

ติดตั้ง Arduino IDE & Library

ขั้นตอน

1. ลงโปรแกรม ขั้นต้นปกติ next install
2. วิธีเพิ่มบอร์ด esp8266
 - Click File >> Preferences
 - ใส่ URLs ในช่อง additional boards Manager URLs:
http://arduino.esp8266.com/stable/package_esp8266com_index.json
 - Tools >> Board >> Board Manager >> search esp8266 แล้ว install
3. วิธีเลือกบอร์ด Tools >> Board >> ถ้า esp ก็เลือก esp8266-12e ถ้า Arduino ก็ดูว่า เป็น UNO หรือ UNO Wifi
4. วิธีดู Com Port คลิกขวา my com >> Manage >> device manager >> Ports
5. ลง Library
 - Sketch >> Include Library >> Manage Library >> search ชื่อ แล้ว install
6. กรณีที่ search library แล้วไม่เจอให้ทำการหาโหลด ส่วนใหญ่จะอยู่ใน git hub
 - Sketch >> Include Library >> Add.Zip Library >> เลือกไฟล์ Library ที่โหลดมา

Library ส่วนใหญ่ที่ต้องลง (ดูตอนที่ compile ว่าขาดเหลืออะไรก็ลงเพิ่มเอาเน้อ)

DHT	Adafruit_Sensor.h	Adafruit_Sensor-master
IoT Tweet	ArduinoJson	Adafruit_LiquidCrystal
Thingspeak	Microgear	

ถ้าลงแล้วยัง error ให้เข้าไปที่ document => arduino => libraries แล้วลบลงใหม่

NodeMCU wiith IoT TWEET

1. เข้า web >> Dashboard >> My IoT garage
2. Add your IoT device (ตั้งชื่อโปรเจค) >> IoT model. (ตั้งชื่ออุปกรณ์ที่นำมาเชื่อมต่อ)
3. กด Add จะได้ key มาเอาไว้ใช้เชื่อมต่อใน code
4. ส่วนที่ใช้ใน code เพื่อเชื่อมต่อเพื่อยิงข้อมูลขึ้น iot tweet จะมี 2 ส่วน คือ key กับ UserID ที่ได้มาตอนสมัคร เมื่อ login เข้า web แล้ว UserID จะอยู่มุมซ้ายบน
5. เมื่อสมัครเสร็จแล้ว ไปที่ dashboard เลือก My IoT device เป็นชื่อโปรเจคที่เราจะใช้
6. เมื่อเลือกแล้วกด edit layout create สิ่งที่จะใช้ เลือก slot ว่าจะใช้แสดงข้อมูลอะไร
7. กรณียิงข้อมูลจาก nodeMcu ขึ้นไปด้วย code บรรทัดนี้

String response =

```
myiot.WriteDashboard(userid,key,data0,data1,data2,data3,private_tweet,public_tweet);
```

ยกตัวอย่าง

Data0 เป็นค่า temp ดังนั้นค่า slot0 ที่เลือกบน iot tweet ก็จะเป็นค่า temp

Data1 เป็นค่า Hum ดังนั้นค่า slot1 ที่เลือกบน iot tweet ก็จะเป็นค่า Hum

NodeMCU with Thingspeak

1. เข้า web >> New channel >> ตั้งชื่อ >> เลือกว่าจะใช้ field เท่าไร เช่นส่งข้อมูล temp กับ hum จาก nodeMCU มาเพื่อใช้แสดงก็ใช้ 2 field << Save channel

The screenshot shows the 'New Channel' page on the Thingspeak website. The form has the following fields:

- Name: Input field with 'IoTGarage' entered.
- Description: Text area.
- Field 1: Input field with 'Temperature' entered.
- Field 2: Input field with 'Humidity' entered.
- Field 3: Input field (empty).
- Field 4: Input field (empty).
- Field 5: Input field (empty).

On the right, the 'Channel Settings' section provides instructions:

- Channel Name: Enter a unique name for the Thingspeak channel.
- Description: Enter a description of the Thingspeak channel.
- Fields: Check the box on the left of the field, and enter a field name. Each Thingspeak channel can have up to 5 fields.
- Metadata: Enter information about the channel data, including units, ID, or ID3 tags.
- Tags: Enter keywords that identify the channel. Separate tags with commas.
- Latitude: Specify the latitude of the sensor or thing that collects data to display. For example, the latitude of the city of London is 51.507.

