BÁO CÁO THỰC HÀNH

**Bài thực hành số 05: Giải pháp IoT hướng đến bảo mật**

**Môn học:** Bảo mật Internet of Things

**Lớp:** NT535.P21.1

**THÀNH VIÊN THỰC HIỆN (Nhóm xx):**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **STT** | **Họ và tên** | **MSSV** |  | **Điểm tự đánh giá** |
| 1 | Phạm Thiều Gia Khang | 21520967 |  |
| 2 | Phan Văn B | 20520002 |
|  |  |  |  |

**ĐÁNH GIÁ KHÁC:**

|  |  |
| --- | --- |
| Tổng thời gian thực hiện |  |
| Phân chia công việc |  |
| Ý kiến *(nếu có)*  + Khó khăn  + Đề xuất, kiến nghị |  |

Phần bên dưới của báo cáo này là báo cáo chi tiết của nhóm thực hiện

MỤC LỤC

[A. BÁO CÁO CHI TIẾT 3](#_Toc197345277)

[1. Triển khai ThingsBoard (https://thingsboard.io) để thực hiện giám sát, xử lý dữ liệu, trực quan hoá dữ liệu và quản lý thiết bị. Tham khảo hướng dẫn cài đặt tại https://thingsboard.io/docs/user-guide/install/ubuntu/ 3](#_Toc197345278)

[a. Cài đặt Thingsboard: 3](#_Toc197345279)

[2. Xây dựng một giải pháp IoT (có thể sử dụng thiết bị thật hoặc ảo hoá) bao gồm ít nhất 02 thiết bị có khả năng tạo dữ liệu (cảm biến, đọc thẻ RFID,…) và các thiết bị như đèn, LCD, âm thanh, servo,… Sử dụng ThingsBoard để xử lý dữ liệu thu thập được và ra tín hiệu điều khiển thiết bị / cảm biến. 5](#_Toc197345280)

[a. Xây dựng kịch bản: 5](#_Toc197345281)

[b. Code: 7](#_Toc197345282)

[c. Demo: 11](#_Toc197345283)

[B. TÀI LIỆU THAM KHẢO 15](#_Toc197345284)

# BÁO CÁO CHI TIẾT

## Triển khai ThingsBoard (https://thingsboard.io) để thực hiện giám sát, xử lý dữ liệu, trực quan hoá dữ liệu và quản lý thiết bị. Tham khảo hướng dẫn cài đặt tại https://thingsboard.io/docs/user-guide/install/ubuntu/

### Cài đặt Thingsboard:

Cài đặt dựa theo hướng dẫn của nhà phát triển tại: <https://thingsboard.io/docs/user-guide/install/ubuntu/?ubuntuThingsboardQueue=kafka-in-docker>.

Cài đặt PostgreSQL để lưu trữ dữ liệu của Thingsboard trên localhost, tạo người dùng

Postgres với password là 1700689350:

sudo -u postgres psql -c "\password"

Kết nối postgres database với user vừa tạo, tạo database Thingsboard:

psql -U postgres -d postgres -h 127.0.0.1 –W

CREATE DATABASE thingsboard;

Cấu hình file thingsboard.conf để nhận SQL server:

#DB Configuration

export DATABASE\_TS\_TYPE=sql

export SPRING\_DATASOURCE\_URL=jdbc:postgresql://localhost:5432/thingsboard

export SPRING\_DATASOURCE\_USERNAME=postgres

export SPRING\_DATASOURCE\_PASSWORD=1700689350

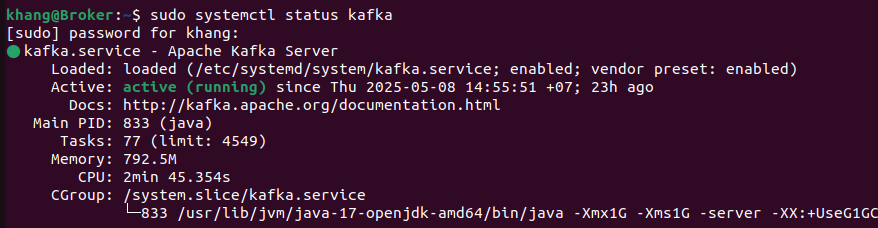
# Specify partitioning size for timestamp key-value storage. Allowed values: DAYS, MONTHS, YEARS, INDEFINITE.

export SQL\_POSTGRES\_TS\_KV\_PARTITIONING=MONTHS

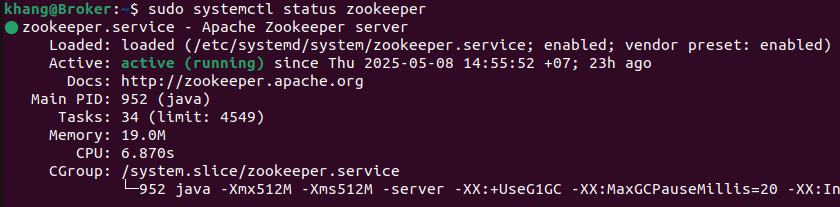
Đối với nền tảng triển khai Thingsboard nhóm chọn Kafka vì đây là vì nó khả năng mở rộng tốt cũng như tốc độ xử lý dữ liệu thời gian thực cao. Đây cũng là lựa chọn được xử dụng trong các hẹ thống IoT lớn:

Cài đặt và triển khai Zookeeper và Kafla trên máy:

* Kafka:



* Zookeeper:



