

TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN



Đề tài:

**Phát triển hệ thống đèn LED thông minh RGB với
ESP32**

TÊN LỚP HỌC PHẦN : Phát triển ứng dụng IoT

MÃ HỌC PHẦN: 2024-2025.2.TIN4024.004

GIẢNG VIÊN HƯỚNG DẪN: Võ Việt Dũng

SINH VIÊN THỰC HIỆN: Nguyễn Tâm Pháp

HUẾ, THÁNG 4 NĂM 2025

LỜI CẢM ƠN

Trước tiên, em xin gửi lời tri ân sâu sắc đến quý thầy cô trong khoa Công nghệ Thông tin, đặc biệt là thầy Võ Việt Dũng – giảng viên bộ môn Phát triển ứng dụng IoT. Trong suốt quá trình học tập, thầy đã mang đến những kiến thức nền tảng vững chắc, giúp em mở rộng hiểu biết và định hướng cách áp dụng lý thuyết vào thực tiễn. Sự nhiệt huyết và phong thái giảng dạy dễ hiểu của thầy đã tạo điều kiện thuận lợi để em hoàn thành bài tiểu luận này.

Bài tiểu luận là kết quả của sự tổng hợp kiến thức và quá trình học tập nghiêm túc. Tuy nhiên, với khả năng còn hạn chế, chắc chắn em khó tránh khỏi những thiếu sót. Em rất mong nhận được sự đánh giá, góp ý quý báu từ thầy cô, đặc biệt là thầy Võ Việt Dũng, để em có thể cải thiện và hoàn thiện hơn trong tương lai.

Cuối cùng, em xin chân thành cảm ơn gia đình và bạn bè – những người đã luôn động viên, khích lệ em trong hành trình học tập. Sự ủng hộ này là nguồn động lực quan trọng để em vượt qua những khó khăn và hoàn thành nhiệm vụ học tập của mình.

Em hy vọng bài tiểu luận này sẽ mang lại những giá trị thiết thực và đóng góp phần nhỏ vào việc phát triển ứng dụng IoT. Một lần nữa, em xin bày tỏ lòng biết ơn sâu sắc đến thầy Võ Việt Dũng và tất cả những người đã hỗ trợ em trong quá trình học tập và nghiên cứu.

Em xin trân trọng cảm ơn!

Nguyễn Tâm Pháp

MỤC LỤC

I. Tổng quan	4
1. Giới thiệu tổng quan	4
2. Mục tiêu của hệ thống	5
II. Nội dung	6
1. ESP32	6
2. LED WS2812 (Neopixel WS2812)	7
3. Cơ chế hoạt động	8
III. Thiết kế hệ thống	9
1. Phần cứng	9
2. Phần mềm	10
3. Cài đặt và lập trình	11
IV. Kết luận	15
1. Kết quả đạt được	15
2. Hạn chế và thách thức	15
3. Hướng phát triển	15
4. Kết luận	16
Tài liệu tham khảo	17

