

**TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN VÀ
TRUYỀN THÔNG THÁI NGUYÊN
KHOA CÔNG NGHỆ TỰ ĐỘNG HÓA**



**ĐỒ ÁN TỐT NGHIỆP
CHUYÊN NGÀNH CÔNG NGHỆ KỸ THUẬT ĐIỆN, ĐIỆN TỬ**

Đề tài:

*Xây dựng ứng dụng IOT trên nền tảng Windows và React Native điều khiển
công tắc thông minh*

Sinh viên thực hiện	: Phan Ngọc Phong
Giáo viên hướng dẫn	: TS Lê Văn Chung
Lớp	: KTDĐT K16A

Thái Nguyên ngày 31 tháng 10 năm 2021

LỜI NÓI ĐẦU

Nếu như Cách mạng công nghiệp đầu tiên sử dụng năng lượng nước và hơi nước để cơ giới hóa sản xuất. Cuộc cách mạng lần 2 diễn ra nhờ ứng dụng điện năng để sản xuất hàng loạt. Cuộc cách mạng lần 3 sử dụng điện tử và công nghệ thông tin để tự động hóa sản xuất. Bây giờ, cuộc Cách mạng Công nghiệp thứ tư đang nảy nở từ cuộc cách mạng lần ba, nó kết hợp các công nghệ lại với nhau, làm mờ ranh giới giữa vật lý, kỹ thuật số và sinh học. Cách mạng công nghệ 4.0 tạo điều kiện thuận lợi cho việc tạo ra các nhà máy thông minh, sự phát triển của internet vạn vật giúp tạo ra bản sao ảo của thế giới vật lý, cho phép mọi người ở khắp nơi trên thế giới kết nối với nhau thông qua mạng internet dịch vụ qua các thiết bị di động ở mọi lúc, mọi nơi.

Công nghiệp 4.0 thúc đẩy chuyển đổi kỹ thuật số của sản xuất thông qua việc tích hợp các hệ thống và quy trình khác nhau trước đây thông qua các hệ thống máy tính được kết nối với nhau qua chuỗi cung ứng và giá trị. Cách mạng Công nghiệp 4.0 đang báo hiệu một sự thay đổi trong bối cảnh sản xuất truyền thống bao gồm ba xu hướng công nghệ thúc đẩy sự chuyển đổi này: kết nối, thông minh và tự động hóa linh hoạt (IOT) .

Nhằm ứng dụng IOT vào trong cuộc sống hàng ngày nên em đã chọn đề tài: *Xây dựng ứng dụng IOT trên nền tảng Windows và React Native điều khiển công tắc thông minh*. Đề tài này đã giúp em hiểu biết hơn về ứng dụng của IOT trong cuộc sống hàng ngày.

Để hoàn thành đề tài này, ngoài sự nỗ lực của bản thân em xin chân thành gửi lời cảm ơn sâu sắc tới các thầy cô Khoa Công Nghệ Tự Động Hóa trường Đại học Công nghệ thông tin và Truyền thông Thái Nguyên, đặc biệt là thầy **TS Lê Văn Chung** đã tận tình hướng dẫn, giúp đỡ và tạo điều kiện tốt nhất cho em kể từ khi nhận đề tài tới khi hoàn thành đề tài này.

Em xin chân thành cảm ơn!

MỤC LỤC

LỜI NÓI ĐẦU	ii
MỤC LỤC	iii
DANH MỤC HÌNH ẢNH	vii
DANH SÁCH BẢNG BIỂU	1
DANH MỤC TỪ VIẾT TẮT	2
CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI	3
1.1 Giới thiệu chung	3
1.1.1 <i>Giới thiệu chung về IOT</i>	3
1.1.2 <i>Công tắc thông minh</i>	4
1.2 Tổng quan các công nghệ sử dụng	5
1.2.1 <i>Giới thiệu ngôn ngữ C#</i>	5
1.2.2 <i>Công nghệ Wifi</i>	7
1.2.3 <i>Giới thiệu tổng quan ngôn ngữ C/C++</i>	8
1.2.4 <i>Giới thiệu tổng quan về React Native</i>	9
1.2.5 <i>Giới thiệu tổng quan về công nghệ Winform Application</i>	12
1.2.6 <i>JavaScript Object Notation</i>	14
1.3 Giao thức MQTT	15
1.3.1 <i>Các khái niệm cơ bản</i>	15
1.3.2 <i>Packet</i>	16
1.3.3 <i>MQTT Publish & Subscribe</i>	18
1.3.4 <i>Ưu điểm, nhược điểm của MQTT và một số ứng dụng triển khai</i>	19

1.4	Bunifu framework	20
1.4.1	<i>Giới thiệu</i>	20
1.4.2	<i>Một số thành phần cơ bản</i>	21
CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH VÀ CÔNG CỤ THIẾT KẾ		24
2.1	Android Studio	24
2.1.1	<i>Giới thiệu</i>	24
2.1.2	<i>Cài đặt Android Studio</i>	24
2.2	Visual Studio	27
2.2.1	<i>Visual Studio là gì ?</i>	27
2.2.2	<i>Các tính năng phần mềm trên Visual Studio</i>	27
2.3	Visual Studio Code và Platform IO	28
2.3.1	<i>Visual Studio Code</i>	28
2.3.2	<i>Platform IO</i>	28
2.4	Altium Design	30
2.4.1	<i>Giới thiệu</i>	30
2.4.2	<i>Một số đặc trưng</i>	30
2.5	MQTT Fx	31
2.5.1	<i>Giới thiệu</i>	31
2.5.2	<i>Các bước kết nối</i>	32
2.6	Esp8266	32
2.6.1	<i>Giới thiệu</i>	32
2.6.2	<i>Thông số kỹ thuật</i>	33
2.6.3	<i>Chức năng các I/O</i>	33
CHƯƠNG 3: MÔ TẢ VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG		35

3.1	Mô tả và yêu cầu của hệ thống	35
3.2	Phân tích thiết kế hệ thống	35
3.2.1	<i>Sơ đồ tổng quan hệ thống</i>	35
3.2.2	<i>Lưu đồ thuật toán phân cứng</i>	36
3.3	Thiết kế mạch điều khiển	37
3.3.1	<i>Các khối cơ bản trong mạch điều khiển</i>	37
3.3.2	<i>Thiết kế mạch điều khiển</i>	37
3.4	Xây dựng chuẩn giao tiếp giữa app và thiết bị phần cứng	42
3.4.1	<i>Server kết nối</i>	42
3.4.2	<i>Lệnh điều khiển thiết bị</i>	42
3.4.3	<i>Lệnh lấy trạng thái thiết bị</i>	43
CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG		44
4.1	Biểu đồ use case tổng quát	44
4.2	Xây dựng ứng dụng Windows	44
4.2.1	<i>Giao diện ứng dụng</i>	44
4.2.2	<i>Mô tả chức năng phần mềm</i>	46
4.3	Xây dựng phần mềm React Native	48
4.3.1	<i>Giao diện cấu hình wifi cho thiết bị</i>	48
4.3.2	<i>Giao diện đổi tên công tắc</i>	49
4.3.3	<i>Giao diện hẹn giờ thiết bị</i>	50
4.3.4	<i>Giao diện điều khiển thiết bị</i>	51
CHƯƠNG 5: KẾT QUẢ ĐÁNH GIÁ		52
5.1	Hướng dẫn sử dụng sản phẩm	52
5.2	Ưu điểm của hệ thống	56

5.3	Nhược điểm của hệ thống.....	56
KẾT LUẬN.....		57
NHẬN XÉT		58
TÀI LIỆU THAM KHẢO		59

DANH MỤC HÌNH ẢNH

<i>Hình 1. 1 Mô hình hoạt động của IOT.....</i>	<i>3</i>
<i>Hình 1. 2 Công tắc cơ truyền thống.....</i>	<i>4</i>
<i>Hình 1. 3 Công tắc thông minh trên thị trường</i>	<i>5</i>
<i>Hình 1. 4 Đặc trưng của C#</i>	<i>6</i>
<i>Hình 1. 5 Công nghệ Wifi</i>	<i>7</i>
<i>Hình 1. 6 Hoạt động của Wifi</i>	<i>7</i>
<i>Hình 1. 7 React Native</i>	<i>10</i>
<i>Hình 1. 8 React Components</i>	<i>10</i>
<i>Hình 1. 9 JSX</i>	<i>11</i>
<i>Hình 1. 10 Props</i>	<i>11</i>
<i>Hình 1. 11 Giao diện winform</i>	<i>12</i>
<i>Hình 1. 12 Cấu trúc Json</i>	<i>14</i>
<i>Hình 1. 13 Chất lượng đường truyền.....</i>	<i>16</i>
<i>Hình 1. 14 Cấu trúc gói tin giao tiếp.....</i>	<i>17</i>
<i>Hình 1. 15 Bảng tra tên packet</i>	<i>17</i>
<i>Hình 1. 16 Mô hình kiến trúc publish / subscribe.....</i>	<i>18</i>
<i>Hình 1. 17 Bunifu UI.....</i>	<i>21</i>
<i>Hình 1. 18 Bunifu Button</i>	<i>21</i>
<i>Hình 1. 19 Bunifu Card.....</i>	<i>22</i>
<i>Hình 1. 20 Bunifu Checkbox</i>	<i>22</i>
<i>Hình 1. 21 Bunifu Page.....</i>	<i>23</i>
<i>Hình 2. 2 Trang chủ tải Android Studio</i>	<i>24</i>
<i>Hình 2. 3 Chọn thư mục cài đặt Android Studio</i>	<i>25</i>

<i>Hình 2. 4 Chọn Install để cài đặt Android Studio</i>	<i>25</i>
<i>Hình 2. 5 Hoàn thành việc cài đặt Android Studio.....</i>	<i>26</i>
<i>Hình 2. 6 Đã hoàn thành việc cài đặt Android Studio</i>	<i>26</i>
<i>Hình 2. 7 Visual Studio</i>	<i>27</i>
<i>Hình 2. 8 Visual Studio Code</i>	<i>28</i>
<i>Hình 2. 9 Platform IO</i>	<i>29</i>
<i>Hình 2. 10 Tạo Project</i>	<i>29</i>
<i>Hình 2. 11 Điền thông tin project.....</i>	<i>29</i>
<i>Hình 2. 12 Cấu trúc project sau khi tạo xong.....</i>	<i>30</i>
<i>Hình 2. 13 Altium Design</i>	<i>30</i>
<i>Hình 2. 14 Mqtt FX.....</i>	<i>31</i>
<i>Hình 2. 15 Nhập thông tin server</i>	<i>32</i>
<i>Hình 2. 16 Topic subscribe.....</i>	<i>32</i>
<i>Hình 2. 17 Esp8266</i>	<i>33</i>
<i>Hình 2. 18 Sơ đồ chân Esp8266</i>	<i>33</i>
<i>Hình 3. 1 Sơ đồ tổng quan hệ thống</i>	<i>35</i>
<i>Hình 3. 2 Lưu đồ thuật toán phản ứng.....</i>	<i>36</i>
<i>Hình 3. 3 Các khối cơ bản</i>	<i>37</i>
<i>Hình 3. 4 Khối đầu ra</i>	<i>38</i>
<i>Hình 3. 5 Khối đầu vào</i>	<i>39</i>
<i>Hình 3. 6 Khối nguồn.....</i>	<i>39</i>
<i>Hình 3. 7 Khối vi điều khiển</i>	<i>40</i>
<i>Hình 3. 8 Top layout</i>	<i>41</i>
<i>Hình 3. 9 Bottom layout.....</i>	<i>41</i>

<i>Hình 4. 1 Biểu đồ use case</i>	44
<i>Hình 4. 2 Giao diện Windows</i>	45
<i>Hình 4. 3 Giao diện đổi tên phòng</i>	46
<i>Hình 4. 4 Thông tin thời gian</i>	46
<i>Hình 4. 5 Giao diện điều khiển công tắc</i>	46
<i>Hình 4. 6 Hẹn giờ công tắc</i>	47
<i>Hình 4. 7 Thông tin thiết bị phân cứng</i>	47
<i>Hình 4. 8 Giao diện cấu hình wifi cho thiết bị</i>	48
<i>Hình 4. 9 Giao diện đổi tên công tắc</i>	49
<i>Hình 4. 10 Giao diện hẹn giờ thiết bị</i>	50
<i>Hình 4. 11 Giao diện điều khiển thiết bị</i>	51
<i>Hình 5. 1 Cấp điện cho thiết bị phân cứng</i>	52
<i>Hình 5. 2 Cấu hình wifi cho thiết bị</i>	52
<i>Hình 5. 3 Mở ứng dụng trên điện thoại</i>	53
<i>Hình 5. 4 Cấu hình wifi cho thiết bị phân cứng</i>	54
<i>Hình 5. 5 Kiểm tra kết nối Wifi</i>	55
<i>Hình 5. 6 Sử dụng app windows để điều khiển thiết bị</i>	55
<i>Hình 5. 7 Kiểm tra trạng thái hoạt động thiết bị</i>	56

DANH SÁCH BẢNG BIỂU

<i>Bảng 3. 2 Server kết nối MQTT</i>	<i>42</i>
<i>Bảng 3. 3 Lệnh điều khiển thiết bị</i>	<i>42</i>
<i>Bảng 3. 4 Lệnh lấy trạng thái thiết bị</i>	<i>43</i>

DANH MỤC TỪ VIẾT TẮT

IOT: Internet of things

APP: Application

MQTT: Message Queuing Telemetry Transport

IDE: Integrated Development Environment

Json: JavaScript Object Notation

CHƯƠNG 1: TỔNG QUAN VỀ ĐỀ TÀI

1.1 Giới thiệu chung

1.1.1 Giới thiệu chung về IOT

a, Giới thiệu

IoT là khái niệm kết nối các thiết bị với nhau và với Internet. IoT là một mạng lưới khổng lồ gồm các vật (things) và con người được kết nối - tất cả đều thu thập và chia sẻ dữ liệu với nhau. Việc kết nối có thể thực hiện qua Wifi, ZigBee, Bluetooth...

Theo wikipedia, IoT 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.

b, Cơ chế hoạt động



Hình 1. 1 Mô hình hoạt động của IOT

Các thiết bị được kết nối với nhau và cùng kết nối trong cùng một mạng. Con người có thể giám sát, điều khiển thiết bị, thu thập dữ liệu ở bất cứ nơi nào và bất cứ thời điểm nào thông qua IOT Platform.

Một IOT Platform cũng thường được coi là phần mềm trung gian IOT, trong đó nhấn mạnh vai trò chức năng của nó như là một trung gian (middleware) giữa phần cứng và các ứng dụng. Các IOT Platform có thể được tích hợp với hầu hết các thiết bị kết nối và các ứng dụng mà thiết bị sử dụng.

Các cảm biến, các thiết bị đo,... gửi dữ liệu thu thập được lên IoT Platform. Sau đó IoT platform sẽ thực hiện tích hợp dữ liệu nhận được và phân tích để chia sẻ thông tin với các ứng dụng để giải quyết vấn đề.

c, Một số ứng dụng của IOT

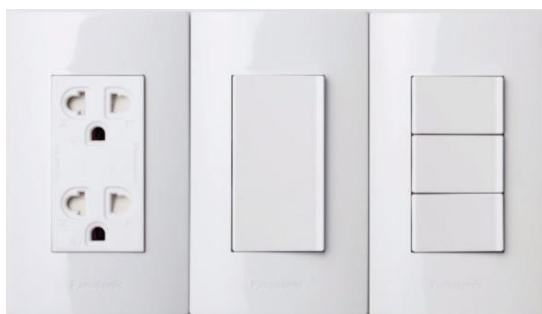
IOT có rất nhiều ứng dụng trong các lĩnh vực đời sống khác nhau. Có thể kể ra bao gồm từ lò vi sóng thông minh, tự động nấu thức ăn trong một khoảng thời gian thích hợp, cho đến những chiếc xe tự lái, có cảm biến phức tạp phát hiện vật thể trên đường đi, đến các thiết bị thể dục có thể đo được nhịp tim của con người và số bước đi đã thực hiện ngày hôm đó, sau đó sử dụng thông tin đó để đề xuất các kế hoạch tập thể dục phù hợp.

d, Kết luận

Hãy tưởng tượng rằng trong thời đại mà IOT bùng nổ, thậm chí có thể điều khiển mọi đồ đạc trong nhà mình chỉ bằng cử chỉ, ánh mắt, thậm chí suy nghĩ. Xe tự lái, bò nuôi trong chuồng không cần đến tận nơi kiểm tra mà cũng biết sức khỏe của nó, biết khi nào nó sẵn sàng cho sữa, cây cần nước sẽ tự tưới, ... Lúc ấy chúng ta sẽ có nhiều thời gian hơn để chăm sóc sức khỏe, nghiên cứu những vấn đề phức tạp hơn.

1.1.2 Công tắc thông minh

Như chúng ta đã biết, hiện nay đại đa số các loại công tắc điện ở nhà. Điều đang dùng loại công tắc cơ đời cũ của các hãng như pana, sino, vanlock,..



Hình 1. 2 Công tắc cơ truyền thống

Đặc điểm loại công tắc này là chúng ta chỉ có thể bật tắt bằng tay và muốn thao tác thì chỉ có cách là lại gần công tắc và bật.

Ngày này, Dù có ở khoảng cách xa nửa vòng trái đất đi chăng nữa. Chúng ta đã có thể nói chuyện và nhìn thấy nhau. Thông qua các thiết bị thông minh như máy tính,...và đặc biệt là điện thoại thông minh.

Vậy với chiếc điện thoại thông minh trên tay, chúng ta có thể bật tắt đèn từ xa khi đã ra khỏi nhà được không?

Chắc chắn là có rồi. Chúng ta sẽ phải sử dụng công tắc thông minh thế hệ mới để thay thế các loại công tắc đời cũ.



