

Mục lục

Phần 1. Giới thiệu.....	2
1. Giới thiệu tổng quan về mobile & pervasive computing	2
Mobile computing.	2
Pervasive Computing.....	2
2. Giới thiệu về project sẽ phát triển	3
Phần 2: Công nghệ.....	4
1. Giao thức MQTT	4
2. Esp8266 v12	7
3. Mosquitto	7
4. OpenHab	8
Phần 3: Thiết kế – Cài đặt.....	8
1. Kiến trúc hệ thống.....	8
2. Sơ đồ use-case.....	12
3. Giao diện ứng dụng.....	14
.....	14
Phụ lục: Hướng dẫn cài đặt và sử dụng.....	16
Hướng dẫn cài đặt Mosquitto	16
Hướng dẫn cài openHab	17

Phần 1. Giới thiệu

1. Giới thiệu tổng quan về mobile & pervasive computing

Mobile computing.

Tính toán di động là sự tương tác giữa con người và máy tính, trong đó máy tính có thể sẽ được vận chuyển trong quá trình sử dụng.

Tính toán di động là “đặt máy tính và tất cả các file cần thiết và phần mềm chung với nhau”. Tính toán di động là bất kỳ loại máy tính nào có liên kết với nhau tạo thành mạng cá nhân không dây.

Có 3 lớp khác nhau của máy tính di động.

Portable computer: máy tính xách tay: trọng lượng nhẹ bao gồm bàn phím vật tập hợp đầy đủ và chủ yếu được dùng như host như: laptop, notebook...

Mobile phone: Điện thoại di động không chỉ giới hạn giao tiếp bằng giọng nói, điện thoại phổ thông, đt thông minh, phonepad...

Wearable: máy tính có thể mặc đeo được: hầu hết được giới hạn một số chức năng chủ yếu như: đồng hồ, kính, dây đeo tay...

Pervasive Computing.

Có nghĩa là “tồn tại khắp mọi nơi”.

Chính là tương lai của mạng máy tính cho phép truyền thông giữa các cá nhân và thiết bị cũng như thiết bị với chính nó.

Các thiết bị rất nhỏ và được nhúng vào bất kỳ đối tượng nào như: quần, áo, xe hơi ...

Được gọi là ***Ubiquitous Computing***.

Ví dụ.

Hệ thống Định vị Toàn cầu (tiếng Anh: Global Positioning System - GPS) là hệ thống xác định vị trí dựa trên vị trí của các vệ tinh nhân tạo, do Bộ Quốc phòng Hoa Kỳ thiết kế, xây dựng, vận hành và quản lý. Trong cùng một thời điểm, tọa độ của một điểm trên mặt đất sẽ được xác định nếu xác định được khoảng cách từ điểm đó đến ít nhất ba vệ tinh.

Nó có thể được sử dụng như một thiết bị định vị của xe, bảo vệ người dùng khỏi bị thất lạc. Khi một thiết bị giám sát GPS được nhúng vào điện thoại di động, nó được sử dụng cho việc tìm kiếm hướng, tính toán khoảng cách, và nhiều hơn nữa. Nó cũng có thể được sử dụng để xác định vị trí của một chiếc xe bị đánh cắp hoặc vật nuôi bị mất.

Những lợi thế khác của hệ thống định vị toàn cầu GPS là tính tương thích Google Earth, dễ dàng sử dụng, nhỏ và dễ dàng che giấu, tính chính xác trọng việc theo dõi, và sức mạnh của nam châm của nó cho phép nó được đặt dưới gầm xe và nó có khả năng chịu nước.

Một điều khác về GPS là nó giúp cha mẹ theo dõi nơi đến của con em họ. Nó có thể được mang theo bởi một đứa trẻ hay được đặt trong một chiếc xe của teen để thông báo cho cha mẹ vị trí chính xác của đứa trẻ. Điều này rất hữu ích khi xảy ra trường hợp khẩn cấp. Hệ thống tự động gửi thông tin cho phụ huynh khi trường hợp như vậy xảy ra.

2. Giới thiệu về project sẽ phát triển

Trong đời sống hiện nay, internet là 1 phần không thể thiếu trong cuộc sống nó góp phần thúc đẩy phát triển, cải tiến năng suất hoạt động. Cần phải khai thác triệt để giữa các thiết bị trong nhà với các thiết bị ngoài giúp cho cuộc sống dễ dàng tiện nghi hơn. Chính vì thế giúp tạo động lực nghiên cứu về hệ thống Smart Home. Smart Home là sử dụng các thiết bị di động mà kết nối internet để điều khiển các thiết bị trong nhà.

Project gồm có phần server xử lý các lệnh gửi về từ client rồi xử lý sẽ gửi lại tất cả các client, phần client và giao diện trên điện thoại gồm button điều khiển các thiết bị, và thông tin thông báo từ các thiết bị gửi lên.

Gồm có phần cứng: các thiết bị điện tử: andruino, esp8266v12, đèn led, relay, cảm biến nhiệt, cảm biến nước. Laptop sẽ đặt server hệ thống, smartphone điều khiển từ client, modem nơi cung cấp điện chỉ IP

Mô tả chi tiết:

Sử dụng giao thức MQTT làm giao thức truyền nhận chính cho hệ thống.

Broker có thể đặt trên laptop, các thiết bị sẽ cùng liên kết với nhau qua mạng localhost. Hoặc Broker có thể đặt trên internet để mở rộng phạm vi hoạt động cũng như điều khiển được mọi nơi qua internet.

Mục đích chính sử dụng OpenHAB để xây dựng giao diện điều khiển. Việc xử lý dữ liệu do các Client thực hiện.

Các Client nhận lệnh điều khiển từ các Topic `/item/Command` để xử lý và gửi dữ liệu ngược lại Topic `/item/Status` để cập nhật trạng thái được chính xác.

