ĐẠI HỌC HUẾ

TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA CÔNG NGHỆ THÔNG TIN

BÀI TIỂU LUẬN

Đề tài:

TẠO MÁY CHỦ WEB CỤC BỘ VỚI ESP32 ĐỂ QUẢN LÝ THIẾT BỊ IOT

Khóa: K45

Giáo viên hướng dẫn: Võ Việt Dũng

Huế, tháng 04 năm 2024

LỜI CÁM 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. Với sự tận tâm hướng dẫn và những kiến thức quý báu mà thầy đã truyền đạt, em không chỉ hoàn thành tiểu luận này mà còn được khơi dậy niềm đam mê khám phá sâu hơn về lĩnh vực Internet of Things. Sự tận tình và định hướng của thầy chính là nguồn động lực lớn lao giúp nhóm chúng em vượt qua những thử thách trong suốt quá trình học tập và nghiên cứu.

Em cũng xin bày tỏ lòng biết ơn chân thành đến các bạn học cùng nhóm 09 – những người đồng hành tuyệt vời. Sự đóng góp ý kiến, chia sẻ kinh nghiệm và tinh thần hợp tác của các bạn đã tiếp thêm sức mạnh để nhóm không ngừng nỗ lực, từng bước hoàn thiện đề tài này.

Dù đã dồn hết tâm huyết thực hiện, tiểu luận của em chắc chắn vẫn còn những hạn chế và thiếu sót. Em rất mong nhận được sự lượng thứ cùng những ý kiến đóng góp quý giá từ thầy cô, đặc biệt là thầy Võ Việt Dũng, cũng như các bạn để đề tài ngày càng hoàn thiện hơn.

Một lần nữa, em xin gửi lời cảm ơn trân trọng nhất đến thầy Võ Việt Dũng và tất cả mọi người đã đồng hành cùng em trong hành trình này!

Em xin chân thành cảm ơn!

DANH MỤC CÁC TỪ VIẾT TẮT

Từ viết tắt	Ý nghĩa đầy đủ
IoT	Internet of Things
ESP32	Espressif Systems Processor 32
SPIFFS	SPI Flash File System
Wi-Fi	Wireless Fidelity
HTTP	HyperText Transfer Protocol
HTML	HyperText Markup Language
CSS	Cascading Style Sheets
JSON	JavaScript Object Notation
GPIO	General Purpose Input/Output
LAN	Local Area Network

MỤC LỤC

CHƯƠNG 1: TỔNG QUAN 5
1.1 Giới thiệu tổng quan và mục tiêu của đề tài. 5
1.1.1 Giới thiệu tổng quan về đề tài 5
1.1.2 Mục tiêu của đề tài 6
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 7
2.1 Tổng quan về ESP32 7
2.2 Khái niệm về máy chủ web cục bộ 8
2.3 Giới thiệu về SPIFFS 9
2.4 Web cục bộ trên ESP32 10
CHƯƠNG 3: TỔNG QUAN VỀ HỆ THỐNG 11
3.1 Tổng quan chi tiết về Hệ thống Web cục bộ trên ESP32 để Điều khiển Thiết bị IoT 11
3.2 Mục tiêu và ý nghĩa của hệ thống 11
3.3. Các thành phần chính của hệ thống 12
3.3.1 Client (Máy tính hoặc thiết bị người dùng) 12
3.3.2 Mạng (WiFi nội bộ) 12
3.3.3 ESP32 (Web Server) 13
3.3.4 Thiết bị IoT 13
CHƯƠNG 4: TRIỂN KHAI VÀ THIẾT KẾ HỆ THỐNG 14
4.1 Các bước triển khai 14
4.1.1 Chuẩn bị phần cứng 14
4.1.2 Chuẩn bị phần mềm 14
4.1.3 Các bước thực hiện 15
CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 22
5.1. Kết quả đạt được 22
5.2. Ý nghĩa của đề tài 22
5.3. Hạn chế 22
5.4. Hướng phát triển 22
5.5. Kết luận 23
TÀI LIỆU THAM KHẢO 23

CHƯƠNG 1: TỔNG QUAN

1.1 Giới thiệu tổng quan và mục tiêu của đề tài.

1.1.1 Giới thiệu tổng quan về đề tài

Trong bối cảnh công nghệ hiện đại, việc quản lý thiết bị IoT (Internet of Things) đã trở thành một nhu cầu thiết yếu, đặc biệt trong các ứng dụng thông minh và tự động hóa của cuộc sống hàng ngày. ESP32, một trong những vi mạch nổi bật hiện nay, mang lại khả năng kết nối linh hoạt với cả Wi-Fi và Bluetooth, cho phép người dùng triển khai các giải pháp IoT một cách đa dạng và hiệu quả. Việc tạo ra một máy chủ web cục bộ với ESP32 mở ra cơ hội mới trong việc kiểm soát và giám sát các thiết bị IoT từ xa, mà không cần phải phụ thuộc vào kết nối Internet liên tục. Điều này không chỉ giảm thiểu chi phí mà còn tăng cường tính bảo mật khi dữ liệu không cần phải truyền qua các mạng công cộng.

Máy chủ web cục bộ không chỉ giúp người dùng dễ dàng giao tiếp với các thiết bị trong mạng nội bộ, mà còn cung cấp an ninh cao trong việc phát triển và quản lý các ứng dụng. Thông qua việc xây dựng giao diện người dùng trên trình duyệt, các nhà phát triển có thể tạo ra các bảng điều khiển tùy chỉnh để theo dõi và điều khiển các thiết bị, hiện thực hóa mong muốn tích hợp công nghệ vào cuộc sống hàng ngày một cách mượt mà hơn. Hơn nữa, khả năng lập trình và cấu hình ESP32 thông qua môi trường phát triển như Arduino hay PlatformIO, mang lại sự tiện lợi và tiếp cận dễ dàng hơn cho những người mới bắt đầu, cũng như tạo nền tảng vững chắc cho các chuyên gia.

Trong phần này, chúng ta sẽ khám phá sâu hoạt động của ESP32 như là một máy chủ web cục bộ, các bước thiết lập ban đầu, và những điều cần lưu ý trong quá trình phát triển. Nội dung này không chỉ giúp giới thiệu khái quát về ý tưởng mà còn đặt nền tảng cho những phần tiếp theo, nơi chúng ta sẽ đi sâu vào từng chi tiết kỹ thuật và ứng dụng cụ thể của việc quản lý thiết bị IoT thông qua máy chủ web cục bộ. Sự kết hợp giữa phần cứng tiên tiến và công nghệ lập trình linh hoạt như ESP32 cho phép chúng ta xây dựng các giải pháp IoT sáng tạo, đáp ứng nhu cầu đa dạng từ cá nhân đến doanh nghiệp

Hình ảnh minh họa cho việc ứng dụng các thiết bị trong nhà.

1.1.2 Mục tiêu của đề tài

Tận dụng ESP32 để xây dựng một máy chủ web cục bộ nhằm quản lý tất cả các thiết bị IoT trong mạng nội bộ, bao gồm việc điều khiển, giám sát trạng thái và tối ưu hóa hoạt động của chúng. Thông qua giao diện web trực quan, người dùng có thể thao tác dễ dàng với các thiết bị mà không cần kết nối Internet, đảm bảo tính bảo mật và hiệu suất cao.

Đề tài không chỉ giới thiệu khái quát về ý tưởng mà còn đặt nền tảng cho các phần tiếp theo, nơi chúng ta sẽ phân tích chi tiết kỹ thuật, từ cấu hình phần cứng, lập trình phần mềm, đến triển khai thực tế. Sự kết hợp giữa ESP32 và công nghệ webserver hứa hẹn tạo ra một giải pháp IoT sáng tạo, đáp ứng nhu cầu quản lý thiết bị thông minh trong gia đình, văn phòng hoặc các không gian quy mô nhỏ.

