DANH MỤC CÁC TỪ VIẾT TẮT

IoT: Internet of Things (Internet vạn vật)

ESP32: Espressif Systems 32 (Tên vi điều khiển)

SPIFFS: SPI Flash File System (Hệ thống tệp flash SPI)

DHT: Digital Humidity and Temperature (Cảm biến nhiệt độ và độ ẩm)

LDR: Light Dependent Resistor (Cảm biến ánh sáng)

PIR: Passive Infrared Sensor (Cảm biến chuyển động hồng ngoại thụ động)

HTTP: HyperText Transfer Protocol (Giao thức truyền siêu văn bản)

API: Application Programming Interface (Giao diện lập trình ứng dụng)

MỤC LỤC

PHẦN MỞ ĐẦU

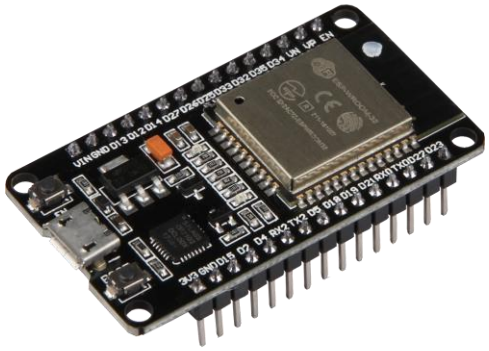
Trong thời đại Công nghệ 4.0, Internet vạn vật (IoT) đã trở thành nền tảng quan trọng kết nối hàng tỷ thiết bị thông minh phục vụ cuộc sống con người. Từ hệ thống chiếu sáng tự động hóa nhà thông minh đến cảm biến giám sát môi trường nông nghiệp, Internet of Things không chỉ mang đến sự tiện lợi mà còn mở ra tiềm năng sáng tạo không giới hạn. Sự xuất hiện của các bộ vi điều khiển nhỏ gọn, tích hợp Wi-Fi, chi phí thấp như ESP32 đưa IoT đến gần hơn với các ứng dụng thực tế. Đề tài "Tạo web server cục bộ sử dụng ESP32 để quản lý thiết bị IoT" được chọn vì nhu cầu xây dựng hệ thống IoT độc lập không yêu cầu kết nối internet liên tục - giải pháp phù hợp cho các ứng dụng gia đình hoặc văn phòng nhỏ. Với Wi-Fi tích hợp và sức mạnh xử lý mạnh mẽ, ESP32 là lựa chọn tốt nhất để triển khai máy chủ web cục bộ. Bên cạnh đó, các công cụ mô phỏng của Wokwi giúp nhanh chóng thử nghiệm các ý tưởng mà không cần đầu tư phần cứng ban đầu, khiến chúng trở nên lý tưởng cho các mục tiêu học tập và nghiên cứu về phát triển ứng dụng IoT. Mục tiêu của bài viết này là thiết kế hệ thống IoT sử dụng ESP32 làm máy chủ web cục bộ, tích hợp SPIFFS để lưu trữ giao diện web nhằm quản lý và giám sát các thiết bị trong mạng cục bộ. Đồng thời, hệ thống sẽ được mô phỏng trên Wokwi thông qua các cảm biến như DHT22, LDR, PIR và dữ liệu sẽ được gửi đến các nền tảng như ThingSpeak, Blynk hoặc Telegram để hiển thị hoặc thông báo. Nội dung tập trung vào lý thuyết và bản vẽ thiết kế. Để phát triển chủ đề này, phương pháp nghiên cứu bao gồm nghiên cứu tài liệu về hệ thống tệp ESP32, SPIFFS và các cảm biến IoT phổ biến. Sau đó, hệ thống được thiết kế và mô phỏng trên Wokwi thông qua sơ đồ kết nối phần cứng. Bài viết này tập trung phân tích lý thuyết về cách ESP32 vận hành máy chủ mạng cục bộ và gửi dữ liệu lên nền tảng để đảm bảo tính logic và khoa học.

