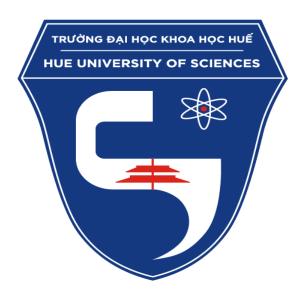






BÀI TIỂU LUẬN



BÀI THI KẾT THÚC HỌC PHẦN

Tên Học Phần: Phát triển ứng dụng IoT

Đề tài tiểu luận: Phát triển hệ thống đèn led thông minh RGB với ESP32

Sinh viên thực hiện: Nguyễn Anh Trí

Lớp học phần: 2024-2025.2.TIN4024.004

Giảng viên hướng dẫn: Võ Việt Dũng

Huế, ngày 4 tháng 4 năm 2025

PHẦN MỞ ĐẦU	
Chương 1: Tổng quan đề tài3	
1.1. Đặt vấn đề nghiên cứu3	
1.2. Mục tiêu nghiên cứu của đề tài4	
1.3. Đưa ra giải pháp5	
1.4. Kết luận chương 16	
Chương 2: Nội dung của đề tài6	
2.1. Các nội dung thực hiện6	
2.1.1. Nghiên cứu và lựa chọn công nghệ	
2.1.2. Thiết kế và xây dựng hệ thống phần cứng	
2.1.3. Phát triển phần mềm điều khiển	
2.1.4. Thiết kế giao diện web điều khiển	
2.1.5. Mô phỏng và kiểm thử hệ thống	
2.1.6. Đánh giá và tối ưu hóa	
2.2 Thiết kế mạch IoT8	
1. ESP328	
2. LED WS2812 (NeoPixel)9	
3. Trình duyệt web (Thiết bị người dùng)9	
4. Điện trở (thường $\sim 330\Omega - 470\Omega$)	
5. Tụ điện (thường $1000\mu F/6.3V$ trở lên)	
6. Nguồn điện (5V DC)	
2.3 Nguyên lí hoạt động10	
2.4 Mảnh vật lí12	
Chương 3: Cài đặt hệ thống12	
3.1 Công cụ phần cứng và phần mềm sử dụng12	
3.2 Quy trình cài đặt và cấu hình hệ thống13	
Chương 4 Kết quả thực hiện14	
Chương 5: Kết luận và hướng phát triển15	
5.1 Kết luận	
5.2 Hướng phát triển15	

PHẦN MỞ ĐẦU

Trong bối cảnh cuộc cách mạng công nghiệp 4.0 phát triển mạnh mẽ, Internet of Things (IoT) đã và đang trở thành một xu hướng quan trọng trong việc xây dựng các hệ thống tự động hóa, đặc biệt trong lĩnh vực nhà thông minh (Smart Home). Một trong những ứng dụng phổ biến của IoT trong nhà thông minh chính là hệ thống chiếu sáng thông minh, cho phép người dùng điều khiển đèn từ xa, thay đổi màu sắc, độ sáng theo nhu cầu và hoàn cảnh sử dụng.

Trong tiểu luận này, tôi thực hiện đề tài "Phát triển hệ thống đèn LED thông minh RGB với ESP32", với mục tiêu xây dựng một hệ thống đèn LED có khả năng:

- Điều khiển màu sắc và độ sáng của dải đèn WS2812 thông qua giao diện web;
- Điều khiển bằng giọng nói, thông qua tích hợp với Google Assistant;
- Kết nối và gửi dữ liệu trạng thái của hệ thống đến các nền tảng IoT như Blynk,
 ThingSpeak hoặc Telegram;
- Mô phỏng hệ thống trên Wokwi, không cần phần cứng thực.
- Đề tài không chỉ giúp sinh viên hiểu rõ hơn về việc ứng dụng ESP32 trong các hệ thống IoT mà còn giúc luyện tập tư duy thiết kế hệ thống, khai thác nền tảng mô phỏng hiện đại, đồng thời tiếp cận nhiều công nghệ hiện đại như Web server, API, hay giao tiếp bằng giọng nói.

