**HỌC VIỆN CÔNG NGHỆ BƯU CHÍNH VIỄN THÔNG**

**TRUNG TÂM THÍ NGHIỆM THỰC HÀNH**

**BÀI THÍ NGHIỆM MÔN HỌC IoT VÀ ỨNG DỤNG**

*(Tài liệu dành cho sinh viên nghành CNTT)*

**Họ tên sinh viên:**....................................

**Mã sinh viên:**…………………………..

**Lớp:**…………………………………….

**BÀI THÍ NGHIỆM SỐ 2**

**GIAO THỨC HTTP VÀ MQTT TRONG IoT**

1. **MỤC ĐÍCH, YÊU CẦU**
   1. **Mục đích:**

* Giúp sinh viên hiểu rõ hơn về các giao thức truyền thông phổ biến trong IoT như giao thức MQTT, HTTP, WebSocket,.. và cách thức các thiết bị trong hệ thống IoT sử dụng những giao thức này để truyền thông tin.
* Nắm được các mô hình trao đổi thông tin trong IoT như mô hình Request-Response, mô hình Publish- Subscribe, mô hình WebSocket API,..
* Nắm được cách thức lập trình cho các thiết bị nhúng sử dụng các giao thức truyền thông với những mô hình trao đổi thông tin được sử dụng phổ biến trong việc thu nhận dữ liệu thô của các cảm biến dùng ngôn ngữ lập trình C trên môi trường phát triển tích hợp IDE Arduino 2.x
  1. **Yêu cầu:**
* Sinh viên đã được học lý thuyết Chương 1 và 2 và 3 của môn học *IoT và ứng dụng*.
* Sinh viên đã có tài liệu thí nghiệm cho môn học *IoT và ứng dụng.*
* Sinh viên đã được học về ngôn ngữ lập trình C/C++. Kỹ thuật vi xử lý và kiến trúc máy tính.

1. **CHUẨN BỊ**

- Sinh viên đọc trước ở nhà nội dung bài thí nghiệm.

- Xem lại nội dung lý thuyết Chương 1, 2, 3 môn học *IoT và ứng dụng*.

- Xem lại lý thuyết của bài thí nghiệm số 1

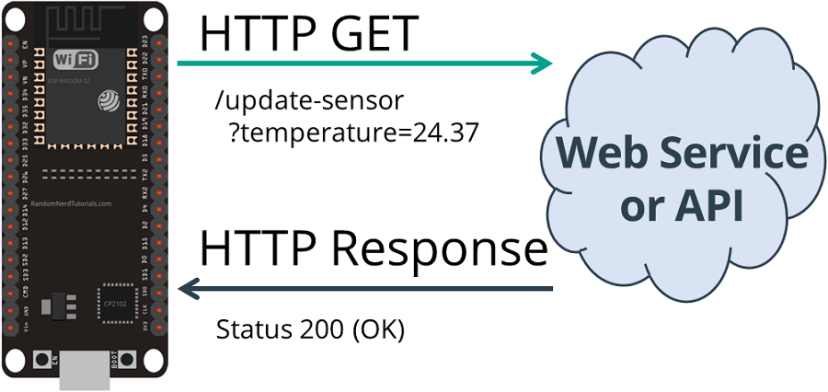
**III. THÍ NGHIỆM**

**3.1 GIAO THỨC HTTP**

**3.1.1. Phương tức GET**

Phương thức **GET** là phương thức phổ biến nhất để lấy dữ liệu về từ server khi sử dụng giao thức HTTP.

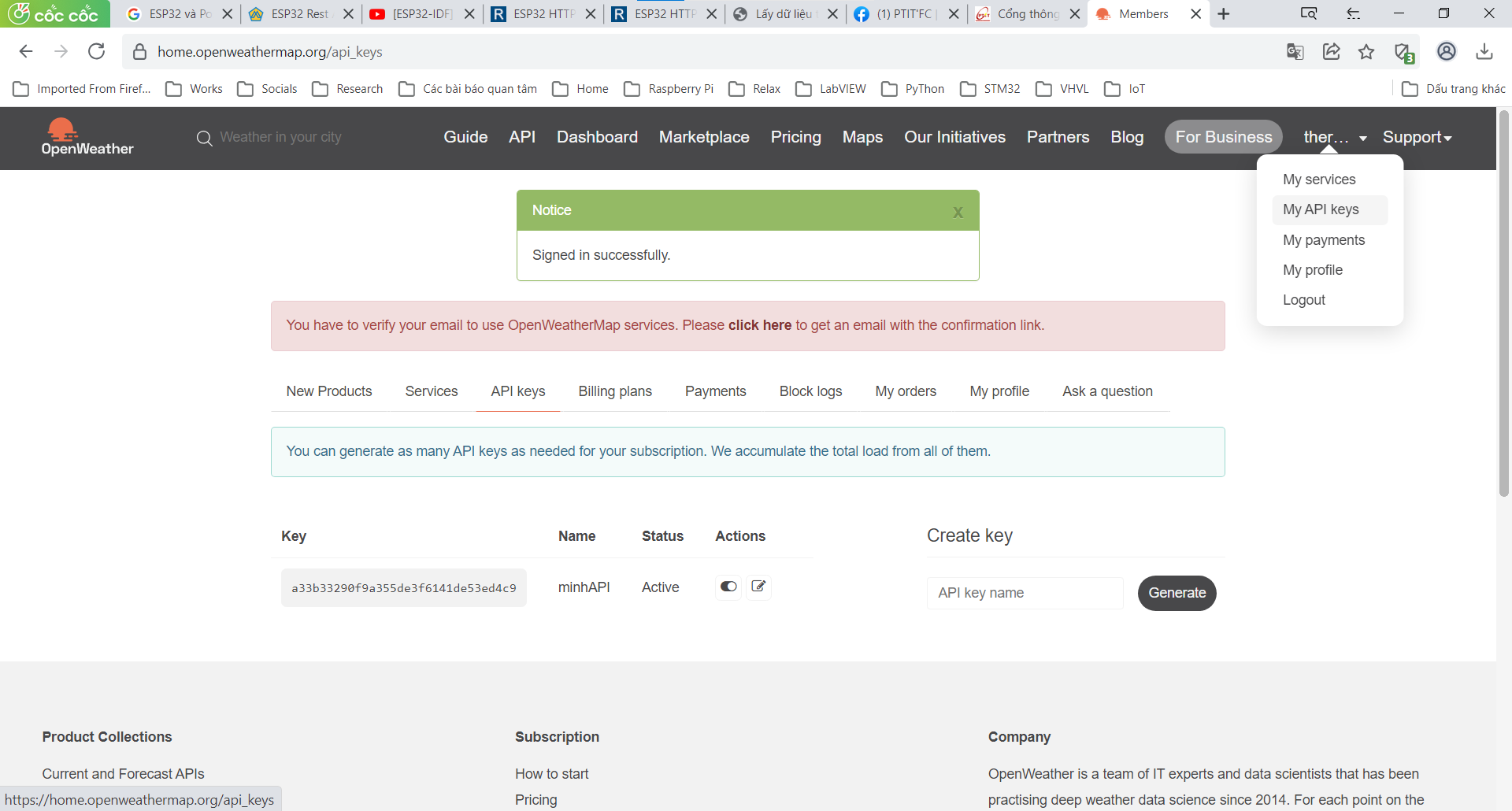
Mô hình hệ thống được cho như sau:



*Hình vẽ 2.1. Mô hình hệ thống thu nhận dữ liệu từ Web Service với phương thức* ***GET***

Hãy viết chương trình sử dụng ESP32 trên KIT thí nghiệm IoT như một HTTP Client để lấy dữ liệu thời tiết từ trang web <http://openweathermap.org> hiển thị trên terminal của máy tính.

