
DESLab IoT Training

Nguyen Vu Minh Thanh

02/10/2023

Mục lục

1	Tổng quan	3
2	Tổng quan về IoT	5
3	Thực hiện hệ thống	9
4	Tổng quan	31
5	Thực hiện hệ thống	37
6	Cài đặt nhận thông báo trên Arduino Cloud	57
	Tài liệu tham khảo	73

Đây là nội dung training hệ thống IoT của phòng thí nghiệm DESLab. Trong tài liệu training, tài liệu sẽ hướng dẫn xây dựng hệ thống IoT đơn giản. Hệ thống IoT sẽ giám sát nhiệt độ phòng thông qua cảm biến [LM35](#), module [Arduino ESP8266 Wifi Shield](#), và [Arduino Cloud](#).

CHƯƠNG 1

Tổng quan

Nội dung training IoT của PTN [DESLab](#) sẽ hướng dẫn xây dựng hệ thống IoT đơn giản giám sát nhiệt độ phòng. Cụ thể, hệ thống IoT sử dụng cảm biến nhiệt độ [LM35](#), module [Arduino ESP8266 Wifi Shield](#), và [Arduino Cloud](#).

1.1 Về kiến thức

Ghi chú: Do thời gian rất có hạn, nếu các bạn chưa có nhiều kiến thức về lập trình C, các bạn chỉ cần đọc qua 6 tài liệu tham khảo được đề cập và **không cần tìm hiểu quá chi tiết**.

Để xây dựng một hệ thống IoT, các kiến thức cần tìm hiểu bao gồm:

1. Kiến thức tổng quan về IoT¹².
2. Đọc hiểu datasheet của các thiết bị điện tử³.
3. Kiến thức lập trình C/C++ cơ bản⁴.
4. Tổng quan về Arduino Framework⁵.
5. Kiến thức tổng quan về Arduino Cloud⁶.

¹ Các khái niệm tổng quan về IoT trên Amazon.

² Khái niệm cơ bản về IoT Infrastructure trên Macrometa.

³ Datasheet của cảm biến LM35 trên TI.

⁴ Khóa học lập trình C cơ bản trên Codelearn.io.

⁵ Tổng quan về Arduino Framework.

⁶ Cơ bản về Arduino Cloud.

1.2 Về môi trường làm việc

Để xây dựng hệ thống IoT với Arduino Cloud, môi trường làm việc cần có những thiết lập sau:

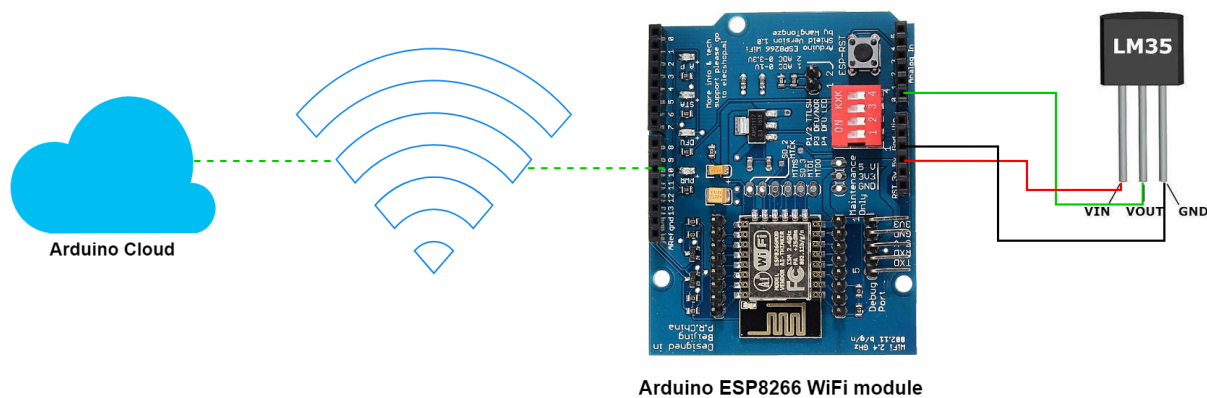
1. Máy tính cá nhân Windows/Linux/macOS.
2. Tài khoản đăng ký trên Arduino Cloud.
3. Phần mềm [Arduino Create Agent](#).

1.3 Về hệ thống IoT trong nội dung training

Hệ thống IoT trong nội dung training là hệ thống IoT giám sát nhiệt độ phòng. Trong hệ thống này:

1. Cảm biến [LM35](#) là ngoại vi giám sát nhiệt độ,
2. Vi điều khiển [ESP8266](#) là thiết bị tương tác với cảm biến và cloud.
3. [Arduino Cloud](#) thu nhận và hiển thị dữ liệu thiết bị.

Trên phần cứng, vi điều khiển ESP8266 tương tác với cảm biến LM35 thông qua chân **A0** (như hình minh họa dưới đây).



Hình1: Tổng quan về sơ đồ mạch.

CHƯƠNG 2

Tổng quan về IoT

Nội dung tổng quan về IoT được trình bày dựa trên tài liệu tham khảo trên Amazon [[AmazonIoT](#)] và Macrometa [[MacrometaIoT](#)].

2.1 Khái niệm IoT

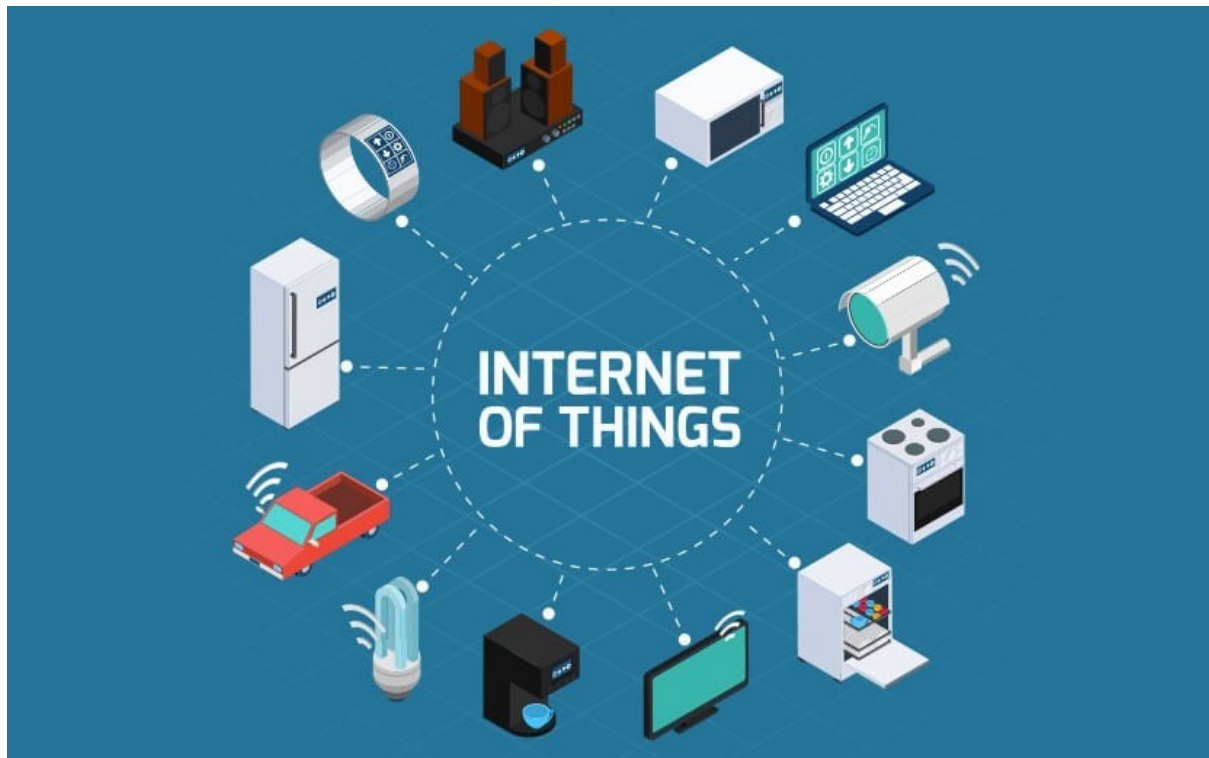
IoT (Internet of Things) là thuật ngữ mô tả tập hợp các network của các thiết bị được kết nối và công nghệ cung cấp giao tiếp giữa các thiết bị và cloud, cũng như giữa các thiết bị với nhau. Nhờ sự xuất hiện của các chip máy tính giá thành thấp và mạng viễn thông có băng thông tốc độ cao, chúng ta hiện nay có hàng tỷ thiết bị được kết nối với Internet. Điều này có nghĩa là các thiết bị hằng ngày như máy hút bụi, xe hơi, và các loại máy móc có thể sử dụng các cảm biến để thu thập dữ liệu và phản hồi một cách thông minh tới người dùng.

2.2 Các thành phần quan trọng trong IoT

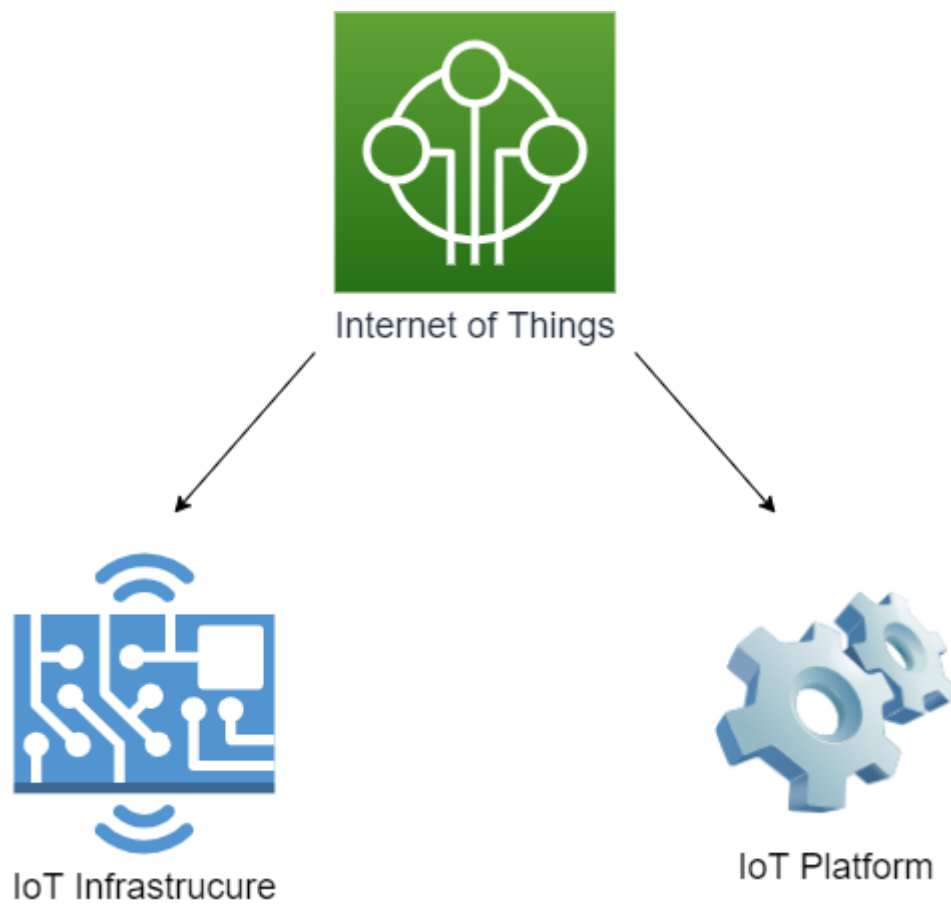
Trong IoT, hạ tầng và nền tảng IoT (minh họa như hình bên dưới) là hai thành phần quan trọng nhất.

Trong đó, trong mức độ cơ bản, IoT đề cập đến bất kỳ hệ thống các thiết bị được kết nối mà có các cảm biến và khả năng xử lý nhúng (embedded). Lưu ý rằng các thiết bị không nhất thiết phải sử dụng Internet. Kể cả các thiết bị được kết nối cục bộ tương tác và trao đổi dữ liệu, là một hệ thống IoT [[MacrometaIoT](#)].

Về IoT platform, một IoT platform là một ứng dụng hoặc dịch vụ mà cung cấp các công cụ có sẵn và khả năng để kết nối vạn vật trong một hệ sinh thái IoT. Bằng cách cung cấp các chức năng bao gồm quản lý chu kỳ thiết bị, giao tiếp thiết bị, phân tích dữ liệu, tích hợp, và hỗ trợ ứng dụng [[Softwareag](#)].



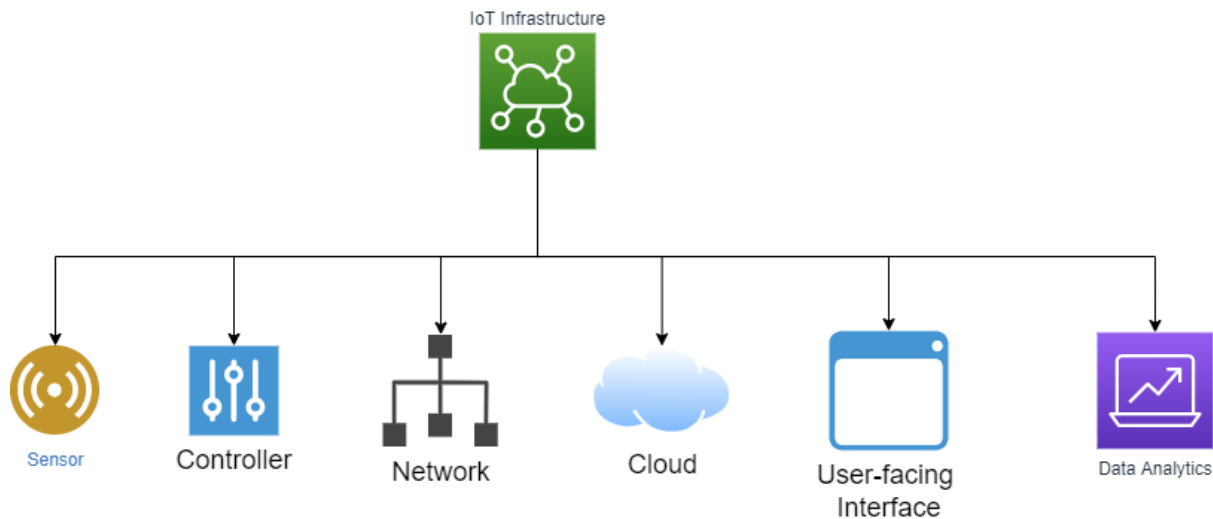
Hình1: Hình minh họa IoT. Nguồn từ [Karofi].



Hình2: Hình minh họa cơ bản về các thành phần trong IoT.

2.3 Các thành phần trong hạ tầng IoT

Trong hạ tầng IoT, hạ tầng IoT bao gồm 6 thành phần chính (theo tham khảo của Macrometa [MacrometaIoT] và minh họa như hình và bảng bên dưới). Đó là sensor, controller, network, cloud, user-facing applications, và data analytics.



Hình3: Hình minh họa cơ bản về các thành phần trong hạ tầng IoT.

Thành phần trong hạ tầng IoT	Mô tả
Sensor	Được sử dụng để đo lường các thông số vật lý mà thiết bị IoT chia sẻ thông qua mạng
Controller	Bộ não của thiết bị; hoạt động như một cầu nối giữa cảm biến và mạng và cũng thực hiện quá trình tính toán và lưu trữ trên board
Network	Công nghệ được sử dụng cho việc trao đổi dữ liệu với cả các thiết bị trong hệ thống hoặc cloud
Cloud	Các tài nguyên tính toán, lưu trữ, và gateway có thể truy cập thông qua Internet
User-facing interface	Các ứng dụng di động và web mà cho phép người dùng tương tác với hệ thống IoT và trực quan hóa dữ liệu
Data Analytics	Các công cụ và tài nguyên (thường trên cloud) mà cho phép người dùng hiểu rõ hơn về dữ liệu gửi lên hệ thống IoT

CHƯƠNG 3

Thực hiện hệ thống

3.1 Tổng quan

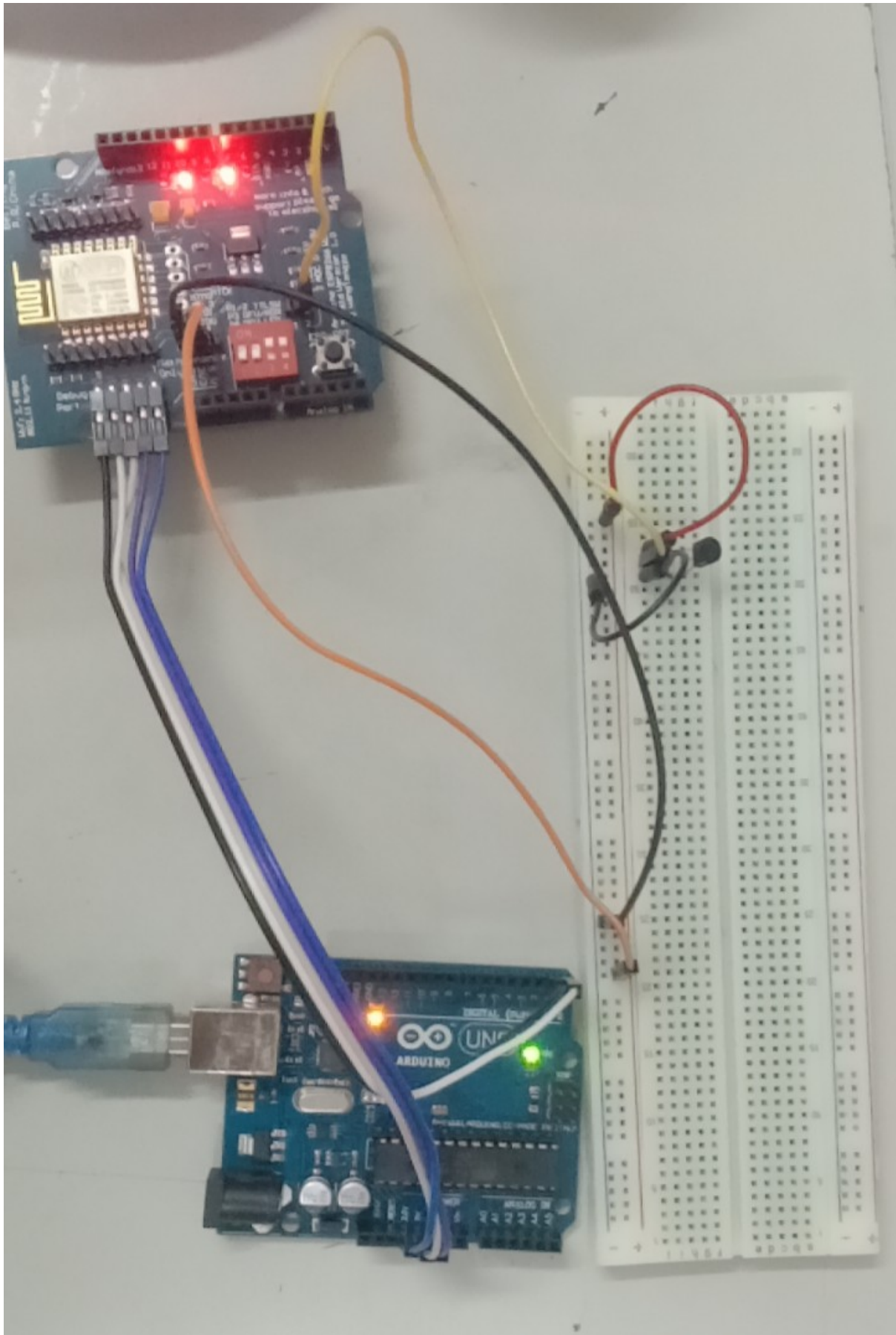
Xây dựng hệ thống IoT giám sát nhiệt độ phòng. Trong đó, bo mạch sử dụng mạch [Arduino Uno R3](#) và [Arduino ESP8266 Wifi Shield](#). Thông số nhiệt độ phòng được đo từ cảm biến [LM35](#), được hiển thị trên giao diện web của Arduino Cloud (minh họa như hai hình bên dưới).