Mặc định địa chỉ của Kafka server khi được cài đặt là localhost:9092, điều chỉnh file thingboard.conf để tham chiếu tới Kafka:

export TB\_QUEUE\_TYPE=kafka

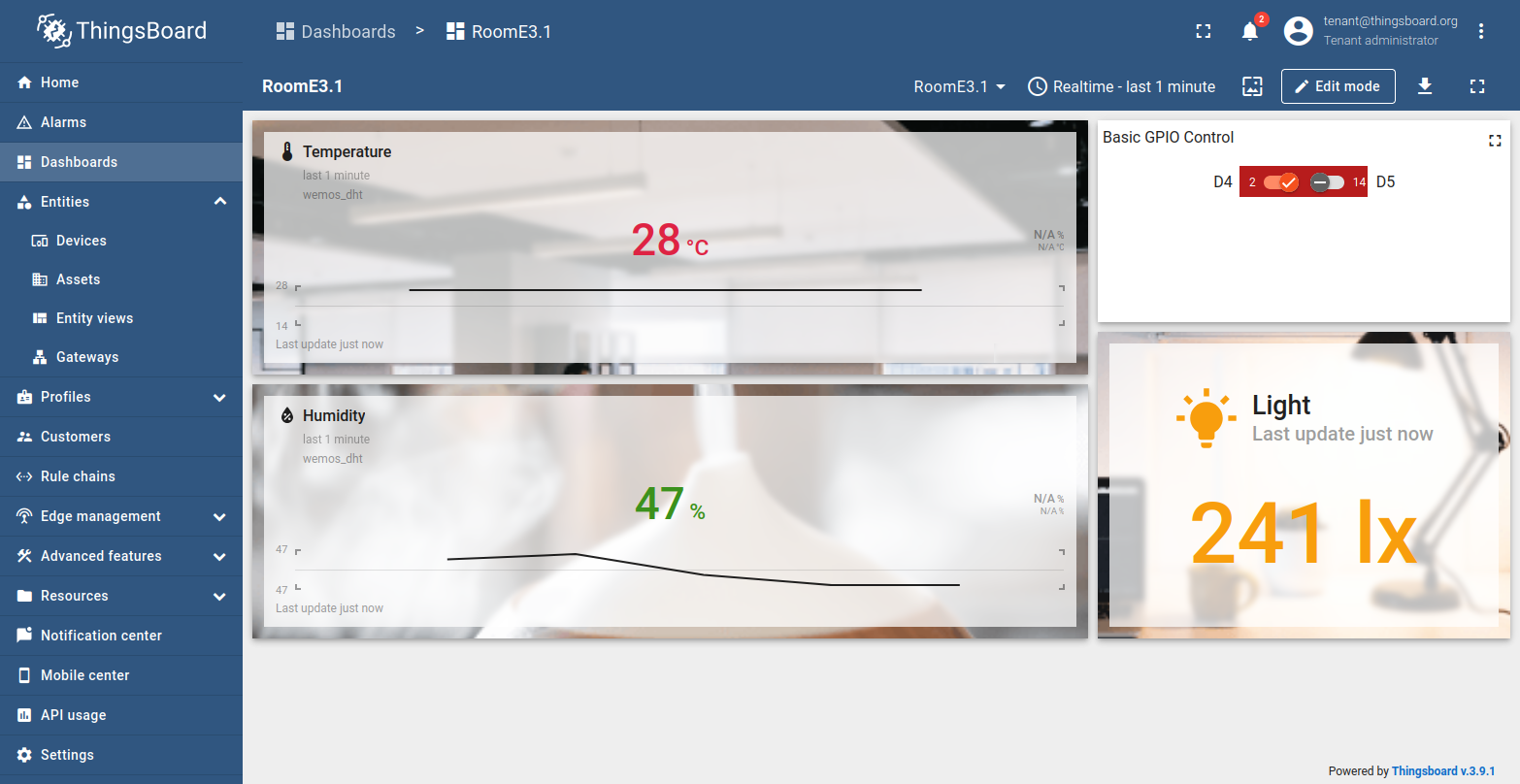
export TB\_KAFKA\_SERVERS=localhost:9092

export MQTT\_BIND\_ADDRESS=0.0.0.0

Kích hoạt thingsboard:

$ sudo systemctl start thingsboard

Đăng nhập vào bằng tài khoản: Tenant Administrator: tenant@thingsboard.org / tenant



Trong file cài đặt mặc định của Thingsboard (thingboard.yml), thingsboard sẽ mở cổng kết nối đến bằng giao thức MQTT bằng port 1883 cho các thiết bị IoT:

# Local MQTT transport parameters

mqtt:

# Enable/disable mqtt transport protocol.

enabled: "${MQTT\_ENABLED:true}"

# MQTT bind-address

bind\_address: "${MQTT\_BIND\_ADDRESS:0.0.0.0}"

# MQTT bind port

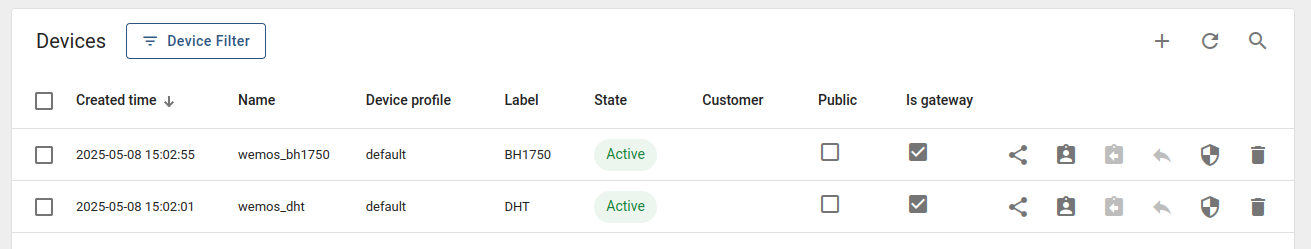
bind\_port: "${MQTT\_BIND\_PORT:1883}"

