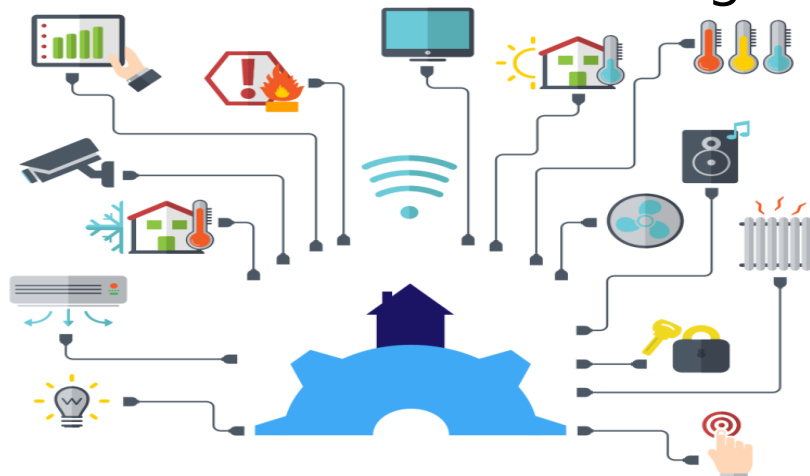




LẬP TRÌNH IOT NÂNG CAO

BÀI 6: CÁC DỰ ÁN NHÀ THÔNG MINH

1. Tổng quan về nhà thông minh
2. Chức năng hiển thị thời gian thực
3. Chức năng điều khiển công tắc điện
4. Chức năng hiển thị thông số từ cảm biến
5. Chức năng phát hiện chuyển động
6. Chức năng gửi thông báo đến điện thoại
7. Chức năng hiển thị Camera IP
8. Chức năng dự báo thời tiết
9. Điều khiển thiết bị qua đám mây
10. Thiết kế giao diện người dùng



1. TỔNG QUAN VỀ SMARTHOME

- ❑ Smarthome là hệ thống nhà được lắp đặt các thiết bị điện, điện tử có tác dụng tự động hoá hoàn toàn hoặc bán tự động, thay thế con người trong thực hiện một hoặc một số thao tác quản lý, điều khiển.
- ❑ Trong căn nhà thông minh, các thiết bị đều gắn các bộ điều khiển điện tử có thể kết nối với internet và điện thoại di động, cho phép chủ nhân điều khiển vật dụng từ xa hoặc lập trình cho thiết bị ở nhà hoạt động theo lịch, nhằm mục đích tập trung vào lợi ích của người sử dụng, tối đa hiệu suất của người sử dụng và cho phép tối đa hóa hiệu quả quản lý ngôi nhà với chi phí tối thiểu.

1. TỔNG QUAN VỀ SMARTHOME

Một ngôi nhà được gọi là nhà thông minh cần thực hiện được các chức năng cơ bản sau:

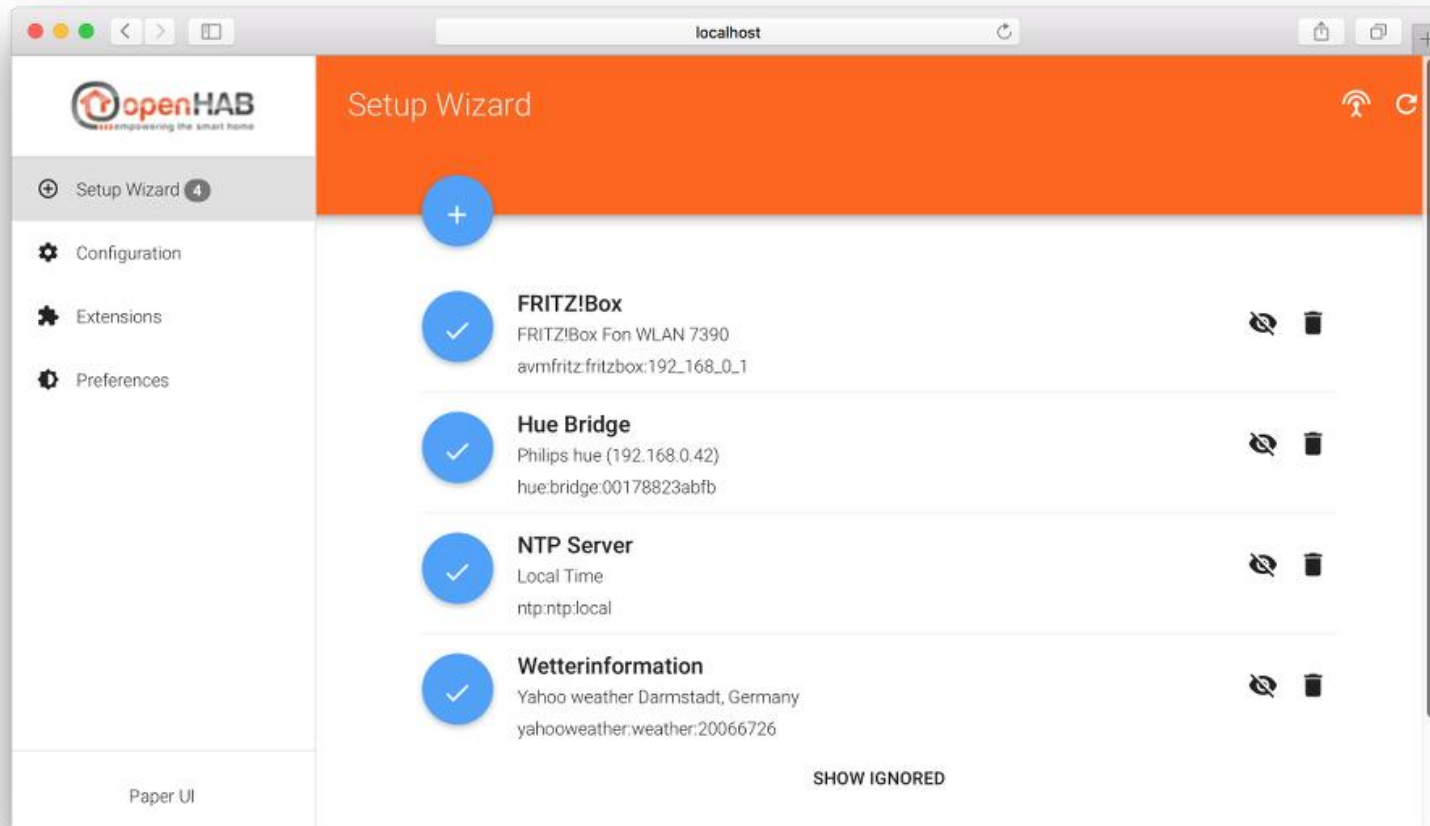
- ❑ *Hệ thống chiếu sáng*: Lập trình và cài đặt thời gian chiếu sáng, bật tắt đèn với từng ngữ cảnh cụ thể.
- ❑ *Hệ thống rèm cửa tự động*: Tùy chỉnh thời gian kéo thả rèm, điều khiển nhiều rèm cùng một lúc khi trong căn nhà hoặc cho dù ở bất cứ đâu.
- ❑ *Hệ thống điều hòa không khí*: Điều khiển nhiệt độ và toàn bộ điều hòa chỉ trong một chạm.
- ❑ *Hệ thống bình nóng lạnh*: Điều khiển bình nước nóng bật tắt theo thời gian quy định cho dù người dùng ở bất cứ đâu.

