

**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:

**NGHIÊN CỨU GIAO THỨC TRUYỀN VÀ BẢO
MẬT TRONG HỆ THỐNG IOT**

Người hướng dẫn:

Ths. TRẦN DUY CHUNG

Sinh viên thực hiện:

VÕ TRUNG KIỆT

Lớp: 18D4

LÊ PHẠM VĨNH THIÊN

Lớp: 18D4

Mã sinh viên:

1811505120329

1811505120242

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:

NGHIÊN CỨU GIAO THỨC TRUYỀN VÀ BẢO MẬT TRONG HỆ THỐNG IOT

Người hướng dẫn:	Ths.Trần Duy Chung
Sinh viên thực hiện:	Võ Trung Kiệt Lê Phạm Vĩnh Thiên
Mã sinh viên:	1811505120329 1811505120242
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: LÊ PHẠM VĨNH THIÊN VÕ TRUNG KIỆT
2. Lớp: 18D4 Mã SV: 1811505120242 1811505120329
3. Tên đề tài: NGHIÊN CỨU GIAO THỨC TRUYỀN VÀ BẢO MẬT TRONG MÔ HÌNH IOT
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:

Trong khuôn khổ đề tài này, nhóm tác giả sẽ đề xuất chủ yếu về hai Phương thức truyền là MQTT và HTTP, đi kèm với đó là một trang web được thiết kế riêng theo từng khối module được mã hoá bảo mật tối ưu dữ liệu cho khách hàng khi sử dụng sản phẩm.

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:

- + Đề tài sử dụng mô các module (module_esp8266) để điều khiển và thu thập dữ liệu.
- + Sử dụng giao thức truyền dẫn và đồng bộ hoá theo thời gian.
- + Sử dụng cảm biến độ ẩm, nhiệt độ, động cơ để thu thập giữ liệu.
- + Sử dụng thẻ RFID để điều khiển động cơ.
- + Giao tiếp giữa esp8266 và các thiết bị ngoại vi (Arduino Uno...).
- + Sử dụng Board mạch module_esp8266 cho phép kết nối máy tính thông qua các giao thức, cổng kết nối giao tiếp SPI.
- + Sử dụng mạng wifi để giám sát và thu thập dữ liệu.
- + Hiện thị dữ liệu và thông tin cho người dùng qua màn hình LCD và các thiết bị trong mạng cục bộ.

- + Sử dụng giao thức truyền Message Queuing Telemetry Transport (MQTT).
Hypertext Transfer Protocol (HTTP) và Arduino.

- + Tạo 1 trang web mang tính cục bộ trên node rep và web sever xampp

Tài liệu:

- + Tìm hiểu thông tin dựa trên các website và 1 số tài liệu.

- + Tra cứu thông tin linh kiện.

- + Tài liệu về công nghệ IOT.

- + Tài liệu về Esp 8266

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:

- + Tiếp cận, tìm hiểu, hoàn thiện sản phẩm

- + Giao tiếp thành công thiết bị dung MQTT & HTTP

- + Mã hoá bảo mật dữ liệu của từng module riêng biệt.

- + Hiểu biết cơ bản về lập trình PHP và cách xây dựng cơ sở dữ liệu trên và sql
phpMyadmin.

- + Ứng dụng sản phẩm dễ dàng, dễ tiếp cận, dễ sử dụng

- + Nâng cao khả năng nghiên cứu, học hỏi vào phát triển

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

- + Giao diện còn quá đơn giản chưa thân thiện với người dùng, cần viết app cho
giao diện.

- + Sử dụng module RFID RC522 để đọc tương đối chính xác nhưng vẫn còn chậm,
chưa thể đáp ứng với số lượng nhiều người liên tục.

- + Truyền nhận qua Node-red vẫn còn nhiều hạn chế về phần xử lý luồng dữ liệu

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 đề 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: **Nghiên cứu giao thức truyền và bảo mật trong hệ thống IOT**

Sinh viên thực hiện: **Võ Trung Kiệt** Msv: **1811505120329** Lớp: **18D4**

Lê Phạm Vĩnh Thiên Msv: **1811505120242** Lớp: **18D4**

Nhóm thực hiện đề tài: “Nghiên cứu giao thức truyền và bảo mật trong hệ thống IOT” sử dụng 2 phương thức truyền gồm có MQTT và HTTP để giao tiếp với khối xử lý trung tâm và nhận dữ liệu từ các module sau đó gửi về web sever để lưu trữ mà bảo mật dữ liệu.

Kết quả sau khi hoàn thành hệ thống sẽ giám sát và điều khiển được các thiết bị từ xa, tiếp nhận và xử lý thông tin chính xác và bảo mật. Tiếp cận được các cách giao tiếp cũng như bảo mật trong kết nối ngoại vi nói riêng và hệ thống IOT thông minh nói chung.

NHIỆM VỤ ĐỒ ÁN TỐT NGHIỆP

Giảng viên hướng dẫn: TRẦN DUY CHUNG

Sinh viên thực hiện: LÊ PHẠM VĨNH THIÊN Mã SV: 1811505120242

VÕ TRUNG KIỆT Mã SV: 1811505120329

1. Tên đề tài:

Nghiên cứu giao thức truyền và bảo mật trong mô hình IOT

2. Các số liệu, tài liệu ban đầu:

Số liệu:

- Đề tài sử dụng mô các module (module_esp8266) để điều khiển và thu thập dữ liệu
- Sử dụng giao thức truyền dẫn và đồng bộ hoá theo thời gian
- Sử dụng cảm biến độ ẩm, nhiệt độ, động cơ để thu thập giữ liệu
- Sử dụng thẻ RFID để điều khiển động cơ
- Giao tiếp giữa esp8266 và các thiết bị ngoại vi (Arduino Uno...)
- Sử dụng Board mạch module_esp8266 cho phép kết nối máy tính thông qua các giao thức, cổng kết nối giao tiếp SPI
- Sử dụng mạng wifi để giám sát và thu thập dữ liệu
- Hiện thị dữ liệu và thông tin cho người dùng qua màn hình LCD và các thiết bị trong mạng cục bộ
- Sử dụng giao thức truyền Message Queuing Telemetry Transport (MQTT). Hypertext Transfer Protocol (HTTP) và Arduino
- Tạo 1 trang web mang tính cục bộ trên node rep và web sever xampp

Tài liệu:

- Tìm hiểu thông tin dựa trên các website và 1 số tài liệu
- Tra cứu thông tin linh kiện
- Tài liệu về công nghệ IOT
- Tài liệu về Esp 8266

3. Nội dung chính của đồ án:

Nghiên cứu giao thức truyền giữa các module từ đó có thể điều khiển và thu thập dữ liệu từ các thiết bị đưa lên một trang web sử dụng cục bộ trong hệ thống mạng wifi từ máy chủ mang tính bảo mật thông tin.

- 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 về IOT**

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: Tính toán và thiết kế hệ thống**

Giới thiệu tổng quan về các yêu cầu của đề tài mà mình thiết kế 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 để giải thuật cho chương trình và viết chương trình cho hệ thống.

Hướng dẫn quy trình sử dụng hệ thống

Xây dựng trang web bảo mật riêng tạo giao diện vận hành đơn giản tối ưu các vấn đề.

- **Chương 4: Thi công hệ thống và kết quả**

Hoàn thành từng bộ phận của mô hình, làm mạch in thủ công, lắp ráp từng bộ phận hoàn thiện đồ án.

Kết quả đạt được là hoàn thiện mô hình hoạt động ổn định qua nhiều lần thử nghiệm.

Nhận xét và đánh giá, hướng phát triển

- Hoàn thành các mục tiêu đề ra
- Làm chủ công nghệ và nội địa hóa sản phẩm.
- Giá thành thấp hơn các sản phẩm cùng loại.
- Làm thiết bị học tập và ứng dụng trong cuộc sống sinh hoạt

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 đồ án tốt nghiệp

5. Ngày giao đồ án: 11/10/2021

6. Ngày nộp đồ án: 18/2/2022

Đà Nẵng, ngày 11 tháng 10 năm 2021

Trưởng Bộ môn

Người hướng dẫn

ĐỀ CƯƠNG ĐỒ ÁN TỐT NGHIỆP

1. Họ và tên sinh viên: Lê Phạm Vĩnh Thiên
Võ Trung Kiệt

2. Mã sinh viên: 1811505120242 - 1811505120329 **Lớp:** 18D4

3. Họ và tên người hướng dẫn: Trần Duy Chung

4. Đề tài

NGHIÊN CỨU GIAO THỨC TRUYỀN VÀ BẢO MẬT TRONG MÔ HÌNH IOT

Thời gian thực hiện: Từ ngày: 11/10/2021 đến ngày: 18/2/2022

5. Mục tiêu

Đề tài “**NGHIÊN CỨU GIAO THỨC TRUYỀN VÀ BẢO MẬT TRONG MÔ HÌNH IOT**” bước đầu cơ bản để tạo nền tảng cho việc làm quen với cách phương thức giao tiếp các thiết bị ngoại vi bằng nhiều giao thức truyền khác nhau, hiểu được cách vận hành và lưu trữ dữ liệu, tạo nên 1 hệ thống bảo mật cục bộ thông tin từ các thiết bị đưa về trang web. Sau khi hoàn thành sản phẩm đạt đc:

- Biết cách lập trình Arduino, biết được cách thức giám sát và điều khiển động cơ, cảm biến và nhiều thiết bị khác.
- Thiết kế được trang web riêng mang tính bảo mật tương đối lưu trữ dữ liệu.
- Giao tiếp thành công giữa các thiết bị ngoại vi qua các giao thức truyền như MQTT, HTTP, Arduino.
- Thiết kế được giao diện phù hợp với mục đích sử dụng
- Lắp ráp và vận hành được sản phẩm theo yêu cầu đặt ra
- Phát triển thêm được kỹ năng làm việc, tìm kiếm tài liệu, nghiên cứu và vận dụng.

6. Nội dung chính

- Thiết kế giao diện để giám sát và điều khiển thiết bị trên máy tính và di động

- Viết chương trình giao tiếp giữa các module để điều khiển các thiết bị điện.
- Tạo chương trình bảo mật cho hệ thống.
- Thi công mô hình.
- Viết báo cáo

7. Kết quả dự kiến đạt được

- Viết chương trình giao tiếp giữa các module để điều khiển các thiết bị điện.
- Viết báo cáo tìm hiểu và ứng dụng được các module
- Tìm hiểu về lập trình web, xây dựng giao diện website
- Thiết kế được mạch ứng dụng từ Relay điều khiển các thiết bị
- Thiết kế được giao diện trên điện thoại, máy tính để điều khiển, giám sát thiết bị.
- Tính toán, thiết kế, chọn linh kiện để hoàn thiện được board mạch chính đủ dòng áp chạy đáp ứng được yêu cầu đề tài
- Nghiên cứu và ứng dụng tốt các giao thức truyền.
- Giám sát được các giá trị nhiệt độ, độ ẩm, dòng điện, đọc thẻ RFID máy tính và di động.
- Bảo mật dữ liệu thông tin hệ thống
- Lập trình cho hệ thống hoạt động ổn định qua nhiều lần kiểm tra.
- Mô hình hoạt động ổn định.

8. Tiến độ thực hiện

TT	Thời gian	Nội dung công việc	Kết quả dự kiến đạt được
1	Tuần 1	Gặp giảng viên hướng dẫn và trao đổi về đề tài đồ án tốt nghiệp.	
2	Tuần 2	Viết đề cương và lịch trình thực hiện đồ án tốt nghiệp.	
3	Tuần 3	Tìm hiểu đề tài và lựa chọn thiết bị.	
4	Tuần 4	Tìm hiểu nguyên lý hoạt động của đề tài.	
5	Tuần 5	Thiết kế sơ đồ khối, sơ đồ nguyên lý.	

6	Tuần 6	Viết chương trình, kiểm tra các chức năng điều khiển trên máy tính, điện thoại.	
7	Tuần 7	Viết chương trình, kiểm tra các chức năng điều khiển trên máy tính, điện thoại.	
8	Tuần 8	Thi công phần cứng, lắp ráp mô hình	
9	Tuần 9	Thi công phần cứng, lắp ráp mô hình	
10	Tuần 10	Thi công phần cứng, lắp ráp mô hình	
11	Tuần 11	Hoàn thiện mô hình, đóng gói hệ thống và chạy thử nghiệm	
12	Tuần 12	Hoàn thiện mô hình, đóng gói hệ thống và chạy thử nghiệm	
13	Tuần 13	Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo.	
14	Tuần 14	Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo.	
15	Tuần 15	Chạy thử nghiệm và cân chỉnh toàn hệ thống. Đánh giá kết quả đạt được, viết báo cáo.	

Đà Nẵng, ngày 11 tháng 10 năm 2022

BỘ MÔN DUYỆT

NGƯỜI HƯỚNG DẪN

SINH VIÊN

Ths. Trần Duy Chung

Võ Trung Kiệt

Lê Phạm Vĩnh Thiên

LỜI NÓI ĐẦU

Để hoàn thành đề tài nghiên cứu này, lời đầu tiên cho chúng em gửi lời cảm ơn đến các quý thầy cô giáo Trường Đại Học Sư Phạm Kỹ Thuật – Đại Học Đà Nẵng và em xin chân thành gửi lời cảm ơn đến các thầy cô trong khoa Điện – Điện Tử đã chỉ dạy tận tình, trang bị cho chúng em những nền tảng và kiến thức chuyên ngành quang trọng, giúp chúng em có cơ sở lý thuyết vững vàng và luôn tạo điều kiện giúp đỡ tốt nhất cho chúng em trong quá trình học tập và nghiên cứu. Đặc biệt, chúng em xin chân thành cảm ơn thầy Trần Duy Chung đã tận tình giúp đỡ, đưa ra những định hướng nghiên cứu cũng như hướng giải quyết các các vấn đề để chúng em thực hiện tốt đề tài. Trong thời gian làm việc với thầy chúng em đã tiếp thu thêm được nhiều kiến thức được chỉ dạy từ thầy, luôn thể hiện mọi thái độ nghiêm túc, hiệu quả và đây cũng là điều rất cần thiết trong quá trình học tập và làm việc sau này đối với chúng em. Mặc dù đã cố gắng hết sức, song do điều kiện thời gian và kinh nghiệm thực tế của nhóm còn ít, cho nên đề tài không thể tránh khỏi thiếu sót. Vì vậy, chúng em rất mong được sự đóng góp ý kiến từ các quý thầy, cô giáo.

Chúng em xin chân thành cảm ơn!

Người thực hiện đề tài

Võ Trung Kiệt – Lê Phạm Vĩnh Thiên

CAM ĐOAN

Đề tài này là do nhóm em tự thực hiện dưới sự hướng dẫn của thầy Trần Duy Chung dựa vào một số tài liệu, từ các kiến thức được học từ thực tế và không sao chép từ tài liệu hay công trình sẵn có nào.

Những số liệu, thông số trong đề án chưa từng được báo hay bảo vệ trước đó. Các hình ảnh, dữ liệu được tham khảo ghi rõ nguồn gốc.

Chúng em cam đoan những lời nói trên là đúng sự thật và hoàn toàn trung thực, nếu có bất kỳ phát giác nào thì nhóm chúng em hoàn toàn chịu trách nhiệm.

Sinh viên thực hiện

Võ Trung Kiệt – Lê Phạm Vĩnh Thiên

MỤC LỤC

Nhận xét của người hướng dẫn
Nhận xét của người phản biện
Tóm tắt
Nhiệm vụ đồ án

	Trang
LỜI NÓI ĐẦU	i
CAM ĐOAN.....	ii
MỤC LỤC	iii
DANH SÁCH CÁC HÌNH VẼ.....	vi
DANH SÁCH CÁC BẢNG.....	viii
DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT.....	ix
MỞ ĐẦU	1
Chương 1: TỔNG QUAN	3
1.1 Định nghĩa.....	3
1.2 Khái niệm về IOT.....	3
1.3 IOT từ góc nhìn kỹ thuật.....	4
1.4 Đặc điểm cơ bản và yêu cầu ở mức cao của một hệ thống IoT.....	6
1.4.1 Đặc tính cơ bản.....	6
1.4.2 Yêu cầu ở mức cao đối với một hệ thống IoT.....	7
Chương 2: CƠ SỞ LÝ THUYẾT VỀ IOT.....	8
2.1 Tổng quan về IoT	8
2.1.1 Application Layer.....	8
2.1.2 Service support and application support layer.....	8
2.1.3 Network layer	9
2.1.4 Device layer	9
2.2 Mạng cảm biến	10
2.3 Phần mềm lập trình cho vi điều khiển	10
2.4 Phần mềm lập trình hiển thị lên Web server	11
2.5 Giao thức truyền nhận trong hệ thống	12
2.5.1 Giao thức MQTT.....	12
2.5.2 Giao thức HTTP.....	14
2.6 Cách thức bảo mật trong hệ thống.....	16

2.6.1	<i>Cập nhật firmware cho esp qua OTA trong mạng nội bộ:</i>	16
2.6.2	<i>Giới thiệu về XAMPP</i>	16
2.6.3	<i>Giới thiệu về phần mềm phpMyAdmin:</i>	17
2.6.4	<i>Giới thiệu về cơ sở dữ liệu MySQL trong phpMyAdmin:</i>	17
2.6.5	<i>Giới thiệu về MQTT Broker Mosquitto</i>	21
2.6.6	<i>Giới thiệu về Node-red</i>	22
Chương 3: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG		23
3.1	Giới thiệu linh kiện	23
3.1.1	<i>Module RFID RC522</i>	23
3.1.2	<i>Thẻ RFID</i>	26
3.1.3	<i>Vi điều khiển NodeMCU ESP8266</i>	27
3.1.4	<i>LCD 16x02</i>	30
3.1.5	<i>Module I2C LCD</i>	32
3.1.6	<i>Chuẩn giao tiếp I2C</i>	32
3.1.7	<i>Cảm biến DHT11</i>	33
3.1.8	<i>Cảm biến hồng ngoại</i>	36
3.1.9	<i>Cảm biến độ ẩm đất</i>	37
3.1.10	<i>Module relay</i>	38
3.1.11	<i>Khởi nguồn</i>	40
3.2	Thiết kế sơ đồ khối trong hệ thống	41
3.2.1	<i>Chức năng của từng khối:</i>	42
3.2.2	<i>Nguyên lý hoạt động:</i>	43
3.3	Sơ đồ nguyên lý từng khối trong hệ thống	44
3.4	Tính toán thiết kế hệ thống	45
Chương 4: THI CÔNG HỆ THỐNG VÀ KẾT QUẢ		47
4.1	Thi công phần cứng	47
4.1.1	<i>Thi công board mạch</i>	47
4.1.2	<i>Thi công mô hình</i>	51
4.2	Lập trình phần mềm	53
4.2.1	<i>Lưu đồ giải thuật cho vi điều khiển</i>	53
4.2.2	<i>Lưu đồ giải thuật phần bảo mật hệ thống</i>	56
4.3	Kết quả thực hiện	58
4.3.1	<i>Kết quả thi công phần cứng</i>	58
4.3.2	<i>Hình ảnh mô phỏng thực tế</i>	58
4.3.3	<i>Kết quả đạt được</i>	61

<i>4.3.4 Hướng dẫn sử dụng phần mềm</i>	62
NHẬN XÉT VÀ ĐÁNH GIÁ, HƯỚNG PHÁT TRIỂN	64
KẾT LUẬN	65
TÀI LIỆU THAM KHẢO	66
PHỤ LỤC	

DANH SÁCH CÁC HÌNH VẼ

Hình 1.1 Kết nối mọi vật	4
Hình 1.2 Hệ thống IoT từ góc nhìn kỹ thuật	4
Hình 1.3 Các loại thiết bị khác nhau và mối quan hệ [2]	5
Hình 2.1 Mô hình IoT	8
Hình 2.2 Biểu tượng phần mềm Arduino	11
Hình 2.3 Mô tả cách thức hoạt động của giao thức MQTT	14
Hình 2.4 Cách thức hoạt động của giao thức HTTP	15
Hình 2.5 Mô tả cách thức cập nhật firmware qua OTA	16
Hình 2.6 Giao diện tổng quan cơ sở dữ liệu PHPMyAdmin	17
Hình 2.7 Tạo một cơ sở dữ liệu	18
Hình 2.8 Database đã được tạo	19
Hình 2.9 Tạo các user kết nối database	19
Hình 2.10 Tạo màn hình tạo account	20
Hình 2.11 Chọn quyền truy cập cho user	21
Hình 2.12 Mô hình đơn giản của “luồng” liên kết các khối lệnh trong NodeRed	22
Hình 3.1 Đầu đọc RFID HF	23
Hình 3.2 Đầu đọc RFID UHF	24
Hình 3.3 Đầu đọc/ghi RFID HF	24
Hình 3.4 Module đọc thẻ MFRC522	25
Hình 3.5 Thẻ RFID	26
Hình 3.6 Hình ảnh thực tế Module ESP8266	28
Hình 3.7 Hình ảnh sơ đồ chân kết nối ESP8266	29
Hình 3.8 Màn hình LCD 16x02	30
Hình 3.9 Module I2C LCD	32
Hình 3.10 Nguyên lý hoạt động của chuẩn giao tiếp I2C	33
Hình 3.11 Cảm biến nhiệt độ DHT11	34
Hình 3.12 Bắt đầu chuẩn bị vào quá trình truyền dữ liệu	35
Hình 3.13 Truyền dữ liệu bit 0	36
Hình 3.14 Truyền dữ liệu bit 1	36
Hình 3.15 Cảm biến hồng ngoại	37
Hình 3.16 Cảm biến độ ẩm đất	38
Hình 3.17 Module relay	39
Hình 3.18 Khối nguồn tổ ong	40
Hình 3.19 Sơ đồ khối hệ thống	41
Hình 3.20 Sơ đồ nguyên lý module 1	44
Hình 3.21 Sơ đồ nguyên lý module 2	44
Hình 3.22 Sơ đồ nguyên lý module 3	45
Hình 4.1 Sơ đồ bố trí linh kiện của module 1	47
Hình 4.2 Mạch in sau khi thiết kế	47
Hình 4.3 Sơ đồ bố trí linh kiện của module 2	48
Hình 4.4 Mạch in sau khi thiết kế	48
Hình 4.5 Sơ đồ bố trí linh kiện của module 3	49
Hình 4.6 Mạch in sau khi thiết kế	49
Hình 4.7 Mô hình khối module 1	51
Hình 4.8 Mô hình khối module 2	52

Hình 4.9 Hình ảnh mô hình khối module 3	52
Hình 4.10 Lưu đồ giải thuật cho vi điều khiển module 1	53
Hình 4.11 Lưu đồ giải thuật cho vi điều khiển module 2.....	54
Hình 4.12 Lưu đồ giải thuật cho vi điều khiển module 3.....	55
Hình 4.13 Lưu đồ giải thuật cho bảo mật phần hiển thị.....	56
Hình 4.14 Lưu đồ giải thuật cho bảo mật phần cứng	57
Hình 4.15 Hình ảnh phần cứng hệ thống.....	58
Hình 4.16 Giao diện trang đăng nhập vào hệ thống	58
Hình 4.17 Giao diện trang đăng kí vào hệ thống.....	59
Hình 4.18 Giao diện trang chủ người dùng	59
Hình 4.19 Giao diện trang chủ module 1	59
Hình 4.20 Giao diện trang chủ module 2	60
Hình 4.21 Giao diện trang chủ module 3	60
Hình 4.22 Giao diện trang đọc thẻ và hiển thị module 1.....	61
Hình 4.23 Giao diện trang đăng kí và hiển thị module 1	61
Hình 4.24 Giao diện trang hiển thị thông tin người dùng module 1	61

DANH SÁCH CÁC BẢNG

Bảng 3.1 Thông số kỹ thuật của LCD 16X02	31
Bảng 4.1 Linh kiện sử dụng trong module 1	50
Bảng 4.2 Linh kiện sử dụng trong module 2	50
Bảng 4.3 Linh kiện sử dụng trong module 3	50

DANH SÁCH CÁC KÝ HIỆU, CHỮ VIẾT TẮT

GPIO	General Purpose Input Output
RFID	Radio Frequency Identification
IFF	Identify friend or foe
PCA	Principal Component Analysis
MQTT	Message Queuing Telemetry Transport
HTTP	Hyper Text Transfer Protocol
IOT	Internet of Thing
OTA	Over the Air

MỞ ĐẦU

Trong thời kỳ công nghệ kỹ thuật phát triển, nên khoa học kỹ thuật trong nước cũng cần phải bắt kịp xu hướng công nghệ mới. Các mạch điện tử, thiết bị thông minh luôn đổi mới, công nghiệp hoá – tự động hoá ngày càng được phổ biến trong cuộc sống nhằm mục đích đáp ứng nhu cầu của con người. Đòi hỏi đến việc giám sát điều khiển từ xa. Nhưng để đáp ứng được các nhu cầu trên chúng ta cần phải tiếp cận tìm hiểu và học hỏi những công nghệ mới thích hợp hơn, tiện lợi hơn, nên nhóm chúng em đã quyết định chọn đề tài “Nghiên cứu giao thức truyền và bảo mật trong hệ thống IOT”. Tuy là đề tài không còn khá mới nhưng vẫn còn tính cần thiết, là bàn đạp để giúp cho chúng ta tiếp cận với những công nghệ mới, thêm vào đó nhóm chúng em đã tích hợp thêm khả năng bảo mật cho hình khi được dùng cho cá nhân hoặc nội bộ một nhóm hoặc một công ty.

