



Online Training

IoT & NETPIE 2020

Mr. Piyawat Jomsathan [Tae]

Cyber-Physical Systems [CPS]

National Electronics and Computer Technology Center [NECTEC], Thailand





Contact Trainer

Mr.Piyawat Jomsathan (Tae)
Cyber – Physical Systems (CPS)
NECTEC Thailand
E-mail : piyawat.jom@nectec.or.th
Tel. 02-564-6900 ext. 2469

Schedule

1

MODULE 1

IoT & NETPIE

2

MODULE 2

NETPIE 2020 Portal

3

MODULE 3

Device Data Management

4

MODULE 4

NETPIE 2020 Freeboard

5

MODULE 5

Restful API

6

MODULE 6

Conclusion NETPIE 2020



Module 1

IoT & NETPIE



แนวคิดของ Internet of Things

แนวคิด Internet of Things นั้นมาจาก Kevin Ashton ในปี ค.ศ.1999 โดยได้เสนอแนวคิดว่า

“การนำสิ่งของต่างๆ ไม่ว่าจะเป็นคอมพิวเตอร์, เครื่องจักร และอุปกรณ์ตรวจจับมาเชื่อมต่อกับเครือข่ายอินเทอร์เน็ต เพื่อรายงานสถานะการทำงาน สถานะข้อมูล และรับรู้คำสั่งควบคุม”

โดยในช่วงเวลานั้นโลกเพิ่งรู้จักอินเทอร์เน็ตได้ไม่นาน และก็มีคนนำแนวคิด IoT ไปใช้ต่อ และมีชื่อเรียกที่แตกต่างไป เช่น

- Machine to Machine [M2M]
- Ubiquitous Computing
- Embedded Computing
- Smart Service
- Industrial Internet



Kevin Ashton
ผู้เสนอแนวคิด IoT

➤ ความหมายและความสำคัญของ IoT

IoT —————> “อินเทอร์เน็ตในทุกสิ่ง”

I = Internet —————> อินเทอร์เน็ต

O = Of —————> ของ/ใน

T = Things —————> ทุกสิ่ง/สิ่งต่างๆ

ความหมายและความสำคัญของ IoT



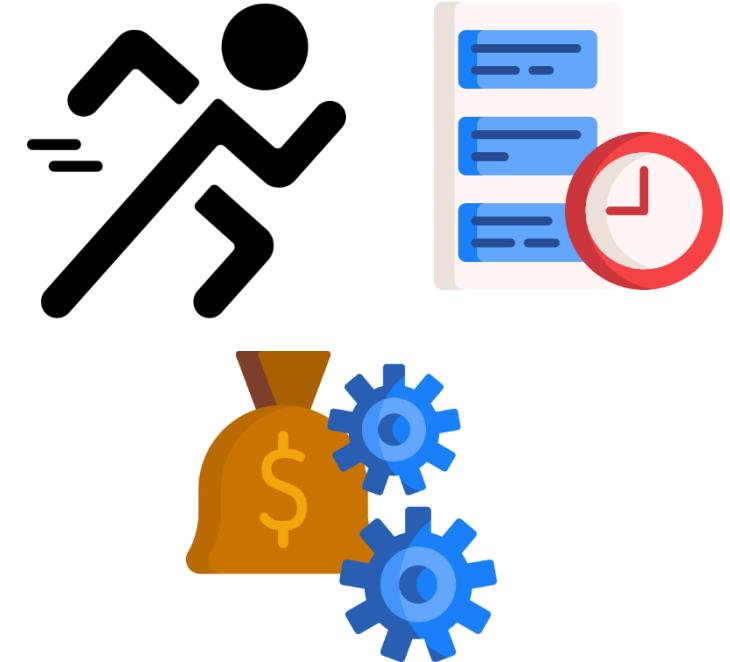
ប័ណ្ណិតទូទៅ IoT

1. เพิ่มความសម្រាប់ការងារក្នុងការងារនិងជំនើយ
2. เพិនគម្រោគនិងពារិភាពក្នុងការងារ
3. លើកទូទៅនៃការងារ ដែលបានប្រើប្រាស់ IoT



ប័ណ្ណិតទូទៅ IoT

1. ការវាទីរបបអិនកេវលិនិត្យនៅក្នុងការងារ
2. របៀបការងារដែលត្រួតពិនិត្យនៅក្នុងការងារ
3. ការប្រើប្រាស់ការងារដែលត្រួតពិនិត្យនៅក្នុងការងារ

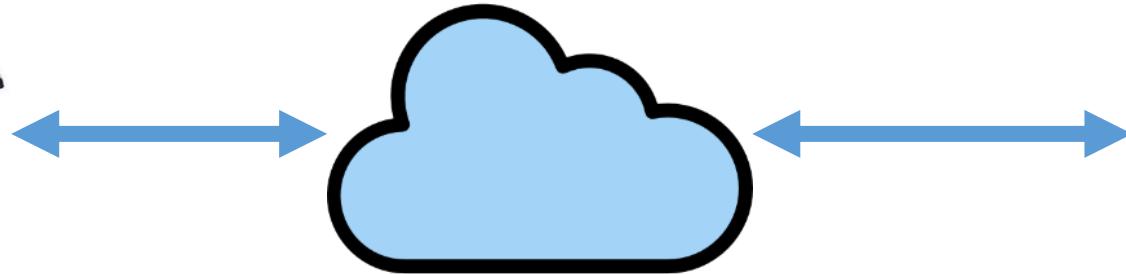


องค์ประกอบของ IoT

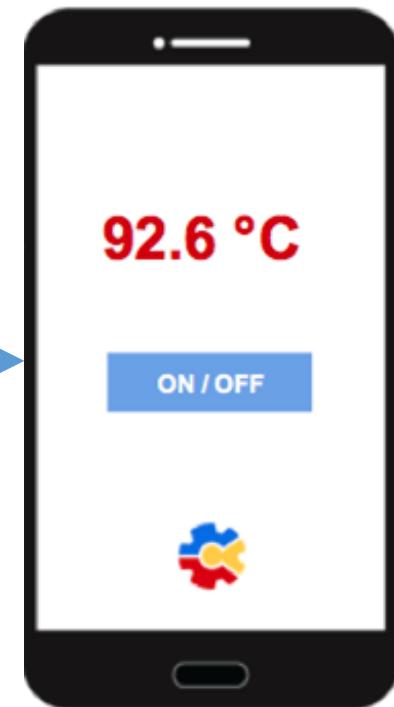
สิ่งที่ต้องการจากหม้อหุงข้าว ดีอ

1. อุณหภูมิในหม้อหุงข้าว
2. ควบคุมการทำงาน [ON/OFF]

สามารถดูอุณหภูมิหม้อหุงข้าว และ
ควบคุมการทำงานผ่านมือถือ

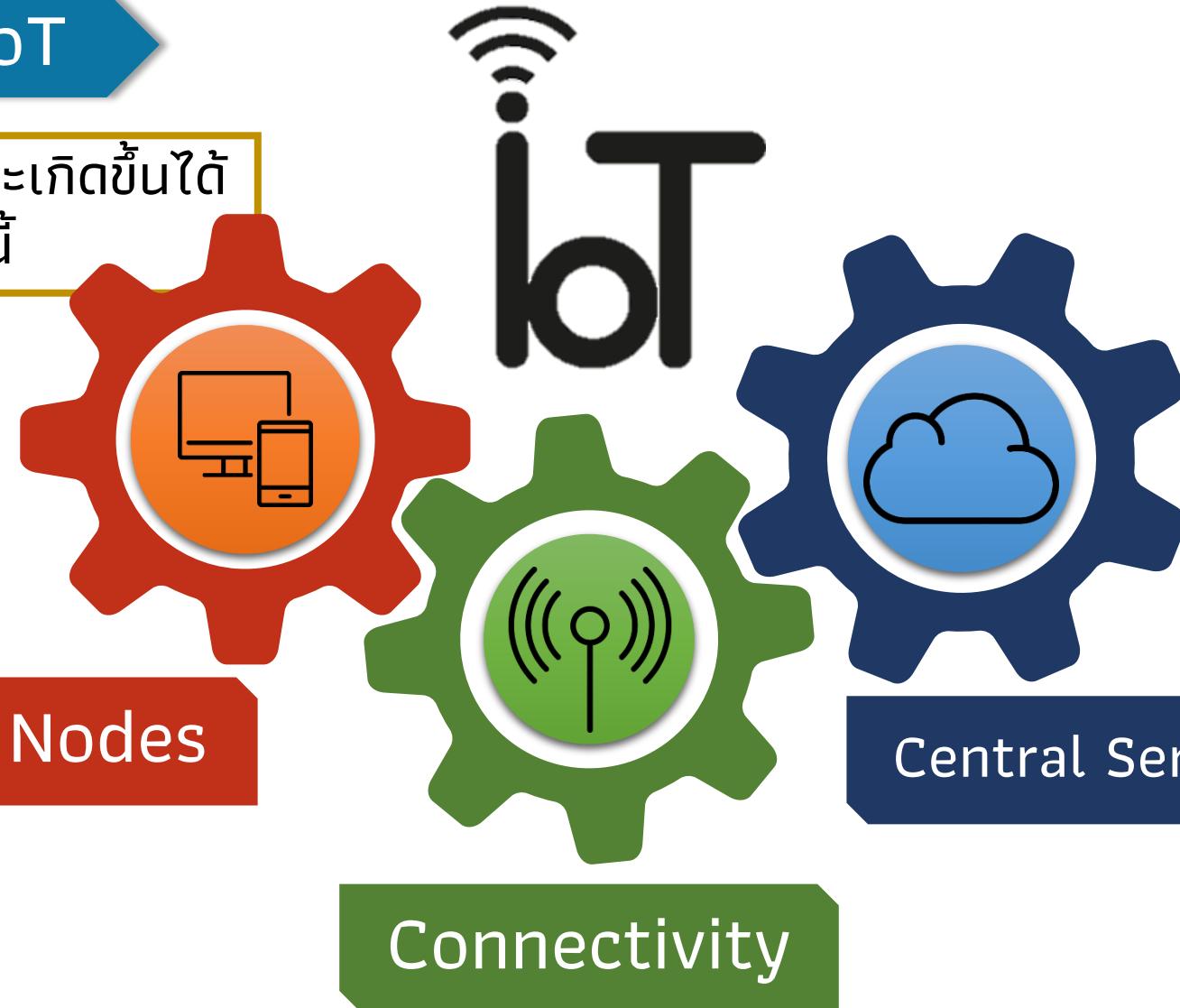


ส่งข้อมูลผ่าน Internet ไปยัง Cloud



องค์ประกอบของ IoT

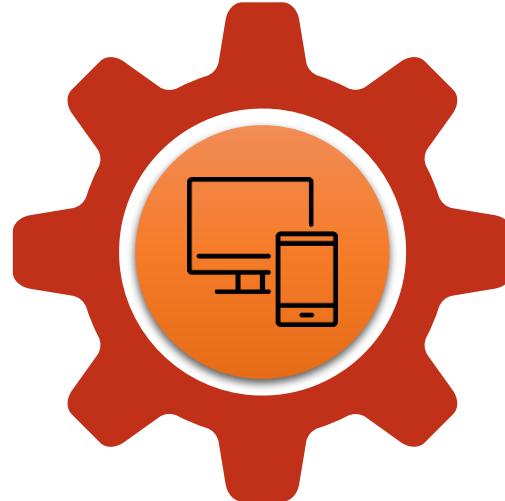
ระบบหรือเทคโนโลยี IoT จะเกิดขึ้นได้
ต้องมีองค์ประกอบ 3 ส่วน ดังนี้



Module 1 : IoT & NETPIE

องค์ประกอบของ IoT

End Nodes



End Nodes គឺ ស៊រុនខែងស៊ែងខែង[ឥឡូឌី, មាខពើទូរ, មេដូឡូ]និងអ្នកប្រើប្រាស់បញ្ជីនៃការគិតថ្លែងក្នុងផ្ទាល់ពាណិជ្ជកម្ម។ វាបានរួចរាល់ជាការត្រួតពិនិត្យ តាមរយៈការប្រើប្រាស់ការពារមិនពារតាមការប្រើប្រាស់ការពារដែលបានរាយការណ៍ឡើង។ នៅពេលនេះ ការប្រើប្រាស់ការពារដែលបានរាយការណ៍ឡើង ត្រូវបានប្រើប្រាស់ជាការត្រួតពិនិត្យ តាមរយៈការប្រើប្រាស់ការពារដែលបានរាយការណ៍ឡើង។ នៅពេលនេះ ការប្រើប្រាស់ការពារដែលបានរាយការណ៍ឡើង ត្រូវបានប្រើប្រាស់ជាការត្រួតពិនិត្យ តាមរយៈការប្រើប្រាស់ការពារដែលបានរាយការណ៍ឡើង។

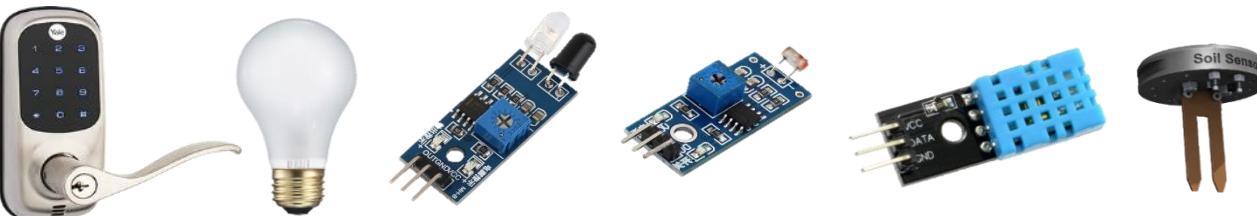
Microcontrollers



Single Board



Actuators and Sensors



องค์ประกอบของ IoT

Connectivity



Connectivity หรือ ระบบเชื่อมต่ออินเทอร์เน็ต คือ ระบบที่ทำให้อุปกรณ์และสิ่งของสามารถเชื่อมต่อกับระบบอินเทอร์เน็ต เพื่อทำการรับและส่งข้อมูลระหว่างอุปกรณ์ไปยัง Cloud โดยระบบเชื่อมต่ออินเทอร์เน็ตนั้นมีหลากหลายประเภทเป็นแบบมีสายหรือไร้สายก็ได้ ขึ้นอยู่กับความเหมาะสมในการใช้งาน



องค์ประกอบของ IoT

Central Servers/Cloud



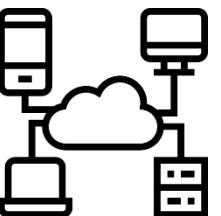
Central Servers/Cloud คือ พื้นที่จัดเก็บข้อมูลเพื่อนำมาสร้างเป็น Web Server, Application Server เป็นต้น โดยปัจจุบันสำหรับระบบ IoT นิยมใช้ Cloud เป็นพื้นที่จัดเก็บข้อมูลโดยที่ Server และ Cloud มีความแตกต่างกันดัง

Server

- :::::
- :::::
- :::::

คือ เครื่องหรือโปรแกรมคอมพิวเตอร์ซึ่งทำงานให้บริการในระบบ เครือข่ายแก่ลูกข่าย เครื่องคอมพิวเตอร์ที่ทำหน้าที่เป็นเซิร์ฟเวอร์นี้ควร มีประสิทธิภาพสูง มีความเสถียร สามารถให้บริการแก่ผู้ใช้ได้เป็น จำนวนมาก

Cloud

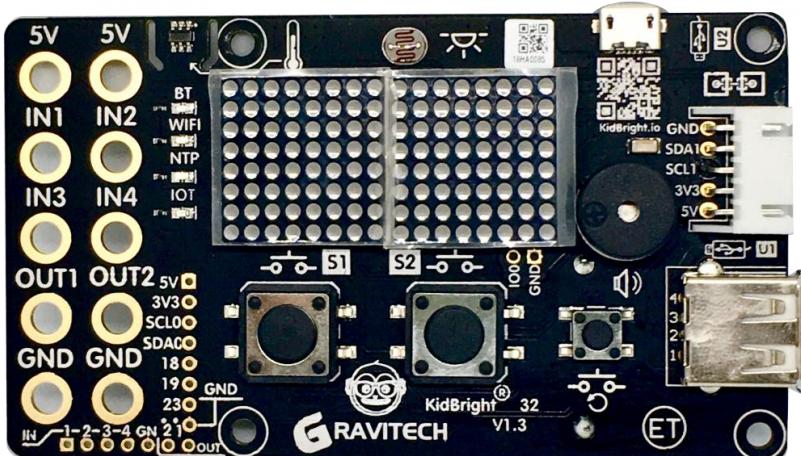


คือ การนำ Server หลายๆเครื่องมาทำงานด้วยกันโดยมีหน่วยประมวลผลระดับสูง โดย Cloud สามารถสร้าง Service ขึ้นมา ทำงานได้หลากหลาย ในปัจจุบัน Cloud มีผู้ให้บริการหลากหลายทั้ง แบบมีและไม่มีค่าใช้จ่าย

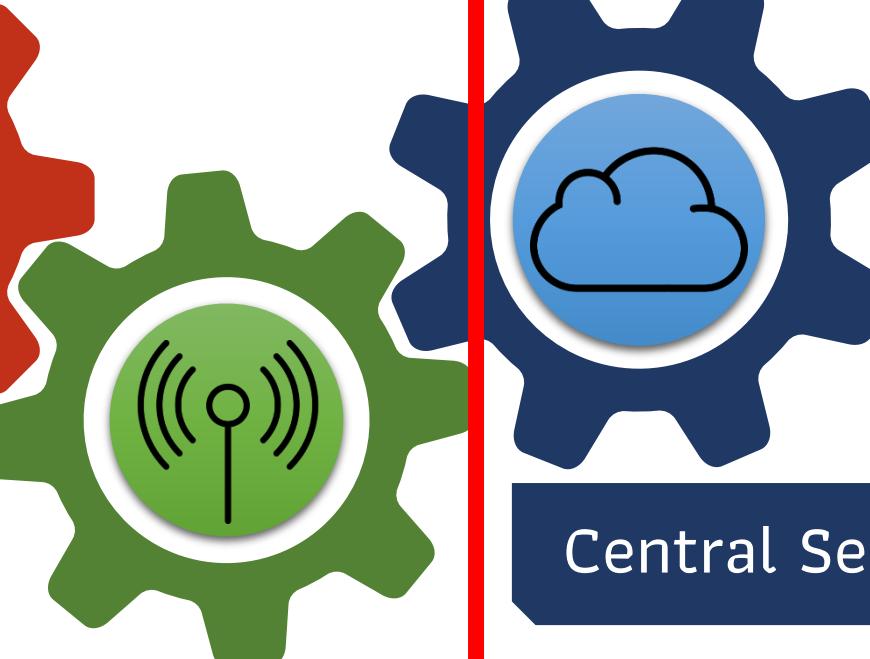
Module 1 : IoT & NETPIE

องค์ประกอบของ IoT

End Nodes



Connectivity

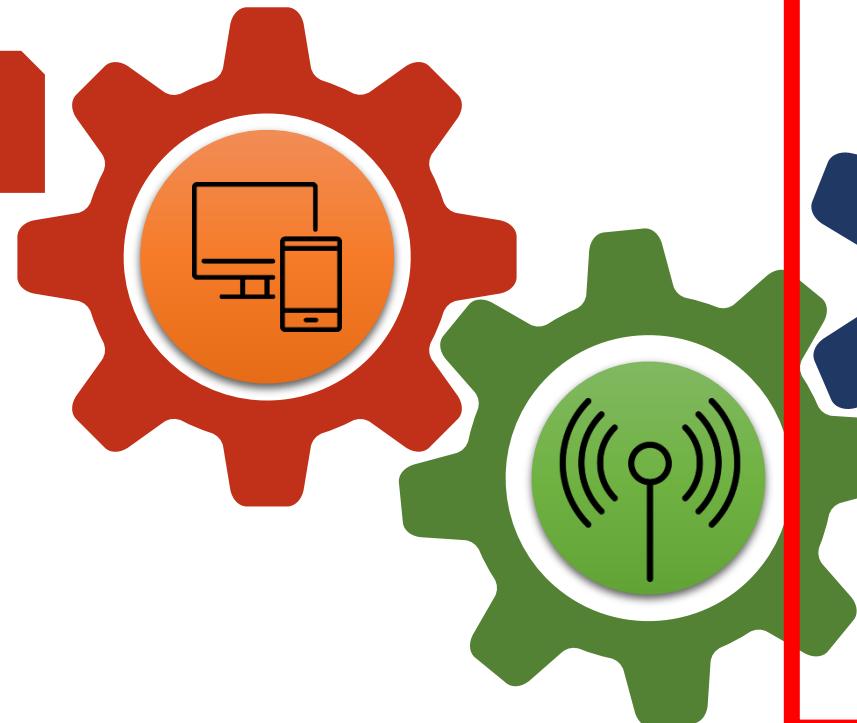


Central Servers/Cloud

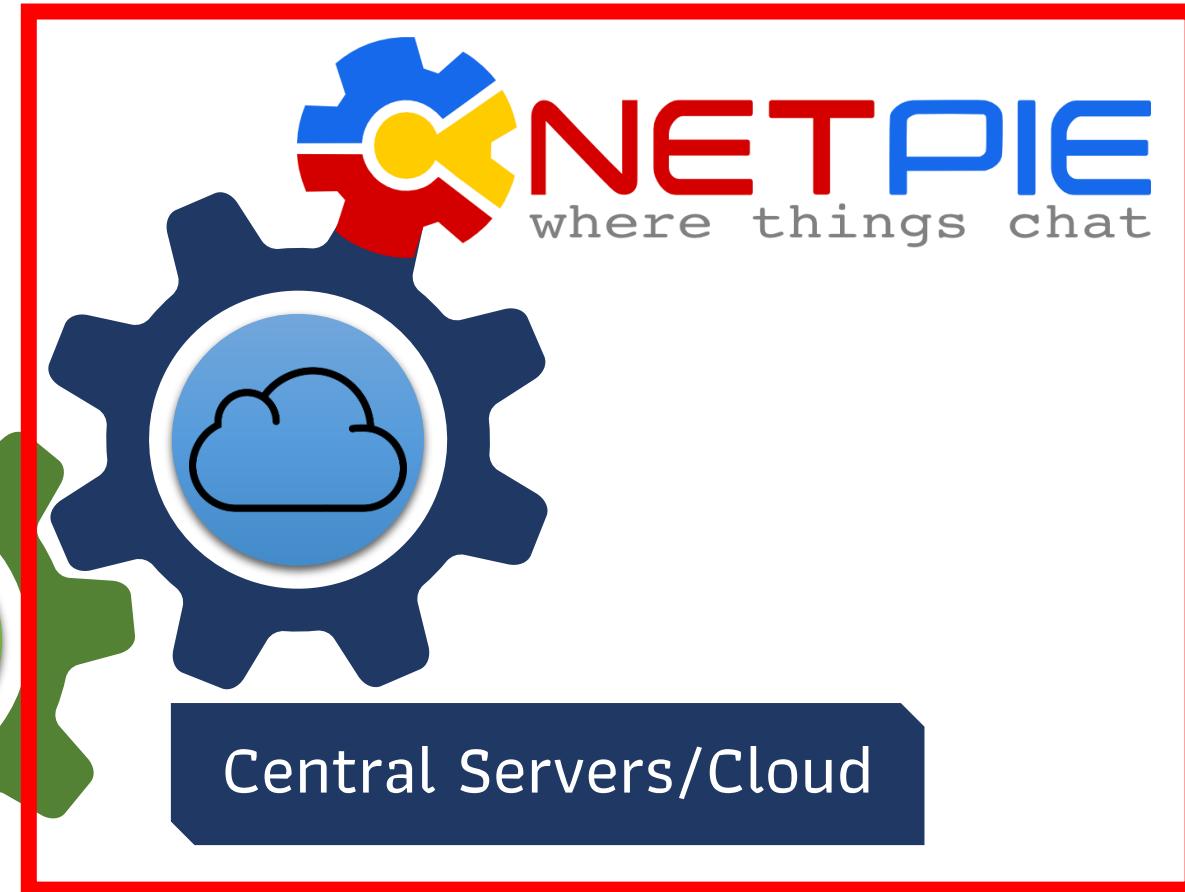
Module 1 : IoT & NETPIE

ឯងគែរករកបែនខំ IoT

End Nodes



Connectivity



Central Servers/Cloud

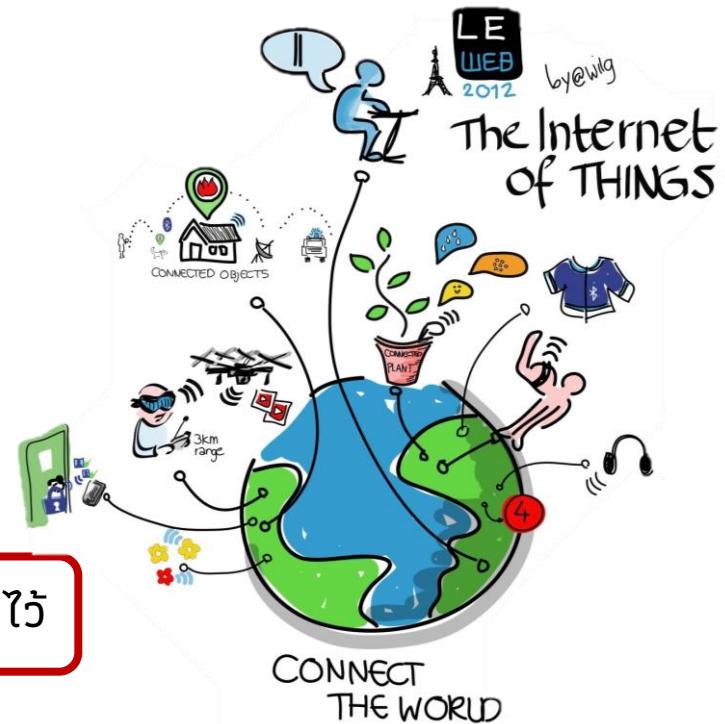
Module 1 : IoT & NETPIE



NETPIE គឺអេឡិចត្រូនីមួយៗ ??

គុណសមបាតិអត្ថក្រុង NETPIE

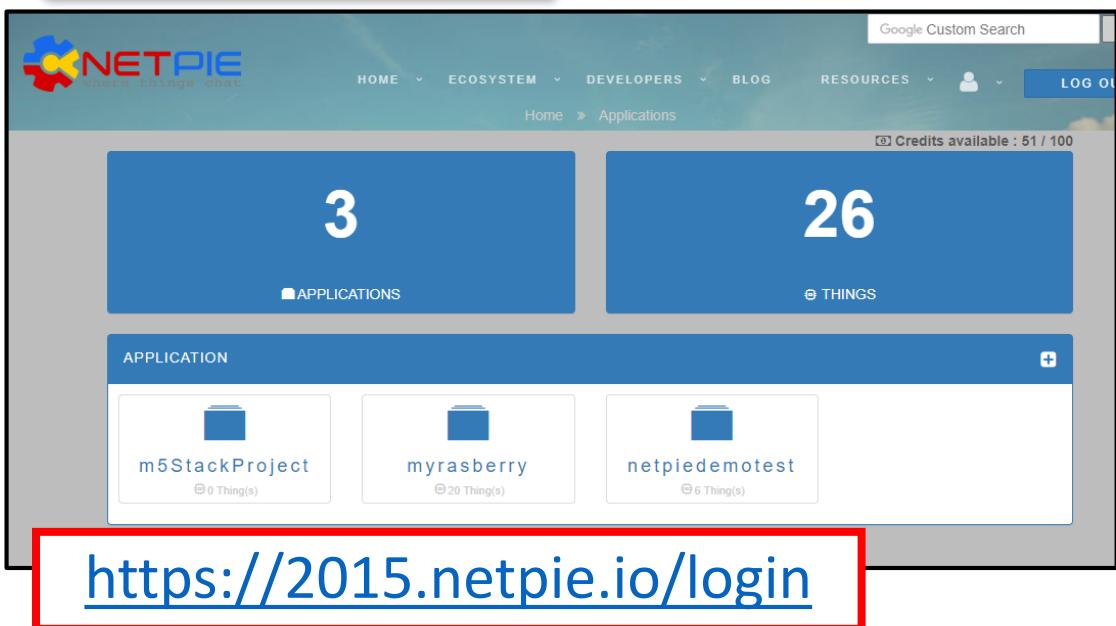
1. Monitoring คือ การแสดงค่าข้อมูลของอุปกรณ์หรือเซนเซอร์แบบ Real-Time
 2. Controlling คือ การควบคุมการทำงานอุปกรณ์ต่างๆ ผ่าน Cloud Platform
 3. Data Storage คือ การเก็บข้อมูลที่ได้จากเซนเซอร์หรืออุปกรณ์
 4. Notification คือ การแจ้งเตือนความผิดปกติของเซนเซอร์หรืออุปกรณ์จากที่ได้กำหนดไว้



Module 1 : IoT & NETPIE

NETPIE 2015

ថ្មីប៉ុណ្ណោះ NETPIE ក្នុងពេលវេលាទីនៅមាត្រា 2 Version គឺ
NETPIE 2015 និង NETPIE 2020



The screenshot shows the NETPIE 2020 web interface. It displays two projects: 'Project_Docume...' (1 device, 0 groups) and 'Smart_Factory_I...' (6 devices, 1 group). The URL <https://auth.netpie.io/login> is highlighted with a red border at the bottom right. The bottom right corner features a blue circular button with a white plus sign (+).

Module 1 : IoT & NETPIE

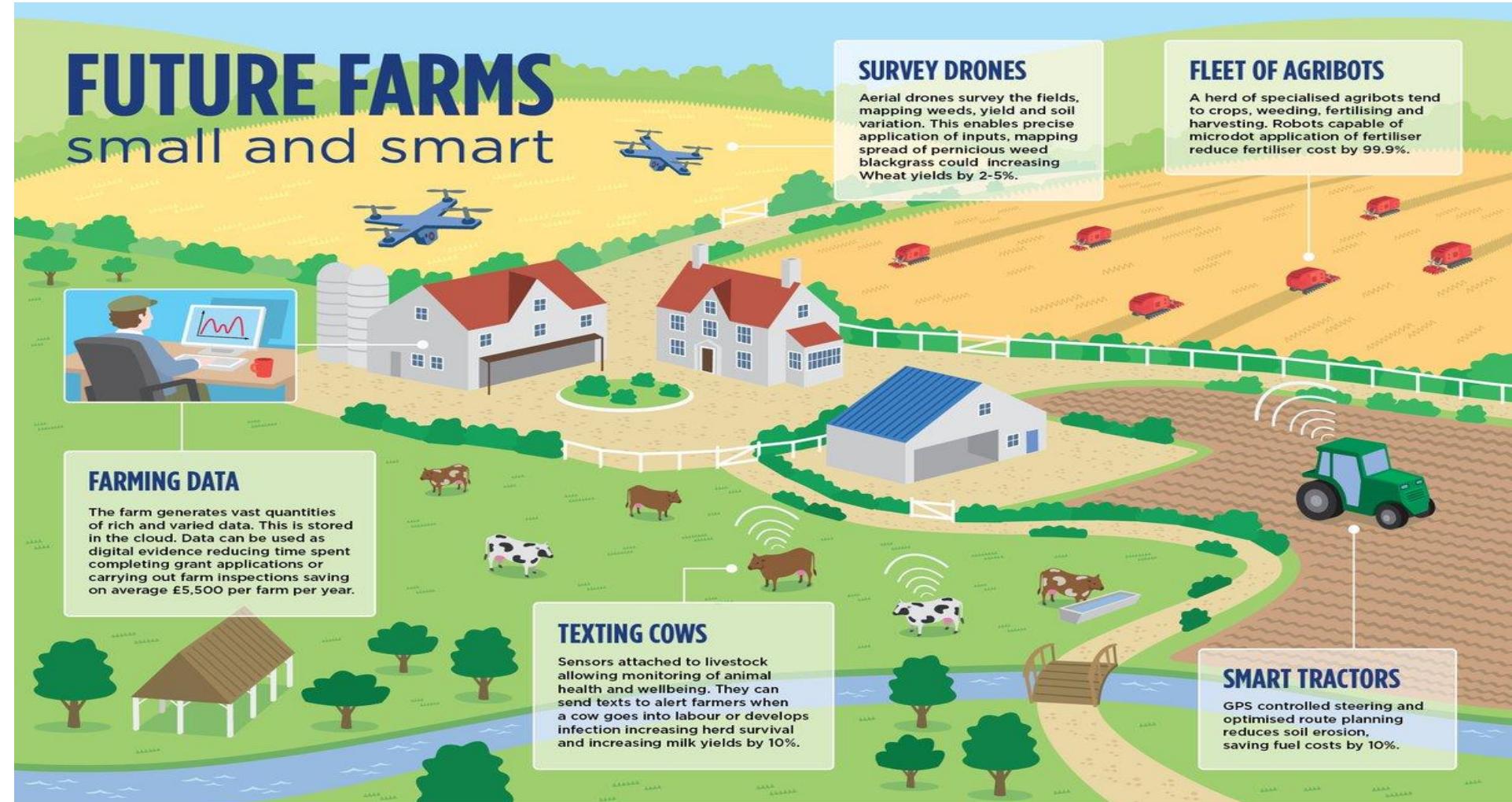
เปรียบเทียบ NETPIE 2020 และ NETPIE 2015

	NETPIE 2020	NETPIE 2015
Design Philosophy	Platform - Centric	Device – Centric
Commercial Ability	Commercial – Ready	Need to re-program Hardware migrate to commercial platform
Suitable Usage	Mass production, Project-based	Project-based
Target User	IoT consumer product makers, Hobbyists, Students	Makers, hobbyists, students
Communication Protocol	MQTT, HTTP	Microgear
Programming Language	Any languages with MQTT library support	Limited to Microgear library
Hardware Support	Unlimited as long as it supports MQTT	Limited to those with Microgear support
Device Identity and Group	No APPID Device identity and group can be adjusted after product is sold/installed	Use APPID Device identity and group must be programmed into firmware
Rate Limit	Allow burst	Everyone is subject to the same rate limit
Trigger	Can set trigger action in cloud platform	Set trigger action inside IoT devices

Module 1 : IoT & NETPIE

តัวอย่างการใช้งาน NETPIE

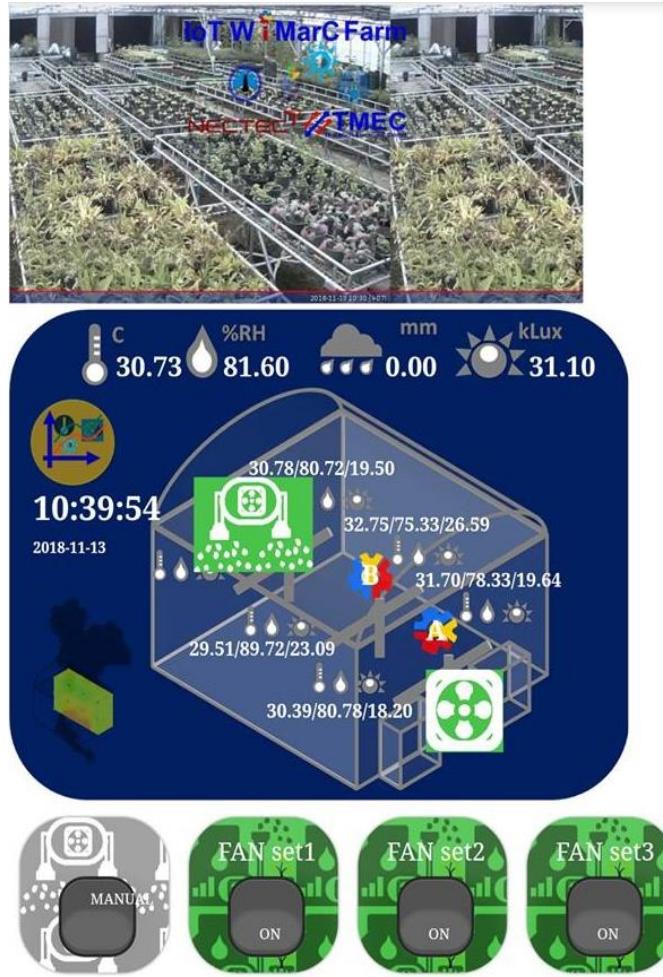
Smart Farm



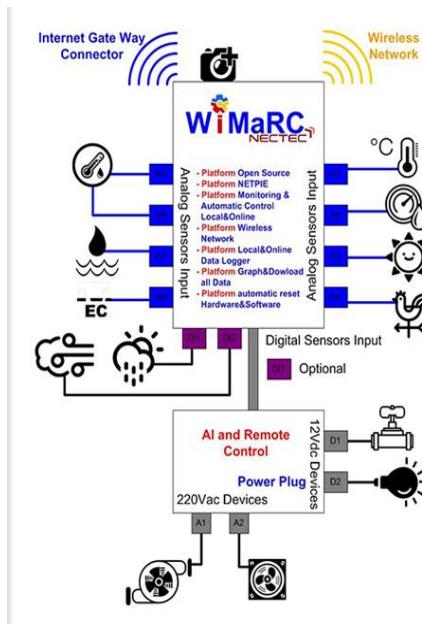
Module 1 : IoT & NETPIE

WiMaRC បរបច្បល់បន្ទាន់ប្រព័ន្ធប្រជាជាតិ

“វិវាយក” គឺជាដំឡើងមេដែលបានរចនាបានដើម្បីទូទាត់ប្រព័ន្ធអំពីការប្រើប្រាស់បច្ចេកទេសទៅកាន់បណ្តុះបណ្តាល និងការប្រើប្រាស់បច្ចេកទេសទៅកាន់ការប្រើប្រាស់បច្ចេកទេស។ វិវាយក គឺជាដំឡើងមេដែលបានរចនាបានដើម្បីទូទាត់ប្រព័ន្ធអំពីការប្រើប្រាស់បច្ចេកទេសទៅកាន់បណ្តុះបណ្តាល និងការប្រើប្រាស់បច្ចេកទេសទៅកាន់ការប្រើប្រាស់បច្ចេកទេស។

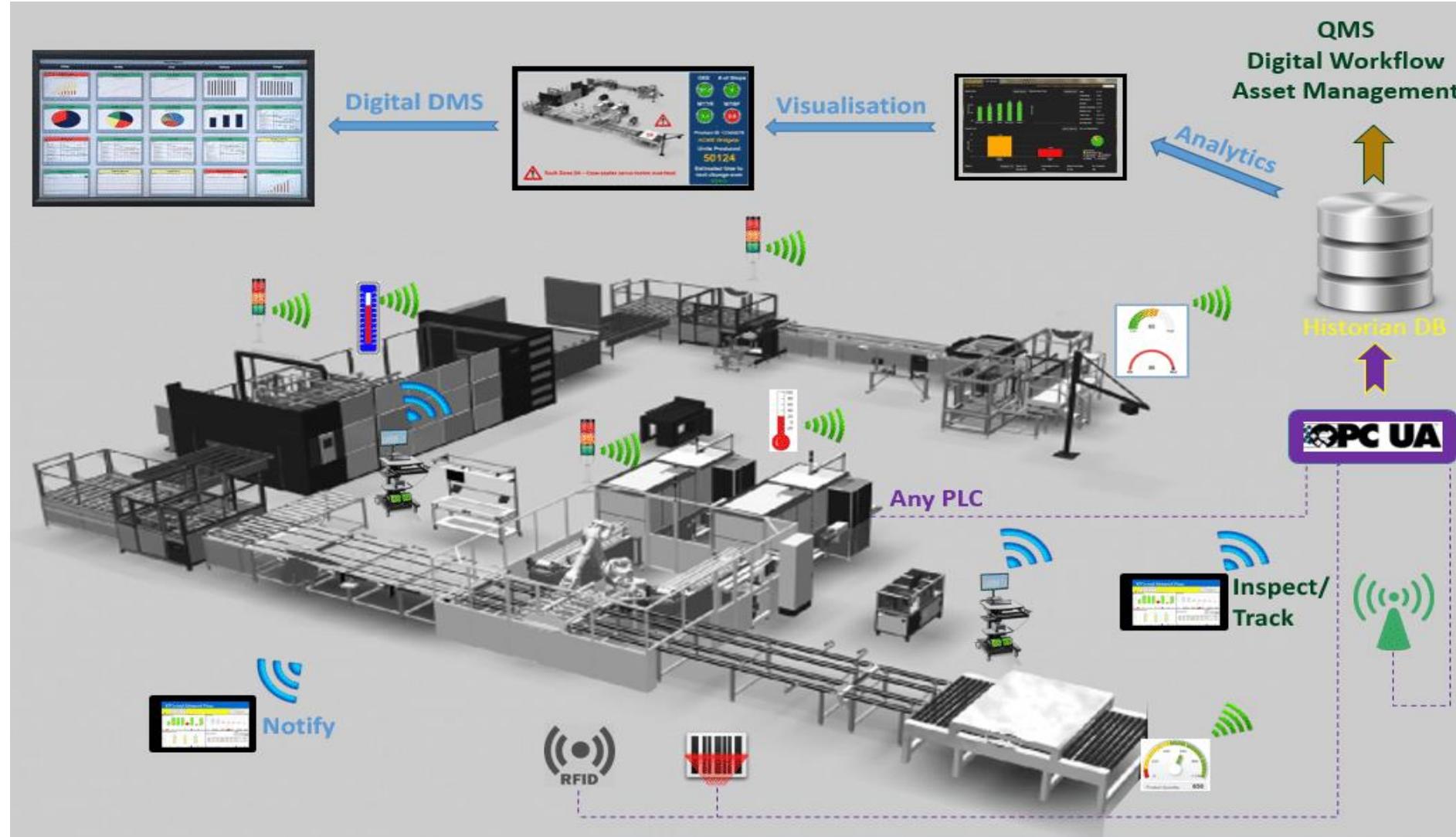


Wireless Sensor Network
for Management
and Remote Control



Module 1 : IoT & NETPIE

Smart Factory

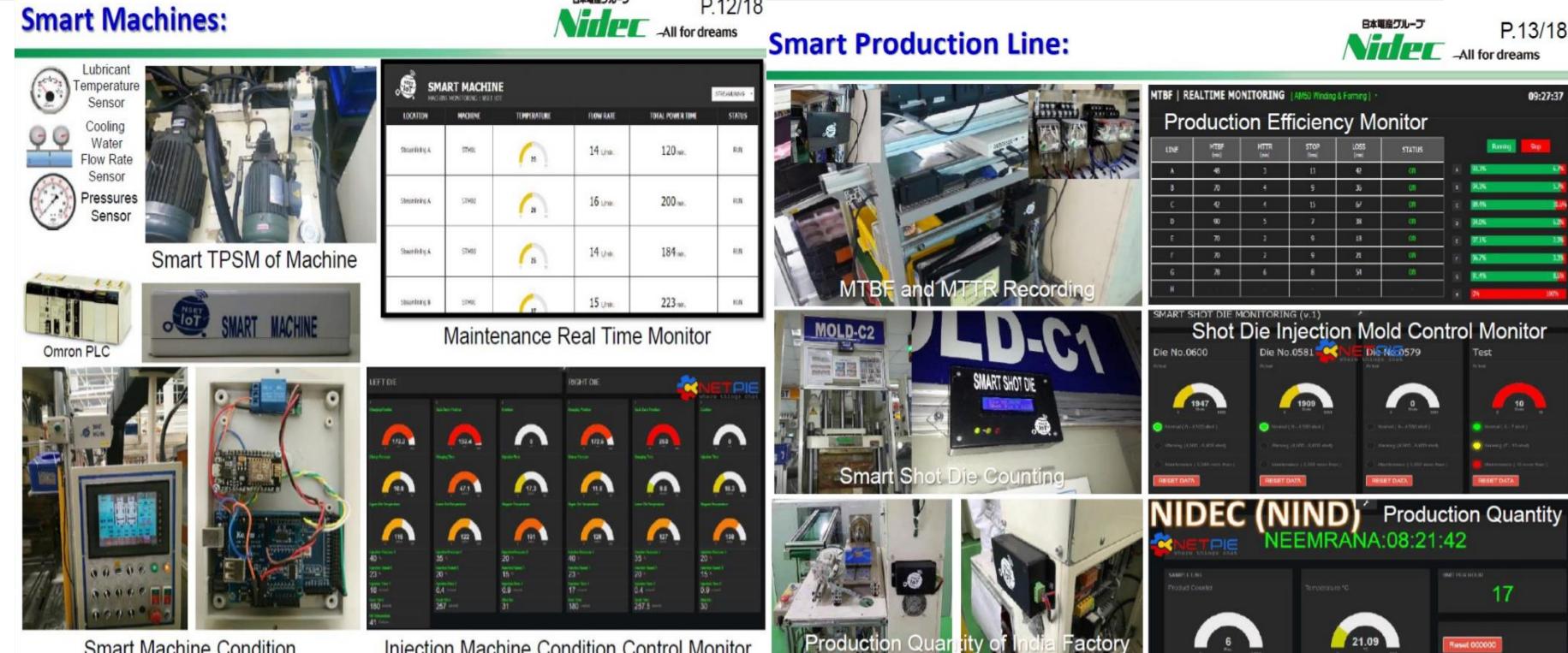


Module 1 : IoT & NETPIE

Smart Factory

บริษัท นีเด็ม ซิบารูรະ อเล็กทรอนิกส์ (ประเทศไทย)

การนำ IoT มาใช้ในโรงงานอุตสาหกรรม เริ่มตั้งแต่ส่วนของสิ่งอำนวยความสะดวกไปยังส่วนของการผลิต เช่น การแสดงสถานะห้องปิซูม และผลิตเดชั่นเปิด/ปิดไฟ ระบบควบคุมอุณหภูมิในห้องตรวจสอบคุณภาพ ไปจนถึงการมอนิเตอร์ค่าต่างของเครื่องจักร และการติดตามผลของฝ่ายการผลิต





Module 2

NETPIE 2020 Portal



เริ่มต้นการใช้งาน NETPIE2020

ลงทะเบียน NETPIE2020 ได้ที่
<https://authx.netpie.io/signup>



NETPIE
2020

EMAIL

required

NAME

required

ORGANIZATION

required

COUNTRY CODE

Thailand (+66)

MOBILE PHONE NUMBER* (NO COUNTRY CODE)

required and number only

I agree to the [Privacy Statement](#) and [Terms of Use](#)

SIGN UP

*Password will be sent to your mobile phone number.

เริ่มต้นการใช้งาน NETPIE2020

เมื่อสมัคร NETPIE2020
แล้วทำการเข้าสู่ระบบ

ซึ่งในปกติสามารถเข้าหน้า login ได้ที่

<https://authx.netpie.io/login>

NETPIE
2020

Connect Everything

Username (Email Address)

Password

SIGN IN

FORGET PASSWORD?

SIGN UP

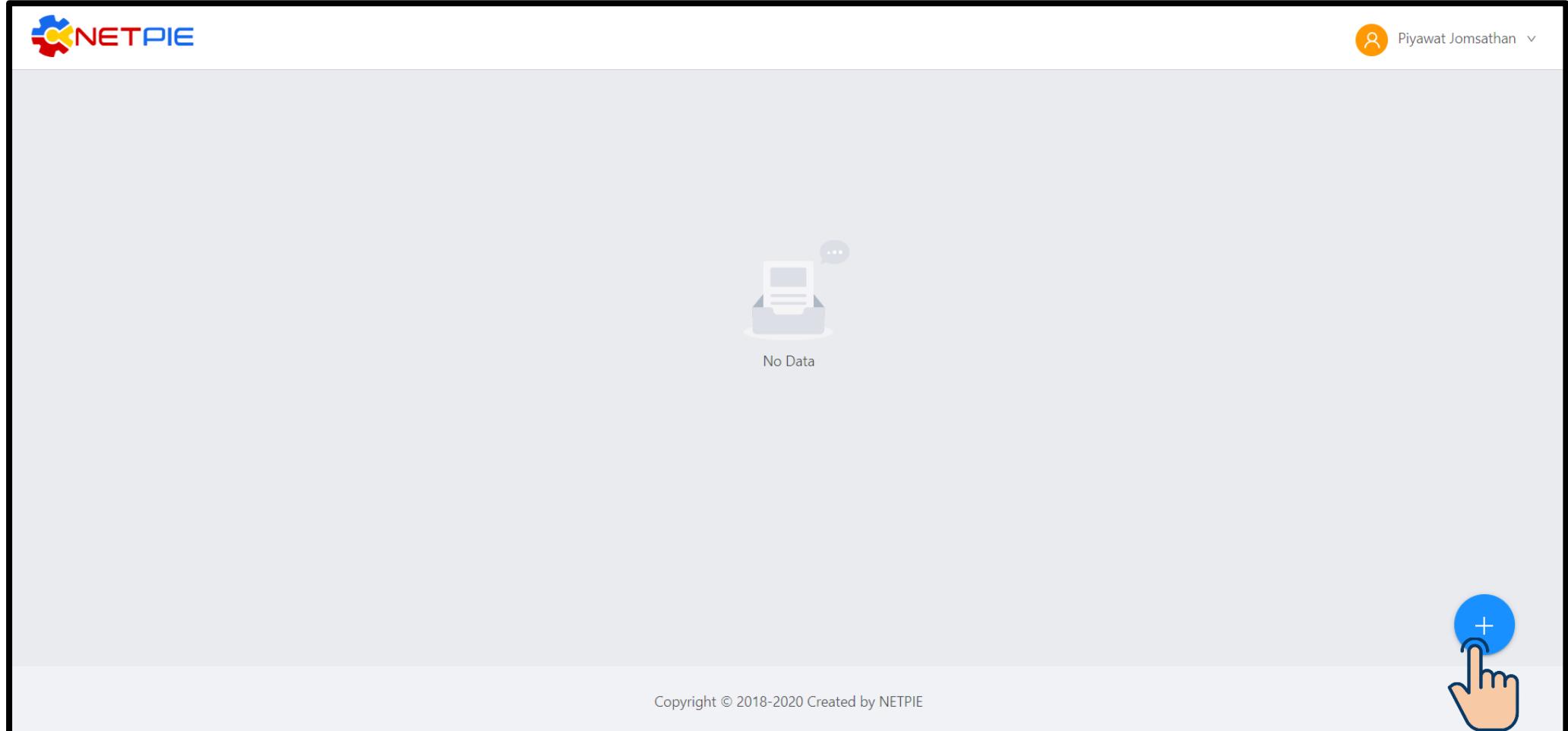
เริ่มต้นการใช้งาน NETPIE2020

The screenshot shows the main interface of the NETPIE 2020 Portal. At the top left is the NETPIE logo, and at the top right is a user profile icon labeled "Piyawat Jomsathan". The central area displays a placeholder icon with three dots and the text "No Data". A large blue banner at the bottom left contains the text "เมื่อเข้ามาแล้วพบหน้าต่างดังรูป ถือว่า Login สำเร็จ". In the bottom right corner of the main content area is a small blue circle with a white plus sign.

Copyright © 2018-2020 Created by NETPIE

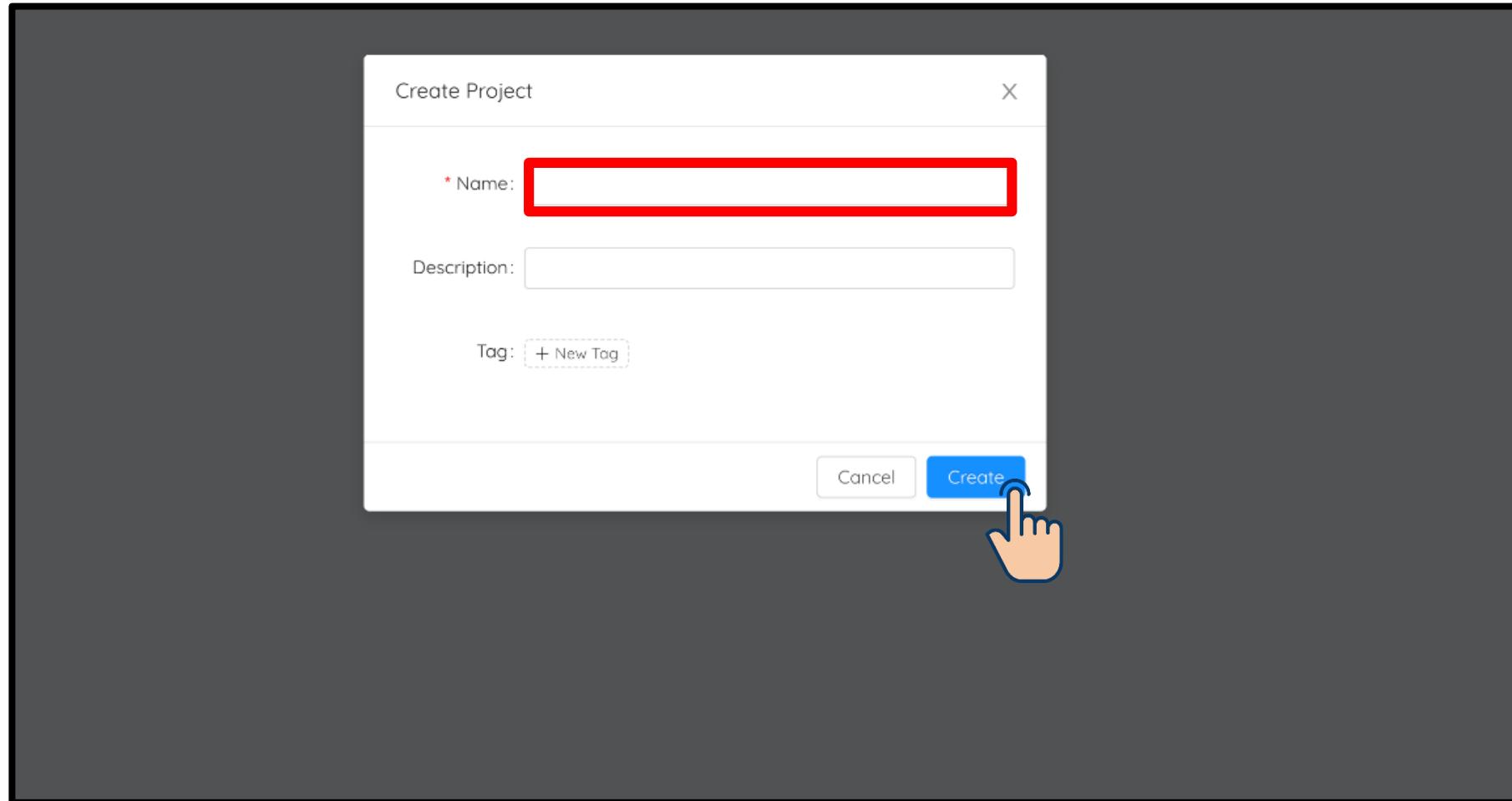
การสร้าง Project

โดยคลิกที่เครื่องหมาย + ดังรูป เพื่อสร้าง Project



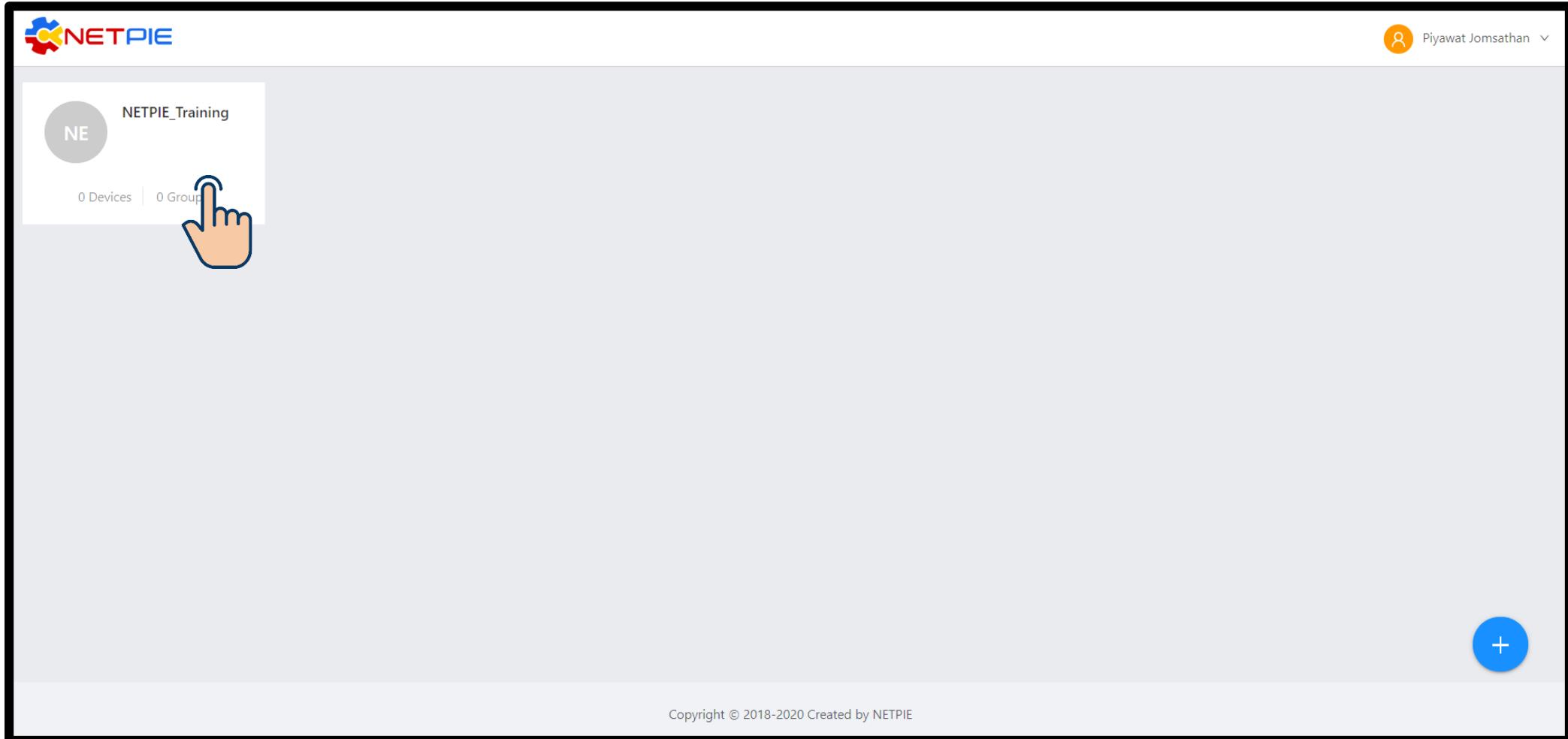
การสร้าง Project

กรอกข้อมูลสำหรับการสร้าง Project
โดย filed ที่มีเครื่องหมาย * กำกับจำเป็นต้องกรอก



การสร้าง Project

คลังจากสร้าง Project แล้วให้คลิกเข้าไปใน Project



การสร้าง Project

The screenshot shows the NETPIE 2020 Portal's Overview page. On the left is a sidebar with options: Overview (selected), Device List, Device Groups, Freeboard, Event Hooks, and Setting. The main area displays the project 'NETPIE_Training / overview'. It includes a 'Descriptions' section with a creation date of 2020-05-08, a 'Devices' section showing 0 online and 0 offline devices, and a 'Groups' section showing 0 groups. A user profile for 'Piyawat Jomsathan' is at the top right.