Chức năng gồm có:

- Xem nhiệt độ trong phòng cảnh báo cháy nổ
- Xem lượng nước trong nhà và tự động bơm nước khi lượng nước thấp.
- Bật tắt đèn trong nhà

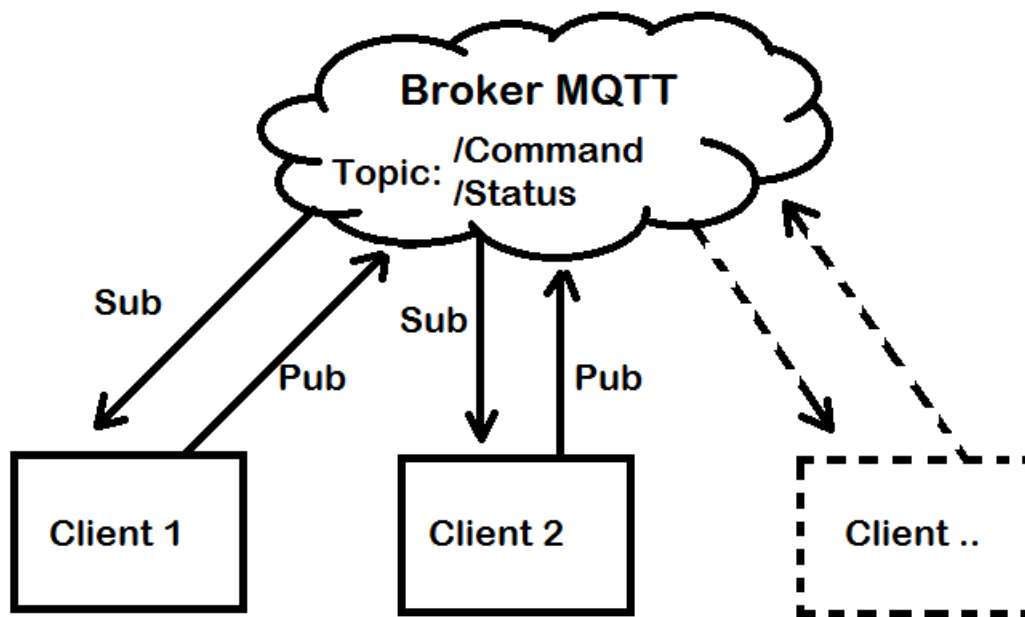
Phần 2: Công nghệ

1. Giao thức MQTT



MQTT được phát minh bởi tiến sĩ Andy Stanford-Clark của IBM, và Arlen Nipper của Arcom (nay Eurotech), vào năm 1999.

MQTT (Message Queuing Telemetry Transport) là một giao thức gửi 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. Hình dưới thể hiện sơ đồ chung của hệ thống sử dụng giao thức MQTT.



Các chức năng cơ bản trong giao thức MQTT:

Publish, subscribe

Trong một hệ thống sử dụng giao thức MQTT (hình 7), nhiều node trạm (gọi là mqtt client - gọi tắt là client) kết nối tới một MQTT server (gọi là broker). Mỗi client sẽ đăng ký một vài kênh (topic), ví dụ như "/client1/channel1", "/client1/channel2". Quá trình đăng ký này gọi là "subscribe. Mỗi client sẽ nhận được dữ liệu khi bất kỳ trạm nào khác gửi dữ liệu và kênh đã đăng ký. Khi một client gửi dữ liệu tới kênh đó, gọi là "publish"

.

QoS

Ở đây có 3 tùy chọn *QoS (Qualities of service) * khi "publish" và "subscribe":

QoS0 Broker/client sẽ gửi dữ liệu đúng 1 lần, quá trình gửi được xác nhận bởi chỉ giao thức TCP/IP, giống kiểu đem con bỏ chợ.

QoS1 Broker/client sẽ gửi dữ liệu với ít nhất 1 lần xác nhận từ đầu kia, nghĩa là có thể có nhiều hơn 1 lần xác nhận đã nhận được dữ liệu.

QoS2 Broker/client đảm bảo khi gửi dữ liệu thì phía nhận chỉ nhận được đúng 1 lần, quá trình này phải trải qua 4 bước bắt tay.

Retain

Nếu RETAIN được set bằng 1, khi gói tin được publish từ Client, Broker phải lưu trữ lại gói tin với QoS, và nó sẽ được gửi đến bất kỳ Client nào subscribe cùng kênh trong tương lai. Khi một Client kết nối tới Broker và subscribe, nó sẽ nhận được gói tin cuối cùng có RETAIN = 1 với bất kỳ topic nào mà nó đăng ký trùng. Tuy nhiên, nếu Broker nhận được gói tin mà có QoS = 0 và RETAIN = 1, nó sẽ hủy tất cả các gói tin có RETAIN = 1 trước đó. Và phải lưu gói tin này lại, nhưng hoàn toàn có thể hủy bất kỳ lúc nào.

Khi publish một gói dữ liệu đến Client, Broker phải set RETAIN = 1 nếu gói được gửi như là kết quả của việc subscribe mới của Client (giống như tin nhắn ACK báo subscribe thành công). RETAIN phải bằng 0 nếu không quan tâm tới kết quả của việc subscribe.

LWT

Gói tin LWT (last will and testament) không thực sự biết được Client có trực tuyến hay không, cái này do gói tin KeepAlive đảm nhận. Tuy nhiên gói tin LWT như là thông tin điều gì sẽ xảy đến sau khi thiết bị ngoại tuyến.

2. Esp8266 v12



ESP8266-12 là module wifi giá rẻ và được đánh giá rất cao cho các ứng dụng liên quan đến Internet và Wifi cũng như các ứng dụng truyền nhận sử dụng thay thế cho các module RF khác.

