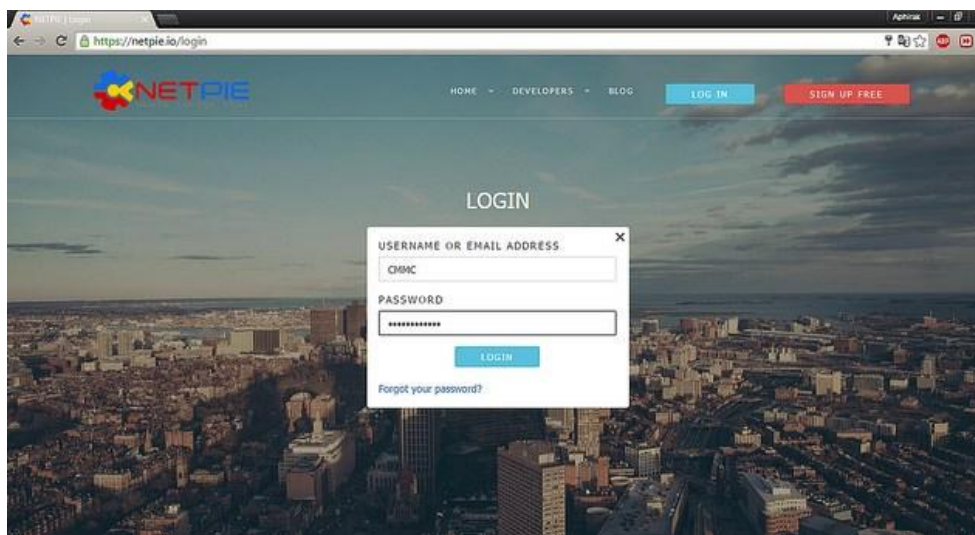


NETPIE.io คืออะไร

“NETPIE แพลตฟอร์ม IoT เพื่อนักพัฒนาและอุตสาหกรรมไทย” ภาควิทยาศาสตร์และเทคโนโลยีโดย ศูนย์เทคโนโลยีอิเล็กทรอนิกส์และคอมพิวเตอร์แห่งชาติ (NECTEC) ได้เปิดตัวแพลตฟอร์มสำหรับนักพัฒนาซึ่งมีชื่อว่า NETPIE โดยย่อมาจาก Network Platform for Internet of Everything โดยเป็นคลาวด์เซิร์ฟเวอร์ที่ให้บริการในรูปแบบ Platform as a Service สำหรับติดต่อสื่อสาร แลกเปลี่ยนข้อมูลระหว่างอุปกรณ์ต่าง ๆ เหมาะสำหรับการพัฒนา IoT ในปัจจุบัน โดยรองรับการเชื่อมต่อผ่านอุปกรณ์ต่าง ๆ อาทิเช่น Arduino, Raspberry Pi, ESP8266, HTML5 และ Node JS เป็นต้น NETPIE นั้นยังช่วยลดภาระด้านความปลอดภัยโดยผู้พัฒนาสามารถกำหนด หรือออกแบบได้เองทั้งหมด ดังนั้นแล้ว NETPIE จึงเป็นอีกทางเลือกหนึ่งสำหรับนักพัฒนา IoT ในยุคปัจจุบัน ซึ่งขั้นตอนการสมัครใช้งานมีดังต่อไปนี้

1) ลงชื่อเข้าใช้งาน netpie.io



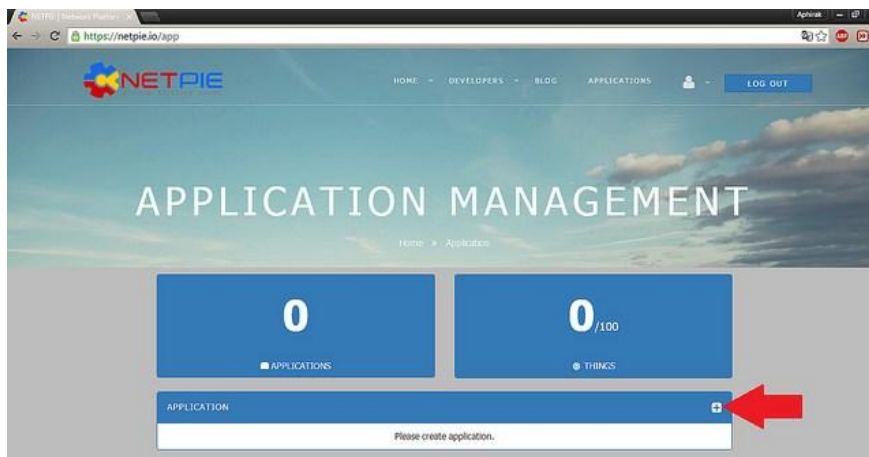
2) คลิกที่ APPLICATIONS เพื่อเข้าใช้งาน



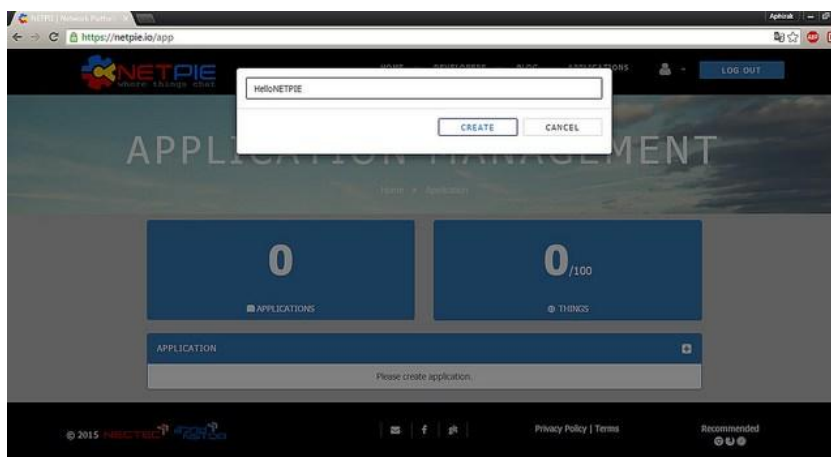
ชมรมเชียงใหม่เมกเกอร์คลับ (Chiang Mai Maker Club)

