

การใช้งาน NETPIE และ True NB-IoT

บทความนี้จะสอนการใช้งาน True NB-IoT ด้วย NETPIE Platform เบื้องต้น โดยการส่งค่าอุณหภูมิและความชื้นไปยัง Web Application สำเร็จรูป NETPIE Freeboard เพื่อเป็นตัวอย่างสำหรับนำไปประยุกต์เป็นอุปกรณ์ IoT แบบต่าง ๆ ต่อไป

อุปกรณ์ที่ใช้

- Arduino UNO หรือ Arduino Mega (บทความนี้เลือกใช้ UNO)
- True NB-IoT Shield
- DHT22 (เซ็นเซอร์วัดอุณหภูมิและความชื้น)

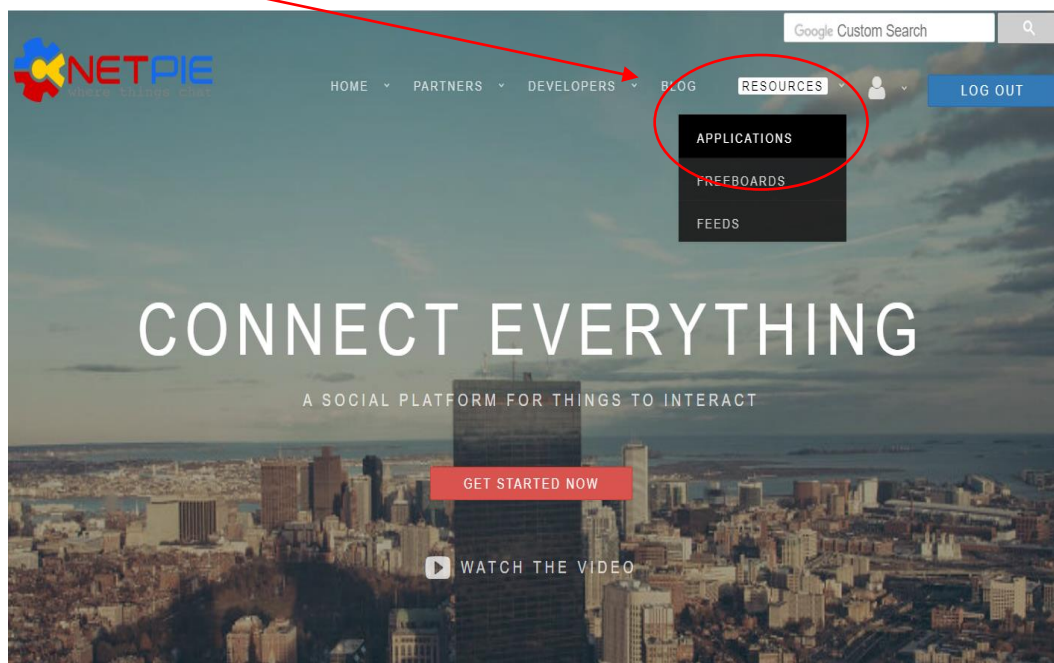
สิ่งที่ต้องมี

- ติดตั้ง Arduino IDE เวอร์ชัน 1.6.0 ขึ้นไปบนเครื่องเรียบร้อยแล้ว
- สมัครสมาชิก netpie.io และสร้าง appid, freeboard เรียบร้อยแล้ว เพื่อให้ได้ APP ID, KEY และ SECRET

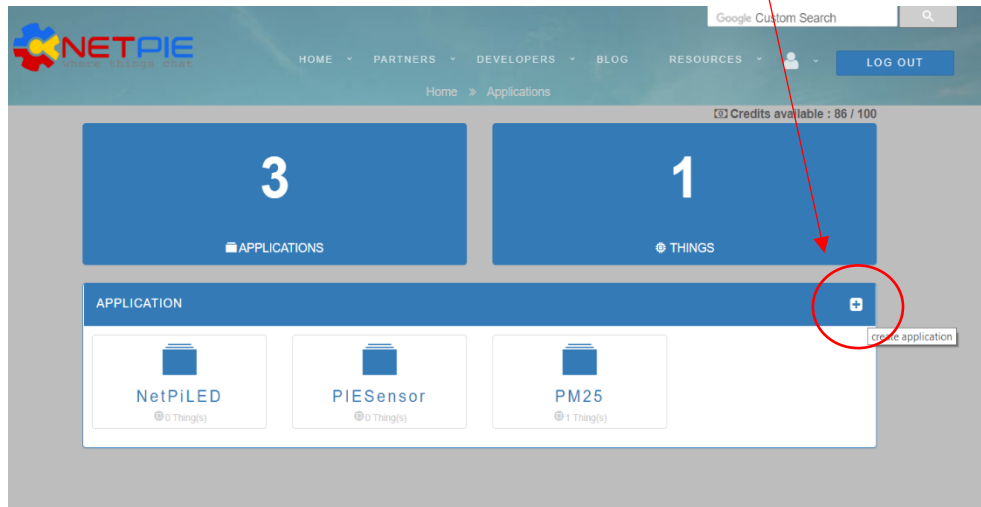
สร้าง Application บน <https://netpie.io/>