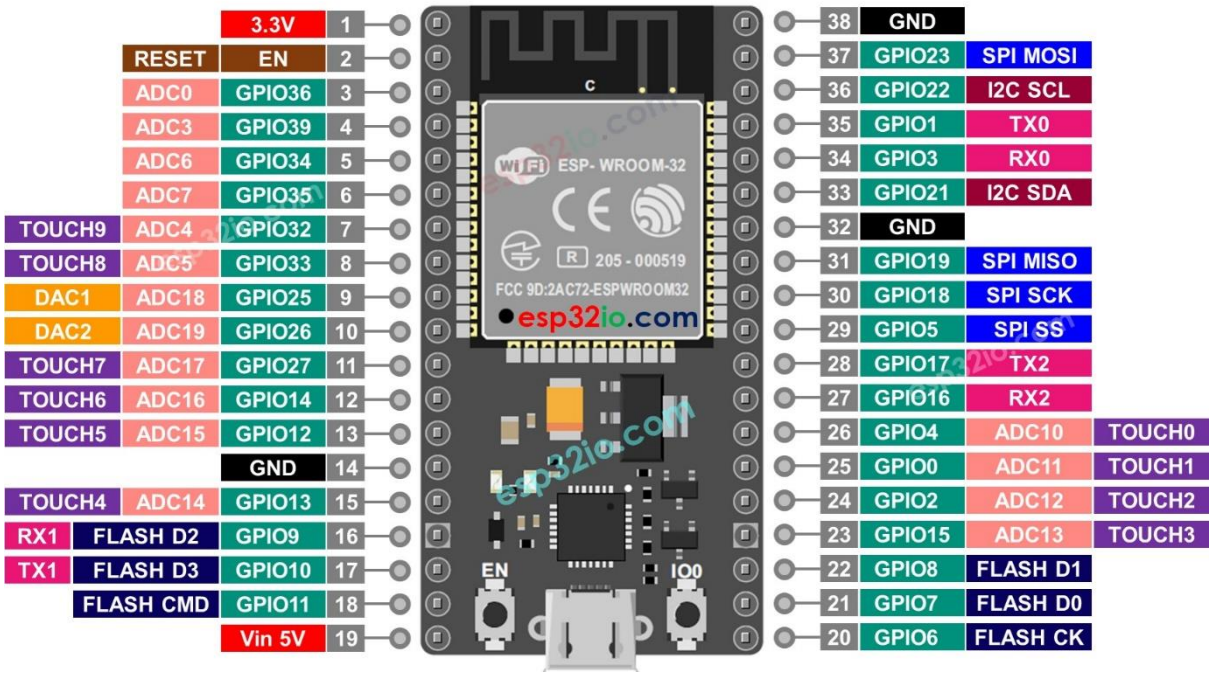
CHƯƠNG 2. CƠ SỞ LÝ THUYẾT

2.1 Tổng quan về ESP32

ESP32, một bộ vi điều khiển linh hoạt và mạnh mẽ được phát triển bởi Espressif Systems, là nền tảng quan trọng trong lĩnh vực IoT (Internet of Things). Hệ thống trên chip (SoC) này tích hợp cả Wi-Fi và Bluetooth, cho phép kết nối liên mạch với nhiều thiết bị. Kiến trúc của nó được xây dựng trên chip ESP32-D0WD, tận dụng các lõi vi xử lý LX6 32-bit có thể hoạt động ở tốc độ lên đến 240 MHz. Khả năng hiệu suất cao này giúp thực hiện các tác vụ phức tạp và quản lý nhiều hoạt động đồng thời, khiến ESP32 trở thành lựa chọn lý tưởng cho các ứng dụng yêu cầu xử lý thời gian thực.

Một tính năng nổi bật của ESP32 là hỗ trợ đa dạng các thiết bị ngoại vi, bao gồm UART, SPI, I2C, PWM, ADC và DAC, giúp dễ dàng tương tác với hệ sinh thái cảm biến và thiết bị truyền động phong phú. Sự linh hoạt này mở ra cơ hội cho nhiều đổi mới trong các dự án, từ tự động hóa gia đình đơn giản đến các ứng dụng công nghiệp phức tạp. Hơn nữa, chip có khả năng tiêu thụ điện năng thấp, phù hợp với các hệ thống chạy bằng pin, giúp kéo dài tuổi thọ hoạt động. Các nhà phát triển cũng được hưởng lợi từ môi trường phát triển mạnh mẽ như ESP-IDF (Espressif IoT Development Framework) và hỗ trợ các ngôn ngữ lập trình phổ biến như C và MicroPython, giúp đơn giản hóa quá trình xây dựng các ứng dụng phức tạp.

Trong lĩnh vực IoT, bảo mật luôn là yếu tố quan trọng. ESP32 tích hợp nhiều tính năng bảo mật, bao gồm khối bảo mật phần cứng, mã hóa flash và hỗ trợ mã hóa phần cứng, giúp tăng cường độ an toàn cho các thiết bị triển khai. Những biện pháp này cho phép các nhà phát triển xây dựng ứng dụng bảo vệ dữ liệu nhạy cảm và đảm bảo quyền riêng tư người dùng. Sự kết hợp giữa khả năng tính toán mạnh mẽ, tùy chọn kết nối đa dạng và bảo mật cao khiến ESP32 trở thành giải pháp giá trị cho các doanh nghiệp hướng tới hệ thống IoT có thể mở rộng, an toàn và hiệu quả. Những ưu điểm này không chỉ nâng cao chức năng mà còn thúc đẩy triển khai công nghệ tiên tiến trong nhà thông minh, chăm sóc sức khỏe và tự động hóa công nghiệp, khẳng định tiềm năng cách mạng của ESP32 trong việc định hình tương lai của các thiết bị kết nối.

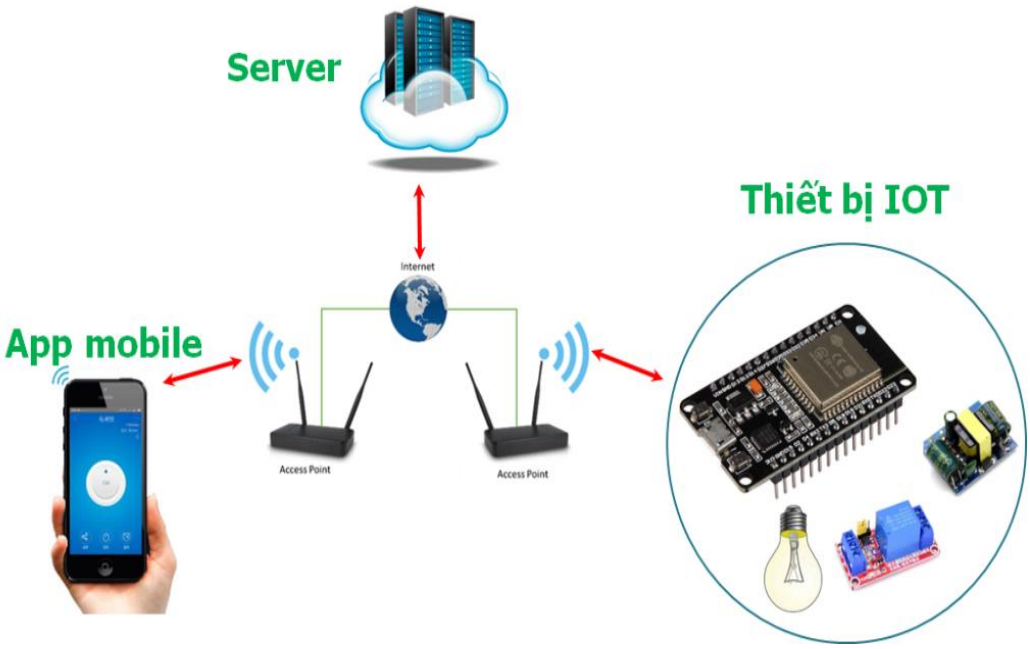


Hình ảnh sơ đồ chân kết nối ESP32

2.2 Khái niệm về máy chủ web cục bộ

Máy chủ web cục bộ là hệ thống cho phép lưu trữ và phân phối tài nguyên web trong mạng nội bộ mà không cần kết nối Internet, trở thành giải pháp lý tưởng để quản lý thiết bị IoT. Khác với máy chủ web truyền thống phục vụ người dùng qua Internet toàn cầu, máy chủ web cục bộ tập trung vào xử lý và giao tiếp dữ liệu giữa các thiết bị trong phạm vi mạng hạn chế như gia đình hoặc doanh nghiệp nhỏ. Mô hình này không chỉ giảm độ trễ trong truyền tải thông tin mà còn tăng cường bảo mật và kiểm soát dữ liệu - yếu tố ngày càng được các tổ chức và cá nhân coi trọng.