2. คลิกเข้าไปที่ channel ที่เราสร้างจากนั้น กด API Keys

3. ถ้าจะนำค่าจาก nodeMcu ไปมอเน็ตอร์ผ่าน Thingspeak ให้ใช้ write API Keys
ถ้าจะสั่งการทำงานจาก thingspeak ไปยัง controller จะใช้ read API Keys
เพราะฉะนั้น ให้ ใช้ WRITE API Keys ใส่ใน code ฝั่ง nodeMcu

```
static const char* host = "api.thingspeak.com";
```

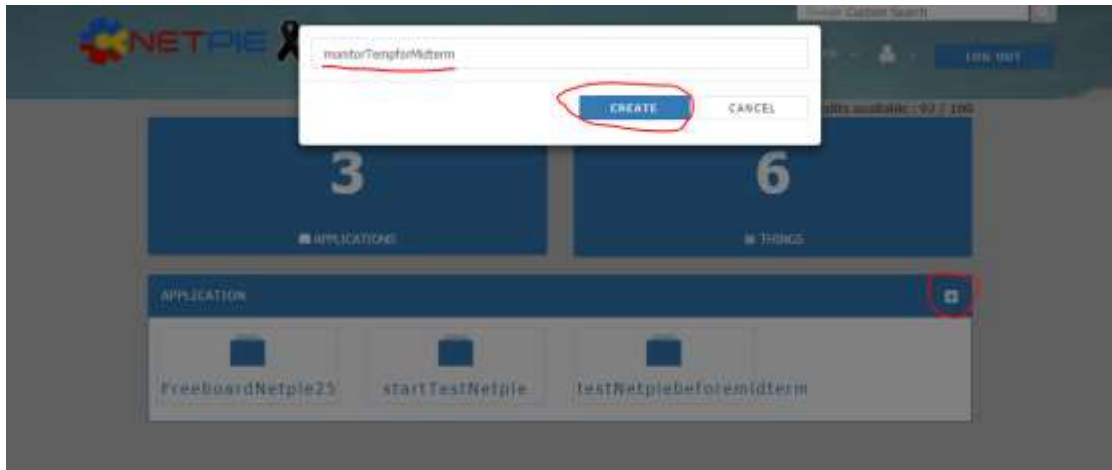
```
static const char* apiKey = "Api keys"
```

The screenshot shows the 'API Keys' tab in the Thingspeak web interface. It has two main sections: 'Write API Key' and 'Read API Keys'. The 'Write API Key' section contains a text input field with the key 'HA5D6XLK1QU0...', a 'Generate New Write API Key' button, and a 'Help' sidebar. The 'Read API Keys' section contains a text input field with the key 'GQIB4I92J48U...', a 'Note' input field, 'Save Note' and 'Delete API Key' buttons, and a 'Generate New Read API Key' button. The right sidebar also contains 'API Keys Settings' and 'API Requests' sections.

