



BỘ MÔN KỸ THUẬT MÁY TÍNH – VIỄN THÔNG
CƠ SỞ VÀ ỨNG DỤNG IOTS
MMH: ITFA436064
Thời gian thực hiện: 2 buổi

Họ và tên:Vũ Tiến Phát - 21151309

Hoàng Thị Diễm Quỳnh - 21151326

Nguyễn Nam Huy - 21151244

Ngô Xuân Thọ - 21119376

Trương Nguyễn Quốc Thắng - 22119231

1. Giới thiệu Google Firebase RTDB

Google Firebase Realtime Database (Firebase RTDB) là một dịch vụ cơ sở dữ liệu thời gian thực và đồng bộ của Google Firebase, một nền tảng phát triển ứng dụng di động và web. Firebase RTDB được thiết kế để cung cấp một cơ sở dữ liệu mạnh mẽ, linh hoạt và dễ sử dụng cho việc phát triển các ứng dụng đa nền tảng.

Firebase RTDB là một cơ sở dữ liệu được lưu trữ dưới dạng cây JSON. Cấu trúc cây JSON cho phép tổ chức dữ liệu một cách dễ dàng và linh hoạt, giúp phát triển ứng dụng trở nên thuận tiện. Cây JSON có thể chứa các object và mảng, và dữ liệu có thể được truy cập, cập nhật và đồng bộ hóa từ cả thiết bị di động và máy chủ.

Firebase RTDB sử dụng mô hình đồng bộ thời gian thực, cho phép dữ liệu được truyền và đồng bộ hóa ngay lập tức giữa các thiết bị và máy chủ. Điều này giúp các ứng dụng hiển thị dữ liệu mới nhất và phản hồi nhanh chóng đối với các thay đổi dữ liệu từ các nguồn khác nhau. Firebase RTDB sử dụng kết nối WebSocket để duy trì liên kết liên tục và truyền dữ liệu một cách hiệu quả.

Firebase RTDB cung cấp các tính năng mạnh mẽ như xử lý sự kiện thời gian thực, theo dõi sự kiện thay đổi dữ liệu và tự động cập nhật giao diện người dùng. Bằng cách sử dụng mô hình “listener”, ứng dụng có thể nhận thông báo về các thay đổi dữ liệu và tự động cập nhật giao diện người dùng mà không cần phải thực hiện các yêu cầu tải lại trang.

Firebase RTDB cũng cung cấp tính năng bảo mật để kiểm soát quyền truy cập vào dữ liệu. Người phát triển có thể xác định quyền truy cập dựa trên người dùng, nhóm người dùng hoặc quy tắc tùy chỉnh. Điều này giúp bảo vệ dữ liệu và đảm bảo rằng chỉ những người được ủy quyền mới có thể truy cập và thay đổi dữ liệu.

Ngoài ra, Firebase RTDB tích hợp với các dịch vụ khác của Firebase như Firebase Authentication, Firebase Cloud Functions và Firebase Hosting. Điều này giúp phát triển ứng dụng một cách nhanh chóng và dễ dàng, kết hợp các tính năng như xác thực người dùng, xử lý logic phía máy chủ và việc lưu trữ và đồng bộ dữ liệu thời gian thực.

Tóm lại, Google Firebase Realtime Database (Firebase RTDB) là một dịch vụ cơ sở dữ liệu thời gian thực mạnh mẽ và linh hoạt cho việc phát triển ứng dụng di động và web. Với việc lưu trữ dữ liệu dưới dạng cây JSON và tính năng đồng bộ thời gian thực, Firebase RTDB cho phép xây dựng các ứng dụng tương tác và đáp ứng nhanh chóng.

2. Các bước thực hiện, giải thích code quá trình cập nhật dữ liệu lên Google Firebase và hình ảnh kết quả thực hiện (video clip demo nếu có).

B1: truy cập web của firebase và tạo project

[Firebase | Google's Mobile and Web App Development Platform](#)

Chọn Go to console -> Add project -> Đặt tên project -> continue

Tại mục Choose or create a Google Analytics account chọn Default account for firebase

Nhấn Create Project.

B2: Tạo Realtime Database

B3: Thêm các thư viện của Firebase trong Arduino IDE

B4: Lập trình trên Arduino IDE

```
//thêm các thư viện firebase, dht, wifi
#include <FirebaseESP32.h>
#include "DHTesp.h"
#include <WiFi.h>

const char* API_KEY = "0hqXDGSnqYNIEfGfL2ISoMA1Ouvh1sOhpkVCnmVR"; // key của
firebase
const char* DATABASE_URL = "https://lab3-143c7-default-rtdb.firebaseio.com/";// url
của firebase
//Cấu hình ssid và mật khẩu wifi để esp32 kết nối
const char* ssid = "Wokwi-GUEST";
const char* password = "";
const int DHT_PIN = 27;//khai báo chân dht kết nối là 27
DHTesp dhtSensor;// biến dữ liệu của dht
WiFiClient espClient;
//Các biến dữ liệu của firebase
FirebaseJson json;
FirebaseData fbdo;
FirebaseAuth auth;
FirebaseConfig config;
//Hàm cấu hình kết nối wifi cho esp32
void setup_wifi()
{
    Serial.println("Connecting to SSID: Wokwi-GUEST");
    WiFi.begin("Wokwi-GUEST", "");
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.print("WiFi connected");
    Serial.print(", IP address: ");
}
```

```

    Serial.println(WiFi.localIP());
}

//Các cấu hình sử dụng trên esp32
void setup() {
    Serial.begin(115200);
    setup_wifi();
    dhtSensor.setup(DHT_PIN, DHTesp::DHT22);
    config.database_url = DATABASE_URL;
    config.signer.tokens.legacy_token = API_KEY;
    Firebase.begin(&config, &auth);
    fbdo.setResponseSize(8192);
    Firebase.reconnectWiFi(true);
}

void loop() {
    TempAndHumidity data = dhtSensor.getTempAndHumidity(); // đọc dữ liệu từ dht22
    Serial.println("Temp: " + String(data.temperature, 1) + "°C"); // in nhiệt độ
    Serial.println("Humidity: " + String(data.humidity, 1) + "%"); // in độ ẩm
    Serial.println("----");
    Firebase.setInt(fbdo, "Nhiet do", data.temperature); // Gửi nhiệt độ lên Firebase
    Firebase.setInt(fbdo, "Do am", data.humidity); // gửi độ ẩm lên firebase
    delay(1000);
}

```

Kết quả: [\[IoT\] Lab 3: Cập nhật dữ liệu lên Google Firebase](#)

3. Giới thiệu Thingspeak

_ ThingSpeak là một nền tảng dành cho Internet of Things (IoT) được phát triển bởi MathWorks, một công ty công nghệ toàn cầu chuyên về phần mềm và công cụ phát triển cho các lĩnh vực khoa học và kỹ thuật.