1. Login ด้วย User Name และ Password ที่ได้ทำการ Register ไว้ แล้วเลือกหัวข้อ APPLICATIONS ตรงเมนู

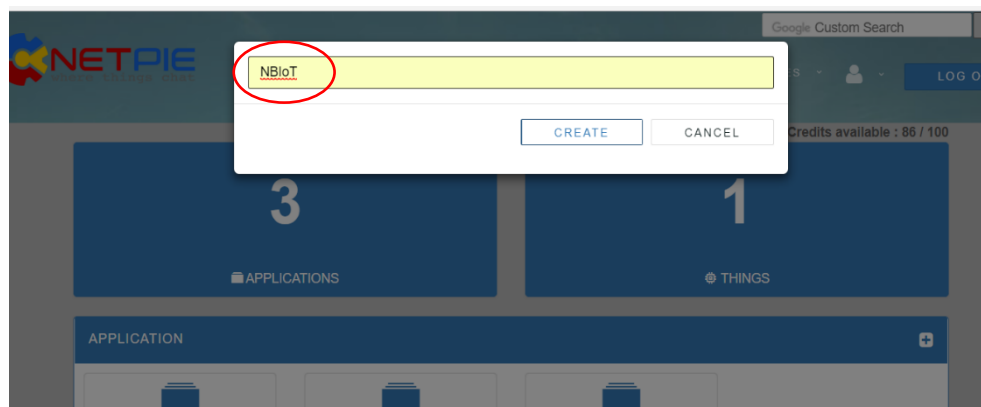
RESOURCES



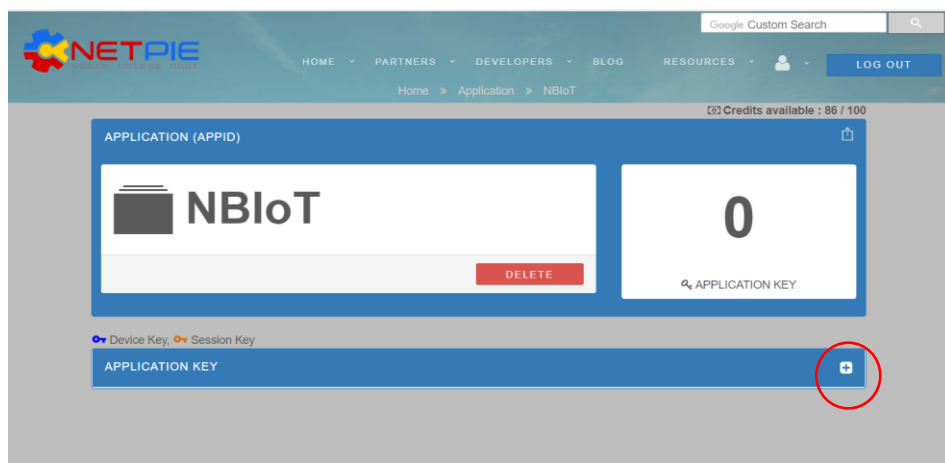
2. สร้าง APPLICATION ขึ้นมา (ในที่นี้ให้ชื่อว่า NBloT) ด้วยการกด ปุ่ม +



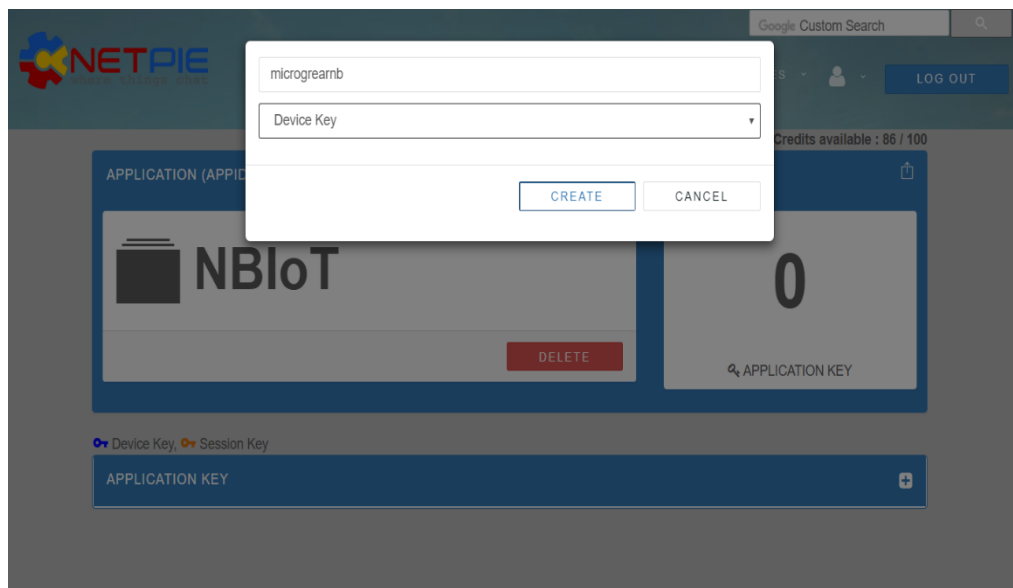
3. ใส่ชื่อ NBloT แล้วกด CREATE (ชื่อนี้จะเป็น APP ID ที่จะเอาไปใส่ในArduino Code)



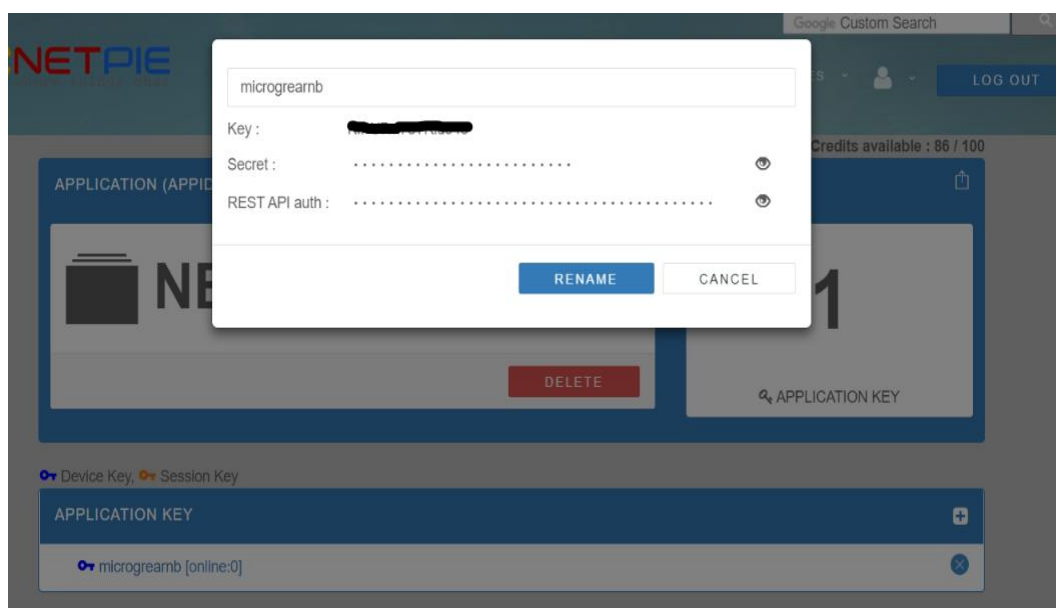
4. ทำการสร้าง APPLICATION KEY ด้วยการกด ปุ่ม +



5. กำหนดชื่อ และเลือกเป็น Device Key แล้ว กด CREATE จะได้ APPLICATION KEY ตามต้องการ



6. เราสามารถ กดดู ค่าต่างที่จำเป็นที่จะใช้ใน Arduino Code เพื่อเชื่อมต่อกับ NETPIE ด้วยการกดไปที่ชื่อ APPLICATION KEY ที่กำหนดไว้ตามข้อ 5.