ส่วนของหน้า Overview ที่แสดงข้อมูลทั้งหมดของ Project ได้แก่

- 1. Descriptions**
- 2. Devices [Device Online/Offline]**
- 3. Groups**

Copyright © 2018-2020 Created by NETPIE

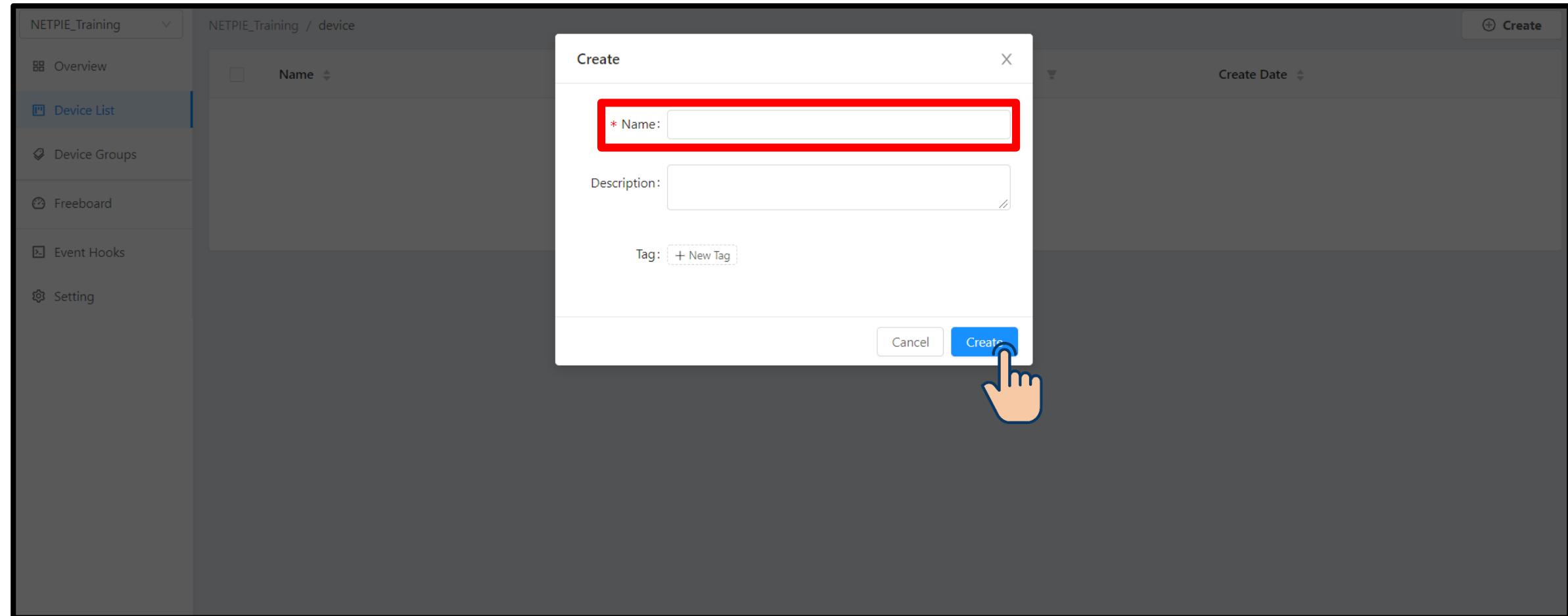
การสร้าง Device

ในการสร้าง Device จะต้องเข้ามาอยังหน้า Device List จะปราศจากหน้าต่างดังรูป
หลังจากนั้นให้คลิก Create เพื่อทำการสร้าง Device

Copyright © 2018-2020 Created by NETPIE

การสร้าง Device

กรอกข้อมูลสำหรับการสร้าง Device
โดย filed ที่มีเครื่องหมาย * กำกับจำเป็นต้องกรอก



Module 2 : NETPIE 2020 Portal

การสร้าง Device

NETPIE_Training / device

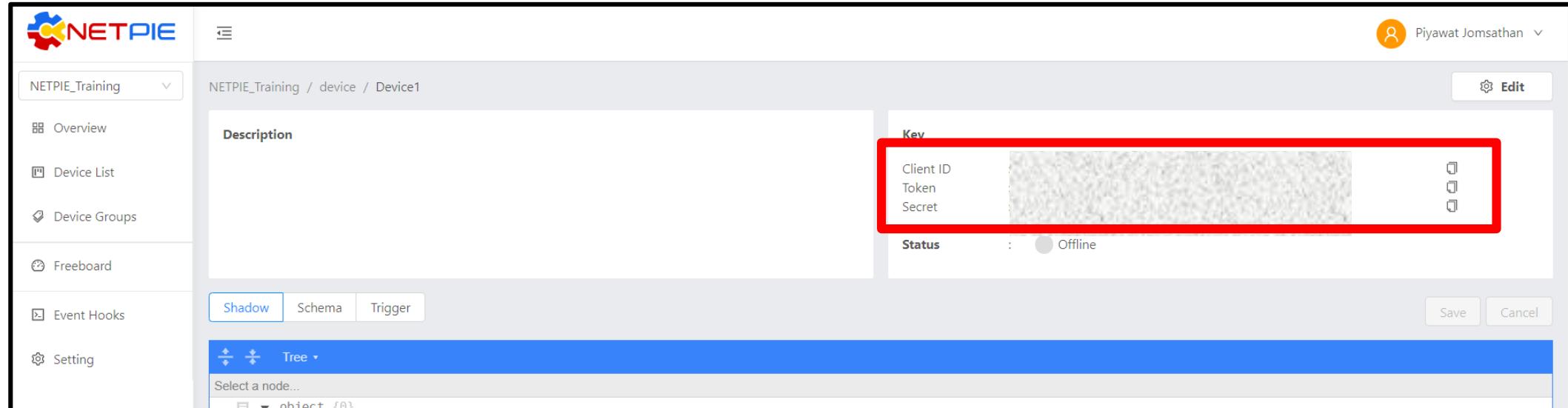
Overview Device List Device Groups Freeboard Event Hooks Setting

Name	Tags	Group	Create Date
Device1	-	-	2020-05-08 14:34

1-1 of 1 items < 1 > 10 / page

คลิ้งจากสร้าง Device และให้คลิกที่ແນ Device เพื่อดูข้อมูลของ Device

การสร้าง Device

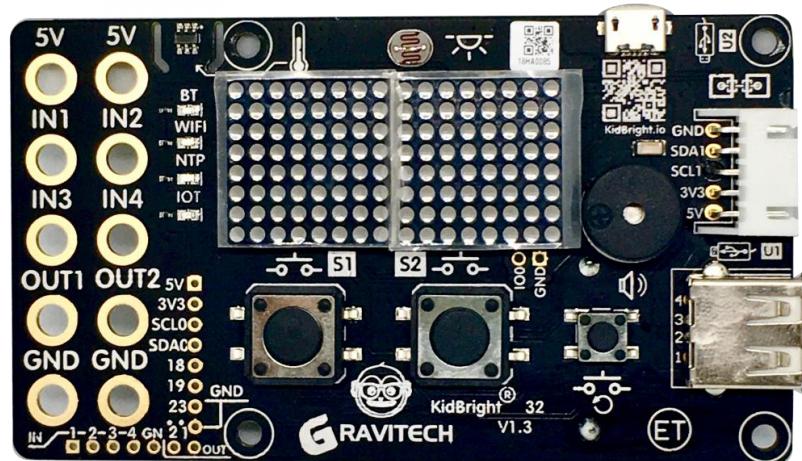


หลังจากเข้ามา�ัง Device จะพบรายละเอียดต่างๆของ Device นั้นคือ

1. Client ID
2. Token
3. Secret

ซึ่งทั้ง 3 Parameter นี้มีส่วนสำคัญในการนำอุปกรณ์ต่อ กับ NETPIE2020 ด้วย Protocol ต่างๆ

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright



KidBright

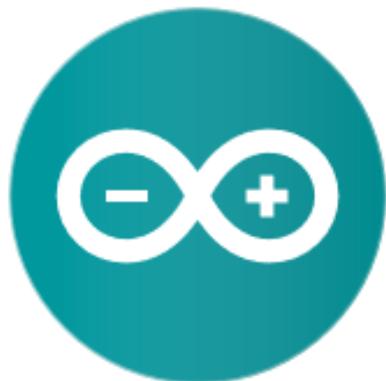
เชื่อมต่อ



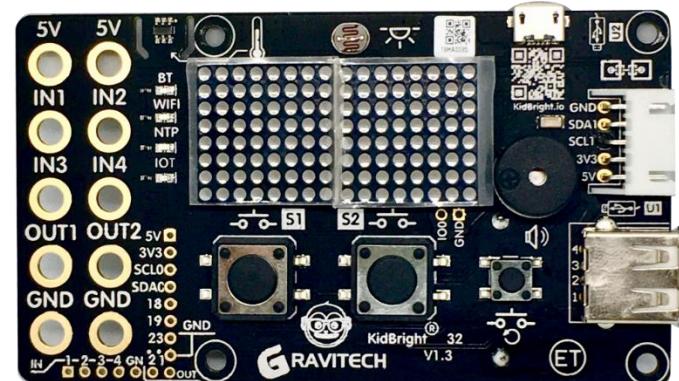
NETPIE2020

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

ในการจะให้ KidBright สามารถเชื่อมต่อกับ NETPIE2020 ได้นั้น
จำเป็นต้องเขียนโปรแกรมบน Arduino IDE เพื่อพัฒนาให้
KidBright เชื่อมต่ออินเทอร์เน็ตแล้วเชื่อมต่อกับ NETPIE2020



Arduino IDE



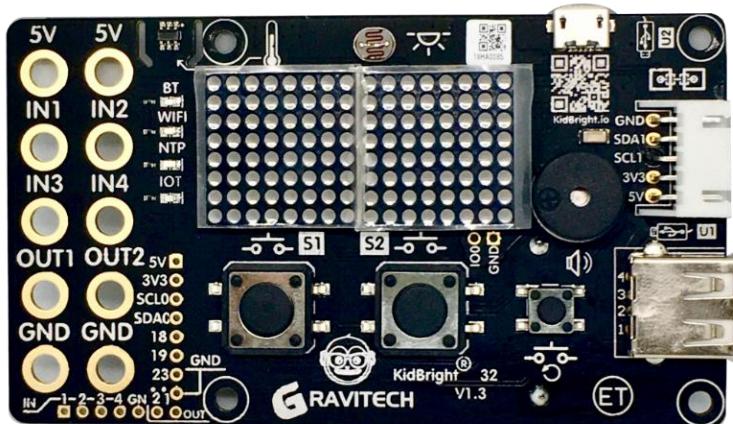
KidBright

เชื่อมต่อ^{วิธีการ}
อินเทอร์เน็ต
โดยอาศัย
WiFi



NETPIE2020

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright



KidBright

ในการจะใช้ KidBright สามารถเชื่อมต่อกับ NETPIE2020 ได้นั้นจำเป็นต้องใช้ Library ดื้อ

1. WiFi

ใช้สำหรับให้ KidBright เชื่อมต่อกับอินเทอร์เน็ตผ่านเครือข่าย WiFi

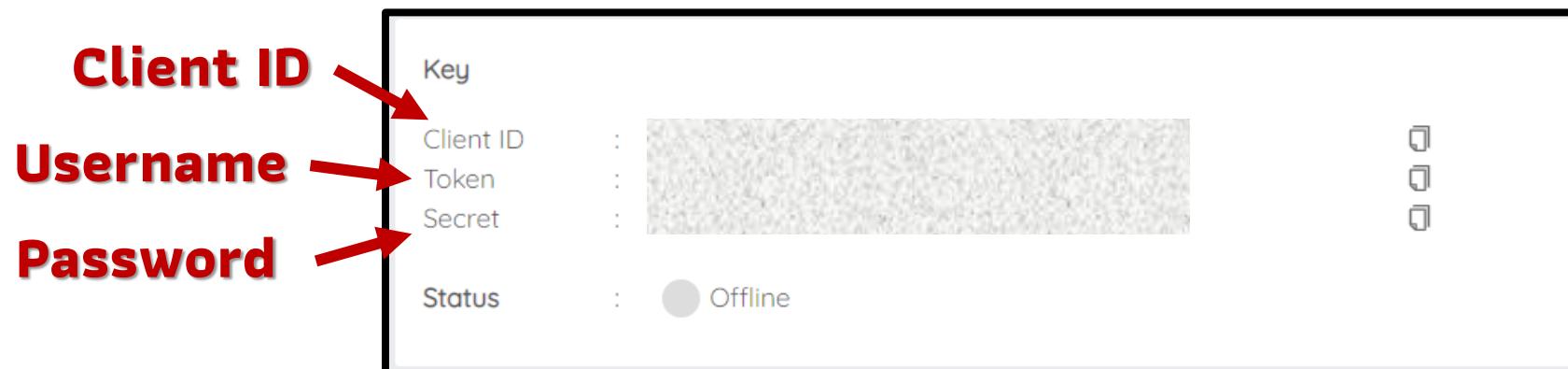
2. PubSubClient

ใช้สำหรับให้ KidBright เชื่อมต่อและสื่อสารบน NETPIE 2020 [MQTT Protocol]

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

การเชื่อมต่อ NETPIE2020 จะต้องใช้ 4 Parameters คือ

1. Host คือ broker.netpie.io
2. Client ID คือ Client ID ของ Device ที่สร้างขึ้นใน NETPIE2020
3. Username คือ Token ของ Device ที่สร้างขึ้นใน NETPIE2020
4. Password คือ Secret ของ Device ที่สร้างขึ้นใน NETPIE2020
[ใช้สำหรับกรณีที่ต้องการตรวจสอบที่เพิ่มมากขึ้น]



Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

เปิดไฟล์ Workshop1.ino ขึ้นมา

Coding ใน Workshop1.ino แบ่งออกเป็น 3 ส่วน

1 ส่วนที่ 1 การเรียกใช้ Library และ ประกาศตัวแปร

เป็นการเรียกใช้ Library ต่างๆ

เป็นการประกาศตัวแปรสำหรับเชื่อมต่อ WiFi

เป็นการประกาศตัวแปรสำหรับเชื่อมต่อ NETPIE2020

การกำหนดและเรียกใช้ชุดคำสั่ง
สำหรับเชื่อมต่อ NETPIE2020

```
#include <WiFi.h>
#include <PubSubClient.h>
```

```
const char* ssid = "Your WiFi SSID";
const char* password = "Your Password";
```

```
const char* mqtt_server = "broker.netpie.io";
const int mqtt_port = 1883;
const char* mqtt_Client = "Your Client-ID";
const char* mqtt_username = "Your Token";
const char* mqtt_password = "Your Secret";
```

```
WiFiClient espClient;
PubSubClient client[espClient];
```

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

2

ส่วนที่ 2 ส่วนของฟังก์ชันต่างๆ

ฟังก์ชันการเชื่อมต่อ NETPIE2020

```
void reconnect() {  
    while (!client.connected()) {  
        Serial.print("Attempting NETPIE2020 connection...");  
        if (client.connect(mqtt_Client, mqtt_username, mqtt_password)) {  
            Serial.println("NETPIE2020 connected");  
        }  
        else {  
            Serial.print("failed, rc=");  
            Serial.print(client.state());  
            Serial.println("try again in 5 seconds");  
            delay(5000);  
        }  
    }  
}
```

เข้าสู่การเชื่อมต่อ NETPIE2020

- หากเชื่อมต่อสำเร็จจะแสดงผลว่า “connected”
 - หากเชื่อมต่อไม่สำเร็จจะแสดงผลว่า “failed ...”
- และจะทำการเชื่อมต่อใหม่อัตโนมัติ

Module 2 : NETPIE 2020 Portal

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

2 | ส่วนที่ 2 ส่วนของพิงก์ชันต่างๆ

```
void setup() {  
    Serial.begin(115200);  
    Serial.println("Starting...");  
    if (WiFi.begin(ssid, password)) {  
        while (WiFi.status() != WL_CONNECTED) {  
            delay(1000);  
            Serial.print(".");  
        }  
    }  
    Serial.println("WiFi connected");  
    Serial.println("IP address: ");  
    Serial.println(WiFi.localIP());  
    client.setServer(mqtt_server, mqtt_port);  
}
```

ຝຶກຂັ້ນການຕັ້ງຄ່າຕ່າງໆ

ในฟังก์ชันนี้จะทำการเชื่อมต่อกับ WiFi และ NETPIE2020 ตามค่าต่างๆที่ได้ตั้งค่าไว้

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

3

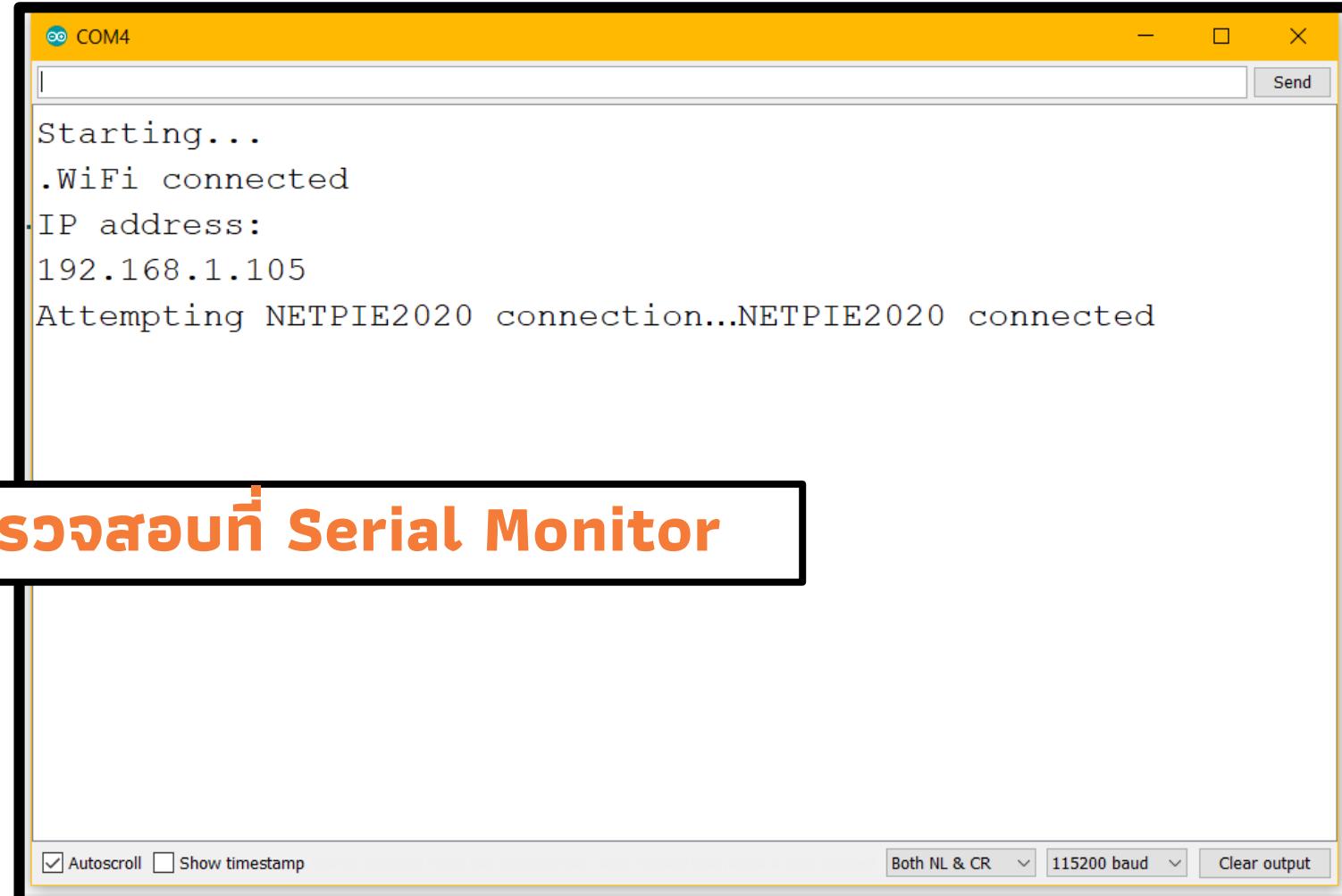
ส่วนที่ 3 ส่วนของฟังก์ชันหลัก

ฟังก์ชันการทำงานหลัก

```
void loop() {  
  
    if (!client.connected()) {  
        reconnect();  
    }  
    client.loop();  
  
}
```

เป็นชุดคำสั่งคงstanะของการเชื่อมต่อและ
การทำงานต่างๆของ NETPIE2020

Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright



Workshop 1 : การเชื่อมต่อ NETPIE2020 ด้วย KidBright

NETPIE_Training / device / Device1

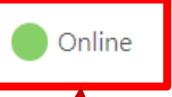
Description

Key

Client ID : 

Token :

Secret :

Status :  Online

Shadow Schema Trigger

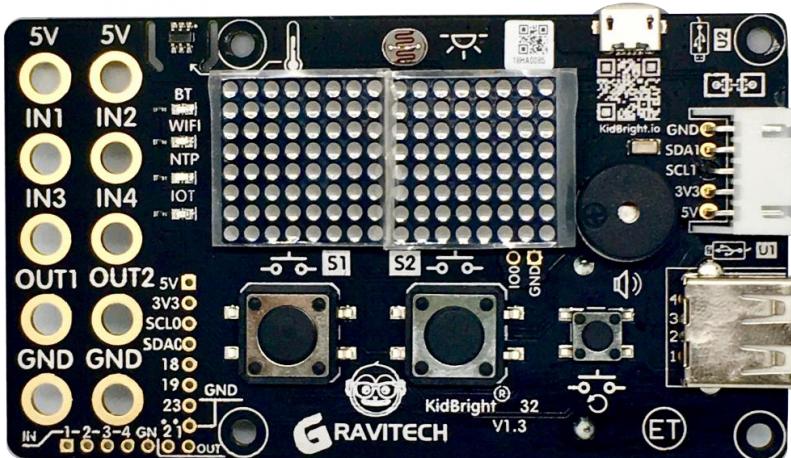
Select a node:

- object {0}
- (empty object)

ตรวจสอบสถานะ การเชื่อมต่อบน NETPIE2020



Workshop 2 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Publish]



KidBright

“Hello NETPIE2020”

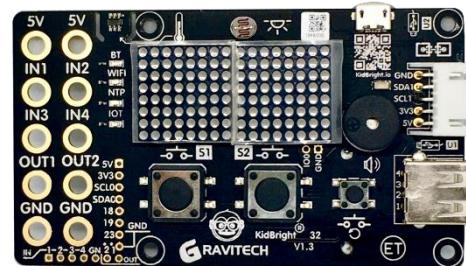


NETPIE2020

Workshop 2 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Publish]

ทำความเข้าใจรูปแบบการสื่อสารบน NETPIE2020

การจะส่งข้อความไปที่ NETPIE2020 จะต้อง
อ้างอิงหัวข้อเรื่องเสมอ ซึ่งเรียกว่า Topic



Topic : @msg/test
“Hello NETPIE2020”



KidBright ทำการส่งข้อความไปที่ NETPIE2020
ด้วยประโยคว่า “Hello NETPIE2020”
ซึ่งเรียกว่า การ Publish

NETPIE2020 ทำหน้าที่
เปรียบเสมือนห้องแชทห้องหนึ่ง
ซึ่งเรียกว่า Broker

Workshop 2 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Publish]

เปิดไฟล์ Workshop2.ino ขึ้นมา

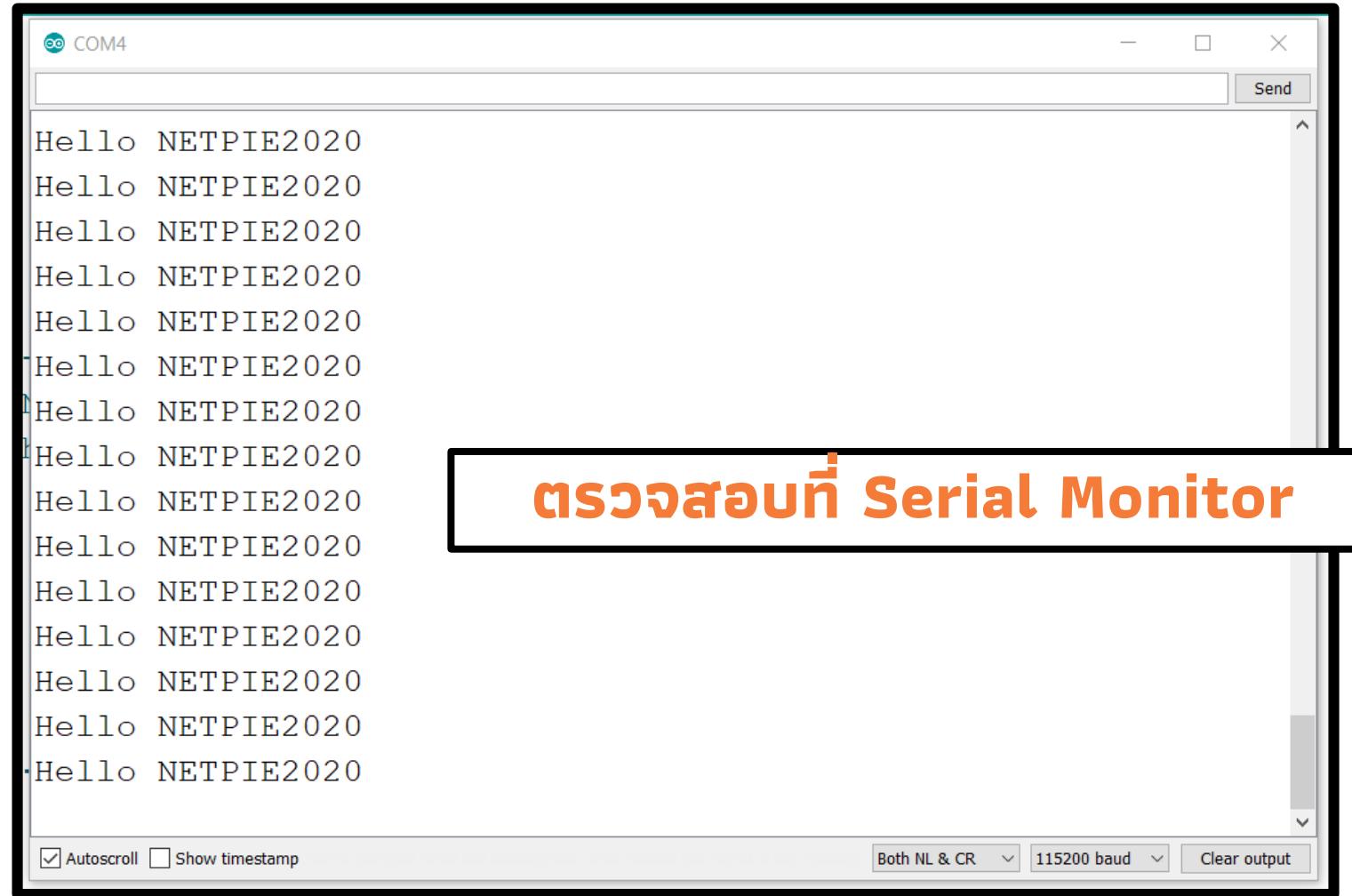
Coding ใน Workshop2.ino เริ่มต้นจะเป็นการเชื่อมต่อ WiFi และ NETPIE2020 ซึ่งมีความคล้ายกับ Workshop1.ino แต่จะมีเพียงส่วนของ void loop ที่เพิ่มเติมขึ้นมา

client.publish เป็นคำสั่งในการ Publish ข้อความไปยัง NETPIE2020 โดยมีรูปแบบดังนี้
client.publish["Topic", "Massage"];

```
void loop() {
  if (!client.connected()) {
    reconnect();
  }
  client.loop();

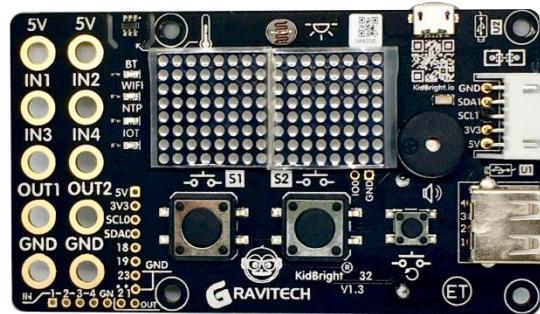
  client.publish["@msg/test", "Hello NETPIE2020"];
  Serial.println("Hello NETPIE2020");
  delay(2000);
}
```

Workshop 2 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Publish]



Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

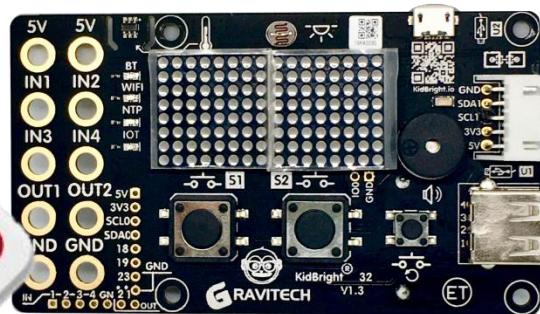


KidBright 1

"Hello NETPIE2020"



NETPIE2020



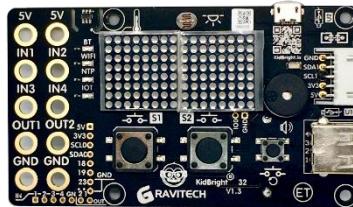
KidBright 2

"Hello NETPIE2020"

ใช้ MQTTBox ได้

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

ทำความเข้าใจรูปแบบการสื่อสารบน NETPIE2020



KidBright 1 ทำการ **Publish**
ข้อความไปยัง Topic @msg/test

*Publish Topic : @msg/test
“Hello NETPIE2020”*



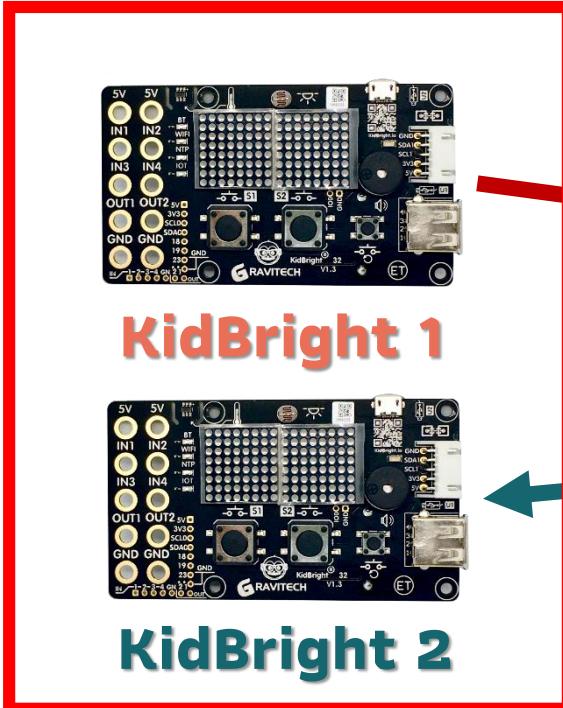
*Subscribe Topic : @msg/test
“Hello NETPIE2020”*



KidBright 2 ต้องการข้อความที่ KidBright 1
ส่งมาต้องทำการเลือกรับข้อความตาม Topic
ซึ่งเรียกว่า Subscribe

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

ทำความเข้าใจรูปแบบการสื่อสารบน NETPIE2020



*Publish Topic : @msg/test
“Hello NETPIE2020”*

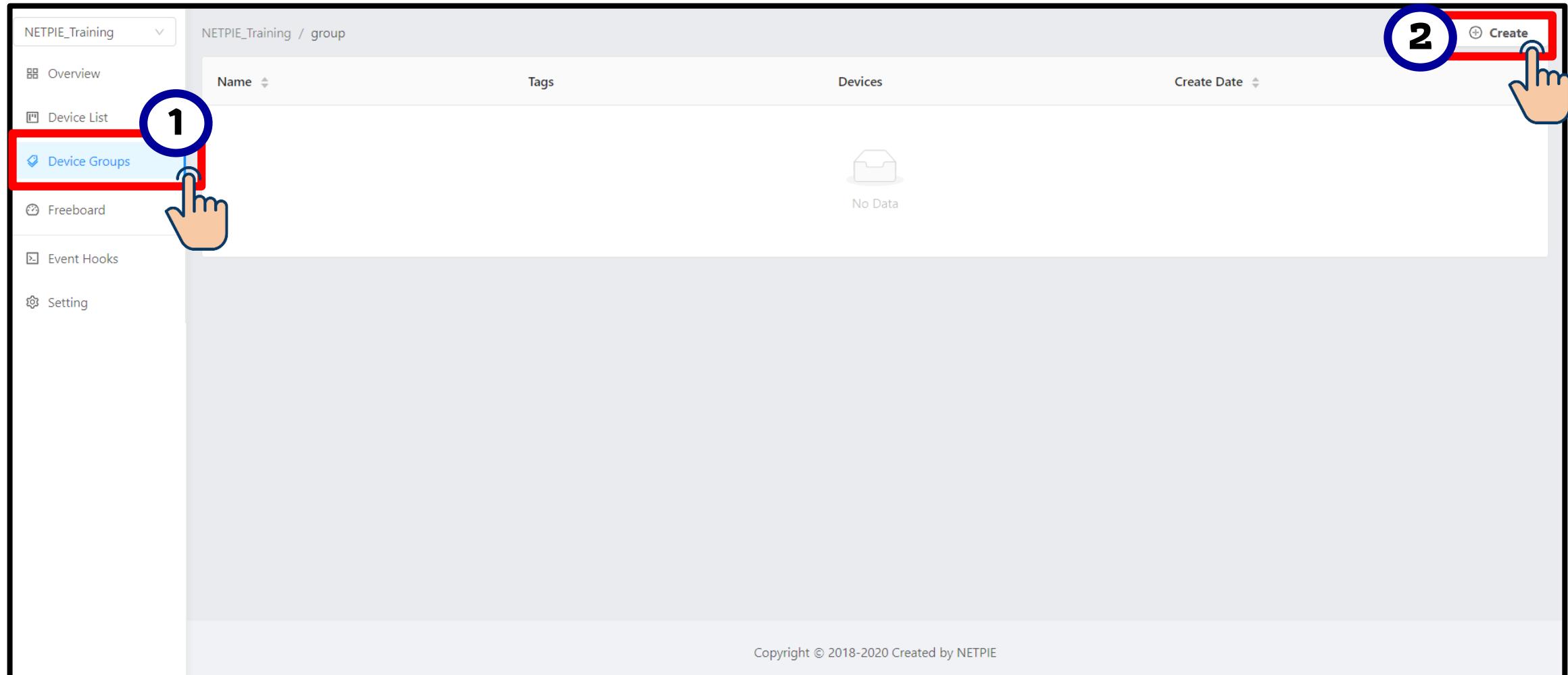
*Subscribe Topic : @msg/test
“Hello NETPIE2020”*



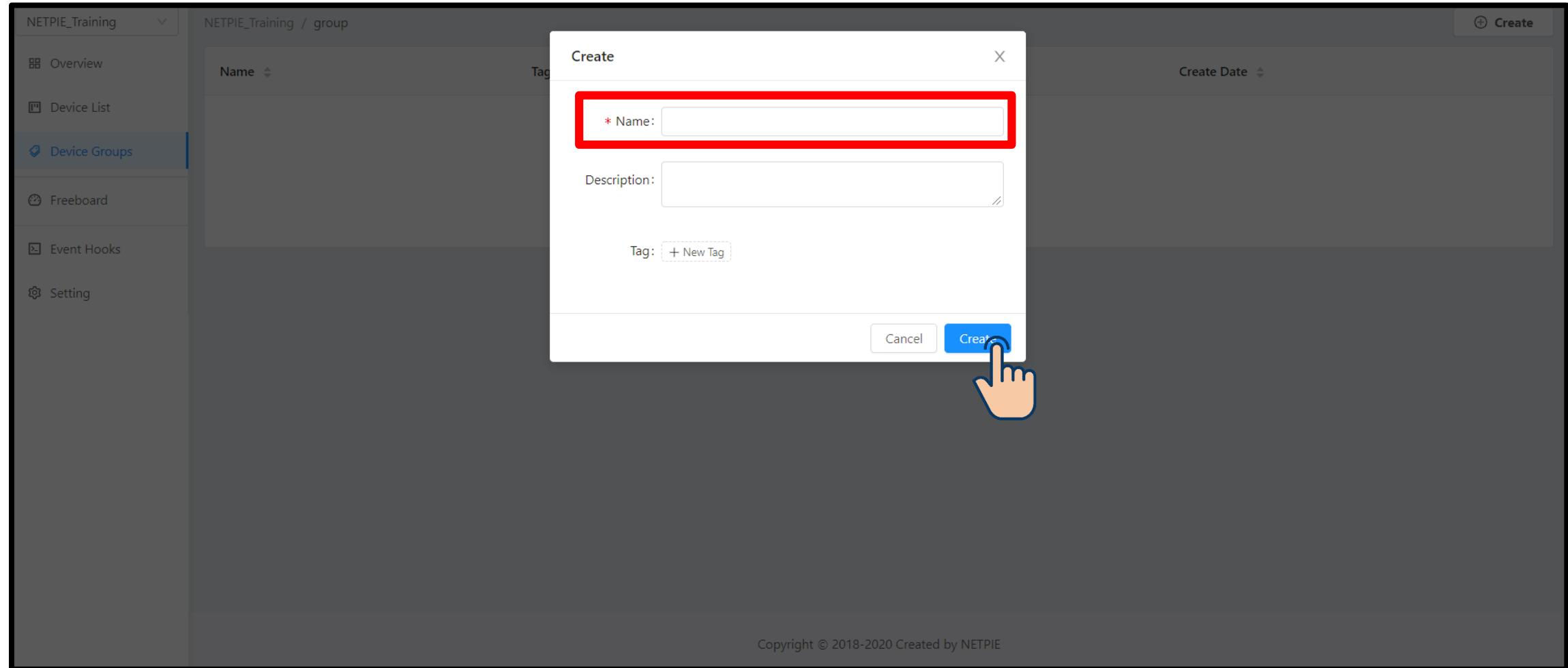
Device ทั้ง 2 จะต้องอยู่ใน Group เดียวกัน

Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

The screenshot shows the NETPIE Platform interface. On the left, there's a sidebar with various options: Overview, Device List, Device Groups (which is selected and highlighted in blue), Freeboard, Event Hooks, and Setting. The main area is titled "NETPIE_Training / group" and contains a table with four columns: Name, Tags, Devices, and Create Date. A single row is visible, representing a group named "Group1". The "Devices" column shows "0 Online" and "0 Offline". The "Create Date" column shows "2020-05-08 23:35". A red box highlights the entire row for "Group1", and a red arrow points from this highlighted row to a large black box containing the following text in Thai:

**Group ที่ได้สร้างขึ้น
แต่ใน Group นั้นยังไม่มี Device จึงต้องนำเข้า Device ก่อน**

Copyright © 2018-2020 Created by NETPIE

Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

The screenshot shows the NETPIE 2020 Portal's Device List interface. On the left, there's a sidebar with options: Overview, Device List (which is selected and highlighted in blue), Device Groups, Freeboard, Event Hooks, and Setting. The main area displays a table with two rows:

	Name	Tags	Group	Create Date
<input type="checkbox"/>	Device1	-	-	2020-05-08 14:34
<input type="checkbox"/>	Device2	-	-	2020-05-08 23:24

At the bottom right of the table, it says "1-2 of 2 items" with navigation buttons (back, forward, page 10). Below the table, there's a message: "สร้าง Device ขึ้นมาใหม่ เพื่อใช้งาน MQTTBox".

สร้าง Device ขึ้นมาใหม่ เพื่อใช้งาน MQTTBox

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

The screenshot shows the NETPIE 2020 Portal interface. On the left is a sidebar with icons for Overview, Device List (which is selected and highlighted in blue), Device Groups, Freeboard, Event Hooks, and Setting. The main area is titled "NETPIE_Training / device" and lists two devices: "Device1" and "Device2". Each device row has a checkbox in the first column, which is checked for both devices. A hand cursor is shown clicking on the checkbox for "Device1". The columns are labeled "Name", "Tags", and "Group".

Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

NETPIE_Training / device

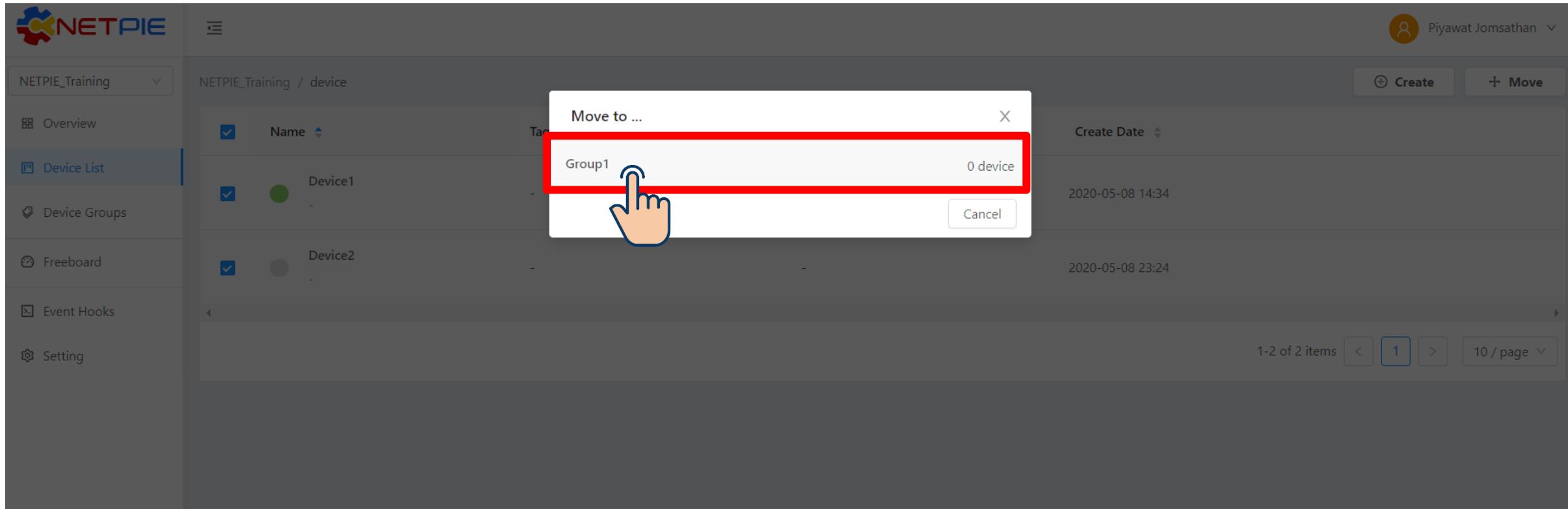
Name	Tags	Group	Create Date
Device1	-	-	2020-05-08 14:34
Device2	-	-	2020-05-08 23:24

Piyawat Jomsathan

Create Move

1-2 of 2 items < 1 > 10 / page

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

NETPIE_Training / device

Name	Tags	Group	Create Date
Device1	-	Group1	2020-05-08 14:34
Device2	-	Group1	2020-05-08 23:24

NETPIE_Training

Overview

Device List

Device Groups

Freeboard

Event Hooks

Setting

Create

Move

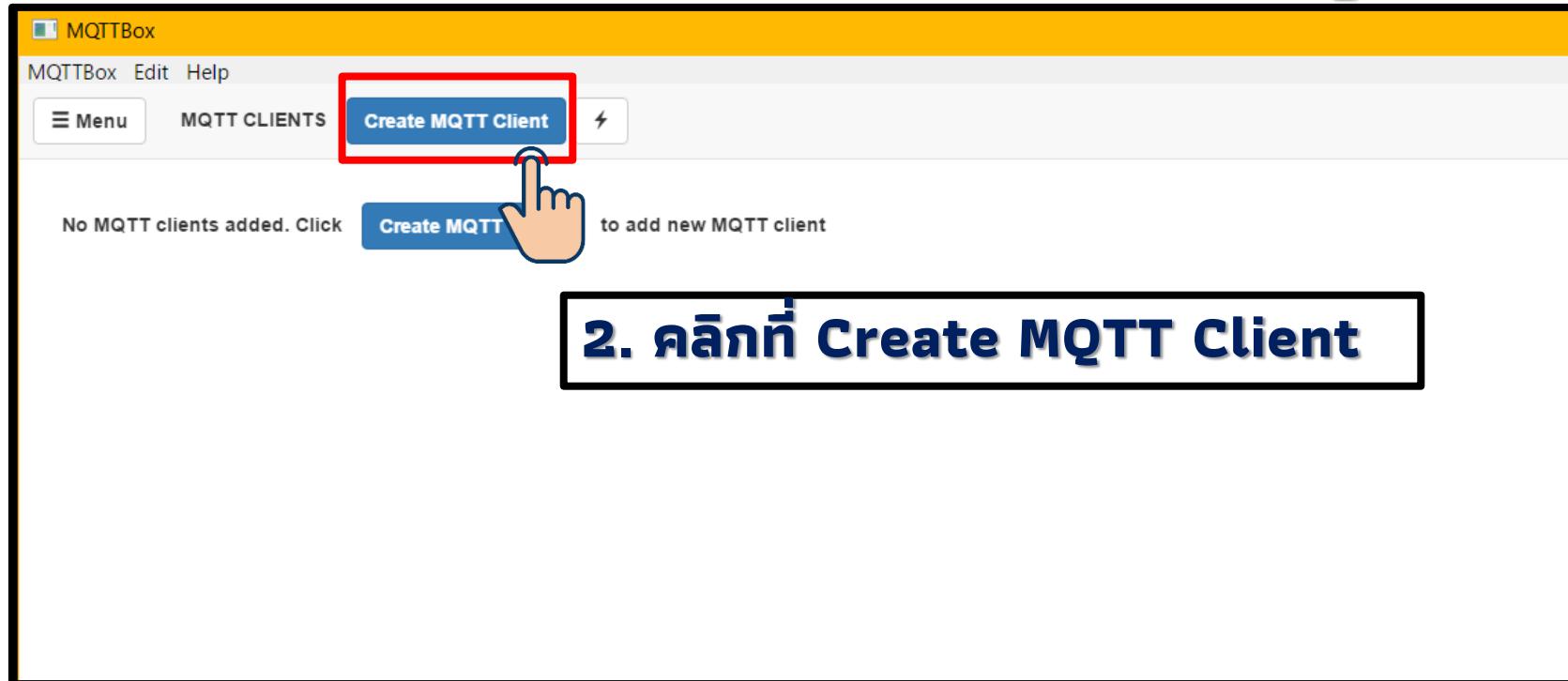
1-2 of 2 items < 1 > 10 / page

แสดง Group ที่ອ้าคัยอยู่

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

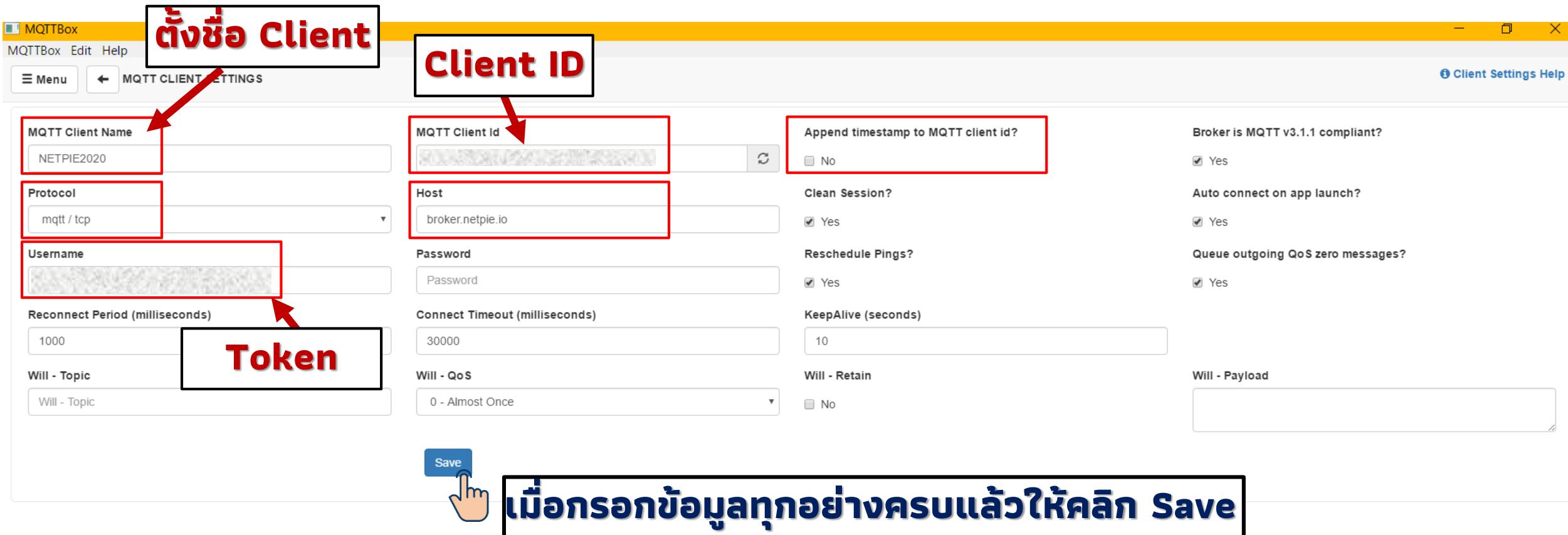
ໃນ Workshop3 នីមួយៗ នឹងប្រើប្រាស់ MQTTBox នៃក្រុម KidBright

1. ចាប់ផ្តើម MQTTBox ឱ្យមាតា

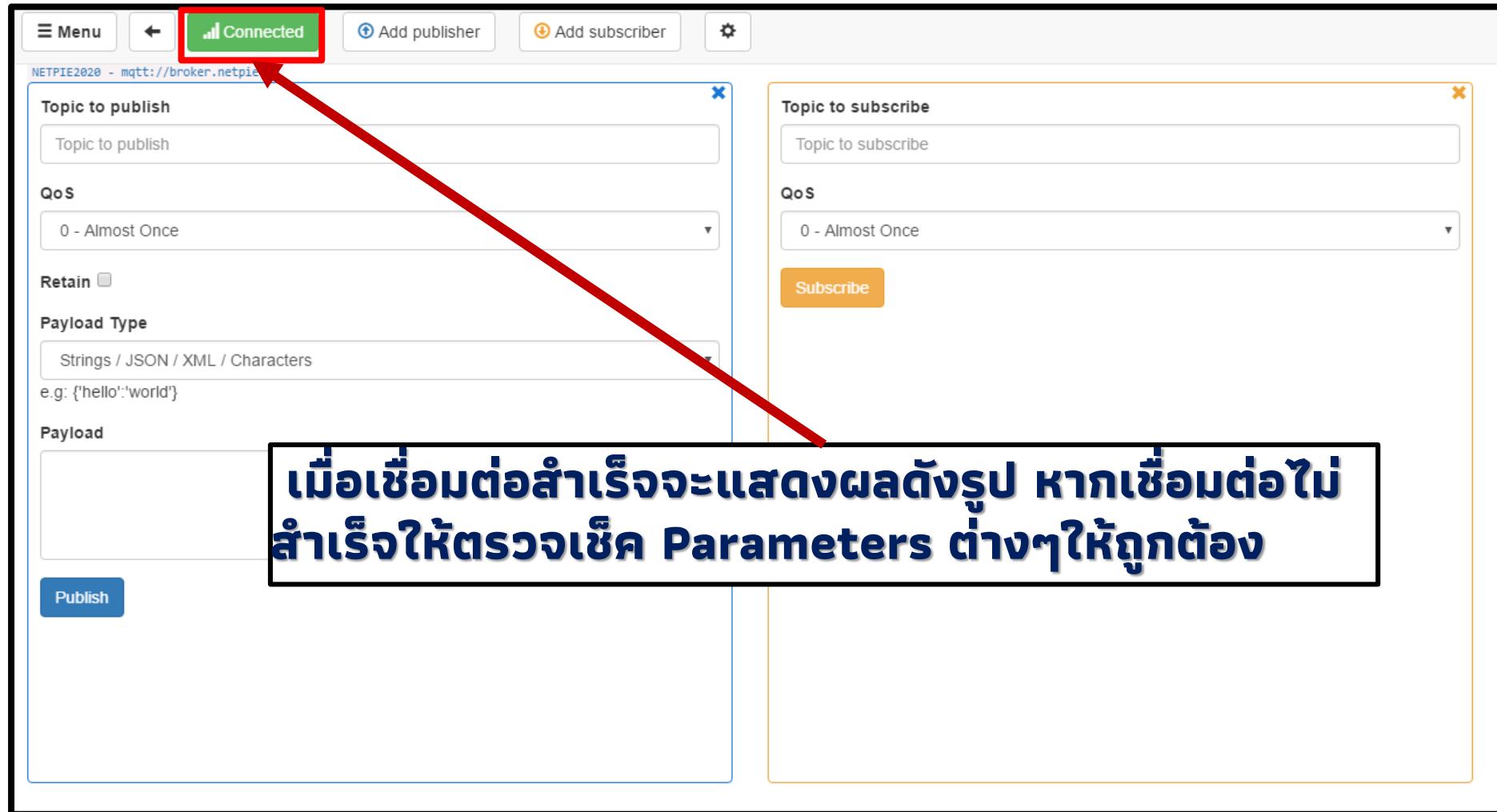


2. គ្រឿងទៅ Create MQTT Client

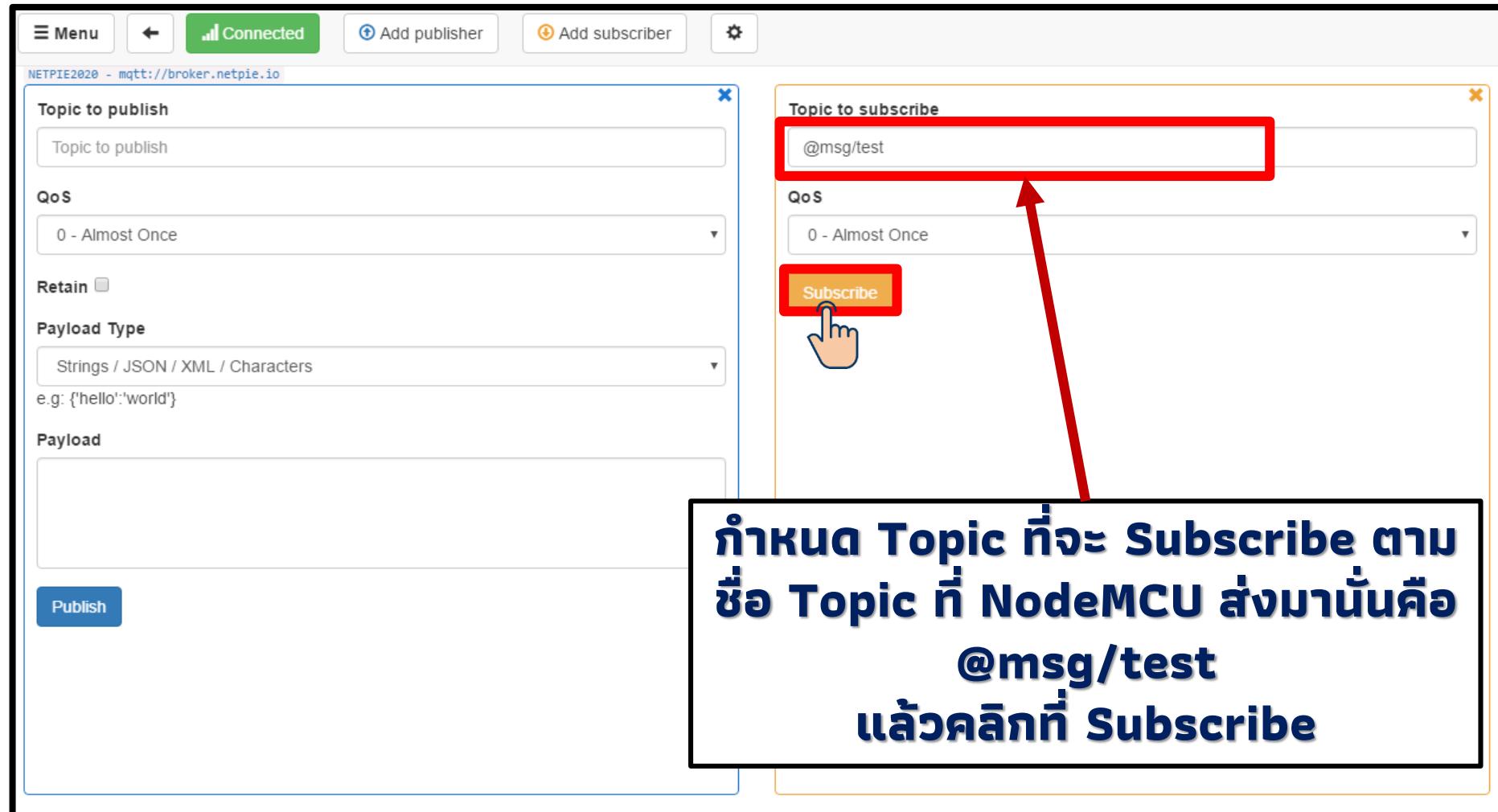
Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

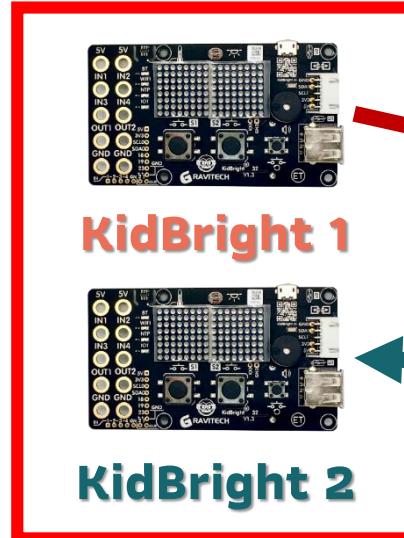
The screenshot shows the NETPIE 2020 Portal interface. On the left, a modal window titled "Topic to publish" is open, showing fields for "Topic to publish" (set to "Topic to publish"), "QoS" (set to "0 - Almost Once"), "Retain" (unchecked), "Payload Type" (set to "Strings / JSON / XML / Characters"), and a "Payload" area containing the JSON string "e.g. {'hello':'world'}". A "Publish" button is at the bottom. On the right, a "Subscriptions" section shows two messages received on the topic "@msg/test": "Hello NETPIE2020" followed by detailed message metadata. A red arrow points from the text in the bottom box to the second message in the list.

