

DẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ
BỘ MÔN VIỄN THÔNG



BÀI TẬP LỚN IOT
ĐỀ TÀI: ỨNG DỤNG IOT TRONG VIỆC
DIỀU KHIỂN CÁC THIẾT BỊ TRONG NHÀ THÔNG QUA MQTT

GVHD: ThS. Nguyễn Khánh Lợi
Sinh viên thực hiện:
Trương Huy Diệu - 1912857
Nguyễn Đức Hậu - 1913304
Võ Khang Khang - 1913714
Nguyễn Tất Thành - 2012053
Trần Minh Ý - 1916069

TP. HỒ CHÍ MINH, THÁNG 12 - 2023



LỜI CẢM ƠN

Để thực hiện và hoàn thành đồ án này, nhóm đã nhận được sự hỗ trợ và giúp đỡ từ nhiều người. Nhóm xin bày tỏ tình cảm chân thành và lòng biết ơn sâu sắc, cho phép nhóm gửi lời cảm ơn đến thầy ThS. Nguyễn Khánh Lợi đã giúp đỡ nhóm trong quá trình nghiên cứu và hoàn thành đồ án.

Tuy có nhiều cố gắng, nhưng trong báo cáo đồ án này không tránh khỏi những thiếu sót, hạn chế. Nhóm mong Quý thầy cô, các chuyên gia, người quan tâm đến đề tài đóng góp ý kiến để đề tài hoàn thiện hơn.

Một lần nữa xin chân thành cảm ơn!

Tp.Hồ Chí Minh, ngày 07 tháng 12 năm 2023



TÓM TẮT ĐỒ ÁN

Đồ án này trình bày về ứng dụng internet of thing vào việc điều khiển và giám sát các thiết bị trong nhà như: đèn (led RGB), quạt thông qua giao thức MQTT, đồng thời ứng dụng đưa ra các thông báo dựa vào điều kiện, dữ liệu từ các cảm biến. Với ưu điểm sử dụng băng thông thấp, hoạt động trong môi trường có độ trễ cao và tiết kiệm năng lượng, MQTT phù hợp với các thiết bị IOT có đường truyền hạn chế và độ tin cậy thấp. MQTT dựa trên mô hình publish/subscribe (cung cấp/ thuê bao) và được ứng dụng nhiều cho các ứng dụng M2M.



Mục lục

1 GIỚI THIỆU	7
1.1 Tổng quan	7
1.2 Nhiệm vụ đề tài	7
1.3 Cấu trúc đề tài	7
2 CƠ SỞ LÝ THUYẾT	8
2.1 Giao thức MQTT	8
2.1.1 Khái niệm	8
2.1.2 Thành phần của MQTT	8
2.1.3 Cách thức hoạt động	8
2.1.4 Một số khái niệm trong MQTT	9
2.2 Linh kiện và module	10
2.2.1 Mcu ESP32	10
2.2.2 Cảm biến nhiệt độ và độ ẩm DHT11/22	12
2.2.3 Cảm biến khí ga MQ2	13
2.2.4 Động cơ DC	14
2.3 NODE RED	14
3 THIẾT KẾ VÀ THỰC HIỆN ỨNG DỤNG	16
3.1 Giao đoạn thiết kế	16
3.2 Flow chart	17
3.3 Thực hiện phần cứng, giao diện người dùng	19
3.3.1 Cài đặt node-RED	19
3.3.2 Thực hiện ứng dụng	19
4 KẾT LUẬN	23
4.1 Kết quả đạt được	23
4.2 Hướng phát triển	23
5 TÀI LIỆU THAM KHẢO:	24
6 PHỤ LỤC	25
6.1 source code (file Node-RED flows.json)	25
6.2 Fix lỗi không kết nối broker trên local, cài đặt node-red admin	25



Danh sách hình vẽ

1	Thành phần MQTT	8
2	ESP32 thegioic	10
3	Kiến trúc ESP32	11
4	Cảm biến nhiệt độ, độ ẩm	12
5	Cảm biến khí ga MQ2	13
6	Dòng cơ DC 3-6V	14
7	Khối cơ bản trong Node-RED	15
8	Ý tưởng thiết kế dự án	16
9	Flow chart điều khiển bật tắt quạt tự động	17
10	Flow chart cảnh báo khí gas cao	18
11	Flow chart bật/tắt quạt	18
12	Flow chart điều khiển màu led RGB	19
13	Giao diện dashboard của ứng dụng	20
14	Mạch thực tế với ESP32	21
15	Flow điều khiển màu đèn led RGB	21
16	Flow đọc và mô hình hóa thông số độ ẩm	21
17	Flow đọc và mô hình hóa thông số gas	22
18	Flow đọc và mô hình hóa thông số Nhiệt độ	22



Danh mục từ viết tắt

Kí hiệu viết tắt	Chữ viết đầy đủ	Ý nghĩa
MQTT	Message Queuing Telemetry Transport	Công nghệ truyền nhận tốc độ tức thời
IoT	Internet of Thing	Internet của sự vật
M2M	Machine to Machine	Máy đến Máy
IP	Internet Protocol	Giao thức Internet
LAN	Local Area Network	Mạng máy tính cục bộ



Phân chia công việc

STT	Sinh viên	MSSV	Công việc	Đánh giá
1	Trương Huy Diệu	1912857	Node-red flow, làm báo cáo	20%
2	Nguyễn Đức Hậu	1913304	Node-red flow, làm báo cáo	20%
3	Võ Khang Khang	1913714	Node-red flow, làm báo cáo	20%
4	Nguyễn Tất Thành	2012053	Viết mã nguồn ESP, test MQTT	20%
5	Trần Minh Ý	1916069	Viết mã nguồn ESP, test MQTT	20%



1 GIỚI THIỆU

1.1 Tổng quan

Với tiêu chí dự án IoT hướng đến tiện nghi, sử dụng hiệu quả nguồn tài nguyên, tính thực tế cao, nhóm hướng đến những dự án quy mô vừa và nhỏ giải quyết được vấn đề liên quan tới sự tiện lợi, đồng thời có thể điều khiển, giám sát từ xa. Và khi tìm hiểu, nhóm nhận thấy đề tài về ngôi nhà thông minh mang tính thiết thực cao, giúp người sử dụng (chủ nhà) có thể điều khiển, giám sát các thiết bị điện, ngôi nhà dễ dàng, thân thiện và sử dụng hiệu quả năng lượng hơn.

