

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**



Bùi Đăng Hoàng

MSV: 13020171

Điện thoại: 0918280595

Email: bdhoang.ht@gmail.com

**THIẾT KẾ Ổ ĐIỆN THÔNG MINH TƯƠNG THÍCH
CHUẨN ECHONET LITE**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

HÀ NỘI – 2017

**ĐẠI HỌC QUỐC GIA HÀ NỘI
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ**

Bùi Đăng Hoàng

**THIẾT KẾ Ổ ĐIỆN THÔNG MINH TƯƠNG THÍCH
CHUẨN ECHONET LITE**

KHÓA LUẬN TỐT NGHIỆP ĐẠI HỌC HỆ CHÍNH QUY

Ngành: Công nghệ thông tin

Cán bộ hướng dẫn: TS. Nguyễn Hoài Sơn

(ký tên)

HÀ NỘI – 2017

LỜI CẢM ƠN

Tôi xin bày tỏ lòng biết ơn sâu sắc tới TS. Nguyễn Hoài Sơn, thầy đã truyền đạt kiến thức, kỹ năng trong học tập và nghiên cứu, hướng dẫn tận tâm giúp tôi hoàn thành khóa luận này.

Và tôi cũng xin chân thành cảm ơn các thầy, cô, các cán bộ làm việc tại trường Đại học Công nghệ, Đại Học Quốc Gia Hà Nội và đặc biệt là các thầy, cô trong khoa Công nghệ thông tin đã trang bị nền tảng kiến thức giúp tôi trong quá trình học tập và làm khóa luận.

Cuối cùng, tôi xin gửi lời cảm ơn tới gia đình, người thân, bạn bè đã quan tâm, động viên, tạo điều kiện cho tôi trong quá trình học tập.

Hà Nội, ngày tháng năm

Sinh viên

Bùi Đăng Hoàng

LỜI CAM ĐOAN

Tôi xin cam đoan đã thực hiện quá trình làm khóa luận một cách khoa học, trung thực, đúng đắn dưới sự hướng dẫn của TS. Nguyễn Hoài Sơn.

Các tài liệu, bài báo khoa học, giáo trình giảng dạy nếu được sử dụng trong khóa luận đều được trích dẫn đầy đủ. Nếu có điều gì sai trái tôi xin chịu trách nhiệm trước hội đồng về kết quả khóa luận của mình.

Hà Nội, ngày tháng năm

Sinh viên

Bùi Đăng Hoàng

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU	1
1.1. Tổng quan IoT	1
1.1.1. Khái niệm IoT (Internet of Things)	1
1.1.2. Lịch sử phát triển IoT – Xu hướng công nghệ trong tương lai	2
1.1.3. Các thiết bị đo điện trong IoT	3
1.2. Hiệu ứng Hall và ứng dụng hiệu ứng Hall trong đo đặc dòng điện	4
1.2.1. Giới thiệu	4
1.2.2. Ứng dụng của hiệu ứng Hall	5
1.3. Giao thức Echonet Lite trong IoT	6
1.3.1. Tổng quan ECHONET và Echonet Lite	6
1.3.2. Cấu trúc gói tin Echonet Lite	8
1.3.3. Truyền nhận gói tin trong giao thức Echonet Lite	12
CHƯƠNG 2: THIẾT KẾ HỆ THỐNG Ồ CẮM ĐIỆN THÔNG MINH	13
2.1. Đặt vấn đề	13
2.2. Thiết kế ổ cắm	14
2.2.1. Ý tưởng thiết kế	14
2.2.2. Thiết kế tổng quan	15
2.2.3. Ứng dụng theo dõi điện năng	20
2.3. Chức năng tự động	21
2.3.1. Nhận dạng thiết bị trên ổ cắm	21
2.3.2. Kiểm tra ngưỡng hoạt động của ổ cắm khi bật thực hiện bật thiết bị ...	22
CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ.....	24
3.1. Sơ đồ cài đặt hệ thống thực tiễn	24
3.1.1 Board mạch Arduino	25
3.1.2 Module đo dòng ACS712.....	26
3.1.3 Module WiFi ESP8266.....	27
3.1.4 Cài đặt Platform IoT	28
3.1.5 Trang web ứng dụng theo dõi.....	31
3.1.6 Các chức năng tự động	32
3.1.7 Giả lập một số thiết bị Echonet Lite	33
3.2 Đánh giá kết quả thử nghiệm.....	33
CHƯƠNG 4: TỔNG KẾT	35
4.1. Kết quả nghiên cứu đạt được.....	35
4.2. Những hạn chế trong khi thực hiện đề tài	35
4.3. Đề xuất, hướng phát triển	35
PHỤ LỤC : HÌNH ẢNH THỰC TẾ CỦA HỆ THỐNG.....	36
TÀI LIỆU THAM KHẢO.....	37

BẢNG KÍ HIỆU VIẾT TẮT

CSDL	Cơ sở dữ liệu
------	---------------

MỤC LỤC BẢNG BIỂU

<i>Bảng 1.1: ESV code cho các thông báo (Notification).....</i>	<i>10</i>
<i>Bảng 1.2: ESV code cho các phản hồi (Response).....</i>	<i>11</i>
<i>Bảng 1.3: ESV code cho các yêu cầu (Request).....</i>	<i>11</i>

<i>Bảng 2. 1: Giá điện dân dụng hiện nay (Nguồn: EVN.com / Biểu giá bán lẻ điện)</i>	<i>20</i>
--	-----------

<i>Bảng 3. 1: Tổng quan các thông số Arduino.....</i>	<i>25</i>
<i>Bảng 3. 2: Đánh giá kết quả chạy thử nghiệm</i>	<i>34</i>

MỤC LỤC HÌNH VẼ

Hình 1.1: IoT Internet of Things.....	1
Hình 1.2: Dự báo sự phát triển của IoT đến năm 2020 (Nguồn mptelecom.com).....	2
Hình 1.3: Bộ sản phẩm theo dõi điện năng Neurio (Nguồn Neur.io)	3
Hình 1.4: Cơ chế hiệu ứng Hall trên một thanh Hall _[2]	4
Hình 1.5: Đầu đo dòng điện dùng hiệu ứng Hall (8mm)	5
Hình 1.6: Kiến trúc hệ thống Echonet Lite _[6]	7
Hình 1.7: Ví dụ phạm vi miền và cấu hình một hệ thống ứng dụng _[6]	8
Hình 1.8: Cấu trúc một gói tin ECHONET _[6]	9
Hình 1.9: Echonet Lite message header 1 _[6]	9
Hình 1.10: Echonet Lite message header 2 _[6]	10
Hình 1.11: Ví dụ OPC = 0x03 thể hiện 3 thuộc tính được yêu cầu	11
Hình 2.1: Sơ đồ tổng quan hệ thống ổ cắm thông minh.....	15
Hình 2.2: Sơ đồ thiết kế ổ cắm điện thông minh	16
Hình 2.3: Xử lý dữ liệu cường độ dòng điện	16
Hình 2.4: Phương thức gửi dữ liệu cường độ dòng điện	17
Hình 2. 5: Ví dụ một gói tin trả về trạng thái của thiết bị _[7]	18
Hình 2. 6: Quá trình thu thập dữ liệu cảm biến	18
Hình 2.7: Một gói tin yêu cầu trạng thái của một thiết bị _[7]	19
Hình 2.8: Chức năng nhận diện thiết bị.....	21
Hình 2. 9: Sơ đồ thuật toán nhận diện thiết bị	22
Hình 2.10: Chức năng kiểm tra ngưỡng hoạt động ổ cắm.....	23
Hình 3. 1: Mô hình tổng quan toàn bộ hệ thống	24
Hình 3. 2: Các thành phần chi tiết của ổ cắm.....	24
Hình 3.3: Board mạch Arduino R3.....	25
Hình 3. 4: Moduel đo dòng ACS712	26
Hình 3. 5: Sơ đồ nối chân Arduino và ACS712.....	26
Hình 3. 6: Module Wifi ESP8266 V1.....	27
Hình 3. 7: Sơ đồ nối chân Arduino và ESP8266.....	28
Hình 3. 8: Cài đặt Homgateway.....	29
Hình 3. 9: Cài đặt Mosquitto Broker.....	29
Hình 3. 10: Cài đặt Platform Server Side	30
Hình 3. 11: Cài đặt HTTP API.....	30
Hình 3. 12: Danh sách thiết bị được trả về bởi API	30
Hình 3. 13: Đăng nhập và lựa chọn căn nhà	31
Hình 3. 14: Trang chủ theo dõi	31
Hình 3. 15: Lịch sử sử dụng điện năng	32
Hình 3. 16: Thực hiện nhận diện một thiết bị.....	32
Hình 3. 17: Sơ đồ một Node Echonet Lite.....	33

MỞ ĐẦU

Bối cảnh

Trong những năm gần đây một khái niệm mới trong ngành công nghệ được nhắc đến nhiều “Internet of Things” – xu hướng công nghệ trong tương lai. Khái niệm Internet of Things (IoT) được đưa vào năm 1999, khi mà người ta bắt đầu nhìn nhận được tiềm năng của xu hướng này, lúc mà các rào cản giới hạn Internet, khoa học công nghệ dần được khai phá. Chúng ta có thể hiểu IoT là mọi vật có thể kết nối và tương tác với nhau thông qua kết nối mạng Internet, người dùng có thể kiểm soát các đồ vật từ xa thông qua các thiết bị thông minh như máy tính (laptop), máy tính bảng hay điện thoại thông minh (smartphone).

Nhắc đến IoT là không thể thiếu các dự án “Nhà thông minh” - một trong những ứng dụng mạnh mẽ của IoT. Nhà thông minh là kiểu nhà được lắp đặt các thiết bị điện, điện tử có chức năng tự động hoặc bán tự động, thay thế con người thực hiện một số thao tác quản lý, điều khiển thủ công mang lại sự tiện lợi, chuyên nghiệp. Hệ thống các thiết bị điện tử này được kết nối với internet hoặc chung một mạng cục bộ, từ đó chủ nhân căn nhà có thể điều khiển, theo dõi hoạt động của chúng thông qua bảng điện tử được lắp đặt, phần mềm điện thoại di động, giao diện web ...

Trong các dự án nhà thông minh nói riêng hay các dự án IoT nói chung thì thứ không thể thiếu đó chính là nguồn điện cung cấp cho các thiết bị hoạt động. Việc theo dõi nguồn điện này là yếu tố thiết yếu để duy trì, đảm bảo hoạt động ổn định cho các thiết bị để đạt hiệu quả cao.

Lý do chọn đề tài

Việc xây dựng một thiết bị theo dõi nguồn điện cũng như công suất hoạt động của các thiết bị IoT là vô cùng cần thiết trong các dự án IoT hoặc cụ thể hơn là dự án “Nhà thông minh”.

Bên cạnh đó, giao thức kết nối Echonet Lite hiện đang được sử dụng trong rất nhiều các dự án nhà thông minh và các thiết bị IoT tại Nhật Bản. Tương lai nó có thể là một chuẩn kết nối chung cho các thiết bị IoT.

Nội dung đề tài

Tại đề tài này, chúng tôi sẽ xây dựng một ổ cắm điện có thể đo được lượng điện năng mà các thiết bị cắm trên nó hoạt động, sau đó gửi dữ liệu đến máy chủ thông qua giao thức Echonet Lite. Tiếp theo xây dựng các ứng dụng chức năng giúp người dùng có thể theo dõi, quản lý thiết bị điện trên ổ cắm.

THIẾT KẾ Ổ ĐIỆN THÔNG MINH TƯƠNG THÍCH CHUẨN ECHONET LITE

Bùi Đăng Hoàng

K58 Công nghệ thông tin

Tóm tắt Khóa luận tốt nghiệp:

Hiện nay, trong kỷ nguyên công nghệ phát triển vô cùng mạnh mẽ, các thiết bị điện tử tự động xuất hiện ngày càng nhiều nhằm phục vụ con người trong đời sống thường ngày cũng như trong công việc. Sự xuất hiện khái niệm Internet of Things (IoT) hay còn được biết đến như một mạng lưới các thiết bị điện tử có thể giao tiếp với nhau. Tuy mới được khai phá nhưng IoT đang ngày càng khẳng định vị thế hiển nhiên quan trọng của nó trong tương lai. Khóa luận này sẽ tập trung xây dựng lên một thiết bị IoT hoàn chỉnh nhằm giúp chúng ta tiếp cận gần hơn với IoT và hiểu được một thiết bị IoT có thể làm được những gì và làm như thế nào. Cụ thể là một “Ổ cắm điện thông minh” kế thừa chuẩn kết nối Echonet Lite – Một chuẩn kết nối không dây cho các thiết bị IoT đang rất phổ biến tại Nhật Bản hiện nay.

Ổ cắm thông minh sẽ giúp người dùng theo dõi được việc sử dụng điện năng trong căn nhà, từ đó có thể đưa ra những cách thức sử dụng thiết bị điện hợp lý hơn nhằm giảm lượng điện năng hao phí, tiết kiệm chi phí sử dụng điện. Bên cạnh đó, ổ cắm thông minh sẽ giúp việc sử dụng thiết bị điện của người dùng an toàn hơn cũng như cảnh báo sự cố xảy ra với nguồn điện cho người dùng.

Từ khóa: *Internet Of Things, Echonet Lite, MQTT, Ổ điện không dây*

Kết cấu của khóa luận:

Ngoài phần giới thiệu, kết luận và tài liệu tham khảo, nội dung của khóa luận bao gồm 4 chương:

Chương 1: Giới thiệu: Tổng quan IoT, Hiệu ứng Hall và ứng dụng, Giao thức Echonet Lite

Chương 2: Thiết kế hệ thống ổ cắm điện thông minh: Đặt vấn đề, Thiết kế ổ cắm, chức năng tự động

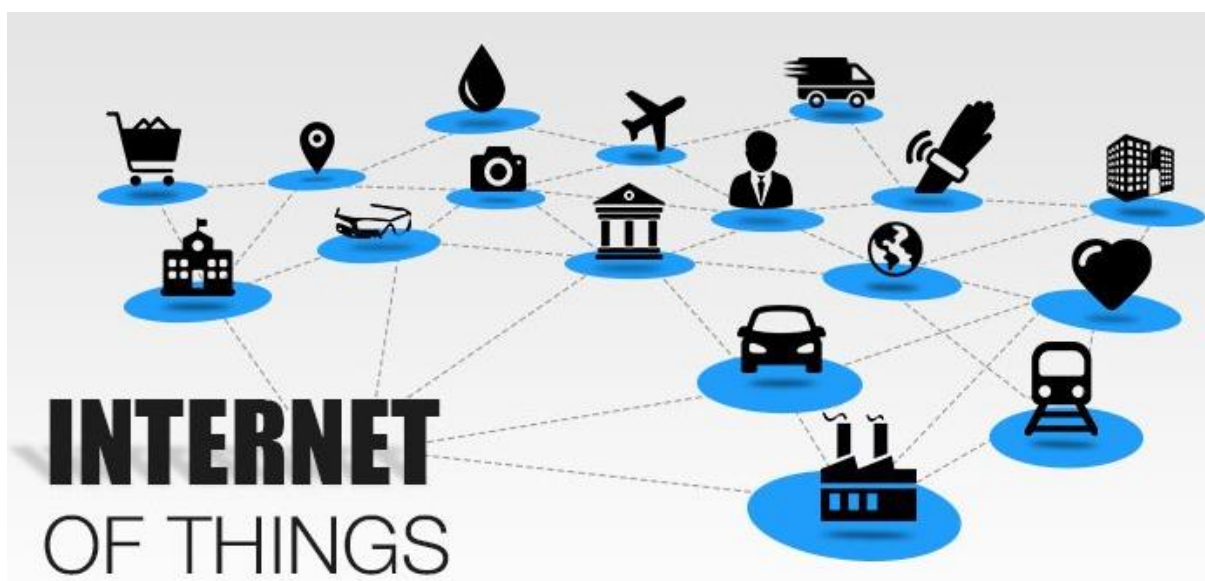
Chương 3: Triển khai hệ thống và đánh giá kết quả: Sơ đồ lắp đặt, Đánh giá kết quả

Chương 4: Tổng kết: Kết quả nghiên cứu, Hạn chế, Đề xuất

CHƯƠNG 1: GIỚI THIỆU

1.1. Tổng quan IoT

1.1.1. Khái niệm IoT (Internet of Things)



Hình 1.1: IoT Internet of Things

“Mạng lưới vạn vật kết nối Internet” hay “Mạng lưới thiết bị kết nối Internet” viết tắt là IoT (Tiếng Anh: Internet of Things) là một kịch bản của một thế giới các đồ vật, khi mà mỗi đồ vật được cung cấp một định danh cho mình và có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng mà không cần sự tương tác trực tiếp với con người. IoT phát triển nhờ sự hội tụ của công nghệ không dây, công nghệ vi – cơ điện tử và Internet. Nói đơn giản là một tập hợp các thiết bị mà có thể được gán một địa chỉ IP rồi kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó^[4].