Trong bối cảnh IoT phát triển mạnh mẽ, máy chủ web cục bộ không chỉ đóng vai trò là nền tảng giao tiếp mà còn là công cụ mạnh để tùy biến và mở rộng chức năng thiết bị. Nhờ thiết kế linh hoạt, nó cho phép nhà phát triển không chỉ tạo ứng dụng tùy chỉnh mà còn tích hợp đa dạng mục tiêu khác nhau, nâng cao hiệu quả hệ thống IoT. Tóm lại, máy chủ web cục bộ không chỉ phản ánh sự tiến bộ công nghệ mà còn mở ra triển vọng lớn cho việc quản lý và điều khiển thiết bị trong tương lai.



Hình ảnh minh họa cho việc sử dụng webserver để quản lý thiết bị trong mạng cục bộ

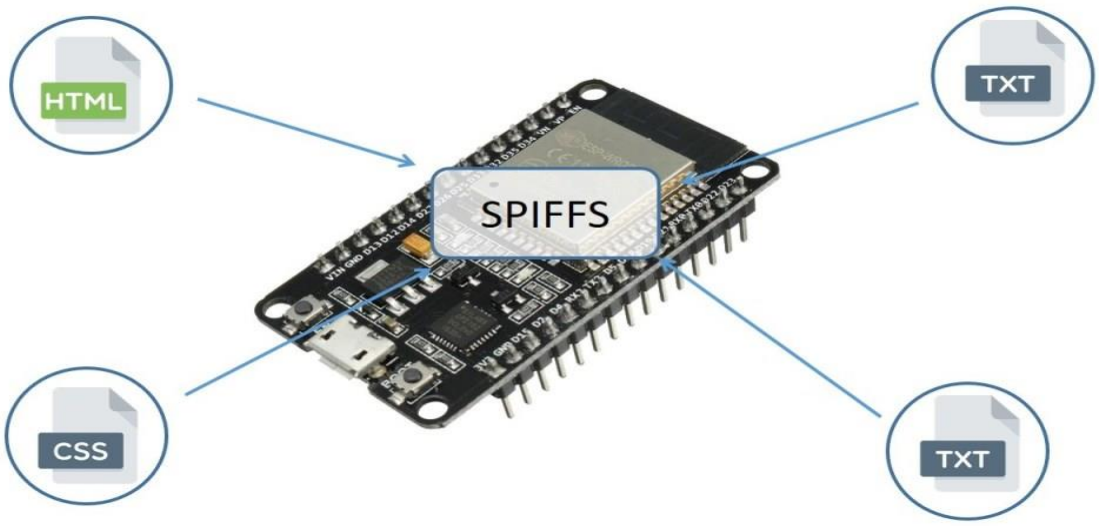
2.3 Giới thiệu về SPIFFS

SPIFFS (SPI Flash File System) là một phương pháp chuyên biệt để quản lý bộ nhớ không bay hơi trên vi điều khiển, đặc biệt là ESP32. Khi hệ thống nhúng mở rộng để đáp ứng nhiều chức năng hơn, quản lý dữ liệu hiệu quả trở nên quan trọng — nhu cầu mà SPIFFS đáp ứng trực tiếp. Khác với hệ thống tệp truyền thống, SPIFFS được thiết kế để hoạt động trong môi trường hạn chế bộ nhớ, cho phép lưu trữ và truy xuất tệp dễ dàng, chẳng hạn như cấu hình, trang web hoặc dữ liệu IoT khác. Kiến trúc của SPIFFS

hỗ trợ truy cập ngẫu nhiên, giúp thao tác tệp linh hoạt, phù hợp cho các dự án cần tính mềm dẻo.

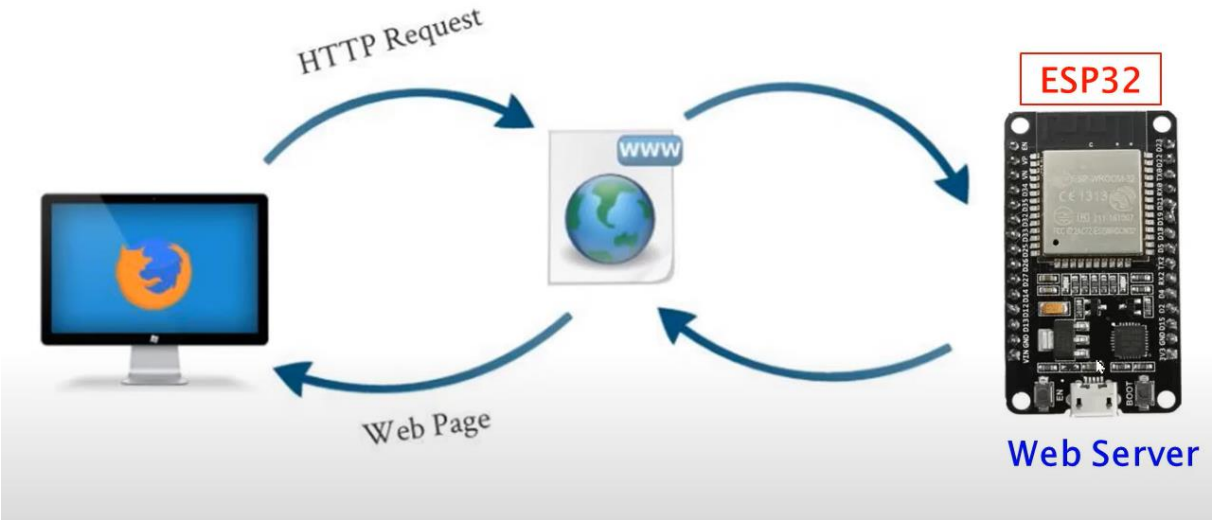
Một đặc điểm nổi bật của SPIFFS là khả năng thích ứng với nhiều loại bộ nhớ thông qua giao tiếp SPI (Serial Peripheral Interface). Tính linh hoạt này giúp nó hoạt động hiệu quả với các loại bộ nhớ flash tích hợp sẵn trên ESP32 hoặc giao tiếp bên ngoài. Hệ thống sử dụng kỹ thuật cân bằng hao mòn (wear leveling) để kéo dài tuổi thọ bộ nhớ flash — yếu tố quan trọng khi thiết bị liên tục đọc/ghi như trong IoT. SPIFFS cũng tích hợp cơ chế quản lý phân mảnh, tối ưu hóa không gian lưu trữ. Kết quả là một giải pháp mạnh mẽ, vừa đơn giản hóa quản lý dữ liệu vừa duy trì hiệu suất tối ưu trong môi trường động.

Trong ứng dụng thực tế, nhà phát triển có thể dùng SPIFFS để tạo máy chủ web động trên ESP32, nơi phân bổ tài nguyên và thao tác tệp rất quan trọng cho hiệu suất. SPIFFS cho phép lưu trữ HTML, CSS và JavaScript trực tiếp trên ESP32, hỗ trợ phát triển giao diện web tương tác để điều khiển và giám sát thiết bị IoT. Đối với máy chủ web cục bộ, khả năng xử lý tệp hiệu quả ảnh hưởng lớn đến trải nghiệm người dùng, giúp tương tác với thiết bị trở nên mượt mà và trực quan. Do đó, hiểu và sử dụng SPIFFS đóng vai trò then chốt trong triển khai thành công các giải pháp IoT, kết nối khả năng phần cứng với chức năng phần mềm.



Hình ảnh minh họa việc sử dụng SPIFFS trên ESP32

2.4 Web cục bộ trên ESP32



Hình ảnh minh họa phương thức hoạt động của webserver trên ESP32

Khi triển khai máy chủ web cục bộ trên nền tảng ESP32 (một vi điều khiển phổ biến trong IoT), người dùng có thể quản lý và điều khiển thiết bị một cách trực tiếp và hiệu quả. ESP32 tích hợp khả năng kết nối Wi-Fi và Bluetooth, cho phép các thiết bị giao tiếp với nhau mà không phụ thuộc vào mạng bên ngoài. Trong mạng cục bộ, thiết bị có thể gửi/nhận dữ liệu qua giao thức HTTP, giúp người dùng dễ dàng truy cập, giám sát và điều chỉnh thông số hoạt động từ xa thông qua giao diện web trực quan. Ngoài ra, việc sử dụng SPIFFS (SPI Flash File System) trên ESP32 cho phép lưu trữ các tệp HTML, CSS và JavaScript trực tiếp trên vi điều khiển, tạo điều kiện thuận lợi cho việc triển khai giao diện web mà không cần phụ thuộc vào bộ nhớ ngoài.

