

TIỂU LUẬN MÔN PHÁT TRIỂN ỨNG DỤNG IoT ĐỀ TÀI: PHÁT TRIỂN HỆ THỐNG ĐÈN LED THÔNG MINH RGB VỚI ESP32

MỤC LỤC

1. [Giới thiệu chung](#)
2. [Mục tiêu đề tài](#)
3. [Tổng quan công nghệ](#)
4. [Thiết bị sử dụng](#)
5. [Kiến trúc hệ thống và sơ đồ khối](#)
6. [Mô phỏng phần cứng trên Wokwi](#)
7. [Điều khiển hệ thống](#)
8. [Phân tích dữ liệu cảm biến qua ThingSpeak](#)
9. [Thuật toán hoạt động hệ thống](#)
10. [Tính năng nâng cao và mở rộng](#)
11. [Đánh giá hệ thống](#)
12. [Kết luận](#)
13. [Tài liệu tham khảo](#)
14. [Phụ lục](#)

1. Giới thiệu chung

Trong thời đại công nghệ 4.0, Internet of Things (IoT) đang trở thành xu hướng phát triển mạnh mẽ trong nhiều lĩnh vực. Trong đó, nhà thông minh (smart home) là một trong những ứng dụng thiết thực nhất. Hệ thống chiếu sáng thông minh không chỉ mang tính thẩm mỹ mà còn giúp tiết kiệm năng lượng và nâng cao tiện nghi.

Trong bài tiểu luận này, em đề xuất và mô phỏng một hệ thống đèn LED RGB (WS2812) có khả năng điều khiển màu sắc và độ sáng thông qua giao diện web hoặc bằng giọng nói thông qua Google Assistant. Đề tài nhằm mục đích nâng cao tính tiện lợi, tiết kiệm điện năng và trải nghiệm người dùng trong không gian sống hiện đại.

2. Mục tiêu đề tài

- Thiết kế hệ thống đèn LED RGB thông minh với khả năng điều chỉnh linh hoạt.
- Cho phép người dùng điều khiển màu sắc và độ sáng thông qua nhiều phương thức: web, app, giọng nói.
- Mô phỏng hệ thống bằng nền tảng Wokwi để kiểm chứng nguyên lý hoạt động.
- Tích hợp các nền tảng IoT như Blynk, Google Assistant, ThingSpeak.
- Phân tích dữ liệu cảm biến (ánh sáng, nhiệt độ) để điều khiển LED theo ngữ cảnh thực tế.

3. Tổng quan công nghệ

3.1. Internet of Things (IoT): IoT là mạng lưới các thiết bị vật lý được kết nối internet và có khả năng thu thập, truyền tải dữ liệu. Trong lĩnh vực chiếu sáng, IoT giúp tăng hiệu quả năng lượng và cá nhân hóa trải nghiệm người dùng.

Thuật ngữ IoT hay Internet vạn vật đề cập đến mạng lưới tập hợp các thiết bị thông minh và công nghệ tạo điều kiện thuận lợi cho hoạt động

giao tiếp giữa thiết bị và đám mây cũng như giữa các thiết bị với nhau. Nhờ sự ra đời của chip máy tính giá rẻ và công nghệ viễn thông băng thông cao, ngày nay, chúng ta có hàng tỷ thiết bị được kết nối với internet. Điều này nghĩa là các thiết bị hàng ngày như bàn chải đánh răng, máy hút bụi, ô tô và máy móc có thể sử dụng cảm biến để thu thập dữ liệu và phản hồi lại người dùng một cách thông minh.

3.2. ESP32: ESP32 là một dòng chip vi điều khiển được phát triển bởi Espressif, với nhiều đặc điểm ưu việt:

- **Giá rẻ:** So với các dòng vi điều khiển khác, ESP32 có giá thành phải chăng hơn rất nhiều, giúp tất cả những ai đam mê công nghệ có thể dễ dàng tiếp cận nó
- **Lượng điện tiêu thụ thấp:** So với các chip điều khiển khác, ESP32 tiêu thụ rất ít năng lượng. Dòng chip này cũng hỗ trợ các trạng thái tiết kiệm năng lượng như Deep Sleep để tiết kiệm điện
- **Có thể kết nối Wi-Fi:** Bạn có thể dễ dàng kết nối ESP32 với mạng Wi-Fi để truy cập vào Internet (chế độ trạm – Station mode) hoặc tạo một mạng WiFi cho riêng nó (chế độ điểm truy cập – Access point) để các thiết bị khác có thể kết nối với nó. Chế độ Access point thường được dùng trong các dự án IoT hoặc tự động hóa trong Smart Home, trong đó bạn có thể cho phép nhiều thiết bị liên lạc và trao đổi thông tin với nhau thông qua WiFi của chúng
- **Hỗ trợ Bluetooth:** ESP32 hỗ trợ cả 2 chế độ: Bluetooth Classic và Bluetooth Low Energy (BLE) – một công cụ rất hữu ích cho những ứng dụng IoT
- **Lỗi kép:** Đa số các dòng chip ESP32 hiện nay đều có lỗi kép, chúng đi kèm với 2 bộ vi xử lý Xtensa 32-bit LX6: lỗi 0 và lỗi 1
- **Đa dạng thiết bị ngoại vi:** ESP32 hỗ trợ nhiều loại thiết bị ngoại vi đầu vào (đọc dữ liệu từ bên ngoài) và đầu ra (gửi lệnh/tín hiệu ra bên ngoài) như cảm ứng điện dung, I2C, DAC, PWM, UART, SPI,... để bạn tự do làm các dự án điện tử mà mình thích
- **Tương thích với Arduino và MicroPython:** ESP32 có thể được lập trình bằng ngôn ngữ lập trình phổ biến Arduino và MicroPython (phiên bản rút gọn của Python 3, phù hợp cho các bộ vi điều khiển và hệ thống nhúng)

3.3. LED WS2812: WS2812 là dải LED RGB thông minh có khả năng điều khiển từng bóng riêng biệt thông qua một dây tín hiệu duy nhất. LED này hỗ trợ 16 triệu màu và hiệu ứng chuyển màu phong phú.

