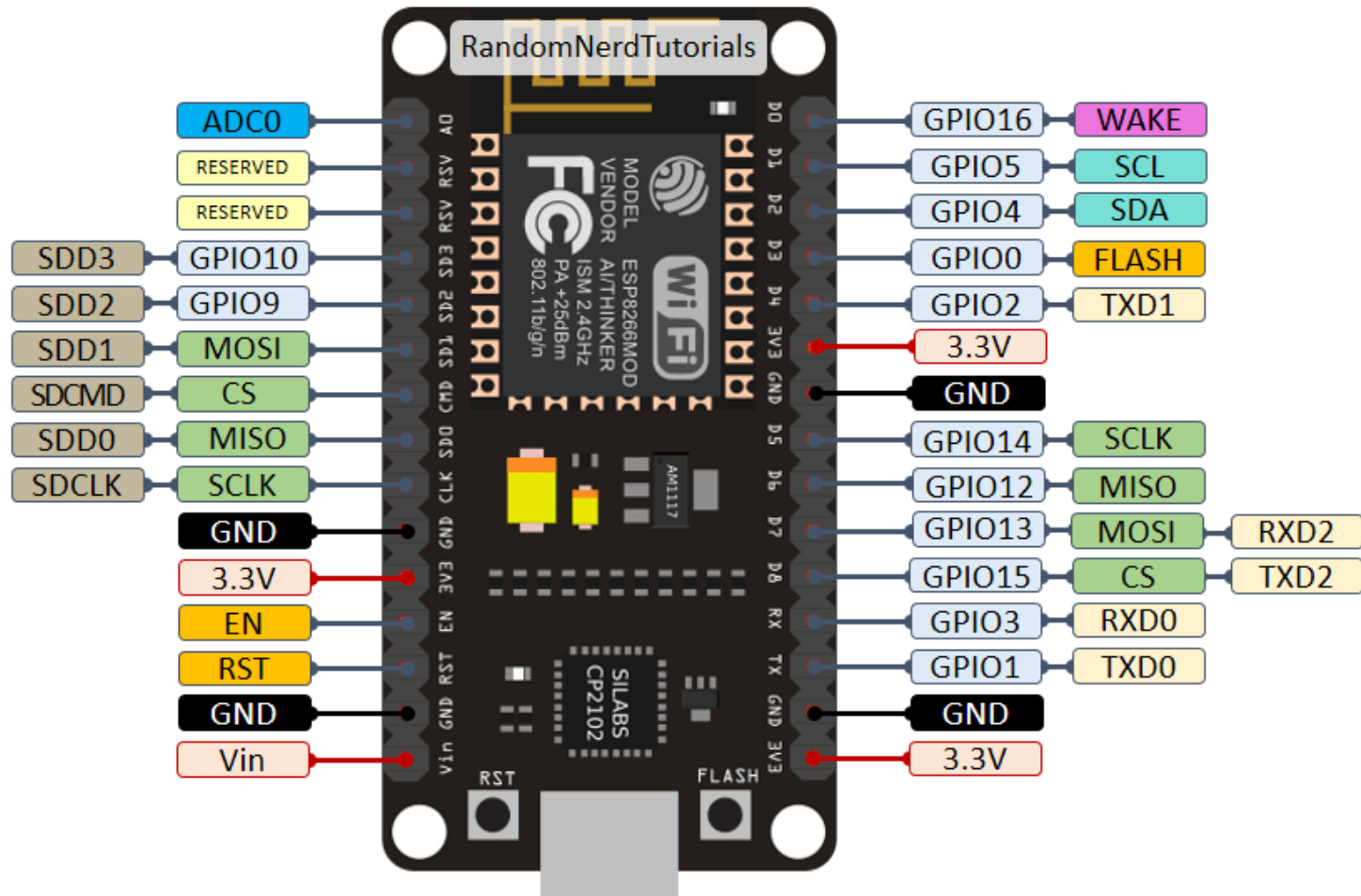


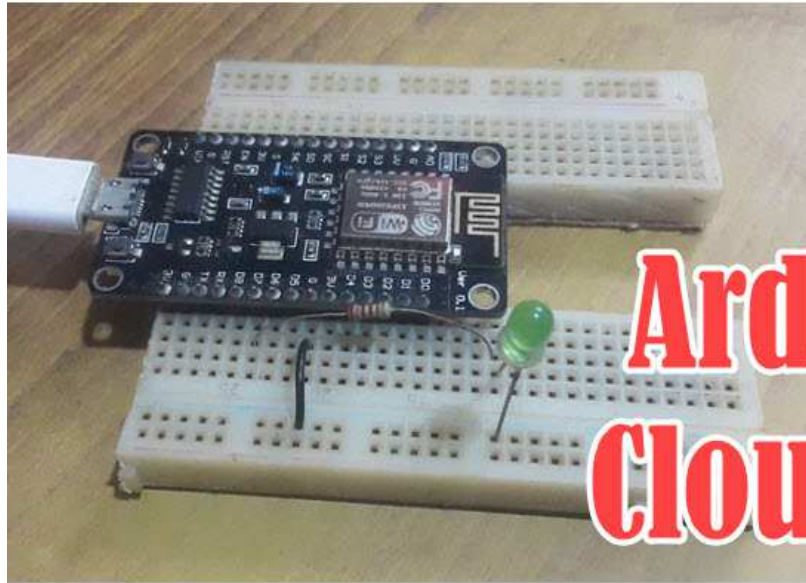
DHT 11 NODEMCU



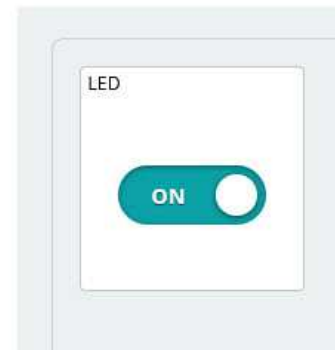
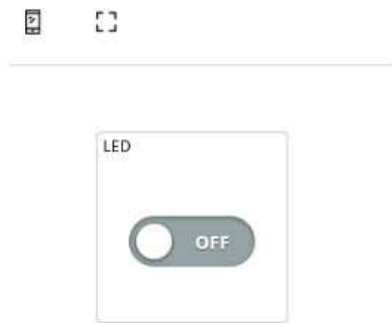
```
#include "DHT.h"
DHT dht;
void setup()
{
  Serial.begin(9600);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)");

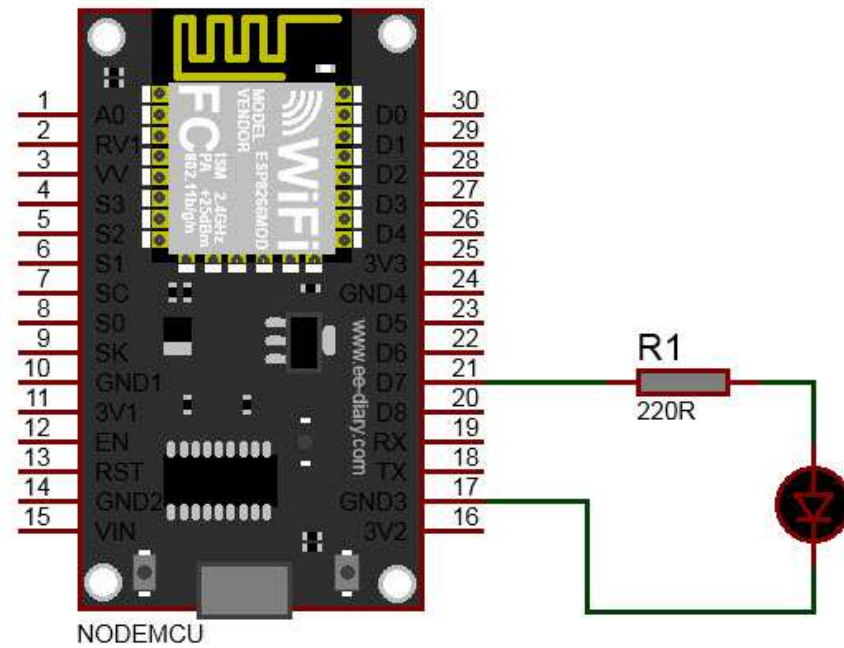
  dht.setup(2); // data pin 2
}

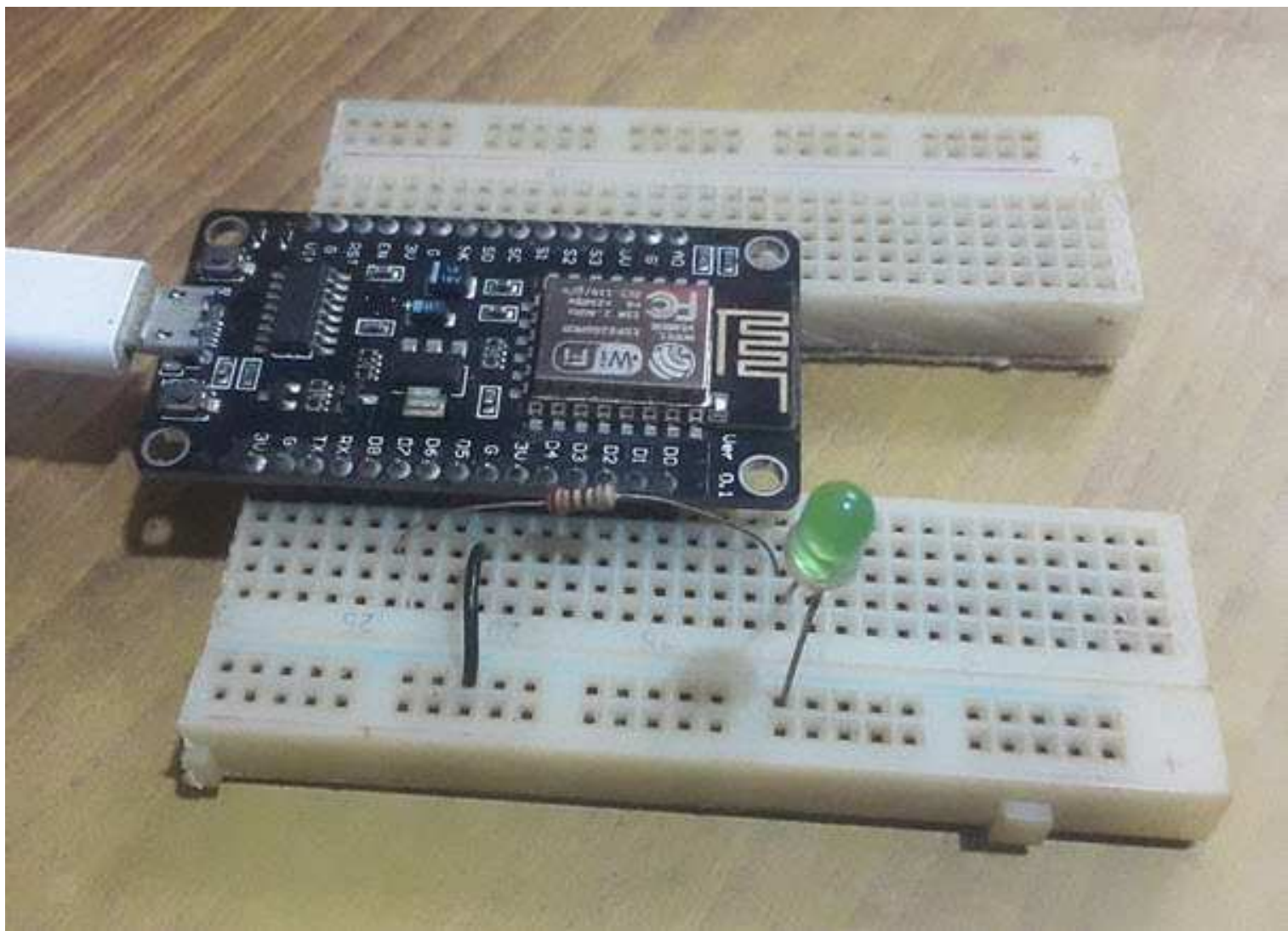
void loop()
{
  delay(dht.getMinimumSamplingPeriod());
  float humidity = dht.getHumidity(); // ดึงค่าความชื้น
  float temperature = dht.getTemperature(); // ดึงค่าอุณหภูมิ
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.println(dht.toFahrenheit(temperature), 1);
  delay(1000);
}
```



Arduino Cloud IoT platform









IOT CLOUD

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN



My Cloud



Dashboards



0DE3*_ga_NEXN8H46L5*MTY2NjU4ODI1My4yLjEuMTY2NjU4OTI0NC4wLjAuMA..



CREATE



Things



Search and filter Things



CREATE

Name ↓

Device

Variables

Last Modified



DHT11

NodeMCU
NodeMCU 1.0 (ESP-12E Mod...

dhthumid +1

22 Oct 2022 09:44:55





Untitled

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

Network

Enter your network credentials to connect your device.

Add variable



Name
myLED

 [Sync with other Things](#) 

boolean eg. true

Declaration