Chương 1: Tổng quan đề tài

1.1. Đặt vấn đề nghiên cứu

Trong bối cảnh công nghệ ngày càng phát triển, các thiết bị thông minh đang dần trở thành một phần không thể thiếu trong đời sống hàng ngày. Trong đó, hệ thống chiếu sáng thông minh là một lĩnh vực nhận được nhiều sự quan tâm nhờ khả năng mang lại sự tiện lợi, tiết kiệm năng lượng và tạo nên không gian sống hiện đại. Việc điều khiển ánh sáng không chỉ dừng lại ở bật/tắt truyền thống, mà còn mở rộng đến khả năng điều chỉnh màu sắc, độ sáng, thậm chí là tạo hiệu ứng theo ngữ cảnh sử dụng.

ESP32, một vi điều khiển tích hợp Wi-Fi, đã trở thành một nền tảng phổ biến trong lĩnh vực phát triển các ứng dụng IoT nhờ tính năng mạnh mẽ, giá thành rẻ và cộng đồng hỗ trợ rộng lớn. Kết hợp với loại đèn LED RGB WS2812 có khả năng hiển thị hàng triệu màu sắc và dễ dàng lập trình, việc xây dựng một hệ thống đèn thông minh trở nên khả thi ngay cả với sinh viên hoặc nhà phát triển độc lập.

Việc điều khiển LED WS2812 qua giao diện web giúp người dùng có thể thay đổi màu sắc và độ sáng theo thời gian thực, chỉ với một trình duyệt trên điện thoại hoặc máy tính mà không cần cài đặt thêm ứng dụng. Điều này không chỉ làm tăng tính linh hoạt trong sử dụng, mà còn mở ra hướng đi mới trong việc xây dựng các giải pháp chiếu sáng thông minh giá rẻ, phù hợp với nhà ở, quán cà phê, hoặc các không gian sáng tạo.

Với lý do đó, đề tài "Phát triển hệ thống đèn LED thông minh RGB với ESP32" được thực hiện với mục tiêu xây dựng một hệ thống điều khiển đèn WS2812 bằng ESP32 thông qua giao diện web đơn giản, dễ sử dụng, và có thể mở rộng trong tương lai.

1.2. Mục tiêu nghiên cứu của đề tài

Đề tài hướng đến việc xây dựng một hệ thống đèn LED thông minh sử dụng vi điều khiển ESP32 và LED WS2812, cho phép người dùng điều khiển màu sắc và độ sáng của đèn thông qua giao diện web. Các mục tiêu cụ thể của đề tài bao gồm:

- Tìm hiểu nguyên lý hoạt động của vi điều khiển ESP32 và cách thiết lập ESP32 làm máy chủ web (web server) phục vụ điều khiển thiết bị ngoại vi từ trình duyệt.
- Tìm hiểu cấu tạo và cách điều khiển LED WS2812, bao gồm điều chỉnh màu sắc bằng mã RGB và độ sáng bằng kỹ thuật PWM hoặc thay đổi giá trị đầu vào.
- Thiết kế và xây dựng giao diện web điều khiển, cho phép người dùng:
 - O Chọn màu sắc bằng bộ chọn màu (color picker).
 - Thay đổi độ sáng bằng thanh trượt (slider).
 - o Gửi lệnh trực tiếp đến ESP32 thông qua mạng Wi-Fi nội bộ.

- Lập trình hệ thống điều khiển trên ESP32 để tiếp nhận yêu cầu từ giao diện web, xử lý dữ liệu và điều khiển đèn LED một cách chính xác, mượt mà.
- Mô phỏng hệ thống trên nền tảng Wokwi, đảm bảo khả năng hoạt động ổn định trước khi triển khai thực tế.
- Đánh giá hiệu quả hoạt động của hệ thống qua mô phỏng và phân tích các yếu tố như độ trễ, tính tương tác thời gian thực và khả năng mở rộng.

Thông qua đề tài này, người thực hiện có cơ hội củng cố kiến thức về lập trình vi điều khiển, thiết kế giao diện người dùng đơn giản, đồng thời làm quen với việc phát triển các ứng dụng IoT phục vụ đời sống thực tế.