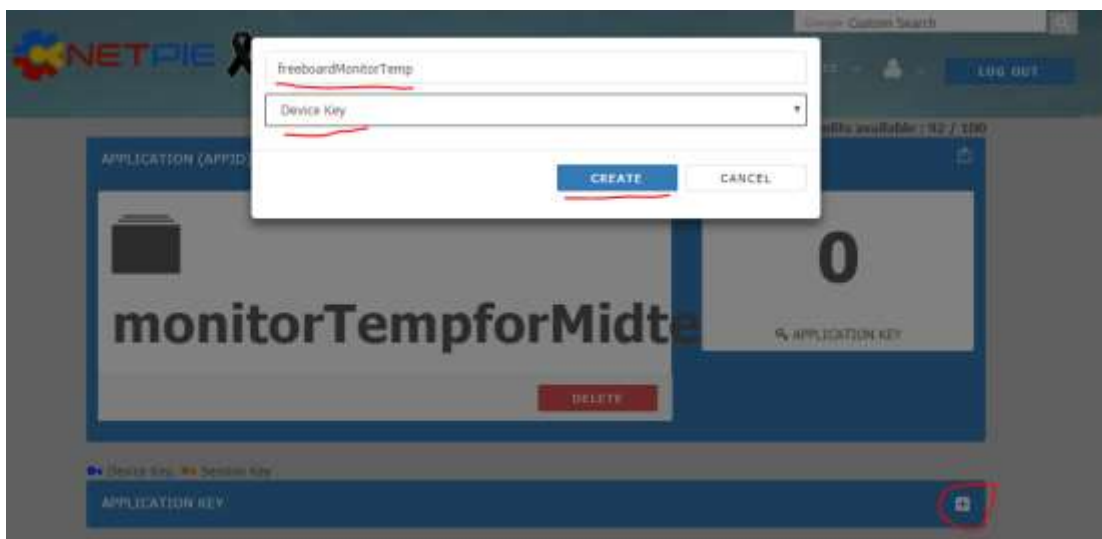
4. code upload data ไปยัง thingspeak คือบรรทัดนี้ `uploadThingsSpeak(t_dht, h_dht);`

NodeMCU wiith Netpie

1. Resource >> Application >> กด + เพื่อ add Application
2. ตั้งชื่อ Application ชื่อส่วนนี้ ก็คือ AppID ซึ่งห้ามซ้ำนั่นเอง แล้วกด create



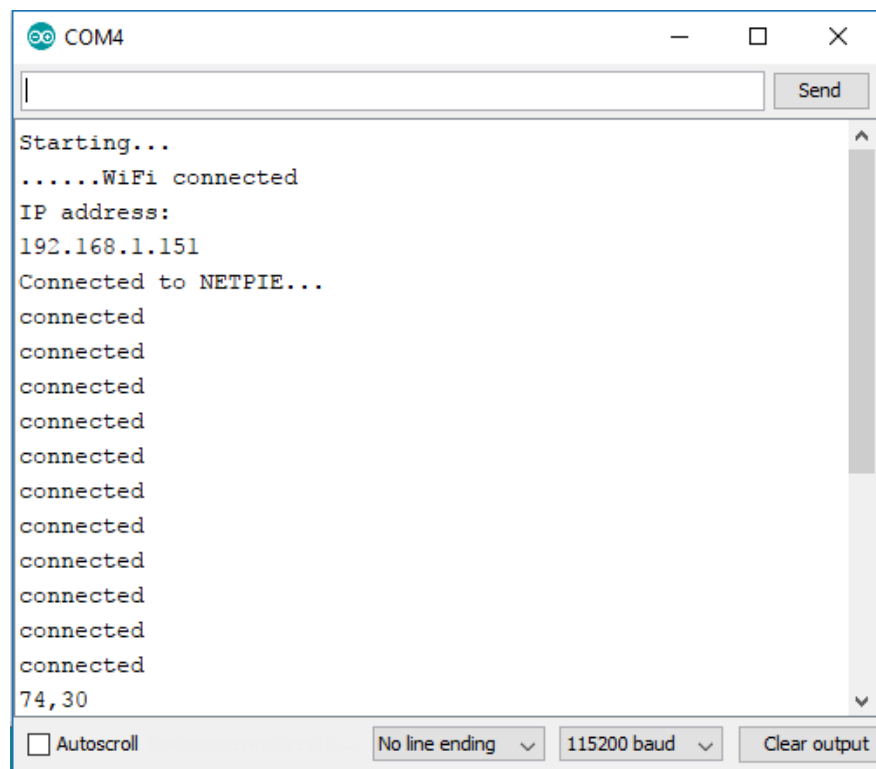
3. กด + เพิ่ม Application key >> ตั้งชื่อ Applicationkey >> เลือก Device key
 - Device key ใช้สำหรับการรับส่งค่าระหว่าง netpie กับ อุปกรณ์ controller
 - Session key ใช้สำหรับ HTML




```
#define APPID "AppID" //*****Change*****
#define KEY "" //*****Change*****
#define SECRET "" //*****Change*****
#define ALIAS ""
```