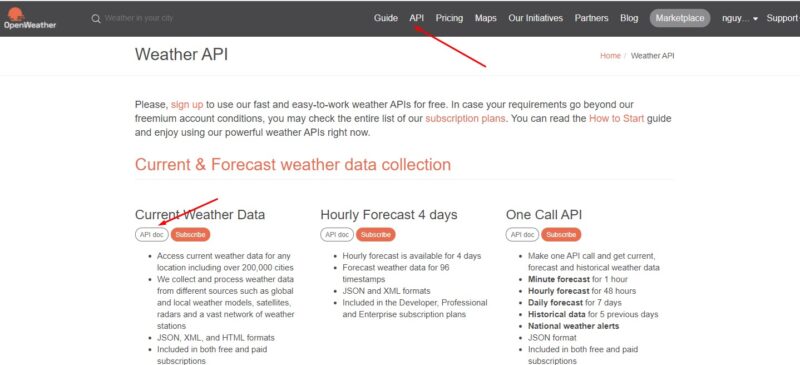
**Bước 1**: Để có thể lấy được dữ liệu từ trang Web openweathermap.org cần phải đăng ký một tài khoản miễn phí trên trang web này và sau đó đăng nhập.



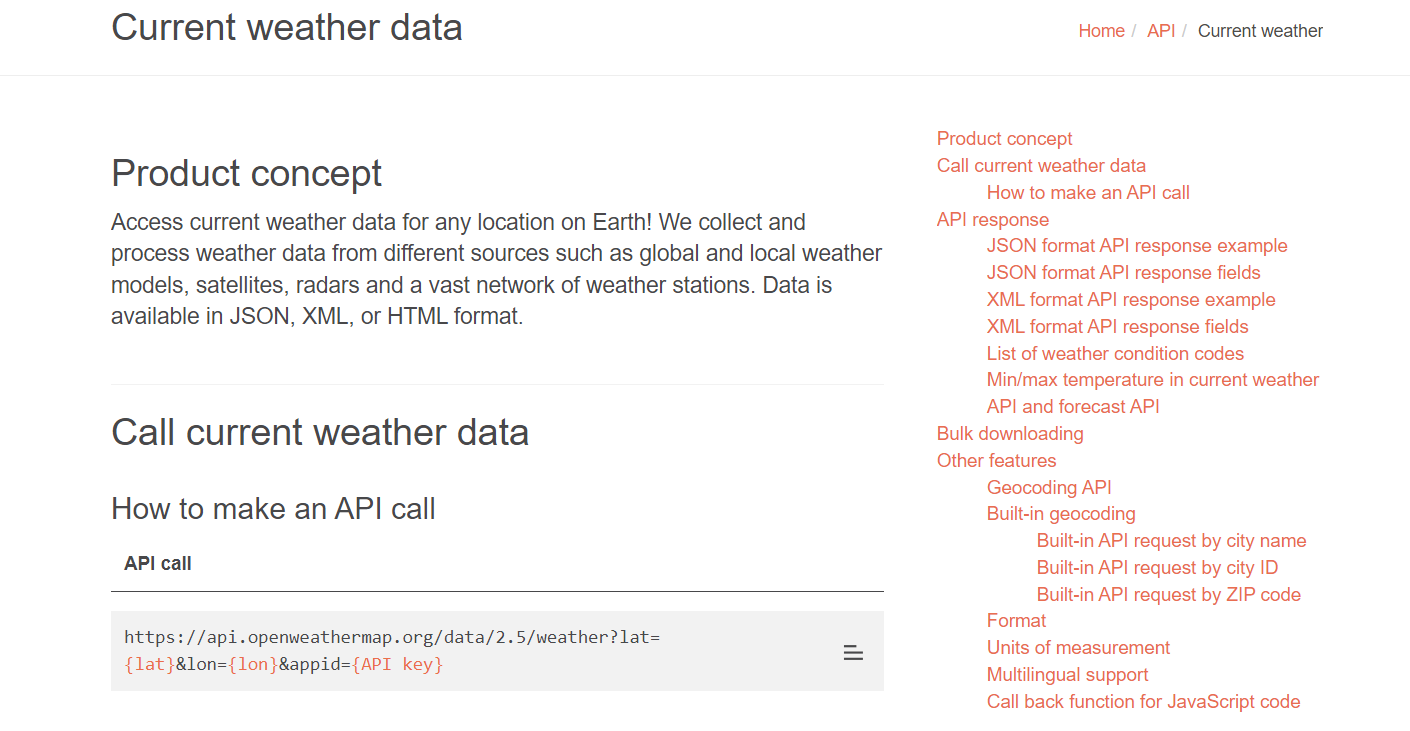
*Hình vẽ 2.2. Đăng nhập và lấy API key*

**Bước 2**: Trong tab User, chọn My API keys, đặt tên API và tạo ra (**Generate**)  một API mới. API key chính là một chìa khóa để định danh các người dùng sử dụng API giúp server quản lý việc user sử dụng tài nguyên nào, không được sử dụng tài nguyên nào.

* Sau khi có được API để có thể truy cập vào dữ liệu của Openweathermap, nhấn vào API key – Current weather data – API doc để xem hướng dẫn sử dụng API (*chú ý: do đăng ký tài khoản miễn phí nên chỉ có thể Call API 60 lần/phút*).



