

TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN



TIỂU LUẬN MÔN HỌC
PHÁT TRIỂN ỨNG DỤNG IOT

ĐIỀU KHIỂN RÈM CỬA TỰ ĐỘNG VỚI ESP32

TÊN LỚP HỌC PHẦN: PHÁT TRIỂN ỨNG DỤNG IOT

MÃ HỌC PHẦN: TIN4024.005

Giảng viên hướng dẫn : Võ Việt Dũng

Sinh viên thực hiện : Lê Hữu Minh Duệ

HUẾ, THÁNG 3 NĂM 2025

Mục Lục

| | |
|---|----|
| Mục Lục | 1 |
| PHẦN 1: GIỚI THIỆU | 1 |
| 1.1. Bối cảnh nhà thông minh và tự động hóa. | 1 |
| 1.2. Tầm quan trọng của tự động hóa rèm cửa. | 1 |
| 1.3. Tổng quan dự án | 2 |
| 1.4. Mục tiêu và phạm vi tiểu luận | 2 |
| 1.5. Sơ đồ khối tổng quan hệ thống | 3 |
| PHẦN 2: TỔNG QUAN CÔNG NGHỆ | 4 |
| 2.1. Giới thiệu về nền tảng mô phỏng Wokwi | 4 |
| 2.2. ESP32: đặc điểm, tính năng và ưu điểm | 5 |
| 2.3. Động cơ bước: nguyên lý hoạt động và ứng dụng | 6 |
| 2.4. Driver A4988: cách thức hoạt động và vai trò | 8 |
| 2.5. Các nền tảng IoT: ThingSpeak, Blynk, Telegram | 9 |
| 2.6. Các cảm biến sử dụng | 10 |
| PHẦN 3: YÊU CẦU HỆ THỐNG | 11 |
| 3.1. Yêu cầu chức năng | 11 |
| 3.2. Yêu cầu phi chức năng | 12 |
| PHẦN 4: THIẾT KẾ PHẦN CỨNG | 13 |
| 4.1. Thành phần phần cứng và lựa chọn | 13 |
| 4.2. Sơ đồ chi tiết kết nối Wokwi | 14 |
| 4.3. Mô tả cách thức mô phỏng các thành phần | 15 |
| PHẦN 5: PHÂN TÍCH HOẠT ĐỘNG | 15 |
| 5.1. Sơ đồ thuật toán tổng quát | 15 |
| 5.2. Sơ đồ luồng dữ liệu | 16 |
| 5.3. Mô tả ngắn và demo cấu trúc code chính | 16 |
| 5.3.1. Cảm biến ánh sáng | 17 |
| 5.3.2. Điều khiển động cơ bước A4988 | 17 |
| 5.3.3. Điều khiển qua Blynk | 17 |
| 5.3.4. Điều khiển bằng nút nhấn vật lý | 18 |
| 5.3.5. Điều khiển qua Telegram | 18 |
| 5.3.6. Chế độ tự động | 18 |
| 5.4. Biểu đồ trạng thái hệ thống | 19 |
| PHẦN 6: MÔ PHỎNG WOKWI | 20 |
| 6.1. Quy trình thiết lập dự án Wokwi | 20 |
| 6.2. Cấu hình các thành phần | 21 |
| 6.2.1. Cấu hình cảm biến ánh sáng | 21 |
| 6.2.2. Cấu hình Driver A4988 | 21 |
| 6.2.3. Cấu hình động cơ bước | 21 |
| 6.2.4. Cấu hình nút nhấn điều khiển | 22 |
| 6.3. Hình ảnh chi tiết từ mô phỏng Wokwi | 22 |
| 6.4. Bảng kết quả mô phỏng với các điều kiện môi trường khác nhau | 23 |
| 6.5. Mô tả các vấn đề phát sinh và giải pháp trong quá trình mô phỏng | 23 |
| 6.5.1. Vấn đề về điều khiển độ chính xác của động cơ bước | 23 |
| 6.5.2. Vấn đề về độ trễ trong việc phản hồi từ nền tảng IoT | 23 |
| 6.5.3. Vấn đề về tiêu thụ năng lượng của động cơ | 23 |
| 6.5.4. Vấn đề về độ nhạy của cảm biến ánh sáng | 24 |
| 6.5.5. Vấn đề về kết nối WiFi trong mô phỏng | 24 |
| PHẦN 7: TÍCH HỢP IOT | 24 |
| 7.1. Tích hợp với Blynk | 24 |
| 7.1.1. Hình ảnh giao diện Blynk thiết kế cho dự án | 24 |
| 7.2. Tích hợp với Telegram | 25 |
| 7.3. Bảng so sánh hiệu suất và độ trễ của các phương thức giao tiếp | 26 |
| PHẦN 8: KẾT QUẢ MÔ PHỎNG VÀ THẢO LUẬN | 26 |
| 8.1. Demo mô phỏng trên Wokwi, Blynk | 26 |
| 8.2. Demo mô phỏng trên Wokwi, Telegram | 26 |
| PHẦN 9: CẢI TIẾN VÀ MỞ RỘNG | 27 |
| 9.1. Cải tiến hệ thống hiện tại | 27 |
| 9.2. Mở rộng chức năng | 28 |

| | |
|---|----|
| 9.3. Sơ đồ kiến trúc hệ thống mở rộng | 29 |
| 9.4. Bảng kết quả dự đoán sau cải tiến (mô phỏng) | 29 |
| PHẦN 10: KẾT LUẬN | 30 |
| 1. Tóm tắt dự án và kết quả đạt được | 30 |
| 2. Bài học kinh nghiệm rút ra từ dự án bao gồm: | 30 |
| 3. Hướng phát triển tương lai | 31 |
| PHẦN 11: TÀI LIỆU THAM KHẢO VÀ PHỤ LỤC | 31 |

PHẦN 1: GIỚI THIỆU

1.1. Bối cảnh nhà thông minh và tự động hóa.

Trong thời đại công nghệ số phát triển mạnh mẽ, khái niệm nhà thông minh (Smart Home) đã dần trở thành xu hướng và nhu cầu thiết yếu trong cuộc sống hiện đại. Nhà thông minh là hệ thống kết nối các thiết bị trong gia đình thông qua mạng internet, cho phép người dùng điều khiển từ xa hoặc tự động hóa các thiết bị gia dụng, hệ thống chiếu sáng, an ninh, điều hòa không khí và các thiết bị khác. Theo báo cáo của GlobalData, thị trường nhà thông minh toàn cầu đã đạt 80 tỷ USD vào năm 2023 và dự kiến sẽ tăng lên 150 tỷ USD vào năm 2026, với tốc độ tăng trưởng kép hàng năm (CAGR) là 15%.

Tự động hóa là một trong những khía cạnh quan trọng nhất của nhà thông minh, giúp tối ưu hóa việc sử dụng năng lượng, nâng cao tiện nghi và tiết kiệm thời gian. Thay vì điều khiển thủ công, các thiết bị trong hệ thống nhà thông minh có thể hoạt động dựa trên các điều kiện môi trường, thời gian trong ngày hoặc thói quen của người dùng. Đây chính là xu hướng phát triển tất yếu của công nghệ nhà ở hiện đại, đặc biệt trong bối cảnh Internet of Things (IoT) đang phát triển mạnh mẽ.

1.2. Tầm quan trọng của tự động hóa rèm cửa.

Rèm cửa đóng vai trò quan trọng trong việc điều tiết ánh sáng, nhiệt độ và đảm bảo tính riêng tư cho ngôi nhà. Việc tự động hóa rèm cửa mang lại nhiều lợi ích đáng kể:

- **Tiết kiệm năng lượng:** Rèm tự động có thể mở hoặc đóng dựa trên nhiệt độ và cường độ ánh sáng, giúp giảm nhu cầu sử dụng điều hòa không khí và ánh sáng nhân tạo. Theo một nghiên cứu của Bộ Năng lượng Hoa Kỳ, việc sử dụng rèm thông minh có thể giảm tới 20% chi phí năng lượng điều hòa trong mùa hè.
- **Tăng cường tiện nghi:** Người dùng không cần phải đi khắp nhà để đóng/mở rèm, đặc biệt là trong nhà có nhiều cửa sổ hoặc với người già, người khuyết tật.
- **Bảo vệ nội thất:** Đóng rèm tự động khi có ánh nắng mạnh giúp bảo vệ đồ nội thất, sàn gỗ và các vật dụng khác khỏi sự phai màu do tia UV.

- **Tăng cường an ninh:** Rèm cửa có thể được lập trình để đóng/mở theo lịch trình, tạo cảm giác có người ở nhà ngay cả khi đi vắng, góp phần ngăn chặn trộm cắp.
- **Tích hợp với hệ sinh thái nhà thông minh:** Rèm tự động có thể hoạt động đồng bộ với các thiết bị khác như đèn, điều hòa để tối ưu hóa không gian sống.

1.3. Tổng quan dự án

Dự án "Điều khiển rèm cửa tự động với ESP32" nhằm thiết kế và mô phỏng một hệ thống rèm cửa thông minh sử dụng vi điều khiển ESP32, động cơ bước và các cảm biến. Hệ thống được thiết kế với ba chế độ hoạt động chính:

1. Chế độ tự động: Hệ thống tự động điều chỉnh rèm dựa trên cường độ ánh sáng môi trường - khi trời sáng thì đóng rèm, khi trời tối thì mở rèm.
2. Chế độ bán tự động: Người dùng có thể điều khiển rèm thông qua nút nhấn vật lý - mỗi lần nhấn sẽ chuyển đổi trạng thái đóng/mở.
3. Chế độ điều khiển từ xa: Hệ thống có thể được điều khiển thông qua các nền tảng IoT như Blynk và Telegram, cho phép người dùng kiểm soát rèm từ bất kỳ đâu thông qua internet.