Hiện tuy vẫn chưa có khái niệm cụ thể về IoT. Nhưng ta có thể nhận biết một thiết bị có thuộc hệ sinh thái IoT hay không qua việc trả lời một số câu hỏi: Sản phẩm của hãng A có thể giao tiếp với sản phẩm của hãng B hay không? Ví dụ như, một khóa của tự động của hãng A có thể giao tiếp gửi/nhận dữ liệu điều khiển đến bóng đèn điện trong nhà của hãng B để tự động bật/tắt đèn khi có người mở cửa hay không? Và trong quá trình đó các thiết bị như điều hòa, ti-vi, ... có thể tham gia cuộc “đối thoại” để có các phản ứng phù hợp hay không?

Ví dụ đơn giản: chiếc tủ lạnh thông thường không được kết nối với thiết bị khác. Nếu chúng ta muốn ghi lại nhiệt độ ở từng thời điểm của tủ lạnh, chúng ta chỉ có thể ghi lại thủ công rồi nhập vào một máy tính hay thiết bị lưu trữ nào đó. Hay như bóng đèn neon ở nhà, khi chúng ta muốn thu thập, điều chỉnh độ sáng của nó thì phải đo thủ công rồi ghi lại. Nếu như máy tính có khả năng giúp con người thu thập tất cả những dữ liệu về mọi thứ xung quanh, chúng ta có thể "theo dõi và đếm mọi thứ, giúp giảm hao phí, chi phí. Chúng ta sẽ biết chính xác khi nào các vật dụng cần phải sửa chữa, thay thế, khi nào chúng còn mới và khi nào thì chúng hết hạn sử dụng. Chưa kể đến

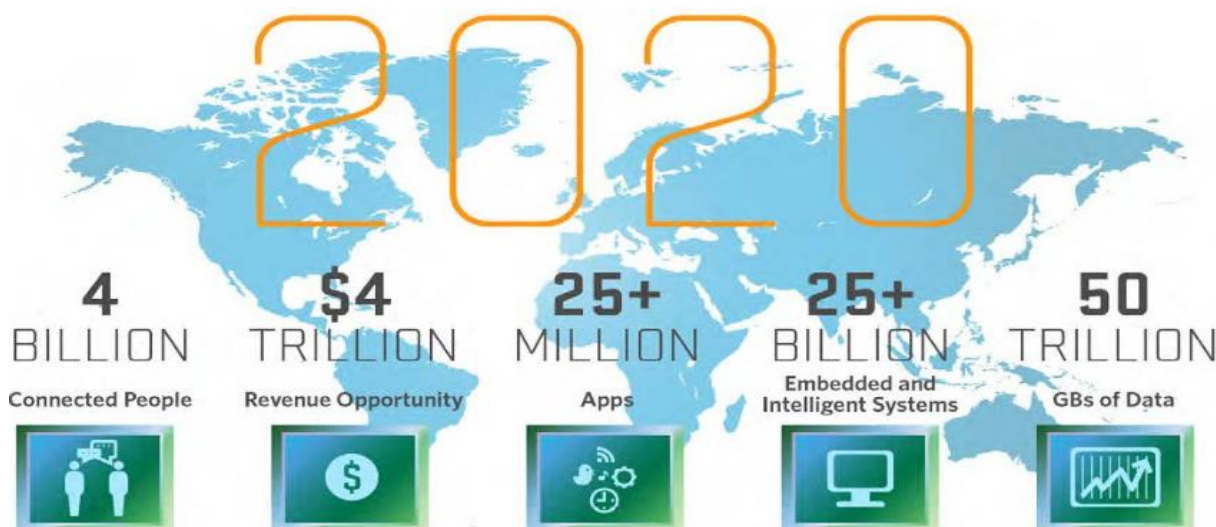
việc chúng ta có thể kiểm soát chúng mọi lúc mọi nơi. IoT có tiềm năng thay đổi thế giới, giống như cách mà Internet đã thay đổi cuộc sống của chúng ta. Ngôi nhà thông minh với các bóng đèn thông minh, máy giặt thông minh, tủ lạnh thông minh... có thể xem là bước đầu của IoT bởi chúng đều được liên kết với nhau và/hoặc liên kết vào Internet.

1.1.2. Lịch sử phát triển IoT – Xu hướng công nghệ trong tương lai

IoT hay “Internet of Things” được đưa ra bởi Kevin Ashton vào năm 1999. Ông là nhà khoa học đã sáng lập ra trung tâm Auto-ID ở Đại học MIT, nơi thiết lập các quy chuẩn toàn cầu cho RFID (một phương thức giao tiếp không dây dùng sóng Radio) cũng như một số loại cảm biến khác. IoT sau đó được dùng nhiều trong các ấn phẩm đến từ các hãng và nhà phân tích.

Mặc dù đã xuất hiện từ khá lâu nhưng trong những năm gần đây IoT mới được nhiều doanh nghiệp cũng như các nhà khoa học để ý nghiên cứu và phát triển. Tại các triển lãm công nghệ CES, triển lãm di động toàn cầu ... rất nhiều hãng sản xuất lớn đồng loạt cho ra mắt các thiết bị thông minh như: ti-vi thông minh, tủ lạnh thông minh, xe tự động cùng rất nhiều ý tưởng về một hệ thống ngôi nhà thông minh. Với tốc độ phát triển chóng mặt của khoa học kỹ thuật trên toàn thế giới, IoT đang dần khẳng định vị thế, khả năng phát triển không giới hạn trong tương lai không xa. Đại kỷ nguyên IoT đang đến rất gần với người tiêu dùng.

Một chi nhánh của Auto-ID tại Châu Âu từng nói về IoT như sau: "Chúng tôi có một tầm nhìn rất rõ ràng - tạo ra một thế giới nơi mà mọi thứ - từ những chiếc máy bay phản lực khổng lồ cho đến từng cây kim khâu - đều được kết nối vào Internet. Mục tiêu này chỉ có thể đạt được khi và chỉ khi tất cả mọi người áp dụng nó ở tất cả mọi nơi". Việc trang bị những công nghệ theo dõi, nhận biết vào những vật thông dụng trong đời sống sẽ làm thay đổi rất nhiều cách chúng ta tương tác với đồ vật cũng như cách tương tác giữa người với người. Theo ước tính của công ty ABI Research, đến năm 2020, toàn thế giới sẽ có 30 tỉ thiết bị được kết nối không dây vào mạng lưới IoT.



Hình 1.2: Dự báo sự phát triển của IoT đến năm 2020 (Nguồn mptelecom.com)

1.1.3. Các thiết bị đo điện trong IoT

Sự phát triển mạnh mẽ của IoT cộng với như sự gia tăng mạnh về các thiết bị sử dụng điện năng cùng với công nghệ ngày càng hiện đại đã dẫn đến việc xuất hiện những thiết bị đo điện với nhiều chức năng nâng cao hơn rất nhiều so với các thiết bị đo điện truyền thống trước đây. Trên thị trường hiện nay có rất nhiều sản phẩm đo điện sử dụng mạng không dây có giúp người dùng theo dõi được dòng điện trong nhà dù không trực tiếp ở nhà hay phải thực hiện những phép đo phức tạp. Một số sản phẩm đo điện thông minh hiện nay trên thị trường:

Neurio – Giải pháp năng lượng thông minh:



Hình 1.3: Bộ sản phẩm theo dõi điện năng Neurio (Nguồn Neur.io)

Đây là sản phẩm của Neurio Technology (Website: <http://neur.io>), với khả năng đo toàn bộ lượng điện năng của căn nhà bằng việc áp dụng hiệu ứng Hall vào thiết bị đo của mình. Sau đó dữ liệu sẽ được đưa lên máy chủ và được xử lý tại đây sau đó xây dựng và cung cấp cho người dùng một số tính năng:

- Theo dõi lưu lượng điện trong thời gian thực (Real-time Energy Monitor)
- Dự toán chi phí sử dụng điện năng
- Theo dõi lịch sử sử dụng điện năng
- Đưa ra một số gợi ý tiết kiệm điện năng

Bên cạnh đó là một số sản phẩm ổ cắm điện thông minh khác, nhìn chung các sản phẩm này đều có một số tính năng chính:

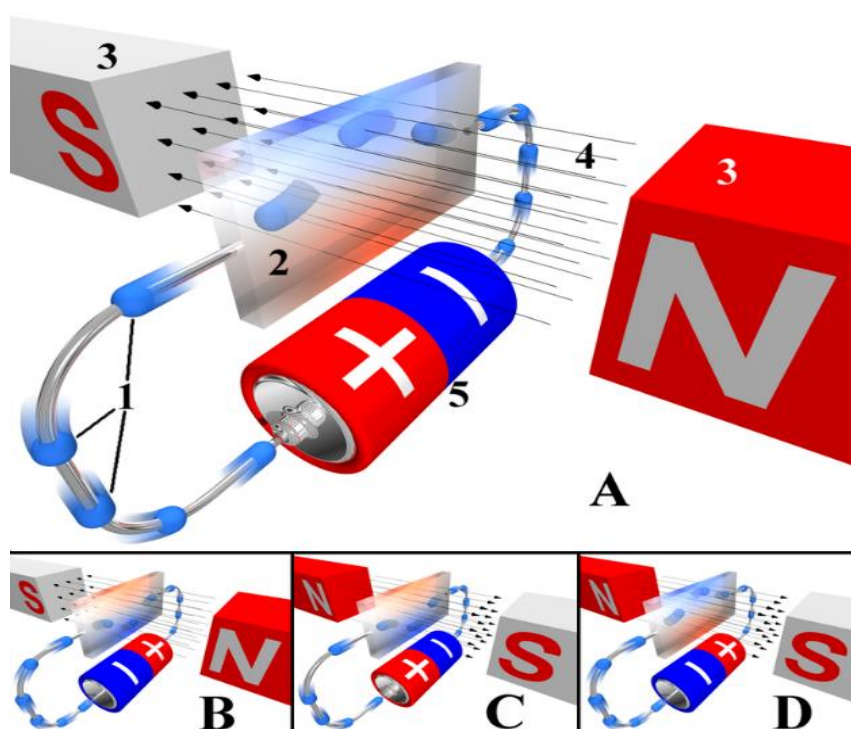
- Đo dòng điện bằng nguyên lý hiệu ứng Hall
- Gửi dữ liệu không dây
- Tích hợp một vài tính năng như đặt lịch, điều khiển từ xa, ...
- Có giao diện web hoặc mobile app cho người dùng

Nhìn chung các loại thiết bị đo điện thông minh này đều áp dụng nguyên lý hiệu ứng Hall và sử dụng mạng không dây để truyền nhận dữ liệu. Đề tài khóa luận của chúng tôi cũng sẽ xây dựng một thiết bị đo với một số tính năng cơ bản như vậy, bên cạnh đó chúng tôi cũng nghiên cứu để tích hợp giao thức Echonet Lite – Một giao thức IoT đang phát triển rất mạnh mẽ ở Nhật Bản hiện nay.

1.2. Hiệu ứng Hall và ứng dụng hiệu ứng Hall trong đo đặc dòng điện

1.2.1. Giới thiệu

Hiệu ứng Hall được khám phá bởi Edwin Herbert Hall vào năm 1879 và trở thành một trong các hiệu ứng cơ bản của vật lý học. Hiệu ứng Hall là hiệu ứng xuất hiện trên một vật dẫn có dòng điện chạy qua khi được đặt vào trong một từ trường. Một hiệu ứng vật lý được thực hiện khi áp dụng một từ trường vuông góc lên một bản làm bằng kim loại hay chất bán dẫn hay chất dẫn điện nói chung (thanh Hall) đang có dòng điện chạy qua. Lúc đó người ta nhận được hiệu điện thế (hiệu thế Hall) sinh ra tại hai mặt đối diện của thanh Hall. Tỷ số giữa hiệu thế Hall và dòng điện chạy qua thanh Hall gọi là điện trở Hall, đặc trưng cho vật liệu làm nên thanh Hall (một chất bán dẫn).



Hình 1.4: Cơ chế hiệu ứng Hall trên một thanh Hall_[2]

Chú thích: 1: Electron; 2: Thanh Hall; 3: Nam châm;

4: Từ trường; 5: nguồn điện

Màu đỏ trên thanh Hall thể hiện sự tập trung của điện tích dương, ngược lại màu xanh thể hiện sự tập trung của điện tích âm. Trên các hình B, C, D chiều của nguồn

điện và từ trường được đổi ngược. Hiệu ứng Hall được giải thích dựa vào bản chất của dòng điện chạy trong vật dẫn điện. Dòng điện này chính là sự chuyển động của các điện tích (ví dụ như electron trong kim loại). Khi chạy qua từ trường, các điện tích chịu lực Lorentz bị đẩy về một trong hai phía của thanh Hall, tùy theo điện tích chuyển động đó là âm hay dương. Sự tập trung điện tích đó tạo ra sự tích điện trái dấu ở hai đầu thanh Hall, gây ra hiệu điện thế Hall.

Công thức liên hệ giữa hiệu điện thế Hall, dòng điện và từ trường là:

$$V_H = \frac{IB}{den}$$

Trong đó: V_H là hiệu điện thế Hall, I là cường độ dòng điện, B là cường độ từ trường, d là độ dày thanh Hall, e là điện tích của hạt mang điện chuyển động trong thanh Hall, n là mật độ các hạt mang điện này trong thanh Hall.

1.2.2. Ứng dụng của hiệu ứng Hall

Hiệu ứng Hall nhạy cảm với từ trường, mà từ trường được sinh ra từ một dòng điện bất kì do đó có thể đo “cường độ dòng điện qua một đoạn dây” khi đưa đoạn dây này gần thiết bị đo. Thiết bị đo có 3 đầu ra: một dây nối đất, một dây nguồn để tạo dòng chạy trong thanh Hall, một dây ra cho biết hiệu điện thế Hall. Phương pháp đo dòng điện này không cần sự tiếp xúc cơ học với mạch điện, hầu như không gây thêm điện trở phụ của máy đo trong mạch điện, và không bị ảnh hưởng bởi nguồn điện của mạch điện, tăng tính an toàn cho phép đo.



Hình 1.5: Đầu đo dòng điện dùng hiệu ứng Hall (8mm)

Công suất tiêu thụ của một mạch điện được tính bằng tích giữa cường độ dòng điện và hiệu điện thế trên mạch. Vậy có thể đo được công suất này bằng cách đo dòng điện ở trên và đồng thời dùng hiệu điện thế của mạch điện để nuôi dòng qua thanh Hall. Phương pháp này có thể được cải tiến để đo công suất dòng điện xoay chiều trong sinh hoạt dân dụng. Nó thường chính xác hơn các thiết bị đo truyền thống mà ít gây cản trở dòng điện.

Hiệu ứng Hall có thể dùng để xác định vị trí cơ học. Các thiết đo kiểu này không có một chi tiết cơ học nào do đó có thể được chế tạo kín, chịu được bụi, độ ẩm, ... điều này giúp các thiết bị đo này hoạt động được ở nhiều vị trí, môi trường hơn các thiết bị quang học hay cơ điện thông thường.

Hiệu ứng Hall cũng có thể ứng dụng trong việc dò các chuyển động quay. Ví dụ khi quay ổ khóa khởi động của ô tô, xe máy: một nam châm gắn vào ổ khóa sẽ quay theo, gây nên thay đổi từ trường, được cảm nhận bởi một thiết bị dùng hiệu ứng Hall, từ đó hệ thống điều khiển sẽ nhận được lệnh khởi động. Phương pháp này tiện lợi vì nó không gây hao mòn như các thiết bị cơ học khác. Việc dò chuyển động quay như vậy cũng được ứng dụng vào hệ thống hãm phanh chống trượt của ô tô, tăng tính an toàn cho người sử dụng đồng thời tăng tuổi thọ của bộ phanh.

Hệ thống IoT cũng có rất nhiều các cảm biến hoặc thiết bị sử dụng hiệu ứng Hall. Do nó mang lại độ chính xác cao cũng như giúp tăng tuổi thọ của các thiết bị.

1.3. Giao thức Echonet Lite trong IoT

1.3.1. Tổng quan ECHONET và Echonet Lite

Trong hệ thống Internet of Things, các thiết bị sẽ phải giao tiếp được với nhau. Cụ thể trong một gia đình, các thiết bị giao tiếp và quá trình giao tiếp cũng phải được diễn ra với tốc độ và băng thông lớn nhưng chi phí thấp. Do vậy việc cho phép các thiết bị giao tiếp với nhau thông qua mạng nội bộ có sẵn của gia đình là rất cần thiết. Hơn nữa để hiểu và nhận dữ liệu qua lại, các thiết bị sẽ phải sử dụng chung một giao thức truyền tin. Nhận thấy điều này, giao thức ECHONET đã được cho ra đời bởi công ty “ECHONET”.