_ ThingSpeak cho phép người dùng thu thập, lưu trữ, phân tích và hiển thị dữ liệu từ các thiết bị IoT. Nền tảng này cung cấp một giao diện đơn giản để người dùng tạo và quản lý các kênh (channels) để thu thập dữ liệu từ các cảm biến hoặc các nguồn dữ liệu khác. Mỗi kênh trong ThingSpeak có thể chứa nhiều lĩnh vực (fields), và mỗi lĩnh vực đại diện cho một loại dữ liệu cụ thể như nhiệt độ, độ ẩm, ánh sáng, v.v.

_ Người dùng có thể sử dụng giao diện web hoặc các API cung cấp bởi ThingSpeak để gửi dữ liệu từ các thiết bị IoT lên kênh của mình. Dữ liệu được lưu trữ trong các bản ghi (entries) và có thể được truy vấn và trực quan hóa bằng cách sử dụng các biểu đồ và đồ thị được tích hợp sẵn trong ThingSpeak.

_ Thingspeak gồm các tính năng:

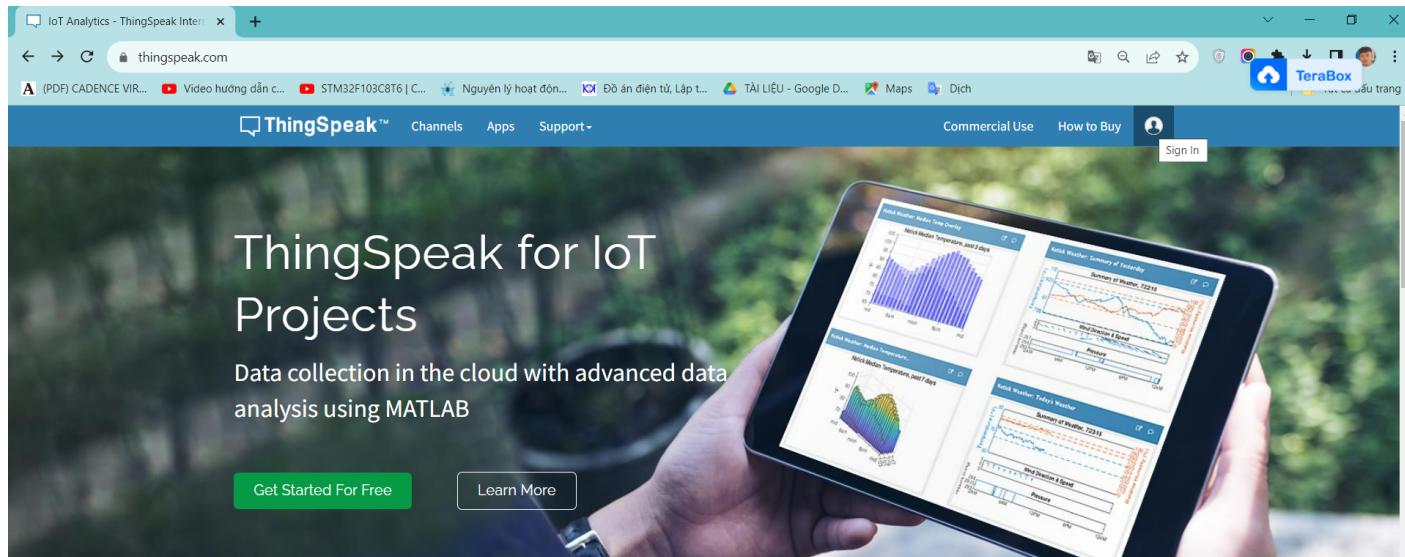
- + Các thiết bị có thể dễ dàng định cấu hình và gửi dữ liệu tới Thingspeak bằng cách sử dụng các giao thức truyền thông.
- + Có thể xem dữ liệu trong thời gian thực
- + Nó có thể nhận dữ liệu từ phần mềm của bên thứ ba.
- + Có thể sử dụng với Matlab để phân tích dữ liệu
- + Nó không cần server và phần mềm web để xây dựng prototype hệ thống IoT.
- + ~~Tự động thực hiện các hành động và giao tiếp bằng phần mềm của bên thứ ba.~~

4. Các bước thực hiện, giải thích code quá trình cập nhật dữ liệu lên server Thingspeak và kết quả thực hiện (video clip demo nếu có).

PHẦN 1: TẠO SERVER THINGSPEAK

Bước 1: Tạo tài khoản/ Đăng nhập Thingspeak

Đầu tiên, truy cập Website: www.thingspeak.com, sau đó chọn Sign Up để tạo một tài khoản ThingSpeak miễn phí.



Các bạn nhập địa chỉ Email, Tên đăng nhập (First Name và Last Name), Mật khẩu, sau đó chọn Continue.

ThingSpeak sẽ gửi cho bạn một Email xác nhận, bạn xác nhận xong lại chọn Sign In để đăng nhập.

Sign In - ThingSpeak IoT

← → 🔒 thingspeak.com/login?skipSSOCHECK=true

A (PDF) CADENCE VIR... Video hướng dẫn c... STM32F103C8T6 | C... Nguyên lý hoạt độn... Đồ án điện tử, Lập t... TÀI LIỆU - Google D... Maps Dịch

ThingSpeak™ Channels Apps Support Commercial Use How to Buy

To use ThingSpeak, you must sign in with your existing MathWorks account or create a new one.

