

ĐẠI HỌC HUẾ
TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN

BÁO CÁO ĐỀ TÀI TIỂU LUẬN

Học Phần : Phát triển ứng dụng IoT - Nhóm 5

Đề tài:

Tạo máy chủ web cục bộ với ESP32 để quản lý thiết bị IoT

Sinh viên thực hiện: PHẠM MẠNH HOÀNG

Giáo viên hướng dẫn: Võ Việt Dũng

Thừa Thiên Huế, tháng 4/2025

MỤC LỤC

MỞ ĐẦU

1. Giới thiệu đề tài.....	1
2. Mục tiêu nghiên cứu.....	1
3. Phương pháp nghiên cứu.....	2

NỘI DUNG

1. Tổng quan về ESP32	
a. Giới thiệu ESP32 và các đặc điểm nổi bật.....	3
b. Ứng dụng của ESP32 trong các hệ thống IoT.....	4
2. Máy chủ web cục bộ trên ESP32	
a. Khái niệm về máy chủ web và ứng dụng trong IoT.....	4
b. Cách hoạt động của máy chủ web cục bộ trên ESP32.....	4
3. Lưu trữ và quản lý giao diện web với SPIFFS	
a. Giới thiệu SPIFFS (SPI Flash File System).....	5
b. Cách ESP32 sử dụng SPIFFS để lưu trữ file HTML, CSS, JavaScript.....	5
4. Giao thức truyền thông	
a. HTTP: Xử lý request từ trình duyệt web.....	5
b. WebSocket: Cập nhật trạng thái thiết bị theo thời gian thực.....	5
5. Thiết kế hệ thống	
a. Phần cứng.....	6
b. Phần mềm , mô phỏng Wokwi.....	7
c. Gửi dữ liệu lên nền tảng IoT.....	8
6. Kết quả và đánh giá	8
7. Kết luận và hướng phát triển.....	9

MỞ ĐẦU

1. Giới thiệu đề tài

Trong thời đại công nghệ 4.0, Internet of Things (IoT) đã trở thành một trong những xu hướng công nghệ nổi bật, mang lại nhiều ứng dụng thực tiễn trong đời sống như nhà thông minh, nông nghiệp thông minh, giám sát môi trường, v.v. Các thiết bị IoT thường được kết nối vào một hệ thống mạng để thu thập dữ liệu, điều khiển và giao tiếp với nhau hoặc với con người thông qua các nền tảng trực tuyến.

ESP32 là một vi điều khiển tích hợp Wi-Fi và Bluetooth, được sử dụng phổ biến trong các dự án IoT nhờ vào giá thành thấp, khả năng kết nối mạnh mẽ và hiệu suất xử lý tốt. Một trong những điểm mạnh của ESP32 là khả năng hoạt động như một máy chủ web cục bộ, cho phép người dùng trực tiếp giao tiếp và điều khiển các thiết bị IoT thông qua giao diện web được lưu trữ ngay trên vi điều khiển.

Đề tài “Tạo máy chủ web cục bộ với ESP32 để quản lý thiết bị IoT” nhằm khai thác khả năng của ESP32 trong việc tích hợp giao diện điều khiển thông qua mạng nội bộ. Hệ thống sẽ sử dụng SPIFFS để lưu trữ giao diện web, giúp người dùng dễ dàng cấu hình và theo dõi trạng thái của các thiết bị IoT một cách trực quan và tiện lợi. Ngoài ra, việc kết hợp các công nghệ như HTTP, WebSocket hoặc MQTT sẽ giúp nâng cao khả năng giao tiếp thời gian thực giữa người dùng và thiết bị.

2. Mục tiêu nghiên cứu

Mục tiêu của đề tài là thiết kế và hiện thực một mô hình quản lý thiết bị IoT sử dụng ESP32 như một máy chủ web cục bộ, có khả năng tương tác trực tiếp với người dùng thông qua trình duyệt, không cần kết nối Internet bên ngoài:

- Tạo máy chủ web trên ESP32 sử dụng SPIFFS:

Cho phép ESP32 đóng vai trò như một máy chủ HTTP mini, phục vụ nội dung giao diện web cho người dùng trong mạng LAN.