`bool myLED ;` 

Variable Permission 

☒ Read & Write ☐ Read Only

Variable Update Policy 

☒ On change ☐ Periodically

ADD VARIABLE

CANCEL



IOT CLOUD

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN



My Cloud



LEDControl

Setup

Sketch 1

Metadata

Cloud Variables

ADD

Name ↓

Last Value

Last Update



myLED

bool myLED;



Associated Device

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



Select Device

Network

Enter your network credentials to connect your device.

Associated Device

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



Select Device

Associate device



Choose the device to associate to
LEDControl



NodeMCU

NodeMCU 1.0 (ESP-12E Module)

 DHT11



NodeMCU

NodeMCU 1.0 (ESP-12E Module)


SET UP NEW DEVICE

Cloud Variables

ADD


Name ↓	Last Value	Last Update
<div><div><input type="checkbox"/></div><div><div>myLED</div><div>bool myLED;</div></div></div>	-	<div>⋮</div>

Associated Device

 NodeMCU

ID:

88d21c7e-7e50-4f1d-8b36-...





Type:

NodeMCU 1.0 (ESP-12E Module)

Status:

Offline

 Change

 Detach

LEDControl

Setup

Sketch 4

Metadata

Cloud Variables

ADD

Name ↓	Last Value	Last Update
<input type="checkbox"/> myLED <code>bool myLED;</code>	-	⋮



Change



Detach

Network

Enter your network credentials to connect your device.