3.4. Wokwi: Wokwi là nền tảng mô phỏng phần cứng miễn phí, cho phép kiểm tra hoạt động của ESP32, cảm biến và các module khác trước khi triển khai thực tế.

3.5. Blynk: Blynk là một nền tảng IoT cung cấp các công cụ để kết nối, quản lý và điều khiển các thiết bị IoT từ xa thông qua mạng Internet. Điểm nổi bật của Blynk là sự dễ dàng và nhanh chóng trong việc tạo và quản lý các ứng dụng IoT, phù hợp cho cả những người mới bắt đầu và những nhà phát triển chuyên nghiệp.

Blynk cung cấp ứng dụng di động và API cho phép người dùng kết nối và điều khiển các thiết bị IoT bằng cách sử dụng các cảm biến và các tín hiệu đầu vào. Giao diện người dùng được tùy chỉnh linh hoạt để điều khiển thiết bị IoT theo cách tùy chỉnh và tạo ra các hành động và tương tác phức tạp thông qua mã code.

App Blynk hỗ trợ các nền tảng phần cứng, bao gồm Arduino, Raspberry Pi, ESP8266 và nhiều nền tảng khác nữa. Điều này cho phép người dùng dễ dàng tích hợp các thiết bị IoT vào các dự án của mình mà không cần có nhiều kinh nghiệm lập trình.

3.6. Google Assistant + IFTTT: IFTTT hay If This Then That (nếu điều này xảy ra thì sau đó điều kia xảy ra) là một nền tảng dịch vụ tự động hoá theo dạng lệnh tùy chỉnh: nếu thỏa mãn điều kiện This thì sẽ thực hiện hành động That. Mỗi lệnh tùy chỉnh IFTTT đó được gọi là một applet. Người dùng có thể tạo ra nhiều applet khác nhau tùy thích để có nhiều lệnh tùy chỉnh tự động hoá Google Assistant theo mong muốn của mình.

Theo Wikipedia, IFTTT là dịch vụ trung gian phát hành bởi IFTTT Inc. từ 07 tháng 09 năm 2011 dưới dạng nền tảng dựa trên web. IFTTT có thể sử dụng để kết nối với rất nhiều nền tảng khác nhau như: Google Assistant, Facebook, Twitter,.. Ứng dụng rất hữu ích vào nhiều lĩnh vực như Marketing, Tự động hoá,.. trong đó dùng để tùy chỉnh Google Assistant là một ứng dụng rất được yêu thích.

Ví dụ khi chưa dùng IFTTT, bạn muốn bật điều hoà sẽ cần dùng câu lệnh:

"Ok Google, turn on the light" hay "Ok Google, bật đèn" và luôn cần như vậy. Nhưng khi dùng IFTTT bạn có thể tùy chỉnh thành "Ok Google, the

house is too dark" hay "Ok Google, nhà tối quá" và sau đó Google Assistant phản hồi bằng cách bật đèn kèm câu trả lời hóm hỉnh "Được thôi, người chủ tuyệt vời của tôi". Thật sự IFTTT đã làm việc sử dụng trợ lý thông minh trở nên thú vị và đa dạng hơn.

3.7. ThingSpeak: một nền tảng IoT (Internet of Things) dựa trên đám mây cho phép người dùng thu thập, lưu trữ, phân tích và trực quan hóa dữ liệu được tạo ra từ các thiết bị hoặc cảm biến kết nối mạng. Được phát triển bởi MathWorks, công ty đứng sau MATLAB, ThingSpeak tận dụng sức mạnh của việc tích hợp dữ liệu và tính toán kỹ thuật để cung cấp một giải pháp toàn diện cho nhu cầu phân tích dữ liệu trong thời gian thực.

