

TRƯỜNG ĐẠI HỌC KHOA HỌC
KHOA CÔNG NGHỆ THÔNG TIN



ĐỒ ÁN MÔN HỌC
PHÁT TRIỂN ỨNG DỤNG IOT
ĐIỀU KHIỂN THIẾT BỊ GIA DỤNG
QUA WIFI VỚI ESP32

Giáo viên hướng dẫn:

ThS. Võ Việt Dũng

Nhóm: 4

Lớp: Phát triển ứng dụng IoT

Năm học: 2024 - 2025

Huế, 4/2025

MỤC LỤC

MỤC LỤC	1
PHẦN MỞ ĐẦU.....	1
I. Giới Thiệu:.....	1
II. Tổng quan về thiết bị gia dụng:.....	1
CHƯƠNG 1 – KIẾN TRÚC VÀ THÀNH PHẦN HỆ THỐNG	3
1.1. Phân cứng:	3
1.2. Phần mềm:	3
CHƯƠNG 2 – MÔ PHỎNG VỚI WOKWI	4
2.1. Wokwi và tổng quan về Wokwi:	4
2.2. Thành phần của hệ thống mô phỏng:.....	4
2.2.1. ESP32:.....	4
2.2.2. Module Relay:.....	5
2.2.3. Cảm biến:	5
2.2.4. OLED:.....	6
2.2.5. LED trạng thái:	6
2.2.6. Nút nhấn:.....	6
2.3. Quy trình và mô phỏng trên Wokwi:	6
2.3.1. Quy trình mô phỏng:	6
2.3.2. Sơ đồ hoạt động:	7
2.3.2. Sơ đồ Wokwi:	8
CHƯƠNG 3 – NGUYÊN LÝ HOẠT ĐỘNG & TRIỂN KHAI HỆ THỐNG	9
3.1. Các tính năng tự động:.....	9
3.1.1. Quạt tự động bật khi nhiệt độ vượt ngưỡng.....	9
3.1.2. Máy phun sương tự động khi độ ẩm thấp	9
3.1.3. Đèn tự động bật khi cảm biến ánh sáng phát hiện trời tối	9
3.2. Chế độ thủ công:	9
3.3. Giao diện trên Blynk:.....	10
3.3.1. Quạt tự động bật khi nhiệt độ vượt ngưỡng.....	10
3.3.2. Máy phun sương tự động khi độ ẩm thấp	10
3.3.3. Đèn tự động bật khi cảm biến ánh sáng phát hiện trời tối	10
KẾT LUẬN.....	11
I. Kết quả đạt được:.....	11

II. Về hướng phát triển của đề tài:	11
DANH MỤC ẢNH	12
TÀI LIỆU THAM KHẢO	13

PHẦN MỞ ĐẦU

I. Giới thiệu:

Với sự phát triển chóng mặt của Internet of Thing trong bối cảnh hiện nay, đặc biệt là trong thời đại công nghệ 4.0, các công việc được thực hiện thủ công giờ đây dần dần thay thế bằng máy móc hiện đại, các kỹ thuật tự động hóa hay điều khiển từ xa và đã trở nên rộng rãi, thiết yếu trong đời sống hiện đại ngày nay.

Bắt kịp xu hướng hiện đại, đề tài nghiên cứu này sẽ tập trung vào mô hình điều khiển từ xa, tự động hóa với việc sử dụng bộ vi điều khiển ESP32 đã tích hợp sẵn Wi-Fi. Hướng tới mục tiêu thiết kế thành công một hệ thống có thể điều khiển được các thiết bị gia dụng (như quạt máy, đèn điện, bếp điện,...) qua nền tảng điều khiển Blynk – cung cấp giao diện thân thiện với người dùng, linh hoạt giúp người dùng dễ dàng theo dõi và điều khiển các thiết bị gia dụng từ xa thông qua bo mạch ESP32. Hệ thống sẽ cung cấp cho người dùng các chức năng liên quan đến bật/tắt các thiết bị đồng thời thu thập các dữ liệu từ các cảm biến. Từ những bước nghiên cứu ban đầu, đề tài không chỉ mang tính chất học thuật đơn thuần mà còn mở đường cho việc ứng dụng thực tiễn, mở rộng và phát triển các hệ thống khác tương tự trong tương lai.

