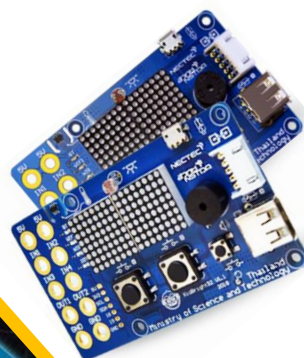




เอกสารและกิจกรรมการเรียนรู้ เรื่อง อินเทอร์เน็ตในทุกสิ่ง

Documents and learning activities in the Internet of Things Concepts



- >> Applications
- >> Freeboard
- >> Feeds

อาจารย์รุ่งเรือง ทศวงศ์ชาย
แผนกอิเล็กทรอนิกส์และแมคคาทรอนิกส์
วิทยาลัยเทคโนโลยีภาคตะวันออก (ชลบุรี)



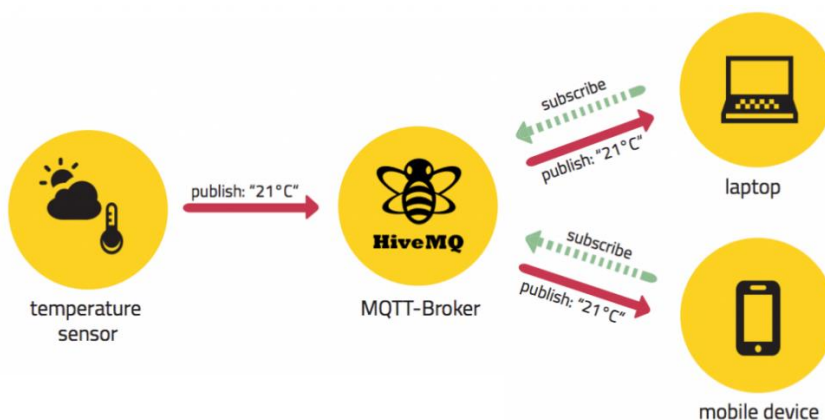
จุดประสงค์

- 1.สามารถเลือกอุปกรณ์สำหรับการทดลอง NETPIE Application ได้
- 2.สามารถเข้าใช้งาน NETPIE ด้วย USER ตัวเองได้
- 3.สามารถใช้งาน MQTT websocket client ได้
- 4.สามารถเชื่อมต่อระหว่าง NETPIE กับบอร์ด KidBright และ MQTT websocket client ได้
- 5.สามารถรับค่าจาก NETPIE เข้าสู่บอร์ด KidBright ผ่าน MQTT websocket client ได้
- 6.สามารถส่งค่าจากบอร์ด KidBright เข้าสู่ NETPIE ผ่าน MQTT websocket client ได้
- 7.สามารถปฏิบัติงานสำเร็จตามกำหนดเวลาได้

ทฤษฎี

1.1 การติดต่อสื่อสารข้อมูลแบบ MQTT

MQTT หมายถึง MQ Telemetry Transport เป็นรูปแบบการสื่อสาร แบบรับ/ส่ง โดยเป็นโปรโตคอลที่ออกแบบมาให้มีขนาดเล็กสำหรับการติดต่อสื่อสาร Machine to Machine ข้อดีคือเป็นโปรโตคอลที่มีขนาดเล็กใช้งานง่าย และ Opensource จึงเหมาะสำหรับการนำไปใช้กับระบบคลาวด์ให้บริการ หรือที่เรียกว่า MQTT Broker

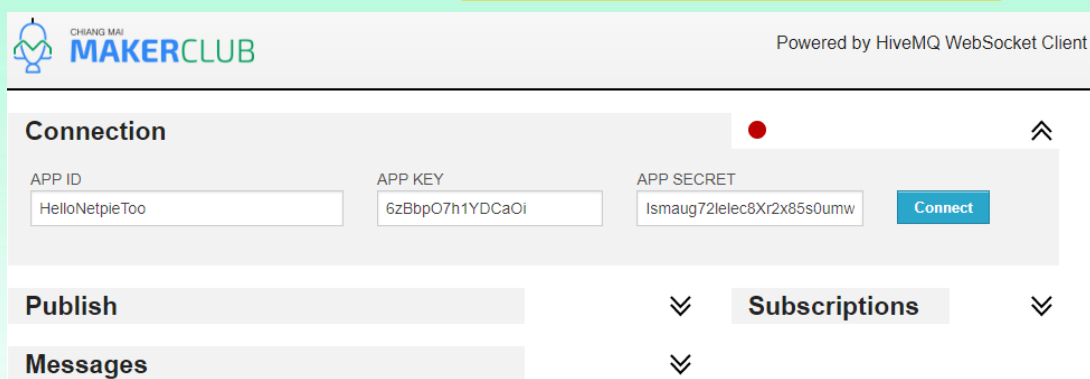


1.2 ศัพท์ควรรู้ใน MQTT

Broker	: จุดศูนย์กลางในการรับส่งข้อมูลระหว่าง Client ผ่านทาง Topic
Topic	: ชื่อช่องทางที่ทำการส่งข้อมูล
Client ID	: ชื่อของอุปกรณ์ เพื่อระบุตัวตน
Publish	: การส่งข้อมูล Topic ผ่าน MQTT
Subscribe	: การรับข้อมูลจาก Topic ที่ได้จาก MQTT
QoS	: คุณภาพ หรือความสำคัญของการส่งข้อมูล
QoS0	: ส่งข้อมูลเพียง 1 ครั้ง ไม่สนว่าจะได้รับหรือไม่ได้รับ
QoS1	: ส่งข้อมูลเพียง 1 ครั้ง ไม่สนว่าจะได้รับหรือไม่ได้รับแต่จะจำข้อมูลที่ส่งไปครั้งล่าสุด เมื่อมีการเชื่อมต่อใหม่จะได้รับข้อมูลที่จำไว้อีกครั้ง
QoS2	: ส่งข้อมูลจนกว่าปลายทางจะได้รับและมีการตรวจสอบการรับส่งข้อมูลตลอดเวลา (ความแม่นยำสูง แต่การส่งข้อมูลช้ากว่า QoS และ QoS1)

1.3 เว็บไซต์สำหรับดูข้อมูลจาก MQTT

<http://cmmc.io/netpie/client>



APP ID	คือ ชื่อของ APPLICATION
APP KEY	คือ รหัสจาก Device Key
APP SECRET	คือ รหัสจาก Device Key

1.4 การใช้งาน Application แพลตฟอร์ม NETPIE.io

NETPIE แพลตฟอร์ม IoT (Network Platform for Internet of Everything) เป็นคลาวด์เซอร์วิสเวอร์ที่ให้บริการในรูปแบบ Platform as a Service สำหรับติดต่อสื่อสารระหว่างอุปกรณ์ต่าง ๆ อาทิเช่น Arduino, Raspberry PI, ESP8266, ESP32, HTML5 และ Node JS เป็นต้น

NETPIE ใช้งานง่ายด้วยไลบรารี Microgear ซึ่งสามารถนำไปใช้กับอุปกรณ์ IoT ต่าง ๆ โดยเป็นการเชื่อมต่อ หรือเข้าสู่บริการอัตโนมัติและยังช่วยลดภาระด้านความปลอดภัยของข้อมูลให้กับผู้ใช้งาน

ข้อควรรู้ การใช้งาน NETPIE ร่วมกับ ESP32 ด้วย Arduino IDE จำเป็นต้องลงไลบรารี Microgear ดูวิธีการติดตั้งที่หัวข้อ 1.6

1.5 การสมัคร และใช้งาน NETPIE.io

- 1) เข้าไปยังเว็บไซต์ netpie.io จากนั้นคลิกที่ SIGN UP FREE

SIGN UP FREE

- 2) กรอกข้อมูลสำหรับการสมัครให้เรียบร้อย แล้วคลิก SIGN UP

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](#)

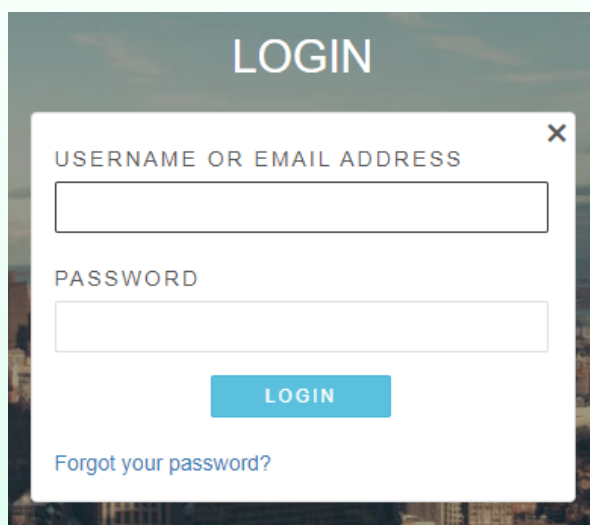
SIGN UP

* Password will be sent to your mobile phone number.

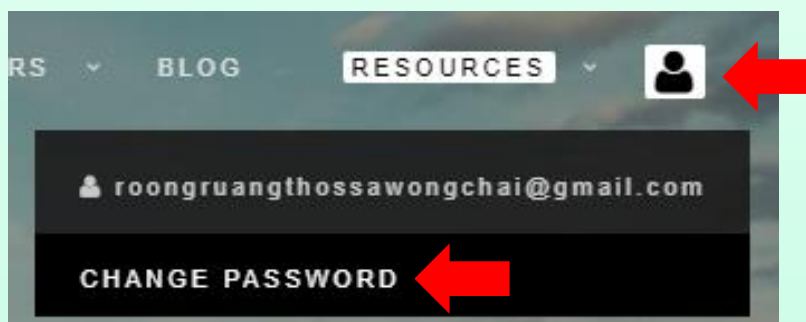
- 3) รอรับข้อความยืนยันในโทรศัพท์จาก NETPIE โดยจะส่งข้อความ เข้ามายังเบอร์โทรศัพท์ที่ได้ลงทะเบียนไว้

Your one-time password for
NETPIE is 685544646

- 4) กดปุ่ม LOG IN เข้าระบบกรอก ID ที่มีครและ Password ที่ได้จากข้อความในโทรศัพท์กด LOG IN

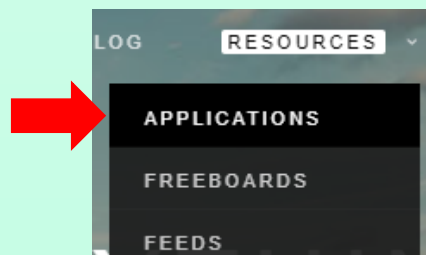


- 5) สามารถ Change Password ที่ icon Profile เพื่อง่ายต่อการเข้าในครั้งต่อไป

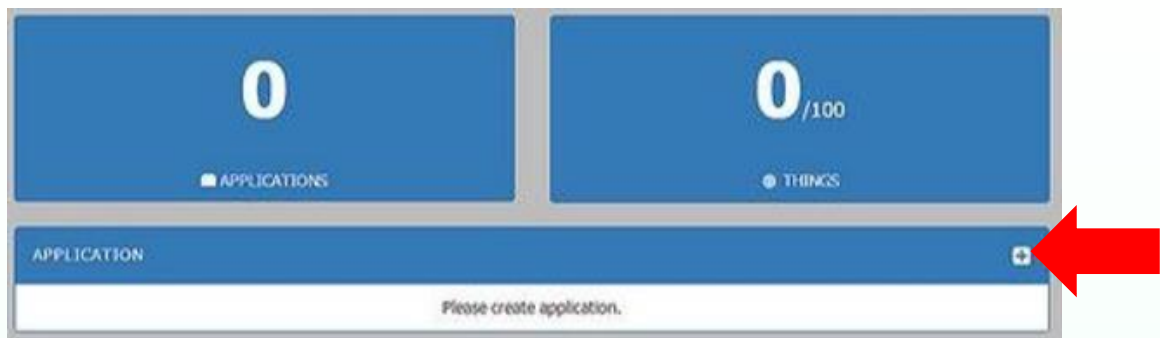


- 6) การเข้าใช้งานเพื่อสร้างห้องให้กับอุปกรณ์สามารถทำได้โดยเลือกแถบ

Resources --> Applications



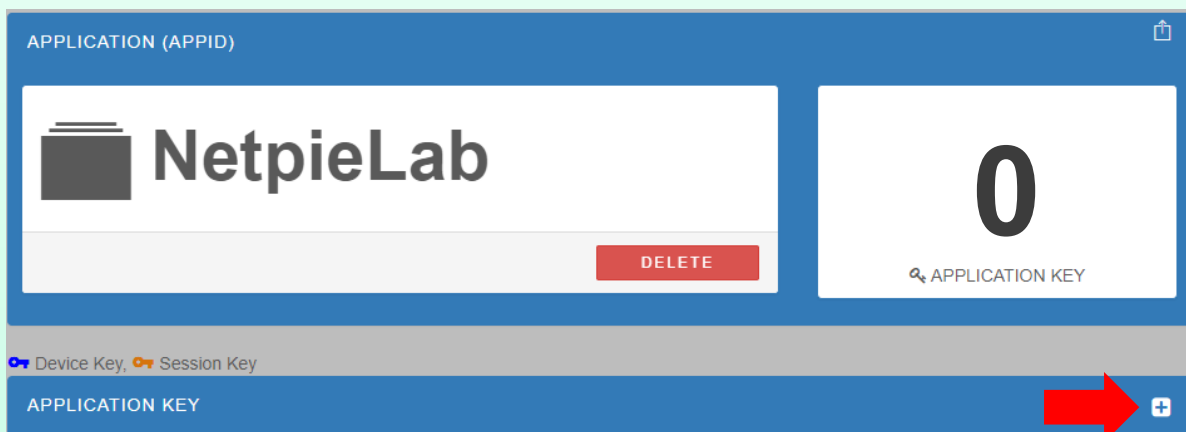
7) คลิกที่เครื่องหมาย + เพื่อสร้าง APPLICATION



8) สร้างชื่อ APPLICATIONS (ชื่อที่สร้างขึ้นมานี้ก็คือ APPID)

A screenshot of the application creation form. It has a text input field containing 'NetpieLab', which is highlighted with a red box. Below the field are two buttons: 'CREATE' (blue) and 'CANCEL' (white with blue border).

