



Nhóm 16

Hệ Thống Nhà Thông Minh

Sử dụng Raspberry PI 4





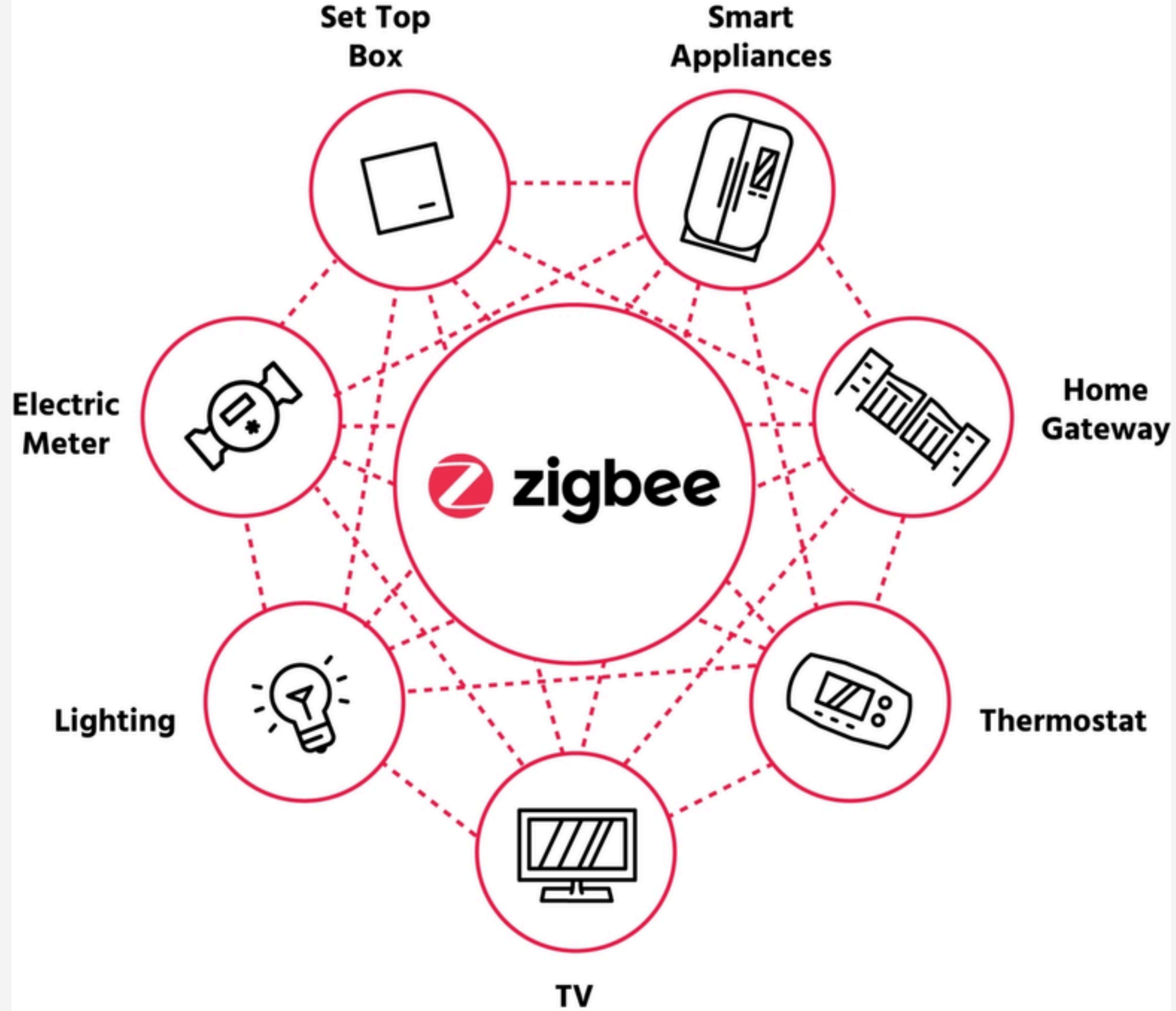
NHÓM 16

Giới thiệu về Dự Án

- Điều khiển thiết bị (cơ, từ xa, tự động, giọng nói ..)
- Thống báo tình huống khẩn cấp



[Quay lại Trang Chương trình](#)