4. Thiết bị sử dụng

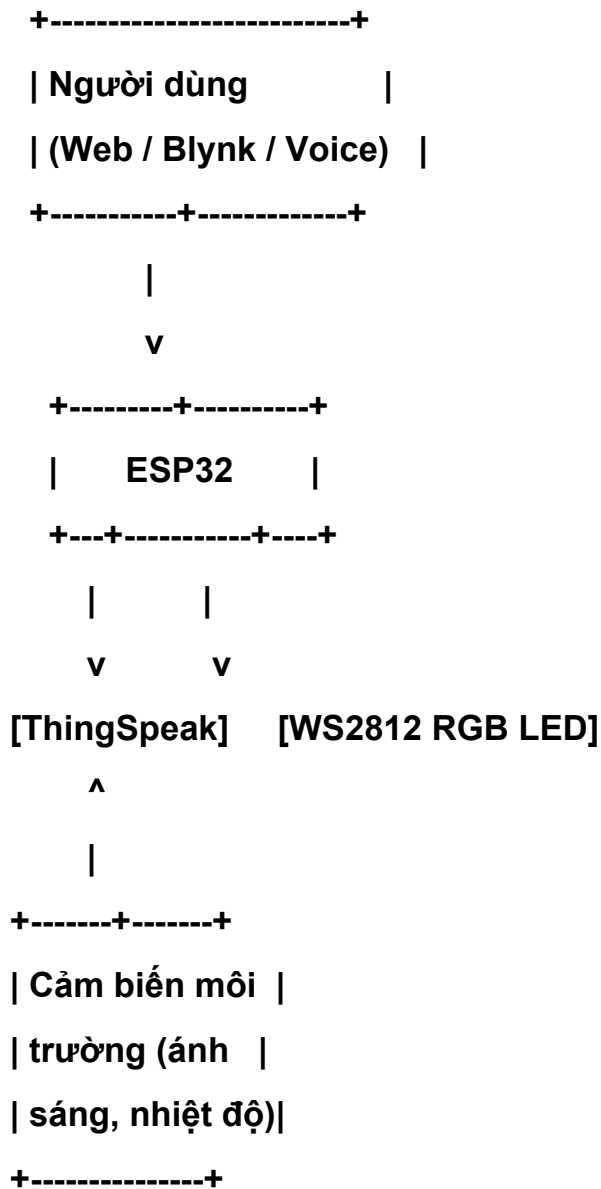
- ESP32 DevKit v1
- Dải LED WS2812 (NeoPixel)
- Cảm biến ánh sáng (light sensor) hoặc giá trị giả lập random
- Cảm biến DHT22 (nhiệt độ, độ ẩm) hoặc random nếu không có sẵn trong Wokwi
- Máy chủ Web (ESP32 WebServer)
- Ứng dụng di động Blynk hoặc Google Assistant (qua IFTTT)

5. Kiến trúc hệ thống và sơ đồ khối

5.1. Kiến trúc tổng thể hệ thống

Hệ thống được xây dựng dựa trên mô hình Client – Server trong IoT, trong đó ESP32 đóng vai trò là trung tâm điều khiển (Server), nhận dữ liệu từ các thiết bị cảm biến và xử lý yêu cầu điều khiển từ người dùng thông qua các nền tảng như Web Server, Blynk hoặc Google Assistant (thông qua IFTTT). Kết quả xử lý được phản hồi bằng cách thay đổi trạng thái của LED RGB (WS2812) và/hoặc gửi dữ liệu lên nền tảng cloud như ThingSpeak.

