

Danh sách hình vẽ

| | | |
|----|---|----|
| 1 | Topo của network | 4 |
| 2 | Tạo 1 block Stream trên Adafruit IO | 6 |
| 3 | Dashboard sau khi tạo 1 block mới | 6 |
| 4 | Adafruit IO Key để MQTT Client có thể được xác thực và lấy dữ liệu từ topic . | 7 |
| 5 | Cách điền vào các field của trigger dành cho Google Assistant | 8 |
| 6 | Applet được tạo ra sau khi thực hiện quy trình | 9 |
| 7 | Kết quả sau khi nói "turn on 2" | 11 |
| 8 | Kết quả sau khi nói "turn off 2" | 12 |
| 9 | Kết quả sau khi nói "turn on 3" | 12 |
| 10 | Kết quả sau khi nói "turn off 3" | 13 |

1 GIỚI THIỆU VÀ MỤC TIÊU ĐỀ TÀI

1.1 Giới thiệu đề tài

Trong bối cảnh Internet vạn vật (IoT) ngày càng phát triển, người dùng càng mong muốn họ không phải tới gần thiết bị để điều khiển mà họ có thể điều khiển từ xa thông qua smartphone của mình. Trong đề tài này nhóm xin trình bày cách thiết kế một mạng cảm biến không dây được điều khiển thông qua ứng dụng Google Assistant của Google.