ESP8266 là một chip tích hợp cao, được thiết kế cho nhu cầu của một thế giới kết nối mới, thế giới Internet of thing (IOT). Nó cung cấp một giải pháp kết nối mạng Wi-Fi đầy đủ và khép kín, cho phép nó có thể lưu trữ các ứng dụng hoặc để giảm tải tất cả các chức năng kết nối mạng Wi-Fi từ một bộ xử lý ứng dụng.

ESP8266 có xử lý và khả năng lưu trữ mạnh mẽ cho phép nó được tích hợp với các bộ cảm biến, vi điều khiển và các thiết bị ứng dụng cụ thể khác thông qua GPIOs với một chi phí tối thiểu và một PCB tối thiểu.

3. Mosquitto

Mosquitto là một broker open source cho phép ta cài đặt và thực thi giao thức MQTT phiên bản 3.1 và 3.1.1 cung cấp các phương thức đơn giản để gửi bản tin theo mô hình publish/subscribe. Đây là một open source rất phổ biến trong việc triển khai các ứng dụng M2M có sử dụng giao thức MQTT.

Trong bài viết này, tôi sẽ hướng dẫn mọi người cài đặt mosquitto trên hệ thống hệ điều hành Linux bao gồm: Ubuntu (các bản phân phối thuộc Debian) và Raspberry Pi. Mosquitto thực chất là một phần mềm cho phép ta triển khai một broker lên một nền tảng bất kỳ (kể cả: windows, mac), tùy

từng ứng dụng và phương pháp triển khai mà sẽ cài đặt broker lên các đối tượng nhất định nào đó.

4.OpenHab

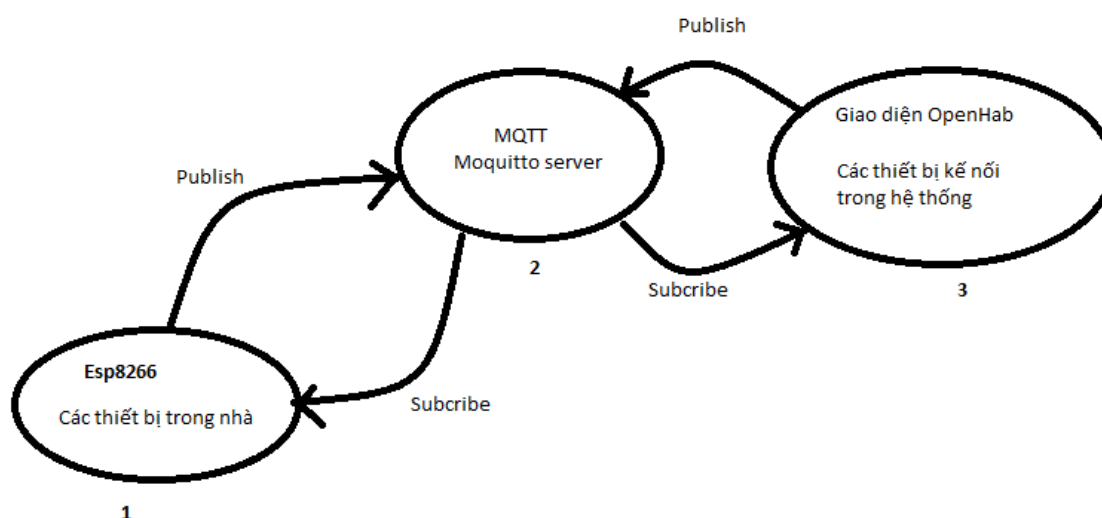
OpenHAB là phần mềm miễn phí nguồn mở có chức năng làm bộ điều khiển trung tâm để giao tiếp với rất nhiều các loại thiết bị khác nhau (kể cả các thiết bị thương mại) của rất nhiều hãng sản xuất vào trong một hệ thống Smart Home hoàn chỉnh.

OpenHAB được cài đặt dưới dạng một website, có thể chạy trên rất nhiều platform (Windows, Linux, ARM...), nó được viết bằng ngôn ngữ Java. Vì vậy các thiết bị có hỗ trợ JVM (Java Virtual Machine) là có thể cài đặt được Openhab. Có thể quan sát và điều khiển các thiết bị trong căn nhà của mình từ các thiết bị như PC, smart phone hay tablet...,

Một chức năng rất hay của OpenHAB là có một engine quản lý và thực thi các rules (thiết lập ngữ cảnh) giúp cho căn nhà thông minh hơn bằng cách tự động điều chỉnh khi trạng thái các cảm biến thay đổi.

Phần 3: Thiết kế – Cài đặt

1. Kiến trúc hệ thống



Kiến trúc hệ thống gồm có 3 phần:

1- Nơi các thiết bị trong nhà kết nối với esp8266 là trung tâm xử lý các câu lệnh và nhận và gửi các lệnh từ server gửi đến. Các thiết bị điện tử chủ yếu giao tiếp thông qua 2 loại dữ liệu:

Digital gồm có 2 giá trị: 0 và 1, ví dụ như sensor cảm biến nước gồm có giá trị có nước và không có nước;

Analog gồm nhiều giá trị: ví dụ: như nhiệt độ thì gồm nhiều giá trị nhiều độ khác nhau

Đây là nơi xử lý các tín hiệu từ người dùng yêu cầu và thực thi yêu cầu của người dùng lên các thiết bị trong nhà

2- Nơi server nhận tất cả các thông tin từ client gửi về và sẽ gửi lại tất cả thông tin đó cho tất cả client nào subscribe các topic. Trong đề án này lấy laptop làm 1 server MQTT.

