

ĐẠI HỌC QUỐC GIA TP. HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA
KHOA ĐIỆN - ĐIỆN TỬ

— * —

BÁO CÁO BÀI TẬP LỚN **MẠNG CẢM BIẾN VÔ TUYẾN**



Sinh viên thực hiện : **Trần Nho Đức**
MSHV : **2170191**

Giảng viên hướng dẫn : **TS. Võ Quế Sơn**

TP. Hồ Chí Minh
Ngày 23 tháng 11 năm 2022

Mục lục

Danh sách hình vẽ	ii
1 Giới thiệu đề tài	4
1.1 Tên đề tài	4
1.2 Yêu cầu	4
2 Thiết kế và triển khai	4
2.1 Mô hình hệ thống	4
2.2 Các bước thực hiện	5
2.2.1 Mô phỏng hệ thống 6LowPAN	5
2.2.2 Tạo Broker trên Adafruit	7
2.2.3 Cấu hình để sử dụng Google Assistant	8
2.2.4 Tạo MQTT Briddge	9
3 Kết quả	9
A Phụ lục	12
A.1 Code chương trình	12
A.1.1 config.conf	12
A.1.2 mqttt-client.c	12

Danh sách hình vẽ

1	Mô hình chung của đề tài	4
2	Sơ đồ hệ thống điều khiển	5
3	Mô phỏng mạng trên Cooja	6
4	Chạy tunslip6 trên Ubuntu	7
5	Các feeds trên Dashboard của Adafruit	7
6	Cấu hình Google Assistant trên IFTTT	8
7	Cấu hình Adafruit trên IFTTT	9
8	Thao tác trên ứng dụng Google Assistant	10
9	Broker nhận được dữ liệu từ Google Assistant thông qua IFTTT	10
10	Đèn của nút mạng số 2 (đèn 1) được bật lên theo đúng như câu lệnh	11

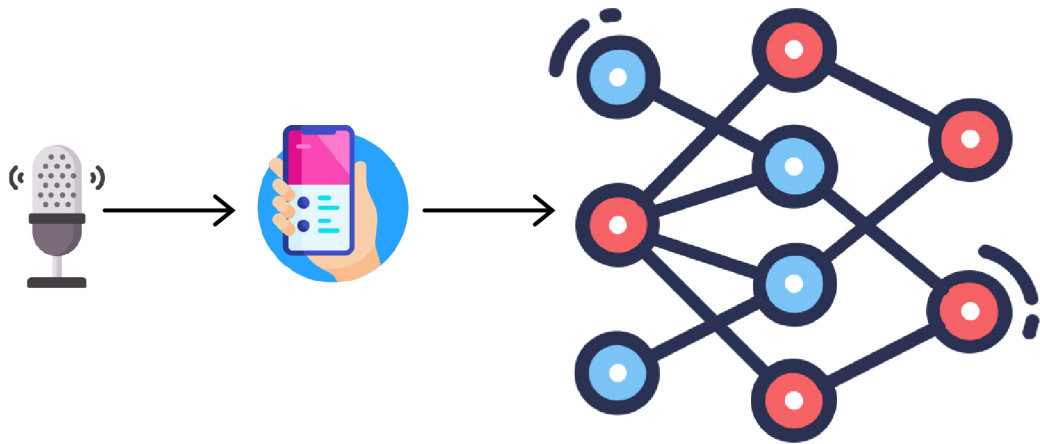
1 Giới thiệu đề tài

1.1 Tên đề tài

Điều khiển mạng cảm biến không dây bằng giọng nói.