1. TỔNG QUAN VỀ SMARTHOME

- ❑ *Hệ thống âm thanh đa vùng:* Điều khiển hệ thống loa và âm li trong căn nhà bằng Smartphone, đồng bộ và tương thích với nhiều hệ điều hành Smartphone khác biệt.
- ❑ *Hệ thống kiểm định môi trường:* Tính toán các chỉ số môi trường: Độ ẩm, ánh sáng, không khí và có những phương hướng tự động cụ thể.
- ❑ *Hệ thống an ninh giám sát:* Giám sát báo động qua các cảm biến chuyển động và giám sát hình ảnh qua Camera.
- ❑ *Hệ thống ánh sáng đa sắc màu:* Điều khiển hệ thống màu sắc đèn điện với hơn 16 triệu màu.
- ❑ *Hệ thống đồng bộ:* Chỉ với "1 chạm", người dùng có thể dễ dàng kích hoạt toàn bộ các hệ thống trên mà thời gian cài đặt chưa tới 10s.

1. Cài đặt NTP add-ons trên OpenHAB2

Extension/addons/NTP Server → install



2. Cài đặt NTP add-ons trên OpenHAB2

Thiết lập thông số

Things

```
ntp:ntp:demo [    hostname="nl.pool.ntp.org",
refreshInterval=60, refreshNtp=30 ]
```

Items

```
DateTime Date "Ngày giờ hiện tại [%1$tA,
%1$td.%1$tm.%1$tY %1$tH:1$tM]" % { channel =
"ntp:ntp:demo:dateTime" }
```

Sitemaps

Text item=Date

3. Kết nối thiết bị với OpenHAB server

```
#include <ESP8266WiFi.h>
#include <OneWire.h>
#include <DallasTemperature.h>

#Định nghĩa
const char* server =
"192.168.137.101";
#String apiKey ="your API key here";
const char* MY_SSID =
"MyPublicWiFi1";
const char* MY_PWD = "12345678";
int sent = 0;

void setup() {
  Serial.begin(115200);
  connectWifi();
}
```

```
void loop() {

}

void connectWifi()
{
  Serial.print("Connecting to "+*MY_SSID);
  WiFi.begin(MY_SSID, MY_PWD);
  while (WiFi.status() != WL_CONNECTED) {
    delay(1000);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("Connected");
  Serial.println("");
} //end connect
```


4. Khai báo chân điều khiển thiết bị

```
const int switchPin1 = D1;
const int switchPin2 = D4;

char const* switchTopic1 =
"myVegPro/switch1";
char const* switchTopic2 =
"myVegPro/switch2";

int enable1 = D6;    // enable reading
sensor A
int enable2 = D7;    // enable reading
sensor B
```

```
void setup() {
    //initialize the switch as an output and set to
    LOW (off)
    pinMode(switchPin1, OUTPUT); // Relay Switch 1
    digitalWrite(switchPin1, LOW);

    pinMode(switchPin2, OUTPUT); // Relay Switch 2
    digitalWrite(switchPin2, LOW);
    ...
}
```

5. Bật tắt thiết bị trong callback

```
void callback(char* topic, byte*
payload, unsigned int length) {
```

```
    //convert topic to string to make it
easier to work with
```

```
    String topicStr = topic;
    //EJ: Note: the "topic" value gets
overwritten everytime it receives
confirmation (callback) message from
MQTT
```

```
    //Print out some debugging info
    Serial.println("Callback update.");
    Serial.print("Topic: ");
    Serial.println(topicStr);
    ....
```

```
if (topicStr == switchTopic1)
{
```

```
    //turn the switch on if the payload is '1' and
publish to the MQTT server a confirmation
message
```

```
    if(payload[0] == '1'){
        digitalWrite(switchPin1, HIGH);
    }
```

```
    //turn the switch off if the payload is '0' and
publish to the MQTT server a confirmation
message
```