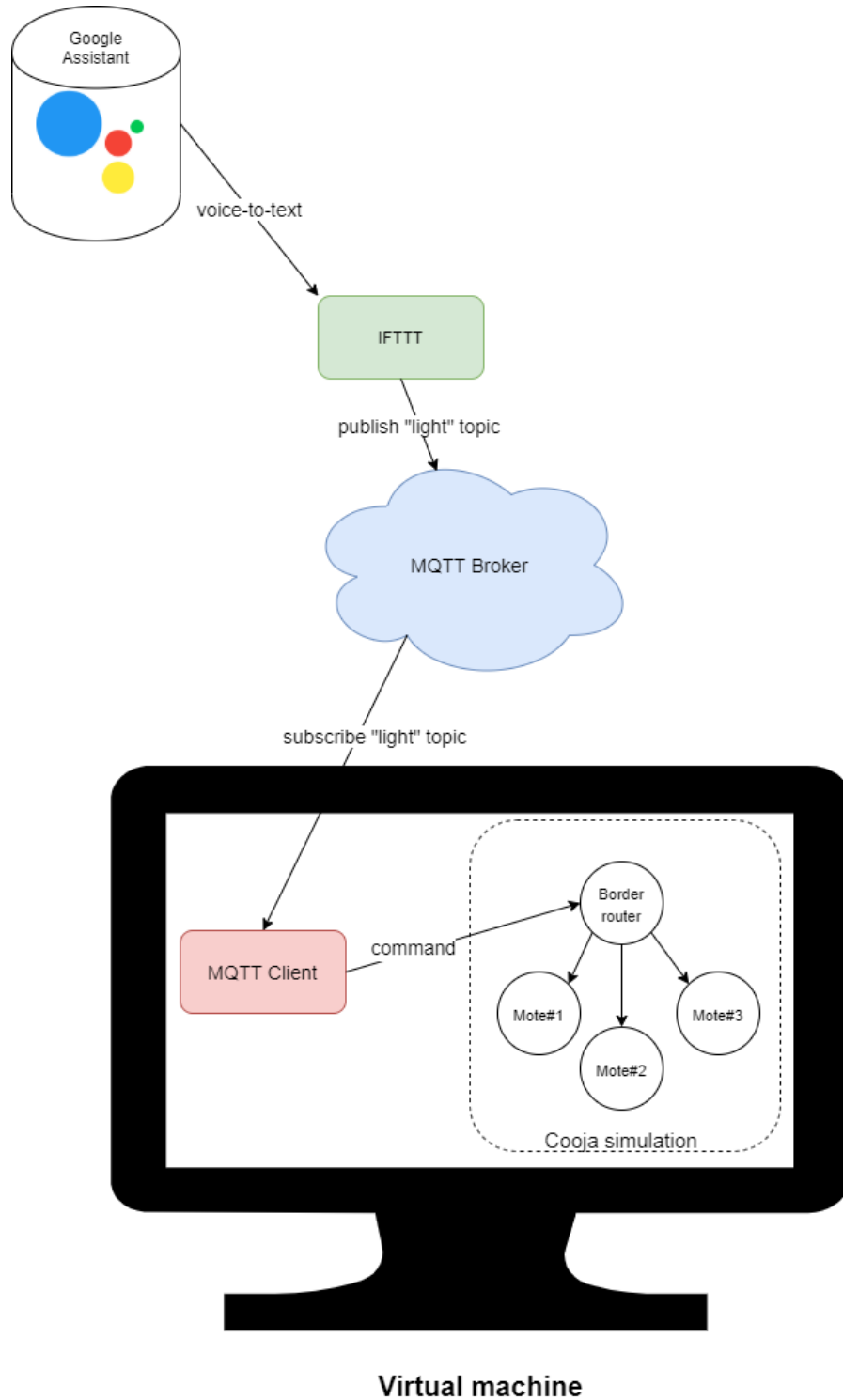
1.2 Mục tiêu đề tài

Mục tiêu đề tài này sẽ tập trung vào thiết kế và mô phỏng bằng Cooja một hệ thống gồm mạng cảm biến không dây và các ứng dụng khác cần thiết trong một máy ảo và hệ thống trên có thể được điều khiển các đèn LED màu xanh bởi Google Assistant.

2 THIẾT KẾ VÀ THỰC HIỆN

2.1 Topo của network

Topo của network, bao gồm mạng cảm biến và các ứng dụng cần thiết được trình bày qua sơ đồ sau



Hình 1: Topo của network

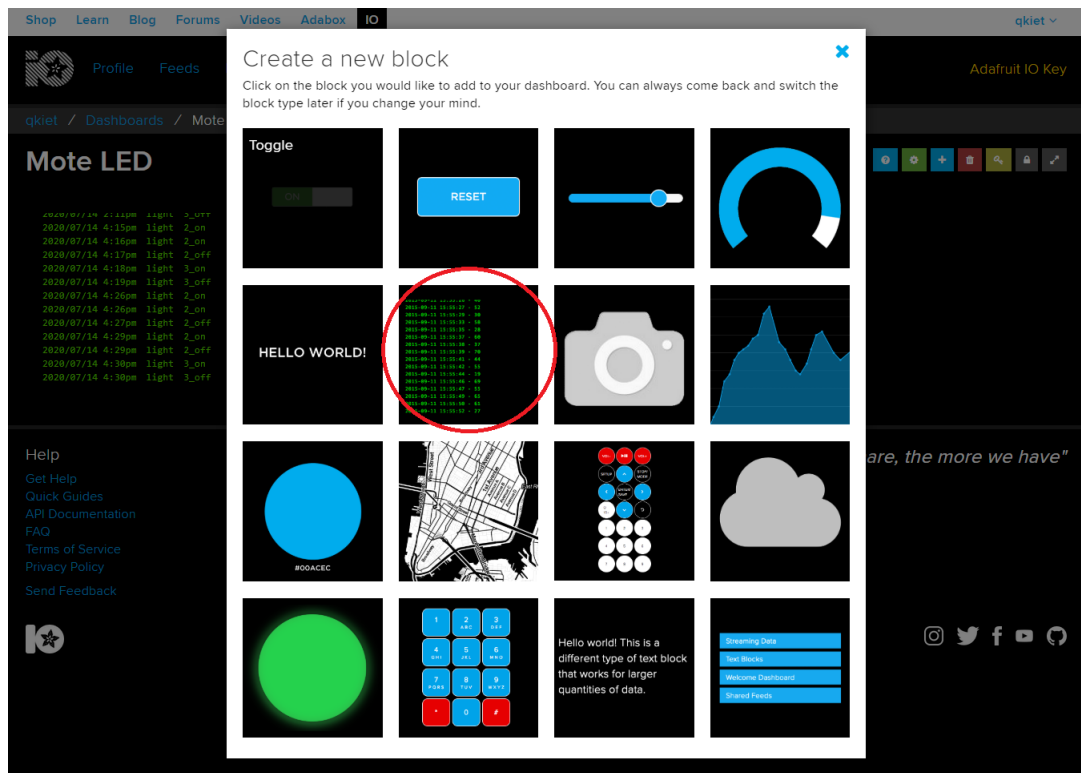
Theo sơ đồ trên thì ta có các phần tử quan trọng trong network như sau