1.2 Nhiệm vụ đề tài

- Nội dung 1: tìm hiểu giao thức MQTT, các thức giao tiếp và các chế độ hoạt động.
- Nội dung 2: tìm hiểu module ESP32.
- Nội dung 3: tìm hiểu và thiết kế ứng dụng trên Node-RED

1.3 Cấu trúc đề tài

Nội dung đồ án được trình bày bao gồm ba phần chính (không kể phần mở đầu, mục lục, kết luận, tài liệu tham khảo và phụ lục) như sau:

- Chương 1: giới thiệu tổng quan về đề tài, quá trình lên ý tưởng.
- Chương 2: trình bày cơ sở lý thuyết về IoT, giao thức MQTT, kiến trúc ESP32 và một số cảm biến.
- Chương 3: trình bày thiết kế và thực hiện ứng dụng.

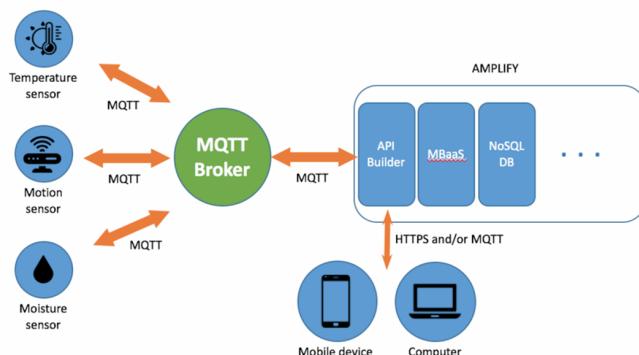
2 CƠ SỞ LÝ THUYẾT

2.1 Giao thức MQTT

2.1.1 Khái niệm

MQTT là từ viết tắt của cụm Message Queuing Telemetry Transport (tạm dịch: giao thức truyền thông diệp). Đây là một trong những giải pháp tiêu chuẩn của IoT (Internet of Things) vì quá trình truyền tải của MQTT rất nhẹ, độ chính xác cao và khả năng kết nối băng thông hiệu quả. MQTT còn được hiểu là một giao thức nhắn tin thông minh và đơn giản, được tạo ra nhằm phục vụ cho các thiết bị hạn chế về băng thông.

2.1.2 Thành phần của MQTT



Hình 1: Thành phần MQTT

- **MQTT broker (máy chủ mô giới):** Là thành phần được tạo ra dưới dạng mã nguồn mở. Một số MQTT broker có hỗ trợ dịch vụ điện toán đám mây. Chức năng của thành phần này là sàng lọc tin nhắn theo kênh. Kế đó, chúng đưa các tin nhắn này đến thiết bị hoặc ứng dụng đã đăng ký kênh ấy.
- **MQTT client (Publisher/Subscriber):** Thành phần này được nối kết với broker để truyền và nhận dữ liệu. Trong đó, client gửi dữ liệu được gọi là publisher, client đăng ký nhận dữ liệu là subscriber
- **Topic:** Là chủ đề hay kênh được quản lý bởi broker và được trao đổi bởi các client với nhau

2.1.3 Cách thức hoạt động

- **Broker:** Được xem là tâm điểm giữa của mọi kết nối đến tất cả client. Broker đảm nhiệm vai trò nhận tin nhắn, thông điệp được truyền tải từ publisher. Sau đó, thành phần này sẽ sắp xếp lại chúng và chuyển đến các subscriber nhất định.
- **Client:** Công việc của client là xuất bản hoặc đăng ký nhiều kênh khác nhau
- **Publisher:** Có chức năng truyền tải thông điệp, tin nhắn đến bất kỳ kênh nào



- **Subscriber:** Có chức năng nhận thông điệp, tin nhắn từ những kênh đã đăng ký. Tuy nhiên, thành phần này chỉ có thể nhận dữ liệu từ publisher khi đã đăng ký chính xác kênh tương ứng.

MQTT hoạt động theo cơ chế client/server, nơi mà mỗi cảm biến là một khách hàng (client) và kết nối đến một máy chủ, có thể hiểu như một Máy chủ môi giới (broker), thông qua giao thức TCP (Transmission Control Protocol). Broker chịu trách nhiệm điều phối tất cả các thông điệp giữa phía gửi đến đúng phía nhận.

MQTT là giao thức định hướng bản tin. Mỗi bản tin là một đoạn rời rạc của tín hiệu và broker không thể nhìn thấy. Mỗi bản tin được publish một địa chỉ, có thể hiểu như một kênh (Topic). Client đăng ký vào một vài kênh để nhận/gửi dữ liệu, gọi là subscribe. Client có thể subscribe vào nhiều kênh. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký. Khi một client gửi một bản tin đến một kênh nào đó gọi là publish.

2.1.4 Một số khái niệm trong MQTT

MQTT – Publish/Subscribe (Xuất bản/Đăng ký)

Khi giao thức MQTT hoạt động, những MQTT client (hay còn được gọi là node trạm) sẽ kết nối với một broker (MQTT server). Các node trạm sẽ đăng ký với một vài kênh (Topic). Chẳng hạn như: “/client1/channel1”, “/client2/channel2”. Hoạt động này được gọi là Subscribe (Đăng ký). Bạn có thể hình dung quá trình này tương tự như việc bạn đăng ký một kênh trên Youtube.

Khi ấy, những dữ liệu và kênh đã Subscribe sẽ được gửi đến mỗi client thông qua các note trạm. Quá trình client gửi dữ liệu đến một kênh gọi là Publish (Xuất bản).

MQTT – Message (Tin nhắn)

Tin nhắn trong giao thức MQTT được định dạng theo kiểu plain-text. Tuy nhiên, người dùng có thể tùy chỉnh tin nhắn thành các định dạng khác. Message là những thông tin trao đổi giữa các thiết bị của bạn. Chúng có thể là lệnh hoặc dữ liệu.

MQTT – Topic (Kênh)

Chủ đề là một trong những khái niệm quan trọng trong giao thức MQTT. Cũng được xem là “sợi dây nối kết” giữa Publish (xuất bản) và Subscribe (đăng ký). Nếu một tin nhắn được xuất bản trong một kênh, những người đăng ký kênh ấy sẽ nhận được tin nhắn này.

Những kênh này trình bày theo dạng chuỗi và phân tách với nhau bởi dấu gạch chéo (/). Trong đó, mỗi dấu gạch chéo biểu thị một cấp độ của kênh. Diễn hình như việc bạn tạo kênh cho tivi trong văn phòng tại nhà của mình: home/office/tivi.

