

TRƯỜNG ĐẠI HỌC BÁCH KHOA - ĐẠI HỌC QUỐC GIA TP.HCM
BỘ MÔN VIỄN THÔNG



Đồ án
**Thiết kế gateway chuyển đổi
mạng Zigbee sang MQTT**

GVHD: TS Võ Quế Sơn (sonvq@hcmut.edu.vn)
Sinh viên: Phan Nguyên Trung - 1814519



Lời cảm ơn

Đầu tiên, em xin bày tỏ lòng biết ơn đến chân thành sâu sắc tới giáo viên hướng dẫn của em, thầy **TS Võ Quê Sơn** - “giảng viên bộ môn Viễn Thông” đã trực tiếp giúp đỡ, hướng dẫn em hoàn thành đồ án này.

Với điều kiện thời gian cũng như kinh nghiệm còn hạn chế của một học viên, bài báo cáo này không thể tránh được những thiếu sót. Em rất mong nhận được sự chỉ bảo, đóng góp ý kiến của các quý thầy cô để em có điều kiện bổ sung, nâng cao ý thức của mình, phục vụ tốt hơn công tác thực tế sau này.

Em xin chân thành cảm ơn.

Tp. Hồ Chí Minh, tháng 05 năm 2021.

Sinh viên



Tóm tắt đồ án

Đồ án với mục tiêu, nghiên cứu thiết kế gateway chuyển đổi mạng zigbee sang MQTT. Đồng thời tìm hiểu cách thức hoạt động của giao thức MQTT, thông qua giao thức MQTT điều khiển các thiết bị trong mạng zigbee.

Đồ án sử dụng một **Module CC2530** và máy tính để tạo gateway chuyển đổi mạng Zigbee sang MQTT. Một module **CC2530** làm node mạng Zigbee cũng là router . Một module **Silabs RGB** là thiết bị zigbee đầu cuối (End Devices). Thông qua giao thức MQTT điều khiển các chức năng trên **Silabs RGB**.

Mục lục

1 Giới thiệu	1
1.1 Tổng quát	1
1.2 Cấu trúc Zigbee2MQTT	1
1.3 Mục tiêu	2
2 Các công cụ cần thiết	3
2.1 Phần cứng	3
2.1.1 Module CC2530	3
2.1.2 Mạch nạp SmartRF04EB	3
2.1.3 Module USB to TTL CH340 G Cable	4
2.1.4 Máy tính	4
2.1.5 Module Silabs RGB (RD-0098-040)	5
2.1.6 Một số phần cứng khác	5
2.2 Phần mềm	6
2.2.1 Thư mục chính của Zigbee2mqtt	6
2.2.2 Mosquitto	6
2.2.3 Node.js	7
2.2.4 SmartRF Flash Programmer	7
2.2.5 MQTT Explorer	8
2.2.6 Một số môi trường cần thiết	8
3 Thiết lập gateway Zigbee2MQTT	8
3.1 Nạp code cho 2 module CC2530	8
3.1.1 Module CC2530 làm Coordinator	8
3.1.2 Module CC2530 làm Router	9
3.2 Tạo MQTT Broker bằng phần mềm Mosquitto	9
3.3 Chạy Zigbee2MQTT trên window	10
3.3.1 Xác định cổng COM được chỉ định cho thiết bị của bạn .	10
3.3.2 Cài đặt	11
3.3.3 Cấu hình và khởi động Zigbee2MQTT	12
4 Cho phép các thiết bị tham gia vào mạng Zigbee	13
4.1 Các thiết bị được hỗ trợ trong Zigbee2MQTT	13
4.2 Các thiết bị không được hỗ trợ trong Zigbee2MQTT	15
4.2.1 Tiến hành cho thiết bị tham gia vào mạng	15
4.2.2 Tạo tệp chuyển đổi bên ngoài	16
5 Điều khiển thiết bị bằng Zigbee2MQTT	17
5.1 Sử dụng MQTT Explorer	17
5.1.1 Tạo kết nối trên MQTT Explorer	17



5.1.2	Điều khiển Light RGB	20
5.2	Sử dụng giao diện người dùng trên web	23
5.2.1	Thiết lập	23
6	Kết luận và hướng phát triển	28
6.1	Kết luận	28
6.2	Hướng phát triển	28

1 Giới thiệu

1.1 Tổng quát

Zigbee là tiêu chuẩn khu vực mạng lưới cá nhân 802.15.4 của IEEE được ứng dụng rộng rãi trong các dự án nhà thông minh. Mặc dù thường có giá cao hơn, các thiết bị Zigbee, đặc biệt là cảm biến, ngày càng phổ biến do những ưu thế riêng của Zigbee. Các thiết bị này rất phong phú về chủng loại như cảm biến, công tắc hay đèn thông minh với rất nhiều nhà sản xuất như Xiaomi, Tuya hoặc Philips.

Để tích hợp các thiết bị Zigbee vào Home Assistant, thông thường có 2 cách: tích hợp thông qua Hub của hãng (như Aqara hoặc IKEA TRÅDFRI hub) và tích hợp trực tiếp thông qua các universal hub như deCONZ hoặc Zigbee2MQTT. Khi sử dụng các Hub của hãng, ta chỉ có thể điều khiển thiết bị thông qua Hub của hãng thiết bị đó. Nghĩa là thiết bị và hub sẽ không thể kết nối và giao tiếp với nhau nếu chúng không cùng nhà sản xuất. Đây là một hạn chế khá lớn. Để giải quyết điều này, các nhà phát triển đã đưa ra một giải pháp mang tên Zigbee2MQTT.

Zigbee2MQTT cho phép sử dụng các thiết bị Zigbee mà không cần thông qua các cầu nối (bridge) hoặc cổng (gateway) của nhà sản xuất. Đồng thời nó kết nối các sự kiện và cho phép điều khiển các thiết bị Zigbee thông qua MQTT. Bằng cách này, ta có thể tích hợp các thiết bị Zigbee của mình với bất kỳ cơ sở hạ tầng nhà thông minh nào.

1.2 Cấu trúc Zigbee2MQTT

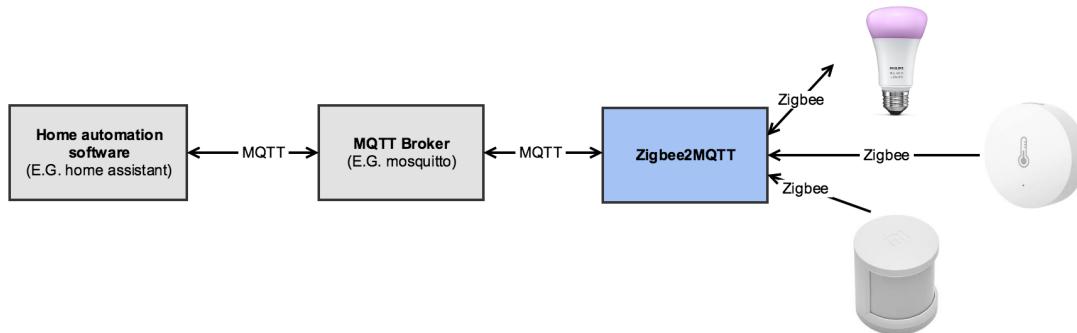
Để có cái nhìn tổng quan và hình dung được cách thức hoạt động của Zigbee2MQTT, đầu tiên ta đi tìm hiểu hai khái niệm Zigbee và MQTT.

Zigbee là tiêu chuẩn khu vực mạng lưới cá nhân **802.15.4** của **IEEE**. Nó được xem là một giải pháp thay thế cho **Wi-Fi** và **Bluetooth** của một số ứng dụng bao gồm các thiết bị sử dụng năng lượng thấp mà không cần nhiều băng thông - như các hệ thống cảm biến trong nhà thông minh.

MQTT (Message Queue Telemetry Transport)¹ là một giao thức mạng kích thước nhỏ (lightweight), hoạt động theo cơ chế publish - subscribe (tạm dịch: xuất bản - đăng ký) theo tiêu chuẩn **ISO (ISO/IEC 20922)** và **OASIS** mở để truyền tin nhắn giữa các thiết bị. Giao thức này hoạt động trên nền tảng **TCP/IP**; tuy nhiên, bất kỳ giao thức mạng nào cung cấp các kết nối theo tuần tự, không mất dữ liệu (lossless), kết nối hai chiều đều có thể hỗ trợ MQTT. MQTT được thiết kế cho các kết nối cho việc truyền tải dữ liệu cho các thiết bị ở xa, các thiết bị hay vi điều khiển nhỏ có tài nguyên hạn chế, hoặc trong

¹MQTT, Wikipedia, vi.wikipedia.org/wiki/MQTT, 10/05/2021

các ứng dụng có băng thông mạng bị hạn chế.



Hình 1: Cấu trúc Zigbee2MQTT

Zigbee2MQTT được tạo thành từ ba modules, mỗi phần được phát triển trong dự án Github của riêng nó. Chúng ta sẽ bắt đầu từ phần cứng là bộ điều hợp (adapter). **Zigbee-herdsman** kết nối với bộ điều hợp (adapter) Zigbee của bạn, cung cấp một API² có sẵn cho các cấp cao hơn của ngăn xếp. Ví dụ phần cứng của Texas Instruments, **zigbee-herdsman** sử dụng **API TI zStack monitoring and test API** để giao tiếp với bộ điều hợp. **Zigbee-herdsman** xử lý giao tiếp Zigbee cốt lõi. Tiếp đến, modules **zigbee-herdsman-converter** xử lý ánh xạ từ các mô hình thiết bị riêng lẻ đến các cụm Zigbee mà chúng hỗ trợ. Các cụm Zigbee là các lớp của giao thức Zigbee nằm trên giao thức cơ sở xác định cách các đèn, cảm biến và công tắc nói chuyện với nhau qua mạng Zigbee. Cuối cùng, Zigbee2MQTT điều khiển **zigbee-herdsman** và ánh xạ các thông báo zigbee thành các thông báo MQTT. **Zigbee2MQTT** cũng theo dõi trạng thái của hệ thống. Nó sử dụng tệp database.db để lưu trữ trạng thái này; tệp văn bản với cơ sở dữ liệu JSON³ của các thiết bị được kết nối và khả năng của chúng.

1.3 Mục tiêu

Xây dựng mạng zigbee. Thiết kế gateway chuyển đổi mạng zigbee sang MQTT trên máy tính với hệ điều hành Window. Điều khiển các thiết bị zigbee thông qua giao thức MQTT.

²API(Application Programming Interface) là các phương thức, giao thức kết nối với các thư viện và ứng dụng khác. API cung cấp khả năng cung cấp khả năng truy xuất đến một tập các hàm hay dùng. Và từ đó có thể trao đổi dữ liệu giữa các ứng dụng.

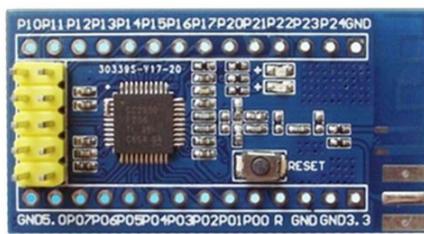
³JSON (JavaScript Object Notation) là một kiểu dữ liệu mở trong JavaScript. Kiểu dữ liệu này bao gồm chủ yếu là text, có thể đọc được theo dạng cặp "thuộc tính - giá trị

2 Các công cụ cần thiết

2.1 Phần cứng

2.1.1 Module CC2530

Trong bài này chúng ta cần hai module CC2530. Một module dùng làm coordinator. Module còn lại dùng vừa là một node mạng vừa là router cho mạng Zigbee.



Hình 2.1.1: Module CC2530

2.1.2 Mạch nạp SmartRF04EB

Mạch nạp SmartRF04EB dùng để nạp chương trình cho hai module CC2530 ở trên.



Hình 2.1.2: Mạch nạp SmartRF04EB

2.1.3 Module USB to TTL CH340 G Cable

Module USB to TTL CH340 G Cable dùng để kết nối mạch nạp với máy tính.



Hình 2.1.3: Module USB to TTL CH340 G Cable

2.1.4 Máy tính

Máy tính dùng để chạy phần mềm của Zigbee2MQTT. Có thể sử dụng Raspberry Pi để tiết kiệm năng lượng hơn. Trong bài này em chọn máy tính xách

tay chạy dưới hệ điều hành window để làm.



Hình 2.1.4: Raspberry Pi

2.1.5 Module Silabs RGB (RD-0098-040)

Module Silabs RGB từ nhà sản xuất Silicon Labs. Đây là một module zigbee chuẩn. Ta có thể điều khiển On/Off, độ sáng tối hoặc đổi màu của bóng đèn trên kit.