1.2 Yêu cầu

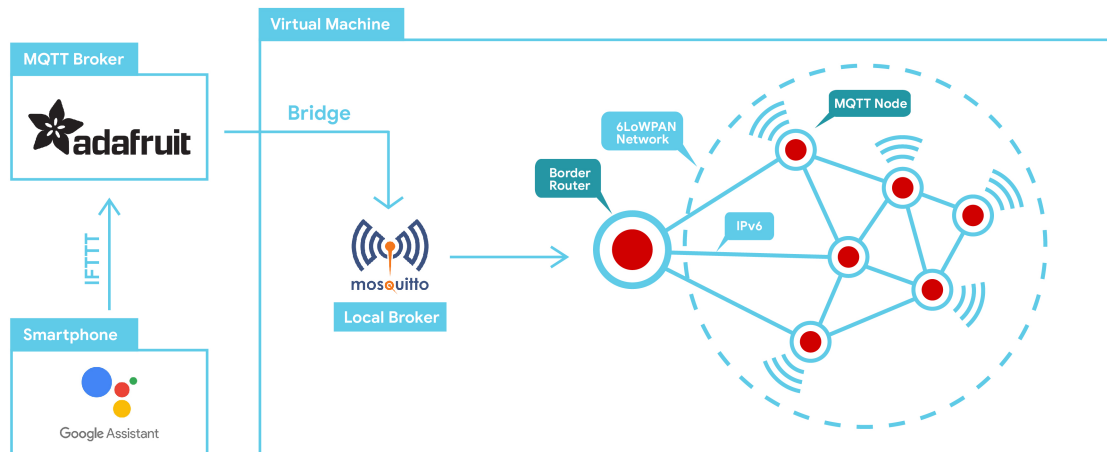
Mạng cảm biến bao gồm các nút mạng sử dụng giao thức 6LoWPAN. Người dùng điều khiển các nút mạng (bật tắt đèn LED) thông qua Google Assistant trên điện thoại thông minh.



Hình 1: Mô hình chung của đề tài

2 Thiết kế và triển khai

2.1 Mô hình hệ thống



Hình 2: Sơ đồ hệ thống điều khiển

Hệ thống bao gồm các thành phần sau:

- Điện thoại thông minh có cài đặt ứng dụng Google Assistant và IFTTT
- MQTT Broker tại Adafruit
- Máy ảo Ubuntu để giả lập hệ thống mạng 6LoWPAN:
 - Local MQTT Broker sử dụng Mosquitto
 - Mạng 6LoWPAN giả lập trên Cooja
 - * 01 nút Border Router kết nối với các nút mạng
 - * Các nút MQTT client kết nối với nhau và với Border Router

Quy trình điều khiển các nút mạng:

1. Người dùng gửi câu lệnh bằng giọng nói thông qua Google Assistant
2. IFTTT là ứng dụng trung gian trên điện thoại di động gửi câu lệnh tới MQTT Broker tại Adafruit
3. Local Broker tại máy ảo sẽ lấy dữ liệu trên Adafruit về và gửi đến cho mạng 6LoWPAN qua MQTT
4. Mỗi nút trong mạng sẽ subscribe một topic riêng, khi nhận được câu lệnh thì các thiết bị sẽ bật/tắt đèn LED