Giao diện này sẽ được lưu trữ trong bộ nhớ Flash của ESP32 thông qua hệ thống file SPIFFS.

- Cung cấp giao diện web cấu hình và giám sát thiết bị:

Người dùng có thể điều khiển thiết bị IoT (như đèn, cảm biến...) thông qua giao diện web thân thiện.

Giao diện cũng hiển thị trạng thái thời gian thực của thiết bị, giúp giám sát dễ dàng.

- Ứng dụng các giao thức HTTP, WebSocket hoặc MQTT:

HTTP dùng để thực hiện các thao tác truy xuất thông tin hoặc thay đổi cấu hình.

WebSocket được sử dụng để truyền dữ liệu hai chiều theo thời gian thực, nâng cao khả năng phản hồi và giám sát.

MQTT (nếu áp dụng) phục vụ trong môi trường mở rộng hoặc hệ thống nhiều thiết bị giao tiếp qua broker trung gian.

3. Phương pháp nghiên cứu

Để đạt được mục tiêu trên, đề tài sử dụng các phương pháp nghiên cứu sau:

- * Tìm hiểu về ESP32 và SPIFFS:

Nắm rõ cấu trúc phần cứng của ESP32, đặc biệt là khả năng kết nối Wi-Fi, bộ nhớ Flash, RAM và khả năng xử lý đa luồng. Nghiên cứu về SPIFFS – hệ thống tập tin đơn giản, thích hợp cho thiết bị nhúng không có hệ điều hành, dùng để lưu trữ file HTML, CSS, JS trực tiếp trong Flash [1].

- * Xây dựng mô hình hệ thống phần cứng - phần mềm:

- Phần cứng: sử dụng ESP32 làm trung tâm điều khiển, kết nối với các thiết bị như relay, cảm biến nhiệt độ, độ ẩm, hoặc đèn LED.

- Phần mềm: Lập trình ESP32 bằng Arduino IDE hoặc ESP-IDF, Tạo và quản lý file hệ thống SPIFFS, Thiết kế giao diện web (frontend) thân thiện người dùng, có thể truy cập qua trình duyệt, Triển khai WebSocket hoặc HTTP API để truyền thông hai chiều giữa giao diện và thiết bị.

* Đánh giá hệ thống và đề xuất cải tiến:

- Thử nghiệm hệ thống trong môi trường mạng nội bộ: kiểm tra độ ổn định, thời gian phản hồi, khả năng xử lý đồng thời nhiều yêu cầu.
- Đánh giá khả năng mở rộng: thêm nhiều thiết bị, hoặc nâng cấp kết nối MQTT nếu muốn triển khai quy mô rộng hơn.
- Đưa ra các cải tiến về bảo mật, tối ưu hóa bộ nhớ hoặc giao diện người dùng.

NỘI DUNG

1. Tổng quan về ESP32

a. Giới thiệu ESP32 và các đặc điểm nổi bật

ESP32 là vi điều khiển do hãng Espressif Systems phát triển, tích hợp Wi-Fi và Bluetooth, rất phù hợp cho các ứng dụng IoT. Một số đặc điểm nổi bật của ESP32 bao gồm:

- CPU: Dual-core Tensilica LX6, xung nhịp lên đến 240 MHz.
- Bộ nhớ RAM: 520 KB SRAM, hỗ trợ bộ nhớ ngoài.
- Tích hợp Wi-Fi 802.11 b/g/n và Bluetooth 4.2 BLE.
- Hỗ trợ nhiều giao tiếp ngoại vi: UART, SPI, I2C, PWM, ADC, DAC...
- Tiêu thụ điện năng thấp, có thể hoạt động trong các chế độ tiết kiệm năng lượng.

ESP32 được hỗ trợ bởi nhiều công cụ phát triển như Arduino IDE, PlatformIO và ESP-IDF – SDK chính thức từ Espressif [2].

