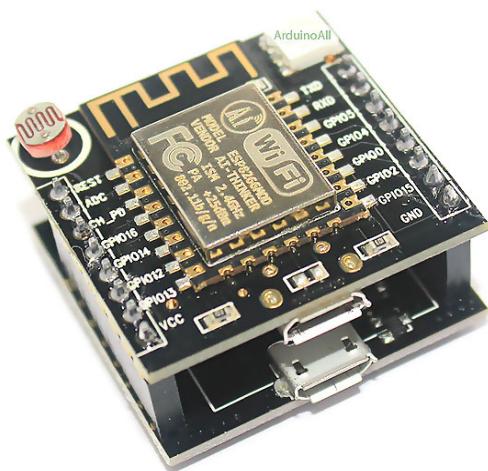


Witty + NetPie

NETPIE SMART LIGHT INDICATOR + FREEBOARD

สวัสดีครับ วันนี้ผมก็มีโปรเจกต์เล็กๆ มานำเสนอนะครับ
เป็นโปรเจกต์สำหรับคนที่อยากรู้สึก NETPIE แบบรวดเร็ว ง่ายๆ และไม่ต้องเตรียม Hardware
นาน

เพราะในโปรเจกต์นี้ เราจะมาลองเล่นกับเพื่อนใหม่ กับ "Mini NodeMCU" กันครับผม



เนื่องจากตัวบอร์ดนี้ มี LDR (ตัวด้านหน้าปรับค่าได้ตามแสง) ติดมาบนบอร์ดตั้งแต่เกิดเลยก็
ว่าได้ครับ
 ผมเลยจะนำมาทำเป็นตัววัดระดับแสงสว่าง สามารถนำไปใช้ในห้อง ทางเดิน หรือสถานที่
 ต่างๆได้นะครับ

โดยจะสามารถวัดระดับความสว่าง ณ ปัจจุบัน ได้ และสามารถตั้งให้จดจำ ค่าความสว่างเมื่อ
 ตอนเปิดไฟ กับตอนปิดไฟเอาไว้

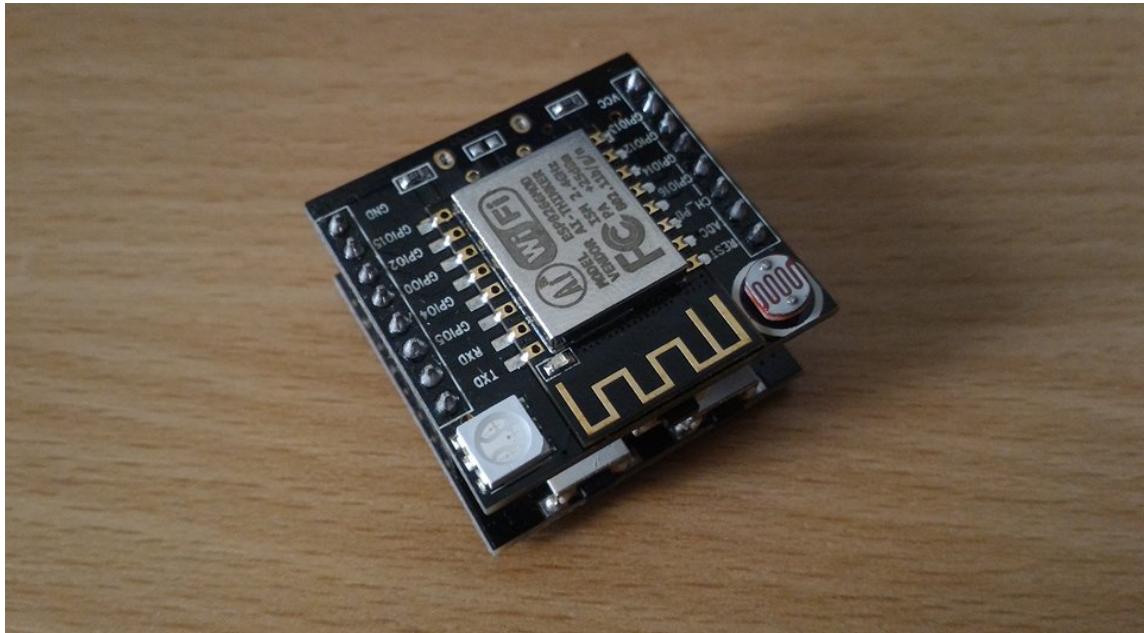
โดยบอร์ดจะสามารถบอกรอ (เดา) ได้ครับว่าตอนนี้ หลอดไฟของเรานั้นเปิด หรือ ปิดอยู่
 นอกจากนี้ที่ตัวบอร์ดก็ยังมี หลอด LED แบบ 3 สี (RGB) ติดมาด้วยเลย
 ซึ่งเราสามารถใช้แสดงผลสถานะของบอร์ดได้ครับ

เรียกได้เลยว่า 'ไม่ต้องเตรียมฮาร์ดแวร์' 'ไม่ต้องใช้პროტოบอร์ด'
 'ไม่ต้องต่อสายไฟอะไรมาก แต่ก็สามารถมีอุปกรณ์ IoT ที่ง่าย'

และทำงานได้จริง เป็นชิ้นเป็นอันได้แล้วครับผม >_<'

ตัวต้านทานปรับค่าได้ตามแสง (LDR) ถูกต่อໄວกับขา A0
RGB LED ขาควบคุม “สีแดง” ถูกต่อໄວกับขา GPIO 15
RGB LED ขาควบคุม “สีเขียว” ถูกต่อໄວกับขา GPIO 12
RGB LED ขาควบคุม “สีแดง” ถูกต่อໄວกับขา GPIO 13
ปุ่ม ที่ติดมาให้บนซีกบนของบอร์ด ถูกต่อໄວกับขา GPIO 4

STEP 1 : บอร์ดนี้มีดียังไง???