- Chạy Localhost: Hệ thống sẽ chạy theo kiểu Localhost. Nghĩa là các Client sẽ cùng kết nối với mạng (Wifi, Ethernet của Router) với Broker. Phù hợp với hệ thống nội bộ.
- Chạy Internet: Cần được cung cấp 1 địa chỉ IP internet tĩnh. Các Client sẽ kết nối với Broker qua IP internet. Điều này sẽ giúp hệ thống mở rộng xa hơn trên môi trường Internet.

3- Nơi giao diện của người dùng, các client gửi và nhận các thông tin từ server mà client khác xử lý. Giao diện được chạy trên các thiết bị điện thoại có kết nối với IP của server đó hoặc có thể kết nối không dùng chung mạng wifi, dùng tài khoản thông qua cloud rồi sẽ kết nối tới server ở nơi đó.

Trong OpenHAB: Ta có 2 topic chính:

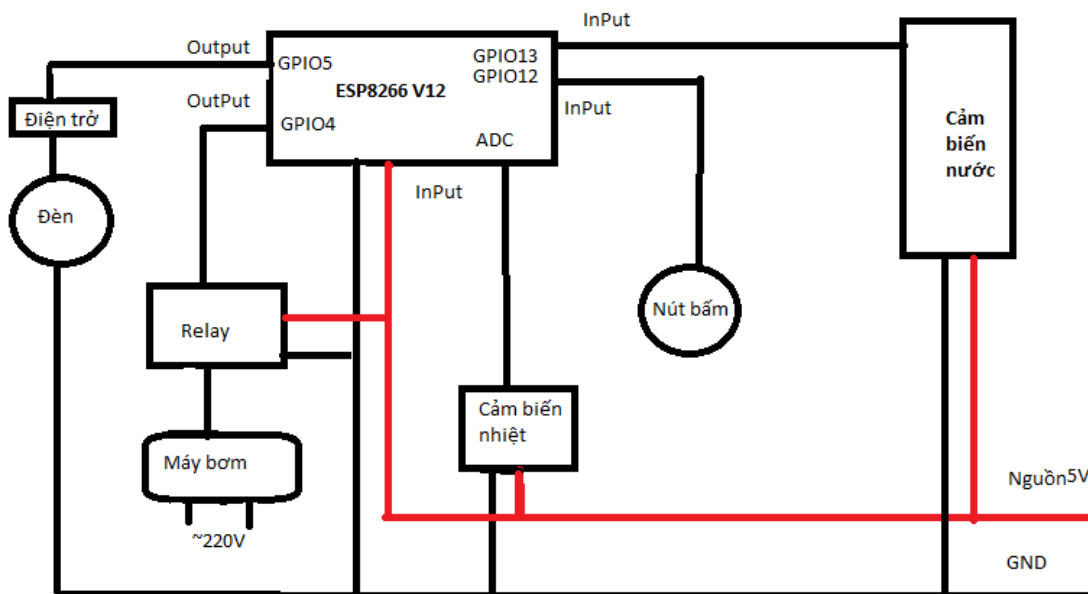
➤ `/item/Command`:

- Khi người dùng thao tác (vd: bật tắt công tắc) thì OpenHAB sẽ gửi(Publish) dữ liệu (vd: ON OFF) đến một Topic (vd: `/Light1/Command`). Khi đó các thiết bị Client nào đã đăng kí (Subscribe) Topic đó sẽ nhận được dữ liệu. Các Client xử lý dữ liệu sau đó gửi giá trị về topic `/item/Status` để cập nhật được giá trị cũng như cho thấy được tính chính xác của hệ thống.

➤ /{item}/Status:

- Topic này chủ yếu dùng để cập nhật trạng thái (vd: trạng thái đóng mở công tắc, v.v...). Ta sẽ gửi dữ liệu cần hiển thị lên Topic này.

Sơ đồ mạch điện trong hệ thống nhà thông minh



Chú thích:

GPIO12, GPIO13, GPIO4, GPIO5: các chân digital

ADC: chân analog

Input. Tin hiệu chuyển vào cho esp

OutPut: tín hiệu chuyển ra từ esp.

