

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
KHOA ĐIỆN – ĐIỆN TỬ

ĐỒ ÁN TỐT NGHIỆP
ĐẠI HỌC



NGÀNH: CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ
CHUYÊN NGÀNH: KỸ THUẬT ĐIỆN TỬ

ĐỀ TÀI:

THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH

Người hướng dẫn: ThS Trần Duy Chung

Sinh viên thực hiện: Phạm Văn Duẩn

Đoàn Liên

Mã sinh viên: 1811505120109 Lớp: 18D3

1811505120331 Lớp: 18D4

Đà Nẵng, 02/2022

ĐẠI HỌC ĐÀ NẴNG
TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT
KHOA ĐIỆN – ĐIỆN TỬ

ĐỒ ÁN TỐT NGHIỆP
ĐẠI HỌC



NGÀNH: CÔNG NGHỆ KỸ THUẬT ĐIỆN – ĐIỆN TỬ
CHUYÊN NGÀNH: KỸ THUẬT ĐIỆN TỬ

ĐỀ TÀI:

THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH

Người hướng dẫn: ThS Trần Duy Chung

Sinh viên thực hiện: Phạm Văn Dẫn

Đoàn Liên

Mã sinh viên: 1811505120109 Lớp: 18D3

1811505120331 Lớp: 18D4

Đà Nẵng, 02/2022

NHẬN XÉT ĐỒ ÁN TỐT NGHIỆP
(Dành cho người hướng dẫn)

1. Thông tin chung:

1. Họ và tên sinh viên: PHẠM VĂN DUẤN
ĐOÀN LIỀN

2. Lớp: 18D3 Mã SV: 1811505120109
18D4 1811505120331

3. Tên đề tài: THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG
NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH

4. Người hướng dẫn: Trần Duy Chung Học hàm/ học vị: Thạc sỹ

II. Nhận xét, đánh giá đồ án tốt nghiệp:

1. Về tính cấp thiết, tính mới, mục tiêu của đề tài:

Sử dụng nền tảng Openhab là một nền tảng nguồn mở được xây dựng cho các ứng dụng Smart Home. OpenHAB hoạt động như một trung tâm điều khiển, có thể cài đặt được trên nhiều nền tảng hệ điều hành và thiết bị khác nhau như Linux, macOS, Windows, Raspberry Pi... và truy cập được dưới dạng ứng dụng web, ứng dụng Android hoặc IOS. OpenHAB có thể xây dựng một ứng dụng có tích hợp giao thức MQTT với bảo mật TLS/SSL.

2. Về kết quả giải quyết các nội dung nhiệm vụ yêu cầu của đồ án:

Qua thời gian nghiên cứu tìm hiểu nhóm thực hiện đề tài đã hoàn thiện và giải quyết các vấn đề cần thiết cho mô hình để có thể đưa vào việc sử dụng thực tiễn và đạt được các yêu cầu đã đề ra:

- + Sử dụng mạng WiFi để truyền dẫn và đồng bộ hóa dữ liệu theo thời gian.
- + Sử dụng cảm biến nhiệt độ ,cảm biến độ ẩm ,cảm biến khí gas, các số liệu về thời tiết, các thiết bị đèn và hệ thống đo lường để lấy dữ liệu kiểm tra giám sát và điều khiển các thiết bị trong nhà.
- + Sử dụng Board mạch Node MCU ESP8266 cho phép kết nối máy tính thông các giao thức MQTT.

- + Các tín hiệu hiển thị đến người dùng là dạng số và biểu đồ cập nhật theo thời gian, giao diện web/app dựa trên nền tảng Openhab để điều khiển và hiển thị trạng thái các thiết bị.

Tài liệu:

Tìm hiểu các thông tin qua các trạng mạng, và tham khảo các sách, báo tài liệu chuyên ngành.

Tài liệu về Cloud Server Openhab.

Tài liệu về lập trình WEB.

Tài liệu về lập trình vi điều khiển, các chuẩn giao tiếp...

Tài liệu về giao thức MQTT bảo mật TLS .

Tài liệu về công nghệ IOT.

3. Về hình thức, cấu trúc, bố cục của đồ án tốt nghiệp:

Trình bày thuyết minh theo format chuẩn nhưng còn một vài lỗi nhỏ..

4. Kết quả đạt được, giá trị khoa học, khả năng ứng dụng của đề tài:

- + Điều khiển thiết bị điện của nhà thông minh qua nền tảng OpenHab cả trên máy tính và điện thoại: điều khiển và giám sát tất cả các khối.
- + Điều khiển thủ công: dùng nút nhấn khối Master điều khiển chính nó hoặc điều khiển các Slave; điều khiển nút nhấn trực tiếp tại các Slave.
- + Hệ thống tự động giám sát và hiển thị, khi khí gas rò rỉ sẽ cảnh báo đến người dùng.
- + Hệ thống hoạt động ổn định
- + Giao diện thiết kế dễ sử dụng và đẹp mắt.

5. Các tồn tại, thiếu sót cần bổ sung, chỉnh sửa:

- + Mở rộng số thiết bị cần điều khiển với công suất lớn.
- + Mở rộng thêm nhiều Slave để mở rộng mạng lưới giám sát và điều khiển.
- + Phát triển thêm hệ thống dập lửa khi phát hiện có cháy xảy ra.
- + Thêm camera giám sát ngôi nhà thông qua web hoặc android.

- + Phát triển ứng dụng điều khiển thiết bị trên website và điều khiển thiết bị bằng giọng nói.
- + Tích hợp thêm các lớp bảo mật khác từ phần cứng đến phần mềm.
- + Cải tiến phần cứng dần trở nên nhỏ gọn, ổn định.

III. Tinh thần, thái độ làm việc của sinh viên:

- + Có tinh thần trách nhiệm và nhận thức được nhiệm vụ khi được giao Đồ án Tốt nghiệp.
- + Ham học hỏi và rất tận tâm trong quá trình làm Đồ án.
- + Thường xuyên trao đổi với Giảng viên trong quá trình làm Đồ án.

IV. Đánh giá:

1. Điểm đánh giá: 9,5/10 (lấy đến 1 số lẻ thập phân)
2. Đề nghị: Được bảo vệ đồ án Bổ sung để bảo vệ Không được bảo vệ

Đà Nẵng, ngày 21 tháng 2 năm 2022

Người hướng dẫn

ThS. Trần Duy Chung

TÓM TẮT

Tên đề tài: **THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH**

Sinh viên thực hiện: **Phạm Văn Tuấn** Mã SV: **1811505120109** Lớp: **18D3**

Đoàn Liên Mã SV: **1811505120331** Lớp: **18D4**

Nhóm thực hiện đề tài “THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH” với giao thức MQTT tích hợp bảo mật SSL/TLS để điều khiển và giám sát nhiệt độ, độ ẩm, đo điện áp và dòng điện... các thiết bị điện sử dụng trong nhà thông qua nền tảng OpenHab và App điện thoại thông qua mạng Wifi. Sử dụng ESP8266 truyền qua MQTT Broker với giao thức MQTT. MQTT Broker ở đây đóng vai trò là một trung gian để nhận và truyền dữ liệu đến các client.

Kết quả đạt được của hệ thống sẽ thiết kế được hệ thống với mục đích giám sát và điều khiển một cách thông minh, giúp điều khiển giám sát nhanh chóng và dễ dàng, phát hiện và cảnh báo sự cố nhanh, giảm bớt thời gian thực hiện.

Nội dung mà nhóm thực hiện gồm : Nghiên cứu 1 cách tổng quan, module liên quan và các cơ sở lý thuyết, thiết kế và xây dựng được phần mềm, phần cứng . Sau đó kiểm tra kết nối phần cứng và phần mềm chạy thử nghiệm, rồi từ đó hoàn thiện kết nối tất cả thành một hệ thống điều khiển từ phần mềm sang phần cứng và ngược lại.

Ở trong đề tài này, nhóm sẽ tiến hành và đề xuất ra các phương án kỹ thuật để thiết kế và vận hành các thiết bị điện trong nhà một cách thông minh.

Đề tài nhóm em gồm có 4 chương :

Chương 1 : Tổng quan :

- Đặt vấn đề
- Hướng giải quyết vấn đề
- Mục tiêu
- Nội dung
- Giới hạn
- Bố cục

Chương 2 : Cơ sở lý thuyết :

- Tổng quan công nghệ IoT
- Giới thiệu các thiết bị vào ra sử dụng trong đề tài
- Giới thiệu phần cứng
- Các chuẩn truyền dữ liệu

- Giới thiệu nền tảng OpenHab

Chương 3: Thiết kế hệ thống

- Giới thiệu
- Thiết kế hệ thống
- Thi công
- Lập trình hệ thống
- Kết quả, nhận xét, đánh giá

Mục đích mà nhóm mong muốn đạt được sau khi hoàn thành sản phẩm là:

- Biết và thành thạo lập trình esp8266 giao tiếp wifi với phần mềm Arduino IDE, giao thức truyền MQTT, biết được phương pháp để điều khiển, giám sát các thiết bị trong nhà một cách dễ dàng và thuận tiện.
- Thiết kế được giao diện Web Server để giám sát trên máy tính và di động với phần mềm visual studio code dùng để lập trình giao diện nền tảng OpenHab
- Xây dựng được mô hình mô phỏng.
- Giúp nhóm phát triển hơn các kỹ năng, nghiên cứu học hỏi, làm việc nhóm, tìm kiếm tài liệu, tự tin thuyết trình.

ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

1. Họ và tên sinh viên: PHẠM VĂN DUẤN

ĐOÀN LIỀN

2. Mã sinh viên: 1811505120109 **Lớp:** 18D3

1811505120331 **Lớp:** 18D4

3. Họ và tên người hướng dẫn: Th.s TRẦN DUY CHUNG

4. Đề tài:

Tên đề tài: THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG
NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH

Thời gian thực hiện: Từ ngày: 11/10/2021 *đến ngày:* 18/02/2022

5. Mô tả đề tài.

Khi cuộc sống con người được nâng cao, những nhu cầu cuộc sống hằng ngày càng cao đòi hỏi phải được hỗ trợ tốt hơn. Và từ những nhu cầu thực tế đó ý tưởng về ngôi nhà thông minh hình thành. Một ngôi nhà mà chứa đựng sự ấm áp yêu thương, tình yêu và hạnh phúc, mọi hoạt động của con người đều được hỗ trợ và giúp đỡ một cách linh hoạt, không những giúp đỡ mà ngôi nhà còn quản lý một cách thông minh. Ngày nay, với sự phát triển một cách nhanh chóng của ngành kỹ thuật điện tử cũng như nhiều ngành khác thì ý tưởng về ngôi nhà thông minh không còn vướng bởi rào cản công nghệ. Việc điều khiển nhà thông minh thông qua Smartphone tạo nên bước ngoặt lớn trong việc điều khiển tự động, không dây một cách linh hoạt, có thể nói sự phát triển không ngừng của những chiếc Smartphone đã làm cho công nghệ thêm bước tiến, việc điều khiển dễ dàng hơn.

Từ ý Tưởng đó, nhu cầu về cuộc sống thoải mái nên việc xây dựng “THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO

NGÔI NHÀ THÔNG MINH” để đáp ứng phần nào trong cuộc sống mọi người. Do thời gian nghiên cứu và thực hiện đề tài chỉ giới hạn trong một khoảng thời gian nhất định, vốn kiến thức và việc tìm hiểu sâu về một hệ thống còn hạn chế nên đề tài này cũng có nhiều thiếu sót đáng kết, mong quý thầy cô và người đọc đóng góp ý kiến để đề tài được hoàn thiện hơn.

6. Mục tiêu.

Mục tiêu xây dựng được một hệ thống điều khiển, giám sát các thiết bị trong nhà một cách dễ dàng nhất làm cho cuộc sống thuận tiện hơn.

- + Điều khiển bật/ tắt các thiết bị trong gia đình như đèn, quạt...
- + Giám sát các chỉ số nhiệt độ, độ ẩm, ánh sáng, khí gas trong phòng.
- + Theo dõi năng lượng điện tiêu thụ trong gia đình.
- + Bảo mật hệ thống.

Với đề tài này giúp sinh viên có nhiều kiến thức về các thiết bị, tập thói quen nghiên cứu phục vụ công việc, có thể ứng dụng trong thực tế.

7. Tính mới của đề tài:

- Sử dụng nền tảng Openhab là một nền tảng nguồn mở được xây dựng cho các ứng dụng Smart Home. OpenHAB hoạt động như một trung tâm điều khiển, có thể cài đặt được trên nhiều nền tảng hệ điều hành và thiết bị khác nhau như Linux, macOS, Windows, Raspberry Pi... và truy cập được dưới dạng ứng dụng web, ứng dụng Android hoặc iOS. OpenHAB có thể xây dựng một ứng dụng có tích hợp giao thức MQTT với bảo mật TLS/SSL.

- TLS/SSL đang được sử dụng rộng rãi để mã hóa dữ liệu. Giao thức bảo mật này cung cấp sự an toàn cho dữ liệu nhằm đảm bảo không ai có thể đọc trộm nội dung trong quá trình truyền tin ngoài client và server, ngoài ra giao thức còn sử dụng các chứng chỉ xác thực số nhằm xác thực thông tin ở cả hai phía client và server.

8. Phạm vi, đối tượng nghiên cứu:

- Đối tượng : THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH

- Phạm vi hoạt động:

Hệ thống ứng dụng cho một căn hộ từ bình dân cho đến cao cấp hay biệt thự lớn hoặc một ngôi nhà thông minh với đầy đủ tính năng toàn diện.

- Phương pháp nghiên cứu:

+ Tìm hiểu lý thuyết và đi đôi với thực hành.

+ Khảo sát, đánh giá các môi trường thực tế và tìm ra những giải pháp hay để phát triển hệ thống.

9. Kết quả dự kiến đạt được :

- Hoàn thành sản phẩm có thể giám sát đo được các tham số các yêu cầu đặt ra .

8. Tiến độ thực hiện

TT	Thời gian	Nội dung công việc	Kết quả dự kiến đạt được
1	Tuần 1	Lên ý tưởng cho đề tài Tìm hiểu và chọn đề tài lên kế hoạch chi tiết	
2	Tuần 2	Chọn đề tài và ghi đề cương nhiệm vụ của đề tài	
3	Tuần 3	Tìm hiểu các đề tài và xây dựng một hệ thống cảnh báo cho căn hộ	
4	Tuần 4	Gặp GVHD để báo cáo về hướng thực hiện đề tài.	
5	Tuần 5	Tìm hiểu về giao thức MQTT và nền tảng Openhab.	
6	Tuần 6	Nghiên cứu thiết kế chế tạo phần cứng .	
7	Tuần 7	Lập trình điều khiển	
8	Tuần 8	Lập trình điều khiển giao tiếp về phần cứng	
9	Tuần 9	Lập trình điều khiển , lắp ráp linh kiện	

10	Tuần 10	Sắp xếp bố trí cho hợp lí	
11	Tuần 11	Báo cáo tiến độ cho GVHD. Hoàn thiện mô hình.	
12	Tuần 12	Viết báo cáo.	
13	Tuần 13	Kiểm tra hoạt động của hệ thống.	
14	Tuần 14	Hoàn tất báo cáo. Khắc phục các sự cố.	
15	Tuần 15	Vá lỗi các sự cố nếu có. Hoàn thiện chính chu báo cáo, chuẩn bị gặp GVPB và bảo vệ.	

Đà Nẵng, ngày 11 tháng 10 năm 2021.

BỘ MÔN DUYỆT

NGƯỜI HƯỚNG DẪN

SINH VIÊN

ThS. Trần Duy Chung

Phạm Văn Duẩn – Đoàn Liên

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Giảng viên hướng dẫn: ThS TRẦN DUY CHUNG

Sinh viên thực hiện: PHẠM VĂN DUẤN

Mã SV: 1811505120109

ĐOÀN LIỀN

Mã SV: 1811505120331

1. Tên đề tài: THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH

2. Các số liệu, tài liệu ban đầu:

Số liệu:

- Sử dụng mạng WiFi để truyền dẫn và đồng bộ hóa dữ liệu theo thời gian.
- Sử dụng cảm biến nhiệt độ, cảm biến độ ẩm, cảm biến khí gas, các số liệu về thời tiết, các thiết bị đèn và hệ thống đo lường để lấy dữ liệu kiểm tra giám sát và điều khiển các thiết bị trong nhà.
- Sử dụng Board mạch Node MCU ESP8266 cho phép kết nối máy tính thông các giao thức MQTT.
- Các tín hiệu hiển thị đến người dùng là dạng số và biểu đồ cập nhật theo thời gian, giao diện web/app dựa trên nền tảng Openhab để điều khiển và hiển thị trạng thái các thiết bị.

Tài liệu ban đầu:

- Tìm hiểu các thông tin qua các trang mạng, và tham khảo các sách, báo tài liệu chuyên ngành.
- Tài liệu về Cloud Server Openhab.
- Tài liệu về lập trình WEB.
- Tài liệu về lập trình vi điều khiển, các chuẩn giao tiếp...
- Tài liệu về giao thức MQTT bảo mật TLS.
- Tài liệu về công nghệ IOT.

3. Nội dung chính của đồ án:

“THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH” để quản lý và điều khiển các thiết bị điện trong nhà, thông qua Web/app có thể hiển thị được trạng thái, điều khiển được thiết bị và quan sát được các số liệu.

• Chương 1: Tổng Quan chung

Trình bày về đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đồ án.

• Chương 2: Cơ Sở Lý Thuyết

Trình bày về các lý thuyết có liên quan đến các vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài.

- **Chương 3: Thiết Kế Và Thi Công,**

Giới thiệu tổng quan về các yêu cầu của đề tài mà mình thiết kế gồm những phần nào. Thiết kế sơ đồ khối hệ thống, sơ đồ nguyên lý toàn mạch, tính toán thiết kế mạch. Trình bày về quá trình vẽ mạch in lắp ráp các thiết bị, đo kiểm tra mạch, lắp ráp mô hình. Thiết kế lưu đồ giải thuật cho chương trình và viết chương trình cho hệ thống. Trình bày về những kết quả đã được mục tiêu đề ra sau quá trình nghiên cứu thi công. Từ những kết quả đạt được để đánh giá quá trình hoàn thành được bao nhiêu phần trăm.

- **Kết Luận Và Hướng Phát Triển**

Trình bày về những kết quả mà đề án đạt được, những hạn chế, từ đó rút ra kết luận và hướng phát triển để giải quyết các vấn đề tồn đọng để đề án hoàn thiện hơn.

4. Các sản phẩm dự kiến

- Hoàn thành sản phẩm có thể giám sát đo các tham số yêu cầu đề ra.
- Báo cáo tổng kết đề án tốt nghiệp

5. Ngày giao đồ án: 11/10/2021

6. Ngày nộp đồ án: 18/02/2022

Đà Nẵng, ngày 11 tháng 10 năm 2021

Trưởng Bộ môn

Người hướng dẫn

ThS. Trần Duy Chung

LỜI NÓI ĐẦU

Ngày nay với sự phát triển của khoa học công nghệ, vi điều khiển và điện tử viễn thông ngày càng thông dụng thì các sản phẩm của được tạo ra từ các kết quả nghiên cứu chế tạo được ứng dụng vào hầu hết tất cả các lĩnh vực đời sống sinh hoạt và sản xuất để đáp ứng nhu cầu của xã hội, giúp con người tiếp cận gần hơn với các thiết bị điện tử hiện đại. Đặc biệt trong những năm gần đây nhà thông minh (smart home) xuất hiện và ngày càng phổ biến từ khách sạn, các Resort sang trọng cho đến những ngôi nhà ở hiện đại đều được lắp đặt hệ thống điều khiển thông minh. Theo xu hướng phát triển của xã hội và tính ứng dụng cao chúng em quyết định lựa chọn nghiên cứu đề tài: **“THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH”**.

Việc hoàn thành đề án lần này giúp chúng em nắm chắc và hiểu rõ hơn các kiến thức đã học, từ đó vận dụng những kiến thức đã học để tạo ra các sản phẩm thực tế, đáp ứng được nhu cầu của xã hội. Trong quá trình thực hiện thì không thể tránh khỏi việc gặp những khó khăn, do đó sẽ kích thích sinh viên tư duy để tìm ra phương án tối ưu và trao đổi thảo luận với thầy cô, bạn bè.

Trong quá trình thực hiện chúng em đã cố gắng hết sức để hoàn thiện một cách tốt nhất. Tuy nhiên do hạn chế về kinh nghiệm thực tế và thời gian thực hiện nên việc giải quyết đề tài là không thể tránh khỏi những thiếu sót. Do đó rất mong các thầy cô và các bạn sinh viên đóng góp ý kiến để chúng em có thể hoàn thiện tốt hơn.

Chúng em xin gửi lời cảm ơn sâu sắc đến thầy Trần Duy Chung. Thầy đã trực tiếp hướng dẫn, theo sát, chỉ bảo nhiệt tình giúp đỡ chúng em từ khi bắt đầu cho đến khi hoàn thành đề tài.

Chúng em cũng xin chân thành cảm ơn quý Thầy, Cô trong Khoa Điện trường Đại học Sư Phạm Kỹ Thuật đã truyền đạt kiến thức trong suốt thời gian chúng em học tập tại trường.

Chúng em xin chân thành cảm ơn!

Sinh viên thực hiện đề tài
Phạm Văn Duẩn – Đoàn Liên

CAM ĐOAN

Chúng em xin cam đoan đây là đề tài nghiên cứu của nhóm chúng em cùng với sự hướng dẫn tận tình của thầy **Trần Duy Chung**

Các nội dung nghiên cứu, số liệu và kết quả đã nêu trong đề án là trung thực và chưa từng được bảo vệ hay báo cáo trước đó. Trong báo cáo có chúng em có tham khảo một số tài liệu và đã được liệt kê ở phần tài liệu tham khảo.

Người thực hiện đề tài
Phạm Văn Duẩn – Đoàn Liên

MỤC LỤC

TÓM TẮT	vi
ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP	viii
NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP	xii
CAM ĐOAN.....	xv
MỤC LỤC	xvi
DANH SÁCH CÁC BẢNG HÌNH VẼ.....	xix
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT	xxi
MỞ ĐẦU.....	1
CHƯƠNG 1: TỔNG QUAN.....	2
1.1 Đặt vấn đề	2
1.2 Hướng giải quyết vấn đề.....	3
1.3 Mục tiêu	3
1.4 Nội dung nghiên cứu	4
1.5 Giới hạn đề tài	4
1.6 Bố cục	4
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ	6
2.1 Tổng quan về công nghệ IOT.....	6
2.1.1 Công nghệ IOT là gì?.....	6
2.1.2 Internet of Things - Xu hướng công nghệ tương lai?	6
2.1.3 Xu hướng và Tính chất của The Internet Of Things Thông minh.....	6
2.1.4 Kiến trúc dựa trên sự kiện	6
2.1.5 Là một hệ thống phức tạp	7
2.1.6 Kích thước.....	7
2.1.7 Vấn đề không gian, thời gian.....	7
2.1.8 Ứng dụng của IoT	7
2.2 Giới thiệu các thiết bị vào ra sử dụng trong đề tài	8

2.3 Giới thiệu phần cứng	8
2.3.1 <i>Cảm biến</i>	8
2.3.2 <i>Module thời gian thực DS1307.....</i>	10
2.3.3 <i>Màn hình hiển thị LCD:</i>	16
2.3.4 <i>Mạch chuyển giao tiếp LCD 1602 và LCD 2004 sang I2C:</i>	17
2.3.5 <i>IC ULN2303.....</i>	18
2.3.6 <i>Module chuyển đổi nguồn điện AC – DC.....</i>	19
2.3.7 <i>Module đo điện năng PZEM – 004T</i>	20
2.3.8 <i>Vi điều khiển.....</i>	21
2.3.9 <i>Relay tiếp điểm cơ khí</i>	26
2.4 Các chuẩn truyền dữ liệu	27
2.4.1 <i>Chuẩn truyền thông UART.....</i>	27
2.4.2 <i>Chuẩn giao tiếp I2C</i>	28
2.5 Giới thiệu nền tảng Openhab.....	31
2.5.1 <i>OpenHAB.....</i>	31
2.5.2 <i>OpenHAB designer.....</i>	31
2.5.3 <i>Java</i>	31
2.5.4 <i>MQTT.....</i>	32
2.5.5 <i>Bảo mật của giao thức MQTT:.....</i>	36
2.5.6 <i>Mosquitto</i>	37
2.5.7 <i>MQTT lends</i>	38
CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG	39
3.1 Giới thiệu	39
3.2 Thiết kế hệ thống.....	39
3.2.1 <i>Thiết kế sơ đồ khối hệ thống.....</i>	39
3.2.2 <i>Sơ đồ nguyên lý toàn mạch</i>	40
3.3 Thi công hệ thống.....	42
3.3.1 <i>Thi công board mạch.....</i>	42

3.3.2 Lắp ráp và kiểm tra.....	46
3.3.3 Đóng gói và thi công mô hình.....	46
3.4 Lập trình hệ thống	46
3.4.1 Lưu đồ giải thuật.....	46
3.4.2 Phần mềm lập trình cho vi điều khiển	48
3.4.3 Phần mềm lập trình giao diện điều khiển	49
3.5 Kết quả - nhận xét - đánh giá.....	51
3.5.1 Giới thiệu	51
3.5.2 Kết quả đạt được.....	51
3.5.3 Kết quả thực nghiệm	52
3.5.4 Nhận xét – Đánh giá	57
KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	59
1. Kết luận	59
2. Hướng phát triển	59
TÀI LIỆU THAM KHẢO.....	60
1. Sách tham khảo	60
2. Bài báo, tạp chí tham khảo	60
3. Đồ án tốt nghiệp, luận án tham khảo	60
4. Tài liệu tham khảo từ Internet	60
PHỤ LỤC	61

DANH SÁCH CÁC BẢNG HÌNH VẼ

Bảng 2. 1: Giao tiếp giữa mạch chuyển đổi LCD sang I2C với MCU.....	18
Bảng 2. 2: Số liệu thực nghiệm	58
Hình 1. 1: Cuộc sống con người và công nghệ	2
Hình 1. 2: Nhà thông minh	3
Hình 2. 1: Ứng dụng của Internet Of Things	7
Hình 2. 2: Cảm biến DHT11	8
Hình 2. 4: Cảm biến khí gaz MQ2	9
Hình 2. 3: Cảm biến khí gaz MQ5	9
Hình 2. 5: Sơ đồ nguyên lý cảm biến điện dung TTP223-BA6	10
Hình 2. 6: Sơ đồ chân DS1307	11
Hình 2. 7: Thanh ghi DS1307.....	12
Hình 2. 8: Tổ chức thanh ghi DS1307.....	13
Hình 2. 9: Module DS1307	15
Hình 2. 10: LCD text 1604 xanh dương.....	16
Hình 2. 12: LCD text 2004 xanh dương.....	16
Hình 2. 13: Mạch chuyển giao tiếp LCD 1602 và LCD 2004 sang I2C	17
Hình 2. 14: Cấu tạo của IC đệm dòng ULN2803.....	18
Hình 2. 15: Sơ đồ nguyên lý 1 kênh của ULN2803	19
Hình 2. 16: Module Hi-Link chuyển đổi nguồn AC-DC	19
Hình 2. 17: Module PZEM - 004T	20
Hình 2. 18: Sơ đồ nối dây để sử dụng module PZEM 004T	21
Hình 2. 19: Sơ đồ nguyên lý ESP8266.....	23
Hình 2. 20: Hình ảnh ESP-01	24
Hình 2. 21: Hình ảnh ESP-07	24
Hình 2. 22: Hình ảnh ESP-12	25
Hình 2. 23: ESP8266 NodeMCU	26
Hình 2. 24: Relay 5V.....	27
Hình 2. 25: Truyền dữ liệu qua lại giữa 2 vi điều khiển và giữa vi điều khiển với PC	28
Hình 2. 26: Bus I2C và các thiết bị ngoại vi	29
Hình 2. 27: Bus I2C và các thiết bị ngoại vi	29
Hình 2. 28: Điều kiện start stop.....	30
Hình 2. 29: Truyền dữ liệu I2C	31

Hình 2. 30: Mô hình Publish/Subscribe	32
Hình 2. 31: Kết nối MQTT.....	32
Hình 2. 32: Publish	33
Hình 2. 33:Mô hình kết nối trong MQTT.....	34
Hình 2. 34: Giao tiếp MQTT client và MQTT Broker.....	34
Hình 2. 35:Subscribe và Suback	35
Hình 2. 36: Unsubscribe và Unsuback	35
Hình 2. 37: Thiết lập và liên kết với MQTT	35
Hình 2. 38: Minh họa SSL/TLS handshake	37
Hình 3. 1: Sơ đồ khối hệ thống.....	39
Hình 3. 2: Sơ đồ nguyên lý mạch giám sát năng lượng	40
Hình 3. 3: Sơ đồ nguyên lý mạch điều khiển thiết bị	41
Hình 3. 4: Sơ đồ nguyên lý mạch cảm biến.....	41
Hình 3. 5: Bố trí linh kiện mặt trên cho mạch giám sát năng lượng	42
Hình 3. 6: Mạch pcb lớp bottom của mạch giám sát năng lượng.....	43
Hình 3. 7: Bố trí linh kiện mặt trên cho mạch điều khiển thiết bị.....	43
Hình 3. 8: Pcb lớp bottom của mạch điều khiển thiết bị	44
Hình 3. 9: Bố trí linh kiện mạch cảm biến	45
Hình 3. 10: PCB lớp bottom mạch cảm biến.....	45
Hình 3. 11: Lưu đồ giải thuật hệ thống 1	47
Hình 3. 12: Lưu đồ giải thuật hệ thống 2	48
Hình 3. 13: Tạo File mới trong Visual Studio Code	49
Hình 3. 14: Ngôn ngữ lập trình trên Visual Studio Code.....	50
Hình 3. 15: Code trên Visual Studio Code.....	50
Hình 3. 16: Mô hình sản phẩm khi hoàn thành	52
Hình 3. 17: Module điều khiển thiết bị.....	53
Hình 3. 18: Module giám sát nhiệt độ, độ ẩm	54
Hình 3. 19: Module giám sát năng lượng.....	54
Hình 3. 20: Trang chủ Web giám sát điều khiển.....	55
Hình 3. 21: Giao diện trang giám sát điều khiển thiết bị.....	55
Hình 3. 22: Giao diện giám sát năng lượng.....	56
Hình 3. 23: Giao diện giám sát các chỉ số cảm biến	56
Hình 3. 24: Giao diện app điện thoại.....	57

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

STT	Ký hiệu	Nguyên nghĩa
1	IoT	Internet of Things
2	DC	Direct Current
3	AC	Alternating Current
4	LCD	Liquid-Crystal Display
5	UART	Universal Asynchronous Receiver / Transmitter
6	SDA	Serial data line
7	SCL	Serial clock line
8	MQTT	Message Queuing Telemetry Transport
9	HTTP	Hyper Text Transfer Protocol
10	HTML	HyperText Markup Language
11	TCP/IP	Transmission Control Protocol IP Internet Protocol
12	VCC	Common collector)
13	In	Input
14	Out	Output
15	GND	Groud
16	IC	Integrated circuit,
17	V	Volt
18	A	Ampe
19	PWM	Pulse-width modulation
20	SPI	Serial Peripheral Interface
21	TLS	Transport Layer Security
22	SSL	Secure Sockets Layer

MỞ ĐẦU

Ngày nay trên thế giới với sự bùng nổ của các ngành công nghệ thông tin , điện tử,... Đã làm cho đời sống của con người ngày càng hoàn thiện.Các thiết bị tự động hóa đã ngày càng xâm lấn vào trong sản xuất và thậm chí là vào cuộc sống sinh hoạt hằng ngày của mỗi con người. Do đó một ngôi nhà thông minh không còn là mơ ước của con người nữa mà nó đã trở thành hiện thực hóa. Sự bùng nổ của các ngành công nghệ thông tin, điện tử là cơ sở quá trình phát triển công nghệ được đẩy nhanh làm cho đời sống khởi sắc. Cùng với sự phát triển đó thì nhu cầu về một cuộc sống không những đầy đủ mà còn thoải mái và tiện lợi của con người cũng ngày một gia tăng. Trong những năm gần đây, ngôi nhà thông minh đã xuất hiện và được ứng dụng rộng rãi.