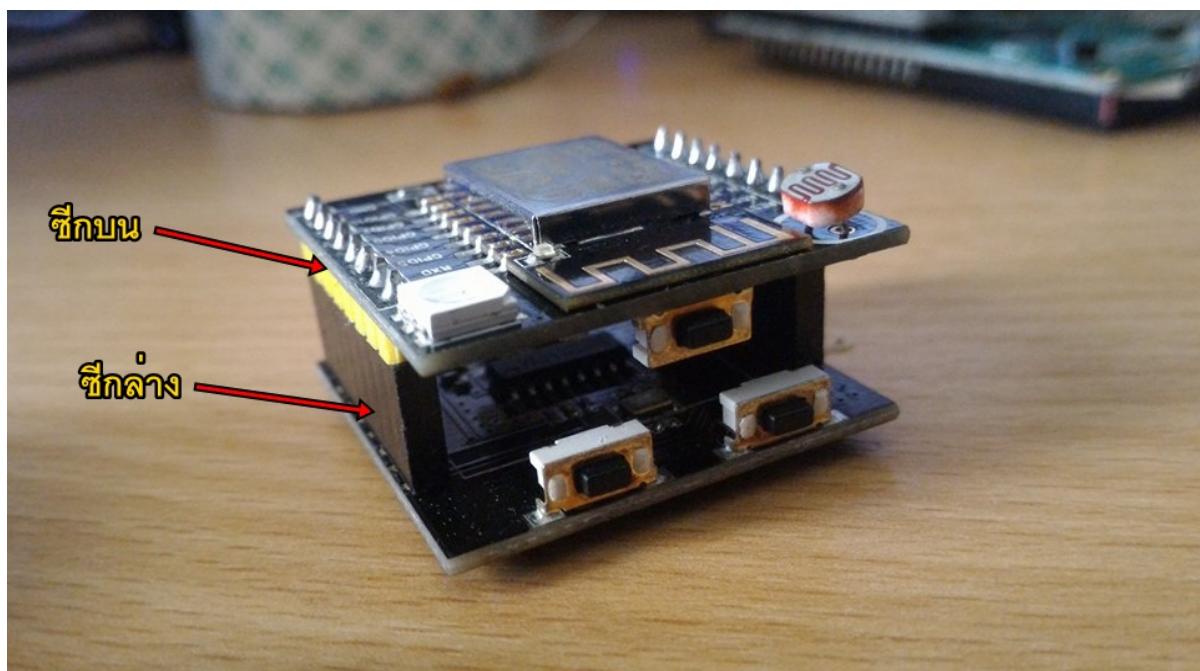
https://blog.netpie.io/archives/570/13652483_884273775039207_961647034_n

สำหรับบอร์ดนี้จะรับก็จะมีชื่อเดิมว่า **Mini NodeMCU ESP 8266 ESP-12F serial WiFi
ESP8266**

ซึ่งจะใช้ชิพ ESP-12F Module ที่มาพร้อมกับ Espressif ESP8266EX SoC โดยรองรับการเชื่อม
ต่อแบบ WiFi 802.11 b/g/n

โดยบอร์ดชุดนี้จะทำการบัดกรีขามาให้เรียบร้อยแล้วนะครับ (พวกขา GPIOs, VCC, GND,
Reset, ADC, และ UART)

โดยบอร์ดนี้จะมาเป็นชุด ซึ่งประกอบด้วย ซีกบน และ ซีกล่าง ซึ่งสามารถถอดแยกเป็น 2 ชิ้นได้
ครับผม

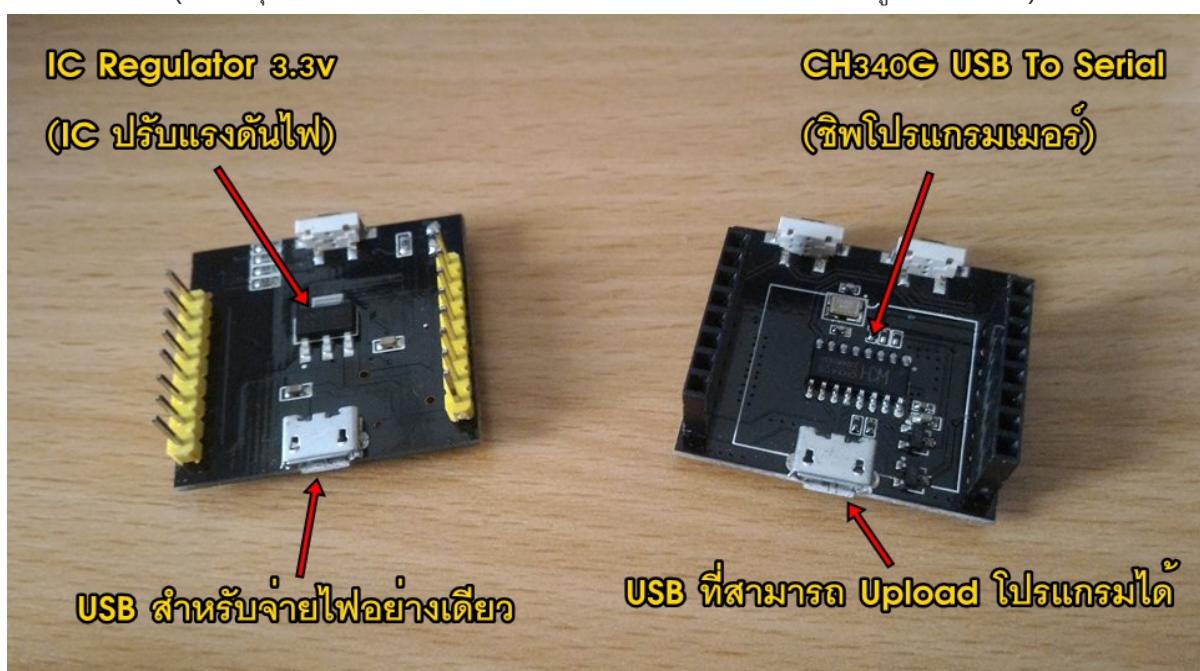


https://blog.netpie.io/archives/570/13735347_884273778372540_1594487111_n-2

และเห็นว่าบอร์ดนี้ มี USB ออยู่ 2 พอร์ต!!! อันนี้ให้เข้าใจตรงกันนะครับว่า พอร์ตที่อยู่ซีกบน (ด้านบน) จะเป็นพอร์ตสำหรับการจ่ายไฟเข้า (VCC) อย่างเดียว (สังเกตว่าจะเห็น IC Regulator ติดอยู่ข้างล่าง)

ส่วนพอร์ตที่อยู่ด้านล่าง จะเป็นพอร์ตที่เราสามารถใช้ในการอัปโหลดโค้ดโปรแกรมได้ครับ

(สังเกตว่าจะเห็นชิป USB to Serial แบบ CH340 นอนอยู่นั่นเองครับ)

