

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



BÀI TIỂU LUẬN

**ĐỀ TÀI: ĐIỀU KHIỂN TỪ XA QUA GIAO
THỨC MQTT VỚI ESP32.**

(Thiết kế hệ thống bật/tắt thiết bị từ xa)

Tên học phần: Phát triển ứng dụng IoT- Nhóm 5

Giảng viên hướng dẫn: Võ Việt Dũng

Khóa: K45 - Hệ chính quy

Huế, 04 – 2025

MỤC LỤC

PHẦN I. MỞ ĐẦU.....	1
1. Lý do thực hiện đề tài.....	1
2. Mục tiêu đề tài.....	1
3. Phương pháp thực hiện.....	1
PHẦN II. NỘI DUNG.....	2
1. Cơ sở lý thuyết.....	2
1.1. Vi điều khiển ESP32.....	2
1.1.1. ESP32 là gì?.....	2
1.1.2. Các tính năng của ESP32. ^[2]	3
1.2. Giao thức MQTT và MosquittoBroker.....	4
1.2.1. Giao thức MQTT là gì? ^[3]	4
1.2.2. Mô hình Publish/Subscriber trong giao thức MQTT ^[3]	5
1.2.3. Cơ chế hoạt động của giao thức MQTT ^[3]	5
1.2.4. MosquittoBroker ^[4]	6
2. Thiết kế mạch.....	7
2.1. Các thành phần thiết kế mô phỏng mạch.....	7
2.2. Sơ đồ mô phỏng mạch bằng Wokwi.....	7
PHẦN III. KẾT LUẬN.....	10
TÀI LIỆU THAM KHẢO	

PHẦN I. MỞ ĐẦU.

1. Lý do thực hiện đề tài.

Trong thời đại công nghệ 4.0, Internet of Things (IoT) ngày càng đóng vai trò quan trọng trong nhiều lĩnh vực như tự động hóa, điều khiển từ xa, nhà thông minh và sản xuất công nghiệp. Các hệ thống IoT giúp con người giám sát và điều khiển thiết bị từ xa một cách hiệu quả, tiết kiệm thời gian và nâng cao hiệu suất.

Trong số các giao thức truyền thông được sử dụng trong IoT, MQTT (Message Queuing Telemetry Transport) là một trong những giao thức phổ biến và được ưa chuộng nhất nhờ tính nhẹ, đáng tin cậy và khả năng hoạt động hiệu quả trên các thiết bị có tài nguyên hạn chế. MQTT hoạt động dựa trên mô hình Publisher - Broker - Subscriber, giúp các thiết bị IoT có thể trao đổi dữ liệu một cách nhanh chóng và dễ dàng.

ESP32 là một vi điều khiển mạnh mẽ với khả năng kết nối WiFi và hỗ trợ nhiều giao thức IoT, trong đó có MQTT. Nhờ hiệu suất cao, chi phí thấp và khả năng lập trình linh hoạt, ESP32 trở thành một lựa chọn lý tưởng để xây dựng các hệ thống điều khiển từ xa.

Dựa trên những lợi ích trên, đề tài "Điều khiển từ xa qua giao thức MQTT với ESP32" được thực hiện nhằm nghiên cứu, thiết kế và mô phỏng một hệ thống điều khiển thiết bị từ xa sử dụng ESP32 làm client và Mosquitto Broker để quản lý dữ liệu truyền nhận.

2. Mục tiêu đề tài.

- Nghiên cứu tổng quan về ESP32 và ứng dụng của nó trong IoT.
- Tìm hiểu chi tiết về giao thức MQTT và Mosquitto Broker.
- Thiết kế hệ thống bật/tắt thiết bị từ xa sử dụng ESP32 và giao thức MQTT.
- Mô phỏng và kiểm thử hệ thống trên nền tảng Wokwi.

3. Phương pháp thực hiện.

- Nghiên cứu lý thuyết: Thu thập tài liệu về ESP32, giao thức MQTT, Mosquitto Broker và các công cụ hỗ trợ mô phỏng.

- Cấu hình Mosquitto Broker: Cài đặt và cấu hình Mosquitto trên máy tính để làm máy chủ trung gian trao đổi dữ liệu giữa các thiết bị.
- Lập trình ESP32: Viết chương trình kết nối ESP32 với Mosquitto Broker và thực hiện điều khiển thiết bị từ xa.
- Mô phỏng trên Wokwi: Thiết kế và kiểm tra hệ thống bằng cách sử dụng nền tảng mô phỏng Wokwi.
- Đánh giá kết quả: Kiểm tra hoạt động của hệ thống, phân tích hiệu suất và đưa ra nhận xét.