Hình 1. 3 Công tắc thông minh trên thị trường

Vì vậy em đã chọn đề tài này để tạo ra một thiết bị giúp người dùng có thể điều khiển bật, tắt tại chỗ hay từ xa thông qua internet giúp tăng độ tiện nghi khi sử dụng các thiết bị điện

1.2 Tổng quan các công nghệ sử dụng

1.2.1 Giới thiệu ngôn ngữ C#

a, Lịch sử phát triển

C# (đọc là "C thăng" hay "C sharp" ("xi-sáp")) là một ngôn ngữ lập trình hướng đối tượng được phát triển bởi Microsoft, là phần khởi đầu cho kế hoạch .NET của họ. Tên của ngôn ngữ bao gồm ký tự thăng theo Microsoft nhưng theo ECMA là C#, chỉ bao gồm dấu số thường. Microsoft phát triển C# dựa trên C++ và Java. C# được miêu tả là ngôn ngữ có được sự cân bằng giữa C++, Visual Basic, Delphi và Java.

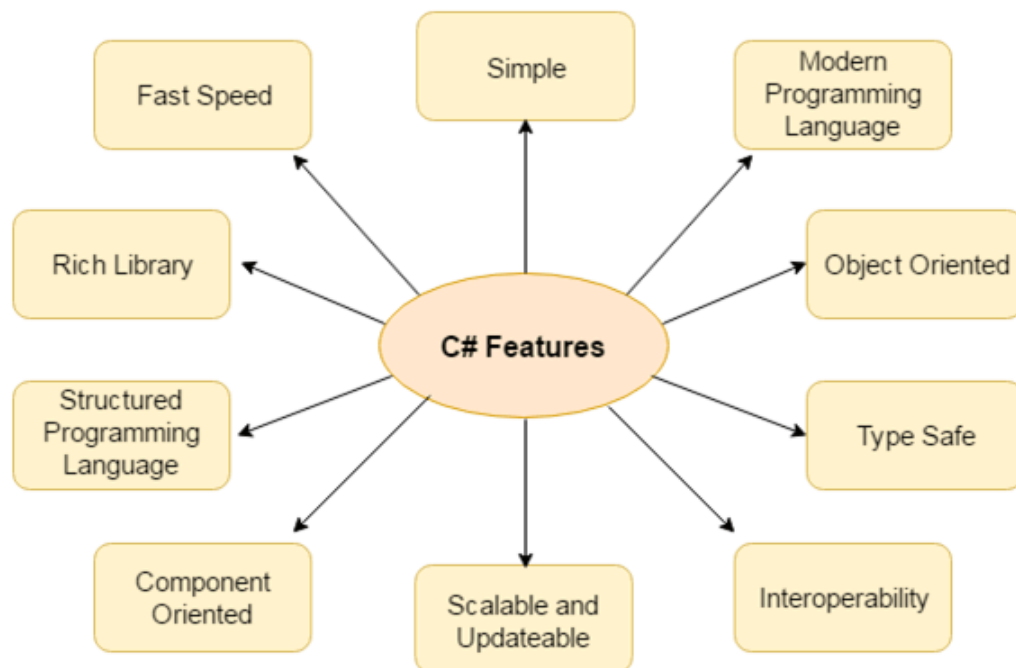
C# được thiết kế chủ yếu bởi Anders Hejlsberg kiến trúc sư phần mềm nổi tiếng với các sản phẩm Turbo Pascal, Delphi, J++, WFC.

C# đã phát triển nhiều kể từ lần phát hành đầu tiên vào năm 2002. C# được giới thiệu với .NET Framework 1.0 và phiên bản hiện tại của C# là 6.0.

b, Các đặc trưng của C#

C# là ngôn ngữ lập trình hướng đối tượng. Nó cung cấp nhiều tính năng :

- Đơn giản(Simple)
- Ngôn ngữ lập trình hiện đại(Modern programming language)
- Lập trình hướng đối tượng(Object oriented)
- Kiểu an toàn(Type safe)
- Khả năng tương tác(Interoperability)
- Có thể mở rộng và cập nhật(Scalable and Updateable)
- Hướng thành phần(Component oriented)
- Ngôn ngữ lập trình có cấu trúc(Structured programming language)
- Nhiều thư viện(Rich Library)
- Tốc độ nhanh(Fast speed)



Hình 1. 4 Đặc trưng của C#

1.2.2 Công nghệ Wifi

a, Khái niệm

Wifi là viết tắt của Wireless Fidelity là hệ thống truy cập internet không dây. Loại sóng vô tuyến này tương tự như sóng điện thoại, truyền hình và radio.



Hình 1. 5 Công nghệ Wifi

b, Nguyên tắc hoạt động của Wifi

Để tạo được kết nối Wifi nhất thiết phải có Router (bộ thu phát), Router này lấy thông tin từ mạng Internet qua kết nối hữu tuyến rồi chuyển nó sang tín hiệu vô tuyến và gửi đi, bộ chuyển tín hiệu không dây (adapter) trên các thiết bị di động thu nhận tín hiệu này rồi giải mã nó sang những dữ liệu cần thiết.

Quá trình này có thể thực hiện ngược lại, Router nhận tín hiệu vô tuyến từ Adapter và giải mã chúng rồi gửi qua Internet.



Hình 1. 6 Hoạt động của Wifi

c, Wifi trong IOT

Xu hướng hỗ trợ đa dạng kết nối dưới một nền tảng chung không giới hạn ở băng thông rộng không dây. Khi Internet of Things (IoT) bắt nhip, sẽ có nhu cầu lớn đối với các kết nối

giữa máy với máy (M2M), đa phần là không dây. Chúng sẽ có nhiều yêu cầu về hiệu năng hơn, sử dụng nhiều trong các trường hợp khác nhau dưới sự bảo trợ của IoT.

Không có công nghệ đơn lẻ nào giải quyết tất cả các yêu cầu thế nên sẽ có một danh sách dài các giao thức IoT không dây. Theo thời gian thì có thể củng cố hơn nữa, nhưng chắc chắn sẽ cần ít nhất một công nghệ mở, được tiêu chuẩn hóa cho một số cấu hình IoT chính. Những cấu hình này thay đổi theo mức độ mà chúng hỗ trợ:

- Công suất cực thấp với công suất vừa phải
- Tầm xa với phạm vi địa phương với phạm vi rất ngắn
- Tốc độ dữ liệu thấp với tốc độ dữ liệu vừa phải
- Độ trễ cực thấp với độ trễ thấp
- Phổ không có giấy phép với phổ được cấp phép

d, Hạn chế Wifi

Bên cạnh những ưu điểm, mạng Wifi cũng tồn tại nhiều nhược điểm chưa thể khắc phục như: phạm vi kết nối của mạng Wifi tới thiết bị có giới hạn, đi càng xa router kết nối càng yếu dần đi. Giải pháp cho vấn đề này là trang bị thêm các Repeater hoặc Access point. Tuy nhiên, gặp nhiều khó khăn do giá thành cao. Nhược điểm tiếp theo của mạng Wifi là về vấn đề băng thông, càng nhiều người kết nối vào mạng thì tốc độ truy cập giảm rõ rệt.

1.2.3 Giới thiệu tổng quan ngôn ngữ C/C++

a, Giới thiệu

Ngôn ngữ lập trình C là một ngôn ngữ mệnh lệnh được phát triển từ đầu thập niên 1970 bởi Dennis Ritchie để dùng trong hệ điều hành UNIX. Từ đó, ngôn ngữ này đã lan rộng ra nhiều hệ điều hành khác và trở thành một những ngôn ngữ phổ dụng nhất. C là ngôn ngữ rất có hiệu quả và được ưa chuộng nhất để viết các phần mềm hệ thống, mặc dù nó cũng được dùng cho việc viết các ứng dụng. Ngoài ra, C cũng thường được dùng làm phương tiện giảng dạy trong khoa học máy tính mặc dù ngôn ngữ này không được thiết kế dành cho người nhập môn.

C là một ngôn ngữ lập trình tương đối nhỏ gọn vận hành gần với phần cứng và nó giống với ngôn ngữ Assembler hơn hầu hết các ngôn ngữ bậc cao. Hơn thế, C đôi khi được đánh giá như là "có khả năng di động", cho thấy sự khác nhau quan trọng giữa nó với ngôn ngữ bậc thấp như là Assembler, đó là việc mã C có thể được dịch và thi hành trong hầu hết các máy

tính, hơn hẳn các ngôn ngữ hiện tại trong khi đó thì Assembler chỉ có thể chạy trong một số máy tính đặc biệt. Vì lý do này C được xem là ngôn ngữ bậc trung.

C++ (đọc là "*C cộng cộng*" hay "*xi-plus-plus*", IPA: /si: pləs pləs/) là một loại ngôn ngữ lập trình. Đây là một dạng ngôn ngữ đa mẫu hình tự do có kiểu tĩnh và hỗ trợ lập trình thủ tục, dữ liệu trừu tượng, lập trình hướng đối tượng, và lập trình đa hình. Từ thập niên 1990, C++ đã trở thành một trong những ngôn ngữ thương mại phổ biến nhất trong khi đó.

b, Một số đặc điểm cơ bản

- Tính cô đọng (compact): C chỉ có 32 từ khóa chuẩn và 40 toán tử chuẩn, nhưng hầu hết đều được biểu diễn bằng những chuỗi ký tự ngắn gọn. 2
- Tính cấu trúc (structured): C có một tập hợp những chỉ thị của lập trình như cấu trúc lựa chọn, lặp... Từ đó các chương trình viết bằng C được tổ chức rõ ràng, dễ hiểu.
- Tính tương thích (compatible): C có bộ tiền xử lý và một thư viện chuẩn vô cùng phong phú nên khi chuyển từ máy tính này sang máy tính khác các chương trình viết bằng C vẫn hoàn toàn tương thích.
- Tính linh động (flexible): C là một ngôn ngữ rất uyển chuyển và cú pháp, chấp nhận nhiều cách thể hiện, có thể thu gọn kích thước của các mã lệnh làm chương trình chạy nhanh hơn.
- Biên dịch (compile): C cho phép biên dịch nhiều tập tin chương trình riêng rẽ thành các tập tin đối tượng (object) và liên kết (link) các đối tượng đó lại với nhau thành một chương trình có thể thực thi được (executable) thống nhất.

1.2.4 Giới thiệu tổng quan về React Native

a, Khái niệm

Được phát triển bởi Facebook, React Native là một framework hướng đến phát triển ứng dụng di động đa nền tảng. Với sự trợ giúp của React Native, lập trình viên (developer) có thể sử dụng JavaScript để tạo ra mobile apps (ứng dụng di động) hỗ trợ cho cả nền tảng Android và iOS. Instagram, Facebook, Skype,... là những ứng dụng nổi bật sử dụng React Native.



Hình 1. 7 React Native

b, Hoạt động

React Native hoạt động được nhờ tích hợp 2 thread là Main Thread và JS Thread cho ứng dụng mobile. Trong đó:

- Main Thread: cập nhật giao diện người dùng(UI) và xử lý tương tác người dùng.
- JS Thread: thực thi và xử lý code Javascript.

Hai luồng Main Thread và JS Thread này hoạt động độc lập với nhau. Hai Thread sẽ tương tác với nhau nhờ một Bridge(cầu nối). Chiếc cầu này sẽ chuyển đổi dữ liệu qua lại giữa các Thread.

c, Các thành phần trong React Native

React Components:

```
import React from 'react';
import { Text } from 'react-native'
;
export default function Name() {
  return (
    <Text>Hello, My name is Tri</Text>
  );
}
```

Hình 1. 8 React Components

JSX:

```
import React from 'react';
import { Text } from 'react-native';

export default function Name() {
  const name = "Chó";
  return (
    <Text>Hello, I am {name}!</Text>
  );
}
```

Hình 1. 9 JSX

Props:

```
import React from 'react';
import { Text, View } from 'react-native';

function Cat(props) {
  return (
    <View>
      <Text>Hello, I am {props.name}!</Text>
    </View>
  );
}

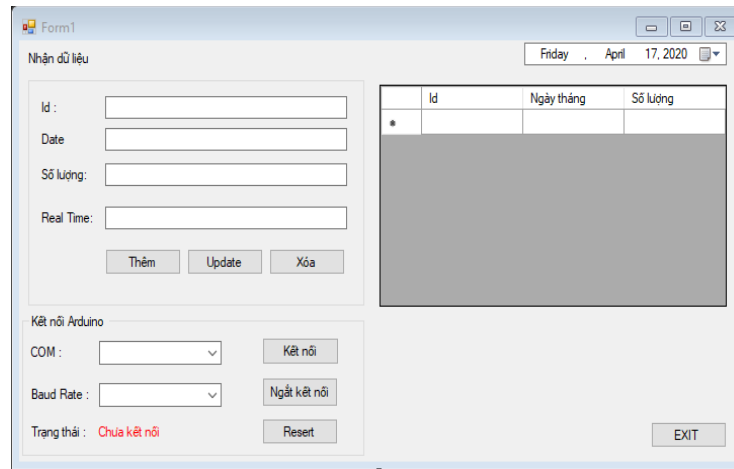
export default function Cafe() {
  return (
    <View>
      <Cat name="Maru" />
      <Cat name="Jellylorum" />
      <Cat name="Spot" />
    </View>
  );
}
```

Hình 1. 10 Props

1.2.5 Giới thiệu tổng quan về công nghệ Winform Application

a, Khái niệm

Windows Forms là một thư viện lớp đồ họa mã nguồn mở và miễn phí được bao gồm như một phần của Microsoft .NET Framework hoặc Mono Framework, cung cấp một nền tảng để viết các ứng dụng khách phong phú cho máy tính để bàn, máy tính xách tay và máy tính bảng.



Hình 1. 11 Giao diện winform

b, Thành phần cấu thành Winform

a, Label

Thuộc tính:

- Name: tên của nhãn(ảnh hưởng trực tiếp tới lập trình)
- Text : Ký tự hiển thị cho người sử dụng(Thường đặt giống với chức năng)
- Backcolor: Màu nền của nhãn

Event:

- Click: Ấn chuột vào đối tượng
- Mouse move: Di chuột vào đối tượng

b, Text Box

Thuộc tính:

- Name: tên của textbox

- Text: ký tự nằm trong textbox
- Multiline: cho phép nhập nhiều dòng
- Enable: cho phép/không cho phép nhập ký tự, tương ứng với 2 lựa chọn true/false
- Backcolor: màu nền của textbox

Event:

- Mouse: khi người dùng nhấn chuột vào textbox
- Click: khi người dùng ấn chuột vào textbox
- Mouse move: khi người dùng di chuột vào textbox

c, Button

Thuộc tính:

- Name: tên của nút bấm
- Backcolor: màu nền
- Background image: ảnh nền của button
- Font: kiểu chữ
- Size: kích cỡ
- Text: kiểu chữ trình bày trên button
- Visible: Ẩn/ hiện chữ trên button
- FlatStyle: Kiểu đường viền của điều khiển Button.

Event:

- Click: khi người dùng ấn chuột vào đối tượng

d, Form

Thuộc tính:

- Name: tên của form
- Backcolor: màu nền của form
- Background: ảnh nền của form
- Language: ngôn ngữ trên form
- Icon: biểu tượng

- Showicon: cho phép hiển thị biểu tượng trên form
- Text: tên hiển thị

Event:

- Load: Nạp hay thực thi khi chạy form

1.2.6 JavaScript Object Notation

a, Json là gì?

JSON là viết tắt của **JavaScript Object Notation**, là một kiểu định dạng dữ liệu tuân theo một quy luật nhất định mà hầu hết các ngôn ngữ lập trình hiện nay đều có thể đọc được. JSON là một tiêu chuẩn mở để trao đổi dữ liệu trên web.

XML

```
<empinfo>
  <employees>
    <employee>
      <name>James Kirk</name>
      <age>40</age>
    </employee>
    <employee>
      <name>Jean-Luc Picard</name>
      <age>45</age>
    </employee>
    <employee>
      <name>Wesley Crusher</name>
      <age>27</age>
    </employee>
  </employees>
</empinfo>
```

JSON

```
{ "empinfo" :
  {
    "employees" : [
      {
        "name" : "James Kirk",
        "age" : 40,
      },
      {
        "name" : "Jean-Luc Picard",
        "age" : 45,
      },
      {
        "name" : "Wesley Crusher",
        "age" : 27,
      }
    ]
  }
}
```

Hình 1. 12 Cấu trúc Json

b, Định nghĩa

Định dạng JSON sử dụng các cặp **key – value** để dữ liệu sử dụng. Nó hỗ trợ các cấu trúc dữ liệu như đối tượng và mảng.

Ta có thể thấy cú pháp của JSON có 2 phần đó là **key** và **value**

- Chuỗi JSON được bao lại bởi dấu ngoặc nhọn { }
- Các **key**, **value** của JSON bắt buộc phải đặt trong dấu nháy kép {“”}, nếu bạn đặt nó trong dấu nháy đơn thì đây không phải là một chuỗi JSON đúng chuẩn. Nếu trường

hợp trong value của bạn có chứa dấu nháy kép " thì hãy dùng dấu \ để đặt trước nó, ví dụ "json là gì".

- Nếu có nhiều dữ liệu thì dùng dấu phẩy , để ngăn cách.
- Các key của JSON bạn nên đặt chữ cái không dấu hoặc số, dấu _ và không có khoảng trắng., ký tự đầu tiên không nên đặt là số.

c, Cấu trúc chuỗi Json

Object trong Json được thể hiện bằng dấu ngoặc nhọn { }. Khái niệm Object trong Json cũng khá tương đồng với Object trong Javascript. Tuy nhiên, Object trong Json vẫn có những giới hạn như:

Key: phải luôn nằm trong dấu ngoặc kép, không được phép là biến số.

Value: Chỉ cho phép các kiểu dữ liệu cơ bản: numbers, String, Booleans, arrays, objects, null. Không cho phép function, date, undefined.

Không cho phép dấu phẩy cuối cùng như Object trong Javascript.

1.3 Giao thức MQTT

1.3.1 Các khái niệm cơ bản

a, Khái niệm

MQTT (Message Queuing Telemetry Transport) là dựa trên giao thức tin nhắn Publish-Subscribe. Sử dụng TCP/IP để truyền dữ liệu. Giao thức được sử dụng phổ biến trong mạng lưới IOT với ưu điểm code đơn giản, gói tin nhẹ sử dụng băng thông nhỏ để truyền dữ liệu thích hợp trong mạng truyền bị giới hạn.

b, Broker

Là server cài MQTT server thu thập dữ liệu và giao tiếp các client.

c, Topic

Là chủ đề mà Broker tạo ra để Device gửi vào. Nó như là folder chứa dữ liệu vậy nên có thể ở dạng đường dẫn abc/topic1.

d, Publish

Là gói tin Device gửi lên Broker. Khi Device Publish dữ liệu vào topic, các Client sẽ nhận được dữ liệu khi đăng kí topic. Gói Publish có thể là gói gửi dữ liệu gửi vào topic hoặc gói cài đặt Retain Message hay LWT Message do cờ Retain, LWT quy định.