Đề tài này được thực hiện dựa trên sự hướng dẫn của thầy Trần Duy Chung, cũng như sự hỗ trợ từ các thiết bị của nhà trường, từ các trang mạng. Rất mong nhận được sự góp ý của các thầy cô trong hội đồng bảo vệ và các bạn sinh viên để có thể nâng cao chất lượng đề án.

Nội dung thực hiện bao gồm: Nghiên cứu tổng quan, cơ sở lý thuyết, các module, hệ thống gồm phần cứng và phần mềm, web sever để liên kết phần cứng và phần mềm. Kiểm tra tín hiệu kết nối giữa các khối trong hệ thống, hệ thống điều khiển từ phần cứng sang phần mềm và cuối cùng là hệ thống bảo mật tối ưu nhất cho sản phẩm.

Trong khuôn khổ đề tài này, nhóm sẽ đề xuất chủ yếu về hai phương thức truyền là MQTT và HTTP, đi kèm với đó là một trang web được thiết kế riêng theo từng khối module được mã hoá bảo mật tối ưu dữ liệu cho khách hàng khi sử dụng sản phẩm. Xong vẫn còn nhiều mặt hạn chế về nhiều mặt nên các phương án nhóm đưa ra cũng chưa phải tốt nhất và cũng khó có thể tránh khỏi những thiếu sót, nên rất mong nhóm nhận được sự đóng góp ý kiến từ quý thầy cô, bạn bè có mặt trong hội đồng bảo vệ từ đó nhóm có thể hoàn thành sản phẩm tối ưu hơn nữa.

Đề tài gồm có bốn chương:

Chương 1: Tổng quan:

- Định nghĩa
- Khái niệm IOT
- IOT từ góc nhìn kỹ thuật
- Đặc điểm cơ bản và yêu cầu ở mức cao của một hệ thống IOT

Chương 2: Cơ sở lý thuyết về IOT

- Tổng quan về IOT
- Mạng cảm biến

- Giao thức truyền MQTT
- Giao thức truyền HTTP
- Phần mềm lập trình cho vi điều khiển
- Phần mềm lập trình cho Webserver
- Truyền nhận không dây OTA
- Giới thiệu về XAMPP
- Giới thiệu về database MySQL trong PHPMyAdmin
- Giới thiệu về MQTT Broker Mosquitto
- Giới thiệu về Node-red

Chương 3: Tính toán và thiết kế hệ thống

- Giới thiệu linh kiện
- Thiết kế sơ đồ khối trong hệ thống
- Giới thiệu về các khối trong hệ thống
- Sơ đồ nguyên lý của từng khối trong hệ thống
- Tính toán và thiết kế hệ thống

Chương 4: Thi công hệ thống và kết quả

- Thi công phần cứng
- Lập trình phần mềm
- Kết quả thực hiện
- Hướng dẫn sử dụng hệ thống

Nhận xét và đánh giá, hướng phát triển

- Kết luận
- Nhận xét và đánh giá
- Hướng phát triển

Nhóm thực hiện đề tài với mục tiêu như sau:

Với đề tài: “Nghiên cứu giao thức truyền và hệ thống bảo mật trong hệ Thống IOT”, nhóm em nghiên cứu về giao thức truyền nhận dữ liệu như MQTT và HTTP từ các thiết bị thông qua wifi để đưa dữ liệu lên hệ thống sever và bảo mật dữ liệu người dùng trong hệ thống.

- Biết cách lập trình phần mềm Arduino, giao thức MQTT & HTTP
- Thiết kế giao diện web sever để giám sát trên máy tính về di động trong mạng nội bộ
- Xây dựng được mô hình và hệ thống bảo mật
- Giúp chúng ta có được kỹ năng tìm kiếm tài liệu và nghiên cứu

Chương 1: **TỔNG QUAN**

1.1 Định nghĩa

Thiết bị (device):

Đối với Internet Of Things, đây là một phần của cả hệ thống với chức năng bắt buộc là truyền thông và chức năng không bắt buộc là: cảm biến, thực thi, thu thập dữ liệu, lưu trữ và xử lý dữ liệu.

Internet Of Things:

Là một cơ sở hạ tầng mang tính toàn cầu cho xã hội thông tin, mang đến những dịch vụ tiên tiến bằng cách kết nối các “Things” (cả physical lẫn virtual) dựa trên sự tồn tại của thông tin, dựa trên khả năng tương tác của các thông tin đó, và dựa trên các công nghệ truyền thông.

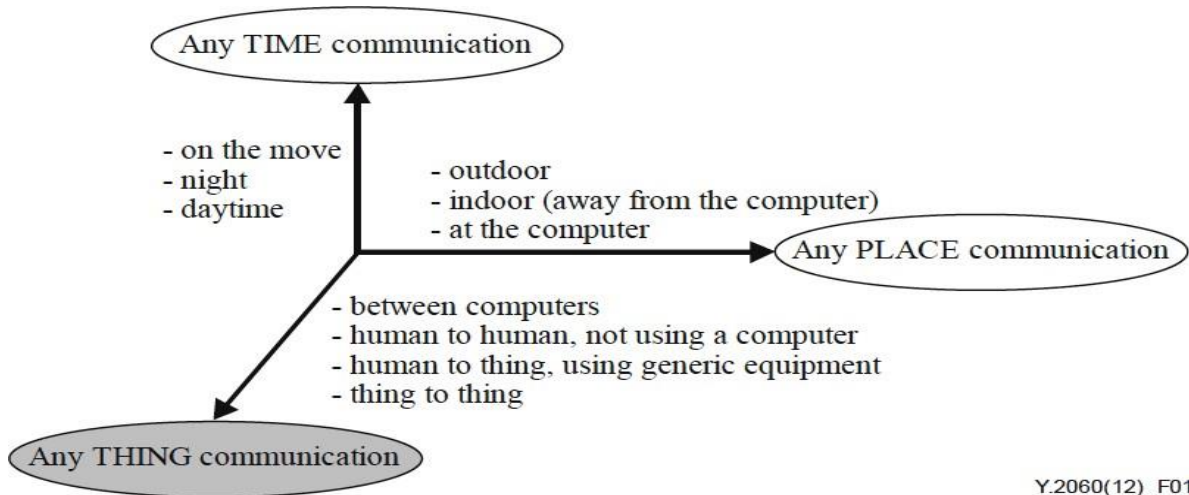
Things:

Đối với Internet Of Things, “Thing” là một đối tượng của thế giới vật chất (physical things) hay thế giới thông tin ảo (virtual things). “Things” có khả năng được nhận diện, và “Things” có thể được tích hợp vào trong mạng lưới thông tin liên lạc. [1]

1.2 Khái niệm về IOT

IoT có thể được coi là một tầm nhìn sâu rộng của công nghệ và cuộc sống. Từ quan điểm của tiêu chuẩn kỹ thuật, IoT có thể được xem như là một cơ sở hạ tầng mang tính toàn cầu cho xã hội thông tin, tạo điều kiện cho các dịch vụ tiên tiến thông qua sự liên kết các “Things”. IoT dự kiến sẽ tích hợp rất nhiều công nghệ mới, chẳng hạn như các công nghệ thông tin machine-to-machine, mạng tự trị, khai thác dữ liệu và ra quyết định, bảo vệ sự an ninh và sự riêng tư, điện toán đám mây.

Như hình dưới, một hệ thống thông tin trước đây đã mang đến 2 chiều – “Any TIME” và “Any PLACE” communication. Giờ IoT đã tạo thêm một chiều mới trong hệ thống thông tin đó là “Any THING” Communication (Kết nối mọi vật).

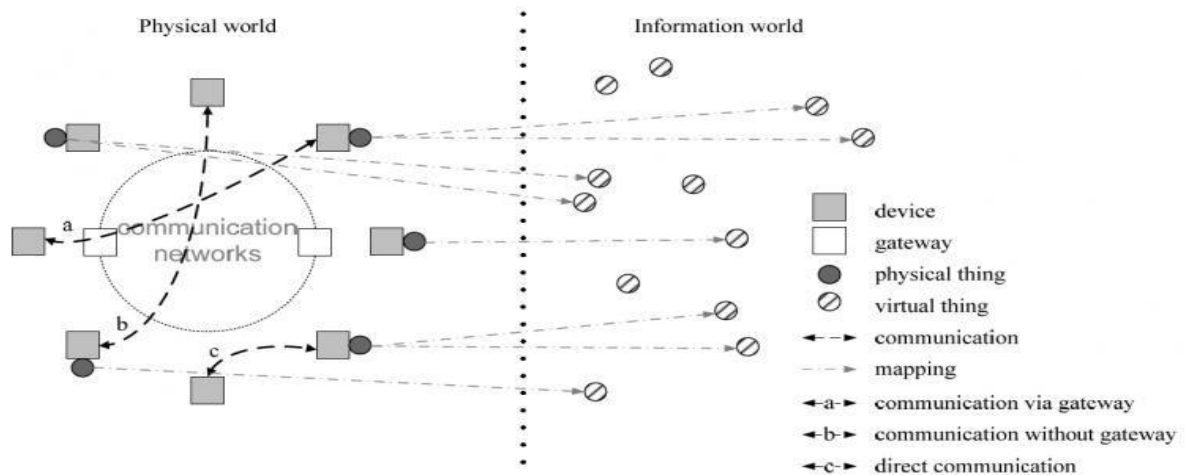


Hình 1.1 Kết nối mọi vật

Trong hệ thống IoT, “Things” là đối tượng của thế giới vật chất (Physical) hoặc các thông tin (Virtual). “Things” có khả năng nhận diện và có thể tích hợp vào mạng thông tin. “Things” có liên quan đến thông tin, có thể là tĩnh hay động. “Physical Things” tồn tại trong thế giới vật lý và có khả năng được cảm nhận, được kích thích và kết nối. Ví dụ về “Physical Things” bao gồm các môi trường xung quanh, robot công nghiệp, hàng hóa, hay thiết bị điện. “Virtual Things” tồn tại trong thế giới thông tin và có khả năng được lưu trữ, xử lý, hay truy cập. Ví dụ về “Virtual Things” bao gồm các nội dung đa phương tiện và các phần mềm ứng dụng.

1.3 IOT từ góc nhìn kỹ thuật

Như đề cập ở hình 1.1, “Things” trong IoT có thể là đối tượng vật lý (Physical) hoặc là đối tượng thông tin (hay còn gọi là đối tượng ảo - Virtual). Hai loại đối tượng này có thể ánh xạ (mapping) qua lại lẫn nhau. Một đối tượng vật lý có thể được trình bày hay đại diện bởi một đối tượng thông tin, tuy nhiên một đối tượng thông tin có thể tồn tại mà không nhất thiết phải được ánh xạ từ một đối tượng vật lý nào.



Hình 1.2 Hệ thống IoT từ góc nhìn kỹ thuật

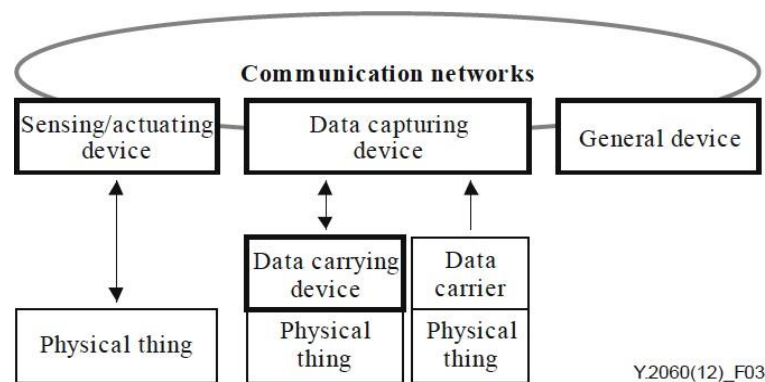
Trong hình 1.1, một “device” là một phần của hệ thống IoT. Chức năng bắt buộc của một thiết bị là giao tiếp, và chức năng không bắt buộc là cảm biến, thực thi, thu thập dữ liệu, lưu trữ dữ liệu và xử lý dữ liệu. Các thiết bị thu thập các loại thông tin khác nhau và cung cấp các thông tin đó cho các network khác, nơi mà thông tin được tiếp tục xử lý. Một số thiết bị cũng thực hiện các hoạt động dựa trên thông tin nhận được từ network.

Truyền thông thiết bị - thiết bị: Có 3 cách các thiết bị sẽ giao tiếp lẫn nhau. (a) Các thiết bị giao tiếp thông qua các mạng lưới thông tin liên lạc gọi là gateway, hoặc (b) các thiết bị giao tiếp qua mạng lưới thông tin liên lạc mà không có một gateway, hoặc (c) các thiết bị liên lạc trực tiếp với nhau qua mạng nội bộ.

Trong hình 1.2, mặc dù ta thấy chỉ có sự tương tác diễn ra ở Physical Things (các thiết bị giao tiếp với nhau). Thực ra vẫn còn hai sự tương tác khác đồng thời diễn ra. Đó là tương tác Virtual Things (trao đổi thông tin giữa các virtual things), và tương tác giữa Physical Things và Virtual Things.

Các ứng dụng IoT rất đa dạng, ví dụ, “hệ thống giao thông thông minh”, “Lưới điện thông minh”, “sức khỏe điện tử”, hoặc “nhà thông minh”. Các ứng dụng có thể được dựa trên một nền tảng riêng biệt, cũng có thể được xây dựng dựa trên dịch vụ chung, chẳng hạn như chứng thực, quản lý thiết bị, tính phí, thanh toán...

Các “Communication networks” chuyển dữ liệu được thu thập từ devices đến các ứng dụng và device khác, và ngược lại, các network này cũng chuyển các mệnh lệnh thực thi từ ứng dụng đến các device. Vai trò của communication network là truyền tải dữ liệu một cách hiệu quả và tin cậy.



Hình 1.3 Các loại thiết bị khác nhau và mối quan hệ [2]

Yêu cầu tối thiểu của các “device” trong IOT là khả năng giao tiếp [2]. Thiết bị sẽ được phân loại vào các dạng như thiết bị mang thông tin, thiết bị thu thập dữ liệu, thiết bị cảm ứng (sensor), thiết bị thực thi:

- Thiết bị mang dữ liệu (Data carrying device): Một thiết bị mang thông tin được gắn vào một Physical Thing để gián tiếp kết nối các Physical Things với các mạng lưới thông tin liên lạc.
- Thiết bị thu thập dữ liệu (Data capturing device): Một device thu thập dữ liệu có thể được đọc và ghi, đồng thời có khả năng tương tác với Physical Things. Sự tương tác có thể xảy ra một cách gián tiếp thông qua device mang dữ liệu, hoặc trực tiếp thông qua dữ liệu gắn liền với Physical Things. Trong trường hợp đầu tiên, các device thu thập dữ liệu sẽ đọc thông tin từ một device mang tin và có ghi thông tin từ các network và các device mang dữ liệu.
- Thiết bị cảm ứng và thiết bị thực thi (sensing device and actuation device): Một device cảm nhận và device thực thi có thể phát hiện hoặc đo lường thông tin liên quan đến môi trường xung quanh và chuyển đổi nó sang tín hiệu dạng số. Nó cũng có thể chuyển đổi các tín hiệu kỹ thuật số từ các mạng thành các hành động (như tắt mở đèn, chuông báo động ...). Nói chung, thiết bị và thiết bị thực thi kết hợp tạo thành một mạng cục bộ giao tiếp với nhau sử dụng công nghệ truyền thông không dây hoặc có dây và các gateway.
- General device: Một general device đã được tích hợp các network thông qua mạng dây hoặc không dây. General device bao gồm các thiết bị và đo lường cho các domain khác nhau của IOT, chẳng hạn như máy móc, thiết bị điện trong nhà, và smart phone.

1.4 Đặc điểm cơ bản và yêu cầu ở mức cao của một hệ thống IoT

1.4.1 Đặc tính cơ bản

Đặc tính cơ bản của IoT bao gồm [1], [2]:

Tính kết nối liên thông (interconnectivity): với IoT, bất cứ điều gì cũng có thể kết nối với nhau thông qua mạng lưới thông tin và cơ sở hạ tầng liên lạc tổng thể.

Những dịch vụ liên quan đến “Things”: hệ thống IoT có khả năng cung cấp các dịch vụ liên quan đến “Things”, chẳng hạn như bảo vệ sự riêng tư và nhất quán giữa Physical Thing và Virtual Thing. Để cung cấp được dịch vụ này, cả công nghệ phần cứng và công nghệ thông tin (phần mềm) sẽ phải thay đổi.

Tính không đồng nhất: Các thiết bị trong IoT là không đồng nhất vì nó có phần cứng khác nhau, và network khác nhau. Các thiết bị giữa các network có thể tương tác với nhau nhờ vào sự liên kết của các network.

Thay đổi linh hoạt: Status của các thiết bị tự động thay đổi, ví dụ, ngủ và thức dậy, kết nối hoặc bị ngắt, vị trí thiết bị đã thay đổi, và tốc độ đã thay đổi... Hơn nữa, số lượng thiết bị có thể tự động thay đổi.

Quy mô lớn: Sẽ có một số lượng rất lớn các thiết bị được quản lý và giao tiếp với nhau. Số lượng này lớn hơn nhiều so với số lượng máy tính kết nối Internet hiện nay. Số lượng các thông tin được truyền bởi thiết bị sẽ lớn hơn nhiều so với được truyền bởi con người.

1.4.2 *Yêu cầu ở mức cao đối với một hệ thống IoT*

Một hệ thống IoT phải thoả mãn các yêu cầu sau:

Kết nối dựa trên sự nhận diện: Nghĩa là các “Things” phải có ID riêng biệt. Hệ thống IOT cần hỗ trợ các kết nối giữa các “Things”, và kết nối được thiết lập dựa trên định danh (ID) của Things.

Khả năng cộng tác: hệ thống IoT khả năng tương tác qua lại giữa các mạng và Things.

Khả năng tự quản của mạng: Bao gồm tự quản lý, tự cấu hình, tự recovery, tự tối ưu hóa và tự có cơ chế bảo vệ. Điều này cần thiết để mạng có thể thích ứng với các lĩnh vực ứng dụng khác nhau, môi trường truyền thông khác nhau, và nhiều loại thiết bị khác nhau.

Dịch vụ thoả thuận: dịch vụ này để có thể được cung cấp bằng cách thu thập, giao tiếp và xử lý tự động các dữ liệu giữa các “Things” dựa trên các quy tắc (rules) được thiết lập bởi người vận hành hoặc tùy chỉnh bởi các người dùng.

Các khả năng dựa vào vị trí (location-based capabilities): Thông tin liên lạc và các dịch vụ liên quan đến một cái gì đó sẽ phụ thuộc vào thông tin vị trí của Things và người sử dụng. Hệ thống IoT có thể biết và theo dõi vị trí một cách tự động. Các dịch vụ dựa trên vị trí có thể bị hạn chế bởi luật pháp hay quy định, và phải tuân thủ các yêu cầu an ninh.

Bảo mật: Trong IoT, nhiều “Things” được kết nối với nhau. Điều này làm tăng mối nguy trong bảo mật, chẳng hạn như bí mật thông tin bị tiết lộ, xác thực sai, hay dữ liệu bị thay đổi hay làm giả.

Bảo vệ tính riêng tư: tất cả các “Things” đều có chủ sở hữu và người sử dụng của nó. Dữ liệu thu thập được từ các “Things” có thể chứa thông tin cá nhân liên quan chủ sở hữu hoặc người sử dụng nó. Các hệ thống IoT cần bảo vệ sự riêng tư trong quá trình truyền dữ liệu, tập hợp, lưu trữ, khai thác và xử lý. Bảo vệ sự riêng tư không nên thiết lập một rào cản đối với xác thực nguồn dữ liệu.

Plug and play: các Things phải được plug-and-play một cách dễ dàng và tiện dụng.

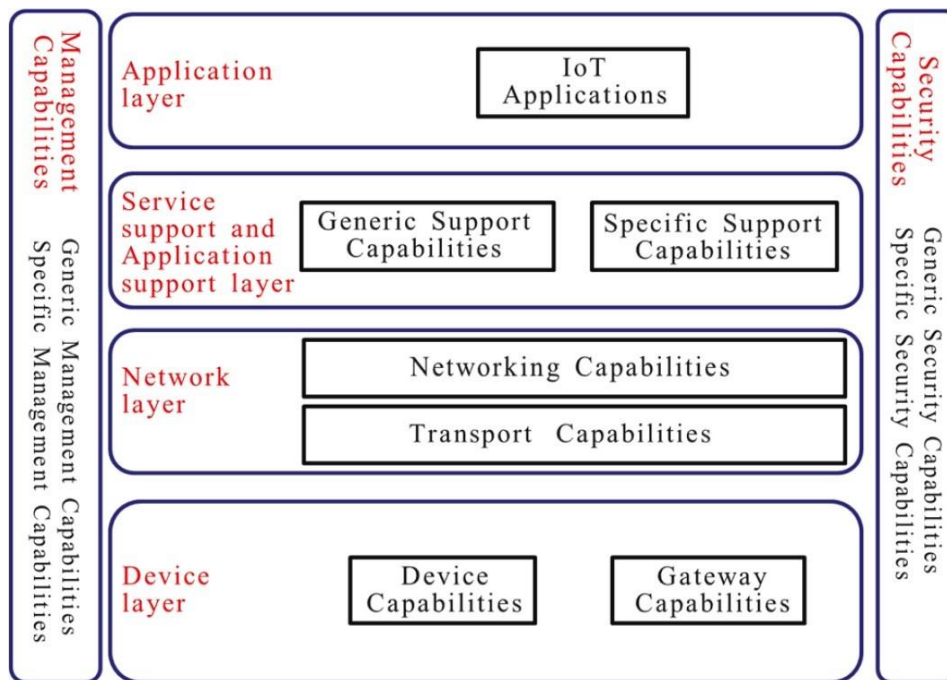
Khả năng quản lý: hệ thống IoT cần phải hỗ trợ tính năng quản lý các “Things” để đảm bảo mạng hoạt động bình thường.

Chương 2: CƠ SỞ LÝ THUYẾT VỀ IOT

2.1 Tổng quan về IoT

Bất kỳ một hệ thống IOT nào cũng được xây dựng lên từ sự kết hợp của 4 layer sau [3]:

- Lớp ứng dụng (Application Layer)
- Lớp Hỗ trợ dịch vụ và hỗ trợ ứng dụng (Service support and application support layer)
- Lớp mạng (Network Layer)
- Lớp thiết bị (Device Layer)



Hình 2.1 Mô hình IoT

2.1.1 *Application Layer*

Lớp ứng dụng cũng tương tự như trong mô hình OSI 7 lớp, lớp này tương tác trực tiếp với người dùng để cung cấp một chức năng hay một dịch vụ cụ thể của một hệ thống IOT.

2.1.2 *Service support and application support layer*

Nhóm dịch vụ chung: Các dịch vụ hỗ trợ chung, phổ biến mà hầu hết các ứng dụng IOT đều cần, ví dụ như xử lý dữ liệu hoặc lưu trữ dữ liệu. Nhóm dịch vụ cụ thể, riêng biệt: Những ứng dụng IOT khác nhau sẽ có nhóm dịch vụ hỗ trợ khác nhau và đặc thù.

Trong thực tế, nhóm dịch vụ cụ thể riêng biệt là tính toán độ tăng trưởng của cây mà đưa ra quyết định tưới nước hoặc bón phân.

2.1.3 *Network layer*

Lớp Network có 2 chức năng [3]:

- Chức năng Networking: cung cấp chức năng điều khiển các kết nối kết nối mạng, chẳng hạn như tiếp cận được nguồn tài nguyên thông tin và chuyển tài nguyên đó đến nơi cần thiết, hay chứng thực, uỷ quyền...
- Chức năng Transporting: tập trung vào việc cung cấp kết nối cho việc truyền thông tin của dịch vụ/ứng dụng IOT.

2.1.4 *Device layer*

Lớp Device chính là các phần cứng vật lý trong hệ thống IOT. Device có thể phân thành hai loại như sau [3]:

- Thiết bị thông thường: Device này sẽ tương tác trực tiếp với network: Các thiết bị có khả năng thu thập và tải lên thông tin trực tiếp (nghĩa là không phải sử dụng gateway) và có thể trực tiếp nhận thông tin (ví dụ, lệnh) từ các network. Device này cũng có thể tương tác gián tiếp với network: Các thiết bị có thể thu thập và tải network gián tiếp thông qua khả năng gateway. Ngược lại, các thiết bị có thể gián tiếp nhận thông tin (ví dụ, lệnh) từ network. Trong thực tế, các Thiết bị thông thường bao gồm các cảm biến, các phần cứng điều khiển motor, đèn, ...
- Thiết bị Gateway: Gateway là cổng liên lạc giữa device và network. Một Gateway hỗ trợ 2 chức năng sau:

Có nhiều chuẩn giao tiếp: Vì các Things khác nhau có kiểu kết nối khác nhau, nên Gateway phải hỗ trợ đa dạng từ có dây đến không dây, chẳng hạn CAN bus, ZigBee, Bluetooth hoặc Wi-Fi. Tại Network layer, gateway có thể giao tiếp thông qua các công nghệ khác nhau như PSTN, mạng 2G và 3G, LTE, Ethernet hay DSL.

