

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC BÁCH KHOA

KHOA ĐIỆN – ĐIỆN TỬ

BỘ MÔN KỸ THUẬT ĐIỆN

-----o0o-----



ĐỀ CƯƠNG MÔN HỌC

THIẾT KẾ MÔ HÌNH NHÀ THÔNG MINH SỬ DỤNG

MẠNG KHÔNG DÂY

GVHD: TS. TRỊNH HOÀNG HƠN

SVTH: DƯƠNG QUỐC HOÀNG MSSV: 1711377

Mục Lục

Lời cảm ơn.....	1
Tóm tắt đề tài.....	2
Chương 1. Giới thiệu đề tài.....	3
1.1. Tổng quan về Nhà thông minh.....	3
1.2. Một số tính năng của Nhà thông minh	3
1.2.1 Tiết kiệm năng lượng	3
1.2.2 Ánh sáng thông minh	3
1.2.3 Đảm bảo an ninh cho ngôi nhà	3
Chương 2. Thiết kế tổng quan hệ thống.....	4
2.1. Lựa chọn giao tiếp các thiết bị trong nhà	4
2.2. Thiết kế về giao diện giao tiếp với người dùng.....	5
2.3. Thiết kế tổng quan của hệ thống	6
Chương 3. Thiết kế các thiết bị trong nhà	9
3.1. Giao tiếp Zigbee	9
3.1.1. Cơ sở lý thuyết	9
3.1.2. Thiết kế giao tiếp giữa các thiết bị.....	11
3.2. Thiết kế các thiết bị.....	14
3.2.1. Thiết kế các khối dùng chung.....	14
3.2.2. Thiết kế bộ điều khiển trung tâm Hub.....	17
3.2.3. Thiết kế node sensor.....	18
3.2.4. Thiết kế node button	24
3.2.5. Thiết kế node relay	28
3.2.6. Thiết kế node relay ADE.....	33
Chương 4. Thiết kế website	44
4.1. Front-end	44
4.1.1. Giới thiệu.....	44
4.1.2. Các ngôn ngữ cơ bản.....	45
4.1.3. Các Framework sử dụng.....	46
4.2. Back-end.....	49
4.2.1. Giới thiệu.....	49
4.2.2. Vai trò của back-end	50
4.2.3. NodeJs.....	51

4.2.4. ExpressJs	52
4.3. Database	53
4.3.1. Giới thiệu.....	53
4.3.2. MongoDB	54
4.3.3. Thiết kế.....	56
Chương 5. Thiết kế kết nối giữa thiết bị và website	57
5.1. Giới thiệu về giao thức MQTT	57
5.1.1. Khái niệm	57
5.1.2. Cách hoạt động	57
5.2. Thiết kế dữ liệu truyền thông qua MQTT	59
5.2.1. Cấu trúc gói tin từ hub lên server	59
5.2.2. Cấu trúc gói tin từ server xuống hub	60
Chương 6. Kết luận và hướng phát triển	62
6.1. Kết luận	62
6.2. Hướng phát triển	62
Tài liệu tham khảo.....	63

Lời cảm ơn

Đầu tiên, Em xin chân thành cảm ơn các thầy, cô giáo trong Trường ĐHBK Tp.HCM nói chung và các thầy, cô giáo Khoa Điện – Điện tử nói riêng đã giảng dạy em trong thời gian những năm em học ở trường ĐHBK. Giúp em có kiến thức nền tảng để phát triển và học tập tốt trong suốt quá trình học tập.

Em chân thành biết ơn đến Thầy Trịnh Hoàng Hôn, người đã quan tâm giúp đỡ và tận tình hướng dẫn em hoàn thành đồ án này trong thời gian qua, người đã đưa ra đề nghị em đến với đề tài lần này, người đã luôn kiên nhẫn giải đáp những thắc mắc của em trong quá trình làm đồ án này.

Em chân thành biết ơn đến gia đình của em, những người đã không ngừng hỗ trợ, khuyến khích và thúc đẩy em cố gắng trong quá trình học tập để em có được những kiến thức như ngày hôm nay.

Em xin chân thành cảm ơn đến những người bạn đã hỗ trợ em hết mình trong quá trình làm đồ án môn học của mình, giúp em có thêm những kiến thức khác từ trong quá trình học tập và cả trong quá trình làm việc.

Em chân thành cảm ơn tới những người đã hiểu và quan tâm đến em !

Tp. Hồ Chí Minh, ngày 26 tháng 12 năm 2022

Sinh viên

Tóm tắt đề tài

Đề tài này trình bày về các vấn đề chính sau:

- Giới thiệu về nhà thông minh.
- Giới thiệu về các chuẩn giao tiếp.
- Thiết kế tổng quan về mô hình nhà thông minh.
- Thiết kế phần cứng, phần mềm cho các thiết bị.

Chương 1. Giới thiệu đề tài

1.1. Tổng quan về Nhà thông minh

Ngày nay với sự phát triển rất nhanh về công nghệ với cuộc cách mạng 4.0, Hệ thống IoT đã và đang phát triển rất mạnh mẽ. Dựa theo hệ thống IoT phát triển thì Smart home hay nhà thông minh cũng đang trên đà phát triển rất mạnh mẽ.

Smart home hay Nhà thông minh là kiểu nhà được lắp đặt các thiết bị điện tử có thể được điều khiển hoặc tự động hóa hoặc bán tự động hóa. Thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển. Hệ thống điện tử ngày nay giao tiếp với người dùng thông qua bảng điện tử trong nhà, ứng dụng trên điện thoại di động, máy tính bảng hoặc một giao diện web.

Trong một căn nhà thông minh hay Smart home, mọi nơi sẽ được kiểm soát bằng thiết bị điện tử. Chúng sẽ sử dụng các cách giao tiếp riêng để hiểu nhau như: Bluetooth, Zigbee, Z-wave, wifi, KNX,...

1.2. Một số tính năng của Nhà thông minh

1.2.1 Tiết kiệm năng lượng

Nhà thông minh giúp tiết kiệm năng lượng vì bây giờ không cần phải đốt tiền bằng cách quên tắt máy lạnh khi bạn đi. Bằng cách theo dõi hệ thống sưởi, nước, điện của bạn, sẽ rất có ít khả năng chi trả hóa đơn tiền điện quá mức.

1.2.2 Ánh sáng thông minh

Hệ thống ánh sáng thông minh luôn là điểm nhấn đặc biệt và phổ biến nhất khi dùng Nhà thông minh. Ánh sáng thông minh sẽ giúp bạn rất nhiều trong cuộc sống, ví dụ như:

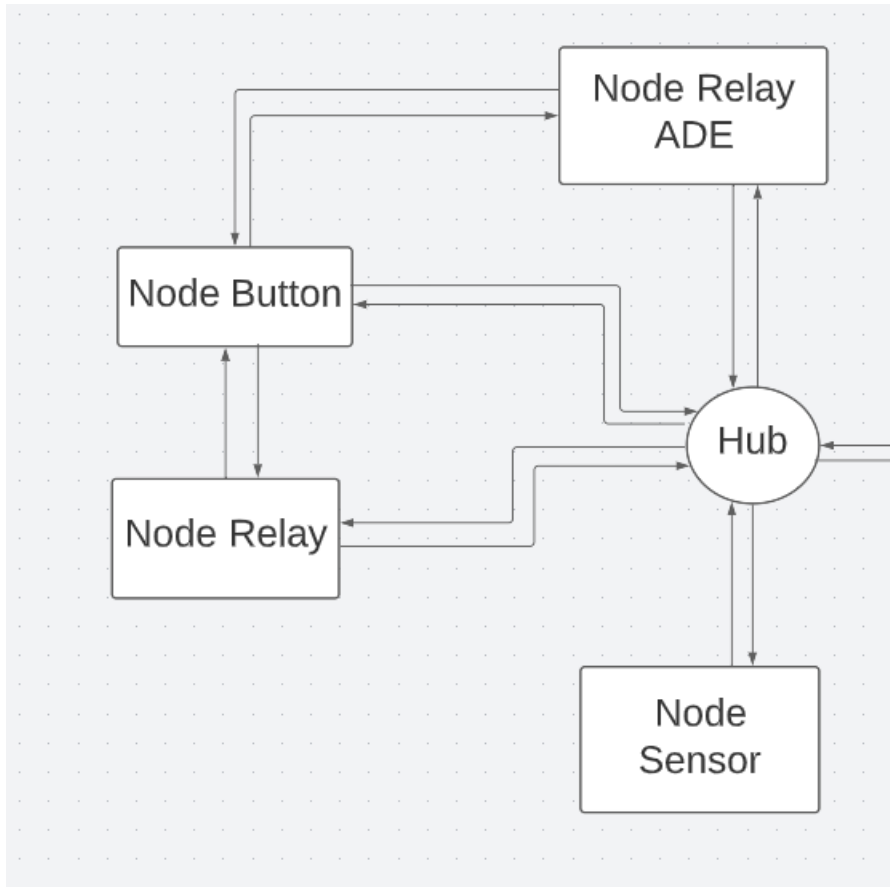
- Bật/ tắt điện ngay từ điện thoại của bạn dù bạn đang ở bất cứ đâu.
- Hẹn giờ lên lịch hoạt động cho các bóng đèn trong nhà. Và bạn sẽ không phải đi bật từng bóng đèn khi trời tối hay tắt chúng vào mỗi sáng.
- Kết hợp cùng các thiết bị điện tử khác ví dụ như các cảm biến. Đây là điểm mà chỉ có các thiết bị chiếu sáng thông minh có thể làm.

1.2.3 Đảm bảo an ninh cho ngôi nhà

Sử dụng các camera thông minh và thiết bị an ninh để bảo vệ căn nhà của bạn. Những camera chất lượng cao thường thiết kết nối trực tiếp với điện thoại của bạn. Và có thể kết hợp với các cảnh báo để thông báo khi có người lạ xuất hiện hay các vấn đề cảnh báo khác nếu có.

Chương 2. Thiết kế tổng quan hệ thống

2.1. Lựa chọn giao tiếp các thiết bị trong nhà



Hình 2.1.1: Tổng quan về các thiết bị trong nhà

Trong hệ thống nhà thông minh, các thiết bị điện tử giao tiếp với nhau thông qua một giao tiếp chung, một số giao tiếp như: Z-wave, Bluetooth, Zigbee, wifi...

Giới thiệu về các giao tiếp:

- Z-wave: Giao thức kết nối này rất nhanh, sử dụng băng tần 908.42MHz, thấp hơn rất nhiều so với các thiết bị không dây khác trong nhà (sử dụng tần số 2.4Ghz), giúp giảm nhiễu và cải thiện hiệu quả giao tiếp cũng như tốc độ. Tương thích với hơn 1500 thiết bị. Z-wave cũng giúp tiết kiệm năng lượng khi sử dụng.
- Wifi: Các thiết bị tương thích sử dụng băng tần 2.4Ghz và 5Ghz, cho khoảng cách khả dụng thực tế lên đến 20 mét, bị ảnh hưởng bởi các yếu tố ngoại cảnh, độ nhiễu, nguồn phát và chất lượng ăng ten. Có thể kết nối tối đa 256 thiết bị cùng lúc, nhưng có khả năng nghẽn mạng xảy ra. Điểm tuyệt vời là tốc độ truyền phát, khoảng cách nhưng cũng có khả năng bị nhiễu tín hiệu vì có quá nhiều thiết bị và vấn đề tiêu hao năng lượng.

- Zigbee: Giao thức này cho khả năng giao tiếp nhanh, sử dụng tần số vô tuyến tiêu chuẩn IEEE 802.15.14, hoạt động trên các băng tần 2.4GHz, 900MHz và 868MHz. Một trong những lợi ích lớn nhất của giao thức kết nối nhà thông minh này chính là khả năng tiêu thụ điện năng siêu thấp, mang lại hiệu quả cực cao. ZigBee cũng giúp cải thiện bảo mật, là một trong những giao thức an toàn nhất tính đến thời điểm hiện tại, sử dụng mã hóa 128Bit, được áp dụng cho các hệ thống tài chính, ngân hàng.
- Bluetooth: Tính tương thích của giao thức kết nối này cực kỳ tuyệt vời, người dùng có thể kết nối bất kỳ thiết bị Bluetooth nào với thiết bị khác rất dễ dàng. Một trong những điều tuyệt vời nhất của kết nối Bluetooth chính là tiết kiệm năng lượng hơn rất nhiều so với các giải pháp khác, lý tưởng cho những ai quan tâm đến tiêu hao điện năng và môi trường. Tuy nhiên, giao thức kết nối này gặp vấn đề về khoảng cách, thiết bị sẽ không làm việc nếu di chuyển ra khỏi tầm khả dụng, mặc dù các nghiên cứu mới đang được tiến hành, cho phép tiết kiệm năng lượng trong khi mở rộng phạm vi sử dụng.

Dựa theo tiêu chí: Mở rộng sau này, tiết kiệm năng lượng, tính bảo mật, khoảng cách giao tiếp giữa các thiết bị.

Tiêu chí	Z-Wave	Wifi	Zigbee	Bluetooth
Mở rộng	232 thiết bị	256 thiết bị	56554 thiết bị	Số lượng lớn
Năng lượng	Tốt	Tiêu hao lớn	Tốt	Rất Tốt
Bảo mật	Tương đối	Tương đối	cao	Tương đối
Khoảng cách	30m	20m	50 – 100m	Dưới 10m

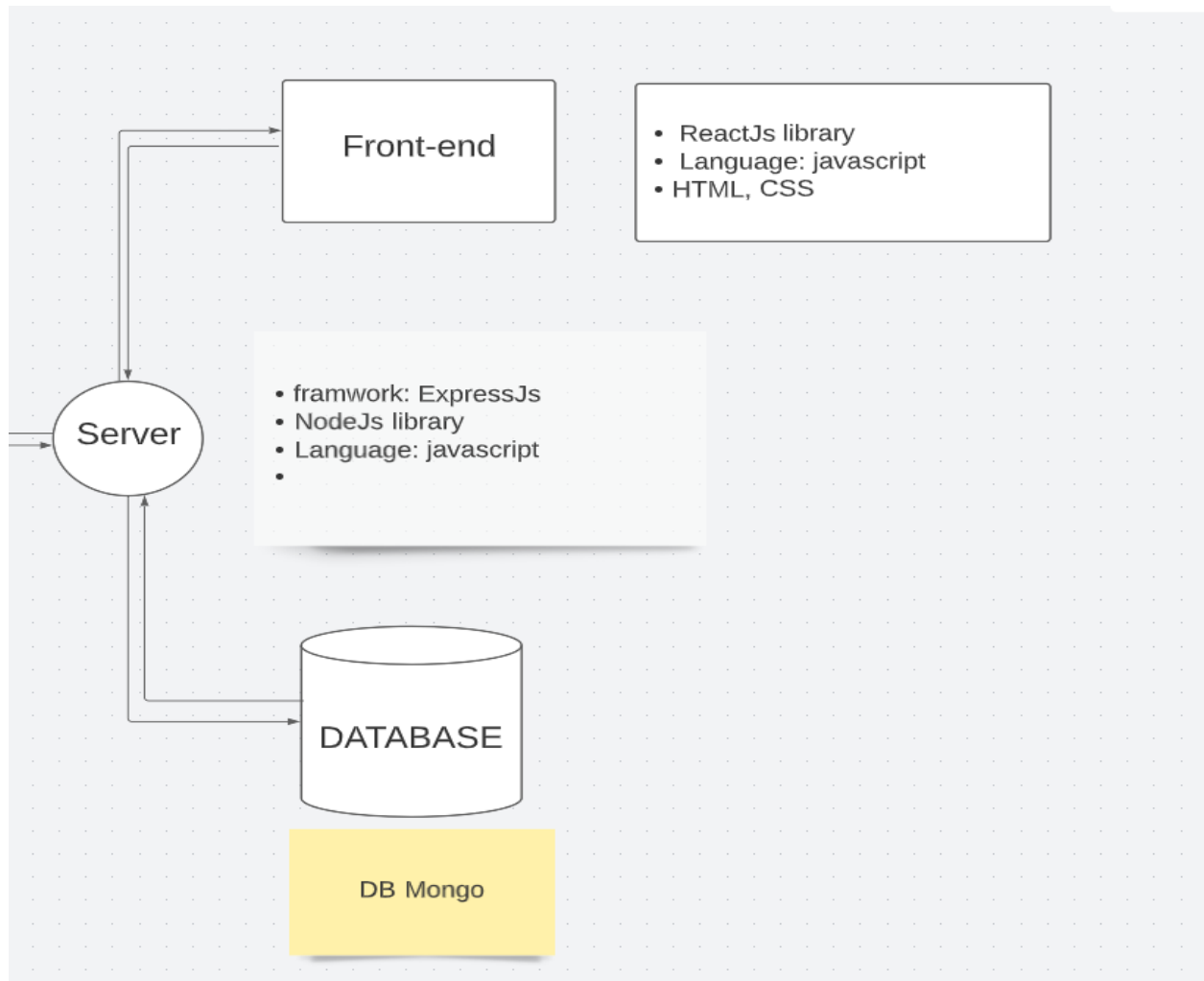
- Dựa theo những tiêu chí trên ta có thể thấy được với số lượng thiết bị kết nối lên tới hơn 56 000 thiết bị và có bảo mật tốt, khoảng cách cao thì lựa chọn mạng Zigbee là hợp lý nhất.

2.2. Thiết kế về giao diện giao tiếp với người dùng

Để người dùng có thể giao tiếp với các thiết bị trong nhà thì người dùng phải thông qua một thiết bị khác để giao tiếp với các thiết bị, ví dụ như: Bảng điện tử trong nhà, thông qua một trang web, ứng dụng có sẵn trên điện thoại,...

Để có tính bảo mật cao thì sẽ loại bỏ các ứng dụng có sẵn trên các thiết bị vì có khả năng rò rỉ thông tin của người dùng rất cao. Sử dụng một bảng điện tử trong nhà thì sẽ mất đi khả năng linh hoạt để điều khiển thiết bị

- ➔ Cho nên lựa chọn thiết kế một trang web để có tính bảo mật cho người dùng, còn có thể giao tiếp linh hoạt ở bất cứ đâu khi có điện thoại hoặc một cái máy tính.



Hình 2.1.2: Thiết kế trang web

Web sẽ bao gồm 3 phần chính:

- + Front-end là phần giao tiếp được với người dùng.
- + Back- end (server): là phần điều hướng, xử lý dữ liệu.
- + Database: là nơi lưu trữ dữ liệu của ngôi nhà

2.3. Thiết kế tổng quan của hệ thống

Để giao tiếp được giữa các node (các thiết bị trong nhà) và web thì phải thông qua một giao thức khác để hai bên hiểu nhau và giao tiếp với nhau.

Data protocols - Giao thức truyền dữ liệu IoT (Lớp ứng dụng): Hiểu nôm na là hình thức đóng gói dữ liệu và truyền gói dữ liệu đó qua lại giữa các thiết bị trong mạng lưới với nhau.

Một số Data protocols phổ biến ứng dụng trong IoT hiện nay:

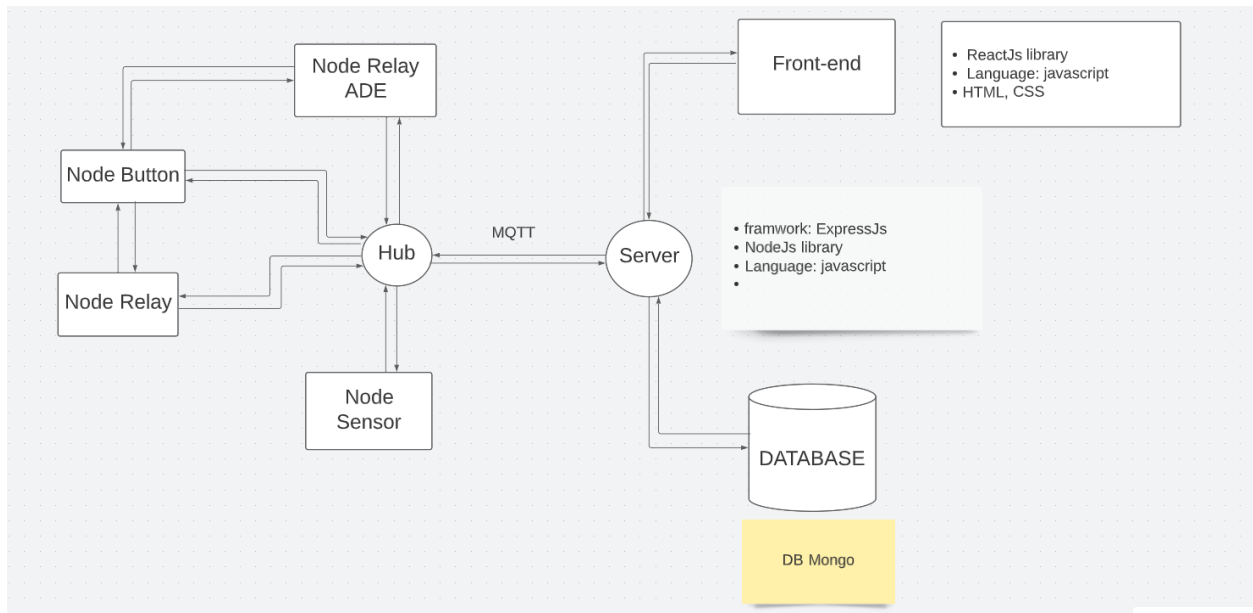
MQTT (Message Queue Telemetry Transport): MQTT là một giao thức theo cơ chế xuất bản/đăng ký, ở đó máy client có thể xuất bản hay nhận bản tin. Nó giúp giao tiếp dễ dàng giữa nhiều thiết bị. Nó là một giao thức nhắn tin đơn giản được thiết kế cho các thiết bị bị hạn chế và có băng thông thấp, vì vậy nó là một giải pháp hoàn hảo cho các ứng dụng internet vạn vật.