[6] ECHONET (Energy Conservation and Homecare Network) là một mạng lưới các thiết bị được nhắc đến ở trên, bao gồm:

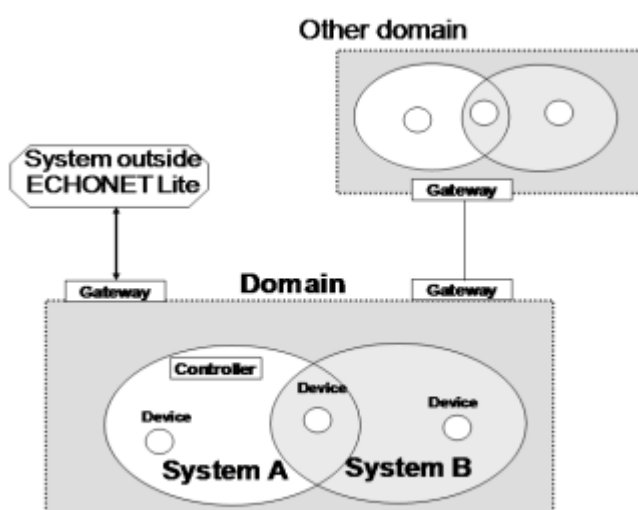
- + Một giao thức truyền tin tin cậy, không dây và kinh tế
- + Tương thích với mọi ngôi nhà
- + Có thể được sử dụng bởi các cá nhân để tạo điều kiện phát triển cho hệ thống ứng dụng
- + Truyền thông trung gian và hỗ trợ các công cụ phát triển nhằm giảm gánh nặng cho nhà phát triển thiết bị

Echonet Lite được phát triển dựa trên ECHONET, nhưng được thiết kế dễ dàng hơn cho nhà phát triển hệ thống mạng gia đình. Các thiết bị tuân thủ Echonet Lite và

ECHONET không thể kết nối với nhau nhưng có thể cùng tồn tại trong một hệ thống. Mục tiêu của Echonet Lite:

- Dễ dàng phát triển cho hệ thống các ngôi nhà khác nhau: Mạng gia đình sẽ hoạt động tốt ngay cả khi kết nối và vận hành các thiết bị từ các nhà sản xuất khác nhau. Một giao thức truyền thông chung giữa các thiết bị giúp liên kết chúng ở cấp hệ thống. Người dùng có thể chọn và cài đặt thiết bị thích hợp nhất với nhu cầu của họ từ một loạt các sản phẩm tương thích Echonet Lite từ các nhà sản xuất khác nhau.
- Kéo dài chu trình hoạt động của các dịch vụ hay các thiết bị: Công việc này được giao cho Echonet Lite adapter trung gian xử lý, các thiết bị chỉ được thiết kế đơn giản nhất có thể. Ví dụ như các thiết bị chỉ phải hồi đáp khi nhận được yêu cầu từ adapter.
- Dễ dàng cài đặt hệ thống, lắp đặt, thay thế và di chuyển thiết bị
- Kết nối và cùng tồn tại với các hệ thống khác

Kiến trúc của một hệ thống Echonet Lite được mô tả như hình 1.6

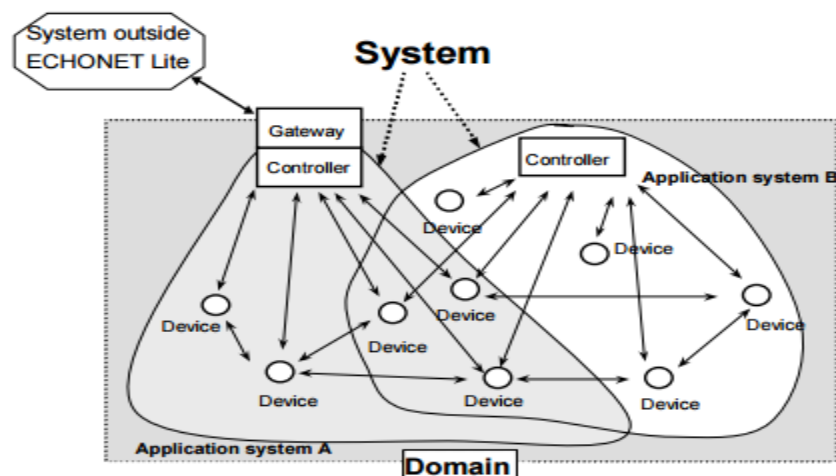


Hình 1.6: Kiến trúc hệ thống Echonet Lite^[6]

Một hệ thống Echonet Lite kết hợp các nhóm thiết bị có cùng chức năng quản lý tài sản, bảo mật, ... Do đó khu vực lớn nhất mà Echonet Lite có thể quản lý được gọi là miền. Mỗi một miền sẽ được chỉ định kiểm soát một loạt các thiết bị (thiết bị gia đình, thiết bị gia dụng và thiết bị điện tử tiêu dùng, cảm biến, điều khiển, điều khiển từ xa, ...) xuất hiện trong phạm vi của nó. Hệ thống nằm trong miền, thực hiện hoạt động truyền thông và liên kết giữa các thiết bị với bộ giám sát, điều khiển. Một hệ thống chỉ thực hiện một số lĩnh vực nhất định, do đó một miền có thể gồm nhiều hệ thống, một thiết bị có thể tồn tại và hoạt động trong một hoặc nhiều hệ thống. Khi một hệ thống

muốn kết nối với một hệ thống khác nằm ngoài miền sẽ phải thông qua một Gateway Interface.

Dưới đây là một ví dụ được xây dựng phổ biến của một hệ thống Echonet Lite:

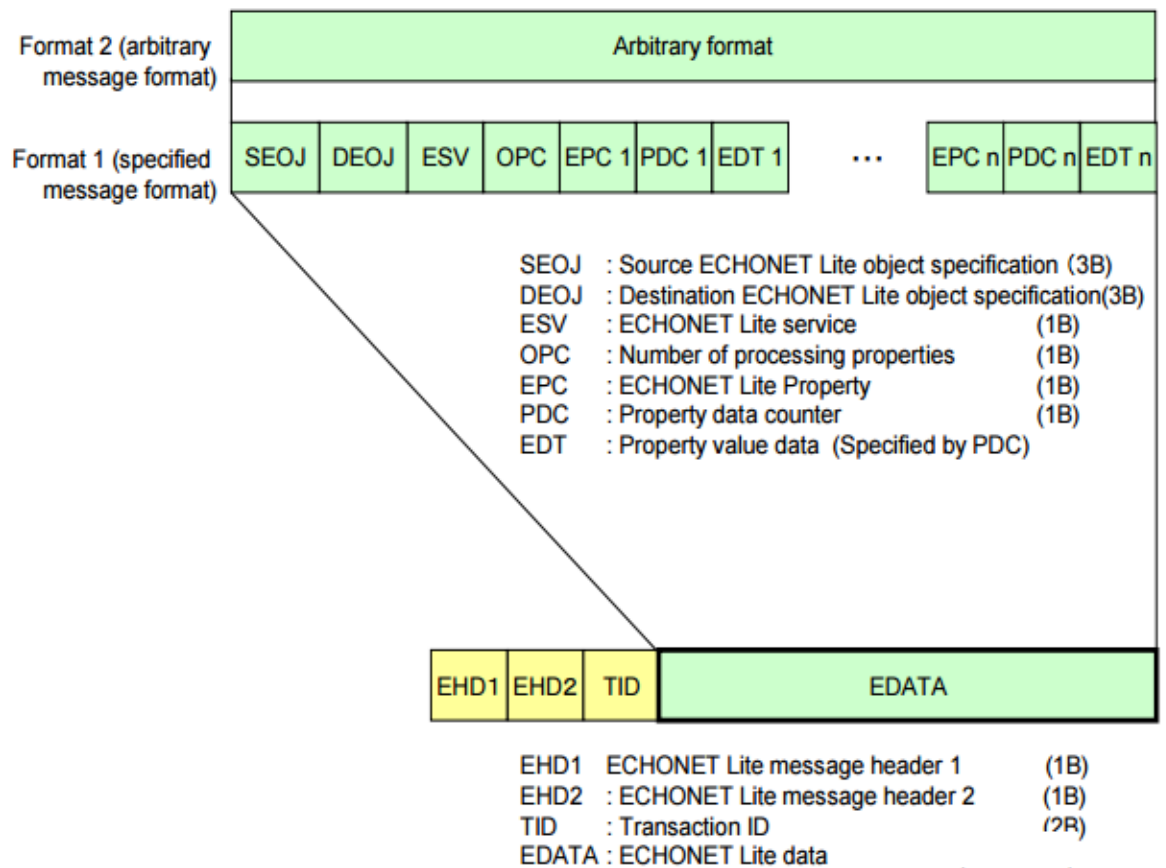


Hình 1.7: Ví dụ phạm vi miền và cấu hình một hệ thống ứng dụng^[6]

Theo hình 1.7, mỗi một Node Echonet Lite (được định nghĩa là một thiết bị hoặc bộ điều khiển kết nối với mạng) trong hệ thống có thể trao đổi dữ liệu một cách tự do với nhau và với bộ điều khiển (cũng là một Node, được gọi chung là Node Controller). Hình này cho hai hệ thống ứng dụng A, B trong cùng một miền, các Node Device trong miền có thể thuộc một, hoặc cả hai hệ thống này. Mỗi hệ thống thực hiện điều khiển, giám sát các thiết bị kết nối với hệ thống mỗi thiết bị không chỉ giao tiếp được với bộ điều khiển trong hệ thống mà còn có thể giao tiếp với các thiết bị trong hệ thống hoặc hệ thống khác.

1.3.2. Cấu trúc gói tin Echonet Lite

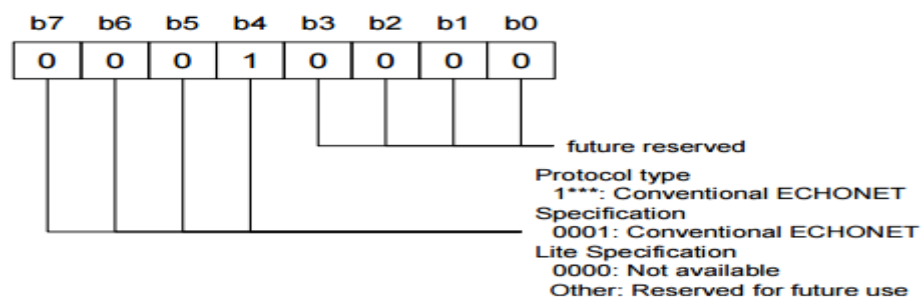
Các thiết bị sử dụng giao thức Echonet Lite sẽ trao đổi với nhau qua các gói tin. Và các gói tin có cấu trúc giống nhau được gọi chung là gói tin Echonet. Cấu trúc trung của các gói tin này như sau:



Hình 1.8: Cấu trúc một gói tin ECHONET_[6]

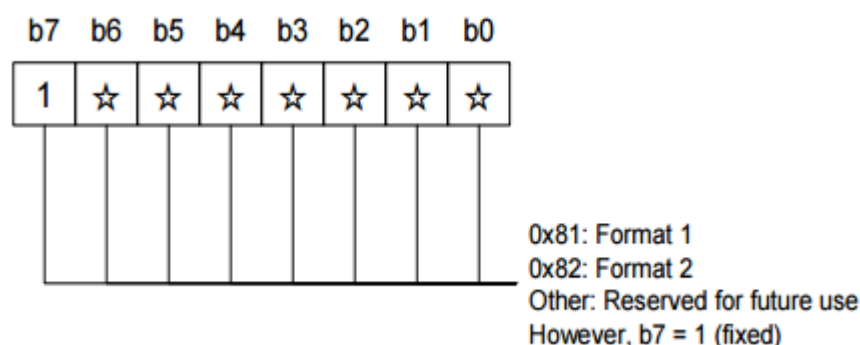
Một gói tin được chia thành hai nhóm Byte chính đó là nhóm Byte header (màu vàng) và nhóm Byte data (màu xanh) trên hình 1.5. Nhóm Byte data có định dạng tự do hoặc theo quy định được xác định bởi phần header.

- EHD1 (Echonet Lite message header 1): Đây là trường header đầu tiên của gói tin có độ dài 1B. Trường này có giá trị như hình 1.9. Giá trị được cố định bởi nhà phát triển.



Hình 1.9: Echonet Lite message header 1_[6]

- EHD2 (Echonet Lite message header 2): Trường này có độ dài 1 B, quy định định dạng của EDATA là sẽ là 1 hay 2. Được mô tả ở hình 1.10.



Hình 1.10: Echonet Lite message header 2_[6]

- TID (Transaction ID): Trường này có độ dài 2B, có nhiệm vụ đánh số ID cho gói tin phục vụ cho quá trình gửi – nhận để phân biệt các gói tin với nhau.

- SEOJ (Source Echonet Lite Object): Trường này có độ dài 3B. Khai báo thiết bị nguồn của gói tin. Các Byte này được kí hiệu X1, X2, X3 với X1: group code, X2: class code, X3: instance code. Mỗi một thiết bị Echonet sẽ được định danh như sau X1.X2.X3. Mã này tương ứng với từng thiết bị được quy định bởi nhà phát triển ECHONET.

- DEOJ (Destination Echonet Lite Object): Cũng có độ dài 3B tương tự như trường SEOJ nhưng khai báo thiết bị mà gói tin sẽ được gửi đến.

- ESV (Echonet Lite service): Có độ dài 1B. Quy định kiểu gói tin là “yêu cầu”, “phản hồi” hay “quảng bá”. Chi tiết được mô tả ở bảng 1.1, 1.2 và 1.3

Service Code (ESV)	ECHONET Lite Service Content	Symbol	Remarks
0x60	Property value write request (no response required)	SetI	Broadcast possible
0x61	Property value write request (response required)	SetC	
0x62	Property value read request	Get	Broadcast possible
0x63	Property value notification request	INF_REQ	Broadcast possible
0x64-0x6D	Reserved for future use		
0x6E	Property value write & read request	SetGet	Broadcast possible
0x6F	Reserved for future use		

Bảng 1.1: ESV code cho các thông báo (Notification)

Service Code (ESV)	ECHONET Lite Service Content	Symbol	Remarks
0x71	Property value Property value write response	Set_Res	ESV=0x61 response; Individual response
0x72	Property value read response	Get_Res	ESV=0x62 response; Individual response
0x73	Property value notification	INF	*1 : Both individual notification and broadcast notification
0x74	Property value notification (response required)	INFC	Individual notification
0x75-0x79	Reserved for future use		
0x7A	Property value notification response	INFC_Res	ESV=0x74 response; Individual response
0x7B-0x7D	Reserved for future use		
0x7E	Property value write & read response	SetGet_Res	ESV=0x6E response; Individual response
0x7F	Reserved for future use		

Bảng 1.2: ESV code cho các phản hồi (Response)

Service Code (ESV)	ECHONET Lite Service Content	Symbol	Remarks
0x50	Property value write request "response not possible"	SetI_SNA	ESV=0x60 response not possible; Individual response
0x51	Property value write request "response not possible"	SetC_SNA	ESV=0x61 response not possible; Individual response
0x52	Property value read "response not possible"	Get_SNA	ESV=0x62 response not possible; Individual response
0x53	Property value notification "response not possible"	INF_SNA	ESV=0x63 response not possible; Individual response
0x54-0x5D	Reserved for future use		
0x5E	Property value write & read "response not possible"	SetGet_SNA	ESV=0x6E response not possible; Individual response
0x5F	Reserved for future use		

Bảng 1.3: ESV code cho các yêu cầu (Request)

- OPC (Processing Target Property Counter): Có độ dài 1B, cho biết số thuộc tính đằng sau của gói tin. Ví dụ như hình 1.11



Hình 1.11: Ví dụ OPC = 0x03 thể hiện 3 thuộc tính được yêu cầu

- EPC (Echonet Lite property): Có độ dài 1B cho biết thuộc tính nào được yêu cầu. Quy định về mã thuộc tính của từng thiết bị được cung cấp bởi nhà phát triển ECHONET

- PDC (Property data counter): Có độ dài 1B cho biết độ dài của trường EDT đằng sau.

- EDT (Property value data): Độ dài được quy định bởi nhà phát triển, giá trị là dữ liệu của thuộc tính đã được quy đổi sang Byte.

1.3.3. Truyền nhận gói tin trong giao thức Echonet Lite

- Sử dụng UDP (User Datagram Protocol) ở Layer 4 và IP (Internet Protocol) ở Layer 3. Mỗi một node Echonet Lite sở hữu một địa chỉ IP. Mỗi một gói tin Echonet tương ứng với một gói tin UDP. Cổng đích của gói tin UDP luôn là 3610 với mọi loại gói tin (yêu cầu, hỏi đáp, thông báo). Để quảng bá gói tin đến tất cả các node trong mạng, Echonet Lite gửi một gói tin Multicast đến địa chỉ IP 224.0.23.0 nếu sử dụng IPv4 hoặc ff02::1 nếu sử dụng IPv6. Một node Echonet Lite luôn lắng nghe gói tin Multicast và Unicast ở cổng 3610. Gói tin Multicast thường được dùng khi một node muốn quảng bá thông tin của nó đến tất cả các node khác trong mạng hoặc tìm kiếm các node khác trong mạng lưới. Sau khi đã biết được địa chỉ IP của các nhau, các node sẽ thực hiện việc trao đổi dữ liệu thông bằng các gói tin Unicast.

