

เอกสารประกอบการอบรม
"การอบรมเชิงปฏิบัติการเรียนรู้และพัฒนาอินเทอร์เน็ตในทุกสรรพสิ่ง
(Internet of Things) (NECTEC IoT Camp 2016)"

ณ ห้องปฏิบัติการนวัตกรรมเสริมทักษะการเรียนรู้ด้วยตนเอง ชั้น 6
อาคารเฉลิมพระเกียรติ มหาวิทยาลัยเทคโนโลยีราชมงคลล้านนา เชียงใหม่
ถนนห้วยแก้ว อำเภอเมือง จังหวัดเชียงใหม่

ติดตั้ง Arduino IDE กับ ESP8266

การติดตั้ง Arduino IDE กับ ESP8266 คือการติดตั้งโปรแกรมเพื่อใช้ในการเขียนโปรแกรมให้กับบอร์ด ESP8266 หรือบอร์ดอื่น ๆ ผ่านทาง Arduino ใช้ภาษาหลักการของภาษา C++ ครอบด้วยภาษาของ Arduino เพื่อให้การเขียนโปรแกรมมีความง่ายมากยิ่งขึ้น

1.ติดตั้ง Arduino IDE เวอร์ชัน 1.6.5 หรือ ใหม่กว่า โดย Arduino IDE สามารถ Support Windows Mac OS X Linux ทั้ง 32bit และ 64 bit ซึ่งหา Download ตัวติดตั้งได้

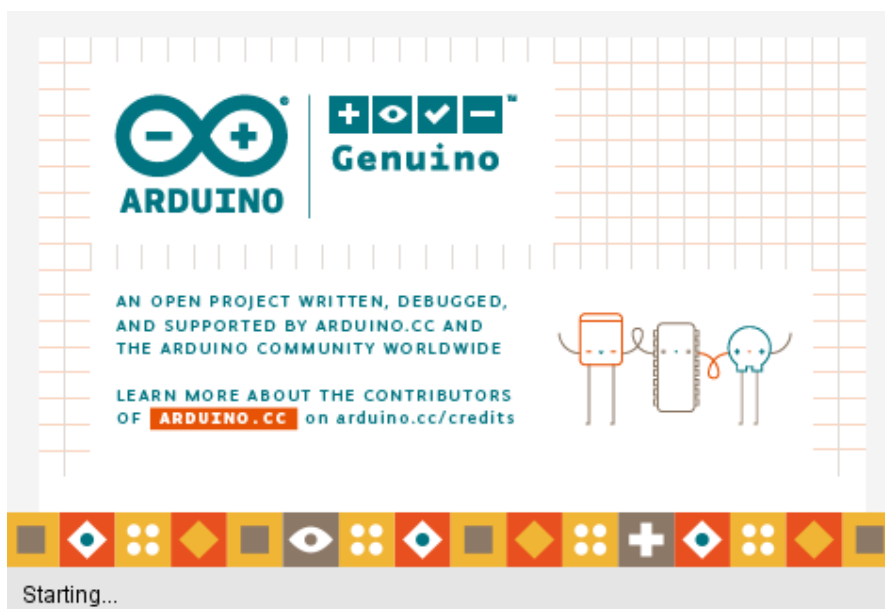
จาก <https://www.arduino.cc/en/main/software>

Download the Arduino Software

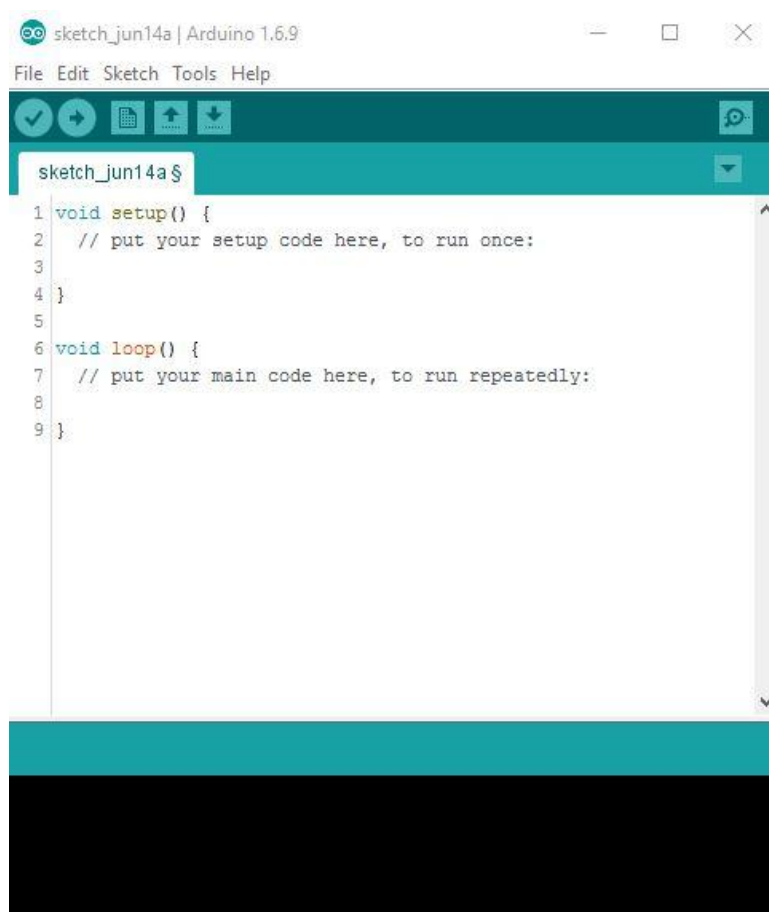


ดาวน์โหลด Arduino IDE เวอร์ชันล่าสุด

สำหรับการติดตั้ง Arduino IDE บน Windows จะมีแบบให้เลือกทั้ง แดกไฟล์ใช้ได้เลย (ZIP file for non admin install)หรือ ติดตั้งเหมือน โปรแกรมทั่วไป (Installer) หากติดตั้งเรียบร้อยแล้ว ให้เปิด Arduino IDE ขึ้นมาจะได้หน้าต่างแบบนี้เลยครับ



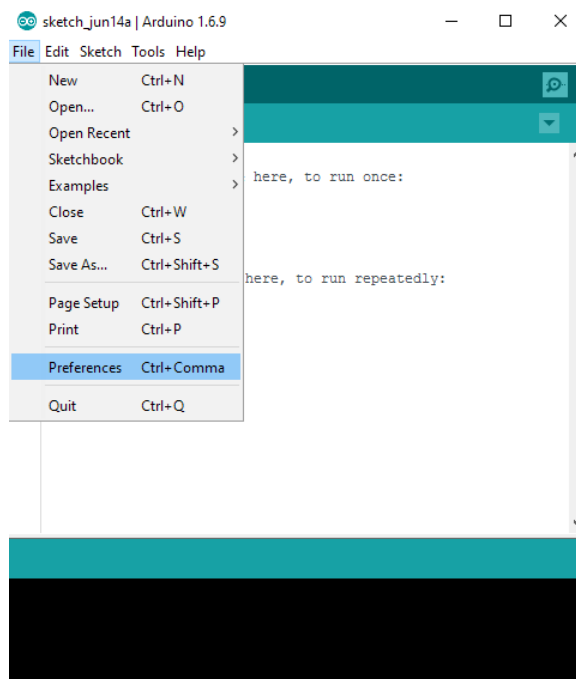
หน้าต่างเริ่มต้นโปรแกรม Arduino



หน้าต่างหลังเปิดใช้งาน Arduino IDE 1.6.9 ครั้งแรก

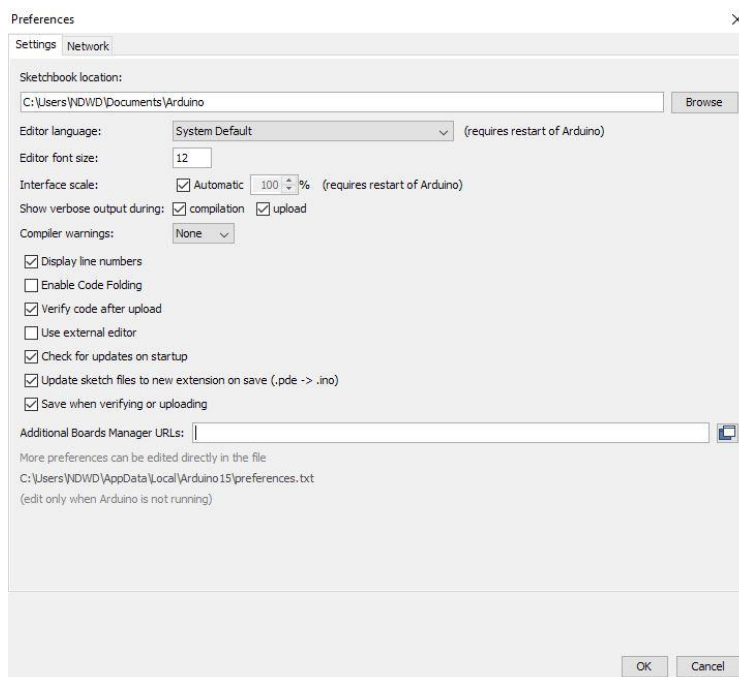
2. ติดตั้ง Broad ESP8266 ลงบน Arduino IDE

1) กดเลือก Menu ไปที่ File >> Preferences



เข้าเมนู File >> Preferences

2) จะขึ้นหน้าต่าง Preferences ให้สังเกตในช่อง Additional Board Manager URLs:



หน้าต่าง Preferences

ในส่วนของ Additional Board Manger บทความนี้ผมจะใช้ Boards Manager ของ Community ESP8266

Available versions

Stable version updated Apr 18, 2016

Boards manager link: http://arduino.esp8266.com/stable/package_esp8266com_index.json

Documentation: <http://esp8266.github.io/Arduino/versions/2.2.0/>

Staging version updated May 11, 2016

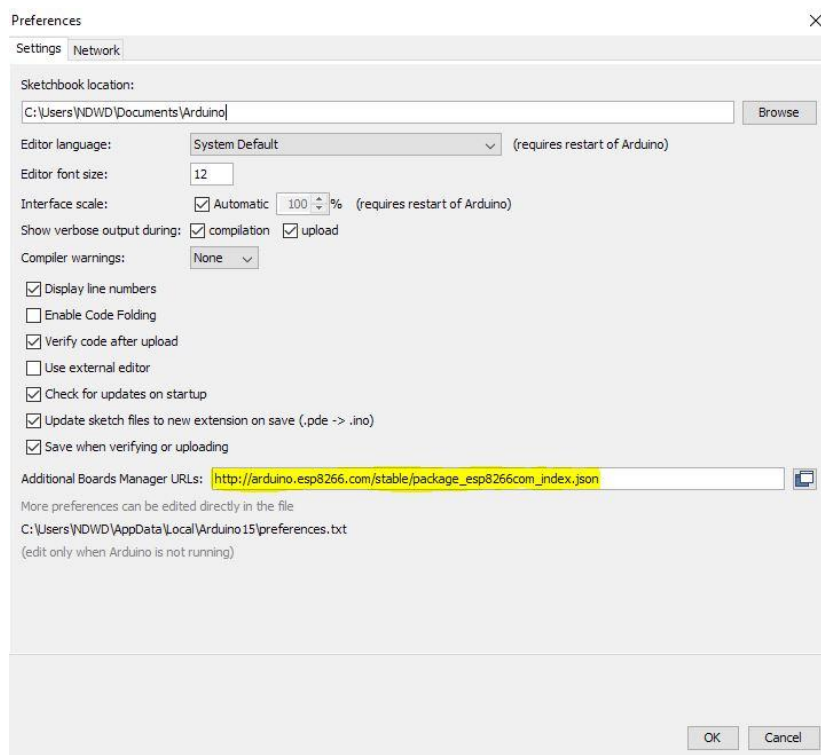
Boards manager link: http://arduino.esp8266.com/staging/package_esp8266com_index.json

Documentation: <http://esp8266.github.io/Arduino/versions/2.2.0-rc1/>

ใช้ Stable Version

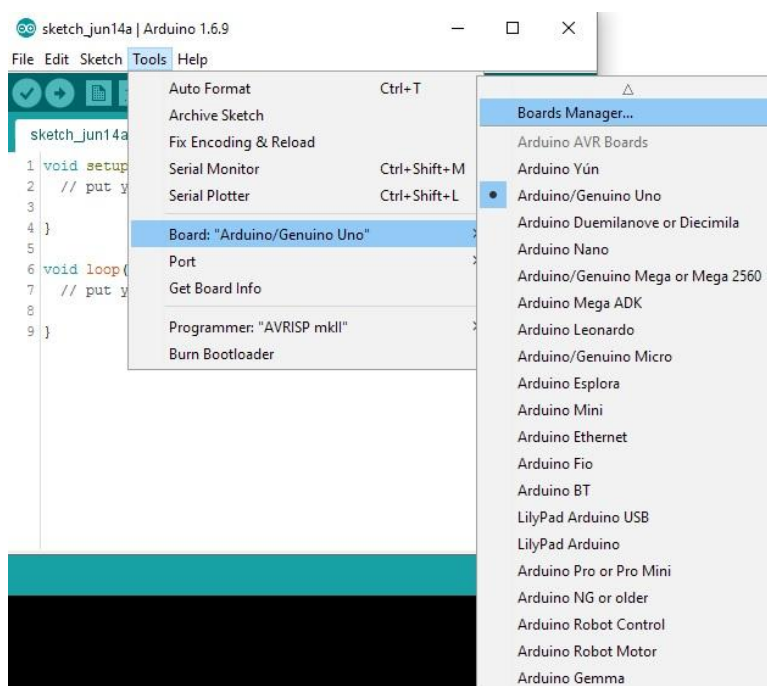
3) ใส่ URL >> ลงใน Addition Board Manager URLs: ดังนี้

http://arduino.esp8266.com/stable/package_esp8266com_index.json



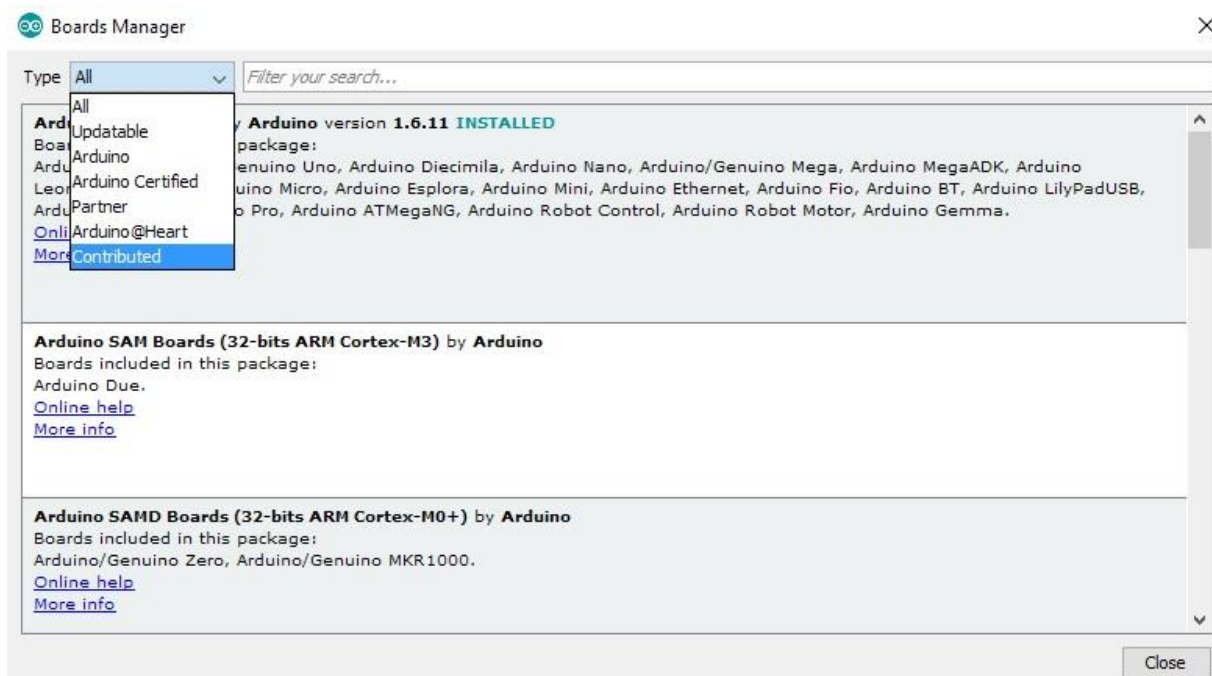
วาง Link ลงใน Boards Manager URLs: จากนั้นกด OK

4) ไปที่ Menu Tools >> Board: "Arduino..." >> Board Manager...