PHẦN II. NỘI DUNG.

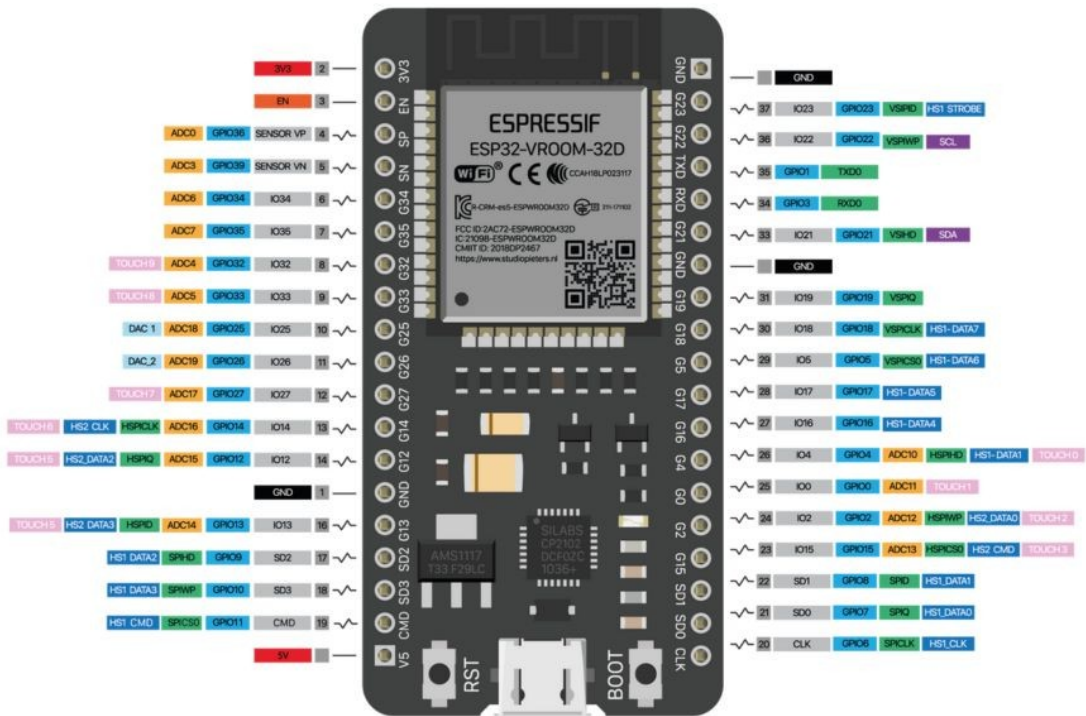
1. Cơ sở lý thuyết.

1.1. Vi điều khiển ESP32.

1.1.1. ESP32 là gì?

“ESP32 là một vi điều khiển giá rẻ, năng lượng thấp có hỗ trợ WiFi và dual-mode Bluetooth (tạm dịch: Bluetooth chế độ kép). Dòng ESP32 sử dụng bộ vi xử lý Tensilica Xtensa LX6 ở cả hai biến thể lõi kép và lõi đơn, và bao gồm các công tắc antenna tích hợp, RF balun, bộ khuếch đại công suất, bộ khuếch đại thu nhiễu thấp, bộ lọc và module quản lý năng lượng.

ESP32 được chế tạo và phát triển bởi Espressif Systems, một công ty Trung Quốc có trụ sở tại Thượng Hải, và được sản xuất bởi TSMC bằng cách sử dụng công nghệ 40 nm. ESP32 là sản phẩm kế thừa từ vi điều khiển ESP8266.”^[1]



Hình 1. Vi điều khiển ESP32.

1.1.2. Các tính năng của ESP32.^[2]

- Bộ xử lý:
 - CPU: Bộ vi xử lý Xtensa lõi kép (hoặc lõi đơn) 32-bit LX6, hoạt động ở tần số 240 MHz (160 MHz cho ESP32-S0WD và ESP32-U4WDH) và hoạt động ở tối đa 600 MIPS (200 MIPS với ESP32-S0WD/ESP32-U4WDH)
 - Bộ đồng xử lý ULP hỗ trợ chế độ deep sleep.
- Hệ thống xung nhịp: CPU Clock(Tần số 80-240 MHz), RTC Clock và Audio PLL Clock(Tần số 32 kHz đến 240 MHz)
- Bộ nhớ nội:
 - 448 KB bộ nhớ ROM cho việc booting và các tính năng lõi
 - 520 KB bộ nhớ SRAM trên chip cho dữ liệu và tập lệnh
- Kết nối không dây:
 - Wi-Fi: 802.11 b/g/n. Hoạt động với tần số: 2.4 GHz. Hỗ trợ cả Station (kết nối với mạng Wi-Fi) và Access Point (tạo điểm phát Wi-Fi).Tốc độ truyền dữ liệu tối đa 150 Mbps.