Configure

Configure network



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

LOUD

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN



My Cloud

LEDControl

Setup

Sketch

Metadata

**NodeMCU** - NodeMCU 1.0 (ESP-12E Module)

Port: COM17



Open full editor



```
1 ▾ /*
2   Sketch generated by the Arduino IoT Cloud Thing "Untitled"
3   https://create.arduino.cc/cloud/things/c1aeddb7-451b-4e85-8402-de878412c455
4
5   Arduino IoT Cloud Variables description
6
7   The following variables are automatically generated and updated when changes are made to the Thing
8
9   bool myLED;
10
11   Variables which are marked as READ/WRITE in the Cloud Thing will also have functions
12   which are called when their values are changed from the Dashboard.
13   These functions are generated with the Thing and added at the end of this sketch.
14  */
15
16  #include "thingProperties.h"
17
18 ▾ void setup() {
```

```
16 #include "thingProperties.h"
17
18 const int my_LED = 13;
19
20 void setup() {
21     // Initialize serial and wait for port to open:
22     Serial.begin(9600);
23     // This delay gives the chance to wait for a Serial Monitor without blockin
24     delay(1500);
25
26     pinMode(my_LED, OUTPUT);
27
28     // Defined in thingProperties.h
29     initProperties();
30 }
```

```
53 ▾ /*  
54     Since MyLED is READ_WRITE variable, onMyLEDChange() is  
55     executed every time a new value is received from IoT Cloud.  
56 */  
57 ▾ void onMyLEDChange() {  
58     // Add your code here to act upon MyLED change  
59     Serial.println(myLED);  
60  
61     if(myLED)  
62         digitalWrite(my_LED,HIGH);  
63     else  
64         digitalWrite(my_LED,LOW);  
65 }  
66
```



LEDControl

Setup

Sketch

Metadata



NodeMCU - NodeMCU 1.0 (ESP-12E Module)

Port: COM17



Open full editor



```
51  
52  
53 ▾ /*  
54 Since MyLED is READ_WRITE variable, onMyLEDChange() is  
55 executed every time a new value is received from IoT Cloud.  
56 */  
57 ▾ void onMyLEDChange() {  
58 // Add your code here to act upon MyLED change  
59 Serial.println(myLED);  
60  
61 if(myLED)  
62 digitalWrite(my_LED,HIGH);  
63 else  
64 digitalWrite(my_LED,LOW);  
65 }  
66
```

Success: Done Uploading LEDControl_oct24a

closing bootloader

flush start

setting serial port timeouts to 1 ms

setting serial port timeouts to 1000 ms

flush complete

LEDControl_oct24a uploaded successfully on board NodeMCU 1.0 (ESP-12E Module) (COM17)

JD

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN



My Cloud

LEDControl

Setup

Sketch

Metadata



NodeMCU - NodeMCU 1.0 (ESP-12E Module)

Port: COM17

</> Open full editor



```
51  
52  
53 /*  
54  Since MyLED is READ_WRITE variable, onMyLEDChange() is  
55  executed every time a new value is received from IoT Cloud.  
56  */  
57 void onMyLEDChange() {  
58  // Add your code here to act upon MyLED change  
59  digitalWrite(LED, value);  
60 }
```

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN



My Cloud

Dashboards



Search and filter Dashboards



CREATE



Name ↓

Last modified

Owned by



DHT11

22 Oct 2022 11:14:45

eestudent



IOT CLOUD

Things

Dashboards

Devices

Integrations

Templates

UPGRADE PLAN



My Cloud



ADD



Untitled



WIDGETS

THINGS

 Search widgets



Switch



Push Button



Slider

LEDControl





IOT CLOUD

Things

Dashboards

Devices



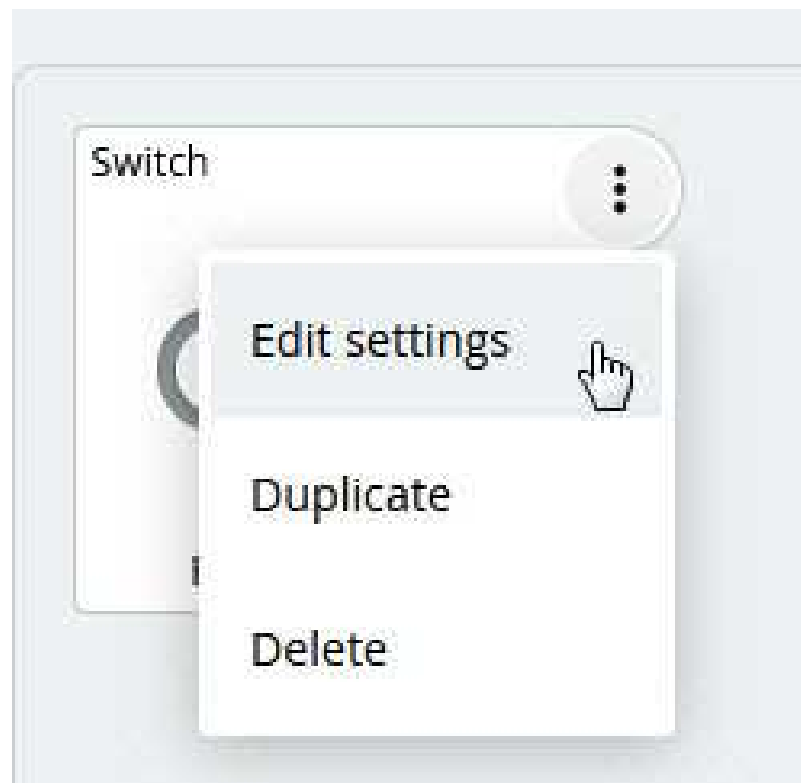
ADD ▾



Switch



Example Data



Switch

Switch



Example Data

Widget Settings

Name

Switch

Hide widget frame



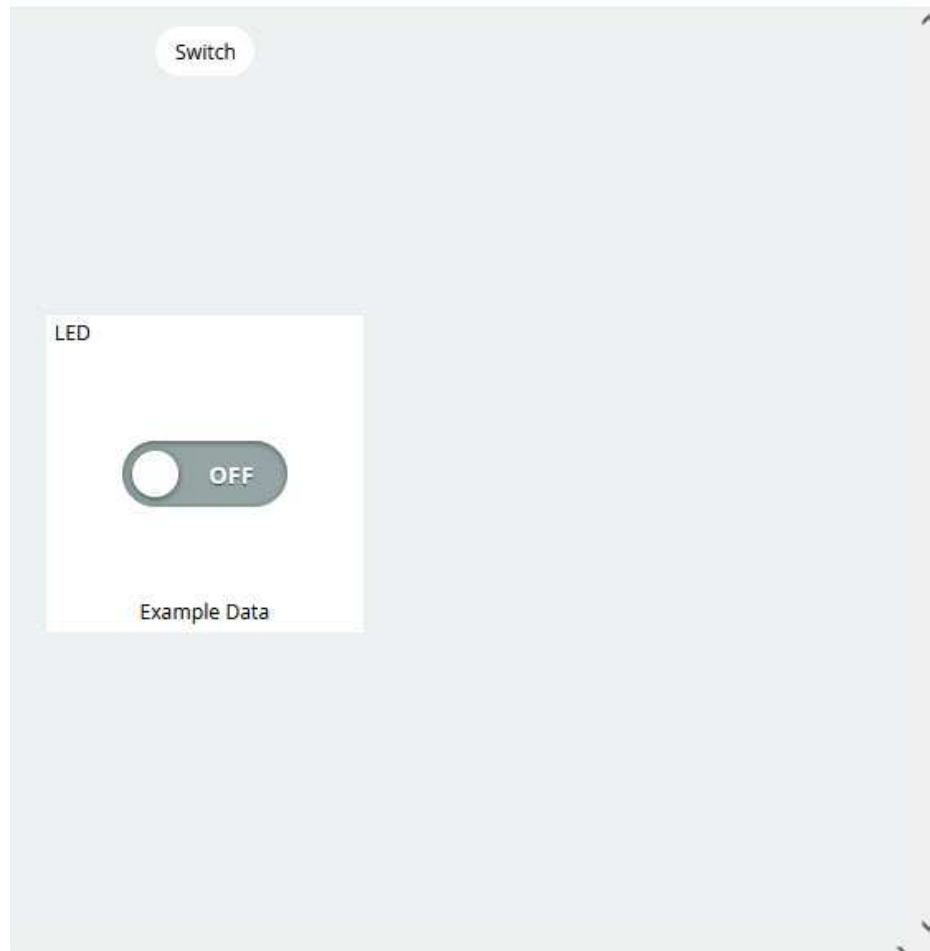
Linked Variable

This widget is displaying example data.
Select a source variable to display its value.



