

**TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ PHẦN MỀM**



**TIỂU LUẬN HỌC PHẦN PHÁT TRIỂN ỨNG DỤNG IOT
ĐỀ TÀI: TẠO MÁY CHỦ WEB CỤC BỘ VỚI ESP32 ĐỂ
QUẢN LÝ THIẾT BỊ IOT
NHÓM 4**

1. DANH MỤC CÁC TỪ VIẾT TẮT

Từ Viết Tắt	Giải Nghĩa
IoT	Internet of Things - Internet Vạn Vật
ESP32	Vi điều khiển tích hợp WiFi/Bluetooth
SPIFFS	SPI Flash File System
HTTP	Hypertext Transfer Protocol
GUI	Giao diện đồ họa người dùng
API	Giao diện lập trình ứng dụng
OTA	Over The Air (nâng cấp từ xa)
STA	Station Mode
AP	Access Point
Blynk	Nền tảng phát triển ứng dụng IoT
JSON	JavaScript Object Notation
PWM	Pulse Width Modulation

2. MỤC LỤC

1. Danh mục từ viết tắt

2. Mục lục

3. Mở đầu

4. Nội dung

- **4.1 Tổng quan về IoT và ESP32**
- **4.2 Máy chủ Web cục bộ và ứng dụng trong IoT**
- **4.3 Giao diện Web bằng SPIFFS**
- **4.4 Mô phỏng thiết bị trên Wokwi**
- **4.5 Tương tác người dùng qua giao diện Web**
- **4.6 Tích hợp Blynk hoặc ThingSpeak**
- **4.7 Khả năng mở rộng và ứng dụng thực tiễn**

5. Kết luận và kiến nghị

6. Tài liệu tham khảo

3. MỞ ĐẦU:

Sự phát triển mạnh mẽ của công nghệ thông tin và vi điện tử trong những năm gần đây đã đưa Internet vạn vật (IoT – Internet of Things) trở thành một xu hướng tất yếu trong quá trình hiện đại hóa, tự động hóa các lĩnh vực của đời sống. IoT không chỉ mang lại sự tiện lợi cho người dùng mà còn mở ra cơ hội to lớn trong việc tối ưu hóa vận hành, giám sát và điều khiển thiết bị từ xa thông qua kết nối mạng. [1]

Một trong những nền tảng phần cứng được sử dụng rộng rãi trong các dự án IoT hiện nay là **ESP32** – một vi điều khiển (microcontroller) tích hợp WiFi và Bluetooth, do Espressif Systems phát triển. Với khả năng xử lý mạnh mẽ, hỗ trợ đa dạng giao thức truyền thông, cộng thêm mức giá rẻ và tài liệu phát triển phong phú, ESP32 đang dần trở thành lựa chọn hàng đầu trong nhiều ứng dụng thực tiễn như: nhà thông minh, nông nghiệp công nghệ cao, giám sát môi trường, thiết bị đeo tay, và nhiều hệ thống tự động hóa khác. [2]

Bên cạnh khả năng kết nối Internet, ESP32 còn cho phép thiết lập **máy chủ web nội bộ (local web server)** hoạt động hoàn toàn độc lập, không cần phụ thuộc vào hạ tầng mạng bên ngoài. Điều này đặc biệt hữu ích trong các trường hợp mạng không ổn định, môi trường hạn chế kết nối Internet, hoặc người dùng chỉ cần một hệ thống điều khiển cục bộ với chi phí thấp và dễ triển khai. [3]

Trong đề tài này, trình bày quá trình xây dựng một hệ thống IoT đơn giản sử dụng ESP32, kết hợp với cảm biến môi trường và giao diện web thân thiện được lưu trữ trên bộ nhớ SPIFFS của vi điều khiển. Thông qua mạng WiFi nội bộ do ESP32 phát ra (WiFi Access Point mode), người dùng có thể kết nối trực tiếp và thực hiện các thao tác như: bật/tắt đèn, theo dõi nhiệt độ, độ ẩm môi trường, ... ngay trên trình duyệt của điện thoại hoặc máy tính mà không cần đến kết nối Internet.

Phương pháp tiếp cận này không chỉ giúp giảm thiểu chi phí phần cứng và vận hành, mà còn tạo điều kiện thuận lợi cho người mới bắt đầu làm quen với IoT, sinh viên kỹ thuật hoặc những người yêu thích công nghệ tự chế tạo và triển khai các hệ thống thông minh trong phạm vi cá nhân hoặc gia đình.

Qua đề tài mong muốn làm rõ tiềm năng ứng dụng thực tế của ESP32 trong việc xây dựng các hệ thống IoT đơn giản, đồng thời cung cấp nền tảng kiến thức cơ bản để phát triển các giải pháp mở rộng trong tương lai như kết nối nhiều thiết bị, thu thập dữ liệu thời gian thực, hoặc tích hợp vào hệ thống giám sát trung tâm.