## Xây dựng một giải pháp IoT (có thể sử dụng thiết bị thật hoặc ảo hoá) bao gồm ít nhất 02 thiết bị có khả năng tạo dữ liệu (cảm biến, đọc thẻ RFID,…) và các thiết bị như đèn, LCD, âm thanh, servo,… Sử dụng ThingsBoard để xử lý dữ liệu thu thập được và ra tín hiệu điều khiển thiết bị / cảm biến.

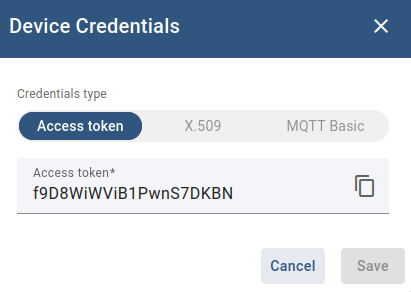
### Xây dựng kịch bản:

Kịch bản sẽ bao gồm 2 Wemos và 1 Host chạy Thingsboard, Node 1 sẽ đo cường độ ánh sang và nhận tín hiệu điều khiển đèn từ Platform, Node 2 sẽ đo nhiệt độ độ ẩm.

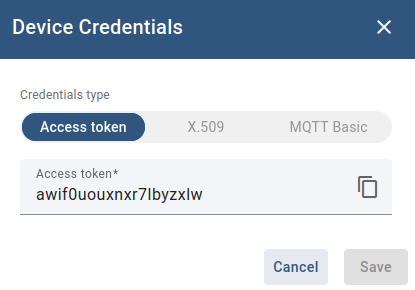
Tạo 1 2 Device Wemos\_light và Wemos\_dht trên Thingsboard, sử dụng Access Token được tạo ra để kết nối 2 thiết bị tới Platform:



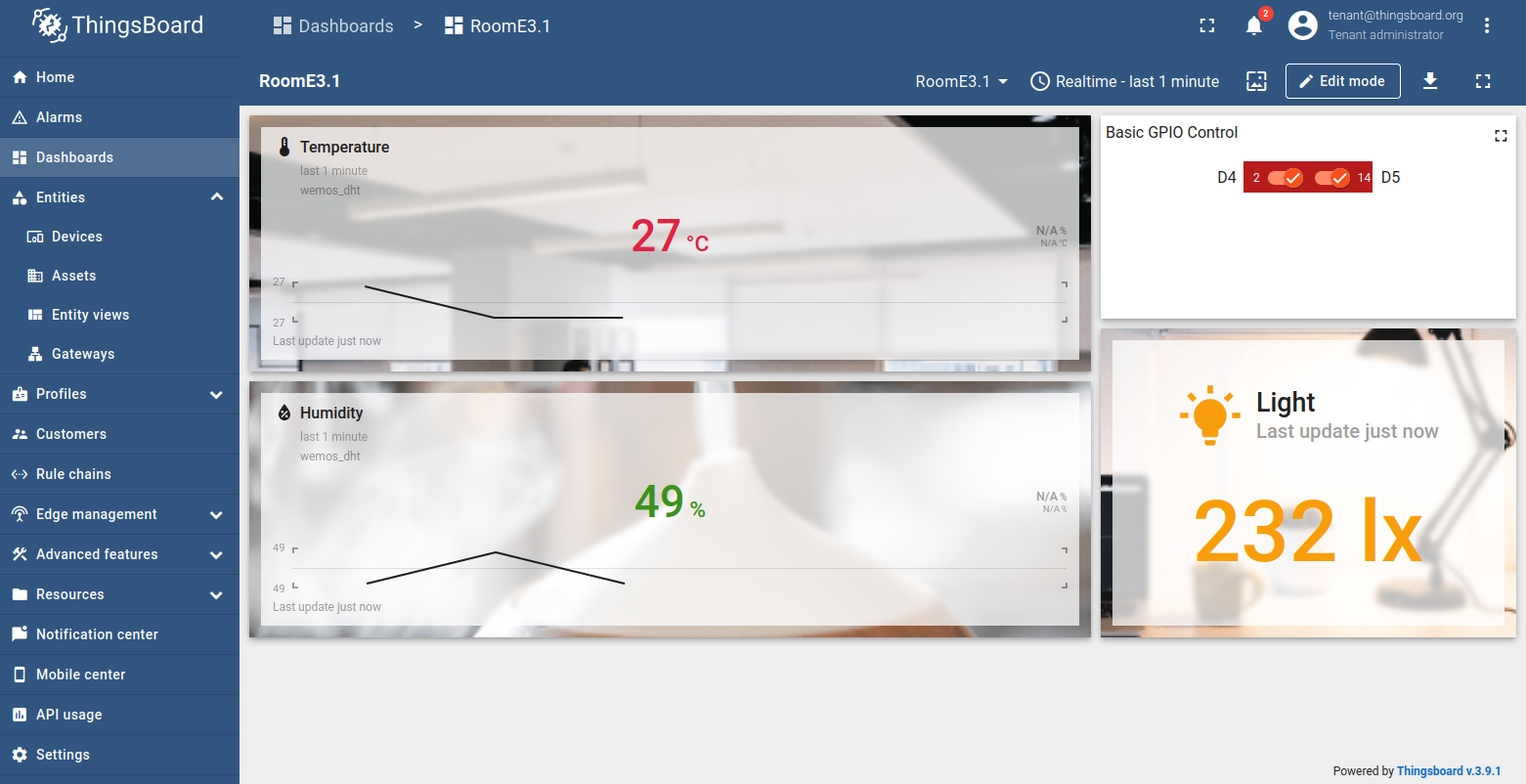
Wemos\_bh1750:



Wemos\_dht:

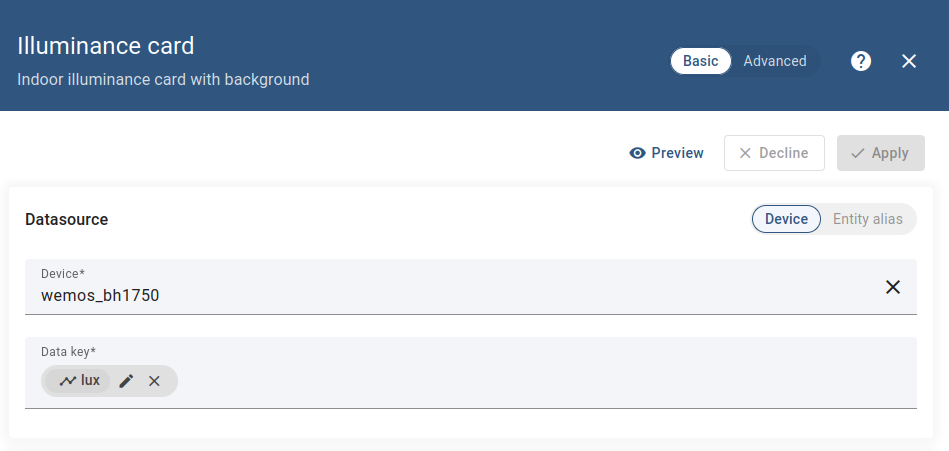


Tạo một Dashboard gồm các thành phần như Line chart, Tempure card, Humidity card, Basic GPIO Control, Indoor horizontal illuminance card và total device:

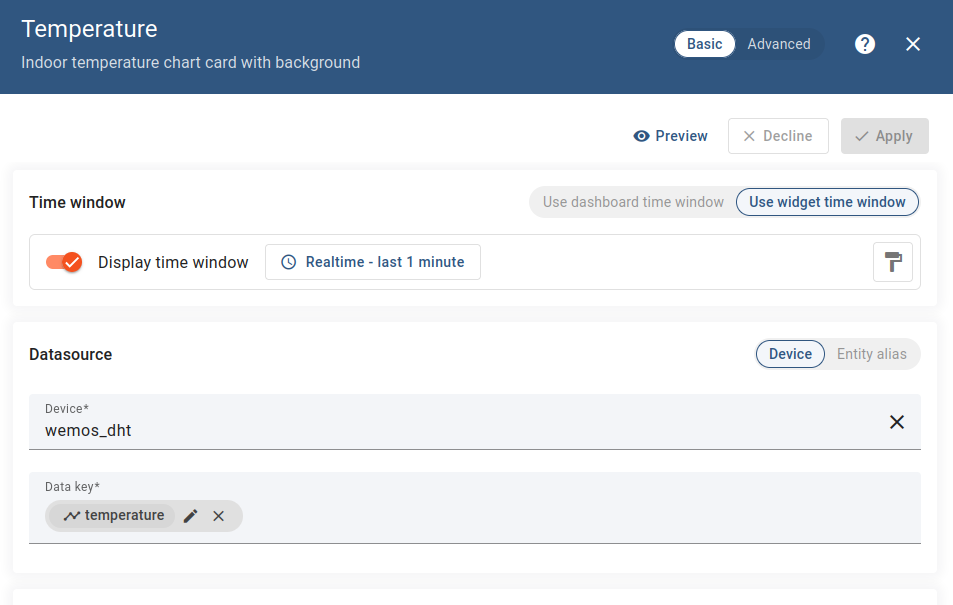


Line Chart và các card tương ứng sẽ được sử dụng để hiển thị giá trị thực tế và theo thời gian, Basic GPIO dùng để điều khiển đèn.

Cấu hình trường “lux” trên illuminance card:



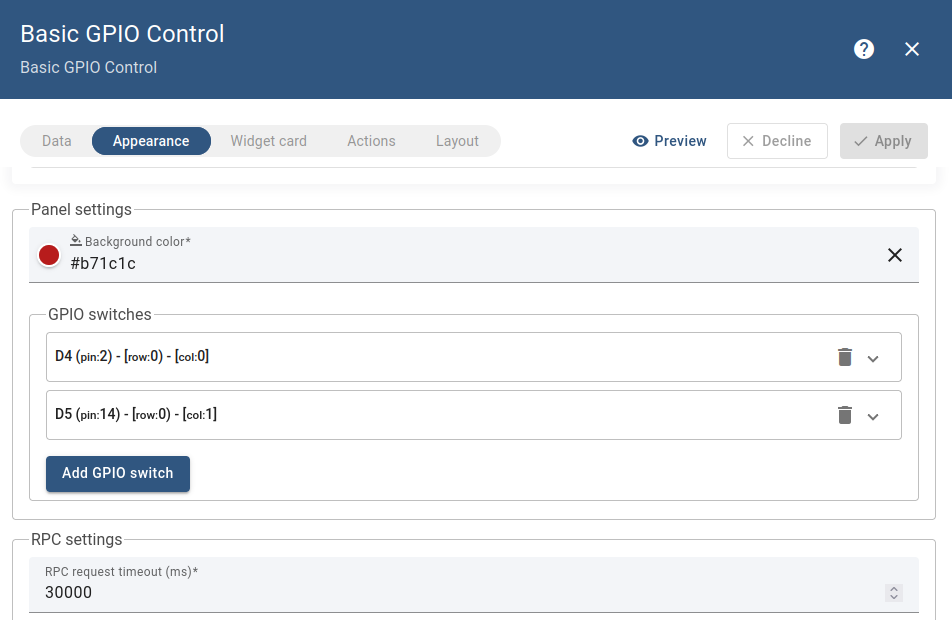
Cấu hình trường “temperature” trên temperature card:



Cấu hình trường “humidity” trên Huminity card:



Thêm 2 cổng D4 (GPIO2) và D5 (GPIO14) trên Basic GPIO Controll để điều khiển đèn:



### Code:

Trên Wemos\_dht:

* Wemos sẽ kết nối với Thingsboard qua địa chỉ 192.168.40.43:1883 bằng token: "ze8vxz8u3yi8lhc0et7k"

// Địa chỉ truy cập Thingsboard

const char\* ssid = "WemosTest";

const char\* password = "01072003";

const char\* thingsboardServer = "192.168.43.40";

const int thingsboardPort = 1883;

const char\* accessToken = "ze8vxz8u3yi8lhc0et7k";

void reconnect() {

while (!client.connected()) {

Serial.println("Connecting to ThingsBoard...");

if (client.connect("Wemos\_dht", accessToken, nullptr)) {

Serial.println("Connected to ThingsBoard");\

} else {

Serial.print("Failed to connect. Retrying in 5 seconds...");

delay(5000);

}

}

}

* Dữ liệu từ DHT22 sẽ được đóng gói thành dạng Json và publish tới topic: ("v1/devices/me/telemetry+"

// Đọc giá trị cảm biến

float h = dht.readHumidity();

float t = dht.readTemperature();

// Tạo gói Json chứa các giá trị

JsonDocument doc;

doc["temperature"] = t;

doc["humidity"] = h;

serializeJson(doc, Serial);

Serial.println();

// ĐÓng gói và publish gói json

String payload = "";

serializeJson(doc, payload);

client.publish("v1/devices/me/telemetry", payload.c\_str());

delay(10000);

}

Trên Wemos\_bh1750:

* Wemos sẽ kết nối với Thingsboard qua địa chỉ 192.168.40.43:1883 bằng token: "ti0v70ox6a118jrm0mph"

// Truy cập vào Thingsboard

const char\* ssid = "WemosTest";

const char\* password = "01072003";

const char\* thingsboardServer = "192.168.43.40";

const int thingsboardPort = 1883;

const char\* accessToken = "ti0v70ox6a118jrm0mph";

void reconnect() {

while (!client.connected()) {

Serial.println("Connecting to ThingsBoard...");

if (client.connect("Wemos", accessToken, nullptr)) {

Serial.println("Connected to ThingsBoard");

client.subscribe("v1/devices/me/rpc/request/+");

client.subscribe("v1/devices/me/attributes/response/+");

Serial.println("Sending current GPIO status ...");

client.publish("v1/devices/me/attributes",get\_gpio\_status().c\_str());

} else {

Serial.print("Failed to connect. Retrying in 5 seconds...");

delay(5000);

}

}

}

Kết nối tới Server và đăng ký nhận Callback:

client.setServer(thingsboardServer, thingsboardPort);

client.setCallback(onMessage);

Đóng gói dữ liệu từ BH1750 và publish đến topic: ("v1/devices/me/telemetry/+"

void loop() {

if (!client.connected()) {

reconnect();

}

client.loop();

float lux = lightMeter.readLightLevel();

Serial.println(lux);

String payload = "{\"lux\": " + String(lux) + "}";

client.publish("v1/devices/me/telemetry", payload.c\_str());

delay(10000);

}

Hàm OnMessage sẽ giải mã gói tin từ Thingsboard khi có hành động thay đổi tín hiệu đèn trên Basic GPIO:

void onMessage(const char\* topic, byte\* payload, unsigned int length) {

Serial.println("On recieved message");

char json[length + 1];

strncpy(json, (char\*)payload, length);

json[length] = '\0';

Serial.print("Topic: ");

Serial.println(topic);

Serial.print("Message: ");

Serial.println(json);

// Giả mã gói Json Request

DynamicJsonDocument jsonBuffer(200);

DeserializationError error = deserializeJson(jsonBuffer, json);

if (error) {

Serial.print("deserializeJson() failed: ");

Serial.println(error.c\_str());

return;

}

// In gói request ra Serial

String methodName = String((const char\*)jsonBuffer["method"]);

if (methodName.equals("getGpioStatus")) {

// Reply with GPIO status

String responseTopic = String(topic);

responseTopic.replace("request", "response");

String gpioStatus = get\_gpio\_status();

client.publish(responseTopic.c\_str(), gpioStatus.c\_str());

}

else if (methodName.equals("setGpioStatus")) {

// Update GPIO status và reply

int pin = jsonBuffer["params"]["pin"];

bool enabled = jsonBuffer["params"]["enabled"];

set\_gpio\_status(pin, enabled);

String responseTopic = String(topic);

responseTopic.replace("request", "response");