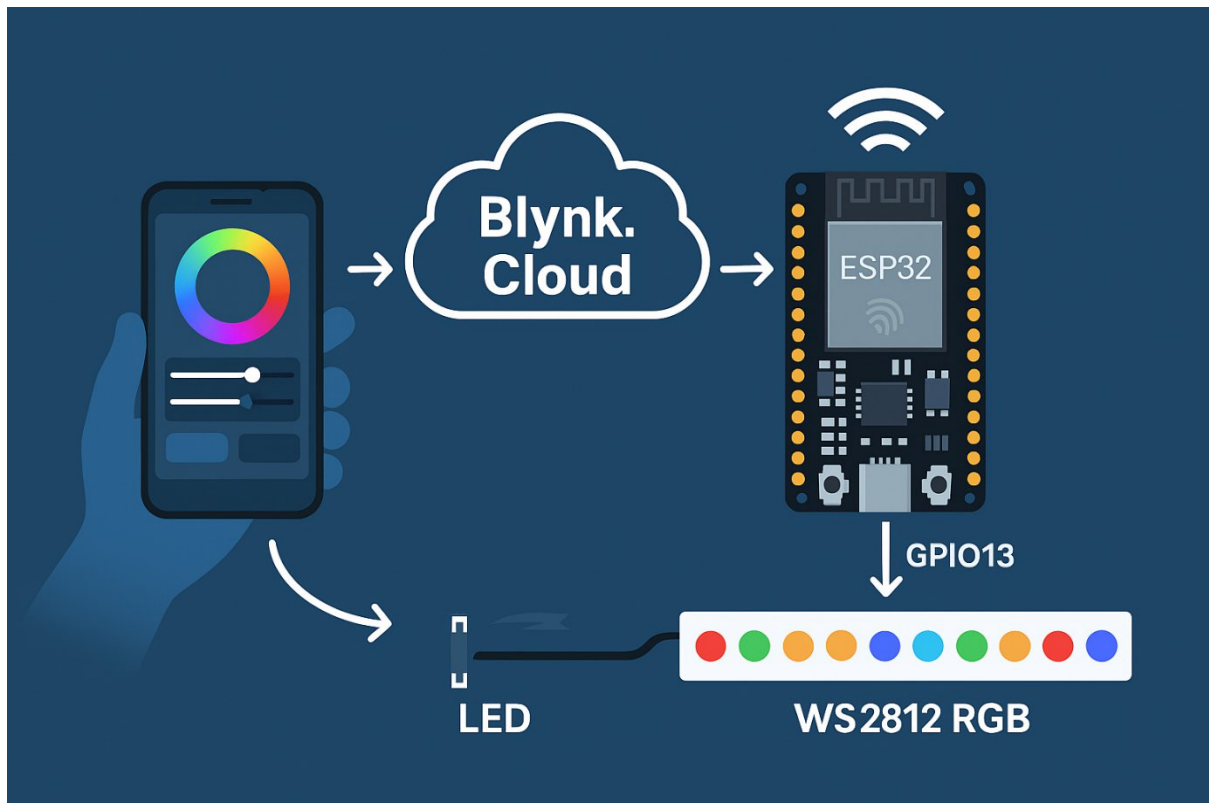
I. Tổng quan

1. Giới thiệu tổng quan

Trong thời đại công nghệ 4.0 hiện nay, các hệ thống nhà thông minh (smart home) đang ngày càng trở nên phổ biến và đóng vai trò quan trọng trong việc nâng cao chất lượng cuộc sống con người. Một trong những thành phần cơ bản nhưng thiết yếu trong hệ sinh thái nhà thông minh chính là hệ thống chiếu sáng. Việc tích hợp công nghệ IoT (Internet of Things) vào các thiết bị chiếu sáng không chỉ mang lại sự tiện lợi mà còn giúp tiết kiệm năng lượng, tăng tính thẩm mỹ và nâng cao trải nghiệm người dùng.

Dự án "Phát triển hệ thống đèn LED thông minh RGB với ESP32" được xây dựng nhằm mục đích tạo ra một giải pháp chiếu sáng hiện đại, linh hoạt và dễ dàng điều khiển từ xa. Hệ thống sử dụng vi điều khiển ESP32 kết hợp với dải đèn LED WS2812 (hay còn gọi là NeoPixel) – một loại đèn LED thông minh có thể điều khiển màu sắc và độ sáng từng bóng một cách độc lập thông qua giao tiếp một dây (One-wire).

Đặc biệt, hệ thống được thiết kế để điều khiển qua ứng dụng web và điều khiển bằng giọng nói, giúp người dùng dễ dàng thay đổi màu sắc, độ sáng hoặc hiệu ứng ánh sáng chỉ bằng vài thao tác trên thiết bị di động hoặc thông qua trợ lý ảo như Google Assistant. Các nền tảng được sử dụng trong dự án bao gồm Visual Studio Code để lập trình, Wokwi.com để mô phỏng và Blynk.Cloud để kết nối thiết bị với Internet và ứng dụng di động.



Hình ảnh mô tả hệ thống

2.Mục tiêu của hệ thống

Thiết kế và triển khai hệ thống điều khiển đèn LED WS2812 sử dụng ESP32, có khả năng điều chỉnh màu sắc và độ sáng theo yêu cầu người dùng.

Phát triển giao diện điều khiển trực quan trên nền tảng web hoặc ứng dụng di động thông qua Blynk.Cloud, cho phép người dùng điều khiển hệ thống chiếu sáng mọi lúc, mọi nơi.

Tích hợp chức năng điều khiển bằng giọng nói, sử dụng Google Assistant hoặc các nền tảng trợ lý ảo tương tự, nhằm tăng tính tiện lợi và hiện đại cho hệ thống.

Tối ưu hóa hiệu suất và độ ổn định của hệ thống, đảm bảo phản hồi nhanh, độ trễ thấp khi nhận lệnh điều khiển, và khả năng mở rộng cho các ứng dụng thực tế như trang trí nội thất, không gian làm việc hoặc chiếu sáng nghệ thuật.

Tạo tiền đề cho việc nghiên cứu và phát triển các ứng dụng IoT khác trong lĩnh vực nhà thông minh, thông qua việc học hỏi và áp dụng các công nghệ như điều khiển từ xa, giao tiếp không dây, và xử lý tín hiệu ánh sáng.