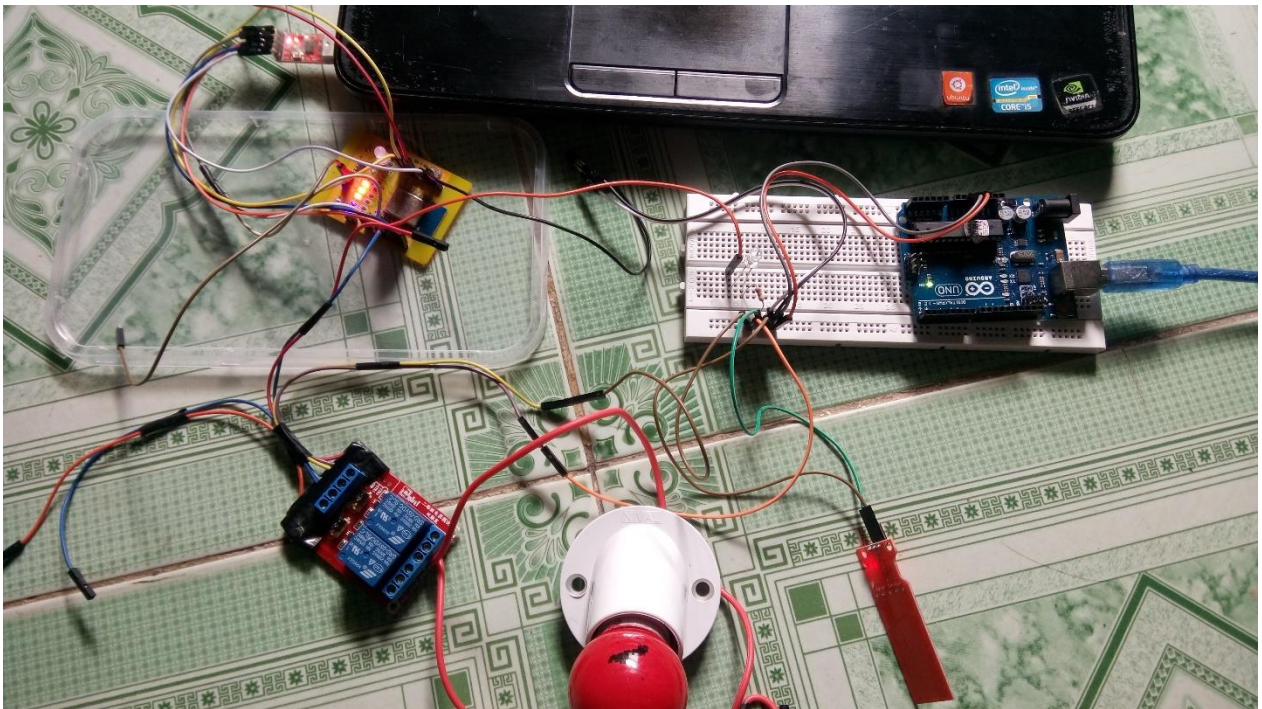
Nguồn 5V: nguồn điện 1 chiều 5V

GND: dây mass

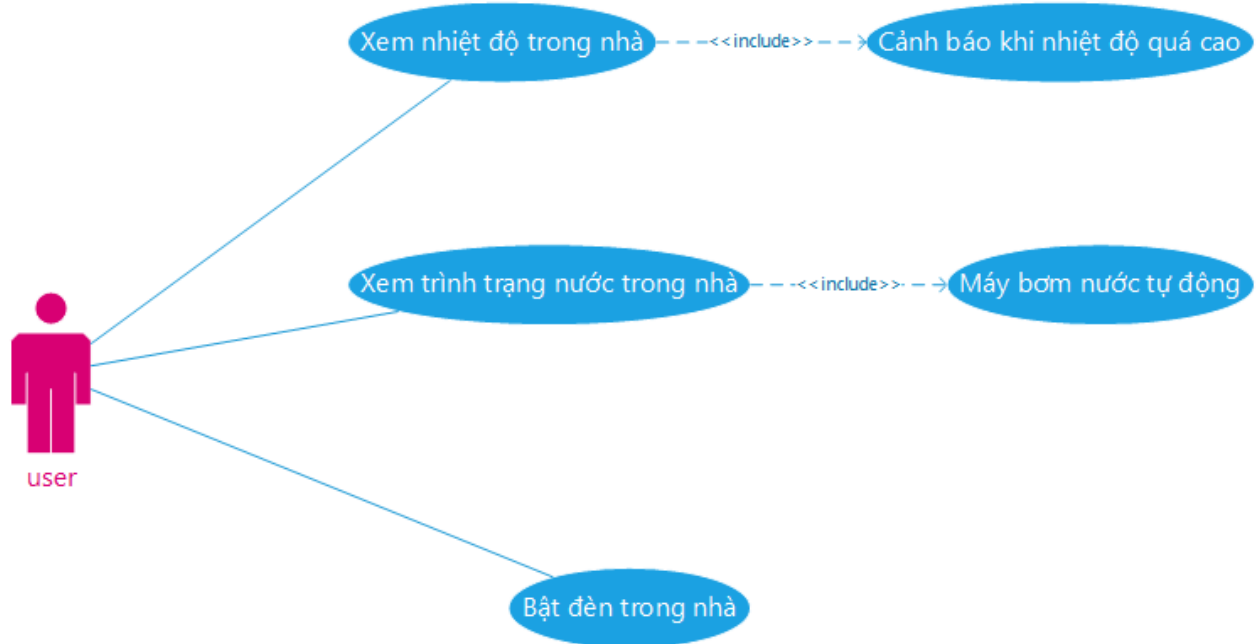
~220V: nguồn điện xoay chiều, điện gia dụng.

Luồng xử lí:

- Cảm biến nước gửi giá trị 1 là có nước và 0 là không có nước vào esp8266, esp8266 sẽ xử lí gửi vào relay khi bằng 1 thì nước đầy sẽ ngắt máy bơm, khi bằng 0 sẽ bật máy bơm, máy bơm sẽ sử dụng nguồn điện 220V (điện gia dụng). Khi nước đầy thì sẽ gửi thông báo cho client biết là nước đầy và máy bơm sẽ ngừng bơm
- Nút bấm thì ấn vào nút bấm sẽ gửi tín hiệu 1 cho esp8266, rồi sẽ xử lí cho đèn. Nếu đèn đang bật thì tắt và ngược lại đang tắt thì sẽ đang mở. ngoài ra người dùng điều khiển đèn từ xa, sẽ gửi lại cho esp xử lí bật tắt đèn.
-
- Cảm biến nhiệt đọc giá trị nhiệt độ trong phòng sẽ gửi về cho esp8266, esp sẽ gửi cho client xem giá trị nhiệt độ.



2. Sơ đồ use-case



a. Xem nhiệt độ trong phòng

Chức năng xem nhiệt độ trong nhà là bao nhiêu sẽ báo cho người dùng biết chính xác nhiệt độ đó, đặc biệt khi nhiệt độ trong phòng tăng quá cao có nguy cơ cháy nổ thì nó sẽ thông báo người dùng liên tục về vấn đề này

Luồng xử lý:

- Nhiệt độ được lấy từ sensor cảm biến nhiệt gửi về cho esp8266
- Esp8266 sẽ gửi qua mạng wifi tới server
- Server sẽ gửi về cho về cho OpenHab xử lý trả về giao diện người dùng

b. Xem lượng nước trong nhà còn nước hay hết nước rồi báo cho người dùng biết như thế nào. Khi mà mực nước thấp thì máy bơm sẽ tự động bơm nước cho đến khi lượng nước đầy máy bơm sẽ tự động ngắt.