String gpioStatus = get\_gpio\_status();

client.publish(responseTopic.c\_str(), gpioStatus.c\_str());

client.publish("v1/devices/me/attributes", gpioStatus.c\_str());

}

}

Gửi trạng thái hiện tại của 2 chân D4 và D5 đến Thingsboard để đồng bộ:

String get\_gpio\_status() {

// Chuẩn bị gpios JSON payload string

DynamicJsonDocument jsonBuffer(200);

JsonObject data = jsonBuffer.to<JsonObject>();

data[String(GPIO2\_PIN)] = gpioState[0] ? true : false;

data[String(GPIO14\_PIN)] = gpioState[1] ? true : false;

String strPayload;

serializeJson(data, strPayload);

Serial.print("Get gpio status: ");

Serial.println(strPayload);

return strPayload;

}

void set\_gpio\_status(int pin, boolean enabled) {

if (pin == GPIO2\_PIN) {

// Output GPIOs state

digitalWrite(GPIO2, enabled ? HIGH : LOW);

// Update GPIOs state

gpioState[0] = enabled;

} else if (pin == GPIO14\_PIN) {

// Output GPIOs state

digitalWrite(GPIO14, enabled ? HIGH : LOW);

// Update GPIOs state

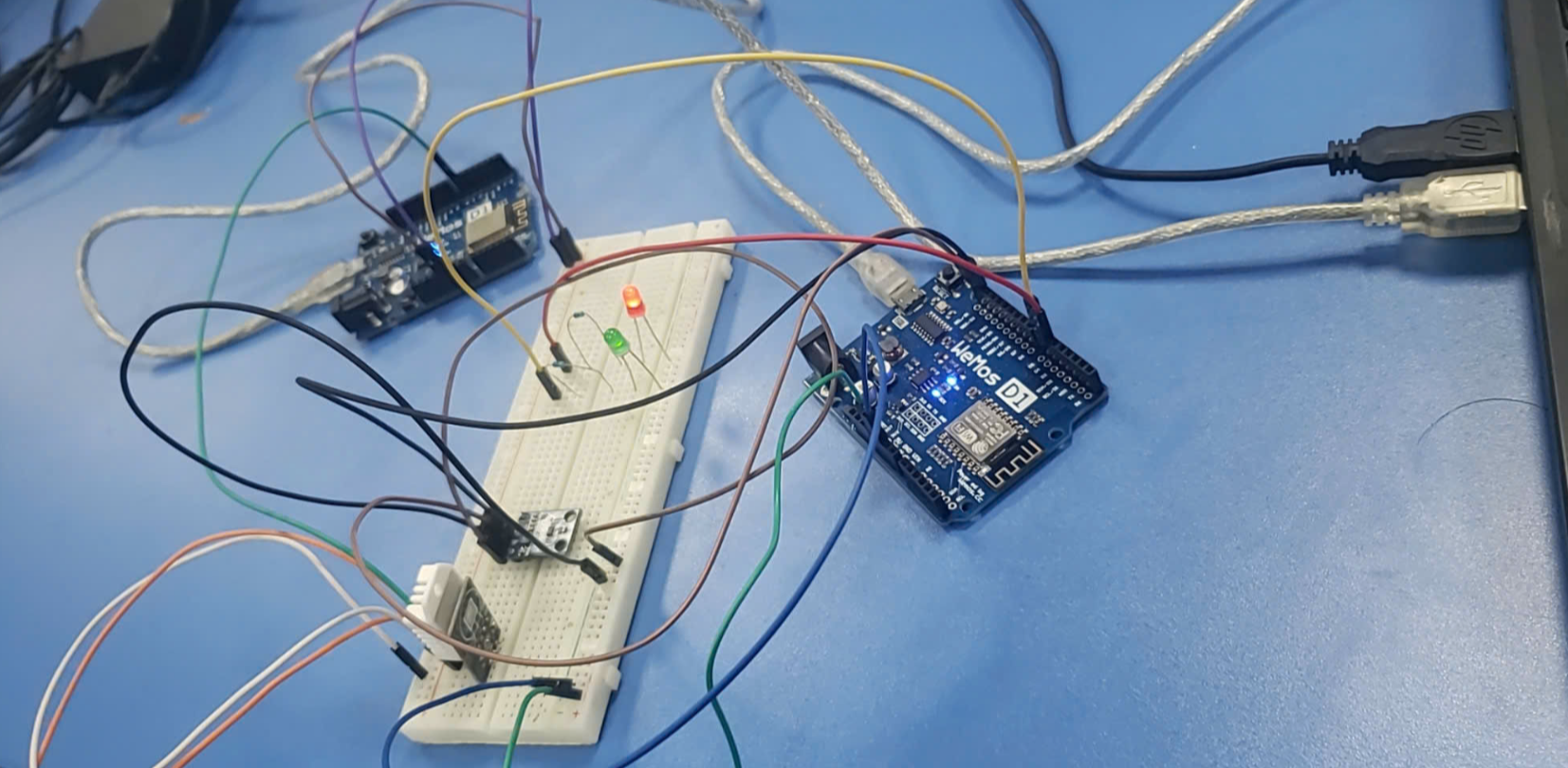
gpioState[1] = enabled;