Link Variable

DONE



Widget Settings

Name
LED

Hide widget frame



Linked Variable

This widget is displaying example data.
Select a source variable to display its value.



Link Variable

DONE



Link Variable to LED

Things

DHT11
NodeMCU - NodeMCU 1.0 (ESP-12E Module)

LEDControl >
NodeMCU - NodeMCU 1.0 (ESP-12E Module)

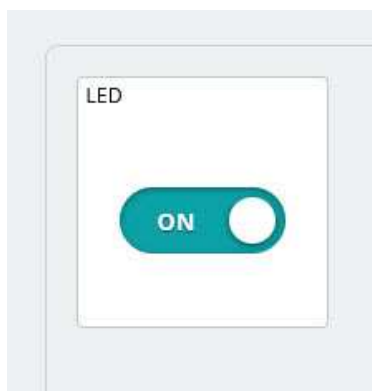
Variables

myLED >
Boolean

myLED

Thing LEDControl
Type Boolean
Last value -
Permission Read/Write
Update policy On change
Last update null

LINK VARIABLE






Humidity Temperature NODEMCU

Link Download Agent เพื่อ install ในคอมพิวเตอร์

- <https://create.arduino.cc/getting-started/plugin/welcome>

สร้าง Thing โดยประกอบด้วย temp และ humidity ดังภาพ

 Thing
humidity_Temperature ▾



Setup

Sketch 5


Metadata


Cloud Variables

ADD


	Name ▾	Last Value	Last Update	
<input type="checkbox"/>	dhthumid float dhthumid;	59	10 Feb 2024 12:25:54	⋮
<input type="checkbox"/>	dhttemp float dhttemp;	31.8	10 Feb 2024 12:25:21	⋮

Associated Device

 BoardhumidityTem

ID: a3350f1f-26ec-4e4d-a7f5-0... 

Type: NodeMCU 1.0 (ESP-12E Module)

Status:  Offline

⌵

Change

⌵

Detach

Network

Set webhook

กำหนดค่า Associated Device และ Network ตาม LAB เดิม

Associated Device

 BoardhumidityTemp

ID: a3350f1f-26ec-4e4d-a7f5-0... 

Type: NodeMCU 1.0 (ESP-12E Module)

Status: ☒ Offline

 Change


 Detach

Network

Wi-Fi Name: S8

Password:

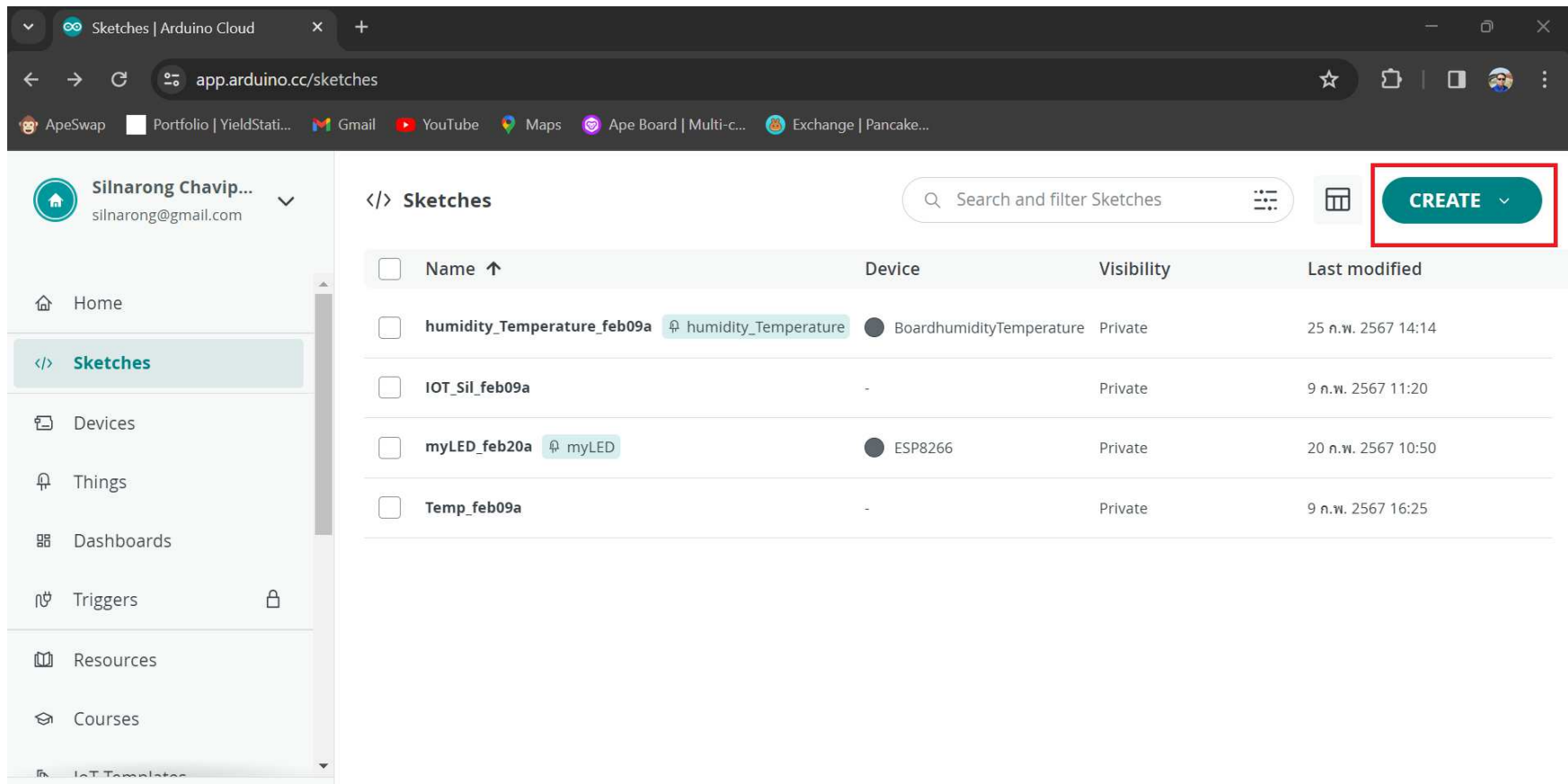
Secret Key:

 Change

การ Add Library DHT เข้าใน Sketches

- ไปที่เมนู Sketches ที่ใช้ในการเขียน Code ให้ Create Sketches โดย Add Library DHT เข้ามาดังนี้