5.2. Sơ đồ luồng hoạt động

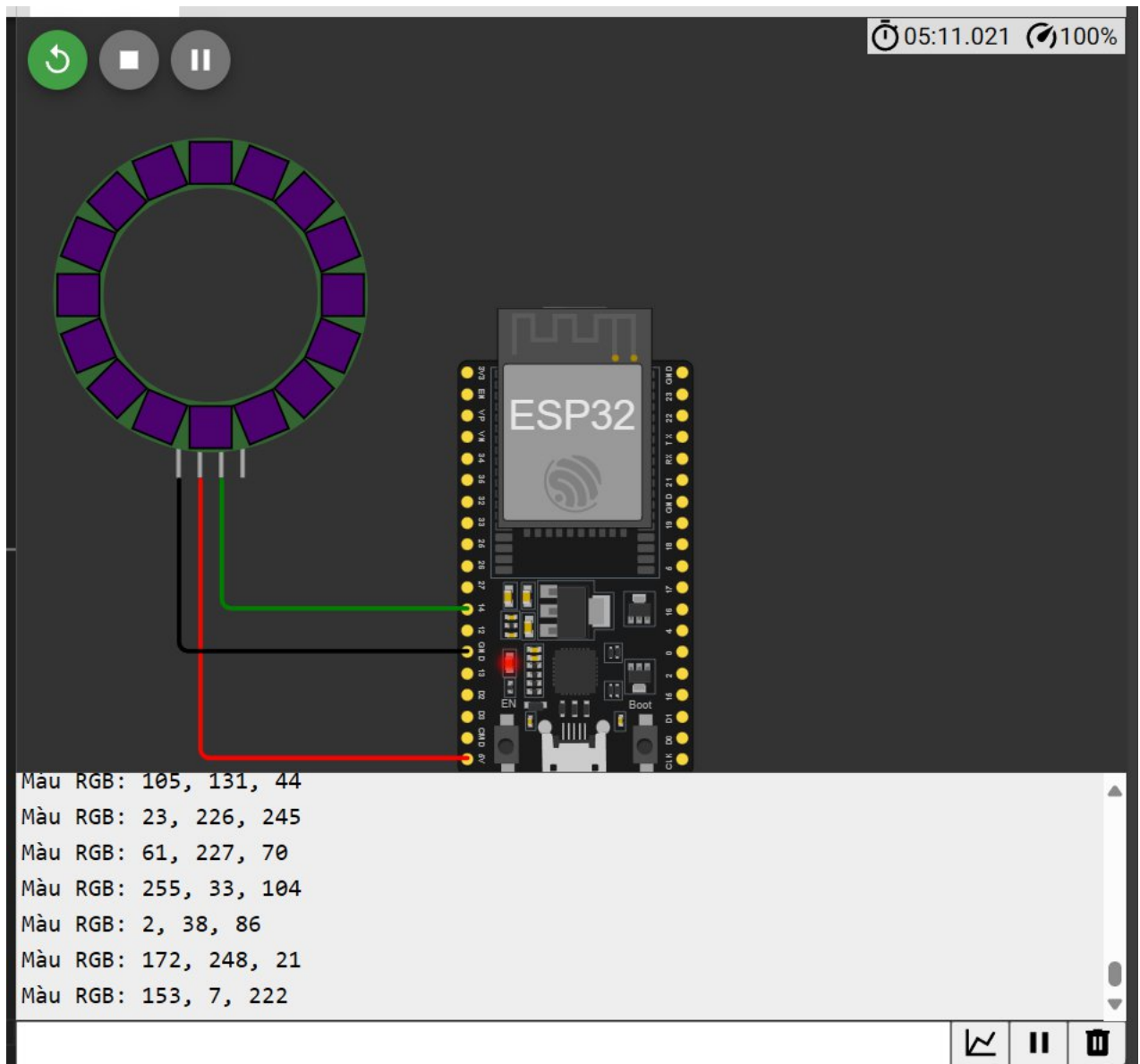


5.3. Mô tả các thành phần trong sơ đồ

- **Người dùng (User Interface):** Tương tác với hệ thống qua 3 cách:
 - Trình duyệt web (Web Server nội bộ của ESP32)
 - Ứng dụng Blynk trên điện thoại
 - Giọng nói qua Google Assistant (kết hợp với IFTTT)
- **ESP32:** Là bộ não trung tâm của hệ thống:

- Thu thập dữ liệu từ các cảm biến môi trường (ánh sáng, nhiệt độ)
- Xử lý yêu cầu từ người dùng và điều khiển LED
- Gửi dữ liệu lên cloud (ThingSpeak) để lưu trữ và phân tích
- **Cảm biến môi trường:**
 - Cảm biến ánh sáng: xác định độ sáng môi trường xung quanh
 - Cảm biến nhiệt độ, độ ẩm: đo điều kiện không khí
 - Nếu không có cảm biến thật trên Wokwi, sử dụng giá trị ngẫu nhiên (random)
- **LED RGB WS2812:**
 - Nhận tín hiệu từ ESP32 để thay đổi màu sắc và độ sáng
 - Có thể thể hiện trạng thái môi trường (ví dụ màu đỏ nếu nóng, xanh nếu mát...)
- **ThingSpeak Cloud:**
 - Lưu trữ dữ liệu cảm biến từ ESP32
 - Hiển thị biểu đồ nhiệt độ, ánh sáng
 - Có thể lập trình logic để phản hồi ngược về hệ thống điều khiển

6. Mô phỏng trên Wokwi



6.1. Kết nối linh kiện

Trong mô hình mô phỏng trên nền tảng Wokwi, các thiết bị được kết nối như sau:

- ESP32 DevKit v1 là trung tâm điều khiển chính.
- Dải LED WS2812 (NeoPixel):
 - Chân tín hiệu kết nối vào GPIO18 của ESP32
 - Chân VCC và GND lần lượt nối nguồn 5V và GND của ESP32
- Cảm biến ánh sáng (photoresistor):
 - Một đầu nối với GPIO34 (analog input) và điện trở pull-down nối GND
- Cảm biến nhiệt độ và độ ẩm DHT22:

- Chân Data nối với GPIO15
- Chân VCC và GND kết nối tương ứng với ESP32

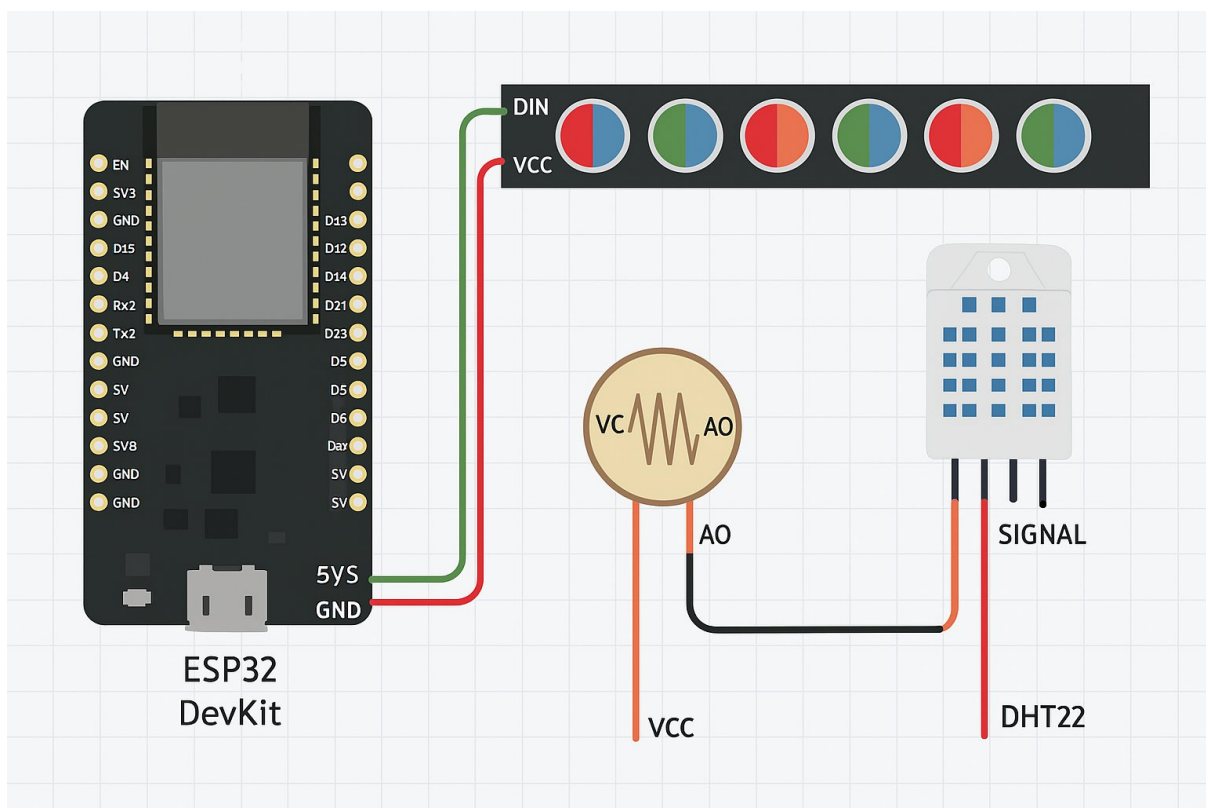
Nếu cảm biến không có sẵn trong Wokwi, có thể thay thế bằng giá trị giả lập (random) thông qua đoạn mã giả lập hoặc điều chỉnh trong giao diện.