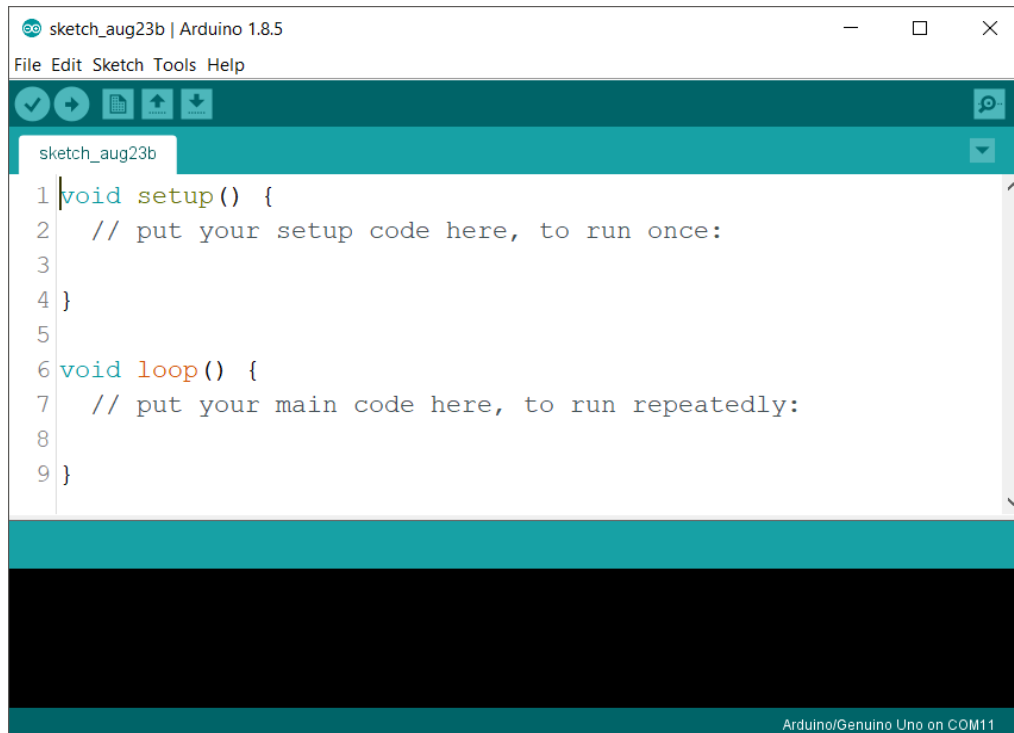
บันทึกค่าต่าง ไว้ ดังนี้ เพื่อใช้ใน Arduino Code