Dự án sử dụng nền tảng mô phỏng Wokwi để thiết kế và thử nghiệm hệ thống trước khi triển khai thực tế, giúp tiết kiệm thời gian và chi phí phát triển.

1.4. Mục tiêu và phạm vi tiểu luận

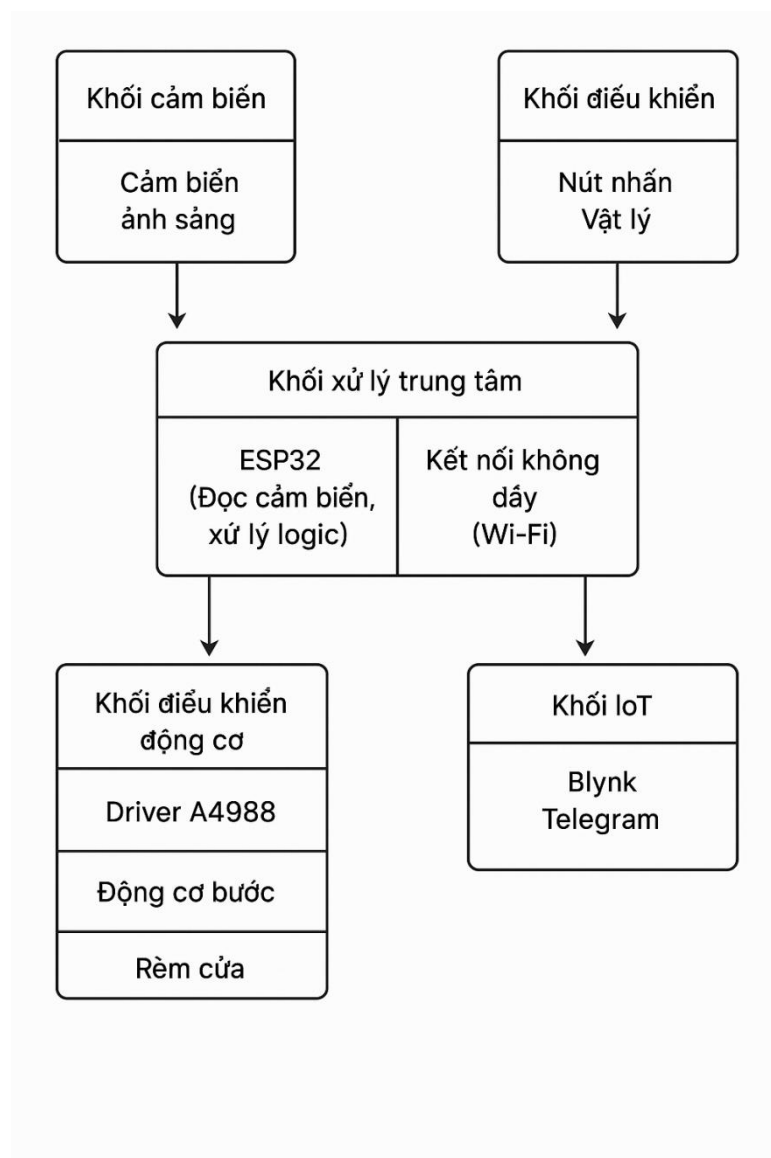
Tiểu luận này nhằm mục đích:

- Nghiên cứu và phân tích các thành phần phần cứng cần thiết cho hệ thống điều khiển rèm cửa tự động.
- Thiết kế sơ đồ mạch điện và kết nối giữa ESP32, động cơ bước và các cảm biến.
- Xây dựng mô phỏng hoàn chỉnh trên nền tảng Wokwi.
- Phân tích và đánh giá hiệu suất của hệ thống thông qua mô phỏng.

- Tích hợp hệ thống với các nền tảng IoT (Blynk, Telegram).
- Đề xuất các cải tiến và mở rộng cho hệ thống trong tương lai.

Phạm vi của tiểu luận giới hạn ở việc thiết kế và mô phỏng, không bao gồm việc triển khai thực tế. Các thành phần được sử dụng trong tiểu luận sẽ dựa trên các module có sẵn trong nền tảng Wokwi hoặc tương đương. Đối với các cảm biến không có sẵn trong Wokwi, tiểu luận sẽ sử dụng giá trị ngẫu nhiên hoặc mô phỏng giả lập.

1.5. Sơ đồ khối tổng quan hệ thống



Sơ đồ khối trên minh họa cấu trúc tổng quan của hệ thống điều khiển rèm cửa tự động, bao gồm 4 khối chính:

1.Khối cảm biến: Thu thập dữ liệu về ánh sáng môi trường.

2.Khối điều khiển: Cung cấp giao diện vật lý cho người dùng thông qua nút nhấn.

3.Khối xử lý trung tâm: ESP32 xử lý dữ liệu từ cảm biến, nút nhấn và kết nối không dây.

4.Khối điều khiển động cơ: Điều khiển động cơ bước thông qua driver A4988 để mở/đóng rèm.

5.Khối IoT/Cloud: Cung cấp giao diện điều khiển từ xa và hiển thị dữ liệu qua các nền tảng như Blynk, Telegram.

Hệ thống hoạt động theo nguyên tắc: ESP32 liên tục đọc dữ liệu từ cảm biến ánh sáng, nút nhấn và lệnh từ các nền tảng IoT, sau đó ra quyết định điều khiển động cơ bước thông qua driver A4988 để đóng hoặc mở rèm cửa. Đồng thời, ESP32 cũng gửi dữ liệu về trạng thái của hệ thống lên các nền tảng IoT để người dùng có thể theo dõi và điều khiển từ xa.

PHẦN 2: TỔNG QUAN CÔNG NGHỆ

2.1. Giới thiệu về nền tảng mô phỏng Wokwi

Wokwi là một nền tảng mô phỏng trực tuyến cho các dự án điện tử và IoT, cho phép thiết kế, mô phỏng và kiểm thử mạch điện tử mà không cần phần cứng vật lý. Nền tảng này mang lại nhiều lợi ích đáng kể cho việc phát triển dự án:

Lợi ích chính:

- Tiết kiệm chi phí: Không cần đầu tư vào phần cứng trong giai đoạn thử nghiệm ban đầu
- Tiết kiệm thời gian: Thiết lập nhanh chóng và thay đổi cấu hình dễ dàng
- An toàn: Không có nguy cơ hư hỏng thiết bị do kết nối sai

- Dễ chia sẻ: Các dự án có thể được chia sẻ trực tuyến với người khác
- Đa nền tảng: Hoạt động trên mọi thiết bị có kết nối internet

Khả năng của Wokwi:

- Hỗ trợ nhiều vi điều khiển phổ biến (ESP32, Arduino, Raspberry Pi Pico)
- Mô phỏng chính xác hoạt động của cảm biến và actuator
- Hỗ trợ giao tiếp I2C, SPI, UART và các giao thức khác
- Có thể kết nối với các dịch vụ cloud thông qua API
- Khả năng lập trình và upload code trực tiếp từ trình duyệt

Với những tính năng trên, Wokwi là công cụ lý tưởng để phát triển và thử nghiệm hệ thống điều khiển rèm cửa tự động trước khi triển khai thực tế.

Trong dự án này, Wokwi giúp mô phỏng toàn bộ hệ thống điều khiển rèm cửa gồm: ESP32, động cơ bước bipolar + driver A4988, nút nhấn, và cảm biến ánh sáng.

2.2. ESP32: đặc điểm, tính năng và ưu điểm

ESP32 là một vi điều khiển mạnh mẽ, tiết kiệm năng lượng được phát triển bởi Espressif Systems. Đây là lựa chọn phù hợp cho dự án điều khiển rèm cửa tự động vì nhiều lý do:

Đặc điểm chính:

- Vi xử lý lõi kép Tensilica Xtensa LX6 với tần số lên đến 240MHz
- Tích hợp Wi-Fi (802.11 b/g/n) và Bluetooth (cả Classic và BLE)
- ROM 448KB, SRAM 520KB
- Hỗ trợ nhiều giao tiếp ngoại vi (I2C, SPI, UART, ADC, DAC, PWM)
- 36 chân GPIO có thể lập trình

Ưu điểm:

- Hiệu năng cao với mức tiêu thụ điện thấp

- Khả năng kết nối đa dạng phù hợp cho ứng dụng IoT
- Giá thành phải chăng
- Hỗ trợ cộng đồng mạnh mẽ
- Tương thích với nhiều thư viện và framework phổ biến

Trong hệ thống, ESP32 có vai trò:

- Đọc dữ liệu từ cảm biến ánh sáng (LDR)
- Nhận tín hiệu nút nhấn vật lý
- Điều khiển driver A4988 để quay động cơ bước
- Giao tiếp IoT với Blynk / Telegram.

Bảng thông số kỹ thuật chi tiết của ESP32:

| Thông số | Giá trị |
|-----------------------|--------------------------------------|
| Vi xử lý | Tensilica Xtensa LX6 lõi kép, 32-bit |
| Tần số xử lý | Lên đến 240 MHz |
| Bộ nhớ RAM | 520 KB SRAM |
| Bộ nhớ Flash | Hỗ trợ đến 16 MB |
| Bluetooth | Bluetooth 4.2 và BLE |
| Wi-Fi | 802.11 b/g/n (2.4 GHz) |
| GPIO | 36 chân |
| ADC | 12-bit, 18 kênh |
| DAC | 8-bit, 2 kênh |
| Giao tiếp | SPI, I2C, I2S, UART, CAN |
| Cảm biến tích hợp | Cảm biến nhiệt độ, cảm biến Hall |
| Bảo mật | Mã hóa phần cứng (AES, SHA, RSA) |
| Chế độ tiết kiệm điện | Deep sleep ($\leq 10 \mu A$) |
| Nguồn điện | 2.3V đến 3.6V |
| Nhiệt độ hoạt động | -40°C đến +125°C |

2.3. Động cơ bước: nguyên lý hoạt động và ứng dụng

Động cơ bước là thiết bị điện-cơ chuyển đổi xung điện thành chuyển động cơ học theo góc xoay cố định. Chúng là lựa chọn lý tưởng cho hệ thống điều khiển rèm cửa tự động vì khả năng định vị chính xác và điều khiển vị trí.