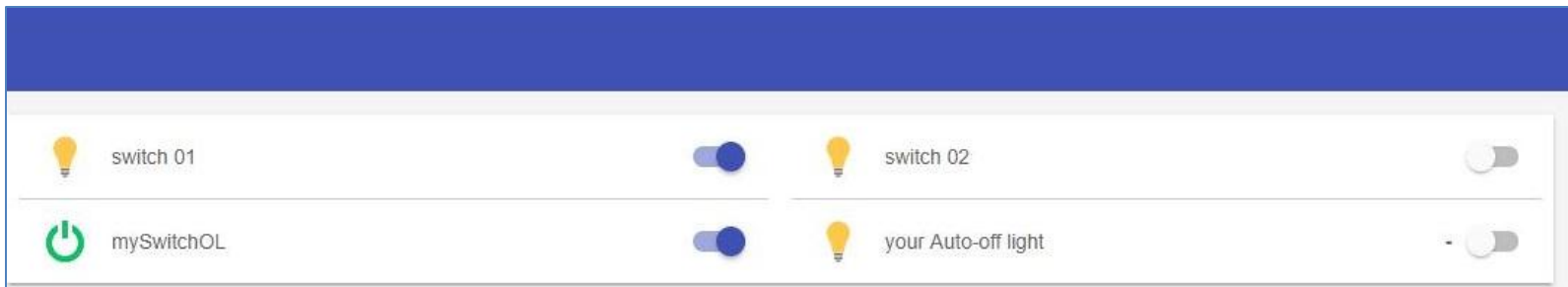
```
    else if (payload[0] == '0'){
        digitalWrite(switchPin1, LOW);
    }
}
```

Items

- ❑ Switch mySwitch01 "switch 01" <light> {
 mqtt=">[localbroker:myVegPro/switch1:command:ON:1],>[localbroker:myVegPro/switch1:command:OFF:0]" }
- ❑ Switch mySwitch02 "switch 02" <light> {
 mqtt=">[localbroker:myVegPro/switch2:command:ON:1],>[localbroker:myVegPro/switch2:command:OFF:0]" }
- ❑ Switch mySwitchOL
 {mqtt="<[localbroker:/office/light:command:ON:1],<[localbroker:/office/light:command:OFF:0]" }

Sitemap

```
sitemap demo label="HỆ THỐNG NHÀ THÔNG  
{  
  Switch item=mySwitch01  
  Switch item=mySwitch02  
  Switch item=mySwitchOL  
  Switch item=GarageDoorClosedSensor  
}
```



1. Thiết lập setup() và đọc các thông số từ các cảm biến

```
#include <ESP8266WiFi.h>
#include <ArduinoOTA.h>
#include<dht.h>
dht DHT;

int sensorPin = A0;
int enable1 = D6;

void setup() {
  int chk = DHT.read11(DHT11_PIN);
  // declare the enable and ledPin as an OUTPUT:
  pinMode(enable1, OUTPUT);//...
  client.subscribe("myDHT11/temperature")
;
//...
}
```

```
// read the value from sensor 1
digitalWrite(enable1, HIGH);
  delay(1000);
  sensorValue1 =
analogRead(sensorPin);
  sensorValue1 =
constrain(sensorValue1, 485, 1023);
  sensorValue1 = map(sensorValue1,
485, 1023, 100, 0);
  //sensorValue1 = 100 - (sensorValue1 *
100);
  digitalWrite(enable1, LOW);

// read the value from sensor 2
//...
// read the value from sensor 3
//...
```

2. Gửi thông số cảm biến lên OpenHAB

```
void Loop()  
...  
client.publish("myDHT11/sensorMoisture", messageSen1);  
  
//publish sensor 2  
//...  
//publish sensor 3  
//...  
} //end loop
```

Items

```

Number      my_sensorTemp      "Nhiệt độ phòng [%.2f %%]"
<soil_moi>                                     {
mqtt="<[localbroker:myDHT11/sensorTemp:state:default]" }
Number      my_sensorMoisture    "Độ ẩm [%.2f %%]"
<soil_moi>                                     {
mqtt="<[localbroker:myDHT11/sensorMoisture:state:default]" }
Number      my_sensorLight      "Cường độ ánh sáng [%.2f %%]"
<weather>                                     {
mqtt="<[localbroker:myDHT11/sensorLight:state:default]" }
  
```

...

Sitemap

```

Text item = my_sensorMoisture
Text item = my_sensorTemp
Text item = my_sensorLight
  
```

1. Phương thức hoạt động

- ☐ Cảm biến phát hiện chuyển động sẽ truyền tín hiệu đến bộ điều khiển trung tâm.
- ☐ khi phát hiện có sự di chuyển vào vùng cảm biến.
- ☐ Bộ điều khiển trung tâm sẽ gửi thông tin đến công tắc kết nối với bóng đèn để bật/tắt theo tín hiệu từ cảm biến phát hiện chuyển động.

2. Đọc thông số chuyển động

```
char const* motionTopic = "myMotion/motion1";
void loop() {
  ...
  //motion
  int inputState = digitalRead(INPUT_PIN);

  if (inputState != oldInputState)
  {
    Serial.print(" motion sensor: " );
    Serial.println(inputState);
    if (inputState == 1)
      client.publish(motionTopic, "1");
    oldInputState = inputState;
  }
  ...
} //end loop()
```

5. CHỨC NĂNG PHÁT HIỆN CHUYỂN ĐỘNG

Nhận tín hiệu từ MQTT khi có chuyển động gửi tín hiệu bằng 1 là bật công tắc thông qua cảm biến chuyển động.