- Bluetooth: v4.2 BR/EDR và BLE (chia sẻ sóng vô tuyến với Wi-Fi). Hỗ trợ kết nối với các thiết bị như điện thoại, tai nghe không dây.
- GPIO & Ngoại vi: Hỗ trợ nhiều giao thức như UART, SPI, I2C, ADC, DAC, PWM.
- Bảo mật:
 - Hỗ trợ tất cả các tính năng bảo mật chuẩn IEEE 802.11, bao gồm WPA, WPA/WPA2 và WAPI.
 - Secure boot (khởi động an toàn): Ngăn chặn việc chạy firmware không được kí bởi nhà sản xuất
 - Flash Encryption(Mã hóa flash): Bảo vệ dữ liệu lưu trữ trong bộ nhớ Flash
 - 1024-bit OTP, lên đến 768-bit cho khách hàng
 - Tăng tốc mã hóa phần cứng: AES, SHA-2, RSA, elliptic curve cryptography(ECC) để mã hoá dữ liệu, đảm bảo an toàn khi truyền qua WiFi hoặc Bluetooth
- Quản lý năng lượng:
 - Hỗ trợ 5 chế độ hoạt động với mức tiêu thụ năng lượng khác nhau: Active, Modem-sleep, Light-sleep, Deep-sleep và Hibernation
 - Bộ ổn áp nội với điện áp rơi thấp (internal low-dropout regulator)
 - Individual power domain (tạm dịch: Miền nguồn riêng) cho RTC
 - Trở lại hoạt động từ ngắt GPIO, timer, đo ADC, ngắt với cảm ứng điện dung
- Hỗ trợ các giao thức IoT: MQTT, HTTP, CoAP.

1.2. Giao thức MQTT và MosquittoBroker

1.2.1. *Giao thức MQTT là gì?* [3]

MQTT (Message Queueing Telemetry Transport) là một giao thức mạng kích thước nhỏ (lightweight), hoạt động theo cơ chế publish – subscribe (tạm dịch: xuất bản – đăng ký) theo tiêu chuẩn ISO (ISO/IEC 20922) và OASIS mở để truyền tin nhắn giữa các thiết bị.

Giao thức này hoạt động trên nền tảng TCP/IP. MQTT được thiết kế cho các kết nối cho việc truyền tải dữ liệu cho các thiết bị ở xa, các thiết bị hay vi điều khiển nhỏ có tài nguyên hạn chế hoặc trong các ứng dụng có băng thông mạng bị hạn chế.

Tính năng, đặc điểm nổi bật

- Dạng truyền thông điệp theo mô hình Pub/Sub cung cấp việc truyền tin phân tán một chiều, tách biệt với phản ứng dụng.
- Việc truyền thông điệp là ngay lập tức, không quan tâm đến nội dung được truyền.
- Sử dụng TCP/IP là giao thức nền.
- Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu (QoS: Quality of service)
 - QoS 0: Broker/client sẽ gửi dữ liệu đúng một lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP.
 - QoS 1: Broker/client sẽ gửi dữ liệu với ít nhất một lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.
 - QoS 2: Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng một lần, quá trình này phải trải qua 4 bước bắt tay.
- Phân bao bọc dữ liệu truyền nhỏ và được giảm đến mức tối thiểu để giảm tải cho đường truyền.