81/21-25 ถ.อารักษ์ ต.สุเทพ อ.เมืองเชียงใหม่ จ.เชียงใหม่ 50200

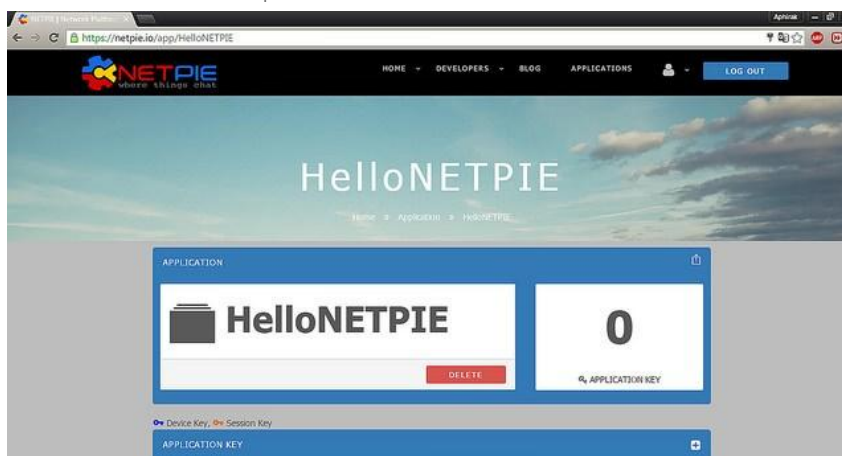
3) คลิกที่เครื่องหมาย + เพื่อสร้าง APPLICATIONS



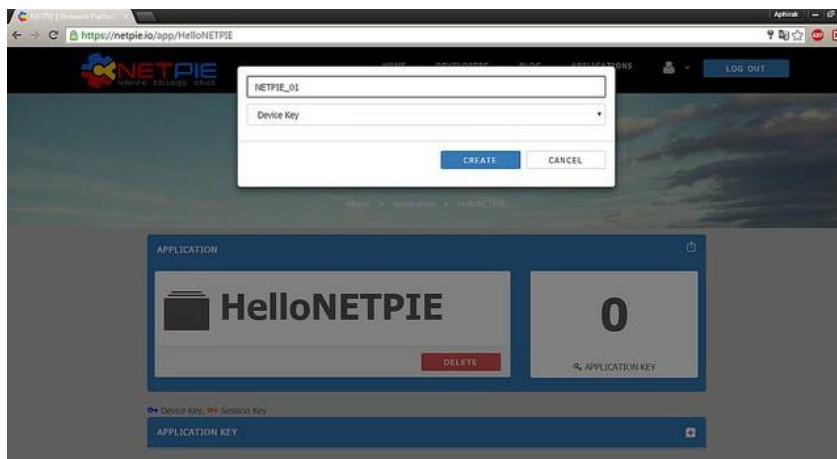
4) สร้างชื่อ APPLICATIONS (ชื่อที่สร้างขึ้นมานี้ก็คือ APPID)



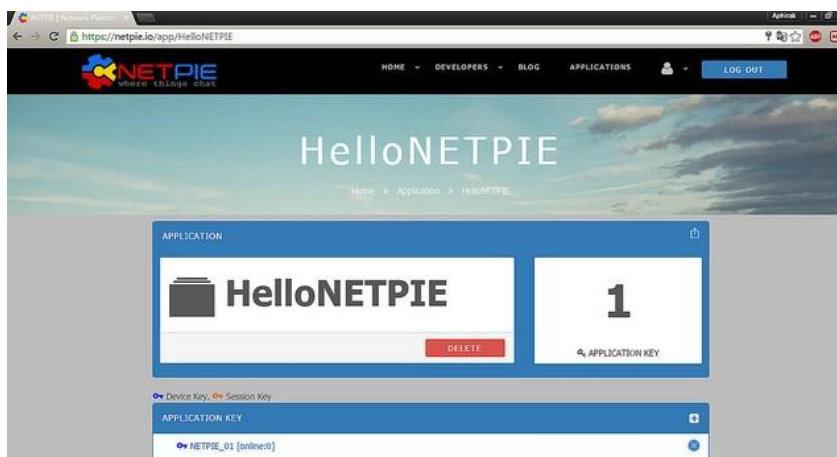
5) แสดงชื่อ APPLICATIONS หรือ APPID ซึ่งตอนนี้ยังไม่มี APPLICATION KEY ให้เราทำการสร้างโดยการคลิกที่เครื่องหมาย + (มุมล่างขวา)