3.2 Lắp ráp phần cứng

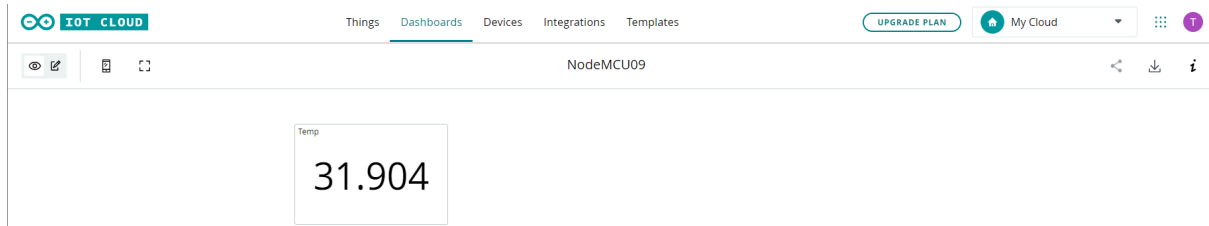
3.2.1 Các linh kiện cần chuẩn bị

6 loại linh kiện và bo mạch cần chuẩn bị (minh họa như hình bên dưới).

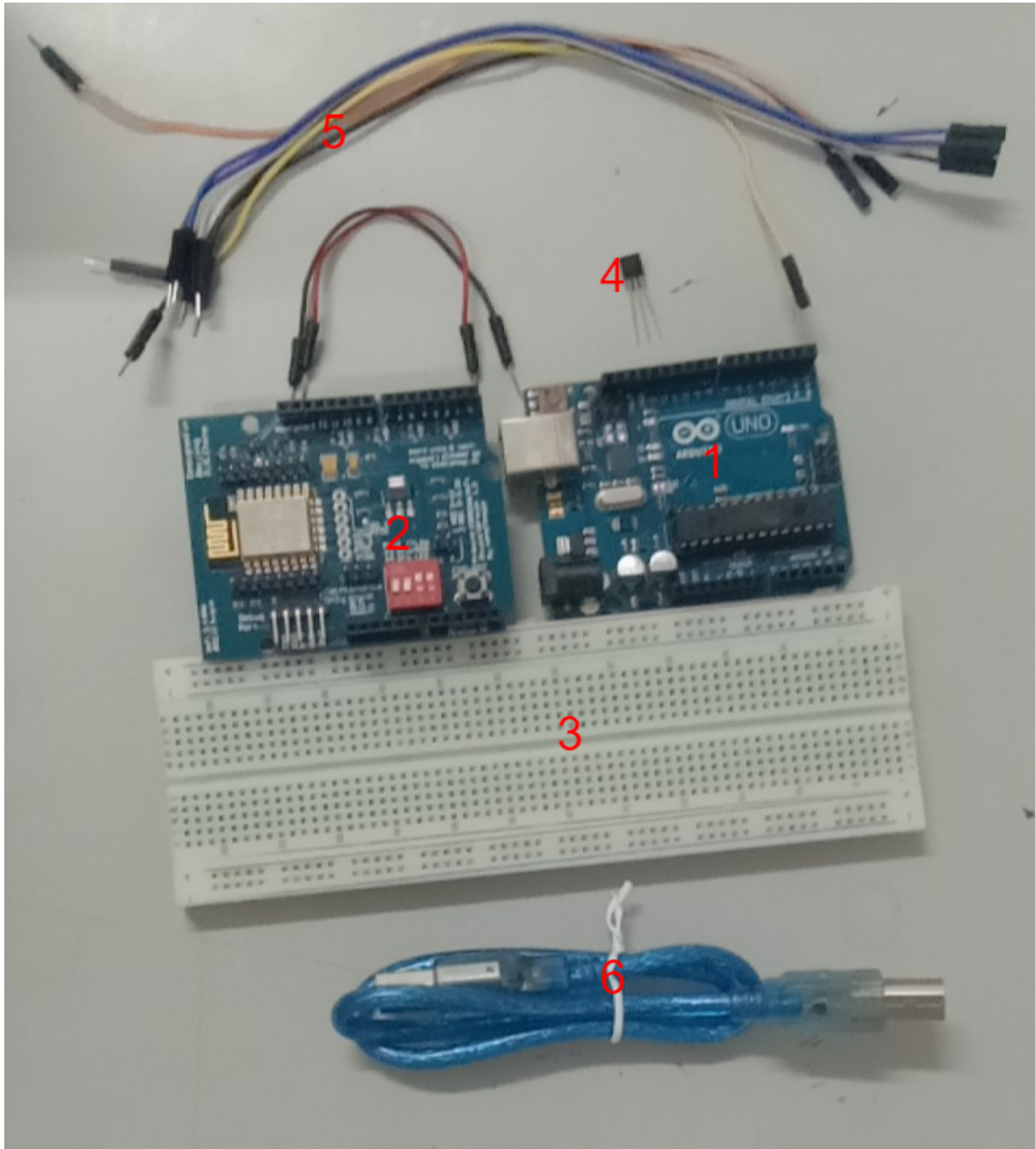
1. 1 mạch Arduino Uno R3.
2. 1 mạch Arduino ESP8266 Wifi Shield.
3. 1 breadboard.
4. 1 cảm biến LM35DZ
5. 8 dây đực-cái và 2 dây đực-đực.
6. 1 dây nạp USB.



Hình1: Hình minh họa tổng quan về mạch IoT.



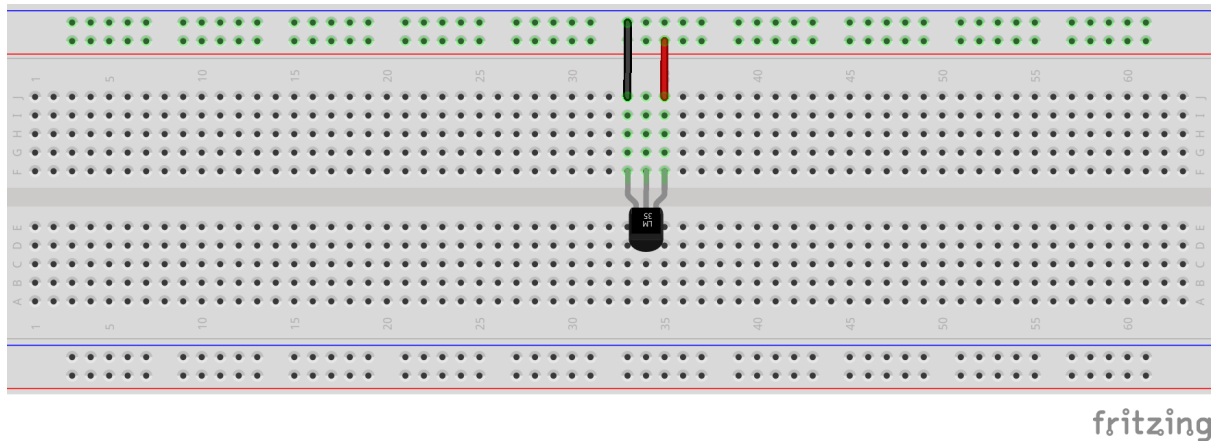
Hình2: Hình minh họa tổng quan về giao diện Web trên Arduino Cloud.



Hình3: Hình minh họa các linh kiện phần cứng.

3.2.2 Sơ đồ mạch

1. Nối cảm biến trên breadboard



2. Kết nối breadboard với Arduino ESP8266 Wifi Shield

Nối dây 3.3V, GND, và chân Analog 3.3V từ Shield tới breadboard.

3. Kết nối Arduino ESP8266 Wifi Shield và Arduino Uno R3

4. Kết dùng cáp USB nối mạch Arduino Uno R3 vào máy tính

3.3 Kết nối Arduino Cloud

3.3.1 Thiết lập cấu hình cloud

1. Mở Arduino Create Agent

Khi Arduino Create Agent được bật, icon của Arduino Create Agent sẽ hiển thị dưới khay icon của Windows.

2. Truy cập vào trang chủ của [Arduino Cloud](#).

Mở trang Arduino IoT Cloud.

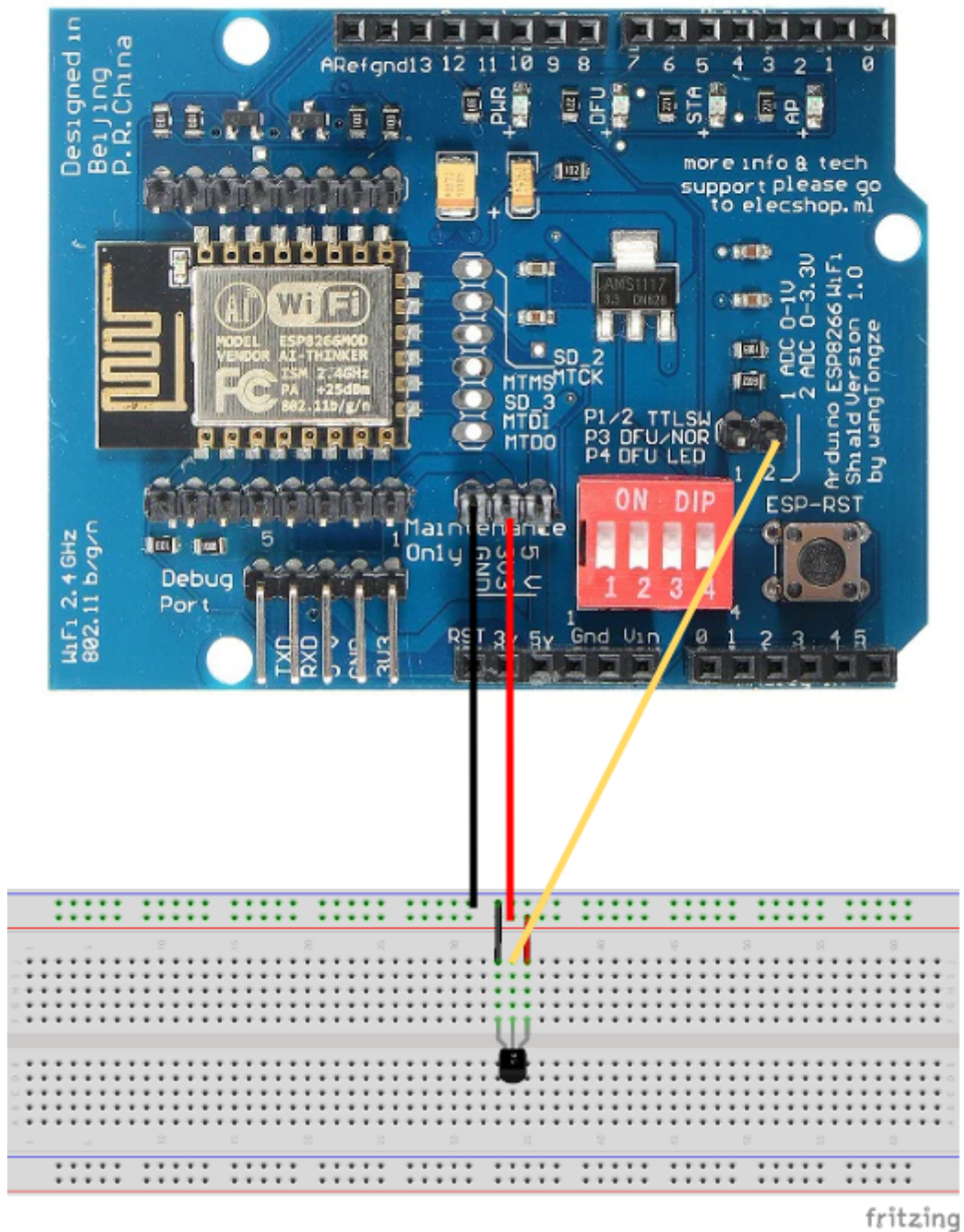
3. Tạo mới một thing

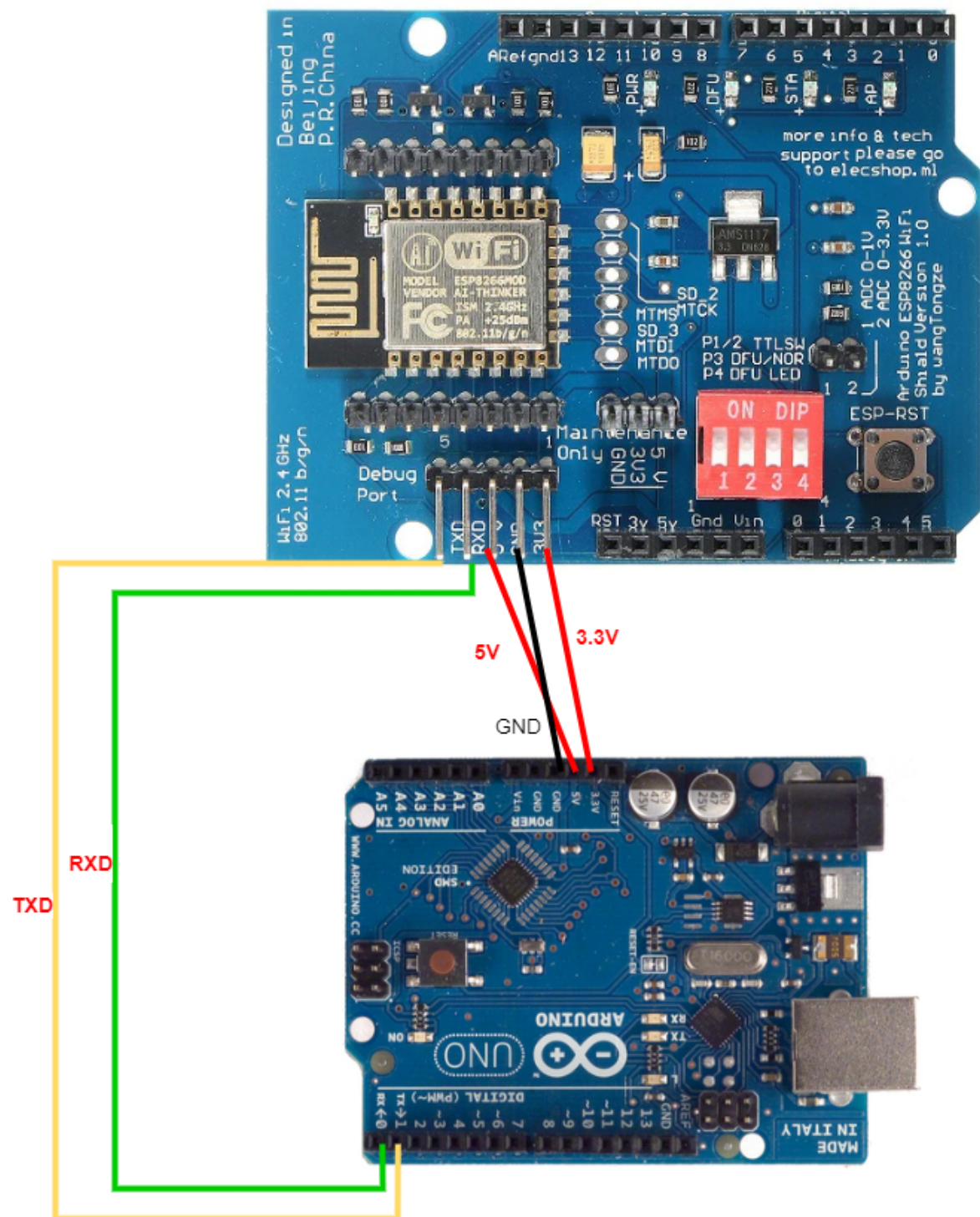
Sau đó, đặt tên cho thing và tạo device.