1.2.2. Mô hình Publish/Subscriber trong giao thức MQTT_[3]

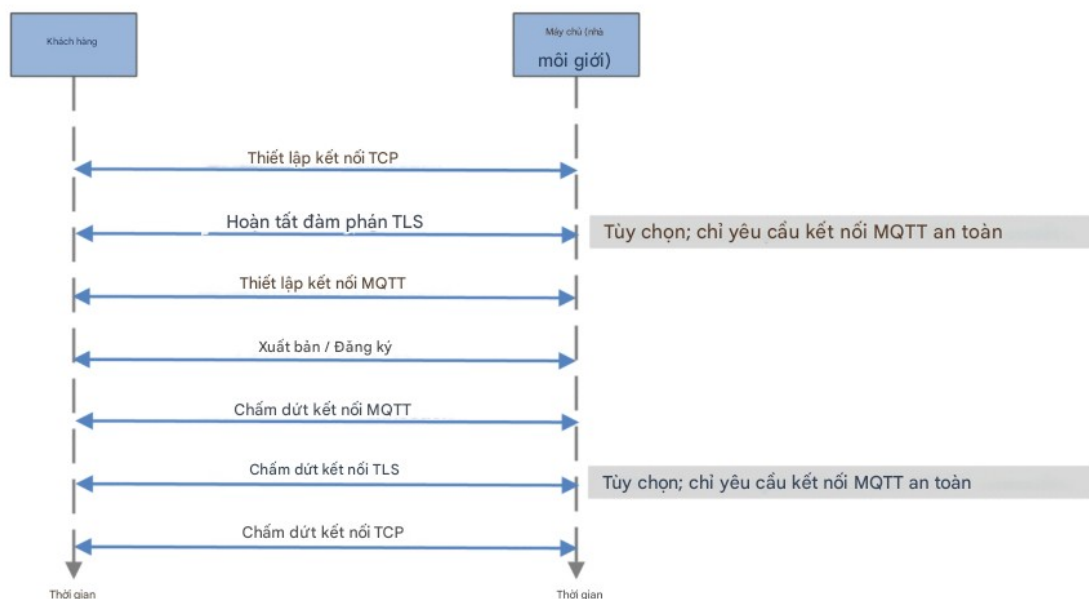
MQTT Broker:

- MQTT Broker hay máy chủ mô giới được coi như trung tâm, nó là điểm giao của tất cả các kết nối đến từ Client (Publisher/Subscriber).
- Nhiệm vụ chính: nhận thông điệp (message) từ Publisher, xếp vào hàng đợi rồi chuyển đến một địa điểm cụ thể.
- Nhiệm vụ phụ:

1.2.3. Cơ chế hoạt động của giao thức MQTT_[3]

Một phiên MQTT được chia thành bốn giai đoạn: kết nối, xác thực, giao tiếp và kết thúc.

- (1) Client (máy khách) bắt đầu bằng cách tạo kết nối Transmission Control Protocol/Internet Protocol (TCP/IP) tới broker bằng cách sử dụng cổng tiêu chuẩn hoặc cổng tùy chỉnh được xác định bởi các nhà phát triển broker.
- (2) Các cổng tiêu chuẩn là 1883 cho giao tiếp không mã hóa và 8883 cho giao tiếp được mã hóa – sử dụng Lớp cổng bảo mật (SSL) / Bảo mật lớp truyền tải (TLS). Trong quá trình giao tiếp SSL/TLS, máy khách cần kiểm chứng và xác thực máy chủ.
- (3) Sau đó, Client sẽ gửi bản tin lên broker nếu là Publisher hoặc nhận bản tin từ broker về nếu là Subscriber. Quá trình kết nối này sẽ được giữ đến khi Kết thúc kết nối.
- (4) Sau khi kết thúc để có thể truyền nhận MQTT, chúng ta lại tiếp tục quay lại các bước trên.



Hình 1. Cơ chế hoạt động của MQTT.

1.2.4. *MosquittoBroker*_[4]

Mosquitto là một MQTT Broker mã nguồn mở cho phép thiết bị truyền nhận dữ liệu theo giao thức MQTT version 5.0, 3.1.1 và 3.1 – Một giao thức nhanh, nhẹ theo mô hình publish/subscribe được sử dụng rất nhiều trong lĩnh vực Internet of Things. Mosquitto cung cấp một thư viện viết bằng ngôn ngữ C để triển khai các MQTT Client và có thể dễ dàng sử dụng bằng dòng lệnh: “mosquitto_pub” và “mosquitto_sub”.

Ngoài ra, Mosquitto cũng là một phần của Eclipse Foundation, là dự án iot.eclipse.org và được tài trợ bởi cedalo.com

Ưu điểm:

- Ưu điểm nổi bật của Mosquitto là tốc độ truyền nhận và xử lý dữ liệu nhanh, độ ổn định cao, được sử dụng rộng rãi và phù hợp với những ứng dụng embedded.
- Mosquitto rất nhẹ và phù hợp để sử dụng trên tất cả các thiết bị.
- Ngoài ra, Mosquitto cũng được hỗ trợ các giao thức TLS/SSL (các giao thức nhằm xác thực server và client, mã hóa các message để bảo mật dữ liệu).