PHẦN NỘI DUNG

I. Cơ sở lý thuyết

1. IoT và ESP32

- Internet of Things (IoT) là hệ sinh thái công nghệ kết nối các thiết bị vật lý qua mạng để thu thập, xử lý và chia sẻ dữ liệu, từ đó nâng cao hiệu quả trong nhiều lĩnh vực như nhà thông minh (điều khiển ánh sáng tự động), nông nghiệp (theo dõi độ ẩm đất) hay y học (theo dõi sức khỏe từ xa).
- Được phát triển bởi Espressif Systems, ESP32 là bộ vi điều khiển 32 bit với Tensilica lõi kép. Thiết bị có thể hoạt động ở chế độ điểm truy cập (AP) để tạo mạng Wi-Fi riêng hoặc ở chế độ máy trạm (STA) để kết nối với mạng Wi-Fi hiện có, mang lại sự linh hoạt cho việc triển khai các hệ thống IoT cục bộ.



Hình 1.1.1 Bo mạch ESP32 DevKit V1

- Chỉ tốn khoảng 5-10 USD, có cộng đồng nguồn mở lớn và khả năng xử lý mạnh mẽ, ESP32 là lựa chọn tốt nhất để xây dựng máy chủ web cục bộ nhằm quản lý các thiết bị IoT mà không cần dựa vào kết nối Internet liên tục, phù hợp với mục tiêu của chủ đề này.

2. Máy chủ web cục bộ

- Máy chủ web là một hệ thống nhận và xử lý các yêu cầu HTTP từ máy khách (chẳng hạn như trình duyệt) thông qua giao thức GET (nhận dữ liệu) hoặc POST (gửi dữ liệu), sau đó trả về phản hồi dưới dạng trang HTML hoặc dữ liệu JSON.
- Trong hệ thống này, ESP32 được cấu hình như một máy chủ web cục bộ, sử dụng thư viện ESPAsyncWebServer để xử lý các yêu cầu không đồng bộ, giảm

độ trễ so với WebServer truyền thống. Ví dụ: người dùng có thể truy cập các địa chỉ IP như 192.168.1.1 thông qua trình duyệt mạng cục bộ để xem trạng thái thiết bị hoặc gửi lệnh điều khiển.

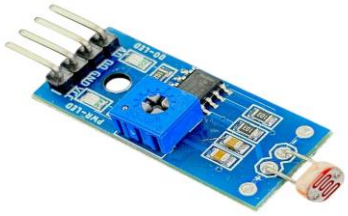
- Để lưu trữ giao diện web, SPIFFS (Hệ thống tệp Flash SPI) được sử dụng, sử dụng bộ nhớ flash 4 MB của ESP32 để lưu trữ các tệp tĩnh như HTML (giao diện), CSS (định dạng) và JavaScript (tương tác).
- SPIFFS được tích hợp sẵn và không yêu cầu phần cứng bổ sung, cho phép hệ thống chạy độc lập mà không cần kết nối internet.
- Ưu điểm của giải pháp này là khả năng chạy trong mạng nội bộ có độ an toàn cao và giao diện nhẹ (thường dưới 1 MB), đảm bảo tốc độ tải nhanh và phù hợp với các ứng dụng IoT nhỏ.

3. Cảm biến IoT

- Cảm biến là thành phần cốt lõi để thu thập dữ liệu trong hệ thống IoT. Cảm biến DHT22 đo nhiệt độ từ -40-80°C với độ chính xác ±2°C và độ ẩm từ 0-100% sử dụng giao thức 1-Dây để truyền tín hiệu đến ESP32 thông qua một chân dữ liệu duy nhất.

Hình 1.3.1 Module cảm biến nhiệt độ, độ ẩm DHT22 AM2302

- LDR (Light Dependent Resistor) hoạt động theo nguyên tắc điện trở giảm khi ánh sáng tăng và được kết nối với chân ADC của ESP32 để đo các giá trị từ 0 (bóng tối hoàn toàn) đến 4095 (đèn sáng), chẳng hạn để kiểm tra ánh sáng trong phòng.