Non-commercial users may use ThingSpeak for free. Free accounts offer limits on certain functionality. Commercial users are eligible for a time-limited free evaluation. To get full access to the MATLAB analysis features on ThingSpeak, log in to ThingSpeak using the email address associated with your university or organization.

To send data faster to ThingSpeak or to send more data from more devices, consider the [paid license options](#) for commercial, academic, home and student usage.

Create MathWorks Account

Email Address

To access your organization's MATLAB license, use your school or work email.

Location United States

First Name

Last Name

Continue Cancel

This site is protected by reCAPTCHA and the Google [Privacy Policy](#) and [Terms of Service](#) apply.

Bước 2: Tạo Channels ThingSpeak

Vào thẻ Channels, chọn New Channel để tạo một kênh mới.

My Channels - ThingSpeak IoT

← → 🔒 thingspeak.com/channels

A (PDF) CADENCE VIR... Video hướng dẫn c... STM32F103C8T6 | C... Nguyên lý hoạt độn... Đồ án điện tử, Lập t... TÀI LIỆU - Google D... Maps Dịch

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy

My Channels

New Channel Search by tag

Help

Collect data in a ThingSpeak channel from a device, from another channel, or from the web.

Click [New Channel](#) to create a new ThingSpeak channel.

Click on the column headers of the table to sort by the entries in that column or click on a tag to show channels with that tag.

Learn to [create channels](#), explore and transform data.

Learn more about [ThingSpeak Channels](#).

Examples

- Arduino
- Arduino MKR1000
- ESP8266
- Raspberry Pi
- Netduino Plus

Upgrade

Need to send more data faster?
Need to use ThingSpeak for a commercial project?

Upgrade

Blog | Documentation | Tutorials | Terms | Privacy Policy

https://thingspeak.com/channels#

← → 🔒 thingspeak.com/channels

A (PDF) CADENCE VIR... Video hướng dẫn c... STM32F103C8T6 | C... Nguyên lý hoạt độn... Đồ án điện tử, Lập t... TÀI LIỆU - Google D... Maps Dịch

ThingSpeak™ Channels Apps Devices Support Commercial Use How to Buy

f s r © 2023 The MathWorks, Inc.

Windows Type here to search 32°C 2:26 CH 09/11/2023

Nhập tên kênh vào ô Name, nhập mô tả (viết gì cũng được) vào ô Description, các ô Field 1, 2, 3,...8 chính là các trường dữ liệu mà các bạn sẽ gửi từ thiết bị lên Server này. Tôi gửi 2 trường dữ liệu là Temp, Humi, Moist tương ứng cho 3 giá trị dữ liệu: Nhiệt độ, độ ẩm không khí và độ ẩm đất.

Sau đó chọn dấu tích bên phải để cho phép trường đó hoạt động và chọn Save Channel để lưu lại.

New Channel

Name: tho

Description: ung dung iot

Field 1: Temp

Field 2: Humi

Field 3: Moist

Field 4:

Field 5:

Field 6:

Field 7:

Field 8:

Metadata:

Tags:

(Tags are comma separated)

Help

Channels store all the data that a ThingSpeak application collects. Each channel includes eight fields that can hold any type of data, plus three fields for location data and one for status data. Once you collect data in a channel, you can use ThingSpeak apps to analyze and visualize it.

Channel Settings

- Percentage complete: Calculated based on data entered into the various fields of a channel. Enter the name, description, location, URL, video, and tags to complete your channel.
- Channel Name: Enter a unique name for the ThingSpeak channel.
- Description: Enter a description of the ThingSpeak channel.
- Field#: Check the box to enable the field, and enter a field name. Each ThingSpeak channel can have up to 8 fields.
- Metadata: Enter information about channel data, including JSON, XML, or CSV data.
- Tags: Enter keywords that identify the channel. Separate tags with commas.
- Link to External Site: If you have a website that contains information about your ThingSpeak channel, specify the URL.
- Show Channel Location:
 - Latitude: Specify the latitude position in decimal degrees. For example, the latitude of the city of London is 51.5072.
 - Longitude: Specify the longitude position in decimal degrees. For example, the longitude of the city of London is -0.1275.
 - Elevation: Specify the elevation position meters. For example, the elevation of the city of London is 35.052.
- Video URL: If you have a YouTube™ or Vimeo® video that displays your channel information, specify the full path of the video URL.
- Link to GitHub: If you store your ThingSpeak code on GitHub®, specify the GitHub

Sau đó ThingSpeak sẽ hiện ra 3 biểu đồ, tuy nhiên sẽ chưa có giá trị nào do chưa gửi dữ liệu lên đây.

tho

Channel ID: 2339329
Author: mwa0000032035881
Access: Private

Private View Public View Channel Settings Sharing API Keys Data Import / Export

Add Visualizations Add Widgets Export recent data MATLAB Analysis MATLAB Visualization

Channel Stats

Created: about a minute ago
Entries: 0

Field 1 Chart: Temp

Field 2 Chart: Humi

Field 3 Chart: Moist

Bước 3: Lấy Channel ID, Write API Key và Read API keys

Vào thẻ API Keys, sẽ thấy các thông tin Channel ID, Write API Key, Read API Keys, đây chính là các thông tin quan trọng cho Data của bạn (nên bảo mật cũng như lưu lại, lát nữa chúng ta sẽ cần), các ID và Keys này sẽ giúp ThingSpeak cho phép thiết bị gửi dữ liệu lên đây.

The screenshot shows the 'API Keys - ThingSpeak IoT' page. At the top, it displays the URL 'thingspeak.com/channels/2339329/api_keys'. Below the header, there are tabs for 'Private View', 'Public View', 'Channel Settings', 'Sharing', 'API Keys' (which is selected), and 'Data Import / Export'. The main content area is divided into two sections: 'Write API Key' and 'Read API Keys'. Under 'Write API Key', there is a 'Key' input field containing a redacted value, a 'Generate New Write API Key' button, and a note field. Under 'Read API Keys', there is a 'Key' input field containing a redacted value, a 'Note' input field, a 'Save Note' button, a 'Delete API Key' button, and a 'Add New Read API Key' button. To the right of these sections, there is a 'Help' section with instructions for Write and Read API Keys, and a 'API Requests' section listing various API endpoints with their URLs.