- Sử dụng TCP (Transmission Control Protocol) ở Layer 4 và IP (Internet Protocol) ở Layer 3. Mỗi một node cũng sở hữu một địa chỉ IP và mỗi một gói tin Echonet tương ứng với một gói tin TCP. Cổng đích của gói tin TCP là 3610, sau khi kết nối được thiết lập, sẽ không dùng đến cổng này nữa. Gói tin trả lời sẽ được trả về ngay trong kết nối hiện tại. Để quảng bá hoặc tìm kiếm các node khác, một node Echonet Lite vẫn phải sử dụng các gói tin UDP Multicast. Một node Echonet Lite hỗ trợ TCP vẫn phải lắng nghe các gói tin UDP Unicast và UDP Multicast ở cổng 3610 để nhận và xử lý chúng.

CHƯƠNG 2: THIẾT KẾ HỆ THỐNG Ổ CẮM ĐIỆN THÔNG MINH

2.1. Đặt vấn đề

Trong ngôi nhà thông minh, tất cả các thiết bị muốn hoạt động đều phải có nguồn điện. Do đó, một ngôi nhà thông minh chắc chắn nó sẽ phải kiểm soát được các nguồn điện này. Người dùng thông qua việc kiểm soát sẽ quản lý được lượng điện năng tiêu thụ để thực hiện điều chỉnh sử dụng thiết bị cho hợp lý. Để thực hiện được điều này, cần phải có các thiết bị đo dòng điện gắn trên các thiết bị, bên cạnh một số thiết bị điện chỉ là thiết bị điện đơn thuần nên việc cho ra đời một ổ cắm thông minh thực hiện đo điện năng của các thiết bị đang hoạt động trên nó là cần thiết.

Hiện nay trên thị trường có rất nhiều ổ cắm thông minh như vậy, tuy nhiên chúng thường làm việc độc lập và không có khả năng tương tác với các thiết bị khác trong ngôi nhà. Hệ thống ổ cắm thông minh của chúng tôi đã được thiết kế để tránh những nhược điểm trên. Qua tìm hiểu, Echonet Lite là một giao thức được thiết kế cho các hệ thống IoT sử dụng mạng không dây. Bên cạnh đó đây là một giao thức mới xuất hiện, được các công ty Nhật Bản ứng dụng nhiều trong các sản phẩm IoT của họ, cùng với số các thiết bị Echonet Lite đã được sản xuất rất lớn. Do đó chúng tôi đã lựa chọn kế thừa Echonet Lite và trong sản phẩm của mình.

Ổ cắm thông minh Echonet Lite cần giúp người dùng theo dõi được lượng điện năng tiêu thụ cũng như phát hiện được các sự cố bất thường xảy ra với hệ thống điện trong ngôi nhà của mình. Dựa theo nhu cầu của một số người dùng mong muốn biết được số tiền mình phải thanh toán cho lượng điện năng mà mình đã sử dụng cũng như dự toán được chi phí sẽ phải thanh toán cho đến cuối tháng, ổ cắm thông minh có thể theo dõi điện năng và tính toán được chi phí này.

Ổ cắm thông minh có thể theo dõi và phát hiện sự bất thường trong hoạt động của các thiết bị. Với các kịch bản có thể xảy ra như: chập cháy thiết bị, đoản mạch dòng điện, mất điện ... ổ cắm sẽ phải thông báo hoặc cảnh báo cho chủ căn nhà hoặc đèn công an, lực lượng phòng cháy chữa cháy tại địa phương.

Mỗi một ổ cắm điện chỉ được thiết kế để có thể chịu một dòng điện có độ lớn nào đó, khi các thiết bị lấy nguồn điện từ ổ cắm này sử dụng dòng điện lớn hơn dòng điện được thiết kế ban đầu của ổ cắm có thể làm quá tải ổ cắm. Điều này có thể làm hỏng ổ cắm, hỏng thiết bị sử dụng điện, nguy cơ chập cháy làm mất điện cả hệ thống điện trong nhà. Do đó ổ cắm thông minh của chúng tôi sẽ nhận lệnh điều khiển bật tắt một thiết bị của người dùng, sau đó thực hiện kiểm tra xem khi thiết bị được bật lên có làm quá tải dòng trên ổ cắm hay không, nếu không mới cho phép thiết bị được bật lên. Để làm được điều này, bài toán đặt ra là phải nhận diện được xem thiết bị nào đang được cắm và hoạt động trên ổ cắm.

2.2. Thiết kế ổ cắm

2.2.1. Ý tưởng thiết kế

Hiệu ứng Hall có thể đo cường độ dòng chạy qua một dây điện khi chỉ cần đưa đoạn dây này gần thiết bị đo. Do đó sử dụng phương pháp Hall để đo dòng điện sẽ không cần sự tiếp xúc cơ học với mạch điện, không gây thêm điện trở phụ của thiết bị đo cho mạch điện, và không bị ảnh hưởng bởi nguồn điện của mạch điện, tăng tính an toàn cho máy đo, người sử dụng. Vì thế nên chúng tôi đã lựa chọn sử dụng các thiết bị đo sử dụng hiệu ứng Hall trong dự án này.

Thiết bị đo sau khi đo được dữ liệu điện sẽ phải gửi dữ liệu này đi thông qua mạng không dây. Qua tìm hiểu, nghiên cứu và phân tích kỹ lưỡng nhu cầu cũng như quy mô của công việc, chúng tôi đã lựa chọn kế thừa giao thức Echonet Lite vào trong thiết bị đo này và chúng tôi sẽ xây dựng một hệ thống nhà thông minh dựa trên giao thức Echonet Lite. Với một platform có sử dụng Echonet Lite được xây dựng bởi Nguyễn Việt Bắc, hệ thống đo điện của chúng tôi sẽ đóng vai trò như một node thiết bị có khả năng gửi – nhận dữ liệu từ node điều khiển của platform.

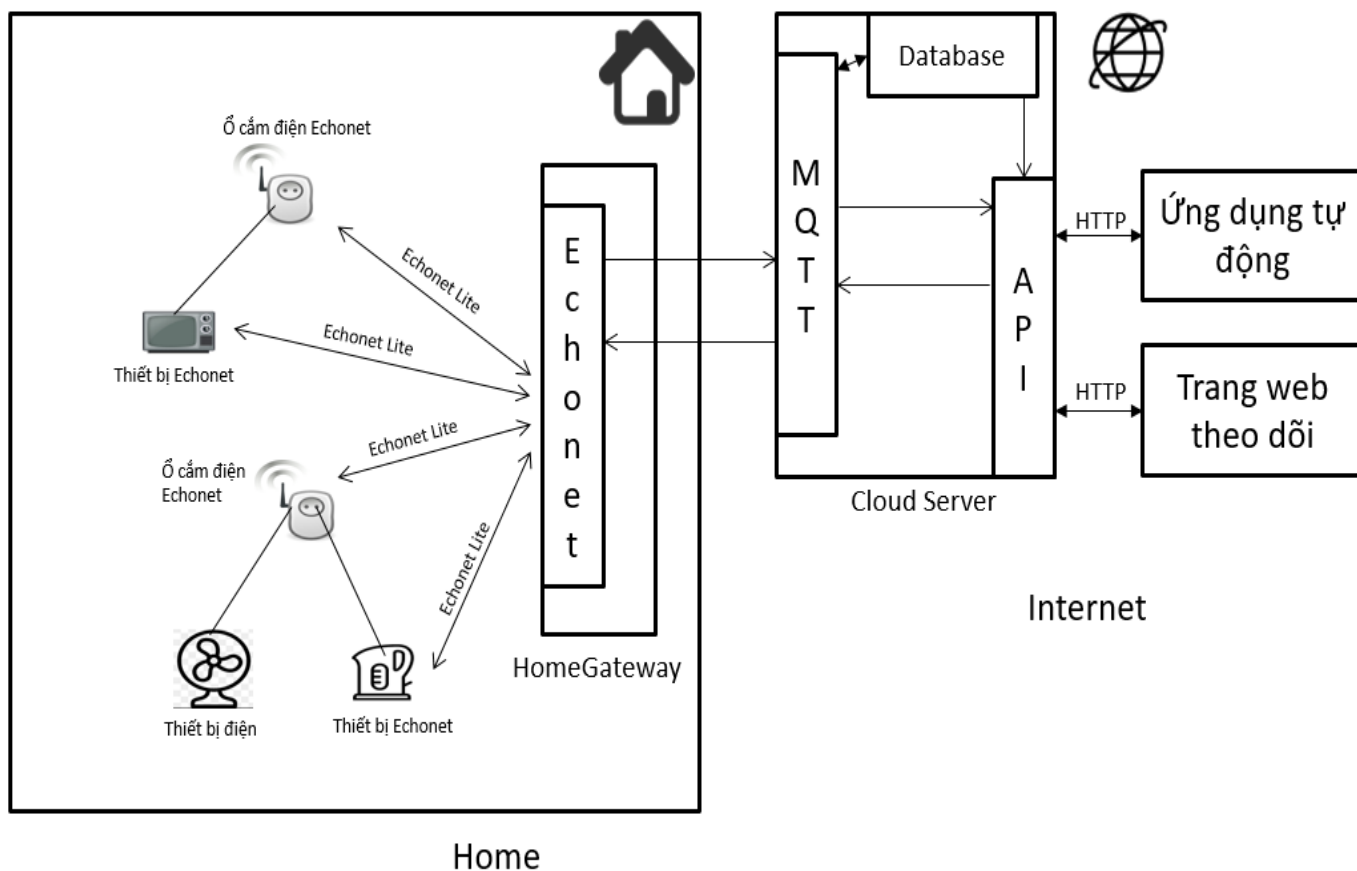
Tiếp theo, thông qua một API được thiết kế trên platform chúng tôi sẽ xây dựng một ứng dụng chức năng để cung cấp một số dịch vụ cho người dùng nhằm hiện đại hóa ngôi nhà của họ, đem đến những trải nghiệm tích cực khi sử dụng hệ thống nhà thông minh của chúng tôi.

Sau khi thực hiện nghiên cứu nhu cầu của nhiều người dùng khác nhau, chúng tôi đã xây dựng được một hệ thống quản lý điện năng với một vài tính năng đáng chú ý như sau:

- Tính năng cho phép người dùng theo dõi điện năng đang sử dụng bởi ngôi nhà của họ
- Tính năng dự toán chi phí sử dụng điện năng cho phép người dùng biết được chi phí phải thanh toán tính đến thời điểm hiện tại dựa trên lượng điện năng mà ngôi nhà đã sử dụng và số tiền họ phải thanh toán đến cuối tháng nếu duy trì việc sử dụng điện năng như hiện tại.
- Tính năng cảnh báo quá tải sẽ thông báo cho người dùng biết việc bật một thiết bị có gây quá tải cho nguồn cấp điện hay không.

2.2.2. Thiết kế tổng quan

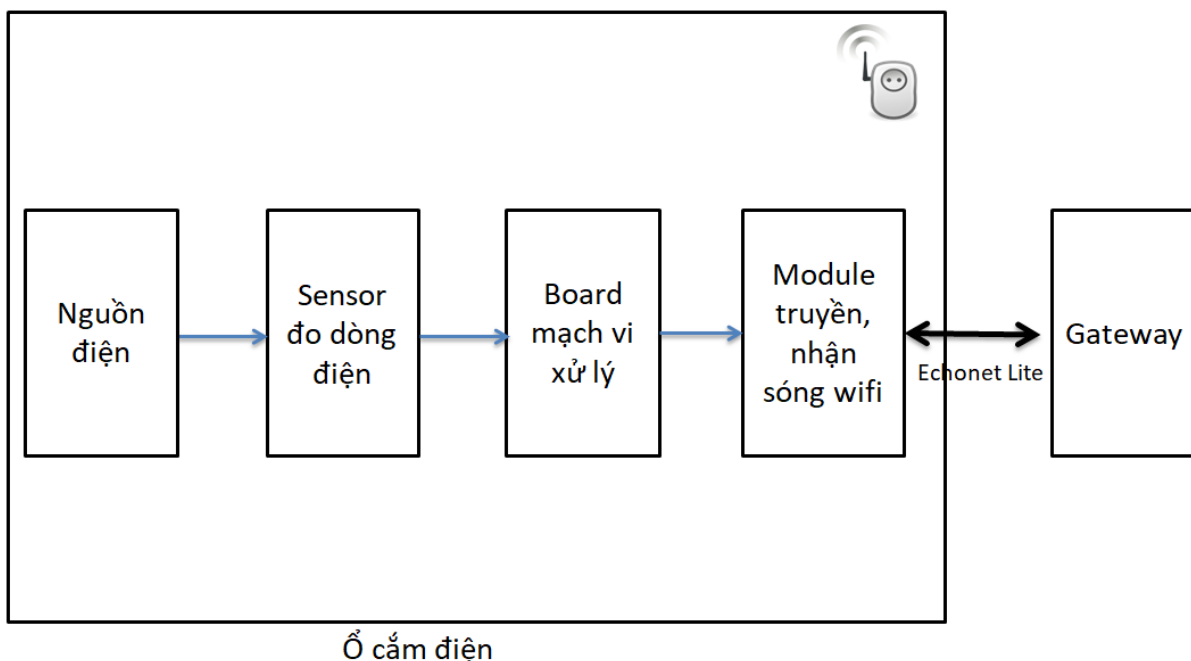
Tổng quan toàn bộ mô hình hoạt động của hệ thống ổ cắm trong nhà thông minh:



Hình 2.1: Sơ đồ tổng quan hệ thống ổ cắm thông minh

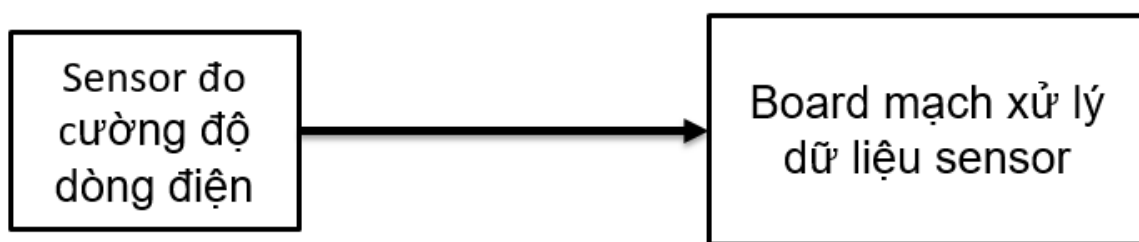
Theo mô hình, các thiết bị Echonet Lite sẽ trao đổi dữ liệu trực tiếp với HomeGateway trong nhà. Ổ cắm điện Echonet Lite sẽ gửi dữ liệu công suất tức thời cho Node điều khiển ở HomeGateway, sau đó HomeGateway publish dữ liệu lên MQTT Broker với chủ đề là dữ liệu của người dùng là chủ căn nhà, khi đó một MQTT Client khác sẽ nhận và xử lý dữ liệu rồi lưu dữ liệu vào cơ sở dữ liệu một. Ứng dụng và trang web sẽ thông qua một API để lấy dữ liệu hoặc yêu cầu điều khiển ngược lại các thiết bị trong nhà thông qua MQTT Broker.

Tổng quan thiết kế của một ổ cắm thông minh không dây:



Hình 2.2: Sơ đồ thiết kế ổ cắm điện thông minh

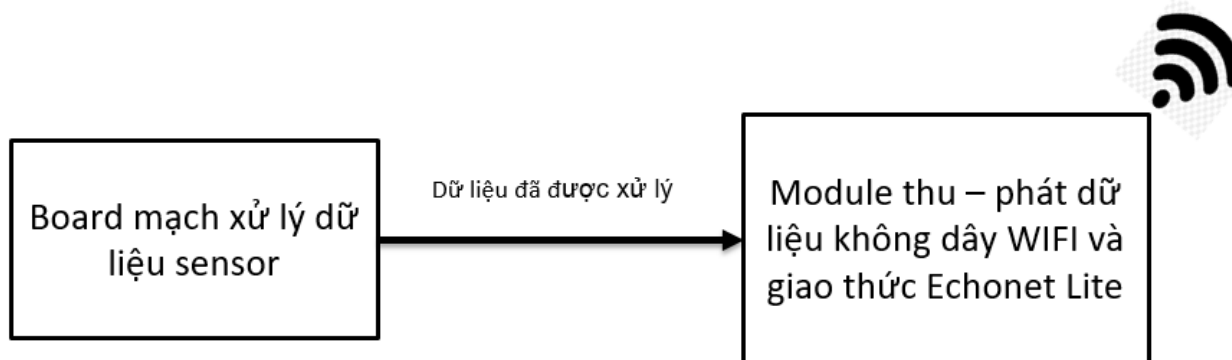
Cảm biến được đấu nối tiếp vào dây nguồn của ổ cắm, board arduino thực hiện đọc dữ liệu đầu ra của sensor rồi chuyển sang thông số cường độ tức thời và công suất tức thời trên dây nguồn cũng là của các thiết bị được cắm trên ổ cắm. Module Wifi lấy dữ liệu công suất của board arduino rồi đóng gói chúng vào các gói tin Echonet Lite để trả về cho Controller khi Controller yêu cầu.



Hình 2.3: Xử lý dữ liệu cường độ dòng điện

Cảm biến hoạt động dựa trên nguyên lý Hall nên OUTPUT của cảm biến là một hiệu điện thế, do đó cần phải có một Board mạch để xử lý chuyển đổi từ dữ liệu trên sang dữ liệu cường độ dòng điện hoặc công suất.

