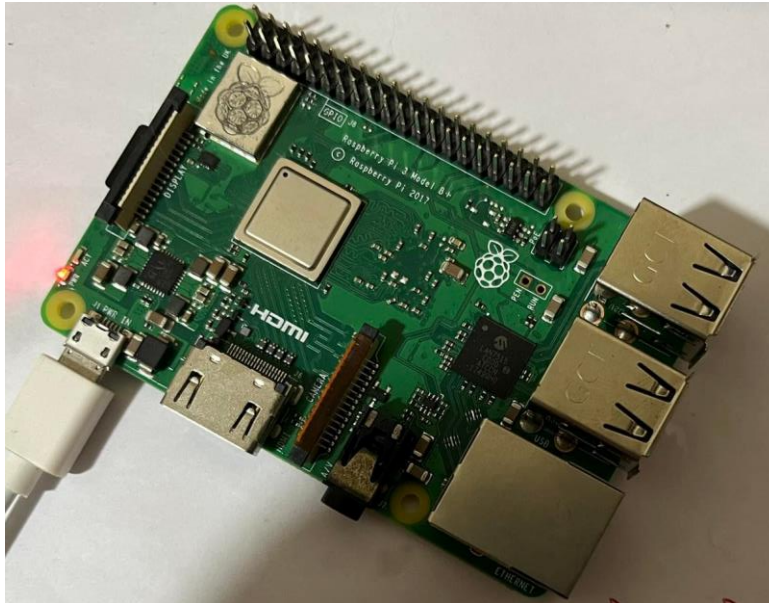


D3 การทดลองที่ 1

แสดงรูป โปรแกรม ของผลการทำงานตามหัวข้อ Lab304 – Using Rule Chains, MAP Widget and Multilayer dashboard



1.ติดตั้ง Raspberry Pi OS จากนั้นตั้งค่าการรีโมทจากอุปกรณ์อื่นและติดตั้ง Thingsboard บน Raspberry Pi สามารถดูข้อมูลเพิ่มเติมจากไฟล์โน้ตโพลเดอร์ Reference หรือ

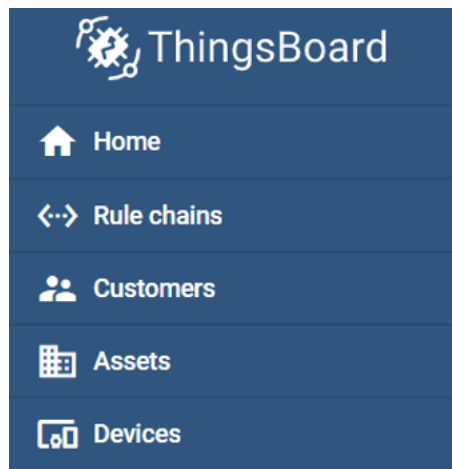
<https://howchoo.com/pi/raspberry-pi-imager>

<https://www.youtube.com/watch?v=ntaXWS8Lk34>

https://www.youtube.com/watch?v=08ZwV9xofWw&ab_channel=PopfizzComputerScience

<https://thingsboard.io/docs/user-guide/install/rpi/>

2. Login ที่ IP:8080 ด้วย Tenant Administrator: tenant@thingsboard.org / tenant จากนั้นสร้าง Device ชื่อ Pk_Station_AA และ Pk_Station_BB ตามรูป จากนั้น copy Token เพื่อนำไปใช้