- **Google Assistant:** đây chính là thiết bị để chuyển giọng nói của người thành chữ. Google Assistant còn đảm nhiệm chức năng gửi text chuyển được lên platform IFTTT để xử lý lệnh từ giọng nói
- **IFTTT:** là 1 platform dùng để kết nối thiết bị và ứng dụng với nhau. Cụ thể ở đây IFTTT sẽ kết nối Google Assistant với MQTT Broker. Như vậy, bất cứ khi nào Google Assistant gửi text tới IFTTT thì IFTTT đảm nhiệm vai trò publisher cho MQTT network và publish cho MQTT Broker
- **MQTT Broker** sẽ là nơi chuyển tiếp message được publish cho các client subscribe vào topic tương ứng. Trong topo này, MQTT Broker đóng vai trò trung gian để IFTTT giao tiếp với MQTT Client ở máy ảo
- **MQTT Client:** là 1 ứng dụng được chạy trong máy ảo để chuyển text chuyển từ Google Assistant trở thành command điều khiển các led trong mạng cảm biến được mô phỏng bằng Cooja. Lưu ý rằng để nhận được text từ Google Assistant thì MQTT Client lần IFTTT phải cùng sử dụng 1 MQTT Broker và 1 topic (trong topo thì topic tên là “light”)
- **Mạng cảm biến:** chính là end user cuối cùng trong topo mạng này. Mạng cảm biến bao gồm 1 border router đóng vai trò liên kết mạng cảm biến với network bên ngoài, như vậy command từ MQTT Client mới có thể truyền tới các mote trong mạng cảm biến thông bằng UDP và Ipv6

2.2 Quy trình thực hiện

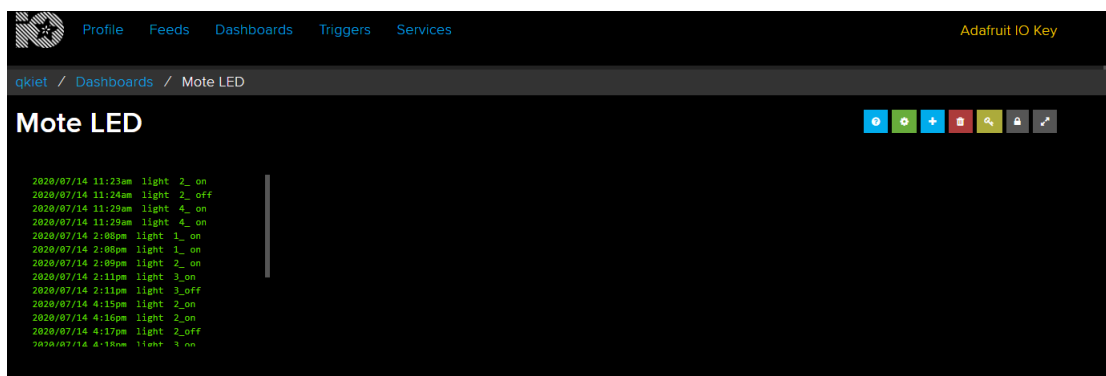
Adafruit IO

Trước hết phải tạo một MQTT Broker để text từ Google Assistant có thể truyền tới MQTT Client và Client sử dụng nó để điều khiển LED. Ta sẽ vào <https://io.adafruit.com> và đăng kí tài khoản. Sau đó tạo topic "light" để IFTTT có thể gửi cho MQTT Client, nhưng topic không tạo trực tiếp trên Adafruit IO mà phải thông qua tạo 1 block trong Dashboard. Block là 1 đơn vị hiển thị thông tin hoặc tương tác với các feed và việc tạo Block đồng thời tạo một feed mới, nghĩa là tạo 1 topic mới. Có rất nhiều loại block khác nhau như button, slider... và nhóm chọn block Stream để có thể theo dõi message nào được gửi lên