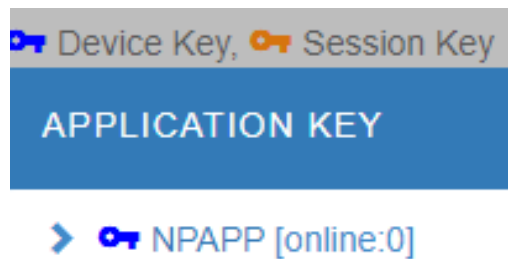
9) แสดงชื่อ APPLICATIONS หรือ APPID ซึ่งตอนนี้ยังไม่มี APPLICATION KEY สร้างโดยการคลิกที่เครื่องหมาย + (มุมล่างขวา)



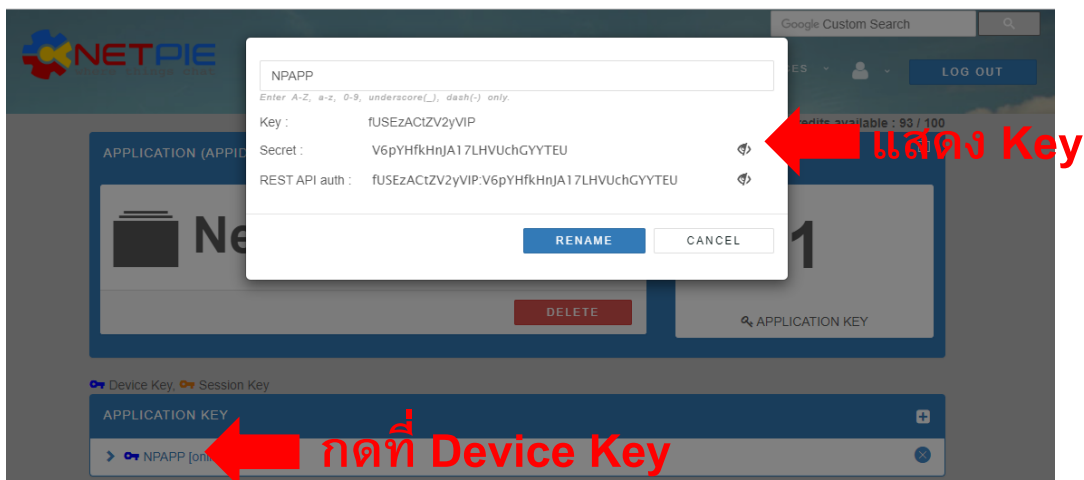
10) ทำการสร้าง Device Key แล้วกด Creat

A screenshot of the Device Key creation form. It has a text input field containing 'NPAPP', which is highlighted with a red box and labeled 'ตั้งชื่อ' (Set name) with a red arrow. Below the field is a dropdown menu with '--- select type ---' and 'Device Key' selected. A red arrow points to the 'Device Key' option, which is labeled 'เลือกชนิด' (Select type). Below the dropdown, it says 'Session Key (Recommended for HTML5 microgears)'.

11) เมื่อสร้างเสร็จจะเห็น Device Key ตามชื่อที่สร้างสีฟ้าด้านล่าง



12) คลิกที่ Device Key จะแสดง KEY และ SECRET เพื่อใช้เชื่อมต่อกับ netpie.io

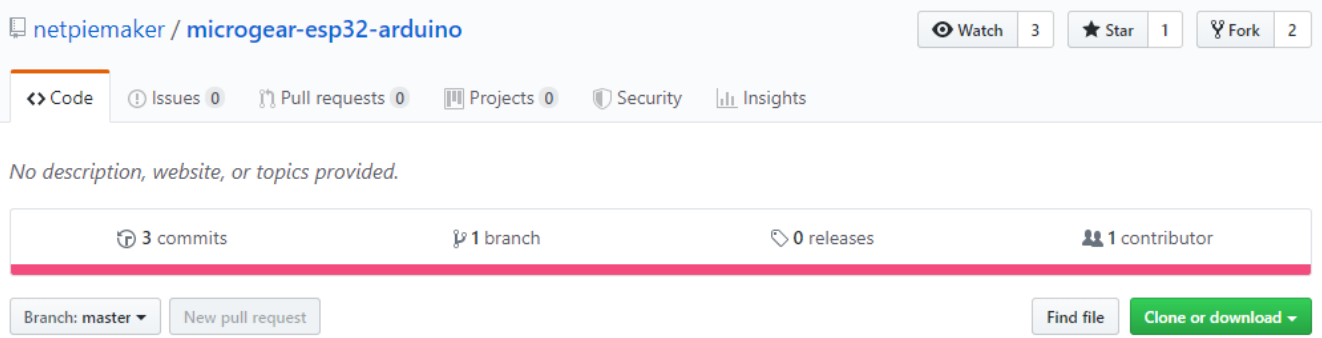


1.6 การติดตั้ง Library NETPIE microgear

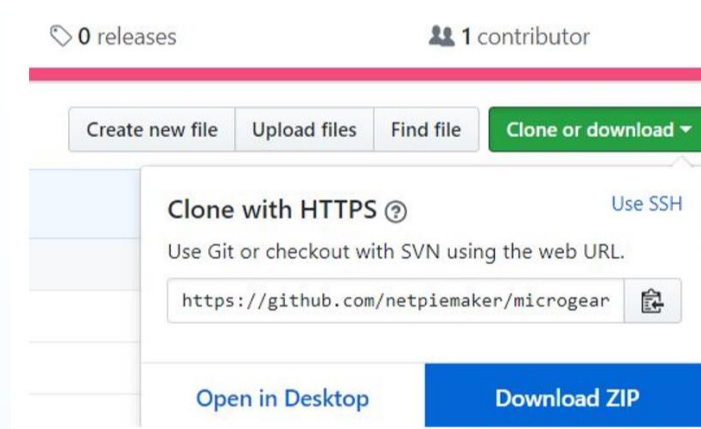
การติดตั้งไลบรารี Microgear สำหรับบอร์ด ESP32 ไม่สามารถติดตั้งผ่าน Manage Libraries ได้ จะต้องทำการดาวน์โหลดไลบรารีเพื่อทำการติดตั้งเอง

1) เข้าไปที่เว็บไซต์

<https://github.com/netpiemaker/microgear-esp32-arduino>

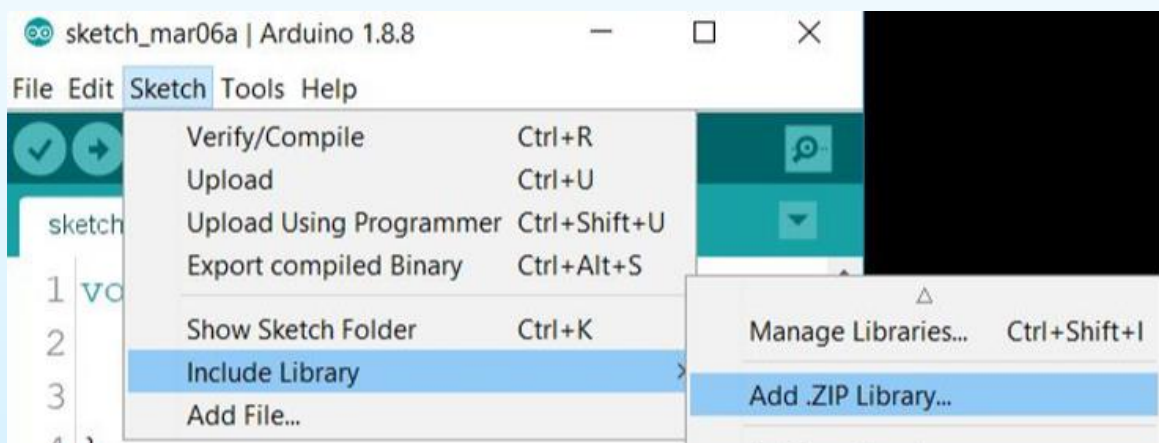


2) คลิก Clone or download เลือก Download ZIP

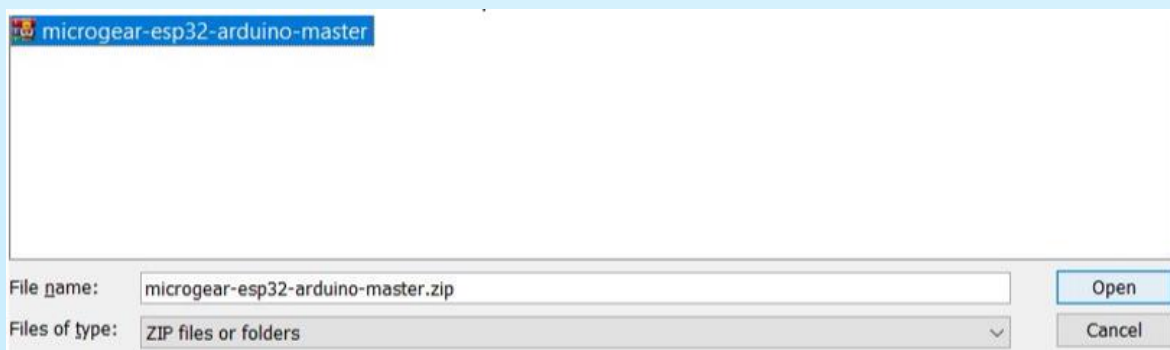


3. เปิดโปรแกรม Arduino IDE เข้าหัวข้อ

Sketch --> Include Library --> Add .ZIP Library...



4) เลือกไลบรารีนามสกุล .ZIP ที่ดาวน์โหลด



ปฏิบัติ

1.7 เครื่องมือและอุปกรณ์การทดลอง

- 1) บอร์ด KidBright พร้อมสาย Micro USB
- 2) คอมพิวเตอร์ที่ติดตั้งโปรแกรม Arduino IDE
- 3) บอร์ด Relay 4 ช่อง
- 4) สายจัม

1.8 ขั้นตอนการทดลอง

- 1) สร้าง APPLICATION NETPIE
- 2) เชื่อมต่อ ESP32 กับ netpie.io
 - เปิดโปรแกรม NPAPP.ino ทำการแก้ไขโปรแกรกดังนี้

NPApp

```
1 #include "MicroGear.h"
2 #include "WiFi.h"
3
4 const char* ssid      = "XXXXXXXXXX"; ← ชื่อไวไฟที่เชื่อมต่อ
5 const char* password = "XXXXXXXXXX"; ← รหัสไวไฟที่เชื่อมต่อ
6
7 #define APPID "XXXXXXXXXX" ← APPID จากเว็บ netpie
8 #define KEY "XXXXXXXXXX" ← Device Key เว็บ netpie
9 #define SECRET "XXXXXXXXXX" ← SECRET จาก Device Key
10 #define ALIAS "XXXXXXXXXX" ← ALIAS (ตั้งชื่ออุปกรณ์)
11
12 #define led 17 ← ตัวแปร led ใช้งาน pin 17 (หลอด LED BT)
13
14 WiFiClient client;
15 int timer = 0;
16 MicroGear microgear(client);
17
```

```

18 void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen)
19 {
20     Serial.print("Incoming message --> ");
21     msg[msglen] = '\0';
22     Serial.println((char *)msg);
23
24     char strMsg[msglen];
25     for (int i = 0; i < msglen; i++)
26     {
27         strMsg[i] = (char)msg[i];
28     }
29
30     String msgStr = String(strMsg).substring(0, msglen);
31     String strTopic = (char *)topic;
32
33     Serial.print("Topic: ");
34     Serial.println(strTopic);
35     Serial.print("Msg: ");
36     Serial.println(msgStr);
37
38     if (msgStr == "ON")
39     {
40         Serial.println("ON");
41         digitalWrite(led, 0);
42     }
43
44     if (msgStr == "OFF")
45     {
46         Serial.println("OFF");
47         digitalWrite(led, 1);
48     }
49 }
50
51 void onFoundgear(char *attribute, uint8_t* msg, unsigned int msglen)
52 {
53     Serial.print("Found new member --> ");
54     for (int i = 0; i < msglen; i++)
55         Serial.print((char)msg[i]);
56     Serial.println();
57 }
58