Nguyên lý hoạt động:

- Động cơ bước hoạt động trên nguyên lý từ trường, với rotor có nam châm vĩnh cửu và stator có các cuộn dây điện từ
- Khi các cuộn dây được cấp điện theo trình tự nhất định, chúng tạo ra từ trường xoay
- Rotor phản ứng với từ trường này, quay một góc xác định (bước)
- Việc điều khiển số xung và thứ tự cấp điện cho các cuộn dây sẽ quyết định hướng và tốc độ quay

Ứng dụng trong hệ thống rèm cửa tự động:

- Điều khiển chính xác vị trí rèm (mở/đóng đến các vị trí xác định)
- Điều chỉnh tốc độ đóng/mở rèm
- Lưu trữ vị trí hiện tại và phục hồi khi cần thiết
- Vận hành êm ái và độ bền cao

Bảng so sánh các loại động cơ bước phổ biến:

| Loại | Ưu điểm | Nhược điểm | Góc bước | Mô-men | Ứng dụng phù hợp |
|--------------------------|--|--------------------------------|--------------------|------------|---|
| Permanent Magnet (PM) | Đơn giản, chi phí thấp | Mô-men thấp, độ phân giải thấp | 7.5° - 15° | Thấp | Thiết bị điều khiển đơn giản, máy in |
| Variable Reluctance (VR) | Tốc độ cao, không có mô-men giữ | Thiếu mô-men giữ | 1.5° - 30° | Trung bình | Ứng dụng tốc độ cao, đồng hồ |
| Hybrid (HB) | Mô-men cao, độ phân giải cao | Chi phí cao | 0.9° - 3.6° | Cao | Máy CNC, robot, thiết bị chính xác, rèm cửa |
| Microstepping | Chuyển động rất mượt, độ phân giải cực cao | Cần driver phức tạp | 1/2 đến 1/256 bước | Biến đổi | Thiết bị chính xác cao, thiết bị y tế |

2.4. Driver A4988: cách thức hoạt động và vai trò

Driver A4988 là một module điều khiển động cơ bước được sử dụng rộng rãi trong các dự án điện tử và tự động hóa. Nó đóng vai trò quan trọng trong việc điều khiển động cơ bước cho hệ thống rèm cửa tự động.

Cách thức hoạt động:

- A4988 chuyển đổi các tín hiệu điều khiển đơn giản (STEP và DIR) từ ESP32 thành các xung điện phức tạp cần thiết để điều khiển động cơ bước
- Mỗi xung vào chân STEP làm động cơ quay một bước
- Trạng thái của chân DIR quyết định hướng quay
- A4988 có khả năng vi bước (microstepping), cho phép chia nhỏ mỗi bước thành nhiều bước nhỏ hơn

Vai trò trong hệ thống rèm cửa tự động:

- Đơn giản hóa việc điều khiển động cơ bước từ ESP32
- Cung cấp đủ dòng điện cho động cơ (ESP32 không thể cung cấp trực tiếp)
- Bảo vệ vi điều khiển khỏi điện áp ngược từ động cơ
- Cho phép điều khiển chính xác vị trí của rèm thông qua tính năng vi bước

Thông số kỹ thuật của A4988:

| Thông số | Giá trị |
|--------------------|------------------------------------|
| Điện áp hoạt động | 8–35V (động cơ), 3–5.5V (logic) |
| Dòng điện tối đa | 2A/pha (có tản nhiệt) |
| Chế độ vi bước | Full, 1/2, 1/4, 1/8, 1/16 |
| Bảo vệ | Quá nhiệt, quá dòng, điện áp ngược |
| Điều khiển đầu vào | STEP, DIR, ENABLE, MS1, MS2, MS3 |
| Điện trở cảm biến | 0.05Ω (cho điều khiển dòng) |
| Tần số xung tối đa | 4MHz |
| Kích thước | 20mm x 15mm x 10mm |

A4988 là lựa chọn phù hợp cho dự án điều khiển rèm cửa tự động vì khả năng điều khiển động cơ bước chính xác, dễ dàng tích hợp với ESP32, và có sẵn nhiều tính năng bảo vệ mạch.

2.5 Các nền tảng IoT: ThingSpeak, Blynk, Telegram

Việc tích hợp các nền tảng IoT cho phép người dùng điều khiển và giám sát hệ thống rèm cửa thông qua internet từ bất kỳ đâu.

ThingSpeak:

- Nền tảng IoT mã nguồn mở của MathWorks
- Chuyên về thu thập, phân tích và trực quan hóa dữ liệu
- Hỗ trợ phân tích dữ liệu với MATLAB
- Giao diện API RESTful cho phép gửi và nhận dữ liệu qua HTTP

Blynk:

- Nền tảng IoT với giao diện người dùng dễ sử dụng
- Cho phép tạo ứng dụng điều khiển di động mà không cần kiến thức lập trình ứng dụng
- Hỗ trợ nhiều widget để hiển thị và điều khiển

- Kết nối với đa dạng phần cứng qua thư viện chuyên dụng

Telegram:

- Nền tảng nhắn tin với khả năng tạo bot
- Cho phép điều khiển thiết bị IoT từ xa thông qua chat
- API đơn giản và dễ tích hợp
- Hỗ trợ đa nền tảng (iOS, Android, Web)

Bảng so sánh ưu nhược điểm của từng nền tảng:

| Nền tảng | Ưu điểm | Nhược điểm | Tính năng nổi bật | Phù hợp cho |
|-------------------|--|---|--|--|
| Blynk | <ul style="list-style-type: none"> - Giao diện đồ họa dễ dùng - Tiện ích widget có sẵn - Ứng dụng di động - Độ trễ thấp | <ul style="list-style-type: none"> - Giới hạn miễn phí - Tùy biến ít trong giao diện | <ul style="list-style-type: none"> - Điều khiển trực quan - Thông báo đầy - Bảng điều khiển tùy chỉnh | Điều khiển trực tiếp, giao diện thân thiện |
| ThingSpeak | <ul style="list-style-type: none"> - Phân tích dữ liệu mạnh mẽ - Tích hợp MATLAB - Lưu trữ dữ liệu lâu dài - API mở | <ul style="list-style-type: none"> - Độ trễ cao hơn - Giao diện người dùng ít trực quan | <ul style="list-style-type: none"> - Phân tích xu hướng - Trực quan hóa dữ liệu - Khai thác dữ liệu | Theo dõi dài hạn, phân tích dữ liệu |
| Telegram | <ul style="list-style-type: none"> - Bảo mật cao - Không giới hạn miễn phí - Không cần máy chủ riêng - Dùng được trên nhiều thiết bị | <ul style="list-style-type: none"> - Cần lập trình bot - Giao diện chỉ dựa trên văn bản - Không có phân tích dữ liệu | <ul style="list-style-type: none"> - Lệnh văn bản - Thông báo tức thời - Xác thực người dùng - Gửi hình ảnh, tệp | Điều khiển từ xa, thông báo, bảo mật cao |

2.6 Các cảm biến sử dụng

Dự án sử dụng một số cảm biến chính để tự động hóa hoạt động của rèm cửa, trong đó cảm biến ánh sáng đóng vai trò quan trọng nhất.

Cảm biến ánh sáng dùng là quang trở (LDR) – loại cảm biến thay đổi điện trở theo cường độ ánh sáng.

| Thông số | Giá trị |
|-------------------|---------------------------|
| Điện áp hoạt động | 3.3V – 5V |
| Tín hiệu đầu ra | Analog (qua mạch chia áp) |
| Dải trở | 1kΩ (sáng) → 100kΩ (tối) |

Nguyên lý hoạt động:

- Khi sáng: điện trở giảm → điện áp chia áp tăng
- Khi tối: điện trở tăng → điện áp chia áp giảm

ESP32 dùng ADC (chuyển đổi tương tự-số) để đọc giá trị điện áp và xác định mức ánh sáng → từ đó điều khiển đóng/mở rèm.

