

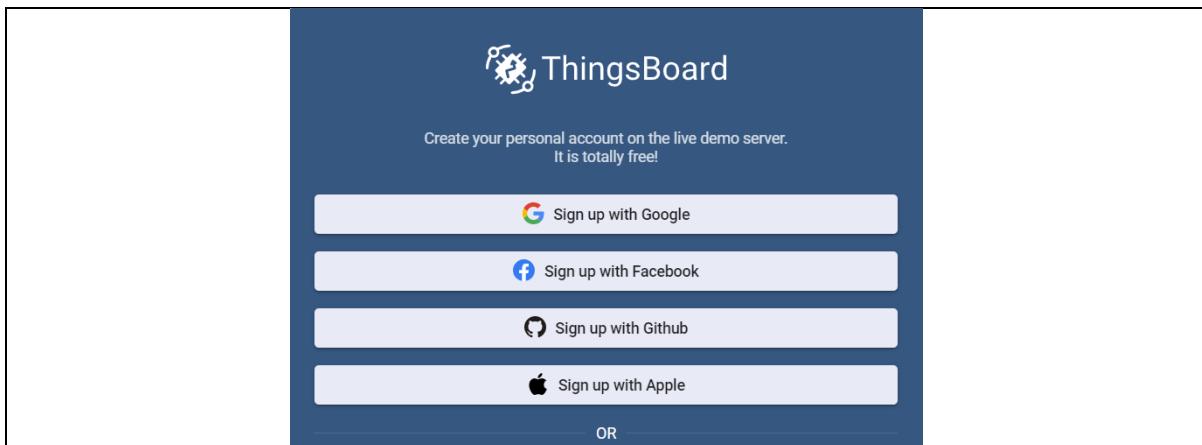
**การใช้งาน ThingsBoard IoTs Platform เพื่อสร้างและจัดการระบบอัตโนมัติ**

**ThingsBoard IoTs Platform for smart system**

ชื่อ-สกุล : นายรชพล พงศ์กิตติศักดิ์

**การลงทะเบียนใช้งาน Thingsboard**

เข้าไปที่ Link: <https://demo.thingsboard.io/signup> และกดไปที่ Sign up with Google



**Quiz\_101 – ThingsBoard Data Monitor**

- Mission - 1/4: ให้ส่งข้อมูลค่า Humidity และ Temperatures จากเซ็นเซอร์ DHT-22 ไปยัง Dashboard

**1. สร้าง Device**

1.1 ทำการสร้าง Device ใหม่ โดยไปที่ Devices -> + -> Add new device

Created time	Name	Device profile	Label	Customer	Public	Is gateway
2022-04-29 21:27:07	OfflineDevice	default				
2022-04-29 20:24:39	BoundaryCheckData	default				

## 1.2 สร้าง Device โดยในช่อง Name ตั้งชื่อตามนี้ จากนั้นกดปุ่ม Add

M2-Q101

Add new device

1 Device details      2 Credentials Optional      3 Customer Optional

Name \*  
M2-Q101

Label

Device profile \*  
Select existing device profile default

Create new device profile

Is gateway

Description

Next: Credentials

Cancel      Add

## 2. ส่งข้อมูล Humidity และ Temperatures ไปยัง Device

### 2.1 Copy Access Token ของ Device ที่สร้าง ไปที่ Devices -> ชื่อ Device -> Copy access token

Devices All

Created time	Name	Device profile
2022-05-02 15:33:08	M2-Q101	default
2022-04-29 21:27:07	OfflineDevice	default
2022-04-29 20:24:39	BoundaryCheckData	default

**M2-Q101**

Device details

Details Attributes Latest telemetry Alarms Events Relations Audit Logs

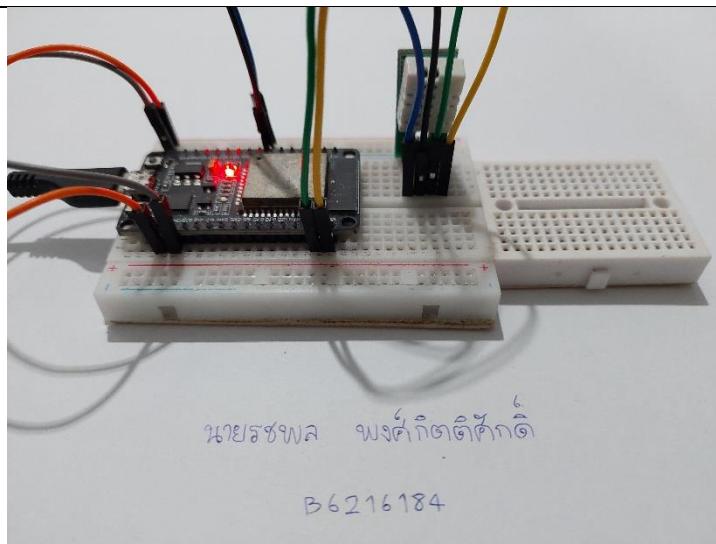
Open details page Make device public Assign to customer Manage credentials Delete device

Copy device id Copy access token

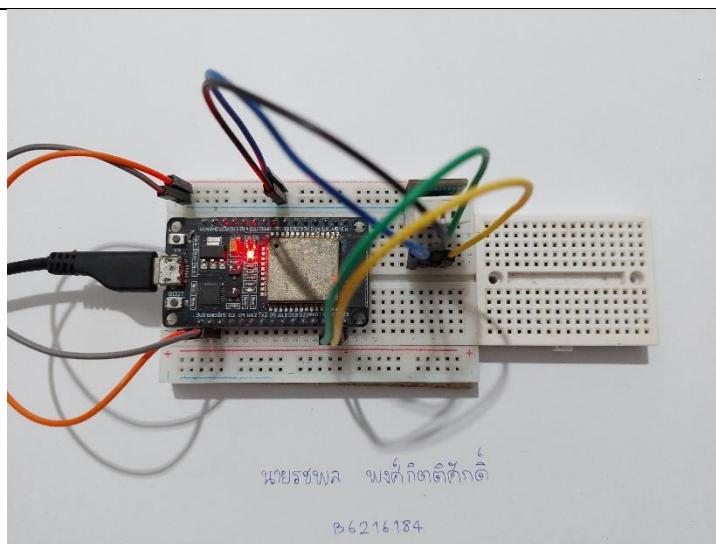
Access Token ก็คือ DEVICE\_TOKEN ใน Code

## 2.2 ต่อวงจร

รูปการต่อวงจร 1 – ด้านข้าง



รูปการต่อวงจร 2 – ด้านบน



## 2.3 นำ Code อัปโหลดลง ESP32

## Code

```
#include <WiFi.h>
#include <Wire.h>
#include <ThingsBoard.h>
#include "SHTC3.h"
#define WIFI_NAME "your-wifi-name"
#define WIFI_PASS "your-wifi-pass"
#define SERVER "demo.thingsboard.io"
#define DEVICE_TOKEN "your-device-token"
```

```
SHTC3 sht(Wire);
WiFiClient esp32Client;
ThingsBoard tb(esp32Client);

void setup() {
    Serial.begin(115200);
    Wire.begin(22, 23);
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi Connected");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
}

void reconnectTB() {
    if(!tb.connect(SERVER, DEVICE_TOKEN)) {
        Serial.println("Failed to connect to ThingsBoard server");
    }
    else {
        Serial.println("Connected to ThingsBoard server");
    }
}

void reconnectWiFi() {
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    randomSeed(millis());
    Serial.println("\nWiFi Connected");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
}

void loop() {
    if(WiFi.status() != WL_CONNECTED) reconnectWiFi();
```

```

if(!tb.connected()) reconnectTB();
tb.loop();
sht.begin();
sht.sample();
float t = sht.readTempC();
float h = sht.readHumidity();
if(t >= -40 && t <= 125) {
    tb.sendTelemetryFloat("temperature", t);
    tb.sendTelemetryFloat("humidity", h);
    Serial.println("temperature: " + String(t) + ", humidity: " + String(h));
}
delay(5000);
}

```

2.4 ผลลัพธ์หลังจากส่งข้อมูล Humidity และ Temperatures ไปยัง Device จะได้ตัวแปร เพื่อนำไปใช้สร้าง Dashboard ต่อไป

The screenshot shows the ThingsBoard interface. On the left is a sidebar with navigation links: Home, Rule chains, Customers, Assets, Devices (selected), Device profiles, OTA updates, Entity Views, and Edge instances. The main area has a header 'Devices' and a search bar with 'All' selected. Below is a table of devices:

	Created time	Name	Device profile
<input type="checkbox"/>	2022-05-03 01:03:38	M2-Q104	default
<input type="checkbox"/>	2022-05-02 23:51:24	M2-Q103	default
<input type="checkbox"/>	2022-05-02 16:32:18	M2-Q102	default
<input type="checkbox"/>	2022-05-02 15:33:08	M2-Q101	default
<input type="checkbox"/>	2022-04-29 21:27:07	OfflineDevice	default

To the right, a modal window is open for 'M2-Q101'. It has tabs for Details, Attributes, Latest telemetry (selected), Alarms, Events, Relations, and Audit Logs. The 'Latest telemetry' section shows the following data:

Last update time	Key ↑	Value
2022-05-03 10:58:24	humidity	69.25354004
2022-05-03 10:58:24	temperature	26.04034424

A red box highlights the 'Latest telemetry' table.

### 3. สร้าง Dashboard

3.1 ทำการสร้าง Dashboard ใหม่ โดยไปที่ Dashboards -> + ->Create new dashboard

The screenshot shows the left sidebar with various menu items like Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, and Dashboards. The 'Dashboards' item is highlighted with a red box. The main area displays a list of existing dashboards with columns for Created time, Title, Assigned to customers, and several action icons. In the top right corner, there is a modal or a floating menu with a '+' icon, a search icon, and a 'Create new dashboard' button, which is also highlighted with a red box.

3.2 สร้าง Dashboard โดยตั้งค่าตามนี้ จากนั้นกดปุ่ม Add

The screenshot shows the 'Add Dashboard' dialog box. It has fields for 'Title' (M2-Q101-ThingsBoard-Data-Monitor) and 'Description'. Under 'Mobile application settings', there is a 'Dashboard image' section with a placeholder 'No image selected' and a file upload area with the instruction 'Drop an image or click to select a file to upload'. A note says 'Maximum upload file size: 512.0 KB'. There is also a checkbox for 'Hide dashboard in mobile application'. At the bottom are 'Cancel' and 'Add' buttons.

3.3 กดเข้าไปที่ Dashboard ที่สร้างขึ้นมาและกด Open dashboard

The screenshot shows the 'Dashboard details' view for the 'M2-Q101-ThingsBoard-Data-Monitor' dashboard. It includes tabs for Details, Audit Logs, and a central area with the dashboard title and a preview. Below the preview are buttons for 'Open dashboard' (highlighted with a red box), 'Export dashboard', 'Make dashboard public', and 'Manage assigned customers'. There is also a delete button labeled 'Delete dashboard'.

3.4 จากนั้นกดที่รูป ด้านล่างขวาเพื่อเข้าสู่โหมดแก้ไขจากนั้นกดไปที่ -> Add alias

Device-M2-Q101

Add alias X

Alias name *	<input type="text" value="Device-M2-Q101"/>	Resolve as multiple entities <input checked="" type="checkbox"/>
Filter type *	<input type="button" value="Single entity"/>	
Type *	Device <input type="button" value="▼"/>	Device * <input type="text" value="M2-Q101"/> <span style="float: right;">X</span>

Cancel Add

เมื่อ Add alias ทั้งหมดแล้วจะได้ผลลัพธ์เป็นแบบนี้ จากนั้นให้กด Save

Entity aliases X

Alias name	Entity filter	Resolve as multiple entities
1. Device-M2-Q101	One device	<input checked="" type="checkbox"/>

Add alias Cancel Save

3.5 สร้าง Widget Analogue gauges โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่ จากนั้นเลือก

Analogue gauges -> Radial gauge

<b>Analogue gauges System</b> Display temperature, humidity, speed, and other latest values on various analog gauge widgets.	<b>Radial gauge Latest values</b> Preconfigured gauge to display any value reading. Allows to configure value range, gradient colors and other settings.
---	---

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

Add Widget: Radial gauge

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q101      Filter

Data settings

Cancel      Add

ที่ tab Advanced ให้ตั้งค่าตามนี้ จากนั้นกด Add

Add Widget: Radial gauge

Data      Settings      Advanced      Actions

Start ticks angle  
45

Ticks angle  
270

Needle circle size  
10

Minimum value  
0

Maximum value  
100

Unit title

Cancel      Add

3.6 สร้าง Widget Analogue gauges โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



Digital gauges -> Digital vertical bar

	<b>Digital gauges</b> System Display temperature, humidity, speed, and other latest values on various digital gauge widgets.	<b>Digital vertical bar</b> Latest values Preconfigured gauge to display any value reading as a vertical bar. Allows to configure value range, gradient colors and other settings.
--	--	--

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q101      Filter

Entity alias \* Device-M2-Q101      Filter

Maximum 1 timeseries/attribute is allowed.

Data settings

Cancel      Add

ที่ tab Advanced ให้ตั้งค่าตามนี้ จากนั้นกด Add

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Minimum value  
0

Maximum value  
100

Gauge type  
Vertical bar

Angle to start from when in donut mode  
90

Neon glow effect brightness, (0-100), 0 - disable effect  
20

Thickness of the stripes, 0 - no stripes  
1.2

Cancel      Add

## 3.7 ทำการสร้าง Widget Chart โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



**Charts -> Timeseries Line Chart**

<p><b>Charts</b> System Display timeseries data using customizable line and bar charts. Use various pie charts to display latest values.</p>	<p><b>Timeseries Line Chart</b> Time series Displays changes to timeseries data over time. For example, temperature or humidity readings.</p>
--	---

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

**Add Widget: Timeseries Line Chart**

Data	Settings	Advanced	Actions						
<input checked="" type="checkbox"/> Use dashboard timewindow <input checked="" type="checkbox"/> Display timewindow <b>Datasources</b> <table border="1"> <tr> <th>Type</th> <th>Parameters</th> </tr> <tr> <td>= 1. Entity</td> <td>           Entity alias * Device-M2-Q101            = 🔍 ⚡ temper... : tempera... 🖊️ ✕            Filter _____         </td> </tr> <tr> <td colspan="2"> <a href="#">+ Add</a> </td> </tr> </table>	Type	Parameters	= 1. Entity	Entity alias * Device-M2-Q101 = 🔍 ⚡ temper... : tempera... 🖊️ ✕ Filter _____	<a href="#">+ Add</a>		Timewindow 	<a href="#">Cancel</a> <a href="#">Add</a>	
Type	Parameters								
= 1. Entity	Entity alias * Device-M2-Q101 = 🔍 ⚡ temper... : tempera... 🖊️ ✕ Filter _____								
<a href="#">+ Add</a>									

## 3.8 สร้าง Widget Analogue gauges โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



**Analogue gauges -> Radial gauge**

<p><b>Analogue gauges</b> System Display temperature, humidity, speed, and other latest values on various analog gauge widgets.</p>	<p><b>Radial gauge</b> Latest values Preconfigured gauge to display any value reading. Allows to configure value range, gradient colors and other settings.</p>
---	---

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

Add Widget: Radial gauge

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q101      humidity: humidity

Filter

Data settings

Cancel      Add

ที่ tab Advanced ให้ตั้งค่าตามนี้ จากนั้นกด Add

Add Widget: Radial gauge

Data      Settings      Advanced      Actions

Start ticks angle  
45

Ticks angle  
270

Needle circle size  
10

Minimum value  
0

Maximum value  
100

Unit title

Cancel      Add

3.9 สร้าง Widget Analogue gauges โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



Digital gauges -> Digital vertical bar

<p><b>Digital gauges</b> System</p> <p>Display temperature, humidity, speed, and other latest values on various digital gauge widgets.</p>	<p><b>Digital vertical bar</b> Latest values</p> <p>Preconfigured gauge to display any value reading as a vertical bar. Allows to configure value range, gradient colors and other settings.</p>
--	--

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q101      = ⚙️ ↗ humidity:humidity ✎ X

Filter

Maximum 1 timeseries/attribute is allowed.

Data settings

Cancel      Add

ที่ tab Advanced ให้ตั้งค่าตามนี้ จากนั้นกด Add

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Minimum value  
0

Maximum value  
100

Gauge type  
Vertical bar

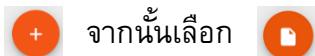
Angle to start from when in donut mode  
90

Neon glow effect brightness, (0-100), 0 - disable effect  
20

Thickness of the stripes, 0 - no stripes  
1.2

Cancel      Add

### 3.10 ทำการสร้าง Widget Chart โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก

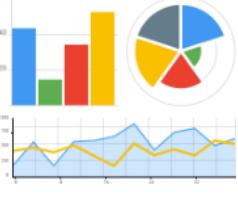


**Charts -> Timeseries Line Chart**

**Charts**

System

Display timeseries data using customizable line and bar charts. Use various pie charts to display latest values.



**Timeseries Line Chart**

Time series

Displays changes to timeseries data over time. For example, temperature or humidity readings.



ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

**Add Widget: Timeseries Line Chart**

Data      Settings      Advanced      Actions

Use dashboard timewindow      Timewindow (Realtime - last minute)

Display timewindow

Datasources

Type	Parameters
= 1. Entity	Entity alias * Device-M2-Q101 = Device-M2-Q101 humidity:humidity
<input type="button" value="+ Add"/>	

ที่ tab Settings ในส่วนของ Title ให้ตั้งชื่อเป็น Timeseries Line Chart จากนั้นกดปุ่ม Add

**Add Widget: Timeseries Line Chart**

Data      Settings      Advanced      Actions

Title

Display widget title

Title: Humidity Timeseries Chart

Title Tooltip:

Title icon

Display title icon

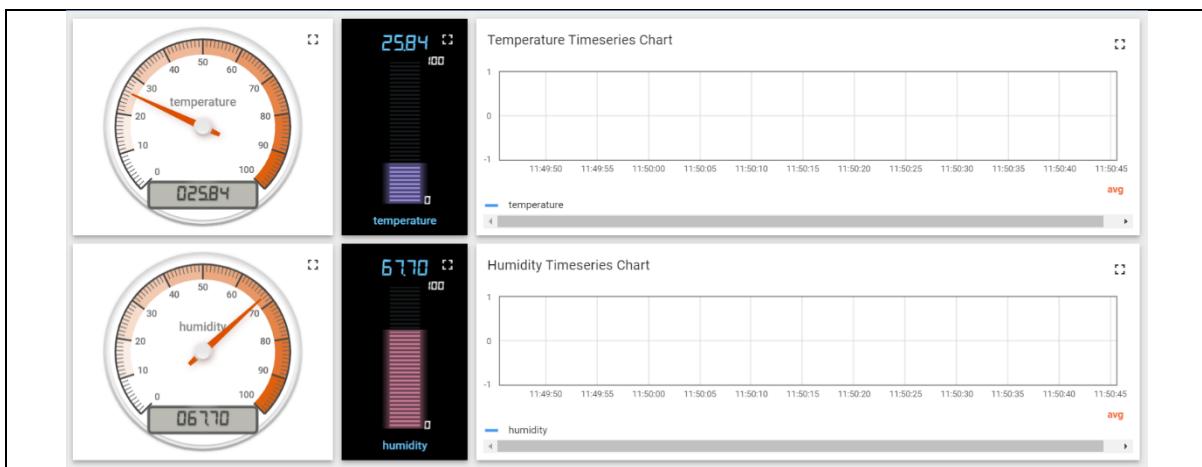
Icon:  Icon color: rgba(0, 0, 0, 0.87) Icon size: 24px

Advanced settings ▼

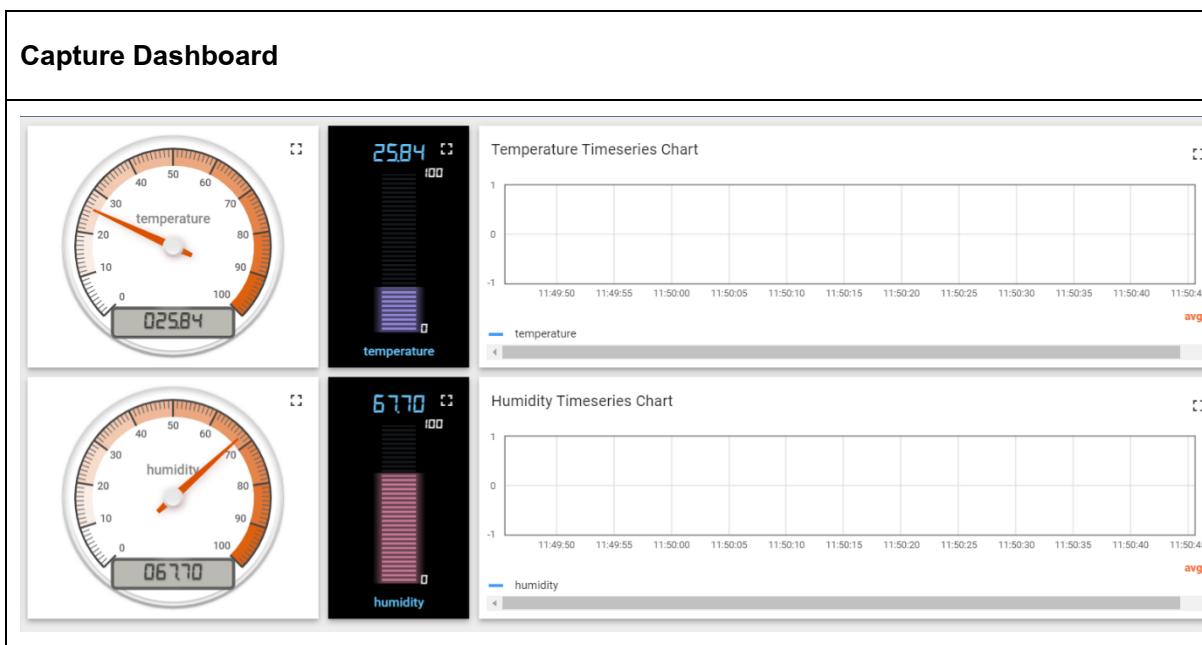
Widget style

Background color: #fff Text color: rgba(0, 0, 0, 0.87) Padding: 8px Margin:

3.11 ปรับหน้าตา Dashboard ให้เป็นตามนี้ จากนั้นกดที่  เพื่อบันทึกการแก้ไข



#### 4. ผลลัพธ์

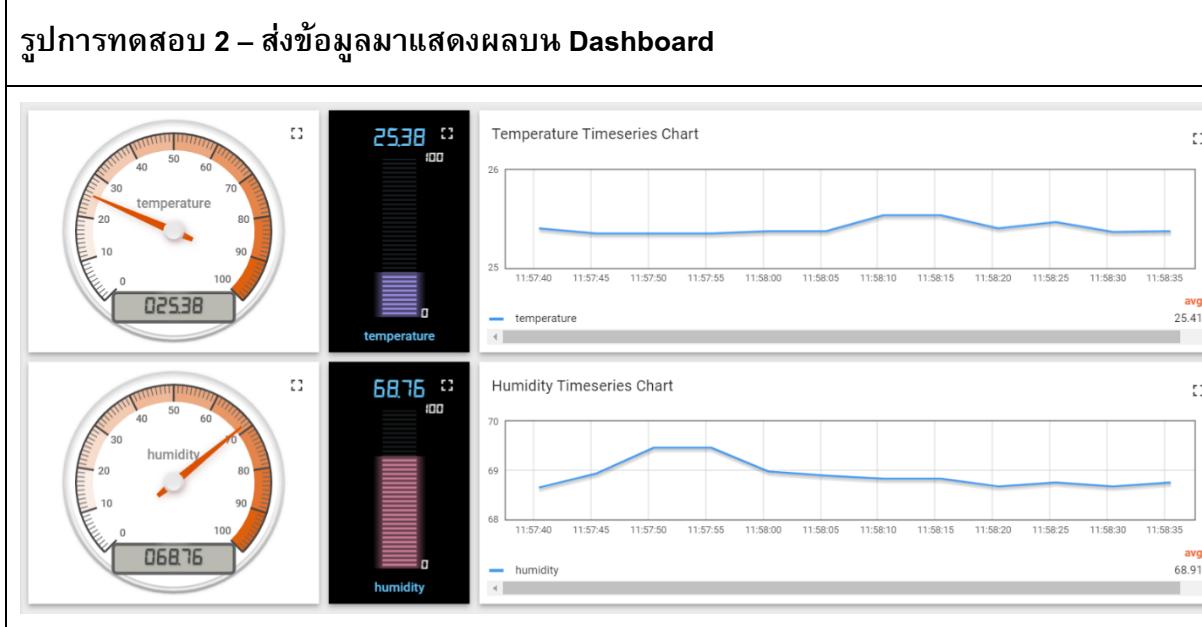


### รูปการทดสอบ 1 – ส่งข้อมูลมา Device

The screenshot shows the ThingsBoard interface. On the left is a sidebar with navigation links: Home, Rule chains, Customers, Assets, Devices (selected), Device profiles, OTA updates, Entity Views, and Edge instances. The main area has a title "M2-Q101" and "Device details". It includes tabs for Details, Attributes, Latest telemetry (selected), Alarms, Events, Relations, and Audit Logs. A red circle with a white edit icon is visible in the top right. Below the tabs is a section titled "Latest telemetry" with columns for Last update time, Key, and Value. The table contains two rows of data.

Last update time	Key	Value
2022-05-03 10:58:24	humidity	69.25354004
2022-05-03 10:58:24	temperature	26.04034424

### รูปการทดสอบ 2 – ส่งข้อมูลมาแสดงผลบน Dashboard



## Quiz\_102 – ThingsBoard Data Monitor and Control

- Mission 2/4: ให้ส่งข้อมูลค่า Humidity และ Temperatures จากเซ็นเซอร์ DHT-22 ไปยัง ThingsBoard  
พร้อมทั้งควบคุม On/Off - 4 LED และ Blink Speed สำหรับอีก 1 LED

### 1. สร้าง Device

1.1 ทำการสร้าง Device ใหม่ โดยไปที่ Devices -> + -> Add new device

1.2 สร้าง Device โดยในช่อง Name ตั้งชื่อตามนี้ จากนั้นกดปุ่ม Add

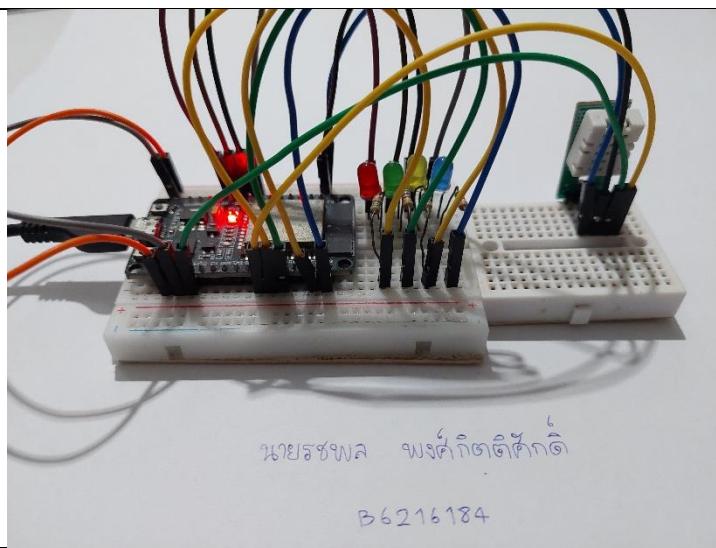
### 2. ส่งข้อมูลไปยัง Device

2.1 Copy Access Token ของ Device ที่สร้าง ไปที่ Devices -> ชื่อ Device -> Copy access token

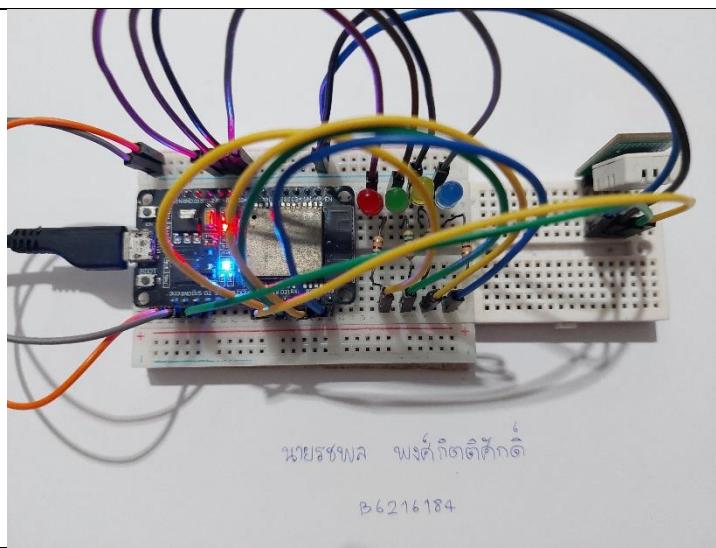
Access Token ก็คือ DEVICE\_TOKEN ใน Code – Main

## 2.2 ต่อวงจร

รูปการต่อวงจร 1 – ด้านข้าง



รูปการต่อวงจร 2 – ด้านบน



## 2.3 นำ Code อัปโหลดลง ESP32

### Code - Main

```
#include <WiFi.h>
#include <Wire.h>
#include <ThingsBoard.h>
#include "SHTC3.h"
#define WIFI_NAME "your-wifi-name"
#define WIFI_PASS "your-wifi-pass"
#define SERVER "demo.thingsboard.io"
```

```
#define DEVICE_TOKEN "your-device-token"

#define COUNT_OF(x) ((sizeof(x)/sizeof(0[x])) / ((size_t)(!(sizeof(x) % sizeof(0[x])))))

#define LED_BLINK 2
bool LED_Blink_Status = 0;
SHTC3 sht(Wire);
WiFiClient esp32Client;
byte LED[] = {19, 21, 22, 23};
bool LED_Status[] = {0, 0, 0, 0};
int loopDaly = 20;
int BlinkLEDdelay = 500;
int counterBlinkLEDdelay = 0;
int sendDataDelay = 5000;
int counterSendData = 0;
ThingsBoard tb(esp32Client);
#include "_ThingBoardRPC.h"
bool subscribeStatus = 0;
char StringEcho[] = "stsLED_1";

void setup() {
    Serial.begin(115200);
    Wire.begin(15, 18);
    pinMode(LED_BLINK, OUTPUT);
    for(int i = 0 ; i < 4 ; i++) {
        pinMode(LED[i], OUTPUT);
    }
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("\nWiFi Connected");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
}

void reconnectTB() {
    if(!tb.connect(SERVER, DEVICE_TOKEN)) {
        Serial.println("Failed to connect to ThingsBoard server");
    }
}
```

```
else {
    Serial.println("Connected to ThingsBoard server");
}

void reconnectWiFi() {
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    randomSeed(millis());
    Serial.println("\nWiFi Connected");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
}

void subscribeForRPC() {
    Serial.println("Subscribing for RPC...");
    if(!tb.RPC_Subscribe(callbacks, COUNT_OF(callbacks))) {
        Serial.println("Failed to subscribing for RPC");
    }
    else {
        Serial.println("Subscribed for RPC");
        subscribeStatus = 1;
    }
}

void loop() {
    if(WiFi.status() != WL_CONNECTED) reconnectWiFi();
    if(!tb.connected()) {
        subscribeStatus = 0;
        reconnectTB();
    }
    if(!subscribeStatus) subscribeForRPC();
    tb.loop();

    if(counterBlinkLEDdelay > BlinkLEDdelay) {
        digitalWrite(LED_BLINK, LED_Blink_Status);
        LED_Blink_Status = 1 - LED_Blink_Status;
        counterBlinkLEDdelay = 0;
    }
}
```

```

}

if(counterSendData > sendDataDelay) {
    sht.begin();
    sht.sample();
    float t = sht.readTempC(); //random(00, 50); //
    float h = sht.readHumidity(); //random(51, 99); //
    if(t >= -40 && t <= 125) {
        tb.sendTelemetryFloat("temperature", t);
        tb.sendTelemetryFloat("humidity", h);
        Serial.println("temperature: " + String(t) + ", humidity: " + String(h));
    }
    Serial.print("LED=");
    for(size_t i = 0 ; i < COUNT_OF(LED); i++) {
        StringEcho[7] = 0x30 + i;
        tb.sendTelemetryInt(StringEcho, LED_Status[i]);
    }
    Serial.print(LED_Status[i]);
}
Serial.println();
counterSendData = 0;
}
delay(loopDaly);
counterSendData += loopDaly;
counterBlinkLEDdelay += loopDaly;
}

```

Code - Main กับ Code - \_ThingBoardRPC ต้องอยู่ใน Folder เดียวกัน

#### Code - \_ThingBoardRPC

```

// file name: _ThingBoardRPC.h
RPC_Response setDelay(const RPC_Data &data) {
    Serial.println("Received request set delay");
    BlinkLEDdelay = data;
    Serial.print("Set new delay: ");
    Serial.print(BlinkLEDdelay);
    return RPC_Response(NULL, BlinkLEDdelay);
}

```

```
RPC_Response getDelay(const RPC_Data &data) {
    Serial.println("Received request get delay");
    return RPC_Response(NULL, BlinkLEDdelay);
}
```

```
RPC_Response setLED(const RPC_Data &data) {
    Serial.println("Received request set LED");
    byte pin = data["pin"];
    bool enable = data["enable"];
    if(pin < COUNT_OF(LED)) {
        Serial.print("Setting LED" );
        Serial.print(pin);
        Serial.print("to state");
        Serial.print(LED_Status[pin]);
        LED_Status[pin] = 1 - LED_Status[pin];
        digitalWrite(LED[pin], LED_Status[pin]);
    }
    return RPC_Response(data["pin"], (bool)data["enable"]);
}
```

```
RPC_Callback callbacks[] = {
    {"setValue", setDelay},
    {"getValue", getDelay},
    {"setGpioStatus", setLED}
};
```

Code - Main กับ Code - \_ThingBoardRPC ต้องอยู่ใน Folder เดียวกัน

## 2.4 ผลลัพธ์หลังจากส่งข้อมูลไปยัง Device จะได้ตัวแปร เพื่อนำไปใช้สร้าง Dashboard ต่อไป

The screenshot shows the ThingsBoard interface. On the left is a sidebar with various navigation options. The main area is divided into two sections: 'Devices' on the left and 'Device details' on the right. In the 'Devices' section, there is a table listing several devices, including M2-Q102, M2-Q104, M2-Q103, and OfflineDevice. The 'Device details' section for M2-Q102 shows its latest telemetry data. A red box highlights the table under 'Latest telemetry', which contains the following data:

Last update time	Key ↑	Value
2022-05-03 00:11:19	humidity	66.68701172
2022-05-03 00:11:19	stsLED_0	0
2022-05-03 00:11:19	stsLED_1	0
2022-05-03 00:11:19	stsLED_2	0
2022-05-03 00:11:19	stsLED_3	0
2022-05-03 00:11:18	temperature	26.68148499

### 3. สร้าง Dashboard

#### 3.1 ทำการสร้าง Dashboard ใหม่ โดยไปที่ Dashboards -> + ->Create new dashboard

The screenshot shows the ThingsBoard interface. On the left is a dark blue sidebar with various menu items: Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge instances, Edge management, Widgets Library, and Dashboards. The 'Dashboards' item is highlighted with a red box. The main content area is titled 'Dashboards' and lists several existing dashboards with columns for Created time, Title, Assigned to customers, and various actions like edit, delete, and download. In the top right corner of this list, there is a button labeled '+ Create new dashboard' with a red box around it.