Sự ra đời của các công nghệ giám sát, điều khiển và Internet of Things (IOT). Vì vậy nhóm em đã nghiên cứu về vấn đề nhà thông minh, nhóm đã chọn đề tài “**THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH**” với mục đích phát triển hệ thống điều khiển không dây trong ngôi nhà, dựa trên nền tảng mã nguồn mở Openhab và giao thức MQTT người dùng có thể thao tác điều khiển., giám sát, các thiết bị dùng trong gia đình thông qua một chiếc smatphone, tablet hoặc máy tính, ... một cách dễ dàng và thuận tiện đáp ứng nhu cầu tiện lợi cho người dùng. Đề tài trên không phải là quá mới, nhưng nó đủ đáp ứng được một lượng kiến thức về điện tử và được ứng dụng rất thực tế đối với đời sống hiện nay.

Đề tài này được thực hiện dựa trên sự hướng dẫn tận tình của thầy Trần Duy Chung, cũng như sự hỗ trợ về thiết bị của nhà trường và qua cá tài liệu có sẵn trước đó để thực hiện đề tài rất mong nhận được sự góp ý của hội đồng bảo vệ, thầy cô, và các bạn sinh viên để có thể nâng cao chất lượng của đề án

Nội dung thực hiện bao gồm: nghiên cứu tổng quan, cơ sở lý thuyết và các module liên quan, thiết kế hệ thống bao gồm phần cứng, phần mềm và sever để liên kết giữa phần cứng và phần mềm. Sau đó thực hiện và kiểm tra từng phần, kiểm tra tín hiệu giữa phần cứng và sever, phần mềm và sever rồi sau đó mới thực hiện kết nối toàn bộ thành một hệ thống điều khiển từ phần cứng sang phần mềm và ngược lại.

Trong khuôn khổ đề tài này, nhóm sẽ đề xuất ra các phương án kỹ thuật để thiết kế hệ thống giám sát và điều khiển các thiết bị điện trong gia đình. Tuy nhiên với kiến thức được học hỏi và tìm hiểu còn nhiều mặt hạn chế, những phương án mà nhóm đưa ra cũng có thể chưa phải là những phương án tối ưu nhất đối với hệ thống và cũng khó có thể mà tránh khỏi những thiếu sót, nên nhóm rất mong nhận được những đóng góp và ý kiến đánh giá từ quý thầy cô, bạn bè và các công ty doanh nghiệp từ đó nhóm có thể hoàn thiện sản phẩm một cách tối ưu và tốt hơn.

CHƯƠNG 1: TỔNG QUAN

1.1 Đặt vấn đề

Ngày nay, trên thế giới sự bùng nổ của các ngành công nghệ thông tin, điện tử đã làm cho đời sống của con người ngày càng hoàn thiện. Các thiết bị thông minh đã ngày càng được ứng dụng vào cuộc sống sinh hoạt hằng ngày của mỗi con người. Đặc biệt là Smartphone đã trở thành vật gắn liền với mỗi cá nhân.



Hình 1. 1: Cuộc sống con người và công nghệ

Đề tài “**THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH**” để nâng cao chất lượng cuộc sống con người và đáp ứng các nhu cầu ngày càng mạnh mẽ trong thời đại công nghệ số.

Với sự phát triển ngày càng mạnh mẽ của khoa học công nghệ, vi điều khiển ngày càng thông dụng và hoàn thiện hơn, Sự xuất hiện của vi điều khiển đã hỗ trợ cho con người rất nhiều trong lập trình và thiết kế, nhất là đối với những người bắt đầu tìm tòi về vi điều khiển mà không có quá nhiều kiến thức, hiểu biết sâu sắc về vật lý và điện tử. Phần cứng của thiết bị đã được tích hợp nhiều chức năng cơ bản và là mã nguồn mở. Chính vì những lý do như vậy nên các dòng vi điều khiển hiện đang dần phổ biến và được phát triển ngày càng mạnh mẽ trên toàn thế giới.

Đặc biệt trong những thập niên gần đây cùng với sự phát triển của hệ thống thông minh, ngành tự động hóa đã phát triển và tạo ra bước ngoặt quan trọng trong lĩnh vực ngôi nhà thông minh phục vụ nhu cầu ngày càng cao của con người trong đời sống. Tại Việt Nam đã bắt đầu có nhiều công ty chuyên lắp đặt ngôi nhà hoặc hệ thống thông minh trong đó phải kể đến công ty BKAV.



Hình 1. 2: Nhà thông minh

Hãy nghĩ về những việc chúng ta phải làm mỗi ngày ở nhà như: Bật tắt bóng đèn, Tivi, hệ thống báo động, báo cháy, bật điều hòa, bật quạt, tưới nước tự động cho vườn cây, ... Ngoài ra ngôi nhà của chúng ta còn có thể giám sát được bằng Smartphone và gửi SMS về cho chủ nhà biết khi có người lạ đột nhập.

Xuất phát từ những thực tiễn nói trên, chúng em quyết định thực hiện đề tài cho đồ án tốt nghiệp của mình: **“THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH”**

1.2 Hướng giải quyết vấn đề

Có nhiều hướng giải quyết thiết kế bộ xử lý trung tâm để điều khiển ngôi nhà gồm có:

- Dùng IC rời
- Dùng vi điều khiển
- Dùng raspberry

Ở đây nhóm đã chọn hướng giải quyết là dùng vi điều khiển esp8266 phù hợp những tiêu chí của nhóm như là muốn tìm hiểu nghiên cứu về Arduino hoặc quan trọng là esp8266 có khả năng kết nối với Internet, Android với giá thành vừa phải không quá tầm tay như PLC, cũng như không phải thiết kế thêm mạch chuyển đổi RS232 để giao tiếp với máy tính như EPROM. Và vì đề tài chỉ được lắp đặt trên mô hình với kích thước nhỏ nên dùng ESP8266 là hợp lý nhất

1.3 Mục tiêu

Thiết kế hệ thống giám sát và điều khiển thiết bị trong nhà, điều khiển các thiết bị thông qua Openhab và phím nhấn cứng. Hệ thống điều khiển bằng hồng ngoại. Các thông số hiển thị như trạng thái cửa, trạng thái thiết bị, nhiệt độ, độ ẩm, độ ẩm đất, khí gas được hiển thị trên màn hình LCD và trên các giao diện App Android, IOS một cách trực quan.

Tìm hiểu giao thức MQTT truyền nhận dữ liệu của vi điều khiển ESP8266

1.4 Nội dung nghiên cứu

Đề tài “**THIẾT KẾ HỆ THỐNG ĐIỀU KHIỂN TỰ ĐỘNG SỬ DỤNG NỀN TẢNG OPENHAB CHO NGÔI NHÀ THÔNG MINH**” có các nội dung chính như sau:

- Tìm hiểu các hoạt động của các mô hình nhà thông minh.
- Thu thập dữ liệu quy trình thiết kế một ngôi nhà thông minh.
- Các giải pháp thiết kế hệ thống, mô hình nhà thông minh.
- Lựa chọn các thiết bị trong việc thiết kế mô hình nhà IoT (ESP8266, Module thời gian thực, relay đóng ngắt, bơm nước, màn hình hiển thị, các cảm biến như cảm biến nhiệt độ, độ ẩm, khí gas, PIR ...)
- Tìm hiểu các chuẩn truyền thông UART, I2C, MQTT
- Thiết kế giao diện điều khiển và giám sát: Web server, App android, Openhab. Thiết kế, thi công mạch.
- Thiết kế, thi công hệ thống điều khiển.
- Thiết kế, thi công mô hình ngôi nhà.
- Viết chương trình cho ESP8266.
- Lắp ráp hệ thống điều khiển vào mô hình và chạy thử nghiệm.
- Chỉnh sửa các lỗi xuất hiện.
- Đánh giá kết quả thực hiện.
- Viết báo cáo luận văn.
- Báo cáo đề tài tốt nghiệp.

1.5 Giới hạn đề tài

Trong phạm vi cho phép nhóm chỉ thi công ngôi nhà thông minh trên mô hình. Trong thời gian thực hiện đề tài là có hạn, với lượng kiến thức được truyền đạt trong suốt khóa học và khả năng có hạn, nhóm thực hiện đề tài chỉ giải quyết những vấn đề sau:

- Quản lý mức tiêu thụ năng lượng điện trong gia đình.
- Giám sát thiết bị điện, nhiệt độ, khí Gas trên Smartphone.
- Điều khiển thiết bị điện trong nhà qua, bật tắt hệ thống an ninh thông qua cảm biến hoặc smartphone.

1.6 Bố cục

Chương 1. Tổng Quan

Chương này trình bày đặt vấn đề dẫn nhập lý do chọn đề tài, mục tiêu, nội dung nghiên cứu, các giới hạn thông số và bố cục đề án.

Chương 2: Cơ Sở Lý Thuyết.

Chương này trình bày các lý thuyết có liên quan đến các vấn đề mà đề tài sẽ dùng để thực hiện thiết kế, thi công cho đề tài.

Chương 3: Thiết Kế và Thi Công Hệ Thống

Chương này giới thiệu tổng quan về các yêu cầu của đề tài về thiết kế và các liên quan đến đề tài. kết quả thi công phần cứng và những kết quả hiển thị trên màn hình hay giao diện điện thoại, đưa ra nhận xét và đánh giá sản phẩm mô hình đã hoàn thành.

Kết Luận và Hướng Phát Triển

Chương này trình bày ngắn gọn những kết quả đã thu được dựa vào những phương pháp, thuật toán đã kiến nghị ban đầu

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT VÀ CÔNG NGHỆ

2.1 Tổng quan về công nghệ IOT

2.1.1 Công nghệ IOT là gì?

Internet of Things (IoT) là một kịch bản của thế giới, khi mà mỗi đồ vật, mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet.

Theo khái niệm ta có thể hiểu Internet of Thing (IoT) là mọi vật đều có thể kết nối với nhau qua Internet, người dùng có thể kiểm soát đồ vật của mình qua một thiết bị thông minh như laptop, table PC hay smartphone.

Internet of Things ra đời đang dần cụ thể hóa các bộ phim khoa học viễn tưởng thành hiện thực trong đời sống.

2.1.2 Internet of Things - Xu hướng công nghệ tương lai?

Mặc dù khái niệm Internet of Things được đưa ra từ lâu. Nhưng trong những năm gần đây nó mới được nhiều doanh nghiệp cũng như các nhà khoa học để ý và phát triển mạnh mẽ. Trong các năm gần đây tại các triển lãm công nghệ CES, triển lãm di động toàn cầu... các hãng sản xuất lớn thay nhau đưa ra các thiết bị thông minh: tivi thông minh, tủ lạnh thông minh và ý tưởng về nhà thông minh... liên tục được giới thiệu.

Và khi gây được sự chú ý của cộng đồng, IoT đã cho thấy tiềm năng của mình bằng những con số đáng kinh ngạc.

2.1.3 Xu hướng và Tính chất của The Internet Of Things Thông minh

Sự thông minh và tự động trong điều khiển thực chất không phải là một phần trong ý tưởng về IoT. Các máy móc có thể dễ dàng nhận biết và phản hồi lại môi trường xung quanh (ambient intelligence), chúng cũng có thể tự điều khiển bản thân (autonomous control) mà không cần đến kết nối mạng. Tuy nhiên, trong thời gian gần đây người ta bắt đầu nghiên cứu kết hợp hai khái niệm IoT và autonomous control lại với nhau. Tương lai của IoT có thể là một mạng lưới các thực thể thông minh có khả năng tự tổ chức và hoạt động riêng lẻ tùy theo tình huống, môi trường, đồng thời chúng cũng có thể liên lạc với nhau để trao đổi thông tin, dữ liệu.

Việc tích hợp trí thông minh vào IoT còn có thể giúp các thiết bị, máy móc, phần mềm thu thập và phân tích các dấu vết điện tử của con người khi chúng ta tương tác với những thứ thông minh, từ đó phát hiện ra các tri thức mới liên quan tới cuộc sống, môi trường, các mối tương tác xã hội cũng như hành vi con người.

2.1.4 Kiến trúc dựa trên sự kiện

Các thực thể, máy móc trong IoT sẽ phản hồi dựa theo các sự kiện diễn ra trong lúc chúng hoạt động theo thời gian thực. Một số nhà nghiên cứu từng nói rằng một mạng lưới các sensor chính là một thành phần đơn giản của IoT.

2.1.5 Là một hệ thống phức tạp

Trong một thế giới mở, IoT sẽ mang tính chất phức tạp bởi nó bao gồm một lượng lớn các đường liên kết giữa những thiết bị, máy móc, dịch vụ với nhau, ngoài ra còn bởi khả năng thêm vào các nhân tố mới.

2.1.6 Kích thước

Một mạng lưới IoT có thể chứa đến 50 đến 100 nghìn tỉ đối tượng được kết nối và mạng lưới này có thể theo dõi sự di chuyển của từng đối tượng. Một con người sống trong thành thị có thể bị bao bọc xung quanh bởi 1000 đến 5000 đối tượng có khả năng theo dõi.

2.1.7 Vấn đề không gian, thời gian

Trong IoT, vị trí địa lý chính xác của một vật nào đó là rất quan trọng. Hiện nay, Internet chủ yếu được sử dụng để quản lý thông tin được xử lý bởi con người. Do đó những thông tin như địa điểm, thời gian, không gian của đối tượng không mấy quan trọng bởi người xử lý thông tin có thể quyết định các thông tin này có cần thiết hay không, và nếu cần thì họ có thể bổ sung thêm. Trong khi đó, IoT về lý thuyết sẽ thu thập rất nhiều dữ liệu, trong đó có thể có dữ liệu thừa về địa điểm, và việc xử lý dữ liệu đó được xem như không hiệu quả. Ngoài ra, việc xử lý một khối lượng lớn dữ liệu trong thời gian ngắn đủ để đáp ứng cho hoạt động của các đối tượng cũng là một thách thức hiện nay.

2.1.8 Ứng dụng của IoT



Hình 2. 1: Ứng dụng của Internet Of Things

IoT có ứng dụng rộng vô cùng, có thể kể ra một số như sau:

- Quản lý chất thải
- Quản lý và lập kế hoạch quản lý đô thị
- Quản lý môi trường

- Phản hồi trong các tình huống khẩn cấp
- Mua sắm thông minh
- Quản lí các thiết bị cá nhân

2.2 Giới thiệu các thiết bị vào ra sử dụng trong đề tài

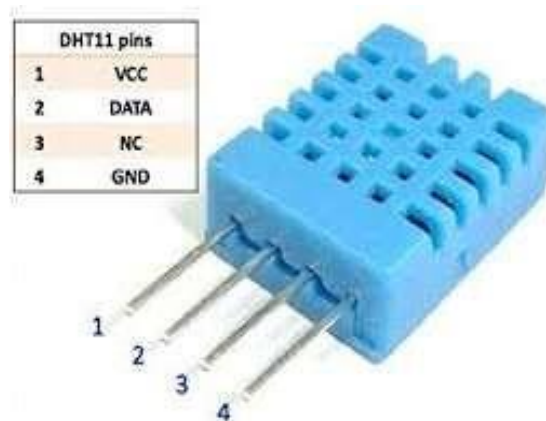
- Thiết bị đầu vào: Cảm biến nhiệt độ - độ ẩm DHT11, cảm biến khí gas, cảm biến hồng ngoại, nút nhấn, RFID, Module đo điện năng PZEM – 004T.
- Thiết bị đầu ra: thiết bị công suất: động cơ servo, bơm nước, động cơ DC, đèn 220V; thiết bị giao tiếp công suất: transistor, opto, relay; thiết bị hiển thị: LCD 16x2
- Thiết bị điều khiển trung tâm: ESP8266
- Các chuẩn truyền dữ liệu: UART, I2C, MQTT, HTTP
- Thiết bị giao diện điều khiển: điện thoại Android, Laptop, Web server, app android.

2.3 Giới thiệu phần cứng

2.3.1 Cảm biến

2.3.1.1 Cảm biến DHT11:

Cảm biến độ ẩm và nhiệt độ DHT11 là cảm biến rất thông dụng hiện nay vì chi phí rẻ và rất dễ lấy dữ liệu thông qua giao tiếp 1 wire. Bộ tiền xử lý tín hiệu tích hợp trong cảm biến giúp bạn có được dữ liệu chính xác mà không phải qua bất kỳ tính toán nào.



Hình 2. 2: Cảm biến DHT11

Thông tin kỹ thuật:

- Nguồn: 3 → 5 VDC.
- Dòng sử dụng: 2.5mA max (khi truyền dữ liệu).
- Đo tốt ở độ ẩm 20-80%RH với sai số 5%.
- Đo tốt ở nhiệt độ 0 → 50°C sai số ±2°C.
- Tần số lấy mẫu tối đa 1Hz (1 giây 1 lần)
- Kích thước 15mm x 12mm x 5.5mm.

- 4 chân, khoảng cách chân 0.1".

Lựa chọn cảm biến: Trong quá trình học cùng với thực tập, nhóm đã được biết cách sử dụng cảm biến nhiệt độ DS18B20 do vậy nên nhóm chọn cảm biến DS18B20 để thực hiện việc đọc nhiệt độ phòng. Để đọc nhiệt độ, độ ẩm cho vườn cây nhóm sử dụng cảm biến DHT11 do nó có chi phí rẻ và có thể đọc được nhiệt độ lẫn độ ẩm.

2.3.1.2 Cảm biến khí Gas:

Cảm biến khí Gas (LPG/CO/CH₄) MQ-2: MQ-2 sử dụng phần tử SnO₂ có độ dẫn điện thấp hơn trong không khí sạch, khi khí dễ cháy tồn tại, cảm biến có độ dẫn điện cao hơn, nồng độ chất dễ cháy càng cao thì độ dẫn điện của SnO₂ sẽ càng cao và được tương ứng chuyển đổi thành mức tín hiệu điện. MQ-2 là cảm biến khí có độ nhạy cao với LPG, Propane và Hydrogen, mê-tan (CH₄) và hơi dễ bắt lửa khác, với chi phí thấp và phù hợp cho các ứng dụng khác nhau. Cảm biến xuất ra cả hai dạng tín hiệu là Analog và Digital, tín hiệu **Cảm biến khí gas MQ5**: Sử dụng để đo khí gas trong môi trường. Cảm biến có độ nhạy cao, khả năng phản hồi nhanh, độ nhạy có thể điều chỉnh được bằng biến trở. Nguồn hoạt động 5V.



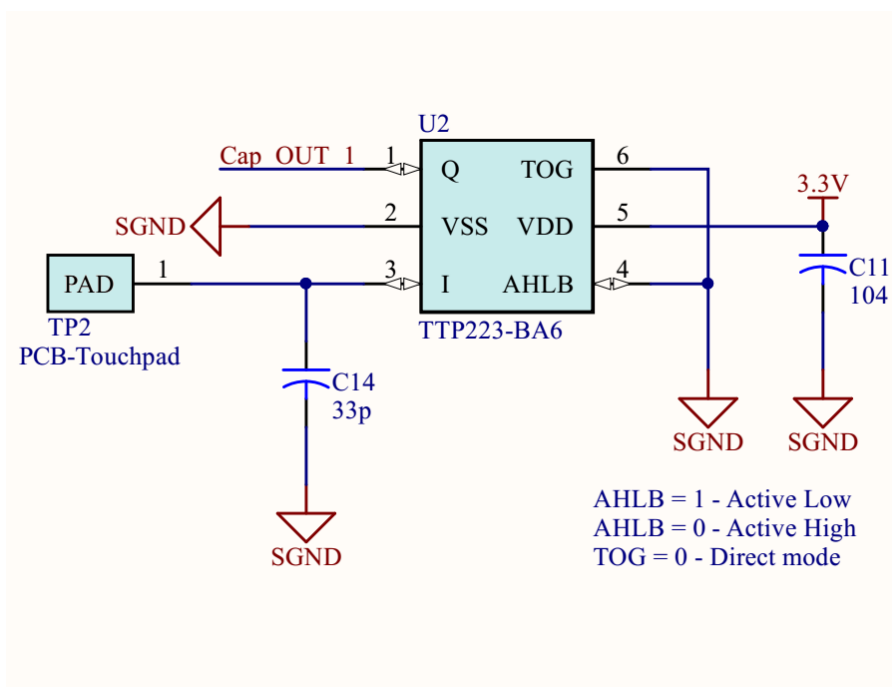
Hình 2. 4: Cảm biến khí gaz MQ5



Hình 2. 3: Cảm biến khí gaz MQ2

Lựa chọn cảm biến: Giữa hai loại cảm biến MQ2 và MQ5 thì MQ2 phổ biến, có giá thành rẻ hơn nên nhóm sử dụng cảm biến MQ2 để đọc khí gas

2.3.1.3 Cảm biến điện dung TTP223-BA6



Hình 2. 5: Sơ đồ nguyên lý cảm biến điện dung TTP223-BA6

Giới thiệu về IC cảm biến điện dung TTP223-BA6

TTP223 là một IC phát hiện chạm phím cảm ứng, việc phát hiện chạm cảm biến của IC được thiết kế để thay thế việc nhấn nút bấm trực tiếp thông thường, với kích thước cảm biến đa dạng, tiêu thụ điện năng thấp và điện áp hoạt động rộng là tiếp xúc tính năng chính cho các ứng dụng DC –AC.

Sơ đồ chân và chức năng

- Chân TOG nối GND chọn mode direct
- Chân AHLB là chân chọn mức điện áp đầu ra chân Q khi chạm phím, nối AHLB với GND chân Q ở mức active high khi chạm phím chân Q ở mức điện áp thấp.

Mục đích sử dụng IC TTP223-BA6

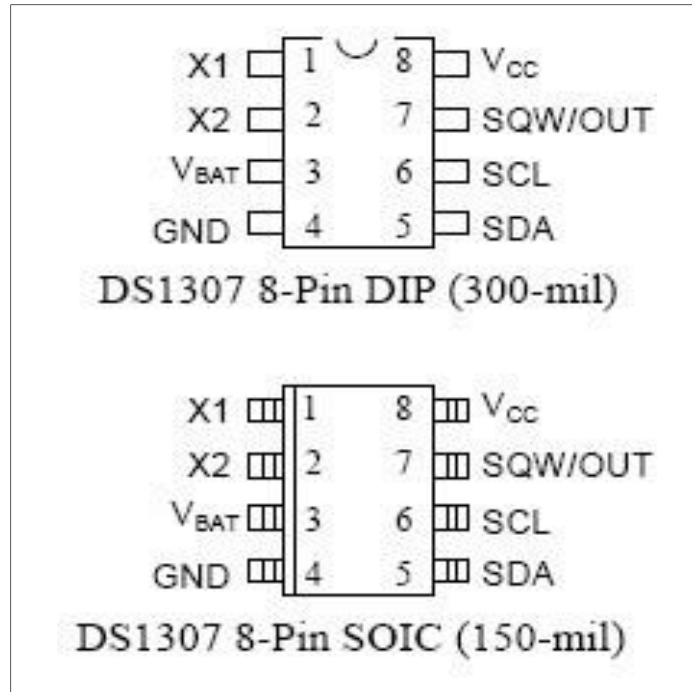
Sử dụng IC TTP223-BA6 để thiết kế mạch cảm biến điện dung 3 kênh dùng để bật tắt thiết bị và điều chỉnh công suất hoạt động của thiết bị tiêu thụ trong đề tài Smart home.

2.3.2 Module thời gian thực DS1307

2.3.2.1 Giới thiệu DS1307

DS1307 là chip thời gian thực hay RTC (Real time clock). Đây là một IC tích hợp cho thời gian bởi vì tính chính xác về thời gian tuyệt đối cho thời gian: Thứ, ngày, tháng, năm, giờ, phút, giây. DS1307 là chế tạo bởi Dallas. Chip này có 7 thanh ghi 8 bit mỗi thanh ghi này chứa: thứ, ngày, tháng, năm, giờ, phút, giây. Ngoài ra DS1307 còn

chứa 1 thanh ghi điều khiển ngõ ra phụ và 56 thanh ghi trống các tshanh ghi này có thể dùng như là RAM. DS1307 được đọc thông qua chuẩn truyền thông I2C nên do đó dễ đọc được và ghi từ DS1307 thông qua chuẩn truyền thông này. Do nó được giao tiếp chuẩn I2C nên cấu tạo bên ngoài nó rất đơn giản.



Hình 2. 6: Sơ đồ chân DS1307

Trên là hai dạng cấu tạo của DS1307. Chip này có 8 chân và chúng ta hay dùng là dạng Dip và các chân nó được mô tả như sau:

- **X1 và X2** là đầu vào dao động cho DS1307. Cần giao động thạch anh 32.768 Khz.
- **Vbat** là nguồn nuôi cho chip. Nguồn này từ (2V- 3.5V) ta lấy pin có nguồn 3V. Đây là nguồn cho chip hoạt động liên tục khi không có nguồn Vcc mà DS1307 vẫn hoạt động theo thời gian.
- **Vcc** là nguồn cho giao tiếp I2C. Điện áp cung cấp là 5V chuẩn và được dùng chung với vi xử lý. Nếu Vcc không có mà Vbat có thì DS1307 vẫn hoạt động bình thường nhưng mà không ghi và đọc được dữ liệu.
- **GND** là nguồn mass chung cho cả Vcc và Vbat.
- **SQW/OUT** là một ngõ ra phụ tạo xung dao động (xung vuông). Chân này không ảnh hưởng đến thời gian thực nên chúng ta không sử dụng chân này trong thời gian thực và bỏ trống chân này.
- **SCL và SDA** là hai bus dữ liệu của DS1307. Thông tin truyền và ghi điều

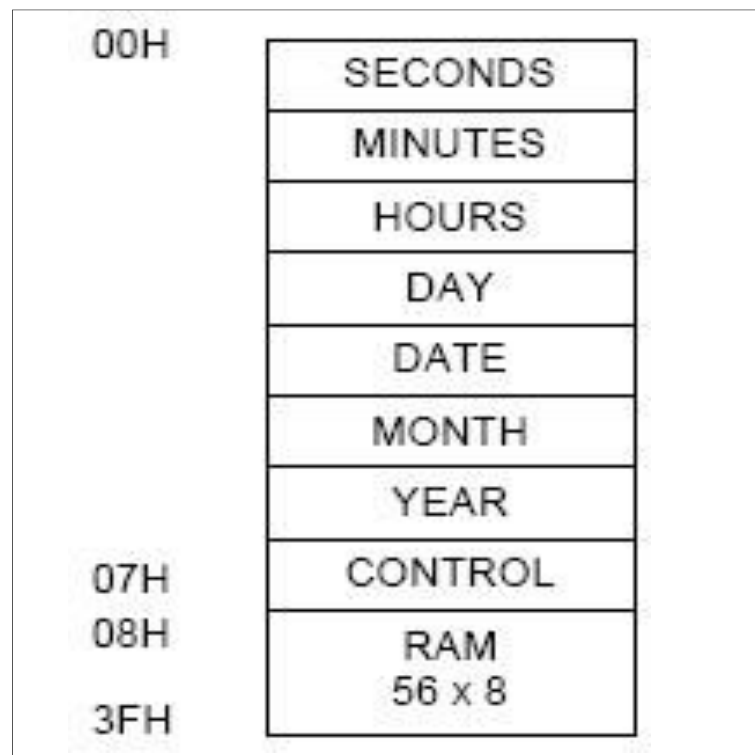
được truyền qua 2 đường truyền này theo chuẩn I2C.

2.3.2.2 Tổ chức thanh ghi trong DS1307

Cấu tạo bên trong của DS1307 bao gồm mạch nguồn, dao động, logic và con trỏ, thanh ghi thực hiện việc ghi đọc. Do trong các bài toán chúng ta thường sử dụng DS1307 cho đồng hồ thời gian thực nên do đó chúng ta chỉ quan tâm đến việc ghi đọc các thanh ghi cần thiết (sec, min, hour...) thông qua chuẩn truyền thông I2C.

Trong bộ nhớ của DS1307 có tất cả 64 thanh ghi địa chỉ từ 0 đến 63 và được bắt đầu từ 0x00 đến 0x3F nhưng trong đó chỉ có 8 thanh ghi đầu là thanh ghi thời gian thực.

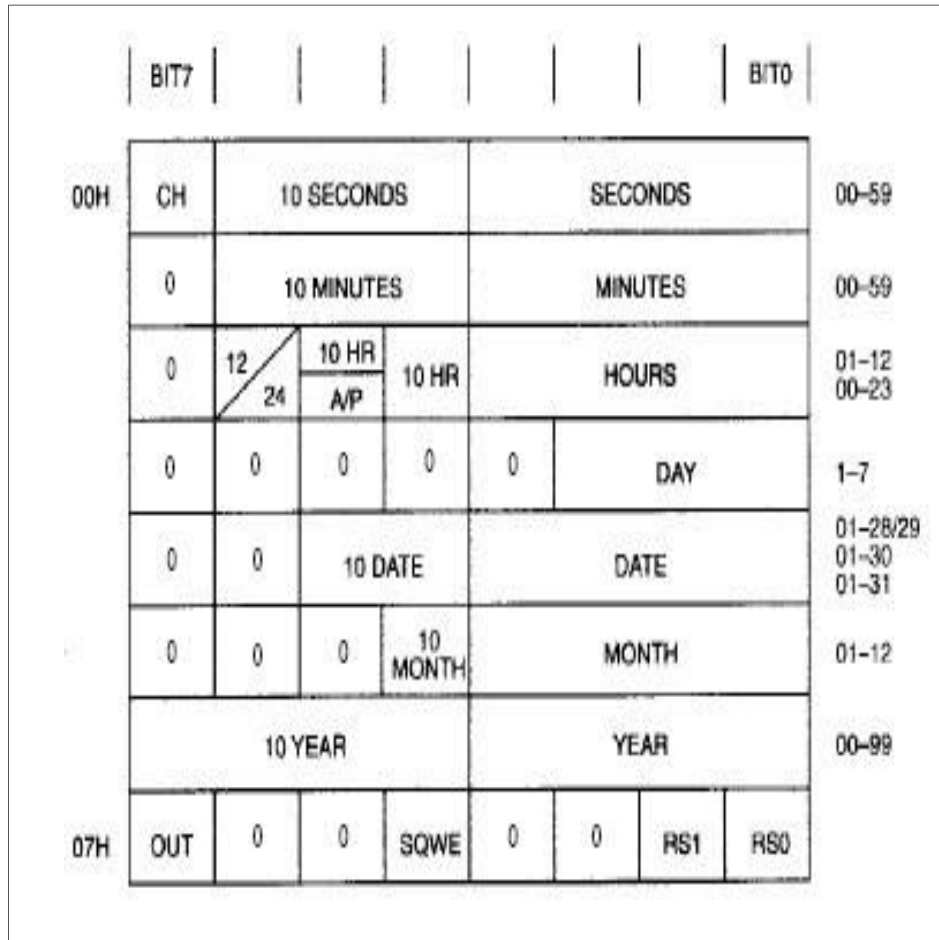
Các thanh ghi thời gian thực nó được sắp xếp theo thứ tự: giây, phút, giờ, thứ,



Hình 2. 7: Thanh ghi DS1307

ngày, tháng, năm và bắt đầu từ thanh ghi giây (0x00) và kết thúc bằng thanh ghi năm (0x06). Riêng thanh ghi Control dùng để điều khiển ngõ ra của chân SQW/OUT nên trong thực tế nên không mấy ai sử dụng thanh ghi này trong thời gian thực nên chúng ta bỏ qua thanh ghi này.

