TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA CÔNG NGHỆ THÔNG TIN

SỐ PHÁCH: ……

TÊN ĐỀ TÀI TIỂU LUẬN

ĐIỀU KHIỂN TỪ XA QUA GIAO THỨC MQTT VỚI ESP32

TÊN LỚP HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG IOT - NHÓM 6

MÃ HỌC PHẦN: 2024-2025.2.TIN4024.006

GIẢNG VIÊN HƯỚNG DẪN: VÕ VIỆT DŨNG

HUẾ, THÁNG 4 NĂM 2025

TRƯỜNG ĐẠI HỌC KHOA HỌC

KHOA CÔNG NGHỆ THÔNG TIN

TÊN ĐỀ TÀI TIỂU LUẬN

ĐIỀU KHIỂN TỪ XA QUA GIAO THỨC MQTT VỚI ESP32

TÊN LỚP HỌC PHẦN : PHÁT TRIỂN ỨNG DỤNG IOT - NHÓM 6

MÃ HỌC PHẦN: 2024-2025.2.TIN4024.006

Giảng viên hướng dẫn : VÕ VIỆT DŨNG

HUẾ, THÁNG 4 NĂM 2025

Mục lục

Phần mở đầu

Trong thời đại công nghệ số phát triển với tốc độ chóng mặt, **Internet of Things (IoT)** – hay còn gọi là Internet vạn vật – đang ngày càng thể hiện vai trò thiết yếu trong cuộc sống hiện đại. Từ các thiết bị gia dụng thông minh trong gia đình, hệ thống giám sát môi trường đến các dây chuyền sản xuất tự động trong công nghiệp, IoT đã và đang góp phần **nâng cao chất lượng sống, tối ưu hóa quy trình vận hành** và thúc đẩy quá trình số hóa trên toàn cầu.

Một trong những yếu tố cốt lõi giúp IoT phát triển mạnh mẽ là **khả năng truyền thông hiệu quả giữa các thiết bị**. Trong đó, **giao thức MQTT (Message Queuing Telemetry Transport)** nổi bật như một giải pháp lý tưởng: nhẹ, ổn định và tiết kiệm băng thông. Được thiết kế dành riêng cho các thiết bị có tài nguyên hạn chế, MQTT giúp việc trao đổi dữ liệu trong môi trường IoT trở nên nhanh chóng, hiệu quả và có độ tin cậy cao.

Cùng với đó, **ESP32** – một dòng vi điều khiển tích hợp Wi-Fi và Bluetooth – nổi lên như một lựa chọn phổ biến trong các ứng dụng IoT nhờ vào **hiệu năng mạnh mẽ, chi phí thấp và khả năng mở rộng linh hoạt**. Khi kết hợp ESP32 với giao thức MQTT, người dùng có thể xây dựng những hệ thống điều khiển từ xa một cách dễ dàng, mở ra nhiều cơ hội cho các ứng dụng thực tiễn như: **nhà thông minh (Smart Home), nông nghiệp chính xác (Smart Agriculture), quản lý năng lượng, an ninh, ...**

Trong khuôn khổ bài tiểu luận này, chúng ta sẽ cùng tìm hiểu và phân tích **cách lập trình điều khiển từ xa sử dụng ESP32 thông qua giao thức MQTT**. Nội dung sẽ bao gồm từ việc thiết lập môi trường phát triển, viết và nạp chương trình, đến kiểm thử hệ thống, đồng thời đề xuất một số cải tiến và hướng mở rộng để ứng dụng hiệu quả hơn trong thực tế. Qua đó, bài viết hy vọng sẽ mang lại cái nhìn rõ ràng và thực tiễn về việc triển khai công nghệ IoT hiện đại bằng những công cụ phổ biến và dễ tiếp cận.

CHƯƠNG 1: TỔNG QUAN VỀ HỆ THỐNG ĐIỀU KHIỂN TỪ XA

1. Giới thiệu chung về hệ thống điều khiển từ xa

1.1. Tầm quan trọng của hệ thống điều khiển từ xa trong đời sống:

Trong thời đại công nghệ số, hệ thống điều khiển từ xa ngày càng đóng vai trò quan trọng trong nhiều lĩnh vực từ cuộc sống hàng ngày đến sản xuất công nghiệp. Việc ứng dụng các hệ thống điều khiển từ xa giúp cải thiện đáng kể trải nghiệm người dùng, tối ưu hóa năng suất và giảm thiểu sự can thiệp trực tiếp của con người vào quá trình vận hành thiết bị.

Tiện lợi và tiết kiệm thời gian: Một trong những lợi ích lớn nhất của hệ thống điều khiển từ xa là giúp người dùng có thể quản lý và điều khiển thiết bị mà không cần có mặt trực tiếp.

Tối ưu hóa năng suất và hiệu quả sử dụng thiết bị: Hệ thống điều khiển từ xa giúp các thiết bị vận hành một cách thông minh và có hiệu suất cao hơn. Trong các hệ thống tự động hóa công nghiệp, việc giám sát và điều khiển máy móc từ xa giúp giảm thời gian chết của thiết bị, tối ưu hóa quy trình sản xuất và đảm bảo hoạt động ổn định.

Tiết kiệm năng lượng và chi phí vận hành: Một hệ thống điều khiển từ xa hiệu quả có thể giúp giảm đáng kể mức tiêu thụ điện năng và chi phí vận hành. Trong sản xuất công nghiệp, việc giám sát và điều khiển các dây chuyền từ xa giúp tránh lãng phí tài nguyên và nâng cao hiệu suất hoạt động.