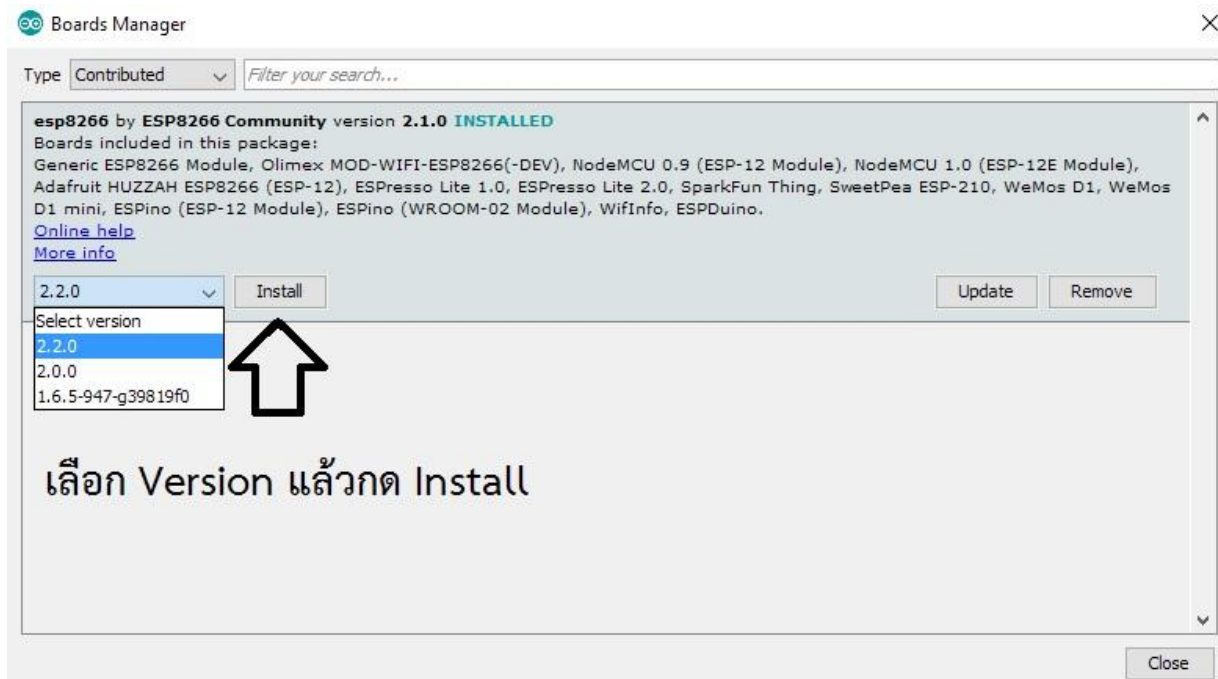
เลือก Boards Manager...

5) จะขึ้นหน้าต่าง Boards Manager เลือก Type เป็น Contributed



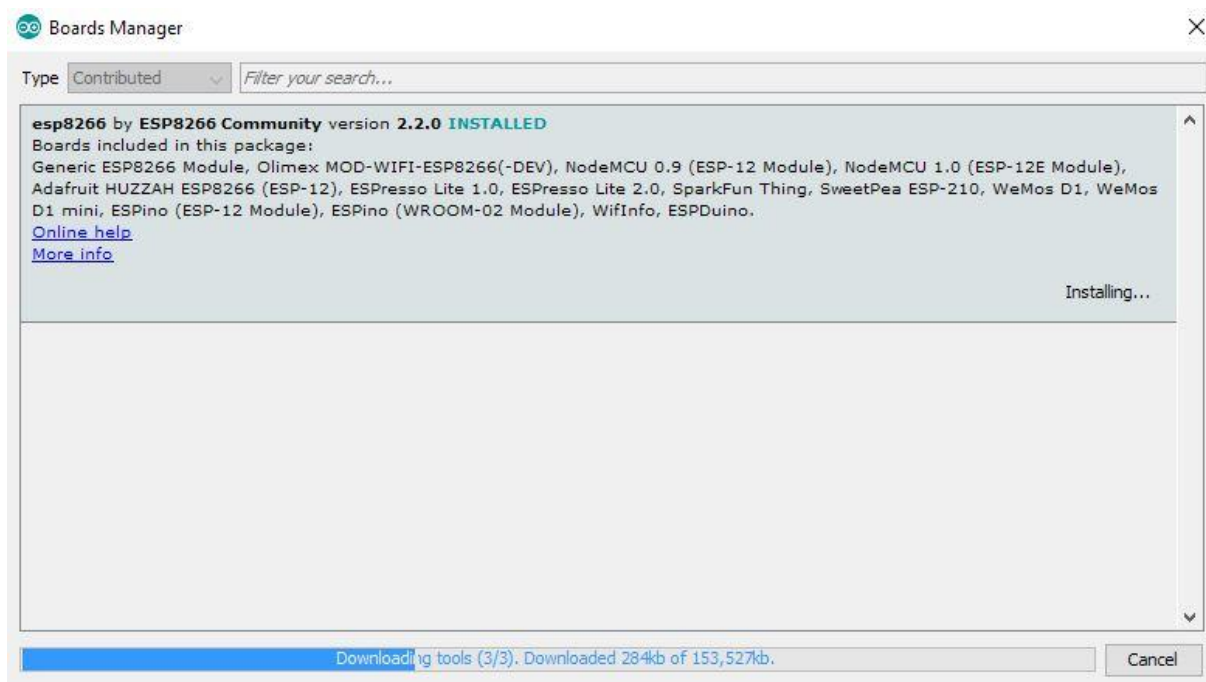
หน้าต่าง Boards Manager

6) จะแสดง Boards ของ ESP8266 เลือกเวอร์ชัน 2.2.0 หรือใหม่กว่าในอนาคต Boards และกด Install



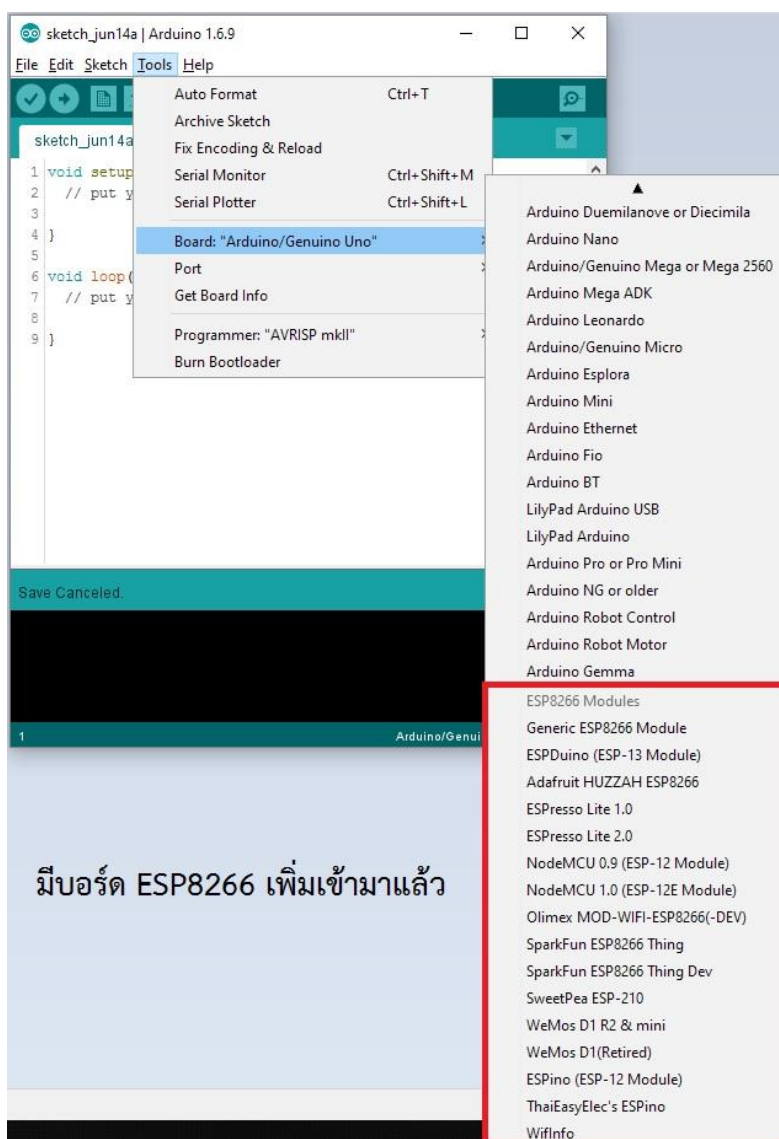
Install Board ESP8266

7) แล้วรอ สัก 2-3 นาที ตัวโปรแกรมจะโหลด Boards ESP8266 ให้ ขนาดไฟล์ประมาณ 150 MB และติดตั้งให้เอง



Download Boards ESP8266

8) เมื่อเราติดตั้งบอร์ด ESP8266 เสร็จเรียบร้อยแล้ว ให้ปิดโปรแกรม Arduino IDE ก่อน แล้วจึงเปิดขึ้นมาใหม่ให้ลองเลือกไปที่ Menu Tools >> Board: “.....” ซึ่งจะพบว่า มี Menu สำหรับเลือกใช้งาน ESP8266 กับ Arduino IDE ขึ้นมาให้เลือกใช้งานแล้วครับ



Board ESP8266 ถูกเพิ่มเข้ามา

ซึ่งหมวดของ ESP8266 Modules นั้นจะมีบอร์ดให้เลือกใช้งานอยู่ด้วยกันหลายบอร์ด ที่มีจำหน่ายทั่ว ๆ ไป เช่น

- Generic ESP8266 Module >> บอร์ด ESP8266 ที่ใช้งานทั่วไป หรือ บอร์ดที่สร้างขึ้นเอง
- ESPresso Lite 1.0 และ 2.0 >> บอร์ด ESP8266 ของ Espert และ Chiangmai Maker Club
- Olimex MOD-WIFI-ESP8266(-DEV) >> บอร์ด ESP8266 ที่บริษัท Olimex เป็นผู้สร้าง
- NodeMCU (ESP8266 ESP-12 Module) >> บอร์ด ESP8266 ที่เป็นบอร์ด NodeMCU
- WeMos D1 >> บอร์ด ESP8266 ที่เป็นบอร์ด wemos

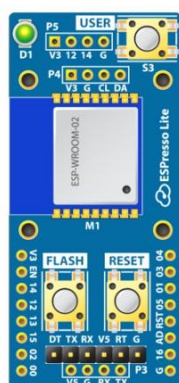
ถ้าหากเรามีบอร์ด ESP8266 ที่รองรับ โมดูล ทั้ง 3 บอร์ด ก็สามารถใช้งานได้เลย บอร์ดรุ่นต่าง ๆ ที่ใช้งานโดยทั่วไป ก็มีดังนี้

1. Generic ESP8266 Module

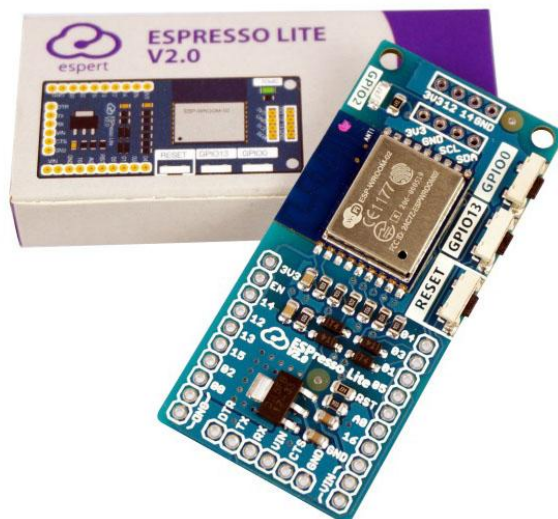
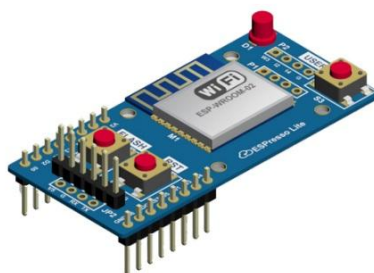


2. ESPresso Lite 1.0 และ 2.0

ESPRESSO Lite



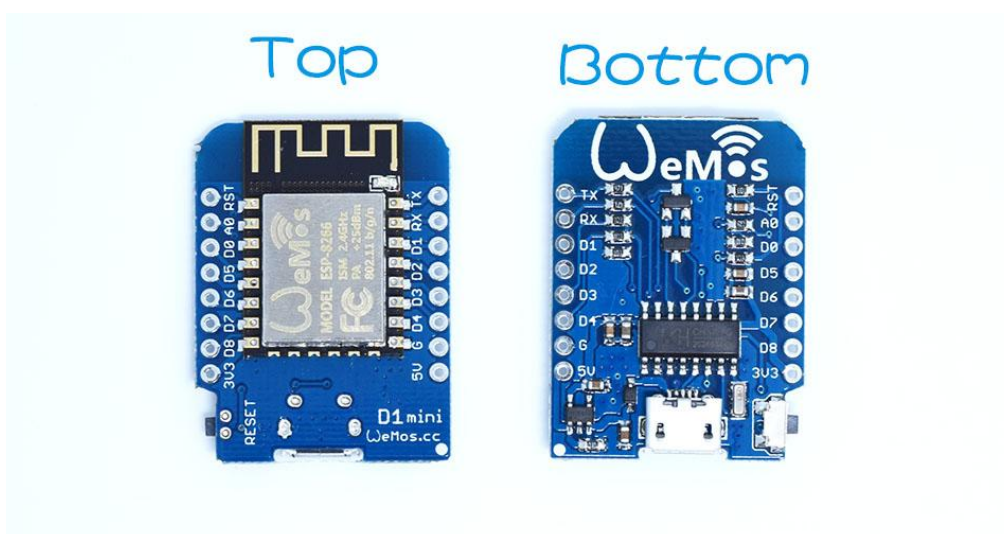
Latest Arduino-compatible,
WiFi-enabled (ESP8266)
development board



3. NodeMCU



4. WeMos



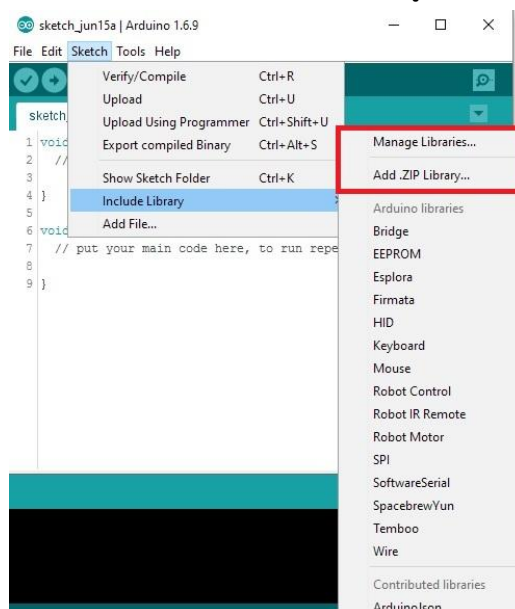
ESP8266-ESP01

ขั้นตอนการติดตั้งโปรแกรมตัวอย่าง Library

การติดตั้งโปรแกรมตัวอย่าง คือ การลงโปรแกรมเสริม หรือ Code สำหรับใช้งานกับ Shield โมดูล หรือ เซนเซอร์ต่าง ๆ ที่ผู้พัฒนาได้มีการพัฒนาให้บอร์ดสามารถใช้งานได้กับอุปกรณ์ต่าง ๆ ที่มีออกมาจำหน่ายอย่างต่อเนื่องซึ่งจำเป็นกับผู้เริ่มต้น เนื่องจากทำให้ผู้ใช้งานประหยัดเวลา มีความสะดวกและ ง่ายมากยิ่งขึ้นในการพัฒนา โดยมีการเพิ่ม Library อยู่ 2 แบบหลักคือ

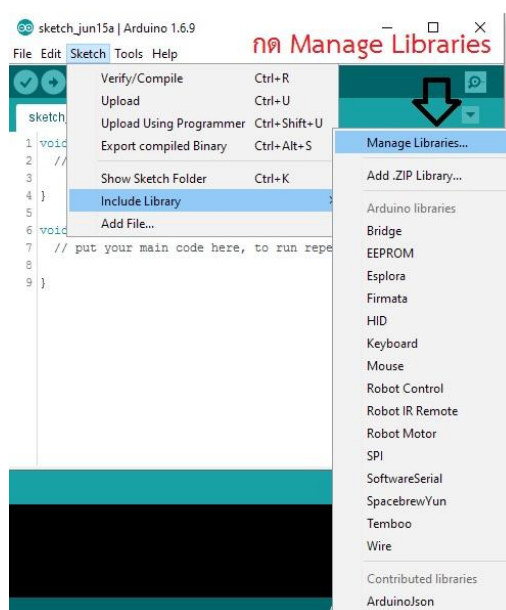
- 1) การเพิ่ม Library โดยวิธี Manage Libraries...
- 2) การเพิ่ม Library โดยวิธี Add.ZIP Library...