Hình 2.1.5: Module Silabs RGB

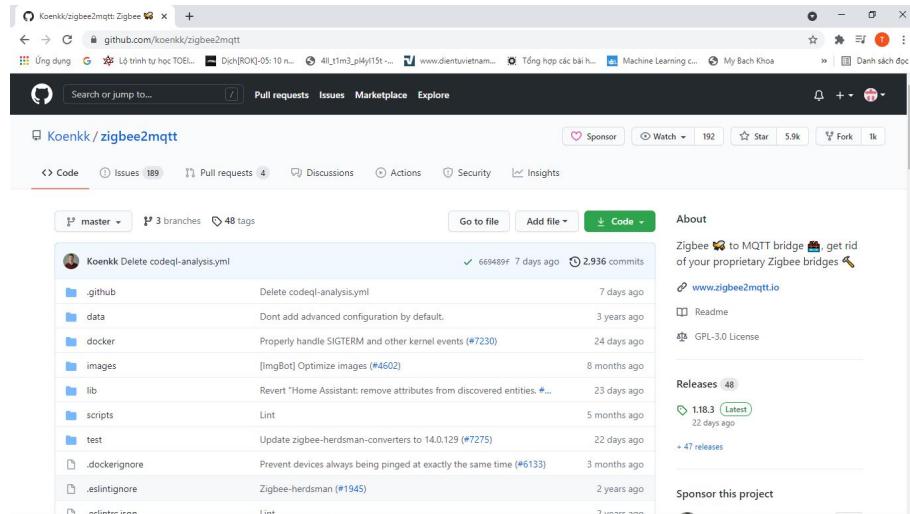
2.1.6 Một số phần cứng khác

Ngoài những thứ trên, chúng ta cần những phần cứng khác như dây nối, bộ nguồn cho các module CC2530 và Silabs RGB.

2.2 Phần mềm

2.2.1 Thư mục chính của Zigbee2mqtt

Được phát triển bởi KoenKK. Ta có thể tải thư mục về từ github. Đường dẫn tới thư mục: github.com/koenkk/Zigbee2MQTT.



Hình 2.2.1: Thư mục Zigbee2MQTT

2.2.2 Mosquitto

Mosquitto là một MQTT Broker mã nguồn mở cho phép thiết bị truyền nhận dữ liệu theo giao thức MQTT versions 3.1 và 3.1.1. Một giao thức nhanh, nhẹ theo mô hình publish/subscribe được sử dụng rất nhiều trong lĩnh vực Internet of Things.



Hình 2.2.2: Phần mềm Mosquitto

2.2.3 Node.js

Node.js là một JavaScript runtime được build dựa trên engine JavaScript V8 của Chrome. Node.js sử dụng kiến trúc hướng sự kiện event-driven, mô hình non-blocking I/O làm cho nó nhẹ và hiệu quả hơn. Hệ thống nén của Node.js, npm⁴, là hệ thống thư viện nguồn mở lớn nhất thế giới. Ta dùng Node.js để giải nén những thư viện cần thiết cho chương trình Zigbee2MQTT.



Hình 2.2.3: Phần mềm node.js

2.2.4 SmartRF Flash Programmer

Phần mềm hỗ trợ nạp code cho 2 module CC2530 với mạch nạp là SmartRF04EB.

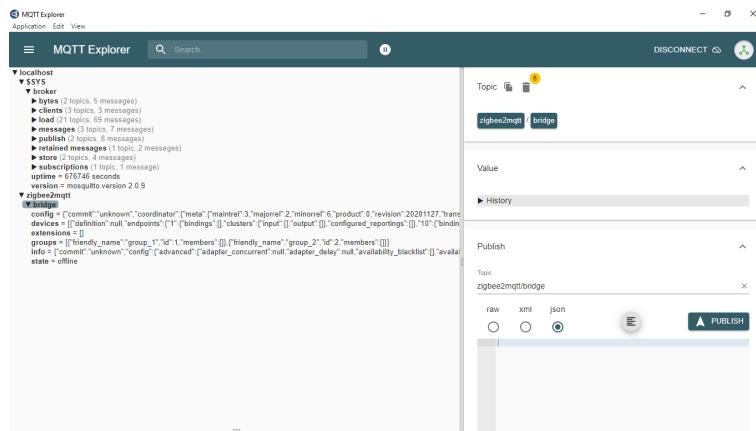


Hình 2.2.4: Phần mềm SmartRF Flash Programmer

⁴NPM là viết tắt của Node package manager là một công cụ tạo và quản lý các thư viện lập trình Javascript cho Node.js. Ta dùng công cụ npm trong bài này để quản lý và cập nhật các thư viện cần thiết cho chương trình.

2.2.5 MQTT Explorer

MQTT Explorer là một ứng dụng MQTT client toàn diện, cung cấp tổng quan các cấu trúc về các chủ đề MQTT của bạn. MQTT Explorer giúp các tác vụ với các thiết bị/dịch vụ trên máy chủ trở nên đơn giản hơn.



Hình 2.2.5: Phần mềm MQTT Explorer

2.2.6 Một số môi trường cần thiết

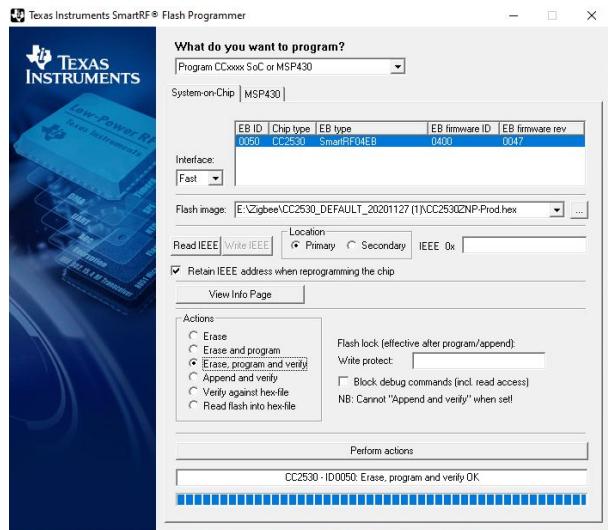
- Driver cho mạch chuyển đổi USB to TTL CH340.
- Java .Bài này ta dùng Java phiên bản 8.0.
- Java Development Kit (JDK).Bài này ta dùng JDK phiên bản 8.0.
- Python. Bài này ta dùng Python phiên bản 3.8.0.
- Visual Studio Build Tools.Bài này ta dùng Visual Studio Build Tools phiên bản 2019.

3 Thiết lập gateway Zigbee2MQTT

3.1 Nạp code cho 2 module CC2530

3.1.1 Module CC2530 làm Coordinator

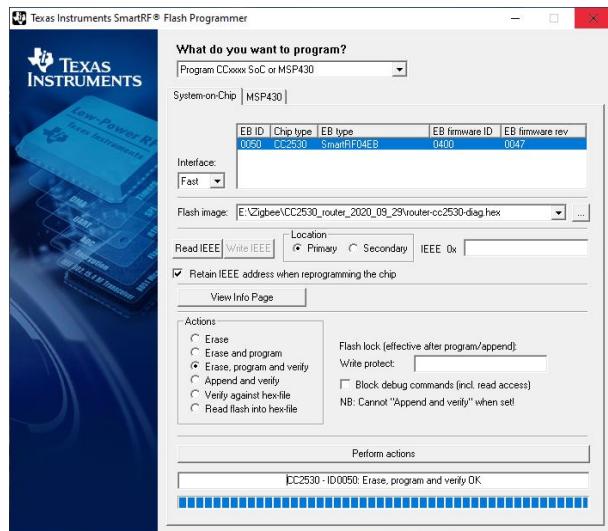
Đầu tiên ta tải code của Module CC2530 làm Coordinator theo đường dẫn sau: <https://bit.ly/34FXnl3>. Kết nối kit với mạch nạp SmartRF04EB. Chọn file code **CC2530ZNP-Prod.hex** để nạp vào kit.



Hình 3.1.1: Nạp code cho module CC2530 làm cordinator

3.1.2 Module CC2530 làm Router

Dầu tiên ta tải code của Module CC2530 làm router theo đường dẫn sau: <https://bit.ly/3fIzesX>. Kết nối kit với mạch nạp SmartRF04EB. Chọn file code **router-cc2530-diag.hex** để nạp vào kit.

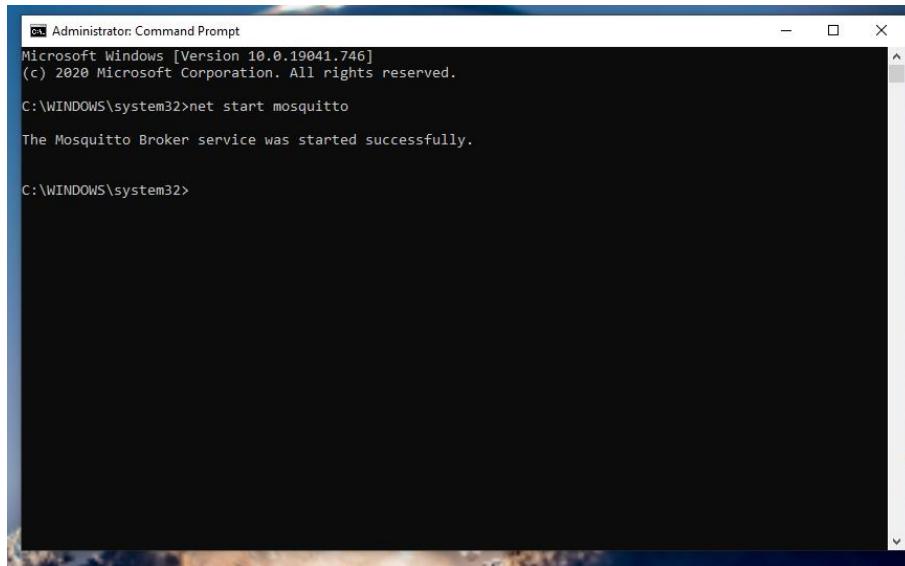


Hình 3.1.2: Nạp code cho module CC2530 làm router

3.2 Tạo MQTT Broker bằng phần mềm Mosquitto

Trước khi bắt đầu chạy Zigbe2MQTT, trên hệ thống phải được cài đặt MQTT broker. Có rất nhiều phần mềm để tạo MQTT broker, trong bài này ta sử dụng phần mềm Mosquitto như đã đề cập ở mục **2.2.2**.

Để cài đặt MQTT broker, đầu tiên ta mở cửa sổ **Command Prompt** với lệnh **Run as administrator**. Tiếp theo nhập lệnh **net start mosquitto**. Như vậy MQTT broker đã được tạo thành công.



Hình 3.2: MQTT broker được tạo thành công

3.3 Chạy Zigbee2MQTT trên window

3.3.1 Xác định cổng COM được chỉ định cho thiết bị của bạn

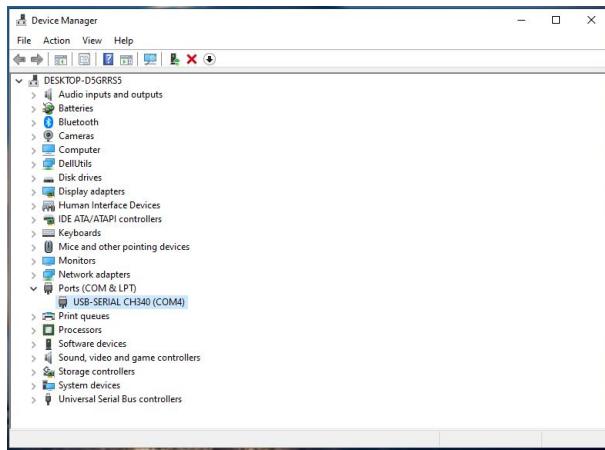
Để kết nối module CC2530 vào máy tính, ta sử dụng module USB to TTL 340G. Thứ tự nối dây như sau:

- Chân VCC của moudlue CC2530 nối với chân VCC của module USB to TTL 340G.
- Chân GND của moudlue CC2530 nối với chân GND của module USB to TTL 340G.
- Chân TXD (P03) của moudlue CC2530 nối với chân RXD của module USB to TTL 340G.
- Chân RXD (P02) của moudlue CC2530 nối với chân TXD của module USB to TTL 340G.

Sau khi thực hiện nối module CC2530 với module USB to TTL CH340 G. Kết nối bộ thiết bị trên vào máy tính qua cổng USB.

Mở menu **Start** và bắt đầu nhập **Device Manager**.