*Hình vẽ 2.3. Đọc hướng dẫn sử dụng API*

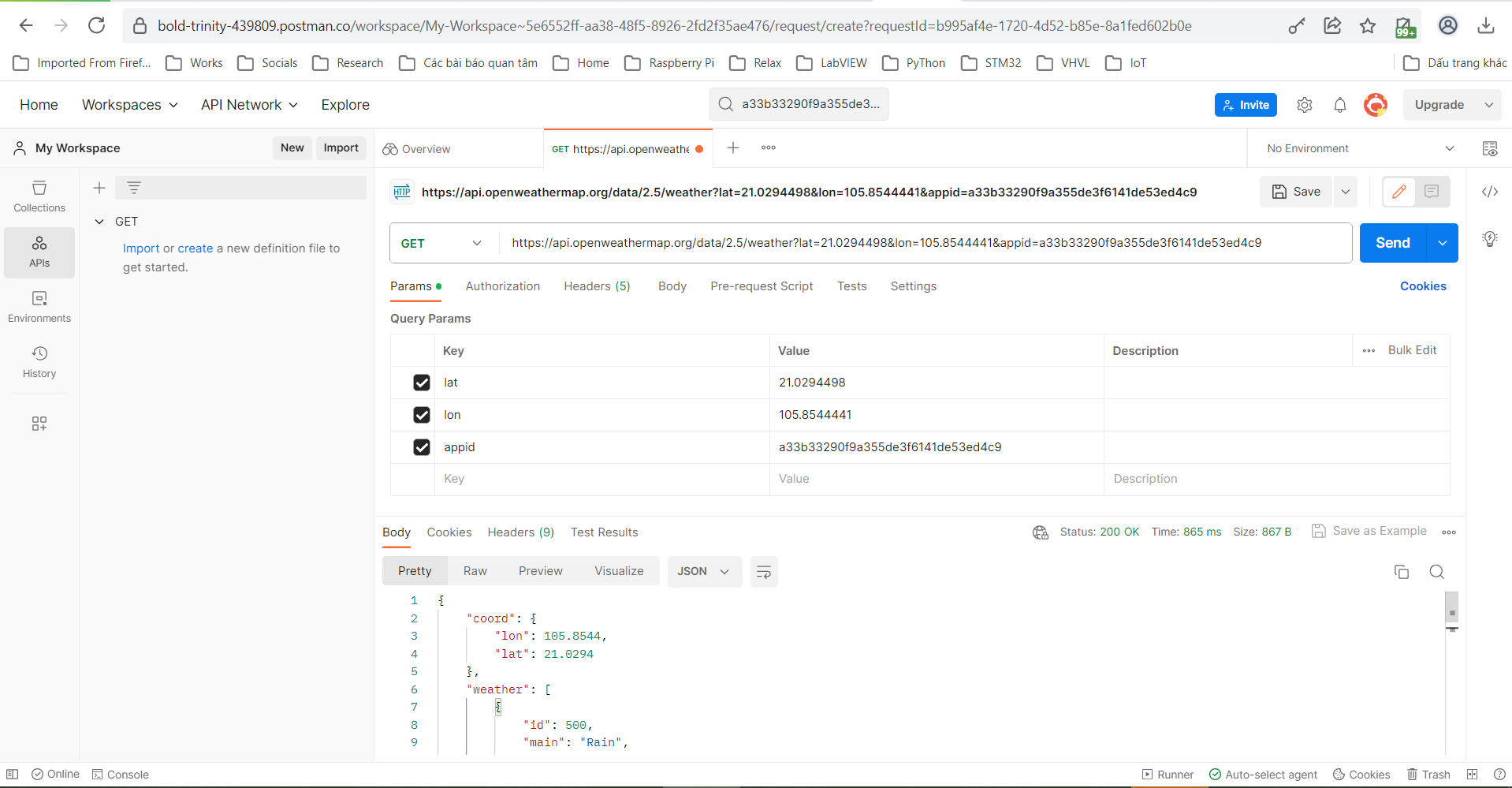


*Hình vẽ 2.4. Các cách dùng API*

* Có thể sử dụng API theo nhiều cách, có thể là
  + Sử dụng tên thành phố (City name)
  + Sử dụng Zip code
  + Sử dụng kinh độ và vĩ độ (longitude, latitude)
* Để sử dụng được phương thức **GET** cần có các thông số sau:
  + ***URL***: Địa chỉ server mà chúng ta trỏ tới, nó có thể là IP và port hoặc domain tương ứng.
  + Phương thức: Chọn phương thức giao tiếp là **GET**
  + ***Query***: sau kí tự ?q= thường sẽ truyền một tham số mà server yêu cầu. Ví dụ như App key, thành phố, id….
  + ***Header***: Truyền vào kiểu dữ liệu trả về, kiểu kết nối, hoặc các dữ liệu liên quan khác.
  + ***Body***: Với phương thức **GET** thường sẽ không truyền Body gì cả.
* Trong bài thí nghiệm này, để lấy được dữ liệu về thời tiết chúng ta cần các thông số sau:
  + Url: api.openweathermap.org/data/2.5/weather
  + query: ?q={city name}&appid={api key mà bạn vừa tạo ra}
  + header: mặc định kiểu dữ liệu trả về là JSON
  + Phương thức: **GET**

**Bước 3**: Sử dụng trang web <https://www.postman.com/> để kiểm tra việc sử dụng API có thành công hay không.

* Hãy giải thích ý nghĩa của các mã phản hồi sau:
* Mã 200………………………………………………………………………………
* Mã 201………………………………………………………………………………
* Mã 202………………………………………………………………………………
* Mã 400………………………………………………………………………………
* Mã 404………………………………………………………………………………
* Mã 500………………………………………………………………………………
* Mã 405………………………………………………………………………………
* Truyền các tham số và URL vừa định nghĩa bên trên, nhấn Send. Kết quả ta sẽ nhận được một chuỗi JSON như sau:



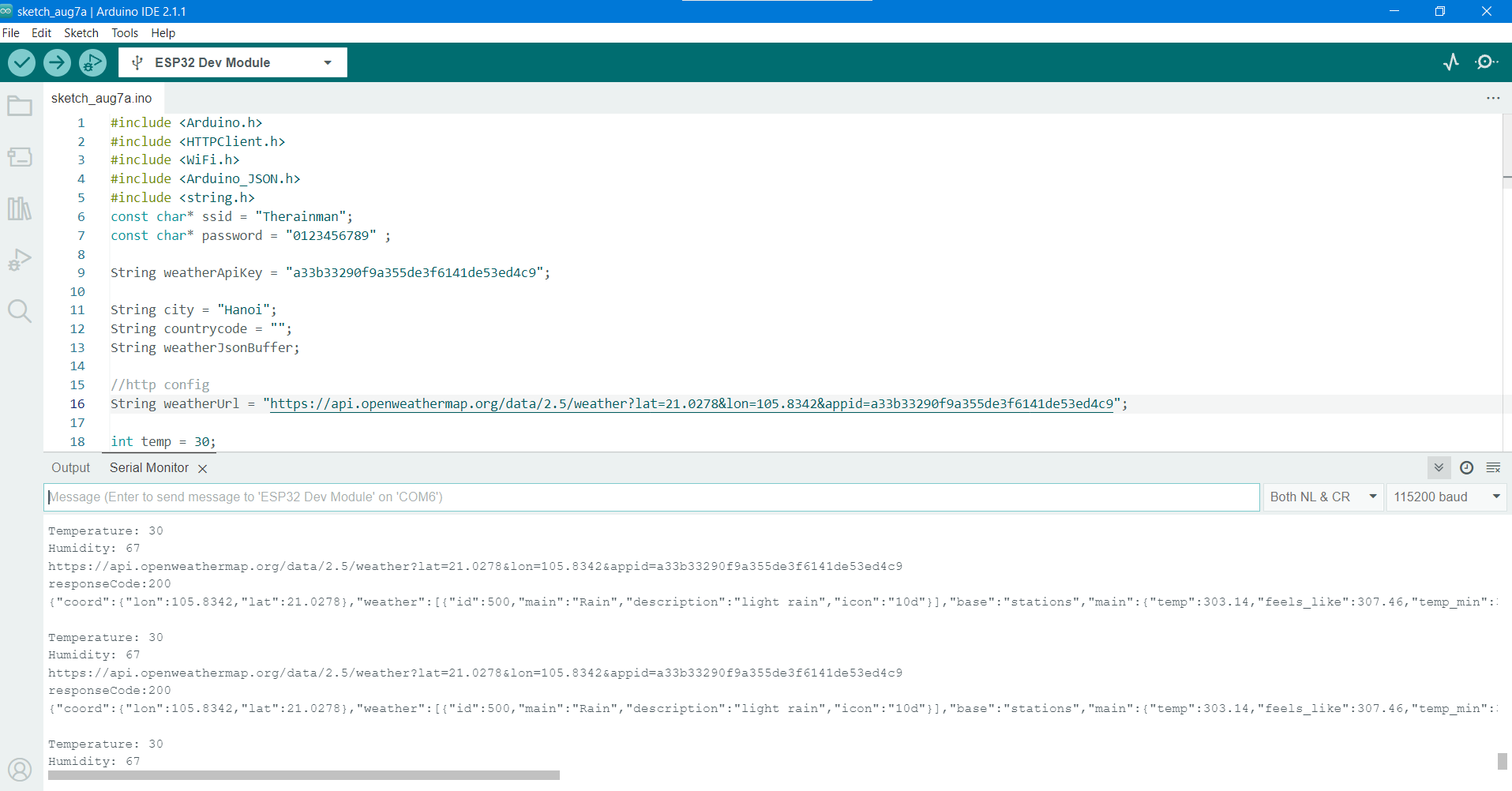
*Hình vẽ 2.5. Kết quả nhận được khi test API trên POSTMAN*

*Chú ý: Một API gọi thành công sẽ có mã phản hồi là 200 trên POSTMAN*

**Bước 4**: Lập trình cho ESP32

Để lập trình cho ESP32 với phương thức **GET** của giao thức HTTP cần sử dụng các thư viện bổ sung cho Arduino IDE sau: *HTTPClient.h*; *Arduino\_JSON.h*; *Wifi.h; String.h*.