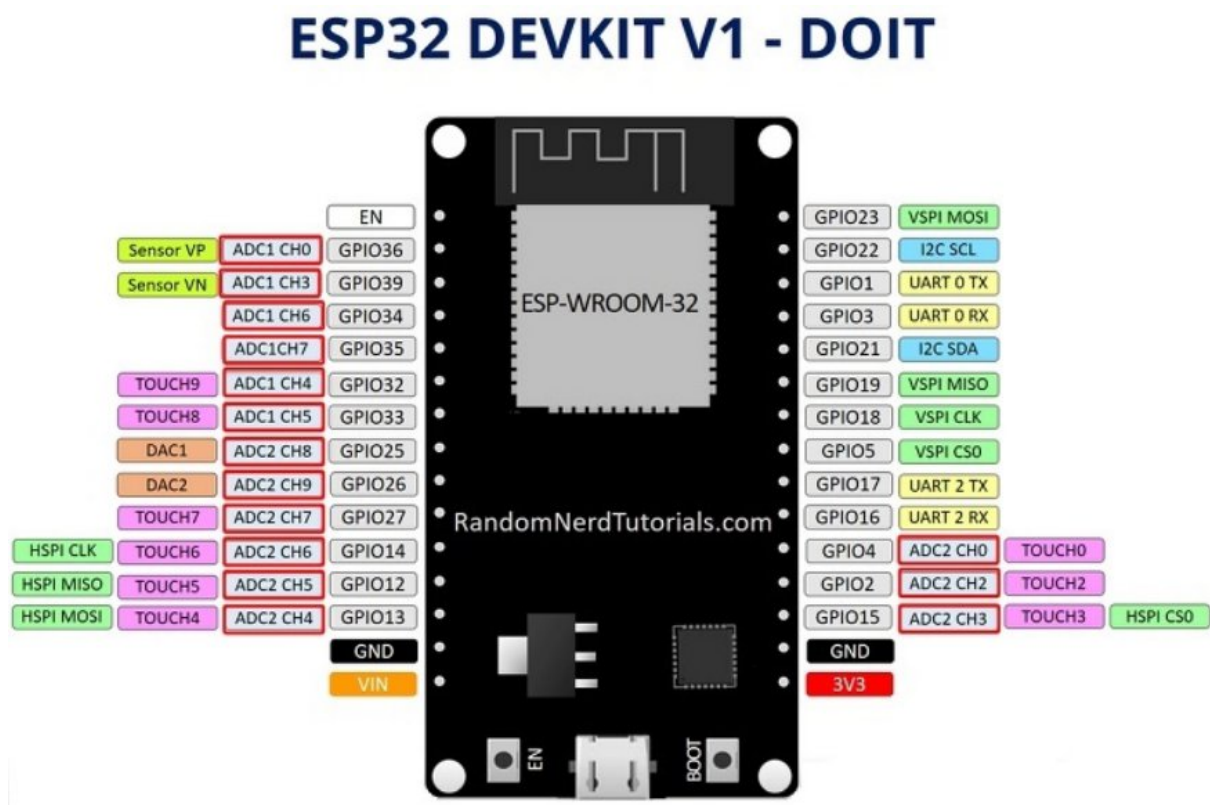
II. Nội dung

1.ESP32

ESP32 là một hệ thống vi điều khiển trên chip (SoC) giá rẻ của Espressif Systems, nhà phát triển của ESP8266 SoC. Nó là sự kế thừa của SoC ESP8266 và có cả hai biến thể lõi đơn và lõi kép của bộ vi xử lý 32-bit Xtensa LX6 của Tensilica với Wi-Fi và Bluetooth tích hợp.

Điểm tốt về ESP32, giống như ESP8266 là các thành phần RF tích hợp của nó như bộ khuếch đại công suất, bộ khuếch đại nhận tiếng ồn thấp, công tắc ăng-ten, bộ lọc và Balun RF. Điều này làm cho việc thiết kế phần cứng xung quanh ESP32 rất dễ dàng vì bạn cần rất ít thành phần bên ngoài.

Một điều quan trọng khác cần biết về ESP32 là nó được sản xuất bằng công nghệ 40 nm công suất cực thấp của TSMC. Vì vậy, việc thiết kế các ứng dụng hoạt động bằng pin như thiết bị đeo, thiết bị âm thanh, đồng hồ thông minh, ..., sử dụng ESP32 sẽ rất dễ dàng.



Hình ảnh minh họa ESP32

***Thông số kỹ thuật:**

- Bộ vi xử lý LX6 32-bit lõi đơn hoặc lõi kép với xung nhịp lên đến 240 MHz.
- 520 KB SRAM, 448 KB ROM và 16 KB SRAM RTC.