Những kênh này có sự phân biệt giữa chữ hoa và chữ thường. Chẳng hạn như: home/office/tivi sẽ khác với home/office/Tivi. Bên cạnh đó, thông qua giao thức MQTT, bạn có thể khai báo các kênh cấp bách.

Ví dụ bạn sử dụng hệ thống cảm biến đo thông tin môi trường trong nhà. Tùy theo số lượng căn phòng trong ngôi nhà của bạn sẽ có bộ cảm biến môi trường riêng. Vậy, kênh truyền tải thông tin môi trường trong nhà bạn sẽ được khai báo như sau:

home/bedroom/temperature: kênh thông tin về độ ẩm trong phòng ngủ home/living-room/humidity: kênh thông tin về độ ẩm trong phòng khách

MQTT – QoS

QoS là từ viết của cụm Qualities of Service (tạm dịch: chất lượng dịch vụ). Người dùng có 3 lựa chọn khi Publish và Subscribe:

- QoS0: Broker và client sẽ gửi dữ liệu duy nhất 1 lần dựa trên giao thức TCP/IP.
- QoS1: Broker và client được phép gửi dữ liệu tối thiểu 1 lần và nhận xác nhận từ đầu kia

- QoS2: Broker và client gửi dữ liệu và đầu bén kia chỉ nhận đúng 1 lần. Hoạt động này thông qua 4 bước bắt tay.

MQTT – Retain

Trong giao thức MQTT, retain đóng vai trò là flag (lá cờ) gắn cho một tin nhắn. Bên cạnh đó, retain chỉ nhận những giá trị là 0 hoặc 1 (tương tự như giá trị false hoặc true). Trong đó, nếu retain là 1, broker sẽ giữ lại tin nhắn cuối cùng của 1 kênh kèm theo QoS tương ứng. Client sẽ nhận được tin nhắn ấy khi đăng ký vào kênh chứa tin nhắn được lưu lại.

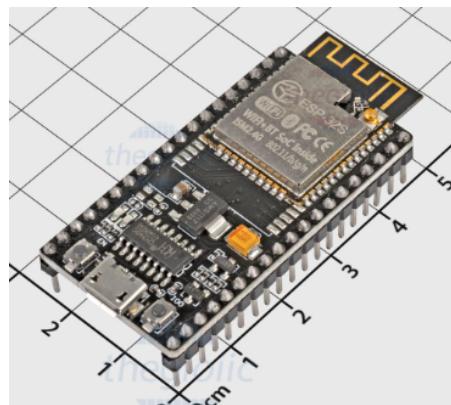
MQTT – Bridge

Dây là một tính năng quan trọng của MQTT broker. Chúng có vai trò giúp MQTT broker kết nối và trao đổi dữ liệu với nhau. Tính năng này chỉ được thực hiện khi có ít nhất 2 broker. Trong đó, có một broker sẽ biến đổi thành bridge với những thông số sau:

- Address: Tên địa chỉ broker cần kết nối
- Bridge_protocol_version: Phiên bản mới của MQTT được sử dụng cho 2 broker
- Topic: Bao gồm: tên topic được trao đổi giữa 2 broker, chiều trao đổi và topic mapping giữa 2 broker

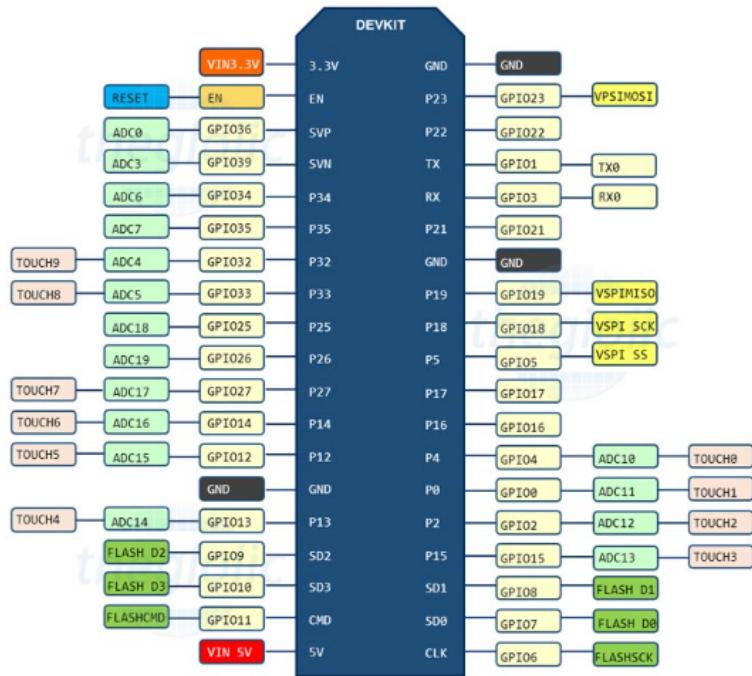
2.2 Linh kiện và module

2.2.1 Mcu ESP32



Hình 2: ESP32 thegioiic

ESP32 là một vi điều khiển hệ thống trên chip (SoC) có chi phí thấp và tiêu thụ điện năng thấp của Espressif Systems, nhà phát triển của vi điều khiển nổi tiếng ESP8266. ESP32 có tích hợp Wi-Fi và Bluetooth, và sử dụng bộ xử lý Xtensa LX6 32 bit của Tensilica với hai phiên bản là đơn nhân và kép nhân. ESP32 có thiết kế mạnh mẽ, tiết kiệm năng lượng và tích hợp nhiều tính năng như ADC, DAC, giao tiếp nối tiếp và nhiều chân GPIO có thể lập trình



Hình 3: Kiến trúc ESP32

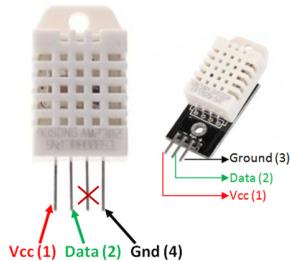
THÔNG SỐ KỸ THUẬT CỦA ESP32

- Bộ vi xử lý LX6 32-bit lõi đơn hoặc lõi kép với tần số xung nhịp lên tới 240 MHz
- 520 KB SRAM, 448 KB ROM và 16 KB RTC SRAM.
- Hỗ trợ kết nối Wi-Fi 802.11 b/g/n với tốc độ lên tới 150 Mbps.
- Hỗ trợ cả thông số kỹ thuật Classic Bluetooth v4.2 và BLE.
- 34 GPIO có thể lập trình.
- Lên đến 18 kênh SAR ADC 12 bit và 2 kênh DAC 8 bit
- Kết nối tiếp bao gồm 4 x SPI, 2 x I2C, 2 x I2S, 3 x UART.
- Ethernet MAC cho giao tiếp mạng LAN vật lý (yêu cầu PHY bên ngoài).
- 1 Bộ điều khiển máy chủ cho SD/SDIO/MMC và 1 Bộ điều khiển phụ cho SDIO/SPI.
- Động cơ xung điện và tối đa 16 kênh LED xung.
- Mã hóa khởi động và flash an toàn.
- Tăng tốc phần cứng mật mã cho AES, Hash (SHA-2), RSA, ECC và RNG