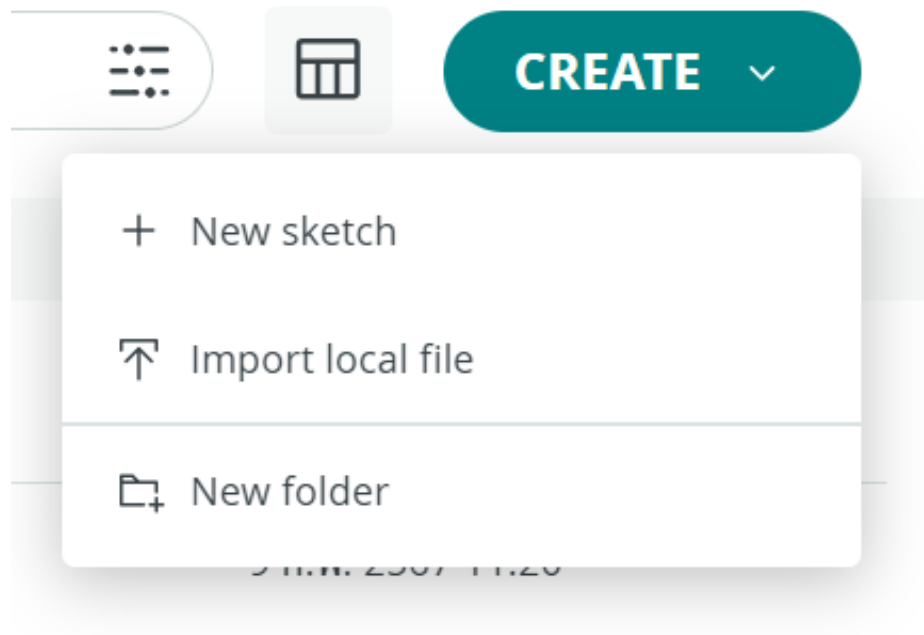
กด Create ด้านขวบน



The screenshot shows the Arduino Cloud interface in a web browser. The address bar displays 'app.arduino.cc/sketches'. The left sidebar contains a user profile for 'Silnarong Chavip...' and a navigation menu with options: Home, Sketches (selected), Devices, Things, Dashboards, Triggers, Resources, and Courses. The main content area is titled '</> Sketches' and includes a search bar labeled 'Search and filter Sketches'. A table lists several sketches, and a green 'CREATE' button with a dropdown arrow is highlighted with a red rectangular box in the top right corner of the main area.

<input type="checkbox"/>	Name ↑	Device	Visibility	Last modified
<input type="checkbox"/>	humidity_Temperature_feb09a <small>🔗 humidity_Temperature</small>	● BoardhumidityTemperature	Private	25 ก.พ. 2567 14:14
<input type="checkbox"/>	IOT_Sil_feb09a	-	Private	9 ก.พ. 2567 11:20
<input type="checkbox"/>	myLED_feb20a <small>🔗 myLED</small>	● ESP8266	Private	20 ก.พ. 2567 10:50
<input type="checkbox"/>	Temp_feb09a	-	Private	9 ก.พ. 2567 16:25

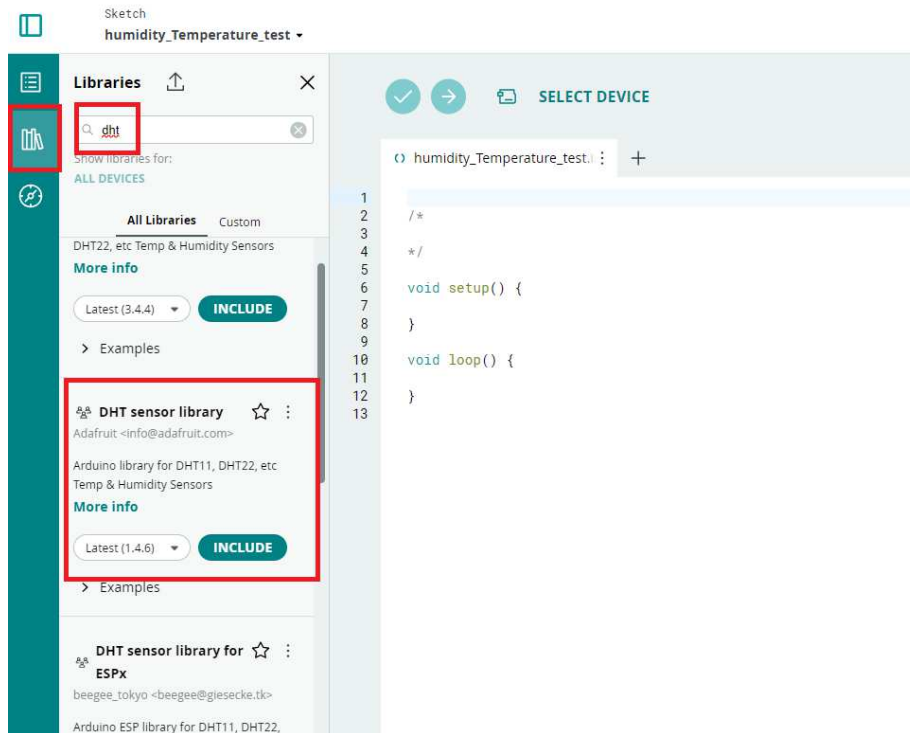
เลือก New Sketch



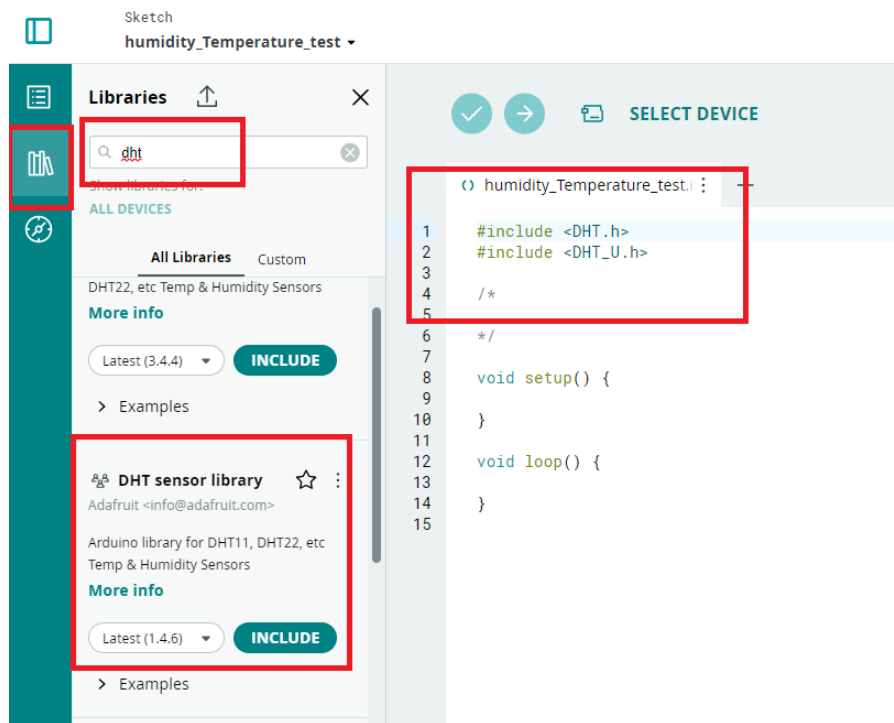
- ตั้งชื่อ **Sketch** ตามความเหมาะสมในที่นี้กำหนดชื่อ
humidity_Temperature_test