PHẦN 2: LẮP ĐẶT THIẾT BỊ VÀ BUILD SOURCE CODE CHO ESP8266

Các thiết bị sử dụng

- Mạch WIFI ESP8266 CP2102 NodeMCU
- Cảm biến nhiệt độ, độ ẩm giao tiếp OneWire AM2301
- Cảm biến độ ẩm đất (Loại ADC).

Bước 1: Lắp đặt thiết bị

Sử dụng cảm biến nhiệt độ, độ ẩm AM2301 để đọc giá trị nhiệt độ, độ ẩm từ môi trường, cách kết nối như sau:

- Chân VVC cảm vào ngõ ra 3.3v của ESP8266
- Chân GND cảm vào ngõ ra GND của ESP8266
- Chân Data cảm vào các chân GPIO của board. Mình sử dụng chân D2.

Với cảm biến độ ẩm đất dạng ADC cấp điện áp dương 3.3V, âm tương tự, chân dữ liệu sử dụng chân ADC là A0 vì board chỉ có một ngõ vào ADC.

Bước 2: Build Source Code cho ESP8266 trên Arduino IDE:

```
#include <DHT.h>  
  
#include <ESP8266WiFi.h>  
  
#include <Wire.h>
```

```

#include "DHT.h"

#define DHTPIN D2
#define SOIL_MOIST_1_PIN A0
#define DHTTYPE DHT21 // DHT 21 (AM2301)
DHT dht(DHTPIN, DHTTYPE);

#define DHTTYPE DHT11 // DHT 11
#define DHTTYPE DHT22 // DHT 22 (AM2302), AM2321


float humi;
float temp;
float moist;

//WiFiSetting
const char* ssid      = "TT_H";
const char* password  = "trunghau0202";

WiFiClient client;

//ThingSpeakSetting
const int channelID   = 2339329// Enter Channel ID;
String writeAPIKey    = "FUPOK7VP4TMF9PRJ";
const char* server     = "api.thingspeak.com";

//FunctionDeclare
void wifiSetup();
void thingConnect();
void readSensor(void);
float getMoist();
void printData(void);

```

```

void setup() {
    Serial.begin(115200);
    Serial.println("Start Reading Sensor!\n");
    dht.begin();
    wifiSetup();
}

void loop() {
    thingConnect();
    readSensor();
    printData();
}

void wifiSetup() {
    Serial.print("Connecting");
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        Serial.print(".");
        delay(100);
    }
    Serial.print("\r\nWiFi connected");
    logoConsole();
}

void thingConnect() {
    if (client.connect(server, 80)) {
        String body = "field1=" + String(temp, 1) + "&field2=" + String(humi, 1)
        + "&field3=" + String(moist, 1);
        client.print("POST /update HTTP/1.1\n");
        client.print("Host: api.thingspeak.com\n");
        client.print("Connection: close\n");

```

```

        client.print("X-THINGSPEAKAPIKEY: " + writeAPIKey + "\n");

        client.print("Content-Type: application/x-www-form-urlencoded\n");

        client.print("Content-Length: ");

        client.print(body.length());

        client.print("\n\n");

        client.print(body);

        client.print("\n\n");

    }

}

void printData(void) {
    Serial.printf("Temp: %s°C - Hum: %s% - Moist: %s%\r\n", String(temp,
1).c_str(), String(humi, 1).c_str(), String(moist, 1).c_str());
}

void readSensor(void) {
    temp = dht.readTemperature();
    humi = dht.readHumidity();
    moist = getMoist();
}

float getMoist() {
    float i = 0;
    float anaValue = 0;
    for (i = 0; i < 10; i++)
    {
        anaValue += analogRead(SOIL_MOIST_1_PIN);
        delay(50);
    }
    anaValue = anaValue / (i);
    anaValue = map(anaValue, 1023, 0, 0, 99);
    return anaValue;
}

```

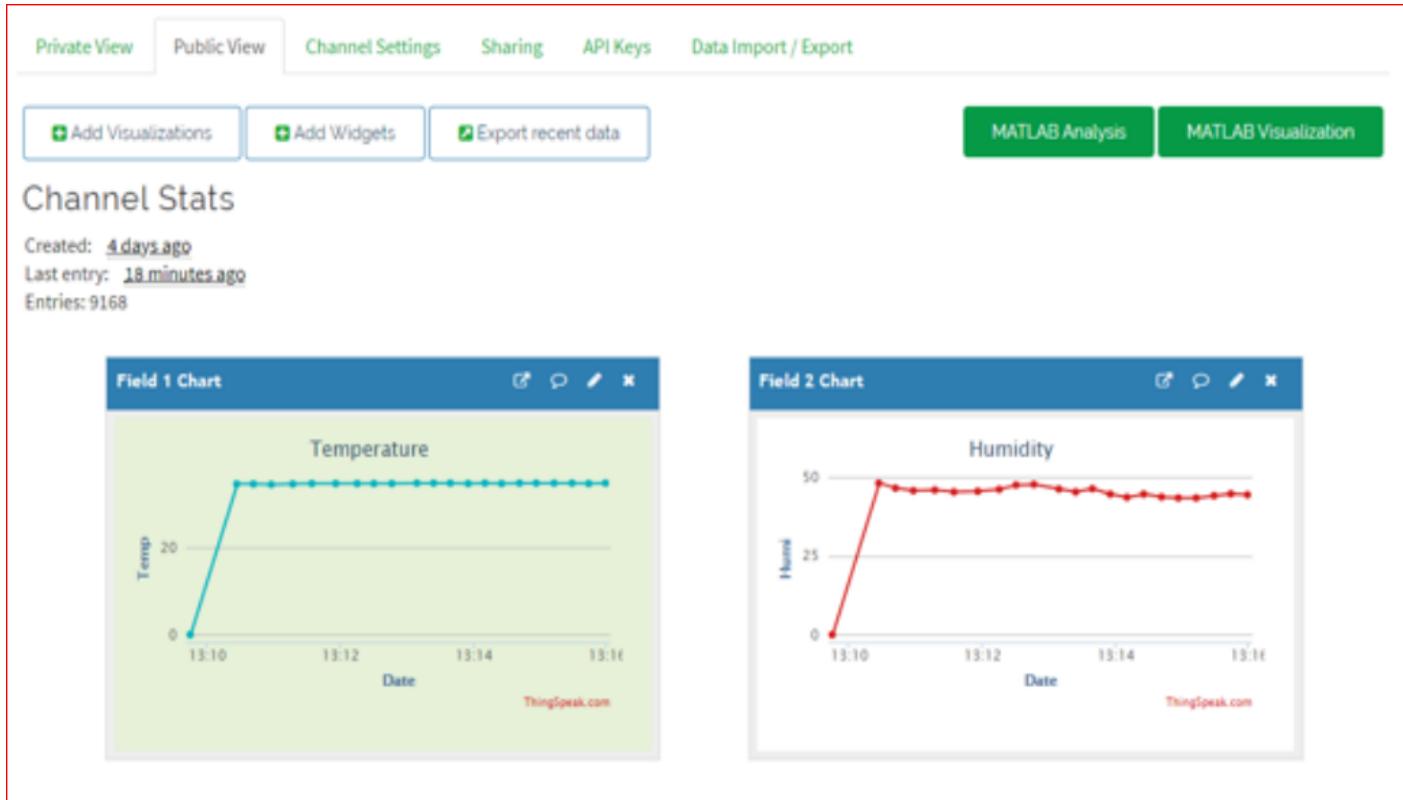
Các bạn nhập SSID (Tên WiFi) vào ô ssid, Nhập PASSWORD (Mật khẩu wifi) vào ô password.