PHẦN 3: YÊU CẦU HỆ THỐNG

3.1. Yêu cầu chức năng

Hệ thống điều khiển rèm cửa tự động với ESP32 cần đáp ứng các yêu cầu chức năng sau:

1. Tự động điều chỉnh rèm cửa dựa trên cường độ ánh sáng:
 - Tự động mở rèm khi cường độ ánh sáng dưới ngưỡng cài đặt trong khoảng thời gian quy định
 - Tự động đóng rèm khi cường độ ánh sáng vượt quá ngưỡng cài đặt
2. Điều khiển thủ công thông qua nút bấm:
 - Cho phép người dùng điều khiển rèm cửa bằng nút bấm vật lý
 - Hỗ trợ chức năng mở/đóng/dừng rèm khi nhấn nút
3. Điều khiển từ xa qua Internet:
 - Cho phép người dùng điều khiển rèm cửa từ xa thông qua ứng dụng Blynk

- Hỗ trợ gửi lệnh điều khiển và nhận thông báo qua Telegram Bot

4. Hiện thị và giám sát:

- Hiện thị cường độ ánh sáng hiện tại
- Hiện thị trạng thái hiện tại của rèm cửa (mở, đóng, đang di chuyển)

5. Cài đặt lịch trình:

- Cho phép người dùng cài đặt lịch trình tự động mở/đóng rèm theo thời gian
- Hỗ trợ cài đặt ngưỡng cường độ ánh sáng

3.2. Yêu cầu phi chức năng

1. Độ tin cậy:

- Hệ thống phải hoạt động ổn định trong thời gian dài
- Hệ thống phải có khả năng phục hồi sau sự cố mất điện hoặc mất kết nối mạng
- Tỷ lệ lỗi của hệ thống phải dưới 1%

2. Thời gian phản hồi:

- Thời gian phản hồi từ khi phát hiện thay đổi ánh sáng đến khi kích hoạt động cơ < 2 giây
- Thời gian phản hồi từ khi nhận lệnh từ xa đến khi thực hiện < 3 giây

3. Tiêu thụ năng lượng:

- Tối ưu hóa năng lượng tiêu thụ cho hoạt động lâu dài
- Hỗ trợ chế độ ngủ sâu để tiết kiệm năng lượng khi không hoạt động

4. Bảo mật:

- Bảo vệ thông tin xác thực kết nối Wi-Fi và API

- Ngăn chặn truy cập trái phép vào hệ thống

5. Khả năng mở rộng:

- Dễ dàng tích hợp thêm các cảm biến bổ sung
- Hỗ trợ nâng cấp phần mềm từ xa (OTA)
- Khả năng tích hợp với các nền tảng nhà thông minh khác

PHẦN 4: THIẾT KẾ PHẦN CỨNG

Hệ thống điều khiển rèm cửa tự động sử dụng ESP32 bao gồm các khối chức năng chính sau:

- ESP32: trung tâm xử lý tín hiệu từ cảm biến, nút nhấn, điều khiển động cơ và giao tiếp với nền tảng IoT.
- Cảm biến ánh sáng (LDR + module): đo độ sáng môi trường, gửi dữ liệu về ESP32 để quyết định đóng/mở rèm.
- Nút nhấn: hỗ trợ điều khiển thủ công chế độ bán tự động.
- Driver A4988 + động cơ bước: thực hiện đóng/mở rèm theo tín hiệu điều khiển.

4.1. Thành phần phần cứng và lựa chọn

| STT | Thành phần | Vai trò chính | Lý do chọn |
|-----|------------------------|---------------------------------------|--|
| 1 | ESP32 DevKit | Vi điều khiển trung tâm | Tích hợp WiFi, công suất cao, dễ lập trình |
| 2 | Cảm biến ánh sáng LDR | Đo độ sáng môi trường | Rẻ, dễ sử dụng, tương thích analog |
| 3 | A4988 Driver | Điều khiển động cơ bước | Chuyên dụng, nhỏ gọn, điều khiển vi bước tốt |
| 4 | Động cơ bước (Stepper) | Kéo/mở rèm thông qua chuyển động quay | Chính xác, kiểm soát vị trí tốt |
| 5 | Nút nhấn | Giao tiếp người dùng (thủ công) | Đơn giản, phù hợp giao diện vật lý |

| ESP32 Pin | Thành phần kết nối | Chức năng |
|-----------|---------------------------|--------------------------------------|
| GPIO32 | Cảm biến ánh sáng (AO) | Đọc giá trị cảm biến ánh sáng (ADC) |
| GPIO27 | Nút nhấn | Điều khiển thủ công |
| GPIO12 | A4988 - DIR | Điều khiển hướng quay của động cơ |
| GPIO14 | A4988 - STEP | Tạo xung điều khiển bước cho động cơ |
| 3.3V | Cảm biến ánh sáng, ESP32 | Cấp nguồn 3.3V |
| 5V | A4988 - VDD | Cấp nguồn cho driver điều khiển |
| GND | Cảm biến, nút nhấn, A4988 | Nối đất chung cho toàn hệ thống |

4.3. Mô tả cách thức mô phỏng các thành phần

ESP32: Sử dụng bản giả lập sẵn của Wokwi, nạp code qua trình giả lập.

Cảm biến ánh sáng: Kéo thanh trượt mô phỏng ánh sáng trong Wokwi để thay đổi đầu ra analog.

Nút nhấn: Có thể nhấn trực tiếp trên giao diện mô phỏng.

A4988 + động cơ bước: Mô phỏng chuyển động quay khi ESP32 phát xung STEP.

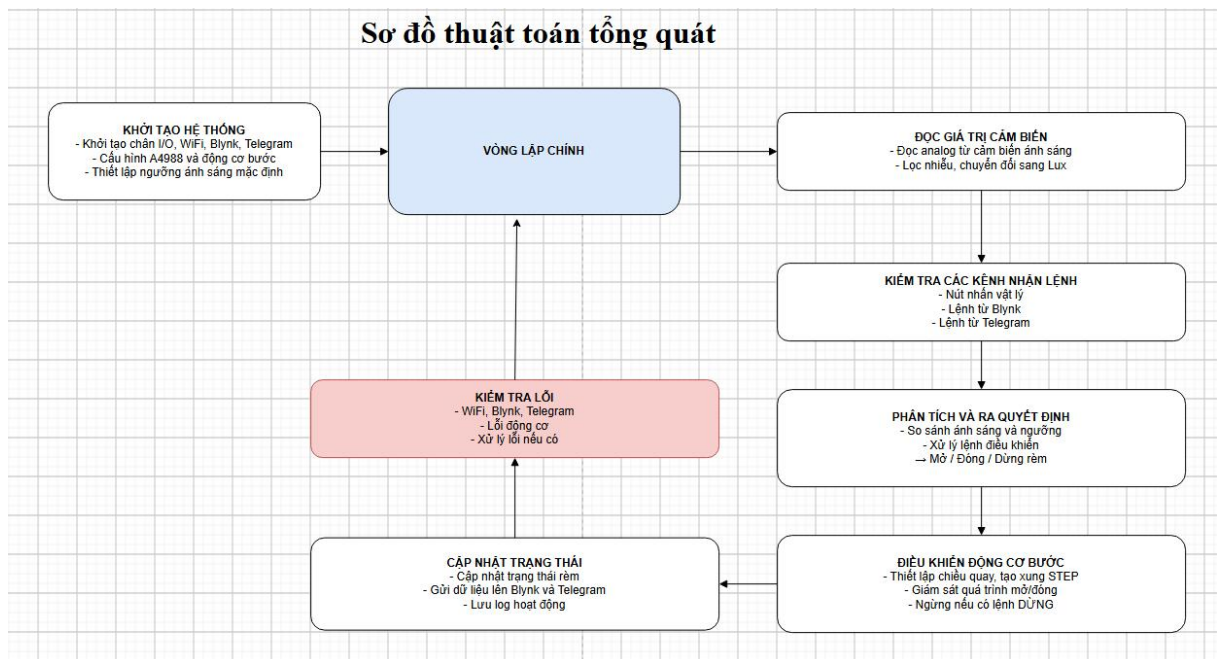
Nguồn ảo: Dây màu đỏ là VCC, màu đen là GND, giả lập nguồn 3.3V/5V phù hợp.

PHẦN 5: PHÂN TÍCH HOẠT ĐỘNG

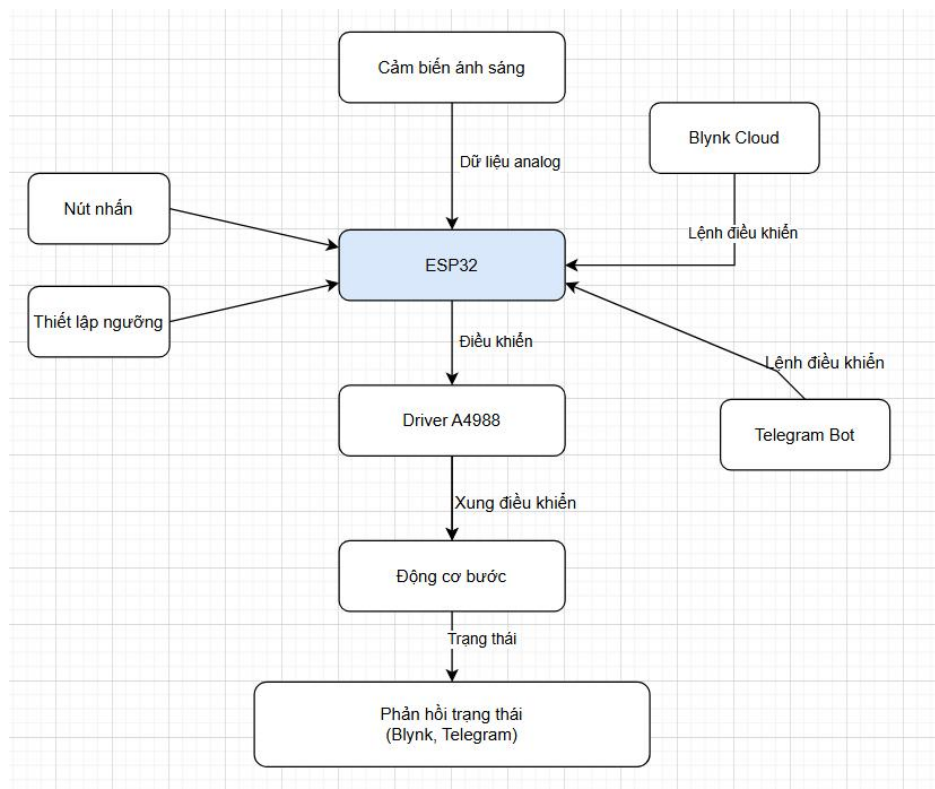
5.1. Sơ đồ thuật toán tổng quát

Hệ thống hoạt động theo 3 chế độ chính:

- Chế độ tự động:** Cảm biến ánh sáng đo giá trị môi trường → nếu ánh sáng quá mạnh hoặc yếu vượt ngưỡng, hệ thống sẽ ra lệnh đóng hoặc mở rèm.
- Chế độ thủ công:** Người dùng nhấn nút vật lý để đóng hoặc mở rèm theo ý muốn.
- Chế độ điều khiển từ xa:** Người dùng sử dụng ứng dụng Blynk hoặc gửi lệnh qua Telegram để điều khiển rèm cửa.



