Term Project Report – Ambient Light System with ESP32

Student Information

- **Full Name:** Berk Kocabörek
- **Student ID:** 20230808607

Project Title

Ambient Light System Using Addressable LEDs with ESP32

Introduction

The objective of this project was to create an ambient lighting system for a computer setup using WS2812B addressable LEDs controlled by an ESP32 Development Kit. The motivation for this project originated from the desire to enhance user experience while watching media or playing games through immersive lighting that dynamically reflects the screen content. Such systems are commercially available but often expensive, so this DIY solution provides an affordable and customizable alternative.

The problem addressed was the lack of synchronized ambient lighting in many setups and the opportunity to leverage IoT tools to automate and control lighting behavior dynamically. The relevance of this system lies in its capability to improve user comfort and visual experience.

System Architecture

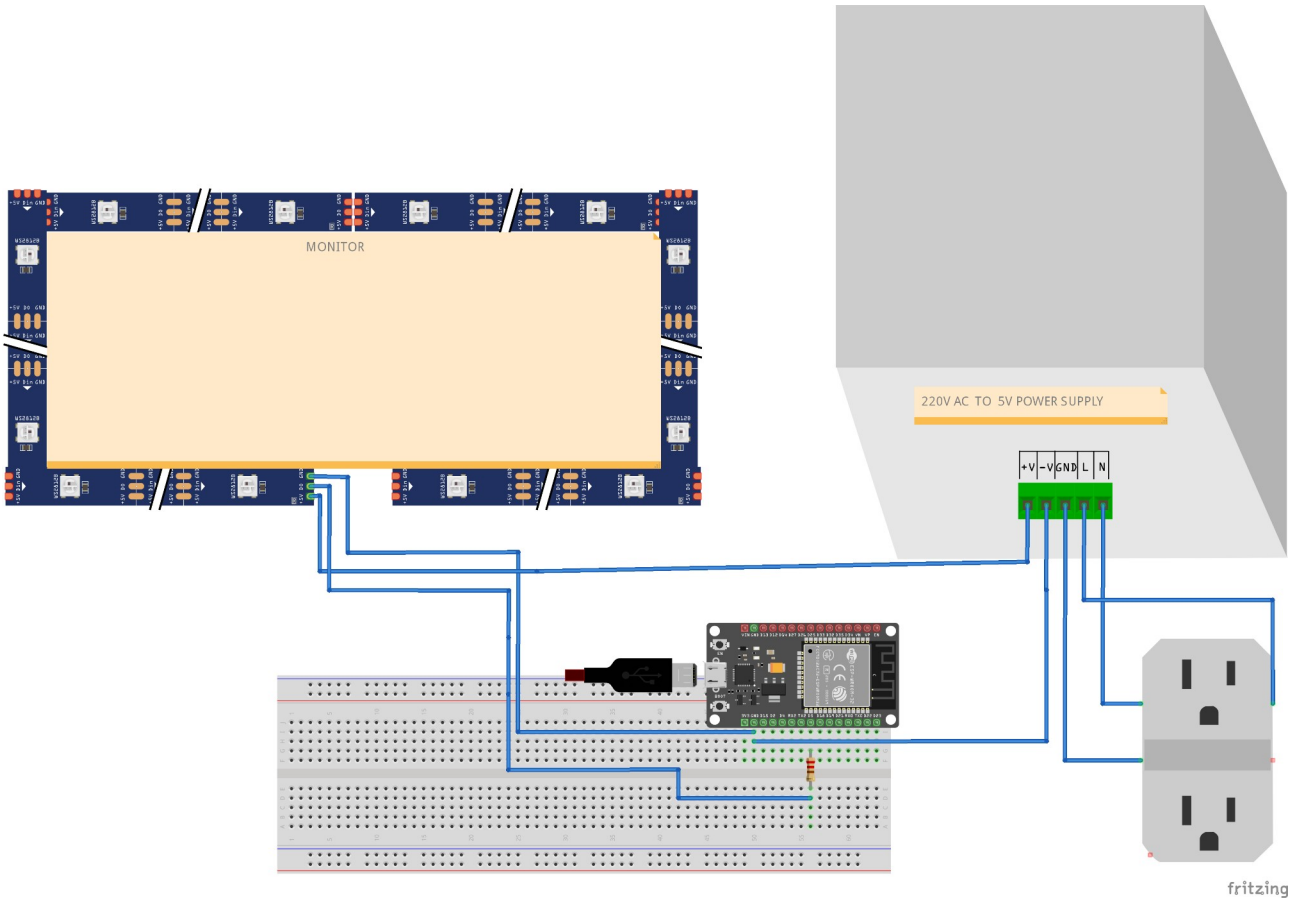
Architecture Overview

The system operates using the following components:

- **ESP32 Development Kit**
- **WS2812B Addressable LEDs**
- **Power Supply (5V)**
- **1x Resistor (330 Ohm)**
- **Breadboard (optional)**
- **WLED Firmware**
- **Ambilight Software (Prismatik)**
- **Adafruit IO Dashboard (for remote control via MQTT)**

The communication between ESP32 and the computer is handled via Wi-Fi using UDP protocol for Ambilight functionality and MQTT for cloud-based LED control.