Smart Home

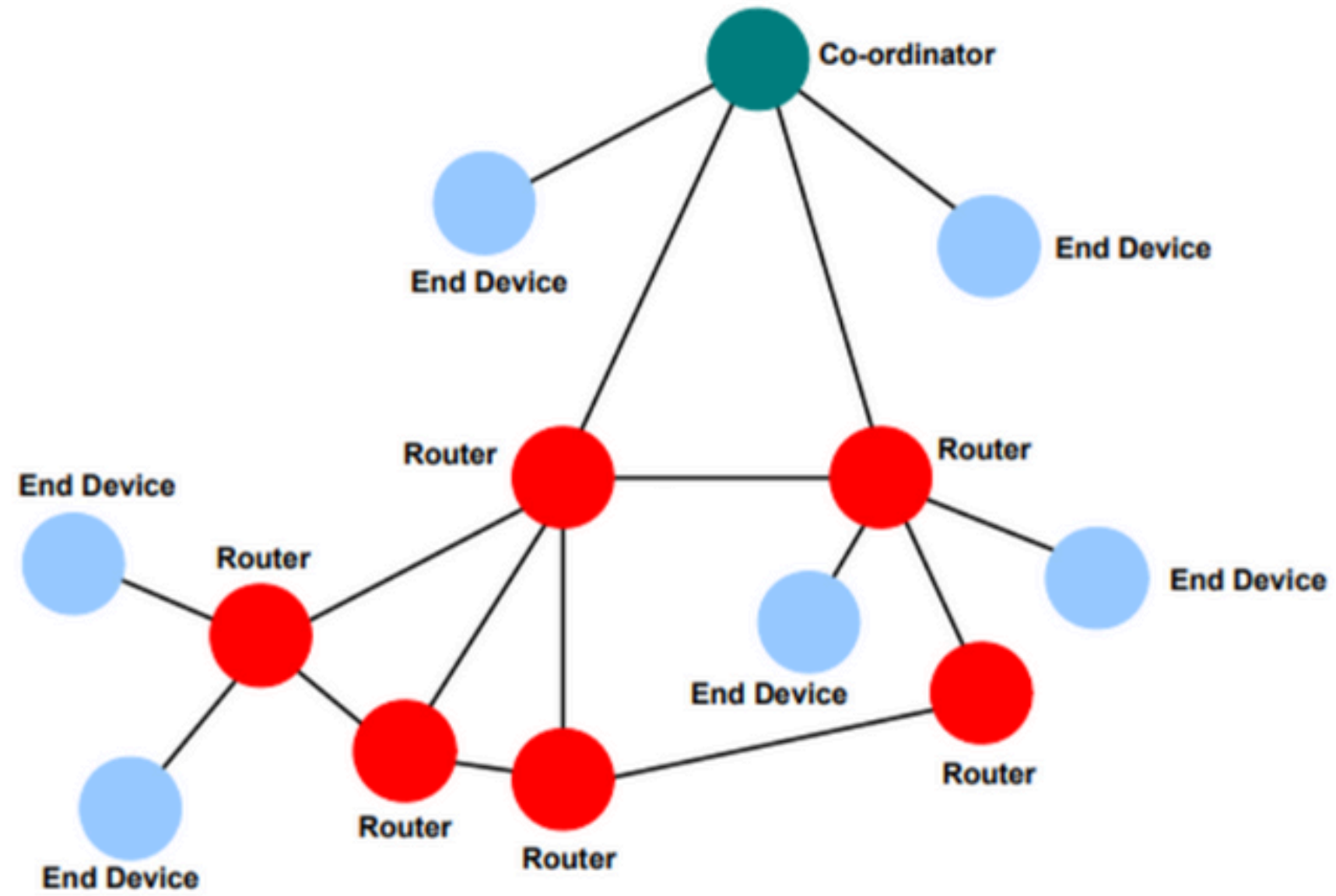


Figure 2: Simple Mesh Network



Công tắc thông minh Tuya Zigbee, hỗ trợ Google và Alexa

ROUTER



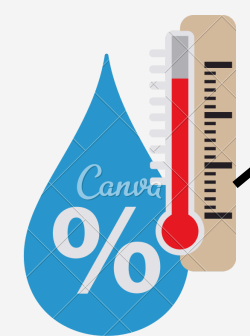
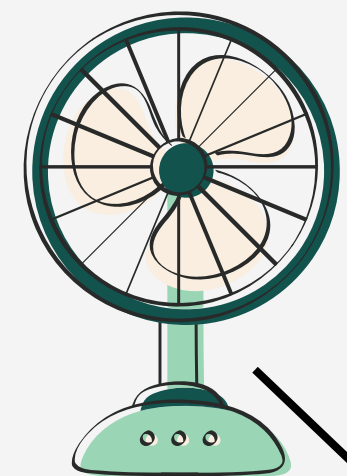
Cảm biến nhiệt độ, độ ẩm Aqara T1