Hình 1.3.2 Light Dependent Resistor Module

- Cảm biến PIR (Passive Infrared) phát hiện chuyển động trong phạm vi 5-7m bằng cách thu nhận tia hồng ngoại phát ra từ cơ thể con người và phát ra tín hiệu cao (3,3V) khi có sự kiện xảy ra, chẳng hạn như phát hiện có người vào phòng.

Hình 1.3.3 PIR Sensor SR501

- Dữ liệu từ các cảm biến này được ESP32 thu thập và xử lý để hiển thị trên giao diện web hoặc gửi đến nền tảng bên ngoài.

4. Nền tảng giao tiếp

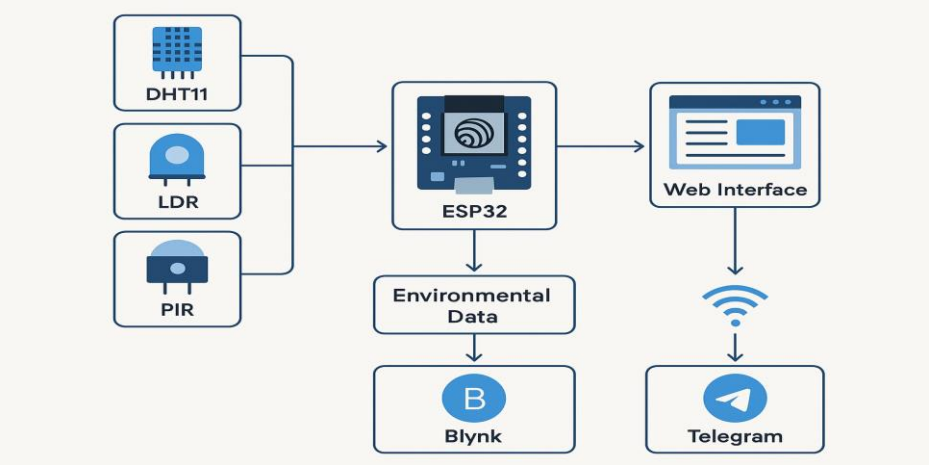
- Để truyền dữ liệu từ ESP32 đến người dùng, các nền tảng giao tiếp đóng vai trò quan trọng.
- ThingSpeak là dịch vụ lưu trữ dữ liệu IoT miễn phí, hỗ trợ tối đa 8 trường (field) cho mỗi kênh, cho phép ESP32 gửi các giá trị như nhiệt độ (Field 1), độ ẩm (Field 2) qua yêu cầu HTTP POST đến API, sau đó hiển thị dưới dạng biểu đồ trực quan để theo dõi xu hướng môi trường.

- Blynk cung cấp ứng dụng di động thân thiện, kết nối với ESP32 qua Auth Token, hỗ trợ các widget như Gauge để hiển thị nhiệt độ hoặc Button để bật/tắt LED từ xa, mang lại trải nghiệm điều khiển thời gian thực.
- Telegram, thông qua bot được tạo bằng BotFather, cho phép gửi thông báo tức thì như 'Motion detected!' khi PIR phát hiện chuyển động, sử dụng yêu cầu HTTP GET với Bot Token và Chat ID để nhắn tin đến người dùng.
- Tất cả các nền tảng này dựa vào khả năng Wi-Fi của ESP32, gửi dữ liệu qua giao thức HTTP đơn giản, trong Wokwi mô phỏng bằng mạng 'Wokwi-GUEST'. Giải pháp này tăng tính linh hoạt, từ theo dõi dữ liệu dài hạn (ThingSpeak) đến điều khiển tức thời (Blynk) và cảnh báo nhanh (Telegram).