โดยการเลือกใช้งาน เมนู Sketch >> Include Library >> ... ดังรูป



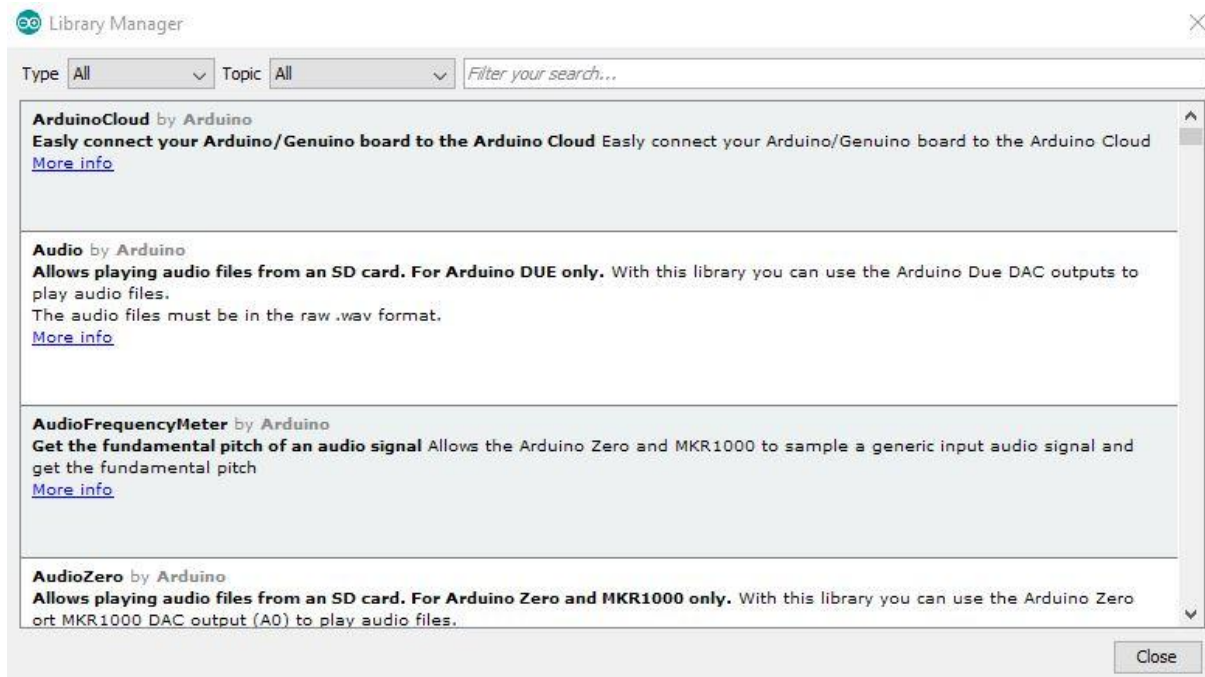
การเพิ่ม Library โดยวิธี Manage Libraries...

เป็นการเพิ่ม Library โดยตรงจากเซิร์ฟเวอร์ของทาง Arduino ซึ่ง Manage Libraries จากผู้พัฒนา ก่อนจะถูกอัปเดตเซิร์ฟเวอร์ของ Arduino จะมีคนคอยตรวจสอบก่อนเสมอ จึงเป็นสากลมากกว่าการลงแบบ Manual และใช้งานได้จริงทดลองลงโปรแกรมตัวอย่างจาก Library ของ ESPert แบบ Manage Libraries... จะมีลักษณะคล้ายกับการลง Broad ESP8266 ข้างต้นซึ่ง "จำเป็นต้องเชื่อมต่อคอมพิวเตอร์กับ Internet ก่อนการเพิ่ม Library โดยวิธีนี้เสมอ" โดยการเลือกใช้งานเมนู Sketch >> Include Library >> Manage Libraries



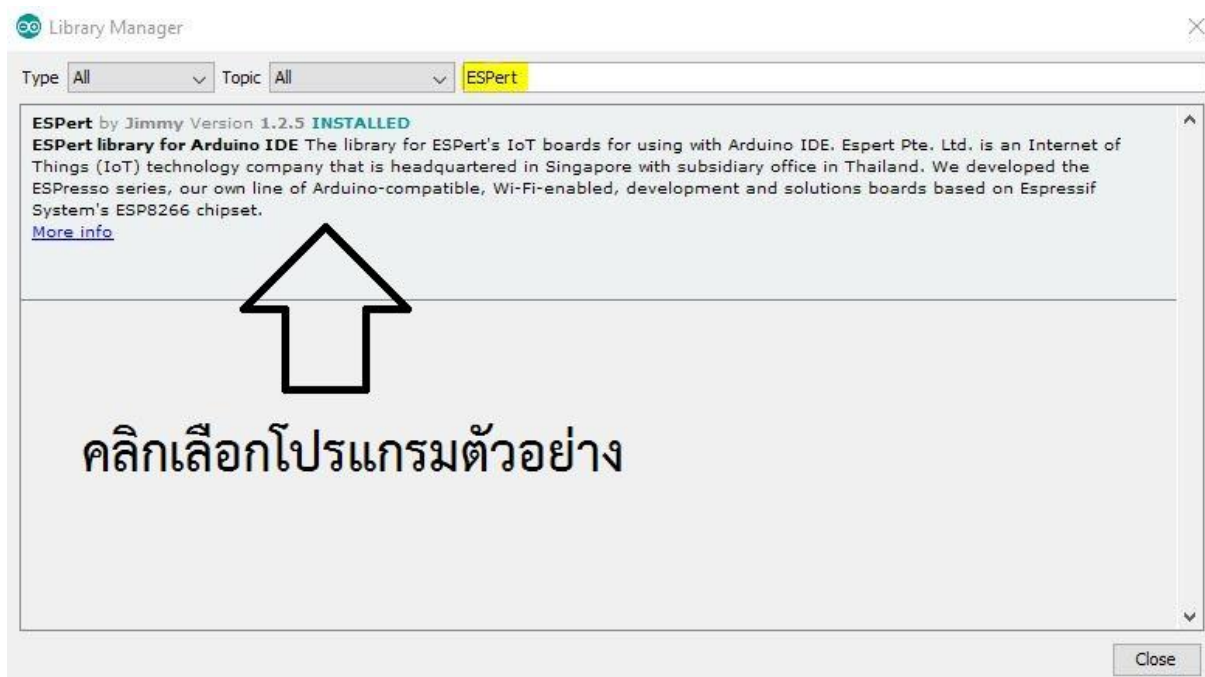
Sketch >> Include Library >>

โปรแกรมจะแสดงหน้าต่างใหม่ขึ้นมาดังรูป



เลือกเมนู Manage Library

ในช่องค้นหาใส่ Library ที่ต้องการเพิ่ม ในที่นี้จะค้นหา Library ของ ESPert ตามรูปที่เน้นคำและคลิกเลือกโปรแกรมที่ค้นหา



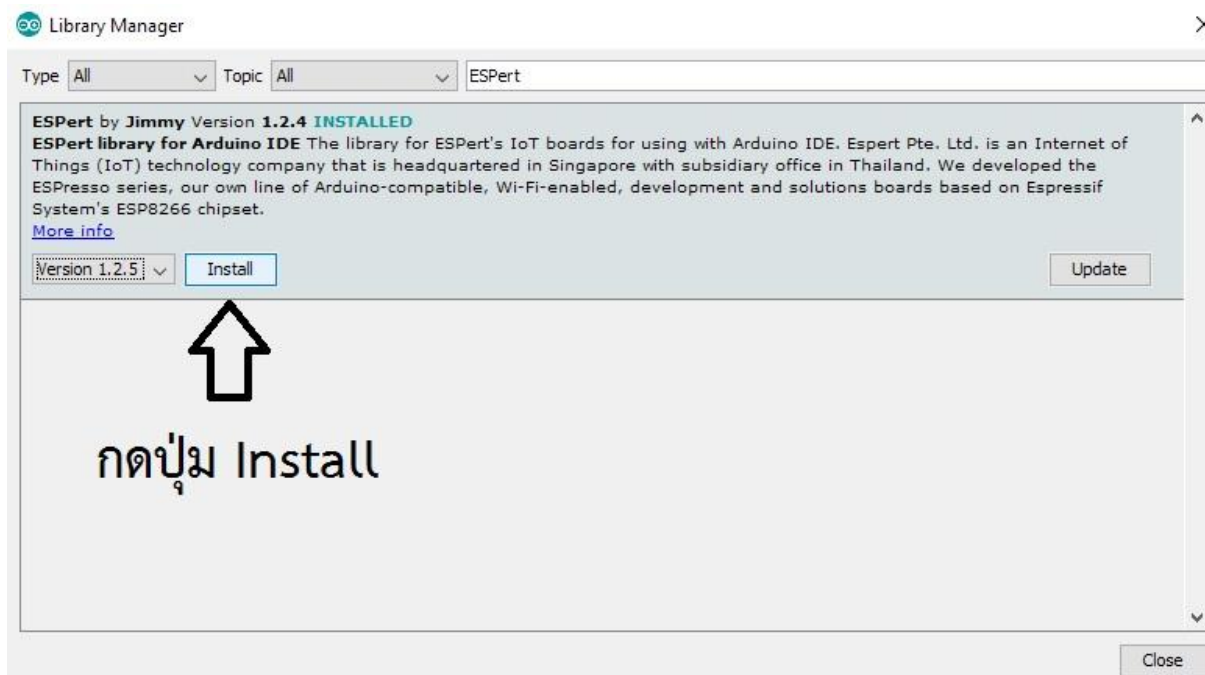
หน้าต่าง Library Manager

เลือก Version ล่าสุด หรือ ในขณะนี้อัปเดตล่าสุดที่ Version 1.2.5



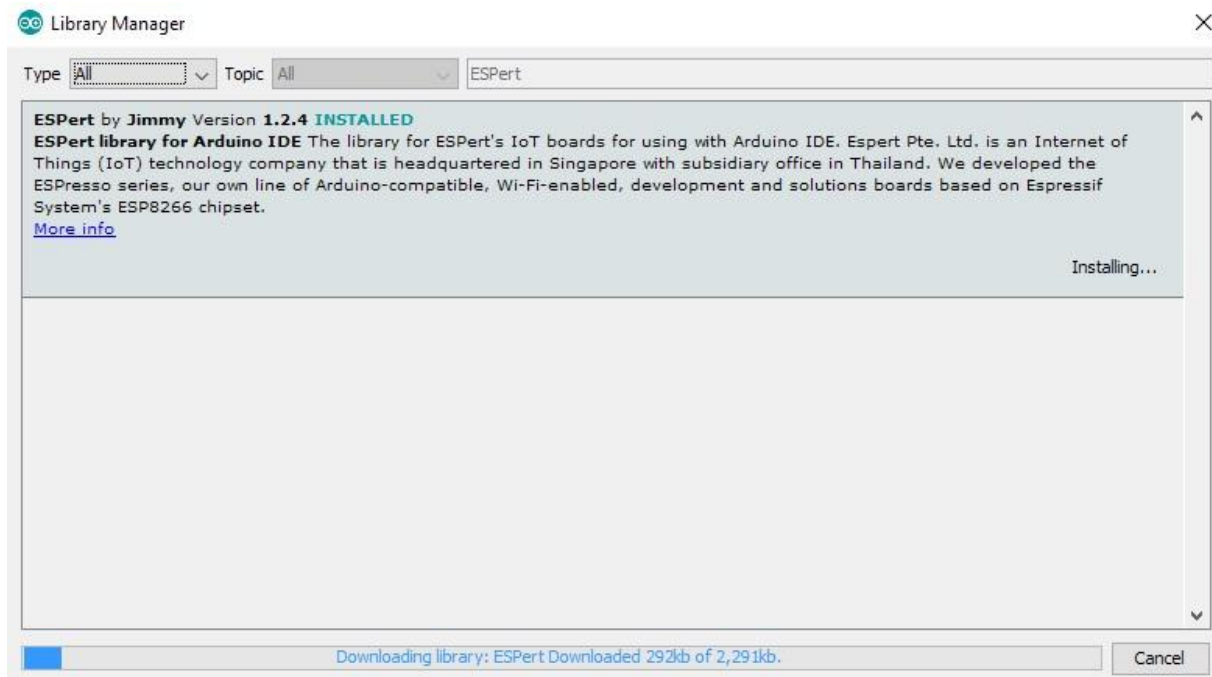
ค้นหาโปรแกรมตัวอย่าง ESPert

กดปุ่ม Install เพื่อเริ่มติดตั้ง Library ลงบน Arduino IDE



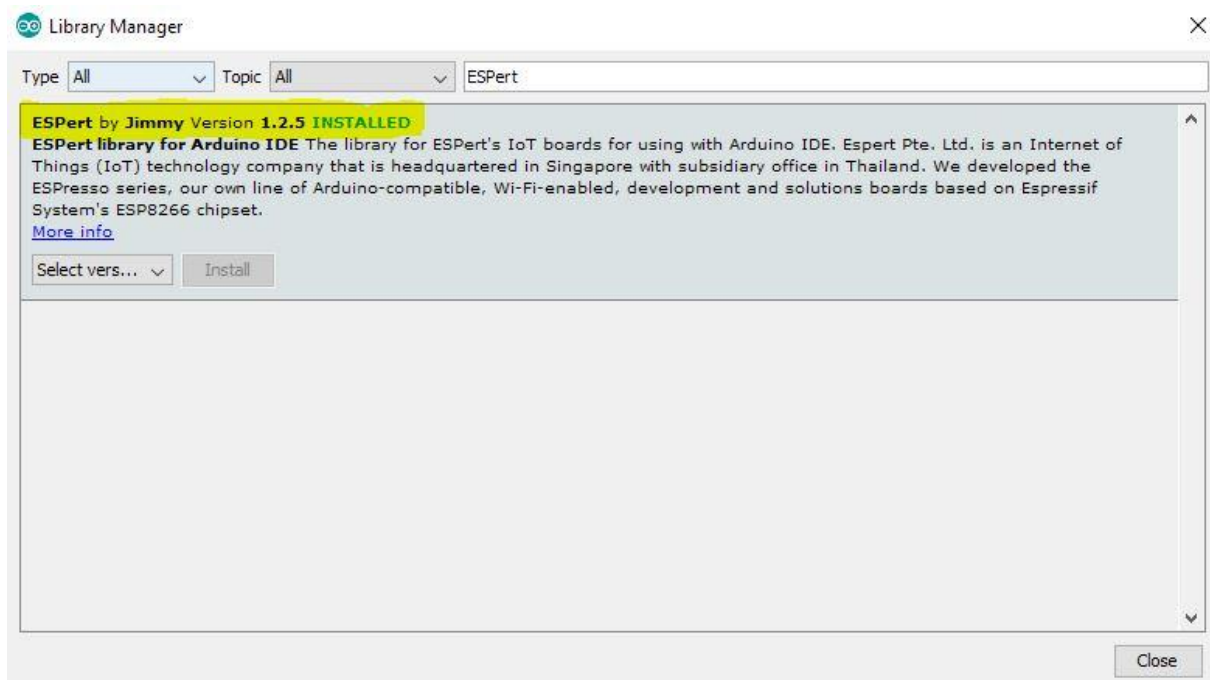
เลือกเวอร์ชันโปรแกรมตัวอย่างล่าสุด

โปรแกรมจะทำการติดตั้ง Library อัตโนมัติจาก Sever ผ่าน Internet



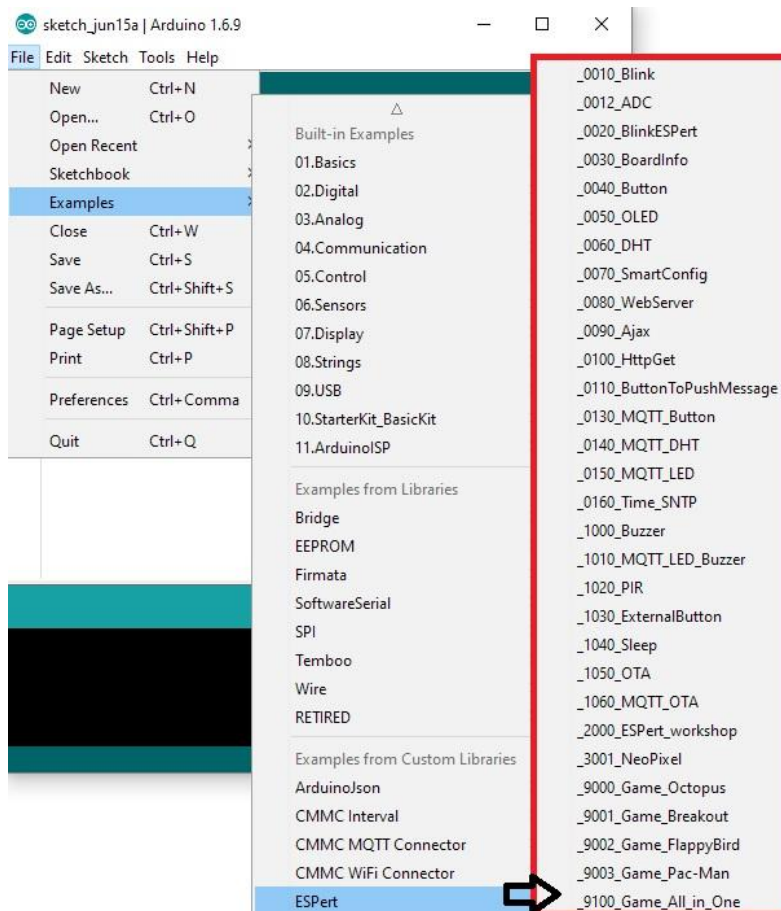
กด Install โปรแกรมตัวอย่าง

โปรแกรมจะแสดงข้อความ INSTALLED ตรงไฮไลต์ตามรูปแสดงว่าเพิ่มโปรแกรมตัวอย่างเรียบร้อยแล้ว