2.2 Các bước thực hiện

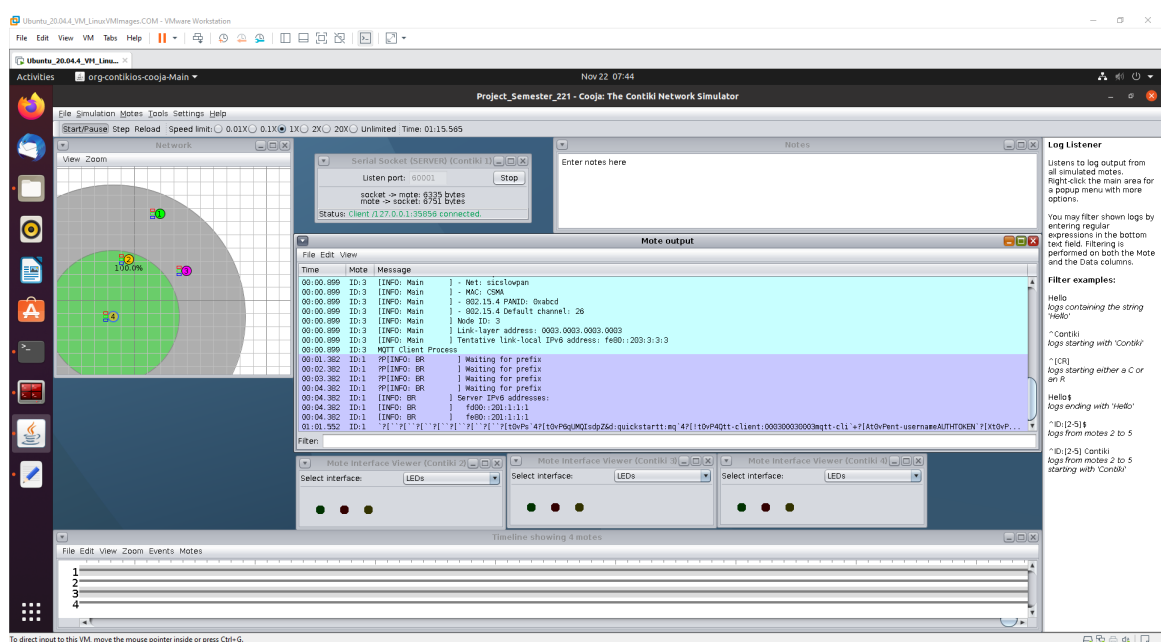
2.2.1 Mô phỏng hệ thống 6LowPAN

Mạng bao gồm các thành phần:

- 01 nút Border Router [1] (*contiki-ng/example/rpl-border-router/rpl-border-router.c*)
- 03 nút MQTT client [2] (*contiki-ng/example/mqtt-client/mqtt-client.c*)

Trog đó, tệp tin mqtt-client.c được chỉnh sửa lại các topic tương ứng trên Adafruit Broker. Chúng ta có các topic mà các nút mạng subscribe lần lượt như sau:

- trannhoduc/feeds/leds/led1 cho nút mạng số 2 (Đèn số 1)
- trannhoduc/feeds/leds/led2 cho nút mạng số 3 (Đèn số 2)
- trannhoduc/feeds/leds/led3 cho nút mạng số 4 (Đèn số 3)