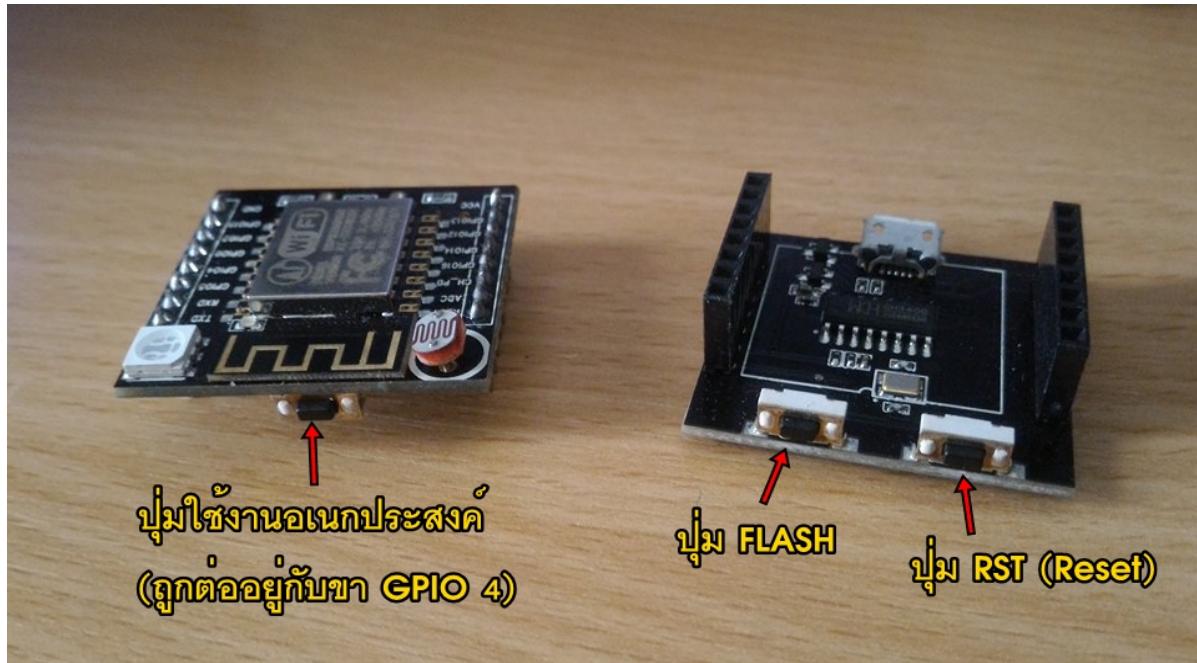


<https://blog.netpie.io/archives/570/cats-17>

และอีกส่วนคือ ปุ่มกด จะมีทั้งหมด 3 ปุ่มนะครับ โดย ปุ่มที่อยู่ที่ซีกบนของบอร์ดนั้น จะต่ออยู่กับ GPIO4

ครับ สำหรับโปรเจกต์ไหนที่ต้องใช้ปุ่มกด ก็ไม่ต้องต่อปุ่มเองให้วุ่นวายครับผม เพราะมีปุ่มฯนี้ ติดมาให้ตั้งแต่แรกเลยครับ

ส่วนอีก 2 ปุ่มที่อยู่กับซีกที่เป็นฐานด้านล่าง จะเป็นปุ่ม FLASH กับ RST (Reset) ครับผม



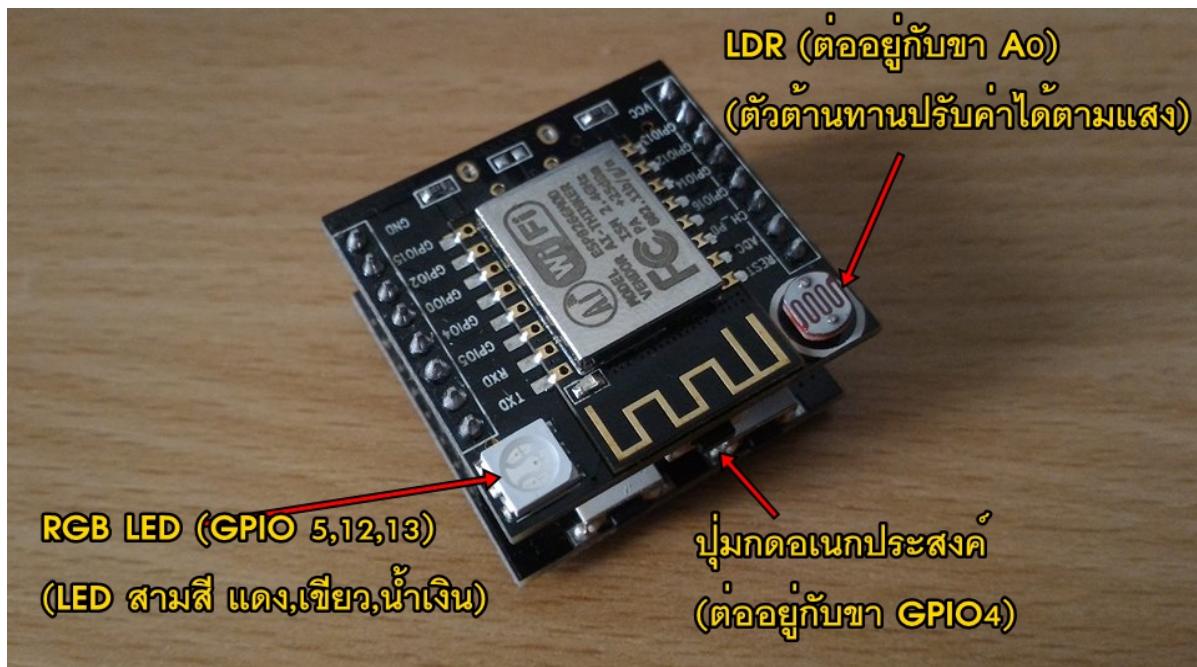
https://blog.netpie.io/archives/570/13689453_884273788372539_527430057_n

ครับและสำหรับบอร์ดนี้ก็จะมีส่วนพิเศษกว่าครอนนั่นก็คือ มี "LDR (ตัวต้านทานปรับค่าได้ตามแสง)"

และ "RGB LED (หลอดไฟ LED 3 สี) ติดมาด้วยตั้งแต่เกิดเลยทีเดียว นอกจากนี้ยังมี "ปุ่ม" ที่ซีกด้านบน

ให้เราได้ใช้งานได้ทันที โดยแทบจะไม่ต้องต่อสายไฟอะไรให้วุ่นวายเลยด้วย นับได้ว่า เป็นบอร์ดที่สะดวก

สำหรับการทำโปรเจกต์ IoT ที่เล็กๆ และไม่เน้นด้าน Hardware และการต่อสายไฟมากๆเลย



https://blog.netpie.io/archives/570/13652483_884273775039207_961647034_n-1

สรุปขาต่างๆของอุปกรณ์ที่ติดมาบนบอร์ดก็จะเป็นตามนี้นะครับ
ตัวต้านทานปรับค่าได้ตามแสง (LDR) ถูกต่อไว้กับขา A0