6. เปลี่ยน พวก wifi ให้เรียบร้อยแล้ว เปริน code ลง esp เลย ถ้า code appid key secret ไม่ผิด และลง Library ครบ

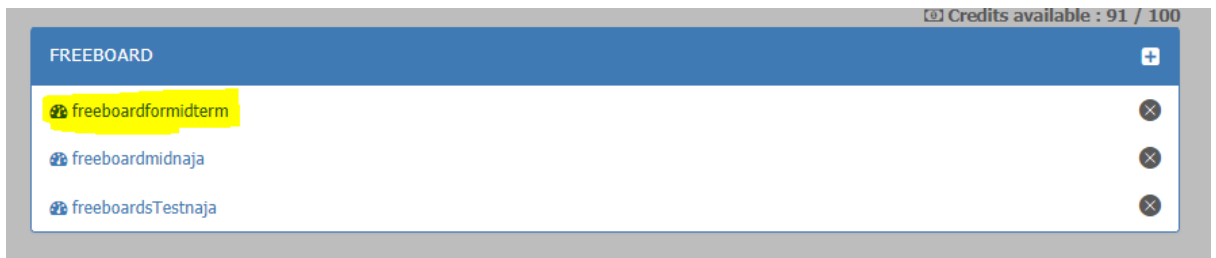
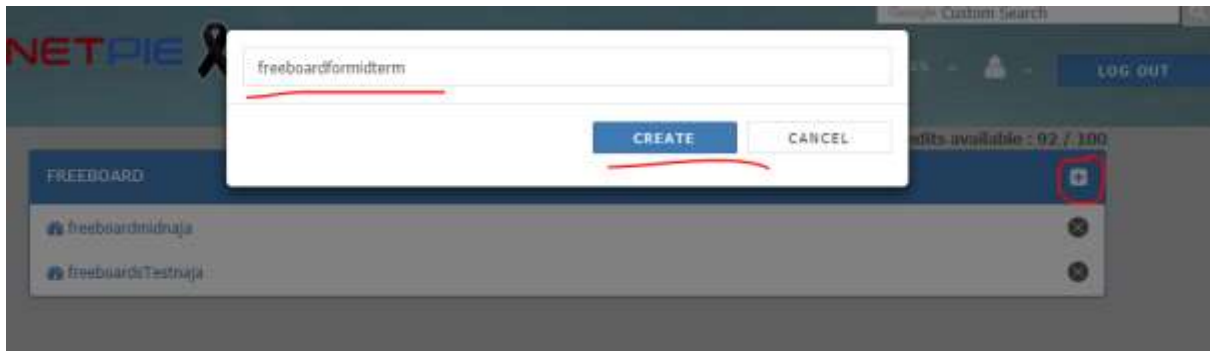
ดูใน monitor serial จะขึ้นแบบนี้



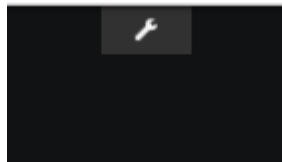
ดูตรง Application KEY จะขึ้น online 1



7. Resource >> freedoards >> กด + เพื่อเพิ่ม freeboard >> ตั้งชื่อ >> Create >> พอได้ freeboard มาแล้วคลิกเข้าไป



8. จะตั้ง tap ใหม่มา เป็นหน้า free board สีดำ สังเกตรูปประแจแล้ว click ที่รูป bar setting จะเลื่อนลงมา >> คลิก ADD เพื่อกำหนด Datasource