Nhập channelID, writeAPIKey ở (Bước 3, Phần 1 đã lưu lại).

Sau đó build và nạp code cho board ESP8266.

Bước 3: Truy cập lại Channel ThingSpeak đã tạo:

Vào Public View các giá trị của cảm biến đã được gửi lên Server và có dạng biểu đồ như sau:



5. So sánh Thingspeak và Google Firebase

	Thingspeak	Google Firebase
Sự mô tả	_ Phần mềm cho phép người dùng giao tiếp với các thiết bị IoT.	_ Nền tảng phát triển ứng dụng toàn diện.
Mã nguồn	_ Có mã nguồn mở.	_ Không có mã nguồn mở, là công nghệ độc quyền của Google.
Đối tượng	_ Phù hợp với các phần cứng yêu cầu sự trực quan hóa và xử lý dữ liệu trực tuyến theo thời gian thực.	_ Phù hợp với các dự án ở mọi quy mô và phạm vi.
Tính năng	<ul style="list-style-type: none">_ Thu thập dữ liệu một cách riêng tư theo yêu cầu từ các nguồn của bên thứ ba._ Sử dụng các công cụ phân tích trực tuyến do MATLAB._ Cung cấp tiện ích để kích hoạt giao tiếp thiết bị và tự động kích hoạt khi có phản hồi giữa các thiết bị với nhau.	<ul style="list-style-type: none">_ Tích hợp dịch vụ phụ trợ._ Tích hợp dữ liệu phân tích của Google._ Có bảng điều khiển để theo dõi dữ liệu và khắc phục sự cố.

Ưu điểm	<ul style="list-style-type: none"> – Dễ dàng thiết lập thiết bị để gửi dữ liệu. – Đọc dữ liệu cảm biến trong thời gian thực. – Thực hiện quét IoT của bạn tự động dựa trên lịch hoặc sự kiện. – Phát triển hệ thống IoT mà không cần thiết lập máy chủ hoặc phát triển phần mềm web. – Truyền đạt dữ liệu bằng các dịch vụ của bên thứ ba và các nền tảng mạng xã hội. 	<ul style="list-style-type: none"> – Cơ sở dữ liệu thời gian thực. Hỗ trợ các hoạt động ngoại tuyến. – Tính bảo mật cao có đám mây để lưu trữ và đồng bộ dữ liệu.
Nhược điểm	<ul style="list-style-type: none"> – Nếu lỗi sẽ lỗi toàn hệ thống. – Giao diện khó cho người mới bắt đầu. 	<ul style="list-style-type: none"> – Thiếu hỗ trợ cập nhật chương trình cơ sở OTA. – Khó quản lý dữ liệu.

– Thingspeak tập trung chủ yếu vào việc thu thập và phân tích dữ liệu từ các thiết bị IoT, trong khi Firebase cung cấp một loạt các dịch vụ phát triển ứng dụng di động và web.

– Firebase có một loạt các dịch vụ rộng hơn, bao gồm cả lưu trữ dữ liệu và các tính năng mở rộng khác như thông báo đẩy và phân tích dữ liệu.

– Thingspeak có tích hợp sẵn các công cụ và tính năng phân tích dữ liệu IoT, trong khi Firebase tập trung vào việc cung cấp một nền tảng phát triển ứng dụng di động và web toàn diện.

– Thingspeak cung cấp miễn phí sử dụng cho các ứng dụng không thương mại, trong khi Firebase có mô hình giá cả linh hoạt dựa trên việc sử dụng tài nguyên.