```

← ประมวลผลตัวอักษรที่ได้รับ

```

59 void onLostgear(char *attribute, uint8_t* msg, unsigned int msglen)
60 {
61     Serial.print("Lost member --> ");
62     for (int i = 0; i < msglen; i++)
63         Serial.print((char)msg[i]);
64     Serial.println();
65 }
66
67 void onConnected(char *attribute, uint8_t* msg, unsigned int msglen)
68 {
69     Serial.println("Connected to NETPIE...");
70     microgear.setAlias(ALIAS);
71 }
72
73
74 void setup()
75 {
76     microgear.on(MESSAGE, onMsghandler);
77     microgear.on(PRESENT, onFoundgear);
78     microgear.on(ABSENT, onLostgear);
79     microgear.on(CONNECTED, onConnected);
80
81     Serial.begin(115200);
82     Serial.println("Starting...");
83
84     pinMode(led, OUTPUT);
85     digitalWrite(led, HIGH);
86
87     if (WiFi.begin(ssid, password))
88     {
89         while (WiFi.status() != WL_CONNECTED)
90         {
91             delay(500);
92             Serial.print(".");
93         }
94     }
95
96     Serial.println("WiFi connected");
97     Serial.println("IP address: ");
98     Serial.println(WiFi.localIP());
99
100     microgear.init(KEY, SECRET, ALIAS);
101

```

```

102 microgear.connect(APPID);
103 microgear.subscribe("/gearname/XXXXXXXXX/data/$/command");
104 }
105
106 void loop()
107 {
108   if (microgear.connected())
109   {
110     microgear.loop();
111
112     if (timer >= 5000)
113     {
114       Serial.println("Publish...");
115
116       microgear.chat(ALIAS, "Hello");
117       microgear.publish("/gearname/XXXXXXXXX/light", analogRead(36));
118       timer = 0;
119     }
120     else timer += 100;
121   }
122   else
123   {
124     Serial.println("connection lost, reconnect...");
125     if (timer >= 5000)
126     {
127       microgear.connect(APPID);
128       timer = 0;
129     }
130     else timer += 100;
131   }
132   delay(100);
133 }

```

ใส่ชื่ออุปกรณ์ ALIAS

ใส่ชื่ออุปกรณ์ ALIAS

- ทำการอัฟโหลดแล้วเปิด Serial Monitor จะแสดงข้อความการเชื่อมต่อ

```

COM26
Starting...
..WiFi connected
IP address:
192.168.12.116
Connected to NETPIE...
Publish...
Incoming message --> Hello

```

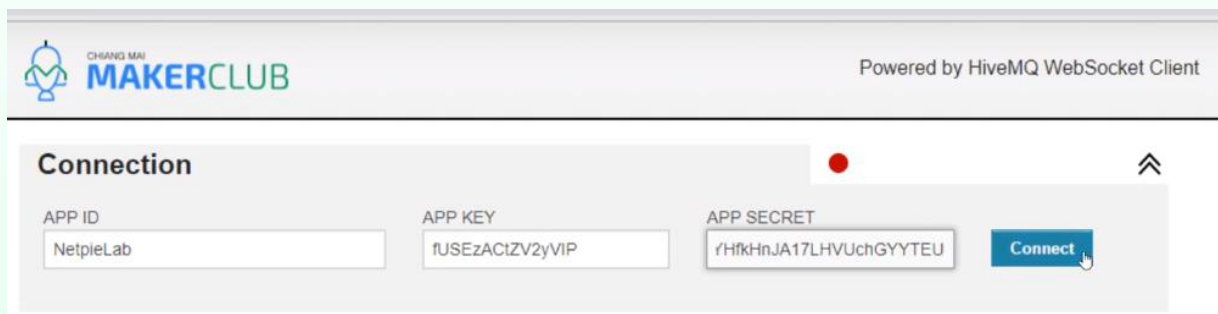
- แสดงการเชื่อมต่อในเว็บ netpie.io




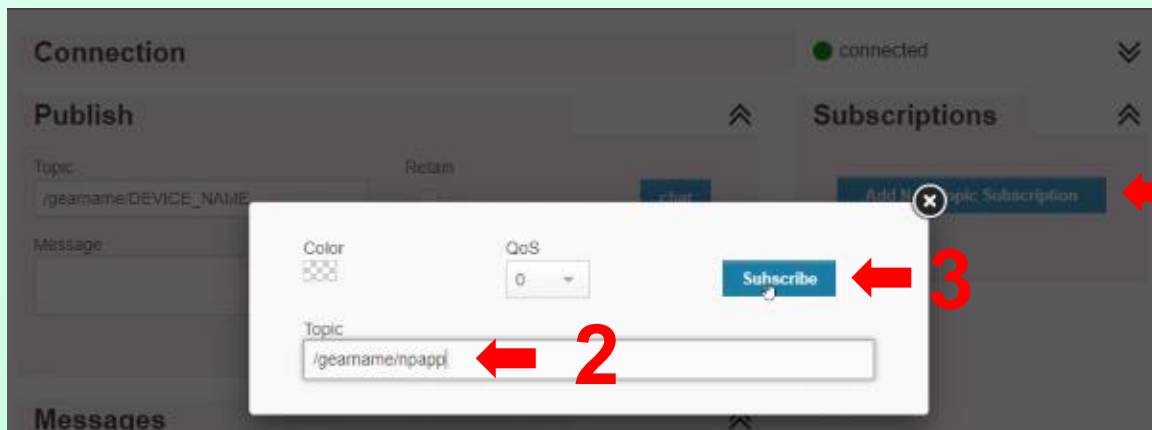
 npapp

3) แสดงพล mqtt ด้วย websocket client

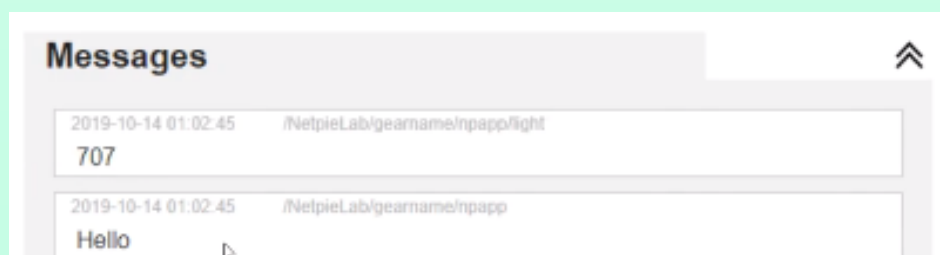
- เปิดเบราว์เซอร์ เข้าเว็บ <http://cmmc.io/netpie/client/>
- แก้ไข APP ID, APP KEY, APP SECRET ของตนเอง กด Connect



- เมื่อเชื่อมต่อสำเร็จ จะแสดงคำว่า connected มุมบนขวา 
- คลิก Add New Topic Subscription เพื่อทำการเรียกดูข้อมูลที่ Topic : /gearname/#



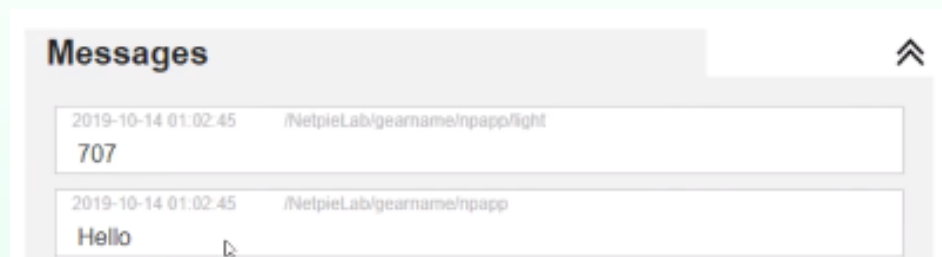
- แสดงข้อมูลที่ส่งขึ้นมายัง netpie.io



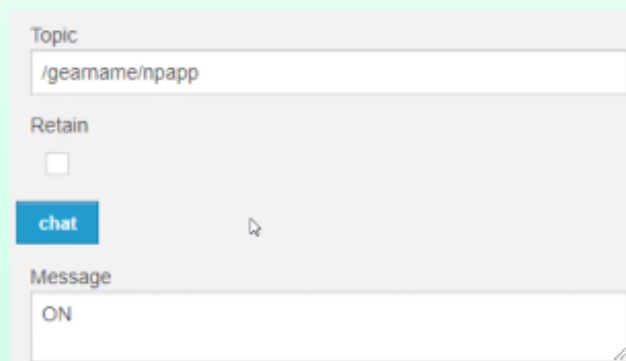
- ใช้งานคำสั่ง microgear.publish โดยมีรูปแบบดังนี้
microgear.publish("/gearname/ALIAS/name", Value);
ALIAS คือ ชื่อของอุปกรณ์
Name คือ ชื่อของข้อมูลที่ต้องการส่งขึ้นไป
Value คือ ข้อมูล

```
116 | microgear.chat(ALIAS, "Hello");
117 | microgear.publish("/gearname/npapp/light", analogRead(36));
```

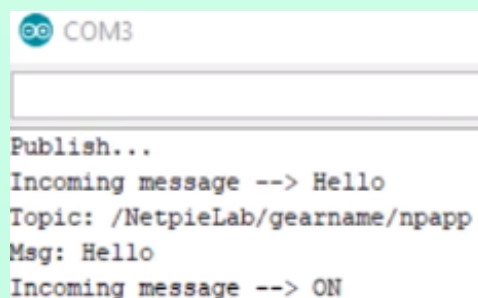
- จากข้างบนจะส่งค่าระดับความสว่างของ LDR ที่อ่านได้จากขา 36



- ส่งข้อมูลไปยังบอร์ด โดยกำหนด
Topic: /gearname/npapp
Message: ON

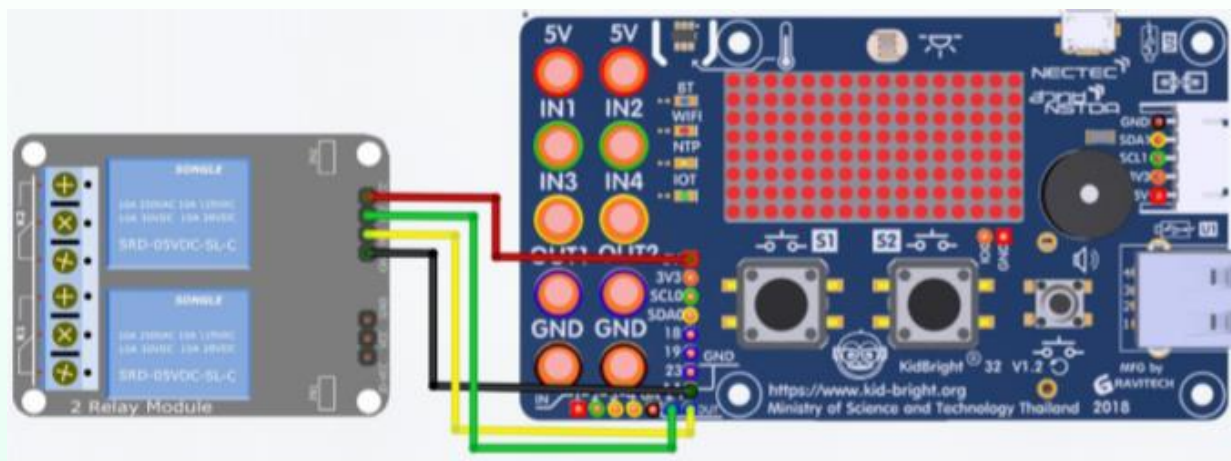


- ช่อง Message จะแสดงข้อความที่ส่งไปยังบอร์ด
- แสดงข้อความที่ส่งมาจากเว็บ ผ่าน Serial Monitor



1.9 โจทย์ฝึกปฏิบัติเพิ่มทักษะ

- 1) ประกอบวงจรและเขียนโปรแกรมรับข้อมูลควบคุม Relay โดยมีเงื่อนไขดังต่อไปนี้
 - รับข้อมูลจาก Topic: /gearname/npapp
 - เมื่อได้รับข้อมูล "ONE" ให้เปิดใช้งาน Relay1 ปิดการใช้งาน Relay2
 - เมื่อได้รับข้อมูล "TWO" ให้เปิดใช้งาน Relay2 ปิดการใช้งาน Relay1