#### 3.2 สร้าง Dashboard โดยตั้งค่าตามนี้ จากนั้นกดปุ่ม Add

The screenshot shows the 'Add Dashboard' dialog box. At the top, it says 'Add Dashboard'. Below that, there's a 'Title \*' field containing 'M2-Q102-ThingsBoard-Data-Monitor-and-Control'. There's also a 'Description' field which is empty. Under 'Mobile application settings', there's a 'Dashboard image' section with a placeholder 'No image selected' and a dashed box for uploading an image, with the text 'Drop an image or click to select a file to upload'. Below this, there's a note 'Maximum upload file size: 512.0 KB'. There are two checkboxes: 'Hide dashboard in mobile application' and 'Dashboard order in mobile application'. At the bottom right of the dialog is a blue 'Add' button.

#### 3.3 กดเข้าไปที่ Dashboard ที่สร้างขึ้นมาและกด Open dashboard

The screenshot shows the 'Dashboards' list again. The newly created dashboard 'M2-Q102-ThingsBoard-Data-Monitor-and-Control' is listed. To its right, a modal window titled 'M2-Q102-ThingsBoard-Data-Monitor-and-Control' is open. Inside this window, there's a 'Dashboard details' tab. At the bottom of this tab, there are four buttons: 'Details', 'Audit Logs', 'Open dashboard' (which is highlighted with a red box), 'Export dashboard', 'Make dashboard public', and 'Manage assigned customers'. The title of the dashboard in the modal is 'M2-Q102-ThingsBoard-Data-Monitor-and-Control'.

3.4 จากนั้นกดที่รูป ด้านล่างขวาเพื่อเข้าสู่โหมดแก้ไขจากนั้นกดไปที่ -> Add alias

Device-M2-Q102

Add alias X

Alias name *	Resolve as multiple entities
Device-M2-Q102	<input checked="" type="checkbox"/>
Filter type *	
Single entity	
Type *	Device *
Device	M2-Q102

Cancel
Add

เมื่อ Add alias ทั้งหมดแล้วจะได้ผลลัพธ์เป็นแบบนี้ จากนั้นให้กด Save

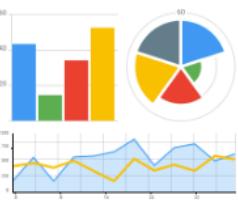
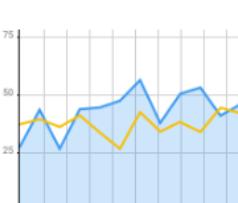
Entity aliases X

Alias name	Entity filter	Resolve as multiple entities
1. Device-M2-Q102	One device	<input checked="" type="checkbox"/>

Add alias Cancel Save

3.5 ทำการสร้าง Widget Chart โดยเมื่อออยู่ในโหมดแก้ไขให้กดไปที่ จากนั้นเลือก

Charts -> Timeseries Line Chart

 <p><b>Charts</b> System Display timeseries data using customizable line and bar charts. Use various pie charts to display latest values.</p>	 <p><b>Timeseries Line Chart</b> Time series Displays changes to timeseries data over time. For example, temperature or humidity readings.</p>
--	--

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

ที่ tab Settings ในส่วนของ Title ให้ตั้งชื่อเป็น Timeseries Line Chart จากนั้นกดปุ่ม Add

3.6 สร้าง 4 Widget Analogue gauges โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



Digital gauges -> Digital vertical bar

	<p><b>Digital gauges</b> System</p> <p>Display temperature, humidity, speed, and other latest values on various digital gauge widgets.</p>	<p><b>Digital vertical bar</b> Latest values</p> <p>Preconfigured gauge to display any value reading as a vertical bar. Allows to configure value range, gradient colors and other settings.</p>
--	--	--

ที่ tab Data ในส่วนของ Datasources ของแต่ละ Widget ให้กด + Add และตั้งค่าตามนี้

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Device-M2-Q102

Entity alias \*      stsLED\_0:stsLED\_0

Filter

Data settings

Cancel      Add

Widget LED0

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Device-M2-Q102

Entity alias \*      stsLED\_1:stsLED\_1

Filter

Data settings

Cancel      Add

Widget LED1

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Device-M2-Q102     

Entity alias \*  
Device-M2-Q102

Filter

Maximum 1 timeseries/attribute is allowed.

Data settings

Cancel      Add

Widget LED2

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Device-M2-Q102     

Entity alias \*  
Device-M2-Q102

Filter

Maximum 1 timeseries/attribute is allowed.

Data settings

Cancel      Add

Widget LED3

ที่ tab Advanced ของทั้ง 4 Widget ตั้งค่าให้เหมือนกันตามนี้

Add Widget: Digital vertical bar

Advanced tab settings:

- Minimum value: 0
- Maximum value: 1
- Gauge type: Vertical bar
- Angle to start from when in donut mode: 90
- Neon glow effect brightness, (0-100), 0 - disable effect: 20
- Thickness of the stripes, 0 - no stripes: 1.2

Buttons: Cancel, Add

ที่ tab Advanced ของแต่ละ Widget ตั้งค่าตามนี้ จากนั้นกด Add

Add Widget: Digital vertical bar

Advanced tab settings for Widget LED0:

- Background color of the gauge element:  rgb(23, 26, 28)
- Use precise value for the color indicator
- Colors of indicator, from lower to upper
  - Color:  rgb(244, 67, 54)

Buttons: Cancel, Add

Widget LED0

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Background color of the gauge element  
● `rgb(23, 26, 28)`

Use precise value for the color indicator

Colors of indicator, from lower to upper

1. X  
Color  
● `rgb(0, 255, 15)`

+ New

Cancel   Add

Widget LED1

Add Widget: Digital vertical bar

Data      Settings      Advanced      Actions

Background color of the gauge element  
● `rgb(23, 26, 28)`

Use precise value for the color indicator

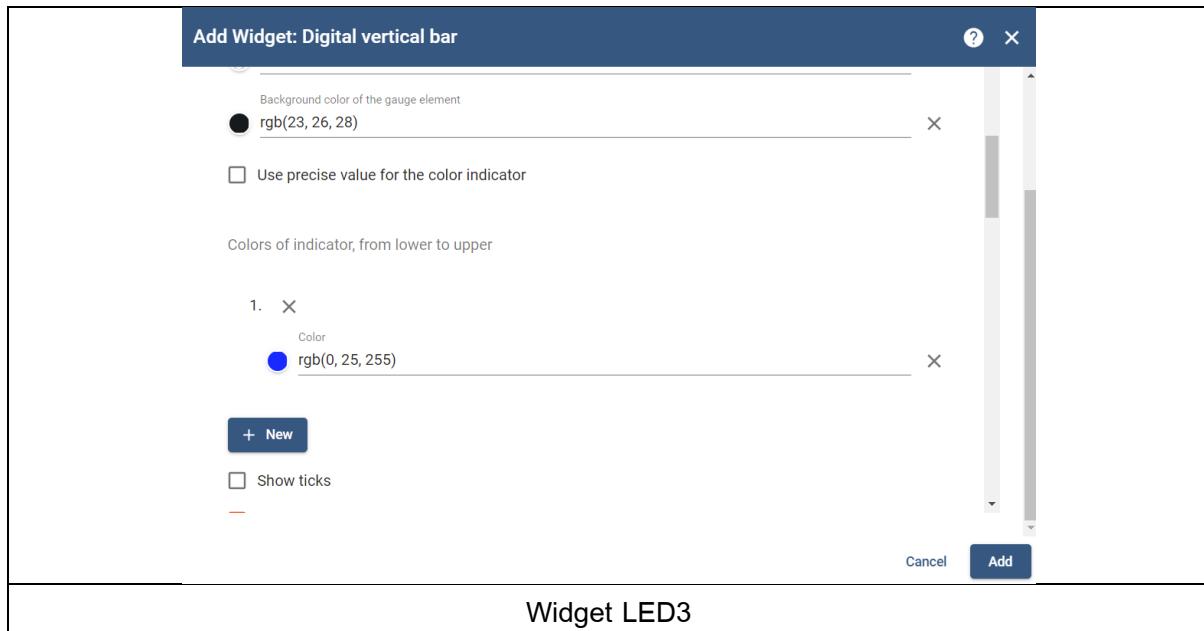
Colors of indicator, from lower to upper

1. X  
Color  
● `rgb(237, 255, 0)`

+ New

Cancel   Add

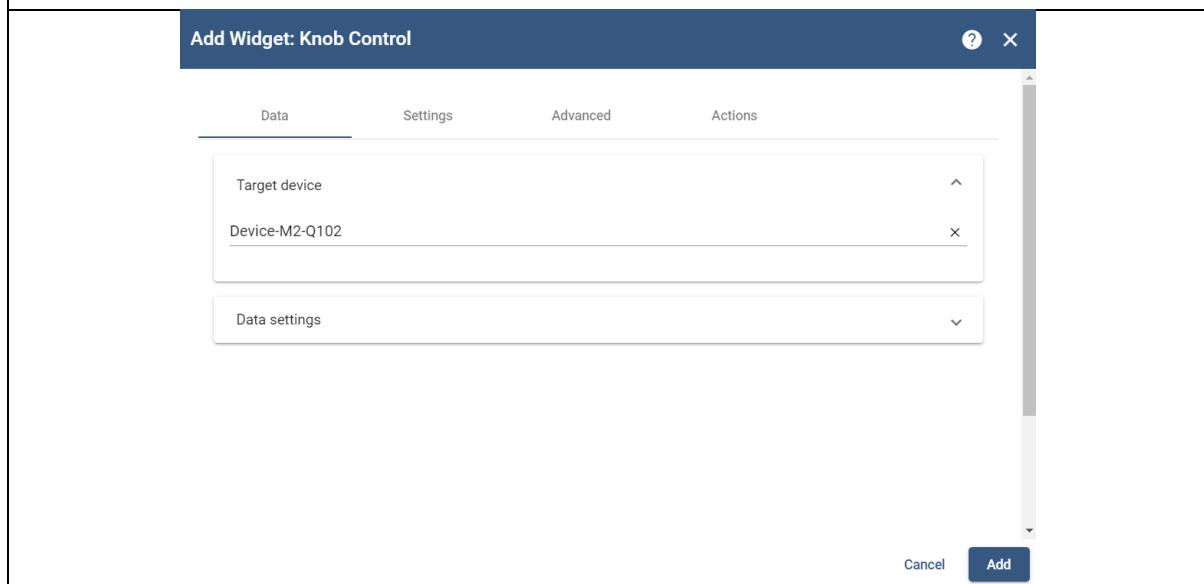
Widget LED2



3.7 ทำการสร้าง Control widget โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่ จากนั้นเลือก

Control widgets -> Knob Control	
<b>Control widgets</b> System Send commands to devices. 	<b>Knob Control</b> Control widget Sends the command to device with specified value each time user changes the value. Uses 'setValue' and 'getValue' RPC calls by default. The name of the RPC calls is configurable in advanced settings.

ที่ tab Data ในส่วนของ Target device ให้ตั้งค่าตามนี้



ที่ tab Advanced ให้ตั้งค่าตามนี้ จากนั้นกด Add

Add Widget: Knob Control

Advanced

Minimum value \*

0

Maximum value \*

1000

Initial value

250

Knob title

Knob control

Get value method \*

getValue

Set value method \*

setValue

RPC request timeout \*

500

Cancel Add

3.8 ทำการสร้าง Control widget โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



GPIO widgets -> Basic GPIO Control

<b>GPIO widgets</b> System Visualization and control of GPIO state for target devices.	<b>Basic GPIO Control</b> Control widget <small>Allows to change state of the GPIO for target device using RPC commands. Requires handling of the RPC commands in the device firmware. Uses 'getGpioStatus' and 'setGpioStatus' RPC calls</small>
--	---