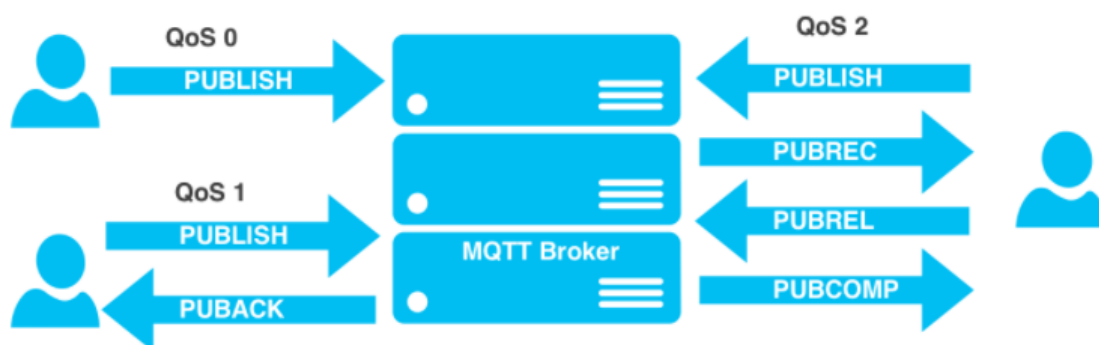
e, Subscribe

Là gói đăng kí nhận data hay thông tin từ topic đăng kí(Broker sẽ tạo một topic mới nếu không tìm được topic đăng kí). Khi Client gửi gói tin Subscribe tới Broker để đăng kí một topic, ví dụ topic “T2Tdemo”. Bất cứ khi nào Device gửi gói tin Publish vào topic “T2Tdemo”. Dữ liệu trong gói Publish chuyển về Client. Giống như bạn Subscribe youtube.

f, QoS

Chất lượng đường truyền

- Qos 0: Device gửi Publish tới Server không cần quan tâm đến gói tin gửi.
- Qos 1: Device gửi Publish tới Server ,Sau đó nếu Server nhận được gói tin và gửi lại Device gói PUBACK để xác nhận đã nhận được gói Publish từ Device
- Qos 2: Device gửi Publish tới Server , Server nhận gói Publish gửi PUBREC lại Device kèm theo ID đã nhận. Device nhận PUBREC gửi PUBREL kèm ID đó lại Server. Server gửi lại PUBCOMP lại Device.



Hình 1. 13 Chất lượng đường truyền

g, Retain

Là cờ báo gói Publish, Device gửi tới Broker cài đặt Retain Message cho topic. Khi Client subscribe vào topic đã cài đặt Retain Message sẽ nhận được ngay Retain Message. Ví dụ “Device online”.

h, LWT(Last Will and Testament)

Là cờ báo gói Publish, Device gửi tới Broker topic cài đặt LWT Message. Khi Client Subscribe vào topic đã cài đặt LWT Message sẽ nhận được tin nhắn LWT Message khi Device offline

1.3.2 Packet

a, Cấu trúc gói tin giao tiếp

bit	7	6	5	4	3	2	1	0
byte 1	Message Type				DUP flag	QoS level		RETAIN
byte 2	Remaining Length							

Hình 1. 14 Cấu trúc gói tin giao tiếp

Có thể có nhiều byte được sử dụng sau Remaining Length

Byte 1: Bits 7-4 Là byte nhận biết đây là packet tên là gì. Dựa vào bảng dưới ta có thể thấy rằng có tất cả 14 loại packet khác nhau.

Mnemonic	Enumeration	Description
Reserved	0	Reserved
CONNECT	1	Client request to connect to Server
CONNACK	2	Connect Acknowledgment
PUBLISH	3	Publish message
PUBACK	4	Publish Acknowledgment
PUBREC	5	Publish Received (assured delivery part 1)
PUBREL	6	Publish Release (assured delivery part 2)
PUBCOMP	7	Publish Complete (assured delivery part 3)
SUBSCRIBE	8	Client Subscribe request
SUBACK	9	Subscribe Acknowledgment
UNSUBSCRIBE	10	Client Unsubscribe request
UNSUBACK	11	Unsubscribe Acknowledgment
PINGREQ	12	PING Request
PINGRESP	13	PING Response
DISCONNECT	14	Client is Disconnecting
Reserved	15	Reserved

Hình 1. 15 Bảng tra tên packet

Bits 0-3 là các flag Qos, Retain mình đã giới thiệu ở topic giới thiệu chung. Dup flag bật khi Client hoặc Server gửi lại một trong các message sau

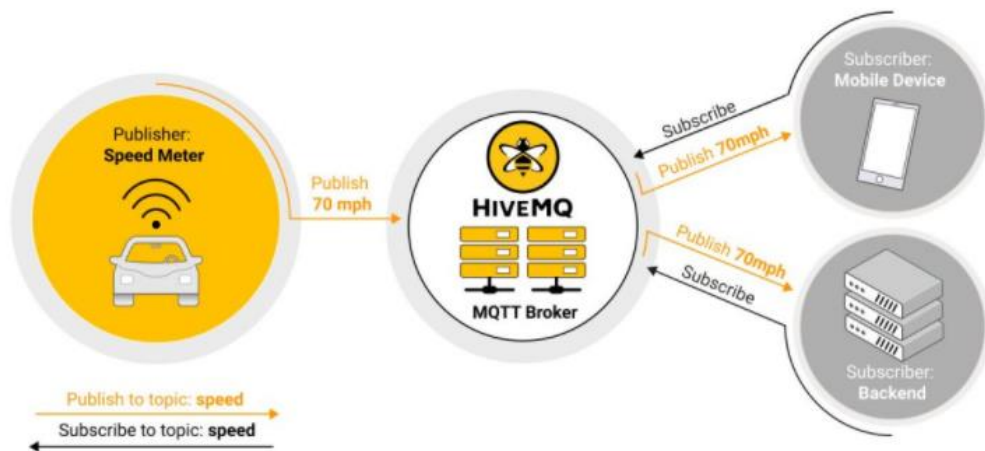
Byte 2: Remain Length: Số byte trong packet. Độ dài này cần được encode chuẩn UTF-8

Encode UTF-8: Trong 1byte: 7 bit đầu dùng để tính kích thước, bit thứ 8 sẽ báo byte có được sử dụng hay không.

1.3.3 MQTT Publish & Subscribe

a, Mô hình kiến trúc publish / subscribe

Mẫu publish/subscribe (còn được gọi là pub/sub) cung cấp một thay thế cho kiến trúc máy khách-máy chủ truyền thống. Trong mô hình máy khách-máy khách, máy khách liên lạc trực tiếp với điểm cuối. Mô hình pub / sub tách riêng máy khách gửi tin nhắn (nhà xuất bản-publish) từ máy khách hoặc máy khách nhận tin nhắn (người đăng ký-subscribe) . Các nhà xuất bản và đăng ký không bao giờ liên lạc trực tiếp với nhau. Trong thực tế, họ thậm chí không nhận thức được rằng cái kia tồn tại. Kết nối giữa chúng được xử lý bởi một thành phần thứ ba (người môi giới-broker) . Công việc của nhà môi giới là lọc tất cả các tin nhắn đến và phân phối chúng một cách chính xác cho người đăng ký. Vì vậy, hãy đi sâu hơn một chút vào một số khía cạnh chung của pub/sub



Hình 1. 16 Mô hình kiến trúc publish / subscribe

Khía cạnh quan trọng nhất của pub / sub là sự tách rời của nhà xuất bản tin nhắn từ người nhận (người đăng ký). Sự tách rời này có một số chiều:

- *Phân tách không gian:* Nhà xuất bản và người đăng ký không cần biết nhau (ví dụ: không trao đổi địa chỉ IP và cổng).
- *Thời gian tách rời:* Nhà xuất bản và người đăng ký không cần phải chạy cùng một lúc.
- *Phân tách đồng bộ hóa:* Hoạt động trên cả hai thành phần không cần phải bị gián đoạn trong quá trình xuất bản hoặc nhận.

b, Khả năng mở rộng

Pub / Sub quy mô tốt hơn so với cách tiếp cận máy khách-máy chủ truyền thống. Điều này là do các hoạt động trên môi giới có thể được song song hóa cao và các thông điệp có thể được xử lý theo cách hướng sự kiện. Bộ nhớ đệm tin nhắn và định tuyến thông minh của tin nhắn thường là một yếu tố quyết định để cải thiện khả năng mở rộng. Tuy nhiên, mở rộng lên đến hàng triệu kết nối là một thách thức. Mức độ kết nối cao như vậy có thể đạt được với các nút môi giới được phân cụm để phân phối tải trên nhiều máy chủ riêng lẻ hơn bằng cách sử dụng bộ cân bằng tải.

1.3.4 Ưu điểm, nhược điểm của MQTT và một số ứng dụng triển khai

a, Ưu điểm

MQTT cho phép hệ thống SCADA của bạn truy cập dữ liệu IoT. MQTT mang lại nhiều ưu điểm vượt trội cho hệ thống của bạn:

- Phân phối thông tin hiệu quả hơn
- Tăng khả năng mở rộng
- Giảm đáng kể tiêu thụ băng thông mạng
- Giảm tốc độ cập nhật xuống giây
- Rất phù hợp cho điều khiển và viễn thám
- Tối đa hóa băng thông đang sử dụng
- Chi phí đầu tư cực kỳ thấp
- Rất an toàn (bảo mật dựa trên sự cấp phép)
- Đã được sử dụng bởi các ngành công nghiệp dầu khí, Amazon, Facebook và các doanh nghiệp lớn khác
- Giảm thiểu thời gian phát triển
- Giao thức publish/subscribe thu thập nhiều dữ liệu hơn với ít băng thông hơn so với giao thức cũ.

b, Nhược điểm

- MQTT có chu kỳ truyền chậm hơn so với CoAP.
- Tài nguyên của MQTT hoạt động dựa trên subscribe động, trong khi CoAP sử dụng hệ thống tài nguyên tĩnh – ổn định.
- MQTT không được mã hóa. Thay vào đó, nó sử dụng TLS / SSL để mã hóa bảo mật.
- Rất khó để tạo ra một mạng MQTT có thể mở rộng toàn cầu.

c, Ứng dụng của MQTT trong các ứng dụng lớn

- Facebook Messenger: Facebook đã sử dụng các khía cạnh của MQTT trong Facebook Messenger để trò chuyện trực tuyến . Tuy nhiên, không rõ MQTT được sử dụng bao nhiêu hoặc để làm gì.
- IECC Scalable: DeltaRail phiên bản mới nhất của hệ thống kiểm soát hiệu IECC của họ đã sử dụng MQTT cho thông tin liên lạc trong các phần khác nhau của hệ thống và các thành phần khác của hệ thống báo hiệu. Nó cung cấp khung truyền thông cơ bản cho một hệ thống tuân thủ các tiêu chuẩn CENELEC cho các thông tin liên lạc quan trọng về an toàn.
- Amazon Web Services đã công bố Amazon IoT dựa trên MQTT vào năm 2015.
- Các tổ chức không gian địa lý SensorThings API đặc điểm kỹ thuật tiêu chuẩn có một phần mở rộng MQTT trong tiêu chuẩn như một giao thức thông báo bổ sung ràng buộc. Nó đã được chứng minh trong một thí điểm IoT của Bộ An ninh Nội địa Hoa Kỳ.
- Các dịch vụ của Cơ sở hạ tầng thượng nguồn OpenStack được kết nối bằng một bus tin nhắn hợp nhất MQTT với Mosquitto là broker MQTT.
- Adafruit đưa ra một MQTT miễn phí Cloud Service cho thí nghiệm IOT và người học gọi Adafruit IO trong năm 2015.
- Microsoft Azure IoT Hub sử dụng MQTT làm giao thức chính cho các tin nhắn từ xa .
- XIM, Inc. đã ra mắt ứng dụng khách MQTT có tên MQTT Buddy vào năm 2017. Đây là ứng dụng MQTT dành cho Android và iOS , nhưng không phải là F-Droid , người dùng có sẵn bằng tiếng Anh, tiếng Nga và tiếng Trung Quốc.
- Node-RED hỗ trợ các nút MQTT kể từ phiên bản 0.14, để định cấu hình đúng các kết nối TLS.
- Nền tảng tự động hóa phần mềm nguồn mở Home Assistant được bật MQTT và cung cấp bốn tùy chọn cho các broker MQTT.

1.4 Bunifu framework

1.4.1 Giới thiệu

Bunifu framework là một thư viện hỗ trợ việc thiết kế cho windows form đơn giản hơn. Với bunifu framework .NET, thiết kế winform chưa bao giờ dễ như thế.



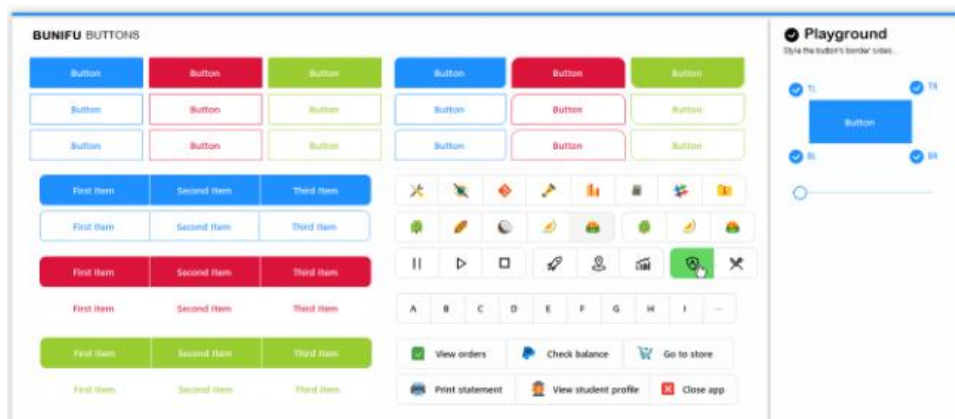
Hình 1. 17 Bunifu UI

Với bunifu bạn có thể làm nhiều thứ dễ dàng hơn như :

- Slide menu hoạt hình (đóng mở)
- Nền winform đẹp hơn với Gradient background - Tô màu cầu vồng cho form (trộn 3 màu)
- Hỗ trợ Flat UI - Thiết kế giao diện phẳng như Metro Design
- Nhiều chức năng khác như progressbar quay tròn,biểu đồ ,rất tiện lợi.

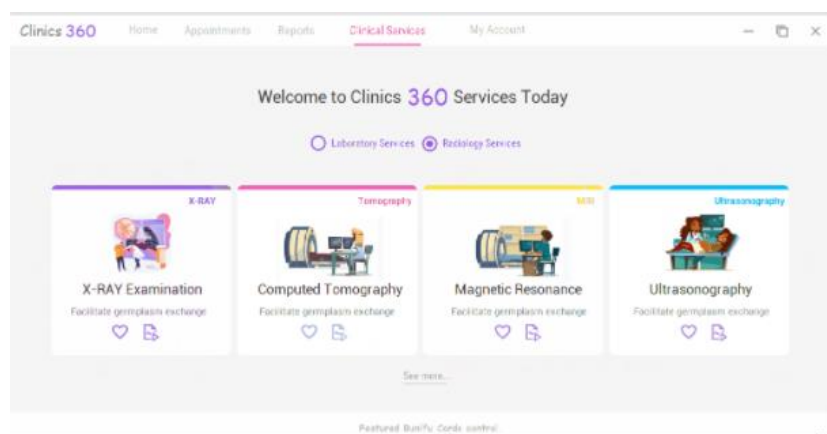
1.4.2 Một số thành phần cơ bản

Button Bunifu thực sự là một trong những loại nút. Có rất nhiều thứ bạn có thể đạt được với một điều khiển này và dễ dàng tùy chỉnh đến mức bạn sẽ tự hỏi mình đã sống như thế nào nếu không có nó ... Với hơn 12 phong cách thiết kế được tích hợp sẵn, tất cả những gì bạn cần là một chút sáng tạo và bạn đã sẵn sàng để tạo ra một cái gì đó đẹp!



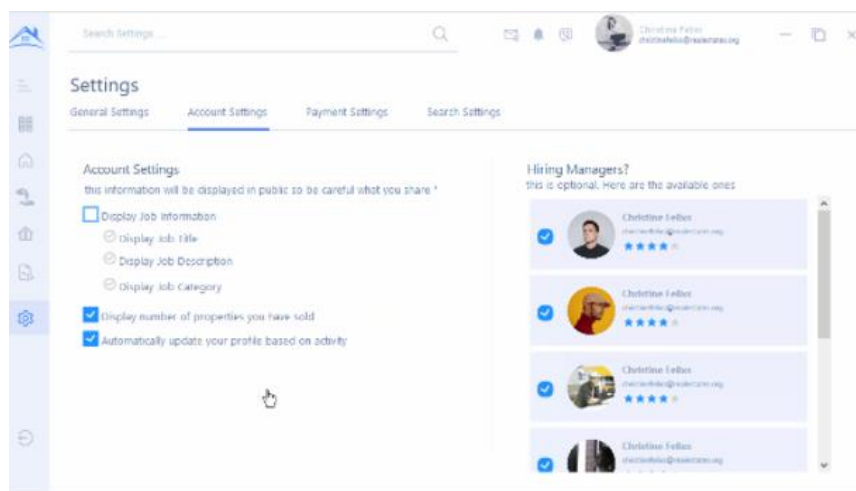
Hình 1. 18 Bunifu Button

Bunifu Cards là một điều khiển được thiết kế để bắt chước một thẻ cơ bản và hữu ích khi sắp xếp thông tin trên trang tổng quan của bạn. Bảng điều khiển sau đã sử dụng thẻ Bunifu để phát triển một bố cục thể hiện sự sắp xếp trang nhã của danh sách các dịch vụ được cung cấp.