4. NỘI DUNG

4.1. Tổng quan về IoT và ESP32

Internet of Things (IoT) – Mạng lưới Vạn vật Kết nối Internet – là một xu hướng công nghệ hiện đại, trong đó các thiết bị vật lý như cảm biến, thiết bị điện tử, máy móc, phương tiện,... có khả năng kết nối và giao tiếp với nhau thông qua mạng Internet. Mục tiêu của IoT là thu thập, trao đổi và xử lý dữ liệu để tự động hóa và tối ưu hóa các hoạt động trong đời sống, sản xuất, và quản lý. [4]

Đặc điểm nổi bật của IoT:

- Tự động hóa: Các thiết bị có thể thực hiện hành động mà không cần sự can thiệp của con người.
- Kết nối: Thiết bị được liên kết qua các giao thức mạng như WiFi, Bluetooth, Zigbee, LoRa,...
- Thông minh: Khả năng thu thập, phân tích dữ liệu giúp hệ thống đưa ra quyết định tối ưu.
- Ứng dụng đa lĩnh vực: Nông nghiệp thông minh, nhà thông minh, y tế, công nghiệp, giao thông,...

ESP32 – Bộ vi điều khiển mạnh mẽ cho IoT:

ESP32 là dòng vi điều khiển do Espressif Systems phát triển, nổi bật với khả năng tích hợp WiFi và Bluetooth ngay trên chip. Đây là giải pháp tối ưu cho các ứng dụng IoT nhờ chi phí thấp, hiệu năng cao, dễ lập trình và có cộng đồng phát triển rộng lớn. [5]

Đặc điểm kỹ thuật tiêu biểu của ESP32: [6]

- CPU: Hai nhân (dual-core) Xtensa® 32-bit LX6, tốc độ xử lý từ 160 đến 240MHz.
- RAM: ~520KB SRAM, đủ dùng cho đa số ứng dụng IoT.
- Bộ nhớ flash: Có thể mở rộng, thường từ 4MB trở lên.
- Tích hợp: WiFi chuẩn 802.11 b/g/n và Bluetooth 4.2 (BLE + BR/EDR).
- Chân GPIO: Lên đến 34 chân, hỗ trợ đa chức năng.

Giao tiếp và giao thức hỗ trợ:

- Giao tiếp phần cứng: UART, SPI, I2C, I2S, PWM, ADC, DAC, CAN, touch sensor,...
- Giao thức mạng: TCP/IP, HTTP/HTTPS, WebSocket, MQTT, mDNS,...
- Lưu trữ tệp: Hỗ trợ SPIFFS và LittleFS – cho phép lưu trữ các file HTML, CSS, JS để tạo giao diện WebServer.
- Cập nhật phần mềm từ xa: OTA (Over-The-Air) giúp cập nhật chương trình mà không cần kết nối vật lý.

4.2. Máy chủ Web cục bộ và ứng dụng

• ESP32 như một Web Server:

Một trong những tính năng nổi bật của ESP32 là khả năng hoạt động như một máy chủ web (Web Server) cục bộ. Điều này cho phép người dùng truy cập và điều khiển thiết bị thông qua trình duyệt web mà không cần máy chủ trung gian hay kết nối Internet.

ESP32 có thể lưu trữ các tệp HTML, CSS, JavaScript ngay trong bộ nhớ flash thông qua hệ thống tệp SPIFFS (SPI Flash File System). Khi có thiết bị truy cập địa chỉ IP của ESP32, Web Server sẽ phản hồi lại bằng các trang giao diện web, cho phép người dùng điều khiển và giám sát các thiết bị IoT một cách trực quan. [7]

• Hai chế độ hoạt động phổ biến:

1. AP Mode (Access Point):

- ESP32 tạo một mạng WiFi riêng biệt.
- Người dùng có thể kết nối trực tiếp đến ESP32 thông qua mạng này, không cần WiFi bên ngoài.
- Thích hợp cho các ứng dụng đơn giản, không cần Internet.

Ví dụ: Điện thoại kết nối đến mạng “ESP32-Device”, sau đó truy cập trang web điều khiển tại 192.168.4.1.

2. STA Mode (Station):

- ESP32 kết nối vào một mạng WiFi hiện có (như mạng trong nhà).
- Thiết bị và người dùng cùng nằm trong một mạng WiFi.

- Cho phép mở rộng chức năng, kết nối Internet, đồng bộ thời gian, gửi dữ liệu lên đám mây, ...

Ví dụ: ESP32 lấy địa chỉ IP nội bộ trong mạng WiFi, có thể truy cập từ các thiết bị khác cùng mạng.