II. Thiết kế hệ thống

1. Kiến trúc hệ thống

- Kiến trúc hệ thống được thiết kế với ESP32 đóng vai trò trung tâm, vận hành máy chủ web cục bộ để quản lý các thiết bị IoT trong mạng nội bộ.
- Các cảm biến như DHT22, LDR, và PIR thu thập dữ liệu môi trường (nhiệt độ, ánh sáng, chuyển động), gửi đến ESP32 để xử lý. ESP32 sau đó hiển thị thông tin qua giao diện web lưu trên SPIFFS hoặc truyền dữ liệu đến các nền tảng như ThingSpeak (biểu đồ), Blynk (điều khiển), hoặc Telegram (thông báo).



Hình 2.1.1 Sơ đồ luồng dữ liệu

- Luồng dữ liệu cơ bản bao gồm: cảm biến gửi tín hiệu đến ESP32, ESP32 xử lý và phản hồi qua giao diện web hoặc gửi đi qua Wi-Fi. Hệ thống hoạt động trong mạng cục bộ, với ESP32 có thể cấu hình ở chế độ Access Point (AP) để tạo

mạng riêng hoặc Station (STA) để kết nối vào mạng Wi-Fi có sẵn, đảm bảo tính độc lập và linh hoạt mà không cần internet liên tục.

2. Thành phần phần cứng
   - Hệ thống sử dụng ESP32 DevKit V1 làm trung tâm điều khiển, với khả năng Wi-Fi tích hợp và nhiều chân GPIO hỗ trợ kết nối cảm biến và thiết bị.
   - Các cảm biến bao gồm DHT22 để đo nhiệt độ (-40°C đến +80°C) và độ ẩm (0-100%), kết nối qua GPIO 4; LDR để đo cường độ ánh sáng (0-4095), kết nối qua GPIO 34 (ADC); và PIR để phát hiện chuyển động trong phạm vi 5-7m, kết nối qua GPIO 13. Ngoài ra, một LED được gắn vào GPIO 2 qua điện trở 220Ω để mô phỏng thiết bị điều khiển, ví dụ bật/tắt đèn dựa trên dữ liệu cảm biến.
   - Tất cả các thành phần này đều có sẵn trong Wokwi, cho phép mô phỏng chính xác mà không cần phần cứng thực tế.
   - Thiết kế này linh hoạt, có thể thay thế hoặc bổ sung cảm biến khác tùy theo nhu cầu ứng dụng.
3. Thành phần phần mền
   - Phần mềm đóng vai trò hỗ trợ triển khai và mô phỏng hệ thống. SPIFFS (SPI Flash File System) được sử dụng để lưu trữ các tệp giao diện web, bao gồm HTML (cấu trúc trang), CSS (định dạng), và JavaScript (tương tác), tận dụng bộ nhớ flash 4MB của ESP32 để cung cấp giao diện quản lý thiết bị mà không cần máy chủ bên ngoài.
   - Wokwi, một công cụ mô phỏng trực tuyến, cho phép thiết kế và kiểm tra hệ thống với ESP32 và các cảm biến như DHT22, LDR, PIR thông qua giao diện kéo-thả.
   - Trong mô phỏng, các thư viện như ESPAsyncWebServer được giả định để xử lý máy chủ web không đồng bộ, cùng với DHT.h và các thư viện khác để đọc dữ liệu cảm biến.
   - Mạng Wi-Fi trong Wokwi được cung cấp qua 'Wokwi-GUEST', mô phỏng kết nối cục bộ mà không cần internet thực tế, phù hợp với mục tiêu thiết kế hệ thống độc lập.