1.3. Đưa ra giải pháp

Để đạt được mục tiêu nghiên cứu đã đề ra, đề tài lựa chọn giải pháp sử dụng vi điều khiển ESP32 làm trung tâm điều khiển, kết hợp với LED RGB WS2812 để hiển thị màu sắc theo yêu cầu của người dùng. Người dùng sẽ tương tác với hệ thống thông qua một giao diện web đơn giản, được tạo và phục vụ trực tiếp từ ESP32 mà không cần đến máy chủ trung gian.

Giải pháp cụ thể như sau:

- ESP32 đóng vai trò như một Web Server: Thiết bị ESP32 sẽ được lập trình để khởi tạo một điểm truy cập Wi-Fi (Access Point) hoặc kết nối vào mạng Wi-Fi sẵn có. Sau đó, nó sẽ tạo ra một trang web chứa giao diện điều khiển với bộ chọn màu và thanh trượt độ sáng.
- Thiết kế giao diện web trực quan: Giao diện được viết bằng HTML/CSS và JavaScript, cho phép người dùng dễ dàng lựa chọn màu sắc (theo mã RGB) và điều chỉnh độ sáng. Khi người dùng thay đổi các tham số này, giao diện sẽ gửi dữ liệu tương ứng đến ESP32 thông qua HTTP request (thường là GET hoặc POST).
- Xử lý yêu cầu và điều khiển LED WS2812: ESP32 tiếp nhận dữ liệu từ giao diện web, phân tích và điều khiển LED WS2812 thông qua thư viện phù hợp (ví dụ: Adafruit NeoPixel hoặc FastLED). Dữ liệu điều khiển bao gồm:

- o Mã màu RGB (ví dụ: đỏ, xanh lá, xanh dương hoặc pha trộn).
- Mức độ sáng từ 0 đến 255.
- Mô phỏng trên nền tảng Wokwi: Để đảm bảo giải pháp hoạt động hiệu quả trước khi triển khai thực tế, toàn bộ hệ thống sẽ được mô phỏng trên nền tảng Wokwi. Đây là một công cụ hỗ trợ lập trình và kiểm thử ESP32 cùng các linh kiện phổ biến trong môi trường trực quan, giúp rút ngắn thời gian phát triển và giảm thiểu lỗi phần cứng.

Giải pháp này có ưu điểm là đơn giản, dễ triển khai, không phụ thuộc vào dịch vụ bên ngoài, đồng thời có thể mở rộng để điều khiển nhiều LED hơn hoặc bổ sung các hiệu ứng ánh sáng tùy chỉnh trong tương lai.

1.4. Kết luận chương 1

Chương 1 đã trình bày tổng quan về đề tài "Phát triển hệ thống đèn LED thông minh RGB với ESP32", từ việc đặt vấn đề nghiên cứu, xác định mục tiêu cần đạt đến việc đề xuất giải pháp thực hiện. Trong bối cảnh nhà thông minh ngày càng phát triển, việc điều khiển hệ thống chiếu sáng thông qua ứng dụng web không chỉ mang lại tính tiện lợi mà còn mở ra nhiều tiềm năng ứng dụng thực tế.

Giải pháp đề xuất sử dụng ESP32 kết hợp với LED WS2812 cho phép xây dựng một hệ thống đơn giản nhưng hiệu quả, có khả năng điều khiển màu sắc và độ sáng đèn theo thời gian thực thông qua trình duyệt web. Đây là tiền đề để triển khai các ứng dụng mở rộng hơn trong lĩnh vực tự động hóa và Internet vạn vật (IoT).

Các chương tiếp theo sẽ trình bày chi tiết hơn về cơ sở lý thuyết, thiết kế hệ thống, mô phỏng, kết quả và đánh giá giải pháp nhằm hoàn thiện đề tài một cách toàn diện.