Tăng cường khả năng giám sát và bảo mật: Hệ thống điều khiển từ xa không chỉ giúp vận hành thiết bị mà còn có thể tích hợp các công nghệ giám sát và cảnh báo an toàn. Các hệ thống camera an ninh, cảm biến chuyển động, cảnh báo cháy nổ có thể được giám sát và điều khiển từ xa để phát hiện sớm các nguy cơ và có biện pháp xử lý kịp thời.

1.2. Các phương pháp điều khiển từ xa phổ biến hiện nay:

Hiện nay, có nhiều phương pháp điều khiển từ xa được sử dụng, bao gồm:

Hồng ngoại (IR):

Nguyên lý hoạt động: Sử dụng tín hiệu ánh sáng hồng ngoại để truyền dữ liệu từ bộ điều khiển đến thiết bị.

Ưu điểm:

Chi phí thấp, dễ triển khai.

Phù hợp với các thiết bị đơn giản như TV, điều hòa, quạt điện

Nhược điểm:

Chỉ hoạt động trong phạm vi ngắn (dưới 10m) và yêu cầu đường truyền không có vật cản.

Không thể điều khiển từ xa qua Internet.

Sống vô tuyến RF (Radio Frequency): Sử dụng sóng vô tuyến để điều khiển thiết bị ở khoảng cách xa hơn so với hồng ngoại.

Nguyên lý hoạt động: Sử dụng sóng vô tuyến ở các tần số như 433MHz hoặc 2.4GHz để gửi tín hiệu điều khiển.

Ưu điểm:

Khoảng cách điều khiển xa hơn hồng ngoại, có thể lên đến hàng trăm mét tùy vào công suất phát.

Không cần đường truyền trực tiếp, có thể hoạt động xuyên vật cản.

Nhược điểm:

Dễ bị nhiễu do có nhiều thiết bị cùng sử dụng băng tần RF.

Không thể kiểm soát từ xa qua Internet nếu không có sự hỗ trợ của các hệ thống bổ sung.

Wi-Fi/Bluetooth: Kết nối thông qua mạng không dây để điều khiển thiết bị thông qua ứng dụng trên điện thoại thông minh hoặc máy tính.

Nguyên lý hoạt động:

Wi-Fi: Kết nối thiết bị với mạng không dây để điều khiển qua ứng dụng trên điện thoại hoặc máy tính.

Bluetooth: Kết nối trực tiếp giữa thiết bị điều khiển và thiết bị nhận trong phạm vi ngắn (dưới 10m).

Ưu điểm:

Wi-Fi: Có thể điều khiển từ xa thông qua Internet.

Bluetooth: Tiết kiệm năng lượng hơn Wi-Fi và không cần kết nối Internet.

Nhược điểm:

Wi-Fi: Yêu cầu mạng Internet ổn định, có thể bị gián đoạn nếu mất kết nối.

Bluetooth: Phạm vi giới hạn, không phù hợp với các ứng dụng điều khiển từ xa qua mạng diện rộng.

Giao thức MQTT: Sử dụng giao thức truyền thông nhẹ để điều khiển và giám sát thiết bị thông qua mạng Internet.

Nguyên lý hoạt động: Sử dụng mô hình **publish/subscribe** để truyền tải dữ liệu giữa các thiết bị thông qua một broker trung gian.

Ưu điểm:

Tiết kiệm băng thông: Giao thức nhẹ, phù hợp với các thiết bị IoT có tài nguyên hạn chế.

Linh hoạt và mở rộng: Có thể kết nối hàng trăm, thậm chí hàng nghìn thiết bị trong cùng một hệ thống.

Điều khiển từ xa qua Internet: Khả năng giám sát và điều khiển thiết bị từ bất kỳ đâu có kết nối mạng.

Nhược điểm:

Cần một MQTT broker để vận hành hệ thống.

Yêu cầu cấu hình bảo mật để tránh rủi ro khi truyền dữ liệu qua Internet

1.3. Xu hướng phát triển các hệ thống điều khiển thông minh:

Với sự phát triển của Internet of Things (IoT), các hệ thống điều khiển ngày càng trở nên thông minh hơn, cho phép người dùng giám sát và điều khiển thiết bị từ xa thông qua các ứng dụng di động, giọng nói hoặc tự động hóa dựa trên các kịch bản được thiết lập trước.

Tích hợp trí tuệ nhân tạo (AI): Hệ thống điều khiển có thể học thói quen người dùng, tối ưu hóa hoạt động dựa trên dữ liệu thu thập được.

Điều khiển bằng giọng nói: Kết hợp với các trợ lý ảo như Google Assistant, Alexa hoặc Siri để cho phép người dùng điều khiển thiết bị bằng giọng nói.

Tự động hóa dựa trên kịch bản: Người dùng có thể thiết lập các kịch bản để hệ thống tự động hoạt động theo điều kiện cụ thể (ví dụ: bật đèn khi trời tối, điều chỉnh nhiệt độ dựa vào thời tiết).