Do 7 thanh ghi đầu tiên là khá quan trọng cho thời gian thực và là thanh ghi quan trọng nhất trong con DS1307 nên chúng ta phải hiểu được cách tổ chức thanh ghi này trong DS1307.



Hình 2. 8: Tổ chức thanh ghi DS1307

Nhìn bảng trên chúng ta thấy các thanh ghi được mã hóa theo bit. Mỗi bit trong thanh ghi đều có chức năng riêng.

Thanh ghi giây (0x00): Đây là thanh ghi giây của DS1307. Nhìn hình trên ta thấy được từ bit 0 đến bit 3 là dùng để mã hóa số BCD hàng đơn vị của giây. Tiếp theo, từ bit 4 đến bit 6 dùng để mã hóa BCD hàng chục của giây. Tại sao nó chỉ sử dụng có 3 bit này là do giây của chúng ta lớn nhất chỉ đến 59 nên hàng chục lớn nhất là 5 nên chỉ cần 3 thanh ghi này là đủ mã hóa. Còn bit thứ 7 có tên là “CH” là “Clock Halt – Treo đồng hồ”, do đó nếu bit 7 này được đưa lên 1 tức là khóa đồng hồ nên do đó nó vô hiệu hóa chip và chip không hoạt động. Nên do vậy lúc nào cũng phải cho bit 7 này luôn xuống 0 từ lúc đầu (cái này sử dụng lệnh

and với 0x7F).

Thanh ghi phút (0x01): Đây là thanh ghi phút của DS1307. Cũng nhìn trên hình thanh ghi này được tổ chức như thanh ghi giây. Cũng là 3 bit thấp dùng để mã hóa BCD chữ số hàng đơn vị và số hàng chục chỉ lớn nhất là 5 nên do đó chỉ cần dùng từ bit 4 đến bit 6 để mã hóa BCD tiếp chữ số hàng chục. Nhưng thanh ghi này có sự khác biệt với thanh ghi giây là bit 7 nó đã mặc định bằng 0 rồi nên do đó chúng ta không phải làm gì với bit 7.

Thanh ghi giờ (0x02): Đây là thanh ghi giờ của DS1307, thanh ghi này được coi là phức tạp nhất nhưng mà nhìn bảng thì thấy các tổ chức của nó cũng hợp lý. Trước tiên chúng ta thấy được rằng từ bit 0 đến bit 3 nó dùng để mã hóa BCD của chữ số hàng đơn vị của giờ. Giờ còn có chế độ 24h và 12h nên do đó nó phức tạp ở các bit cao (bit 4 đến bit 7) và sự chọn chế độ 12h và 24h nó lại nằm ở bit 6. Nếu bit 6 = 0 thì ở chế độ 24h thì do chữ số hàng chục lớn nhất là 2 nên do đó nó chỉ dùng 2 bit (bit 4 và bit 5) để mã hóa BCD chữ số hàng chục của giờ. Nếu bit 6 = 1 thì chế độ 12h được chọn nhưng do chữ số của hàng chục của giờ trong chế độ này chỉ lớn nhất là 1 nên do đó bit thứ 4 là đủ để mã hóa BCD chữ số hàng chục của giờ rồi nhưng mà bit thứ 5 nó lại dùng để chỉ buổi sáng hay chiều, nếu mà bit 5 = 0 là AM và bit 5 = 1 là PM. Trong cả 2 chế độ 12h và 24h thì bit 7 = 0 nên ta ko cần chú ý đến thanh ghi này.

Thanh ghi thứ (0x03): Đây là thanh ghi thứ trong tuần của DS1307 và thanh ghi này khá là đơn giản trong DS1307. Nó dùng số để chỉ thứ trong tuần nên do đó nó chỉ lấy từ 1 đến 7 tương đương từ thứ hai đến chủ nhật. Nên do đó nó dùng 3 bit thấp (bit 0 đến bit 2) để mã hóa BCD ra thứ trong ngày. Còn các bit từ 3 đến 7 thì nó mặc định bằng 0 và ta không làm gì với các bit này.

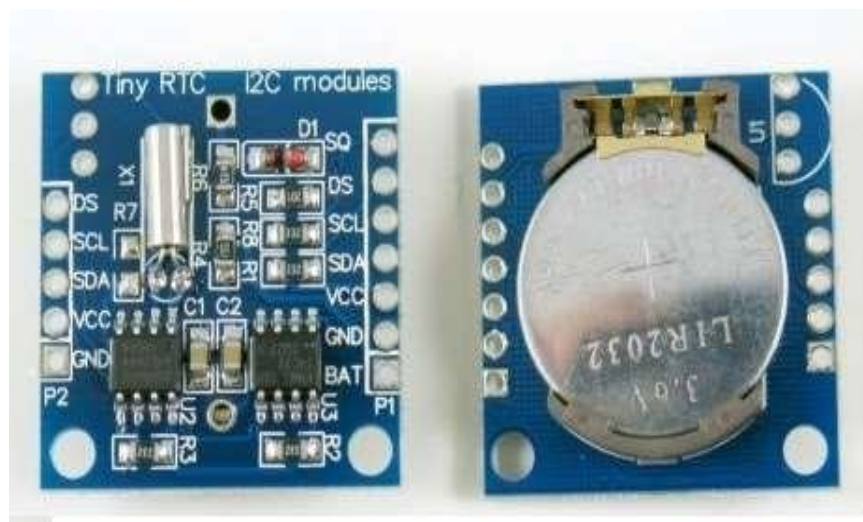
Thanh ghi ngày (0x04): Đây là thanh ghi ngày trong tháng của DS1307. Do trong các tháng có số ngày khác nhau nhưng mà nằm trong khoảng từ 1 đến 31 ngày. Do đó thanh ghi này các bit được tổ chức khá là đơn giản. Nó dùng 4 bit thấp (bit 0 đến bit 3) dùng để mã hóa BCD ra chữ số hàng đơn vị của ngày trong tháng. Nhưng do chữ số hàng chục của ngày trong tháng chỉ lớn nhất là 3 nên chỉ dùng bit 4 và bit 5 là đủ mã hóa BCD rồi. Còn bit 6 và bit 7 chúng ta không làm

gì và nó mặc định bằng 0.

Thanh ghi tháng (0x05): Đây là thanh ghi tháng trong năm của DS1307. Tháng trong năm chỉ có từ 1 đến 12 tháng nên việc tổ chức trong bit cũng tương tự như ngày trong tháng nên do cũng 4 bit thấp (từ bit 0 đến bit 3) mã hóa BCD hàng đơn vị của tháng. Nhưng do hàng chục chỉ lớn nhất là 1 nên chỉ dùng 1 bit thứ 4 để mã hóa BCD ra chữ số hàng chục và các bit còn lại từ bit 5 đến bit 7 thì bỏ trống và nó mặc định cho xuống mức 0.

Thanh ghi năm (0x06): Đây là thanh ghi năm trong DS1307. DS1307 chỉ có 100 năm tương đương với 00 đến 99 nên nó dùng tất cả các bit thấp và bit cao để mã hóa BCD ra năm.

Thanh ghi điều khiển (0x07): Đây là thanh ghi điều khiển quá trình ghi của DS1307 và Quá trình ghi phải được kết thúc bằng địa chỉ 0x93.



Hình 2. 9: Module DS1307

Thông tin kỹ thuật:

- Nguồn cung cấp: 5VDC.
- Khả năng lưu trữ 32K bit với EEPROM AT24C32.
- Sử dụng giao thức 2 dây I2C.
- Lưu trữ thông tin giờ phút giây AM/PM.
- Lịch lưu trữ chính xác lên đến năm 2100.
- Có pin đồng hồ lưu trữ thông tin.

- Có ngõ ra tần số 1Hz.
- Kích thước: 16 x 22 x 23mm

2.3.3 Màn hình hiển thị LCD:

2.3.3.1 LCD text 1604 xanh dương:



Hình 2. 10: LCD text 1604 xanh dương

Màn hình text LCD1602 xanh dương sử dụng driver HD44780, có khả năng hiển thị 2 dòng với mỗi dòng 16 ký tự, màn hình có độ bền cao, rất phổ biến.

Thông số kỹ thuật:

- Điện áp hoạt động là 5 V.
- Kích thước: 80 x 36 x 12.5 mm. □ Chữ trắng, nền xanh dương.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu.

2.3.3.2 LCD text 2004 xanh dương:



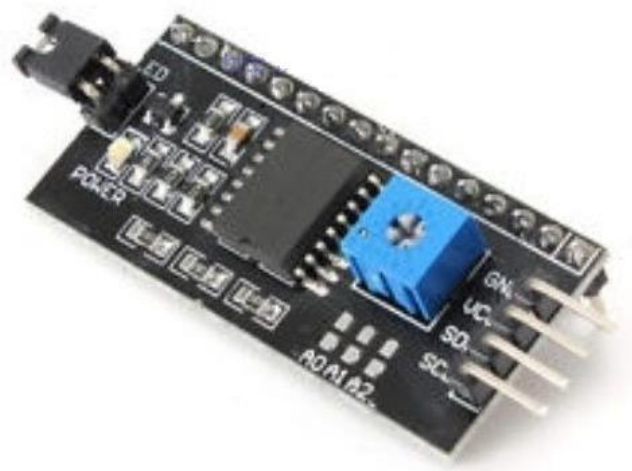
Hình 2. 11: LCD text 2004 xanh dương

Màn hình text LCD2004 xanh dương sử dụng driver HD44780, có khả năng hiển thị 4 dòng với mỗi dòng 20 ký tự, màn hình có độ bền cao, rất phổ biến.

Thông số kỹ thuật:

- Điện áp hoạt động là 5 V.
- Kích thước: 98 x 60 x 13.5 mm. □ Chữ trắng, nền xanh dương.
- Có đèn led nền, có thể dùng biến trở hoặc PWM điều chỉnh độ sáng để sử dụng ít điện năng hơn.
- Có thể được điều khiển với 6 dây tín hiệu.

2.3.4 Mạch chuyển giao tiếp LCD 1602 và LCD 2004 sang I2C:



Hình 2. 12: Mạch chuyển giao tiếp LCD 1602 và LCD 2004 sang I2C

Để sử dụng các loại LCD có driver là HD44780 (LCD 1602, LCD 2004, ...) cần có ít nhất 6 chân của MCU kết nối với các chân RS, EN, D7, D6, D5 và D4 để có thể giao tiếp với LCD. Nhưng với module chuyển giao tiếp LCD sang I2C, các bạn chỉ cần 2 chân (SDA và SCL) của MCU kết nối với 2 chân (SDA và SCL) của module là đã có thể hiển thị thông tin lên LCD. Ngoài ra có thể điều chỉnh được độ tương phản bởi biến trở gắn trên module.

Mạch chuyển đổi	MCU
-----------------	-----

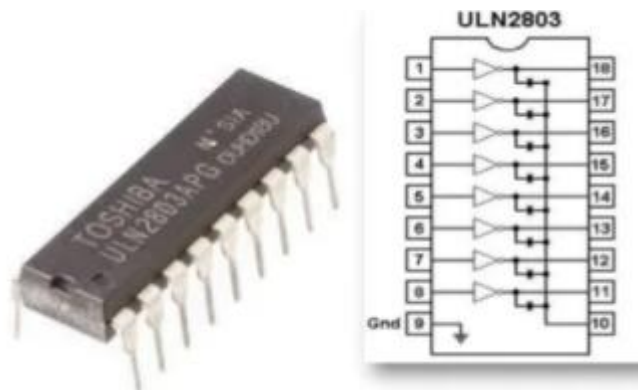
VCC	5V
GND	GND
SCL	SCL
SDA	SDA

Bảng 2. 1: Giao tiếp giữa mạch chuyển đổi LCD sang I2C với MCU

Ngoài ra mạch còn cho phép thay đổi địa chỉ I2C bằng cách nối các jump để có thể cùng lúc điều khiển nhiều LCD với chỉ một bộ I2C của MCU.

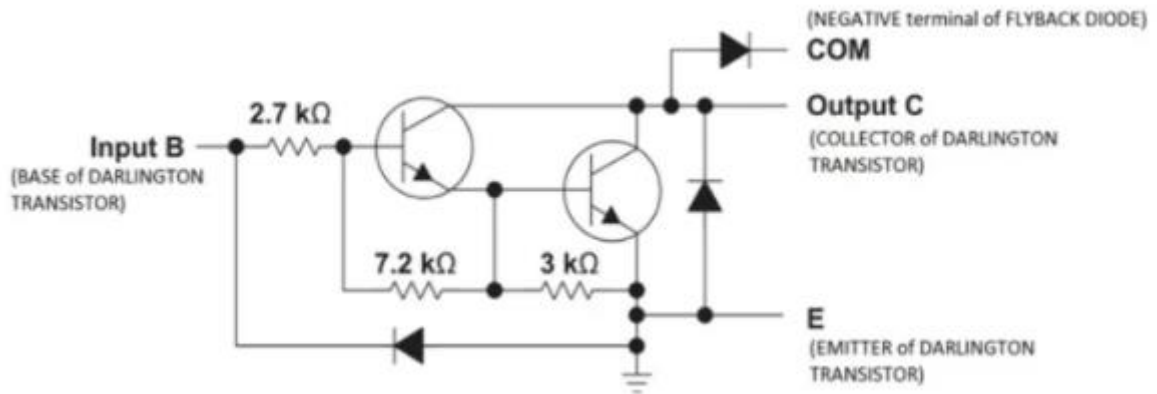
2.3.5 IC ULN2803

ULN2803 là một vi mạch đệm, bản chất cấu tạo là các mảng darlington chịu được dòng điện lớn và điện áp cao, trong đó có chứa 8 cặp transistor NPN ghép darlington cực góp hở với cực phát chung. Mỗi kênh của ULN2803 có một diode chặn có thể sử dụng trong trường hợp tải có tính năng cảm ứng, cụ thể ở đây là relay.



Hình 2. 13: Cấu tạo của IC đệm dòng ULN2803

ULN2803 có khả năng điều khiển 8 kênh riêng biệt, có thể nối trực tiếp với vi điều khiển 5V. Bên cạnh đó mỗi kênh của ULN2803 có khả năng chịu được dòng điện lớn trong một khoảng thời gian dài lên tới 500mA với biên độ đỉnh lên tới 600mA.



Hình 2. 14: Sơ đồ nguyên lý 1 kênh của ULN2803

Ứng dụng của ULN2803 được sử dụng trong các mạch đệm điều khiển động cơ một chiều, động cơ bước, khối hiển thị ma trận led, ...

Thông số kỹ thuật:

- Điện áp ra max: 50V (V_{ce}).
- Điện áp vào max: 30V (V_{in}).
- Dòng điện đầu ra liên tục: $I_c = 500\text{mA}$.
- Dòng điện đầu vào liên tục: $I_N = 25\text{mA}$.
- Công suất tiêu tán trên mỗi cặp darlington: 1W.
- Nhiệt độ làm việc: $-55 - 150^\circ\text{C}$.

Do ULN2803 có nhiệm vụ đệm dòng điều khiển cho Arduino vì thế Arduino sẽ mắc vào các chân I của ULN2803 khi đó các chân O của ULN sẽ được nối với các Relay để điều khiển các Relay 12v. nguyên nhân phải thông qua IC đệm dòng là Arduino không thể điều khiển các Relay 12v được vì Arduino không có ngõ ra 12v cấp cho các tải và dòng tối đa của chân 5v cũng chỉ đạt 0.5A vì thế không thể cấp cho toàn bộ Relay khi sử dụng.

2.3.6 Module chuyển đổi nguồn điện AC – DC



Hình 2. 15: Module Hi-Link chuyển đổi nguồn AC-DC

Trong đề tài cần sử dụng nguồn 220VAC cho các thiết bị cần giám sát được hoạt động, đồng thời cần nguồn có điện áp 5VDC cho các module LCD, NodeMCU ESP8266, PZEM004T. Nên để tiện lợi ta dùng một dây nguồn duy

nhất cho nguồn AC và chuyển từ nguồn 220VAC sang 5VDC thông qua module nguồn HLK-5M05 có ngõ ra 5VDC cấp cho các module và mạch điều khiển.

Thông số kỹ thuật:

- Điện áp vào: 100V ~ 240V (AC)
- Điện áp ngõ ra: 5VDC
- Công suất: 5W
- Độ gợn điện áp và nhiễu thấp
- Mạch bảo vệ quá tải và ngắn mạch
- Nhiệt độ hoạt động: -20°C ~ +80°C
- Khối lượng: <= 40g
- Độ ẩm tương đối: 5 ~ 95%
- Kích thước: 38x23x18mm

2.3.7 Module đo điện năng PZEM – 004T



Hình 2. 16: Module PZEM - 004T

Chức năng:

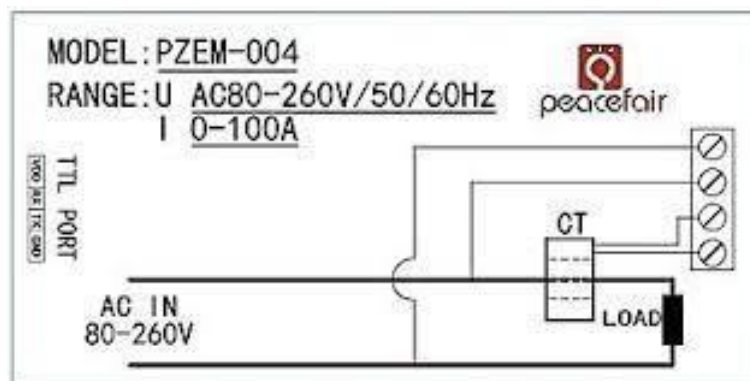
- Đo lường được các thông số điện như: điện áp, dòng điện, công suất hoạt động và điện năng tiêu thụ.
- Cài đặt ngưỡng báo động bằng phần mềm.
- Báo động quá tải (khi công suất tiêu thụ của thiết bị điện vượt quá giới hạn mà người sử dụng cài đặt thì module sẽ phát tín hiệu báo động quá tải).
- Lưu trữ dữ liệu về điện năng tiêu thụ khi tắt nguồn (lưu trữ năng lượng tích lũy trước khi tắt nguồn) và có thể reset điện năng tiêu thụ tích lũy bằng phím cứng trên module.
- Giao tiếp nối tiếp với vi điều khiển hoặc máy tính qua UART (9600/8N1).
- Module sử dụng phương pháp đo dòng điện gián tiếp thông qua cuộn dây CT.

- Ngoài ra, module còn có Opto quang cách ly giữa mạch đo điện năng và mạch nhận tín hiệu UART đảm bảo an toàn cho mạch điều khiển và người sử dụng.

Thông số kỹ thuật:

- Nhà sản xuất Peacefair. Sử dụng IC đo điện năng chuyên dụng của hãng SDIC.
- Điện áp đo và hoạt động: 80 ~ 260VAC / 50 - 60Hz, sai số 0.01
- Dòng điện đo và hoạt động: 0 ~ 100A, sai số 0.01
- Công suất đo và hoạt động: 0 ~ 26000W
- Năng lượng đo và hoạt động: 0~9999kWh.
- Giao tiếp UART mức logic TTL 5VDC baudrate mặc định 9600, 8, 1.
- Nhấn giữ nút Reset 5 giây để xóa các thông số về 0.
- Kích thước: 30 x 75 mm

Sơ đồ nối dây



Hình 2. 17: Sơ đồ nối dây để sử dụng module PZEM 004T

Giao tiếp:

Module sử dụng giao tiếp nối tiếp TTL qua UART với tốc độ Baudrate mặc định 9600bits/s, khung truyền gồm 8bits, 1bit stop và không có bit kiểm tra parity.

2.3.8 Vi điều khiển

Chip ESP8266 được phát triển bởi Espressif để cung cấp giải pháp giao tiếp Wi-Fi cho các thiết bị IoT. Điểm đặc biệt của dòng ESP8266 là nó được tích hợp các mạch RF như balun, antenna switches, TX power amplifier và RX filter ngay bên trong chip với kích thước rất nhỏ chỉ 5x5mm nên các board sử dụng ESP8266 không cần kích thước board lớn cũng như không cần nhiều linh kiện xung quanh. Ngoài ra, giá thành của ESP8266 cũng rất thấp đủ để hấp dẫn các nhà phát triển sản phẩm IoT.

2.3.8.1 Cấu trúc phần cứng của dòng chip ESP8266

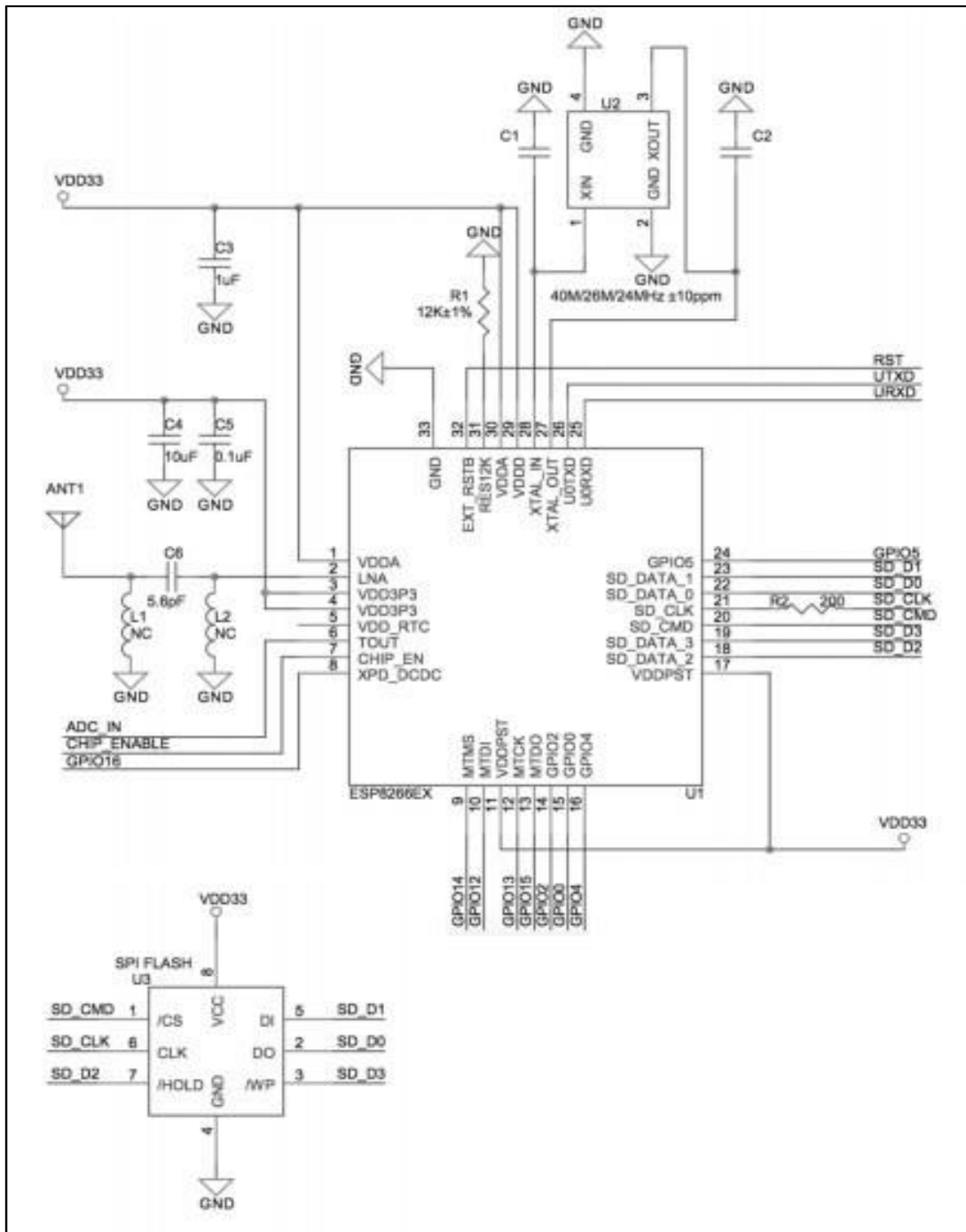
- Sử dụng 32-bit MCU core có tên là Tensilica
- Tốc độ system clock có thể set ở 80MHz hoặc 160MHz
- Không tích hợp bộ nhớ Flash để lưu chương trình
- Tích hợp 50KB RAM để lưu dữ liệu ứng dụng khi chạy
- Có đầy đủ các ngoại vi chuẩn để giao tiếp như 17 GPIO, 1 Slave SDIO, 3 SPI, 1 I2C, 1 I2S, 2 UART, 2 PWM.
- Tích hợp các mạch RF để truyền nhận dữ liệu ở tần số 2.4GHz
- Hỗ trợ các hoạt động truyền nhận các IP packages ở mức hardware như Acknowledgement, Fragmentation và Defragmentation, Aggregation, Frame, Encapsulation... (và phần stack TCP/IP sẽ được thực hiện trên firmware của ESP8266).
- Do không hỗ trợ bộ nhớ Flash nên các board sử dụng ESP8266 phải gắn thêm chip Flash bên ngoài và thường là Flash SPI để ESP8266 có thể đọc chương trình ứng dụng với chuẩn SDIO hoặc SPI.

2.3.8.2 Mạch nguyên lý đầy đủ của ESP8266

Chúng ta có thể thấy board ESP8266 chỉ cần thạch anh và SPI flash chip và vài linh kiện điện trở rất đơn giản. Do đó việc tích hợp giao tiếp Wi-Fi vào board ứng dụng với ESP8266 rất dễ dàng và nhanh chóng.

Mô hình lập trình ứng dụng với ESP8266 có thể chia làm 2 loại như sau:

- Sử dụng firmware được cung cấp bởi Espressif và giao tiếp thông qua AT commands.
- Lập trình firmware trực tiếp vào ESP8266 sử dụng bộ thư viện SDK cung cấp bởi Espressif.



Hình 2. 18: Sơ đồ nguyên lý ESP8266

2.3.8.3 Các loại module cho ESP8266 trên thị trường

Ngoại trừ module ESP-WROOM-02 được phát triển bởi chính Espressif cho mục đích nghiên cứu các tính năng của ESP8266, các module ứng dụng phổ biến hiện nay của ESP8266 đều được phát triển bởi công ty AI-Thinker.

Hiện tại có khá nhiều module khác nhau cho ESP8266 được sản xuất bởi công ty AI-Thinker. Đặc điểm khác nhau giữa các module này bao gồm:

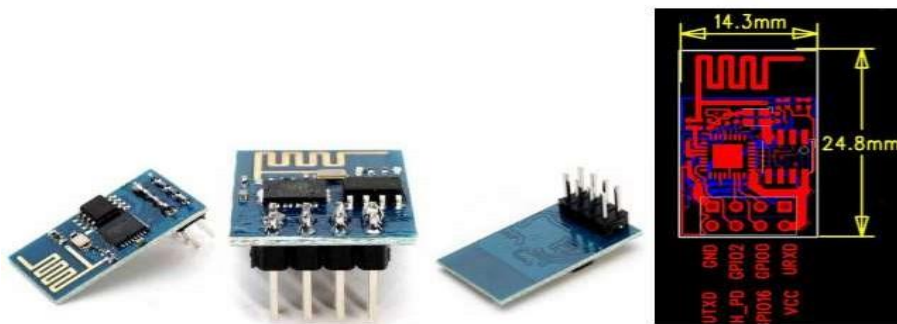
- Loại anten sử dụng (PCB anten, chip anten hoặc gắn anten ngoài)

- Dung lượng của chip Flash SPI trên board
- Kích thước board của module.
- Có gắn khung nhôm chống nhiễu hay không
- Số lượng pin GPIO đưa ra chân kết nối

Hiện tại AI-Thinker sản xuất 14 loại module cho ESP từ module ESP-01 đến ESP-

14. Ở thị trường Việt Nam thì các module là ESP-01, ESP-07 và ESP-12, ESP NodeMCU khá phổ biến.

ESP-01



Hình 2. 19: Hình ảnh ESP-01

- Sử dụng on-board PCB antenna.
- Có 2 LED trên board để báo nguồn và báo TX.
- Cung cấp 3 chân GPIO (GPIO0, GPIO2 và GPIO6) và 2 chân TXD/RXD cho UART.
- Dung lượng SPI Flash 4 MByte.
- Đưa chân ra jumper luôn nên có thể kết nối trực tiếp với các board khác 1 cách nhanh chóng.

ESP-07

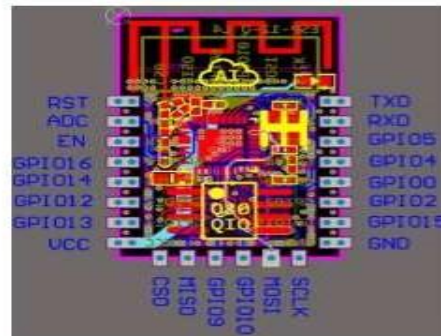
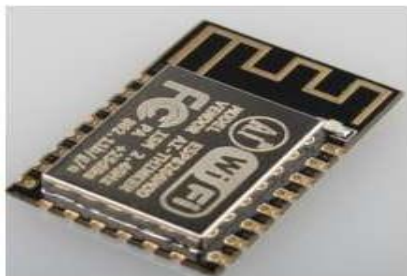


Hình 2. 20: Hình ảnh ESP-07

Sử dụng chip anten on-board và có IPEX connector hỗ trợ gắn thêm anten ngoài để tăng khoảng cách truyền.

- Có 2 LED trên board để báo nguồn và báo TX.
- Đưa ra 9 chân GPIO, 2 chân TX/RX cho UART, 1 chân REST để reset chip, 1 chân ADC, 1 chân CH_PD để đưa chip vào chế độ low power.
- Dung lượng SPI Flash trên board là 4MByte.
- Có thể hàn thêm jumper để kết nối trực tiếp với board khác hoặc hàn trực tiếp lên board ứng dụng.

ESP-12



Hình 2. 21: Hình ảnh ESP-12

- Sử dụng PCB anten on-board.
- Đưa ra 11 chân GPIO, 2 chân TX/RX cho UART, các chân cho SPI, chân RST để reset chip, 1 chân ADC.
- Dung lượng SPI Flash là 4MByte.
- Có thể hàn jumper để cắm dây vào các board khác hoặc hàn trực tiếp lên board ứng dụng.