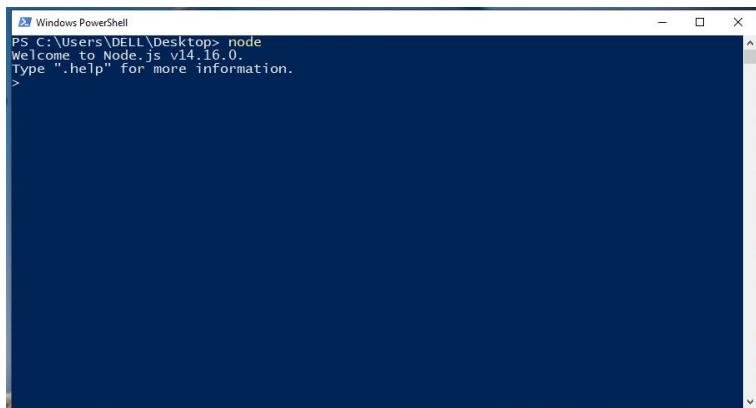
Mở rộng **Ports(COM & LPT)**. Chọn **USB-Serial CH340 (COM4)**. Ghi lại số cổng (**COM4**) , nó sẽ cần thiết trong quá trình cấu hình.



Hình 3.2.1: Xác định cổng COM được kết nối với máy tính của thiết bị

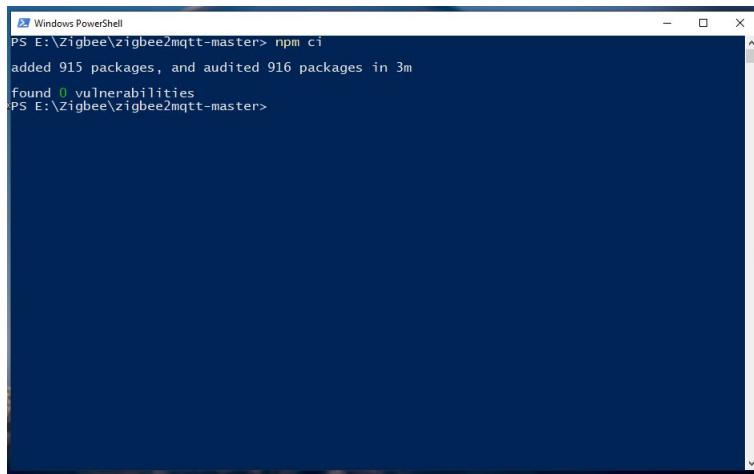
3.3.2 Cài đặt

Sau khi cài đặt chương trình node.js như đề cập ở phần **2.2.3.** Kiểm tra việc cài đặt của bạn bằng cách mở **Command prompt** hoặc **Powershell** từ **Start menu**. Nhập lệnh **node**.



Hình 3.2.2a: Chương trình node.js đã được cài đặt thành công

Di chuyển đến thư mục chứa Zigbee2MQTT. Mở **Powershell** tại thư mục này. Chạy câu lệnh **npm ci** để cài đặt các gói cần thiết cho Zigbee2MQTT.



Hình 3.2.2b: Sử dụng công cụ **npm** cài đặt các gói cần thiết cho Zigbee2MQTT

3.3.3 Cấu hình và khởi động Zigbee2MQTT

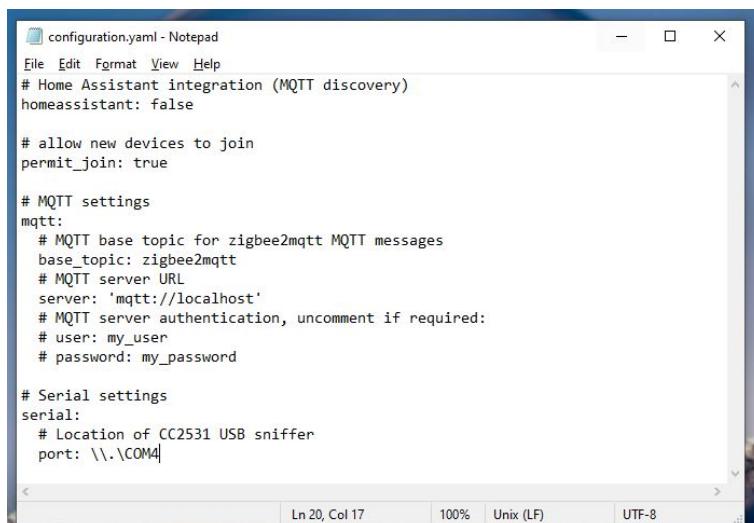
Trong thư mục Zigbee2MQTT, mở thư mục data\configuration.yaml ở dạng text.

Thay đổi cấu hình cổng nối tiếp để phù hợp với thiết lập của bạn.

serial:

port: \\.\COM4

Lưu thay đổi và thoát.



Hình 3.2.3a: Thư mục cấu hình cho Zigbee2MQTT

Để khởi động Zigbee2MQTT, ta di chuyển đến thư mục Zigbee2MQTT . Mở **Powershell** tại thư mục này. Khởi động Zigbee2MQTT bằng câu lệnh **npm**

start .

Để dừng Zigbee2MQTT, ta nhấn tổ hợp lệnh **Ctrl C**.

```
PS E:\Zigbee\zigbee2mqtt-master> npm start
> zigbee2mqtt@1.18.1 start
> node index.js

Zigbee2MQTT:info 2021-05-13 15:40:25: Logging to console and directory: 'E:\Zigbee\zigbee2mqtt-maste
r\data\log\2021-05-13_15-40-21' filename: 'log.txt'
Zigbee2MQTT:info 2021-05-13 15:40:25: Starting Zigbee2MQTT version 1.18.1 (commit #unknown)
Zigbee2MQTT:info 2021-05-13 15:40:25: Starting zigbee-herdsman (0.13.7!1)
Zigbee2MQTT:info 2021-05-13 15:40:32: zigbee-herdsman started
Zigbee2MQTT:info 2021-05-13 15:40:32: Coordinator firmware version: {"meta":{"mainrel":3,"majorrel":2,"minorrel":6,"product":0,"revision":20201127,"transportrev":2},"type":"zStack12"}
Zigbee2MQTT:info 2021-05-13 15:40:32: Currently 0 devices are joined!
Zigbee2MQTT:warn 2021-05-13 15:40:32: permit_join set to true in configuration.yaml.
Zigbee2MQTT:warn 2021-05-13 15:40:32: Allowing new devices to join.
Zigbee2MQTT:info 2021-05-13 15:40:32: Set permit_join to false once you joined all devices.
Zigbee2MQTT:info 2021-05-13 15:40:32: Zigbee: allowing new devices to join.
Zigbee2MQTT:info 2021-05-13 15:40:33: Connecting to MQTT server at mqtt://localhost
Zigbee2MQTT:info 2021-05-13 15:40:33: Connected to MQTT server
Zigbee2MQTT:info 2021-05-13 15:40:33: MQTT publish: topic 'zigbee2mqtt/bridge/state', payload 'onlin
e'
Zigbee2MQTT:info 2021-05-13 15:40:33: MQTT publish: topic 'zigbee2mqtt/bridge/config', payload '{"co
mmit": "unknown", "coordinator": {"meta": {"mainrel": 3, "majorrel": 2, "minorrel": 6, "product": 0, "revision": 20201127, "transportrev": 2}, "type": "zStack12"}, "log_level": "info", "network": {"channel": 11, "extendedPanID": "0xaaaaaaaaaaaaaaaa", "panID": 6754}, "permit_join": true, "version": "1.18.1"}'
```

Hình 3.2.3b: Zigbee2MQTT được khởi động thành công.

4 Cho phép các thiết bị tham gia vào mạng Zigbee

4.1 Các thiết bị được hỗ trợ trong Zigbee2MQTT

Các thiết bị được hỗ trợ trong Zigbee2MQTT là những thiết bị mà các nhà phát triển đã cấu hình sẵn. Danh sách các thiết bị đã được hỗ trợ có trong đường dẫn sau đây: https://www.zigbee2mqtt.io/information/supported_devices.html. Để có thêm thiết bị mới vào mạng zigbee, trong file **configuration.yaml** phải chỉnh sửa **permit\join: true**. Sau đó khởi động lại Zigbee2MQTT và reset lại thiết bị muốn thêm vào mạng zigbee.

Sau đây chúng ta sẽ thêm module CC2530 với chức năng router vào mạng zigbee của .

```
>Select npm start
> node index.js
Zigbee2MQTT:info 2021-05-13 19:09:36: Logging to console and directory: 'E:\Zigbee\zigbee2mqtt-master\data\log\2021-05-13.19-09-35' filename: log.txt
Zigbee2MQTT:info 2021-05-13 19:09:36: Starting Zigbee2MQTT version 1.18.1 (commit #unknown)
Zigbee2MQTT:info 2021-05-13 19:09:36: Starting zigbee-herdsman (0.13.71)
Zigbee2MQTT:info 2021-05-13 19:09:38: zigbee-herdsman started
Zigbee2MQTT:info 2021-05-13 19:09:38: Coordinator firmware version: {"meta":{"mainrel":3,"majorrel":2,"minorrel":6,"product":0,"revision":20201127,"transportrev":2}, "type": "zStack12"}
Zigbee2MQTT:info 2021-05-13 19:09:38: Currently 0 devices are joined.
Zigbee2MQTT:warn 2021-05-13 19:09:38: 'permit_join' set to 'true' in configuration.yaml.
Zigbee2MQTT:warn 2021-05-13 19:09:38: Allow new devices to join.
Zigbee2MQTT:warn 2021-05-13 19:09:38: Set permit join to false once you joined all devices.
Zigbee2MQTT:info 2021-05-13 19:09:38: Zigbee: allowing new devices to join.
Zigbee2MQTT:info 2021-05-13 19:09:38: Connecting to MQTT server at mqtt://localhost
Zigbee2MQTT:info 2021-05-13 19:09:38: Connected to MQTT server.
Zigbee2MQTT:info 2021-05-13 19:09:38: MQTT publish: topic 'zigbee2mqtt/bridge/state', payload 'online'
Zigbee2MQTT:info 2021-05-13 19:09:38: MQTT publish: topic 'zigbee2mqtt/bridge/config', payload '{"commit":"unknown","coordinator":{"meta":{"mainrel":3,"majorrel":6,"minorrel":6,"product":0,"revision":20201127,"transportrev":2}, "type": "zStack12"}, "log_level": "info", "network": {"channel": 11, "extendPanID": "0xdccccccccccc"}, "panID": "6754"}, "permit_join": true, "version": "1.18.1"}'
Zigbee2MQTT:info 2021-05-13 19:09:47: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload '{"data": {"friendly_name": "0x00124b0010010545", "ieee_address": "0x00124b0010010545"}, "type": "device_joined"}'
Zigbee2MQTT:info 2021-05-13 19:09:47: Starting interview of '0x00124b0010010545'
Zigbee2MQTT:info 2021-05-13 19:09:47: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload {"data": {"friendly_name": "0x00124b0010010545", "ieee_address": "0x00124b0010010545"}, "status": "started", "type": "device_interview"}
Zigbee2MQTT:info 2021-05-13 19:09:47: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload {"message": {"friendly_name": "0x00124b0010010545", "type": "device_connected"}}
Zigbee2MQTT:info 2021-05-13 19:09:47: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload {"message": "interview_started", "meta": {"friendly_name": "0x00124b0010010545"}, "type": "pairing"}
Zigbee2MQTT:info 2021-05-13 19:09:47: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload {"message": "interview_started", "meta": {"friendly_name": "0x00124b0010010545", "ieee_address": "0x00124b0010010545"}, "type": "device_announced"}
Zigbee2MQTT:info 2021-05-13 19:09:48: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload {"message": "announce", "meta": {"friendly_name": "0x00124b0010010545"}, "type": "device_announced"}
Zigbee2MQTT:info 2021-05-13 19:09:49: Successfully interviewed '0x00124b0010010545', device has successfully been paired
Zigbee2MQTT:info 2021-05-13 19:09:49: Device '0x00124b0010010545' is supported, identified as: Custom devices (DiY) [CC2530 router](http://ptvo.info/cc2530-based-zigbee-coordinator-and-router-112/) (CC2530.ROUTER)
Zigbee2MQTT:info 2021-05-13 19:09:49: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload {"data": {"definition": {"description": "CC2530 router (http://ptvo.info/cc2530-based-zigbee-coordinator-and-router-112/)"}, "exposes": [{"access": 1, "name": "led", "property": "led", "type": "binary", "value_on": "off", "value_off": "on"}, {"access": 1, "name": "linkquality", "property": "linkquality", "type": "numeric", "unit": "percent"}], "type": "description"}'}
```

Hình 3.3.1: Module CC2530 thành công tham gia vào mạng.