Chức năng chuyển đổi giao thức: Chức năng này cần thiết trong hai tình huống là: (1) khi truyền thông ở lớp Device, nhiều device khác nhau sử dụng giao thức khác nhau, ví dụ, ZigBee với Bluetooth, và (2) là khi truyền thông giữa các Device và Network, device dùng giao thức khác, network dùng giao thức khác, ví dụ, device dùng ZigBee còn tầng network thì lại dùng công nghệ 3G.

Trong thực tế, Gateway có thể được build từ các board như Raspberry Pi hay Arduino, hoặc Gateway được sản xuất công nghiệp bởi các tập đoàn lớn như Intel hay Texas Instrument.

2.2 Mạng cảm biến

Mạng cảm biến hay còn gọi là mạng cảm biến không dây (Wireless Sensor Network) là sự kết hợp các khả năng cảm biến, xử lý thông tin và các thành phần liên lạc để tạo khả năng quan sát, phân tích và phản ứng lại với các sự kiện, hiện tượng xảy ra trong môi trường cụ thể nào đó.

Các ứng dụng cơ bản của mạng cảm biến chủ yếu gồm thu thập dữ liệu, giám sát, theo dõi, và các ứng dụng trong y học. Tuy nhiên ứng dụng của mạng cảm biến tùy theo yêu cầu sử dụng còn rất ã dạng và không bị giới hạn.

Có 4 thành phần cơ bản cấu tạo nên một mạng cảm biến [4]:

- Các cảm biến được phân bố theo mô hình tập trung hay phân bố rải.
- Mạng lưới liên kết giữa các cảm biến (có dây hay vô tuyến) điểm trung tâm tập hợp dữ liệu (Clustering) Bộ phận xử lý dữ liệu ở trung tâm.
- Một node cảm biến được định nghĩa là sự kết hợp cảm biến và bộ phận xử lý.
- Mạng cảm biến không dây (WSN) là mạng cảm biến trong đó các kết nối giữa các node cảm biến bằng sóng vô tuyến.

Hiệu quả sử dụng công suất của WSN dựa trên 3 yếu tố:

- Chu kỳ hoạt động ngắn.
- Xử lý tín hiệu nội bộ tại các node để giảm thời gian truyền.
- Mô hình dạng multihop làm giảm chiều dài đường truyền.

Một vài đặc điểm của mạng cảm biến:

- Các node phân bố dày đặc.
- Các node dễ hỏng.
- Giao thức mạng thay đổi thường xuyên.
- Node bị giới hạn về khả năng tính toán, công suất, bộ nhớ.

2.3 Phần mềm lập trình cho vi điều khiển

Arduino là một dự án được các sinh viên trường Interaction Design Institute Ivrea tại Interaction Institute Ivrea thực hiện. Là một trong những nền tảng mã nguồn mở được

nhiều người trên thế giới đón nhận và tốc độ lan tỏa một cách nhanh chóng. Việc sử dụng nền tảng một cách đơn giản đã giúp cho những người yêu thích điện tử, sinh viên và giới chuyên nghiệp hoàn thành các đề tài, dự án một cách nhanh chóng và tiết kiệm chi phí.



Hình 2.2 Biểu tượng phần mềm Arduino

2.4 Phần mềm lập trình hiển thị lên Web server

Visual studio là một phần mềm hỗ trợ đắc lực hỗ trợ công việc lập trình website. Công cụ này được tạo lên và thuộc quyền sở hữu của ông lớn công nghệ Microsoft. Năm 1997, phần mềm lập trình nay có tên mã Project Boston. Nhưng sau đó, Microsoft đã kết hợp các công cụ phát triển, đóng gói thành sản phẩm duy nhất.

Visual Studio là hệ thống tập hợp tất cả những gì liên quan tới phát triển ứng dụng, bao gồm trình chỉnh sửa mã, trình thiết kế, gỡ lỗi. Tức là, bạn có thể viết code, sửa lỗi, chỉnh sửa thiết kế ứng dụng dễ dàng chỉ với 1 phần mềm Visual Studio mà thôi. Không dừng lại ở đó, người dùng còn có thể thiết kế giao diện, trải nghiệm trong Visual Studio như khi phát triển ứng dụng Xamarin, UWP bằng XAML hay Blend vậy.

Phần mềm Visual studio được chia thành 2 phiên bản Visual Studio Enterprise và Visual Studio Professional, các phiên bản cao cấp có tính phí này được sử dụng nhiều bởi các công ty chuyên về lập trình. Bên cạnh đó, Microsoft cũng cho ra mắt phiên bản Community (phiên bản miễn phí) của gói phần mềm, cung cấp cho người dùng những tính năng cơ bản nhất, phù hợp với các đối tượng lập trình không chuyên, mới tiếp cận tìm hiểu về công nghệ (đối tượng nghiên cứu, nhà phát triển cá nhân, hỗ trợ dự án mở, các tổ chức phi doanh nghiệp dưới 5 người dùng).

Gồm nhiều tính năng:

+ Đa nền tảng

+ Đa ngôn ngữ lập trình

- + Kho tiện ích mở rộng phong phú
- + Lưu trữ phân cấp
- + Kho lưu trữ an toàn
- + Màn hình đa nhiệm
- + Hỗ trợ viết code
- + Hỗ trợ thiết bị đầu cuối
- + Hỗ trợ Git
- + Tính năng comment

2.5 Giao thức truyền nhận trong hệ thống

2.5.1 *Giao thức MQTT*

2.5.1.1 Định nghĩa:

MQTT (Message Queuing Telemetry Transport) là giao thức truyền thông điệp (message) theo mô hình publish/subscribe (cung cấp / thuê bao), được sử dụng cho các thiết bị IoT với băng thông thấp, độ tin cậy cao và khả năng được sử dụng trong mạng lưới không ổn định. Nó dựa trên một Broker (tạm dịch là “Máy chủ môi giới”) “nhẹ” (khá ít xử lý) và được thiết kế có tính mở (tức là không đặc trưng cho ứng dụng cụ thể nào), đơn giản và dễ cài đặt.

2.5.1.2 Ứng dụng:

MQTT là lựa chọn lý tưởng trong các môi trường như:

- _ Những nơi mà giá mạng viễn thông đắt đỏ hoặc băng thông thấp hay thiếu tin cậy.
- _ Khi chạy trên thiết bị nhúng bị giới hạn về tài nguyên tốc độ và bộ nhớ.
- _ Bởi vì 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 (Machine to Machine).
- _ MQTT cũng là giao thức được sử dụng trong Facebook Messenger

2.5.1.3 Tính năng, đặc điểm nổi bật:

- _ Dạng truyền thông điệp theo mô hình Pub/Sub cung cấp việc truyền tin phân tán một chiều, tách biệt với phần ứng dụng.
- _ Việc truyền thông điệp là ngay lập tức, không quan tâm đến nội dung được truyền.
- _ Sử dụng TCP/IP là giao thức nền.

_Tồn tại ba mức độ tin cậy cho việc truyền dữ liệu (QoS: Quality of service)

_QoS 0: Broker/client sẽ gửi dữ liệu đúng một lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP.

_QoS 1: Broker/client sẽ gửi dữ liệu với ít nhất một lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.

_QoS 2: Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng một lần, quá trình này phải trải qua 4 bước bắt tay.

_Phần bao bọc dữ liệu truyền nhỏ và được giảm đến mức tối thiểu để giảm tải cho đường truyền.

2.5.1.4 *Ưu điểm của MQTT:*

Với những tính năng, đặc điểm nổi bật trên, MQTT mang lại nhiều lợi ích nhất là trong hệ thống SCADA (Supervisory Control And Data Acquisition) khi truy cập dữ liệu IoT.

Truyền thông tin hiệu quả hơn.

Tăng khả năng mở rộng.

Giảm đáng kể tiêu thụ băng thông mạng.

Rất phù hợp cho điều khiển và đo thám.

Tối đa hóa băng thông có sẵn.

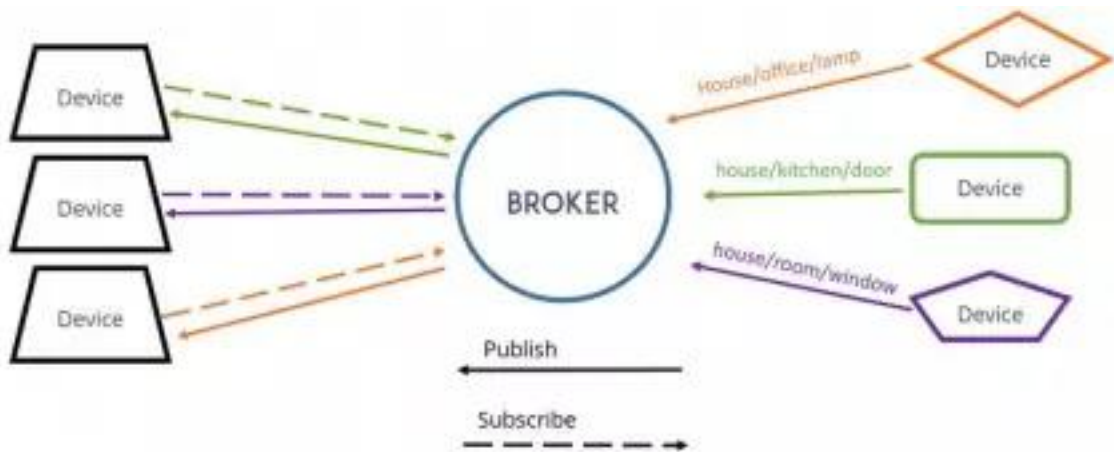
Chi phí thấp.

Rất an toàn, bảo mật.

Được sử dụng trong các ngành công nghiệp dầu khí, các công ty lớn như Amazon, Facebook,

Tiết kiệm thời gian phát triển.

Giao thức publish/subscribe thu thập nhiều dữ liệu hơn và tốn ít băng thông hơn so với giao thức cũ.



Hình 2.3 Mô tả cách thức hoạt động của giao thức MQTT

2.5.2 *Giao thức HTTP*

2.5.2.1 *Định nghĩa:*

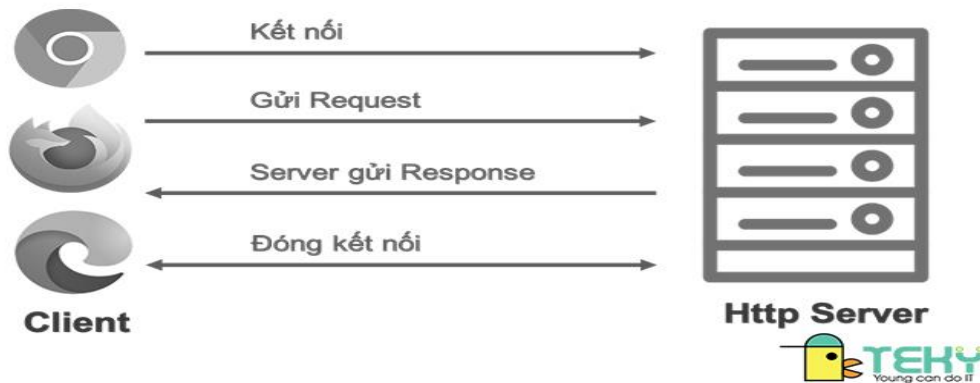
Http (HyperText Transfer Protocol) là giao thức truyền tải siêu văn bản được sử dụng trong www dùng để truyền tải dữ liệu giữa Web server đến các trình duyệt Web và ngược lại. Giao thức này sử dụng cổng 80 (port 80) là chủ yếu.

Hay bạn có thể hiểu khi bạn gõ vào 1 địa chỉ vào trình duyệt Web, lúc này trình duyệt Web sẽ gửi 1 yêu cầu qua giao thức Http đến Web server. Web server và sẽ nhận yêu cầu này và trả lại kết quả cho trình duyệt Web.

2.5.2.2 *Ứng dụng:*

HTTP được dùng trong www (world wide web) với mục đích tạo nên nền tảng kết nối giữa client và server. Client ở đây đại diện cho bất kỳ loại thiết bị hoặc chương trình nào, có thể là PC, smartphone... Còn Server được dùng để chỉ những máy tính trên đám mây. HTTP là một giao thức cho phép trao đổi và sử dụng các nguồn tài nguyên khác nhau, chẳng hạn như HTML doc. Một doc hoàn chỉnh sẽ được tạo nên từ nhiều doc con bao gồm văn bản, layout, media, video, script... HTTP lần đầu được giới thiệu vào những năm 90. Cho đến ngày nay, nó không ngừng được mở rộng và chiếm một vị trí rất quan trọng trong thế giới Internet. HTTP được coi như là một giao thức ứng dụng của bộ các giao thức nền tảng cho Internet TCP/IP. Nó cũng có thể được gửi thông qua kết nối TCP được mã hóa TLS. Nhờ vào đặc tính mở rộng không giới hạn của nó, ngoài các tài liệu siêu văn bản, HTTP còn được dùng để fìm nạp hình ảnh và video. Thậm chí là đăng tải cả nội dung lên server. Chính vì thế, HTTP đóng vai trò quan trọng trong quy trình cập nhật website thông qua khả năng nạp dữ liệu cho các doc theo yêu cầu.

2.5.2.3 *Tính năng đặc điểm nổi bật:*



Hình 2.4 Cách thức hoạt động của giao thức HTTP

- HTTP có thiết kế đơn giản

Đơn giản chính là đáp án nổi bật nhất cho câu hỏi đặc điểm của giao thức HTTP là gì. Thật vậy, mục đích xuất hiện của HTTP là tạo ra một công cụ thật thân thiện, có thể được sử dụng dễ dàng bởi mọi đối tượng. Các HTTP message đều có cấu tạo rất đơn giản để những người mới sử dụng cũng có thể đọc hiểu được.

- HTTP có khả năng mở rộng

HTTP sở hữu tính linh hoạt rất cao. Nó không có bất kỳ một giới hạn nào về sự nâng cấp và mở rộng. Thậm chí, chỉ cần bằng một thỏa thuận thống nhất giữa client và server là một tính năng mới của HTTP đã được hình thành.

- HTTP là stateless – Giao thức HTTP là gì

Bởi mọi phản hồi của HTTP là độc lập nên người dùng không thể tạo sự liên kết thông tin giữa các phản hồi được. Điều này có thể sẽ trở thành một nhược điểm đối với những trường hợp người dùng cần có các tương tác mạch lạc và hỗ trợ cho nhau, ví dụ như shopping cart trên các trang thương mại điện tử.

Để khắc phục vấn đề này, HTTP cho phép mở rộng tự do các header. Trong đó, người dùng có thể tự tạo cho mình session trên mỗi request nhằm mục đích chia sẻ các ngữ cảnh hoặc trạng thái giữa các request với nhau. Sở dĩ trường hợp này có thể thực hiện được vì bản thân HTTP là stateless.

2.5.2.4 Ưu điểm của HTTP:

Khả năng tìm kiếm: Mặc dù HTTP là một giao thức nhắn tin đơn giản, nó bao gồm khả năng tìm kiếm cơ sở dữ liệu với một yêu cầu duy nhất. Điều này cho phép giao thức được sử dụng để thực hiện các tìm kiếm SQL và trả về kết quả được định dạng thuận tiện trong tài liệu HTML.

Dễ lập trình: HTTP được mã hóa dưới dạng văn bản thuần túy và do đó dễ theo dõi và triển khai hơn các giao thức sử dụng mã yêu cầu tra cứu. Dữ liệu được định dạng dưới dạng dòng văn bản chứ không phải dưới dạng chuỗi biến hoặc trường;

Bảo mật: HTTP 1.0 tải xuống từng tệp qua một kết nối độc lập và sau đó đóng kết nối. Vì vậy, điều này làm giảm đáng kể nguy cơ bị đánh chặn trong quá trình truyền.

2.6 Cách thức bảo mật trong hệ thống

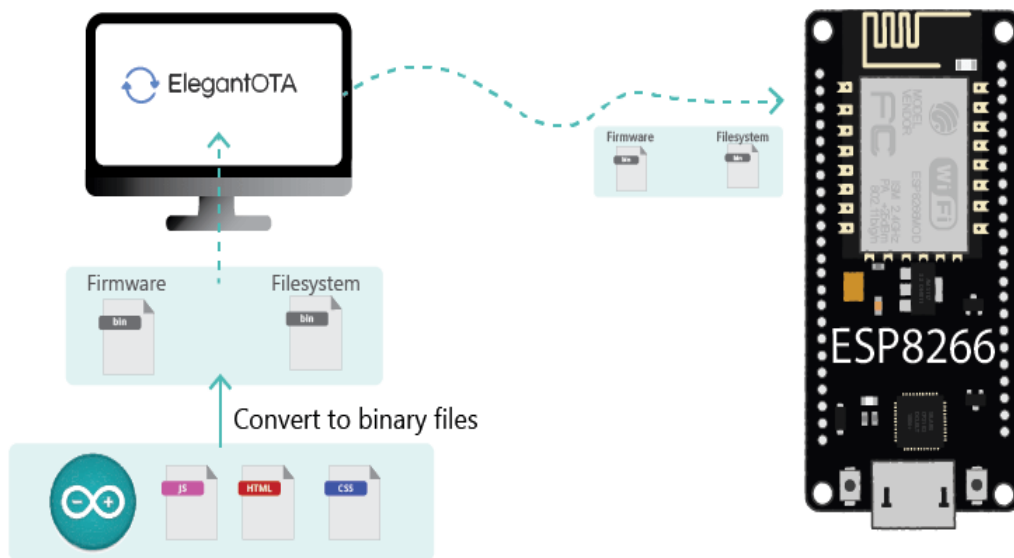
2.6.1 Cập nhật firmware cho esp qua OTA trong mạng nội bộ:

OTA (Over-the-Air) Programming

Cập nhật OTA (Over-the-Air) là quá trình tải firmware mới vào bo mạch ESP8266 bằng kết nối Wi-Fi mà không phải qua cổng giao tiếp nối tiếp. Chức năng này cực kỳ hữu ích trong trường hợp không có quyền truy cập vật lý vào bo mạch ESP8266.

Có nhiều cách khác nhau để thực hiện cập nhật OTA. Ở đây, chúng em sẽ dùng thư viện AsyncElegantOTA. Theo chúng em, đây là một trong những cách tốt và dễ dàng nhất để thực hiện cập nhật OTA, và cập nhật một cách có bảo mật trong mạng nội bộ.

Thư viện AsyncElegantOTA tạo ra một máy chủ web mà bạn có thể truy cập trên mạng cục bộ của mình để tải lên chương trình cơ sở hoặc tệp mới vào hệ thống tệp (LittleFS). Các tệp bạn tải lên phải ở định dạng .bin.



Hình 2.5 Mô tả cách thức cập nhật firmware qua OTA

2.6.2 Giới thiệu về XAMPP

Xampp là chương trình tạo máy chủ Web (Web Server) được tích hợp sẵn Apache, PHP, MySQL, FTP Server, Mail Server và các công cụ như phpMyAdmin. Không như

Appserv, Xampp có chương trình quản lý khá tiện lợi, cho phép chủ động bật tắt hoặc khởi động lại các dịch vụ máy chủ bất kỳ lúc nào.

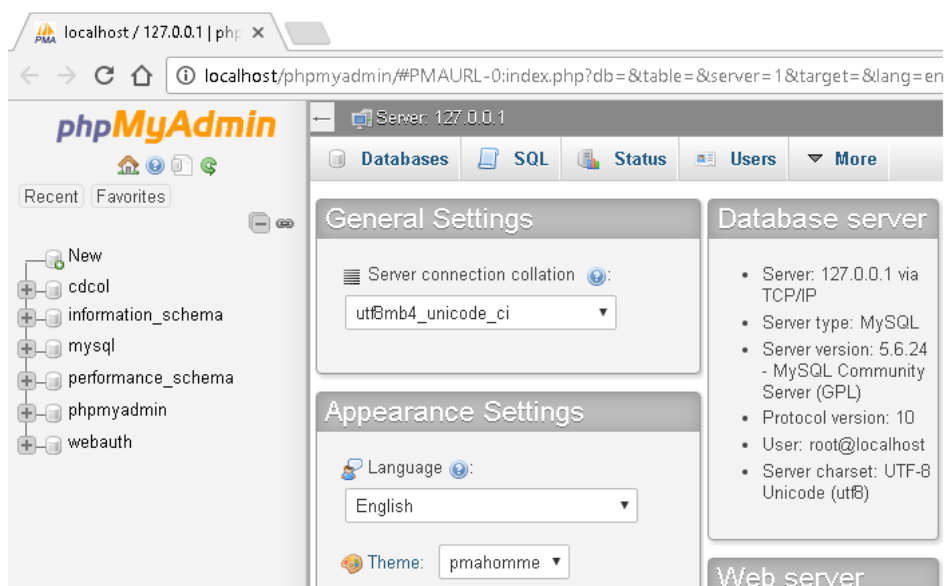
Xampp là một chương trình mã nguồn mở máy chủ web đa nền được phát triển bởi Apache Friends, bao gồm chủ yếu là Apache HTTP Server, MariaDB database, và interpreters dành cho những đối tượng sử dụng ngôn ngữ PHP và Perl. Xampp là viết tắt của Cross-Platform (đa nền tảng-X), Apache (A), MariaDB (M), PHP (P) và Perl (P). Nó phân bố Apache nhẹ và đơn giản, khiến các lập trình viên có thể dễ dàng tạo ra máy chủ web local để kiểm tra và triển khai trang web của mình. Tất cả mọi thứ cần cho phát triển một trang web - Apache (ứng dụng máy chủ), Cơ sở dữ liệu (MariaDB) và ngôn ngữ lập trình (PHP) được gói gọn trong 1 tệp. Xampp cũng là 1 chương trình đa nền tảng vì nó có thể chạy tốt trên cả Linux, Windows và MacOS. Hầu hết việc triển khai máy chủ web thực tế đều sử dụng cùng thành phần như XAMPP nên rất dễ dàng để chuyển từ máy chủ local sang máy chủ online.

2.6.3 *Giới thiệu về phần mềm phpMyAdmin:*

PhpMyAdmin là một công cụ nguồn mở được viết bằng ngôn ngữ PHP để giúp người dùng quản lý cơ sở dữ liệu của MySQL thông qua một trình duyệt web. Đây là công cụ quản trị MySQL phổ biến nhất được sử dụng bởi hàng triệu người dùng trên toàn thế giới, đặc biệt là các nhà quản trị cơ sở dữ liệu hay database administrator.

2.6.4 *Giới thiệu về cơ sở dữ liệu MySQL trong phpMyAdmin:*

Trong XAMPP đã có sẵn phpMyAdmin, chúng ta chỉ cần mở trang bằng cách click vào button Admin tương ứng với MySQL, giao diện phpMyAdmin có dạng



Hình 2.6 Giao diện tổng quan cơ sở dữ liệu PHPMyAdmin

Ta sẽ thấy phpMyAdmin có dạng ban đầu với một số database có sẵn.

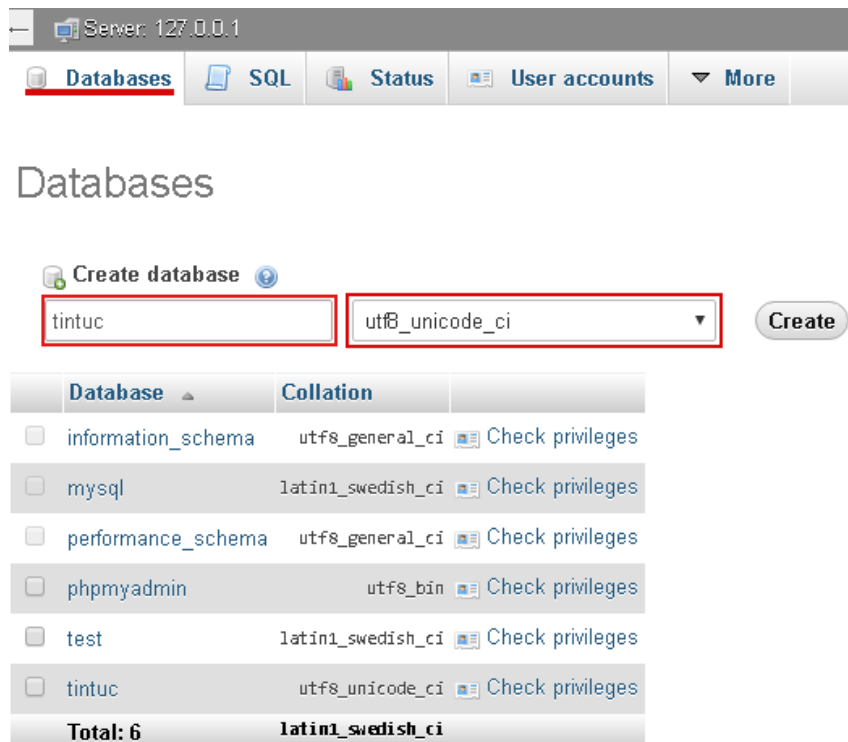
Tạo database

Click vào tab Databases, màn hình tạo database sẽ hiện ra, ở màn hình này ta chỉ chú ý phần Create database:

Database name: điền tên database cần tạo, ví dụ: tintuc.