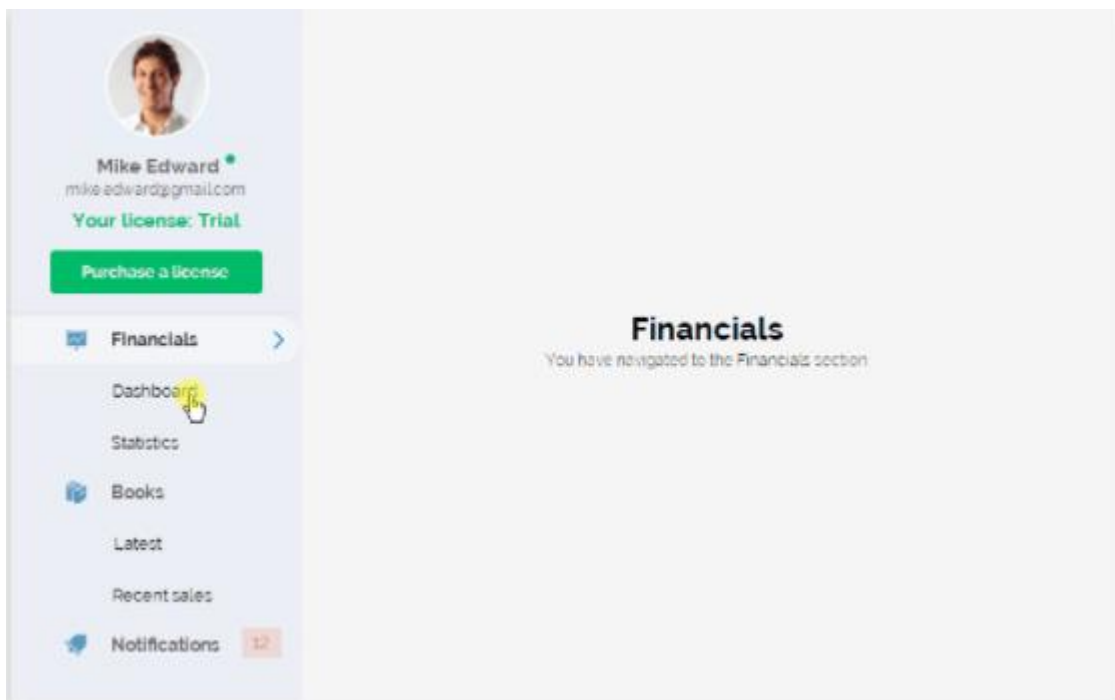
Hình 1. 19 Bunifu Card

Bunifu CheckBox hỗ trợ Tristate: Các trạng thái được kiểm tra, không được kiểm tra và không xác định; Tùy chỉnh giao diện người dùng linh hoạt cho các trạng thái đã chọn và không được chọn.



Hình 1. 20 Bunifu Checkbox

Bunifu Pages là một điều khiển đa vùng chứa trong Windows Forms độc đáo cho phép dễ dàng phát triển giao diện người dùng với nhiều chế độ xem trong cùng một Biểu mẫu, vùng chứa hoặc điều khiển người dùng tùy chỉnh.



Hình 1. 21 Bunifu Page

CHƯƠNG 2: MÔI TRƯỜNG LẬP TRÌNH VÀ CÔNG CỤ THIẾT KẾ

2.1 Android Studio

2.1.1 Giới thiệu

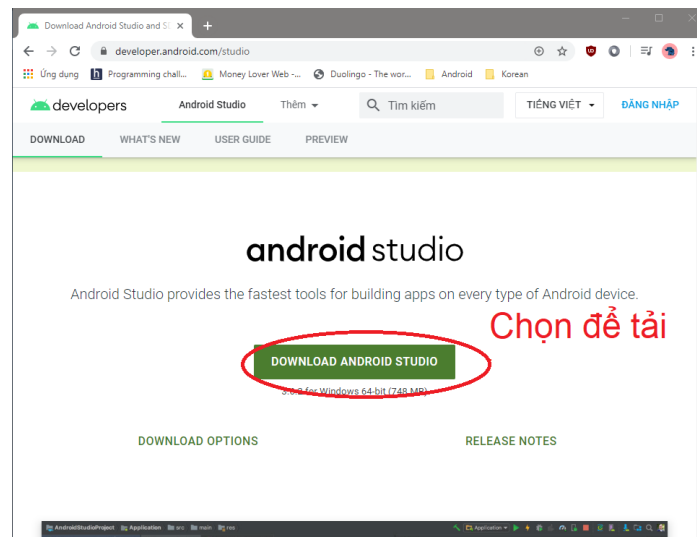
Android Studio là môi trường phát triển tích hợp IDE chính thức dành cho phát triển nền tảng Android.

Nó được ra mắt vào ngày 16 tháng 5 năm 2013, được phát hành miễn phí theo giấy phép Apache Licence 2.0.

Android Studio được phát triển dựa trên phần mềm IntelliJ IDEA của JetBrains, Android Studio được thiết kế đặc biệt để phát triển ứng dụng Android. Nó hỗ trợ các hệ điều hành Windows, Mac OS và Linux. Là IDE chính thức của Google để phát triển ứng dụng Android gốc để thay thế cho Android Development Tools dựa trên Eclipse.

2.1.2 Cài đặt Android Studio

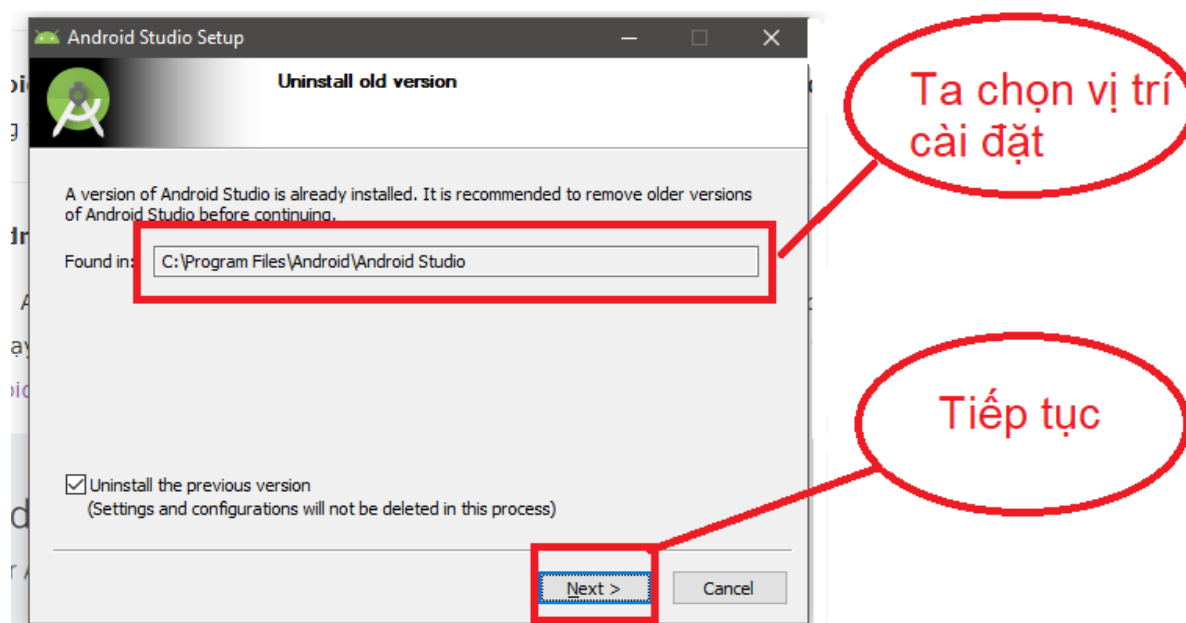
Bước 1: Chúng ta truy cập vào trang chủ của Android Studio để tải bản mới nhất <https://developer.android.com/studio>.



Hình 2. 1 Trang chủ tải Android Studio

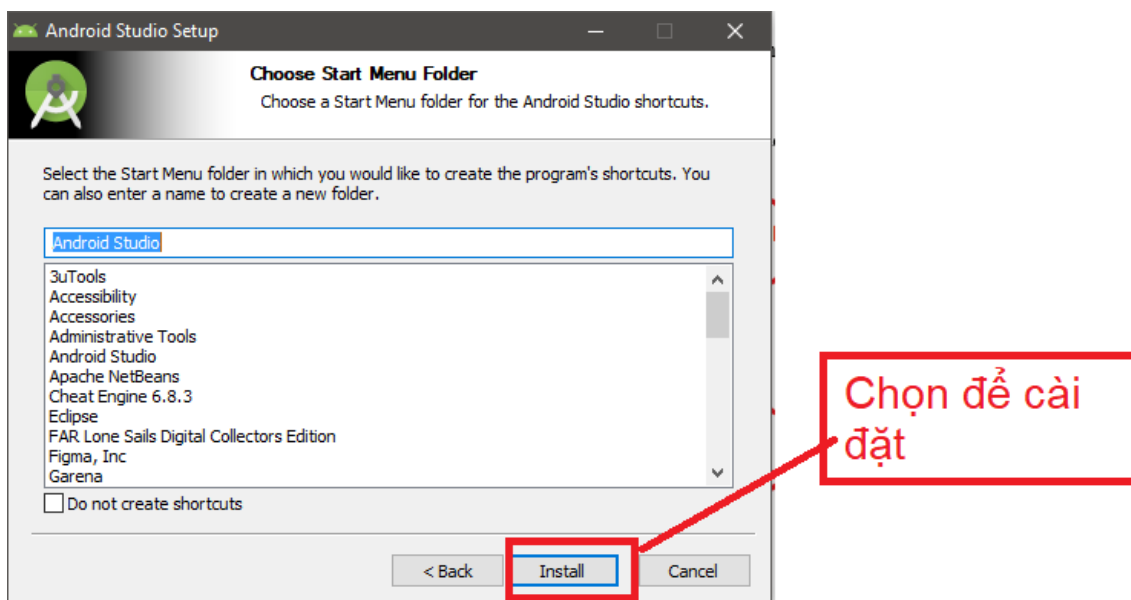
Bước 2: Tiến hành cài đặt trên Windows. Sau khi tải xuống sau tại thực hiện việc chạy file vừa tải xuống.

Ta có thể chọn vị trí lưu cài đặt Android Studio. Sau đó ấn **Next** để tiếp tục.



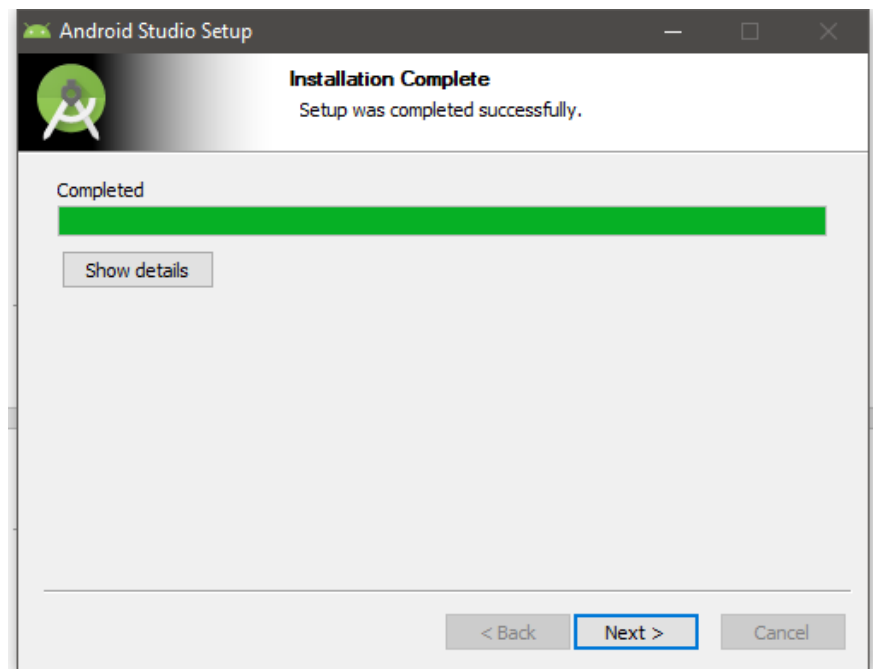
Hình 2. 2 Chọn thư mục cài đặt Android Studio

Bước 3: Sau đó ta chọn Install để cài đặt. Và đợi Android Studio tự cài đặt.

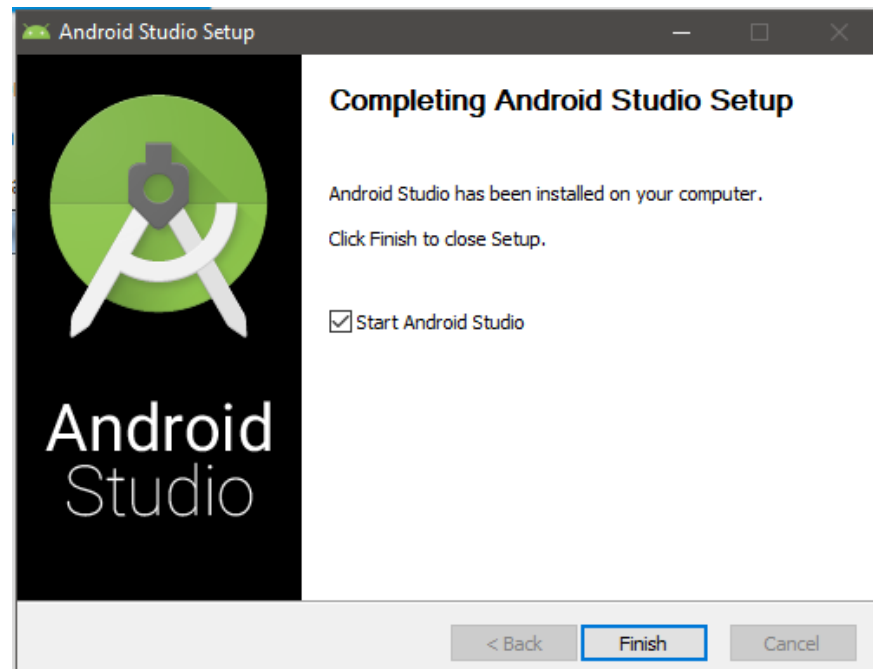


Hình 2. 3 Chọn Install để cài đặt Android Studio

Bước 4: Hoàn thành việc cài đặt. Ta chọn NEXT để hoàn thành khi Android Studio đã cài đặt xong.



Hình 2. 4 Hoàn thành việc cài đặt Android Studio



Hình 2. 5 Đã hoàn thành việc cài đặt Android Studio

2.2 Visual Studio

2.2.1 Visual Studio là gì ?

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

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



Hình 2. 6 Visual Studio

2.2.2 Các tính năng phần mềm trên Visual Studio

- Đa nền tảng
- Đa ngôn ngữ lập trình
- Hỗ trợ website
- Kho tiện ích mở rộng phong phú
- Lưu trữ phân cấp
- Kho lưu trữ an toàn
- Hỗ trợ viết code
- Hỗ trợ thiết bị đầu cuối
- Hỗ trợ Git

2.3 Visual Studio Code và Platform IO

2.3.1 Visual Studio Code

Là một trình biên tập lập trình code miễn phí dành cho Windows, Linux và macOS, Visual Studio Code được phát triển bởi Microsoft. Nó được xem là một sự kết hợp hoàn hảo giữa IDE và Code Editor.

Visual Studio Code hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nhờ tính năng tùy chỉnh, Visual Studio Code cũng cho phép người dùng thay đổi theme, phím tắt, và các tùy chọn khác.



Hình 2. 7 Visual Studio Code

Một số tính năng của Visual Studio Code:

- Hỗ trợ nhiều ngôn ngữ lập trình
- Hỗ trợ đa nền tảng
- Cung cấp kho tiện ích mở rộng
- Kho lưu trữ an toàn
- Hỗ trợ Web, lưu trữ dữ liệu đa dạng phân cấp

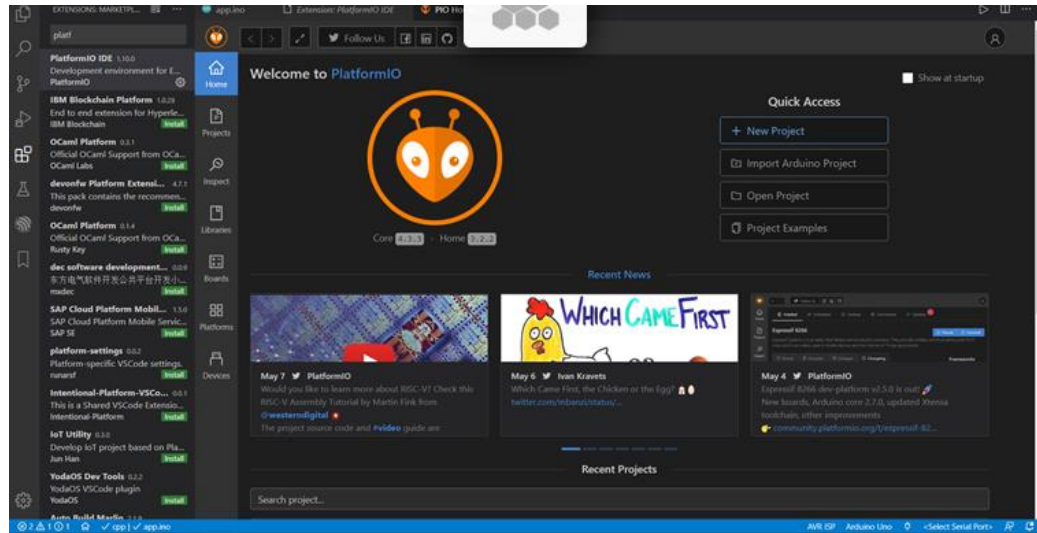
2.3.2 Platform IO

PlatformIO là một plugin có khả năng lập trình Arduino/ ARM mbed nó hỗ trợ tới hơn 250 board khác nhau.