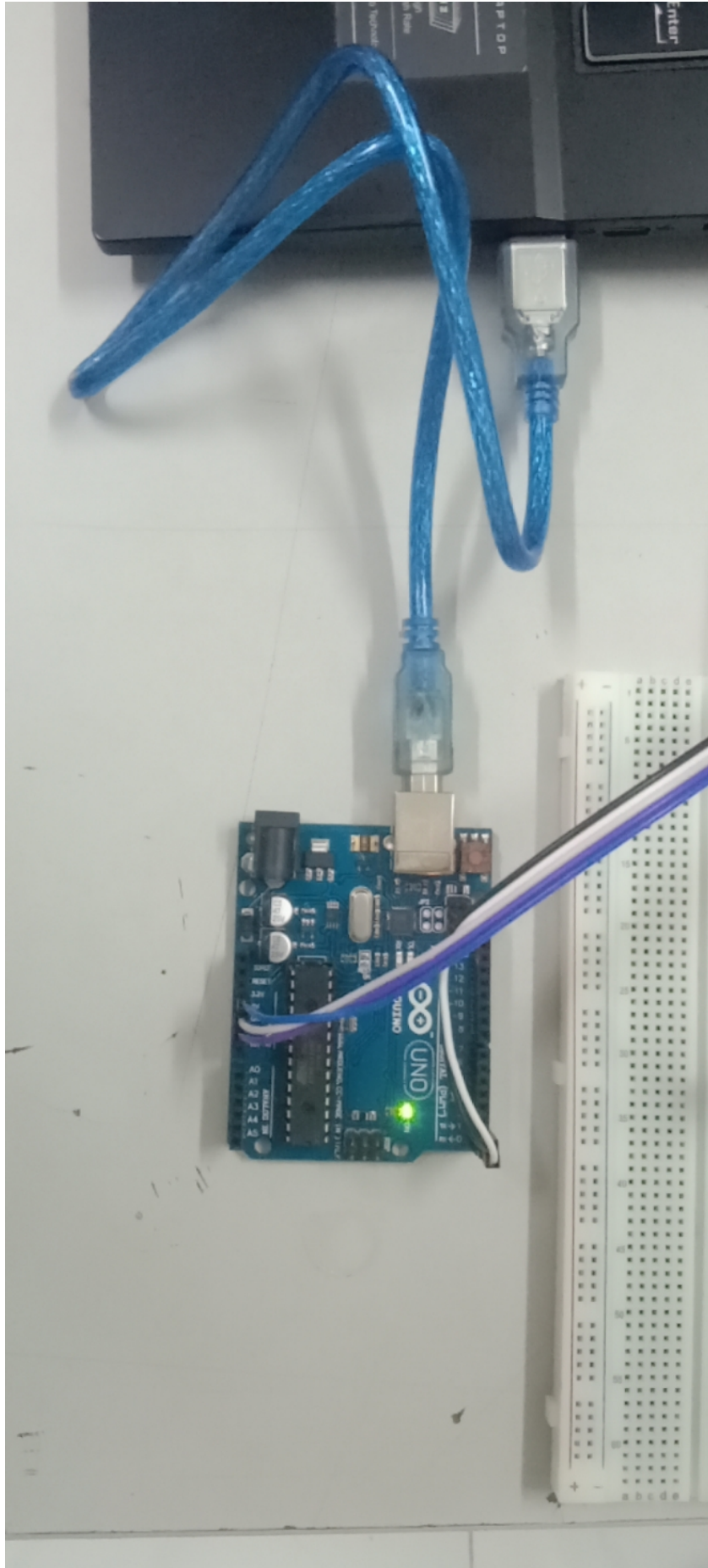
4. Tạo mới một device

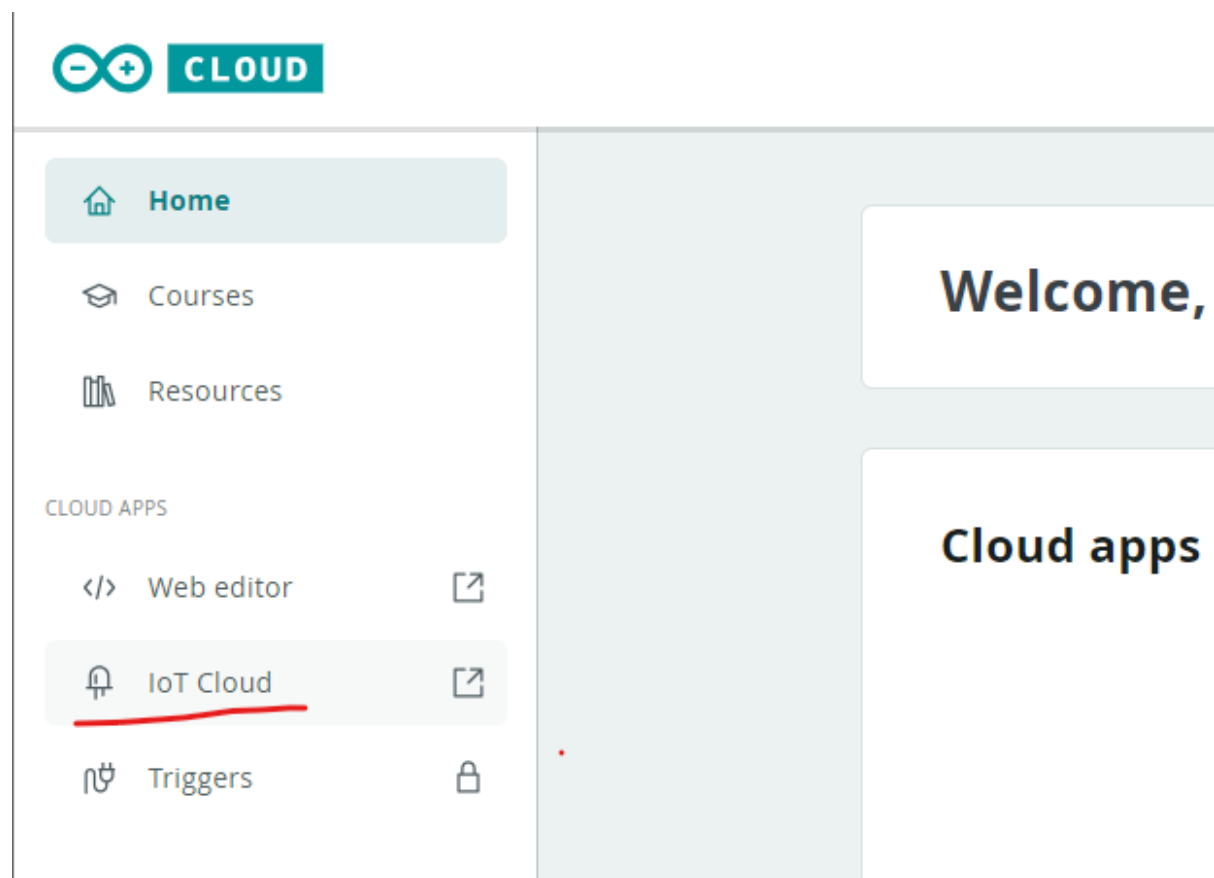
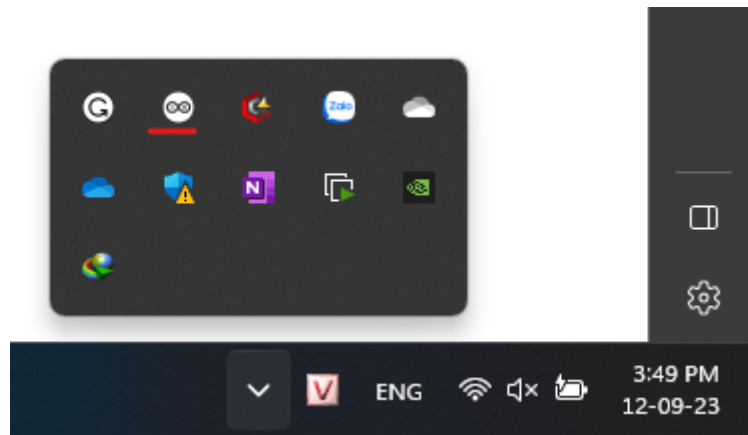
Chọn **Third party device**.

Lựa chọn thiết bị **ESP8266** và board **NodeMCU 0.9 (ESP-12 Module)**. Sau đó chọn **CONTINUE**.









[Things](#)
[Dashboards](#)
[Devices](#)
[Integrations](#)
[Templates](#)

[UPGRADE PLAN](#)

My Cloud

Create your first Thing

A Thing is a connected device that can communicate with the cloud. You can make your Things interact with other Things or anything else in the physical world.

[CREATE THING](#)

[Things](#)
[Dashboards](#)
[Devices](#)
[Integrations](#)
[Templates](#)

[UPGRADE PLAN](#)

My Cloud

NodeMCU09

[Setup](#)
[Sketch](#)
[Metadata](#)

Cloud Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

[ADD VARIABLE](#)

Associated Device

Select the device you want to use or configure a new one.

Select Device

Associate device

Choose the device to associate to NodeMCU09

NO DEVICE FOUND

[SET UP NEW DEVICE](#)

Setup Device
✕

RECOMMENDED

AUTOMATIC
Auto-sketch generation, online editing, easy upload



Arduino board ⓘ
</> Arduino language (C++)



Third party device ⓘ
</> Arduino language (C++)

MANUAL
Manual programming and upload, offline editing



DIY

Any Device ⓘ
</> Python, MicroPython, JavaScript (NodeJS)

← Setup Device ×

Select device type

Please select the device type and model you want to configure

☒ ESP8266
 ☐ ESP32
 ☐ LoRaWAN

NodeMCU 0.9 (ESP-12 Module) ▼

CONTINUE

Đặt tên cho device và chọn **NEXT**.

Lưu các thông tin của device, chọn xác nhận đã lưu thông tin device, và chọn **CONTINUE**.

5. Cấu hình network

Chọn vào ô cấu hình **network**.

Điền thông tin **Wi-Fi SSID**, **Wi-Fi PASSWORD**, và **Secret key** trong file PDF lưu thông tin device. Sau đó chọn **Save**.

6. Tạo Variable.

Đặt tên (**Name**), chọn kiểu dữ liệu (**Floating Point Number**), cấu hình **Variable Permission**, và chọn **ADD VARIABLE**.

Chuyển qua tab **SKETCH** và mở **full editor**.

6. Lập trình firmware của bo mạch.

Cấu hình loại **board** và **com port**.

Chọn loại board và port, và chọn **OK**.

Thêm các dòng code đo nhiệt độ.


```
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/2abe9f1e-53be-444a-98c4-
→066f070ac92e
```

(continues on next page)

←

Setup Device

×



Give your device a name

Name your device so you will be able to recognize it.


Device Name

NodeMCU-Device

↻

NEXT

Setup Device ✕



Make your device IoT-ready

To use this board you will need a Device ID and a Secret Key, please copy and save them or [download the PDF](#).


Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID

2e74266d-f16f-4fe1-b998-02740e513c76

Secret Key

UVRACKGB3Y4WFBVRV80AC

**Secret key cannot be recovered**

Please keep it safe, if you lose it you will have to delete and setup your device again.

☒ I saved my device ID and Secret Key [CONTINUE](#)

3.3. Kết nối Arduino Cloud

21

Network

Enter your network credentials to connect your device.



Configure

Configure network



You will find these network parameters in the secret tab in your sketch, and your device will be able to connect to the network once the sketch will be uploaded.

Wi-Fi Name *

Password



Secret Key *



SAVE

Cloud Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

ADD VARIABLE



(continued from previous page)

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when
→ changes are made to the Thing

```
float temperature;
```

Variables which are marked as READ/WRITE in the Cloud Thing will also
→ have functions

which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of
→ this sketch.

```
*/
```

```
#include "thingProperties.h"
```

```
+ int sensorPin = A0;
```

```
void setup() {
```

```
// Initialize serial and wait for port to open:
```

```
Serial.begin(9600);
```

```
// This delay gives the chance to wait for a Serial Monitor without  
→ blocking if none is found
```

```
delay(1500);
```

```
// Defined in thingProperties.h
```

```
initProperties();
```

```
// Connect to Arduino IoT Cloud
```



```
ArduinoCloud.begin(ArduinoIoTPreferredConnection);
```

```
/*
```


(continues on next page)


Add variable
X

Name
Temperature


 Sync with other Things 

Floating Point Number eg. 1.55

Declaration
float temperature ; 

Variable Permission 

☐ Read & Write
☒ Read Only

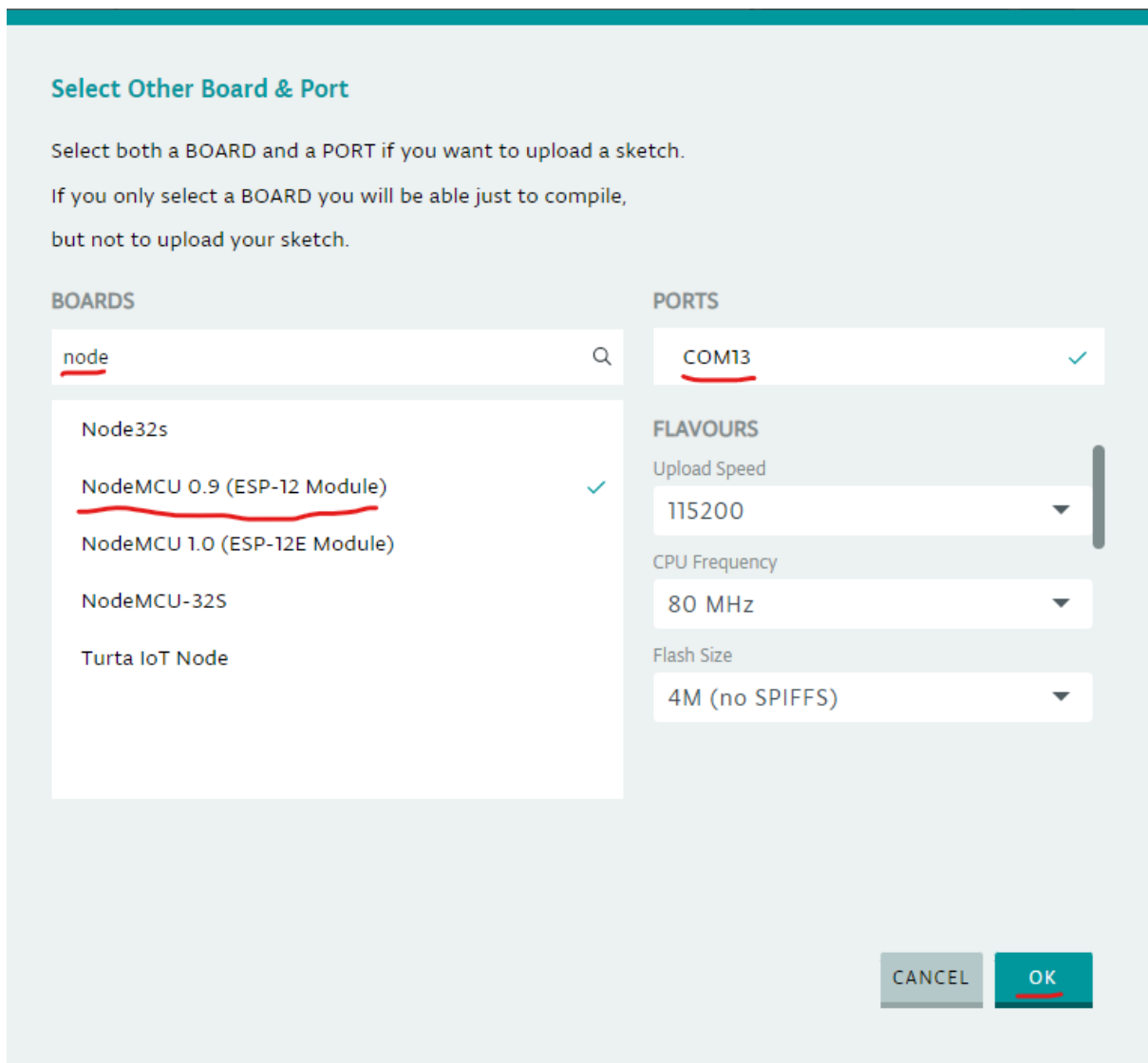
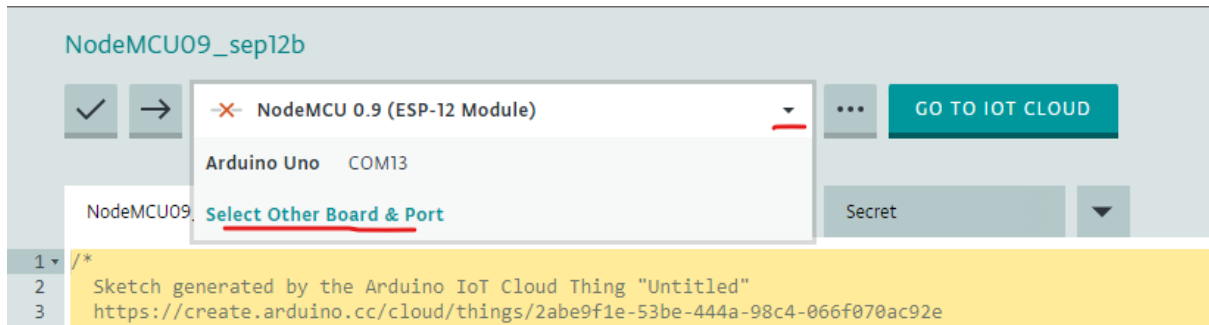
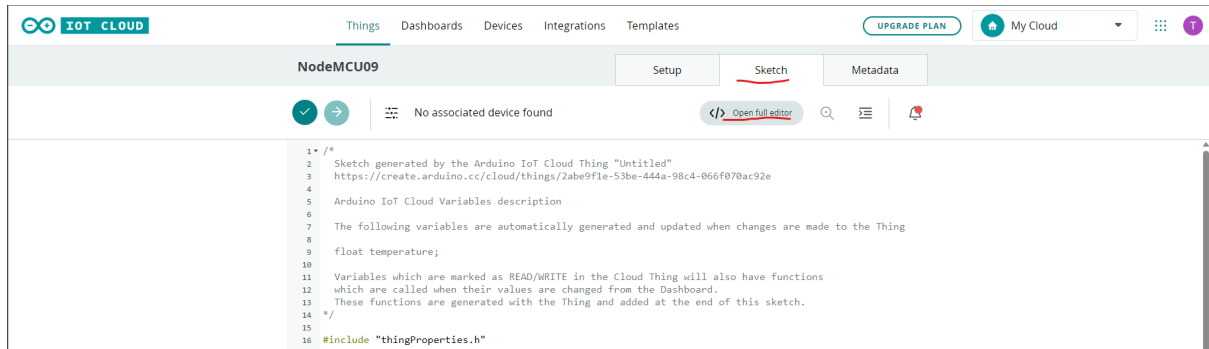
Variable Update Policy 

☒ On change
☐ Periodically

Threshold
0

ADD VARIABLE

CANCEL



(continued from previous page)

```

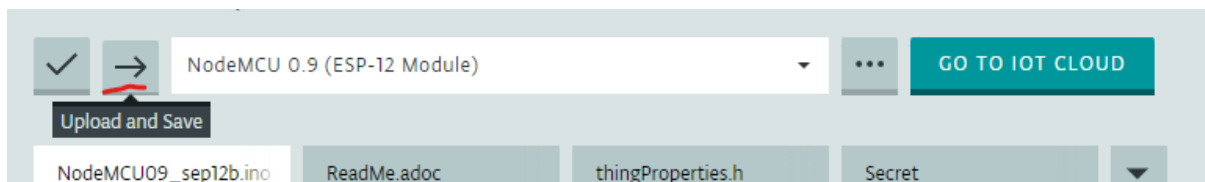
The following function allows you to obtain more information
related to the state of network and IoT Cloud connection and errors
the higher number the more granular information you'll get.
The default is 0 (only errors).
Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  + int reading = analogRead(sensorPin);
  + float voltage = reading * 3.3 / 1024.0;
  + temperature = voltage * 100.0;

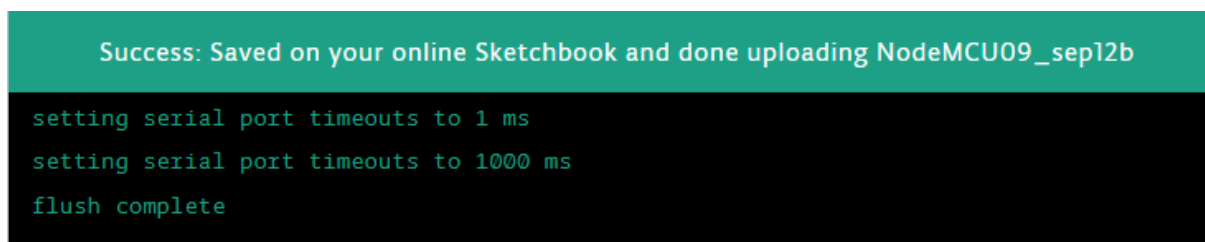
  + Serial.println(temperature);
  + delay(1000);
}

```

Nạp code vào board



Thông báo quá trình upload thành công



Chọn chế độ **Monitor** và theo dõi các giá trị nhiệt độ đo được.

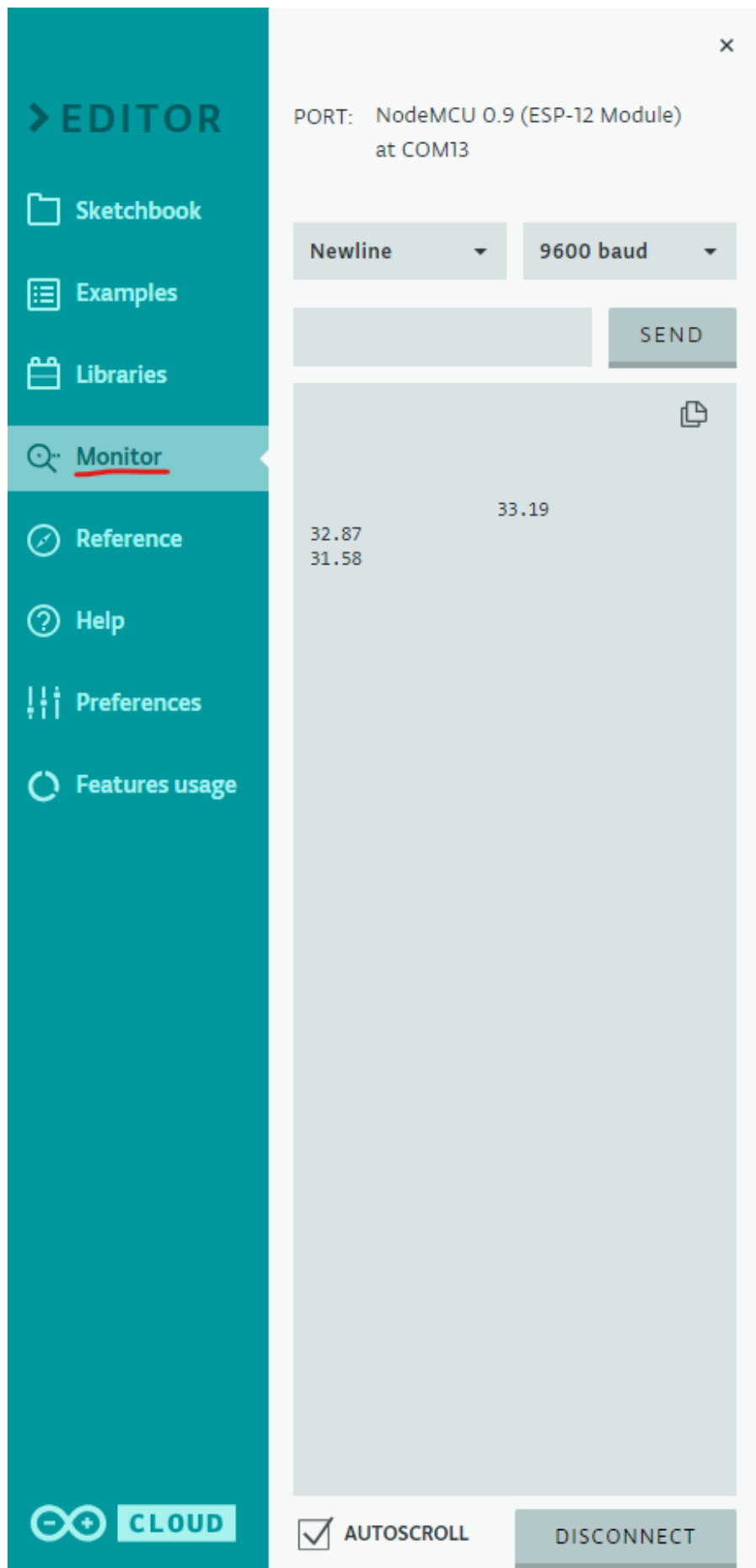
7. Tạo Dashboard

Vào IoT Cloud, chọn tab **Dashboards**, và chọn **BUILD DASHBOARD**.

