

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA KHOA HỌC & KỸ THUẬT MÁY TÍNH**



LUẬN VĂN TỐT NGHIỆP ĐẠI HỌC

**PHÁT TRIỂN BOARD ĐIỀU KHIỂN VÀ
HỆ THỐNG HỖ TRỢ NÔNG NGHIỆP ĐÔ THỊ**

HỘI ĐỒNG: Công nghệ phần mềm

GVHD: ThS. Nguyễn Cao Trí

GVPB: ThS. Lê Đình Thuận

SVTH 1: Huỳnh Bá Thạch (51303742)

SVTH 2: Tạ Chí Tây (51303574)

SVTH 3: Nguyễn Đình Dũng (51300665)

SVTH 4: Nguyễn Lê Minh Khôi (51301906)

TP Hồ Chí Minh, 01/2018

LỜI CAM ĐOAN

Tôi xin cam đoan đây là công trình nghiên cứu và thực hiện của nhóm và được sự hướng dẫn của ThS. Nguyễn Cao Trí. Các nội dung nghiên cứu, hiện thực trong đề tài này là trung thực và chưa được công bố dưới bất kỳ hình thức nào trước đây. Những thông tin phục vụ cho việc phân tích, nhận xét, đánh giá được chính nhóm thu thập từ các nguồn khác nhau có ghi rõ trong phần tài liệu tham khảo. Nếu phát hiện có bất kỳ sự gian lận nào nhóm xin hoàn toàn chịu trách nhiệm về mặt nội dung luận văn tốt nghiệp của mình. Trường đại học Bách Khoa – Đại Học Quốc Gia Thành Phố Hồ Chí Minh không liên quan đến những vi phạm tác quyền, bản quyền do nhóm gây ra trong quá trình thực hiện (nếu có).

LỜI CẢM ƠN

Trên thực tế không có sự thành công nào mà không gắn liền với những sự hỗ trợ, giúp đỡ dù ít hay nhiều, dù trực tiếp hay gián tiếp của người khác. Trong suốt thời gian từ khi bắt đầu học tập ở trường đại học Bách Khoa – Đại Học Quốc Gia Thành Phố Hồ Chí Minh cho đến nay, nhóm chúng tôi đã nhận được rất nhiều sự quan tâm, giúp đỡ của quý thầy, cô, gia đình và bạn bè. Với lòng biết ơn chân thành và sâu sắc nhất, chúng tôi xin gửi lời cảm ơn đến ThS. Nguyễn Cao Trí. Thầy đã tạo điều kiện cho chúng tôi có cơ hội được tham gia và thực hiện đề tài với chủ đề "**Phát triển board điều khiển và hệ thống hỗ trợ cho nông nghiệp đô thị**". Trong quá trình thực hiện, thầy thường xuyên trao đổi, hỗ trợ tài liệu và giải đáp mọi thắc mắc liên quan đến nội dung cần tìm hiểu. Qua đó, chúng tôi có thể trau dồi kiến thức và kỹ năng giải quyết vấn đề cũng như định hướng được các công việc trong quá trình thực hiện thực tập tốt nghiệp cho đến luận văn tốt nghiệp. Ngoài ra, nhóm cũng xin gửi lời cảm ơn chân thành đến các thầy, cô của trường đại học Bách Khoa – Đại Học Quốc Gia Thành Phố Hồ Chí Minh nói chung cũng như các thầy, cô của khoa Khoa Học và Kỹ Thuật Máy Tính nói riêng, đã cung cấp những kiến thức nền tảng vô cùng quý giá để chúng tôi có thể áp dụng nó vào đề tài luận văn tốt nghiệp này. Báo cáo luận văn tốt nghiệp được hoàn thành trong thời gian khoảng ba tuần. Do đó, không tránh khỏi những thiếu sót là điều chắc chắn, nhóm rất mong nhận được những ý kiến đóng góp quý báu của quý thầy, cô và các bạn sinh viên để nhóm ngày càng hoàn thiện hơn. Sau cùng, nhóm xin kính chúc các thầy, cô trong Khoa Khoa Học và Kỹ Thuật Máy Tính nói chung, ThS. Nguyễn Cao Trí nói riêng thật dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh truyền đạt kiến thức cao đẹp của mình cho thế hệ sinh viên mai sau.

Nhóm thực hiện đề tài

TÓM TẮT

Sự phát triển của công nghệ trong thời đại hiện nay đã mang lại nhiều thay đổi tích cực trong tất cả các lĩnh vực. Và lĩnh vực nông nghiệp cũng có những bước tiến lớn trong thời đại của cuộc cách mạng công nghiệp 4.0. Hòa cùng với xu thế của thời đại, sinh viên nhóm chúng tôi đã tìm hiểu và thực hiện dự án về hệ thống công nghệ hỗ trợ dành cho nông nghiệp, đặc biệt đối với nông nghiệp trong đô thị. Sự phát triển mạnh mẽ của Internet of Things và sự cần thiết của nó dành cho nông nghiệp trong môi trường đô thị, nơi mà không có nhiều không gian dành cho nông nghiệp, đã được thực hiện ở giai đoạn Thực tập tốt nghiệp, để tài tiếp tục được phát triển ở giai đoạn Luận văn tốt nghiệp. Tại giai đoạn này, nhóm tiếp tục xây dựng hệ thống hoàn chỉnh hơn từ hình mẫu đã được thiết kế và xây dựng từ giai đoạn trước, đầy đủ hơn các chứng năng cơ bản để hỗ trợ giám sát và chăm sóc cho một hệ thống thủy canh trong môi trường đô thị. Hệ thống bao gồm phần cứng và phần mềm trên nền tảng ứng dụng web và ứng dụng di động được cải thiện về mặt giao diện, tương tác thân thiện hơn với người dùng và cung cấp các API hữu ích cho những nhà phát triển.

Mục lục

Danh sách hình vẽ	viii
Danh sách bảng	xiii
Danh sách chữ viết tắt	xv
1 GIỚI THIỆU ĐỀ TÀI	1
1.1 Ý tưởng	1
1.2 Mục tiêu	1
1.2.1 Phân nghiên cứu	1
1.2.2 Phân kết quả	2
2 CƠ SỞ LÝ THUYẾT	3
2.1 Giới thiệu về Internet of Things	3
2.1.1 Internet of Things (IoT)	3
2.1.2 Khả năng định danh độc nhất	4
2.1.3 Xu hướng và tính chất	4
2.1.4 Ứng dụng	5
2.2 Giới thiệu về Hệ thống thủy canh	6
2.2.1 Hệ thống dạng bắc (wick system)	7
2.2.2 Hệ thống thủy canh tĩnh (water culture)	8
2.2.3 Hệ thống ngập và rút định kỳ (ebb and flow system)	8
2.2.4 Hệ thống nhỏ giọt (Drip system)	9
2.2.5 Hệ thống màng dinh dưỡng NFT (Nutrient Film Technique)	9
2.2.6 Khí canh (Aeroponics)	11
2.3 Hệ thống thủy canh với IoT	11
2.4 Các hệ thống thủy canh tự động hiện có trên thị trường	12
2.5 Giới thiệu về các công nghệ được sử dụng	12
2.5.1 RESTful web service	12
2.5.2 Nodejs	14
2.5.3 Framework Expressjs	15

2.5.4	Sequelizejs	15
2.5.5	Angularjs	16
2.5.6	Angular 2	16
2.5.7	Framework Ionic 2	16
2.5.8	TypeScript	17
2.5.9	JWT	17
2.5.10	Postgresql	18
2.5.11	NodeBB	19
2.5.12	Git và GitHub	19
2.5.13	XXTEA	20
2.5.14	Arduino	20
2.5.15	ARM	21
2.5.16	Mbed	21
2.5.17	MQTT	22
3	THIẾT KẾ HỆ THỐNG	23
3.1	Kiến trúc mô hình hệ thống	23
3.2	Hệ thống board điều khiển	23
3.2.1	Tổng quan mô hình board điều khiển	23
3.2.2	Sơ đồ hoạt động	31
3.3	Kiến trúc phần mềm	45
3.3.1	Sequence diagram	45
3.3.2	Class diagram	46
3.4	Ứng dụng web	52
3.4.1	Sơ đồ use case	52
3.4.2	Thiết kế database	70
3.4.3	Thiết kế API	72
3.5	Ứng dụng di động	72
3.6	Giao tiếp giữa Board điều khiển và Web	73
3.6.1	Giao thức	73
3.6.2	Cấu trúc các gói tin	74

4 HIỆN THỰC HỆ THỐNG	77
4.1 Hệ thống Board điều khiển	77
4.1.1 Cài đặt phần mềm	77
4.1.2 Sử dụng cảm biến	78
4.1.3 Thời gian thực(RTC-DS1307)	79
4.1.4 Sử dụng Relay	81
4.1.5 Sử dụng module wifi ESP8266	83
4.1.6 Cấu trúc thư mục mã nguồn	88
4.1.7 Hoạt động của Esp8266	89
4.1.8 Thiết kế mạch điều khiển	90
4.1.9 Mạch chức năng đo ppm	91
4.2 Ứng dụng web	101
4.2.1 Hiện thực server API	101
4.2.2 Xây dựng database	118
4.2.3 Một số hình ảnh	120
4.3 Ứng dụng di động	128
4.3.1 Một số hình ảnh	128
4.4 Tương tác hệ thống	131
5 KIỂM THỬ HỆ THỐNG	135
5.1 Web server	135
5.2 Board điều khiển	135
6 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN CHO TƯƠNG LAI	136
6.1 Tổng kết	136
6.1.1 Kết quả đạt được	136
6.1.2 Thuận lợi và khó khăn trong quá trình thực hiện đề tài	136
6.1.3 Hướng phát triển	137
6.1.4 Kiến thức thu thập được	139
Tài liệu	139

Danh sách hình vẽ

1	Mô hình tương tác của mạng lưới thiết bị kết nối Internet.	3
2	Cả thế giới chìm trong Internet.	6
3	Mô hình hệ thống dạng bắc.	7
4	Mô hình hệ thống thủy canh tĩnh.	8
5	Mô hình hệ thống ngập và rút định kỳ.	9
6	Mô hình hệ thống nhỏ giọt.	10
7	Mô hình hệ thống màng dinh dưỡng NFT.	10
8	Mô hình hệ thống khí canh.	11
9	Cấu trúc một JSON.	14
10	Cơ chế hoạt động của Nodejs.	15
11	Quá trình đăng nhập bằng JWT.	18
12	Mô tả các thiết bị liên lạc với nhau thông qua MQTT broker.	22
13	Kiến trúc hệ thống.	23
14	Cảm biến nhiệt độ, độ ẩm.	24
15	Module đo ppm.	25
16	Module BH1750.	25
17	ESP8266.	26
18	RTC-DS1307	27
19	Module Realtime Clock DS1307.	27
20	Arduino UNO R3.	28
21	Relay.	29
22	Thanh ghi dịch.	29
23	Board ARM STM32F03C8T6.	30
24	Board NodeMCU ESP8266	31
25	Mô hình tổng quan.	32
26	Hoạt động độc lập, định thời đọc dữ liệu.	33
27	Máy bơm nước.	34
28	Máy sục oxy.	34
29	Đèn led.	34

30	Hoạt động của thanh ghi dịch.	35
31	Hoạt động của các actuators.	36
32	Hoạt động của tín hiệu điều khiển relay.	37
33	Hoạt động Thời gian thực.	37
34	Hoạt động thời gian biểu.	38
35	Hoạt động Lưu trữ thông tin actuator.	40
36	MQTT client.	41
37	Wifi station.	42
38	Wifi AP.	42
39	Wifi application.	43
40	Cấu trúc giao thức UART.	44
41	Cách hoạt động của giao thức UART.	44
42	Wifi sequence diagram.	45
43	NodeMCU Server sequence diagram.	45
44	Mqtt sequence diagram.	46
45	NodeMCU WifiStation class diagram.	46
46	NodeMCU MqttClient class diagram.	47
47	NodeMCU UserEEPROM class diagram.	47
48	NodeMCU Sensor class diagram.	48
49	NodeMCU SerialTransport class diagram.	48
50	NodeMCU Webserver class diagram.	49
51	NodeMCU SwitchAP class diagram.	49
52	NodeMCU Application class diagram.	50
53	STM SerialTransport class diagram.	50
54	STM Actuator class diagram.	51
55	STM Application class diagram.	51
56	Các chức năng của khách.	52
57	Đăng ký.	53
58	Đăng nhập.	54
59	Các chức năng của thành viên.	55

60	Use case thay đổi thông tin.	56
61	Use case thêm thiết bị.	58
62	Use case tương tác với diễn đàn.	59
63	Use case tìm kiếm mùa vụ.	61
64	Use case xem thông tin thiết bị.	62
65	Use case sửa thông tin thiết bị.	63
66	Use case tạo relay.	64
67	Use case thêm mùa vụ mới.	66
68	Use case sửa thông tin mùa vụ.	67
69	Use case cài đặt chương trình mùa vụ.	68
70	Use case cài đặt ngưỡng dữ liệu.	69
71	Thiết kế database	71
72	Mô hình cụm server	72
73	Mô hình RESTful API	72
74	Mô hình các giao thức trong hệ thống	73
75	Visual Studio 2017	77
76	Sublime Text 2	78
77	Sublime Text 2	78
78	Kết quả đọc DHT	79
79	Pin I2C trên board Arduino Uno.	79
80	Một số hàm trong thư viện Wire.h	80
81	Sơ đồ lắp đặt RTC với board Uno	80
82	DS1307 Timekeeper	81
83	Thông số trên relay.	82
84	Module CP2102-USB2UART.	83
85	Bảng nối chân Esp với CP2102.	84
86	Step 1.	85
87	Step 2.	85
88	Step 3.	86
89	Step 4.	87

90	Cấu trúc thư mục mã nguồn.	88
91	Giao diện web của esp8266.	89
92	Sơ đồ nguyên lý mạch điều khiển.	90
93	Sơ đồ mạch in.	91
94	Giới hạn với cây trồòng.	93
95	Các thành phần ion.	93
96	Trường hợp sử dụng dòng AC.	94
97	Trường hợp sử dụng dòng DC.	94
98	Schematic.	95
99	Oscillator cầu Wein.	95
100	4 di-ốt.	96
101	Nguồn đôi 12V - Trước.	97
102	Nguồn đôi 12V - Sau.	97
103	Jack DC.	98
104	IC TL074.	98
105	TL074 PIN.	99
106	Diode Zener.	99
107	Tụ 0.015uF.	100
108	Tụ 0.22uF.	100
109	Cầu Diode.	101
110	Cây thư mục	117
111	Thiết kế class	119
112	Trang chủ.	120
113	Trang đăng ký.	120
114	Trang đăng nhập.	121
115	Trang cá nhân.	121
116	Thêm một board mới.	122
117	Danh sách các board điều khiển.	122
118	Danh sách các actuator của một board.	123
119	Thêm một actuator.	123

120	Danh sách các mùa vụ.	124
121	Thêm một mùa vụ.	124
122	Thông tin mùa vụ.	125
123	Chi tiết mùa vụ.	125
124	Lịch trình hoạt động của các actuator.	126
125	Thêm lịch trình cho actuator.	126
126	Nguồn dữ liệu.	127
127	Trang forum.	127
128	Trang đăng nhập.	128
129	Trang chủ sau khi đăng nhập.	129
130	Thông tin mùa vụ và dữ liệu.	130
131	Danh sách các actuator.	131
132	Thao tác đăng nhập vào hệ thống sử dụng JWT.	132
133	Nhóm các thao tác CRUD.	133
134	Nhóm các thao tác tương tác với board điều khiển.	134
135	Kit RF thu phát wifi Node MCU.	138

Danh sách bảng

1	Mô tả use case đăng ký.	53
2	Mô tả use case đăng nhập.	54
3	Mô tả use case hay đổi thông tin cá nhân.	56
4	Use case thay đổi hình ảnh cá nhân.	57
5	Mô tả use case thay đổi mật khẩu.	57
6	Mô tả use case thêm thiết bị.	58
7	Mô tả use case tạo bài viết mới.	59
8	Mô tả use case xóa bài viết.	60
9	Mô tả use case thêm bình luận bài viết.	60
10	Mô tả use case xóa thiết bị.	60
11	Mô tả use case tìm kiếm mùa vụ.	61
12	Mô tả use case gửi bình luận mùa vụ.	62
13	Mô tả use case sửa thông tin thiết bị.	63
14	Mô tả use case tạo relay.	64
15	Mô tả use case xóa relay.	65
16	Mô tả use case sửa relay.	65
17	Mô tả use case tạo mùa vụ mới.	66
18	Mô tả use case sửa thông tin mùa vụ.	67
19	Mô tả use case xóa mùa vụ.	68
20	Mô tả use case cài đặt chương trình cho mùa vụ.	69
21	Mô tả use case cài đặt ngưỡng dữ liệu.	70
22	Cấu trúc chung của các gói tin.	74
23	Cấu hình.	74
24	Lịch trình cho actuator.	75
25	Điều khiển actuator.	75
26	Dữ liệu gửi từ các cảm biến.	75
27	Đồng bộ thời gian thực.	75
28	Thêm bớt actuator.	76
29	Gói tin phản hồi.	76

30	Thêm, bớt một board điều khiển	76
----	--	----

Danh sách chữ viết tắt

1. **IoT** Internet of Things
2. **RFID** Radio Frequency Identification
3. **NFC** Near Field Communication
4. **IP** Internet Protocol
5. **API** Application Programming Interface
6. **CSDL** Cơ Sở Dữ Liệu
7. **HTML** Hypertext Markup Language
8. **MVC** Model - Controller - View
9. **SPA** Single Page Application
10. **UI** User Interface
11. **IDE** Integrated Development Environment

1 GIỚI THIỆU ĐỀ TÀI

1.1 Ý tưởng

Công nghệ thủy canh đã được nghiên cứu từ thế kỷ 17. Đến nay, công nghệ này đã hoàn thiện, hướng đến những nông sản sạch, xanh, không ô nhiễm. Với quy mô gia đình nhỏ lẻ, những chậu hoa hay rau xanh có thể phát triển mạnh mẽ không cần đất nơi góc sân thượng, lan can, bậu cửa. Với quy mô thương mại, những nhà kính trồng hoa, rau, củ, quả... phát triển sạch, năng suất cao, chủ động, đáp ứng đầy đủ các tiêu chuẩn an toàn vì không bị ảnh hưởng của những nguồn ô nhiễm từ đất. Không những vậy, vì trồng không cần đất nên có thể chia không gian thành nhiều tầng để mở rộng diện tích sản xuất.

Trên thế giới hiện đã có những ứng dụng sử dụng IoT để tự động hóa hoàn toàn quy trình trồng rau sạch thủy canh nhà phố, trong khi ở Việt Nam còn khá mới mẻ. Với những yếu tố trên, nhóm đã hiện thực và phát triển Hệ thống thủy canh theo hướng IoT với mục đích tạo ra nông sản sạch, không ô nhiễm cho mọi người.

1.2 Mục tiêu

1.2.1 Phân nghiên cứu

- Tìm hiểu và thực hành quá trình thiết kế và xây dựng một dự án.
- Tìm hiểu nắm vững các nền tảng, thư viện cũng như các công cụ để thực hiện dự án này.
- Áp dụng các kiến thức về thiết kế mềm, kĩ thuật lập trình vào dự án.
- Tổ chức dự án cho phép dễ dàng bảo trì, mở rộng.
- Tìm hiểu về Hệ thống thủy canh Hydroponics:
 - Tổng quan, khái niệm về thủy canh.
 - Các mô hình thủy canh và cách vận hành của từng mô hình.
- Học hỏi và biết cách sử dụng các công nghệ mới hiện nay.

1.2.2 Phần kết quả

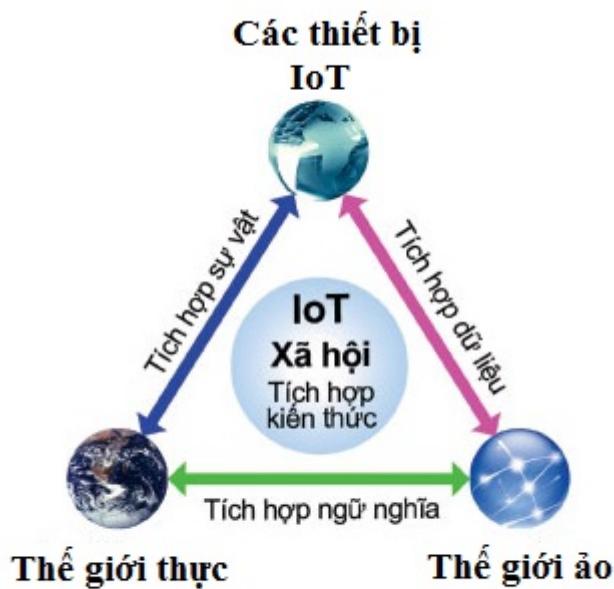
- Hiện thực mạch đo giá trị ppm (parts per million) trong môi trường dung dịch.
- Hệ thống board điều khiển với các cảm biến (nhiệt độ, độ ẩm, ánh sáng, ppm) và các relay điều khiển (máy bơm nước, quạt, đèn) có thể đáp ứng các tiêu chí:
 - Vận hành trong thời gian dài, ổn định.
 - Chức năng thu thập dữ liệu các cảm biến phải hoạt động tốt, ổn định, sai số chấp nhận được.
 - Chức năng điều khiển các thiết bị như máy bơm nước, xục oxi, đèn chiếu sáng.. đáp ứng thời gian thực.
 - Các kết nối về sóng wifi ổn định, cung cấp chức năng cấu hình cho module wifi.
 - Protocol giao tiếp giữa board điều khiển và web-server hoạt động chính xác.
 - Thiết kế hiện thực board điều khiển về mặt phần cứng, phần khung, vỏ bảo vệ.
- Xây dựng website cho người dùng với các chức năng:
 - Thu nhận các thông tin từ thiết bị gửi lên từ các cảm biến.
 - Phân tích các thông tin gửi lên, giám sát, đưa ra cảnh báo và giải pháp xử lý đối với trường hợp các thông tin vượt ngưỡng cho phép.
 - Tạo một lịch trình và nạp xuống cho thiết bị.
 - Theo dõi thường xuyên, upload hình ảnh về các mùa vụ đang trồng.
 - Chia sẻ thông tin của mình cho các người dùng khác, xây dựng một cộng đồng chung để mọi người trao đổi và chia sẻ thông tin trong lĩnh vực thủy canh.
- Hiện thực mô hình thủy canh thu nhỏ cho demo.

2 CƠ SỞ LÝ THUYẾT

2.1 Giới thiệu về Internet of Things

2.1.1 Internet of Things (IoT)

Internet of Things - Mạng lưới vật kết nối Internet là một kịch bản của thế giới, khi mà mỗi đồ vật, con người được cung cấp một định danh của riêng mình, và tất cả có khả năng truyền tải, trao đổi thông tin, dữ liệu qua một mạng duy nhất mà không cần đến sự tương tác trực tiếp giữa người với người, hay người với máy tính. IoT đã phát triển từ sự hội tụ của công nghệ không dây, công nghệ vi cơ điện tử và Internet. Nói đơn giản là một tập hợp các thiết bị có khả năng kết nối với nhau, với Internet và với thế giới bên ngoài để thực hiện một công việc nào đó.



Hình 1: Mô hình tương tác của mạng lưới thiết bị kết nối Internet.

2.1.2 Khả năng định danh độc nhất

Điểm quan trọng của IoT đó là các đối tượng phải có thể được nhận biết và định dạng (identifiable). Nếu mọi đối tượng, kể cả con người, được "đánh dấu" để phân biệt bản thân đối tượng đó với những thứ xung quanh thì chúng ta có thể hoàn toàn quản lý được nó thông qua máy tính. Việc đánh dấu (tagging) có thể được thực hiện thông qua nhiều công nghệ, chẳng hạn như RFID, NFC, mã vạch, mã QR, watermark kĩ thuật số... Việc kết nối thì có thể thực hiện qua Wi-Fi, mạng viễn thông băng rộng (3G, 4G), Bluetooth, ZigBee, hồng ngoại...

Ngoài những kĩ thuật nói trên, nếu nhìn từ thế giới web, chúng ta có thể sử dụng các địa chỉ độc nhất để xác định từng vật, chẳng hạn như địa chỉ IP. Mỗi thiết bị sẽ có một IP riêng biệt không nhầm lẫn. Sự xuất hiện của IPv6 với không gian địa chỉ cực kì rộng lớn sẽ giúp mọi thứ có thể dễ dàng kết nối vào Internet cũng như kết nối với nhau.

2.1.3 Xu hướng và tính chất

Thông minh

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

Kiến trúc dựa trên sự kiện

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

Là một hệ thống phức tạp

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

Kích thước

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

Vấn đề không gian, thời gian

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

Luồng năng lượng mới

Hiện nay, IoT đang trải qua giai đoạn phát triển "bộc phát" và điều này xảy ra nhờ vào một số nhân tố, trong đó gồm IPv6, 4G, chi phí, tính sẵn có của công nghệ.

2.1.4 Ứng dụng

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

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

- Quản lý môi trường.
- Phản hồi trong các tình huống khẩn cấp.
- Mua sắm thông minh.
- Quản lý các thiết bị cá nhân.
- Đồng hồ đo thông minh.
- Tự động hóa ngôi nhà.



Hình 2: Cả thế giới chìm trong Internet.

2.2 Giới thiệu về Hệ thống thủy canh

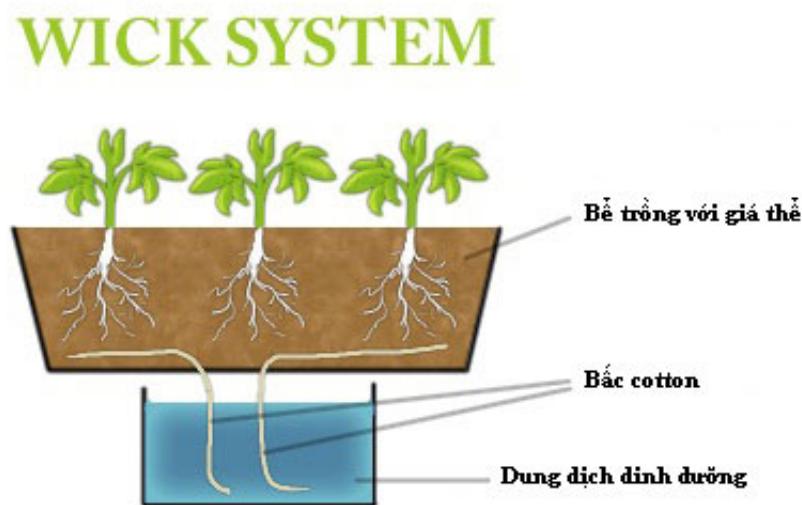
Trồng cây trong dung dịch (thủy canh) là kỹ thuật trồng cây không dùng đất mà trồng trực tiếp vào môi trường dinh dưỡng hoặc giá thể mà không phải là đất. Các giá thể có thể là cát, trấu, vỏ xơ dừa, than bùn, vermiculite perlite... Thường được định nghĩa như là “trồng cây trong nước” hoặc “trồng cây không cần đất”, kỹ thuật thủy canh là một

trong những nghề làm vườn hiện đại. Bí quyết của kỹ thuật này là cung cấp đủ và đúng lúc cho cây trồng các nguyên tố khoáng cần thiết. Cung cấp đầy đủ cái ăn, bảo đảm đủ ánh sáng, CO₂ cho quá trình quang hợp, O₂ cho quá trình hô hấp, cây trồng có thể phát triển khỏe mạnh theo ý muốn của người trồng.

Hiện nay hầu hết các máng trồng đều được làm bằng nhựa, nhưng có thể làm bằng các vật liệu khác như bê tông, thủy tinh, kim loại và gỗ. Các máng trồng nên được che nắng để không cho tảo, rong rêu phát triển trong dung dịch thủy canh. Các mô hình dưới đây có thể nghiên cứu, biến tấu thành các kiểu khác nhau phù hợp với điều kiện từng nơi.

2.2.1 Hệ thống dạng bắc (wick system)

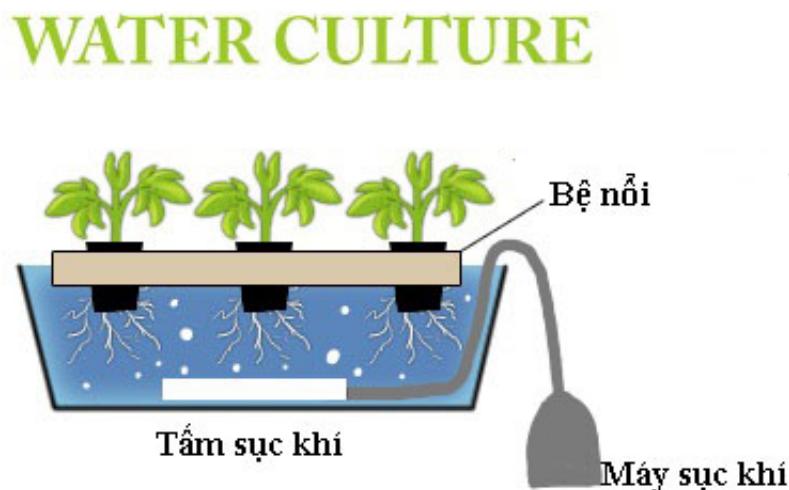
Hệ thống dạng bắc cho đến nay là dạng hệ thống thủy canh đơn giản nhất. Đúng như tên gọi, bí quyết của hệ thống này nằm ở chõ sợi bắc. Đặt một đầu của sợi bắc hút sao cho chạm vào phần rễ cây. Đầu kia của bắc chìm trong dung dịch dinh dưỡng. Sợi bắc này sẽ làm nhiệm vụ hút nước và dung dịch dinh dưỡng lên cung cấp cho rễ cây (tương tự như sợi bắc trong đèn dầu, hút dầu lên để duy trì sự cháy). Như vậy cây sẽ có đủ nước và chất dinh dưỡng để phát triển.



Hình 3: Mô hình hệ thống dạng bắc.

2.2.2 Hệ thống thủy canh tĩnh (water culture)

Hệ thống thường thùng hay nước chứa dung dịch thủy canh, phần bệ giữ các cây thường làm bằng chất dẻo nhẹ như xốp và đặt nổi ngay trên dung dịch dinh dưỡng, rễ cây ngập chìm trong nước có chứa dung dịch dinh dưỡng. Vì môi trường thiếu khí oxy nên cần có 1 máy bơm bơm khí vào khối sủi bọt để cung cấp oxy cho rễ. Hệ thống thủy canh dạng này thường dùng phổ biến trong dạy học. Hệ thống ít tốn kém, có thể tận dụng bể chứa nước hay những bình chứa không rỉ khác.

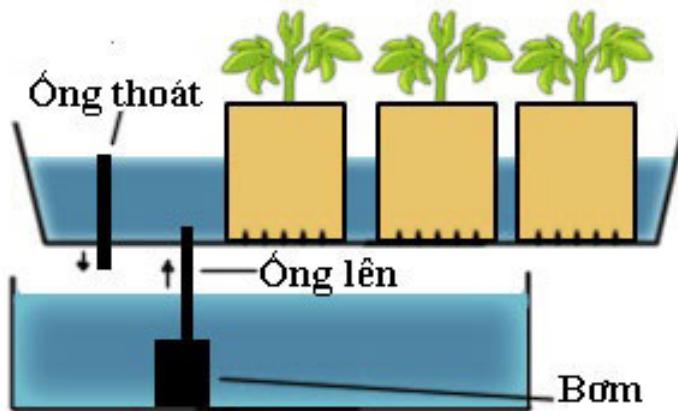


Hình 4: Mô hình hệ thống thủy canh tĩnh.

2.2.3 Hệ thống ngập và rút định kỳ (ebb and flow system)

Không giống như hệ thống thủy canh tĩnh ở trên, phần rễ cây luôn chìm trong nước chỉ thích hợp cho một số ít cây trồng. Hệ thống ngập và rút định kỳ có một máy bơm điều khiển để có thể bơm dung dịch dinh dưỡng vào khay trồng và rút ra theo chu kỳ đã được định sẵn. Như vậy rễ cây sẽ có những lúc không ngập trong nước để “thở” một cách tự nhiên, tránh bị ngập, úng. Hệ thống này thường được áp dụng cho mô hình aquaponics.

EBB AND FLOW



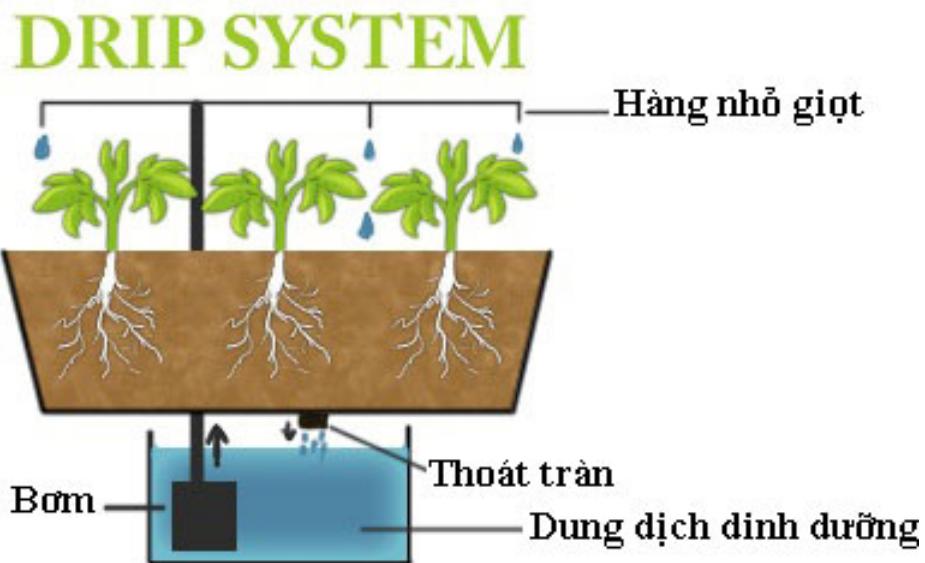
Hình 5: Mô hình hệ thống ngập và rút định kỳ.

2.2.4 Hệ thống nhỏ giọt (Drip system)

Hệ thống nhỏ giọt là loại hệ thống thủy canh được sử dụng rộng rãi nhất trên thế giới. Máy bơm sẽ bơm dung dịch dinh dưỡng lên, nhỏ trực tiếp vào gốc của cây trồng bởi những đường ống nhỏ giọt theo định kỳ. Dung dịch dinh dưỡng dư chảy xuống sẽ được thu hồi trong bể tái sử dụng. Như vậy, hệ thống này sử dụng dung dịch dinh dưỡng khá hiệu quả, nước dư ra được tái sử dụng, không bị hao phí. Hệ thống này có thể dùng để trồng cây thảo mộc và các loại hoa, các loại cây ăn trái như cà chua, dưa leo, dưa lưới, ớt,...

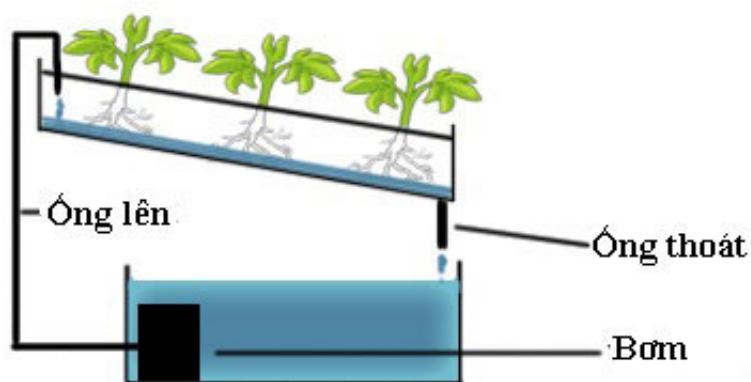
2.2.5 Hệ thống màng dinh dưỡng NFT (Nutrient Film Technique)

Trong hệ thống màng dinh dưỡng, dung dịch dinh dưỡng được bơm liên tục vào khay trồng và chảy qua rễ của cây, sau đó chúng chảy về bồn chứa để tái sử dụng. Thường thì trong hệ thống màng dinh dưỡng không cần dùng thêm chất trồng, giúp tiết kiệm chi phí thay chất trồng sau mỗi vụ mùa. Hệ thống này thường sử dụng trong quy mô lớn với mục đích thương mại.



Hình 6: Mô hình hệ thống nhỏ giọt.

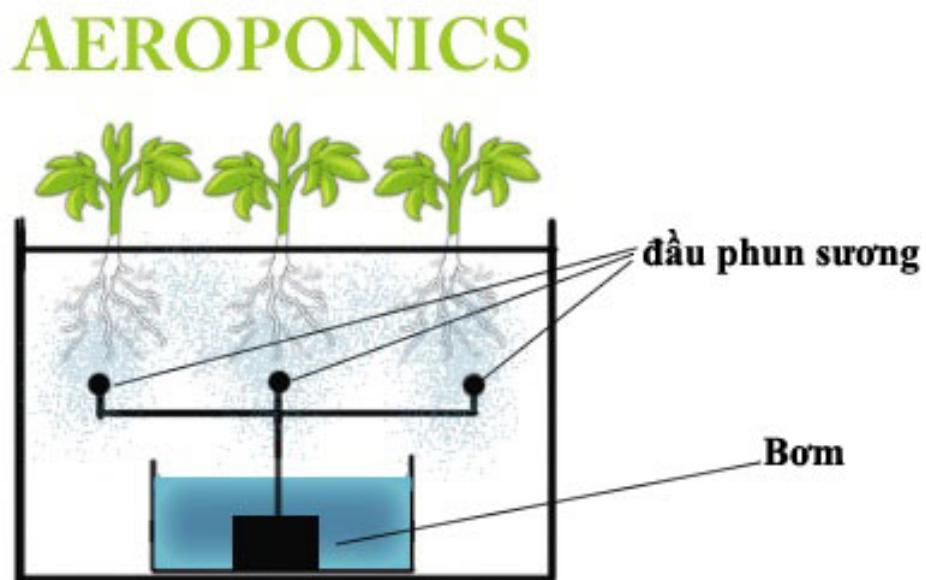
NFT SYSTEM



Hình 7: Mô hình hệ thống màng dinh dưỡng NFT.