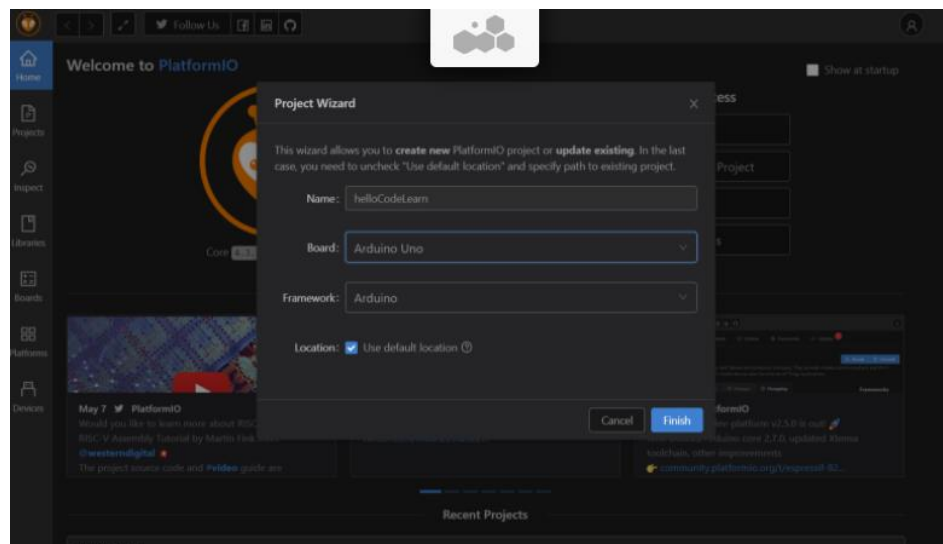
Hình 2. 8 Platform IO

Để bắt đầu với project mới, các bạn chọn New Project:



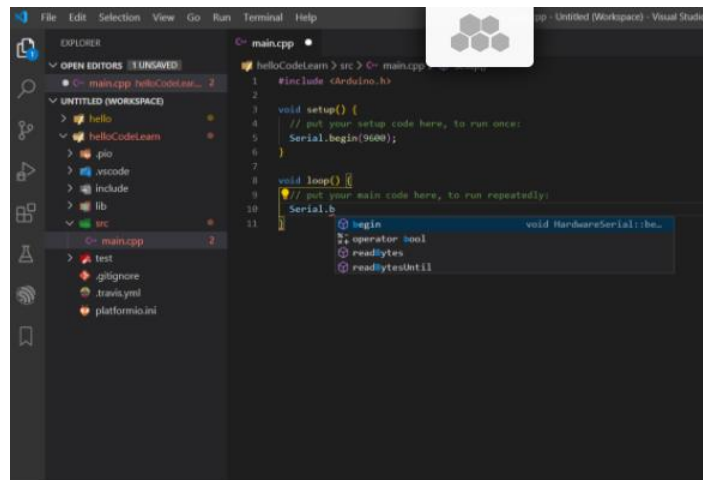
Hình 2. 9 Tạo Project

Điền thông tin vào, nhớ rằng platformio sẽ cấu hình luôn board mạch bạn sử dụng ngay khi khởi tạo project:



Hình 2. 10 Điền thông tin project

Kết quả sau khi tạo project:



Hình 2. 11 Cấu trúc project sau khi tạo xong

2.4 Altium Design

2.4.1 Giới thiệu

Altium Designer trước kia có tên gọi quen thuộc là Protel DXP, là một trong những công cụ vẽ mạch điện tử mạnh nhất hiện nay. Được phát triển bởi hãng Altium Limited. Altium designer là một phần mềm chuyên ngành được sử dụng trong thiết kế mạch điện tử. Nó là một phần mềm mạnh với nhiều tính năng thú vị, tuy nhiên phần mềm này còn được ít người biết đến so với các phần mềm thiết kế mạch khác như orcad hay proteus.



Hình 2. 12 Altium Design

2.4.2 Một số đặc trưng

Giao diện thiết kế, quản lý và chỉnh sửa thân thiện, dễ dàng biên dịch, quản lý file, quản lý phiên bản cho các tài liệu thiết kế.

Hỗ trợ mạnh mẽ cho việc thiết kế tự động, đi dây tự động theo thuật toán tối ưu, phân tích lắp ráp linh kiện. Hỗ trợ việc tìm các giải pháp thiết kế hoặc chỉnh sửa mạch, linh kiện, netlist có sẵn từ trước theo các tham số mới.

Mở, xem và in các file thiết kế mạch dễ dàng với đầy đủ các thông tin linh kiện, netlist, dữ liệu bản vẽ, kích thước, số lượng...

Hệ thống các thư viện linh kiện phong phú, chi tiết và hoàn chỉnh bao gồm tất cả các linh kiện nhúng, số, tương tự...

Đặt và sửa đổi tượng trên các lớp cơ khí, định nghĩa các luật thiết kế, tùy chỉnh các lớp mạch in, chuyển từ schematic sang PCB, đặt vị trí linh kiện trên PCB.

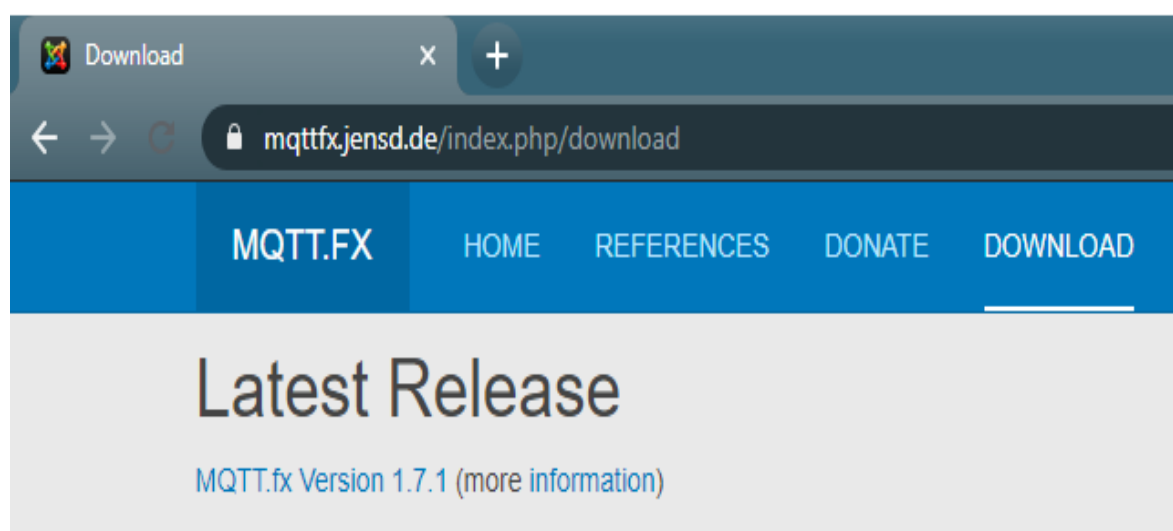
Mô phỏng mạch PCB 3D, đem lại hình ảnh mạch điện trung thực trong không gian 3 chiều, hỗ trợ MCAD-ECAD, liên kết trực tiếp với mô hình STEP, kiểm tra khoảng cách cách điện, cấu hình cho cả 2D và 3D

Hỗ trợ thiết kế PCB sang FPGA và ngược lại.

2.5 MQTT Fx

2.5.1 Giới thiệu

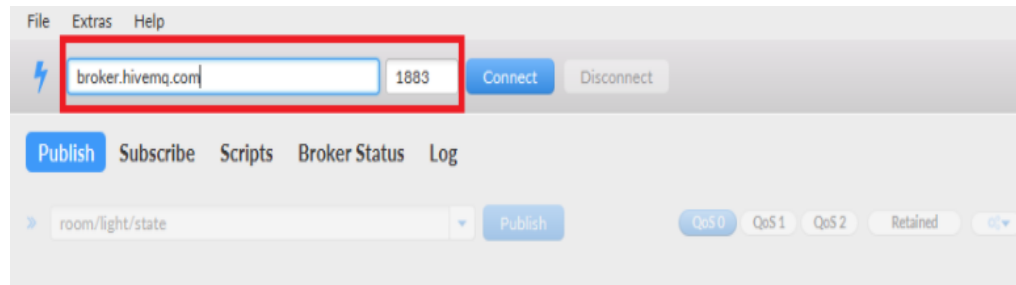
MQTT.fx là một MQTT Client được viết bằng Java dựa trên Eclipse Paho. Chúng ta có thể dễ dàng giao tiếp với các thiết bị IoT thông qua ứng dụng này.



Hình 2. 13 Mqtt FX

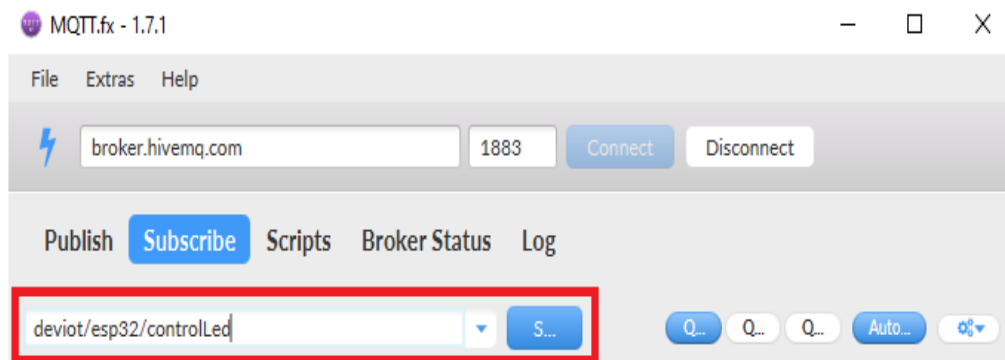
2.5.2 Các bước kết nối

Đầu tiên các bạn nhập tên Broker sử dụng và cổng kết nối sau đó nhấn nút connect ngay bên cạnh.



Hình 2. 14 Nhập thông tin server

Sau khi connect nhấn vào Subscribe chúng ta sẽ nhập tên topic và nhấn vào nút Sub ngay bên cạnh. Như vậy ta đã Subscribe một topic có tên như hình dưới.



Hình 2. 15 Topic subscribe

2.6 Esp8266

2.6.1 Giới thiệu

Mạch thu phát WiFi ESP8266 Uart ESP-01 sử dụng IC Wifi SoC ESP8266 của hãng Espressif, được sử dụng để kết nối với vi điều khiển thực hiện chức năng truyền nhận dữ liệu qua Wifi, mạch có thiết kế nhỏ gọn, sử dụng giao tiếp UART với bộ thư viện và code mẫu rất nhiều từ cộng đồng (search google esp-01), mạch thu phát wifi ESP-01 được sử dụng trong các ứng dụng IoT và điều khiển thiết bị qua Wifi,...

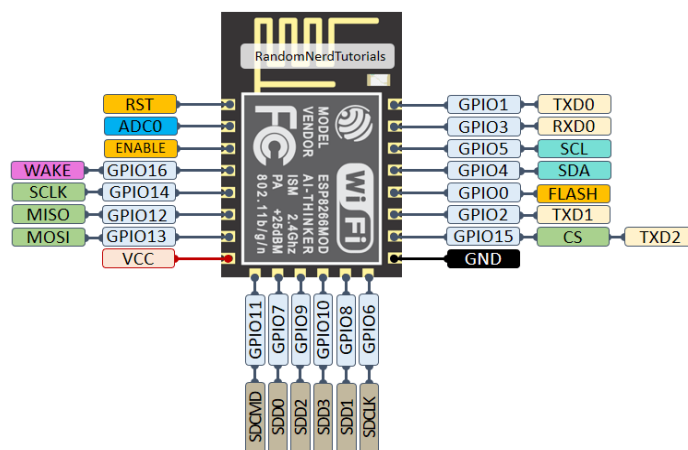


Hình 2. 16 Esp8266

2.6.2 Thông số kỹ thuật

- Điện áp sử dụng: 3.3VDC
- Điện áp giao tiếp: 3.3VDC
- Dòng tiêu thụ: Max 320mA (nên sử dụng module cấp nguồn riêng cho mạch).
- Hỗ trợ chuẩn 802.11 b/g/n.
- Wi-Fi 2.4 GHz, hỗ trợ các chuẩn bảo mật như: OPEN, WEP, WPA_PSK, WPA2_PSK, WPA_WPA2_PSK.
- Hỗ trợ cả 2 giao tiếp TCP và UDP.
- Chuẩn giao tiếp UART với Firmware hỗ trợ bộ tập lệnh AT Command, tốc độ Baudrate mặc định 9600 hoặc 115200.
- Có 3 chế độ hoạt động: Client, Access Point, Both Client and Access Point.
- Kích thước: 24.8 x 14.3mm

2.6.3 Chức năng các I/O



Hình 2. 17 Sơ đồ chân Esp8266

Tất cả các GPIO đều có trở kéo lên nguồn bên trong (ngoại trừ GPIO16 có trở kéo xuống GND). Người dùng có thể cấu hình kích hoạt hoặc không kích hoạt trở kéo này.

GPIO1 và GPIO3: hai GPIO này được nối với TX và RX của bộ UART0, NodeMCU nạp code thông qua bộ UART này nên tránh sử dụng 2 chân GPIO này.

GPIO0, GPIO2, GPIO15: đây là các chân có nhiệm vụ cấu hình mode cho ESP8266 điều khiển quá trình nạp code nên bên trong NodeMCU (có tên gọi là strapping pins) có các trở kéo để định sẵn mức logic cho chúng như sau: GPIO0: HIGH, GPIO2: HIGH, GPIO15: LOW. Vì vậy khi muốn sử dụng các chân này ở vai trò GPIO cần phải thiết kế một nguyên lý riêng để tránh xung đột đến quá trình nạp code.

GPIO9, GPIO10: hai chân này được dùng để giao tiếp với External Flash của ESP8266 vì vậy cũng không thể dùng được.

Như vậy, các GPIO còn lại: GPIO 4, 5, 12, 13, 14, 16 có thể sử dụng bình thường.

CHƯƠNG 3: MÔ TẢ VÀ PHÂN TÍCH THIẾT KẾ HỆ THỐNG

3.1 Mô tả và yêu cầu của hệ thống

Hệ thống có hai ứng dụng trên máy tính windows và điện thoại di động để điều khiển và giám sát thông qua mạng internet.

Hệ thống cần đảm bảo chức năng và giao diện thân thiện với người sử dụng. Xây dựng ứng dụng UI dựa trên nền tảng C# và React Native.

Hiệu năng của app chạy đảm bảo không ảnh hưởng đến hệ thống, hỗ trợ các thiết bị có cấu hình thấp.

Module phần cứng cần nhỏ gọn, có kết nối wifi để nhận dữ liệu từ ứng dụng gửi về.

Module phần cứng có khả năng đóng cắt nguồn điện xoay chiều 220V.

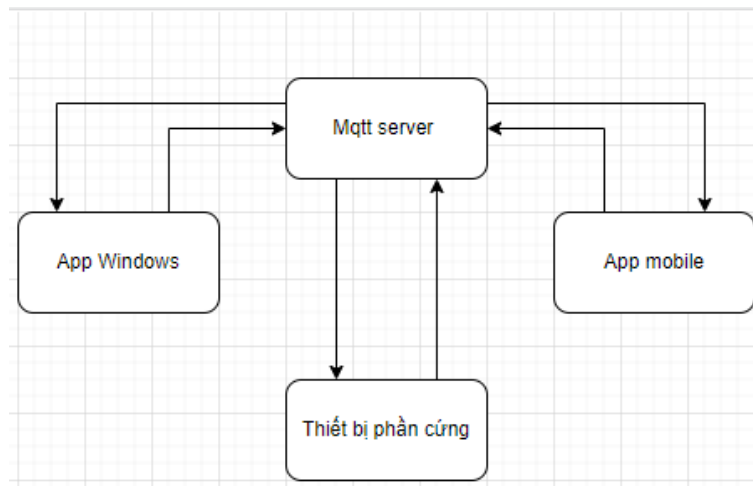
Module có khả năng kết nối wifi thông minh để dễ dàng cho việc di chuyển thiết bị từ nơi này sang nơi khác.

Đồng bộ trạng thái dữ liệu đầu ra với giao diện hiển thị trên ứng dụng.

3.2 Phân tích thiết kế hệ thống

3.2.1 Sơ đồ tổng quan hệ thống

a, Sơ đồ tổng quan



Hình 3. 1 Sơ đồ tổng quan hệ thống

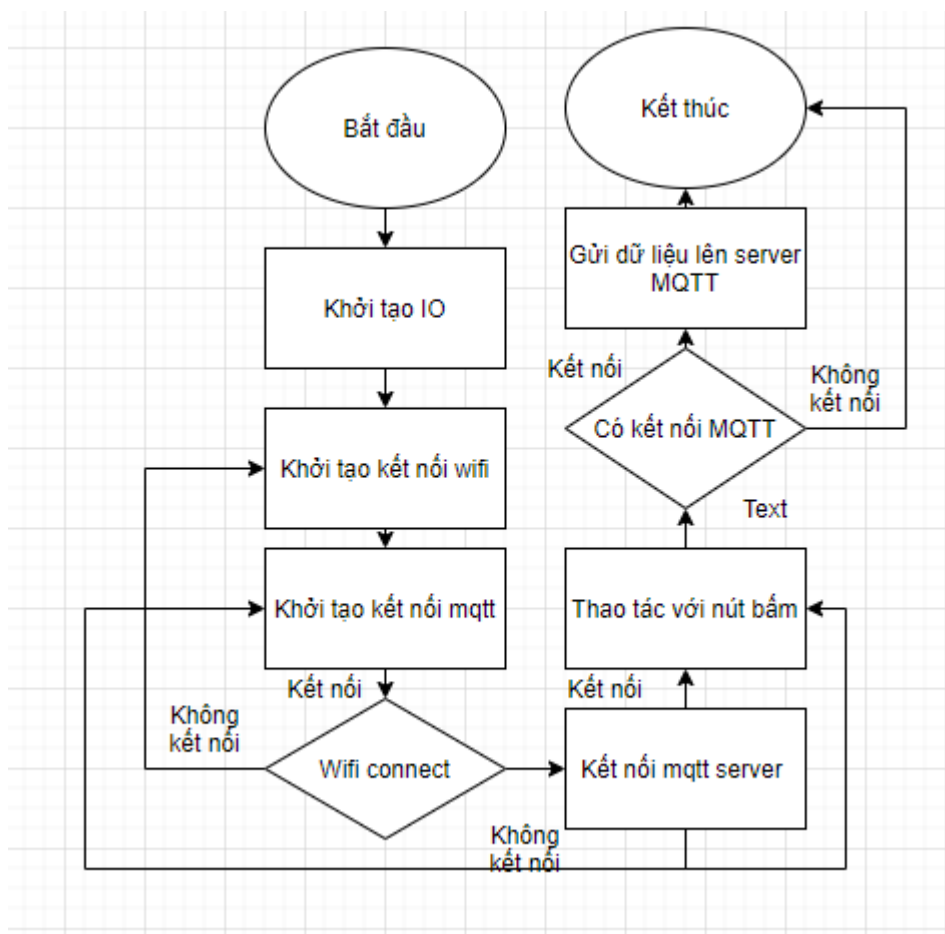
b, Mô tả chức năng các khối

App Windows và App mobile là các ứng dụng trên máy tính và điện thoại di động giúp người dùng dễ dàng thao tác với thiết bị phần cứng thông qua một server.