- Cảm biến nước sẽ đo xem trong hồ nước có nước hay không, sẽ trả về giá trị cho esp
- Esp xử lý:

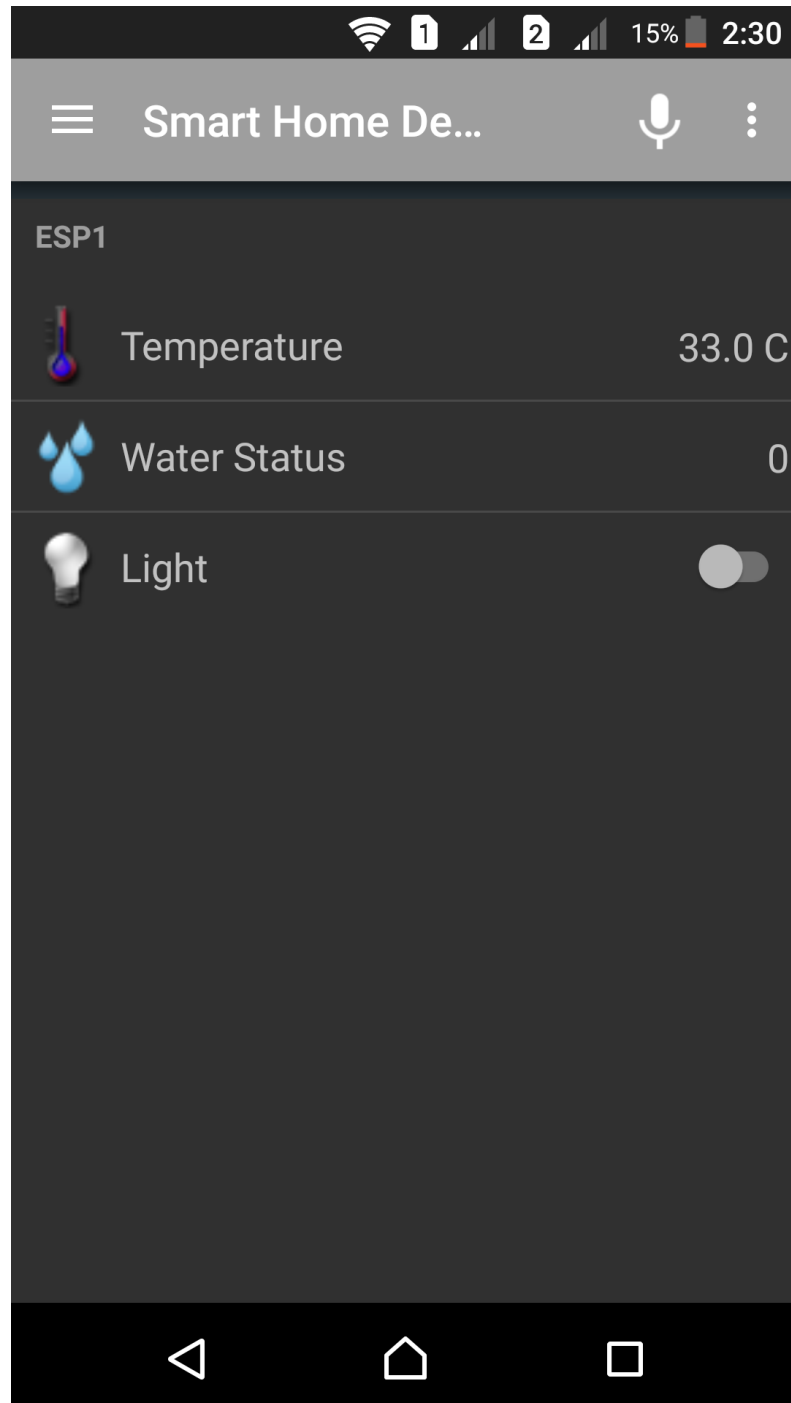
- i. Nếu nước đầy thì sẽ gửi thông báo cho server, server sẽ gửi qua openhab, rồi sẽ hiển thị cho người dùng. Sẽ gửi thông báo đến relay mà điều khiển máy bơm sẽ tắt máy bơm đi
- ii. Nếu nước chưa đầy thì sẽ gửi thông báo cho relay điều khiển máy bơm bơm nước cho đến khi đầy.

c. Bật tắt đèn trong nhà: có 2 cách để bật đèn trong nhà là 1 dùng điều khiển như điện thoại để bật tắt đèn, 2 là dùng công tắc để bật tắt đèn