ESP NodeMCU

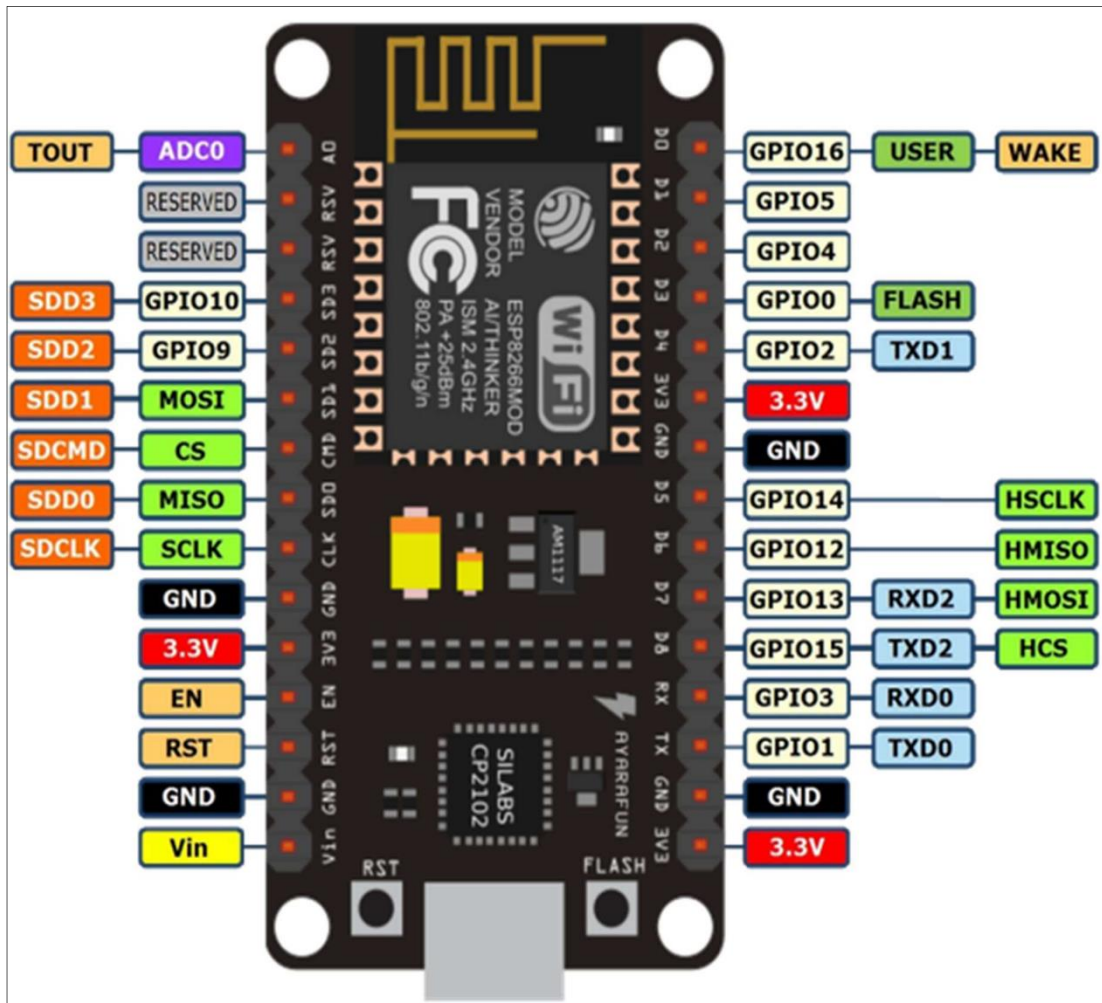
Kit RF thu phát Wifi ESP8266 NodeMCU là phiên bản NodeMCU sử dụng IC nạp giá rẻ CP2102 từ Lolin với bộ xử lý trung tâm là module Wifi SoC ESP8266, kit có thiết kế dễ sử dụng và đặc biệt là có thể sử dụng trực tiếp trình biên dịch của Arduino để lập trình và nạp code, điều này khiến việc sử dụng và lập trình các ứng dụng trên ESP8266 trở nên rất đơn giản.

Kit RF thu phát Wifi ESP8266 NodeMCU Lua V3 CH340 được dùng cho các ứng dụng cần kết nối, thu thập dữ liệu và điều khiển qua sóng Wifi, đặc biệt là các ứng dụng liên quan đến IoT.

Thông số kỹ thuật:

- IC chính: ESP8266 Wifi SoC.
- Phiên bản firmware: NodeMCU Lua.
- Chip nạp và giao tiếp UART: CH340.

- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cấp nguồn: 5VDC MicroUSB hoặc Vin.
- GIPO giao tiếp mức 3.3 VDC.
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 59 x 32mm.



Hình 2. 22: ESP8266 NodeMCU

2.3.9 Relay tiếp điểm cơ khí

Dùng điện áp 220VAC để cung cấp cho thiết bị công suất. Để cách ly tín hiệu điều khiển với phần thiết bị công suất có nhiều phương án thực hiện như relay tiếp điểm cơ khí, Solid State Relay, MOC + Triac...

Relay là một công tắc (khóa K) nhưng khác với công tắc thường là Relay được kích hoạt bằng điện. Trong mạch ta sử dụng relay với mục đích đóng ngắt các thiết bị điện theo yêu cầu.



Hình 2. 23: Relay 5V

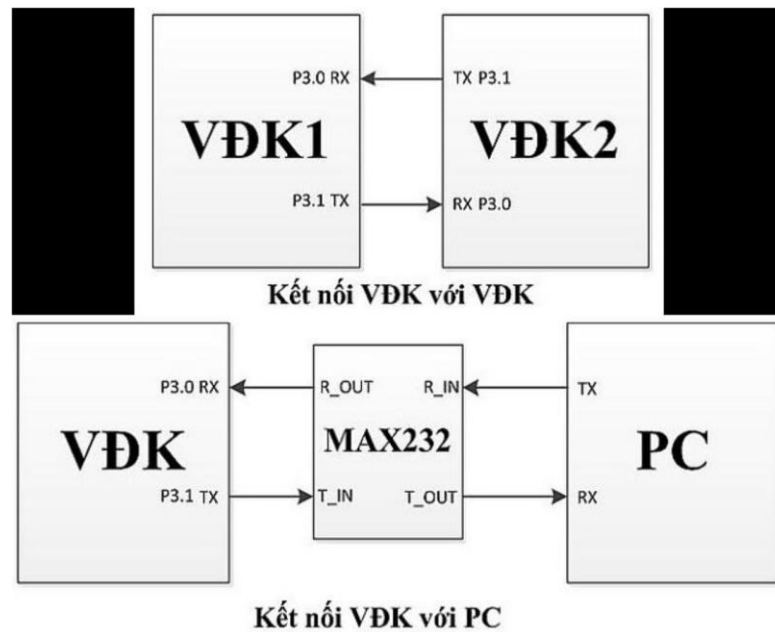
Thông số kĩ thuật:

- Điện áp điều khiển: 5V.
- Dòng tiêu thụ: 80mA.
- Dòng điện AC và điện áp AC cực đại: 10A → 250VAC.
- Thời gian tác động: 10ms.
- Thời gian nhả hãm: 5ms.
- Nhiệt độ hoạt động: -45 → 75 °C

2.4 Các chuẩn truyền dữ liệu

2.4.1 Chuẩn truyền thông UART

UART là viết tắt của Universal Asynchronous Receiver – Transmitter là kiểu truyền thông tin nối tiếp không đồng bộ thường là một mạch tích hợp. Mục đích của UART là để truyền tín hiệu qua lại lẫn nhau (ví dụ truyền tín hiệu từ Laptop vào Modem hay ngược lại) hay truyền từ vi điều khiển tới vi điều khiển, từ laptop tới vi điều khiển.



Hình 2. 24: Truyền dữ liệu qua lại giữa 2 vi điều khiển và giữa vi điều khiển với PC

Một số thông số:

Baud rate (tốc độ Baud): Khi truyền nhận không đồng bộ để hai module hiểu được nhau thì cần quy định một khoảng thời gian cho 1 bit truyền nhận, nghĩa là trước khi truyền thì tốc độ phải được cài đặt đầu tiên. Theo định nghĩa thì tốc độ baud là số bit truyền trong một giây.

Frame (khung truyền): Do kiểu truyền thông nối tiếp này rất dễ mất dữ liệu nên ngoài tốc độ, khung truyền cũng được cài đặt từ ban đầu để tránh bớt sự mất mát dữ liệu này. Khung truyền quy định số bit trong mỗi lần truyền, các bit báo như start, stop, các bit kiểm tra như parity và số bit trong một data.

Bit Start: Là bit bắt đầu trong khung truyền bit này nhằm mục đích báo cho thiết bị nhận biết quá trình truyền bắt đầu trên AVR bit Start có trạng thái là 0.

Data: Dữ liệu cần truyền Data không nhất thiết phải 8 bit có thể là 5, 6, 7, 8, 9.

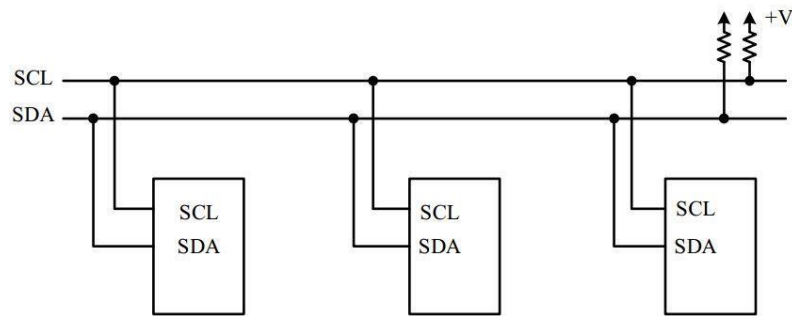
Trong UART bit LSB được truyền đi trước, Bit MSB được truyền đi sau.

Parity bit: Là bit kiểm tra dữ liệu đúng không, có 2 loại parity: chẵn (even parity), lẻ (old parity). Parity chẵn là bit parity thêm vào để số số 1 trong data + parity = chẵn parity lẻ là bit parity thêm vào để số số 1 trong data + parity = lẻ. Bit Parity là không bắt buộc nên có thể dùng hoặc không.

Stop: là bit báo cáo kết thúc khung truyền, thường là mức 5V. Và có thể có 1 hoặc 2 stop.

2.4.2 Chuẩn giao tiếp I2C

I2C là tên viết tắt của cụm từ Inter-Integrated Circuit. Đây là đường Bus giao tiếp giữa các IC với nhau. Bus I2C được sử dụng làm bus giao tiếp ngoại vi cho rất nhiều loại IC khác nhau như các loại Vi điều khiển 8051, PIC, AVR, ARM...



Hình 2. 25: Bus I2C và các thiết bị ngoại vi

Đặc điểm giao tiếp I2C

Một giao tiếp I2C gồm có 2 dây: Serial Data (SDA) và Serial Clock (SCL). SDA là đường truyền dữ liệu 2 hướng, còn SCL là đường truyền xung đồng hồ để đồng bộ và chỉ theo một hướng. Khi một thiết bị ngoại vi kết nối vào đường bus I2C thì chân SDA của nó sẽ nối với dây SDA của bus, chân SCL sẽ nối với dây SCL.

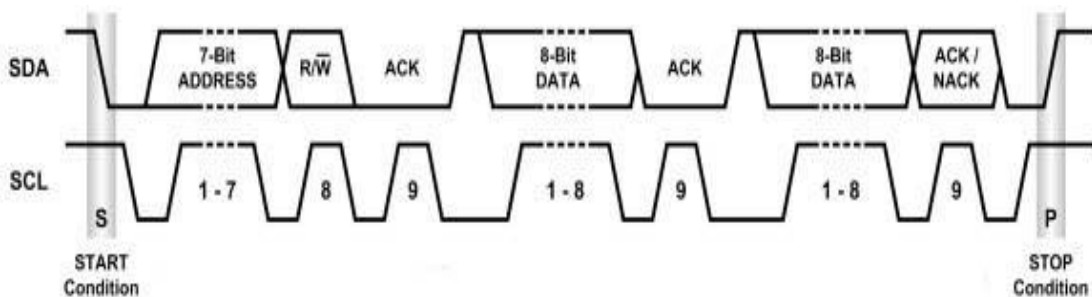
Mỗi dây SDA hay SCL đều được nối với điện áp dương của nguồn cấp thông qua một điện trở kéo lên (pullup resistor). Sự cần thiết của các điện trở kéo này là vì chân giao tiếp I2C của các thiết bị ngoại vi thường là dạng cực máng hở (opendrain hay opencollector). Giá trị của các điện trở này khác nhau tùy vào từng thiết bị và chuẩn giao tiếp, thường dao động trong khoảng 1k đến 4.7k.

Chế độ hoạt động (tốc độ truyền):

Các bus I2C có thể hoạt động ở ba chế độ, hay nói cách khác các dữ liệu trên bus I2C có thể được truyền trong ba chế độ khác nhau.

- Chế độ tiêu chuẩn (Standard mode)
- Chế độ nhanh (Fast mode)
- Chế độ cao tốc High-Speed (Hs) mode

Trình tự truyền bit trên đường truyền:



Hình 2. 26: Bus I2C và các thiết bị ngoại vi

- Thiết bị chủ tạo một điều kiện start. Điều kiện này thông báo cho tất cả các thiết bị tớ lắng nghe dữ liệu trên đường truyền.
- Thiết bị chủ gửi địa chỉ của thiết bị tớ mà thiết bị chủ muốn giao tiếp và chờ

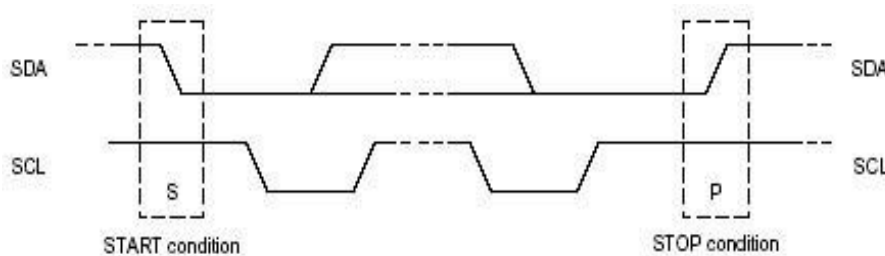
đọc/ghi dữ liệu (nếu cờ thiết lập lên 1 thì byte tiếp theo được truyền từ thiết bị tới đến thiết bị chủ, nếu cờ thiết lập xuống 0 thì byte tiếp theo truyền từ thiết bị chủ đến thiết bị tới).

- Khi thiết bị tới trên bus I2C có địa chỉ đúng với địa chỉ mà thiết bị chủ gửi sẽ phản hồi lại bằng một xung ACK.
- Giao tiếp giữa thiết bị chủ và tới trên bus dữ liệu bắt đầu. Cả chủ và tới đều có thể nhận hoặc truyền dữ liệu tùy thuộc vào việc truyền thông là đọc hay viết. Bộ truyền gửi 8 bit dữ liệu tới bộ nhận, bộ nhận trả lời với một bit ACK.
- Để kết thúc quá trình giao tiếp, thiết bị chủ tạo ra một điều kiện stop.

Điều kiện START và STOP (START and STOP conditions)

START và STOP là những điều kiện bắt buộc phải có khi một thiết bị chủ muốn thiết lập giao tiếp với một thiết bị nào đó trên bus I2C. START là điều kiện khởi đầu, báo hiệu bắt đầu của giao tiếp, còn STOP báo hiệu kết thúc một giao tiếp. Hình dưới đây mô tả điều kiện START và STOP.

Ban đầu khi chưa thực hiện quá trình giao tiếp, cả hai đường SDA và SCL đều ở mức cao (**SDA = SCL = HIGH**). Lúc này bus I2C được coi là rỗi (“bus free”), sẵn sàng cho một giao tiếp. Hai điều kiện START và STOP là không thể thiếu trong việc giao tiếp giữa các thiết bị I2C với nhau.



Hình 2. 27: Điều kiện start stop

Điều kiện START: một sự chuyển đổi trạng thái từ cao xuống thấp trên đường SDA trong khi đường SCL đang ở mức cao (cao = 1; thấp = 0) báo hiệu một điều kiện START

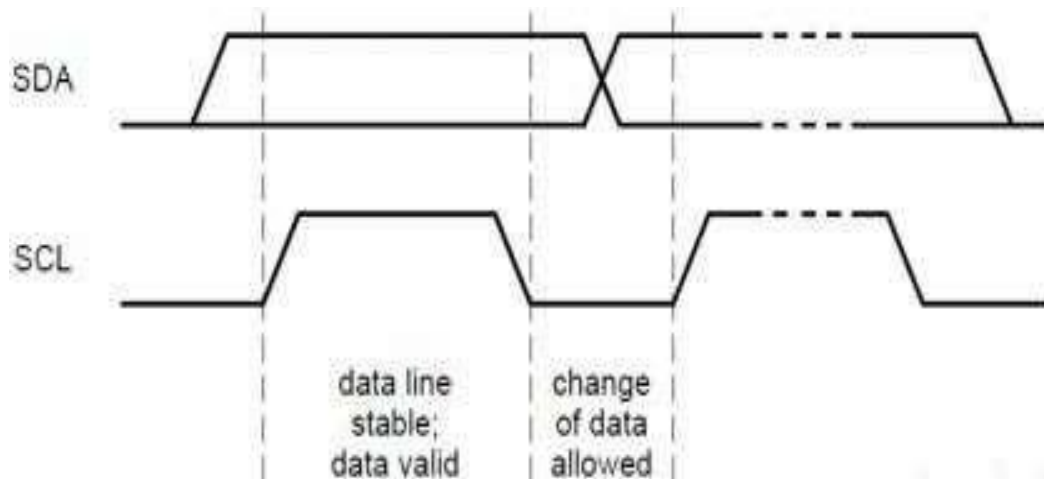
Điều kiện STOP: một sự chuyển đổi trạng thái từ mức thấp lên cao trên đường SDA trong khi đường SCL đang ở mức cao. Cả hai điều kiện START và STOP đều được tạo ra bởi thiết bị chủ. Sau tín hiệu START, bus I2C coi như đang trong trạng thái làm việc (busy). Bus I2C sẽ rỗi, sẵn sàng cho một giao tiếp mới sau tín hiệu STOP từ phía thiết bị chủ.

Sau khi có một điều kiện START, trong quá trình giao tiếp, khi có một tín hiệu START được lặp lại thay vì một tín hiệu STOP thì bus I2C vẫn tiếp tục trong trạng thái bận. Tín hiệu START và lặp lại START (Repeated START) đều có chức năng giống nhau là khởi tạo một giao tiếp.

Truyền dữ liệu:

Mỗi xung clock có một bit dữ liệu được truyền. Mức tín hiệu SDA chỉ được thay đổi khi xung clock đang ở mức thấp, và ổn định khi xung clock ở mức cao.

Thiết bị tớ có thể lấy mẫu dữ liệu khi xung clock ở mức cao.



Hình 2. 28: Truyền dữ liệu I2C

2.5 Giới thiệu nền tảng Openhab

2.5.1 OpenHAB

- OpenHAB là một phần mềm mã nguồn mở có chức năng là bộ điều khiển trung tâm, với khả năng kết nối giao tiếp và điều khiển tới nhiều loại thiết bị khác nhau trong hệ thống SmartHome. OpenHab cung cấp nhiều giao diện người dùng (web site, android, ios....) giúp cho quá trình làm việc với OpenHab dễ dàng và thuận tiện hơn.

2.5.2 OpenHAB designer

- Là phần mềm hỗ trợ cho việc thiết lập:
- 1 trang mạng cho openHAB cùng với địa chỉ của nó.
- Giao diện openHAB.
- MQTT cho openHAB.
- Các giao thức truyền nhận dữ liệu, cũng như các quy luật làm việc cho openHAB...

2.5.3 Java

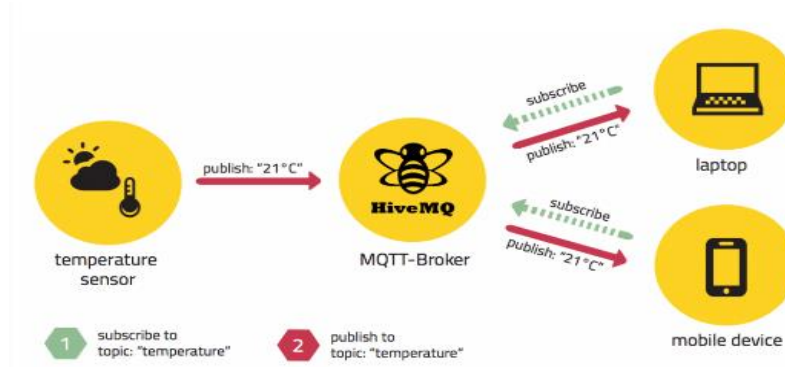
- OpenHAB được cài đặt dưới dạng một website, có thể chạy trên rất nhiều platform (Windows, Linux, ARM...), nó được viết bằng ngôn ngữ Java. Vì vậy các thiết bị cần có hỗ trợ JVM (Java Virtual Machine) để có thể cài đặt được Openhab.

2.5.4 MQTT

- MQTT (Message Queuing Telemetry Transport) là một giao thức gửi dạng publish/subscribe sử dụng cho các thiết bị Internet of Things với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định.
- Giao thức này sử dụng băng thông thấp trong môi trường có độ trễ cao nên nó là một giao thức lý tưởng cho các ứng dụng M2M và IoT.

Mô hình Publish/Subscribe

- Mô hình publish/subscribe (pub / sub) là một thay thế cho mô hình client-server truyền thống. Tuy nhiên Pub / Sub có những điểm đặc biệt, đó chính là nhà xuất bản (publish) và người đăng ký (subscribe) không biết về sự tồn tại của nhau. Có một thành phần thứ ba, được gọi là môi giới, được biết bởi cả nhà xuất bản và thuê bao, bộ lọc này lọc tất cả các tin nhắn gửi đến và phân phối chúng cho phù hợp.
- Việc tách riêng Pub/Sub có 3 chiều:



Hình 2. 29: Mô hình Publish/Subscribe

- +Không gian: Nhà xuất bản và thuê bao không cần phải biết nhau (theo chỉ ip và cổng)
- +Thời gian tách: Nhà xuất bản và người đăng ký không cần phải chạy đồng thời.
- +Đồng bộ: Các thao tác trên cả hai thành phần không bị dừng trong quá trình xuất bản hoặc tiếp nhận

2.5.4.1 Kết nối MQTT

- Giao thức MQTT dựa trên giao thức TCP / IP. Cả khách hàng (MQTT client) và nhà môi giới (broker) đều cần có ngăn TCP / IP.



Hình 2. 30: Kết nối MQTT

- Kết nối được bắt đầu thông qua một máy khách gửi một thông báo CONNECT tới người môi giới. Phản hồi của người môi giới với CONNACK và mã trạng thái. Khi kết nối được thiết lập, người môi giới sẽ giữ kết nối luôn mở, miễn là khách hàng không gửi lệnh ngắt kết nối hoặc mất kết nối.

2.5.4.2 MQTT Publish, Subscribe & Unsubscribe

•**Publish:** Sau khi một MQTT client được kết nối với một broker, client có thể publish tin nhắn. MQTT có một bộ lọc theo chủ đề dựa trên các message trên broker, do đó mỗi tin nhắn phải chứa một chủ đề, broker dễ dàng phân loại và chuyển tiếp thông điệp tới những khách hàng nào quan tâm.

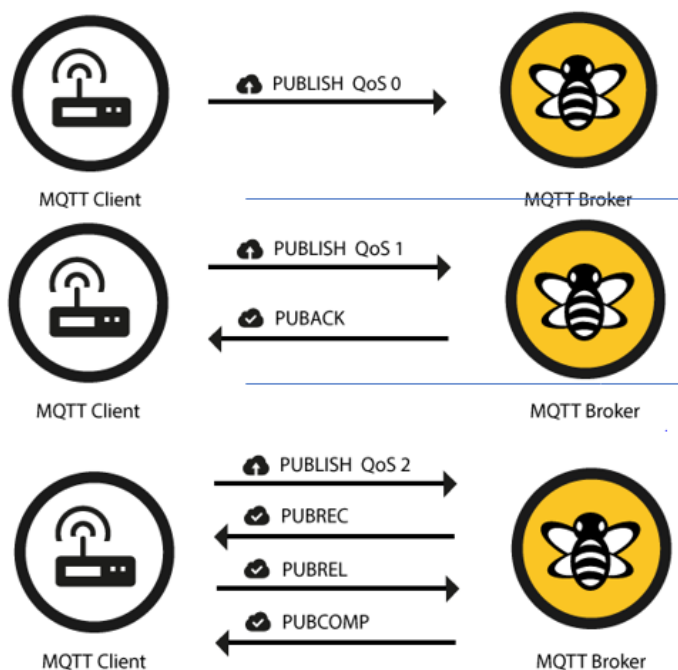
➤ Topic Name: Một chuỗi đơn giản, được cấu trúc theo thứ bậc với dấu

MQTT-Packet:	
PUBLISH	
contains:	Example
packetId (always 0 for qos 0)	4314
topicName	"topic/1"
qos	1
retainFlag	false
payload	"temperature:32.5"
dupFlag	false

Hình 2. 31: Publish

gạch chéo phía trước như các dấu phân cách. Ví dụ như là "myhome / livingroom / temperature" hoặc "Germany / Munich / Oktoberfest / people".

➤ Qos: Mức Chất lượng Dịch vụ (QoS). Mức (0,1 hoặc 2) xác định sự bảo đảm của một thông điệp đến đầu kia (client hoặc broker).



Hình 2. 32: Mô hình kết nối trong MQTT

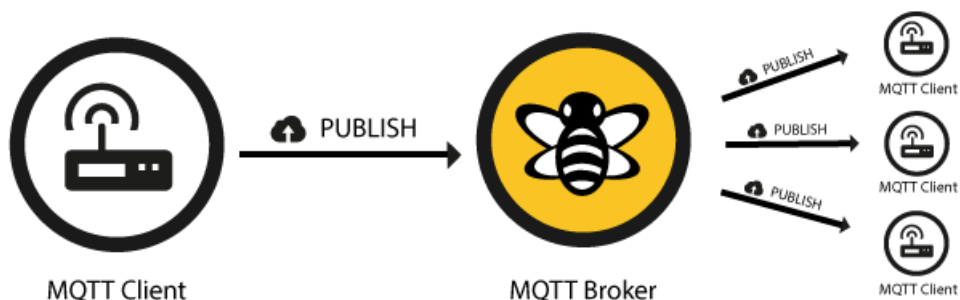
➤ Retain-Flag: Cờ này được xác định nếu message sẽ được lưu bởi broker cho một chủ đề nhất định. Client mới có thể đăng ký vào chủ đề đó và sẽ nhận được message được giữ lại cuối cùng về chủ đề đó ngay sau khi đăng ký.

➤ Payload: Đây là nội dung thực tế của tin nhắn.

➤ Packet Identifier: Bộ nhận dạng gói tin là sự nhận dạng duy nhất giữa client và broker để xác định một gói tin trong luồng chảy dữ liệu. Chỉ được đề cập khi sử dụng các Qos lớn hơn Qos 0. Việc đặt mã nhận dạng nội bộ MQTT này là trách nhiệm của thư viện ở client và / hoặc là trách nhiệm của broker.

➤ DUP Flag: Cờ trùng lặp chỉ ra, rằng message này là một bản sao và bị gửi lại bởi vì đâu kia không chấp nhận message gốc. Điều này chỉ xảy ra ở các QoS lớn hơn Qos 0. Cơ chế gửi lại / trùng lặp này thường được xử lý bởi thư viện ở MQTT client hoặc broker.

⇒ Vì vậy, khi một client gửi publish tới một MQTT broker, broker sẽ đọc, chấp nhận mess nếu cần (theo Cấp QoS) và sau đó xử lý nó. Quá trình xử lý bao gồm xác định những client nào đã đăng ký vào chủ đề đó và sau đó gửi thông tin đó tới các khách hàng đã chọn đăng ký vào chủ đề đó.

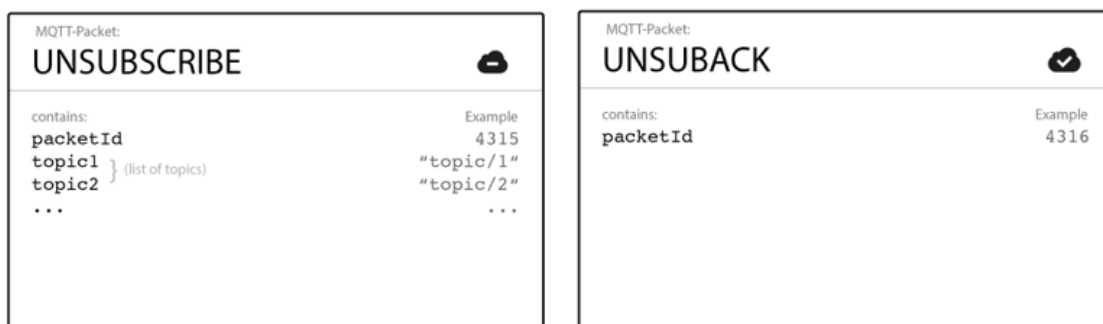


Hình 2. 33: Giao tiếp MQTT client và MQTT Broker

- **Subscribe và Suback:** là quá trình đăng kí để nhận dữ liệu từ broker của 1 client nào đó. Quá trình đó diễn ra bắt đầu khi một client bất kì muốn đăng kí nhận dữ liệu cho một topic của broker, client sẽ gửi gói subscribe đến, khi đó broker sẽ chấp nhận và gửi lại gói Suback có ý nghĩa là chấp nhận và bắt đầu gửi dữ liệu publish cho client.



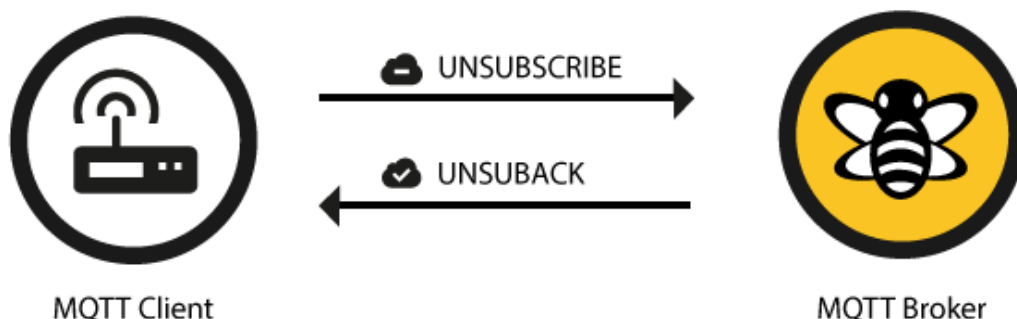
Hình 2. 34:Subscribe và Suback



Hình 2. 35: Unsubscribe và Unsuback

- **Unsubscribe và Unsuback:** Ngược lại với subscribe và suback, khi client không muốn nhận dữ liệu của 1 topic nào đó, nó sẽ gửi gói unsubscribe và khi đó, broker sẽ trả lại gói unsuback xác nhận kết thúc quá trình truyền nhận dữ liệu và không gửi dữ liệu topic đó cho client nữa.

- Thiết lập và liên kết được với MQTT là mấu chốt cho việc thực hiện đề tài này.



Hình 2. 36: Thiết lập và liên kết với MQTT

-Việc cài đặt MQTT thực hiện bởi việc thêm tập tin ‘org.openhab.binding.mqtt-1.8.3’ vào mục addons của openhab.