6.2. Giao diện mô phỏng trong Wokwi

Wokwi cho phép người dùng trực quan kết nối các linh kiện, thử thay đổi điều kiện môi trường, quan sát giá trị cảm biến và hành vi của LED RGB trên mô hình ảo. Qua đó, người dùng có thể:

- Điều chỉnh ánh sáng và nhiệt độ để xem LED phản ứng như thế nào
- Mô phỏng dữ liệu được gửi về ThingSpeak
- Kiểm tra giao tiếp giữa các module

6.3. Sơ đồ mô phỏng phần cứng (Wokwi)



Đây là hình ảnh mô phỏng phần cứng được xây dựng trên nền tảng Wokwi, thể hiện rõ sự kết nối giữa ESP32 với dải LED WS2812 và các cảm biến tương ứng.

7. Điều khiển hệ thống

7.1. Qua WebServer (ESP32):

- ESP32 tạo một web server nội bộ, truy cập qua IP trong mạng WiFi.
- Giao diện có Color Picker và Slider để điều chỉnh đèn.
- Mỗi thay đổi gửi dữ liệu qua HTTP GET request.

7.2. Qua giọng nói (Google Assistant + IFTTT):

- IFTTT kết nối Google Assistant và webhook gửi đến ESP32
- Ví dụ: “Hey Google, set LED to red” → IFTTT gửi request: `/led?color=255,0,0`

7.3. Qua app Blynk:

- Giao diện trên điện thoại gồm:
 - Slider điều chỉnh độ sáng
 - Color picker điều chỉnh màu
- ESP32 kết nối cloud server của Blynk qua mã token API

7.4 Điều khiển màu sắc và độ sáng của đèn LED WS2812 qua ứng dụng web hoặc giọng nói (Google Assistant)

Trong hệ thống đề xuất, người dùng có thể điều khiển dải đèn LED WS2812 theo hai cách chính: (1) thông qua giao diện web được lưu trữ trên ESP32 và (2) thông qua giọng nói với sự hỗ trợ của Google Assistant, kết hợp IFTTT và Webhook. Mỗi phương pháp có cơ chế hoạt động riêng biệt, phù hợp với các nhu cầu sử dụng khác nhau.

a. Điều khiển thông qua giao diện web

Nguyên lý hoạt động:

ESP32 sẽ hoạt động như một máy chủ web mini (web server). Sau khi được kết nối vào mạng Wi-Fi, ESP32 sẽ lắng nghe các yêu cầu từ trình duyệt. Khi người dùng truy cập vào địa chỉ IP của ESP32, một giao diện HTML sẽ được hiển thị với các thành phần:

- Color Picker: Cho phép chọn màu sắc RGB theo sở thích.
- Slider (thanh kéo): Dùng để điều chỉnh độ sáng của đèn.

Khi người dùng tương tác:

- Màu sắc và độ sáng sẽ được gửi dưới dạng HTTP request (GET hoặc POST) tới ESP32.
- ESP32 sẽ xử lý dữ liệu nhận được và gửi tín hiệu điều khiển tới LED WS2812 để thay đổi theo yêu cầu.

Ưu điểm:

- Không cần cài ứng dụng riêng.
- Tương thích mọi thiết bị có trình duyệt: điện thoại, máy tính bảng, laptop.
- Phản hồi thời gian thực, độ trễ thấp.

Hạn chế:

- Chỉ hoạt động trong cùng mạng nội bộ (trừ khi cấu hình truy cập từ xa).
- Cần biết địa chỉ IP của ESP32 để truy cập.

Hình ảnh minh họa giao diện (có thể vẽ thêm):

markdown

Sao chépChỉnh sửa

```
-----  
| 🎨 Chọn màu: [ColorPicker] |  
| 🌞 Độ sáng: [Slider] |  
|                               |  
| [ Cập nhật LED ] |  
-----
```

b. Điều khiển thông qua giọng nói (Google Assistant)

Để điều khiển LED qua giọng nói, hệ thống sử dụng kết hợp Google Assistant, IFTTT và Webhook.

