IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

ESP8266-Based Smart Room Decoration System Using Addressable LEDs

Prashant Kaushal[1], Prince Kumar[2], Koushik Pal[3], Anirban Ghosal[4], Anurima Majumdar[5], Antara Ghoshal[6]
Department of Electronics & Communication Engineering,
Gurunanak Institute of Technology, Khardaha, West Bengal[1,2,3,5,6]
JIS College of Engineering, Kalyani, West Bengal[4]
prashantkaushal996@gmail.com

Abstract: *With the rise of smart homes, ambient lighting has become a vital element in enhancing the atmosphere and user interaction experience. This paper introduces a firmware-based IoT solution for smart room decoration using addressable RGB LEDs controlled by an ESP8266 microcontroller. The system offers features such as customizable color themes, support for multiple LED types (WS2812, SK6812, etc.), a Wi-Fi-based web application for interactive control, and robust API support for automation. The user-friendly interface hosted on a local web server allows seamless switching between access point (hotspot) mode and home Wi-Fi. The primary aim of this project is to create an affordable, customizable, and extensible LED control system that integrates modern IoT practices with aesthetic lighting.*

Keywords: ESP8266 microcontroller

I. INTRODUCTION

In the modern era of smart living and automation, the integration of the Internet of Things (IoT) into everyday spaces has transformed how we interact with our environment. Lighting, one of the most fundamental aspects of interior design, is no longer static. Through the use of addressable RGB LEDs and microcontroller-based firmware, lighting systems have evolved into dynamic, programmable, and interactive decor solutions.

This project aims to create an IoT-based ambient lighting system using the ESP8266 microcontroller, which enables wireless control and customization of addressable LEDs (e.g., WS2812B). The firmware is designed to support real-time user interaction through a responsive web interface, hosted locally on the ESP8266 module and accessible through the local network via mDNS (ambi.local) or a static IP address. It includes support for AP (Access Point) mode for initial configuration and STA (Station) mode for regular use, ensuring ease of setup and deployment.

Unlike traditional lighting systems that rely on static switches or physical remote controls, this project provides wireless control through any device with a web browser, offering a modern and convenient approach to smart room decoration. Users can switch color themes, adjust brightness, apply lighting effects, and even integrate the system with third-party automation tools via API calls.

The motivation for this work comes from the growing demand for open-source, DIY smart lighting solutions that are both functional and aesthetic. By combining embedded systems, networking, and interactive design, this project offers a holistic learning experience and serves as a cost-effective alternative to commercial smart lighting kits.

Theory

To understand the functionality and importance of the ESP8266-based IoT lighting system, various theoretical concepts from embedded systems, wireless communication, and LED control technologies are involved. This section outlines the foundational technologies and how they work together to make the system operational and user-friendly.

DOI: 10.48175/IJARSCT-28190

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

Sr No.	Component Name	Function/use
1	ESP8266 NodeMCU Board	Core microcontroller unit that handles Wi-Fi communication, LED control logic, and web server hosting.
2	WS2812B / SK6812 Addressable LEDs	RGB LEDs that receive serial data signals for individual color control and visual effects.
3	Power Supply (5V, 2A or higher)	Provides sufficient current for the LED strip and stable operation of the ESP8266.
4	USB to Micro USB Cable	Used for flashing firmware to the ESP8266 board via Arduino IDE.
5	LittleFS File System (Software)	Filesystem used to store and serve web application assets (HTML, CSS, JS) from the ESP8266 flash memory.
6	Resistor (330Ω - optional)	Limits current on the LED data line to protect the first LED from voltage spikes.
7	Breadboard and Jumper Wires	Stabilizes voltage supply to prevent sudden surges during LED power-up.

II. EMBEDDED SYSTEMS

An embedded system is a hardware-software computing unit designed for a specific function. In this project, the ESP8266 functions as an embedded platform that executes firmware logic for wireless connectivity, web server hosting, LED signal generation, and API handling. It combines the efficiency of real-time operation with the flexibility of over-the-air control.

Key features of embedded systems used in this project include:

- Low power consumption
- Real-time responsiveness
- Integrated Wi-Fi capabilities
- Flash memory for storing web resources (using LittleFS)

Addressable LEDs (e.g., WS2812B)

Addressable LEDs are intelligent RGB lighting modules where each LED is individually controllable using a single data line. These LEDs operate using a protocol where each LED passes the control signal downstream after processing its own instructions. This allows for pixel-level animations, gradients, and dynamic effects.