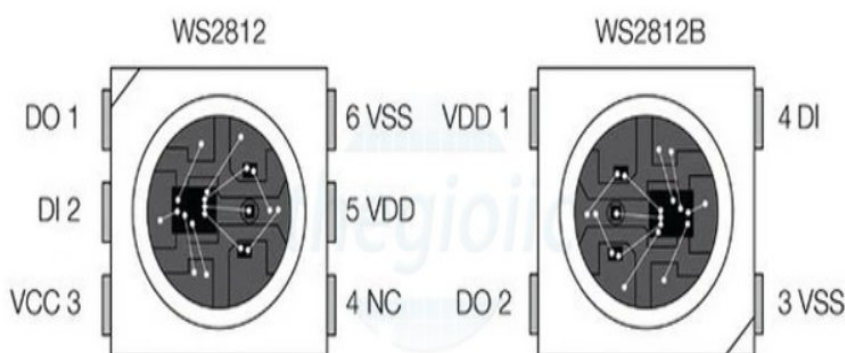
- Hỗ trợ kết nối Wi-Fi 802.11 b / g / n với tốc độ lên đến 150 Mbps.
- Hỗ trợ cho cả thông số kỹ thuật Bluetooth v4.2 và BLE cổ điển.
- 34 GPIO có thể lập trình.
- 18 kênh SAR ADC 12 bit và 2 kênh DAC 8 bit
- Kết nối nối tiếp bao gồm 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC cho giao tiếp mạng LAN vật lý (yêu cầu PHY bên ngoài).
- 1 bộ điều khiển host cho SD / SDIO / MMC và 1 bộ điều khiển slave cho SDIO/ SPI.
- Động cơ PWM và 16 kênh LED PWM.
- Khởi động an toàn và mã hóa Flash.
- Tăng tốc phần cứng mật mã cho AES, Hash (SHA-2), RSA, ECC và RNG.

2.LED WS2812(Neopixel WS2812)

Neopixel WS2812 là một mạch tích hợp 3 đèn Led RGB và 1 IC điều khiển. Điều đặc biệt ở loại led này có thể nối tiếp nhiều bóng với nhau nhưng vẫn có thể điều khiển tới 144 bóng riêng biệt mà chỉ cần dùng 1 đường tín hiệu điều khiển.

WS2812 có ba đèn LED (đỏ, xanh dương và xanh lá) với một bộ điều khiển trong cùng một package. Bộ điều khiển có một thanh ghi 24bit, nhận dữ liệu nối tiếp từ chân DIN, lưu trữ và hiển thị lên LED tương ứng. Thanh ghi 24bit được chia thành ba phần, mỗi phần 8 bit và lưu một giá trị độ sáng khác nhau và có tới 256 mức độ sáng cho mỗi đèn LED. Và vì có ba màu nên có tổng cộng gần 17 triệu màu cho mỗi LED.

Các chân dữ liệu trên đèn LED được thiết kế thành chuỗi, đầu ra của bộ điều khiển được đệm để duy trì tín hiệu ngay cả khi có nhiều đèn LED được kết nối với nhau.



Hình ảnh minh họa Neopixel WS2812

***Thông số kỹ thuật:**

Tên chân	Mô tả
VDD	Chân cấp nguồn LED
DOUT	Chân đầu ra tín hiệu
GDN	Chân cấp tham chiếu ground
DIN	Chân đầu vào tín hiệu
VCC	Chân cấp nguồn cho IC

- Góc nhìn: 120 độ
- Màu đỏ: (620-630nm) @ 550-700mcd
- Màu xanh lá: (515-530nm) @ 1100-1400mcd
- Màu xanh dương: (465-475nm) @ 200-400mcd
- Điện áp cấp từ 3.5V đến 5.3V
- Điện áp đầu vào tín hiệu từ 0,5V đến VCC
- Điện dung đầu vào 15pF
- Dòng điện cấp tín hiệu 1uA

3. Cơ chế hoạt động

Hệ thống đèn LED thông minh RGB sử dụng ESP32 hoạt động theo cơ chế kết hợp giữa phần mềm điều khiển từ xa (Blynk.Cloud) và phần cứng xử lý tín hiệu LED (ESP32 và WS2812). Cụ thể:

1. **Giao tiếp người dùng:** Người dùng tương tác với ứng dụng Blynk trên điện thoại hoặc trình duyệt web. Ứng dụng này cung cấp giao diện với các widget như thanh chọn màu (Color Picker) và thanh trượt độ sáng (Slider).
2. **Truyền tín hiệu qua internet:** Khi người dùng chọn màu hoặc thay đổi độ sáng, tín hiệu này sẽ được gửi tới máy chủ của Blynk.Cloud. ESP32 được kết nối Wi-Fi sẽ liên tục lắng nghe các tín hiệu từ máy chủ thông qua Auth Token riêng.
3. **Phân tích và xử lý tín hiệu:** ESP32 nhận được tín hiệu màu sắc (dưới dạng mã HEX) và mức độ sáng (từ 0 đến 255). Vi điều khiển sẽ tách mã HEX thành ba giá trị màu RGB tương ứng (Red, Green, Blue) và điều chỉnh mức độ sáng tương ứng.
4. **Xuất dữ liệu điều khiển LED:** Sau khi xử lý, ESP32 sẽ gửi chuỗi dữ liệu điều khiển thông qua chân GPIO13 (được nối với chân DIN của dải LED WS2812). Mỗi bóng LED sẽ nhận tín hiệu điều khiển riêng cho nó.