ที่ tab Data ในส่วนของ Target device ให้ตั้งค่าตามนี้

Add Widget: Basic GPIO Control

Data

Target device

Device-M2-Q103

Data settings

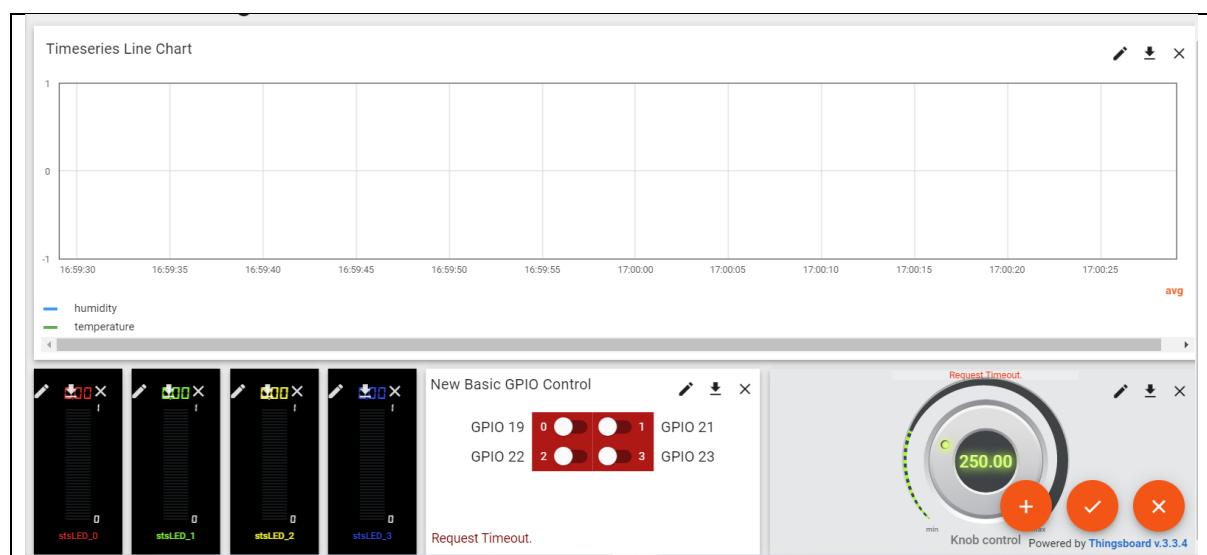
Cancel Add

ที่ tab Advanced ให้เพิ่ม Gpio switch 4 อันตามการตั้งค่าตามนี้ จากนั้นกด Add

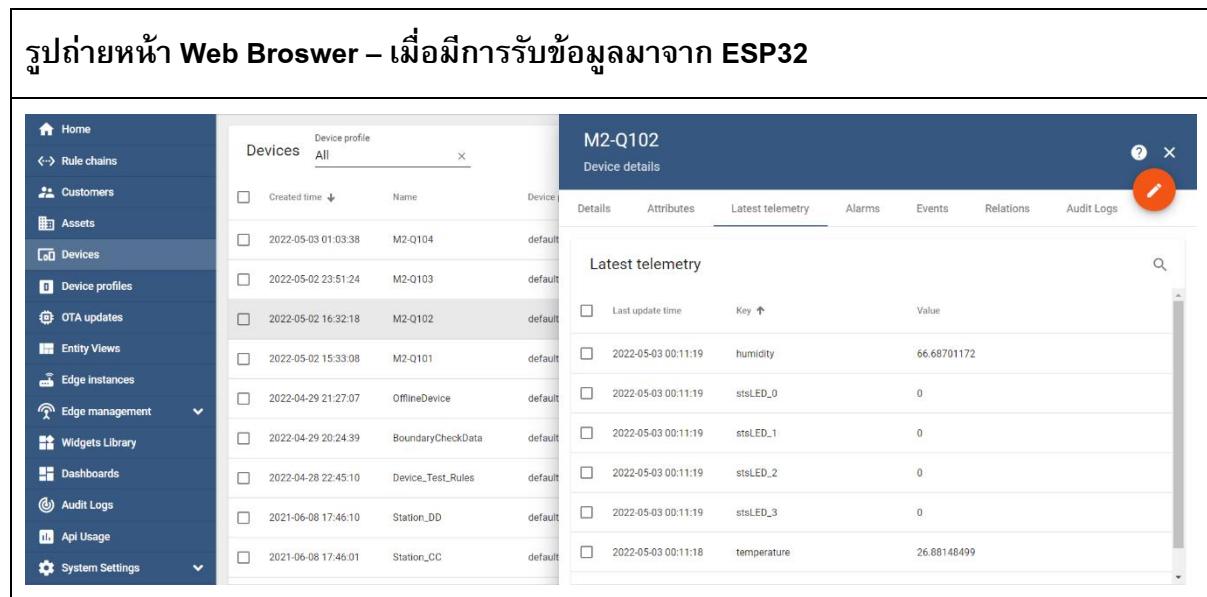
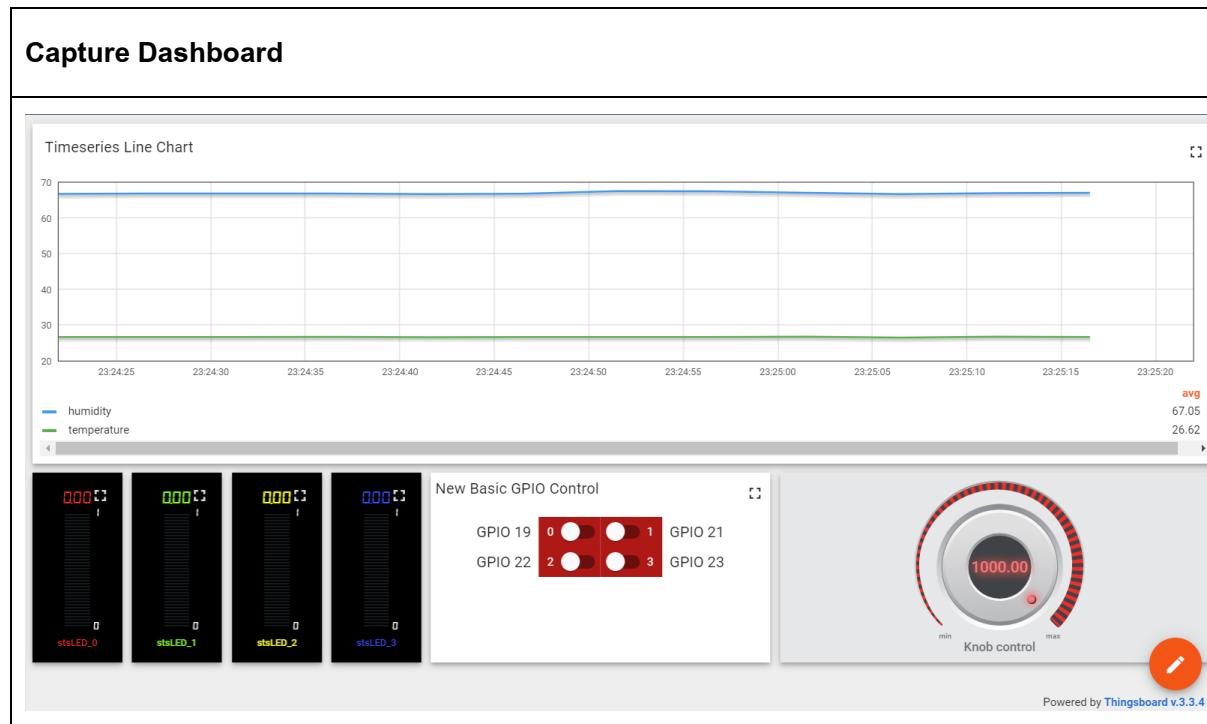
	Pin: 0 Label: GPIO 19 Row: 0, Column: 0
	Pin: 1 Label: GPIO 21 Row: 0, Column: 1
	Pin: 2 Label: GPIO 22 Row: 1, Column: 0
	Pin: 3 Label: GPIO 23 Row: 1, Column: 1

เพิ่ม Gpio switch ใหม่ด้วยการกด + New

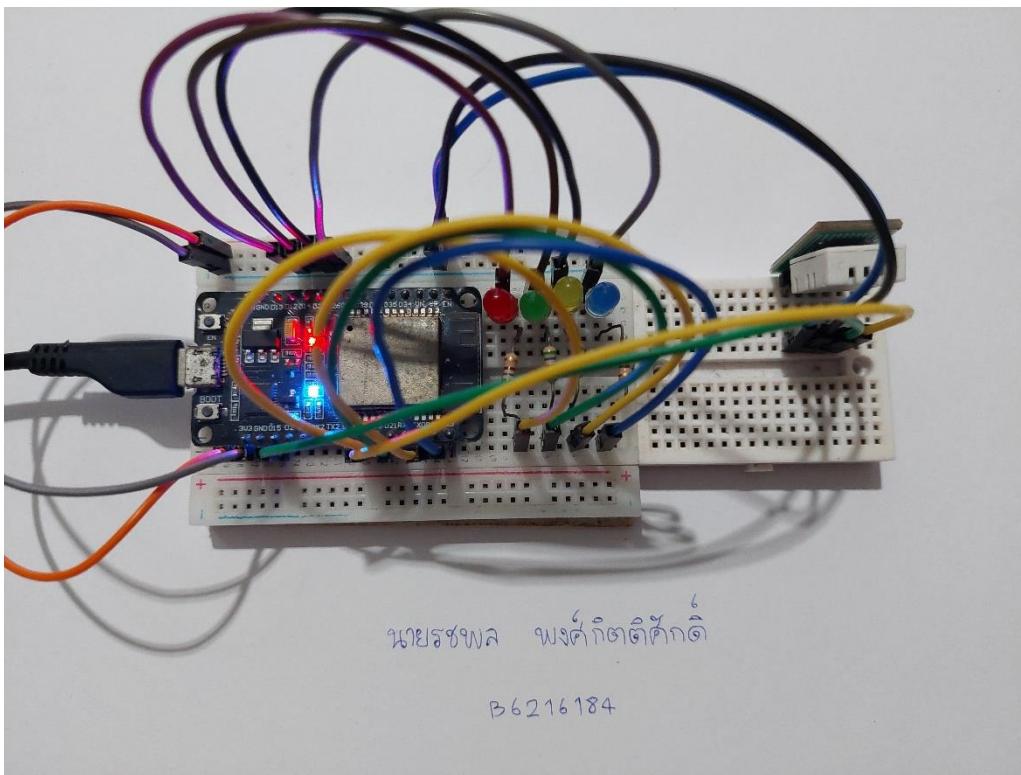
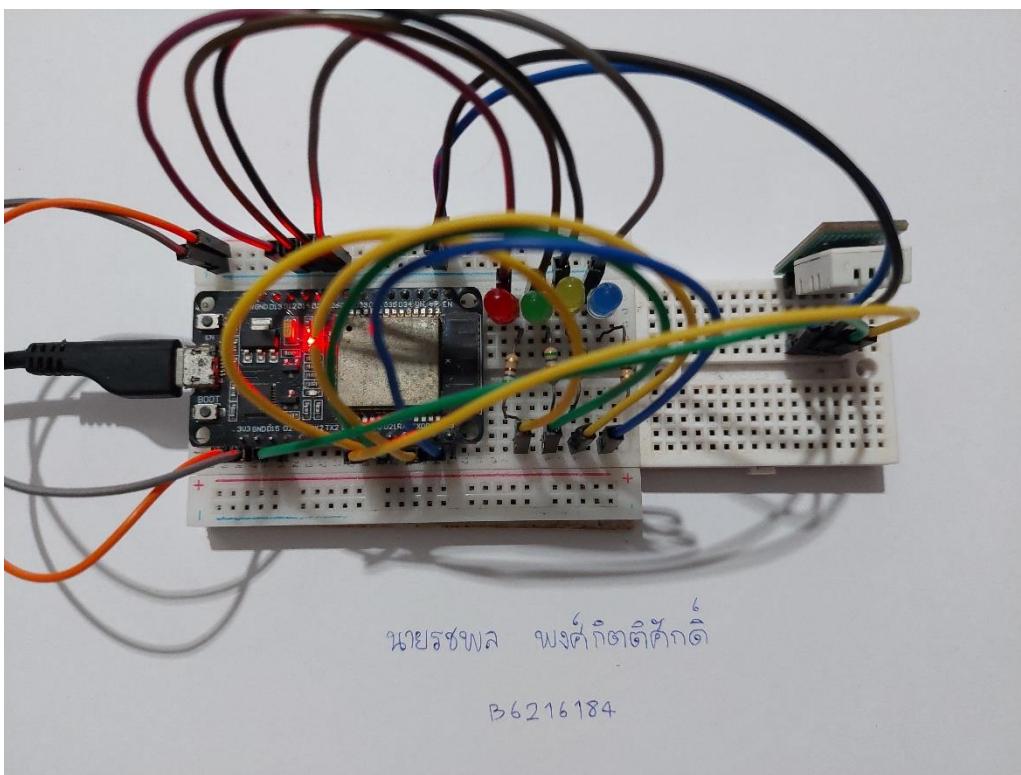
3.9 ปรับหน้าตา Dashboard ให้เป็นตามนี้ จากนั้นกดที่ เพื่อบันทึกการแก้ไข



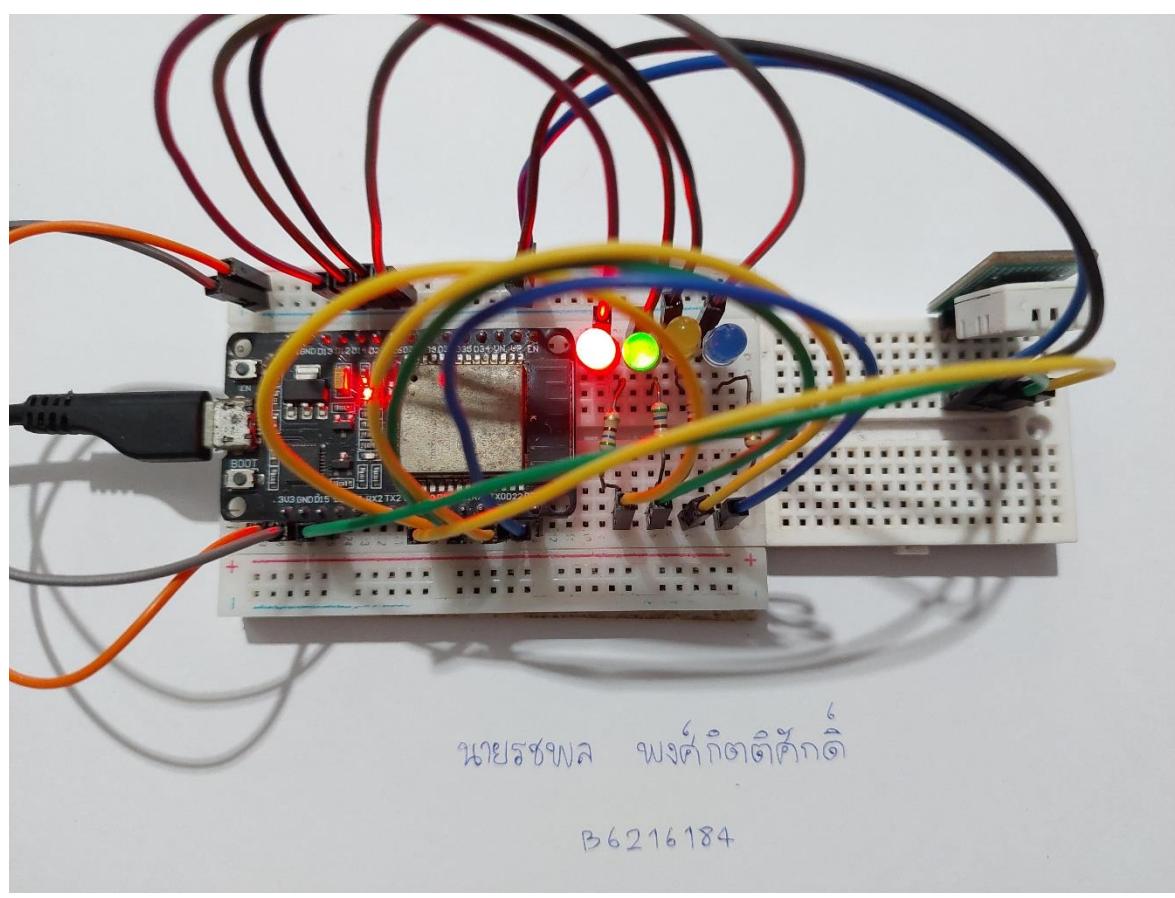
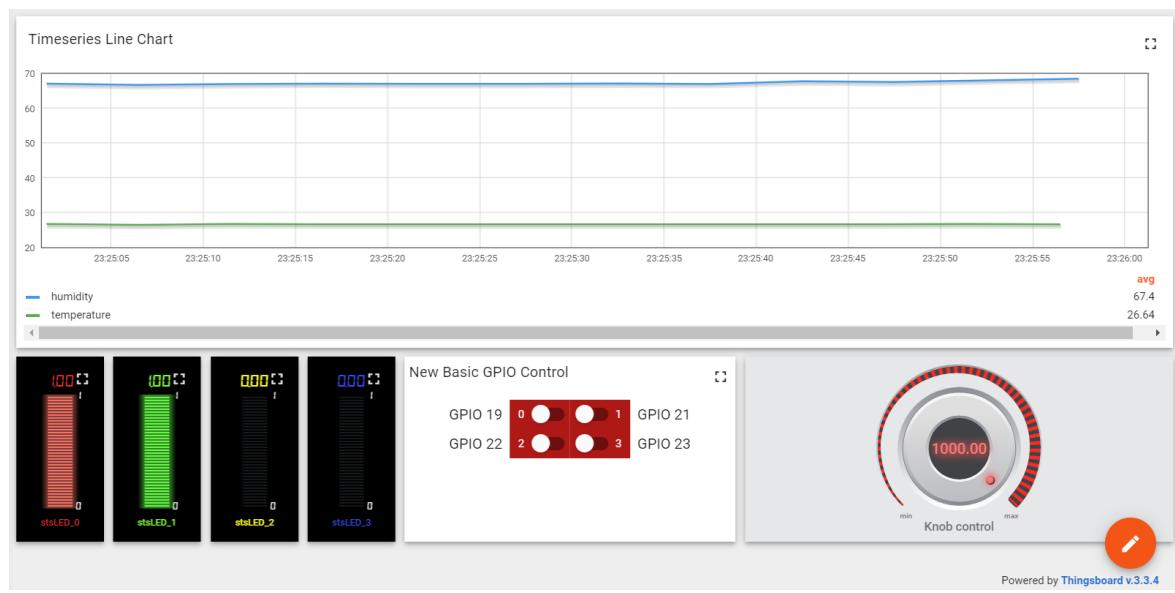
## 4. ພລລັບ



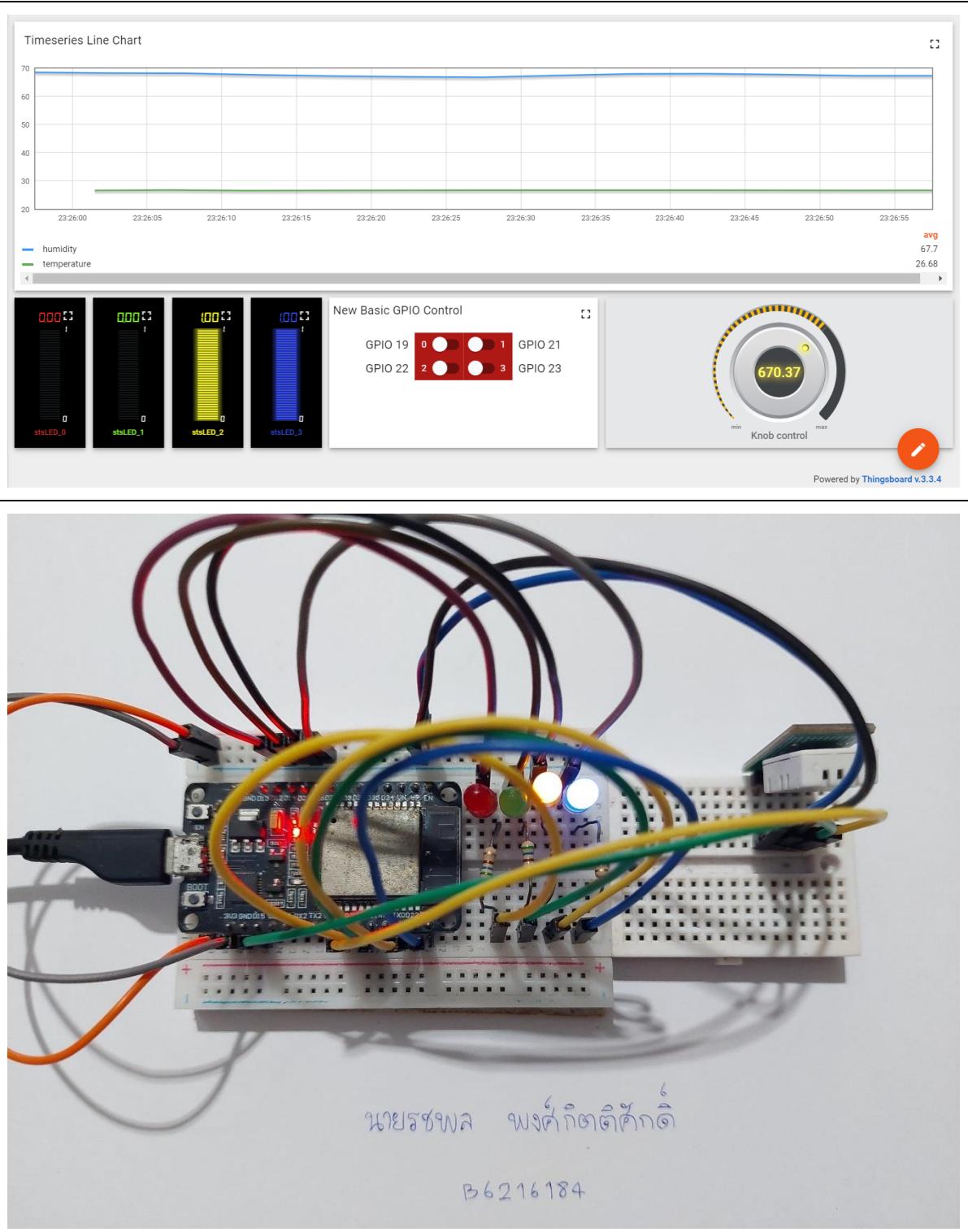
รูปการทดสอบ 3 – LED กระพริบตามความเร็วที่ปรับใน Knob Control