Đặt tên cho Dashboard, và tạo widget **Value**.

Đặt tên cho widget và chọn **Linked Variable**.

Liên kết widget với **Temperature** Variable. Sau đó, chọn **Done**.










Monitor your Things


Build a Dashboard to easily monitor the status of your Things and control them.











BUILD DASHBOARD

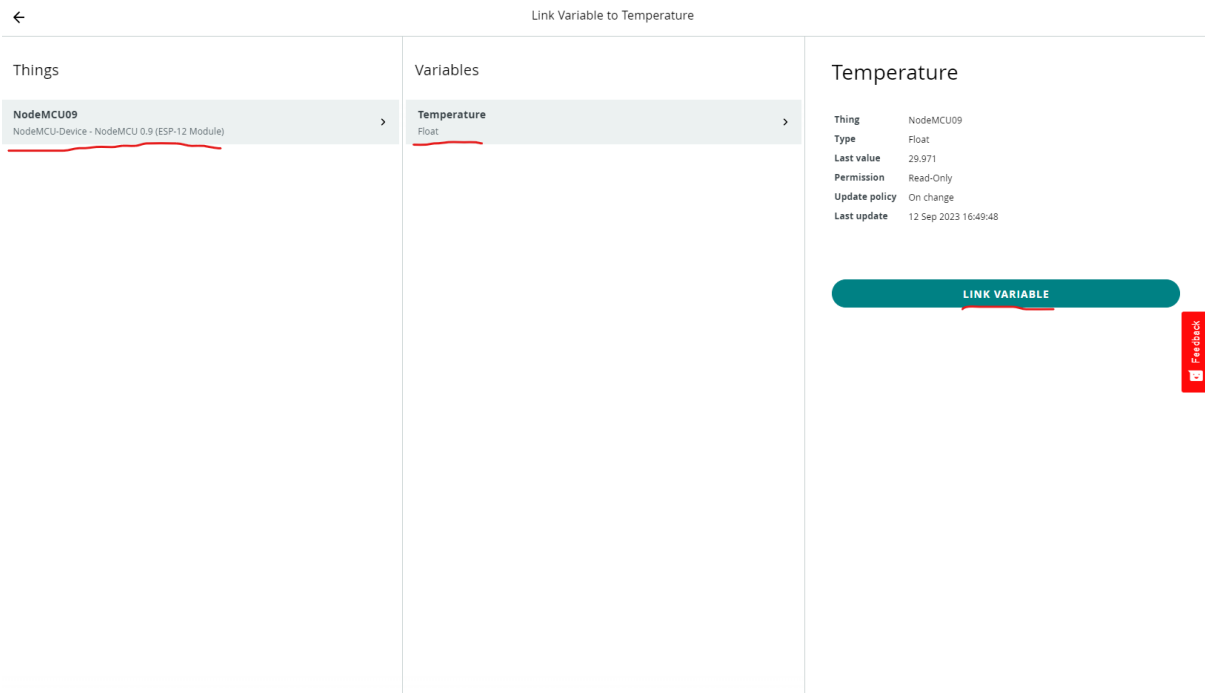
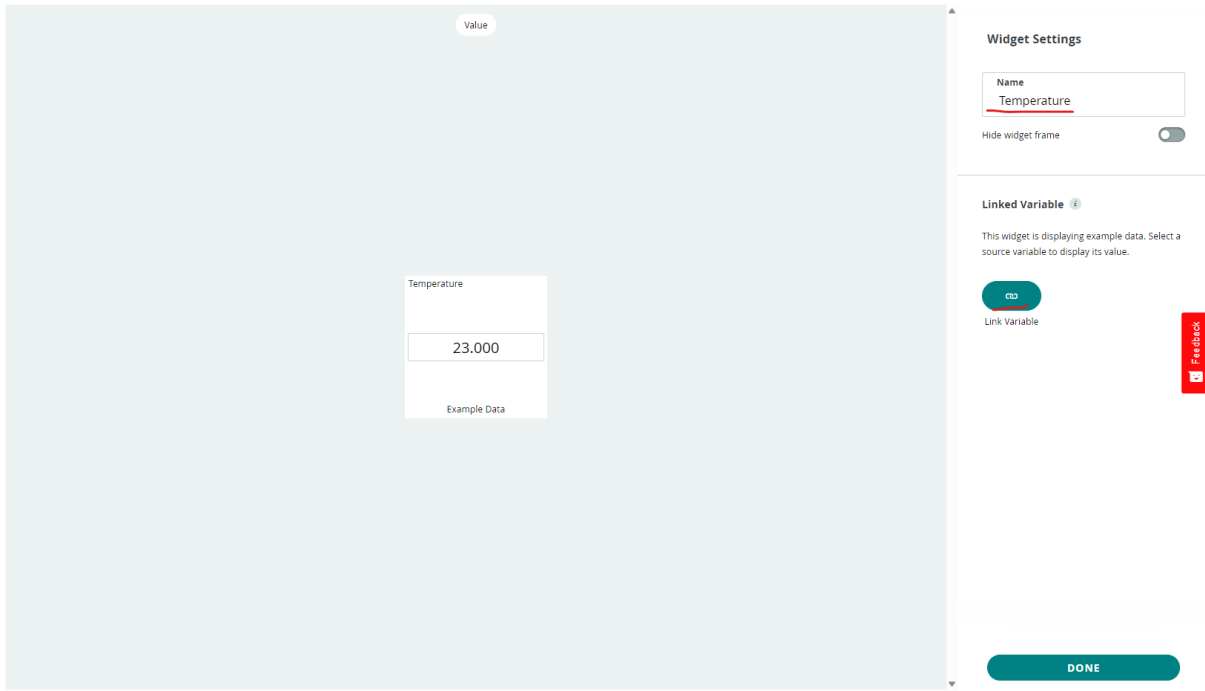
 **IOT CLOUD** Things Dashboards Devices Integrations Templates

  **ADD**   NodeMCU09

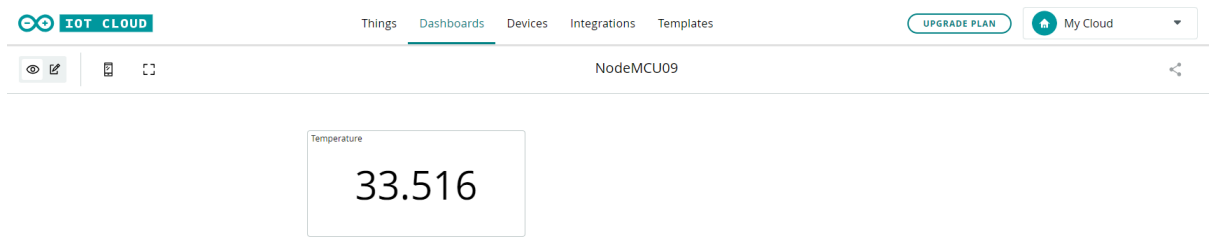
WIDGETS **THINGS**

 Search widgets

-  Dimmed light
-  Colored light
-  Time Picker
-  Scheduler
-  Value
-  Value Selector
-  Value Dropdown
-  Status
-  Gauge
-  Percentage



Quan sát dữ liệu nhiệt độ real-time trên dashboard đã tạo.



Ghi chú: Do có nhiều lỗi phát sinh khi dùng Arduino Cloud, nội dung training dùng Arduino IDE sẽ được trình bày theo nội dung dưới đây.

CHƯƠNG 4

Tổng quan

Nội dung training hướng dẫn xây dựng hệ thống IoT giám sát nhiệt độ phòng sử dụng cảm biến LM35, module [NodeMCU V3 CH340](#).

4.1 Môi trường làm việc

Môi trường làm việc cần được thiết lập các thành phần sau đây:

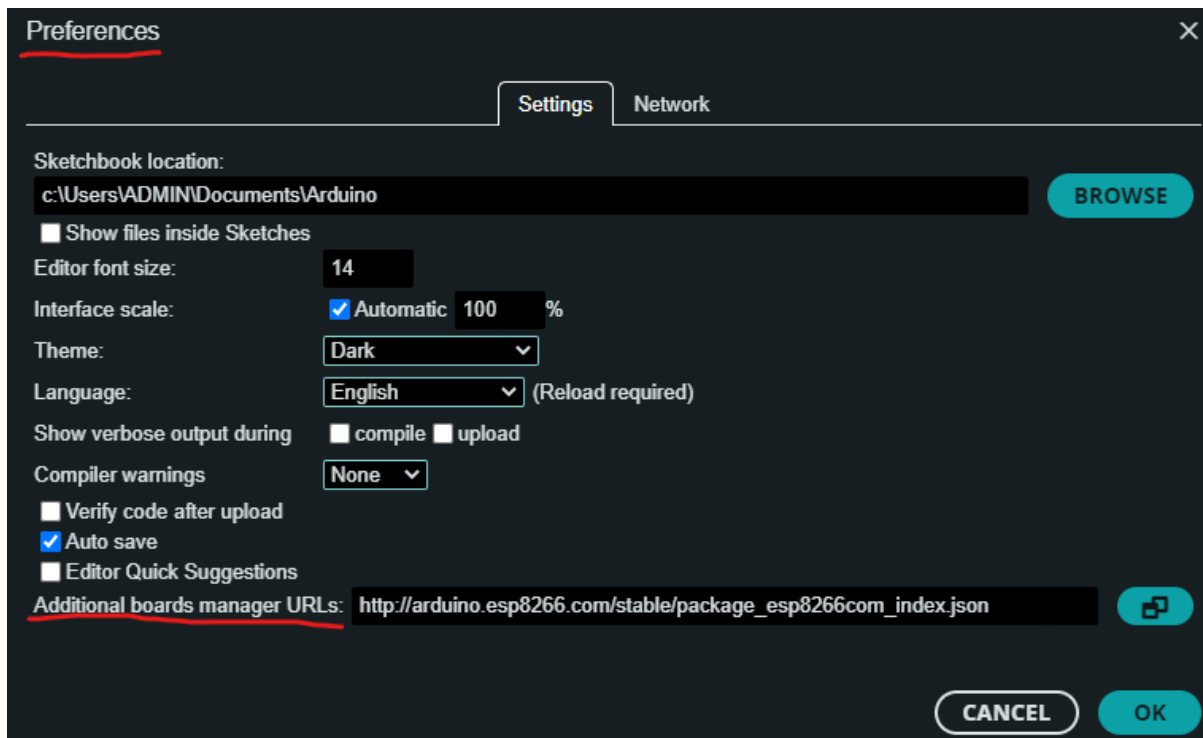
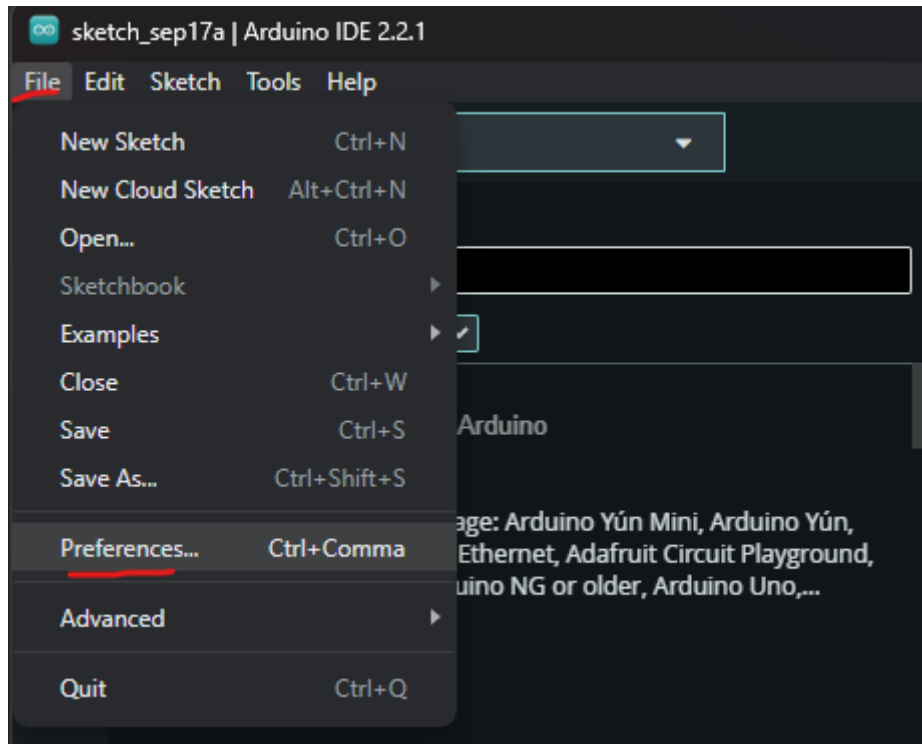
1. Cảm biến LM35, module [NodeMCU V3 CH340](#), breadboard, và dẫn dẫn (được cung cấp khi tham gia training).
2. Máy tính Windows/Linux/MacOS.
3. Driver [CH340](#).
4. Phần mềm [Arduino IDE 2.x](#).
5. Tài khoản [Arduino Cloud](#).

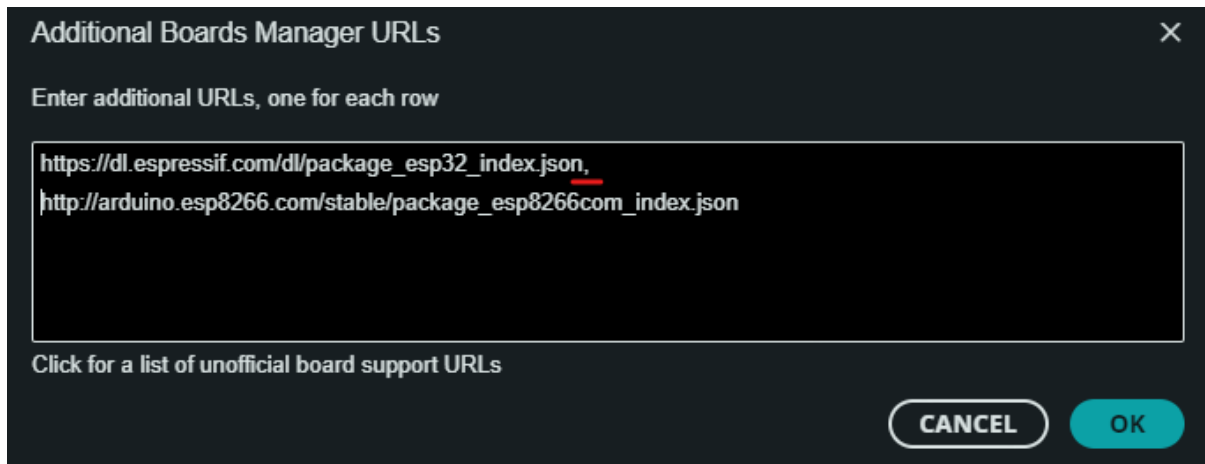
4.2 Các thiết lập cần có trên Arduino IDE

1. Cài đặt **Additional Boards Manager URLs**.

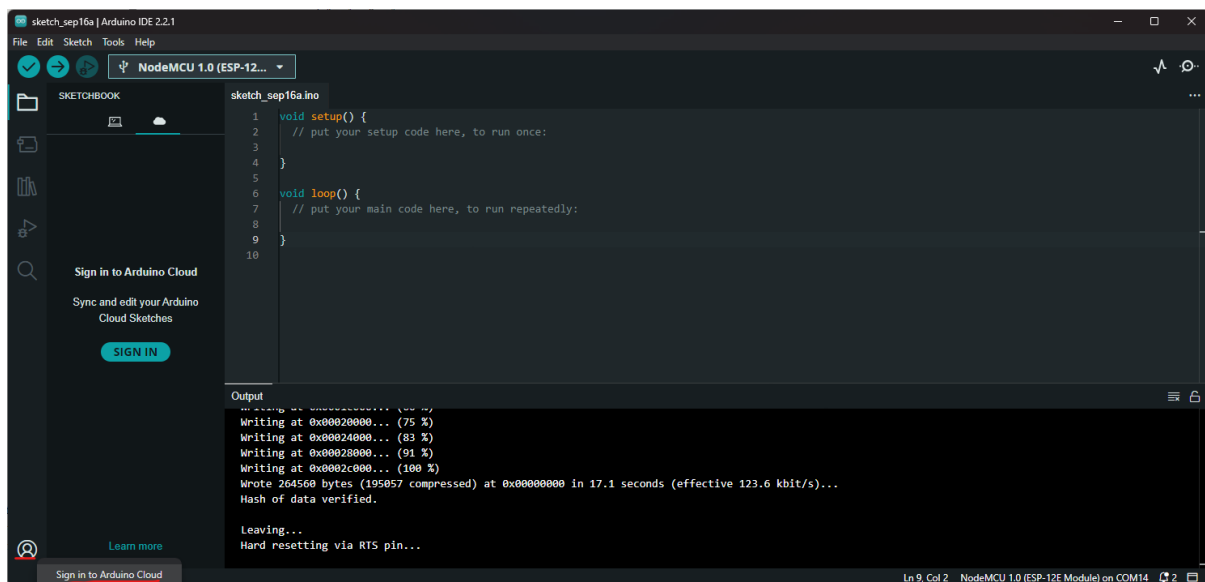
Vào **File** → **Preferences...** Sau đó, trong cửa sổ **Preferences**, thêm đường dẫn http://arduino.esp8266.com/stable/package_esp8266com_index.json vào mục **Additional boards manager URLs**.

Ghi chú: Nếu đã có URL trong mục này, thêm dấu phẩy "," và thêm URL trên, vào sau đó.