5. **LED phát sáng:** Dải WS2812 sẽ hiển thị đúng màu và độ sáng mà người dùng đã chọn. LED đầu tiên nhận tín hiệu và truyền phần còn lại cho LED tiếp theo, tạo thành chuỗi hoạt động nối tiếp nhau.

III. Thiết kế hệ thống

1. Phần cứng

Hệ thống được thiết kế trên nền tảng vi điều khiển ESP32 với các linh kiện sau:

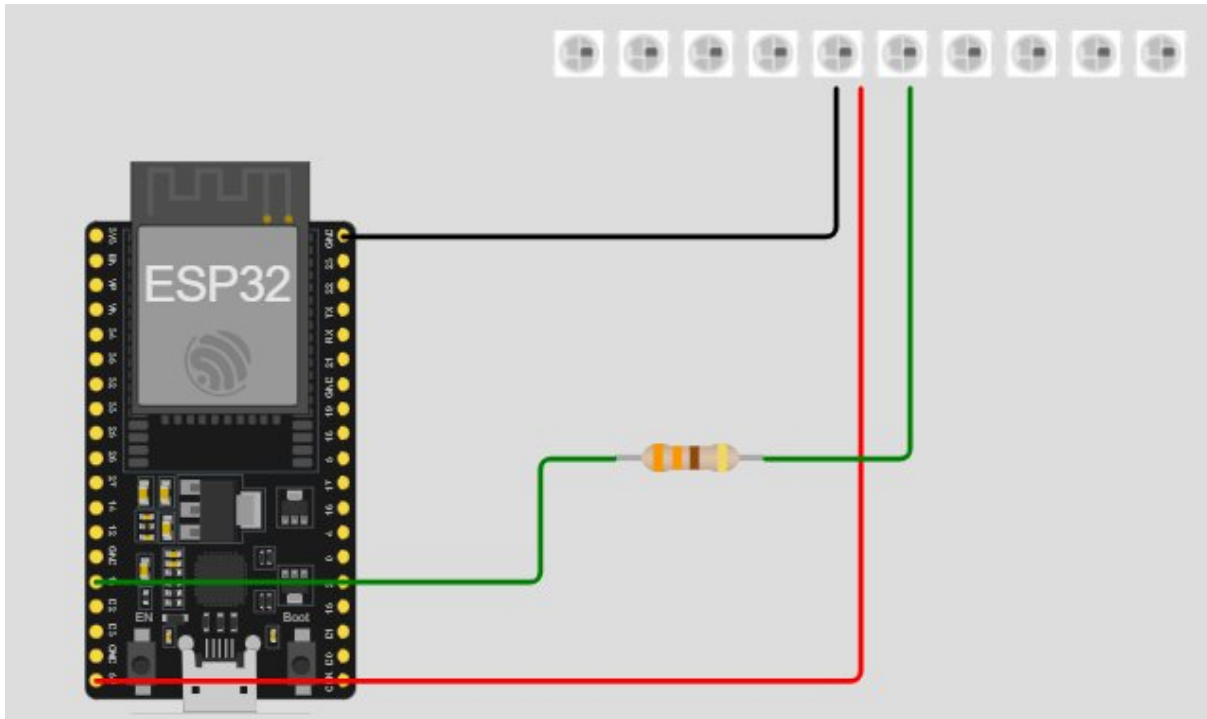
- ESP32 Devkit V1: Vi điều khiển chính, hỗ trợ Wi-Fi, điều khiển LED qua PWM.
- Dải LED WS2812 (NeoPixel): Bao gồm 10 bóng LED RGB có thể lập trình
- Điện trở 330Ω nối giữa GPIO13 và chân DATA của LED WS2812: giúp giảm nhiễu tín hiệu DATA.
- Nguồn điện 5V cấp cho LED: từ ESP32 hoặc nguồn ngoài nếu dùng nhiều LED.

Sơ đồ kết nối cơ bản:

ESP32 GPIO13 → Điện trở 330Ω → DIN WS2812

GND ESP32 → GND LED WS2812

5V ESP32 → VCC LED WS2812



Hình ảnh sơ đồ mạch điện mô phỏng giữa ESP32 và Neopixel WS2812

2. Phần mềm

Các công cụ và nền tảng phần mềm được sử dụng trong dự án bao gồm:

- **Visual Studio Code (PlatformIO)** : Là môi trường phát triển tích hợp (IDE) dùng để viết, biên dịch và nạp chương trình vào vi điều khiển ESP32.
- **Thư viện BlynkSimpleEsp32.h** : Hỗ trợ kết nối ESP32 với nền tảng đám mây Blynk. Nhờ đó, ESP32 có thể nhận lệnh điều khiển từ người dùng thông qua Internet.
- **Thư viện WiFi.h** : Dùng để kết nối ESP32 với mạng Wi-Fi. Đây là bước bắt buộc để thiết bị có thể giao tiếp với Blynk.Cloud.
- **Thư viện Adafruit_NeoPixel.h** : Chuyên dùng để điều khiển dải LED WS2812. Thư viện này hỗ trợ truyền dữ liệu điều khiển LED qua một chân GPIO duy nhất (ví dụ GPIO13), giúp lập trình LED đơn giản hơn và có thể điều chỉnh từng LED riêng lẻ.
- **Ứng dụng Blynk trên điện thoại hoặc nền web Blynk.Cloud** : Cho phép tạo giao diện người dùng để điều khiển LED từ xa. Ứng dụng này giao tiếp với ESP32 qua Internet.