Key technical aspects:

- Operate on 5V DC
- Require accurate timing (handled using the FastLED library)
- Controlled using digital pins (typically D4 or GPIO2 on ESP8266)
- Offer 24-bit color (8 bits each for Red, Green, and Blue)

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

FastLED Library

The FastLED library is a powerful C++ library designed for driving various types of addressable LEDs with high precision. It abstracts the timing-sensitive protocol of different LED types and provides an easy API for controlling color patterns, brightness, and animations.

Features used in this project:

- CRGB objects for color manipulation
- Built-in animations and rainbow effects
- Frame rate control for smooth transitions
- Brightness dimming and gamma correction

ESP8266 and Web Server

The ESP8266 is a Wi-Fi-enabled microcontroller with GPIO pins and sufficient flash memory to act as a lightweight web server. The firmware uses:

- **ESPAsyncWebServer** for handling HTTP requests asynchronously
- **LittleFS** for serving HTML, CSS, and JS files to the client device
- **mDNS (Multicast DNS)** to allow access via ambi.local instead of relying on IP addresses

The device begins in AP mode if Wi-Fi is not configured, creating a temporary network called "Ambi" to accept configuration input. Once credentials are stored, it reboots into Station mode and connects to the user's home network.

Capacitive UI vs. Web UI

While some projects use capacitive or physical button interfaces, this project relies entirely on web-based input. A modern HTML UI is rendered directly from the ESP8266, allowing interaction from any browser without needing a mobile app or proprietary software. This interface communicates with the ESP8266 via HTTP requests and JavaScript, allowing real-time color and brightness control.

REST API Integration

For automation, the firmware also supports a set of RESTful APIs. These can be used with external systems like Node-RED, Home Assistant, or even voice assistants (via bridges), making it extensible for more advanced users.

III. WORKING PRINCIPLE

The working of the smart lighting system is based on the combination of embedded programming, real-time wireless communication, and addressable LED control. The ESP8266 microcontroller operates as the central controller, managing both the web-based interface and the LED output through a sequence of programmed events.

1. Initialization and Setup Mode

Upon first boot or reset (when no Wi-Fi credentials are stored), the ESP8266 initializes in **Access Point (AP)** mode. It broadcasts a hotspot SSID titled "Ambi", which users can connect to using any Wi-Fi enabled device such as a mobile phone or laptop. Once connected, the user can navigate to http://192.168.4.1 or http://ambi.local in a web browser. This opens a configuration interface served from onboard memory (LittleFS), allowing the user to input their home Wi-Fi SSID and password.

After credentials are submitted, they are saved to flash memory, and the system reboots into **Station (STA)** mode. The ESP8266 now connects to the user's home network automatically upon each startup.

2. Web Server and User Interaction

Once connected to the local Wi-Fi, the ESP8266 starts hosting an asynchronous HTTP web server using the ESPAsyncWebServer library. When the user visits http://ambi.local or the ESP's IP address (e.g., http://192.168.0.181), the stored HTML/CSS/JS interface is loaded from flash memory and served to the client's browser.

This web application allows the user to:

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

Select color themes (static and dynamic)
Adjust brightness
Trigger lighting animations
Turn the LED system ON or OFF
Save and apply custom themes
The interaction between the user interface and the ESP8266 is managed through JavaScript and AJAX calls, ensuring real-time updates without reloading the webpage.

3. LED Control and Effects

The ESP8266 uses the FastLED library to generate timing-accurate signals for controlling the addressable RGB LEDs. When a user changes color or mode via the web interface, the relevant command is received by the ESP8266, which then processes and translates it into a signal sent through the GPIO data pin (commonly D2/GPIO4 or D1/GPIO5).
The signal contains a 24-bit color code (8 bits each for Red, Green, Blue) and positional information, which is interpreted by each LED in the chain. This allows individual LEDs to light up in specific colors and patterns, enabling effects such as:
Static color modes
Rainbow cycle
Breathing/pulsing effect
Theater chase and wave animations
The FastLED library ensures the correct signal timing required by WS2812B/SK6812 LEDs, which typically need 800 Kbps signal input.