System Diagram



Implementation Details

Hardware Setup

1. Connected ESP32 pin D5 to the LED data input through a 330-ohm resistor.
2. GND pin of the ESP32 was connected to the GND of both the LED and power supply.
3. Power supply +5V connected directly to LED's power input.
4. Power supply input terminals (+, -, GND) were soldered to AC plug wires.
5. Finally, the system was powered on via a wall plug and ESP32 was powered via micro-USB.

Firmware and Software Configuration

1. Installed **WLED** firmware on the ESP32.
2. Connected the ESP32 to Wi-Fi.
3. Installed **Prismatik** software on the computer.
4. Configured Ambilight over UDP to send screen data to WLED based on LED layout.
5. Set up MQTT settings in WLED to connect with **Adafruit IO**.
6. Created a control dashboard on Adafruit IO to remotely trigger lighting effects.

Communication Protocols

- **UDP (Wi-Fi)**: Used for real-time Ambilight data transmission.
- **MQTT**: Used for cloud control via Adafruit IO dashboard.

Cloud Service

- **Adafruit IO**: For remote dashboard-based control over MQTT.

Data Processing

- Screen content is captured and analyzed by Prismatik, converted into LED color signals, and sent to ESP32 via UDP.

Security Features

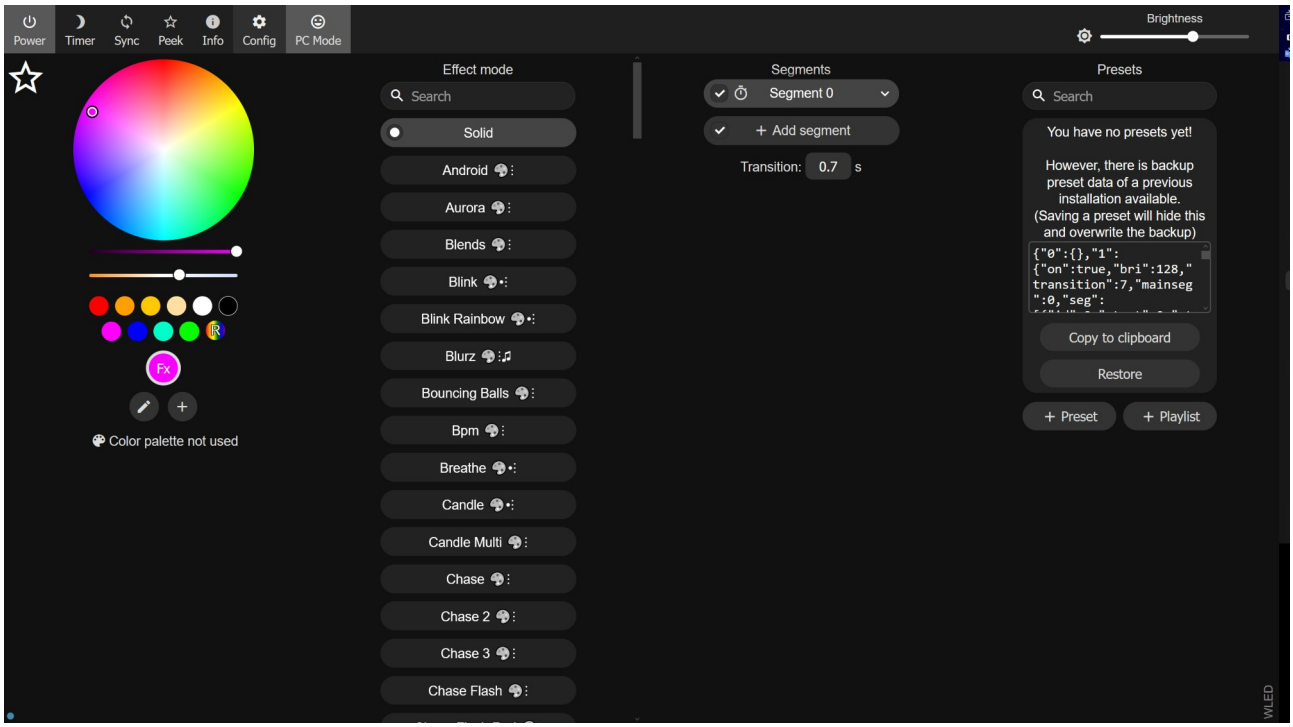
- The local network is password protected.
- Adafruit IO requires user authentication for dashboard access.