3. Cài đặt và lập trình

Tạo tài khoản và Template trên Blynk.Cloud: Truy cập trang web của Blynk, đăng ký tài khoản, tạo template mới, sau đó lấy Auth Token để cấu hình trong chương trình.

```
#define BLYNK_TEMPLATE_ID "TMPL6jBpsnQWj"  
#define BLYNK_TEMPLATE_NAME "LEDRGB"  
#define BLYNK_AUTH_TOKEN "85cizTr1YnnTFIjAJNudOLfS6zOigqJL"
```

Hình ảnh minh họa cấu hình kết nối với Blynk

Cài đặt ứng dụng Blynk trên điện thoại hoặc sử dụng bản web: Sau khi đăng nhập, tạo giao diện điều khiển gồm các widget như Color Picker (V0) để chọn màu và Slider (V1) để điều chỉnh độ sáng.



Hình ảnh minh họa hệ thống Blynk

Lập trình Visual Studio Code với PlatformIO IDE: Cài đặt thư viện Blynk và Adafruit NeoPixel từ Library Manager. Tạo project mới, khai báo thông tin Wi-Fi, Auth Token và thiết lập các chân kết nối LED.

```

1  #define BLYNK_TEMPLATE_ID "TMPL6j8psnQWj"
2  #define BLYNK_TEMPLATE_NAME "LEDRGB"
3  #define BLYNK_AUTH_TOKEN "85ScizTr1YnnTFIjAJNudOLF56zOigqJL"
4
5  #include <WiFi.h>
6  #include <BlynkSimpleEsp32.h>
7  #include <Adafruit_NeoPixel.h>
8
9  // --- Thông tin kết nối WiFi ---
10 char ssid[] = "TAMPHAP";
11 char pass[] = "tamphap117";
12
13 // --- Blynk Template ---
14 #define BLYNK_TEMPLATE_ID "TMPL6j8psnQWj"
15 #define BLYNK_TEMPLATE_NAME "LEDRGB"
16 char auth[] = "85ScizTr1YnnTFIjAJNudOLF56zOigqJL";
17
18 // --- Cấu hình LED ---
19 #define LED_PIN 13
20 #define NUM_LEDS 10
21 Adafruit_NeoPixel strip(NUM_LEDS, LED_PIN, NEO_GRB + NEO_KHZ800);
22
23 // --- Biến điều khiển ---
24 int brightness = 255; // Mức độ sáng (0-255)
25 int r = 255, g = 255, b = 255; // Màu mặc định ban đầu
26
27 // --- Nhận màu từ Color Picker (V0) ---
28 BLYNK_WRITE(V0) {
29   String color = param.asString();
30   long number = strtol(&color[1], NULL, 16);
31   r = number >> 16;
32   g = number >> 8 & 0xFF;
33   b = number & 0xFF;
34   updateLEDs();
35 }
36
37 // --- Nhận độ sáng từ Slider (V1) ---
38 BLYNK_WRITE(V1) {
39   brightness = param.asInt();
40   updateLEDs();
41 }
42
43 // --- Cập nhật màu và độ sáng cho tất cả LED ---
44 void updateLEDs() {
45   for (int i = 0; i < NUM_LEDS; i++) {
46     strip.setPixelColor(i, strip.Color(r * brightness / 255, g * brightness / 255, b * brightness / 255));
47   }
48   strip.show();
49 }
50
51 // --- Thiết lập ---
52 void setup() {
53   Serial.begin(115200);
54   strip.begin();
55   strip.show(); // Tắt tất cả LED ban đầu
56   Blynk.begin(auth, ssid, pass);
57 }
58
59 // --- Vòng lặp chính ---
60 void loop() {
61   Blynk.run();
62 }
63

```

Hình ảnh minh họa mã nguồn main.cpp

***Quy trình hoạt động của chương trình:**

Chương trình điều khiển LED RGB thông minh sử dụng ESP32 hoạt động theo một chuỗi các bước có trình tự rõ ràng. Quy trình này kết hợp giữa xử lý tín hiệu điều khiển từ người dùng và điều khiển phần cứng LED WS2812. Cụ thể gồm các bước:

Bước 1: Khởi tạo hệ thống

- ESP32 được nạp chương trình và khởi động.
- Kết nối Wi-Fi được thiết lập bằng cách sử dụng SSID và mật khẩu được lập trình sẵn.
- ESP32 kết nối đến nền tảng **Blynk.Cloud** bằng Blynk.begin() cùng với Auth Token định danh thiết bị.