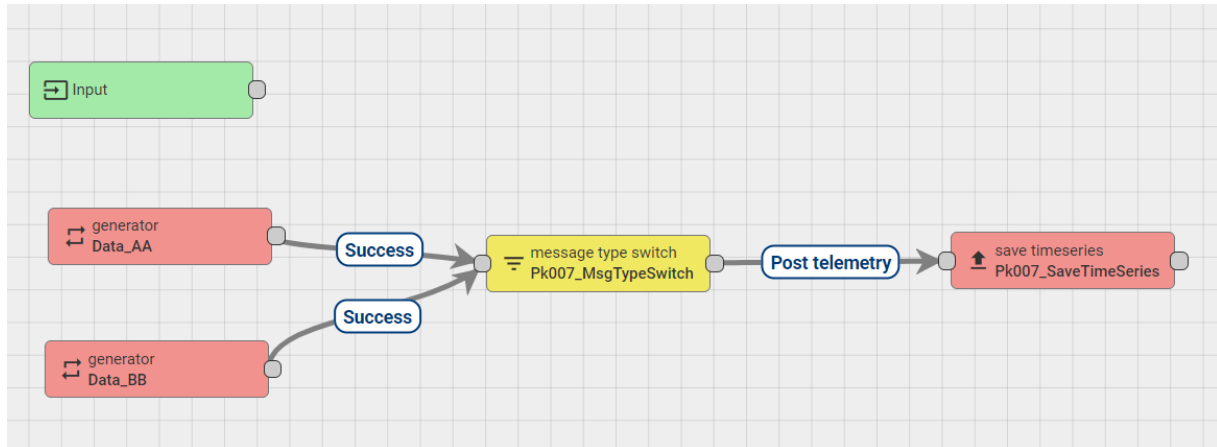


Devices							
Device profile							
All							
<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	Customer	Public	Is gateway
<input type="checkbox"/>	2022-05-09 20:37:46	Pk_Station_BB	default			<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	2022-05-09 20:37:29	Pk_Station_AA	default			<input type="checkbox"/>	<input type="checkbox"/>

3.สร้าง rule chains

Rule chains		
<input type="checkbox"/>	Created time ↓	Name
<input type="checkbox"/>	2022-05-11 00:11:57	Pk007_Chain

4.สร้าง rule chain ดังรูป



5.กำหนดค่าของ generator แต่ละตัวดังนี้

Name *
Data_AA ☐ Debug mod

Message count (0 - unlimited) *
0

Period in seconds *
1

Originator Type Device
Device ▼ Pk_Station_A

Generate

```
function Generate(prevMsg, prevMetadata, prevMsgType) {  
  1 var msg = {  
  2     station_name : "SUT F11",  
  3     latitude : 14.876488085551973,  
  4     longitude : 102.01498458834561,  
  5     temperature: (Math.random()*10+40).toFixed(2),  
  6     humidity: (Math.random()*10+80).toFixed(2) };  
  7 var metadata = { data: 10 };  
}
```

Generator: Data_AA

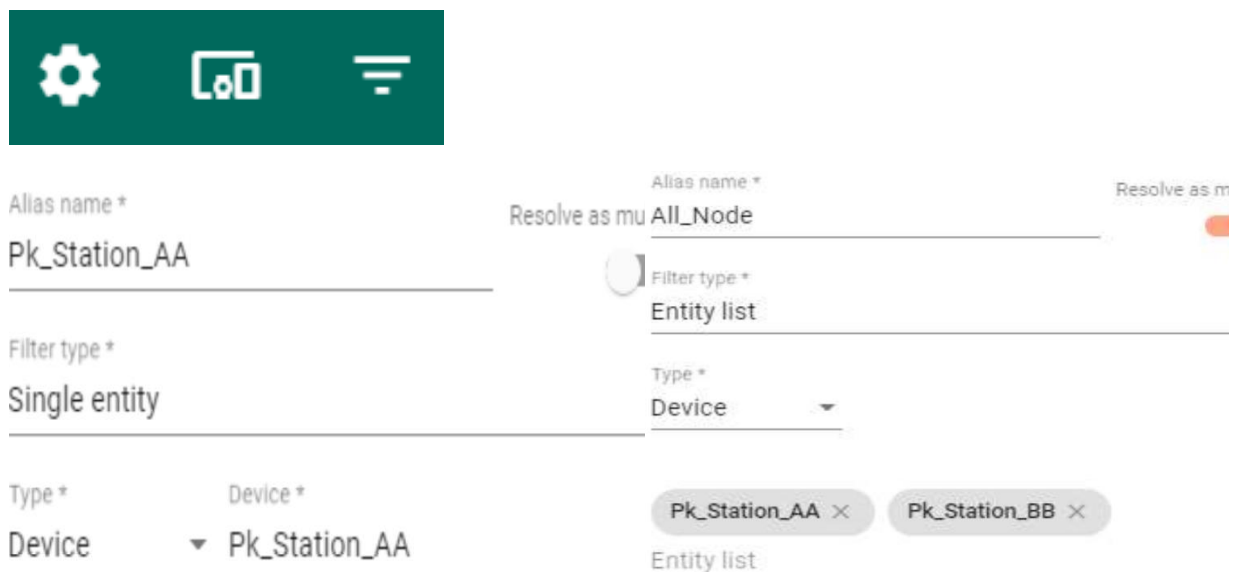
```
var msg = {  
  station_name : "SUT F11",  
  latitude : 14.876488085551973,  
  longitude : 102.01498458834561,  
  temperature: (Math.random()*10+40).toFixed(2),  
  humidity: (Math.random()*10+80).toFixed(2) };
```

```
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";
return { msg: msg, metadata: metadata, msgType: msgType };
```

Generator: Data_BB

```
var msg = {
  station_name : "SurathamPitak School",
  latitude : 14.91002308772653,
  longitude : 102.0703244094149,
  temperature: (Math.random()*10+30).toFixed(2),
  humidity: (Math.random()*10+70).toFixed(2) };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";
return { msg: msg, metadata: metadata, msgType: msgType };
```