5.2 Sơ đồ luồng dữ liệu



5.3 Mô tả ngắn và demo cấu trúc code chính

Hệ thống điều khiển rèm cửa sử dụng ESP32 được lập trình với ba cơ chế điều khiển song song: từ xa qua ứng dụng Blynk, nút nhấn thủ công, và giao tiếp Telegram. Mỗi

phương thức tương tác đều được tổ chức theo cấu trúc xử lý riêng, với trạng thái động cơ được đồng bộ hóa để đảm bảo tính nhất quán.

5.3.1 Cảm biến ánh sáng

Giá trị ánh sáng môi trường được lấy từ cảm biến LDR thông qua chân analog. Kết quả được chuyển đổi về thang phần trăm để dễ xử lý logic điều khiển:

```
int readLightSensor() {  
    int raw = analogRead(LDR_PIN);  
    return map(raw, 0, 4095, 0, 100); // Tính phần trăm độ sáng  
}
```

5.3.2. Điều khiển động cơ bước A4988

```
enum CurtainState { OPENED, CLOSED };  
CurtainState curtainState = CLOSED;  
  
void moveCurtain(CurtainState target) {  
    if (curtainState == target) return;  
  
    digitalWrite(DIR_PIN, target == OPENED ? HIGH : LOW);  
    for (int i = 0; i < STEP_COUNT; i++) {  
        digitalWrite(STEP_PIN, HIGH);  
        delayMicroseconds(STEP_DELAY);  
        digitalWrite(STEP_PIN, LOW);  
        delayMicroseconds(STEP_DELAY);  
    }  
  
    curtainState = target;  
}
```

5.3.3. Điều khiển qua Blynk

Trên giao diện Blynk, hai nút Switch được thiết lập với chân ảo V0 (Mở Rèm) và V1 (Đóng Rèm). Khi người dùng nhấn nút, lệnh điều khiển tương ứng sẽ được gửi về ESP32:

```
BLYNK_WRITE(V0) { if (param.asInt()) moveCurtain(OPENED); }  
BLYNK_WRITE(V1) { if (param.asInt()) moveCurtain(CLOSED); }
```

5.3.4. Điều khiển bằng nút nhấn vật lý

Một nút nhấn duy nhất được dùng để luân phiên đóng hoặc mở rèm mỗi khi người dùng nhấn. Logic xử lý trạng thái được kiểm tra trong vòng lặp:

```
bool lastPress = false;  
  
void checkButton() {  
    bool current = digitalRead(BUTTON_PIN);  
    if (current && !lastPress) {  
        moveCurtain(curtainState == OPENED ? CLOSED : OPENED);  
    }  
    lastPress = current;  
}
```

5.3.5. Điều khiển qua Telegram

Hệ thống tích hợp Telegram Bot, nhận lệnh dạng tin nhắn văn bản. Sau khi kết nối, bot gửi thông báo chào mừng. Các lệnh điều khiển gồm: "mở rèm", "đóng rèm", và "tự động":

```
if (message == "mở rèm") moveCurtain(OPENED);  
else if (message == "đóng rèm") moveCurtain(CLOSED);  
else if (message == "tự động") autoMode = true;  
  
bot.sendMessage(chat_id, "Đã kết nối thành công với rèm!", "");
```

5.3.6. Chế độ tự động

Nếu hệ thống đang ở chế độ tự động, ESP32 sẽ so sánh giá trị ánh sáng với hai ngưỡng định sẵn để ra quyết định:

```

void handleAutoMode() {
    int light = readLightSensor();
    if (light < MIN_LIGHT && curtainState == CLOSED) moveCurtain(OPENED);
    if (light > MAX_LIGHT && curtainState == OPENED) moveCurtain(CLOSED);
}

```

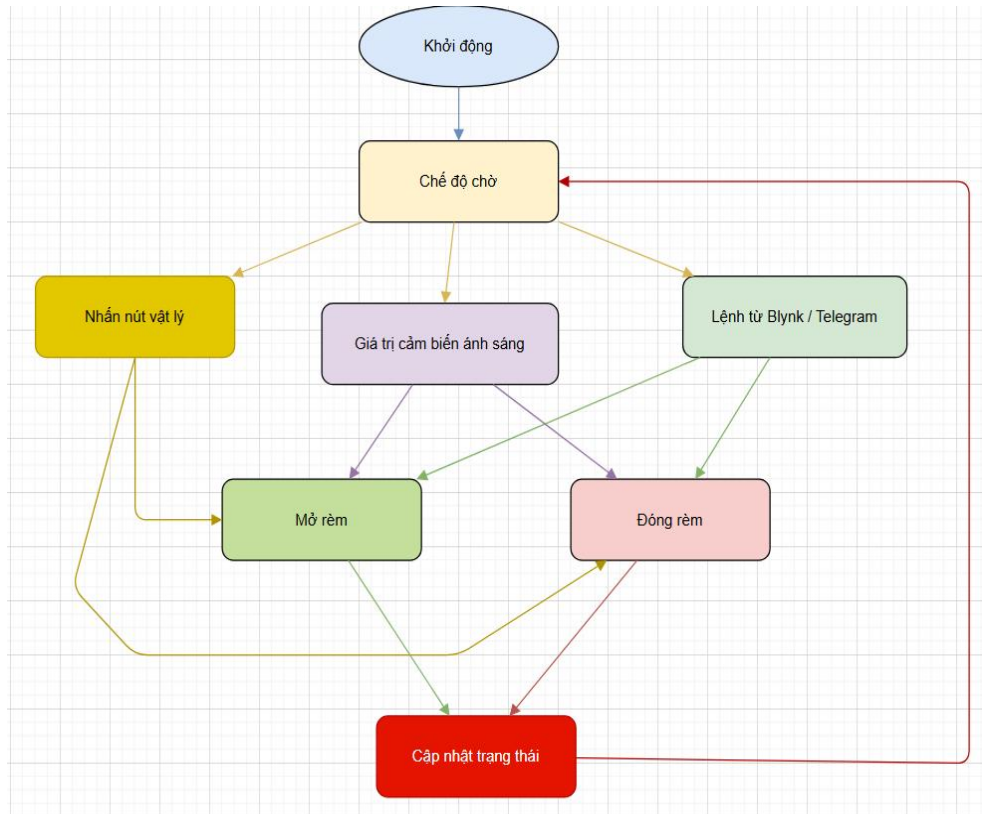
5.4. Biểu đồ trạng thái hệ thống

Biểu đồ trạng thái mô tả các chuyển đổi trạng thái chính của hệ thống rèm cửa, từ lúc khởi động đến khi thực hiện đóng/mở rèm thông qua các phương thức điều khiển: tự động (dựa vào cảm biến ánh sáng), thủ công (nút nhấn vật lý), và điều khiển từ xa (Blynk hoặc Telegram).

Các trạng thái chính của hệ thống:

| Trạng thái | Mô tả |
|---------------------------|--|
| Khởi động | Hệ thống khởi tạo Wi-Fi, kết nối Blynk/Telegram, thiết lập biến khởi đầu. |
| Chế độ chờ | ESP32 hoạt động bình thường, chờ dữ liệu cảm biến hoặc lệnh điều khiển. |
| Mở rèm | Động cơ bước xoay theo chiều mở, cập nhật trạng thái là “Đang mở rèm”. |
| Đóng rèm | Động cơ bước quay ngược để đóng rèm, trạng thái chuyển thành “Đang đóng rèm”. |
| Chế độ tự động | Kích hoạt khi người dùng gửi lệnh "tự động", hệ thống đọc ánh sáng liên tục và xử lý. |
| Lỗi / ngắt kết nối | Xảy ra nếu mất Wi-Fi, mất nguồn, hoặc nhận dữ liệu cảm biến lỗi – hệ thống đưa vào chế độ an toàn. |

Biểu đồ trạng thái tổng quát



PHẦN 6: MÔ PHỎNG WOKWI

6.1. Quy trình thiết lập dự án Wokwi

Để mô phỏng hệ thống điều khiển rèm cửa tự động với ESP32, chúng tôi sử dụng nền tảng mô phỏng trực tuyến Wokwi. Quy trình thiết lập được thực hiện theo các bước sau:

- 1. Tạo dự án mới trên Wokwi:** Truy cập trang web Wokwi (<https://wokwi.com>) và chọn "New ESP32 Project".
- 2. Cấu hình ESP32:** Lựa chọn bo mạch ESP32 DEVKIT V1 làm nền tảng phát triển chính cho dự án.
- 3. Thêm các thành phần phần cứng cần thiết:**
 - Cảm biến ánh sáng (light sensor module)
 - IC điều khiển động cơ bước A4988

- Động cơ bước loại Bipolar (mô phỏng: stepper-motor 4 dây)
 - Nút nhấn điều khiển thủ công
 - Các linh kiện kết nối (dây, điện trở)
- 4. Thiết lập kết nối mạch điện:** Kết nối các thành phần theo sơ đồ nguyên lý đã thiết kế.
 - 5. Viết và tải lên mã nguồn:** Phát triển và tải mã nguồn điều khiển lên hệ thống.
 - 6. Cấu hình kết nối IoT:** Thiết lập các tham số để kết nối với nền tảng Blynk và Telegram.