CHƯƠNG 3: TỔNG QUAN VỀ HỆ THỐNG

3.1 Tổng quan chi tiết về Hệ thống Web cục bộ trên ESP32 để Điều khiển Thiết bị IoT

Hệ thống web server trên ESP32 để điều khiển các thiết bị IoT là một ví dụ điển hình về cách ứng dụng công nghệ Internet of Things (IoT) trong việc quản lý và tự động hóa các thiết bị thông minh. Với sự kết hợp giữa phần cứng mạnh mẽ của ESP32 và phần mềm linh hoạt, hệ thống này cho phép người dùng điều khiển các thiết bị như đèn LED, quạt, và máy bơm,.. thông qua một giao diện web thân thiện được xây dựng dựa trên SPIFFS (SPI Flash File System). Tổng quan này sẽ phân tích chi tiết các thành phần, luồng hoạt động, công nghệ được sử dụng, ưu điểm, hạn chế, và tiềm năng mở rộng của hệ thống.

3.2 Mục tiêu và ý nghĩa của hệ thống

Hệ thống được thiết kế nhằm cung cấp một giải pháp IoT đơn giản nhưng hiệu quả, cho phép người dùng điều khiển các thiết bị từ xa thông qua trình duyệt web mà không cần cài đặt ứng dụng bổ sung. Mục tiêu chính bao gồm:

- **Tự động hóa và điều khiển từ xa**: Người dùng có thể bật/tắt các thiết bị như đèn LED, quạt, hoặc máy bơm từ bất kỳ thiết bị nào có trình duyệt, miễn là cùng kết nối vào mạng WiFi nội bộ.
- **Giao diện thân thiện**: Cung cấp một bảng điều khiển (dashboard) trực quan, hiển thị trạng thái thiết bị và cho phép điều khiển dễ dàng.

- **Hiệu suất cao**: Sử dụng web server không đồng bộ để xử lý nhiều yêu cầu cùng lúc, đảm bảo hệ thống hoạt động mượt mà.
- **Chi phí thấp và dễ triển khai**: ESP32 là một vi điều khiển giá rẻ, dễ lập trình, phù hợp cho các dự án IoT quy mô nhỏ đến trung bình.

Hệ thống này có ý nghĩa lớn trong bối cảnh tự động hóa nhà thông minh ngày càng phổ biến. Nó không chỉ giúp tiết kiệm thời gian và công sức mà còn mở ra tiềm năng ứng dụng trong nhiều lĩnh vực khác như nông nghiệp thông minh (điều khiển máy bơm tưới cây), công nghiệp (giám sát và điều khiển thiết bị), hoặc giáo dục (dự án học tập về IoT).

3.3. Các thành phần chính của hệ thống

Hệ thống bao gồm ba thành phần chính, được minh họa rõ ràng trong sơ đồ luồng dữ liệu:

3.3.1 Client (Máy tính hoặc thiết bị người dùng)

- **Vai trò**: Client là thiết bị mà người dùng sử dụng để tương tác với hệ thống, thường là máy tính, điện thoại hoặc máy tính bảng có trình duyệt web (như Firefox, Chrome, Safari).
- **Chức năng**:
    - Hiển thị giao diện web (HTML) được gửi từ ESP32.
    - Gửi yêu cầu HTTP đến ESP32 để lấy dữ liệu hoặc điều khiển thiết bị.
    - Cập nhật trạng thái thiết bị trên giao diện mà không cần tải lại trang.
- **Liên hệ với sơ đồ**: Trong hình ảnh, client được biểu thị bằng màn hình máy tính với logo Firefox, đại diện cho trình duyệt web.

3.3.2 Mạng (WiFi nội bộ)

- **Vai trò**: Mạng là cầu nối giữa client và ESP32, cho phép hai thiết bị giao tiếp với nhau.
- **Chi tiết**:

- Hệ thống sử dụng mạng WiFi nội bộ (LAN), không yêu cầu kết nối Internet toàn cầu.
- ESP32 và client phải cùng kết nối vào một mạng WiFi (cùng SSID và mật khẩu).
- ESP32 hoạt động ở chế độ Station (STA), kết nối với router WiFi hiện có.

- **Liên hệ với sơ đồ**: Mạng được biểu thị bằng biểu tượng quả địa cầu với chữ "WWW", ám chỉ kết nối mạng (dù trong trường hợp này là mạng nội bộ).

3.3.3 ESP32 (Web Server)

- **Vai trò**: ESP32 là trung tâm của hệ thống, đóng vai trò vừa là web server vừa là bộ điều khiển phần cứng.
- **Chức năng**:
  - **Web Server**: Xử lý các yêu cầu HTTP từ client và trả về phản hồi (trang HTML, dữ liệu JSON, hoặc trạng thái thiết bị).
  - **Điều khiển phần cứng**: Sử dụng các chân GPIO để điều khiển các thiết bị IoT như đèn LED, quạt, và máy bơm.
  - **Lưu trữ tệp**: Sử dụng hệ thống tệp SPIFFS để lưu trữ file HTML và các tài nguyên tĩnh khác.
- **Liên hệ với sơ đồ**: ESP32 được biểu thị bằng hình ảnh bo mạch với nhãn "ESP32" và "Web Server", thể hiện vai trò của nó trong hệ thống.

3.3.4 Thiết bị IoT

- **Vai trò**: Đây là các thiết bị vật lý được điều khiển bởi ESP32.
- **Chi tiết**:
  - Hệ thống quản lý ba thiết bị: đèn LED (GPIO2), quạt (GPIO4), và máy bơm (GPIO5),..
  - Mỗi thiết bị được định nghĩa trong một mảng cấu trúc Device, bao gồm ID, tên, chân GPIO và trạng thái (ON/OFF).

- ESP32 điều khiển thiết bị bằng cách đặt mức logic cao (HIGH) hoặc thấp (LOW) trên các chân GPIO tương ứng.

CHƯƠNG 4: TRIỂN KHAI VÀ THIẾT KẾ HỆ THỐNG

4.1 Các bước triển khai

4.1.1 Chuẩn bị phần cứng

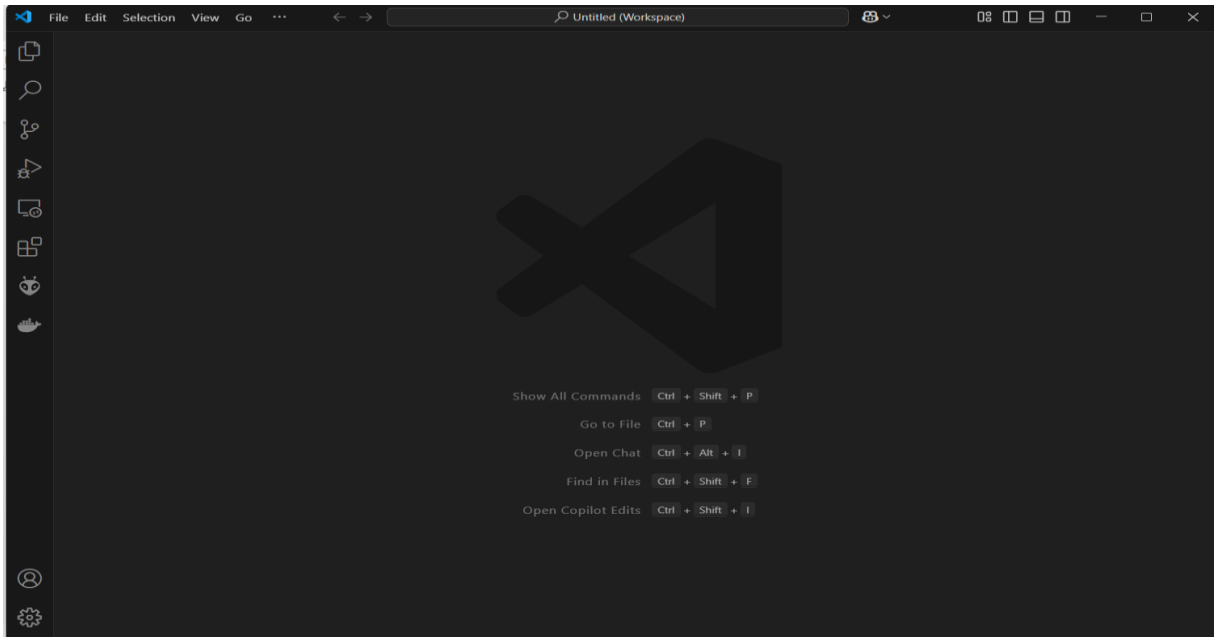
- Một module ESP32 (ESP32 Dev Module).
- Đèn LED (kèm điện trở 220Ω).
- Quạt mini (5V hoặc dùng relay nếu công suất lớn).
- Máy bơm mini (5V hoặc dùng relay).
- Breadboard và dây jumper để kết nối.
- Kết nối: Cáp USB để kết nối ESP với máy tính.