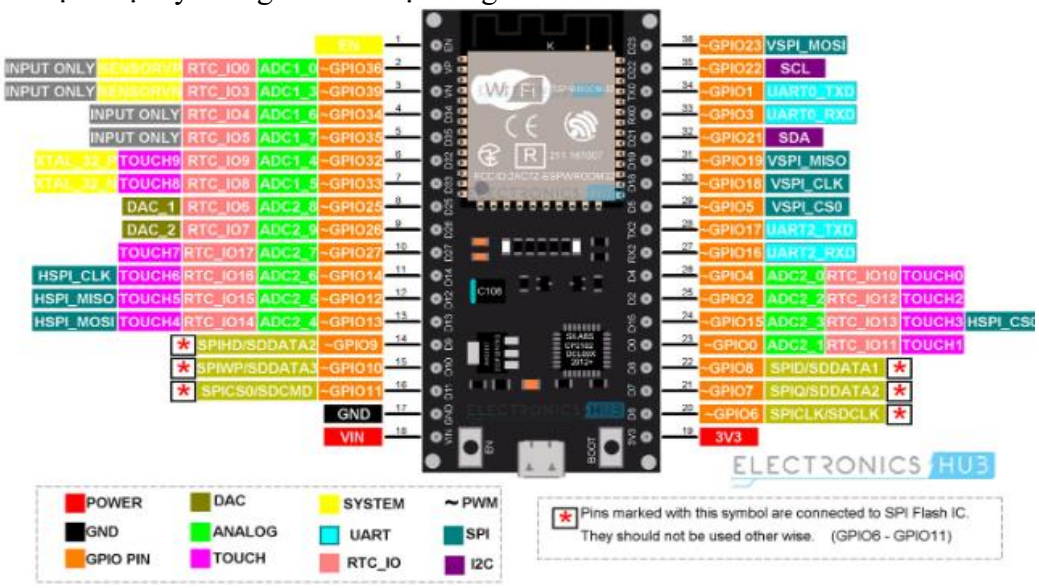
Kết nối đa nền tảng: Các thiết bị có thể giao tiếp với nhau thông qua các nền tảng như Apple HomeKit, Google Home, Samsung SmartThings để tạo ra hệ sinh thái thông minh.

Cải thiện bảo mật: Với việc gia tăng số lượng thiết bị kết nối Internet, bảo mật trong hệ thống điều khiển thông minh ngày càng được chú trọng, với các biện pháp như mã hóa dữ liệu, xác thực người dùng và bảo vệ chống lại các cuộc tấn công mạng.

2. Công nghệ ESP32 và giao thức MQTT

2.1. Giới thiệu về ESP32:

ESP32 là một vi điều khiển tích hợp Wi-Fi và Bluetooth, được thiết kế cho các ứng dụng IoT. Với hiệu suất cao, tiêu thụ năng lượng thấp và khả năng xử lý đa nhiệm, ESP32 là lựa chọn lý tưởng cho các hệ thống điều khiển từ xa.



Khả năng tích hợp Wi-Fi và Bluetooth

ESP32 được trang bị sẵn mô-đun kết nối không dây, giúp nó có thể:

Giao tiếp trực tiếp với Internet thông qua Wi-Fi, mà không cần thêm mô-đun mở rộng.

Giao tiếp gần (short-range) với các thiết bị như điện thoại thông minh, cảm biến, hoặc thiết bị đeo qua **Bluetooth** (hỗ trợ cả Bluetooth truyền thống và BLE – Bluetooth Low Energy).

Việc tích hợp sẵn cả hai công nghệ không dây giúp ESP32 cực kỳ linh hoạt trong các ứng dụng như:

Điều khiển thiết bị qua mạng nội bộ hoặc qua cloud (Wi-Fi).

Giao tiếp với ứng dụng điện thoại hoặc các thiết bị BLE (Bluetooth).

Hiệu suất xử lý cao

ESP32 sử dụng bộ vi xử lý lõi kép (dual-core) **Xtensa LX6** có thể hoạt động ở tốc độ lên tới **240 MHz**, cùng với bộ nhớ SRAM tích hợp khoảng **520 KB**. So với nhiều vi điều khiển khác cùng phân khúc, ESP32 vượt trội hơn về khả năng:

Xử lý đồng thời nhiều tác vụ (multitasking).

Đọc và xử lý tín hiệu cảm biến, giao tiếp với server, điều khiển thiết bị, tất cả trong thời gian thực.

Chạy các ứng dụng có tính toán logic, AI nhỏ, hoặc xử lý tín hiệu (signal processing).

Tiêu thụ năng lượng thấp

Một điểm nổi bật khác của ESP32 là khả năng **tiết kiệm năng lượng**, giúp nó phù hợp với các hệ thống yêu cầu hoạt động lâu dài mà không có nguồn điện cố định. Chip hỗ trợ nhiều chế độ hoạt động:

Active mode: hoạt động tối đa, tiêu tốn nhiều năng lượng.

Modem-sleep / Light-sleep: tạm thời tắt các phần không cần thiết.

Deep-sleep: mức tiêu thụ điện năng cực thấp, phù hợp cho các thiết bị chạy bằng pin.

Điều này giúp ESP32 **rất phù hợp cho các ứng dụng chạy bằng pin như thiết bị giám sát môi trường, cảm biến nông nghiệp, thiết bị đeo tay hoặc camera mini không dây**.

Lựa chọn lý tưởng cho hệ thống điều khiển từ xa

Nhờ kết hợp các yếu tố:

Kết nối mạng linh hoạt (Wi-Fi, Bluetooth)

Hiệu suất xử lý mạnh mẽ

Tiêu thụ năng lượng thấp

Chi phí thấp, dễ sản xuất và triển khai

Hỗ trợ lập trình với Arduino, MicroPython, ESP-IDF...

...ESP32 trở thành **giải pháp phần cứng lý tưởng** cho các hệ thống điều khiển từ xa, cho phép người dùng **giám sát và điều khiển thiết bị qua Internet** từ bất kỳ đâu. Ví dụ:

Người dùng có thể **bật tắt đèn, quạt, hệ thống tưới tiêu** thông qua smartphone.

Có thể **giám sát trạng thái thiết bị (nhiệt độ, độ ẩm, tình trạng bật/tắt)** theo thời gian thực.