ดาวน์โหลดโปรแกรมตัวอย่างอัตโนมัติ

การเรียกใช้งานโปรแกรมตัวอย่างโดยเข้าไปที่ File >> Examples >> ESPert >>



หน้าต่างแสดงการทำงานสมบูรณ์

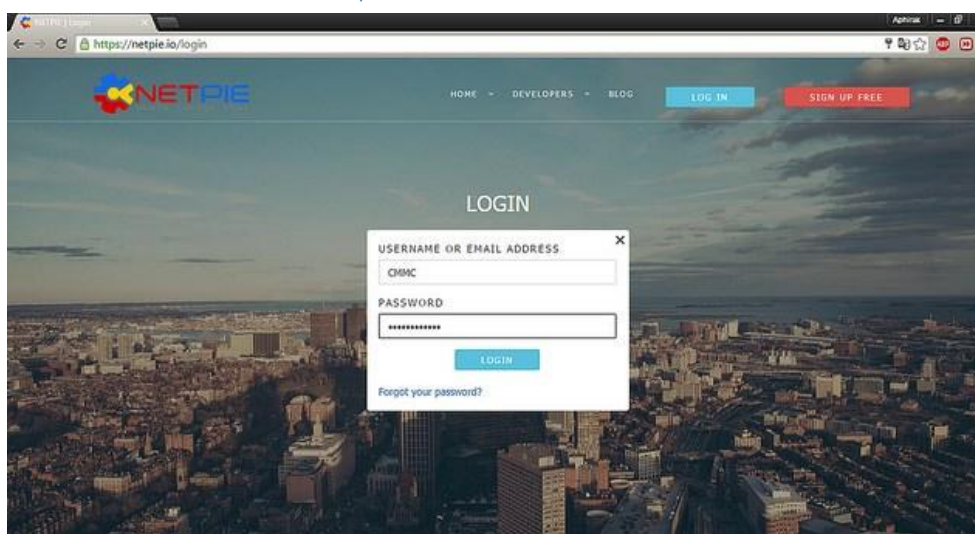
*หมายเหตุ Library ที่จำเป็นกับ ESPresso Lite V2.0

Library	by	current version
Adafruit NeoPixel	Adafruit	1.0.5
ArduinoJson	Benoit Blanchon	5.5.0
DHT sensor library	Adafruit	1.2.3
ESP8266 Oled Driver for SSD1306	Daniel Eichborn	3.0.0
DHT sensor library	Adafruit	1.2.3
HttpClient	Adrian McEwen	2.2.0
PubSubClient	Nick O'Leary	2.6.0

NETPIE.io คืออะไร

“NETPIE แพลตฟอร์ม IoT เพื่อนักพัฒนาและอุตสาหกรรมไทย” กระทรวงวิทยาศาสตร์และเทคโนโลยีโดย 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

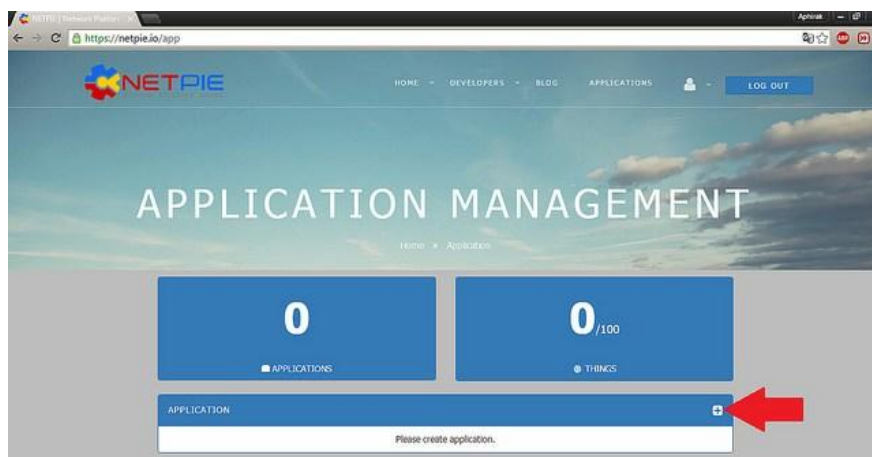
1) ลงชื่อเข้าใช้งาน netpie.io



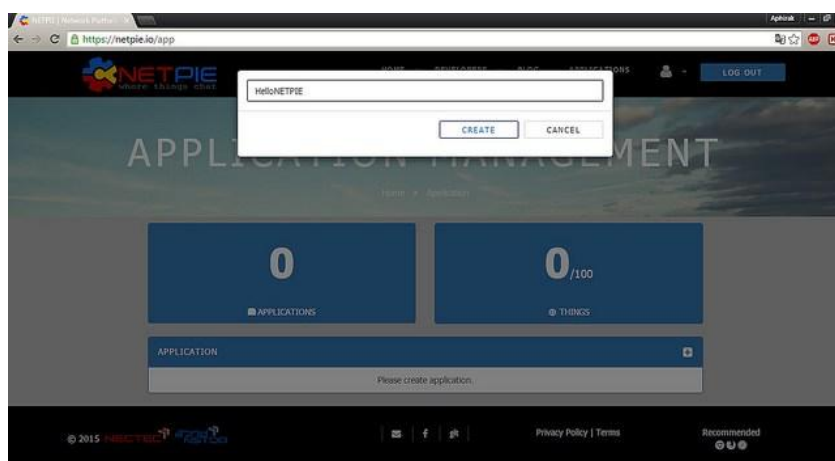
2) คลิกที่ APPLICATIONS เพื่อเข้าใช้งาน



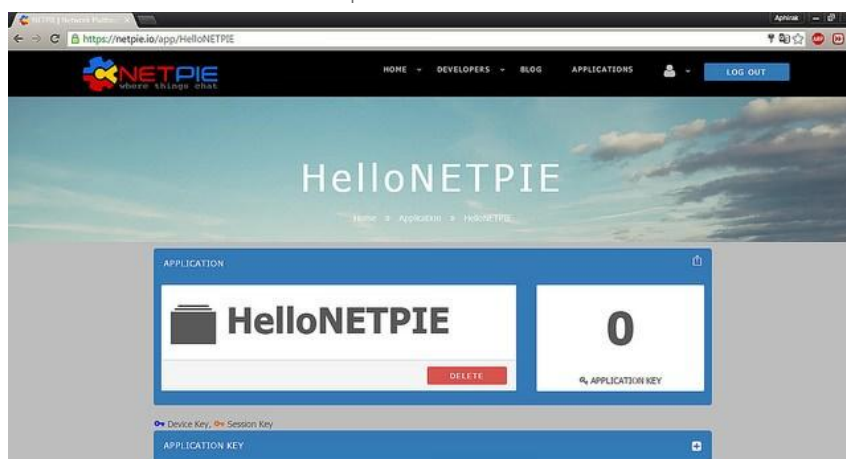
3) คลิกที่เครื่องหมาย + เพื่อสร้าง APPLICATIONS



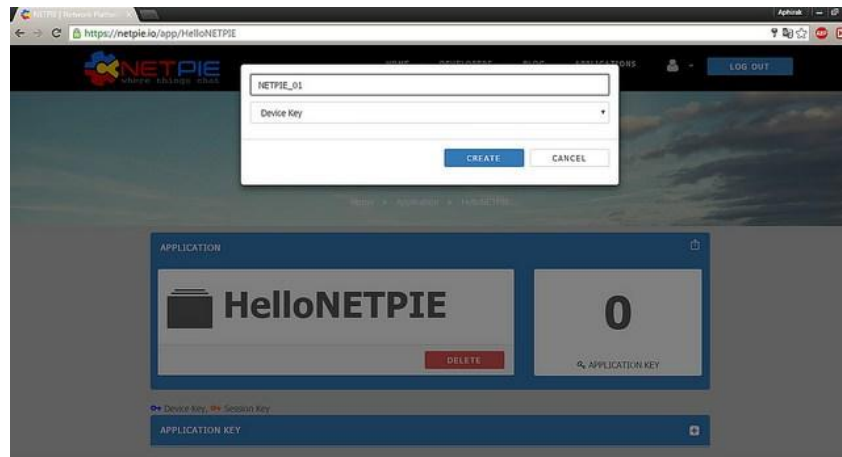
4) สร้างชื่อ APPLICATIONS (ชื่อที่สร้างขึ้นมานี้ก็คือ APPID)



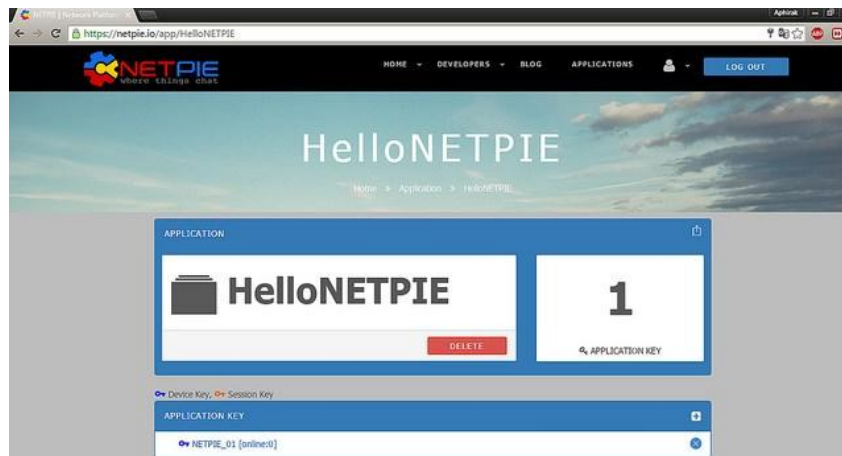
5) แสดงชื่อ APPLICATIONS หรือ APPID ซึ่งตอนนี้ยังไม่มี APPLICATION KEY ให้เราทำการสร้าง โดยการคลิกที่เครื่องหมาย + (มุมล่างขวา)



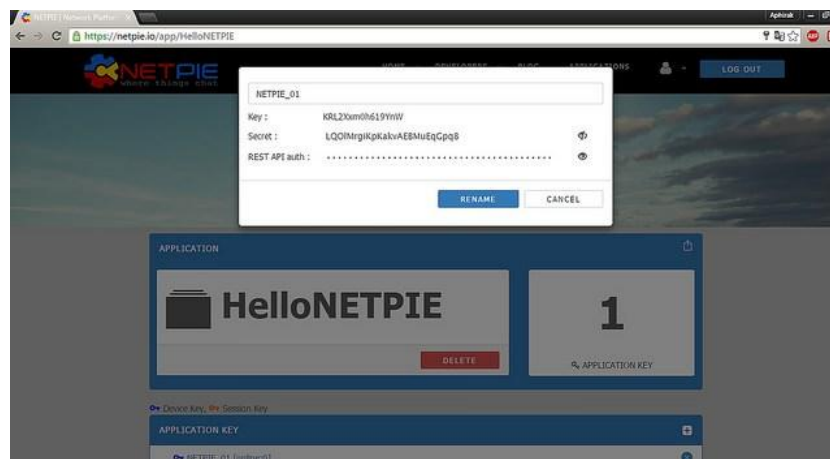
6) ทำการสร้าง Device Key



7) เมื่อสร้างเสร็จจะเห็น Device Key ตามชื่อที่เราสร้างสีฟ้า ๆ ด้านล่าง



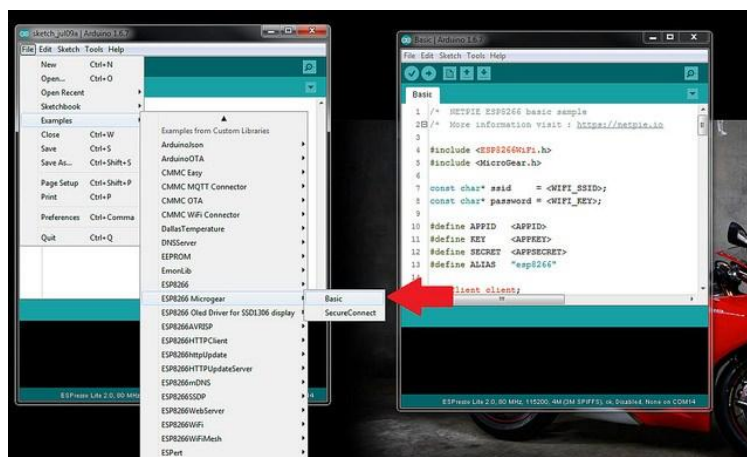
8) คลิกที่ Device Key จะแสดง KEY และ SECRET เพื่อใช้ในการเชื่อมต่อกับ netpie.io



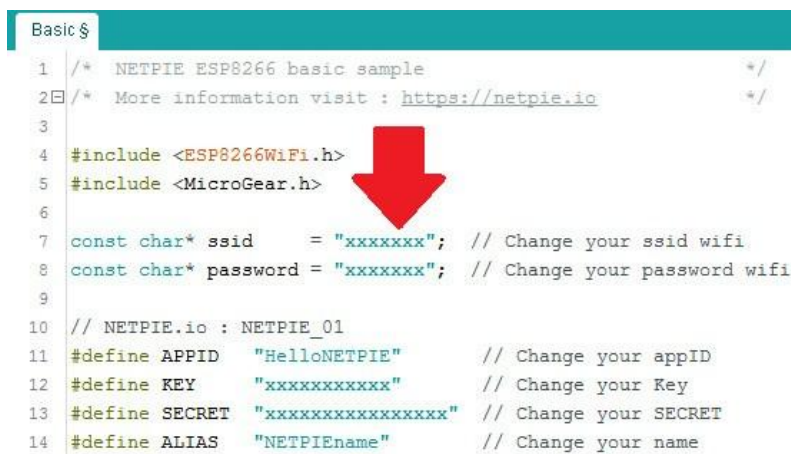
9) ทำการสร้าง Session Key เพื่อใช้ในการเชื่อมต่อกับ Dashboard (Device Key สีฟ้า, Session Key สีส้ม)



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



12) อัปโหลดโค้ดลงบอร์ด ESPresso Lite แล้วเปิด Serial port จะแสดงการเชื่อมต่อ และข้อความที่ส่ง

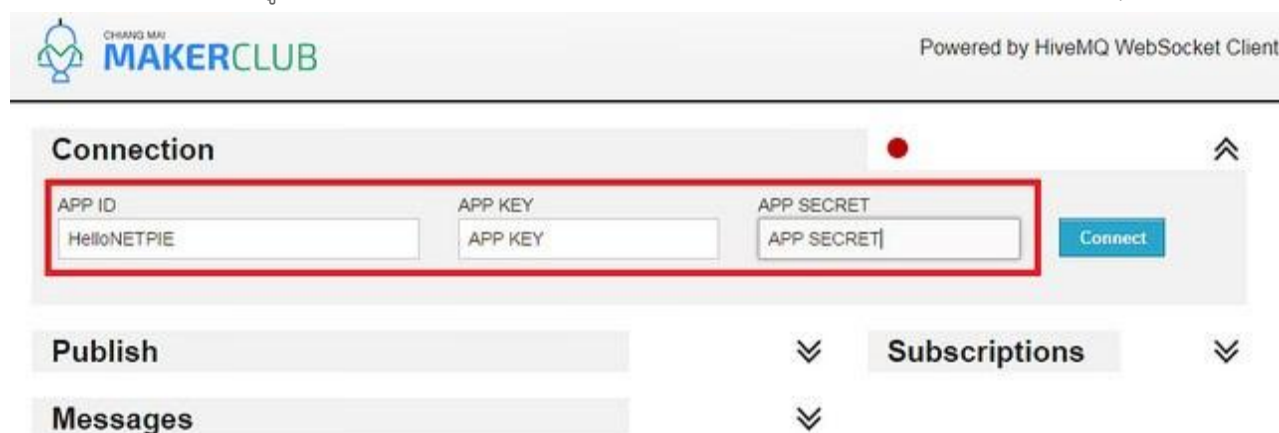
```
99 void loop() {
90   /* To check if the microgear is still connected */
91   if (microgear.connected()) {
92     Serial.println("connected");
93
94     /* Call this method regularly otherwise the connection may be lost */
95     microgear.loop();
96
97     if (timer >= 1000) {
98       Serial.println("Publish...");
99
100      /* Chat with the microgear named ALIAS which is myself */
101      microgear.chat(ALIAS, "Hello");
102      timer = 0;
103    }
104    else timer += 100;
105  }
106  else {
107    Serial.println("connection lost, reconnect...");
108    if (timer >= 5000) {
109      microgear.connect(APPID);
110      timer = 0;
111    }
112    else timer += 100;
113  }
114  delay(100);
115 }
```



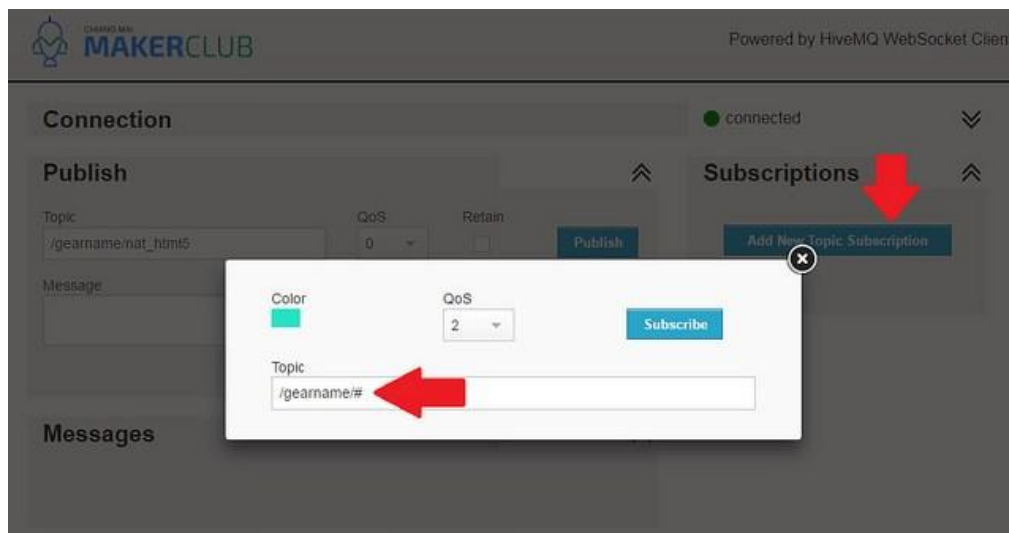
13) กลับไปยัง netpie.io หากมีการเชื่อมต่อจะแสดงชื่อ ALIAS ที่ถูกส่งข้อมูลขึ้นมา