- 2) ประกอบวงจรและเขียนโปรแกรมรับข้อมูลควบคุม Relay โดยมีเงื่อนไขดังต่อไปนี้ (ใช้โมดูล Relay 4 CH)
 - รับข้อมูลจาก Topic: /gearname/npapp
 - เมื่อได้รับข้อมูล "1" ให้เปิดใช้งาน Relay1 ปิดการใช้งาน Relay ทุกตัว
 - เมื่อได้รับข้อมูล "2" ให้เปิดใช้งาน Relay2 ปิดการใช้งาน Relay ทุกตัว
 - เมื่อได้รับข้อมูล "3" ให้เปิดใช้งาน Relay3 ปิดการใช้งาน Relay ทุกตัว
 - เมื่อได้รับข้อมูล "4" ให้เปิดใช้งาน Relay4 ปิดการใช้งาน Relay ทุกตัว

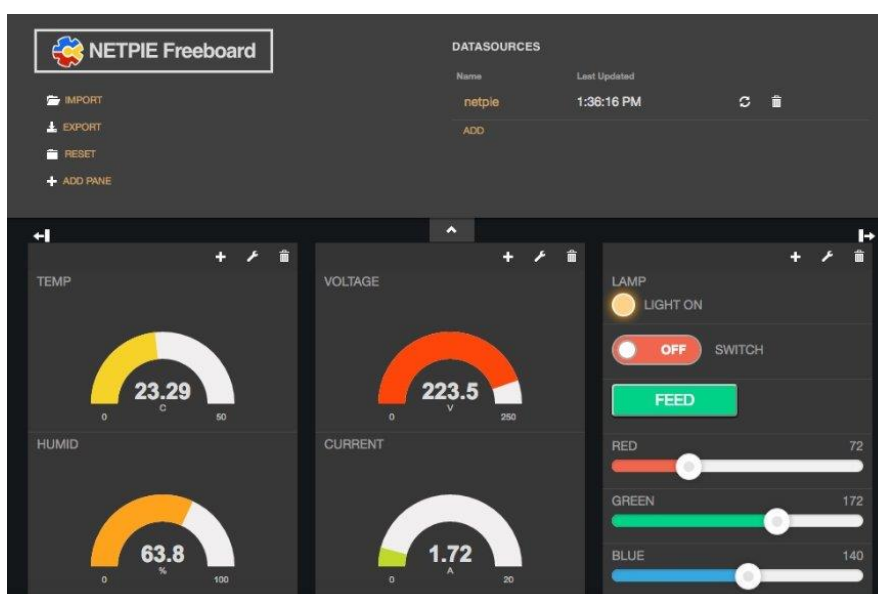
จุดประสงค์

- 1.สามารถเลือกใช้อุปกรณ์สำหรับการทดลอง NETPIE Freeboards ได้
- 2.สามารถเข้าใช้งาน NETPIE Freeboards ได้
- 3.สามารถเลือกใช้งานและตั้งค่า Widget NETPIE Freeboards ได้
- 4.สามารถแก้ไขโปรแกรมให้บอร์ด KidBright เชื่อมต่อกับ NETPIE ได้
- 5.สามารถแก้ไขโปรแกรมให้ส่งค่าจากบอร์ด KidBright แสดงผลที่หน้า Freeboards ได้
- 6.สามารถปฏิบัติงานสำเร็จตามกำหนดเวลาได้

ทฤษฎี

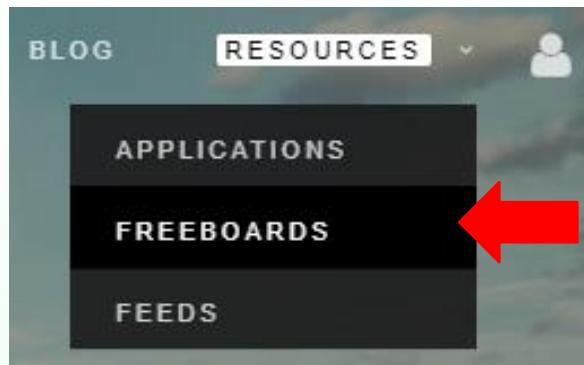
2.1 NETPIE Freeboards

NETPIE Freeboard เป็น Web Application สามารถสร้างหน้าแสดงผลข้อมูลที่เชื่อมต่อเข้ากับ netpie.io ได้ง่าย โดยที่นักพัฒนาไม่จำเป็นต้องเขียนโปรแกรมสร้างเอง มี Widget Plugins ที่สามารถรองรับการใช้งาน เช่น บุ่มกดสวิตช์ เทจวัดกราฟเส้นรองรับการติดต่อสื่อสารแบบ Real-Time นำไปใช้งานได้หลากหลายรูปแบบ

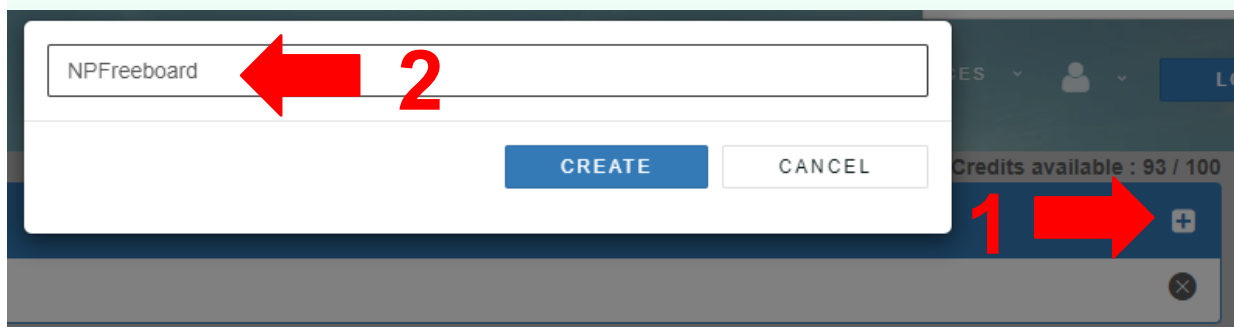


2.2 การใช้งาน FreeBoard

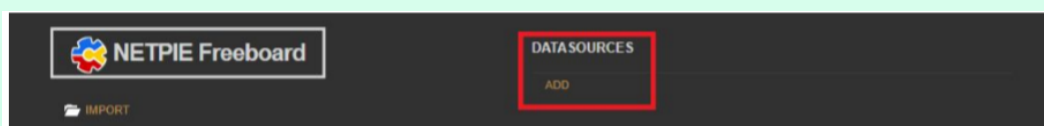
- 1) ลือตอินเว็บ NETPIE.io เลือทหั่วข้อ RESOURCES
- 2) ทดเลือท FREEBOARDS



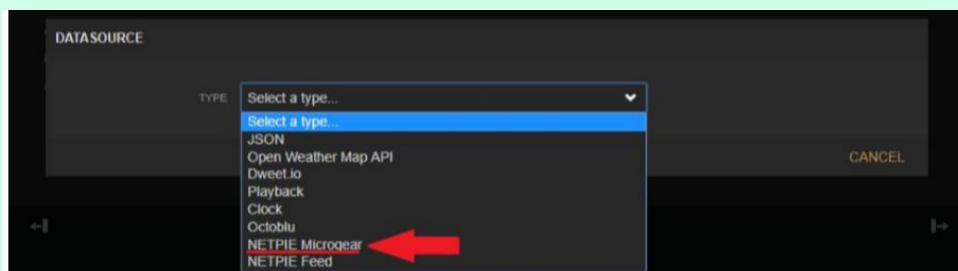
- 3) สร้างหน้าแสดงพลโดยทดที่เครื่องหมายบวก (+) ทำการตั้งชื่อ Freeboard



- 4) เมื่อคลิกที่ Freeboard ที่สร้าง จะปรากฏหน้าแสดงพลคลิก ADD เพื่อเพิ่ม DATASOURCES



- 5) เลือท TYPE เป็น NETPIE Microgear



6) หน้าต่างสำหรับตั้งค่าเชื่อมต่อ NETPIE

DATASOURCE

Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID. The microgear of this datasource is referenced by microgear[DATASOURCENAME]

TYPE	NETPIE Microgear
NAME	
APP ID	
	NETPIE App ID obtained from https://netpie.io/app
KEY	
	Key
SECRET	
	Secret
SUBSCRIBED TOPICS	/#
	Topics of the messages that this datasource will consume. the default is /# which means all messages in this app ID.
ONCREATED ACTION	
	JS code to run after a datasource is created
ONCONNECTED ACTION	
	JS code to run after a microgear datasource is connected to NETPIE

7) ทำการแก้ไข DATASOURCE เพื่อทำการเชื่อมกับ NETPIE

TYPE คือ รูปแบบการเชื่อมต่อ

NAME คือ ชื่อของ Freeboard

APPID คือ ชื่อ APPLICATIONS NAME

KEY คือ รหัสกุญแจ สำหรับเชื่อมต่อ

SECRET คือ รหัสลับสำหรับเชื่อมต่อ



SUBSCRIBED TOPICS คือ ชื่อ TOPICS ที่ต้องการ SUBSCRIBED

Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID. The microgear of this datasource is referenced by microgear[DATASOURCENAME]

TYPE	NETPIE Microgear
NAME	NPFreeboard
APP ID	NetpieLab <small>NETPIE App ID obtained from https://netpie.io/app</small>
KEY	fUSEzActZV2yVIP <small>Key</small>
SECRET	V6pYHfkHnJA17LHVUchGYYTEU <small>Secret</small>
SUBSCRIBED TOPICS	/# <small>Topics of the messages that this datasource will consume, the default is /# which means all messages in this app ID.</small>
ONCREATED ACTION	 <small>JS code to run after a datasource is created</small>
ONCONNECTED ACTION	 <small>JS code to run after a microgear datasource is connected to NETPIE</small>

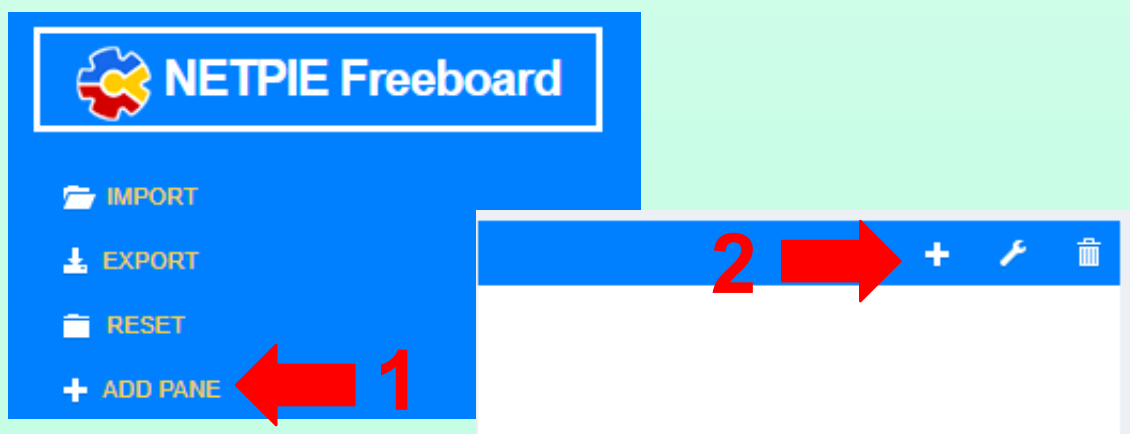
SAVE
CANCEL

8) เมื่อเชื่อมต่อสำเร็จ จะแสดงเวลาอัปเดตล่าสุด

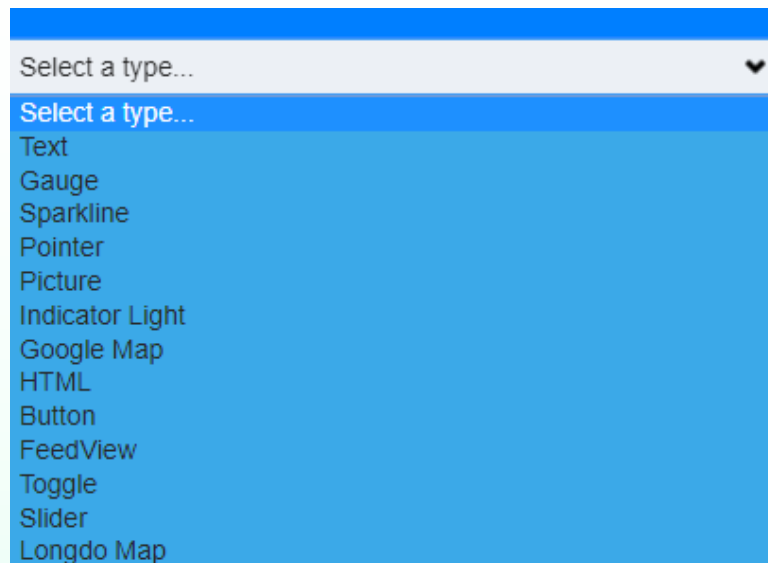
DATASOURCES		
Name	Last Updated	
NPFreeboard	9:57:30 AM	 

9) กด ADD PANE เพื่อสร้างส่วนแสดงผล

10) กดเครื่องหมายบวก (+) ใน PANE ที่สร้าง เพื่อเลือกรูปแบบการแสดงผล



11) วิดเจ็ต (WIDGET) ของ Freeboard มีให้เลือกใช้งานได้หลากหลาย ซึ่งมีดังต่อไปนี้



Text

TITLE	คือ ชื่อข้อความ
SIZE	คือ ขนาดของข้อความ
VALUE	คือ รับข้อมูลแสดงผล
UNITS	คือ หน่วยข้อมูล