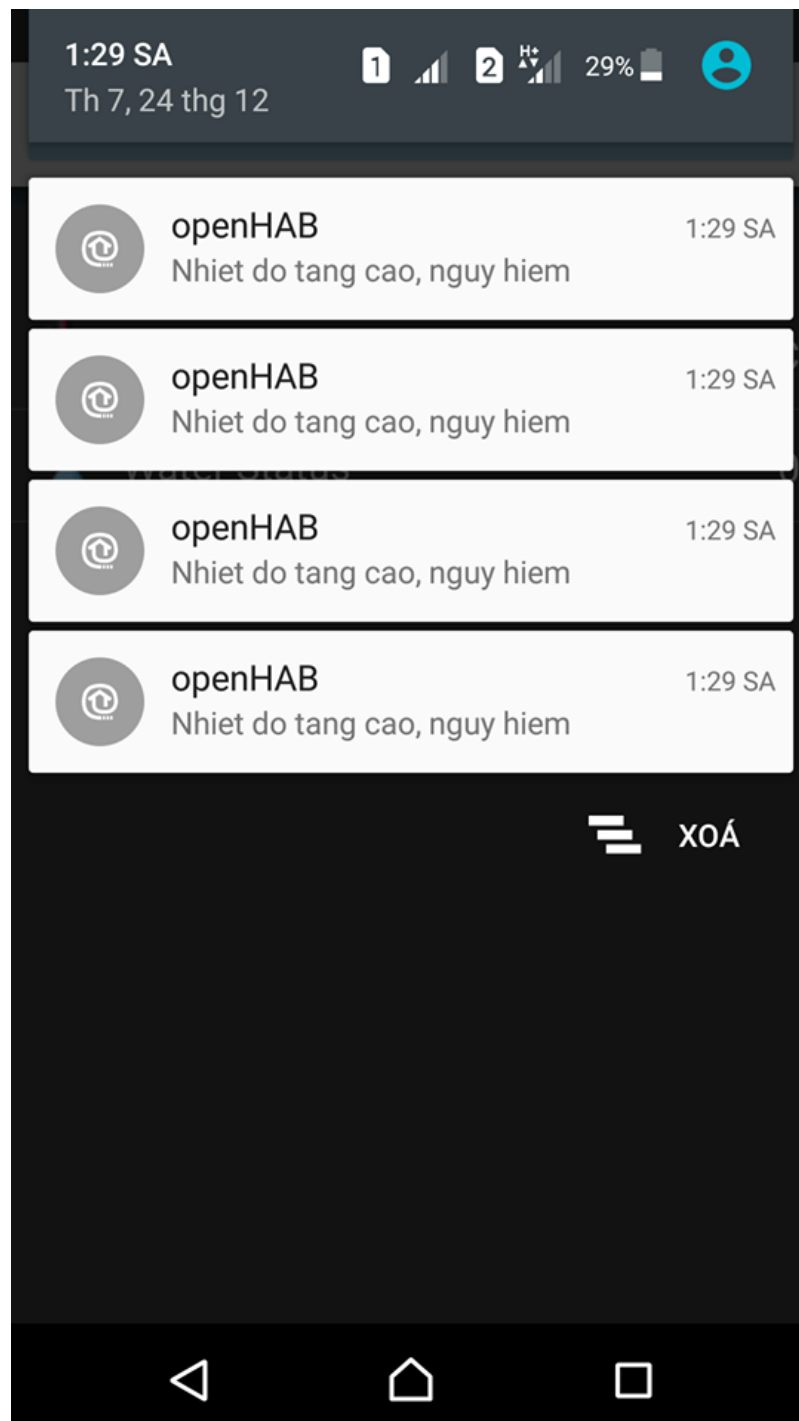
Bật tắt đèn thì:

- Cách 1: Người dùng điều khiển đèn trực tiếp thông qua nút bấm trong nhà, bấm vào nút bấm thì sẽ gửi về cho esp tín hiệu bằng 1. Esp xử lý lấy trạng thái của output đèn xem đang bằng 1 thì esp sẽ xử lý cho nó xuống bằng 0; nếu trạng thái output đèn bằng 0 thì esp sẽ xử lý cho nó bằng 1. (Bằng 0 có nghĩa là không có điện vào thì đèn sẽ tắt và ngược lại). rồi esp sẽ gửi trạng thái đèn cho server, server chuyển cho openhab hiển thị ra giao diện người dùng
- Cách 2: Người dùng có thể điều khiển bằng nút bấm trên giao diện smarthome, khi nhấn vào bật tắt đèn, openhab sẽ gửi lệnh về server, server sẽ gửi về cho esp biết lệnh bật hay tắt đèn. Xử lý xong, esp gửi trạng thái lại cho người dùng.