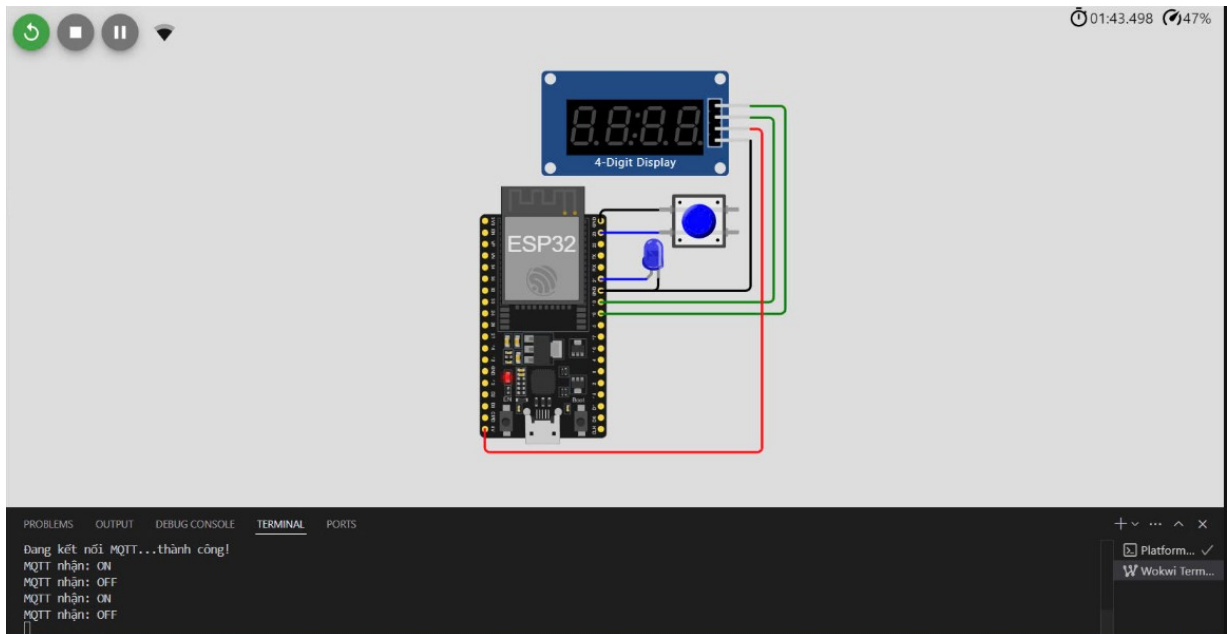
Nhược điểm: Khó thiết kế khi làm những ứng dụng lớn và ít phương thức xác thực thiết bị nên khả năng bảo mật vẫn chưa tối ưu.

2. Thiết kế mạch

2.1. Các thành phần thiết kế mô phỏng mạch

- ESP32: Vi điều khiển chính, kết nối WiFi để giao tiếp MQTT và điều khiển các thiết bị.
- Màn hình LED 4 digit: Hiển thị trạng thái(ON/OFF) hoặc thông tin cần thiết.
- Nút nhấn (Push Button): Dùng để bật/tắt thiết bị hoặc kích hoạt hành động.
- Đèn LED: Mô phỏng thiết bị cần điều khiển (bật/tắt theo MQTT hoặc nút nhấn).
- Dây nối: Để kết nối các thành phần.
- Nguồn cấp 3.3V từ ESP32: Cung cấp điện cho các linh kiện như màn hình, LED, nút nhấn.
- Kết nối WiFi & MQTT Broker: Cho phép ESP32 giao tiếp với MQTT server để nhận lệnh điều khiển từ xa.