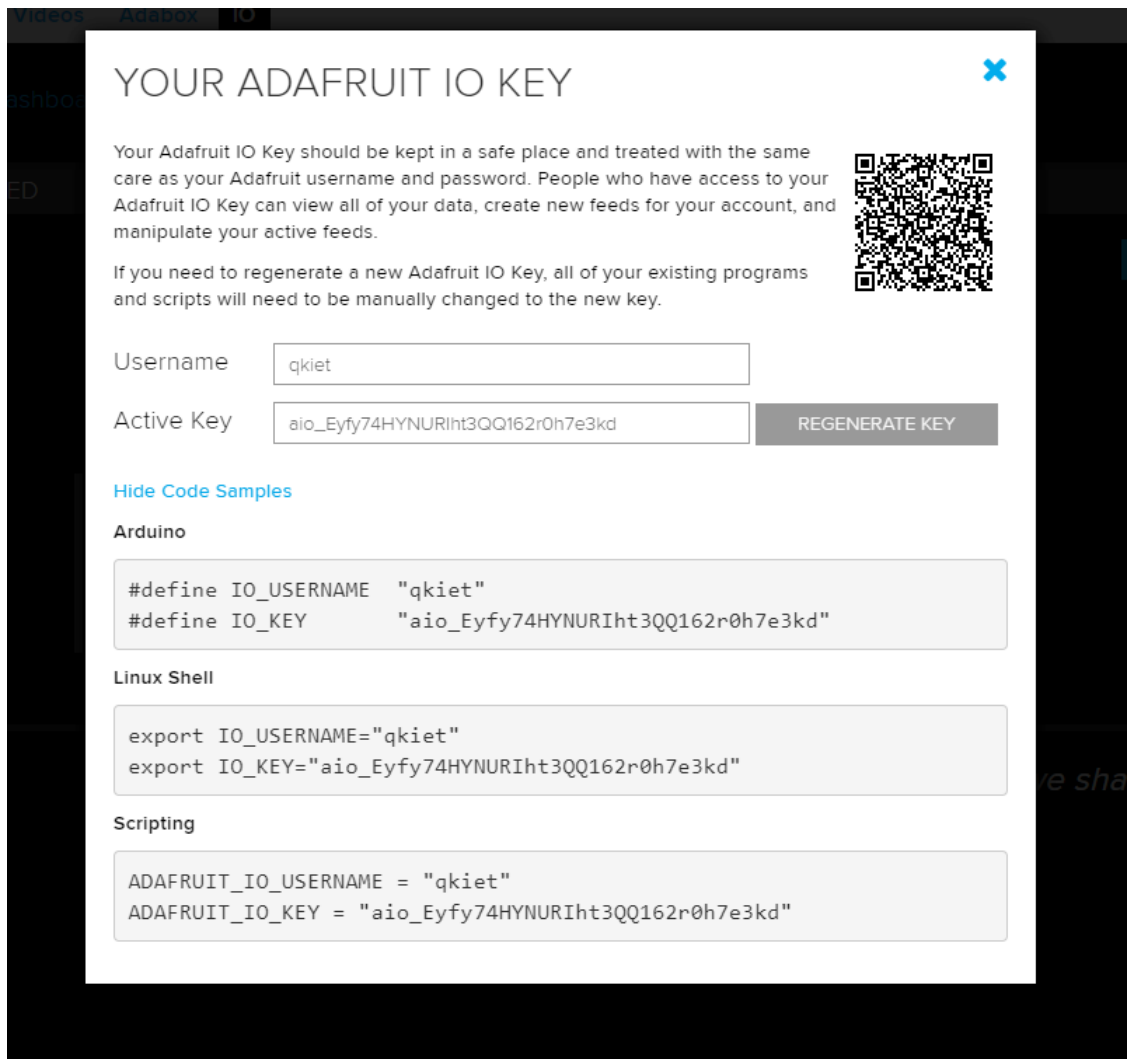
Hình 2: Tạo 1 block Stream trên Adafruit IO

Như vậy ta đã tạo 1 topic mới, và dashboard hiển thị 1 block mới như sau:



Hình 3: Dashboard sau khi tạo 1 block mới

Tuy nhiên ngoài việc tạo topic mới ta cũng cần lưu lại các thông tin quan trọng để MQTT Client có thể connect vào Adafruit IO vì Aafruit IO không phải là 1 public Broker nên nó sẽ có cơ chế bảo mật riêng. Vào Adafruit IO Key ở phía trên bên phải và lưu lại username và active key



Hình 4: Adafruit IO Key để MQTT Client có thể được xác thực và lấy dữ liệu từ topic

IFTTT

IFTTT hoạt động thông qua việc người dùng tạo ra các applet để kết nối một ứng dụng với một service. Applet gồm một trigger là điều kiện kích hoạt applet này và service là nơi applet sẽ thực hiện. Như vậy đối với đề tài thì trigger sẽ là Google Assistant và service chính là 1 MQTT Broker.