ESP32 cân bằng tốt giữa chi phí – hiệu năng – tính năng, rất phù hợp cho các ứng dụng IoT vừa và nhỏ, trong khi Raspberry Pi thường dùng cho các hệ thống cần xử lý phức tạp hơn [3].

b. Ứng dụng của ESP32 trong các hệ thống IoT

ESP32 được ứng dụng rộng rãi trong:

- Nhà thông minh: điều khiển đèn, quạt, camera.
- Nông nghiệp thông minh: giám sát nhiệt độ, độ ẩm, điều khiển tưới tiêu.
- Các hệ thống cảnh báo, giám sát từ xa.
- Thiết bị đeo thông minh, hệ thống theo dõi sức khỏe cá nhân.

2. Máy chủ web cục bộ trên ESP32

a. Khái niệm về máy chủ web và ứng dụng trong IoT

Máy chủ web (web server) là một hệ thống phần mềm cho phép gửi nội dung web (HTML, CSS, JS...) đến trình duyệt của người dùng thông qua giao thức HTTP. Trong hệ thống IoT, máy chủ web giúp người dùng:

- Truy cập, giám sát trạng thái thiết bị.
- Thực hiện điều khiển từ xa thông qua trình duyệt.
- Giao tiếp với thiết bị mà không cần ứng dụng bên thứ ba.

b. Cách hoạt động của máy chủ web cục bộ trên ESP32

ESP32 có thể hoạt động như một máy chủ HTTP nhỏ gọn. Khi người dùng truy cập vào địa chỉ IP của ESP32 trong mạng nội bộ:

- ESP32 nhận request HTTP từ trình duyệt.
- Dữ liệu HTML/CSS/JS được gửi từ SPIFFS tới trình duyệt.
- Người dùng tương tác qua web, các thao tác như bật/tắt thiết bị được gửi lại đến ESP32 để xử lý [4].

3. Lưu trữ và quản lý giao diện web với SPIFFS

a. Giới thiệu SPIFFS (SPI Flash File System)

SPIFFS là một hệ thống file nhẹ dành cho vi điều khiển không có hệ điều hành, cho phép lưu trữ các file dạng văn bản, như HTML, CSS, JavaScript trên bộ nhớ Flash của ESP32 [5].

b. Cách ESP32 sử dụng SPIFFS để lưu trữ file HTML, CSS, JavaScript

Quá trình sử dụng SPIFFS thường gồm các bước:

1. Thiết kế giao diện web bằng HTML, CSS, JS.
2. Dùng công cụ ESP32 Sketch Data Upload (nếu dùng Arduino IDE) để nạp các file web vào bộ nhớ Flash của ESP32.
3. ESP32 sẽ truy xuất các file này giống như truy cập file từ hệ điều hành.
4. Khi có request từ trình duyệt, ESP32 sẽ gửi file tương ứng về như một máy chủ web thực thụ [6].

4. Giao thức truyền thông

a. HTTP: Xử lý request từ trình duyệt web

HTTP được dùng để nhận và phản hồi các truy vấn từ người dùng. Khi một nút được bấm trên giao diện, trình duyệt sẽ gửi request (thường là GET hoặc POST) tới ESP32 để thực hiện hành động tương ứng.