Items

```
Switch      myMotion    "chuyen dong"    <light>
{
    mqtt=">[localbroker:myMotion/motion1:command:ON:1],>[localbroker:myMotion/motion1:command:OFF:0]"
}
```

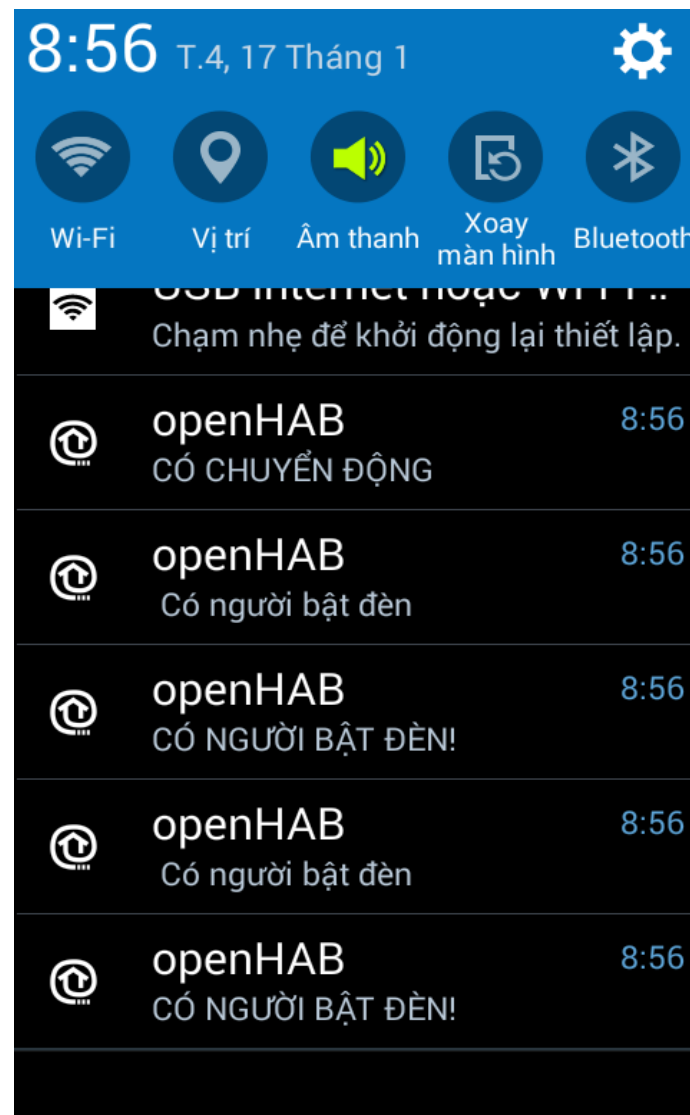
- ❑ khi công tắc bật lên thì sau 10 giây sẽ tự động tắt:
\conf\rules\chuyendong.rules

Rules

```
rule "Phát hiện chuyển động"
when
    Item myMotion1 changed from OFF to ON
Then
    sendBroadcastNotification("CÓ CHUYỂN ĐỘNG")
    var int seconds = 10
    MyTimer=createTimer(now.plusSeconds(seconds))[ |
        myMotion.sendCommand(OFF) ]
end
```

6. CHỨC NĂNG GỬI THÔNG BÁO ĐẾN ĐIỆN THOẠI

- ❑ Các thông báo sẽ được gửi đến cho thiết bị di động khi có tác động ví dụ có người bật đèn, có chuyển động và cảnh báo khi nhiệt độ quá cao. để thực hiện ta cần phải tạo file rules.



❑ Luật gửi thông báo lên điện thoại

rule “Gửi thông báo nhiệt độ qua điện thoại”

when

Item DHT11Temp changed

then

var int myTemp = (DHT11Temp .state as

DecimalType).intValue()

if (myTemp > 40)

sendBroadcastNotification("Nhiệt độ phòng quá cao!")

End

rule “Chuyển động và thông báo”

when

Item **myMotion** changed from OFF to ON







then

sendBroadcastNotification(“Có chuyển động”)

Sitemaps

Image url="http://192.168.173.1:8081/cam_1.jpg"
label="Camera 2" refresh=50



 Phát sóng wifi	<input type="checkbox"/>	 TÌ VI	<input type="checkbox"/>
 Đèn cầu thang	<input type="checkbox"/>	 Máy điều hòa	<input type="checkbox"/>
 Nhiệt độ hiện tại	- °C	 Độ ẩm hiện tại	-%

8. CHỨC NĂNG DỰ BÁO THỜI TIẾT

- ❑ Thu thập dữ liệu thời tiết hiện tại và dự báo từ các nhà cung cấp khác nhau với API thời tiết miễn phí.
- ❑ Bạn cũng có thể hiển thị dữ liệu thời tiết với bố cục và biểu tượng html có thể tùy chỉnh cao.
- ❑ Các bước thiết lập:
 - ❖ Cài đặt Weather binding add-on
 - ❖ Thiết lập tài khoản Config
 - ❖ Tạo items, sitemap

❑ Cài đặt Weather binding add-on

⚙ Add-ons

⚙ Preferences

V	binding-powermax1 - 1.9.0.SNAPSHOT	INSTALL
V	Vitotronic Binding binding-vitotronic - 2.0.0.SNAPSHOT	INSTALL
W	Waterkotte Ecotouch Binding binding-ecotouch1 - 1.9.0.SNAPSHOT	INSTALL
W	Weather Binding binding-weather1 - 1.9.0.SNAPSHOT	INSTALL
W	Wemo Binding binding-wemo - 0.9.0.SNAPSHOT	INSTALL