กด Icon Library ค้นหา Library ชื่อว่า dht ดังภาพ

- ทำการ Library ให้ Version ตรงกับภาพด้านล่าง



เมื่อคลิก Include จะปรากฏ Library ตามบรรทัดที่ 1 และ 2

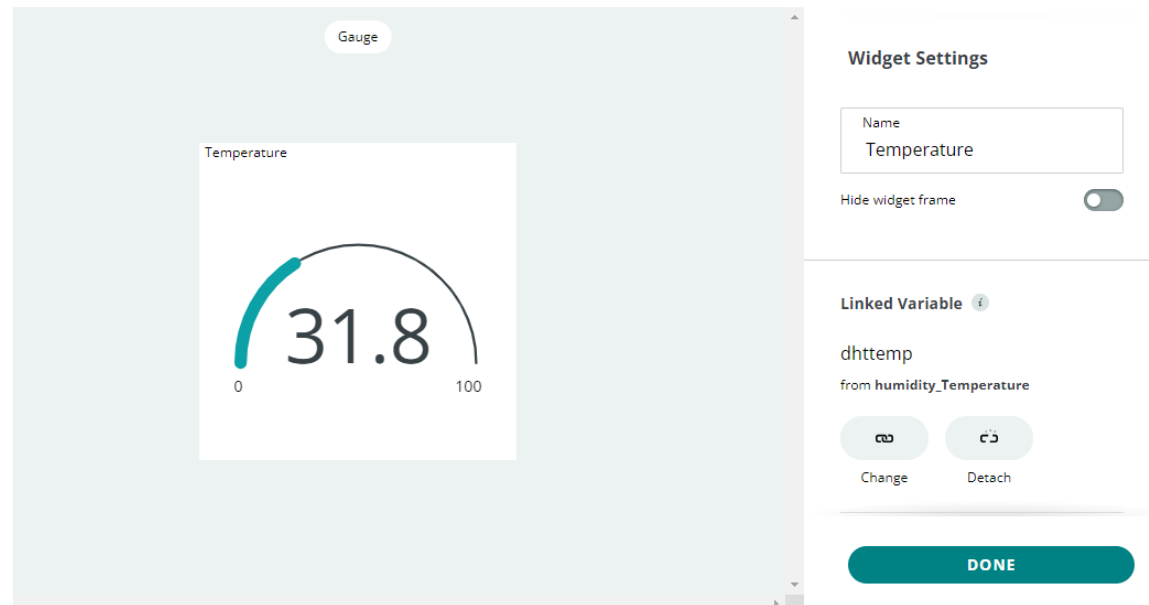
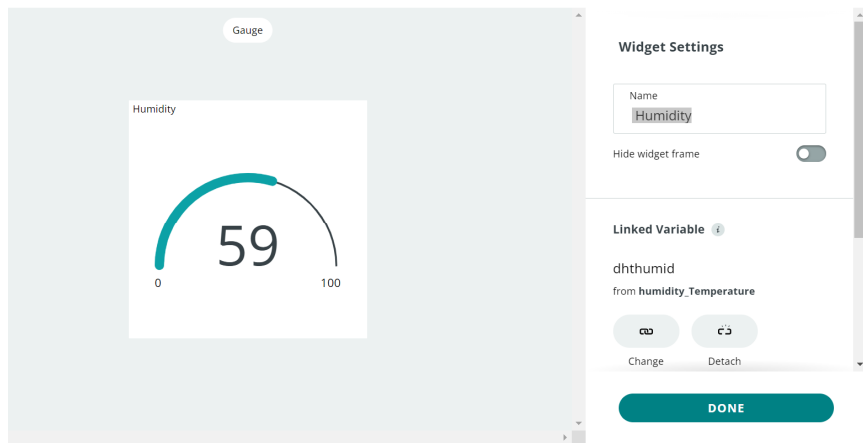


เขียน Code ใน Slide ต่อไป

```
1 // DHT sensor library - Version: Latest
2 #include <DHT.h>
3 #include <DHT_U.h>
4
5 #include "thingProperties.h"
6
7 DHT dht(13,DHT11);
8
9 void setup() {
10     // Initialize serial and wait for port to open:
11     Serial.begin(9600);
12     // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
13     delay(1500);
14
15     dht.begin();
16
17     // Defined in thingProperties.h
18     initProperties();
19
20     // Connect to Arduino IoT Cloud
21     ArduinoCloud.begin(ArduinoIoTPreferredConnection);
22
23     /*
24     The following function allows you to obtain more information
25     related to the state of network and IoT Cloud connection and errors
26     the higher number the more granular information you'll get.
27     The default is 0 (only errors).
28     Maximum is 4
29     */
30     setDebugMessageLevel(2);
31     ArduinoCloud.printDebugInfo();
32 }
33
34 void loop() {
35     ArduinoCloud.update();
36     // Your code here
37     delay(2000);
38
39     dhttemp = dht.readTemperature();
40     dhthumid = dht.readHumidity();
41
42 }
43
44
45
```

การตั้งค่า Dash Board

- สร้าง Dash Board โดยเลือกเป็นแบบ Gauge ทั้ง Humidity และ Temp



การ Link Variable ของ dhttempe และ dhthumid

- ให้ทำการ Link Variable ในแต่ละตัวตรงกับ Gauge ที่สร้าง
- ให้ Link ทั้งสองค่าให้ถูกต้องตามลำดับ

← Link Variable to Temperature

Things	Variables	dhthumid
humidity_Temperature BoardhumidityTemperature - NodeMCU 1.0 (ESP-12E M... >	dhthumid Float >	Thing humidity_Temperature
myLED ESP8266 - NodeMCU 1.0 (ESP-12E Module)	dhttemp Float	Type Float
		Last Value 59
		Permission Read-Only
		Update Policy On change - Threshold 0
		Last Update 10 Feb 2024 12:25:54
		LINK VARIABLE

ทดสอบการเชื่อมต่อ

- 1.ติดตั้ง **Agent** เพื่อเชื่อมต่อบอร์ดกับ **Arduino IOT**
- 2.เปิด **hostspot Wifi**
- 3.ดูผลการทดลองผ่าน **Dashboard** และบนมือถือ

- ต่อวงจรตามรูป



ให้นักศึกษากำหนดค่า

- บอร์ดที่ใช้เชื่อมต่อ
- ค่า **Think** และการเชื่อมต่อ **WiFi**

กำหนดค่าตัวแปร

- การกำหนดค่าตัวแปรจะมี 2 ตัวแปร จากรูปวงจรมีตัวแปรหลอดไฟ **LED** และ ตัวต้านทานแบบปรับค่าได้
- ตัวแปรแรกของเราคือ **Button** ซึ่งจำเป็นสำหรับการเปิด/ปิด **Led** ที่เชื่อมต่อกับ **Esp8266**

The screenshot shows the 'Add variable' dialog box in a web application. The dialog has a title bar with 'Add variable' and a close button. The main content area contains the following fields and options:

- Name:** A text input field containing 'Button'. A red arrow points to this field.
- Sync with other Things:** A toggle switch that is currently turned on. A red arrow points to this toggle.
- Boolean:** A dropdown menu showing 'Boolean eg. true'.
- Declaration:** A text area containing the code 'bool button;'. A red arrow points to this text area.
- Variable Permission:** Two radio buttons: 'Read & Write' (selected) and 'Read Only'. A red arrow points to the 'Read & Write' radio button.
- Variable Update Policy:** Two radio buttons: 'On change' (selected) and 'Periodically'. A red arrow points to the 'On change' radio button.

The background of the application shows a sidebar with 'Things', 'Dashboards', 'Devices', 'Integrations', and 'Templates'. The main content area shows a 'Device' card for '3266_IOT' with a 'Metadata' tab selected.

กำหนดค่าตัวแปร

- ตัวแปรที่สองของเรากำหนดชื่อ **Data** คือให้ตัวต้านทานแบบปรับค่าได้แสดงข้อมูลบนแดชบอร์ดบน Arduino IOT

The screenshot shows the 'Add variable' dialog box in the Arduino IOT dashboard. The dialog has a title bar with 'Add variable' and a close button. The main content area includes the following fields and options:

- Name:** A text input field containing 'Data'.
- Sync with other Things:** A button with a circular arrow icon.
- Integer Number eg. 1:** A dropdown menu showing the selected data type.
- Declaration:** A text input field containing 'int data;'.
- Variable Permission:** A section with two radio buttons: 'Read & Write' and 'Read Only'. The 'Read Only' option is selected.
- Variable Update Policy:** A section with two radio buttons: 'On change' and 'Periodically'. The 'On change' option is selected.
- Threshold:** A text input field containing '0'.