* Mở chương trình Arduino IDE và nhập viết mã cho chương trình
* Biên dịch và nạp mã cho ESP32 (*Chú ý tên mạng WiFi đang dùng và password*)
* Kết quả khi chương trình chạy tốt sẽ có dạng như sau



*Hình vẽ 2.6. Kết quả khi chạy chương trình lấy dữ liệu nhiệt độ, độ ẩm trên trang https:\\www.openweather.org*

**Bước 5**:Trả lời các câu hỏi

* Mạng Wifi đang dùng có tên gì ?.............................................................................
* Pass word mạng Wifi đang dùng cho ESP32 là gì:………………………………..
* Kinh độ và vĩ độ của thành phố Hà Nội:…………………………………………….
* Viết lại API có được trên trang *https:\\www.openweather.org* của riêng nhóm thí nghiệm xuống dưới đây:

………………………………………..……………………………………………..

* Có mấy cách gọi API trên trang *https:\\www.openweather.org* để có thể lấy được dữ liệu thời tiết ?..........................................................................................................
* Hãy sử dụng các cách gọi API khác nhau để lấy dữ liệu thời tiết của thành phố Hà Nội (sửa trực tiếp trong mã lệnh chương trình của ESP32).
* Viết lại các chuỗi String với các cách gọi API khác nhau xuống dưới đây ?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………- Ngoài dữ liệu về nhiệt độ và độ ẩm, còn có thể lấy được thông tin gì về thời tiết trên trang *https:\\www.openweather.org* nữa không ?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

- Hãy sửa lại chương trình để đọc được các thông tin về dự báo Mưa, Gió, in ra trên màn hình serial monitor của máy tính.

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

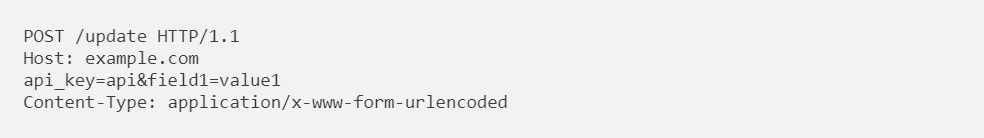
………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

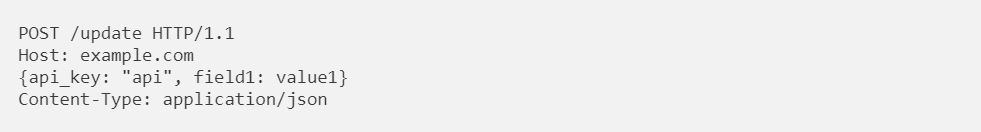
………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**3. 1. 2 Phương thức POST**

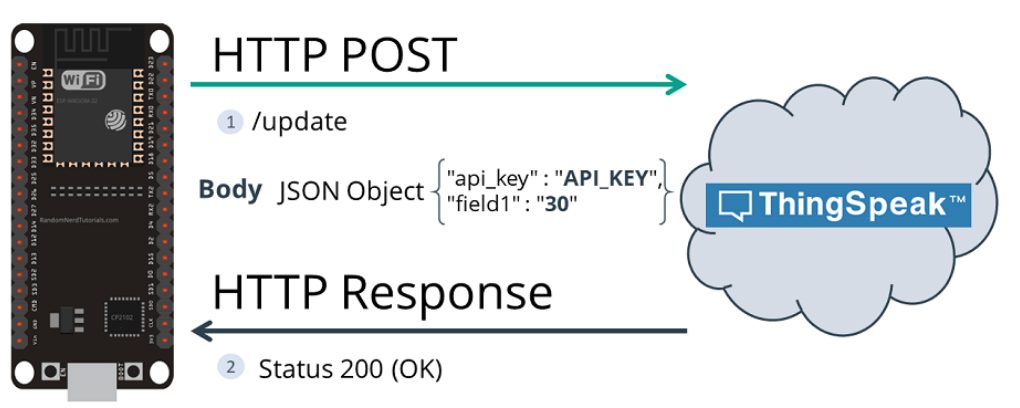
Phương thức POST được sử dụng trong giao thức HTTP để gửi dữ liệu đến máy chủ, nhằm tạo hoặc cập nhật tài nguyên. Ví dụ: POST các dữ liệu thu được từ cảm biến nhiệt độ, độ ẩm lên máy chủ. Dữ liệu được gửi lên máy chủ bằng POST được chứa trong phần thân của yêu cầu HTTP REQUEST. Ví dụ:



Trong phần thân của REQUEST cũng có thể gửi một đối tượng JSON



Trong phần này sinh viên sẽ lập trình cho ESP32 để tạo một HTTP POST nhằm gửi một giá trị mới tới trang ThingSpeak.com. Mô hình hệ thống được cho như sau:

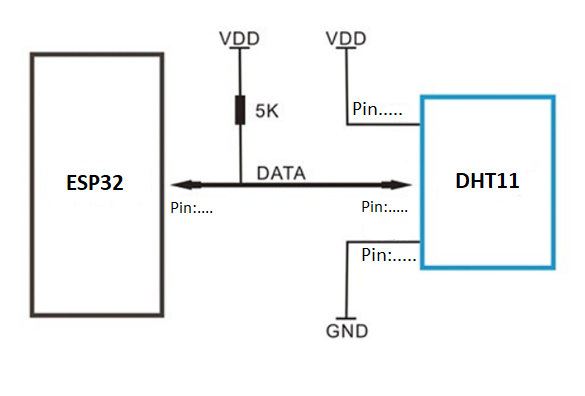


*HÌnh vẽ 2.7. Mô hình hệ thống cho HTTP POST*

Các bước thực hiện :

**Bước 1**:

* Xác định khối mạch cảm biến DHT11 và các chân cắm tương ứng trên KIT thí nghiệm IoT.
* Xác định các chân cắm tương ứng của KIT Arduino ESP32 được sử dụng để nối với cảm biến DHT11.
* Cắm dây nhảy để kết nối giữa ESP32 và DHT11. Kết nối KIT thí nghiệm IoT với máy tính qua cổng USB. Bật nguồn điện KIT thí nghiệm IoT. Điền đầy đủ dữ liệu vào *Hình vẽ 2.8* dưới đây.