Tích hợp với các giao thức như **MQTT, HTTP, WebSocket** để gửi/nhận dữ liệu nhanh chóng và hiệu quả.

2.2. Giao thức MQTT:

MQTT (Message Queuing Telemetry Transport) là một giao thức truyền thông nhẹ, hoạt động theo mô hình publish/subscribe, tối ưu cho các thiết bị có tài nguyên hạn chế và mạng không ổn định. MQTT cho phép các thiết bị gửi và nhận thông điệp thông qua một broker trung gian như Mosquitto.

Publisher (thiết bị xuất bản): Gửi dữ liệu (message) đến một chủ đề (topic) nhất định.

Subscriber (thiết bị đăng ký): Nhận dữ liệu từ broker khi có thiết bị khác gửi dữ liệu đến topic mà nó quan tâm.

Broker (máy chủ trung gian): Tiếp nhận dữ liệu từ publisher và phân phối lại đến các subscriber tương ứng.

2.2.1. Đặc điểm nổi bật của MQTT

Nhẹ và tiết kiệm tài nguyên

MQTT được thiết kế để hoạt động với bộ nhớ và băng thông tối thiểu. Các gói tin trong giao thức này rất nhỏ, phù hợp với các vi điều khiển có dung lượng bộ nhớ và năng lượng giới hạn như ESP32, STM32, Arduino, …

Tối ưu trong môi trường mạng không ổn định

Một trong những lợi thế lớn nhất của MQTT là khả năng làm việc hiệu quả trong môi trường có kết nối mạng không ổn định hoặc có độ trễ cao. Giao thức này hỗ trợ cơ chế giữ kết nối (*keep-alive*), tự động gửi lại thông điệp khi bị mất kết nối và hỗ trợ nhiều mức chất lượng dịch vụ (QoS) khác nhau.

Mở rộng dễ dàng và linh hoạt

Nhờ vào mô hình pub/sub, việc thêm hoặc bớt các thiết bị trong hệ thống trở nên đơn giản. Các thiết bị mới chỉ cần đăng ký đúng chủ đề là có thể nhận được dữ liệu liên quan.

Mức độ đảm bảo dịch vụ (QoS)

MQTT hỗ trợ ba mức độ QoS để đảm bảo độ tin cậy khi truyền dữ liệu:

QoS 0 (At most once): Gửi một lần, không cần xác nhận.

QoS 1 (At least once): Gửi ít nhất một lần, có thể lặp lại nếu chưa được xác nhận.

QoS 2 (Exactly once): Gửi đúng một lần duy nhất, độ tin cậy cao nhất.

2.2.2. MQTT Broker – Trái tim của hệ thống

Broker đóng vai trò trung tâm trong hệ thống MQTT. Nó chịu trách nhiệm tiếp nhận và phân phối tất cả các thông điệp giữa các thiết bị. Một trong những broker phổ biến nhất hiện nay là **Mosquitto**, mã nguồn mở, nhẹ và dễ triển khai trên nhiều nền tảng như Windows, Linux, hoặc Raspberry Pi. Các broker MQTT hiện nay còn tích hợp các tính năng bảo mật như xác thực người dùng, mã hóa TLS, kiểm soát truy cập theo topic.

2.2.3. Ứng dụng của MQTT trong thực tế

Giao thức MQTT đang được ứng dụng rộng rãi trong nhiều lĩnh vực, đặc biệt là:

Nhà thông minh: Điều khiển đèn, quạt, máy lạnh, khóa cửa từ xa thông qua MQTT.

Nông nghiệp thông minh: Gửi dữ liệu cảm biến độ ẩm, nhiệt độ đất, điều khiển tưới tiêu tự động.

Công nghiệp 4.0: Giám sát trạng thái máy móc, điều khiển dây chuyền sản xuất.

Y tế: Truyền dữ liệu bệnh nhân từ thiết bị y tế đến trung tâm điều hành để theo dõi liên tục.

Giao thông thông minh: Quản lý và giám sát tín hiệu giao thông, bãi đỗ xe thông minh.

2.3. Ứng dụng của ESP32 và MQTT trong các hệ thống điều khiển từ xa:

2.3.1. Ưu điểm khi ứng dụng ESP32 kết hợp với MQTT

Sự kết hợp giữa ESP32 và MQTT cho phép xây dựng các hệ thống điều khiển thiết bị từ xa hiệu quả, linh hoạt và tiết kiệm chi phí. ESP32 có thể kết nối với broker MQTT để nhận lệnh điều khiển và gửi trạng thái thiết bị, cho phép người dùng giám sát và điều khiển thiết bị từ bất kỳ đâu có kết nối Internet.

Hiệu quả về chi phí

ESP32 có giá thành rẻ nhưng cung cấp đầy đủ tính năng mạnh mẽ như Wi-Fi, Bluetooth, GPIO, ADC, I2C, SPI… Điều này giúp người phát triển không cần đầu tư phần cứng đắt tiền để xây dựng hệ thống IoT. Đồng thời, MQTT là giao thức mã nguồn mở, nhiều broker miễn phí, giúp giảm chi phí vận hành.

Tính linh hoạt và mở rộng cao

Hệ thống MQTT sử dụng mô hình publish/subscribe nên có thể dễ dàng thêm, bớt thiết bị mà không ảnh hưởng tới toàn hệ thống. Một broker có thể quản lý hàng trăm thiết bị ESP32 cùng lúc.

Giám sát và điều khiển từ xa mọi lúc, mọi nơi

Chỉ cần có Internet, người dùng có thể giám sát và điều khiển thiết bị thông qua MQTT từ bất kỳ đâu – tại nhà, tại cơ quan, hoặc khi đang di chuyển.