- a) APP ID (NBloT)
- b) KEY
- c) Secret

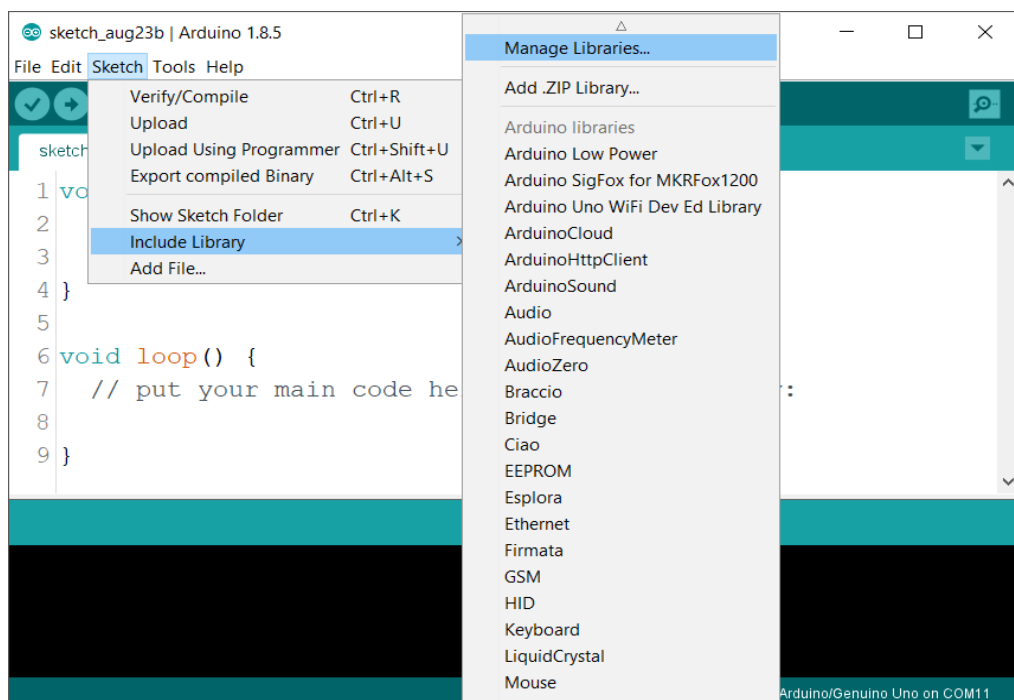
ติดตั้ง Arduino Library

Microgear NB-IoT

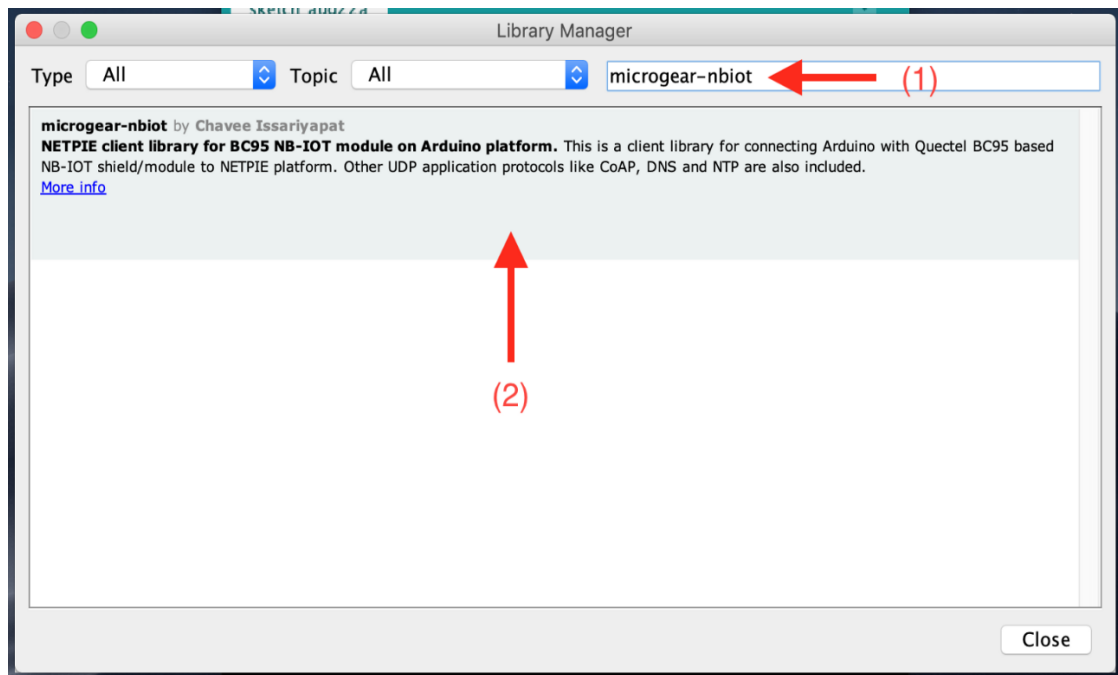
1. เปิดโปรแกรม Arduino IDE



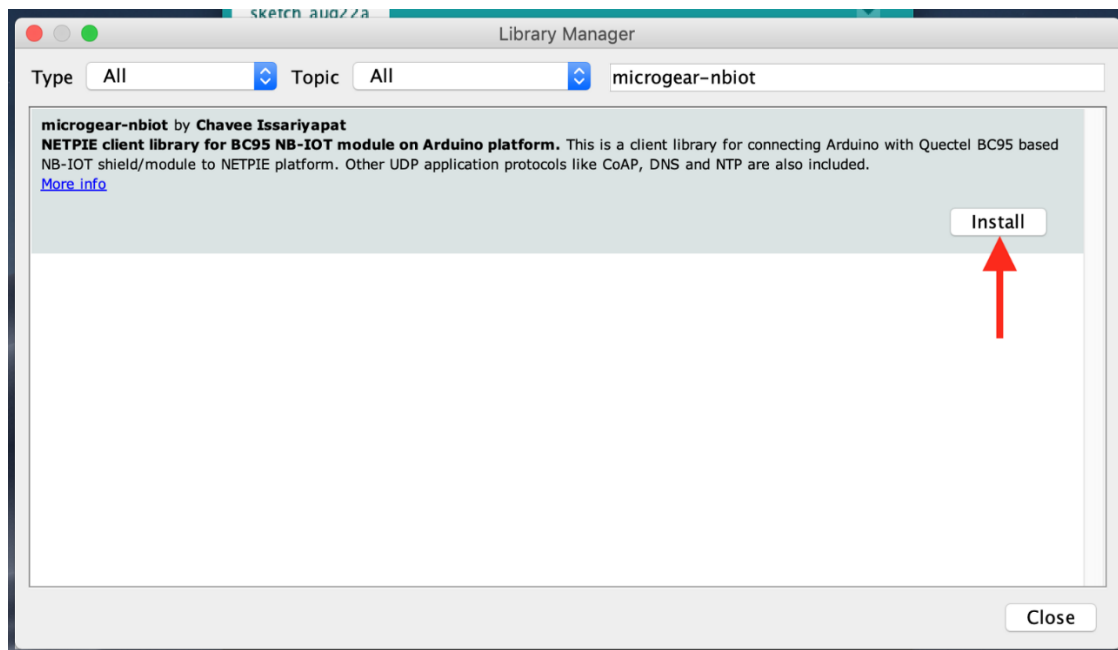
2. คลิกที่เมนู Manage Libraries... โดยเข้าจาก Sketch -> Include Library->Manage Libraries



3. กรอกข้อความ “microgear-nbiot” ในช่องค้นหา และคลิกที่ microgear-nbiot ที่แสดงขึ้นมา 1 ครั้ง



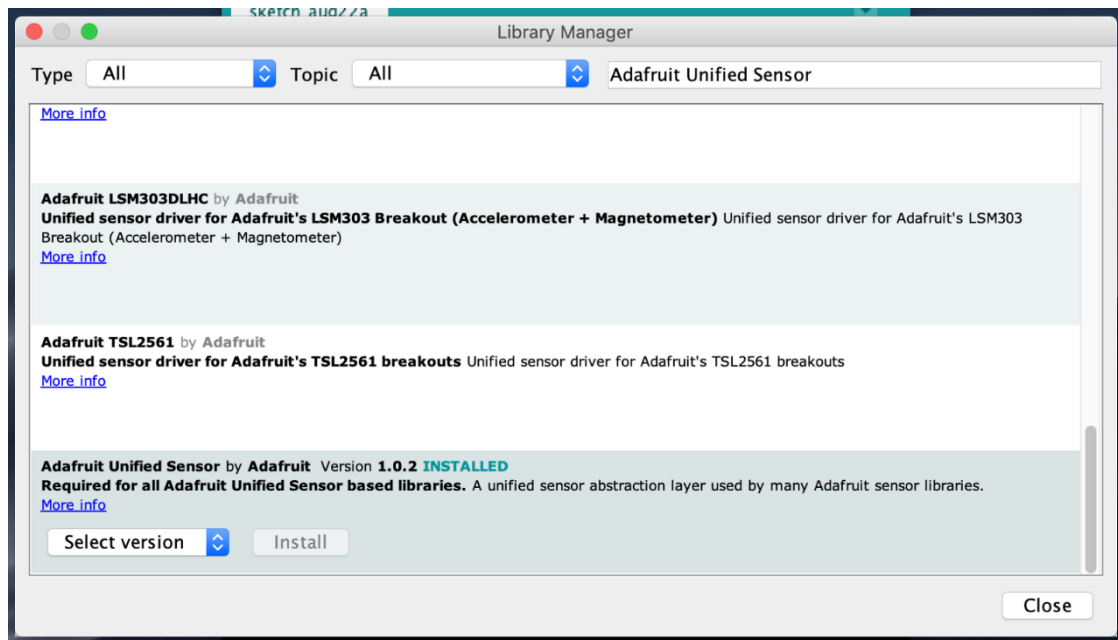
4. คลิกปุ่ม Install ที่แสดงขึ้นมา



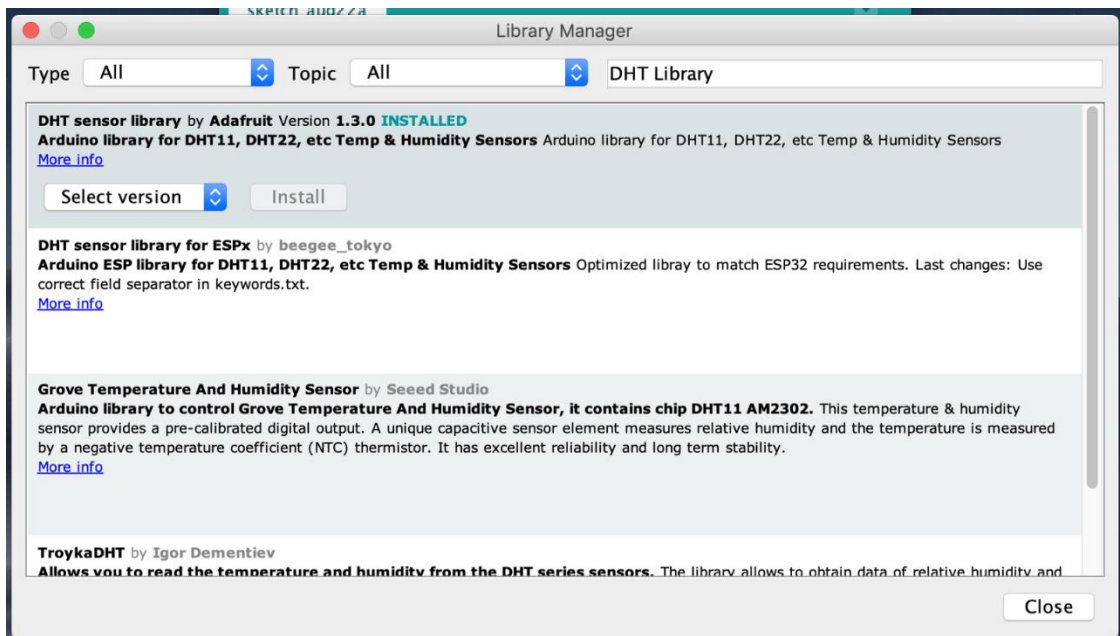
Adafruit Unified Sensor และ DHT Library

ทำซ้ำวิธีเดียวกับ Microgear NB-IoT ในขั้นตอน 3 และ 4 แต่เปลี่ยนข้อความสำหรับค้นหาในขั้นตอนที่ 3 เป็น Adafruit Unified Sensor และ DHT Library