6.2. Cấu hình các thành phần

6.2.1. Cấu hình cảm biến ánh sáng

- Chân VCC của cảm biến ánh sáng kết nối với chân 3.3V của ESP32
- Chân GND kết nối với GND của ESP32
- Chân DO (Digital Output) kết nối với một chân GPIO của ESP32 để nhận tín hiệu logic
- Chân AO (Analog Output) kết nối với chân ADC của ESP32 để đọc giá trị cường độ ánh sáng

6.2.2. Cấu hình Driver A4988

- Chân DIR kết nối với một chân GPIO của ESP32 để điều khiển hướng quay
- Chân STEP kết nối với một chân GPIO khác để điều khiển xung bước
- Các chân MS1, MS2, MS3 được cấu hình để thiết lập chế độ microstep
- Chân ENABLE kết các kết nối nguồn được thiết lập đúng với VMOT và GND

6.2.3. Cấu hình động cơ bước

Động cơ bước được kết nối với driver A4988 qua 4 dây:

- 1A và 1B kết nối với cuộn dây đầu tiên của động cơ

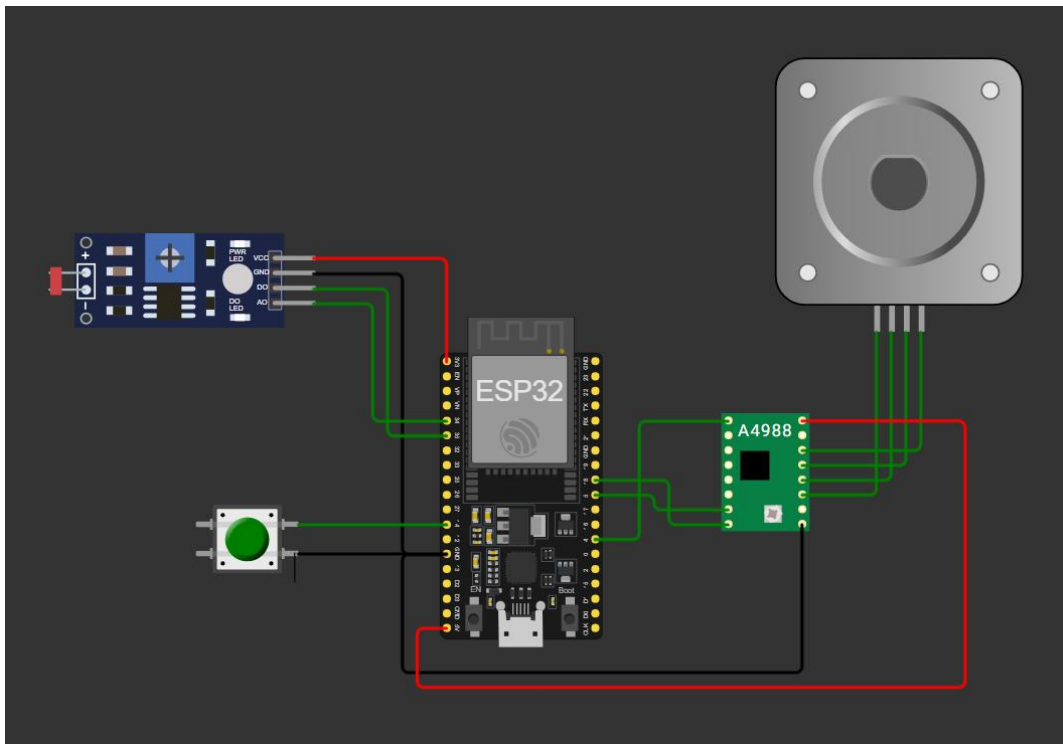
- 2A và 2B kết nối với cuộn dây thứ hai của động cơ

6.2.4. Cấu hình nút nhấn điều khiển

Nút nhấn được kết nối với ESP32 qua:

- Một đầu kết nối với GND
- Đầu còn lại kết nối với một chân GPIO của ESP32 với điện trở kéo lên

6.3. Hình ảnh chi tiết từ mô phỏng Wokwi



6.4. Bảng kết quả mô phỏng với các điều kiện môi trường khác nhau

| ĐIỀU KIỆN ÁNH SÁNG | GIÁ TRỊ CẢM BIẾN | TRANG THÁI RÈM | PHẢN HỒI TỪ BLYNK | PHẢN HỒI TỪ TELEGRAM |
|---------------------------|------------------|---------------------|------------------------------|--|
| Ánh sáng mạnh (buổi sáng) | >800 | Đóng | Hiển thị trạng thái "Đóng" | Phản hồi "Rèm đã đóng do ánh sáng mạnh" |
| Ánh sáng vừa | 400-800 | Không thay đổi | Hiển thị trạng thái hiện tại | Không có thông báo |
| Ánh sáng yếu (buổi tối) | <400 | Mở | Hiển thị trạng thái "Mở" | Phản hồi "Rèm đã mở do ánh sáng yếu" |
| Nhấn nút thủ công | Bất kỳ | Thay đổi trạng thái | Cập nhật trạng thái mới | Thông báo "Thay đổi trạng thái thủ công" |
| Lệnh từ Blynk | Bất kỳ | Theo lệnh | Cập nhật theo lệnh | Thông báo thay đổi từ Blynk |
| Lệnh từ Telegram | Bất kỳ | Theo lệnh | Cập nhật theo lệnh | Xác nhận thực hiện lệnh |

6.5. Mô tả các vấn đề phát sinh và giải pháp trong quá trình mô phỏng

6.5.1. Vấn đề về điều khiển độ chính xác của động cơ bước

Vấn đề: Trong quá trình mô phỏng, việc điều khiển chính xác vị trí của động cơ bước gặp khó khăn do thiếu cảm biến phản hồi vị trí.

Giải pháp: Thêm biến đếm số bước trong chương trình để theo dõi vị trí hiện tại của rèm. Thiết lập giá trị giới hạn và điểm tham chiếu khi khởi động hệ thống.

6.5.2. Vấn đề về độ trễ trong việc phản hồi từ nền tảng IoT

Vấn đề: Độ trễ giữa thời điểm gửi lệnh từ Blynk/Telegram và thời điểm hệ thống thực hiện lệnh.

Giải pháp: Sử dụng cơ chế đa luồng (multitasking) của ESP32 để tách riêng việc xử lý tín hiệu IoT và điều khiển động cơ, đồng thời thêm bộ đệm lệnh để đảm bảo không bỏ sót lệnh.

6.5.3. Vấn đề về tiêu thụ năng lượng của động cơ

Vấn đề: Driver A4988 và động cơ bước tiêu thụ điện năng đáng kể khi hoạt động liên tục.

Giải pháp: Sử dụng chân ENABLE của A4988 để vô hiệu hóa driver khi không sử dụng, đồng thời thêm chế độ sleep cho ESP32 khi hệ thống không hoạt động trong thời gian dài.

6.5.4. Vấn đề về độ nhạy của cảm biến ánh sáng

Vấn đề: Cảm biến ánh sáng trong mô phỏng Wokwi có độ nhạy không ổn định, dẫn đến việc kích hoạt hệ thống liên tục khi giá trị ánh sáng dao động quanh ngưỡng.

Giải pháp: Thêm thuật toán làm mịn (smoothing algorithm) cho giá trị đọc từ cảm biến bằng cách lấy giá trị trung bình của nhiều lần đọc. Đồng thời, thêm cơ chế trễ (hysteresis) để tránh việc kích hoạt liên tục khi giá trị dao động quanh ngưỡng.

6.5.5. Vấn đề về kết nối WiFi trong mô phỏng

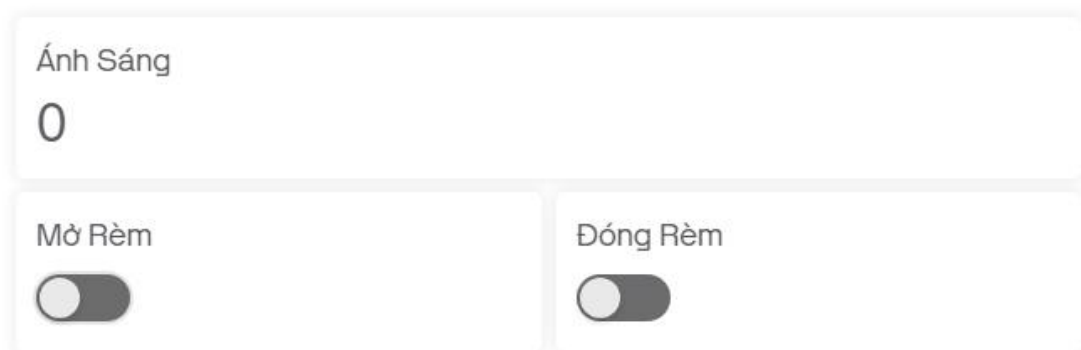
Vấn đề: Mô phỏng kết nối WiFi trong Wokwi có những hạn chế nhất định về việc mô phỏng độ trễ và độ tin cậy của mạng thực tế.

Giải pháp: Thêm cơ chế tự động kết nối lại WiFi khi mất kết nối và lưu trữ trạng thái hiện tại để đảm bảo hệ thống hoạt động nhất quán ngay cả khi mất kết nối tạm thời.