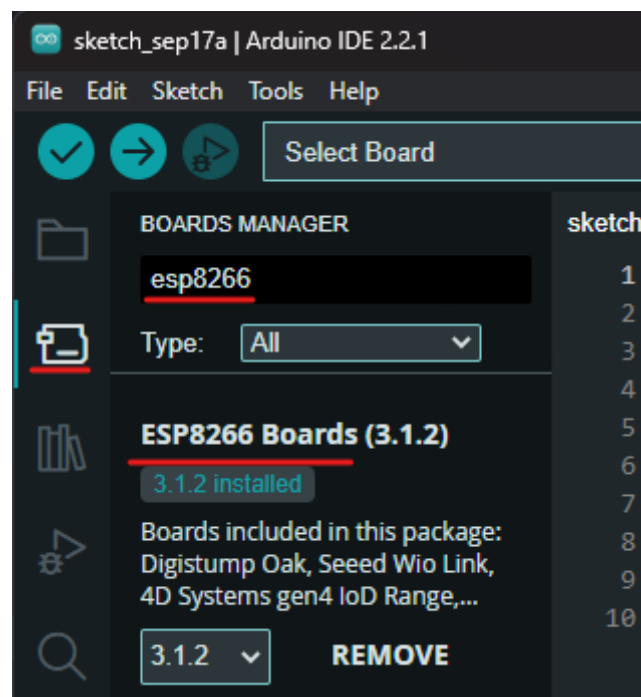


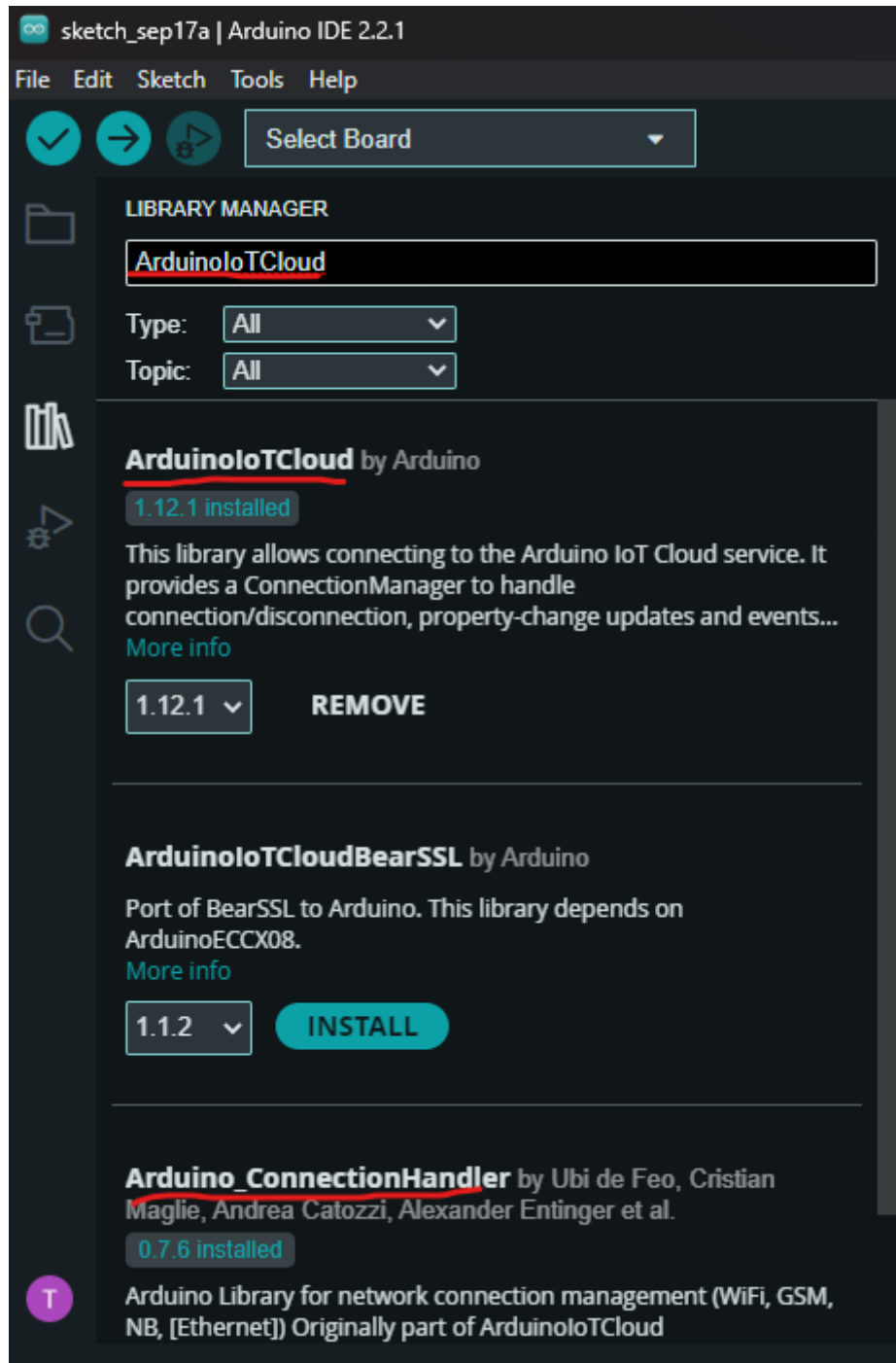
2. Đăng nhập tài khoản Arduino Cloud.



3. Cài đặt dữ liệu của vi điều khiển **ESP8266** trong phần **Boards Manager**.

4. Cài đặt thư viện **ArduinoIoTCloud** và **Arduino_ConnectionHandler** trong phần **Library Manager**.





CHƯƠNG 5

Thực hiện hệ thống

5.1 Tổng quan

Nội dung training hướng dẫn xây dựng hệ thống IoT giám sát nhiệt độ phòng. Cụ thể, hệ thống sử dụng cảm biến **LM35**, module **NodeMCU V3 CH340**, **Arduino Cloud**, và **Arduino IDE 2.x**.

5.2 Thiết lập bo mạch

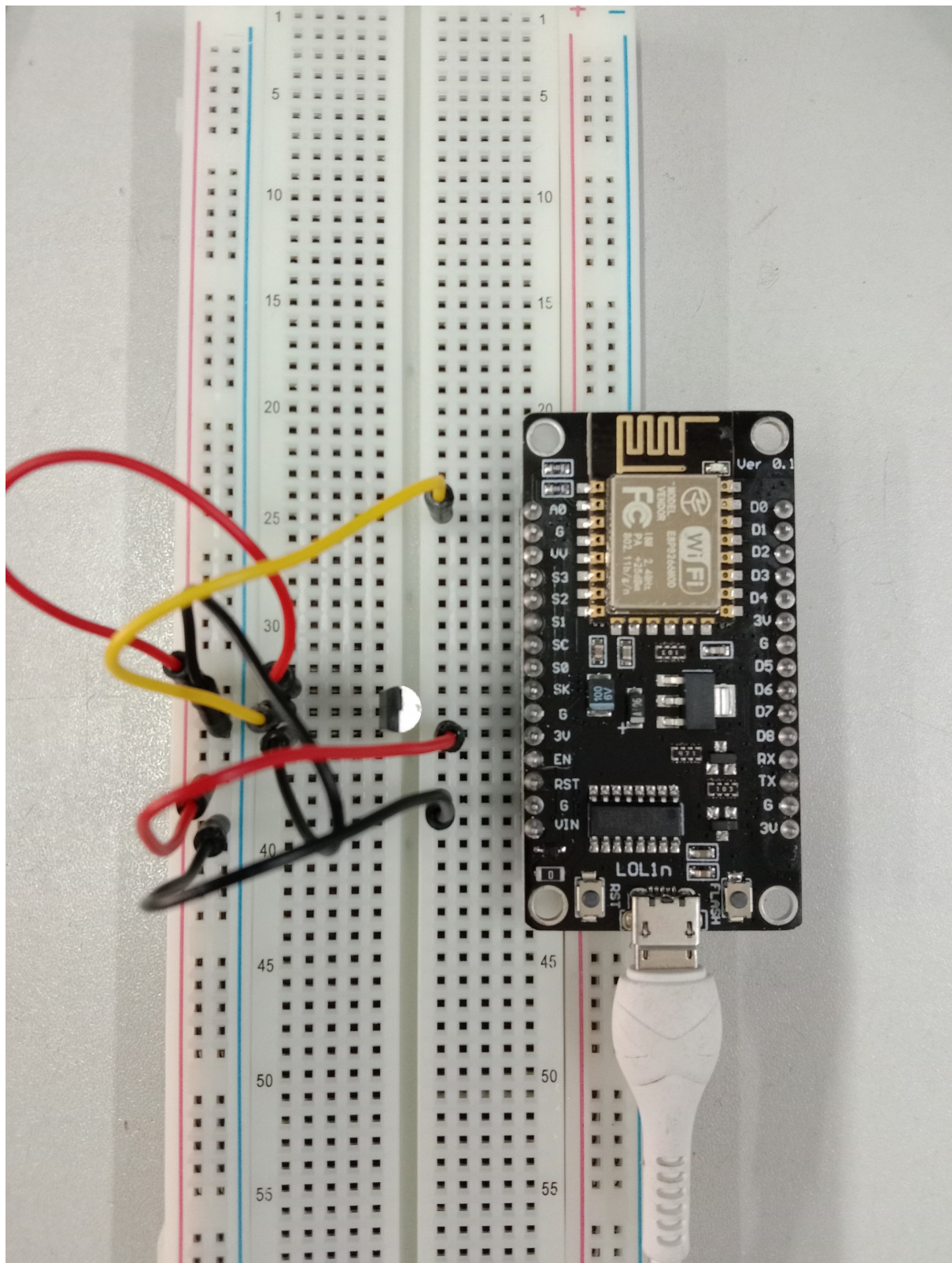
5.2.1 1. Tổng quan về bo mạch

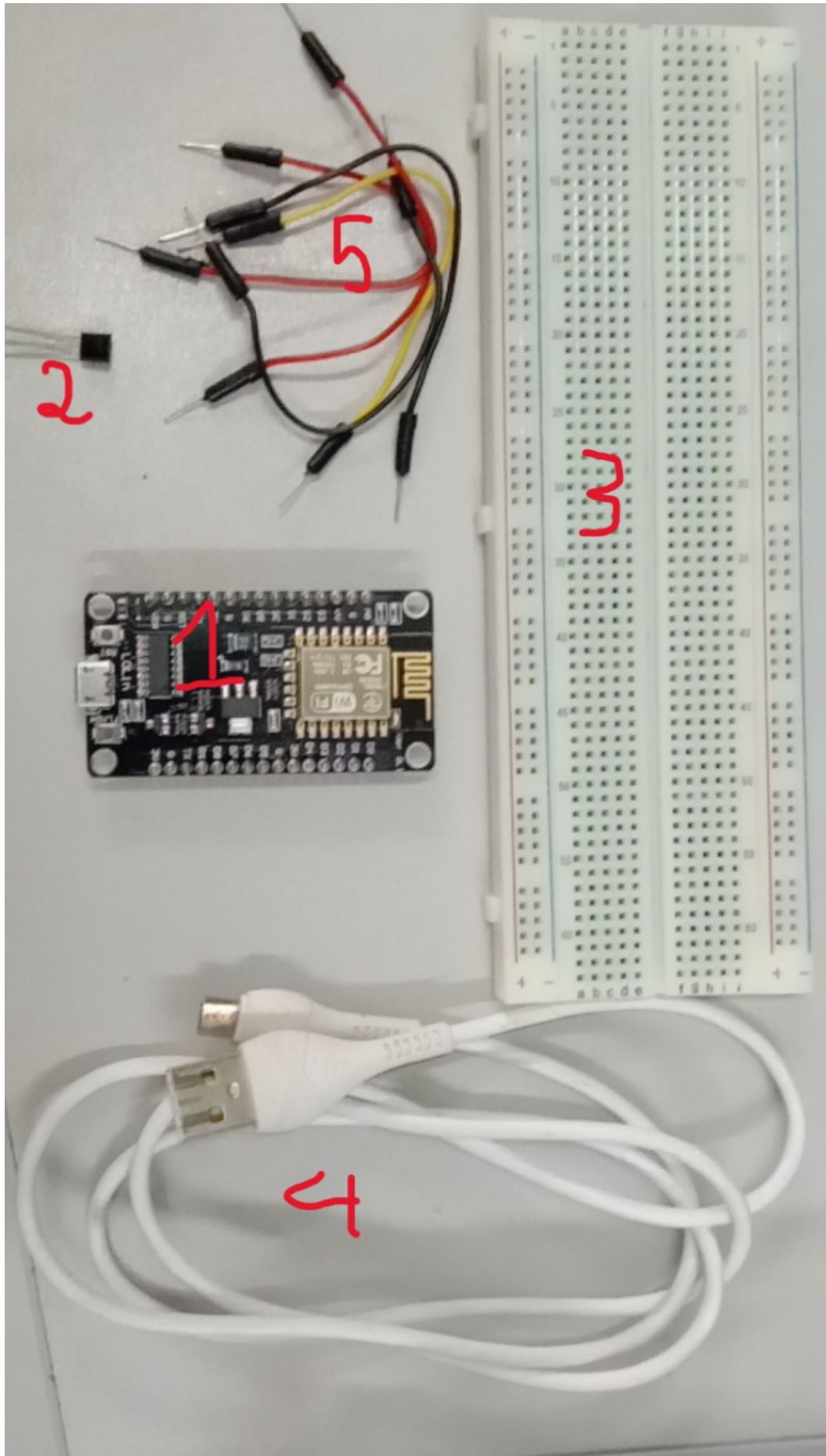
Bo mạch được xây dựng trên các thành phần sau: 1. **Module NodeMCU V3 CH340**. 2. Cảm biến **LM35**. 3. Breadboard. 4. Cáp micro USB. 5. Dây dẫn đực-đực.

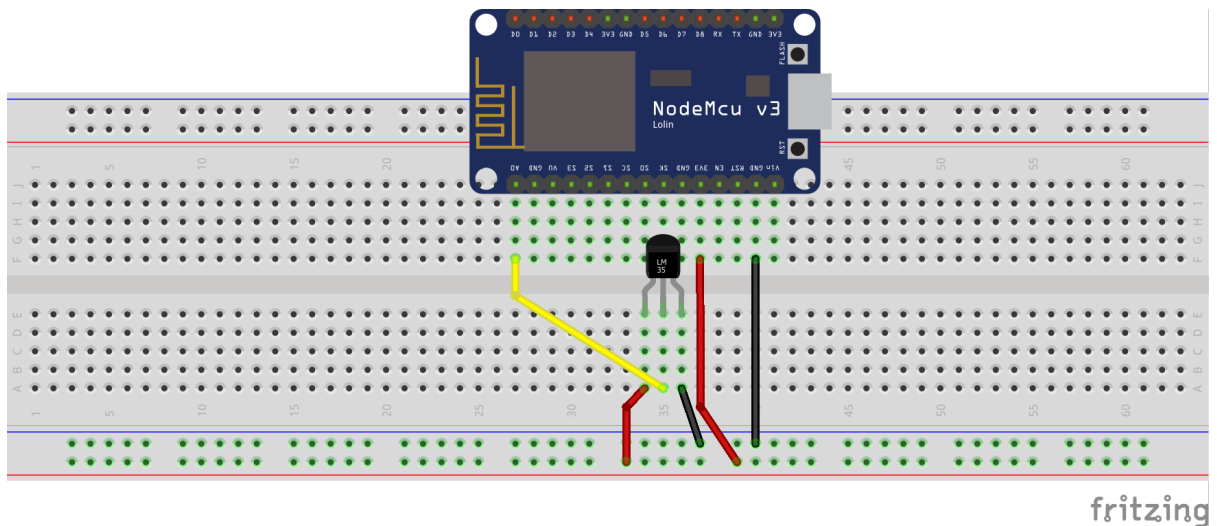
5.2.2 2. Kết nối bo mạch

Để kết nối bo mạch, tiến hành kết nối các thành phần theo các bước sau:

1. Kết nối module NodeMCU vào breadboard.
2. Kết nối cảm biến LM35 vào breadboard.
3. Kết nối chân nguồn và chân đất của cảm biến LM35.
4. Kết nối chân output của cảm biến LM35 vào chân **A0** của module ESP8266, cũng như chân nguồn vào chân **3V** và chân ground vào chân **G**.
5. Kết nối cáp micro USB với module và máy tính.







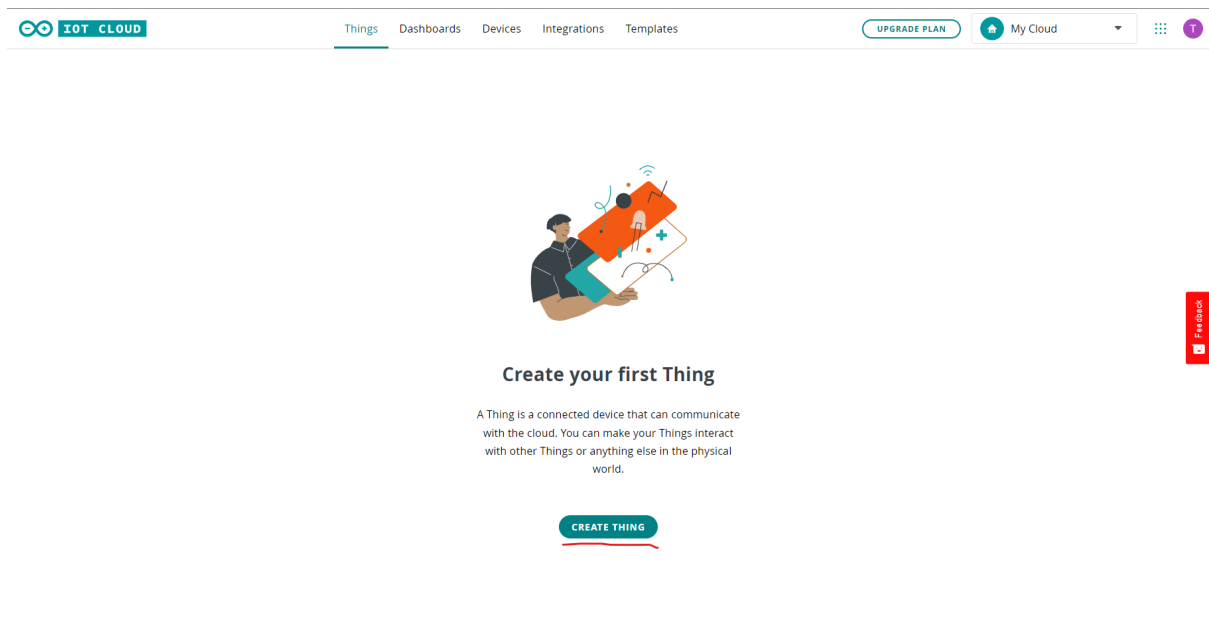
5.3 Thiết lập Arduino Cloud

Mục đích của việc thiết lập là tạo dữ liệu cloud cho module ESP8266 trên Arduino Cloud.

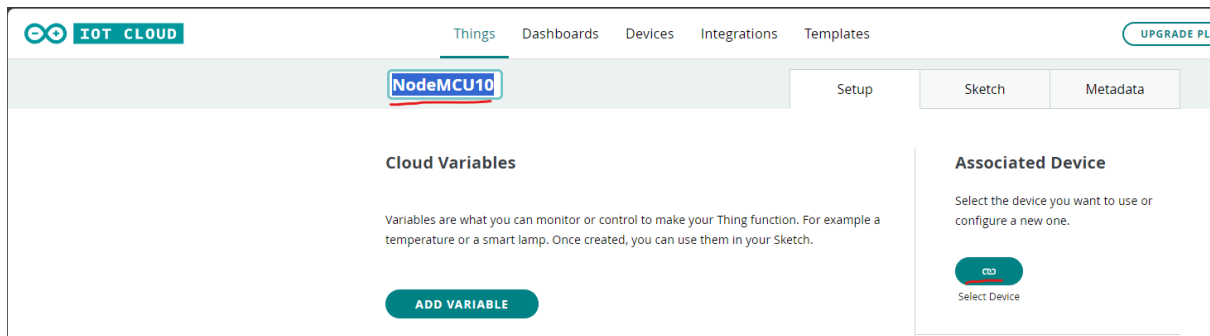
5.3.1 Tạo Thing

1. Đăng nhập vào Arduino Cloud.
2. Vào trang [IoT Cloud](#) của Arduino Cloud.

Sau đó, tạo dữ liệu của một thing.

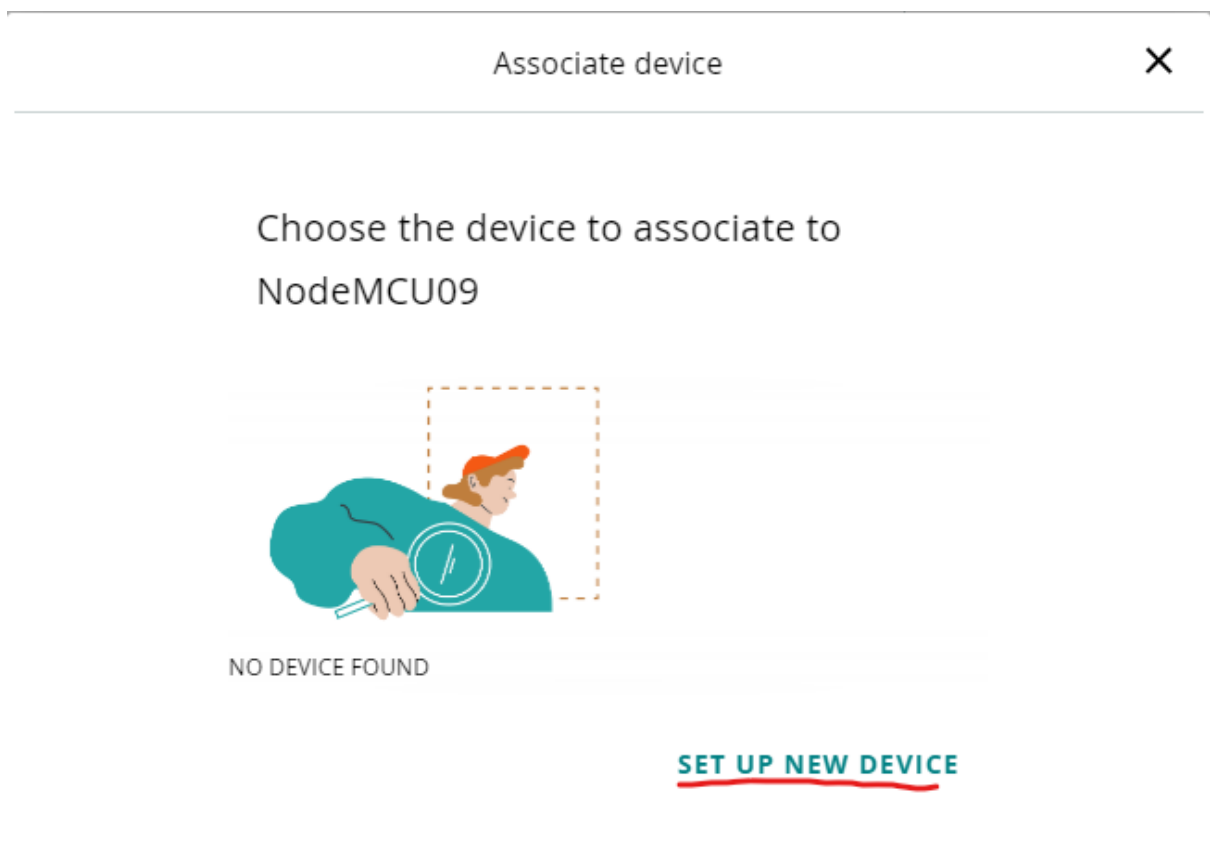


3. Đặt tên cho thing và chọn Select device.