END DEVICE

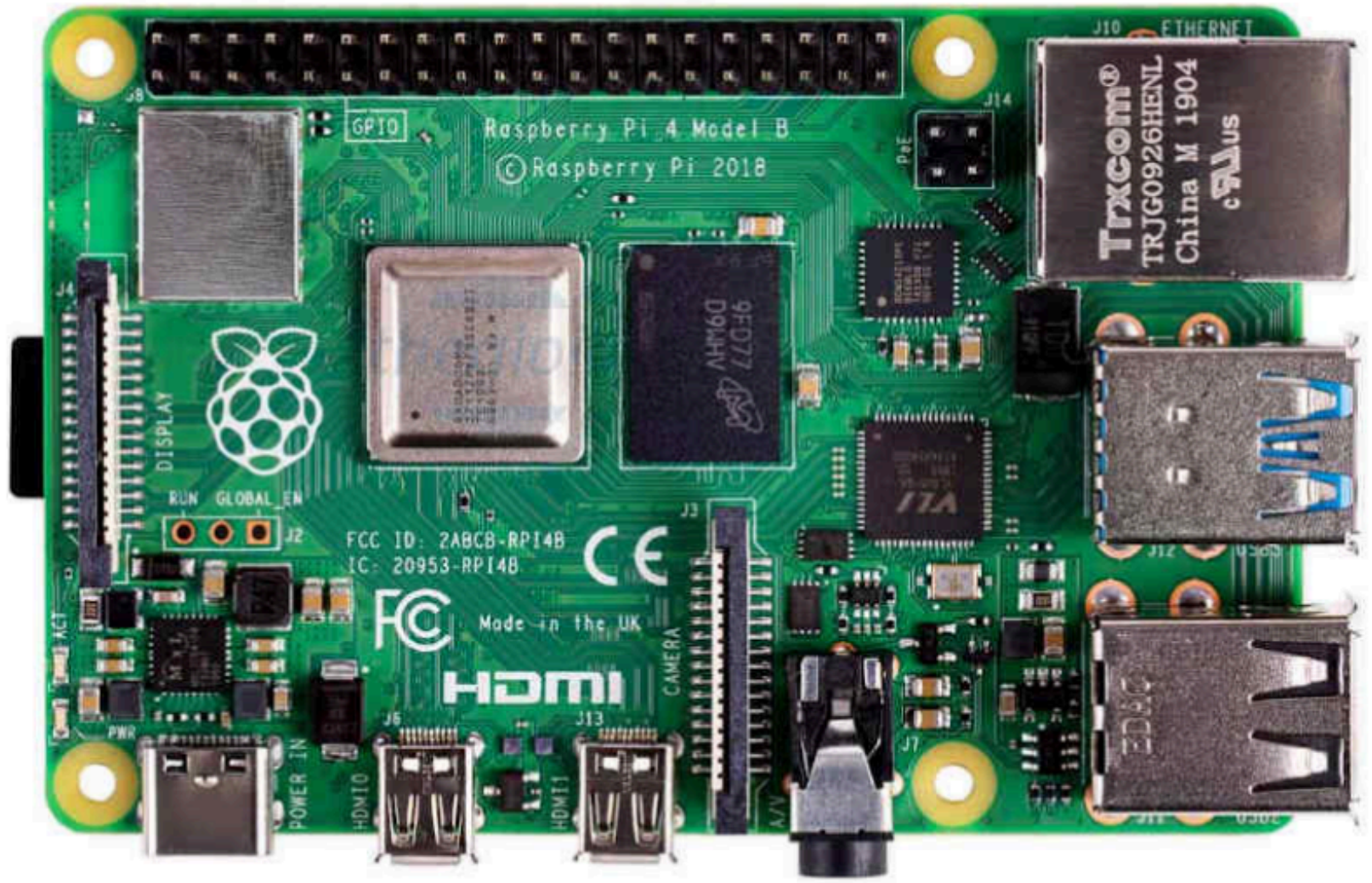


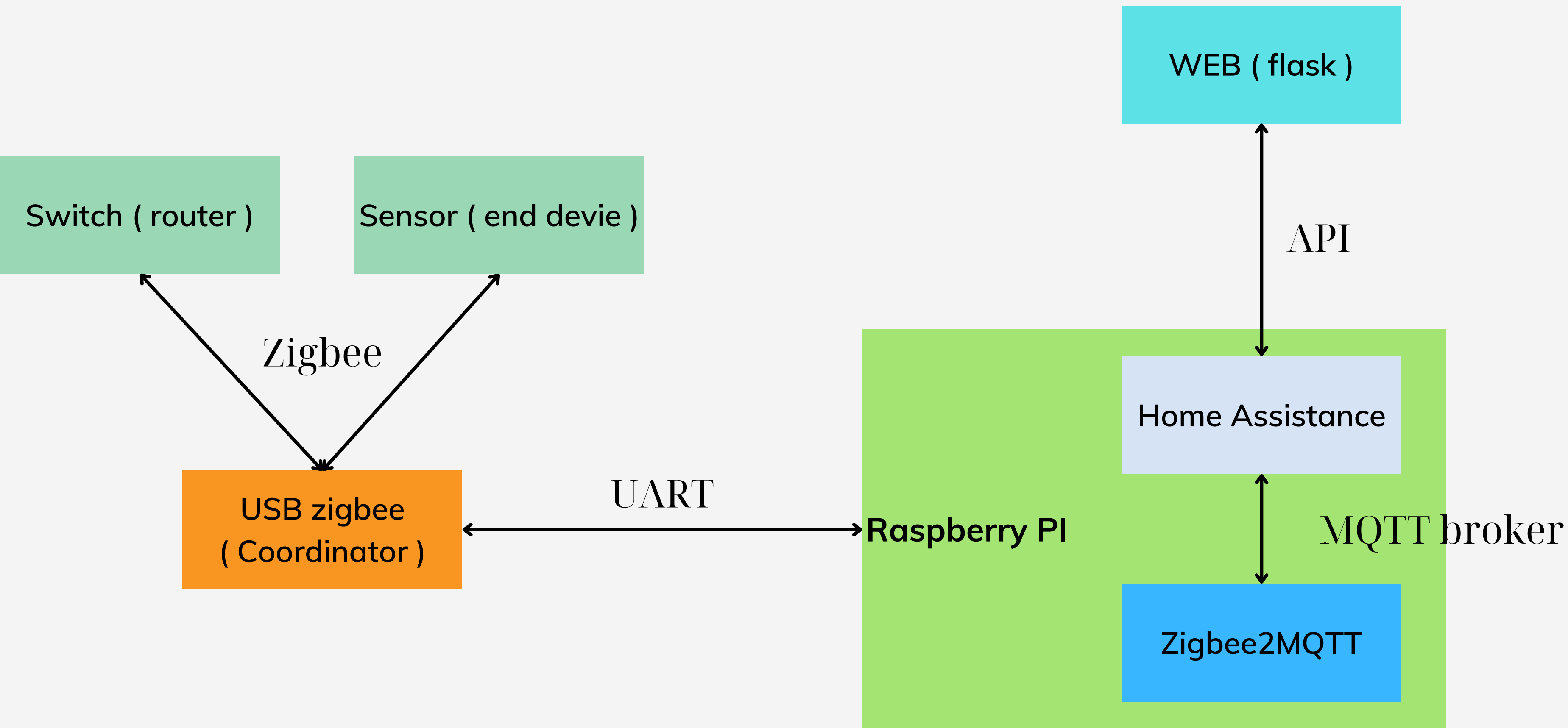
Mạch Thu Phát ZigBee CC2531 USB 2.4Ghz

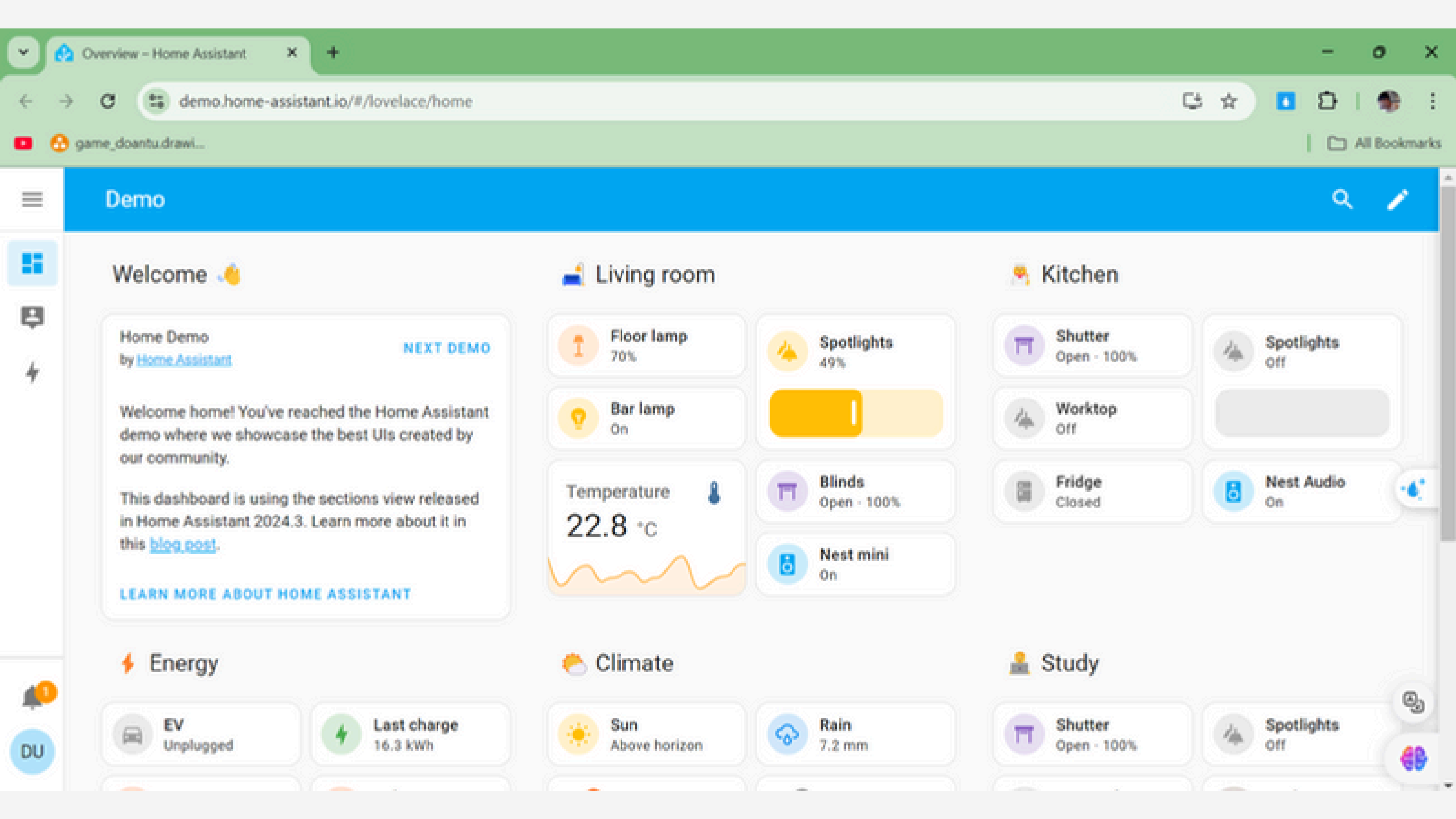
COORDINATOR



Raspberry Pi 4 Model B







THIẾT LẬP PHẦN CỨNG

mô-đun USB Zigbee gắn vào Raspberry Pi., Các thiết bị vật lý khác

LẬP TRÌNH giao tiếp với PI

- Viết mã để giao tiếp với các thiết bị Zigbee bằng Raspberry Pi.
 - Kiểm tra việc giao tiếp giữa Raspberry Pi và các thiết bị Zigbee.
- Xử lý dữ liệu cảm biến:
- Viết mã để đọc dữ liệu nhiệt độ và độ ẩm từ các cảm biến Zigbee
 - Thực hiện logic điều khiển để bật/tắt các thiết bị (đèn, quạt) dựa trên đầu vào của người dùng hoặc các kích hoạt tự động (ví dụ: nhiệt độ).

[Quay lại Trang Chương trình](#)

PHÁT TRIỂN GIAO DIỆN

- Phát triển giao diện người dùng để điều khiển các thiết bị trong nhà thông minh (ví dụ: bật/tắt đèn hoặc quạt).

LẬP TRÌNH giao tiếp với PI

- Viết mã để giao tiếp với các thiết bị Zigbee bằng Raspberry Pi.
- Thực hiện các quy tắc tự động hóa (ví dụ: bật quạt khi nhiệt độ vượt quá ngưỡng nhất định).
- Thực hiện thông báo thời gian thực về nhiệt độ và độ ẩm.

[Quay lại Trang Chương trình](#)

Các bước thực hiện dự án

Phần chính

- Flash ZUSB Zigbee CC2531
- Cài Home Assistant
- Cài Zigbee2mqtt
- Code thiết bị mới (chưa được hỗ trợ).
- Show data lên Home Assistant
- Kết hợp google home mini

Phần làm thêm (cá nhân hóa).

- Lên ý tưởng
- frontend
- backend
- API home assistant
- Ngôn ngữ PYTHON

Bước 1:

Flash USB Zigbee / Chạy zigbee2mqtt

Tải và cài đặt phần mềm HOME ASSISTANT cho PI



[Quay lại chương trình làm việc](#)

Chạy các lệnh trong Pi



LINK github của Nhóm

<https://github.com/teemocoolpro/Hung-s-PROJECT.git>





LINK web tham khảo flash

<https://konnected.vn/diy/diy-flash-zigbee-cc2531-de-them-vao-hass-2020-04-13>