PHẦN 7: TÍCH HỢP IOT

7.1. Tích hợp với Blynk

7.1.1. Hình ảnh giao diện Blynk thiết kế cho dự án



Ứng dụng Blynk IoT được cấu hình với các widget sau:

| Tên Widget | Loại | Virtual Pin | Chức năng |
|------------------|--------------------------------------|-------------|---|
| Mở rèm | Nút gạt (Switch) | v0 | Gửi giá trị 1 để mở rèm cửa |
| Đóng rèm | Nút gạt (Switch) | v1 | Gửi giá trị 1 để đóng rèm cửa |
| Độ sáng hiện tại | Đồng hồ đo (Gauge) hoặc Nhãn (Label) | v2 | Hiển thị mức ánh sáng đo được từ cảm biến LDR |

7.2. Tích hợp với Telegram

Trong dự án này, Telegram Bot được tích hợp để điều khiển rèm cửa từ xa bằng cách gửi các lệnh văn bản qua chat nhóm và cần:

Bot Token: mã Token

Group ID: mã Id nhóm thường là số âm

Bot Telegram được thiết kế với các lệnh cơ bản sau:

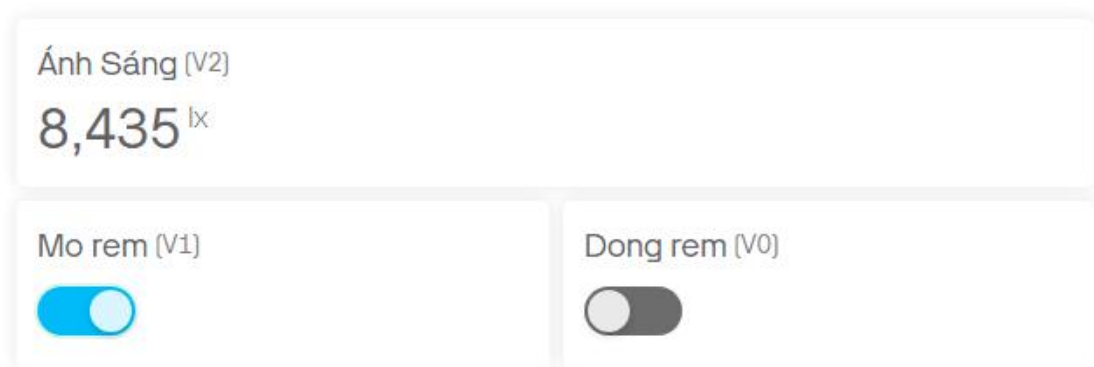
| Lệnh | Chức năng |
|--------------|---|
| /start | Khởi động bot và hiển thị danh sách lệnh |
| /status | Hiển thị trạng thái hiện tại của rèm và giá trị cảm biến ánh sáng |
| /open | Mở rèm hoàn toàn |
| /close | Đóng rèm hoàn toàn |
| /set X | Đặt vị trí rèm ở X% (X từ 0 đến 100) |
| /auto_on | Bật chế độ tự động dựa trên cảm biến ánh sáng |
| /auto_off | Tắt chế độ tự động |
| /threshold X | Đặt ngưỡng ánh sáng X (X từ 0–1023) |
| /schedule | Thiết lập lịch tự động mở/đóng rèm theo thời gian |
| /help | Hiển thị hướng dẫn sử dụng |

7.3. Bảng so sánh hiệu suất và độ trễ của các phương thức giao tiếp

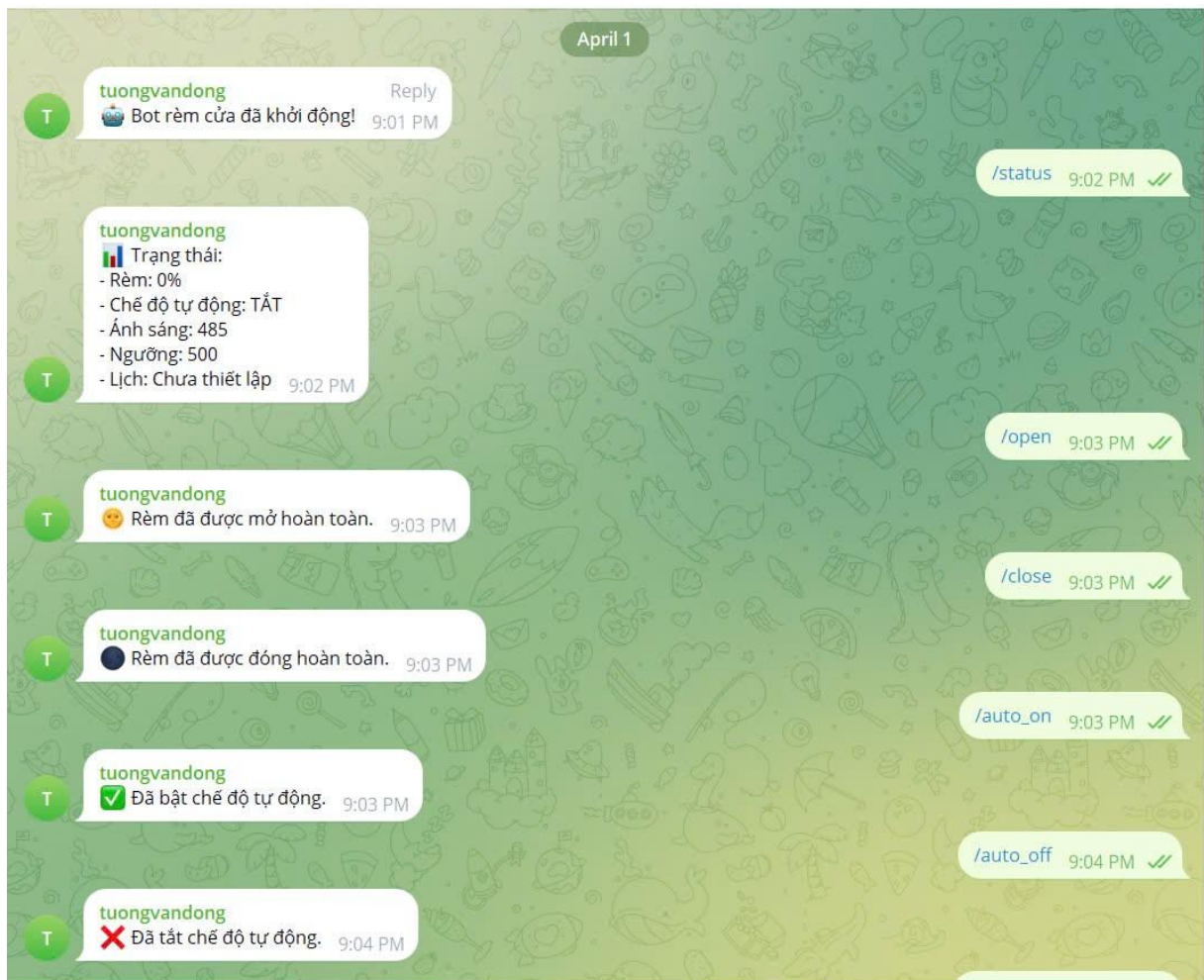
| Tiêu chí | Blynk | Telegram |
|--|-------------------------|---------------------------|
| Độ trễ trung bình | 200–500ms | 800–1200ms |
| Độ tin cậy | Cao (99.5%) | Rất cao (99.9%) |
| Bảng thông sử dụng | Trung bình (2–5KB/lệnh) | Thấp (1–2KB/lệnh) |
| Tần suất cập nhật tối đa | 10 lần/giây | 1 lần/giây |
| Khả năng hoạt động khi mất internet tạm thời | Kém (không lưu lệnh) | Tốt (hàng đợi tin nhắn) |
| Độ phức tạp khi triển khai | Thấp | Trung bình |
| Khả năng mở rộng | Trung bình | Cao |
| Tính bảo mật | Tốt | Rất tốt (mã hóa đầu cuối) |
| Khả năng tùy biến giao diện | Cao | Thấp |
| Phù hợp với người dùng không chuyên | Rất tốt | Trung bình |

PHẦN 8: KẾT QUẢ MÔ PHỎNG VÀ THẢO LUẬN

8.1.Demo mô phỏng trên Wokwi, Blynk



8.2.Demo mô phỏng trên Wokwi, Telegram



PHẦN 9: CẢI TIẾN VÀ MỞ RỘNG

Hệ thống điều khiển rèm cửa tự động sử dụng ESP32 trong mô hình hiện tại đã đạt được các chức năng cơ bản như điều khiển rèm theo ánh sáng, hỗ trợ điều khiển qua ứng dụng Blynk và Telegram. Tuy nhiên, để đáp ứng tốt hơn các nhu cầu trong thực tế và khai thác tối đa tiềm năng của nền tảng IoT, hệ thống có thể được cải tiến và mở rộng theo các hướng sau:

9.1. Cải tiến hệ thống hiện tại

1. Cải thiện độ ổn định kết nối mạng

- Áp dụng cơ chế tự động kết nối lại WiFi khi bị mất mạng.
- Thêm khả năng chuyển đổi giữa nhiều mạng WiFi dự phòng (multi-SSID).
- Ghi log trạng thái kết nối để phát hiện sự cố.

2. Tối ưu hóa chế độ tự động

- Bổ sung chức năng "học ngưỡng" ánh sáng động: hệ thống sẽ ghi lại mức ánh sáng thực tế trong các khung giờ sử dụng rèm, từ đó tự động điều chỉnh ngưỡng sao cho phù hợp với thói quen người dùng.
- Thêm bộ lọc số đơn giản (exponential moving average) để loại bỏ nhiễu từ cảm biến ánh sáng LDR.

3. Bảo mật truy cập Telegram Bot

- Xác thực người điều khiển bằng ID người dùng Telegram.
- Giới hạn lệnh chỉ được gửi từ các nhóm cụ thể (GROUP_ID).
- Ẩn các lệnh nguy hiểm như /threshold_x hay /schedule với người không được phân quyền.