Red arrows are drawn on the image to highlight the 'Name' field, the 'Integer Number' dropdown, the 'Read Only' radio button, and the 'On change' radio button.

เขียน Code ดังนี้(1)

Things Dashboards Devices Integrations Templates

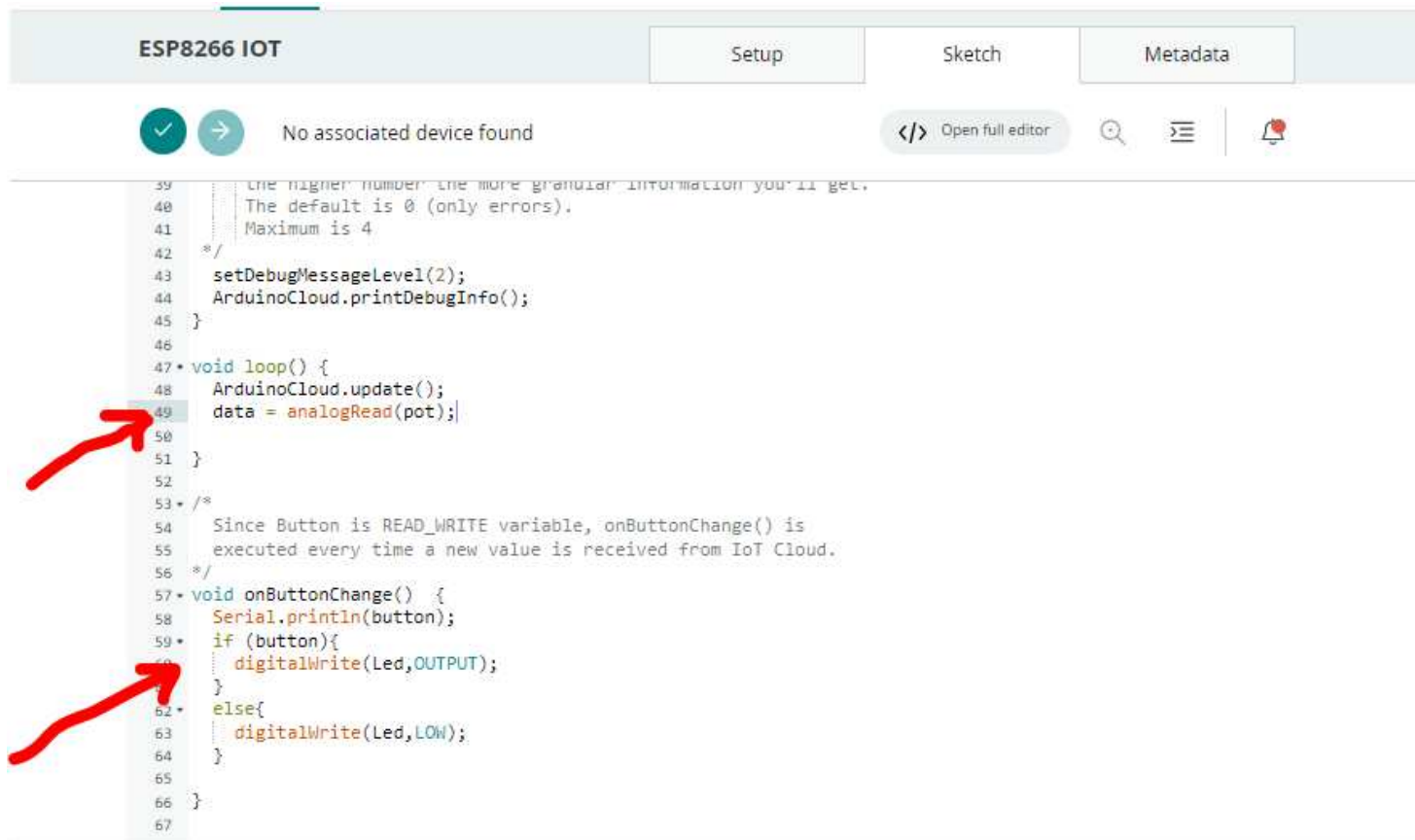
UPGRADE PLAN My Cloud

ESP8266 IOT Setup Sketch Metadata

✓ → No associated device found </> Open full editor 🔍 ☰ 🔔

```
16
17 #include "thingProperties.h"
18 int Led = 15;
19 int pot = A0;
20
21
22 void setup() {
23   // Initialize serial and wait for port to open:
24   pinMode(Led, OUTPUT);
25
26   Serial.begin(9600);
27   // This delay gives the chance to wait for a Serial Monitor without blocking if none is found
28   delay(1500);
29
30   // Defined in thingProperties.h
31   initProperties();
32
33   // Connect to Arduino IoT Cloud
34   ArduinoCloud.begin(ArduinoIoTPreferredConnection);
35
36   /*
37    * The following function allows you to obtain more information
38    * related to the state of network and IoT Cloud connection and errors
39    * the higher number the more granular information you'll get.
40    * The default is 0 (only errors).
41    * Maximum is 4
42    */
43   setDebugMessageLevel(2);
44   ArduinoCloud.printDebugInfo();
```

เขียน Code ดังนี้(2)