II. Tổng quan về thiết bị gia dụng:

Thiết bị gia dụng, còn được gọi là thiết bị điện gia đình hoặc thiết bị nội trợ, là các máy móc hoặc thiết bị được sử dụng trong gia đình để hỗ trợ các chức năng như nấu ăn, làm sạch, bảo quản thực phẩm, và các công việc nội trợ khác [1]. Theo Britannica, các thiết bị này bao gồm các thiết bị điện, cơ điện hoặc chạy bằng gas, được giới thiệu chủ yếu trong thế kỷ 20 nhằm tiết kiệm lao động và thời gian trong gia đình. Chúng đã có tác động sâu sắc đến xã hội và kinh tế, đặc biệt trong các cộng đồng đô thị [2]. Với việc áp dụng mô hình điều khiển từ xa và cập nhật dữ liệu tự động, chúng có thể mang lại những lợi ích cho người dùng như sau:

Tiết kiệm năng lượng: Bằng cách tối ưu hóa thời gian hoạt động, điều khiển bật/tắt từ xa, cài đặt tự động tắt khi không cần sử dụng.

Tích hợp trên nhiều nền tảng: Các thiết bị có thể được điều khiển từ xa qua ứng dụng bởi người dùng trên nhiều nền tảng như trang web, điện thoại. Tạo sự linh hoạt, tiện lợi, thoải mái cho người dùng.

Hoạt động tối ưu: Các thiết bị có thể được điều khiển bởi các lịch trình có sẵn hay tự động xử lý các tình huống phát sinh bất ngờ.

Hệ thống IoT không chỉ là trình quản lý điều khiển các hoạt động đơn thuần như bật/tắt thiết bị mà còn bao gồm:

Xử lý các sự kiện cảnh báo: Khi có tình huống, sự cố hay thay đổi bất ngờ thì hệ thống sẽ gửi những thông báo đến Blynk về những thay đổi đó.

Giám sát trạng thái hoạt động: Cho phép đo lường các thông số môi trường như độ ẩm, nhiệt độ, ánh sáng,.. theo thời gian thực đến hệ thống để người dùng có thể kiểm soát và kịp thời xử lý.

Mở rộng và tích hợp: Hệ thống có thể mở rộng các chức năng mới trong tương lai, cho phép thêm các cảm biến mới.

CHƯƠNG 1 – KIẾN TRÚC VÀ THÀNH PHẦN HỆ THỐNG

1.1. Phần cứng:

ESP32: ESP32 là một bộ vi điều khiển mạnh mẽ với Wi-Fi tích hợp, đảm bảo kết nối mạng ổn định và khả năng xử lý cao, phù hợp cho các ứng dụng IoT. Nó có thể nhận lệnh điều khiển, thu thập dữ liệu từ các cảm biến và giao tiếp qua Wi-Fi để thực hiện các tác vụ tự động hóa. [3][4]

Cảm biến: Các cảm biến như DHT22 cho phép giám sát môi trường xung quanh, từ đó điều chỉnh hoạt động của các thiết bị như máy lạnh, quạt. Ngoài ra, cảm biến ánh sáng (LDR) có thể phát hiện sự thay đổi ánh sáng hoặc chuyển động, giúp điều khiển đèn chiếu sáng theo thời gian hợp lý. [3][4]

Module Relay: Mạch điện sẽ được chuyển đổi, hơn nữa còn cho phép bật/tắt các thiết bị gia dụng. Mạch ESP32 và mạch điện của thiết bị cần điều khiển sẽ được relay cách ly tín hiệu điện tử từ.

OLED (Organic Light Emitting Diode) – thiết bị công nghệ hiển thị tiên tiến, OLED là công nghệ hiển thị tiên tiến, được ứng dụng rộng rãi trong các thiết bị điện tử như TV, smartphone, đồng hồ thông minh, và các module hiển thị nhỏ trong các dự án IoT. Nó cung cấp khả năng hiển thị sắc nét và tiết kiệm năng lượng. [4]

1.2. Phần mềm:

Ứng dụng Web/Mobile: Đơn giản với các nút bấm cho phép bật/tắt thiết bị và hiển thị trạng thái của các cảm biến theo thời gian thực.

Blynk: ESP32 kết nối đến Blynk server để gửi và nhận dữ liệu. Đảm bảo các lệnh được truyền tải một cách nhanh chóng và ổn định.