Ngoài ra, ESP32 cũng có thể hoạt động ở chế độ kép (AP + STA) – vừa tạo mạng riêng, vừa kết nối vào mạng WiFi chính.

• Ứng dụng thực tế của Web Server trên ESP32:

1. Nhà thông minh (Smart Home):

- Điều khiển từ xa các thiết bị như đèn, quạt, máy lạnh, rèm cửa,...
- Giao diện đơn giản, dễ sử dụng qua trình duyệt web trên điện thoại hoặc máy tính.
- Có thể kết hợp với cảm biến để tự động hóa (ví dụ: tự bật đèn khi trời tối).

2. Nông nghiệp thông minh:

- Theo dõi nhiệt độ, độ ẩm đất, ánh sáng, ... từ xa.
- Hiển thị dữ liệu trực tiếp trên giao diện web.
- Cảnh báo khi vượt ngưỡng cho phép (gửi qua Telegram, Blynk, ...).

3. Quản lý thiết bị công nghiệp/học tập:

- Giao diện giám sát và điều khiển thiết bị máy móc, thí nghiệm,...
- Thiết lập lịch hoạt động, giám sát trạng thái (bật/tắt, hư hỏng).
- Đồng bộ trạng thái giữa nhiều thiết bị.

4. Một số chức năng mở rộng:

- Cập nhật thời gian thực (real-time) bằng WebSocket.
- Gửi dữ liệu đến nền tảng IoT như Blynk, ThingSpeak.
- Cập nhật phần mềm từ xa (OTA) thông qua trình duyệt.
- Giao diện tùy biến: dùng CSS và JavaScript để tăng tính thân thiện.

4.3. Giao diện Web bằng SPIFFS

- SPIFFS – Hệ thống tệp tích hợp trong ESP32:

SPIFFS (SPI Flash File System) là hệ thống tập tin nhẹ được thiết kế cho các vi điều khiển có bộ nhớ flash hạn chế. Trên ESP32, SPIFFS cho phép lưu trữ và truy xuất các file như HTML, CSS, JavaScript, hình ảnh, v.v., trực tiếp từ bộ nhớ flash, giúp xây dựng một giao diện Web đầy đủ mà không cần phụ thuộc vào máy chủ bên ngoài.

Khi ESP32 hoạt động như một Web Server, các file giao diện này sẽ được gửi đến trình duyệt người dùng khi truy cập địa chỉ IP của thiết bị. [3]

- Các thành phần cơ bản của giao diện Web:

1. index.html:

- Là giao diện chính của Web Server.
- Chứa nội dung hiển thị như nút điều khiển, văn bản, biểu tượng, khung hiển thị dữ liệu,...
- Được tải đầu tiên khi người dùng truy cập ESP32.

Ví dụ: Gồm các nút bật/tắt LED, hiển thị nhiệt độ/độ ẩm,...

2. style.css:

- Định dạng giao diện người dùng: màu sắc, bố cục, hiệu ứng,...
- Tăng tính thẩm mỹ, giúp trang web dễ nhìn và dễ thao tác.
- Có thể sử dụng hiệu ứng CSS3 như đổ bóng, hiệu ứng chuyển động (hover, fade,...).

3. script.js:

- Thêm chức năng tương tác cho giao diện: lắng nghe sự kiện từ người dùng, gửi yêu cầu đến ESP32 (qua HTTP hoặc WebSocket).
 - Cập nhật dữ liệu theo thời gian thực, chẳng hạn như tình trạng thiết bị, nhiệt độ mới,...
 - Gửi lệnh điều khiển từ người dùng đến ESP32 (ví dụ: bật LED, đổi trạng thái relay,...).
-

- Lợi ích của việc sử dụng SPIFFS:

- Không phụ thuộc Internet:
Giao diện được lưu ngay trên thiết bị, có thể truy cập nội bộ mà không cần mạng toàn cầu.
- Dễ tùy biến:
Việc cập nhật giao diện chỉ cần thay đổi các file HTML/CSS/JS, rất dễ quản lý và thiết kế lại khi cần.
- Cập nhật qua OTA:
Có thể kết hợp với tính năng Over-The-Air để cập nhật file từ xa mà không cần kết nối cáp.
- Tối ưu tài nguyên:
Giao diện gọn nhẹ, phù hợp với bộ nhớ hạn chế của ESP32, nhưng vẫn mang lại trải nghiệm người dùng tốt.
- Phản hồi nhanh, hoạt động cục bộ:
Do không cần truy cập máy chủ bên ngoài nên tốc độ tải giao diện và phản hồi người dùng rất nhanh.