4. Luồng hoạt động

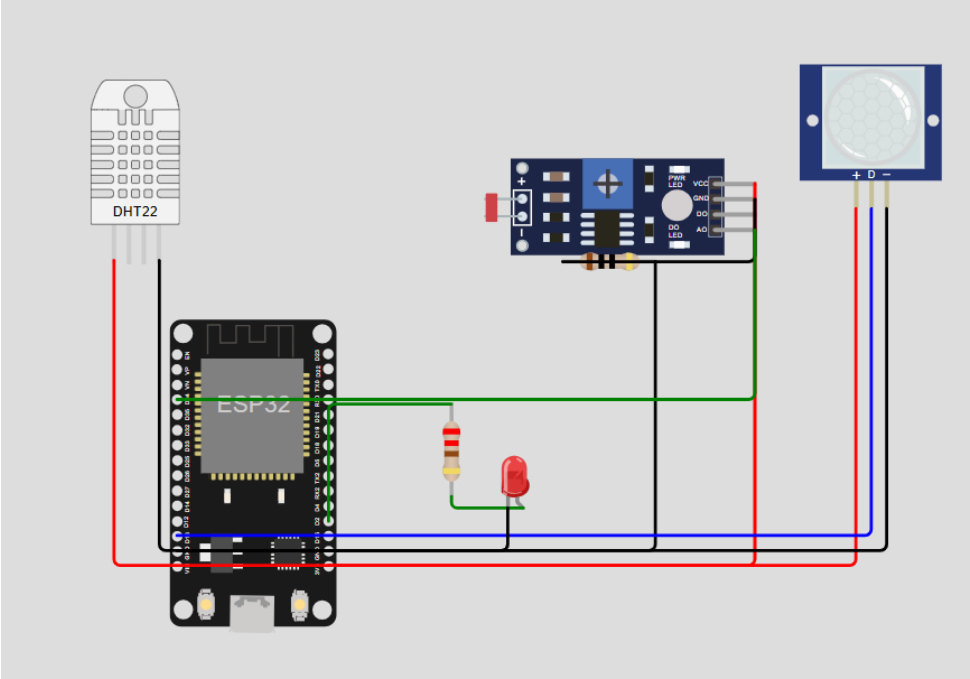
- Luồng hoạt động của hệ thống được thiết kế như sau:
  - Đầu tiên, các cảm biến thu thập dữ liệu môi trường – DHT22 đo nhiệt độ và độ ẩm, LDR đo ánh sáng, PIR phát hiện chuyển động – và gửi tín hiệu đến ESP32 qua các chân GPIO tương ứng (GPIO 4, 34, 13).
  - Thứ hai, ESP32 xử lý dữ liệu này, ví dụ chuyển tín hiệu analog từ LDR thành giá trị số (0-4095) hoặc đọc tín hiệu HIGH từ PIR, sau đó lưu trữ tạm thời trong bộ nhớ.
  - Thứ ba, dữ liệu được sử dụng theo hai cách: hiển thị trực tiếp trên giao diện web lưu trong SPIFFS (người dùng truy cập qua IP như 192.168.1.1) hoặc gửi đến các nền tảng bên ngoài qua Wi-Fi – ThingSpeak cập nhật biểu đồ nhiệt độ/độ ẩm, Blynk hiển thị dữ liệu và điều khiển LED, Telegram gửi thông báo như 'Motion detected!' khi PIR kích hoạt.
- Ví dụ, nếu PIR phát hiện chuyển động, ESP32 sẽ gửi yêu cầu HTTP đến bot Telegram để cảnh báo người dùng tức thì. Luồng này đảm bảo hệ thống hoạt động linh hoạt và hiệu quả trong mạng cục bộ.

III. Mô phỏng trên Wokwi

1. Sơ đồ kết nối

- Sơ đồ kết nối trên nền tảng Wokwi được xây dựng với trung tâm điều khiển là ESP32 DevKit V1, đóng vai trò chính trong vận hành hệ thống. Các thành phần cảm biến và thiết bị ngoại vi được liên kết như sau:
  - DHT22: Cảm biến đo nhiệt độ và độ ẩm, kết nối chân DATA với GPIO 4, chân VCC với nguồn 3.3V, và chân GND với GND.
  - LDR: Cảm biến ánh sáng, một chân được nối qua điện trở 10kΩ tới GND, chân còn lại kết nối với GPIO 34 (ADC) và nguồn 3.3V.
  - PIR: Cảm biến phát hiện chuyển động, chân OUT gắn vào GPIO 13, chân VCC nối với nguồn 3.3V, và chân GND nối với GND.
  - LED đỏ: Mô phỏng thiết bị điều khiển, được nối từ GPIO 2 tới một điện trở 220Ω (chân Anode), còn chân Cathode kết nối với GND.