Chương 2: Nội dung của đề tài

2.1. Các nội dung thực hiện

Để thực hiện đề tài "Phát triển hệ thống đèn LED thông minh RGB với ESP32", các công việc chính sẽ được tiến hành theo các bước cụ thể dưới đây:

2.1.1. Nghiên cứu và lựa chọn công nghệ

- ESP32: Lựa chọn ESP32 làm vi điều khiển trung tâm điều khiển vì tính năng mạnh mẽ, hỗ trợ cả Wi-Fi và Bluetooth, giúp dễ dàng kết nối với các thiết bị khác và xây dựng hệ thống IoT.
- LED WS2812: Lựa chọn LED WS2812 (NeoPixel) cho hệ thống chiếu sáng vì khả năng điều khiển màu sắc và độ sáng dễ dàng qua giao tiếp 1 dây (one-wire protocol), phù hợp với yêu cầu của đề tài.
- Giao diện web: Phát triển giao diện điều khiển web sử dụng HTML, CSS và JavaScript, giúp người dùng dễ dàng tương tác với hệ thống từ bất kỳ thiết bị nào có trình duyệt web.

2.1.2. Thiết kế và xây dựng hệ thống phần cứng

- Mạch điện: Xây dựng sơ đồ mạch điện kết nối ESP32 với LED
 WS2812. Mạch điện sẽ bao gồm các thành phần như nguồn cung cấp cho LED, điện trở bảo vệ, tụ lọc và kết nối GPIO của ESP32 với chân
 Data In của LED WS2812.
- Mô phỏng trên Wokwi: Sử dụng công cụ mô phỏng Wokwi để kiểm tra mạch điện, kiểm thử phần mềm điều khiển và đảm bảo hệ thống hoạt động ổn định trước khi triển khai thực tế.

2.1.3. Phát triển phần mềm điều khiển

- Lập trình ESP32: Viết chương trình cho ESP32 sử dụng môi trường phát triển Arduino IDE hoặc ESP-IDF để tạo Web Server phục vụ giao diện web. Chương trình sẽ xử lý các yêu cầu từ giao diện web để điều khiển màu sắc và độ sáng của LED WS2812.
- Điều khiển LED WS2812: Sử dụng thư viện như Adafruit NeoPixel hoặc FastLED để điều khiển các dải LED WS2812, bao gồm việc điều chỉnh màu sắc và độ sáng theo giá trị nhận được từ người dùng.

2.1.4. Thiết kế giao diện web điều khiển

 Giao diện người dùng: Xây dựng giao diện web với các thành phần chính như bộ chọn màu (color picker) và thanh trượt độ sáng (brightness

- slider). Giao diện này sẽ cho phép người dùng dễ dàng tương tác với hệ thống để thay đổi các tham số của đèn LED.
- Giao tiếp giữa Web và ESP32: Thiết lập giao tiếp giữa giao diện web và
 ESP32 thông qua HTTP requests (GET hoặc POST). Dữ liệu người
 dùng nhập vào giao diện web sẽ được gửi tới ESP32, giúp điều khiển
 LED một cách chính xác.

2.1.5. Mô phỏng và kiểm thử hệ thống

- Mô phỏng hoạt động: Sử dụng nền tảng Wokwi để mô phỏng hoạt động của hệ thống, kiểm tra khả năng điều khiển LED WS2812 qua giao diện web và đảm bảo sự ổn định của hệ thống.
- Kiểm thử tính năng: Thực hiện kiểm thử trên ESP32 để đảm bảo việc điều khiển màu sắc và độ sáng của LED hoạt động đúng theo yêu cầu của người dùng. Đo lường độ trễ và đánh giá hiệu suất của hệ thống.

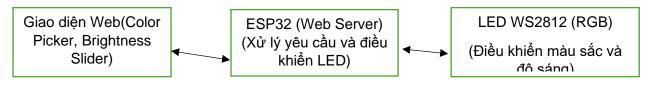
2.1.6. Đánh giá và tối ưu hóa

Đánh giá kết quả mô phỏng: Phân tích hiệu quả của hệ thống qua mô phỏng và so sánh với yêu cầu ban đầu của đề tài. Đánh giá khả năng điều khiển và độ ổn định của hệ thống khi người dùng thay đổi các tham số màu sắc và độ sáng.

Tối ưu hóa hệ thống: Đề xuất các cải tiến về phần mềm, phần cứng hoặc giao diện người dùng để nâng cao hiệu quả và trải nghiệm người dùng.