## รูปการทดสอบ 2 – เปิด/ปิด LED ทั้ง 4 หลอด



### รูปการทดสอบ 3 – เปิด/ปิด LED ทั้ง 4 หลอด



### Quiz\_103 – ThingsBoard Data Monitor and control with MQTT Protocol

- Mission 3/4: ให้ใช้ MQTT กับ ThingsBoard
  - ปรับปรุงเพื่อให้ทำงานควบคุมการ On/Off - 4 LED
  - เพิ่มเติม คือ ทดสอบส่งข้อมูล 1 ค่าแบบสุ่มระหว่าง 00 – 50 ไปแสดงที่ Dashboard ด้วย ได้หรือไม่ ทำอย่างไรบ้างให้อธิบาย {Read <https://thingsboard.io/docs/user-guide/device-profiles/> }

#### 1. สร้าง Device

1.1 ทำการสร้าง Device ใหม่ โดยไปที่ Devices -> + -> Add new device

1.2 สร้าง Device โดยในช่อง Name ตั้งชื่อตามนี้ จากนั้นกดปุ่ม Add

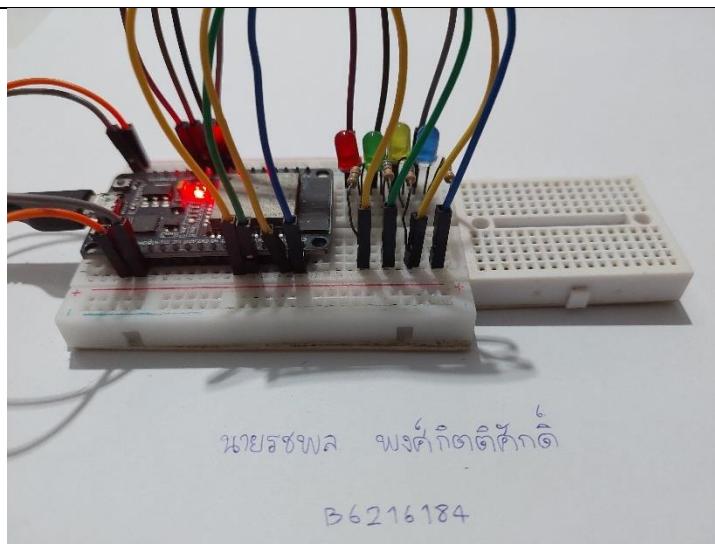
## 2. ส่งข้อมูลไปยัง Device

### 2.1 Copy Access Token ของ Device ที่สร้าง ไปที่ Devices -> ชื่อ Device -> Copy access token

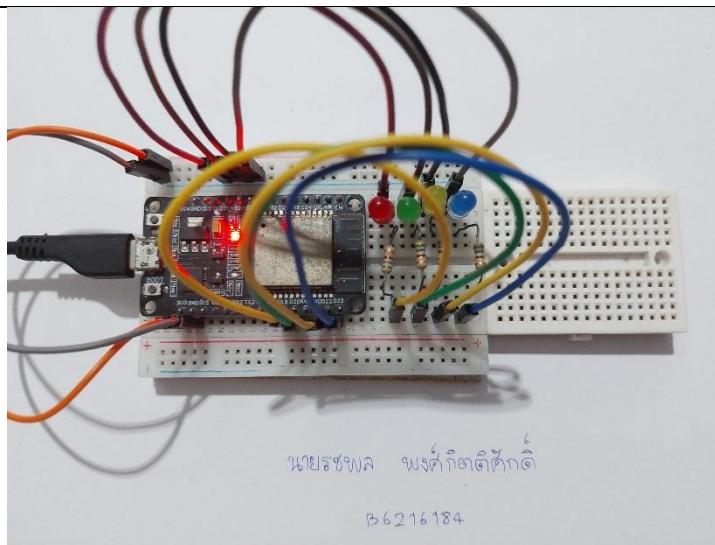
Access Token ก็คือ DEVICE\_TOKEN ใน Code – Main

### 2.2 ต่อวงจร

รูปการต่อวงจร 1 – ด้านข้าง



รูปการต่อวงจร 2 – ด้านบน



## 2.3 นำ Code อัปโหลดลง ESP32

**Code – Main**

```
#include <WiFi.h>
#include <Wire.h>
#include <PubSubClient.h>
#include <ArduinoJson.h>
#include "SHTC3.h"

#define WIFI_NAME "your-wifi-name"
#define WIFI_PASS "your-wifi-pass"
#define SERVER "demo.thingsboard.io"
#define DEVICE_NAME "M2-Q103"
#define DEVICE_TOKEN "your-device-token"

SHTC3 sht(Wire);
WiFiClient esp32Client;
PubSubClient client(esp32Client);
byte LED[] = {19, 21, 22, 23};
bool LED_Status[] = {0, 0, 0, 0};
#include "_HandOnMQTT.h"

void setup() {
    Serial.begin(115200);
    Wire.begin(15, 18);
    for(int i = 0 ; i < 4 ; i++) {
        pinMode(LED[i], OUTPUT);
    }
    WiFi.begin(WIFI_NAME, WIFI_PASS);
    while(WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    randomSeed(millis());
    Serial.println("\nWiFi Connected");
    Serial.print("IP Address: ");
    Serial.println(WiFi.localIP());
    client.setServer(SERVER, 1883);
    client.setCallback(on_message);
    xTaskCreatePinnedToCore(
        sendRandomValue, // Task function
        "Send Random Value", 1000, NULL, 1, &xHandle);
}
```

```

"sendRandomValue", // name of task
2000, // Stack size of task
NULL, // parameter of task
1, // priority od the task
NULL, // Task handle to keep track of created task
1 // pin task to core 1
);

}

void sendRandomValue(void *pvParameterrs) {
while(true) {
client.publish("v1/devices/me/telemetry", getRandomValue().c_str());
vTaskDelay(5000/portTICK_PERIOD_MS);
}
}

void reconnectTB() {
while(!client.connected()) {
if(client.connect(DEVICE_NAME, DEVICE_TOKEN, NULL)) {
Serial.println(" [DONE] ");
client.subscribe("v1/devices/me/rpc/request/+");
client.publish("v1/devices/me/attributes", get_gpio_status().c_str());
}
else {
Serial.print("[FAILED] [ rc = ");
Serial.print(client.state());
Serial.println(" : retrying in 5 seconds]");
delay(5000);
}
}
}

void reconnectWiFi() {
WiFi.begin(WIFI_NAME, WIFI_PASS);
while(WiFi.status() != WL_CONNECTED) {
delay(500);
Serial.print(".");
}
randomSeed(millis());
Serial.println("\nWiFi Connected");
}

```

```

Serial.print("IP Address: ");
Serial.println(WiFi.localIP());
}

void loop() {
    if(WiFi.status() != WL_CONNECTED) reconnectWiFi();
    if(!client.connected()) reconnectTB();
    client.loop();
}

```

Code - Main กับ Code - \_HandOnMQTT ต้องอยู่ใน Folder เดียวกัน

#### Code - \_HandOnMQTT

```

// file name: _HandOnMQTT.h

String get_gpio_status() {
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject & data = jsonBuffer.createObject();
    data[String(LED[0])] = LED_Status[0];
    data[String(LED[1])] = LED_Status[1];
    data[String(LED[2])] = LED_Status[2];
    data[String(LED[3])] = LED_Status[3];
    char payload[256];
    data.printTo(payload, sizeof(payload));
    String strPayload = String(payload);
    Serial.print("Get GPIO Status: ");
    Serial.println(strPayload);
    return strPayload;
}

String getRandomValue() {
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject & data = jsonBuffer.createObject();
    data["randomValue"] = random(0, 50);
    char payload[256];
    data.printTo(payload, sizeof(payload));
    String strPayload = String(payload);
    Serial.print("Get Random Value: ");
    Serial.println(strPayload);
    return strPayload;
}

```

```

void set_gpio_status(int pin, boolean enabled) {
    if (pin == LED[0]) {
        LED_Status[0] = 1 - LED_Status[0];
        digitalWrite(LED[0], LED_Status[0]);
    }
    if (pin == LED[1]) {
        LED_Status[1] = 1 - LED_Status[1];
        digitalWrite(LED[1], LED_Status[1]);
    }
    if (pin == LED[2]) {
        LED_Status[2] = 1 - LED_Status[2];
        digitalWrite(LED[2], LED_Status[2]);
    }
    if (pin == LED[3]) {
        LED_Status[3] = 1 - LED_Status[3];
        digitalWrite(LED[3], LED_Status[3]);
    }
}

void on_message(const char* topic, byte* payload, unsigned int length) {
    Serial.println("\nOn message");
    char json[length + 1];
    strncpy (json, (char*)payload, length);
    json[length] = '\0';
    Serial.print("Topic: "); Serial.println(topic);
    Serial.print("Message: "); Serial.println(json);
    // Decode JSON request
    StaticJsonBuffer<200> jsonBuffer;
    JsonObject& data = jsonBuffer.parseObject((char*)json);
    if (!data.success()) {
        Serial.println("parseObject() failed");
        return;
    }

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

    if (methodName.equals("getGpioStatus")) {
        String responseTopic = String(topic);
        responseTopic.replace("request", "response");
        client.publish(responseTopic.c_str(), get_gpio_status().c_str());
    }
}

```

}

```

if (methodName.equals("setGpioStatus")) {
    set_gpio_status(data["params"]["pin"], data["params"]["enabled"]);
    String responseTopic = String(topic);
    responseTopic.replace("request", "response");
    client.publish(responseTopic.c_str(), get_gpio_status().c_str());
    client.publish("v1/devices/me/attributes", get_gpio_status().c_str());
}
}

```

**Code – Main กับ Code – \_HandOnMQTT** ต้องอยู่ใน Folder เดียวกัน

2.4 ผลลัพธ์หลังจากส่งข้อมูลไปยัง Device จะได้รับแพร เพื่อนำไปใช้สร้าง Dashboard ต่อไป

ข้อมูลที่ tab Attributes

Last update time	Key ↑	Value
2022-05-03 00:55:15	19	false
2022-05-03 00:55:15	21	true
2022-05-03 00:55:15	22	true
2022-05-03 00:55:15	23	false

ข้อมูลที่ tab Last telemetry

Last update time	Key ↑	Value
2022-05-03 00:11:19	humidity	66.68701172
2022-05-03 00:11:19	stsLED_0	0
2022-05-03 00:11:19	stsLED_1	0
2022-05-03 00:11:19	stsLED_2	0
2022-05-03 00:11:19	stsLED_3	0
2022-05-03 00:11:18	temperature	26.88146499

### 3. สร้าง Dashboard

3.1 ทำการสร้าง Dashboard ใหม่ โดยไปที่ Dashboards -> + ->Create new dashboard

3.2 สร้าง Dashboard โดยตั้งค่าตามนี้ จากนั้นกดปุ่ม Add

3.3 กดเข้าไปที่ Dashboard ที่สร้างขึ้นมาและกด Open dashboard

3.4 จากนั้นกดที่รูป ด้านล่างขวาเพื่อเข้าสู่โหมดแก้ไขจากนั้นกดไปที่ -> Add alias

Device-M2-Q103

Add alias X

Alias name \*  Resolve as multiple entities

Filter type \*  ▼

Type *	Device *
<input style="border: none; border-bottom: 1px solid black; padding: 2px 5px; margin-right: 10px;" type="button" value="Device"/>	<input type="text" value="M2-Q103"/> <span>X</span>

Cancel Add

เมื่อ Add alias ทั้งหมดแล้วจะได้ผลลัพธ์เป็นแบบนี้ จากนั้นให้กด Save

Entity aliases

X

1.	Alias name	Entity filter	Resolve as multiple entities
1.	<input type="text" value="Device-M2-Q103"/>	One device	<input checked="" type="checkbox"/>

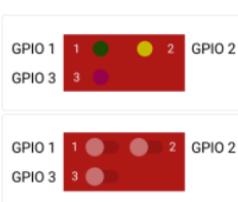
Add alias Cancel Save

3.5 ทำการสร้าง Control widget โดยเมื่อออยู่ในโหมดแก้ไขให้กดไปที่ จากนั้นเลือก

GPIO widgets -> Basic GPIO Control

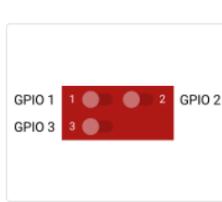
**GPIO widgets**  
System

Visualization and control of GPIO state for target devices.

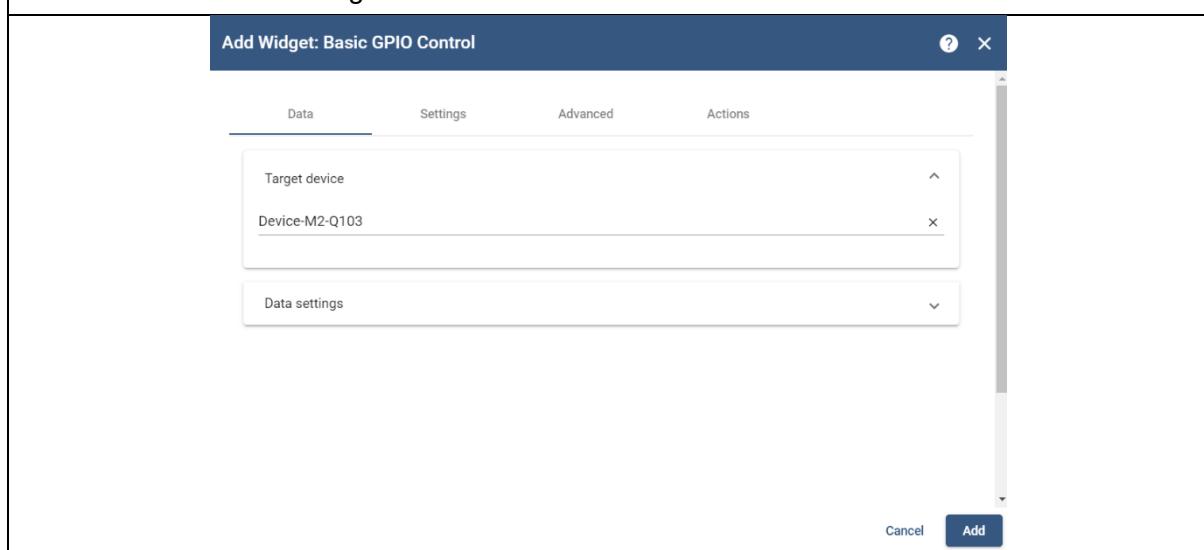


**Basic GPIO Control**  
Control widget

Allows to change state of the GPIO for target device using RPC commands. Requires handling of the RPC commands in the device firmware. Uses 'getGpioStatus' and 'setGpioStatus' RPC calls



ที่ tab Data ในส่วนของ Target device ให้ตั้งค่าตามนี้



ที่ tab Advanced ให้เพิ่ม Gpio switch 4 อันตามการตั้งค่าตามนี้ จากนั้นกด Add

<p>Add Widget: Basic GPIO Control</p> <p>Data      Settings      Advanced      Actions</p> <p>Gpio switch</p> <p>Pin * 4</p> <p>Label * GPIO 23</p> <p>Row * 1</p> <p>Column * 1</p> <p>+ New</p>	<p>Pin: 1 Label: GPIO 19 Row: 0, Column: 0</p> <p>Pin: 2 Label: GPIO 21 Row: 0, Column: 1</p> <p>Pin: 3 Label: GPIO 22 Row: 1, Column: 0</p> <p>Pin: 4 Label: GPIO 23 Row: 1, Column: 1</p>
---	---

เพิ่ม Gpio switch ใหม่ด้วยการกด + New

3.5 ทำการสร้าง Control widget โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



GPIO widgets -> Basic GPIO Control

<p><b>GPIO widgets</b> System</p> <p>Visualization and control of GPIO state for target devices.</p>	<p><b>Basic GPIO Panel</b> Latest values</p> <p>Allows to display state of the GPIO for target device using latest attribute values. You should set the label of the selected data key to GPIO pin number (e.g. '1') and use boolean values for widget to display the data.</p>
--	---