Thiết bị phần cứng : Đây là phần quan trọng nhất trong hệ thống, nó giúp người dùng điều khiển đóng cắt nguồn điện 220V từ xa hoặc trực tiếp trên thiết bị

Mqtt server là trung gian giúp chuyển dữ liệu giữa thiết bị phần cứng và app điều khiển

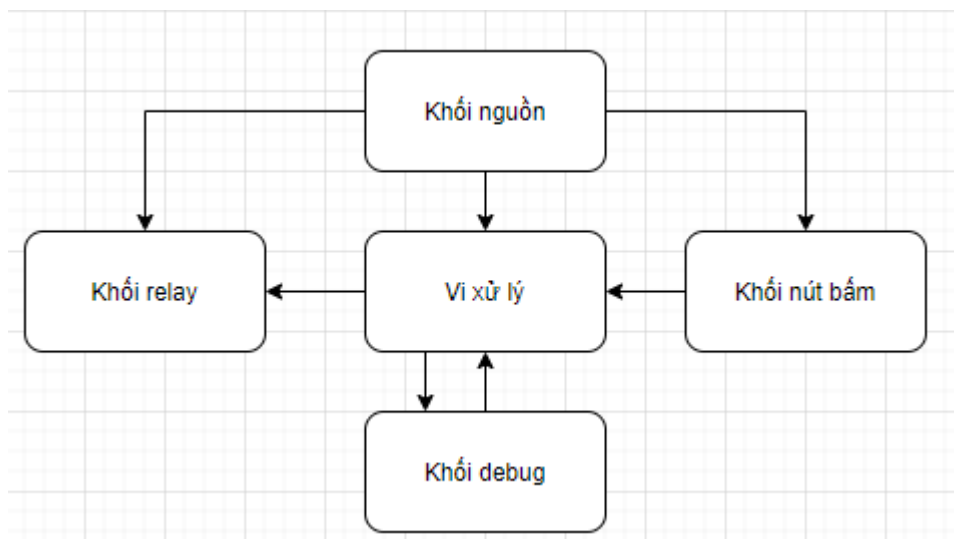
3.2.2 Lưu đồ thuật toán phần cứng



Hình 3. 2 Lưu đồ thuật toán phần cứng

3.3 Thiết kế mạch điều khiển

3.3.1 Các khối cơ bản trong mạch điều khiển



Hình 3. 3 Các khối cơ bản

Khối nguồn: Đầu vào là nguồn 220 AC được hạ áp cung cấp điện áp ổn định tới các khối trong hệ thống, khối nguồn được sử dụng trong hệ thống có điện áp 5 VDC và 3.3 VDC.

Khối vi điều khiển: Có chức năng kết nối Wifi nhận dữ liệu từ khối điều khiển Application, tính toán dữ liệu và gửi về khối Relay. Vi điều khiển được sử dụng trong hệ thống là ESP8266

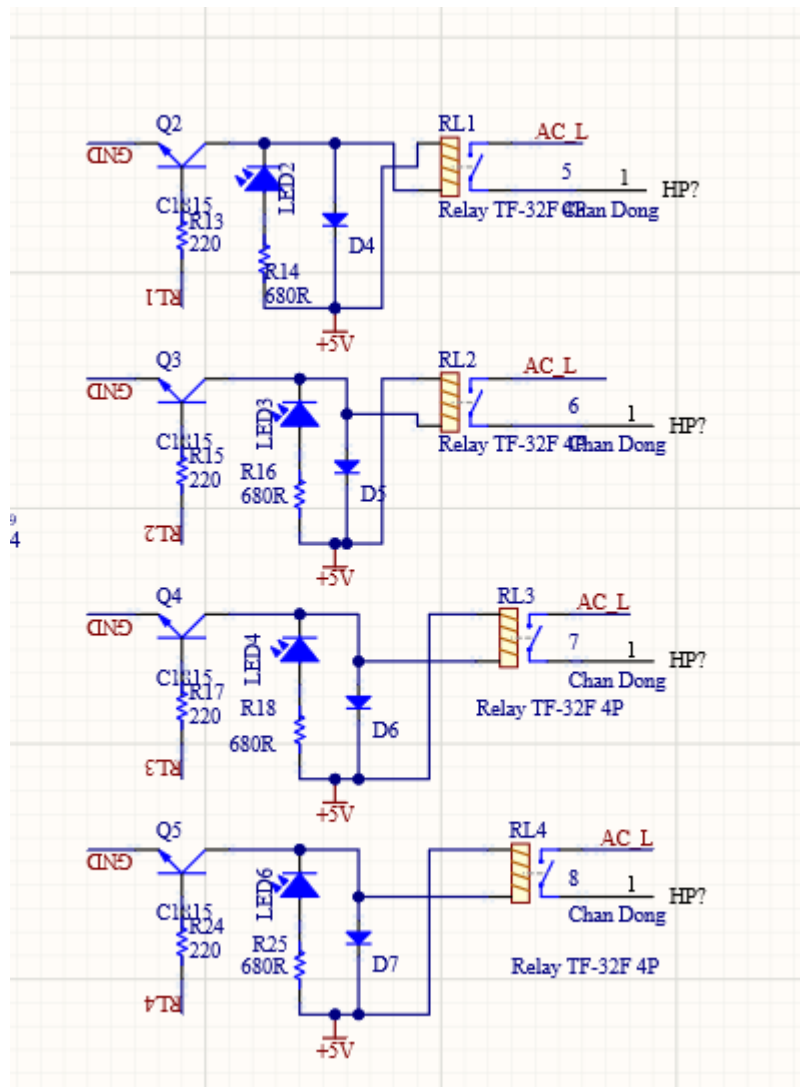
Khối debug: Gồm 6 chân để kết nối với máy tính qua giao thức UART để nạp chương trình và hiển thị lỗi trong quá trình code

Khối Relay: Khi nhận được tín hiệu từ vi điều khiển nó thực hiện đóng / mở theo tín hiệu nhận được.

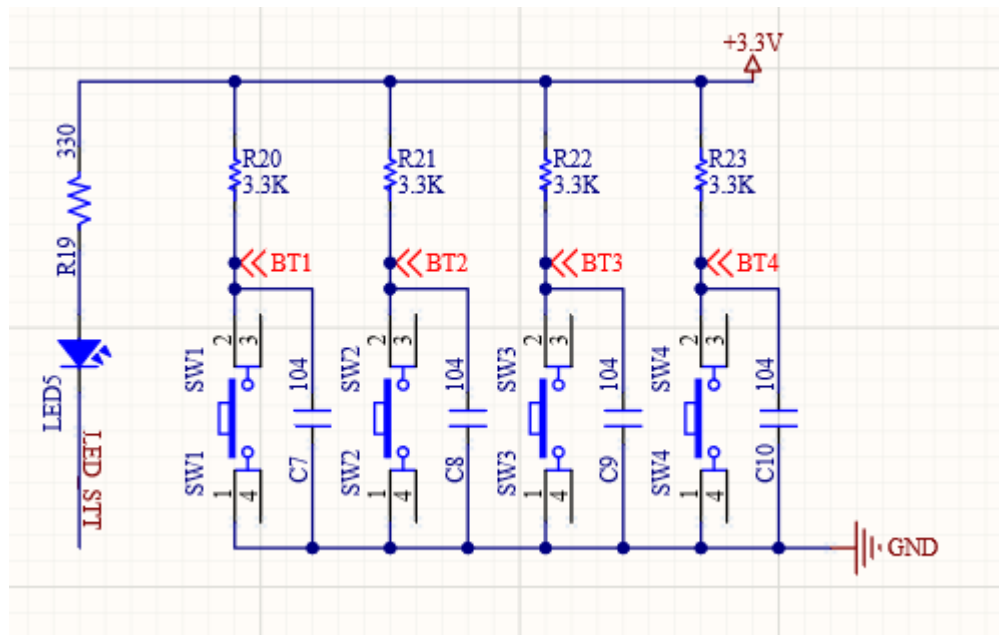
Khối nút bấm : Thực hiện thao tác của người dùng tùy theo mục đích sử dụng như đóng hay mở relay thông qua vi điều khiển.

3.3.2 Thiết kế mạch điều khiển

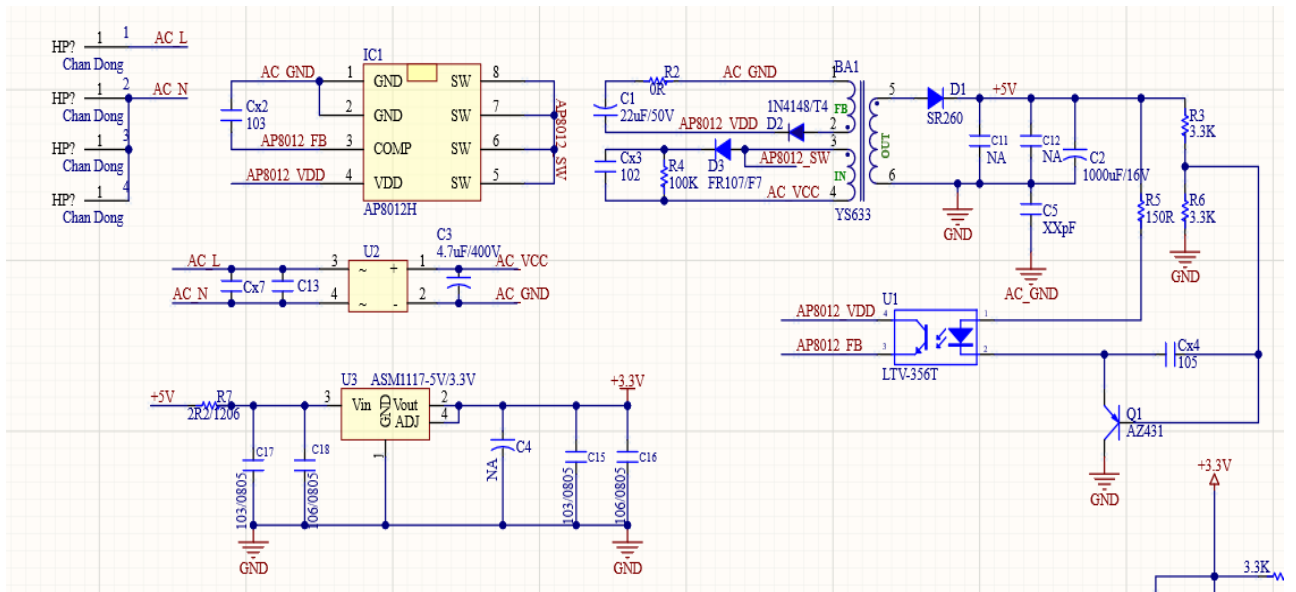
a, Sơ đồ nguyên lý



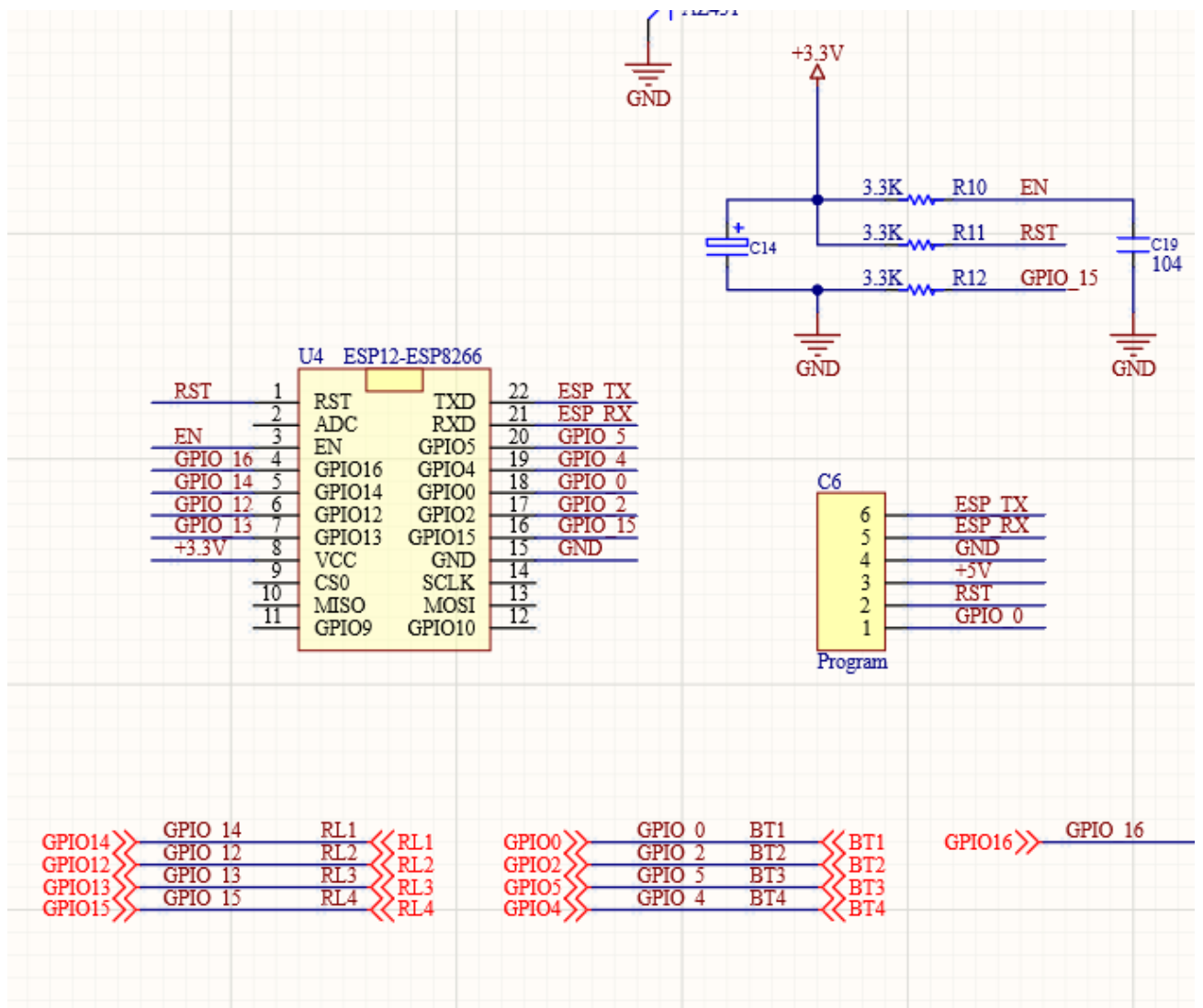
Hình 3. 4 Khối đầu ra



Hình 3. 5 Khối đầu vào

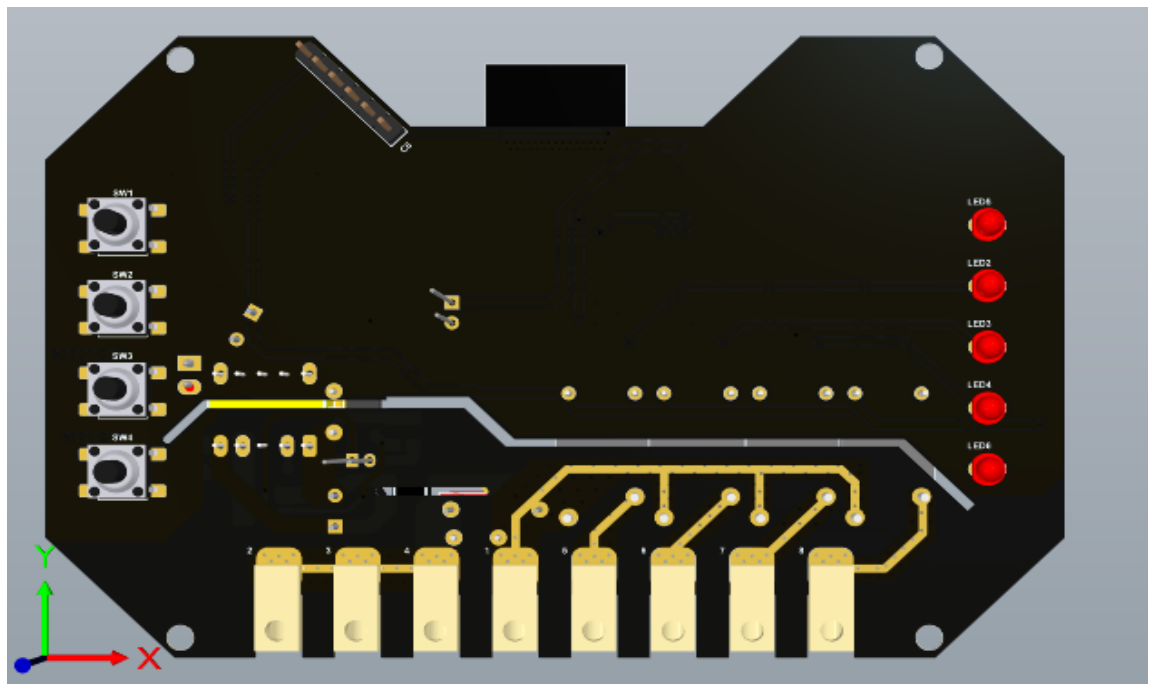


Hình 3. 6 Khối nguồn

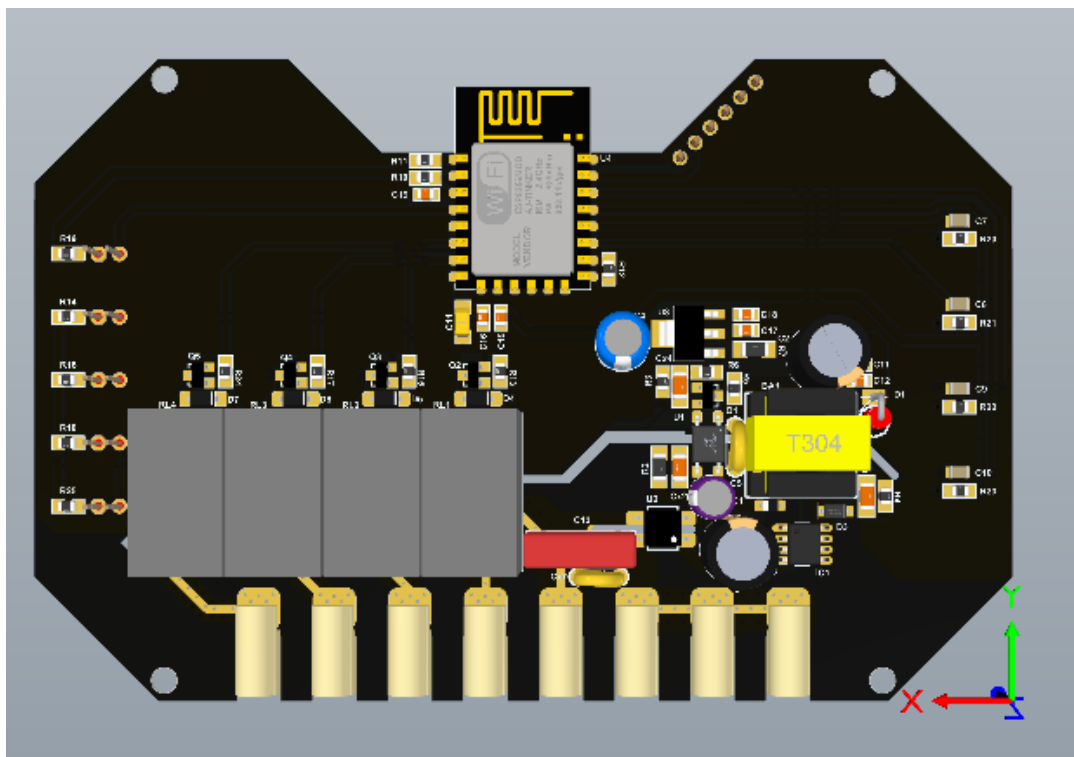


Hình 3. 7 Khối vi điều khiển

b, PCB layout



Hình 3. 8 Top layout



Hình 3. 9 Bottom layout

3.4 Xây dựng chuẩn giao tiếp giữa app và thiết bị phần cứng

3.4.1 Server kết nối

Mqtt Server	
Server	<u>mqtt.ngoinhaiot.com</u>
user	ngocphong260899
password	ngocphong260899
port	1111 or 2222
topic sub	ngocphong260899/device
topic pub	ngocphong260899/app