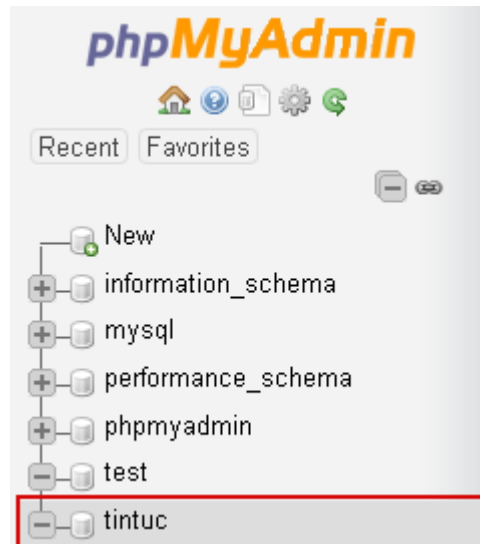
Collation: chọn dạng ngôn ngữ hiển thị, bạn có thể chọn utf8_general_ci.

Click button Create để tạo database.



Hình 2.7 Tạo một cơ sở dữ liệu

Sau khi tạo xong, nhìn bên trái ta thấy xuất hiện database có tên tintuc.



Hình 2.8 Database đã được tạo

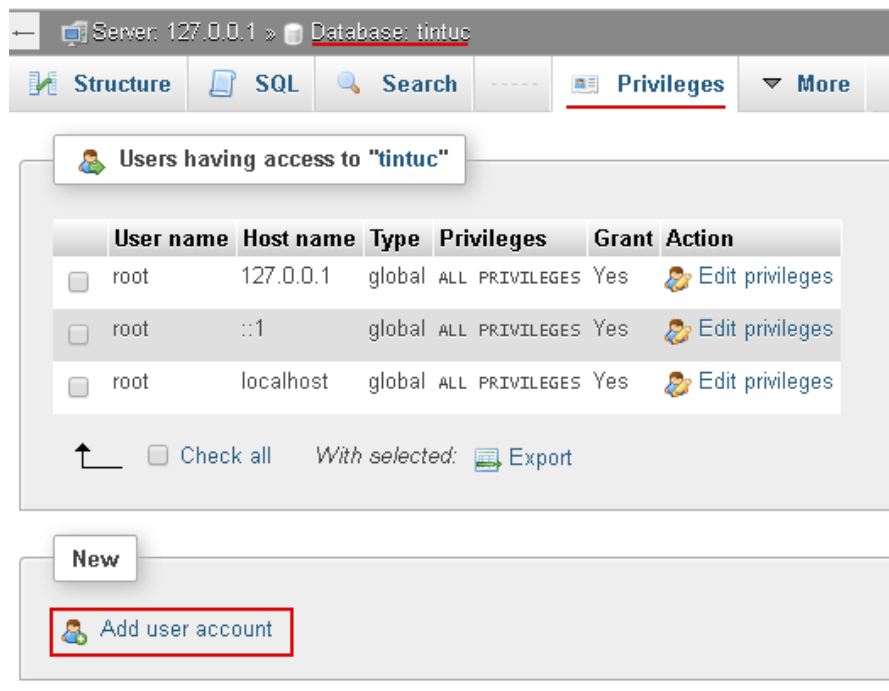
Tạo user kết nối database

_Thông thường ta có thể sử dụng luôn username root của MySQL để kết nối database, tuy nhiên về lý do security, chúng ta nên tạo riêng cho mỗi database một username riêng.

_Click vào database có tên tintuc từ danh sách database bên trái, để thao tác những gì liên quan chỉ mỗi database tintuc.

_Chọn tab Privileges để tạo user.

_Click chọn Add user account để bắt đầu tạo:



Hình 2.9 Tạo các user kết nối database

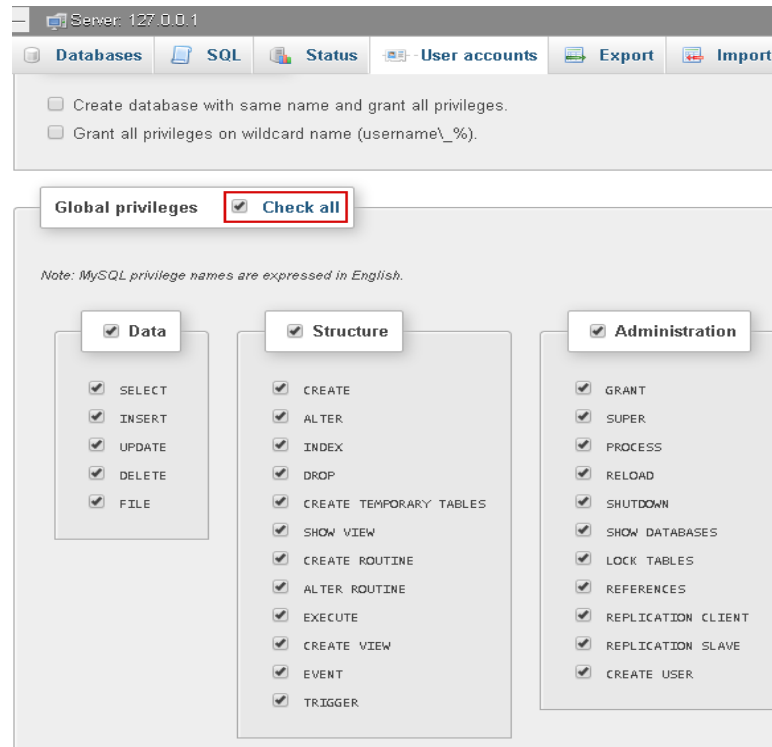
Tại màn hình tạo account, chú ý những vị trí được đánh dấu trong hình:

- ①: Đặt tên username, ví dụ user_tintuc.
- ②: Chọn host name, thông thường chọn local, bên còn lại sẽ tự động hiển thị localhost, cách chọn này cũng phù hợp khi cấu hình server thực tế.
- ③: Đặt password, ví dụ đặt 123456.
- ④: Xác nhận lại password, gõ như trên.
- ⑤: Click chọn để tạo user cho database tintuc.

The screenshot shows the 'Add user account' form in a MySQL server interface. The form is divided into two sections: 'Login Information' and 'Database for user account'. In the 'Login Information' section, the 'User name' field contains 'user_tintuc' (marked 1), the 'Host name' dropdown is set to 'Local' (marked 2), the 'Password' field contains six dots (marked 3), and the 'Re-type' field also contains six dots (marked 4). The 'Authentication Plugin' is set to 'Native MySQL authentication'. In the 'Database for user account' section, the checkbox 'Grant all privileges on database "tintuc"' is checked (marked 5). A 'Generate' button is visible below the password fields.

Hình 2.10 Tạo màn hình tạo account

Cũng ở màn hình tạo account này, click chọn Check all để cấp quyền truy cập cho user vừa tạo, tất nhiên bạn cũng có thể có lựa chọn riêng để giới hạn quyền của user.



Hình 2.11 Chọn quyền truy cập cho user

Click button Go bên dưới để kết thúc việc tạo username.

2.6.5 Giới thiệu về MQTT Broker Mosquitto

Mosquitto là một MQTT Broker mã nguồn mở cho phép thiết bị truyền nhận dữ liệu theo giao thức MQTT version 5.0, 3.1.1 và 3.1 – Một giao thức nhanh, nhẹ theo mô hình publish/subscribe được sử dụng rất nhiều trong lĩnh vực Internet of Things. Mosquitto cung cấp một thư viện viết bằng ngôn ngữ C để triển khai các MQTT Client và có thể dễ dàng sử dụng bằng dòng lệnh: “mosquitto_pub” và “mosquitto_sub”. Ngoài ra, Mosquitto cũng là một phần của Eclipse Foundation, là dự án iot.eclipse.org và được tài trợ bởi cedalo.com

➤ Ưu điểm:

- Ưu điểm nổi bật của Mosquitto là tốc độ truyền nhận và xử lý dữ liệu nhanh, độ ổn định cao, được sử dụng rộng rãi và phù hợp với những ứng dụng embedded.
- Mosquitto rất nhẹ và phù hợp để sử dụng trên tất cả các thiết bị.
- Ngoài ra, Mosquitto cũng được hỗ trợ các giao thức TLS/SSL (các giao thức nhằm xác thực server và client, mã hóa các message để bảo mật dữ liệu).

➤ Nhược điểm:

- Một số nhược điểm của mosquitto là khó thiết kế khi làm những ứng dụng lớn và ít phương thức xác thực thiết bị nên khả năng bảo mật vẫn chưa tối ưu.

2.6.6 ***Giới thiệu về Node-red***

NodeRed là một công cụ dựa trên NodeJS nhằm giúp tạo nên 1 webserver mà bạn có thể cấu hình tùy chỉnh các chức năng bằng cách kéo thả các khối lệnh trên trình duyệt web. Một ứng dụng NodeRed hoạt động theo mô hình “luồng” dữ liệu (flow), một “luồng” bao gồm các khối lệnh (gọi là các Node) liên kết với nhau theo dạng Input (Dữ liệu vào) => Operation (Xử lý) => Output (Trả kết quả).



Hình 2.12 Mô hình đơn giản của “luồng” liên kết các khối lệnh trong NodeRed

Có thể hiểu đơn giản như sau: “Luồng” chính là 1 dây chuyền sản xuất, đầu “luồng” là dữ liệu vào Input tương ứng với nguyên liệu đưa vào dây chuyền, sau đó nguyên liệu sẽ trải qua hàng loạt khâu chế biến và trả ra thành phẩm ở cuối dây chuyền. Tương tự như vậy, dữ liệu vào sẽ trải qua hàng loạt Node xử lý và cuối cùng trả ra kết quả (như xuất ra màn hình, gửi thông báo, gửi lệnh điều khiển ...).

Người dùng có thể sử dụng các Node có sẵn hoặc tự tạo Node bằng cách lập trình bằng ngôn ngữ Javascript.

Chương 3: THIẾT KẾ VÀ THI CÔNG HỆ THỐNG

3.1 Giới thiệu linh kiện

3.1.1 Module RFID RC522

Giới thiệu các loại đầu đọc RFID

Trên thị trường có rất nhiều loại đầu đọc thẻ RFID. Nó đa dạng về kiểu, màu sắc, loại kết nối... Sau đây là một số đầu đọc có trên thị trường:



Hình 3.1 Đầu đọc RFID HF

Hình trên trình bày loại đầu đọc ở dãy tần HF 13.56MHZ. Là loại đầu đọc có dây và có thêm bộ driver đính kèm cho việc giao tiếp máy tính. Thiết bị này chỉ có chức năng đọc dữ liệu từ thẻ mà không thể ghi dữ liệu.



Hình 3.2 Đầu đọc RFID UHF

Hình 3.2 trình bày loại đầu đọc hoạt động ở dải tần số UHF. Là loại đầu đọc di động và chỉ cho chức năng đọc dữ liệu từ thẻ mà không thể ghi dữ liệu. Đây cũng là loại có giá cao nhất trong các loại đầu đọc RFID trên thị trường.

Hình 3.1 trình bày loại đầu đọc/ghi dữ liệu rẻ nhất trên thị trường. Thiết bị hoạt động ở tần số 13.56MHz. Cho phép ghi/ đọc dữ liệu lên thẻ cũng như tương thích kết nối với đa số vi xử lý trên thị trường. Vì những thuận lợi mà thiết bị này mang đến nên nhóm nghiên cứu đã chọn module này làm thiết bị đọc RFID chính.



Hình 3.3 Đầu đọc/ghi RFID HF

Giới thiệu module MFRC522



Hình 3.4 Module đọc thẻ MFRC522

Module MFRC522 là module đọc/ghi trong môi trường giao tiếp tại tần số 13.56 MHz. Module hỗ trợ đọc các chuẩn ISOMECA 1443 A/MIFARE và NTAG. Module MFRC522 hỗ trợ hầu hết các loại thẻ MF1XXS20, MF1XXS70 và MF1XXS50. Module còn hỗ trợ giao tiếp và cho phép tốc độ truyền lên tới 848 kbps trong cả hai chiều đối với thẻ MIFARE.

_ Chức năng chân và Thông số kỹ thuật.

- Chức năng chân:

- + SDA(CS): chân lựa chọn chip khi giao tiếp SPI (kích hoạt mức thấp).
- + SCK: chân xung trong chế độ SPI.
- + MOSI (SDI): master data out - slave in trong chế độ giao tiếp SPI.
- + MISO (SDO): master data in – slave out trong chế độ giao tiếp SPI.
- + IRQ: chân ngắt.
- + GND: chân mass.
- + RST: chân reset module.
- + VCC: nguồn cung cấp 3.3V.

- Thông số kỹ thuật cơ bản MFRC522:

- + Nguồn: 3.3VDC, 13-26mA.
- + Tần số sóng mang: 13.56MHz.

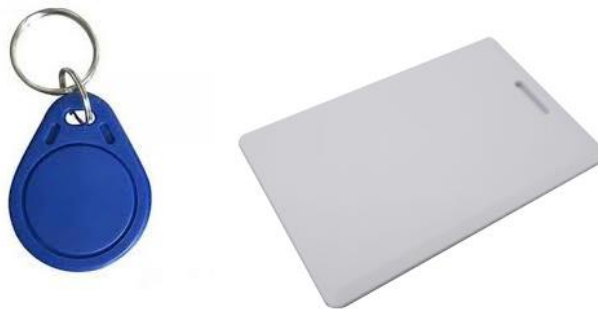
- + Khoảng cách hoạt động: 0 - 60mm (mifarel card).
- + Giao tiếp: SPI, I2C, UART.
- + Tốc độ truyền dữ liệu: tối đa 10Mbit/s.
- + Các loại Card RFID (tag) hỗ trợ: mifarel s50, mifarel S70, mifarel Ultralight, mifarel Pro, mifarel Desfire.
- + Kích thước: 40mm x 60 mm.

Nguyên lý hoạt động

Module MFRC522 là một reader nên có sẽ phát ra sóng điện từ có tần số 13.56MHZ qua anten. Khi có một thẻ tag nằm trong vùng hoạt động, thẻ tag sẽ nhận ra sóng điện từ này và thẻ sẽ thu nhận sóng này làm năng lượng. Từ đó phát lại cho module MFRC522 biết mã số cũng như dữ liệu của thẻ. Module sẽ đọc mã số thẻ và dữ liệu để thực thi một nhiệm vụ mà người sử dụng yêu cầu.

3.1.2 Thẻ RFID

Hiện nay trên thị trường có rất nhiều loại thẻ RFID với nhiều kích thước và mẫu mã khác nhau nhưng chúng đều có cấu tạo các phần cơ bản như nhau gồm 2 thành phần cơ bản: Chip lưu trữ một số thứ tự duy nhất, hoặc thông tin khác dựa trên loại thẻ và anten được gắn với vi mạch truyền thông tin từ chip đến bộ đọc, một anten được gắn với vi mạch truyền thông tin từ chip đến bộ đọc, một anten càng lớn.



Hình 3.5 Thẻ RFID

Các loại thẻ RFID được chia làm 3 loại cơ bản và thông dụng là:

- Thẻ thụ động (Passive tag): Thẻ thụ động hay nhãn dán thụ động là thẻ không có nguồn năng lượng. Ngay khi mà dòng điện được gây ra bởi những tín hiệu sóng

radio đi vào trong ăng-ten cung cấp đủ năng lượng cho mạch tích hợp CMOS (IC) trong thẻ, mạch bắt đầu hoạt động và thẻ truyền tín hiệu phản hồi lại. Điều này có nghĩa là khi thẻ thụ động đi qua máy đọc, năng lượng của sóng radio phát từ máy đọc sẽ cung cấp năng lượng cho chip và đánh thức nó để thu nhận thông tin mà nó lưu giữ. Điều này có nghĩa là anten phải thiết kế để thu năng lượng từ cả hai tín hiệu đến và tín hiệu phản lại truyền ra. Chính vì nó không có nguồn nuôi bên trong thẻ, nên những thẻ thụ động và nhãn dán thụ động có kích thước khá nhỏ, vì thẻ nó cũng không có khoảng cách đọc quá xa.

- Thẻ bán chủ động (Semi-active tag): Thẻ bán chủ động RFID tương đối giống với thẻ thụ động, trừ phần có thêm một pin nhỏ. Pin này cho phép IC của thẻ được cấp nguồn liên tục, giảm bớt sự cần thiết trong thiết kế anten thu năng lượng từ tín hiệu quay lại. Các thẻ bán thụ động không chủ động truyền tín hiệu vô tuyến về đầu đọc, mà nó nằm im bảo tồn năng lượng cho tới khi nhận được tín hiệu vô tuyến từ đầu đọc, nó sẽ kích hoạt hệ thống hoạt động. Thẻ bán chủ động RFID nhanh hơn trong sự phản hồi lại vì vậy mạnh hơn trong việc đọc số truyền so với thẻ chủ động. Do đó khoảng cách đọc của nó cũng xa hơn so với thẻ thụ động.

- Thẻ chủ động (Active tag): Đây là loại thẻ khác so với hai loại trên, thẻ chủ động RFID có nguồn năng lượng trong chính bản thân, được sử dụng để cung cấp nguồn cho tất cả các IC và phát ra tín hiệu. Chúng thường được gọi là đèn hiệu bởi vì chúng phát các tín hiệu mà chúng nhận được. Thẻ chủ động có vùng hoạt động rộng hơn, có thể lên tới vài chục mét, trong khi bộ nhớ của nó cũng lớn hơn, cho phép lưu trữ và truyền nhiều dữ liệu hơn.

Trong khuôn khổ đề tài này sử dụng thẻ thụ động (Passive tag).

3.1.3 *Vi điều khiển NodeMCU ESP8266*

➤ Khái niệm: Module ESP8266 là module wifi được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác với khoảng cách truyền lên tới 100

mét (Môi trường không có vật cản). Trên 400m với anten và router thích hợp.

_ESP8266 cung cấp một giải pháp kết nối mạng Wi-Fi hoàn chỉnh và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng.

_Khi ESP8266 là máy chủ các ứng dụng hay khi nó chỉ là bộ vi xử lý ứng dụng có trong thiết bị, nó có thể khởi động trực tiếp từ một flash ngoài. Nó có tích hợp bộ nhớ cache để cải thiện hiệu suất của hệ thống trong các ứng dụng này, và để giảm thiểu các yêu cầu bộ nhớ.

_Luôn phiên, phục vụ như một bộ chuyển đổi Wi-Fi, truy cập internet không dây có thể được thêm vào bất kỳ thiết kế vi điều khiển nào dựa trên kết nối đơn giản qua

_Khả năng lưu trữ và xử lý mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với chi phí tối thiểu và một PCB tối thiểu. Với mức độ tích hợp cao trên chip, trong đó bao gồm các anten chuyển đổi balun, bộ chuyển đổi quản lý điện năng...



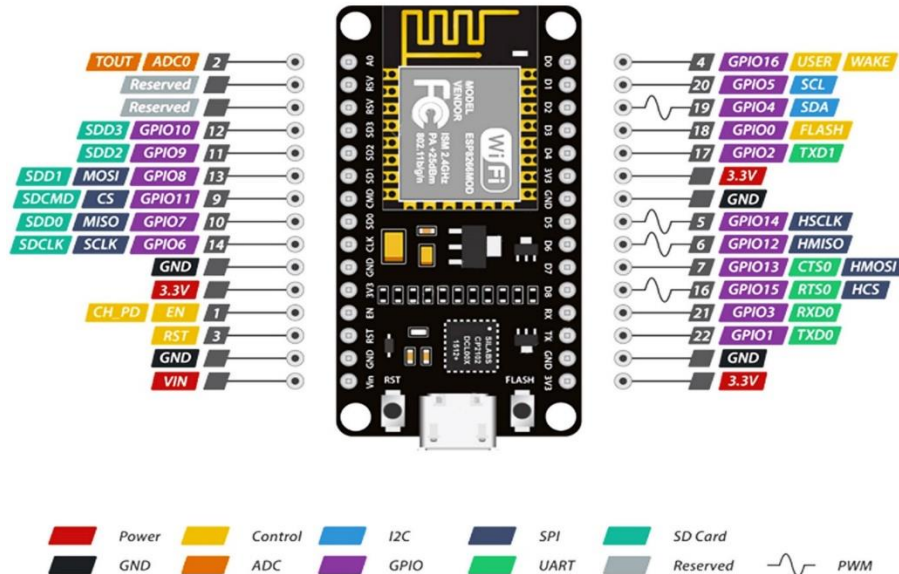
Hình 3.6 Hình ảnh thực tế Module ESP8266

➤ Cấu tạo của NODEMCU ESP8266

Module ESP8266 có các chân dùng để cấp nguồn và thực hiện kết nối. Chức năng của các chân như sau:

- + VCC: 3.3V lên đến 300MA
- + GND: Chân nối đất.
- + Tx: Chân Tx của giao thức UART, kết nối đến chân Rx của vi điều khiển.

- + Rx: Chân Rx của giao thức UART, kết nối đến chân Tx của vi điều khiển.
- + RST: chân reset, kéo xuống mass để reset.
- + 10 chân GPIO từ D0 – D8, có chức năng PWM, IIC, giao tiếp SPI, 1-Wire và ADC trên chân A0
- + Kết nối mạng wifi (có thể là sử dụng như điểm truy cập và/hoặc trạm máy chủ lưu trữ một, máy chủ web), kết nối internet để lấy hoặc tải lên dữ liệu.



Hình 3.7 Hình ảnh sơ đồ chân kết nối ESP8266

- Tính năng của Node MCU ESP8266
 - Hỗ trợ chuẩn 802.11 b/g/n.
 - Wi-Fi 2.4 GHz, hỗ trợ WPA/WPA2.
 - Chuẩn điện áp hoạt động: 3.3V.
 - Chuẩn giao tiếp nối tiếp UART với tốc độ Baud lên đến 115200
 - Tích hợp ngăn xếp giao thức TCP / IP.
 - Tích hợp chuyển đổi TR, balun, LNA, bộ khuếch đại công suất và phù hợp với mạng.
 - Tích hợp PLL, bộ quản lý, và các đơn vị quản lý điện năng.
 - Công suất đầu ra +19.5dBm trong chế độ 802.11b.
 - Tích hợp cảm biến nhiệt độ.

- Hỗ trợ nhiều loại anten.
- Wake up và truyền các gói dữ liệu trong <2ms.
- Chế độ chờ tiêu thụ điện năng <1.0mW (DTIM3).
- Hỗ trợ cả 2 giao tiếp TCP và UDP

Làm việc như các máy chủ có thể kết nối với 5 máy con

- Hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK.

Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.

3.1.4 LCD 16x02

_Giới thiệu:

LCD là một thiết bị được sử dụng rất nhiều trong các ứng dụng của vi điều khiển.

LCD có nhiều ưu điểm hơn các dạng hiển thị khác như: có khả năng hiển thị đa dạng nhiều ký tự, các ký tự rất trực quan (gồm chữ, số, ký tự đề đồ họa), dễ dàng ứng dụng theo nhiều giao thức giao tiếp khác nhau, ít chiếm tài nguyên hệ thống và giá thành hợp lý. LCD có nhiều loại về số chân nhưng chủ yếu là hai loại 14 chân và 16 chân. Điểm khác nhau giữa hai loại này là 16 chân sẽ có thêm chân nguồn cho đèn nền, còn các chân còn lại không thay đổi.