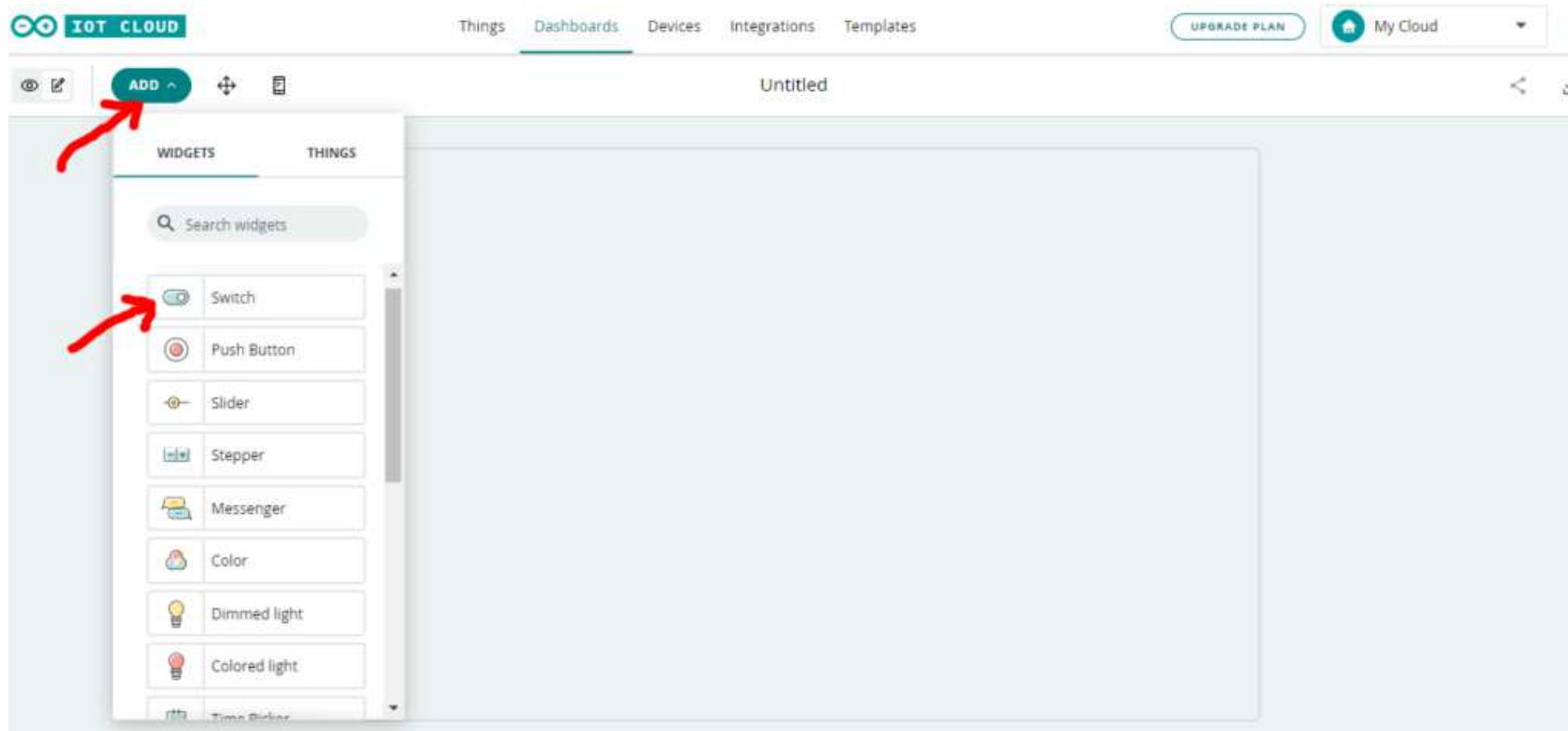


```
ESP8266 IOT Setup Sketch Metadata

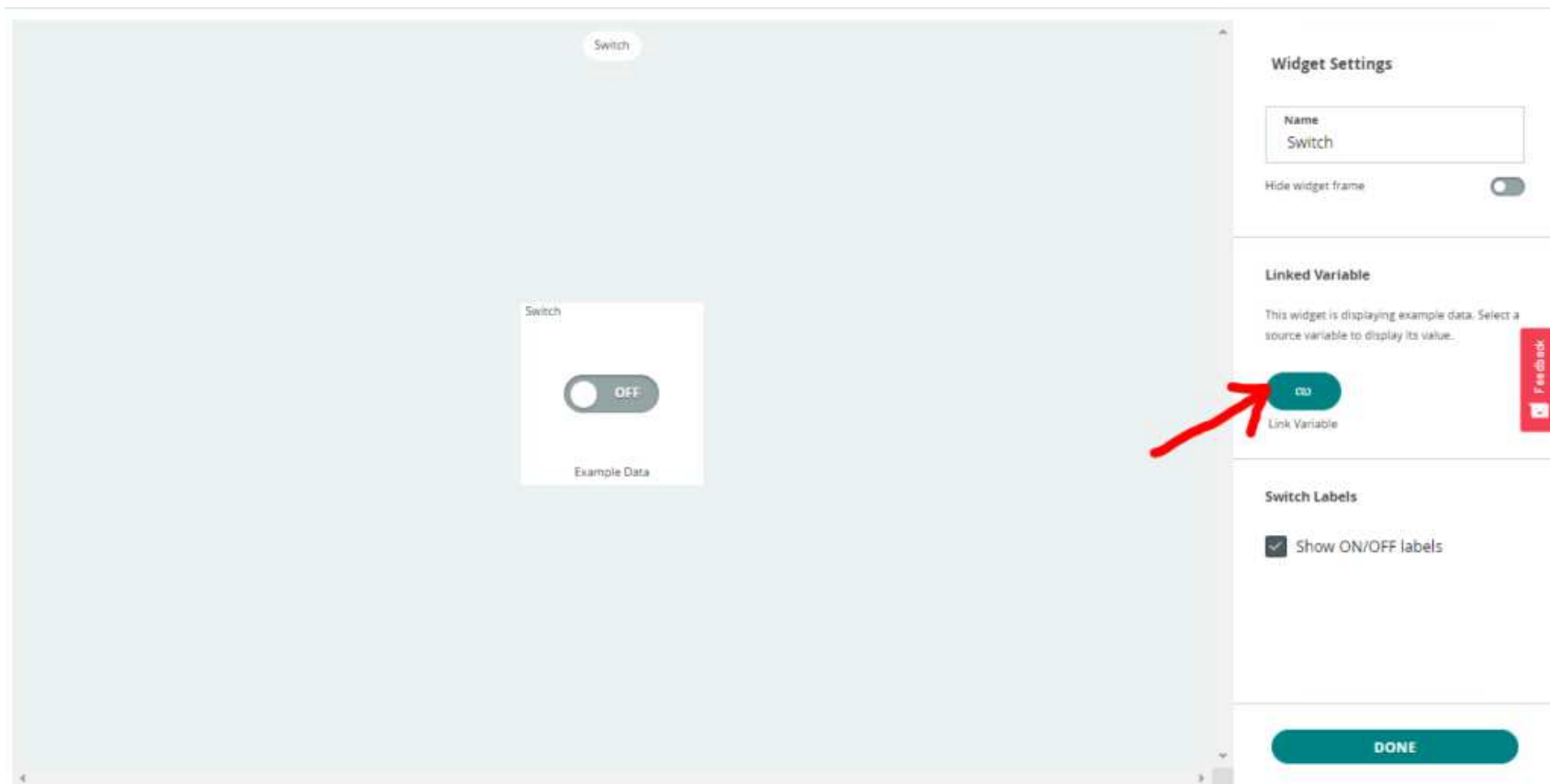
No associated device found </> Open full editor

39  /* The higher number the more granular information you'll get.
40  /* The default is 0 (only errors).
41  /* Maximum is 4.
42  */
43  setDebugMessageLevel(2);
44  ArduinoCloud.printDebugInfo();
45  }
46
47  void loop() {
48  ArduinoCloud.update();
49  data = analogRead(pot);
50  }
51  }
52
53  /*
54  /* Since Button is READ_WRITE variable, onButtonChange() is
55  /* executed every time a new value is received from IoT Cloud.
56  */
57  void onButtonChange() {
58  Serial.println(button);
59  if (button){
60  digitalWrite(Led,OUTPUT);
61  }
62  else{
63  digitalWrite(Led,LOW);
64  }
65  }
66  }
67  }
```

การสร้าง Dashboard - 1



การสร้าง Dashboard - 2



The screenshot displays a dashboard editor interface. On the left, a light blue workspace contains a 'Switch' widget. The widget is a white square with a grey toggle switch in the 'OFF' position. Above the widget is a small label 'Switch', and below it is 'Example Data'. On the right, a settings panel is open. It has a title 'Widget Settings' and a text input field for 'Name' containing 'Switch'. Below this is a toggle for 'Hide widget frame'. The next section is 'Linked Variable', which includes a text prompt: 'This widget is displaying example data. Select a source variable to display its value...'. A red arrow points to a teal button labeled 'Link Variable' in this section. Below that is the 'Switch Labels' section with a checked checkbox for 'Show ON/OFF labels'. At the bottom of the settings panel is a teal 'DONE' button. A red 'Feedback' button is visible on the far right edge of the settings panel.

การสร้าง Dashboard - 3

← Link Variable to Switch

Things

ESP8266 IOT >

Variables

Data >

Button >

Button

Thing	ESP8266 IOT
Type	Boolean
Last value	-
Permission	Read/Write
Update policy	On change
Last update	null

LINK VARIABLE

Feedback

การสร้าง Dashboard - 4

- ทำการเชื่อมต่อ **Dashboard** ให้ถูกต้องในแต่ละตัวแปร
- และสร้าง **Dashboard** ที่แสดงผลในส่วนของคุณค่าตัวคำนวณแบบปรับค่าได้

ทดสอบการเชื่อมต่อ

- ดูผลงานหน้าจอกอมพิวเตอร์และบนมือถือของตัวเอง

การสร้าง Dashboard