*Hình vẽ 2.8. Kết nối ESP32 và DHT11*

**Bước 2**: Viết chương trình cho ESP32

* Viết chương trình cho KIT Arduino ESP32 bằng phần mềm Arduino IDE 2.x.x trên máy tính với mục đích thu thập dữ liệu từ cảm biến DHT11 và hiển thị trên cửa sổ terminal của chương trình Arduino IDE.
* Thực hiện chương trình và quan sát kết quả dữ liệu Nhiệt độ, Độ ẩm thay đổi theo thời gian. Ghi lại kết quả vào bảng sau:

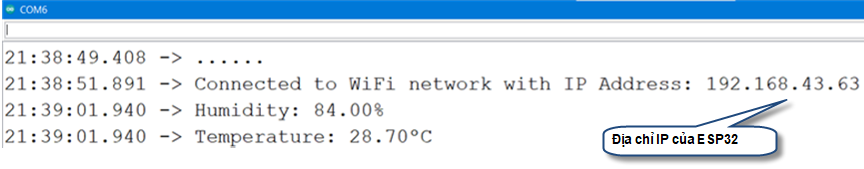
|  |  |  |  |
| --- | --- | --- | --- |
| **Thứ tự phép đo** | **Giá trị Nhiệt độ (ºC)** | **Giá trị Độ ẩm (%)** | **Ghi chú** |
| 1 |  |  |  |
| 2 |  |  |  |
| 3 |  |  |  |
| 4 |  |  |  |
| 5 |  |  |  |
| 6 |  |  |  |
| 7 |  |  |  |
| 8 |  |  |  |
| 9 |  |  |  |
| 10 |  |  |  |

**Bảng 2.1***. Một số giá trị thông số nhiệt độ, độ ẩm môi trường*

**Bước 3**:

* Sửa lại chương trình đọc dữ liệu nhiệt độ, độ ẩm để có thêm chức năng kết nối Internet thông qua WiFi.
* Nạp chương trình mới cho ESP32 và cho hệ thống hoạt động. Quan sát cửa sổ terminal của Arduino IDE. Ghi lại địa chỉ IP của ESP32 trong mạng WiFi của mình vào đây:……………………………………………………………………………….

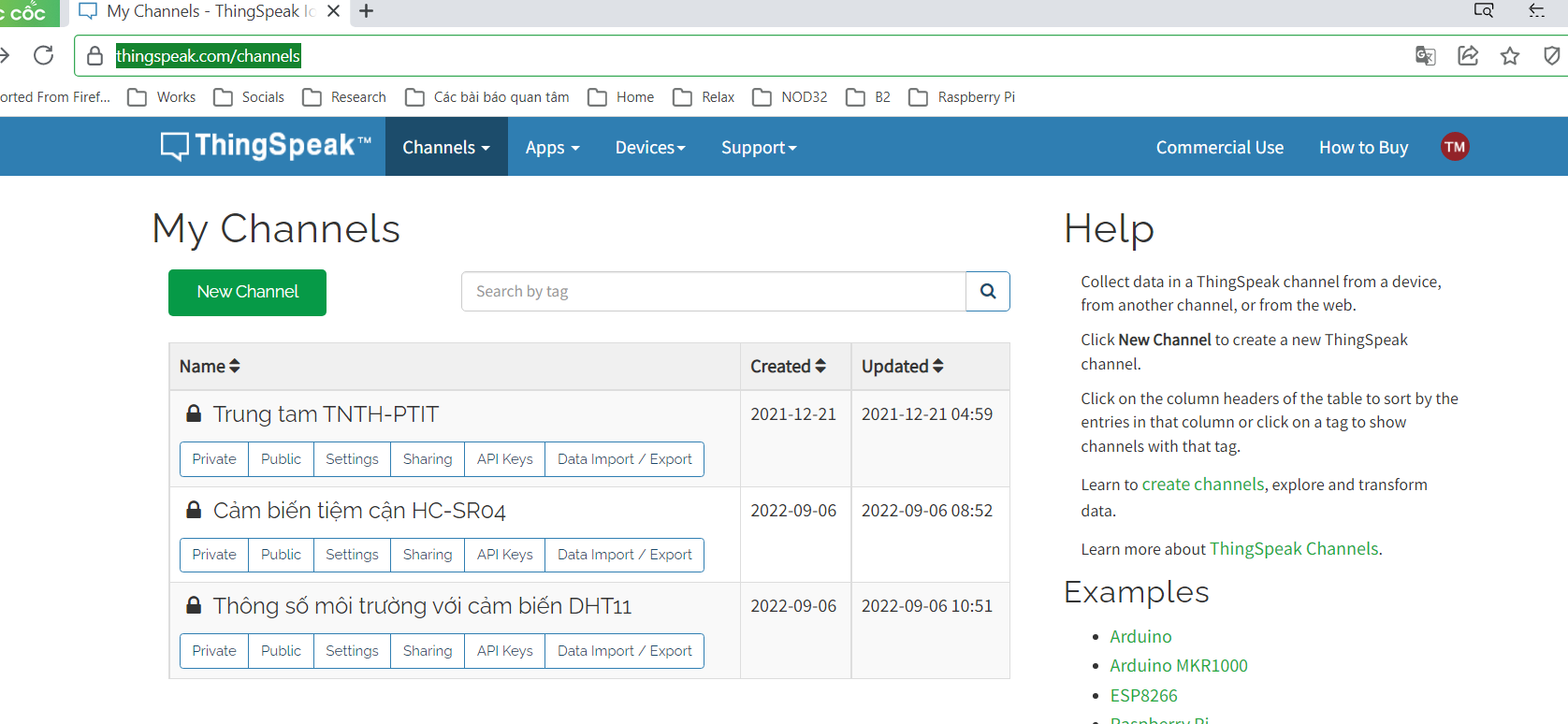
Dữ liệu trên cửa sổ terminal của Arduino IDE có dạng như *Hình vẽ 2.9*



*Hình vẽ 2.9. Dữ liệu thu được trên cửa sổ terminal của Arduino IDE khi đã kết nối WiFi*

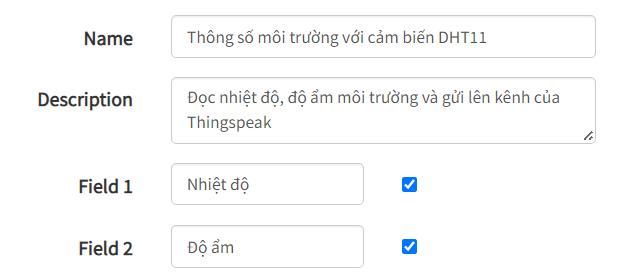
**Bươc 4**: Đẩy dữ liệu cảm biến lên Cloud

* Vào trang Web có địa chỉ [*https://thingspeak.com/channels*](https://thingspeak.com/channels) để tạo kênh mới của riêng mình và đặt tên kênh (*Ví dụ: Thông số môi trường với cảm biến DHT11*).



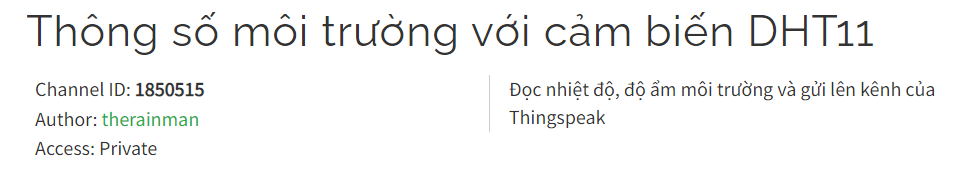
*Hình vẽ 2.10. Thiết lập kênh dữ liệu mới với Thingspeak*

* Chỉnh sửa các tham số của kênh (*chanel*) với hai trường (*Field1 và Field2*) là *Nhiệt độ* và *Độ ẩm*. Ví dụ như *Hình vẽ 2.11* sau đây:

****

*Hình vẽ 2.12. Thiết lập các thông số của kênh dữ liệu trên Thingspeak*

* Trong kênh của mình hãy ghi lại ID của kênh:…………………………………...