❑ Thiết lập tài khoản weather.cfg

```

weather.cfg x
1 # The apikey for the different weather providers, at least one must be specified
2 # Note: Hamweather requires two apikeys: client_id=apikey, client_secret=apikey2
3 #apikey.ForecastIo=
4 #apikey.OpenWeatherMap=
5 #apikey.WorldWeatherOnline=
6 #apikey.Wunderground=
7 #apikey.Hamweather=
8 #apikey2.Hamweather=
9 #apikey.Meteoblue=
10
11 # location configuration, you can specify multiple locations
12 #location.<locationId1>.name=
13 #location.<locationId1>.latitude=      (not required for Yahoo provider)
14 #location.<locationId1>.longitude=    (not required for Yahoo provider)
15 #location.<locationId1>.woeid=        (required for Yahoo provider)
16 #location.<locationId1>.provider=
17 #location.<locationId1>.language=
18 #location.<locationId1>.updateInterval=
19
20 #location.<locationId2>.name=
21 #location.<locationId2>.latitude=      (not required for Yahoo provider)
22 #location.<locationId2>.longitude=    (not required for Yahoo provider)
23 #location.<locationId2>.woeid=        (required for Yahoo provider)
24 #location.<locationId2>.provider=
25 #location.<locationId2>.language=
26 #location.<locationId2>.updateInterval=

```

8. CHỨC NĂNG DỰ BÁO THỜI TIẾT

❑ Thiết lập tài khoản weather.cfg

```
weather:apikey.OpenWeatherMap=766967gdfgdfgs9g76dsf
g5ds76g521
```

```
weather:location.home-OWM.latitude=47.8011
```

```
weather:location.home-OWM.longitude=13.0448
```

```
weather:location.home-OWM.provider=OpenWeatherMap
```

```
weather:location.home-OWM.language=de
```

```
weather:location.home-OWM.updateInterval=10
```

```
Number Temperature "Temperature [%.2f °C]"
```

```
{weather="locationId=home-
FIO,type=temperature,property=current"}
```

```
Number Humidity "Humidity [%d %%]"
```

```
{weather="locationId=home-
OWM,type=atmosphere,property=humidity"}
```

8. CHỨC NĂNG DỰ BÁO THỜI TIẾT

Một số thông tin trong thời tiết hỗ trợ:

- ☐ **Khí quyển:** độ ẩm, khả năng hiển thị, khả năng hiển thị, áp suất, áp suất, áp suất Xu hướng, ozone, uv Index
- ☐ **Mây:** phần trăm
- ☐ **Điều kiện:** văn bản, thời gian quan sát, id, lastUpdate, commonId
- ☐ **Lượng mưa:** mưa, tuyết, xác suất
- ☐ **Nhiệt độ:** hiện tại, min, max, cảm thấy, điểm sương, minax
- ☐ **Gió:** tốc độ, hướng, độ, cơn gió, lạnh
- ☐ ...

Items

- ❑ String Temperatur_MinMax "Min/Max [%s °C]"
{weather="locationId=home,forecast=0,type=temperature,property=minMax"}
- ❑ > Temperatur_MinMax state updated to 8.91/17.25
- ❑ MinMax "Min/Max [%s °C]"
{weather="locationId=home,forecast=0,type=temperature,property=minMax,scale=0"}
- ❑ > Temperatur_MinMax state updated to 9/17
- ❑ String Temperatur_MinMax "Min/Max [%s °C]"
{weather="locationId=home,forecast=0,type=temperature,property=minMax,roundingMode=down,scale=0"}
- ❑ ...

Sitemap

Text label="Thời tiết" icon="rain" {

 Text item=Station_Name

 Text item=Rain

 Text item=Snow

 Text item=Humidity123

 Text item=Wind_Speed

 Text item=Condition

 Textitem=Clouds

Textitem=Weather_Temperature12345

valuecolor=[>25="orange",>15="green",>5="orange",<=5="blue"]

Text item=Weather_Temperature_F

valuecolor=[>25="orange",>15="green",>5="orange",<=5="blue"]

Text item=Date

valuecolor=[>25="orange",>15="green",>5="orange",<=5="blue"]

8. CHỨC NĂNG DỰ BÁO THỜI TIẾT

	Station Name	Buon Ma Thuot		Rain	0.00 mm/h
	Snow	0.00 mm/h		Humidity	95 %
	Windspeed	6.73 km/h		Condition	mây cụm
	Clouds	64 %		Outside Temperature	21.8 °C
	Temperature_F	71.2 °F		Date	Tuesday, 07.11.2017 07:57:10

Items

- ❑ Group Demo_weather
- ❑ Number Dm_Weather_Chart_Period "Chart Period"
- ❑ Number Dm_Temperature "Temperatura [% .1f °C]" <temperature> (Demo_weather) {weather = "locationId=1586896, type=temperature, property = current"}
- ❑ Number Temp_Feel "Temperature feel [% .2f °C]" (Demo_weather) {weather = "locationId = 1586896, type = temperature, property=feel"}

Thời gian cập nhật thông số **rrd4j**

Strategies {

 everyMinute : "0 * * * * ?"

 everyHour : "0 0 * * * ?"

 everyDay : "0 0 0 * * ?"

 everyWeek : "0 0 0 * * ?"