9. เลือก type เป็น netpie microgear

Name อะไรก็ได้

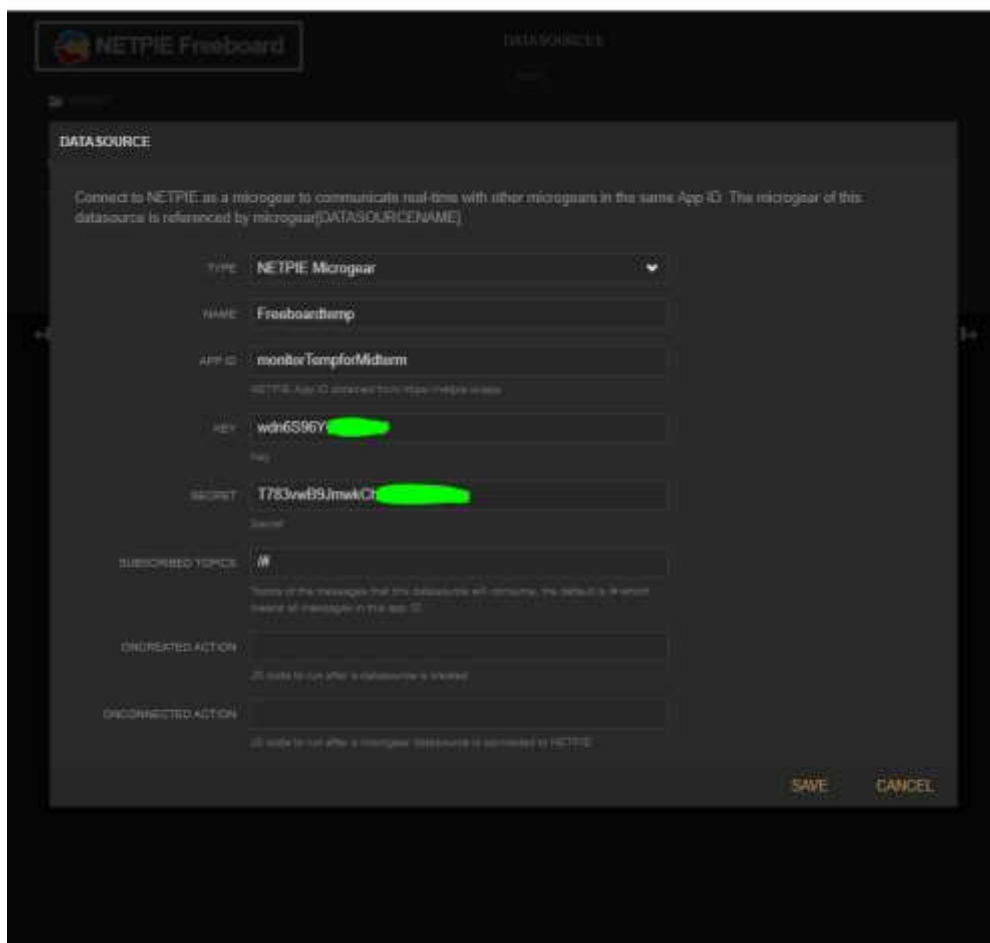
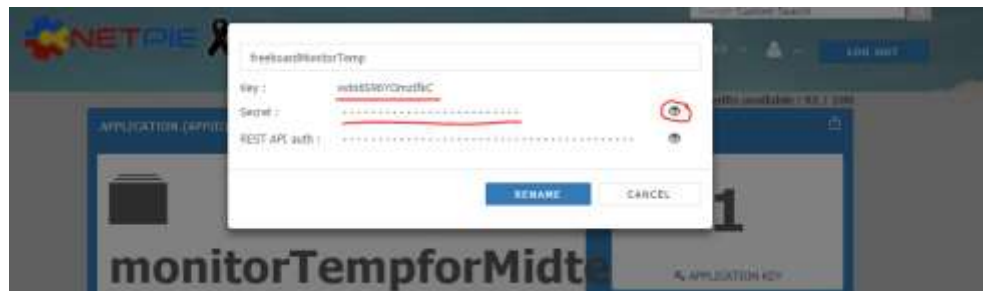
App ID ใน nodeMcu นั้นเอง

Key ของ Application Key อันเดียวกับที่ใช้ใส่ code ใน nodeMcu นั้นเอง

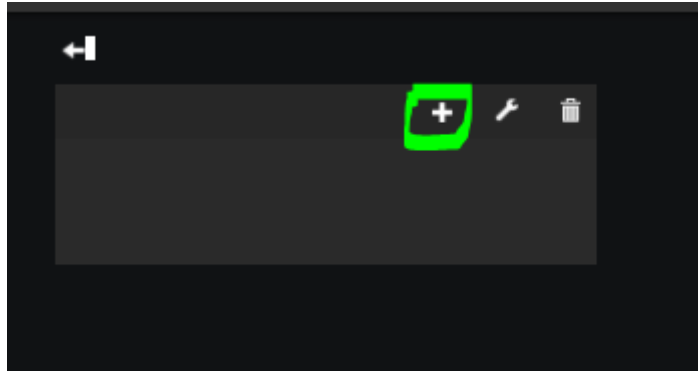
Secret ของ Application Key ที่อันเดียวกับที่ใช้ใส่ code ใน nodeMcu นั้นเอง