6. So sánh Amazon AWS IoT và Microsoft Azure IoT hub (Optional)

AWS	Azure
<ul style="list-style-type: none"> • AWS IoT là một dịch vụ điện toán đám mây mạnh mẽ của Amazon Web Services (AWS) dành cho IoT. • Cung cấp các tính năng như đăng ký và quản lý thiết bị, gửi và nhận dữ liệu, xác thực và quản lý người dùng, quyền truy cập và phân quyền. • Hỗ trợ giao thức MQTT và HTTP cho việc giao tiếp giữa thiết bị IoT và nền tảng. • Có tích hợp sẵn với các dịch vụ AWS khác như S3 (Simple Storage Service), DynamoDB, Lambda, và Amazon Kinesis. • Cung cấp tính năng phân tích dữ liệu và trực quan hóa thông qua AWS IoT Analytics và AWS IoT Core Rules. • Hỗ trợ tính năng bảo mật mạnh mẽ bao gồm xác thực và mã hóa dữ liệu. • Lưu trữ EBS rất nhanh chóng đối với dữ liệu lớn. • AWS cung cấp giải pháp cloud với hạ tầng hoàn thiện và trải rộng trên nhiều lĩnh vực. • Trả tiền theo giờ. • Các vai trò của user được xác định bằng các quyền kiểm soát duy nhất được sử dụng để 	<p>Azure IoT Hub là một dịch vụ của Microsoft Azure dành cho quản lý và kết nối các thiết bị IoT.</p> <p>Cung cấp các tính năng như đăng ký và quản lý thiết bị, gửi và nhận dữ liệu, xác thực và quản lý người dùng, quyền truy cập và phân quyền.</p> <p>Hỗ trợ giao thức MQTT, AMQP và HTTP cho việc giao tiếp giữa thiết bị IoT và nền tảng.</p> <p>Có tích hợp sẵn với các dịch vụ Azure khác như Azure Functions, Azure Stream Analytics, và Azure Machine Learning.</p> <p>Cung cấp tính năng phân tích dữ liệu và trực quan hóa thông qua Azure IoT Analytics và Power BI.</p> <p>Hỗ trợ tính năng bảo mật mạnh mẽ bao gồm xác thực, mã hóa dữ liệu và quản lý khóa.</p> <p>Khó khăn trong việc lưu trữ dữ liệu lớn nhưng đang được cải thiện và phát triển.</p> <p>Azure cũng mạnh cloud mà còn tích hợp thêm nhiều thứ khác trong hệ sinh thái của Microsoft</p> <p>Trả tiền theo phút.</p> <p>Azure Active Directory là một nguồn duy nhất để quản lý quyền và ủy quyền. Ở AWS người dùng, liên kết và quyền truy cập vào từng tài khoản phải được định cấu hình,</p>

cung cấp bảo mật, cung cấp việc triển khai các nhóm bảo mật và IAM chi tiết.

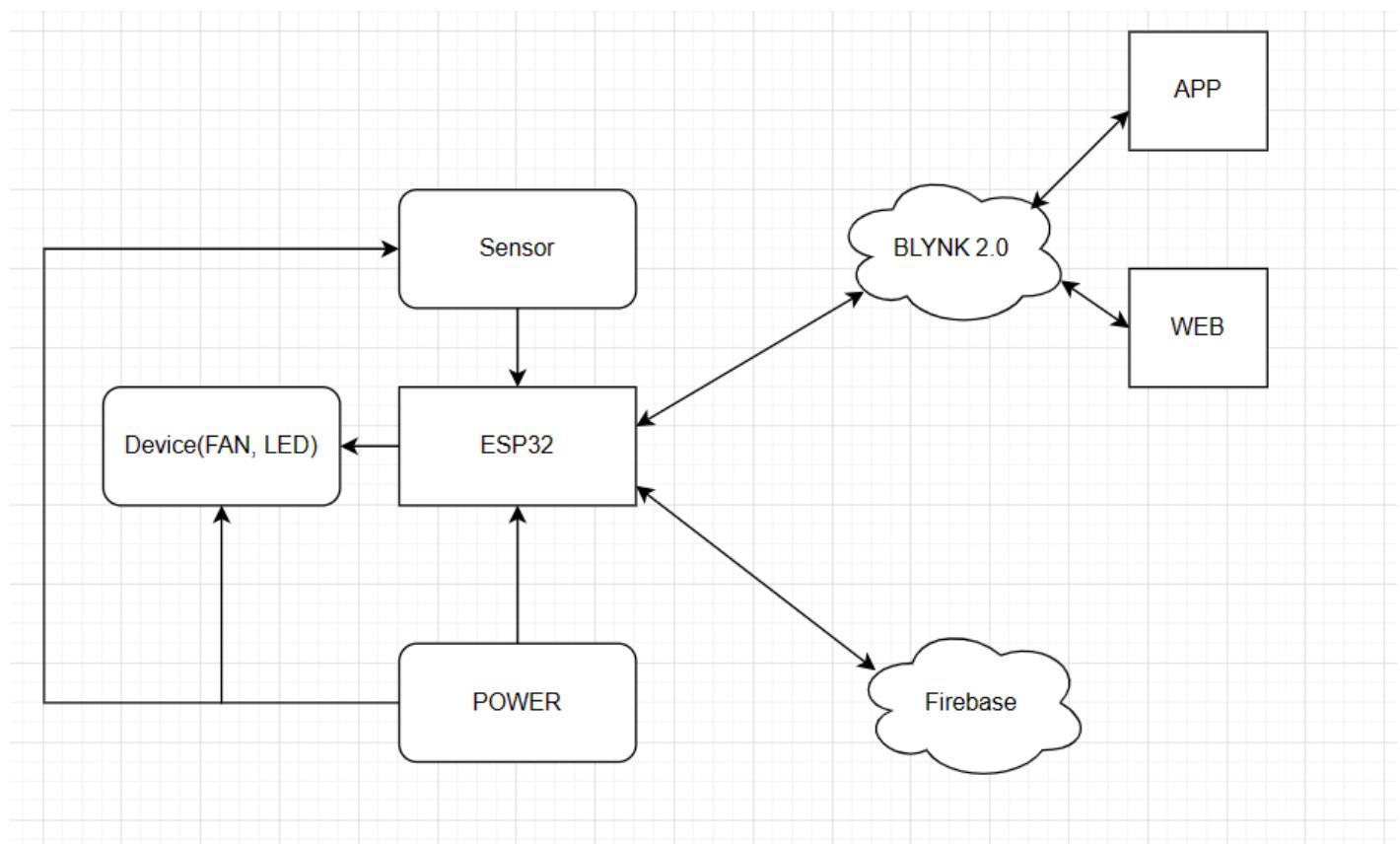
Azure cho phép bạn thực hiện việc này từ một thư mục duy nhất.