LINK zigbee2mqtt


<https://www.zigbee2mqtt.io/guide/getting-started/#installation>

[Quay lại Trang Chương trình](#)

Enter search criteria							
#	Pic	Friendly name	IEEE Address	Manufacturer	Model	LQI	Power
1		Nhiệt Độ của Hùng	0x54ef441000d1f33a (0x4EF8)	Aqara	WSDCGQ12LM	23	
2		Công tắc của HÙNG	0xa4c138d7168fb636 (0x75B1)	Unsupported	TS0601	21	

Thiết bị mới

Aqara WSDCGQ12LM

Model	WSDCGQ12LM
Vendor	Aqara
Description	Temperature and humidity sensor T1
Exposes	temperature, humidity, pressure, device_temperature, battery, voltage, power_outage_count, linkquality
Picture	
White-label	Aqara TH-S002D, Yandex YNDX-00523




```
1 const fz = require('zigbee-herdsman-converters/converters/fromZigbee');
2 const tz = require('zigbee-herdsman-converters/converters/toZigbee');
3 const exposes = require('zigbee-herdsman-converters/lib/exposes');
4 const tuya = require('zigbee-herdsman-converters/lib/tuya');
5 const e = exposes.presets;
6
7 const definition = {
8   fingerprint: [{modelID: 'TS0601', manufacturerName: '_TZE204_wzoxrizd'}],
9   model: 'TS0601_switch_4_gang',
10  vendor: 'Tuya',
11  description: '4 gang switch',
12  fromZigbee: [tuya.fz.datapoints],
13  toZigbee: [tuya.tz.datapoints],
14  configure: tuya.configureMagicPacket,
15  exposes: [
16    e.switch().withEndpoint('l1'),
17    e.switch().withEndpoint('l2'),
18    e.switch().withEndpoint('l3'),
19    e.switch().withEndpoint('l4'),
20  ],
21  endpoint: (device) => {
22    return {l1: 1, l2: 1, l3: 1, l4: 1};
23  },
24  meta: {
25    multiEndpoint: true,
26    tuyaDatapoints: [
27      [1, 'state_l1', tuya.valueConverter.onOff],
28      [2, 'state_l2', tuya.valueConverter.onOff],
29      [3, 'state_l3', tuya.valueConverter.onOff],
30      [4, 'state_l4', tuya.valueConverter.onOff],
31    ],
32  },
33 };
34
35 module.exports = definition;
```