2.2.6 Khí canh (Aeroponics)

Khí canh là hệ thống thủy canh dạng kỹ thuật cao nhất. Giống như hệ thống màng dinh dưỡng, chất tròng chủ yếu là không khí. Rễ phơi trong không khí và được phun sương bằng dung dịch dinh dưỡng. Việc phun sương thường được thực hiện mỗi vài phút. Như vậy, cây vừa có đủ thức ăn, vừa có đủ nước uống và luôn có không khí để thở. Hiện nay khí canh được ứng dụng trong mô hình trồng khoai tây.



Hình 8: Mô hình hệ thống khí canh.

2.3 Hệ thống thủy canh với IoT

Việc kết hợp thủy canh và IoT ở nước ngoài đã trở nên khá phổ biến trong khi ở Việt Nam vẫn còn mới mẻ. Việc kết hợp IoT vào thủy canh giúp người trồng rau kiểm soát một cách dễ dàng tình trạng vườn trồng của mình thông qua số liệu về nhiệt độ, độ ẩm, pH ... mà các cảm biến gửi về qua mạng Internet. Đồng thời người dùng có thể chăm sóc vườn rau của mình mà không cần thiết phải tiếp xúc với nó. Tất cả mọi hoạt động tưới tiêu, bơm oxy, cung cấp ánh sáng, phân bón,... đều được tự động hóa theo quy trình cố sẵn. Người dùng chỉ việc thiết lập thông số phù hợp cho từng cây trồng trong mỗi mùa vụ và chờ ngày thu hoạch. Kiểm soát thông qua mạng Internet cho phép người dùng quản

lý vườn trồng của mình ở bất kỳ đâu, trong mọi thời điểm. Thủy canh kết hợp với IoT là xu hướng mới và là tương lai của nông nghiệp đô thị nhờ tính tiện ích của nó.

2.4 Các hệ thống thủy canh tự động hiện có trên thị trường

Theo thông tin nhóm nghiên cứu thì hiện nay trên thị trường chưa có nhiều hệ thống thủy canh tự động mang tính thương mại. Một số công ty start up nổi tiếng trong lĩnh vực này như Hachi, Lisado nhưng những hệ thống này có các đặc điểm chung:

- Mới chỉ phát triển ở khu vực phía Bắc.
- Chỉ hỗ trợ nền tảng di động, chưa hỗ trợ môi trường web trên PC hay laptop.
- Chưa có một hệ thống forum để cộng đồng có thể chia sẻ kinh nghiệm với nhau.
- Chưa hỗ trợ tự động đo nồng độ dinh dưỡng ppm.

Nhóm đang hướng tới sự khác biệt bằng cách tạo ra một hệ thống hỗ trợ các tính năng nói trên.

2.5 Giới thiệu về các công nghệ được sử dụng

2.5.1 RESTful web service

REST là viết tắt của cụm từ Representational State Transfer (đôi khi còn được viết là ReST) là một kiểu kiến trúc được sử dụng trong việc giao tiếp giữa các máy tính (máy tính cá nhân và máy chủ của trang web) trong việc quản lý các tài nguyên trên internet. Trong REST thì tất cả mọi thứ đều là tài nguyên và nguồn tài nguyên này sẽ được truy xuất thông qua giao thức HTTP.

Một kiến trúc REST thường có một server cung cấp các API truy xuất tài nguyên và client truy xuất, chỉnh sửa các tài nguyên đó.

REST cho phép các tài nguyên biểu diễn dưới dạng text, XML, JSON, ... REST client có thể yêu cầu một kiểu dữ liệu cụ thể thông qua giao thức HTTP.

Các phương thức HTTP được sử dụng:

- POST: sử dụng khi tạo một tài nguyên mới.

- GET: dùng để truy xuất tài nguyên.
- PUT: cập nhật tài nguyên.
- DELETE: dùng để xóa bỏ tài nguyên.

JSON (Javascript Object Notation)

Trong ứng dụng mà nhóm hiện thực sử dụng JSON để trao đổi dữ liệu.

JSON là 1 định dạng hoán vị dữ liệu nhanh. Chúng là cơ sở dựa trên tập hợp của Ngôn Ngữ Lập Trình JavaScript, tiêu chuẩn ECMA-262 phiên bản 3 - tháng 12 năm 1999.

JSON được biểu diễn dưới dạng cặp tên và giá trị name – value. Trong những ngôn ngữ khác nhau, đây được nhận thấy như là một đối tượng (object), bản ghi (record), cấu trúc (struct), từ điển (dictionary), bảng băm (hash table), danh sách khóa (keyed list), hay mảng liên hợp.

JSON có 5 kiểu dữ liệu chính:

- Number: kiểu số bao gồm số nguyên và số thực.
- String: kiểu chuỗi, nội dung bao bối cặp dấu nháy kép “ ”, những ký tự đặt biệt được escape bởi dấu . Theo chuẩn JSON thì không sử dụng dấu nháy đơn như Javascript để bọc chuỗi.
- Boolean: kiểu luận lý bao gồm 2 giá trị là true và false.
- Array: kiểu mảng, gồm các phần tử phân cách nhau bởi dấu phẩy ‘,’ và mảng được bao bối cặp dấu [và].
- Object: kiểu đối tượng, gồm những cặp giá trị đi cùng nhau, mỗi cặp phân cách bởi dấu phẩy ‘,’, đối tượng được bao bối cặp dấu và , cặp giá trị bao gồm tên và giá trị được phân cách bởi dấu hai chấm ‘:’.
- Null: giá trị null.

JSON được sử dụng rộng rãi vì những ưu điểm của nó:

- Có thể đọc hiểu được (human-readability).



Hình 9: Cấu trúc một JSON.

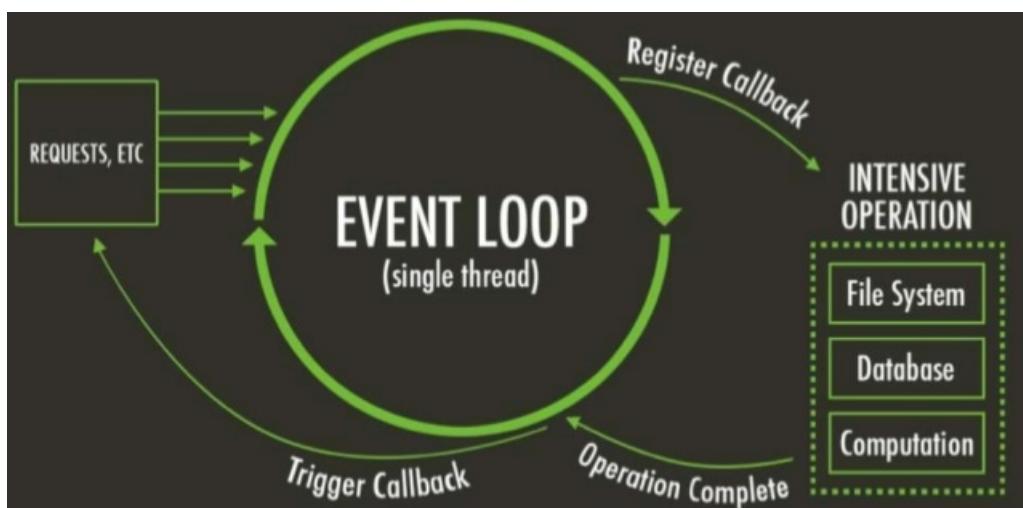
- Là kiểu dữ liệu trên nền cơ sở Javascript nên dễ dàng tiếp cận.
- Dữ liệu truyền tải ngắn gọn so với những định dạng dữ liệu khác như: xml, html...
- Đễ dàng chuyển đổi(parse) dữ liệu từ dạng chuỗi (nhận từ server) sang dữ liệu có thể sử dụng được (thành Object, Number, Array)...
- Đễ truy cập nội dung.

2.5.2 Nodejs

Node.js là một phần mềm mã nguồn mở được viết dựa trên ngôn ngữ JavaScript cho phép lập trình viên có thể xây dựng các ứng dụng chạy trên máy chủ. Ban đầu, Node.js được phát triển bởi Ryan Dahl. Phiên bản đầu tiên của Node.js được cho ra mắt vào năm 2009. Node.js có thể chạy được trên nhiều nền tảng khác nhau như Windows, Linux hay Mac OS. Node.js được phát triển sử dụng V8 Engine là bộ thư viện JavaScript được Google phát triển để viết trình duyệt web Chrome.

Bản thân Node.js không phải là một ngôn ngữ lập trình mới, thay vào đó Node.js là một nền tảng mã nguồn mở được viết dựa trên ngôn ngữ JavaScript.

Nodejs đang rất được ưa chuộng bởi tốc độ chạy rất nhanh của nó. Sở dĩ Nodejs chạy rất nhanh vì nó thực hiện cơ chế non-blocking I/O và asynchronous (bất đồng bộ). Khi một request đến, nó không dừng lại chờ cho request đó xử lý xong mà vòng lặp event loop



Hình 10: Cơ chế hoạt động của Nodejs.

sẽ tiếp nhận các request liên tục và xử lý chúng liên tục. Khi một request xử lý xong, hàm callback sẽ được gọi để trả về kết quả cho client hoặc thực hiện một tác vụ nào đó. Như vậy các request sẽ được xử lý song song với nhau, tiết kiệm được rất nhiều thời gian. Nhưng đây cũng là một điểm khó đòi hỏi người lập trình phải quản lý chặt chẽ thứ tự thực hiện các hành động vì không phải lúc nào các tác vụ cũng chạy song song nhau, có những tác vụ phải thực hiện theo tuần tự.

2.5.3 Framework Expressjs

Expressjs là một framework có thể coi là nhỏ gọn và đơn giản nhất của Nodejs. Với Expressjs, lập trình viên có thể tạo ra một RESTful web service một cách rất nhanh chóng, đơn giản và dễ dàng mở rộng.

2.5.4 Sequelizejs

Sequelize là một ORM dành cho Node.js. Nó hỗ trợ truy cập một cách dễ dàng đến PostgreSQL, MySQL, MariaDB, SQLite và MSSQL cùng với các tính năng như là relations, transaction, replication ...

ORM (Object Relational Mapping) là một kỹ thuật lập trình để ánh xạ dữ liệu giữa các CSDL quan hệ và đối tượng trong các ngôn ngữ lập trình hướng đối tượng như Java, C#, ... Trong đó, các đối tượng ánh xạ với các bảng, các quan hệ của đối tượng ánh xạ với các ràng buộc liên quan giữa các bảng. Sequelizejs sẽ làm công việc này, nó giúp chúng

ta gọi các hàm tương ứng thay vì viết những câu truy vấn bằng ngôn ngữ SQL. Đây là một module rất tiện lợi cho người lập trình web.

2.5.5 Angularjs

Angular là một bộ Javascript Framework rất mạnh và thường được sử dụng để xây dựng project Single Page Application (SPA). Nó hoạt động dựa trên các thuộc tính mở rộng HTML (các attributes theo quy tắc của Angular).

Ưu điểm của Angularjs:

- Phát triển dựa trên Javascript
- Tạo các ứng dụng client-side theo mô hình MVC.
- Khả năng tương thích cao, tự động xử lý mã javascript để phù hợp với mỗi trình duyệt.
- Mã nguồn mở, miễn phí hoàn toàn và được sử dụng rộng rãi.

2.5.6 Angular 2

Angular 2 là 1 framework UI để xây dựng ứng dụng web trên desktop và mobile. Nó được xây dựng dựa trên Javascript. Chúng ta có thể dùng nó để xây dựng 1 ứng dụng client side thú vị dùng HTML, CSS và Javascript. Angular 2 có rất nhiều cải tiến so với Angular 1 để dễ dàng học và phát triển các ứng dụng quy mô doanh nghiệp. Với angular 2 thì chúng ta dễ dàng xây dựng được 1 ứng dụng có thể dễ dàng mở rộng, bảo trì, kiểm nghiệm và chuẩn hóa ứng dụng của mình. Angular 2 hoàn toàn được viết bằng Typescript. Điều đó đồng nghĩa là nó hỗ trợ cho ES6 Modules, class, ...

2.5.7 Framework Ionic 2

Ionic là một framework dùng để xây dựng các ứng dụng lai bằng cách sử dụng HTML, CSS, và JavaScript. Nó đi kèm với một tập hợp các thành phần giao diện và các hàm mà bạn có thể sử dụng để tạo ra các ứng dụng di động đầy đủ chức năng và hấp dẫn. Ionic được xây dựng trên Cordova stack. Bạn không thể tạo các ứng dụng di động với một mình Ionic bởi vì nó chỉ xử lý một phần giao diện người dùng. Nó cần phải làm việc

với Angular, cái mà xử lý các logic của ứng dụng, và Cordova, framework ứng dụng đa nền tảng cho phép bạn biên dịch ứng dụng của bạn thành một tập tin có thể cài đặt và chạy nó bên trong web view của thiết bị di động. Ứng dụng được xây dựng với Cordova và Ionic có thể chạy trên cả thiết bị Android và iOS. Điểm cải tiến so với phiên bản 1 là Ionic 2 sử dụng Angular 2 để phát triển các ứng dụng của mình.

2.5.8 Typescript

TypeScript là một ngôn ngữ mã nguồn mở miễn phí hiện đang được phát triển và bảo trì bởi Microsoft. Nó là tập cha của JavaScript, với các bổ sung các tùy chọn kiểu tĩnh và lớp trên cơ sở lập trình hướng đối tượng cho ngôn ngữ này. Anders Hejlsberg, kiến trúc sư ngôn ngữ C# và là người tạo ra ngôn ngữ Delphi và Turbo Pascal đã tham gia phát triển TypeScript. TypeScript có thể sử dụng để phát triển ứng dụng chạy phía client, hay phía server (Node.js) TypeScript được thiết kế để phát triển ứng dụng lớn và được biến đổi - biên dịch sang JavaScript. Vì TypeScript là tập cha của JavaScript nên bất kỳ chương trình JavaScript nào đã có cũng đều là chương trình TypeScript hợp lệ.

2.5.9 JWT

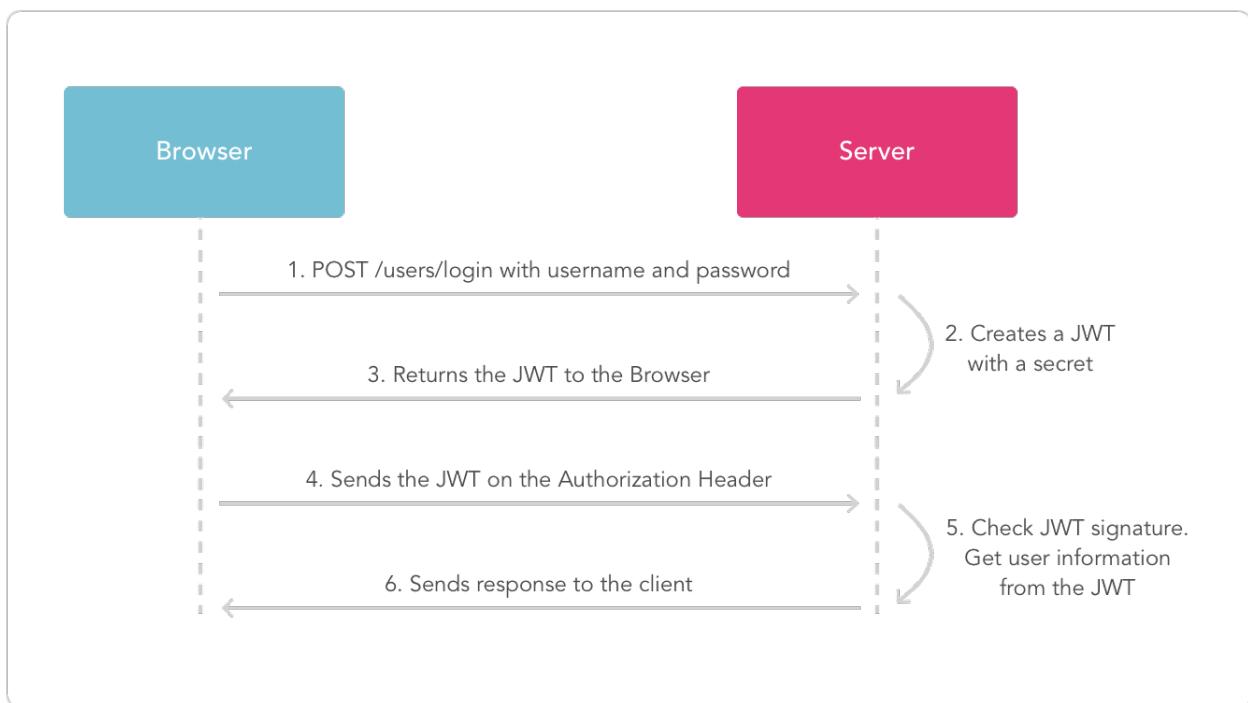
JSON Web Token (JWT) là 1 tiêu chuẩn mở (RFC 7519) định nghĩa cách thức truyền tin an toàn giữa các thành viên bằng 1 đối tượng JSON. Thông tin này có thể được xác thực và đánh dấu tin cậy nhờ vào "chữ ký" của nó. Phần chữ ký của JWT sẽ được mã hóa lại bằng HMAC hoặc RSA.

JWT gồm có 3 phần:

- Header

Header bao gồm hai phần chính: loại token (mặc định là JWT - Thông tin này cho biết đây là một Token JWT) và thuật toán đã dùng để mã hóa (HMAC SHA256 - HS256 hoặc RSA).

```
{ "alg": "HS256", "typ": "JWT" }
```



Hình 11: Quá trình đăng nhập bằng JWT.

- Payload Payload chứa các claims. Claims là một các biểu thức về một thực thể (chẳng hạn user) và một số metadata phụ trợ. Có 3 loại claims thường gặp trong Payload: reserved, public và private claims.
- Signature Chữ ký Signature trong JWT là một chuỗi được mã hóa bởi header, payload cùng với một chuỗi bí mật theo nguyên tắc sau:

HMACSHA256(base64UrlEncode(header) + "." + base64UrlEncode(payload),
secret)

Do bản thân Signature đã bao gồm cả header và payload nên Signature có thể dùng để kiểm tra tính toàn vẹn của dữ liệu khi truyền tải.

2.5.10 Postgresql

PostgreSQL là một hệ quản trị cơ sở dữ liệu quan hệ và đối tượng dựa trên POSTGRES, bản 4.2, được khoa điện toán của đại học California tại Berkeley phát triển. POSTGRES mở đường cho nhiều khái niệm quan trọng mà các hệ quản trị dữ liệu thương mại rất lâu sau mới có.

PostgreSQL được phổ biến bằng giấy phép BSD cổ điển. Nó không quy định những hạn chế trong việc sử dụng mã nguồn của phần mềm. Bởi vậy PostgreSQL có thể được dùng, sửa đổi và phổ biến bởi bất kỳ ai cho bất kỳ mục đích nào.

PostgreSQL cũng là hệ quản trị cơ sở dữ liệu hỗ trợ mạnh trong việc lưu trữ dữ liệu không gian. PostgreSQL kết hợp với module Postgis cho phép người dùng lưu trữ các lớp dữ liệu không gian. Khi sử dụng PostgreSQL, Postgis kết hợp với các phần mềm GIS hỗ trợ hiển thị, truy vấn, thống kê hoặc xử lý dữ liệu không gian.

2.5.11 NodeBB

NodeBB là một nền tảng diễn đàn thế hệ mới sử dụng công nghệ web sockets để có thể tương tác ngay lập tức và thông báo thời gian thực. NodeBB có nhiều tính năng hiện đại với tư duy sáng tạo như tương tác mạng xã hội và dòng chảy thảo luận.

NodeBB là một phần mềm mã nguồn mở được sử dụng khá rộng rãi.

2.5.12 Git và GitHub

Git

Git là một hệ thống quản lý phiên bản cho những tập tin trên máy tính và các công việc chung cùng thực hiện trên những tập tin này giữa nhiều người. Git là một trong những hệ thống quản lý phiên bản phân tán phổ biến nhất hiện nay. Nó giúp mỗi máy tính có thể lưu trữ nhiều phiên bản khác nhau của một mã nguồn được nhân bản (clone) từ một kho chứa mã nguồn (repository), mỗi thay đổi vào mã nguồn trên máy tính sẽ có thể ủy thác (commit) rồi đưa lên máy chủ nơi đặt kho chứa chính. Một máy tính khác (nếu họ có quyền truy cập) cũng có thể clone lại mã nguồn từ kho chứa.

Do đó Git có nhiều ưu điểm:

- Giúp làm việc và quản lý mã nguồn của hệ thống đang xây dựng theo nhóm một cách đơn giản và hiệu quả.
- Dễ sử dụng và an toàn.
- Quản lý mã nguồn và tài liệu, có một host đóng vai trò là server làm kho chứa mã nguồn, các máy khác có thể clone mã nguồn từ server này ở bất kỳ đâu.

GitHub

Trong quá trình xây dựng hệ thống này, nhóm sử dụng GitHub làm công cụ để quản lý mã nguồn của hệ thống được xây dựng và các tài liệu thực hiện cho luận văn. GitHub là một dịch vụ máy chủ repository công cộng, người dùng có thể tạo tài khoản trên GitHub để tạo ra các kho chứa của riêng mình.

Dường dẫn mã nguồn của hệ thống này trên GitHub:

- Mã nguồn web server: <https://github.com/bathach95/hydroponic>
- Mã nguồn mobile application: <https://github.com/bathach95/HydroponicMobileApp>

2.5.13 XXTEA

XXTEA là một thuật toán mã hóa nâng cấp của XTEA.

XTEA viết tắt của Extended Tiny Encryption Algorithm, được thiết kế bởi David Wheeler and Roger Needham của phòng thí nghiệm máy tính Cambridge. Giải thuật này không hề được đăng ký bản quyền do đó bất kỳ ai cũng có thể sử dụng nó một cách tự do.

XTEA là dạng mã hóa mật mã khối, kích thước khối 64 bit, khóa bí mật 128 bit.

Đúng theo tên gọi Tiny, XTEA là 1 giải thuật cực kỳ đơn giản và rất dễ code. Nó vốn được thiết kế với mục đích áp dụng trong các hệ thống với cấu hình phần cứng hạn chế

2.5.14 Arduino

Arduino là một board mạch vi xử lý, nhằm xây dựng các ứng dụng tương tác với nhau hoặc với môi trường được thuận lợi hơn. Phần cứng bao gồm một board mạch nguồn mở được thiết kế trên nền tảng vi xử lý AVR Atmel 8bit, hoặc ARM Atmel 32-bit. Những Model hiện tại được trang bị gồm 1 cổng giao tiếp USB, 6 chân đầu vào analog, 14 chân I/O kỹ thuật số tương thích với nhiều board mở rộng khác nhau. Nền tảng Arduino là mã nguồn mở bao gồm cả phần mềm và phần cứng hướng tới việc 'easy to use' đối với những người mới bắt đầu. Arduino cung cấp:

- Ngôn ngữ lập trình Arduino dựa trên C/C++.
- Môi trường phát triển ứng dụng (IDE) Arduino IDE, ngoài ra rất nhiều phần mềm hỗ trợ lập trình arduino.

- Thư viện mã nguồn mở.
- Hệ thống các board tham chiếu như Arduino Uno, Due, Mega.

Tạo sao lại lựa chọn arduino:

- Chi phí: các board arduino có giá tương đối so với các board khác, ngoài ra arduino còn là mã nguồn mở sẽ tiết kiệm được tiền bản quyền.
- Đa nền tảng: Arduino IDE được hỗ trợ trên nhiều nền tảng như windows, linux, macOS.
- Môi trường lập trình, nghiên cứu đơn giản và dễ tiếp cận, tiết kiệm thời gian.
- Hệ thống phần mềm mã nguồn mở được xây dựng bởi cộng đồng người dùng lớn.
- Phần cứng có thể được phát triển theo mục đích người sử dụng dựa trên các board sẵn có.

2.5.15 ARM

Cấu trúc ARM (viết tắt từ tên gốc là Advanced RISC Machine) là một loại cấu trúc vi xử lý kiểu RSIC được sử dụng rộng rãi trong các thiết kế nhúng. Do có đặc điểm tiết kiệm năng lượng, các bộ cpu ARM chiếm ưu thế trong các sản phẩm điện tử di động, mà với các sản phẩm này việc tiêu tán công suất thấp là một mục tiêu thiết kế quan trọng hàng đầu.

Đi kèm với hệ thống vi xử lý là nền tảng lập trình đa dạng, các thư viện mã nguồn mở, hệ thống các board tham chiếu.

2.5.16 Mbed

Mbed là nền tảng và hệ điều hành được phát triển cho các thiết bị kết nối internet dựa trên các thiết bị 32-bit Arm Cortex-M.

Mbed cung cấp nhiều phương pháp tiếp cận lập trình khác nhau:

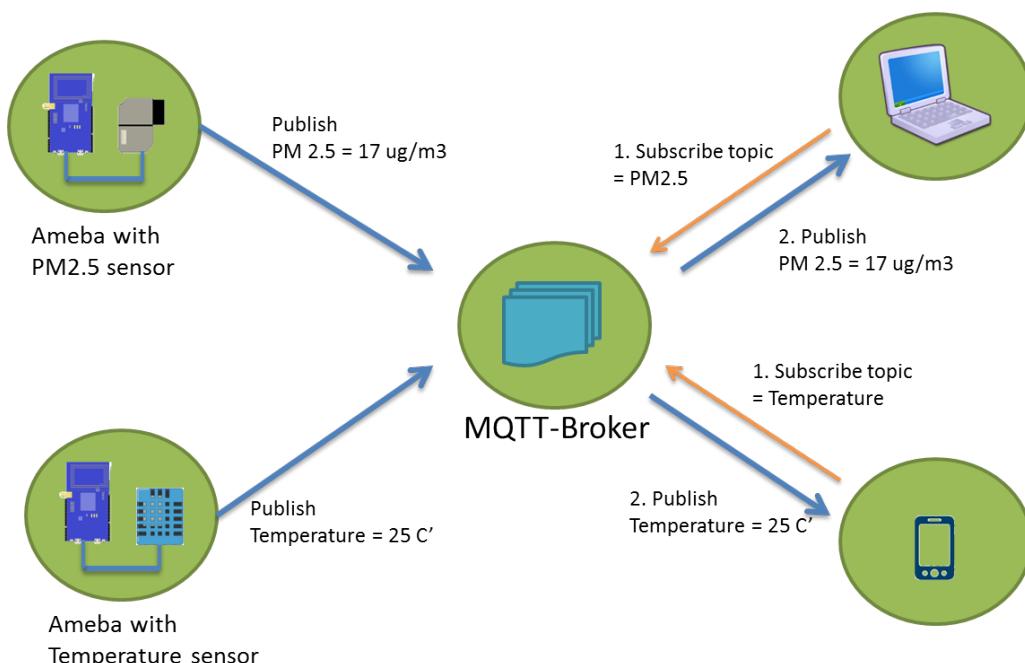
- Online: Chúng ta có thể lập trình online, biên dịch, tải file thực thi về rồi nạp vào thiết bị.
- Offline: Lập trình trên nền tảng linux và make.

2.5.17 MQTT

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

MQTT có mô hình client/server, nơi mà mỗi cảm biến là một khách hàng (client) và kết nối đến một máy chủ, có thể hiểu như một nhà môi giới (broker), thông qua giao thức TCP (Transmission Control Protocol).

MQTT là giao thức định hướng bản tin. Mỗi bản tin là một đoạn rời rạc của tín hiệu và broker không thể nhìn thấy. Mỗi bản tin được publish một địa chỉ, có thể hiểu như một kênh. Client đăng ký vào một vài kênh để nhận/gửi dữ liệu, gọi là subscribe. Client có thể subscribe vào nhiều kênh. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu vào kênh đã đăng ký. Khi một client gửi một bản tin đến một kênh vào đó, gọi là publish.

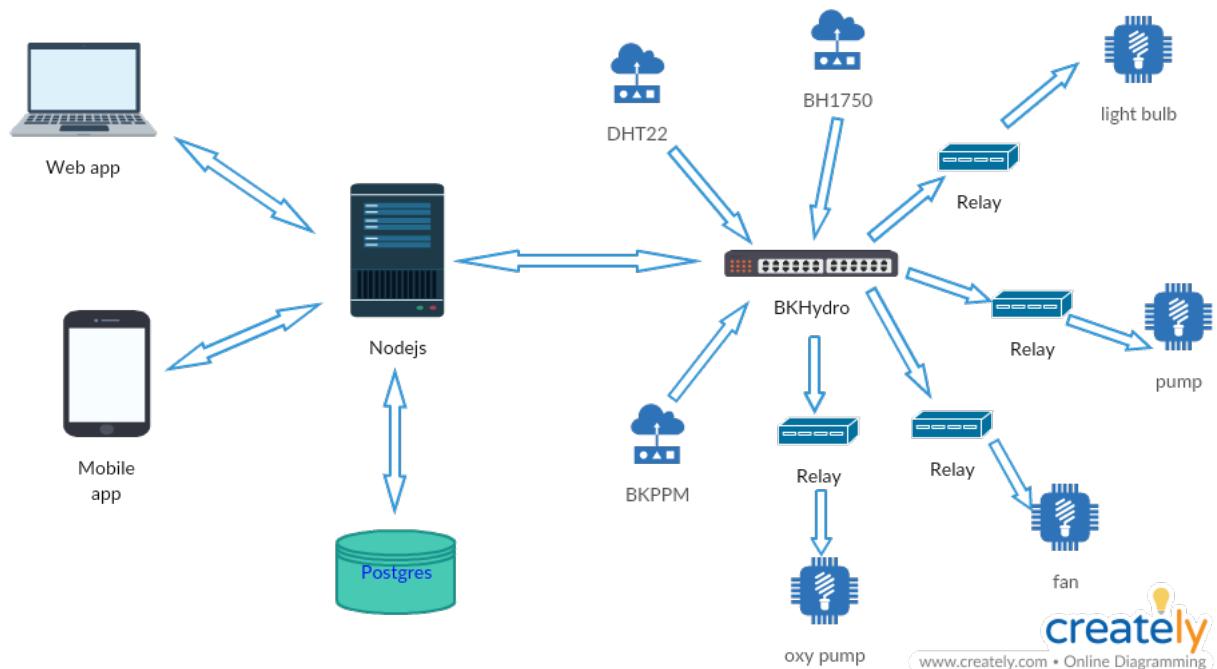


Hình 12: Mô tả các thiết bị liên lạc với nhau thông qua MQTT broker.

3 THIẾT KẾ HỆ THỐNG

3.1 Kiến trúc mô hình hệ thống

Kiến trúc hệ thống bao gồm các thành phần: web server, web app, database và hệ thống board điều khiển.



Hình 13: Kiến trúc hệ thống.

3.2 Hệ thống board điều khiển

3.2.1 Tổng quan mô hình board điều khiển

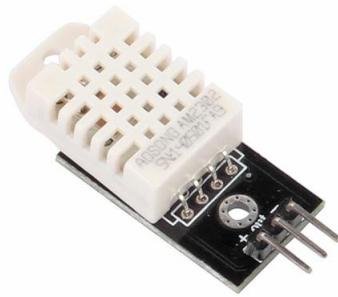
Hệ thống Board điều khiển có các chức năng

- Lấy dữ liệu, lưu trữ và xử lý từ các sensors được đặt trong môi trường thủy canh.
- Điều khiển các hệ thống: tưới tiêu, bơm, chiếu sáng thông qua relay.
- Liên kết với Web Server thông qua môi trường Wifi để trao đổi thông tin:
 - Gửi dữ liệu sensor lên Server.

- Nhận dữ liệu, lịch trình hoạt động của các hệ thống điều khiển.
- Các chức năng liên quan đến Wifi được thực hiện bởi module ESP8266 v1.

Các module, cảm biến của hệ thống

- Cảm biến nhiệt độ, độ ẩm
- Nhiệt độ, độ ẩm được đo bởi module DHT22.



Hình 14: Cảm biến nhiệt độ, độ ẩm.

Thông tin kỹ thuật:

- Nguồn 3-5 VDC.
- Dòng sử dụng: max là 2.5mA.
- Đo tốt ở độ ẩm 0x100%RH, sai số 2-5%.
- Đo tốt ở nhiệt độ 0-50°C, sai số 2°C.
- Tần số lấy mẫu tối đa 0.5Hz (2 giây 1 lần).
- Kích thước 27mm x 58mm x 13.5mm
- 3 chân, khoảng cách 0.1 inch.
- Module PPM

PPM(parts per million) là đơn vị đo mật độ thường dành cho các mật độ tương đối thấp. Nó thường chỉ tỷ lệ của lượng một chất trong tổng số lượng của hỗn hợp chứa chất đó.



Hình 15: Module đo ppm.

- Cường độ ánh sáng

Cường độ ánh sáng được đo bởi module BH1750.



Hình 16: Module BH1750.

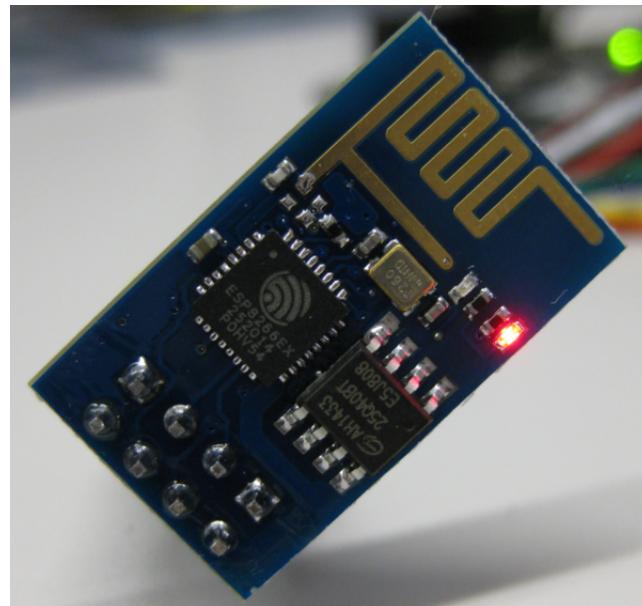
Thông tin kỹ thuật:

- Nguồn 3-5 VDC.
- Protocol giao tiếp: I2C
- Khoảng đo: 1->65535 lux.
- Độ tắt ở nhiệt độ 0-50°C, sai số 2°C.

- Kích cỡ: 21x16x3.3 mm.
- Module Wifi Module ESP8266 V1 là module truyền nhận Wifi đơn giản dựa trên chip ESP8266 SoC (System on Chip) của hãng Espressif. Module này đã được nạp sẵn firmware giúp người dùng giao tiếp với Wifi rất dễ dàng qua tập lệnh AT thông qua giao tiếp UART.

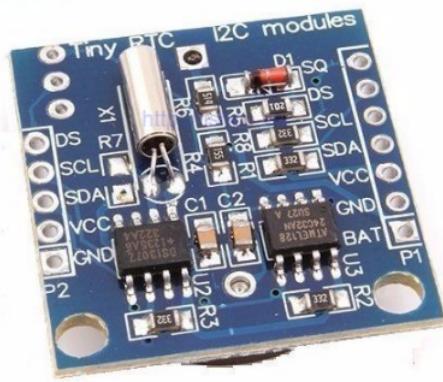
Thông tin kỹ thuật

- Hỗ trợ chuẩn 802.11 b/g/n.
- Wifi 2.4GHz, hỗ trợ WPA/WPA2.
- Chuẩn điện áp hoạt động: 3.3V
- Chuẩn giao tiếp nối tiếp UART với baudrate lên đến 115200.
- Có 3 chế độ hoạt động: Client, Access point, Both Client and Access point.
- Hỗ trợ các chuẩn bảo mật: OPEN, WEP, WPA...
- Hỗ trợ cả 2 giao tiếp TCP và UDP.
- Làm việc như các máy chủ có thể kết nối với 5 máy con.



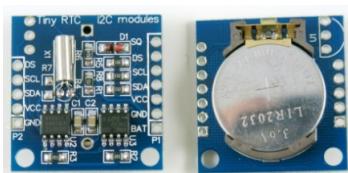
Hình 17: ESP8266.

- Module Realtime



Hình 18: RTC-DS1307

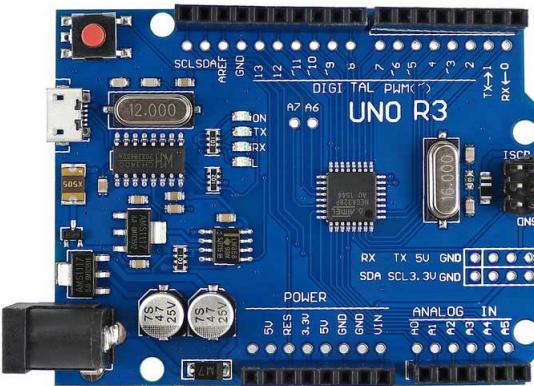
Thời gian thực được đo sử dụng module Realtime clock DS1307. Module có chức năng lưu trữ thông tin ngày tháng năm cũng như giờ phút giây, nó sẽ hoạt động như một chiếc đồng hồ và có thể xuất dữ liệu ra ngoài qua giao thức I2C. Module được thiết kế kèm theo một viên pin có khả năng cung cấp năng lượng tới 10 năm. Module đi kèm với EEPROM AT24C32 có khả năng lưu trữ lên đến 32Kbit.



Hình 19: Module Realtime Clock DS1307.

Thông tin kỹ thuật:

- Nguồn cấp: 5VDC.
- Khả năng lưu trữ: 32K (EEPROM AT24C32).
- Protocol: I2C.
- Có pin chạy độc lập.
- Tần số ra 1 HZ.
- Kích thước 16x22x23mm.



Hình 20: Arduino UNO R3.

- Module Arduino UNO R3

Arduino UNO là board mạch rất phổ biến trong dòng Arduino hiện nay. Nó có ứng dụng rất mạnh mẽ từ đơn giản đến phức tạp như điều khiển đèn LED nhấp nháy, xử lý tín hiệu cho xe điều khiển từ xa, làm một trạm đo nhiệt độ, độ ẩm và hiển thị lên màn hình LCD...