HTTP (HyperText Transfer Protocol): cho phép trao đổi thông tin (chủ yếu ở dạng siêu văn bản) qua Internet. HTTP hiện là nền tảng truyền dữ liệu của ứng dụng duyệt web ngày nay và được sử dụng rộng rãi trong hệ thống Internet of Things. Mặc dù giao thức http có nhiều nhược điểm trong việc truyền dữ liệu và không phù hợp bằng các giao thức tối ưu khác như MQTT, CoAP, AMQP áp dụng cho IoT, nhưng giao thức này vẫn phổ biến trong lĩnh vực nhà thông minh cũng như việc sử dụng nhiều trong bộ vi điều khiển và vi xử lý tiên tiến.

CoAP (Constrained Applications Protocol): CoAP là một giao thức đơn giản chi phí thấp được thiết kế riêng cho các thiết bị hiệu năng thấp (chẳng hạn như vi điều khiển) và nơi mạng có băng thông thấp. Giao thức này được sử dụng để trao đổi dữ liệu M2M và rất giống với HTTP. Do đó, CoAP có thể được coi là sự lựa chọn giao thức tốt nhất cho mạng truyền thông gia đình.

AMQP (Advanced Message Queue Protocol) là một giao thức làm trung gian cho các gói tin trên lớp ứng dụng với mục đích thay thế các hệ thống truyền tin độc quyền và không tương thích. AMQP là một giao thức có dây (wire-protocol), có khả năng diễn tả các message phù hợp với định dạng dữ liệu, có thể triển khai với rất nhiều loại ngôn ngữ lập trình.

→ Dựa trên điểm dễ dàng trao đổi dữ liệu, và hoạt động được trên các thiết bị có băng thông thấp nên lựa chọn MQTT.



Hình 2.1.3: Tổng quan hệ thống thiết kế.

Như thiết kế ở trên ta có thể tách hệ thống thành 2 phần chính: Là các node trong nhà và phần giao tiếp với người dùng (website). Và kết hợp hai phần đó với nhau thông qua giao thức MQTT để giao tiếp giữa hai thành phần với nhau.

Chương 3. Thiết kế các thiết bị trong nhà

3.1. Giao tiếp Zigbee

3.1.1. Cơ sở lý thuyết

Zigbee là một giao thức truyền thông được phát triển dựa trên chuẩn truyền thông không dây IEEE 802.15.4. Giao thức này được tạo ra nhằm phục vụ cho những ứng dụng yêu cầu giá thành và công suất thấp nhưng phải có khả năng linh động trong phạm vi rộng. Chuẩn Zigbee được phát triển và xúc tiến bởi hãng Zigbee Alliance với sự hỗ trợ hơn 200 công ty trên thế giới.

- **Tính năng của Zigbee:**

- Tính ổn định: Mạng Zigbee hình lưới có đặc điểm tự thích nghi, tức là chúng có khả năng tự xây dựng lại và hoạt động bình thường ngay cả khi có một vài nút bị hỏng, hoặc tìm đường đi khác khi đường đi thông thường bị chặn.
- Tính bảo mật: Chuẩn Zigbee hỗ trợ bảo mật trên nhiều tầng, gồm có tầng xác thực cơ bản, mã hóa AES 128-bits, bảo mật trong cơ cấu hình thành và sát nhập nút.
- Khả năng mở rộng: Cơ chế định địa chỉ 64-bits có thể mở rộng đến hơn 65000 nút mạng, có khả năng bao quát toàn bộ nhà máy.
- Giá thành rẻ: Bao gồm chi phí mua thiết bị, chi phí lắp đặt và chi phí bảo trì. Các thiết bị Zigbee có thể hoạt động bằng pin chính trong vài năm mà không cần bộ sạc, đồng thời các nút có thể hoạt động ở chế độ nghỉ (sleep mode) giúp tiết kiệm đáng kể năng lượng.
- Khả năng hỗ trợ: Chuẩn mở với nhiều nhà cung cấp, hỗ trợ nhiều ứng dụng và ngày càng được cải tiến, phát triển rộng rãi.

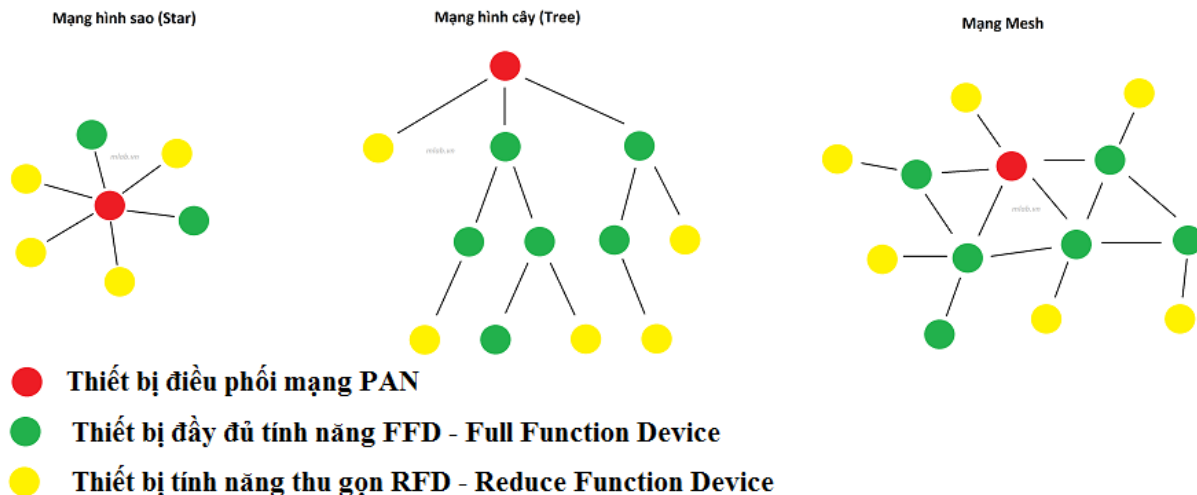
- **Giải tần của Zigbee:**

- Khu vực Bắc Mỹ: 915MHz
- Khu vực Châu Âu, Nhật: 868MHz
- Khu vực khác: 2.4GHz

- **Thành phần trong mạng Zigbee:**

- Zigbee Coordinator (ZC): Đây là thiết bị gốc có nhiệm vụ quyết định cấu hình mạng, quy định cách đánh địa chỉ và lưu trữ bảng địa chỉ. Mỗi mạng chỉ có duy nhất một ZC và nó cũng là thiết bị duy nhất nói chuyện được với các mạng khác.
- Zigbee Router (ZR): Thiết bị này có nhiệm vụ định tuyến trung gian trong việc truyền dữ liệu, nó sẽ tự phát hiện và lập bản đồ các nút xung quanh cũng như là theo dõi các điều khiển hoạt động bình thường.

- Zigbee End Device (ZED): Gọi là thiết bị điểm cuối, nó sẽ giao tiếp với ZC, ZR gần nó nhất. Chúng có nhiệm vụ đọc thông tin từ các thành phần vật lý, chúng thường ở trạng thái nghỉ và làm việc khi chuyển nhận thông điệp nào đó.
- **Mô hình mạng Zigbee:**
 - Mạng hình sao (Star network): Gồm 1 nút trung tâm ZC, tất cả các nút khác đều được kết nối với nút trung tâm này, mạng hình sao bị hạn chế khoảng cách và sự mở rộng.
 - Mạng hình cây (Cluster Tree network): Gồm 1 nút ZC, các nút khác được liên kết với nhau theo mô hình giống một cái rễ cây, dạng này có khả năng mở rộng cao, tăng khoảng cách và quy mô hệ thống.
 - Mạng hình lưới (Mesh network): Gồm 1 nút ZC, các thiết bị trong mạng đều có thể kết nối với nhau (trừ ZED). Khi một đường truyền bị lỗi, sẽ tự động tìm đường truyền khác, tăng tính tin cậy và kết nối trong mạng.

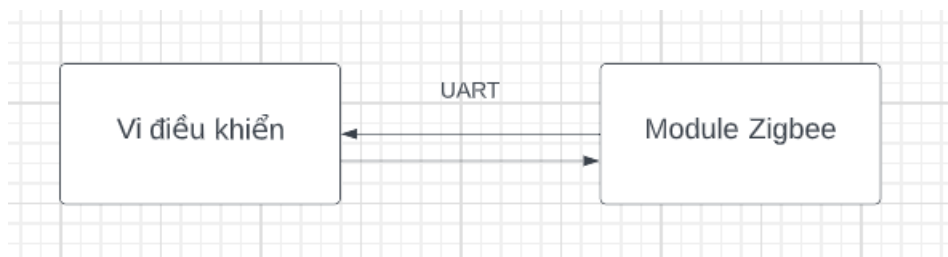


Hình 1 Các dạng mô hình mạng trong Zigbee

- **Ứng dụng của Zigbee:** Hướng ưu tiên của Zigbee là mạng cảm biến không dây, ứng dụng giám sát và điều khiển không dây.

3.1.2. Thiết kế giao tiếp giữa các thiết bị

3.1.2.1. Frame truyền gói tin giao tiếp với module Zigbee.



Hình 3.1.2.1: Giao tiếp giữa Vi điều khiển và Module Zigbee

Để giao tiếp được giữa Vi điều khiển và Module Zigbee thì ta sẽ sử dụng giao tiếp uart để truyền và nhận dữ liệu giữa các thiết bị.

SOF	MT CMD	FCS
Byte: 1	3-256	1

Hình 3.1.2.2: Cấu trúc Frame giao tiếp với module Zigbee

Cấu trúc một Frame truyền gói tin giao tiếp với module Zigbee sẽ gồm 3 thành phần chính:

- **SOF** (Start of Frame): đây là 1-byte có giá trị 0xFE dùng để xác định việc bắt đầu của mỗi frame truyền.
- **MT CMD** (Monitor Test Command): là thành phần chứa dữ liệu và lệnh để điều khiển với module Zigbee.
- **FCS** (Frame Check Sequence): đây là 1-byte dùng để đảm bảo tính toàn vẹn của gói tin. Giá trị của byte này sẽ được tính bằng cách XOR tất cả các byte của **MT CMD**. Thiết bị khi nhận sẽ dùng byte này so sánh với XOR của các byte trong **MT CMD** để quyết định lấy dữ liệu này xử lý hay không.

Cụ thể nội dung của **MT CMD** sẽ bao gồm 3 thành phần chính:

LEN	CMD	DATA
Byte: 1	2	0-250

Hình 3.1.2.3: Miền MT DATA trong frame truyền

- **LEN** (Length): đây là 1-byte có giá trị biểu diễn độ dài byte của miền **DATA**
- **CMD** (Command): đây là miền 2-byte dùng để xác định loại gói tin khi giao tiếp giữa vi điều khiển và module Zigbee. Các lệnh điều khiển trong gói tin có nhiều loại ví dụ như lệnh cấu hình module, lệnh gửi gói tin, ...
- **DATA**: đây là miền chứa thành phần dữ liệu mà mình muốn gửi kèm với lệnh ở miền **CMD**. Đối với từng gói tin thì miền này sẽ có thể có các độ dài khác nhau từ 0~250-byte.

Ví dụ của Frame truyền gói tin với module Zigbee:

Lệnh **SYS_PING** khi gửi sẽ có dạng **0xFE 0x00 0x21 0x01 0x20**

SOF	LEN	CMD0	CMD1	DATA	FCS
Byte: 1	1	1	1	0	1
0xFE	0x00	0x21	0x01	N/A	0x20

Hình 3.1.2.4: Frame truyền lệnh **SYS_PING**

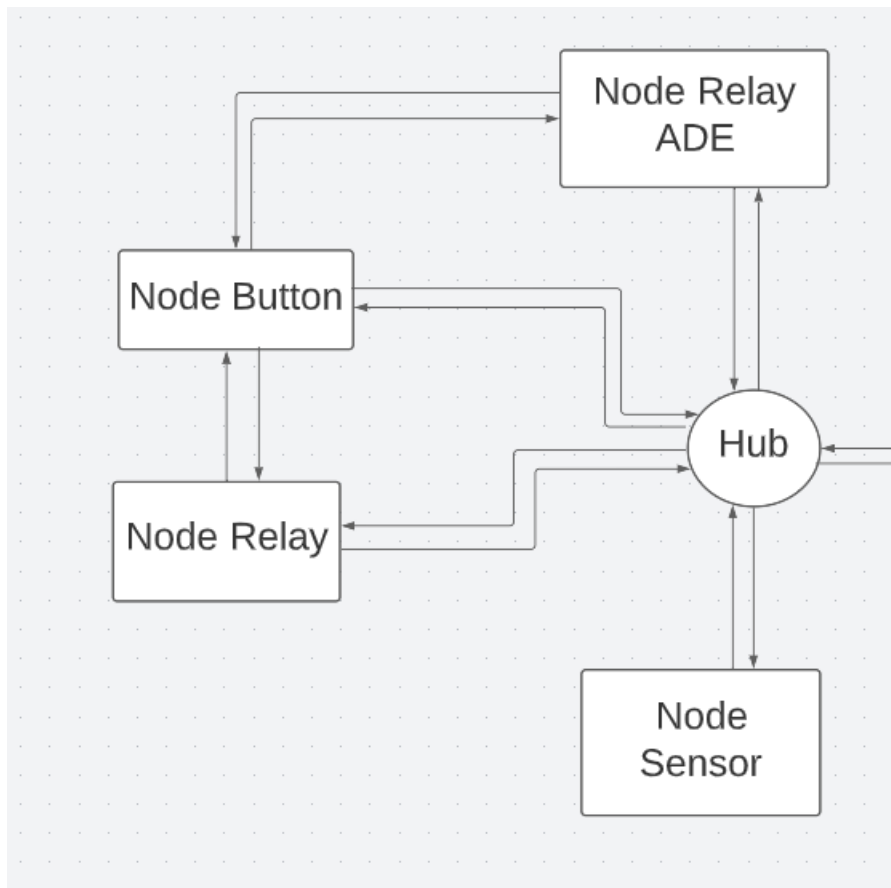
Và phản hồi của lệnh **SYS_PING** từ module sẽ có dạng **0xFE 0x02 0x61 0x01 0x11 0x00 0x73** hoặc tương tự như vậy

SOF	LEN	CMD0	CMD1	DATA0	DATA1	FCS
Byte: 1	1	1	1	1	1	1
0xFE	0x02	0x61	0x01	0x11	0x00	0x73

Hình 3.1.2.5: Frame phản hồi từ lệnh **SYS_PING**

Module Zigbee khi nhận gói tin từ các thiết bị Zigbee khác trong mạng sẽ gửi lại gói tin về cho vi điều khiển thông qua giao tiếp UART với cấu trúc Frame truyền đã nêu trên. Khi nhận gói tin đó, vi điều khiển sẽ kiểm tra cấu trúc Frame để xác thực gói tin và trích xuất dữ liệu phần miền **DATA** ra để xử lý.

3.1.2.2. Frame truyền dữ liệu giao tiếp giữa các Hub và Node



Hình 3.1.2.6: Giao tiếp giữa các Hub và Node

Destination Address	Source Address	Endpoint	Cluster ID	Data
2 bytes	2 bytes	1 byte	2 bytes	N bytes

Hình 3.1.2.6: Frame truyền dữ liệu giữa các Hub và Node

Frame truyền dữ liệu giữa các thiết bị với nhau phía trên là phần dữ liệu trong miền **DATA** của **MT CMD** với lệnh **CMD** tương ứng là truyền nhận dữ liệu. Trong đó sẽ bao gồm các thành phần chính sau:

- **Destination Address:** Chứa địa chỉ đích của gói tin.
- **Source Address:** Chứa địa chỉ nguồn của gói tin.
- **Endpoint:** Là thành phần phân biệt loại gói tin muốn gửi giữa các thiết bị. Với loại gói tin khác nhau thì sẽ có giá trị endpoint riêng.

- **Cluster ID:** Phân biệt loại chức năng thiết bị. Các loại thiết bị có chức năng khác nhau khi gửi gói tin đến HUB sẽ có các giá trị Cluster ID riêng.
- **Data:** Phần dữ liệu được truyền đi.

3.1.2.3. Phân loại gói tin giao tiếp.

Gói tin truyền dữ liệu giao tiếp giữa Hub và Node sẽ được chia thành 3 loại với các giá trị Endpoint khác nhau:

- **Provision:** Là loại gói tin của Node dùng để cung cấp dữ liệu thông tin về chức năng cũng như loại thiết bị để gửi đến cho Hub. Gói tin này sẽ được gửi khi thiết bị tham gia vào mạng Zigbee thành công. Giá trị **Endpoint:** 0x01.
- **Telemetry:** Là loại gói tin của Node dùng để gửi dữ liệu thông số và trạng thái thiết bị đến Hub nhằm mục đích giám sát. Gói tin này sẽ được gửi khi có dữ liệu cập nhật mới. Giá trị **Endpoint:** 0x02.
- **Command:** Là loại gói tin của Hub dùng để gửi các lệnh, yêu cầu điều khiển đến thiết bị Node. Gói tin này được gửi đi khi người dùng thực hiện thao tác điều khiển thiết bị thông qua giao diện. Giá trị **Endpoint:** 0x03.

3.2. Thiết kế các thiết bị

3.2.1. Thiết kế các khối dùng chung

a. Khối giao tiếp Zigbee

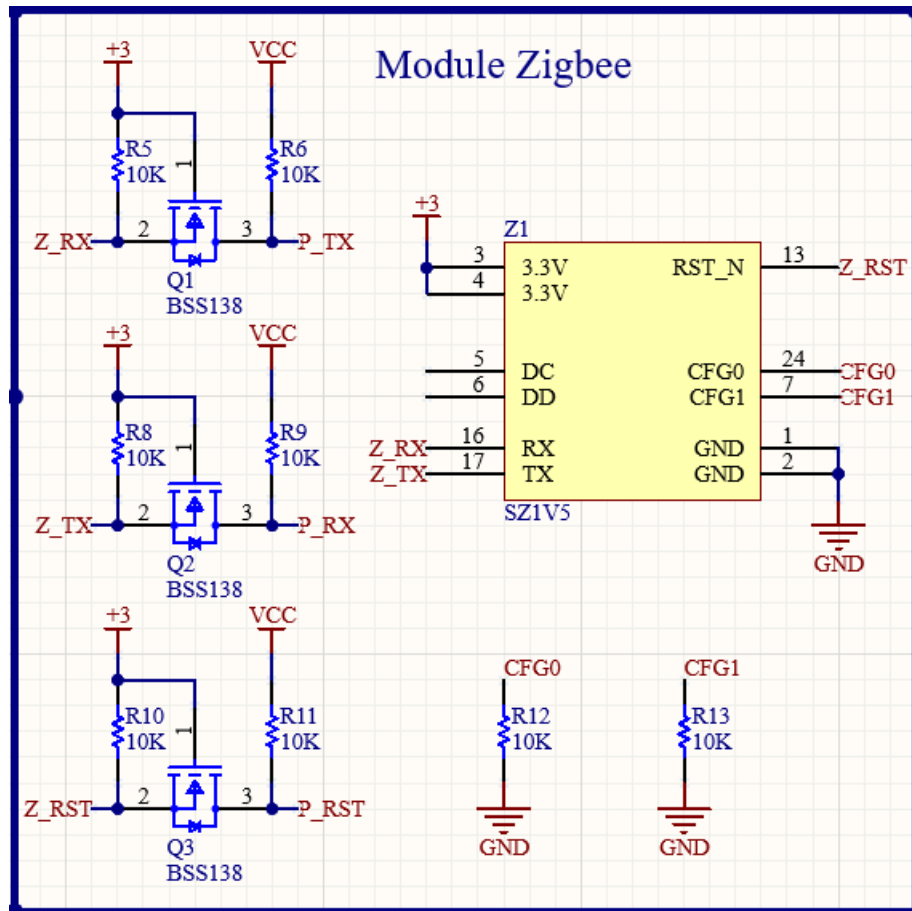
Module ZigBee sz1v5 CC2530 2.4G:

- 2.4Ghz IEEE802.15.4 tương thích thu phát RF
- Tốc độ truyền dữ liệu 250kbps
- Bộ nhớ Flash 256KB
- RAM 8KB
- Hỗ trợ giao tiếp UART
- Cấp nguồn từ 2V đến 3.6V



Module Zigbee sz1v5

Module Zigbee được sử dụng để giao tiếp giữa bộ Hub với các Node

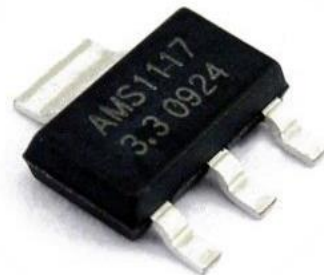


Mạch nguyên lý module Zigbee với các VĐK 5V

b. Khởi nguồn

IC AMS1117 - 3.3V là IC ổn áp có độ ổn định cao, Là IC chuyên dụng để ổn định điện áp tuyến tính với sụt áp thấp. IC có các đặc điểm sau:

- Điện áp ngõ ra: 3.3V
- Dòng ra tối đa: 800mA
- Áp ngõ vào: $1.5 < V_{in} - V_{out} < 12V$ (4.8-8.7VDC)
- Nhiệt độ hoạt động: âm 40 đến 125 độ C

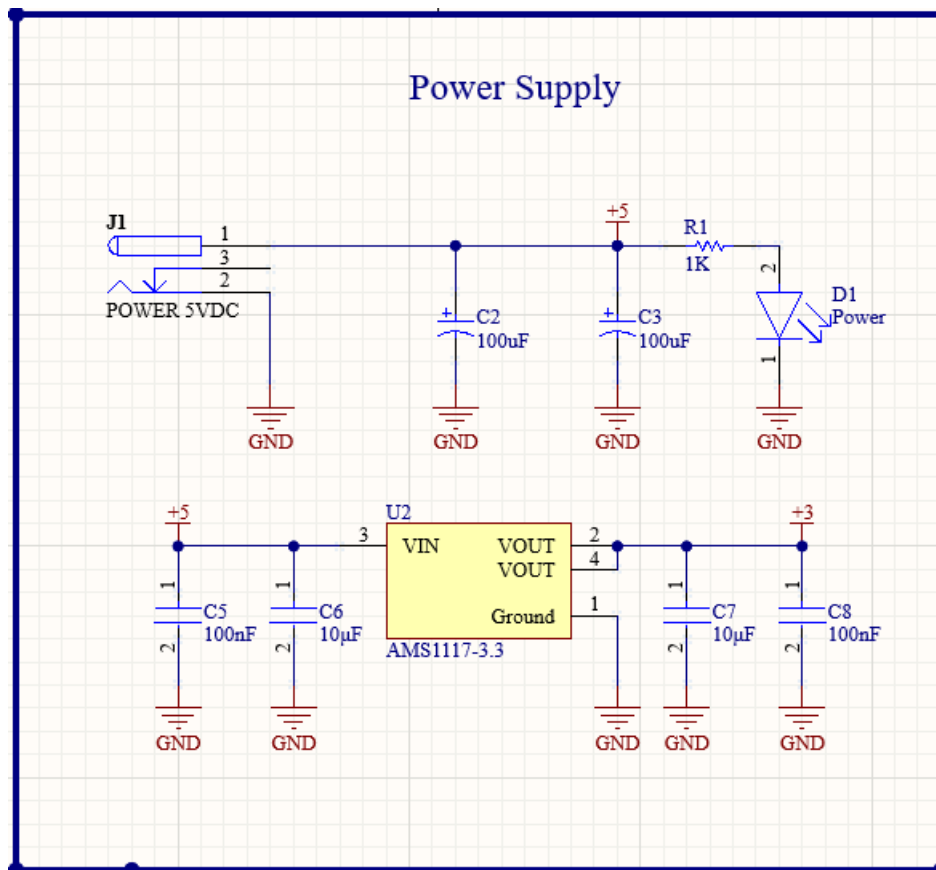


IC AMS1117-3.3V

Sử dụng nguồn Adapter biến đổi điện áp xoay chiều 220V thành 5VDC, sau đó áp 5VDC sẽ được đi vào mạch điện có dùng IC ASM1117 để hạ điện áp xuống còn 3.3VDC.