ติดตั้ง Adafruit Unified Sensor

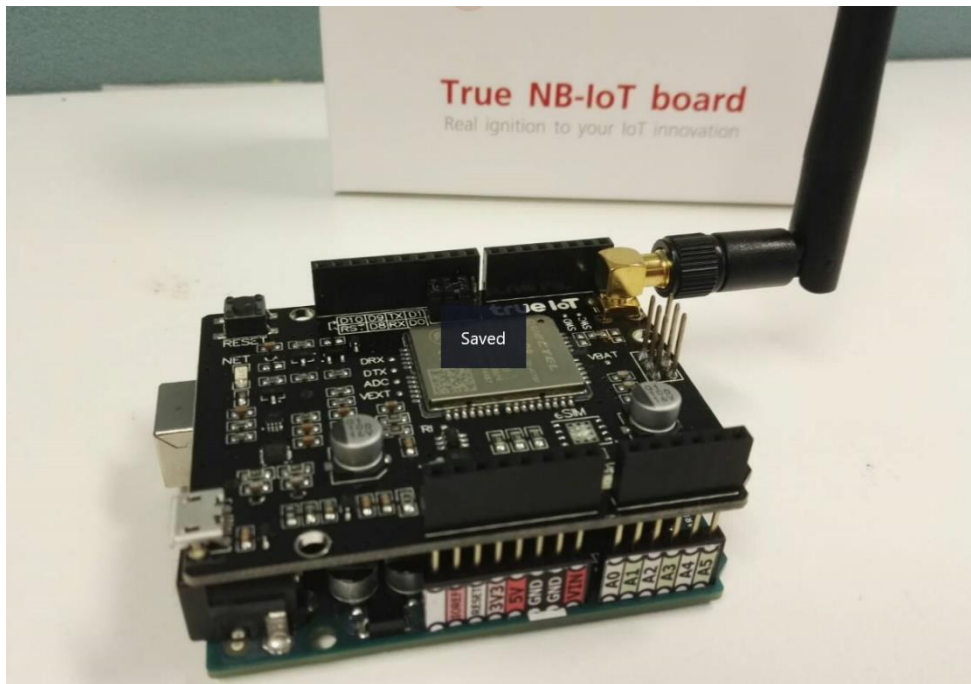


ติดตั้ง DHT Library

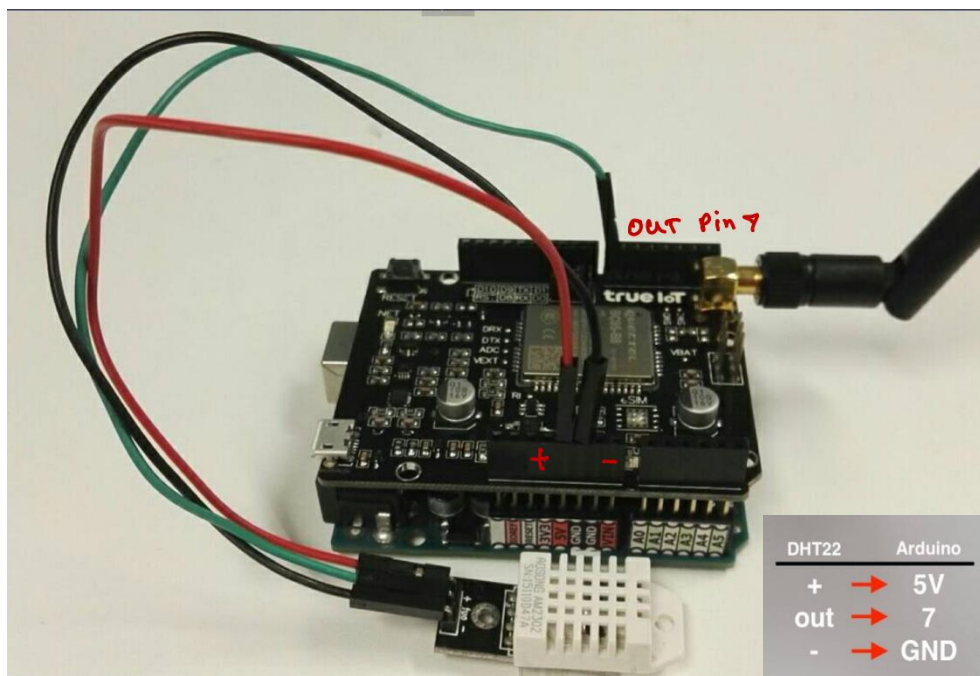


การเชื่อมต่ออุปกรณ์

1. ประกอบ NB-IoT Shield เข้ากับ Arduino



2. เชื่อมต่อ DHT-22 เข้ากับ NB-IoT Shield ดังรูปภาพ



กำหนด Code ลง Arduino IDE (อย่าลืมนำ appid, key, secret ของตัวเองลงไปใส่ใน code ด้วยนะ)

```
#if defined(__AVR_ATmega328P__) || defined(__AVR_ATmega168__) // Arduino UNO
#include <AltSoftSerial.h>
AltSoftSerial bc95serial;
#elif defined(__AVR_ATmega1280__) || defined(__AVR_ATmega2560__) // Arduino MEGA2560
#define bc95serial Serial1
#endif

#include <Arduino.h>
#include "DHT.h"
#include "BC95Udp.h"
#include "MicrogearNB.h"
// กำหนดข้อมูลเชื่อมต่อ NETPIE
#define APPID "YOUR_APPID" // ต้องแก้ไข
#define KEY "YOUR_KEY" // ต้องแก้ไข
#define SECRET "YOUR_SECRET" // ต้องแก้ไข

// กำหนดข้อมูลเชื่อมต่อ DHT22
#define DHTPIN 7
#define DHTTYPE DHT22
DHT dht(DHTPIN, DHTTYPE);

BC95UDP client;
Microgear mg(&client);

#define MAXBUFSIZE 32
char payload[MAXBUFSIZE];
long lastDHTRead = 0;

void setup() {
  bc95serial.begin(9600);
  BC95.begin(bc95serial);
  BC95.reset();

  Serial.begin(9600);
  Serial.println(F("Microgear Arduino NB-IoT Start!"));
  Serial.print(F("IMEI: "));
  Serial.println(BC95.getIMEI());
  Serial.print(F("IMSI: "));
  Serial.println(BC95.getIMSI());

  Serial.print(F("Attach Network..."));
  while (!BC95.attachNetwork()) {
    Serial.print(".");
    delay(1000);
  }

  Serial.println(F("\nNB-IOT attached!"));
```



```

Serial.print(F("RSSI: "));
// ค่าความแรงสัญญาณ NB-IoT
Serial.println(BC95.getSignalStrength());
Serial.print(F("IPAddress: "));
Serial.println(BC95.getIPAddress());

mg.init(APPID, KEY, SECRET);
mg.begin(5555);
}

void loop() {
// เซนเซอร์ DHT อ่านถี่เกินไปไม่ได้ จะให้ค่า error
// จึงต้องเช็คเวลาครั้งสุดท้ายที่อ่านค่าว่าทิ้งช่วงนานพอหรือยัง ในที่นี้ตั้งไว้ 1 วินาที
if(millis() - lastDHTRead > 1000){
float humid = dht.readHumidity(); // อ่านค่าความชื้น
float temp = dht.readTemperature(); // อ่านค่าอุณหภูมิ