Cách làm applet này như sau:

- Trước hết phải đăng ký một tài khoản với IFTTT để có thể tạo ra applet
- Đăng nhập tài khoản IFTTT, sau đó bắt đầu tạo ra một applet mới bằng cách bấm vào nút "Create" góc trên bên phải
- Khi yêu cầu chọn service cho trigger của applet thì ta chọn "Google Assistant" với kiểu kích hoạt "Say a phrase with both a number and a text ingredient" vì khi điều khiển bằng Google Assistant ta mong muốn có thể nói bật tắt và mở ID
- Điền vào trigger như hình dưới đây:

What do you want to say?

turn \$ #

Enter a \$ where you want to say the text ingredient and a # where you want to say the number

What's another way to say it? (optional)

Enter a \$ where you want to say the text ingredient and a # where you want to say the number

And another way? (optional)

Enter a \$ where you want to say the text ingredient and a # where you want to say the number

What do you want the Assistant to say in response?

I turn \$ node #

You can enter a \$ or a # where you want to hear the text or number in the response

Language

English

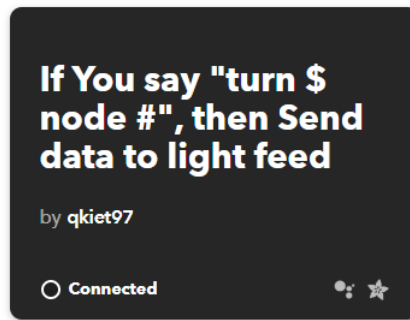
Hình 5: Cách điền vào các field của trigger dành cho Google Assistant

Như vậy cú pháp để tắt mở các đèn led là "turn + (on/off) + mote ID". Ví dụ ta muốn mở đèn LED mote số 2 thì chỉ cần nói "turn on 2"

- Trong phần action service ta sẽ chọn Adafruit và chọn Send data to Adafruit IO. Adafruit IO chính là MQTT Broker mà IFTTT sẽ gửi data khi nhận trigger từ Google Assistant
- Sau đó chọn **Feed name** là "light", feed mà nhóm đã tạo trước 1 feed trên Adafruit và đây cũng là feed mà MQTT client sẽ subscribe. **Data to save** sẽ là `{{NumberField}}_{{TextField}}`, tương ứng là (mote ID)_(on/off). Ví dụ khi tắt đèn led của mode ID 3 thì data gửi lên Adafruit IO sẽ là 3_off

Kết quả ta có 1 applet như sau:

My Applets



Hình 6: Applet được tạo ra sau khi thực hình quy trình

MQTT Client

MQTT Client nhận data subscribe tại feed "light" và được chạy tại máy ảo. MQTT Client được viết bằng python và sử dụng thư viện Adafruit_io để ứng dụng có thể tạo kết nối với MQTT Broker Adafruit IO và lấy data về. Ngoài ra MQTT client còn xử lý data lấy từ topic "light" và chuyển thành command và địa chỉ IPv6 cần gửi.

Code của MQTT Client được trình bày như sau:

```
import logging
import paho.mqtt.client as mqtt
import time
import sys

import socket

HOST = [ 'aaaa::212:7402:2:202', 'aaaa::212:7403:3:303', 'aaaa::212:7404:4:404' ]
# Standard loopback interface address (localhost)

PORT = 3000          # Port to listen on (non-privileged ports are > 1023)

USERNAME = 'tuannv'
KEY       = 'aio_vCRp50VNZfCzjGLQou8mf9mjBlTj'

ADAFRUIT_SERVER = 'io.adafruit.com'
ADAFRUIT_PORT   = 1883

PATH          = USERNAME + '/feeds/Light'

def on_connect(client, userdata, flags, rc):
    print('Connected!')
    client.subscribe(PATH)
    print('Subscribed to path {0}'.format(PATH))

def on_disconnect(client, userdata, rc):
    print('Disconnected!')
```



```

def on_message(client, userdata, msg):
    cmd = msg.payload.decode('utf-8')
    s = socket.socket(socket.AF_INET6, socket.SOCK_DGRAM)
    print('Received on {0}: {1}'.format(msg.topic, cmd))
    if (int(cmd[0]) == 2): #findout which address
        destination_index = 0
    elif (int(cmd[0]) == 3):
        destination_index = 1
    elif (int(cmd[0]) == 4):
        destination_index = 2
    if (cmd[2:] == "ON"):
        send_data = "led_on"
    elif (cmd[2:] == "OFF"):
        send_data = "led_off"

    s.connect((HOST[destination_index], PORT))
    print(destination_index)
    print(send_data)
    s.sendall(send_data.encode())
    sys.stdout.flush()
    s.close()

client = mqtt.Client()
client.username_pw_set(USERNAME, KEY)
client.on_connect = on_connect
client.on_disconnect = on_disconnect
client.on_message = on_message
client.connect(ADAFRUIT_SERVER, port=ADAFRUIT_PORT)
client.loop_forever()

```

Như vậy bản chất MQTT Client hoạt động là 1 MQTT Subscriber. Client sẽ được cho trước topic, username và active key và sau đó thực hiện kết nối lên MQTT Broker. Sau đó nó đợi nhận message mới và tiến hành xử lý message và gửi cho mote tương ứng.