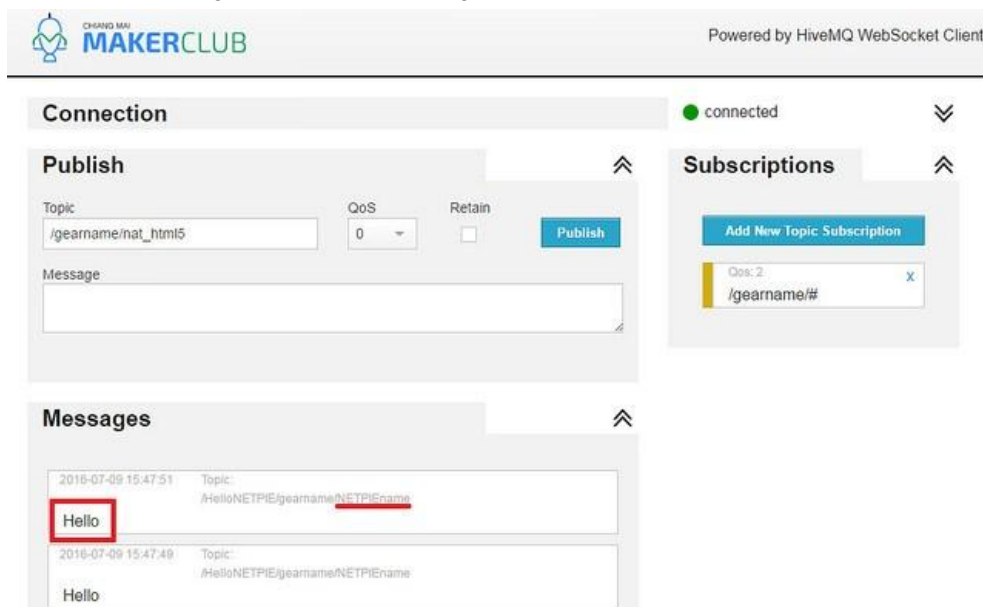
14) ทดลองใช้ **MQTT Websocket Client** เพื่อแสดงข้อมูลที่ถูกส่งขึ้นมา โดยกรอก APP ID, APP KEY, APP SECRET ให้ถูกต้องแล้วกด Connect (APP KEY, APP SECRET ให้ใช้ของ Session Key)



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



16) เมื่อมีการส่งข้อมูล จะแสดงข้อความที่ถูกส่งว่ามาจากใคร



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

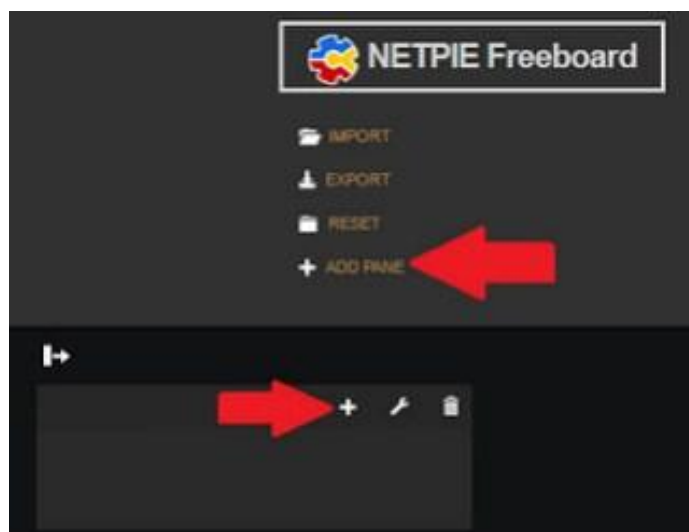


18) คลิก ADD จะแสดงหน้าจอตั้งรูปแบบด้านล่าง กรอกข้อมูลให้ครบถ้วนโดย KEY และ SECRET จะได้จาก Session Key ของเว็บ netpie.io ส่วน DEVICE ALIAS จะต้องใช้ชื่อเดียวกับที่ส่งเข้ามายัง Device Key ที่ต้องการเชื่อมต่อ และ MICROGEAR REFERENCE จะใช้สำหรับอ้างอิงข้อมูลใน dashboard

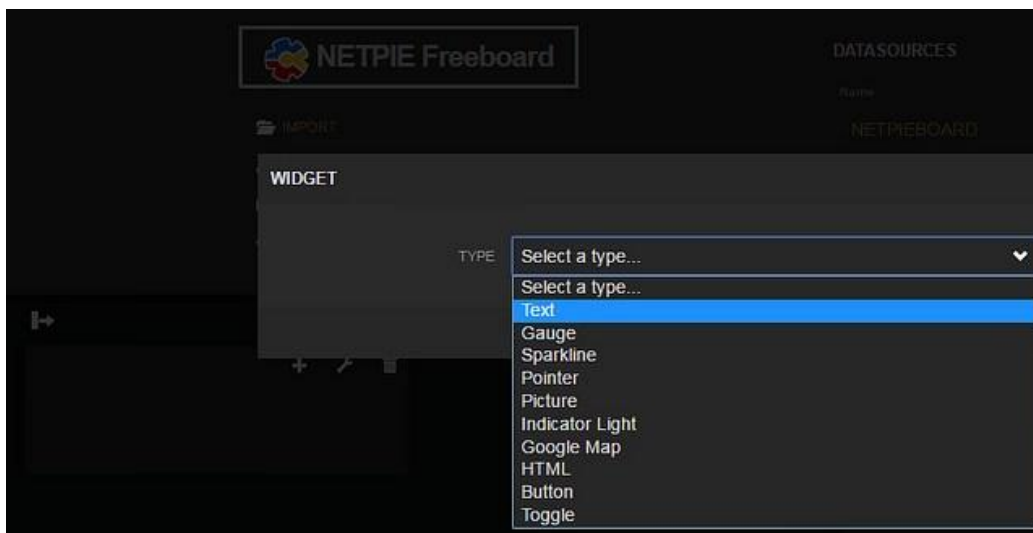
Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID.

TYPE	NETPIE Microgear
NAME	NETPIEBOARD
APP ID	HelloNETPIE <small>NETPIE App ID obtained from https://netpie.io/app</small>
KEY	<input type="text"/> <small>Key</small>
SECRET	<input type="text"/> <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>

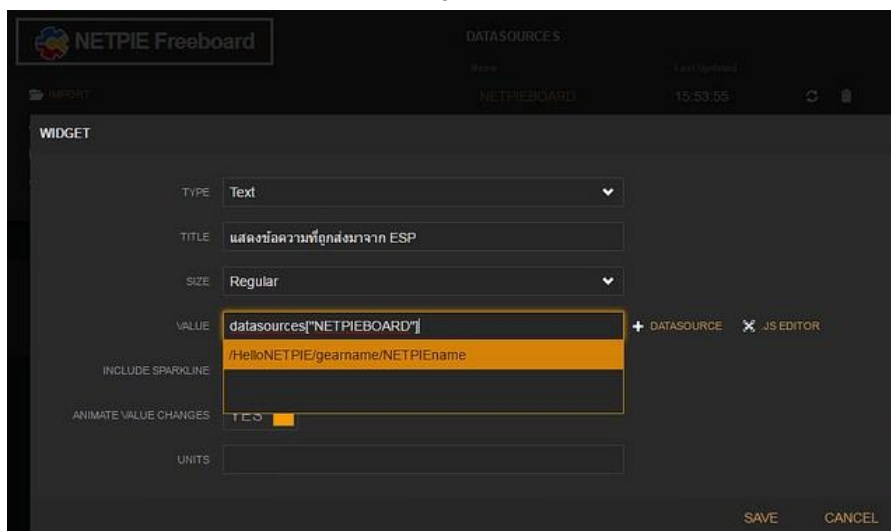
19) จากนั้นทำการสร้างตัวแสดงผล (WIDGET) เพื่อแสดงข้อมูลที่ถูกส่งมาจาก ESP โดยคลิก ADD PANE —> (+) เพื่อสร้าง WIDGET



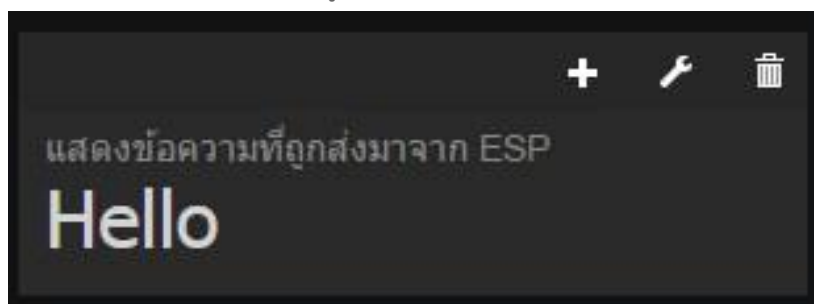
20) TYPE ให้เลือกที่ Text



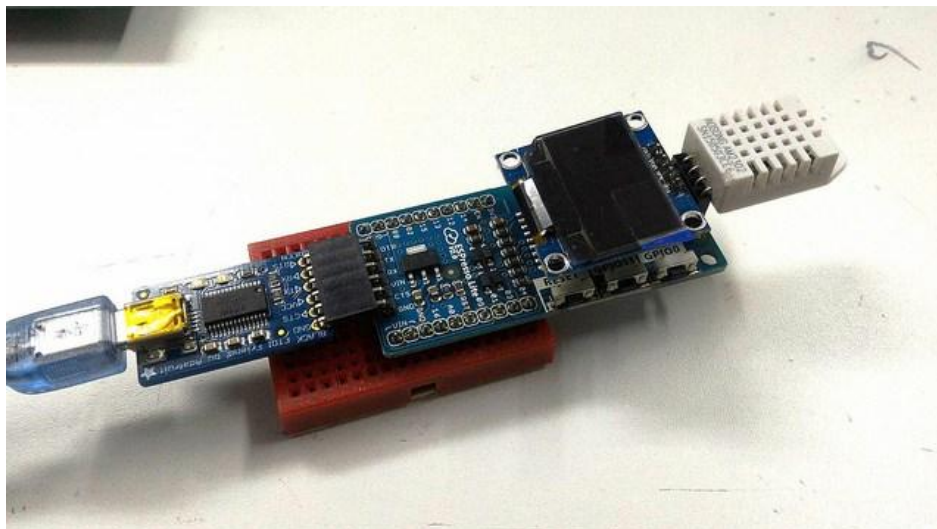
21) คลิก +DATASOURCE แล้วเลือกรับข้อมูลจาก DEVICE ALIAS (ไม่ควรพิมพ์ในช่อง VALUE)



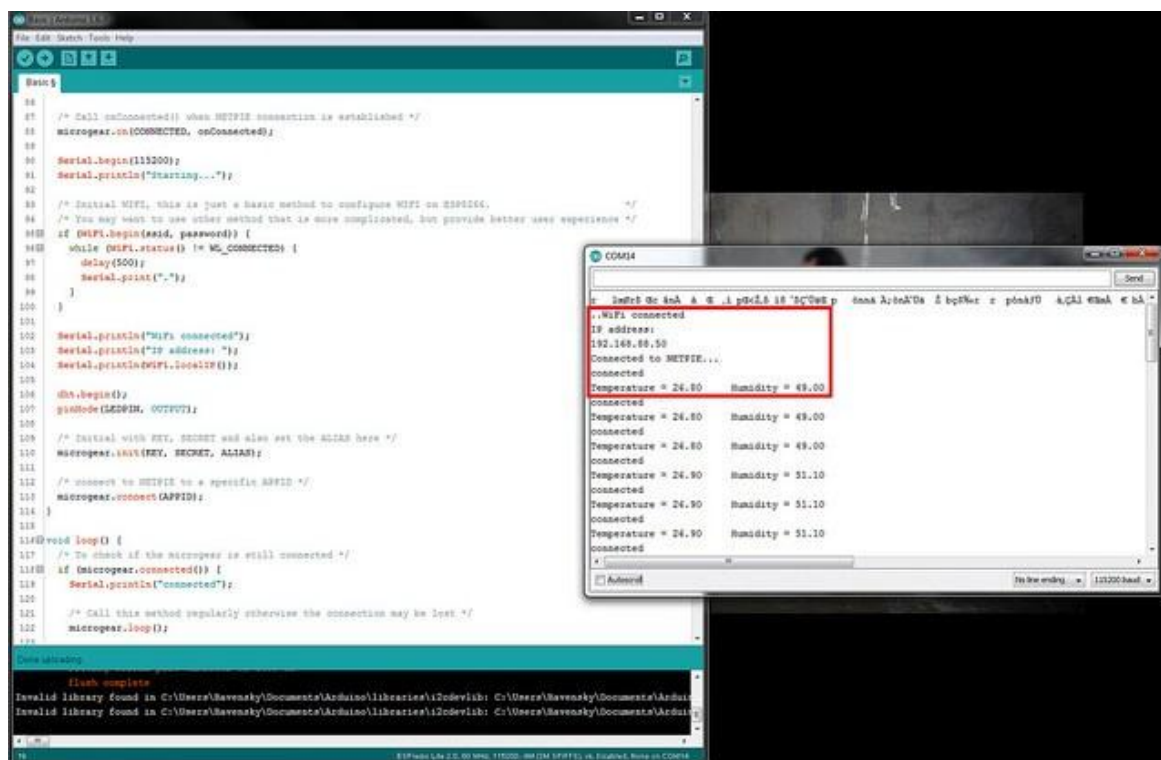
22) เมื่อกด SAVE จะแสดงข้อความที่ถูกส่งมาทางหน้า Widget ที่เราสร้าง



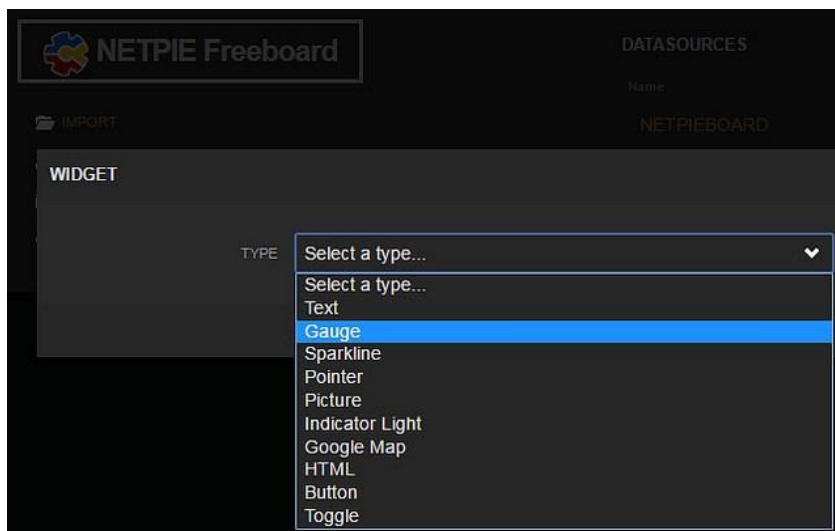
23) ขั้นตอนต่อไปจะทำการส่งค่าที่อ่านได้จากเซ็นเซอร์ DHT22 และควบคุมการติดดับของ LED บนบอร์ด ESPresso Lite กันครับ โดยสามารถดาวน์โหลดโค้ดตัวอย่างได้ [ที่นี่](#)



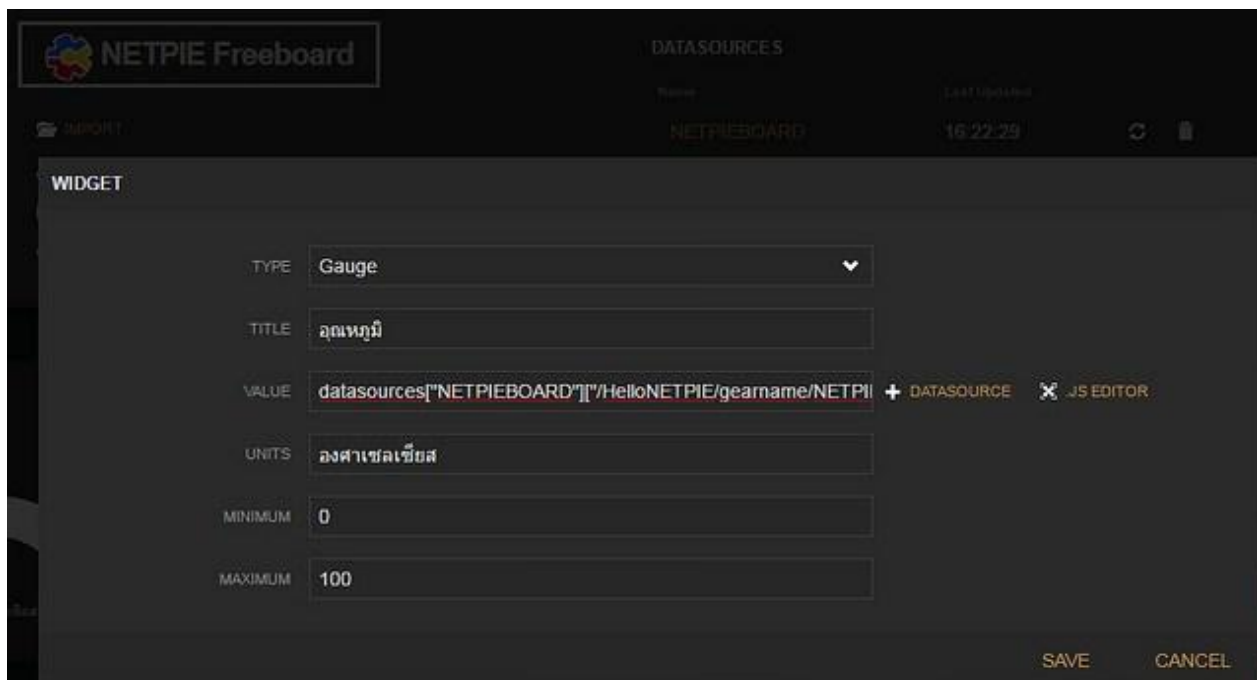
24) ทำการเปิดโค้ดตัวอย่าง อัปโหลดโค้ดไปยังบอร์ด และเปิด Serial port จะแสดงข้อมูลอุณหภูมิ ความชื้นที่ขึ้นไปยัง netpie.io (ในฟังก์ชัน void loop อย่าลืมแก้ ALIAS ใน microgear.chat ด้วยนะ ครับ)