b. WebSocket: Cập nhật trạng thái thiết bị theo thời gian thực

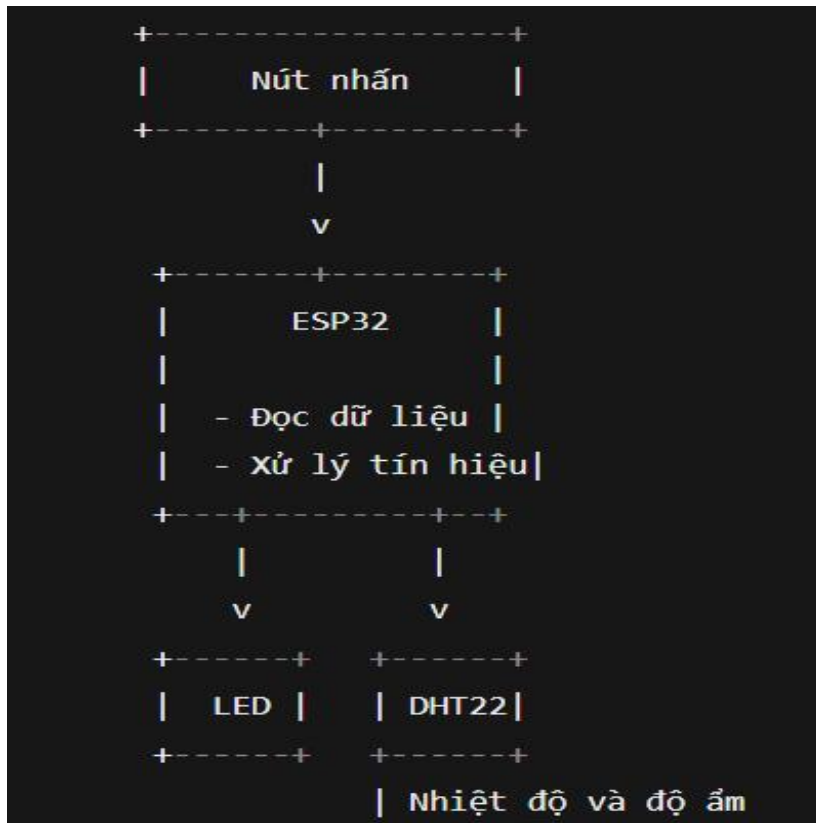
WebSocket thiết lập kết nối hai chiều giữa ESP32 và trình duyệt. Thay vì liên tục gửi request như HTTP, WebSocket giúp thiết bị tự động gửi trạng thái mới cho client mỗi khi có thay đổi. Điều này giúp phản hồi nhanh và mượt hơn nhiều [7].

c. AJAX: Gửi request từ web client đến ESP32 mà không cần tải lại trang

AJAX (Asynchronous JavaScript and XML) cho phép gửi request HTTP ngầm từ trang web đến ESP32 mà không cần reload trang. AJAX thường được dùng để lấy dữ liệu cảm biến hoặc gửi lệnh điều khiển một cách mượt mà.