Ban đầu thông báo **Zigbee2MQTT:info 2021-05-13 19:09:38: Currently 0 devices are joined:** nghĩa là chưa có thiết bị nào tham gia vào mạng Zigbee. Sau khi reset module CC2530 thì thông báo **Zigbee2MQTT:info 2021-05-13 19:09:47: Device '0x00124b0010010545' joined** nghĩa là thiết bị của ta đang bắt đầu tham gia vào mạng.

Zigbee2MQTT sẽ bắt đầu phỏng vấn thiết bị. Điều này có thể thấy qua thông báo **Zigbee2MQTT:info 2021-05-13 19:09:47: Starting interview of '0x00124b0010010545'**.

Sau quá trình phỏng vấn thiết bị mới xuất hiện thông báo **Zigbee2MQTT:info 2021-05-13 19:09:49: Successfully interviewed '0x00124b0010010545', device has successfully been paired** nghĩa là quá trình phỏng vấn thành công, đồng thời thiết bị đã thành công ghép nối vào mạng.

Zigbee2MQTT cũng cho biết loại thiết bị của chúng ta là gì và có được hỗ trợ ghép nối vào mạng không qua thông báo **Zigbee2MQTT:info 2021-05-13 19:09:49: Device '0x00124b0010010545' is supported, identified as: Custom devices (DiY) [CC2530 router](http://ptvo.info/cc2530-based-zigbee-coordinator-and-router-112/) (CC2530.ROUTER)**.

Sau khi ghép nối thành công vào mạng thì Zigbee2MQTT liên tục thông báo tình trạng của thiết bị trong mạng. Có thể thấy qua các thông báo như **Zigbee2MQTT:info 2021-05-13 19:10:47: MQTT publish: topic 'Zigbee2MQTT/0x00124b0010010545', payload {"description": "0/0x00124b001001081D", "led": true, "linkquality": 52, "rss": 47, "type": "COORD"}**.

4.2 Các thiết bị không được hỗ trợ trong Zigbee2MQTT

4.2.1 Tiến hành cho thiết bị tham gia vào mạng

Các thiết bị không được hỗ trợ trong Zigbee2MQTT là những thiết bị chưa được cấu hình chuyển đổi trong Zigbee2MQTT. Ta không thể điều khiển hay đọc trạng thái từ thiết bị này nếu chưa có thư mục chuyển đổi. Để thêm chúng vào mạng zigbee thì người dùng phải tự cấu hình và lập trình cho thiết bị đó theo các thư viện của Zigbee2MQTT.

Sau đây ta sẽ cho Module Silabs RGB là một thiết bị không được hỗ trợ tham gia vào Zigbee2MQTT. Trước tiên, trong file **configuration.yaml** ta cũng phải chỉnh sửa **permit_join: true**. Sau đó khởi động lại Zigbee2MQTT và cho phép Module Silabs RGB tham gia mạng vào zigbee bằng cách nhấn 10 lần nút reset.

```
Windows PowerShell
> node index.js
zigbee2MQTT:info 2021-05-13 19:53:29: Logging to console and directory: 'E:\Zigbee\zigbee2mqtt-master\data\log\2021-05-13.19-53-28' filename: log.txt
zigbee2MQTT:info 2021-05-13 19:53:29: Starting Zigbee2MQTT version 1.18.1 (commit #unknown)
zigbee2MQTT:info 2021-05-13 19:53:29: Starting zigbee-herdsman (0.13.7)
zigbee2MQTT:info 2021-05-13 19:53:31: zigbee-herdsman started
zigbee2MQTT:info 2021-05-13 19:53:31: Coordinator firmware version: '['meta': {'maintrel': 3, 'majorrel': 2, 'minorrel': 6, 'product': 0, 'revision': 20201127}, 'transportrev': 2, 'type': 'zStack12'}'
zigbee2MQTT:info 2021-05-13 19:53:31: Currently 1 devices are joined:
zigbee2MQTT:info 2021-05-13 19:53:31: 0x00124b0010010545: CC2530.ROUTER - Custom devices (DiY) [CC2530 router]([http://ptvo.info/cc2530-based-zigbee-coordinator-and-router-112/])
zigbee2MQTT:warn 2021-05-13 19:53:31: permit_join set to 'true' in configuration.yaml.
zigbee2MQTT:warn 2021-05-13 19:53:31: Allowing new devices to join.
zigbee2MQTT:warn 2021-05-13 19:53:31: Set permit_join to 'false' once you joined all devices.
zigbee2MQTT:info 2021-05-13 19:53:31: Zigbee: allowing new devices to join.
zigbee2MQTT:info 2021-05-13 19:53:32: Connecting to MQTT server at mqtt://localhost
zigbee2MQTT:info 2021-05-13 19:53:32: Connected to MQTT server
zigbee2MQTT:info 2021-05-13 19:53:32: MQTT publish: topic 'zigbee2mqtt/bridge/state', payload 'online'
zigbee2MQTT:info 2021-05-13 19:53:32: MQTT publish: topic 'zigbee2mqtt/0x00124b0010010545', payload '{"description": "0x00124b001001081d", "led": true, "rssi": 40, "type": "COORD"}'
zigbee2MQTT:info 2021-05-13 19:53:32: MQTT publish: topic 'zigbee2mqtt/bridge/config', payload '{"commit": "unknown", "coordinator": {"meta": {"maintrel": 3, "majorrel": 2, "minorrel": 6, "product": 0, "revision": 20201127}, "transportrev": 2, "type": "zStack12"}, "log_level": "info", "network": {"channel": 11, "extend_edpanID": "0xdxxxxxxxxxxxxxx", "panID": 67541, "permit_join": true, "version": "1.18.1"}}'
zigbee2MQTT:info 2021-05-13 19:53:37: Device '0x000b57ffe09794d' joined
zigbee2MQTT:info 2021-05-13 19:53:37: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload '{"data": {"friendly_name": "0x000b57ffe09794d", "ieee_address": "0x000b57ffe09794d"}, "status": "joined"}'
zigbee2MQTT:info 2021-05-13 19:53:37: Starting interview of '0x000b57ffe09794d'
zigbee2MQTT:info 2021-05-13 19:53:37: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload '{"data": {"friendly_name": "0x000b57ffe09794d", "ieee_address": "0x000b57ffe09794d"}, "status": "started", "type": "device_interview"}'
zigbee2MQTT:info 2021-05-13 19:53:37: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload '{"message": "friendly_name": "0x000b57ffe09794d", "type": "device_connected"}'
zigbee2MQTT:info 2021-05-13 19:53:37: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload '{"message": "interview_started", "meta": {"friendly_name": "0x000b57ffe09794d"}, "type": "pairing"}'
zigbee2MQTT:info 2021-05-13 19:53:37: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload '{"data": {"friendly_name": "0x000b57ffe09794d", "ieee_address": "0x000b57ffe09794d"}, "status": "device_announce"}'
zigbee2MQTT:info 2021-05-13 19:53:37: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload '{"message": "interview_successful", "meta": {"friendly_name": "0x000b57ffe09794d"}, "type": "device_announced"}'
zigbee2MQTT:info 2021-05-13 19:53:38: Successfully interviewed '0x000b57ffe09794d', device has successfully been paired
zigbee2MQTT:warn 2021-05-13 19:53:38: Device '0x000b57ffe09794d' with Zigbee model 'HaColorControlLight' and manufacturer name 'SiliconLabs' is NOT supported, please follow https://www.zigbee2mqtt.io/how_tos/how_to_support_new_devices.html
zigbee2MQTT:info 2021-05-13 19:53:38: MQTT publish: topic 'zigbee2mqtt/bridge/event', payload '{"data": {"definition": null, "friendly_name": "0x000b57ffe09794d", "ieee_address": "0x000b57ffe09794d"}, "status": "successful", "supported": false, "type": "device_interview"}'
zigbee2MQTT:info 2021-05-13 19:53:38: MQTT publish: topic 'zigbee2mqtt/bridge/log', payload '{"message": "interview_successful", "meta": {"friendly_name": "0x000b57ffe09794d"}, "type": "pairing"}'
zigbee2MQTT:warn 2021-05-13 19:55:54: Received message from unsupported device with Zigbee model 'HaColorControlLight' and manufacturer name 'SiliconLabs'
zigbee2MQTT:warn 2021-05-13 19:58:39: Please see: https://www.zigbee2mqtt.io/how_tos/how_to_support_new_devices.html
zigbee2MQTT:warn 2021-05-13 19:58:39: Received message from unsupported device with Zigbee model 'HaColorControlLight' and manufacturer name 'SiliconLabs'
```

Hình 3.3.2: Module Silabs RGB tham gia vào mạng zigbee.

Tương tự như module CC2530, module Silabs RGB cũng bắt đầu tham gia vào mạng sau khi reset. **Zigbee2MQTT:info 2021-05-13 19:53:37: Device '0x000b57ffe09794d' joined**. Zigbee2MQTT cũng sẽ bắt đầu phỏng vấn thiết bị. Điều này có thể thấy qua thông báo **Zigbee2MQTT:info 2021-05-13 19:53:37: Starting interview of '0x000b57ffe09794d'**

Sau quá trình phỏng vấn thiết bị mới xuất hiện thông báo **Zigbee2MQTT:info 2021-05-13 19:53:38: Successfully interviewed '0x000b57ffe09794d', device has successfully been paired** nghĩa là quá trình phỏng vấn thành công, đồng thời thiết bị đã thành công ghép nối vào mạng.

Tuy nhiên, có cảnh báo mới xuất hiện Zigbee2MQTT:warn 2021-05-13 19:55:54: Received message from unsupported device with Zigbee model 'HaColorControlLight' and manufacturer name 'SiliconLabs' Thông báo này có nghĩa là thiết bị chưa được hỗ trợ.

4.2.2 Tạo tệp chuyển đổi bên ngoài

Đầu tiên vào đường dẫn <https://github.com/Koenkk/zigbee-herdsman-converters> tải thư mục về. Sau đó giải nén thư mục và ghi thư mục theo đường dẫn **zigbee2mqtt-master\ node _ modules**. Tiếp theo là tạo tệp chuyển đổi bên ngoài. Tệp này phải được tạo bên cạnh **configuration.yaml**, trong ví dụ này, chúng ta sẽ gọi nó là **HaColorControlLight.js** (đảm bảo rằng nó kết thúc bằng .js). Trong tệp HaColorControlLight.js có nội dung như sau:

```
E:\Zigbee\zigbee2mqtt-master\data\HaColorControlLight.js - Sublime Text (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
HaColorControlLight.js x
1 const fz = require('zigbee-herdsman-converters/converters/fromZigbee');
2 const tz = require('zigbee-herdsman-converters/converters/toZigbee');
3 const exposes = require('zigbee-herdsman-converters/lib/exposes');
4 const reporting = require('zigbee-herdsman-converters/lib/reporting');
5 const extend = require('zigbee-herdsman-converters/lib/extend');
6 const e = exposes.presets;
7 const ea = exposes.access;
8 const definition = {
9   zigbeeModel: ['HaColorControlLight'], // The model ID from: Device with modelID 'lumi.sens' is not supported.
10  model: 'RD-0098-0401', // Vendor model number, look on the device for a model number
11  vendor: 'SiliconLabs', // Vendor of the device (only used for documentation and startup logging)
12  description: 'ZigBee lighting applications ', // Description of the device, copy from vendor site. (only used for documentation
13  extend: extend.light_onoff_brightness(),
14  extend: extend.light_onoff_brightness_color()
15 };
16
17
18 module.exports = definition;
```

Hình 4.2.2a: HaColorControlLight.js

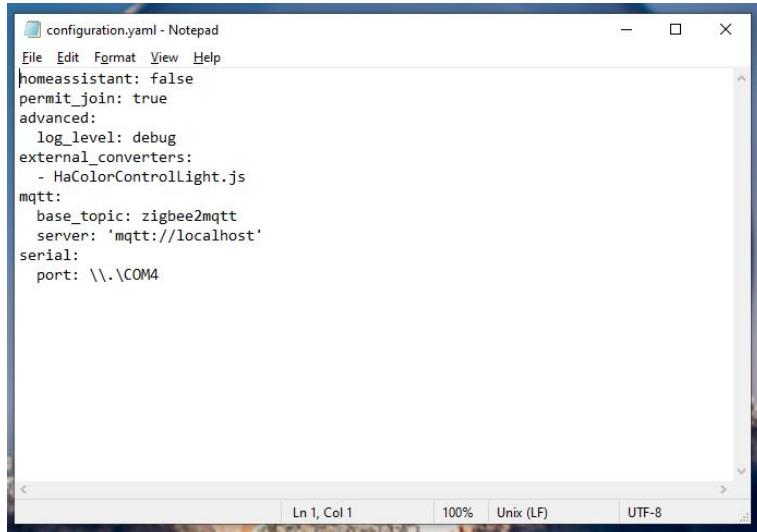
Bây giờ đặt Zigbee2MQTT **log_level** để debug và bật trình chuyển đổi bên ngoài bằng cách thêm phần sau vào Zigbee2MQTT **configuration.yaml**.

advanced:

log_level: debug

external converters:

- HaColorControlLight.js



Hình 4.2.2b: Configuration

Thiết bị đã được cấu hình thành công.