Thông tin thiết bị

```
fingerprint:[{modelID:'TS0601',manufacturerName: '_TZE204_wzoxrizd'}],  
model: 'TS0601_switch_4_gang',  
vendor: 'Tuya',  
description: '4 gang switch',
```

ID : tên của 1 nhóm thiết bị cùng or gần giống chức năng với nhau : TS0601 là các loại công tắc, rèm cửa , ổ khóa
manufacturerName là tên của thiết bị chính nói ra là biết nó là thiết bị nào EX: ip16

Khởi báo thư viện

```
// Import module `fromZigbee` dùng để xử lý dữ liệu từ thiết bị Zigbee gửi đến  
const fz = require('zigbee-herdsman-converters/converters/fromZigbee');
```

```
// Import module `toZigbee` dùng để gửi lệnh từ hệ thống đến thiết bị Zigbee  
const tz = require('zigbee-herdsman-converters/converters/toZigbee');
```

```
// Import module `exposes` dùng để xác định các tính năng của thiết bị  
const exposes = require('zigbee-herdsman-converters/lib/exposes');
```

```
// Import module `tuya`, chứa các công cụ hỗ trợ cho thiết bị Tuya Zigbee  
const tuya = require('zigbee-herdsman-converters/lib/tuya');
```

Xác định tên tính năng

```
exposes: [  
  e.switch().withEndpoint('l1'),  
  e.switch().withEndpoint('l2'),  
  e.switch().withEndpoint('l3'),  
  e.switch().withEndpoint('l4'),  
],  
meta: { multiEndpoint: true,  
endpointtuyaDatapoints: [  
  [1, 'state_l1', tuya.valueConverter.onOff],  
  [2, 'state_l2', tuya.valueConverter.onOff], )  
  [3, 'state_l3', tuya.valueConverter.onOff],  
  [4, 'state_l4', tuya.valueConverter.onOff],  
],  
},
```


/opt/zigbee2mqtt/data/external_converters/

**/opt/zigbee2mqtt/data/
configuration.yaml**

external_converters:

- /opt/zigbee2mqtt/data/external_converters/tuya.js**

Enter search criteria

X

#

Pic

Friendly name

IEEE Address

Manufacturer

Model

LQI

Power

1



Nhiệt độ của Hùng

0x54ef441000a1f33a
(0x1900)

Aqara

WSDCCQ12UM

84



2



Công tắc của Hùng

0xa4c138a7168fb636
(0x6100)