}

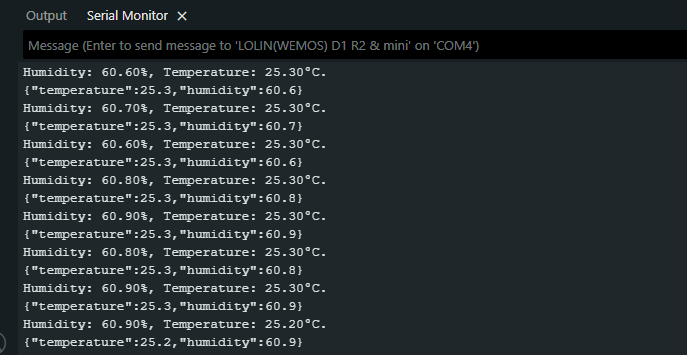
}

### Demo:

Tổng quan mô hình:



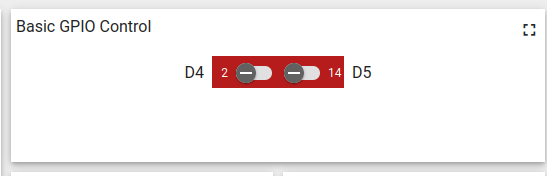
Các cảm biến sẽ liên tục gửi dữ liệu đến Thingsboard sau mỗi 10s:

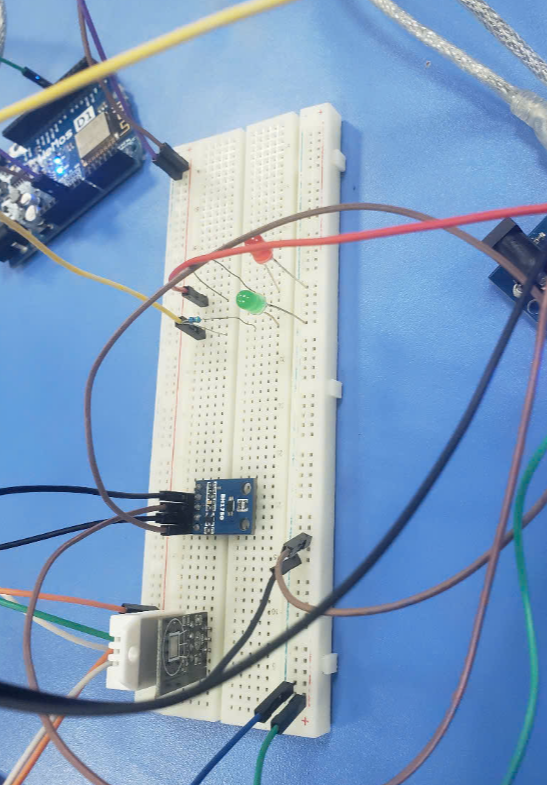




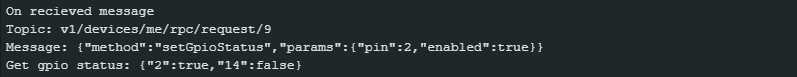
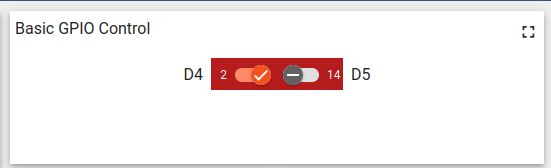
Khi bật tắt đèn trên Platform sẽ in ra gói request trên Serial và điều chỉnh đèn tương ứng:

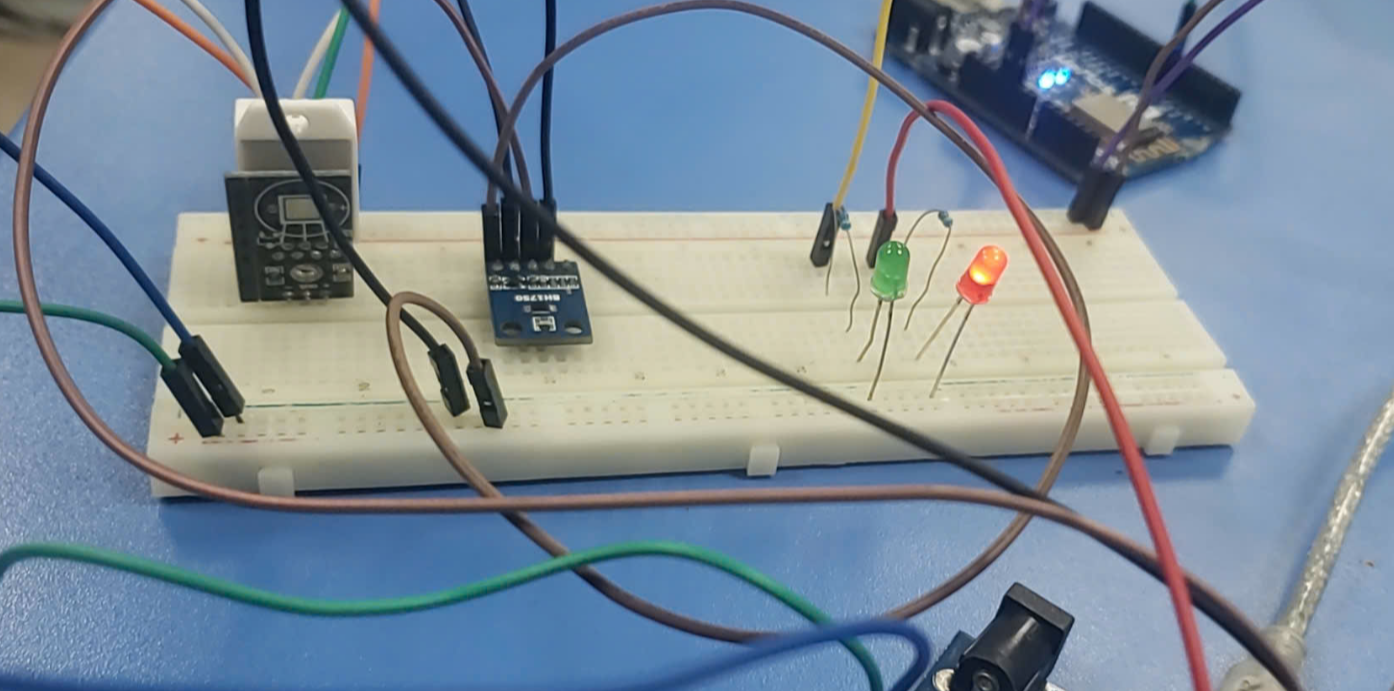
Khi chưa bật:



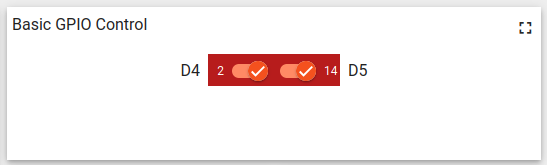


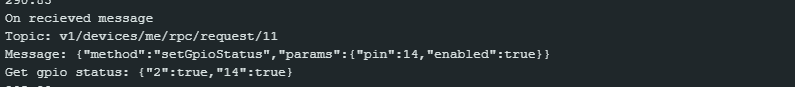
Khi bật D4:

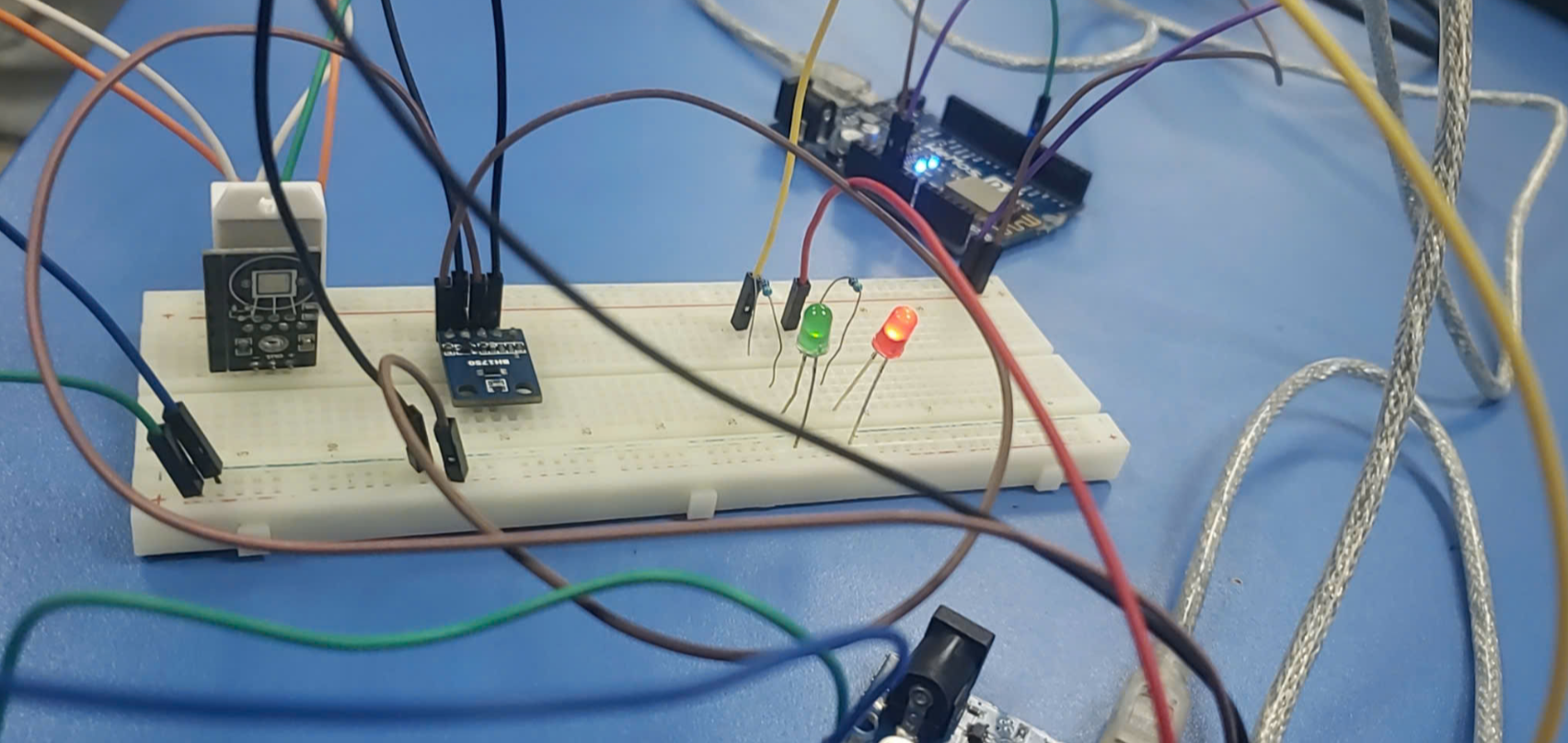




Khi bật D5:







Chi tiết video demo tại đây: https://drive.google.com/drive/folders/1QNJnq-nDIibQMRxdFCL9Av-zv3dd3AkT?usp=sharing

# TÀI LIỆU THAM KHẢO