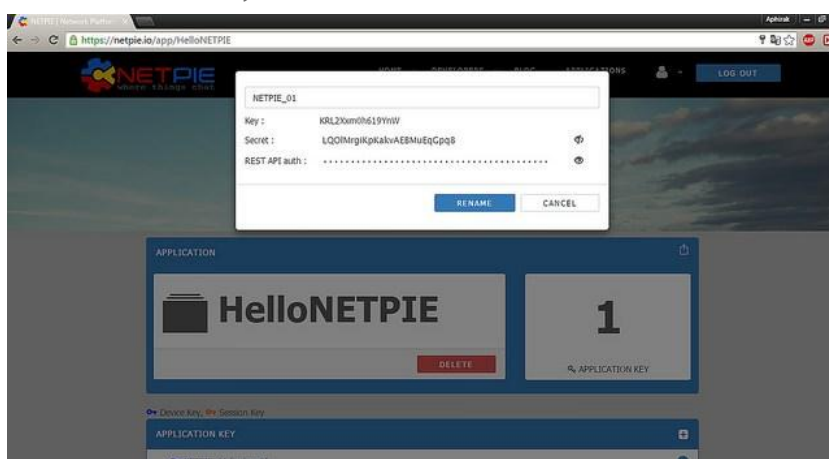
6) ทำการสร้าง Device Key



7) เมื่อสร้างเสร็จจะเห็น Device Key ตามชื่อที่เราสร้างสีฟ้า ๆ ด้านล่าง



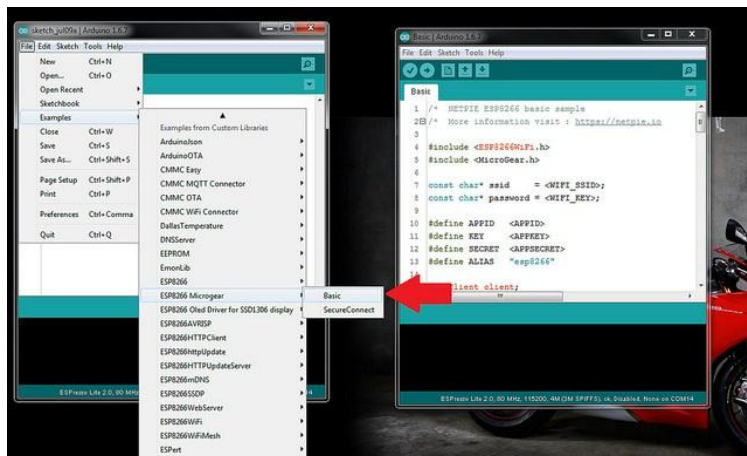
8) คลิกที่ Device Key จะแสดง KEY และ SECRET เพื่อใช้ในการเชื่อมต่อกับ netpie.io



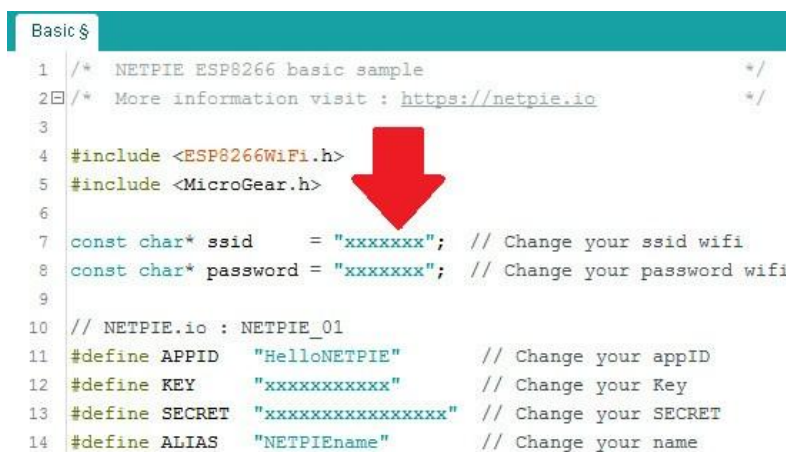
9) ทำการสร้าง Session Key เพื่อใช้ในการเชื่อมต่อกับ Dashboard (Device Key สีฟ้า, Session Key สีส้ม)



10) เปิดโปรแกรม Arduino IDE —> Example —> ESP8266 Microgear —> Basic (หากยังไม่มีให้ทำการติดตั้งไลบรารี Microgear v 1.1.7)



11) ทำการแก้ไขการเชื่อมต่อไวไฟ และการเชื่อมต่อ netpie.io โดยสามารถดูค่า KEY กับ SECRET ได้จาก Device Key



ชมรมเชียงใหม่เมกเกอร์คลับ (Chiang Mai Maker Club)


81/21-25 ถ.อารักษ์ ต.สุเทพ อ.เมืองเชียงใหม่ จ.เชียงใหม่ 50200

12) อัปโหลดโค้ดลงบอร์ด ESPresso Lite แล้วเปิด Serial port จะแสดงการเชื่อมต่อ และข้อความที่ส่ง

```

99 void loop() {
100   /* To check if the microgear is still connected */
101   if (microgear.connected()) {
102     Serial.println("connected");
103
104     /* Call this method regularly otherwise the connection may be lost */
105     microgear.loop();
106
107     if (timer >= 1000) {
108       Serial.println("Publish...");
109
110       /* Chat with the microgear named ALIAS which is myself */
111       microgear.chat(ALIAS, "Hello");
112       timer = 0;
113     }
114     else timer += 100;
115   }
116   else {
117     Serial.println("connection lost, reconnect...");
118     if (timer >= 5000) {
119       microgear.connect(APPID);
120       timer = 0;
121     }
122     else timer += 100;
123   }
124   delay(100);
125 }


```



13) กลับไปยัง netpie.io หากมีการเชื่อมต่อจะแสดงชื่อ ALIAS ที่ถูกส่งข้อมูลขึ้นมา



14) ทดลองใช้ MQTT Websocket Client เพื่อแสดงข้อมูลที่ถูกส่งขึ้นมา โดยกรอก APP ID, APP KEY, APP SECRET ให้ถูกต้องแล้วกด Connect (APP KEY, APP SECRET ให้ใช้ของ Session Key)