4.4. Mô phỏng thiết bị trên Wokwi

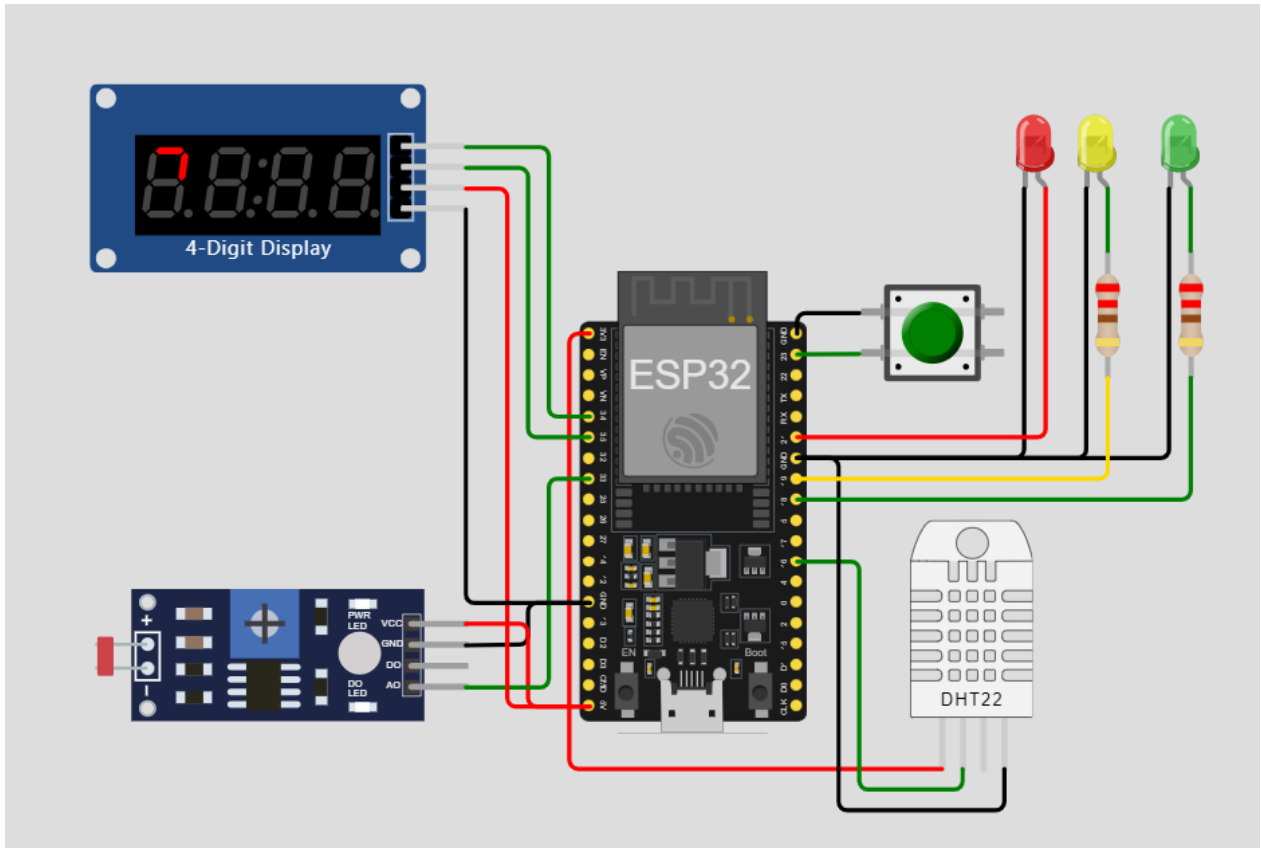
- Giới thiệu Wokwi:

Wokwi là nền tảng mô phỏng phần cứng miễn phí và trực tuyến, cho phép lập trình và kiểm tra mã Arduino, ESP32, ESP8266,... mà không cần phần cứng thực tế.

Người dùng có thể tạo sơ đồ mạch điện, nạp mã nguồn, theo dõi tín hiệu, cảm biến và kết quả hiển thị trực tiếp trên giao diện web.

Trang web: <https://wokwi.com> [8]

- Sơ đồ kết nối mô phỏng (dự án tiêu biểu):



Hệ thống được xây dựng mô phỏng một ứng dụng IoT quản lý thiết bị và giám sát môi trường gồm các phần tử chính:

- **ESP32 DevKit V4**
 - Là trung tâm điều khiển, xử lý tín hiệu từ cảm biến và điều khiển các thiết bị.
- **LED RGB (3 đèn riêng biệt: đỏ, vàng, xanh):**
 - Mô phỏng đèn tín hiệu (giao thông, trạng thái thiết bị).
 - Có thể điều khiển độc lập từng màu để thể hiện các trạng thái khác nhau.
- **Cảm biến DHT22 (nhiệt độ và độ ẩm):**
 - Mô phỏng môi trường thực tế.
 - Dữ liệu có thể được cập nhật ngẫu nhiên hoặc định sẵn để kiểm thử.
- **Nút nhấn (PushButton):**