lastDHTRead = millis();

Serial.print("Humid: "); Serial.print(humid); Serial.print(" %, ");
Serial.print("Temp: "); Serial.print(temp); Serial.println(" C ");

// ตรวจสอบค่า humid และ temp เป็นตัวเลขหรือไม่
if (isnan(humid) || isnan(temp)) {
Serial.println("Failed to read from DHT sensor!");
}
else {
// เตรียมสตริงในรูปแบบ "humid,temp"
String datastring = (String)humid + "," + (String)temp;

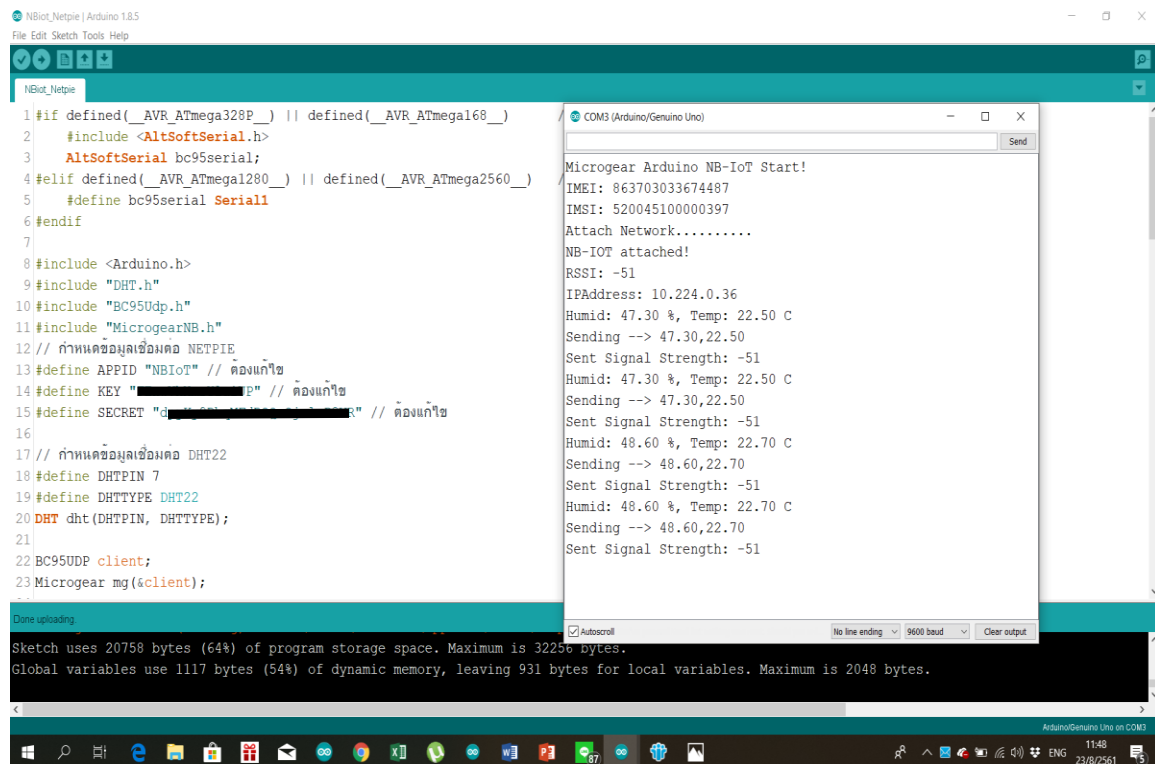
Serial.print(F("Sending --> "));
Serial.println(datastring);
datastring.toCharArray(payload, MAXBUFSIZE-1);
mg.publish("/nbiot/sensor", payload);
}

Serial.print(F("Sent Signal Strength: "));
Serial.println(BC95.getSignalStrength());
mg.publish("/nbiot/rssi", BC95.getSignalStrength());
}

mg.loop();
}

```

เสร็จแล้ว Upload และกด Serial Monitor ดู ถ้าสมบูรณ์และไม่มีอะไรผิดพลาด ก็จะได้ผลลัพธ์แบบนี้



The screenshot shows the Arduino IDE interface. The main window displays the source code for the NetPie sketch, which includes headers for `AltSoftSerial`, `DHT`, and `MicrogearNB`, and defines for APPID, KEY, and SECRET. The code is being uploaded to an Arduino Uno. The Serial Monitor window on the right shows the output of the sketch, which includes the Microgear Arduino NB-IoT Start! message, IMEI and IMSI numbers, network attachment status, and sensor readings for humidity and temperature.

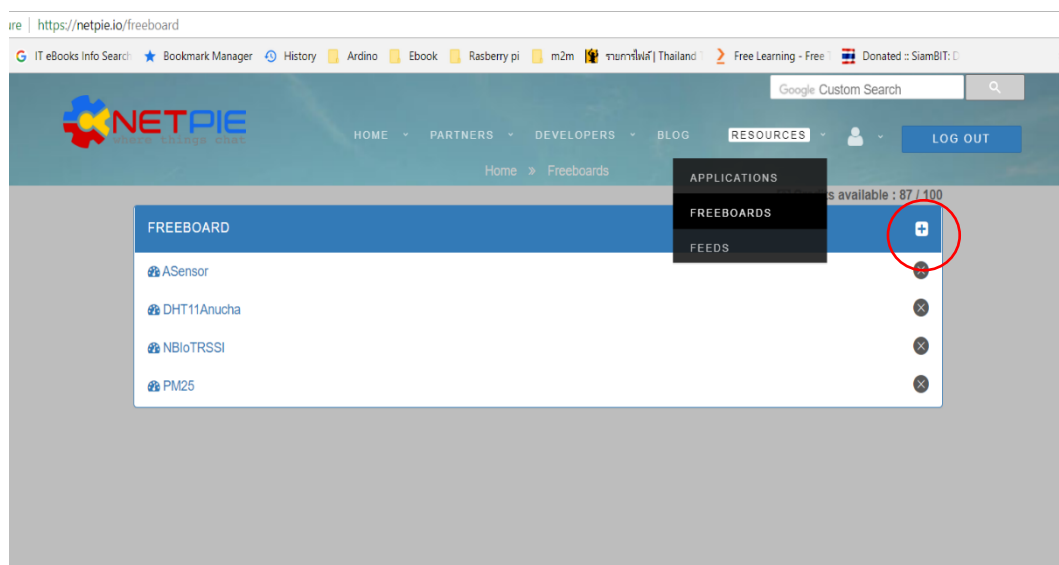
```
1 #if defined( _AVR_ATmega328P_ ) || defined( _AVR_ATmega168_ )
2   #include <AltSoftSerial.h>
3   AltSoftSerial bc95serial;
4 #elif defined( _AVR_ATmega1280_ ) || defined( _AVR_ATmega2560_ )
5   #define bc95serial Serial1
6 #endif
7
8 #include <Arduino.h>
9 #include "DHT.h"
10 #include "BC95Udp.h"
11 #include "MicrogearNB.h"
12 // กำหนดข้อมูลเชื่อมต่อ NETPIE
13 #define APPID "NBioT" // ต้องแก้ไข
14 #define KEY "XXXXXXXXXX" // ต้องแก้ไข
15 #define SECRET "XXXXXXXXXX" // ต้องแก้ไข
16
17 // กำหนดข้อมูลเชื่อมต่อ DHT22
18 #define DHTPIN 7
19 #define DHTTYPE DHT22
20 DHT dht(DHTPIN, DHTTYPE);
21
22 BC95UDP client;
23 Microgear mg(client);
```