4. API Access and Automation (Optional)

In addition to manual control via the web interface, the firmware supports a set of REST APIs. These APIs allow integration with external IoT platforms such as:

- Home Assistant
- Node-RED
- IFTTT
- Google Assistant/Alexa (via bridges)

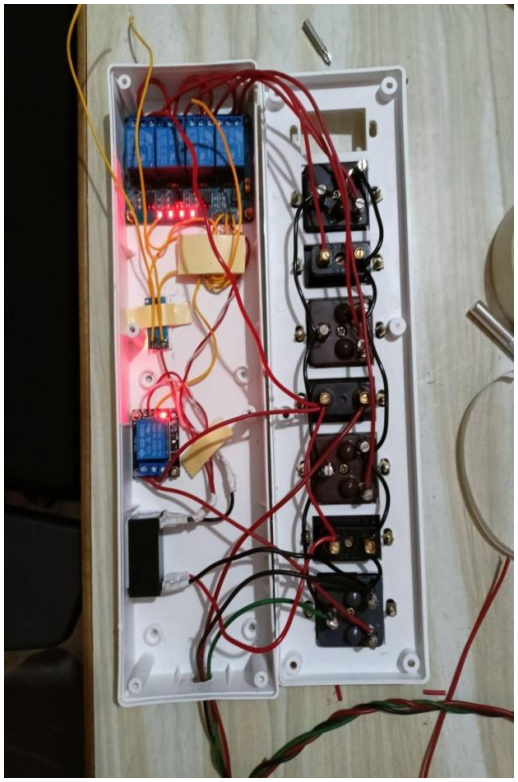
Example API functions:
/api/on – Turns LEDs on
/api/off – Turns LEDs off
/api/color?hex=#FF00FF – Sets a specific color
/api/theme?mode=rainbow – Triggers a specific animation
This makes the system scalable and programmable for advanced users and smart home enthusiasts.

5. Safety and Stability

To maintain system stability:
A **330Ω resistor** may be placed in the data line to protect the first LED from voltage spikes.
A **1000 µF capacitor** is recommended across the power supply to handle inrush current when many LEDs turn on simultaneously.
The **optional logic level shifter** ensures 3.3V to 5V signal conversion for long LED strips, improving performance and signal integrity.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

IV. WORKING MODEL



Model Description:

The proposed system is a miniature prototype of a smart IoT-based LED lighting controller built using an ESP8266 NodeMCU board and WS2812B addressable LEDs. The setup demonstrates a fully functional, customizable LED control environment that is wirelessly operable via a web-based application hosted on the ESP8266 microcontroller itself.

In its physical form, the model consists of:

An **ESP8266 NodeMCU** board acting as the central controller and Wi-Fi module.

A short strip of **WS2812B LEDs** used to demonstrate lighting effects.

A **5V regulated power source**, sufficient to drive the ESP8266 and LEDs.

Supporting electronic components like a **resistor** on the data line and a **capacitor** for power stabilization.

The device boots up in Access Point mode for initial configuration, allowing the user to enter Wi-Fi credentials. Once connected to a home network, it switches to Station mode and hosts a fully functional web application at http://ambi.local or a static IP address. This interface allows real-time control of lighting parameters such as color, brightness, and animation mode.

When a command is given via the web interface (e.g., selecting a color or triggering an animation), the ESP8266 receives the HTTP request, processes the command, and sends appropriate data signals to the LED strip. This signal tells each LED which color to display and in what sequence, enabling synchronized lighting patterns or user-defined static modes.

The model showcases:

Seamless Wi-Fi-based interaction between a user's device and the lighting system.

Accurate LED control using the FastLED library.