Adapter 220VAC/5VDC

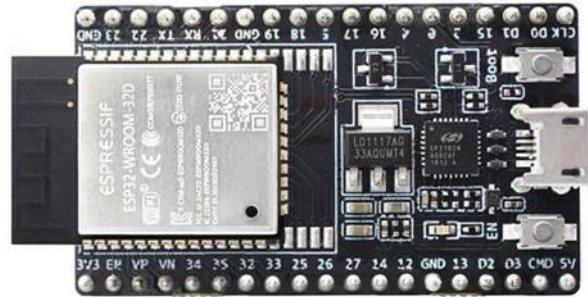


Mạch nguyên lý khối nguồn

3.2.2. Thiết kế bộ điều khiển trung tâm Hub.

3.2.3.1. Sơ đồ khối chức năng

ESP32-DevKitC V4 (ESP32) là một mạch phát triển dựa trên ESP32 có kích thước nhỏ được sản xuất bởi Espressif. Hầu hết các chân được chia ra thành các đầu ghim ở cả hai bên để dễ dàng giao tiếp. Một số đặc điểm của ESP32-DevKitC V4 như sau:



ESP32-DevkitC-v4

- Bộ nhớ có 520 KByte SRAM trên chip cho dữ liệu và lệnh.
- Hầu hết các chân trên module ESP đều được đưa ra các đầu ghim trên bo mạch để thuận tiện lập trình ESP32 để kích hoạt nhiều chức năng như PWM, ADC, DAC, I2C, I2S, SPI, v.v.
- Có cổng chuyển đổi USB sang UART nhằm giao tiếp và nạp code dễ dàng.
- Có Webserver riêng, có thể làm trạm phát Wifi hoặc bắt Wifi.
- Cấp nguồn thông qua USB hoặc nguồn ngoài 5V / 3.3V.

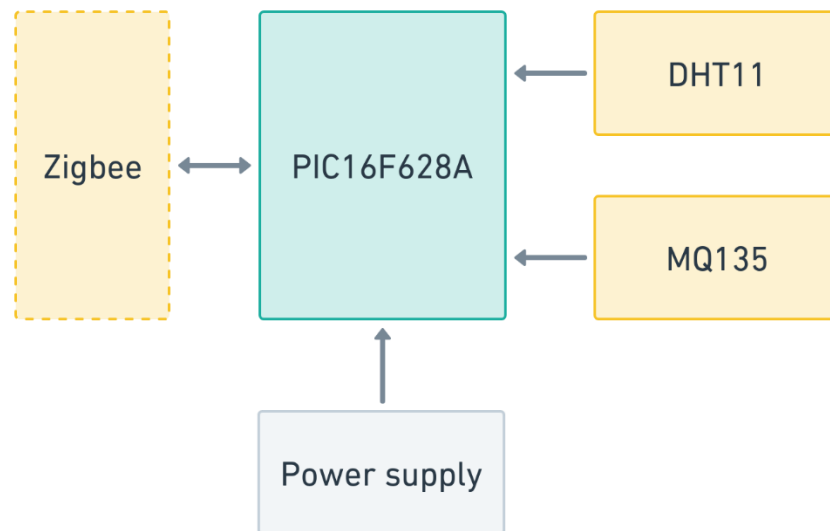
ESP32 được sử dụng để:

- Kết nối WiFi
- Gửi thông số thiết bị đo được thông qua WiFi để đến giao diện người dùng.
- Làm vi điều khiển chính thực hiện các chức năng sau:
 - + Giao tiếp I2C với module DS1307 để cài đặt và lấy thông số về thời gian thực.
 - + Giao tiếp UART với Module Zigbee để điều khiển và nhận dữ liệu các node
 - + Giao tiếp SPI với màn hình LCD5110 để hiển thị các thông số lên màn hình

3.2.3. Thiết kế node sensor

3.2.3.1. Sơ đồ khối chức năng

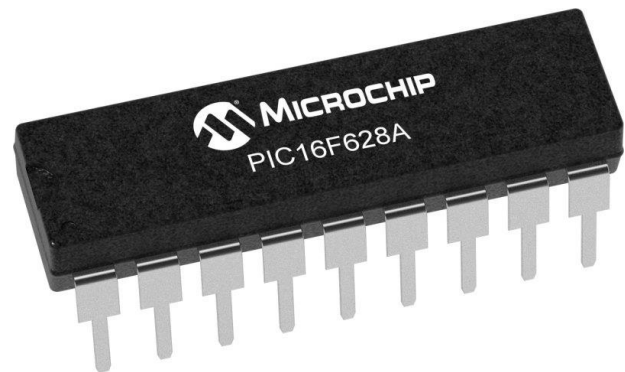
Node Sensor là thiết bị kết nối Zigbee có chức năng đọc cảm biến về thông số môi trường như nhiệt độ, độ ẩm và chất lượng không khí. Được sử dụng để giám sát điều kiện môi trường cũng như cảnh báo các chất khí dễ cháy như khí gas.



Sơ đồ khối chức năng Node Sensor

a. PIC16F628A

PIC16F628A là một chip vi điều khiển được sản xuất bởi hãng Microchip thuộc họ Pic, là một bộ vi điều khiển 8-bit dựa trên kiến trúc RISC với các thông số kỹ thuật sau:



Vi điều khiển PIC16F628A

Bộ nhớ Flash	2KB
Bộ nhớ EEPROM	128B
Bộ nhớ SRAM	224B
Internal Oscillator	4MHz \pm 1%

IO	16
UART	1
Analog comparator	1
Capture, Compare, PWM (CCP)	1
Timer 8-bits/16-bits	2/1

b. MQ-135

Cảm biến chất lượng không khí MQ-135 sử dụng để kiểm tra chất lượng không khí trong môi trường, thường được sử dụng trong các thiết bị kiểm tra chất lượng không khí bên trong cao ốc, văn phòng, thích hợp để phát hiện NH₃, NO_x, Alcol, Benzen, khói, CO₂, ...

Thông số kỹ thuật:

- Khoảng phát hiện:
 - 10 ~ 300 ppm NH₃
 - 10 ~ 1000 ppm Benzene
 - 10 ~ 300 Alcol



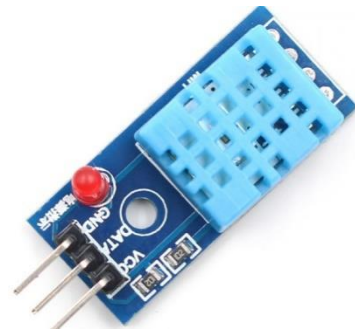
Cảm biến chất lượng không khí MQ135

c. DHT11

Cảm biến nhiệt độ, độ ẩm DHT11 sử dụng để thu thập dữ liệu nhiệt độ và độ ẩm trong môi trường

Thông số kỹ thuật:

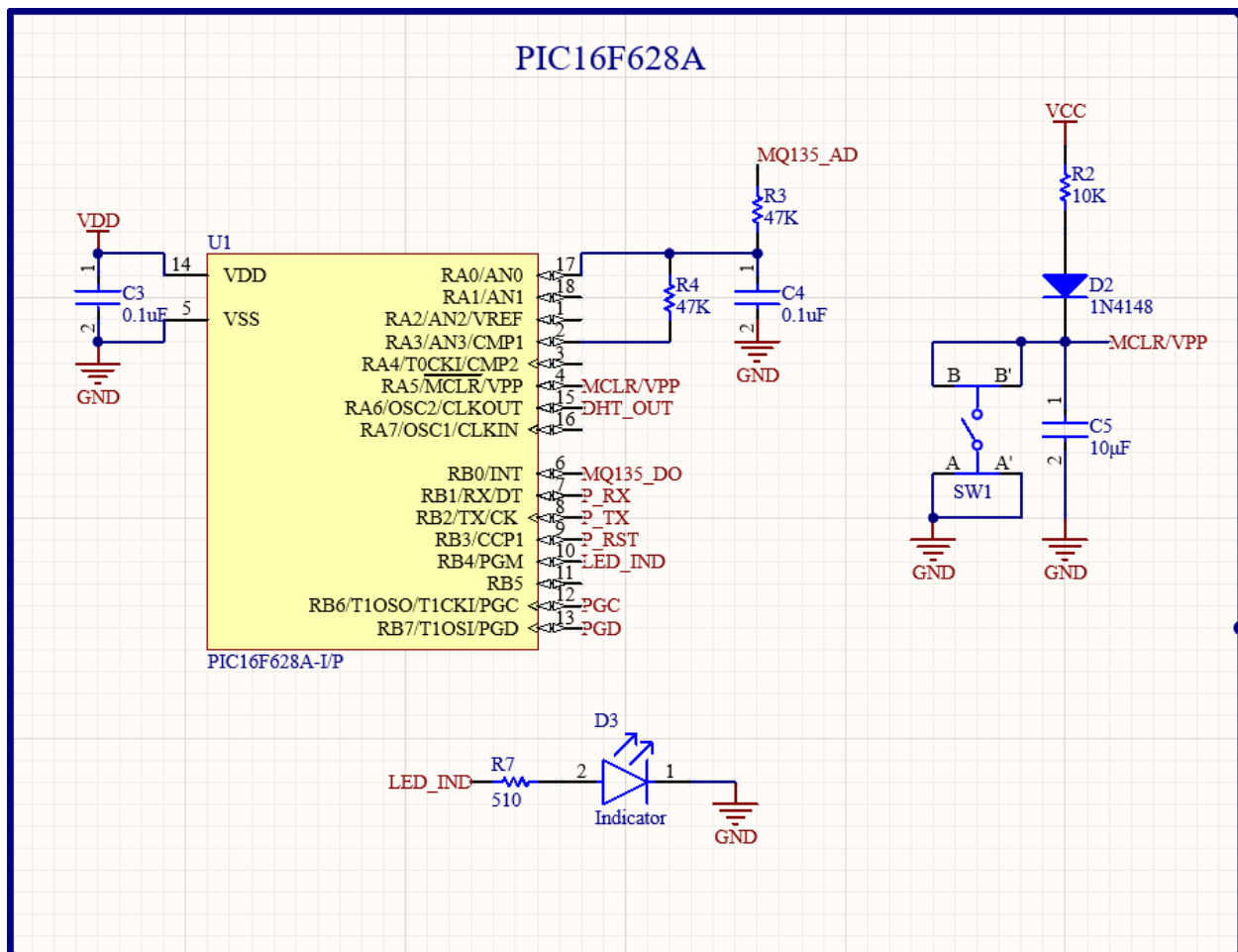
- Khoảng đo nhiệt độ: 0 ~ 50°C ± 2°C
- Khoảng đo độ ẩm: 20% ~ 80% RH ± 5% RH
- Tần số lấy mẫu tối đa: 1Hz

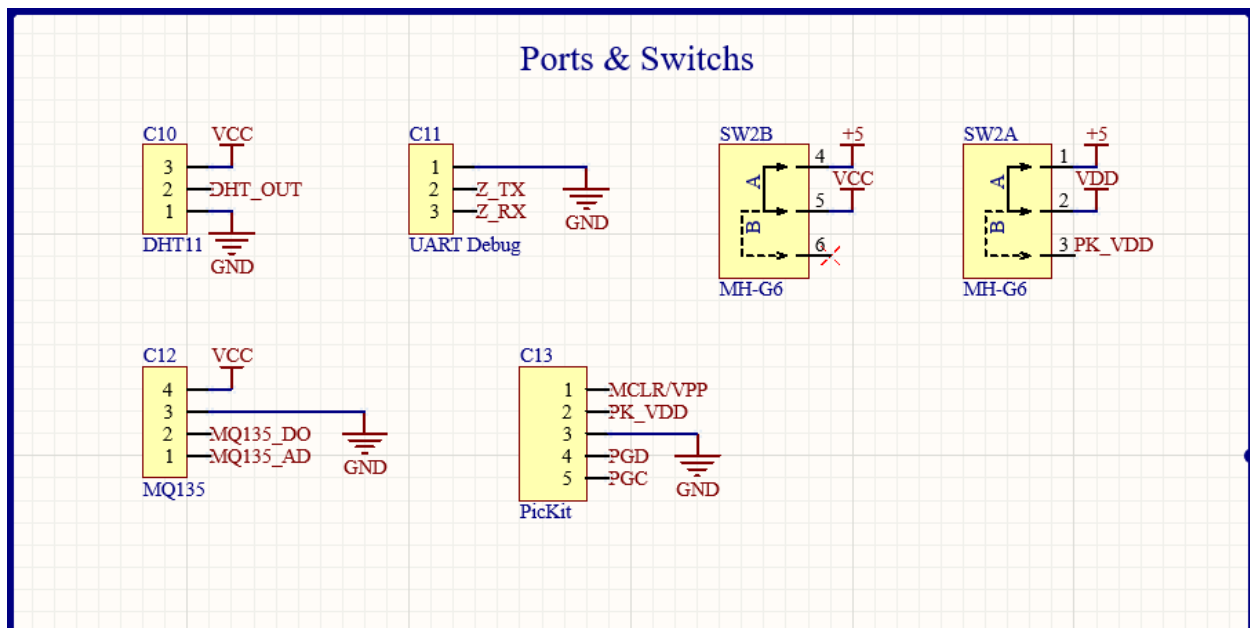


Cảm biến nhiệt độ, độ ẩm DHT11

3.2.3.2. Thiết kế mạch

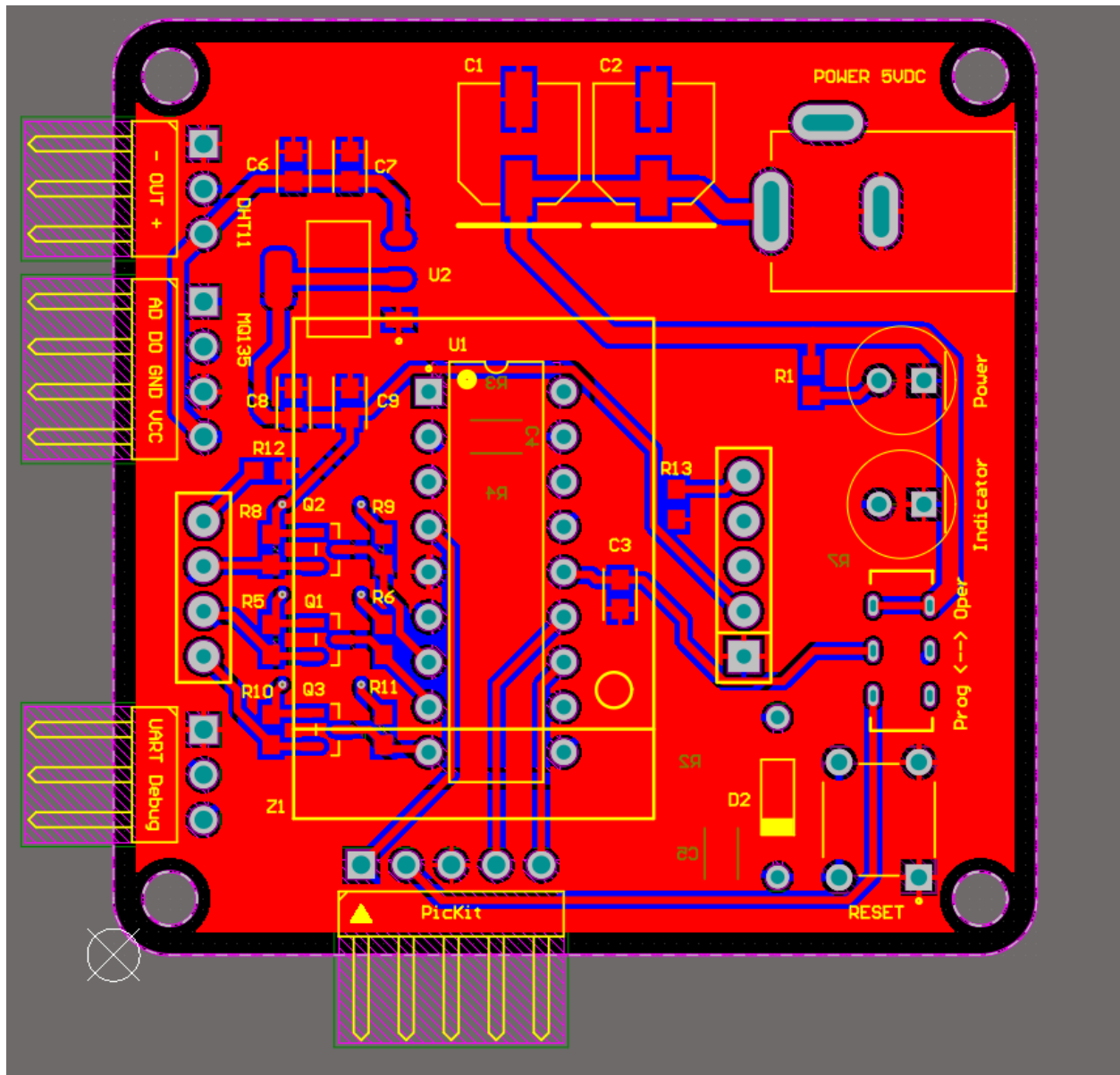
a. Mạch nguyên lý

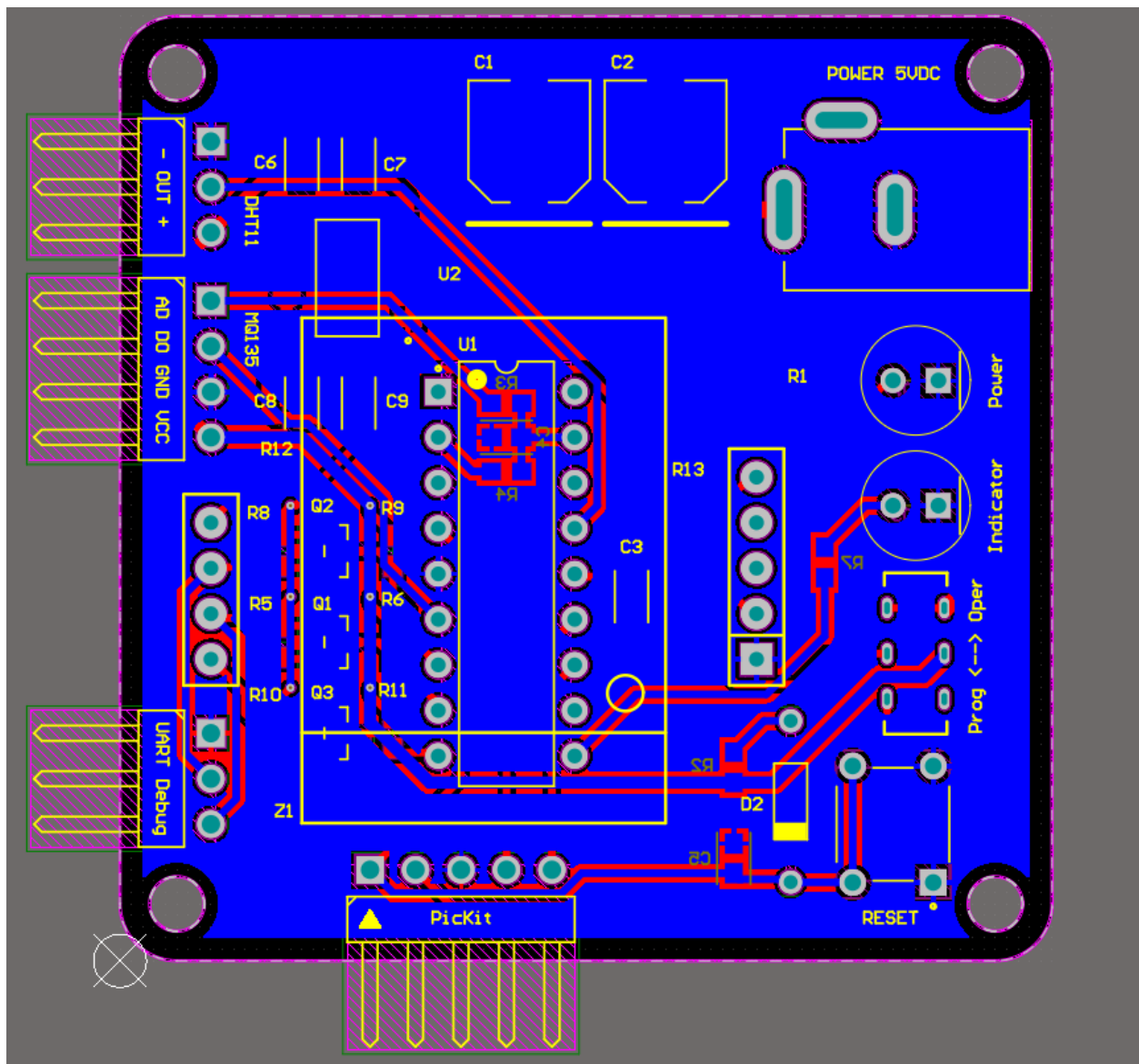




Sơ đồ nguyên lý Node Sensor

b. PCB layout

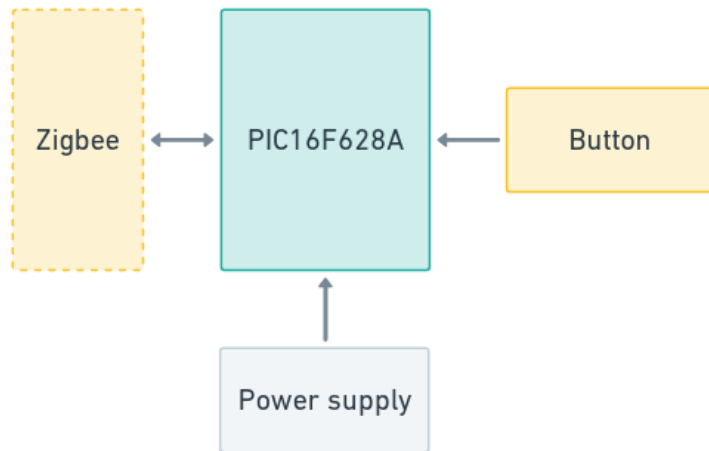




PCB Node Sensor

3.2.4. Thiết kế node button

3.2.4.1. Sơ đồ khối chức năng



Sơ đồ khối chức năng Node Button

Zigbee Node Button là thiết bị kết nối Zigbee có các nút nhấn để người dùng có thể sử dụng điều khiển thiết bị hoặc chức năng nhất định nào đó.