Thời gian phản hồi nhanh

MQTT có độ trễ thấp và hỗ trợ truyền dữ liệu theo thời gian thực, giúp hệ thống phản ứng kịp thời với các lệnh điều khiển hoặc thay đổi trạng thái từ cảm biến.

2.3.2. Ứng dụng thực tiễn

Nhà thông minh (Smart Home):

Điều khiển đèn, quạt, điều hòa từ xa qua smartphone.
Hệ thống rèm cửa tự động theo ánh sáng.
Bật tắt thiết bị điện theo lịch hoặc cảm biến chuyển động.

Nông nghiệp thông minh:

Cảm biến đo độ ẩm đất gửi dữ liệu qua MQTT.
ESP32 điều khiển hệ thống tưới nước tự động.
Giám sát khí hậu nhà kính từ xa.

Công nghiệp và tự động hóa:

Giám sát dây chuyền sản xuất.
Điều khiển động cơ, băng chuyền, thiết bị cơ khí từ trung tâm điều khiển.
Báo cáo lỗi hoặc cảnh báo nguy hiểm từ xa.

3. Broker Mosquitto và vai trò trong hệ thống

3.1. Giới thiệu về Mosquitto:

Mosquitto là một broker MQTT mã nguồn mở, nhẹ và dễ triển khai, được sử dụng rộng rãi trong các ứng dụng IoT để quản lý việc truyền tải thông điệp giữa các client.

3.2. Vai trò của Mosquitto trong hệ thống điều khiển từ xa:

Mosquitto đóng vai trò trung gian, tiếp nhận các thông điệp từ các client (như ESP32 hoặc ứng dụng di động) và phân phối chúng đến các client đăng ký tương ứng. Điều này giúp đảm bảo việc truyền tải thông điệp hiệu quả và đáng tin cậy giữa các thành phần trong hệ thống.

3.3. Đặc điểm nổi bật của Mosquitto

Mã nguồn mở và miễn phí
Nhẹ và hiệu quả
Hỗ trợ bảo mật
Dễ cài đặt và cấu hình
Khả năng mở rộng

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG

1. Mô hình hệ thống

1.1. Sơ đồ khối của hệ thống

Thiết bị điều khiển (ESP32) → Kết nối Wi-Fi → **Broker Mosquitto** → Kết nối Wi-Fi → **Ứng dụng điều khiển (di động/web)**

1.2. Nguyên lý hoạt động của hệ thống:

Ứng dụng điều khiển gửi lệnh bật/tắt đến broker Mosquitto thông qua một topic cụ thể. ESP32, đã đăng ký (subscribe) vào topic này, nhận lệnh và điều khiển trạng thái của thiết bị (ví dụ: bật/tắt đèn). Đồng thời, ESP32 có thể gửi trạng thái hiện tại của thiết bị lên một topic khác để ứng dụng điều khiển cập nhật thông tin.

Bước 1: Gửi lệnh điều khiển từ ứng dụng
Bước 2: Broker Mosquitto tiếp nhận lệnh
Bước 3: ESP32 nhận lệnh điều khiển
Bước 4: ESP32 gửi trạng thái lên broker
Bước 5: Ứng dụng nhận trạng thái

2. Thành phần phần cứng

2.1. Danh sách linh kiện và đặc điểm kỹ thuật:

Để xây dựng một hệ thống điều khiển thiết bị điện từ xa sử dụng vi điều khiển ESP32 và giao thức MQTT, cần chuẩn bị một số linh kiện cơ bản. Mỗi linh kiện đảm nhận một vai trò cụ thể trong việc đảm bảo hệ thống hoạt động ổn định, linh hoạt và an toàn.

ESP32 – Vi điều khiển chính

Chức năng: Là bộ não trung tâm của hệ thống, ESP32 thực hiện các chức năng như:

Kết nối Wi-Fi để truy cập Internet.
Giao tiếp với MQTT Broker (Mosquitto).
Nhận lệnh điều khiển và phản hồi trạng thái thiết bị.
Điều khiển tín hiệu đầu ra đến module relay.

Relay Module (Module rơ-le)

Chức năng: Là cầu nối giữa vi điều khiển và thiết bị điện, giúp bật/tắt thiết bị điện gia dụng thông qua tín hiệu điều khiển từ ESP32.

Bộ nguồn 5V

Chức năng: Cung cấp điện áp ổn định cho ESP32 và module relay, đặc biệt quan trọng trong các hệ thống hoạt động liên tục hoặc ngoài trời.

Thiết bị điện cần điều khiển

Chức năng: Là tải đầu ra, được điều khiển từ xa thông qua ESP32 và module relay.