6.สร้าง Dashboard จากนั้น สร้าง Entity Aliases โดยมี 3 alias คือของ Device Pk_Station_AA และ Pk_Station_BB ส่วนสุดท้ายคือ All_Node ดังรูป



Alias name *

Resolve as mu

Alias name *

Resolve as m

Filter type *

Entity list

Type *

Device

Type *

Device *

Device

Pk_Station_AA

Pk_Station_AA

Pk_Station_BB

Entity list

7.ที่ dashboard เพิ่ม widget ที่เป็นแผนที่คือ Google map จากนั้นตั้งค่าดังรูป

Datasources

Type	Parameters
= 1. Entity	Entity alias * All_Node x
	Filter

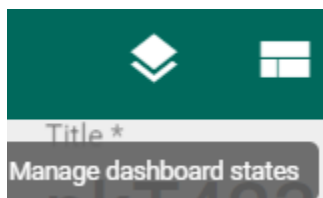
= humidity: humidity ✎ ✕
 = latitude: latitude ✎ ✕
 = longitude: longitude ✎ ✕
 = station... : station_... ✎ ✕
 = temper... : tempera... ✎ ✕

8. เพิ่ม widget ชื่อ Entity Admin Widget – Device Admin Table with Data

Type	Parameters
= 1. Entity	Entity alias * All_Node x
	Filter

= humidity: humid
 = temper... : tempe
 = station... : statio

9. เพิ่มหน้า dashboard โดย Edit → Manage dashboard states



Name *

`${entityName}`

State Id *

SUT_Data

☐ Root state

Name *

`${entityName}`

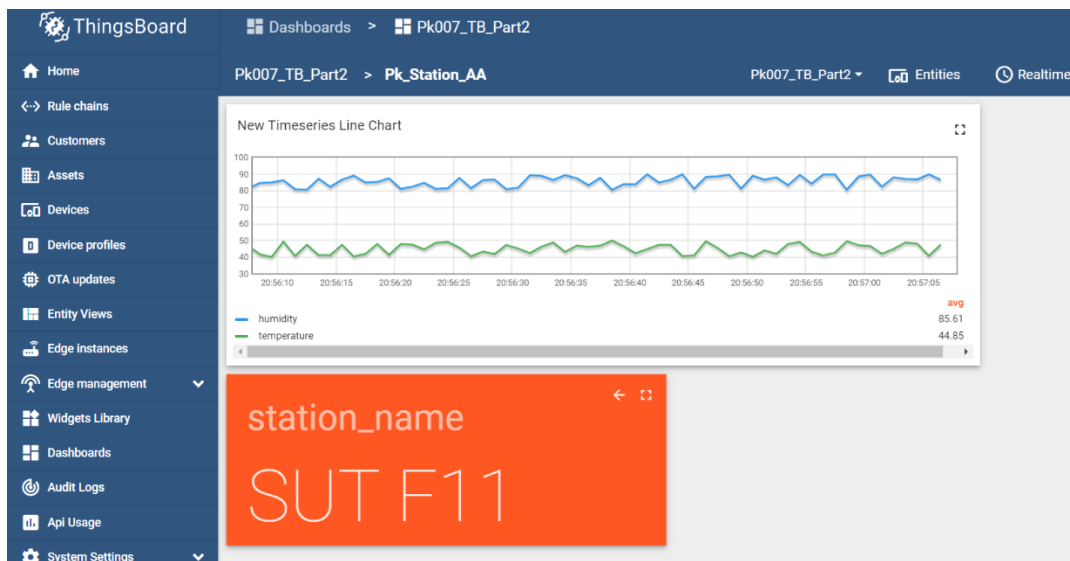
State Id *

STP_Data

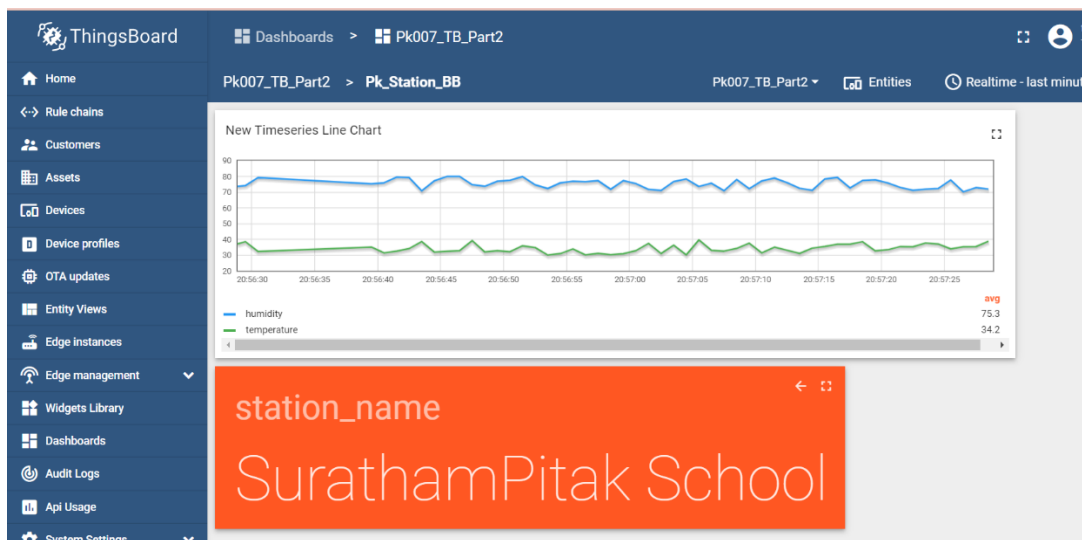
☐ Root state

10.เข้าไปแก้ไขหน้า dashboard ที่สร้างขึ้นใหม่ของแต่ละที่ดังรูป

SUT_Data












SPT_Data



11.แก้ไข Device Admin Table โดยลบของเก่าออกแล้วเพิ่มใหม่ตามรูป

ลบของเก่าออก

Data	Settings	Advanced	Actions
Actions + Q			
Action source ↑	Name	Icon	Type
Action cell button	Edit device		Custom action (with HTML template)  
Action cell button	Delete device		Custom action  
Widget header button	Add device		Custom action (with HTML template)  

ของใหม่

Action source *

Action cell button ▼

Name *

Cell_AdminTable



Icon
play_arrow

☐ Show/hide action using function