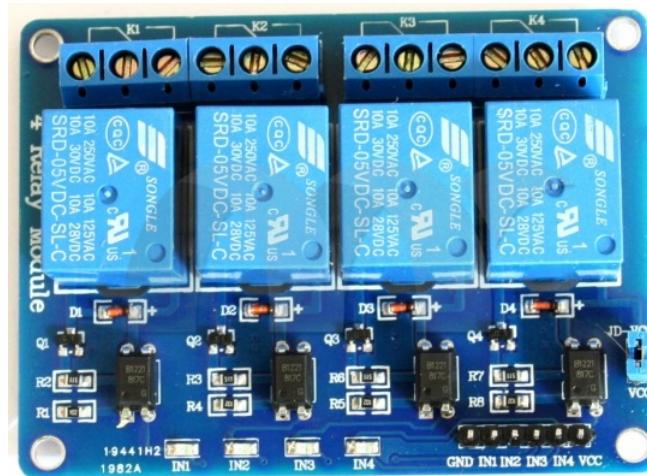
Thông tin kỹ thuật

- Chip điều khiển chính: ATmega328.
- Nguồn: 5VDC từ cổng USB được kết nối với PC.
- Số chân Digital: 14 (hỗ trợ 6 chân PWM).
- Số chân Analog: 6.
- Dòng ra trên chân Digital: 40mA max.
- Dòng ra trên chân 3.3V: 50mA.
- Dung lượng bộ nhớ FLASH: 32KB.
- SRAM: 2KB.
- EEPROM: 1KB.
- Tốc độ: 16MHz.

- Relay

Các actuators (Máy bơm nước, xúc Oxi, led...) được kích thông qua relay.

Thông số kỹ thuật:



Hình 21: Relay.

- Nguồn: 5 VDC.
- Điện áp kích: 5 VDC.
- Thanh ghi dịch

Thanh ghi dịch sử dụng chip 74HC595.

Thông số kỹ thuật:



Hình 22: Thanh ghi dịch.

- Nguồn: 5 VDC.
- Các chân Output: 5 VDC.

- Protocol giao tiếp: SPI.
- Board ARM STM32F03C8T6

Board ARM STM32F103C8T6 được sử dụng để làm board điều khiển các actuators.

Thông số kỹ thuật:



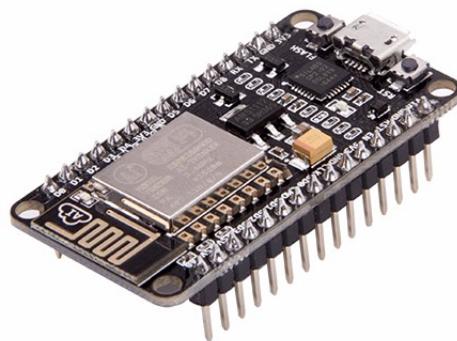
Hình 23: Board ARM STM32F03C8T6.

Level of support	Good
Bootloader	Flash with [1]
Flash	64 KB/128 KB
RAM	20 KB
User LED(s)	PC13 (lights when PC13 is LOW)
User button(s)	None
RTC Crystal	Yes
ST-Link header	Yes
Schematic	media:Vcc-gnd.com-STM32F103C8-schematic.pdf
Manufacturer data	http://vcc-gnd.com/read.php?tid=369&fid=6
Pinout	 media:Bluepillpinout.gif http://reblag.dk.wordpress/wp-content/uploads/2016/07/The-Generic-STM32F103-Pinout-Diagram.pdf

- Board NodeMCU ESP8266

Board NodeMCU được sử dụng để làm board truyền nhận dữ liệu qua wifi.

Thông số kỹ thuật:

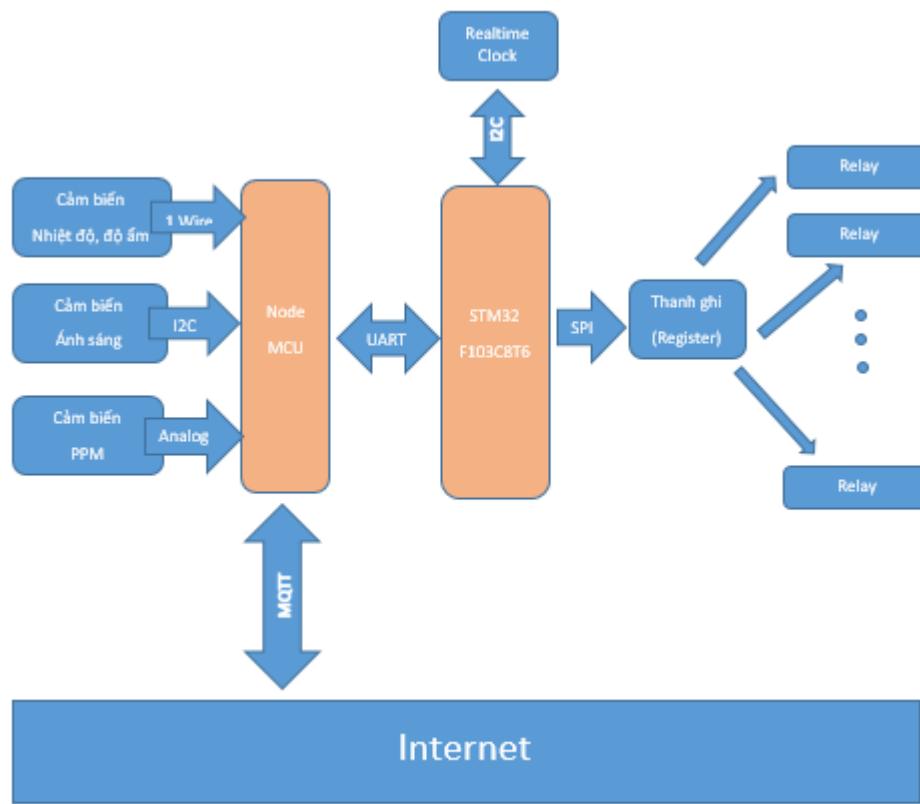


Hình 24: Board NodeMCU ESP8266

- IC chính: ESP8266 WiFi SoC.
- Phiên bản firmware: NodeMCU Lua
- Chip nạp và giao tiếp UART: CP2102.
- GPIO tương thích hoàn toàn với firmware Node MCU.
- Cấp nguồn: 5VDC MicroUSB hoặc Vin.
- GPIO giao tiếp mức 3.3VDC
- Tích hợp Led báo trạng thái, nút Reset, Flash.
- Tương thích hoàn toàn với trình biên dịch Arduino.
- Kích thước: 25 x 50 mm

3.2.2 Sơ đồ hoạt động

- Arduino Uno đóng vai trò là board điều khiển trung tâm, có nhiệm vụ lấy tin từ các cảm biến và các module như: nhiệt độ, độ ẩm, pH và ppm.
- Các sensor được đặt trong môi trường thủy canh, lấy thông tin từ môi trường kết nối trực tiếp với board điều khiển.
- Các thiết bị máy bơm, đèn chiếu sáng, quạt làm mát sẽ được điều khiển bởi board Arduino Uno thông qua relay (chức năng của relay như là công tắc).
- Hệ thống hoạt động theo thời gian thực được lấy từ module RTC-DS1307.
- Hệ thống sử dụng module wifi ESP8266 để kết nối với server thông qua internet, Giao tiếp giữa ESP8266 với Arduino Uno là UART(Serial).



Hình 25: Mô hình tổng quan.

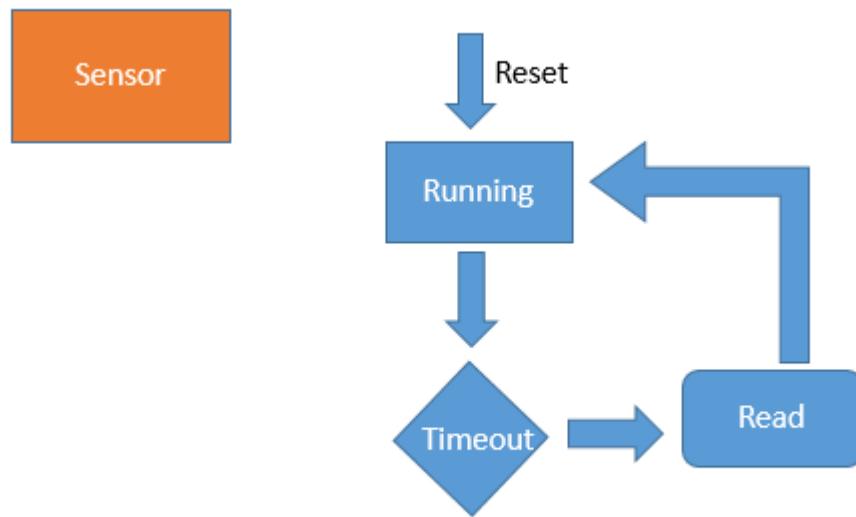
Thu thập dữ liệu cảm biến

Các cảm biến được kết nối trực tiếp với board NodeMCU và lấy dữ liệu trực tiếp.

- Nhiệt độ, độ ẩm: module DHT22 giao tiếp 1 wire với board NodeMCU. Dữ liệu nhiệt độ (C), độ ẩm (
- Cường độ ánh sáng: module BH1750 giao tiếp I2C với board NodeMCU. Dữ liệu cường độ ánh sáng (lux) được đọc sau mỗi (1s).
- Module ppm hoạt động độc lập, đọc dữ liệu từ môi trường dung dịch, xuất đầu ra analog được đọc bởi board NodeMCU sau mỗi (1s).
- Định thời điểm đọc dữ liệu cảm biến sử dụng timer, timeout là 1s.

Hoạt động đọc, quản lý dữ liệu cảm biến sẽ được thực hiện bởi 1 đối tượng:

- Quản lý các thông tin: nhiệt độ, độ ẩm, PPM, ánh sáng.



Hình 26: Hoạt động độc lập, định thời đọc dữ liệu.

- Hoạt động độc lập, định thời đọc dữ liệu theo sơ đồ sau:

Điều khiển các actuators

Thông số các actuator:

Máy bơm nước:

- Nguồn: 12 VDC.
- Công suất:
- Khả năng phun nước độ cao 2m phương thẳng đứng.

Máy sục oxy:

- Nguồn: 12 VDC.
- Công suất:

Đèn chiếu sáng:

- Nguồn: 12 VDC.



Hình 27: Máy bơm nước.



Hình 28: Máy sục oxy.



Hình 29: Đèn led.

- Công suất:

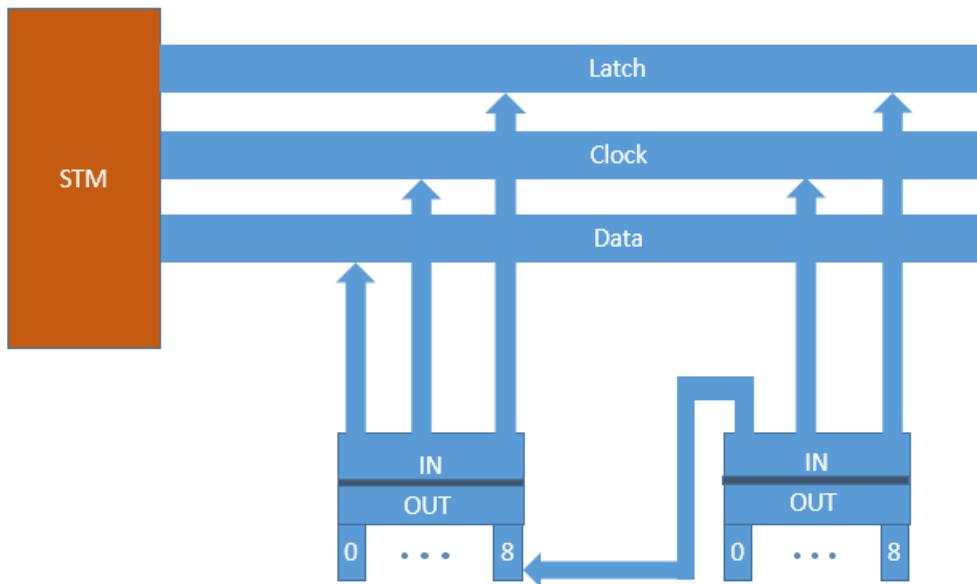
Quạt:

- Nguồn: 12 VDC.

- Công suất:

Cách điều khiển:

Hoạt động của thanh ghi dịch:

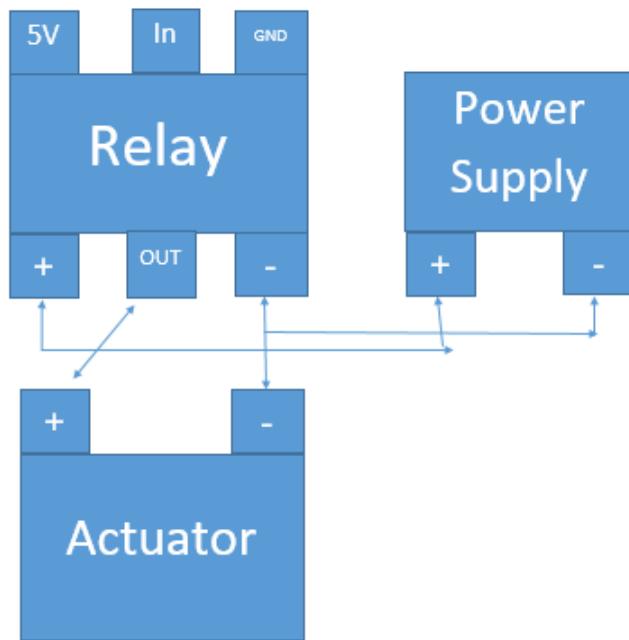


Hình 30: Hoạt động của thanh ghi dịch.

Giao tiếp của thanh ghi dịch với board điều khiển là SPI, các chân điều khiển:

- DATA: chân dữ liệu.
- LATCH: đẩy dữ liệu từ thanh ghi bên trong chip xuống các pinout, cụ thể là 8 chân, điện áp ra là 5V.
- CLOCK: tín hiệu clock lấy từ board điều khiển.

Chân DATA của thanh ghi đầu tiên sẽ nối với chân tín hiệu đầu ra của board điều khiển, DATA của các thanh ghi tiếp theo sẽ lấy từ pinout thứ 9 của thanh ghi liền trước nó. Các actuators sẽ được kích hoạt trực tiếp thông qua relay:



Hình 31: Hoạt động của các actuators.

Tín hiệu điều khiển relay sẽ lấy từ thanh ghi:

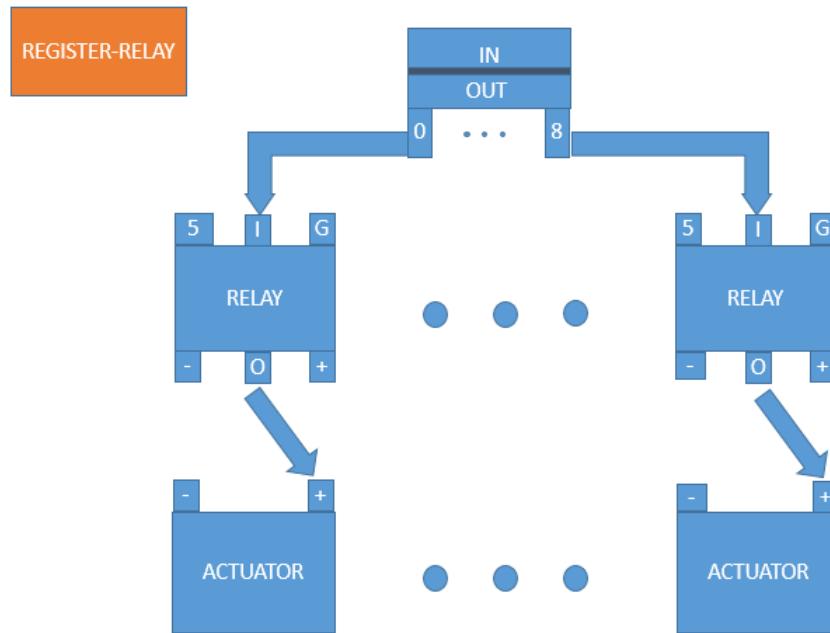
Thời gian thực:

Module Realtime Clock DS1307 sẽ hoạt động độc lập với pin, cù thê, các thông số thời gian thực trong module không bị ảnh hưởng bởi nguồn nuôi của toàn mạch. Thời gian thực (year, month, day, hour, min, sec) có thể được đọc và ghi bởi board điều khiển, cù thê là NodeMCU.

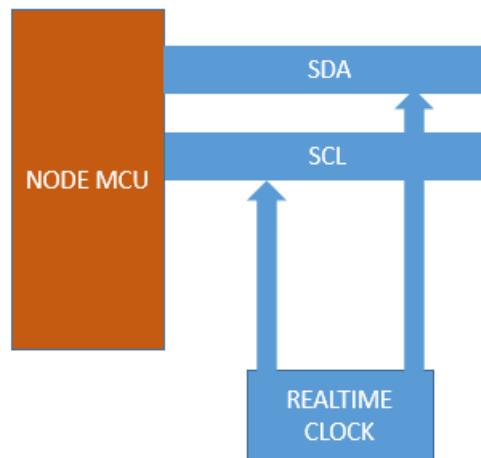
Định thời đọc dữ liệu thời gian thực sử dụng timer, timeout là 1/3s.

Tổ chức, quản lý hoạt động của các actuators.

Chân điều khiển: Các actuator được điều khiển (tắt, mở) thông qua tín hiệu các chân của thanh ghi dịch (74HC595) và relay nên không phụ thuộc vào các pinout của board điều



Hình 32: Hoạt động của tín hiệu điều khiển relay.



Hình 33: Hoạt động Thời gian thực.

khiển (STM32F103C8T6). Có thể tăng số chân điều khiển (cũng như tăng số actuators) bằng cách thêm các thanh ghi dịch, số pinout sử dụng luôn là 3, cụ thể là các chân SPI (DATA, LATCH, CLOCK).

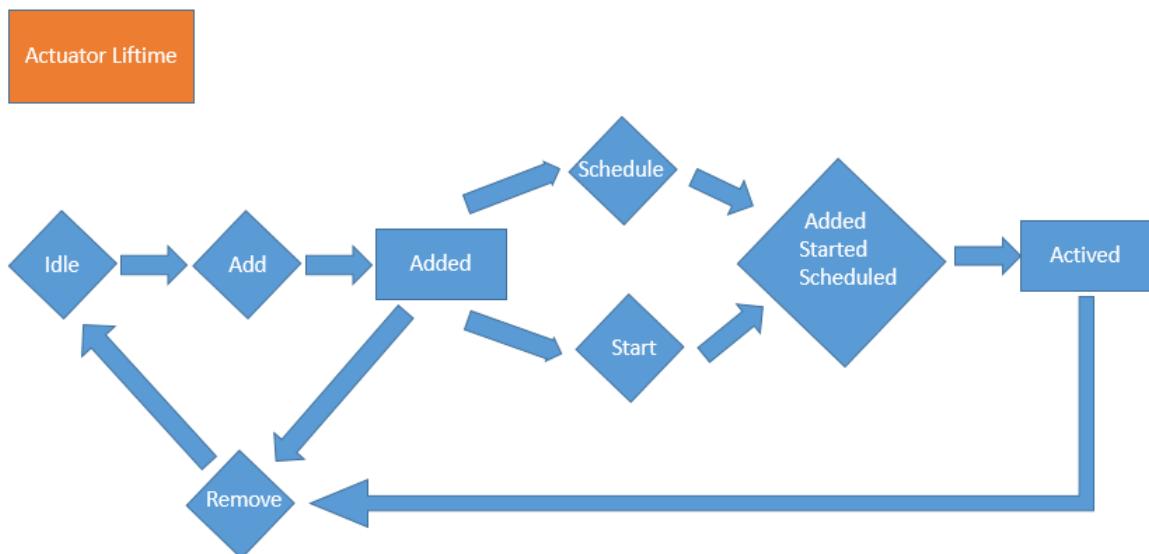
Thời gian biểu: Mỗi actuator sẽ hoạt động theo một thời gian biểu, cụ thể trong thủy

cạnh:

Start	Duration	Repeat Interval	Stop
Thời điểm bắt đầu hoạt động.	Thời lượng actuator được mở.	Thời lượng tắt actuator giữa hai lần mở.	Thời điểm tắt actuator cho đến thời điểm Start kế tiếp.

Dịnh dạng thời gian cho mỗi kiểu như trên là hhmmss. Mỗi thời gian biểu cho mỗi ngày sẽ bao gồm nhiều mẫu như trên, ví dụ với máy bơm: Máy bơm sẽ bắt đầu bơm nước từ 08:00:00 (start), mỗi lần tưới sẽ kéo dài trong 15s (duration), sau đó sẽ tắt trong 10s (repeat interval), lại bật tiếp 15s, cứ thế cho đến 09:00:00 (stop).

Lifetime của mỗi actuator (phần mềm).



Hình 34: Hoạt động thời gian biểu.

Mỗi actuator sẽ là một đối tượng (phần mềm), giải thích lifetime: Ban đầu, sẽ không có actuator nào, các actuator không được sét tĩnh. Khi nhận gói tin (Add) từ web-server, một đối tượng actuator sẽ được tạo ra, là một node trong danh sách liên kết (linked-list) các actuators. Mỗi đối tượng chứa các thông tin:

- Actuator ID: định danh của actuator.

- Primary: là actuator chính hay phụ.
- Schedule: thời gian biểu hoạt động.
- Started: điều kiện để hoạt động.

Sau khi được add (Added), đối tượng sẽ chờ nhận thêm hai gói tin (thứ tự bắt kì) là (Start) và (Schedule) để chuyển sang trạng thái (Actived). Nhận gói (Schedule): đối tượng sẽ lưu lại thông tin schedule. Nhận gói (Start): đối tượng sẽ cập nhập thông tin (Started). Ở trạng thái (Added) và (Actived) khi nhận được gói (Remove) , đối tượng sẽ bị xóa. Ở trạng thái (Actived), đối tượng actuator sẽ hoạt động với thông tin (Schedule) theo thời gian thực đọc được từ module Realtime Clock DS1307.

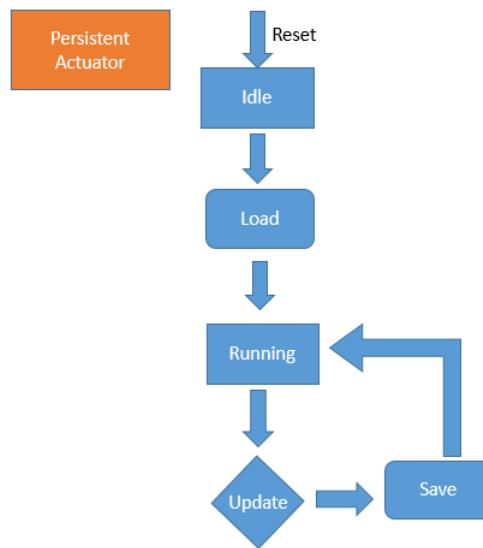
Lưu trữ thông tin actuator.

Để đáp ứng tính độc lập của board điều khiển, các thông tin của mỗi actuator sẽ được lưu lại trong bộ nhớ ROM, tránh trường hợp reset của board (có thể do nhiều nguyên nhân khác nhau: mất điện, crash..). Thông tin actuator được thêm trong lúc board đang hoạt động ở thời điểm ngẫu nhiên (runtime), không xét tinh, đây là ưu điểm của giải pháp này, cho phép mở rộng, thay đổi các actuator. Bộ nhớ lưu trữ là EEPROM, phân vùng lưu trữ sẽ nêu cụ thể trong phần (Lưu trữ dữ liệu) nằm trên board NodeMCU. Cấu trúc dữ liệu lưu trữ được nêu cụ thể trong phần hiện thực.

Dưới đây là cách hoạt động của việc lưu trữ (Save) và tải lại (Load). Qui ước: Các sự kiện (Add), (Start), (Schedule), (Update) (Remove) sẽ được gọi chung là (Update). Các trạng thái (Added), (Actived) được gọi chung là (Running).

Truyền nhận dữ liệu với giao thức MQTT.

- Đối tượng thực hiện công việc này hoạt động trên board NodeMCU với vai trò là một MQTT Client, bao gồm:
 - Lưu trữ các thông tin:
 - * MQTT broker.
 - * MQTT port.
 - * MQTT subscribe channel.

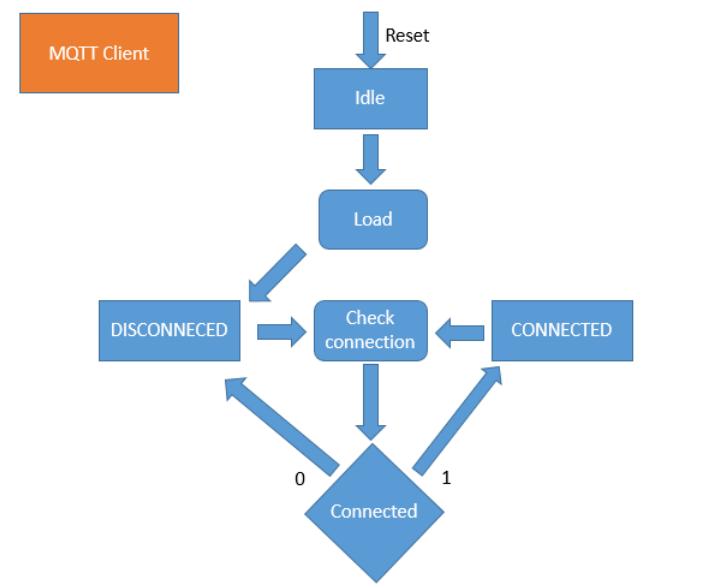


Hình 35: Hoạt động Lưu trữ thông tin actuator.

- * MQTT publish channel.
- Các chức năng:
 - * Publish: truyền dữ liệu tới kênh web-server đăng ký (cũng đóng vai trò là một MQTT Client).
 - * Subscribe: đăng ký kênh nhận dữ liệu từ web-server.
- Các thông tin của đối tượng MQTT Client cũng được lưu trữ trong ROM, nếu trong phần (Lưu trữ dữ liệu), và cho phép cấu hình (Cấu hình WiFi).

Cấu hình, hoạt động của các chế độ WiFi.

- Board NodeMCU cho phép hoạt động ở 3 chế độ:
 - Station: đóng vai trò là một thiết bị kết nối vào mạng wifi tương tự như điện thoại.
 - Access Point: Phát wifi, cho phép thiết bị khác kết nối vào.
 - Both: cả hai chức năng trên.
- Ở dự án này, 2 chế độ được sử dụng là (Station) và (Access Point), cụ thể:
 - Station: Ở chế độ này, board sẽ thực hiện chức năng của MQTT Client.

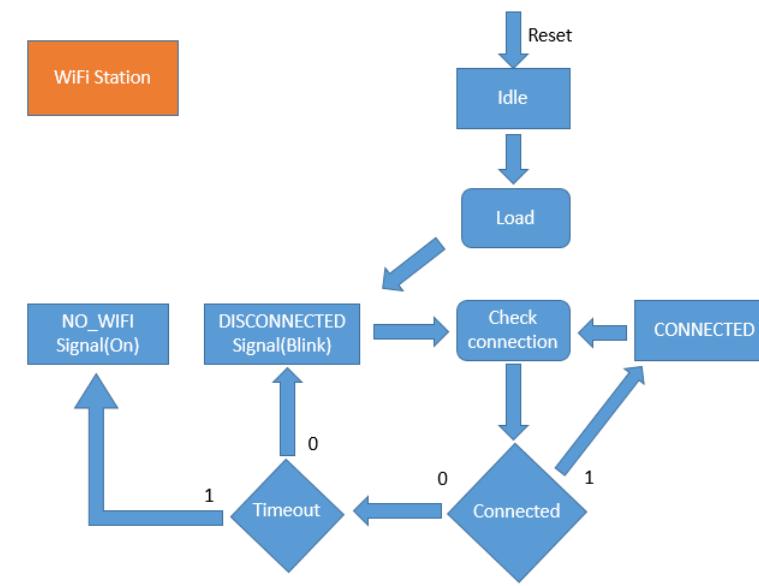


Hình 36: MQTT client.

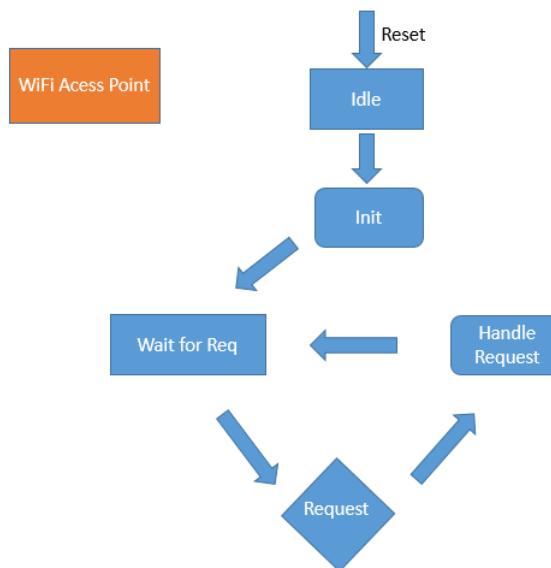
- Access Point: cung cấp chức năng của một Web-server nhỏ, cho phép cấu hình:
 - * Các thông số wifi (tên wifi, mật khẩu) khi board hoạt động ở (Station).
 - * Thông tin của đối tượng MQTT Client (broker, port...).
 - * Các thông tin sau khi cấu hình sẽ được lưu trữ trong EEPROM.
- Chức năng chuyển chế độ sẽ được thực hiện bởi một nút nhấn.
- Các chế độ hoạt động như sau:

Đệm dữ liệu (FIFO)

- Chức năng:
 - Để đảm bảo tính ổn định, đồng bộ giữa quá trình nhận - xử lý các gói tin.
 - Giảm tải thời gian xử lý interrupt (có khả năng gây xung đột các task).
- Các vị trí cần sử dụng bộ đệm:
 - Khi nhận các gói tin serial (uart).



Hình 37: WiFi station.

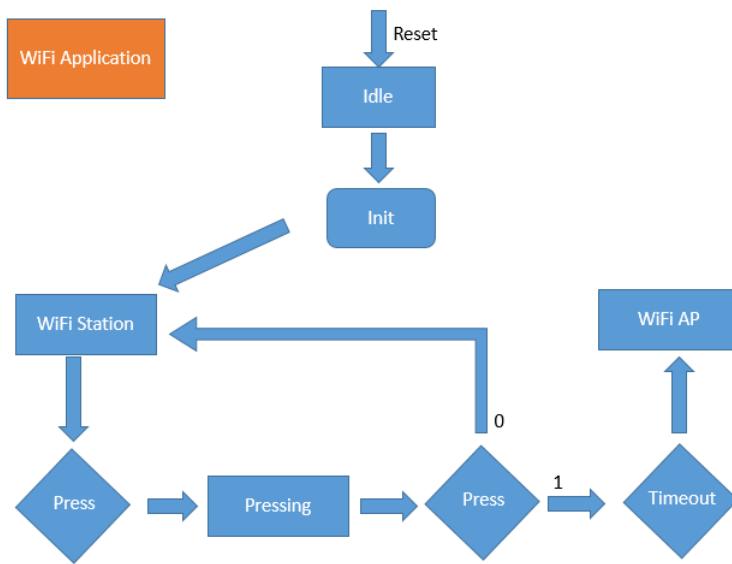


Hình 38: WiFi AP.

- Khi gửi các gói tin MQTT: việc gửi các gói tin MQTT là thông qua môi trường wifi, nên không phải lúc nào cũng đảm bảo tính kết nối giữa client với broker, vì vậy cần bộ đệm để lưu trữ các gói tin trước khi gửi.
- Cấu trúc bộ đệm: các gói tin xếp theo thứ tự FIFO.

Lưu trữ dữ liệu.

- Các thông tin cần được lưu trữ trong EEPROM:

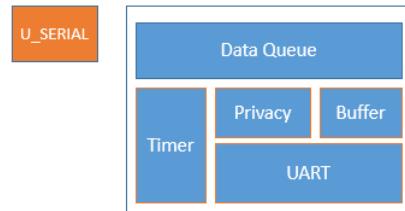


Hình 39: Wifi application.

- Thông tin wifi station: ssid, password.
 - Thông tin MQTT: broker, port, subscribe channel, publish channel.
 - Actuators.
 - Việc lưu trữ và quản lý sẽ do một đối tượng thực hiện.
 - EEPROM đặt trên board NodeMCU, đối tượng Actuators lại hoạt động trên board STM nên sẽ cần một giao thức để tải dữ liệu từ NodeMCU sang STM.

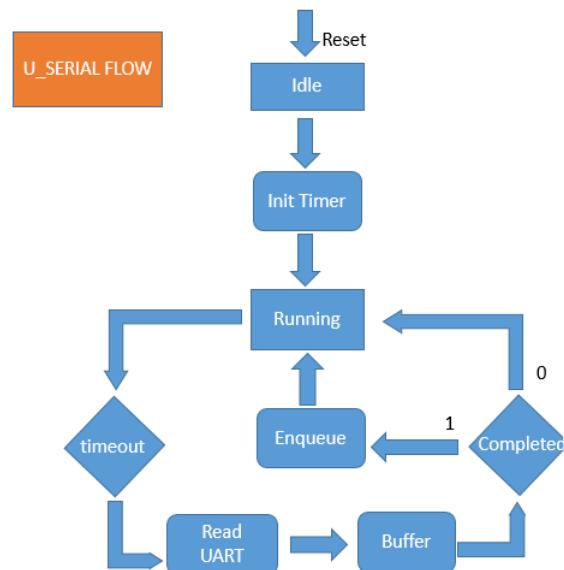
Giao tiếp giữa NodeMCU và STM32F103C8T6.

- Giao thức lớp dưới được sử dụng là UART.
 - Một giao thức đơn giản được xây dựng trên lớp UART nhằm:
 - Truyền, nhận các gói tin lớn.
 - Đảm bảo tính toàn vẹn của gói tin.
 - Cấu trúc giao thức:
 - * Các thành phần chính:
 - . Timer: định thời đọc dữ liệu có sẵn trong UART ở lớp dưới.



Hình 40: Cấu trúc giao thức UART.

- Buffer: bộ đệm tạm thời cho dữ liệu rời rạc.
- Privacy: chính sách truyền nhận.
- Data Queue: gói tin toàn vẹn được đưa vào queue (FIFO), nếu có thẻ trong phần (Đệm dữ liệu).
- UART: giao thức truyền nhận lớp dưới.

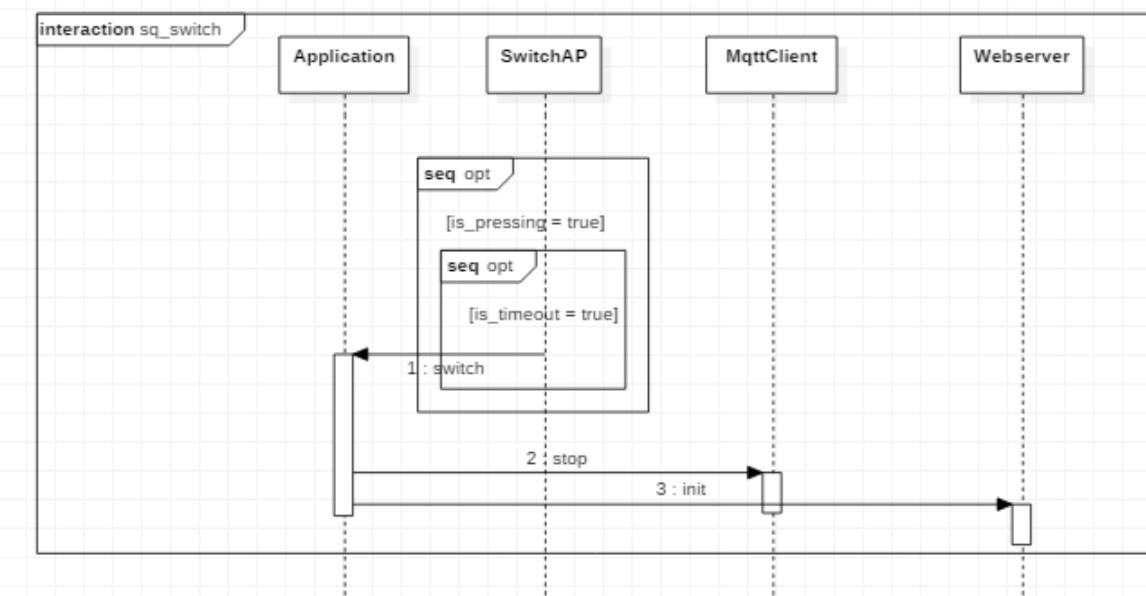


Hình 41: Cách hoạt động của giao thức UART.

3.3 Kiến trúc phần mềm

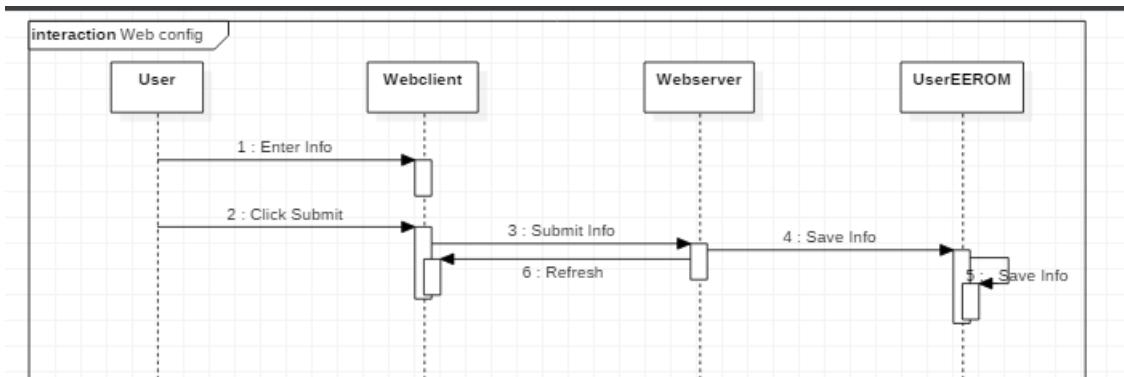
3.3.1 Sequence diagram

Câu hình Wifi



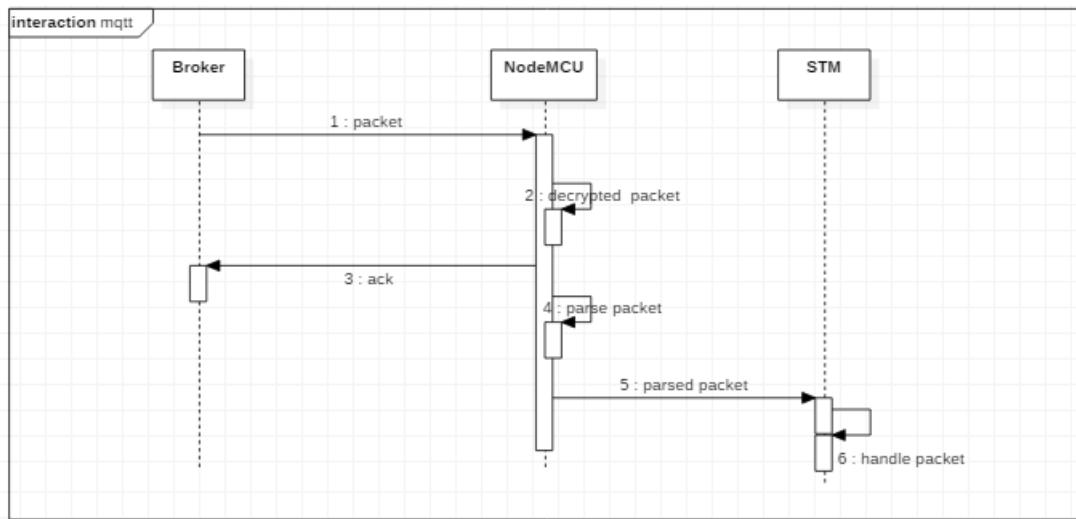
Hình 42: Wifi sequence diagram.