Nguyên lý hoạt động:

1. Người dùng nói lệnh, ví dụ:
 - “Hey Google, turn LED red”
 - “Hey Google, set LED brightness to 30 percent”

2. Google Assistant nhận lệnh → Kích hoạt Applet IFTTT đã cấu hình sẵn.
3. IFTTT thực hiện hành động “Webhook” → Gửi HTTP GET request đến địa chỉ IP của ESP32 (có thể bao gồm tham số như ?color=red&brightness=30).
4. ESP32 nhận yêu cầu, phân tích lệnh và thay đổi màu sắc hoặc độ sáng LED WS2812 theo nội dung yêu cầu.

Ưu điểm:

- Điều khiển hoàn toàn rảnh tay, tiện lợi trong sinh hoạt hàng ngày.
- Tăng tính hiện đại, phù hợp xu hướng nhà thông minh.

Hạn chế:

- Phụ thuộc vào dịch vụ IFTTT (miễn phí có giới hạn).
- ESP32 cần có địa chỉ IP tĩnh hoặc sử dụng dịch vụ DDNS để định tuyến.
- Tốc độ phản hồi chậm hơn một chút do qua trung gian IFTTT.

Cấu hình cơ bản IFTTT:

- IF: Google Assistant nhận lệnh dạng text (custom phrase).
- THEN: Webhooks → URL (ESP32 IP) → Gửi màu/độ sáng.

c. So sánh hai phương pháp

Tiêu chí	Giao diện web	Giọng nói (Google Assistant)
Thiết bị điều khiển	Trình duyệt	Thiết bị hỗ trợ Google Assistant
Cách thức kết nối	Trực tiếp qua Wi-Fi	Qua Internet thông qua IFTTT
Thời gian phản hồi	Nhanh (gần như tức thì)	Có độ trễ nhỏ qua trung gian
Độ thân thiện người dùng	Cần thao tác tay	Rảnh tay, hiện đại
Độ phức tạp triển khai	Trung bình	Cao hơn (cấu hình IFTTT, Webhook)

d. Đánh giá tổng quan

Việc triển khai đồng thời hai phương thức điều khiển mang lại tính linh hoạt cao cho hệ thống, phù hợp với nhu cầu sử dụng trong nhiều bối cảnh khác nhau:

- Người dùng có thể bật đèn nhanh chóng bằng giọng nói khi đang bận tay (như đang nấu ăn, đọc sách).
- Hoặc có thể tùy chỉnh chính xác màu sắc/độ sáng bằng giao diện web khi cần thiết lập cụ thể.

Đây là ví dụ điển hình cho việc kết hợp nhiều công nghệ IoT hiện đại nhằm mang lại trải nghiệm người dùng tiện lợi, linh hoạt và thông minh hơn.

8. Phân tích dữ liệu cảm biến qua ThingSpeak

Hệ thống không chỉ hỗ trợ điều khiển đèn LED RGB mà còn có khả năng thu thập, gửi và phân tích dữ liệu cảm biến môi trường (ánh sáng và nhiệt độ) theo thời gian thực thông qua nền tảng ThingSpeak.

8.1. Gửi dữ liệu cảm biến

ESP32 thu thập dữ liệu từ:

- Cảm biến ánh sáng (hoặc giá trị giả lập)
- Cảm biến DHT22 (nhiệt độ và độ ẩm)

Cứ mỗi 10 giây, dữ liệu được gửi lên nền tảng ThingSpeak bằng API Key thông qua HTTP POST.

8.2. Hiển thị dữ liệu theo thời gian thực

ThingSpeak cung cấp công cụ trực quan hóa:

- Biểu đồ đường (Line Chart) theo thời gian
- Mỗi loại cảm biến sẽ hiển thị ở một channel riêng

Điều này giúp người dùng dễ dàng theo dõi xu hướng, xác định điều kiện môi trường hiện tại và trong quá khứ.

8.3. Ứng dụng logic điều khiển tự động

Từ dữ liệu môi trường, có thể thiết lập các điều kiện điều khiển LED tự động như:

- Nếu ánh sáng môi trường < 200 lux \rightarrow Tự động bật LED với độ sáng cao
- Nếu nhiệt độ $> 30^{\circ}\text{C}$ \rightarrow Chuyển màu LED sang màu xanh dương để tạo cảm giác mát
- Nếu nhiệt độ $< 20^{\circ}\text{C}$ \rightarrow Đổi sang màu vàng cam (ấm áp)

8.4. Ưu điểm khi tích hợp ThingSpeak

- Miễn phí, dễ sử dụng, không cần server riêng
- Cho phép chia sẻ công khai hoặc riêng tư
- Hỗ trợ MATLAB Analytics nếu cần xử lý dữ liệu nâng cao

9. Thuật toán hoạt động hệ thống

Để đảm bảo hệ thống hoạt động ổn định, có khả năng phản hồi với nhiều phương thức điều khiển khác nhau, thuật toán chính của hệ thống sẽ được tổ chức theo các bước sau:

Bước 1: Khởi động hệ thống

- ESP32 khởi động và kết nối vào mạng Wi-Fi nội bộ.
- Kết nối đến các nền tảng cần thiết như:
 - Blynk Cloud (qua Auth Token)
 - ThingSpeak API
 - Webhook từ IFTTT (nếu có lệnh từ Google Assistant)