Hình 3.8 Màn hình LCD 16x02

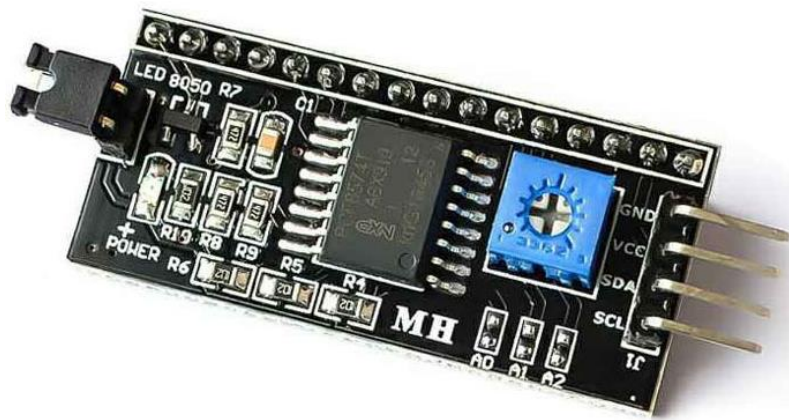
_ Thông số kỹ thuật

Bảng 3.1 Thông số kỹ thuật của LCD 16X02

Chân	Ký hiệu	Mô tả
1	Vss	Chân nối đất cho LCD, khi thiết kế mạch ta nối chân này với GND của mạch điều khiển
2	VDD	Chân cấp nguồn cho LCD, khi thiết kế mạch ta nối chân này với VCC=5V của mạch điều khiển
3	VEE	Điều chỉnh độ tương phản của LCD.
4	RS	Chân chọn thanh ghi (Register select). Nối chân RS với logic “0” (GND) hoặc logic “1” (VCC) để chọn thanh ghi. + Logic “0”: Bus DB0-DB7 sẽ nối với thanh ghi lệnh IR của LCD (ở chế độ “ghi” - write) hoặc nối với bộ đếm địa chỉ của LCD (ở chế độ “đọc” - read) + Logic “1”: Bus DB0-DB7 sẽ nối với thanh ghi dữ liệu DR bên trong LCD.
5	R/W	Chân chọn chế độ đọc/ghi (Read/Write). Nối chân R/W với logic “0” để LCD hoạt động ở chế độ ghi, hoặc nối với logic “1” để LCD ở chế độ đọc.
6	E	Chân cho phép (Enable). Sau khi các tín hiệu được đặt lên bus DB0DB7, các lệnh chỉ được chấp nhận khi có 1 xung cho phép của chân E. + Ở chế độ ghi: Dữ liệu ở bus sẽ được LCD chuyển vào (chấp nhận) thanh ghi bên trong nó khi phát hiện một xung (high-to-low transition) của tín hiệu chân E. + Ở chế độ đọc: Dữ liệu sẽ được LCD xuất ra DB0-DB7 khi phát hiện cạnh lên (low-to-high transition) ở chân E và được LCD giữ ở bus đến khi nào chân E xuống mức thấp.

7 - 14	DB0 - DB7	Tám đường của bus dữ liệu dùng để trao đổi thông tin với MPU. Có 2 chế độ sử dụng 8 đường bus này: + Chế độ 8 bit: Dữ liệu được truyền trên cả 8 đường, với bit MSB là bit DB7. + Chế độ 4 bit: Dữ liệu được truyền trên 4 đường từ DB4 tới DB7, bit MSB là DB7
15	A	Nguồn dương cho đèn nền
16	K	GND cho đèn nền

3.1.5 Module I2C LCD



Hình 3.9 Module I2C LCD

Module I2C LCD là module tích hợp IC PCF8574 cho phép mở rộng chân trong giao tiếp I2C. Để sử dụng các loại LCD có driver là HD44780 thì cần ít nhất 7 chân giao tiếp giữa LCD và MCU. Nhưng trong một số dự án sử dụng MCU có số chân hạn chế thì việc kết nối với LCD làm mất rất nhiều chân kết nối. Vì vậy module này ra đời cho phép giao tiếp với LCD thông qua hai chân SDA và SCL.

3.1.6 Chuẩn giao tiếp I2C

_Giới thiệu

I2C là chuẩn giao tiếp nối tiếp hai dây được phát triển bởi hãng Phillips. Đây là đường giao tiếp giữa các IC với nhau. Chuẩn giao tiếp này được sử dụng rộng rãi trên toàn thế giới và nó trở thành một chuẩn công nghiệp cho các giao tiếp điều khiển.

Đặc điểm của chuẩn I2C là sử dụng hai dây SDA và SCL. SDA là đường truyền dữ liệu, SCL là đường truyền xung đồng hồ để đồng bộ. Mỗi thiết bị sử dụng chuẩn I2C đều có địa chỉ giao tiếp, nó cho phép duy trì mối quan hệ chủ - tớ trong suốt thời gian kết nối.

Nguyên lý hoạt động

Thiết bị chủ tạo một điều kiện START: chuyển từ trạng thái cao xuống thấp trên đường SDA trong khi đường SCL đang ở mức cao. Điều kiện này thông báo cho thiết bị tớ lắng nghe dữ liệu trên đường truyền. Sau đó thiết bị chủ gửi địa chỉ mà thiết bị chủ muốn kết nối với thiết bị tớ. Khi thiết bị tớ trên đường truyền I2C nhận ra đúng địa chỉ thì sẽ phản hồi cho thiết bị chủ bằng một xung ACK.

Lúc này giao tiếp giữa thiết bị chủ và tớ trên đường dữ liệu bắt đầu. Mỗi xung clock từ đường SCL sẽ có một bit dữ liệu được truyền. Mức tín hiệu SDA chỉ thay đổi khi xung clock đang ở mức thấp và ổn định khi ở mức cao.

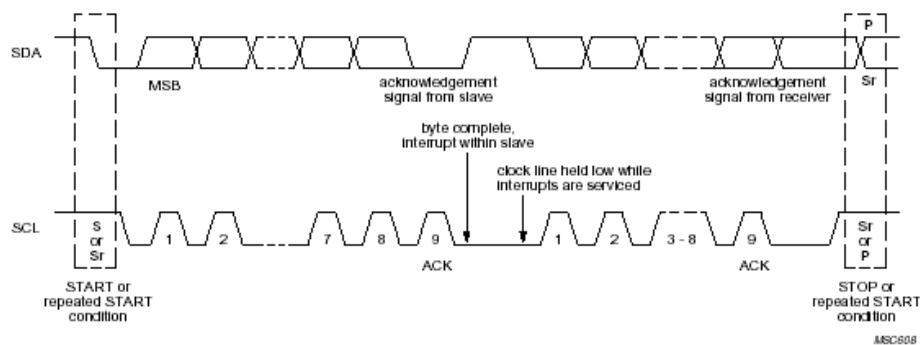


Fig.6 Data transfer on the I²C-bus.

Hình 3.10 Nguyên lý hoạt động của chuẩn giao tiếp I2C

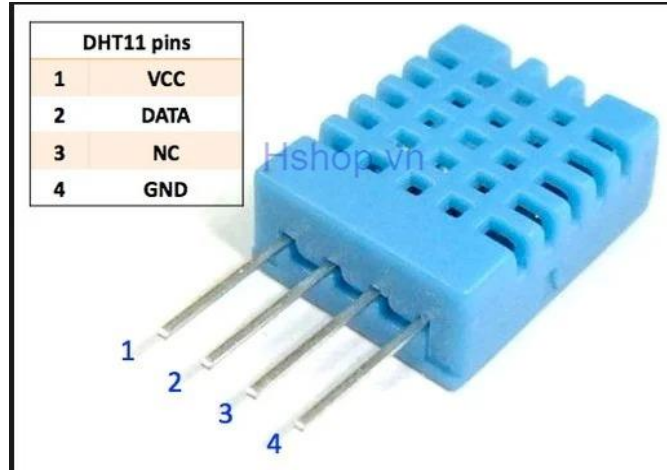
Cả thiết bị master và thiết bị slave đều có thể truyền hoặc nhận dữ liệu tùy thuộc vào quá trình truyền thông là đọc hay viết. Bên truyền sẽ gửi dữ liệu 8 bit và bên nhận sẽ phản hồi với một bit ACK.

Để kết thúc quá trình, thiết bị chủ tạo một điều kiện STOP: chuyển từ trạng thái cao sang thấp trên đường SDA trong khi đường SCL đang ở mức cao.

3.1.7 **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 (giao tiếp digital 1 dây truyền dữ liệu duy nhất). 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 3.11 Cảm biến nhiệt độ DHT11

_Thông số 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 to 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".

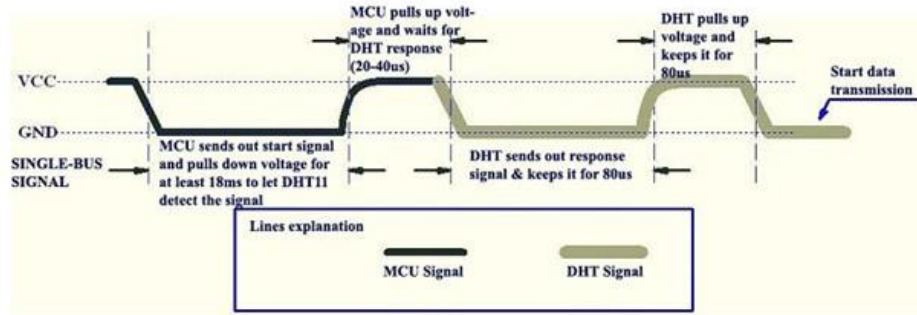
_Nguyên lý hoạt động:

Để có thể giao tiếp với DHT11 theo chuẩn 1 chân vi xử lý thực hiện theo 2 bước:

Gửi tín hiệu muốn đo (Start) tới DHTI 1, sau đó DHTI1 xác nhận lại.

- Khi đã giao tiếp với DHT11, cảm biến sẽ gửi lại 5 byte dữ liệu và nhiệt độ đo được.

Bước 1: Gửi tín hiệu Start



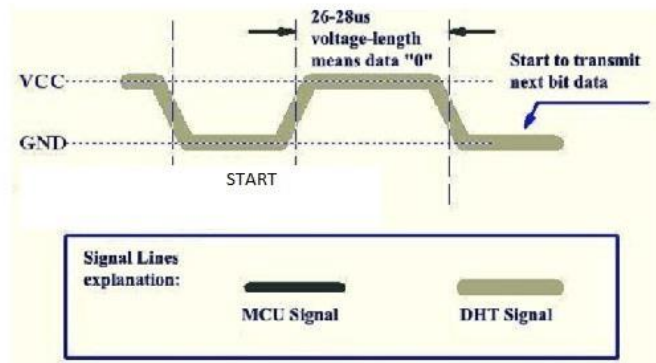
Hình 3.12 Bắt đầu chuẩn bị vào quá trình truyền dữ liệu

- MCU thiết lập chân DATA là Output, kéo chân DATA xuống 0 trong khoảng thời gian >18ms. Trong Code mình để 25ms. Khi đó DHT11 sẽ hiểu MCU muốn đo giá trị nhiệt độ và độ ẩm.
 - MCU đưa chân DATA lên 1, sau đó thiết lập lại là chân đầu vào.
 - Sau khoảng 20-40us, DHT11 sẽ kéo chân DATA xuống thấp. Nếu >40us mà chân DATA ko được kéo xuống thấp nghĩa là ko giao tiếp được với DHT11.
 - Chân DATA sẽ ở mức thấp 80us sau đó nó được DHT11 kéo lên cao trong 80us. Bằng việc giám sát chân DATA, MCU có thể biết được có giao tiếp được với DHT11 ko. Nếu tín hiệu đo được DHT11 lên cao, khi đó hoàn thiện quá trình giao tiếp của MCU với DHT.
- Bước 2: đọc giá trị trên DHT11
- DHT11 sẽ trả giá trị nhiệt độ và độ ẩm về dưới dạng 5 byte. Trong đó:
 - Byte 1: giá trị phần nguyên của độ ẩm (RH%)
 - Byte 2: giá trị phần thập phân của độ ẩm (RH%)
 - Byte 3: giá trị phần nguyên của nhiệt độ (TC)
 - Byte 4: giá trị phần thập phân của nhiệt độ (TC)
 - Byte 5: kiểm tra tổng.
 - Nếu Byte 5 = (8 bit) (Byte1 +Byte2 +Byte3 + Byte4) thì giá trị độ ẩm và nhiệt độ là chính xác, nếu sai thì kết quả đo không có nghĩa.

Đọc dữ liệu:

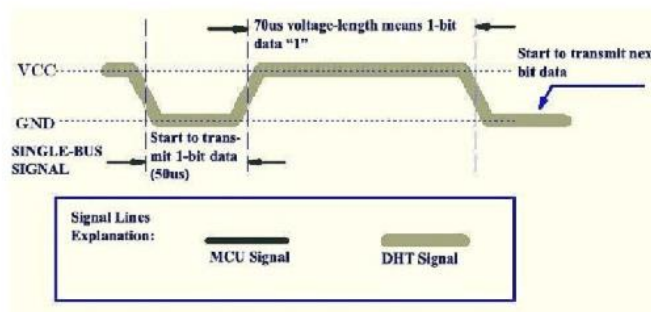
Sau khi giao tiếp được với DHT11, DHT11 sẽ gửi liên tiếp 40 bit 0 hoặc 1 về MCU, tương ứng chia thành 5 byte kết quả của Nhiệt độ và độ ẩm.

Bit 0:



Hình 3.13 Truyền dữ liệu bit 0

Bit 1:



Hình 3.14 Truyền dữ liệu bit 1

Sau khi tín hiệu được đưa về 0, ta đợi chân DATA của MCU được DHT11 kéo lên 1. Nếu chân DATA là 1 trong khoảng 26-28 us thì là 0, còn nếu tồn tại 70us là 1. Do đó trong lập trình ta bắt sườn lên của chân DATA, sau đó delay 50us. Nếu giá trị đo được là 0 thì ta đọc được bit 0, nếu giá trị đo được là 1 thì giá trị đo được là 1. Cứ như thế ta đọc các bit tiếp theo.

3.1.8 *Cảm biến hồng ngoại*

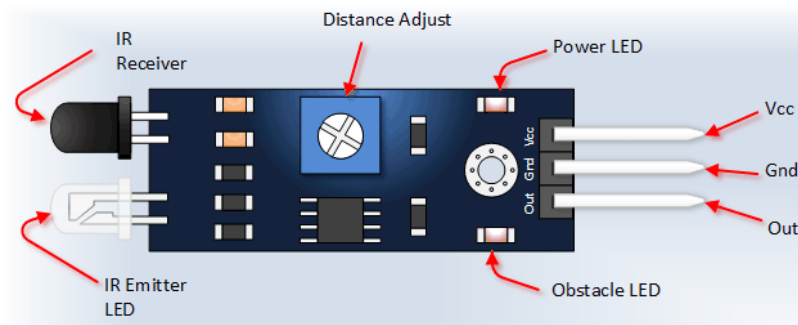
Cảm biến vật cản hồng ngoại có khả năng thích nghi với môi trường, có một cặp truyền và nhận tia hồng ngoại. Tia hồng ngoại phát ra một tần số nhất định, khi phát hiện hướng truyền có vật cản (mặt phản xạ), phản xạ vào đèn thu hồng ngoại, sau khi so sánh, đèn màu xanh sẽ sáng lên, đồng thời đầu cho tín hiệu số đầu ra (một tín hiệu bậc thấp)

Khoảng cách làm việc hiệu quả 2 ~ 5cm, điện áp làm việc là 3.3 V đến 5V. Độ nhạy sáng của cảm biến vật cản hồng ngoại được điều chỉnh bằng chiết áp, cảm biến dễ lắp ráp, dễ sử dụng,

Có thể được sử dụng rộng rãi trong robot tránh chướng ngại vật, xe tránh chướng ngại vật và dò đường....

_Thông số kỹ thuật:

- Bộ so sánh sử dụng LM393, làm việc ổn định
- Điện áp làm việc: 3.3V – 5V DC.
- Khi bật nguồn, đèn báo nguồn màu đỏ sáng.
- Lỗ vít 3 mm, dễ dàng cố định, lắp đặt.
- Kích thước: 3.2cm * 1.4cm
- Các mô-đun đã được so sánh điện áp ngưỡng thông qua chiết áp, nếu sử dụng ở chế độ thông thường, xin vui lòng không tự ý điều chỉnh chiết áp.



Hình 3.15 Cảm biến hồng ngoại

_Nguyên lý hoạt động:

Cảm biến hồng ngoại hoạt động bằng cách sử dụng một cảm biến ánh sáng cụ thể nào đó để phát hiện ra bước sóng ánh sáng chọn trong phổ hồng ngoại (IR).

Muốn xem cường độ của ánh sáng nhận được, bạn sử dụng đèn LED tạo ra được ánh sáng có cùng bước sóng với cảm biến hồng ngoại đang tìm kiếm. Khi vật đó đang ở gần cảm biến, ánh sáng từ đèn LED sẽ ra khỏi vật thể và đi vào cảm biến ánh sáng. Đây được gọi là một bước nhảy lớn về cường độ.

3.1.9 Cảm biến độ ẩm đất

_Giới thiệu: Cảm biến độ ẩm đất dùng để đo độ ẩm của đất. Module nhỏ gọn giá thành thấp nên có thể dễ dàng triển khai qua mô lớn. Ngõ ra analog và digital có thể kích relay.

Module cảm biến độ ẩm đất ứng dụng trong các hệ thống tưới tự động, vườn thông minh, chăm sóc cây cảnh.



Hình 3.16 Cảm biến độ ẩm đất

_Thông số kỹ thuật:

Điện áp hoạt động: 3 – 5VDC

Đầu ra analog và digital, ngưỡng kích hoạt có thể chỉnh bằng biến trở trên board.

_Nguyên lý

Cảm biến đo độ ẩm hoạt động dựa trên nguyên lý: sự hấp thụ hơi nước làm biến đổi tính chất của thành phần cảm nhận trong cảm biến làm thay đổi điện trở của cảm biến qua đó xác định được độ ẩm.

Đối với một ẩm kế điện dung, không khí chảy vào giữa hai tấm kim loại. Sự thay đổi độ ẩm không khí tỷ lệ thuận với sự thay đổi điện dung giữa các bản.

Trong nguyên lý đo độ ẩm điện trở, polymer hoặc sự hấp thụ độ ẩm, sau đó ảnh hưởng đến điện trở suất của nó. Và được kết nối với một mạch trong đó độ ẩm ảnh hưởng đến điện trở của vật liệu. Từ đó độ ẩm tương đối sau đó được xác định dựa trên sự thay đổi của dòng điện.

3.1.10 *Module relay*

- Giới thiệu:

Relay hay còn gọi là rơ – le là tên gọi theo tiếng Pháp, là một công tắc (khóa K) điện từ được vận hành bởi một dòng điện tương đối nhỏ có thể bật hoặc tắt một dòng điện lớn hơn nhiều. Bản chất của relay là một nam châm điện (một cuộn dây trở thành một nam châm tạm thời khi dòng điện chạy qua nó) và hệ thống các tiếp điểm đóng cắt có thiết kế module hóa dễ dàng lắp đặt.



Hình 3.17 Module relay

- Thông số kỹ thuật:

- Điện áp nuôi mạch: 12VDC.
- Dòng tiêu thụ: khoảng 200mA/1Relay
- Tín hiệu kích: High (12VDC) hoặc Low (0VDC) chọn bằng Jumper.
- Relay trên mạch:
- Nguồn nuôi: 12VDC.
- Tiếp điểm đóng ngắt max: 250VAC-10A hoặc 30VDC-10A
- Kích thước: 72 (L) * 55 (W) * 19 (H) mm.
- Nguyên lý hoạt động:
- Trong khi vận hành có một số bước cơ bản xảy ra khi rơ le cơ điện được cấp điện hay ngắt điện
- Điện được cung cấp cho cuộn dây tạo ra từ trường
- Từ trường được chuyển thành lực cơ học bằng cách hút phần ứng
- Phần ứng động đóng/mở một hoặc nhiều tiếp điểm điện
- Các tiếp điểm cho phép chuyển mạch điện sang tải như động cơ, bóng đèn, ...
- Sau khi điện áp cuộn bị loại bỏ từ trường biến mất các tiếp điểm tách ra và trở về vị trí bình thường
- Các tiếp điểm có thể thường đóng hoặc thường mở

3.1.11 **Khối nguồn**

- Giới thiệu:

Nguồn tổ ong 5V 5A sử dụng điện áp đầu vào: 180V-240V, công suất: 60W, có chức năng bảo vệ quá tải 105% - 150% công suất định mức và phục hồi tự động, có chức năng bảo vệ ngắn mạch tự động...



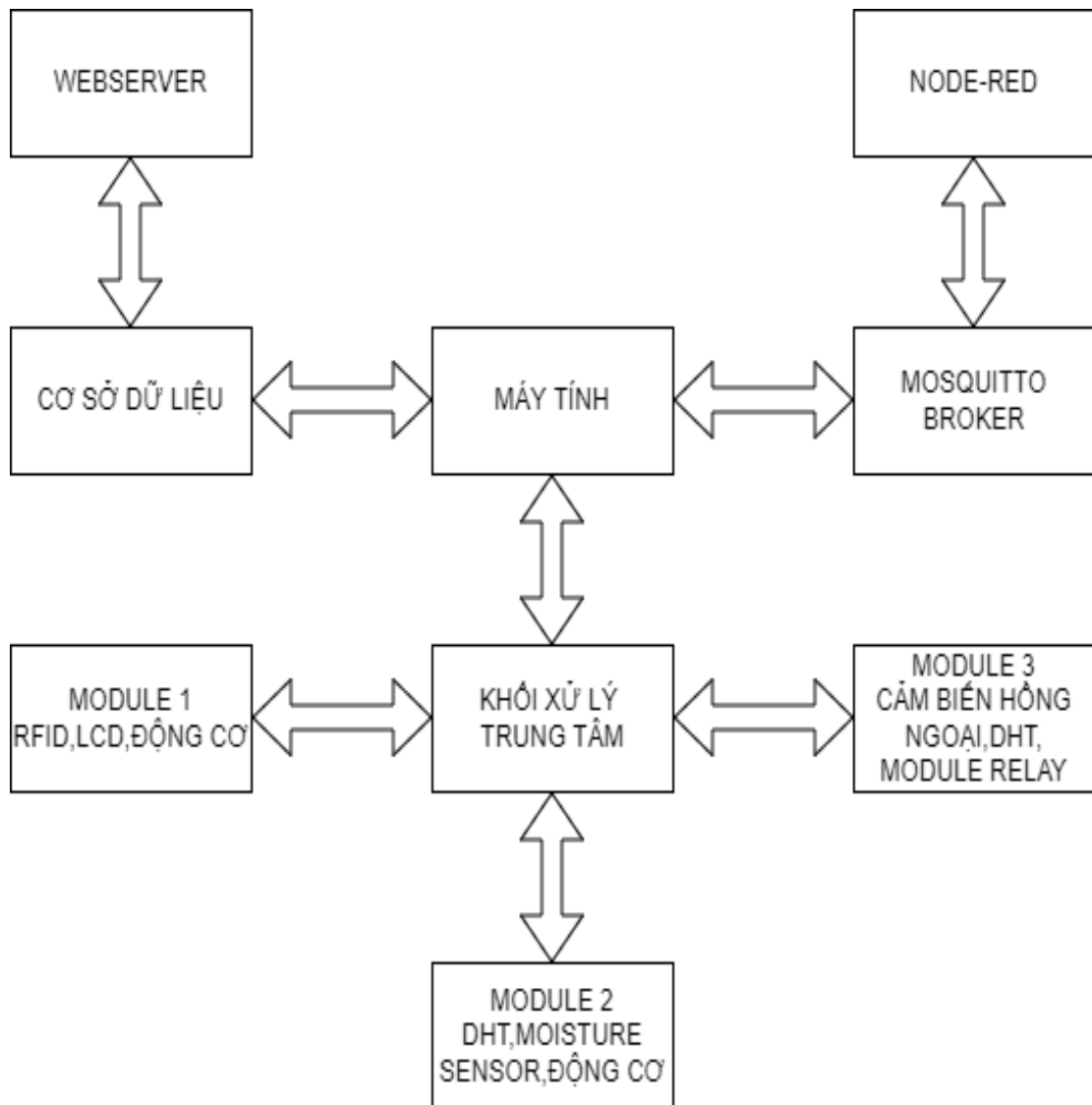
Hình 3.18 Khối nguồn tổ ong

- Thông số kỹ thuật:

- Điện áp đầu vào: 180V-240V
- Tần số hoạt động: 47 ~ 63HZ
- Công suất: 60W
- Điện áp đầu ra: 5V
- Dòng điện tối đa: 5A
- Điện áp điều chỉnh: $\pm 10\%$
- Hiệu suất $\geq 85\%$
- Điều chỉnh điện áp (Đầy tải) $\leq 0.3\%$
- Bảo vệ quá tải 105% - 150% công suất định mức, phục hồi tự động
- Chức năng bảo vệ ngắn mạch tự động
- Bảo vệ quá áp 105% - 150% điện áp định mức
- Nhiệt độ làm việc: $-20\text{ }^{\circ}\text{C} \sim 60\text{ }^{\circ}\text{C}$

- Nhiệt độ bảo quản -40 °C ~ 85 °C
- Nguyên lý hoạt động:
- Nguồn được thiết kế sẵn các cổng điện áp vào và ra rất dễ cho người sử dụng. Với nguồn công suất lớn thường sẽ có 2 cổng điện vào 220VAC cho 2 pha L-N, 3 cổng 0V, 3 cổng điện áp (+), cổng đất cho thiết bị nào cần.
 - Với các loại nguồn có công suất nhỏ hơn thì số lượng cổng đầu ra sẽ ít hơn 1 chút nhưng vẫn rất tiện cho người dùng. Chúng ta chỉ việc đấu nối dây vào sao cho chuẩn xác.
 - Các bạn cần tính toán công suất sử dụng chính xác. Tránh trường hợp quá tải sẽ gây hỏng nguồn hoặc hỏng thiết bị điện.

3.2 Thiết kế sơ đồ khối trong hệ thống



Hình 3.19 Sơ đồ khối hệ thống

3.2.1 Chức năng của từng khối:

_Khối xử lý trung tâm: nhận dữ liệu từ các module và đưa dữ liệu lên server và biểu diễn lên Web server và Node-red.