Tuya

TS0601_switch_4_gang

84



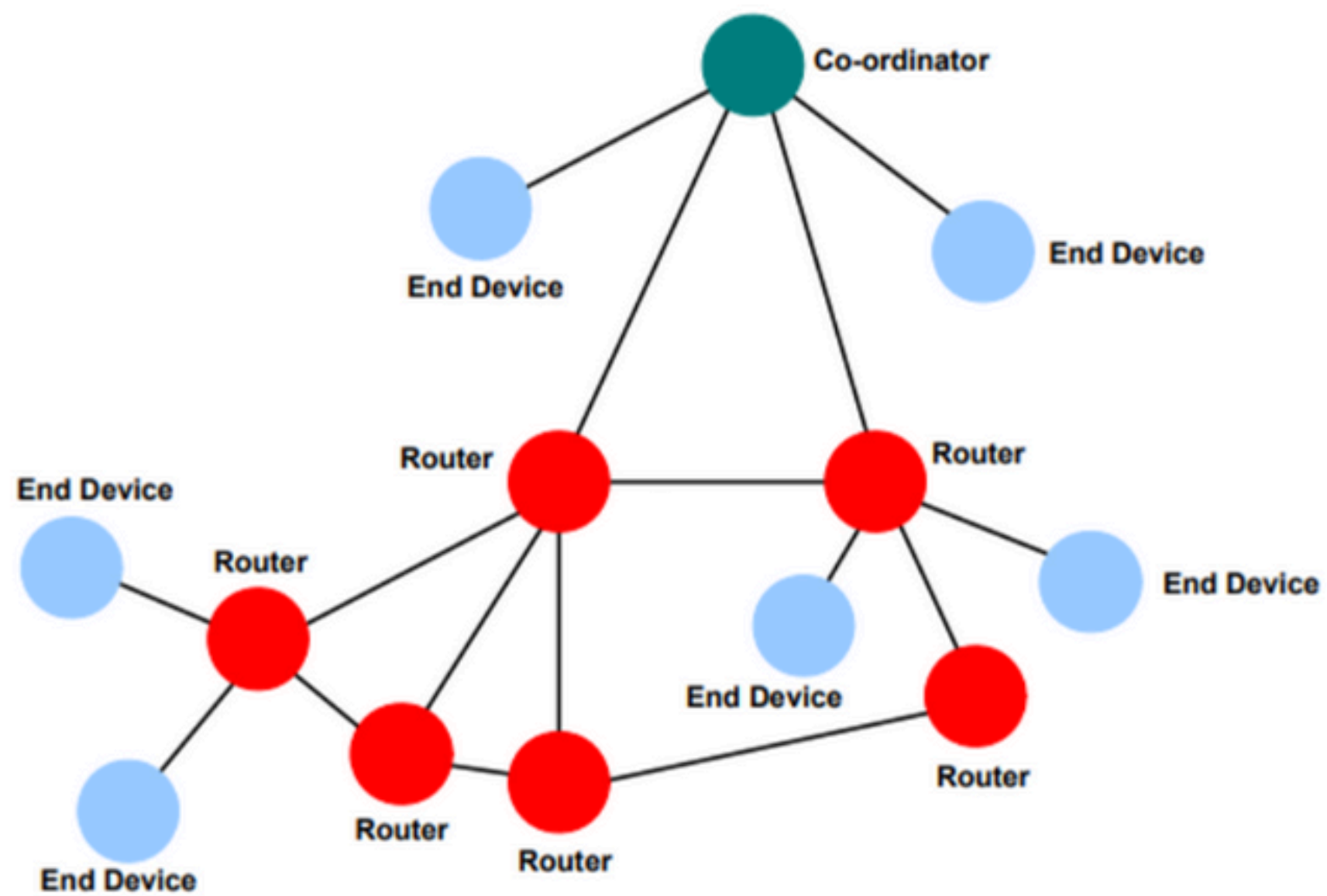
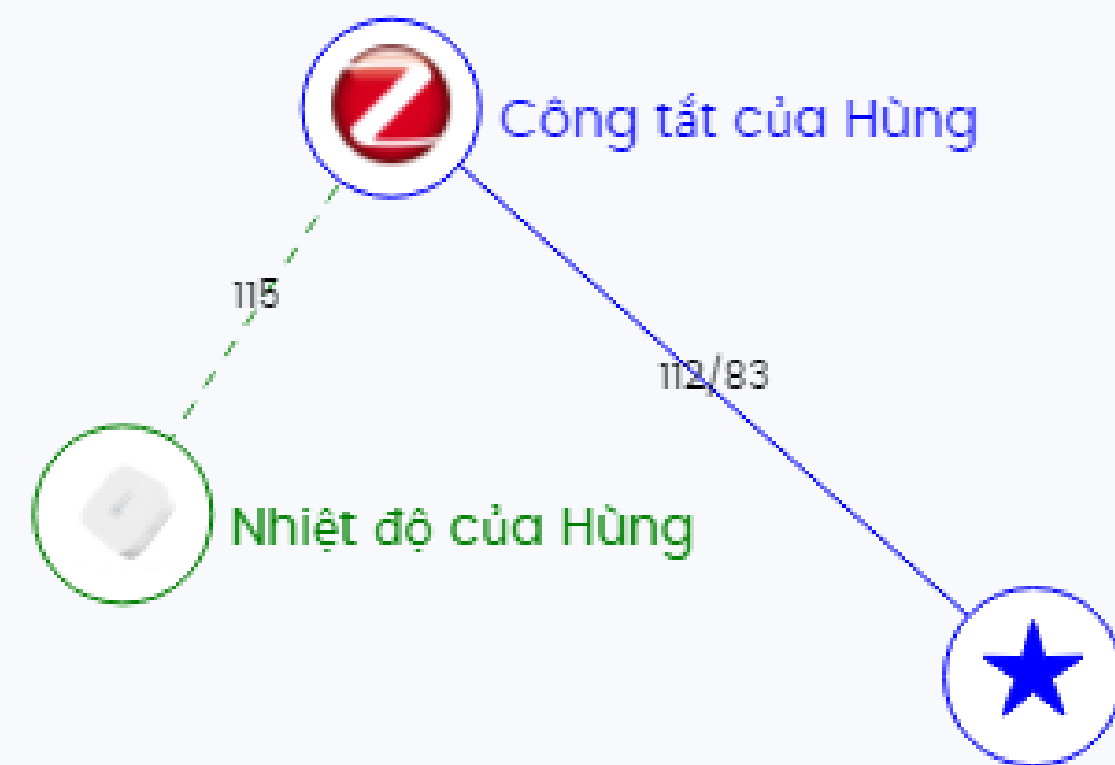
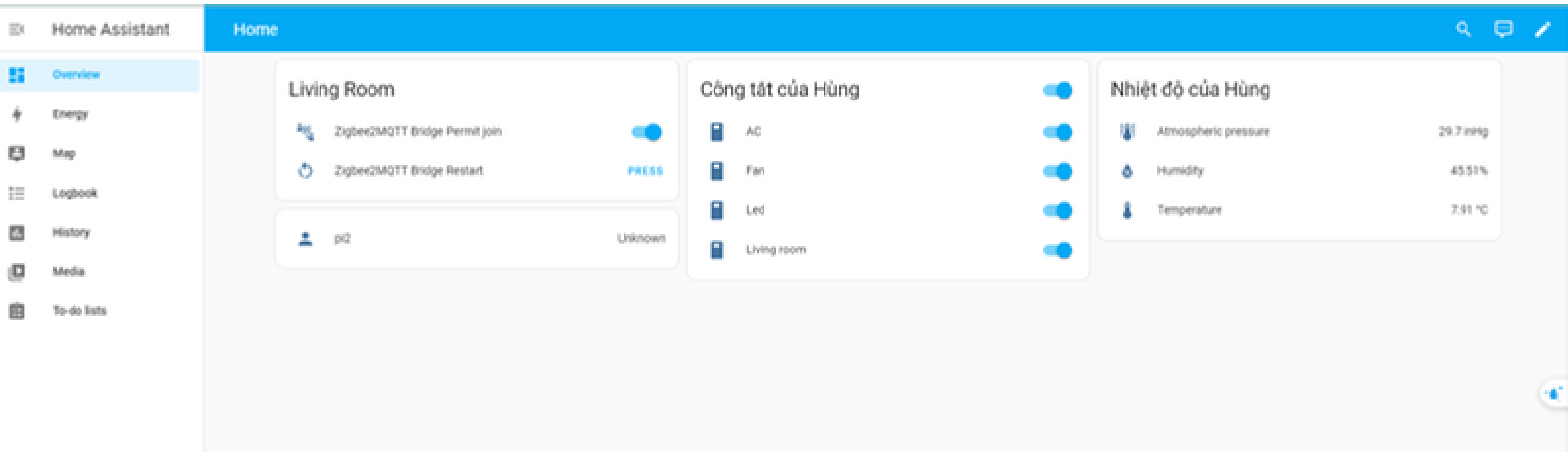