Hoạt động của web-server hoạt động trên board NodeMCU



Hình 43: NodeMCU Server sequence diagram.

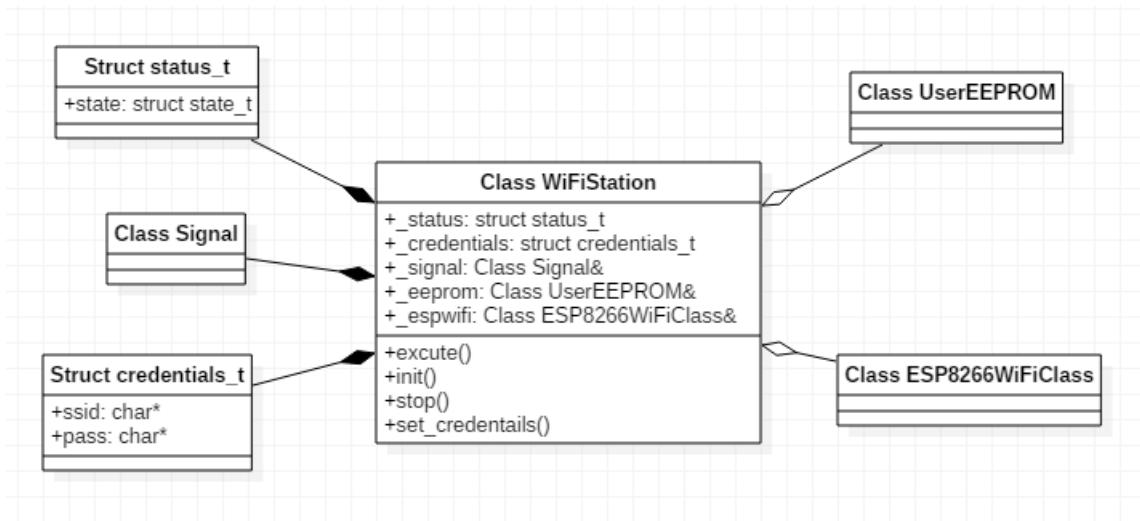
Truyền nhận dữ liệu



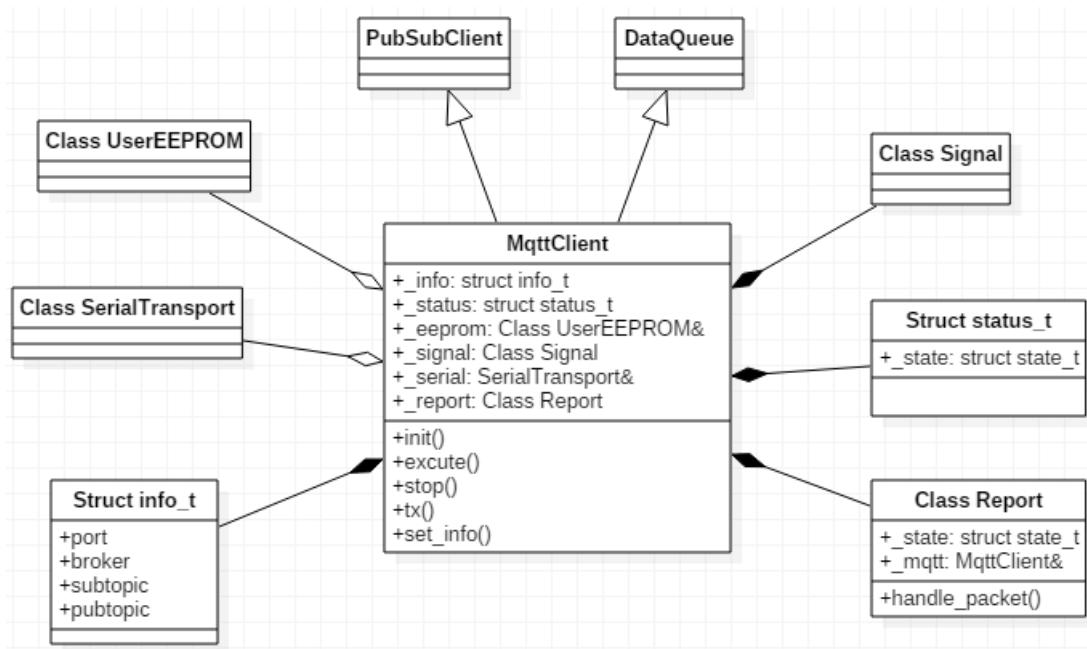
Hình 44: Mqtt sequence diagram.

3.3.2 Class diagram

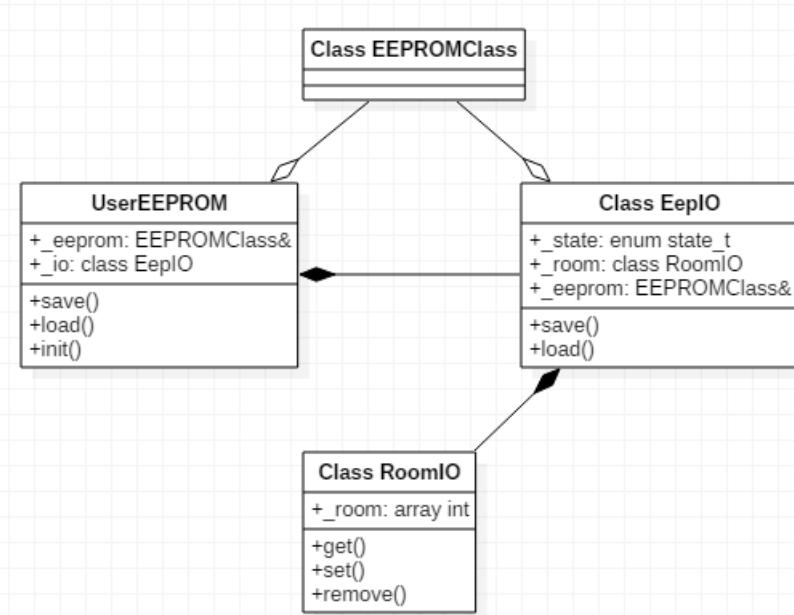
Các class hoạt động trên board NodeMCU



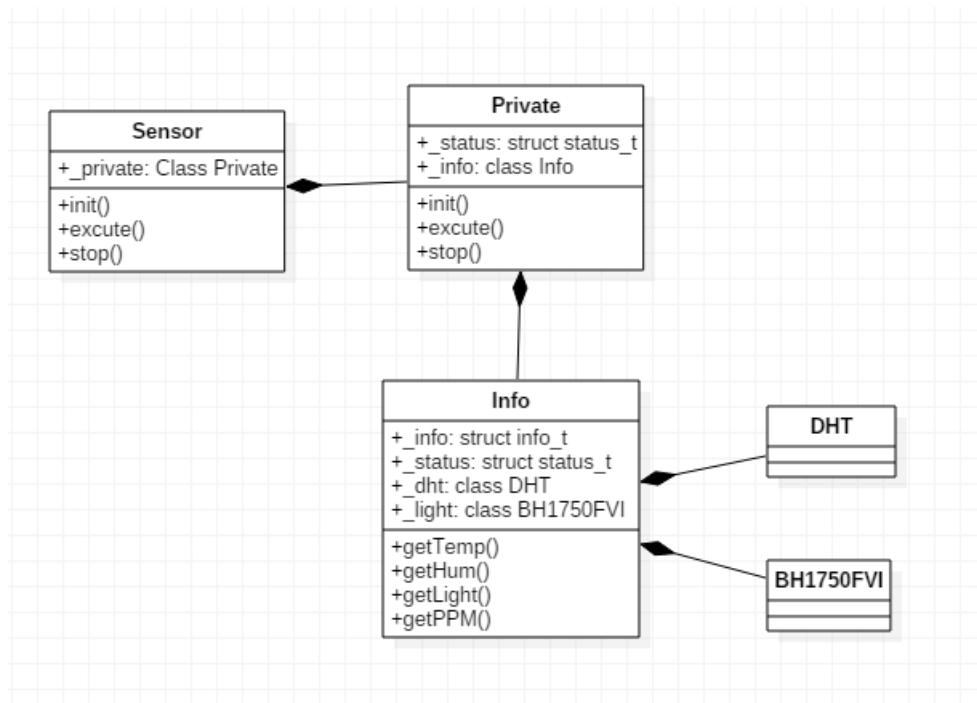
Hình 45: NodeMCU WiFiStation class diagram.



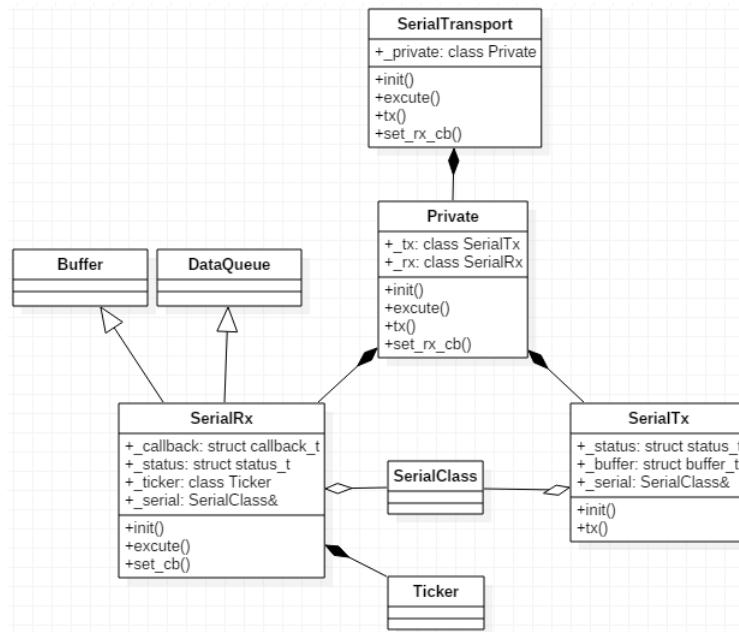
Hình 46: NodeMCU MqttClient class diagram.



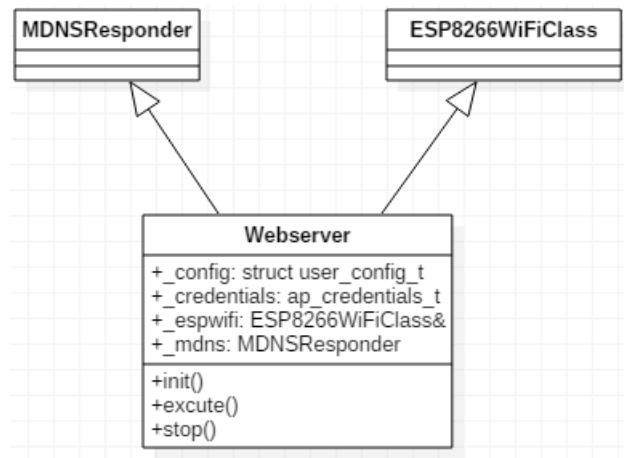
Hình 47: NodeMCU UserEEPROM class diagram.



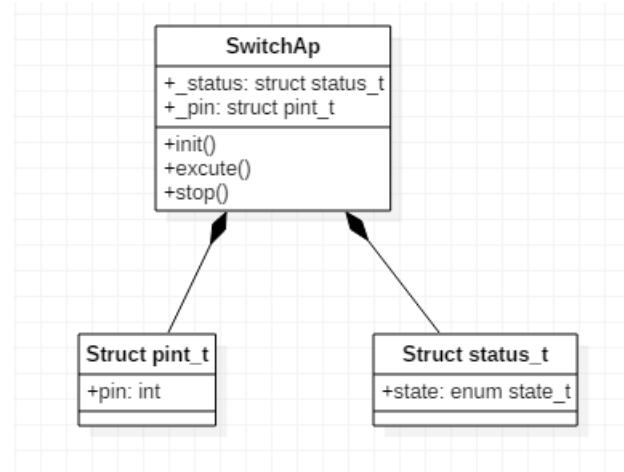
Hình 48: NodeMCU Sensor class diagram.



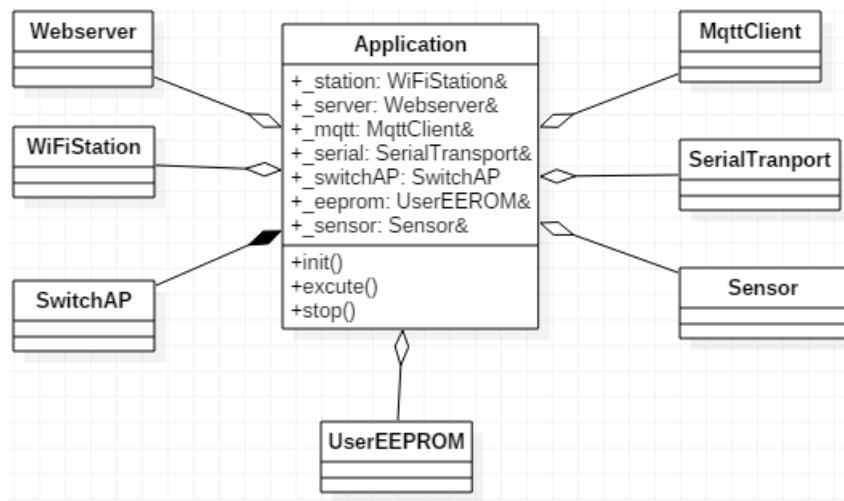
Hình 49: NodeMCU SerialTransport class diagram.



Hình 50: NodeMCU Webserver class diagram.

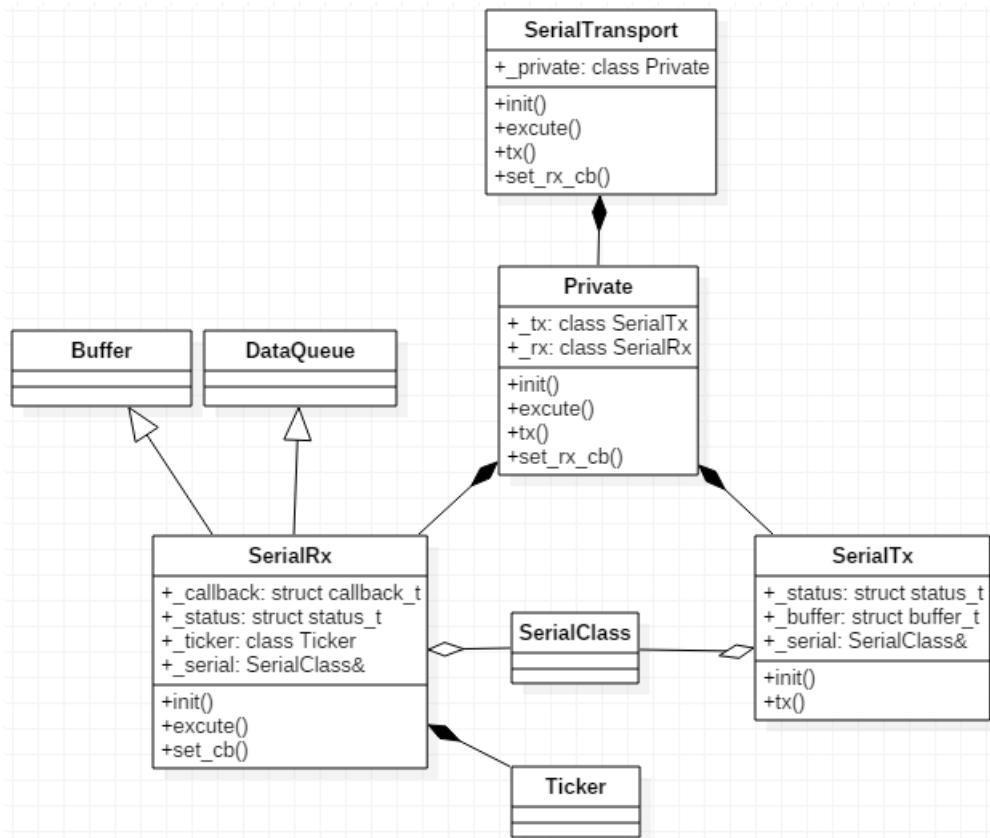


Hình 51: NodeMCU SwitchAP class diagram.

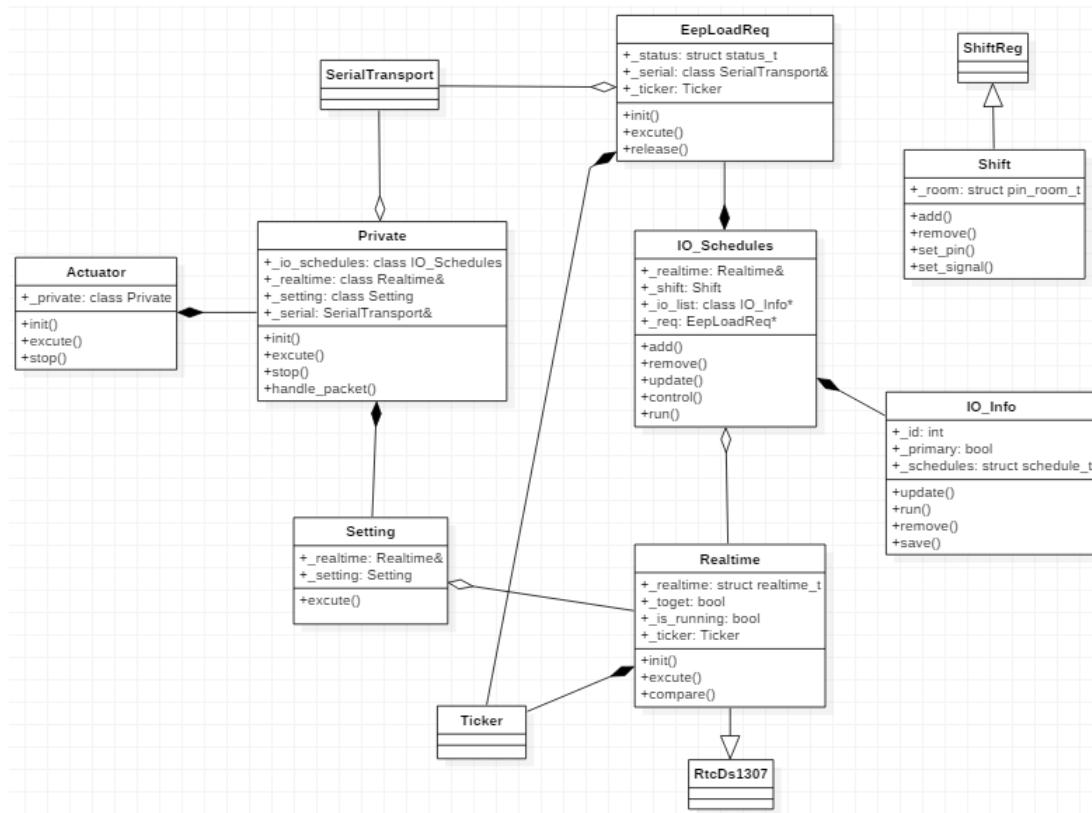


Hình 52: NodeMCU Application class diagram.

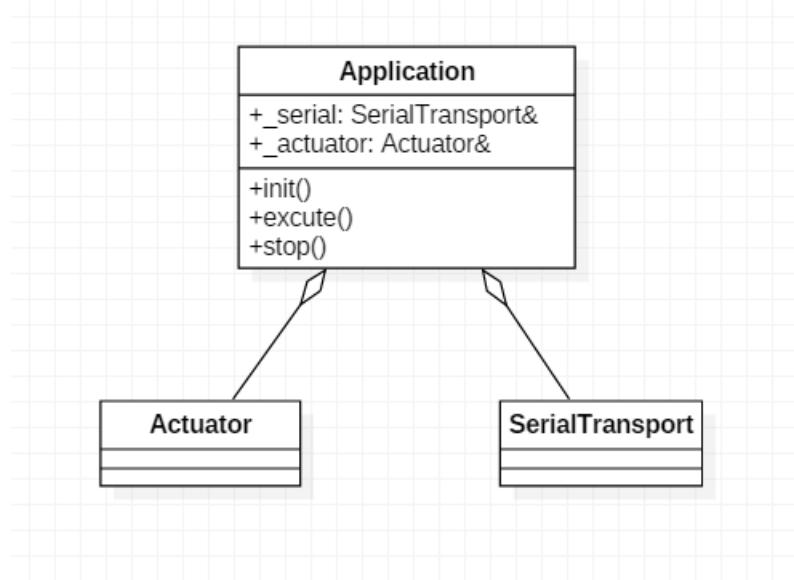
Các class hoạt động trên board STM



Hình 53: STM SerialTransport class diagram.



Hình 54: STM Actuator class diagram.

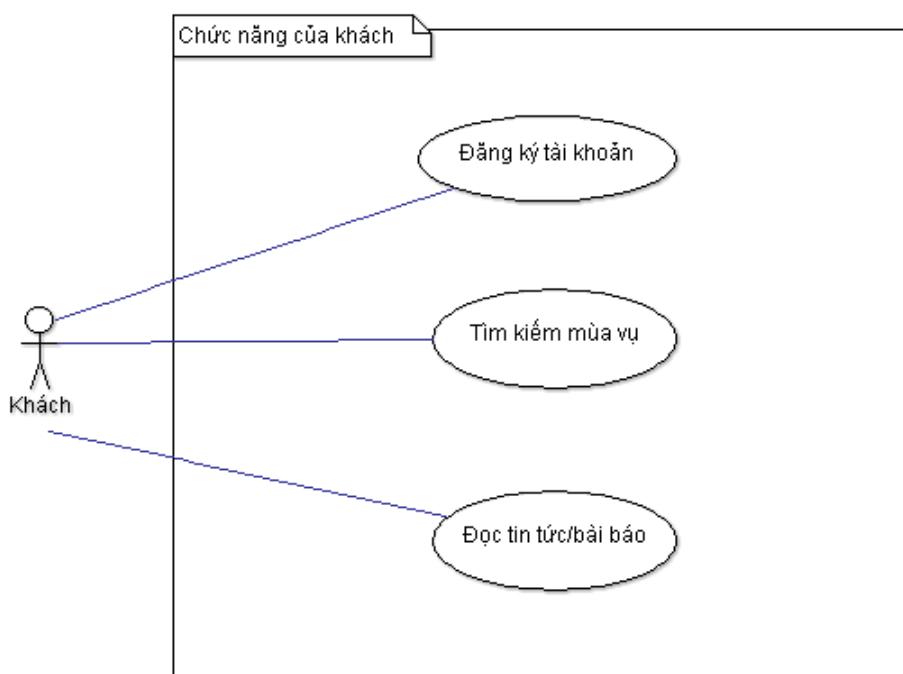


Hình 55: STM Application class diagram.

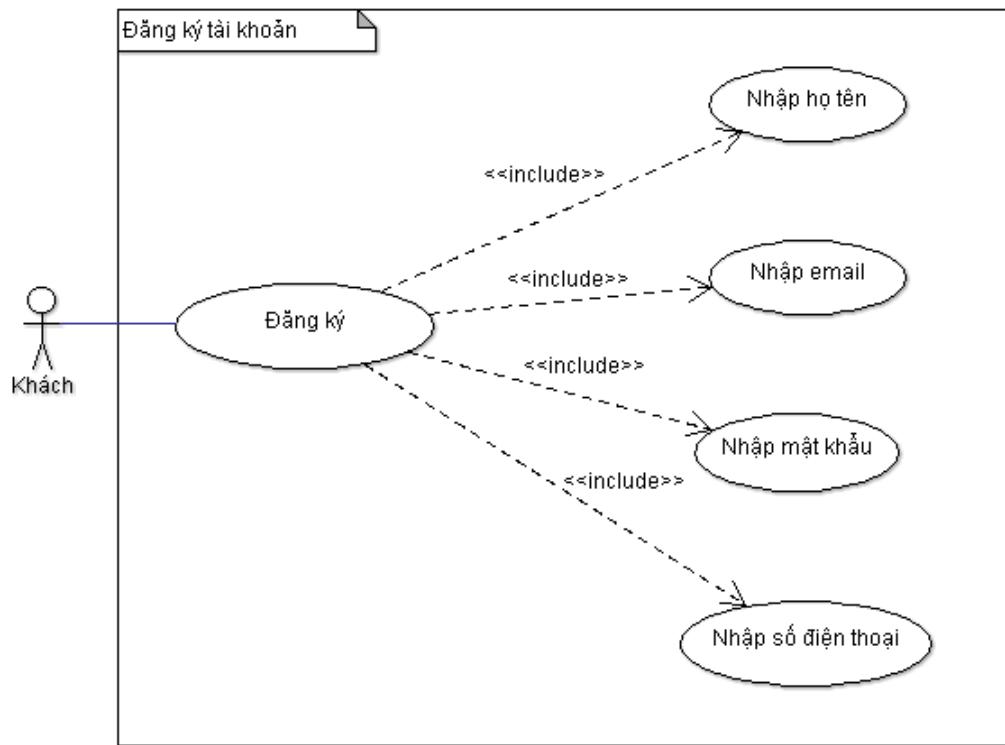
3.4 Ứng dụng web

3.4.1 Sơ đồ use case

Các chức năng của tài khoản khi chưa thực hiện đăng nhập tài khoản.



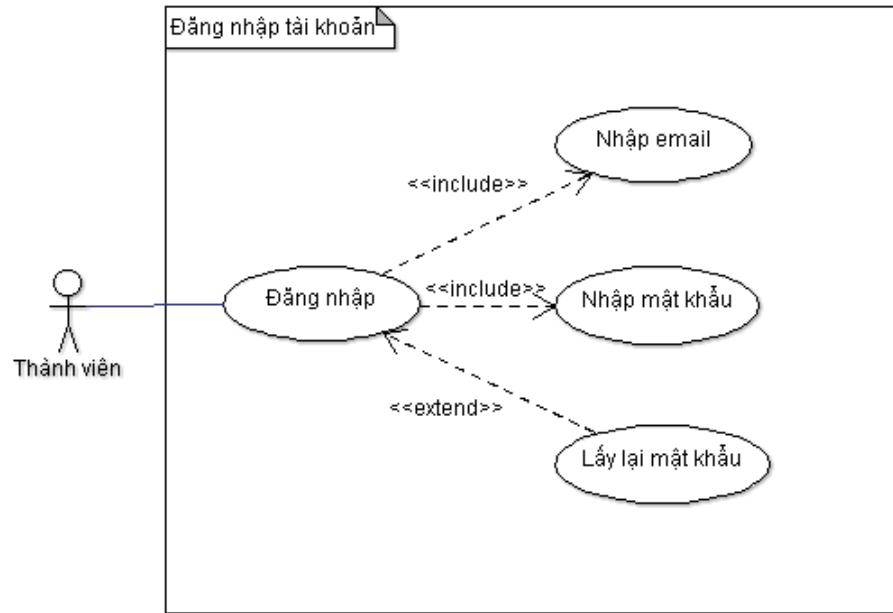
Hình 56: Các chức năng của khách.



Hình 57: Đăng ký.

ID UC_REG_001	Name Đăng ký tài khoản	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Khách			
Mô tả ngắn gọn: khách có thể đăng ký thành viên			
Kịch bản chính:			
1. Người dùng khởi động khung đăng ký bằng cách nhấn nút Register 2. Người dùng nhập các thông tin cần thiết trong khung đăng ký và nhấn Register. 3. Hệ thống tạo người dùng mới 4. Hệ thống thông báo người dùng đã đăng ký thành công.			
Kịch bản phụ:			
2.1. Các ô thông tin bị đe dọa trống hoặc nhập sai định dạng, mật khẩu không khớp, email đã đăng ký thì hệ thống sẽ thông báo.			
Các quan hệ khác:			
<ul style="list-style-type: none"> - Include: nhập họ tên - Include: nhập email - Include: nhập mật khẩu - Include: nhập số điện thoại 			

Bảng 1: Mô tả use case đăng ký.

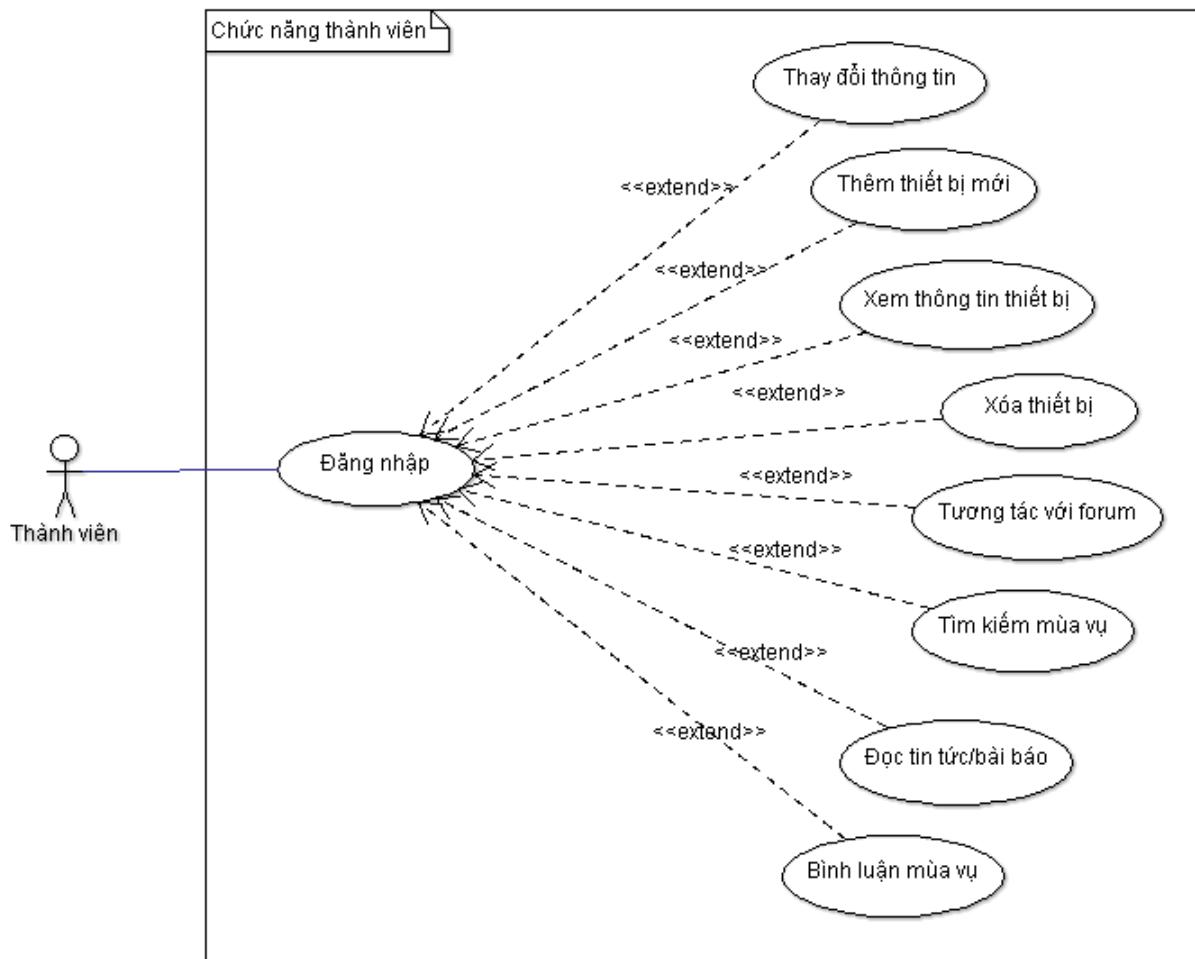


Hình 58: Đăng nhập.

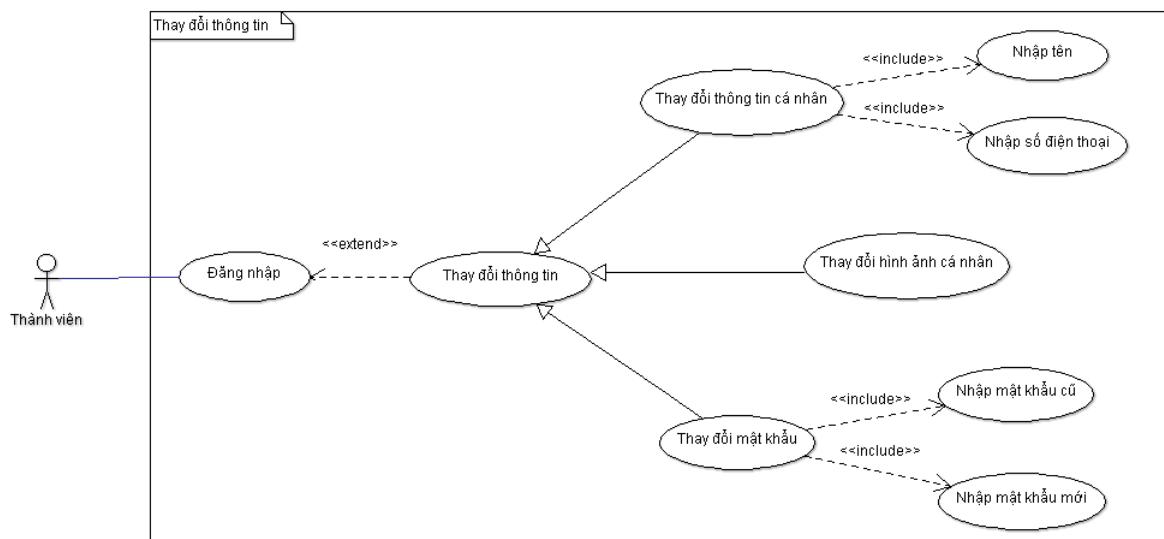
ID	Name	Created by	Created at
UC_LOGIN_001	Đăng nhập	Tạ Chí Tây	20/3/2017
Actor: Khách			
Mô tả ngắn gọn: khách có thể đăng nhập vào hệ thống			
Tiền điều kiện: người dùng đã đăng ký thành công			
Kịch bản chính:			
1. Người dùng khởi động khung đăng nhập bằng cách nhấn vào nút Login. 2. Người dùng nhập email và mật khẩu vào các ô tương ứng và nhấn nút Sign In. 3. Hệ thống xác thực người dùng thông qua email và mật khẩu. 4. Hệ thống thông báo người dùng đã đăng nhập thành công.			
Kịch bản phụ:			
2.1 Tên truy cập không tồn tại trong cơ sở dữ liệu hoặc mật khẩu không chính xác. Hệ thống hiển thị lại biểu mẫu Đăng nhập và hiển thị thông báo lỗi trên màn hình và yêu cầu nhập lại. 5. Nhấn vào nút "Quên mật khẩu". Hệ thống sẽ gửi đường link reset mật khẩu đến email của người dùng và thông báo cho người dùng biết.			
Các quan hệ khác:			
- Include: nhập email - Include: nhập mật khẩu			

Bảng 2: Mô tả use case đăng nhập.

Các chức năng của tài khoản thành viên sau khi đăng nhập.



Hình 59: Các chức năng của thành viên.



Hình 60: Use case thay đổi thông tin.

ID	Name	Created by	Created at
UC_CHANGEINFO_001	Thay đổi thông tin cá nhân	Tạ Chí Tâm	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng nhập tên và số điện thoại mình muốn thay đổi để cập nhật thông tin			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng khởi động khung thay đổi thông tin bằng cách nhấn vào nút Edit. 2. Người dùng nhập tên và số điện thoại mới vào các ô tương ứng và nhấn nút OK. 3. Hệ thống xác thực người dùng và cập nhật thông tin. 4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Người dùng để các ô nhập rỗng, hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập tên - Include: nhập số điện thoại			

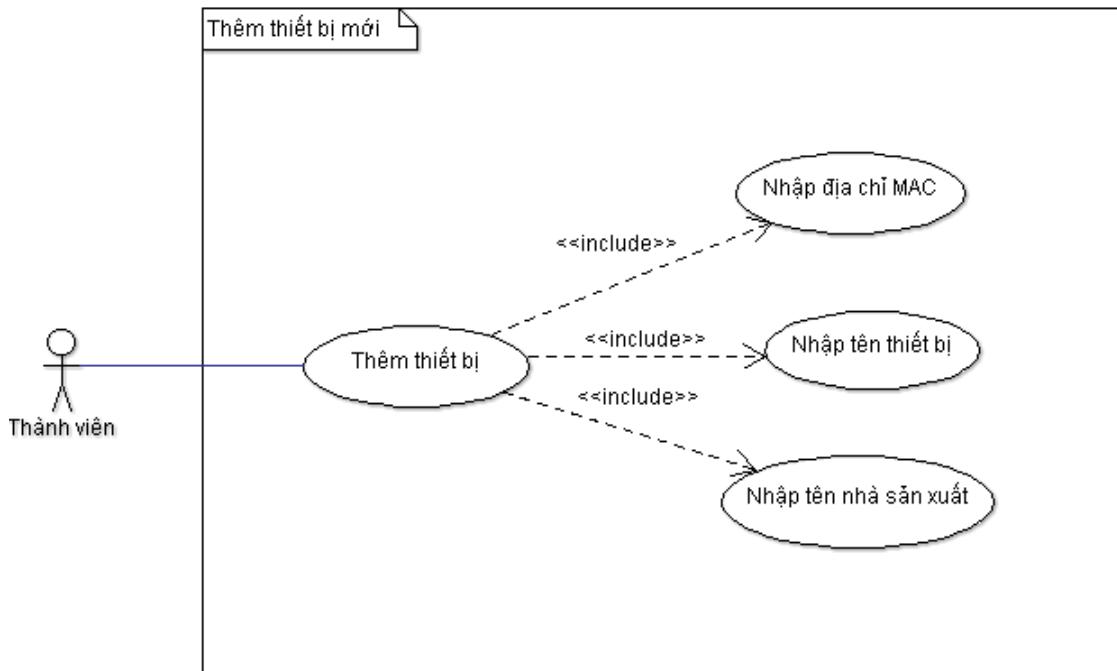
Bảng 3: Mô tả use case hay đổi thông tin cá nhân.