WIDGET

TYPE: Text

TITLE:

SIZE: Regular

VALUE: [+ DATASOURCE](#) [JS EDITOR](#)

INCLUDE SPARKLINE: ☐ NO

ANIMATE VALUE CHANGES: ☒ YES

UNITS:

[SAVE](#) [CANCEL](#)

Gauge

TITLE	คือ ชื่อเกจ
VALUE	คือ รับข้อมูลแสดงผล
UNITS	คือ หน่วยข้อมูล
MINIMUM	คือ ค่าต่ำสุดของเกจ
MAXIMUM	คือ ค่าสูงสุดของเกจ

WIDGET

TYPE

Gauge

TITLE

VALUE

+ DATASOURCE

JS EDITOR

UNITS

MINIMUM

0

MAXIMUM

100

SAVE

CANCEL

Sparkline

TITLE	คือ ชื่อของกราฟเส้น
VALUE	คือ รับข้อมูลแสดงผล
LEGEND	คือ จุดเริ่มต้น

WIDGET

TYPE

Sparkline

TITLE

VALUE

+ DATASOURCE

JS EDITOR

+ ADD

INCLUDE LEGEND

☐

NO

LEGEND

Comma-separated for multiple sparklines

SAVE

CANCEL

Pointer

DIRECTION

คือ รับข้อมูลเพื่อแสดงทิศทาง

VALUE TEXT

คือ รับข้อมูลเพื่อแสดงค่าข้อมูล

UNITS

คือ หน่วยข้อมูล

WIDGET

TYPE

Pointer

DIRECTION

+ DATASOURCE

JS EDITOR

In degrees

VALUE TEXT

+ DATASOURCE

JS EDITOR

UNITS

SAVE

CANCEL

Picture

IMAGE URL

คือ ลิงก์รูปภาพ

REFRESH EVERY

คือ กำหนดเวลาโหลดรูปใหม่

WIDGET

TYPE

Picture

IMAGE URL

+ DATASOURCE

JS EDITOR

REFRESH EVERY

SECONDS

Leave blank if the image doesn't need to be refreshed

SAVE

CANCEL

Indicator Light

TITLE

คือ ชื่อของหลอดไฟ

VALUE

คือ รับข้อมูลแสดงผล

ON TEXT

คือ แสดงข้อความเมื่อมีการเปิด

OFF TEXT

คือ แสดงข้อความเมื่อมีการปิด

WIDGET

TYPEIndicator Light

TITLE

VALUE

+ DATASOURCE

JS EDITOR

ON TEXT

+ DATASOURCE

JS EDITOR

OFF TEXT

+ DATASOURCE

JS EDITOR

SAVECANCEL

Google Map

LATITUDE

คือ เส้นละติจูด

LONGITUDE

คือ เส้นลองจิจูด

WIDGET

TYPEGoogle Map

LATITUDE

+ DATASOURCE

JS EDITOR

LONGITUDE

+ DATASOURCE

JS EDITOR

SAVECANCEL

HTML

HTML

คือ โปรแกรมด้วยภาษา HTML

HEIGHT BLOCKS

คือ ขนาดของบล็อกแสดงผล

WIDGET

TYPEHTML

HTML

+ DATASOURCE

JS EDITOR

Can be literal HTML, or javascript that outputs HTML

HEIGHT BLOCKS4

A height block is around 60 pixels

SAVECANCEL

Button

BUTTON CAPTION	คือ ชื่อแสดงบนปุ่มกด
LABEL TEXT	คือ ข้อความของปุ่มกด
BUTTON COLOR	คือ สีของปุ่มกด
ONCLICK ACTION	คือ ส่งข้อมูลเมื่อมีการกดปุ่มกด

WIDGET

A simple button widget that can perform Javascript action.

TYPE

Button

BUTTON CAPTION

LABEL TEXT

BUTTON COLOR

Red

ONCLICK ACTION

+ DATASOURCE

JS EDITOR

Add some Javascript here. You can chat and publish with a datasource's microgear like this :
microgear["mygear"].chat("mylamp","ON"), where "mygear" is a datasource name.

ONCREATED ACTION

JS code to run after a button is created

SAVE

CANCEL

FeedView

TITLE	คือ ชื่อฟีด
DATA SOURCE	คือ แหล่งข้อมูลที่ตั้งมาแสดงผล
FILTER	คือ ตัวกรองข้อมูลที่จะแสดงผล
TYPE OF CHART	คือ ชนิดของกราฟแบบ Line และแบบ STEP
X AXIS TITLE	คือ ข้อความกำหนดบนแกน X
Y AXIS TITLE	คือ ข้อความกำหนดบนแกน Y
BEGIN AT 0	คือ ตั้งค่าแกน Y ที่ตำแหน่ง 0
LINE COLORS	คือ สีเส้นกราฟ (ค่าสี HTML)
MARKER	คือ วงกลมที่ตำแหน่งจุดข้อมูล
MULTIPLE AXIS	คือ แยกกราฟสำหรับแสดงแกน Y
AUTO GAP	คือ แทรกช่องว่าง เมื่อมีข้อมูลหาย
HEIGHT BLOCKS	คือ ขนาดของบล็อกแสดงผล

WIDGET

TYPE

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

☐

AUTO GAP

☐

NO

HEIGHT BLOCKS

4

SAVE

CANCEL

Toggle

TOGGLE CAPTION	คือ ชื่อปุ่มสลับ
TOGGLE STATE	คือ สถานะของปุ่มสลับ
ON TEXT	คือ ชื่อปุ่มสลับเมื่ออยู่ในโหมดเปิด
OFF TEXT	คือ ชื่อปุ่มสลับเมื่ออยู่ในโหมดปิด
ONTOGGLE ON ACTION	คือ ส่งข้อมูลเมื่ออยู่ในโหมดเปิด
ONTOGGLE OFF ACTION	คือ ส่งข้อมูลเมื่ออยู่ในโหมดปิด

WIDGET

A simple toggle widget that can perform Javascript action.

TYPE	<input type="text" value="Toggle"/>	
TOGGLE CAPTION	<input type="text"/>	
TOGGLE STATE	<input type="text"/>	+ DATASOURCE JS EDITOR
	<small>Add a condition to switch a toggle state here. Otherwise it just toggle by click.</small>	
ON TEXT	<input type="text" value="ON"/>	
OFF TEXT	<input type="text" value="OFF"/>	
ONTOGGLEON ACTION	<input type="text"/>	
	<small>JS code to run when a toggle is switched to ON</small>	
ONTOGGLEOFF ACTION	<input type="text"/>	
	<small>JS code to run when a toggle is switched to OFF</small>	
ONCREATED ACTION	<input type="text"/>	
	<small>JS code to run after a toggle is created</small>	

SAVE

CANCEL

Slider

SLIDER CAPTION	คือ ชื่อสไลด์
FILLED COLOR	คือ สีของสไลด์
DISPLAY VALUE	คือ แสดงค่าข้อมูลหรือไม่
MIN VALUE	คือ ค่าต่ำสุดของสไลด์
MAX VALUE	คือ ค่าสูงสุดของสไลด์
STEP	คือ ความละเอียดของสไลด์
INITIAL VALUE	คือ ค่าเริ่มต้น
AUTO UPDATED VALUE	คือ อัปเดตค่าสไลด์อัตโนมัติ
ONSTART ACTION	คือ ส่งข้อมูลเมื่อเริ่มใช้งาน
ONSLIDE ACTION	คือ ส่งข้อมูลเมื่อเลื่อนสไลด์
ONSTOP ACTION	คือ ส่งข้อมูลเมื่อหยุดสไลด์
ONCREATED ACTION	คือ ส่งข้อมูลเมื่อสร้างสไลด์

WIDGET

A slider widget that can perform Javascript action.

TYPE	Slider	
SLIDER CAPTION		
FILLED COLOR		
DISPLAY VALUE	YES	<input type="checkbox"/>
MIN VALUE	0	
MAX VALUE	100	
STEP	1	
INITIAL VALUE	0	
	The default value set only the first time the widget is loaded.	
AUTO UPDATED VALUE		+ DATASOURCE ✕ JS EDITOR
	Slider will be updated upon the change of variables (e.g. other data sources).	
ONSTART ACTION		+ DATASOURCE ✕ JS EDITOR
	Add some Javascript here. You can access to a slider attribute using variables 'value' and 'percent'.	
ONSLIDE ACTION		+ DATASOURCE ✕ JS EDITOR
	Add some Javascript here. You can access to a slider attribute using variables 'value' and 'percent'.	
ONSTOP ACTION		+ DATASOURCE ✕ JS EDITOR
	Add some Javascript here. You can access to a slider attribute using variables 'value' and 'percent'.	
ONCREATED ACTION		
	JS code to run after a button is created	

SAVE

CANCEL

Map

LATITUDE	คือ เส้นละติจูด
LONGITUDE	คือ เส้นลองจิจูด
ZOOM	คือ ขยายแผนที่
STYLES	คือ รูปแบบการแสดงผล
TRAFFIC	คือ แสดงเส้นทางการเดินทาง
HEIGHT BLOCKS	คือ ขนาดของบล็อกแสดงผล

WIDGET

TYPE

Longdo Map

LATITUDE

+ DATASOURCE

JS EDITOR

LONGITUDE

+ DATASOURCE

JS EDITOR

ZOOM

1

STYLES

Map

TRAFFIC

NO

TOOLBAR

NO

HEIGHT BLOCKS

1

SAVE

CANCEL

ปฏิบัติ

2.3 เครื่องมือและอุปกรณ์การทดลอง

- 1) บอร์ด KidBright พร้อมสาย Micro USB
- 2) คอมพิวเตอร์ที่ติดตั้งโปรแกรม Arduino IDE
- 3) บอร์ด Relay 4 ช่อง
- 4) DHT11 เซ็นเซอร์จับอุณหภูมิและความชื้น
- 5) สายจัม








2.4 ขั้นตอนการทดลอง

- 1) เข้าใช้งาน netpie Feeboard
- 2) เปิดโปรแกรมตัวอย่าง NPFreeboard.ino
- 3) เชื่อมต่อ ESP32 กับ netpie.io
 - เปิดโปรแกรม NPFreeboard.ino ทำการแก้โปรแกรมดังนี้

```

1 #include "MicroGear.h"
2 #include "WiFi.h"
3
4 const char* ssid      = "XXXXXXXXXX";
5 const char* password = "XXXXXXXXXX";
6
7 #define APPID      "XXXXXXXXXX"
8 #define KEY        "XXXXXXXXXX"
9 #define SECRET     "XXXXXXXXXX"
10 #define ALIAS      "npapp"
11
12 #define led 17
13
14 WiFiClient client;
15 int timer = 0;
16 MicroGear microgear(client);
17
18 void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen)
19 {
20     Serial.print("Incoming message --> ");
21     msg[msglen] = '\0';
22     Serial.println((char *)msg);
23
24     char strMsg[msglen];
25     for (int i = 0; i < msglen; i++)
26     {
27         strMsg[i] = (char)msg[i];
28     }
29
30     String msgStr = String(strMsg).substring(0, msglen);
31     String strTopic = (char *)topic;
32
33     Serial.print("Topic: ");
34     Serial.println(strTopic);
35     Serial.print("Msg: ");
36     Serial.println(msgStr);
37

```

 ชื่อไวไฟที่เชื่อมต่อ
 รหัสไวไฟที่เชื่อมต่อ
 APPID จากเว็บ netpie
 Device Key เว็บ netpie
 SECRET จาก Device Key
 ALIAS (ตั้งชื่ออุปกรณ์)
 ตัวแปร led ใช้งาน pin 17 (หลอด LED)

```

38     if (msgStr == "ON")
39     {
40         Serial.println("ON");
41         digitalWrite(led, 0);
42     }
43
44     if (msgStr == "OFF")
45     {
46         Serial.println("OFF");
47         digitalWrite(led, 1);
48     }
49 }
50
51 void onFoundgear(char *attribute, uint8_t* msg, unsigned int msglen)
52 {
53     Serial.print("Found new member --> ");
54     for (int i = 0; i < msglen; i++)
55         Serial.print((char)msg[i]);
56     Serial.println();
57 }
58
59 void onLostgear(char *attribute, uint8_t* msg, unsigned int msglen)
60 {
61     Serial.print("Lost member --> ");
62     for (int i = 0; i < msglen; i++)
63         Serial.print((char)msg[i]);
64     Serial.println();
65 }
66
67 void onConnected(char *attribute, uint8_t* msg, unsigned int msglen)
68 {
69     Serial.println("Connected to NETPIE...");
70     microgear.setAlias(ALIAS);
71 }
72
73
74 void setup()
75 {
76     microgear.on(MESSAGE, onMsghandler);
77     microgear.on(PRESENT, onFoundgear);
78     microgear.on(ABSENT, onLostgear);
79     microgear.on(CONNECTED, onConnected);
80
81     Serial.begin(115200);
82     Serial.println("Starting...");
83
84     pinMode(led, OUTPUT);
85     digitalWrite(led, HIGH);
86