RGB LED ขาควบคุม “สีแดง” ถูกต่อไว้กับขา GPIO 15

RGB LED ขาควบคุม “สีเขียว” ถูกต่อไว้กับขา GPIO 12

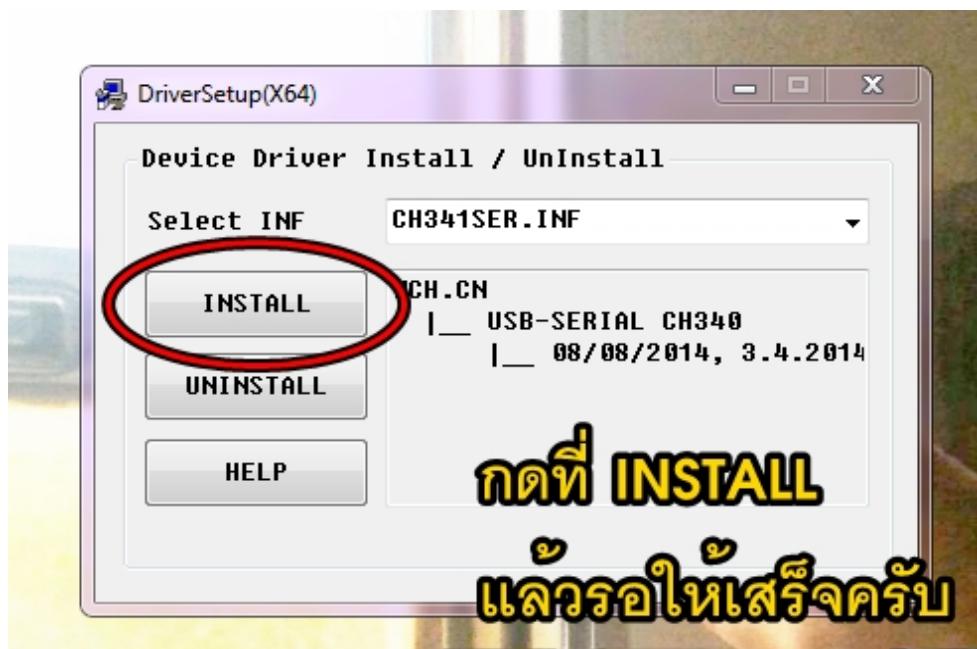
RGB LED ขาควบคุม “สีแดง” ถูกต่อไว้กับขา GPIO 13

ปุ่ม ที่ติดมาให้บนซีกบนของบอร์ด ถูกต่อไว้กับขา GPIO 4

**ดังนั้น ในการเล่นกับบอร์ดนี้ หากจะต่ออุปกรณ์อย่างอื่นพ่วง ควรต่อที่ขาอื่นๆนอกจากนี้
นะครับ**

ไม่งั้นจะทำงานกันระหว่างของที่ติดมาบนบอร์ดกับของที่เราติดเพิ่มเข้าไปครับผม
สำหรับใครที่เสียบสาย USB และไม่เห็นบอร์ด อาจจะเป็นเพราะยังไม่ได้ติดตั้ง

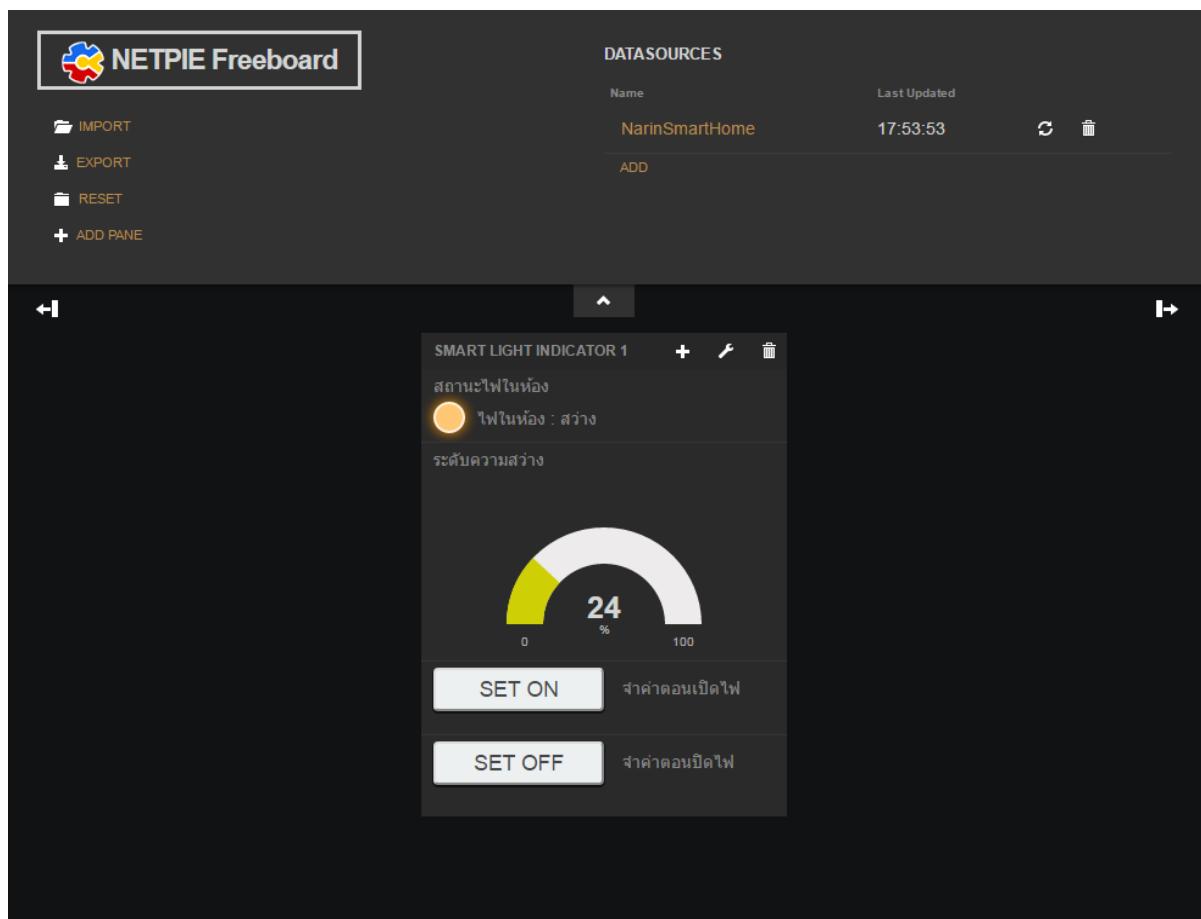
ไ/drive CH340G USB to Serial นะครับ ก็ให้ทำการดาวน์โหลด [จากที่นี่](#) ไปติดตั้งก่อนครับผม
เมื่อโหลดมาแล้วก็ให้เปิดไฟล์ และกดที่ Install ร่องนี้เสร็จก็ปิดหน้าไป



<https://blog.netpie.io/archives/570/cats-21>

จากนั้นคอมฯ ของเรา ก็จะมองเห็นบอร์ดแล้วครับผม ^_^

STEP 2 : ออกแบบโปรแกรม...