Figure 2: Simple Mesh Network



http://192.168.2.138:8123





Overview



laptop



led



light



tempurate



Áp suất khí quyển

Unknown



Độ ẩm

67.05%



Nhiệt độ

29.64 °C

Zigbee2MQTT Bridge



Permit join



Restart

PRESS



Overview



pi

Away



Cloudy

Forecast Home

25.6 °C

95%

switch room



fan



laptop



led



light



tempurate

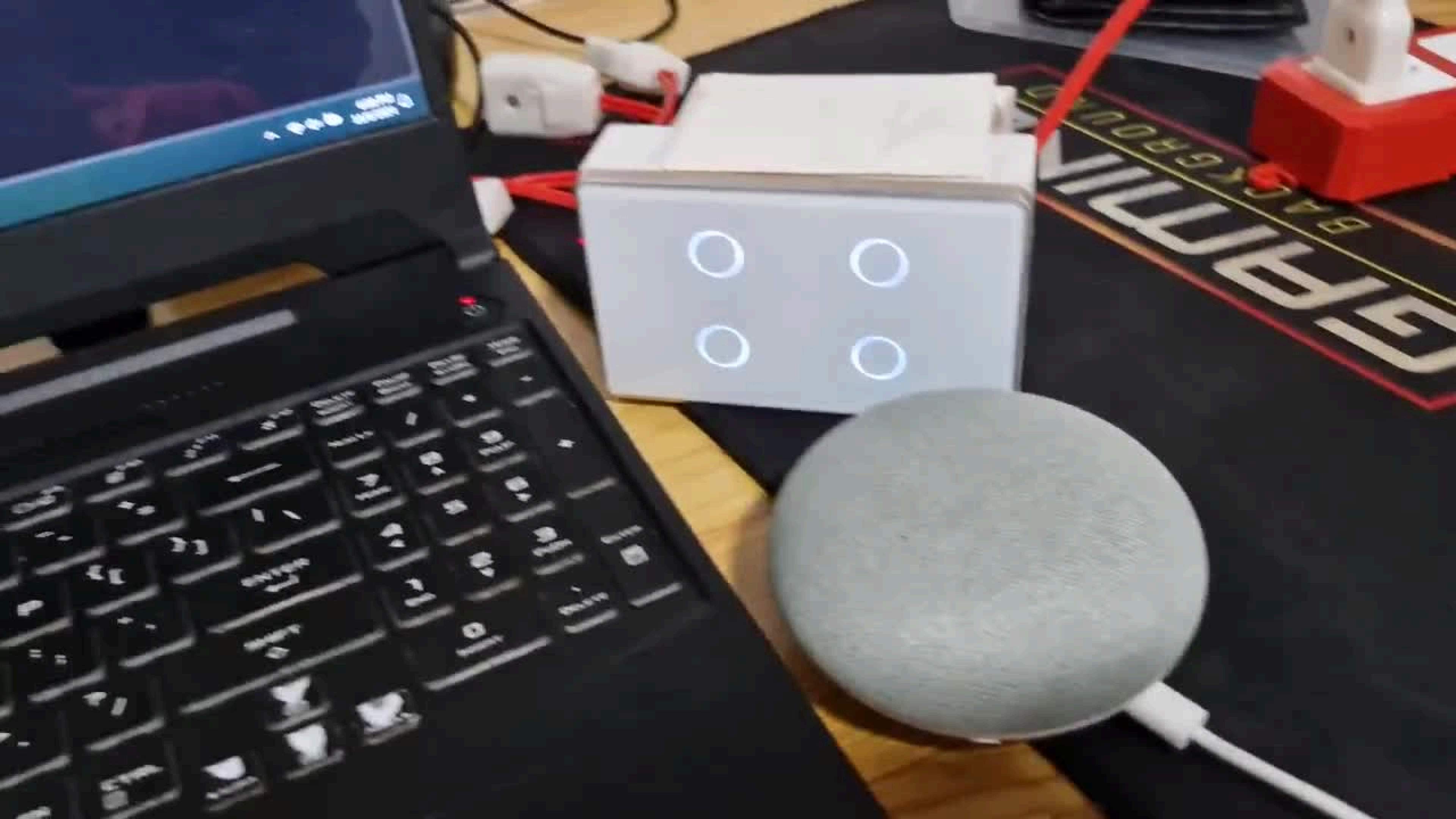


Áp suất khí quyển

Unknown



Tích hợp thêm google home mini
để điều khiển bằng giọng nói



Các bước thực hiện dự án

Phần chính

- Flash ZUSB Zigbee CC2531
- Cài Home Assistant
- Cài Zigbee2mqtt
- Code thiết bị mới (chưa được hỗ trợ).
- Show data lên Home Assistant
- Kết hợp google home mini

Phần làm thêm (cá nhân hóa).

- Lên ý tưởng
- frontend
- backend
- API home assistant
- Ngôn ngữ PYTHON

Các bước thực hiện

tạo 1 thư mục : `/home/pi/frontend/`

Cài flask : internet, giáo trình mạng cảm biến

Flask là một **framework nhẹ, dễ sử dụng, và rất phổ biến để xây dựng các ứng dụng web**. Nó thuộc loại micro-framework, nghĩa là Flask cung cấp các tính năng cốt lõi cần thiết để phát triển một ứng dụng web, nhưng không bắt buộc bạn phải sử dụng các thành phần bổ sung hoặc cấu hình phức tạp.

BACK-END

tạo 1 file .py : /home/pi/frontend/app.py

app.py : là nơi mà các bạn sẽ viết bằng nn PYTHON

GITHUB : <https://github.com/teemocoolpro/Hung-s-PROJECT.git>

KHAI BÁO

THÔNG TIN NƠI MÀ MÌNH LẤY DATA

HA_URL1 = "http://192.168.2.138:8123/api/states/switch.0xa4c138d7168fb636_l1"

HA_URL2 = "http://192.168.2.138:8123/api/states/switch.0xa4c138d7168fb636_l2"

HA_URL3 = "http://192.168.2.138:8123/api/states/switch.0xa4c138d7168fb636_l3"

HA_URL4 = "http://192.168.2.138:8123/api/states/switch.0xa4c138d7168fb636_l4"

HA_URL_PRESSURE = "http://192.168.2.138:8123/api/states/sensor.0x54ef441000d1f33a_pressure"

HA_URL_HUMIDITY = "http://192.168.2.138:8123/api/states/sensor.0x54ef441000d1f33a_humidity"

HA_URL_TEMPERATURE = "http://192.168.2.138:8123/api/states/sensor.0x54ef441000d1f33a_temperature"

