**USER MANUAL**

***Home Lighting Controller with Wi-Fi Connectivity***

***Aseka Attanayake***

***2024/AE/032***

Table of Contents

[01. User Instructions 3](#_Toc199621402)

[01.1 Introduction – Getting Started 3](#_Toc199621403)

[01.2 18650 Li-ion Batteries with 5V Power Bank Module and Switch 4](#_Toc199621404)

[01.3 4 Relay Module 4](#_Toc199621405)

[01.4 DHT22 Temperature and Humidity Sensor 5](#_Toc199621406)

[01.5 OLED Display 5](#_Toc199621407)

[01.6 ESP32 Microcontroller 6](#_Toc199621408)

[01.7 Voltage Divider 6](#_Toc199621409)

[02. Summary of All Tools 7](#_Toc199621410)

[02.1 Core Electronics 7](#_Toc199621411)

[02.2 Power Supply 7](#_Toc199621412)

[02.3 Communication Interfaces 7](#_Toc199621413)

[02.4 Relay Outputs 8](#_Toc199621414)

[02.5 Software Libraries Used 8](#_Toc199621415)

# 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 |