ที่ MQTTBox จะได้รับข้อความที่ KidBright ส่งมานั่นคือ “Hello NETPIE2020”

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

ทดสอบถ้ามีอุปกรณ์ไม่ได้อยู่ในกลุ่ม Subscribe

Group1

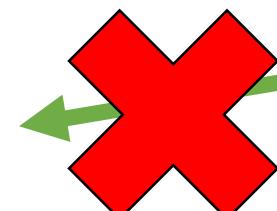


*Publish Topic : @msg/test
“Hello NETPIE2020”*

*Subscribe Topic : @msg/test
“Hello NETPIE2020”*



KidBright 3



*Subscribe Topic : @msg/test
“Hello NETPIE2020”*

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

NETPIE_Training / device

Name	Tags	Group	Create Date
Device3	-	-	2020-05-09 08:32
Device2	-	Group1	2020-05-08 23:24
Device1	-	Group1	2020-05-08 14:34

สร้าง Device ขึ้นมาใหม่ เพื่อใช้สำหรับ MQTTBox

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

ເປີດ MQTTBox ຂຶ້ນມາອັກຫຼາຕ່າງ



Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

MQTTBox

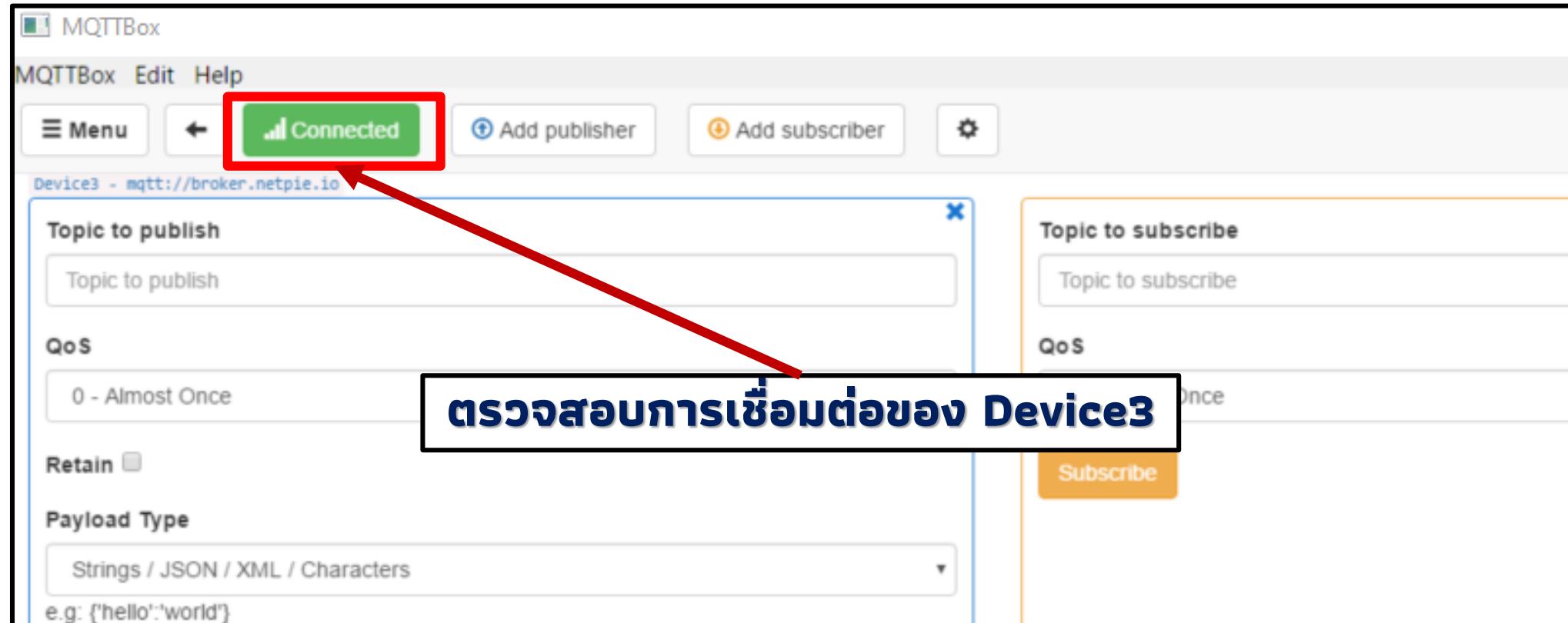
MQTTBox Edit Help

☰ Menu ← MQTT CLIENT SETTINGS ⓘ Client S

MQTT Client Name Device3	MQTT Client Id Device3	Append timestamp to MQTT client id? <input type="checkbox"/> No	Broker is MQTT v3.1.1 compliant? <input checked="" type="checkbox"/> Yes
Protocol mqtt / tcp	Host broker.netpie.io	Clean Session? <input checked="" type="checkbox"/> Yes	Auto connect on app launch? <input checked="" type="checkbox"/> Yes
Username Device3	Password Device3	Reschedule Pings? <input checked="" type="checkbox"/> Yes	Queue outgoing QoS zero messages? <input checked="" type="checkbox"/> Yes
Reconnect Period (milliseconds) 1000	Connect Timeout (milliseconds) 30000	KeepAlive (seconds) 10	
Will - Topic Will - Topic	Will - QoS 0 - Almost Once	Will - Retain <input type="checkbox"/> No	Will - Payload

Save

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]



Module 2 : NETPIE 2020 Portal

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

The image shows two screenshots of the NETPIE 2020 Portal interface.
Device1 (left): Shows a 'Topic to publish' field containing '@msg/test'. Below it are QoS settings (0 - Almost Once), Retain checkbox, and Payload Type dropdown (Strings / JSON / XML / Characters). A red box highlights the 'Topic to subscribe' field in a modal window, which also contains '@msg/test' and a 'Subscribe' button. A red arrow points from the 'Topic to subscribe' field in the modal to the same field in Device3's modal.
Device3 (right): Shows a similar setup with a 'Topic to publish' field and a 'Topic to subscribe' field containing '@msg/test'. Below it is a 'QoS' dropdown set to '0 - Almost Once' and a 'Subscribe' button. A red box highlights the 'Topic to subscribe' field in Device3's modal.
A large callout box at the bottom left contains the following text in Thai:
**กำหนด Topic ที่จะ Subscribe ตาม
ชื่อ Topic ที่ KidBright ส่งมาแล้วคือ
@msg/test
แล้วคลิกที่ Subscribe**

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

The screenshot shows the NETPIE 2020 Portal interface. On the left, a configuration panel for Device1 is shown with the following settings:

- Topic to publish: @msg/test
- QoS: 0 - Almost Once
- Retain:
- Payload Type: Strings / JSON / XML / Characters
- e.g.: {"hello": "world"}
- Payload: Hello Smart Factory IoT Challenge 2020

A blue "Publish" button is at the bottom. To the right, a list of received messages from the broker is displayed:

- Message 1: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 1, length: 27, Raw payload: 721 011081081113278698480736950485048)
- Message 2: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 2, length: 27, Raw payload: 721 011081081113278698480736950485048)
- Message 3: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 3, length: 27, Raw payload: 721 011081081113278698480736950485048)
- Message 4: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 4, length: 27, Raw payload: 721 011081081113278698480736950485048)

ที่ Device1 จะได้รับข้อความที่ KidBright ส่งมานั่นคือ “Hello NETPIE2020”

The screenshot shows the NETPIE 2020 Portal interface. On the left, a configuration panel for Device3 is shown with the following settings:

- Topic to publish: Topic to p
- QoS: 0 - Almost Once
- Retain:
- Payload Type: Strings / JSON / XML / Characters
- e.g.: {"hello": "world"}
- Payload: (empty text area)

A blue "Publish" button is at the bottom. To the right, a list of received messages from the broker is displayed:

- Message 1: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 1, length: 27, Raw payload: 721 011081081113278698480736950485048)
- Message 2: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 2, length: 27, Raw payload: 721 011081081113278698480736950485048)
- Message 3: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 3, length: 27, Raw payload: 721 011081081113278698480736950485048)
- Message 4: Hello NETPIE2020 (topic: @msg/test, qos: 0, retain: false, cmd: publish, dup: false, topic: @msg/test, messageId: 4, length: 27, Raw payload: 721 011081081113278698480736950485048)

แต่ที่ Device3 จะได้ไม่ได้รับข้อความ
เนื่องจากอยู่นอกกลุ่ม

Workshop 3 : การสื่อสารบน NETPIE2020 ด้วย KidBright [Subscribe]

การสื่อสารรูปแบบนี้บน NETPIE2020 จะถูกเรียกว่าการสื่อสารบน **MQTT Protocol**

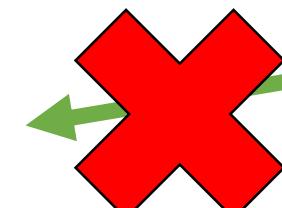
Group1



*Publish Topic : @msg/test
"Hello NETPIE2020"*



*Subscribe Topic : @msg/test
"Hello NETPIE2020"*

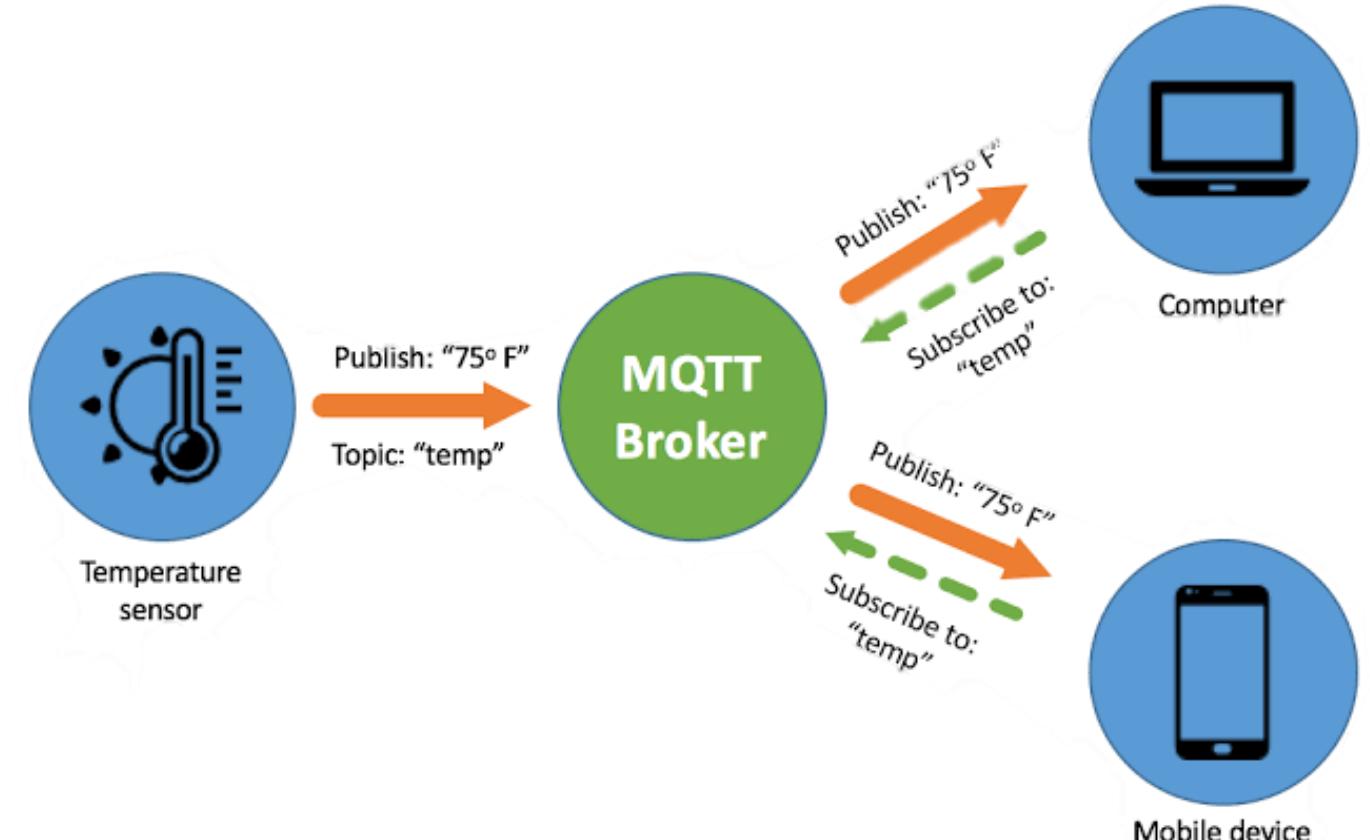


*Subscribe Topic : @msg/test
"Hello NETPIE2020"*

การสื่อสารบน NETPIE2020 [MQTT]

MQTT คืออะไร ??

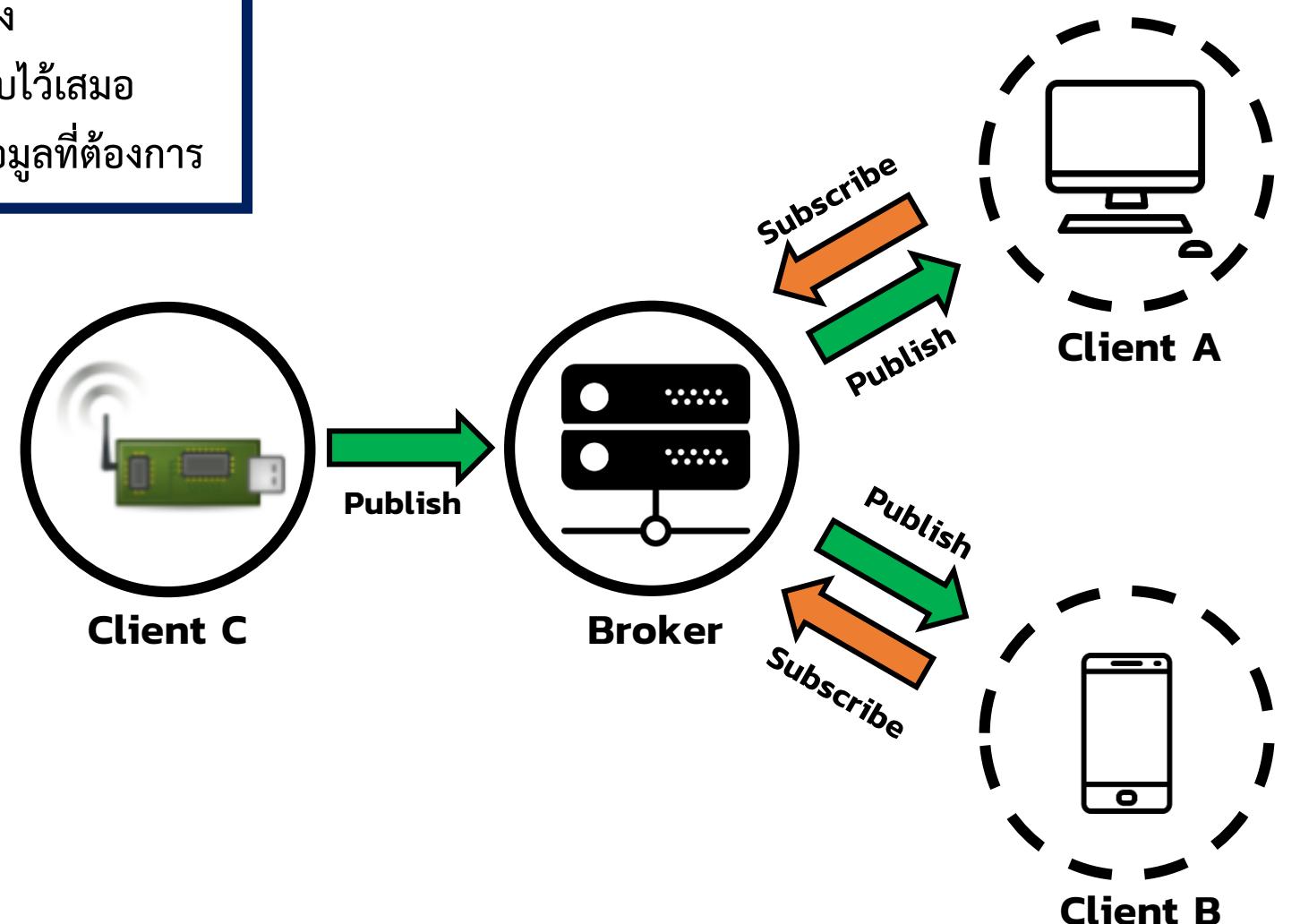
- MQTT เป็น Protocol ที่ออกแบบมาเพื่อการเชื่อมต่อแบบ M2M หรือ อุปกรณ์กับอุปกรณ์ ซึ่งเป็นการสนับสนุน IoT
- ใช้หลักการการรับส่งข้อมูลแบบ Publish/Subscribe คล้ายกับหลักการที่ใช้ใน Web Service ที่ต้องใช้ Web Server เป็นตัวกลางระหว่างคอมพิวเตอร์ของผู้ใช้งาน
- แต่ MQTT ใช้ตัวกลางที่เรียกว่า Broker ทำหน้าที่จัดการลำดับการรับ – ส่ง ข้อมูลระหว่างอุปกรณ์และทั้งที่เป็น Publish/Subscribe



Module 2 : NETPIE 2020 Portal

การส่งข้อมูลบน MQTT จะมี Topic เป็นตัวอ้างอิงหลัก ซึ่ง

- ข้อมูลที่จะ Publish ไปยัง Broker ต้องมี Topic กำกับไว้เสมอ
- ส่วนของ Subscribe ก็ต้องอ้างอิง Topic เพื่อเรียกข้อมูลที่ต้องการ



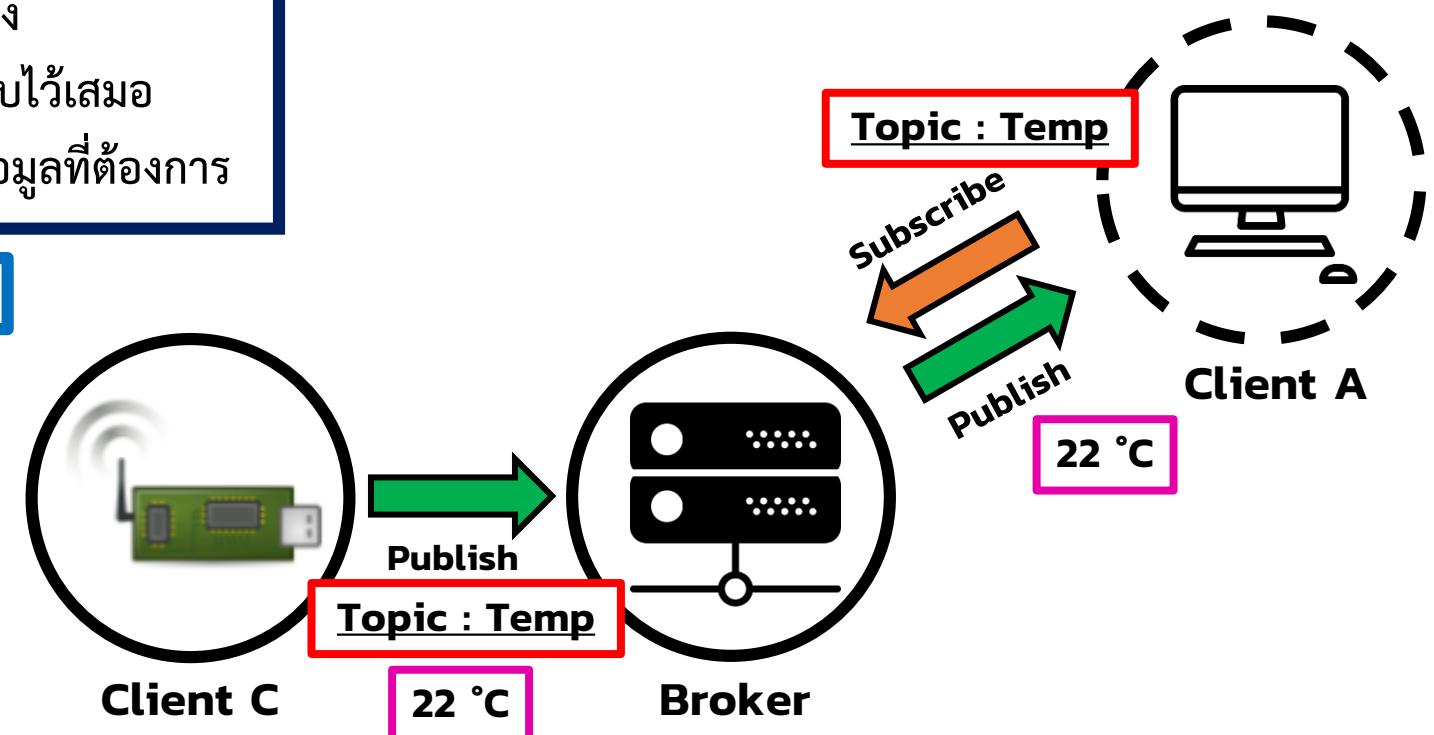
Module 2 : NETPIE 2020 Portal

การส่งข้อมูลบน MQTT จะมี Topic เป็นตัวอ้างอิงหลัก ซึ่ง

- ข้อมูลที่จะ Publish ไปยัง Broker ต้องมี Topic กำกับไว้เสมอ
- ส่วนของ Subscribe ก็ต้องอ้างอิง Topic เพื่อเรียกข้อมูลที่ต้องการ

ยกตัวอย่างการส่งข้อมูลแบบ MQTT

- Client C นั้น Publish ข้อมูลบน Topic ที่มีชื่อว่า Temp และมี Client A ที่ Subscribe Topic Temp ไว้อยู่
- โดยข้อมูลจาก Client C นั้นคือ อุณหภูมิ ที่มีค่าเท่ากับ 22 °C ดังนั้น Client A จะได้รับข้อมูลด้วย



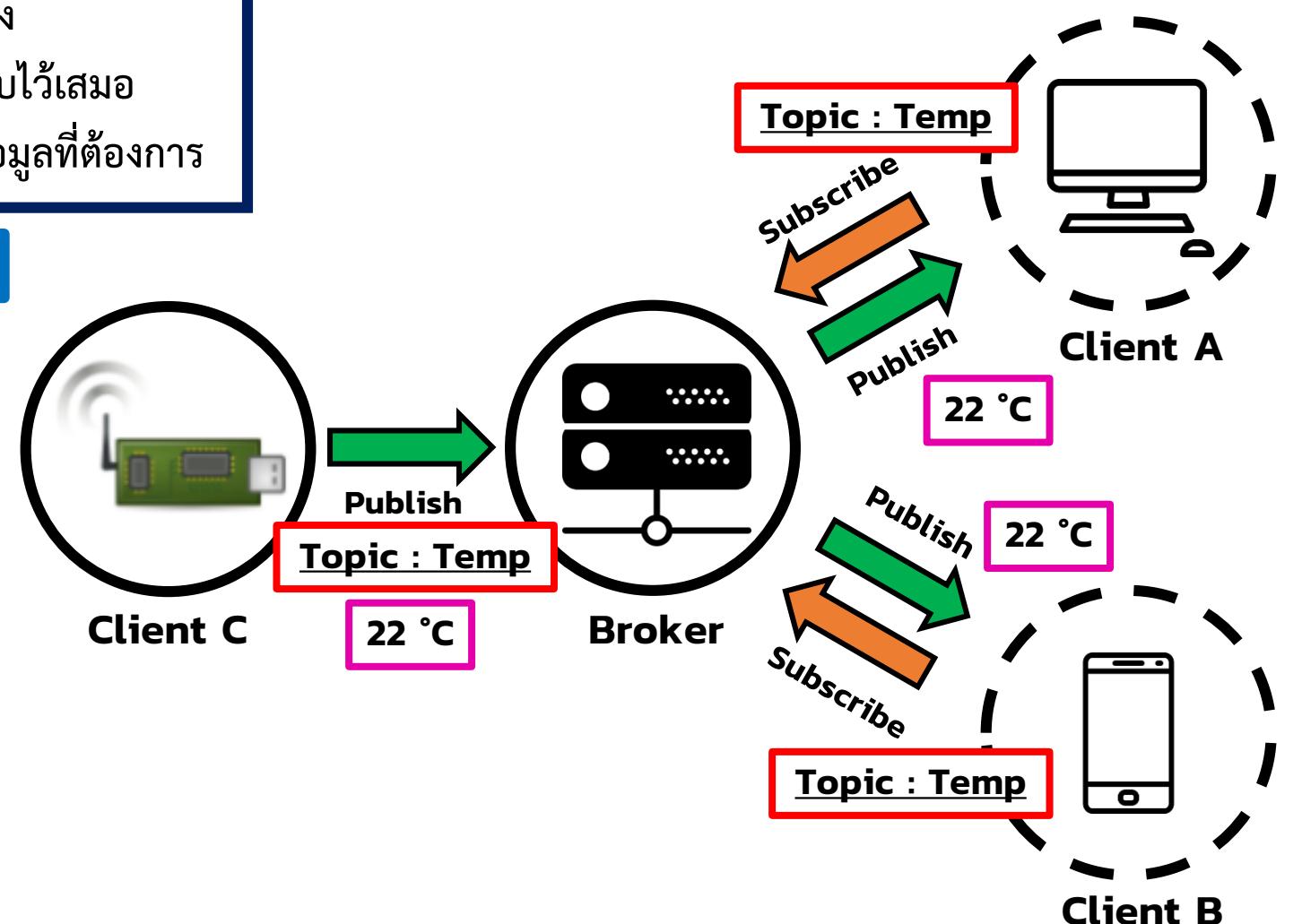
Module 2 : NETPIE 2020 Portal

การส่งข้อมูลบน MQTT จะมี Topic เป็นตัวอ้างอิงหลัก ซึ่ง

- ข้อมูลที่จะ Publish ไปยัง Broker ต้องมี Topic กำกับไว้เสมอ
- ส่วนของ Subscribe ก็ต้องอ้างอิง Topic เพื่อเรียกข้อมูลที่ต้องการ

ยกตัวอย่างการส่งข้อมูลแบบ MQTT

- Client C นั้น Publish ข้อมูลบน Topic ที่มีชื่อว่า Temp และมี Client A ที่ Subscribe Topic Temp ไว้อยู่
- โดยข้อมูลจาก Client C นั้นคือ อุณหภูมิ ที่มีค่าเท่ากับ 22 °C ดังนั้น Client A จะได้รับข้อมูลด้วย
- ภายหลัง Client B ทำการ Subscribe Topic Temp ด้วย ดังนั้นข้อมูล อุณหภูมิที่ถูกเก็บไว้จะถูกส่งไปยัง Client B ทันทีหลังจากการ Subscribe



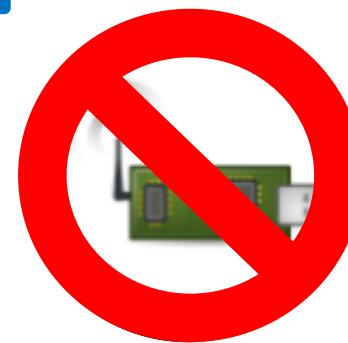
Module 2 : NETPIE 2020 Portal

การส่งข้อมูลบน MQTT จะมี Topic เป็นตัวอ้างอิงหลัก ซึ่ง

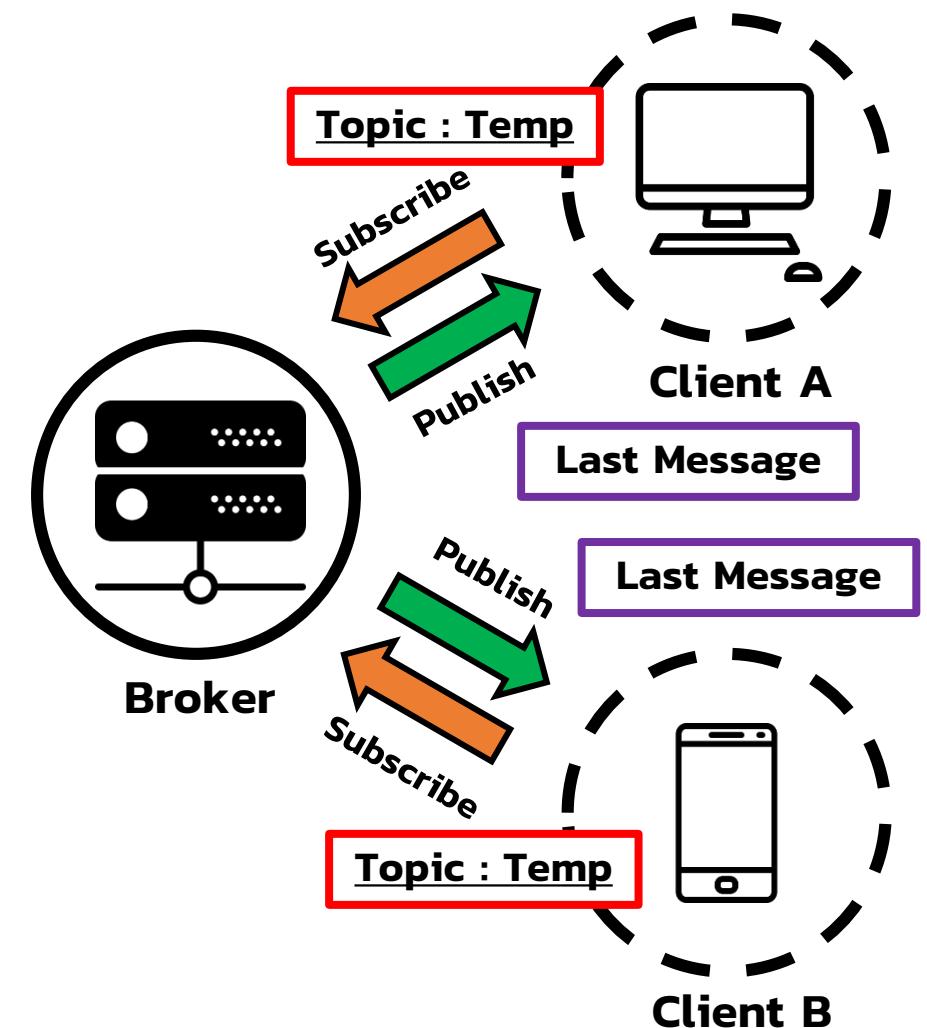
- ข้อมูลที่จะ Publish ไปยัง Broker ต้องมี Topic กำกับไว้เสมอ
- ส่วนของ Subscribe ก็ต้องอ้างอิง Topic เพื่อเรียกข้อมูลที่ต้องการ

ยกตัวอย่างการส่งข้อมูลแบบ MQTT

- Client C นั้น Publish ข้อมูลบน Topic ที่มีชื่อว่า Temp และมี Client A ที่ Subscribe Topic Temp ไว้อยู่
- โดยข้อมูลจาก Client C นั้นคือ อุณหภูมิ ที่มีค่าเท่ากับ 22°C ดังนั้น Client A จะได้รับข้อมูลด้วย
- ภายหลัง Client B ทำการ Subscribe Topic Temp ด้วย ดังนั้นข้อมูล อุณหภูมิที่ถูกเก็บไว้จะถูกส่งไปยัง Client B ทันทีหลังจากทำการ Subscribe
- แต่เมื่อ Client C นั้นขาดการเชื่อมต่อทำให้ไม่มีข้อมูล Publish ไปยัง Broker สิ่งที่ Client A และ B จะได้แสดงผลคือ ข้อความสุดท้ายที่ Client C ส่งไว้



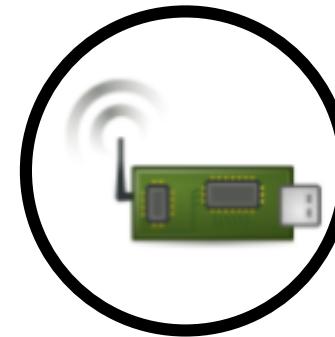
Client C



ข้อความที่ส่งไปยัง NETPIE2020 ผ่าน MQTT แบ่งเป็น 2 ชนิด

1

Message



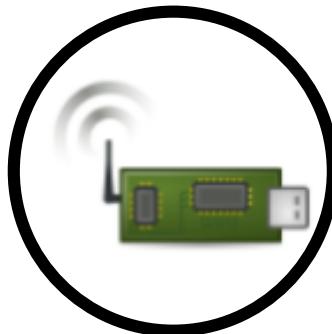
“Hello NETPIE2020”

MQTT Protocol



2

Data



MQTT Protocol
Temp = 25

MQTT บน NETPIE2020 แบ่งการส่งเป็น 2 แบบคือ

1

Message

เป็นการส่งข้อความผ่าน MQTT โดยอ้างอิง Topic
ชึ้นรูปแบบ topic ดื้อ **@msg/topic**

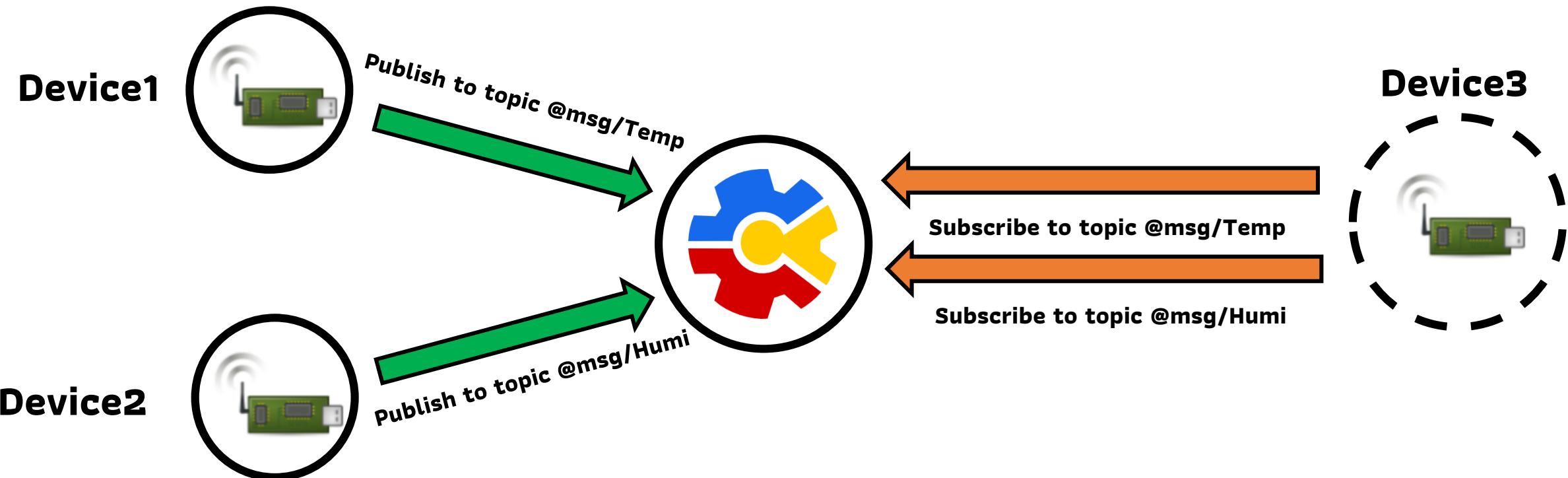
2

Shadow [Data]

เป็นการส่งข้อความผ่าน MQTT ไปยัง Device Shadow เพื่อบันทึก
ข้อความล่าสุด มีรูปแบบ topic ดื้อ **@shadow/data/update**
ชึ้นรูปแบบข้อความเป็น JSON ตัวอย่างเช่น **{data:{temp:24}}**

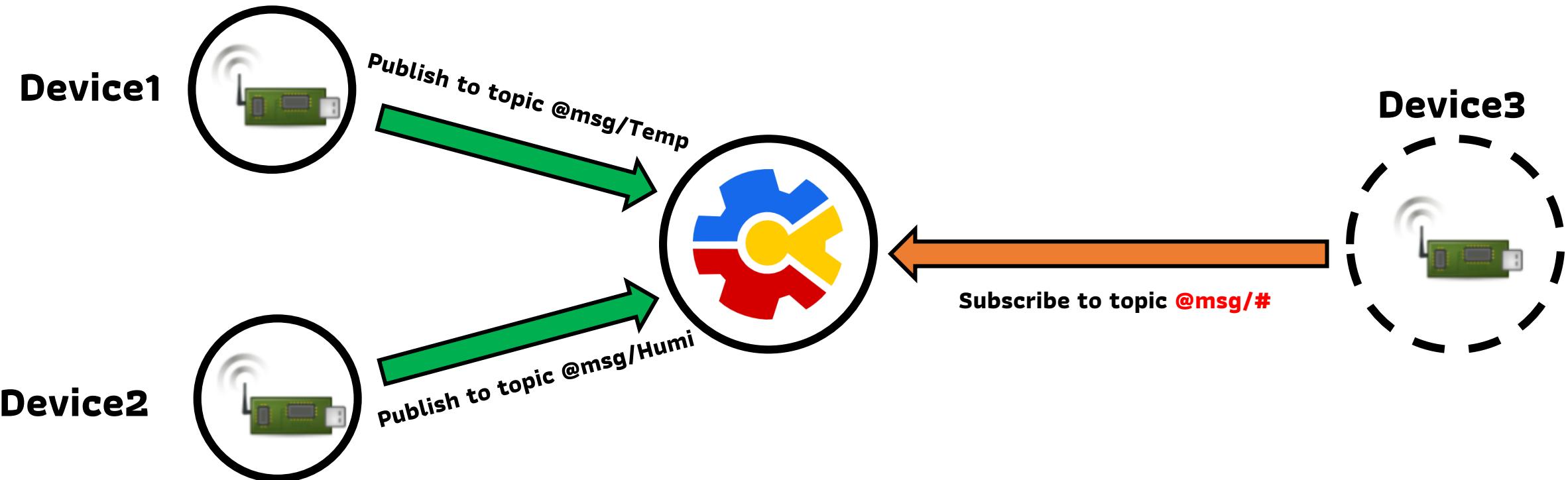
Topic และ Wildcard

หากต้องการให้ Device 3 ได้รับข้อมูลความจาก Device1 และ Device2
จำเป็นต้อง Subscribe 2 Topic

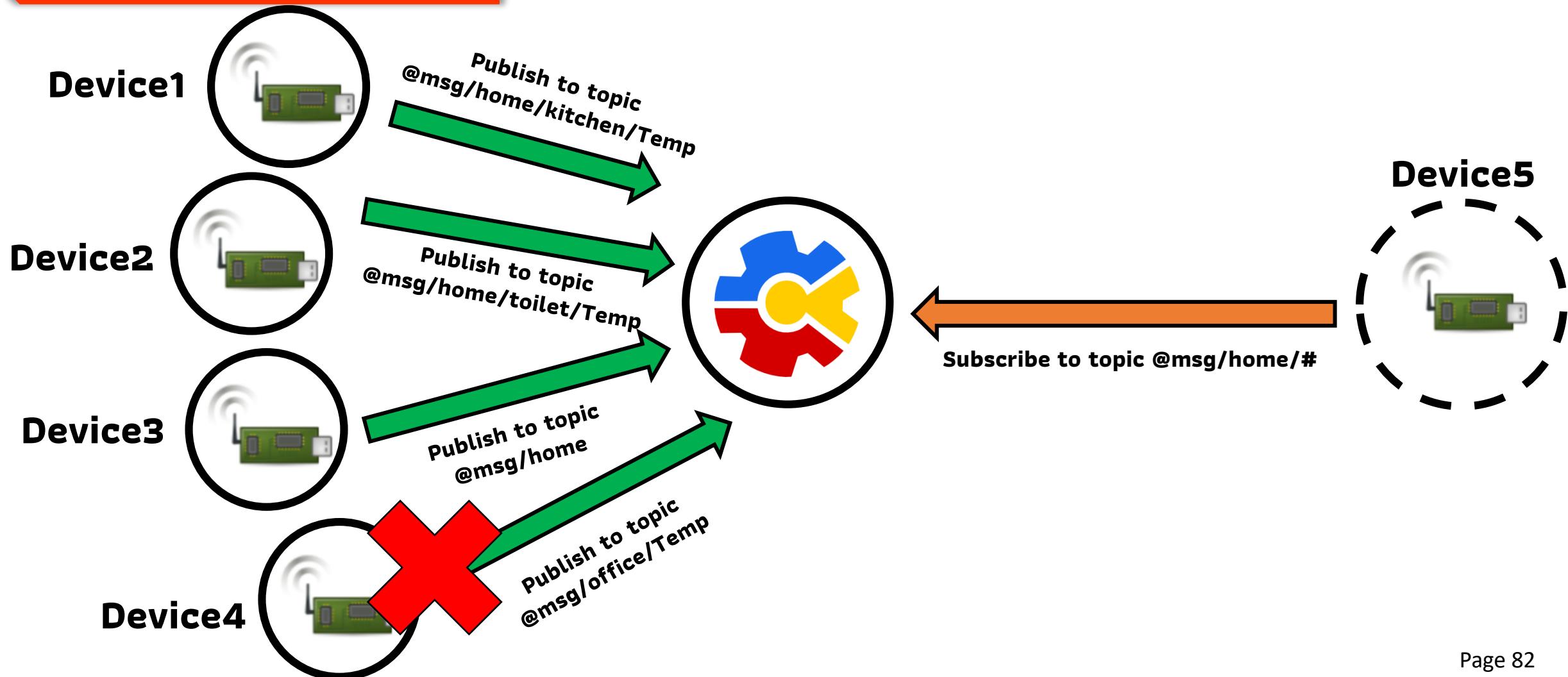


Topic และ Wildcard

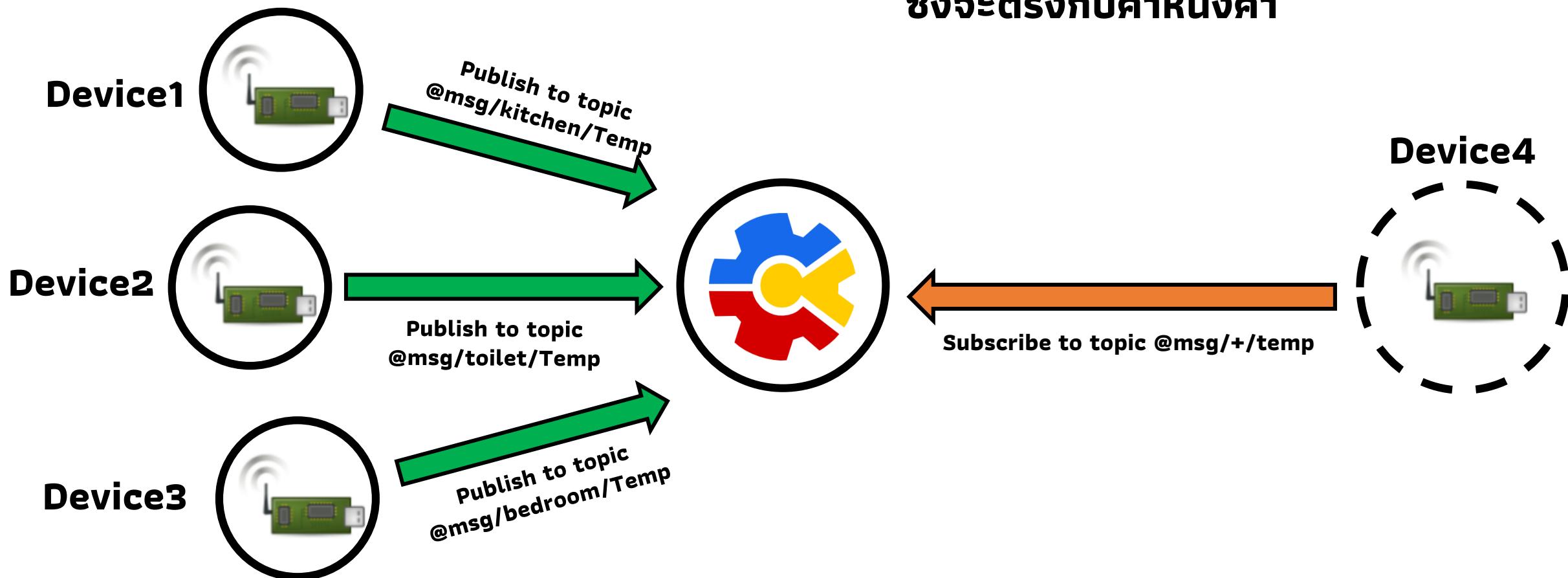
สามารถใช้สัญลักษณ์ # เป็นส่วนหนึ่งใน topic ซึ่งจะตรงกับคำทุกคำ และทุกจำนวน / ของคำถ้า Subscribe topic /# จะเทียบเท่ากับการ Subscribe All



Topic ແບບ Wildcard



Topic และ Wildcard





Module 3

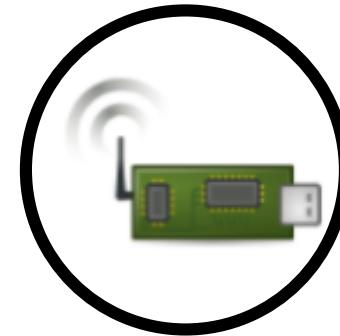
Device Data Management



ข้อความที่ส่งไปยัง NETPIE2020 ผ่าน MQTT แบ่งเป็น 2 ชนิด

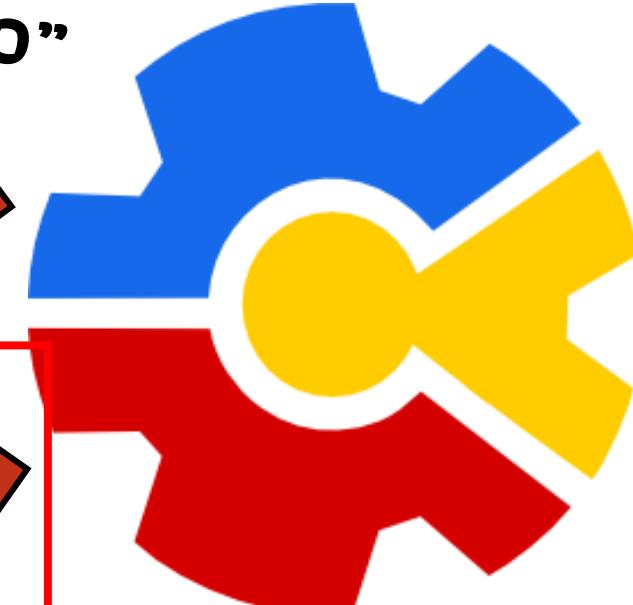
1

Message



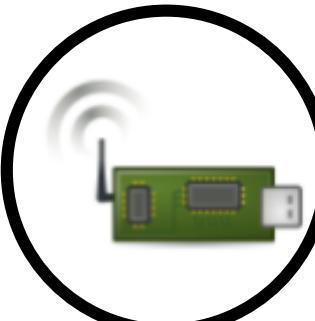
“Hello NETPIE2020”

MQTT Protocol



2

Data



MQTT Protocol
Temp = 25



NETPIE2020 มีการจัดการข้อมูลอุปกรณ์แบ่งเป็น 4 ส่วนหลักๆ

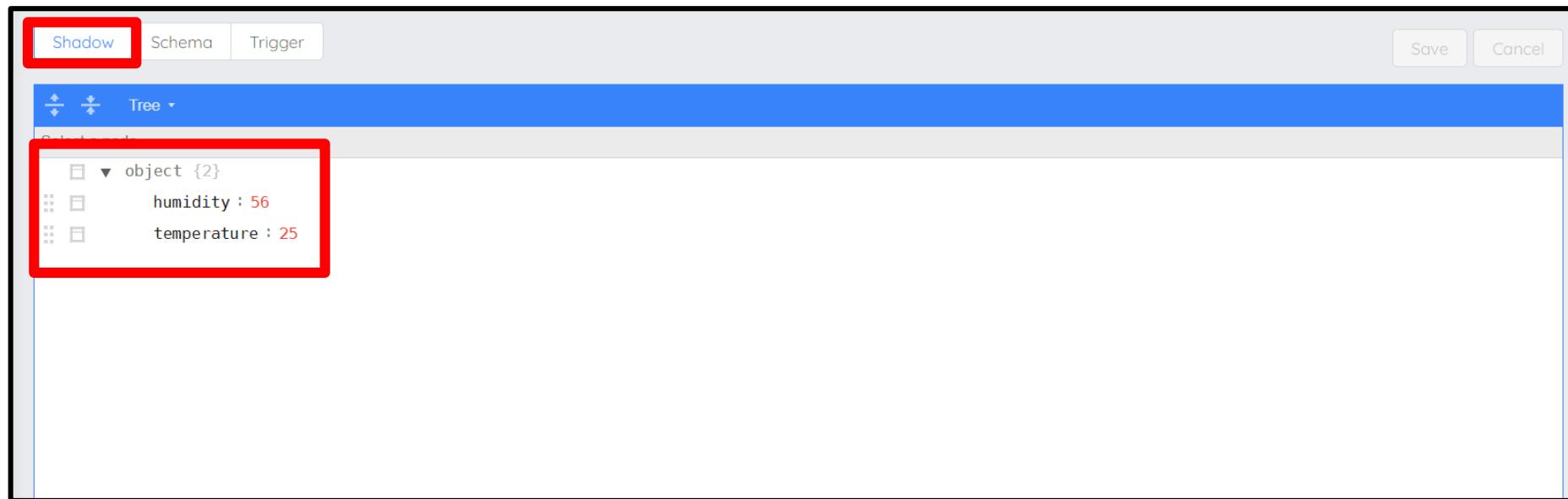
- 1 **Device Shadow** : ฐานข้อมูลล่าสุดของอุปกรณ์
- 2 **Device Schema** : โครงสร้างข้อมูลของอุปกรณ์
- 3 **Device Trigger** : เงื่อนไขการส่งข้อมูลอุปกรณ์
- 4 **Event Hooks** : รูปแบบการส่งข้อมูลอุปกรณ์



NETPIE2020

Device Shadow

1. Device Shadow Data เช่น ข้อมูลจากเซนเซอร์ เป็นต้น
 2. Device Shadow State เช่น สถานะ online/offline ของ Device เป็นต้น



Device Shadow

MQTT Topic ที่เกี่ยวข้องกับการจัดการ Device Shadow คือ

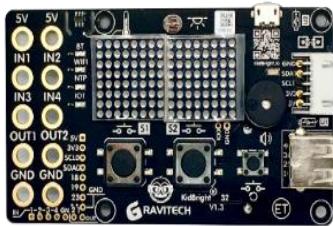
Shadow Topic

ใช้สำหรับจัดการ Device Shadow ของตัวเอง มีทั้ง Publish เพื่อแก้ไขข้อมูล Shadow และ Subscribe เพื่อรับข้อมูล Shadow โดยมีรูปแบบดังนี้

Publish Topic	Description	Subscribe Topic
@shadow/data/update	เป็นการอัพเดตค่าใน Shadow Data โดยส่ง Payload ในรูปแบบ JSON	@shadow/data/updated

Device Shadow

ตัวอย่างการส่งข้อมูลในรูปแบบ JSON



KidBright

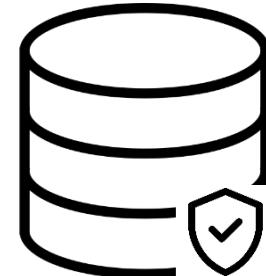
@shadow/data/update

Publish : { "data": { "Temp" : 24, "light" : 80 }}



NETPIE2020

"Temp" : 24
"light" : 58



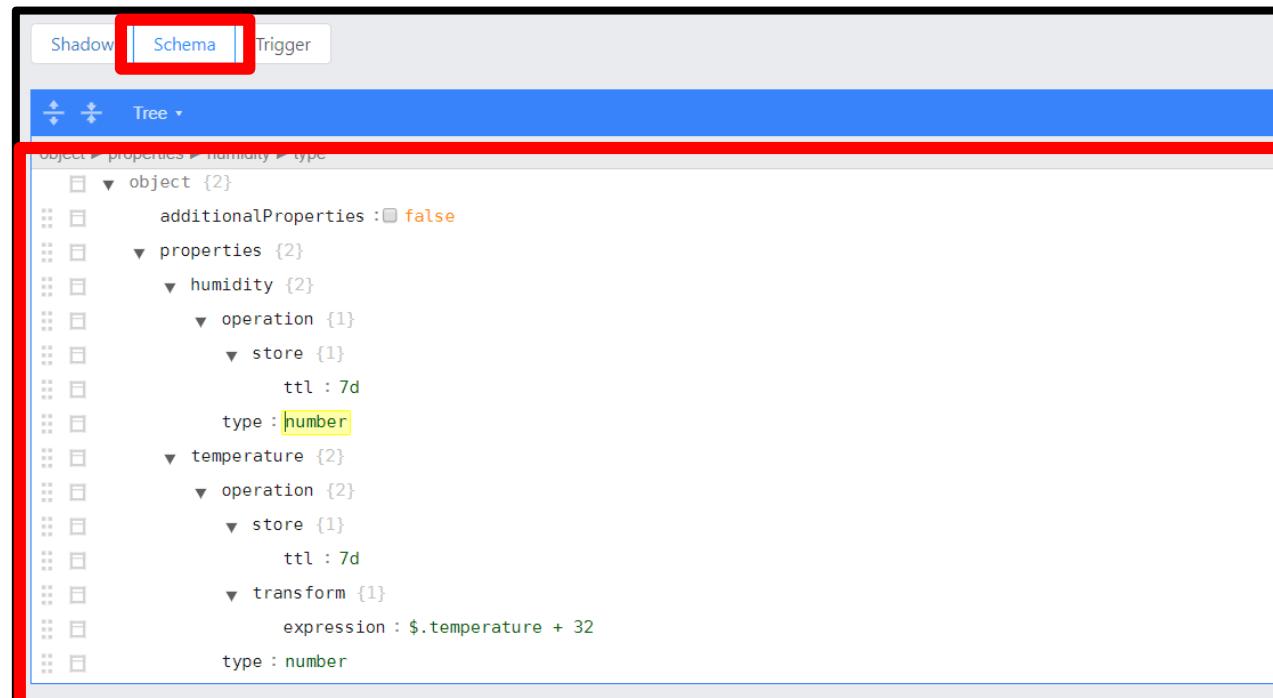
Device Shadow

แต่การจะจัดเก็บข้อมูลลง Device Shadow ได้นั้นจำเป็นต้องสร้าง Device Schema ก่อน

Device Schema

Device Schema គឺ គូរសរាយខ្លួនមូលដ្ឋានកំណត់ដើម្បីធ្វើឱ្យការងារទាំងអស់នៃការប្រើប្រាស់ Device Shadow ត្រូវបានការពារ។ ការប្រើប្រាស់ Device Schema ជំនួយនេះ នឹងធ្វើឡើងថា Server ត្រូវបានការពារក្នុងការប្រើប្រាស់ទិន្នន័យ។

- ពារចាយការងារទាំងអស់នៃការប្រើប្រាស់ទិន្នន័យ
- ការប្រើប្រាស់ការងារទាំងអស់នៃការប្រើប្រាស់ទិន្នន័យ ដើម្បីផ្តល់ព័ត៌មានអនុញ្ញាតទៅក្នុងការប្រើប្រាស់ទិន្នន័យ
- **ការប្រើប្រាស់ការងារទាំងអស់នៃការប្រើប្រាស់ទិន្នន័យ ដើម្បីផ្តល់ព័ត៌មានអនុញ្ញាតទៅក្នុងការប្រើប្រាស់ទិន្នន័យ**



Device Schema

การประกษา Device Schema ในรูปแบบ JSON

```
{  
    "additionalProperties": false,  
    "properties": {  
        "light": {  
            "operation": {  
                "store": {  
                    "ttl": "7d"  
                }  
            },  
            "type": "number"  
        },  
        "temperature": {  
            "operation": {  
                "store": {  
                    "ttl": "7d"  
                }  
            },  
            "transform": {  
                "expression": "[$.temperature]*1.8 + 32"  
            }  
        },  
        "type": "number"  
    }  
}
```

Device Schema จะประกอบด้วย 2 ส่วนหลักคือ

additionalProperties

Properties

Device Schema

การประมวล Device Schema ในรูปแบบ JSON

```
{  
    "additionalProperties": false,  
    "properties": {  
        "light": {  
            "operation": {  
                "store": {  
                    "ttl": "7d"  
                }  
            },  
            "type": "number"  
        },  
        "temperature": {  
            "operation": {  
                "store": {  
                    "ttl": "7d"  
                }  
            },  
            "transform": {  
                "expression": "[$.temperature]*1.8 + 32"  
            }  
        },  
        "type": "number"  
    }  
}
```

additionalProperties

เป็นสถานะการอนุญาตให้บันทึกข้อมูลลง Shadow หรือ Timeseries Database
ในกรณีที่ข้อมูลไม่ตรงตามที่กำหนด Properties มี 2 สถานะ คือ

true คือ อนุญาตให้บันทึกลง Shadow หรือ Timeseries Database

false คือ ไม่อนุญาตให้บันทึกเฉพาะส่วนที่ไม่ตรงตาม Properties

อย่างในตัวอย่าง properties 2 ค่า คือ humidity และ temperature

ถ้าข้อมูลที่ส่งมาคือ temp, humid, light

additionalProperties = true จะจัดเก็บทั้ง temperature, humidity และ light

additionalProperties = false จะจัดเก็บเพียง temperature, light

Device Schema

การประกษา Device Schema ในรูปแบบ JSON

```
{
  "additionalProperties": false,
  "properties": {
    "light": {
      "operation": {
        "store": {
          "ttl": "7d"
        }
      },
      "type": "number"
    },
    "temperature": {
      "operation": {
        "store": {
          "ttl": "7d"
        }
      },
      "transform": {
        "expression": "[$.temperature]*1.8 + 32"
      }
    },
    "type": "number"
  }
}
```

Properties

เริ่มจากกำหนดชื่อฟล็อกต์ [ตัวอย่างคือ “light” และ “Temperature”] และกำหนดคุณสมบัติของแต่ละฟล็อกซึ่งแบ่งเป็น 2 ส่วนคือ

Operation สำหรับตั้งค่าการจัดการข้อมูลในฟล็อกนั้นๆ ประกอบไปด้วย

store สำหรับตั้งค่าการเก็บข้อมูลลง Timeseries Database

ttl คือ ระยะเวลาของการเก็บข้อมูลใน Timeseries Database ซึ่งแต่ละข้อมูลมีอายุการเก็บครบทามกำหนดจะถูกลบทิ้งอัตโนมัติ
ถ้าต้องการจัดการเก็บข้อมูลระบบจำเป็นต้องกำหนดค่านี้ มีหน่วยเป็น ms[มิลลิวินาที], s[วินาที], m[นาที], h[ชั่วโมง], d[วัน], y[ปี]

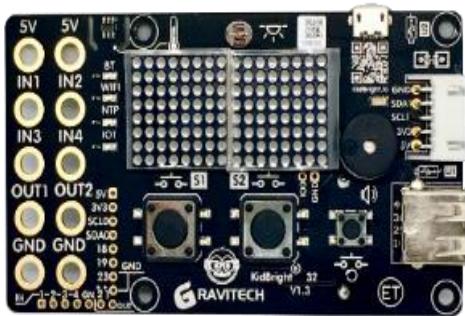
Transform การแปลงข้อมูลก่อนจัดเก็บ

expression คือ สูตรหรือวิธีการแปลงข้อมูลก่อนจัดเก็บ
ตัวอย่าง แปลงหน่วยอุณหภูมิจากเซลเซียสเป็นฟาเรนไฮต์ = [\$.temperature*1.8] + 32

Type คือ ชนิดข้อมูลในฟล็อกนั้นๆ ได้แก่ number, string, array, object

Module 3 : Device Data Management

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB



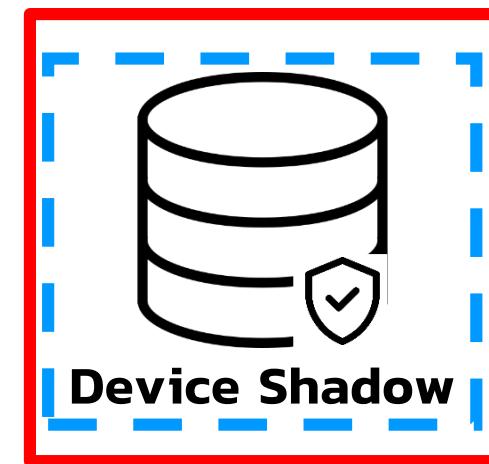
KidBright

@shadow/data/update

Publish : { "data": { "Temp" : 24, "light" : 80, "place" : "NECTEC" }}



NETPIE2020



**"Temp" : 75
"light" : 80
"place" : "NECTEC"**

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

จัดเตรียม Device Schema ก่อนส่งข้อมูลจาก KidBright

The screenshot shows the NETPIE_Device interface for a device named 'Device1'. On the left, there's a sidebar with options like Overview, Device List, Device Groups, Freeboard, Event Hooks, and Setting. The main area has tabs for Description, Key, and Schema. The Schema tab is currently active, indicated by a red box and a hand cursor pointing at it. Below the tabs is a tree view with a single node: 'object {0}' under '(empty object)'. At the bottom right of the interface are Save and Cancel buttons.