<https://blog.netpie.io/archives/570/untitled-5>

ครับ หลังจากที่ได้อธิบายคุณสมบัติของบอร์ดไปพอสมควรแล้ว ต่อไปก็จะเป็น การออกแบบโปรแกรมนะครับ เนื่องจากว่า ผู้มีต้องการสร้าง “เครื่องวัดระดับความสูง” โดยจะทำการแสดงผล และการควบคุมหั้งหมุดอยู่บน NETPIE Freeboard โดยที่พึ่งค์ชั้นหลักๆ ก็คือ

จะต้องแสดงค่าความสูง ที่ตอกกระทนบน LDR ออกมานะ เป็นเปอร์เซนต์ 0 – 100% และนอกจากนั้นก็คือ เราจะสามารถกดปุ่มบน Freeboard เพื่อให้อุปกรณ์นั้นจำค่าความสูง ในตอนที่เราปิดไฟไว้ และตอนที่เราปิดไฟ โดยในตอนที่เรากดปุ่ม เมื่อบอร์ดได้รับคำสั่งเข้ามา LED จะเป็นสีน้ำเงิน เป็นเวลา 0.2 วิ (เพื่อบอกเราว่า ตอนนี้มีคำสั่งเข้ามา) ซึ่งบอร์ดจะนำค่าที่เราจำไว้มาเฉลี่ยกัน (สูงสุด + ต่ำสุด และหารด้วย 2) เพื่อให้ได้ค่ากลางมา ซึ่งเมื่อแสงที่วัดได้ มากกว่าค่ากลาง นั่นอาจหมายความว่า ในห้องมีการปิดไฟ ก็จะให้บัน Freeboard แสดงสถานะว่าตอนนี้ไฟในห้องอาจจะเปิดอยู่ พร้อมทั้ง LED บนบอร์ดก็จะเป็นสี เชียว

และถ้าอ่านได้น้อยกว่าค่ากลางที่คำนวนได้ ก็จะให้แสดง ว่าไฟในห้องอาจจะปิดอยู่ พร้อม LED ติดเป็นสีแดงครับ นอกจากนี้ผมยังออกแบบให้ เมื่ออุปกรณ์เริ่มทำงาน จะมี LED ติดค้างเป็นสีน้ำเงินก่อน จนกว่าเราจะทำการ Config WiFiManager ให้เสร็จเรียบร้อย จนเมื่ออุปกรณ์ต่อ Wi-Fi สำเร็จ LED ก็จะเปลี่ยนเป็นสีเชียว และหากเชื่อมต่อไม่ได้ทันในเวลาที่กำหนดไว้ 3 นาที

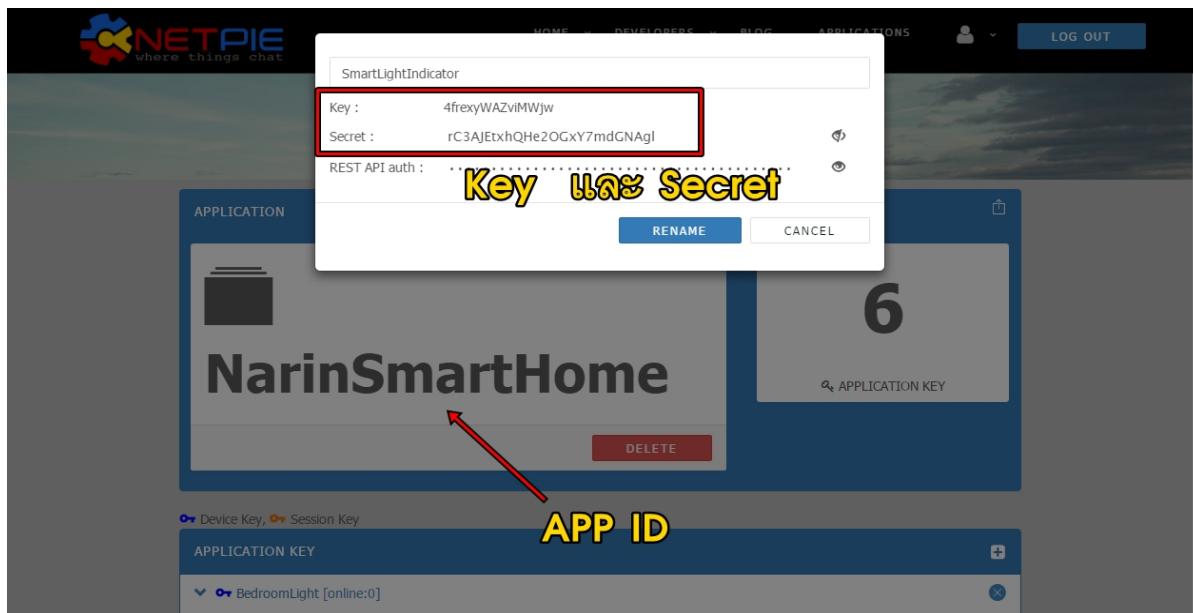
หรือการเชื่อมต่อไม่ปัญหา LED ก็จะเปลี่ยนเป็นสีแดง และบอร์ดก็จะ Reset ตัวเองครับและใน ที่สุดเมื่ออุปกรณ์

เชื่อมต่อกับ NETPIE สำเร็จ LED สีเชียว ก็จะกระพิบ และดับไปเป็นเวลา 3 วิ
จากนั้นก็จะติดใหม่เป็นไฟแสดงสถานะ แดง เชียว ตามการปิดปิดไฟแล้วครับ

STEP 3 : อัพโหลดโค้ดโปรแกรม...

ครับผม ที่นี่ก็จะมาทำการอัพโปรแกรมให้กับบอร์ดกันนะครับ แต่ก่อนอื่นก็ให้ทำการสร้าง Key บน NETPIE เพื่อนำมาให้อุปกรณ์นี้ใช้ในการเข้าสู่ระบบ (เข้าสู่ AppID ของเรา) ก่อนนะครับ ซึ่งในที่นี้ผมจะสร้าง Key ที่ชื่อ SmartLight1 และทำการเก็บ

Key และ Secret เอาไว้เพื่อนำไปใช้ใส่ในโค้ดของโปรแกรมครับผม



<https://blog.netpie.io/archives/570/cats-18>

สำหรับการสมัครใช้งาน NETPIE ไปจนถึงการสร้าง Key

สามารถศึกษาได้จากบทความนี้นะครับ >> [Smart Home Part 5](#)