5.3.2 Tạo device và network

1. Trong cửa sổ **Associate Device**, tạo dữ liệu của một device mới.



2. Trong cửa sổ **Setup Device**, chọn **Third party device**.

Sau đó, chọn loại vi điều khiển **ESP8266** và board **NodeMCU 1.0 (ESP-12E Module)**.

Tiếp theo, đặt tên cho device.

Cuối cùng, lưu các thông tin của device, chọn xác nhận đã lưu thông tin device, và chọn **CONTINUE**.

Tiếp theo, cấu hình network.

Đầu tiên, chọn button **Configure** trong phần network trong tab **Things** của IoT cloud.

Sau đó, điền thông tin **Wi-Fi Name**, **Password**, và **Secret Key**.


Setup Device

×

RECOMMENDED

AUTOMATIC

Auto-sketch generation, online editing, easy upload


Arduino board ⓘ
⌘ Arduino language (C++)




Third party device ⓘ
⌘ Arduino language (C++)

MANUAL

Manual programming and upload, offline editing


DIY

Any Device ⓘ
⌘ Python, MicroPython, JavaScript (NodeJS)

 Setup Device 

Select device type

Please select the device type and model you want to configure

☒ ESP8266

☐ ESP32

☐ LoRaWAN

NodeMCU 1.0 (ESP-12E Module)

▼

CONTINUE

5.3.3 Tạo Variable

Trong dữ liệu của một thing, tạo mới dữ liệu của một **Variable**.

Sau đó, đặt tên (**Name**), chọn kiểu dữ liệu (**Floating Point Number**), cấu hình **Variable Permission**, và chọn **ADD VARIABLE**.


5.4 Thiết lập và lập trình trên Arduino IDE 2.x

5.4.1 1. Tiến hành đồng bộ dữ liệu sketch với Arduino Cloud.

Trong tab **SKETCHBOOK**, chọn tab **Cloud Sketchbook**.

Sau đó, tiến hành đồng bộ với cloud thông qua nút **Sync** (nút đánh dấu số 1) và pull dữ liệu sketch thông qua nút **Pull sketch** (nút được khoanh tròn như hình bên dưới).

←Setup Device×



Give your device a name

Name your device so you will be able to recognize it.


Device Name

NodeMCU10-Device

↻

NEXT

Setup Device ✕



Make your device IoT-ready

To use this board you will need a Device ID and a Secret Key, please copy and save them or [download the PDF](#).


Also, keep in mind that this device authentication has a lower security level compared to other Arduino devices.

Device ID

2e74266d-f16f-4fe1-b998-02740e513c76

Secret Key

UVRACKGB3Y4WFBVRV80AC

**Secret key cannot be recovered**

Please keep it safe, if you lose it you will have to delete and setup your device again.

☒ I saved my device ID and Secret Key [CONTINUE](#)

5.4. Thiết lập và lập trình trên Arduino IDE 2.x

45

Network

Enter your network credentials to connect your device.



Configure

Configure network



You will find these network parameters in the secret tab in your sketch, and your device will be able to connect to the network once the sketch will be uploaded.

Wi-Fi Name *

Password



Secret Key *



SAVE

Cloud Variables

Variables are what you can monitor or control to make your Thing function. For example a temperature or a smart lamp. Once created, you can use them in your Sketch.

ADD VARIABLE



5.4.2 2. Thiết lập board và port.

1. Trong cửa sổ của IDE, chọn phần lựa chọn board và chọn **Select other board and port....**
2. Sau đó, chọn board **NodeMCU 1.0 (ESP-12E Module)** và port kết nối tương ứng.
3. Thêm các đoạn code tính nhiệt độ.

```
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/2abe9f1e-53be-444a-98c4-
→066f070ac92e

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when
→changes are made to the Thing

float temperature;

Variables which are marked as READ/WRITE in the Cloud Thing will also
→have functions
which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of
→this sketch.
*/

#include "thingProperties.h"
+ int sensorPin = A0;
void setup() {
// Initialize serial and wait for port to open:
Serial.begin(9600);
// This delay gives the chance to wait for a Serial Monitor without
```

(continues on next page)

Add variable
×

Name
Temperature

↻ Sync with other Things i

Floating Point Number eg. 1.55

Declaration
float temperature ; i

Variable Permission i

☐ Read & Write
☒ Read Only

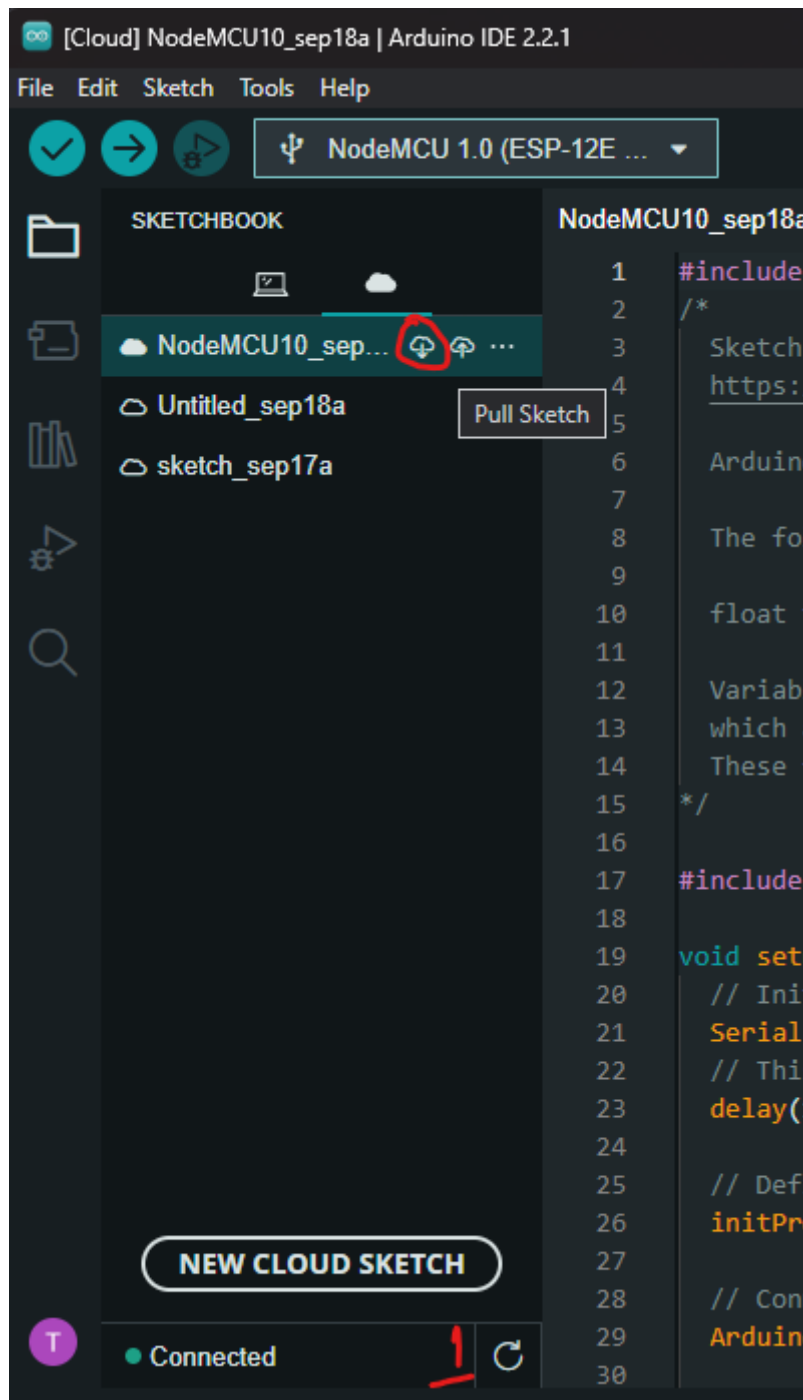
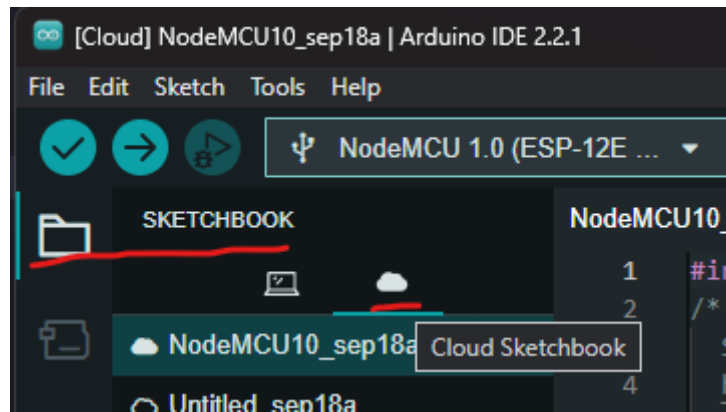
Variable Update Policy i

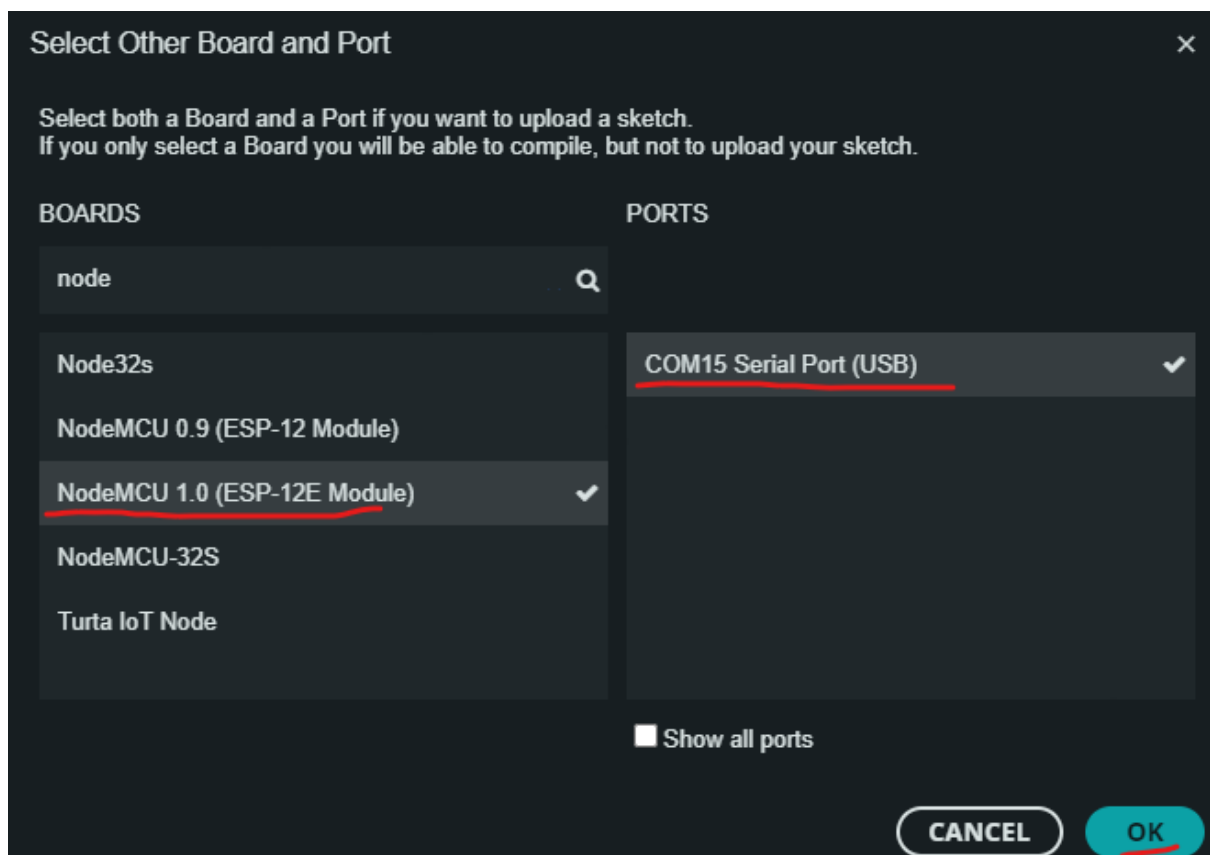
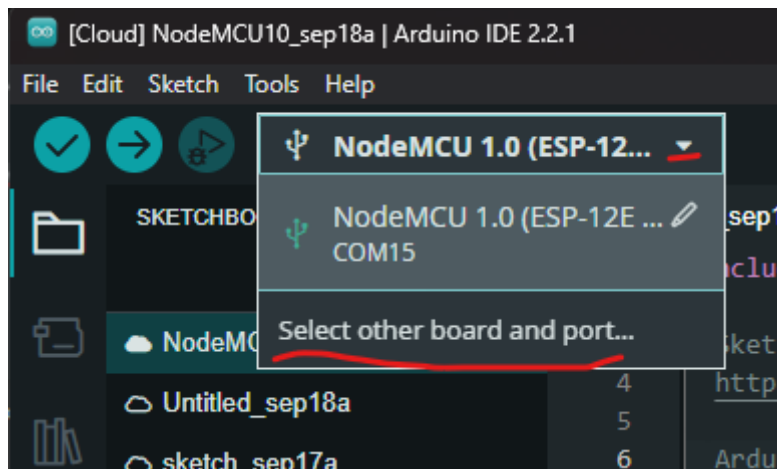
☒ On change
☐ Periodically

Threshold
0

ADD VARIABLE

CANCEL





(continued from previous page)

```
→blocking if none is found
delay(1500);

// Defined in thingProperties.h
initProperties();

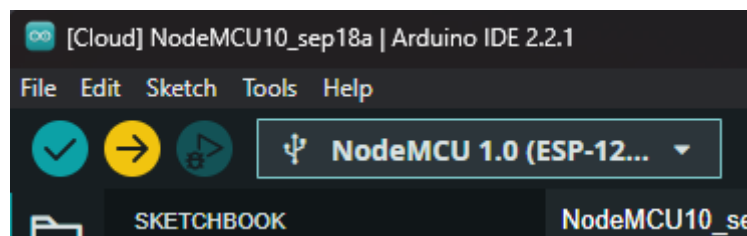
// Connect to Arduino IoT Cloud
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/*
  The following function allows you to obtain more information
  related to the state of network and IoT Cloud connection and errors
  the higher number the more granular information you'll get.
  The default is 0 (only errors).
  Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();
}

void loop() {
  ArduinoCloud.update();
  // Your code here
  + int reading = analogRead(sensorPin);
  + float voltage = reading * 3.3 / 1024.0;
  + temperature = voltage * 100.0;

  + Serial.println(temperature);
  + delay(1000);
}
```

4. Click nút **Upload**.



Hình1: Upload button.

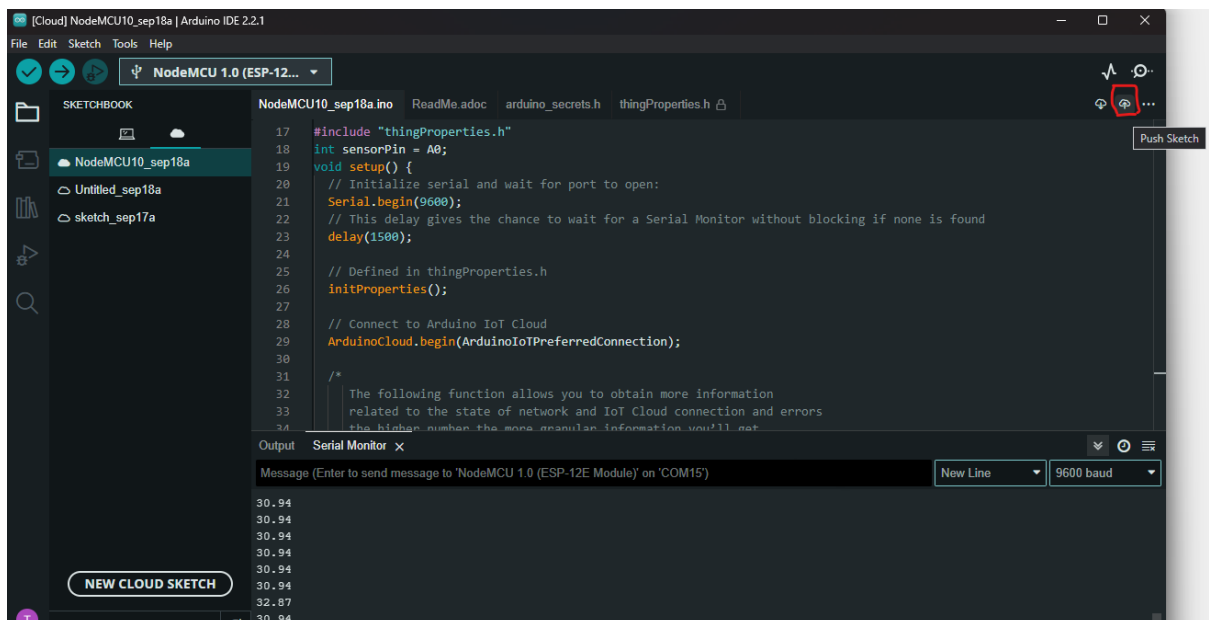
5. Push dữ liệu sketch lên cloud.

Trên IDE, chọn nút **Push sketch**.

```
Output
Writing at 0x0003c000... (84 %)
Writing at 0x00040000... (89 %)
Writing at 0x00044000... (94 %)
Writing at 0x00048000... (100 %)
Wrote 421328 bytes (309269 compressed) at 0x00000000 in 27.3 seconds (effective 123.6 kbit/s)...
Hash of data verified.

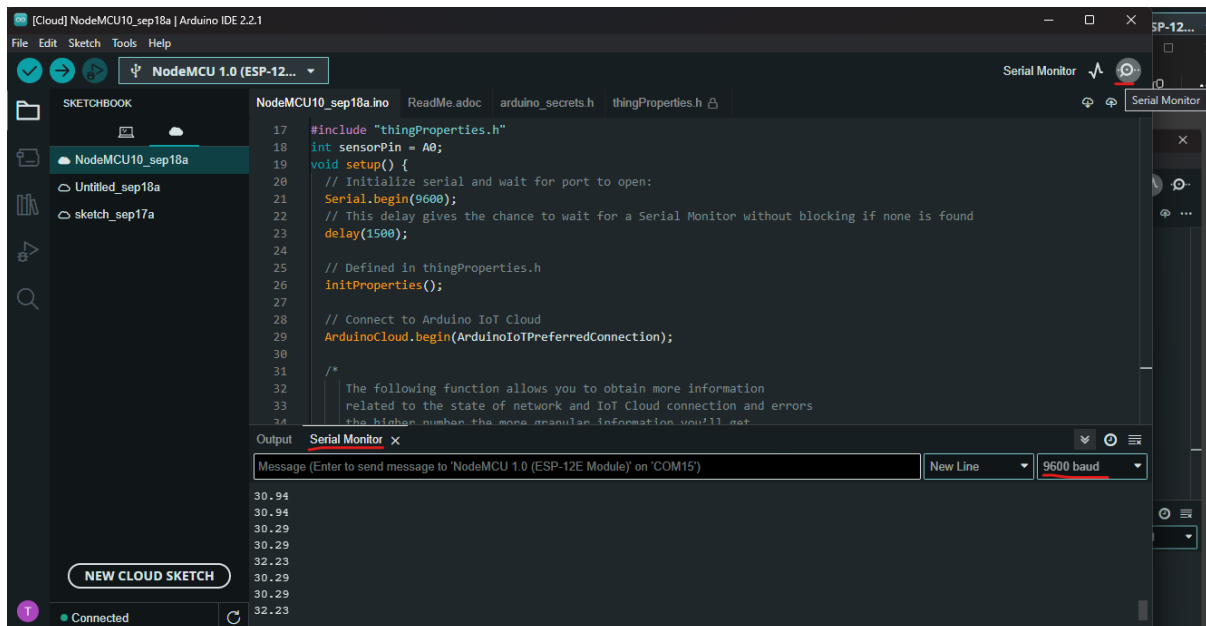
Leaving...
Hard resetting via RTS pin...
```

Hình2: Thông báo upload thành công.