– Các linh kiện này có thể dễ dàng thêm từ thư viện trên Wokwi bằng cách nhấn dấu "+", sau đó kéo thả các thành phần vào khu vực làm việc. Khi thực hiện kết nối dây, cần đảm bảo sử dụng các GPIO khác nhau cho từng thiết bị nhằm tránh xung đột tín hiệu.



Hình 3.1.1 Sơ đồ kết nối trên nền tảng Wokwi

2. Mô phỏng cảm biến

– Trong Wokwi, các cảm biến được thiết kế để mô phỏng dữ liệu giả lập, giúp tái hiện hoạt động thực tế của chúng. Cảm biến DHT22 giả lập nhiệt độ trong khoảng từ 20-30°C và độ ẩm từ 40-80%, nằm trong phạm vi hoạt động thực tế (-40°C đến +80°C, 0-100%), và gửi tín hiệu qua GPIO 4. Cảm biến LDR tạo dữ liệu phản ánh cường độ ánh sáng, với giá trị từ 0 (tối) đến 4095 (sáng mạnh), dưới dạng tín hiệu analog qua GPIO 34. Cảm biến PIR mô phỏng tín hiệu chuyển động bằng cách xuất mức HIGH (3.3V) khi phát hiện chuyển động và mức LOW khi không, truyền qua GPIO 13.

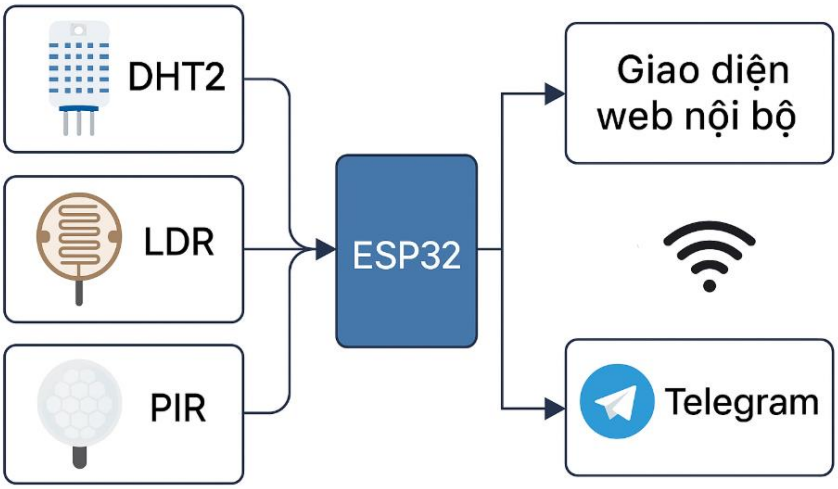
– Trong quá trình chạy mô phỏng trên Wokwi, các giá trị này được tự động tạo ra thông qua nút "Start Simulation". Tuy nhiên, giao diện không hiển thị trực tiếp các số liệu, mà tập trung mô phỏng hoạt động của cảm biến trong kết nối phần

cứng. Nếu có nhu cầu thêm các cảm biến chưa được hỗ trợ trên Wokwi, chẳng hạn như BMP180 dùng để đo áp suất, bạn có thể giả lập các giá trị ngẫu nhiên. Ví dụ, áp suất có thể được cung cấp trong phạm vi từ 900-1100 hPa để đảm bảo sự linh hoạt trong thiết kế hệ thống và thử nghiệm phần mềm.

- Cách tiếp cận này hướng tới việc kiểm tra tương tác phần cứng giữa các thiết bị hơn là dữ liệu thực tế, từ đó hỗ trợ việc phát triển và tối ưu hóa hệ thống một cách dễ dàng và hiệu quả.