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

จัดเตรียม Device Schema ก่อนส่งข้อมูลจาก KidBright

The screenshot shows the NETPIE Device Management interface. On the left, there's a sidebar with options like Overview, Device List, Device Groups, Freeboard, Event Hooks, and Setting. The main area shows a device named 'Device1' under 'NETPIE_Training / device'. The 'Schema' tab is selected. At the bottom, there's a code editor with a toolbar above it. The 'Code' button in the toolbar is highlighted with a red box and has a cursor pointing at it. A callout box with the text 'เลือกจาก Tree เป็น Code' (Select from Tree as Code) is positioned over the 'Code' button. The right side of the screen shows the 'Key' section with fields for Client ID, Token, Secret, and Status (Offline). There are also 'Edit' and 'Save' buttons.

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

Device Schema ในรูปแบบ JSON

```
{  
  "additionalProperties": false,  
  "properties": {  
    "light": {  
      "operation": {  
        "store": {  
          "ttl": "7d"  
        }  
      },  
      "type": "number"  
    },  
    "temperature": {  
      "operation": {  
        "store": {  
          "ttl": "7d"  
        },  
        "transform": {  
          "expression": "[($temperature)*1.8] + 32"  
        }  
      },  
      "type": "number"  
    },  
    "place": {  
      "operation": {  
        "store": {  
          "ttl": "7d"  
        }  
      },  
      "type": "string"  
    }  
  }  
}
```

คัดลอกทั้งหมดไปไว้ใน
Device Schema

Module 3 : Device Data Management

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

Status : Offline

Shadow Schema Trigger

Code

```
20
21
22 "temperature": {
23   "operation": {
24     "store": {
25       "ttl": "7d"
26     },
27     "transform": {
28       "expression": "(($.temperature)*1.8) + 32"
29     }
30   },
31   "type": "number"
32 },
33 "place": {
34   "operation": {
35     "store": {
36       "ttl": "7d"
37     }
38   },
39   "type": "string"
40 }
41
42 }
43 |
```

Ln:43 Col:1

powered by ace

Save Cancel

คลิก Save เพื่อบันทึก
Device Schema

Copyright © 2018-2020 Created by NETPIE

Module 3 : Device Data Management

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

Device Schema ئۇشۇمچى JSON

ตั้งค่า **additionalProperties = false**
เพื่อไม่ให้บันทึกค่าอื่นที่นอกเหนือจาก
properties ลง Shadow

- จัดเก็บตัวแปรประเภทแรกด้วย **light**
โดยมีคุณสมบัติดังนี้
จัดเก็บจำนวน 7 วัน
จัดเก็บตัวแปรประเภท number

```
"additionalProperties": false
"properties": {
  "light": {
    "operation": {
      "store": {
        "ttl": "7d"
      }
    },
    "type": "number"
  },
  "type": "string"
}
```

```
"temperature": {
    "operation": {
        "store": {
            "ttl": "7d"
        },
        "transform": {
            "expression": "[($.temperature)*1.8] + 32"
        }
    },
    "type": "number"
},
"place": {
    "operation": {
        "store": {
            "ttl": "7d"
        }
    },
    "type": "string"
}
}
```

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

Device Schema ในรูปแบบ JSON

```
{
  "additionalProperties": false,
  "properties": {
    "light": {
      "operation": {
        "store": {
          "ttl": "7d"
        }
      },
      "type": "number"
    }
  }
}
```

```
"temperature": {
  "operation": {
    "store": {
      "ttl": "7d"
    },
    "transform": {
      "expression": "[[$.temperature]*1.8] + 32"
    }
  },
  "type": "number"
},
"place": {
  "operation": {
    "store": {
      "ttl": "7d"
    }
  },
  "type": "string"
}
```

จัดเก็บตัวแปรคือ **temperature** โดยมีคุณสมบัติดังนี้

- จัดเก็บจำนวน 7 วัน
- แปลงข้อมูลอุณหภูมิ
- จัดเก็บตัวแปรประเภท number

จัดเก็บตัวแปรคือ **place** โดยมีคุณสมบัติดังนี้

- จัดเก็บจำนวน 7 วัน
- จัดเก็บตัวแปรประเภท string

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

เปิดไฟล์ Workshop4.ino ขึ้นมา

Coding ใน Workshop4.ino แบ่งออกเป็น 3 ส่วน

1 ส่วนที่ 1 การเรียกใช้ Library และ ประการตัวแปร

เป็นการเรียกใช้ Library ต่างๆ

ประการขาของ LM73 เช่นเซอร์วัตอุณหภูมิบน KidBright

เป็นการประการตัวแปรและกำหนดค่าสำหรับเชื่อมต่อ WiFi และ MQTT

ประการตัวแปรต่างๆ

การกำหนดและเรียกใช้ชุดคำสั่ง MQTT

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>
```

```
#define LM73_ADDR 0x4D
```

```
const char* ssid = "Your_SSID";
const char* password = "Your_PASSWORD";
const char* mqtt_server = "broker.netpie.io";
const int mqtt_port = 1883;
const char* mqtt_Client = "Client_ID";
const char* mqtt_username = "Token";
const char* mqtt_password = "Password";
```

```
int temperature = 0;
int light = 0;
int analog_value = 0;
char msg[100];
```

```
WiFiClient espClient;
PubSubClient client[espClient];
```

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

2

ส่วนที่ 2 ส่วนของฟังก์ชันต่างๆ

```
float readTemperature() {  
    Wire1.beginTransmission(LM73_ADDR);  
    Wire1.write(0x00);  
    Wire1.endTransmission();  
    uint8_t count = Wire1.requestFrom(LM73_ADDR, 2);  
    float temp = 0.0;  
    if (count == 2) {  
        byte buff[2];  
        buff[0] = Wire1.read();  
        buff[1] = Wire1.read();  
        temp += [int][buff[0] << 1];  
        if [buff[1] & 0b10000000] temp += 1.0;  
        if [buff[1] & 0b01000000] temp += 0.5;  
        if [buff[1] & 0b00100000] temp += 0.25;  
        if [buff[0] & 0b10000000] temp *= -1.0;  
    }  
    return temp;  
}
```

ฟังก์ชันสำหรับอ่านค่าอุณหภูมิจาก
เซนเซอร์ LM73 บนบอร์ด KidBright

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

2

ส่วนที่ 2 ส่วนของฟังก์ชันต่างๆ

```
float readlight() {  
    analog_value = analogRead(36);  
    if [analog_value > 1000] {  
        analog_value = 1000;  
    }  
    int light = (1000 - analog_value) * 100 / 1000;  
    return light;  
}
```

ฟังก์ชันสำหรับอ่านค่าแสงจากเซนเซอร์ LDR บนบอร์ด KidBright

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

2

ស៊ូបទទី 2 សំគាល់នូវការបង្កើតផ្ទាល់ពីកម្មវិធី

```
void reconnect() {
    while (!client.connected()) {
        Serial.print("Attempting MQTT connection...");
        if (client.connect(mqtt_Client, mqtt_username, mqtt_password)) {
            Serial.println("connected");
        } else {
            Serial.print("failed, rc=");
            Serial.print(client.state());
            Serial.println("try again in 5 seconds");
            delay(5000);
        }
    }
}
```

ដឹងការបង្កើតផ្ទាល់ពីកម្មវិធី

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

2

ส่วนที่ 2 ส่วนของฟังก์ชันต่างๆ

```
void setup() {  
    Serial.begin(115200);  
    Wire1.begin(4, 5);  
    Serial.println("Starting...");  
  
    if (WiFi.begin(ssid, password)) {  
        while (WiFi.status() != WL_CONNECTED) {  
            delay(1000);  
            Serial.print(".");  
        }  
        Serial.println("WiFi connected");  
        Serial.println("IP address: ");  
        Serial.println(WiFi.localIP());  
        client.setServer(mqtt_server, mqtt_port);  
    }  
}
```

ฟังก์ชันการตั้งค่าต่างๆ

คำสั่งเรียกใช้งาน I2C ของ
เซนเซอร์ LM73

ในฟังก์ชันนี้จะทำการเชื่อมต่อ กับ WiFi และ MQTT
ตามค่าต่างๆ ที่ได้ตั้งค่าไว้

Module 3 : Device Data Management

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

3

ส่วนที่ 3 ส่วนของฟังก์ชันหลัก

```
void loop() {  
    if (!client.connected()) {  
        reconnect();  
    }  
    client.loop();  
    temperature = readTemperature();  
    light = readlight();  
    String place = "NECTEC";
```

เป็นชุดคำสั่งคงสถานะของการเชื่อมต่อและการทำงานต่างๆของ MQTT

ประจำการอ่านค่า อุณหภูมิ และกำหนดสถานที่ก่อนส่งไป NETPIE2020

```
String data = "{\"data\": {\"temperature\":\"" + String[temperature] + "\", \"light\":\"" + String[light] + "\",  
\"place\": \"\" + String[place] + "\"}}";  
Serial.println(data);  
data.toCharArray(msg, [data.length() + 1]);  
client.publish("@shadow/data/update", msg);  
delay[2000];  
}
```

เป็นชุดคำสั่งส่งข้อมูลความไปยัง Topic : @shadow/data/update ทุกๆ 2 วินาที

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

การส่งข้อมูลความในรูปแบบ JSON บน Arduino IDE

ข้อมูลที่ต้องการส่งคือ

```
{ "data" : { "temperature" : 25, "light" : 60, "place" : "NECTEC Thailand" } }
```

แต่บน Arduino IDE มอง " เป็นการประกาศข้อมูลเป็น String วิธีแก้คือการนำ \ ไว้ข้างหน้า " เพื่อให้โปรแกรมมอง " เป็นส่วนหนึ่งของข้อมูลความ

```
String data = "{\"data\": {\"temperature\":\"" + String[temperature] + ",\n\"light\":\"" + String[light] + ",\n\"place\":\"" + String[place] + "}}";
```

Module 3 : Device Data Management

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

```
Starting...
.WiFi connected
IP address:
192.168.1.105
Attempting NETPIE2020 connection...NETPIE2020 connected
{"data": {"temperature":34, "light":67, "place": "NECTEC" } }
{"data": {"temperature":34, "light":65, "place": "NECTEC" } }
{"data": {"temperature":34, "light":65, "place": "NECTEC" } }
```

ข้อความที่ส่งไปยัง NETPIE2020

Autoscroll Show timestamp Both NL & CR 115200 baud Clear output

Module 3 : Device Data Management

Workshop 4 : ส่งค่า Temperature และ Light จาก KidBright ไปยัง NETPIE2020 และบันทึกใน Timeseries DB

NETPIE_Training / device / Device1

Description

Key

- Client ID :
- Token :
- Secret :
- Status : Online

Shadow Schema Trigger Save Cancel

Tree Select a node...

obj [2]

- light : 60
- place : NECTEC
- temperature : 91.4

Data តើត្រូវបានពិភាក្សា Device Shadow

Device Trigger and Event Hook

เป็นระบบที่ผูกกับการเปลี่ยนแปลงข้อมูลของ Device Shadow เข้ากับการกระทำภายนอก [Event Hook] เช่น การตั้งค่าแจ้งเตือนตามสถานะต่างๆ ตามเงื่อนไขการทำงานของ Device ที่ถูกตั้งค่าไว้ โดยการใช้งาน Trigger จะประมวลผลรูปแบบ JSON มีลักษณะดังนี้

```
{  
    "enabled": true,  
    "trigger": [  
        {  
            "action": "EVENT_HOOK_NAME",  
            "event": "SHADOW.UPDATED or DEVICE.STATUSCHANGED",  
            "condition": "Operation List ==, !=, >, >=, <, <=, in",  
            "msg": "text",  
            "option": {}  
        }  
    ]  
}
```

Device Trigger and Event Hook

ໂដຍ Trigger Format សាមរណແບងໄດ້ 2 ส່ວນគັດ

```
{  
  "enabled": true, ←  
  "trigger": [  
    {  
      "action": "EVENT_HOOK_NAME",  
      "event": "SHADOW.UPDATED or DEVICE.STATUSCHANGED",  
      "condition": "Operation List ==, !=, >, >=, <, <=, in",  
      "msg": "text",  
      "option": {}  
    }  
  ]  
}
```

1. enable ດືວ ສານະເປີດ/ປັດກາຣໃຫ້ງານ Trigger

Device Trigger and Event Hook

โดย Trigger Format สามารถแบ่งได้ 2 ส่วนคือ

2. trigger เป็นการตั้งค่าต่างๆของ Trigger

```
{  
  "enabled": true,  
  "trigger": [  
    {  
      "action": "EVENT_HOOK_NAME",  
      "event": "SHADOW.UPDATED or DEVICE.STATUSCHANGED",  
      "condition": "Operation List ==, !=, >, >=, <, <=, in",  
      "msg": "text",  
      "option": {}  
    }  
  ]  
}
```

- action เมื่อเกิด Trigger จะให้กระทำอะไร โดยระบุชื่อ Event Hook ที่ต้องการทำ

- event ประเภทการเปลี่ยนแปลงข้อมูลของ Device Shadow มี 2 ประเภทคือ

SHADOW.UPDATED เกิดเมื่อ Device Shadow Data มีการเปลี่ยนแปลงตรงตาม condition ที่กำหนดไว้ [กรณีนี้จำเป็นต้องกำหนด condition ถ้าไม่กำหนดจะไม่เกิด Trigger]

DEVICE.STATUSCHANGED เกิดขึ้นเมื่อ Device เปลี่ยนสถานะการณ์เชื่อมต่อ Platform จาก Online เป็น Offline หรือ Offline เป็น Online

Device Trigger and Event Hook

โดย Trigger Format สามารถแบ่งได้ 2 ส่วนคือ

```
{  
  "enabled": true,  
  "trigger": [  
    {  
      "action": "EVENT_HOOK_NAME",  
      "event": "SHADOW.UPDATED or DEVICE.STATUSCHANGED",  
      "condition": "Operation List ==, !=, >, >=, <, <=, in",  
      "msg": "text",  
      "option": {}  
    }  
  ]  
}
```

Condition คือ เงื่อนไขการเปลี่ยนแปลงของ Device Shadow Data จะใช้ในกรณีที่ SHADOW.UPDATED

msg คือ ข้อความที่ต้องการให้แจ้งเตือนกรณีเกิด Trigger

option ใช้สำหรับกำหนดค่าอื่นๆ นอกเหนือจากที่มีระบุไว้ในข้างต้น

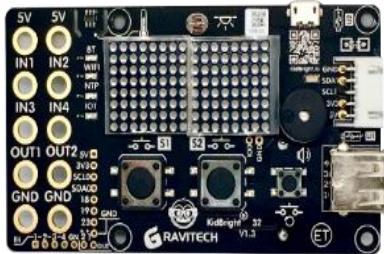
สำหรับการอ้างอิงตัวแปรใน Trigger สามารถศึกษาได้จาก

<https://docs.nexpie.io/device-config.html#device-trigger-and-event-hook>

Module 3 : Device Data Management

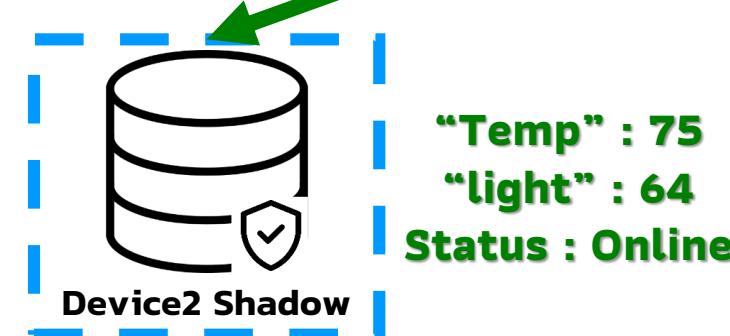
Device Trigger and Event Hook

▶ តัวอย่างการทำงานของ Trigger and Event Hook



KidBright

Publish : @shadow/data/update



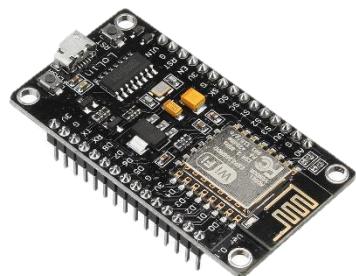
Event Hooks
LINE NOTIFY



Trigger : DEVICE.STATUSCHANGE

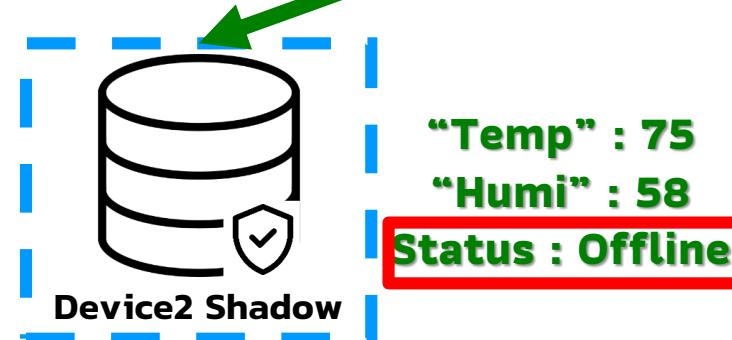
Device Trigger and Event Hook

▶ តัวอย่างการทำงานของ Trigger and Event Hook



Device2

Publish : @shadow/data/update



Trigger : DEVICE.STATUSCHANGE

Event Hooks
LINE NOTIFY



NETPIE2020

การสร้าง Event Hook

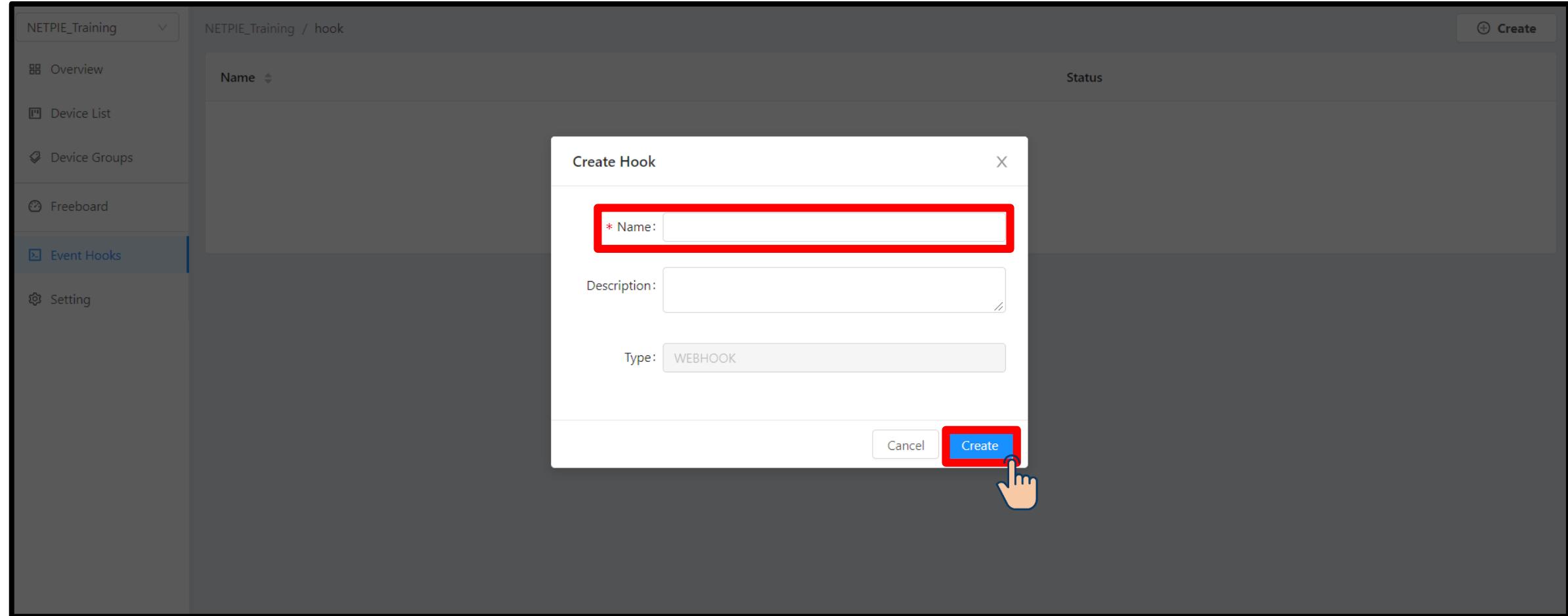
The screenshot shows the NETPIE_Device Data Management interface. On the left, there is a sidebar with the following menu items:

- NETPIE_Training
- Overview
- Device List
- Device Groups
- Freeboard
- Event Hooks** (highlighted with a red box and a hand cursor icon)
- Setting

The main content area is titled "NETPIE_Training / hook". It displays a table with two columns: "Name" and "Status". There is one entry in the table:

Name	Status
	No Data

การสร้าง Event Hook



การสร้าง Event Hook

The screenshot shows the NETPIE platform interface. On the left, there is a sidebar with the following menu items:

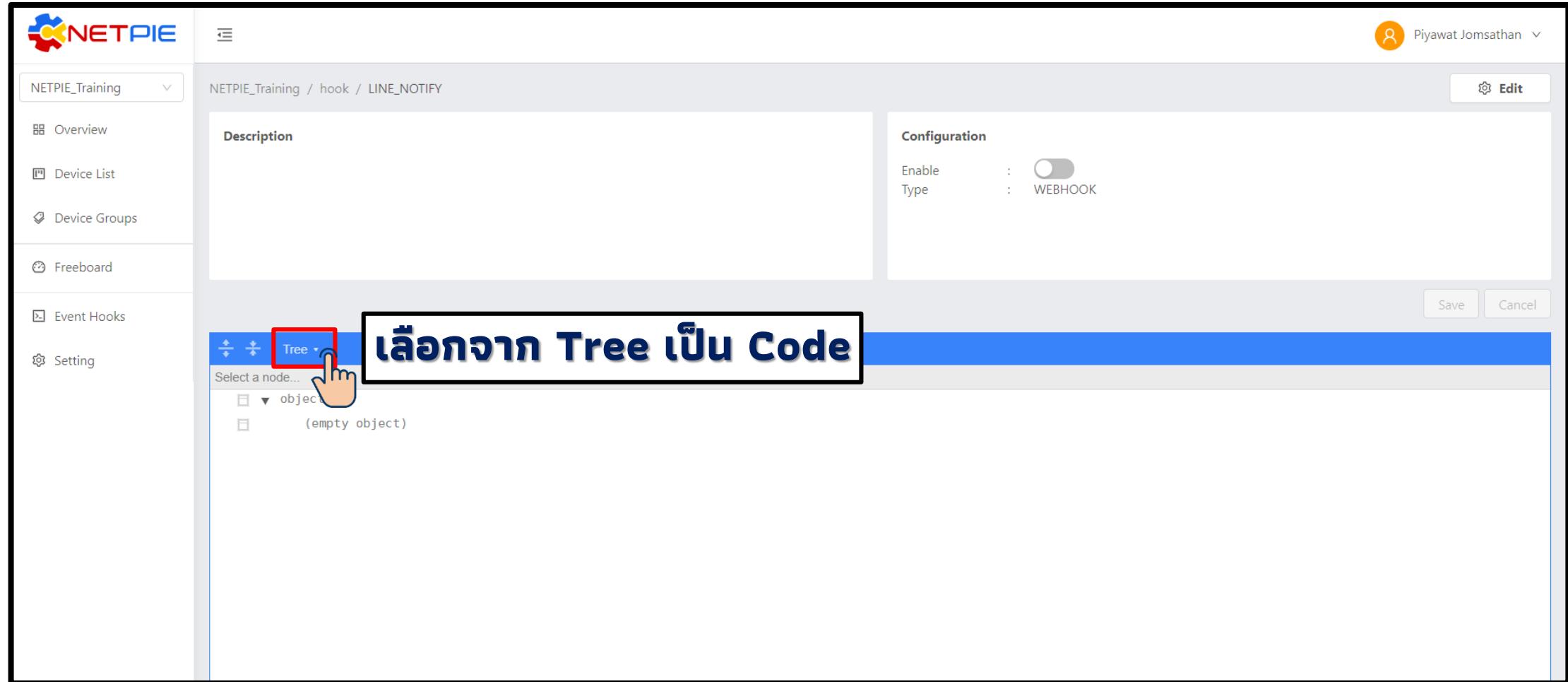
- NETPIE_Training (selected)
- Overview
- Device List
- Device Groups
- Freeboard
- Event Hooks (highlighted with a blue border)
- Setting

The main content area displays a table titled "NETPIE_Training / hook". The table has two columns: "Name" and "Status". There is one entry in the table:

Name	Status
LINE_NOTIFY	✓

A large red rectangle highlights the entire row for "LINE_NOTIFY". A hand cursor icon is positioned over the "LINE_NOTIFY" entry, indicating it is being selected or clicked.

การสร้าง Event Hook



การสร้าง Event Hook

Event Hook Example เป็นตัวอย่างการทำ Line Notify ซึ่งกำหนดค่าได้ 4 Attributes

```
{  
  "body": "message={{msg}}",  
  "header": {  
    "Authorization": "Bearer {{option.linetoken}}",  
    "Content-Type": "application/x-www-form-urlencoded"  
  },  
  "method": "POST",  
  "uri": "https://notify-api.line.me/api/notify"  
}
```

គណន៍កំណត់មុនដោយ NETPIE2020

การสร้าง Event Hook

The screenshot shows a web-based configuration interface for creating an Event Hook. The top navigation bar includes a dropdown for 'Smart_Factory_Io...', a breadcrumb path 'Smart_Factory_IoT_Challenge_2020 / hook / LINE_NOTIFY', and a 'Edit' button. On the left, a sidebar lists 'Overview', 'Device Lists', 'Device Groups', 'Event Hooks' (selected), and 'Setting'. The main area has a 'Description' section and a 'Configuration' section. In the 'Configuration' section, there are two fields: 'Enable' (with a toggle switch) and 'Type' (set to 'WEBHOOK'). A large blue callout box with the text 'เลือกให้เป็น Enable' (Select to enable) points to the 'Enable' switch, which is highlighted with a red box and a hand cursor icon. Below the configuration section is a tree view titled 'Tree' with the instruction 'Select a node...'. The tree structure shows an 'object' node with four children: 'uri' (set to <https://notify-api.line.me/api/notify>), 'method' (set to POST), 'header' (with two children: 'Authorization' set to 'Bearer {{option.linetoken}}' and 'Content-Type' set to 'application/x-www-form-urlencoded'), and 'body' (set to 'message={{msg}}').

การสร้าง Event Hook

Event Hook Example เป็นตัวอย่างการทำ Line Notify ซึ่งกำหนดค่าได้ 4 Attributes

```
{  
  "body": "message={{msg}}",  
  "header": {  
    "Authorization": "Bearer {{option.linetoken}}",  
    "Content-Type": "application/x-www-form-urlencoded"  
  },  
  "method": "POST",  
  "uri": "https://notify-api.line.me/api/notify"  
}
```

body ส่วนของข้อมูลในที่นี่คือ ข้อความที่จะส่งไปยังปลายทาง

header คือ ข้อมูลเพิ่มเติมที่ต้องการส่งไปยังปลายทาง เช่น Authorization, Content-Type เป็นต้น

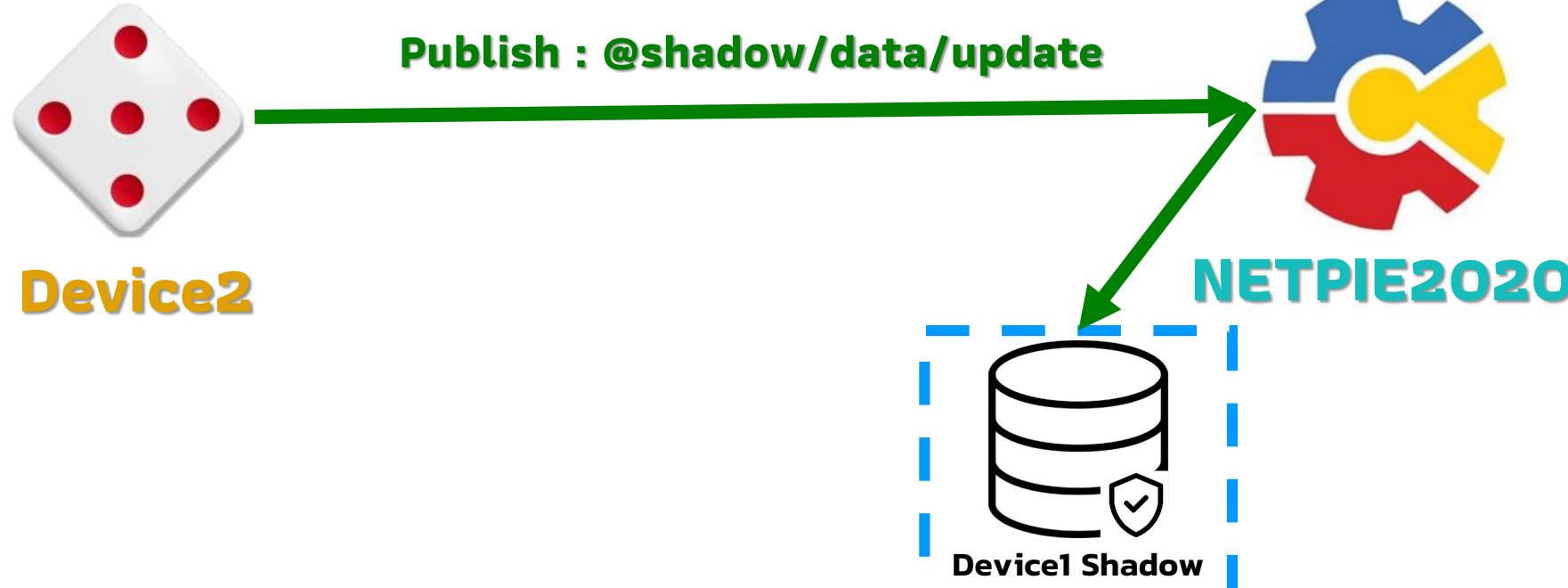
method คือ ส่วนที่กำหนดว่าปลายทางต้องการให้ส่งไปในแบบไหน GET, POST หรือ PUT

uri คือ Endpoint ปลายทางที่กำหนดว่าต้องการให้ส่งไปที่ใด

*** ใน Event Hook สามารถอ้างอิงตัวแปรต่างๆ ที่ส่งมาจาก Trigger ได้ โดยใช้สัญลักษณ์ {{...}} ครอบตัวแปรนั้นๆ ตัวอย่างเช่น จะอ้างอิงตัวแปร msg จาก Trigger จะใช้เป็น {{msg}} หรือจะใช้ linetoken ที่สร้างใน option จะใช้เป็น {{option.linetoken}}

Module 3 : Device Data Management

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



Trigger : SHADOW UPDATE & DEVICE.STATUSCHANGE

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device

เข้ามายัง Device1 เพื่อกำกการสร้าง Trigger

The screenshot shows the NETPIE Device Management interface for a device named 'Device2'. The left sidebar includes options like Overview, Device List, Device Groups, Freeboard, Event Hooks, and Setting. The main area displays a 'Description' section and a 'Key' section with Client ID, Token, Secret, and Status (Online). A blue navigation bar at the bottom has tabs for Shadow, Schema, and Trigger, with the Trigger tab highlighted and a red box and mouse cursor pointing to it. Below the tabs is a tree view labeled 'Select a node...' with an 'object {0}' node and an '(empty object)' node under it. Buttons for Save and Cancel are located in the bottom right corner.

Module 3 : Device Data Management

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device

The screenshot shows the NETPIE Device Management interface. On the left, there's a sidebar with options like Overview, Device List, Device Groups, Freeboard, Event Hooks, and Setting. The main area shows a device named 'Device2' under 'NETPIE_Training / device'. The 'Trigger' tab is selected. A red box highlights the 'Code' button in the toolbar above the code editor. A hand cursor is shown clicking on the 'Code' button. A large blue box with the text 'เลือกจาก Tree เป็น Code' (Select from Tree as Code) is overlaid on the code editor area.

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device