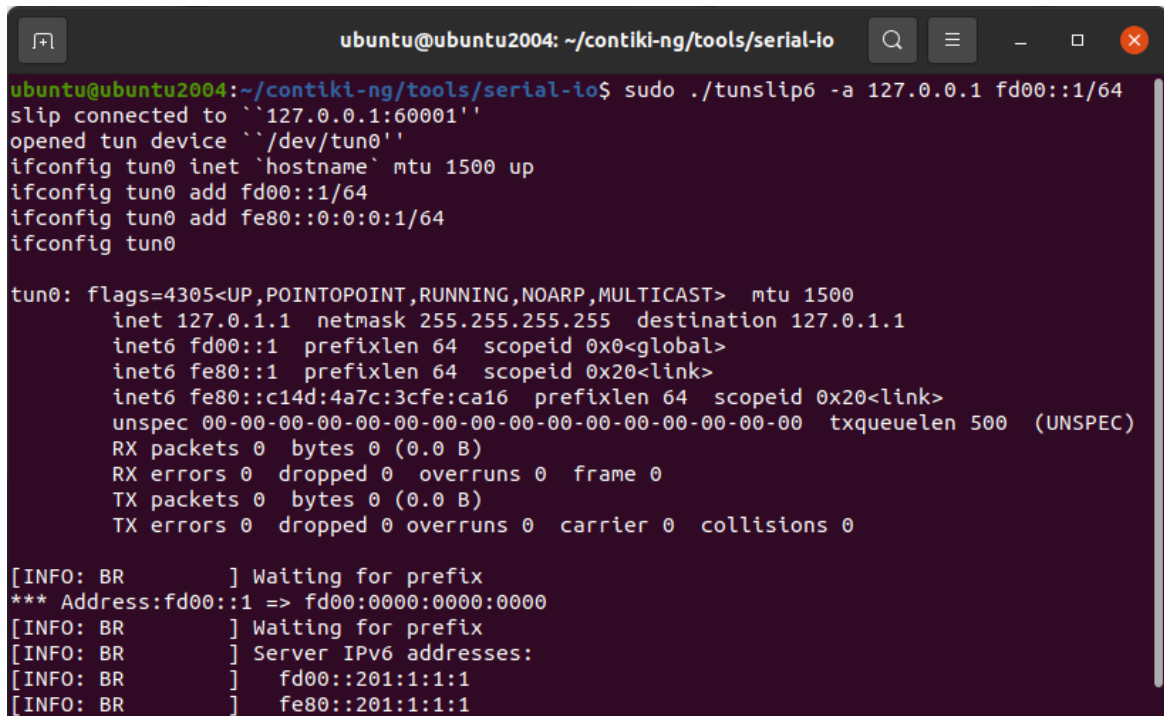


Hình 3: Mô phỏng mạng trên Cooja

Bây giờ, chúng ta cần tạo cầu nối giữa mạng RPL được mô phỏng trên Cooja và máy ảo. Điều này có thể được thực hiện bằng cách nhấp chuột phải vào nút Border Router. Chọn *More tools for border router* rồi chọn *Serial Socket (SERVER)*, chọn port **60001** và bấm **START**.

Ngoài ra, cần phải chạy file tunslip6 để kết nối Border Router với máy ảo. Tại thư mục */contiki-ng/tools/serial-io*, chạy câu lệnh.

```
sudo ./tunslip6 -a 127.0.0.1 fd00::1/64
```



```
ubuntu@ubuntu2004: ~/contiki-ng/tools/serial-io
ubuntu@ubuntu2004:~/contiki-ng/tools/serial-io$ sudo ./tunslip6 -a 127.0.0.1 fd00::1/64
slip connected to `127.0.0.1:60001'
opened tun device `/dev/tun0'
ifconfig tun0 inet `hostname' mtu 1500 up
ifconfig tun0 add fd00::1/64
ifconfig tun0 add fe80::0:0:0:1/64
ifconfig tun0

tun0: flags=4305<UP,POINTOPOINT,RUNNING,NOARP,MULTICAST> mtu 1500
    inet 127.0.0.1 netmask 255.255.255.255 destination 127.0.0.1
    inet6 fd00::1 prefixlen 64 scopeid 0x0<global>
    inet6 fe80::1 prefixlen 64 scopeid 0x20<link>
    inet6 fe80::c14d:4a7c:3cfe:ca16 prefixlen 64 scopeid 0x20<link>
    unspec 00-00-00-00-00-00-00-00-00-00-00-00-00-00-00-00 txqueuelen 500 (UNSPEC)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 0 bytes 0 (0.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

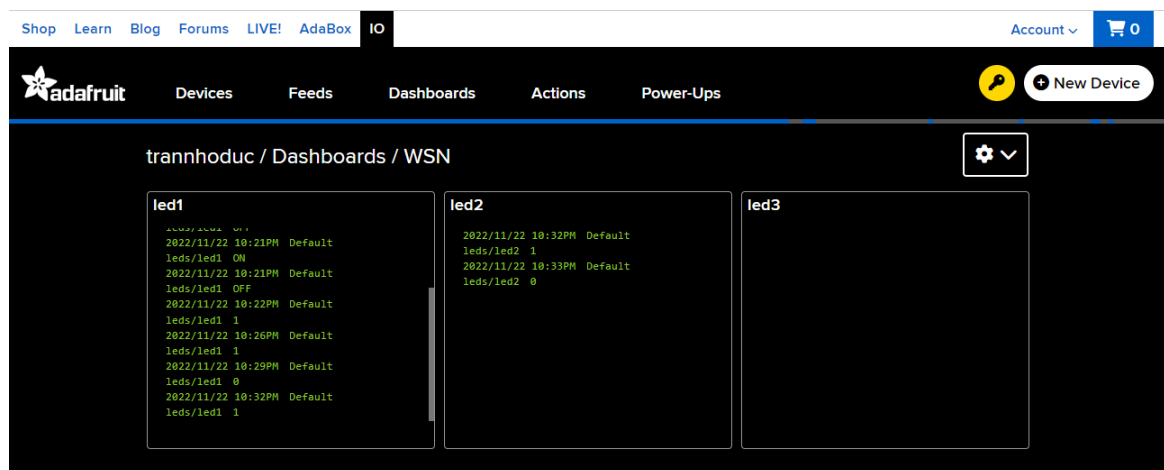
[INFO: BR      ] Waiting for prefix
*** Address:fd00::1 => fd00:0000:0000:0000
[INFO: BR      ] Waiting for prefix
[INFO: BR      ] Server IPv6 addresses:
[INFO: BR      ] fd00::201:1:1:1
[INFO: BR      ] fe80::201:1:1:1
```