4.1.2 Chuẩn bị phần mềm

4.1.2.1 Phần mềm phát triển

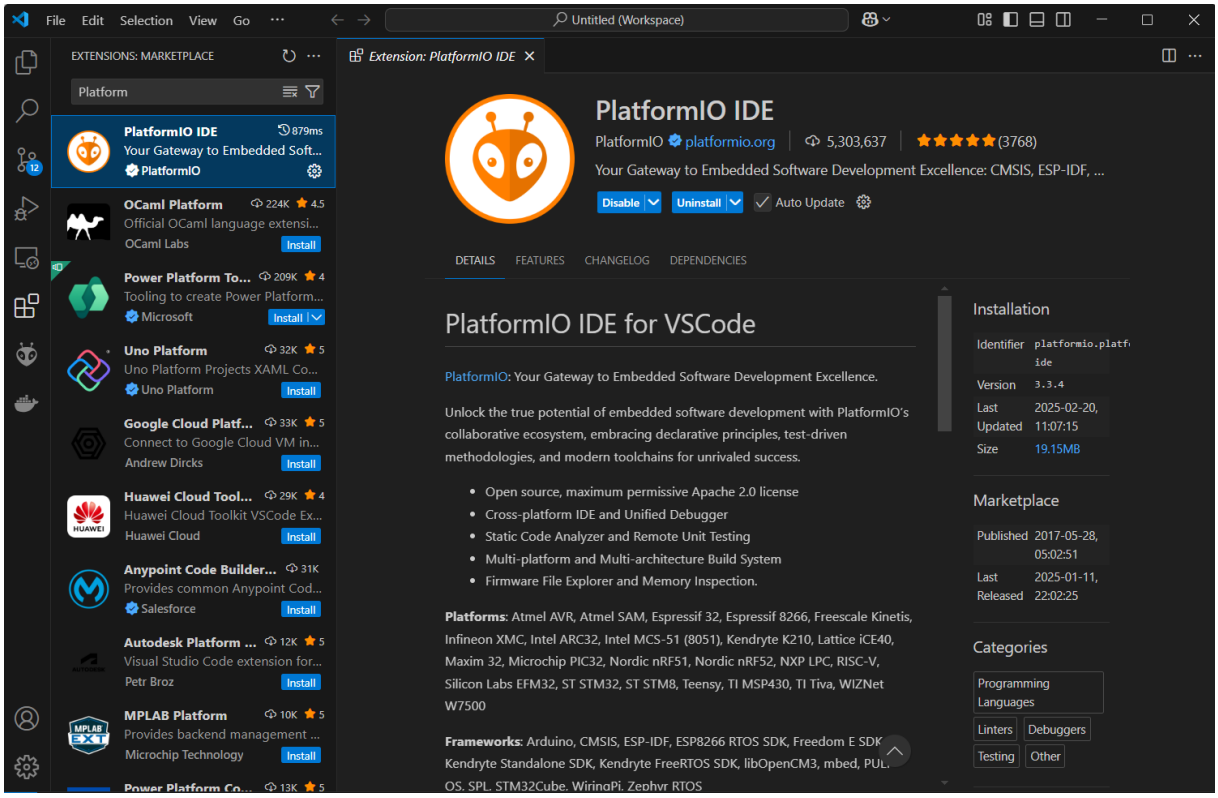
Bước 1: Cài đặt VS Code

- Tải và cài đặt VS Code từ trang chính thức: https://code.visualstudio.com/



Bước 2: Cài đặt tiện ích PlatformIO

- Mở VS Code, vào Extensions (Ctrl + Shift + X)
- Tìm PlatformIO IDE và cài đặt
- Sau khi cài đặt xong, khởi động lại VS Code



4.1.2.2 Trình duyệt web

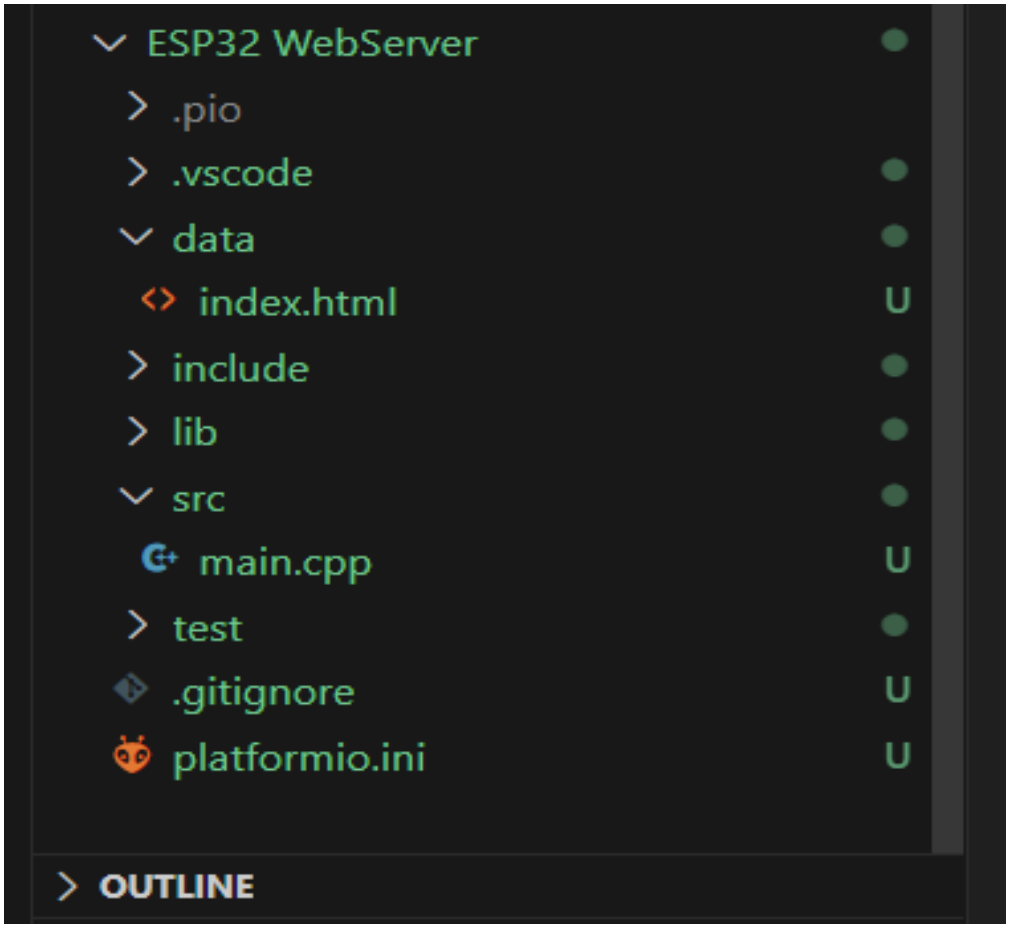
- Dùng Chrome, Firefox hoặc bất kỳ trình duyệt nào để truy cập giao diện web sau khi hoàn thiện.