Bảng 3. 1 Server kết nối MQTT

3.4.2 Lệnh điều khiển thiết bị

Lệnh điều khiển			
STT	Tên	Lệnh	Chip trả về
1	cmd1_on	{ "sw_wifi": 1,"status":1 }	{ "sw_wifi":1,"pos":1,"status":1 }
2	cmd1_off	{ "sw_wifi": 1,"status":2 }	{ "sw_wifi":1,"pos":1,"status":0 }
3	cmd2_on	{ "sw_wifi": 1,"status":3 }	{ "sw_wifi":1,"pos":2,"status":1 }
4	cmd2_off	{ "sw_wifi": 1,"status": 4 }	{ "sw_wifi":1,"pos":2,"status":0 }
5	cmd3_on	{ "sw_wifi": 1,"status":5 }	{ "sw_wifi":1,"pos":3,"status":1 }
6	cmd3_off	{ "sw_wifi": 1,"status":6 }	{ "sw_wifi":1,"pos":3,"status":0 }

Bảng 3. 2 Lệnh điều khiển thiết bị

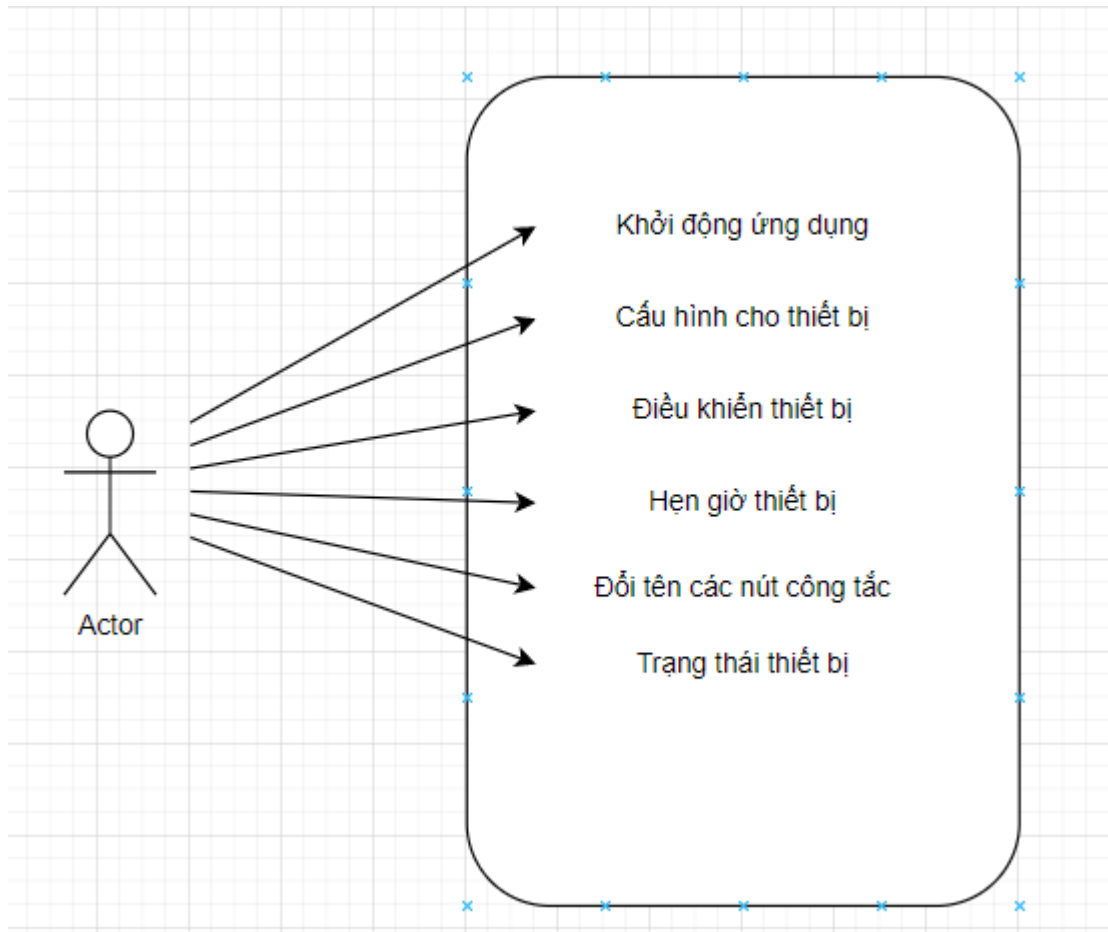
3.4.3 *Lệnh lấy trạng thái thiết bị*

Lấy trạng thái thiết bị			
<i>STT</i>	<i>Tên</i>	<i>Lệnh</i>	<i>Chip trả về</i>
1	get_State_device	{ "sw_wifi":2 }	{ "sw_wifi":2,"pos":[0,1,0]}
			{ "sw_wifi":2,"pos":0,"status":0,"ssid":"TP-Link_C554","streng":-53}

Bảng 3. 3 Lệnh lấy trạng thái thiết bị

CHƯƠNG 4: XÂY DỰNG ỨNG DỤNG

4.1 Biểu đồ use case tổng quát



Hình 4. 1 Biểu đồ use case

4.2 Xây dựng ứng dụng Windows

4.2.1 Giao diện ứng dụng

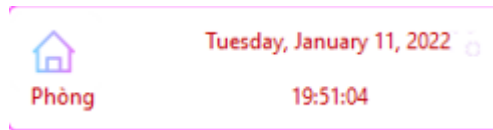
a, Giao diện tổng quan

Hình 4. 3 Giao diện đổi tên phòng

Đổi tên các công tắc tương ứng với các thiết bị đã nối với đầu ra của công tắc giúp người sử dụng dễ dàng nhận biết là đang điều khiển thiết bị nào.

4.2.2 Mô tả chức năng phần mềm

a, Thông tin thời gian



Hình 4. 4 Thông tin thời gian

Giúp người dùng biết được thời gian hiện tại.

b, Điều khiển công tắc



Hình 4. 5 Giao diện điều khiển công tắc

Điều khiển công tắc theo ý muốn của người sử dụng thông qua giao diện phần mềm (ON là mở, OFF là tắt).

c, Hẹn giờ bật tắt công tắc

Hình 4. 6 Hẹn giờ công tắc

Giúp người sử dụng hẹn giờ bật tắt thiết bị theo thời gian mong muốn trong ngày.

d, Thông tin thiết bị phần cứng

Hình 4. 7 Thông tin thiết bị phần cứng

Cung cấp trạng thái hiện tại của các công tắc và thông tin Wifi mà thiết bị đang sử dụng để kết nối tới server mqtt

4.3 Xây dựng phần mềm React Native

4.3.1 Giao diện cấu hình wifi cho thiết bị

8:29 ● ● P ● 8:29 ● ● P ● ● ● 40%

←

Tìm kiếm & kết nối tới Thiết bị

Hunonic

Password 🔍

Bạn vui lòng chọn wifi 2.4Ghz, nhập đúng mật khẩu wifi và ấn nút cấu hình. Quá trình cấu hình có thể kéo dài từ 1-2 phút, trong khi cấu hình vui lòng chờ kết nối tới thiết bị

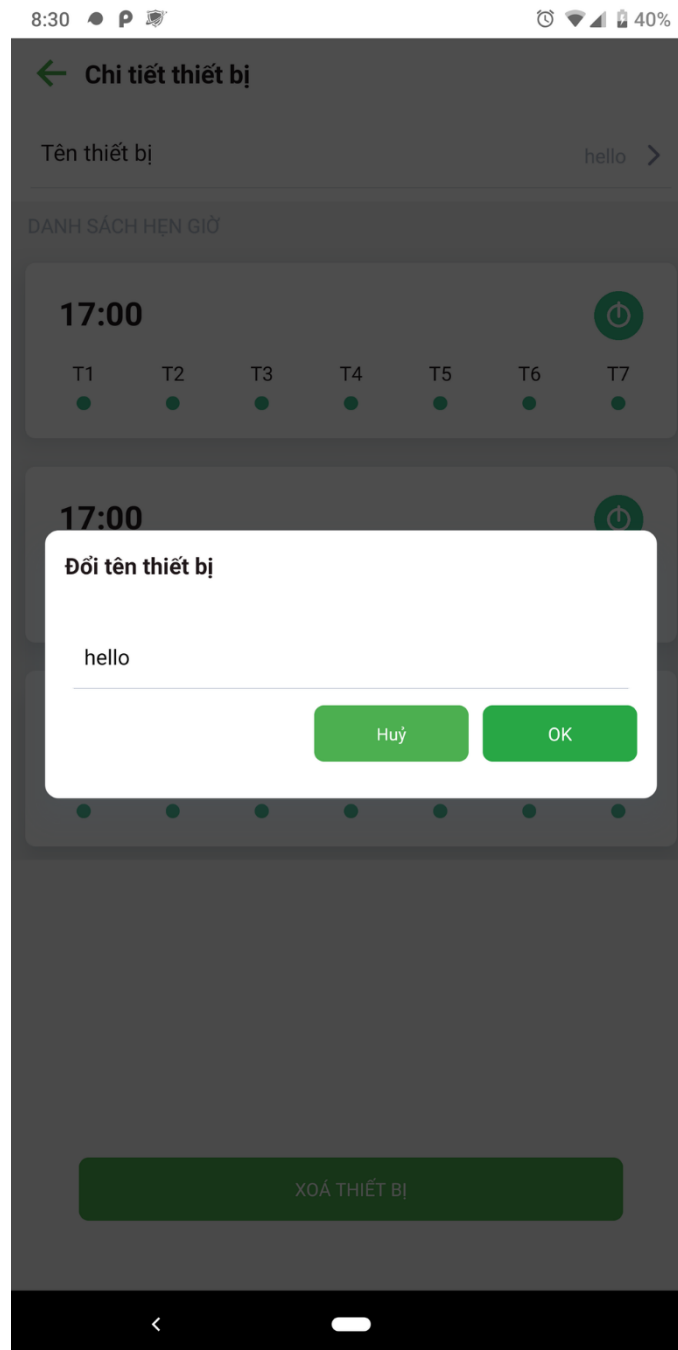
TÌM KIẾM →

< —

Hình 4. 8 Giao diện cấu hình wifi cho thiết bị

Gửi tên và mật khẩu wifi từ app sang cho thiết bị phần cứng kết nối để lấy thông tin lên mạng kết nối MQTT server.

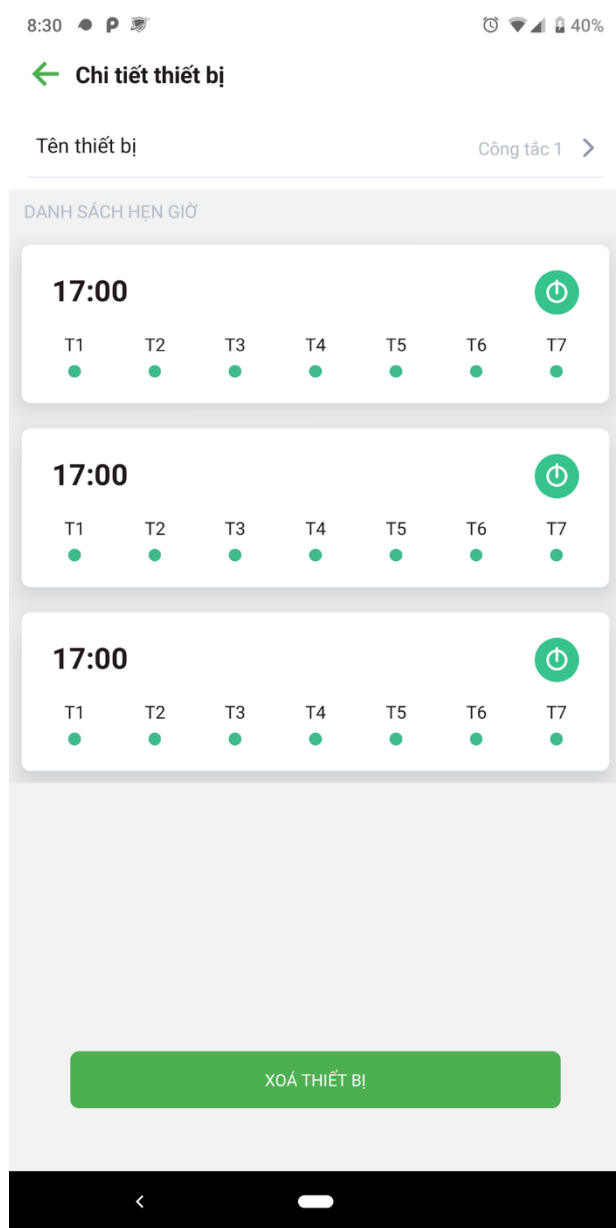
4.3.2 Giao diện đổi tên công tắc



Hình 4. 9 Giao diện đổi tên công tắc

Đổi tên các nút công tắc theo mong muốn của người sử dụng

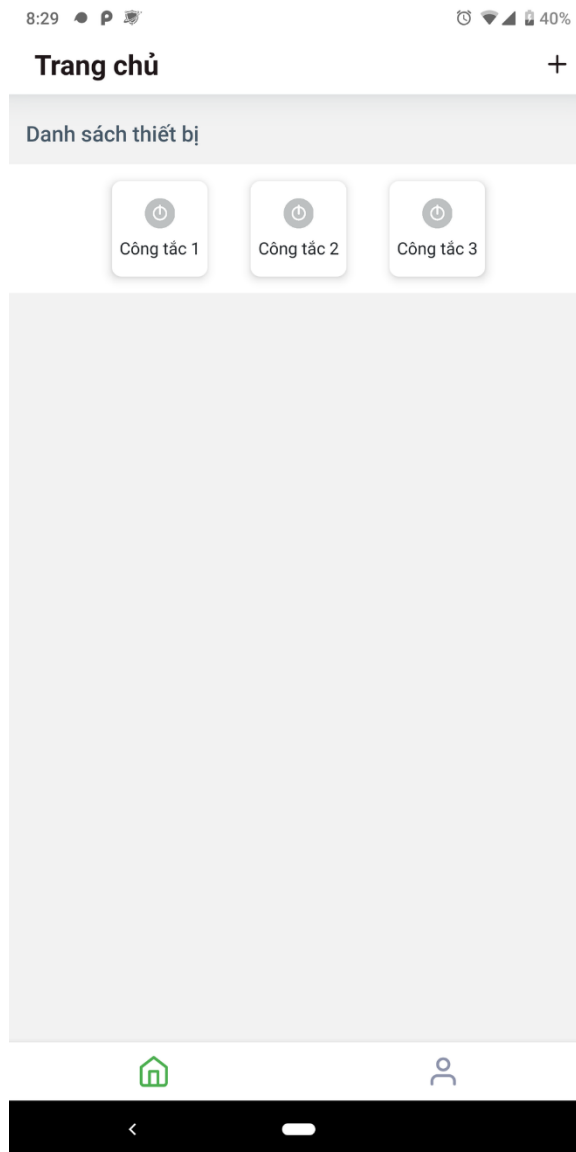
4.3.3 Giao diện hẹn giờ thiết bị



Hình 4. 10 Giao diện hẹn giờ thiết bị

Đặt thời gian bật, tắt thiết bị theo mong muốn của người dùng trong khoảng thời gian trong ngày.