ID	Name	Created by	Created at
UC_CHANGEINFO_002	Thay đổi ảnh cá nhân	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thay đổi ảnh đại diện của mình			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng khởi động khung thay đổi ảnh cá nhân bằng cách nhấn chọn "Change avatar".			
2. Người dùng chọn đường dẫn đến tập tin ảnh mới trên máy và nhấn OK.			
3. Hệ thống xác thực người dùng và cập nhật thông tin.			
4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Người dùng không chọn đường dẫn thì hệ thống sẽ thông báo lỗi.			

Bảng 4: Use case thay đổi hình ảnh cá nhân.

ID	Name	Created by	Created at
UC_CHANGEINFO_003	Thay đổi mật khẩu	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thay đổi mật khẩu của mình			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng khởi động khung thay đổi mật khẩu bằng cách nhấn nút "Change password".			
2. Người dùng nhập thông tin vào các ô tương ứng và nhấn OK.			
3. Hệ thống xác thực người dùng và cập nhật thông tin.			
4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập sai mật khẩu cũ, nhập mật khẩu mới không trùng nhau hoặc để trống các ô thì hệ thống sẽ thông báo.			
Các ràng buộc: thông tin hiển thị dưới dạng ký tự *.			
Các quan hệ khác:			
- Include: nhập mật khẩu cũ			
- Include: nhập mật khẩu mới			

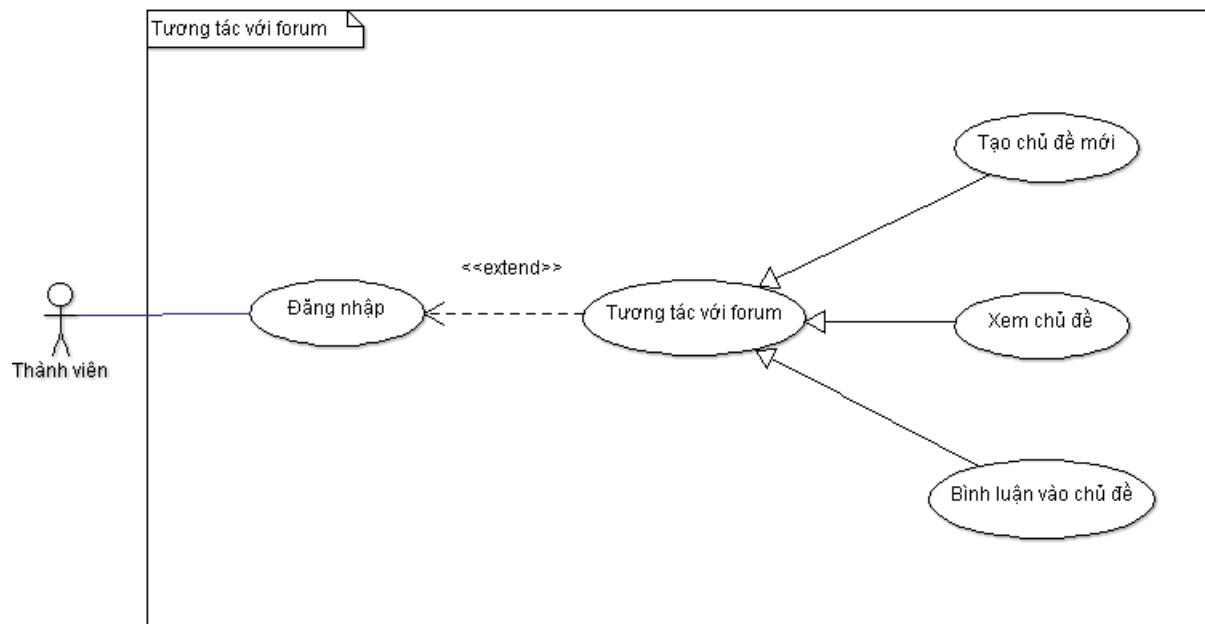
Bảng 5: Mô tả use case thay đổi mật khẩu.



Hình 61: Use case thêm thiết bị.

ID	Name	Created by	Created at
UC_ADDDEVICE_001	Thêm board điều khiển	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thêm board điều khiển mới.			
Tiền điều kiện: người dùng đã đăng nhập thành công			
Kịch bản chính:			
1. Người dùng nhấn vào nút thêm thiết bị mới trong trang quản lý board điều khiển. 2. Người dùng nhập thông tin vào các ô tương ứng và nhấn OK. 3. Hệ thống xác thực người dùng và thêm board điều khiển mới. 4. Hệ thống thông báo người dùng đã thay đổi thành công.			
Kịch bản phụ:			
2.1 Địa chỉ MAC nhập không đúng định dạng, các ô nhập để trống hệ thống sẽ báo lỗi.			
Các ràng buộc: địa chỉ MAC phải đúng định dạng.			
Các quan hệ khác:			
<ul style="list-style-type: none"> - Include: nhập MAC - Include: nhập tên board điều khiển - Include: nhập tên nhà sản xuất 			

Bảng 6: Mô tả use case thêm thiết bị.



Hình 62: Use case tương tác với diễn đàn.

ID	Name	Created by	Created at
UC_FORUM_001	Tạo bài viết mới	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: thành viên tạo một bài viết mới.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang ở mục Bài viết			
Kịch bản chính:			
1. Người dùng khởi động khung tạo bài viết mới bằng cách nhấn nút New article. 2. Người dùng nhập tiêu đề của bài viết, nội dung cụ thể và nhấn OK. 3. Hệ thống xác thực người dùng và tạo bài viết mới. 4. Hệ thống thông báo người dùng đã tạo thành công.			
Kịch bản phụ:			
2.1 Các ô thông tin bị để trống thì hệ thống sẽ thông báo lỗi.			

Bảng 7: Mô tả use case tạo bài viết mới.

ID UC_FORUM_002	Name Xóa bài viết	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Mod			
Mô tả ngắn gọn: mod thực hiện xóa bài viết.			
Tiền đề kiện: người dùng đã đăng nhập thành công và có quyền là mod			
Kịch bản chính:			
1. Người dùng nhấn vào nút xóa bài viết trên danh sách các bài viết. 2. Hệ thống hỏi lại người dùng có chắc chắn hay không. 3. Hệ thống xóa bài đăng nếu người dùng nhấn OK. 4. Hệ thống thông báo người dùng đã xóa thành công.			

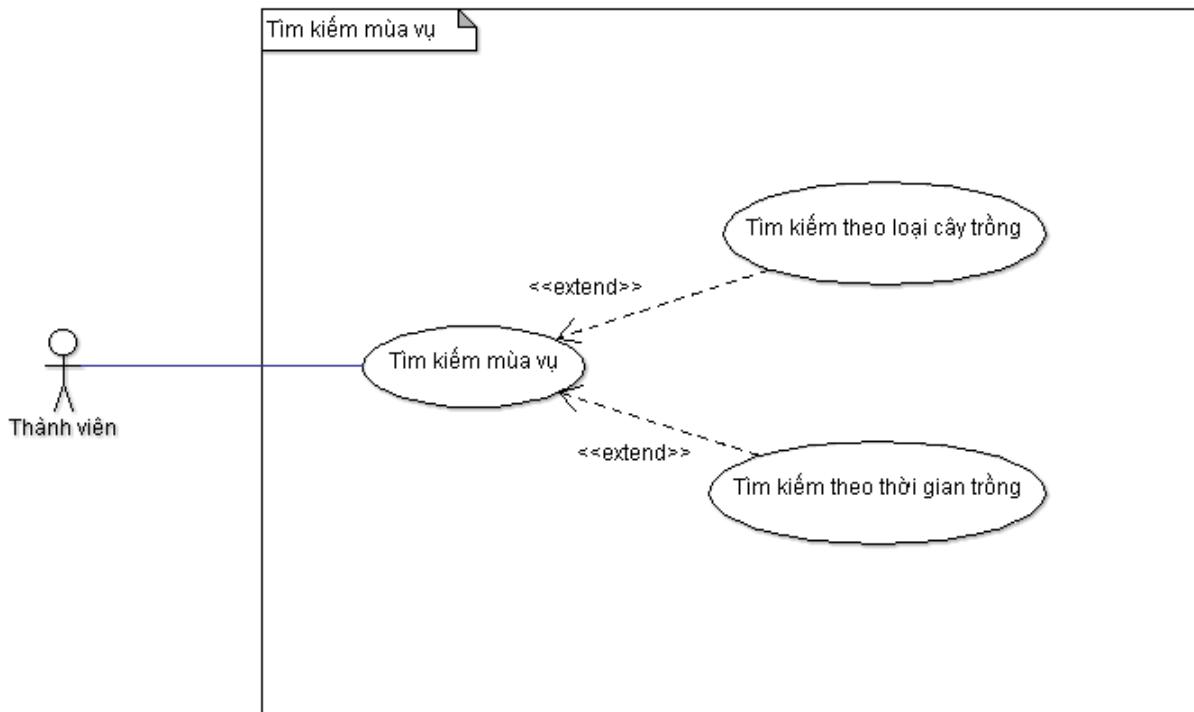
Bảng 8: Mô tả use case xóa bài viết.

ID UC_FORUM_003	Name Bình luận bài viết	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: thành viên tạo một bình luận mới trong bài viết.			
Tiền đề kiện: người dùng đã đăng nhập thành công và đang xem bài viết.			
Kịch bản chính:			
1. Người dùng nhấn chọn vào nút bên luận trên bài đăng đang xem. 2. Người dùng nhập nội dung bình luận của mình vào bài đăng và nhấn OK. 3. Hệ thống xác thực người dùng và tạo bình luận mới.			
Kịch bản phụ:			
2.1 Các ô thông tin bị để trống thì hệ thống sẽ thông báo lỗi.			

Bảng 9: Mô tả use case thêm bình luận bài viết.

ID UC_REMOVEDEVICE_001	Name Xóa board điều khiển	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng xóa board điều khiển của mình.			
Tiền đề kiện: người dùng đã đăng nhập thành công và đang ở trang hiển thị các board điều khiển.			
Kịch bản chính:			
1. Người dùng nhấn nút xóa một board điều khiển trong danh sách. 2. Hệ thống hiển thị thông báo xác nhận xóa board điều khiển của người dùng. 3. Người dùng chọn OK để xóa đi board điều khiển muốn xóa.			
Kịch bản phụ:			
2.1 Người dùng chọn hủy để hủy xóa board điều khiển.			

Bảng 10: Mô tả use case xóa thiết bị.



Hình 63: Use case tìm kiếm mùa vụ.

ID UC_FIND_001	Name Tìm kiếm mùa vụ	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thực hiện Tìm kiếm mùa vụ để xem thông tin mùa được chia sẻ.			
Tiền điều kiện: người dùng đã đăng nhập thành công.			
Kịch bản chính:			
1. Người dùng nhấn vào nút tìm kiếm trong trang của mình sau khi đã đăng nhập. 2. Người dùng nhập vào thông tin của mùa vụ cần tìm kiếm: tìm kiếm theo tên loại cây trồng, tìm kiếm theo tháng... 3. Hệ thống thực hiện tìm kiếm các mùa vụ thỏa điều kiện tìm kiếm của người dùng và hiển thị trong trang kết quả tìm kiếm. 4. Người sử dụng nhấn chọn vào mục mong muốn để xem thông tin của mùa vụ.			
Kịch bản phụ:			
3.1 Hệ thống không hiển thị kết quả tìm kiếm nào phù hợp với yêu cầu của người dùng.			

Bảng 11: Mô tả use case tìm kiếm mùa vụ.

ID	Name	Created by	Created at
UC_FIND_002	Gửi bình luận mùa vụ	Tạ Chí Tây	20/3/2017

Actor: Thành viên

Mô tả ngắn gọn: người dùng thực hiện bình luận mùa vụ của người khác.

Tiền điều kiện: người dùng đã đăng nhập thành công và đang xem thông tin mùa vụ của người khác chia sẻ.

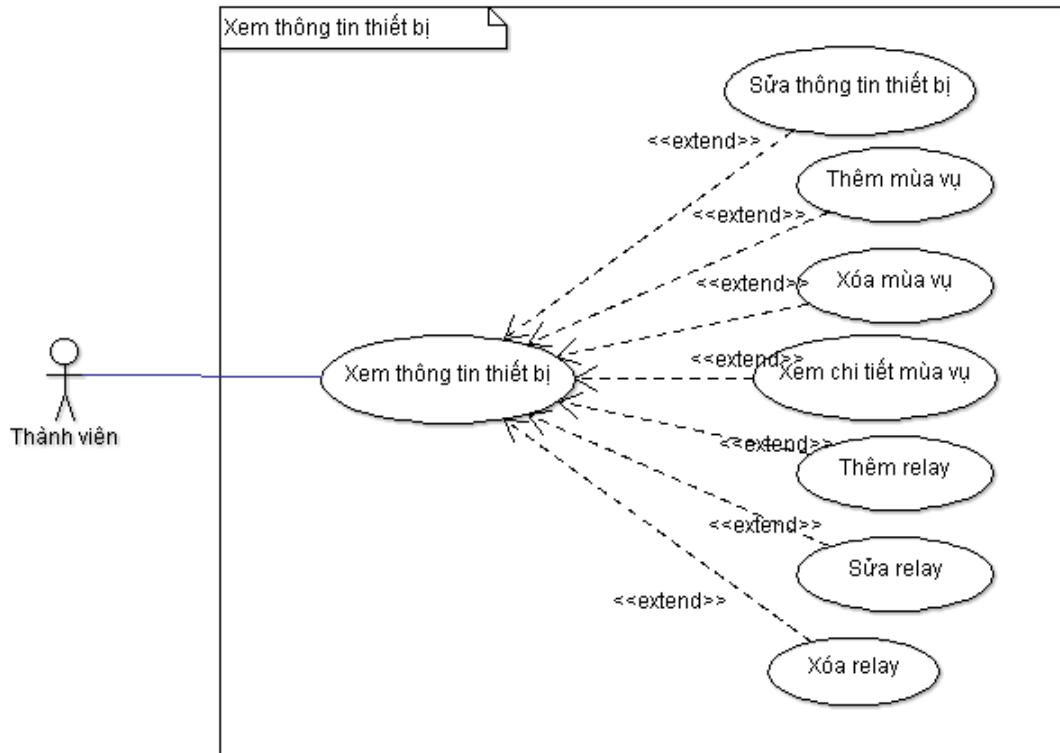
Kịch bản chính:

1. Người dùng nhập bình luận cho mùa vụ.
2. Người dùng chọn điểm đánh giá cho mùa vụ.
3. Người dùng nhấn nút để gửi bình luận lên server.
4. Hệ thống hiển thị thông báo người dùng gửi bình luận thành công.

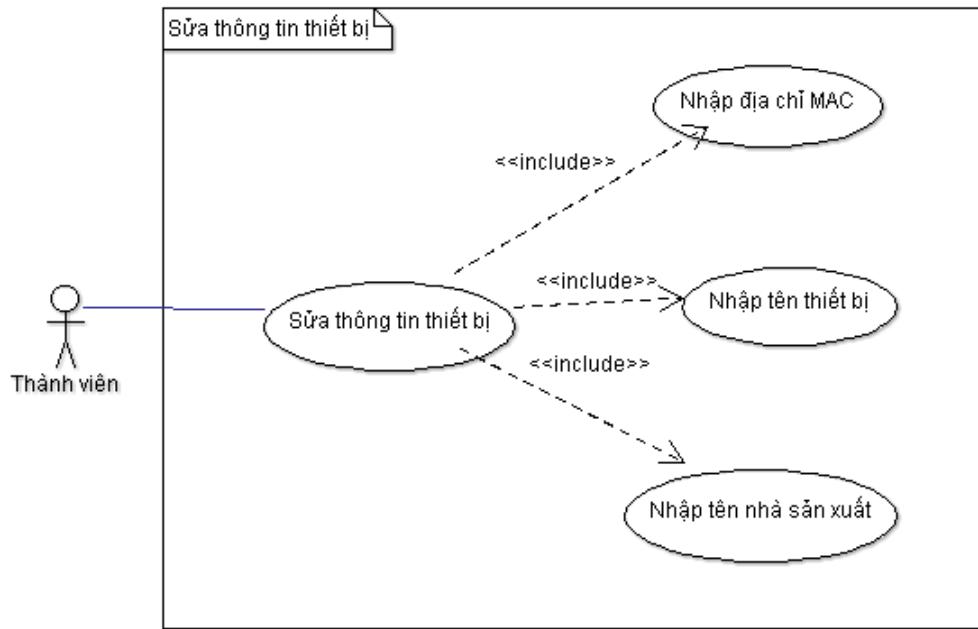
Kịch bản phụ:

- 4.1 Hệ thống hiển thị thông báo người dùng gửi không thành công do có lỗi.

Bảng 12: Mô tả use case gửi bình luận mùa vụ.



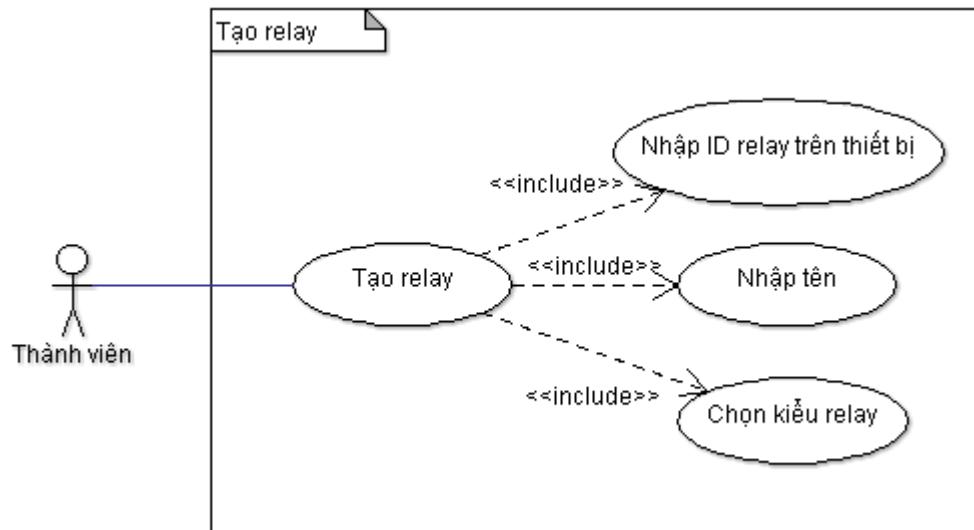
Hình 64: Use case xem thông tin thiết bị.



Hình 65: Use case sửa thông tin thiết bị.

ID	Name	Created by	Created at
UC_DEVICE_001	Sửa thông tin board điều khiển	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thực hiện chỉnh sửa thông tin của một board điều khiển của mình.			
Tiền điều kiện: người dùng đã đăng nhập thành công và đang ở trang board điều khiển.			
Kịch bản chính:			
1. Người dùng nhấn vào nút chỉnh sửa thông tin board điều khiển. 2. Người dùng nhập vào các thông tin cập nhật mới về board điều khiển của mình và nhấn OK. 3. Hệ thống thực hiện cập nhật thông tin của board điều khiển mới. 4. Hệ thống thông báo người dùng đã cập nhật thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập MAC - Include: nhập tên thiết bị - Include: nhập tên nhà sản xuất			

Bảng 13: Mô tả use case sửa thông tin thiết bị.



Hình 66: Use case tạo relay.

ID UC_RELAY_001	Name Tạo relay	Created by Tạ Chí Tây	Created at 20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng thêm relay mới vào hệ thống.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin thiết bị.			
Kịch bản chính:			
1. Người dùng nhấn vào nút tạo mới relay trên trang thông tin của thiết bị. 2. Người dùng nhập các dữ liệu cần thiết: ID của relay trên thiết bị, tên, loại relay. 3. Hệ thống tạo một relay mới. 4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			

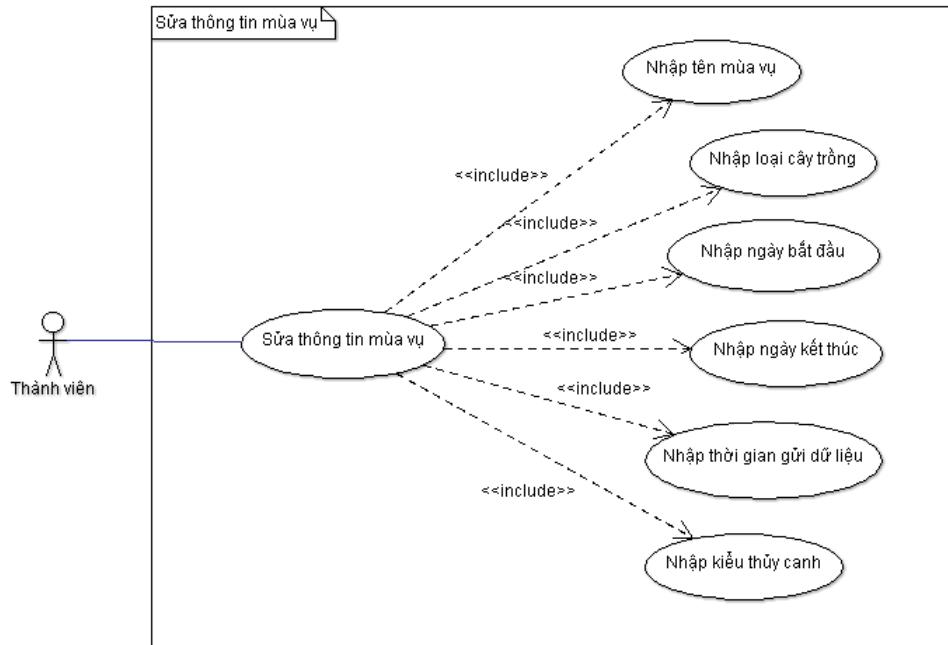
Bảng 14: Mô tả use case tạo relay.

ID	Name	Created by	Created at
UC_RELAY_002	Xóa relay	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng xóa relay trên thiết bị.			
Tiền đề kiện: người dùng đã đăng nhập thành công và đang ở trang hiển thị thông tin thiết bị (có danh sách các relay).			
Kịch bản chính:			
1. Người dùng nhấn nút xóa một relay trong danh sách. 2. Hệ thống hiển thị thông báo xác nhận xóa relay của người dùng. 3. Người dùng chọn OK để xóa đi relay muốn xóa.			
Kịch bản phụ:			
2.1 Người dùng chọn hủy để hủy xóa relay.			

Bảng 15: Mô tả use case xóa relay.

ID	Name	Created by	Created at
UC_RELAY_003	Sửa relay	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng sửa trạng thái của relay trên thiết bị.			
Tiền đề kiện: người dùng đã đăng nhập thành công và đang ở trang hiển thị thông tin thiết bị (có danh sách các relay).			
Kịch bản chính:			
1. Người dùng nhấn nút chuyển trạng thái của relay hoặc nhấn nút chuyển ưu tiên relay thành Primary/Secondary. 2. Hệ thống hiển thị thông báo xác nhận thay đổi trên relay của người dùng. 3. Người dùng chọn OK để xác nhận sự thay đổi muốn thực hiện.			
Kịch bản phụ:			
2.1 Người dùng chọn hủy để hủy thao tác thay đổi relay.			

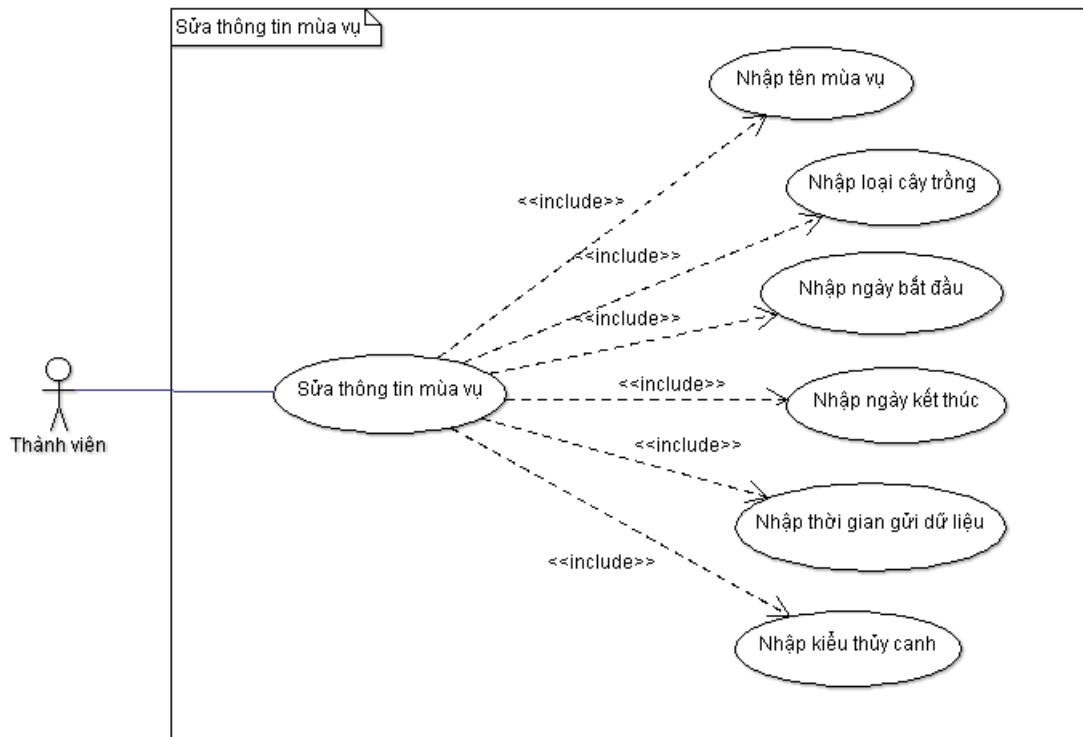
Bảng 16: Mô tả use case sửa relay.



Hình 67: Use case thêm mới mùa vụ.

ID	Name	Created by	Created at
UC_CROP_001	Tạo mới mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng tạo thêm mùa vụ mới cho thiết bị.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin của thiết bị.			
Kịch bản chính:			
1. Người dùng khởi động khung thêm thiết bị mới bằng cách nhấn vào nút thêm thiết bị mới trong trang thông tin thiết bị của mình.			
2. Người dùng nhập thông tin của mùa vụ mới vào và nhấn OK.			
3. Hệ thống thực hiện lưu thông tin mùa vụ mới vào cơ sở dữ liệu.			
4. Hệ thống thông báo người dùng đã thực hiện thêm mùa vụ mới vào thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập tên mùa vụ. / - Include: nhập loại cây trồng. / - Include: nhập ngày bắt đầu. / - Include: nhập ngày kết thúc. / - Include: nhập thời gian gửi dữ liệu. / - Include: nhập kiểu thủy canh.			

Bảng 17: Mô tả use case tạo mới mùa vụ.



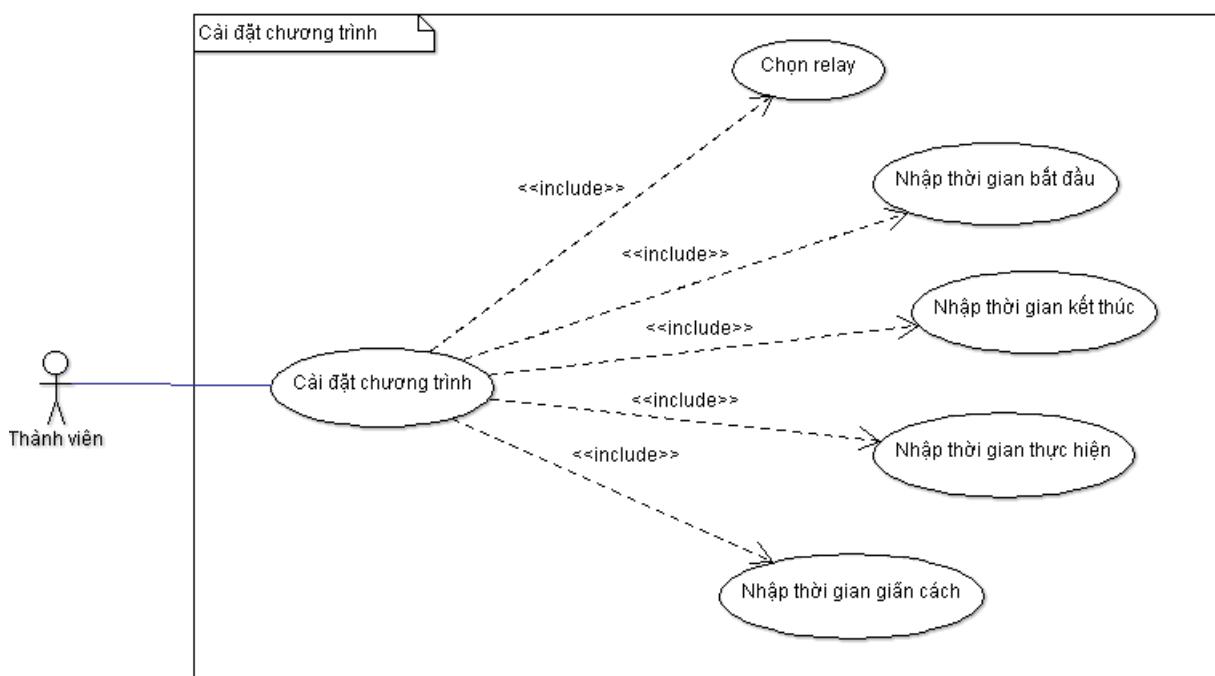
Hình 68: Use case sửa thông tin mùa vụ.

ID	Name	Created by	Created at
UC_CROP_003	Sửa thông tin mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng sửa thông tin mùa vụ hiện có.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin mùa vụ.			
Kịch bản chính:			
1. Người dùng khởi động khung chỉnh sửa thông tin mùa vụ bằng cách nhấp vào nút chỉnh sửa.			
2. Người dùng nhập thông tin cập nhật mới của mùa vụ và nhấp nút cập nhật.			
3. Hệ thống cập nhật thông tin mùa vụ mới đã được người dùng nhập.			
4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			
Các quan hệ khác:			
- Include: nhập tên mùa vụ. / - Include: nhập loại cây trồng. / - Include: nhập ngày bắt đầu. / - Include: nhập ngày kết thúc. / - Include: nhập thời gian gửi dữ liệu. / - Include: nhập kiểu thủy canh			

Bảng 18: Mô tả use case sửa thông tin mùa vụ.

ID	Name	Created by	Created at
UC_CROP_002	Xóa mùa vụ	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng xóa mùa vụ của mình.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin của thiết bị.			
Kịch bản chính:			
1. Người dùng nhấn nút xóa một mùa vụ trong danh sách các mùa vụ đã có. 2. Hệ thống hiển thị thông báo xác nhận xóa mùa vụ của người dùng. 3. Người dùng chọn OK để xóa mùa vụ. 4. Hệ thống thông báo xóa thành công. 5. Hệ thống thông báo xóa thành công.			
Kịch bản phụ:			
3.1 Người dùng chọn hủy để hủy xóa mùa vụ.			

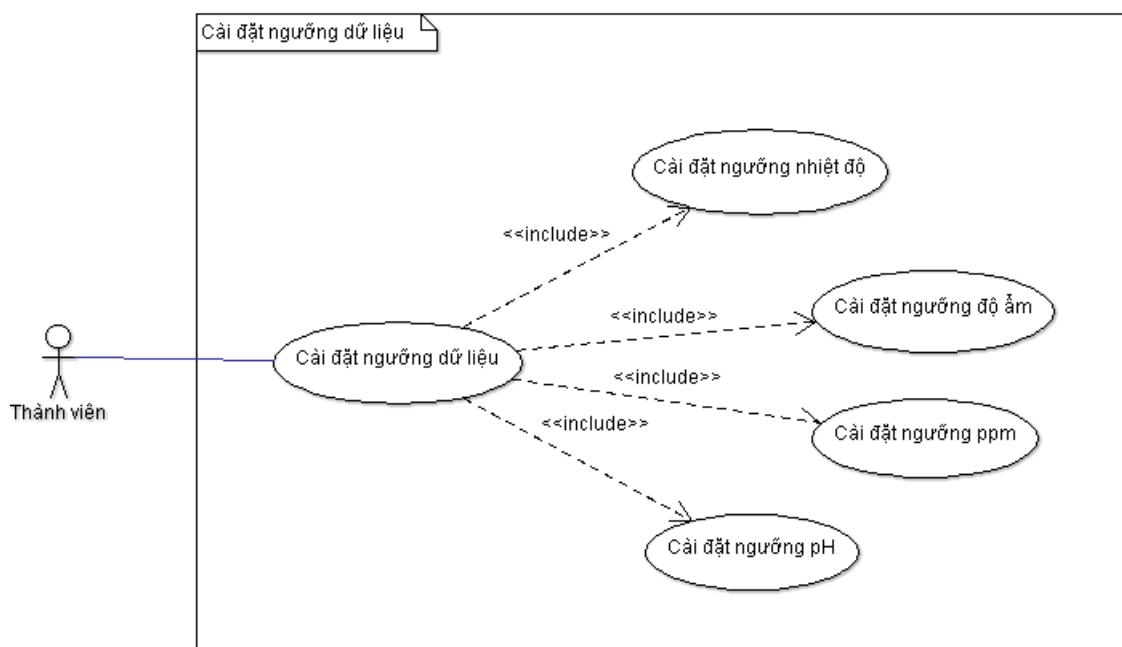
Bảng 19: Mô tả use case xóa mùa vụ.



Hình 69: Use case cài đặt chương trình mùa vụ.

ID	Name	Created by	Created at
UC_CROP_004	Cài đặt chương trình	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng cài đặt lịch trình cho mùa vụ.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin mùa vụ.			
Kịch bản chính:			
1. Người dùng mở khung cài đặt bằng cách nhấn vào nút Edit bên lịch trình. 2. Người dùng nhập các dữ liệu cần thiết. 3. Hệ thống tạo một lịch trình mới, lưu vào database và gửi xuống thiết bị. 4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			

Bảng 20: Mô tả use case cài đặt chương trình cho mùa vụ.



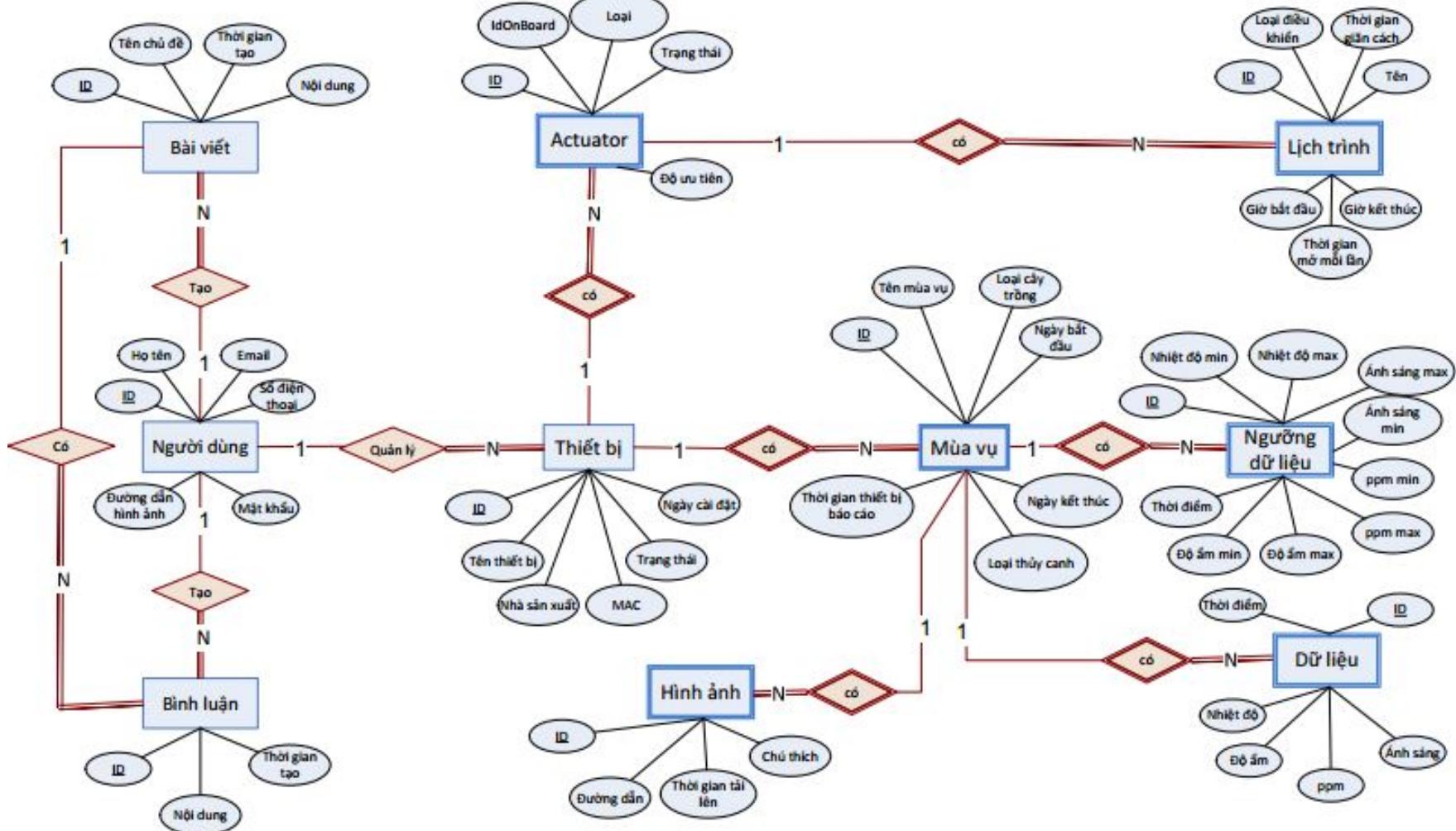
Hình 70: Use case cài đặt ngưỡng dữ liệu.