Type *

Custom action ▼

function (\$event, widgetContext, entityId, entityName, additionalParams, entityLabel) {

Tidy



```
1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext.servicesMap.get
  ('deviceService'));
3 deviceService.getDevice(entityId.id).subscribe(function(device) {
4 if (device.name == 'Pk Station AA') {
```

```
var $injector = widgetContext.$scope.$injector;
```

```
let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));
```

```
deviceService.getDevice(entityId.id).subscribe(function(device) {
```

```

if (device.name == 'Pk_Station_AA') {
  openDashboardState('SUT_Data',device)}
else if(device.name == 'Pk_Station_BB') {
  openDashboardState('STP_Data',device)}
});
function openDashboardState(stateId,device) {
  var params = {
    entityId: entityId,
    entityName: device.name};
  widgetContext.stateController.openState(stateId, params, false);}

```

Action source *

On row click

Name *

Row_AdminTable



Icon

more_horiz

Type *

Update current dashboard state

Target dashboard state

☐ Open right dashboard layout (mobile view)

☒ Set entity from widget

State entity parameter name

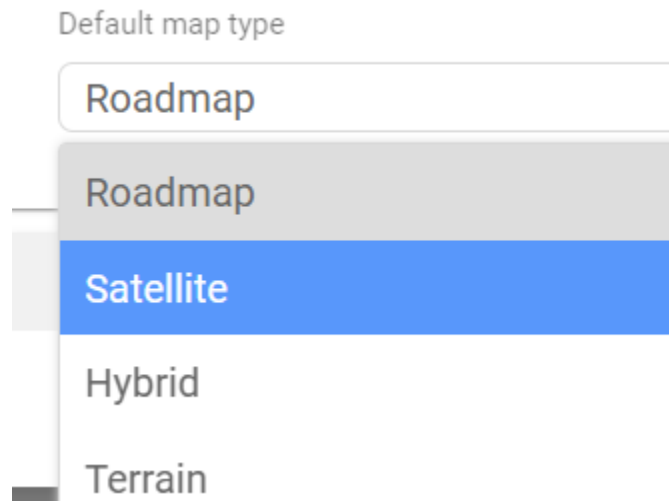
By default

จะได้อ

Device admin table					
Entity name ↑	Entity type	humidity	temperature	station_name	
Pk_Station_AA	Device	88.31	41.45	SUT F11	▶
Pk_Station_BB	Device	77.63	39.61	SurathamPlitak School	▶

12.แก้ไขการตั้งค่าของ google map ใหม่โดยเลือกที่หัวข้อ Advance จากนั้น

เลือก Default map type เป็น Satellite



แก้ไขค่าใน Label เพื่อเปลี่ยนรูปแบบของป้ายแสดงข้อมูลบนแผนที่

☒ Show label

☐ Use label function

Label (pattern examples: '{entityName}', '{entityName}: (Text {keyName} units.)')