Mạng cảm biến

Mạng cảm biến sẽ bao gồm 1 sky mote được nạp firmware border router và các mote khác sẽ là các mote được Google Assistant điều khiển và được nạp firmware *udp-echo-Broker* trong folder **contiki/example/cc2538dk/demo_cc2538** vì firmware này đã có sẵn các lệnh để điều khiển LED màu xanh. Để đơn giản hóa việc mô phỏng thì mạng cảm biến chỉ dùng 2 mote *udp-echo-Broker*

3 KẾT QUẢ

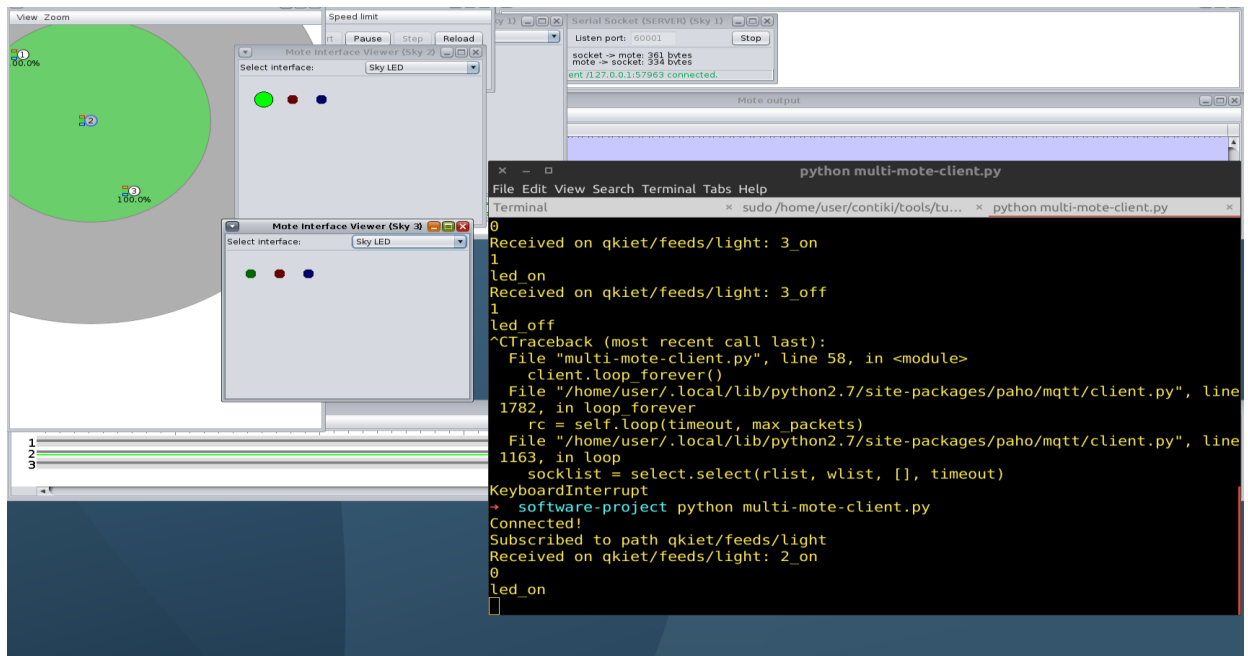
3.1 Kịch bản mô phỏng

Lần lượt nói các cú pháp sau và quan sát kết quả