Powered by HiveMQ WebSocket Client

Connection

APP ID
HelloNETPIE

APP KEY
APP KEY

APP SECRET
APP SECRET

Connect

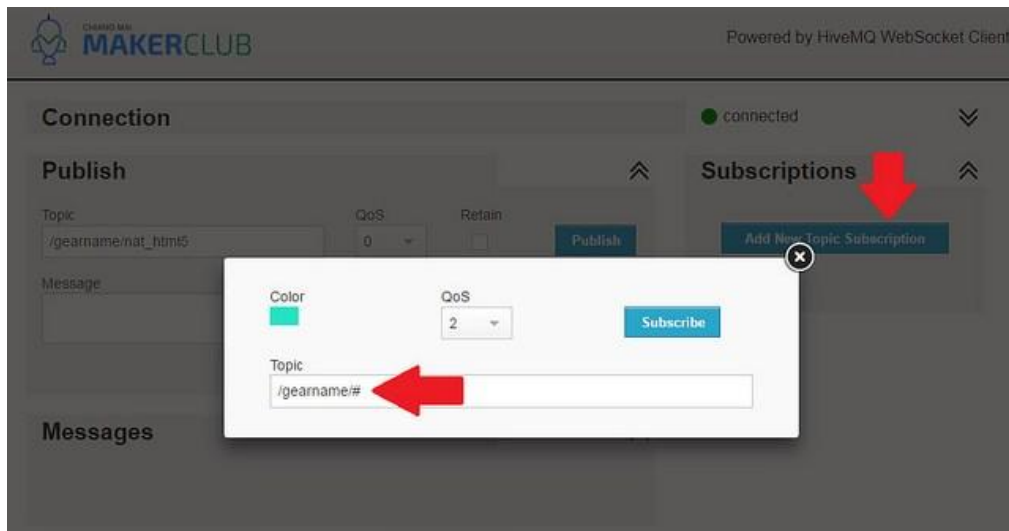
Publish

Messages

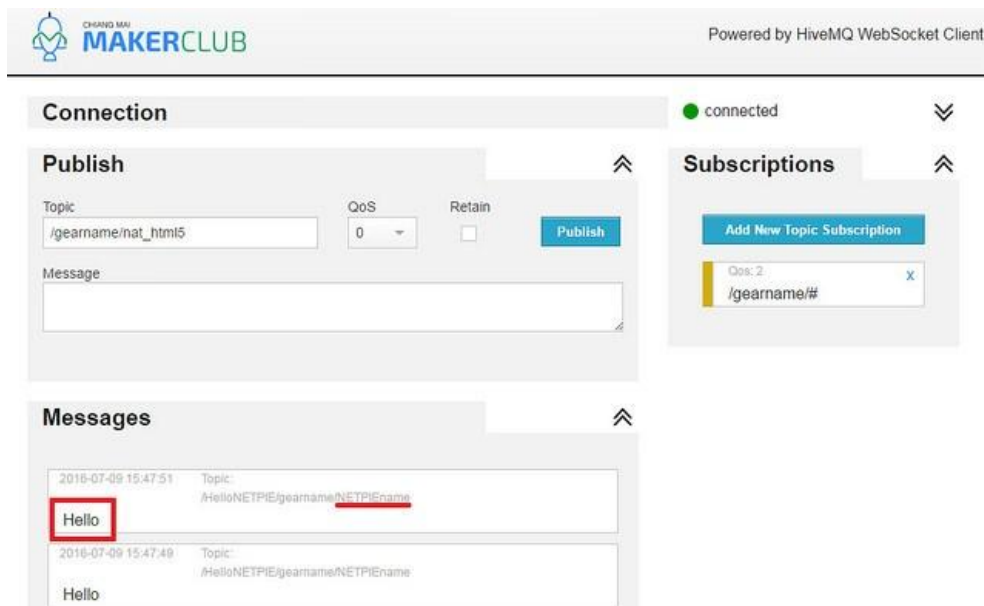
Subscriptions

Messages

15) คลิก Add New Topic Subscription แล้วเลือกดูทั้งหมดโดยพิมพ์ /gearname/#



16) เมื่อมีการส่งข้อมูล จะแสดงข้อความที่ถูกส่งว่ามาจากใคร



17) ลำดับต่อไป ทดลองใช้ Dashboard โดยเข้าไปที่ [Dashboard](#)

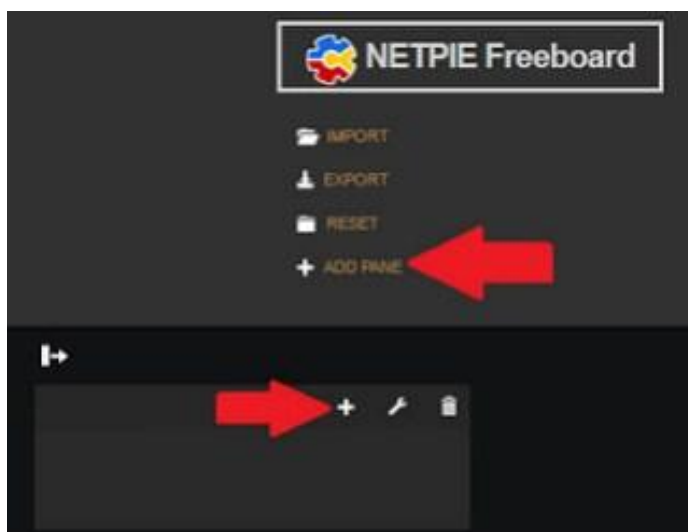


18) คลิก ADD จะแสดงหน้าจอตั้งรูปแบบด้านล่าง กรอกข้อมูลให้ครบถ้วนโดย KEY และ SECRET จะได้จาก Session Key ของเว็บ netpie.io ส่วน DEVICE ALIAS จะต้องใช้ชื่อเดียวกับที่ส่งเข้ามายัง Device Key ที่ต้องการเชื่อมต่อ และ MICROGEAR REFERENCE จะใช้สำหรับอ้างอิงข้อมูลใน dashboard

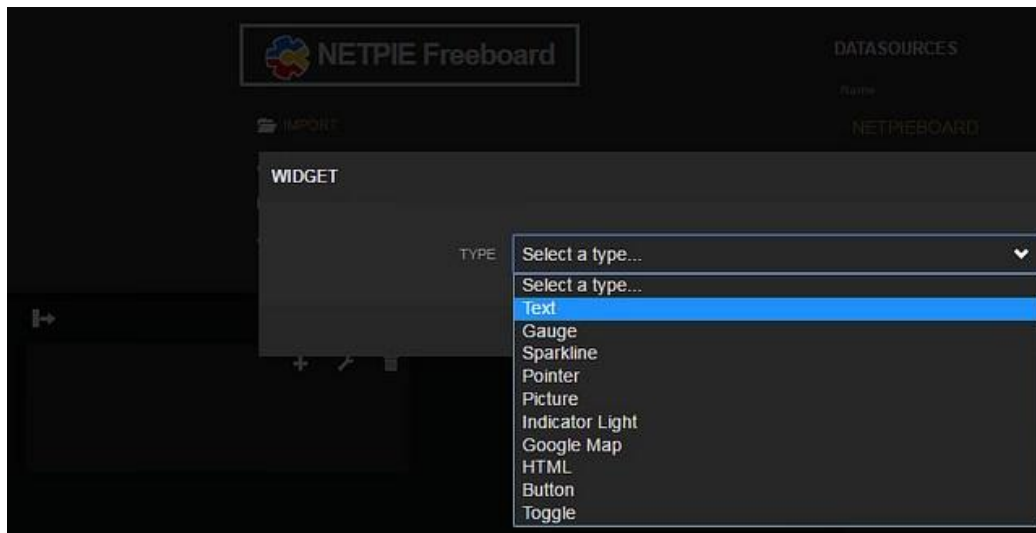
Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID.