Bước 2: Lắng nghe tín hiệu điều khiển từ Blynk

- Trên ứng dụng Blynk (di động hoặc web), người dùng có thể thay đổi màu sắc qua widget **Color Picker** (chân ảo V0) và điều chỉnh độ sáng bằng widget **Slider** (chân ảo V1).
- Các widget này gửi dữ liệu đến máy chủ của Blynk và chuyển tiếp đến thiết bị ESP32 thông qua internet.

Bước 3: Nhận và xử lý dữ liệu tại ESP32

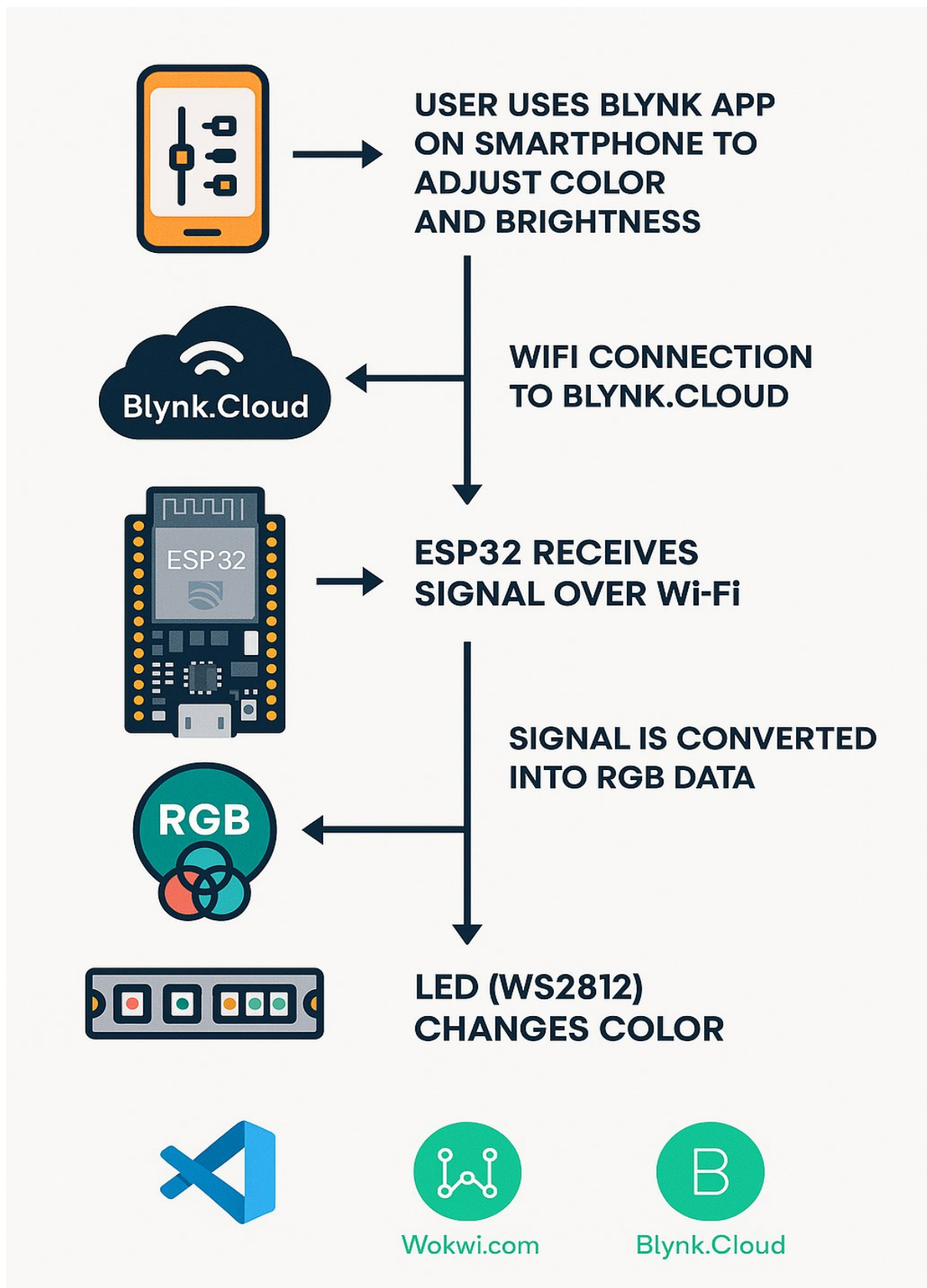
- ESP32 nhận giá trị màu dưới dạng mã HEX từ V0 (VD: #FF0000 cho màu đỏ) và giá trị độ sáng từ V1 (0–255).
- Sau đó, chương trình sẽ phân tách mã HEX thành ba thành phần RGB: Red, Green, Blue.
- Độ sáng sẽ được áp dụng lên các giá trị RGB để điều chỉnh cường độ ánh sáng.