Sau khi thực hiện đo và tính toán dữ liệu bài toán tiếp theo cần xử lý là lựa chọn phương thức truyền dữ liệu đo được cho các Gateway, máy chủ, cloud ... Với tốc độ phát triển chóng mặt và ứng dụng mạnh mẽ của mình, sóng không dây hiện nay xuất hiện trong hầu hết mọi gia đình. Do đó chúng tôi chọn phương thức truyền dữ liệu qua sóng không dây, cụ thể là sóng WiFi. Và ứng dụng giao thức Echonet Lite cho việc khởi tạo, truyền, nhận gói tin. Phương thức gửi dữ liệu đã được mô tả ở hình dưới:



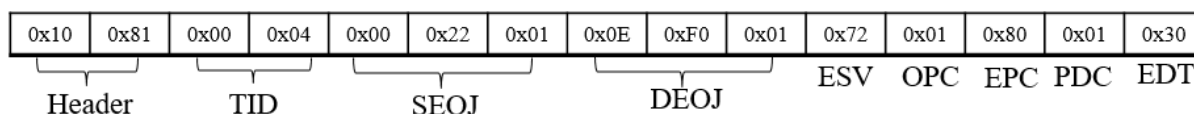
Hình 2.4: Phương thức gửi dữ liệu cường độ dòng điện

Cụ thể hơn module WiFi sẽ đảm nhận nhiệm vụ tham gia vào mạng WiFi trong nhà, sau đó lắng nghe và trả lời những gói tin Echonet Lite cho Node điều khiển trong mạng Echonet. Ổ cắm thông minh sẽ đóng vai trò một Node cảm biến đo dòng điện ở trong mạng, node này có EOJ Code như sau: Group Code: 0x00, Class Code : 0x22, Instance Code : 0x01-0xF7 (số ID của sensor, ví dụ hệ thống có nhiều sensor thì mỗi sensor sẽ có một Instance Code khác nhau). Theo quy định chung của Echonet Lite, Node cảm biến dòng điện chứa rất nhiều thuộc tính, nhưng hệ thống của chúng tôi sẽ chỉ kế thừa một vài thuộc tính thật sự cần thiết cho việc thực hiện dự án này cụ thể^[7]:

- EPC = 0x80 có độ dài 1Byte: Trạng thái của thiết bị với hai giá trị (0x30: Bật và 0x31: Tắt)
- EPC = 0x99 có độ dài 2Byte từ 0x0000 đến 0xFFFF tương đương với 0 đến 65536W: Thể hiện giới hạn công suất của ổ cắm.
- EPC = 0xE1 có độ dài 4Byte từ 0xC4653601 đến 0x3B9AC9FF tương đương với -999,999 999 đến 999,999 999 W: Thể hiện công suất tức thời mà cảm biến đo được

Bên cạnh đó cũng kế thừa một số thuộc tính khác như: vị trí lắp đặt thiết bị, điện thế hoạt động, ...

Dưới đây là một gói tin ví dụ mà thiết bị gửi về khi nhận được yêu cầu lấy trạng thái thiết bị từ node điều khiển

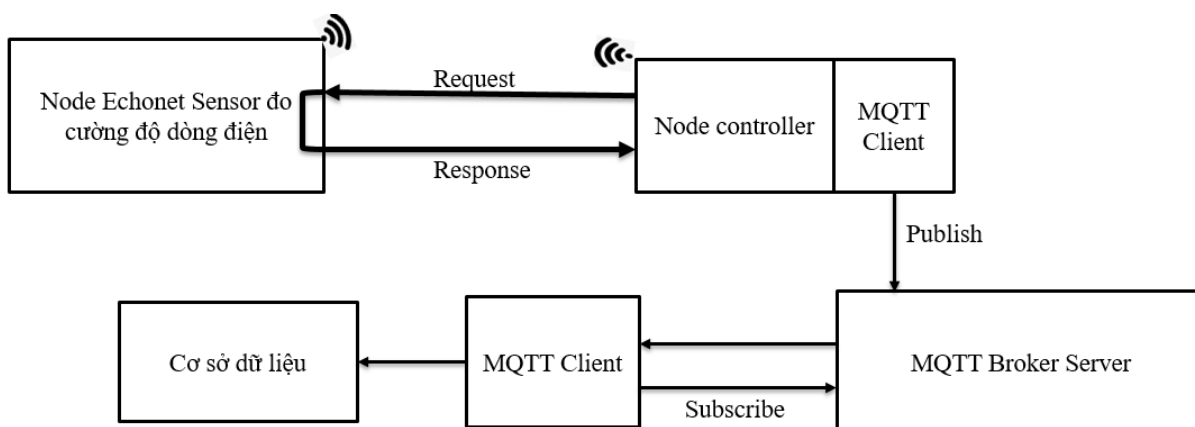


Hình 2. 5: Ví dụ một gói tin trả về trạng thái của thiết bị^[7]

Gói tin trên chứa các thông tin sau:

- Header: Trường này được cố định bởi nhà sản xuất
- TID: ID của phiên giao dịch, ở đây có giá trị 04
- SEOJ: Node gửi, ở đây là Node cảm biến dòng điện
- DEOJ: Node nhận, ở đây là Node điều khiển
- ESV: Loại gói tin, 0x72 = gói tin phản hồi (Response)
- OPC: Số thuộc tính đăng sau mà gói tin chứa (1 thuộc tính)
- EPC: Thuộc tính đầu tiên, ở đây cũng là duy nhất (0x80 = Thuộc tính trạng thái của thiết bị)
- PDC: Độ lớn của thuộc tính, thuộc tính trạng thái được quy định bởi 1Byte nên trường này có giá trị là 0x01
- EDT: Giá trị của thuộc tính, ở đây 0x30 là giá trị thể hiện thiết bị đang ở trạng thái bật

Toàn bộ hệ thống ổ cắm sẽ hoạt động như một node trong mạng lưới các thiết bị Echonet Lite trong ngôi nhà. Node này sẽ thực hiện nghe và trả lời các gói tin yêu cầu từ node Controller được phát triển trong platform IoT của Nguyễn Việt Bắc.



Hình 2. 6: Quá trình thu thập dữ liệu cảm biến

Node điều khiển trong platform có nhiệm vụ thu thập thông tin của các Node khác trong mạng bằng cách gửi những gói tin yêu cầu đến từng node. Sau đó tổng hợp dữ liệu lại rồi thông qua giao thức MQTT publish dữ liệu qua Broker đến Server.

Hình dưới đây mô tả cấu trúc một gói tin yêu cầu trạng thái của một thiết bị:



Hình 2.7: Một gói tin yêu cầu trạng thái của một thiết bị^[7]

Gói tin trên chứa các thông tin sau:

- Header: Trường này được cố định bởi nhà sản xuất
- TID: ID của phiên giao dịch, ở đây có giá trị 04
- SEOJ: Node gửi, ở đây là Node điều khiển
- DEOJ: Node nhận, ở đây là Node cảm biến dòng điện
- ESV: Loại gói tin, 0x62 = gói tin yêu cầu (Request)
- OPC: Số thuộc tính đằng sau mà gói tin chứa (1 thuộc tính)
- EPC: Thuộc tính đầu tiên, ở đây cũng là duy nhất (0x80 = Yêu cầu thuộc tính trạng thái của thiết bị)

Sau khi lấy và tổng hợp được các thuộc tính của một thiết bị, Node điều khiển sẽ thực hiện Publish dữ liệu này lên MQTT Broker Server với topic (chủ đề) được quy định riêng cho từng người dùng “user/X/data” với X là tên đăng kí của người chủ đại diện cho căn nhà, khi đó sẽ có một MQTT Client lắng nghe từ topic này rồi thực hiện lưu dữ liệu vào cơ sở dữ liệu. Quy trình này sẽ được mô tả chi tiết hơn trong IoT Platform của Nguyễn Việt Bắc.

Trong platform, để điều khiển một thiết bị trong nhà, một MQTT client sẽ publish một đoạn dữ liệu có chứa địa chỉ của một Node và các thuộc tính cần cài đặt cho Node đó đến Broker thông qua topic (chủ đề) “user/X/command”. Ví dụ một gói tin yêu cầu tắt một bóng đèn có GroupCode.ClassCode.InstanceCode lần lượt là 0x02.0x90.0x01 chuyển sang trạng thái tắt như sau:

(“group_code”:2,“class_code”:112;“instance_code”:1;“OperationStatus”:false)

2.2.3. Ứng dụng theo dõi điện năng

❖ Chức năng của ứng dụng

Đây là ứng dụng được xây dựng cho người dùng. Cụ thể hơn, người dùng – chủ căn nhà sẽ có Username và Password được cung cấp bởi Platform sau khi đã đăng kí. Mỗi người dùng sẽ được theo dõi ngôi nhà của mình, các thiết bị có trong ngôi nhà đó và các thống kê về việc sử dụng điện trong ngôi nhà. Dựa vào các nhu cầu phổ biến hiện nay của người dùng mà ứng dụng theo dõi sẽ có các tính năng chính như sau:

- Hiện thị điện năng tiêu thụ tức thời của toàn bộ các ổ cắm, điện năng sẽ ở đơn vị Wat
- Hiện thị lượng điện năng đã tiêu thụ (theo ngày, giờ) dựa theo công thức: $W=P.t$ với P là công suất còn t là thời gian sử dụng điện năng.
- Hiện thị chi phí phải thanh toán dựa trên thang tính tiền điện với hộ gia đình tại Việt Nam dựa trên tổng lượng điện năng tiêu thụ trên các thiết bị. Giả sử đã tính được lượng điện năng đã sử dụng tính từ đầu tháng đến thời gian sử dụng là X kWh, sau đó chia X thành các thang và dựa theo thang tính tiền điện dân dụng của nhà nước áp dụng với các hộ gia đình để tính ra số tiền phải thanh toán

TT	Nhóm đối tượng khách hàng	Giá bán điện (đồng/kWh)
1	Giá bán lẻ điện sinh hoạt	
	Bậc 1: Cho kWh từ 0 - 50	1.484
	Bậc 2: Cho kWh từ 51 - 100	1.533
	Bậc 3: Cho kWh từ 101 - 200	1.786
	Bậc 4: Cho kWh từ 201 - 300	2.242
	Bậc 5: Cho kWh từ 301 - 400	2.503
	Bậc 6: Cho kWh từ 401 trở lên	2.587
2	Giá bán lẻ điện sinh hoạt dùng công tơ thẻ trả trước	2.141

Bảng 2. 1: Giá điện dân dụng hiện nay (Nguồn: EVN.com / Biểu giá bán lẻ điện)

- Dự toán chi phí sẽ phải thanh toán đến cuối tháng nếu duy trì việc sử dụng như hiện tại.
- Biểu đồ thống kê lịch sử tiêu thụ điện cho cả ngôi nhà, cho từng ổ cắm hoặc toàn bộ căn nhà.

❖ Thiết kế ứng dụng

Nhằm đơn giản hóa việc xây dựng ứng dụng, chúng tôi đã thiết kế một HTTP API Server, API này nằm trên máy chủ đám mây (Hình 2.1) của hệ thống và có nhiệm vụ trả về cho ứng dụng danh sách các ngôi nhà, thiết bị trong nhà, dữ liệu cảm biến dòng điện, dữ liệu người dùng, ... dưới dạng dữ liệu JSON. Ứng dụng cũng có thể thực hiện bật tắt thiết bị trong nhà nếu người dùng yêu cầu. Bên trong API là chương trình thực hiện truy xuất vào cơ sở dữ liệu cũng như publish dữ liệu lên MQTT Broker Server của Platform.

Sau khi lấy được dữ liệu dòng điện và người dùng thông qua API, ứng dụng sẽ thực hiện một số bài toán xử lý dữ liệu nhằm tính toán điện năng tiêu thụ, tính toán chi phí phải thanh toán ... sau đó hiển thị cho người dùng.

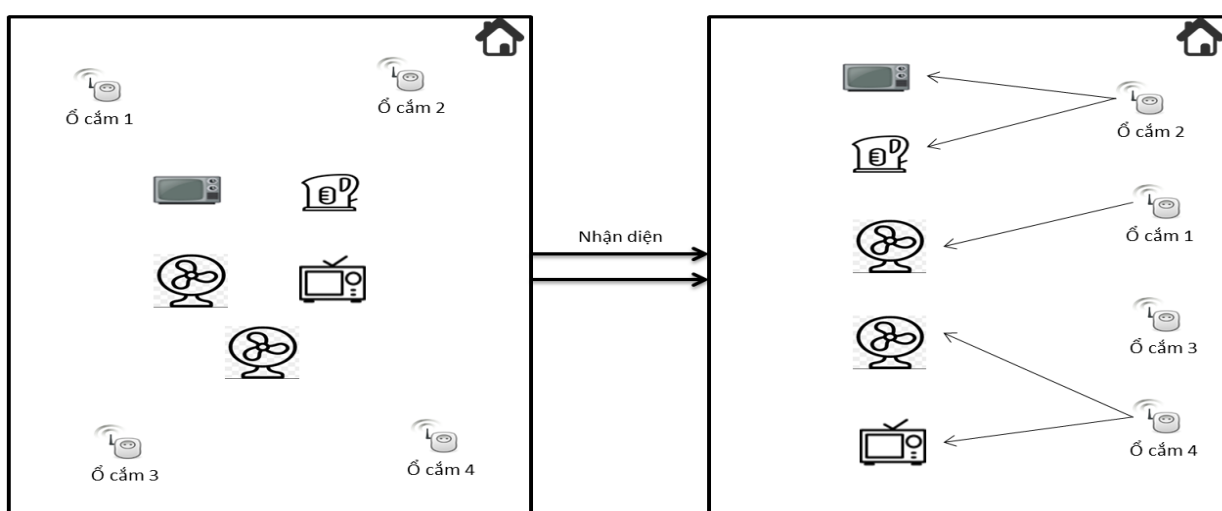
2.3. Chức năng tự động

2.3.1. Nhận dạng thiết bị trên ổ cắm

Mỗi một thiết bị khi hoạt động sẽ được cảm qua một ổ cắm nào đó. Nhận diện xem thiết bị nằm trên ổ cắm nào sẽ giúp cho việc quản lý thiết bị cũng như theo dõi điện năng được tối ưu hơn.

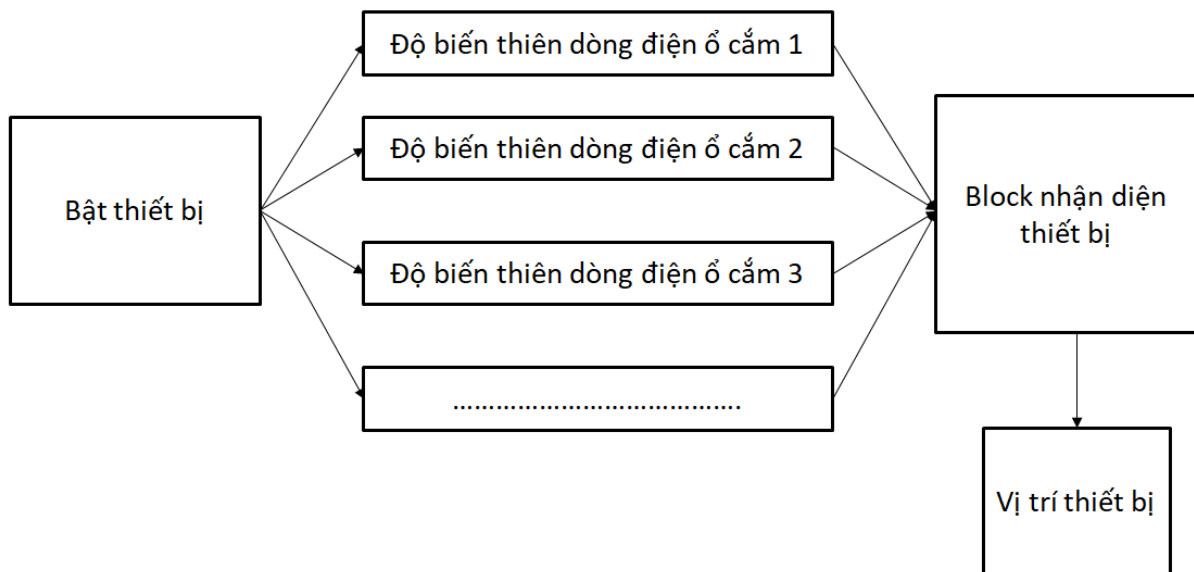
Qua tìm hiểu, có rất nhiều tài liệu viết về cách nhận diện các thiết bị dựa trên đặc trưng biến thiên dòng điện khi thiết bị đó hoạt động. Tuy nhiên việc lấy dữ liệu biến thiên này đòi hỏi quá trình đo có độ trễ cực kì thấp cùng những thiết bị có độ phức tạp và chính xác cao chỉ phục vụ cho việc đo dòng điện. Tuy nhiên với mô hình thiết kế hệ thống ô điện thông minh thì không thể đáp ứng được yêu cầu như vậy.

Do đó để thực hiện chức năng này, chúng tôi chỉ dựa vào sự thay đổi của dòng điện trên từng ổ điện khi một thiết bị được bật lên hoặc tắt đi. Quá trình nhận diện sẽ diễn ra dựa vào thao tác thay đổi trạng thái hoạt động từ bật sang tắt hoặc tắt sang bật của các thiết bị trong ngôi nhà của người dùng.