TYPE	NETPIE Microgear
NAME	NETPIEBOARD
APP ID	HelloNETPIE <small>NETPIE App ID obtained from https://netpie.io/app</small>
KEY	 <small>Key</small>
SECRET	 <small>Secret</small>
DEVICE ALIAS	NETPIEname <small>A nick name of this freeboard that other device can chat to</small>
MICROGEAR REFERENCE	dashboard <small>Define a reference for a microgear of this datasource. For example if you set this to 'mygear' you can access the microgear object by microgear['mygear']</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>
CMMC JSON MESSAGES?	<input type="checkbox"/> NO <small>If the messages on your topic are in JSON format they will be parsed so the individual fields can be used in freeboard widgets</small>

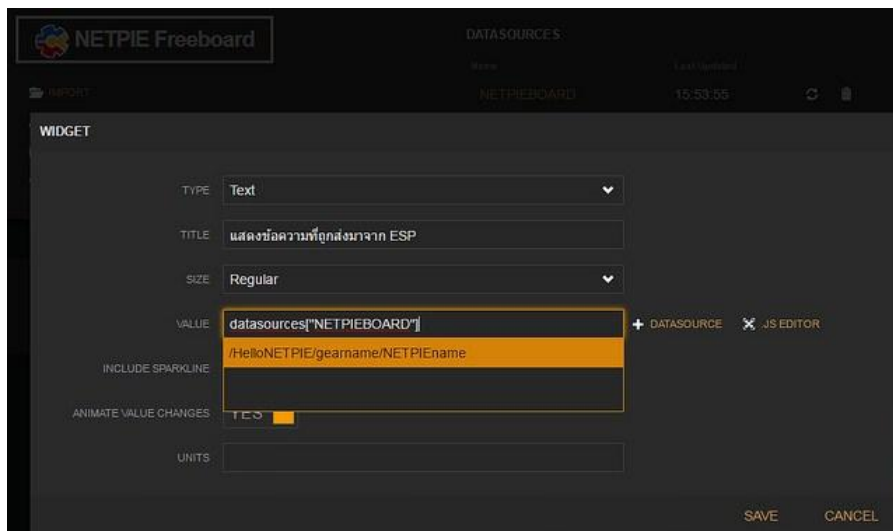
19) จากนั้นทำการสร้างตัวแสดงผล (WIDGET) เพื่อแสดงข้อมูลที่ถูกส่งมาจาก ESP โดยคลิก ADD PANE —> (+) เพื่อสร้าง WIDGET



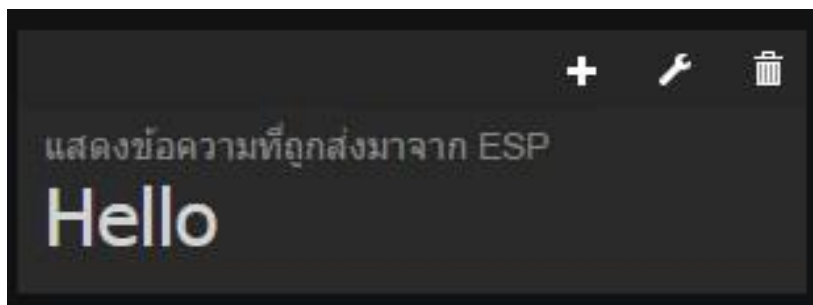
20) TYPE ให้เลือกที่ Text



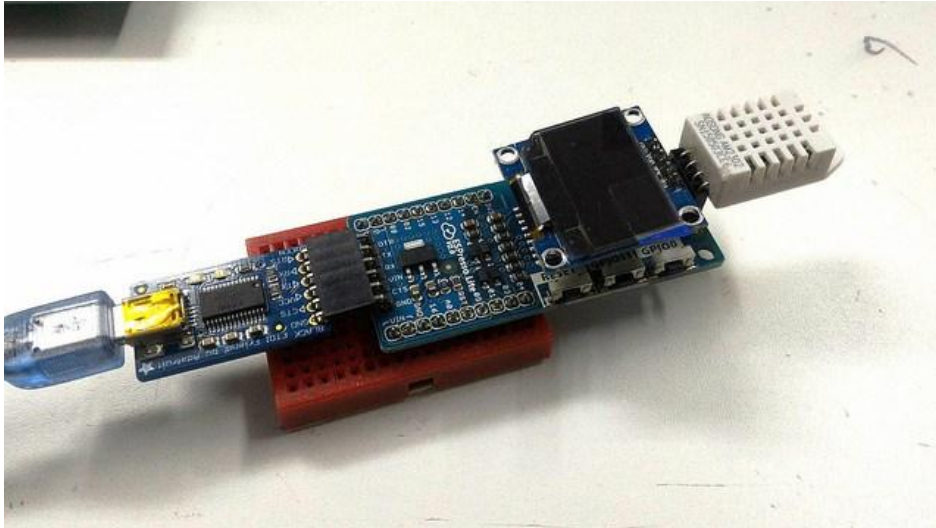
21) คลิก +DATASOURCE แล้วเลือกรับข้อมูลจาก DEVICE ALIAS (ไม่ควรพิมพ์ในช่อง VALUE)