2.2.2 Cảm biến nhiệt độ và độ ẩm DHT11/22

DHT22 - CẢM BIẾN NHIỆT ĐỘ VÀ ĐỘ ẨM

Đối với cảm biến DHT22



Hình 4: Cảm biến nhiệt độ, độ ẩm

- 1 Vcc Nguồn điện 3.5V đến 5.5V
- 2 Dữ liệu Đầu ra cả Nhiệt độ và Độ ẩm thông qua Dữ liệu nối tiếp
- 3 NC Không có kết nối và do đó không được sử dụng
- 4 Đất Kết nối với mặt đất của mạch

Đối với mô-đun DHT22

- 1 Vcc Nguồn điện 3.5V đến 5.5V
- 2 Dữ liệu Đầu ra cả Nhiệt độ và Độ ẩm thông qua Dữ liệu nối tiếp
- 3 Đất kết nối với mặt đất của mạch

THÔNG SỐ KỸ THUẬT DHT22

- Điện áp hoạt động: 3.5V đến 5.5V
- Dòng hoạt động: 0,3mA (đo) 60uA (chế độ chờ)
- Đầu ra: Dữ liệu nối tiếp
- Phạm vi nhiệt độ: -40 ° C đến 80 ° C
- Phạm vi độ ẩm: 0
- Độ phân giải: Nhiệt độ và Độ ẩm đều là 16-bit
- Độ chính xác: ± 0,5 ° C và ± 1

2.2.3 Cảm biến khí ga MQ2



Hình 5: Cảm biến khí ga MQ2

Module:

- 1 Vcc Chân cấp nguồn, điện áp hoạt động thường là + 5V
- 2 Ground Chân nối đất
- 3 Digital Out Chân xuất đầu ra digital, bằng cách đặt giá trị ngưỡng trên chiết áp
- 4 Analog Out Chân xuất điện áp analog 0-5V dựa trên nồng độ khí

Cảm biến:

- 1 2 chân H Trong hai chân H, một chân được kết nối với nguồn cấp và chân kia nối đất
- 2 2 chân A Các chân A và chân B có thể hoán đổi cho nhau. Các chân này sẽ được mắc vào điện áp nguồn cấp.
- 3 2 chân B Các chân A và chân B có thể hoán đổi cho nhau. Một chân sẽ là đầu ra trong khi chân kia sẽ được nối đất.

THÔNG SỐ KỸ THUẬT :

- Điện áp hoạt động: 3.3V-5V
- Kích thước PCB: 3cm * 1.6cm
- Led đỏ báo nguồn vào, Led xanh báo gas
- IC so sánh : LM393
- VCC: 3.3V-5V
- GND: 0V
- DO: Đầu ra tín hiệu số (0 và 1)
- AO: Đầu ra Analog (Tín hiệu tương tự)
- Cấu tạo từ chất bản dẫn SnO₂
- Có 2 dạng tín hiệu: Analog(AO) và Digital (DO)
 - Dạng tín hiệu : TTL đầu ra 100mA (Có thể sử dụng trực tiếp Relay, Còi công suất nhỏ...)
 - Điều chỉnh độ nhạy bằng biến trở.
 - Sử dụng LM393 để chuyển AO → DO

2.2.4 Động cơ DC



Hình 6: Động cơ DC 3-6V

- Đường kính trục: 2MM
- Độ dài trục: 10MM
- Kích thước Động cơ: 20x28x15MM
- Động cơ chạy ổn định ở 3V
- Tốc độ: 3V 12.000r/Min
- Dòng tiêu thụ: 0.3-0.4A

2.3 NODE RED

Node-RED là một công cụ mã nguồn mở và trực quan được sử dụng để xây dựng các luồng làm việc (workflows) và ứng dụng Internet of Things (IoT). Nó cung cấp một giao diện đồ họa dựa trên trình duyệt web, cho phép người dùng kết nối các nút (node) với nhau để xử lý dữ liệu và tương tác với các thiết bị và dịch vụ khác nhau.

Node-RED được xây dựng trên nền tảng Node.js và sử dụng trình duyệt web để tạo ra một giao diện dễ sử dụng. Người dùng có thể kéo và thả các nút từ thư viện có sẵn để tạo ra luồng làm việc theo ý muốn. Các nút có thể thực hiện các nhiệm vụ khác nhau, bao gồm xử lý dữ liệu, kết nối và tương tác với các dịch vụ web, cơ sở dữ liệu, thiết bị IoT và nhiều hơn nữa.

Node-RED có một cộng đồng rộng lớn và có sẵn nhiều bộ nút mở rộng, cho phép người dùng tương tác với các nền tảng và dịch vụ phổ biến như MQTT, HTTP, MySQL, MongoDB, Twitter, Raspberry Pi, Arduino và nhiều hơn nữa. Công cụ này rất linh hoạt và phù hợp cho việc phát triển các ứng dụng IoT, tự động hóa và quản lý dữ liệu.

CÁC TÍNH NĂNG CỦA NODE-RED

Node-RED có nhiều tính năng hữu ích và mạnh mẽ để xây dựng và quản lý các luồng làm việc (workflows) và ứng dụng IoT. Dưới đây là một số tính năng chính của Node-RED:

Giao diện trực quan: Node-RED cung cấp một giao diện đồ họa dễ sử dụng dựa trên trình duyệt web. Người dùng có thể kéo và thả các nút để tạo và kết nối các luồng làm việc một cách trực quan.

Các nút và luồng đa dạng: Node-RED đi kèm với một bộ sưu tập nút mở rộng lớn, cho phép bạn tương tác với các thiết bị, dịch vụ và giao thức khác nhau. Các nút này bao gồm MQTT, HTTP, cơ sở dữ liệu, xử lý dữ liệu, gửi email, gửi tin nhắn và nhiều hơn nữa. Bạn cũng có thể tạo các nút tùy chỉnh của riêng mình.