Nút nhấn dùng trong các mạch điện tử thông dụng, có thể dùng để làm nút reset hoặc nút nguồn....

Thông số kỹ thuật của nút nhấn sử dụng

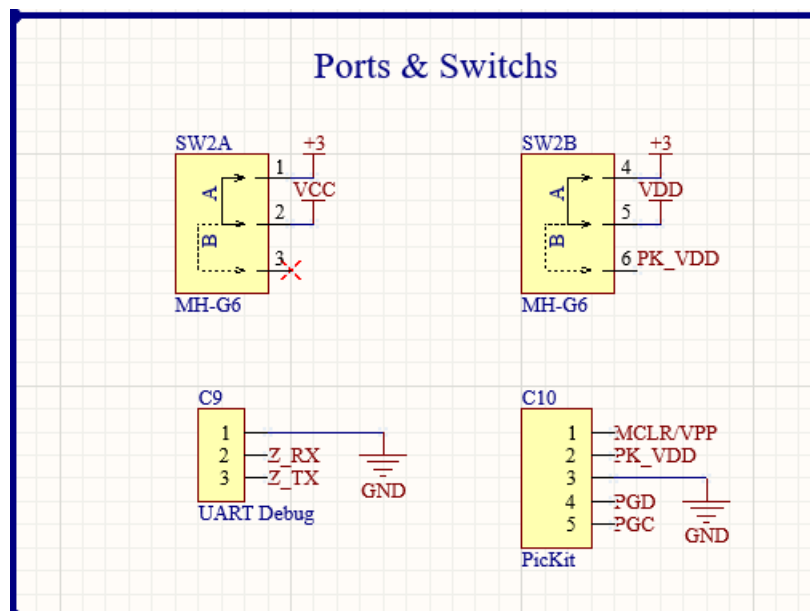
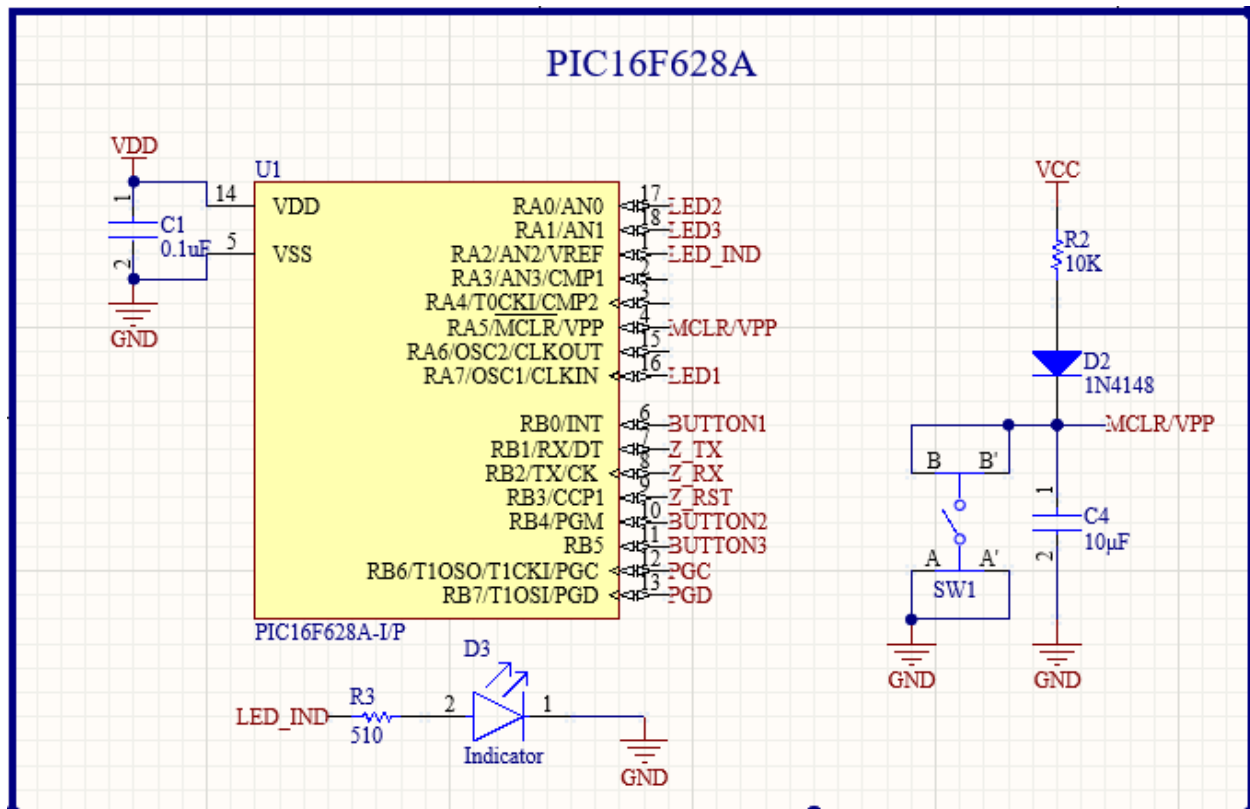
- Kích thước: 6x6x8mm
- Loại chân: cắm 4 chân.



Nút nhấn 6x6x8mm

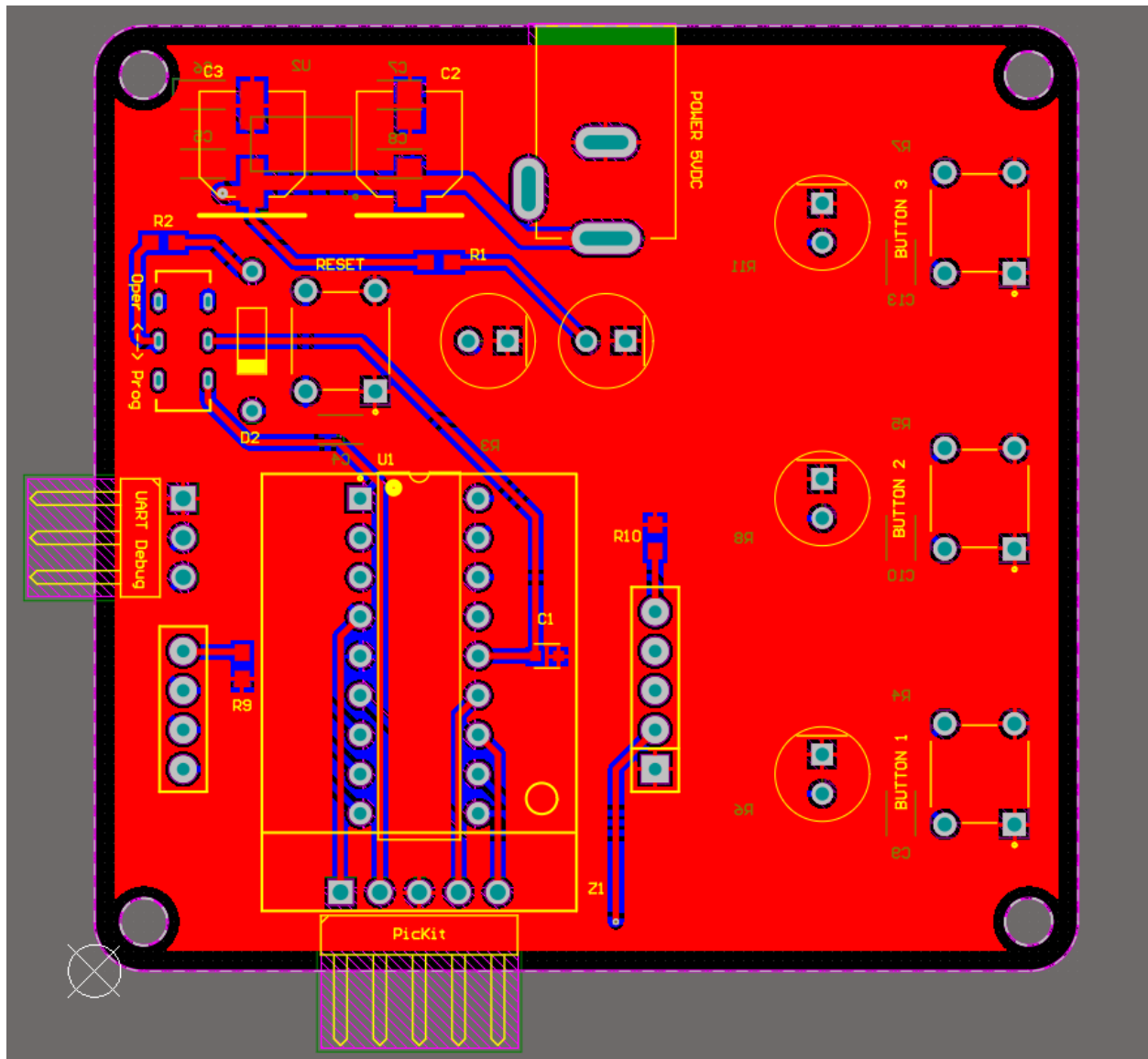
3.2.4.2. Thiết kế mạch

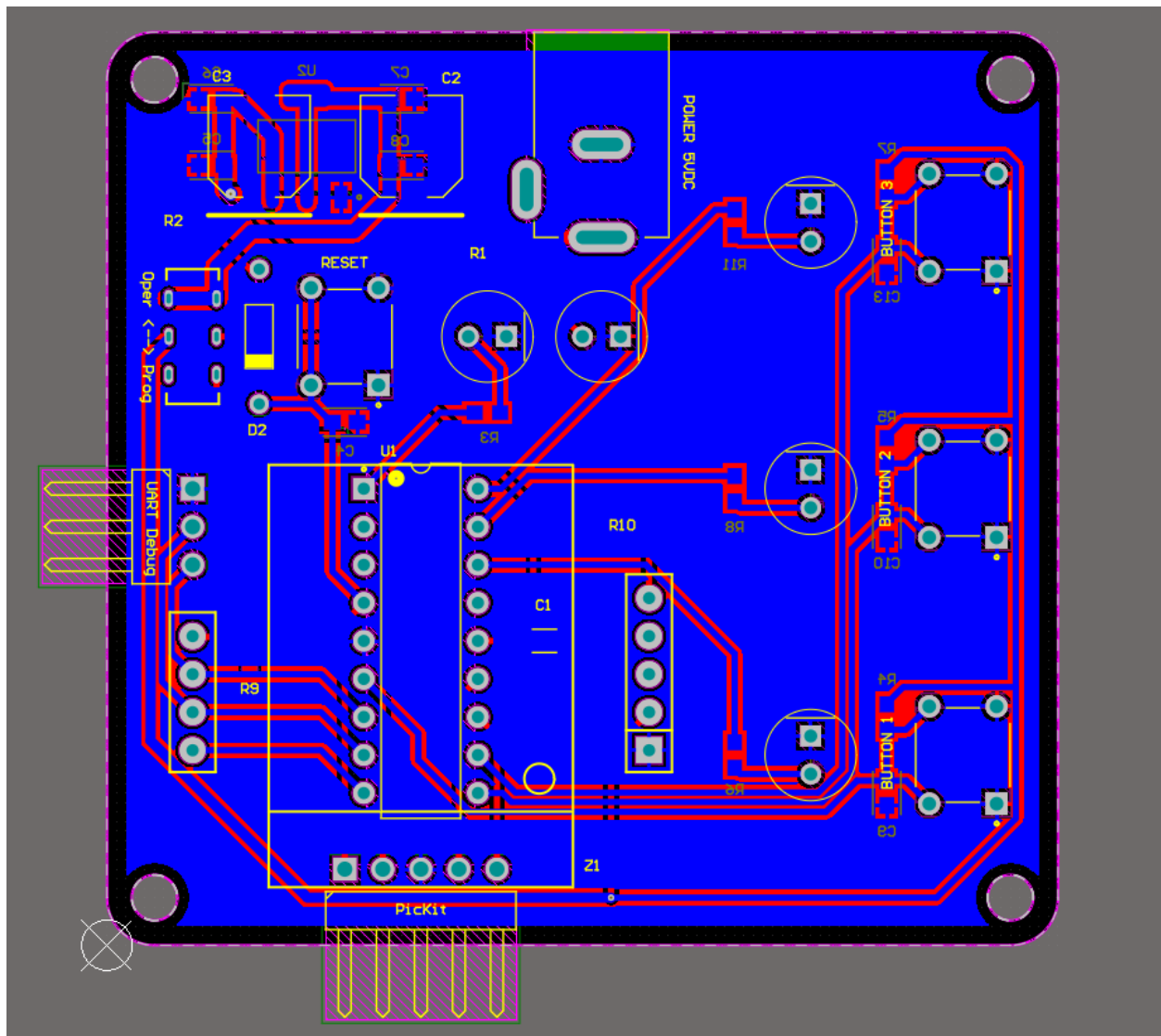
a. Mạch nguyên lý



Sơ đồ nguyên lý Node Button

b. PCB layout





PCB Node Button

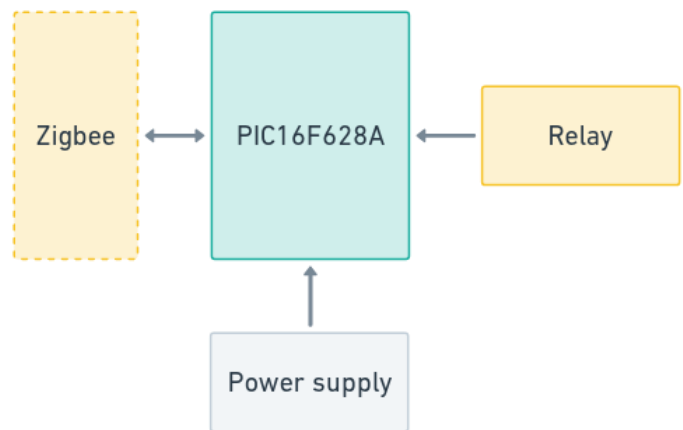
3.2.5. Thiết kế node relay

3.2.5.1. Sơ đồ khối chức năng

Zigbee Node Relay là thiết bị kết nối Zigbee có nhiều kênh Relay có thể dùng để điều khiển đóng ngắt. Có thể sử dụng lắp đặt cho các thiết bị như đèn, quạt thông gió, ...



Relay SRD-05VDC-SL-C



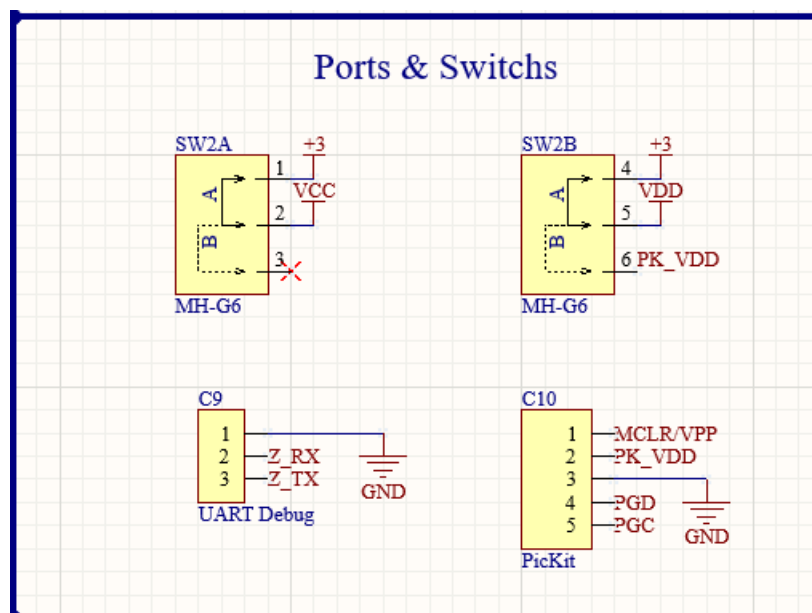
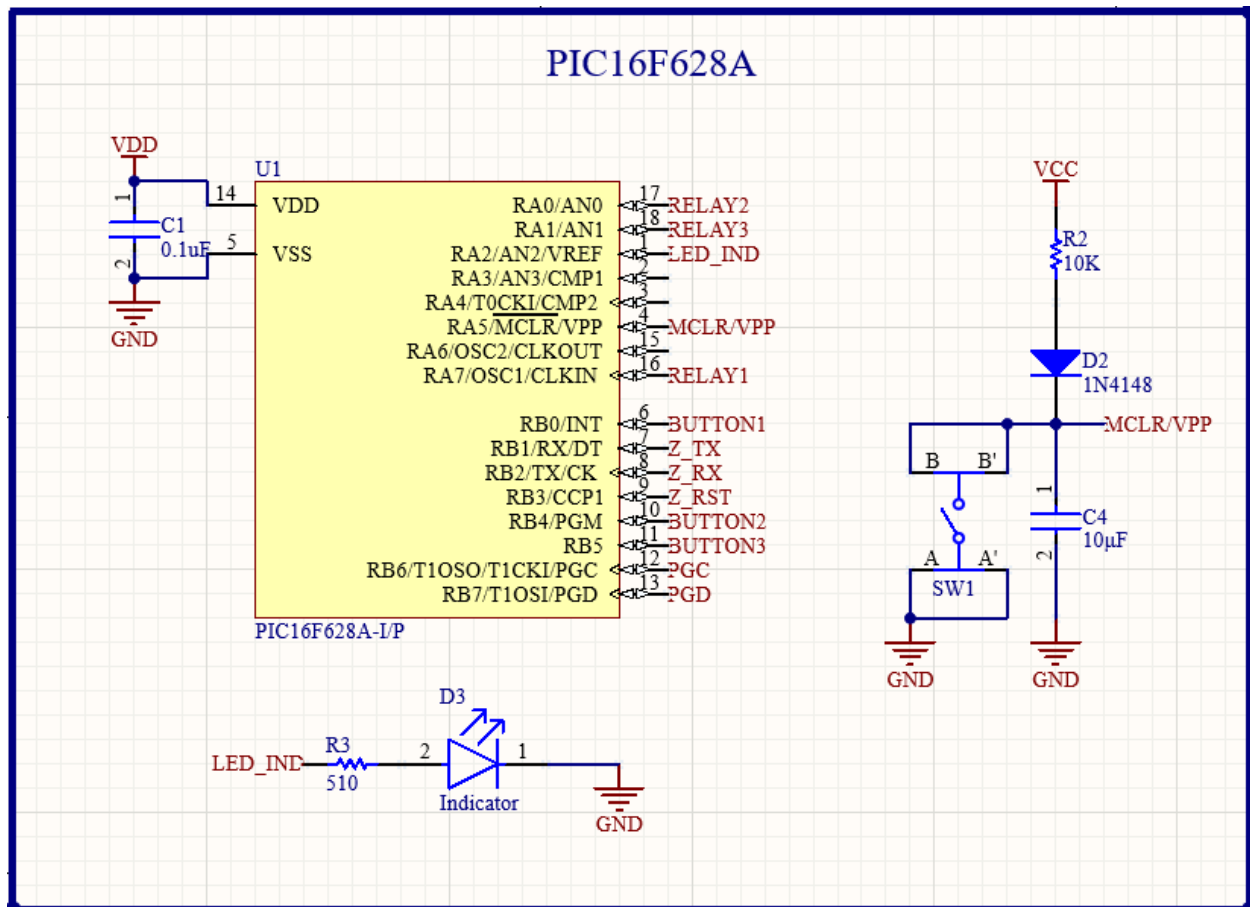
Sơ đồ khối chức năng Node Relay

Thông số:

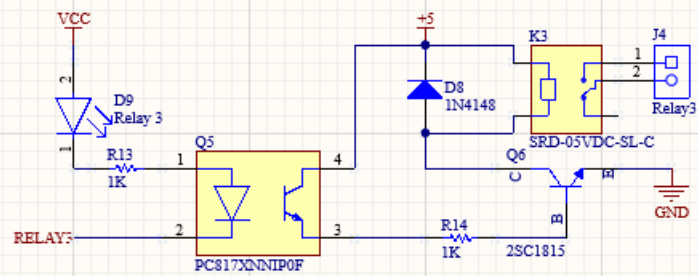
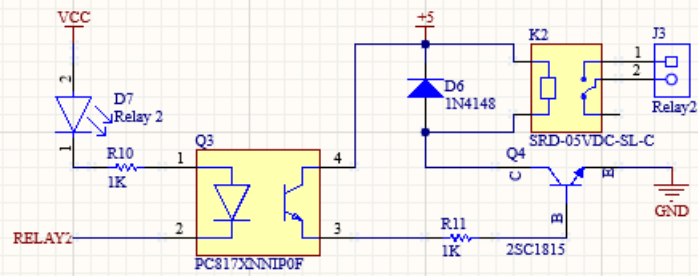
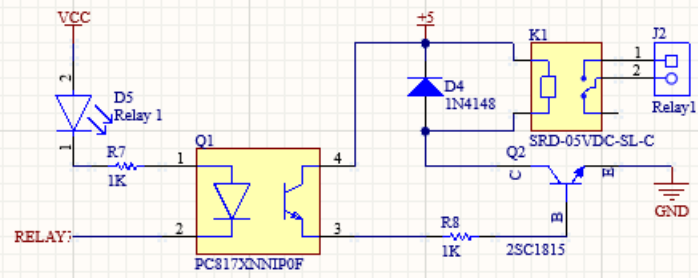
- Điện áp định mức: 5VDC
- Dòng tải 10A
- Cuộn kháng: $400\Omega \pm 10\%$
- Dòng điện hoạt động: 40mA

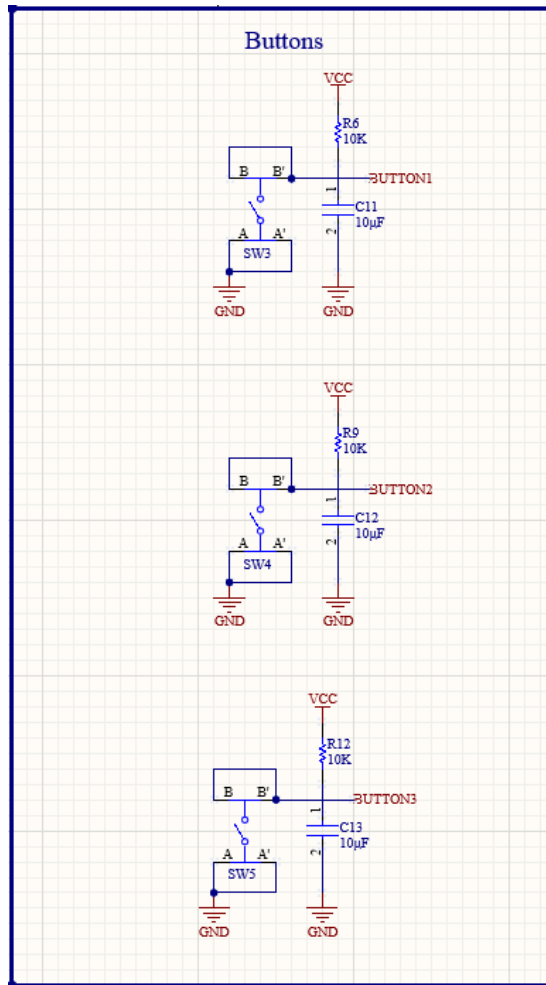
3.2.5.2. Thiết kế mạch

a. Mạch nguyên lý



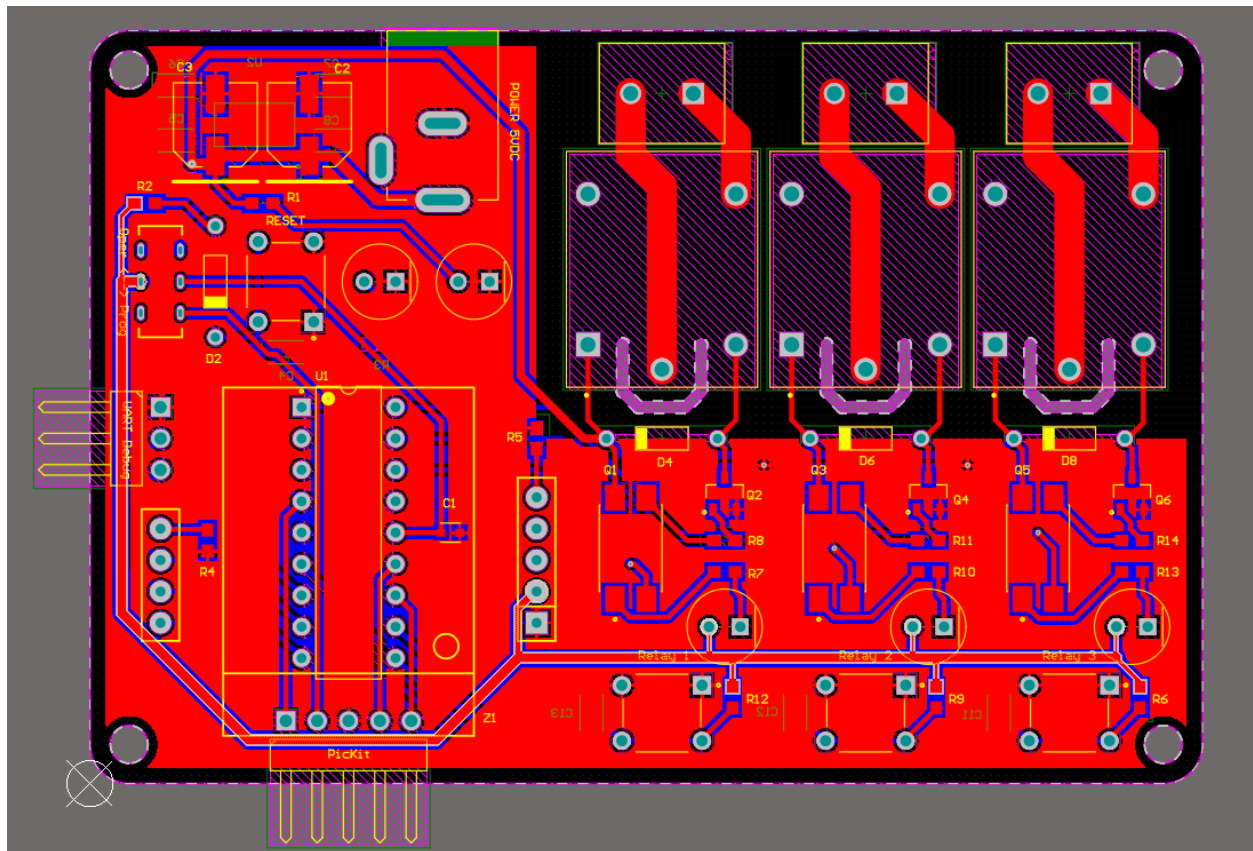
Relays

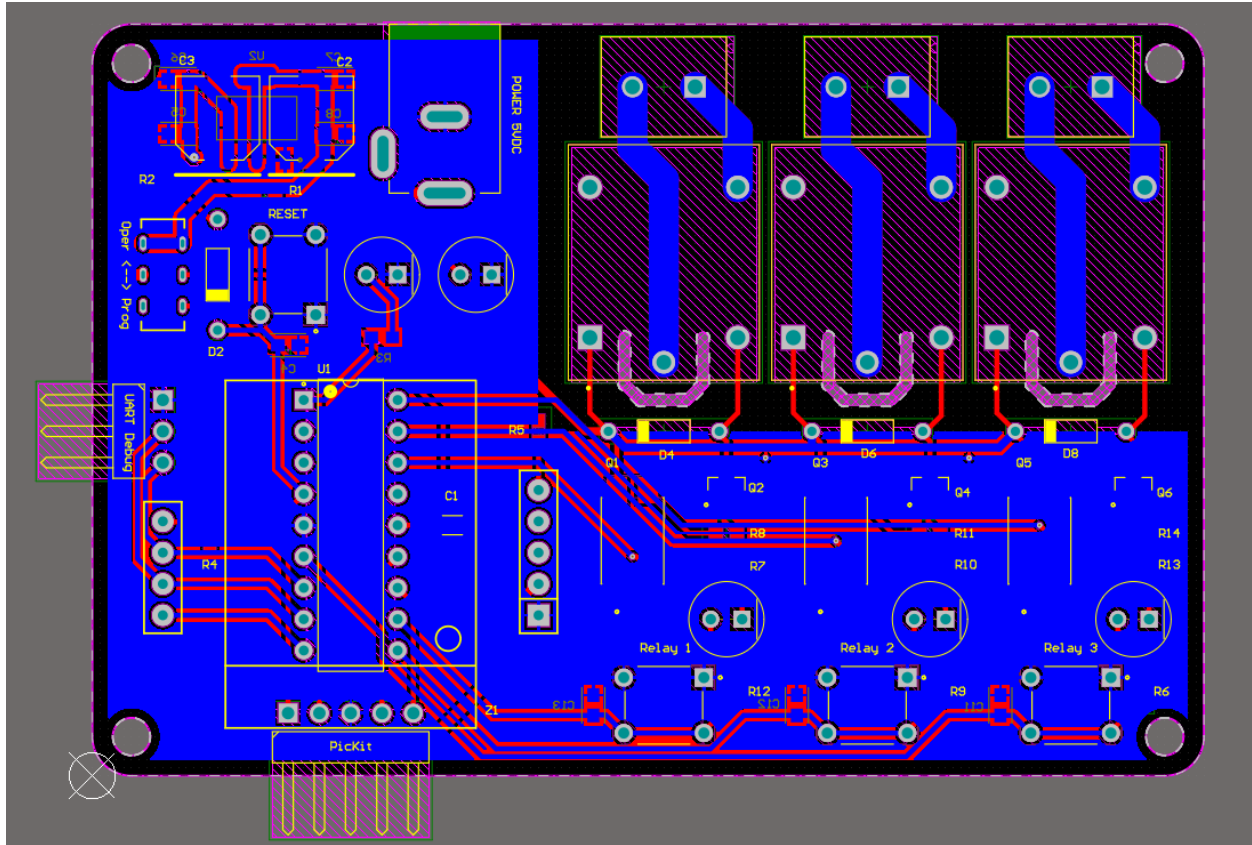




Sơ đồ nguyên lý Node Relay

b. PCB layout



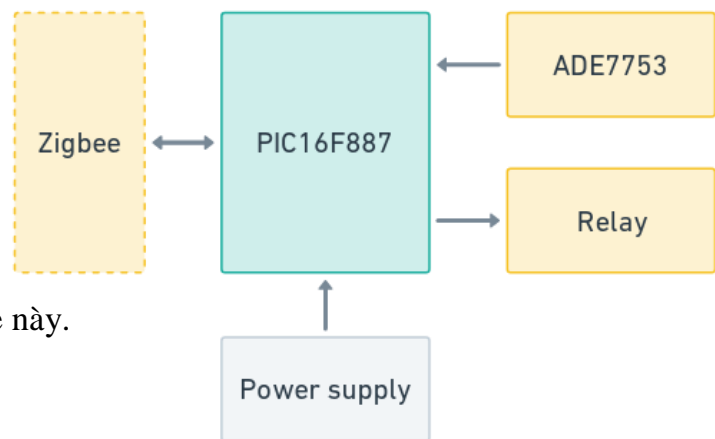


PCB Node Relay

3.2.6. Thiết kế node relay ADE

3.2.6.1. Sơ đồ khối chức năng

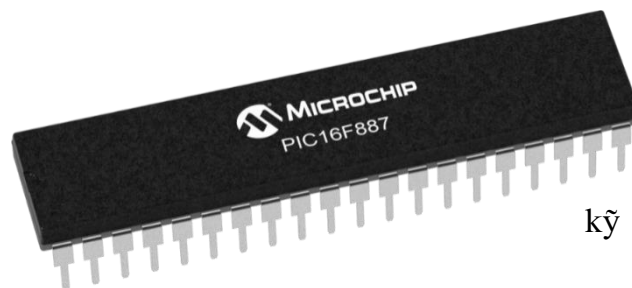
Zigbee Node Power Meter là thiết bị kết nối Zigbee có chức năng điều khiển đóng ngắt như Node Relay nhưng ngoài ra có thêm tính năng đọc và giám sát và đo lường các thông số điện năng như V_{rms} , I_{rms} , công suất và năng lượng tiêu thụ của thiết bị được lắp đặt với Node này.



Sơ đồ khối chức năng Node Power Meter

- **Khối vi điều khiển**

PIC16F887 là một chip vi điều khiển được sản xuất bởi hãng Microchip thuộc họ Pic, là một bộ vi điều khiển 8-bit dựa trên kiến trúc RISC với các thông số thuật sau:

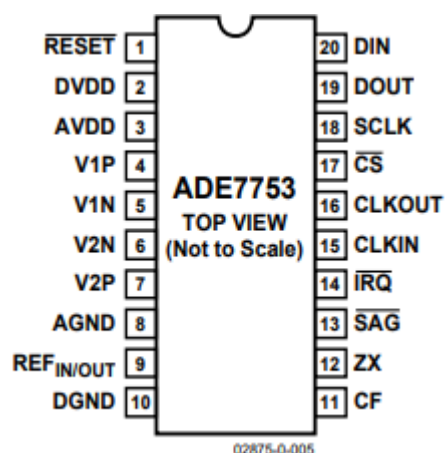


kỹ

Vi điều khiển PIC16F887

Bộ nhớ Flash	8KB
Bộ nhớ EEPROM	256B
Bộ nhớ SRAM	368B
Internal Oscillator	31kHz ~ 8MHz $\pm 1\%$
IO	35
EUSART	1
Analog comparator	2
ECCP/CCP	1/1
Timer 8-bits/16-bits	2/1
MSSP	1
10-bit ADC	14

- **ADE7753**



IC ADE7753

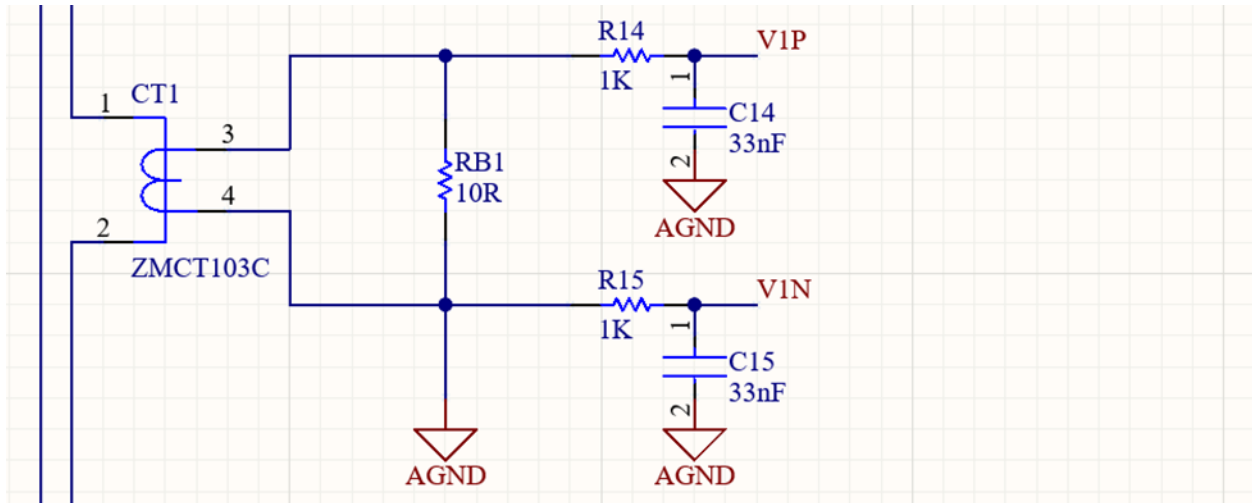
Sơ đồ chân của IC ADE7753

ADE7753 (ADE) là IC (Intergated Circuit) do hãng Analog Device sản xuất. ADE7753 có những tính năng cơ bản sau:

- Độ chính xác cao, tuân theo chuẩn ICE 61036/60687/61268, IEC62053-21, IEC 62053-22, và 62053-23.
- Tích hợp bộ tích phân số cho phép kết nối trực tiếp tới cảm biến dòng điện đầu ra tỉ lệ với di/dt.
- Một bộ PGA trong kênh dòng điện cho phép giao diện trực tiếp tới shunt và bộ biến dòng điện.
- Tính năng lượng hoạt động và năng lượng biểu kiến, dạng sóng và giá trị hiệu dụng của dòng điện và điện áp với sai số nhỏ hơn 0.1%.
- Chế độ tích lũy năng lượng dương.
- Cho phép người dùng đặt chương trình ngưỡng cho sự sụt áp, quá điện áp.
- Hiệu chuẩn số cho nguồn, pha và bù đầu vào.
- Cảm biến đo nhiệt độ trên chip ($\pm 3^{\circ}\text{C}$).
- Truyền thông nối tiếp SPI.
- Lập trình tần số xung ngõ ra.
- Yêu cầu ngắt ở chân IRQ và thanh ghi trạng thái.
- Điện áp chuẩn 2.4V, cho đưa từ ngoài.
- Nguồn nuôi 5V, công suất thấp (25 mW).

IC ADE7753 được sử dụng để đo các thông số như sau:

- Tín hiệu điện: cường độ dòng điện xoay chiều, hiệu điện thế xoay chiều.
- Các loại công suất như công suất có ích.
 - **Khởi chuyển dòng điện**



Khối chuyển dòng điện của Node Power Meter

Vì dòng tải yêu cầu đo có thể rất lớn (lên đến vài Ampere) nên ta thiết sử dụng cuộn biến dòng và điện trở burden để đo dòng đi vào IC ADE7753.

Theo tài liệu kỹ thuật của IC ADE7753, kênh V1P/V1N sẽ dùng để đo tín hiệu dòng điện. Độ lệch áp giữa V1P và V1N tối đa cho phép là ± 0.5 V.

Ta chọn hệ số an toàn là 2, chọn biến dòng 1000:1, dòng hiệu dụng tối đa là 10A. ta có công thức hiệu điện thế U1 đi vào hai chân V1P và V1N:

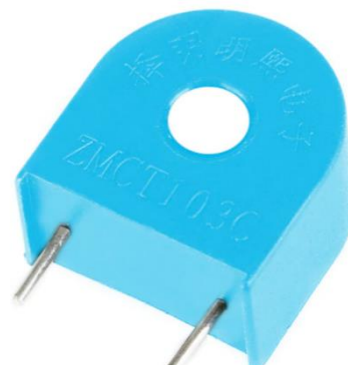
$$U1 = \frac{10\sqrt{2}}{1000} * RB1 < \frac{0.5}{2}$$

$$\Rightarrow RB1 < 17.68 \Omega$$

$$\Rightarrow \text{Chọn } RB1 = 10 \Omega$$

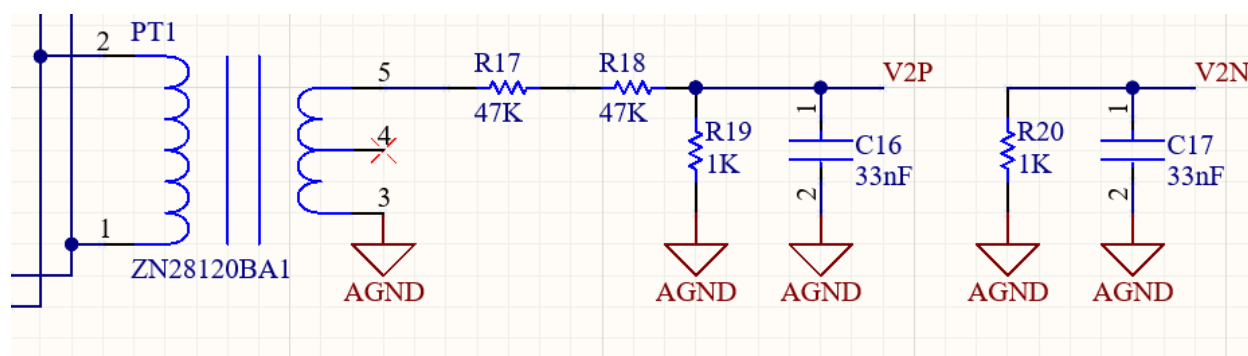
Thông số kỹ thuật:

- Tỷ lệ: 1000:1
- Phạm vi tuyến tính: 0~15A
- Sai số: 0.2%
- Dải nhiệt độ: -40 ~ +70 °C



Biến dòng ZMCT103C

• Khối chuyển điện áp



Khối chuyển điện áp của Node Power Meter

Theo tài liệu kỹ thuật của IC ADE7753, kênh V2P/V2N sẽ dùng để đo tín hiệu điện áp. Độ lệch áp giữa V2P và V2N tối đa cho phép là ± 0.5 V.