5.4.3 Quan sát kết quả trên IDE

Trên giao diện IDE, chọn nút **Serial Monitor** và quan sát giá trị nhiệt độ đo được trên board.

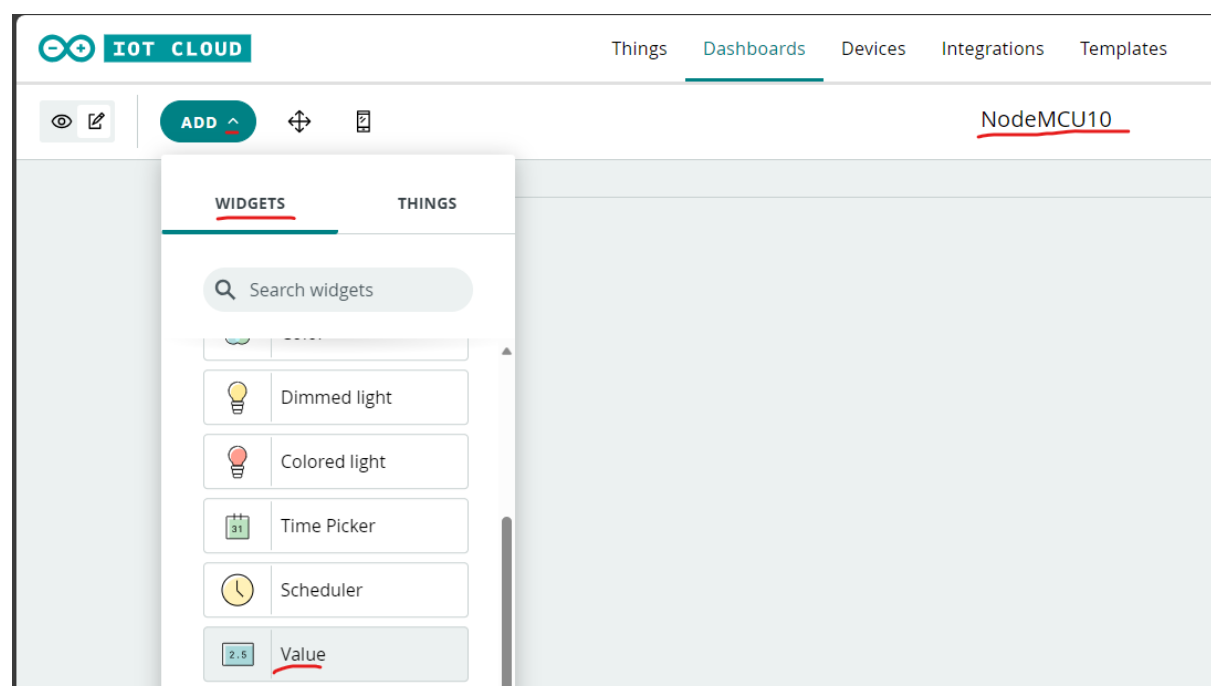
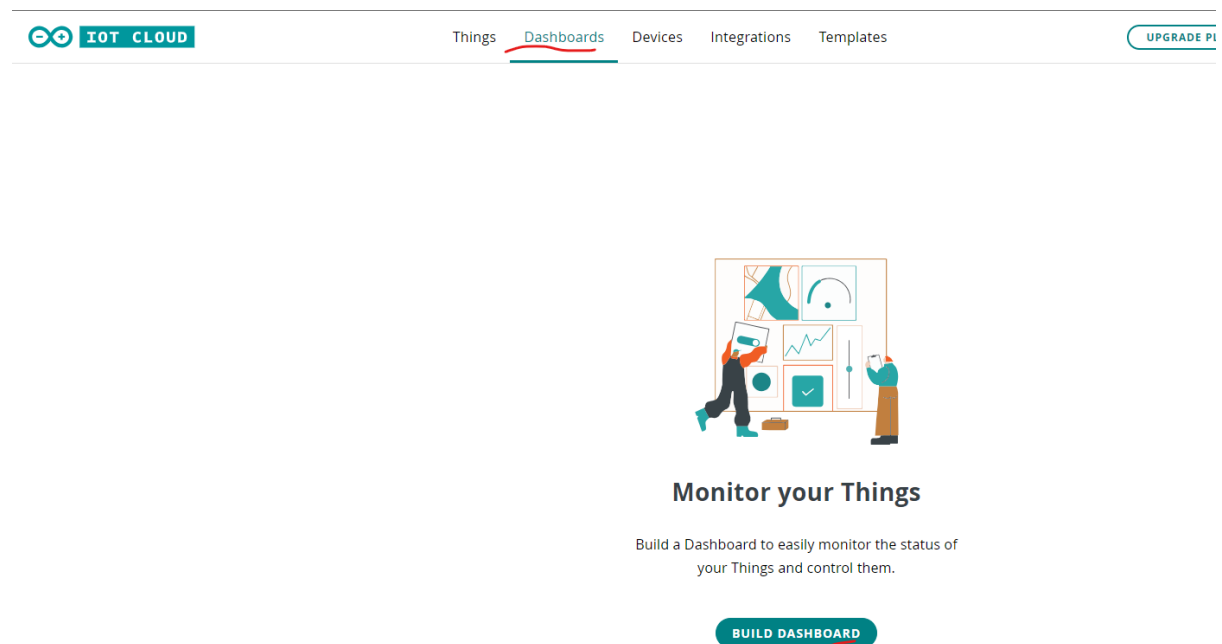


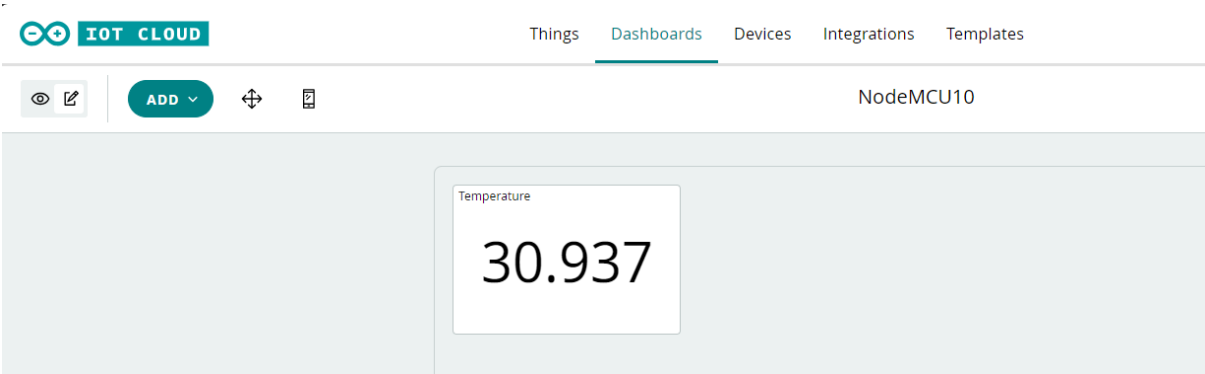
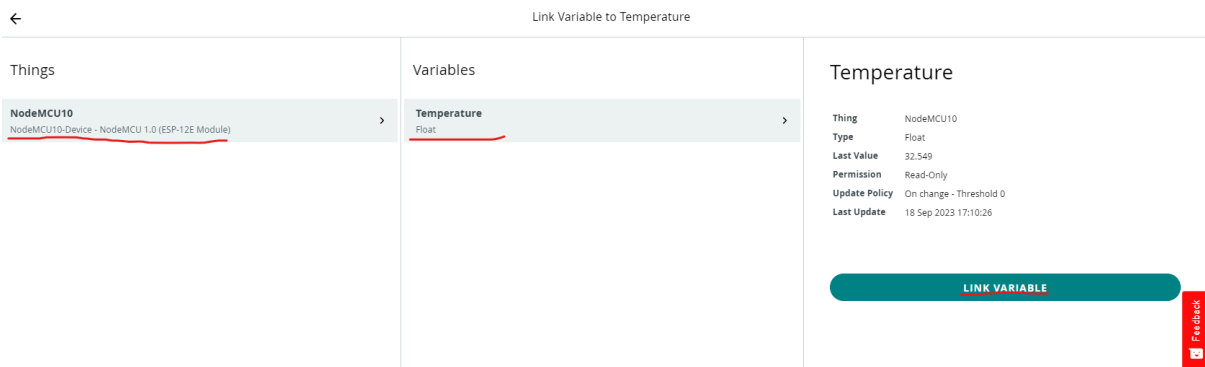
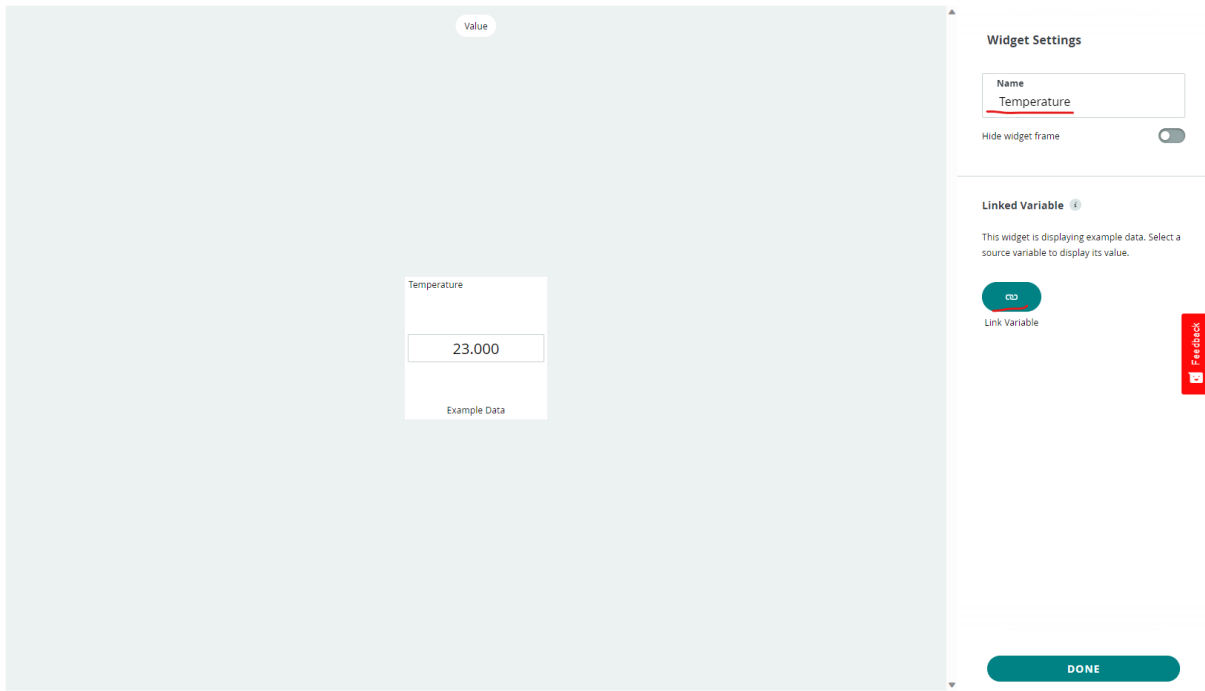
5.4.4 Quan sát kết quả trên IoT Dashboard

1. Tạo dữ liệu một Dashboard.

Trong giao diện web của **IoT Cloud**, chọn tab **Dashboards**, và tạo mới dữ liệu của một **Dashboard**.

2. Đặt tên cho dashboard và tạo **Value** widget.
3. Đặt tên cho widget và chọn **Link Variable**.
4. Liên kết với variable đã tạo và chọn **LINK VARIABLE**.
5. Quan sát giá trị nhiệt độ đo được trên dashboard.





Cài đặt nhận thông báo trên Arduino Cloud

6.1 Tổng quan về nội dung thực hiện

Nội dung thực hiện sẽ trình bày cách cài đặt nhận thông báo về trạng thái led trên module NodeMCU V3 CH340.

6.2 Môi trường làm việc

Để cài đặt nhận thông báo trên Arduino Cloud, cần có môi trường làm việc như đã đề cập ở phần *tổng quan* và tài khoản trên nền tảng IFTTT.

Ngoài ra, các bước thực hiện bên dưới, sẽ dựa trên hoạt động đã thực hiện ở phần *thực hiện hệ thống*.

6.3 Phần 1: Các bước thực hiện để cấu hình điều khiển led thông qua cloud

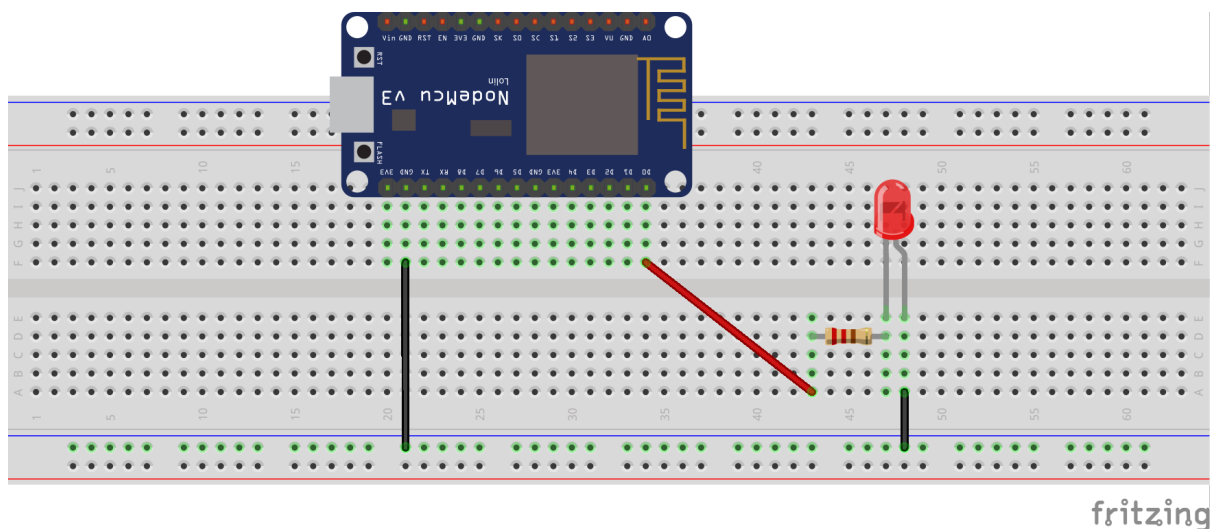
6.3.1 1. Thiết lập bo mạch

Chuẩn bị các linh kiện như phần *thực hiện hệ thống*, ngoại trừ cảm biến LM35. Sau đó, chuẩn bị thêm led đơn và điện trở 220 ohm.

Tiếp theo, tiến hành nối các thành phần (minh họa như hình bên dưới) dựa trên breadboard.

1. Cắm NodeMCU ESP8266 lên breadboard.

2. Cắm Led đơn và điện trở.
3. Nối dây GND vào chân âm của led và một đầu của điện trở vào chân **D0** của module.



6.3.2 2. Cài đặt trên Arduino Cloud

Đầu tiên, dựa trên thing đã tạo ở phần *thực hiện hệ thống*, tạo thêm một variable để cập nhật trạng thái led.

6.3.3 3. Lập trình trên Arduino IDE

Đầu tiên, tiến hành pull sketch để cập nhật thay đổi trên cloud.

Sau đó, dựa trên callback **onLedStateChange**, lập trình các đoạn code điều khiển led thông qua cloud.

```
#include "arduino_secrets.h"
/*
Sketch generated by the Arduino IoT Cloud Thing "Untitled"
https://create.arduino.cc/cloud/things/f9b143ca-27f9-4bde-8e17-
→b2664c5f61dd
```

Arduino IoT Cloud Variables description

The following variables are automatically generated and updated when
→changes are made to the Thing

```
float temperature;
bool ledState;
```

Variables which are marked as READ/WRITE in the Cloud Thing will also
→have functions

(continues on next page)

Variable Info



ledState

Declaration

`bool ledState`

Type

Boolean

Variable Permission

Read & Write

Update Policy

On change

ID

41a93432-e9b6-4bca-9a5e-837d5bf85566



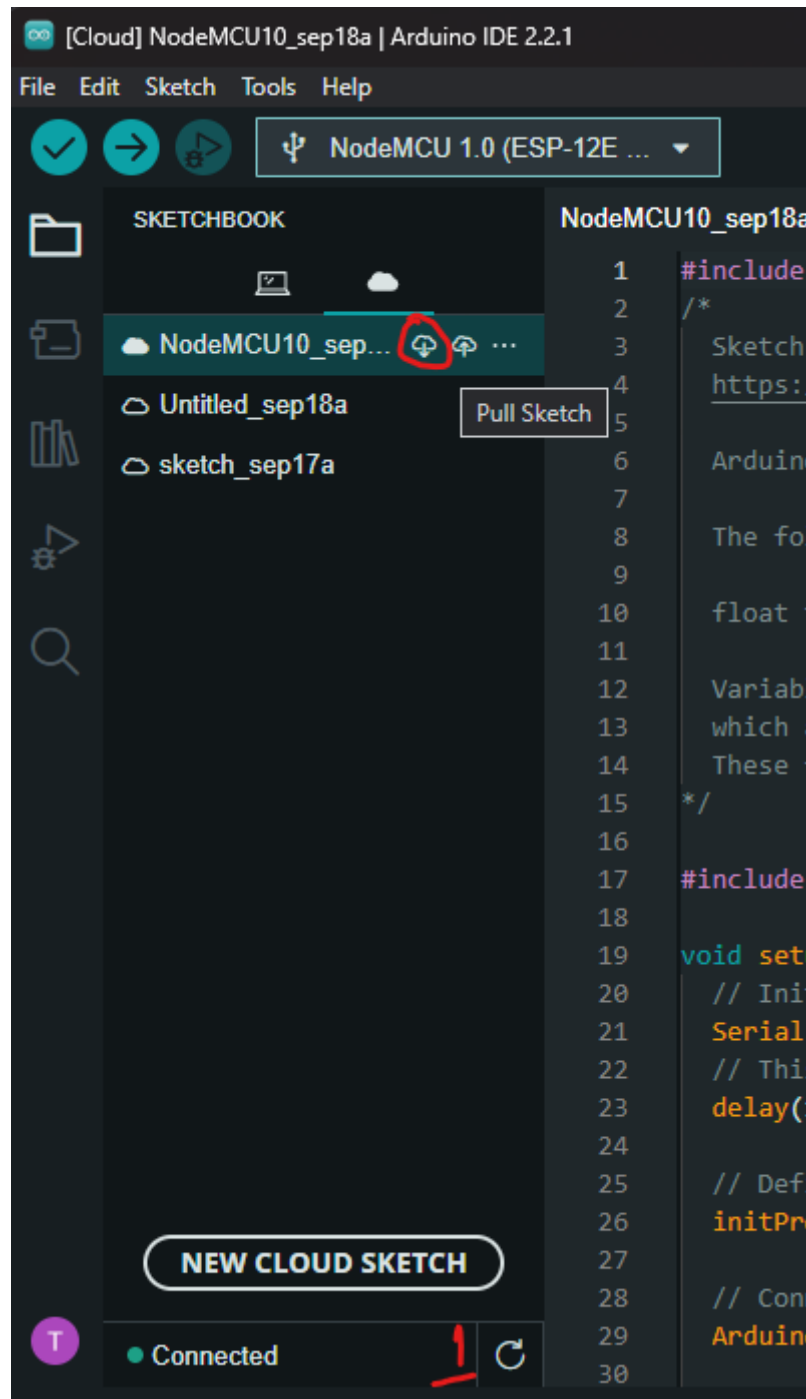
Last Value

false

Last Update

02 Oct 2023 16:43:29

[EDIT VARIABLE](#)