ที่ tab Data ในส่วนของ Datasource ให้ตั้งค่าตามนี้

Add Widget: Basic GPIO Panel

Data      Settings      Advanced      Actions

Datasources

Type	Parameters
= 1. Entity	Entity alias * Device-M2-Q103 Filter

+ Add      Cancel      Add

ที่ tab Advanced ให้เพิ่ม Gpio switch 4 อันตามการตั้งค่าตามนี้ จากนั้นกด Add

Pin: 1	Label: GPIO 19
Row: 0, Column: 0	Color: แดง
Pin: 2	Label: GPIO 21
Row: 0, Column: 1	Color: เขียว
Pin: 3	Label: GPIO 22
Row: 1, Column: 0	Color: เหลือง
Pin: 4	Label: GPIO 23
Row: 1, Column: 1	Color: นำเงิน

Add Widget: Basic GPIO Panel

Data      Settings      Advanced      Actions

4. X

Pin \*  
4

Label \*  
GPIO 23

Row \*  
1

Column \*  
1

Color \*  
rgb(18, 0, 255)

+ New      Cancel      Add

เพิ่ม Gpio switch ใหม่ด้วยการกด + New

## 3.6 ทำการสร้าง 5 Widget Card โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



**Cards -> Simple card**

<b>Entities</b> Timeseries table Name ↓    Charged, %    Time ↓    Humidity, % Arduino    99.6    10:48:15    61.2 Linkit SE    37.2    10:48:14    64.5  Your HTML code here	<b>Cards</b> <b>System</b> Tables and cards to display latest and historical values for multiple entities simultaneously.	<b>Simple card</b> Latest values  Designed to display single value of the selected attribute or timeseries data. Widget styles are customizable.
---	---	---

ที่ tab Data ในส่วนของ Datasources ของ Widget Card แตะลําอัน ให้กด + Add และตั้งค่าตามนี้  
จากนั้นกด Add

**Add Widget: Simple card**

Data      Settings      Advanced      Actions  <b>Datasources</b> Maximum 1 datasource is allowed. Type      Parameters = 1. Entity      Entity alias * Device-M2-Q103      = ⚡ 19:19 ✎ ✖ Filter  Data settings	Cancel      Add		
<p><b>Add Widget: Simple card</b></p> <table border="1"> <tr> <td style="vertical-align: top;">           Data      Settings      Advanced      Actions    <b>Datasources</b>            Maximum 1 datasource is allowed.            Type      Parameters            = 1. Entity      Entity alias * Device-M2-Q103      = ⚡ 21:21 ✎ ✖            Filter              Data settings         </td> <td style="vertical-align: top;">           Cancel      Add         </td> </tr> </table>		Data      Settings      Advanced      Actions  <b>Datasources</b> Maximum 1 datasource is allowed. Type      Parameters = 1. Entity      Entity alias * Device-M2-Q103      = ⚡ 21:21 ✎ ✖ Filter  Data settings	Cancel      Add
Data      Settings      Advanced      Actions  <b>Datasources</b> Maximum 1 datasource is allowed. Type      Parameters = 1. Entity      Entity alias * Device-M2-Q103      = ⚡ 21:21 ✎ ✖ Filter  Data settings	Cancel      Add		

Add Widget: Simple card

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q103      x

Filter

Maximum 1 timeseries/attribute is allowed.

Data settings

Cancel      Add

Add Widget: Simple card

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q103      x

Filter

Maximum 1 timeseries/attribute is allowed.

Data settings

Cancel      Add

Add Widget: Simple card

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q103      x

= ⚙️ ↗ random...:random...      ↗

Filter

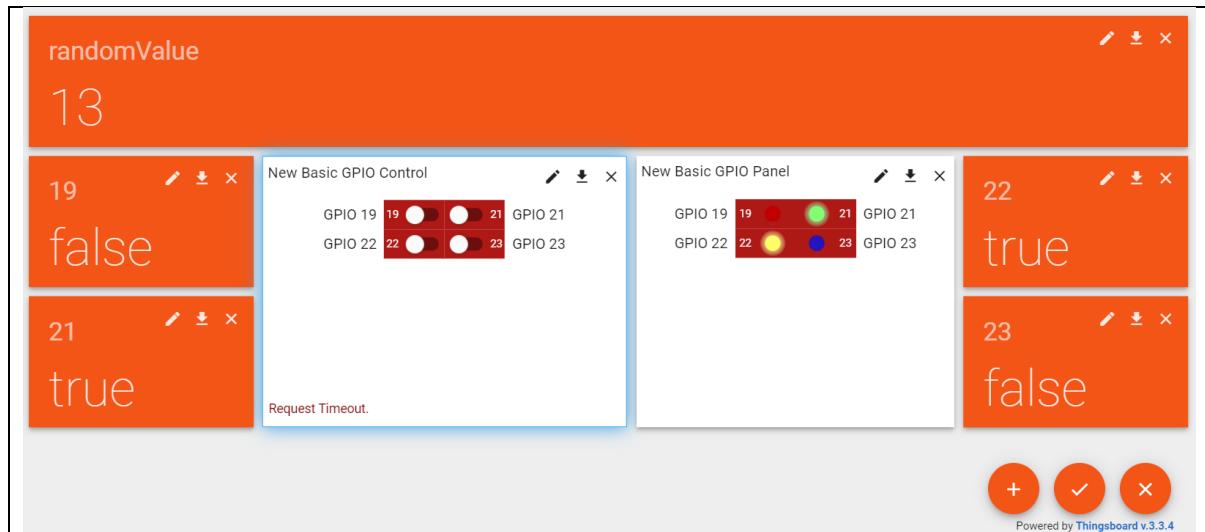
Maximum 1 timeseries/attribute is allowed.

Data settings

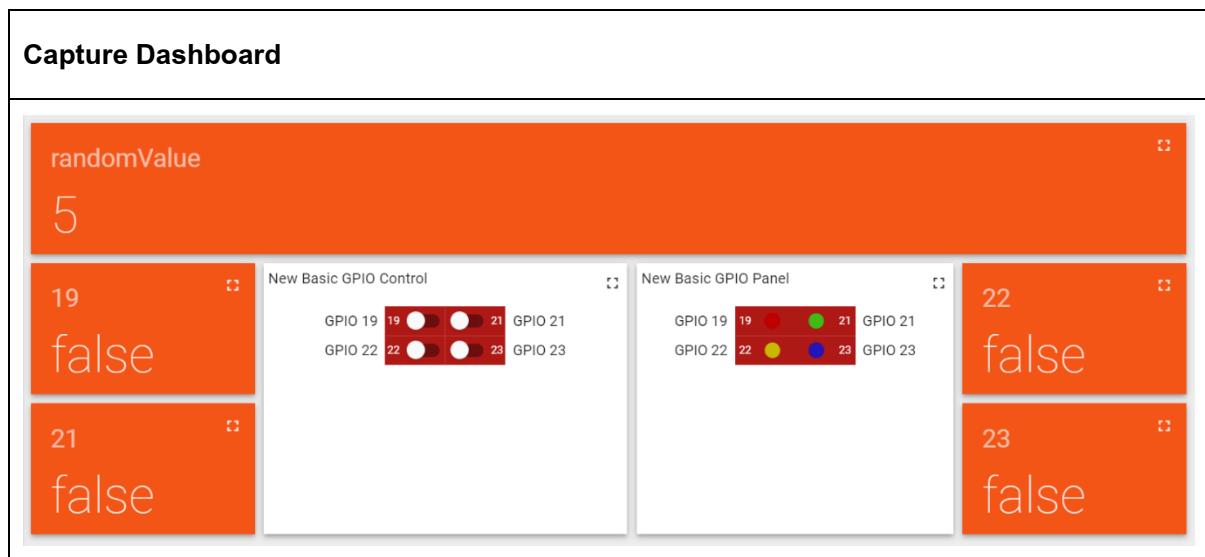
Special symbol to show next to value      Number of digits after floating point  
0

Cancel      Add

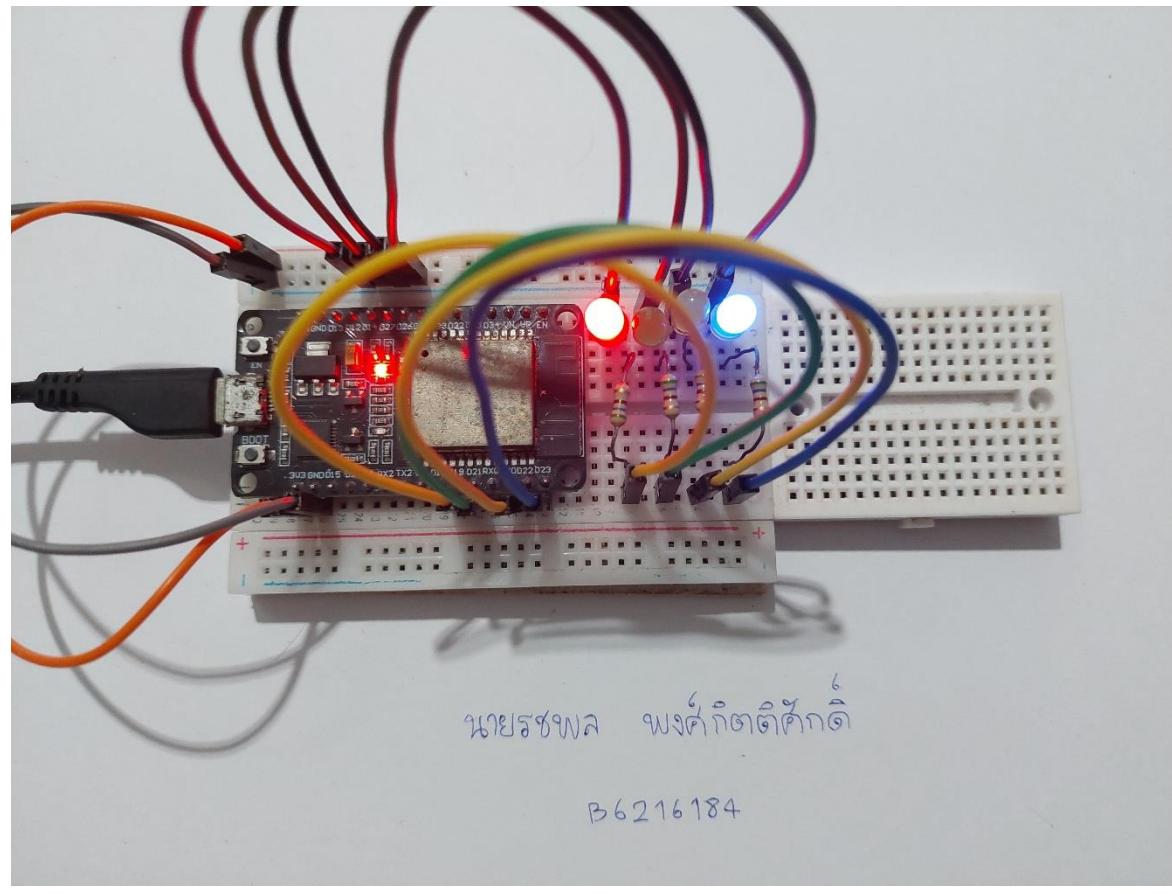
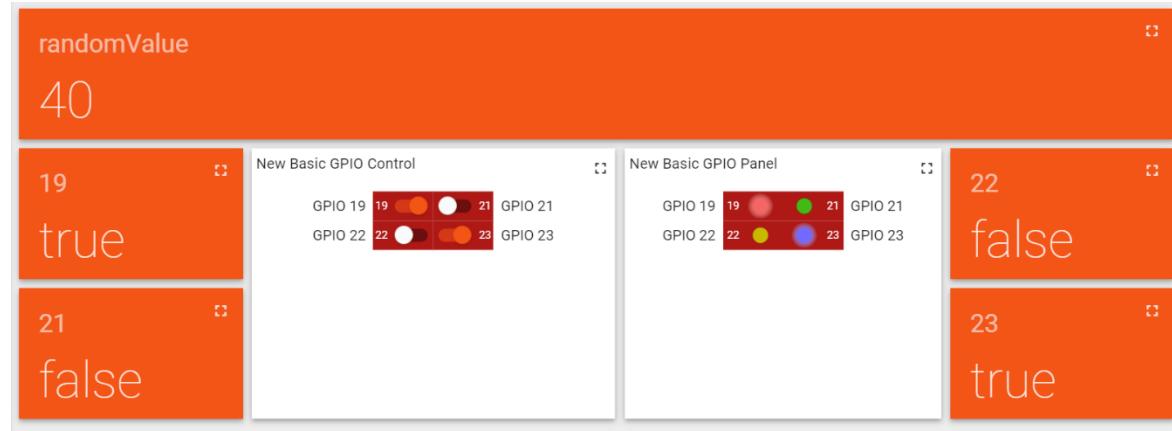
3.7 ปรับหน้าตา Dashboard ให้เป็นตามนี้ จากนั้นกดที่  เพื่อบันทึกการแก้ไข



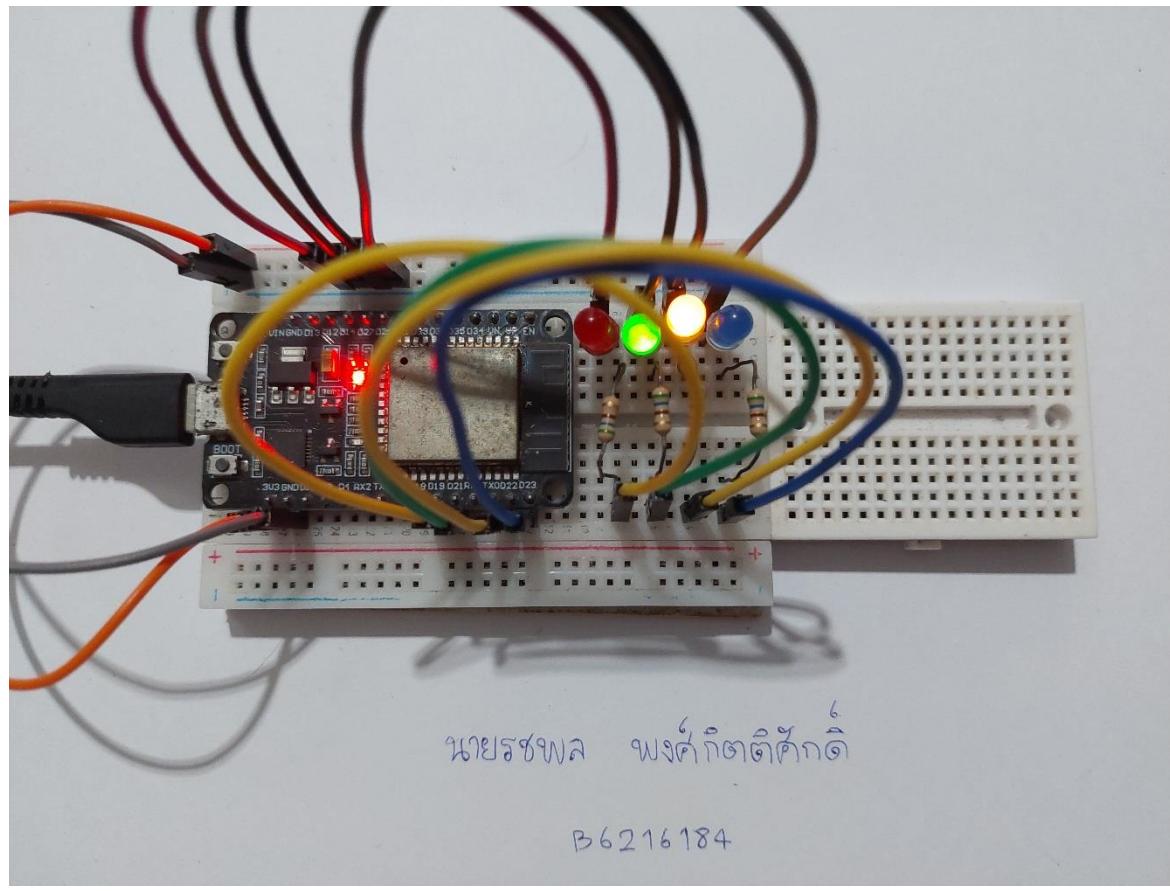
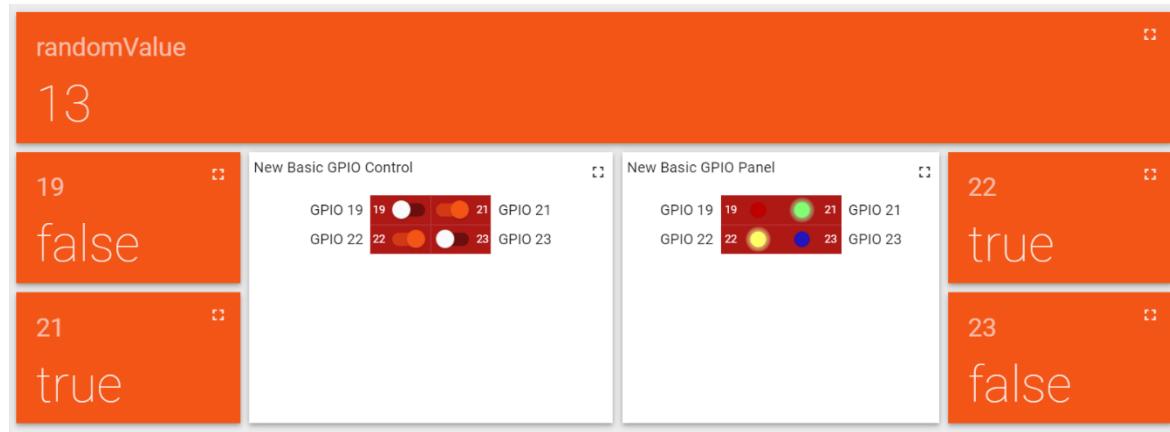
#### 4. ผลลัพธ์