Hình 4.2.2c: Thiết bị được cấu hình thành công

5 Điều khiển thiết bị bằng Zigbee2MQTT

5.1 Sử dụng MQTT Explorer

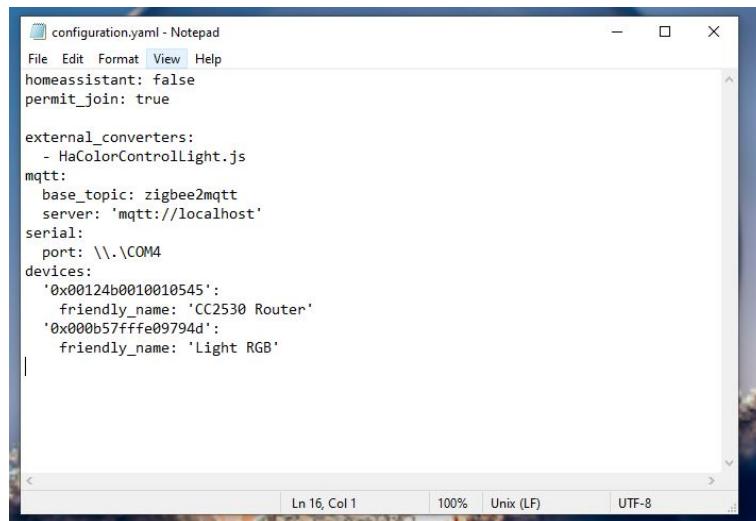
5.1.1 Tao kết nối trên MQTT Explorer

MQTT Explorer là một ứng dụng MQTT client toàn diện, cung cấp tổng quan các cấu trúc về các chủ đề MQTT của bạn. MQTT Explorer giúp các thao tác

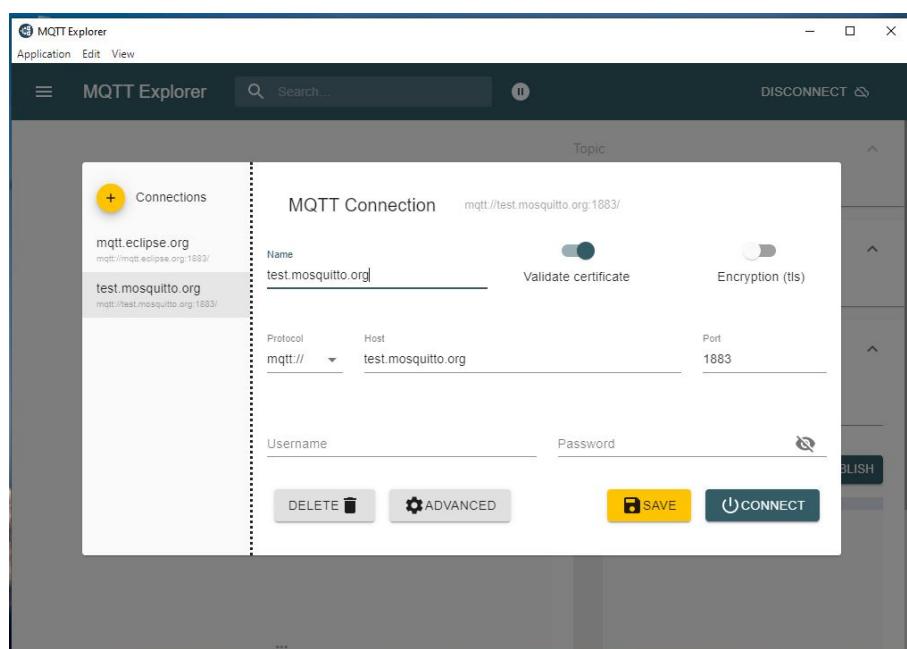
với các thiết bị/dịch vụ trên máy chủ trỏ nên đơn giản hơn.

Sau đây chúng ta sẽ điều khiển thiết module Silabs RGB bằng phần mềm MQTT Explorer. Trước tiên để dễ dàng trong việc nhập lệnh, ta sẽ đổi tên các thiết bị bằng những nhãn trong phần tệp **configuration.yaml** như sau:

Tiếp theo ta mở chương trình MQTT Explorer.



Hình 5.1.1: Đặt tên cho các thiết bị

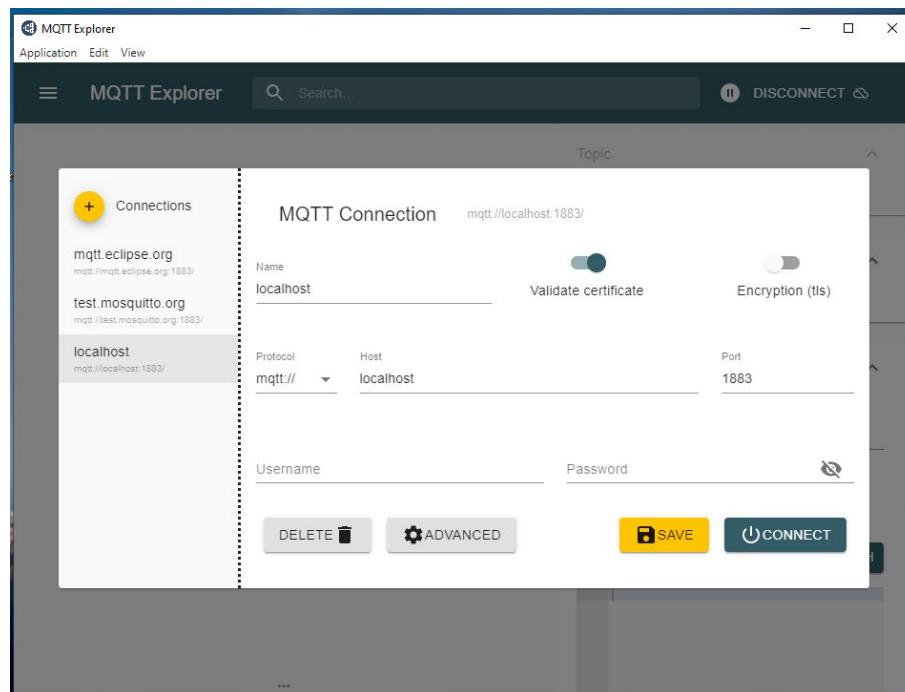


Hình 5.1.2: Chương trình MQTT Explorer

Vì sever của Zigbee2MQTT của chúng ta là "**mqtt://localhost**" (có thể thấy trong tệp **configuration.yaml**), nên ta phải tạo kết nối "**mqtt://localhost**"

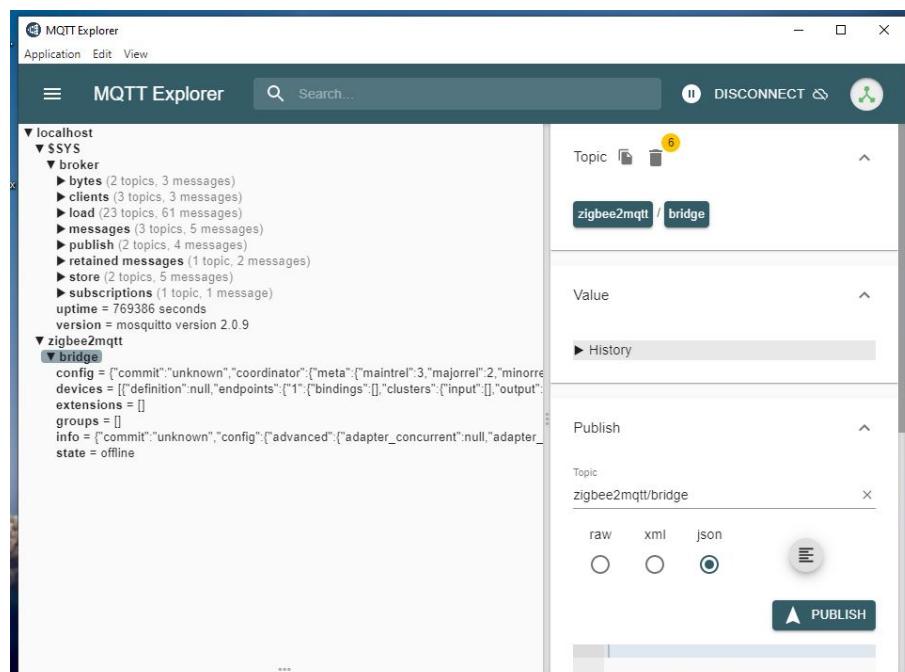
trong chương trình MQTT Explorer. Để tạo kết nối mới, ta nhấp vào **Connections** và nhập nội dung như hình dưới đây.

Sau đó nhấp chuột vào **CONNECT**.



Hình 5.1.3: Tao kết nối cho Zigbee2MQTT

Kết nối được tạo thành công.

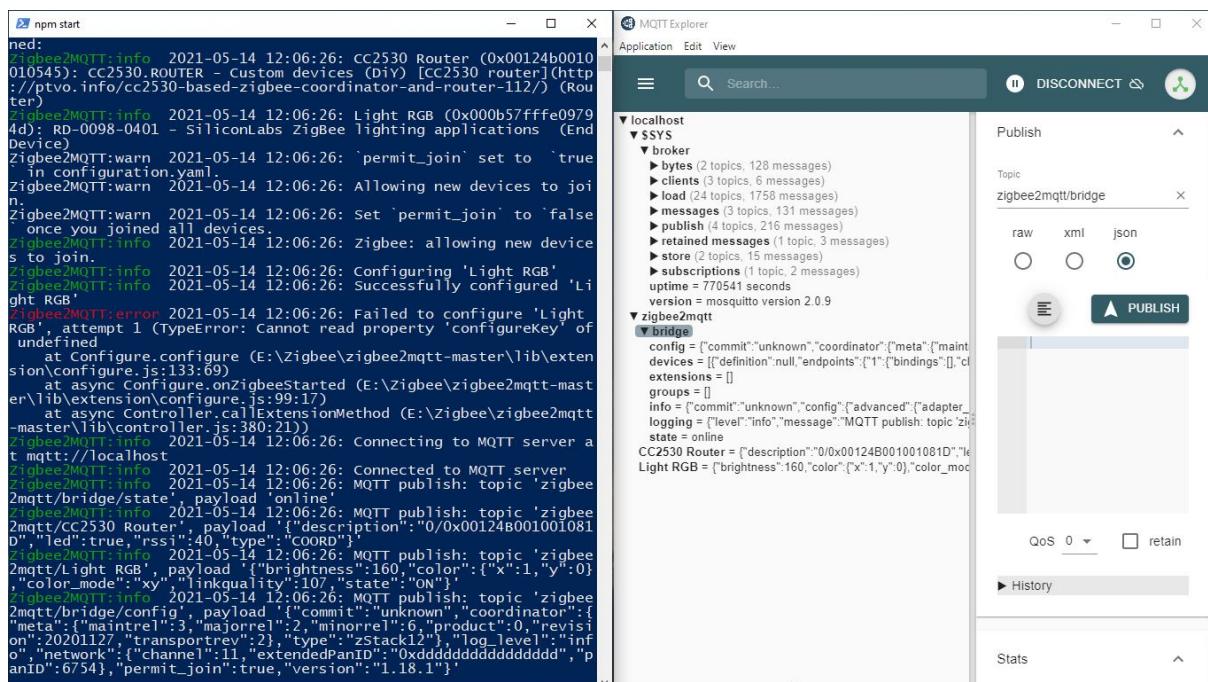


Hình 5.1.4: Localhost được tạo thành công

5.1.2 Điều khiển Light RGB

Thiết bị Light RGB (Module Silabs RGB) là thiết bị chuẩn Zigbee chuẩn. Ta có thể điều khiển các thao tác như bật tắt LED, chỉnh độ sáng của LED, hay đổi màu cho LED.