```
{  
  "enabled": true,  
  "trigger": [  
    {  
      "action": "LINE_NOTIFY",  
      "event": "SHADOW.UPDATED",  
      "condition": "$.temperature!= $$.temperature",  
      "msg": "My temperature 2 was change from {{$$.temperature}} to {{$_.temperature}}.",  
      "option": {  
        "linetoken": "Your_Token_LINE"  
      }  
    },  
    {  
      "action": "LINE_NOTIFY",  
      "event": "DEVICE.STATUSCHANGED",  
      "msg": "My Device {{$_.statustext}}, statuscode: {{$_.status}}.",  
      "option": {  
        "linetoken": "Your_Token_LINE"  
      }  
    }  
  ]  
}
```

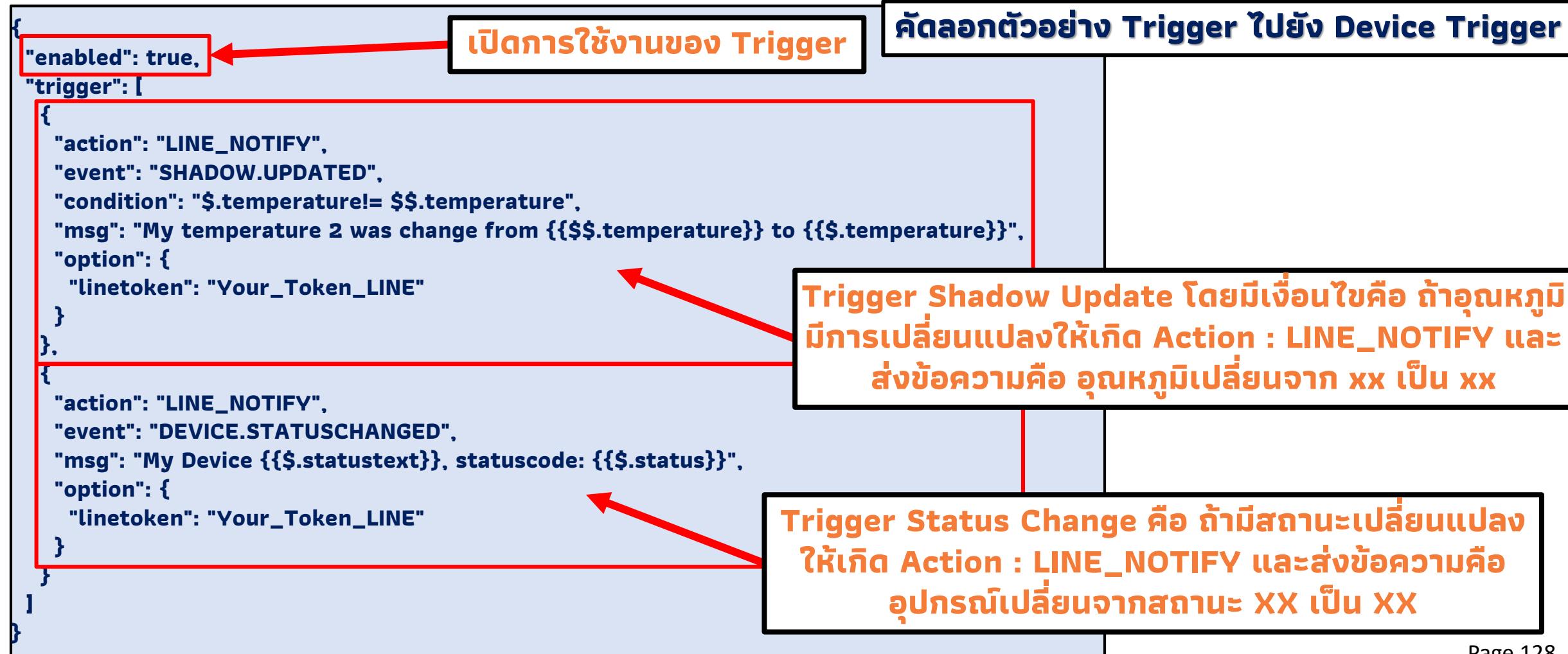
Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device

The screenshot shows a user interface for managing device triggers. On the left, there's a sidebar with links: Overview, Device List, Device Groups, Freeboard, Event Hooks, and Setting. The main area has tabs for Description, Shadow, Schema, and Trigger, with Trigger selected. In the Trigger tab, there's a 'Key' section with fields for Client ID, Token, Secret, and Status (set to Online). Below this is a code editor window titled 'Code' with the following JSON configuration:

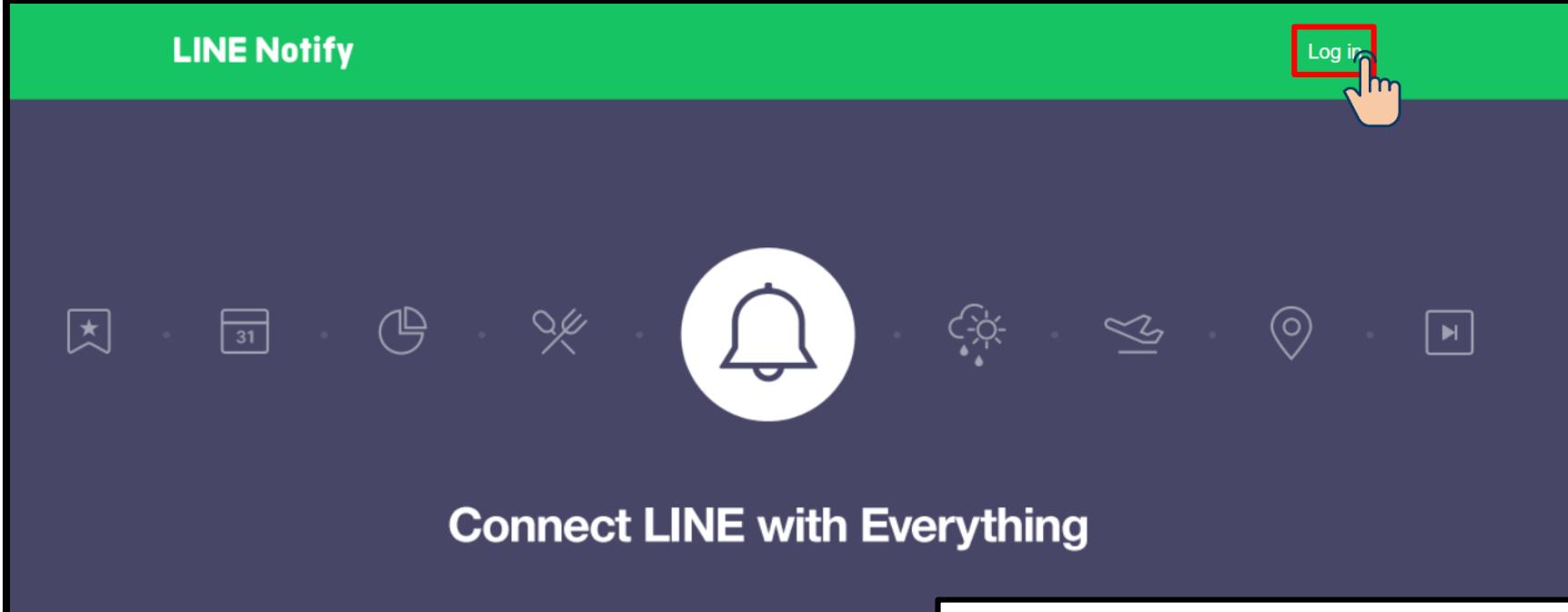
```
1  [
2    "enabled": true,
3    "trigger": [
4      {
5        "action": "LINE_NOTIFY",
6        "event": "SHADOW.UPDATED",
7        "condition": "$.temperature!= $$.temperature",
8        "msg": "My temperature 2 was change from {{$temperature}} to {{$temperature}}",
9        "option": {
10          "linetoken": "Your_Token_Line"
11        }
12      },
13      {
14        "action": "LINE_NOTIFY",
15        "event": "DEVICE.STATUSCHANGED",
16        "msg": "My Device {{$statustext}}, statuscode: {{$status}}",
17        "option": {
18          "linetoken": "Your_Token_Line"
19        }
20      }
21    ]
22 }
```

At the bottom of the code editor, it says 'Ln:1 Col:1'.

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



LINE Notify

Log in

Connect LINE with Everything

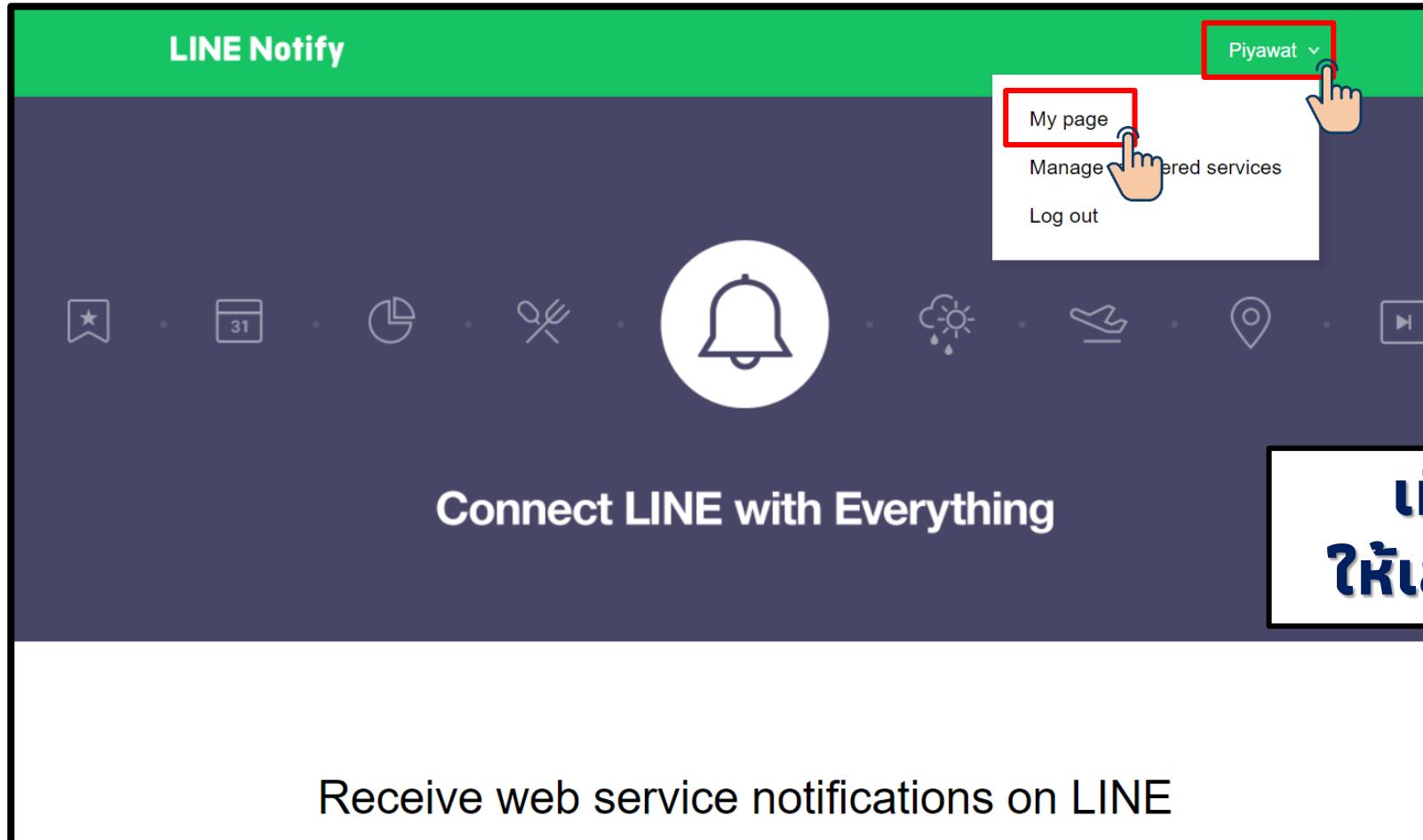
Token Line สามารถรับได้จาก
<https://notify-bot.line.me/en/>

Receive web service notifications on LINE

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device

เลือนลงมาข้างล่างแล้วเลือก
Generate Token

Generate access token (For developers)

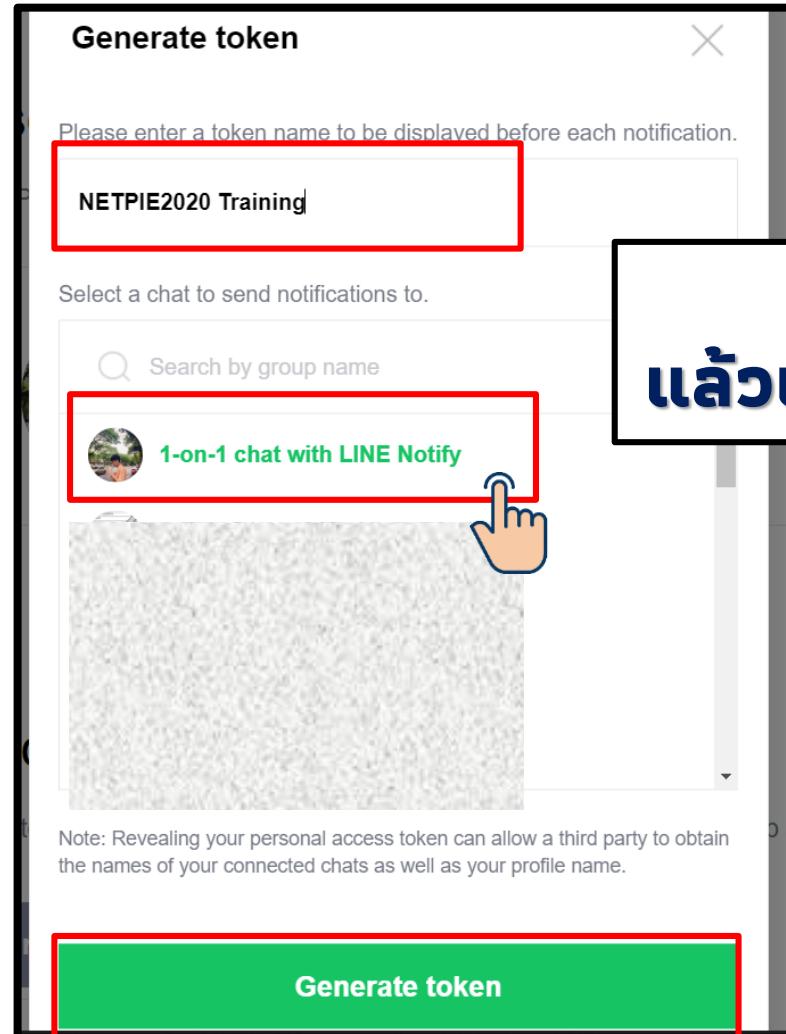
By using personal access tokens, you can configure notifications without having to add a web service.

Generate token

LINE Notify API Document

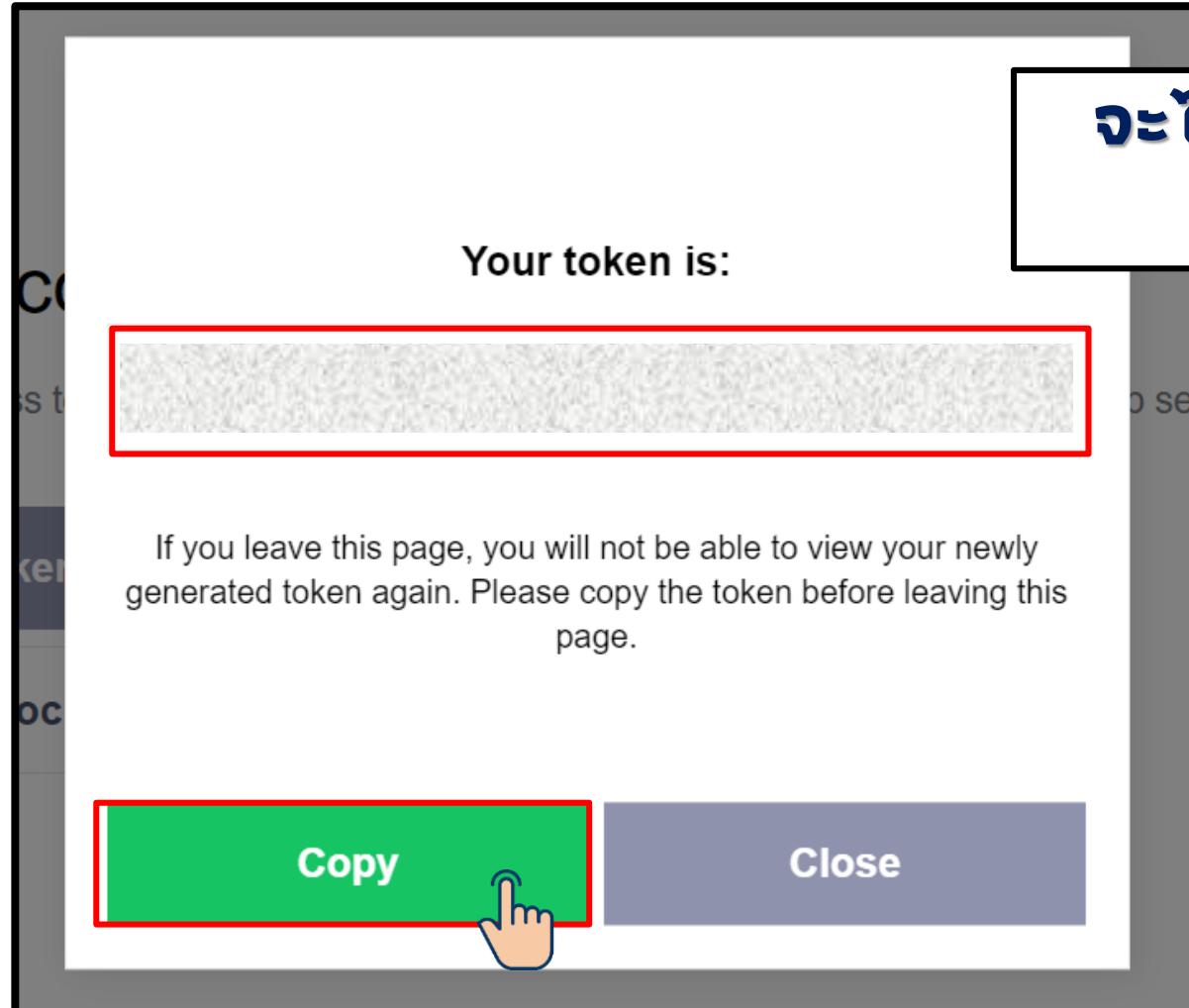


Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



ตั้งชื่อหัวข้อ การสนทนา
แล้วเลือกแชทกับตัวเอง และคลิก Generate Token

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



จะได้รับ Token ให้ Copy แล้วไปวางใน
linetoken ใน Device Trigger

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device

Shadow Schema Trigger

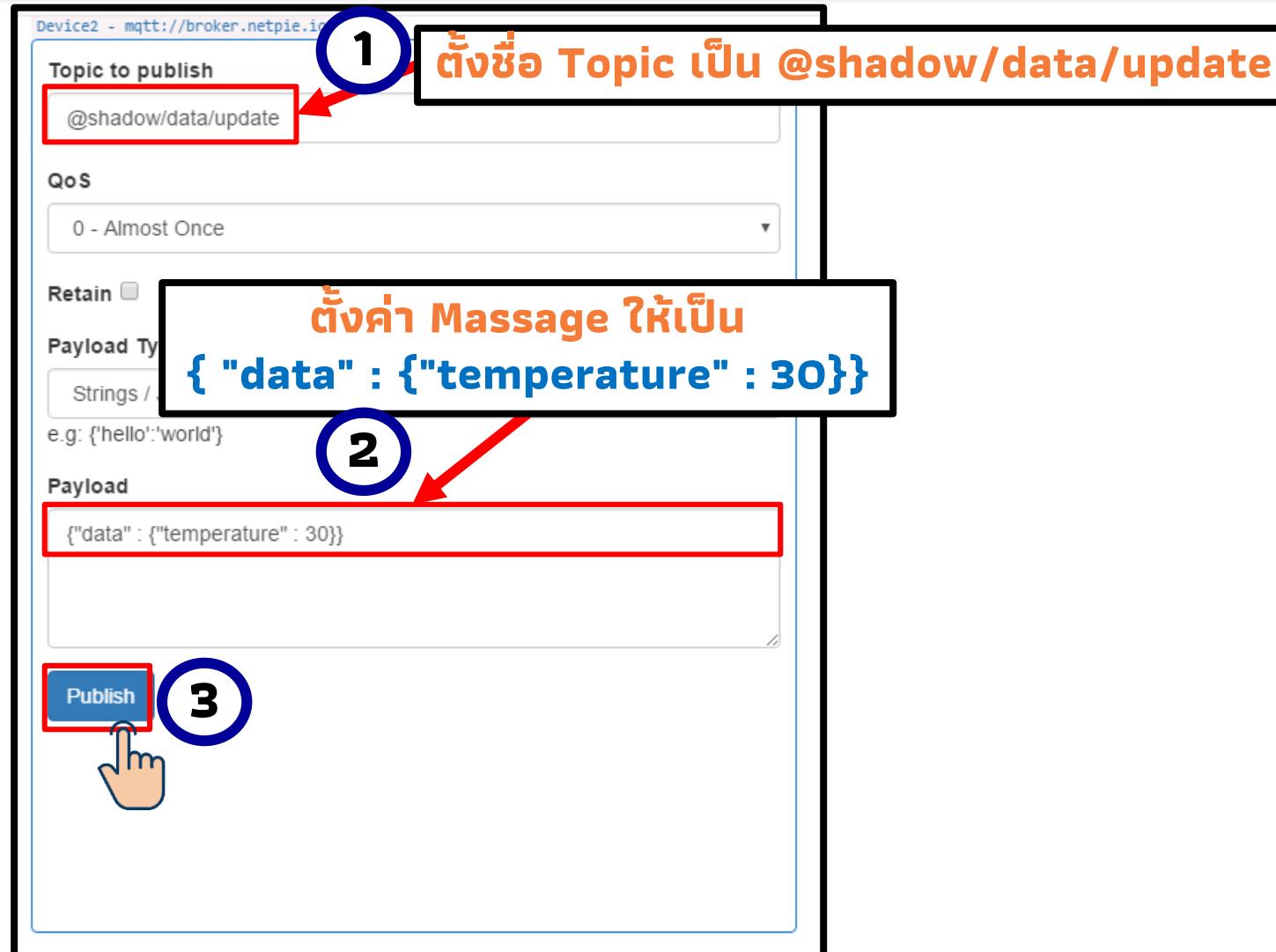
Save Cancel

Code

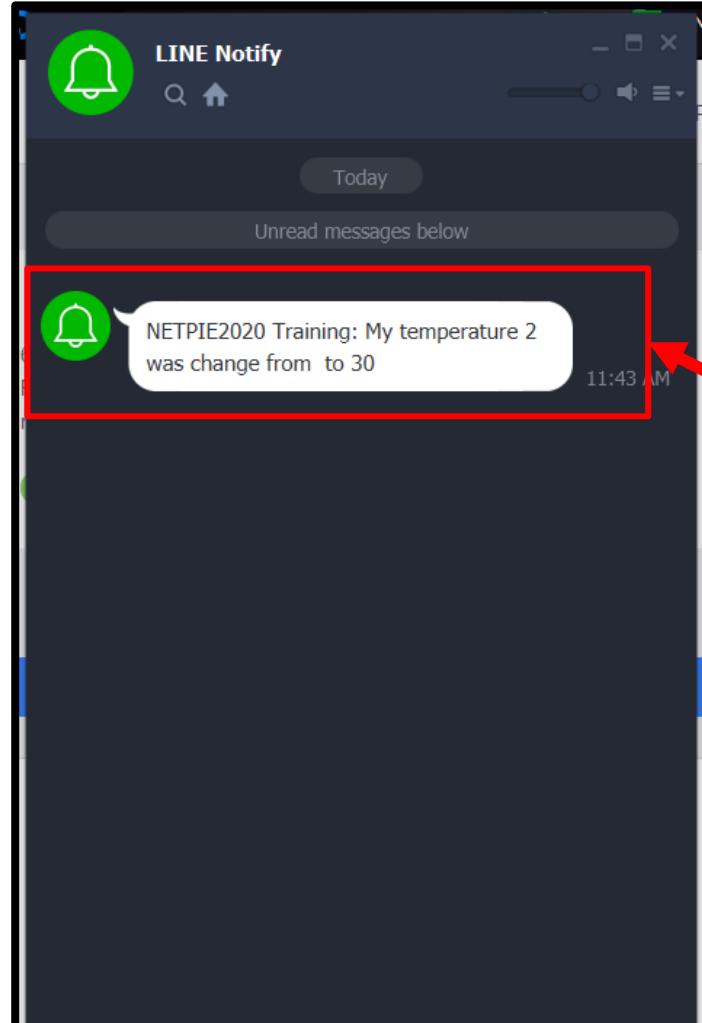
```
1  [
2   "enabled": true,
3   "trigger": [
4     {
5       "action": "LINE_NOTIFY",
6       "event": "SHADOW.UPDATED",
7       "condition": "$.temperature!= $$.temperature",
8       "msg": "My temperature 2 was change from {$$.temperature} to {{$temperature}}",
9       "option": {
10         "linetoken": "Your_Token_LINE"
11       }
12     },
13     {
14       "action": "LINE_NOTIFY",
15       "event": "DEVICE.STATUSCHANGED",
16       "msg": "My Device {{$statustext}}, statuscode: {{$status}}",
17       "option": {
18         "linetoken": "Your_Token_LINE"
19       }
20     }
21   ]
22 }
```

Ln: 22 Col: 2

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



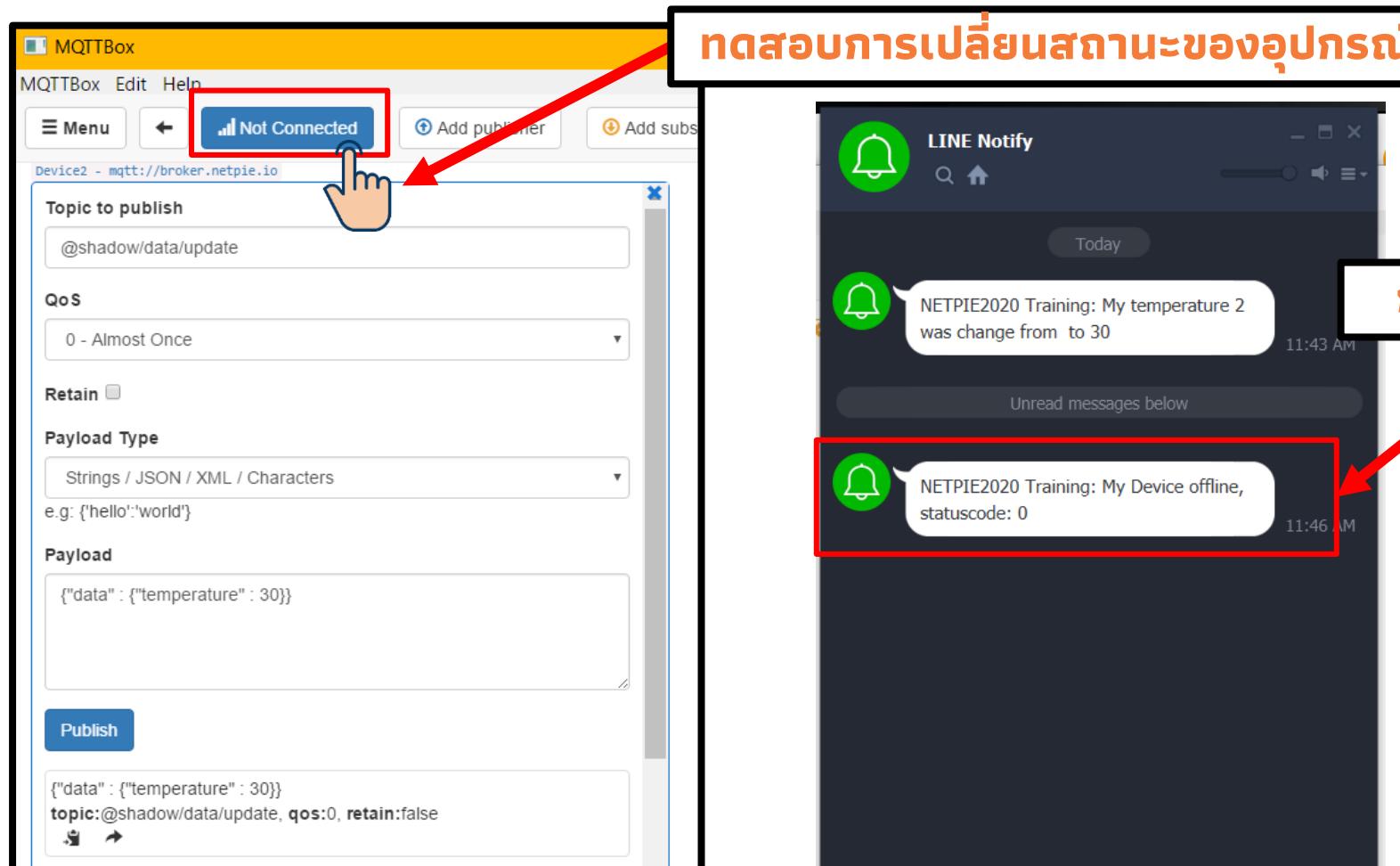
Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



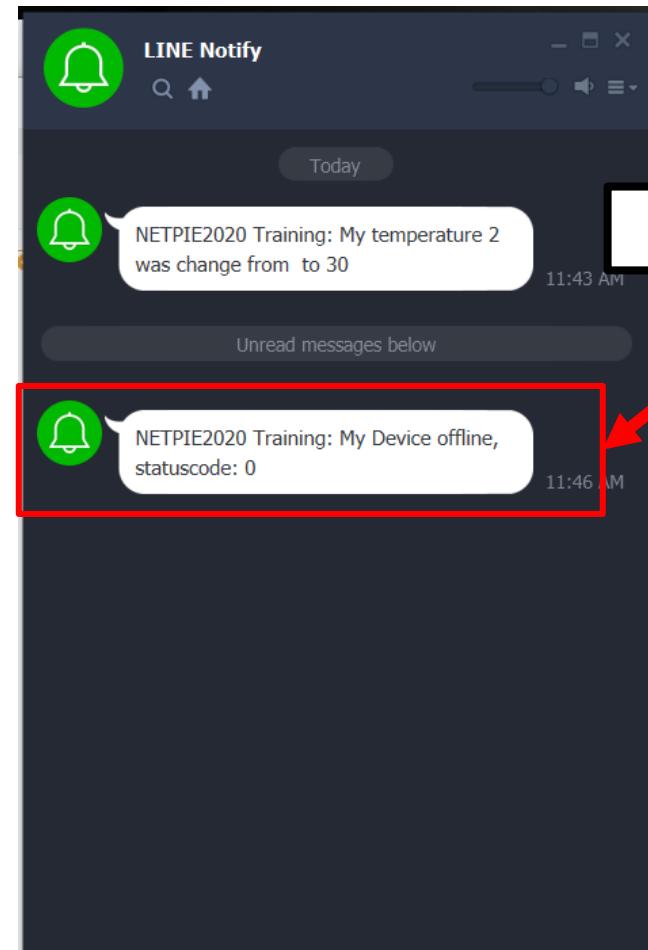
การแจ้งเตือนการเปลี่ยนอุณหภูมิจาก LINE

Module 3 : Device Data Management

Workshop 5 : การแจ้งเตือนการเปลี่ยนแปลงอุณหภูมิ และสถานะ Device



ทดสอบการเปลี่ยนสถานะของอุปกรณ์



การแจ้งเตือนการเปลี่ยนสถานะจาก LINE



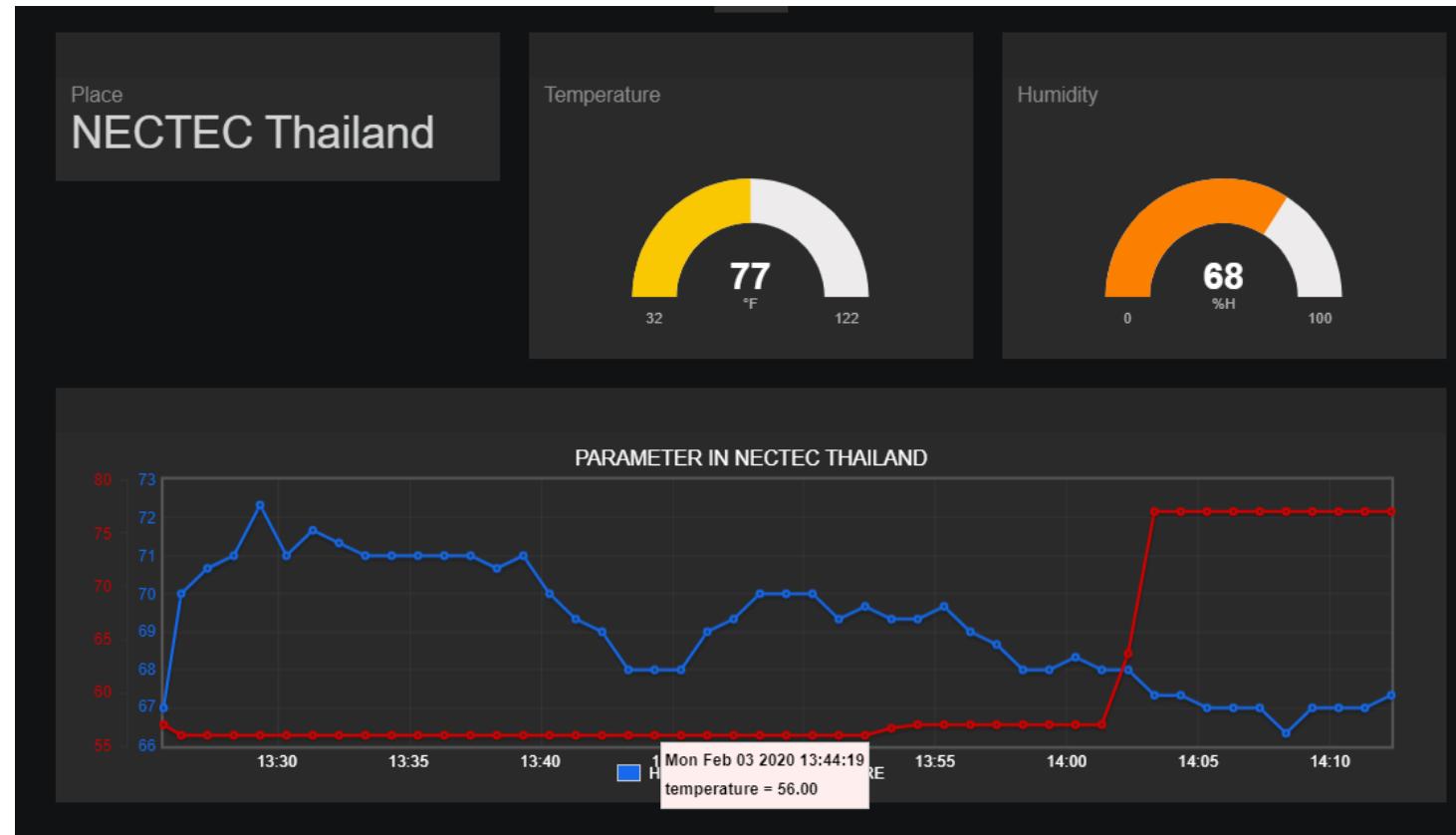
Module 4

NETPIE 2020 Freeboard



Module 4 : NETPIE 2020 Freeboard

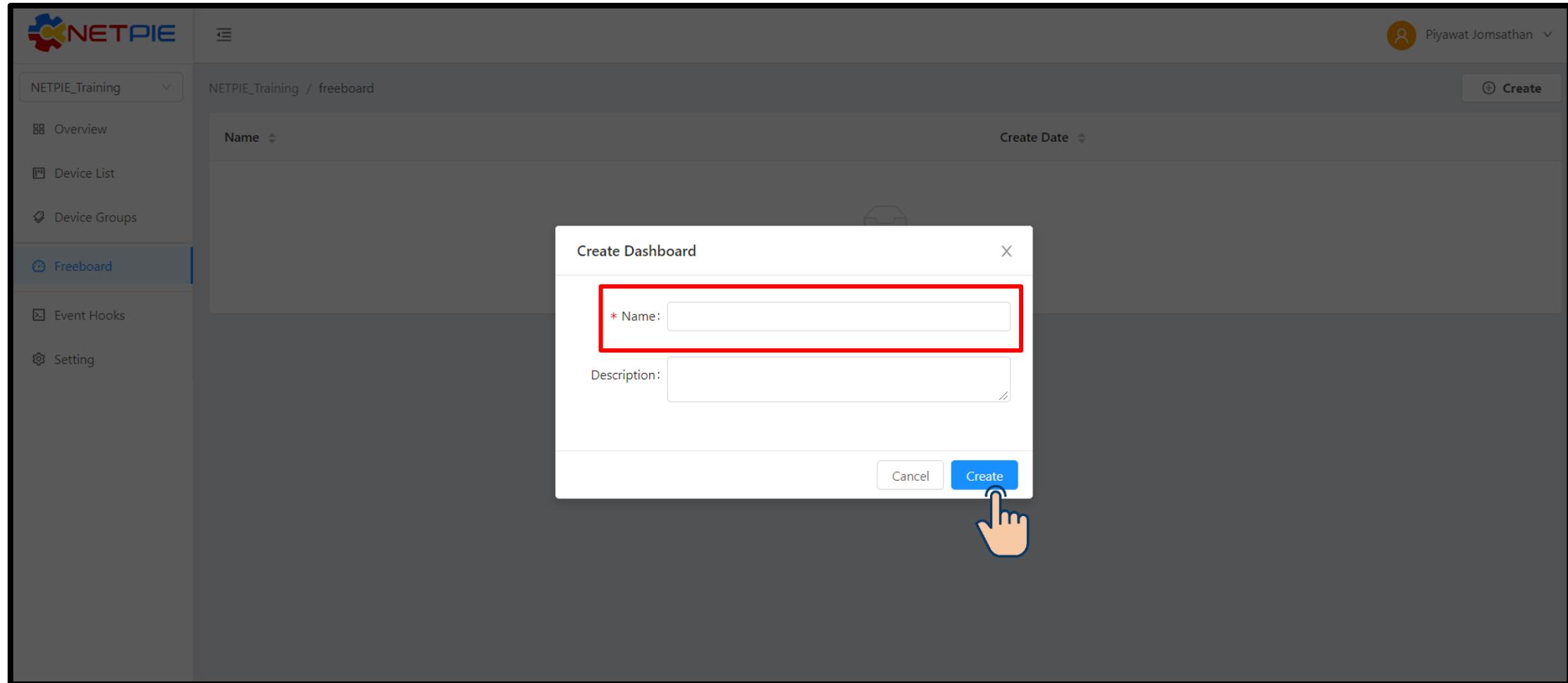
What is Freeboard?



การสร้าง Freeboard

The screenshot shows the NETPIE Freeboard interface. On the left, a sidebar menu includes 'NETPIE_Training' (selected), 'Overview', 'Device List', 'Device Groups', 'Freeboard' (highlighted with a red box), 'Event Hooks', and 'Setting'. The main area displays a table with columns 'Name' and 'Create Date', showing one entry: 'No Data'. In the top right corner, there is a 'Create' button with a plus sign and the word 'Create'. A hand cursor is pointing at this 'Create' button, which is also highlighted with a red box.

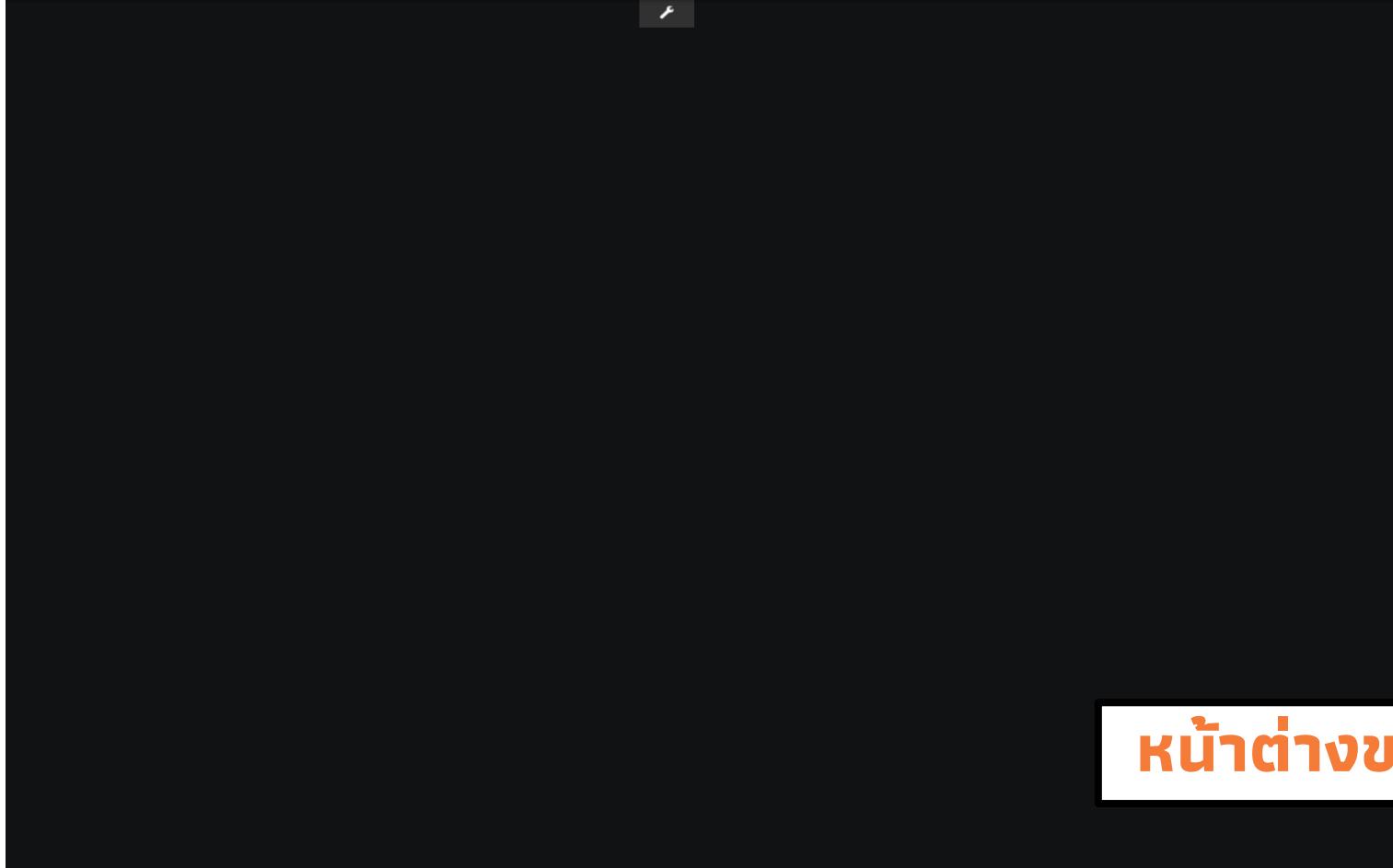
การสร้าง Freeboard



ការសរោះ Freeboard

The screenshot shows the NETPIE web interface for managing Freeboards. On the left, a sidebar menu includes 'NETPIE_Training' (selected), 'Overview', 'Device List', 'Device Groups', 'Freeboard' (selected), 'Event Hooks', and 'Setting'. The main content area displays a table titled 'NETPIE_Training / freeboard' with two columns: 'Name' and 'Create Date'. A single row is present, showing 'Freeboard1' and '2020-05-09 12:43'. A red box highlights the entire row, and a hand cursor is positioned over the 'Freeboard1' entry. At the bottom right of the table, there are pagination controls: '1-1 of 1 items', page number '1', and '10 / page'.

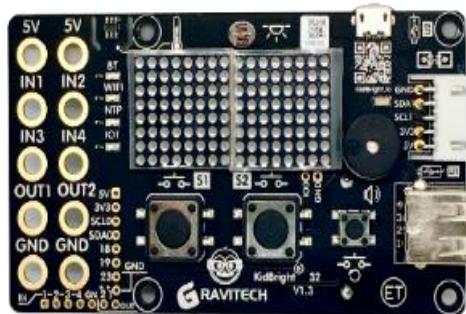
การสร้าง Freeboard



หน้าต่างของ Freeboard

Module 4 : NETPIE 2020 Freeboard

Workshop 6 : การสร้าง Freeboard uu NETPIE2020



KidBright

@shadow/data/update

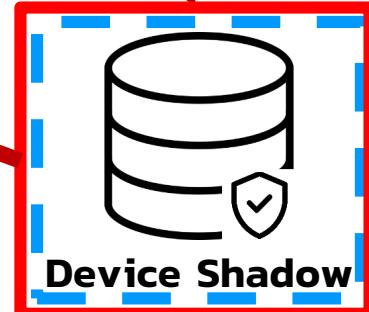
Publish : { "data": { "Temp" : 24, "light" : 80, "place" : "NECTEC" }}



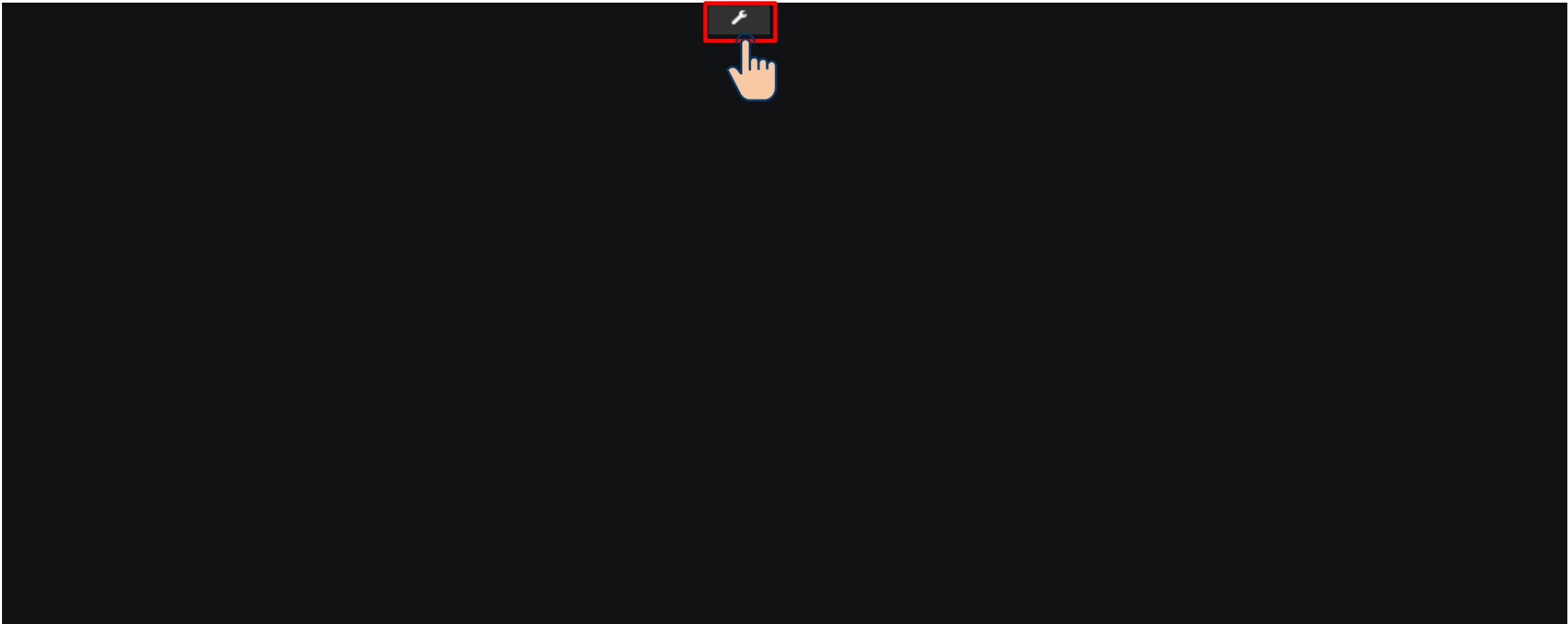
NETPIE2020



"Temp" : 91
"light" : 80

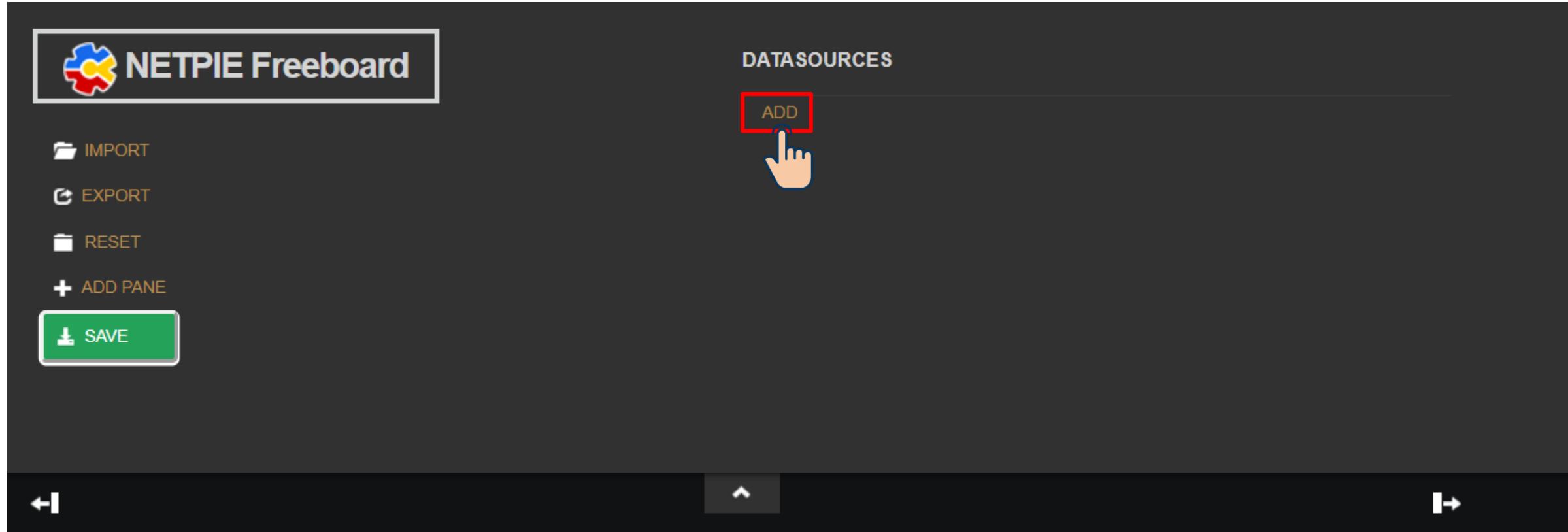


Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Module 4 : NETPIE 2020 Freeboard

Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020

The screenshot shows the configuration interface for a new Data Source in the NETPIE Freeboard. The interface is dark-themed with red highlights on input fields.

DATA SOURCE

- NAME:** [Input field highlighted with a red border]
- DEVICE ID:** [Input field highlighted with a red border]
Client ID នៃ Device ដែលត្រូវទិន្នន័យ
- DEVICE TOKEN:** [Input field highlighted with a red border]
Token នៃ Device ដែលត្រូវទិន្នន័យ

SUBSCRIBED TOPICS: [Input field]
Topic ដែលត្រូវទិន្នន័យ

FEED: [Input field with a checked checkbox labeled "NO"]
[Text label: เปิดการใช้งาน FEED]

SINCE: [Input field with value "6"]
Hour
Display data points since ... ago.

DOWN SAMPLING: [Input field with value "1"]
Minute
Resolution of the data points.

Buttons: SAVE, CANCEL

Module 4 : NETPIE 2020 Freeboard

Workshop 6 : การสร้าง Freeboard uu NETPIE2020

DATASOURCE

NAME **ชื่อ Datasource**

DEVICE ID **Client ID ของ Device**
Client ID នៃកាមុននៃការទូទាត់ទូទាត់ពីកាមុន

DEVICE TOKEN **Token នៃកាមុន**
Token នៃកាមុនដើម្បីទូទាត់ទូទាត់ពីកាមុន

SUBSCRIBED TOPICS
Topic ដើម្បីទូទាត់ទូទាត់

FEED **ເປីកការប្រើប្រាស់ FEED**

SINCE **Hour**
Display data points since ... ago.

DOWN SAMPLING **Minute**
Resolution of the data points.

SAVE  **CANCEL**



Module 4 : NETPIE 2020 Freeboard



NECTEC
a member of NSTDA

Workshop 6 : การสร้าง Freeboard 用 NETPIE2020

The screenshot shows the NETPIE Freeboard application interface. On the left, there is a sidebar with icons for IMPORT, EXPORT, RESET, ADD PANE, and a large green SAVE button. The main area is titled 'DATASOURCES' and lists a single entry: 'NETPIE_Training' last updated at '8:41:38 PM'. To the right of this entry are edit and delete icons. A red arrow points from a purple text box at the bottom right towards the 'NETPIE_Training' entry. The purple text box contains the Thai text: 'เมื่อเชื่อมต่อสำเร็จจะขึ้นสถานะดังรูป'.

NETPIE Freeboard

IMPORT

EXPORT

RESET

+ ADD PANE

SAVE

DATASOURCES

Name Last Updated

NETPIE_Training 8:41:38 PM

ADD

เมื่อเชื่อมต่อสำเร็จจะขึ้นสถานะดังรูป

Workshop 6 : การสร้าง Freeboard uu NETPIE2020

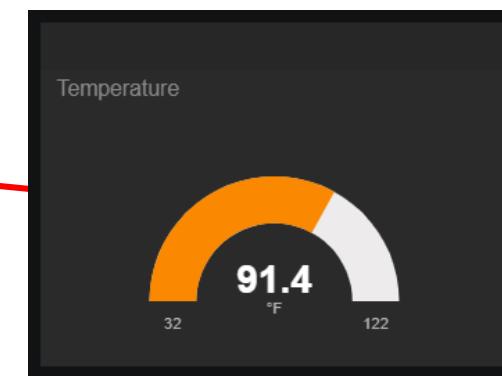
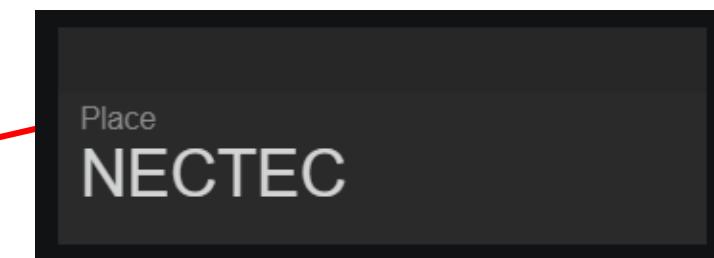
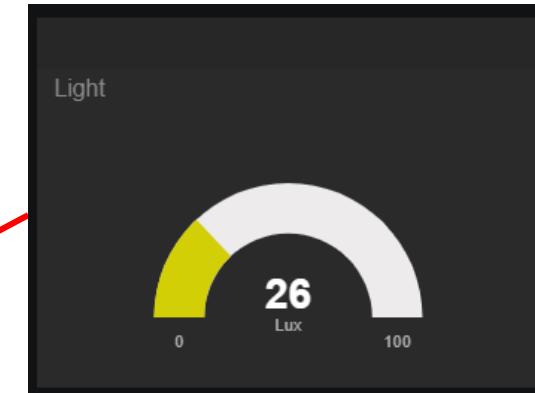
Data ក្នុងប័ណ្ណការងារ Device Shadow

Shadow Schema Trigger

Tree ▾

Select a node...

- object {3}
 - light : 29
 - place : NECTEC
 - temperature : 93.2



Module 4 : NETPIE 2020 Freeboard



NECTEC
a member of NSTDA

Workshop 6 : การสร้าง Freeboard บน NETPIE2020

The screenshot shows the NETPIE Freeboard interface. On the left, there is a sidebar with icons for IMPORT, EXPORT, and RESET, and a large green 'SAVE' button with a hand cursor icon. A red box highlights the '+ ADD PANE' button. The main area is titled 'DATASOURCES' and lists a single entry: 'NETPIE_Training' last updated at 8:41:38 PM, with edit and delete icons next to it. There is also an 'ADD' button.

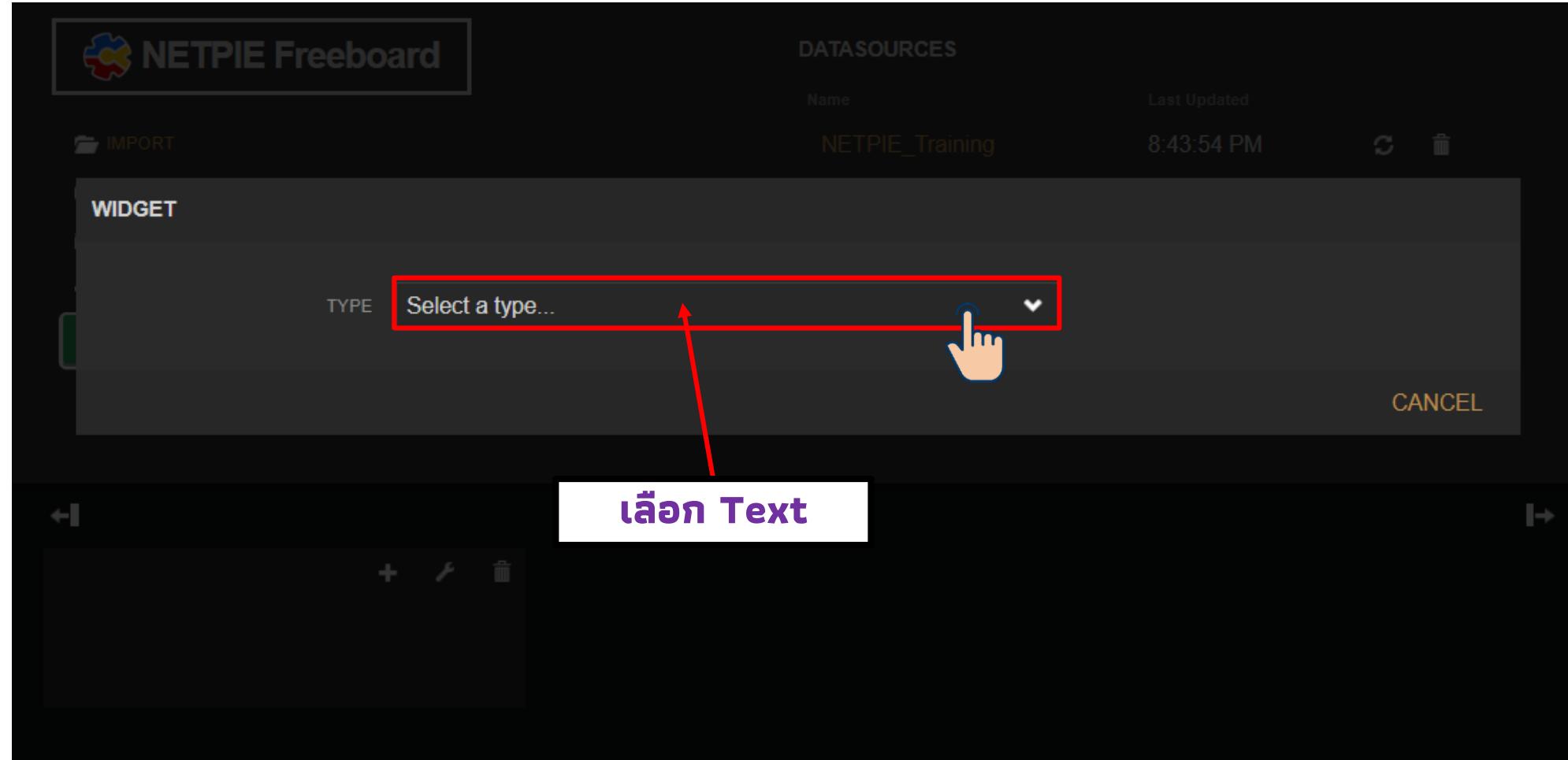
Module 4 : NETPIE 2020 Freeboard



Workshop 6 : การสร้าง Freeboard บน NETPIE2020

The screenshot shows the NETPIE Freeboard interface. On the left, there's a sidebar with icons for IMPORT, EXPORT, RESET, ADD PANE, and a large green SAVE button. The main area is titled "DATASOURCES" and lists a single entry: "NETPIE_Training" last updated at "8:43:26 PM". Below this is an "ADD" button. At the bottom, there's a dark panel with a red-bordered plus sign icon, which is being clicked by a hand cursor. Navigation arrows and a top bar are also visible.

Workshop 6 : การสร้าง Freeboard uu NETPIE2020

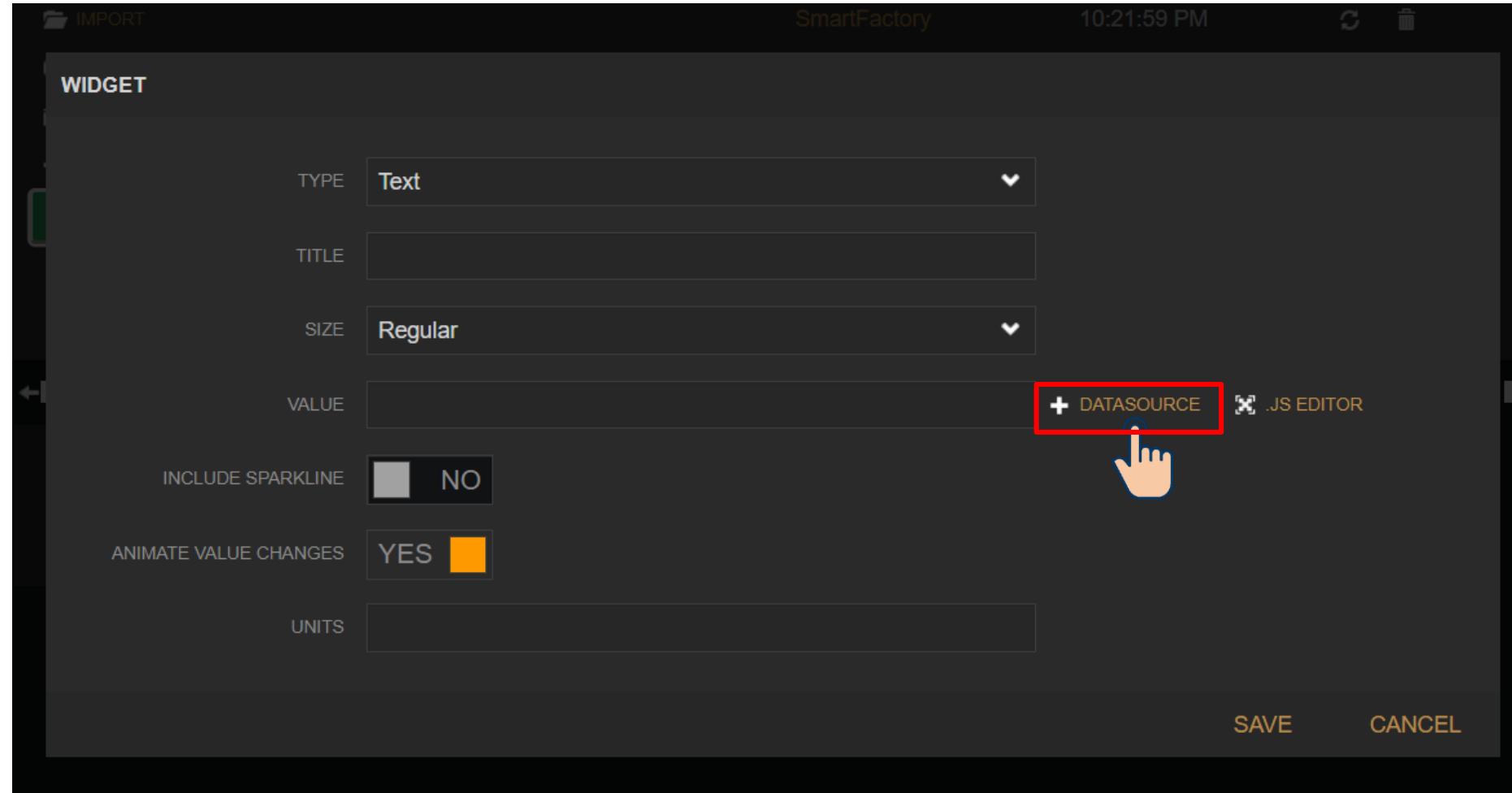


Module 4 : NETPIE 2020 Freeboard

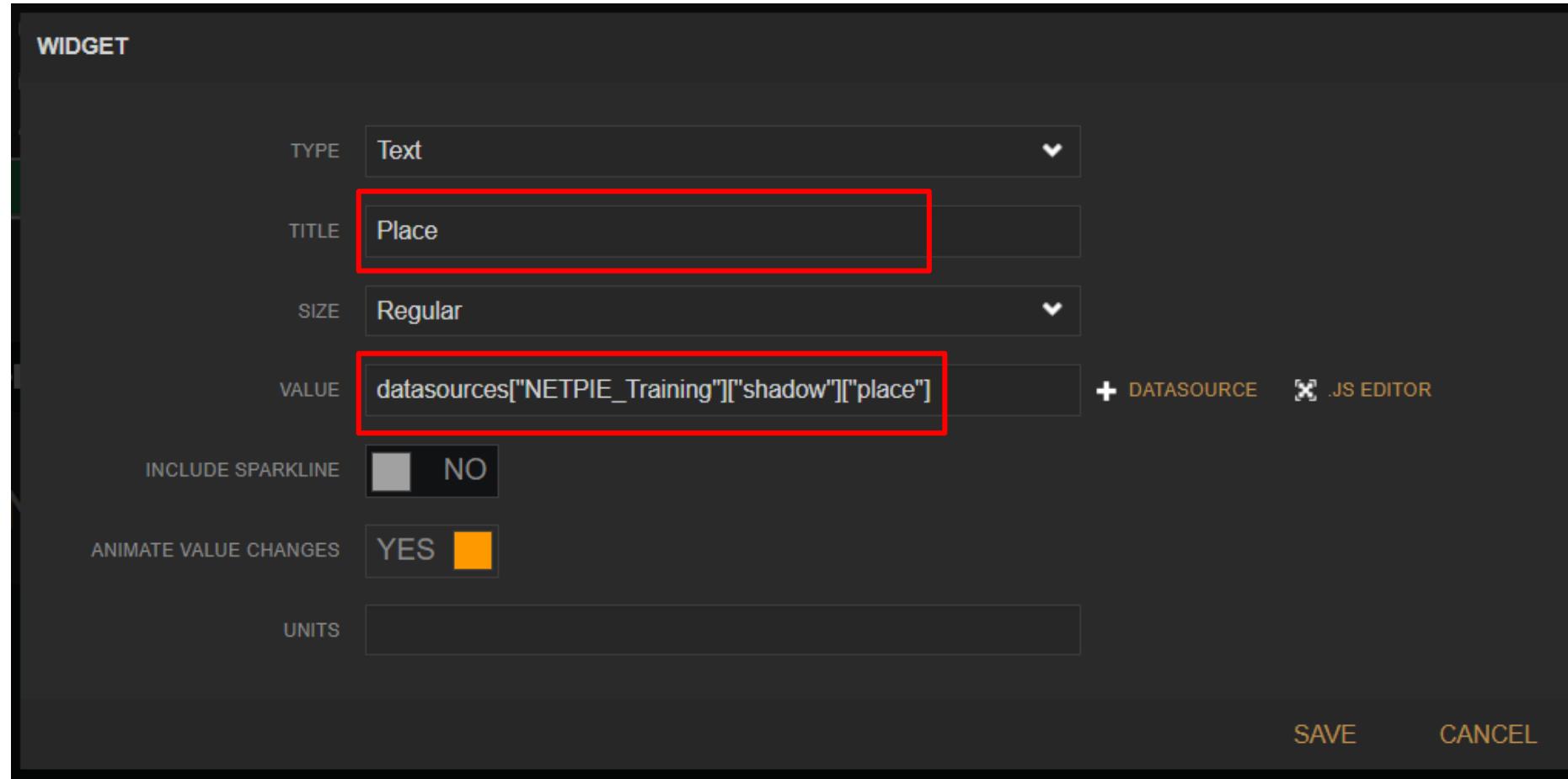


NECTEC
a member of NSTDA

Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard 用 NETPIE2020

The screenshot shows the NETPIE Freeboard application interface. On the left, there's a sidebar with icons for IMPORT, EXPORT, RESET, ADD PANE, and a large green SAVE button. The main area has a title bar "Widget สำหรับแสดงสถานที่จาก Shadow". Below it, a red box highlights a "Place" input field containing "NECTEC" and a row of three icons: a plus sign, a gear, and a trash can. A red arrow points from this highlighted area towards the bottom-left corner of the screen.

DATA SOURCES

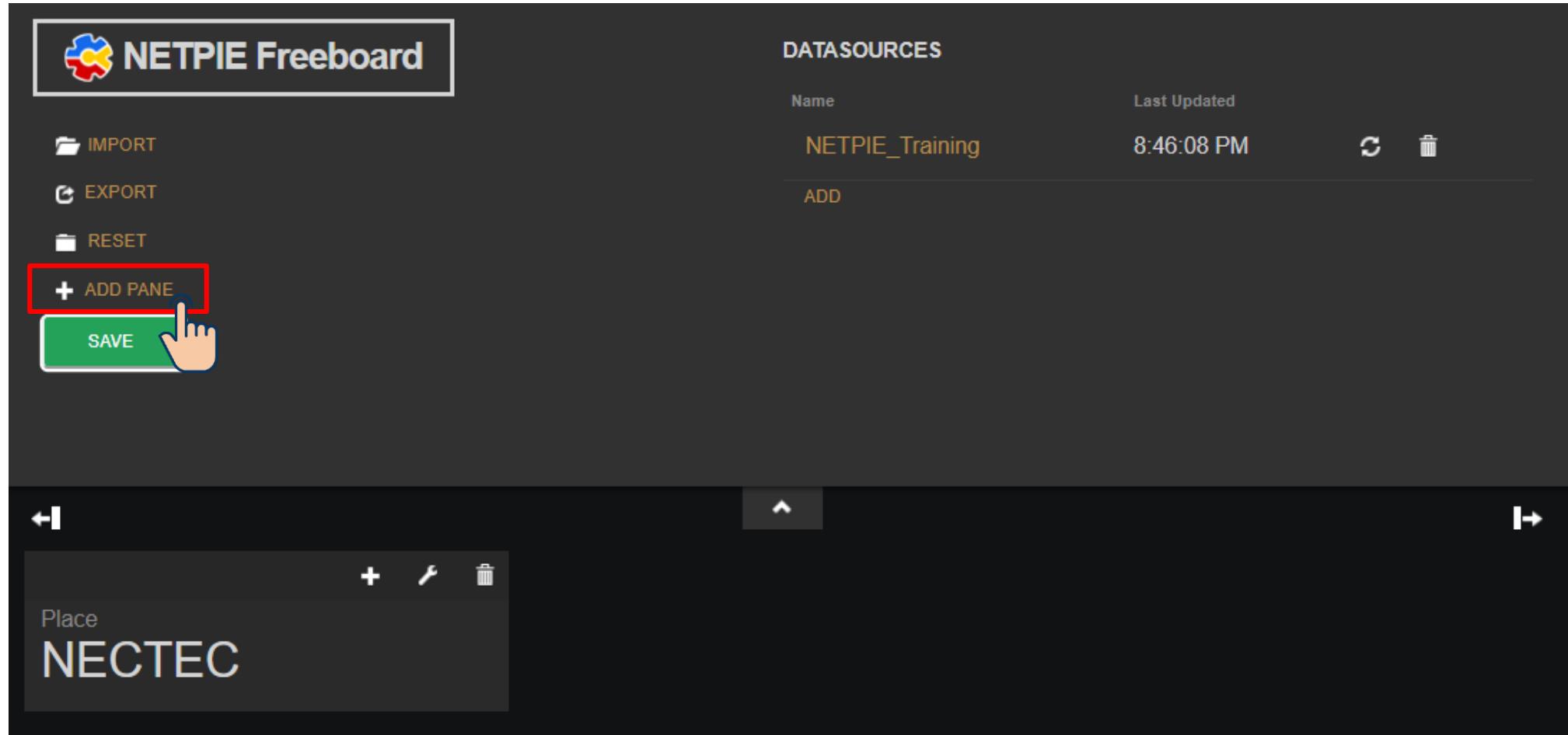
Name	Last Updated
NETPIE_Training	8:46:08 PM

ADD

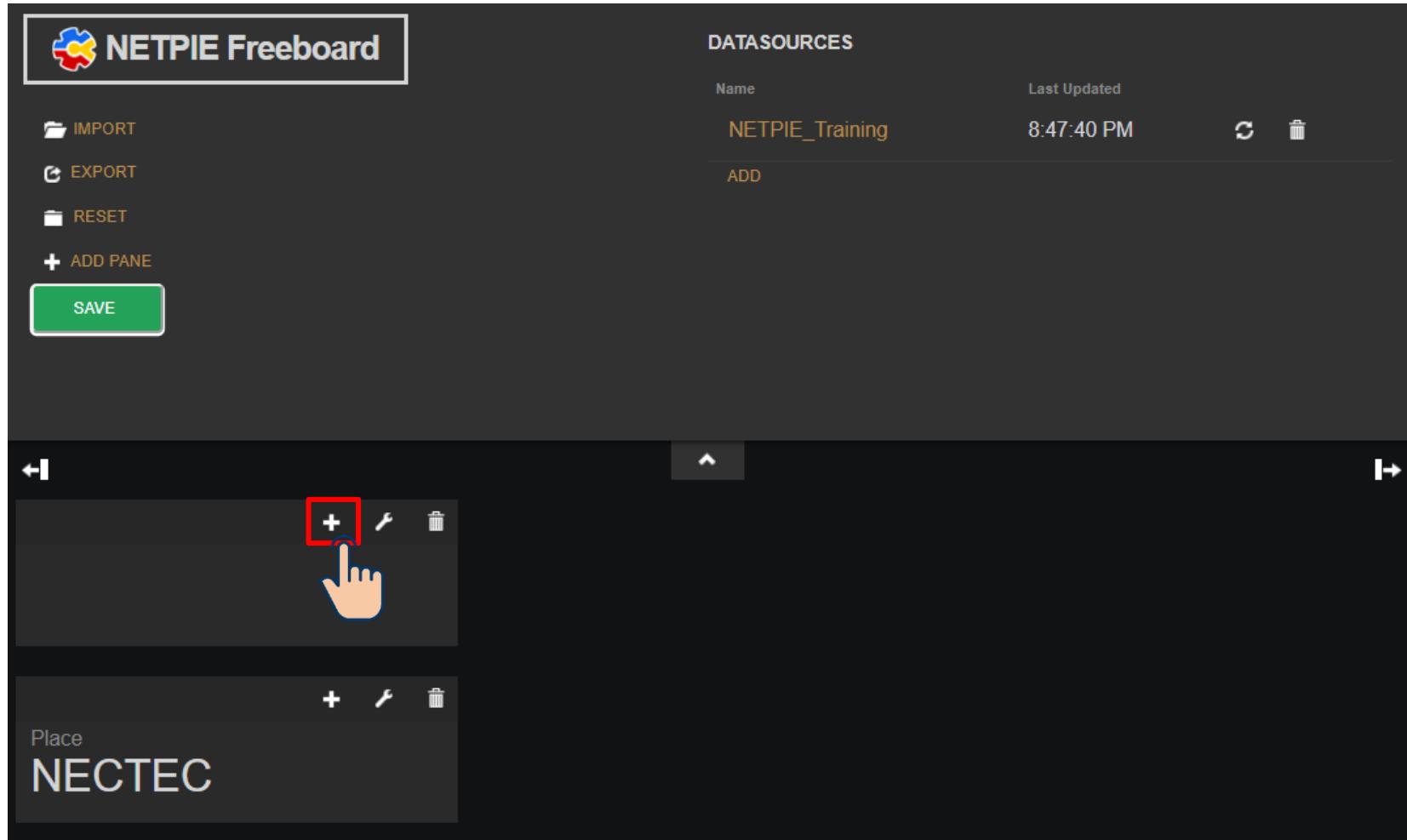
Widget สำหรับแสดงสถานที่จาก Shadow

Place NECTEC

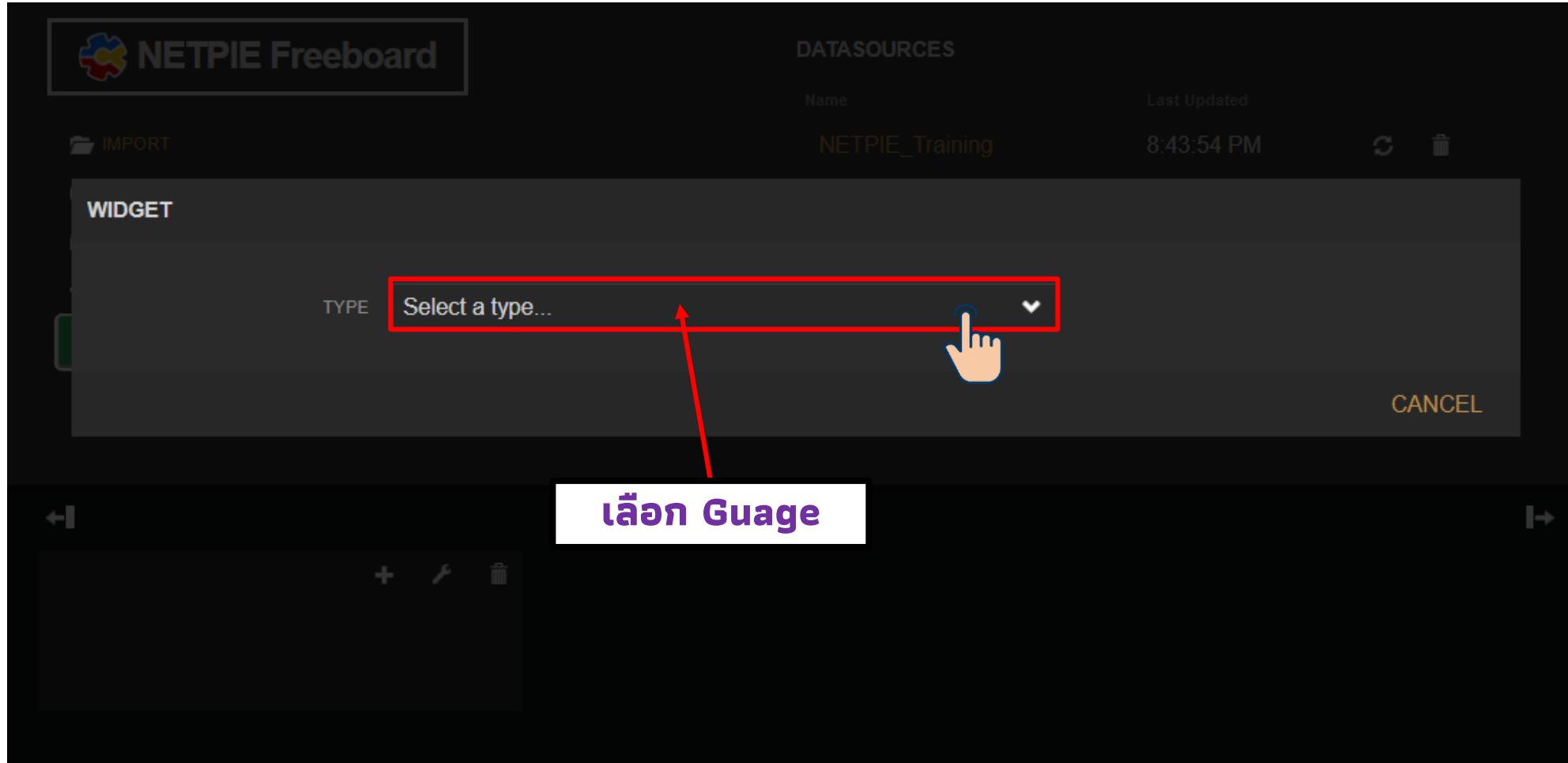
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



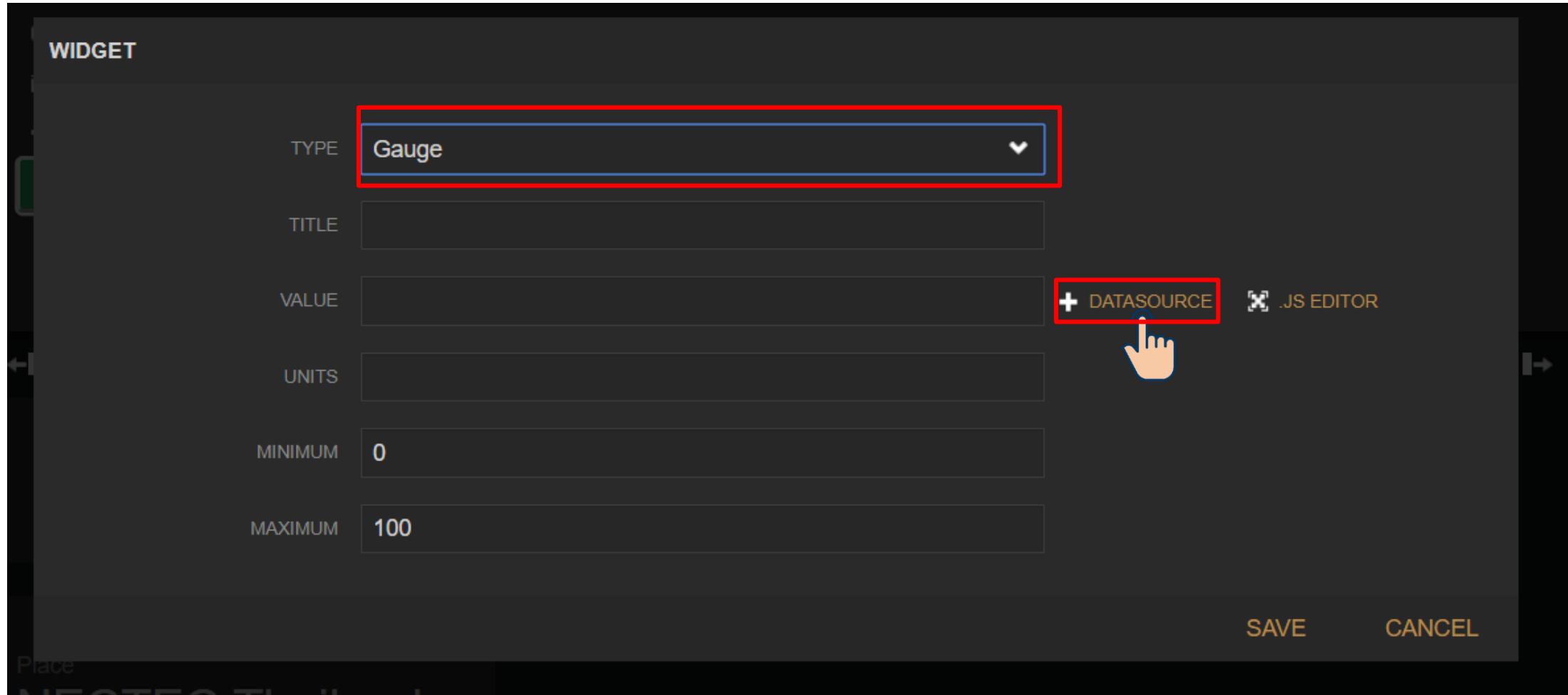
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Module 4 : NETPIE 2020 Freeboard



Workshop 6 : การสร้าง Freeboard บน NETPIE2020

WIDGET

TYPE Gauge

TITLE Temperature

VALUE `datasources["NETPIE_Training"]["shadow"]["temperature"]`

UNITS °F

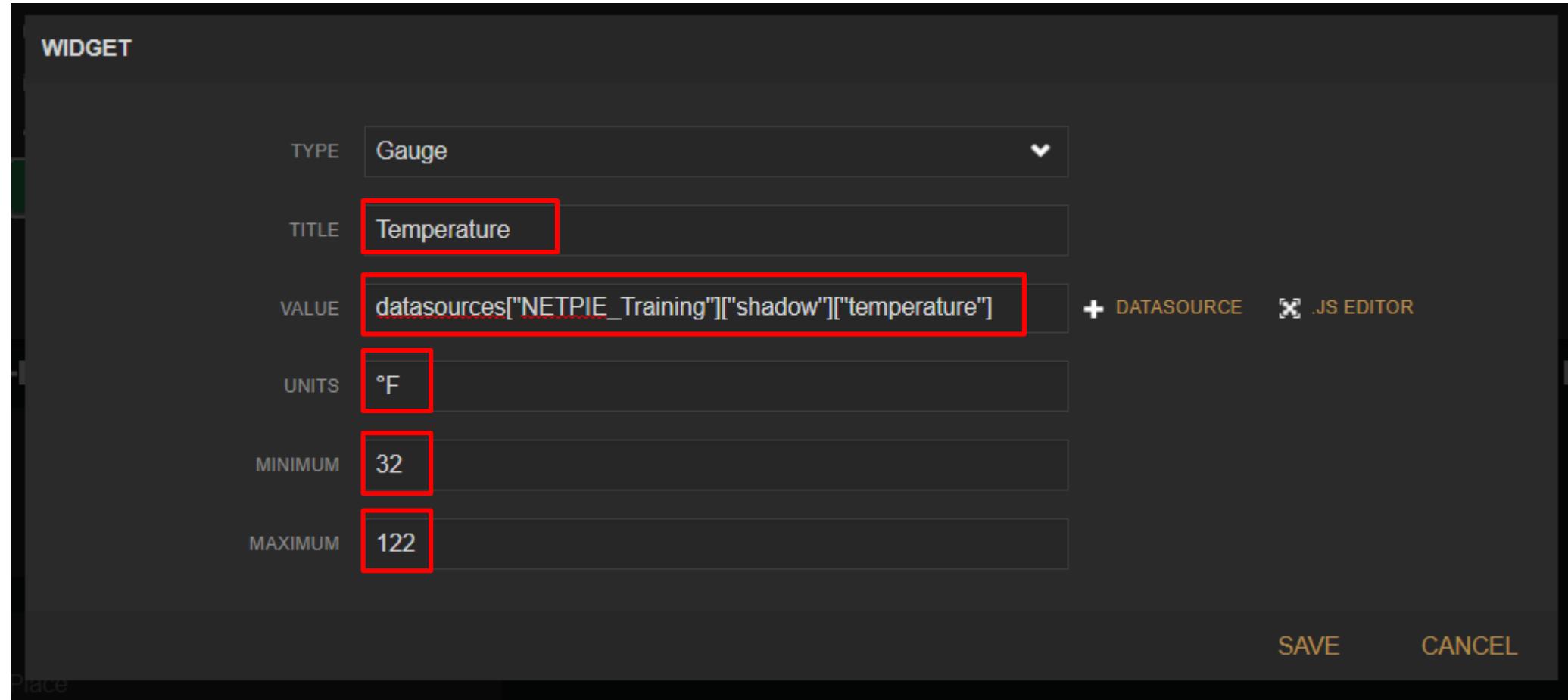
MINIMUM 32

MAXIMUM 122

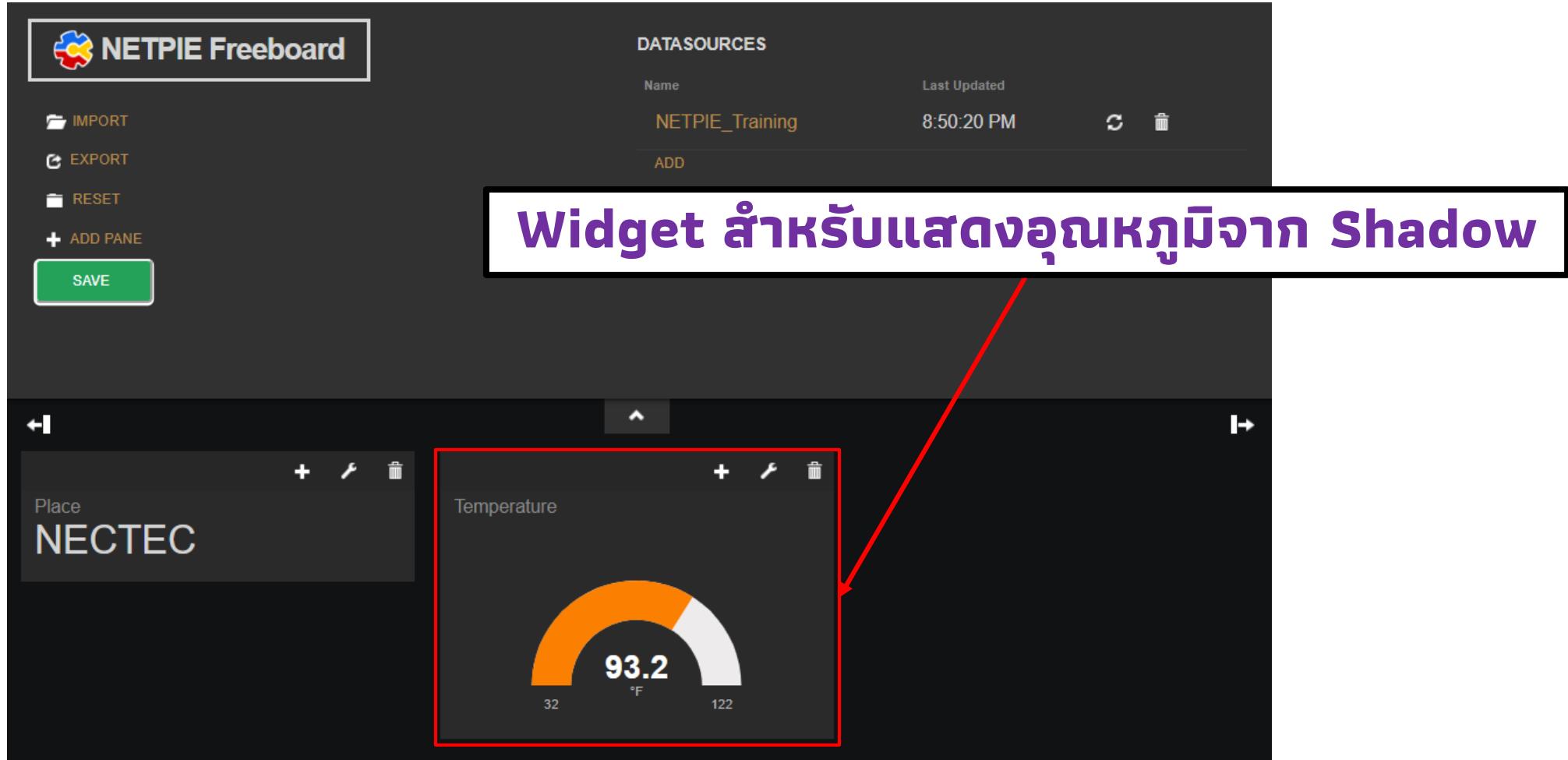
+ DATASOURCE X JS EDITOR

Place

SAVE CANCEL



Workshop 6 : การสร้าง Freeboard 用 NETPIE2020



Module 4 : NETPIE 2020 Freeboard



NECTEC
a member of NSTDA

Workshop 6 : การสร้าง Freeboard บน NETPIE2020

The screenshot shows the NETPIE Freeboard interface. On the left, there is a sidebar with options: IMPORT, EXPORT, RESET, ADD PANE (which is highlighted with a red box and a cursor pointing to it), and SAVE. Below the sidebar, there is a panel labeled 'Place' containing the text 'NECTEC'. On the right, the main area is titled 'DATASOURCES' and lists a single entry: 'NETPIE_Training' with a 'Last Updated' timestamp of '8:50:20 PM'. There are also 'EDIT' and 'DELETE' icons next to the entry. Below the DATASOURCES section, there is an 'ADD' button. At the bottom of the screen, there is a preview of a dashboard card titled 'Temperature' showing a gauge with the value '93.2 °F' and scale markers at 32 and 122.

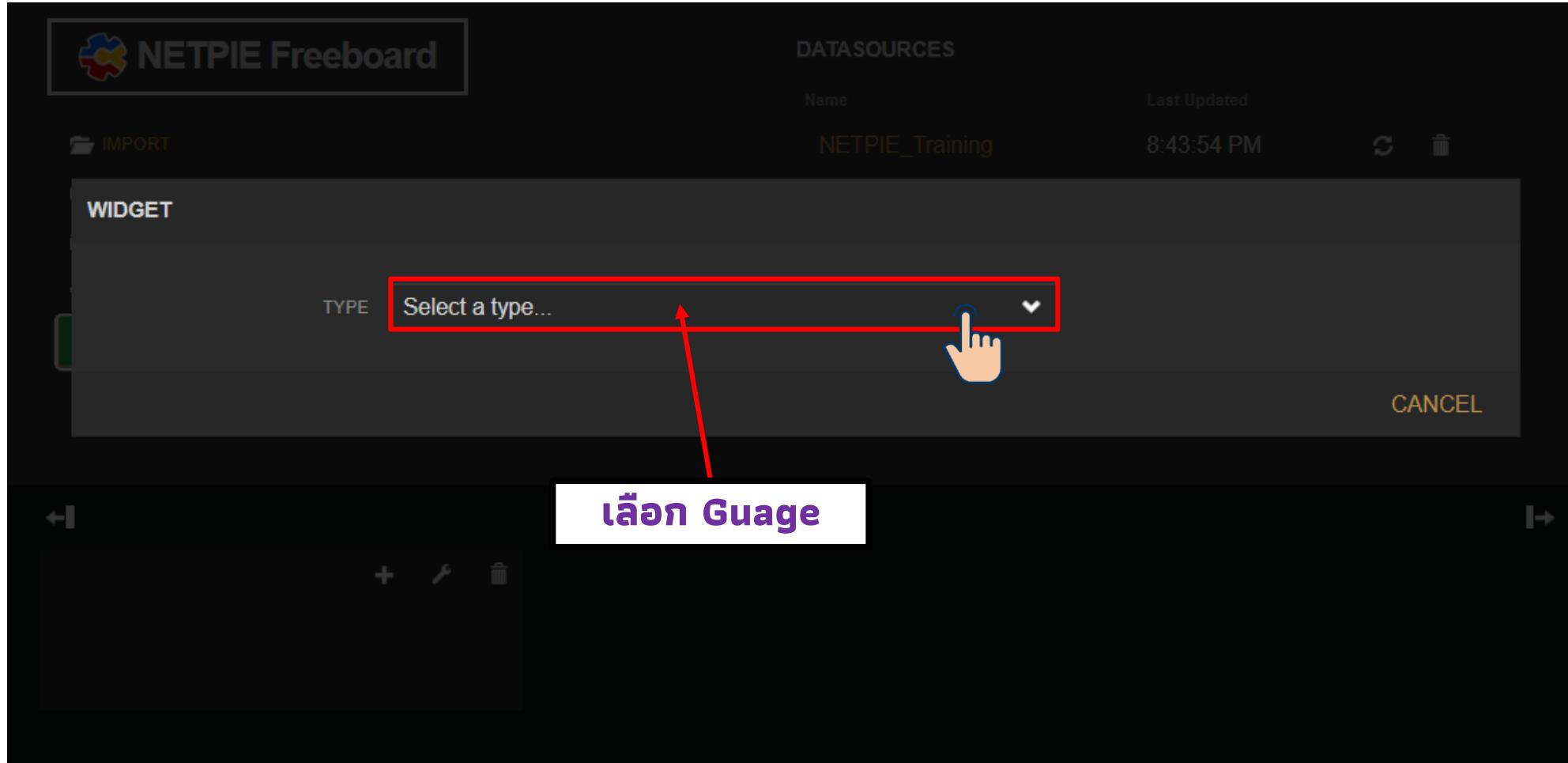
Module 4 : NETPIE 2020 Freeboard



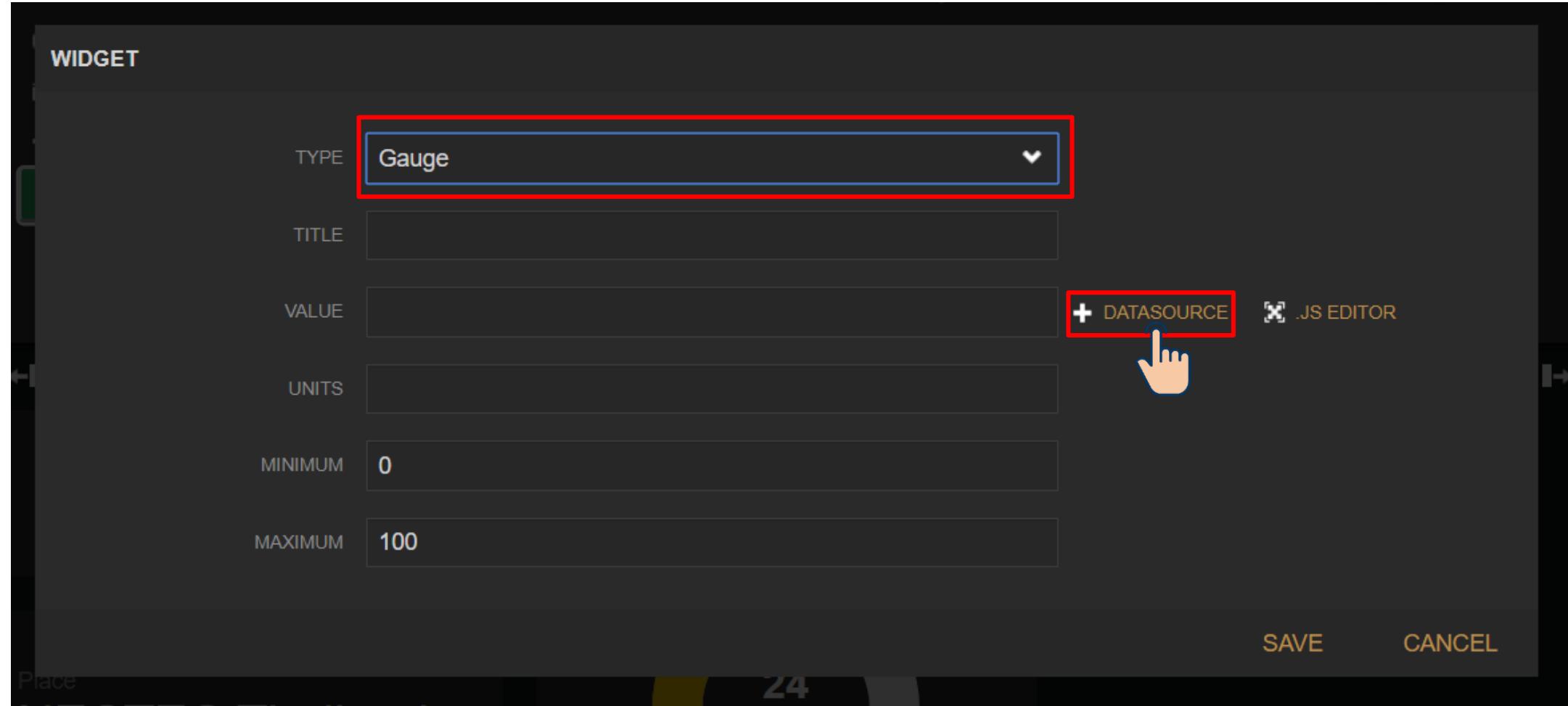
Workshop 6 : การสร้าง Freeboard บน NETPIE2020

The screenshot shows the NETPIE Freeboard interface. On the left, there's a sidebar with icons for IMPORT, EXPORT, RESET, ADD PANE, and a large green SAVE button. The main area is titled 'DATASOURCES' and lists a single entry: 'NETPIE_Training' last updated at 8:51:30 PM, with options to EDIT or REMOVE. Below this is an 'ADD' button. In the center, there's a gauge chart titled 'Temperature' with a value of 93.2 °F, ranging from 32 to 122. To the left of the gauge is a panel with a '+' icon highlighted with a red box and a hand cursor, and the text 'Place NECTEC' below it.

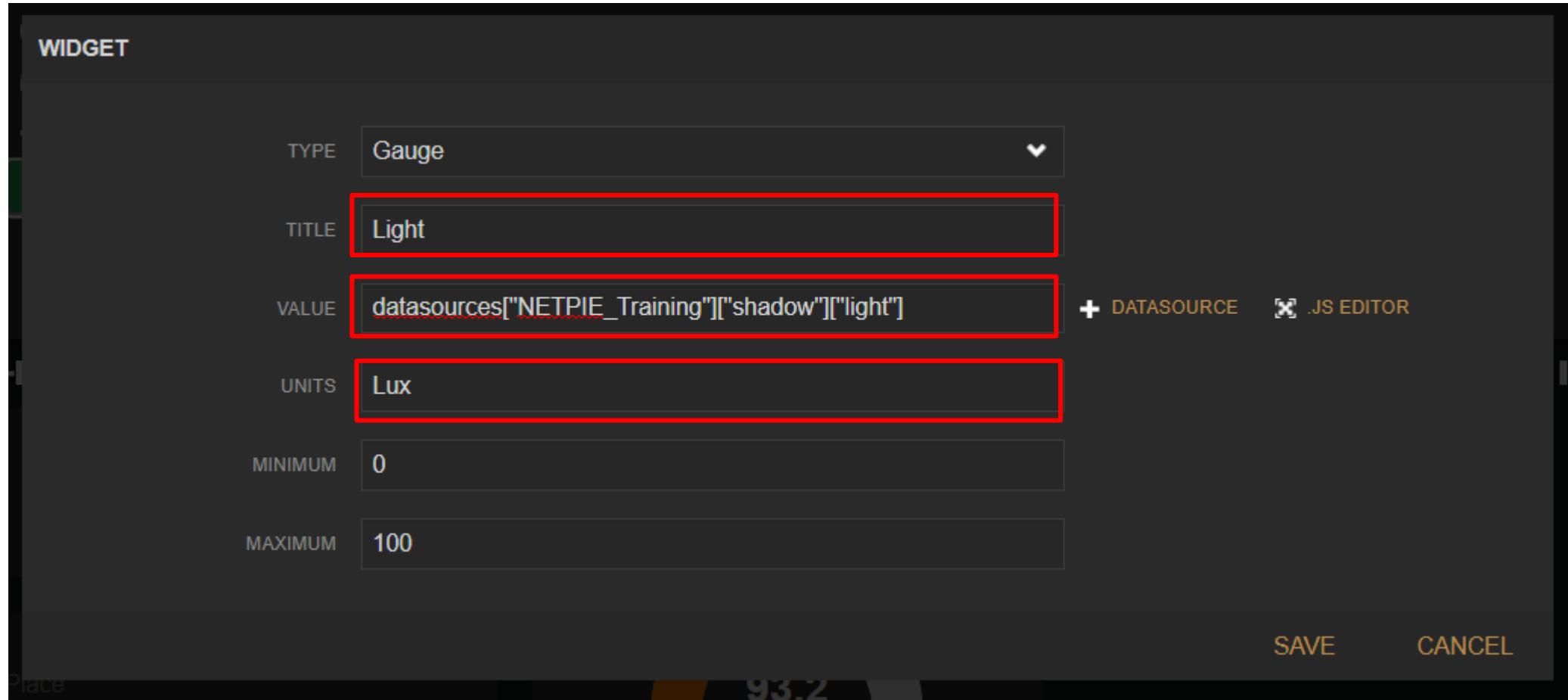
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



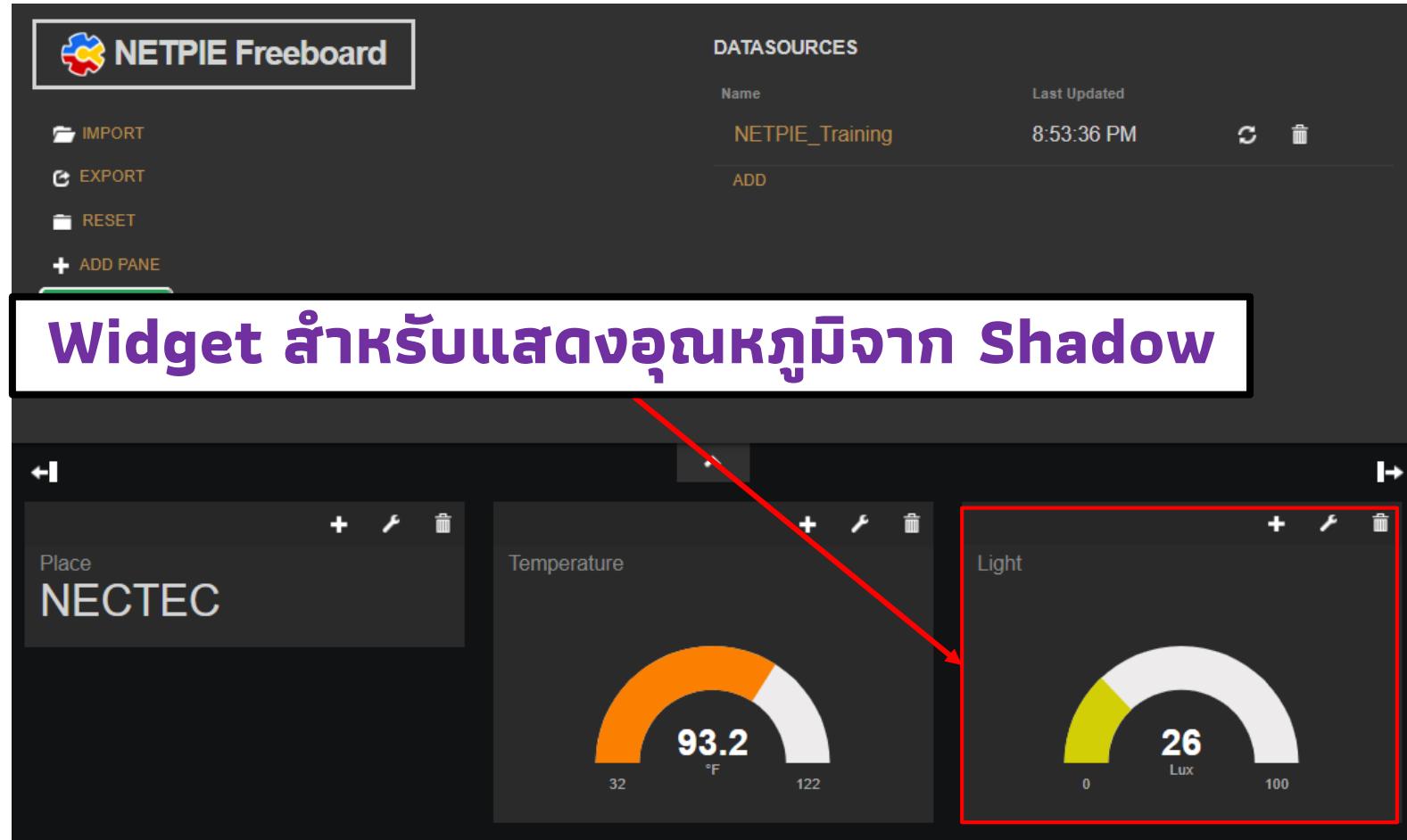
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020

The screenshot shows the NETPIE Freeboard application interface. On the left, there's a sidebar with icons for IMPORT, EXPORT, RESET, ADD PANE, and a large green SAVE button. The main area has a header titled "DATA SOURCES" with a single entry: "NETPIE_Training" last updated at "8:53:36 PM". Below this is an "ADD" button. A large central box contains the text "ต่อไปจะสร้าง FeedView ใช้สำหรับดูค่าของ อุณหภูมิและแสงเทียบกับเวลา" (Next will create FeedView to view temperature and light values). At the bottom, there are three circular gauge charts. The first chart, labeled "Place NECTEC", has a value of 93.2. The second chart, labeled "Temperature", has a value of 93.2 °F. The third chart, labeled "Light", has a value of 26 Lux.

ต่อไปจะสร้าง FeedView ใช้สำหรับดูค่าของ
อุณหภูมิและแสงเทียบกับเวลา

Place
NECTEC

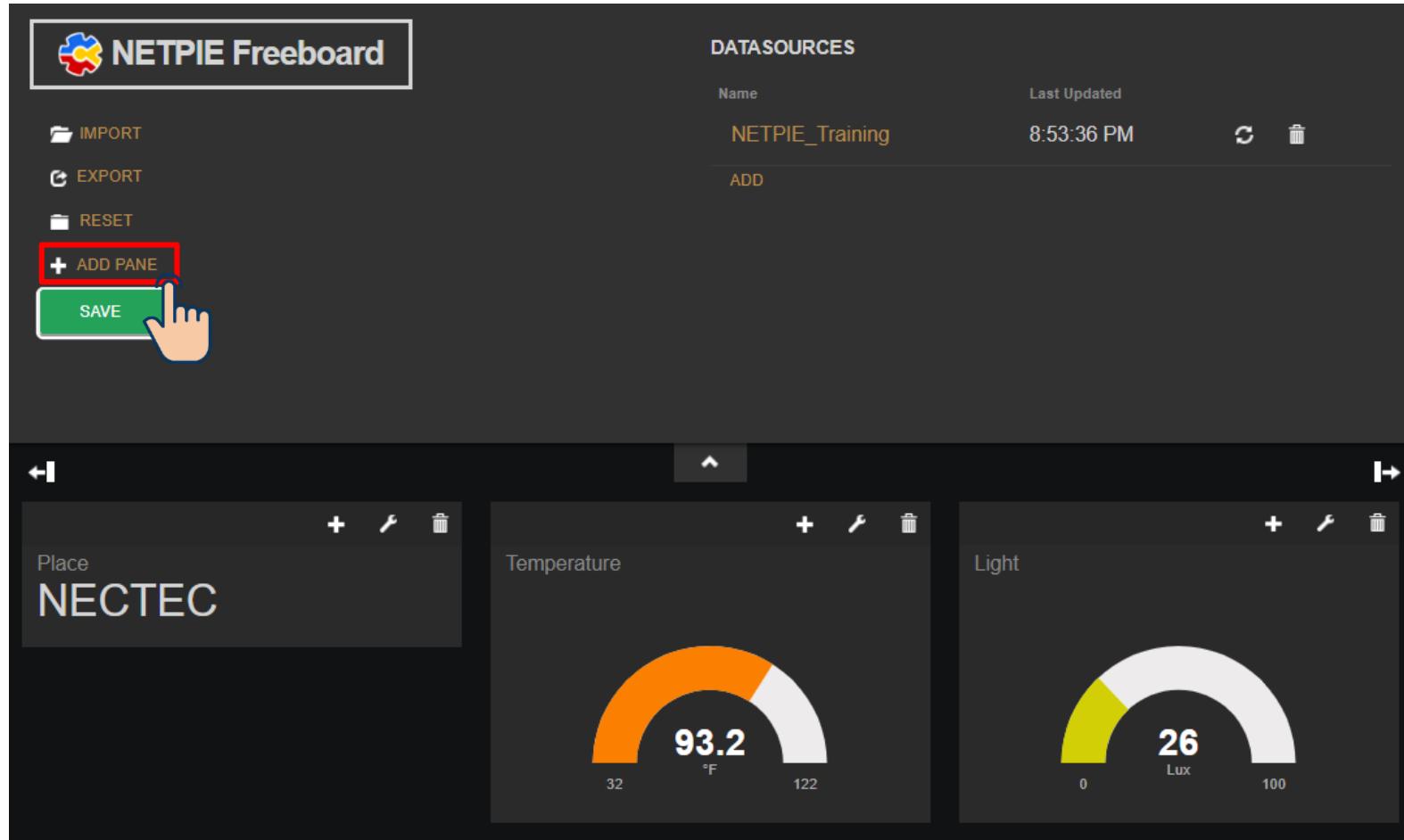
Temperature

Light

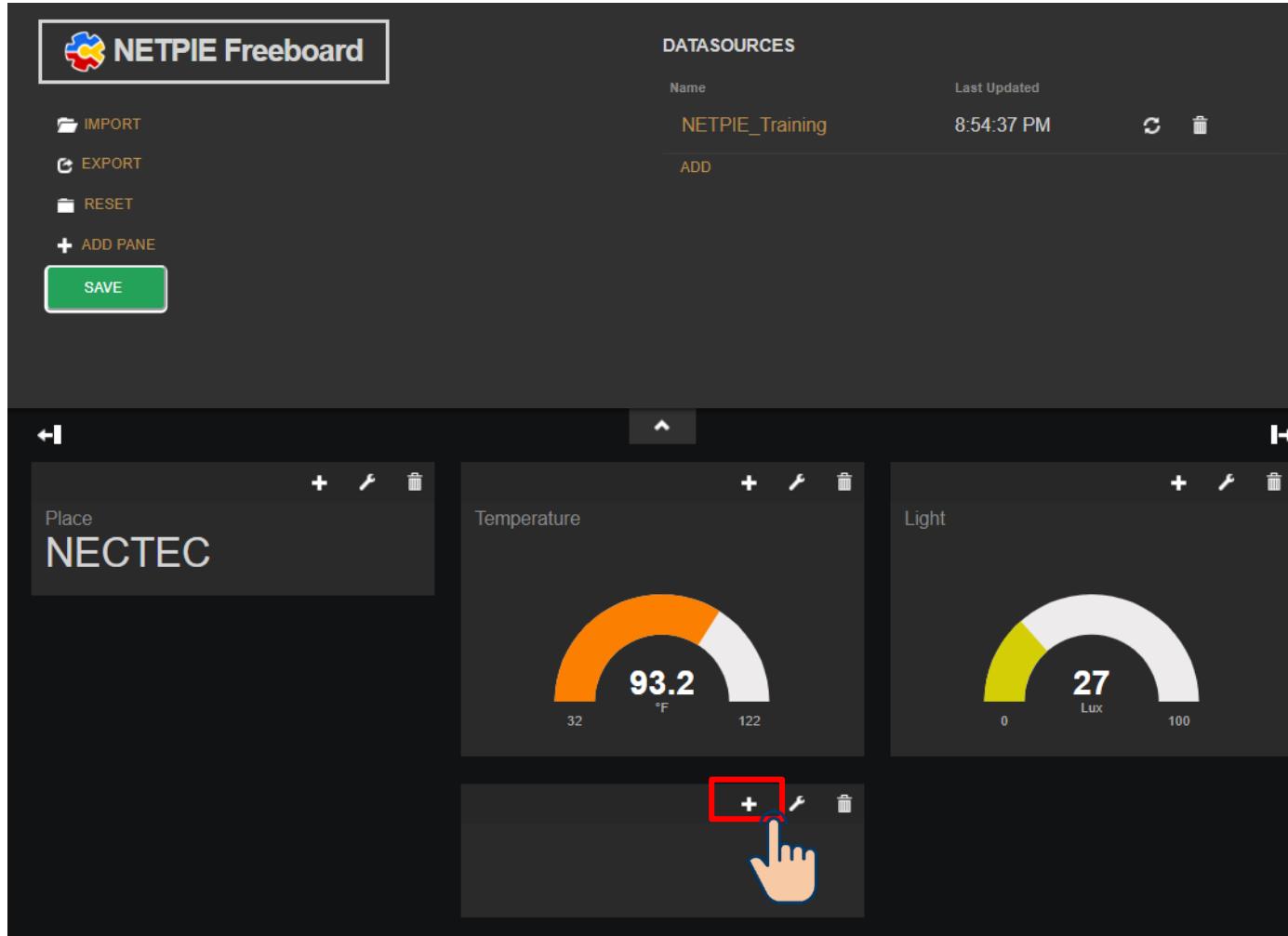
93.2 °F

26 Lux

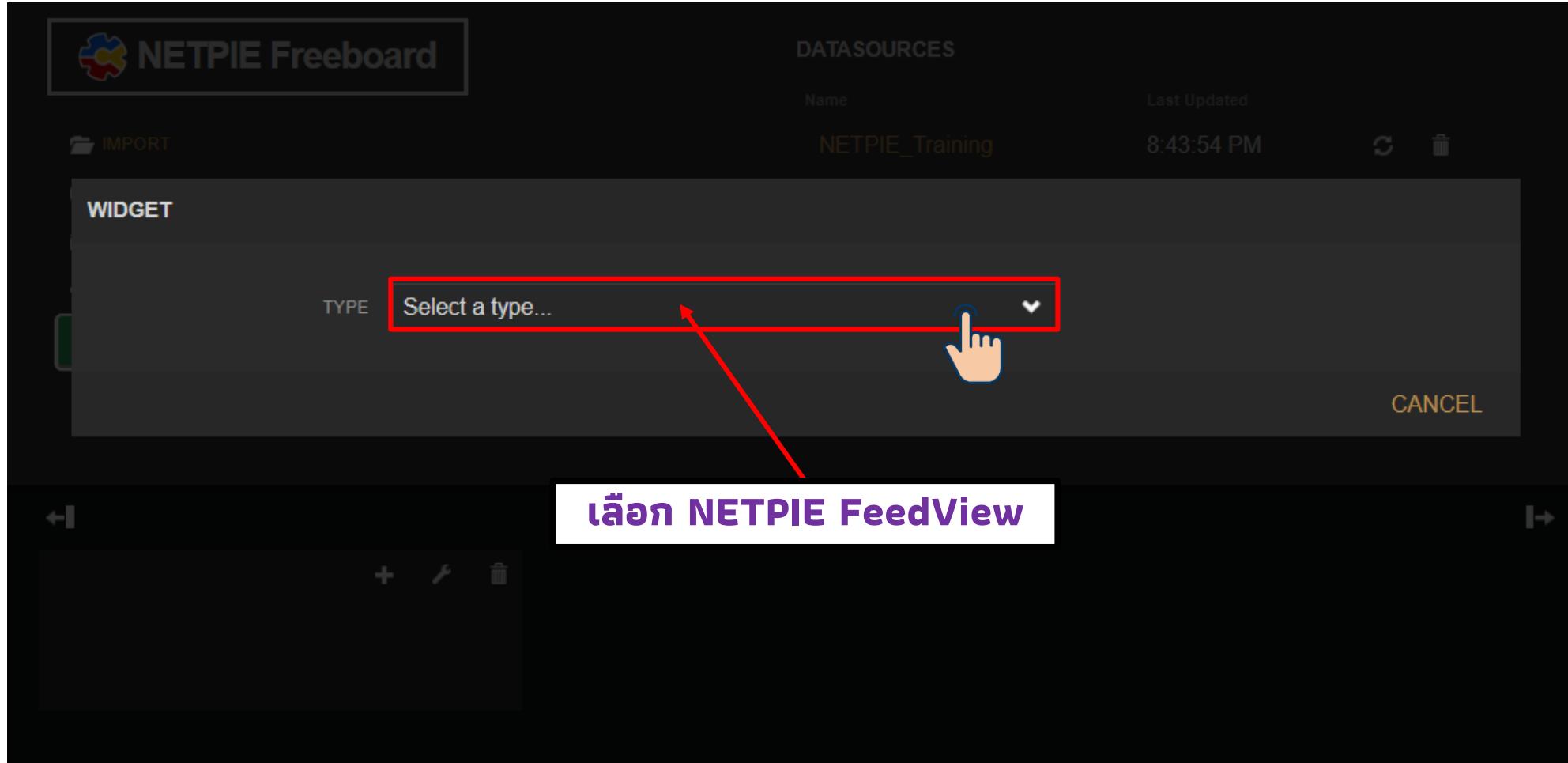
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Module 4 : NETPIE 2020 Freeboard



NECTEC
a member of NSTDA

Workshop 6 : การสร้าง Freeboard uu NETPIE2020

WIDGET

TYPE: NETPIE FeedView

TITLE:

DATA SOURCE: + DATASOURCE ✖ JS EDITOR

FILTER:
Data fields separated with comma e.g. temp,humid,light. Blank means display all fields.

TYPE OF CHART: Line

X AXIS TITLE:

Y AXIS TITLE:

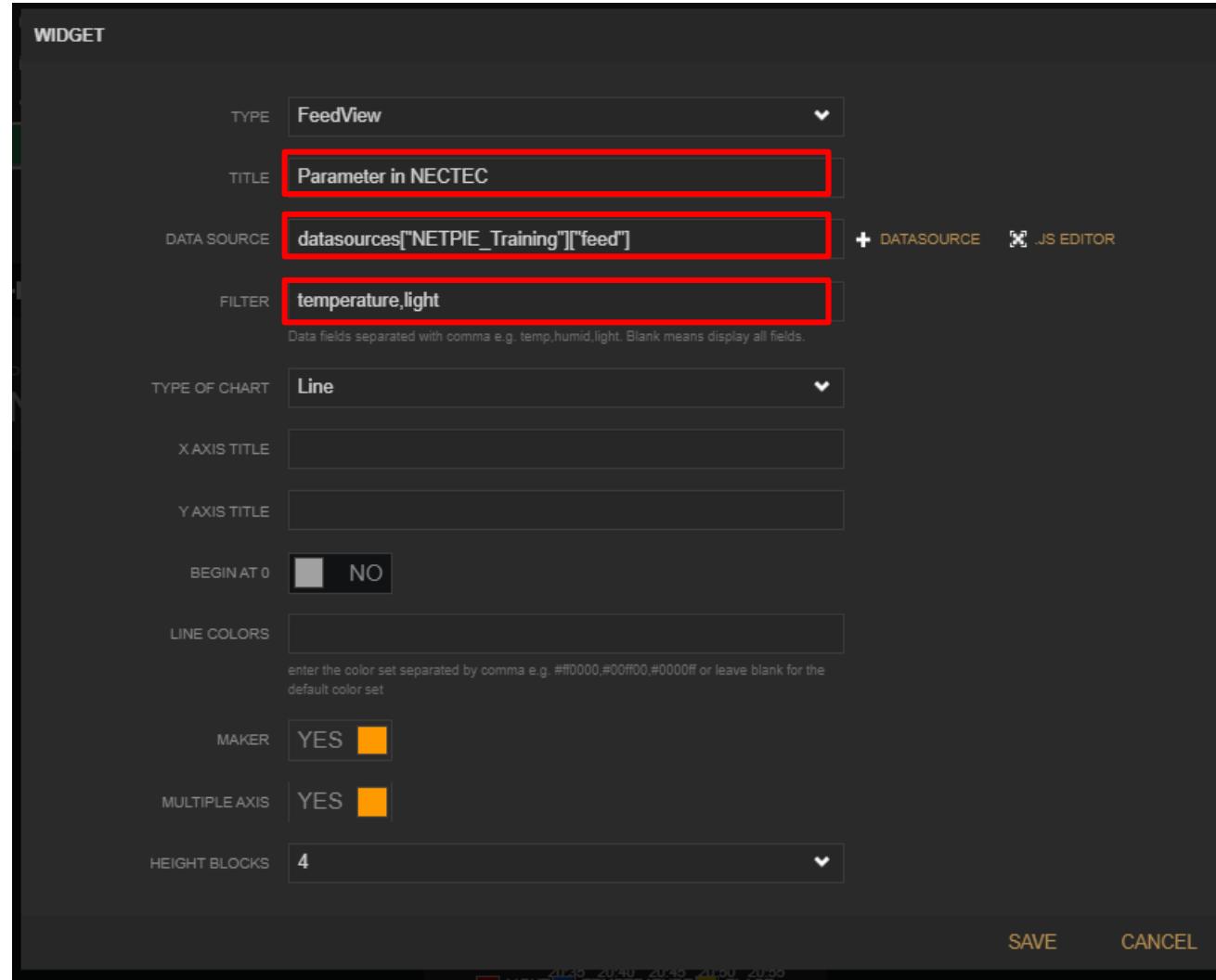
BEGIN AT 0: NO

LINE COLORS:
enter the color set separated by comma e.g. #ff0000,#00ff00,#0000ff or leave blank for the default color set

MAKER: YES

MULTIPLE AXIS: YES

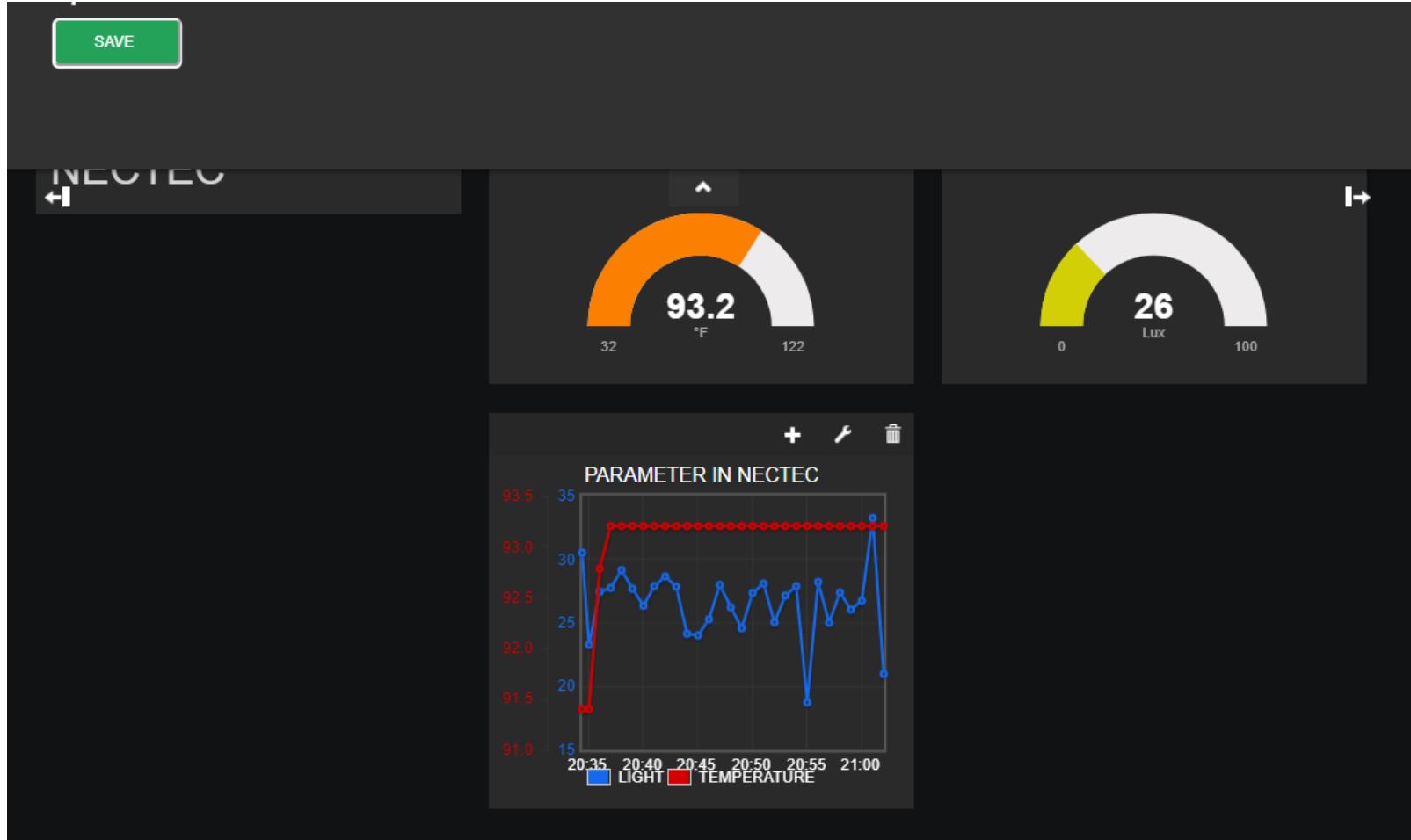
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Module 4 : NETPIE 2020 Freeboard



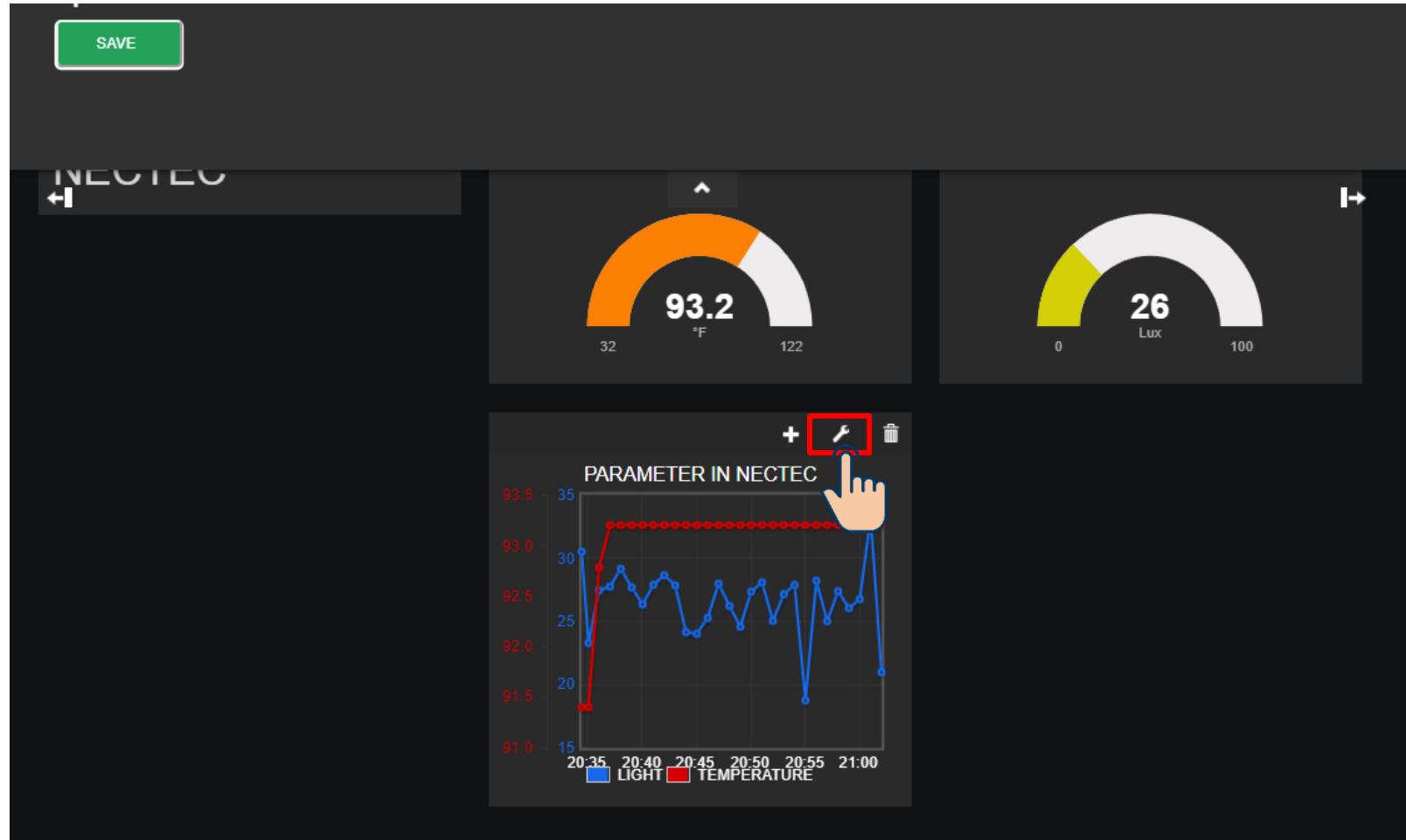
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



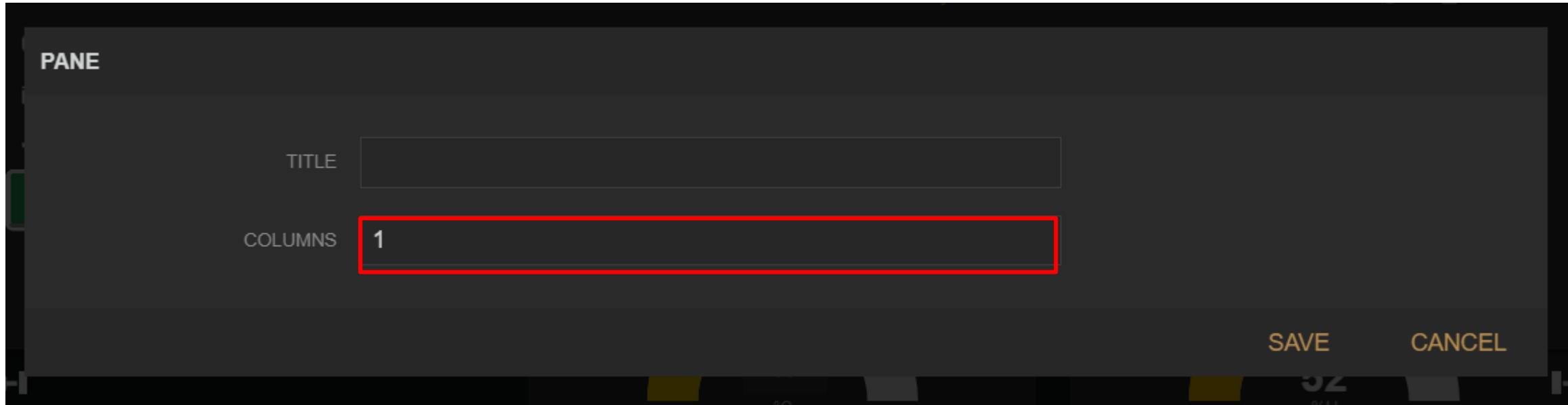
Module 4 : NETPIE 2020 Freeboard



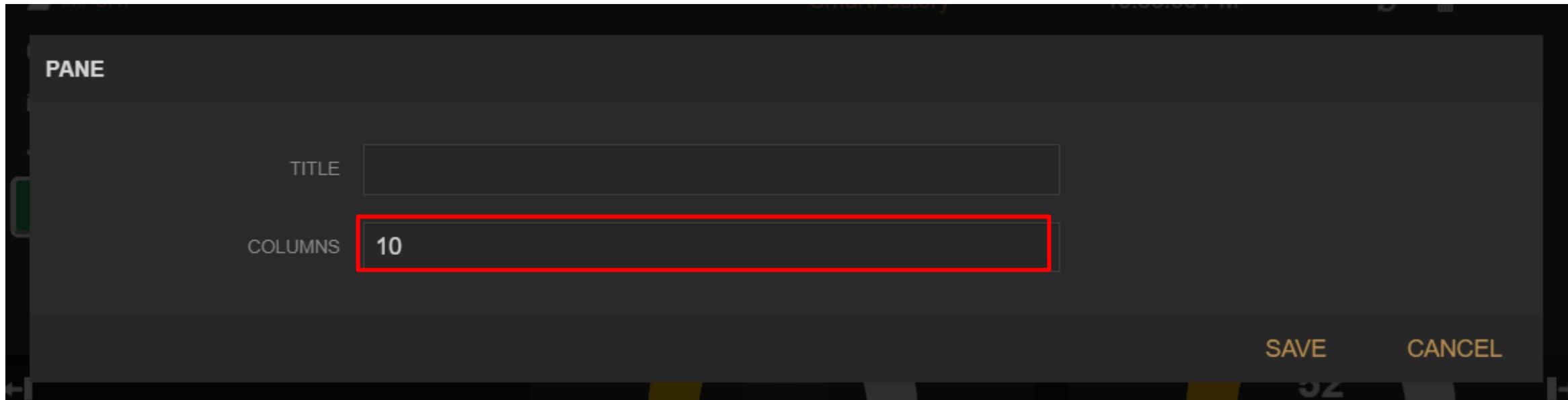
Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Workshop 6 : การสร้าง Freeboard uu NETPIE2020

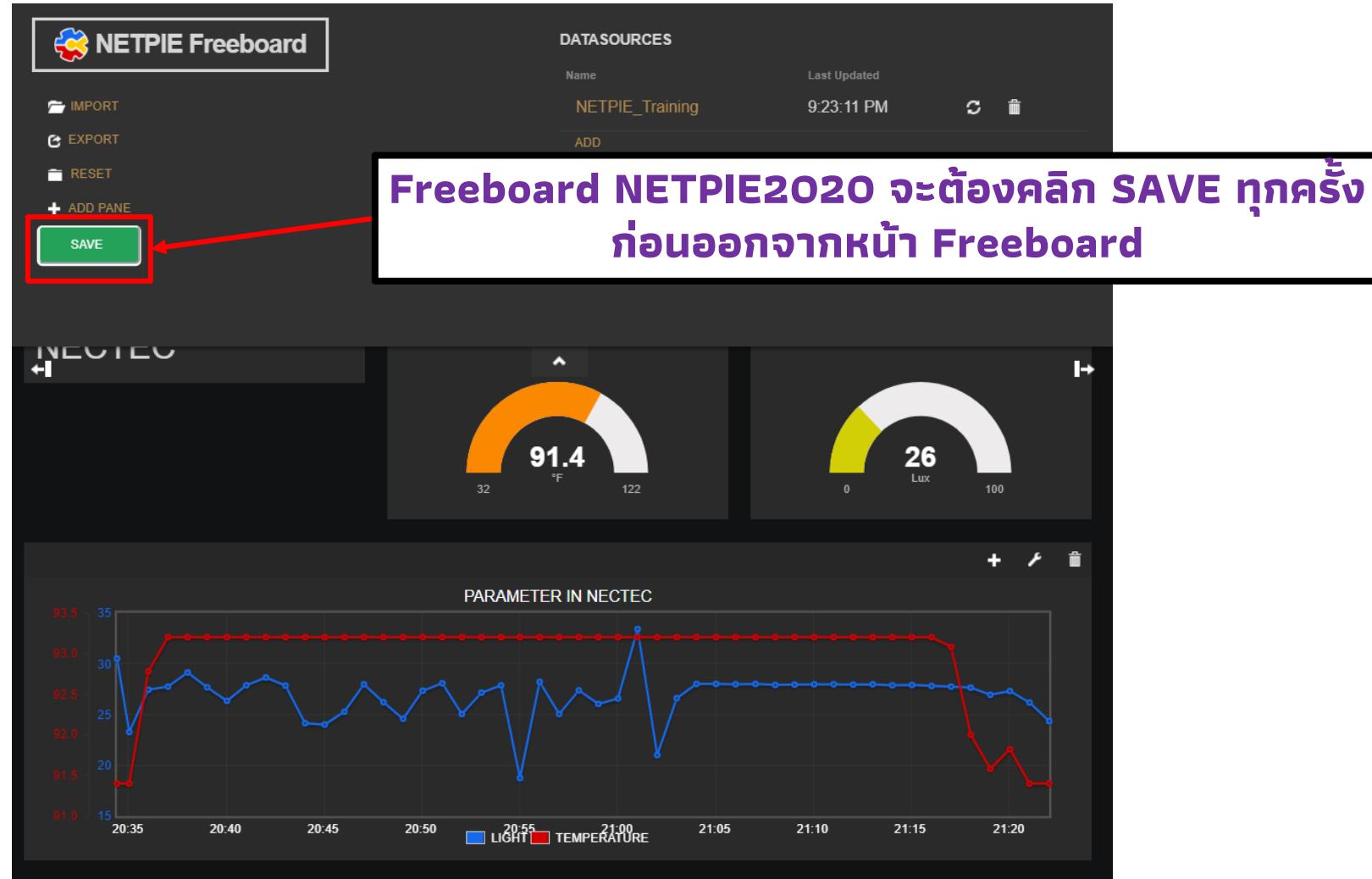


Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Module 4 : NETPIE 2020 Freeboard

Workshop 6 : การสร้าง Freeboard uu NETPIE2020

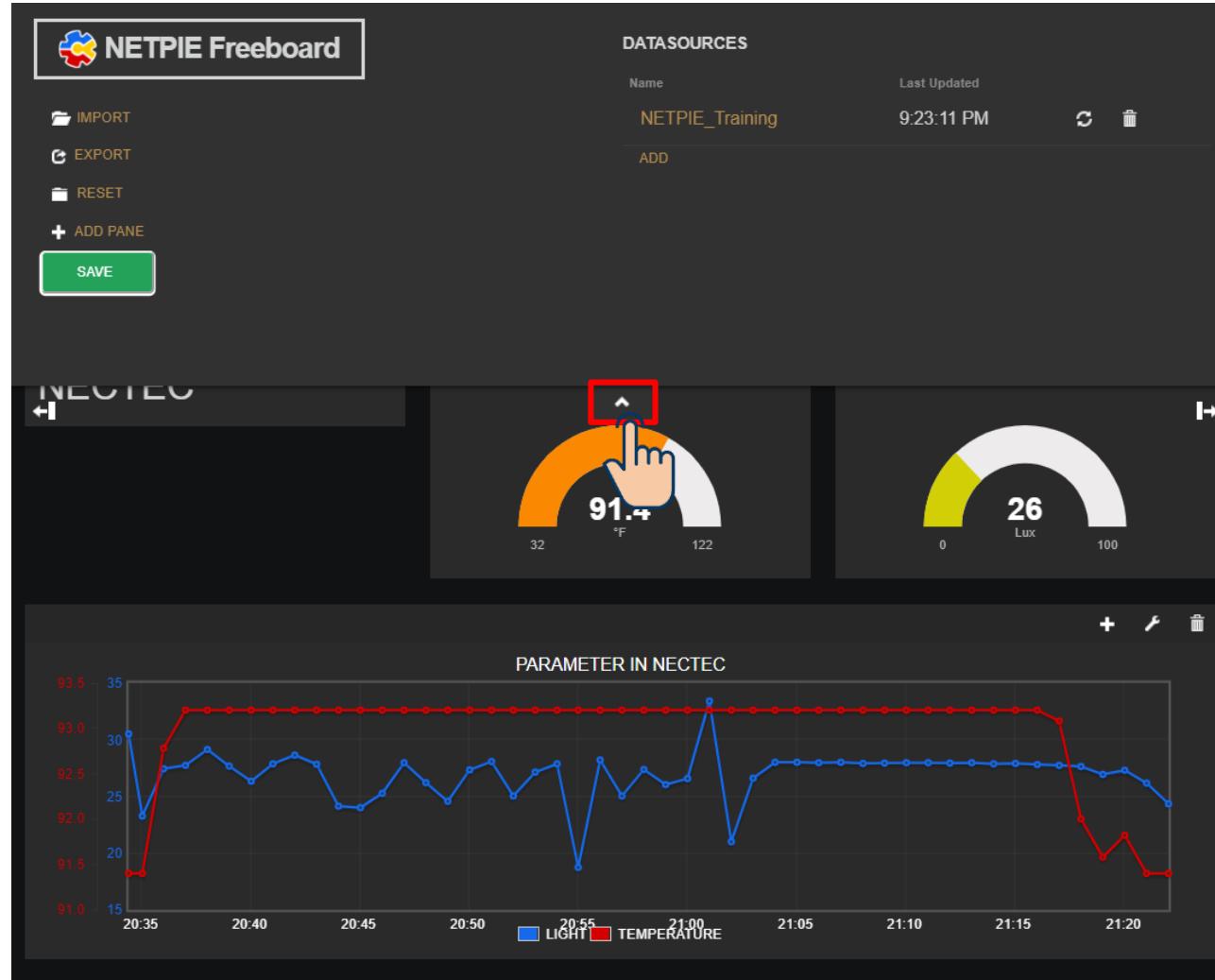


Module 4 : NETPIE 2020 Freeboard



NECTEC
a member of NSTDA

Workshop 6 : การสร้าง Freeboard uu NETPIE2020



Module 4 : NETPIE 2020 Freeboard

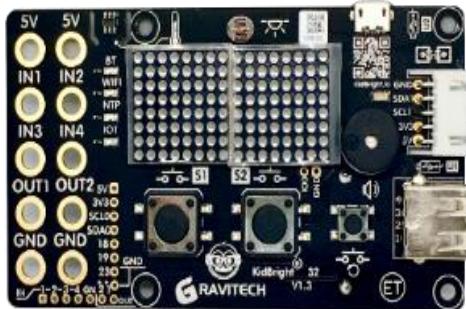


Workshop 6 : การสร้าง Freeboard บน NETPIE2020



Module 4 : NETPIE 2020 Freeboard

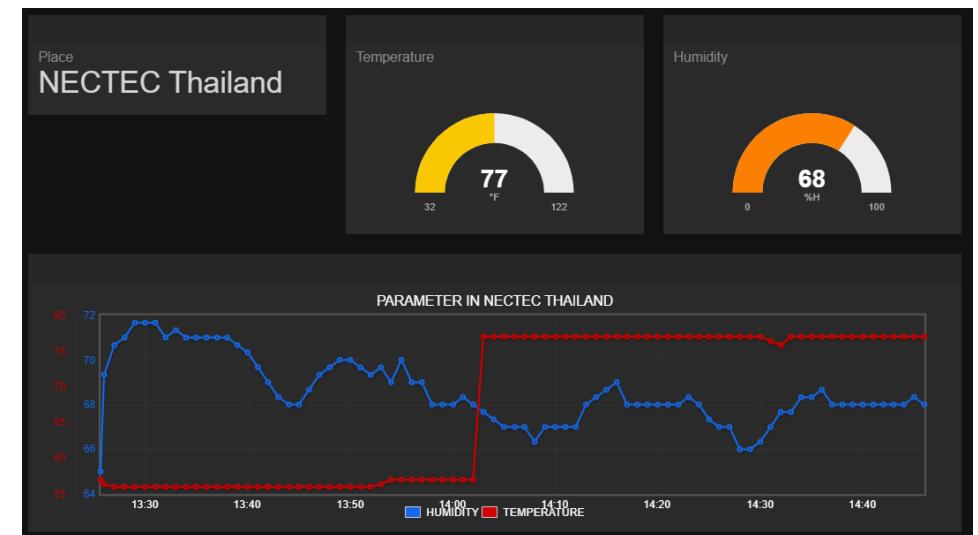
Workshop 7 : การควบคุม LED ผ่าน KidBright ผ่าน Freeboard



KidBright



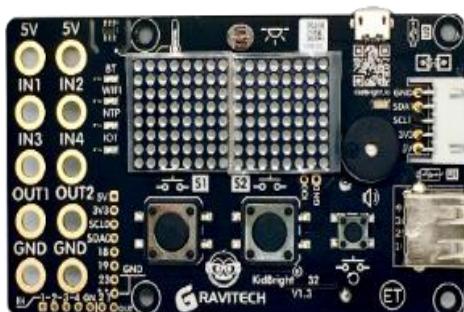
Freeboard



Workshop 7 : การควบคุม LED ผ่าน KidBright ผ่าน Freeboard

การควบคุม LED ผ่าน Freeboard สามารถทำได้ 2 วิธี

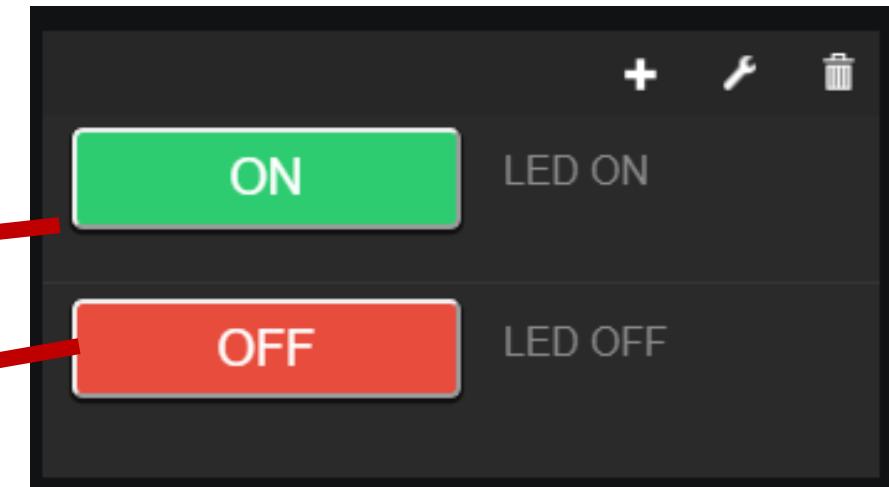
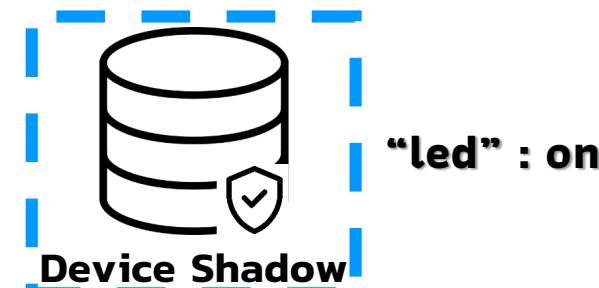
1. ควบคุมโดยสั่งผ่าน Shadow



KidBright

@shadow/data/update
“led” : “on”

@shadow/data/update
“led” : “off”



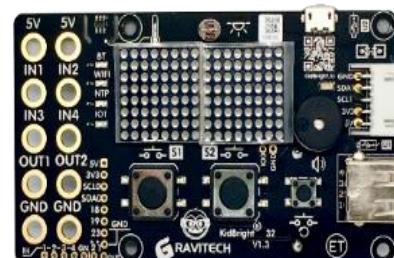
Workshop 7 : การควบคุม LED ผ่าน KidBright ผ่าน Freeboard

การควบคุม LED ผ่าน Freeboard สามารถทำได้ 2 วิธี

1. ควบคุมโดยสั่งผ่าน Shadow

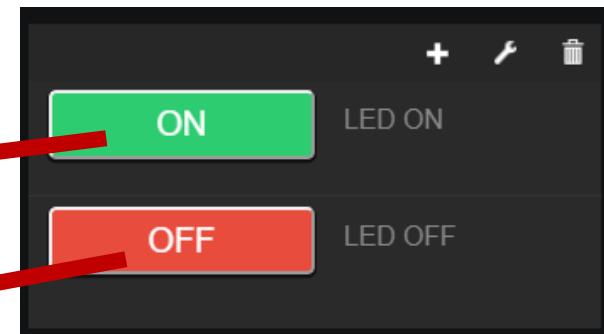
ข้อความที่ได้รับจากการส่งด้วยวิธี shadow มีรูปแบบดังนี้