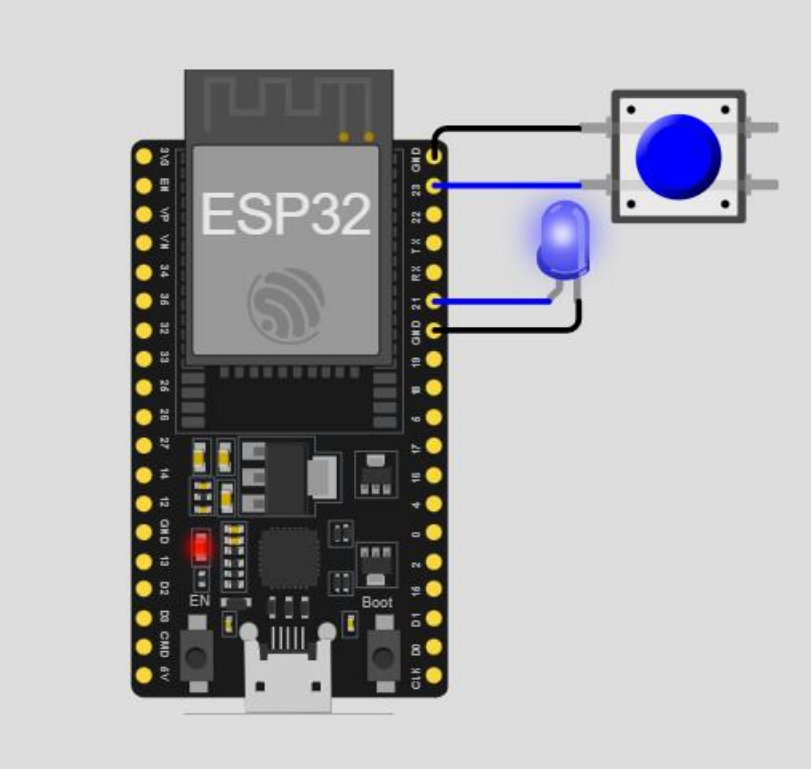
Dây kết nối, đế breadboard (nếu dùng mẫu thử)

Dùng để kết nối các linh kiện trên breadboard trong giai đoạn thử nghiệm.

Có thể thay thế bằng mạch in PCB khi hoàn thiện hệ thống.

2.2. Cách đấu nối phần cứng

Chân điều khiển của relay được kết nối với một chân GPIO trên ESP32. Nguồn VCC và GND của relay được nối tương ứng với 5V và GND. Thiết bị điện được kết nối qua relay để có thể bật/tắt bằng cách điều khiển relay.



Hình ảnh mô phỏng

3. Thành phần phần mềm

3.1. Lập trình điều khiển từ xa qua giao thức MQTT với ESP32

Trong thời đại công nghệ 4.0, các hệ thống nhúng kết nối Internet ngày càng trở nên phổ biến. Một trong những giải pháp phổ biến và hiệu quả cho truyền thông giữa các thiết bị IoT là sử dụng giao thức MQTT – một giao thức nhắn tin nhẹ, được thiết kế tối ưu cho các thiết bị có tài nguyên hạn chế. Trong tiểu luận này, chúng tôi trình bày cách lập trình ESP32 để thực hiện điều khiển thiết bị từ xa qua giao thức MQTT, điển hình là điều khiển đèn LED.

ESP32 được lập trình thông qua **Arduino IDE**, sử dụng các thư viện hỗ trợ như WiFi.h và PubSubClient.h để kết nối và giao tiếp với máy chủ MQTT. Phần mềm chạy trên ESP32 thực hiện chức năng chính là:

Kết nối đến mạng WiFi
Kết nối đến MQTT Broker
Đăng ký (subscribe) vào một topic để chờ lệnh từ xa
Nhận lệnh dạng văn bản như "ON" hoặc "OFF" và điều khiển thiết bị tương ứng (ví dụ: bật/tắt LED)

Việc sử dụng MQTT giúp hệ thống phản hồi nhanh, không cần phải gửi yêu cầu liên tục (polling) như các giao thức truyền thống, giảm tiêu thụ tài nguyên hệ thống.

3.2. MQTT Broker và phần mềm trung gian (Middleware)

Để hệ thống hoạt động trơn tru, cần có một MQTT Broker đóng vai trò trung gian tiếp nhận và phân phối thông điệp giữa các thiết bị. Có 2 lựa chọn:

a. Broker công cộng

Ví dụ: broker.hivemq.com, test.mosquitto.org

Ưu điểm:

Không cần cài đặt
Dễ sử dụng cho thử nghiệm, học tập

Nhược điểm:

Không bảo mật
Dễ bị gián đoạn khi quá tải

b. Broker cài đặt nội bộ (local)

Cài đặt trên máy tính, Raspberry Pi, hoặc VPS qua phần mềm như Mosquitto MQTT Broker

Ưu điểm:

Toàn quyền kiểm soát
Có thể thiết lập bảo mật
Phù hợp cho triển khai thực tế

Ngoài ra, một số phần mềm trung gian (middleware) hoặc công cụ hỗ trợ test/debug MQTT thường được sử dụng là:

MQTT.fx: Giao diện đồ họa để gửi và nhận dữ liệu từ các topic MQTT, rất hữu ích trong giai đoạn kiểm thử.
Node-RED: Công cụ mã nguồn mở cho phép xây dựng luồng xử lý dữ liệu IoT bằng cách kéo thả, tích hợp rất tốt với MQTT.

3.3. Phần mềm giám sát và giao diện người dùng (nếu có)

Để điều khiển và giám sát thiết bị từ xa, có thể xây dựng giao diện người dùng (UI) như:

Ứng dụng web hoặc mobile (sử dụng HTML, JavaScript hoặc Flutter, React Native, …)

Dashboard hiển thị trạng thái (nối với MQTT để hiển thị tình trạng các thiết bị theo thời gian thực)
Home Assistant: Một nền tảng mã nguồn mở mạnh mẽ cho smart home, hỗ trợ MQTT rất tốt, có thể tích hợp trực tiếp với ESP32
Nếu làm đơn giản, có thể dùng web MQTT client như:
HiveMQ Websocket Client

CHƯƠNG 3: TRIỂN KHAI VÀ KIỂM THỬ

1. Cài đặt và nạp chương trình cho ESP32

1.1. Môi trường phát triển (Arduino IDE):

ESP32 là một vi điều khiển mạnh mẽ với khả năng kết nối Wi-Fi và Bluetooth tích hợp, được ứng dụng rộng rãi trong các dự án IoT hiện đại. Để lập trình cho ESP32, có nhiều môi trường phát triển khả dụng như Arduino IDE, PlatformIO, Espressif IoT Development Framework (ESP-IDF), Mỗi nền tảng có thế mạnh riêng, tuy nhiên trong khuôn khổ dự án này, Arduino IDE được lựa chọn nhờ các yếu tố sau:

Giao diện đơn giản, dễ tiếp cận: Phù hợp cho cả người mới học lập trình vi điều khiển lẫn người đã có kinh nghiệm.