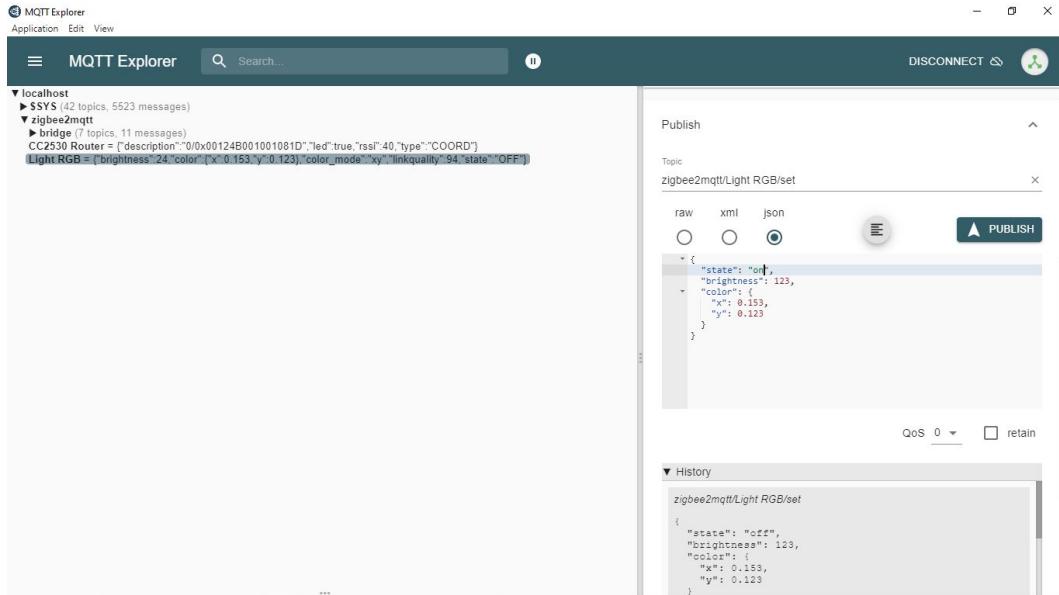
Dầu tiên khởi động Zigbee2MQTT. Di chuyển đến thư mục Zigbee2MQTT . Mở Powershell tại thư mục này. Chạy câu lệnh **npm start** để khởi động Zigbee2MQTT.



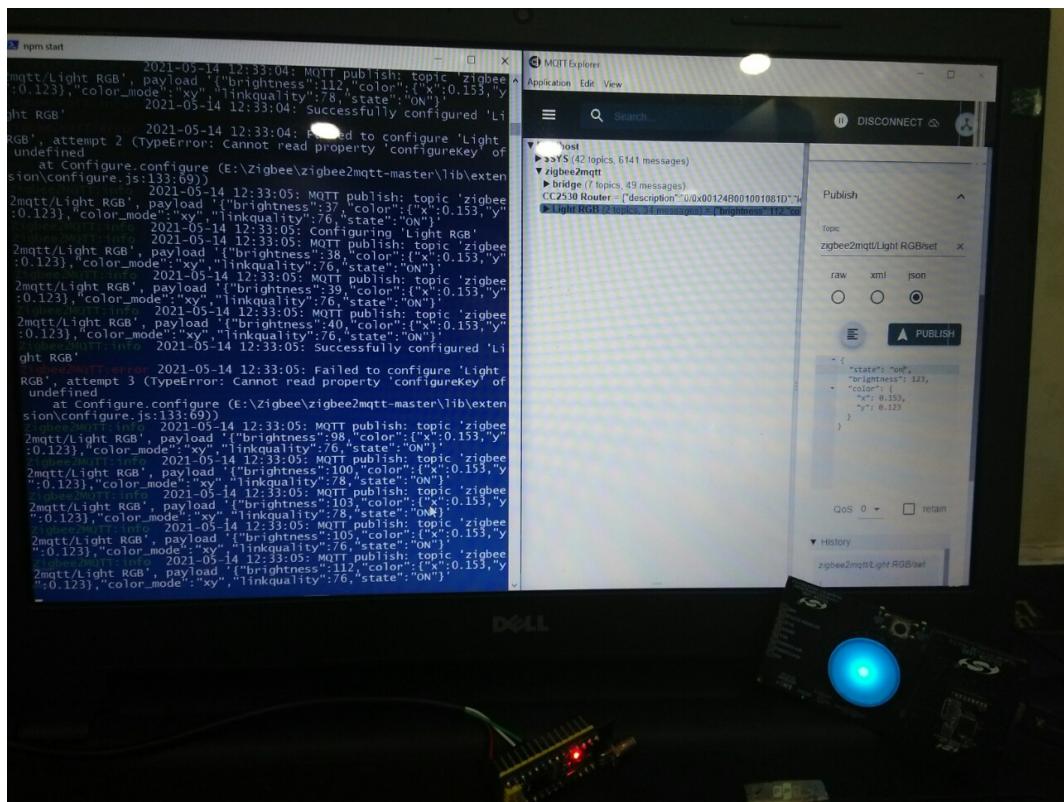
Hình 5.1.5: Khởi động Zigbee2MQTT

Để điều khiển bất kì hoạt động nào của LED, trong phần **Topic** của MQTT Explorer, ta nhập lệnh theo cú pháp **zigbee2mqtt/FRIENDLY_NAME/set** với **FRIENDLY_NAME** là tên thiết bị của bạn.

Sau đó nhập nội dung điều khiển thiết bị của vào cửa sổ commnad. Ta sẽ điều khiển trạng thái LED là bật ("state": "on"), độ sáng LED là 123 ("brightness": 123) và màu của LED theo x và y ("color": "x": 0.153, "y": 0.123). Sau đó nhập chuột vào **PUBLISH** để thực hiện điều khiển.

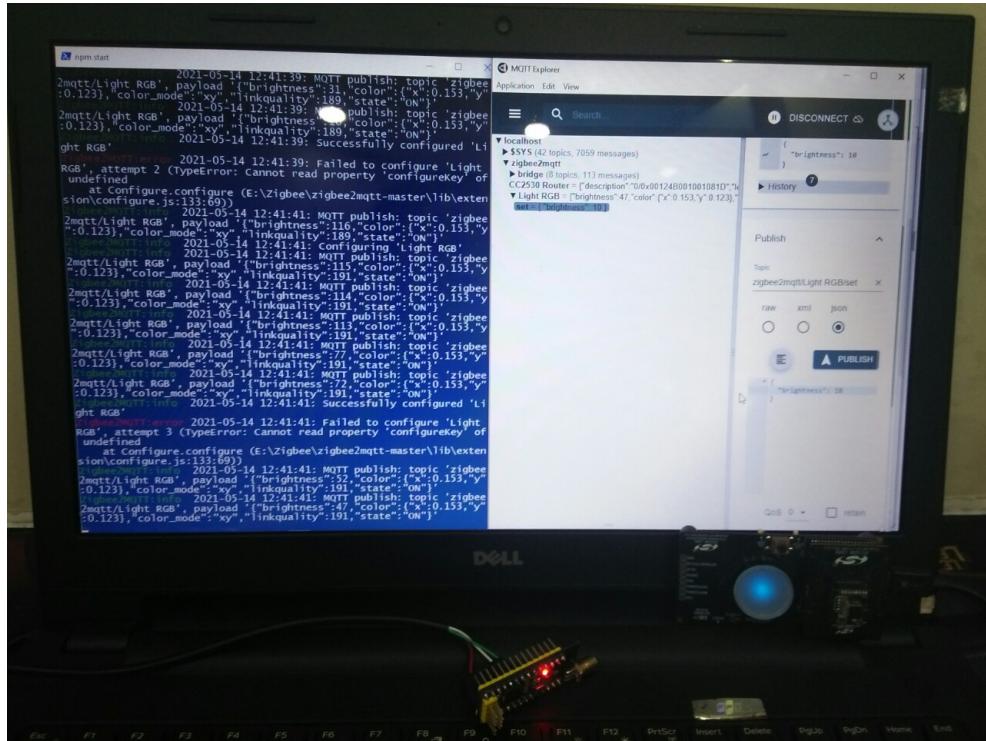


Hình 5.1.6: Điều khiển LED



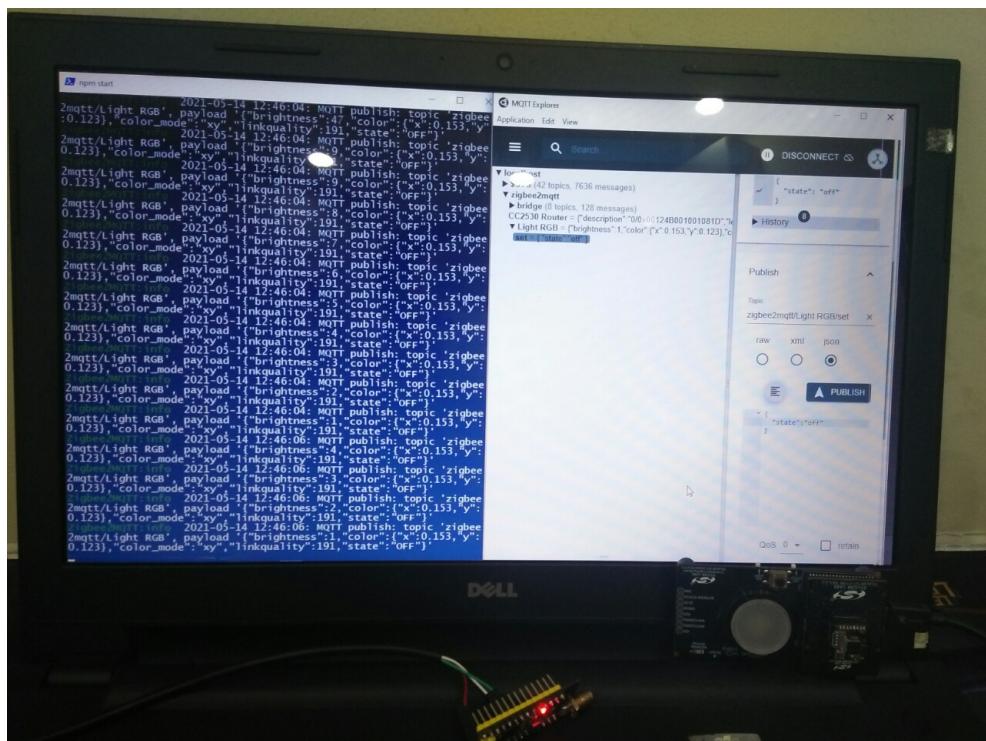
Hình 5.1.7: Kết quả

Có thể thấy đèn được bật và rất sáng. Bây giờ ta sẽ điều khiển độ sáng của LED thấp lại .Nhập nội dung "brightness": 10



Hình 5.1.8: Điều khiển độ sáng LED thấp lại

Điều khiển tắt LED.Nhập nội dung "state": "off"



Hình 5.1.9: Điều khiển LED tắt

5.2 Sử dụng giao diện người dùng trên web

5.2.1 Thiết lập

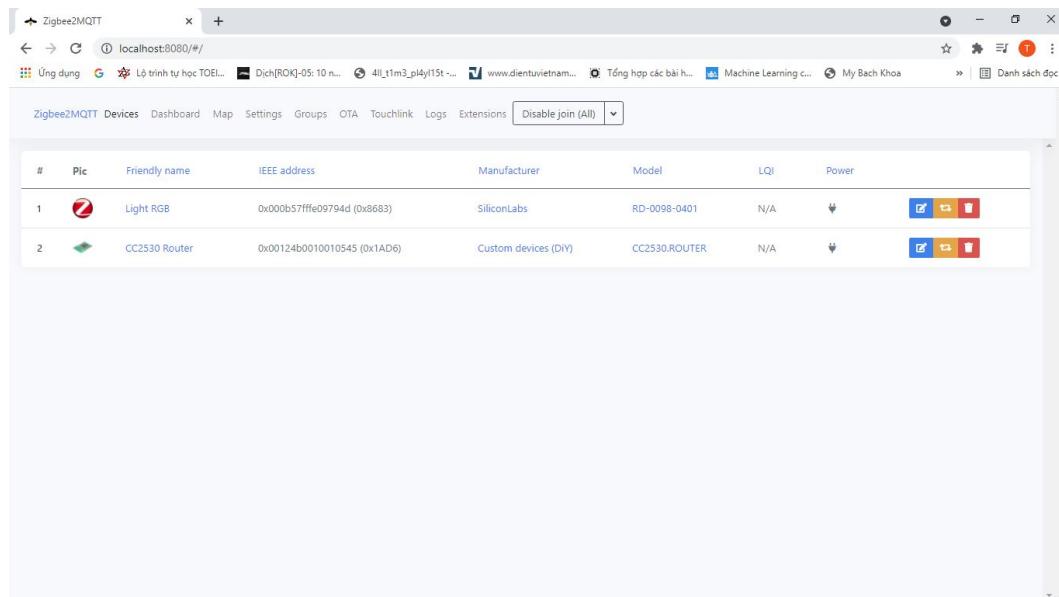
Để điều khiển bằng giao diện người dùng, ta phải chỉnh sửa tệp **configuration.yaml** theo nội dung như sau:

```
*configuration.yaml - Notepad
File Edit Format View Help
homeassistant: false
permit_join: true

external_converters:
- HaColorControlLight.js
mqtt:
base_topic: zigbee2mqtt
server: 'mqtt://localhost'
serial:
port: '\\.\COM4
devices:
'0x00124b0010010545':
friendly_name: 'CC2530 Router'
'0x000b57ffffe09794d':
friendly_name: 'Light RGB'
frontend:
# Optional, default 8080
port: 8080
# Optional, default 0.0.0.0
host: 0.0.0.0
# Optional, enables authentication, disabled by default
auth_token: trungphan|
```