Hình 4: Chạy tunslip6 trên Ubuntu

2.2.2 Tạo Broker trên Adafruit

Truy cập trang web <https://io.adafruit.com/>, tạo tài khoản và tạo 03 feed tên là *leds/led1*, *leds/led2* và *leds/led3*.

Chọn *Create a new block*, chọn *Stream* và cho hiển thị 3 feed lên *Dashboard*.

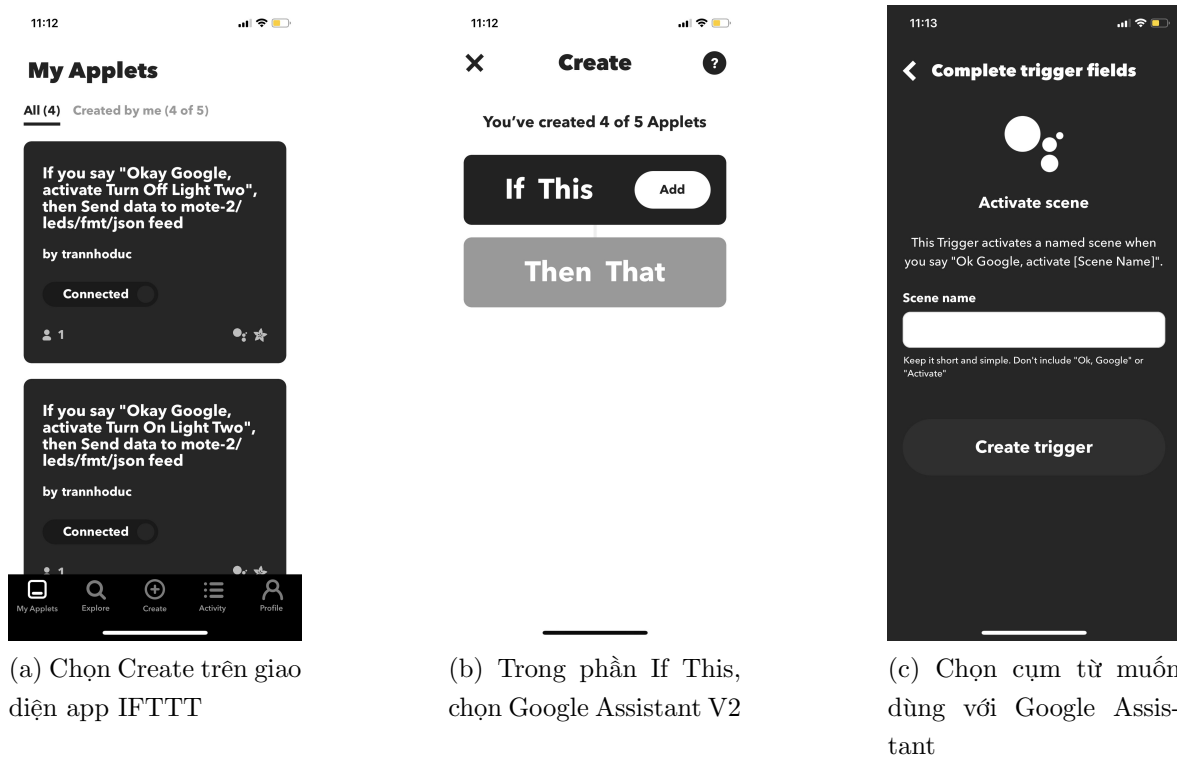


Hình 5: Các feeds trên Dashboard của Adafruit

Các topic mà chúng ta đã tạo lần lượt là *trannhoduc/feeds/leds/led1*, *trannhoduc/feeds/leds/led2* và *trannhoduc/feeds/leds/led3*.