Hình 2.8: Chức năng nhận diện thiết bị

Dưới đây là quy trình xử lý việc nhận diện thiết bị:



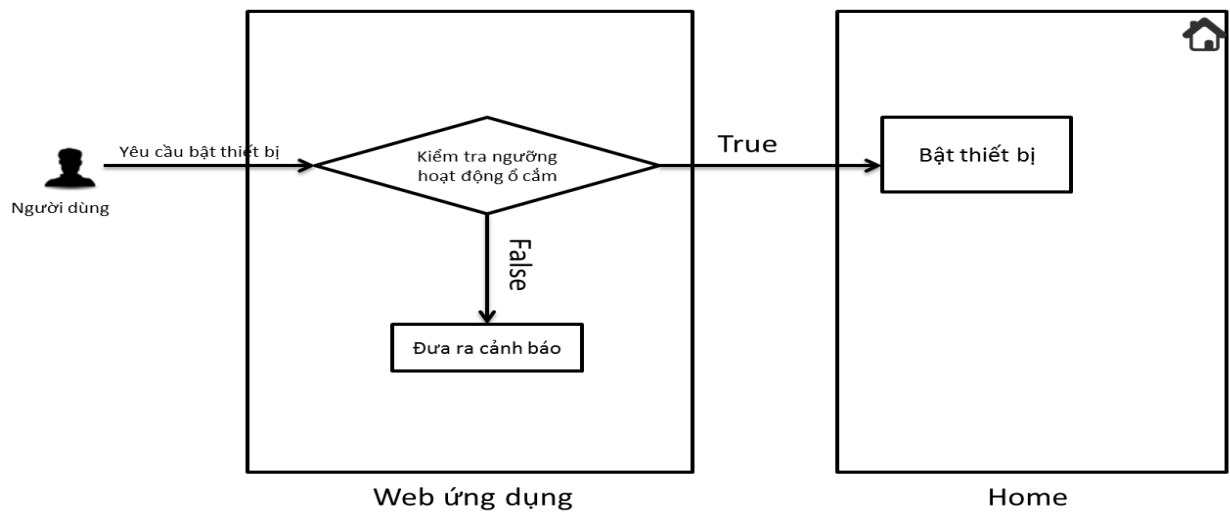
Hình 2. 9: Sơ đồ thuật toán nhận diện thiết bị

Khi người dùng thực hiện yêu cầu lấy vị trí của thiết bị, ứng dụng sẽ thực hiện lệnh đánh dấu các mức công suất hiện tại trên các ổ cắm, sau đó bật hoặc tắt thiết bị cần tìm vị trí rồi thực hiện đánh dấu lần 2 với các mức công suất mới trên các ổ cắm. Sau đó trừ 2 công suất này, nếu chúng có giá trị gần bằng với công suất đã được khai báo của thiết bị thì cập nhật vị trí ổ cắm này vào trong bảng lưu trữ thuộc tính thiết bị trong CSDL.

2.3.2. Kiểm tra ngưỡng hoạt động của ổ cắm khi bật thực hiện bật thiết bị

Mỗi một ổ cắm điện thường chỉ chịu được một mức tải công suất nhất định, việc sử dụng quá nhiều thiết bị có công suất cao trên ổ điện có thể làm quá tải dòng điện dẫn đến chập cháy các thiết bị cũng như chính ổ điện đó.

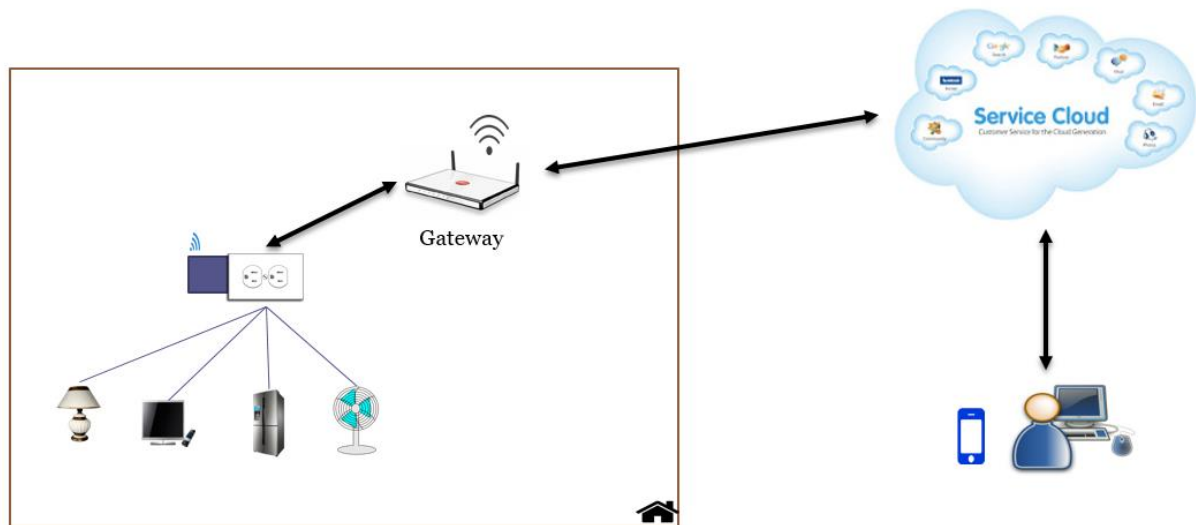
Tại đây, khi người dùng thực hiện một thiết bị đã được nhận dạng ở trên. Hệ thống sẽ thực hiện kiểm tra xem khi thiết bị được bật lên có làm vượt ngưỡng hoạt động của ổ cắm đã được khai báo từ trước hay không. Nếu không làm quá ngưỡng hoạt động mới cho phép thiết bị được bật lên. Ngược lại, không cho phép bật đồng thời đưa ra cảnh báo cho người dùng.



Hình 2.10: Chức năng kiểm tra ngưỡng hoạt động ổ cắm

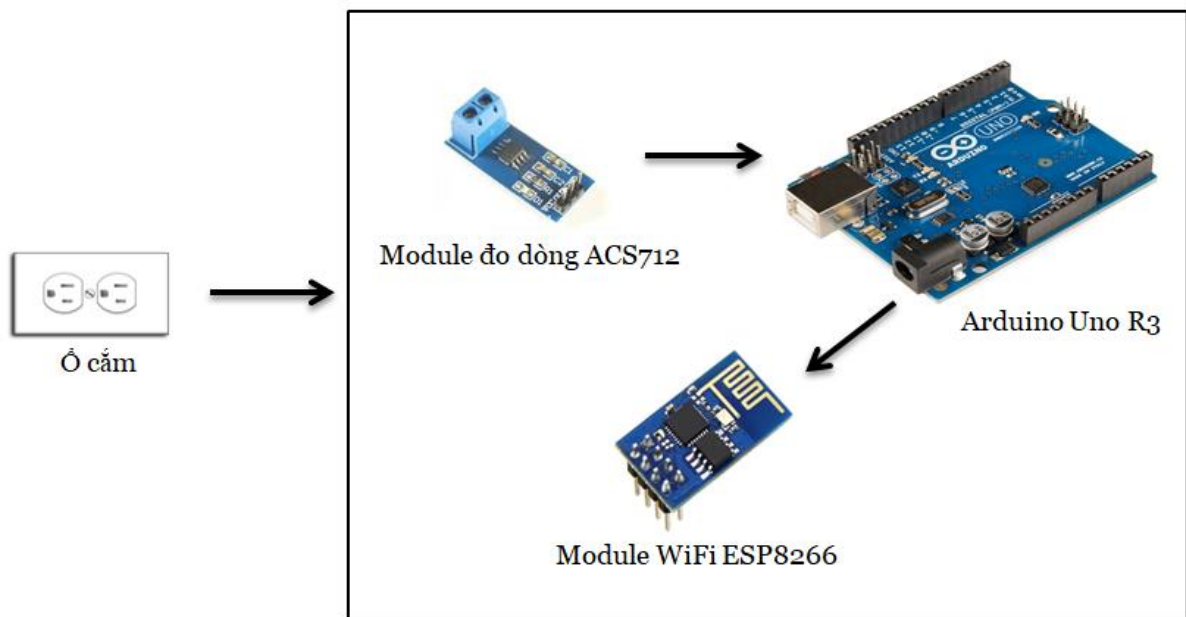
CHƯƠNG 3: TRIỂN KHAI HỆ THỐNG VÀ ĐÁNH GIÁ KẾT QUẢ

3.1. Sơ đồ cài đặt hệ thống thực tiễn



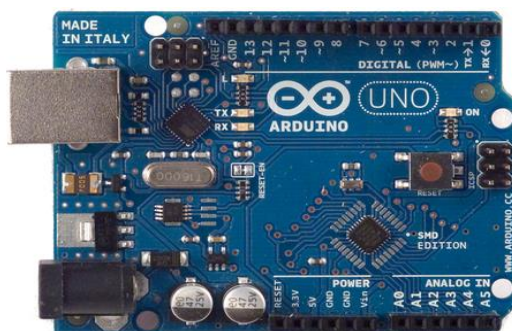
Hình 3. 1: Mô hình tổng quan toàn bộ hệ thống

Các thiết bị cắm trên ổ cắm có thể là các thiết bị Echonet Lite hoặc các thiết bị điện thông thường.



Hình 3. 2: Các thành phần chi tiết của ổ cắm

3.1.1 Board mạch Arduino



Hình 3.3: Board mạch Arduino R3

Đây là một board mạch vi xử lý rất được ưa chuộng trong các dự án IoT hiện nay với thiết kế phù hợp cho người mới tham gia tìm hiểu và cũng phù hợp cho đối tượng nghiên cứu sinh. Arduino giống như một máy tính thu nhỏ có thể được lập trình để làm nhiều việc khác nhau và nó có thể tương tác với các thiết bị điện tử, đèn, cảm biến, động cơ ... Hiện nay, đã có rất nhiều dự án đã được thực hiện với Arduino như: làm máy in 3D, làm robot tự động, điều khiển hệ thống LED quảng cáo ...

Các thiết bị Arduino được lập trình bằng ngôn ngữ riêng. Ngôn ngữ này dựa trên ngôn ngữ Wiring được viết cho phần cứng nói chung mà Wiring lại là một biến thể của C/C++ nên được cộng đồng phát triển gọi chung là ngôn ngữ Arduino. Đây là một ngôn ngữ trực quan, đọc hiểu rất đơn giản giúp một người mới lập trình cũng có thể nắm bắt được nội dung của code. Để lập trình cho Arduino, nhóm phát triển thiết bị này đã cung cấp cho người dùng một môi trường lập trình Arduino được gọi là Arduino IDE (Integrated Development Environment).

Dưới đây là một vài thông số chi tiết của thiết bị:

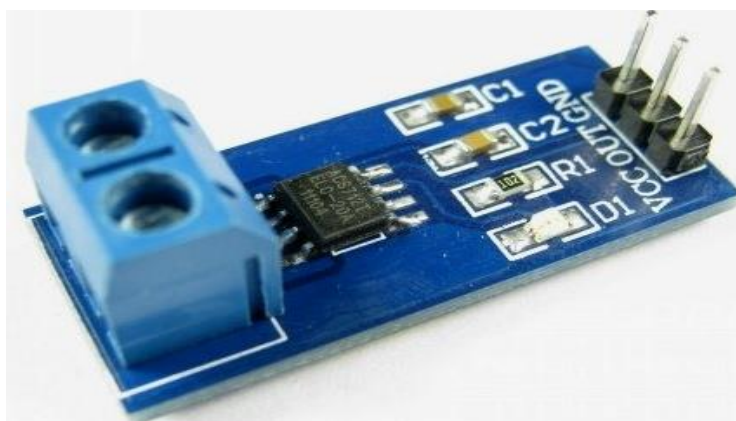
Vi điều khiển	ATmega328 8bit
Điện áp hoạt động	5V DC
Tần số hoạt động	16 MHz
Dòng tiêu thụ	Khoảng 30 mA
Điện áp vào khuyến dùng	7-12V DC
Điện áp giới hạn	6-20V DC
Số chân Digital I/O	14 (6 chân hardware PWM)
Số chân Analog	6 (Độ phân giải 10 bit)
Dòng tối đa trên mỗi chân I/O	30 mA
Dòng ra tối đa (5V)	500 mA
Dòng ra tối đa (3.3V)	50 mA
Bộ nhớ Flash	32 KB
SRAM	2 KB
EEPROM	1 KB

Bảng 3. 1: Tổng quan các thông số Arduino

Trong dự án này, Board arduino có vai trò tính toán và xử lý dữ liệu đo điện từ cảm biến, sau đó sẽ chuyển dữ liệu qua cho module WiFi. Bên cạnh đó Arduino còn cung cấp nguồn hoạt động cho hai thiết bị kể trên.

3.1.2 Module đo dòng ACS712

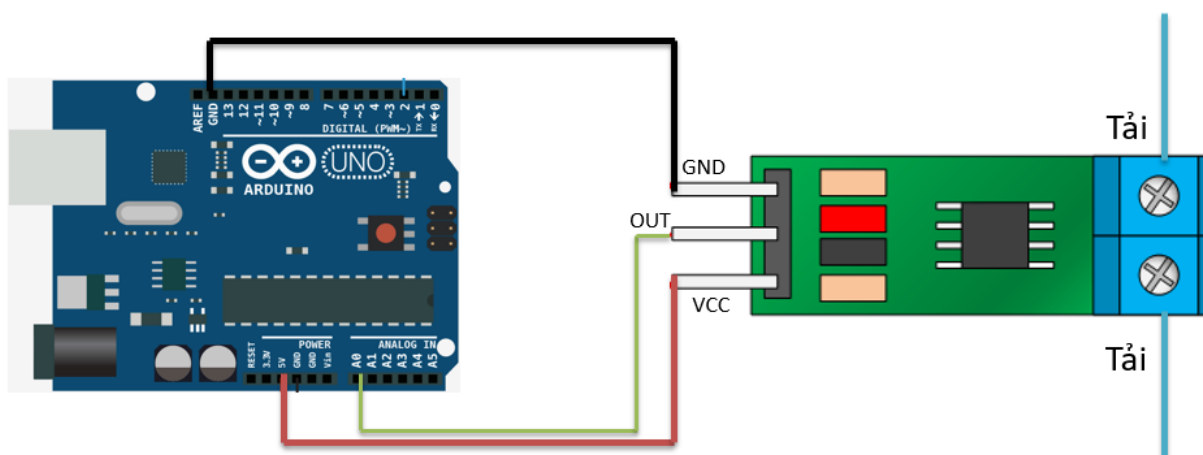
Như đã trình bày ở chương 1, hiệu ứng Hall được ứng dụng rất nhiều trong các thiết bị đo điện điện tử. Xét trên nhu cầu cũng như quy mô của dự án, chúng tôi đã lựa chọn module đo dòng điện ACS712 cho hệ thống của mình.



Hình 3. 4: Modul đo dòng ACS712

Thiết bị có 3 đầu ra:

1. GND (-): Dây nối đất ground
2. VCC (+): Nguồn để tạo dòng chạy trong thanh Hall
3. OUT: Đầu ra của hiệu điện thế Hall. Hiệu điện thế biến đổi tuyến tính dựa theo biến đổi của dòng điện qua thanh Hall.



Hình 3. 5: Sơ đồ nối chân Arduino và ACS712

OUTPUT của ACS712 là một hiệu điện thế biến thiên trong khoảng 0 - 5V, trong đó đối với module 20A thì độ nhạy của thiết bị $\sim 100\text{mV/A}$ (Tức là cứ 100mV thay

đổi thì tương ứng với dòng điện trên tải đã thay đổi 1A) tương tự với module 5A độ nhạy là 185mV/A và đối với module 30A là 66mV/A. Arduino sẽ đọc hiệu điện thế này tại chân Analog A0 với độ lớn từ 0-5V tương đương với giá trị số được xuất ra bằng hàm `Analog.read()` trên Arduino trong khoảng 0-1024. (Ví dụ khi dòng điện vào chân A0 có độ lớn là 2.5V thì hàm `Analog.read()` cho ta giá trị 512). Và khi không có tải, giá trị V_{OUT} của cảm biến luôn là 2.5V. Do đó chúng ta xây dựng được công thức tính cường độ dòng điện chạy qua đoạn dây như sau:

$$I = (\text{Analog.Read}(A0) - 512) * (5 / 1024) * (1000 / 100) \text{ (Ampe)}[10]$$

Sau khi tính được cường độ dòng điện qua đoạn dây, ta tính công suất của dòng điện dựa trên công thức:

$$P = U \times I \text{ (Wat)}$$

Trong đó ta coi $U = 220V$ là giá trị hiệu điện thế của nguồn điện dân dụng trong gia đình.

Sau khi tính được giá trị công suất tức thời của ổ cắm, ta phải thực hiện chuyển giá trị này sang một mảng Byte có độ dài là 4. Do trong Echonet Lite, với Node cảm biến dòng điện, giá trị công suất được quy định bởi 4Bytes.