```
b'{"deviceid":"xxxxxxxx-xxxx-xxxx-xxxx-xxxxxxxxxxxx","data":{"led":"on"},"rev":3,"modified":1579864929867}'
```



@shadow/data/update
"led" : "on"

@shadow/data/update
"led" : "off"



Workshop 7 : การควบคุม LED ผ่าน KidBright ผ่าน Freeboard

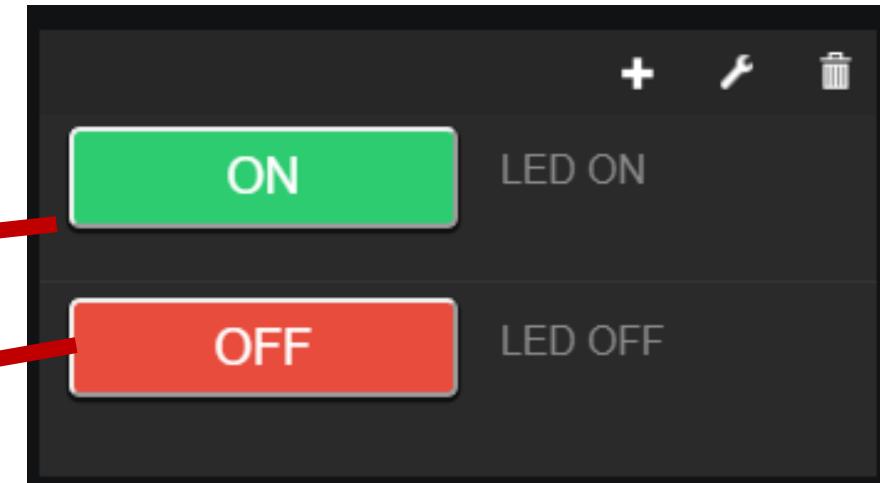
การควบคุม LED ผ่าน Freeboard สามารถทำได้ 2 วิธี

2. ควบคุมโดยสั่งผ่าน Message



@msg/led
“on”

@msg/led
“off”



@shadow/data/update
“led” : “on”



“led” : “on”

Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្តាង Freeboard

ເປີດໄຟຣ Workshop7.ino ខ្លឹម

Coding ໃນ Workshop7.ino ແບ່ງອອກເປັນ 3 ส່ວນ

1 ส່ວນទី 1 ການເຮັດວຽກໃຫ້ Library ແລະ ປະກາສຕັວແປ

ປະກາສຕັວແປແລະປະກາສຫາ LED