2.2. Sơ đồ mô phỏng mạch bằng Wokwi



Hình 2. Sơ đồ mô phỏng mạch bằng Wokwi

Mô tả chi tiết cách các linh kiện được kết nối với ESP32 để tạo hệ thống điều khiển thiết bị từ xa qua MQTT.:

- Kết nối phần cứng ESP32 và các thiết bị ngoại vi

Linh kiện	Chân linh kiện	Kết nối với ESP32	Chức năng
Màn hình LED 4 digit	VCC	3.3V	Cung cấp nguồn điện cho màn hình LED
	GND	GND	Nối đất chung để đảm bảo hoạt động đúng
	CLK (Clock)	GPIO18	Gửi xung clock để đồng bộ dữ liệu hiển thị
	DIO (Data)	GPIO19	Truyền dữ liệu để hiển thị số trên màn hình
Nút nhấn	Chân 1	GPIO4	Nhận tín hiệu từ nút nhấn (HIGH/LOW)
	Chân 2	GND	Tạo đường nối đất khi nút được nhấn

Đèn LED	Anode (Xanh dương)	GPIO2	Điều khiển bật/tắt
	Cathode (Đen)	GND	Kết nối đất để đóng mạch điện
Nguồn cấp	3.3V	3.3V	Cấp nguồn cho linh kiện
	GND	GND	Nối đất chung để đảm bảo hoạt động đúng

- Kết nối ESP32 WiFi & MQTT Broker : ESP32 sẽ:
 - Kết nối WiFi để truy cập internet.
 - Giao tiếp với MQTT Broker thông qua địa chỉ IP của máy chủ MQTT.
 - Nhận tín hiệu MQTT ("ON"/"OFF") từ thiết bị điều khiển từ xa.
 - Cập nhật trạng thái lên màn hình LED 4 digit và đèn LED theo lệnh MQTT
- Mô phỏng trên Wokwi:
 - Khi ESP32 nhận lệnh MQTT "ON", đèn LED sáng, màn hình hiển thị "ON".
 - Khi ESP32 nhận lệnh "OFF", đèn LED tắt, màn hình hiển thị "OFF".
 - Nút nhấn giúp điều khiển thủ công tại chỗ.

Sơ đồ này mô phỏng một hệ thống điều khiển thiết bị IoT qua MQTT, trong đó ESP32 đóng vai trò client MQTT, có khả năng gửi và nhận lệnh từ một MQTT Broker. Điều này cho phép người dùng điều khiển các thiết bị từ xa thông qua mạng Internet.

PHẦN III. KẾT LUẬN.

Sau quá trình nghiên cứu, thiết kế và thực nghiệm, đề tài "Điều khiển từ xa qua giao thức MQTT với ESP32" đã đạt được các mục tiêu đề ra. Hệ thống đã được xây dựng thành công, giúp bật/tắt thiết bị từ xa thông qua giao thức MQTT một cách hiệu quả và ổn định.

Thông qua việc tìm hiểu ESP32, giao thức MQTT và Mosquitto Broker, ta có thể thấy rằng MQTT là một giải pháp tối ưu cho các hệ thống IoT nhờ tính nhẹ, bảo mật và khả năng truyền dữ liệu nhanh chóng. ESP32 với khả năng kết nối WiFi mạnh mẽ đã chứng minh được vai trò quan trọng trong việc phát triển các ứng dụng IoT thực tế.

Mô phỏng trên Wokwi cho kết quả chính xác, hệ thống hoạt động theo đúng nguyên lý mong đợi. Tuy nhiên, để ứng dụng vào thực tế, cần tiếp tục tối ưu về phần cứng và phần mềm nhằm nâng cao tính ổn định và bảo mật. Đồng thời, có thể mở rộng hệ thống bằng cách tích hợp thêm các tính năng như điều khiển qua ứng dụng di động, giám sát dữ liệu thời gian thực và bảo mật nâng cao.

Với tiềm năng rộng lớn, đề tài này có thể tiếp tục phát triển và ứng dụng trong nhiều lĩnh vực khác nhau như nhà thông minh, công nghiệp tự động hóa và quản lý năng lượng.

TÀI LIỆU THAM KHẢO

[1] “Lập Trình ESP32 Từ A Tới Z”. Khuê Nguyễn Creator.

<https://khuenguyencreator.com/lap-trinh-esp32-tu-a-toi-z/>

[2] “ESP32”. Bách khoa toàn thư mở Wikipedia.

<https://vi.wikipedia.org/wiki/ESP32>

[3] “Giao thức MQTT là gì? Cách sử dụng trong lập trình IoT”. Khuê Nguyễn Creator.

<https://khuenguyencreator.com/giao-thuc-mqtt-la-gi-cach-su-dung/>

[4] “Tìm hiểu về MQTT Mosquitto broker và cách cài đặt”. Khuê Nguyễn Creator.

<https://viblo.asia/p/tim-hieu-ve-mqtt-mosquitto-broker-va-cach-cai-dat-yMnKMjgrZ7P>