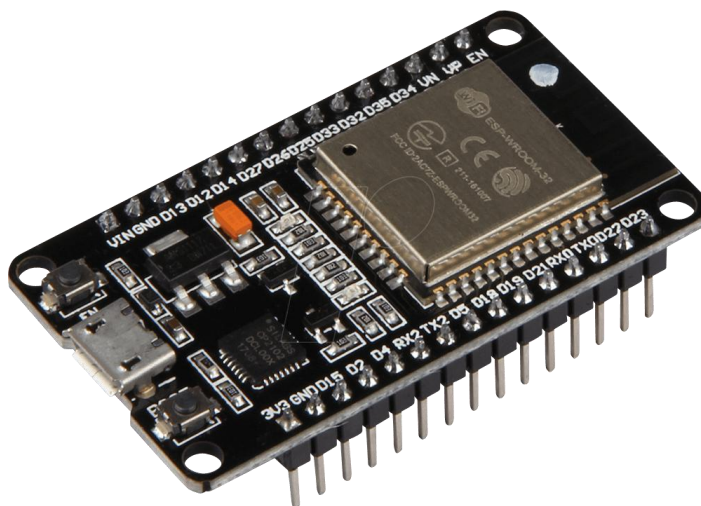
CHƯƠNG 2 – MÔ PHỎNG VỚI WOKWI

2.1. Wokwi và tổng quan về Wokwi:

Wokwi là một nền tảng mô phỏng trực tuyến được thiết kế để hỗ trợ các dự án liên quan đến vi điều khiển như ESP32, Arduino, và các module cảm biến khác. Nền tảng này cho phép người dùng xây dựng, kiểm thử mạch điện tử và lập trình trực tiếp trong môi trường mô phỏng mà không cần phần cứng thực tế. Wokwi hỗ trợ nhiều loại vi điều khiển và ngôn ngữ lập trình như Arduino IDE, MicroPython, và CircuitPython, giúp người dùng dễ dàng thử nghiệm và tối ưu hóa các giải pháp IoT. Trong dự án này, Wokwi được sử dụng để kiểm tra tính khả thi của việc điều khiển thiết bị gia dụng thông qua ESP32 trước khi triển khai thực tế [5]. Trong dự án này, Wokwi được dùng để kiểm tra tính khả thi của giải pháp điều khiển thiết bị gia dụng thông qua ESP32 trước khi triển khai thực tế.

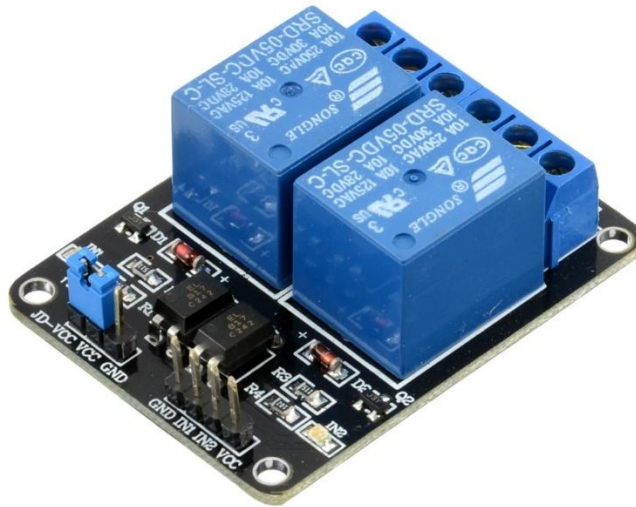
2.2. Thành phần của hệ thống mô phỏng:

2.2.1. ESP32:



Hình 2. 1. Hình minh họa bộ điều khiển ESP32

2.2.2. Module Relay:

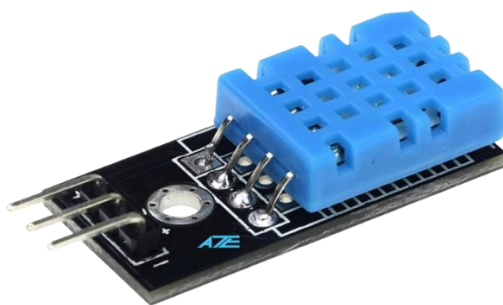


Hình 2. 2. Hình minh họa cho module Relay

2.2.3. Cảm biến:

Cảm biến DHT22: Thu thập thông tin về môi trường (như nhiệt độ, độ ẩm). Nếu nhiệt độ $> 30^{\circ}\text{C}$ \rightarrow ESP32 kích hoạt relay để bật quạt. Nếu độ ẩm $< 40\%$ \rightarrow ESP32 kích hoạt relay để bật máy phun sương.