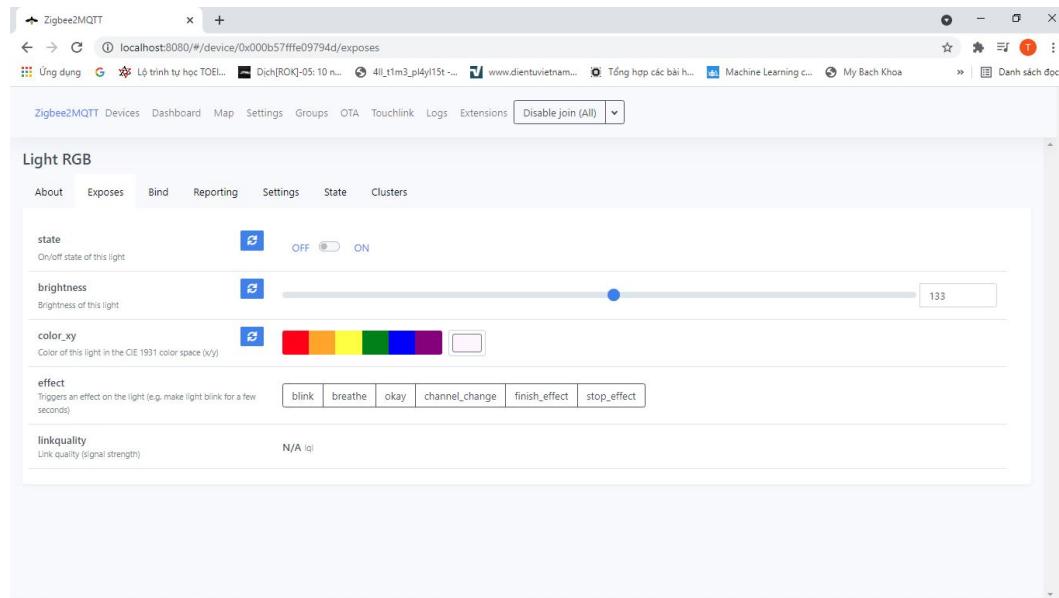
Hình 5.2.1: Cấu hình

Lưu và thoát khỏi tệp. Sau đó khởi động lại Zigbee2MQTT. Sau đó mở trình duyệt của bạn, nhập <http://localhost:8080/> vào thanh tìm kiếm.



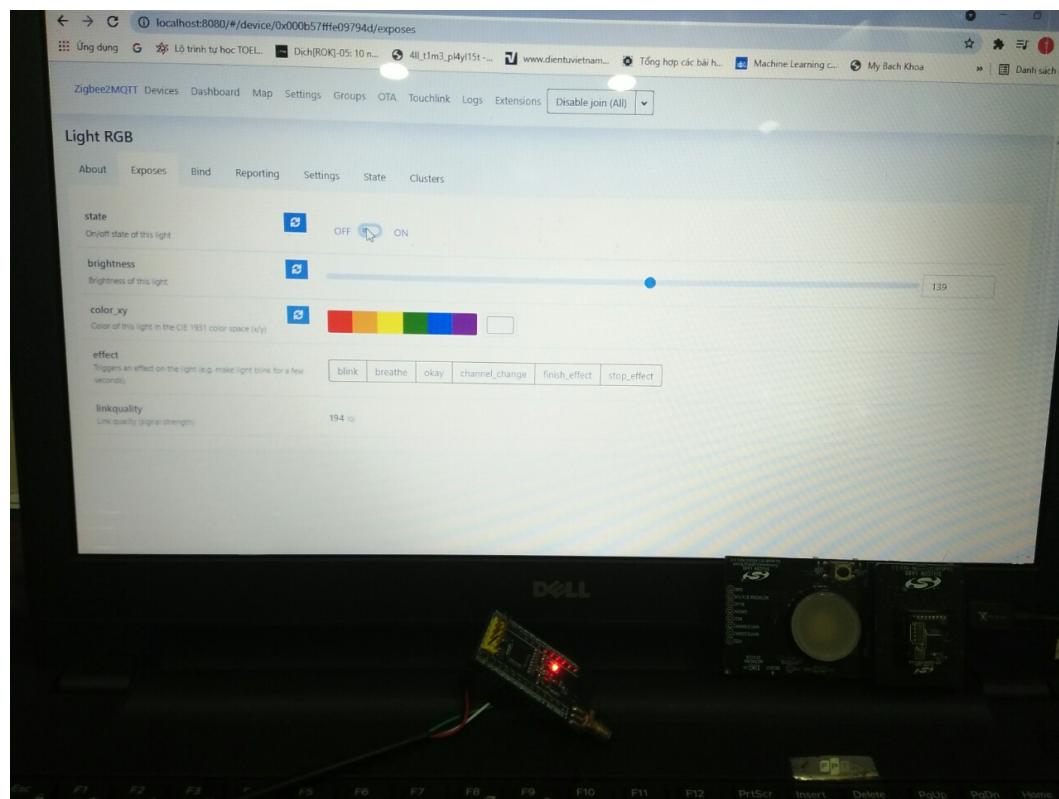
Hình 5.2.2: Giao diện của sever Zigbee2MQTT

Để điều khiển thiết bị Light RGB, nhấp vào tên thiết bị. Vào mục expose



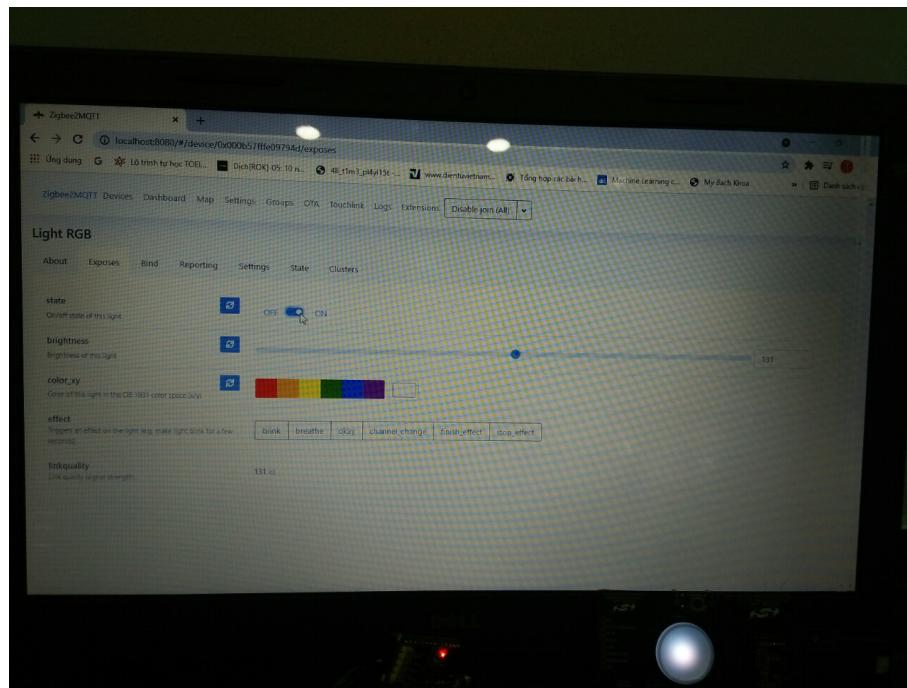
Hình 5.2.3: Giao diện của điều khiển thiết bị

Điều khiển tắt đèn. Tại mục state nhấp chọn off



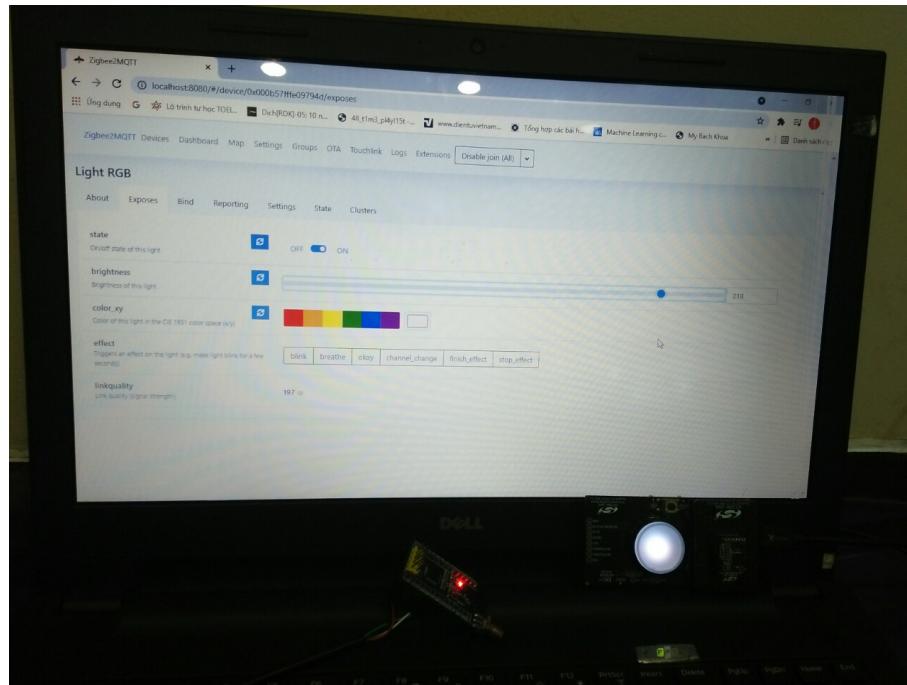
Hình 5.2.4: Đèn tắt.

Điều khiển bật đèn. Tại mục state nhấp chọn on



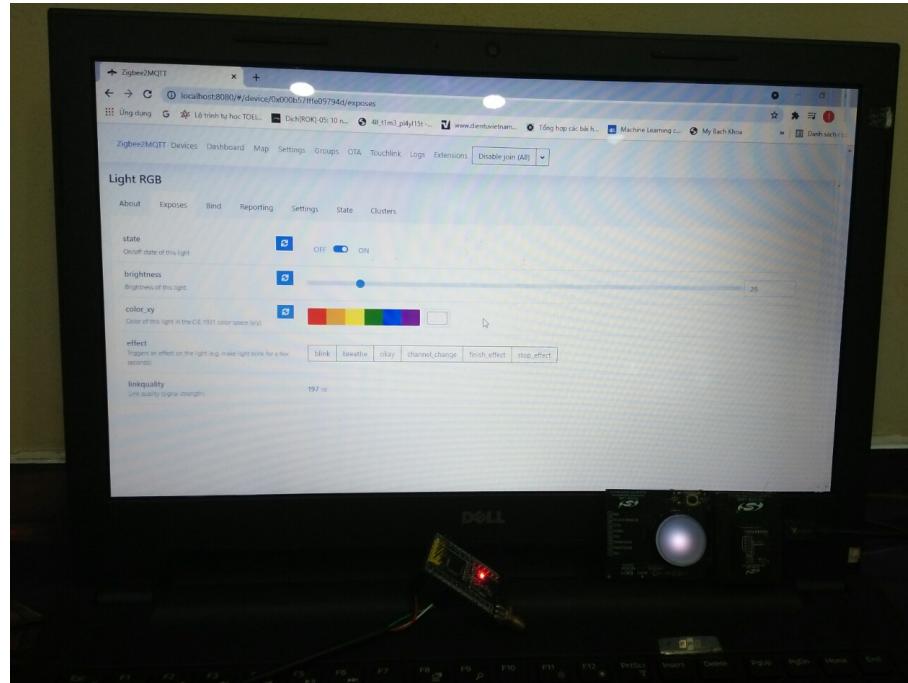
Hình 5.2.5: Đèn được bật.

Để tăng độ sáng đèn. Tại mục brightness kéo con trỏ trên thanh đèn độ sáng cần chỉnh.