3.1.3 Module WiFi ESP8266

Module WiFi ESP8266 là một module đơn giản sử dụng sóng WiFi để truyền dữ liệu trong các dự án IoT. Module được xây dựng trên chip ESP8266 SoC (System on Chip) của hãng Espressif. Do đó chúng ta cũng có thể nạp chương trình cho chip này như trên Arduino.

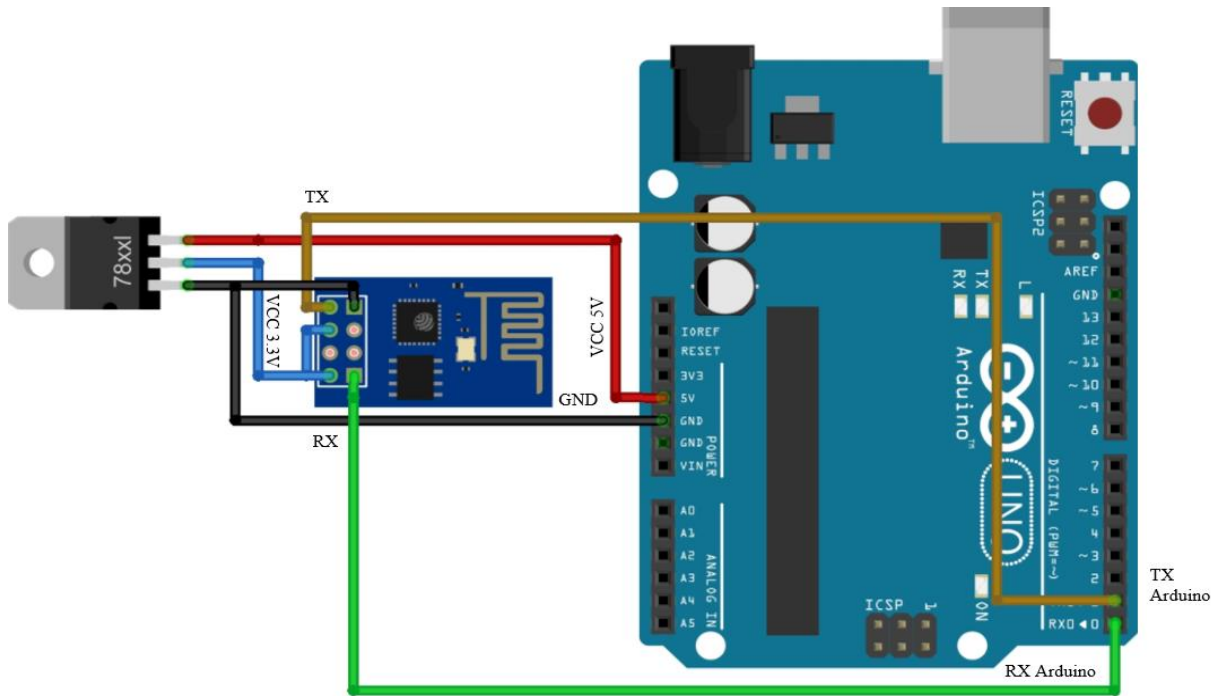


Hình 3. 6: Module Wifi ESP8266 V1

Các tính năng chính:

1. Hỗ trợ chuẩn 802.11 b/g/n
2. WiFi 2.4 GHz, hỗ trợ WPA/WPA2
3. Chuẩn điện áp hoạt động 3.3V DC
4. Chuẩn giao tiếp kết nối UART với tốc độ Baud lên đến 115200
5. Có 3 chế độ hoạt động: Access Point, Client và Both Client & Access Point
6. Hỗ trợ hầu hết các chuẩn bảo mật phổ biến hiện nay

7. Hỗ trợ cả hai giao tiếp TCP và UDP
8. Làm việc như một máy chủ có thể kết nối tối đa với 5 máy con



Hình 3. 7: Sơ đồ nối chân Arduino và ESP8266

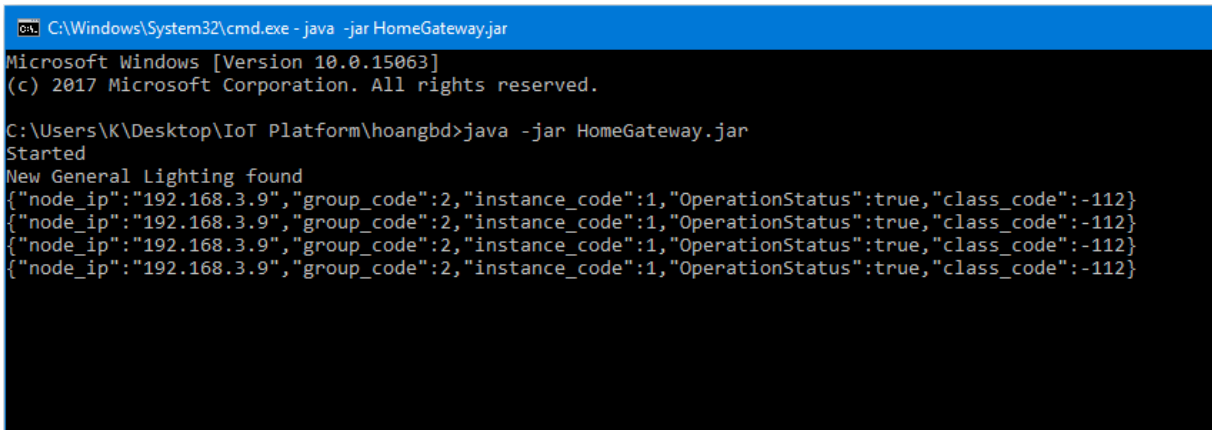
Do ESP8266 hoạt động ở mức điện áp 3.3V nên ta cần một IC nguồn để đổi dòng từ 5V xuống 3.3V để cấp nguồn cho ESP8266. Thực hiện nối hai chân TX – RX của ESP8266 với hai chân TX - RX trên Arduino, khi đó ESP8266 có thể nhận dữ liệu từ chân TX của Arduino, dữ liệu này chính là 4Bytes đã được chuyển đổi từ công suất tức thời tính được từ cảm biến dòng. Sau đó tiến hành nạp chương trình lên Arduino và ESP8266, như vậy là đã hoàn thành cơ bản một Node ở cảm Echonet Lite^[9].

3.1.4 Cài đặt Platform IoT

3.1.4.1 Cài đặt HomeGateway^[1]

Đây là một chương trình java được cung cấp bởi Platform, chương trình có thể chạy trên một máy tính ở trong ngôi nhà của người dùng hoặc một máy tính nhúng Raspberry Pi. Chương trình được khởi chạy với tham số để kết nối đến MQTT Broker lần lượt là: hostname, mqttusername, mqttpassword trong đó (hostname: địa chỉ của MQTT Broker, mqttusername và mqttpassword là tên đăng nhập và mật khẩu để đăng nhập vào mqtt broker). Khi khởi chạy, chương trình sẽ tìm kiếm các Node Echonet Lite có trong mạng gia đình, sau đó publish thông tin của các Node này lên MQTT Broker.

Ví dụ khởi chạy HomeGateway cho ngôi nhà của Bùi Đăng Hoàng:



```
C:\Windows\System32\cmd.exe - java -jar HomeGateway.jar
Microsoft Windows [Version 10.0.15063]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\K\Desktop\IoT Platform\hoangbd>java -jar HomeGateway.jar
Started
New General Lighting found
{"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
{"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
{"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
{"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
```

Hình 3. 8: Cài đặt Homegateway

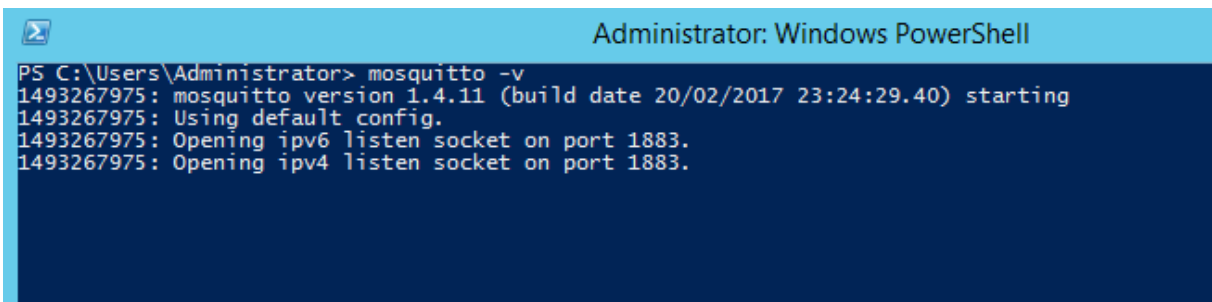
Chương trình sẽ tìm kiếm các Node Echonet Lite trong mạng sau đó tổng hợp một số thuộc tính sau đó publish dữ liệu lên MQTT Broker dưới dạng chuỗi JSON. Như hình trên: Chương trình đã tìm thấy một Node General Lighting (GroupCode.ClassCode.InstanceCode = 2.-112.1) sau đó publish lên MQTT Broker có địa chỉ tên miền “hoangbd.info” cho chủ đề “user/hoangbd/data” với dữ liệu có dạng:

**{“node_ip”:192.168.3.9,“group_code”:2,“instance_code”:1;
“OperationStatus”:true;“class_code”:-112}**

3.1.4.2 Cài đặt Server Cloud

Hiện tại hệ thống đang hoạt động trên một Cloud Computer Windows có địa chỉ IP là **52.77.245.199** với domain “hoangbd.info”. Tại đây đã được cài đặt mosquitto broker, MySQL Server với cơ sở dữ liệu được cung cấp bởi Platform.

Đầu tiên, khởi chạy mosquitto Broker Server:

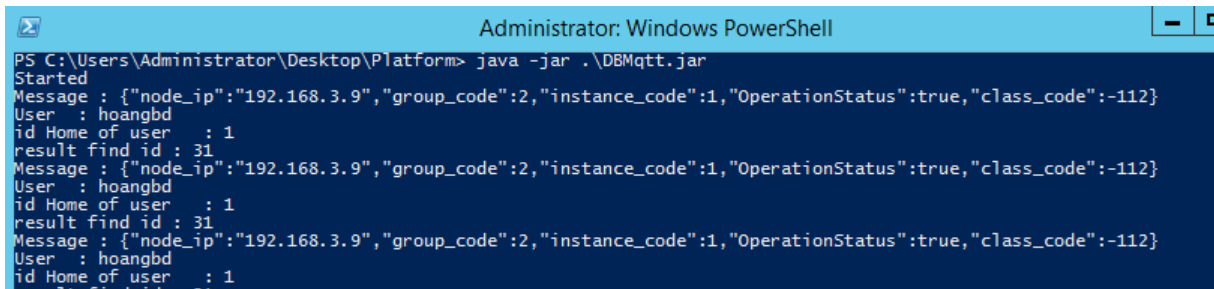


```
Administrator: Windows PowerShell
PS C:\Users\Administrator> mosquitto -v
1493267975: mosquitto version 1.4.11 (build date 20/02/2017 23:24:29.40) starting
1493267975: Using default config.
1493267975: Opening ipv6 listen socket on port 1883.
1493267975: Opening ipv4 listen socket on port 1883.
```

Hình 3. 9: Cài đặt Mosquitto Broker

Sau đó khởi chạy chương trình của Platform phía server với các tham số MQTT Broker host, MySQL username/password và MQTT username/password:

“java –jar DBMqtt.jar 127.0.0.1 iot_user 1 123456 1 hoangbd 123456”



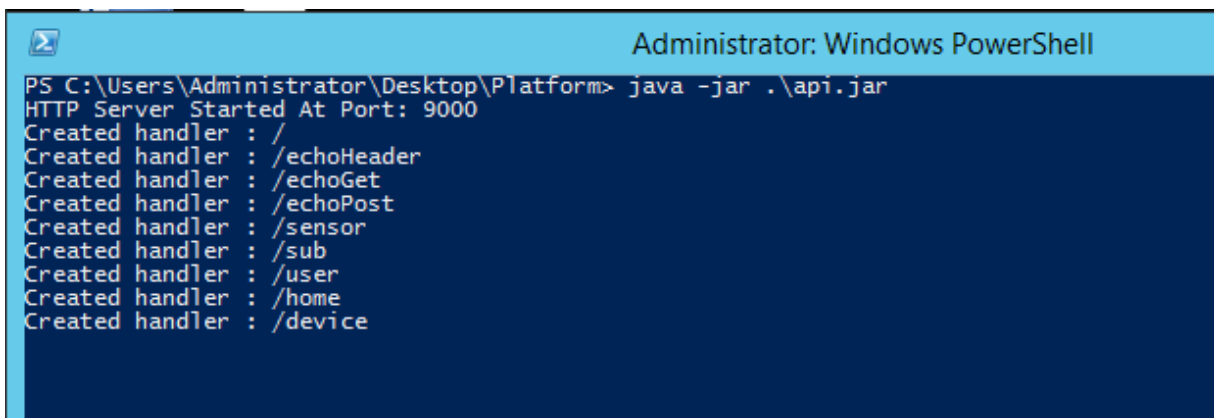
```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Desktop\Platform> java -jar .\DBMQtt.jar
Started
Message : {"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
User : hoangbd
id Home of user : 1
result find id : 31
Message : {"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
User : hoangbd
id Home of user : 1
result find id : 31
Message : {"node_ip":"192.168.3.9","group_code":2,"instance_code":1,"OperationStatus":true,"class_code":-112}
User : hoangbd
id Home of user : 1
```

Hình 3. 10: Cài đặt Platform Server Side

Chương trình trên đóng vai trò như một MQTT client đăng kí theo dõi chủ đề “user/hoangbd/data” từ MQTT Broker, khi có một dữ liệu được publish lên Broker, client này sẽ nhận được, sau đó xử lý và thêm hoặc cập nhật dữ liệu thiết bị vào trong cơ sở dữ liệu.

3.1.4.3 Cài đặt HTTP API

API là một chương trình java, chương trình sẽ thực hiện khởi tạo một HTTP Server tại cổng 9000 của Server. API này sẽ trả về dữ liệu dưới dạng JSON khi người dùng gửi request. Khởi chạy:



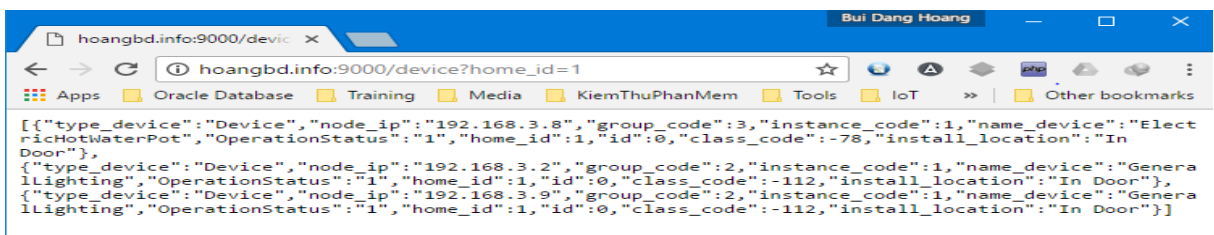
```
Administrator: Windows PowerShell
PS C:\Users\Administrator\Desktop\Platform> java -jar .\api.jar
HTTP Server Started At Port: 9000
Created handler : /
Created handler : /echoHeader
Created handler : /echoGet
Created handler : /echoPost
Created handler : /sensor
Created handler : /sub
Created handler : /user
Created handler : /home
Created handler : /device
```

Hình 3. 11: Cài đặt HTTP API

Ví dụ, khi ứng dụng muốn lấy danh sách các thiết bị trong một ngôi nhà sẽ thực hiện gửi một HTTP Request đến API như sau:

“http://hoangbd.info:9000/device?home_id=1”

Khi đó API sẽ trả về danh sách các thiết bị dưới dạng JSON:

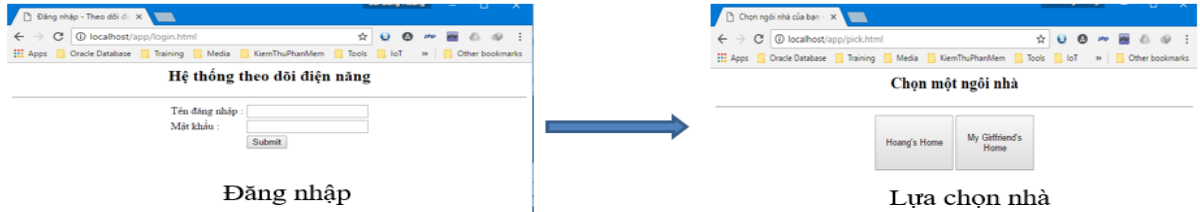


```
hoangbd.info:9000/device?home_id=1
[{"type_device": "Device", "node_ip": "192.168.3.8", "group_code": 3, "instance_code": 1, "name_device": "ElectrichHotWaterPot", "OperationStatus": "1", "home_id": 1, "id": 0, "class_code": -78, "install_location": "In Door"}, {"type_device": "Device", "node_ip": "192.168.3.2", "group_code": 2, "instance_code": 1, "name_device": "GeneralLighting", "OperationStatus": "1", "home_id": 1, "id": 0, "class_code": -112, "install_location": "In Door"}, {"type_device": "Device", "node_ip": "192.168.3.9", "group_code": 2, "instance_code": 1, "name_device": "GeneralLighting", "OperationStatus": "1", "home_id": 1, "id": 0, "class_code": -112, "install_location": "In Door"}]
```

Hình 3. 12: Danh sách thiết bị được trả về bởi API

3.1.5 Trang web ứng dụng theo dõi

- ✓ Đăng nhập: Người sử dụng phải thực hiện đăng nhập vào hệ thống, sau đó hệ thống sẽ hiển thị danh sách các ngôi nhà mà người dùng đó đang quản lý. Nếu đăng nhập thành công, người dùng thực hiện chọn một căn nhà để vào trang chủ theo dõi.