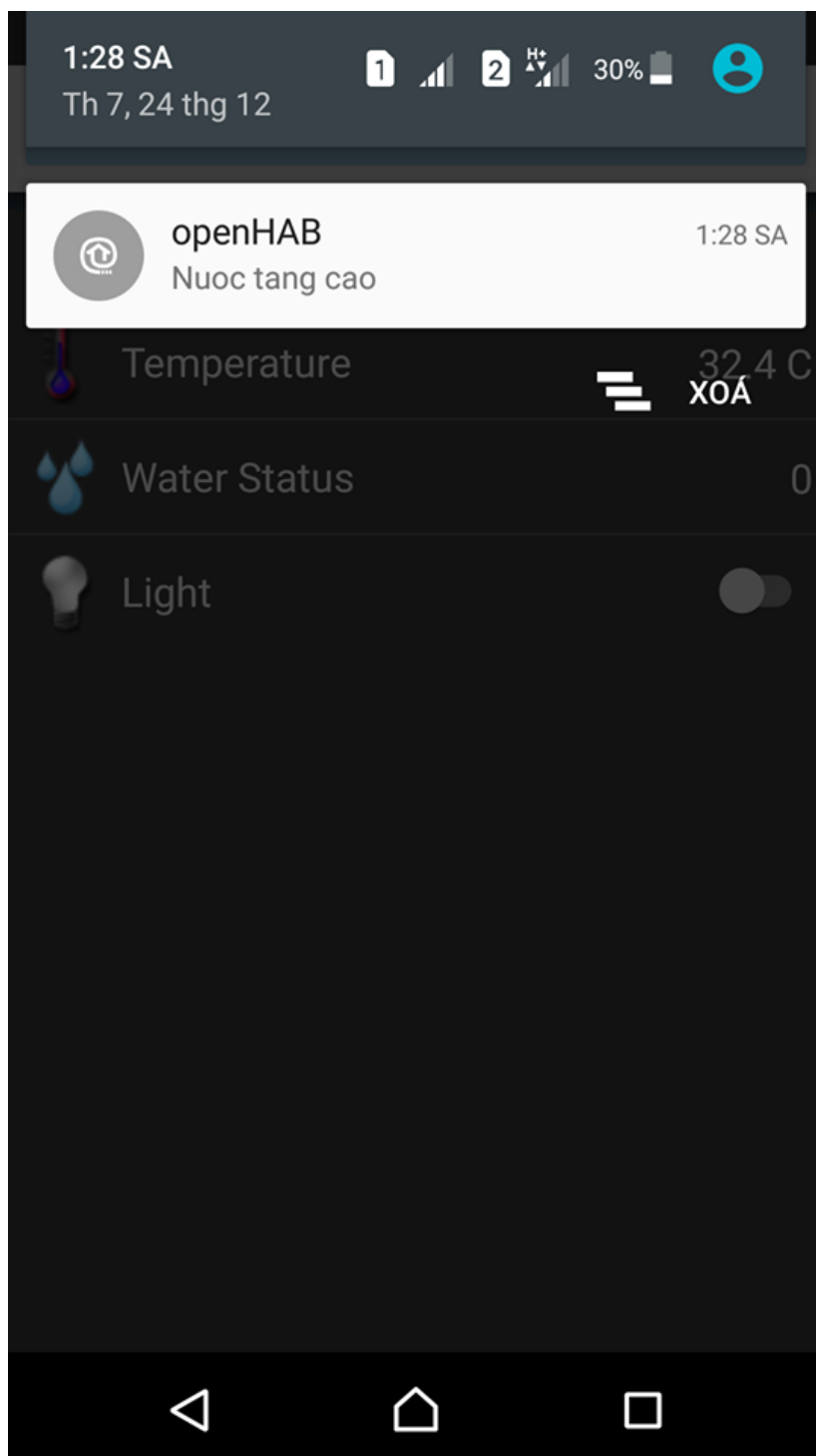
3. Giao diện ứng dụng



Giao diện chính của ứng dụng



Giao diện khi có thông báo nhiệt độ tăng cao.



Giao diện khi có nước đầy.

Phụ lục: Hướng dẫn cài đặt và sử dụng

Hướng dẫn cài đặt Mosquitto

Chuẩn bị:

File cài đặt mosquito

Bản tải ở địa chỉ [Mosquitto download](#). Nhớ tải file cho win32 nhé.

Chú ý: với file cài đặt là win32 chỉ sử dụng cho HĐH từ Windows vista trở lên.

Thư viện để cài đặt

[pThreads](#)

Tải file pthreadVC2.dll

[OpenSSL](#)

Tải về cài đặt và sử dụng 2 file thư viện là: libeay32.dll và ssleay32.dll

Cài đặt:

Cài đặt file Mosquitto-1.4.8-install-win32.exe

Copy file thư viện pthreadVC2.dll vừa tải và 2 file thư viện libeay32.dll và ssleay32.dll trong thư mục cài đặt của OpenSSL

Cài lại mosquito (vẫn sử file mosquito-1.4.8-install-win32.exe)












Check xem mosquito đã chạy chưa: mở CMD và gõ

```
1 | netstat -an
```

Hướng dẫn cài openHab:

Cài như bình thường <http://www.openhab.org/>

Khi cài đặt xong sẽ vào thư mục configurations để code tùy chỉnh giao diện code, controller của nó.

	items	21/12/2016 11:38 ...	File folder	
	persistence	15/12/2016 3:51 PM	File folder	
	rules	23/12/2016 10:25 ...	File folder	
	scripts	21/12/2016 11:38 ...	File folder	
	sitemaps	15/12/2016 4:05 PM	File folder	
	transform	21/12/2016 11:38 ...	File folder	
	logback	22/05/2016 9:24 PM	XML Document	3 KB
	logback_debug	22/05/2016 9:24 PM	XML Document	3 KB
	openhab.cfg	23/12/2016 10:10 ...	CFG File	83 KB
	openhab_default.cfg	22/05/2016 9:24 PM	CFG File	83 KB
	users.cfg	22/05/2016 9:24 PM	CFG File	1 KB

Chú ý các phần thiết yếu:

Mở file openhan.cfg thêm vào 2 dòng này:

```
# URL to the MQTT broker, e.g. tcp://localhost:1883 or ssl://localhost:8883
#mqtt:<broker>.url=tcp://<host>:1883
mqtt:server.url=tcp://localhost:1883
mqtt:server.retain=true
```

Items : chứa các khai báo và các link liên kết của item đó. Trong item có file default.items có dạng như sau

```
Group All
Group Esp1 "Smart Home" <sofa> (All)

//Webberry
Number Temperature "Temperature [%1f C]" <temperature> (Web) {mqtt="<[server:/Esp1/Temperature:state:default]"}
Number Water "Water Status [%0f]" <water> (Web) {mqtt="<[server:/Esp1/Water:state:default]"}
```

Khai báo các đối tượng: tên, thuộc tính, link publish....

Rules: những luật giống như controller trong web, khi nhận được giá trị xem giá trị đó có xử lý gì không. Giống như cái dưới đây, khi mà nhiệt độ tăng lên quá 40 độ sẽ gửi thông báo về client

```

import org.openhab.core.library.types.*
import org.openhab.model.script.actions.*

rule "Temperature up"
    when
        Item Temperature received update
    then
        if(Temperature.state>40){
            sendBroadcastNotification("Nhiet do tang cao, nguy hiem")
        }
    end
end

```

rule “Temperature up” tên của rule đó

when khi mà cái gì xảy ra

then là sẽ làm như thế nào khi ở trên xảy trong ví dụ là khi nhiệt độ update sẽ xử lí như thế nào

Sitemaps: là nơi chứa các item để hiển thị ra bên ngoài giao diện

```

sitemap default label="Smart Home Demo"
{
    Frame label="Esp1" {
        Text item=Temperature
        Text item=Water
        Switch item=Light
    }
}

```

Frame: là 1 vùng chứa

Text là kiểu dữ liệu sẽ hiển thị

Temperature: là item bên Item folder khai báo

Switch là lựa chọn trong 2 cái: bật tắt bóng đèn.