the USB input on the power bank module to a USB charger.
- On startup:
    - The **ESP32** automatically connects to the pre-configured Wi-Fi network.
    - It synchronizes the time using an **NTP server**.
    - It starts reading data from the **DHT22 temperature and humidity sensor**.
- Users can:
    - Monitor temperature, humidity, battery status, and relay state via the **OLED display**.
    - Control relays through:
        - The **HTTP interface** (browser-based control page using ESP32's IP address).
        - **MQTT protocol** (publish/subscribe to specific relay and sensor topics).
        - **UDP messages** (using PowerShell or custom apps).
    - Schedule **Garden relay** ON/OFF times via the HTTP interface.
- If the battery voltage drops **below 3.0V**, the system:
    - Displays a **low battery warning** on the OLED.
    - Enters **deep sleep mode** to conserve power.
    - Wakes up every **1 minute** to recheck the battery level.
- **Configuration parameters** like Wi-Fi credentials and MQTT topics are set in the **source code**.
    - **Relay schedules** can be adjusted via the **web interface**.

- Sensor data is also transmitted every 2 minutes to a server via **HTTP** and **MQTT**.

01.2 18650 Li-ion Batteries with 5V Power Bank Module and Switch

- **(1)** Two 18650 batteries are connected **in parallel**, providing stable 3.7V with doubled capacity for extended runtime.
- **(2)** A **5V power bank module** boosts this voltage to 5V to power the ESP32 and peripherals.
- **(3)** An **external switch** allows the user to turn the system ON/OFF without disconnecting the batteries.
- This setup allows full **mobility**, **rechargeability**, and **manual control** over power delivery to the ESP32.

01.3 4 Relay Module

- The 4-channel **relay board** controls:
  - **Living Room**
  - **Bedroom**
  - **Kitchen**
  - **Garden**
- Each relay is assigned a dedicated **GPIO pin** from the ESP32.
- Relays can be switched via:
  - HTTP (/relay?room=...&state=on/off)
  - MQTT (home/relay/room)
  - UDP (LIVING ON, BED OFF, etc.)
- Garden relay has a user-configurable **ON/OFF timer** set via the web interface.

- The module features **opto-isolators** for protection and supports **future connection to grid-powered lights**.
- The system can support **manual switching** of the Garden light independently if needed.

01.4 DHT22 Temperature and Humidity Sensor

- Measures **temperature and humidity** of the environment.
- Connected to **GPIO 4** of ESP32.
- Data is:
    - Shown on the **OLED display**.
    - Transmitted every **2 minutes** via **MQTT** and **HTTP**.
- Plays a crucial role in environmental monitoring for long-term data collection and power analysis.

01.5 OLED Display

- A **0.96" 128x64 SPI OLED screen** is connected via:
    - MOSI (GPIO 23), CLK (GPIO 18), DC (GPIO 17), CS (GPIO 5), RESET (GPIO 16)
- Displays:
    - **Time** (12-hour format with seconds)
    - **Temperature and humidity**
    - **Battery voltage** and **battery level icon**
    - **Relay states** (using: L, B, K, G)
- Works without needing an app or browser, offering **real-time offline status** monitoring.

01.6 ESP32 Microcontroller

- Acts as the **central controller** for:
    - Sensor data collection
    - Relay operation
    - Communication protocols (Wi-Fi, HTTP, MQTT, UDP)
- Connects to Wi-Fi and syncs time via **NTP**.
- Monitors battery level and enters **deep sleep mode** if voltage < 3.0V.
- Wakes up automatically every **60 seconds** to resume operation.
- Provides a web server and real-time control with minimal power usage.

01.7 Voltage Divider

- Uses **100kΩ and 47kΩ resistors** to scale down battery voltage to 0–3.3V range.
- Voltage is read through **GPIO 34** (ADC pin) of the ESP32.
- Battery level is calculated and displayed.
- If the battery drops below **3.0V**:
    - A warning is shown on the OLED.
    - Device enters **deep sleep mode** after 3 seconds.
- This ensures protection of Li-ion cells from deep discharge.

02. Summary of All Tools

02.1 Core Electronics

Component	Details / Notes
ESP32 Dev Board	Main microcontroller with 38 pins, Wi-Fi and dual-core support
DHT22 Sensor	Temperature and Humidity Sensor
0.96" OLED Display	128x64 SPI OLED (7-pin, SSD1306 controller)
4-Channel Relay Module	JQC3F-05VDC-C relays, opto-isolated

02.2 Power Supply

Component	Details / Notes
2x 18650 Li-ion Batteries	3.7V, 5000mAh each, used in parallel (3.7V total)
5V Power Bank Module	Step-up boost converter to provide 5V from 3.7V battery
External Toggle/Push Switch	Used to control power to ESP32 (ON/OFF functionality)
Voltage Divider Resistors	100kΩ and 47kΩ for battery voltage monitoring that connects to ESP32 GPIO 34 Pin

02.3 Communication Interfaces

Component / Protocol	Purpose
Wi-Fi	Internet connection for MQTT, HTTP, UDP
MQTT (via PubSubClient)	Sensor and relay data transmission
HTTP Client (via HTTPClient.h)	Data upload to phys.cmb.ac.lk and to pre-created web inteface
UDP (via WiFiUDP.h)	Relay control via PC command line

02.4 Relay Outputs

Relay Channel	Usage
Relay 1	Living Room Light
Relay 2	Bedroom Light
Relay 3	Kitchen Light
Relay 4	Garden Light (With scheduled control)

02.5 Software Libraries Used

Library	Purpose
WiFi.h, WebServer.h	WiFi & HTTP server control
PubSubClient.h	MQTT support
NTPClient.h	Time sync with NTP
WiFiUDP.h	UDP communication
HTTPClient.h	HTTP POST to remote server
DHT.h	DHT22 sensor reading
Adafruit_GFX.h	OLED graphics library
Adafruit_SSD1306.h	OLED driver
ArduinoJson.h	For live JSON updates in web UI
SPI.h	For OLED SPI communication
esp_sleep.h	For deep sleep power saving