*Hình vẽ 2.13. ID của một kênh dữ liệu trên Thingspeak*

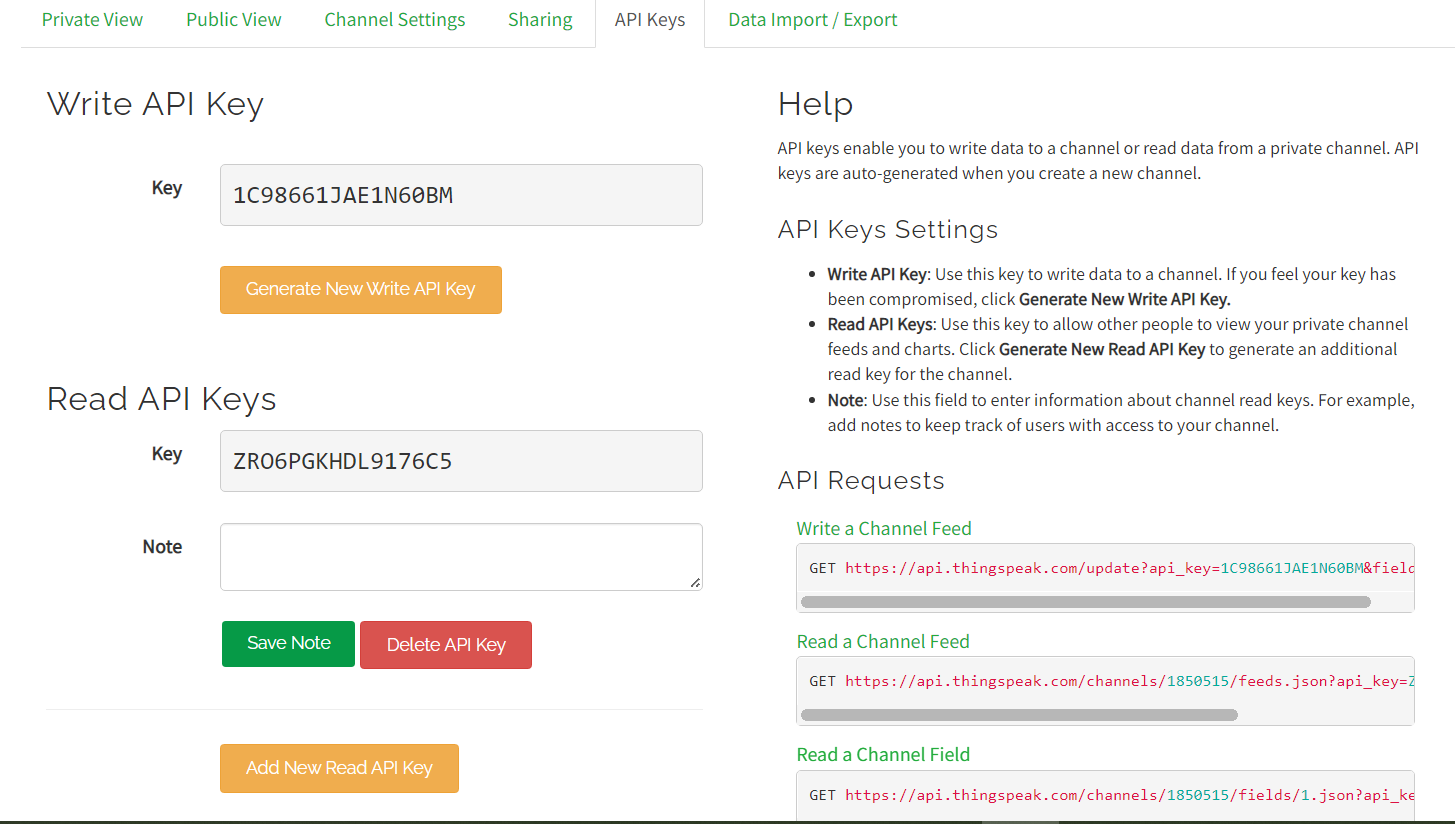
* Xem API key trên kênh của mình và ghi lại

Wirite API key:……………………………………………………………

Read API key:……………………………………………………………..

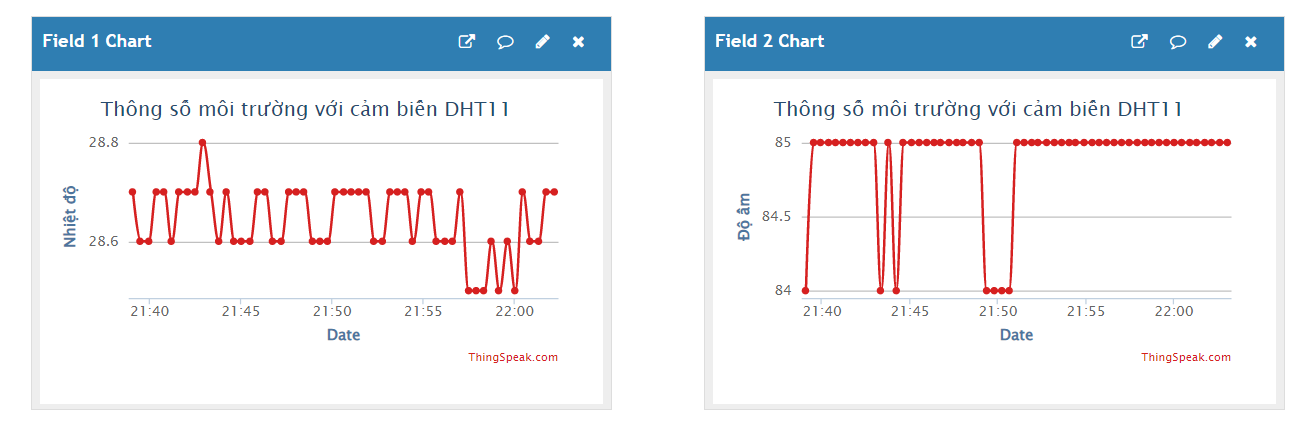
* Nêu vắn tắt chức năng của các API key này ?

................................................................................................................................................................................................................................................................................................................................................................................................................................................ ................................................................................................................................................................................................................................................................................................................................................................................................................................................



*Hình vẽ 2.13.. Ví dụ về API của một kênh dữ liệu trên Thingspeak*

* Với các API key đã có hãy sửa lại chương trình viết trên Arduino IDE để ESP32 gửi dữ liệu đọc từ DHT11 lên dịch vụ đám mây trên Thingspeak dùng giao thức HTTP và phương thức **GET**. *Chú ý: các đoạn mã viết với các mục đích riêng rẽ, cụ thể, nên được viết dưới dạng các chương trình con*:
* Hàm đo nhiệt độ, độ ẩm
* Hàm kết nối WiFi với ESP32
* Hàm gửi thông tin dữ liệu lên đám mây ThingSpeak
* Nạp chương trình cho vi điều khiển trên KIT IoT và quan sát kết quả thu được trên kênh của Thingspeak.
* Tác động vào cảm biến để thay đổi nhiệt độ và độ ẩm môi trường nhằm thay đổi các giá trị đo. Quan sát sự thay đổi trên Widget được dùng để hiển thị trên kênh của Thingspeak. Kết quả thu được có dạng như *Hình vẽ 2.14*



*Hình vẽ 2.14. Biểu diễn dữ liệu từ cảm biến DHT11 trên Thingspeak*

* Thay đổi các thiết lập của kênh và các Widget (Chart, Gauge, Numeric Display) cũng như các thiết lập của từng Widget để xem các cách thức biểu diễn dữ liệu của ThingSpeak. Hãy chọn 1 cách hiển thị dữ liệu dễ hiểu và trực quan nhất.
* Có bao nhiêu Widget trên kênh của ThingSpeak:……………………………………

Đó là những Widget nào ?.....................................................................................................