{entityName}

```
<div style="position: relative; white-space: nowrap; text-align: center; font-size: 14px; top: 5px;"> <span style="border: 2px solid #000; border-radius: 10px; color: #000; background-color: #fff; padding-left: 5px; padding-right: 5px; padding-top: 3px; padding-bottom: 3px;">{entityName}</span> </div>
```

แก้ไขค่าใน Tooltip เพื่อสามารถโยงไปที่หน้า Dashboard อื่นที่สร้างไว้

☒ Auto-close tooltips

☐ Use tooltip function

Tooltip (for ex. 'Text {keyName} units.' or <link-act name='my-action'>Lir

```
<b>{entityName}</b><br/><br/><b>Latitude:</b> {la  
<br/><b>Temperature:</b> {temperature} °C<br/><sr
```

```
<b>{entityName}</b><br/><br/>
```

```
<b>Station:</b> ${station_name}<br/>
<b>Temperature:</b> ${temperature} °C<br/>
<b>Humidity:</b> ${humidity} %<br/>
<link-act name='Sensor_Details'>Sensor Details</link-act>
```

แก้ไขในส่วนของแถบ Action โดยเพิ่ม Action เพื่อ update และ การเปลี่ยนหน้า Dashboard เมื่อกดที่หมุดบนแผนที่ ดังรูป

Action source *

Tooltip tag action

Name *

Tool_Action



Icon

more_horiz

Type *

Update current dashboard state

Target dashboard state

☐ Open right dashboard layout (mobile view)

☒ Set entity from widget

State entity parameter name

By default

Action source *

On marker click

Name *

Sensor_Details



Icon

play_arrow

Type *

Custom action

```
function ($event, widgetContext, entityId, entityName, additionalParams, entityLabel) {
```

Tidy ?

```
1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext.servicesMap.get
  ('deviceService'));
3 deviceService.getDevice(entityId.id).subscribe(function(device) {
4   if (device.name == 'Pk_Station_AA') {
5     openDashboardState('SUT_Data'.device)
6   }
7 }
```

var \$injector = widgetContext.\$scope.\$injector;

```
let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));
```

```

deviceService.getDevice(entityId.id).subscribe(function(device) {

if (device.name == 'Pk_Station_AA') {

openDashboardState('SUT_Data',device)}

else if(device.name == 'Pk_Station_BB') {

openDashboardState('STP_Data',device)}});

function openDashboardState(stateId,device) {

var params = {

entityId: entityId,

entityName: device.name};

widgetContext.stateController.openState(stateId, params, false);}

```

13.เมื่อได้แล้วจะเป็นดังภาพ

The screenshot displays the ThingsBoard dashboard interface. On the left is a sidebar with navigation options: Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Widgets Library, Dashboards, Audit Logs, Api Usage, and System Settings. The main area is titled 'Pk007_TB_Part2' and contains two widgets. The first widget, 'New OpenStreetMap', shows a satellite map with two red location pins labeled 'Pk_Station_AA' and 'Pk_Station_BB'. The second widget, 'Device admin table', displays a table with the following data:

Entity name ↑	Entity type	humidity	temperature	station_name
Pk_Station_AA	Device	88.31	41.45	SUT F11
Pk_Station_BB	Device	77.63	39.61	SurathamPitak School

At the bottom right of the dashboard, it says 'Powered by Thingsboard v.3.3.0'.

D3 การทดลองที่ 2

จากหัวข้อ Quiz_301 ลองปรับเพิ่มจุดข้อมูลจากสองจุด(STP,SUT) เป็นสี่จุดข้อมูล ตามแต่ผู้เรียนกำหนด