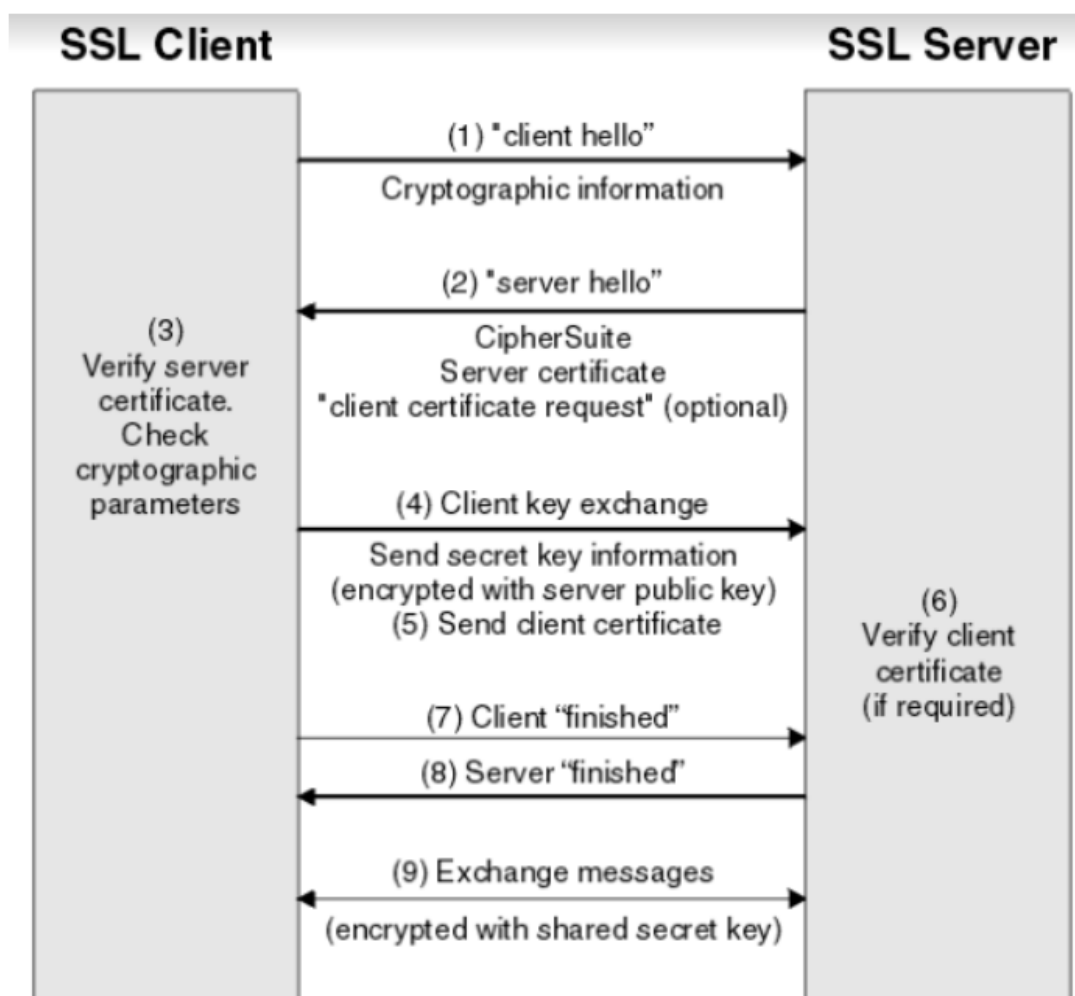
- Kế đến nó được đặt địa chỉ thông qua việc thiết kế openHAB và thiết kế code trên Arduino.

2.5.5 Bảo mật của giao thức MQTT:

MQTT broker có thể yêu cầu tên người dùng và mật khẩu xác thực từ client để kết nối. Để đảm bảo tính bảo mật, kết nối TCP có thể được mã hóa với SSL/TLS (Transport Layer Security (TLS) protocol, Secure Sockets Layer (SSL) protocol).

TLS (Transport Layer Security) và SSL (Secure Sockets Layer) cung cấp một kênh truyền thông an toàn giữa client và server. Cụ thể hơn, TLS/SSL là các giao thức mã hóa sử dụng cơ chế handshake (bắt tay) để tạo kết nối an toàn giữa máy khách và máy chủ. Sau khi hoàn thành các bước “handshake”, một kênh giao tiếp mã hóa giữa client và server được thiết lập và đảm bảo không kẻ tấn công nào có khả năng nghe trộm thông tin trong suốt quá trình giao tiếp. Server cung cấp một X509 certificate, là chứng chỉ thường được phát hành bởi một cơ quan tin cậy, và client sẽ sử dụng để chứng thực danh tính của server.

MQTT và TLS: Thông thường giao thức MQTT dựa vào TCP làm giao thức tầng giao vận, nghĩa là mặc định kết nối giữa broker và client là kết nối không bảo mật. Để tăng tính bảo mật và có khả năng mã hóa các kênh giao tiếp, hầu hết các MQTT broker hiện nay như Mosquitto, HiveMQ đều cho phép sử dụng TLS. Port 8883 được chuẩn hóa để sử dụng cho kết nối MQTT có bảo mật. Tên được đặt theo tổ chức IANA là “secure-mqtt” và port 8883 được dành riêng cho giao thức MQTT sử dụng TLS. Tuy nhiên, dễ đoán rằng có một nhược điểm khi sử dụng TLS cho giao thức MQTT đó là tăng tính bảo mật đồng nghĩa với tiêu tốn năng lượng cho CPU nhiều hơn và quá trình giao tiếp sẽ không mượt mà như mọi khi. Mặc dù so với TCP thông thường, việc sử dụng CPU dành cho giao thức bảo mật TLS nhiều hơn không đáng kể nhưng đối với các thiết bị được thiết kế dành cho một nhiệm vụ chuyên sâu nào đó rất hạn chế về tốc độ, bộ nhớ hay băng thông kênh truyền thì đó là một vấn đề lớn. Vì vậy kỹ thuật Session Resump thường được sử dụng để tăng hiệu suất khi sử dụng TLS, đây là một kỹ thuật dùng để nối lại phiên làm việc cũ mà không cần bắt đầu lại tất cả các bước “handshake”.



Hình 2. 37: Minh họa SSL/TLS handshake

Mặc dù MQTT được thiết kế gọn nhẹ, nó có hai nhược điểm làm hạn chế các thiết bị:

Mỗi client MQTT phải hỗ trợ TCP và thường sẽ giữ một kết nối mở đến broker ở mọi thời điểm. Đối với một số môi trường mà xác suất mất bản tin cao hay không có sẵn máy tính thì đây là cả một vấn đề.

Tên kênh MQTT thường là các chuỗi dài làm chúng không thỏa đáng với tiêu chuẩn 802.15.4

Cả hai thiết sót được giải quyết bằng giao thức MQTT-SN, trong đó xác định một ánh xạ UDP của MQTT và thêm khả năng lập chỉ mục (indexing) tên các kênh cho broker.

2.5.6 Mosquitto

- Mosquitto là một phần mềm mã nguồn mở MQTT broker sử dụng giao thức MQTT version 3.1 và 3.1.1. Nó đóng vai trò trung gian để tiếp nhận, xử lý và chuyển tiếp các message đến các thiết bị trong hệ thống. Data được truyền và

nhận thông qua các topic (hay còn gọi là channel). Với các channel, ta có thể gửi data tới đó hoặc đăng ký nhận bất kỳ dữ liệu nào gửi tới nó.

2.5.7 MQTT lends

- MQTT lends là công cụ được hỗ trợ trên các nền tảng có chức năng quan sát, xác nhận khi các liên kết được thiết lập, quan sát quá trình truyền nhận dữ liệu.

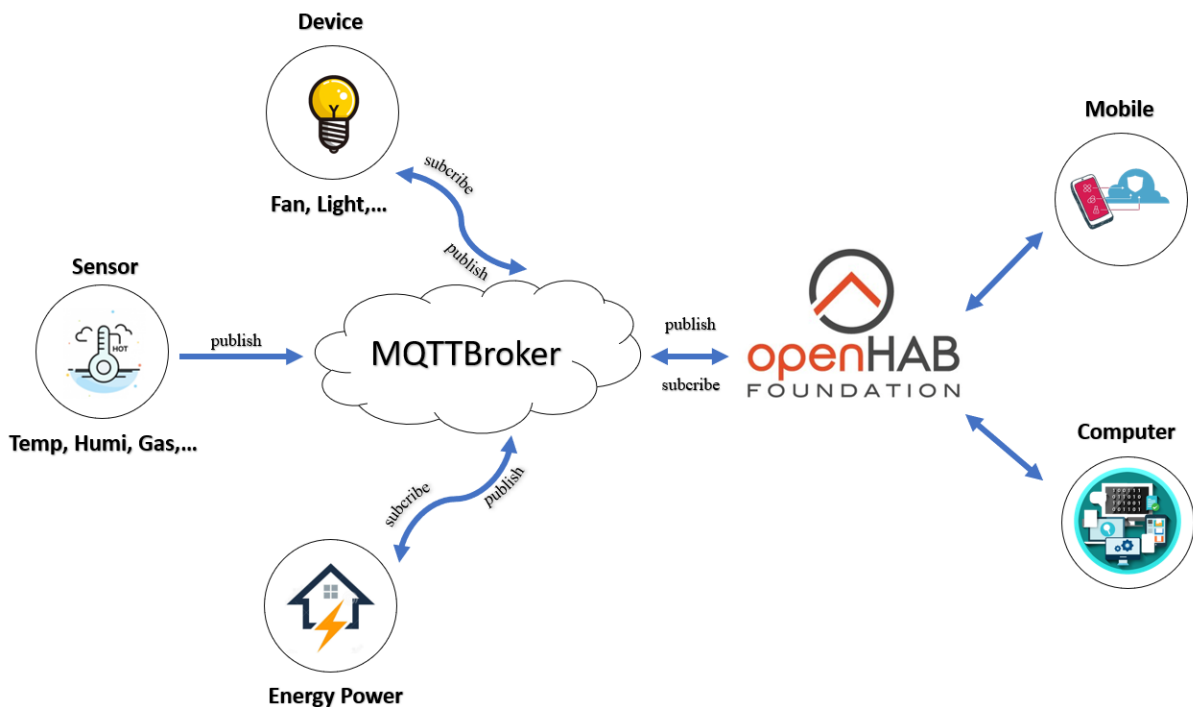
CHƯƠNG 3: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG

3.1 Giới thiệu

Trong chương này, trình bày về cách thiết kế, sơ đồ khối, sơ đồ nguyên lý của các board mạch của hệ thống: mạch điều khiển các thiết bị ngõ ra có sử dụng relay, mạch nguồn cung cấp điện áp – dòng điện cho cả hệ thống. Quá trình vẽ mạch in lắp ráp các thiết bị, đo kiểm tra mạch, lắp ráp mô hình. Thiết kế lưu đồ giải thuật cho chương trình và viết chương trình cho hệ thống. Trình bày về những kết quả đã được mục tiêu đề ra sau quá trình nghiên cứu thi công.

3.2 Thiết kế hệ thống

3.2.1 Thiết kế sơ đồ khối hệ thống



Hình 3. 1: Sơ đồ khối hệ thống

Chức năng từng khối:

- **Khối nguồn và pin:** Đây là khối quyết định sự hoạt động của hệ thống, khối này có nhiệm vụ cấp nguồn cho toàn hệ thống hoạt động một cách ổn định. Trong đề tài này sử dụng 3 nguồn: 5VDC cấp mạch relay, 3.3VDC cấp cho ESP8266, 220VAC cấp cho bóng đèn, quạt và các thiết bị khác.
- **Khối đo điện năng:** Sử dụng module đo điện năng chuyên dụng để đo các thông số về điện áp, dòng điện, công suất, điện năng tiêu thụ sau đó gửi về cho vi điều khiển xử lý.
- **Khối xử lý và điều khiển:** Là khối điều khiển trung tâm cho toàn hệ thống, có nhiệm vụ kết nối, nhận dữ liệu của các cảm biến đọc được và điều khiển gửi lên cơ sở dữ liệu để người sử dụng có thể giám sát và điều khiển từ xa. Ngoài ra, khối này cũng có thể điều khiển và giám sát được các thiết bị điện như các khối

xử lý và điều khiển khác trong hệ thống.

- **Khối cảm biến:** để thu thập các thông số dữ liệu từ các cảm biến gửi về khối điều khiển trung tâm từ đó đưa ra các hoạt động đến khối chấp hành. Các cảm biến được dùng như là:

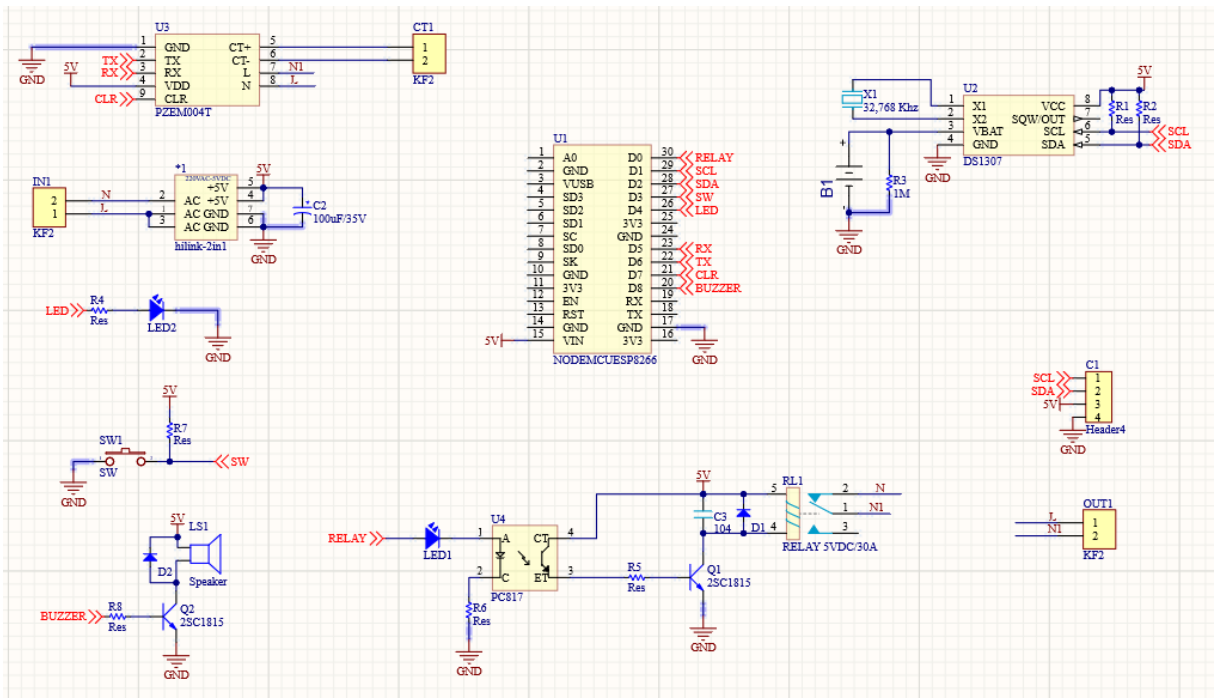
+ **Cảm biến khí Gas:** Đo nồng độ khí Gas và các chất dễ cháy trong không khí gửi về vi điều khiển đo và xử lý.

+ **Cảm biến nhiệt độ, độ ẩm:** Đo nhiệt độ môi trường và gửi về vi điều khiển xử lý

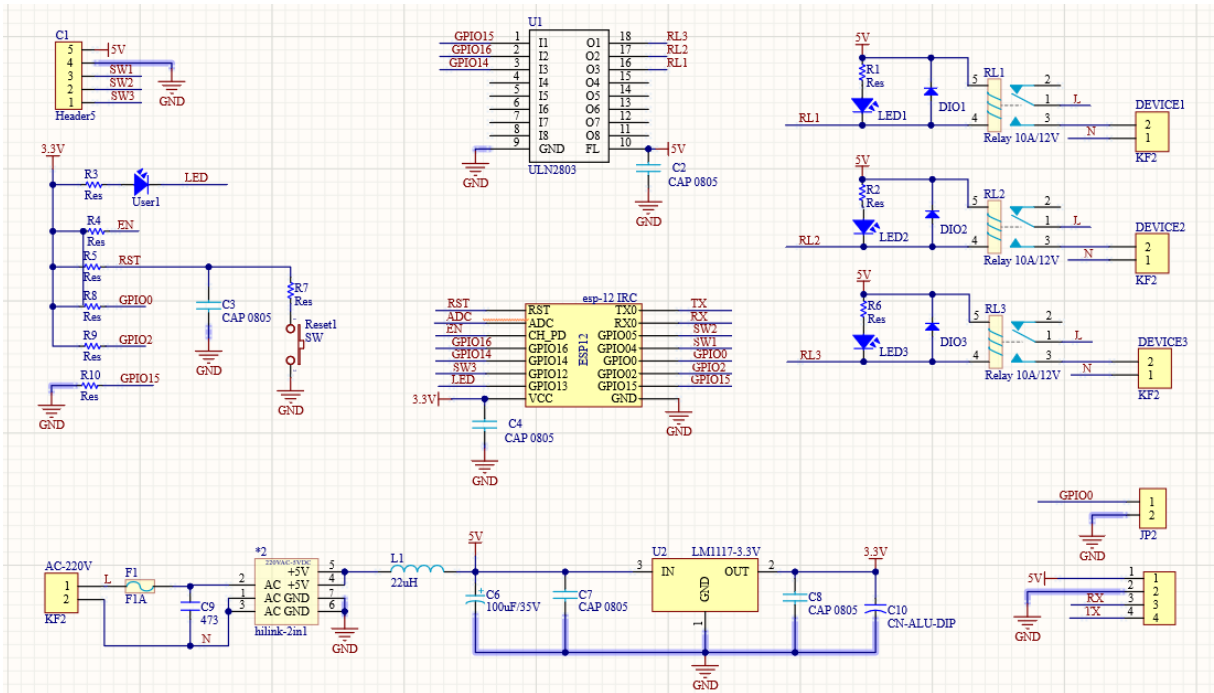
- **Khối chấp hành:** Là nơi chấp hành các dữ liệu mà vi điều khiển đã tổng hợp xử lý và gửi đến. Bao gồm điều khiển thiết bị như quạt, đèn, động cơ, ...

- **Khối hiển thị:** Hiển thị thời gian, giá trị cảm biến, trạng thái thiết bị trên màn hình thông qua các phần mềm nền tảng trên máy tính hoặc điện thoại.

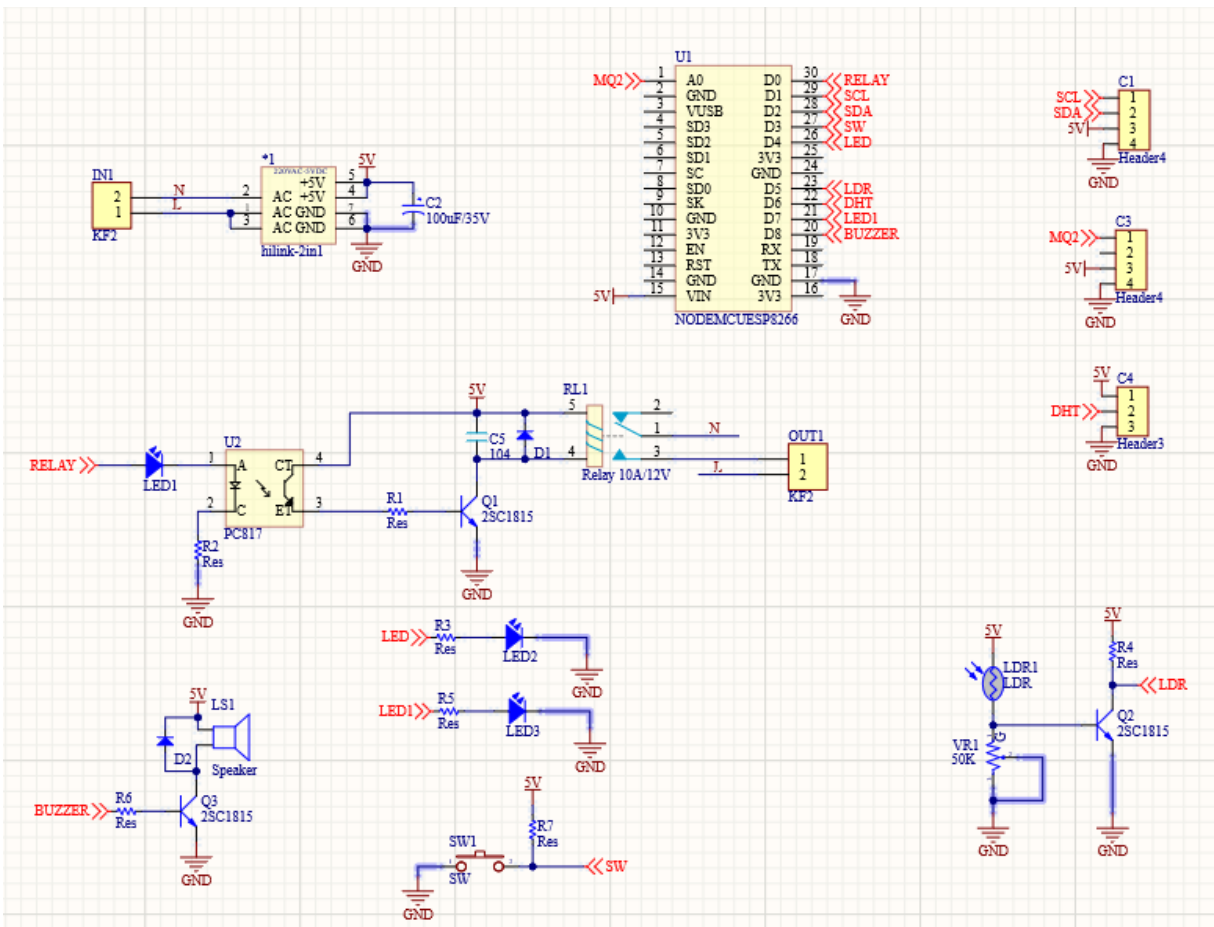
3.2.2 Sơ đồ nguyên lý toàn mạch



Hình 3. 2: Sơ đồ nguyên lý mạch giám sát năng lượng



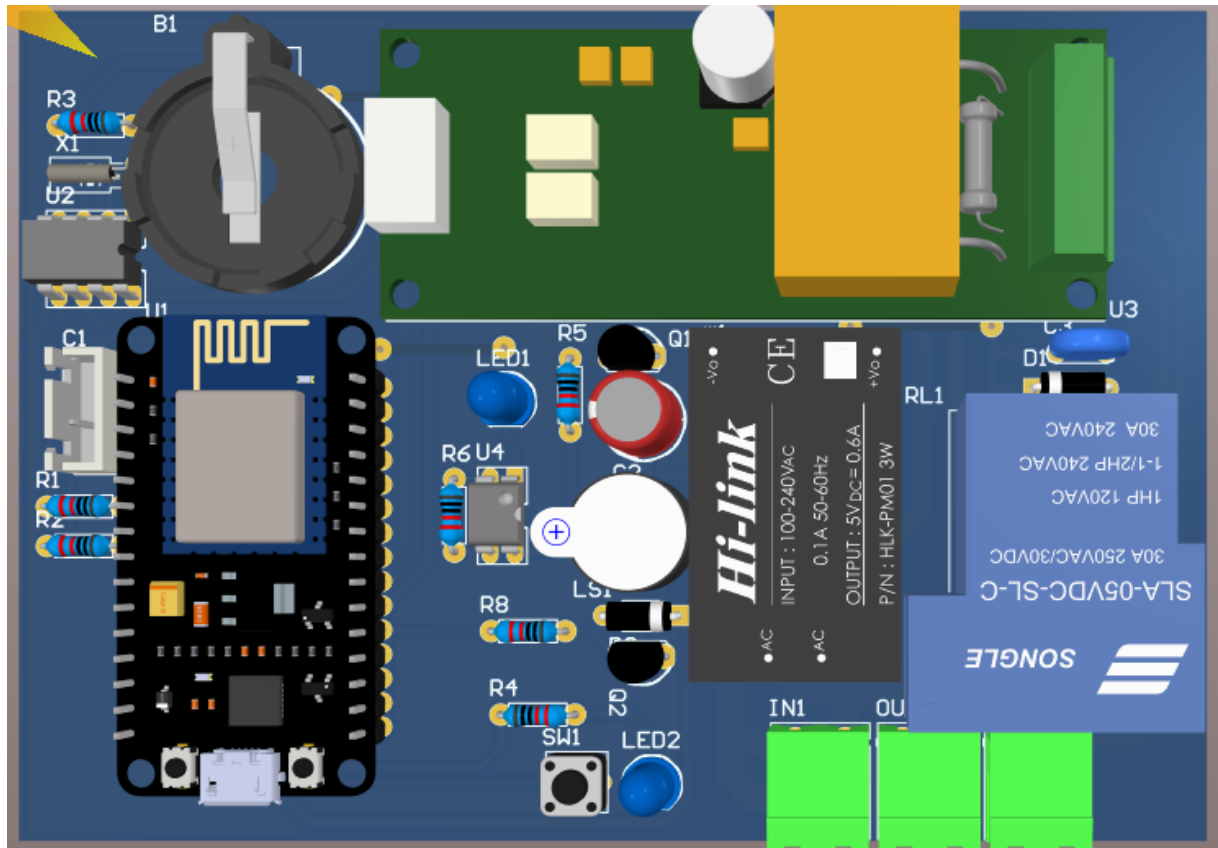
Hình 3. 3: Sơ đồ nguyên lý mạch điều khiển thiết bị



Hình 3. 4: Sơ đồ nguyên lý mạch cảm biến

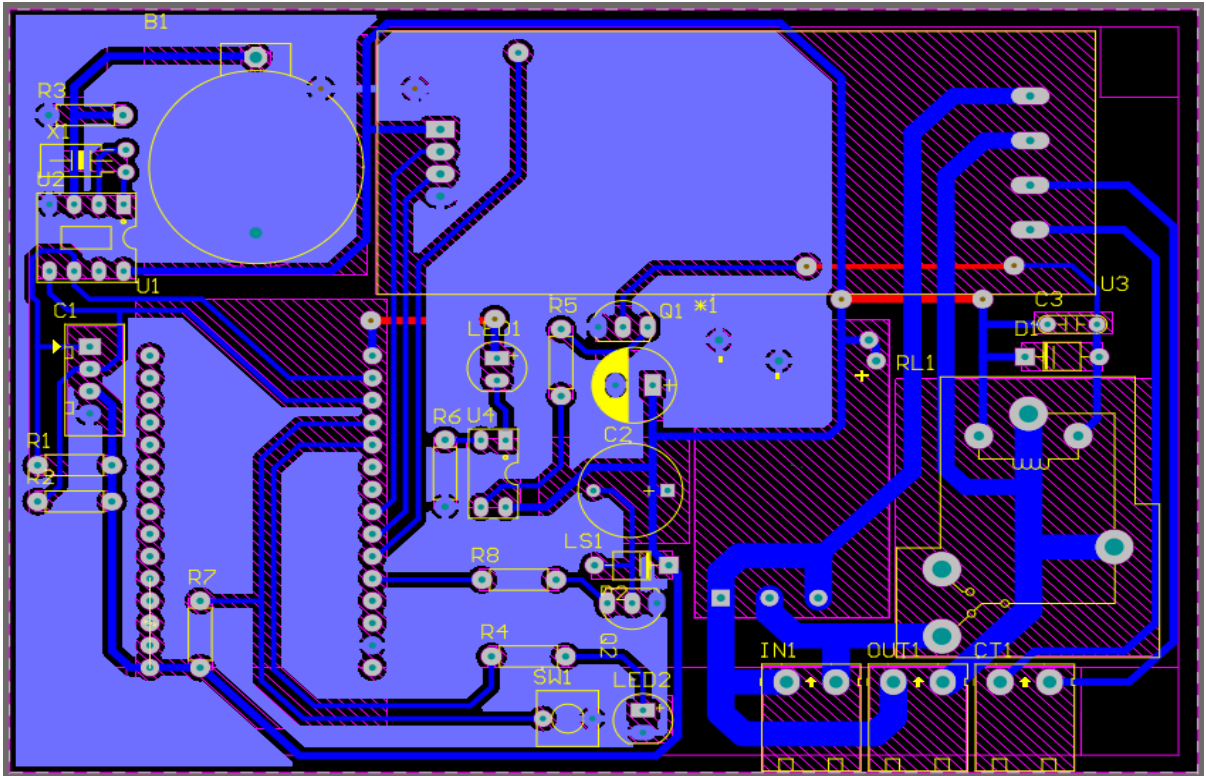
3.3 Thi công hệ thống

3.3.1 Thi công board mạch

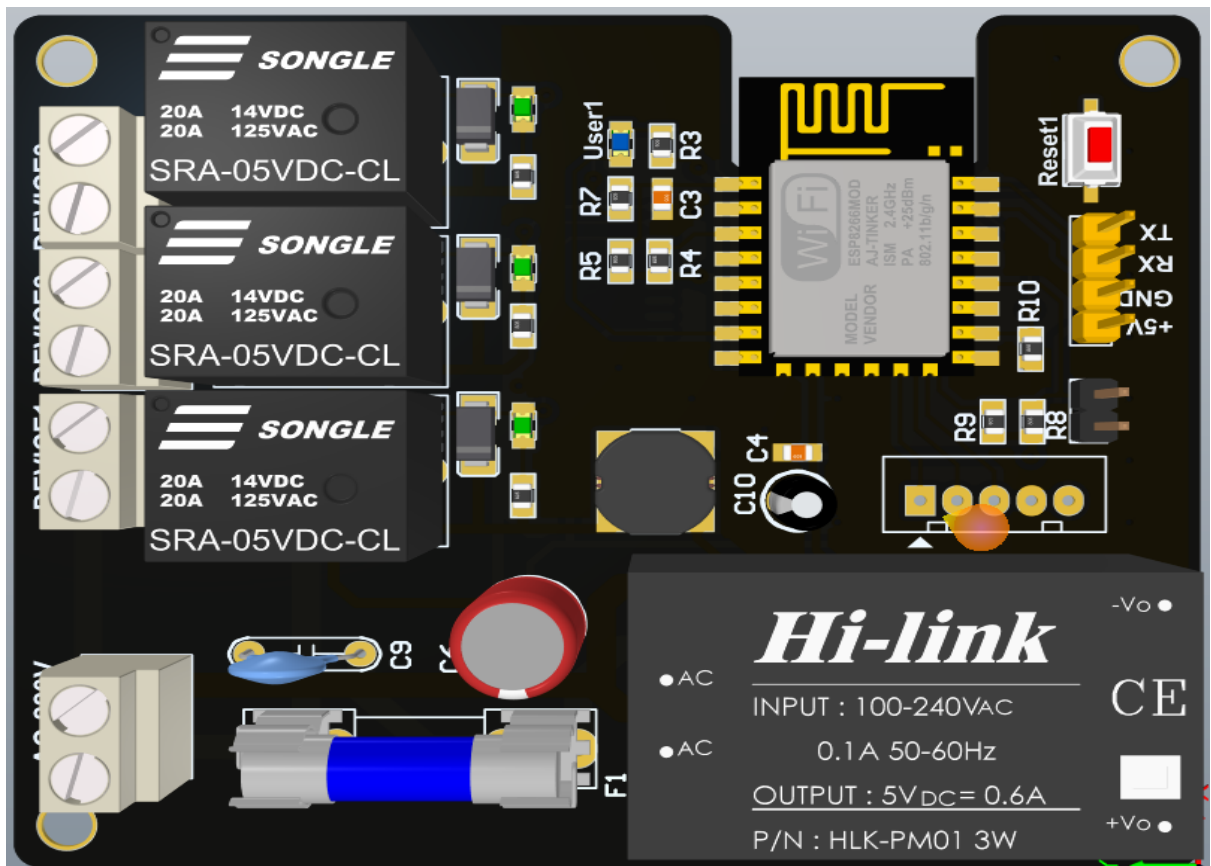


Hình 3. 5: Bố trí linh kiện mặt trên cho mạch giám sát năng lượng

Toàn thể sơ đồ bố trí được thiết kế rõ ràng. Hình 4.1 bao gồm khối nguồn gồm có module chuyển nguồn AC – DC. Module đo điện năng PZEM. Tiếp đến là khối điều khiển chính làm module vi điều khiển esp8266, và khối điều khiển ngõ ra công suất relay để giao tiếp mạch nút nhấn từ người sử dụng.