……........................................................................................................................................................................................................................................................................................

* Viết lại chương trình khi dùng POST với URL encoded và Plain text. Nhận xét 2 cách này khác gì khi dùng với JSON ?

……........................................................................................................................................................................................................................................................................................

……........................................................................................................................................................................................................................................................................................

……........................................................................................................................................................................................................................................................................................

……........................................................................................................................................................................................................................................................................................

……........................................................................................................................................................................................................................................................................................

……........................................................................................................................................................................................................................................................................................

**3.2 Giao thức MQTT**

**Đo nhiệt độ, độ ẩm môi trường, gửi dữ liệu đo lên dịch vụ đám mây HiveMQ**



*Hình vẽ 2.15. Sơ đồ khối hệ thống IoT với HiveMQ*

**Bước 1**: Kết nối thiết bị phần cứng trên KIT IoT

* Xác định khối mạch cảm biến DHT11 và các chân cắm tương ứng trên KIT thí nghiệm IoT.
* Xác định các chân cắm tương ứng của KIT Arduino ESP32 được sử dụng để nối với cảm biến DHT11.
* Cắm dây nhảy để kết nối giữa ESP32 và DHT11, ESP32 và LED đơn. Kết nối KIT thí nghiệm IoT với máy tính qua cổng USB. Bật nguồn điện KIT thí nghiệm IoT. Điền đầy đủ dữ liệu vào *Hình vẽ 1.10* dưới đây.



*Hình vẽ 2.16. Kết nối ESP32 và DHT11, LED đơn*

**Bước 2**: Lập trình cho KIT Arduino ESP32

* Cài đặt thư viện *PubSubClient* bổ sung cho Arduino IDE để có thể có các thư viện hỗ trợ kết nối giữa ESP32 và Broker HiveMQ theo giao thức MQTT.
* Viết chương trình cho ESP32 đọc dữ liệu nhiệt độ, độ ẩm và publish lên trang HiveMQ theo giao thức MQTT. Đồng thời ESP32 nhận dữ liệu *message* điều khiển bật tắt đèn LED đơn trên KIT thí nghiệm IoT.

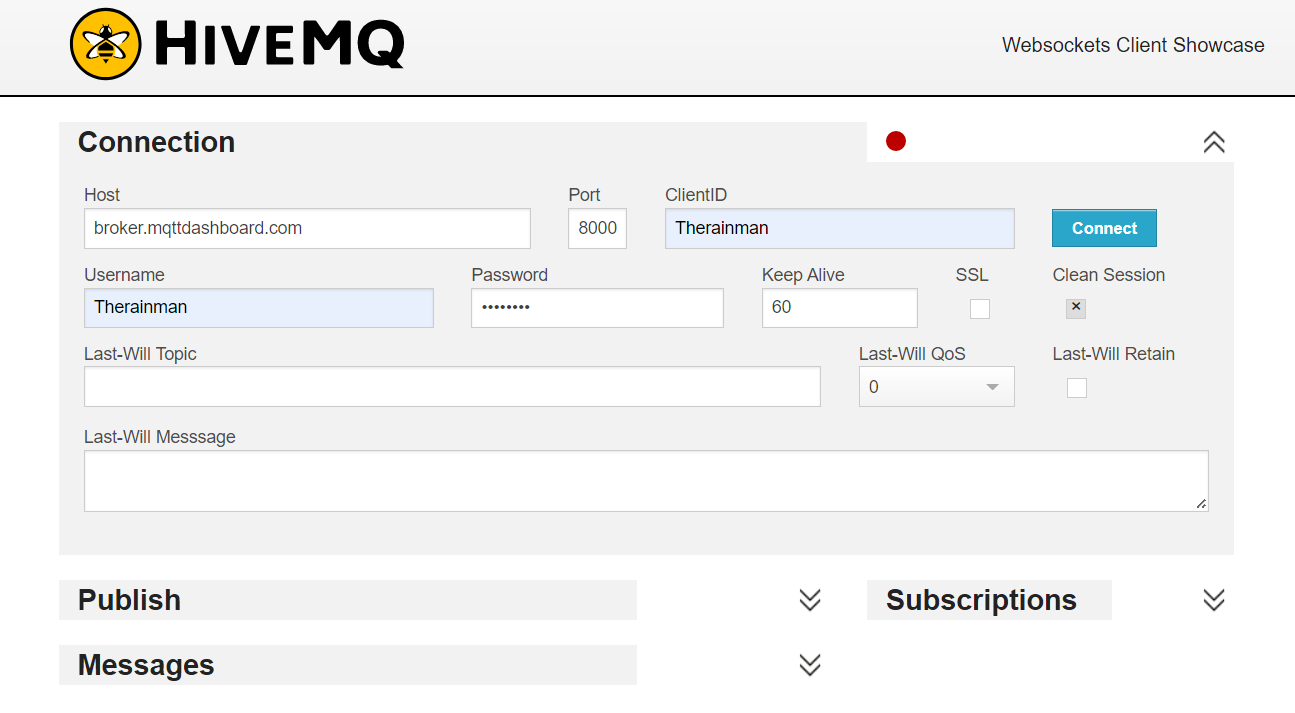
*Chú ý: chương trình vẫn có các nhiệm vụ chính như sau:*

* + Đọc dữ liệu nhiệt độ, độ ẩm của DHT11
  + Kết nối WiFi
  + Gửi và nhận dữ liệu nhiệt độ, độ ẩm với HiveMQ theo giao thức MQTT