Serial Monitor Output:

```
Microgear Arduino NB-IoT Start!
IMEI: 863703033674487
IMSI: 520045100000397
Attach Network.....
NB-IoT attached!
RSSI: -51
IPAddress: 10.224.0.36
Humid: 47.30 %, Temp: 22.50 C
Sending --> 47.30,22.50
Sent Signal Strength: -51
Humid: 47.30 %, Temp: 22.50 C
Sending --> 47.30,22.50
Sent Signal Strength: -51
Humid: 48.60 %, Temp: 22.70 C
Sending --> 48.60,22.70
Sent Signal Strength: -51
Humid: 48.60 %, Temp: 22.70 C
Sending --> 48.60,22.70
Sent Signal Strength: -51
```

ตั้งค่าการแสดงผลบนเว็บ

1. เพิ่ม Datasource ด้วยการ เข้าไปที่ RESOURCES → FREEBOARDS แล้วกด + ตั้งชื่อ และ CREATE



2. s]y'0kd ตั้งชื่อ กด ADD และใส่ข้อมูล ตามรูป โดยเลือก (1) เป็น NETPIE Microgear (2) ชื่อ และ (3) APP ID (4) KEY (5) SECRET โดย ข้อมูล (3),(4),(5) ตามที่ได้ระบุใน Arduino Code

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 (1) ▼

NAME: microgear (2)

APP ID: ใส่ appid จาก netpie.io (3)
NETPIE App ID obtained from https://netpie.io/app

KEY: ใส่ key จาก netpie.io (4)
Key

SECRET: ใส่ secret จาก netpie.io (5)
Secret

SUBSCRIBED TOPICS: /#

หลังจากใส่แล้วกด SAVE

TYPE: NETPIE Microgear ▼

NAME: microgear

APP ID: ใส่ appid จาก netpie.io
NETPIE App ID obtained from https://netpie.io/app

KEY: ใส่ key จาก netpie.io
Key

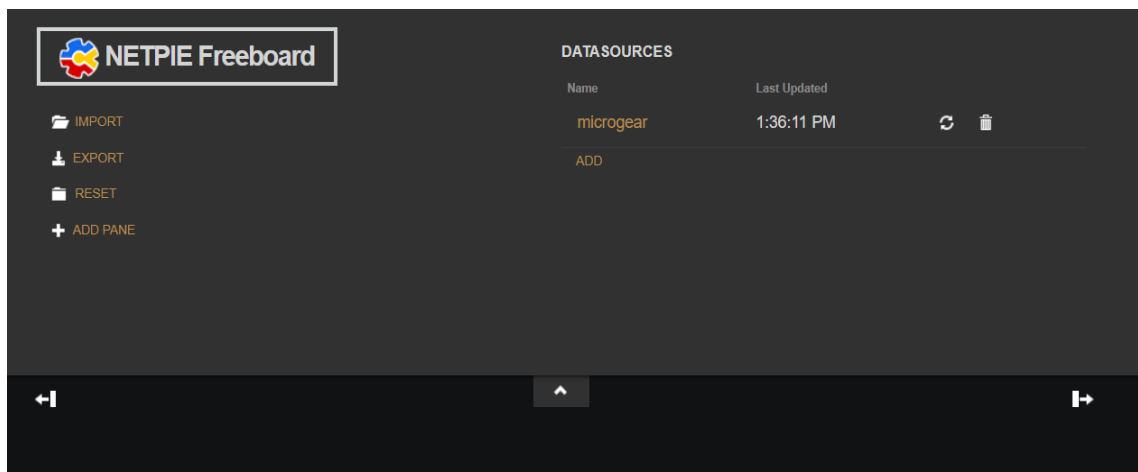
SECRET: ใส่ secret จาก netpie.io
Secret

SUBSCRIBED TOPICS: /#
Topics of the messages that this datasource will consume, the default is /# which means all messages in this app ID.

ON ACTION:
JS code to run after a datasource is created

ON ACTION:
JS code to run after a microgear datasource is connected to NETPIE

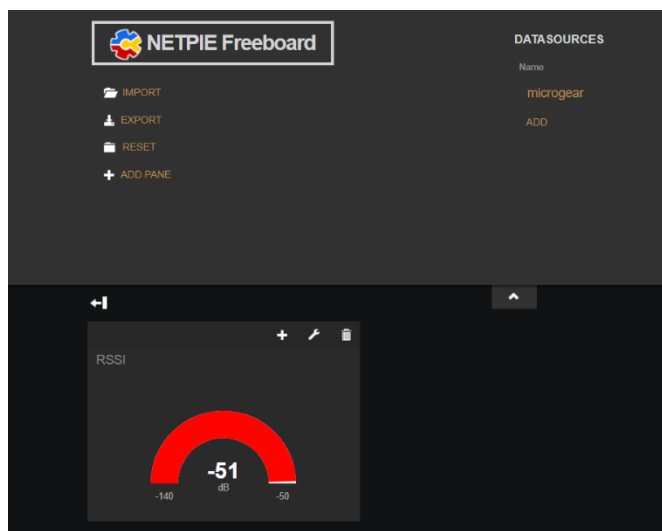
SAVE CANCEL



3. เพิ่ม Gauge แสดงค่า RSSI จาก NB-IoT ด้วยการกด ADD PANE ใส่ค่า VALUE ให้กด +DATASOURCE `datasources["microgear"]["NBIoT/nbiot/rssi"]` ตาม APP ID ที่ท่านตั้งชื่อไว้ กด SAVE

The screenshot shows the 'WIDGET' configuration form. The 'TYPE' is set to 'Gauge'. The 'TITLE' is 'RSSI'. The 'VALUE' field contains the expression `datasources["microgear"]["NBIoT/nbiot/rssi"]`, with a '+ DATASOURCE' button next to it. The 'UNITS' are 'dB'. The 'MINIMUM' is '-140' and the 'MAXIMUM' is '-50'. At the bottom right are 'SAVE' and 'CANCEL' buttons.