และการออกแบบโปรแกรมที่บอกไว้ข้างต้น ทำให้ผมเขียนออกมาระบบ Code ได้ดังนี้ครับ

โค้ดโปรแกรม SmartLightIndicator บน GitHub

ก็ให้ทำการนำโค้ดไปวางใน Arduino นะครับ และให้ทำการเปลี่ยนตรง

APPID KEY และ SECRET ให้เป็นข้อมูลของเราเองครับ

ใส่ข้อมูล

```

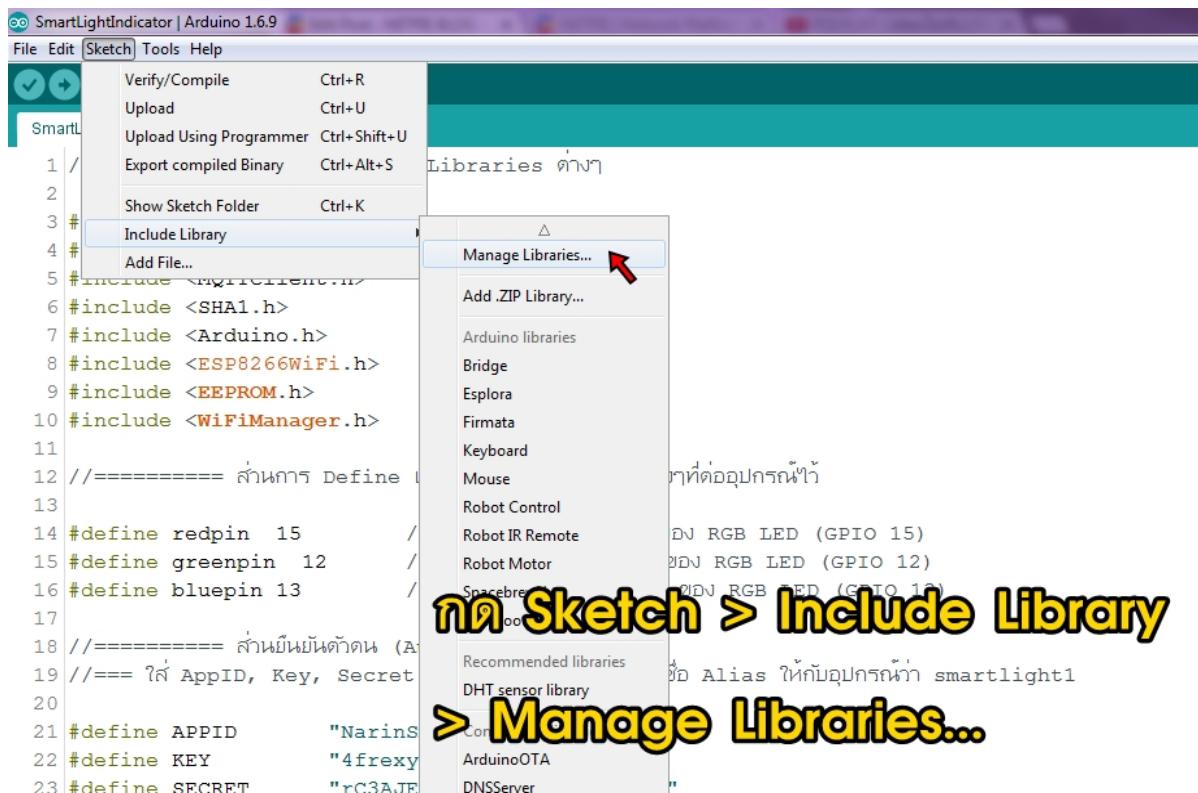
1 //===== ស່ານການ Include Libraries ດ້ວຍ
2
3 #include <AuthClient.h>
4 #include <MicroGear.h>
5 #include <MQTTClient.h>
6 #include <SHA1.h>
7 #include <Arduino.h>
8 #include <ESP8266WiFi.h>
9 #include <EEPROM.h>
10 #include <WiFiManager.h>
11
12 //===== ស່ານການ Define ເພື່ອກຳທະດໜາ (Pins) ດ້ວຍທີ່ດ້ວຍບຸກຮັກໄວ້
13
14 #define redpin 15      //==== ຂາດົມຄຸມ ສີແລງ 20J RGB LED (GPIO 15)
15 #define greenpin 12    //==== ຂາດົມຄຸມ ສີເຊິ້ນ 20J RGB LED (GPIO 12)
16 #define bluepin 13     //==== ຂາດົມຄຸມ ສີໜ້າເປັນ 20J RGB LED (GPIO 13)
17
18 //===== គ່ານມື່ນມື່ນຕົກດານ (Authentication)
19 //== ໃໝ່ AppID, Key, Secret ທີ່ທັງໝົດ NETPIE ແລະທັງໝົດ Alias ໃຫ້ບຸກຮັກໄວ້ smartlight1
20
21 #define APPID        "NarinSmartHome"
22 #define KEY          "4frexyWAZv1MWjw"
23 #define SECRET       "rc3AJEtxhQHe2OGxY7mdGNAGl"
24 #define ALIAS        "smartlight1"
25
26 //===== ស່ານການປະກາດຕົວແປງດ້ວຍ
27
28 int lightlevel;           //==== ດ້ວຍປະກາດຫັນເກີບດໍາຄາມສ່າງຂອງແສງ ດະ ປັບປຸນ
29
30 int lightlevel_high;     //==== ທີ່ແປງສ່າຫະບັນບັດກຳ (ຈຳ) ທ່ານການສ່າງສູງສຸດ
31 int lightlevel_low;       //==== ທີ່ແປງສ່າຫະບັນບັດກຳ (ຈຳ) ທ່ານການສ່າງຕໍ່າສຸດ

```

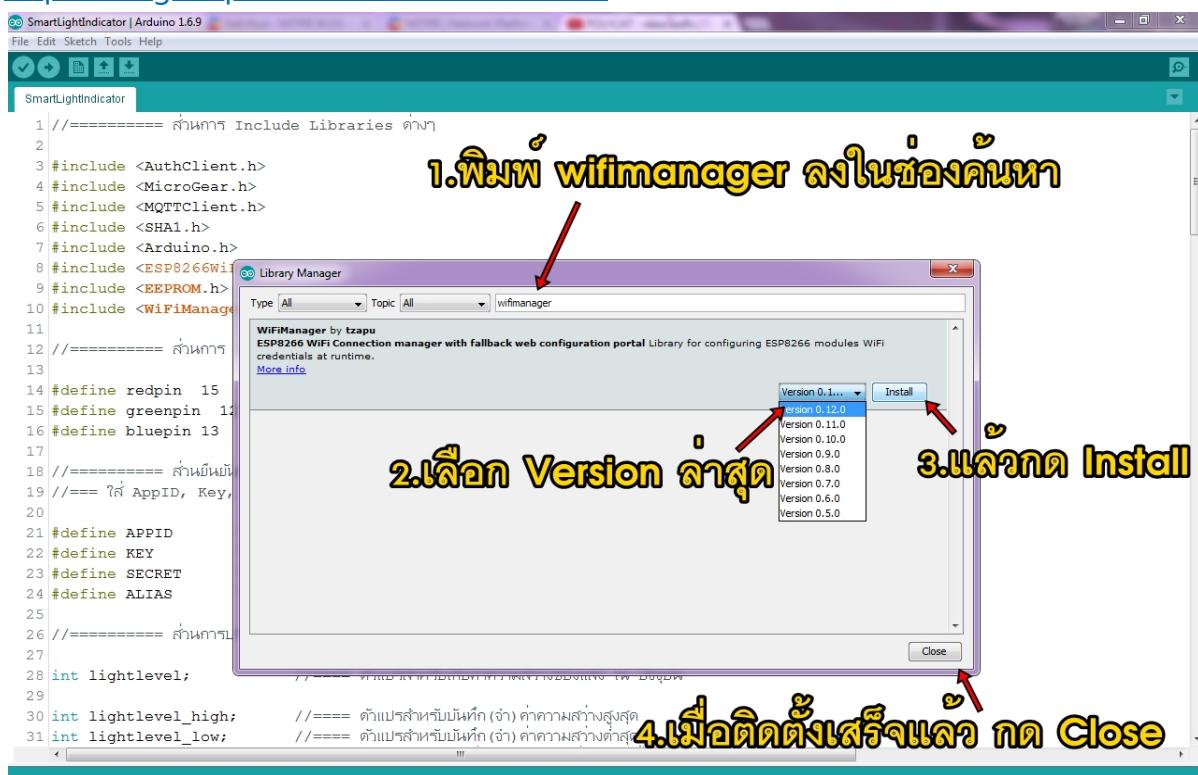
NodeMCU 1.0 (ESP-12E Module), 80 MHz, 921600, 4M (3M SPIFFS) en COM22
TH 16:16
14/7/2559