_Khối module 1:

- Giới thiệu:

Ở đây, nhóm chúng em làm hệ thống nhận dạng, mở cửa, dùng webserver trên server xampp, truy cập trong mạng nội bộ. (GET/POST)

- Cơ sở lý thuyết sử dụng:

HTTP GET/POST

Server XAMPP

Database MySQL PHPMyAdmin.

- Chức năng của khối:

Khối module 1 gồm Node MCU ESP8266 kết nối với module RFID, thông qua server XAMPP trên máy tính kết nối với mạng và gửi dữ liệu lên database MySQL PHPMyAdmin. Từ đó hiển thị thông tin thẻ lên webserver và điều khiển động cơ mở đóng bằng thẻ RFID, ở đây nếu thông tin trong thẻ đúng thì mở động cơ và nếu thông tin trong thẻ không đúng hoặc thẻ chưa được đăng kí thì động cơ không mở.

_Khối module 2:

- Giới thiệu:

Ở đây, nhóm chúng em làm hệ thống theo dõi nhiệt độ, độ ẩm DHT11 hiển thị lên Node-red theo thời gian thực và điều khiển động cơ theo độ ẩm đất.

- Cơ sở lý thuyết sử dụng:

MQTT, NODE-RED

- Chức năng của khối:

Khối module 2 gồm Node MCU ESP8266 kết nối với cảm biến nhiệt độ, độ ẩm DHT11 và cảm biến độ ẩm đất. Qua broker mosquitto trên máy tính hiển thị nhiệt độ, độ ẩm lên Node-Red qua đó ta có thể giám sát các thông số theo thời gian thực.

_Khối module 3:

- Giới thiệu:

Ở đây, nhóm chúng em làm hệ thống điều khiển mở tắt đèn theo cảm biến hồng ngoại và qua Node-Red theo giao thức MQTT.

- Cơ sở lý thuyết:

Node-Red, MQTT.

- Chức năng của khối:

Khối module 3 gồm Node MCU ESP8266 kết nối với cảm biến nhiệt độ, độ ẩm DHT11 và cảm biến hồng ngoại. Qua broker mosquitto trên máy tính hiển thị nhiệt độ, độ ẩm lên Node-Red qua đó ta có thể giám sát nhiệt độ, độ ẩm theo thời gian thực và bật tắt đèn thông qua Node-Red hoặc cảm biến hồng ngoại.

3.2.2 Nguyên lý hoạt động:

Khi ta bật XAMPP và các chương trình cần thiết thì chúng ta có thể đi đến trang đăng nhập của hệ thống. Trang đăng nhập của hệ thống đã được liên kết với cơ sở dữ liệu được tích hợp trong XAMPP. Khi chúng ta đăng nhập vào hệ thống thì sẽ có trang Homepage cho tất cả các module, muốn truy cập vào module nào thì cần có mật khẩu của module đó (ở đây tất cả các mật khẩu để vào module đều được cài đặt trước trong chương trình và không thay đổi bên ngoài), khi đăng nhập đúng mật khẩu của module thì sẽ được truy cập vào trang hiển thị dữ liệu của module đó.

- Module 1:

-Khi đăng nhập vào hệ thống và quét thẻ RFID, mã thẻ sẽ được khối vi điều khiển gửi lên cơ sở dữ liệu thông qua XAMPP trên máy tính đồng thời hiển thị tên người dùng qua màn hình LCD, từ trang để hiển thị thông tin của Webserver sẽ lấy mã thẻ đó truy xuất lên cơ sở dữ liệu để lấy thông tin về hiển thị lên Webserver.

-Trong trang hiển thị thông tin của Webserver chúng ta có thể thêm thẻ mới, xóa thẻ, thay thế sửa thông tin người dùng.

- Module 2:

-Khi đăng nhập vào hệ thống ta sẽ thấy được nhiệt độ và độ ẩm được cập nhật lên Node-red theo thời gian thực.

-Trong module này chúng ta còn có chức năng điều khiển động cơ theo độ ẩm đất.

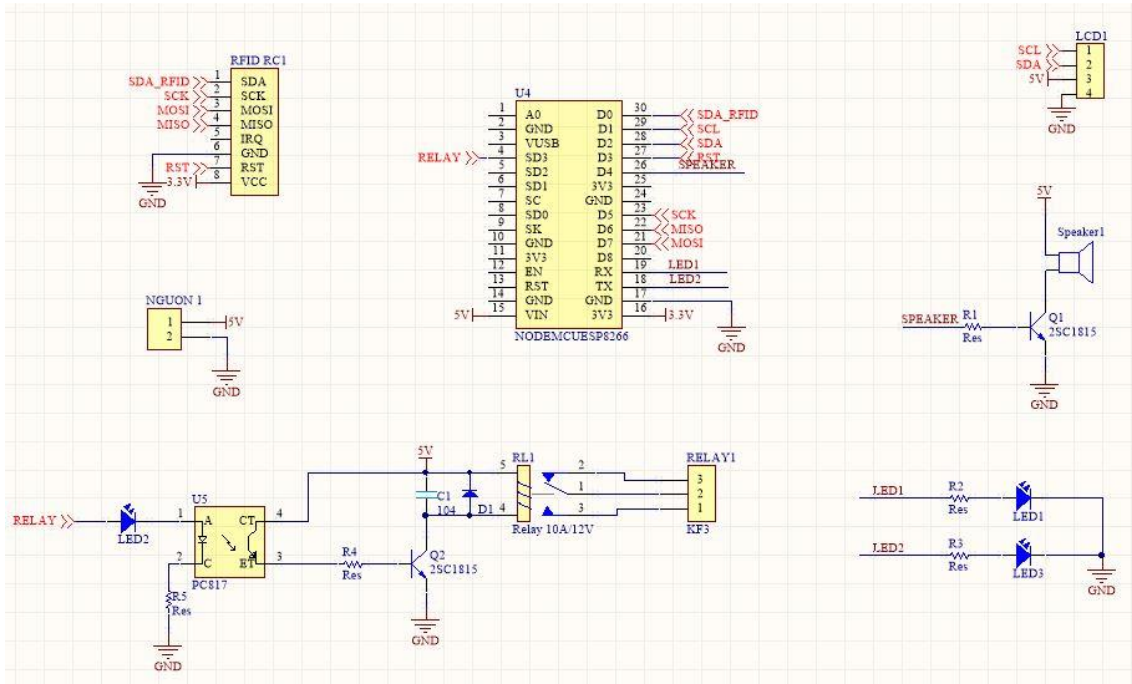
- Module 3:

-Khi đăng nhập vào hệ thống ta sẽ thấy được nhiệt độ và độ ẩm được cập nhật theo thời gian thực lên Node-red và chúng ta còn có thể điều khiển đèn thông qua công tắc trên đó.

-Trong module này chúng ta còn có chức năng điều khiển động cơ theo cảm biến hồng ngoại.

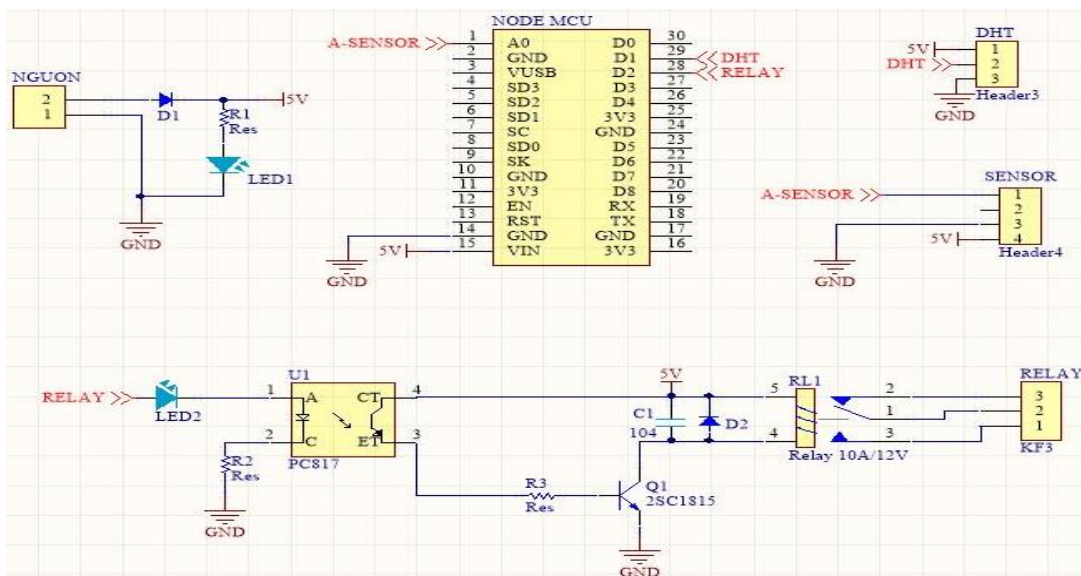
3.3 Sơ đồ nguyên lý từng khối trong hệ thống

_Module 1:



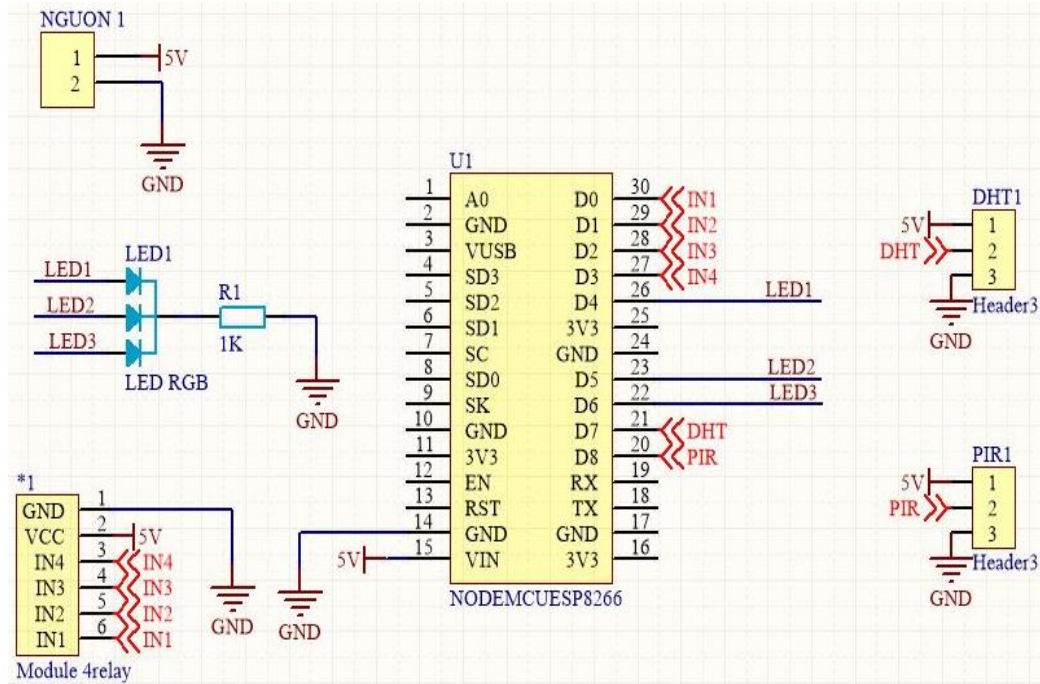
Hình 3.20 Sơ đồ nguyên lý module 1

_Module 2



Hình 3.21 Sơ đồ nguyên lý module 2

_Module 3



Hình 3.22 Sơ đồ nguyên lý module 3

3.4 Tính toán thiết kế hệ thống

Thiết kế khối ngõ ra công suất

Chọn relay 5 Vde, loại này chỉ cần đặt áp 5 Vde vào 2 đầu cuộn dây và dòng trên cuộn dây đảm bảo trên 70mA là các tiếp điểm hoạt động. Mặt khác, dòng tối đa mà rơle này có thể chịu được tối đa trên tiếp điểm là 10A, nên đảm bảo dòng điện trong quạt, đèn khi chạy qua các tiếp điểm sẽ an toàn. Sử dụng Transistor đóng ngắt loại NPN - D468, vì loại này có dòng lên tới 1A để đóng ngắt cho relay.

Gắn thêm 1 Diode giữa 2 đầu cuộn dây mỗi Relay, để chống quá áp ngược từ cuộn dây, tránh hỏng các transistor. Chọn diode loại IN4007, loại này chịu được áp ngược tối đa 700V.

Để đảm bảo an toàn cho mạch điều khiển thì nhóm sử dụng thêm Opto PC817 (hay còn gọi là cách ly quang). Nó là linh kiện tích hợp có cấu tạo gồm 1 Led và 1 photodiode hay 1 phototransistor. Được sử dụng để cách ly giữa khối điều khiển với khối ngõ ra công suất.

Led đỏ cần dòng 10mA và áp rơi là 1.9 - 4V. Từ đó ta suy ra giá trị:

$$R_{ledmax} = \frac{VCC - V_{led} - 0.2}{I_{led}} = \frac{5 - 1.9 - 0.2}{10} = 290 \Omega$$

$$R_{ledmin} = \frac{VCC - V_{led} - 0.2}{I_{led}} = \frac{5 - 4 - 0.2}{10} = 80 \Omega$$

Chọn $R_{led} = 220 \Omega$

Dòng tối thiểu qua cuộn dây:

$$I_{cdây} = \frac{U_2 \text{ đầu cuộn dây}}{\text{Nội trở cuộn dây}} = \frac{5}{70} = 70 \text{ mA}$$

$$I_c = I_{cdây} = 70 \text{ mA}$$

$$\text{Mà } I_c = \beta \times I_b$$

Transistor có hệ số β khoảng 80 – 240, chọn β khoảng 100.

$$I_b = 0.7 \text{ mA} \rightarrow R_b = \frac{5-0.2}{0.7} = 6.9 \text{ k}\Omega \rightarrow R_b = 10 \text{ k}\Omega$$

Chương 4: THI CÔNG HỆ THỐNG VÀ KẾT QUẢ

4.1 Thi công phần cứng

4.1.1 Thi công board mạch

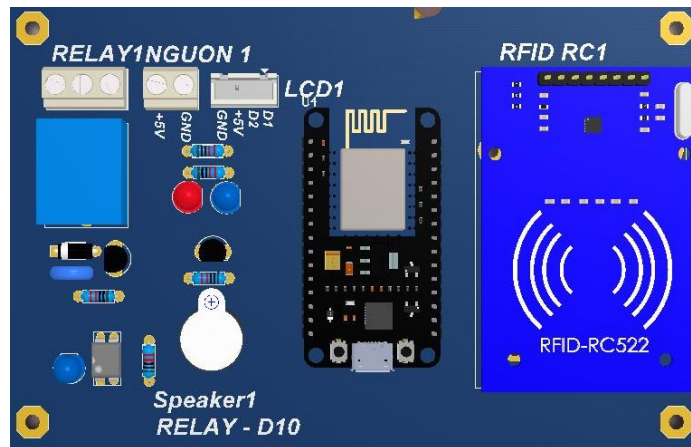
➤ Các bước thi công phần cứng

-Mạch in được thiết kế trên phần mềm Altimum 17.0.7

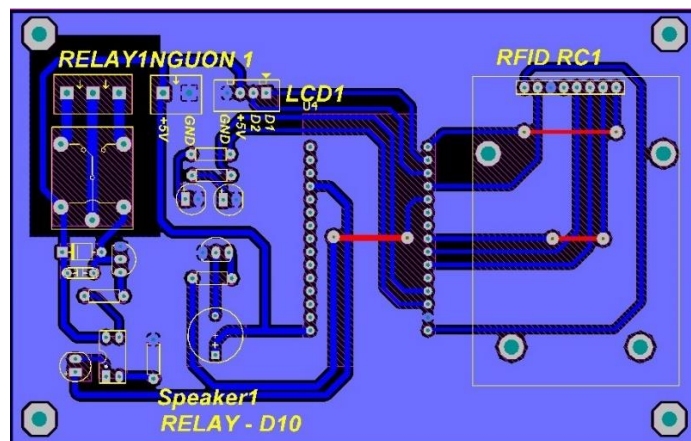
-Thực hiện thi công mạch và tiến hành thi công board mạch.

-Sau khi thi công sẽ dùng đồng hồ VOM để kiểm tra ngõ vào, ngõ ra để xem có lỗi trong lúc thực hiện hay không.

➤ Mạch in



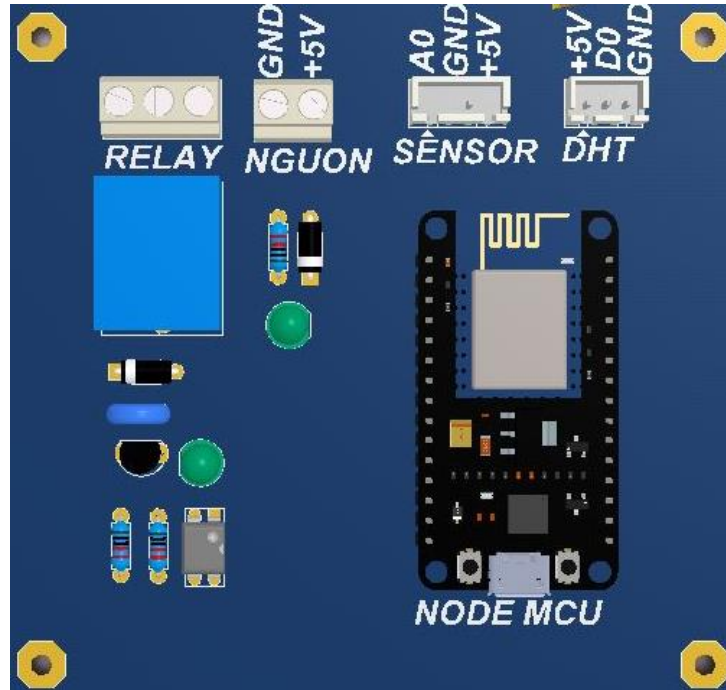
Hình 4.1 Sơ đồ bố trí linh kiện của module 1



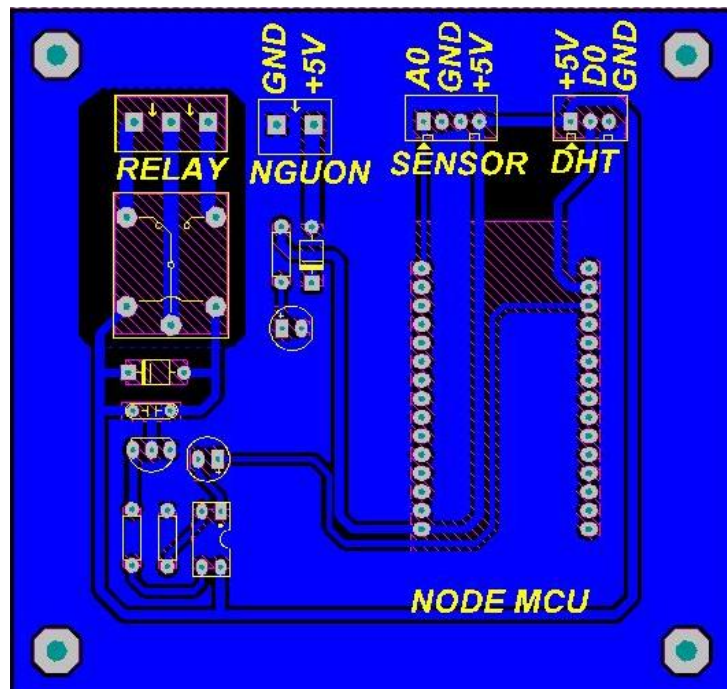
Hình 4.2 Mạch in sau khi thiết kế

○ Mô tả sơ đồ module 1:

Sơ đồ bao gồm một khối Node MCU ESP8266 kết nối với module RFID, màn hình LCD, relay và buzzer và các linh kiện khác như trở, led, opto, ...



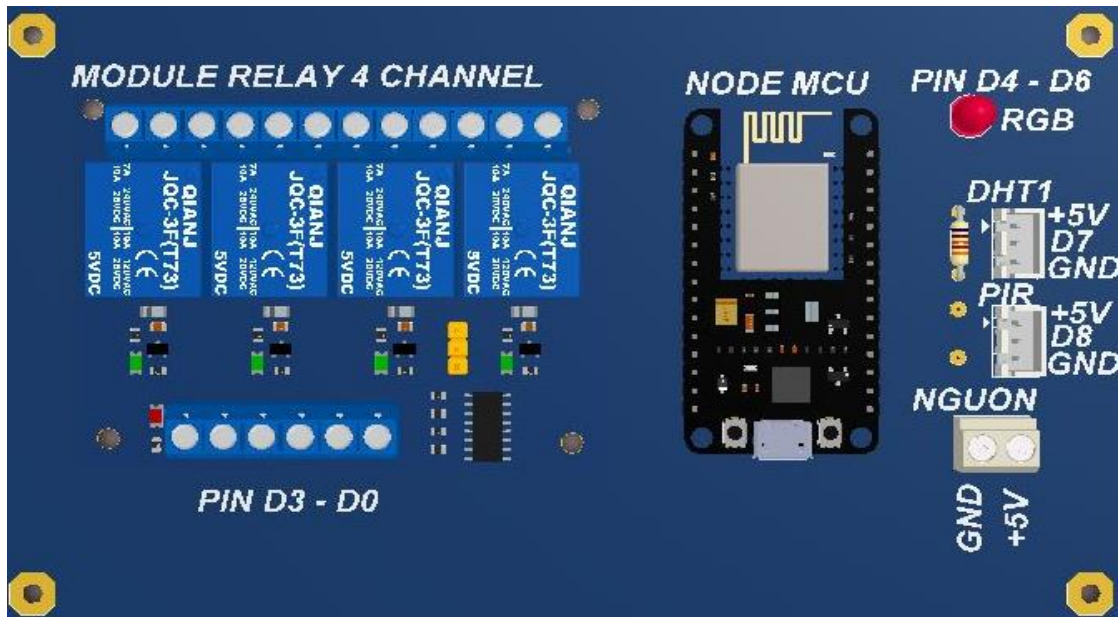
Hình 4.3 Sơ đồ bố trí linh kiện của module 2



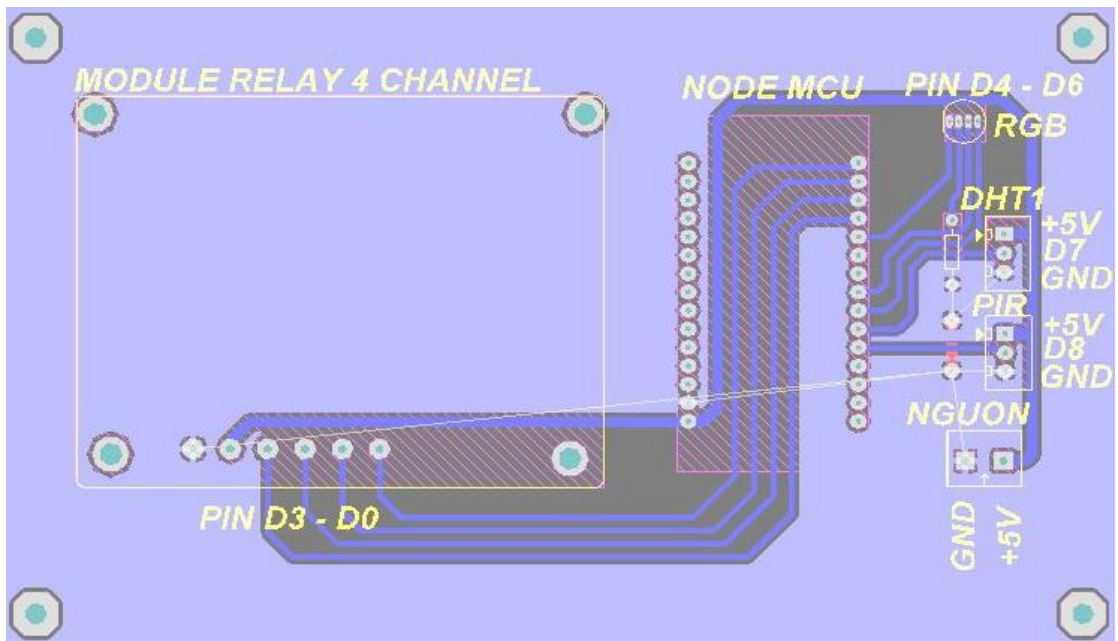
Hình 4.4 Mạch in sau khi thiết kế

o Mô tả sơ đồ module 2

Sơ đồ bao gồm một khối Node MCU ESP8266 kết nối với header để các cảm biến như cảm biến nhiệt độ, độ ẩm DHT, cảm biến độ ẩm đất, relay và các linh kiện khác như trở, led, opto, ...



Hình 4.5 Sơ đồ bố trí linh kiện của module 3



Hình 4.6 Mạch in sau khi thiết kế

- Mô tả sơ đồ module 3

Sơ đồ bao gồm một khối Node MCU ESP8266 kết nối với header để các cảm biến như cảm biến nhiệt độ, độ ẩm DHT, cảm biến hồng ngoại, module relay và các linh kiện khác như trở, led, opto, ...