5. Thiết kế hệ thống

a. Phần cứng

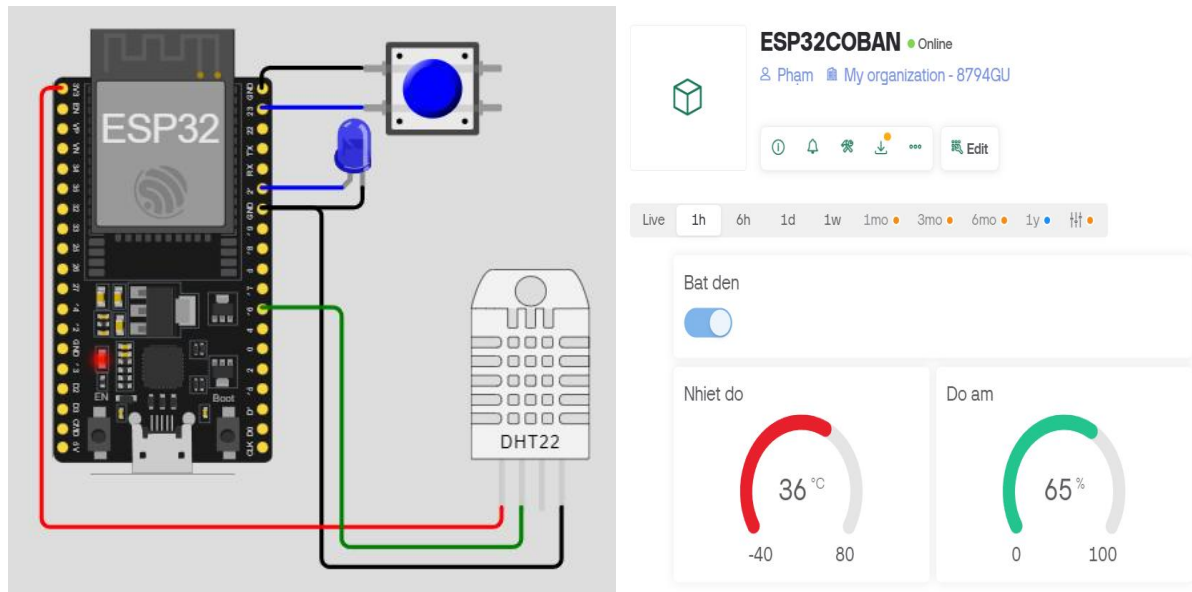


Giải thích sơ đồ khối trên ta có :

- Nút nhấn: Gửi tín hiệu vào ESP32 khi được nhấn (gửi HIGH hoặc LOW).
- ESP32: Nhận tín hiệu từ nút nhấn , Đọc dữ liệu nhiệt độ và độ ẩm từ DHT22 , Có thể điều khiển bật/tắt LED tùy theo điều kiện (ví dụ: nếu nhiệt độ quá cao, hoặc khi nhấn nút).
- LED: Hiển thị trạng thái (có thể báo nhiệt độ vượt ngưỡng, hoặc trạng thái hệ thống).
- Cảm biến DHT22: Cung cấp dữ liệu nhiệt độ và độ ẩm cho ESP32.

b. Phần mềm, Mô phỏng Wokwi.

Kết nối ESP32 với các bộ phận của phần cứng ta có mô phỏng như sau (đèn led đại diện cho 1 thiết bị IoT bất kì)



Nguyên lý hoạt động của bản mô phỏng trên:

Bước 1 : Khởi động hệ thống:

- Khi cấp nguồn, ESP32 khởi tạo các thiết bị ngoại vi: nút nhấn, LED và cảm biến DHT22.
- DHT22 được cấu hình để gửi dữ liệu nhiệt độ và độ ẩm.
- LED được cấu hình là ngõ ra để hiển thị trạng thái.
- Nút nhấn là ngõ vào số, được cấu hình để phát hiện sự kiện nhấn.

Bước 2 : Đọc dữ liệu từ DHT22:

- ESP32 gửi tín hiệu yêu cầu DHT22 truyền dữ liệu.
- Cảm biến DHT22 phản hồi và gửi về giá trị nhiệt độ và độ ẩm.
- ESP32 xử lý dữ liệu này và có thể in ra serial monitor (trong thực tế là hiển thị lên màn hình hoặc gửi về server nếu cần).

Bước 3 : Tương tác với nút nhấn:

- Khi người dùng nhấn nút, ESP32 sẽ phát hiện tín hiệu thay đổi ở chân đầu vào (ví dụ: từ HIGH sang LOW).