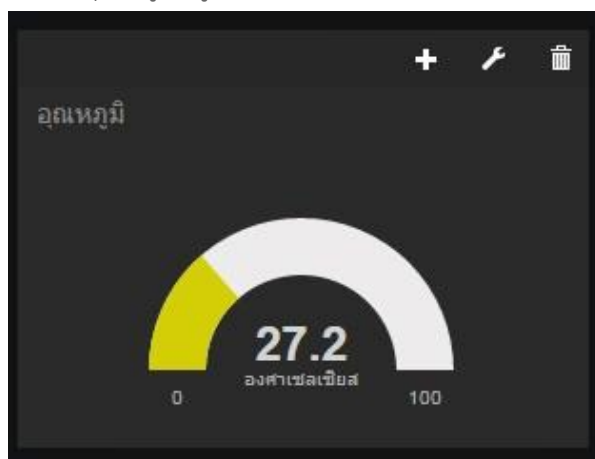
25) กลับมายังหน้า Dashboard ให้ทำการสร้าง Widget สำหรับแสดงค่าอุณหภูมิโดยคลิก ADD PANE —> (+) —> Gauge



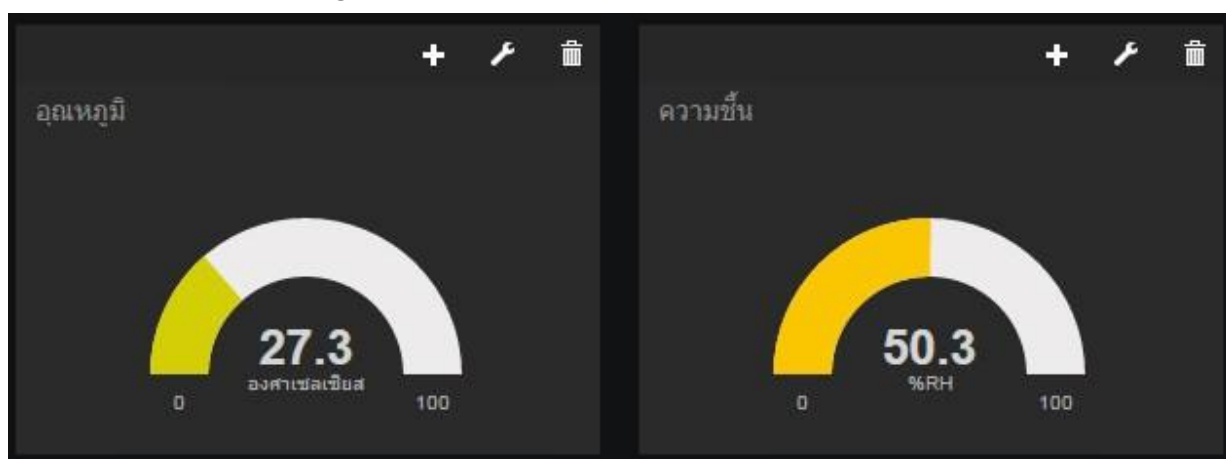
26) ทำการเลือก +DATASOURCE ให้ถูกต้อง โดยเลือก
 datasources["NETPIEBOARD"][/HelloNETPIE/gearname/NETPIName/Temperature"]
 (NETPIEBOARD คือ Name และ NETPIName คือ DEVICE ALIAS ตามข้อ 18)



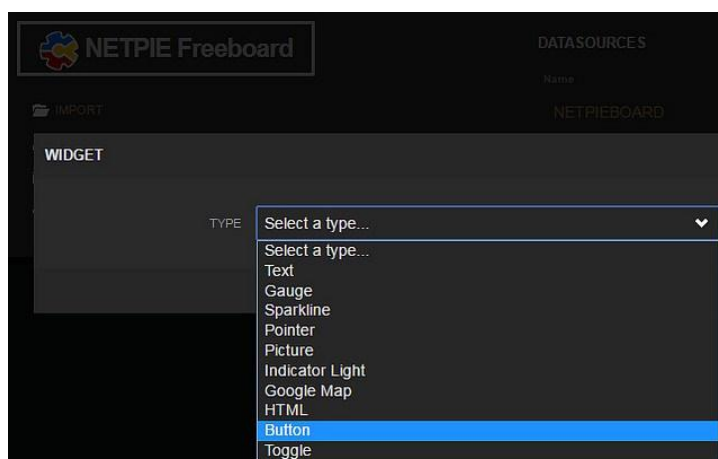
27) กด Save จะแสดงค่าอุณหภูมิที่ถูกส่งมาจาก ESPresso Lite



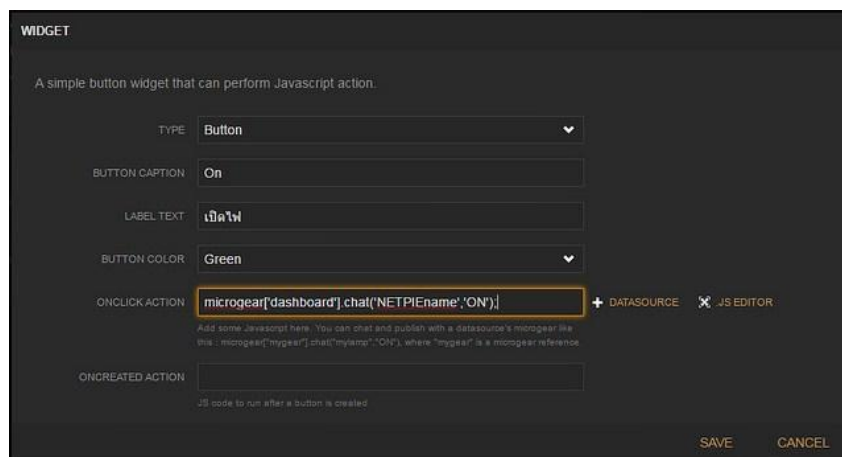
28) ทดลองเพิ่ม Widget ความชื้น



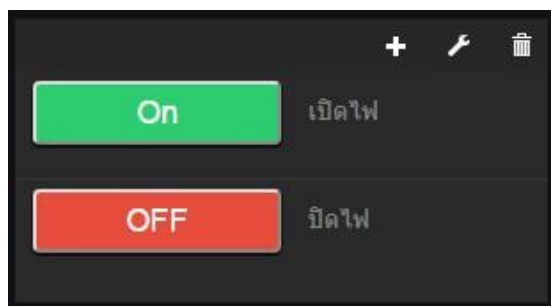
29) ขั้นตอนต่อไป จะเป็นการสร้าง Widget สำหรับสั่งเปิด - ปิด LED บนบอร์ด ESPresso Lite โดยทำการสร้าง Widget Button



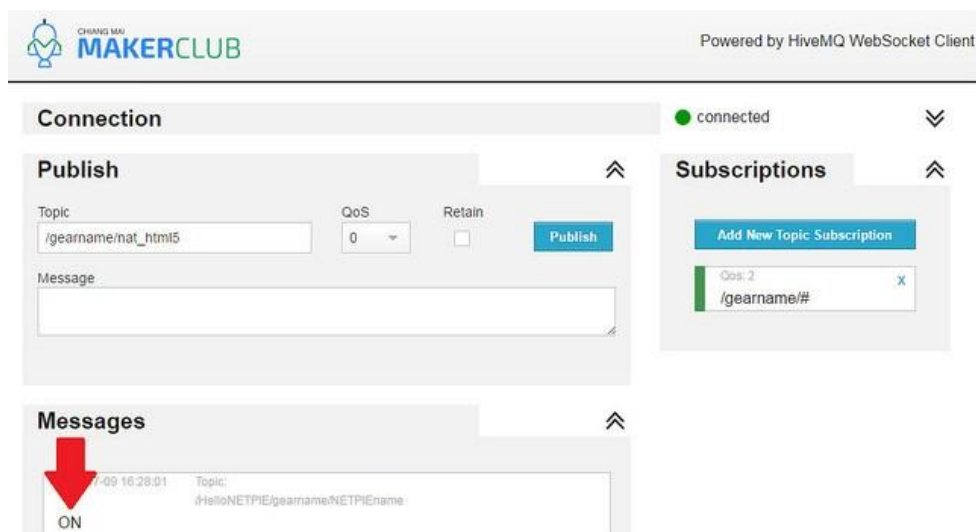
30) ในช่อง ONCLICK ACTION ให้เพิ่ม `microgear["dashboard"].chat("NETPIName","ON");` ซึ่งจะส่งคำว่า ON มาให้ ESPresso Lite รับ (dashboard คือ MICROGEAR REFERENCE และ NETPIName คือ DEVICE ALIAS ตามชื่อที่เราตั้งในการ Add Dashboard)



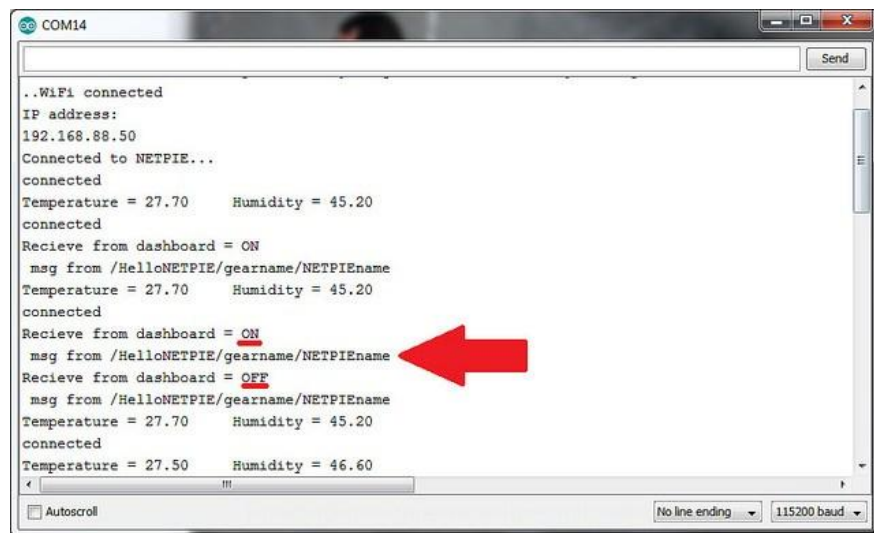
31) สร้างปุ่ม OFF ขึ้นมาอีกปุ่มเพื่อปิด LED โดยแก้ ONCLICK ACTION ให้เป็น `microgear["dashboard"].chat("NETPIName","OFF");`



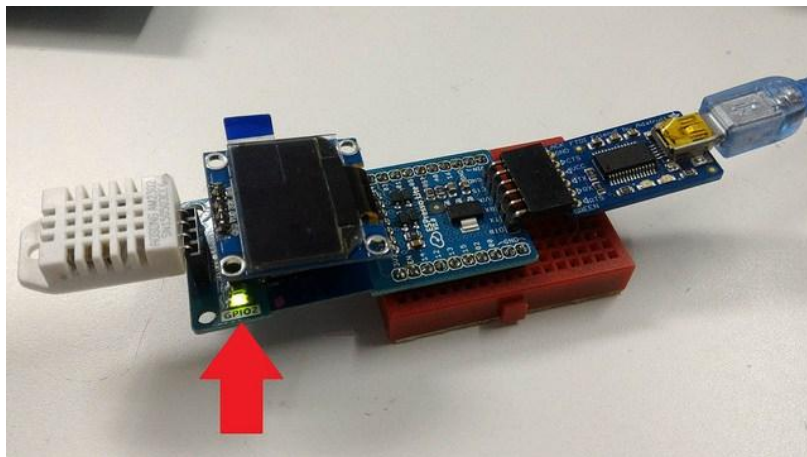
32) เปิดหน้า [MQTT Websocket Client](#) เมื่อทำการคลิกที่ปุ่ม ON ใน Dashboard จะเห็นข้อความที่ส่งเข้ามายัง DEVICE ALIAS



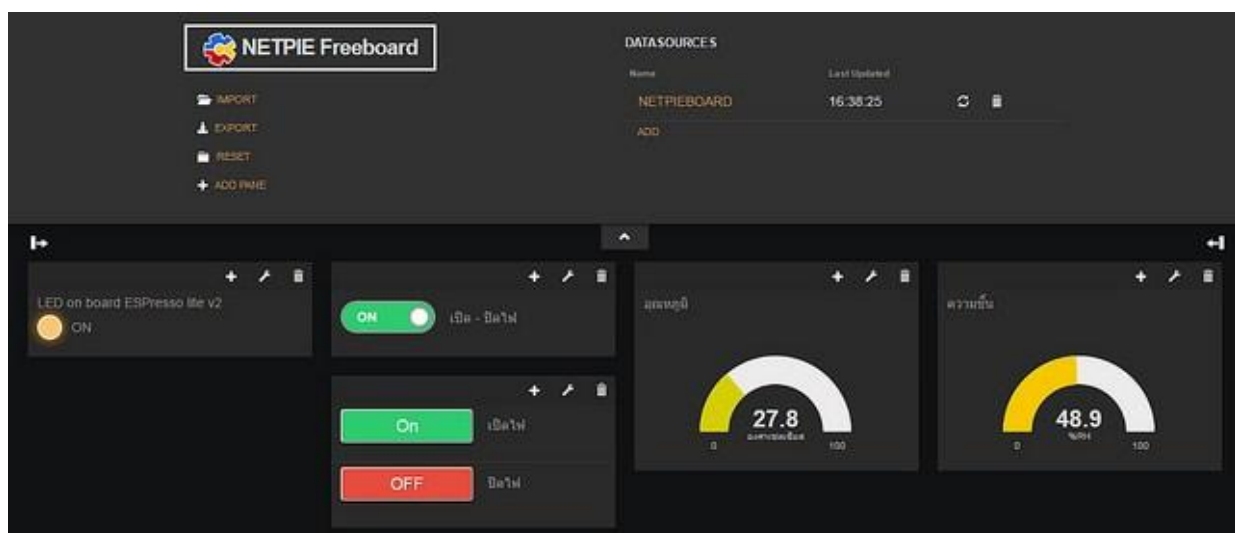
33) เปิดหน้าต่าง Serial port จาก Arduino IDE จะแสดงข้อความที่รับเข้า



34) ทดลองกดปุ่ม ON – OFF และสังเกต LED บนบอร์ด ESPresso Lite



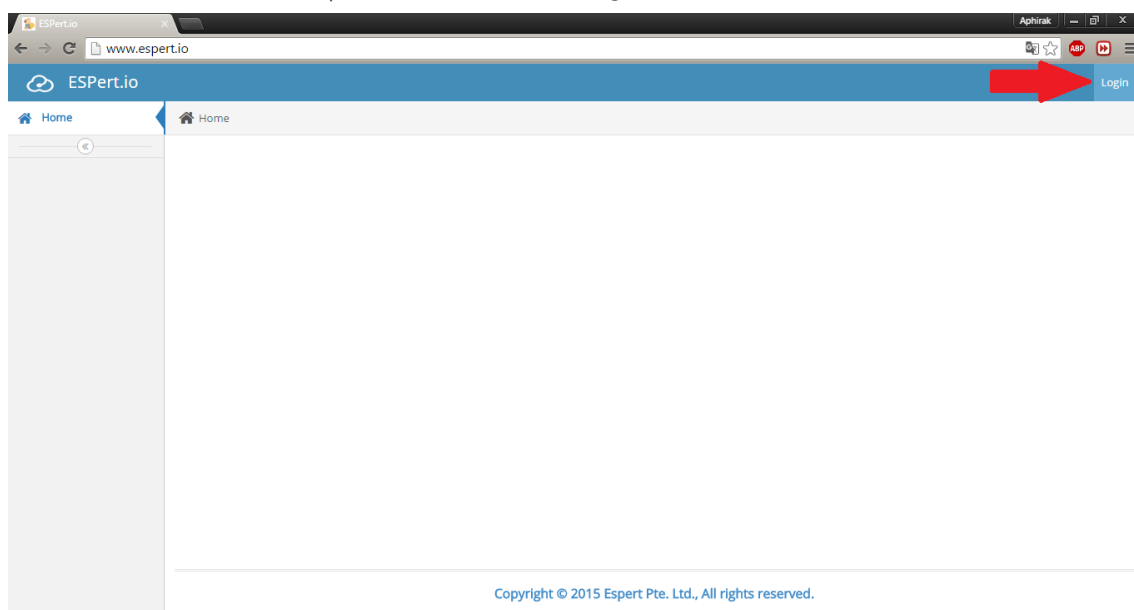
35) ทดลองเพิ่ม Widget ต่าง ๆ โดยสามารถดาวน์โหลด Dashboard ตัวอย่างได้จาก [ที่นี่](#)