➤ **Thống kê linh kiện trong hệ thống**

Bảng 4.1 Linh kiện sử dụng trong module 1

STT	Tên linh kiện	Số lượng
1	ESP8266	1
2	RFID Reader	1
3	Relay	1
4	Opto	1
5	LCD và I2C	1
6	Buzzer	1
7	Led	3

Bảng 4.2 Linh kiện sử dụng trong module 2

STT	Tên linh kiện	Số lượng
1	ESP8266	1
2	Terminal	2
3	Relay	1
4	Opto	1
5	Cảm biến DHT	1
6	Cảm biến độ ẩm đất	1
7	Led	2

Bảng 4.3 Linh kiện sử dụng trong module 3

STT	Tên linh kiện	Số lượng
1	ESP8266	1
2	Terminal	1
3	Module Relay	1

4	Cảm biến DHT	1
5	Cảm biến hồng ngoại	1
6	Bóng đèn	2

4.1.2 Thi công mô hình

Sau khi đã kết nối phần cứng, ta kết nối với máy tính thông qua cổng USB, như vậy ta đã được một mô hình hoàn chỉnh. Sau đó, tiến hành lập trình cho Arduino, xây dựng phần mềm quản lý và cơ sở dữ liệu.



Hình 4.7 Mô hình khối module 1



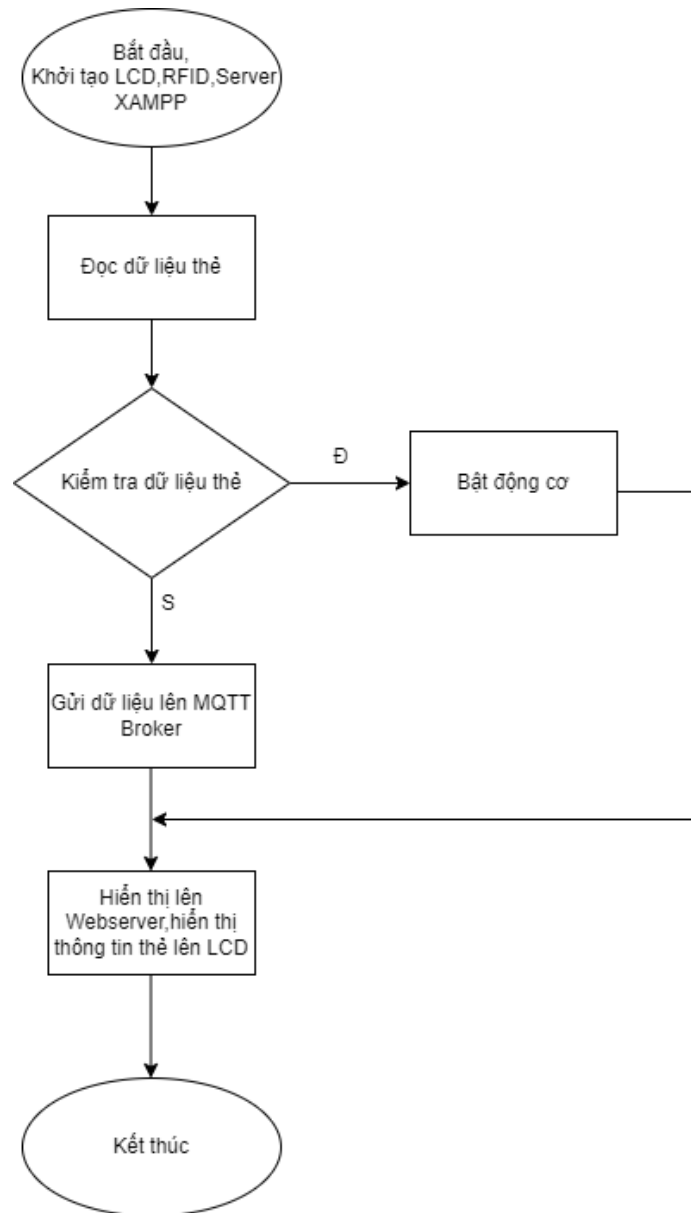
Hình 4.8 Mô hình khối module 2



Hình 4.9 Hình ảnh mô hình khối module 3

4.2 Lập trình phần mềm

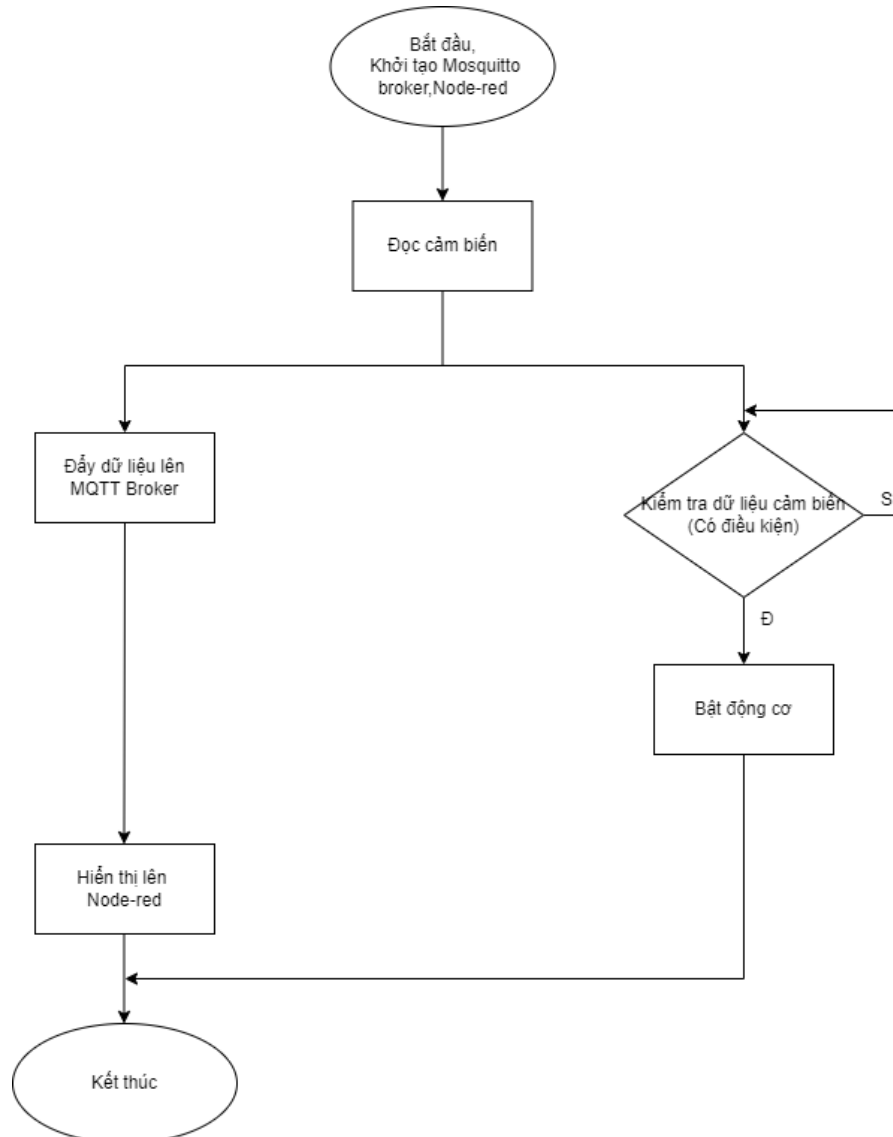
4.2.1 Lưu đồ giải thuật cho vi điều khiển



Hình 4.10 Lưu đồ giải thuật cho vi điều khiển module 1

- Giải thích lưu đồ module 1:

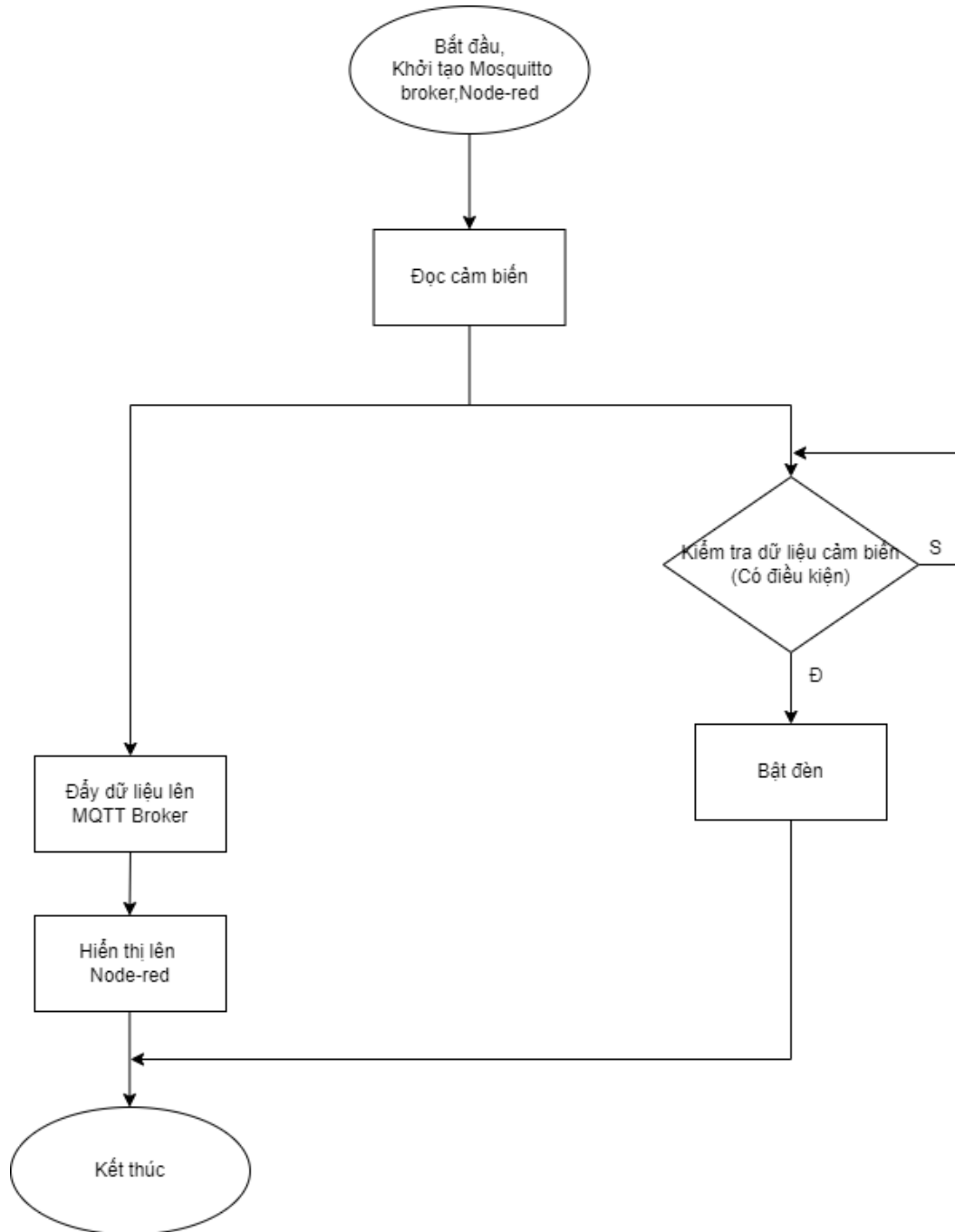
Sau khi ta kết nối nguồn và nạp chương trình cho vi điều khiển, các khối module LCD, RFID sẽ được khởi tạo, sau đó ta tiến hành bật XAMPP server để kết nối đến cơ sở dữ liệu MySQL PHPMyAdmin, tiếp theo ta đọc thẻ và kiểm tra tín hiệu RFID. Nếu có tín hiệu nhận được từ thẻ RFID thì vi điều khiển sẽ xử lý và gửi dữ liệu của mã thẻ RFID lên cơ sở dữ liệu và hiển thị lên màn hình LCD và phần hiển thị của Web server. Nếu vi điều khiển không nhận được tín hiệu từ thẻ RFID thì sẽ tiếp tục quay lại kiểm tra tín hiệu RFID.



Hình 4.11 Lưu đồ giải thuật cho vi điều khiển module 2

- Giải thích module 2:

Sau khi ta kết nối nguồn và nạp chương trình cho vi điều khiển, sau đó ta tiến hành bật Mosquitto Broker và phần mềm Node-red, tiếp theo dữ liệu cảm biến sẽ được gửi qua Mosquitto Broker và hiển thị lên Node-red. Nếu có tín hiệu từ cảm biến độ ẩm đất thì vi điều khiển sẽ xử lý, nếu điều kiện đúng thì quay động cơ, điều kiện sai thì không quay động cơ. Nếu có tín hiệu liên tục từ cảm biến nhiệt độ, độ ẩm thì sẽ gửi liên tục qua broker và biểu diễn lên Node-red, người dùng có thể theo dõi theo thời gian thực.

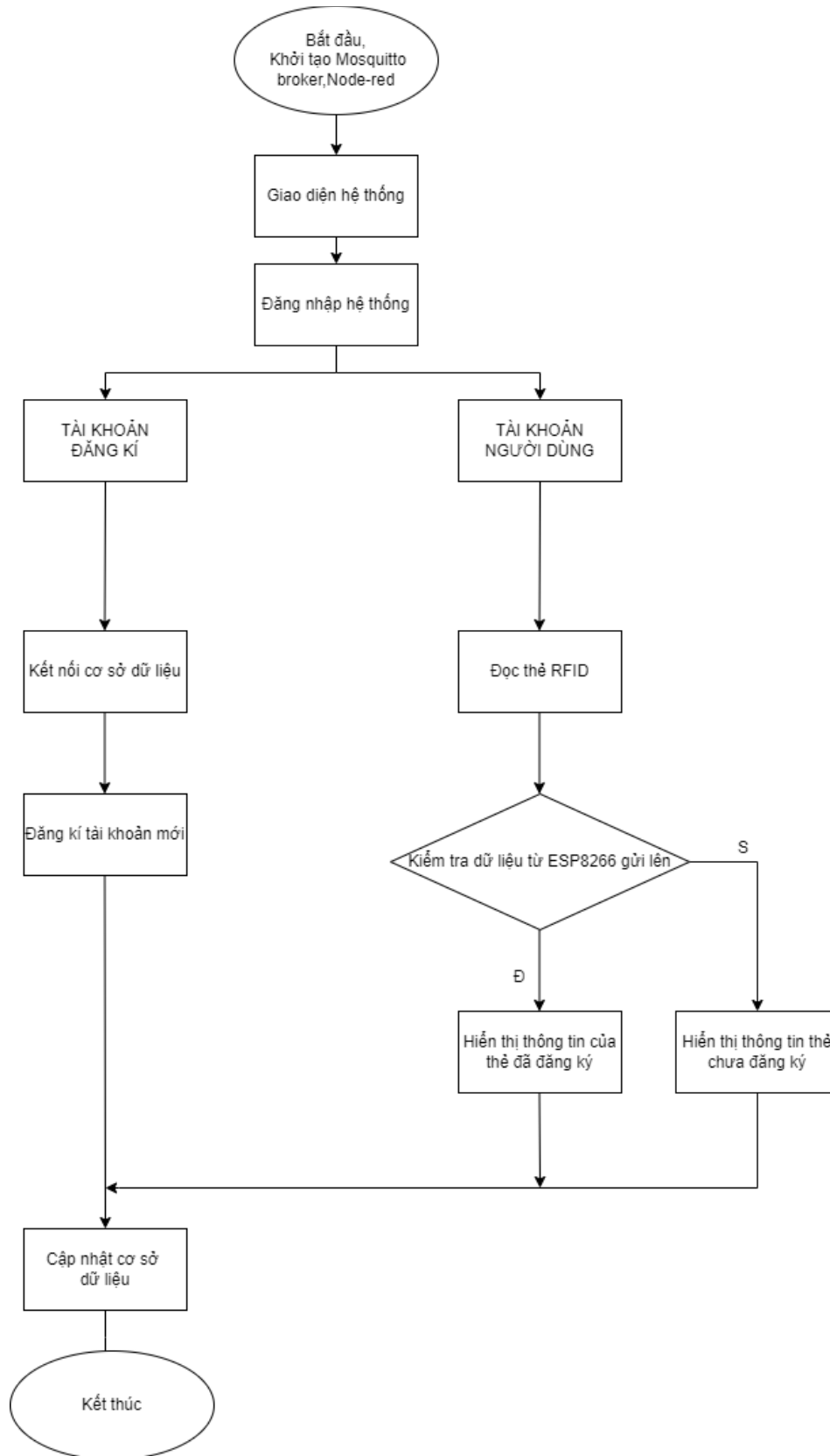


Hình 4.12 Lưu đồ giải thuật cho vi điều khiển module 3

- Giải thích module 3

Sau khi ta kết nối nguồn và nạp chương trình cho vi điều khiển, sau đó ta tiến hành bật Mosquitto Broker và phần mềm Node-red, tiếp theo dữ liệu cảm biến sẽ được gửi qua Mosquitto Broker và hiển thị lên Node-red. Nếu có tín hiệu từ cảm biến hồng ngoại vật cản thì vi điều khiển sẽ xử lý, nếu điều kiện đúng thì bật đèn, điều kiện sai thì không bật đèn. Nếu có tín hiệu liên tục từ cảm biến nhiệt độ, độ ẩm thì sẽ gửi liên tục qua broker và biểu diễn lên Node-red, người dùng có thể theo dõi theo thời gian thực, người dùng cũng có thể điều khiển đèn qua Node-red.

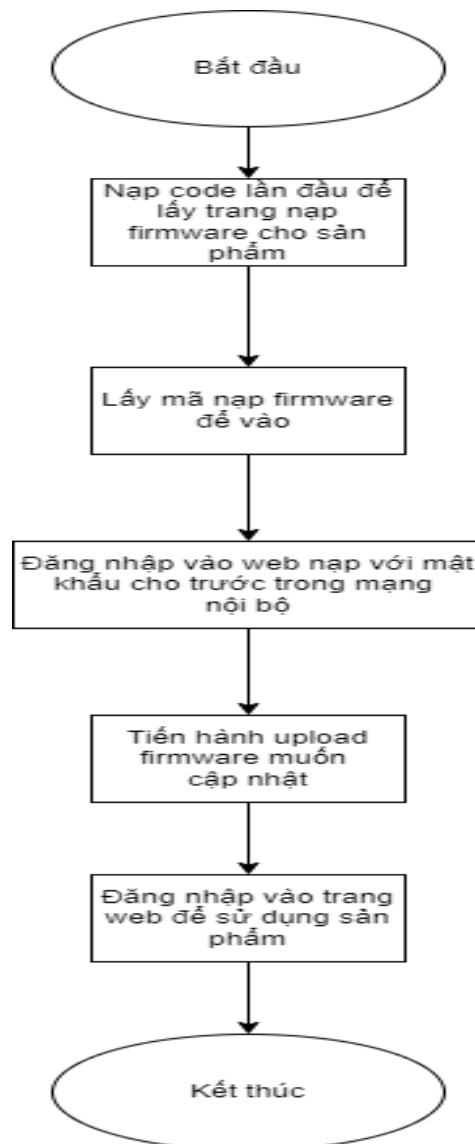
4.2.2 Lưu đồ giải thuật phân bảo mật hệ thống



Hình 4.13 Lưu đồ giải thuật cho bảo mật phân hiển thị

Giải thích lưu đồ:

Khi chúng ta khởi động chương trình cho phép vào giao diện hệ thống. Hệ thống sẽ có một tài khoản quản lý và tạo thêm các user là tài khoản admin, tài khoản được liên kết với cơ sở dữ liệu và có thể theo dõi được trang cơ sở dữ liệu. Vào trang đăng nhập của hệ thống và sử dụng một tài khoản đã được tạo trước đó để đăng nhập. Khi đăng nhập đúng thì sẽ vào trang của tài khoản bạn, trong trang này có các mật khẩu của từng module, nếu nhập đúng thì sẽ được quyền vào phần của từng module. Khi vào module 1 sẽ có các phần như hiển thị người dùng để hiển thị các user đã đăng kí, phần đăng kí user vào cơ sở dữ liệu, phần hiển thông tin người dùng khi quét thẻ. Khi vào module 2 sẽ đưa qua Node-red, có dữ liệu nhiệt độ, độ ẩm lên Node-red theo thời gian thực. Khi vào module 3 sẽ đưa qua Node-red, có dữ liệu nhiệt độ, độ ẩm lên theo thời gian thực và có thể điều khiển bật/tắt đèn qua Node-red.



Hình 4. 14 Lưu đồ giải thuật cho bảo mật phần cứng

Giải thích lưu đồ:

Khi chúng ta muốn cài đặt hệ thống, ta nạp chương trình lần đầu để lấy mã nạp firmware qua OTA. Sau khi chương trình nạp xong, ta sẽ tiến hành loại bỏ cổng nạp vật lý. Khi ta muốn nạp firmware lại thì phải vào đúng mạng nội bộ và truy cập vào trang nạp firmware để cập nhật. Khi ta cập nhật xong thì vào trang web đã được cấp để truy cập vào các module.

4.3 Kết quả thực hiện

4.3.1 Kết quả thi công phần cứng



Hình 4.15 Hình ảnh phần cứng hệ thống

4.3.2 Hình ảnh mô phỏng thực tế

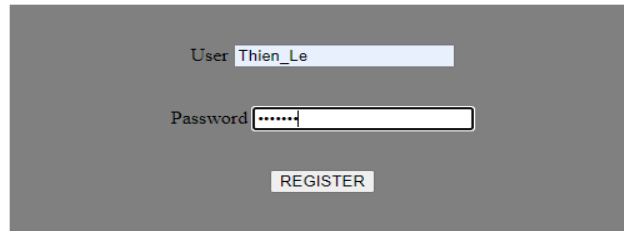
Hình ảnh phần mềm chạy thực tế

Login Form



Hình 4.16 Giao diện trang đăng nhập vào hệ thống

Register Form



Hình 4.17 Giao diện trang đăng kí vào hệ thống

USER PAGE

t

Logout

Module 1

Password: Đăng nhập

Module 2

Password: Đăng nhập

Module 3

Password: Đăng nhập

Hình 4.18 Giao diện trang chủ người dùng

Hệ thống điểm danh, mở cửa, dùng webserver trên server xampp, truy cập trong mạng nội bộ.



Hệ thống điểm danh, mở cửa, dùng webserver trên server xampp, truy cập trong mạng nội bộ.



Hình 4.19 Giao diện trang chủ module 1



Hình 4.20 Giao diện trang chủ module 2



Hình 4.21 Giao diện trang chủ module 3



Please Scan Tag to Display ID or User Data

User Data	
ID	: -----
Name	: -----
Gender	: -----
Email	: -----
Mobile Number	: -----

Hình 4.22 Giao diện trang đọc thẻ và hiển thị module 1



Registration Form

ID

Name

Gender

Email Address

Mobile Number

Hình 4.23 Giao diện trang đăng kí và hiển thị module 1



User Data Table

Name	ID	Gender	Email	Mobile Number	Action
t	13A3DC02	Male	t@gmail.com	0166666	<input type="button" value="Delete"/>

Hình 4.24 Giao diện trang hiển thị thông tin người dùng module 1

4.3.3 Kết quả đạt được

Qua thời gian nghiên cứu tìm hiểu và sự hướng dẫn tận tình của thầy Trần Duy Chung nhóm chúng em đã hoàn thiện được đề tài, 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:

- _Tiếp cận, tìm hiểu, hoàn thiện sản phẩm.
- _Giao tiếp thành công thiết bị dung MQTT & HTTP.
- _Mã hoá bảo mật dữ liệu của từng module riêng biệt.

_Ứng dụng sản phẩm dễ dàng, dễ tiếp cận, dễ sử dụng.

_Nâng cao khả năng nghiên cứu, học hỏi và phát triển.

4.3.4 **Hướng dẫn sử dụng phần mềm**

Sau khi mở trang đăng nhập, chúng ta đăng nhập vào tài khoản admin để đăng kí thêm người dùng hoặc đăng nhập vào tài khoản user đã được đăng kí trước để và trang chủ của user.

➤ Trường hợp 1: Đăng nhập vào tài khoản admin

Tài khoản admin sẽ chỉ được đưa cho người chủ của sản phẩm.

Khi đăng nhập hợp lệ sẽ truy cập vào trang đăng kí.

Ở đây khi ta bấm đăng kí thì sẽ tạo được một tài khoản user mới và đã liên kết với cơ sở dữ liệu.

Tài khoản admin cũng sẽ được cấp địa chỉ trang cơ sở dữ liệu để truy cập vào và quản lý thông tin tài khoản.

➤ Trường hợp 2: Đăng nhập vào tài khoản user

Tài khoản user được đưa cho người dùng hệ thống.

Khi đăng nhập vào hợp lệ sẽ truy cập vào trang chủ của user.

Ở đây ta sẽ thấy có các ô mật khẩu của module, mật khẩu ở đây được cài đặt trước trong chương trình và chỉ người tạo chương trình mới sửa được nên nó sẽ được đưa cho admin để đưa lại cho user.

Nếu nhập đúng mật khẩu của module nào thì module đó sẽ được mở, nhập sai thì sẽ được thông báo là nhập sai và quay về trang chủ của user.

_Đăng nhập vào module 1:

Ở đây ta sẽ thấy một trang chứa các thanh công cụ như Home, Register, ...