```
#include <WiFi.h>
#include <PubSubClient.h>
#include <Wire.h>

#define LM73_ADDR 0x4D
#define LED 12

const char* ssid = "Your_SSID";
const char* password = "Your_PASSWORD";
const char* mqtt_server = "broker.netpie.io";
const int mqtt_port = 1883;
const char* mqtt_Client = "Client_ID";
const char* mqtt_username = "Token";
const char* mqtt_password = "Password";

int temperature = 0;
int light = 0;
int analog_value = 0;
char msg[100];
long lastMsg = 0;

WiFiClient espClient;
PubSubClient client[espClient];
```

Workshop 7 : การគុបគុម LED ឬ KidBright ជាន Freeboard

2

ស៊ូបទំ 2 ស៊ូនខែងកែចំនាំពារ

```
float readTemperature() {  
    Wire1.beginTransmission(LM73_ADDR);  
    Wire1.write(0x00);  
    Wire1.endTransmission();  
    uint8_t count = Wire1.requestFrom(LM73_ADDR, 2);  
    float temp = 0.0;  
    if [count == 2] {  
        byte buff[2];  
        buff[0] = Wire1.read();  
        buff[1] = Wire1.read();  
        temp += [int][buff[0] << 1];  
        if [buff[1] & 0b10000000] temp += 1.0;  
        if [buff[1] & 0b01000000] temp += 0.5;  
        if [buff[1] & 0b00100000] temp += 0.25;  
        if [buff[0] & 0b10000000] temp *= -1.0;  
    }  
    return temp;  
}
```

ផែងកែចំនាំរាយតាមអ្នកមិនលេង

```
float readlight() {  
    analog_value = analogRead(36);  
    if [analog_value > 1000] {  
        analog_value = 1000;  
    }  
    int light = [1000 - analog_value] * 100 / 1000;  
    return light;  
}
```

Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្តាម Freeboard

2 ស៊ូបទ់ 2 ស៊ូបនខែងកិច្ចនពេញពេញ

ផែងកិច្ចនការចេះអំពេល MQTT

```
void reconnect() {  
    while (!client.connected()) {  
        Serial.print("Attempting MQTT connection...");  
        if (client.connect(mqtt_Client, mqtt_username, mqtt_password)) {  
            Serial.println("connected");  
            client.subscribe("@msg/led");  
        }  
        else {  
            Serial.print("failed, rc=");  
            Serial.print(client.state());  
            Serial.println("try again in 5 seconds");  
            delay(5000);  
        }  
    }  
}
```

កំការ Subscribe Topic នៃ Freeboard តាមសំណង់ខ្លាំង

Module 4 : NETPIE 2020 Freeboard

Workshop 7 : การควบคุม LED 用 KidBright ผ่าน Freeboard

2 | ส่วนที่ 2 ส่วนของพิงก์ชั้นต่างๆ

ฟังก์ชันการรับข้อความจาก MQTT

```
void callback(char* topic, byte* payload, unsigned int length) {  
    Serial.print("Message arrived [");  
    Serial.print(topic);  
    Serial.print("] ");  
    String message;  
    for (int i = 0; i < length; i++) {  
        message = message + (char)payload[i];  
    }  
    Serial.println(message);  
}
```

ชุดคำสั่งนำข้อมูลที่ได้รับมาจาก MQTT มาแสดงผล

```
if[String(topic) == "@msg/led"] {  
    if [message == "on"]{  
        digitalWrite(LED1,0);  
        client.publish["@shadow/data/update", "{\"data\": {\"led\": \"on\"}}"];  
        Serial.println("LED ON"); }  
    else if [message == "off"] {  
        digitalWrite(LED1,1);  
        client.publish["@shadow/data/update", "{\"data\": {\"led\": \"off\"}}"];  
        Serial.println("LED OFF"); }  
}
```

เงื่อนไขการเปิด/ปิด LED
โดยเริ่มจากการเช็ค Topic ว่าเป็น @msg/led หรือไม่
แล้วทำการเช็คข้อความที่ได้รับมาว่าเป็น on หรือ off
เนื่องจาก LED บน NodeMCU เป็นการต่อวงจรแบบ Pull Up
จึงทำให้ทุกครั้งที่ได้รับ on จะส่ง 0 ไปที่ขา LED
และส่ง 1 ไปที่ขา LED เมื่อได้รับ off

Workshop 7 : การគុបគុម LED ឬ KidBright ជាន Freeboard

2

ស៊ូនទី 2 ស៊ូនខែងកិច្ចនតោញា

```
void setup() {  
    Serial.begin(115200);  
    Wire1.begin(4, 5);  
    pinMode(LED, OUTPUT);  
    digitalWrite(LED, 1);  
  
    Serial.println("Starting...");  
    if (WiFi.begin(ssid, password)) {  
        while (WiFi.status() != WL_CONNECTED) {  
            delay(1000);  
            Serial.print(".");  
        }  
        Serial.println("WiFi connected");  
        Serial.println("IP address: ");  
        Serial.println(WiFi.localIP());  
        client.setServer(mqtt_server, mqtt_port);  
        client.setCallback(callback);  
    }  
}
```

ឯងកិច្ចនការព័ត៌មានតោញា

ការព័ត៌មាន LED ឬកិច្ចនតោញា ត្រូវបានរំពៀនដោយការបង្កើតការព័ត៌មានតោញានៅក្នុងកិច្ចន។ ការបង្កើតការព័ត៌មានតោញានឹងត្រូវការពេលវេលាដែលត្រូវបានរំពៀនដោយការបង្កើតការព័ត៌មានតោញាន។

ការបង្កើតការព័ត៌មានតោញានត្រូវបានរំពៀនដោយការបង្កើតការព័ត៌មានតោញាន។ ការបង្កើតការព័ត៌មានតោញានត្រូវបានរំពៀនដោយការបង្កើតការព័ត៌មានតោញាន។

Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្លាស់ប្តូរ Freeboard

3

ស៊ីវភៅ 3 ស៊ីវភៅខិះខំលក់

```
void loop() {
    if (!client.connected()) {
        reconnect();
    }
    client.loop();
    temperature = readTemperature();
    light = readlight();
    String place = "NECTEC";
```

ໃច្ចុដការណើបវេលាយោប់ millis នៃការិយៈ delay ដើម្បីអាមេរកការងារ
ការងារនៃការណើបទាំងនេះត្រូវការពេលវេលាយោប់រហូតដល់ការណើបទាំងនេះ

```
long now = millis();
if (now - lastMsg > 5000) {
    lastMsg = now;
    ++value;
    String data = "{\"data\": {\"humidity\":\"" + String(humidity) + "\", \"temperature\":\"" + String(temperature) + ",
    \"place\": \"" + String(place) + "\"}}";
    Serial.println(data);
    data.toCharArray(msg, (data.length() + 1));
    client.publish("@shadow/data/update", msg); }
delay[1];
```

Workshop 7 : การควบคุม LED ผ่าน KidBright ผ่าน Freeboard

Device Schema ในรูปแบบ JSON

```
{  
  "additionalProperties": false,  
  "properties": {  
    "humidity": {  
      "operation": {  
        "store": {  
          "ttl": "7d"  
        }  
      },  
      "type": "number"  
    },  
    "temperature": {  
      "operation": {  
        "store": {  
          "ttl": "7d"  
        }  
      },  
      "transform": {  
        "expression": "[$.temperature]*1.8 + 32"  
      },  
      "type": "number"  
    }  
  }  
}
```

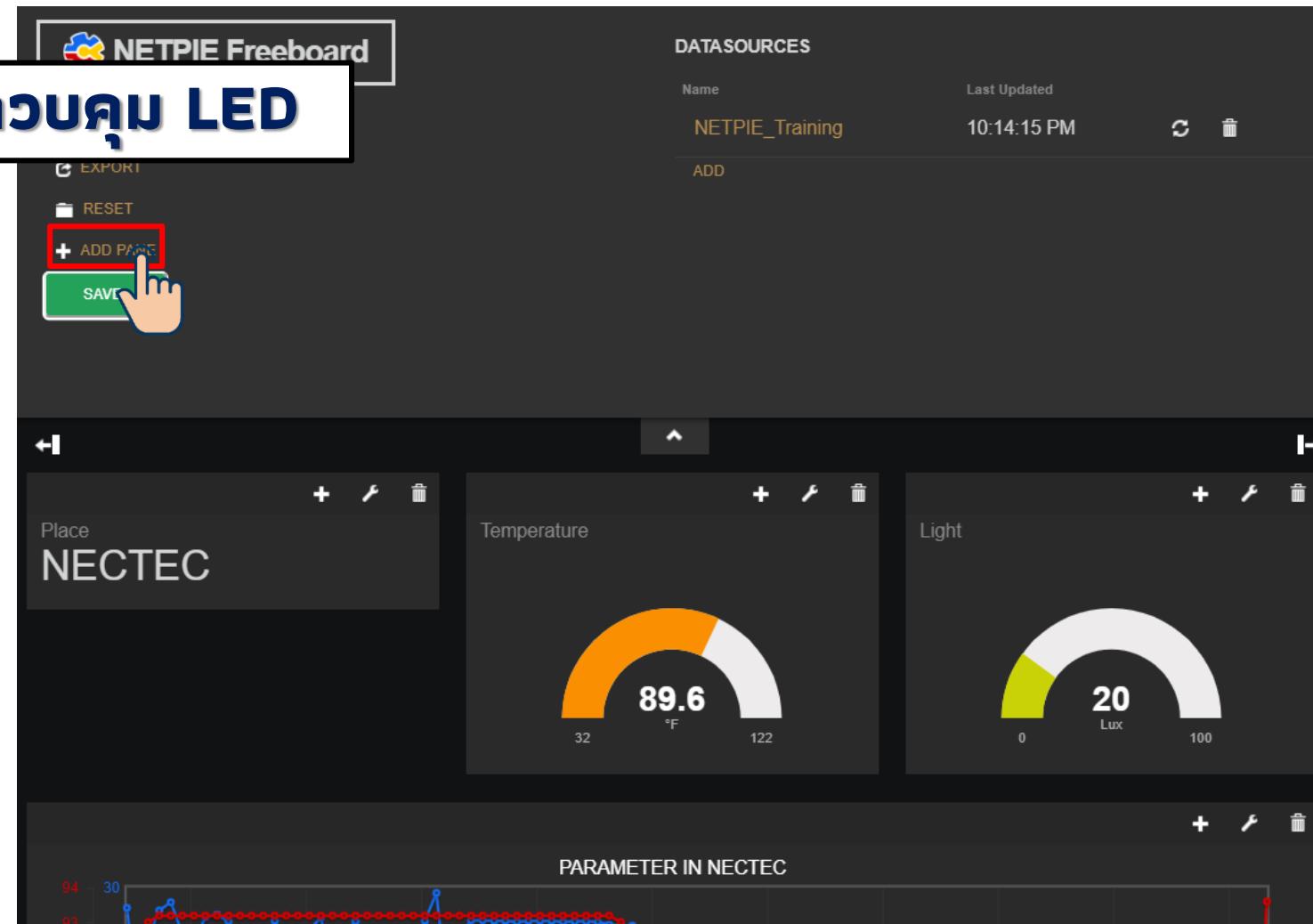
เพิ่มข้อมูลที่ต้องการจัดเก็บบน Schema

```
"place": {  
  "operation": {  
    "store": {  
      "ttl": "7d"  
    }  
  },  
  "type": "string"  
},
```

```
"led": {  
  "operation": {  
    "store": {  
      "ttl": "7d"  
    }  
  },  
  "type": "string"  
}  
}
```

- จัดเก็บตัวแปรประเภทตัวอักษรโดยมีคุณสมบัติดังนี้
 - จัดเก็บจำนวน 7 วัน
 - จัดเก็บตัวแปรประเภท string

Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្តាម Freeboard



Module 4 : NETPIE 2020 Freeboard

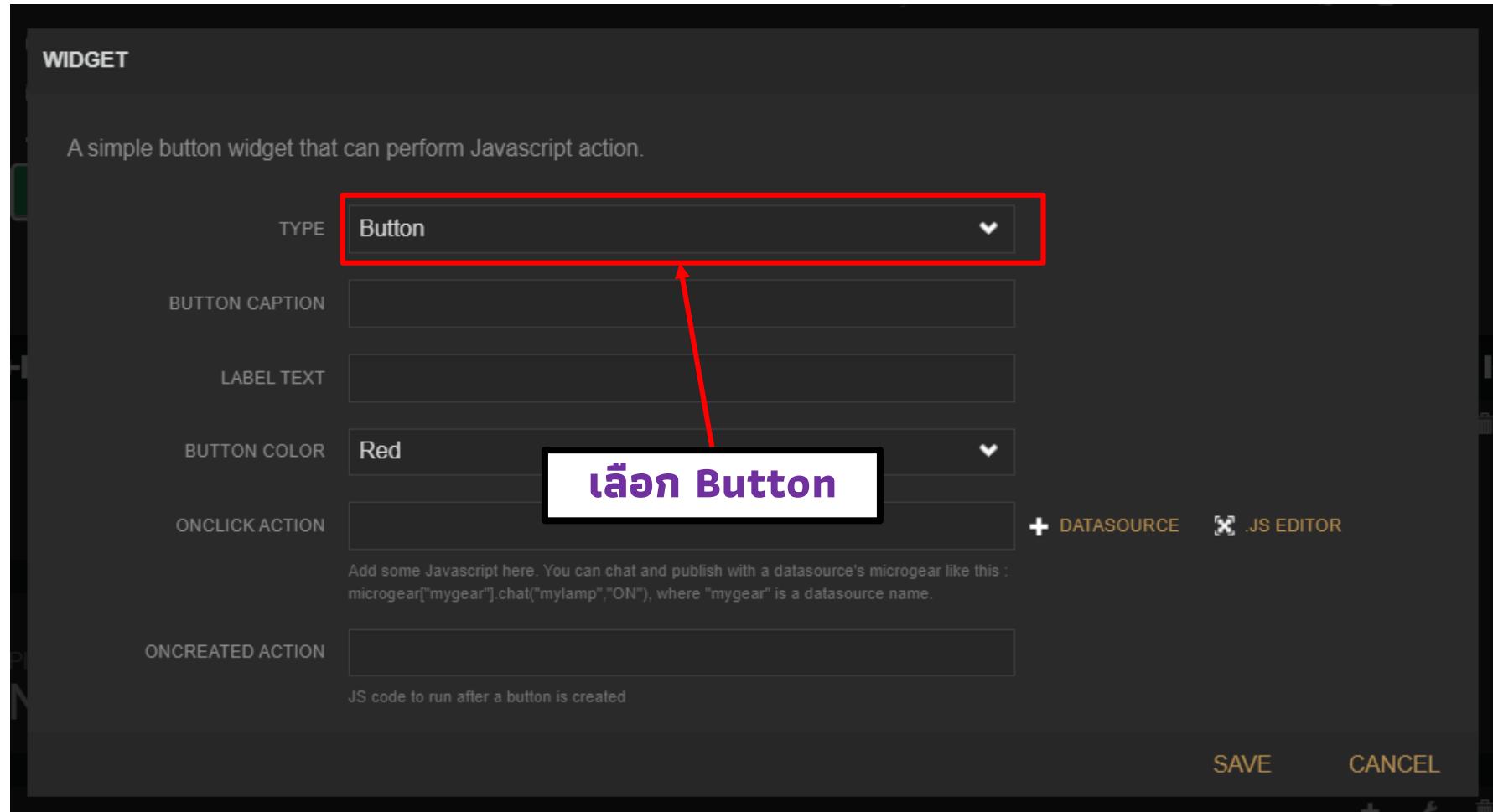


NECTEC
a member of NSTDA

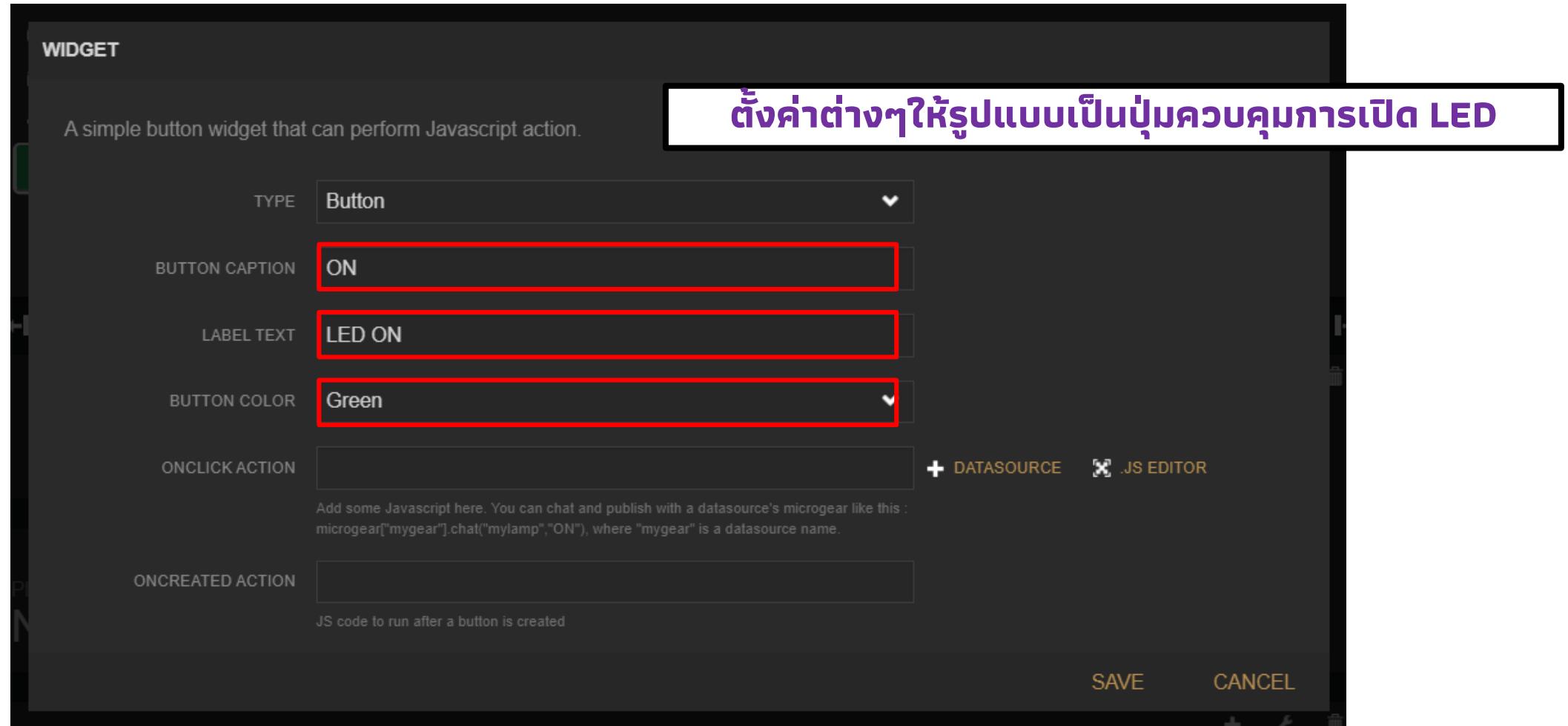
Workshop 7 : การควบคุม LED ด้วย KidBright ผ่าน Freeboard

The screenshot shows the NETPIE Freeboard interface. At the top left is the title "NETPIE Freeboard" with a logo. To its right is a "DATA SOURCES" section showing a single entry: "NETPIE_Training" last updated at "10:14:15 PM". Below this are "ADD", "C", and "DELETE" buttons. On the left side, there's a sidebar with "IMPORT", "EXPORT", "RESET", "ADD PANE", and a green "SAVE" button. The main area contains three gauge widgets. The first gauge is labeled "Temperature" with a value of "89.6 °F" and scale marks at 32 and 122. The second gauge is labeled "Light" with a value of "20 Lux" and scale marks at 0 and 100. The third gauge is labeled "Place" and has "NECTEC" written in it. Below the gauges is a footer bar with the text "PARAMETER IN NECTEC" and some numerical values: 94, 30, and 1.

Workshop 7 : การควบคุม LED uu KidBright ผ่าน Freeboard



Workshop 7 : การគុបគុម LED ឱ្យ KidBright ជាន Freeboard



Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្តាន Freeboard

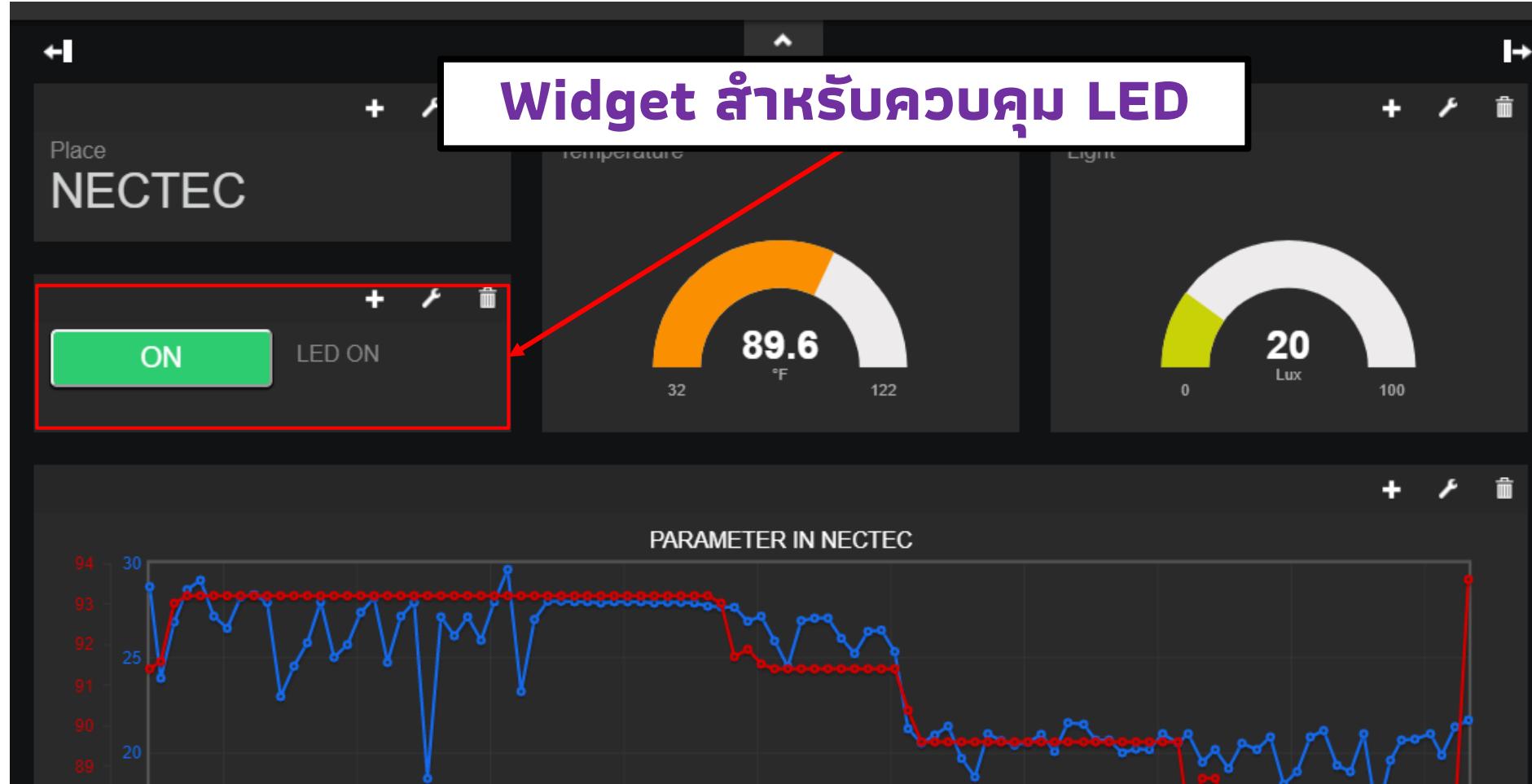
ONCLICK ACTION គឺ រូបແບບខ្លះគ្រាមអនីអូមូលទៀត
ត้องការសំងខែកមើំនៅតិចឡើងទៀត
ដើម្បីរួមចូលរួមជាប្រព័ន្ធដែលបានរាយ

The screenshot shows the configuration interface for a button widget. The configuration fields are:

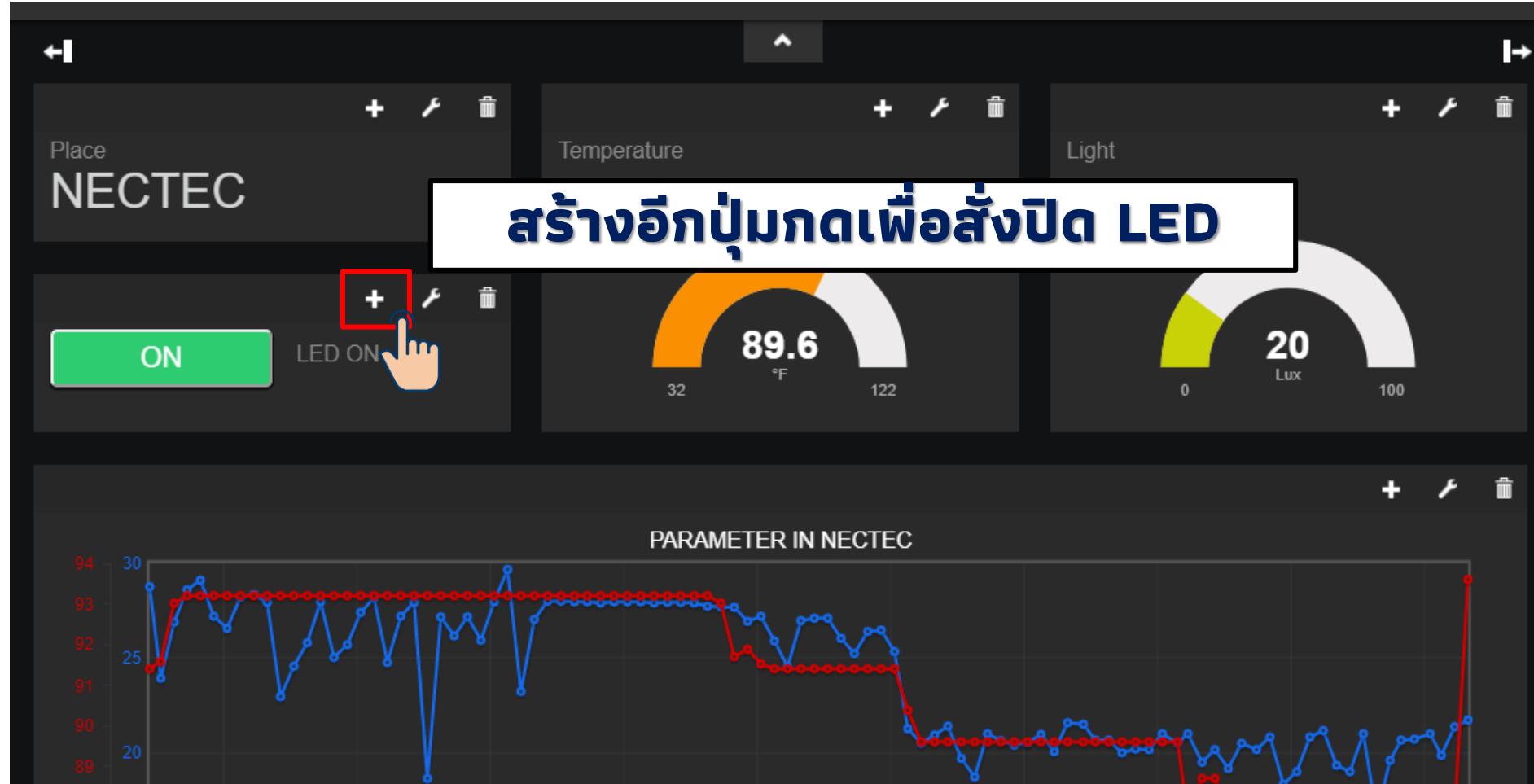
- WIDGET**: A simple button widget that can perform Java
- TYPE**: Button
- BUTTON CAPTION**: ON
- LABEL TEXT**: LED ON
- BUTTON COLOR**: Green
- ONCLICK ACTION**: netpie["datasources name"].publish("@msg/led","on")
- ONCREATED ACTION**: Add some Javascript here.

A red arrow points from the explanatory text above to the **ONCLICK ACTION** field, highlighting the code. The **ONCLICK ACTION** field is also highlighted with a red border.

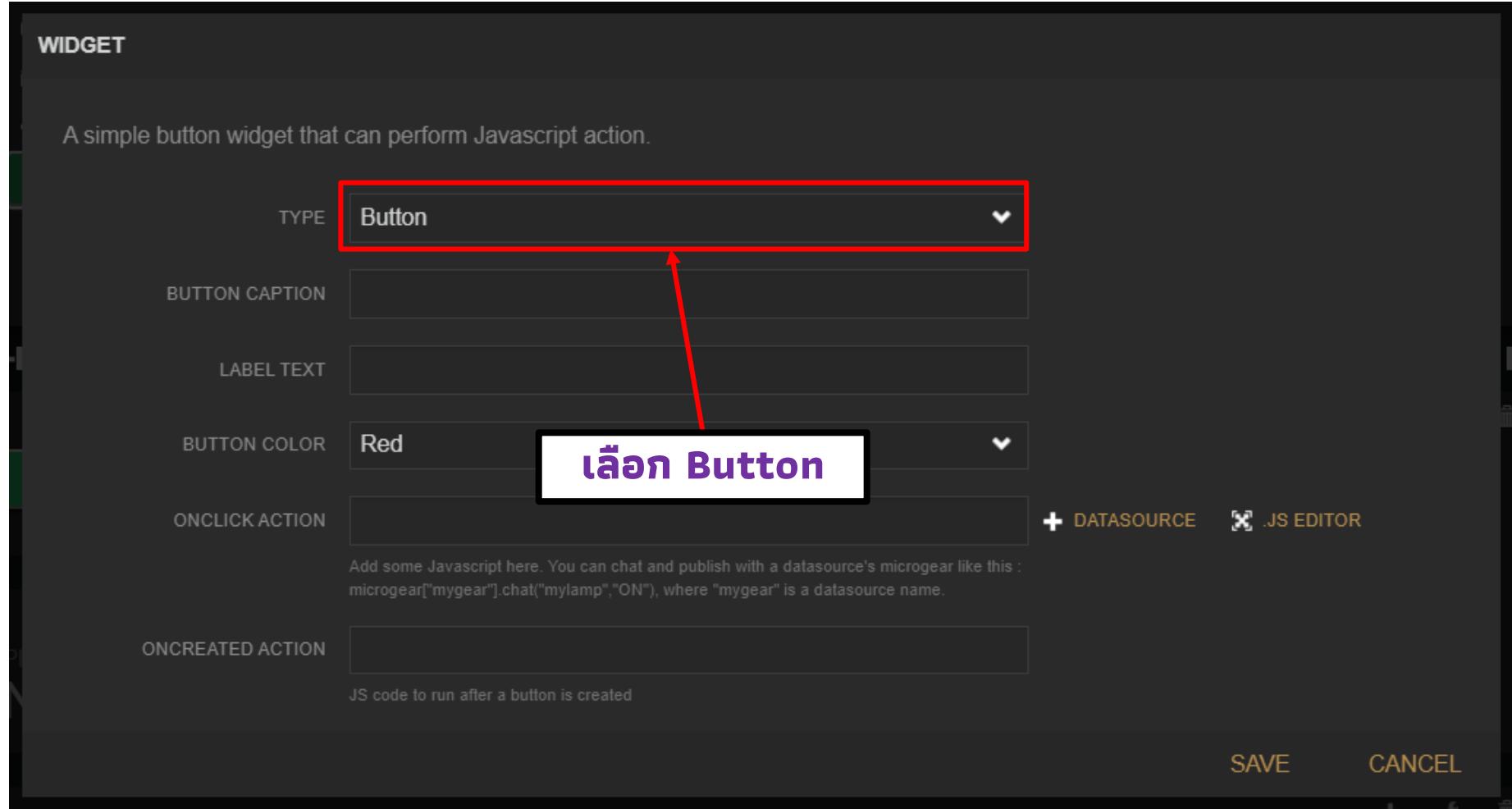
Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្ទាល់ Freeboard



Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្ទាល់ Freeboard

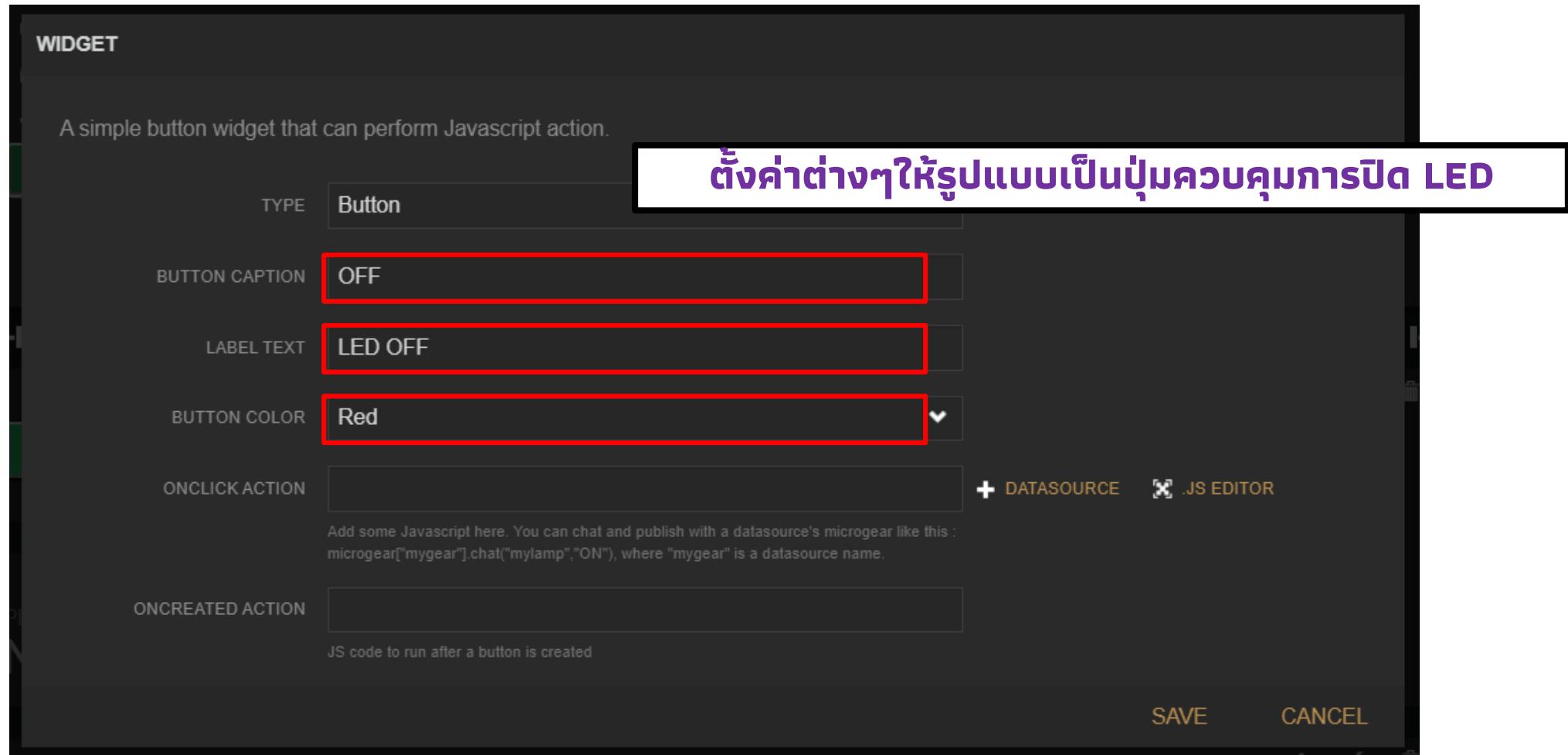


Workshop 7 : การគុបគុម LED ឱ្យ KidBright ជាន Freeboard



Module 4 : NETPIE 2020 Freeboard

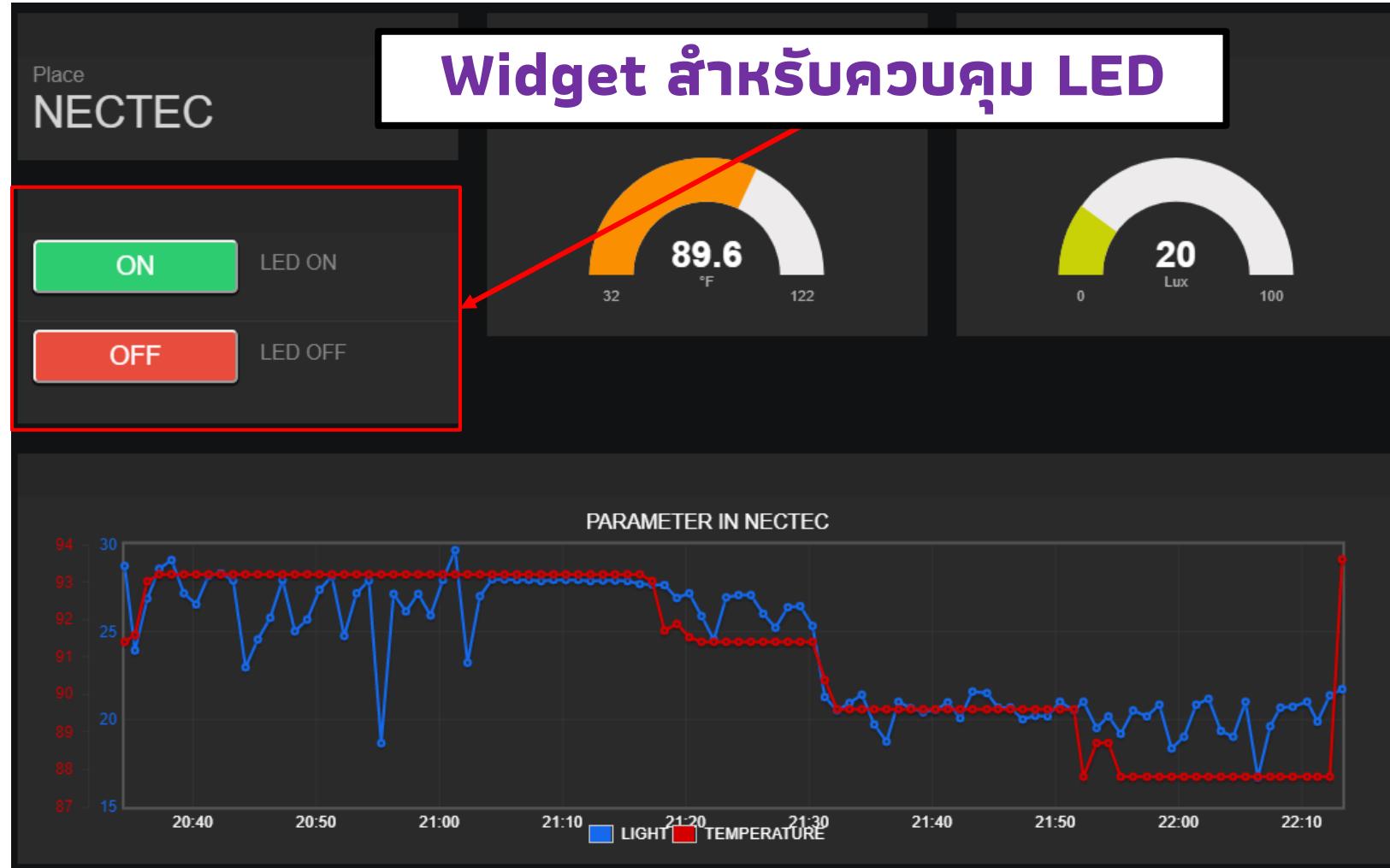
Workshop 7 : การควบคุม LED ด้วย KidBright ผ่าน Freeboard



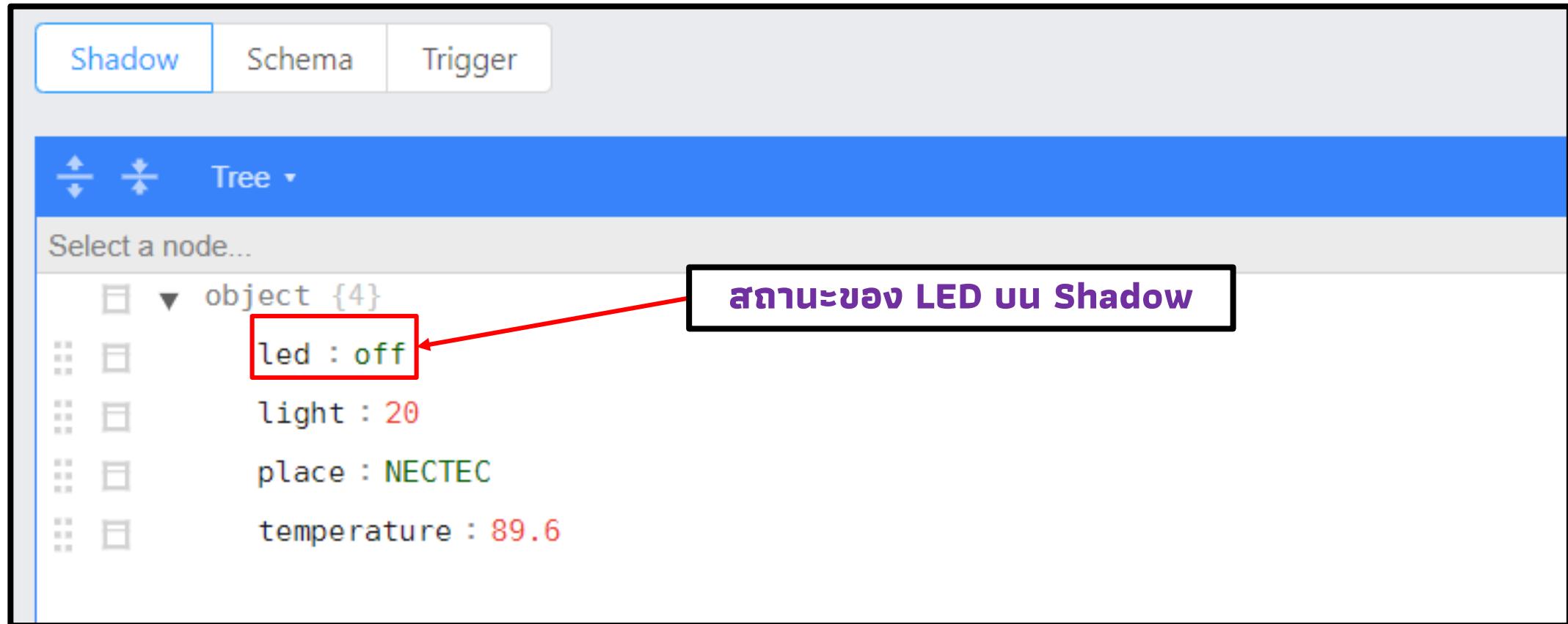
Module 4 : NETPIE 2020 Freeboard

Workshop 7 : การควบคุม LED 用 KidBright ผ่าน Freeboard

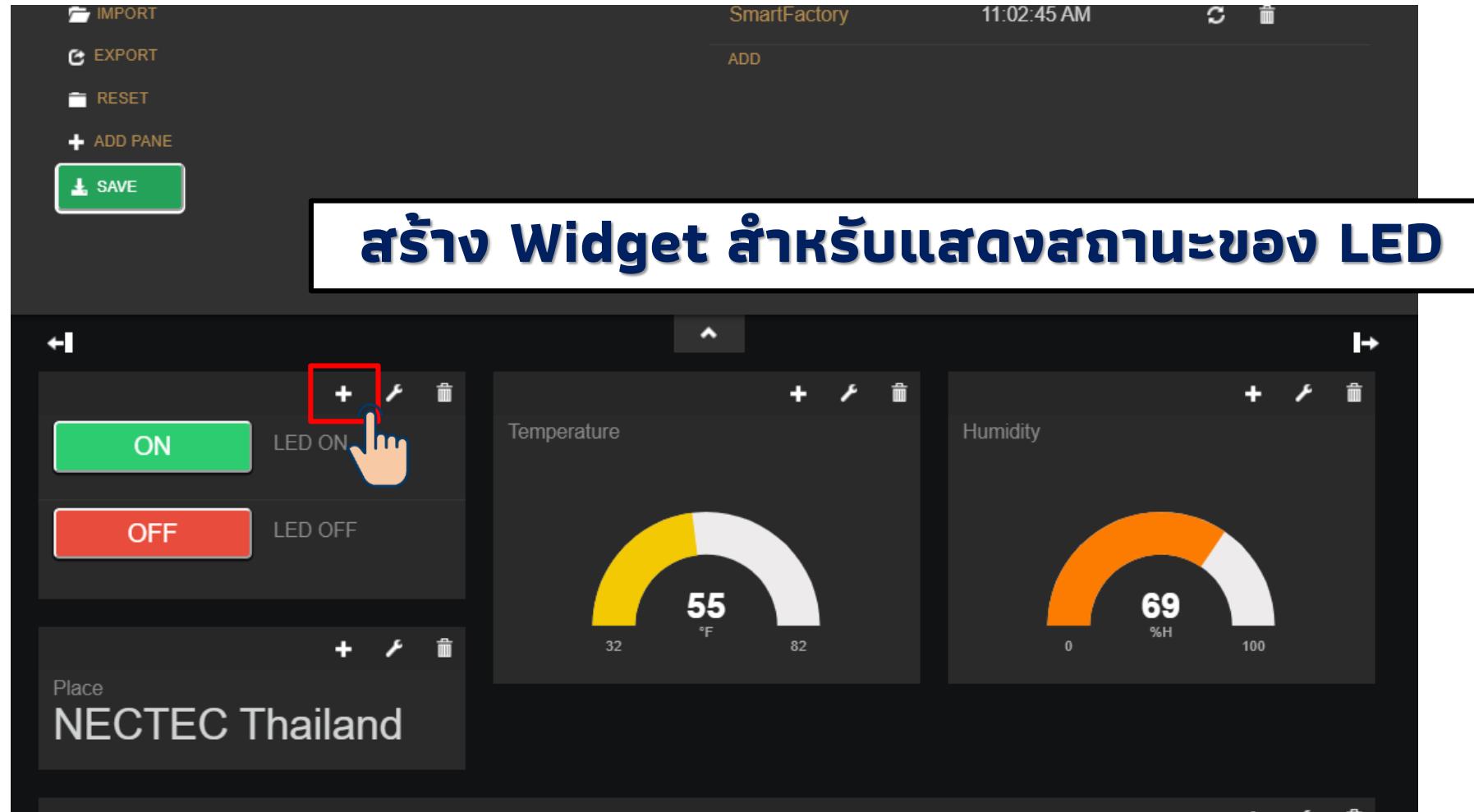
Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្តាង Freeboard



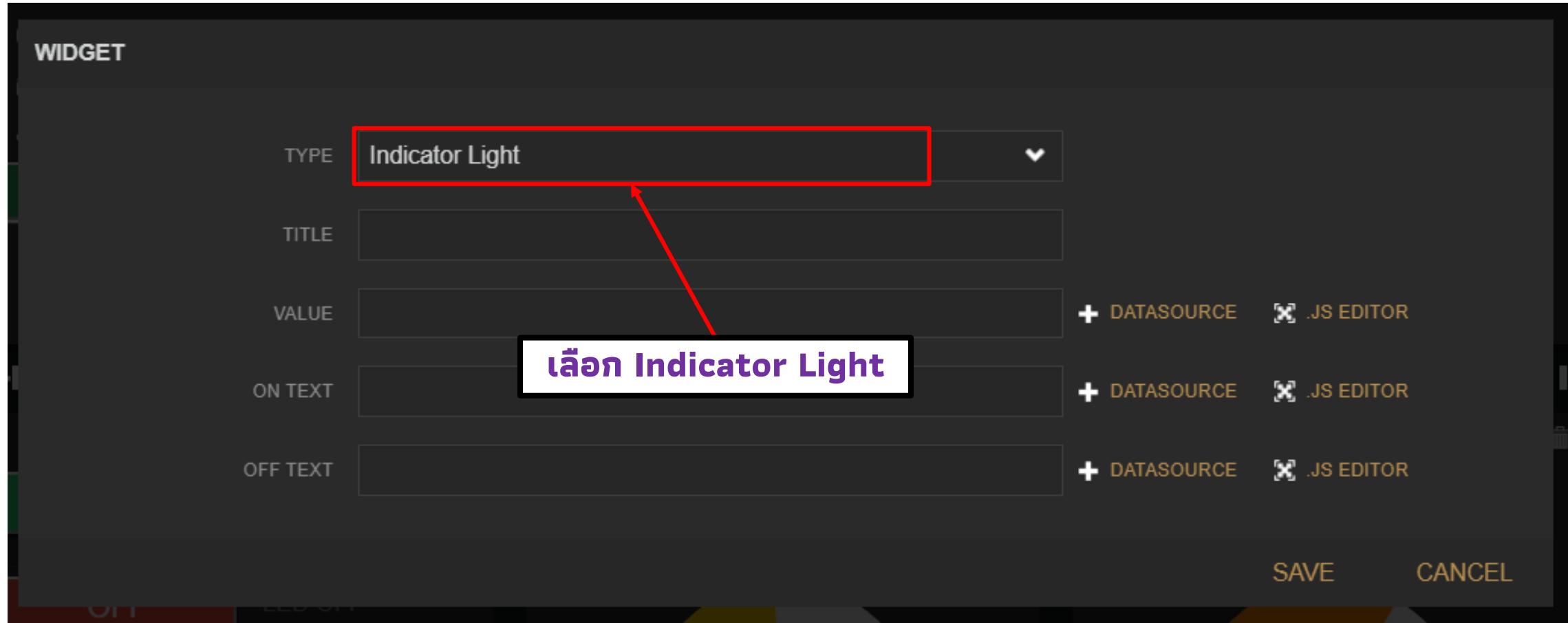
Workshop 7 : ការគិតបញ្ជី LED ឬ KidBright ជាន Freeboard



Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្ទាល់ Freeboard



Workshop 7 : การគោបកុម LED ឱ្យ KidBright ផ្លាស Freeboard



Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្តាង Freeboard

WIDGET

TYPE: Indicator Light

TITLE: Status LED

VALUE: + DATASOURCE JS EDITOR

ON TEXT: LED ON + DATASOURCE JS EDITOR

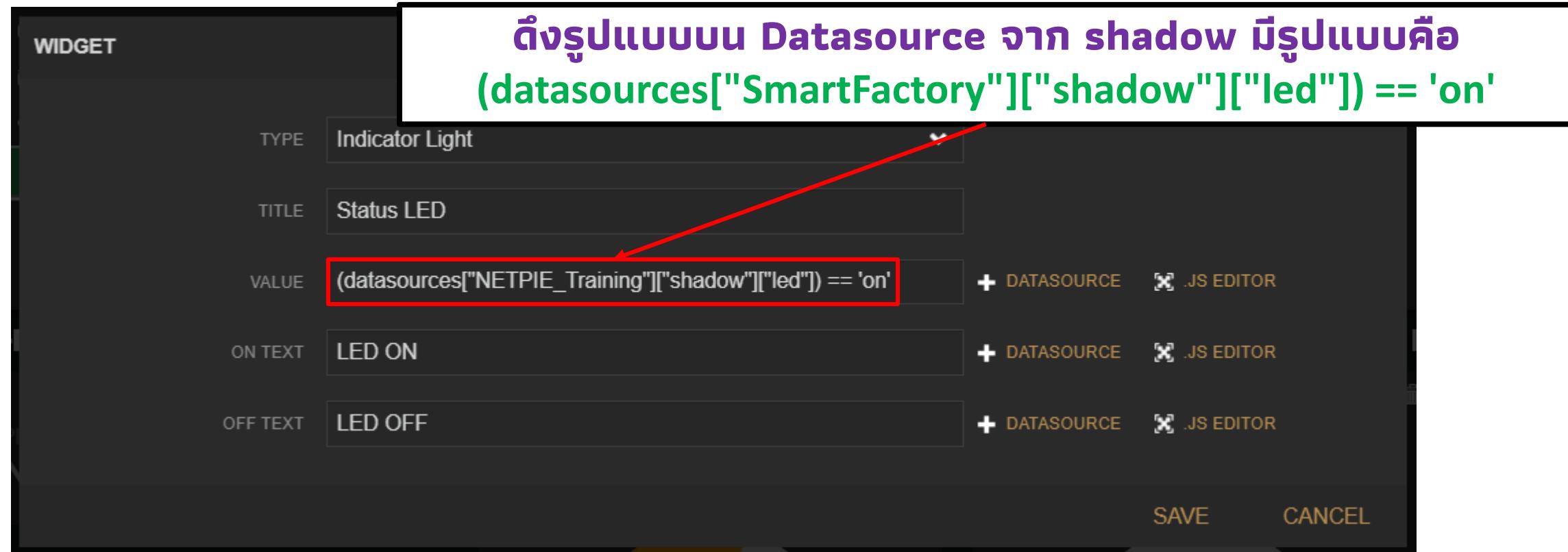
OFF TEXT: LED OFF + DATASOURCE JS EDITOR

SAVE CANCEL

ON LED OFF

The screenshot shows the configuration screen for an 'Indicator Light' widget. The title is 'Status LED'. The 'ON TEXT' field contains 'LED ON' and the 'OFF TEXT' field contains 'LED OFF'. Both fields have red boxes around them, indicating they are selected or highlighted. There are '+ DATASOURCE' and 'JS EDITOR' buttons next to each field. At the bottom right are 'SAVE' and 'CANCEL' buttons.

Workshop 7 : การគុបគុម LED ឱ្យ KidBright ផ្ទាល់ Freeboard

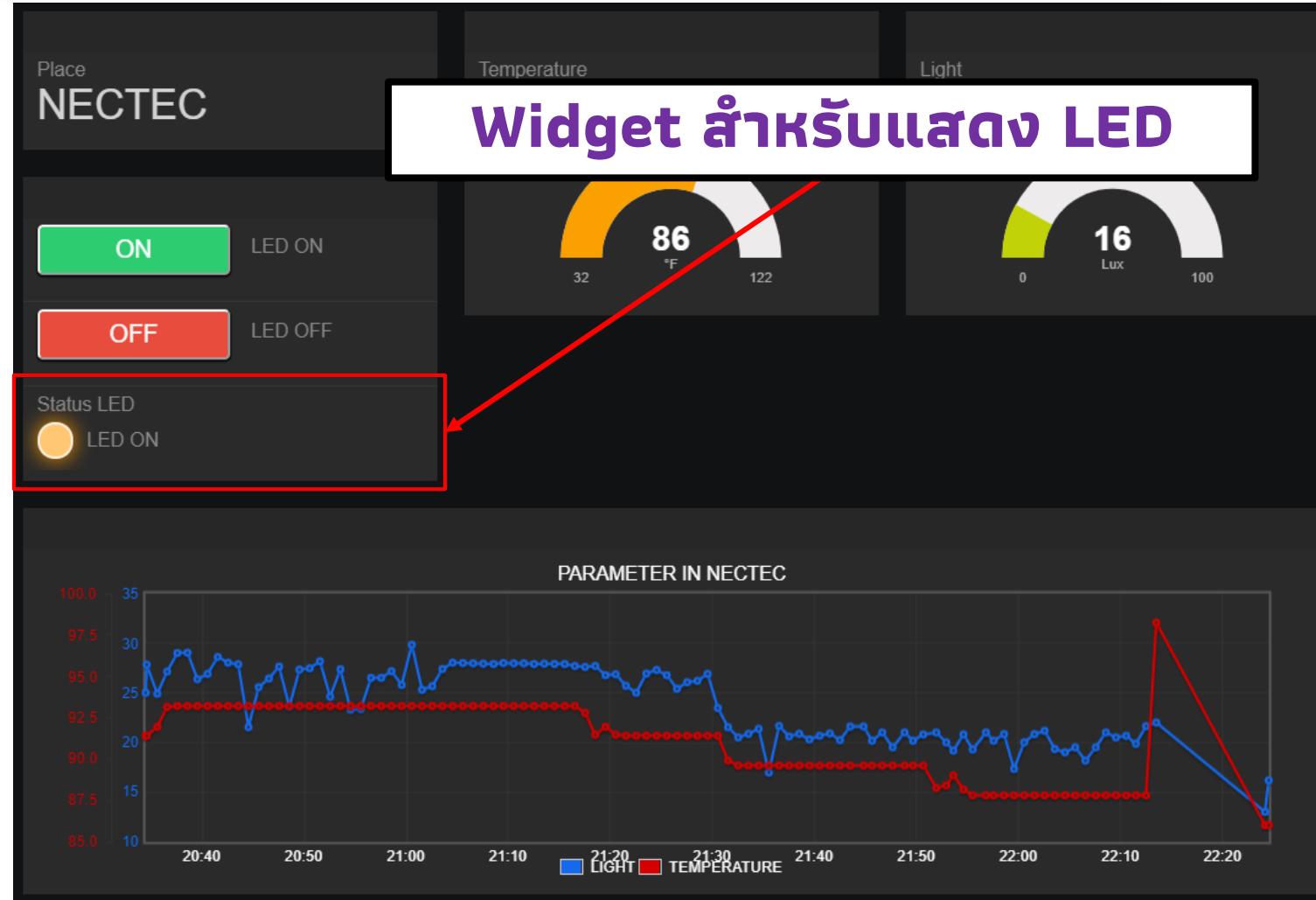


Module 4 : NETPIE 2020 Freeboard



NECTEC
a member of NSTDA

Workshop 7 : การควบคุม LED ผ่าน KidBright ผ่าน Freeboard





Module 5

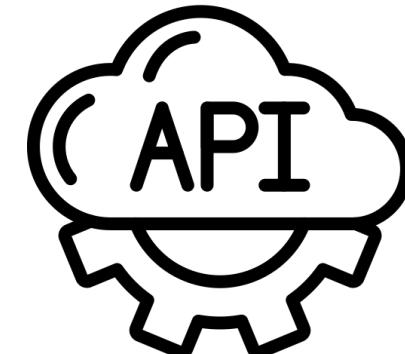
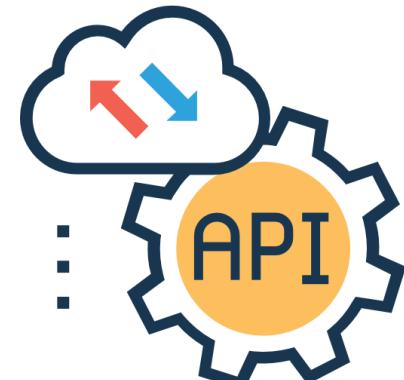
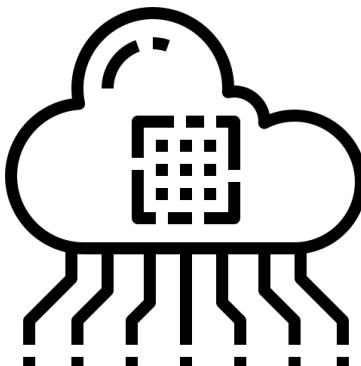
Restful API



What is RESTful API?

เป็นช่องทางสำหรับให้ Device เรียกใช้บริการ Platform ผ่าน RESTful API ซึ่งใช้ HTTP Protocol เมมาระสำหรับใช้เป็นช่องทางในการผสานรวมระบบต่างๆ ทั้งที่มีอยู่แล้วหรือกำลังจะพัฒนาขึ้นมาใหม่โดยไม่จำต้องต้องพัฒนาจากภาษาโปรแกรมใด สำหรับ API แบ่งเป็น 2 กลุ่มคือ

1. Device API
2. Data Store API



Device API

เป็น API ที่เกี่ยวข้องกับการจัดการหรือเรียกใช้ Device โดยมีEndPoint คือ <https://api.netpie.io/v2/device>

▶ Publish ข้อความไปที่ Topic ต่างๆ