<https://blog.netpie.io/archives/570/cats-19>

ແລະໃນໂຄດນີ້ ຈະມີການໃໝ່ໄລບຽບ WiFiManager ນະຄຽບ ສໍາຫຼັບໄຣທີ່ຍັງໄມ່ໄດ້ຕິດຕັ້ງ ກໍໃຫ້ທໍາຕາມ
ນີ້ເລີຍນະຄຽບ



<https://blog.netpie.io/archives/570/cats-20>



<https://blog.netpie.io/archives/570/cats-22>

สำหรับใครที่ยังไม่ได้เตรียมซอฟท์แวร์ในการเขียนโปรแกรมให้ ESP8266 ด้วย Arduino
เชิญที่บบทความนี้เลยครับ >>> [Smart Home Part 2](#)

ครับ เมื่อทุกอย่างเรียบร้อยแล้วก็ให้ลองกด Compile ดูรับว่ามีข้อผิดพลาดในการ

แก้ไขโปรแกรมหรือเปล่า หากไม่มีก็ให้ทำการอัพโหลดโคด์ได้เลยครับผม

STEP 4 : การ Config WiFi Manager

เดียวที่นี่พอโคด์เราอัพลงบอร์ดไปเรียบร้อยแล้ว บอร์ดก็จะทำงานขึ้นมา
พร้อมกับไฟ LED สีน้ำเงินติดค้างครับ (หมายถึง บอร์ดรอการ Config / รอการเชื่อมต่อ WiFi อยู่
ครับ)

ชิ้งตอนนี้ บอร์ดของเรากำลังปล่อยสัญญาณ WiFi (ทำตัวเป็น Access Point) ตัวหนึ่งอยู่ครับ
ซึ่งเราต้องทำการนำโทรศัพท์ / หรือคอมพิวเตอร์ ต่อเข้าไป Config ก่อน เช่นตอนนี้เมื่อผมเปิด
หน้า

เชื่อมต่อ WiFi ของโทรศัพท์ผมขึ้นมา ก็จะเห็นรายชื่อเครือข่ายต่างๆครับ
จะเห็นว่า มี SmartLightIndicator ซึ่งเป็น WiFi ที่บอร์ดเรากำลังปล่อยอยู่ด้วย



WiFi



เปิด

เปิด

SmartLightIndicator



BB_C4



Mkittisak



peackcheek



Pacharapeam



TTTB BBB



BB_C99



BB_C5



<https://blog.netpie.io/archives/570/cats-23>

ที่นี่ก็ให้ทำการเชื่อมต่อไปยัง SmartLightIndicator เลยครับ



<https://blog.netpie.io/archives/570/cats-24>

จากนั้น ก็จะเด้งมาที่หน้านี้ครับ ก็ให้ทำการกด Configure WiFi



ลงชื่อเข้าใช้เครือข่าย

ยกเลิก

SmartLightIndicator

WiFiManager

Configure WiFi

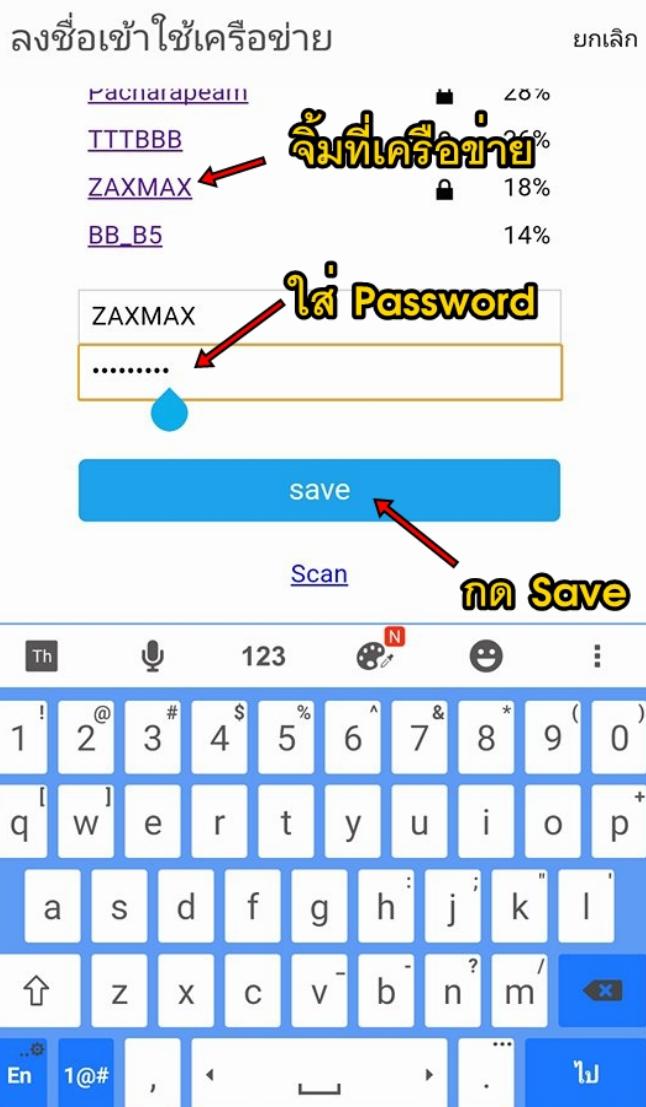
Configure WiFi (No Scan)

Info

Reset

<https://blog.netpie.io/archives/570/cats-25>

จากนั้นก็จะเห็นรายชื่อเครือข่าย WiFi ที่สามารถต่อได้ในตอนนี้
ก็ให้ทำการกดที่ชื่อเครือข่าย และพิมพ์ Password ลงไป และกด Save ครับ



<https://blog.netpie.io/archives/570/cats-26>

จากนั้นก็จะเด้งมาหน้านี้ ก็จะหมายความว่าตอนนี้อุปกรณ์ของเรารับบันทึกค่าไว้เรียบร้อยแล้วครับ



ลงชื่อเข้าใช้เครือข่าย

ยกเลิก

Credentials Saved

Trying to connect ESP to network.