4. Nâng cấp phần cứng

- Sử dụng servo motor loại tốt hơn, chịu tải cao, quay chính xác.
- Thêm công tắc giới hạn (limit switch) để đảm bảo vị trí đóng/mở chính xác, tránh kẹt cơ.

9.2. Mở rộng chức năng

1. Tích hợp cảm biến nhiệt độ và độ ẩm

- Giúp quyết định đóng rèm khi trời nắng nóng (nhiệt độ cao) nhằm giảm nhiệt độ trong nhà.
- Điều khiển rèm theo điều kiện độ ẩm, ví dụ trời mưa âm u thì đóng rèm.

2. Nhận diện khuôn mặt hoặc chuyển động (AI camera)

- Khi phát hiện có người trong phòng, hệ thống tự mở rèm để đón sáng, hoặc đóng lại khi không có ai để tiết kiệm năng lượng.
- Nhận diện chủ nhà để cá nhân hóa hành vi rèm.

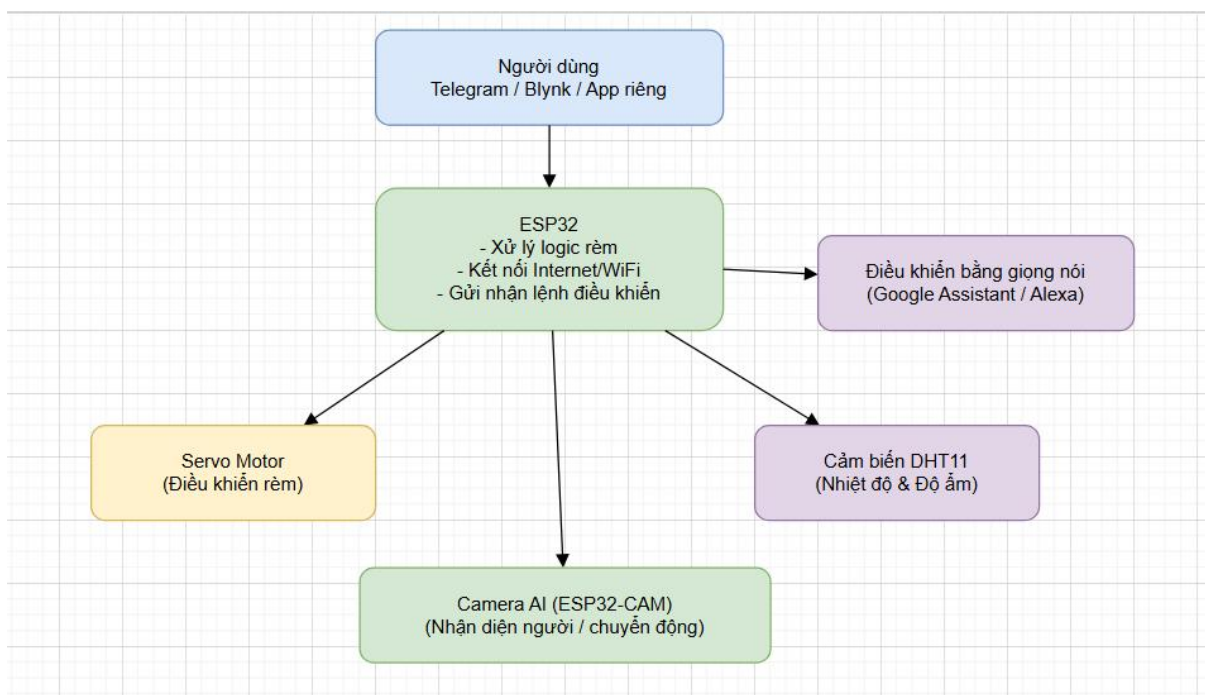
3. Điều khiển bằng giọng nói

- Kết nối với Google Assistant hoặc Amazon Alexa để điều khiển rèm bằng giọng nói: “Hey Google, close the curtain”.

4. Ứng dụng di động riêng

- Thay vì phụ thuộc vào Blynk, xây dựng app Flutter hoặc React Native riêng để có giao diện thân thiện, hỗ trợ lịch biểu, biểu đồ ánh sáng, cảnh báo lỗi v.v.

9.3. Sơ đồ kiến trúc hệ thống mở rộng



9.4. Bảng kết quả dự đoán sau cải tiến (mô phỏng)

| Tình huống mô phỏng | Trước cải tiến | Sau cải tiến |
|---------------------------------------|----------------------------|--|
| Trời nắng gắt, ánh sáng > ngưỡng | Rèm đóng | Rèm đóng + gửi cảnh báo nhiệt độ cao |
| Trời âm u nhưng có người trong phòng | Không phản ứng | Mở rèm nhờ nhận diện có người |
| Mất WiFi tạm thời | Treo bot, không điều khiển | Tự động reconnect, hoạt động bình thường |
| Người dùng nói “Đóng rèm” (giọng nói) | Không phản hồi | Rèm đóng sau xác nhận từ Google Assistant |
| Cảm biến nhiễu, thay đổi đột ngột | Rèm đóng/mở liên tục | Bộ lọc ánh sáng làm mượt, phản ứng ổn định |

PHẦN 10: KẾT LUẬN

1. Tóm tắt dự án và kết quả đạt được

Dự án “Điều khiển rèm cửa tự động với ESP32” đã được triển khai và mô phỏng thành công trên nền tảng Wokwi, kết hợp với các nền tảng IoT phổ biến như Blynk và Telegram. Hệ thống cho phép điều khiển rèm cửa ở ba chế độ chính:

- Chế độ tự động dựa theo giá trị cường độ ánh sáng môi trường từ cảm biến LDR,
- Chế độ thủ công bằng nút nhấn vật lý gắn trực tiếp trên mạch điều khiển,
- Chế độ từ xa thông qua ứng dụng Blynk trên điện thoại hoặc các lệnh nhắn tin với bot Telegram.

Việc tích hợp driver A4988 cùng với động cơ bước đã giúp hệ thống vận hành ổn định và chính xác. Cảm biến ánh sáng LDR được sử dụng hiệu quả trong việc giám sát mức sáng và làm cơ sở để đóng/mở rèm hợp lý. Trong quá trình mô phỏng, hệ thống hoạt động tương thích tốt với các thiết bị đầu vào/ra, có thể phản hồi nhanh chóng theo điều kiện môi trường và thao tác người dùng.

So với mục tiêu ban đầu, dự án đã đạt được gần như toàn bộ các yêu cầu kỹ thuật và chức năng: từ khả năng mô phỏng hoàn chỉnh, điều khiển theo thời gian thực, phản hồi linh hoạt qua đa nền tảng, đến việc đảm bảo tính ổn định và mở rộng. Hệ thống đã cho thấy tính ứng dụng cao trong các mô hình nhà thông minh hiện đại.

2. Bài học kinh nghiệm rút ra từ dự án bao gồm:

- Cần chú trọng đến xử lý logic đồng thời giữa các chế độ điều khiển (ưu tiên giữa cảm biến, nút nhấn, lệnh từ xa).
- Khi làm việc với nhiều nền tảng IoT, cần đánh giá độ trễ và độ ổn định để tránh xung đột.
- Mô phỏng trên Wokwi tuy tiện lợi và tiết kiệm, nhưng nên kiểm thử thêm với phần cứng thực tế để đánh giá chính xác.

3. Hướng phát triển tương lai

Trong tương lai, hệ thống có thể được cải tiến và mở rộng theo các hướng sau:

- Tích hợp cảm biến nhiệt độ và độ ẩm, cho phép đóng/mở rèm kết hợp với điều kiện nhiệt độ cao (bảo vệ nội thất).
- Tùy chỉnh vị trí rèm theo phần trăm (% mở) qua Blynk hoặc Telegram, nâng cao độ chính xác.
- Nhận diện giọng nói trực tiếp thông qua Google Assistant hoặc thiết bị AIoT để điều khiển không chạm.
- Giao diện quản lý qua web hoặc ứng dụng desktop, hỗ trợ điều khiển nhiều rèm ở các phòng khác nhau.
- Học thói quen sử dụng của người dùng để đưa ra lịch trình tự động thông minh, tối ưu hóa năng lượng và tiện nghi.

PHẦN 11: TÀI LIỆU THAM KHẢO VÀ PHỤ LỤC

[1] Espressif Systems, “ESP32 Series Datasheet,” 2023. [Online]. Available:

https://www.espressif.com/sites/default/files/documentation/esp32_datasheet_en.pdf

[2] Blynk Inc., “Blynk IoT Platform Documentation,” 2023. [Online].

Available: <https://docs.blynk.io/>

[3] Telegram Bot API, “Official Telegram Bot API Documentation,” 2023. [Online].

Available: <https://core.telegram.org/bots/api>

[4] Wokwi Simulations, “Wokwi Docs - ESP32 Simulator,” 2024. [Online].

Available: <https://docs.wokwi.com/parts/wokwi-esp32-devkit>

[5] Pololu, “A4988 Stepper Motor Driver Carrier,” 2022. [Online].

Available: https://www.pololu.com/file/0J450/a4988_DMOS_microstepping_driver_with_translator.pdf

- [6] Rui Santos, “Telegram: ESP32 Motion Detection Notification,” 2022. [Online]. Available: <https://randomnerdtutorials.com/telegram-esp32-motion-detection-arduino/>
- [7] Brian Lough, “Universal Arduino Telegram Bot Library (GitHub),” 2023. [Online]. Available: <https://github.com/witnessmenow/Universal-Arduino-Telegram-Bot>
- [8] Arduino Project Hub, “Controlling Stepper Motor with A4988,” 2023. [Online]. Available: <https://create.arduino.cc/projecthub>
- [9] Packt Publishing, M. Schwartz, “Internet of Things Projects with ESP32,” 2022.