Tích hợp dịch vụ web: Node-RED cho phép bạn tương tác với các dịch vụ web phổ biến như Twitter, Facebook, Dropbox và nhiều dịch vụ khác. Bạn có thể gửi yêu cầu API, nhận dữ liệu từ các dịch vụ này và thực hiện các hành động liên quan.

Tích hợp IoT: Node-RED được phát triển đặc biệt cho việc xây dựng ứng dụng IoT. Bạn có thể tương tác với các thiết bị như Raspberry Pi, Arduino, các cảm biến và các nền tảng IoT khác. Node-RED cung cấp các nút và giao thức như MQTT, CoAP và Modbus để liên kết và thu thập dữ liệu từ các thiết bị IoT.

Xử lý dữ liệu linh hoạt: Node-RED cho phép bạn xử lý, biến đổi và lọc dữ liệu theo ý muốn. Bạn có thể sử dụng các nút có sẵn để thực hiện các hoạt động như chuyển đổi định dạng, phân tích văn bản, tính toán, lưu trữ và đồng bộ dữ liệu.

Giao tiếp thời gian thực: Node-RED hỗ trợ giao tiếp thời gian thực và khả năng phát hiện sự kiện. Bạn có thể tạo các luồng làm việc để theo dõi, giám sát và phản ứng.

KHỐI CƠ BẢN TRÊN NODE-RED

STT	Biểu tượng	Chức năng
1		Khối input, nhập giá trị
2		Show ra màn hình debug
3		Khối tùy chỉnh chức năng
4		Hỗ trợ vẽ biểu đồ
5		MQTT in out

Hình 7: Khối cơ bản trong Node-RED

3 THIẾT KẾ VÀ THỰC HIỆN ỨNG DỤNG

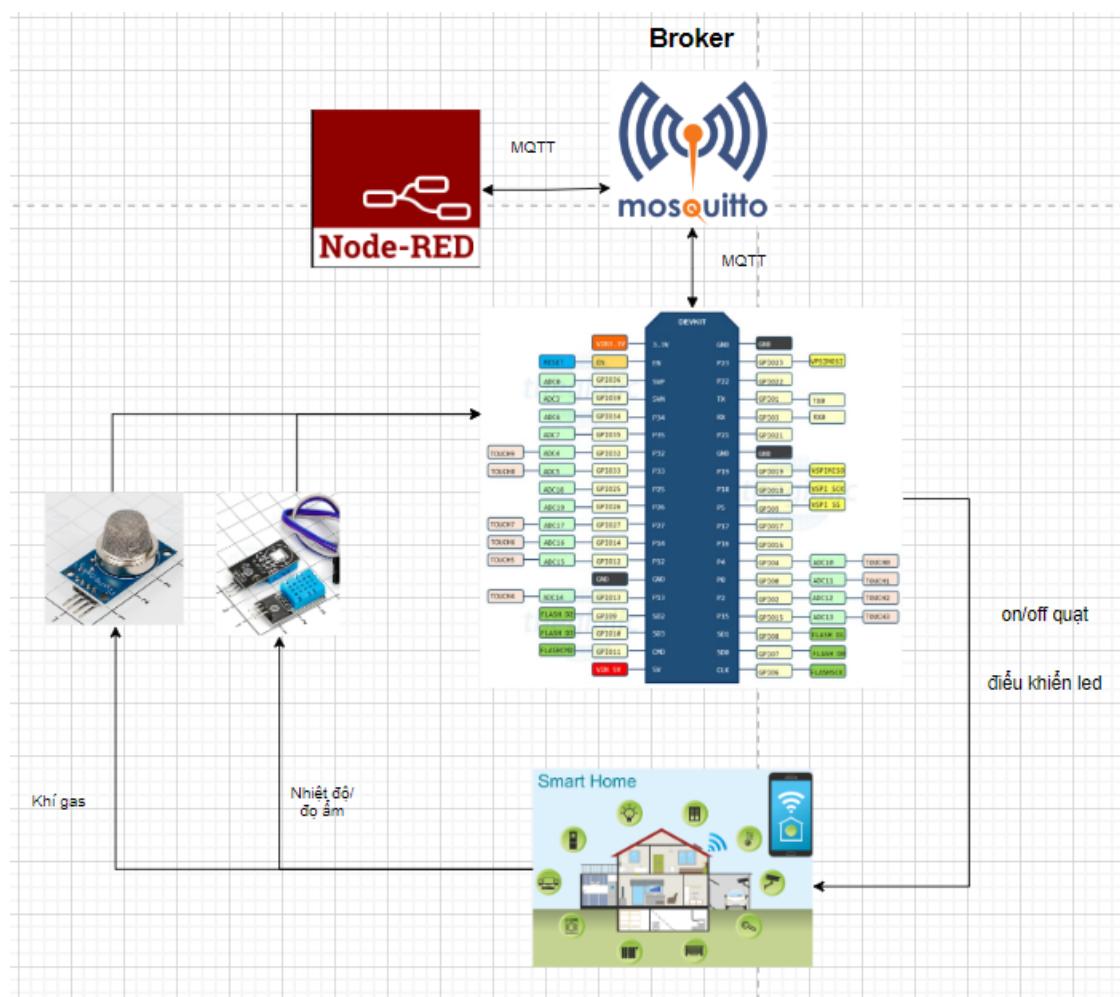
3.1 Giao đoạn thiết kế

Các thông số cần cần giám sát trong smarthome: nhiệt độ, độ ẩm, khí gas.

Phản ứng: nhóm sử dụng ESP32 do giá thành rẻ, thông dụng, tương thích với nhiều cảm biến.