2.2 Thiết kế mạch IoT

• Sơ đồ logic của hệ thống:



- Chức năng các linh kiện:
 - 1. ESP32

Chức năng:

O Là vi điều khiển trung tâm của hệ thống.

- Tích hợp Wi-Fi, giúp tạo điểm truy cập hoặc kết nối mạng để giao tiếp với trình duyệt web.
- Tạo Web Server để người dùng truy cập và điều khiển LED từ xa.
- Tiếp nhận và xử lý dữ liệu điều khiển (mã màu RGB, độ sáng).
- Gửi tín hiệu điều khiển đến LED WS2812.
 Ưu điểm:
- Tốc độ xử lý nhanh, hỗ trợ đa tác vụ.
- O Có thể lập trình dễ dàng qua Arduino IDE hoặc ESP-IDF.
- o Hỗ trợ nhiều giao tiếp: GPIO, PWM, UART, SPI, I2C,...



2. LED WS2812 (NeoPixel)

Chức năng:

- O Là loại LED RGB tích hợp điều khiển, có thể phát ra hàng triệu màu sắc.
- Mỗi LED có chip điều khiển riêng, cho phép điều khiển màu sắc và độ sáng độc lập qua một dây tín hiệu.
- Thực hiện hiển thị ánh sáng theo yêu cầu từ ESP32.
 Ưu điểm:
- O Điều khiển theo chuỗi nhiều LED với chỉ một chân tín hiệu.
- Hiệu ứng ánh sáng mượt mà, linh hoạt, dễ lập trình với thư viện hỗ trợ như
 Adafruit NeoPixel, FastLED.
 - 3. Trình duyệt web (Thiết bị người dùng) Chức năng:
 - O Là phương tiện giao tiếp giữa người dùng và hệ thống.

- Giao diện web cho phép người dùng điều chỉnh màu sắc (qua color picker)
 và độ sáng (qua slider).
- o Gửi HTTP request đến ESP32 để điều khiển LED.

Ưu điểm:

- O Không cần cài đặt phần mềm.
- Có thể điều khiển từ bất kỳ thiết bị nào có kết nối Wi-Fi (điện thoại, máy tính bảng, laptop...).
- 4. Điện trở (thường $\sim 330\Omega 470\Omega$)

Chức năng:

- Gắn nối tiếp giữa chân tín hiệu của ESP32 và chân Data In của LED WS2812.
- O Giảm nhiễu, bảo vệ mạch khỏi xung điện áp đột ngột.

Ưu điểm:

- o Bảo vệ linh kiện, giúp tín hiệu truyền ổn định hơn.
- 5. Tụ điện (thường 1000µF/6.3V trở lên)

Chức năng:

- O Gắn song song với nguồn cấp cho LED để lọc nhiễu và ổn định điện áp.
- O Giúp LED hoạt động ổn định khi có nhiều LED hoạt động đồng thời.

Ưu điểm:

- Tăng tuổi thọ và độ tin cậy của hệ thống.
- 6. Nguồn điện (5V DC)

Chức năng:

- Cung cấp năng lượng cho ESP32 và các LED WS2812.
- Nếu số lượng LED nhiều, cần nguồn riêng biệt đủ dòng (thường 5V –
 2A hoặc hơn).

2.3 Nguyên lí hoạt động

Hệ thống điều khiển đèn LED RGB thông minh sử dụng ESP32 và LED WS2812 hoạt động dựa trên mô hình giao tiếp giữa người dùng và thiết bị IoT thông qua mạng Wi-Fi. Quá trình hoạt động của hệ thống có thể chia thành các giai đoạn chính như sau:

• Bước 1: Khởi tạo hệ thống

Khi được cấp nguồn, ESP32 sẽ khởi động và kết nối vào mạng Wi-Fi nội bộ hoặc tạo điểm truy cập riêng (Access Point).

Đồng thời, ESP32 khởi tạo một Web Server để sẵn sàng phục vụ các yêu cầu HTTP từ trình duyệt web của người dùng.

• Bước 2: Người dùng truy cập giao diện điều khiển

Người dùng sử dụng trình duyệt web (trên điện thoại, máy tính hoặc máy tính bảng) để truy cập vào địa chỉ IP của ESP32.