```

← ประมวลผลตัวอักษรที่ได้รับ

```

87   if (WiFi.begin(ssid, password))
88   {
89       while (WiFi.status() != WL_CONNECTED)
90       {
91           delay(500);
92           Serial.print(".");
93       }
94   }
95
96   Serial.println("WiFi connected");
97   Serial.println("IP address: ");
98   Serial.println(WiFi.localIP());
99
100  microgear.init(KEY, SECRET, ALIAS);
101
102  microgear.connect(APPID);
103  microgear.subscribe("/gearname/npapp/data/~/command");
104 }
105
106 void loop()
107 {
108     if (microgear.connected())
109     {
110         microgear.loop();
111
112         if (timer >= 5000)
113         {
114             Serial.println("Publish...");
115
116             microgear.chat(ALIAS, "Hello");
117             microgear.publish("/gearname/npapp/light", analogRead(36));
118             timer = 0;
119         }
120         else timer += 100;
121     }
122     else
123     {
124         Serial.println("connection lost, reconnect...");
125         if (timer >= 5000)
126         {
127             microgear.connect(APPID);
128             timer = 0;
129         }
130         else timer += 100;
131     }
132     delay(10);
133 }

```

- ทำการอัปเดตแล้วเปิด Serial Monitor จะแสดงข้อความการเชื่อมต่อ

```
COM26
Starting...
..WiFi connected
IP address:
192.168.12.116
Connected to NETPIE...
Publish...
Incoming message --> Hello
```

4) สร้าง WIDGET เพื่อแสดงข้อความ Text ที่ส่งมา

WIDGET

TYPE

Text

TITLE

Word

SIZE

Regular

VALUE

datasources["NPFreeboard"]["/NetpieLab/gearname/npapp"]

+ DATASOURCE

JS EDITOR

INCLUDE SPARKLINE

☐ NO

ANIMATE VALUE CHANGES

YES ☒

UNITS

SAVE

CANCEL

5) สร้าง WIDGET เพื่อแสดงข้อความ Text จากการอ่านค่า LDR

WIDGET

TYPE

Text

TITLE

LDR

SIZE

Regular

VALUE

datasources["NPFreeboard"]["/NetpieLab/gearname/npapp/light"]

+ DATASOURCE

JS EDITOR

INCLUDE SPARKLINE

☐ NO

ANIMATE VALUE CHANGES

YES ☒

UNITS

SAVE

CANCEL

6) สร้าง WIDGET เพื่อแสดงปริมาณความเข้มแสงจากการอ่านค่า LDR

WIDGET

TYPE

Gauge

TITLE

LDR

VALUE

`datasources["NPFreeboard"]["/NetpieLab/gearname/npapp/light"]`

+ DATASOURCE

JS EDITOR

UNITS

MINIMUM

0

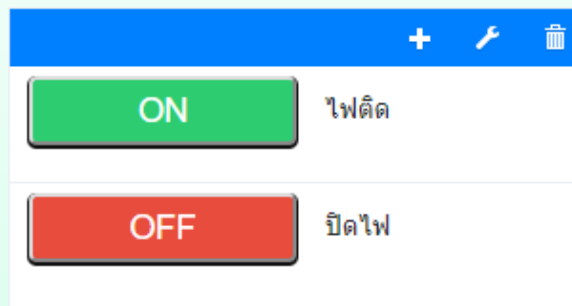
MAXIMUM

1024

SAVE

CANCEL

7) สร้าง WIDGET Button เพื่อควบคุมการเปิดปิดหลอดไฟ LED



WIDGET

A simple button widget that can perform Javascript action.

TYPE

Button

BUTTON CAPTION

ON

LABEL TEXT

ไฟติด

BUTTON COLOR

Green

ONCLICK ACTION

`microgear["NPFreeboard"].chat("npapp/data/$/command", "ON")`

+ DATASOURCE

JS EDITOR

Add some Javascript here. You can chat and publish with a datasource's microgear like this :
microgear["mygear"].chat("mylamp","ON"), where "mygear" is a datasource name.

ONCREATED ACTION

JS code to run after a button is created

SAVE

CANCEL

WIDGET

A simple button widget that can perform Javascript action.

TYPE: Button

BUTTON CAPTION: OFF

LABEL TEXT: ปิดไฟ

BUTTON COLOR: Red

ONCLICK ACTION: `microgear["NPFreeboard"].chat("npapp/data/$/command", "OFF")` + DATASOURCE JS EDITOR

ONCREATED ACTION:
 JS code to run after a button is created

SAVE CANCEL

8) สร้าง Sparkline ดูการแสดงผลเส้นกราฟจากการรับค่าแสง LDR

WIDGET

TYPE: Sparkline

TITLE: LDR

VALUE: `datasources["NPFreeboard"]["/NetpieLab/geaname/npapp/light"]` + DATASOURCE JS EDITOR

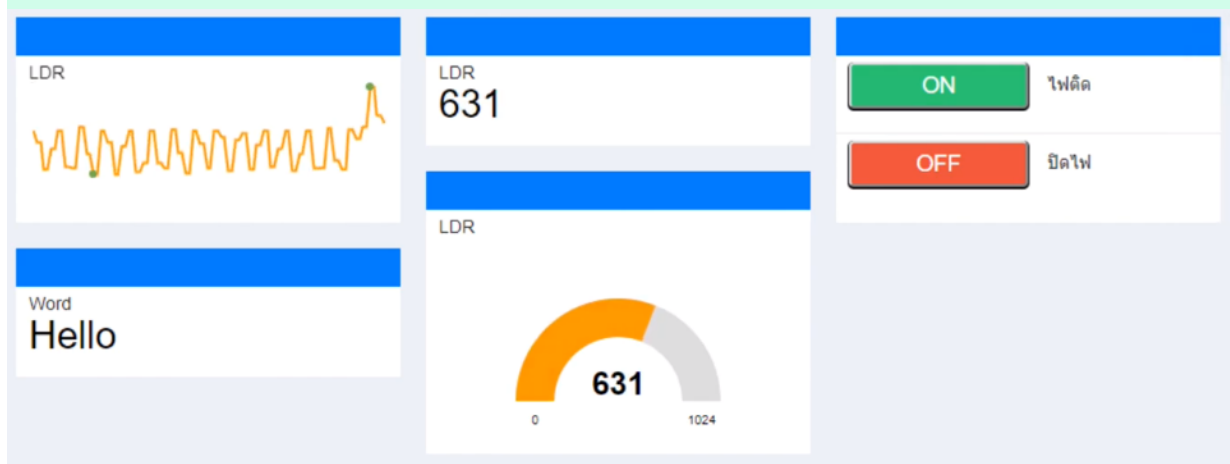
+ ADD

INCLUDE LEGEND: ☐ NO

LEGEND:
 Comma-separated for multiple sparklines

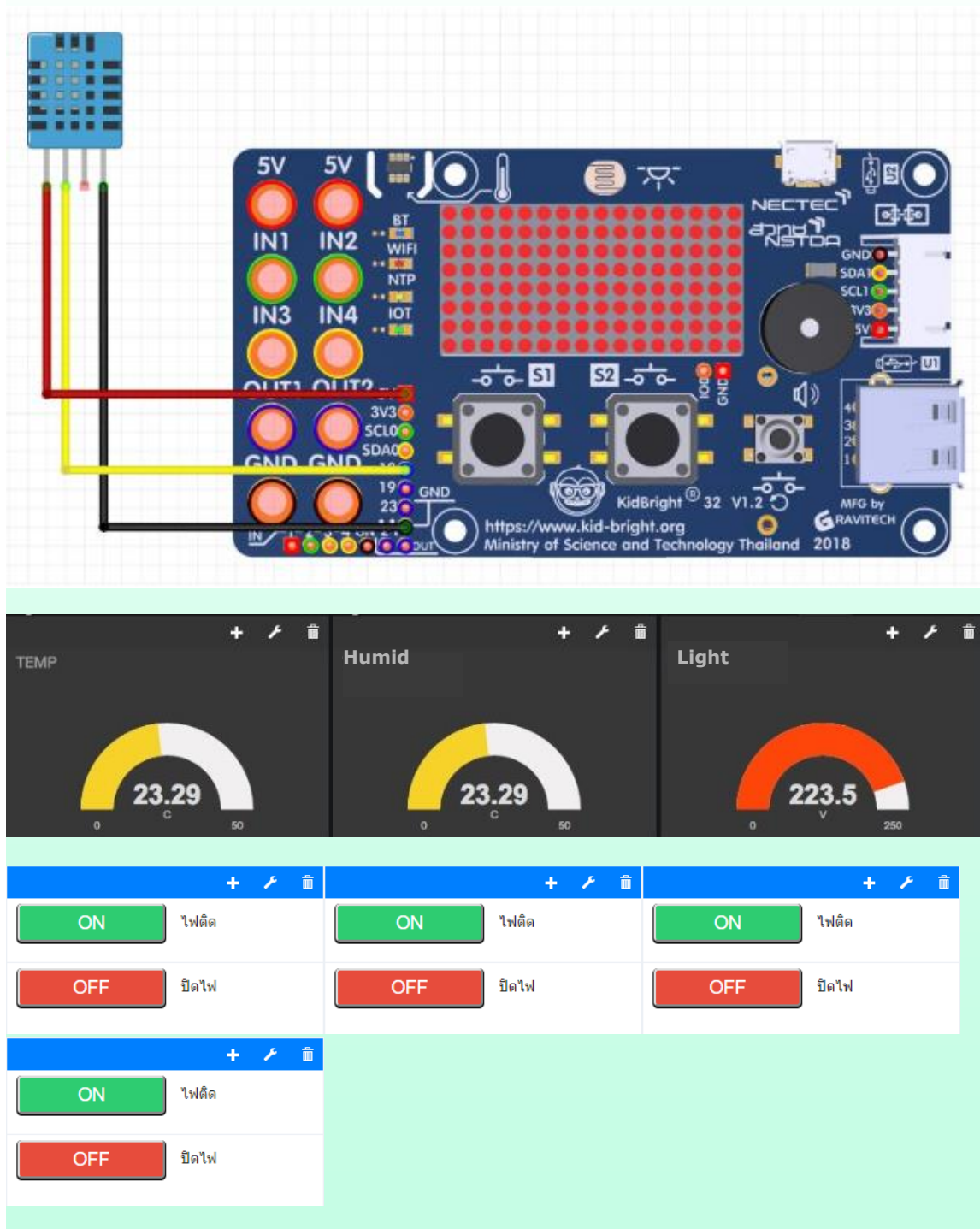
SAVE CANCEL

9) ทดสอบการแสดงผล



2.5 โจทย์ฝึกปฏิบัติเพิ่มทักษะ

- 1) ประกอบวงจรโดยต่อ DHT11 และ Relay Module 4 CH กับบอร์ด
- 2) เขียนโปรแกรมส่งข้อมูลอุณหภูมิ (Temperature) ไปแสดงผลที่ Gauge1
- 3) เขียนโปรแกรมส่งข้อมูลความชื้นในอากาศ (Humidity) ไปแสดงผลที่ Gauge2
- 4) เขียนโปรแกรมส่งข้อมูลแสงไปแสดงผลที่ Gauge3
- 5) เขียนโปรแกรมควบคุม Relay 4 ช่อง โดยใช้ Button 4 ชุด
- 6) เพิ่มเต็ม WIDGET และปรับแต่งตามใจชอบ



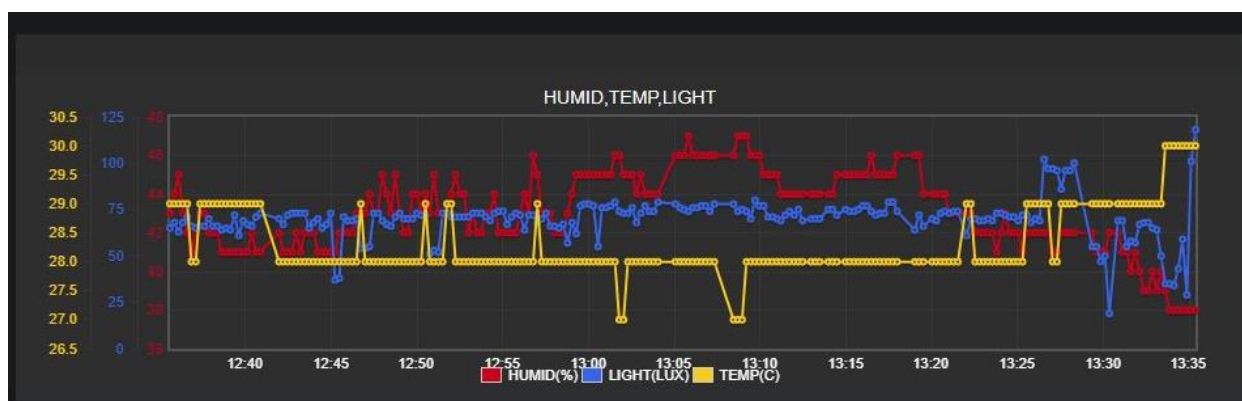
จุดประสงค์

- 1.สามารถเลือกใช้อุปกรณ์สำหรับการทดลอง NETPIE Feeds ได้
- 2.สามารถเข้าใช้งาน NETPIE Feeds ได้
- 3.สามารถตั้งค่าและปรับแต่งการแสดงผลบน NETPIE Feeds ได้
- 4.สามารถแก้ไขโปรแกรมให้บอร์ด KidBright เชื่อมต่อกับ NETPIE ได้
- 5.สามารถแก้ไขโปรแกรมให้ส่งค่าจากบอร์ด KidBright แสดงผลที่หน้า Feeds และ Freeboard ได้
- 6.สามารถปฏิบัติงานสำเร็จตามกำหนดเวลาได้