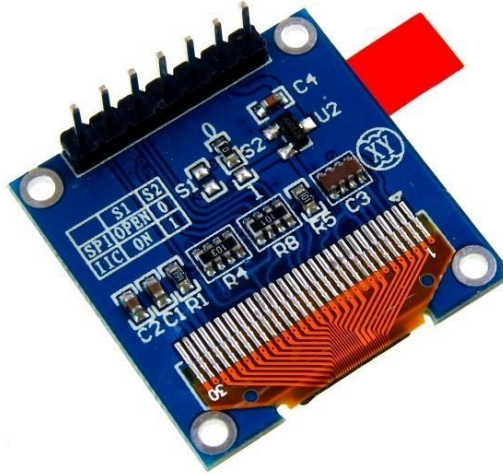
Cảm biến ánh sáng LDR: Theo dõi cường độ ánh sáng và điều chỉnh trạng thái thiết bị phù hợp. Nếu ánh sáng < 300 lux \rightarrow ESP32 kích hoạt relay để bật đèn.



Hình 2. 3. Hình minh họa cho module cảm biến nhiệt độ ẩm DHT22

2.2.4. OLED:

Loại màn hình SSD1306 thực hiện chức năng hiển thị thông tin cảm biến (nhiệt độ, độ ẩm, ánh sáng) và trạng thái của các thiết bị (quạt, máy phun sương, đèn).



Hình 2. 4. Hình minh họa cho module màn hình SSD1306

2.2.5. LED trạng thái:

Thực hiện chức năng hiển thị trạng thái bật/tắt của các thiết bị được điều khiển. Bao gồm: led_quat, led_phunsuong, led_den.

2.2.6. Nút nhấn:

Chế độ thủ công: Khi *btn_manual* được nhấn (tín hiệu HIGH), hệ thống chuyển sang chế độ thủ công. Các btn khác sẽ thực hiện chức năng của mình như *btn_quat* thực hiện chức năng bật/tắt quạt, *btn_phunsuong* thực hiện chức năng bật/tắt máy phun sương và *btn_den* thực hiện chức năng bật/tắt đèn.

Chế độ tự động: Khi *btn_manual* không nhấn (tín hiệu LOW), hệ thống hoạt động dựa vào dữ liệu từ các cảm biến (DHT22 và LDR).

2.3. Quy trình và mô phỏng trên Wokwi:

2.3.1. Quy trình mô phỏng:

Hệ thống mô phỏng bao gồm các bước hoạt động như sau:

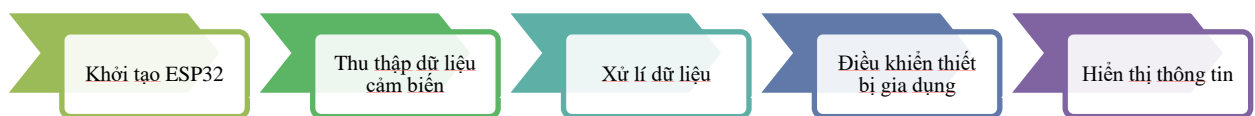
Khởi tạo ESP32: ESP32 được kích hoạt, kết nối với mạng Wi-Fi, và thiết lập giao tiếp với ứng dụng Blynk để gửi và nhận lệnh điều khiển.

Thu thập dữ liệu cảm biến: ESP32 tiếp nhận thông tin từ các cảm biến như nhiệt độ, độ ẩm, ánh sáng, và chuyển động.

Xử lý dữ liệu: Dữ liệu phân tích từ cảm biến của ESP32 sẽ xác định trạng thái và đưa ra quyết định điều khiển.