Giao diện web do ESP32 cung cấp hiển thị các thành phần:

Color Picker: Cho phép chọn màu RGB.

Brightness Slider: Cho phép điều chỉnh độ sáng (từ 0% đến 100%).

• Bước 3: Gửi lệnh điều khiển

Khi người dùng thay đổi giá trị màu sắc hoặc độ sáng, giao diện web gửi yêu cầu HTTP (GET hoặc POST) chứa thông tin màu sắc và độ sáng đến ESP32.

Dữ liệu gửi có thể là các tham số như:

R - giá trị màu đỏ (0-255)

G – giá trị màu xanh lá (0–255)

 $B-gi\acute{a}$ trị màu xanh dương (0–255)

Brightness – mức độ sáng tổng thể

• Bước 4: ESP32 xử lý dữ liệu

ESP32 tiếp nhận yêu cầu, trích xuất các tham số từ HTTP request.

Sau đó, vi điều khiển tính toán màu sắc và độ sáng, sau đó truyền lệnh điều khiển đến LED WS2812 thông qua giao tiếp một dây.

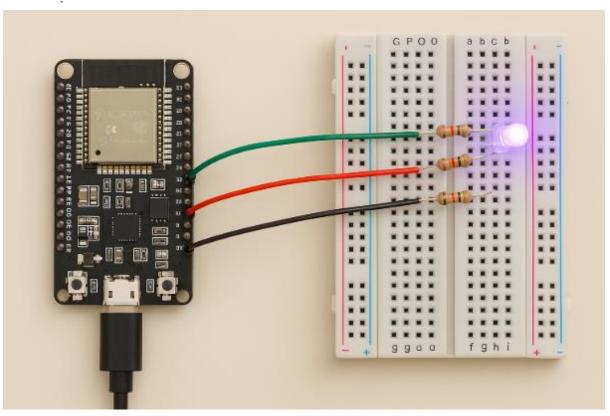
• Bước 5: LED WS2812 phản hồi

LED WS2812 nhận tín hiệu từ ESP32 và hiển thị màu sắc tương ứng.

Đèn có thể hiển thị bất kỳ màu nào trong không gian RGB, với độ sáng điều chỉnh được nhờ kỹ thuật điều xung độ rộng (PWM – Pulse Width Modulation).

Nếu có nhiều LED trong chuỗi, mỗi LED có thể nhận dữ liệu riêng và hiển thị màu khác nhau.

2.4 Månh vât lí



Chương 3: Cài đặt hệ thống

3.1 Công cụ phần cứng và phần mềm sử dụng

- Phần cứng
 - ESP32 Dev Module: Vi điều khiển có tích hợp Wi-Fi, là trung tâm xử lý của hệ thống.
 - LED WS2812 RGB: LED RGB tích hợp chip điều khiển, cho phép đổi màu linh hoạt
 - Điện trở 330Ω: Giới hạn dòng điện cho LED, bảo vệ linh kiện.

 Breadboard + Dây nối (jumper wires): Dùng để kết nối các linh kiện với nhau

Phần mềm

- o Arduino IDE: Môi trường lập trình chính để nạp chương trình cho ESP32.
- o Trình duyệt Web truy cập giao diện điều khiển đèn LED
- Wokwi (wokwi.com) mô phỏng phần cứng (ESP32, LED WS2812...) trực tuyến
- Adafruit NeoPixel Library thư viện Arduino hỗ trợ điều khiển LED
 WS2812
- HTML/CSS/JavaScript cơ bản tạo giao diện web điều khiển màu sắc và độ sáng
- Wi-Fi nội bộ / Access Point Kết nối ESP32 với thiết bị người dùng để điều khiển từ xa

3.2 Quy trình cài đặt và cấu hình hệ thống

- Bước 1: Cài đặt Arduino IDE và thư viện ESP32

Tải và cài đặt Arduino IDE từ https://www.arduino.cc/en/software.

Thêm board ESP32 vào Arduino IDE bằng cách:

Vào File → Preferences → Thêm URL:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Vào Tools → Board Manager → Tìm "ESP32" và nhấn Install.

-Bước 2: Viết và nạp chương trình

Mở Arduino IDE và viết chương trình điều khiển LED RGB bằng tín hiệu PWM.