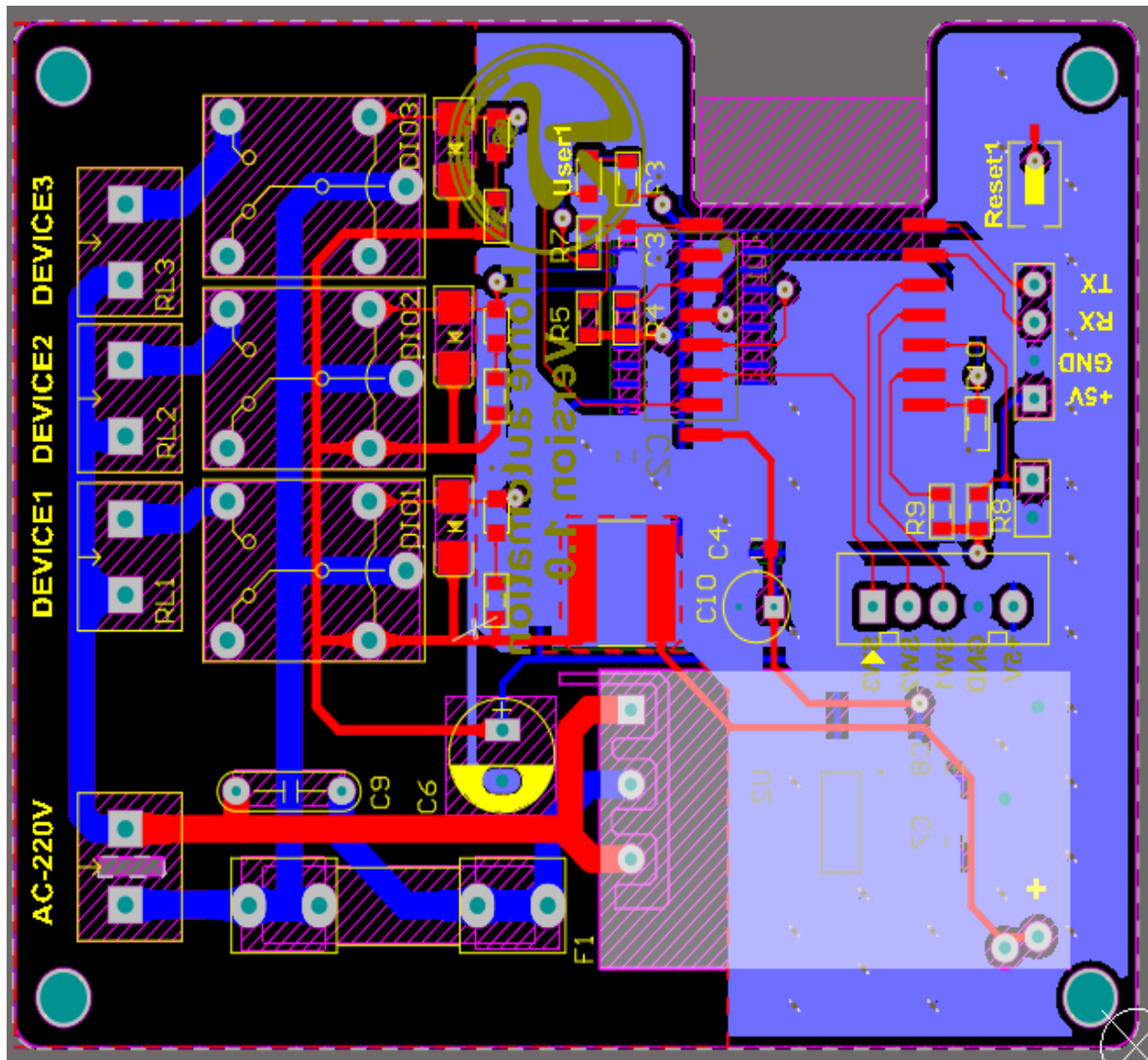


Hình 3. 6: Mạch pcb lớp bottom của mạch giám sát năng lượng

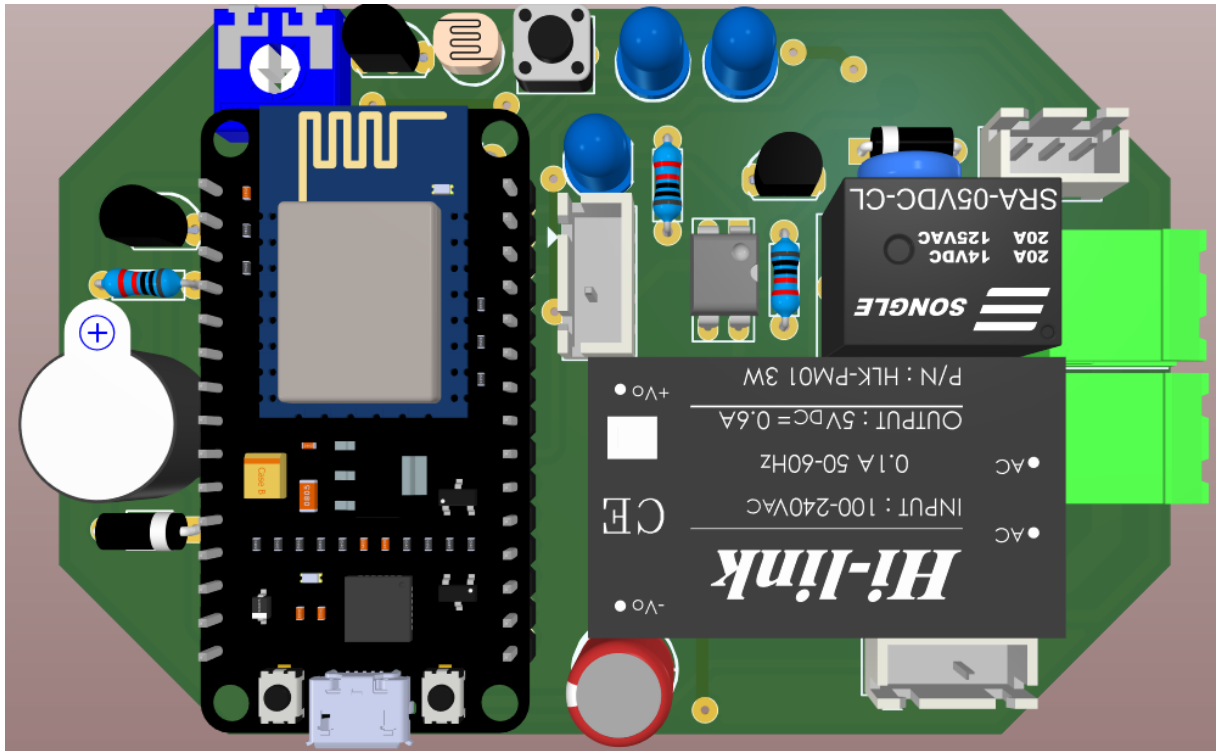


Hình 3. 7: Bố trí linh kiện mặt trên cho mạch điều khiển thiết bị

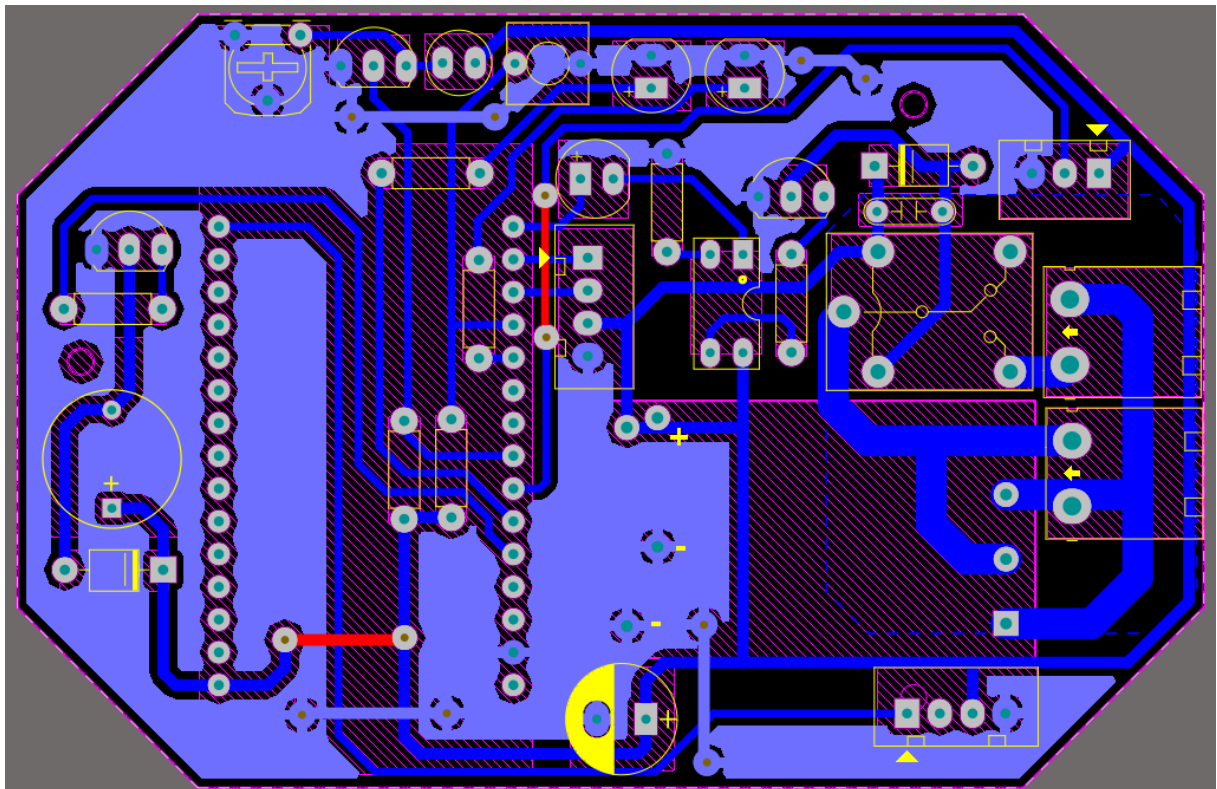
Mạch điều khiển thiết bị cũng được thiết kế tương tự mạch chính với các khối được phân biệt rõ ràng.



Hình 3. 8: Pcb lớp bottom của mạch điều khiển thiết bị



Hình 3. 9: Bố trí linh kiện mạch cảm biến



Hình 3. 10: PCB lớp bottom mạch cảm biến

3.3.2 Lắp ráp và kiểm tra

Sau khi thiết kế, nhóm tiến hành, ủi mạch, rửa mạch và hàn linh kiện. Kiểm tra mạch xem hệ thống bị đứt hay chập mạch hay không nhờ vào đồng hồ VOM. Kiểm tra chân nguồn, chân điều khiển. Chú ý, quan sát kỹ các đường mạch có bị sát vào nhau hay không, nếu có thì dùng mũi khoan cỡ 0.8 ly tách nhẹ đường mạch bị sát vào nhau. Kiểm tra nếu có bị đứt dây thì ta dùng dây điện nối lại hoặc dùng chì để nối

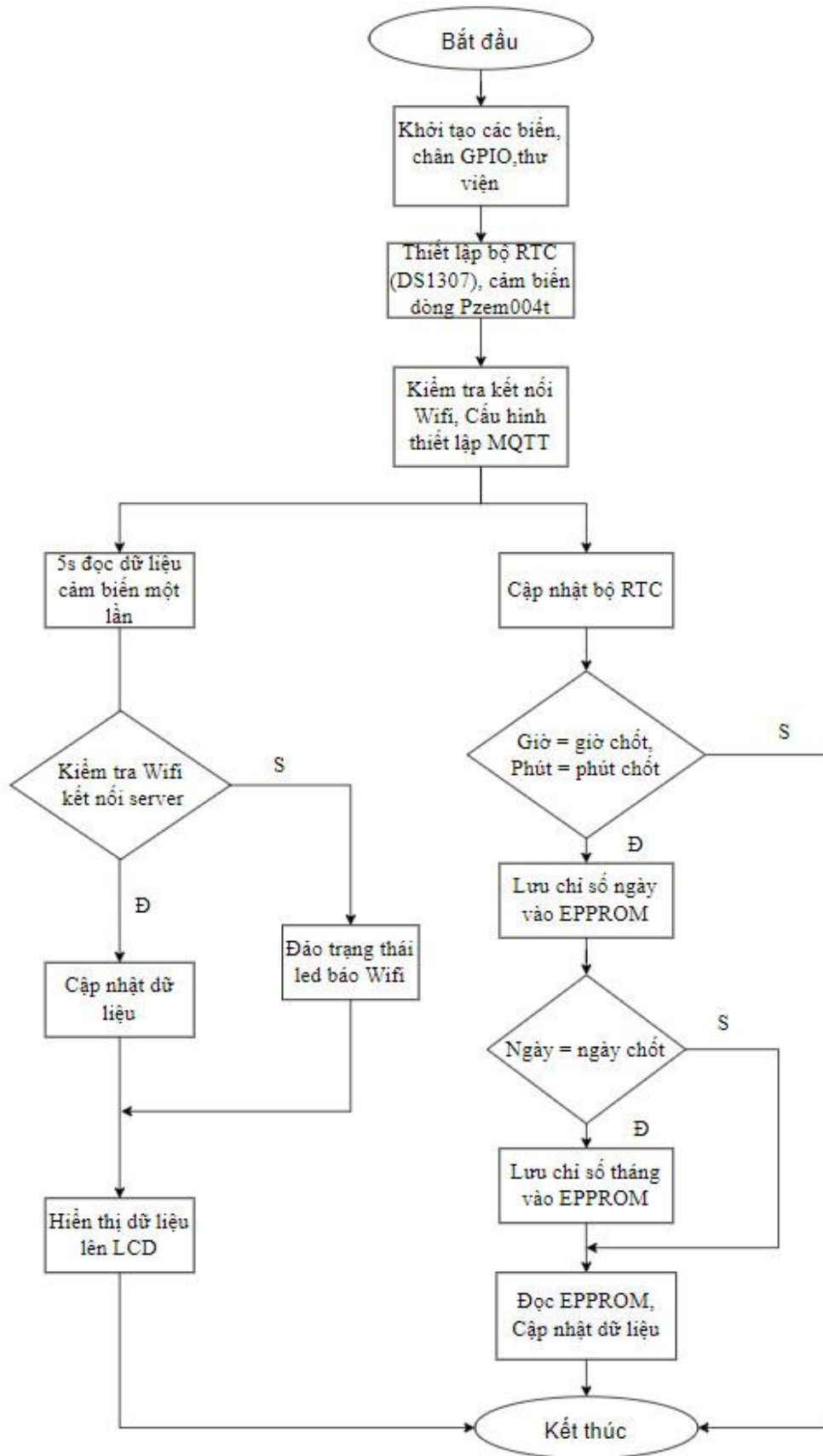
3.3.3 Đóng gói và thi công mô hình

Sau khi thi công mạch xong, nhóm tiến hành kiểm tra và lắp ráp mạch vào hộp nhựa cứng có kích thước dài x rộng x cao: 16.5 x 9.0 x 5.6 cm như hình bên dưới.

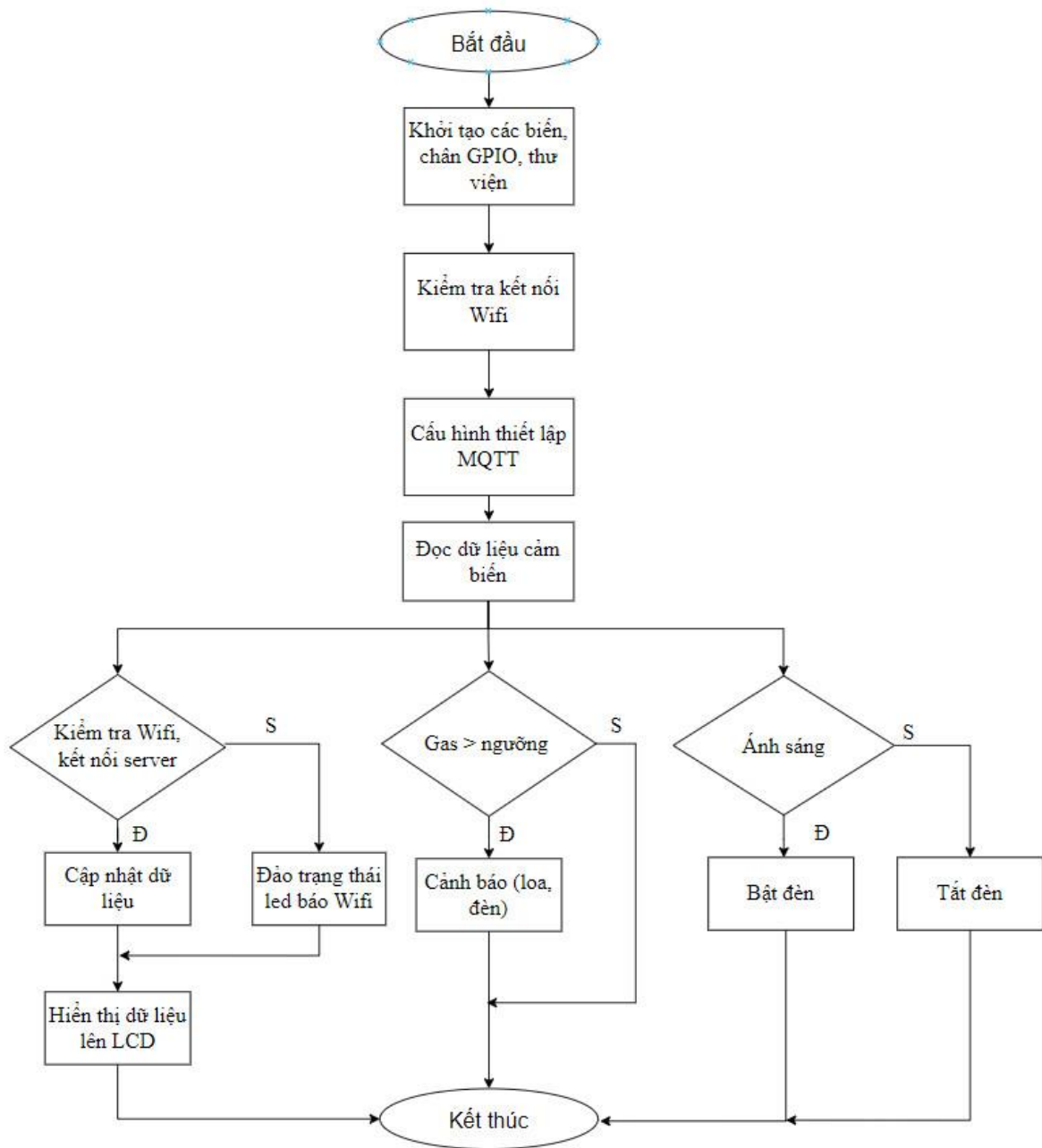
Hộp nhựa cứng là loại hộp kín, để gắn màn hình hiển thị cũng như tạo khoảng trống để kết nối dây điện nguồn, các thiết bị thì nhóm dùng lười cắt để tạo hình. Cách làm này thủ công, khó và độ chính xác, thẩm mỹ chỉ ở mức tương đối. Tuy nhiên cách này tiết kiệm chi phí so với đặt hàng đúng kích thước mong muốn.

3.4 Lập trình hệ thống

3.4.1 Lưu đồ giải thuật



Hình 3. 11: Lưu đồ giải thuật hệ thống 1



Hình 3. 12: Lưu đồ giải thuật hệ thống 2

3.4.2 Phần mềm lập trình cho vi điều khiển

Arduino IDE (Integrated Development Environment) là môi trường lập trình tích hợp mã nguồn mở hỗ trợ người dùng viết code và tải nó lên board mạch arduino. Đây là môi trường lập trình đa nền tảng hỗ trợ hầu hết các board Arduino hiện nay. Một số tính năng chính của phần mềm như viết code và biên dịch cho Arduino, có nhiều ví dụ mẫu, hỗ trợ nhiều thư viện như SoftwareSerial, FirebaseArduino, SPI, ESP8266, Servo, UART,... Phiên bản Arduino IDE đầu tiên là Arduino 1.0 sau đó được phát triển dần lên phiên bản Arduino 1.0.1, 1.0.2,... và phiên bản mới nhất hiện nay là Arduino 1.8.7.

Trong luận văn Arduino IDE được sử dụng kèm với những thư viện PubSubClient,

NTPClient, ESP8266WiFi,...để lập trình cho module Wifi ESP8266 NodeMCU.

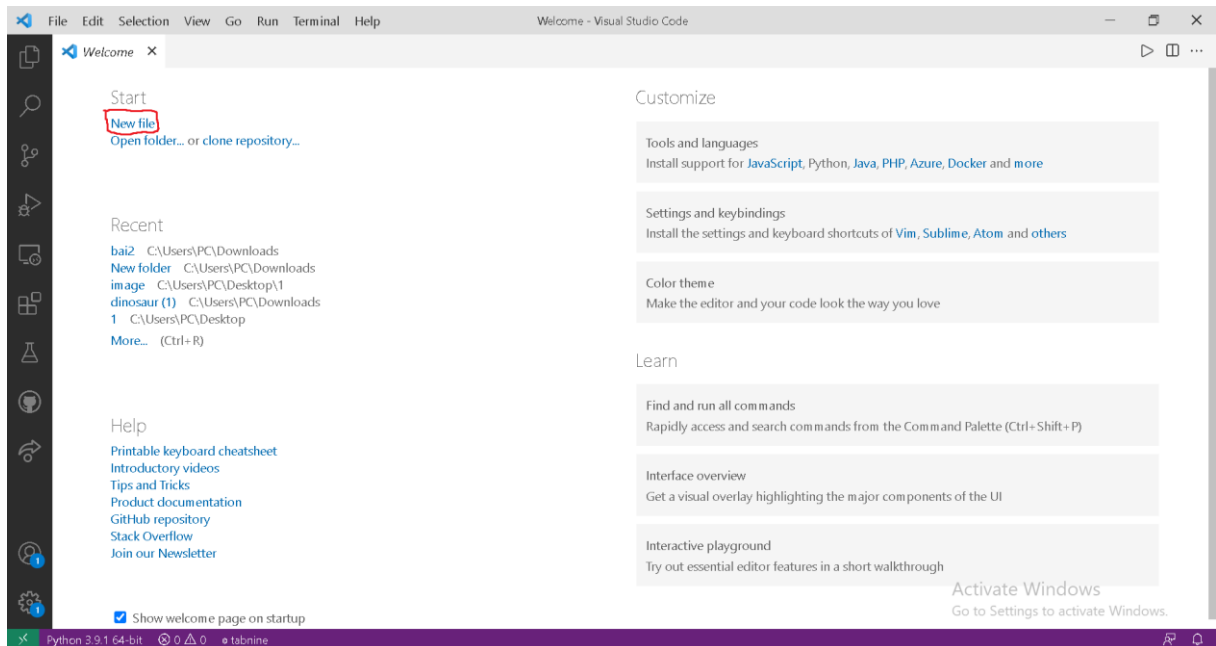
3.4.3 Phần mềm lập trình giao diện điều khiển

VISUAL STUDIO CODE: Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

- Chức năng của Visual Studio Code
 - Hỗ trợ nhiều ngôn ngữ lập trình
 - Hỗ trợ đa nền tảng
 - Cung cấp kho tiện ích mở rộng
 - Kho lưu trữ an toàn
 - Hỗ trợ web
 - Lưu trữ dữ liệu dạng phân cấp
 - Hỗ trợ viết Code
 - Màn hình đa nhiệm
 - Hỗ trợ thiết bị đầu cuối

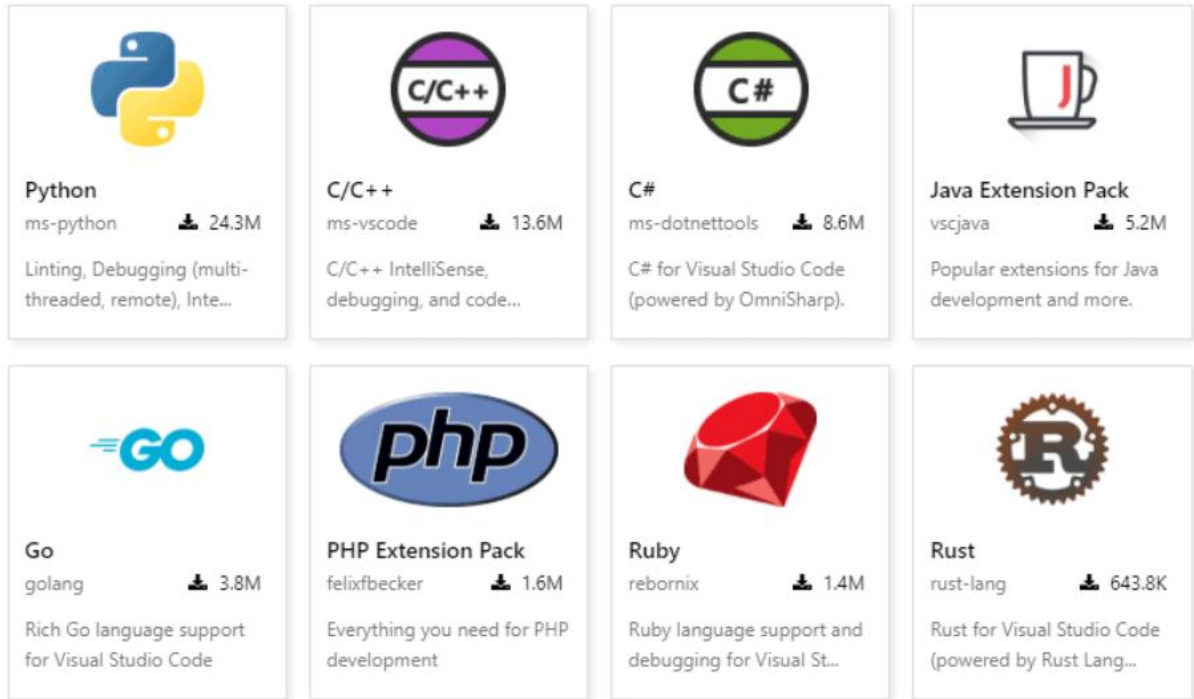
Về giao diện và cách sử dụng, visual studio code trông rất hiện đại và dễ dàng tiếp cận cho các bạn newbie. Để tìm hiểu rõ về các chức năng chính của phần mềm này, mời bạn tham khảo các bước sau.

- Thiết lập Visual Studio Code



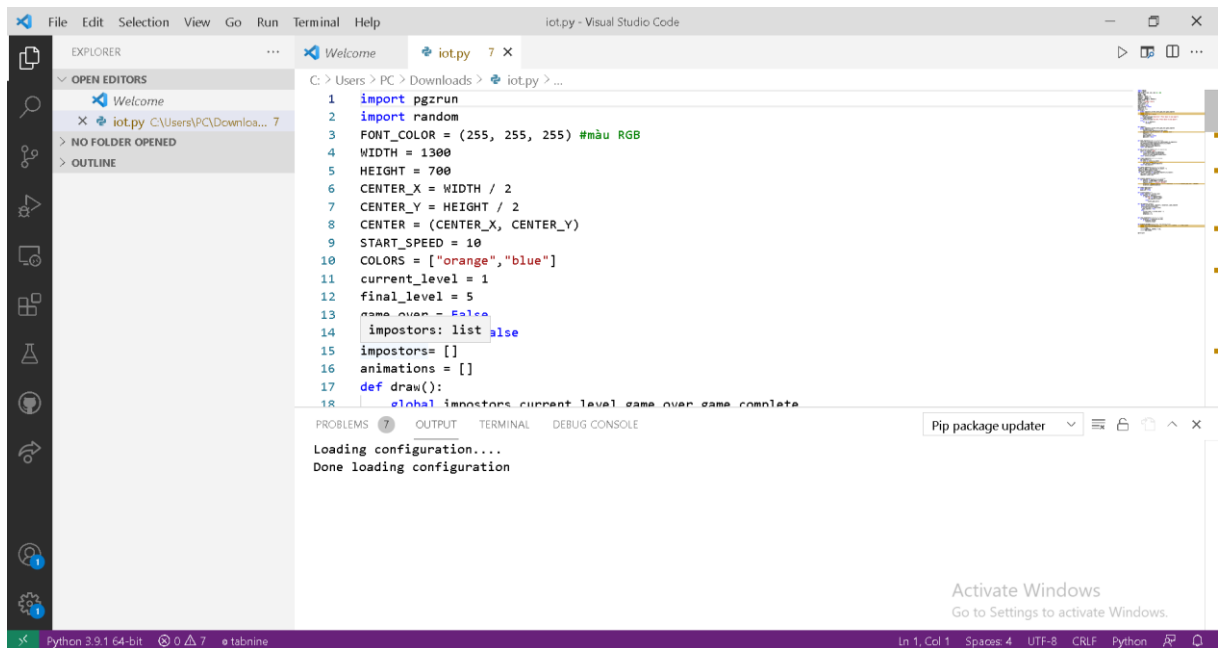
Hình 3. 13: Tạo File mới trong Visual Studio Code

- Visual Studio Code hỗ trợ những ngôn ngữ dưới đây



Hình 3. 14: Ngôn ngữ lập trình trên Visual Studio Code

- Code theo phong cách của bạn



Hình 3. 15: Code trên Visual Studio Code

3.5 Kết quả - nhận xét - đánh giá

3.5.1 Giới thiệu

Đây là nơi trình bày kết quả của cả quá trình nghiên cứu làm đề tài trong thời gian 15 tuần. Bên cạnh đó là nhận xét, đánh giá và đề xuất hướng phát triển của sản phẩm mô hình để hoàn thiện và có thể đi vào thực tế.

3.5.2 Kết quả đạt được

Qua quá trình nghiên cứu và thực hiện đề tài, nhóm nhận thấy được đề tài có khả năng ứng dụng vào thực tiễn cao, đáp ứng được nhu cầu cuộc sống hiện đại ngày nay và đi đúng xu hướng của thời đại. Đồng thời đề tài này cũng là một nguồn tài liệu có giá trị cho các bạn sinh viên những khóa tiếp theo có thể tham khảo khi nghiên cứu những đề tài có liên quan. Bên cạnh đó, nhóm cũng bổ sung cho mình những kiến thức hay và bổ ích như:

Biết cách sử dụng các cảm biến: Các cảm biến được sử dụng trong đề tài này (DHT11, MQ2, cảm biến ánh sáng) là những cảm biến cần thiết trong cuộc sống. Có thể nói, các cảm biến ngày nay được sản xuất ngày một đơn giản và nhỏ gọn hơn, nhằm đáp ứng nhu cầu của người tiêu dùng. Quá trình nghiên cứu đề tài này đã giúp nhóm thực hiện nâng cao kỹ năng đọc – hiểu cảm biến, trau dồi thêm kiến thức về các loại cảm biến. Đặc biệt là, tăng khả năng vận dụng lý thuyết vào thực tế, có khả năng lựa chọn được loại cảm biến phù hợp với từng yêu cầu của thực tiễn đời sống.

Biết cách sử dụng Module Wifi ESP8266 v12: ESP8266 đang được sử dụng rất phổ biến trong các thiết bị giao tiếp với smartphone hay webserver thông qua Wifi nhờ giá thành rẻ, module nhỏ gọn và đặc biệt dễ sử dụng. Đây là công cụ cải tiến quá trình giám sát từ có dây sang không dây, giúp ta hội nhập vào cuộc sống hiện đại ngày nay. Qua quá trình nghiên cứu, nhóm đã biết cách sử dụng module này để giao tiếp truyền và nhận dữ liệu với Database (Firebase của Google) giúp cho việc giám sát và điều khiển thiết bị điện được thực hiện từ xa. Ngoài ra, nhóm cũng đã biết cách sử dụng ngoại vi HSPI của module này để giao tiếp với vi điều khiển chính (ARM STM32) phục vụ cho việc truyền nhận dữ liệu cảm biến và lệnh điều khiển.