1.เพิ่มเติมจากการทดลองที่ 1 โดยการสร้าง Device เพิ่มอีก 2 device ดังรูป

























Devices

tenant@thingsboard.org
Tenant administrator

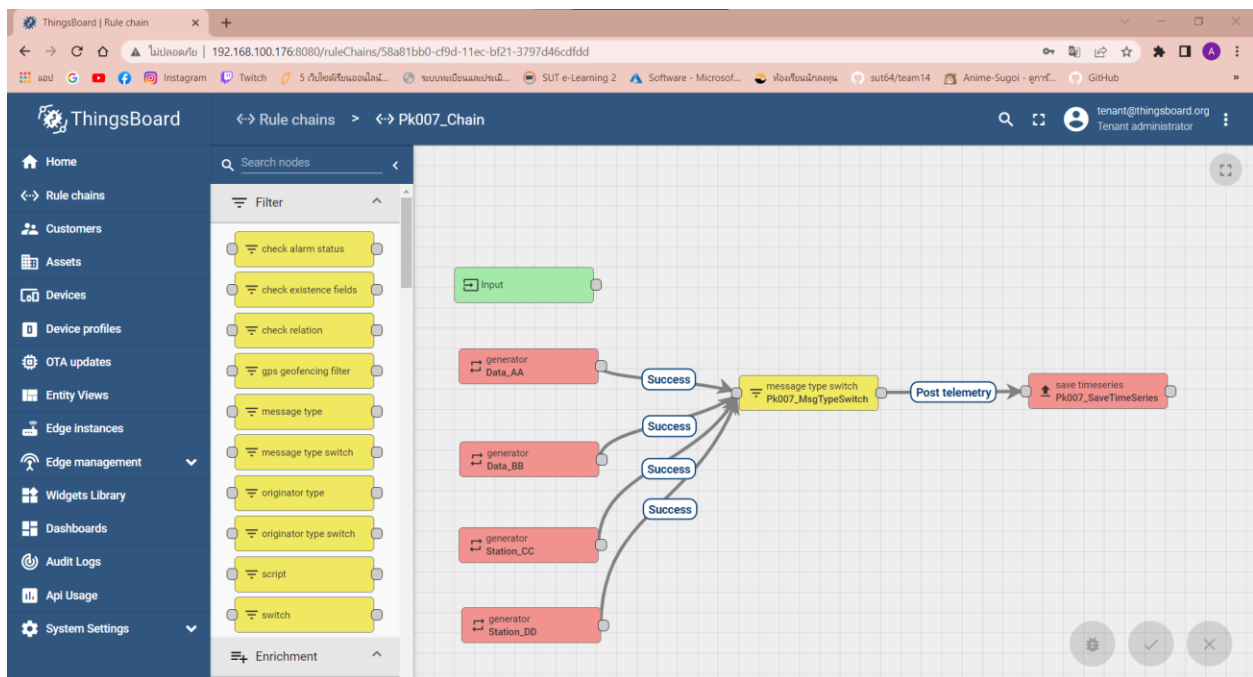
Devices

Device profile
All

+ ↺ 🔍

<input type="checkbox"/>	Created time ↓	Name	Device profile	Label	Customer	Public	Is gateway	
<input type="checkbox"/>	2022-05-09 21:16:15	Station_DD	default			<input type="checkbox"/>	<input type="checkbox"/>	     
<input type="checkbox"/>	2022-05-09 21:16:05	Station_CC	default			<input type="checkbox"/>	<input type="checkbox"/>	     
<input type="checkbox"/>	2022-05-09 20:37:46	Pk_Station_BB	default			<input type="checkbox"/>	<input type="checkbox"/>	     
<input type="checkbox"/>	2022-05-09 20:37:29	Pk_Station_AA	default			<input type="checkbox"/>	<input type="checkbox"/>	     

2.เพิ่ม generator ใน rule chain อีก 2 อันดังภาพ



Generator: Station_CC


```
var msg = {  
  station_name : "MyHome",
```

```
latitude : 14.879326867844068,
longitude : 102.03256640847032,
temperature: (Math.random()*10+40).toFixed(2),
humidity: (Math.random()*10+80).toFixed(2) };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";
return { msg: msg, metadata: metadata, msgType: msgType };
```

Generator: Station_DD

```
var msg = {
station_name : "SaveOne Market",
latitude : 14.958084879144675,
longitude : 102.04416091950054,
temperature: (Math.random()*10+40).toFixed(2),
humidity: (Math.random()*10+80).toFixed(2) };
var metadata = { data: 40 };
var msgType = "POST_TELEMETRY_REQUEST";
return { msg: msg, metadata: metadata, msgType: msgType };
```

3.เพิ่ม Entity Aliase อีก 2 อัน และแก้ไข All_Node โดยการเพิ่ม Device ที่สร้างใหม่เพิ่มเข้าไป



Alias name *

Station_CC

Resolve as multiple ent ☐

Filter type *

Single entity

Type *

Device ▼ **Device ***

Device is required.