ทฤษฎี

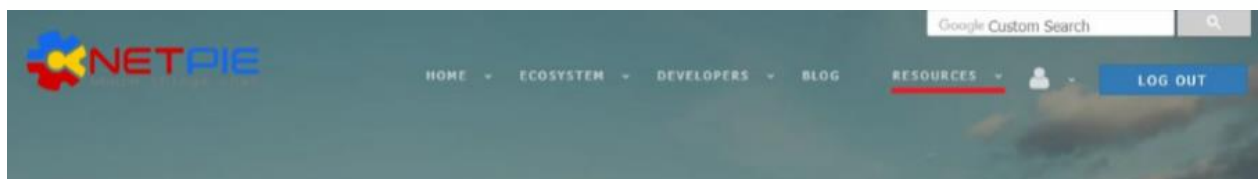
3.1 NETPIE Feeds

Feeds เป็นการเก็บข้อมูลและสามารถนำมาแสดงในลักษณะเหมือน Freeboard แต่จะออกมาในรูปของกราฟ อีกทั้งสามารถนำค่า Feeds ที่ได้มาแสดงผลในส่วนของ Freeboard ได้

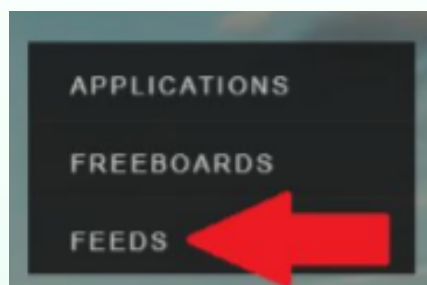


3.2 การใช้งาน Feeds

1) ลือตอินเว็บ NETPIE.io เลือกหัวข้อ RESOURCES



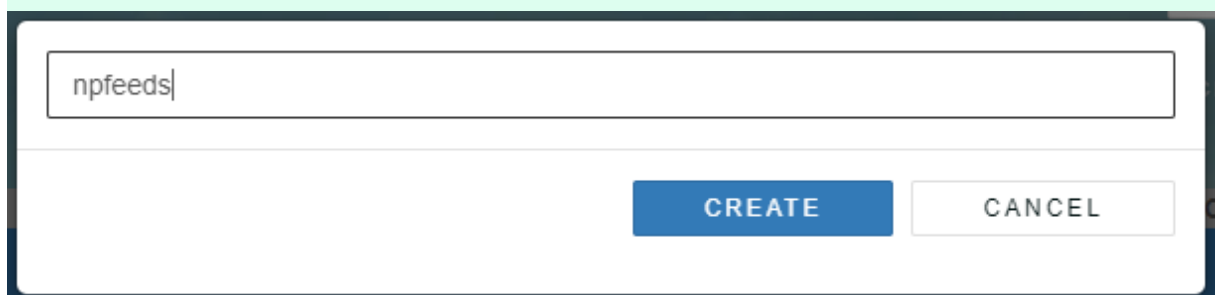
2) กดเลือก FEEDS



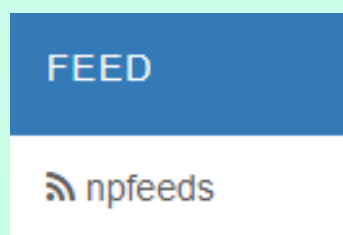
3) กดเครื่องหมายบวก (+) เพื่อเพิ่ม FEEDS



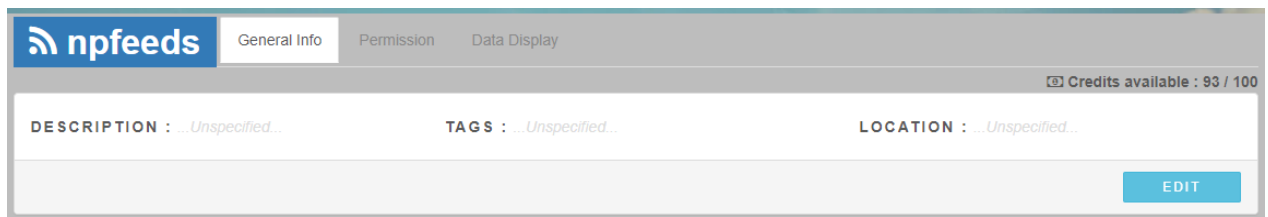
4) ตั้งชื่อ FEEDS



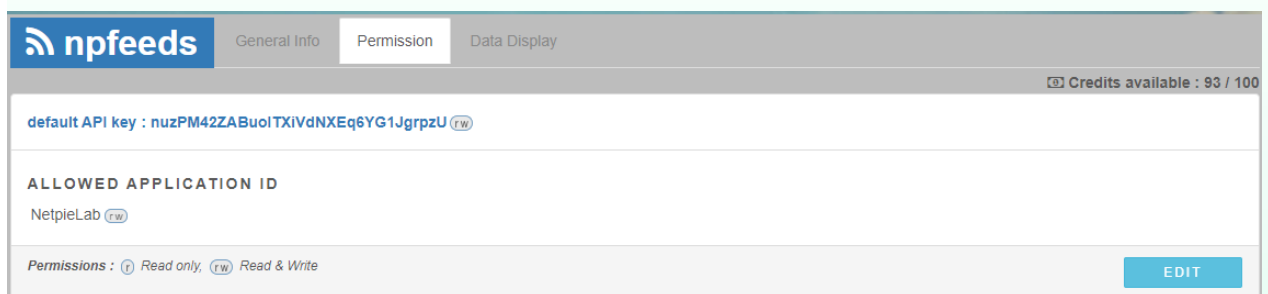
5) แสดงชื่อ FEEDS ที่สร้างซึ่งจะใช้อ้างอิงบนเว็บ NETPIE



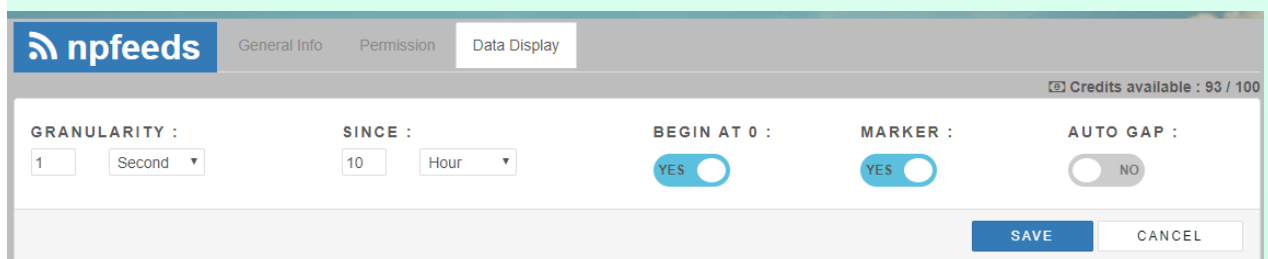
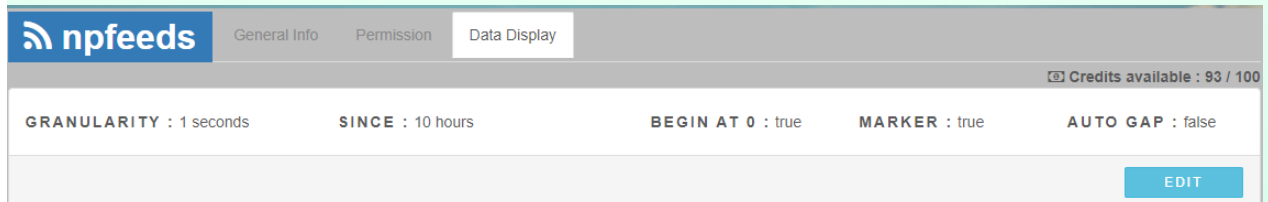
6) แถบ General Info คลิก EDIT กรอกข้อมูลหรือไม่ก็ได้



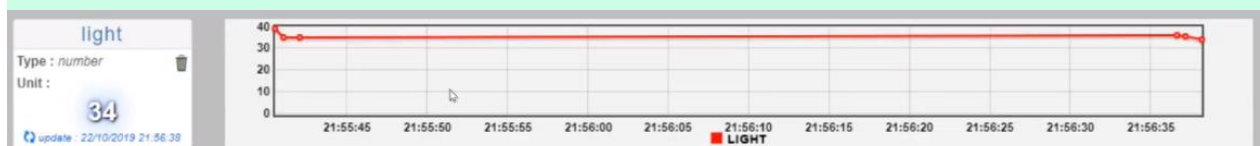
7) แถบ Permission คลิก EDIT ทำการ Sync API Key และเลือก APPID ที่จะนำค่ามาที่ Feeds



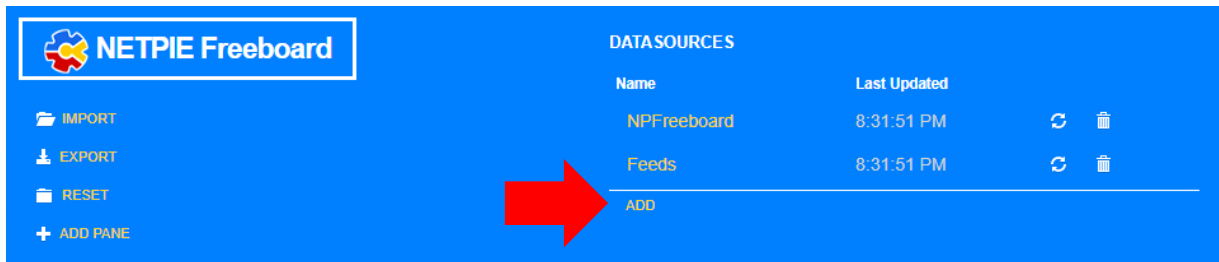
8) แถบ Data Display คลิก EDIT ปรับแต่งช่วงเวลาการแสดงผล



9) แสดงหน้า FEEDS



10) กรณีต้องการนำไปแสดงผลบนหน้า Freeboard จะต้องเพิ่ม DATASOURCE --> NETPIE Feed ที่หน้า Freeboard

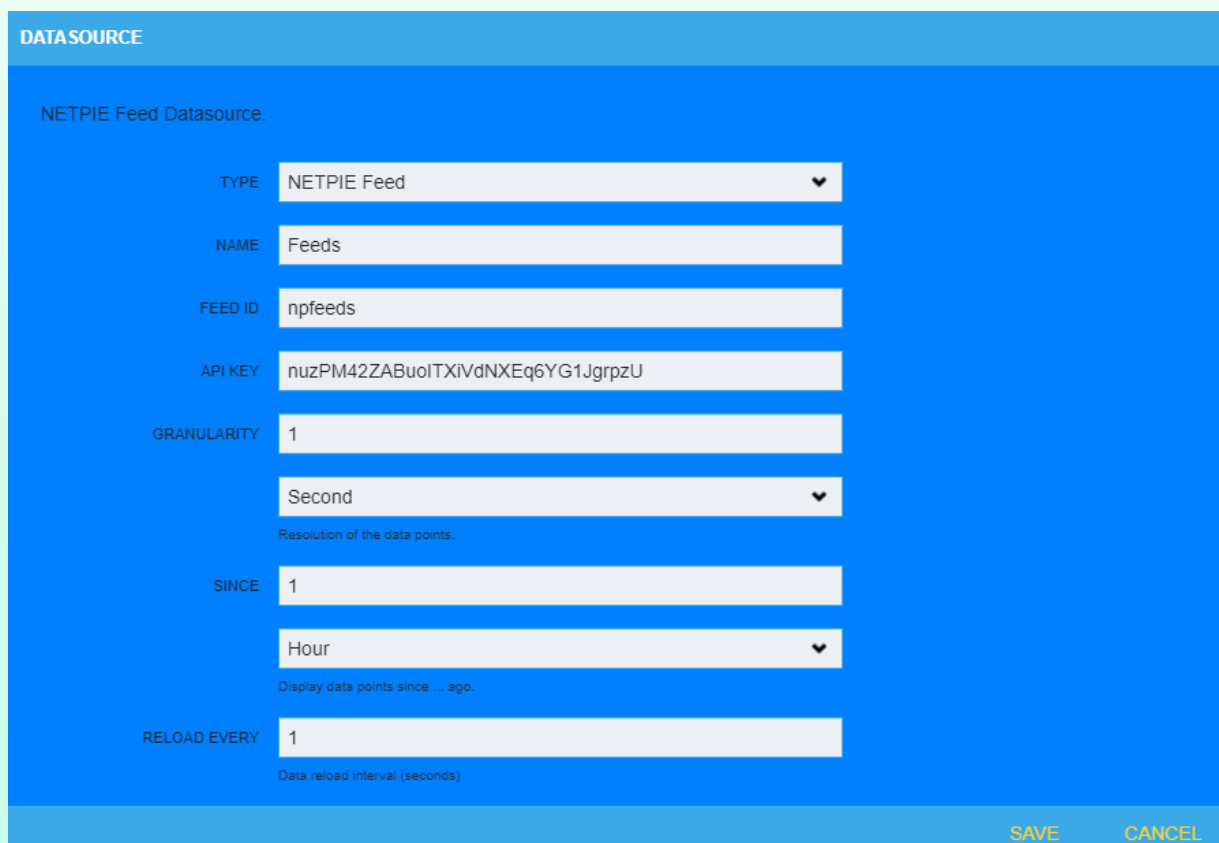


The screenshot shows the NETPIE Freeboard interface. On the left, there is a sidebar with buttons: IMPORT, EXPORT, RESET, and ADD PANE. The main area is titled 'DATASOURCES' and contains a table with the following data:

Name	Last Updated		
NPFreeboard	8:31:51 PM	↺	🗑️
Feeds	8:31:51 PM	↺	🗑️
ADD			

11) DATASOURCE NETPIE Feed

NAME	คือ ชื่อสำหรับเชื่อมต่อ
FEED ID	คือ ชื่อ Feed ที่สร้าง
API KEY	คือ รหัสผ่าน
GRANULARITY	คือ แสดงจุดข้อมูลตามเวลา
SINCE	คือ ระยะเวลาในการแสดงผล
RELOAD EVERY	คือ โหลดข้อมูลใหม่ตามเวลา

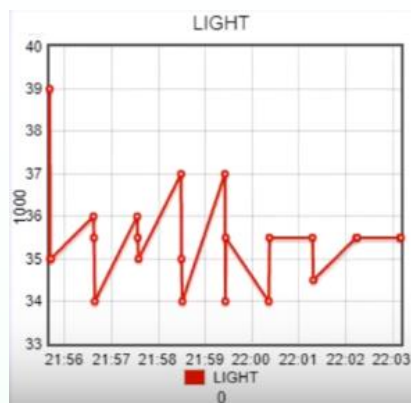


The screenshot shows the 'DATASOURCE' configuration form for 'NETPIE Feed Datasource'. The form contains the following fields:

- TYPE: NETPIE Feed (dropdown)
- NAME: Feeds
- FEED ID: npfeeds
- API KEY: nuzPM42ZABuoITXiVdNXEq6YG1JgrpzU
- GRANULARITY: 1 (dropdown, showing 'Second')
Resolution of the data points.
- SINCE: 1 (dropdown, showing 'Hour')
Display data points since ... ago.
- RELOAD EVERY: 1 (dropdown, showing '1')
Data reload interval (seconds)

At the bottom right, there are 'SAVE' and 'CANCEL' buttons.

12) หน้าแสดงผล



13) การเพิ่ม FEEDS ให้กลับไป FEEDS

light

Type : number

Unit : N/A

+ ADD

เพิ่ม

Field name

Number

Unit name

SAVE CANCEL

ชื่อ Field
เป็นจำนวน
หน่วยที่ใช้

14) แสดง temp และ light ที่เพิ่มเสร็จเรียบร้อยแล้ว

temp

Type : number

Unit : C

N/A

light

Type : number

Unit : N/A

N/A

07:00:00 TEMP(C)

07:00:00 LIGHT

ปฏิบัติ

3.3 เครื่องมือและอุปกรณ์การทดลอง

- 1) บอร์ด KidBright พร้อมสาย Micro USB
- 2) คอมพิวเตอร์ที่ติดตั้งโปรแกรม Arduino IDE
- 3) DHT11 เซ็นเซอร์จับอุณหภูมิและความชื้น
- 4) สายจัม

3.4 ขั้นตอนการทดลอง

- 1) เข้าใช้งาน netpie Feeds
- 2) เปิดโปรแกรมตัวอย่าง NPfeeds แล้วทำการแก้ไขโปรแกรม

NPfeeds

```
1 #include "MicroGear.h"
2 #include "WiFi.h"
3
4 const char* ssid      = "RRT";
5 const char* password = "99887766";
6
7 #define APPID      "NetpieLab"
8 #define KEY        "fUSEzACtZV2yVIP"
9 #define SECRET     "V6pYHfkHnJAl7LHVUchGYYTEU"
10 #define ALIAS      "npapp"
11
12 #define FEEDS_ID    "npfeeds"
13 #define FEEDS_KEY   "nuzPM42ZABuoITXiVdNXEq6YG1JgrpzU"
14
15 #define led 17
16 #define LDR 36
17
18 WiFiClient client;
19 int timer = 0;
20 MicroGear microgear(client);
21
22 int readLight()
23 {
24     int light = 0;
25     light = map(analogRead(LDR), 0, 1023, 100, 0);
26     return light;
27 }
```

← ใส่ Feeds ID

← ใส่ API Key

```

28
29 void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen)
30 {
31     Serial.print("Topic --> ");
32     Serial.println((char *)topic);
33
34     Serial.print("Incoming message --> ");
35     msg[msglen] = '\0';
36     Serial.println((char *)msg);
37
38     char strMsg[msglen];
39     for (int i = 0; i < msglen; i++)
40     {
41         strMsg[i] = (char)msg[i];
42     }
43
44     String msgStr = String(strMsg).substring(0, msglen);
45     String strTopic = (char *)topic;
46
47     Serial.print("Topic: ");
48     Serial.println(strTopic);
49     Serial.print("Msg: ");
50     Serial.println(msgStr);
51
52     if (msgStr == "ON")
53     {
54         Serial.println("ON");
55         digitalWrite(led, 0);
56     }
57
58     if (msgStr == "OFF")
59     {
60         Serial.println("OFF");
61         digitalWrite(led, 1);
62     }
63 }
64
65 void onFoundgear(char *attribute, uint8_t* msg, unsigned int msglen)
66 {
67     Serial.print("Found new member --> ");
68     for (int i = 0; i < msglen; i++)
69         Serial.print((char)msg[i]);
70     Serial.println();
71 }
72

```

```

73 void onLostgear(char *attribute, uint8_t* msg, unsigned int msglen)
74 {
75     Serial.print("Lost member --> ");
76     for (int i = 0; i < msglen; i++)
77         Serial.print((char)msg[i]);
78     Serial.println();
79 }
80
81 void onConnected(char *attribute, uint8_t* msg, unsigned int msglen)
82 {
83     Serial.println("Connected to NETPIE...");
84     microgear.setAlias(ALIAS);
85 }
86
87
88 void setup()
89 {
90     microgear.on(MESSAGE, onMsghandler);
91     microgear.on(PRESENT, onFoundgear);
92     microgear.on(ABSENT, onLostgear);
93     microgear.on(CONNECTED, onConnected);
94
95     Serial.begin(115200);
96     Serial.println("Starting...");
97
98     pinMode(led, OUTPUT);
99     digitalWrite(led, HIGH);
100
101     if (WiFi.begin(ssid, password))
102     {
103         while (WiFi.status() != WL_CONNECTED)
104         {
105             delay(500);
106             Serial.print(".");
107         }
108     }
109
110     Serial.println("WiFi connected");
111     Serial.println("IP address: ");
112     Serial.println(WiFi.localIP());
113
114     microgear.init(KEY, SECRET, ALIAS);
115
116     microgear.connect(APPID);
117     microgear.subscribe("/gearname/npapp/data/$/command");
118 }
119

```

```

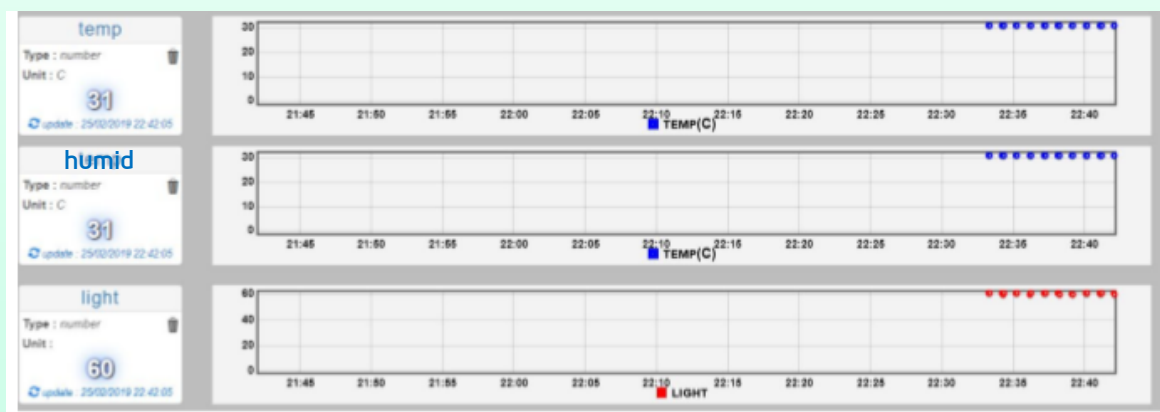
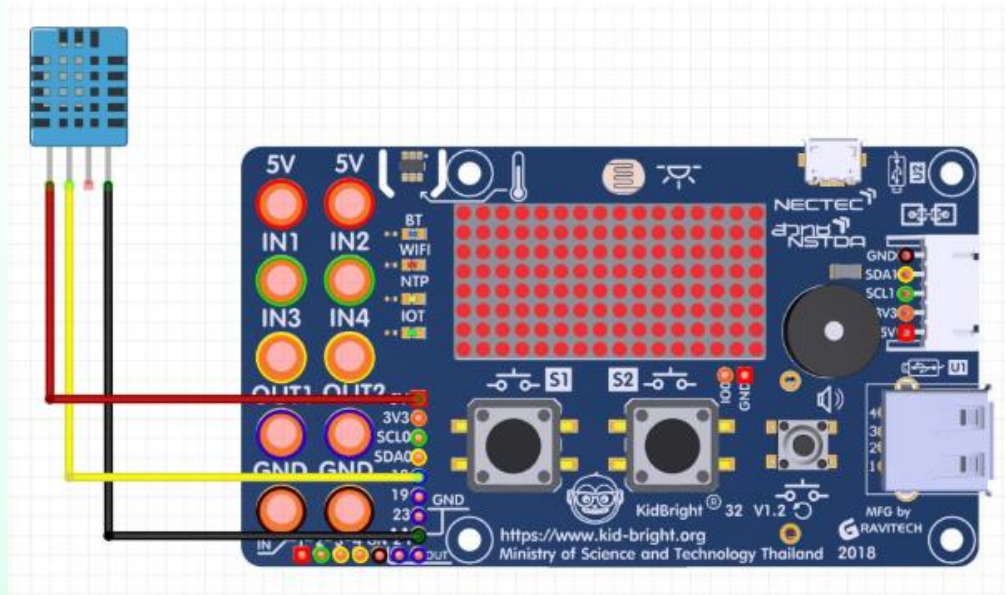
120 void loop()
121 {
122   if (microgear.connected())
123   {
124     microgear.loop();
125
126     if (timer >= 5000)
127     {
128       Serial.println("Publish...");
129
130       microgear.chat(ALIAS, "Hello");
131       microgear.publish("/gearname/npapp/light", analogRead(36));
132       String dataFeeds = "light:" + String(readLight());
133       microgear.writeFeed(FEEDS_ID, dataFeeds, FEEDS_KEY);
134       timer = 0;
135     }
136     else timer += 100;
137   }
138   else
139   {
140     Serial.println("connection lost, reconnect...");
141     if (timer >= 5000)
142     {
143       microgear.connect(APPID);
144       timer = 0;
145     }
146     else timer += 100;
147   }
148   delay(10);
149 }

```

ส่งค่าเข้า Feeds

3.5 โจทย์ฝึกปฏิบัติเพิ่มทักษะ

- 1) ประกอบวงจรดังภาพ
- 2) เขียนโปรแกรมส่งข้อมูลอุณหภูมิไปแสดงผลที่ Feeds View 1
- 3) เขียนโปรแกรมส่งข้อมูลอุณหภูมิไปแสดงผลที่ Feeds View 2
- 4) เพิ่มโปรแกรมส่งข้อมูลแสงไปแสดงผลที่ Feeds View 3
- 5) นำข้อมูล Feeds View ทั้ง 3 ไปแสดงยัง Freeboard



แสดงผลหน้า Freeboard