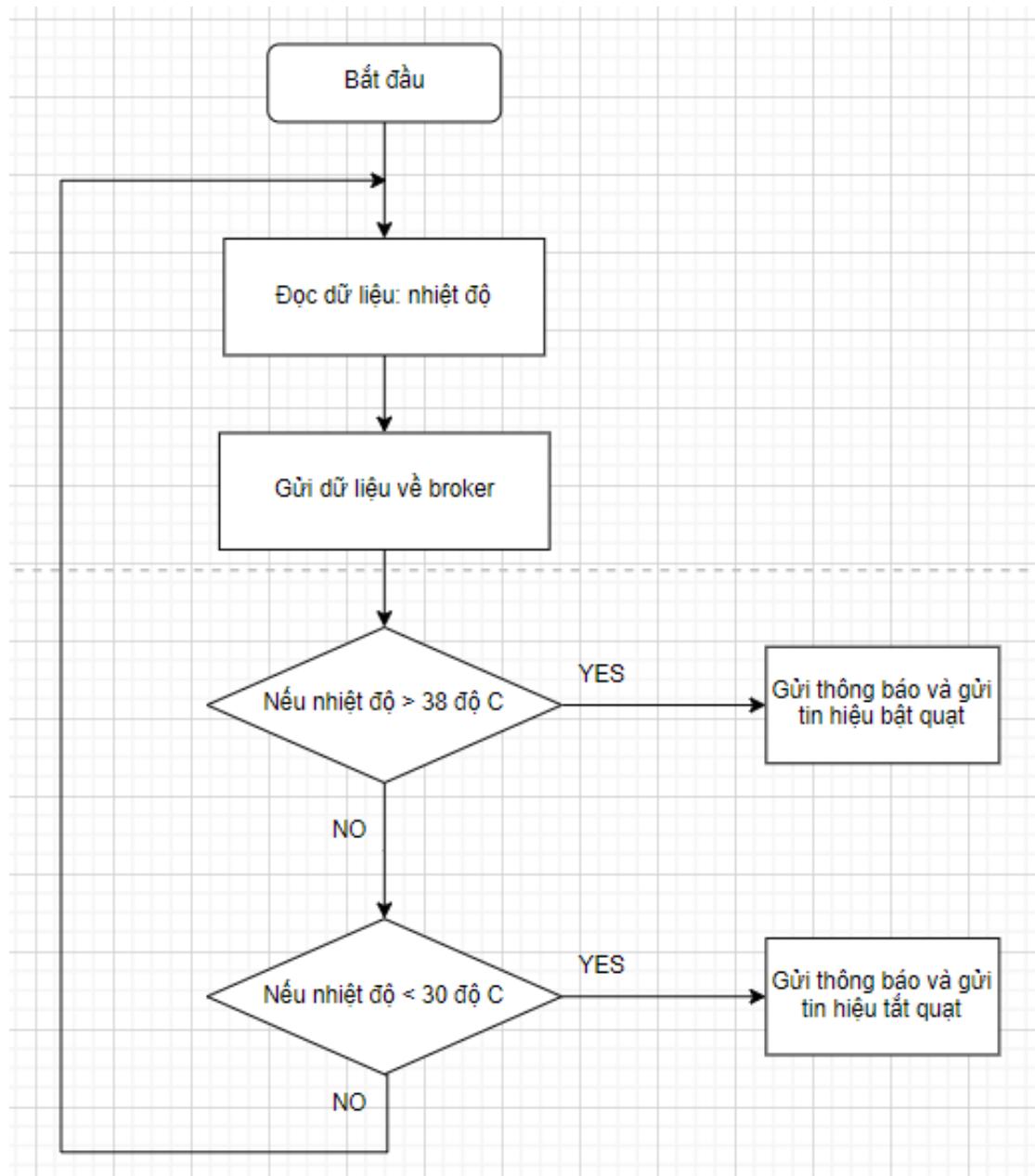
Về lập trình: nhóm chia thành lập tình cho vi điều khiển ESP32 (arduino IDE) và lập trình giao diện, chức năng với người sử dụng (node-RED).

Ý tưởng của nhóm được thể hiện qua sơ đồ sau

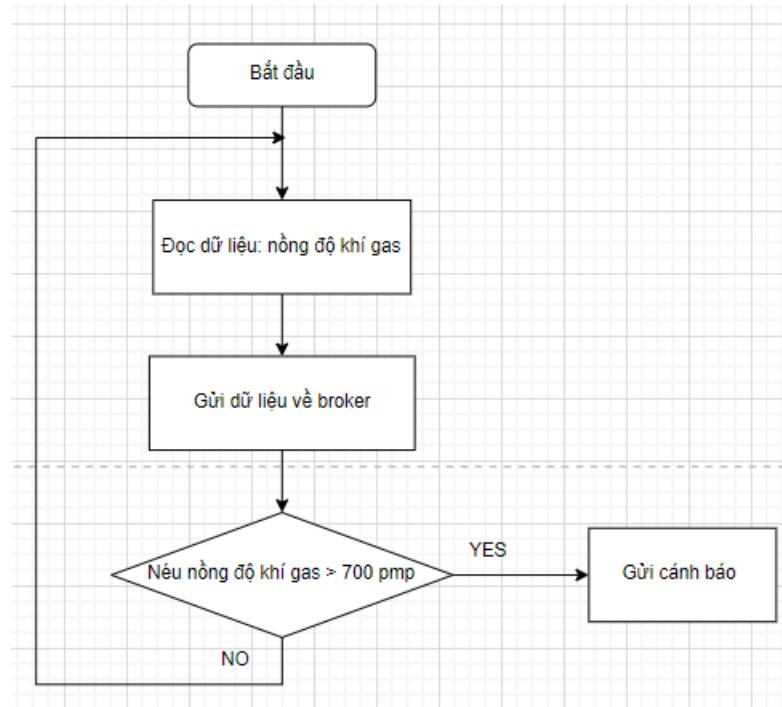


Hình 8: Ý tưởng thiết kế dự án

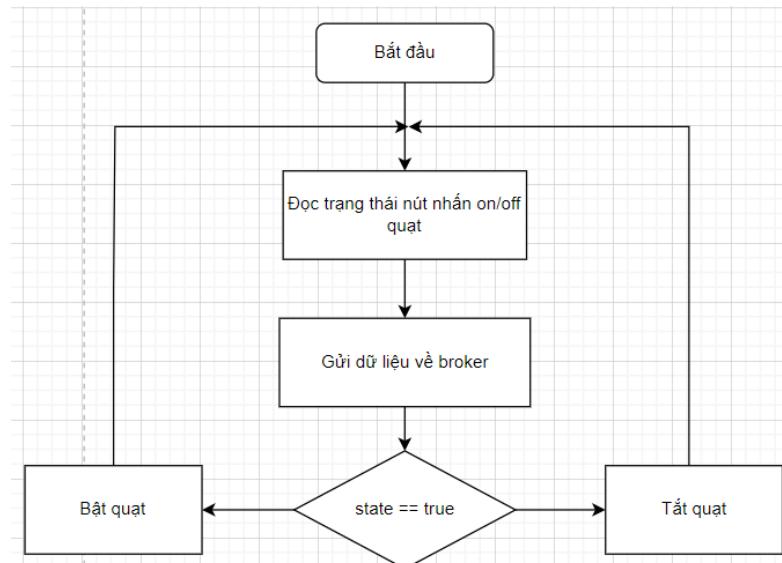
3.2 Flow chart



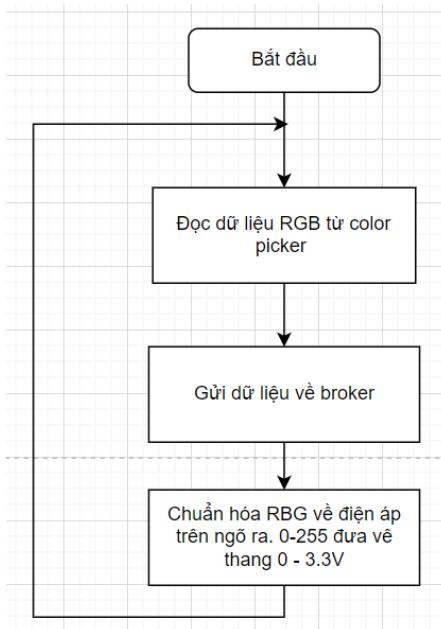
Hình 9: Flow chart điều khiển bật tắt quạt tự động