Khai báo các chân GPIO điều khiển từng màu LED.

Kết nối ESP32 với máy tính qua cáp USB và chọn đúng cổng COM.

Nhấn Upload để nạp chương trình.

-Bước 3: Kết nối và kiểm tra

Sau khi nạp xong, ESP32 sẽ tự kết nối Wi-Fi và sẵn sàng nhận lệnh.

Kiểm tra bằng giao diện Web

Quan sát LED RGB thay đổi màu theo lệnh gửi đến.

-Bước 4: Mô phỏng (nếu không có thiết bị thật)

Truy cập Wokwi.com, tạo project với board ESP32.

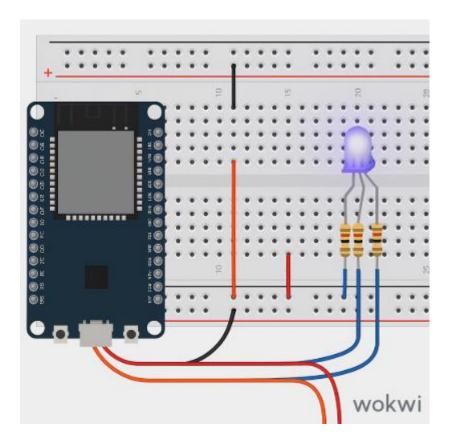
Kéo các linh kiện cần thiết (ESP32, LED RGB, điện trở) vào và kết nối đúng sơ đồ.

Dán mã Arduino vào trình soạn thảo và chạy mô phỏng.

Chương 4 Kết quả thực hiện

Khi chương trình được nạp và chạy trong Wokwi:

- LED RGB sáng lên theo đúng giá trị màu đã lập trình.
- Khi thay đổi giá trị R, G, B trong mã nguồn hoặc giao diện điều khiển, màu sắc của LED thay đổi tương ứng.
- Mạch hoạt động ổn định, không có lỗi ngắn mạch hay sai chức năng.



Chương 5: Kết luận và hướng phát triển

5.1 Kết luân

Trong khuôn khổ đề tài, hệ thống đèn LED RGB thông minh sử dụng ESP32 và dải LED WS2812 đã được xây dựng và hoạt động ổn định. Hệ thống cho phép:

- Điều khiển màu sắc và độ sáng của đèn LED WS2812 một cách linh hoạt thông qua giao diện web thân thiện với người dùng.
- ESP32 đóng vai trò như một máy chủ web (Web Server), xử lý các yêu cầu từ phía người dùng và cập nhật trạng thái đèn LED tương ứng.
- Giao diện được xây dựng đơn giản, dễ sử dụng, có thể truy cập từ bất kỳ thiết bị nào hỗ trợ trình duyệt (điện thoại, máy tính bảng, PC).

Hệ thống đã chứng minh được **tính hiệu quả, chi phí thấp**, và **ứng dụng thực tế cao**, đặc biệt phù hợp trong các ứng dụng trang trí thông minh, chiếu sáng nội thất, hoặc làm nền tảng cho các hệ thống IoT phức tạp hơn.

5.2 Hướng phát triển

Mặc dù hệ thống hiện tại đã đáp ứng yêu cầu cơ bản, vẫn còn nhiều tiềm năng để nâng cấp và mở rộng trong tương lai:

- **Kết nối MQTT / Blynk / Firebase**: Giúp tích hợp vào các hệ sinh thái IoT lớn hơn, hoặc điều khiển qua Internet (không cần chung mạng Wi-Fi).
- Thêm các hiệu ứng động: Tạo các chế độ hiển thị ánh sáng như chuyển màu tự động, nhấp nháy theo nhạc (âm thanh), sóng màu,...
- Hỗ trợ điều khiển bằng giọng nói: Thông qua Google Assistant hoặc Alexa.
- Lưu cấu hình màu sắc và hiệu ứng: Đảm bảo trạng thái ánh sáng được khôi phục sau khi mất điện hoặc khởi động lại.
- Tối ưu giao diện web: Giao diện responsive, tích hợp biểu tượng hiệu ứng,
 lịch trình bật/tắt đèn,...