- Mô phỏng nút điều khiển vật lý (ví dụ: dừng/tiếp tục hệ thống, chuyển chế độ, bật/tắt LED...).
 - **Màn hình TM1637 (4-digit 7-segment):**
 - Dùng để hiển thị dữ liệu như: thời gian đếm ngược, trạng thái thiết bị, số liệu cảm biến,...
-

- **Mô phỏng điều kiện môi trường:**

Wokwi hỗ trợ mô phỏng dữ liệu cảm biến một cách linh hoạt, bao gồm:

- **Ánh sáng (sử dụng LDR ảo):**
Có thể chỉnh độ sáng môi trường để thử phản ứng của thiết bị (ví dụ: bật đèn khi trời tối).
 - **Nhiệt độ, độ ẩm (DHT22):**
Dữ liệu có thể được lập trình để thay đổi ngẫu nhiên hoặc nhập thủ công, giúp kiểm tra khả năng phản hồi của hệ thống.
 - **Các cảm biến khác như: PIR, âm thanh, khí gas,...**
Đều có thể thêm vào để mở rộng mô hình theo yêu cầu.
-

- **Lợi ích khi sử dụng Wokwi:**

- Không cần phần cứng thật, tiết kiệm chi phí.
 - Lập trình trực tiếp trên nền web, hỗ trợ cả PlatformIO và Arduino IDE.
 - Dễ dàng chia sẻ dự án qua link, thuận tiện cho thảo luận nhóm và báo cáo.
 - Giúp sinh viên và người mới học IoT dễ dàng thử nghiệm các ý tưởng.
 - Hỗ trợ hiển thị Serial Monitor, điều chỉnh giá trị cảm biến trực tiếp.
-

- **Tình huống sử dụng thực tế:**

- Mô phỏng hệ thống đèn giao thông thông minh, có thể chuyển chế độ hoạt động tùy theo ánh sáng môi trường hoặc nút nhấn.
- Hiển thị nhiệt độ - độ ẩm đo được từ cảm biến DHT22 lên TM1637.

- Kết hợp điều khiển qua giao diện web hoặc ứng dụng Blynk trong mô phỏng ESP32.

4.5. Tương tác người dùng qua giao diện Web

• Mục tiêu:

Giao diện Web trên ESP32 không chỉ hiển thị thông tin mà còn cho phép người dùng điều khiển thiết bị và theo dõi trạng thái theo thời gian thực, tạo ra trải nghiệm tương tác trực quan như một bảng điều khiển thông minh.

• Các chức năng chính:

1. Bật/Tắt đèn từ Web:

- Người dùng truy cập giao diện Web và sử dụng nút bấm để bật hoặc tắt các thiết bị như LED.
- Khi nút được bấm, trình duyệt gửi yêu cầu HTTP đến ESP32, ví dụ:
`http://192.168.4.1/led?state=1`
- Trên ESP32, URL này được xử lý trong đoạn mã theo một route cụ thể để bật đèn.
- Tương tự, `state=0` dùng để tắt đèn:

`http://192.168.4.1/led?state=0`

2. Xem dữ liệu cảm biến:

- Các giá trị như nhiệt độ, độ ẩm từ cảm biến DHT22 được đọc định kỳ và hiển thị trên trang web (dưới dạng số hoặc biểu đồ).
- Dữ liệu có thể được cập nhật tự động bằng JavaScript (script.js) thông qua:
 - AJAX: Tự động gửi yêu cầu đến ESP32 để lấy dữ liệu mới.
 - WebSocket: Cập nhật thời gian thực (ít độ trễ, tương tác liên tục).

3. Điều chỉnh ngưỡng cảnh báo:

- Giao diện cho phép người dùng nhập hoặc chọn giá trị ngưỡng cảnh báo (ví dụ: nhiệt độ vượt 30°C thì cảnh báo).
- Dữ liệu gửi lên ESP32 có thể qua:

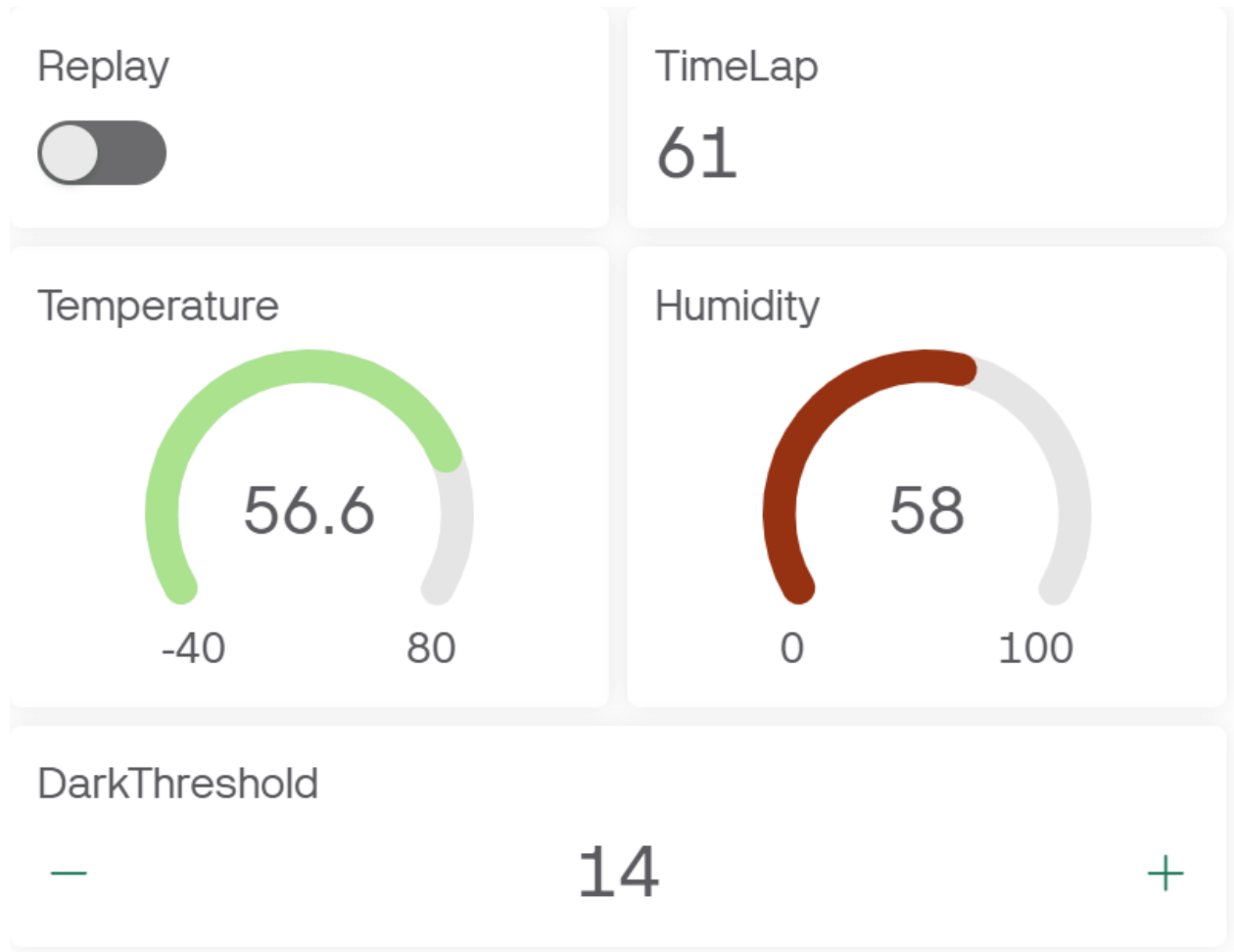
`/set_threshold?temp=30&humidity=40`

- Trên ESP32, khi cảm biến vượt ngưỡng đã cài, có thể:
 - Bật đèn cảnh báo.
 - Gửi thông báo (qua Telegram, ThingSpeak, Blynk,...).
 - Hiển thị thông điệp trên giao diện Web.
-

• **Lợi ích:**

- Dễ sử dụng: Điều khiển thiết bị chỉ với 1 nút bấm.
- Giao tiếp thời gian thực: Phản hồi nhanh chóng, giao diện luôn đồng bộ với trạng thái thiết bị.
- Cá nhân hóa: Cho phép người dùng tự đặt ngưỡng hoạt động phù hợp với từng môi trường, nhu cầu.
- Không phụ thuộc nền tảng: Tương thích với mọi thiết bị có trình duyệt: PC, điện thoại, tablet,...

4.6. Tích hợp Blynk hoặc ThingSpeak:



Việc kết hợp nền tảng điều khiển từ xa (Blynk) và lưu trữ dữ liệu trên đám mây (ThingSpeak) giúp hệ thống IoT với ESP32 trở nên linh hoạt và mạnh mẽ hơn, đồng thời đáp ứng nhu cầu giám sát cục bộ và toàn cầu.

- Blynk – Ứng dụng điều khiển từ xa: [9]

- Giới thiệu:

- Là một nền tảng IoT phổ biến hỗ trợ lập trình điều khiển thiết bị từ xa thông qua ứng dụng điện thoại (Android/iOS).
 - ESP32 giao tiếp với Blynk Cloud thông qua WiFi và sử dụng các Virtual Pins (V0 – V4) để truyền/nhận dữ liệu.