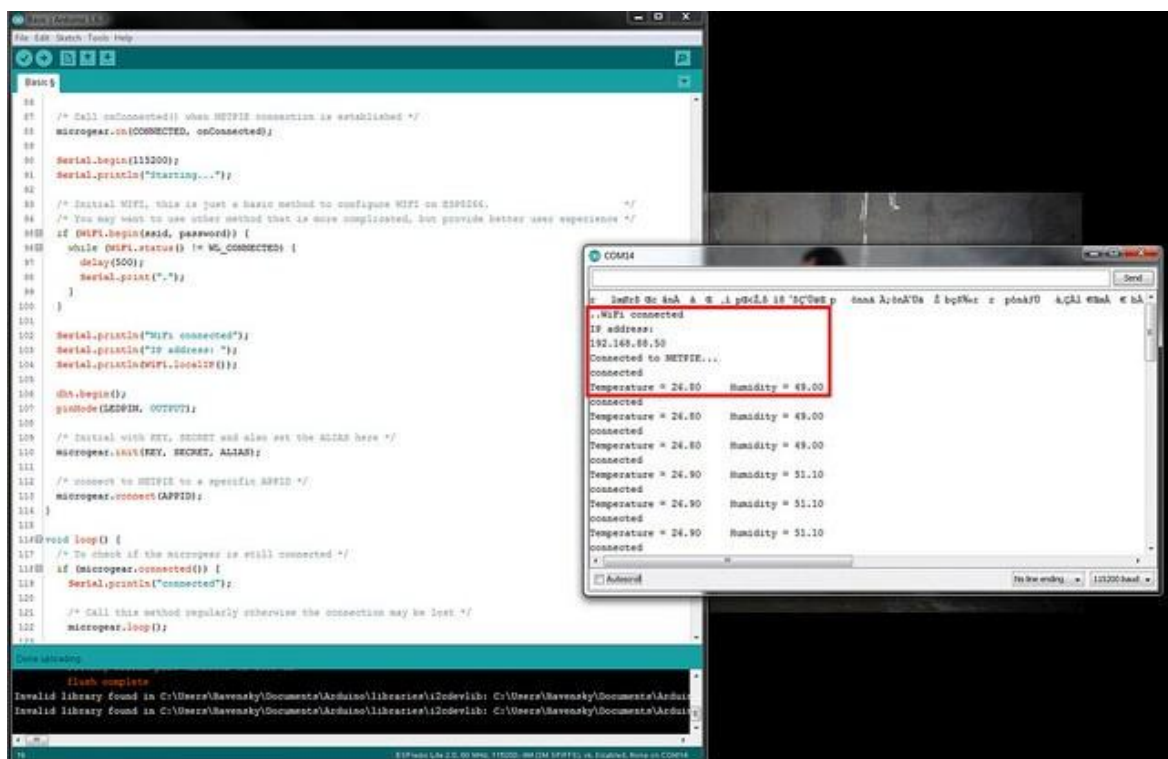
22) เมื่อกด SAVE จะแสดงข้อความที่ถูกส่งมาทางหน้า Widget ที่เราสร้าง



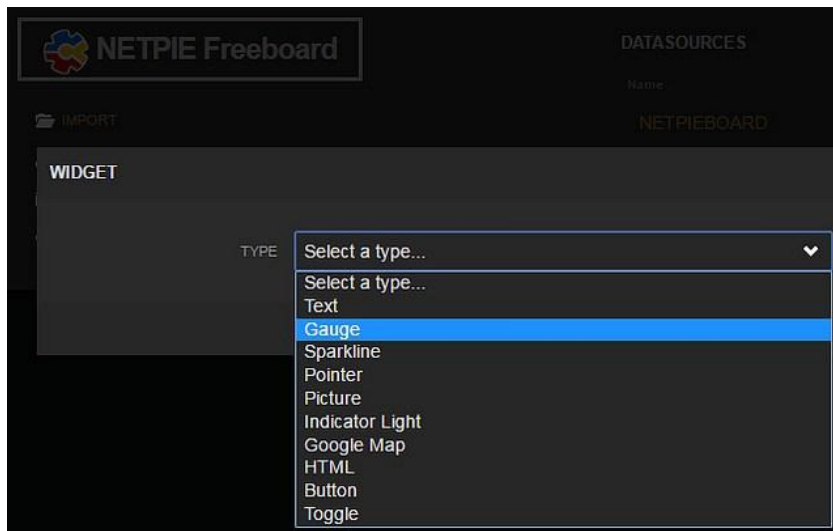
23) ขั้นตอนต่อไปจะทำการส่งค่าที่อ่านได้จากเซ็นเซอร์ DHT22 และควบคุมการติดดับของ LED บนบอร์ด ESPresso Lite กันครับ โดยสามารถดาวน์โหลดโค้ดตัวอย่างได้ [ที่นี่](#)



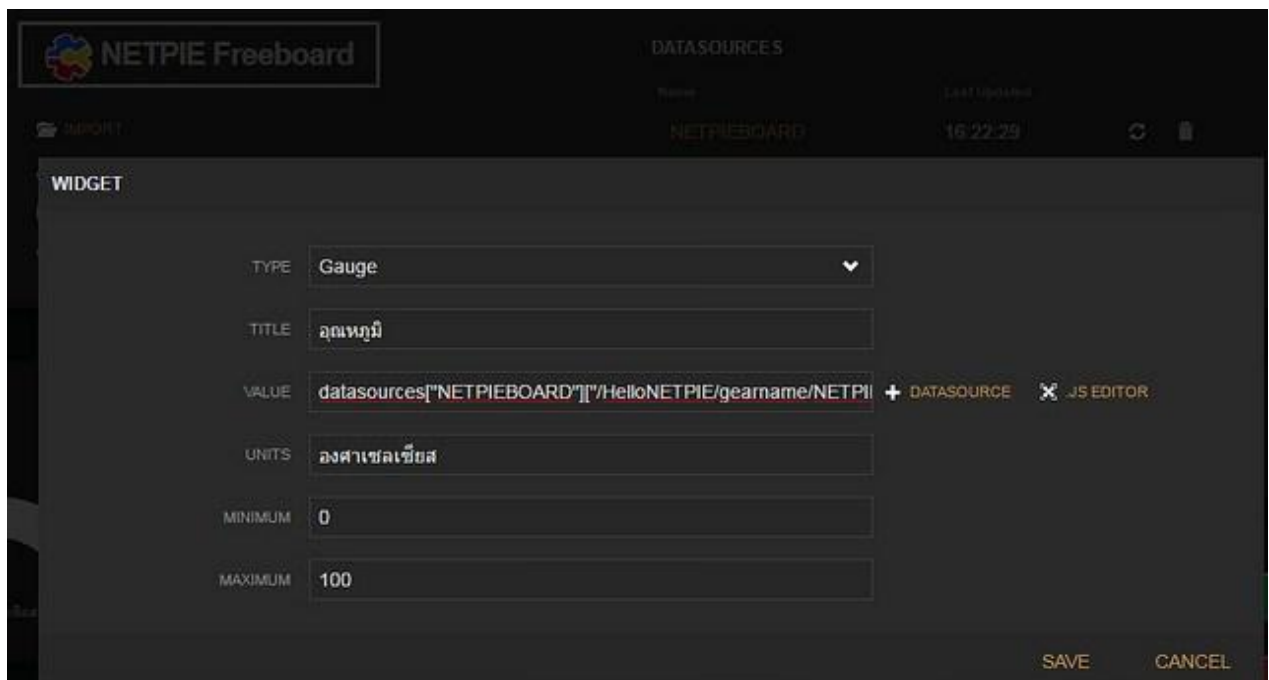
24) ทำการเปิดโค้ดตัวอย่าง อัปโหลดโค้ดไปยังบอร์ด และเปิด Serial port จะแสดงข้อมูลอุณหภูมิ ความชื้นที่ขึ้นไปยัง netpie.io (ในฟังก์ชัน void loop อย่าลืมแก้ ALIAS ใน microgear.chat ด้วยนะครับ)