 default = everyChange

}

Items {

 Demo_weather*,Demo_weather*:

 strategy=everyChange,everyMinute,restoreOnStartup

}

sitemap

```

Frame{
    Switch    item=    Dm_Weather_Chart_Period    label="Biểu đồ"
    icon="line"
    mappings=[0="hour",1="Day",2="week",3="Month",4="year"]
    Chart    item=Demo_weather    period=h    refresh=600
    visibility=[Dm_Weather_Chart_Period
    ==0,Dm_Weather_Chart_Period=="Uninitialized"]
    Chart    item=Demo_weather    period=D    refresh=600
    visibility=[Dm_Weather_Chart_Period==1]
    Chart    item=Demo_weather    period=W    refresh=600
    visibility=[Dm_Weather_Chart_Period==2]
    Chart    item=Demo_weather    period=M    refresh=600
    visibility=[Dm_Weather_Chart_Period==3]
    Chart    item=Demo_weather    period=Y    refresh=600
    visibility=[Dm_Weather_Chart_Period==4]
}
    
```

8. CHỨC NĂNG DỰ BÁO THỜI TIẾT BIỂU ĐỒ THỜI TIẾT



9. ĐIỀU KHIỂN THIẾT BỊ QUA Đám Mây

- ❑ Trình kết nối đám mây openHAB cho phép kết nối thời gian chạy openHAB cục bộ với một đám mây OpenHAB ở xa, chẳng hạn như myopenHAB.org, là một ví dụ của dịch vụ đám mây openHAB do Tổ chức OpenHAB tổ chức.
- ❑ Dịch vụ đám mây openHAB (và do đó kết nối với nó) rất hữu ích cho các trường hợp sử dụng khác nhau:
- ❑ Nó cho phép truy cập từ xa đến các thể hiện openHAB cục bộ mà không cần phải phơi bày các cổng vào Internet hoặc cần thiết lập VPN phức tạp.
- ❑ Nó hoạt động như một kết nối với Google Cloud Messaging (GCM) và Apple Push Notifications (APN) để thúc đẩy thông báo cho các ứng dụng điện thoại di động.
- ❑ Nó mang lại khả năng tích hợp với các dịch vụ yêu cầu xác thực OAuth2 đối với máy chủ web, chẳng hạn như IFTTT hoặc Amazon Alexa Skills.

9. ĐIỀU KHIỂN THIẾT BỊ QUA ĐÁM MÂY

- ❑ Để xác thực với OpenHAB Cloud, thời gian chạy openHAB cục bộ của bạn sẽ tạo ra hai giá trị cần được nhập vào cài đặt tài khoản của bạn trong dịch vụ đám mây openHAB.
- ❑ Một trong những đầu tiên là mã định danh duy nhất, cho phép xác định thời gian chạy của người dùng chính là tên người dùng để xác thực đám mây.
- ❑ Thứ hai là một khóa bí mật ngẫu nhiên với chức năng là mật khẩu.

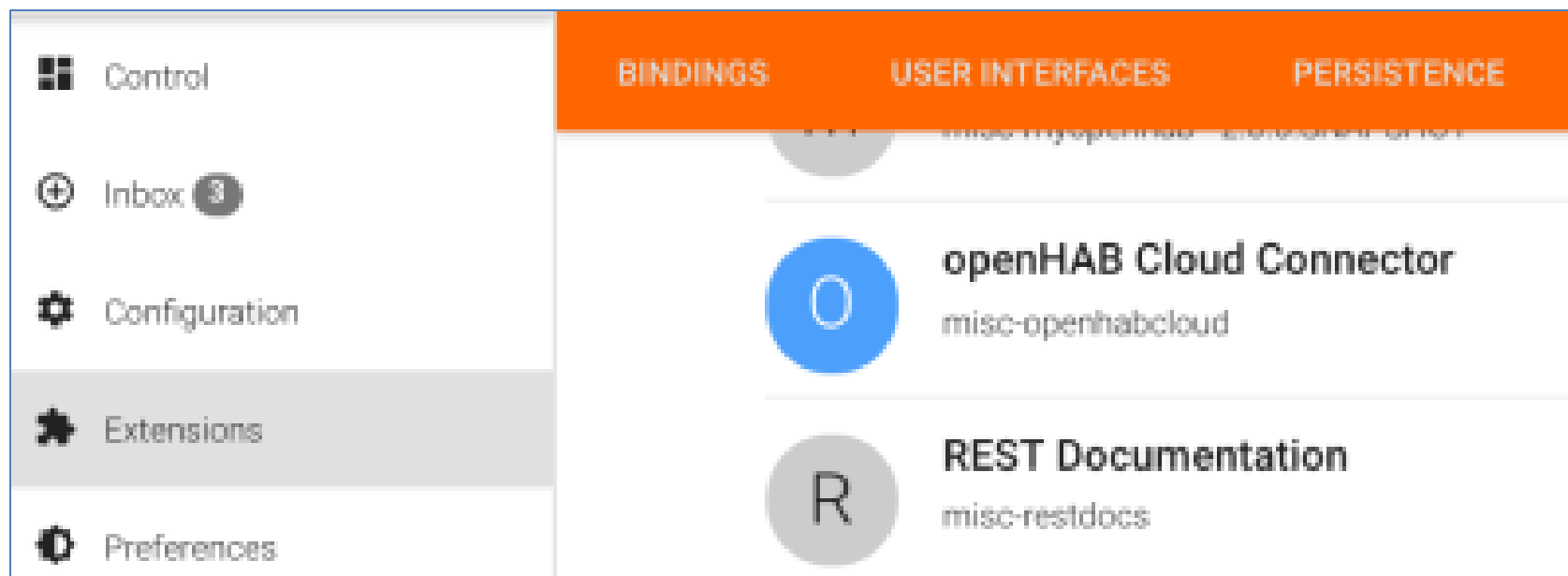
9. ĐIỀU KHIỂN THIẾT BỊ QUA Đám Mây

- ❑ Cả hai giá trị được ghi vào hệ thống tệp cục bộ khi cài đặt. Nếu bạn mất các tệp này vì một số lý do nào đó, openHAB sẽ tự động tạo ra những tệp mới.
- ❑ Ta cần phải cấu hình lại UUID và mật khẩu trong dịch vụ đám mây openHAB.