Biết cách sử dụng Module đo điện năng PZEM004T: Việc đo lường điện năng là rất cần thiết độ chính xác cao nhưng lại khá nguy hiểm vì phải thao tác với điện áp cao và cần những công cụ, thiết bị chuyên dụng. Nhằm đơn giản hóa việc đo lường điện năng, nhóm đã tìm hiểu, so sánh các phương pháp và quyết định sử dụng module đo điện năng chuyên dụng PZEM004T. Qua quá trình thực hiện đề tài, nhóm đã biết cách thức giao tiếp và sử dụng module này, nhóm nhận thấy module này sử dụng khá đơn giản và độ chính xác khá cao, rất thích hợp cho những đề tài về đo lường điện

Biết cách viết một trang web để điều khiển thiết bị: Việc viết một trang web phục vụ cho những mục đích riêng là rất cần thiết nhưng không phải dễ, đặc biệt đây

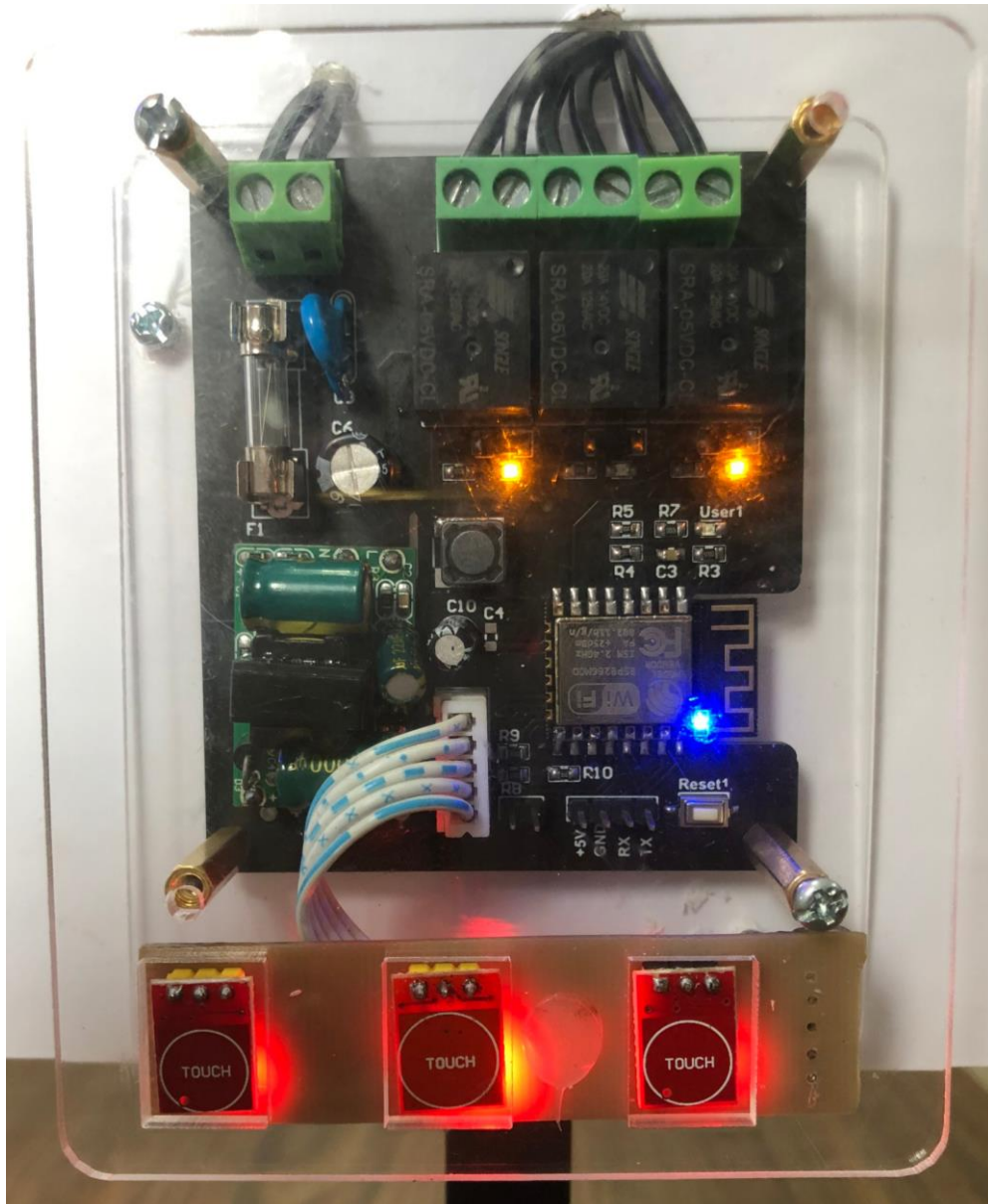
không phải là chuyên ngành của các thành viên trong nhóm. Nhưng nhóm cũng đã cố gắng tìm hiểu cũng như cũng đã tự viết cho mình một web chạy trên nền tảng Openhab

Hiểu rõ hơn về cách truyền nhận dữ liệu và các chuẩn truyền dữ liệu: Các chuẩn truyền dữ liệu như UART, I2C, MQTT, HTML ... được sử dụng rất phổ biến trong các vi điều khiển, các thành viên trong nhóm cũng đã được học về chúng trong các môn học liên quan tới *Vi điều khiển* trong quá trình học tập tại trường nhưng vẫn có phần chưa hiểu rõ về chúng. Qua đề tài này, nhóm đã có phần hiểu rõ hơn về chúng và đã sử dụng được các chuẩn này trong việc giao tiếp giữa hai vi điều khiển hay vi điều khiển với ngoại vi.

3.5.3 Kết quả thực nghiệm



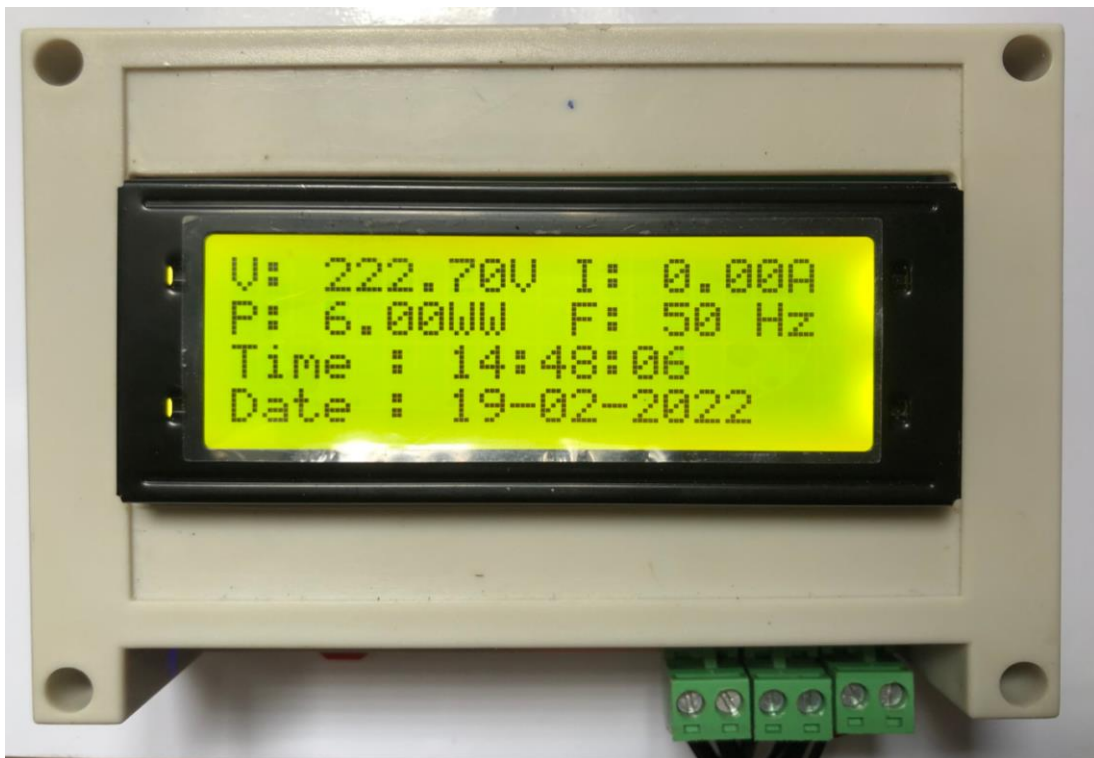
Hình 3. 16: Mô hình sản phẩm khi hoàn thành



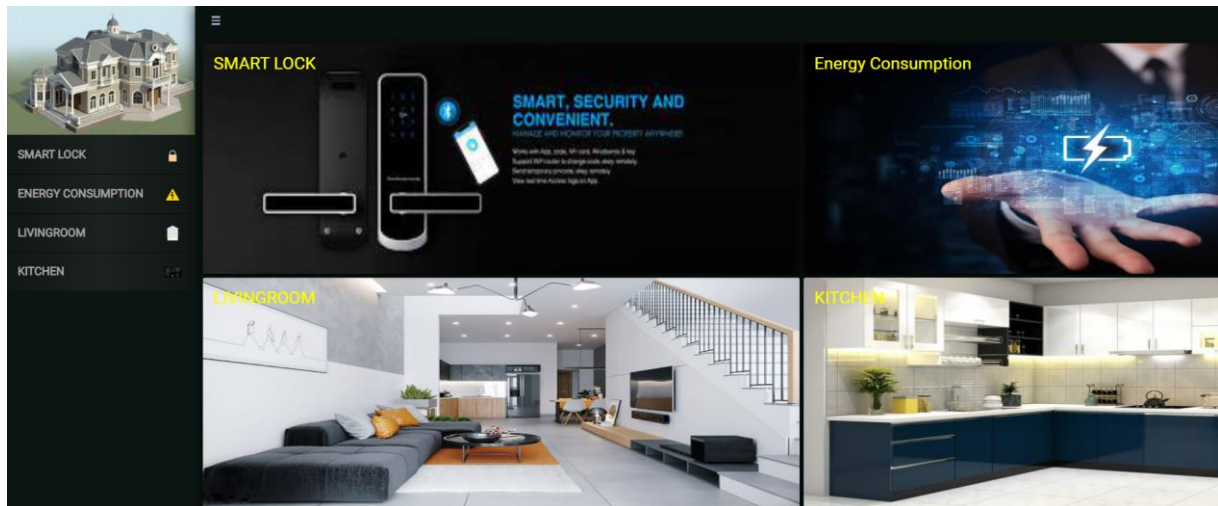
Hình 3. 17: Module điều khiển thiết bị



Hình 3. 18: Module giám sát nhiệt độ, độ ẩm



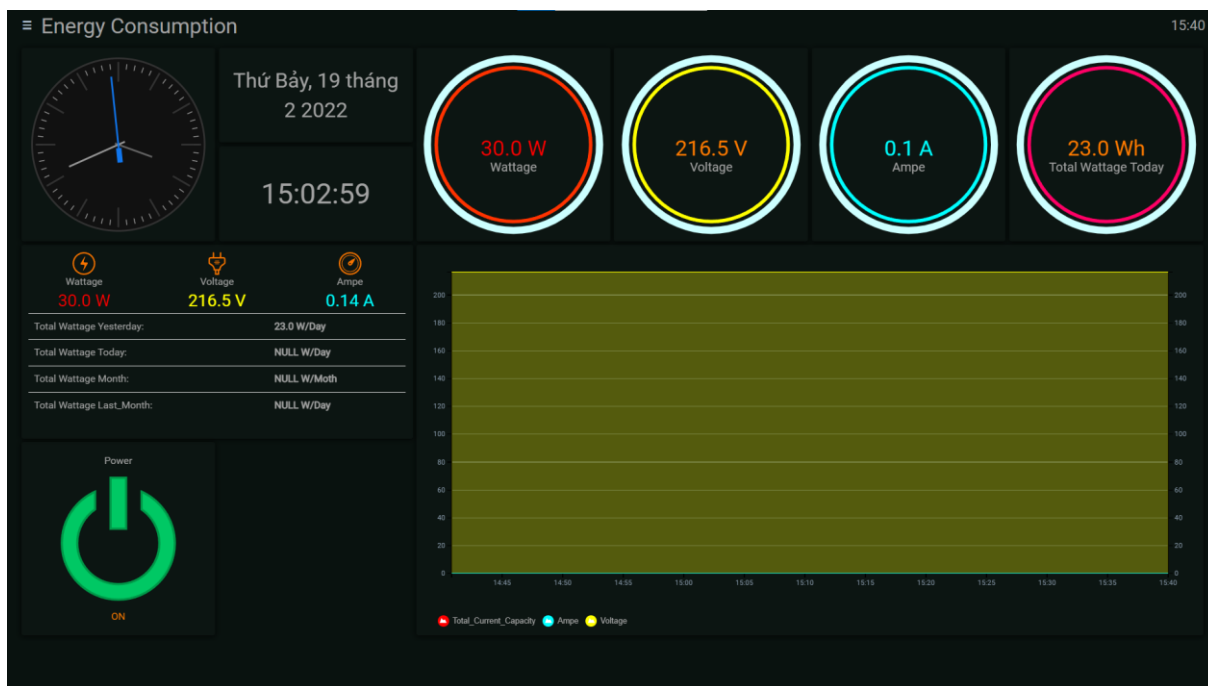
Hình 3. 19: Module giám sát năng lượng



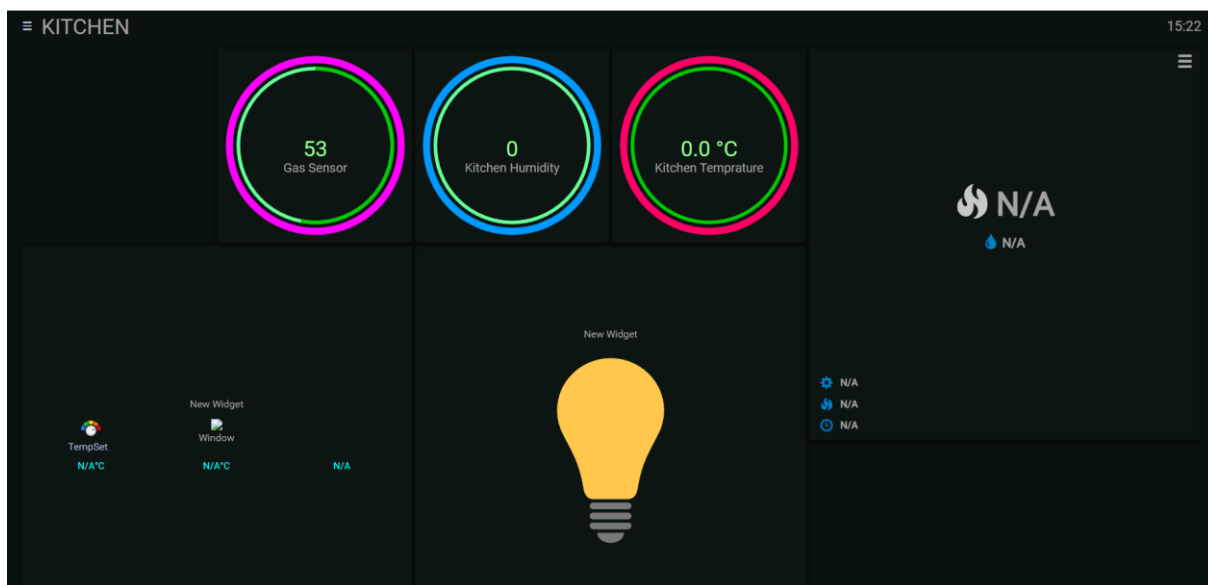
Hình 3. 20: Trang chủ Web giám sát điều khiển



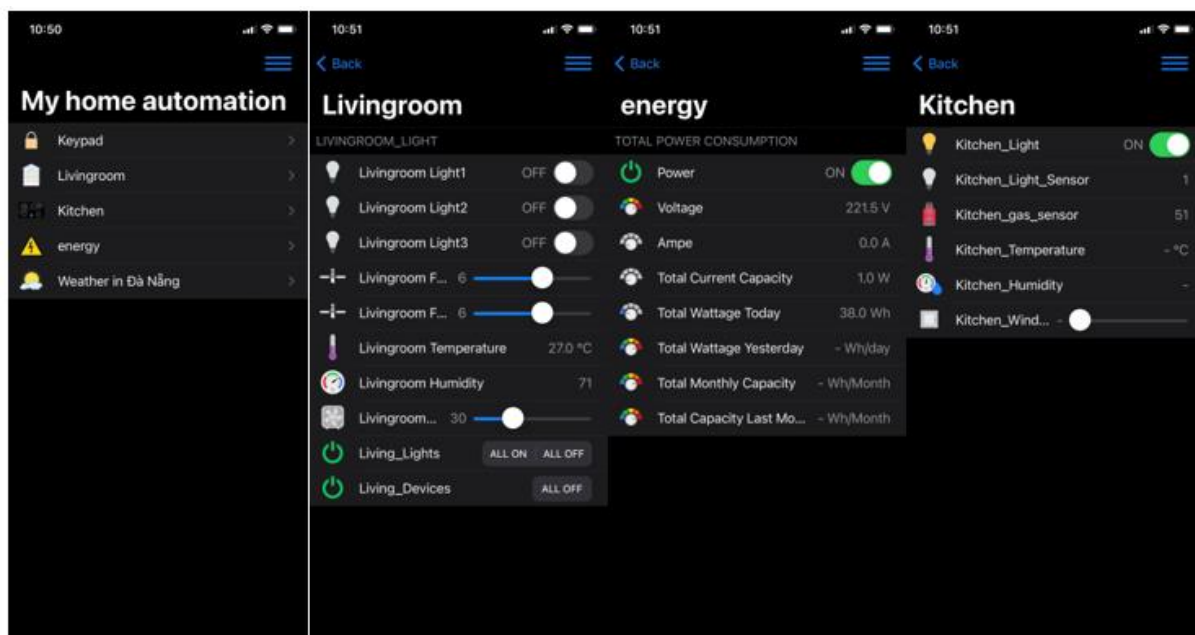
Hình 3. 21: Giao diện trang giám sát điều khiển thiết bị



Hình 3. 22: Giao diện giám sát năng lượng



Hình 3. 23: Giao diện giám sát các chỉ số cảm biến



Hình 3. 24: Giao diện app điện thoại

3.5.4 Nhận xét – Đánh giá

Sau thời gian 15 tuần nghiên cứu và thực hiện đề tài, mạch cơ bản đáp ứng được yêu cầu thiết kế ban đầu. Dưới đây là một số nhận xét:

3.5.4.1 Nhận xét

Ưu điểm:

- Hệ thống hoạt động ổn định
- Hệ thống sử dụng nền tảng Openhab và giao thức MQTT bảo mật TLS/SSL
- Hệ thống có tính ứng dụng cao và cần thiết.
- Phần cứng có tích hợp thêm các mạch bảo vệ và thiết kế chống nhiễu.
- Hệ thống tự động giám sát và hiển thị, khi khí gas rò rỉ sẽ cảnh báo.
- Giao diện thiết kế dễ sử dụng và đẹp mắt.

Hạn chế

- Các cảm biến còn có độ sai số nhất định.
- Chưa được tối ưu về phần cứng và hình thức

3.5.4.2 Đánh giá

Sau quá trình vận hành thử hệ thống, nhóm thực hiện đã thu về những số liệu như bảng sau:

	Web server	App android	Đánh giá	Ghi chú
--	------------	-------------	----------	---------

Điều khiển và giám sát thiết bị (50 lần)	48 lần thành công	48 lần thành công	Đạt	
Giám sát các cảm biến 1 ngày	Hoạt động ổn định	Hoạt động ổn định	Đạt	
Cảnh báo bảo mật	Hoạt động ổn định	Hoạt động ổn định	Đạt	

Bảng 2. 2: Số liệu thực nghiệm

Qua những số liệu bảng trên, nhóm đánh giá hệ thống đạt yêu cầu với những mục tiêu đã đề ra. Mô hình có tính thẩm mỹ, an toàn, bảo mật và dễ sử dụng. Sau thời gian test thử, mạch cho thấy sự ổn định. Tuy nhiên vẫn còn một số hạn chế cần khắc phục nếu muốn đưa vào thực tế đời sống như: tốc độ điều khiển và phản hồi còn chậm, chưa có chức năng nhận dạng chủ nhà, chưa có chức năng báo trộm khi có trộm đột nhập từ trên mái nhà, chưa hệ thống dập lửa, hiện tượng nhiễu do sai sót trong quá trình thi công mạch in và hàn linh kiện.

KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

1. Kết luận

Sau khoảng thời 15 tuần nghiên cứu và tìm hiểu, nhóm đã hoàn thành đồ án và thi công mô hình theo những yêu cầu đã đặt ra ban đầu. Trong quá trình thực hiện, nhóm đã thu được những kết quả nhất định.

- Sản phẩm đạt yêu cầu về giám sát thông số điện năng, trạng thái của các thiết bị thông qua màn hình hiển thị và Internet.
- Sản phẩm đạt yêu cầu về điều khiển bao gồm:
 - + Điều khiển qua nền tảng OpenHab cả trên máy tính và điện thoại: điều khiển và giám sát tất cả các khối
 - + Điều khiển thủ công: dùng nút nhấn khối Master điều khiển chính nó hoặc điều khiển các Slave; điều khiển nút nhấn trực tiếp tại các Slave
- Khi có rò rỉ khí gas thì sẽ thông báo đến người dùng.

2. Hướng phát triển

- Mở rộng số thiết bị cần điều khiển với công suất lớn.
- Mở rộng thêm nhiều Slave để mở rộng mạng lưới giám sát và điều khiển.
- Phát triển thêm hệ thống dập lửa khi phát hiện có cháy xảy ra.
- Thêm camera giám sát ngôi nhà thông qua web hoặc android.
- Phát triển ứng dụng điều khiển thiết bị trên website và điều khiển thiết bị bằng giọng nói.
- Tích hợp thêm các lớp bảo mật khác từ phần cứng đến phần mềm.
- Cải tiến phần cứng dần trở nên nhỏ gọn, ổn định.

TÀI LIỆU THAM KHẢO

1. Sách tham khảo

- [1] Trần Thu Hà – Trương Thị Bích Nga – Nguyễn Thị Lương – Bùi Thị Tuyết Đan – Phù Thị Ngọc Hiếu – Dương Thị Cẩm Tú, Giáo trình *Điện tử cơ bản*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2013.
- [2] Nguyễn Đình Phú, Giáo trình *Vi điều khiển*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh
- [3] Nguyễn Đình Phú – Nguyễn Trường Duy, Giáo trình *Kỹ thuật số*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2013
- [4] Hoàng Ngọc Văn, Giáo trình *Điện tử công suất*, Trường ĐH Sư Phạm Kỹ Thuật TP. Hồ Chí Minh, 2007
- [5] Nguyễn Văn Hiệp, Giáo trình *Công nghệ nhận dạng bằng sóng vô tuyến*, Nhà xuất bản đại học quốc gia TP. Hồ Chí Minh, 2014.

2. Bài báo, tạp chí tham khảo

- [6] Nguyễn Quốc Bảo, Báo cáo khoa học: *Ngôi nhà thông minh*

3. Đồ án tốt nghiệp, luận án tham khảo

- [7] Võ Hồng Sơn – Nguyễn Lâm Hoàng Minh Tuấn, Đồ án tốt nghiệp: Thiết kế và thi công hệ thống giám sát điện năng tiêu thụ kết hợp với điều khiển thiết bị điện tử xa thông qua Internet và Lora
- [8] Huỳnh Xuân Dũng – Trần Nhật Minh, Đồ án tốt nghiệp: *Hệ thống IoT điều khiển và giám sát ngôi nhà*, 2018.
- [9] Lê Văn Hiếu, Luận văn tốt nghiệp: *Hệ thống điều khiển nhà thông minh*

4. Tài liệu tham khảo từ Internet

- [10] Github.com
- [11] <https://www.openhab.org/>
- [12] <https://hshop.vn/>
- [13] Arduino.vn
- [14] “Thị trường công nghệ nhà thông minh tại Việt Nam hiện nay”. Internet: <https://tinhte.vn/threads/thi-truong-cong-nghe-nha-thong-minh-tai-viet-namhien-nay.2676688/>
- [15] Phạm vân, Bài báo: *Cuộc sống thông minh sẽ trở thành xu hướng ở Việt Nam*

PHỤ LỤC

Chương trình ESP8266

```
/*-----LCD-I2C_DS1307-----  
-----*/
```

```
#include <Wire.h>
```

```
#include "RTClib.h"
```

```
#include <LiquidCrystal_I2C.h>
```

```
RTC_DS1307 rtc;
```

```
LiquidCrystal_I2C lcd(0x27, 20, 4);
```

```
char Time[20], Date[20];
```

```
void printDateTime();
```

```
byte flag = 1;
```

```
boolean displayMainMenu = 0;
```

```
boolean clearLcd = 0;
```

```
/*-----MQTT-Wifi-----  
-----*/
```

```
#include <ESP8266WiFi.h>
```

```
#include <PubSubClient.h>
```

```
#define mqtt_server "192.168.1.8"
```

```
#define mqtt_topic_sub1 "cmdn/livingRoom1"
```

```
#define mqtt_topic_pub "homeAutomatic/energy"
```

```
#define mqtt_user "Client1"
```

```
#define mqtt_password "123456"
```

```
const uint16_t mqtt_port = 8883;
```

```
WiFiClient espClient;
```

```
PubSubClient client(espClient);
```

```
void setup_wifi();
```

```
void callback(char* topic, byte* payload, unsigned int length) ;
```

```
void setupMQTT();
```

```
void loopMQTT();
```

```
char buffer[10];
```

```
/*-----Json-----  
--*/
```

```
#include <ArduinoJson.h>
```

```
StaticJsonBuffer<100> JSONbuffer;
```

```
JsonObject& json = JSONbuffer.createObject();
```

```
char JSONmessageBuffer[100];
```

```
/*-----update-time-----  
-----*/
```

```
#include <WiFiUdp.h>
```

```
#include <NTPClient.h> // include NTPClient library
```

```
#include <TimeLib.h> // include Arduino time library
```

```
WiFiUDP ntpUDP;
```

```
// 'time.nist.gov' is used (default server) with +1 hour offset (3600 seconds) 60 seconds  
(60000 milliseconds) update interval
```

```
NTPClient timeClient(ntpUDP, "time.nist.gov", 7 * 3600, 60000);
```

```
byte last_second, second_, minute_, hour_, day_, month_, year_;
```

```
void SNYCTime();
```

```
void setupTime();
```

```
/*-----Pzem004T-----  
-----*/
```

```
#include <EEPROM.h>
```

```
#include <SoftwareSerial.h>
#include <PZEM004T.h>
PZEM004T pzem(D6, D5); // (RX,TX) connect to TX,RX of PZEM
IPAddress ip(192, 168, 1, 1);

#define button D3
#define led D4
float v = 0;
float i = 0;
float p = 0;
float e = 0;

float csHomnay, csHomqua, csThangnay, csThangtruoc, tongHomqua;

int ngayChotSo = 21;
int gioChotSo = 13, phutChotSo = 33;
boolean resetE = false;
unsigned long times = millis();
unsigned long timeSaveData;

void readChiSo();
void writeChiSo();
void setupPzem();
/*-----*/
-----*/

#include "syncTime.h"
#include "pzem004t.h"
#include "mqtt.h"
#include<Ticker.h>
Ticker blynkLed;
```

```
void checkConnectWifi()
{
  if (WiFi.status() != WL_CONNECTED) // check WiFi connection status
    digitalWrite(led, !digitalRead(led));
  else
    digitalWrite(led, 1);
}
```

```
void setup ()
{
  Serial.begin(115200);
  pinMode(D7,OUTPUT);
  lcd.begin();
  lcd.backlight();
  if (! rtc.begin()) {
    Serial.println("Couldn't find RTC");
  }

  if (! rtc.isrunning()) {
    Serial.println("RTC is NOT running, let's set the time!");
  }

  setup_wifi();
  lcd.clear();
  setupTime();
  setupMQTT();
  setupPzem();
  blynkLed.attach(0.7, checkConnectWifi);
}
```

```
void loop ()
```

```
{
loopMQTT();

if (millis() - times > 5000) {
    times = millis();

    readPzem();

    if (WiFi.status() == WL_CONNECTED) // check WiFi connection status
    {
        json["voltage"] = v;
        json["ampe"] = i;
        json["power"] = p;
        json["enegry"] = e;
        json["homNay"] = csHomnay;

        json.printTo(JSONmessageBuffer, sizeof(JSONmessageBuffer));
        client.publish(mqtt_topic_pub, JSONmessageBuffer);
    }
    //
    // sprintf(buffer, "%.1f", v);
    // client.publish("homeAutomatic/energy/voltage", buffer);
    // sprintf(buffer, "%.1f", i);
    // client.publish("homeAutomatic/energy/current", buffer);
    // sprintf(buffer, "%.1f", p);
    // client.publish("homeAutomatic/energy/power", buffer);
    // sprintf(buffer, "%.1f", e);
    // client.publish("homeAutomatic/energy/enegry", buffer);
    // sprintf(buffer, "%.1f", csHomnay);
    // client.publish("homeAutomatic/energy/homNay", buffer);
}
```



```
}
```

```
if (millis() - timeSaveData > 60000) {
```

```
    timeSaveData = millis();
```

```
    writeChiSo();
```

```
    Serial.println("check write");
```

```
}
```

```
printLCD();
```

```
}
```

```
void printDateTime()
```

```
{
```

```
    DateTime now = rtc.now();
```

```
    sprintf(Time, "Time : %02d:%02d:%02d", now.hour(), now.minute(), now.second());
```

```
    sprintf(Date, "Date : %02d-%02d-%02d", now.day(), now.month(), now.year());
```

```
    lcd.setCursor(0, 2);
```

```
    lcd.print(Time);
```

```
    lcd.setCursor(0, 3);
```

```
    lcd.print(Date);
```

```
    // sprintf(Time, "Time : %02d:%02d:%02d", now.hour(), now.minute(),  
now.second());
```

```
    // sprintf(Date, "Date : %02d-%02d-%02d", now.day(), now.month(), now.year());
```

```
    // Serial.println(Time);
```

```
    // Serial.println(Date);
```

```
}
```

```
void mainMenu()
```

```
{
```

```
    lcd.setCursor(0, 0);
```

```
    lcd.print("V: " + String(v) + "V");
```

```
lcd.setCursor(11, 0);
lcd.print("I: " + String(i) + "A");
lcd.setCursor(0, 1);
lcd.print("P: " + String(p) + "W");
lcd.setCursor(11, 1);
lcd.print("F: 50 Hz");
printDateTime();
}
void printLCD()
{
  if (clearLcd == 1)
  {
    clearLcd = 0;
    lcd.clear();
  }
  if (flag == 1)
  {
    mainMenu();
  }
  else if (flag == 2) {
    lcd.setCursor(0, 0);
    lcd.print("Today  : " + String(csHomnay) + "Wh");
    lcd.setCursor(0, 1);
    lcd.print("Tomorrow: " + String(csHomqua) + "Wh");
    lcd.setCursor(0, 2);
    lcd.print("Month   : " + String(csThangnay) + "Wh");
    lcd.setCursor(0, 3);
    lcd.print("Last month: " + String(csThangtruoc) + "Wh");
  }
  else if (flag == 3)
```

```
{  
  lcd.setCursor(4, 0);  
  lcd.print("Latch time ");  
  lcd.setCursor(0, 1);  
  lcd.print("Hour : " + String(gioChotSo));  
  lcd.setCursor(0, 2);  
  lcd.print("Minute: " + String(phutChotSo));  
  lcd.setCursor(0, 3);  
  lcd.print("Day  : " + String(ngayChotSo));  
}  
}
```

```
/*-----MQTT-Wifi-----*/  
-----*/
```

```
void setup_wifi() {  
  Serial.println();  
  Serial.print("Connecting to ");  
  Serial.println(ssid);  
  WiFi.begin(ssid, password);  
  int i = 0;  
  lcd.setCursor(0, 0);  
  lcd.print("Connecting.");  
  while (WiFi.status() != WL_CONNECTED) {  
    delay(500);  
    ++i; if (i == 10) return ;  
    Serial.print(".");  
    lcd.print(".");  
  }  
  Serial.println("IP address: ");  
  Serial.println(WiFi.localIP());  
}
```

```
SNYCTime());
}

void callback(char* topic, byte* payload, unsigned int length) {
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  String p = "";
  for (int i = 0; i < length; i++)
  {
    p += (char)payload[i];
  }

  Serial.print(p);

  if (p == "reset")
  {
    resetPzem() ;
  }
  else Serial.println(" Khong hop le");
  Serial.println();
}

void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    // Attempt to connect
    if (client.connect("ESP8266Client", mqtt_user, mqtt_password)) {
```

```
Serial.println("connected");

client.subscribe(mqtt_topic_sub1);

} else {
  Serial.print("failed, rc=");
  Serial.print(client.state());
  Serial.println(" try again in 5 seconds");
  // Wait 5 seconds before retrying
  delay(1000);
}
}
}

void setupMQTT()
{
  client.setServer(mqtt_server, mqtt_port);
  client.setCallback(callback);
}

void loopMQTT()
{
  if (WiFi.status() == WL_CONNECTED) // check WiFi connection status
  {
    if (!client.connected()) { // no connect -> connect mqtt and sync time
      reconnect();
      SNYCTime();
    }
    client.loop();
  }
}
```

```
}
/*-----Pzem004T-----
-----*/

void readPzem() {
  v = pzem.voltage(ip);
  i = pzem.current(ip);
  p = pzem.power(ip);
  e = pzem.energy(ip);
  csHomnay = e - tongHomqua;

  if (v < 0.0)
  {
    v = 0.0;
    // Serial.print(v);
    // Serial.print("V; ");
  }

  if (i < 0.0) {
    i = 0.0;
    // Serial.print(i);
    // Serial.print("A; ");
  }

  if (p < 0.0) {
    p = 0.0;
    // Serial.print(p);
    // Serial.print("W; ");
  }

  // if (e < 0.0) {
  /// Serial.print(e);
  /// Serial.println("Wh; ");
}
```

```
// }
// Serial.print("hom nay: ");
// Serial.println(csHomnay);
// Serial.print("hom qua: ");
// Serial.println(csHomqua);
// Serial.print("thang nay: ");
// Serial.println(csThangnay);
// Serial.print("thang truoc: ");
// Serial.println(csThangtruoc);
// Serial.println();
}
//-----Ghi dữ liệu kiểu float vào bộ nhớ EEPROM-----//
float readFloat(unsigned int addr) {
    union {
        byte b[4];
        float f;
    } data;
    for (int i = 0; i < 4; i++) {
        data.b[i] = EEPROM.read(addr + i);
    }
    return data.f;
}
void writeFloat(unsigned int addr, float x) {
    union {
        byte b[4];
        float f;
    } data;
    data.f = x;
    for (int i = 0; i < 4; i++) {
        EEPROM.write(addr + i, data.b[i]);
    }
}
```

```
}  
}  
void readChiSo() {  
    csHomqua = readFloat(0);  
    delay(10);  
    csThangnay = readFloat(4);    //tháng này  
    delay(10);  
    tongHomqua = readFloat(8);  
    delay(10);  
    csThangtruoc = readFloat(12); // tháng trước  
    delay(10);  
    Serial.print("Chỉ số tổng hôm qua: "); Serial.println(tongHomqua);  
    Serial.print("Chỉ số hôm qua: "); Serial.println(csHomqua);  
    Serial.print("Chỉ số tháng này: "); Serial.println(csThangnay);  
    Serial.print("Chỉ số tháng trước: "); Serial.println(csThangtruoc);  
  
    sprintf(buffer, "%.1f", csHomqua);  
    client.publish("homeAutomatic/energy/homqua", buffer);  
    sprintf(buffer, "%.1f", csThangnay);  
    client.publish("homeAutomatic/energy/thangNay", buffer);  
    sprintf(buffer, "%.1f", csThangtruoc);  
    client.publish("homeAutomatic/energy/thangTruoc", buffer);  
  
}  
  
void writeChiSo() {  
    DateTime now = rtc.now();  
    if (now.hour() == gioChotSo && now.minute() == phutChotSo) {  
        Serial.println("Ghi số ngày mới!");  
    }  
}
```



```
writeFloat(4, readFloat(4) + csHomnay); // cộng chỉ số hôm nay vào chỉ số tháng này
writeFloat(0, csHomnay); // ghi chỉ số hôm nay thành ngày hôm qua
writeFloat(8, e);
if (now.day() == ngayChotSo) {
    Serial.println("Ghi số tháng mới!");
    writeFloat(12, csThangnay); // ghi chỉ số tháng này vào tháng trước
}
EEPROM.commit();
readChiSo();
}
}
```

```
void resetPzem() {
    for (int i = 0; i < 16; ++i) {
        EEPROM.write(i, 0);
        delay(10);
        digitalWrite(led, !digitalRead(led));
    }
    EEPROM.commit();
    readChiSo();
    Serial.println("Reset energy!");
}
```

```
ICACHE_RAM_ATTR void handleButton() {
    delay(60);
    ++flag;
    clearLcd = 1;
    if (flag == 4) flag = 1;
    // while (digitalRead(button) == 0);
}
```

```
}

void setupPzem()
{
  pinMode(button, INPUT_PULLUP);
  pinMode(led, OUTPUT);
  digitalWrite(led, HIGH);
  attachInterrupt(digitalPinToInterrupt(button), handleButton, FALLING);

  EEPROM.begin(512);
  delay(10);
  readChiSo();
}
/*-----update-time-----
-----*/

void setupTime()
{
  timeClient.begin();
  delay(100);
}

void SNYCTime()
{
  if (WiFi.status() == WL_CONNECTED) // check WiFi connection status
  {
    timeClient.update();
    unsigned long unix_epoch = timeClient.getEpochTime(); // get UNIX Epoch time

    second_ = second(unix_epoch); // get seconds from the UNIX Epoch time
    if (last_second != second_) // update time & date every 1 second
```

```

{
    minute_ = minute(unix_epoch);    // get minutes (0 - 59)
    hour_   = hour(unix_epoch);      // get hours   (0 - 23)
    day_    = day(unix_epoch);       // get month day (1 - 31, depends on month)
    month_  = month(unix_epoch);     // get month (1 - 12 with Jan is month 1)
    year_   = year(unix_epoch) - 2000; // get year with 4 digits - 2000 results 2 digits
    year (ex: 2018 --> 18)

    rtc.adjust(DateTime(year_, month_, day_, hour_, minute_, second_));
    last_second = second_;
}
}
}