Ta chọn biến áp 220/12, chọn điện áp đỉnh $V_{max} = 250\sqrt{2}$, hệ số an toàn là 2 ta có công thức hiệu điện thế U_2 trên R_{19} (1k Ω), đồng thời cũng là hiệu điện thế đi vào hai chân V2P và V2N:

$$U_2 = \frac{250\sqrt{2} * \frac{12}{220}}{R_{17} + R_{18} + R_{19}} * R_{19} < \frac{0.5}{2}$$
$$\Rightarrow R_{17} + R_{18} > 77,14 \text{ k}\Omega$$
$$\Rightarrow \text{Chọn } R_{17} = R_{18} = 47 \text{ k}\Omega$$

Thông số kỹ thuật:

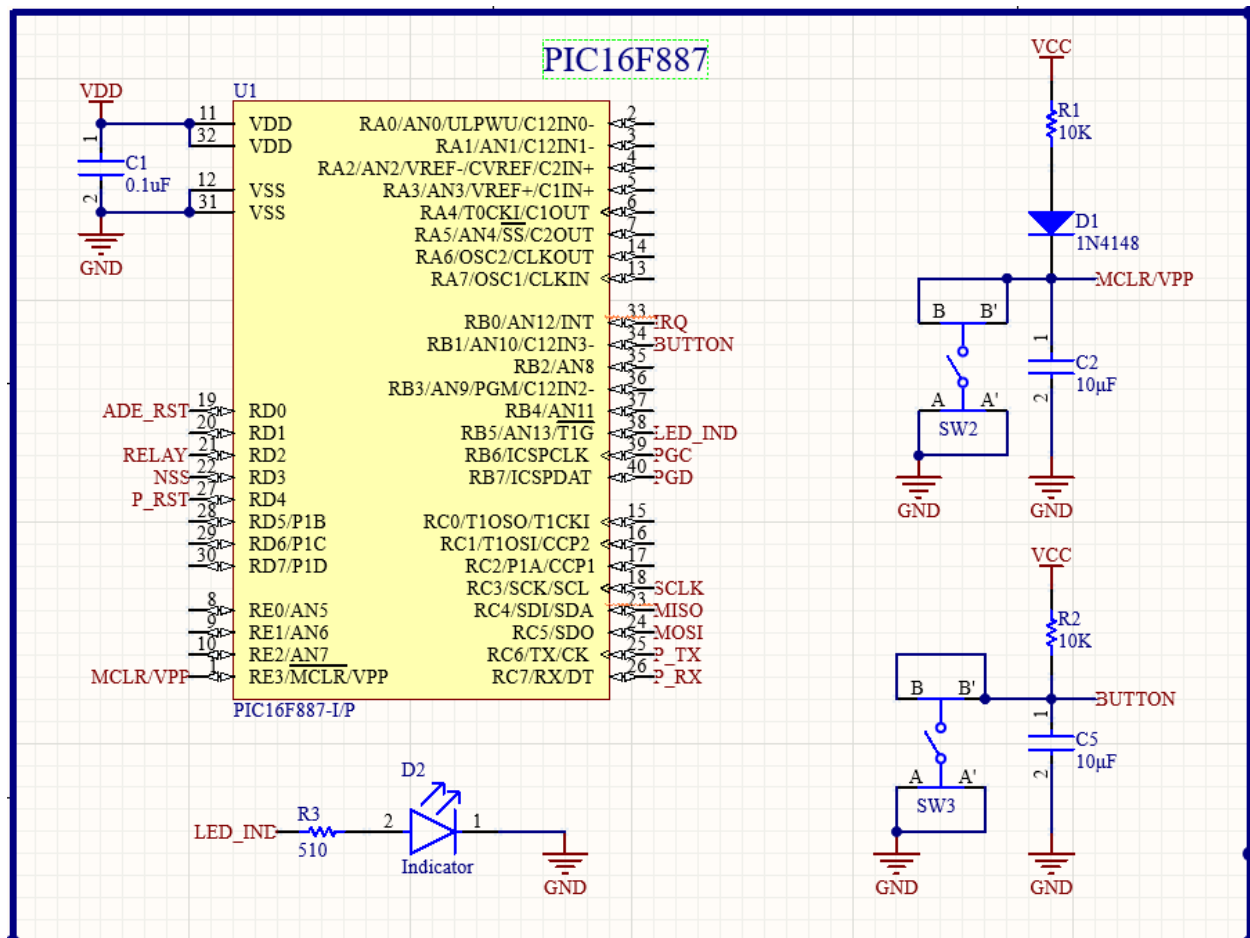
- Điện áp vào: 220VAC
- Điện áp ra: 12VAC
- Công suất 1VA



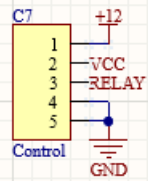
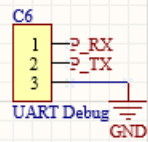
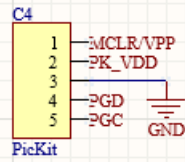
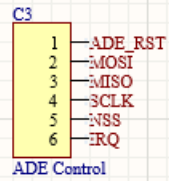
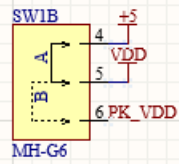
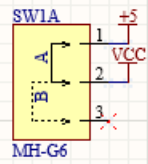
Biến áp ZN28120BA1

3.2.6.2. Thiết kế mạch

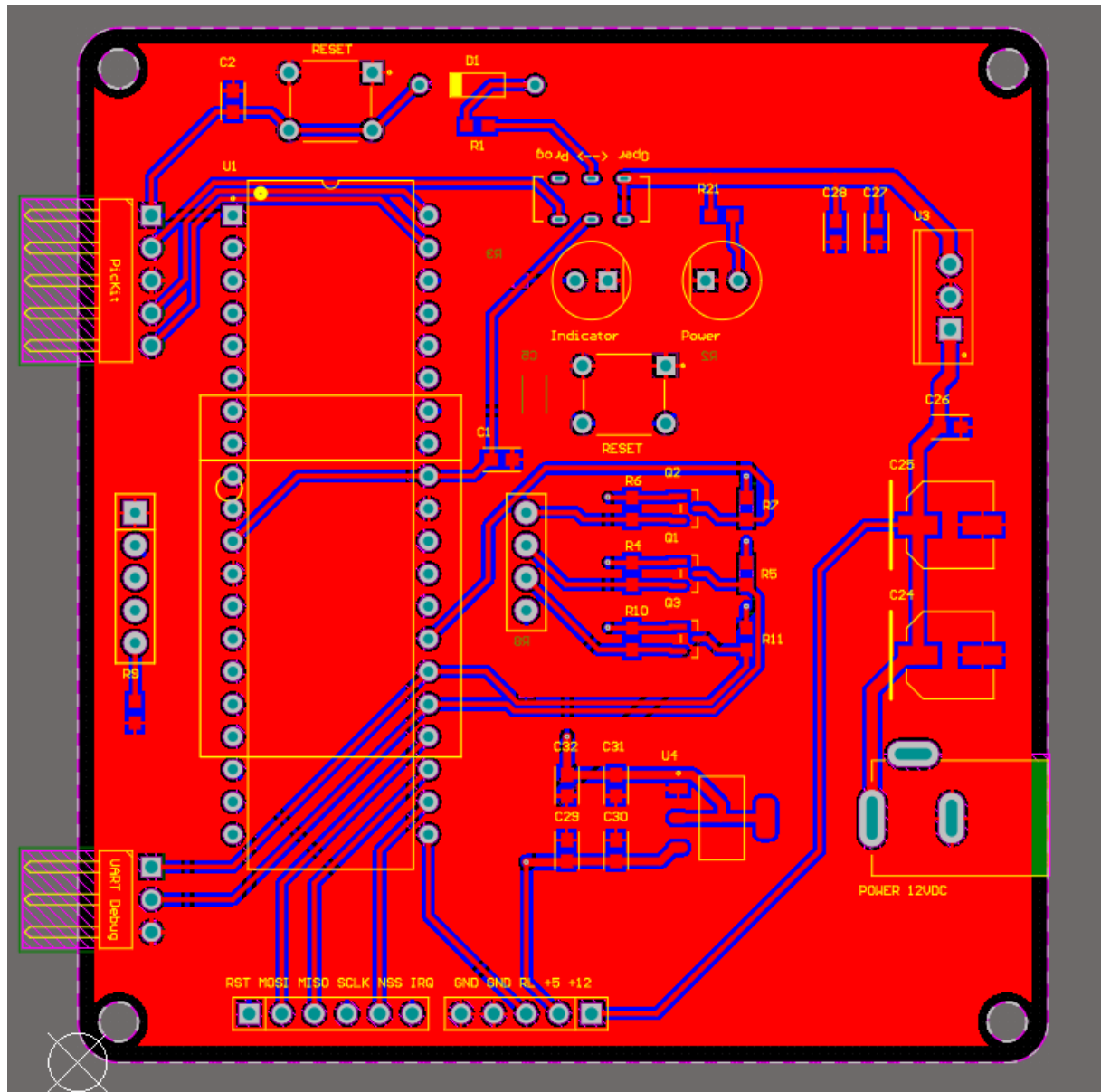
a. Mạch nguyên lý

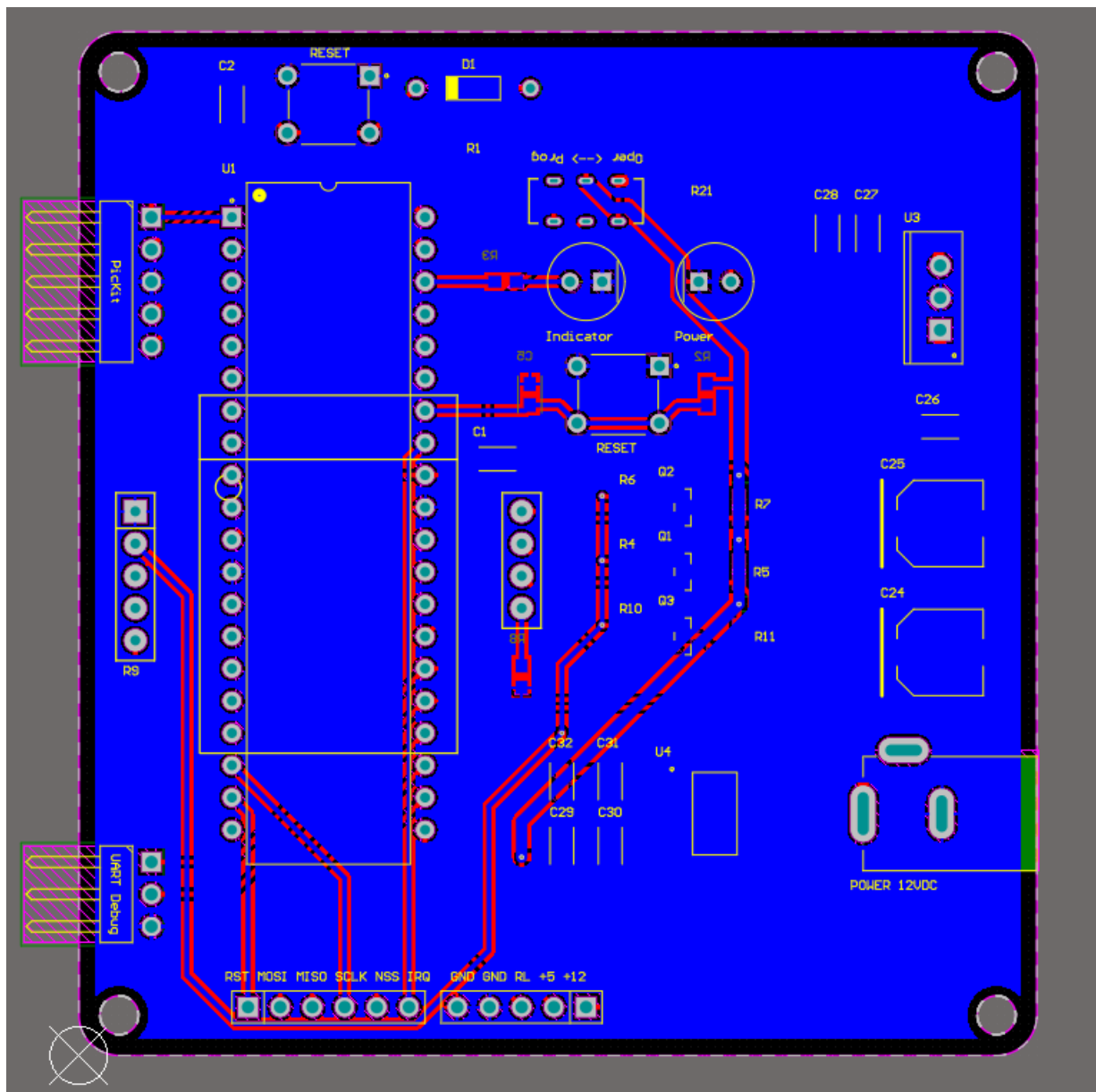


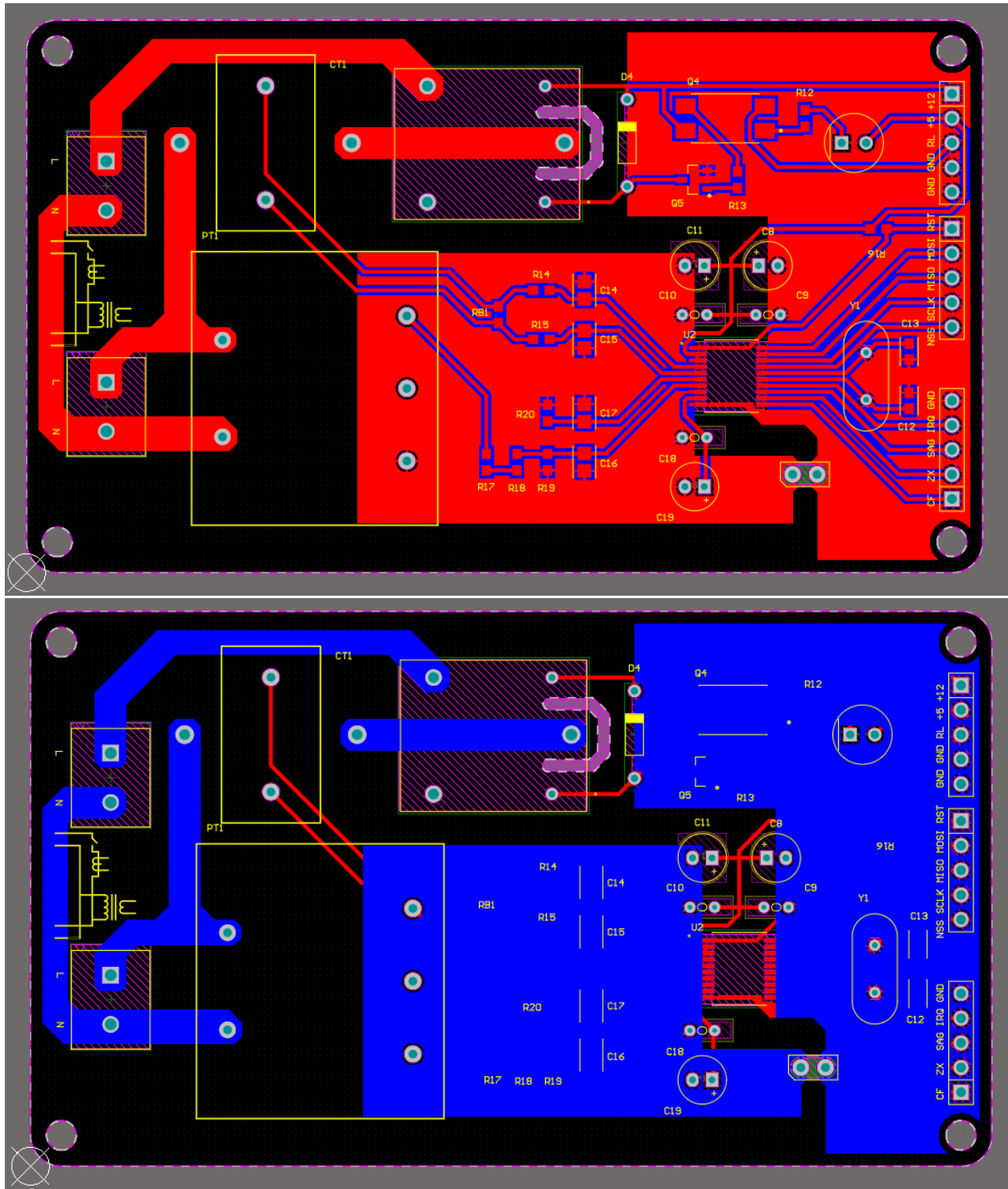
Port & Switch



b. PCB layout







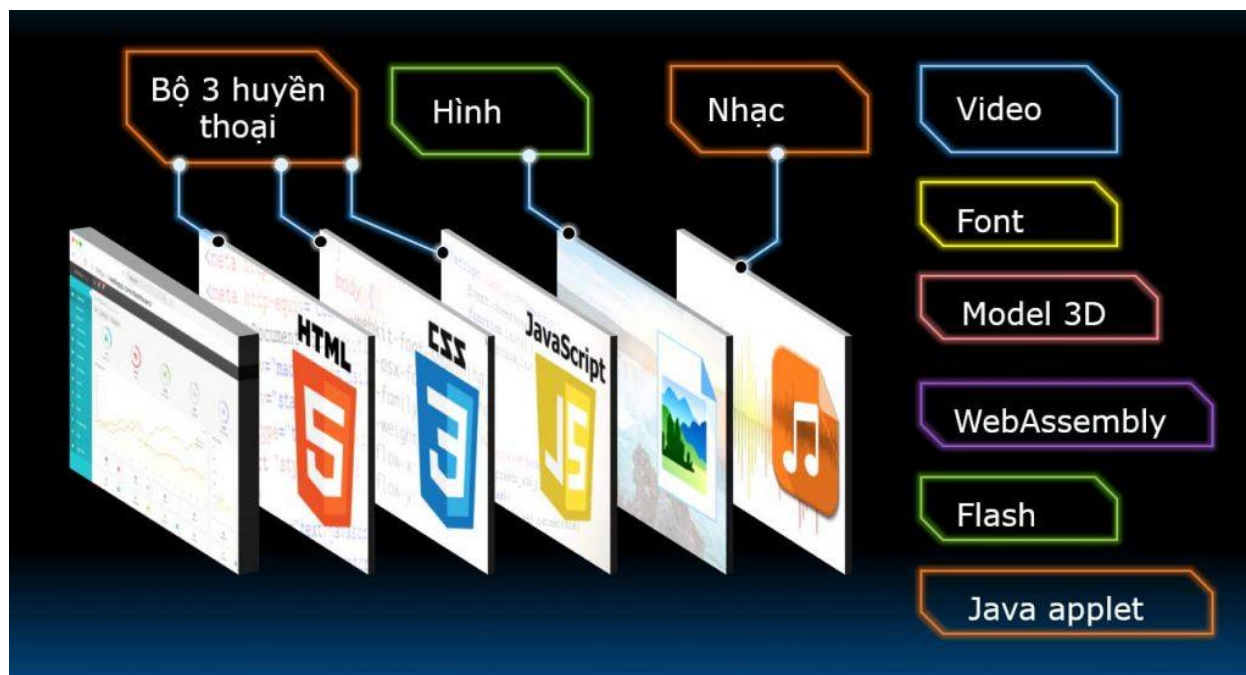
PCB Node Power Meter

Chương 4. Thiết kế website

4.1. Front-end

4.1.1. Giới thiệu

Front-end là những gì mà người dùng nhìn thấy khi truy cập vào các trang web hay ứng dụng web. Hiểu một cách đơn giản, Front-end chính là giao diện của một website. Nó mang lại cho người sử dụng các trải nghiệm tương tác (nghe, nhìn) trên trang web đó. Vì thế, Front-end còn được gọi là “Client-side”.



Front-end gồm các yếu tố tạo nên giao diện của các trang web (Nguồn: CodeSchool)

Trong thiết kế web, Front-end được tạo ra bởi 3 loại ngôn ngữ cơ bản là HTML, CSS và JavaScript. Các lập trình viên Front-end sẽ cần đảm bảo nội dung hiển thị tốt trên mọi nền tảng khác nhau.

Thông thường, Front-end của một trang web sẽ được người dùng tương tác trực tiếp theo các khía cạnh như:

- Nhận biết màu sắc chủ đạo, logo
- Khai thác thông tin trên web (văn bản, hình ảnh, âm thanh...)
- Sử dụng các button, toolbars...

4.1.2. Các ngôn ngữ cơ bản

a. HTML

HyperText Markup Language (HTML) là một trong 3 ngôn ngữ chính giúp phát triển Front-end cho trang web. Nó được dùng để mô tả cấu trúc của một trang web trên trình duyệt.

Ngôn ngữ HTML có một số đặc điểm cơ bản sau:

- Là ngôn ngữ đơn giản, dễ sử dụng với nhiều thẻ định dạng
- Sử dụng linh hoạt để thiết kế web với văn bản
- Dễ dàng thêm hình ảnh, âm thanh vào các trang web
- Có thể liên kết đến nhiều trang web khác

b. CSS

Cascading Style Sheets (CSS) cũng là một yếu tố “xương sống” trong xây dựng Front-end. Ngôn ngữ này được dùng để mô tả giao diện và định dạng hiển thị của một trang web. Nhờ có CSS, trang web sẽ trở nên trực quan và hấp dẫn hơn.

Trong lập trình web, CSS sẽ mô tả định dạng theo tài liệu được viết bằng ngôn ngữ HTML. Nó cung cấp tính năng bổ sung cho HTML. Ngoài HTML, CSS còn có thể áp dụng cho nhiều loại tài liệu khác như XML, XUL và SVG.

Về cơ bản, ngôn ngữ CSS có các đặc điểm chính như:

- CSS bổ sung các thuộc tính chi tiết hơn HTML
- Là ngôn ngữ “làm đẹp” cho trang web
- Có thể áp dụng cho nhiều dạng tài liệu khác nhau, bao gồm cả HTML, XUL, SVG...

c. Javascript

Đây là một ngôn ngữ lập trình cao cấp mà các lập trình viên Front-end luôn phải nắm rõ. Cùng với HTML và CSS, JavaScript là một trong 3 ngôn ngữ thiết yếu trong xây dựng giao diện trang web.

Đối với lập trình viên Front-end, JavaScript có một số ưu điểm như:

- JavaScript hỗ trợ tất cả các trình duyệt mà không cần plug-in
- Hỗ trợ thiết kế trang web động và các hiệu ứng hình ảnh qua DOM
- Có thể thực hiện được một số tác vụ như tự động thay đổi hình ảnh, kiểm tra thông tin nhập vào...