Bước 2: Khởi tạo cảm biến và LED

- Khởi tạo LED WS2812, thiết lập số lượng LED và chân điều khiển (GPIO18).
- Đọc dữ liệu cảm biến môi trường:
 - Cảm biến nhiệt độ/độ ẩm DHT22
 - Cảm biến ánh sáng hoặc dữ liệu giả lập

Bước 3: Nhận lệnh điều khiển từ người dùng

ESP32 sẽ kiểm tra đồng thời các kênh điều khiển:

1. WebServer nội bộ:

- Khi người dùng truy cập địa chỉ IP, ESP32 trả về giao diện web có thanh chọn màu (Color Picker) và thanh trượt độ sáng (Slider).
- Lệnh sẽ được gửi qua HTTP GET như:

bash

/led?color=255,0,0&brightness=150

2. Blynk App:

- Thông qua Virtual Pins (V0 - màu sắc, V1 - độ sáng).
- Khi người dùng thay đổi màu/sáng, Blynk sẽ gửi giá trị về ESP32.

3. Google Assistant + IFTTT:

- Khi người dùng nói "Hey Google, turn LED red", IFTTT sẽ gọi Webhook gửi request đến ESP32.
- Dữ liệu được phân tích và gán màu tương ứng.

Bước 4: Cập nhật trạng thái LED

- Sau khi nhận lệnh điều khiển, ESP32 sẽ:
 - Giải mã chuỗi màu RGB
 - Điều chỉnh độ sáng theo mức quy định
 - Cập nhật LED WS2812 tương ứng

Bước 5: Gửi dữ liệu cảm biến lên ThingSpeak

- Sau mỗi chu kỳ (ví dụ: 10 giây), dữ liệu môi trường được gửi lên kênh ThingSpeak.
- Điều này cho phép theo dõi nhiệt độ và ánh sáng liên tục.

Bước 6: Điều chỉnh thông minh (nếu cần)

- Nếu hệ thống được kích hoạt chế độ “Tự động”:
 - Dựa vào cảm biến ánh sáng → Điều chỉnh độ sáng LED.
 - Dựa vào nhiệt độ → Chuyển màu sắc tạo cảm giác phù hợp (lạnh – xanh, nóng – đỏ).

Bước 7: Lặp lại vòng lặp

- Quay lại bước 2 → Tiếp tục cập nhật trạng thái hệ thống và phản hồi đầu vào từ người dùng hoặc môi trường.

10. Tính năng nâng cao và mở rộng

Hệ thống đèn LED thông minh với ESP32 có thể được nâng cấp để đáp ứng các nhu cầu thực tế đa dạng và phức tạp hơn trong mô hình nhà thông minh. Dưới đây là một số tính năng mở rộng tiêu biểu:

10.1. Tự động điều chỉnh theo điều kiện môi trường

- Ngưỡng cảnh: Khi trời tối, đèn tự động bật và điều chỉnh độ sáng phù hợp.
- Giải pháp kỹ thuật: Đọc giá trị cảm biến ánh sáng, nếu dưới ngưỡng quy định → bật LED và tăng độ sáng.
- Ưu điểm: Tối ưu hóa điện năng, tăng tính tự động hóa.

10.2. Điều chỉnh màu sắc theo nhiệt độ

- Ngưỡng cảnh: Khi nhiệt độ phòng vượt 30°C → LED chuyển sang màu xanh dương.
- Mục tiêu: Tạo cảm giác mát mẻ thông qua hiệu ứng ánh sáng.
- Thực hiện: Đọc cảm biến DHT22 → nếu $t > 30$, đổi màu LED RGB sang #00BFFF.

10.3. Nhận diện giọng nói offline

- Giải pháp: Sử dụng thư viện nhận diện giọng nói như ESP-Skainet (của Espressif).

- **Ưu điểm:**
 - Không cần kết nối internet.
 - Bảo mật và tốc độ phản hồi nhanh hơn.
- **Khó khăn:** Cần đủ bộ nhớ flash và tùy chỉnh mô hình nhận diện.

10.4. Cảm biến chuyển động PIR

- **Ứng dụng:** Tự động bật đèn khi phát hiện có người trong khu vực.
- **Kịch bản thực tế:** Người bước vào phòng → đèn bật lên trong 10 giây.
- **Kết nối:** PIR Sensor → ESP32 GPIO (ví dụ: GPIO14)

10.5. Ghi lại lịch sử và học thói quen người dùng

- **Phân tích:** Ghi nhận thời gian, màu đèn thường được chọn, mức sáng phổ biến.
- **Lưu trữ:** Dữ liệu có thể lưu vào thẻ nhớ SD hoặc gửi lên Firebase, Google Sheets,...
- **Mục tiêu:** Tự động điều chỉnh đèn theo thói quen (ví dụ: buổi tối thường chọn màu vàng → gợi ý màu vàng khi trời tối).

10.6. Giao diện Web nâng cấp

- **Chức năng thêm:** Hiển thị biểu đồ nhiệt độ, ánh sáng, nhật ký điều khiển.
- **Công nghệ dùng:** HTML, CSS, JavaScript, Chart.js hoặc Google Charts.
- **Ví dụ:** Thêm nút bật/tắt đèn theo thời gian định trước (scheduler), chọn hiệu ứng nhấp nháy.