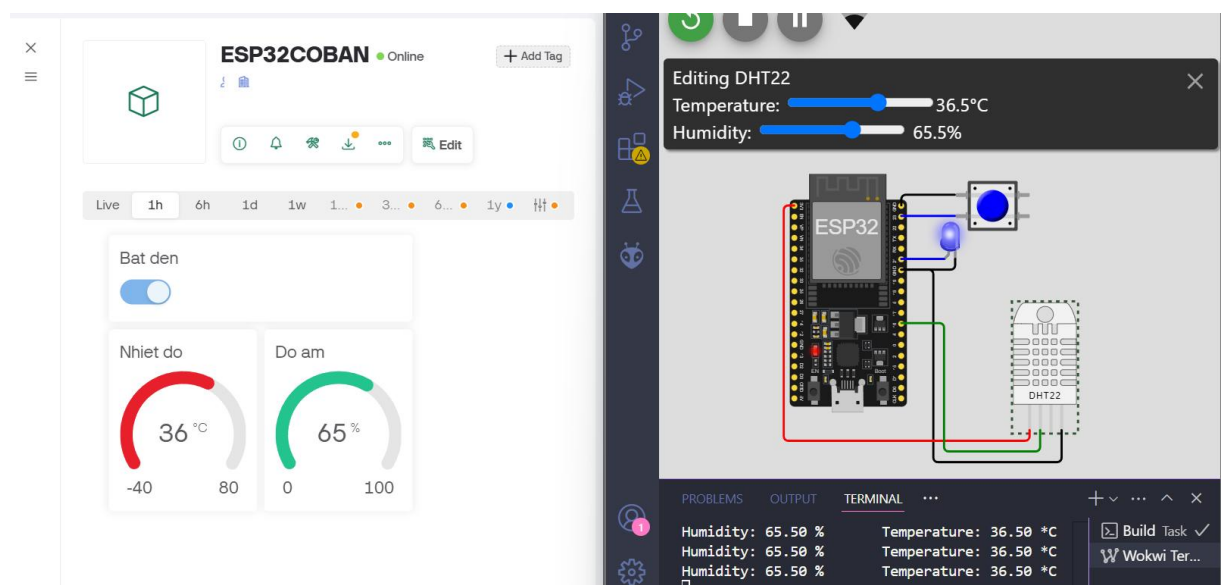
- Khi đó, ESP32 có thể: Bật/tắt LED, Kích hoạt đọc lại cảm biến, Gửi dữ liệu đi (nếu có kết nối Wi-Fi), Thực hiện hành động khác tùy theo logic lập trình.

Bước 5 : Điều khiển LED

- LED được bật/tắt dựa vào: Trạng thái nút nhấn, Ngưỡng nhiệt độ/độ ẩm (ví dụ: nếu nhiệt độ > 30°C thì bật LED cảnh báo), Trạng thái hệ thống.

c. Gửi dữ liệu lên nền tảng IoT

Thông qua Blynk để điều khiển thiết bị theo mong muốn như bật , tắt , điều chỉnh nhiệt độ ,vv



6. Kết quả và đánh giá

* Ưu điểm : Máy chủ web cục bộ giúp quản lý thiết bị mà không cần internet , Giao diện web trực quan, dễ sử dụng , Tiết kiệm chi phí so với các giải pháp IoT dựa trên cloud.

* Hạn chế : Phạm vi hoạt động giới hạn trong mạng nội bộ, Cần thiết lập mạng Wi-Fi nếu không có router.

7. Kết luận và hướng phát triển

a. Kết luận

- Xây dựng thành công máy chủ web trên ESP32 để quản lý thiết bị IoT.
- Hệ thống hoạt động ổn định và có thể mở rộng với nhiều thiết bị hơn.

b. Hướng phát triển

- Bảo mật: Thêm xác thực đăng nhập và mã hóa dữ liệu.
- Tích hợp cloud: Kết hợp với Firebase hoặc MQTT để điều khiển từ xa qua internet.
- Cải thiện giao diện: Sử dụng Vue.js hoặc React để tạo giao diện web động hơn.
- Thêm chức năng tự động hóa: Ví dụ, bật/tắt thiết bị dựa trên cảm biến hoặc lịch trình.

TÀI LIỆU THAM KHẢO

[1] Official SPIFFS documentation

- [2] Espressif - Giới thiệu ESP32: <https://www.espressif.com/en/products/socs/esp32>
- [3] So sánh ESP32, ESP8266 và Raspberry Pi: <https://randomnerdtutorials.com/esp32-vs-esp8266-vs-raspberry-pi/>
- [4] ESP32 Web Server Tutorial: <https://randomnerdtutorials.com/esp32-web-server-arduino-ide/>
- [5] SPIFFS: <https://github.com/espressif/arduino-esp32/blob/master/docs/filesystem.md>
- [6] Hướng dẫn lưu giao diện web bằng SPIFFS:
<https://randomnerdtutorials.com/install-esp32-filesystem-uploader-arduino-ide/>
- [7] WebSocket với ESP32: <https://randomnerdtutorials.com/esp32-websocket-server-arduino>.