4.1.3. Các Framework sử dụng

Hiện nay, nhờ sự hỗ trợ của các framework, lập trình viên Front-end có thể đảm bảo nội dung hiển thị tốt trên các nền tảng khác nhau.

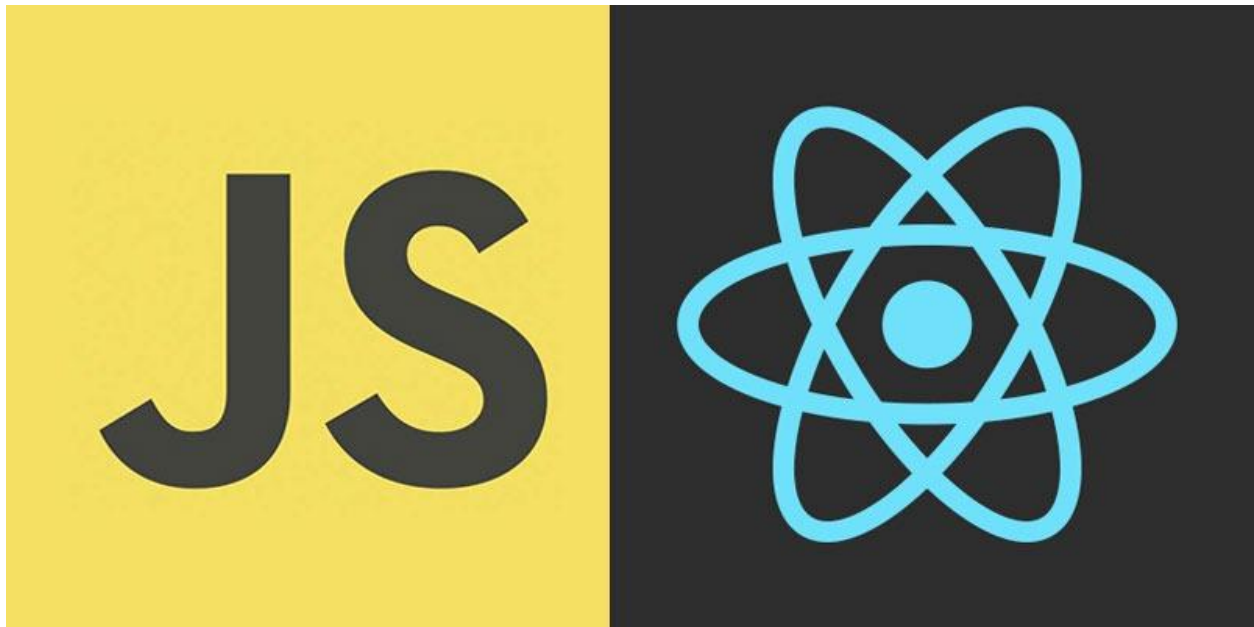
Các công ty công nghệ đã cho ra đời rất nhiều framework hỗ trợ phát triển Front-end như:

- Google: Material, AngularJS...
- Facebook: React native, React,...
- Twitter: Bootstrap, hogan.js...

Ở đây thì có sử dụng ReactJs và MUI.

4.1.3.1. ReactJs

ReactJS là một opensource được phát triển bởi Facebook, ra mắt vào năm 2013, bản thân nó là một thư viện Javascript được dùng để xây dựng các tương tác với các thành phần trên website. Một trong những điểm nổi bật nhất của ReactJS đó là việc render dữ liệu không chỉ thực hiện được trên tầng Server mà còn ở dưới Client nữa.



Javascript and ReactJs

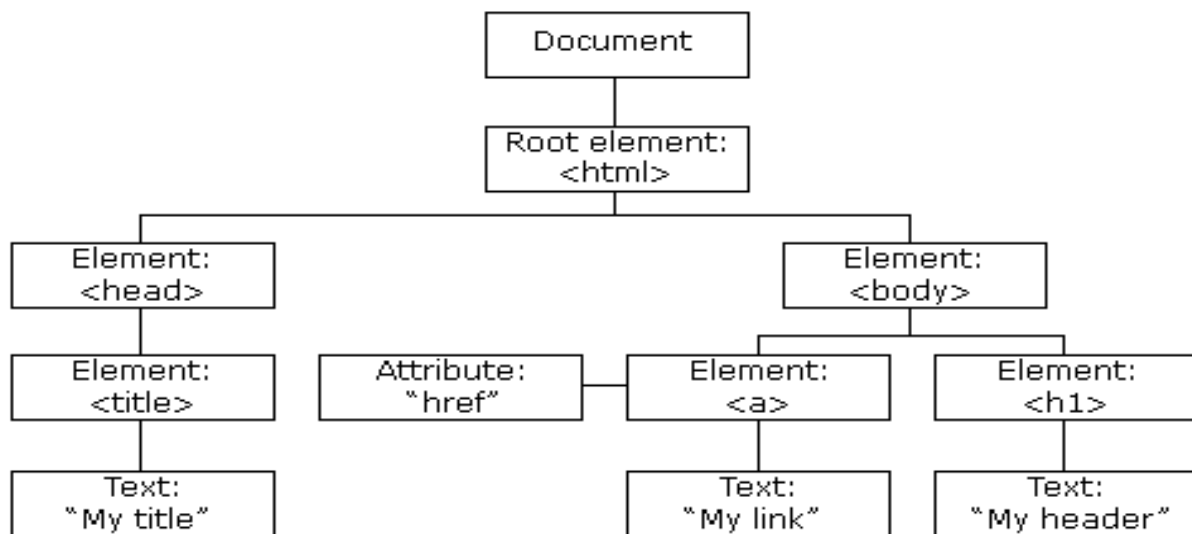
ReactJS là một thư viện JavaScript chuyên giúp các nhà phát triển xây dựng giao diện người dùng hay UI. Trong lập trình ứng dụng front-end, lập trình viên thường sẽ phải làm việc chính trên 2 thành phần sau: UI và xử lý tương tác của người dùng. UI là tập hợp những thành phần mà bạn nhìn thấy được trên bất kỳ một ứng dụng nào, ví dụ có thể kể đến bao gồm: menu, thanh tìm kiếm, những nút nhấn, card,... Giả sử bạn đang lập trình một website thương mại điện tử, sau khi người dùng chọn được sản phẩm ưng ý rồi

và nhấn vào nút “Thêm vào giỏ hàng”, thì việc tiếp theo mà bạn phải làm đó là thêm sản phẩm được chọn vào giỏ hàng và hiển thị lại sản phẩm đó khi user vào xem => xử lý tương tác.

Trước khi có ReactJS, lập trình viên thường gặp rất nhiều khó khăn trong việc sử dụng “vanilla JavaScript”(JavaScript thuần) và JQuery để xây dựng UI. Điều đó đồng nghĩa với việc quá trình phát triển ứng dụng sẽ lâu hơn và xuất hiện nhiều bug, rủi ro hơn. Vì vậy vào năm 2011, Jordan Walke – một nhân viên của Facebook đã khởi tạo ReactJS với mục đích chính là cải thiện quá trình phát triển UI.

Hơn nữa, để tăng tốc quá trình phát triển và giảm thiểu những rủi ro có thể xảy ra trong khi coding, React còn cung cấp cho chúng ta khả năng Reusable Code (tái sử dụng code) bằng cách đưa ra 2 khái niệm quan trọng bao gồm:

- JSX.
- Virtual DOM.
- a. JSX



DOM (Nguồn: https://www.w3schools.com/js/js_htmlDOM.asp)

Trọng tâm chính của bất kỳ website cơ bản nào đó là những HTML documents. Trình duyệt Web đọc những document này để hiển thị nội dung của website trên máy tính, tablet, điện thoại của bạn. Trong suốt quá trình đó, trình duyệt sẽ tạo ra một thứ gọi là Document Object Model (DOM) – một tree đại diện cho cấu trúc website được hiển thị như thế nào. Lập trình viên có thể thêm bất kỳ dynamic content nào vào những dự án của họ bằng cách sử dụng ngôn ngữ JavaScript để thay đổi cây DOM.

JSX (nói ngắn gọn là JavaScript extension) là một React extension giúp chúng ta dễ dàng thay đổi cây DOM bằng các HTML-style code đơn giản. Và kể từ lúc ReactJS

browser hỗ trợ toàn bộ những trình duyệt Web hiện đại, bạn có thể tự tin sử dụng JSX trên bất kỳ trình duyệt nào mà bạn đang làm việc.

```
1  import React from "react";
2  import "./App.css";
3
4  function App() {
5    return (
6      <div className="App">
7        <header className="App-header">
8          <h2 style={{ color: "blue", backgroundColor: "white" }}>
9            Hello From React
10          </h2>
11        </header>
12      </div>
13    );
14  }
15
16  export default App;
```

JSX Code style

JSX code Style

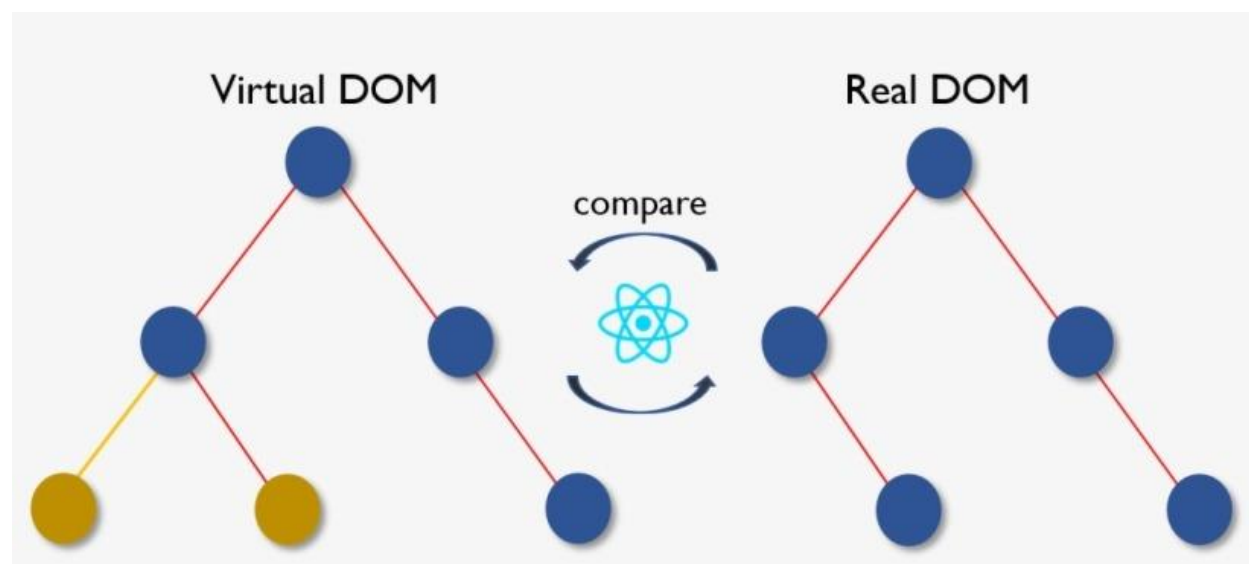
b. Virtual DOM

Nếu bạn không sử dụng ReactJS (và JSX), website của bạn sẽ sử dụng HTML để cập nhật lại cây DOM cho chính bản nó (quá trình thay đổi diễn ra tự nhiên trên trang mà người dùng không cần phải tải lại trang), cách làm này sẽ ổn cho các website nhỏ, đơn giản, static website. Nhưng đối với các website lớn, đặc biệt là những website thiên về xử lý các tương tác của người dùng nhiều, điều này sẽ làm ảnh hưởng performance website cực kỳ nghiêm trọng bởi vì toàn bộ cây DOM phải reload lại mỗi lần người dùng nhấn vào tính năng yêu cầu phải tải lại trang).

Tuy nhiên, nếu bạn sử dụng JSX thì bạn sẽ giúp cây DOM cập nhật cho chính DOM đó, ReactJS đã khởi tạo một thứ gọi là Virtual DOM (DOM ảo). Virtual DOM (bản chất của nó theo đúng tên gọi) là bản copy của DOM thật trên trang đó, và ReactJS sử dụng bản copy đó để tìm kiếm đúng phần mà DOM thật cần cập nhật khi bất kỳ một sự kiện nào đó khiến thành phần trong nó thay đổi (chẳng hạn như user nhấn vào một nút bất kỳ).

Ví dụ, khi người dùng bình luận vào khung comment vào bất kỳ bài Blog nào trên website của bạn và nhấn “Enter”. Dĩ nhiên, người dùng của bạn sẽ cần phải thấy được bình luận của mình đã được thêm vào danh sách bình luận. Giả sử trong trường hợp không sử dụng ReactJS, toàn bộ cây DOM sẽ phải cập nhật để báo hiệu sự thay đổi mới

này. Còn khi bạn sử dụng React, nó sẽ giúp bạn scan qua Virtual DOM để xem những gì đã thay đổi sau khi người dùng thực hiện hành động trên (trong trường hợp này, thêm mới bình luận) và lựa chọn đúng nơi đúng chỗ cần cập nhật sự thay đổi mà thôi.



Với việc cập nhật đúng chỗ như vậy, khỏi phải nói nó tiết kiệm cho chúng ta rất nhiều tài nguyên cũng như thời gian xử lý. Ở các website lớn và phức tạp như thương mại điện tử, đặt món ăn, v.v bạn sẽ thấy việc này là vô cùng cần thiết và quan trọng để làm tăng trải nghiệm của khách hàng và performance được cải thiện đáng kể.

4.1.3.2. MUI

Material UI là một thư viện thành phần React mã nguồn mở triển khai bởi Google's Material Design. Nó bao gồm một bộ sưu tập toàn diện các thành phần dựng sẵn sẵn sàng để sử dụng trong sản xuất ngay khi xuất xưởng.

Material UI có thiết kế đẹp mắt và có một bộ tùy chọn tùy chỉnh giúp dễ dàng triển khai hệ thống thiết kế tùy chỉnh của riêng bạn trên các thành phần.



MUI

4.2. Back-end

4.2.1. Giới thiệu

Back-End là thuật ngữ được dùng để chỉ những phần không thể nhìn thấy và không thể tương tác được của một trang web. Những thành phần này sẽ chạy trong nền để cung cấp các chức năng cần thiết cho người dùng. Có thể hiểu rằng Back-End chính là máy chủ của một trang web, có nhiệm vụ lưu trữ và sắp xếp dữ liệu sao cho website hoạt động hiệu quả nhất.

Theo đó, lập trình viên Back-End là người có nhiệm vụ xử lý những vấn đề liên quan đến dữ liệu và những nghiệp vụ chức năng để hệ thống hoạt động trơn tru. Cụ thể, Back-End sẽ thực hiện việc xây dựng mã và ngôn ngữ cho chạy trên máy chủ của website. Công việc của một lập trình viên Back-End thường không được thể hiện ra bên ngoài bởi vì họ thường thao tác với server và dữ liệu, mà những thành phần này có yêu cầu rất cao về bảo mật nên thường được mã hóa.

4.2.2. Vai trò của back-end

Trong lĩnh vực lập trình web, các lập trình viên Back-End có vai trò tạo dựng môi trường với những thuật toán logic phù hợp để ứng dụng hay website hoạt động chính xác. Việc này được thực hiện bằng cách sử dụng các ngôn ngữ lập trình như Java, .NET, PHP hay Ruby.

Các Back-End cũng đồng thời tạo ra một giải pháp lưu trữ dữ liệu tối ưu với database và đảm bảo tối ưu hóa tốc độ cùng hiệu quả của ứng dụng, website. Đối với các ứng dụng, website thì database là thành phần rất quan trọng, bởi vì đó là nơi lưu trữ toàn bộ thông tin về người dùng, bài đăng, bình luận và những tài nguyên dữ liệu khác.

Bên cạnh đó, Back-End cũng giữ vai trò phát triển hệ thống xử lý dữ liệu và đảm bảo lưu trữ dữ liệu an toàn. Họ cũng chịu trách nhiệm quản lý tài nguyên API trên các thiết bị, tham gia vào việc xây dựng khung hoặc kết cấu để việc lập trình được thuận lợi hơn.

Thông thường một Back-End sẽ đảm nhận những nhiệm vụ chính sau:

- Xác thực người dùng: đảm bảo tính chính xác các chi tiết tài khoản của người dùng và thiết lập quyền của người dùng.
- Kiểm soát trình tự hoạt động: đảm bảo tính logic các hoạt động được thực hiện trên trang web, không để xảy ra sai sót.
- Tối ưu hóa hiệu quả hoạt động: đảm bảo các chức năng của trang web hoạt động nhanh nhất có thể.

4.2.3. NodeJs



NodeJS là một trình thông dịch JavaScript, được xây dựng trên JavaScript V8 Engine của Google Chrome, và được nhiều công ty nổi tiếng trên thế giới sử dụng nó. Thông thường, NodeJS được sử dụng để chạy JavaScript trên server, và bạn có thể sử dụng nó để phát triển các trang phát video trực tuyến, hay là các ứng dụng chat online. Bạn hoàn toàn có thể sử dụng nó miễn phí và chỉnh sửa nó thoải mái bởi vì đây là một dự án mã nguồn mở.

Bạn có thể sử dụng NodeJS trên đa số các nền tảng hệ điều hành máy tính hiện nay, chẳng hạn như Windows, Linux hay là macOS. Ngoài ra, nhờ nguồn JavaScript Module phong phú, vậy nên các lập trình viên có thể tiết kiệm khá khá thời gian, đồng thời đơn giản hóa quy trình lập trình của mình. Một số ứng dụng thực tế của NodeJS mà bạn nên biết đó là Ad Server, Real-time Data Application, Cloud Services, Websocket server, RESTful API hay là Fast File Upload Client.

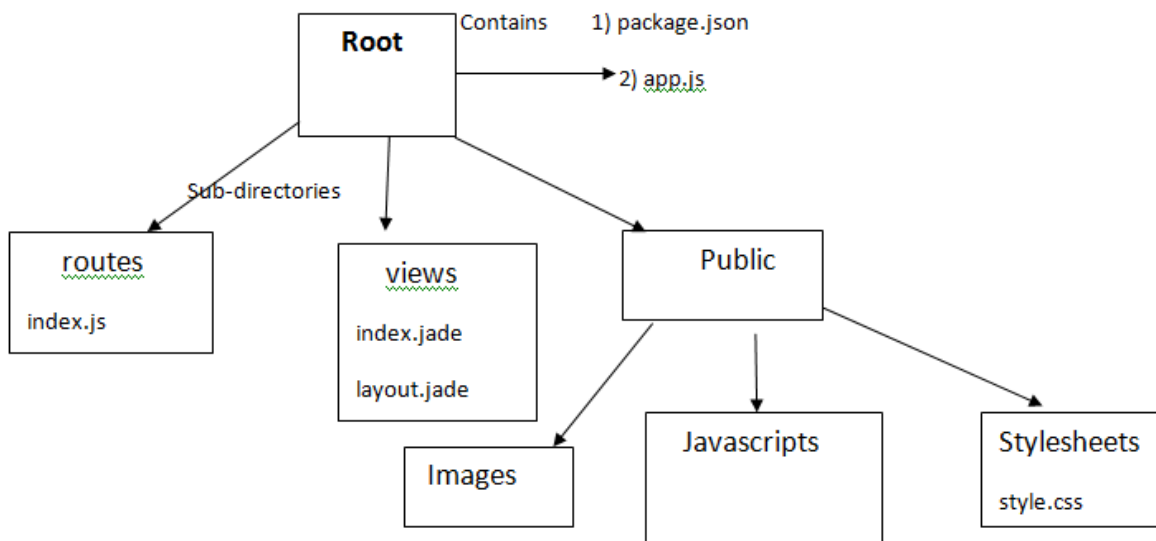
Những đặc điểm của NodeJS:

- Các APIs nằm bên trong thư viện NodeJS sẽ không được đồng bộ bởi vì NodeJS không cần đợi API trả dữ liệu về.
- NodeJS hỗ trợ lập trình viên phát triển web một cách độc lập, nhưng cũng rất nhanh chóng nhờ vào việc đây là một platform chứ không phải là framework.
- Một số nền tảng hệ điều hành máy tính hỗ trợ NodeJS đó là Windows, Linux hay là macOS.
- NodeJS không thể hỗ trợ đa luồng bởi vì NodeJS được xem như là một máy chủ đơn luồng.

- Lập trình viên cần nắm vững các kiến thức lập trình cơ bản như là các giao thức hay là JavaScript, bởi vì NodeJS không được xem như là một ngôn ngữ lập trình.
- Người mới bắt đầu sử dụng NodeJS không cần quá lo lắng bởi vì cộng đồng hỗ trợ cho NodeJS thường rất lớn, nên bạn hoàn toàn có thể tìm thấy sự giúp đỡ của họ khi gặp khó khăn.
- Tốc độ xử lý tương đối cao nhờ vào việc được viết bằng ngôn ngữ C++, do đó nó đáp ứng được thời gian chạy trên đa thiết bị, đa nền tảng.