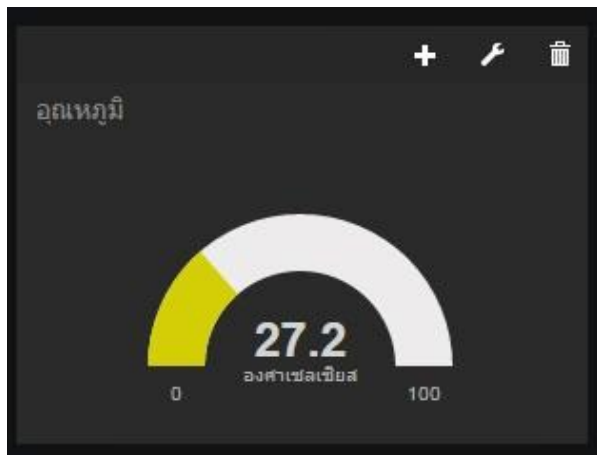
25) กลับมายังหน้า Dashboard ให้ทำการสร้าง Widget สำหรับแสดงค่าอุณหภูมิโดยคลิก ADD PANE —> (+) —> Gauge



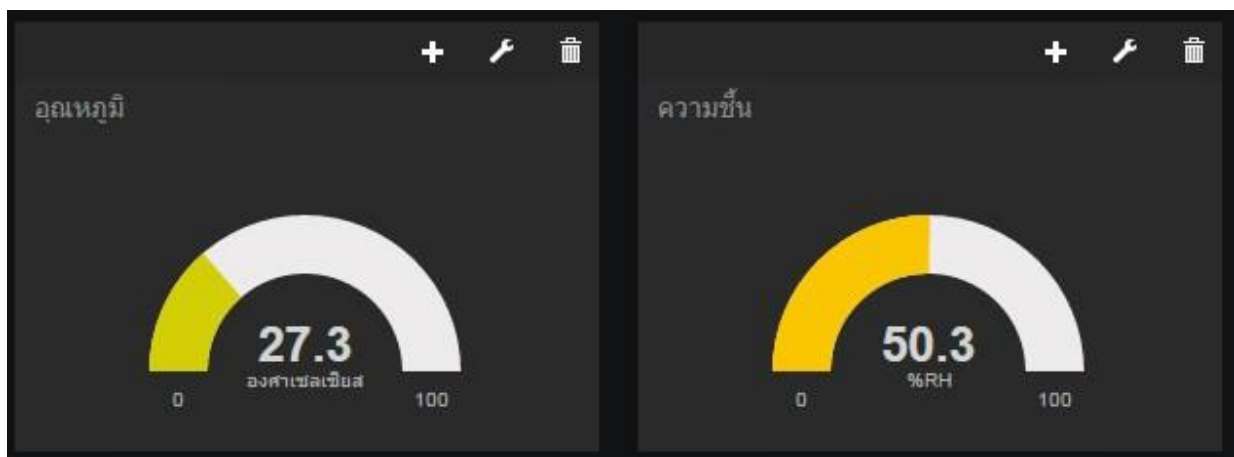
26) ทำการเลือก +DATASOURCE ให้ถูกต้อง โดยเลือก `datasources["NETPIEBOARD"][/HelloNETPIE/gearname/NETPIName/Temperature]` (NETPIE BOARD คือ Name และ NETPIName คือ DEVICE ALIAS ตามข้อ 18)



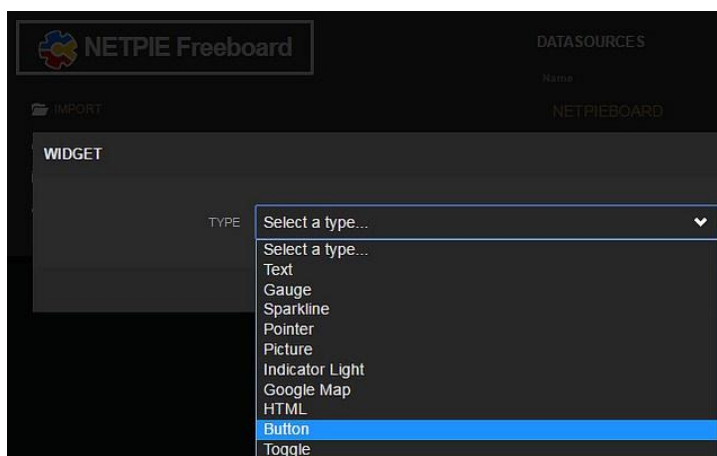
27) กด Save จะแสดงค่าอุณหภูมิที่ถูกส่งมาจาก ESPresso Lite



28) ทดลองเพิ่ม Widget ความชื้น



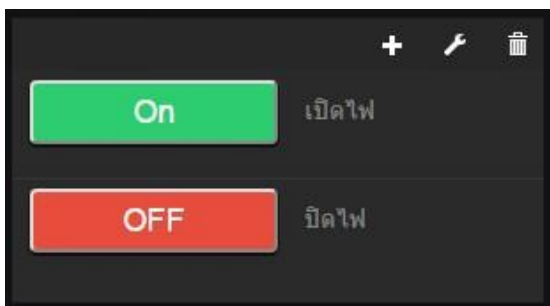
29) ขั้นตอนต่อไป จะเป็นการสร้าง Widget สำหรับสั่งเปิด - ปิด LED บนบอร์ด ESPresso Lite โดยทำการสร้าง Widget Button