4.เพิ่มหน้า dashboard โดย Edit → Manage dashboard states

Title *

Manage dashboard states

Name *

{entityName}

State Id *

MyHome

☐ Root state

Name *

{entityName}

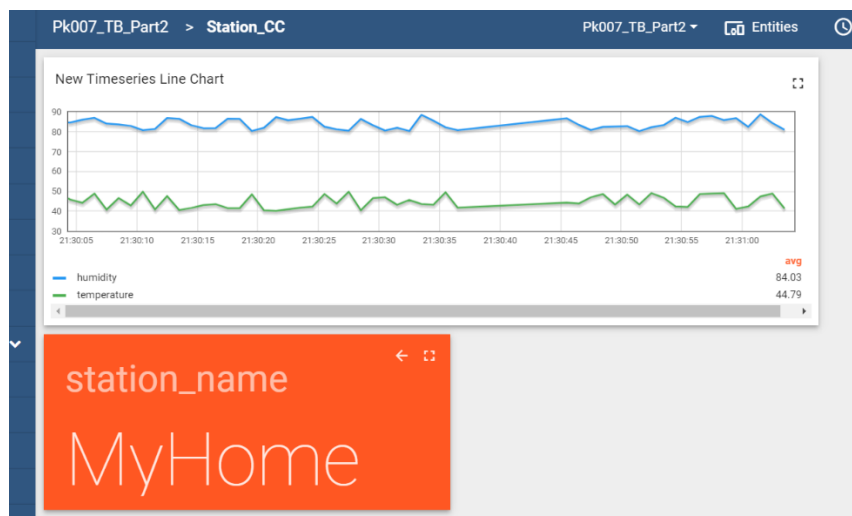
State Id *

SaveOne_Market

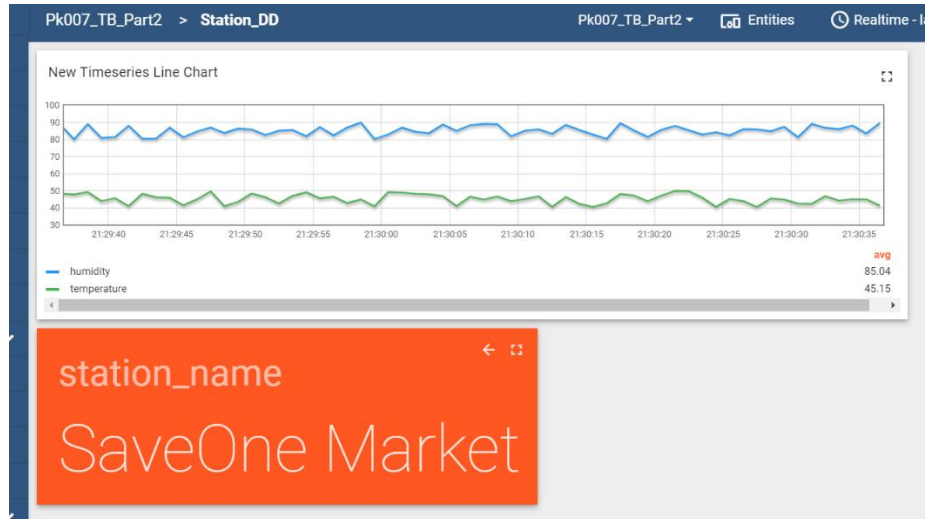
☐ Root state

4.เข้าไปแก้ไขหน้า dashboard ที่สร้างขึ้นใหม่ของแต่ละที่ดังรูป

MyHome



SaveOne Market



5.แก้ไข Device Admin Table ในแถบ Action โดยแก้ไขที่ function ที่ใช้

Action source *

Action cell button

Name *

Cell_AdminTable

Icon

play_arrow

☐ Show/hide action using function

Type *

Custom action

function (\$event, widgetContext, entityId, entityName, additionalParams, entityLabel) {

```
1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext.servicesMap.get
  ('deviceService'));
3 deviceService.getDevice(entityId.id).subscribe(function(device) {
4 if (device.name == 'Pk Station AA') {
```

```
var $injector = widgetContext.$scope.$injector;
let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));
deviceService.getDevice(entityId.id).subscribe(function(device) {
if (device.name == 'Pk_Station_AA') {
openDashboardState('SUT_Data',device)}
```

```

else if(device.name == 'Pk_Station_BB') {
  openDashboardState('STP_Data',device)}}
else if(device.name == 'Station_CC') {
  openDashboardState(MyHome',device)}}
else if(device.name == 'Station_DD') {
  openDashboardState('SaveOne_Market',device)}}
});
function openDashboardState(stateId,device) {
  var params = {
    entityId: entityId,
    entityName: device.name;
  };
  widgetContext.stateController.openState(stateId, params, false);}

```

6.แก้ไข Action ของ google map ในส่วนของ Sensor_Details โดยเพิ่มโค้ดที่จะโยนไป Dashboard ที่สร้างขึ้นใหม่

Action source *

On marker click

Name *

Sensor_Details



Icon

play_arrow

Type *

Custom action

```
function ($event, widgetContext, entityId, entityName, additionalParams, entityLabel) {
```

Tidy ?

```

1 var $injector = widgetContext.$scope.$injector;
2 let deviceService = $injector.get(widgetContext.servicesMap.get
  ('deviceService'));
3 deviceService.getDevice(entityId.id).subscribe(function(device) {
4   if (device.name == 'Pk_Station_AA') {
5     openDashboardState('SUT_Data',device)