4.1.3 Các bước thực hiện

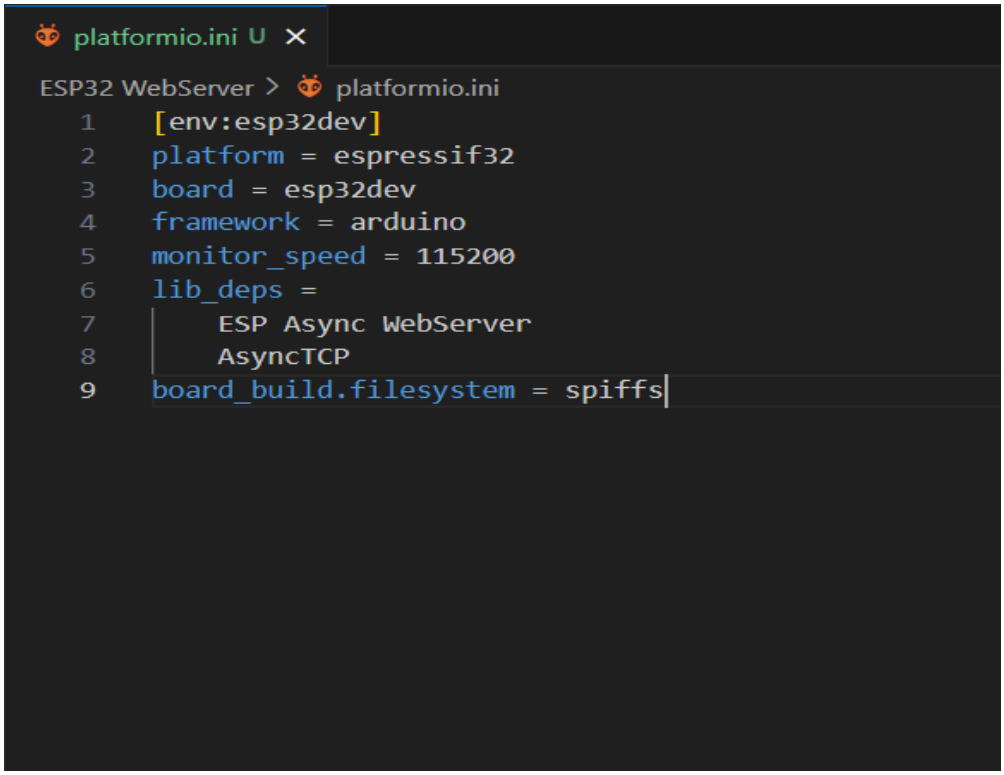
Bước 1: Cài đặt môi trường ESP32

- Mở PlatformIO Home
- Chọn New Project → Đặt tên dự án (ESP32 WebServer).
- Ở mục Board, chọn ESP32 Dev Module
- Ở mục Framework, chọn Arduino hoặc ESP-IDF (tùy vào nhu cầu lập trình)
- Nhấn Finish để tạo dự án

Bước 2: Cấu trúc tệp ESP32 WebServer



Bước 3: Cấu hình platformio.ini



Bước 4: Thiết kế giao diện web

Tạo tệp data/index.html bao gồm và JavaScript

```html
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>ESP32 IoT Dashboard</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            background-color: #f4f4f4;
            margin: 0;
            padding: 20px;
            text-align: center;
        }
        h1 {
            color: #333;
            margin-bottom: 20px;
        }
        .container {
            max-width: 800px;
            margin: 0 auto;
            display: flex;
            flex-wrap: wrap;
            justify-content: center;
        }
        .device {
            background-color: white;
            border-radius: 10px;
            box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1);
            width: 200px;
            padding: 20px;
            margin: 10px;
            text-align: center;
        }
        .device h3 {
            margin: 0 0 10px;
            color: #555;
        }
        .status {
            font-weight: bold;
            margin: 10px 0;
        }
        .button {
            padding: 10px 20px;
            font-size: 14px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
            transition: background-color 0.3s;
        }
        .on {
            background-color: #4CAF50;
            color: white;
        }
        .on:hover {
            background-color: #45a049;
        }
        .off {
            background-color: #f44336;
            color: white;
        }
        .off:hover {
            background-color: #da190b;
        }
    </style>
</head>
<body>
    <h1>ESP32 IoT Dashboard</h1>
    <div class="container" id="devices"></div>

    <script>
        // Tải danh sách thiết bị khi mở trang
        function loadDevices() {
            fetch('/devices')
                .then(response => response.json())
                .then(data => {
                    let devicesDiv = document.getElementById('devices');
                    devicesDiv.innerHTML = '';
                    data.forEach(device => {
                        devicesDiv.innerHTML += `
                            <div class="device">
                                <h3>${device.name}</h3>
                                <p class="status">Status: <span id="status-${device.id}">${device.state}</span></p>
                                <button class="button on" onclick="setDevice('${device.id}', 'ON')">Turn ON</button>
                                <button class="button off" onclick="setDevice('${device.id}', 'OFF')">Turn OFF</button>
                            </div>`;
                    });
                });
        }

        // Gửi yêu cầu điều khiển thiết bị
        function setDevice(id, state) {
            fetch(`/control?id=${id}&state=${state}`)
                .then(response => response.text())
                .then(data => document.getElementById(`status-${id}`).innerText = data);
        }

        // Tải thiết bị khi trang mở
        window.onload = loadDevices;
    </script>
</body>
</html>
```

Mã nguồn index.html minh họa

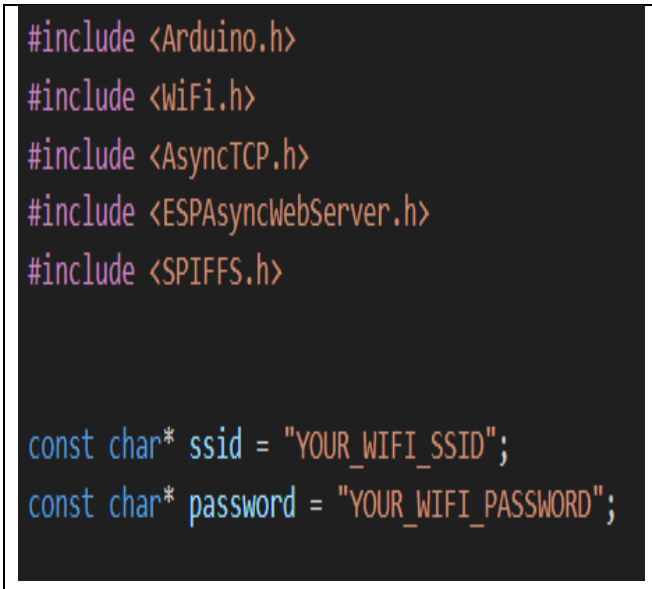
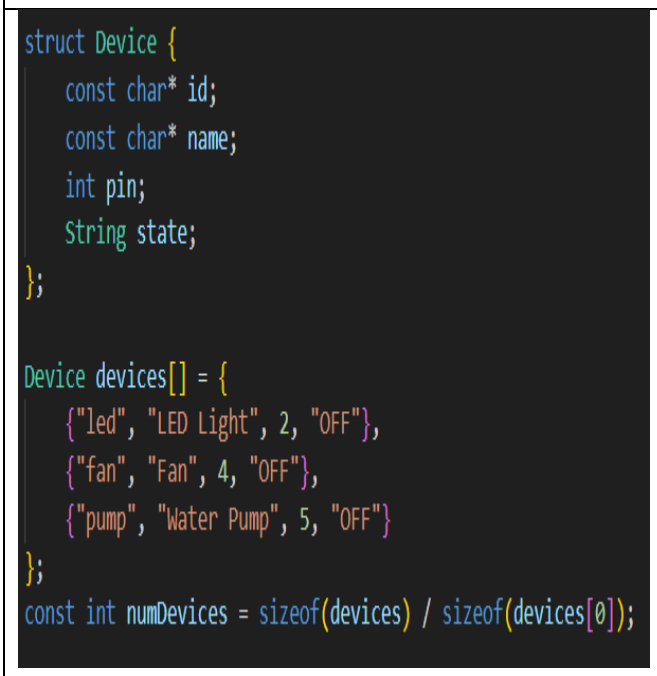
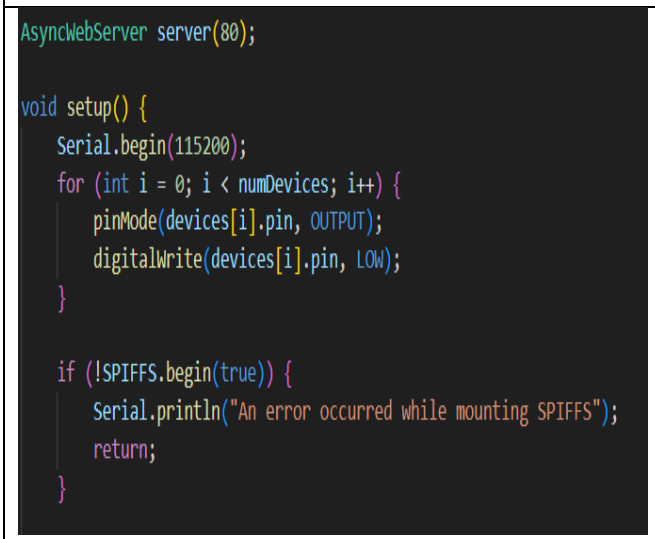
Bước 5: Viết mã nguồn