10.7. Đồng bộ với hệ sinh thái nhà thông minh

- **Tương thích:** Tích hợp vào Home Assistant, MQTT broker,...
- **Ưu điểm:** Điều khiển qua nhiều nền tảng, đồng bộ các thiết bị khác như quạt, điều hòa, rèm cửa.

11. Đánh giá hệ thống

Việc phát triển hệ thống đèn LED thông minh RGB sử dụng ESP32 kết hợp với các nền tảng IoT như WebServer, Blynk, Google Assistant và ThingSpeak đã mang lại một mô hình hoàn chỉnh, có thể ứng dụng trong nhiều ngữ cảnh thực tế. Dưới đây là một số đánh giá chi tiết:

11.1. Ưu điểm

Chi phí thấp, dễ triển khai:

- **ESP32 và LED WS2812 là các linh kiện phổ biến, giá rẻ, dễ mua.**
- **Phù hợp cho sinh viên, nhà nghiên cứu, hoặc các ứng dụng dân dụng nhỏ.**

Linh hoạt trong điều khiển:

- **Hỗ trợ đa nền tảng: có thể điều khiển bằng web, app, hoặc giọng nói.**
- **Cho phép người dùng tùy biến màu sắc, độ sáng dễ dàng.**

Khả năng mở rộng cao:

- **Có thể kết nối thêm nhiều cảm biến khác (PIR, khí gas, ánh sáng...).**
- **Dễ tích hợp với các nền tảng nhà thông minh lớn như Home Assistant, MQTT.**

Giao diện trực quan, thân thiện:

- **Giao diện Web và App dễ thao tác.**
- **Hỗ trợ giọng nói (Google Assistant) mang lại trải nghiệm hiện đại.**

11.2. Hạn chế

Phụ thuộc vào kết nối mạng:

- **Việc điều khiển từ xa qua Blynk, Google Assistant cần mạng WiFi ổn định.**
- **Nếu mất mạng, nhiều chức năng có thể bị tạm thời vô hiệu.**

Độ trễ điều khiển bằng giọng nói:

- **Google Assistant + IFTTT có thể chậm hơn 1-2 giây vì qua nhiều tầng dịch vụ.**

Giới hạn tài nguyên phần cứng:

- ESP32 có bộ nhớ hạn chế, nếu thêm nhiều tính năng nâng cao cần tối ưu mã nguồn hoặc dùng thêm vi điều khiển phụ trợ.

Bảo mật chưa cao nếu triển khai thực tế:

- Các webhook hoặc HTTP request không được mã hóa (nếu không cẩn thận) có thể bị khai thác.
- Cần thiết lập thêm xác thực, HTTPS khi triển khai ra bên ngoài.

11.3. Khả năng ứng dụng

- Mô hình demo trong lớp học IoT hoặc hội thảo công nghệ.
- Ứng dụng trong phòng ngủ, quán cà phê, khu nghỉ dưỡng thông minh.
- Nền tảng mẫu để phát triển hệ thống chiếu sáng tự động tại nhà.
- Khởi đầu cho các dự án nhà thông minh đa chức năng hơn.

12. Kết luận

Trong kỷ nguyên của công nghệ kết nối, việc áp dụng Internet of Things (IoT) vào các ứng dụng đời sống hàng ngày không chỉ giúp tăng cường tiện ích mà còn mang lại trải nghiệm mới mẻ, thông minh và tiết kiệm năng lượng hơn.

Đề tài “Phát triển hệ thống đèn LED thông minh RGB với ESP32” là một ví dụ thực tiễn cho việc xây dựng một hệ thống chiếu sáng thông minh, có khả năng điều khiển từ xa qua Web App, ứng dụng di động Blynk, và giọng nói với Google Assistant, đồng thời hỗ trợ việc thu thập, phân tích dữ liệu môi trường bằng ThingSpeak.

Thông qua việc mô phỏng bằng nền tảng Wokwi, em đã:

- Xây dựng sơ đồ mạch đầy đủ và hợp lý.
- Thiết kế giao diện điều khiển thân thiện, dễ sử dụng.
- Tích hợp hệ thống giám sát môi trường theo thời gian thực.

Đề tài cũng mở ra nhiều hướng phát triển trong tương lai như:

- Tự động hóa hoàn toàn theo điều kiện môi trường.
- Tích hợp với trợ lý ảo offline, tăng tính bảo mật.

- **Phát triển phiên bản thực tế có tính ổn định cao hơn.**

Từ đó có thể thấy, đây là một mô hình IoT đơn giản nhưng rất hiệu quả, hoàn toàn có thể áp dụng trong các hệ thống nhà thông minh hiện đại.

13. Tài liệu tham khảo

- 1. <https://wokwi.com>**
- 2. <https://randomnerdtutorials.com>**
- 3. <https://ifttt.com>**
- 4. <https://thingspeak.com>**
- 5. <https://docs.blynk.io>**
- 6. <https://www.espressif.com/en/products/socs/esp32>**
- 7. <https://github.com/Wokwi/diagram-json>**
- 8. <https://www.arduino.cc>**
- 9. Tài liệu học phần Phát triển ứng dụng IoT**

14. Phụ lục

Phụ lục A: Sơ đồ mạch kết nối trong Wokwi

Phụ lục B: Giao diện Web Server

Phụ lục C: Giao diện App Blynk

Phụ lục D: Mẫu applet IFTTT sử dụng Google Assistant

Phụ lục E: Biểu đồ dữ liệu từ ThingSpeak