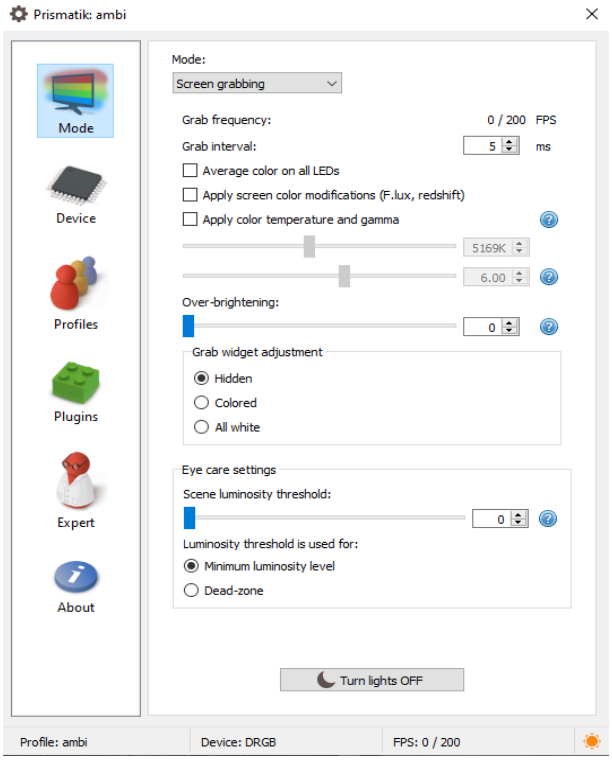
Results and Analysis

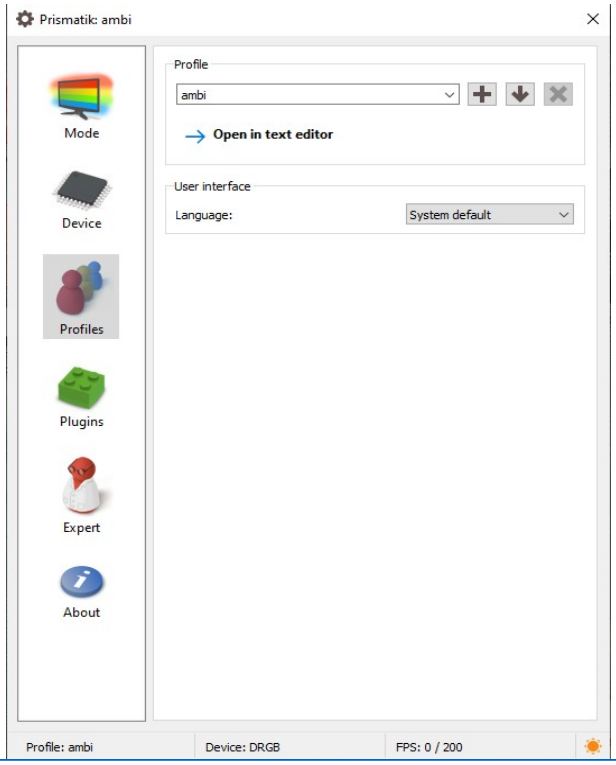
Outputs

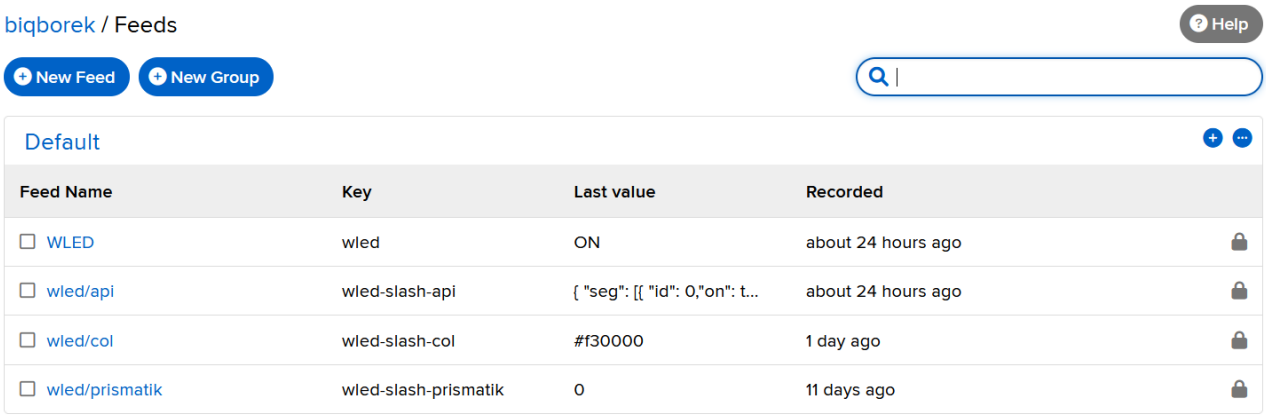
- LED strip dynamically reflects screen content via Prismatik.
- User can change colors/effects through Adafruit dashboard.
- LED control is also available through local WLED web interface.