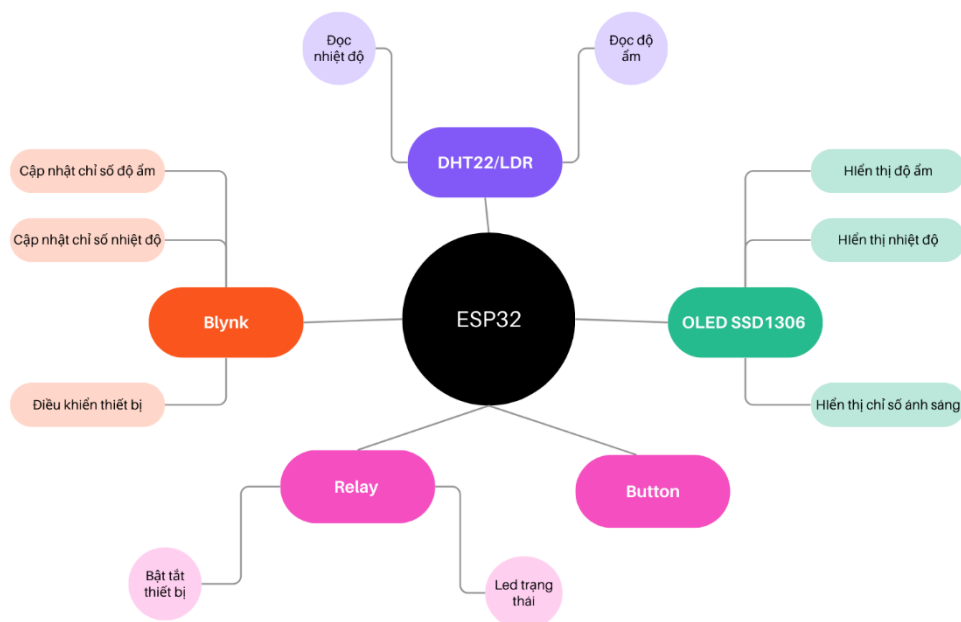
Điều khiển thiết bị gia dụng: Module relay nhận tín hiệu từ ESP32 để bật hoặc tắt các thiết bị như đèn hoặc quạt, dựa vào trạng thái cảm biến hoặc yêu cầu từ ứng dụng.

Hiển thị thông tin: Dữ liệu trạng thái thiết bị và thông tin cảm biến được gửi đến ứng dụng Blynk để hiển thị trên giao diện mobile, giúp người dùng giám sát và tùy chỉnh hệ thống.



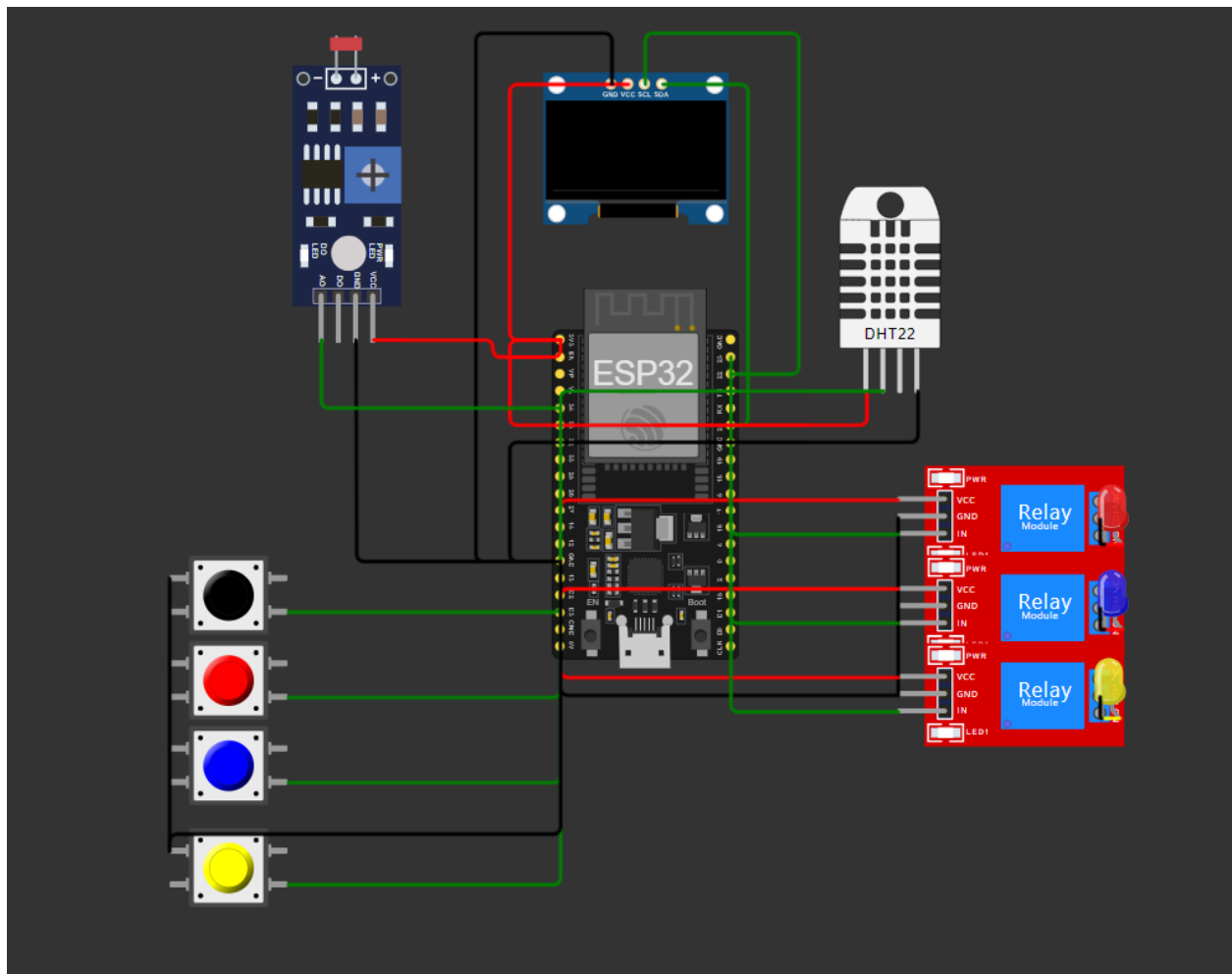
Hình 2. 4. Quy trình mô phỏng trên hệ thống