รูปการทดสอบ 1 – เปิด/ปิด LED ทั้ง 4 หลอด และมีการแสดงค่าที่สุ่มได้บน Dashboard



รูปการทดสอบ 2 – เปิด/ปิด LED ทั้ง 4 หลอด และมีการแสดงค่าที่ส่งได้บน Dashboard



### รูปถ่ายหน้า Web Browser 1 – ที่ tab Last telemetry มีการส่งตัวเลขสุ่มมา

Last update time	Key ↑	Value
2022-05-03 01:20:15	randomValue	13

### รูปถ่ายหน้า Web Browser 2 – ที่ tab Attribute มีการส่งสถานะของ LED มา

Last update time	Key ↑	Value
2022-05-03 00:55:15	19	false
2022-05-03 00:55:15	21	true
2022-05-03 00:55:15	22	true
2022-05-03 00:55:15	23	false

### Quiz\_104 – Web Control 4 LED and Monitor Humid/Temperature

- Mission 4/4: การตรวจสอบและควบคุม อุณหภูมิ-ความชื้น ของโรงเรือนเลี้ยงไก่
  - ให้ใช้ ESP32 ส่งข้อมูลแบบสุ่มสองจำนวน คือ
    - Tempp\_A      สุ่มระหว่าง 20-40
    - Hudmid\_A      สุ่มระหว่าง 60-80
  - ข้อมูลทั้งสองค่าจะนำมาแสดงที่ Dashboard
  - สร้าง Alarm โดย หาก Tempp\_A > 35 หรือ Hudmid\_A > 70 ให้ Alarm
  - ศึกษาการตั้ง Alarm - <https://thingsboard.io/docs/user-guide/alarms/>
  - กำหนดรอบการตรวจสอบทุกๆ 20 วินาที
  - แชร์ Dashboard ไปให้ผู้ใช้งาน

#### 1. สร้าง Device

1.1 ทำการสร้าง Device ใหม่ โดยไปที่ Devices -> + -> Add new device

1.2 สร้าง Device โดยในช่อง Name ตั้งชื่อตามนี้ จากนั้นกดปุ่ม Add

The dialog box shows the following fields:

- Device details:**
  - Name \***: M2-Q104 (highlighted with a red box)
  - Label**: (empty)
  - Select existing device profile**: default (radio button selected)
  - Create new device profile**: (checkbox)
  - Is gateway**: (checkbox)
  - Description**: (empty)
- Credentials**: (Optional) (step 2)
- Customer**: (Optional) (step 3)

Buttons at the bottom right of the dialog box include: **Next: Credentials**, **Cancel**, and **Add** (highlighted with a red box).

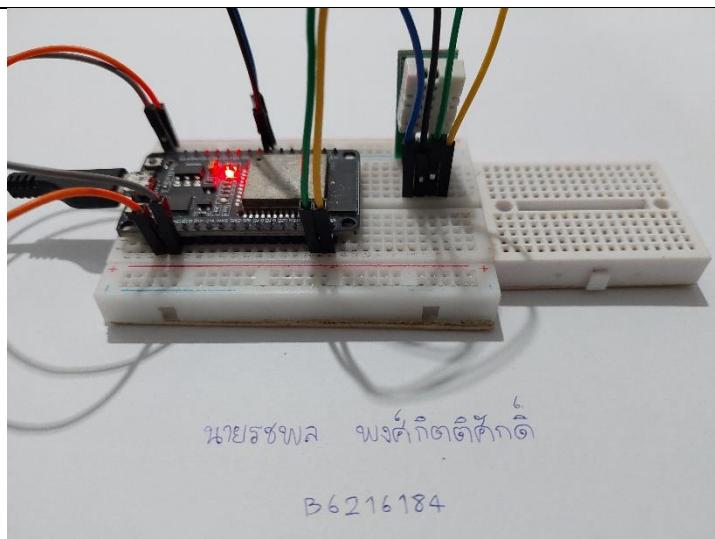
## 2. ส่งข้อมูลไปยัง Device

2.1 Copy Access Token ของ Device ที่สร้าง ไปที่ Devices -> ชื่อ Device -> Copy access token

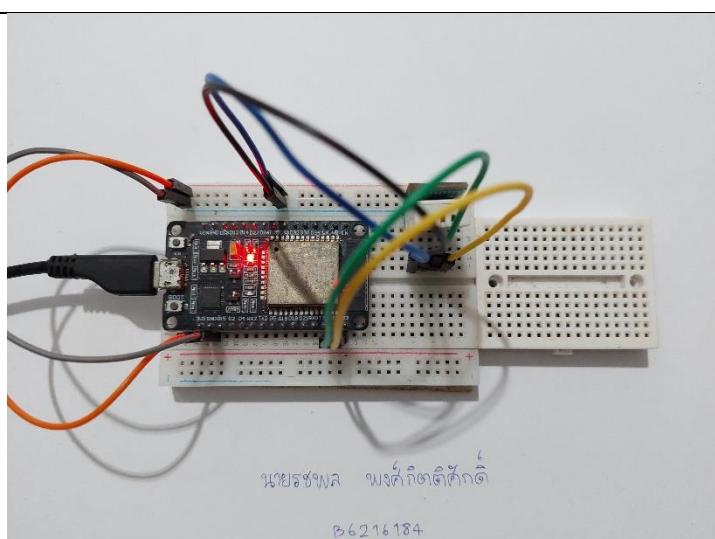
Access Token ก็คือ DEVICE\_TOKEN ใน Code – Main

2.2 ต่อวงจร

รูปการต่อวงจร 1 – ด้านข้าง



รูปการต่อวงจร 2 – ด้านบน



## 2.3 นำ Code อัปโหลดลง ESP32

Code
<pre>// arduinojson 6.x #include &lt;WiFi.h&gt; #include &lt;ThingsBoard.h&gt; #include "SHTC3.h"  #define WIFI_NAME "your-wifi-name" #define WIFI_PASS "your-wifi-pass" #define SERVER "demo.thingsboard.io" #define DEVICE_TOKEN "your-device-token"  WiFiClient esp32Client; int loopDaly = 20; int sendDataDelay = 5000; int counterSendData = 0; ThingsBoard tb(esp32Client);  void setup() {     Serial.begin(115200);     WiFi.begin(WIFI_NAME, WIFI_PASS);     while(WiFi.status() != WL_CONNECTED) {         delay(500);         Serial.print(".");     }     Serial.println("\nWiFi Connected");     Serial.print("IP Address: ");     Serial.println(WiFi.localIP()); }  void reconnectTB() {     if(!tb.connect(SERVER, DEVICE_TOKEN)) {         Serial.println("Failed to connect to ThingsBoard server");     }     else {         Serial.println("Connected to ThingsBoard server");     } }  void reconnectWiFi() {</pre>

```

WiFi.begin(WIFI_NAME, WIFI_PASS);
while(WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
}
randomSeed(millis());
Serial.println("\nWiFi Connected");
Serial.print("IP Address: ");
Serial.println(WiFi.localIP());
}

void loop() {
    if(WiFi.status() != WL_CONNECTED) reconnectWiFi();
    if(!tb.connected()) reconnectTB();
    tb.loop();

    if(counterSendData > sendDataDelay) {
        float t = random(2000, 4000)/100.0;//sht.readTempC();
        float h = random(6000, 8000)/100.0;//sht.readHumidity();
        if(t >= -40 && t <= 125) {
            tb.sendTelemetryFloat("temperature", t);
            tb.sendTelemetryFloat("humidity", h);
            Serial.println("temperature: " + String(t) + ", humidity: " + String(h));
        }
        counterSendData = 0;
    }
    delay(loopDaly);
    counterSendData += loopDaly;
}

```

2.4 ผลลัพธ์หลังจากส่งข้อมูลไปยัง Device จะได้ตัวแปร เพื่อนำไปใช้สร้าง Dashboard ต่อไป

The screenshot shows the ThingsBoard interface. On the left, there's a sidebar with various navigation options like Home, Rule chains, Customers, Assets, Devices (which is selected), Device profiles, OTA updates, Entity Views, and Edge instances. The main area has a header 'M2-Q104' and 'Device details'. Below the header, there are tabs for Details, Attributes, Latest telemetry (which is currently selected), Alarms, Events, Relations, and Audit Logs. A red box highlights the 'Latest telemetry' section. This section contains a table with three rows of data:

Last update time	Key ↑	Value
2022-05-03 12:43:26	humidity	75.5
2022-05-03 12:43:26	temperature	24.40999985

### 3. สร้าง Dashboard

3.1 ทำการสร้าง Dashboard ใหม่ โดยไปที่ Dashboards -> + ->Create new dashboard

3.2 สร้าง Dashboard โดยตั้งค่าตามนี้ จากนั้นกดปุ่ม Add

3.3 กดเข้าไปที่ Dashboard ที่สร้างขึ้นมาและกด Open dashboard

3.4 จากนั้นกดที่รูป  ด้านล่างขวาเพื่อเข้าสู่โหมดแก้ไขจากนั้นกดไปที่  -> Add alias

Device-M2-Q104

Add alias X

Alias name \*  Resolve as multiple entities

Filter type \*  ▾

Type \* Device  M2-Q104 X

Cancel Add

เมื่อ Add alias ทั้งหมดแล้วจะได้ผลลัพธ์เป็นแบบนี้ จากนั้นให้กด Save

Entity aliases X

Alias name	Entity filter	Resolve as multiple entities
1. Device-M2-Q104	One device	<input checked="" type="checkbox"/>  

Add alias Cancel Save

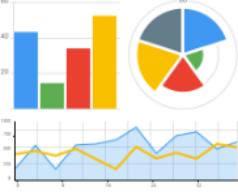
3.5 ทำการสร้าง Widget Chart โดยเมื่อออยู่ในโหมดแก้ไขให้กดไปที่  จากนั้นเลือก 

Charts -> Timeseries Line Chart

**Charts**

System

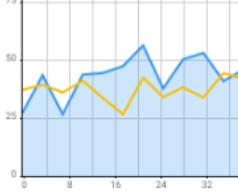
Display timeseries data using customizable line and bar charts. Use various pie charts to display latest values.



**Timeseries Line Chart**

Time series

Displays changes to timeseries data over time. For example, temperature or humidity readings.



ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

Add Widget: Timeseries Line Chart

Data      Settings      Advanced      Actions

Use dashboard timewindow      Timewindow: Realtime - last minute

Display timewindow

Datasources

Type	Parameters
= 1. Entity	Entity alias * Device-M2-Q104 = blue ^ temper... : temperature = green ^ humidity: humidity
Filter	

+ Add      Cancel      Add

ที่ tab Settings ในส่วนของ Title ให้ตั้งชื่อเป็น Timeseries Line Chart จากนั้นกดปุ่ม Add

Add Widget: Timeseries Line Chart

Data      Settings      Advanced      Actions

Title: Timeseries Line Chart

Display widget title

Title icon: Icon

Icon color: rgba(0, 0, 0, 0.87)

Icon size: 24px

Advanced settings

Widget style

Background color: #fff

Text color: rgba(0, 0, 0, 0.87)

Padding: 8px

Margin: 0px

+ Add

3.6 ทำการสร้าง Control widget โดยเมื่ออยู่ในโหมดแก้ไขให้กดไปที่



จากนั้นเลือก



Entity admin widgets -> Device admin table

<table border="1"> <thead> <tr> <th colspan="2">Asset admin table</th> </tr> <tr> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>WM452</td> <td>Device</td> </tr> <tr> <td>KL514</td> <td>Device</td> </tr> </tbody> </table> <table border="1"> <thead> <tr> <th colspan="2">Device admin table</th> </tr> <tr> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>WM452</td> <td>Device</td> </tr> <tr> <td>KL514</td> <td>Device</td> </tr> </tbody> </table>	Asset admin table		Name	Type	WM452	Device	KL514	Device	Device admin table		Name	Type	WM452	Device	KL514	Device	<p><b>Entity admin widgets</b></p> <p>System</p> <p>Templates of complex widgets that allow to list and create/update/delete devices and assets.</p>	<table border="1"> <thead> <tr> <th colspan="2">Device admin table</th> </tr> <tr> <th>Name</th> <th>Type</th> </tr> </thead> <tbody> <tr> <td>42JKR</td> <td>Device</td> </tr> <tr> <td>IC-34</td> <td>Device</td> </tr> </tbody> </table> <p><b>Device admin table</b></p> <p>Latest values</p> <p>Customized entity table widget with preconfigured actions to create, update and delete devices.</p>	Device admin table		Name	Type	42JKR	Device	IC-34	Device
Asset admin table																										
Name	Type																									
WM452	Device																									
KL514	Device																									
Device admin table																										
Name	Type																									
WM452	Device																									
KL514	Device																									
Device admin table																										
Name	Type																									
42JKR	Device																									
IC-34	Device																									

ที่ tab Data ในส่วนของ Datasources ให้กด + Add และตั้งค่าตามนี้

Add Widget: Device admin table

Data      Settings      Advanced      Actions

Datasources  
Maximum 1 datasource is allowed.

Type      Parameters

= 1. Entity      Entity alias \* Device-M2-Q104      Entity alias \* humidity: humidity

Filter

Data settings

Cancel      Add

ที่ tab Settings ในส่วนของ Title ให้ตั้งชื่อเป็น Device admin table จากนั้นกดปุ่ม Add

Add Widget: Device admin table

Data      Settings      Advanced      Actions

Title  
Display widget title  
Title  
Device admin table      Title Tooltip

Title icon  
Display title icon  
Icon more\_horiz      Icon color  
rgba(0, 0, 0, 0.87)      Icon size  
24px

Widget style  
Background color  
rgb(255, 255, 255)      Text color  
rgba(0, 0, 0, 0.87)      Padding  
4px      Margin