ID	Name	Created by	Created at
UC_CROP_005	Cài đặt ngưỡng dữ liệu	Tạ Chí Tây	20/3/2017
Actor: Thành viên			
Mô tả ngắn gọn: người dùng cài đặt ngưỡng cảnh báo dữ liệu.			
Tiền điều kiện: người dùng đã đăng nhập, vào xem thông tin mùa vụ.			
Kịch bản chính:			
1. Người dùng nhấn vào nút Edit bên ô hiển thị ngưỡng để mở ô cài đặt.			
2. Người dùng nhập các dữ liệu cần thiết.			
3. Hệ thống tạo một ngưỡng mới.			
4. Hệ thống thông báo người dùng đã thực hiện thành công.			
Kịch bản phụ:			
2.1 Người dùng nhập thông tin sai hệ thống sẽ báo lỗi.			

Bảng 21: Mô tả use case cài đặt ngưỡng dữ liệu.

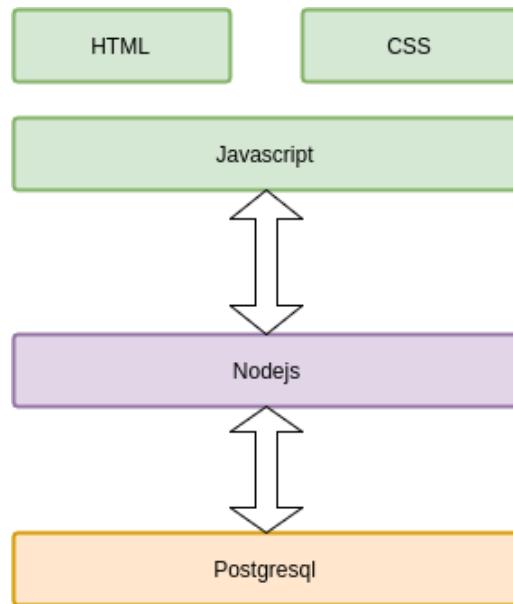
3.4.2 Thiết kế database

71



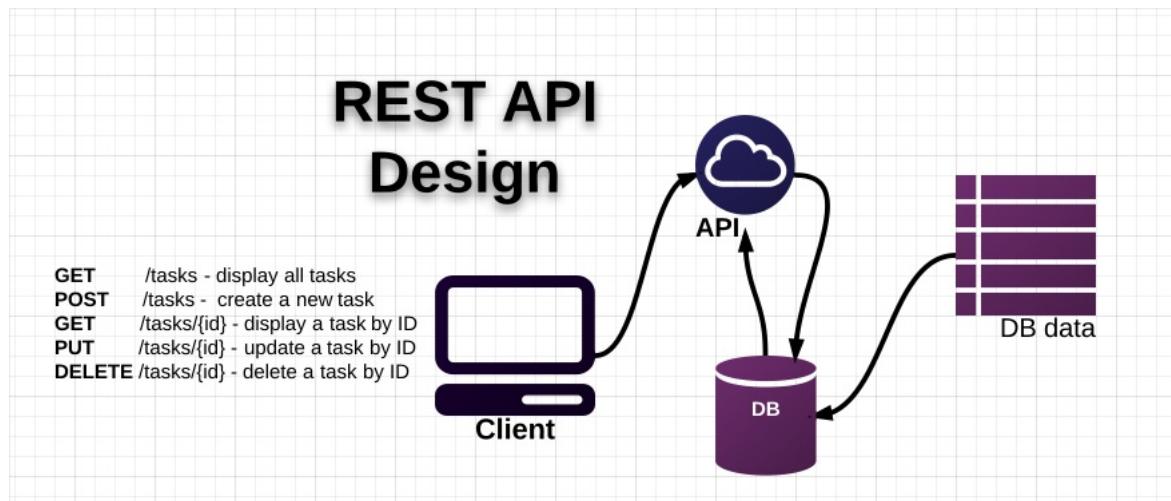
Hình 71: Thiết kế database

3.4.3 Thiết kế API



Hình 72: Mô hình cụm server

API được thiết kế theo chuẩn RESTful



Hình 73: Mô hình RESTful API

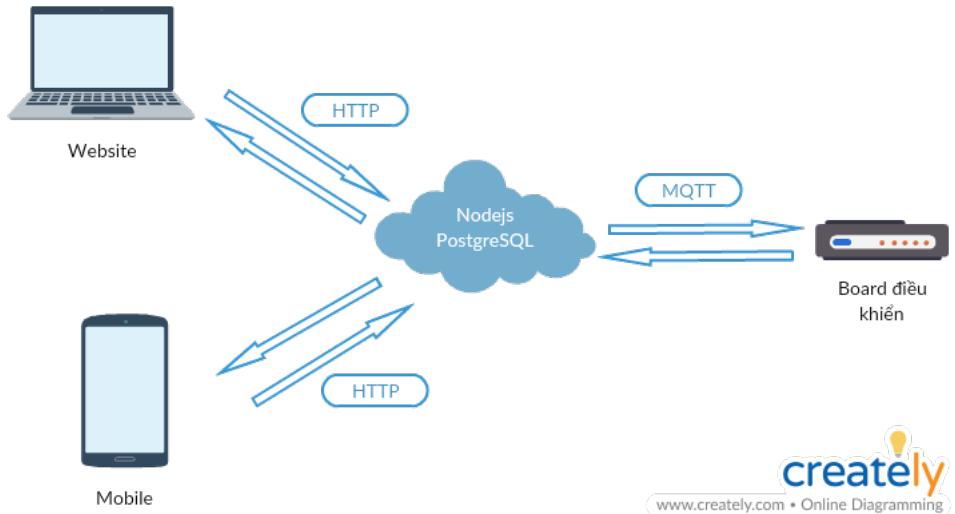
3.5 Ứng dụng di động

Ứng dụng được viết trên framework Ionic 2, sử dụng angular 2 và có thể chạy trên android hoặc iOS (hiện tại nhóm mới thử nghiệm trên android)

- Ứng dụng nhằm mục đích giúp người dùng có thể theo dõi trạng thái của hệ thống một cách nhanh chóng.
- Chức năng chính:
 - Thêm một board mạch mới bằng cách quét mã QR.
 - Xem thông tin tình trạng hệ thống hiện tại (nhiệt độ, độ ẩm, cường độ ánh sáng, nồng độ dinh dưỡng).
 - Xem số lượng và trạng thái các relay trên board.
 - Có thể tắt, mở các relay trên board.
- Cách thức hoạt động: truy vấn API của web service để lấy và gửi thông tin.

3.6 Giao tiếp giữa Board điều khiển và Web

3.6.1 Giao thức



Hình 74: Mô hình các giao thức trong hệ thống

Sử dụng giao thức MQTT cho cả hai chiều tương tác là gửi dữ liệu từ thiết bị lên và từ server gửi xuống.

3.6.2 Cấu trúc các gói tin

Các bảng sau mô tả cụ thể cấu trúc các gói tin:

Cấu trúc gói tin					
Số thứ tự	SN	MAC	CMD_ID	DATA_LENGTH	DATA
Trường	SN	MAC	CMD_ID	DATA_LENGTH	DATA
Kích thước (byte)	4	12	2	4	>0
Mô tả	số thứ tự gói tin gửi và nhận	địa chỉ MAC	định danh gói tin	độ dài gói tin	dữ liệu

Bảng 22: Cấu trúc chung của các gói tin

Chi tiết các gói tin

Gửi từ	Server		
CMD_ID	01		
DATA_LENGTH	0034		
Trường dữ liệu	thời gian bắt đầu	thời gian kết thúc	thời gian gửi dữ liệu
Kích thước (byte)	14	14	6
Cấu trúc	yyyymmddhhmmss	yyyymmddhhmmss	hhmmss
Mô tả, ví dụ	yyyy: năm mm: tháng dd: ngày hh: giờ mm: phút ss: giây		

Bảng 23: Cấu hình

Gửi từ	Server			
CMD_ID	02			
DATA_LENGTH	$(2+2+6x4xSCH_NUM)xIO_NUM$			
Trường dữ liệu	IO_NUM	IO_ID	SCH_NUM	lịch trình
Kích thước (byte)	2	2	2	$6x4xSCH_NUM$
Cấu trúc	XX	XX	XX	$4x[hhmmss]xSCH_NUM$
Mô tả, ví dụ	số lượng actuator trên board	ID actuator	số lịch trình	hhmmss: 050000

Bảng 24: Lịch trình cho actuator

Gửi từ	Server	
CMD_ID	03	
DATA_LENGTH	0003	
Trường dữ liệu	IO_ID	lệnh điều khiển
Kích thước (byte)	2	1
Cấu trúc	XX	X
Mô tả, ví dụ		giá trị: 0/1 0: tắt 1: bật

Bảng 25: Điều khiển actuator

Gửi từ	Board điều khiển			
CMD_ID	04			
DATA_LENGTH	0012			
Trường dữ liệu	nhiệt độ	độ ẩm	ánh sáng	ppm
Kích thước (byte)	2	2	4	4
Cấu trúc	XX	XX	XXXX	XXXX

Bảng 26: Dữ liệu gửi từ các cảm biến

Gửi từ	Board điều khiển
CMD_ID	05
DATA_LENGTH	0014
Trường dữ liệu	REAL_TIME
Kích thước (byte)	14
Cấu trúc	yyyymmddhhmmss

Bảng 27: Đồng bộ thời gian thực

Gửi từ	Server		
CMD_ID	06		
DATA_LENGTH	0004		
Trường dữ liệu	IO_ID	lệnh điều khiển	mức ưu tiên
Kích thước (byte)	2	1	1
Cấu trúc	XX	X	X
Mô tả, ví dụ	11 - 19: bơm nước 21 - 29: đèn 31 - 39: quạt 41 - 49: bơm oxy	0: thêm actuator 1: xóa actuator	0: primary 1: secondary

Bảng 28: Thêm bớt actuator

Gửi từ	Board điều khiển
CMD_ID	07
DATA_LENGTH	0001
Trường dữ liệu	HANDLED
Kích thước (byte)	1
Cấu trúc	X
Mô tả, ví dụ	0: gói tin sai, không nhận được 1: xử lý thành công gói tin

Bảng 29: Gói tin phản hồi

Gửi từ	Server
CMD_ID	08
DATA_LENGTH	0001
Trường dữ liệu	ACTIVE
Kích thước (byte)	1
Cấu trúc	X
Mô tả, ví dụ	0: thêm board mới 1: xóa board

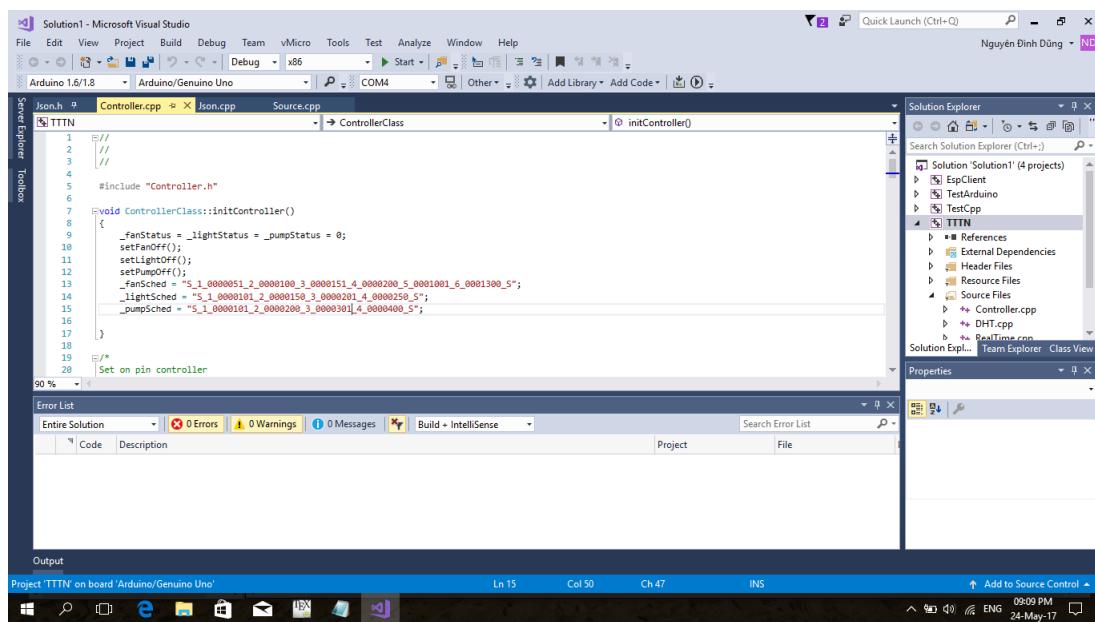
Bảng 30: Thêm, bớt một board điều khiển

4 HIỆN THỰC HỆ THỐNG

4.1 Hệ thống Board điều khiển

4.1.1 Cài đặt phần mềm

- Môi trường lập trình ứng dụng (IDE) ở đây sử dụng Visual Studio 2017 download từ trang chủ Microsoft.



Hình 75: Visual Studio 2017

Sau khi cài đặt Visual Studio, tải plugin Arduino từ <https://marketplace.visualstudio.com/>

- Cài đặt Sublime Text để lập trình giao diện web.

Link download <https://www.sublimetext.com/>

```

D:\Study\Arduino\EspClient\EspClient.ino (Arduino) - Sublime Text 2 (UNREGISTERED)
File Edit Selection Find View Goto Tools Project Preferences Help
FOLDERS
  ▾ Arduino
    ▾ Debug
    ▾ Device
    ▾ EspClient
      ▾ _vm
      ▾ Debug
        EspClient.ino
      EspClient.vcxproj
      EspClient.vcxproj.filters
      WebClient.cpp
      WebClient.h
    ▾ libraries
    ▾ TestArduino
    ▾ TestCpp
    ▾ TestEverything
    ▾ TTN
  ...
EspClient.ino
1  /*
2   * Hello world web server
3   * circuits4you.com
4   *
5   * #include <ESP8266WiFi.h>
6   * #include <WiFiClient.h>
7   * #include <ESP8266WebServer.h>
8   * #include <PubSubClient.h>
9   */
10 // esp mode
11 #define AP 1
12 #define STA 2
13 #define HTTP_PORT 80
14 #define POST_DATA String("POST_DATA_")
15 #define GET_DATA String("GET_DATA_")
16 #define REST_DATA String("RESPONSE_DATA_")
17 #define SCHEDULE_DATA String("SCHEDULE_DATA_")
18 #define SENSOR_DATA String("SENSOR_DATA_")
19 ...
20 #define INTERVAL_TIME_CHECK_UNO 2000000
21 ...
22 ...
23 ...
24 // credentials, AP mode
25 String apId = "ESP8266";
26 String apPass = "12345678";
27 String apSSID = "ESP8266";
28 String staID = "real_man";
29 String staPass = "00000000";
30 String mqtt_server = "broker.mqtt-dashboard.com";
31 String mqtt_port;
32 String sensorDataFromUno;
33 String scheduleDataFromServer;
34 ...
35 long unsigned int lastCheckDataFromUno;
36 WiFiClient espClient;
37 PubSubClient client(espClient);
38 ...
39 ...
40 int curMode = AP;
41 int nextMode = AP;
42 ...
43 ...
44 ...
45 ESP8266WebServer server(80); //server on port 80
46 ...
47 //----- This routine is executed when you open its IP in browser
48 ...

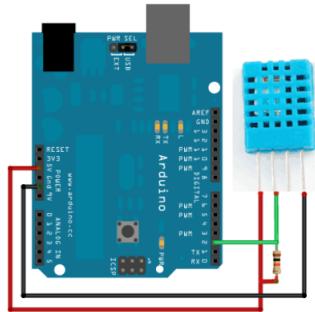
```

Line 19, Column 1 Tab Size: 4 C++ 09:19 PM 24-May-17

Hình 76: Sublime Text 2

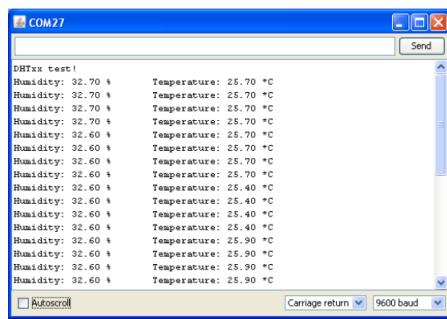
4.1.2 Sử dụng cảm biến

- Nhiệt độ, độ ẩm



Hình 77: Sublime Text 2

- Thư viện giao tiếp DHT có sẵn trên github.
- Như đã trình bày trong phần thiết kế, dòng cấp cho DHT 11 là 3-5V.
- Dữ liệu đầu ra là digital
- Thư viện tải về có kèm theo ví dụ, dưới đây là kết quả:



Hình 78: Kết quả đọc DHT

4.1.3 Thời gian thực(RTC-DS1307)

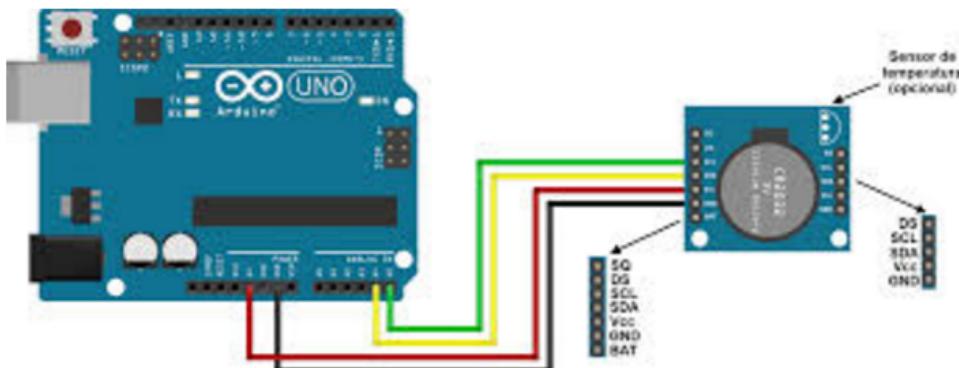
- Nhắc lại về I2C: I2C là viết tắt của "Inter-Integrated Circuit", một chuẩn giao tiếp được phát minh bởi Philips' semiconductor division (giờ là NXP) nhằm đơn giản hóa việc trao đổi dữ liệu giữa các ICs. Đôi khi nó cũng được gọi là Two Wire Interface (TWI) vì chỉ sử dụng 2 kêt nối để truyền tải dữ liệu, 2 kêt nối của giao tiếp I2C gồm: SDA (Serial Data Line) và SCL (Serial Clock Line).
- Trên board Arduino UNO, SDA là chân analog 4, SCL là chân analog 5.



Hình 79: Pin I2C trên board Arduino Uno.

- Lưu ý: Vì Arduino sử dụng điện áp 5V nên các thiết bị khác cũng phải sử dụng mức này.
- Sử dụng thư viện I2C-Wire.h.
- Chúng ta sẽ lắp module Real time clock với Arduino như sau:

- 1 `Wire.begin(address (optional));`
 - Khởi tạo thư viện Wire.h và tham gia vào I2C bus.
 - address: 7-bit địa chỉ của thiết bị "Slave" (optional); nếu không có địa chỉ thì coi như "Master".
- 2 `Wire.beginTransmission(address);`
 - Bắt đầu truyền dữ liệu đến thiết bị "Slave" với address đã có.
- 3 `Wire.endTransmission();`
 - Kết thúc truyền dữ liệu đến thiết bị "Slave" đã được bắt đầu bởi Wire.beginTransmission(address).
- 4 `Wire.write(value);`
 - Ghi dữ liệu lên thiết bị "Slave", được gọi giữa beginTransmission() và endTransmission().
- 5 `Wire.read();`
 - Đọc dữ liệu được truyền từ thiết bị "Slave" đến Arduino, được gọi sau requestFrom().
- 6 `Wire.requestFrom(address, quantity);`
 - Được sử dụng bởi thiết bị "Master" để yêu cầu dữ liệu từ thiết bị "Slave".
 - address: là địa chỉ của thiết bị "Slave".
 - quantity: số lượng bytes yêu cầu.

Hình 80: Một số hàm trong thư viện **Wire.h**.

Hình 81: Sơ đồ lắp đặt RTC với board Uno

- Đây là các Timekeeper registers của DS1307, chúng ta sẽ dựa vào bảng này để read/write IC DS1307 qua I2C:

ADDRESS	BIT 7	BIT 6	BIT 5	BIT 4	BIT 3	BIT 2	BIT 1	BIT 0	FUNCTION	RANGE
00h	CH	10 Seconds			Seconds			Seconds	00-59	
01h	0	10 Minutes			Minutes			Minutes	00-59	
02h	0	12	10 Hour	10 Hour	Hours			Hours	1-12 +AM/PM 00-23	
		24	PM/ AM							
03h	0	0	0	0	0	DAY			Day	01-07
04h	0	0	10 Date		Date			Date	01-31	
05h	0	0	0	10 Month	Month			Month	01-12	
06h	10 Year				Year			Year	00-99	
07h	OUT	0	0	SQWE	0	0	RS1	RS0	Control	—
08h-3Fh									RAM 56 x 8	00h-FFh

Hình 82: DS1307 Timekeeper

4.1.4 Sử dụng Relay

- Thông số của một relay

Một module rơ-le được tạo nên bởi 2 linh kiện thu động cơ bản là rơ-le và transistor, nên module rơ-le có những thông số của chúng:

- * Hiệu điện thế kích tối ưu
- * Các mức hiệu điện thế tối đa và cường độ dòng điện tối đa của đồ dùng điện khi nối vào module rơ-le



Hình 83: Thông số trên relay.

- 10A - 250VAC: Cường độ dòng điện tối đa qua các tiếp điểm của rơ-le với hiệu điện thế $\leq 250V$ (AC) là 10A.
 - 10A - 30VDC: Cường độ dòng điện tối đa qua các tiếp điểm của rơ-le với hiệu điện thế $\leq 30V$ (DC) là 10A.
 - 10A - 30VDC: Cường độ dòng điện tối đa qua các tiếp điểm của rơ-le với hiệu điện thế $\leq 30V$ (DC) là 10A.
 - 10A - 28VDC: Cường độ dòng điện tối đa qua các tiếp điểm của rơ-le với hiệu điện thế $\leq 28V$ (DC) là 10A.
 - SRD-05VDC-SL-C: Hiện điện thế kích tối ưu là 5V.
- Cách sử dụng Relay

- * Relay bình thường gồm có 6 chân. Trong đó có 3 chân để kích, 3 chân còn lại nối với đồ dùng điện công suất cao.
- * 3 chân dùng để kích
 - +: cấp hiệu điện thế kích tối ưu vào chân này.
 - -: nối với cực âm

- S: chân tín hiệu, tùy vào loại module rơ-le mà nó sẽ làm nhiệm vụ kích rơ-le

* 3 chân còn lại nối với đồ dùng điện công suất cao

- COM: chân nối với 1 chân bất kỳ của đồ dùng điện, nhưng mình khuyên bạn nên mắc vào dây chân lửa (nóng) nếu dùng hiệu điện thế xoay chiều và cực dương nếu là hiệu điện một chiều.
- ON hoặc NO: chân này bạn sẽ nối với chân lửa (nóng) nếu dùng điện xoay chiều và cực dương của nguồn nếu dòng điện một chiều.
- OFF hoặc NC: chân này bạn sẽ nối chân lạnh (trung hòa) nếu dùng điện xoay chiều và cực âm của nguồn nếu dòng điện một chiều.

4.1.5 Sử dụng module wifi ESP8266

- Cách nạp firmware cho Esp8266V1.

Để nạp firmware cho Esp8266, sử dụng module PL2102-USB2UART



Hình 84: Module CP2102-USB2UART.

Nối chân(pin) giữa Esp8266 với CP2102:

ESP8266	PL2303	Nguồn 3.3V	Ghi chú
VCC	x	VCC	Nối VCC của ESP với nguồn 3V3
GND	GND	GND	Nối chung GND của ESP, PL2303 và GND của nguồn 3V3
RST	x	x	Khi ESP bị treo hoặc không nạp được thì nối với GND sau đó ngắt kết nối
CH_PD	x	VCC	Nối với 3V3 cho phép kích hoạt phát wifi
TXD	RXD	x	Nối ESP với PL2303 để giao tiếp nạp firmware
RXD	TXD	x	Nối ESP với PL2303 để giao tiếp nạp firmware
GPIO0	GND	GND	Nối xuống GND để có thể update được firmware
GPIO2	x	x	

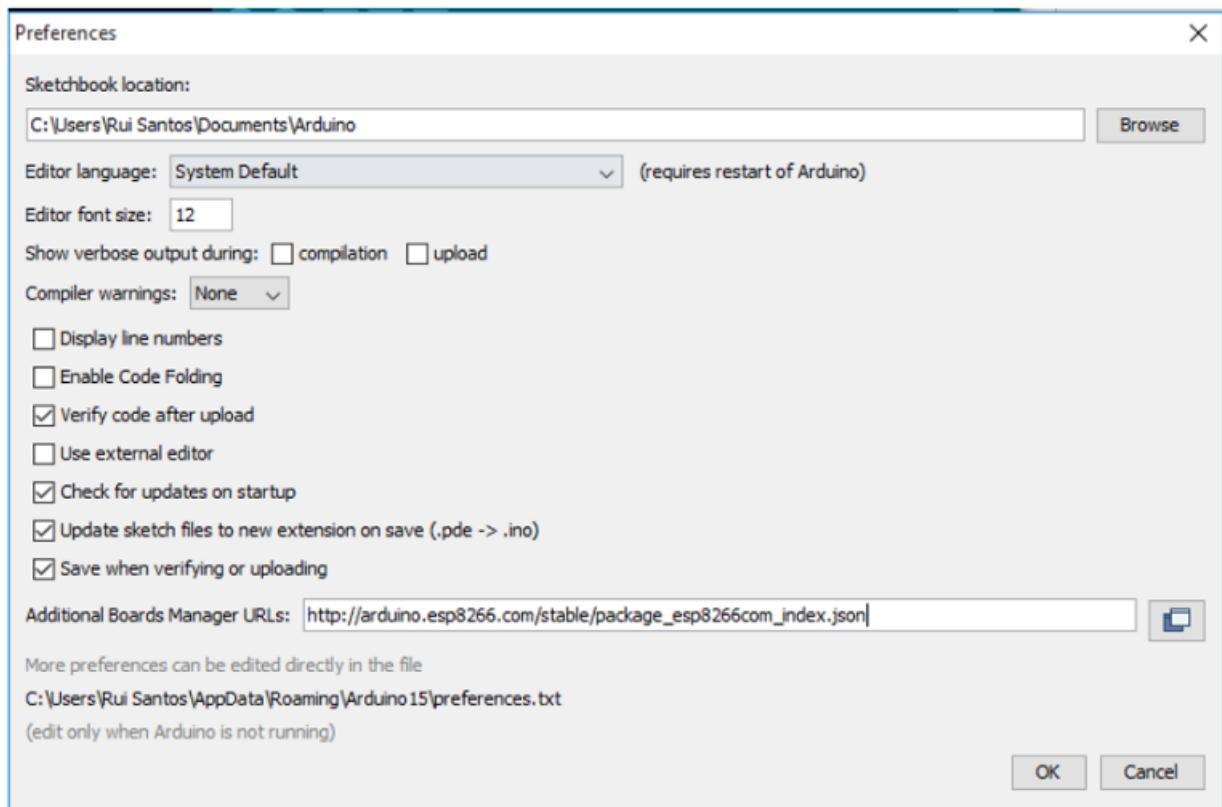
Hình 85: Bảng nối chân Esp với CP2102.

Lưu ý: ESP8266V1 hoạt động ở mức điện áp 3.3V

Sau khi nối chân ta tiến hành nạp code tương tự với board Arduino Uno.

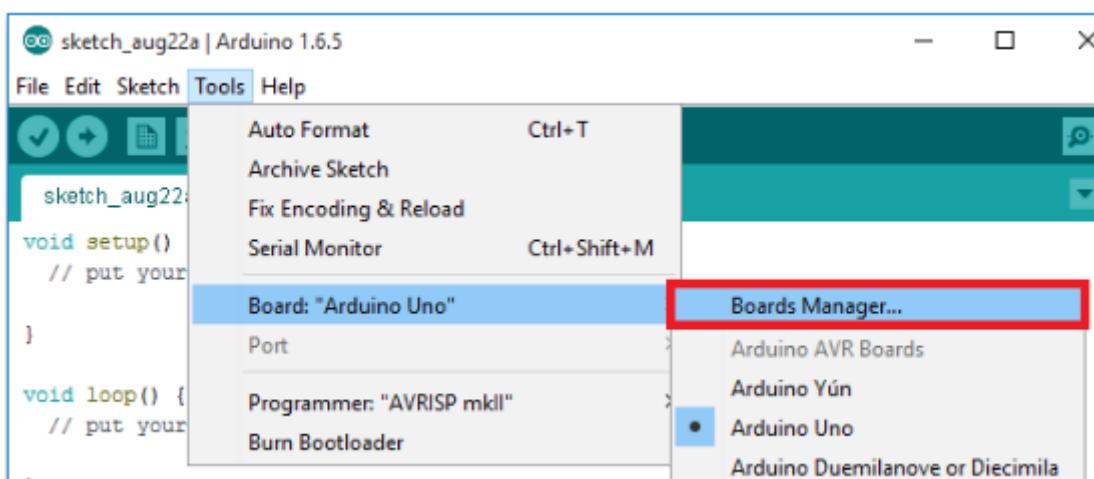
- Lập trình Esp8266

Để lập trình cho Esp8266, trước tiên phải cài gói thư viện dành cho board này Đây là thư viện mã nguồn mở có sẵn trên github.



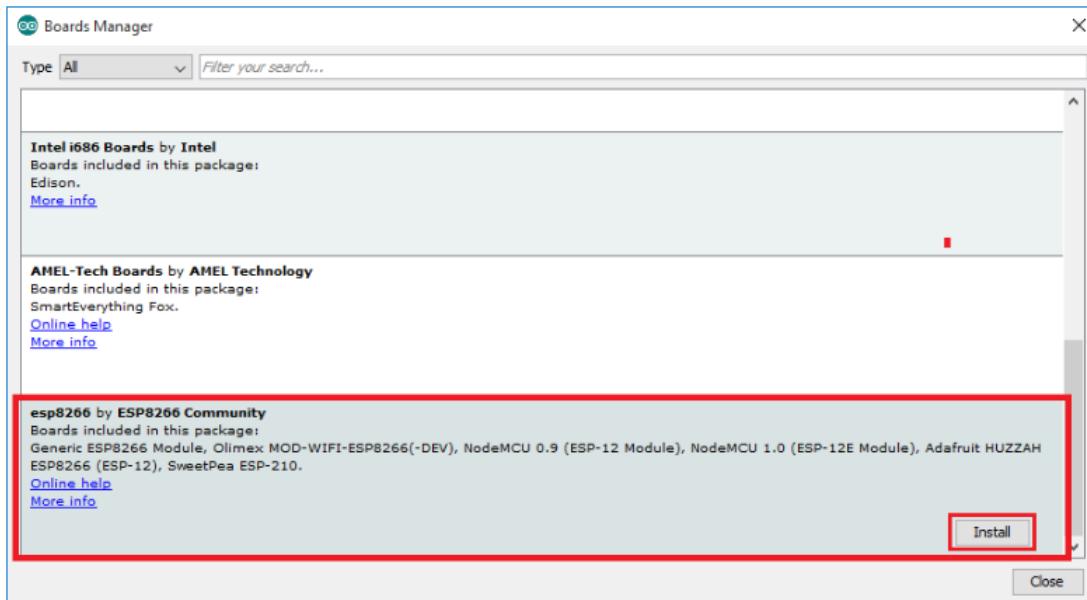
Hình 86: Step 1.

Tiếp theo vào Tool/Board Boards/Manager.



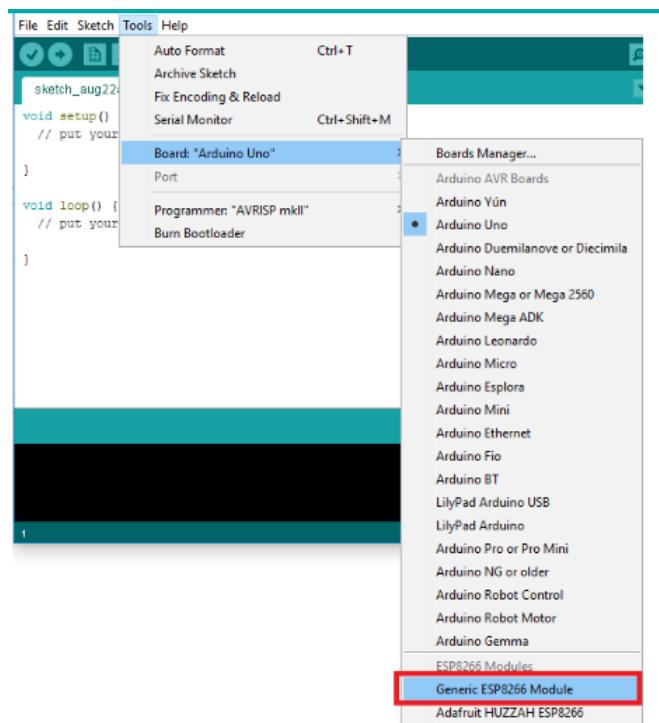
Hình 87: Step 2.

Đợi một lát để chương trình tìm kiếm. Ta kéo xuống và click vào ESP8266 by ESP8266 Community, click vào Install. Chờ phần mềm tự động download và cài đặt.



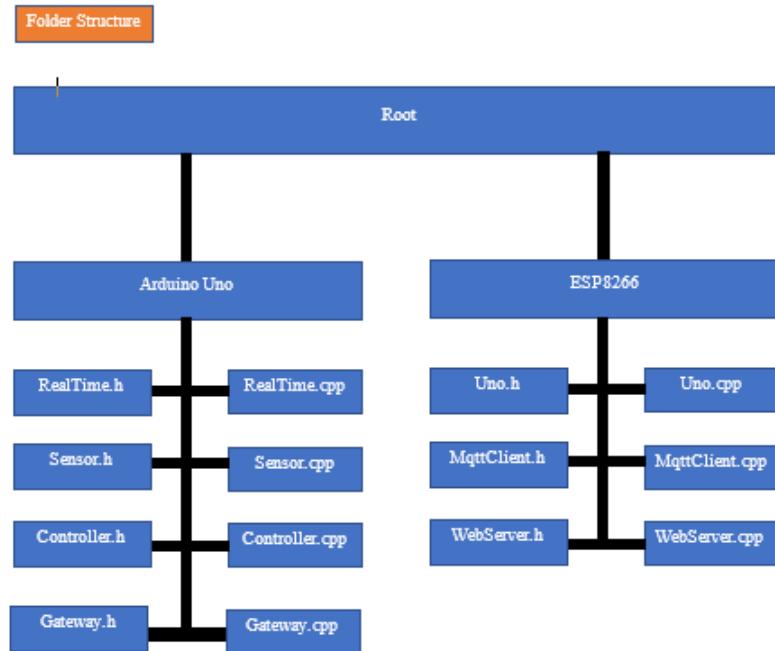
Hình 88: Step 3.

Kết nối module USB-to-UART vào máy tính. Vào Tool/Board/Generic ESP8266 Module, chọn cổng COM tương ứng với module USB-to-UART tương ứng.



Hình 89: Step 4.

4.1.6 Cấu trúc thư mục mã nguồn



Hình 90: Cấu trúc thư mục mã nguồn.

Mã nguồn được chia làm 2 thành phần chính:

- Thư mục Arduino Uno: chứa mã nguồn thực thi chương trình nạp lên board Arduino Uno
- Thư mục ESP8266: chứa mã nguồn thực thi chương trình nạp lên board ESP8266.
- Trong mỗi thư mục file có đuôi **ino** là file thực thi chính, còn lại là các lớp(class).

4.1.7 Hoạt động của Esp8266

Welcome to ESP Config page

ID:

PASSWORD:

Submit

Hình 91: Giao diện web của esp8266.

Khi triển khai thiết bị, để thiết bị wifi Esp8266 cần được cấu hình để có thể tham gia vào mạng internet.

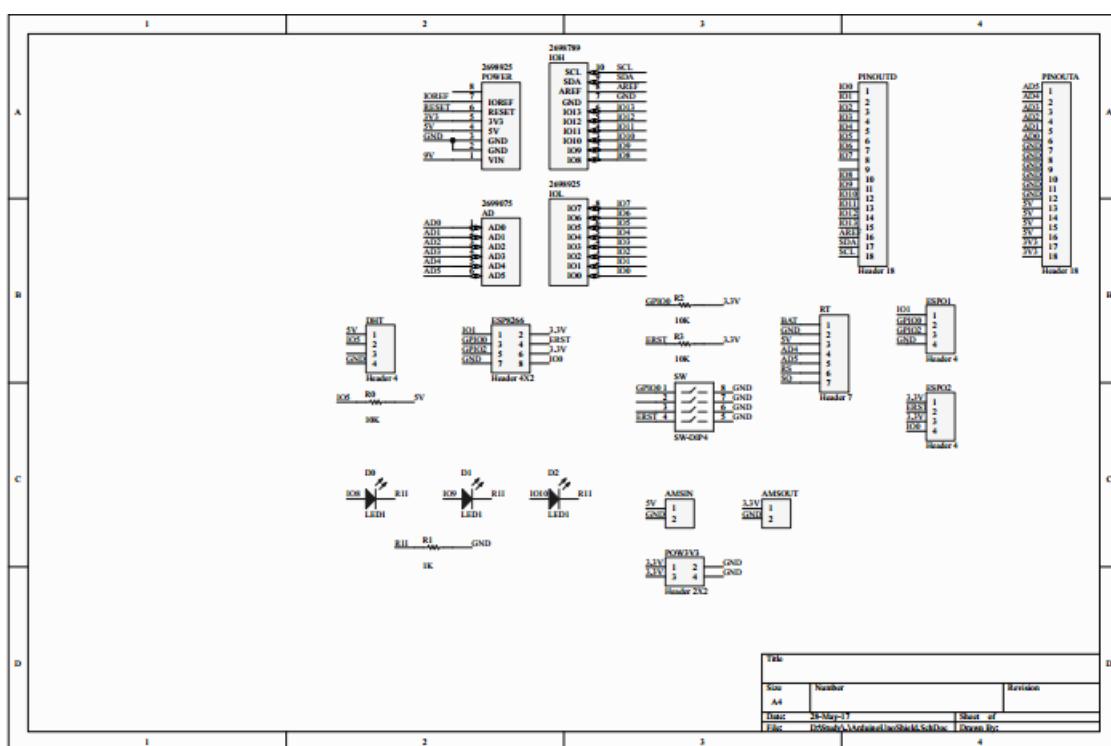
Để thực hiện chức năng này module Esp8266 cần hoạt động ở hai chế độ

- Mqtt Client-ở chế độ này nó thực hiện chức năng:
 - Hoạt động ở chế độ STA (station): là một wifi client truy cập vào điểm phát wifi gần đó.
 - Là một **MQTT** client trao đổi thông tin với Web server (cũng là một MQTT client)
 - Kết nối với mạch điều khiển Arduino Uno thông qua UART để trao đổi một số thông tin: dữ liệu các sensor, dữ liệu về lịch trình hoạt động các bộ điều khiển, thời gian thực.
- Web server-thực hiện các chức năng:

- Hoạt động ở chế độ AP(Access Point), nó sẽ phát wifi, laptop có thể truy cập wifi của nó.
- Là một Web server đơn giản cho mục đích cấu hình wifi để Esp8266 có thể tham gia vào internet.
- Chế độ tự chuyển chế độ giữa STA và AP.