2.3.2. Sơ đồ hoạt động:



Hình 2. 5. Sơ đồ hoạt động của hệ thống

2.3.2. Sơ đồ Wokwi:



Hình 2. 5. Sơ đồ wokwi của hệ thống

CHƯƠNG 3 – NGUYÊN LÝ HOẠT ĐỘNG & TRIỂN KHAI HỆ THỐNG

3.1. Các tính năng tự động:

3.1.1. Quạt tự động bật khi nhiệt độ vượt ngưỡng

Nguyên lý hoạt động: Cảm biến nhiệt độ (DHT22/DHT11) đo nhiệt độ môi trường. Nếu vượt ngưỡng (ví dụ: 30°C), ESP32 kích hoạt relay để bật quạt.

Cách triển khai: Đầu tiên ta cần kết nối phần cứng: Cảm biến nhiệt độ → ESP32 → Relay → Quạt. Sau đó cung cấp cho bảng mạch đoạn lập trình thực hiện các chức năng như đọc dữ liệu cảm biến, kiểm tra nhiệt độ, và điều khiển relay.

Kết quả: Trình bày trạng thái quạt (ON/OFF) trên ứng dụng Blynk với ảnh minh họa giao diện hiển thị nhiệt độ.

3.1.2. Máy phun sương tự động khi độ ẩm thấp

Nguyên lý hoạt động: Khi độ ẩm môi trường giảm xuống dưới ngưỡng (ví dụ: 40%), ESP32 kích hoạt relay để bật máy phun sương.

Cách triển khai: Kết nối phần cứng có cảm biến độ ẩm → ESP32 → Relay → Máy phun sương. Lập trình yêu cầu đọc dữ liệu cảm biến, so sánh với ngưỡng độ ẩm, và bật relay nếu cần.

Kết quả: Hiển thị giao diện Blynk với giá trị ánh sáng và trạng thái đèn.

3.1.3. Đèn tự động bật khi cảm biến ánh sáng phát hiện trời tối

Nguyên lý hoạt động: Cảm biến ánh sáng (LDR hoặc BH1750) đo độ sáng (*lux*). Nếu ánh sáng yếu (dưới 300 lux), ESP32 sẽ bật relay để sáng đèn.

Cách triển khai: Phần cứng được kết nối như sau cảm biến ánh sáng → ESP32 → Relay → Đèn. Đối với phần lập trình, đo giá trị lux từ cảm biến, so sánh ngưỡng để điều khiển đèn.