Cộng đồng hỗ trợ lớn: Có sẵn rất nhiều ví dụ, diễn đàn, thư viện liên quan đến ESP32.

Tương thích với nhiều hệ điều hành: Chạy ổn định trên Windows, Linux và macOS.

Các bước thiết lập môi trường phát triển cho ESP32 trên Arduino IDE:

Bước 1: Cài đặt Arduino IDE:

Tải về phiên bản mới nhất tại arduino.cc

Khuyến nghị sử dụng phiên bản từ 1.8.10 trở lên để đảm bảo tương thích với ESP32

Bước 2: Cài đặt Board ESP32:

Mở Arduino IDE → File → Preferences

Tại mục Additional Board Manager URLs, thêm đường dẫn:

https://raw.githubusercontent.com/espressif/arduino-esp32/gh-pages/package_esp32_index.json

Sau đó vào Tools → Board → Boards Manager, tìm kiếm "ESP32" và cài đặt board ESP32 by Espressif Systems.

Bước 3: Cài đặt thư viện hỗ trợ:

Vào Sketch → Include Library → Manage Libraries

Tìm và cài đặt các thư viện cần thiết:

WiFi.h: Dùng để kết nối mạng Wi-Fi

PubSubClient.h: Thư viện hỗ trợ giao tiếp MQTT

Bước 4: Cấu hình thông số nạp:

Trong mục Tools, chọn:

Board: ESP32 Dev Module

Upload Speed: 115200

Port: Chọn đúng cổng COM được cấp khi ESP32 kết nối USB vào máy tính

Các thông số khác có thể để mặc định nếu không có yêu cầu đặc biệt

Việc thiết lập đúng môi trường là bước quan trọng giúp việc lập trình và debug ESP32 diễn ra suôn sẻ và ổn định.

1.2. Nạp chương trình:

Sau khi hoàn tất viết mã chương trình điều khiển (ví dụ: kết nối Wi-Fi, điều khiển thiết bị qua MQTT), bước tiếp theo là nạp (upload) chương trình vào ESP32 để thiết bị có thể hoạt động độc lập.

Các bước thực hiện:

Bước 1: Kết nối ESP32 với máy tính:

Dùng dây micro-USB hoặc USB-C tùy phiên bản board

Đảm bảo driver đã được cài đặt để máy tính nhận diện board

Bước 2: Chọn cổng kết nối đúng (COM Port):

Vào Tools → Port, chọn cổng có dạng COMx (ESP32 Dev Module) hoặc tương tự

Bước 3: Tiến hành nạp chương trình:

Nhấn nút Upload trên Arduino IDE

Trong một số trường hợp, nếu ESP32 không tự động vào chế độ bootloader, người dùng cần nhấn và giữ nút "BOOT" trên board cho đến khi thấy quá trình nạp bắt đầu trên cửa sổ log

Bước 4: Kiểm tra kết quả nạp:

Nếu nạp thành công, Arduino IDE sẽ hiển thị dòng "Done uploading"

Người dùng có thể mở Serial Monitor (Ctrl + Shift + M) để theo dõi hoạt động của chương trình theo thời gian thực, ví dụ như thông báo kết nối Wi-Fi thành công, phản hồi từ MQTT, tín hiệu thiết bị, …

2. Kiểm thử hệ thống

Sau khi lập trình và nạp chương trình thành công cho ESP32, bước tiếp theo là tiến hành kiểm thử hệ thống để đảm bảo tất cả các thành phần – từ kết nối mạng đến giao tiếp MQTT và điều khiển thiết bị – đều hoạt động như mong đợi. Giai đoạn này đóng vai trò quan trọng nhằm phát hiện và khắc phục các lỗi tiềm ẩn trước khi đưa hệ thống vào hoạt động thực tế.

2.1. Kiểm tra kết nối Wi-Fi:

Mục tiêu đầu tiên là xác định **ESP32 đã kết nối thành công với mạng Wi-Fi nội bộ**, vì toàn bộ hệ thống IoT hoạt động dựa trên nền tảng kết nối Internet.

Các bước kiểm tra:

Bước 1: Mở Serial Monitor trên Arduino IDE (Ctrl + Shift + M) sau khi nạp chương trình xong.

Bước 2: Quan sát các dòng thông báo. Nếu kết nối thành công, ESP32 sẽ hiển thị dòng:

Connected to WiFi.

IP address: 192.168.x.x

Đây là địa chỉ IP mà router cấp cho ESP32 thông qua DHCP.

Bước 3: Đăng nhập vào router (qua địa chỉ gateway, thường là 192.168.1.1) để kiểm tra danh sách thiết bị kết nối. Nếu ESP32 xuất hiện trong danh sách, thường sẽ được hiển thị tên như "espressif", "ESP_XXXX" hoặc "unknown device".

Ý nghĩa của bước này:

Đảm bảo ESP32 đã được cấp IP và có thể giao tiếp với các thiết bị trong mạng nội bộ.