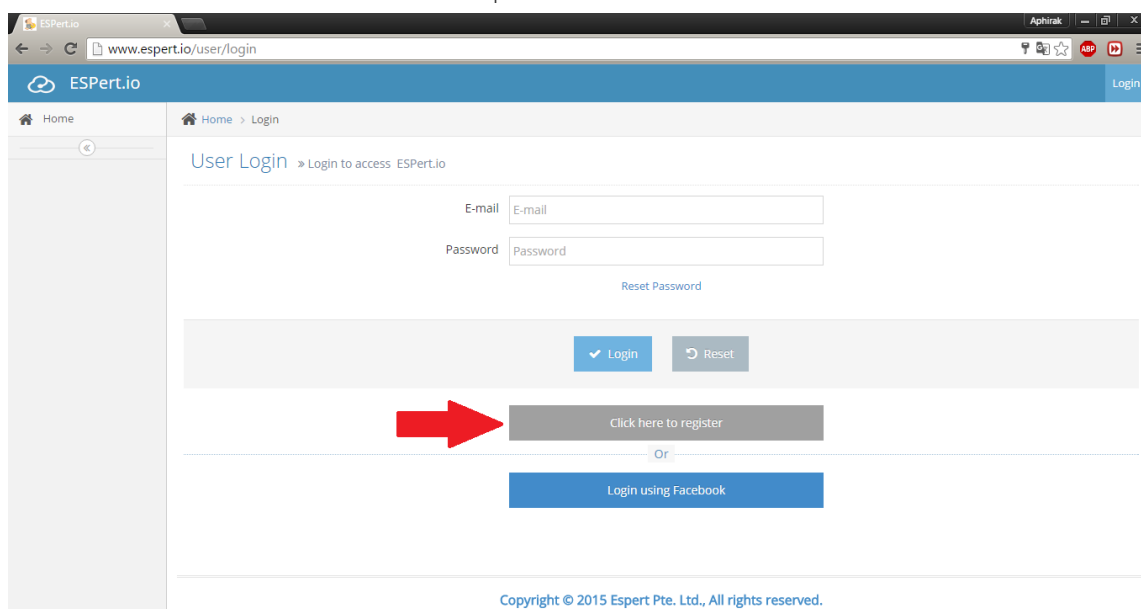
ขั้นตอนการสมัครใช้งาน ESPert.io เพื่อแจ้งเตือนข้อความบนโทรศัพท์มือถือ

ESPert.io คืออะไร ESPert.io คือ เว็บเซอร์วิสสำหรับส่งการแจ้งเตือนผ่านโทรศัพท์มือถือ หรือ Notification หากใครยังไม่ทราบว่า Notification คืออะไรนั้น เอาง่าย ๆ ก็เหมือนกับการแจ้งเตือนแชทของเฟสบุ๊ก หรือไลน์นั่นเอง เมื่อมีคนส่งข้อความเข้ามาก็จะส่งเสียง และข้อความแจ้งเตือนให้เราทราบ อีกทั้งยังส่ง Push Notification ผ่าน http GET ซึ่งง่ายต่อการใช้งานเหมาะสำหรับนักพัฒนา IoT โดย espert.io ถือเป็นหนึ่งในบริการที่น่าใช้ และนำไปต่อยอดใช้งานในด้านต่าง ๆ ได้อย่างมากมาย โดยขั้นตอนการสมัครใช้งานมีดังต่อไปนี้

1) เข้าไปยังเว็บไซต์ espert.io จากนั้นคลิกที่Login



2) ลงทะเบียนใช้งานโดยคลิกที่ปุ่ม Click here to register



3) ทำการกรอกข้อมูลพื้นฐานให้ครบทุกช่อง

ESPert.io

Home > Register

User Registration

Please enter your information

Username: CNX_CMMC

E-mail:

Password:

Confirm Password:

Name: aphirak sang-ngenchai

Gender: Male

☒ I agree with the Terms and Conditions

Register Reset

4) เมื่อกด Register จะแสดงหน้าจอให้ยืนยันในอีเมล

ESPert.io

Home > Register

User Registration

Please enter your information

Registration

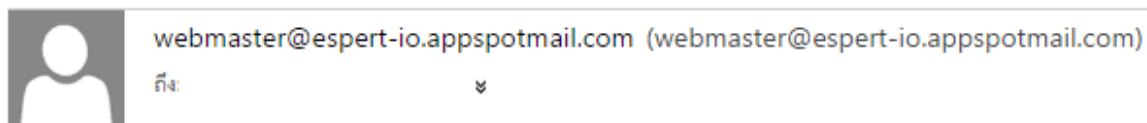
Success.

A confirmation e-mail was sent to you,
please follow the instruction to enable your account.

Copyright © 2015 Espert Pte. Ltd., All rights reserved.

5) ให้ทำการเข้าไปยืนยันในอีเมลที่เราได้สมัครเอาไว้

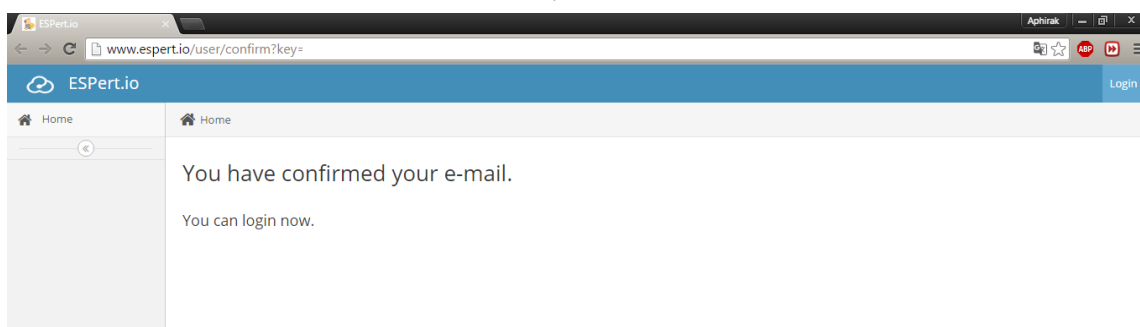
Please confirm your e-mail to access ESPert.io



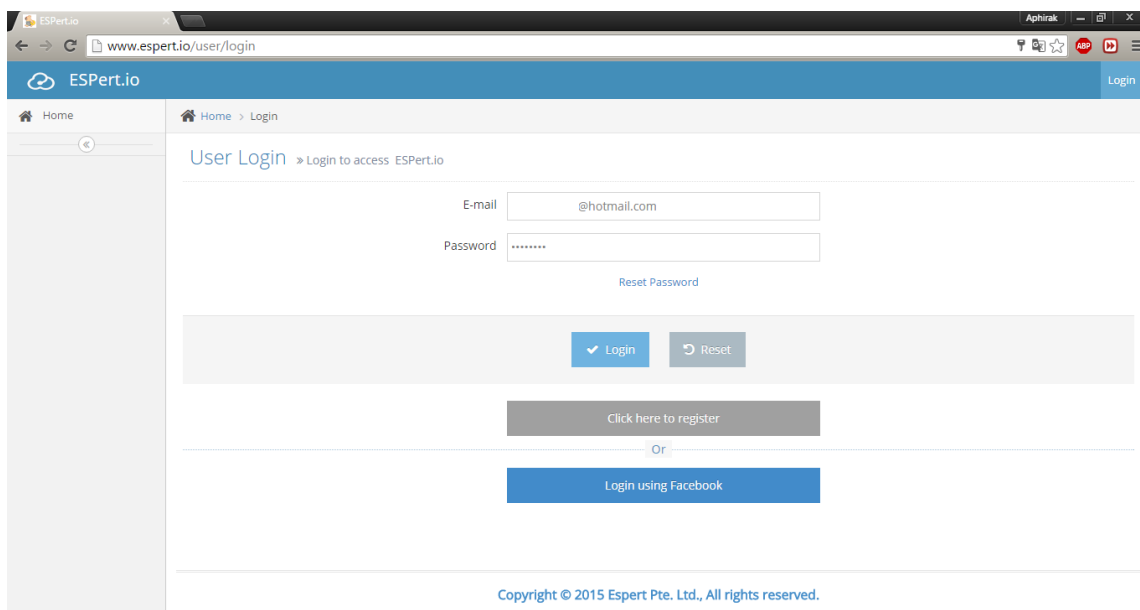
Dear aphirak sang-ngenchai ,
[Press this link to confirm your e-mail address.](#)

Best Regards,
 ESPert.io team

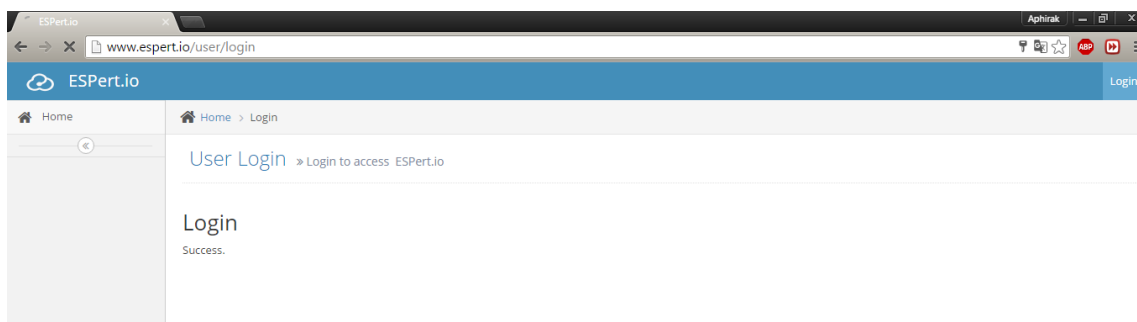
6) เมื่อยืนยันเรียบร้อยแล้วจะกลับมายังเว็บ espert.io เพื่อแสดงการยืนยันการสมัคร



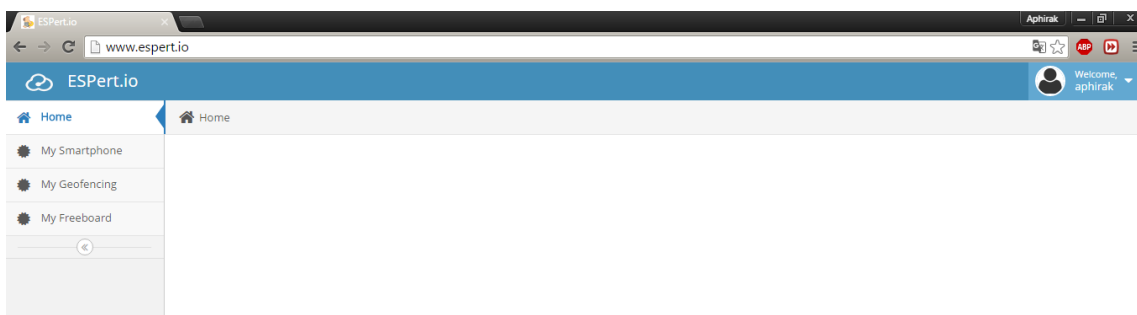
7) ให้ทำการเข้าระบบด้วยการใส่ E-mail และ Password ที่เราได้สมัครเอาไว้



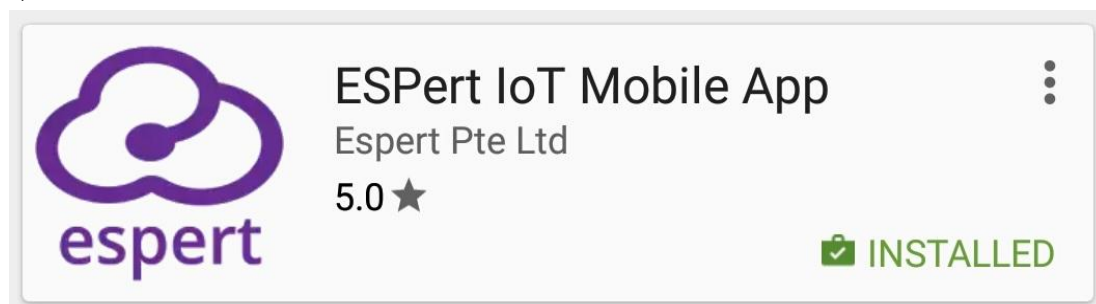
8) เมื่อเข้าระบบสำเร็จจะแสดงหน้าจอตั้งภาพด้านล่าง



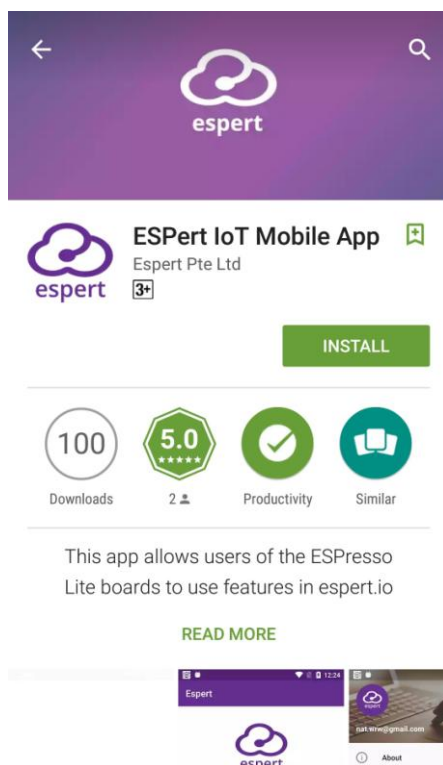
9) แสดงหน้าเว็บพร้อมใช้งาน



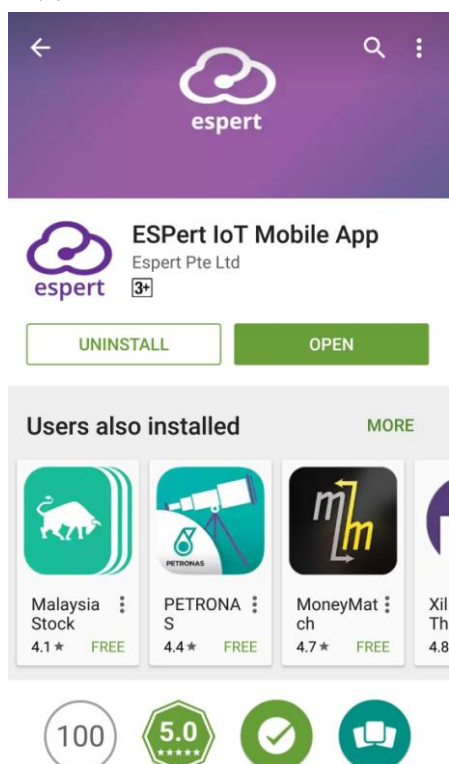
10) จากนั้นให้ทำการดาวน์โหลด Application สำหรับโทรศัพท์มือถือ โดยสามารถดาวน์โหลดได้จาก play store สำหรับ android หรือ IOS สามารถดาวน์โหลดจากลิงก์นี้ได้ [ลิงค์ดาวน์โหลด App](#) (ในที่นี้ขออนุญาตอธิบายวิธีการลงบนโทรศัพท์ android)



11) ทำการติดตั้ง Application




12) เมื่อติดตั้งเสร็จให้เปิด Application ขึ้นมา



13) จะได้หน้าจอคั่นดังภาพด้านล่าง ให้เราทำการใส่ Email และ Password ที่ได้ทำการสมัครไว้เพื่อเข้าใช้งาน