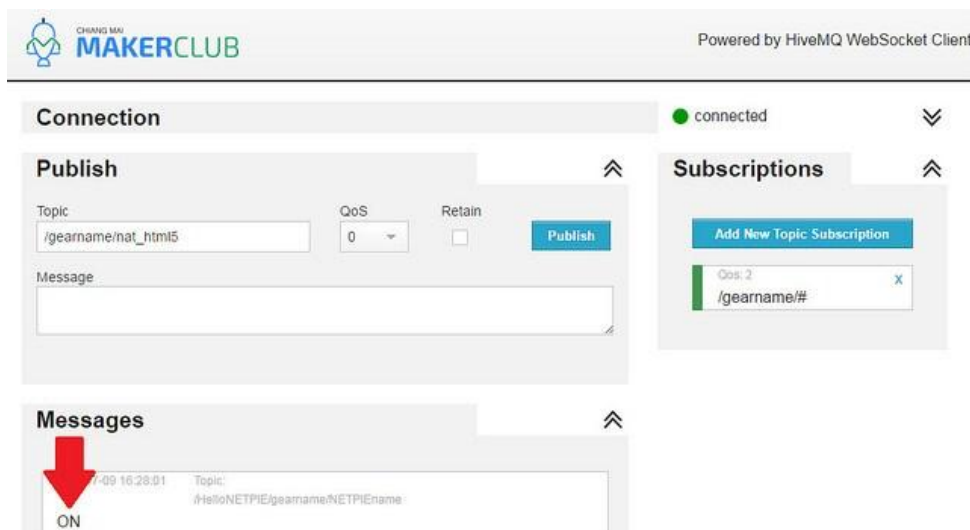
30) ในช่อง ONCLICK ACTION ให้เพิ่ม microgear[“dashboard”].chat(“NETPIName”,“ON”); ซึ่งจะส่งคำว่า ON มาให้ ESPresso Lite รับ (dashboard คือ MICROGEAR REFERENCE และ NETPIName คือ DEVICE ALIAS ตามชื่อที่เราตั้งในการ Add Dashboard)

The screenshot shows the 'WIDGET' configuration screen for a 'Button' widget. The 'TYPE' is set to 'Button'. The 'BUTTON CAPTION' is 'On'. The 'LABEL TEXT' is 'เปิดไฟ'. The 'BUTTON COLOR' is 'Green'. The 'ONCLICK ACTION' is set to the JavaScript code: `microgear["dashboard"].chat("NETPIName","ON");`. There are links for '+ DATASOURCE' and 'JS EDITOR'. Below the code, there is a small explanatory text: '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 microgear reference.' The 'ONCREATED ACTION' is empty. At the bottom right, there are 'SAVE' and 'CANCEL' buttons.

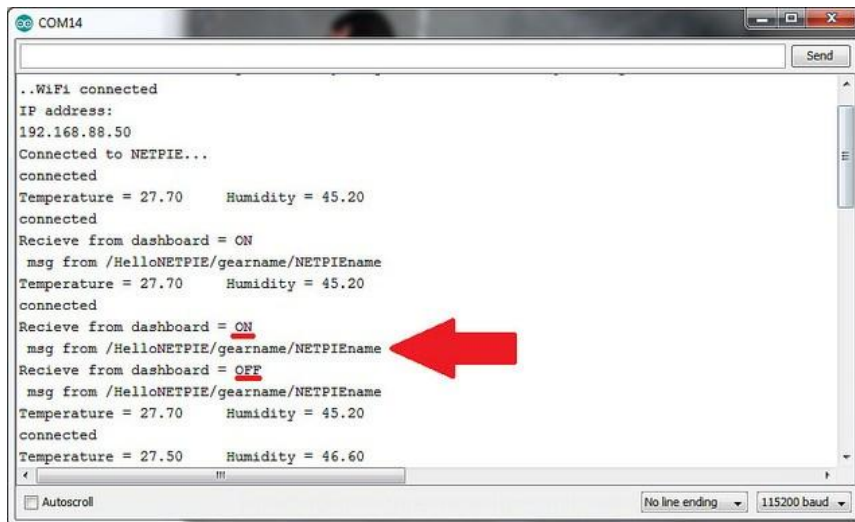
31) สร้างปุ่ม OFF ขึ้นมาอีกปุ่มเพื่อปิด LED โดยแก้ ONCLICK ACTION ให้เป็น microgear[“dashboard”].chat(“NETPIName”,“OFF”);



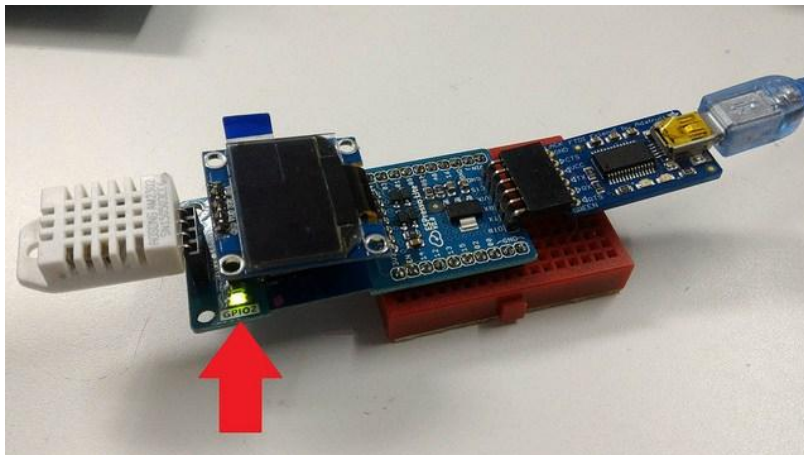
32) เปิดหน้า MQTT Websocket Client เมื่อทำการคลิกที่ปุ่ม ON ใน Dashboard จะเห็นข้อความที่ส่งเข้ามาไปยัง DEVICE ALIAS



33) เปิดหน้า Serial port จาก Arduino IDE จะแสดงข้อความที่รับเข้า



34) ทดลองกดปุ่ม ON – OFF และสังเกต LED บนบอร์ด ESPresso Lite



35) ทดลองเพิ่ม Widget ต่าง ๆ โดยสามารถดาวน์โหลด Dashboard ตัวอย่างได้จาก [ที่นี่](#)