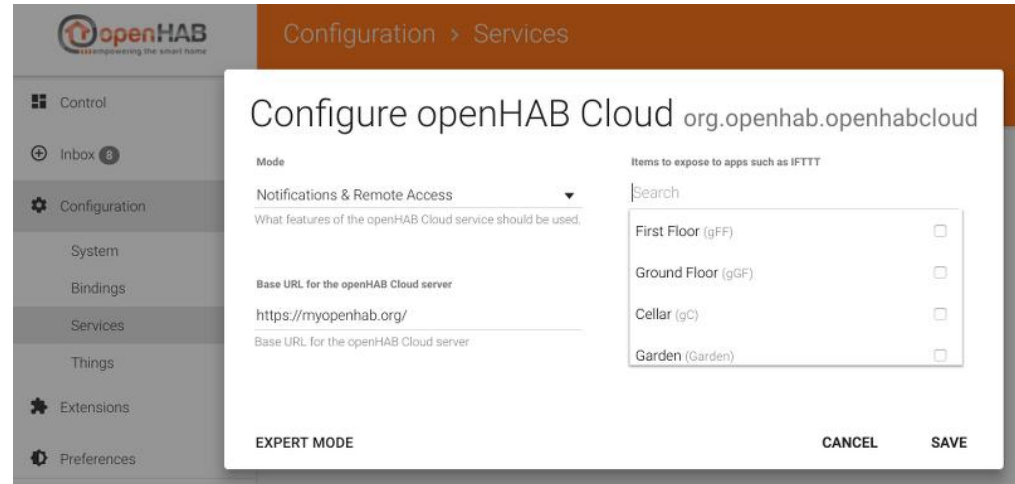
Tập tin	Vị trí	Cài đặt APT
UUID	\$ {openhab.home} / webapps / static / uuid	/ usr / share / openhab / webapps / static / uuid
Mật khẩu	\$ {openhab.home} / webapps / static / secret	/ usr / share / openhab / webapps / tĩnh / bí mật

1. Cài đặt OpenHAB Cloud Connector addon

Trong mục PaperUI/ Misc, chọn install



2. Thiết lập hệ thống



conf/services/openhabcloud.cfg

```
##### openHAB Cloud Connector #####

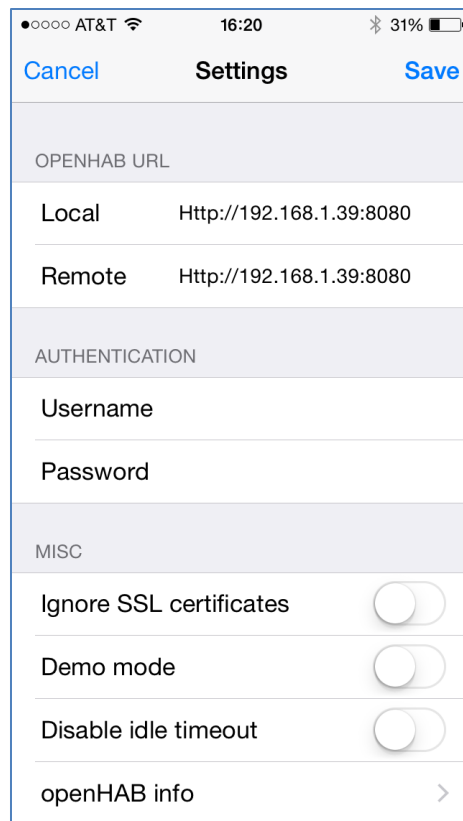
# The URL of the openHAB Cloud service to connect to.
# Optional, default is set to the service offered by the openHAB Foundation
# (https://myopenhab.org/)
#baseUrl=

# Defines the mode in which you want to operate the connector.
# Possible values are:
# - notification: Only push notifications are enabled, no remote access is allowed.
# - remote: Push notifications and remote access are enabled.
# Optional, default is 'remote'.
#mode=

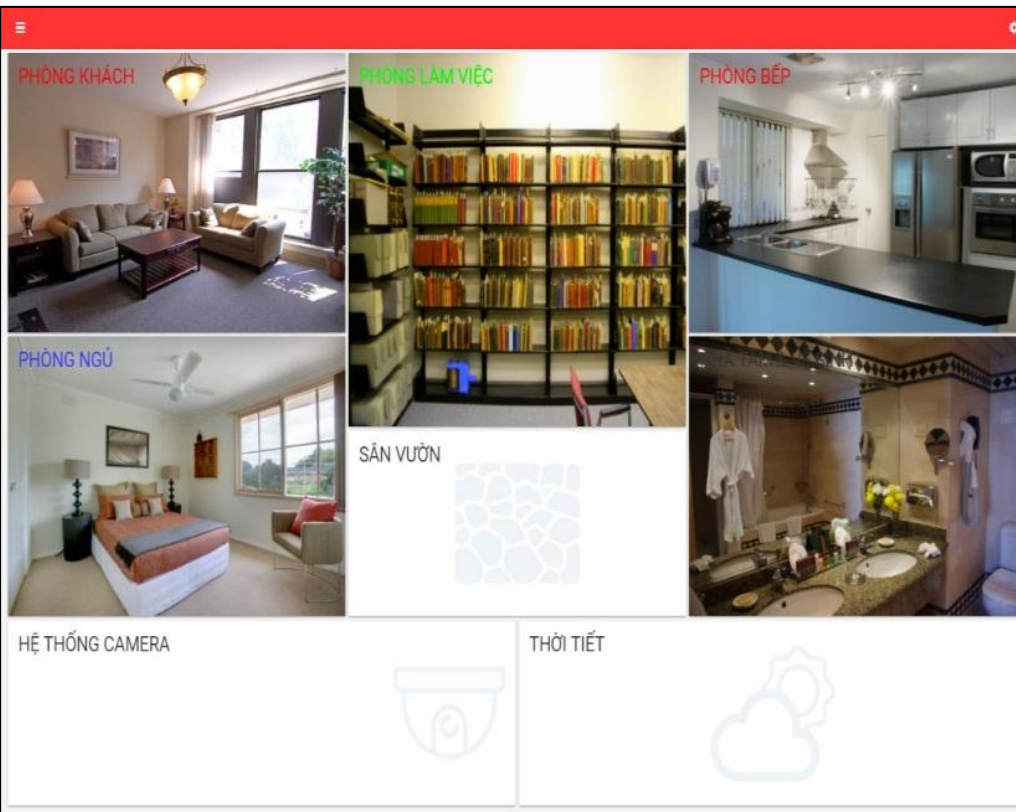
# A comma-separated list of items to be exposed to external services like IFTTT.
# Events of those items are pushed to the openHAB Cloud and commands received for
# these items from the openHAB Cloud service are accepted and sent to the local bus.
# Optional, default is an empty list.
#expose=
```