/# เอาไว้เหมือนเดิม

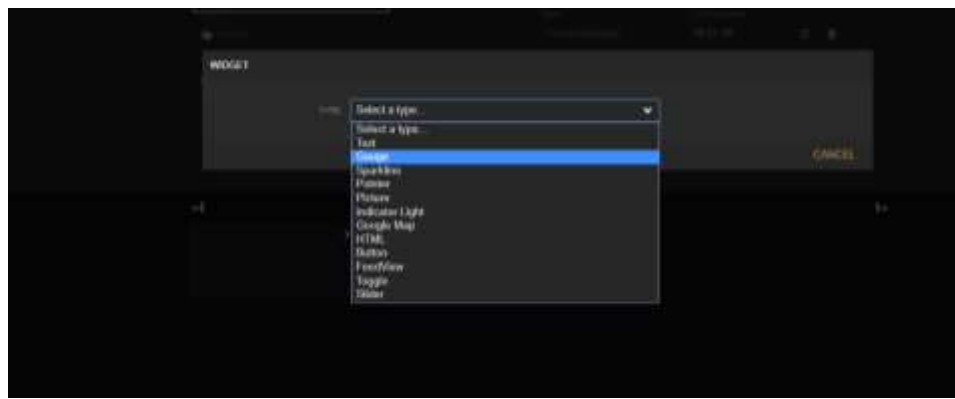
save



10. กด +ADD PANE จะขึ้น box มา บน freeboard >> กด +



11. ตัวอย่าง select type เป็น อะไรก็ได้ยกตัวอย่างเลือก gauge



12. ตั้ง Title ที่จะแสดง

ตั้ง value ex.

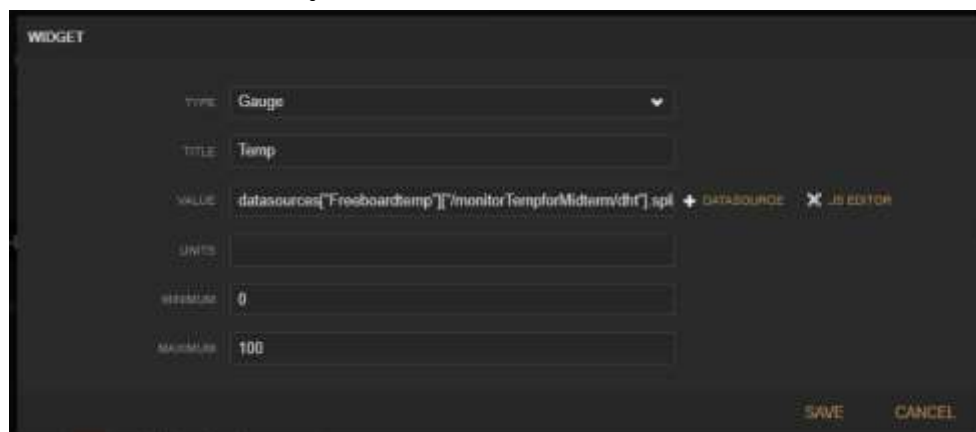
`datasources["Freeboardtemp"]["/monitorTempforMidterm/dht"].split(",")[0]`

○ `datasource["ชื่อdatasource"]["/AppID/dht"].split(",")[1]` **save**

○ [0] คือ array ช่องที่ 0 ที่ส่งค่ากลับมา คือค่า hum

[1] คือ array ช่องที่ 1 ที่ส่งค่ากลับมา คือค่า temp

ตรง /dht ให้ดูจาก code



13. Success ถ้าไม่ขึ้นให้ทำการเปลี่ยนชื่อ Alias ใน code แล้ว เบิร์นลงบอร์ดใหม่ บางครั้งอาจจะบัค.