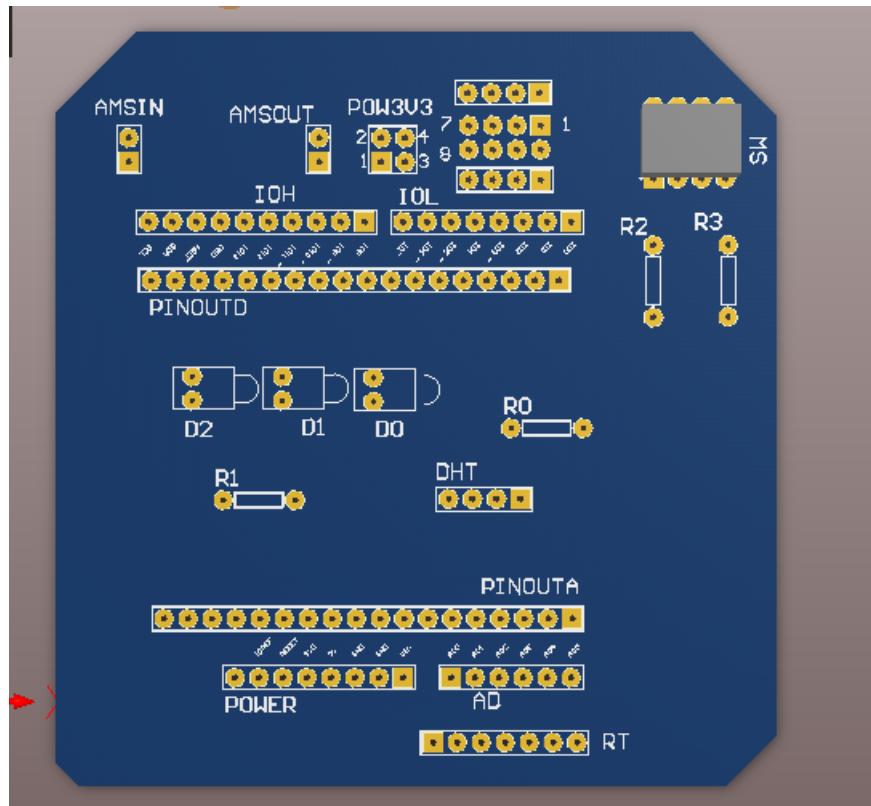
4.1.8 Thiết kế mạch điều khiển

- Sơ đồ nguyên lý (Schematic)



Hình 92: Sơ đồ nguyên lý mạch điều khiển.

- Sơ đồ mạch in 3D (PCB)



Hình 93: Sơ đồ mạch in.

4.1.9 Mạch chức năng đo ppm

Lý thuyết

- Độ dẫn điện (EC):

Chỉ số EC (electro-conductivity) (tương đối) là chỉ số điện tử tổng nồng độ ion hòa tan trong dung dịch. Độ dẫn điện có thể được thể hiện bằng một số đơn vị khác nhau, nhưng đơn vị tiêu biểu được dùng để đo lường EC là millisiemens trên centimet (mS / cm). Chỉ số EC không diễn tả nồng độ của từng chất trong dung dịch đồng thời cũng không thể hiện mức độ cân bằng của các chất dinh dưỡng trong dung dịch. EC là thước đo độ dẫn điện từ hai đầu dò 1cm.

Trong suốt quá trình tăng trưởng, cây hấp thu khoáng chất mà chúng cần, do vậy duy trì EC ở một mức ổn định là rất quan trọng. Nếu dung dịch có chỉ số EC cao thì sự hấp

thu nước của cây diễn ra nhanh hơn sự hấp thu khoáng chất. Điều này làm nồng độ dung dịch tăng cao và gây ngộ độc cho cây. Khi đó ta phải bổ sung thêm nước vào môi trường. Ngược lại, nếu EC thấp, cây sẽ hấp thu khoáng chất nhanh hơn hấp thu nước. Khi đó, nồng độ dung dịch giảm mạnh, cây sẽ không được cung cấp đầy đủ khoáng chất, chậm lớn và phát triển kém.

- Chỉ số TDS:

Chỉ số TDS (Total Dissolved Solids) là chỉ số đo tổng lượng chất rắn hòa tan, tổng số các ion mang điện tích bao gồm khoáng chất, muối hoặc kim loại tồn tại trong một khối lượng nước nhất định. TDS thường được biểu thị bằng hàm số ml/L hoặc ppm (Parts Per Million). 1 ppm tương ứng với 1mg chất rắn hòa tan trong một lít nước. Hầu hết nước máy sẽ có chỉ số PPM rơi vào khoảng từ 200 – 400ppm.

Chỉ số TDS cũng ảnh hưởng lớn đến sự phát triển của cây: Nếu TDS lên quá cao, nồng độ dung dịch vượt mức cho phép sẽ gây ra hiện tượng ngộ độc cho cây. Ngược lại, khi chỉ số TDS xuống thấp, dung dịch thủy canh sẽ không đảm bảo cung cấp đủ chất dinh dưỡng cho cây trồng.

- Tương quan giữa EC và TDS:

Mặc dù có một mối tương quan giữa EC và TDS nhưng chúng không giống nhau. TDS và EC là 2 tham số riêng biệt. TDS là tổng lượng chất rắn hòa tan trong nước. EC là khả năng của các chất co thể gây ra dòng điện. Lượng chất rắn như muối trong phân bón tỉ lệ trực tiếp với độ dẫn điện của nó, vì vậy lượng chất rắn cao gây độ dẫn cao. Vì khi phân bón hòa tan trong nước chúng trở thành các "ion", có mang điện tích âm hoặc dương, nên chúng sinh ra dòng điện.

Mối quan hệ của TDS và độ dẫn đặc hiệu của nước ngầm có thể được ước lượng bằng phương trình sau:

$$\text{TDS} = k_e * \text{EC}$$

Trong đó: TDS có đơn vị mg/L và EC là độ dẫn điện ở microsiemens trên mỗi centimet ở 25°C. Yếu tố tương quan k_e dao động từ 0,55 đến 0,8.

Giới hạn với cây trồng:

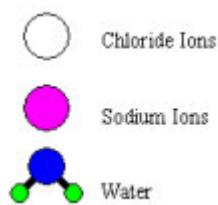
Cây trồng	EC (mS/cm)	TDS (ppm)
Cà chua	2.4 - 5.0	1400 - 2450
Địa lan	0.6 - 1.5	420 - 560
Hoa hồng	1.5 - 2.4	1050 - 1750
Xà lách	0.6 - 1.5	280 - 1260
Cây chuối	2.4 - 5.0	1260 - 1540
Dâu tây	1.5 - 2.4	1260 - 1540
Ớt	1.5 - 2.4	1260 - 1540

Hình 94: Giới hạn với cây trồng.

- Tại sao lại dùng thiết bị này?

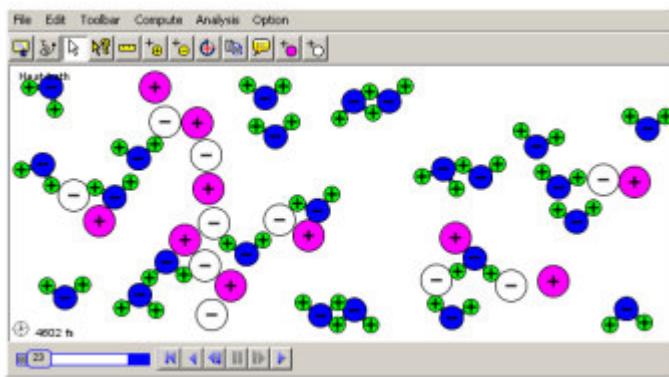
Nếu EC / PPM chỉ là đo điện dẫn (hoặc kháng) thì tại sao không sử dụng một đồng hồ volt / ohm trực tiếp? Bởi chúng đi qua dòng điện DC thông qua các đầu dò và bạn không thể đo độ dẫn của muối với dòng điện DC vì nó sẽ tách các phân tử ra ngoài, và vì các phân tử là điện dẫn điện, các phân tử điện sẽ thay đổi liên tục và sẽ không thu được kết quả gì. Bằng cách sử dụng một tín hiệu AC, với tần số đủ cao ($> 1\text{khz}$) các phân tử không có thời gian để di chuyển ra ngoài trước khi chúng được kéo theo hướng ngược lại.

Các thành phần ion:



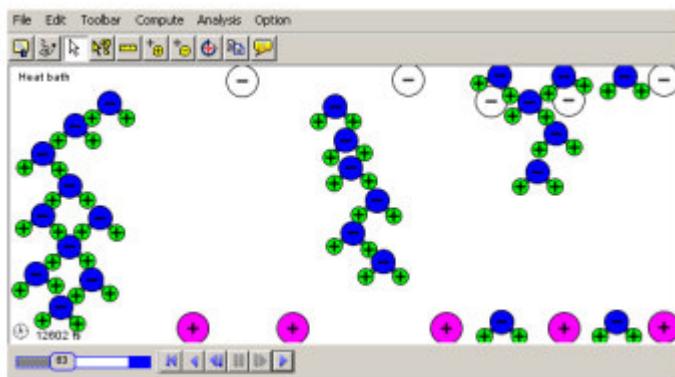
Hình 95: Các thành phần ion.

Trường hợp sử dụng dòng AC:



Hình 96: Trường hợp sử dụng dòng AC.

Trường hợp sử dụng dòng DC:



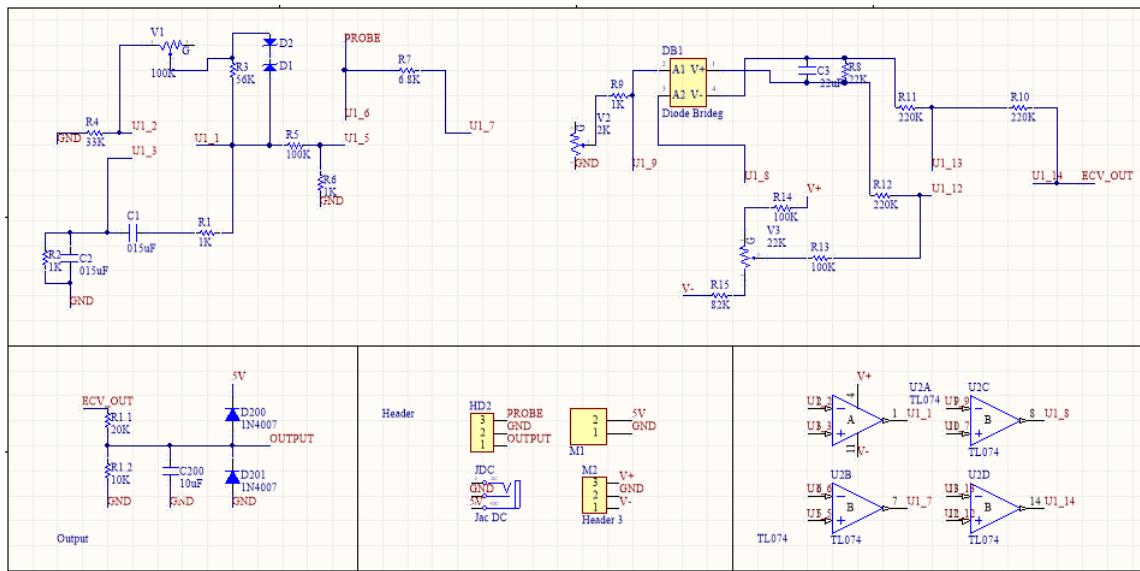
Hình 97: Trường hợp sử dụng dòng DC.

- **Hoạt động:**

Hai điện cực với một điện áp xoay chiều được đặt trong dung dịch. Điều này tạo ra một dòng điện phù thuộc vào bản chất dẫn điện của dung dịch. Thiết bị đọc dòng điện này và hiển thị theo đơn vị EC hoặc ppm.

Linh kiện, thiết kế

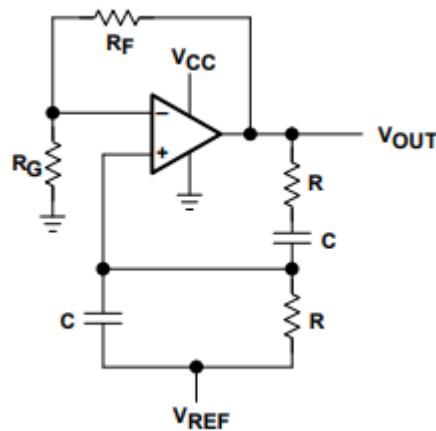
- **Schematic:**



Hình 98: Schematic.

Thành phần chính trong mạch:

- Oscillator cầu Wein: là mạch điện tử tạo ra tín hiệu điện tử dao động, thường là một sóng sin hay sóng vuông. Bộ tạo dao động chuyển đổi dòng điện một chiều (DC) từ nguồn điện sang tín hiệu dòng xoay chiều (AC). Ưu điểm của mạch là chỉ có một vài thành phần và sự ổn định tần số tốt. Phản nhược điểm của mạch là biên độ đầu ra biến dạng cao gây khó khăn trong việc thu thập. Có một vài cách để giảm thiểu tác động này.



Hình 99: Oscillator cầu Wein.

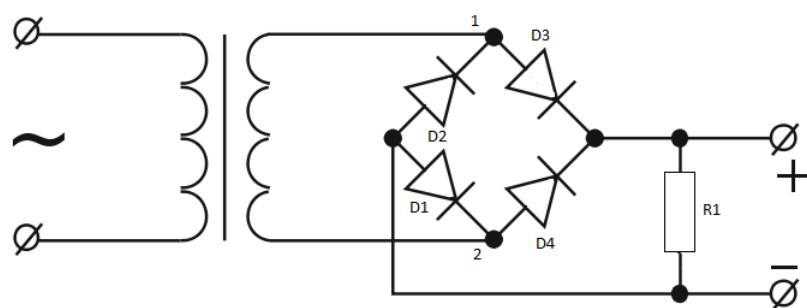
Các giá trị được cung cấp để tạo ra mạch dao động có tần số là 10Khz.

Ta có:

$$R = 1k \text{ Ohms}, C = 0.015\mu\text{F}, \text{suy ra } F = 1/(2\pi R C)$$

Chúng được sử dụng rộng rãi trong nhiều thiết bị điện tử. Ví dụ phổ biến của tín hiệu được tạo ra bởi dao động bao gồm các tín hiệu phát sóng của đài phát thanh và truyền hình, tín hiệu đồng hồ mà điều chỉnh máy tính và đồng hồ thạch anh, âm thanh được tạo ra bởi beepers điện tử và trò chơi điện tử... Ta sử dụng một tín hiệu AC để đo độ dãy của muối, có thể thay đổi giá trị R, C trong oscillator để đảm bảo rằng tần số đầu ra là trên 1kHz, tần số thấp hơn sẽ cung cấp các giá trị không ổn định.

- Gain loop: mạch khuếch đại nguồn tín hiệu nhận được từ đầu dò 1cm đặt trong dung dịch
- Bộ chuyển đổi AC thành DC: Một mạch chỉnh lưu, một mạch điện bao gồm các linh kiện điện - điện tử dùng để biến đổi dòng điện xoay chiều thành dòng điện một chiều. Mạch chỉnh lưu cả chu kỳ thường dùng 4 Diode mắc theo hình cầu (còn gọi là mạch chỉnh lưu cầu). Mạch chỉnh lưu có thể được sử dụng trong các bộ nguồn cung cấp dòng điện một chiều, hoặc trong các mạch tách sóng tín hiệu vô tuyến điện trong các thiết bị vô tuyến, sử dụng mạch chỉnh lưu nhiều diốt (4 di-ốt) để có thể biến đổi từ xoay chiều thành một chiều.

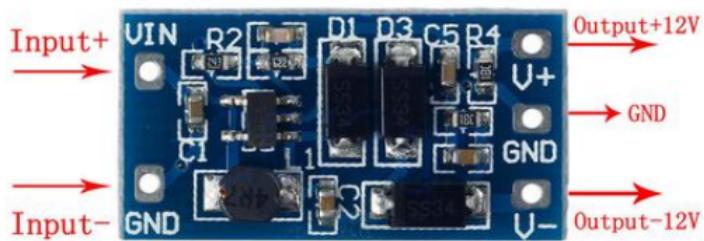


Hình 100: 4 di-ốt.

Nửa chu kì đầu, khi điểm 1 dương so với điểm 2, dòng chạy từ 1 qua D3 qua tải R1 qua D1 về đầu âm. Nửa chu kì sau, khi điểm 2 dương so với điểm 1, dòng chạy từ 2 qua D4 qua tải R1 qua D2 về đầu âm. Cả chu kì đều có dòng điện chạy qua tải.

- Linh kiện

- Nguồn đôi 12V: Mạch tạo nguồn đôi 12V có chức năng đổi từ nguồn đơn DC sang nguồn đôi + - 12V. Với điện áp ngõ vào từ 2.8V-5VDC.



Hình 101: Nguồn đôi 12V - Trước.



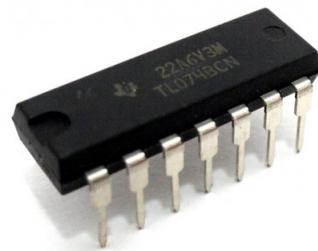
Hình 102: Nguồn đôi 12V - Sau.

- Jack DC: kết nối với nguồn DC để cấp nguồn cho mạch

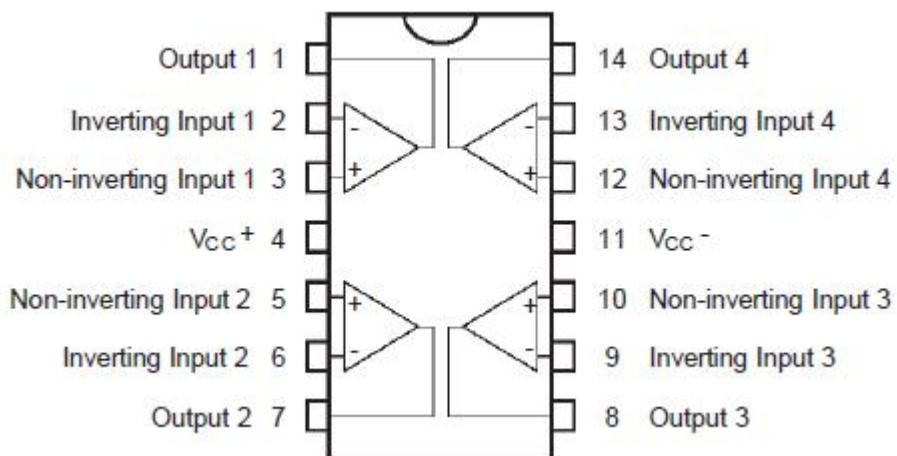


Hình 103: Jack DC.

- IC TL074: gồm 4 opamp trong 1 IC, dùng chung 1 nguồn đối.



Hình 104: IC TL074.



Hình 105: TL074 PIN.

- Diode Zener: diode ổn áp, nó được chế tạo sao cho khi phân cực ngược thì diode Zener sẽ ghim một mức điện áp gần cố định bằng giá trị ghi trên diode, làm ổn áp cho mạch điện.



Hình 106: Diode Zener.

- Tụ điện:

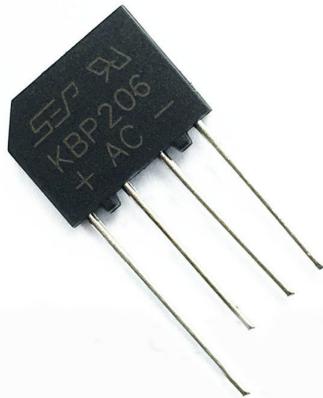


Hình 107: Tụ 0.015uF.



Hình 108: Tụ 0.22uF.

- Cầu Diode: gồm 4 diode đơn mắc thành cầu chỉnh lưu được đóng gói trong một vỏ duy nhất, dùng để chỉnh lưu dòng điện xoay chiều thành một chiều.
 - Điện áp tối đa: 600V.
 - Dòng điện định mức: 4A.
 - Nhiệt độ hoạt động: -55°C đến 150°C.



Hình 109: Cầu Diode.

4.2 Ứng dụng web

4.2.1 Hiện thực server API

Mẫu dữ liệu trả về cho tất cả các API:

- Không có data:

```
{  
    success: success,  
    message: message  
}
```

- Có data:

```
{  
    success: success,  
    data: data (*),  
    message: message  
}
```

Trong đó success nhận giá trị true nếu request được xử lý thành công và fail nếu thất bại.

API được thiết kế cho các resource:

- Người dùng (User)

- Lấy thông tin người dùng:
 - * Phương thức: GET
 - * URL: /user/info
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên: { user.id: 12 }
 - * Dữ liệu trả về (*): thông tin của người dùng (không chứa mật khẩu và token kích hoạt).
- Kiểm tra email mới có hợp lệ không:
 - * Phương thức: POST
 - * URL: /user/checkemail
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên: { email: "email" }
- Đăng ký người dùng:
 - * Phương thức: POST
 - * URL: /user/register
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:
 - { name: "name",
 - password: "123",
 - email: "email",
 - phone: "12345" }
- Đăng nhập:
 - * Phương thức: POST
 - * URL: /user/login
 - * Yêu cầu đăng nhập: không

- * Quyền truy cập: không yêu cầu
- * Dữ liệu gửi lên:

```
{ password: "123",
  email: "email", }
```
- Cập nhật thông tin người dùng:
 - * Phương thức: PUT
 - * URL: /user/update
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ name: "name",
  email: "email",
  phone: "1234567" }
```
- Thay đổi mật khẩu người dùng:
 - * Phương thức: PUT
 - * URL: /user/changepass
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ currPass: "123",
  newPassword: "567" }
```
- Reset mật khẩu người dùng:
 - * Phương thức: POST
 - * URL: /user/resetpass
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ email: "email" }
```

– Kích hoạt tài khoản người dùng:

- * Phương thức: PUT
- * URL: /user/active
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
 { email: "email",
 activeToken: "abc" }

– Chứng thực JWT token của người dùng:

- * Phương thức: GET
- * URL: /user/verifytoken
- * Yêu cầu đăng nhập: không
- * Quyền truy cập: không yêu cầu
- * Dữ liệu gửi lên:
 { token: "abc" }

– Lấy danh sách tất cả người dùng:

- * Phương thức: GET
- * URL: /user/all
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: admin
- * Dữ liệu trả về: mảng các đối tượng người dùng

– Xóa người dùng:

- * Phương thức: DELETE
- * URL: /user/delete
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: admin
- * Dữ liệu gửi lên:
 { userId: 12 }

- Lấy thông tin người dùng:
 - * Phương thức: GET
 - * URL: /user/detail
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ userId: 12 }
```
 - * Dữ liệu trả về: tất cả thông tin của người dùng (bao gồm mật khẩu và token kích hoạt).
- Cập nhật role của người dùng:
 - * Phương thức: PUT
 - * URL: /user/detail
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: admin
 - * Dữ liệu gửi lên:

```
{ userId: 12,  
newRole: "mod" }
```
- Board điều khiển (Device):
 - Lấy tất cả board điều khiển:
 - * Phương thức: GET
 - * URL: /device/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ user.id: 12 }
```
 - * Dữ liệu trả về: mảng các đối tượng board điều khiển.
 - Lấy thông tin của các board đang chạy:

- * Phương thức: GET
- * URL: /device/running
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ user.id: 12 }
- * Dữ liệu trả về: mảng các đối tượng board điều khiển đang chạy..
- Cập nhật trạng thái board điều khiển:
 - * Phương thức: PUT
 - * URL: /device/status
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ mac: "112233445566",
newStatus: "running" }
- Lấy thông tin của một board điều khiển:
 - * Phương thức: GET
 - * URL: /device/one
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ mac: "112233445566", }
 - * Dữ liệu trả về: một đối tượng board điều khiển
- Thêm board điều khiển mới:
 - * Phương thức: POST
 - * URL: /device/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member

- * Dữ liệu gửi lên:

```
{ mac: "112233445566",
  name: "device 1",
  manufacture: "sony",
  status: "no connection", }
```
- Thêm board điều khiển mới:
 - * Phương thức: POST
 - * URL: /device/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ mac: "112233445566",
  name: "device 1",
  manufacture: "sony",
  status: "no connection", }
```
- Xóa board điều khiển:
 - * Phương thức: DELETE
 - * URL: /device/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ mac: "112233445566" }
```
- Lịch trình (schedule)
 - Lấy thông tin tất cả lịch trình hiện tại của một mùa vụ:
 - * Phương thức: GET
 - * URL: /schedule/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member

- * Dữ liệu gửi lên:
{ cropId: 12 }
- * Dữ liệu trả về: mảng các đối tượng lịch trình (schedule)
- Xóa một lịch trình:
 - * Phương thức: DELETE
 - * URL: /schedule/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12,
scheduleId: 12 }
- Đồng bộ lịch trình hiện tại xuống thiết bị:
 - * Phương thức: GET
 - * URL: /schedule/sync
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12,
mac: "112233445566" }
- Thêm vào một lịch trình cài đặt:
 - * Phương thức: POST
 - * URL: /schedule/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12,
actuatorId: 12,
starttime: 030303,

```

endtime: 050505,
intervaltime: 45,
delaytime: 5 }
```

- Nguồn dữ liệu (threshold):

- Lấy nguồn dữ liệu mới nhất:
 - * Phương thức: GET
 - * URL: /threshold/newest
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { cropId: 12 }
 - * Dữ liệu trả về: Nguồn dữ liệu cao và thấp mới nhất của mùa vụ hiện tại.
- Thêm vào một nguồn dữ liệu mới:
 - * Phương thức: POST
 - * URL: /threshold/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
 - { temperatureLower: 23,
 - temperatureUpper: 25,
 - humidityLower: 45,
 - humidityUpper: 50,
 - ppmLower: 100,
 - ppmUpper: 200,
 - lightLower: 1000,
 - lightUpper: 3000 }

- Actuator (máy bơm, quạt, đèn, ...)

- Thêm một actuator mới cho board

- * Phương thức: POST
- * URL: /actuator/add
- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:

```
{ actuator: { status: "off",  
idOnboard: 12,  
name: "water 1",  
type: "water",  
priority: "Primary" },  
deviceMac: "112233445566" }
```
- Lấy tất cả các actuator trên board:
 - * Phương thức: GET
 - * URL: /actuator/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ deviceMac: "112233445566" }
```
 - * Dữ liệu trả về: mảng các đối tượng actuator
- Thay đổi trạng thái của actuator:
 - * Phương thức: PUT
 - * URL: /actuator/status
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ Id: 12,  
Status: "on" }
```
- Thay đổi độ ưu tiên của một actuator:

- * Phương thức: PUT
 - * URL: /actuator/priority
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ Id: 12,
priority: "Secondary" }
- Xóa một actuator:
 - * Phương thức: DELETE
 - * URL: /actuator/priority
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ Id: 12 }
- Bài viết (article)
 - Lấy tất cả các bài viết:
 - * Phương thức: GET
 - * URL: /article/all
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu trả về: mảng các đối tượng article
 - Lấy một bài viết theo ID:
 - * Phương thức: GET
 - * URL: /article/one
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu trả về: đối tượng article

- Thêm một bài viết mới:
 - * Phương thức: POST
 - * URL: /article/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ content: "content",  
  title: "title" }
```
- Xóa một bài viết:
 - * Phương thức: DELETE
 - * URL: /article/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: mod
 - * Dữ liệu gửi lên:

```
{ id: 12 }
```
- Kiểm duyệt bài viết:
 - * Phương thức: PUT
 - * URL: /article/check
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: mod
 - * Dữ liệu gửi lên:

```
{ id: 12,  
  checked: true }
```
- Bình luận (comment)
 - Thêm một bình luận cho bài viết:
 - * Phương thức: POST
 - * URL: /comment/add

- * Yêu cầu đăng nhập: có
- * Quyền truy cập: member
- * Dữ liệu gửi lên:
{ content: "content",
ArticleId: 12 }
- Lấy tất cả các comment trong bài viết:
 - * Phương thức: GET
 - * URL: /comment/all
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:
{ ArticleId: 12 }
 - * Dữ liệu trả về: mảng các đối tượng comment
- Xóa một comment:
 - * Phương thức: DELETE
 - * URL: /comment/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: mod
 - * Dữ liệu gửi lên:
{ commentId : 12 }
- Mùa vụ (crop)
 - Lấy tất cả các mùa vụ của một board:
 - * Phương thức: GET
 - * URL: /crop/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ mac: "112233445566" }

- Lấy một mùa vụ bằng ID:
 - * Phương thức: GET
 - * URL: /crop/one
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12 }
 - * Dữ liệu trả về: một đối tượng crop
- Lấy mùa vụ mới nhất đang chạy trên board:
 - * Phương thức: GET
 - * URL: /crop/newest
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ mac: "112233445566" }
 - * Dữ liệu trả về: một đối tượng crop
- Thêm một mùa vụ mới:
 - * Phương thức: POST
 - * URL: /crop/add
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ DeviceMac: "112233445566",
status: "pending",
name: "crop 1",
startdate: "12-20-2017 05:00:00",
closedate: "01-20-2017 05:00:00",
treetype: "salad",
reporttime: 60 }

- Xóa một mùa vụ:
 - * Phương thức: DELETE
 - * URL: /crop/delete
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

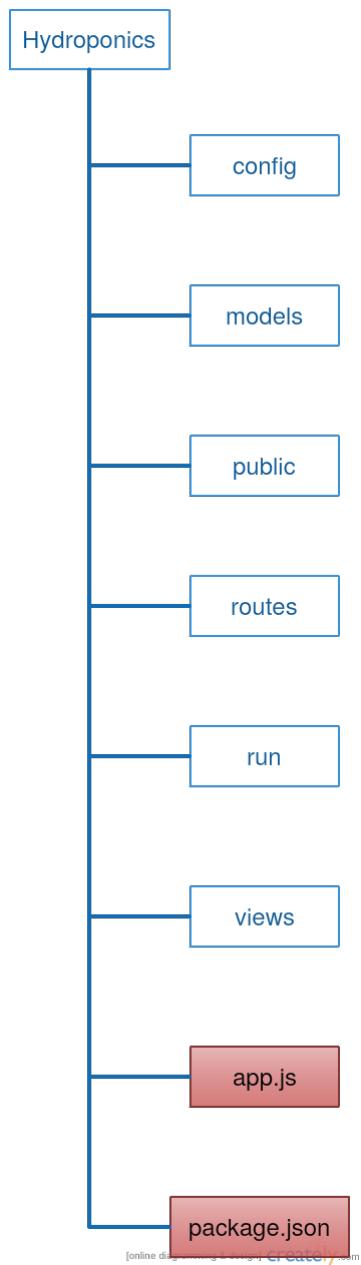
```
{ cropId: 12 }
```
- Thay đổi thông tin mùa vụ:
 - * Phương thức: PUT
 - * URL: /crop/edit
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ name: "crop 1",  
startdate: "12-20-2017 05:00:00",  
closedate: "01-20-2017 05:00:00",  
treetype: "salad",  
reporttime: 60 }
```
- Thay đổi chế độ chia sẻ:
 - * Phương thức: PUT
 - * URL: /crop/share
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:

```
{ Id: 12,  
share: true }
```
- Tìm mùa vụ:
 - * Phương thức: GET

- * URL: /crop/search
 - * Yêu cầu đăng nhập: không
 - * Quyền truy cập: không yêu cầu
 - * Dữ liệu gửi lên:
{ tree: "salad",
month: 12 }
- Dữ liệu cảm biến (data):
 - Lấy tất cả dữ liệu từ board gửi lên :
 - * Phương thức: GET
 - * URL: /data/all
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12 }
 - * Dữ liệu trả về: mảng các đối tượng data.
 - Lấy dữ liệu mới nhất:
 - * Phương thức: GET
 - * URL: /data/newest
 - * Yêu cầu đăng nhập: có
 - * Quyền truy cập: member
 - * Dữ liệu gửi lên:
{ cropId: 12 }
 - * Dữ liệu trả về: đối tượng data.

Mô hình cụm server



Hình 110: Cây thư mục

Cấu trúc thư mục Server được tổ chức theo mô hình MVC với các thư mục chính:

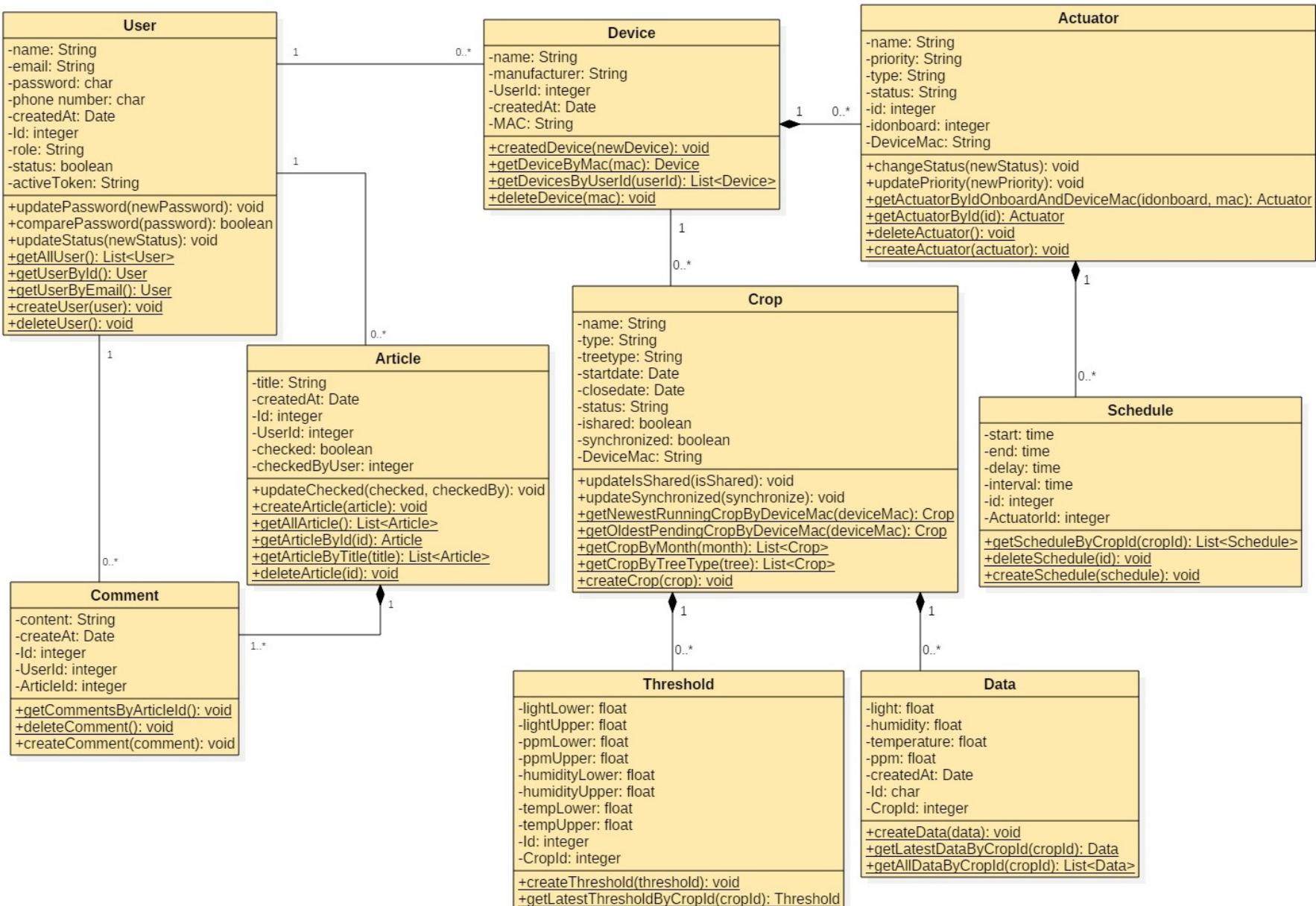
- Models: chứa các model cho các đối tượng tương ứng trong database
- Routes: controller tiếp nhận các request và xử lý
- Views : các file html hiển thị

Ngoài ra, các thư mục khác:

- Config: thiết lập thông tin database khi kết nối
- Public: chứa mã css, javascript, angularjs, images, các thư viện front-end cần dùng.
- Run: cấu hình cho server như port,...
- File app.js: file server chính
- File package.js: các gói hỗ trợ trong ứng dụng dành cho nodejs

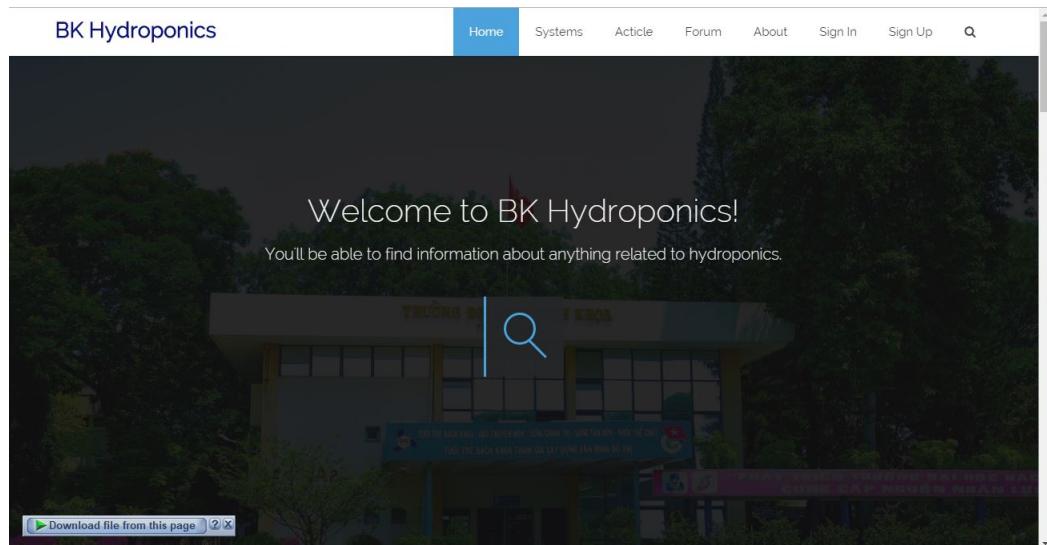
4.2.2 Xây dựng database

Sử dụng Sequelize để hiện thực database giống như sơ đồ sau:



Hình 111: Thiết kế class

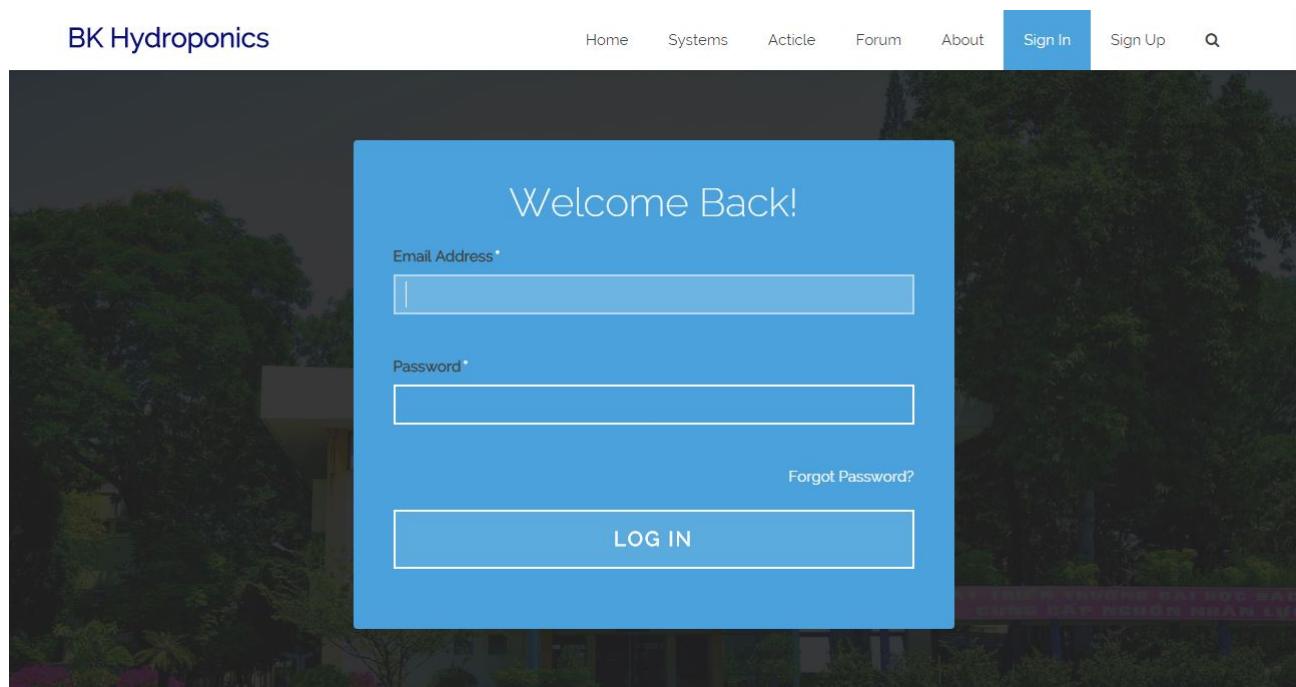
4.2.3 Một số hình ảnh



Hình 112: Trang chủ.

A screenshot of the BK Hydroponics sign-up page. The header includes the site name "BK Hydroponics" and a navigation bar with Home, Systems, Article, Forum, About, Sign In, and a prominent blue "Sign Up" button. The main content is a form titled "Sign Up" with fields for "Your Name*", "Email Address*", "Phone*", "Set A Password*", and "Confirm Password*". Each field is accompanied by a small asterisk indicating it is required. Below the fields is a large blue "GET STARTED" button.

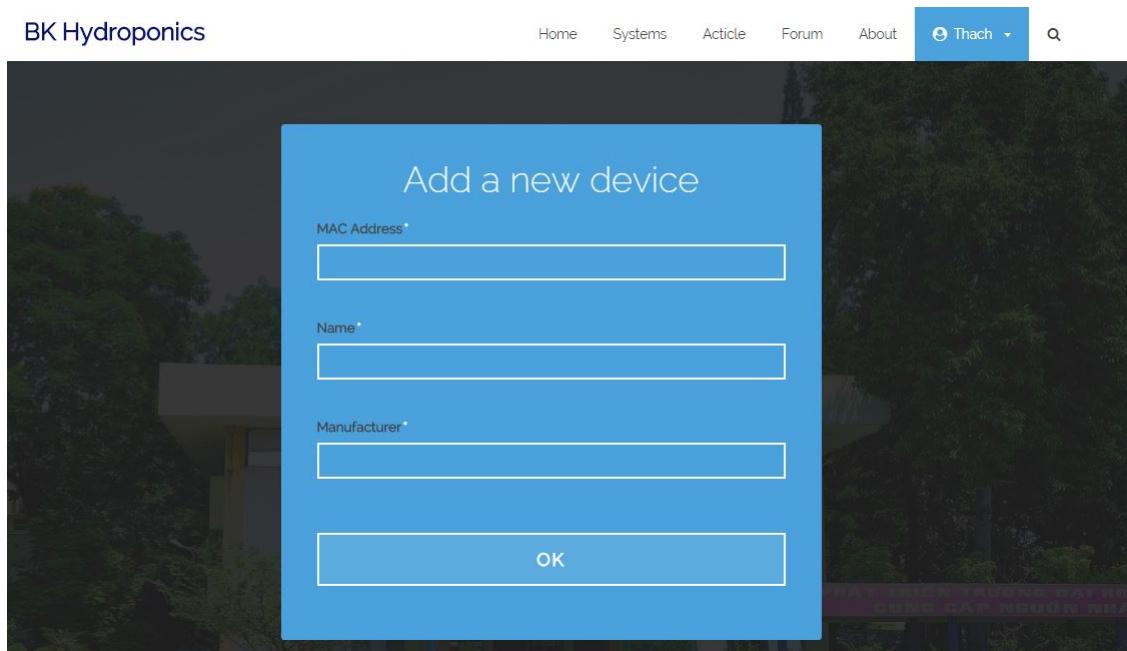
Hình 113: Trang đăng ký.



Hình 114: Trang đăng nhập.

A screenshot of a user profile page for "Thach". The top navigation bar is identical to the login page. The main content area shows a placeholder profile picture, the name "Thach", and three contact details: an envelope icon with the email "hbathach@gmail.com", a phone icon with the number "01696030126", and a plug icon with the text "List Devices". A "Edit profile" button is located at the bottom of the profile section.

Hình 115: Trang cá nhân.



Hình 116: Thêm một board mới.

YOUR DEVICES							
List of your devices.							
No.	MAC Address	Device Name	Installation Date	Status	Remarks	Operation	
1	223344556677	smart device 22	13-12-2017 10:35:22	stop	sony		
Showing 1 to 1 of 1 entries							
 Add new device							

Hình 117: Danh sách các board điều khiển.

List relays							
No.	ID On Board	Name	Type	Priority	Status	Created Date	Operation
1	11	water 1	Water	Primary	on	09-12-2017 17:29:47 +0700	
Showing 1 to 1 of 1 entries							
New Relay							

Hình 118: Danh sách các actuator của một board.

Add new actuator for this device

ID	<input type="text"/>
Priority	<input type="text" value="--Please select--"/>
Name	<input type="text"/>
Type	<input type="text" value="--Please select--"/>

On Board 11

Created Date 09-12-2017 17:29:47

OK CLOSE

Hình 119: Thêm một actuator.

List of crops										
List of crops of your device: 112233445566										
No.	Crop Name	Type of Hydroponics	Tree Type	Start Date	End Date	Status	Operation	Share		
1	crop 1	aeroponics	salad	10-12-2017 15:51:00	31-12-2017 15:51:00	running				

Showing 1 to 1 of 1 entries

Previous 1 Next

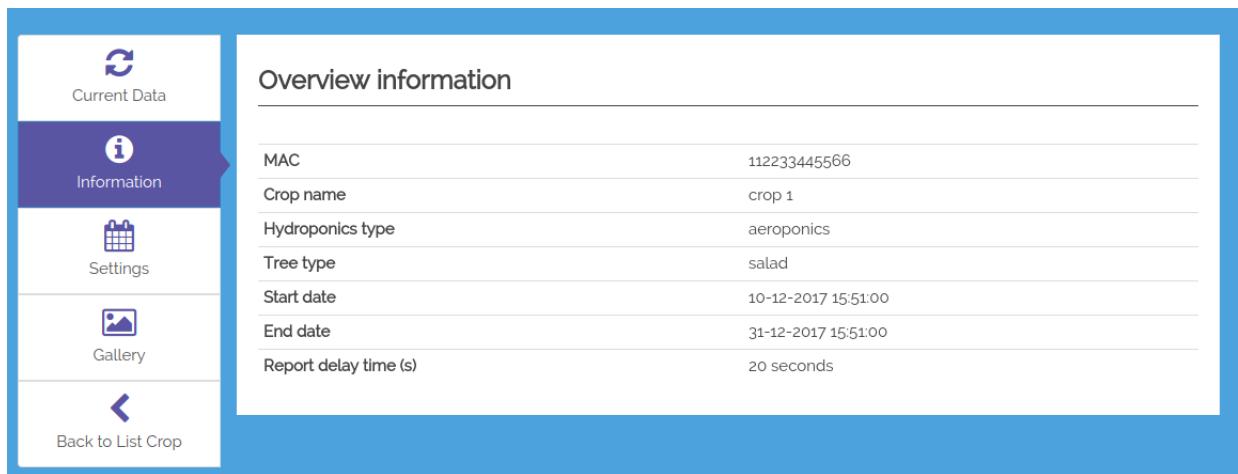
[Add new crop](#)

Hình 120: Danh sách các mùa vụ.

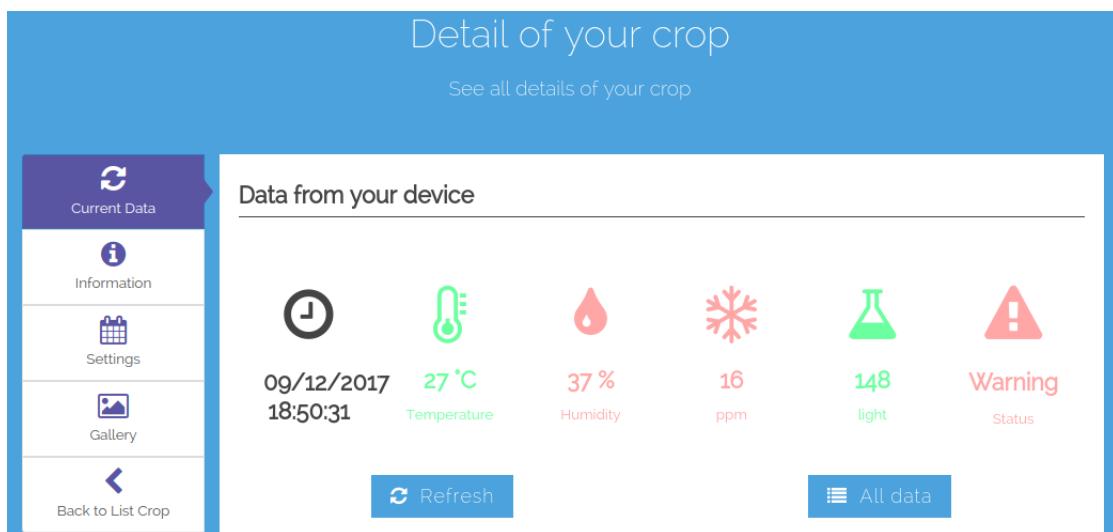
Add new crop for this device

Crop Name	<input type="text"/>
Tree Type	<input type="text"/>
Starting Date	<input type="text"/>
Closing Date	<input type="text"/>
Report Time	<input type="text"/> seconds
Type:	<input type="text" value="---Please select---"/>
OK CLOSE	

Hình 121: Thêm một mùa vụ.



Hình 122: Thông tin mùa vụ.



Hình 123: Chi tiết mùa vụ.

The screenshot shows a sidebar menu with icons for Current Data, Information, Settings (selected), Gallery, and Back to List Crop. The main area is titled 'Hardware Settings' with a status indicator (Status: checked) and a sync button. It displays a table of scheduled entries:

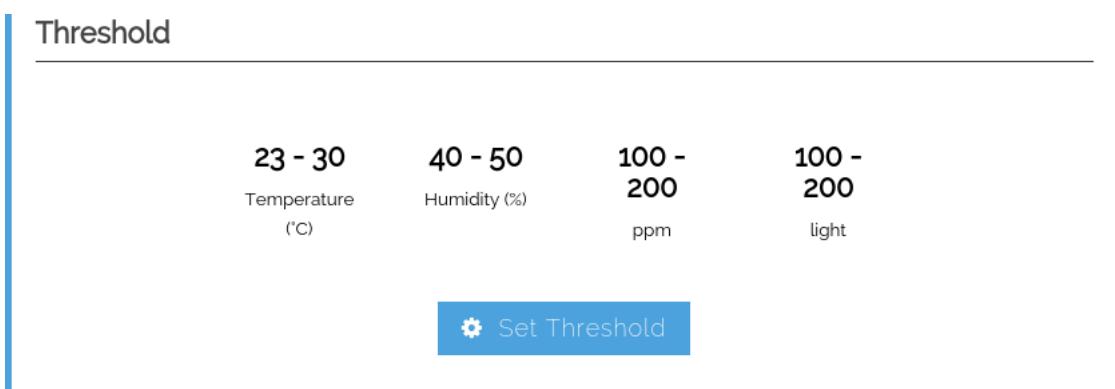
Type	Name	Start Time	End Time	Interval Time	Delay Time	Operation
Water	water 1	17:33:00	17:40:00	5	5	Edit Settings

Showing 1 to 1 of 1 entries. Navigation buttons: Previous, 1, Next.

Hình 124: Lịch trình hoạt động của các actuator.

The modal dialog is titled 'Add new settings for this device'. It contains fields for Relay Type (dropdown), Relay Name (dropdown), Start Time (text input with a clock icon), End Time (text input with a clock icon), Interval Time (text input with 'seconds' unit), and Delay Time (text input with 'seconds' unit). At the bottom are 'OK' and 'CLOSE' buttons.

Hình 125: Thêm lịch trình cho actuator.



Hình 126: Nguồn dữ liệu.

The screenshot shows a NodeBB forum interface with the following categories:

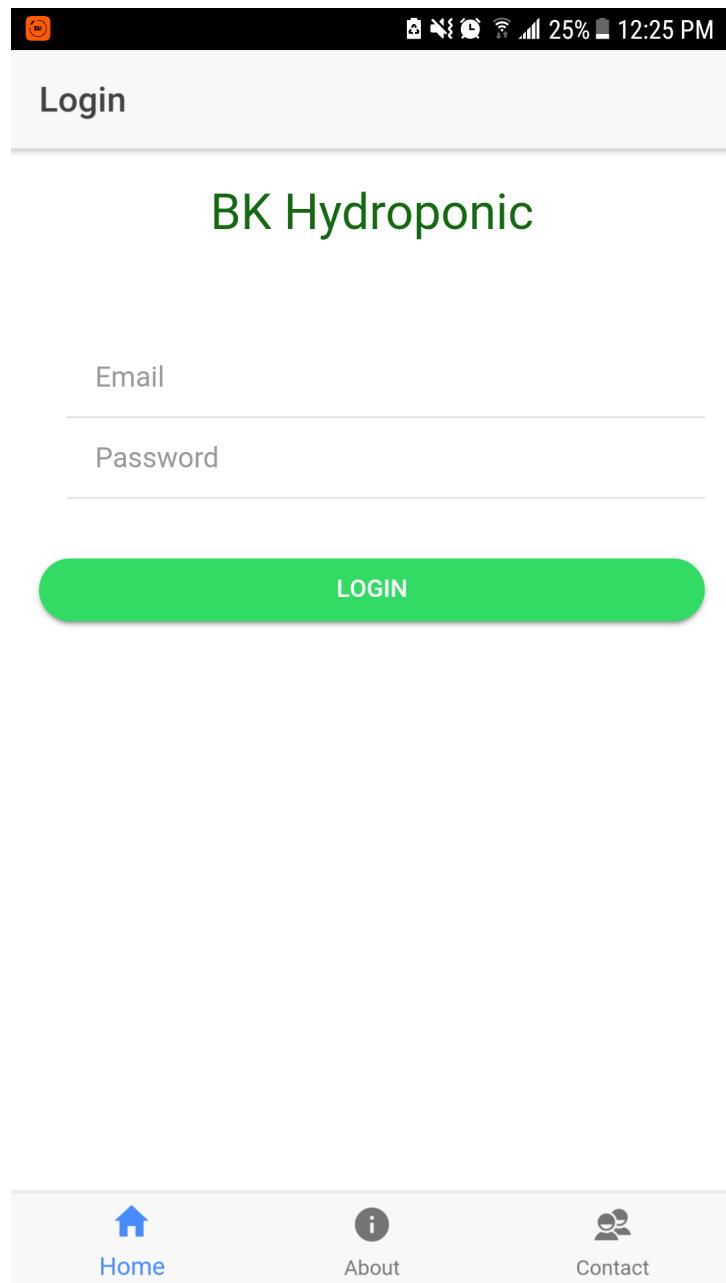
- Thông báo**: Các thông báo của ban quản trị. 0 topics, 0 posts. Status: No new posts.
- Hỏi đáp**: Đặt câu hỏi và giải đáp các thắc mắc. 1 topics, 1 posts. Status: about 5 hours from now. Welcome to your brand new NodeBB forum! This is what a topic and post looks like. As
- Chia sẻ**: Các bài viết chia sẻ kinh nghiệm. 0 topics, 0 posts. Status: No new posts.
- Góc kỹ thuật**: Trao đổi các kỹ thuật trồng trọt. 0 topics, 0 posts. Status: No new posts.

At the bottom, it says "Powered by NodeBB | Contributors".

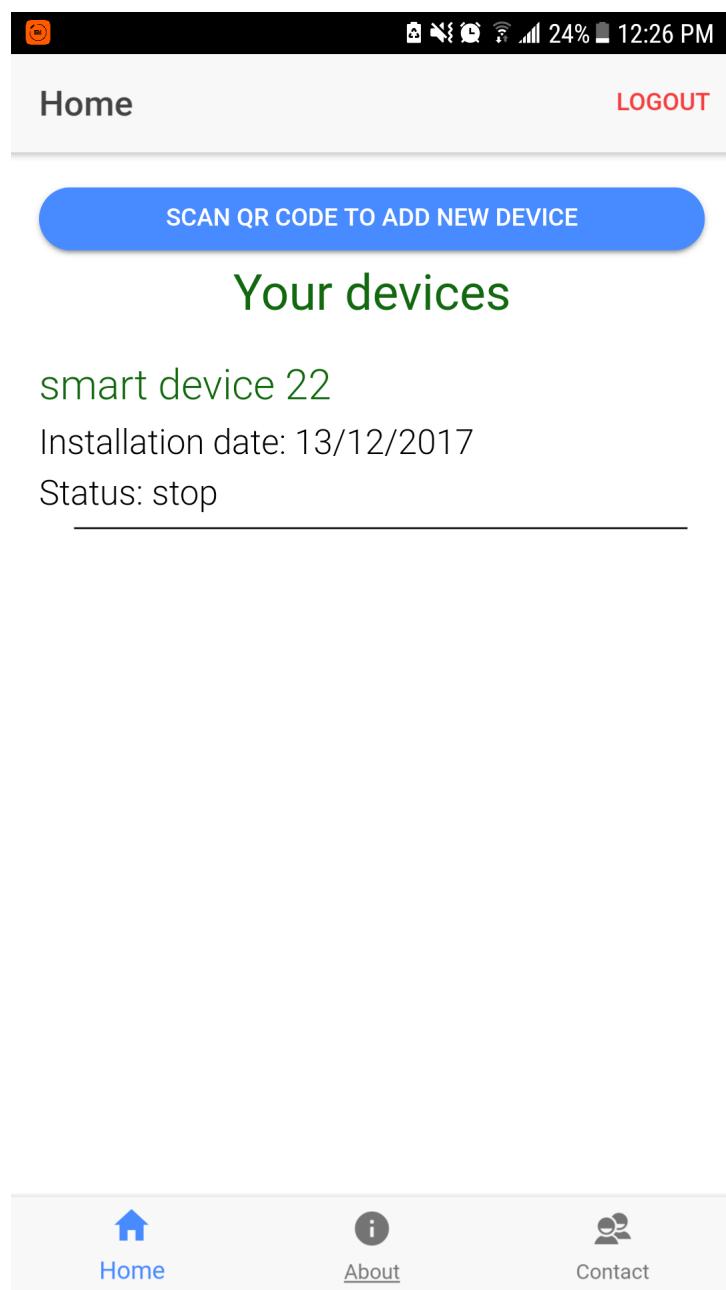
Hình 127: Trang forum.

4.3 Ứng dụng di động

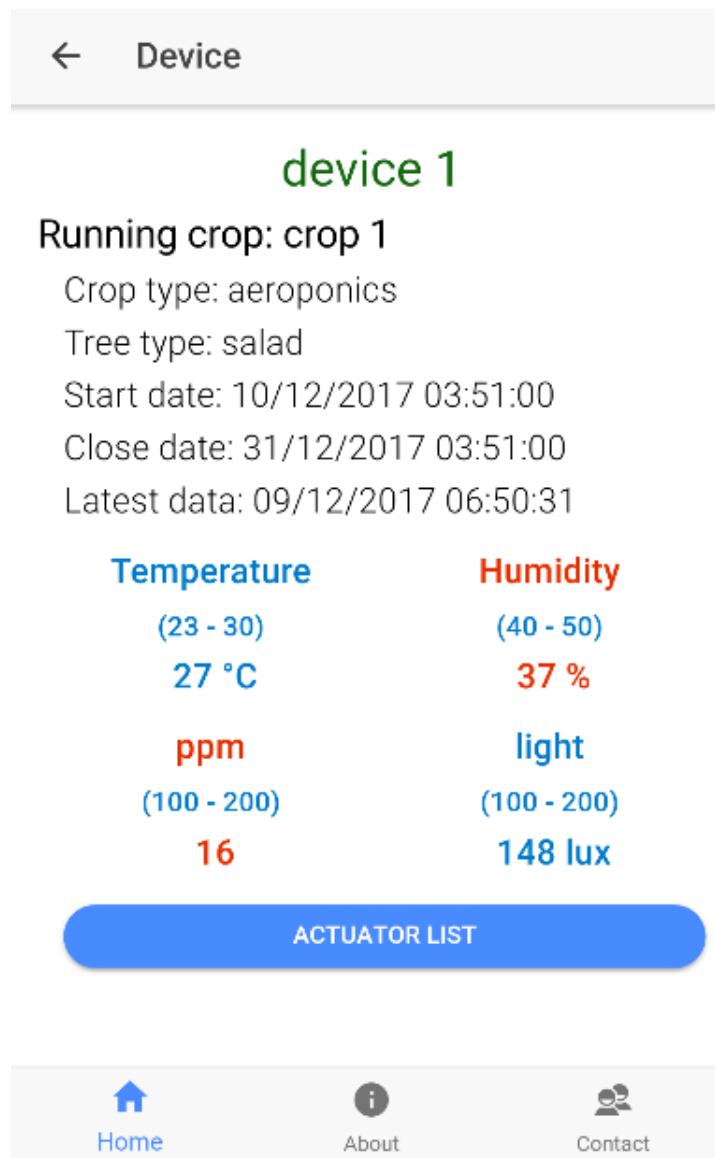
4.3.1 Một số hình ảnh



Hình 128: Trang đăng nhập.



Hình 129: Trang chủ sau khi đăng nhập.



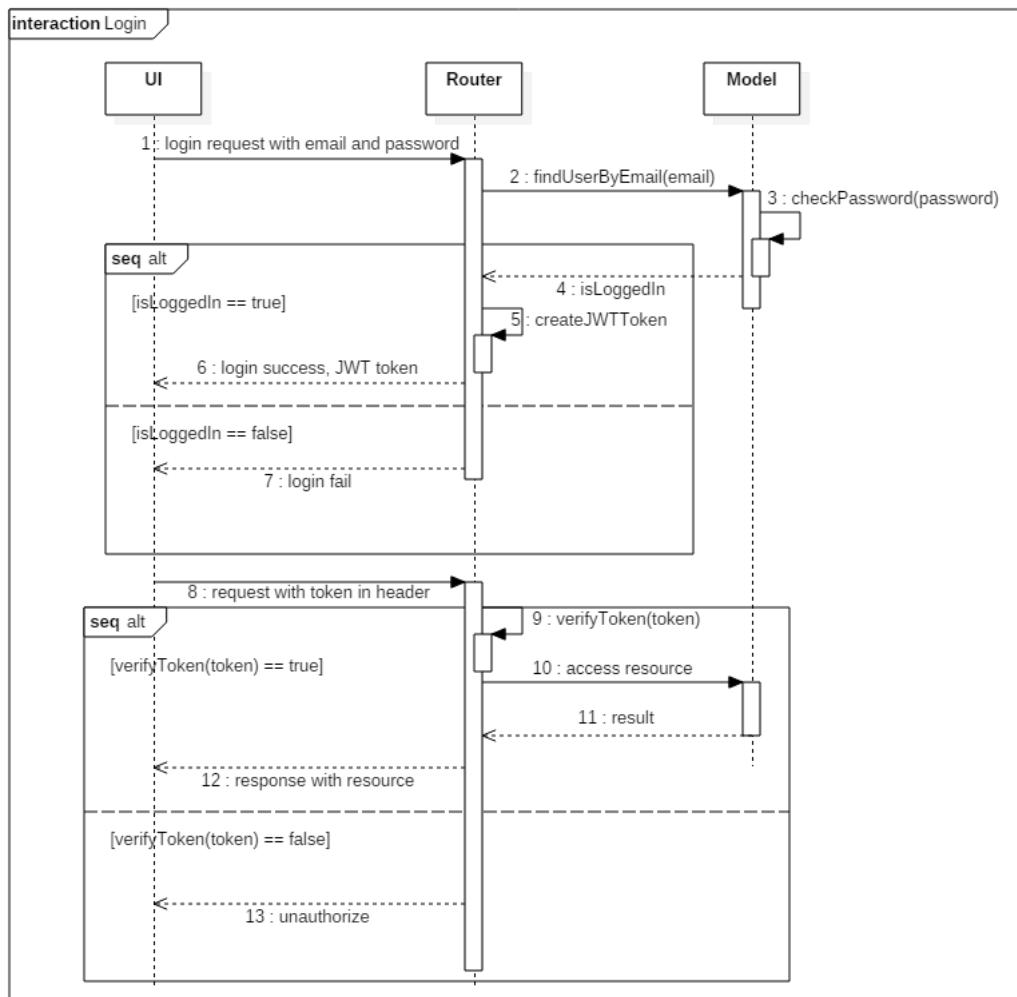
Hình 130: Thông tin mùa vụ và dữ liệu.

Type	Name	Priority	Active
Water	water 1	Primary	Active

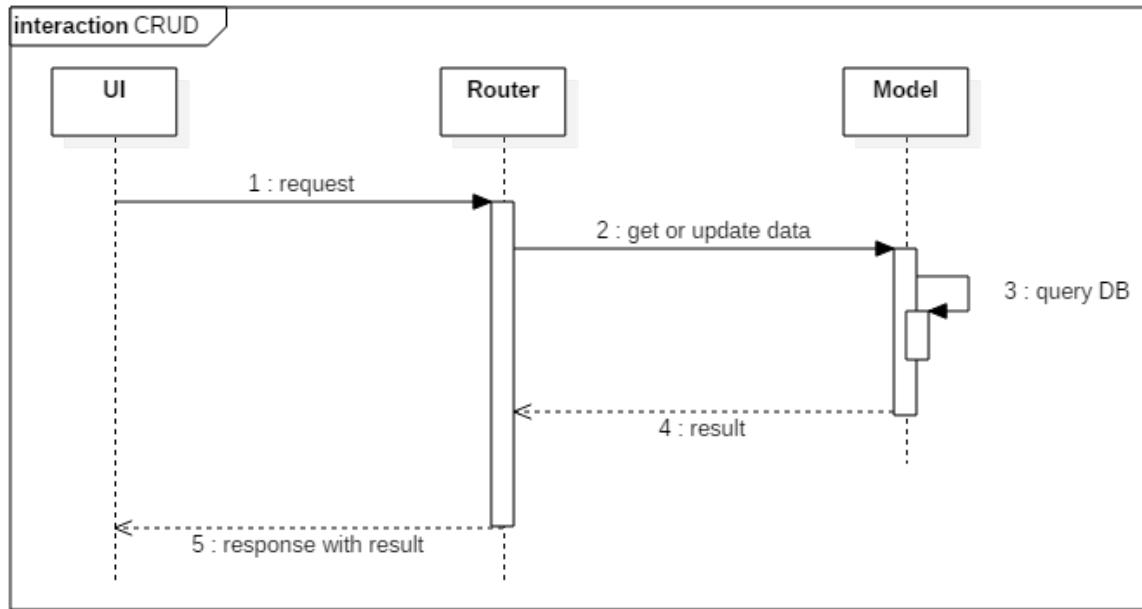
Hình 131: Danh sách các actuator.

4.4 Tương tác hệ thống

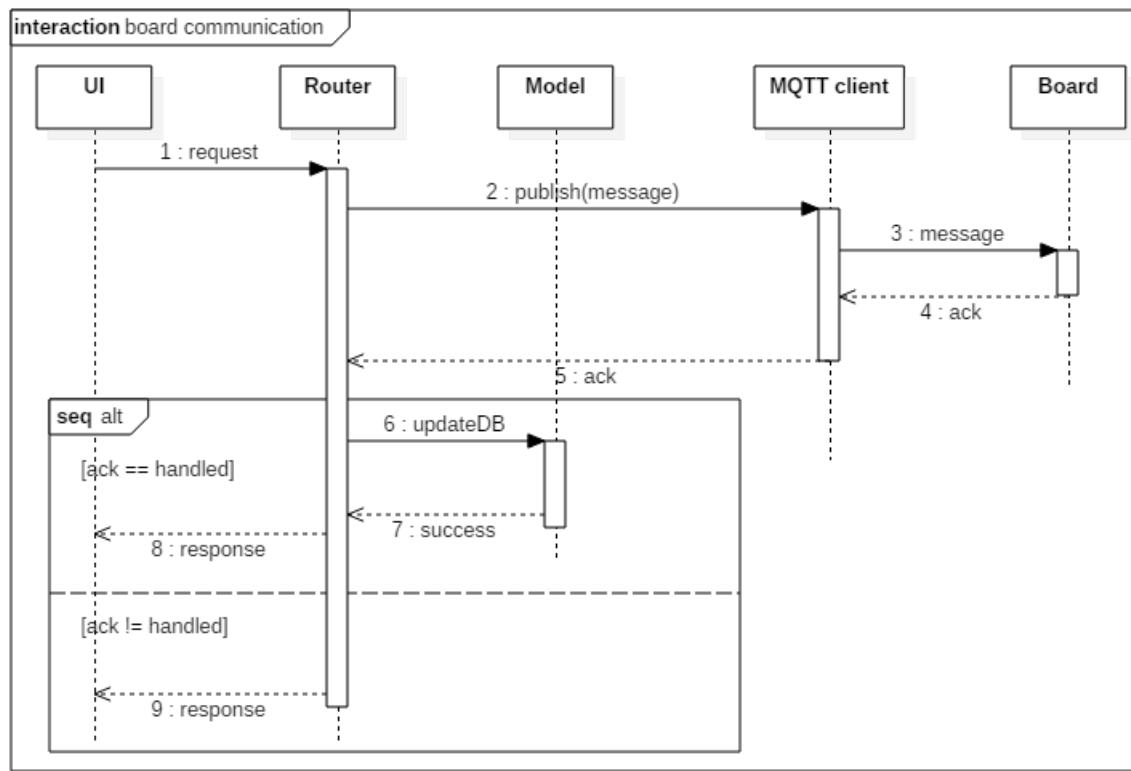
Các biểu đồ tuần tự sau sẽ thể hiện sự tương tác giữa người dùng và hệ thống:



Hình 132: Thao tác đăng nhập vào hệ thống sử dụng JWT.



Hình 133: Nhóm các thao tác CRUD.



Hình 134: Nhóm các thao tác tương tác với board điều khiển.

Toàn bộ các gói tin truyền giữa board điều khiển và web server đều được mã hóa XXTEA.

5 KIỂM THỬ HỆ THỐNG

5.1 Web server

5.2 Board điều khiển

Hệ thống mạch điều khiển đã thực hiện được các chức năng:

- Lấy được dữ liệu từ các sensor nhiệt độ, độ ẩm.
- Lấy được dữ liệu từ module thời gian thực.
- Điều khiển được relay cho các bộ điều khiển.
- Chức năng Web server của Esp8266.
- Chức năng Mqtt client của Esp8266.

6 TỔNG KẾT VÀ HƯỚNG PHÁT TRIỂN CHO TƯƠNG LAI

6.1 Tổng kết

6.1.1 Kết quả đạt được

- Web server API và web app:

Sử dụng Nodejs và Angularjs hiện thực thành công các chức năng:

- Cung cấp hệ thống API cho web app và mobile app.
- Phân quyền người dùng: member, mod và admin.
- Vẽ đồ thị theo dõi các thông số môi trường.
- Giao tiếp với board điều khiển thông qua hệ thống giao thức đã thiết kế.

- Mobile app:

Sử dụng framework Ionic 2 hiện thực thành công các chức năng:

- Đăng nhập
- Xem danh sách các board điều khiển.
- Quét mã QR để nhanh chóng thêm board điều khiển mới.
- Giám sát thông tin môi trường.
- Theo dõi các actuator và điều khiển bật, tắt chúng.

6.1.2 Thuận lợi và khó khăn trong quá trình thực hiện đề tài

Thuận lợi

- Được sự hỗ trợ về kinh phí và kiến thức từ giáo viên hướng dẫn.

Khó khăn

- Dành nhiều thời gian để tìm hiểu các công nghệ mới.

- Dành nhiều thời gian cho việc hiện thực, chưa có nhiều thời gian để thử nghiệm hệ thống.
- Thiếu các thiết bị di động để thử nghiệm ứng dụng di động trên nhiều nền tảng khác nhau.
- Khó khăn trong việc tìm thiết kế và hiện thực giao thức giao tiếp, gửi nhận các gói tin giữa board điều khiển và web api.

6.1.3 Hướng phát triển

Ứng dụng web

- Nghiên cứu kỹ hơn về ảnh hưởng của các nhân tố môi trường (nhiệt độ, độ ẩm, pH, ppm, ánh sáng, oxy) đến sự phát triển của các loại cây phổ biến trong từng thời kỳ, từ đó đưa ra các giải pháp tự động hóa việc chăm sóc cho từng giai đoạn phát triển của cây.
- Người dùng có thể upload ảnh đại diện, upload ảnh về các mùa vụ mà mình đang trồng và chia sẻ nó.
- Cho phép người dùng xuất ra các file setting của một mùa vụ nào đó và chia sẻ nó, người dùng khác có thể nạp file đó vào profile của mình và chạy trên thiết bị.
- Khắc phục thời gian chờ gói tin ACK từ board điều khiển bằng cách thiết lập một khoảng thời gian nhất định cho việc chờ gói tin ACK. Quá khoảng thời gian đó thì phản hồi về cho website là gói tin không đến được board điều khiển.
- Cải thiện hơn về giao diện.

Ứng dụng di động

- Cải thiện hơn về giao diện.

Board điều khiển

- Thay thế board Arduino + ESP8266 bằng một board khác tích hợp cả hai chức năng.



Hình 135: Kit RF thu phát wifi Node MCU.

- Cài tiến module đo pH, ppm
- Cài tiến tính năng điều khiển tự động của board.
- Cải thiện tính thời gian thực.
- Cài tiến chức năng web server của ESP8266.

6.1.4 Kiến thức thu thập được

Tài liệu

- [1] http://vietjack.com/nodejs/nodejs_la_gi.jsp
- [2] <https://freetuts.net/javascript-la-gi-viet-ung-dung-javascript-dau-tien-263.html>
- [3] <http://docs.sequelizejs.com/>
- [4] <https://freetuts.net/json-la-gi-cau-truc-chuoi-json-236.html>
- [5] <https://freetuts.net/tong-quan-ve-angularjs-200.html>
- [6] <http://arduino.vn/bai-viet/1236-tim-hieu-giao-thuc-mqtt>
- [7] <https://viblo.asia/p/gioi-thieu-ve-angular-2-APqzearpzVe>
- [8] <https://code.tutsplus.com/vi/articles/introduction-to-ionic-2–cms-28193>
- [9] <https://vi.wikipedia.org/wiki/TypeScript>
- [10] <https://vi.wikipedia.org/wiki/PostgreSQL>
- [11] <https://docs.archive.nodebb.org/vi/latest/index.html>
- [12] <https://en.wikipedia.org/wiki/XXTEA>