Là điều kiện tiên quyết để tiếp tục kết nối với MQTT Broker, vốn thường nằm trên Internet hoặc trong cùng mạng nội bộ.

2.2. Kiểm tra MQTT:

Sau khi kết nối Wi-Fi, ESP32 tiến hành kết nối đến MQTT Broker. Có thể sử dụng phần mềm MQTT.fx hoặc HiveMQ Web Client để kiểm thử:

Đăng ký (subscribe) vào topic: esp32/led

Gửi (publish) dữ liệu "ON" hoặc "OFF" đến topic

Quan sát log phản hồi trên Serial Monitor

Nếu kết nối thành công, ESP32 sẽ hiển thị "Message received: ON" và thực hiện hành động điều khiển thiết bị tương ứng.

2.3. Kiểm tra điều khiển thiết bị:

Phần cuối cùng của kiểm thử là **xác thực khả năng điều khiển thiết bị vật lý** (ví dụ: LED, relay, quạt mini) dựa trên dữ liệu nhận từ MQTT.

Các bước kiểm tra:

Bước 1: Kết nối thiết bị với ESP32 qua chân GPIO được lập trình (ví dụ: GPIO 2).

Bước 2: Khi ESP32 nhận "ON" từ MQTT, chân GPIO sẽ xuất mức HIGH, khiến đèn LED sáng.

Bước 3: Khi nhận "OFF", chân GPIO sẽ xuống mức LOW → đèn tắt.

Dấu hiệu hoạt động đúng:

Gửi "ON" → đèn sáng

Gửi "OFF" → đèn tắt

Phản hồi trên Serial Monitor khớp với hành động thiết bị

Ý nghĩa kỹ thuật:

Xác nhận hệ thống hoạt động từ đầu vào (MQTT message) → xử lý (ESP32) → đầu ra (thiết bị vật lý).

Độ trễ phản hồi rất thấp (gần như tức thì), thể hiện **ưu thế của giao thức MQTT** so với HTTP trong ứng dụng IoT real-time.

3. Một số cải tiến và mở rộng

3.1. Tích hợp thêm cảm biến:

Hệ thống hiện tại chỉ thực hiện điều khiển một chiều (đi từ người dùng → thiết bị). Để mở rộng, có thể tích hợp các cảm biến như:

Cảm biến nhiệt độ/độ ẩm DHT11, DHT22

Cảm biến ánh sáng, chuyển động (PIR)

ESP32 sẽ publish dữ liệu cảm biến lên các topic MQTT, giúp xây dựng hệ thống giám sát từ xa, kết hợp điều khiển và cảnh báo thông minh.

3.2. Điều khiển qua ứng dụng di động:

Thay vì sử dụng phần mềm kỹ thuật như MQTT.fx, có thể xây dựng ứng dụng mobile thân thiện bằng các nền tảng như:

Blynk: giao diện kéo-thả, hỗ trợ MQTT và ESP32

MIT App Inventor: tạo app Android đơn giản

Flutter/React Native: tạo app chuyên nghiệp hơn

Điều này giúp người dùng cuối có thể điều khiển thiết bị mọi lúc, mọi nơi, chỉ bằng vài thao tác trên điện thoại.

3.3. Bảo mật hệ thống:

Trong môi trường thực tế, bảo mật là yếu tố then chốt. Một số hướng nâng cấp bảo mật gồm:

Sử dụng **MQTT có mã hóa TLS/SSL**

Thiết lập tài khoản và mật khẩu truy cập broker

Giới hạn IP kết nối đến broker (nếu dùng local)

Mã hóa nội dung payload, xác thực message

Ngoài ra, có thể sử dụng nền tảng như **AWS IoT Core**, **Google Cloud IoT** để có mức bảo mật và ổn định cao hơn.

CHƯƠNG 4: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

1.1. Tổng kết

Hệ thống điều khiển thiết bị từ xa sử dụng ESP32 và MQTT đã được thiết kế và triển khai thành công, cho phép điều khiển và giám sát thiết bị qua mạng Internet một cách hiệu quả.

1.2. Đánh giá hiệu quả:

Hệ thống hoạt động ổn định, thời gian phản hồi nhanh và có khả năng mở rộng để tích hợp thêm các thiết bị và cảm biến khác.

2. Hướng phát triển trong tương lai

Cải thiện độ tin cậy: Nâng cao khả năng xử lý lỗi và khôi phục sau sự cố mạng.

Tối ưu hóa hiệu suất: Giảm thiểu tiêu thụ năng lượng và tối ưu hóa băng thông sử dụng.

Mở rộng tính năng: Thêm các tính năng như lập lịch điều khiển, thông báo qua email hoặc tin nhắn khi có sự kiện đặc biệt.

Tích hợp AI: Sử dụng trí tuệ nhân tạo để phân tích dữ liệu thu thập được, dự đoán và tự động hóa các quy trình điều khiển.

Tài Liệu Tham Khảo

Hướng dẫn MQTT Arduino ESP32

Lập trình ESP32 với MQTT – Giới thiệu

Điều khiển thiết bị qua internet dùng ESP32 và Blynk IOT

Hướng dẫn lập trình ESP32 và các dự án IoT liên quan

Tổng hợp hướng dẫn Internet of Things với NodeMCU ESP8266 và ESP32

Lập trình IoT cho ESP32 sử dụng ESP-IDF, Node-RED, MQTT đi từ cơ bản

ESP32 và IoT Khám phá Tiềm Năng Kết Nối Van Vật

Internet of Things: Điều khiển, giám sát thiết bị gia đình từ xa

Điều khiển thiết bị qua internet dùng ESP32 và Blynk IOT