ESPert IoT Mobile App


espert

Log in with Facebook

or

Email

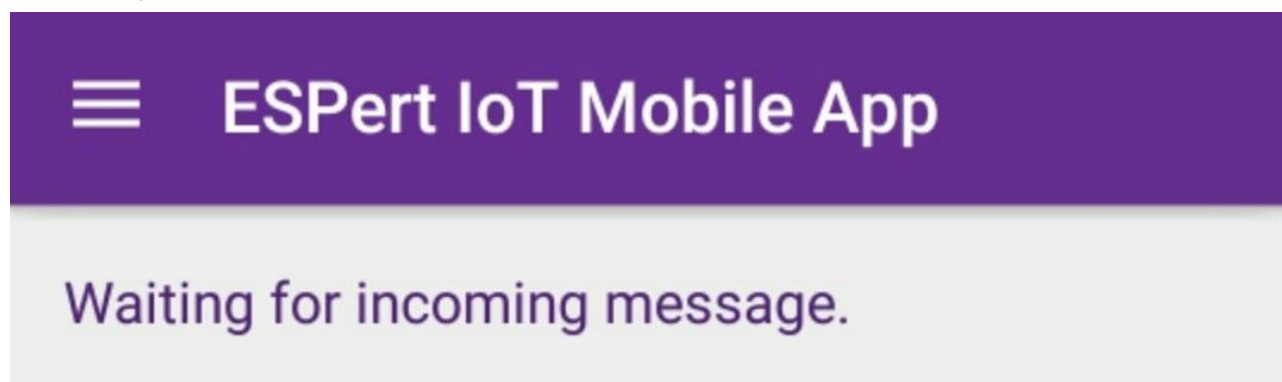
@hotmail.com

Password

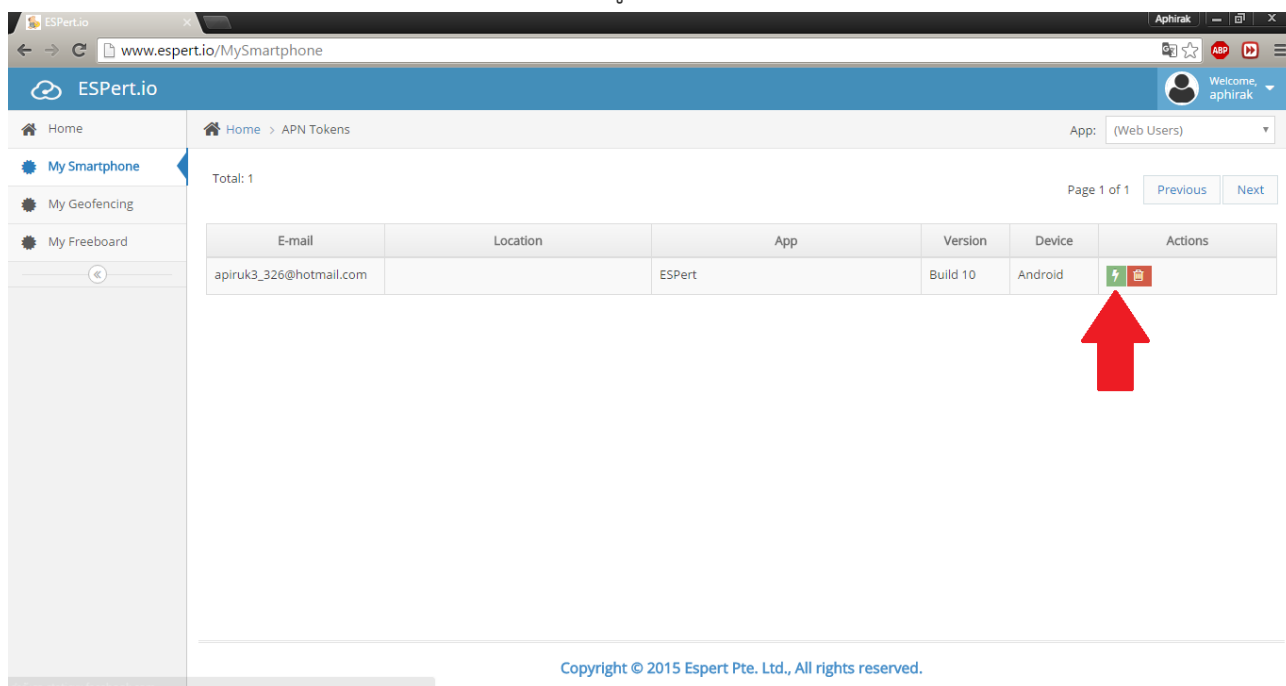
.....

Log in

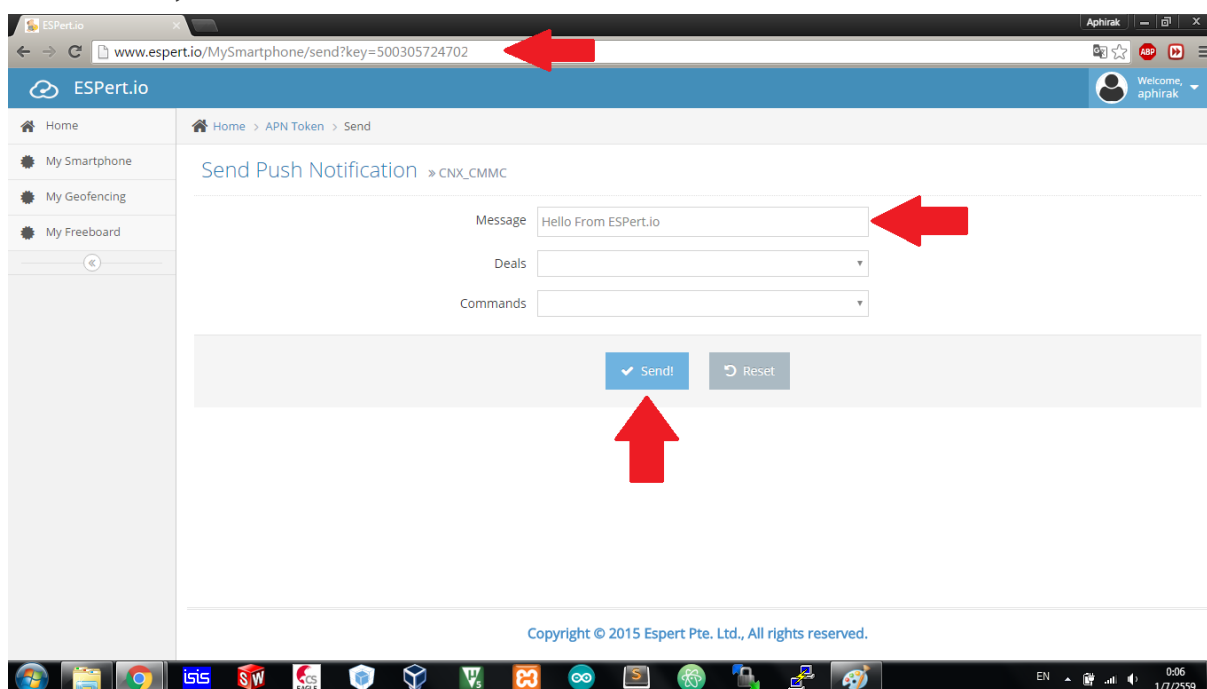
14) เมื่อเข้าใช้งานได้แล้ว จะแสดงข้อความ Waiting for incoming message. เพื่อรอข้อความจากเว็บ espert.io



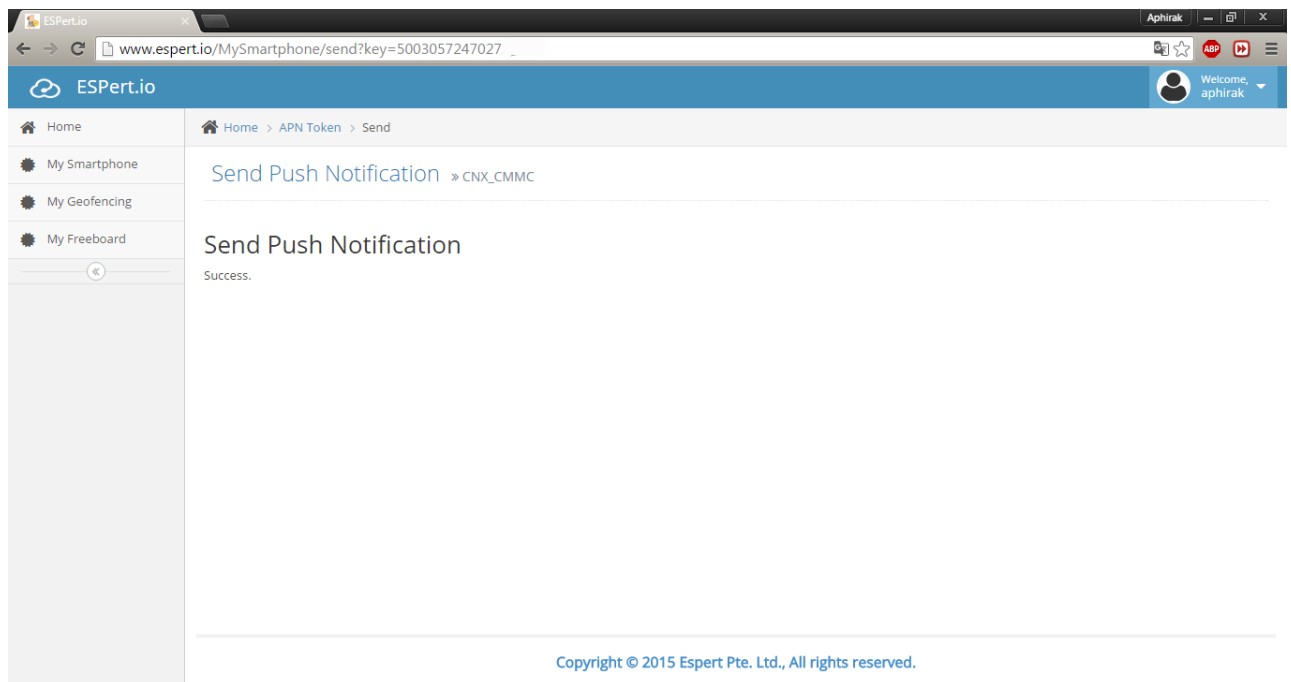
15) กลับมาที่เว็บ espert.io จะเห็นว่ามีอุปกรณ์เพิ่มเข้ามา ซึ่งก็คือโทรศัพท์ของเรานั่นเอง ต่อไปเราจะทดลองส่งข้อความเข้าโทรศัพท์กัน โดยให้คลิกที่รูปสายฟ้าสีเขียวดังภาพด้านล่าง



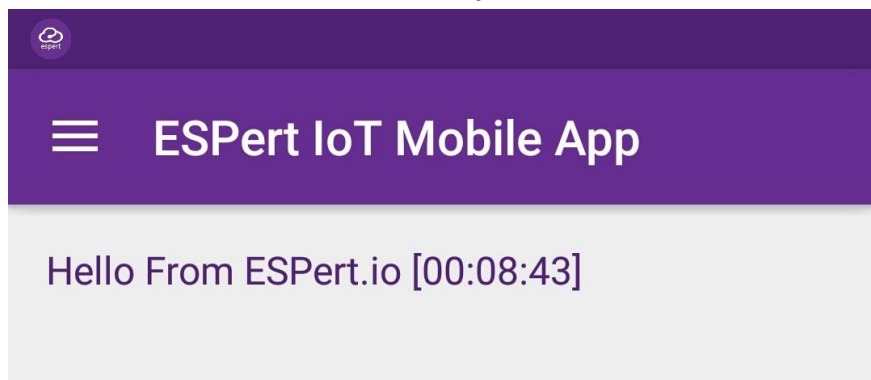
16) ในหน้า Send Push Notification ก็จะมีช่องให้เราใส่ข้อความ และปุ่มกดส่ง อีกทั้งเรายังสามารถนำ Key หรือรหัสโทรศัพท์ของเราไปใช้งานได้อีกด้วย



17) เมื่อกด Send จะแสดงข้อความว่ามีการส่งข้อความสำเร็จ



18) ข้อความที่เราพิมพ์จากเว็บ espert.io ก็จะถูกส่งมายังโทรศัพท์มือถือของเราดังภาพด้านล่าง



การจัดวางวัตถุ (UI) ใน React Native

การจัดหน้าตาใน react-native สามารถทำได้โดยใช้ FlexBox API ซึ่ง จะประกอบไปด้วย flexDirection, justifyContent และ alignItems การปรับตั้งค่า flexDirection จะสามารถเป็นไปได้ 2 ค่า คือ 'row' จะแสดงผลในแนวนอน และ 'column' จะเป็นการแสดงผลในแนวตั้ง ค่าปริยายของ flexDirection จะเป็น 'column' จะแสดงผลเป็นแนวตั้งดังภาพ



หากปรับค่า flexDirection เป็น row จะทำให้วัตถุแสดงผลในแนวนอนดังนี้



การจัดกลางสามารถใช้ flexDirection, justifyContent และ alignItems ร่วมกันได้ดังภาพข้างล่าง
ซึ่งใช้ flexDirection เป็น column

```

1 import React, { Component } from 'react';
2 import { AppRegistry, View } from 'react-native';
3
4 class AlignItemsBasics {
5   render() {
6     return (
7       // Try setting `alignItems` to 'flex-start'
8       // Try setting `justifyContent` to 'flex-end'.
9       // Try setting `flexDirection` to 'row'.
10      <View style={{
11        flex: 1,
12        flexDirection: 'column',
13        justifyContent: 'center',
14        alignItems: 'center',
15      }}>
16        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />
17        <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />
18        <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />
19      </View>
20    );
21  }
22 }
23
24 AppRegistry.registerComponent('AlignItemsBasics', () => AlignItemsBasics);

```



No Errors Show Details

หากจัดในแบบเดียวกันโดยใช้ flexDirection เป็น row จะแสดงผล การจัดกลาง ในแนวนอน ดังภาพ

```

1 import React, { Component } from 'react';
2 import { AppRegistry, View } from 'react-native';
3
4 class AlignItemsBasics {
5   render() {
6     return (
7       // Try setting `alignItems` to 'flex-start'
8       // Try setting `justifyContent` to 'flex-end'.
9       // Try setting `flexDirection` to 'row'.
10      <View style={{
11        flex: 1,
12        flexDirection: 'row',
13        justifyContent: 'center',
14        alignItems: 'center',
15      }}>
16        <View style={{width: 50, height: 50, backgroundColor: 'powderblue'}} />
17        <View style={{width: 50, height: 50, backgroundColor: 'skyblue'}} />
18        <View style={{width: 50, height: 50, backgroundColor: 'steelblue'}} />
19      </View>
20    );
21  }
22 }
23
24 AppRegistry.registerComponent('AlignItemsBasics', () => AlignItemsBasics);

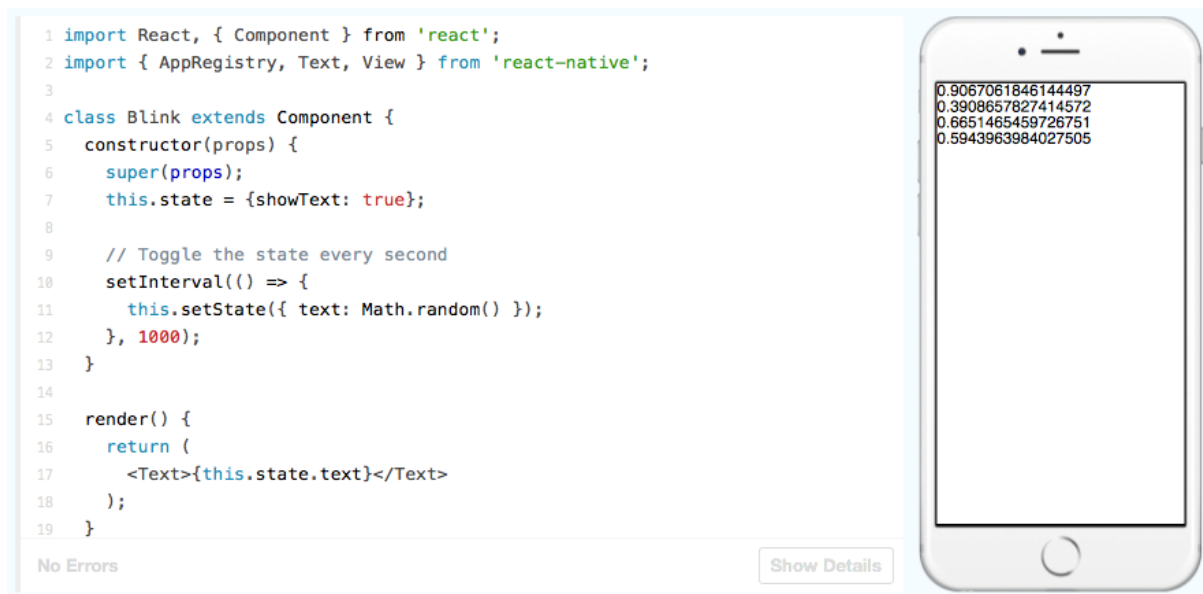
```



No Errors Show Details

เอกสารพารามิเตอร์โดยละเอียดสามารถดูเพิ่มเติมได้จากเอกสารของ react-native ต้นทางได้ที่
<https://facebook.github.io/react-native/docs/layout-props.html>

การอัปเดตค่าใน Views (UI) สามารถทำได้ผ่านคำสั่ง `setState` ดังตัวอย่างต่อไปนี้



หมายเหตุ สามารถดูโค้ดฉบับเต็มได้ที่ : <https://facebook.github.io/react-native/docs/state.html>

การติดต่อกับเครื่องแม่ข่ายสามารถใช้ :

[Fetch API](https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API) (https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API)

การสร้าง POST request ตัวอย่างต่อไปนี้ หากไม่ระบุว่าเป็น method จะใช้ค่า GET ซึ่งเป็นค่าปริยาย

```

fetch('https://mywebsite.com/endpoint/', {
  method: 'POST',
  headers: {
    'Accept': 'application/json',
    'Content-Type': 'application/json',
  },
  body: JSON.stringify({
    firstParam: 'yourValue',
    secondParam: 'yourOtherValue',
  })
})

```


และสามารถจัดการกับ request ที่เกิดขึ้นได้ดังตัวอย่างต่อไปนี้

```
getMoviesFromApiAsync() {  
  return fetch('http://facebook.github.io/react-native/movies.json')  
    .then((response) => response.json())  
    .then((responseJson) => {  
      return responseJson.movies;  
    })  
    .catch((error) => {  
      console.error(error);  
    });  
}
```

สามารถดูรายละเอียดเพิ่มเติมได้ที่ : <https://facebook.github.io/react-native/docs/network.html>