Hình 10: Flow chart cảnh báo khí gas cao



Hình 11: Flow chart bật/tắt quạt



Hình 12: Flow chart điều khiển màu led RGB

3.3 Thực hiện phần cứng, giao diện người dùng

3.3.1 Cài đặt node-RED

- Node.js <https://nodejs.org/en>
- Eclipse Mosquitto <https://mosquitto.org/>
- MQTT explorer <https://mqtt-explorer.com/>

Cài đặt node-red và dashboard

```
npm install -g --unsafe-perm node-red node-red-dashboard
```

3.3.2 Thực hiện ứng dụng

Cài đặt wifi

```
const char* ssid      = "07.13";
const char* password = "071307130713";
const char *mqtt_server = "192.168.1.95";
const char *mqtt_user = "";
const char *mqtt_password = "";
const int mqtt_port = 1883;
```

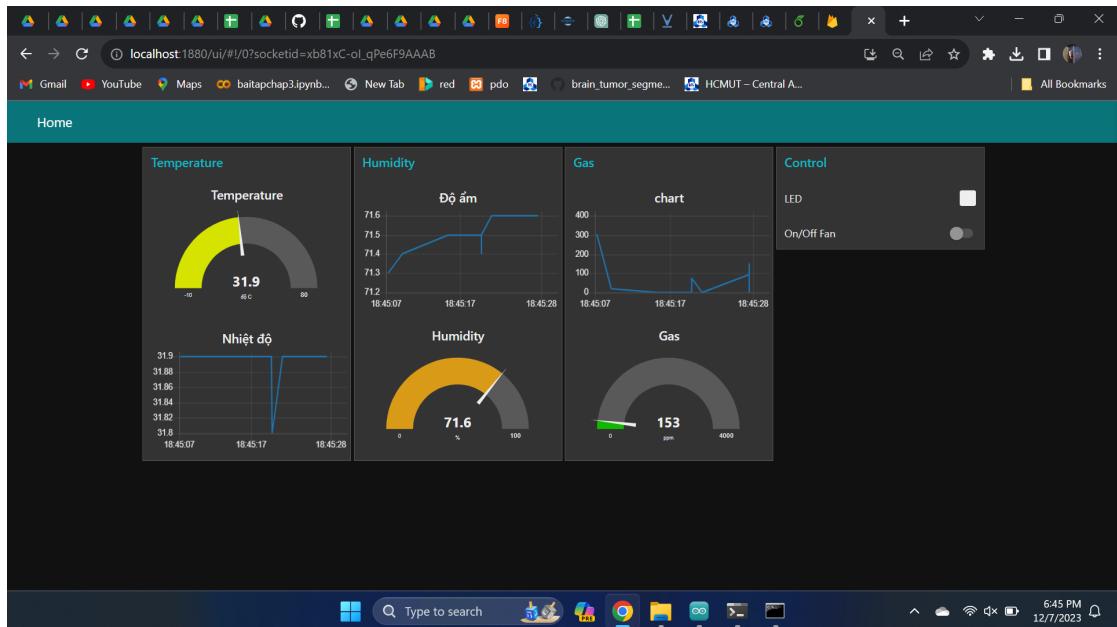
Hàm ESP gửi giá trị cảm biến lên Server

```
void sendMQTTvalues(float temp, float hum, float gas)
{
```



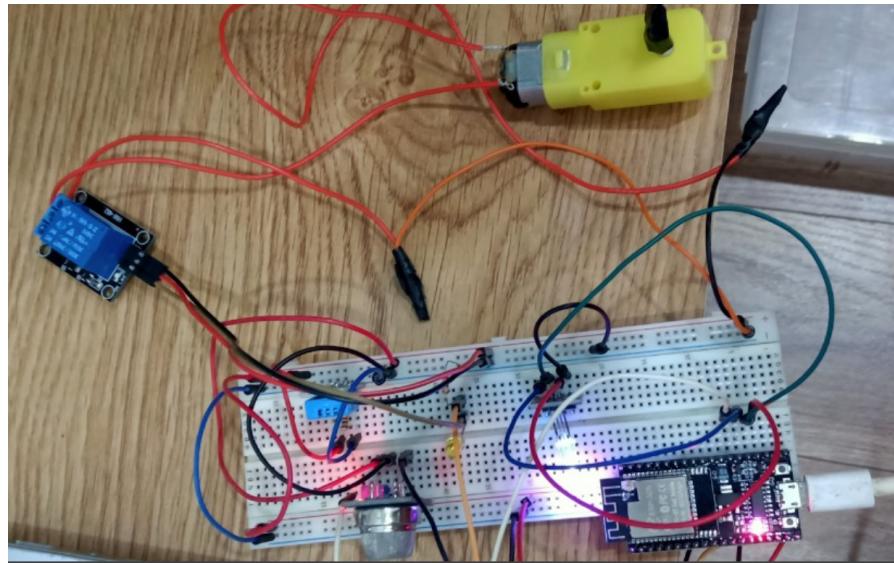
```
StaticJsonDocument<256> doc;  
  
doc["device"] = "ESP32";  
doc["temperature"] = temp;  
doc["humidity"] = hum;  
doc["gas"] = gas;  
  
char buff[256];  
serializeJson(doc, buff);  
client.publish("ESPValues",buff);  
}  
  
Hàm điều khiển led RGB và điều khiển on/off quạt  
  
void setColor(int R, int G, int B) {  
    analogWrite(PIN_RED, R);  
    analogWrite(PIN_GREEN, G);  
    analogWrite(PIN_BLUE, B);  
}  
  
void controlFan(String isFanOn, int speed) {  
    isFanOn == "true" ? analogWrite(FAN, speed) : analogWrite(FAN, 0);  
}
```