```

Chương trình OPENHAB:

//Group

```

Group:Switch:OR(ON, OFF) Living_Lights "ALL_Living_Lights [(%d)]"
["Lights"]
{channel="mqtt:topic:MyMQTTBroker:livingRoom:Groups:Living_Lights" }

```

```

Group:Switch:OR(ON, OFF) Living_Devices "ALL_Living_Devices [(%d)]"

```

//LivingRoom

```

Switch Livingroom_Light1      "Livingroom Light1"          <light>
(Living_Lights) ["Lights"]
{channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Light1" }

```

```

Switch Livingroom_Light2      "Livingroom Light2"          <light>
(Living_Lights) ["Lights"]
{channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Light2" }

```

```

Switch Livingroom_Light3      "Livingroom Light3"          <light>
(Living_Lights) ["Lights"]
{channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Light3" }

```

```

Dimmer Livingroom_Fan1        "Livingroom Fan1"             <fan_ceiling>
["Lighting"] {channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Fan1"
}

```

```

Dimmer Livingroom_Fan2        "Livingroom Fan2"             <fan_ceiling>
["Lighting"] {channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Fan2"
}

```

```
Number Livingroom_Temperature    "Livingroom Temperature [% .1f °C]"
<temperature>
{ channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Temperature" }
```

```
Number Livingroom_Humidity      "Livingroom Humidity  [%d %unit%]"
<humidity>
{ channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Humidity" }
```

```
Dimmer Livingroom_Harmonic      "Livingroom_Harmonic"          <fan>
{ channel="mqtt:topic:MyMQTTBroker:livingRoom:Livingroom_Harmonic" }
```

```
String Keypad_Char              "Keypad Char"                  <lock>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Keypad_Char"
  [profile="system:follow"] }
```

//ENERGY

```
Number Voltage                  "Voltage [JSONPATH($.homeautomation/energy):%s v]"
<heating>    { channel="mqtt:topic:MyMQTTBroker:Energy:Voltage" }
```

```
Number Ampe                     "Ampe  [% .1f A]"                <heating>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Ampe" }
```

```
Number Current                  "Total Current Capacity [% .1f W]"    <heating>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Current" }
```

```
Number Today                    "Total Wattage Today [% .1f W/day]"    <heating>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Today" }
```

```
Number Yesterday                "Total Wattage Yesterday [% .1f W/day]" <heating>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Yesterday" }
```

```
Number Month                     "Total Monthly Capacity [% .1f W/Month]" <heating>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Month" }
```

```
Number Last_Month               "Total Capacity Last Month [% .1f W/Month]" <heating>
{ channel="mqtt:topic:MyMQTTBroker:Energy:Last_Month" }
```

```
Switch Power                    "Power"
{ channel="mqtt:topic:MyMQTTBroker:Energy:Power" }
```

```
// Switch Outside_Light "Outside Light"          <light>
{ channel="mqtt:topic:MyMQTTBroker:Lock_The_Door:Outside_Light" }
```

//KITCHEN

```

Number Kitchen_Temperature "Kitchen_Temperature [%d °C]"
<temperature>
{ channel="mqtt:topic:MyMQTTBroker:Kitchen:Kitchen_Temperature" }

Number Kitchen_Humidity "Kitchen_Humidity [%d %unit%]" <humidity>
{ channel="mqtt:topic:MyMQTTBroker:Kitchen:Kitchen_Humidity" }

Number Kitchen_Light_Sensor "Kitchen_Light_Sensor" <light>
{ channel="mqtt:topic:MyMQTTBroker:Kitchen:Kitchen_Light_Sensor" }

Number Kitchen_gas_sensor "Kitchen_gas_sensor [%d ]" <gas>
{ channel="mqtt:topic:MyMQTTBroker:Kitchen:Kitchen_gas_sensor" }

Dimmer Kitchen_Window "Kitchen_Window [MAP(en.map):%s]" <window>
{ channel="mqtt:topic:MyMQTTBroker:Kitchen:Kitchen_Window" }

Switch Kitchen_Light "Kitchen_Light" <light>
{ channel="mqtt:topic:MyMQTTBroker:Kitchen:Kitchen_Light" }

//Wether in Đà Nẵng

// String          localStationId          "ID [%s]"
{ channel="openweathermap:weather-and-forecast:api:local:station#id" }

// String          localStationName        "Name [%s]"
{ channel="openweathermap:weather-and-forecast:api:local:station#name" }

// Location        localStationLocation    "Location [%2$s°N %3$s°E]"
<location> { channel="openweathermap:weather-and-forecast:api:local:station#location" }

// DateTime        localLastMeasurement    "Timestamp of last measurement
[%1$tY-%1$tm-%1$dt%1$tH:%1$tM:%1$tS]" <time> {
channel="openweathermap:weather-and-forecast:api:local:current#time-stamp" }

// String          localCurrentCondition    "Current condition [%s]"
<sun_clouds> { channel="openweathermap:weather-and-forecast:api:local:current#condition" }

// String          localCurrentConditionId  "ConditionId
[%s]" {
channel="openweathermap:weather-and-forecast:api:local:current#condition-id" }

// Image          localCurrentConditionIcon "Icon"
{ channel="openweathermap:weather-and-forecast:api:local:current#icon" }

// String          localCurrentConditionIconId "IconId [%s]"
{
channel="openweathermap:weather-and-forecast:api:local:current#icon-id" }

```

```
// Number:Temperature    localCurrentTemperature    "Current temperature [%].1f
%unit%]"                <temperature> { channel="openweathermap:weather-
and-forecast:api:local:current#temperature" }

// Number:Temperature    localCurrentApparentTemperature "Current apparent
temperature [%].1f %unit%]"                <temperature> {
channel="openweathermap:weather-and-forecast:api:local:current#apparent-
temperature" }

// Number:Pressure       localCurrentPressure        "Current barometric pressure
[%].1f %unit%]"                <pressure> { channel="openweathermap:weather-
and-forecast:api:local:current#pressure" }

// Number:Dimensionless localCurrentHumidity        "Current atmospheric
humidity [%d %unit%]"          <humidity> {
channel="openweathermap:weather-and-forecast:api:local:current#humidity" }

// Number:Speed          localCurrentWindSpeed        "Current wind speed [%].1f
km/h]"                          <wind> { channel="openweathermap:weather-and-
forecast:api:local:current#wind-speed" }

// Number:Angle          localCurrentWindDirection    "Current wind direction [%d
%unit%]"                          <wind> { channel="openweathermap:weather-
and-forecast:api:local:current#wind-direction" }

// Number:Dimensionless localCurrentCloudiness    "Current cloudiness [%d
%unit%]"                          <clouds> { channel="openweathermap:weather-
and-forecast:api:local:current#cloudiness" }

// Number:Length         localCurrentRainVolume        "Current rain volume [%].1f
%unit%]"                          <rain> { channel="openweathermap:weather-and-
forecast:api:local:current#rain" }

// Number:Length         localCurrentSnowVolume        "Current snow volume [%].1f
%unit%]"                          <snow> { channel="openweathermap:weather-
and-forecast:api:local:current#snow" }

// Number:Length         localCurrentVisibility        "Current visibility [%].1f km]"
<visibility> { channel="openweathermap:weather-and-
forecast:api:local:current#visibility" }

// DateTime localDailyForecastTodayTimestamp "Timestamp of forecast [%1$tY-
%1$tm-%1$td]"                <time> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#time-stamp" }

// String localDailyForecastTodayCondition "Condition for today [%s]"
<sun_clouds> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#condition" }
```

```
// Image localDailyForecastTodayConditionIcon "Icon"
{ channel="openweathermap:weather-and-forecast:api:local:forecastToday#icon" }

// Number:Temperature localDailyForecastTodayMinTemperature "Minimum
temperature for today [%d %unit%]" <temperature> {
channel="openweathermap:weather-and-forecast:api:local:forecastToday#min-
temperature" }

// Number:Temperature localDailyForecastTodayMaxTemperature "Maximum
temperature for today [%d %unit%]" <temperature> {
channel="openweathermap:weather-and-forecast:api:local:forecastToday#max-
temperature" }

// Number:Pressure localDailyForecastTodayPressure "Barometric pressure for today
[%d %unit%]" <pressure> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#pressure" }

// Number:Dimensionless localDailyForecastTodayHumidity "Atmospheric humidity
for today [%d %unit%]" <humidity> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#humidity" }

// Number:Speed localDailyForecastTodayWindSpeed "Wind speed for today [%d %unit%]"
<wind> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#wind-speed" }

// Number:Angle localDailyForecastTodayWindDirection "Wind direction for today
[%d %unit%]" <wind> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#wind-direction" }

// Number:Dimensionless localDailyForecastTodayCloudiness "Cloudiness for today
[%d %unit%]" <clouds> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#cloudiness" }

// Number:Length localDailyForecastTodayRainVolume "Rain volume for today [%d %unit%]"
<rain> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#rain" }

// Number:Length localDailyForecastTodaySnowVolume "Snow volume for today
[%d %unit%]" <snow> { channel="openweathermap:weather-and-
forecast:api:local:forecastToday#snow" }
```

// // Open Weather Map

```
Number:Temperature Weather_OWM_Temperature
"Temperature [%d %unit%]" <temperature>
```

```

    { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Temperature" }

Number:Temperature      Weather_OWM_ApparentTemperature
    "Temperature [% .1f %unit%]"      <temperature>
    { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_ApparentTemperature" }

Number:Pressure      Weather_OWM_Pressure      "Pressure [% .1f %unit%]"
<pressure>      { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Pressure" }

Number:Dimensionless      Weather_OWM_Humidity      "Humidity [%d
%unit%]"
    <humidity>
    { channel="openweathermap:weather-and-forecast:local:Weather_OWM_Humidity" }

Number:Speed      Weather_OWM_Wind_Speed
    "Windspeed [% .1f m/s]"      <wind>
    { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Wind_Speed" }

Number:Angle      Weather_OWM_WindDirection      "Current wind direction
[%d %unit%]" <wind>      { channel="penweathermap:weather-and-
forecast:local:Weather_OWM_WindDirection" }

Number:Length      Weather_OWM_RainVolume
    "RainVolume [% .1f %unit%]"      <rain>
    { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_RainVolume" }

Number:Length      Weather_OWM_Visibility      "Visibility [% .1f
km]"      <visibility>      { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Visibility" }

Number:Dimensionless      Weather_OWM_Cloudiness      "Current cloudiness [%d
%unit%]" <cloud>      { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Cloudiness" }

Number:Length      Weather_OWM_SnowVolume      "Current snow volume
[% .1f %unit%]" <snow>      { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Snow" }

String      Weather_OWM_StationId      "ID [%s]"
    { channel="openweathermap:weather-and-forecast:local:Weather_OWM_StationId" }

String      Weather_OWM_Station_Name      "Station Name [%s]"
    { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Station_Name" }

DateTime      Weather_OWM_ObservationTime      "Observation
time [% 1$tY-% 1$tm-% 1$td]"      <time>

```



```

{ channel="openweathermap:weather-and-
forecast:local:Weather_OWM_ObservationTime" }

DateTime Sunrise "Sunrise [%1$ta %1$tR]"
<sunrise> { channel="openweathermap:weather-and-forecast:local:Sunrise " }

DateTime Sunset "Sunset [%1$ta %1$tR]"
<unset> { channel="openweathermap:weather-and-forecast:local:Sunset " }

String Weather_OWM_Condition "Condition [%s]"
<sun_clouds> { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Condition" }

String Weather_OWM_ConditionId "ConditionId
[%s]" { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_ConditionId" }

Image Weather_OWM_ConditionIcon "Icon [%s]"
{ channel="openweathermap:weather-and-
forecast:local:Weather_OWM_ConditionIcon" }

String Weather_OWM_ConditionIconId "IconId [%s]"
{ channel="openweathermap:weather-and-
forecast:local:Weather_OWM_ConditionIconId" }

Location Weather_OWM_Location "Location [%2$s°N %3$s°E]"
<location> { channel="openweathermap:weather-and-
forecast:local:Weather_OWM_Location" }

// // Number:Temperature Weather_OWM_Temp0 "Temperature
[%1f %unit%]" <temperature> { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#Weather_OWM_Temp0" }

// // Number:Temperature Weather_OWM_Temp1 "Temperature
[%1f %unit%]" <temperature> { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#Weather_OWM_Temp1" }

// // Number:Temperature Weather_OWM_Temp2 "Temperature
[%1f %unit%]" <temperature> { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours48#Weather_OWM_Temp2" }

// // Number:Temperature Weather_OWM_Temp3 "Temperature
[%1f %unit%]" <temperature> { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours72#Weather_OWM_Temp3" }

// // DateTime Weather_OWM_Forecast_Time0 "Forecast time
[%1$td.%1$tm.%1$tY %1$tH:%1$tM]" <time>

```

```

        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#time-stamp" }

// // DateTime      Weather_OWM_Forecast_Time1      "Forecast time
[%1$d.%1$m.%1$Y %1$tH:%1$tM]" <time>
        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#time-stamp" }

// // DateTime      Weather_OWM_Forecast_Time2      "Forecast time
[%1$d.%1$m.%1$Y %1$tH:%1$tM]" <time>
        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours48#time-stamp" }

// // DateTime      Weather_OWM_Forecast_Time3      "Forecast time
[%1$d.%1$m.%1$Y %1$tH:%1$tM]" <time>
        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours72#time-stamp" }

// // String        Weather_OWM_Condition0          "Condition [%s]"
        <sun_clouds>      { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#condition" }

// // String        Weather_OWM_Condition1          "Condition [%s]"
        <sun_clouds>      { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#condition" }

// // String        Weather_OWM_Condition2          "Condition [%s]"
        <sun_clouds>      { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours48#condition" }

// // String        Weather_OWM_Condition3          "Condition [%s]"
        <sun_clouds>      { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours72#condition" }

// // String        Weather_OWM_ConditionId0        "Icon [%s]"
        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#condition-id" }

// // String        Weather_OWM_ConditionId1        "Icon [%s]"
        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#condition-id" }

// // String        Weather_OWM_ConditionId2        "Icon [%s]"
        { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours48#condition-id" }

```

```
// // String          Weather_OWM_ConditionId3          "Icon [%s]"
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours72#condition-id" }

Number:Temperature          Weather_OWM_Temp_h3          "Temperature
[% .1f %unit%]" <temperature>          { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#temperature" }

Number:Temperature          Weather_OWM_Temp_h6          "Temperature
[% .1f %unit%]" <temperature>          { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours06#temperature" }

Number:Temperature          Weather_OWM_Temp_h9          "Temperature
[% .1f %unit%]" <temperature>          { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours09#temperature" }

Number:Temperature          Weather_OWM_Temp_h12         "Temperature [% .1f %unit%]" <temperature>
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours12#temperature" }

Number:Temperature          Weather_OWM_Temp_h15         "Temperature [% .1f %unit%]" <temperature>
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours15#temperature" }

Number:Temperature          Weather_OWM_Temp_h18         "Temperature [% .1f %unit%]" <temperature>
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours18#temperature" }

Number:Temperature          Weather_OWM_Temp_h21         "Temperature [% .1f %unit%]" <temperature>
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours21#temperature" }

Number:Temperature          Weather_OWM_Temp_h24         "Temperature [% .1f %unit%]" <temperature>
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#temperature" }

Number:Temperature          Weather_OWM_Temp_h27         "Temperature [% .1f %unit%]" <temperature>
    { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours27#temperature" }

Number:Temperature          Weather_OWM_Temp_h30         "Temperature [% .1f %unit%]" <temperature>
```

```
{ channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours30#temperature" }  
Number:Temperature          Weather_OWM_Temp_h33  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours33#temperature" }  
Number:Temperature          Weather_OWM_Temp_h36  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours36#temperature" }  
Number:Temperature          Weather_OWM_Temp_h39  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours39#temperature" }  
Number:Temperature          Weather_OWM_Temp_h42  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours42#temperature" }  
Number:Temperature          Weather_OWM_Temp_h45  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours45#temperature" }  
Number:Temperature          Weather_OWM_Temp_h48  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours48#temperature" }  
Number:Temperature          Weather_OWM_Temp_h51  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours51#temperature" }  
Number:Temperature          Weather_OWM_Temp_h54  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours54#temperature" }  
Number:Temperature          Weather_OWM_Temp_h57  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours57#temperature" }  
Number:Temperature          Weather_OWM_Temp_h60  
    "Temperature [% .1f %unit%]" <temperature>
```

```
{ channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours60#temperature" }  
Number:Temperature          Weather_OWM_Temp_h63  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours63#temperature" }  
Number:Temperature          Weather_OWM_Temp_h66  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours66#temperature" }  
Number:Temperature          Weather_OWM_Temp_h69  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours69#temperature" }  
Number:Temperature          Weather_OWM_Temp_h72  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours72#temperature" }  
Number:Temperature          Weather_OWM_Temp_h75  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours75#temperature" }  
Number:Temperature          Weather_OWM_Temp_h78  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours78#temperature" }  
Number:Temperature          Weather_OWM_Temp_h81  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours81#temperature" }  
Number:Temperature          Weather_OWM_Temp_h84  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours84#temperature" }  
Number:Temperature          Weather_OWM_Temp_h87  
    "Temperature [% .1f %unit%]" <temperature>  
    { channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours87#temperature" }  
Number:Temperature          Weather_OWM_Temp_h90  
    "Temperature [% .1f %unit%]" <temperature>
```

```
{channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours90#temperature" }
```

```
Number:Temperature          Weather_OWM_Temp_h93  
    "Temperature [% .1f %unit%]" <temperature>  
    {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours93#temperature" }
```

```
Number:Temperature          Weather_OWM_Temp_h96  
    "Temperature [% .1f %unit%]" <temperature>  
    {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours96#temperature" }
```

```
String      Weather_OWM_Condition_h3      "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours03#condition" }
```

```
String      Weather_OWM_Condition_h6      "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours06#condition" }
```

```
String      Weather_OWM_Condition_h9      "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours09#condition" }
```

```
String      Weather_OWM_Condition_h12     "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours12#condition" }
```

```
String      Weather_OWM_Condition_h15     "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours15#condition" }
```

```
String      Weather_OWM_Condition_h18     "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours18#condition" }
```

```
String      Weather_OWM_Condition_h21     "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours21#condition" }
```

```
String      Weather_OWM_Condition_h24     "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours24#condition" }
```

```
String      Weather_OWM_Condition_h27     "Condition [%s]"  
    <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours27#condition" }
```

String Weather_OWM_Condition_h30 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours30#condition"}

String Weather_OWM_Condition_h33 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours33#condition"}

String Weather_OWM_Condition_h36 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours36#condition"}

String Weather_OWM_Condition_h39 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours39#condition"}

String Weather_OWM_Condition_h42 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours42#condition"}

String Weather_OWM_Condition_h45 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours45#condition"}

String Weather_OWM_Condition_h48 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours48#condition"}

String Weather_OWM_Condition_h51 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours51#condition"}

String Weather_OWM_Condition_h54 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours54#condition"}

String Weather_OWM_Condition_h57 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours57#condition"}

String Weather_OWM_Condition_h60 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours60#condition"}

String Weather_OWM_Condition_h63 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours63#condition"}

String Weather_OWM_Condition_h66 "Condition [%s]"
 <sun_clouds> {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours66#condition"}

```
String      Weather_OWM_Condition_h69      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours69#condition" }

String      Weather_OWM_Condition_h72      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours72#condition" }

String      Weather_OWM_Condition_h75      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours75#condition" }

String      Weather_OWM_Condition_h78      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours78#condition" }

String      Weather_OWM_Condition_h81      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours81#condition" }

String      Weather_OWM_Condition_h84      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours84#condition" }

String      Weather_OWM_Condition_h87      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours87#condition" }

String      Weather_OWM_Condition_h90      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours90#condition" }

String      Weather_OWM_Condition_h93      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours93#condition" }

String      Weather_OWM_Condition_h96      "Condition [%s]"
    <sun_clouds> {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours96#condition" }

String      Weather_OWM_ConditionId_h3      "Condition [%s]"
    {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#condition-id" }

String      Weather_OWM_ConditionId_h6      "Condition [%s]"
    {channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours06#condition-id" }
```



```
String      Weather_OWM_ConditionId_h9      "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours09#condition-id" }

String      Weather_OWM_ConditionId_h12     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours12#condition-id" }

String      Weather_OWM_ConditionId_h15     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours15#condition-id" }

String      Weather_OWM_ConditionId_h18     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours18#condition-id" }

String      Weather_OWM_ConditionId_h21     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours21#condition-id" }

String      Weather_OWM_ConditionId_h24     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#condition-id" }

String      Weather_OWM_ConditionId_h27     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours27#condition-id" }

String      Weather_OWM_ConditionId_h30     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours30#condition-id" }

String      Weather_OWM_ConditionId_h33     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours33#condition-id" }

String      Weather_OWM_ConditionId_h36     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours36#condition-id" }

String      Weather_OWM_ConditionId_h39     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours39#condition-id" }

String      Weather_OWM_ConditionId_h42     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours42#condition-id" }

String      Weather_OWM_ConditionId_h45     "Condition [%s]"
            { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours45#condition-id" }
```

```
String      Weather_OWM_ConditionId_h48      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours48#condition-id" }

String      Weather_OWM_ConditionId_h51      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours51#condition-id" }

String      Weather_OWM_ConditionId_h54      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours54#condition-id" }

String      Weather_OWM_ConditionId_h57      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours57#condition-id" }

String      Weather_OWM_ConditionId_h60      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours60#condition-id" }

String      Weather_OWM_ConditionId_h63      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours63#condition-id" }

String      Weather_OWM_ConditionId_h66      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours66#condition-id" }

String      Weather_OWM_ConditionId_h69      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours69#condition-id" }

String      Weather_OWM_ConditionId_h72      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours72#condition-id" }

String      Weather_OWM_ConditionId_h75      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours75#condition-id" }

String      Weather_OWM_ConditionId_h78      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours78#condition-id" }

String      Weather_OWM_ConditionId_h81      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours81#condition-id" }

String      Weather_OWM_ConditionId_h84      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours84#condition-id" }
```

```

String      Weather_OWM_ConditionId_h87      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours87#condition-id" }

String      Weather_OWM_ConditionId_h90      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours90#condition-id" }

String      Weather_OWM_ConditionId_h93      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours93#condition-id" }

String      Weather_OWM_ConditionId_h96      "Condition [%s]"
           { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours96#condition-id" }

DateTime    Weather_OWM_DateTime_h3          "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours03#time-stamp" }

DateTime    Weather_OWM_DateTime_h6          "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours06#time-stamp" }

DateTime    Weather_OWM_DateTime_h9          "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours09#time-stamp" }

DateTime    Weather_OWM_DateTime_h12         "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours12#time-stamp" }

DateTime    Weather_OWM_DateTime_h15         "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours15#time-stamp" }

DateTime    Weather_OWM_DateTime_h18         "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours18#time-stamp" }

DateTime    Weather_OWM_DateTime_h21         "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours21#time-stamp" }

DateTime    Weather_OWM_DateTime_h24         "Forecast time
[% 1$tD.% 1$tM.% 1$tY % 1$tH:% 1$tM]" { channel="openweathermap:weather-and-
forecast:2fd1295b:local:forecastHours24#time-stamp" }

```

DateTime Weather_OWM_DateTime_h27 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours27#time-stamp"}

DateTime Weather_OWM_DateTime_h30 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours30#time-stamp"}

DateTime Weather_OWM_DateTime_h33 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours33#time-stamp"}

DateTime Weather_OWM_DateTime_h36 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours36#time-stamp"}

DateTime Weather_OWM_DateTime_h39 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours39#time-stamp"}

DateTime Weather_OWM_DateTime_h42 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours42#time-stamp"}

DateTime Weather_OWM_DateTime_h45 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours45#time-stamp"}

DateTime Weather_OWM_DateTime_h48 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours48#time-stamp"}

DateTime Weather_OWM_DateTime_h51 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours51#time-stamp"}

DateTime Weather_OWM_DateTime_h54 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours54#time-stamp"}

DateTime Weather_OWM_DateTime_h57 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours57#time-stamp"}

DateTime Weather_OWM_DateTime_h60 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours60#time-stamp"}

DateTime Weather_OWM_DateTime_h63 "Forecast time
[%1\$td.%1\$tm.%1\$tY %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours63#time-stamp"}

DateTime Weather_OWM_DateTime_h66 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours66#time-stamp"}

DateTime Weather_OWM_DateTime_h69 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours69#time-stamp"}

DateTime Weather_OWM_DateTime_h72 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours72#time-stamp"}

DateTime Weather_OWM_DateTime_h75 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours75#time-stamp"}

DateTime Weather_OWM_DateTime_h78 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours78#time-stamp"}

DateTime Weather_OWM_DateTime_h81 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours81#time-stamp"}

DateTime Weather_OWM_DateTime_h84 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours84#time-stamp"}

DateTime Weather_OWM_DateTime_h87 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours87#time-stamp"}

DateTime Weather_OWM_DateTime_h90 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours90#time-stamp"}

DateTime Weather_OWM_DateTime_h93 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours93#time-stamp"}

DateTime Weather_OWM_DateTime_h96 "Forecast time
[%1\$d.%1\$m.%1\$Y %1\$tH:%1\$tM]" {channel="openweathermap:weather-and-forecast:2fd1295b:local:forecastHours96#time-stamp"}