```
curl -X PUT "https://api.netpie.io/v2/device/message?topic=%40msg%2Ftest" -H  
"Authorization: Basic ClientID:Token" -H "Content-Type: text/plain" -d "message"
```

▶ อ่านข้อมูล Shadow Data ของ Device

```
curl -X GET "https://api.netpie.io/v2/device/shadow/data" -H "Authorization:  
Basic ClientID:Token"
```

▶ เขียนข้อมูลลง Shadow Data แบบเขียน Merge

```
curl -X PUT "https://api.netpie.io/v2/device/shadow/data" -d "{data:{humid:63.7,  
temp:25.2}}" -H "Authorization: Device ClientID:Token"
```

Device API

▶ เขียนข้อมูลลง Shadow Data และ Overwrite

```
curl -X POST "https://api.netpie.io/v2/shadow/data" -d "{data:{temp:31.7}}" -H  
"Authorization: Device ClientID:Token"
```

▶ การเขียน Shadow แบบระบุเวลาอย้อนหลัง

```
curl -X PUT " https://api.netpie.io/v2/shadow/data" -d "{data:{humid:61.9,  
temp:28.6,timestamp:1566863843262}}" -H "Authorization: Device ClientID:Token"
```

Data Store API

เป็น API ที่เกี่ยวข้องกับการดึงข้อมูลที่เก็บอยู่ใน Timeseries Data โดย Domain name ของ API คือ <https://api.netpie.io/v2/feed> ฐานข้อมูลที่ใช้เก็บอยู่ในรูป KairosDB ลักษณะการ Query ข้อมูล Parameter ต่างๆที่จะส่งไปจะเป็นรูปแบบเดียวกับ KairosDB

อ่านข้อมูลที่เก็บอยู่ใน Timeseries Database ของ Device

```
curl -X POST "https://api.netpie.io/v2/feed/api/v1/datapoints/query" -H "Content-Type: application/json" -H "Authorization: Bearer userToken" -d ' {  
  "start_relative": { "value": 1, "unit": "days" },  
  "metrics": [ { "name": "ClientID", "tags": [ { "attr": "temperature" } ],  
    "limit": 50,  
    "group_by": [ { "name": "tag", "tags": [ "attr" ] } ],  
    "aggregators": [ { "name": "avg", "sampling": { "value": 1, "unit": "minutes" } } ] } }
```

หน้าเว็บสำหรับ Generated Code ในการเรียก API

<https://trial-api.netpie.io/>

การเชื่อมต่อ Device บน NETPIE2020 ผ่านช่องทางการสื่อสารอื่น

หากจำ Workshop3 ได้บื้น การที่จะให้ KidBright 2 เส็บข้อความของ KidBright 1 ได้บื้นจำเป็นต้องสร้าง
Device ขึ้นมาอีก 1 ชุด และทำการจัดให้ออยู่กกลุ่มเดียวกัน
แต่การใช้งานประเภทนี้ต้องเสียจำนวนสิบครรในการใช้งาน Device บน NETPIE2020

Device 1

Client-ID: AA

Token: AA

Secret: AA



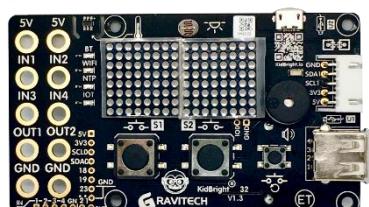
KidBright 1

Device 2

Client-ID: BB

Token: BB

Secret: BB



KidBright 2

MQTT Protocol

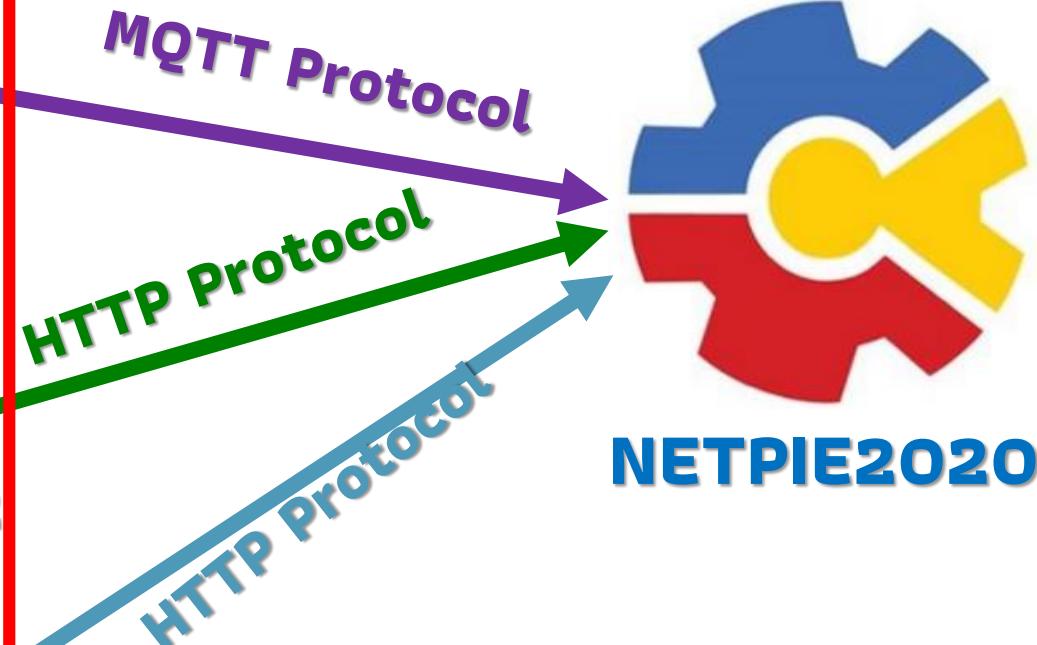
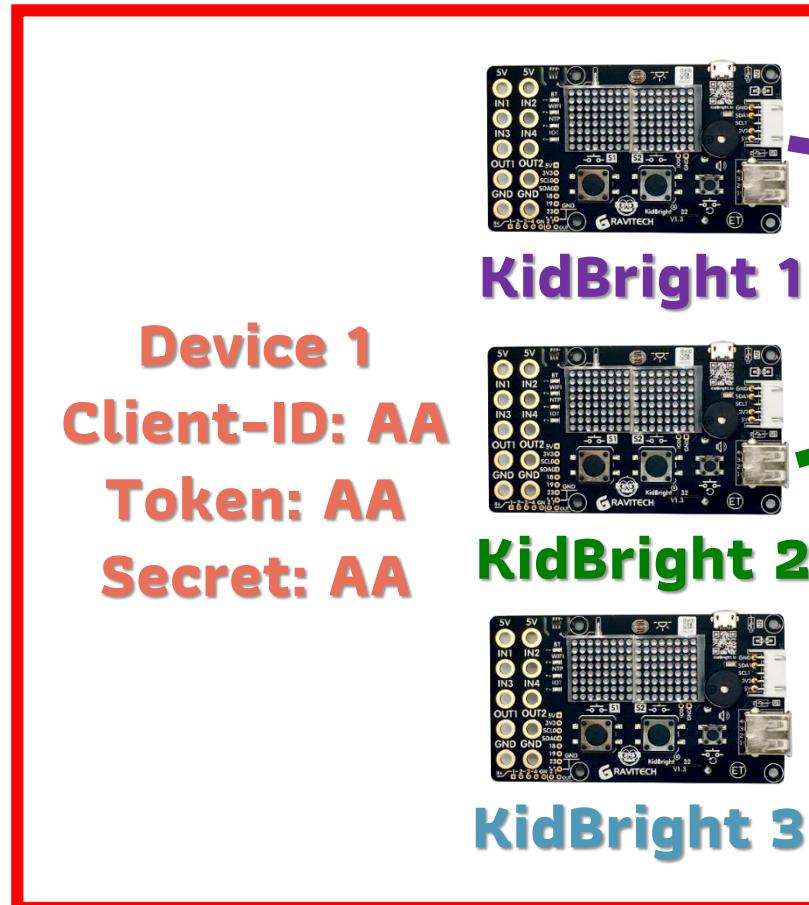
Publish Topic : @msg/test
“Hello NETPIE2020”

Subscribe Topic : @msg/test
“Hello NETPIE2020”



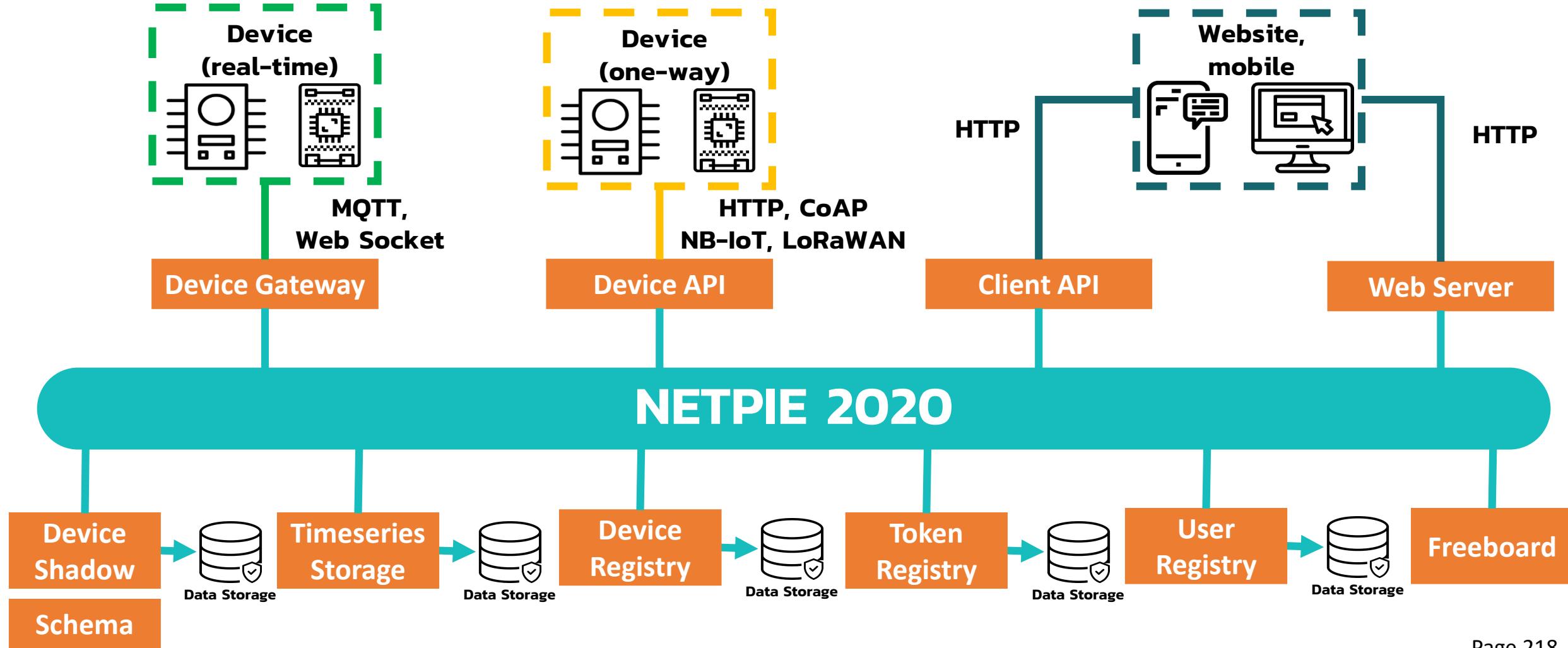
การเชื่อมต่อ Device บน NETPIE2020 ผ่านช่องทางการสื่อสารอื่น

หากต้องการใช้ Device เดิมในการเชื่อมต่อต้องเลือกใช้ Protocol อื่นในการเชื่อมต่อ



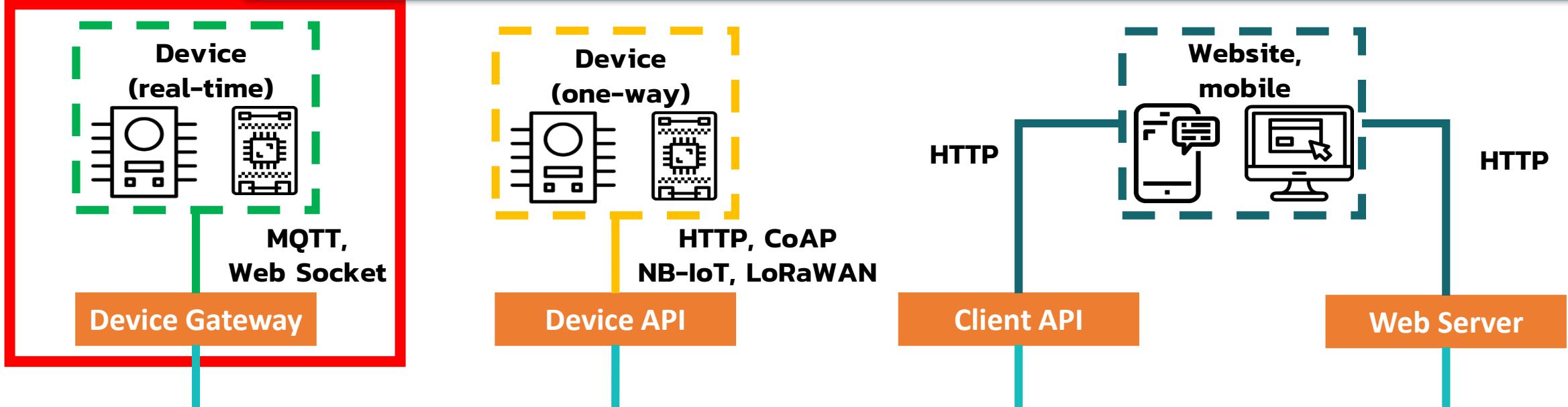
Module 5 : Restful API

NETPIE 2020 Architecture



Module 5 : Restful API

ត្រូវការ Monitoring នៅមូលបែប Real-Time ឱ្យ NETPIE2020 ត្រូវបានគ្រប់ទៅកាន់ការ
សេវាទាំង Device Gateway ដែលត្រូវបានគ្រប់ទៅកាន់ការប្រើប្រាស់ MQTT Protocol នៃការសេវាទាំង



Device
Shadow
Schema



Timeseries
Storage



Device
Registry



Token
Registry



User
Registry



Freeboard



Module 6

Conclusion NETPIE 2020

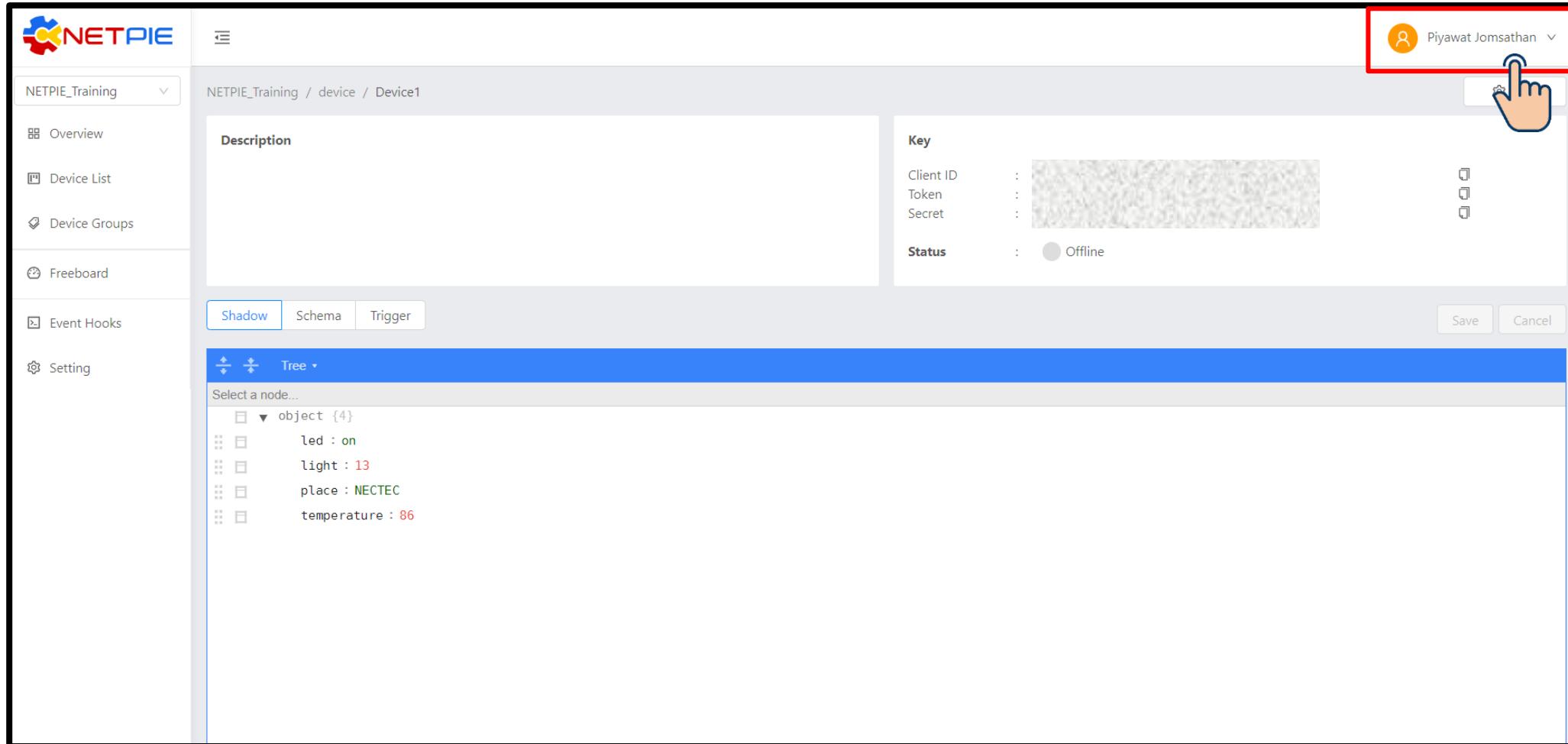


Module 6 : Conclusion NETPIE 2020

NETPIE2020 Quota

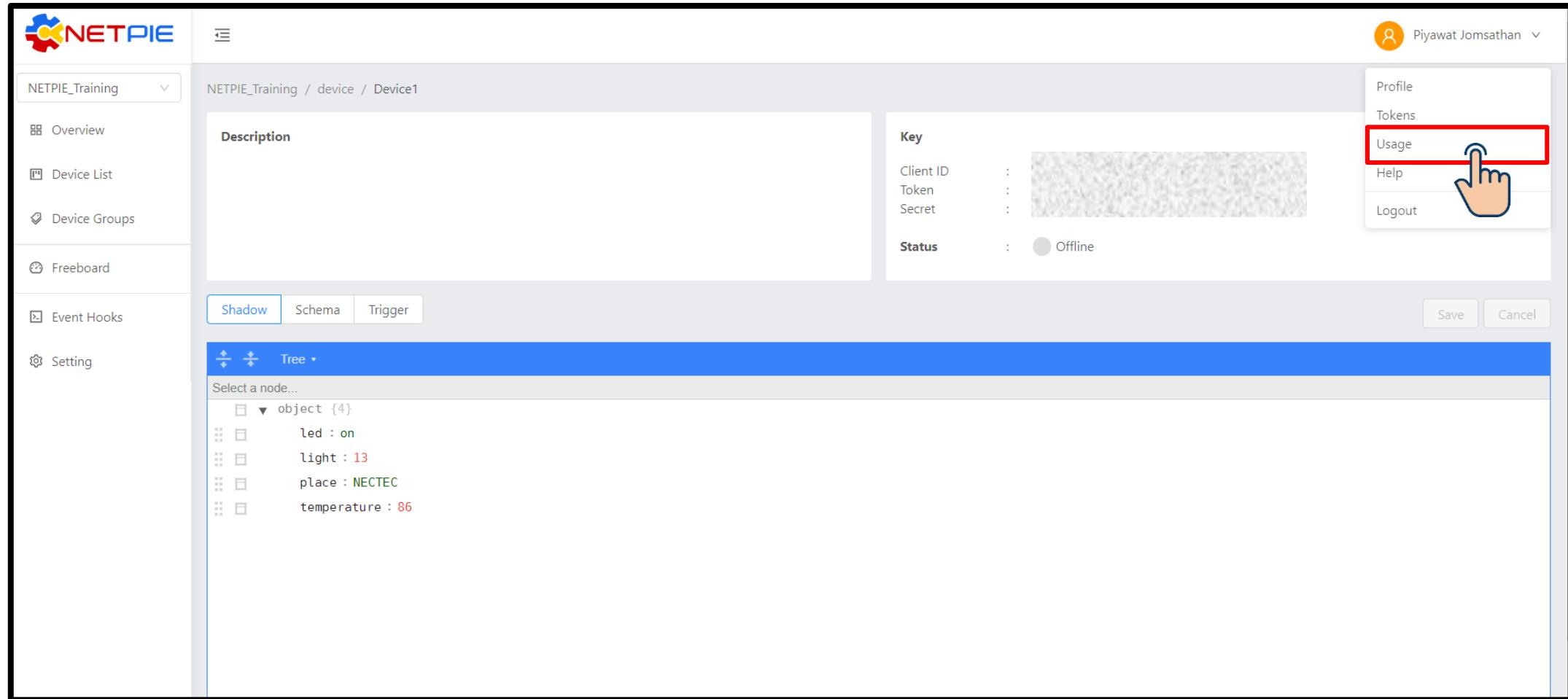
Connecting limit	10 Devices
Project	3 Projects
Real time message	9,000,000 Messages/Month
Data Storage	1,000,000 Points-Month
Shadow read/write	500,000 Operations/Month
API Call	800,000 Operations/Month
Trigger and Action	5,000 Operations/Month
Dashboard	3 Freeboards/Project
Freeboard Connection	3 Concurrent Views

สามารถตรวจสอบยอดการใช้งานบน NETPIE2020 ได้ดังนี้



Module 6 : Conclusion NETPIE 2020

NETPIE2020 Quota



The screenshot shows the NETPIE2020 Quota interface. On the left, there is a sidebar with the following options:

- NETPIE_Training (selected)
- Overview
- Device List
- Device Groups
- Freeboard
- Event Hooks
- Setting

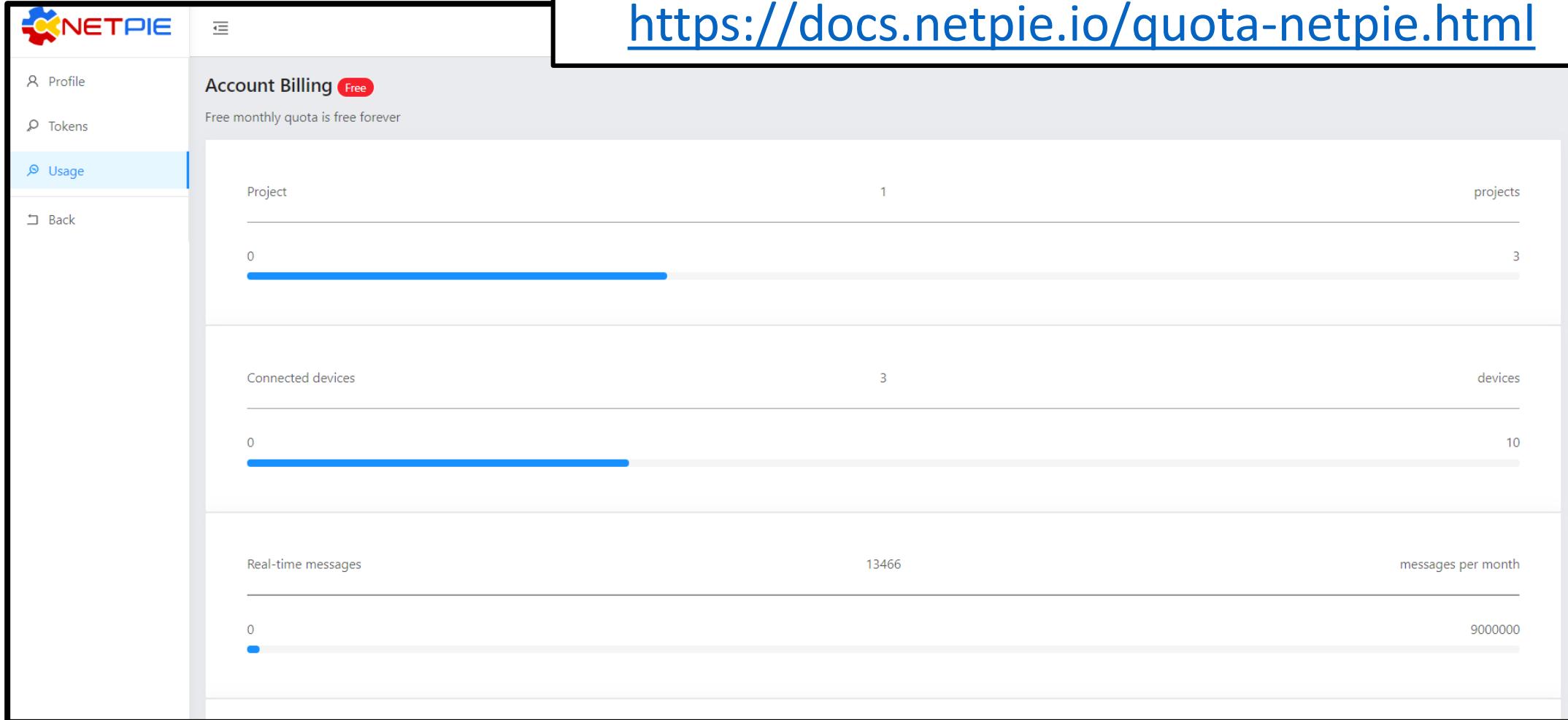
The main area displays a device configuration screen for "NETPIE_Training / device / Device1". The screen includes:

- A "Description" section.
- A "Key" section showing Client ID, Token, Secret, and Status (Offline).
- A "Shadow" tab (selected), which contains a "Tree" view showing the following data structure:

```
object {4}
  led : on
  light : 13
  place : NECTEC
  temperature : 86
```
- A "Save" and "Cancel" button at the bottom right.

In the top right corner, there is a user profile for "Piyawat Jomsathan" with a dropdown menu. The "Usage" option in this menu is highlighted with a red box and has a hand cursor pointing at it.

NETPIE2020 Quota

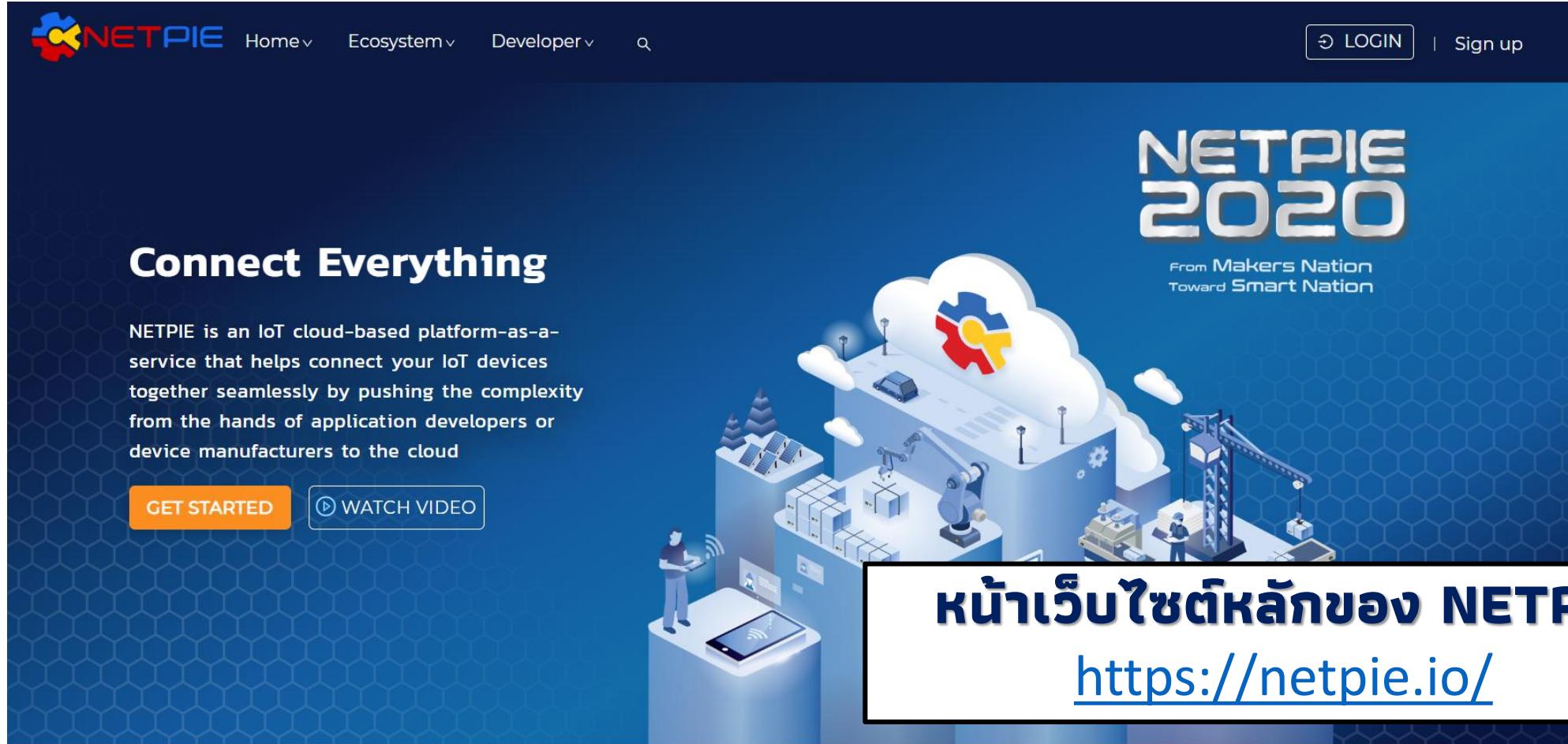


គំរាលការគាំនុញ្ញការឱ្យបង្កើតពីរ

<https://docs.netpie.io/quota-netpie.html>

Module 6 : Conclusion NETPIE 2020

NETPIE Community



The image shows the homepage of the NETPIE 2020 website. The header features the NETPIE logo and navigation links for Home, Ecosystem, Developer, and a search bar. On the right, there's a 'LOGIN' button and a 'Sign up' link. The main banner has a blue hexagonal background with the text 'NETPIE 2020' and 'From Makers Nation Toward Smart Nation'. Below the banner is a large 3D illustration of various IoT devices connected to a central cloud, which contains the NETPIE logo. To the left, a section titled 'Connect Everything' describes NETPIE as an IoT cloud-based platform-as-a-service. It includes two buttons: 'GET STARTED' and 'WATCH VIDEO'.

NETPIE 2020
From Makers Nation
Toward Smart Nation

Connect Everything

NETPIE is an IoT cloud-based platform-as-a-service that helps connect your IoT devices together seamlessly by pushing the complexity from the hands of application developers or device manufacturers to the cloud

GET STARTED | **WATCH VIDEO**

หน้าเว็บไซต์หลักของ NETPIE
<https://netpie.io/>

Module 6 : Conclusion NETPIE 2020



NECTEC
a member of NSTDA

NETPIE Community

The screenshot shows the NETPIE Community website interface. At the top, there is a navigation bar with links for Home, Ecosystem, and Dev. Below the navigation bar, there are three main sections: Group, NodeMCU, and NETPIE2020. The Group section has a link to "เลือกแล้วได้รับความนิยมสำหรับค่า..." (Selected for popularity). The NodeMCU and NETPIE2020 sections have their respective logos.

Hardware

- M5Stack**: ตัวอย่างการเชื่อมต่อและใช้งาน ชาร์ดแวร์ประเภท M5Stack ผ่าน NETPIE2020. It shows a diagram of a M5Stack board connected to a DHT11 sensor and an LED, with data being sent to a NETPIE2020 device.
- NodeMCU**: ตัวอย่างการเชื่อมต่อและใช้งาน ชาร์ดแวร์ประเภท NodeMCU ผ่าน NETPIE2020. It shows a diagram of a NodeMCU board connected to a DHT11 sensor, an LED, and a relay, with data being sent to a NETPIE2020 device via a Freeboard.
- Raspberry Pi**: ตัวอย่างการเชื่อมต่อและใช้งาน ชาร์ดแวร์ประเภท Raspberry Pi ผ่าน NETPIE2020. It shows a diagram of a Raspberry Pi 3B+ board connected to a DHT11 sensor, an LED, and a relay, with data being sent to a NETPIE2020 device via a Device Dashboard.

WebSocket

- A diagram showing a smartphone connected to a NETPIE2020 device via a WebSocket connection.
- A diagram showing a NodeMCU/ESP8266 board connected to a DHT11 sensor, with data being sent to a NETPIE2020 device via a WebSocket connection.

NETPIE Community

The screenshot shows the NETPIE website's 'Example Projects' section. At the top, there is a navigation bar with the NETPIE logo, 'Home', 'Ecosystem', and 'Dev' dropdown menus. Below the navigation, the title 'Example Projects' is displayed, followed by a horizontal line.

The first project listed is 'Smart Home'. It features a diagram illustrating a PIR & LED Control system. The setup includes a NodeMCU ESP8266 connected to a PIR Motion Sensor and a Relay. The Relay is connected to a Light Bulb. A NETPIE 2020 device is shown receiving data from the NodeMCU via MQTT, specifically 'LEDstate : on/off' and 'PIRstate : on/off'. The NETPIE device then triggers a 'Device Shaker' action. Below the diagram, the text reads: "PIR & LED Control" and "‘ตัวอย่างการประยุกต์ใช้งาน NodeMCU กับ NETPIE 2020 สำหรับใช้เป็นระบบ Smart...’".

Below the Smart Home project, there is another project titled 'Smart Farm' which is partially visible.

តัวอย่างการสร้าง Project ประเภทต่างๆโดยใช้ NETPIE
<https://netpie.io/exampleprojects>

Module 6 : Conclusion NETPIE 2020

NETPIE Community



Facebook Group ຂອງ NETPIE

<https://www.facebook.com/groups/netpie/>

Module 6 : Conclusion NETPIE 2020

NETPIE Community



**หากสิทธิในการใช้ NETPIE ไม่เพียงพอ แล้วต้องการอัพเกรด
สามารถทำได้โดยการเปลี่ยนจาก NETPIE เป็น NEXPIE**

<https://nexpie.io/>



NETPIE Community

ราคainการอัพเกรดจาก NETPIE เป็น NEXPIE เบื้องต้น

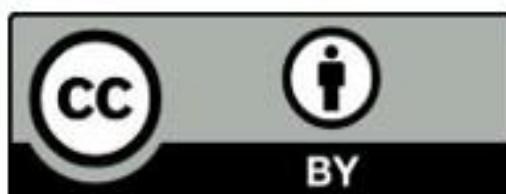
The screenshot shows the NEXPIE IoT Platform pricing page. At the top, there's a navigation bar with the NEXPIE logo, Home, Developer, Pricing, and a CONTACT US button. Below the navigation, the title "ราคา NEXPIE IoT Platform" is displayed. A sub-section title "NEXPIE IoT Platform จะเป็นเครื่องมือทุนแรงช่วยให้องค์กรของคุณพัฒนา Internet of Things (IoT) ได้รวดเร็วและมีเสถียรภาพ" follows. The main content is a comparison chart of three pricing tiers: STARTUP, GROWTH, and SCALE. Each tier has a cloud icon and a price in baht per month. Below the tiers is a table comparing features across the three levels.

	STARTUP	GROWTH	SCALE
	290 บาท/เดือน	2,800 บาท/เดือน	13,000 บาท/เดือน
Connected devices (devices)	10	100	500
Real-time messages (messages per month)	9,000,000	110,000,000	650,000,000
Time-series data storage (point-month)	1,000,00	20,000,000	130,000,000
Shadow read/write (operations per month)	500,000	8,000,000	50,000,000

เอกสารฉบับนี้เป็นลิขสิทธิ์ของศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ

เอกสารฉบับนี้เผยแพร่ภายใต้สัญญาอนุญาตแบบ

Creative Commons Attribution 4.0 International License



อนุญาตให้นำผลงานไปใช้ได้ แก้ไขได้ ดัดแปลงเนื้อหาได้ ใช้ในเชิงพาณิชย์ได้ โดยอ้างอิงที่มา

ดูรายละเอียดสัญญาอนุญาตที่ <https://creativecommons.org/licenses/by/4.0/>



Thank You