3. Gửi dữ liệu

- Trong môi trường mô phỏng Wokwi, ESP32 được thiết lập để giả lập việc gửi dữ liệu tới giao diện web và các nền tảng bên ngoài thông qua mạng Wi-Fi "Wokwi-GUEST". Hệ thống hoạt động theo các bước sau:
    - Giao diện web, lưu trữ trên SPIFFS, dùng để hiển thị các thông số như nhiệt độ, độ ẩm và cường độ ánh sáng qua một địa chỉ IP riêng. Người dùng có thể truy cập giao diện này từ trình duyệt trong cùng mạng nội bộ.
    - Nền tảng ThingSpeak nhận dữ liệu nhiệt độ và độ ẩm từ ESP32 thông qua yêu cầu HTTP POST. Các giá trị (Field 1: 25°C, Field 2: 60%) được cập nhật để tạo biểu đồ trực quan hóa dữ liệu.
    - Ứng dụng Blynk hiển thị thông số theo thời gian thực trên thiết bị di động, sử dụng Auth Token để kết nối và đồng thời điều khiển LED trên GPIO 2 thông qua các lệnh bật/tắt.
    - Telegram được sử dụng để gửi cảnh báo, chẳng hạn "Motion detected!", khi cảm biến PIR phát hiện chuyển động (lúc tín hiệu chuyển sang HIGH), thông qua yêu cầu HTTP GET



Hình 3.3.1 Hình ảnh mô phỏng gửi đến nền tảng thứ ba là Telegram

– Lưu ý rằng Wokwi chỉ mô phỏng hoạt động của phần cứng và mạng Wi-Fi, không thực hiện các kết nối thực tế đến các nền tảng trên. Toàn bộ quá trình truyền và hiển thị dữ liệu được giả định dựa vào nguyên lý hoạt động dự đoán của ESP32, đảm bảo mô hình hóa khả thi cho thiết kế hệ thống.

PHẦN KẾT LUẬN

- Để tối ưu hóa hiệu quả của hệ thống, cần xem xét một số giải pháp cải tiến quan trọng. Trước hết, việc triển khai hệ thống trên phần cứng thực tế dựa trên ESP32 có thể cung cấp đánh giá chính xác về hiệu suất và giúp khắc phục những hạn chế hiện tại của môi trường giả lập Wokwi. Bên cạnh đó, thay thế bộ nhớ SPIFFS bằng bộ nhớ ngoài như thẻ SD sẽ mang lại khả năng lưu trữ tốt hơn cho các giao diện web có tính phức tạp cao, từ đó mở rộng khả năng ứng dụng.
- Đặc biệt, việc tích hợp giao thức MQTT sẽ tạo điều kiện quản lý đồng thời nhiều thiết bị IoT trong hệ thống, đáp ứng nhu cầu của các mô hình lớn hơn như nhà thông minh hoặc nông nghiệp thông minh. Trong các hướng nghiên cứu tương lai, có thể cân nhắc đến việc sử dụng các loại cảm biến phức tạp như camera hoặc thử nghiệm vận hành hệ thống trên mạng diện rộng kết hợp công nghệ điện toán đám mây. Những cải tiến này không chỉ tăng tính thực tiễn mà còn mang lại giá trị lâu dài, mở rộng tiềm năng ứng dụng cho các dự án IoT trong tương lai.

TÀI LIỆU THAM KHẢO

1 Espressif Systems. (2023). *ESP32 Technical Reference Manual*. https://www.espressif.com

2 Wokwi. (2024). *ESP32 Simulation Platform*. Truy cập tại: https://wokwi.com

3 Arduino Project Hub. (n.d.). *ESP32 Web Server with SPIFFS*. Truy cập tại: https://create.arduino.cc/projecthub

4 ThingSpeak. (n.d.). *Data Collection and Visualization Platform for IoT*. Truy cập tại: https://thingspeak.com

5 Blynk IoT Platform. (2024). *Documentation for ESP32 Devices*. Truy cập tại: https://docs.blynk.io

6 Telegram Bot API. (n.d.). *Official Bot API Documentation*. Truy cập tại: https://core.telegram.org/bots/api

7 Random Nerd Tutorials. (2023). *ESP32 Web Server Projects and Sensor Readings*. Truy cập tại: https://randomnerdtutorials.com