จะได้รูปแสดงค่า RSSI



4. เพิ่ม Gauge แสดงค่า อุณหภูมิและความชื้น จาก NB-IoT ด้วยการกด ADD PANE ตั้งชื่อ และเพิ่ม DATASOURCE เนื่องจากข้อความอุณหภูมิและความชื้นมาคู่กัน เช่น 27.61,60.10 แต่เราอยากได้แค่ อุณหภูมิอย่างเดียว คือ 27.61 จึงจำเป็นต้องใช้ split เพื่อแยกข้อความ 2 ตัวที่เชื่อมกันด้วยเครื่องหมาย , ออกมา

DATASOURCE ของ อุณหภูมิ เป็น `datasources["microgear"]["NBloT/nbiot/sensor"].split(',')[1]`

WIDGET

TYPE: Gauge

TITLE: Teperature

VALUE: `datasources["microgear"]["NBloT/nbiot/sensor"].split(',')[1]` + DATASOURCE .JS EDITOR

UNITS: C

MINIMUM: 0

MAXIMUM: 100

SAVE CANCEL

DATASOURCE ของ ความชื้น เป็น `datasources["microgear"]["NBloT/nbiot/sensor"].split(',')[0]`

WIDGET

TYPE: Gauge

TITLE: Humid

VALUE: `datasources["microgear"]["NBloT/nbiot/sensor"].split(',')[0]` + DATASOURCE .JS EDITOR

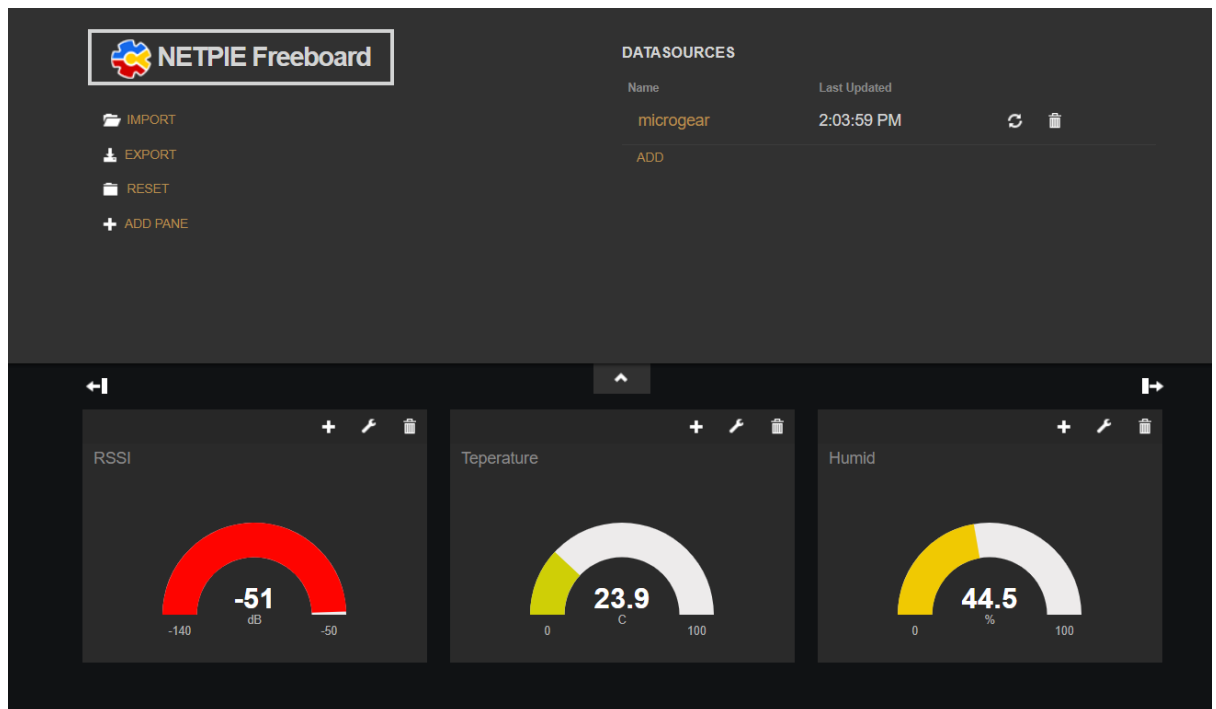
UNITS: %

MINIMUM: 0

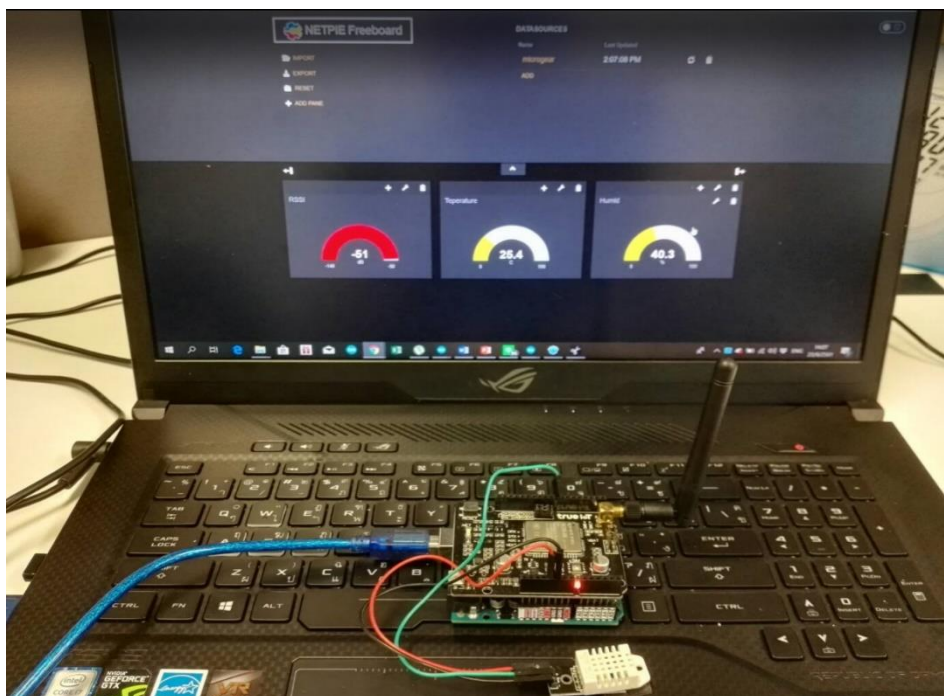
MAXIMUM: 100

SAVE CANCEL

เมื่อกด SAVE ก็จะได้อุปกรณ์ส่งค่าความแรงสัญญาณ, อุณหภูมิ และความชื้น ขึ้น Internet เปิดที่ไหนบนโลกก็ได้แล้วตอนนี้



หวังว่าจะมีประโยชน์กับหลาย ๆ ท่านเพื่อนำไปพัฒนาเป็นอุปกรณ์ IoT ที่ตอบโจทย์ผู้ใช้งานยิ่ง ๆ ขึ้นไป และหลากหลายยิ่งขึ้น



ข้อมูลอ้างอิง <https://blog.netpie.io/archives/3313>

ขอบคุณครับ