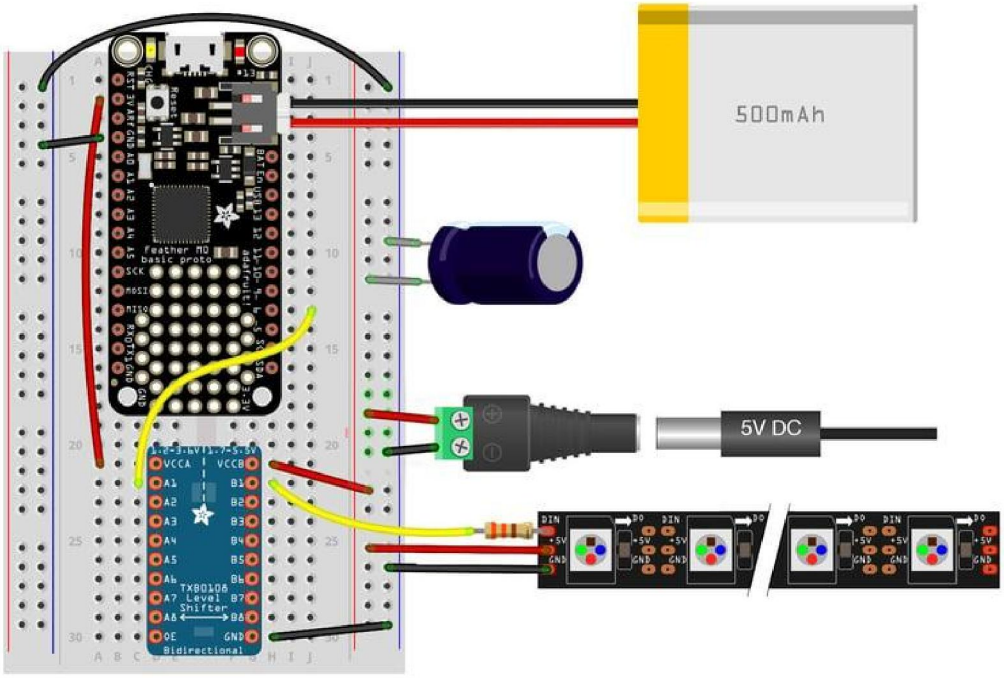
Mobile and desktop compatibility, as the system can be controlled from any standard web browser.

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

Modular and expandable design, allowing additional features like voice control, scheduling, or remote access via internet in future iterations.

This working prototype forms the basis for a scalable home automation lighting system that can be used for room aesthetics, ambient mood settings, festive decorations, or smart task lighting. It is designed to be compact, low-cost, and easily reproducible, making it suitable for both hobbyists and educational demonstrations.

V. CIRCUIT DIAGRAM



VI. CONNECTION FLOW

Power (+5 V & GND): Connect the 5 V supply to VIN on ESP8266 and 5 V (VCC) on the LED strip. Tie all grounds together for a common reference.

Capacitor (100 µF–1000 µF): Place near the LED strip's power input to buffer voltage spikes and maintain smooth performance.

Data Line: Use a GPIO pin (commonly D2/GPIO4 on ESP8266) for data output. Maintain a short wire run to the strip.

Resistor (220–330 Ω): Add in series with the data line, close to the first LED, to prevent signal ringing and safeguard the LEDs.

Logic Level Shifter (optional): Include if extending wiring or facing reliability issues; ensures proper logic-level translation from 3.3 V to 5 V.

LED Strip Input: Data-in (DIN), VCC, and GND wires enter the strip, with all protective components nearby for optimal operation.

VII. CONCLUSION

The proposed ESP8266-based smart room decoration system successfully demonstrates wireless control of addressable LEDs through a web interface. It provides a cost-effective, customizable, and user-friendly platform for ambient lighting using common IoT components. The system is scalable, easy to deploy, and can be enhanced further with automation and smart home integrations.

DOI: 10.48175/IJARSCT-28190

IJARSCT
International Journal of Advanced Research in Science, Communication and Technology
International Open-Access, Double-Blind, Peer-Reviewed, Refereed, Multidisciplinary Online Journal
ISSN: 2581-9429
Volume 5, Issue 8, June 2025
Impact Factor: 7.67

REFERENCES

[1]. Rui Santos, Sara Santos, "ESP8266 NodeMCU Web Server with Arduino IDE," *Random Nerd Tutorials*, [Online]. Available: https://randomnerdtutorials.com/esp8266-nodemcu-web-server-arduino-ide/

[2]. Daniel Garcia, "FastLED Library Documentation," [Online]. Available: https://fastled.io/

[3]. Espressif Systems, "ESP8266 Technical Reference Manual," [PDF]. Available: https://www.espressif.com/sites/default/files/documentation/esp8266-technical_reference_en.pdf

[4]. Arduino, "ESP8266 Core Documentation," *Arduino.cc*, [Online]. Available: https://github.com/esp8266/Arduino

WLED Project, "WLED Documentation," [Online]. Available: https://kno.wled.ge/

[5]. Tontek Design, "TTP223 Capacitive Touch Sensor Module Datasheet," [Online]. Available: https://datasheet.lcsc.com/lcsc/1811161535_Tontek-TTP223-BA6_C82902.pdf

[6]. Adafruit, "NeoPixel Überguide," [Online]. Available: https://learn.adafruit.com/adafruit-neopixel-uberguide