2.2.3 Cấu hình để sử dụng Google Assistant

Sử dụng ứng dụng IFTTT trên AppStore hoặc CHPlay, đăng nhập và bấm vào nút Create.



Hình 6: Cấu hình Google Assistant trên IFTTT

Đối với cụm từ dùng với Google Assistant, đề tài này sử dụng lần lượt là: **Turn On Light 1, Turn On Light 2, Turn Off Light 1, Turn Off Light 2.**

Sau đó, tiếp tục chọn vào phần **Then That** và tìm chọn **Adafruit**. Chọn Feed name là Broker đã tạo trên server của **Adafruit**, chọn lần lượt

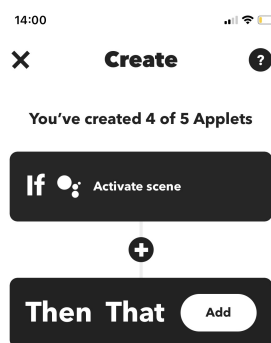
- leds/led1
- leds/led2
- leds/led3

cho lần lượt các nút mạng.

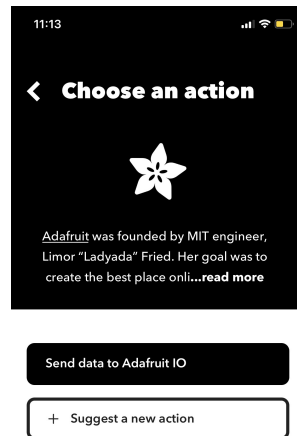
Data to save là nội dung message sẽ được gửi khi kích hoạt câu lệnh từ Google Assistant, chọn lần lượt

- 1 để bật đèn LED
- 0 để tắt đèn LED

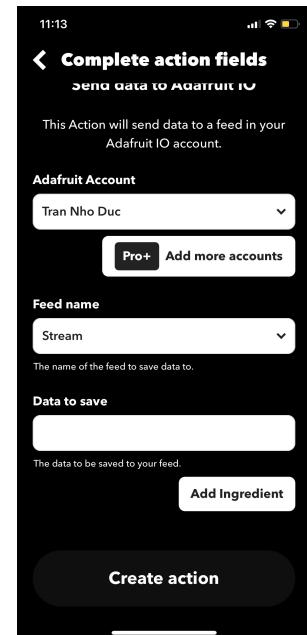
Lưu ý: Topic mà client (smartphone) publish sẽ có dạng: username + "/feeds/" + feedname.



(a) Chọn Add trong phần Then That



(b) Tìm và chọn Adafruit, chọn Send data to Adafruit ID



(c) Chọn Feed name và Data

Hình 7: Cấu hình Adafruit trên IFTTT

2.2.4 Tạo MQTT Briddge

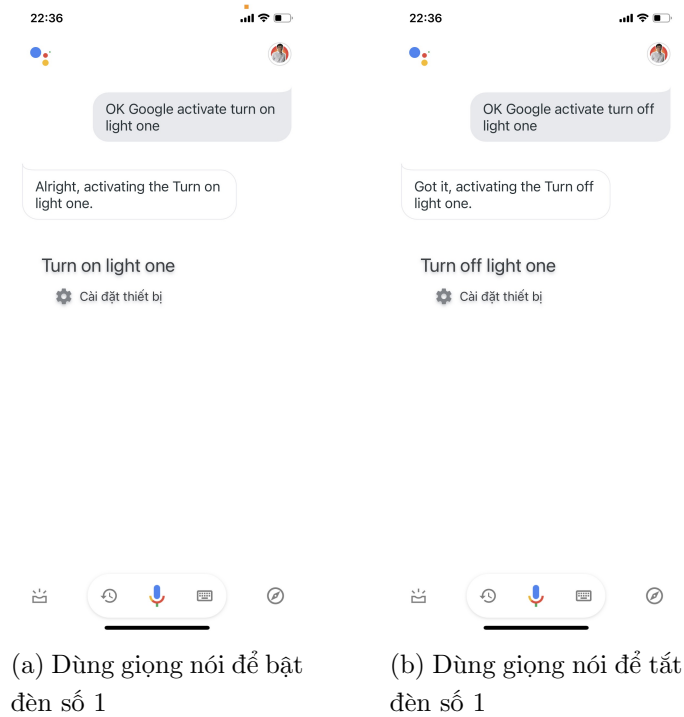
Như đã đề cập ở phần mô hình hệ thống, cần có một client trên máy ảo subscribe những topic từ Adafruit Broker và Publish ngược về lại Local Broker - Broker mà các motes MQTT kết nối.