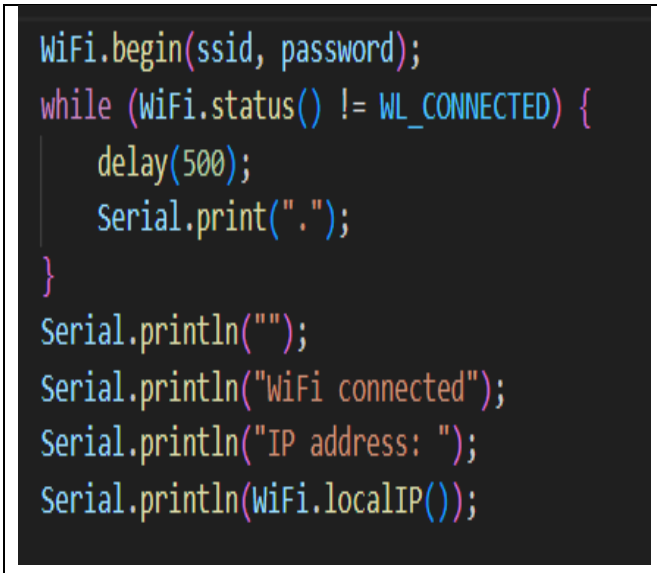
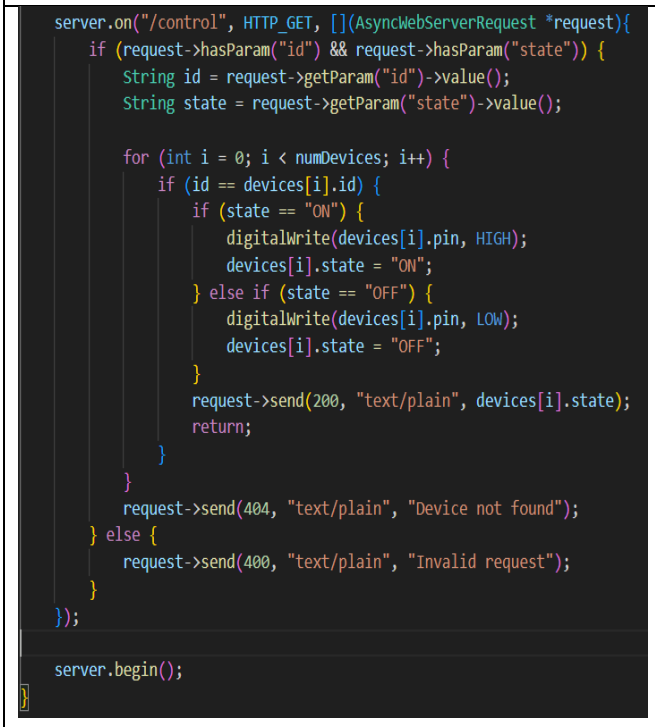
Mở src/main.cpp

```cpp
#include <Arduino.h>
#include <WiFi.h>
#include <AsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <SPIFFS.h>

// Thông tin WiFi
const char* ssid = "YOUR_WIFI_SSID";
const char* password = "YOUR_WIFI_PASSWORD";

// Cấu hình các thiết bị IoT
struct Device {
    const char* id;
    const char* name;
    int pin;
    String state;
};

Device devices[] = {
    {"led", "LED Light", 2, "OFF"},     // GPIO2 cho đèn LED
    {"fan", "Fan", 4, "OFF"},           // GPIO4 cho quạt
    {"pump", "Water Pump", 5, "OFF"}    // GPIO5 cho máy bơm
};
const int numDevices = sizeof(devices) / sizeof(devices[0]);

// Tạo đối tượng server
AsyncWebServer server(80);

void setup() {
    Serial.begin(115200);

    // Khởi tạo các chân GPIO
    for (int i = 0; i < numDevices; i++) {
        pinMode(devices[i].pin, OUTPUT);
        digitalWrite(devices[i].pin, LOW); // Tắt tất cả thiết bị ban đầu
    }

    // Khởi tạo SPIFFS
    if (!SPIFFS.begin(true)) {
        Serial.println("An error occurred while mounting SPIFFS");
        return;
    }

    // Kết nối WiFi
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("WiFi connected");
    Serial.println("IP address: ");
    Serial.println(WiFi.localIP());

    // Route: Trang chính
    server.on("/", HTTP_GET, [](AsyncWebServerRequest *request){
        request->send(SPIFFS, "/index.html", "text/html");
    });

    // Route: Lấy danh sách thiết bị (JSON)
    server.on("/devices", HTTP_GET, [](AsyncWebServerRequest *request){
        String json = "[";
        for (int i = 0; i < numDevices; i++) {
            json += "{\"id\":\"" + String(devices[i].id) + "\",";
            json += "\"name\":\"" + String(devices[i].name) + "\",";
            json += "\"state\":\"" + devices[i].state + "\"}";
            if (i < numDevices - 1) json += ",";
        }
        json += "]";
        request->send(200, "application/json", json);
    });

    // Route: Điều khiển thiết bị
    server.on("/control", HTTP_GET, [](AsyncWebServerRequest *request){
        if (request->hasParam("id") && request->hasParam("state")) {
            String id = request->getParam("id")->value();
            String state = request->getParam("state")->value();

            for (int i = 0; i < numDevices; i++) {
                if (id == devices[i].id) {
                    if (state == "ON") {
                        digitalWrite(devices[i].pin, HIGH);
                        devices[i].state = "ON";
                    } else if (state == "OFF") {
                        digitalWrite(devices[i].pin, LOW);
                        devices[i].state = "OFF";
                    }
                    request->send(200, "text/plain", devices[i].state);
                    return;
                }
            }
            request->send(404, "text/plain", "Device not found");
        } else {
            request->send(400, "text/plain", "Invalid request");
        }
    });

    // Khởi động server
    server.begin();
}

void loop() {
    // Không cần xử lý trong loop
}
```

Mã nguồn main.cpp minh họa

Giải thích mã nguồn main.cpp

	Thư viện cơ bản của Arduino Thư viện để kết nối WiFi trên Thư viện hỗ trợ TCP không đồng bộ Thư viện để tạo web server không đồng bộ Thư viện để quản lý hệ thống tệp trên ESP32 Tên WiFi (SSID) của bạn Mật khẩu WiFi của bạn
	**Cấu hình các thiết bị IoT** ID duy nhất của thiết bị Tên thiết bị hiển thị trên giao diện Số chân GPIO mà thiết bị được kết nối Trạng thái hiện tại của thiết bị **Danh sách các thiết bị IoT** Thiết bị đèn LED nối với chân GPIO2 Thiết bị quạt nối với chân GPIO4 Thiết bị máy bơm nối với chân GPIO5 Tính số lượng thiết bị
	Tạo đối tượng server chạy trên cổng 80 Khởi động cổng Serial để debug với tốc độ 115200 baud Khởi tạo các chân GPIO cho thiết bị Đặt chân GPIO thành chế độ OUTPUT Tắt tất cả thiết bị khi khởi động Khởi tạo SPIFFS (hệ thống tệp để lưu trữ file HTML)