Kết quả: Hiển thị giao diện Blynk với giá trị ánh sáng và trạng thái đèn.

3.2. Chế độ thủ công:

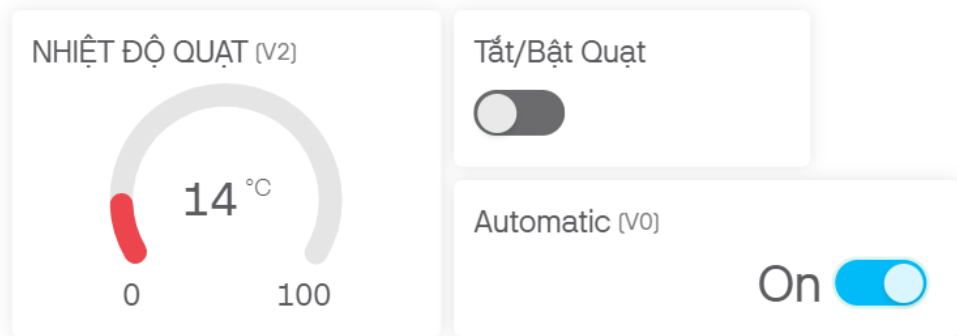
Nguyên lý hoạt động: Người dùng có thể điều khiển trực tiếp quạt, máy phun sương, và đèn thông qua nút bấm trên ứng dụng Blynk.

Cách triển khai: Thêm nút bấm (Button) trên Blynk, thiết lập điều khiển relay qua tín hiệu từ ESP32. Lập trình để xử lý tín hiệu từ Blynk. Nếu bật chế độ thủ công, hệ thống sẽ tạm dừng tự động hóa.

Kết quả: Minh họa giao diện với nút bấm bật/tắt và trạng thái thiết bị trên Blynk.

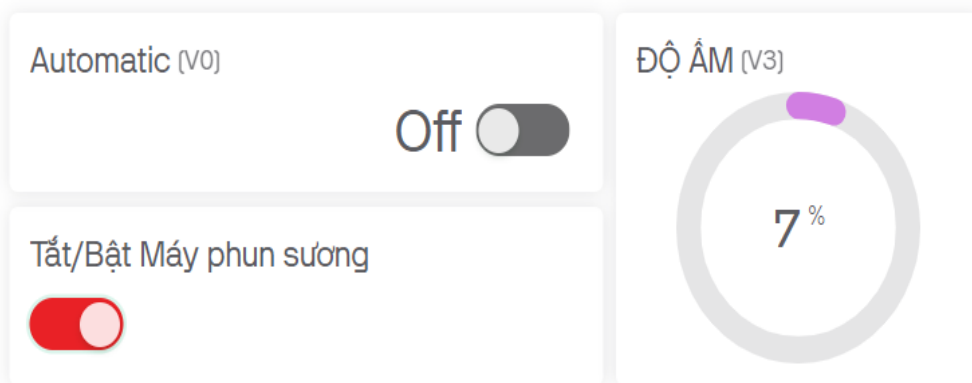
3.3. Giao diện trên Blynk:

3.3.1. Quạt tự động bật khi nhiệt độ vượt ngưỡng



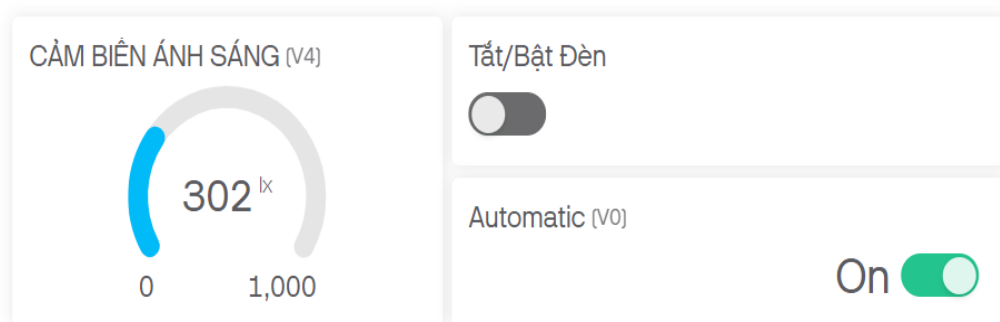
Hình 3. 1. Giao diện Blynk của hệ thống quạt tự động