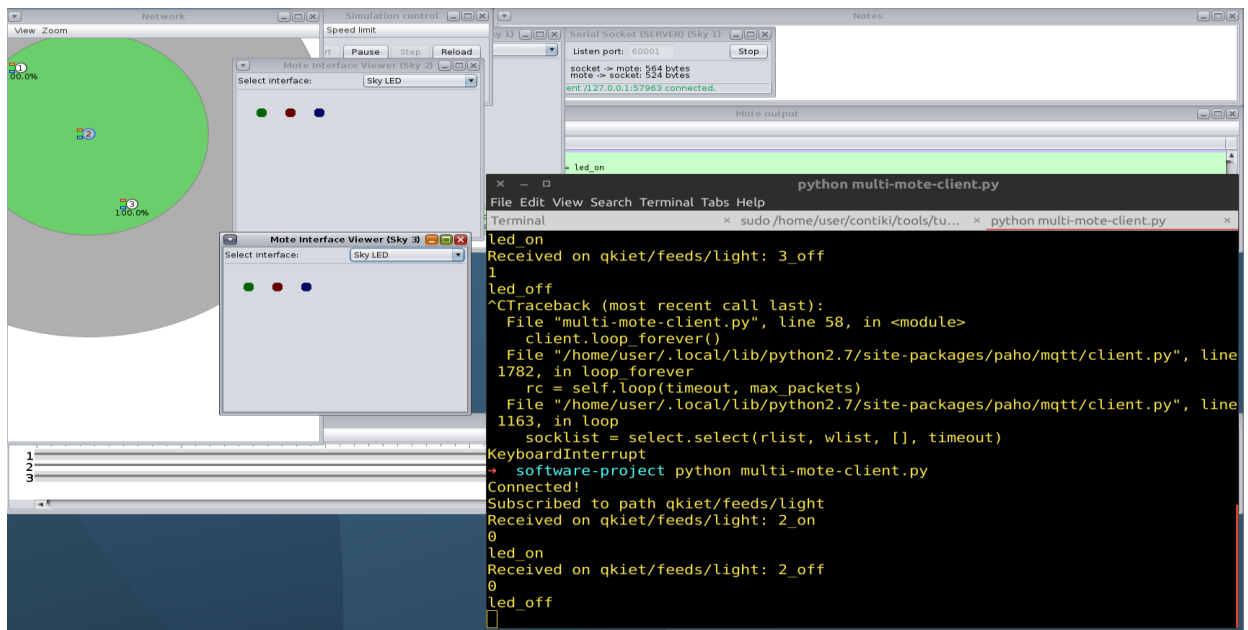
- "turn on 2". Mong đợi kết quả đèn LED xanh lá của mote Id 2 mở
- "turn off 2". Mong đợi kết quả đèn LED xanh lá của mote Id 2 tắt
- "turn on 3". Mong đợi kết quả đèn LED xanh lá của mote Id 3 mở
- "turn off 3". Mong đợi kết quả đèn LED xanh lá của mote Id 3 tắt

3.2 Kết quả mô phỏng

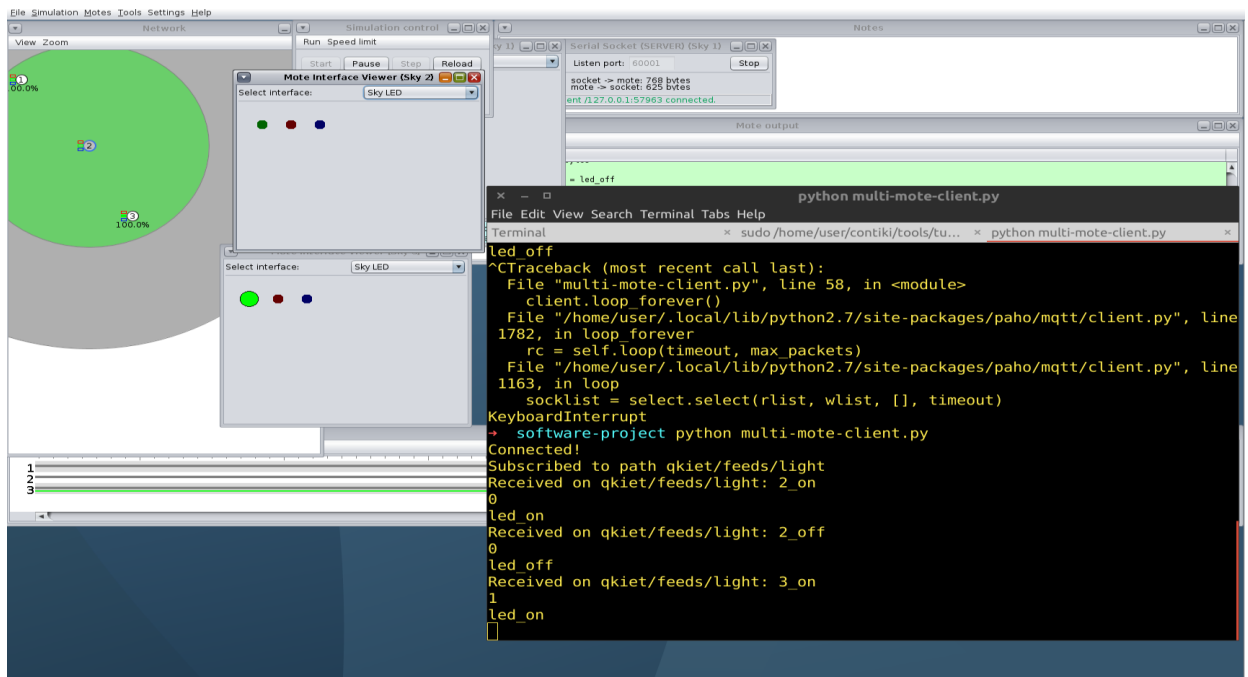
Sau đây là kết quả sau khi nói các cú pháp trong kịch bản. Lưu ý rằng trạng thái các đèn LED ở phía trên của mote 2 và ở phía dưới là của mote 3.