- Các chức năng tích hợp phổ biến:

- V0: Hiện thị thời gian hoạt động của hệ thống.
 - V1: Nút bật/tắt đèn từ xa.
 - V2: Nhiệt độ từ DHT22.
 - V3: Độ ẩm từ DHT22.
 - V4: Ngưỡng cảnh báo có thể chỉnh từ app.
 - Ưu điểm:
 - Tương thích tốt với ESP32.
 - Giao diện thân thiện, dễ kéo-thả.
 - Có thể dùng OTA (Over-the-Air) để cập nhật chương trình.
 - Giám sát và điều khiển từ mọi nơi có Internet.
-

• ThingSpeak – Lưu trữ & trực quan hóa dữ liệu: [10]

- Giới thiệu:
 - Là một nền tảng Cloud miễn phí từ MathWorks (MATLAB), hỗ trợ thu thập, phân tích và trực quan hóa dữ liệu IoT.
 - Dữ liệu cảm biến từ ESP32 được gửi lên channel của ThingSpeak định kỳ thông qua HTTP GET/POST.
 - Ứng dụng phổ biến:
 - Gửi dữ liệu như: temperature, humidity, light, v.v.
 - Theo dõi dữ liệu qua biểu đồ thời gian thực.
 - Có thể thiết lập cảnh báo, tự động phân tích với MATLAB.
 - Ưu điểm:
 - Biểu đồ đẹp, cập nhật tự động.
 - Miễn phí, dễ dùng, tài liệu rõ ràng.
 - Hỗ trợ chia sẻ channel công khai hoặc riêng tư.
-

- Tích hợp đồng thời WebServer và Blynk/ThingSpeak:

ESP32 hoàn toàn có thể:

- Làm Web Server nội bộ (hiển thị giao diện Web qua SPIFFS).
- Kết nối Blynk để điều khiển từ xa.
- Gửi dữ liệu lên ThingSpeak định kỳ.

Nhờ khả năng đa nhiệm (dual-core), ESP32 xử lý tốt việc giao tiếp cùng lúc với nhiều nền tảng mà vẫn đảm bảo hiệu suất ổn định.

4.7. Khả năng mở rộng và ứng dụng thực tiễn

Hệ thống sử dụng ESP32 không chỉ đơn thuần là một mô hình thử nghiệm, mà hoàn toàn có thể mở rộng thành các giải pháp thực tiễn phục vụ cuộc sống hàng ngày và công nghiệp. Với khả năng kết nối đa giao thức, hỗ trợ lưu trữ SPIFFS, tích hợp giao diện Web và kết nối Cloud, hệ thống này rất linh hoạt trong các kịch bản thực tế.

• Tự động hóa căn hộ thông minh (Smart Home):

- Kết nối nhiều thiết bị như: đèn, quạt, máy lạnh, cửa cuốn, cảm biến chuyển động,...
- Thiết lập ngữ cảnh tự động: ví dụ, khi trời tối và có chuyển động, đèn tự bật.
- Điều khiển toàn bộ hệ thống từ Web hoặc ứng dụng điện thoại (Blynk).
- Hệ thống có thể hoạt động độc lập khi mất kết nối Internet nhờ WebServer nội bộ.

• Điều khiển hệ thống điện sinh hoạt từ xa:

- Bật/tắt thiết bị điện (đèn học, quạt, máy lạnh...) thông qua giao diện Web hoặc Blynk.
- Lên lịch hoạt động theo giờ giấc học tập/làm việc.
- Tiết kiệm năng lượng bằng cách tự động tắt khi không sử dụng.

• Quản lý trạng thái thiết bị theo thời gian thực:

- Giao diện Web hiển thị trạng thái cập nhật liên tục (như thiết bị đang bật/tắt, nhiệt độ hiện tại,...).

- Dữ liệu cảm biến được gửi định kỳ lên Cloud (ThingSpeak) giúp theo dõi lịch sử.
 - Có thể thiết lập cảnh báo qua email/Telegram nếu có bất thường (quá nóng, độ ẩm thấp...).
-

• **Hỗ trợ nâng cấp firmware từ xa qua OTA (Over-the-Air):**

- Người dùng có thể cập nhật chương trình mới mà không cần kết nối cáp USB.
 - Quá trình OTA có thể kích hoạt từ:
 - Nút nhấn vật lý.
 - Lệnh từ Web interface.
 - Ứng dụng Blynk hoặc thời gian định kỳ.
 - Đảm bảo hệ thống luôn an toàn, linh hoạt và cập nhật nhanh chóng.
-

• **Đồng bộ dữ liệu giữa WebServer nội bộ và nền tảng Cloud:**

- Khi người dùng điều khiển từ Web nội bộ hoặc App Blynk, trạng thái thiết bị sẽ đồng bộ 2 chiều.
 - ESP32 có thể xử lý logic đảm bảo:
 - Nếu mất Internet, hệ thống vẫn hoạt động cục bộ.
 - Khi Internet có lại, dữ liệu sẽ tiếp tục được đồng bộ hóa lên Cloud.
 - Điều này giúp người dùng không bị phụ thuộc hoàn toàn vào mạng bên ngoài.
-