Mosquitto Broker có hỗ trợ tạo Bridge đơn giản thông qua file *mosquitto.conf* tại thư mục */etc/mosquitto*.

Tạo một file tên là *config.conf* tại thư mục */etc/mosquitto/conf.d/*, có thể sử dụng file mẫu ở địa chỉ */usr/share/doc/mosquitto/examples/mosquitto.conf* để tùy chỉnh lại theo đúng yêu cầu. Tất cả nội dung của file này sẽ tự động được import vào *mosquitto.conf*.

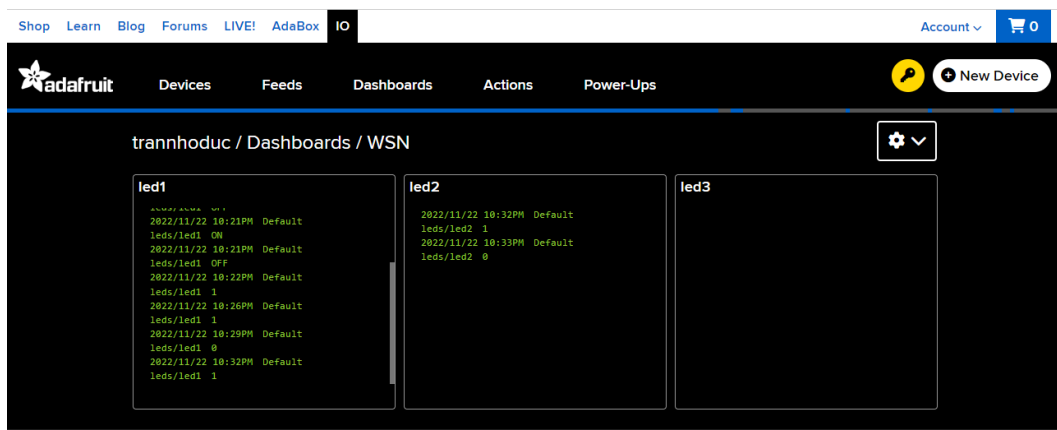
3 Kết quả

Giao diện khi thực hiện thao tác với Google Assistant.