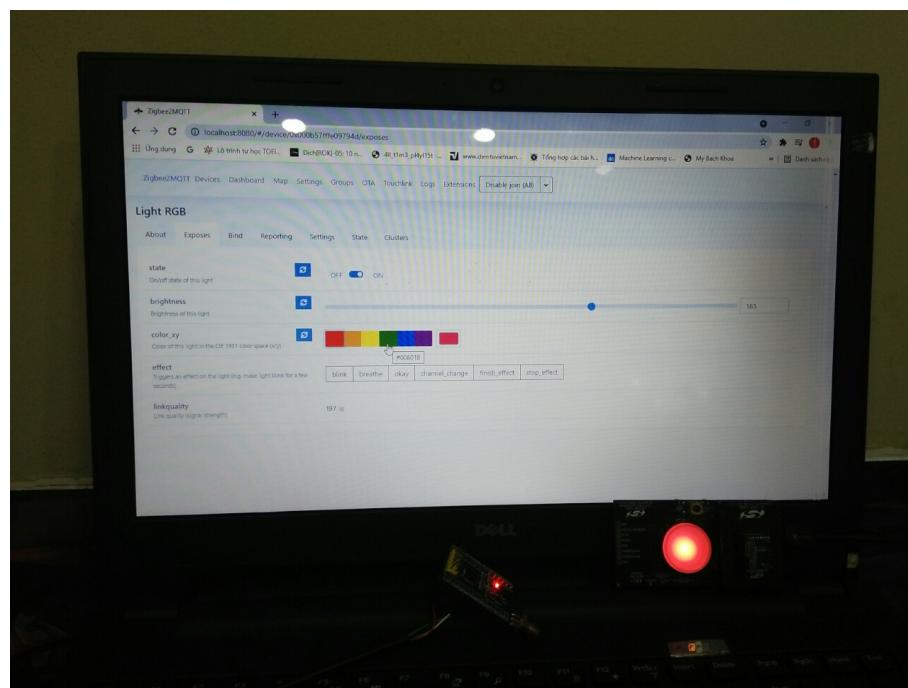
Hình 5.2.6: Tăng độ sáng đèn.

Để giảm độ sáng đèn. Tại mục **brightness** kéo con trỏ trên thanh đèn độ sáng cần chỉnh.

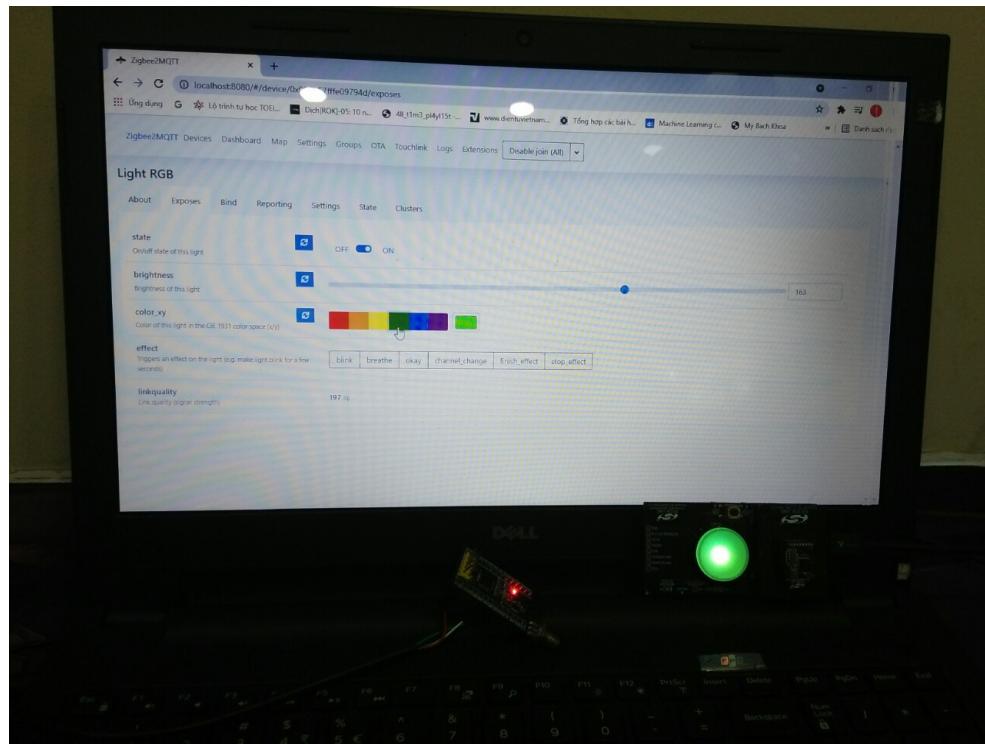


Hình 5.2.7: Giảm độ sáng đèn.

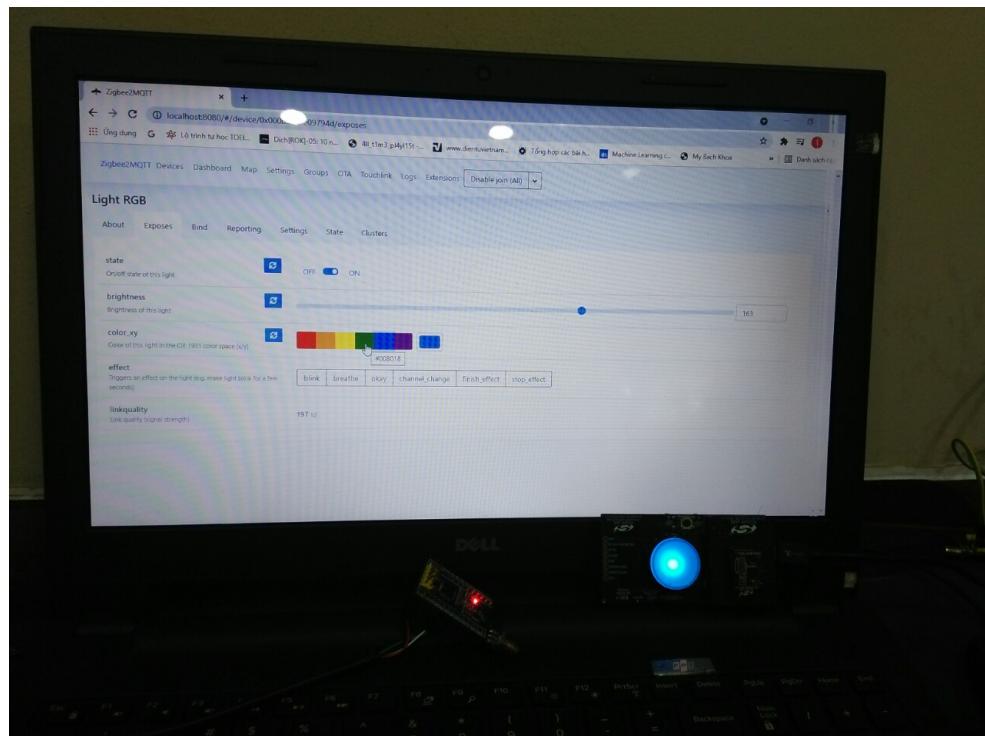
Ngoài điều chỉnh độ sáng và bật tắt bóng đèn, ta cũng có thể điều chỉnh màu cho đèn. Để điều chỉnh màu, trong mục **color_xy** chọn màu cho đèn.



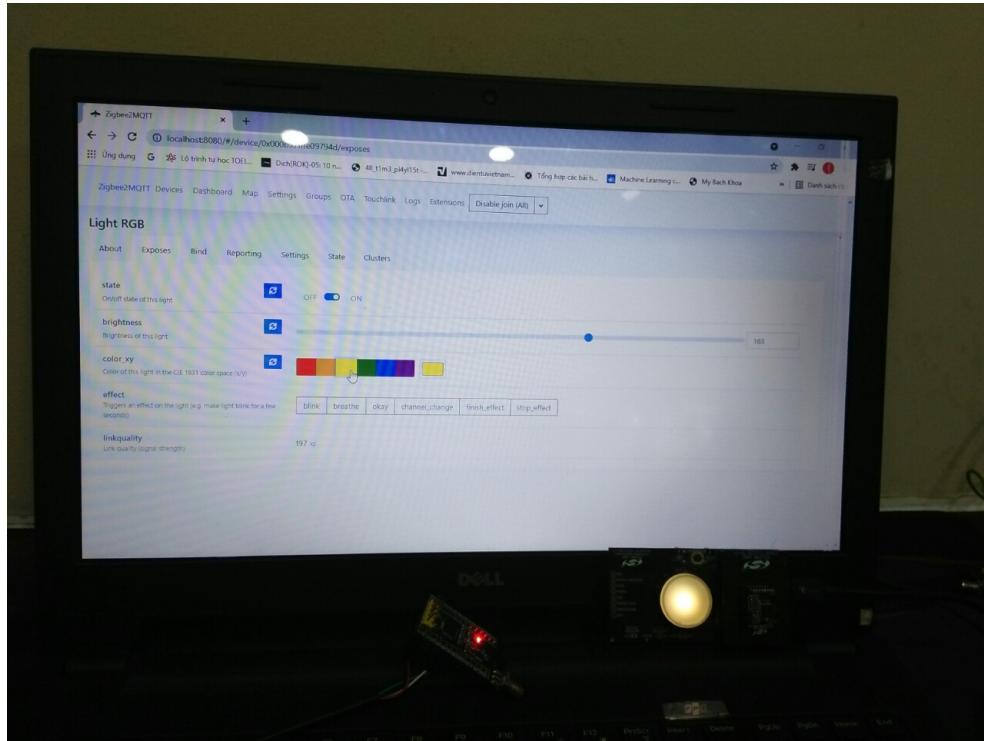
Hình 5.2.8: Đổi đèn sang màu đỏ.



Hình 5.2.9: Đổi đèn sang màu xanh lá.



Hình 5.2.10: Đổi màu đèn sang màu xanh dương.



Hình 5.2.11: Đổi màu đèn sang màu vàng.

6 Kết luận và hướng phát triển

6.1 Kết luận

Về kết quả đề tài, em đã thành công trong việc thiết kế gateway chuyển đổi mạng Zigbee sang MQTT; hoàn thành thử nghiệm thiết kế Zigbe2mqtt trên module CC2530 và đã điều khiển thành công các thiết bị zigbee thông qua Zigbee2MQTT.

Tuy nhiên, em vẫn chưa thực hiện được việc cho module CC2530 làm một node trong mạng Zigbee và điều khiển nó. Chỉ có thể nạp chương trình cho module CC2530 làm router trong mạng Zigbee.

Sau quá trình tìm hiểu và hoàn thành đề tài, bản thân em đã học thêm được nhiều kiến thức về mạng zigbee và giao thức MQTT. Thêm vào đó nhờ việc nghiên cứu và tìm hiểu, em đã có cái nhìn rõ ràng hơn về mảng IOT.

6.2 Hướng phát triển

Đề tài có thể được áp dụng để tạo một mạng zigbee, kết nối và điều khiển các thiết bị zigbee thông qua giao thức MQTT. Đây là một giải pháp hữu ích cho nhà thông minh.

Ngoài ra, Zigbee2MQTT có thể kết nối với máy chủ **Hass**⁵ và tích hợp sâu trong **Hass**. **Hass** cũng là một add-ons chính thức của **Zigbee2MQTT**.

Không chỉ thực hiện trên nền tảng **Window**, Zigbee2MQTT còn có thể triển khai trên **Docker**, **Linux**.

Để tiết kiệm năng lượng, Zigbee2MQTT trên máy tính nhỏ gọn như Raspberry Pi. Moudule CC2530 có thể thay thế bằng các module khác như: Texas Instruments CC2538 PCB, Texas Instruments CC2538 HAT, ZigStar Stick v4, Tube's CC2652P2 USB Coordinator, Texas Instruments CC2531, Gio-dot Z-Bee Duo with CC2652P, ...

⁵Home Assistant là một nền tảng tự động hóa mã nguồn mở chạy trên Python .Hass theo dõi và kiểm soát tất cả các thiết bị trong nhà .

Tài liệu tham khảo

- [1] KoenKK, "Zigbee2MQTT", <https://github.com/koenkk/zigbee2mqtt>.
- [2] KoenKK, "Zigbee2MQTT", <https://www.zigbee2mqtt.io/>.
- [3] KoenKK, "What do I need?", bit.ly/3ucNwXS.
- [4] KoenKK, "Running Zigbee2MQTT", bit.ly/3fasqVJ.
- [5] KoenKK, "Allowing devices to join", bit.ly/34fZYvj.
- [6] KoenKK, "Supported devices", bit.ly/3wovPpV.
- [7] KoenKK, "Zigbee network", bit.ly/3wuH54e.
- [8] KoenKK, "Supported adapters", bit.ly/2SjIEmy.
- [9] KoenKK, "MQTT topics and message structure", bit.ly/3hNkdbV.
- [10] KoenKK, "Configuration", bit.ly/3yAk1CC.
- [11] KoenKK, "Running Zigbee2MQTT on Windows", bit.ly/3vfvq8X.
- [12] KoenKK, "How to create a CC2530 router", bit.ly/3wthmsE.