Tại tab Home sẽ là thông tin chung về hệ thống module 1

Tại tab User sẽ là thông tin của các user đã đăng kí và liên kết với cơ sở dữ liệu, tại đây ta có thể chỉnh sửa và xoá thông tin của các user. Nhưng chỉ admin là người có thể biết địa chỉ của trang cơ sở dữ liệu.

Tại tab Register sẽ là một form điền thông tin, khi ta chạm thẻ vào module 1 (RFID Reader) thì sẽ hiện lên thông số UID của thẻ. Khi ta đã điền đúng tất cả thông tin theo

form thì bấm nút Register trên trang thì sẽ đăng kí một người dùng trong hệ thống và có liên kết với cơ sở dữ liệu.

Tại tab Hiển thị sẽ là một trang để hiện các thông tin người dùng khi người dùng chạm thẻ vào module. Khi người dùng chạm thẻ vào module này thì sẽ hiện lên thông tin của thẻ và người đang truy cập qua trang có thể theo dõi được.

_Đăng nhập vào module 2:

Ở đây ta sẽ thấy được một giao diện với 2 thông số nhiệt độ, độ ẩm được cập nhật theo thời gian thực

_Đăng nhập vào module 3:

Ở đây ta sẽ thấy được một giao diện với 2 thông số nhiệt độ, độ ẩm được cập nhật theo thời gian thực và 1 công tắc để bật tắt đèn.

NHẬN XÉT VÀ ĐÁNH GIÁ, HƯỚNG PHÁT TRIỂN

Nhận xét và đánh giá

➤ Nhận xét:

Quá trình thực nghiệm được thực hiện tại phòng MakerSpace của khoa Điện - Điện tử được cho ra kết quả rất tốt. Đáp ứng được nhu cầu của người dùng, tạo tiền đề cơ sở để tiếp cận được với lĩnh vực IOT đặc biệt về giao tiếp các thiết bị và dữ liệu.

➤ Đánh giá:

Hiểu được cách thức hoạt động của các module ngoại vi và giao thức truyền nhận dữ liệu qua Internet.

Tạo ra được sản phẩm có tính ứng dụng.

Hệ thống có tính phản hồi ổn định nhưng chưa đáp ứng được nhu cầu ở tần suất cao.

Hướng phát triển

Sau khi nghiên cứu và xây dựng mô hình mô phỏng, để có thể áp dụng vào thực tiễn tốt hơn thì cần phải phát triển thêm các yếu tố sau:

- Triển khai mô hình thành một hệ thống hoàn thiện có liên kết với nhau
- Triển khai thêm nhiều module để đáp theo nhu cầu sử dụng
- Sử dụng nhiều hơn các biện pháp để bảo mật cho hệ thống và các module
- Hoàn thiện hơn về trang web để dễ dàng cho người sử dụng
- Dùng mã hoá bảo mật cấp cao như Prim'X, Dashlane....

KẾT LUẬN

Qua thời gian thực hiện đề tài từ ngày 5/10/2021 đến ngày 10/1/2022 nhóm đã hoàn thành đồ án đúng thời gian dự kiến để báo cáo trước hội đồng. Nhóm thực hiện đề tài đã tạo ra một sản phẩm cụ thể, đó là:

- Tạo nên một hệ thống điều khiển hoặc giám sát các thiết bị ngoại vi.
- Hệ thống IOT có bảo mật.

Sau thời gian nghiên cứu và thi công, nhóm đã thực hiện được công việc như xây dựng phần cứng, viết giao diện quản lý, xây dựng cơ sở dữ liệu cho hệ thống. Hiểu được cách lập trình Arduino, hiểu biết cơ bản về cơ chế và cách thức lập trình PHP và sql phpMyAdmin. Từ những yêu cầu đặt ra khi nghiên cứu, nhóm đã đạt được một số yêu cầu sau:

- Hoàn thiện được hoàn chỉnh mô hình, các bộ đọc quét được mã thẻ nhanh chóng.
- Phần mềm hoàn thiện và hoạt động ổn định.
- Hiểu biết cơ bản về lập trình PHP và cách xây dựng cơ sở dữ liệu trên và sql phpMyadmin.
- Hiểu biết về Node MCU ESP8266, module RFID, cảm biến DHT và các ngoại vi khác.

Ngoài ra còn một số hạn chế như:

Giao diện chưa thân thiện với người dùng.

Mô hình còn chưa tiệm cận với các mô hình thực tế.

Sử dụng module RFID RC522 để đọc tương đối chính xác nhưng vẫn còn chậm, chưa thể đáp ứng với số lượng nhiều người liên tục.

Truyền nhận qua Node-red vẫn còn nhiều hạn chế về phần xử lý luồng dữ liệu.

TÀI LIỆU THAM KHẢO

- [1] Luigi Atzori, Antonio Iera, Giacomo Morabito, Internet of Things: A survey, *Computer Networks* 54 (2010) 2787–2805.
- [2] Dr. Ovidiu Vermesan, Dr. Peter Friess, Patrick Guillemin, Internet of Things Strategic Research Roadmap, 2009 Strategic Research Agenda, The IoT European Research Cluster - European Research Cluster on the Internet of Things (IERC).
- [3] Everton Cavalcante, Marcelo Pitanga Alves, An Analysis of Reference Architectures for the Internet of Things, Corba 2015.
- [4] Anna Ha’c, Wireless Sensor Network Designs, University of Hawaii at Manoa, Honolulu, USA, John Wiley & Sons Ltd, Copyright 2003.
- [5] Edgar H.Callaway, Jr. Wireless Sensor Networks: Architectures and Protocols, A CRC Press Company, Copyright © 2004 CRC Press LLC.
- [6] Ovidiu Vermesan, Peter Friess, Internet of Things – Converging Technologies For Smart Environments and Integrated Ecosystems, River Publishers Series in Communications.
- [7] Kiran Maraiya, Kamal Kant, Nitin Gupta, Application based Study on Wireless Sensor Network, *International Journal of Computer Application* (0975-8887), Volume 21, No.8, May 2011.
- [8] I.F. Akyildiz, W. Su*, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: A survey, *Broadband and Wireless Networking Laboratory, School of Electrical and Computer Engineering, Georgia Institute of Technology, Atlanta, GA 30332*, Received 12 December 2001; accepted 20 December 2001. <<http://www.ece.gatech.edu/research/labs/bwn/sensornets.pdf>>.
- [9] Mainwaring, Polastre, et al. Wireless Sensor Networks For Habitat Monitoring, online posting. 2002 ACM International Workshop on Wireless Sensor Networks and Applications September 28, 2002. Atlanta, GA. (also Intel Research, IRB-TR-02-006, June 2002.) 12 Dec 2002. <<http://www.cs.berkeley.edu/~polastre/papers/wsna02.pdf>>.
- [10] Matt Richardson & Shawn Wallace, Getting Started with Raspberry

PHỤ LỤC

➤ Code cho module 1

```
#include <Arduino.h>
#include <ESP8266WiFi.h>
#include <ESPAsyncTCP.h>
#include <ESPAsyncWebServer.h>
#include <AsyncElegantOTA.h>
// #include <ESP8266WebServer.h>
#include <ESP8266HTTPClient.h>
#include <WiFiClient.h>

WiFiClient wifiClient;
const char *host = "http://192.168.43.56/MODULE_RFID_DATN/getUID.php";
const char* ssid = ".";
const char* password = "1303992020";

#include <SPI.h>
#include <MFRC522.h>
#define SS_PIN D0 //--> SDA / SS is connected to pinout D2
#define RST_PIN D3 //--> RST is connected to pinout D1
MFRC522 mfrc522(SS_PIN, RST_PIN); //--> Create MFRC522 instance.

#define ON_Board_LED 2 //--> Defining an On Board LED, used for indicators when
the process of connecting to a wifi router

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



```
LiquidCrystal_I2C lcd(0x27, 16, 2);
#define I2C_SDA D2
#define I2C_SCL D1
#define Relay 10 //opto
//const int BUZ = 4;
#define BUZ D4
AsyncWebServer async_server(81);

AsyncWebServer server(80);
//ESP8266WebServer server(80); //--> Server on port 80

int readsuccess;
byte readcard[4];
char str[32] = "";
String StrUID;

void setup() {
  Serial.begin(115200);
  WiFi.mode(WIFI_STA);
  SPI.begin();    //--> Init SPI bus
  mfrc522.PCD_Init(); //--> Init MFRC522 card
  Wire.begin(I2C_SDA, I2C_SCL);
  lcd.begin();
  lcd.backlight();
  lcd.setCursor(0, 0);
  lcd.print("Hello! Open here");
  delay(500);
```

```
WiFi.begin(ssid, password);
Serial.println("");

pinMode(ON_Board_LED,OUTPUT);
digitalWrite(ON_Board_LED, HIGH); //--> Turn off Led On Board

//-----Wait for connection
Serial.print("Connecting");

// Wait for connection
while (WiFi.status() != WL_CONNECTED) {
  delay(500);
  Serial.print(".");
  digitalWrite(ON_Board_LED, LOW);
  delay(250);
  digitalWrite(ON_Board_LED, HIGH);
  delay(250);
}
digitalWrite(ON_Board_LED, HIGH); //--> Turn off the On Board LED when it is
connected to the wifi router.

Serial.println("");
Serial.print("Connected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP());
```

```
async_server.on("/", HTTP_GET, [](AsyncWebServerRequest *request) {  
    request->send(200, "text/plain", "Hello world.");  
});  
  
AsyncElegantOTA.begin(&async_server,"TL","123456");  
async_server.begin();  
Serial.println("HTTP server started");  
}
```

```
void loop() {
  digitalWrite(BUZ,HIGH);
  lcd.setCursor(0, 0);
  lcd.print("Hello! Open here");
  // put your main code here, to run repeatedly
  readsuccess = getid();

  if(readsuccess) {
    digitalWrite(ON_Board_LED, LOW);
    HTTPClient http; //Declare object of class HTTPClient

    String UIDresultSend, postData;
    UIDresultSend = StrUID;

    //Post Data
    postData = "UIDresult=" + UIDresultSend;

    http.begin(wifiClient,host);
    // "http://192.168.0.0/NodeMCU_RC522_Mysql/getUID.php"); //Specify request
    destination
    http.addHeader("Content-Type", "application/x-www-form-urlencoded");
    //Specify content-type header

    int httpCode = http.POST(postData); //Send the request
    String payload = http.getString(); //Get the response payload

    Serial.println(UIDresultSend);
  }
}
```

```
Serial.println(httpCode); //Print HTTP return code
Serial.println(payload); //Print request response payload

http.end(); //Close connection
delay(1000);
digitalWrite(ON_Board_LED, HIGH);
}
AsyncElegantOTA.loop();
}
int getid() {
  if(!mfr522.PICC_IsNewCardPresent()) {
    return 0;
  }
  if(!mfr522.PICC_ReadCardSerial()) {
    return 0;
  }
  Serial.print("THE UID OF THE SCANNED CARD IS : ");
  for(int i=0;i<4;i++){
    readcard[i]=mfr522.uid.uidByte[i]; //storing the UID of the tag in readcard
    array_to_string(readcard, 4, str);
    StrUID = str;
  }
  ////////////LCD-INTERFACE//////////
  if(StrUID == "13A3DC02") //"D9 AB 1D B4" //13 A3 DC 02 //FA 8A 79 82
  {
    //accessGranted();
    digitalWrite(BUZ, LOW);
```

```
delay(500);
digitalWrite(BUZ, HIGH);
delay(500);
lcd.setCursor(0, 0);
lcd.print("Access - Right");
lcd.setCursor(0, 1);
lcd.print("LEPHAMVINHTHIEN");
Serial.print("Right");
digitalWrite(Relay,HIGH);
delay(5000);
digitalWrite(Relay,LOW);
lcd.clear();
delay(1000);
}
else
{
  lcd.setCursor(0, 0);
  lcd.print("Access - Wrong");
  Serial.print("Wrong");
  digitalWrite(Relay,LOW);
  digitalWrite(BUZ, LOW);
  delay(2000);
  lcd.clear();
  delay(1000);
}
mfrc522.PICC_HaltA();
return 1;
```

```
}  
/-//  
//-----Procedure to change the result of reading an array  
UID into a string-----//  
void array_to_string(byte array[], unsigned int len, char buffer[]) {  
    for (unsigned int i = 0; i < len; i++)  
    {  
        byte nib1 = (array[i] >> 4) & 0x0F;  
        byte nib2 = (array[i] >> 0) & 0x0F;  
        buffer[i*2+0] = nib1 < 0xA ? '0' + nib1 : 'A' + nib1 - 0xA;  
        buffer[i*2+1] = nib2 < 0xA ? '0' + nib2 : 'A' + nib2 - 0xA;  
    }  
    buffer[len*2] = '\0';  
}
```

➤ Code cho module 2

```
/******  
Module 2 RFID DHT, DO AM DAT, MOTOR  
DK DONG CO TU DONG (KHONG QUA APP); APP THEO DOI NHIET  
DO,DO AM DAT VA TUOI TU DONG.  
*****/  
  
#include <ESP8266WiFi.h>  
#include <PubSubClient.h>  
#include <ArduinoJson.h>  
const char* ssid = ".";  
const char* password = "1303992020";  
const char* mqtt_server = "192.168.43.56";  
  
// Initializes the espClient. You should change the espClient name if you have  
multiple ESPs running in your home automation system  
WiFiClient espClient22;  
PubSubClient client(espClient22);  
String ChuoiJson="";  
#include <DHT.h>  
#define DHTPIN D1  
#define DHTTYPE DHT11  
DHT dht(DHTPIN, DHTTYPE);  
#define SOIL_MOIST_1_PIN A0 // Chân A0 nối với cảm biến độ ẩm đất  
int soilMoist;  
float Temp = 0 ;  
float Humi = 0 ;  
  
// Lamp - LED - GPIO 4 = D2 on ESP-12E NodeMCU board  
const int led = D5;
```



```
const int relay = D2;

boolean DV1_State = digitalRead(led);
boolean DV2_State = digitalRead(relay);

int last = 0;
String device = "device";
const char* inTopic = "room/light";
const char* outTopic = "room/sensors";

//Connect your NodeMCU to your router
void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - NodeMCU IP address: ");
  Serial.println(WiFi.localIP());
}
StaticJsonDocument<200> jsonObj;
void callback(char* topic, byte* payload, unsigned int length)
```

```
{//topic là để xác định kênh đã đăng kí nhận dữ liệu, payload là mảng chứa dữ liệu
nhận được từ broker, length là độ dài dữ liệu

String strPayload = String((char*)payload).substring(0, length); //Chuỗi nhận
được từ broker

Serial.println("Receive: " + strPayload+"\r\n");

if (String(topic) == inTopic)
{
DeserializationError error = deserializeJson(jsonObj, payload);// đọc dữ liệu từ
chuỗi json (payload)

if (error)

Serial.println("Xay ra loi khi doc chuoi JSON");

else
{
// Lấy dữ liệu từ đối tượng JSON

String st_dv1 = jsonObj["device1"].as<String>();
String st_dv2 = jsonObj["device2"].as<String>();

//Xử lý

if(st_dv1 == "1" )
{digitalWrite(led,HIGH); }

if(st_dv1 == "0" )
{digitalWrite(led,LOW); }

if(st_dv2 == "1")
{digitalWrite(relay,HIGH); }

if(st_dv2 == "0")
{digitalWrite(relay,LOW); }

}
}
```

```
}  
}  
  
// This functions reconnects your ESP8266 to your MQTT broker  
// Change the function below if you want to subscribe to more topics with your  
ESP8266  
void reconnect() {  
    // Loop until we're reconnected  
    while (!client.connected()) {  
        Serial.print("Attempting MQTT connection...");  
  
        if (client.connect("ESP8266Client22")) {  
            Serial.println("connected");  
            // Subscribe or resubscribe to a topic  
            // You can subscribe to more topics (to control more LEDs in this example)  
            // client.subscribe("room/light");  
            // client.subscribe(outTopic);  
            client.subscribe(inTopic);  
        } else {  
            Serial.print("failed, rc=");  
            Serial.print(client.state());  
            Serial.println(" try again in 5 seconds");  
            // Wait 5 seconds before retrying  
            delay(5000);  
        }  
    }  
}
```

```
// The setup function sets your ESP GPIOs to Outputs, starts the serial
communication at a baud rate of 115200

// Sets your mqtt broker and sets the callback function

// The callback function is what receives messages and actually controls the LEDs

void setup()
{
  pinMode(led, OUTPUT);
  pinMode(relay, OUTPUT);
  // pinMode(CBHN, INPUT);
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);

  last=millis();
  dht.begin();
  readdad();
}

// For this project, you don't need to change anything in the loop function. Basically
it ensures that the NodeMCU is connected to MQTT broker

void loop()
{
  if (!client.connected())
  {
    reconnect();
  }
  if(!client.loop())
```

```
{
  client.connect("ESP8266Client22");
}
if(millis() - last >= 1000)
{
  readsensors();
  dataJson(String (device) , String (Temp) , String (Humi) , String (DV1_State),
String (DV2_State) );
  readdad();
  // IN thông tin ra màn hình
  Serial.print("Do am dat: ");
  Serial.print(soilMoist);
  Serial.println(" %");
  if(soilMoist <= 42)
  {digitalWrite(relay,HIGH);}
  if(soilMoist >= 82)
  {digitalWrite(relay,LOW); }
  last=millis();
}
}
void readsensors()
{
  Temp = dht.readTemperature();
  Humi = dht.readHumidity();
}
int getSoilMoist()
{
```

```
int i = 0;

int anaValue = 0;

for (i = 0; i < 10; i++) //
{
    anaValue += analogRead(SOIL_MOIST_1_PIN); //Đọc giá trị cảm biến độ ẩm
đất
    delay(50); // Đợi đọc giá trị ADC
}

anaValue = anaValue / (i);

anaValue = map(anaValue, 1023, 0, 0, 100); //Ít nước:0% ==> Nhiều nước 100%

return anaValue;
}

void readdad(void)
{
    soilMoist = getSoilMoist();    //Đọc cảm biến độ ẩm đất
}

void dataJson(String device , String Temp , String Humi , String DV1_State, String
DV2_State )
{
    ChuoiJson= "";
    ChuoiJson= "{\"from\":\"" + String(device)+"\", " +
        "\"Temp\":\"" + String(Temp)+"\", " +
        "\"Humi\":\"" + String(Humi)+"\", " +
        "\"DV1_State\":\"" + String(DV1_State)+"\", " +
        "\"DV2_State\":\"" + String(DV2_State)+"\"}";
}
```

```
StaticJsonDocument<256>doc;  
deserializeJson(doc, ChuoiJson);  
char out[256];  
int b =serializeJson(doc, out);  
//Serial.print("publishing bytes = ");  
//Serial.println(b,DEC);  
boolean rc = client.publish(outTopic, out);  
doc.clear();  
}
```

➤ Code cho module 3

```
/*  
Module 3 MODULE RELAY, CAM BIEN HONG NGOAI, CB NHIET DO  
DK LED BANG APP, DK LED BANG CAM BIEN (NODE-RED; MQTT)  
*/  
#include <ESP8266WiFi.h>  
#include <PubSubClient.h>  
#include <ArduinoJson.h>  
// Change the credentials below, so your ESP8266 connects to your router  
const char* ssid = ".";  
const char* password = "1303992020";  
  
// Change the variable to your Raspberry Pi IP address, so it connects to your MQTT broker  
const char* mqtt_server = "192.168.43.56";  
  
// Initializes the espClient. You should change the espClient name if you have multiple ESPs  
running in your home automation system  
WiFiClient espClient11;  
PubSubClient client(espClient11);  
String ChuoiJson="";  
#include <DHT.h>  
#define DHTPIN D8  
#define DHTTYPE DHT11  
DHT dht(DHTPIN, DHTTYPE);  
#define CBHN D7  
int HNVALUE;  
float Temp = 0 ;  
float Humi = 0 ;  
// Lamp - LED - GPIO 4 = D2 on ESP-12E NodeMCU board  
const int led1 = D2;  
const int led2 = D3;  
boolean DV1_State = digitalRead(led1);  
boolean DV2_State = digitalRead(led2);  
  
int last = 0;  
String device = "device";  
//const char* inTopic = "room/light";  
//const char* outTopic = "room/sensors";
```



```
const char* outTopic = "room/demo2"; //Kênh này để esp đẩy dữ liệu lên broker Đăng kí
topic(kênh truyền),muốn nhận thì phải subscribe đúng kênh truyền.
const char* inTopic = "room/demo1"; //Kênh này để esp nhận dữ liệu từ broker

//Connect your NodeMCU to your router
void setup_wifi() {
  delay(10);

  Serial.println();

  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(100);
    Serial.print(".");
  }
  Serial.println("");
  Serial.print("WiFi connected - NodeMCU IP address: ");
  Serial.println(WiFi.localIP());
}
// This functions is executed when some device publishes a message to a topic that your
NodeMCU is subscribed to

StaticJsonDocument<200> jsonObj;
void callback(char* topic, byte* payload, unsigned int length)
{//topic là để xác định kênh đã đăng kí nhận dữ liệu, payload là mảng chứa dữ liệu nhận được từ
broker, length là độ dài dữ liệu
  String strPayload = String((char*)payload).substring(0, length); //Chuỗi nhận được từ broker
  Serial.println("Receive: " + strPayload+"\r\n");
  if (String(topic) == inTopic)
  {
    DeserializationError error = deserializeJson(jsonObj, payload);// đọc dữ liệu từ chuỗi json
(payload)
    if (error)
      Serial.println("Xay ra loi khi doc chuoai JSON");
    else
```

```
{
  // Lấy dữ liệu từ đối tượng JSON
  String st_dv1 = jsonObj["device1"].as<String>();
  String st_dv2 = jsonObj["device2"].as<String>();
  //Xử lý
  // if(st_dv1 == "1" )
  // {digitalWrite(led1,HIGH); }
  //
  // if(st_dv1 == "0" )
  // {digitalWrite(led1,LOW); }

  if(st_dv2 == "1")
    {digitalWrite(led2,HIGH); }
  if(st_dv2 == "0")
    {digitalWrite(led2,LOW); }
}
}
}

// This functions reconnects your ESP8266 to your MQTT broker
// Change the function below if you want to subscribe to more topics with your ESP8266
void reconnect() {
  // Loop until we're reconnected
  while (!client.connected()) {
    Serial.print("Attempting MQTT connection...");
    if (client.connect("ESP8266Client11")) {
      Serial.println("connected");
      // Subscribe or resubscribe to a topic
      // You can subscribe to more topics (to control more LEDs in this example)
      // client.subscribe("room/light");
      // client.subscribe(outTopic);
      client.subscribe(inTopic);
    } else {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      // Wait 5 seconds before retrying
      delay(5000);
    }
  }
}
```

```
}  
}  
  
// The setup function sets your ESP GPIOs to Outputs, starts the serial communication at a baud  
rate of 115200  
// Sets your mqtt broker and sets the callback function  
// The callback function is what receives messages and actually controls the LEDs  
void setup()  
{  
  pinMode(led1, OUTPUT);  
  pinMode(led2, OUTPUT);  
  pinMode(CBHN, INPUT);  
  Serial.begin(115200);  
  setup_wifi();  
  client.setServer(mqtt_server, 1883);  
  client.setCallback(callback);  
  last=millis();  
  dht.begin();  
  digitalWrite(led1,LOW);  
  digitalWrite(led2,LOW);  
}  
  
// For this project, you don't need to change anything in the loop function. Basically it ensures  
that the NodeMCU is connected to MQTT broker  
void loop()  
{  
  READ_CBHN();  
  if (!client.connected())  
  {  
    reconnect();  
  }  
  if(!client.loop())  
    client.connect("ESP8266Client11");  
  if(millis() - last >= 1000)  
  {  
    readsensors();  
    dataJson(String (device) , String (Temp) , String (Humi) , String (DV1_State), String  
(DV2_State) );  
  }  
}
```

```
    last=millis();
  }
}
void READ_CBHN(void)
{
  HNVALUE = digitalRead(CBHN);
  // Serial.println(HNVALUE);
  if( HNVALUE == 0 )
  {
    digitalWrite(led1,HIGH);
    delay(2000);
    digitalWrite(led1,LOW);
  }
  else
  { digitalWrite(led1,LOW); }
}
void readsensors()
{
  Temp = dht.readTemperature();
  Humi = dht.readHumidity();
}
void dataJson(String device , String Temp , String Humi , String DV1_State, String DV2_State )
{

  ChuoiJson= "";
  ChuoiJson= "{\"from\":\"" + String(device)+"\" ," +
    "\"Temp\":\"" + String(Temp)+"\" ," +
    "\"Humi\":\"" + String(Humi)+"\" ," +
    "\"DV1_State\":\"" + String(DV1_State)+"\" ," +
    "\"DV2_State\":\"" + String(DV2_State)+"\"}";

  StaticJsonDocument<256>doc;
  deserializeJson(doc,ChuoiJson);
  char out[256];
  int b =serializeJson(doc, out);
  //Serial.print("publishing bytes = ");
  //Serial.println(b,DEC);
  boolean rc = client.publish(outTopic, out);
```

```
doc.clear();  
}
```