Advanced settings

Cancel      Add

3.7 ทำการสร้าง Widget การแจ้งเตือนโดยเมื่อออยู่ในโหมดแก้ไขให้กดไปที่



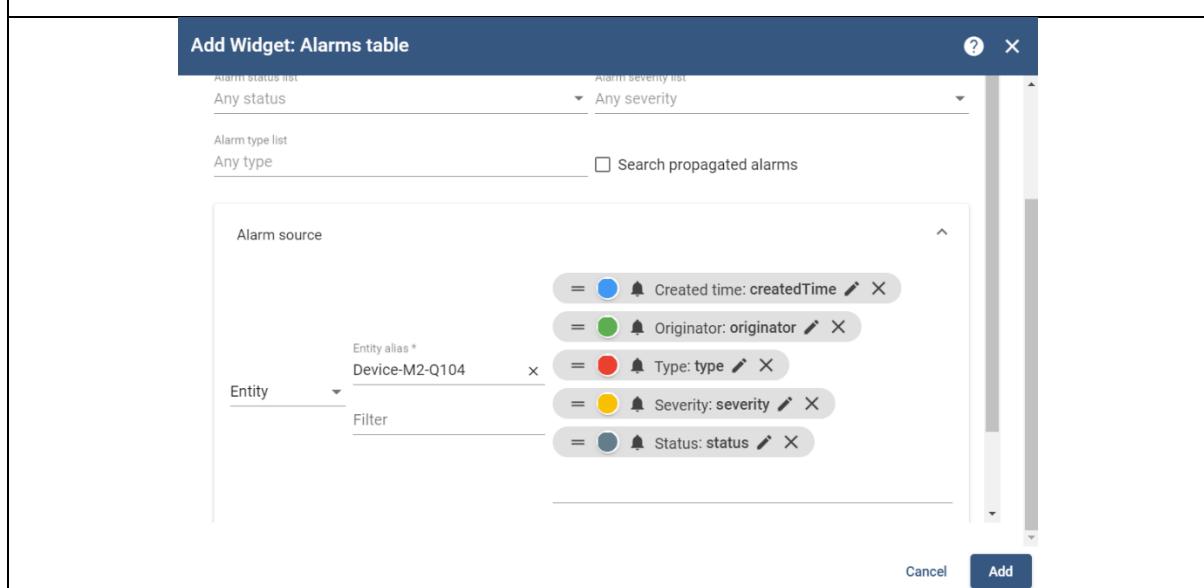
จากนั้นเลือก



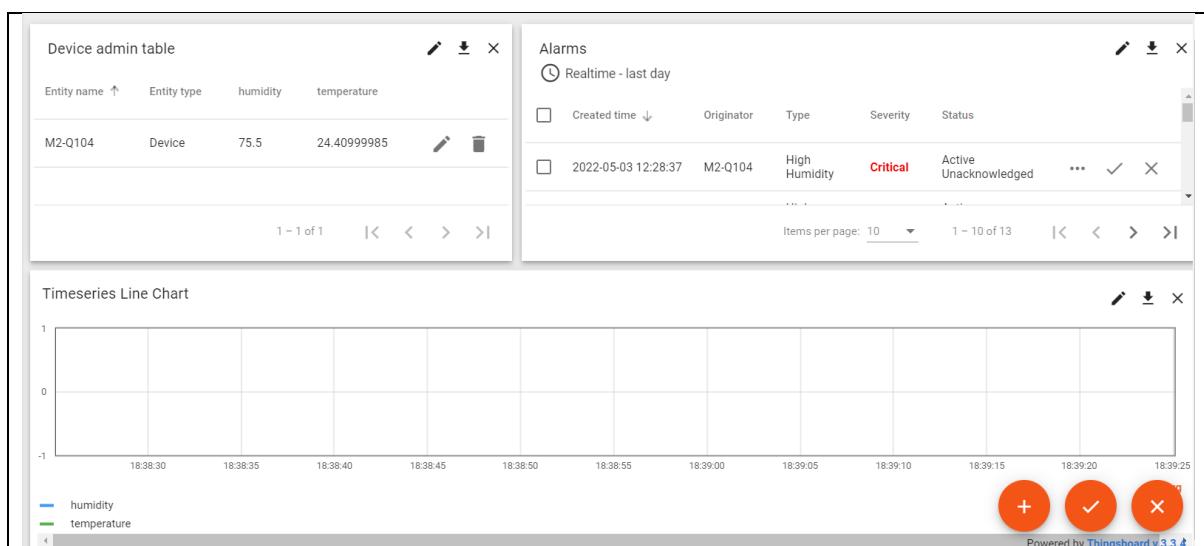
Alarm widgets -> Alarm table

<p>Alarms</p> <table border="1"> <thead> <tr> <th>Type</th> <th>Severity</th> <th>Status</th> </tr> </thead> <tbody> <tr> <td>Temperature</td> <td>Major</td> <td>Cleared</td> </tr> <tr> <td>Temperature</td> <td>Critical</td> <td>Cleared</td> </tr> <tr> <td>Low Humidity</td> <td>Warning</td> <td>Active</td> </tr> <tr> <td>Low Humidity</td> <td>Warning</td> <td>Active</td> </tr> </tbody> </table>	Type	Severity	Status	Temperature	Major	Cleared	Temperature	Critical	Cleared	Low Humidity	Warning	Active	Low Humidity	Warning	Active	<p><b>Alarm widgets</b></p> <p>System</p> <p>Visualization of alarms for devices, assets and other entities.</p>	<p><b>Alarms table</b></p> <p>Alarm widget</p> <p>Displays alarms based on defined time window and other filters.</p> <table border="1"> <thead> <tr> <th>Type</th> <th>Severity</th> </tr> </thead> <tbody> <tr> <td>Temperature</td> <td>Major</td> </tr> <tr> <td>Temperature</td> <td>Critical</td> </tr> <tr> <td>Low Humidity</td> <td>Warning</td> </tr> </tbody> </table>	Type	Severity	Temperature	Major	Temperature	Critical	Low Humidity	Warning
Type	Severity	Status																							
Temperature	Major	Cleared																							
Temperature	Critical	Cleared																							
Low Humidity	Warning	Active																							
Low Humidity	Warning	Active																							
Type	Severity																								
Temperature	Major																								
Temperature	Critical																								
Low Humidity	Warning																								

ที่ tab Data ในส่วนของ Datasources ให้ตั้งค่าตามนี้ จากนั้นกด Add

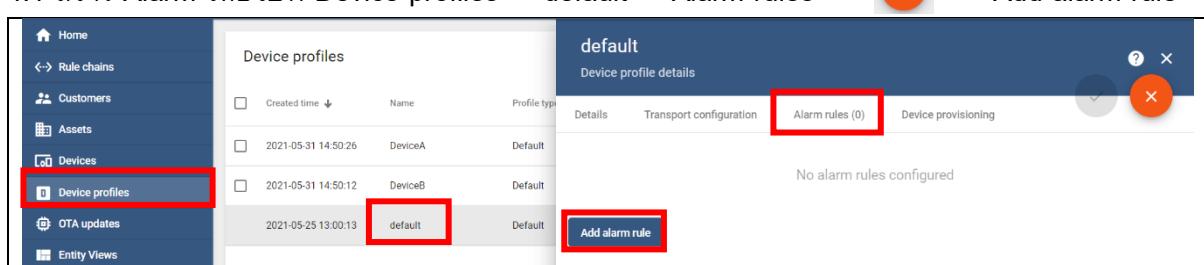


3.8 ปรับหน้าตา Dashboard ให้เป็นตามนี้ จากนั้นกดที่ เพื่อบันทึกการแก้ไข



#### 4. สร้าง Alarm

4.1 สร้าง Alarm โดยไปที่ Device profiles -> default -> Alarm rules -> -> Add alarm rule



### 4.3 ทำการสร้าง Alarm สำหรับแจ้งเตือน High Temperature โดยตั้งค่าตามนี้ จากนั้นกดไปที่

Please add alarm rule condition

The screenshot shows the ThingsBoard interface. On the left sidebar, 'Device profiles' is selected. In the main area, a 'default' device profile is open. Under the 'Alarm rules (1)' tab, a new rule for 'High Temperature' is being configured. The 'Condition' field contains the placeholder text 'Please add alarm rule condition'. A red box highlights this field. Below it, the 'Severity' is set to 'Critical' and the 'Schedule' is set to 'Active all the time'. At the bottom, there's a note about using a mobile dashboard.

**Alarm type:** High Temperature

**Severity:** Critical

ให้ตั้งค่าตามนี้สำหรับแจ้งเตือนค่า temperature หลังจากนั้นกด Add ในกรอบสีแดง

The screenshot shows the 'Add key filter' dialog. It has three main sections: 'Key type \*' (Timeseries), 'Key name \*' (temperature), and 'Value type' (123 Numeric). Below these, a 'Filters' section shows a table with columns 'Operation' and 'Value', stating 'No filters configured'. At the bottom are two buttons: 'Add' (highlighted with a red box) and 'Add complex'.

**Add key filter**

Key type *	Key name *	Value type
Timeseries	temperature	123 Numeric

**Filters**

Operation	Value
No filters configured	

**Add** **Add complex**

ให้ตั้งค่าในส่วน Filters ตามนี้สำหรับแจ้งเตือนค่า temperature ที่เกิน 35 องศาเซลเซียส หลังจากนั้น กด Add ในกรอบสีแดง

**Add key filter**

Key type \*: Timeseries      Key name \*: temperature      Value type: 123 Numeric

Filters

Operation	Value
greater than	35 Default value

Add    Add complex

Cancel    **Add**

4.4 ทำการสร้าง Alarm สำหรับแจ้งเตือน High Humidity ซึ่งสร้างต่อจาก Alarm ของ temperature โดย กดไปที่ Add alarm rule

default

Device profile details

Details    Transport configuration    **Alarm rules (1)**    Device provisioning

Alarm type: High Temperature

Create alarm rules

Severity: Critical	Condition: temperature greater than 35
Schedule: Active all the time	Add details +
Mobile dashboard: No dashboard selected	

Add create condition    Clear alarm rule    Add clear condition

**Add alarm rule**

ให้ตั้งค่าตามนี้ จากนั้นกดไปที่ Please add alarm rule condition

The screenshot shows the ThingsBoard interface under the 'Device profiles' section. A new alarm rule is being configured for the 'default' profile. The 'Alarm rules (2)' tab is selected. A red box highlights the 'Condition: Please add alarm rule condition +' button. Below it, the 'Severity' is set to 'Critical' and the 'Schedule' is 'Active all the time'. The 'Add details +' button is also visible.

Alarm type: High Humidity

Severity: Critical

ให้ตั้งค่าตามนี้สำหรับแจ้งเตือนค่า humidity หลังจากนั้นกด Add ในกรอบสีแดง

Add key filter

The 'Add key filter' dialog box is open. It shows the configuration for a 'humidity' key. The 'Key type' is 'Timeseries', 'Key name' is 'humidity', and 'Value type' is 'Numeric'. The 'Filters' section is empty, showing 'No filters configured'. At the bottom, there are 'Add' and 'Add complex' buttons, with 'Add' highlighted by a red box. There are also 'Cancel' and 'Add' buttons at the bottom right.

ให้ตั้งค่าในส่วน Filters ตามนี้สำหรับแจ้งเตือนค่า humidity ที่เกิน 70 เปอร์เซ็นต์ หลังจากนั้นกด Add ในกรอบสีแดง

Add key filter

Key type \* Timeseries Key name \* humidity Value type 123 Numeric

Filters

Operation	Value
greater than	70 Default value

Add Add complex

Cancel Add

4.5 หลังจากที่สร้าง Alarm ครบแล้วให้กดที่ เพื่อบันทึกการแก้ไข

Device profiles

Created time	Name	Profile type
2021-05-31 14:50:26	DeviceA	Default
2021-05-31 14:50:12	DeviceB	Default

default

Device profile details

Details Transport configuration Alarm rules (2) Device provisioning

Clear alarm rule Add clear condition

## 5. Share Dashboard

5.1 ในการ Share Dashboard จะต้อง Share ทั้ง Dashboard และ Device ที่ใช้ใน Dashboard นี้ ทั้งหมดเพื่อเป็นการอนุญาติให้ทั้งรูปแบบ Dashboard และข้อมูลใน Dashboard ให้คนอื่นเข้ามาดูได้

Dashboards -> ชื่อ Dashboard -> เมื่อ Public เสร็จจะได้ Public ->

Dashboards

Created time	Title	Assigned to customers	Actions
2022-05-03 01:01:30	M2-Q104-Web-Control-4 LED-and-Monitor-Humid-Temperature	<input checked="" type="checkbox"/> Public	
2022-05-03 00:12:26	M2-Q103-ThingsBoard-Data-Monitor-and-Control-with-MQTT-Protocol	<input type="checkbox"/>	
2022-05-02 16:31:29	M2-Q102-ThingsBoard-Data-Monitor-and-Control	<input type="checkbox"/>	
2022-05-02 15:25:34	M2-Q101-ThingsBoard-Data-Monitor	<input type="checkbox"/>	
2022-04-30 11:42:11	Guidline	<input type="checkbox"/>	
2022-04-29 20:39:09	Set&ClearAlarm	<input type="checkbox"/>	
2021-06-08 17:41:57	Pk007_TB_Part2_4Node	<input checked="" type="checkbox"/>	

Devices -> ชื่อ Device -> เมื่อ Public เสร็จจะได้ Public ->

The screenshot shows the 'Devices' section of the ThingsBoard interface. A device named 'M2-Q104' is highlighted with a red box. The 'Public' checkbox in the row for M2-Q104 is checked, indicated by a checkmark inside a red box. Other columns include 'Created time', 'Name', 'Device profile', 'Label', 'Customer', and 'Is gateway'.

## 6. ผลลัพธ์

**Capture Dashboard**

The dashboard includes:

- Device admin table:** Shows a single entry for 'M2-Q104' as a 'Device' with ID '62.54999924' and serial '26.35000038'. Buttons for edit and delete are shown.
- Alarms:** Displays two critical alarms for 'M2-Q104' from May 3rd: 'High Humidity' and 'High Temperature', both marked as 'Unacknowledged'.
- Timeseries Line Chart:** A chart showing 'humidity' over time from 12:20:25 to 12:21:15. The blue line represents humidity values fluctuating between 20 and 80.

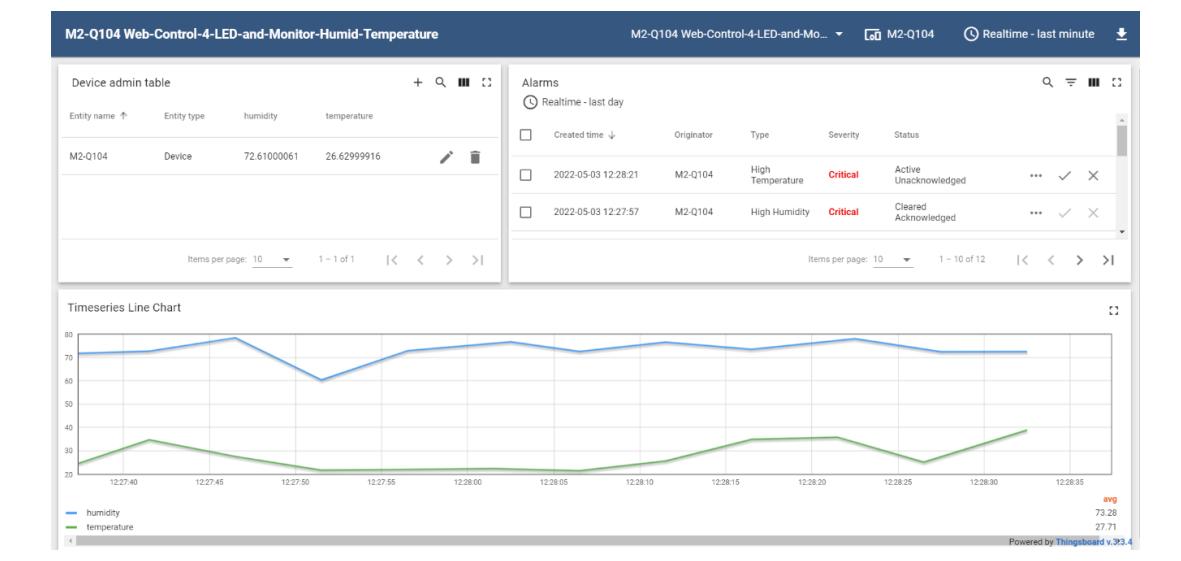
Text at the bottom: 'ในรูปนี้เกิด Alarm ทั้ง 2 พร้อมกัน'

**รูปถ่ายหน้า Web Browser - เมื่อมีการส่งข้อมูล temperature และ humidity มาจาก ESP32**

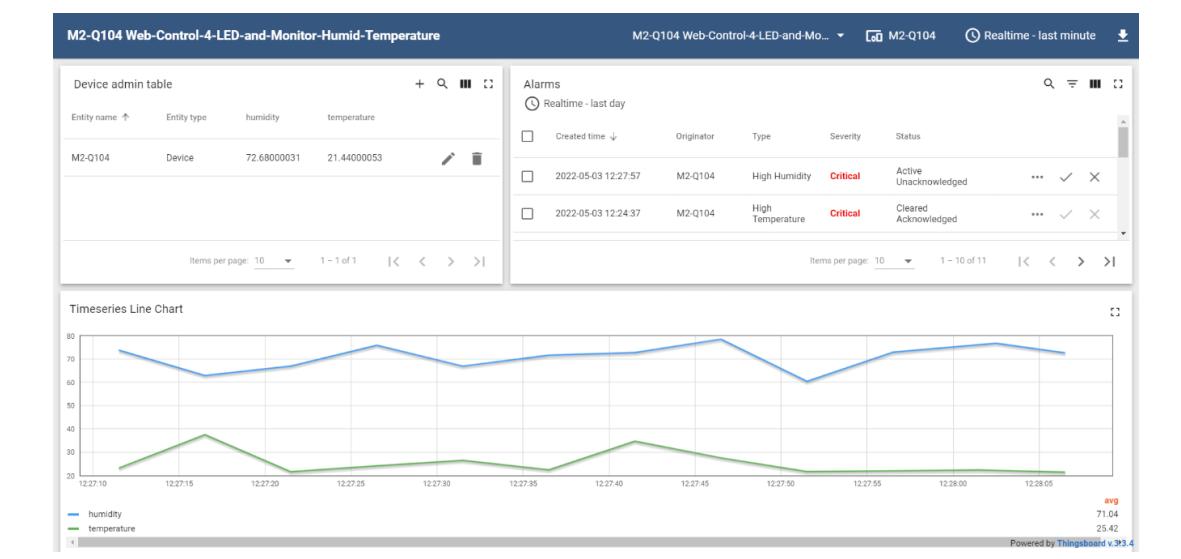
The screenshot shows the 'Device details' page for 'M2-Q104'. The 'Latest telemetry' tab is active, displaying the following data:

Last update time	Key	Value
2022-05-03 12:22:10	humidity	77.69000244
2022-05-03 12:22:09	temperature	21.45999908

## รูปการทดสอบ 1 – เมื่อมี High Temperature Alarm เกิดขึ้น



## รูปการทดสอบ 2 – เมื่อมี High Humidity Alarm เกิดขึ้น



Alarm ทั้ง 2 จะไม่เกิดข้ออึกถ้า Alarm เก่ายังไม่ถูกกด clear และเมื่ออุณหภูมิ หรือความชื้นเกินค่าที่กำหนดไว้จะเกิด Alarm ทันที ด้วยเหตุผลทั้ง 2 นี้ทำให้มีต้องกำหนดเวลาตรวจสอบอุณหภูมิ และความชื้นตามที่หัวข้อนี้ต้องการให้ทำ

## Link Shared Dashboard

<https://demo.thingsboard.io/dashboard/e4dfcd00-ca41-11ec-9a68-6b50da95566e?publicId=94eeb190-bd26-11eb-a311-cf80d7127bfd>