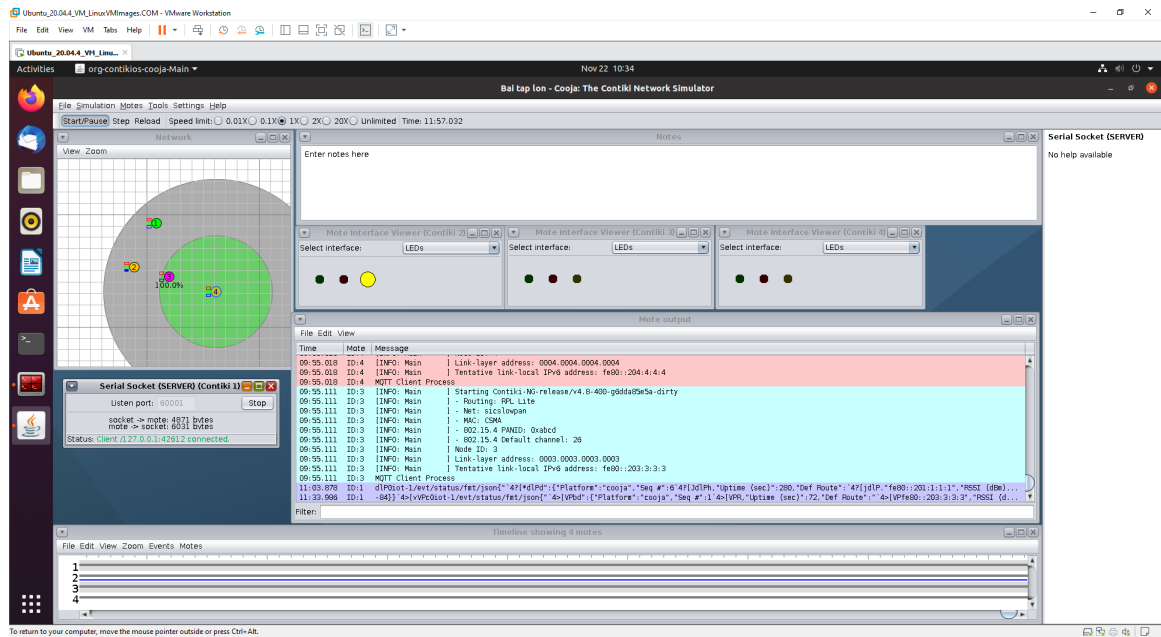
Hình 8: Thao tác trên ứng dụng Google Assistant

MQTT Broker nhận được dữ liệu



Hình 9: Broker nhận được dữ liệu từ Google Assistant thông qua IFTTT

Đèn được bật/tắt theo đúng câu lệnh.



Hình 10: Đèn của nút mạng số 2 (đèn 1) được bật lên theo đúng như câu lệnh

A Phụ lục

A.1 Code chương trình

A.1.1 config.conf

```
address io.adafruit.com:1883
bridge_attempt_unsubscribe false
cleansession false
connection bridge
notifications false
remote_username your_username
remote_password your_password
topic trannhoduc/feeds/leds/+ out
topic trannhoduc/feeds/leds/+ in
try_private true
```

A.1.2 mqtt-client.c

```
static void
pub_handler(const char *topic, uint16_t topic_len, const uint8_t *chunk,
            uint16_t chunk_len)
{
    LOG_DBG("tesst tsssst Pub Handler: topic='%s' (len=%u), chunk_len=%u,
            chunk='%s'\n", topic,
            topic_len, chunk_len, chunk);

    /* If we don't like the length, ignore */
    if(topic_len != 26 || chunk_len != 1) {
        LOG_ERR("Incorrect topic or chunk len. Ignored\n");
        return;
    }

    /* If the format != json, ignore */
    if(strncmp(&topic[topic_len - 4], "led1", 4) != 0) {
        LOG_ERR("Incorrect format\n");
    }

    if(strncmp(&topic[17], "leds", 4) == 0) {
        LOG_DBG("Received MQTT SUB\n");
        if(chunk[0] == '1') {
            leds_on(LED_RED);
        }
    }
}
```

```
    } else if(chunk[0] == '0') {
        leds_off(LED_RED);
    }
    return;
}
}
/*-----*/
static int
construct_sub_topic(void)
{
    int len = snprintf(sub_topic, BUFFER_SIZE, "tranhoduc/feeds/%s/led1",
                      conf.cmd_type);

    /* len < 0: Error. Len >= BUFFER_SIZE: Buffer too small */
    if(len < 0 || len >= BUFFER_SIZE) {
        LOG_INFO("Sub Topic: %d, Buffer %d\n", len, BUFFER_SIZE);
        return 0;
    }

    return 1;
}
```

Tài liệu tham khảo

- [1] Contiki-NG Documentation, “Border router tutotial,” November 2022. <https://docs.contiki-ng.org/en/develop/doc/tutorials/MQTT.html>, Last accessed on 2022-11-1.
- [2] C.-N. Documentation, “Mqtt tutotial,” November 2022. <https://docs.contiki-ng.org/en/develop/doc/tutorials/MQTT.html>, Last accessed on 2022-11-1.