3.3.2. Máy phun sương tự động khi độ ẩm thấp



Hình 3. 2. Giao diện Blynk của hệ thống máy phun sương tự động

3.3.3. Đèn tự động bật khi cảm biến ánh sáng phát hiện trời tối



Hình 3. 3. Giao diện Blynk của hệ thống đèn tự động

KẾT LUẬN

I. Kết quả đạt được:

Đề tài "Phát triển ứng dụng IoT điều khiển thiết bị gia dụng qua wifi bằng ESP32" không chỉ đáp ứng yêu cầu tự động hóa mà còn mở rộng khả năng điều khiển linh hoạt, thích nghi với nhiều điều kiện môi trường khác nhau, tạo ra một hệ thống thông minh, linh hoạt, và cung cấp tiện nghi cho người sử dụng. Tính ứng dụng của hệ thống sẽ còn được mở rộng trong tương lai nhờ tính triển vọng của dự án.

II. Về hướng phát triển của đề tài:

1. Phát triển ứng dụng nền tảng di động: Với Blynk, hệ thống sẽ triển khai nền tảng di động để thuận tiện cho việc quản lý mọi lúc mọi nơi.

2. Áp dụng trí tuệ nhân tạo vào hệ thống: Cùng với tốc độ phát triển chóng mặt của công nghệ AI, ta có thể vận dụng AI vào việc phân tích dữ liệu, dự đoán các trường hợp phát sinh, và tối ưu hóa thời gian xử lý.

3. Tối ưu hóa hiệu quả năng lượng: Bổ sung thêm mô-đun liên quan giúp tiết kiệm năng lượng để hướng tới một dự án năng lượng tối ưu và tiết kiệm.

Tóm lại, dự án không chỉ hoàn thiện về mặt kỹ thuật mà còn đặt nền móng cho sự phát triển sâu rộng trong tương lai. Tiềm năng của dự án như tích hợp ứng dụng di động, kết nối đám mây, đặc biệt tích hợp AI để tối ưu hóa vận hành hứa hẹn sẽ nâng tầm hệ thống, góp phần vào xu hướng cách mạng công nghiệp 4.0 và thúc đẩy cuộc sống thông minh, bền vững hơn.

DANH MỤC ẢNH

Hình 2. 1. Hình minh họa bộ điều khiển ESP32.....	4
Hình 2. 2. Hình minh họa cho Module Relay	5
Hình 2. 3. Hình minh họa cho Module cảm biến nhiệt độ ẩm DHT22	5
Hình 2. 4. Hình minh họa cho Module màn hình SSD1306.....	6
Hình 2. 4. Quy trình mô phỏng trên hệ thống.....	7
Hình 2. 5. Sơ đồ hoạt động của hệ thống.....	7
Hình 2. 5. Sơ đồ wokwi của hệ thống.....	8
Hình 3. 1. Giao diện Blynk của hệ thống quạt tự động	10
Hình 3. 2. Giao diện Blynk của hệ thống máy phun sương tự động	10
Hình 3. 3. Giao diện Blynk của hệ thống đèn tự động	10

TÀI LIỆU THAM KHẢO

- [1]Wikipedia Contributors. (2019, July 2). *Home appliance*. Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Home_appliance
- [2]Gregersen, E. (n.d.). *Home appliance - Appliances for cleaning*. Encyclopedia Britannica. <https://www.britannica.com/technology/home-appliance/Appliances-for-cleaning>
- [3]*ESP32 with DHT11/DHT22 Temperature and Humidity Sensor using Arduino IDE / Random Nerd Tutorials*. (2019, April 25). Random Nerd Tutorials. <https://randomnerdtutorials.com/esp32-dht11-dht22-temperature-humidity-sensor-arduino-ide/>
- [4]Team, E. E. (2025, February 25). *IoT-Based Smart Temperature & Humidity Monitoring System Using ESP32*. Educational Engineering Team. <https://eduengteam.com/iot-based-smart-temperature-humidity-monitoring-system-using-esp32/>
- [5]*ESP32 Simulation / Wokwi Docs*. (n.d.). Docs.wokwi.com. <https://docs.wokwi.com/guides/esp32>