KEY :

```
HEADERS = {
```

```
    "Authorization": "Bearer
```

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJpc3MiOiIxY2FiZjUxMzk5ZTg0NGZkYmYxMzI5YTc0NzgwMzNhMC
```

```
lsImIhdCI6MTcyOTYwMjgwNiwiZXhwljoyMDQ0OTYyODA2fQ.aS_iRJPbK3tec2iYIgJXf_NxEHyDqZcJeq-
```

```
ea8QQmt0",
```

```
    "Content-Type": "application/json"
```

```
}
```

Các route lấy dữ liệu cảm biến

```
@app.route('/get_temperature')
```

```
def get_temperature():
```

```
    response = requests.get(HA_URL_TEMPERATURE, headers=HEADERS)
```

```
    if response.status_code == 200:
```

```
        return jsonify(response.json())
```

```
    else:
```

```
        return jsonify({"error": "Could not retrieve temperature"}), 500
```

Các route điều khiển công tắc

```
@app.route('/turn_on_switch/<int:switch_number>', methods=['POST'])
```

```
def turn_on_switch(switch_number):
```

```
    url = "http://192.168.2.138:8123/api/services/switch/turn_on"
```

```
    data = {"entity_id": f"switch.0xa4c138d7168fb636_l{switch_number}"}
```

```
    response = requests.post(url, headers=HEADERS, json=data)
```

```
    if response.status_code == 200:
```

```
        return jsonify({"success": True})
```

```
    else:
```

```
        return jsonify({"error": "Could not turn on switch"}), 500
```

```
if __name__ == '__main__':  
    threading.Thread(target=save_data_to_file, daemon=True).start()  
    app.run(host='0.0.0.0', port=5000, debug=True)
```

http://192.168.2.138:5000

Trạng Thái Công Tắc

Công tắc 1  Công tắc 1: on

Công tắc 2  Công tắc 2: on

Công tắc 3  Công tắc 3: on

Công tắc 4  Công tắc 4: on

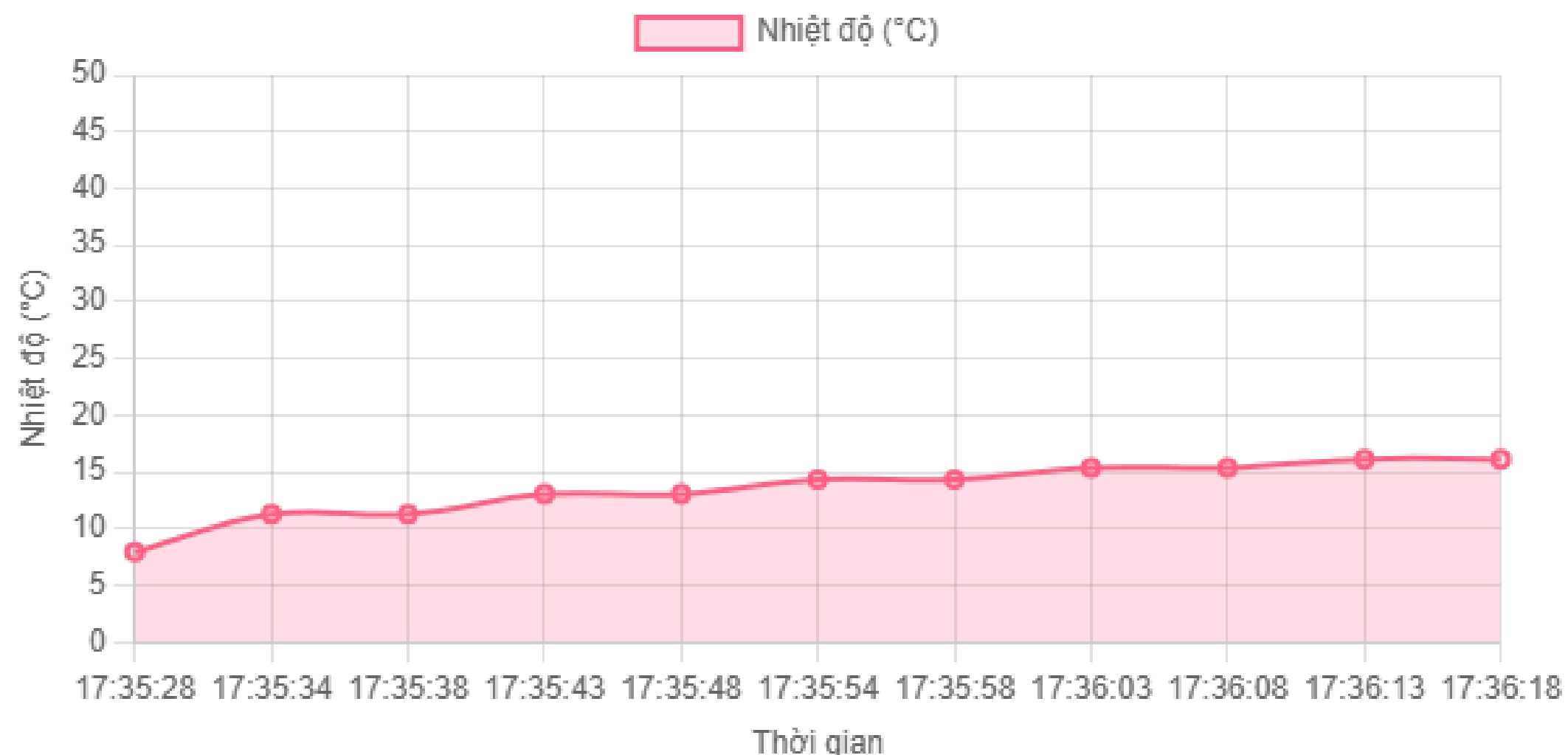
Dữ liệu cảm biến

Áp suất: 29.7

Độ ẩm: 67.08

Nhiệt độ: 31.44

Biểu đồ Nhiệt Độ



Trạng Thái Công Tắc

Công tắc 1  Công tắc 1: off

Công tắc 2  Công tắc 2: off

Công tắc 3  Công tắc 3: off

Công tắc 4  Công tắc 4: off

Dữ liệu cảm biến

Áp suất: 29.7

Độ ẩm: 67.08

Nhiệt độ: 31.44



Search



ENG



12:5
10/18