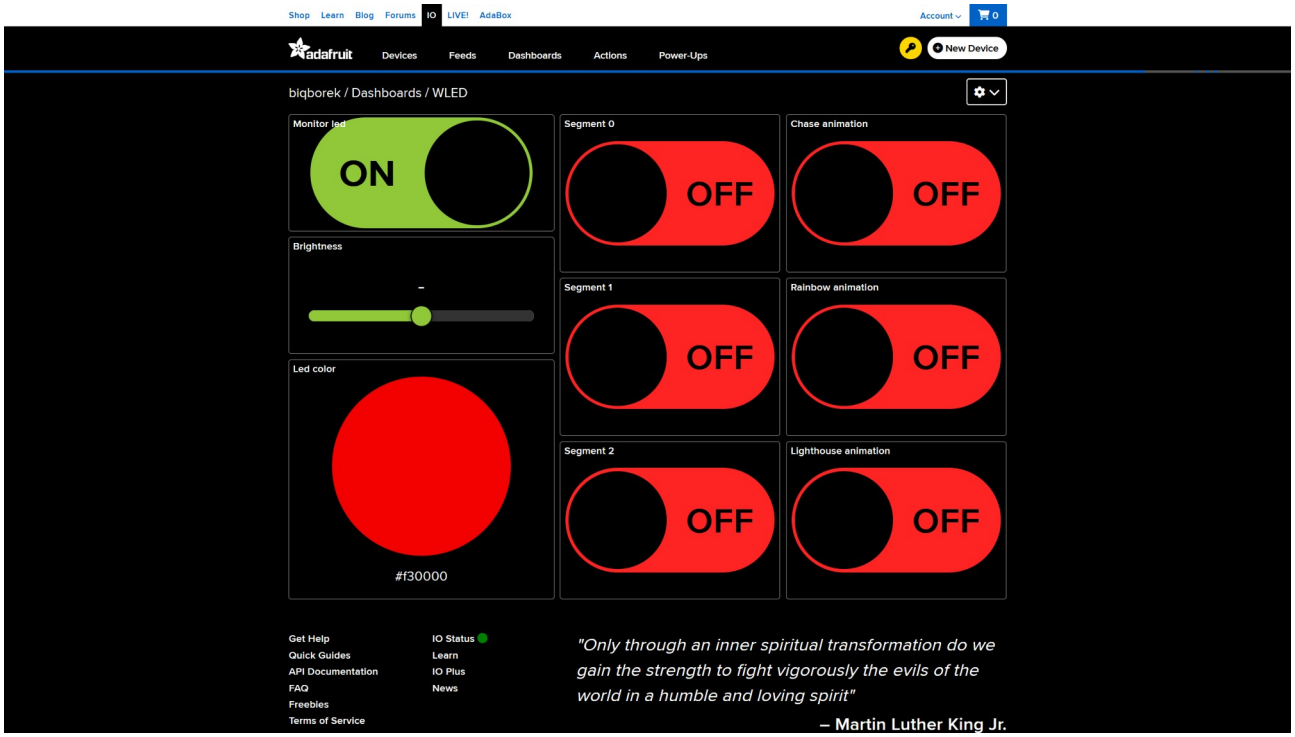
Visual Proof











Interpretation

- **What worked:** Prismatik integration, WLED firmware, MQTT setup.

- **What didn’t:** Initial power management required careful attention to prevent instability.
- **Why:** Stable 5V supply and good soldering were essential for WS2812B reliability.

Conclusion and Future Work

Summary

This project achieved the creation of a reliable and interactive ambient lighting system using ESP32, WLED, and IoT services. It demonstrated real-time screen content synchronization as well as remote LED control.

Limitations

- The system currently uses only one mode of ambient detection (screen mirroring).

Future Improvements

- Add light sensor support to adapt brightness based on room lighting.
- Integrate voice assistant support (e.g., Google Assistant).

References

- WLED Documentation
- Prismatik Software
- Adafruit IO MQTT Guide

GitHub Repository

https://github.com/littleborek/IoT