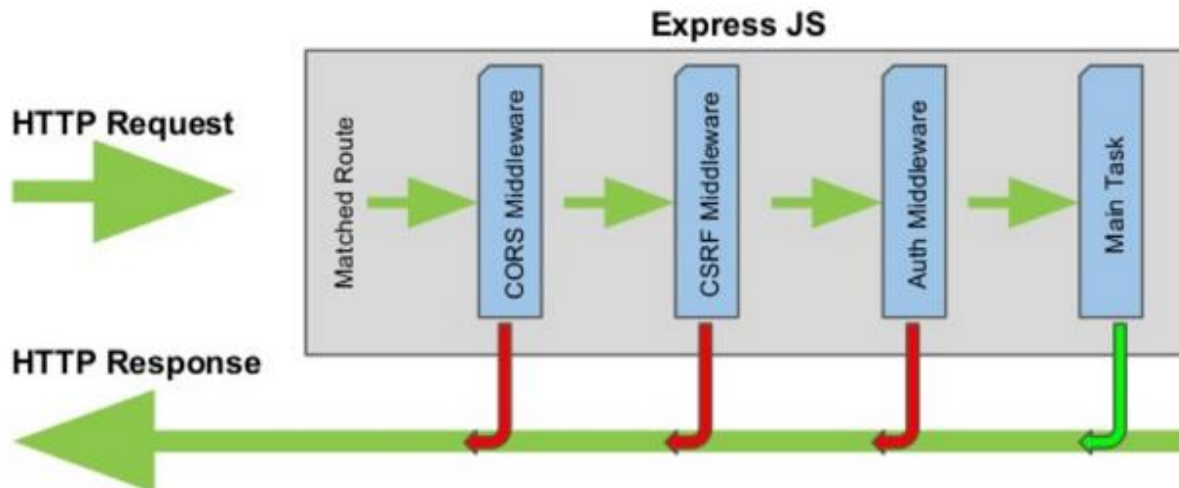
4.2.4. ExpressJs

Expressjs là một framework được xây dựng trên nền tảng của Nodejs. Nó cung cấp các tính năng mạnh mẽ để phát triển web hoặc mobile. Expressjs hỗ trợ các method HTTP và middleware tạo ra API vô cùng mạnh mẽ và dễ sử dụng.



Tổng hợp một số chức năng chính của Expressjs như sau:

- Thiết lập các lớp trung gian để trả về các HTTP request.
- Define router cho phép sử dụng với các hành động khác nhau dựa trên phương thức HTTP và URL.
- Cho phép trả về các trang HTML dựa vào các tham số.



4.3. Database

4.3.1. Giới thiệu

Database là một tập hợp dữ liệu có hệ thống hay còn được gọi là cơ sở dữ liệu. Database được hình dung như một kho sách ngăn nắp có chia đề mục và phân loại rõ ràng. Chỉ khác là Database hoạt động trên nền tảng điện tử có phần trừu tượng hơn. Tuy nhiên nó cũng sở hữu đủ các tính năng như quản lý, điều phối, trích xuất, tìm kiếm hay chỉnh sửa... dữ liệu.

Database có thể được lưu trữ trên các phần mềm online, offline dưới dạng một tệp tin hoặc một hệ thống dữ liệu. Database cũng thường được lưu trên các thiết bị có chức năng ghi nhớ như ổ cứng, thẻ nhớ, USB...

Thành phần chính của database:

- **Phần cứng:** Phần cứng được hiểu như các thiết bị vật lý giúp kết nối người dùng và hệ thống dữ liệu. Phần cứng thường là máy tính, điện thoại di động, các thiết bị I/O hay USB, thẻ nhớ... Ví dụ, nếu cơ sở dữ liệu của bạn nằm trên máy tính cá nhân thì máy tính này sẽ là phần cứng. Vẫn là cơ sở dữ liệu đó nhưng bạn lại tải về và bật nó trên điện thoại thì phần cứng sẽ là điện thoại.
- **Phần mềm:** Ngược lại với phần cứng, phần mềm là các chương trình trừu tượng không thể có tiếp xúc vật lý với người dùng. Phần mềm được dùng để quản lý và điều khiển các Database. Có rất nhiều loại phần mềm khác nhau, có thể là phần mềm điều hành, phần mềm chia sẻ dữ liệu giữa nhiều người dùng khác nhau hoặc bản thân Database cũng được hiểu như một dạng phần mềm.
- **Dữ liệu:** Dữ liệu còn được gọi là Data theo ngôn ngữ công nghệ. Các dữ liệu này đến từ nhiều nguồn khác nhau, được lưu trữ bằng nhiều cách khác nhau. Dữ liệu cũng có thể tồn tại dưới rất nhiều hình thức, ví dụ như hình ảnh, âm thanh, văn

bản... và thậm chí là hệ ngôn ngữ nhị phân. Tuy nhiên Data vẫn chưa thể được sử dụng ngay lập tức, nó vẫn tồn tại dưới dạng “nguyên liệu thô” và đang chờ để được Database xử lý.

- **Quy trình:** Một Database có thể được sử dụng bởi nhiều người dùng khác nhau, nhất là trong một team phát triển sản phẩm. Vì thế thật bất tiện nếu những thành viên trong nhóm cứ phải liên tục hỏi nhau về cách thao tác trong Database phải không nào? Để giải quyết vấn đề này, người tạo Database sẽ soạn một hướng dẫn chi tiết về cách mà Database vận hành cũng như cách sử dụng nó. Thường thì quy trình này sẽ được tài liệu hóa để người đọc có thể dễ hiểu hơn.
- **Ngôn ngữ truy cập:** Ngôn ngữ truy cập được dùng để tham gia vào một Database. Nó cho phép người dùng truy cập vào dữ liệu cũ, cập nhật dữ liệu mới hoặc trích xuất và thực hiện nhiều thao tác khác trong Database. Ngôn ngữ truy cập thường sẽ được quy định bởi người khởi tạo ra Database. Nếu người dùng không biết cách sử dụng ngôn ngữ có sẵn của Database thì có thể dùng các trình biên dịch.

4.3.2. MongoDB



MongoDB là một dạng phần mềm cơ sở dữ liệu sử dụng mã nguồn mở NoSQL. Nó có thể hỗ trợ trên nhiều nền tảng khác nhau và được thiết kế với mục đích hướng đến đối tượng. MongoDB hoạt động dựa vào các khái niệm Collection và Document. Đồng thời, nó có hiệu suất cao cùng với tính khả dụng tốt và dễ dàng mở rộng.

Các Collection trong MongoDB có cấu trúc cực kỳ linh hoạt. Điều này cho phép dữ liệu không cần thiết phải tuân theo bất kỳ một dạng cấu trúc nào. Vì thế, MongoDB có thể lưu trữ những dữ liệu có cấu trúc đa dạng và phức tạp. Dữ liệu trong MongoDB được lưu bằng định dạng kiểu JSON.

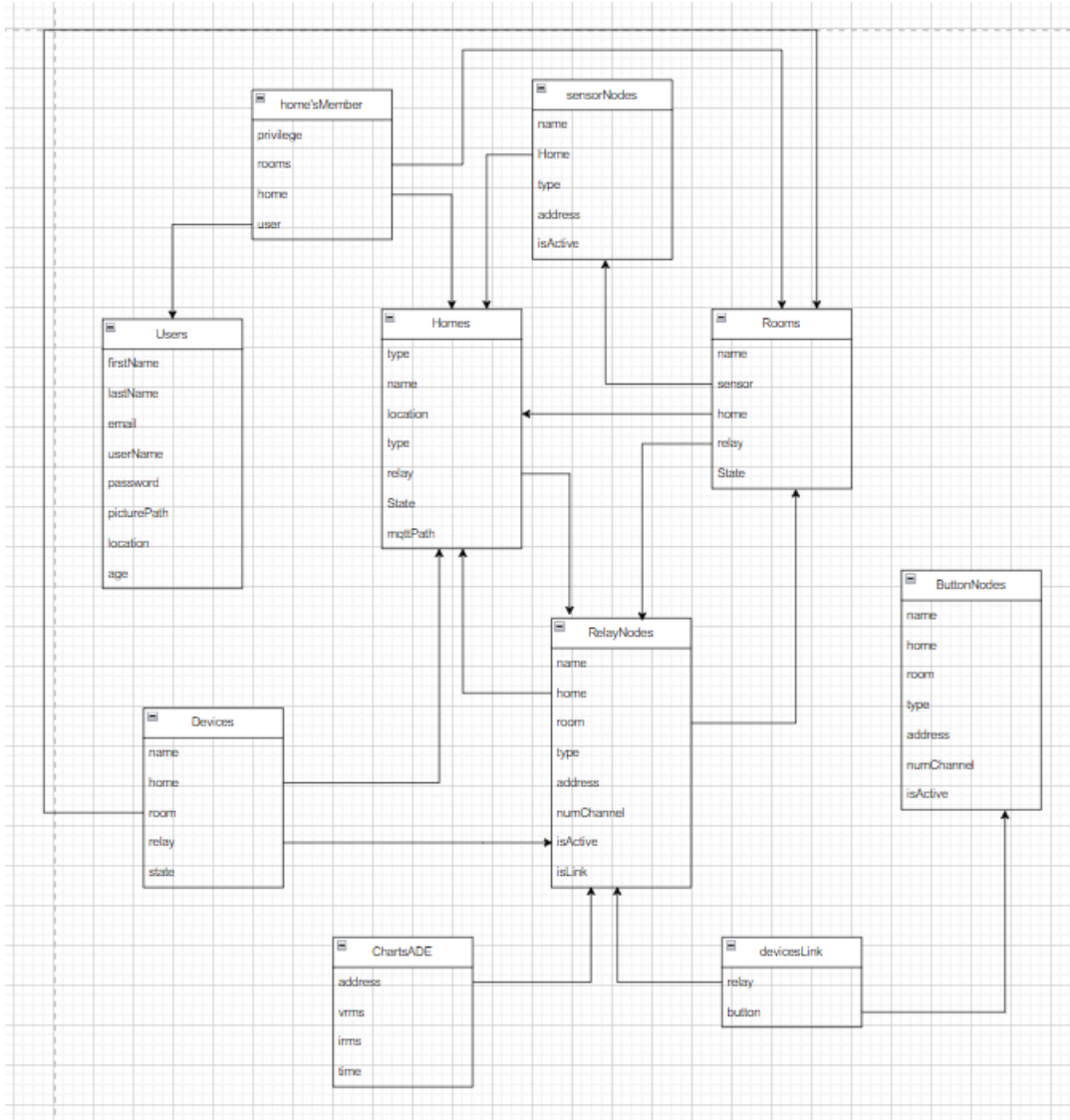
NoSQL còn được xem là mảnh vá cho những hạn chế, khiếm khuyết của mô hình Hệ quản trị cơ sở dữ liệu quan hệ (Relational Database Management System - RDBMS). Đó là cải tiến về tốc độ, tính năng cũng như khả năng mở rộng.

NoSQL cho phép mở rộng dữ liệu mà không cần tạo khóa ngoại, khóa chính hay kiểm tra tính ràng buộc... Có thể thấy, dạng cơ sở dữ liệu này có thể linh động, sẵn sàng bỏ qua tính toàn vẹn cứng nhắc của transaction hay dữ liệu nhằm đổi lấy hiệu suất, cùng với khả năng mở rộng tốt. Chính vì ưu điểm này mà nhiều Tập đoàn lớn rất ưa chuộng NoSQL như Facebook, Google,...

Hoạt động của MongoDB:

- Nguyên tắc hoạt động của MongoDB là dưới một tiến trình dịch vụ ngậm và mở một cổng (mặc định là cổng 27017), để có thể tiếp nhận các yêu cầu truy vấn, thao tác; sau đó tiến hành xử lý.
- Mỗi bản ghi của MongoDB (document) được gán một trường có tên “_id” nhằm xác định tính duy nhất của bản ghi. Có thể hiểu id này như tên gọi của một bản ghi và dùng phân biệt chúng với các bản ghi khác. Đồng thời, nó còn được sử dụng cho mục đích truy vấn hoặc tìm kiếm thông tin. Trường dữ liệu “_id” được tự động đánh chỉ mục (index) để đảm bảo tốc độ truy vấn đạt hiệu suất tối ưu.
- Mỗi truy vấn dữ liệu đều được ghi đệm lên bộ nhớ RAM nên các truy vấn sau đó sẽ diễn ra nhanh hơn. Bởi nó không cần đọc dữ liệu từ ổ cứng.
- Khi thực hiện thêm, xóa hay sửa bản ghi thì MongoDB đều mất 60s để ghi các dữ liệu được thay đổi từ RAM xuống ổ cứng. Điều này nhằm mục đích đảm bảo hiệu suất mặc định của chương trình.

4.3.3. Thiết kế



Bản thiết kế database

Chương 5. Thiết kế kết nối giữa thiết bị và website

5.1. Giới thiệu về giao thức MQTT

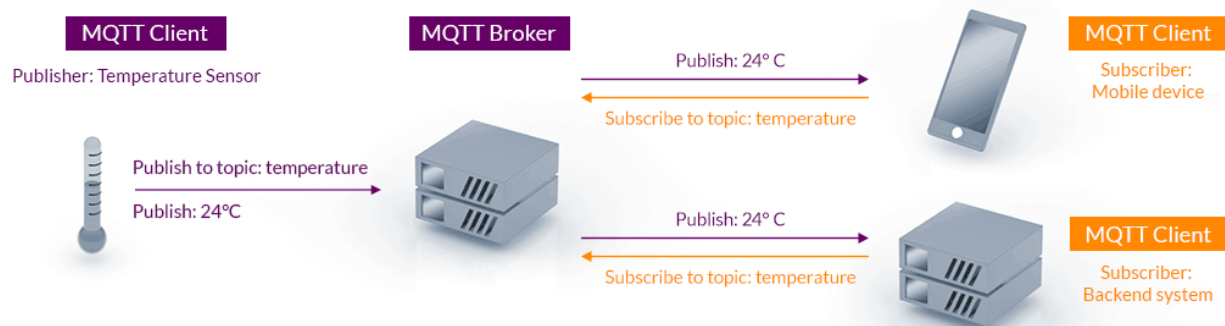
5.1.1. Khái niệm

MQTT (Message Queuing Telemetry Transport) là một giao thức nhắn tin tiêu chuẩn OASIS cho Internet of Things (IoT). Nó được thiết kế như một phương tiện truyền tải tin nhắn publish/subscribe (xuất bản/đăng ký) cực kỳ nhẹ, lý tưởng để kết nối các thiết bị từ xa với băng thông mạng thấp. MQTT ngày nay được sử dụng trong nhiều ngành công nghiệp, chẳng hạn như ô tô, sản xuất, viễn thông, dầu khí, v.v.

Organization For The Advancement Of Structured Information Standards (OASIS) là Tổ chức vì sự tiến bộ của tiêu chuẩn thông tin có cấu trúc (OASIS).

OASIS xây dựng các tiêu chuẩn mở hướng tới giảm chi phí toàn cầu và thúc đẩy sự đổi mới. Các tổ chức đã xây dựng các tiêu chuẩn mở trong nhiều lĩnh vực công nghệ then chốt, trong đó có an ninh, điện toán đám mây, công nghệ nội dung, dịch vụ Web và chính phủ điện tử. Mỗi tiêu chuẩn được xây dựng với sự giúp đỡ của các thành viên OASIS và chỉ được công bố nếu được chấp thuận theo đa số.

5.1.2. Cách hoạt động



Một phiên MQTT được chia thành bốn giai đoạn: kết nối, xác thực, giao tiếp và kết thúc. Client (máy khách) bắt đầu bằng cách tạo kết nối Transmission Control Protocol/Internet Protocol (TCP/IP) tới broker bằng cách sử dụng cổng tiêu chuẩn hoặc cổng tùy chỉnh được xác định bởi các nhà phát triển broker. Khi tạo kết nối, điều quan trọng là phải nhận ra rằng máy chủ có thể tiếp tục một phiên cũ nếu nó được cung cấp ID máy khách mà được sử dụng lại.

Các cổng tiêu chuẩn là 1883 cho giao tiếp không mã hóa và 8883 cho giao tiếp được mã hóa – sử dụng Lớp cổng bảo mật (SSL) / Bảo mật lớp truyền tải (TLS). Trong quá trình giao tiếp SSL/TLS, máy khách cần kiểm chứng và xác thực máy chủ. Máy khách cũng có thể cung cấp tính xác thực máy khách cho broker trong quá trình giao tiếp. Broker có thể sử dụng điều này để xác thực máy khách. Mặc dù không phải là một phần

cụ thể của đặc trưng MQTT, nhưng các broker đã trở thành thông lệ để hỗ trợ xác thực máy khách bằng SSL/TLS phía máy khách.

Trong đó:

- Broker: là một thành phần trung gian (môi giới)
- SSL/TLS (Secure Sockets Layer/Transport Layer Security) là kỹ thuật mã hóa truyền tin trên internet.

Giao thức MQTT được phát hành nhằm mục đích trở thành một giao thức dành cho các thiết bị IoT và hạn chế tài nguyên tiêu tốn, SSL/TLS không phải là một tùy chọn bắt buộc và trong một số trường hợp nó không cần thiết. MQTT được gọi là một giao thức nhẹ vì tất cả các thông điệp của nó chỉ có một mã nhỏ. Mỗi thông báo bao gồm một tiêu đề cố định – 2 byte – một tiêu đề biến tùy chọn, dung lượng thông điệp được giới hạn ở 256 megabyte (MB) và mức chất lượng dịch vụ (QoS).

Trong giai đoạn giao tiếp, máy khách có thể thực hiện các thao tác publish, subscribe, unsubscribe và ping. Thao tác publish sẽ gửi một khối dữ liệu nhị phân – nội dung – đến một chủ đề do nhà xuất bản xác định.

MQTT hỗ trợ các thông điệp đối tượng nhị phân có kích thước lớn lên đến 256 MB. Định dạng của nội dung sẽ dành riêng cho từng ứng dụng. Subscribe được thực hiện bằng cách sử dụng gói SUBSCRIBE/SUBACK và việc unsubscribe được thực hiện tương tự bằng cách sử dụng gói UNSUBSCRIBE/UNSUBACK.

Một thao tác khác mà máy khách có thể thực hiện trong giai đoạn giao tiếp là ping broker server bằng cách sử dụng gói PINGREQ/PINGRESP. Thao tác này không có chức năng nào khác ngoài việc duy trì kết nối trực tiếp và đảm bảo kết nối TCP không bị ngắt bởi gateway hoặc router.

Khi một publisher hoặc subscriber muốn kết thúc một phiên MQTT, nó sẽ gửi một thông báo DISCONNECT – HỦY KẾT NỐI đến broker và sau đó ngắt kết nối. Điều này được gọi là graceful shutdown vì nó mang lại cho máy khách khả năng dễ dàng kết nối lại bằng cách cung cấp ID máy khách và tiếp tục lại nơi nó đã dừng.

Nếu việc ngắt kết nối xảy ra đột ngột mà không có thời gian để publisher gửi thông điệp HỦY KẾT NỐI, broker có thể gửi cho subscriber một thông điệp từ publisher mà broker đã lưu vào bộ nhớ cache trước đó. Thông điệp, được gọi là testament, cung cấp cho subscriber hướng dẫn về những việc cần làm nếu publisher ngắt kết nối đột ngột.

5.2. Thiết kế dữ liệu truyền thông qua MQTT

5.2.1. Cấu trúc gói tin từ hub lên server

- Gói tin khi thiết bị mới được thêm vào mạng

```
{  
    "action": "provision",  
    "dev_addr": ".." (địa chỉ của thiết bị gửi lên),  
    "dev_type": "Sensor"/"RelayAde"/"Relay3"  
}
```

- Gói tin gửi thông tin dữ liệu thiết bị

```
{  
    "action": "telemetry",  
    "dev_addr": ".." (địa chỉ của thiết bị gửi lên),  
    "attributes": {  
        "temperature": 25,  
        "humidity": 40,  
        "air_quality" : 0.2  
    }  
}
```

```
{  
    "action": "telemetry",  
    "dev_addr": ".." (địa chỉ của thiết bị gửi lên),  
    "status1" : "ON"/"OFF",  
    ..  
}
```

```

{
  "action": "telemetry",
  "dev_addr": ".." (địa chỉ của thiết bị gửi lên),
  "status": true,
  "attributes": {
    "irms": 25,
    "vrms": 40,
    "power" : 0.2
  }
}

```

5.2.2. Cấu trúc gói tin từ serber xuống hub

- Gói tin để thêm thiết bị mới vào mạng

```

{
  "action": "command",
  "command": "permit_join"
}

```

- Gói tin để loại bỏ thiết bị ra khỏi mạng

```

{
  "action": "command",
  "command": "leave_req",
  "dev_addr": ".." (địa chỉ của thiết bị rời đi)
}

```

- Gói tin điều khiển thiết bị

```
{  
  "action": "control",  
  "dev_addr": ".." (địa chỉ của thiết bị gửi lên),  
  "status": "ON"  
}  
  
{  
  "action": "control",  
  "dev_addr": ".." (địa chỉ của thiết bị gửi lên),  
  "status1": "ON",  
  "status2": "OFF",  
  "status3": "OFF",  
}
```


Chương 6. Kết luận và hướng phát triển

6.1. Kết luận

Kết nối và điều khiển từ xa được bằng mạng Zigbee.

Giao tiếp được MQTT giữa server và Hub.

Chưa hoàn thiện lại được một số chức năng.

6.2. Hướng phát triển

Hoàn thiện lại trang web.

Thiết kế lại phần kết nối giữa node Button và Relay.

Hoàn thiện lại node Power Meter.

Cải thiện kích thước các node.

Tài liệu tham khảo

- [1] Giải pháp tự động hóa IoT, <https://giaiphap.mctt.com.vn> [online]. Available:
<https://giaiphap.mctt.com.vn/tong-quan-ve-mang-khong-day-zigbee-ung-dung-cua-zigbee-trong-thuc-te/>
[Accessed 25 05 2022].
- [2] Wikipedia, https://vi.wikipedia.org/wiki/Trang_Chinh [online]. Available:
https://vi.wikipedia.org/wiki/Vi_dieu_khiển
[Accessed 30 05 2022].
https://vi.wikipedia.org/wiki/Nhà_thông_minh
[Accessed 30 05 2022].
- [3] Analog Devices, <https://www.analog.com/en/index.html> [online]. Available:
<https://www.analog.com/media/en/technical-documentation/data-sheets/ADE7753.pdf>
[Accessed 30 05 2022].
- [4] Microchip Technology, <https://www.microchip.com> [online]. Available:
<https://www.microchip.com/en-us/product/PIC16F887>
[Accessed 30 05 2022].
- [5] Manualzz, <https://manualzz.com> [online]. Available:
<https://manualzz.com/doc/17406994/cc2530>
[Accessed 30 05 2022].
- [6] Báo cáo luận văn, “Thiết kế và thi công mô hình nhà thông minh sử dụng mạng không dây Zigbe” của bạn Phạm Minh Trí và bạn Phạm Đức Ái.