```

```
var $injector = widgetContext.$scope.$injector;
```

```
let deviceService = $injector.get(widgetContext.servicesMap.get('deviceService'));
```

```
deviceService.getDevice(entityId.id).subscribe(function(device) {
```

```
if (device.name == 'Pk_Station_AA') {
```

```
openDashboardState('SUT_Data',device)}
```

```
else if(device.name == 'Pk_Station_BB') {
```

```
openDashboardState('STP_Data',device)}
```



```

else if(device.name == 'Station_CC') {

openDashboardState('MyHome',device)}

else if(device.name == 'Station_DD') {

openDashboardState('SaveOne_Market',device)}

});

function openDashboardState(stateId,device) {

var params = {

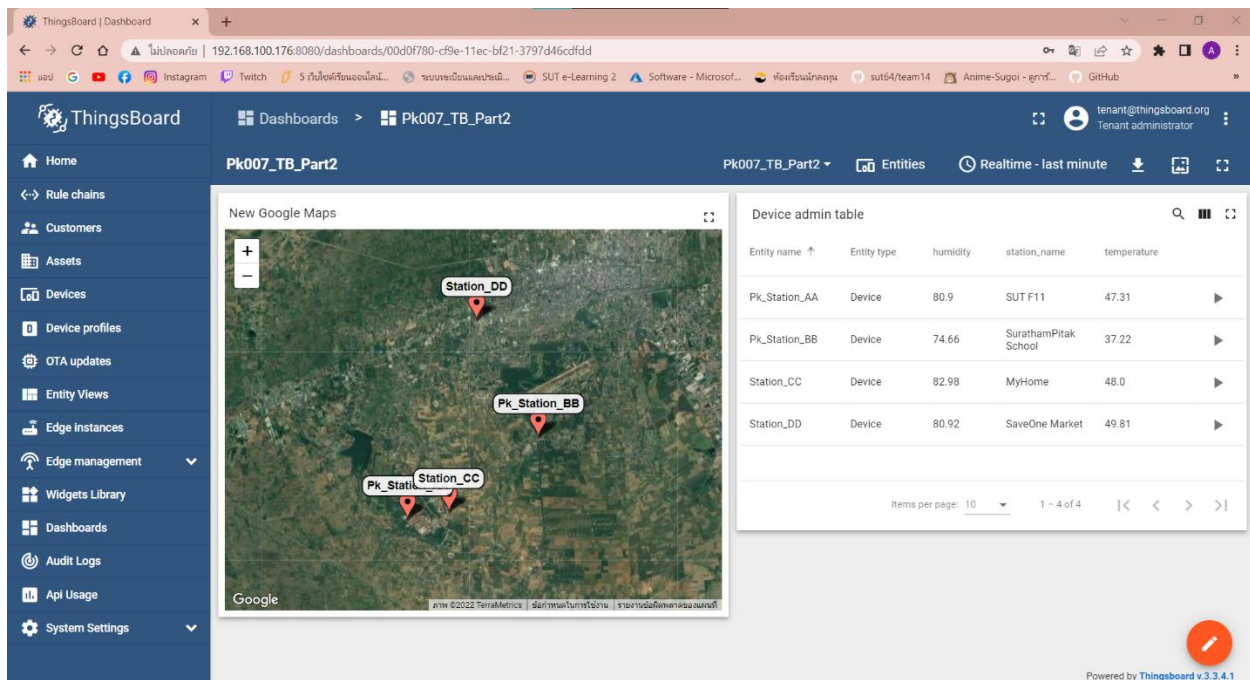
entityId: entityId,

entityName: device.name};

widgetContext.stateController.openState(stateId, params, false);}

```

7.ผลลัพธ์ที่ได้จะเป็นดังนี้



The screenshot shows the ThingsBoard dashboard interface. The left sidebar contains navigation options: Home, Rule chains, Customers, Assets, Devices, Device profiles, OTA updates, Entity Views, Edge Instances, Edge management, Widgets Library, Dashboards, Audit Logs, Api Usage, and System Settings. The main area displays a dashboard titled 'Pk007_TB_Part2'. It features a 'New Google Maps' widget showing a satellite map with four red location pins labeled 'Station_DD', 'Pk_Station_BB', 'Pk_Station_CC', and 'Pk_Station_AA'. To the right of the map is a 'Device admin table' widget displaying a table of device data.

Entity name	Entity type	humidity	station_name	temperature
Pk_Station_AA	Device	80.9	SUT F11	47.31
Pk_Station_BB	Device	74.66	SurathamPitak School	37.22
Station_CC	Device	82.98	MyHome	48.0
Station_DD	Device	80.92	SaveOne Market	49.81

At the bottom right, there is a red circular button with a white pencil icon, and a footer indicating 'Powered by Thingsboard v.3.3.4.1'.