3. Kết nối từ điện thoại

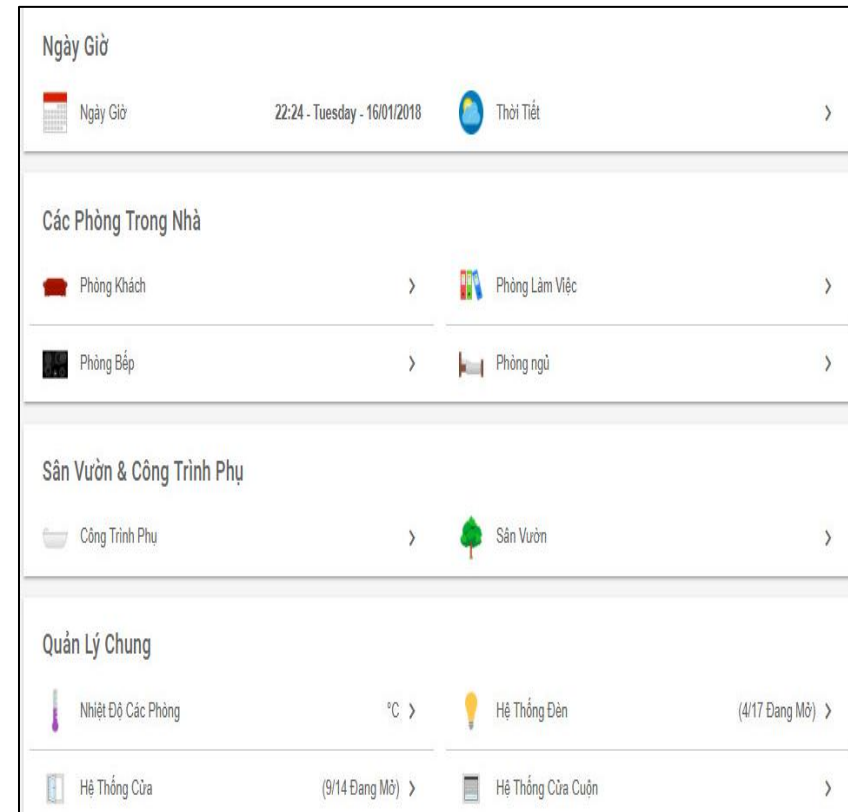
Thiết lập điều khiển trên điện thoại:
Nhập UUID và mật khẩu



Giao diện HABPanel

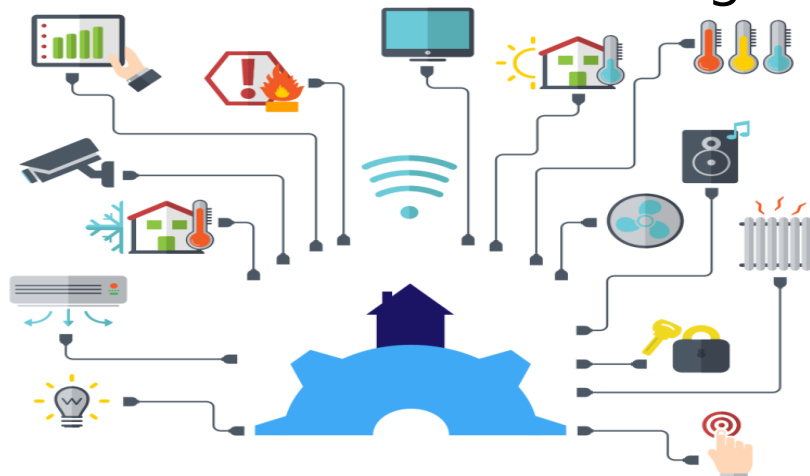


Giao diện Basic UI



Giao diện SmartHome

1. Tổng quan về nhà thông minh
2. Chức năng hiển thị thời gian thực
3. Chức năng điều khiển công tắc điện
4. Chức năng hiển thị thông số từ cảm biến
5. Chức năng phát hiện chuyển động
6. Chức năng gửi thông báo đến điện thoại
7. Chức năng hiển thị Camera IP
8. Chức năng dự báo thời tiết
9. Điều khiển thiết bị qua đám mây
10. Thiết kế giao diện người dùng





Cảm ơn