Hình 3.13: Đăng nhập và lựa chọn căn nhà

Tiếp theo là giao diện của trang chủ:

Quản lý điện năng

Xin chào : **hoangbd!** [Đăng xuất](#) [Thay đổi ngôi nhà](#)

Danh sách các thiết bị

IP	ID	Group Code	Class Code	Instance Code	Tên thiết bị	Trạng thái	Location	Function
192.168.0.7	53	2	-112	1	GeneralLighting	OFF	Detected (55)	Turn ON
192.168.0.6	54	3	-78	1	ElectricHotWaterPot	OFF	Detected (55)	Turn ON
192.168.0.3	55	0	34	1	electricenergysensor			

Thống kê sử dụng điện

Hiện tại đang sử dụng : 64.37 W
Tổng lưu lượng điện năng đã sử dụng : 0.08 kWh
Số tiền phải thanh toán : 118.72 VNĐ

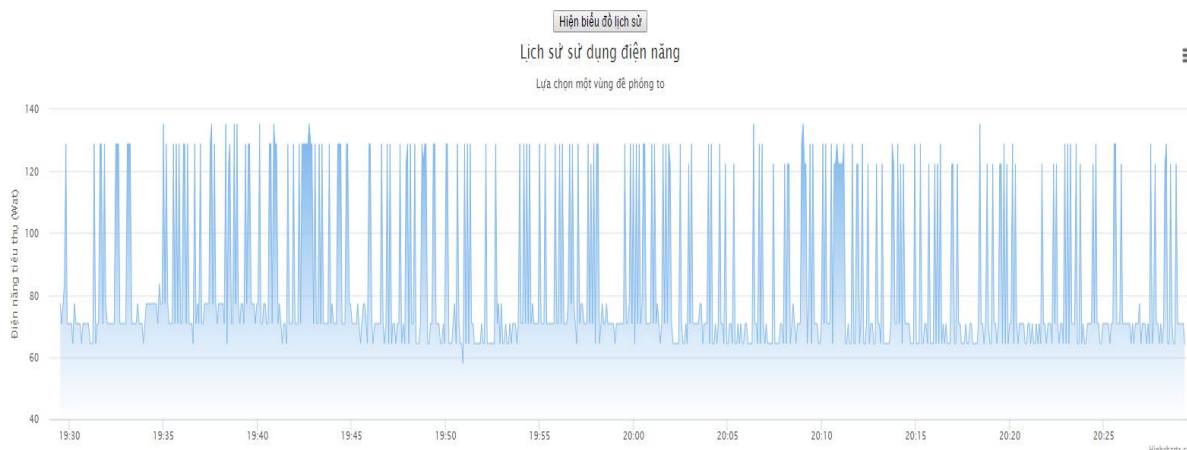
[Hiện biểu đồ lịch sử](#)

Hình 3.14: Trang chủ theo dõi

Tại trang chủ, cung cấp cho người dùng một số thông tin như sau:

- ✓ Theo dõi công suất của các ổ cắm điện trong thời gian thực
- ✓ Hiện thị tổng lượng điện năng đã tiêu thụ
- ✓ Hiện thị số tiền phải thanh toán
- ✓ Dự tính số tiền phải thanh toán

Người dùng có thể chọn xem lịch sử sử dụng điện năng dưới dạng biểu đồ theo thời gian để có thể biết được chi tiết hơn việc sử dụng điện của mình đã được thực hiện như thế nào để có thể đưa ra biện pháp điều chỉnh phù hợp hơn.



Hình 3. 15: Lịch sử sử dụng điện năng

3.1.6 Các chức năng tự động

- Chức năng nhận diện thiết bị trên ổ cắm

Đây là chức năng giúp nhận diện xem thiết bị đang hoạt động trên ổ cắm nào. Mặc định một thiết bị khi chưa xác định được vị trí sẽ có “Location = Not Detected”. Mỗi khi người dùng thực hiện bật hoặc tắt một thiết bị nào đó, chương trình sẽ theo dõi sự thay đổi công suất của các ổ điện, nếu ổ điện nào có công suất mới lệch bằng hoặc sắp xỉ với công suất của thiết bị tức là thiết bị đang nằm trên ổ cắm đó. Càng thực hiện kiểm tra nhiều lần thì độ chính xác của phép tính càng cao.

Trường hợp người dùng có thể tự ra lệnh để nhận diện thủ công bằng cách ấn nút “Find Location” trên trang chủ rồi thực hiện bật/tắt thiết bị bằng tay.

Tên	Trạng thái	Location	Function	
ElectricHotWaterPot	ON	Detected	Turn OFF	Find location
GeneralLighting	ON	Detected	Turn OFF	Find location
GeneralLighting	ON	Detected	Turn OFF	Find location
electricenergysensor	ON	Not Detected	Turn OFF	Find location

Hình 3. 16: Thực hiện nhận diện một thiết bị

Khi thực hiện lệnh “Find Location”, ứng dụng sẽ gửi lệnh bật một thiết bị đồng thời kèm theo yêu cầu thực hiện theo dõi sự biến thiên của dòng điện trên tất cả các ổ cắm. Sau đó kiểm tra xem ổ cắm nào có lượng điện năng biến thiên gần nhất với công suất của thiết bị thì tiến hành cập nhật vị trí của thiết bị vào trong CSDL.

- Chức năng kiểm tra ngưỡng hoạt động của ổ cắm

Chức năng này chỉ hoạt động trên những thiết bị đã xác định được Location tức đã biết thiết bị đang hoạt động trên ổ cắm nào, khi đó nếu người dùng thực

hiện bật thiết bị, chương trình kiểm tra sẽ được gọi. Chương trình này có nhiệm vụ kiểm tra xem nếu thiết bị bật lên có làm quá ngưỡng tải của ổ cắm chứa nó hay không. Nếu có chương trình sẽ trả về “False” và đưa ra cảnh báo cho người dùng, nếu không việc thực hiện bật thiết bị được diễn ra bình thường.

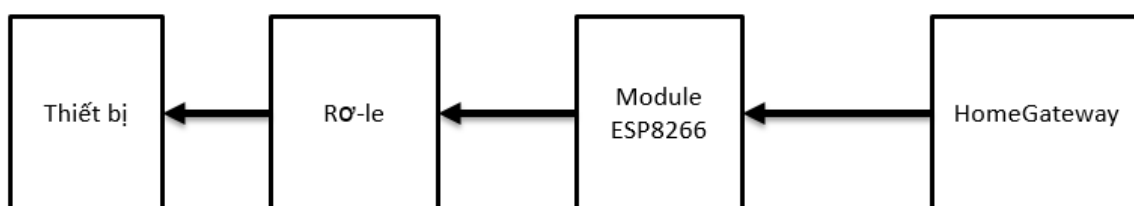
3.1.7 Giả lập một số thiết bị Echonet Lite

Do các thiết bị Echonet Lite chưa được phát hành và bày bán tại Việt Nam, nên để thực hiện chạy thử các chức năng của ổ cắm thông minh, chúng tôi phải tiến hành tạo ra những thiết bị Echonet Lite thật trong gia đình.

Các thiết bị Echonet Lite này cũng có một số thành phần giống với ổ cắm Echonet Lite như sau:

- 1 rơ-le để điều khiển đóng ngắt nguồn điện cho thiết bị
- 1 Module WiFi ESP8266 để kế thừa và tạo ra một node Echonet Lite
- Module WiFi này cũng đóng nhiệm vụ điều khiển rơ-le khi nhận được lệnh.

Dưới đây là sơ đồ của một Node Echonet Lite mà chúng tôi đã tạo ra:



Hình 3. 17: Sơ đồ một Node Echonet Lite

Hiện tại có hai thiết bị Echonet Lite đó là bóng đèn (General Lighting – GroupCode = 0x02, ClassCode = 0x90, InstanceCode = 0x01) và ấm đun nước nóng (Electric Hot Water Pot – GroupCode = 0x03, ClassCode = 0xB2, InstanceCode = 0x01). Trên mỗi node thiết bị này chứa 2 thông tin : Trạng thái thiết bị và công suất thiết kế của thiết bị.

Chúng tôi sẽ thực hiện đo và đánh giá kết quả hệ thống với hai thiết bị này.

3.2 Đánh giá kết quả thử nghiệm

Sau khi tiến hành lắp đặt, hệ thống đã được đưa vào vận hành với hai thiết bị demo là bóng điện và một ấm đun nước điện được sử dụng trung bình 1h/ngày. Bên cạnh đó là một ổ đo điện thông minh. Chúng tôi đánh giá kết quả qua thang điểm như sau:

Các mức đánh giá: 5: Rất tốt; 4: Tốt; 3: Khá; 2: Trung bình; 1: Không hoạt động

Chức năng	Kết quả
Đo công suất ổ điện	Với một ấm đun nước nóng (Công suất ghi trên vỏ ấm 2000W). Sau 10 lần đo, cho giá trị trung bình là 1800W sai số ~ 7%. Đây là kết quả tạm chấp nhận được do hiện nay chưa có công cụ chuẩn để đo công suất hoạt động của một thiết bị.
Gửi - Nhận dữ liệu giữa Node ổ cắm và HomeGateway	Quá trình gửi – nhận không phát sinh lỗi. Thời gian từ lúc dữ liệu bắt đầu được gửi đến lúc nhận ~ 0,1s
Gửi – Nhận dữ liệu giữa HomeGateway và Server	Quá trình gửi – nhận không phát sinh lỗi. Thời gian từ lúc dữ liệu bắt đầu được gửi đến lúc nhận ~ 0,5s (Với Cloud Server Amazone)
Trang web theo dõi điện năng	Chạy ổn định, không lỗi
Chức năng nhận diện thiết bị	Quá trình nhận diện hoàn thành trong ~ 5s, sai số càng cao với các thiết bị công suất nhỏ
Chức năng ngưỡng hoạt động thiết bị	Hoạt động tốt

Bảng 3. 2: Đánh giá kết quả chạy thử nghiệm

Kết luận đánh giá: Sau một thời gian chạy thử nghiệm, hệ thống tuy vẫn còn một vài vấn đề như xuất hiện sai số trong phép đo, chưa xử lý được các trường hợp đặc biệt như mất mạng, mất điện (mỗi lần gặp lỗi là phải khởi chạy lại toàn bộ hệ thống). nhưng nhìn chung vẫn hoạt động và đáp ứng được mục tiêu của khóa luận.

CHƯƠNG 4: TỔNG KẾT

4.1. Kết quả nghiên cứu đạt được

Việc tồn tại một thiết bị đo điện trong ngôi nhà thông minh là hết sức cần thiết. Đề tài nghiên cứu và phát triển một thiết bị thông minh trong mạng lưới các thiết bị IoT là một đề tài còn đang rất mới mẻ, nó mang lại cho chúng ta cái nhìn chi tiết hơn về cách thức hoạt động của các thiết bị trong mạng lưới. Bên cạnh đó là khả năng ứng dụng và phát triển không giới hạn các thiết bị trong đời sống hiện đại của con người sau này.

Đề tài khóa luận cũng tập trung nghiên cứu về giao thức Echonet Lite, hiện đã có thể kế thừa giao thức từ các gói tin Bytes. Làm tiền đề cho các nghiên cứu sau này.

Cho ra được một sản phẩm IoT hoàn chỉnh và cung cấp được nền tảng ứng dụng cho sản phẩm này.

4.2. Những hạn chế trong khi thực hiện đề tài

Độ chính xác khi thực hiện đo công suất của các thiết bị chưa được cao, nhất là với các thiết bị có công suất nhỏ.

Hiện đề tài mới chỉ dừng ở mức nghiên cứu sản phẩm và demo chức năng nên các thiết bị được kết nối với nhau qua hệ thống dây nối nhỏ, chưa được gọn gàng

Việc thiếu thiết bị để thực hiện thử nghiệm sản phẩm. Hiện nay mới có 3 thiết bị điện Echonet Lite (Bóng đèn, Quạt điện, Âm điện) và một ổ đo điện thông minh. Việc thử nghiệm sẽ hiệu quả hơn nếu có nhiều thiết bị

4.3. Đề xuất, hướng phát triển

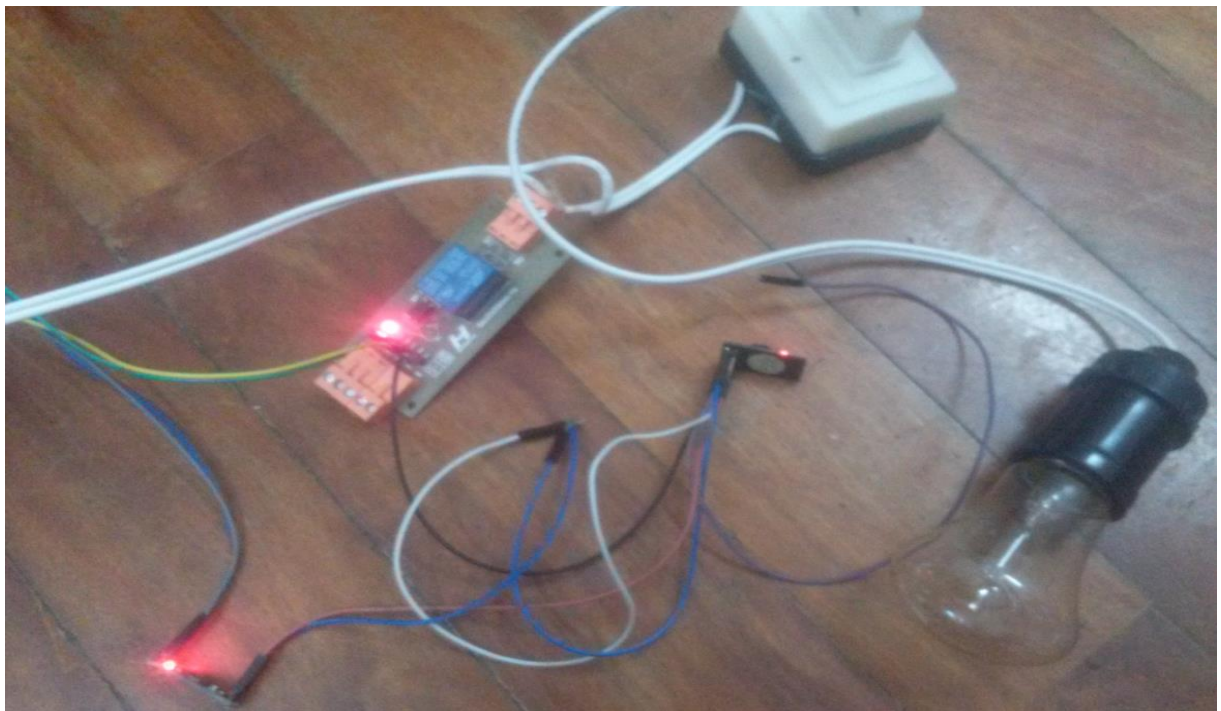
Thu gọn thiết bị lại trên một bảng mạch hoặc một box ổ điện

Đưa hệ thống vào một ngôi nhà thật sự

PHỤ LỤC HÌNH ẢNH THỰC TẾ CỦA HỆ THỐNG



Hình phụ lục: Ổ cắm thông minh



Hình phụ lục: Bóng đèn Echonet Lite

TÀI LIỆU THAM KHẢO

Tiếng Việt

- [1] “Khóa luận tốt nghiệp: Xây dựng IoT Platform cho nhà thông minh”, Nguyễn Việt Bắc – ĐHCN ĐHQG Hà Nội
- [2] “Giáo trình Từ học”, Lưu Tuấn Tài (2008), NXBĐHQG, Hà Nội
- [3] “Giáo trình cảm biến” Phan Quốc Phô (chủ biên), Nguyễn Đức Chiến NXBKHKHT

Tiếng Anh

- [4] “Device democracy Saving the future of the Internet of Things”, IBM Institute for Business Value
- [5] “A gap analysis of Internet-of-Things platforms”, Julien Minerauda, Oleksiy Mazhelisb, Xiang Suc, Sasu Tarkomaa
- [6] “ECHONET Lite Specification 5 Chapters”, ECHONET Consortium
- [7] “APPENDIX, Detailed Requirements for ECHONET Device Objects”, ECHONET Consortium
- [8] “OASIS MQTT Version 3.1.1”, OASIS Standard 29 October 2014
- [9] “ESP8266 Datasheet English Version 4.3”, Espressif Systems IOT Team
- [10] “ACS712 Datasheet”, Allegro MicroSystems