7. Thiết kế và thực hiện 01 hệ thống giám sát và điều khiển thiết bị dựa trên các cảm biến và phần cứng ESP32, kết hợp Thingspeak hoặc Firebase. (các bước thực hiện đầy đủ như đã hướng dẫn trong Lab 2)

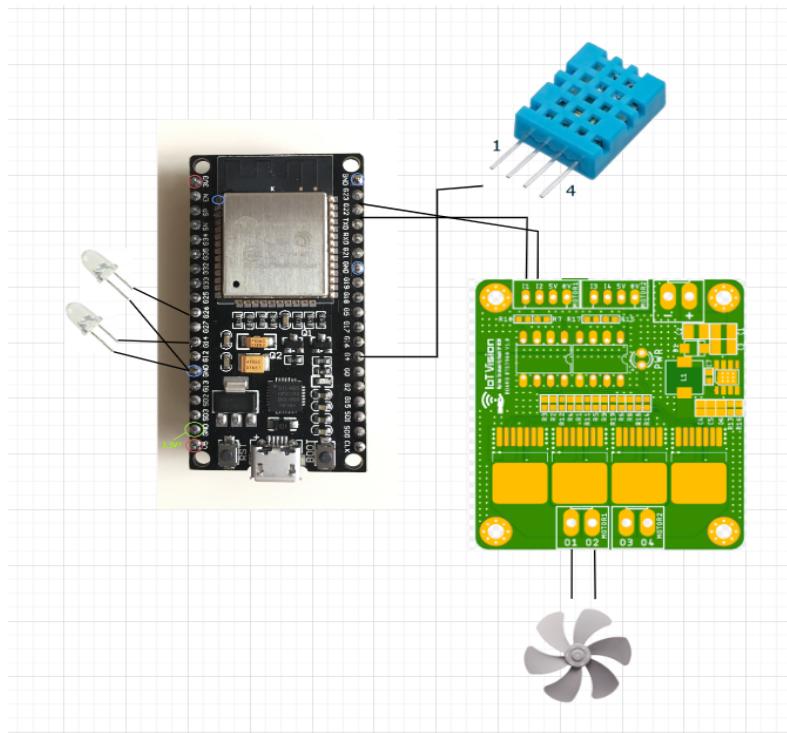
Yêu cầu thiết kế:

Sử dụng cảm biến nhiệt độ, độ ẩm để đo nhiệt độ và độ ẩm kết hợp điều khiển tốc độ quạt

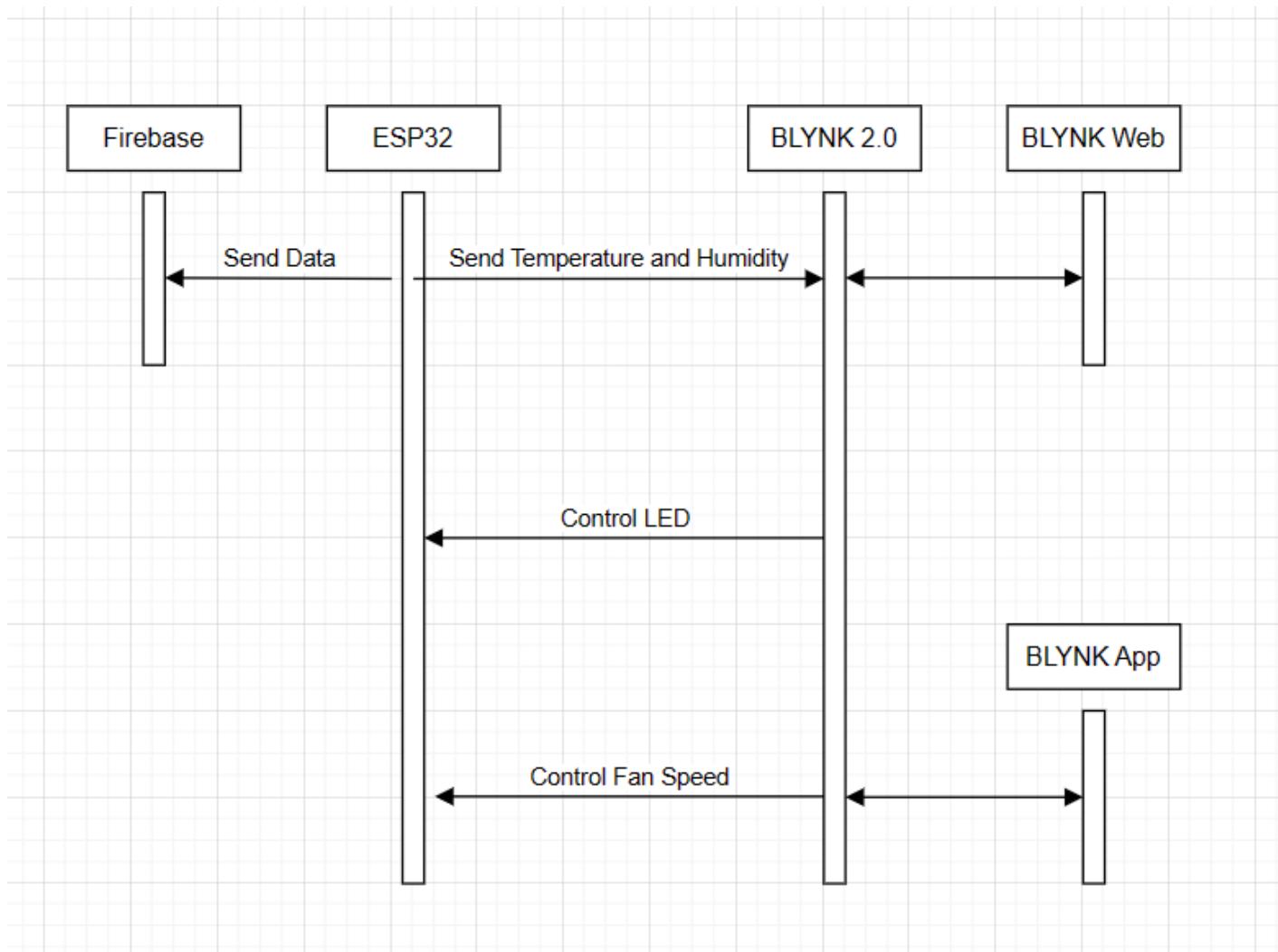
Sơ đồ khái:



Sơ đồ kết nối:

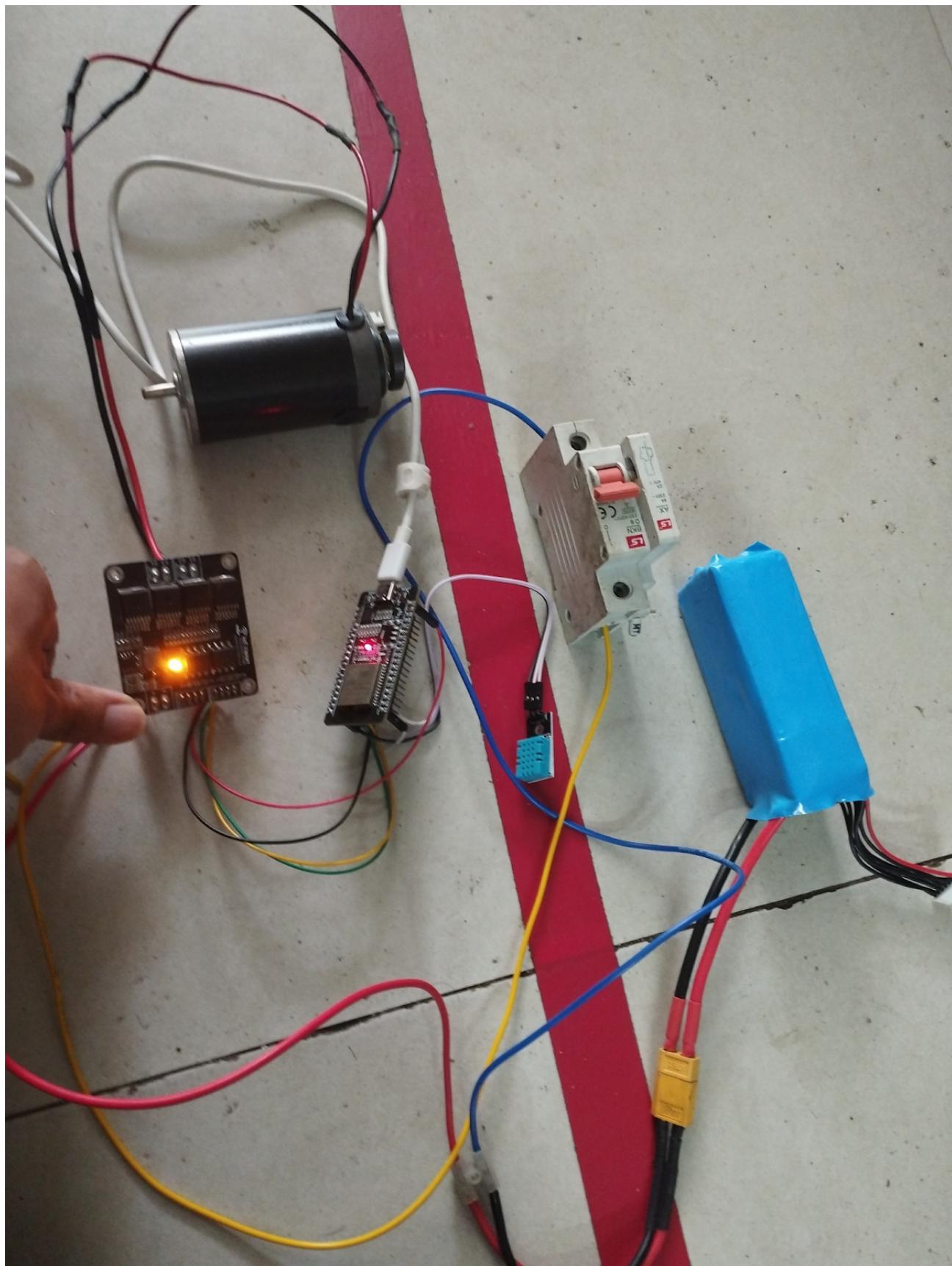


Lưu đồ giải thuật:

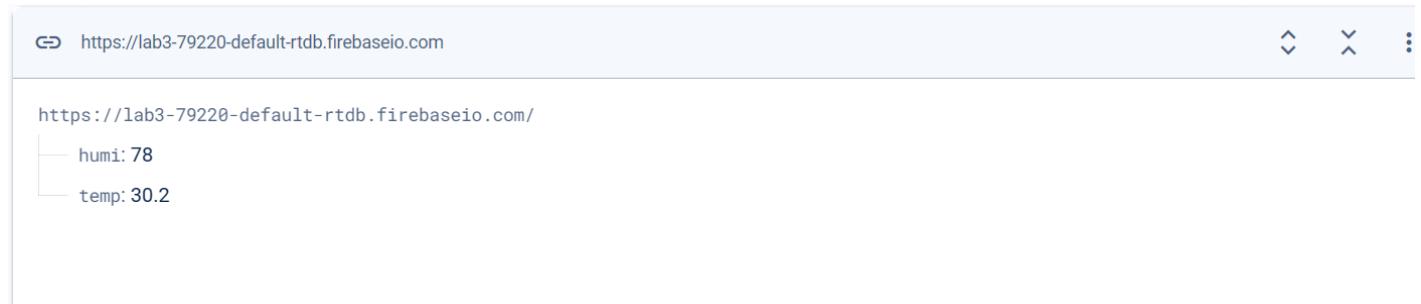


Kết quả:

Hardware:



Firebase:



A screenshot of a web browser displaying the Firebase Realtime Database at the URL <https://lab3-79220-default.firebaseio.com/>. The database structure shows two child nodes: "humi" with a value of 78 and "temp" with a value of 30.2.

```
https://lab3-79220-default.firebaseio.com/
  humi: 78
  temp: 30.2
```

Tài liệu tham khảo (liệt kê theo đúng chuẩn IEEE)

Hình ảnh làm việc nhóm