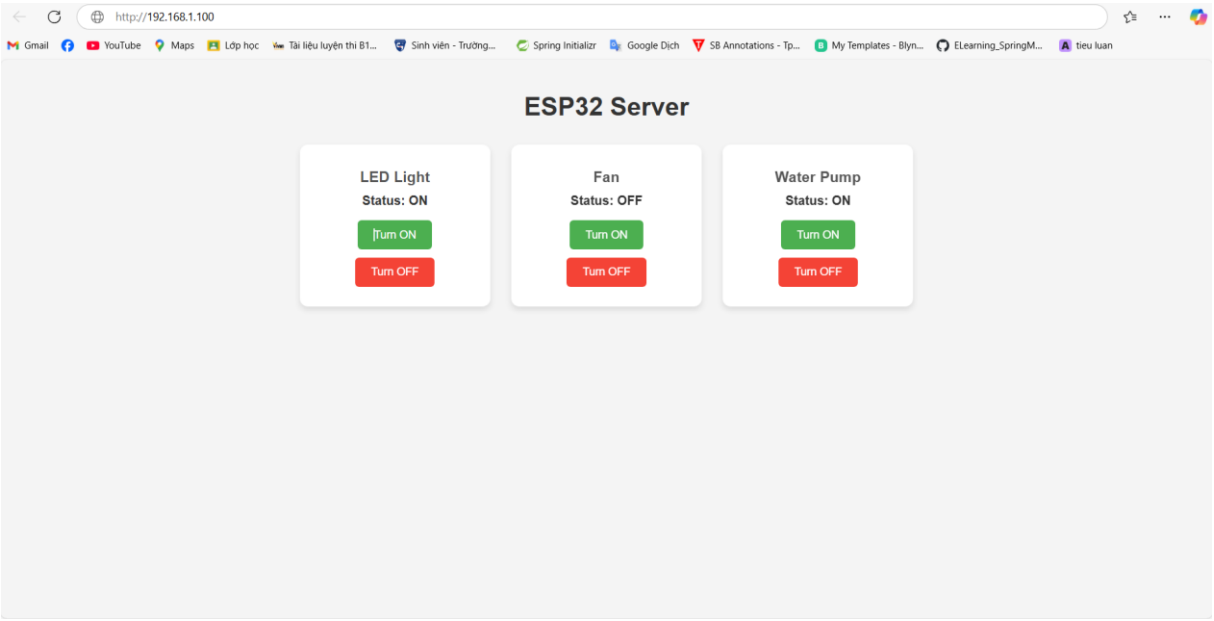
	**Kết nối WiFi** Bắt đầu kết nối với WiFi Chờ đến khi kết nối thành công Thông báo kết nối thành công Hiển thị địa chỉ IP của ESP32
	Trả về trang chính (index.html) Gửi file HTML từ SPIFFS Trả về danh sách thiết bị dưới dạng JSON khi truy cập "/devices" Kết thúc chuỗi JSON Gửi phản hồi JSON
	Điều khiển thiết bị khi truy cập"/control" Bật thiết bị Cập nhật trạng thái Tắt thiết bị Cập nhật trạng thái Gửi trạng thái mới Khởi động server

Bước 6: Nạp mã và tệp

- Nạp mã lên ESP32: Nhấn Upload trong PlatformIO.
- Nạp tệp HTML vào SPIFFS: Nhấn Upload File System Image.

Bước 7: Kiểm tra

- Mở Serial Monitor để lấy địa chỉ IP (ví dụ: 192.168.1.100).
- Truy cập http://192.168.1.100 trên trình duyệt.



Giao diện webserver quản lý các thiết bị Iot

CHƯƠNG 5: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1. Kết quả đạt được

Đề tài đã xây dựng thành công máy chủ web cục bộ trên ESP32 để quản lý thiết bị IoT, với các kết quả nổi bật:

- Tạo máy chủ web không đồng bộ, điều khiển thiết bị (đèn LED, quạt, máy bơm) qua giao diện web trực quan.
- Sử dụng SPIFFS để lưu trữ file HTML, đảm bảo quản lý tài nguyên hiệu quả.
- Giao diện web thân thiện, dễ dàng sử dụng
- Hệ thống chi phí thấp, dễ triển khai, phù hợp cho tự động hóa nhà thông minh và giáo dục.

5.2. Ý nghĩa của đề tài

Hệ thống minh họa tiềm năng của ESP32 trong IoT. Mang lại giải pháp tiện lợi cho tự động hóa. Đề tài cũng có giá trị giáo dục, giúp sinh viên hiểu về lập trình nhúng, thiết kế web, và ứng dụng IoT.

5.3. Hạn chế

- Chỉ hoạt động trong mạng nội bộ, không hỗ trợ điều khiển qua Internet.
- Chưa có bảo mật (xác thực, mã hóa).
- SPIFFS giới hạn dung lượng, chưa tích hợp cảm biến để tự động hóa thông minh.
- Chưa tích hợp cảm biến: Hệ thống hiện chỉ tập trung vào điều khiển thiết bị (bật/tắt) mà chưa tích hợp các cảm biến để giám sát môi trường, chẳng hạn như cảm biến nhiệt độ, độ ẩm, hoặc ánh sáng. Điều này làm giảm khả năng tự động hóa thông minh của hệ thống.

5.4. Hướng phát triển

- Hỗ trợ điều khiển qua Internet (MQTT, Blynk).
- Tăng bảo mật (HTTPS, xác thực người dùng).
- Tích hợp cảm biến (nhiệt độ, độ ẩm) và tự động hóa thông minh.
- Lưu trữ dữ liệu (Firebase) và nâng cấp giao diện (biểu đồ, thông báo).

- Tích hợp với hệ thống nhà thông minh (Google Home, Alexa).

5.5. Kết luận

Đề tài đã đạt mục tiêu, tạo nền tảng cho các ứng dụng IoT thực tiễn. Dù còn hạn chế, hệ thống có tiềm năng mở rộng lớn. Em xin cảm ơn thầy Võ Việt Dũng và nhóm 09 đã hỗ trợ trong quá trình thực hiện.

TÀI LIỆU THAM KHẢO

Espressif Systems. (2023). *ESP32 Technical Reference Manual*. Truy cập từ: https://www.espressif.com/sites/default/files/documentation/esp32_technical_reference_manual_en.pdf

(Tài liệu chính thức từ Espressif Systems, cung cấp thông tin chi tiết về kiến trúc, tính năng, và cách sử dụng ESP32 trong các ứng dụng IoT.)

Random Nerd Tutorials. (2024). *ESP32 Web Server with Arduino IDE*. Truy cập từ: https://randomnerdtutorials.com/esp32-web-server-arduino-ide/

(Hướng dẫn chi tiết cách xây dựng máy chủ web trên ESP32, bao gồm mã nguồn và cách tích hợp giao diện HTML.)

Espressif Systems. (2023). *SPIFFS Documentation*. Truy cập từ: https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-reference/storage/spiffs.html

(Tài liệu chính thức về SPIFFS, giải thích cách sử dụng hệ thống tệp trên ESP32 để lưu trữ tài nguyên web.)

PlatformIO. (2024). *PlatformIO Documentation*. Truy cập từ: https://docs.platformio.org/en/latest/

(Tài liệu hướng dẫn sử dụng PlatformIO để phát triển các dự án nhúng, bao gồm cấu hình và triển khai mã trên ESP32.)

Blynk. (2024). *Blynk IoT Platform Documentation*. Truy cập từ: https://docs.blynk.io/en/

(Tài liệu về nền tảng Blynk, hữu ích khi mở rộng hệ thống để điều khiển thiết bị IoT từ xa.)

Arduino. (2024). *WiFi Library for ESP32*. Truy cập từ: https://www.arduino.cc/reference/en/libraries/wifi/
(Thư viện WiFi của Arduino, cung cấp các hàm cơ bản để kết nối và quản lý mạng trên ESP32.)

Instructables. (2023). *Building an ESP32 Local Web Server for IoT Control*. Truy cập từ: https://www.instructables.com/ESP32-Local-Web-Server-for-IoT-Control/
(Hướng dẫn thực tế về cách xây dựng máy chủ web cục bộ trên ESP32 để điều khiển các thiết bị IoT như đèn LED và quạt.)

Microchip Technology. (2023). *Introduction to IoT and Embedded Systems*. Truy cập từ: https://www.microchip.com/en-us/solutions/internet-of-things
(Bài viết cung cấp cái nhìn tổng quan về IoT, vai trò của vi điều khiển trong tự động hóa và quản lý thiết bị.)