

# Student Information

**Name:** Berk KOCABOREK

**Student ID:** 20230808607

## Project Title

**Ambient Light System Using Addressable LEDs with ESP32**

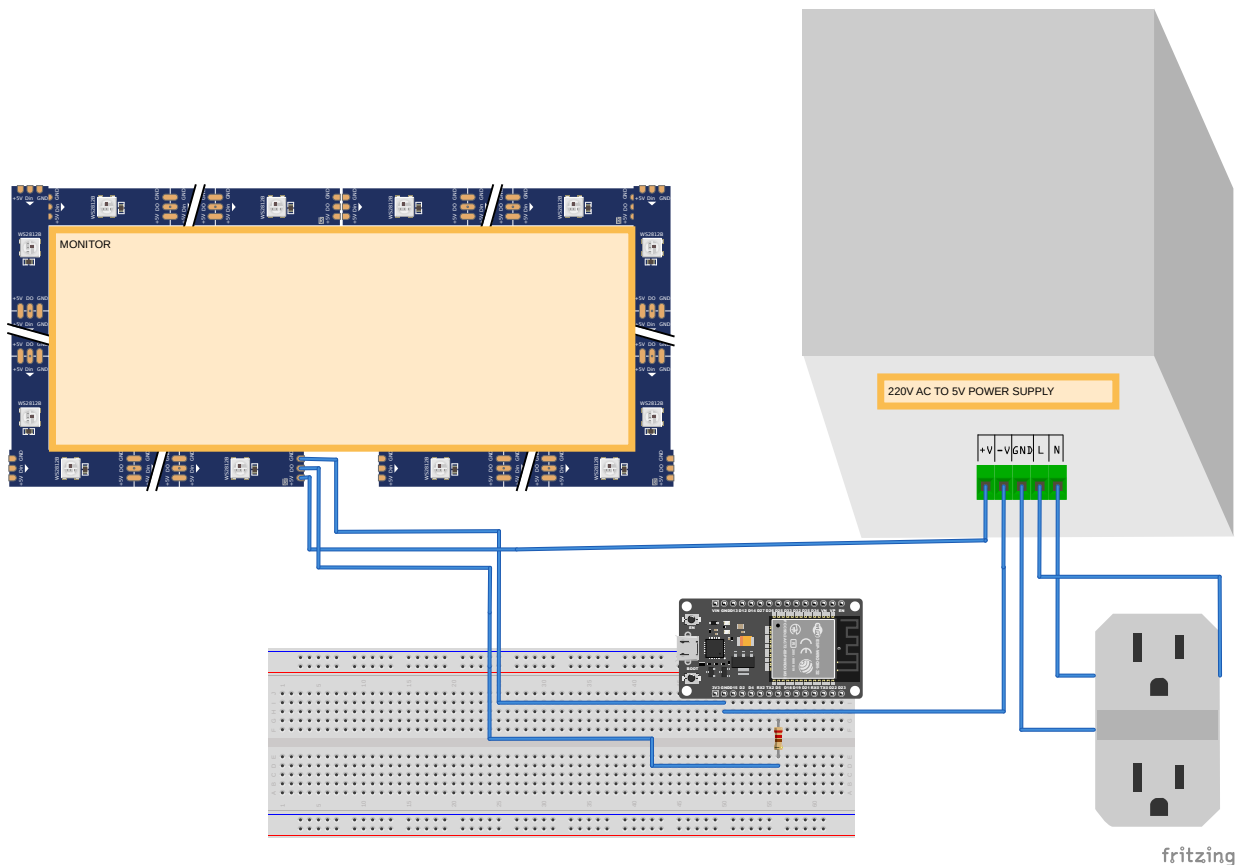
## Project Description

This project aimed to develop an ambient light system for computers using WS2812B addressable LEDs controlled by the ESP32 Development Kit. The system dynamically adjusted the ambient lighting based on screen content to provide an immersive and comfortable user experience.

The main objectives were to integrate WS2812B LEDs with the ESP32, establish data communication via UDP over Wi-Fi between the computer and the ESP32, and process screen data to control LED lighting patterns in real-time. The system included secure communication mechanisms to ensure data integrity and privacy.

Additional capabilities were also integrated, such as remote control via MQTT through Adafruit IO, and local control via the WLED interface.

## System Architecture Diagram



# Components Used

## Hardware

- ESP32 Development Kit
- WS2812B Addressable LEDs
- Power supply (5V) for LEDs
- Resistors
- Optional: Breadboard

## Software

- Arduino IDE or ESP-IDF for ESP32 programming
- WLED
- Prismatic
- Ambilight software
- Adafruit IO

## Physical Setup

1. Connected ESP32 pin 5 to the LED data input pin through a resistor.
2. Connected ESP32 GND to both the power supply -V and LED GND.
3. Connected the power supply's +5V output to the LED's power input.
4. Soldered the power supply's L, N, and GND terminals to the corresponding wires of a wall plug.
5. Plugged in the power supply and provided power to the ESP32 via micro-USB (or another method).

*Physical assembly completed.*

## Software Configuration

1. Installed WLED firmware on the ESP32 and connected it to the Wi-Fi network.
2. Installed Ambilight software on the computer.
3. Configured synchronization over the local network using UDP.
  - Adjusted Ambilight settings based on LED layout and count.
4. Set up synchronization between WLED and Adafruit IO (via WLED interface).
5. Created a control dashboard on Adafruit IO to execute custom LED commands.
  - Documentation: [Adafruit IO MQTT Guide](#)

## Communication Protocols

The system used **UDP over Wi-Fi** to synchronize LED behavior with the computer's screen content in real-time. This ensured low-latency, efficient control within the local network. Additionally, **MQTT** protocol was used for cloud-based remote control via Adafruit IO.

## Usage Instructions

- To activate the Ambilight system, launch the Prismatic application and allow control via WLED (Prismatic settings can be adjusted as needed).
- To control the LEDs remotely, visit the Adafruit IO dashboard (no need to be on the local network).
- Alternatively, the LEDs can be locally controlled via any browser by navigating to the ESP32's WLED local IP address. (Ambilight also communicates via this local WLED instance.)

## Conclusion

This project successfully combined hardware and software components to deliver a responsive and secure ambient lighting system. With both local and remote control options, it provided flexibility, automation, and an enhanced visual experience for computer users.

## Documentation Links

- [WLED Documentation](#)
- [Ambilight / Prismatic GitHub Repository](#)
- [Adafruit IO MQTT Guide](#)

## GitHub Repository Link

<https://github.com/littleborek/IoT>

---