4.3.4 Giao diện điều khiển thiết bị



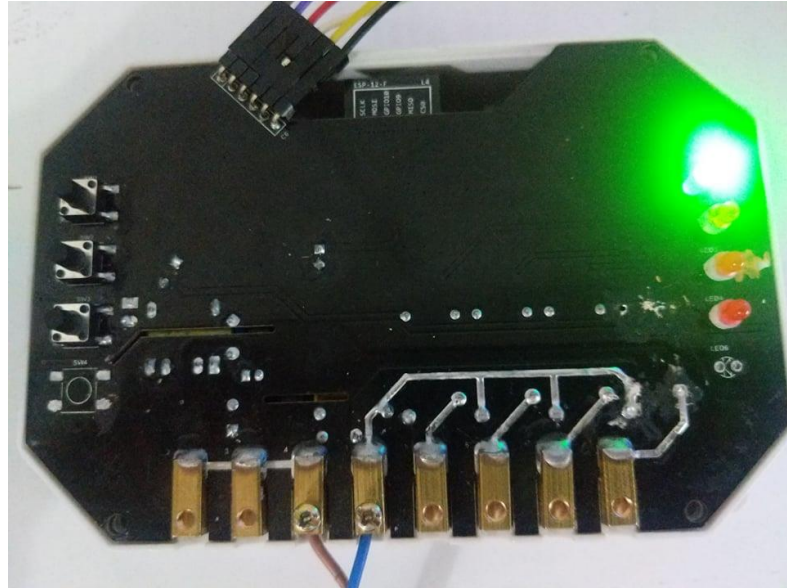
Hình 4. 11 Giao diện điều khiển thiết bị

Điều khiển bật, tắt thiết bị theo cách sử dụng của người dùng

CHƯƠNG 5: KẾT QUẢ ĐÁNH GIÁ

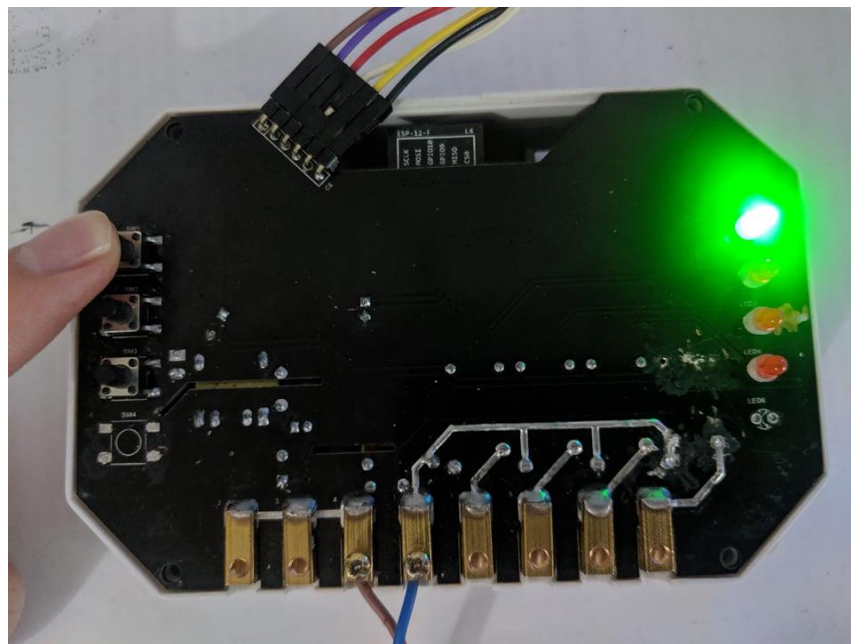
5.1 Hướng dẫn sử dụng sản phẩm

Bước 1: Cấp nguồn cho thiết bị phần cứng



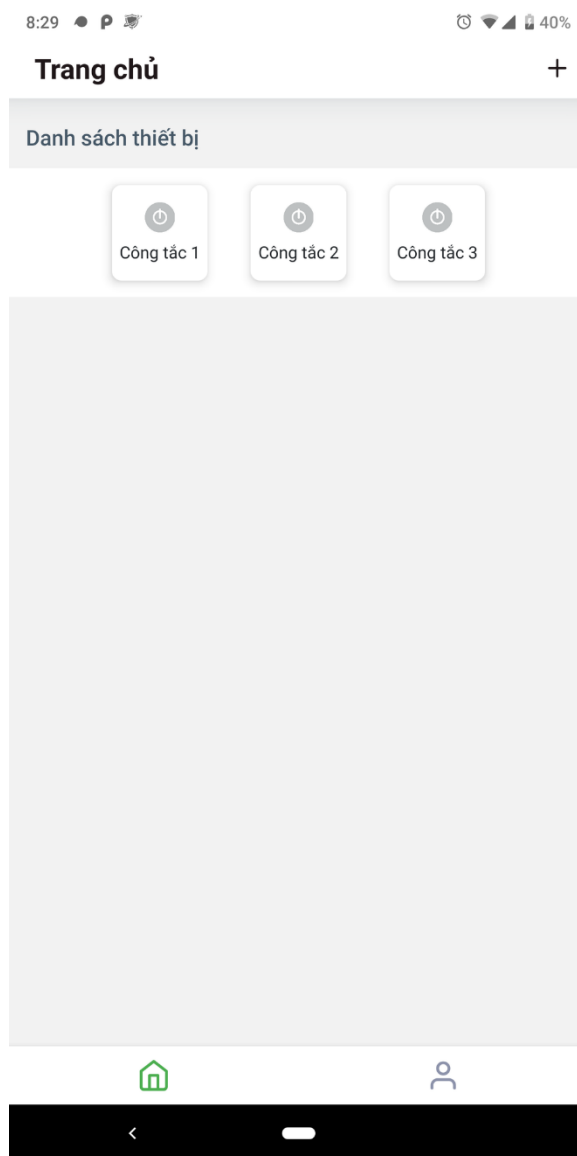
Hình 5. 1 Cấp điện cho thiết bị phần cứng

Bước 2: Nhấn giữ nút trên thiết bị trong khoảng 5 giây cho đến khi led nhấp nháy đều theo nhịp thì bỏ tay khỏi nút.



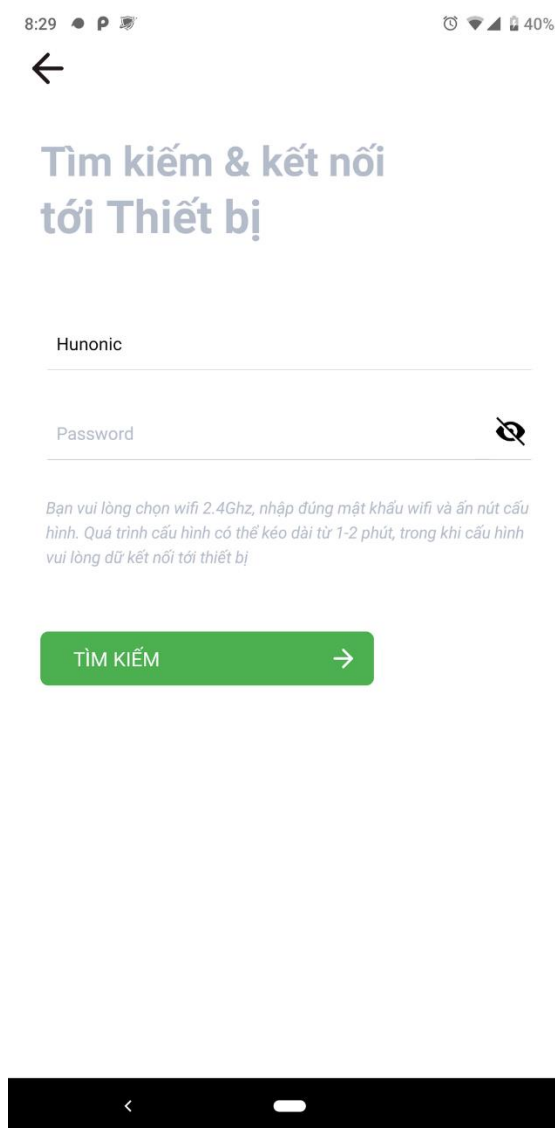
Hình 5. 2 Cấu hình wifi cho thiết bị

Bước 3: Mở app trên điện thoại android



Hình 5. 3 Mở ứng dụng trên điện thoại

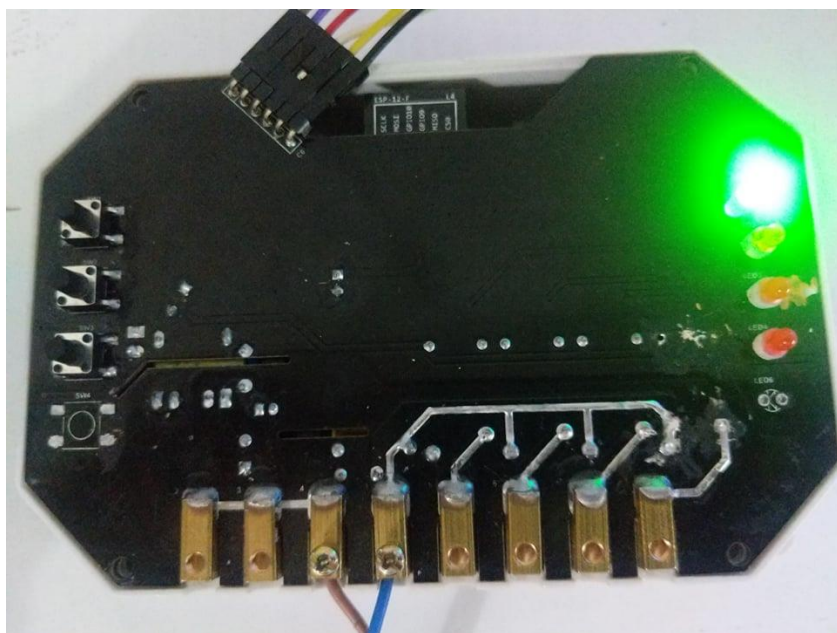
Bước 4: Ấn vào dấu + trên góc màn hình bên phải để vào phần config thiết bị



Hình 5. 4 Cấu hình wifi cho thiết bị phần cứng

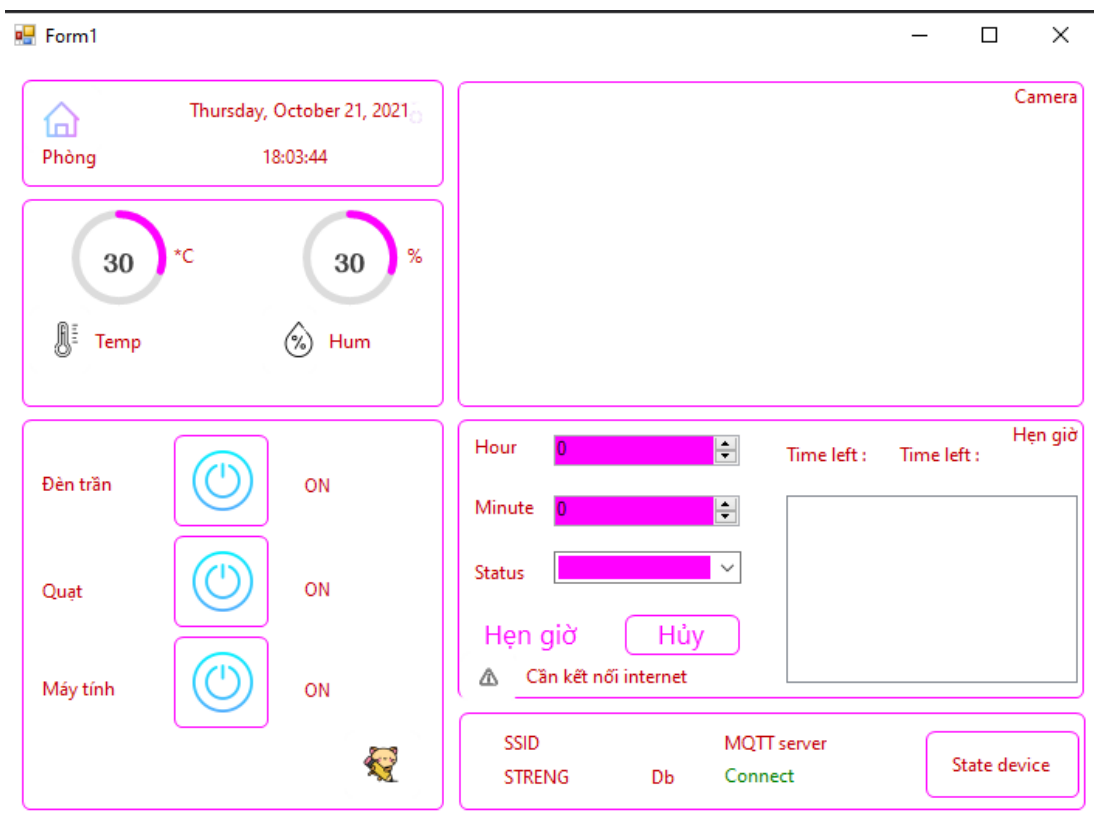
Bước 5: Nhập thông tin Wifi vào và bấm kết nối đến khi có thông báo thành công

Bước 6: Kiểm tra thiết bị phần cứng đã nhận được kết nối wifi và server hay chưa. Nếu kết nối thì đèn led màu xanh báo sáng và ngược lại.



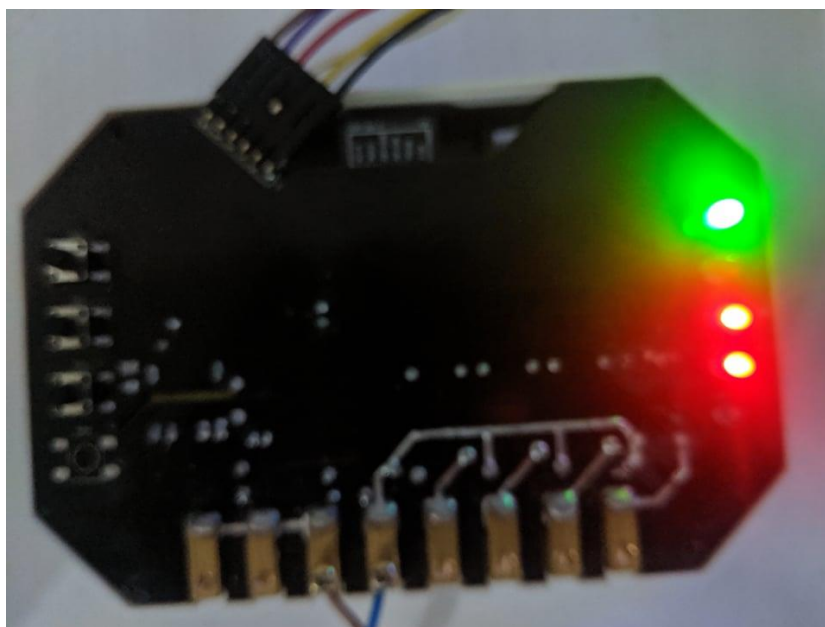
Hình 5. 5 Kiểm tra kết nối Wifi

Bước 7: Mở app Windows và app Android bắt đầu điều khiển.



Hình 5. 6 Sử dụng app windows để điều khiển thiết bị

Bước 8: Kết quả điều khiển(led đỏ là bật, không là tắt)



Hình 5. 7 Kiểm tra trạng thái hoạt động thiết bị

5.2 Ưu điểm của hệ thống

- Phần mềm sử dụng đa nền tảng: Windows, Android, Ios.
- Giao diện trực quan dễ dàng thao tác.
- Thiết bị phân cứng nhỏ gọn, dễ dàng kết nối với wifi.
- Điều khiển thiết bị từ xa thông qua internet giúp người sử dụng tiết kiệm thời gian
- Đồng bộ trạng thái đầu ra giữa thiết bị và App.
- Dễ dàng thao tác lắp đặt thiết bị với các thiết bị điện gia dụng khác.
- Điều khiển đồng thời từ nút bấm trên thiết bị và từ trên app điện thoại, máy tính

5.3 Nhược điểm của hệ thống

- Công suất hoạt động các kênh của thiết bị nhỏ (220v, 2A).
- Một số trường hợp không thể gửi thông tin wifi cho thiết bị.
- Xảy ra hiện tượng mất kết nối wifi và không kết nối lại wifi cũ để hoạt động.
- Thiết bị không thể hẹn giờ lưu trực tiếp trên thiết bị.
- Khi hẹn giờ trên app thì chỉ hẹn được tối đa một kênh trên một lần sử dụng.
- Nếu đang trong hẹn giờ mà thoát app thì chức năng hẹn giờ bị vô hiệu hóa

KẾT LUẬN

Hệ thống được tạo ra nhằm thay thế các loại công tắc cơ truyền thống đang được sử dụng trong các hộ gia đình hiện tại, giúp giảm chi phí lắp đặt và sử dụng cùng với đó là chi phí rẻ, hệ thống bớt công kênh.

Việc tạo ra một ứng dụng đa nền tảng nhằm giúp người dùng dễ dàng kiểm soát thiết bị mình đang sử dụng, khả năng điều khiển từ xa giúp người dùng linh hoạt trong thời gian sử dụng.

Những công việc đã làm được

Qua quá trình thực hiện đề tài đồ án tốt nghiệp này, em đã thu được những kết quả chính như sau:

- Nắm được cơ bản về lập trình React Native, Winform Application, C/C++
- Xây dựng và phát triển được ứng dụng với các ngôn ngữ trên.
- Kết nối và xử lý dữ liệu với mqtt server và json.
- Giải quyết được vấn đề đồng bộ dữ liệu giữa thiết bị phần cứng và app.
- Thiết kế và lập trình được mạch điều khiển dựa trên vi điều khiển Esp8266.

Một số hạn chế

- Hai ứng dụng trên máy tính và điện thoại không thể thay thế cho nhau do hai hệ điều hành khác nhau
- Việc cài đặt ứng dụng trên hệ điều hành Ios rất khó khăn đòi hỏi môi trường Mac OS
- Vấn đề hẹn giờ thiết bị chưa được xử lý triệt để.

Hướng phát triển

- Triển khai server: Hướng tới lưu trữ dữ liệu trên server, sử dụng server để đồng bộ hẹn giờ giữa các thiết bị.
- Thêm các tính năng mới như: Đổi wifi, khóa trẻ em
- Nâng cao công suất hoạt động của thiết bị

NHẬN XÉT

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

.....

Thái Nguyên, NgàyThángNăm 2022

Giáo viên hướng dẫn

TÀI LIỆU THAM KHẢO

- [1] <https://docs.microsoft.com/en-us/dotnet/csharp/>
- [2] <https://reactnative.dev/docs/components-and-apis>
- [3] <http://codientuvina.com/thiet-ke-mach-dien-bang-altium.html>