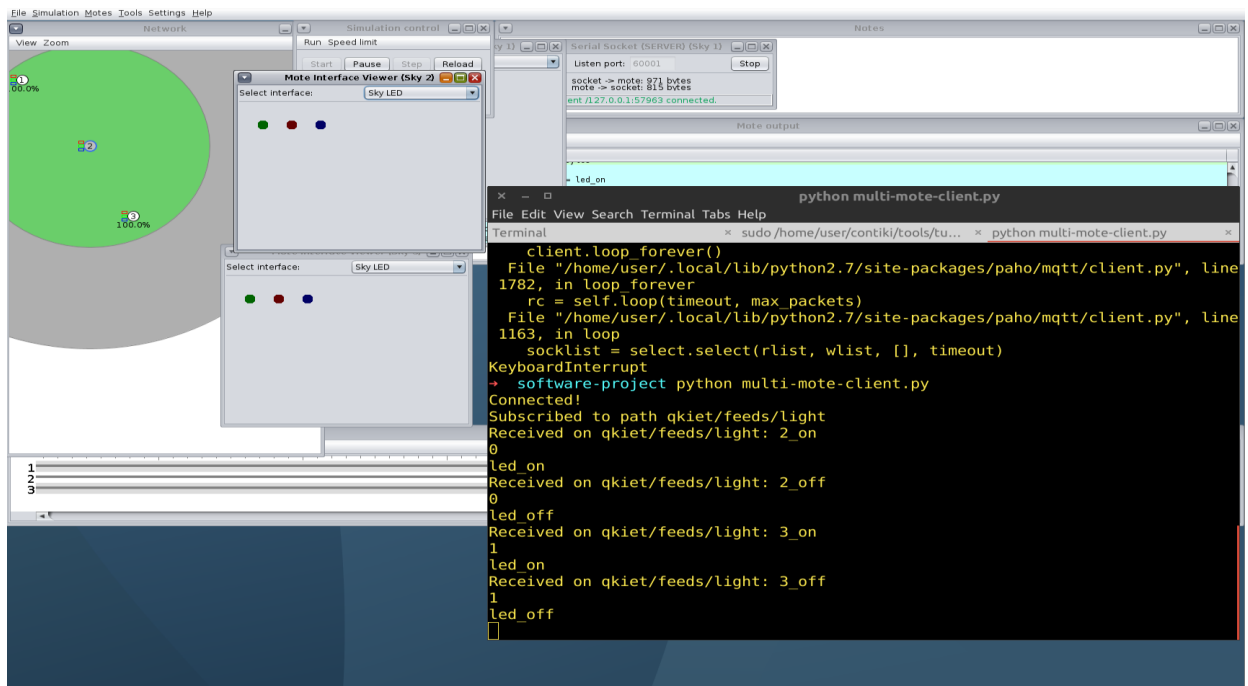
Hình 7: Kết quả sau khi nói "turn on 2"



Hình 8: Kết quả sau khi nói "turn off 2"



Hình 9: Kết quả sau khi nói "turn on 3"



Hình 10: Kết quả sau khi nói "turn off 3"

Tham khảo link youtube sau để có thể quan sát được quá trình Google Assistant điều khiển mạng cảm biến được mô phỏng ra sao: <https://youtu.be/GaGcWuapAZ0>

3.3 Kết luận

Như vậy hệ thống đã hoạt động đúng như những gì kịch bản mô phỏng đề ra và có thể thấy rằng hệ thống hoạt động tốt