**Bước 3**: Gửi dữ liệu lên Brocker HiveMQ

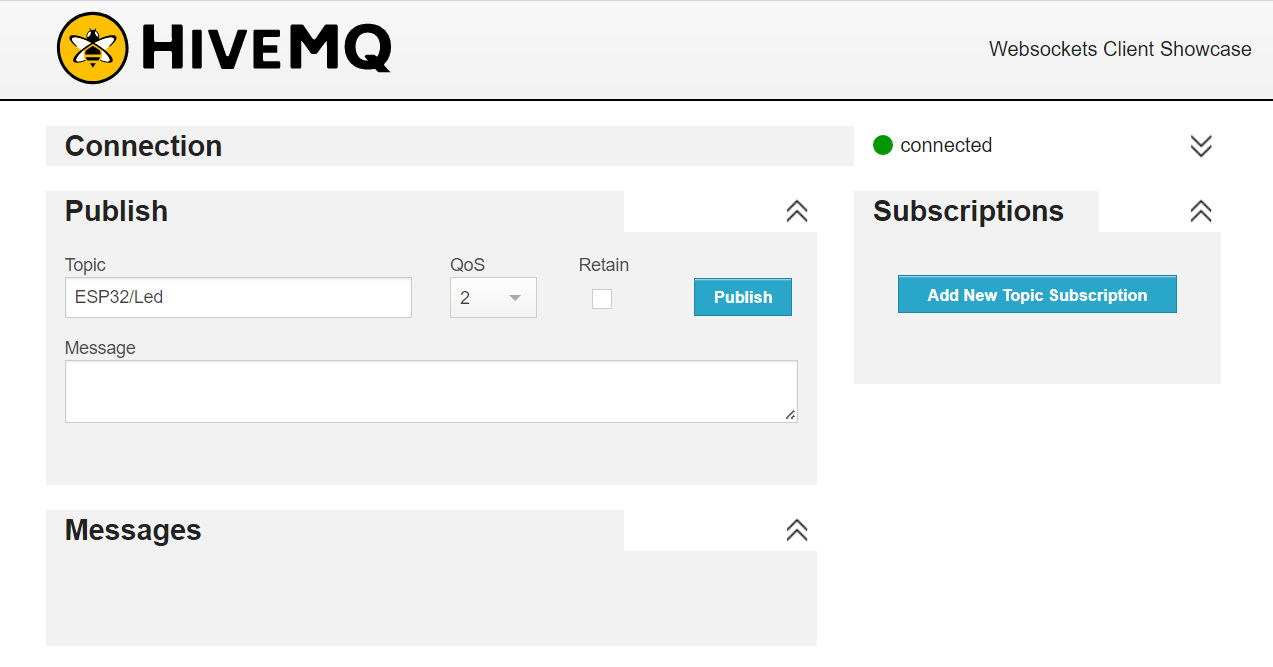
* Vào trang Web có địa chỉ [*http://www.hivemq.com/demos/websocket-client/*](http://www.hivemq.com/demos/websocket-client/)*.* Tạo một acount miễn phí. Ghi lại tên User Name và Password vào dưới đây:
  1. **User Name**:……………………………….
  2. **Password**:…………………………………

Đây là trang web cho phép tạo client kết nối trực tiếp tới HiveMQ Brocker . Tại trang web này, hãy tạo một client kết nối tới broker để subscribe 2 topic *Nhiệt độ* và *Độ ẩm* mà ESP32 publish dữ liệu lên và kiểm tra xem ESP32 có thực hiện publish thành công hay không. Đồng thời, client này cũng sẽ publish message tới topic đèn LED, để ESP32 nhận được message và điều khiển đèn LED đơn.



*Hình vẽ 2.17. Tạo client kết nối trực tiếp trên Web dùng MQTT*

* Nhấn vào nút *Connect* để kết nối và tạo topic: **ESP32/Led**

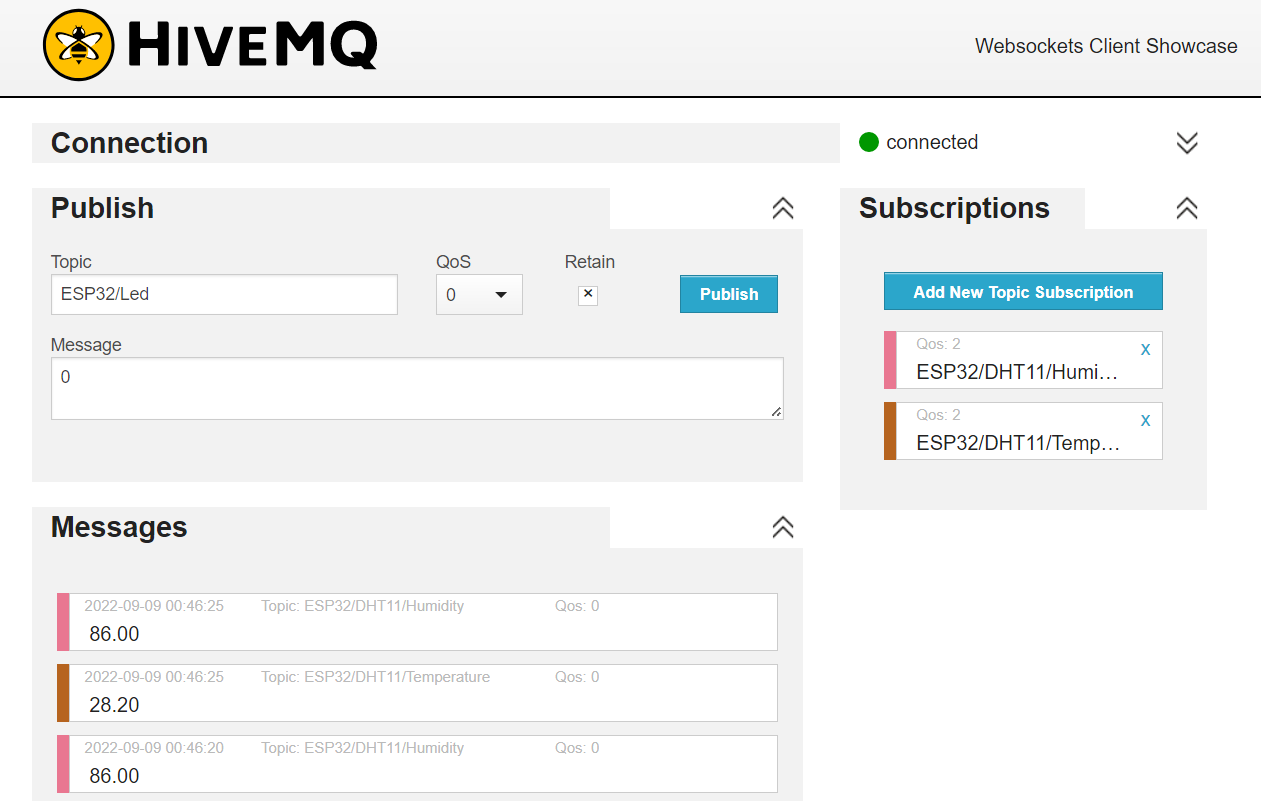


*Hình vẽ 2.18. Kết nối với HiveMQ*

* Nhấn vào nút *Add New Topic Subscription* để subscript các topic sau:
  + **ESP32/DHT11/Temperature**
  + **ESP32/DHT11/Humidity**

*Chú ý:*

* Topic **ESP32/Led** và topic được subscript là **ESP32/DHT11/Temperature** và **ESP32/DHT11/Humidity** phải tuyệt đối giống với các khai báo trong mã nguồn chương trình viết cho ESP32.
* **User Name** và **Password** nhập vào trang HiveMQ cũng phải giống với User Name và Password trong mã nguồn chương trình viết cho ESP32.
* Bật nguồn cho KIT thí nghiệm IoT và quan sát kết quả trên trang HiveMQ



*HÌnh vẽ 2.19. Kết quả chương trình*

* Trong phần message hãy nhập số 0 và sau đó nhấn nút Publish. Quan sát kết quả trên LED đơn được gắn với ESP32. Trạng thái đèn LED lúc đó là tắt hay sáng:………………………………………
* Trong phần message hãy nhập số 1 và sau đó nhấn nút Publish. Quan sát kết quả tren LED đơn được gắn với ESP32. Trạng thái đèn LED lúc đó là tắt hay sáng:……………………………………….
* Nhận xét về tốc độ bật tắt của LED khi nhấn nút Publish và giải thích hoạt động của hệ thống

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

* Ghi nhận và phân tích kết quả, so sánh hai giao thức HTTP và MQTT ?

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………