Giao diện dashboard node-RED dashboard



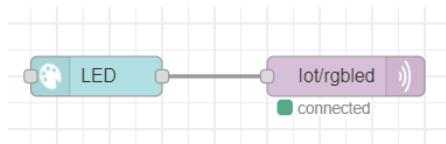
Hình 13: Giao diện dashboard của ứng dụng

Lắp mạch thực tế

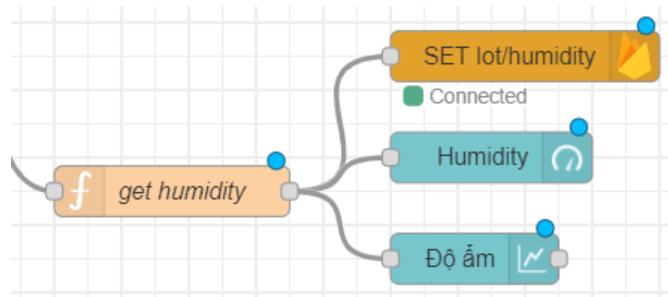


Hình 14: Mạch thực tế với ESP32

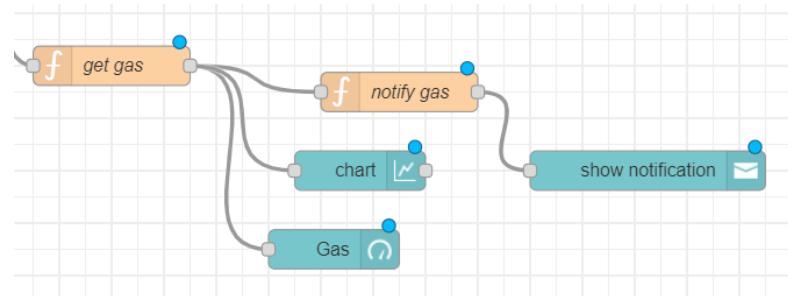
Flow trên node-RED



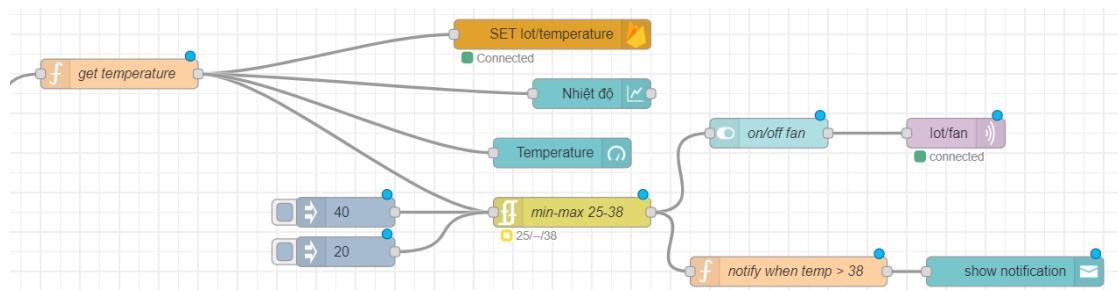
Hình 15: Flow điều khiển màu đèn led RGB



Hình 16: Flow đọc và mô hình hóa thông số độ ẩm



Hình 17: Flow đọc và mô hình hóa thông số gas



Hình 18: Flow đọc và mô hình hóa thông số Nhiệt độ



4 KẾT LUẬN

4.1 Kết quả đạt được

Sản phẩm sau khi thực hiện đã đạt được những yêu cầu chức năng cơ bản: đo nhiệt độ, độ ẩm, khí gas; biểu đồ, đồ thị hóa các thông số; giao diện điều khiển trực quan.

Trong quá trình thực hiện, nhóm gặp nhiều khó khăn trong việc cài đặt phần mềm, thư viện, tìm hiểu cách sử dụng công cụ đặc biệt trong việc test MQTT.

Sản phẩm cuối cùng chưa có nhóm chức năng về báo cáo hiệu suất, phân tích dữ liệu cảm biến, và sản phẩm chủ được thử nghiệm trên môi trường local. Nhưng qua đó, nhóm cũng có được những kinh nghiệm và kiến thức bổ ích như:

- Hiểu được giao thức MQTT và mô hình IoT với MQTT.
- Hiểu kiến trúc ESP32 và cách giao tiếp, điều khiển ngoại vi với ESP32.
- Hiểu cách sử dụng, thiết kế flow trên node-RED.
- Hiểu và khắc phục các khó khăn khi thiết kế, triển khai dự án IoT trên thực tế.

4.2 Hướng phát triển

Qua quá trình sử dụng ứng dụng, nhóm nhận thấy giao diện node-RED dashboard (mặc định) tuy dễ sử dụng, đầy đủ tính năng, tuy nhiên giao diện chưa thân thiện, khó tùy chỉnh với người sử dụng. Việc xây dựng giao diện lại là điều cần thiết khi triển khai ứng dụng ra thực tế.

Một số chức năng chưa hoàn thiện như lưu trữ dữ liệu để phân tích và báo cáo về hiệu suất sử dụng, trạng thái thiết bị là hướng đi tiếp theo của đề tài và rất mong nhận được sự giúp đỡ và góp ý về mọi mặt của những ai quan tâm.



5 TÀI LIỆU THAM KHẢO:

Tài liệu

- [1] Document node-RED, <https://nodered.org/docs/>
- [2] Markus Edenhauser, *Node-RED: IoT projects with ESP32, MQTT and Docker*, <https://www.udemy.com/course/node-red-iot-projects-with-esp32/>
- [3] Trần Văn Tiệp, *Thiết kế mô hình smart home đơn giản sử dụng module wifi ESP8266*, <https://www.studocu.com/vn/document/truong-cao-dang-thuc-hanh-fpt/quan-he-cong-chung/smart-home-teacher-andy/40300955>
- [4] Thanh Thư, *MQTT là gì? Vai trò của MQTT trong IoT*, <https://viblo.asia/p/mqtt-la-gi-vai-tro-cua-mqtt-trong-iot-V3m5WL3bK07>



6 PHỤ LỤC

6.1 source code (file Node-RED flows.json)

<https://drive.google.com/drive/folders/1htmsV1qCJrh-kTZQByAjyVn84Jv2rWwr?usp=sharing>

6.2 Fix lỗi không kết nối broker trên local, cài đặt node-red admin

<https://drive.google.com/file/d/1TNhUkHUFR6sU020rlQQcVXlAOUSV2dJE/view?usp=sharing>