Bước 4: Điều khiển LED WS2812

- ESP32 sử dụng thư viện **Adafruit_NeoPixel** để gửi dữ liệu đến dải LED WS2812.
- Dữ liệu được gửi qua chân GPIO13 (đã nối với chân DIN của LED thông qua điện trở 330Ω).
- Mỗi LED nhận tín hiệu điều khiển và truyền tiếp phần còn lại tới LED kế tiếp, tạo thành một chuỗi đồng bộ ánh sáng.

Bước 5: Hiển thị màu sắc

- Dải LED sẽ phát sáng theo màu và độ sáng mà người dùng đã chọn. Quá trình thay đổi diễn ra gần như **thời gian thực** (real-time).
- Người dùng có thể tiếp tục gửi tín hiệu mới, và hệ thống sẽ cập nhật liên tục mà không cần khởi động lại thiết bị.



Hình ảnh mô tả quy trình hệ thống

IV. Kết luận

1. Kết quả đạt được

Dự án đã xây dựng thành công hệ thống đèn LED RGB thông minh hoạt động ổn định, với khả năng điều khiển thời gian thực thông qua giao diện Blynk.

Các thuật toán điều khiển màu sắc đã được phát triển mượt mà với tốc độ chuyển đổi hiệu ứng dưới 100ms, cho phép thay đổi linh hoạt giữa các chế độ ánh sáng.

Hệ thống đảm bảo đồng bộ trạng thái giữa phần cứng và ứng dụng điều khiển, giúp người dùng có trải nghiệm nhất quán và tin cậy.

ESP32 đã được sử dụng làm nền tảng chính cho hệ thống, cho thấy hiệu quả và khả năng mở rộng cao trong các ứng dụng IoT.

2. Hạn chế và thách thức

Mô phỏng hệ thống trên Wokwi chưa hỗ trợ đầy đủ kết nối với Blynk, gây khó khăn trong việc kiểm tra và thử nghiệm thực tế.

Số lượng đèn LED bị giới hạn trong môi trường mô phỏng, ảnh hưởng đến việc triển khai các hiệu ứng phức tạp.

Độ trễ trong truyền dữ liệu qua internet là không thể tránh khỏi, đôi khi ảnh hưởng đến trải nghiệm điều khiển.

Vấn đề bảo mật vẫn chưa được giải quyết triệt để, tiềm ẩn một số rủi ro khi triển khai trong môi trường thực tế.

4. Hướng phát triển

Mở rộng quy mô hệ thống để hỗ trợ điều khiển đồng bộ hoặc độc lập nhiều dải LED, phù hợp với không gian sống hiện đại.

Nâng cấp giao diện điều khiển với các tính năng nâng cao như quản lý cảnh sáng, cài đặt lịch và điều kiện môi trường.

Tích hợp hệ thống cảm biến (chuyển động, ánh sáng, âm thanh, nhiệt độ...) để phản ứng tự động theo môi trường.

Phát triển khả năng điều khiển bằng giọng nói thông qua Google Assistant, Amazon Alexa hoặc IFTTT.

Ứng dụng trí tuệ nhân tạo nhằm học hỏi thói quen người dùng và tối ưu hóa điều khiển ánh sáng theo nhu cầu thực tế.

5. Kết luận

Dự án đèn LED RGB thông minh sử dụng ESP32 không chỉ là một sản phẩm ứng dụng công nghệ IoT, mà còn là bước đệm quan trọng hướng đến xây dựng không gian sống hiện đại, linh hoạt và tiết kiệm năng lượng.

Thông qua quá trình triển khai, dự án đã giúp tôi tích lũy kiến thức thực tiễn trong các lĩnh vực như vi điều khiển, lập trình nhúng, mạng không dây và thiết kế giao diện người dùng.

Những trải nghiệm trong việc vượt qua hạn chế của môi trường mô phỏng cũng góp phần rèn luyện kỹ năng giải quyết vấn đề và tư duy sáng tạo.

Với nền tảng đã được xây dựng, hệ thống có tiềm năng cao để mở rộng và tích hợp vào các giải pháp nhà thông minh, góp phần thúc đẩy xu hướng phát triển không gian sống bền vững trong tương lai.

Tài liệu tham khảo

<https://dientutuonglai.com/esp32-la-gi.html>

<https://www.iotzone.vn/esp32/esp32-co-ban/esp32-adc-doc-gia-tri-analog/>

<https://www.thegioiic.com/tin-tuc/gioi-thieu-ve-neopixel-ws2812>

<http://arduino.vn/tutorial/1570-gioi-thieu-module-esp32-va-huong-dan-cai-trinh-bien-dich-tren-arduino-ide>

<http://arduino.vn/bai-viet/1006-fiat-lux-hay-co-anh-sang-phan-2-viet-code-blink-thanh-cho-neopixel-ws2812>