(continued from previous page)

```

which are called when their values are changed from the Dashboard.
These functions are generated with the Thing and added at the end of
→ this sketch.
*/

#include "thingProperties.h"
int sensorPin = A0;
+ int ledPin = 16; // GPIO16 (D0)
void setup() {
// Initialize serial and wait for port to open:
Serial.begin(9600);
// This delay gives the chance to wait for a Serial Monitor without
→ blocking if none is found
delay(1500);

// Defined in thingProperties.h
initProperties();

// Connect to Arduino IoT Cloud
ArduinoCloud.begin(ArduinoIoTPreferredConnection);

/*
    The following function allows you to obtain more information
    related to the state of network and IoT Cloud connection and errors
    the higher number the more granular information you'll get.
    The default is 0 (only errors).
    Maximum is 4
*/
setDebugMessageLevel(2);
ArduinoCloud.printDebugInfo();

+ pinMode(ledPin, OUTPUT);
}

void loop() {
ArduinoCloud.update();
// Your code here
int reading = analogRead(sensorPin);
float voltage = reading * 3.3 / 1024.0;
temperature = voltage * 100.0;

Serial.println(temperature);
delay(1000);
}

```

(continues on next page)

6.3. Phần 1: Các bước thực hiện để cấu hình điều khiển led thông qua cloud 61

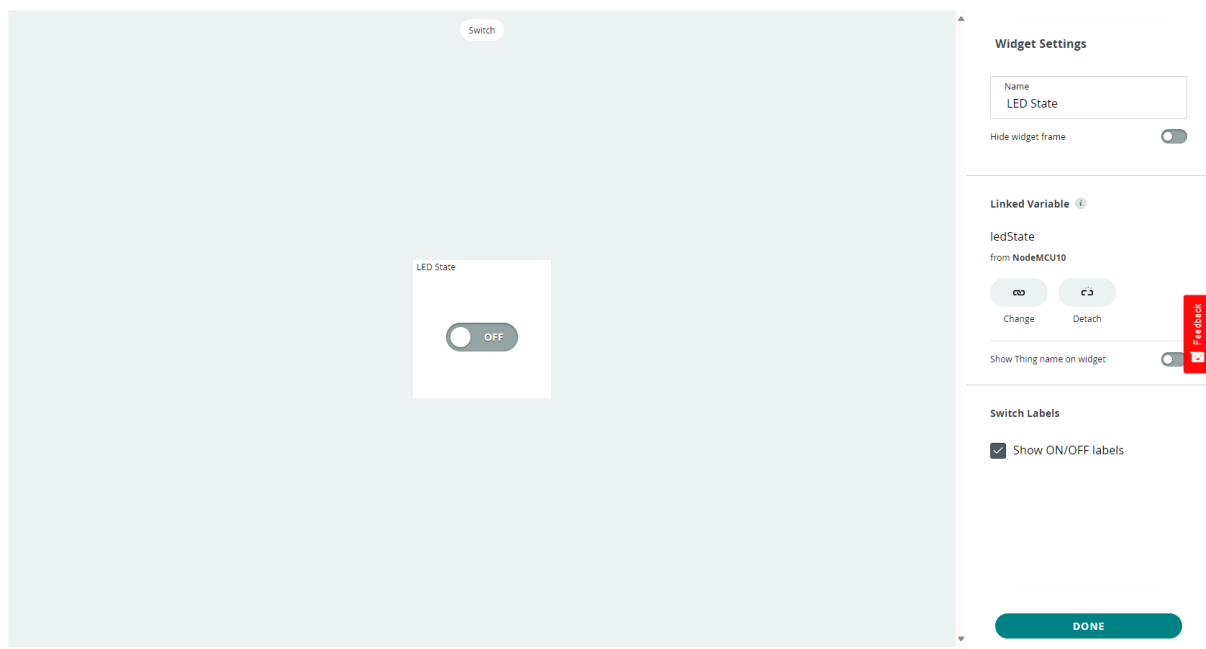
(continued from previous page)

```

/*
Since LedState is READ_WRITE variable, onLedStateChange() is
executed every time a new value is received from IoT Cloud.
*/
void onLedStateChange() {
// Add your code here to act upon LedState change
+ if(ledState)
+   digitalWrite(ledPin, HIGH);
+ else
+   digitalWrite(ledPin, LOW);
+ Serial.printf("\r\n Led State is %d", digitalRead(ledPin));
}

```

Sau đó, trên phần Dashboards của Arduino Cloud, tạo một switch widget và link với ledState variable đã tạo.



Lúc này, việc thiết lập điều khiển trên cloud đã hoàn thành và có thể bật/tắt led thông qua widget trên Dashboard.

6.4 Phần 2: Các bước thực hiện để cấu hình gửi thông báo

6.4.1 1. Cấu hình trên IFTTT

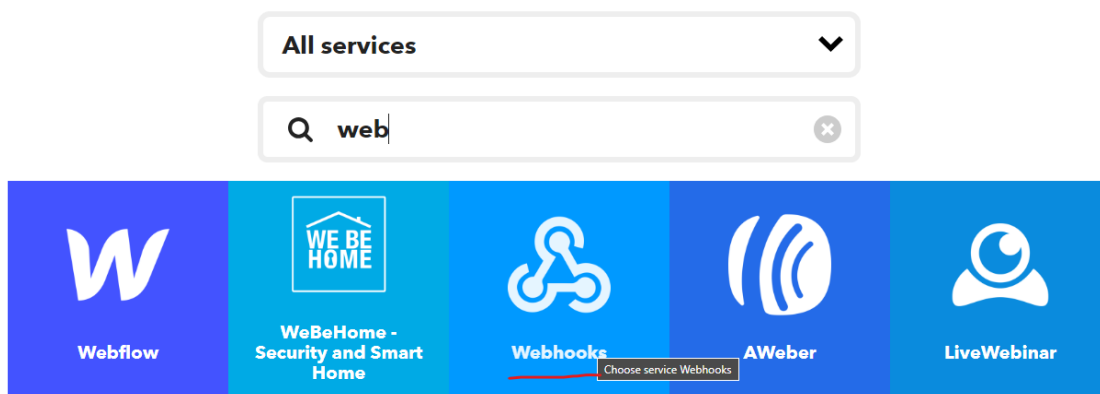
Đầu tiên, đăng nhập vào nền tảng IFTTT và chọn **Create**.

Trong phần tiếp theo, trong phần **If This**, chọn **Add**.



Trong phần chọn service, chọn **Webhooks**.

Choose a service



Trong phần **Choose a trigger**, chọn **Receive a web request**. Sau đó, chọn **Connect**.

Trong phần **Complete trigger fields**, đặt **Event Name** và chọn **Create trigger**.

Tiếp theo, quay về màn hình cũ, trong phần **Then That**, chọn **Add**.

Tiếp theo, trong phần **Choose a service**, chọn **Gmail**.

Sau đó, trong phần **Choose an action**, chọn **Send yourself an email**. Sau đó, chọn **Connect**.

Cuối cùng, trong phần **Complete action fields**, chọn **Create action**.

Choose a trigger



Webhooks

Receive a web request with a JSON payload

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap

Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)



Suggest a new trigger

Complete trigger fields



Receive a web request

This trigger fires every time the Maker service receives a web request to notify it of an event. For information on triggering events, go to your Maker service settings and then the listed URL (web) or tap your username (mobile)

Event Name



The name of the event, like "button_pressed" or "front_door_opened". Use only letters, numbers, and underscores

Create trigger



Choose a service

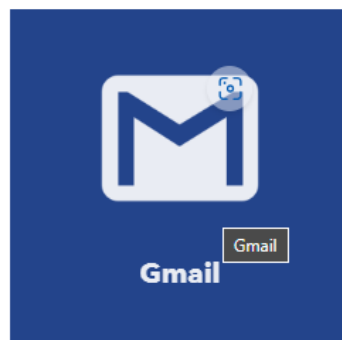
All services

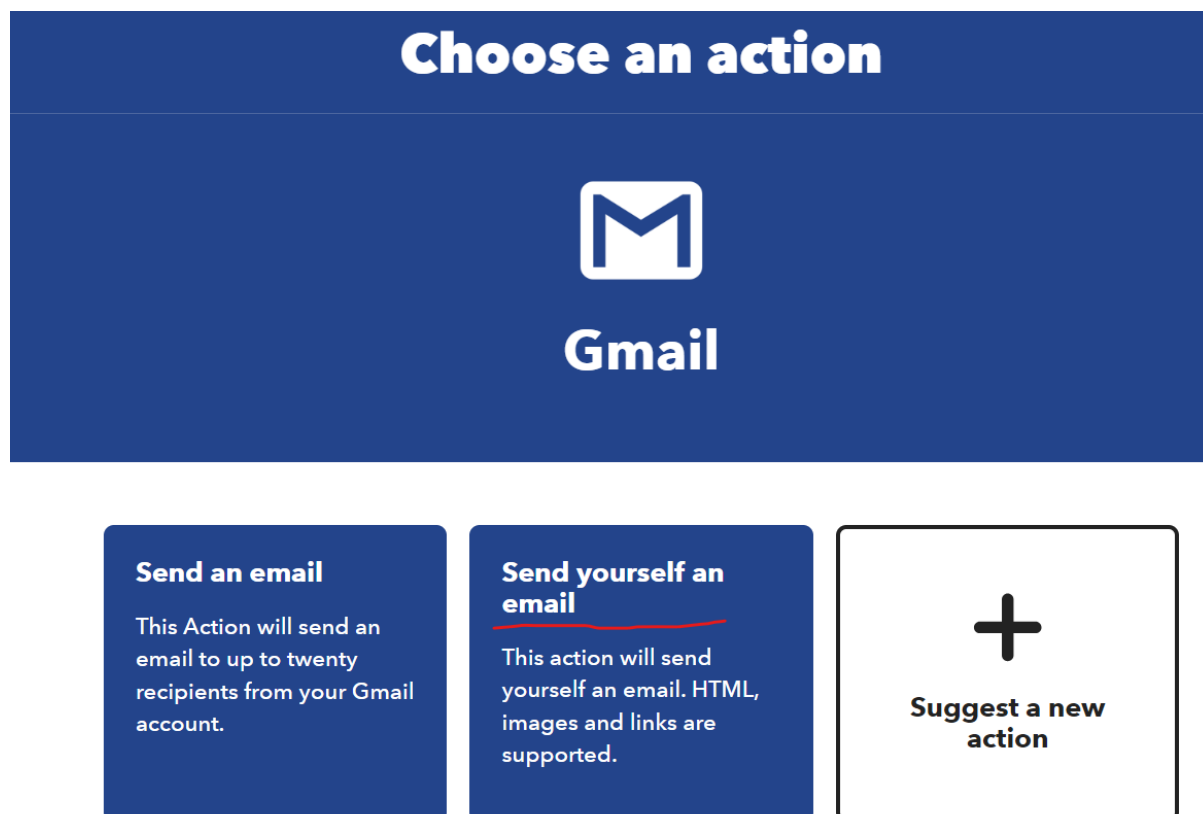
▼

Q

gmail

×





Chọn **Continue** và **Finish** khi quay về màn hình cũ.

Trong màn hình tiếp theo, chọn icon **Webhooks Receive a web request**.

Trong phần **Webhooks integrations**, chọn **Documentation**.

Cuối cùng, trong phần **To trigger an Event with 3 JSON values**, nhập event name, bấm thử **Test it**, và copy phần URL được tạo ra.

6.4.2 2. Cấu hình trên Arduino Cloud

Trên Arduino Cloud, chọn vào 1 thing, sau đó chọn Set webhook.

Sau đó, trong cửa sổ **Set webhook**, nhập URL đã tạo ra từ IFTTT và chọn **Set webhook**.

Cuối cùng, quan sát các thông báo từ email thông qua nền tảng IFTTT.

Subject

The event named " **EventName** " occurred on the Maker Webhooks service

Add ingredient

Body

What: **EventName**

When: **OccurredAt**

Extra Data: **Value1** , **Value2** ,
Value3 ,

Some HTML ok

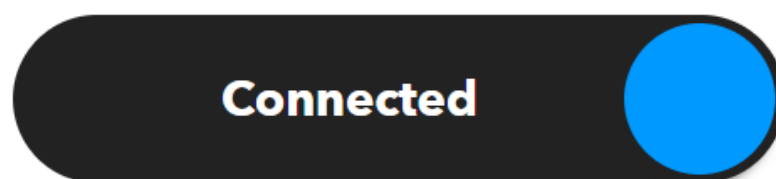
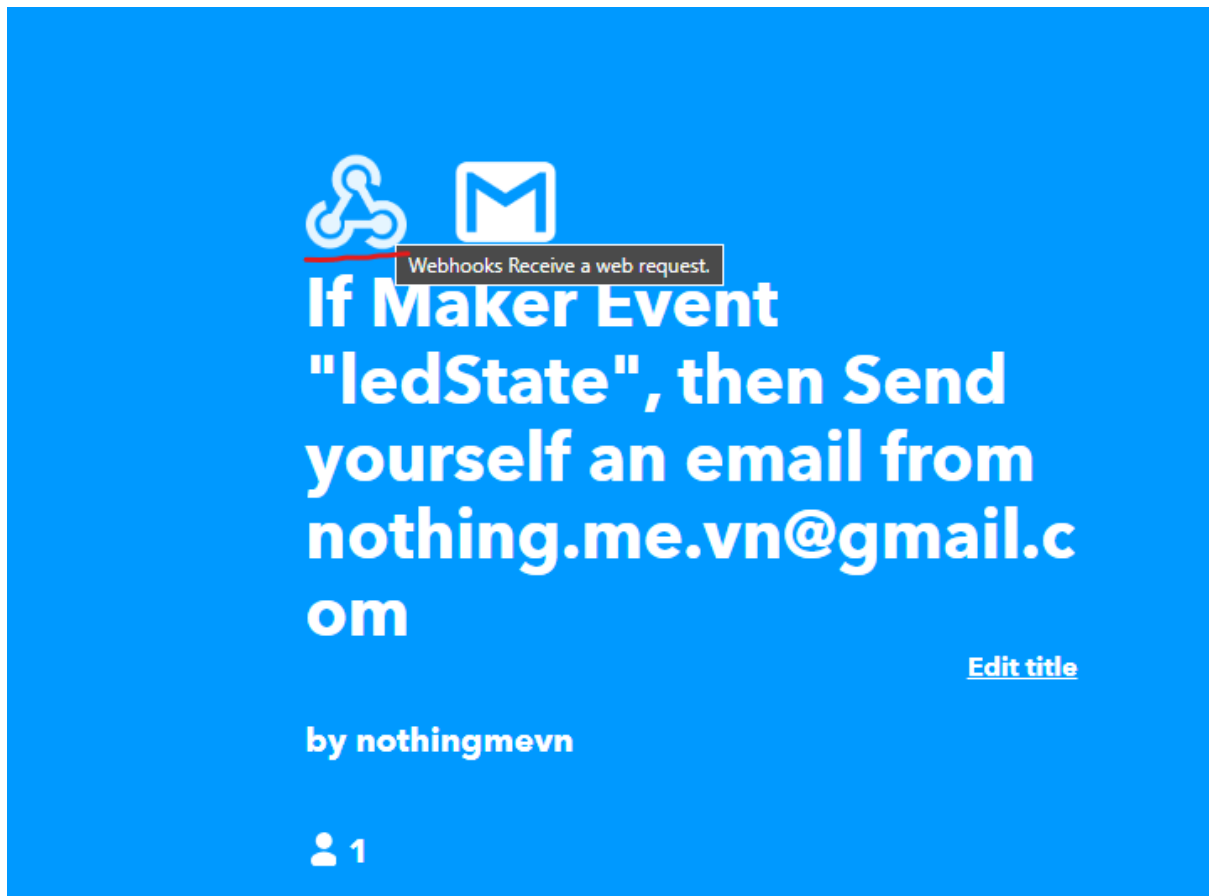
Add ingredient

Attachment URL

URL to include as an attachment

Add ingredient

Create action



More details



Webhooks integrations

Learn to use webhooks and integrate other services on IFTTT with your DIY projects. You can create Applets that work with any device or app that can make or receive a web request. If you'd like to build your own service and Applets, [check out the IFTTT Developer Dashboard](#). Connect a webhook to any of the 750+ apps and services on IFTTT.

[Create](#)[Documentation](#)

To trigger an Event with 3 JSON values

Make a POST or GET web request to:

```
https://maker.ifttt.com/trigger/ledState/with/key/
```

With an optional JSON body of:

```
{ "value1" : "", "value2" : "", "value3" : "" }
```

The data is completely optional, and you can also pass `value1`, `value2`, and `value3` as query parameters or form variables. This content will be passed on to the action in your Applet.

You can also try it with `curl` from a command line.

```
curl -X POST https://maker.ifttt.com/trigger/ledState/with/key/
```

Please read [our FAQ](#) on using Webhooks for more info.

[Test it](#)

[Things](#)
[Dashboards](#)
[Devices](#)
[Integrations](#)
[Templates](#)

UPGRADE PLAN

NodeMCU10

Setup

Sketch 5

Metadata

Cloud Variables

ADD

Name ↓	Last Value	Last Update
<input type="checkbox"/> ledState <small>bool ledState;</small>	true	02 Oct 2023 19:18:52
<input type="checkbox"/> Temperature <small>float temperature;</small>	2.9	02 Oct 2023 20:01:58

Associated Device

NodeMCU10-Device

ID: e34d1e0d-e912-4140-8ffc-1...

Type: NodeMCU 1.0 (ESP-12E Module)

Status: Online

Change

Detach

Network

Wi-Fi Name: FETEL@...

Password:

Secret Key:

Change

Set webhook

Set webhook

X

webhooks allow you to send and receive automated messages to other services. If you're new to webhooks, check this [sample project](#).

Enter a valid webhook URL for your Thing, you can generate it from one of these available domains:

script.google.com

hooks.zapier.com

cloudfunctions.net

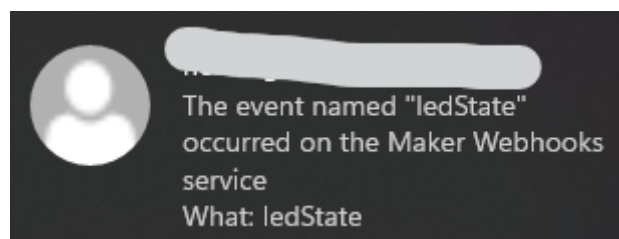
in.treasuredata.com

webhook.site

Webhook URL

<https://maker.ifttt.com/trigger/ledState/w>

SET WEBHOOK



Tài liệu tham khảo

- [AmazonIoT] What is IoT (Internet of Things)? - Everything you need to know - AWS. (n.d.). Amazon Web Services, Inc. <https://aws.amazon.com/what-is/iot/>
- [MacrometaIoT] IoT Infrastructure Guide. (n.d.). Macrometa. <https://www.macrometa.com/iot-infrastructure>
- [Karofi] Internet vạn vật (IoT) - Mạng lưới thiết bị kết nối Internet. (2022, November 6). Karofi. <https://karofi.com/internet-van-vat-iot-mang-luoi-thiet-bi-ket-noi-internet-bv3037.html>
- [Softwareag] What is an IoT Platform? | IoT Platform Advantages & Capabilities. (n.d.). Software AG. https://www.softwareag.com/en_corporate/resources/iot/article/iot-platform.html