5. KẾT LUẬN VÀ KIẾN NGHỊ

5.1. Kết luận

Đề tài “Tạo máy chủ web cục bộ với ESP32 để quản lý thiết bị IoT” đã chứng minh được tính khả thi và hiệu quả trong việc xây dựng một hệ thống giám sát và điều khiển thiết bị thông minh với chi phí thấp, dễ triển khai và vận hành. Các kết quả đạt được có thể tóm tắt như sau:

- ESP32 đóng vai trò trung tâm trong hệ thống IoT, vừa thu thập dữ liệu cảm biến, vừa cung cấp giao diện Web qua SPIFFS.

- Máy chủ Web cục bộ (WebServer) cho phép người dùng giám sát và điều khiển thiết bị trong mạng LAN mà không cần kết nối Internet, đảm bảo tính ổn định và chủ động trong sử dụng.
- Việc tích hợp với các nền tảng như Blynk và ThingSpeak mở rộng thêm khả năng điều khiển từ xa và trực quan hóa dữ liệu trên Internet, mang lại sự linh hoạt và tiện lợi cho người dùng.
- Mô phỏng trên Wokwi giúp kiểm thử nhanh và chính xác trước khi triển khai thực tế, tiết kiệm thời gian và chi phí phần cứng.

Như vậy, đề tài không chỉ mang lại ý nghĩa học thuật mà còn có tiềm năng ứng dụng cao trong thực tế, đặc biệt trong các hệ thống nhà thông minh, nông nghiệp số và quản lý môi trường.

5.2. Kiến nghị

Dựa trên kết quả và tiềm năng mở rộng của hệ thống, nhóm đề xuất một số hướng phát triển trong tương lai:

- **Tích hợp trí tuệ nhân tạo (AI):**
Áp dụng các mô hình AI để nhận diện giọng nói, hình ảnh (ví dụ: mở cửa khi nhận diện đúng khuôn mặt), nâng cao mức độ tương tác và tự động hóa.
- **Cảnh báo nâng cao:**
Hệ thống có thể gửi email, tin nhắn SMS hoặc thông báo Telegram khi phát hiện các điều kiện bất thường như nhiệt độ quá cao, độ ẩm quá thấp,... giúp phản ứng nhanh chóng với các sự cố.
- **Mở rộng thiết bị điều khiển:**
Bổ sung các thiết bị như relay, motor, quạt, cảm biến khói, PIR, camera,... để mở rộng hệ thống thành giải pháp điều khiển toàn diện cho căn hộ, trang trại hoặc nhà xưởng.
- **Hỗ trợ nhiều người dùng:**
Cho phép phân quyền người dùng (quản trị, khách truy cập), phù hợp với các hệ thống nhiều người sử dụng.
- **Ứng dụng di động riêng:**
Xây dựng App tùy biến riêng thay vì phụ thuộc vào nền tảng thứ ba, giúp tối ưu hóa giao diện và tăng khả năng thương mại hoá.

6. TÀI LIỆU THAM KHẢO

Tiếng Việt:

- [1] Nguyễn Trọng Nhân (2020). *Internet of Things – Tổng quan và ứng dụng trong đời sống*, Tạp chí Khoa học và Công nghệ Việt Nam, số 6, tr. 45-50.
- [4] Trường Đại học Bách khoa Hà Nội. (2021). *Bài giảng Hệ thống IoT*.
- [7] Học viện Công nghệ Bưu chính Viễn thông. (2022). *Ứng dụng IoT trong nhà thông minh*.
- [8] Wokwi. (2023). *Wokwi: Mô phỏng mạch điện trực tuyến miễn phí*. <https://wokwi.com>

Tiếng Anh:

- [2] Espressif Systems. (2023). *ESP32 Series Datasheet*. Available at: https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf (Accessed: 7 April 2025).
- [3] Random Nerd Tutorials. (n.d.). *ESP32 Web Server using SPIFFS (SPI Flash File System)*. Available at: <https://randomnerdtutorials.com/esp32-web-server-spiiffs> (Accessed: 7 April 2025).
- [5] Espressif Systems. (2023). *ESP32 Technical Reference Manual*. <https://www.espressif.com>
- [6] Rui Santos & Sara Santos. (2022). *ESP32 Web Server Tutorial*. Random Nerd Tutorials. <https://randomnerdtutorials.com>
- [9] Blynk Documentation. (2023). *ESP32 with Blynk*. <https://docs.blynk.io>
- [10] ThingSpeak Documentation. (2023). *Using ESP32 to send data to ThingSpeak*. <https://thingspeak.com>
-