If it fails reconnect to AP to try again

<https://blog.netpie.io/archives/570/cats-27>

ก็ให้รออีกซักพักอุปกรณ์จะไปต่อ WiFi ตามที่เราได้ Config ไปเมื่อขั้นตอนนี้ครับ
โดยเมื่ออุปกรณ์ต่อ WiFi ได้สำเร็จแล้ว ก็จะขึ้นไฟสีเขียวแบบนี้ครับ



หลังจากขึ้นไฟสีเขียว ชักพักนึงก็จะเห็นไฟสีเขียวบนกระปิบ และดับไป 3 วินาที นั่นเป็น สัญญาณ

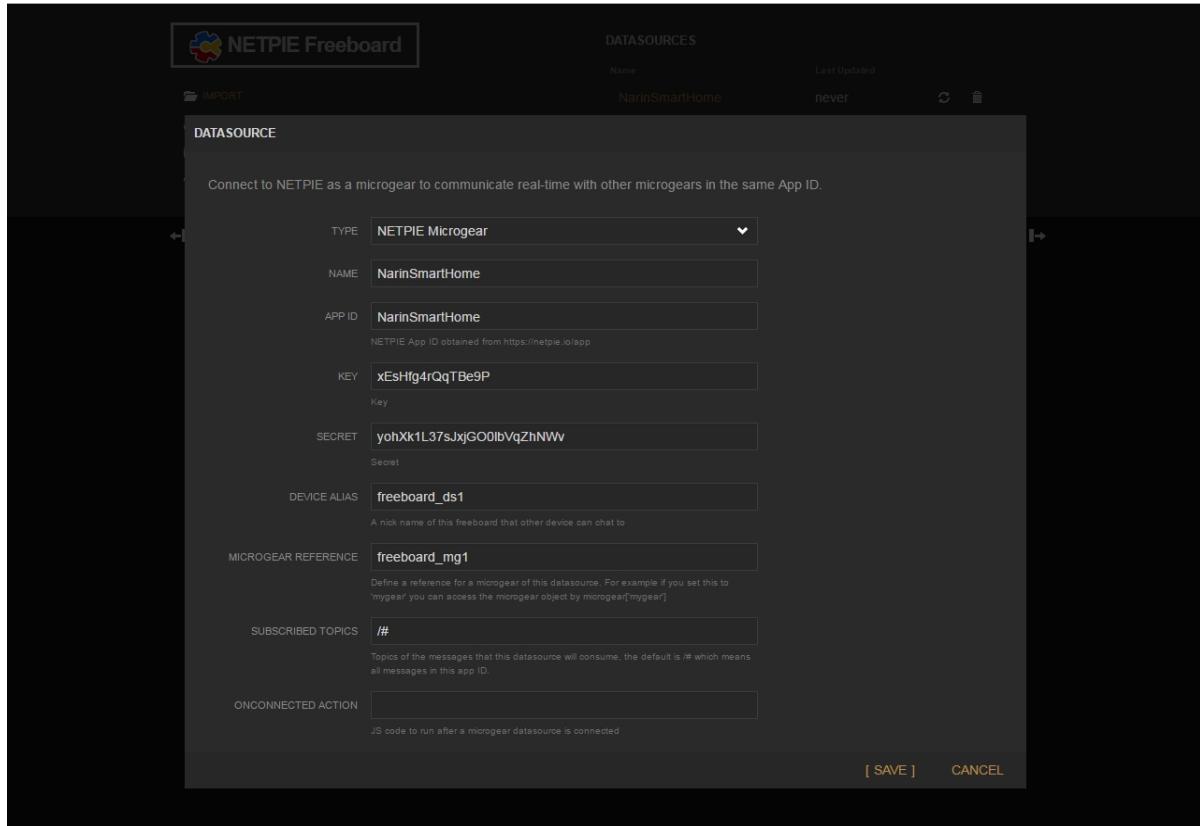
ที่บ่งบอกว่า ตอนนี้อุปกรณ์ของเราราได้ต่อเข้ากับ NETPIE เรียบร้อย และเริ่มทำงานตามโปรแกรม
แล้วครับ ^_^

STEP 5 : ที่นี่มาดูบัน Freeboard กันบ้าง...

ครับ หลังจากที่อุปกรณ์เราทำงานแล้ว ต่อไปก็จะเป็นการสร้างตัวแสดงผลข้อมูล
ด้วย NETPIE Freeboard กันนะครับ สำหรับบทความพื้นฐานการใช้งาน NETPIE Freeboard
ผมได้เขียนไว้แล้วนะครับ สำหรับใครที่ยังไม่เคยทำก็ให้ไปอ่านได้ที่นี่ครับ >> [NETPIE](#)

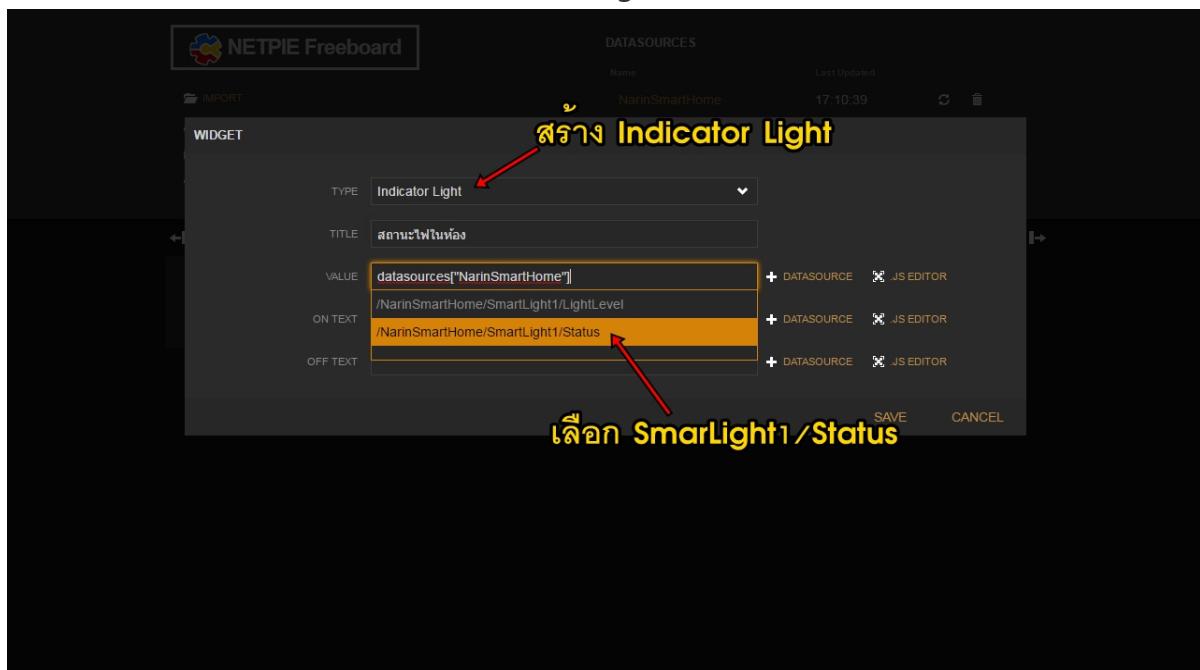
[Freeboard Part 1](#)

ครับผม ขั้นแรกก็ให้เปิด Freeboard ขึ้นมาครับ และทำการสร้าง Datasource ให้เรียบร้อย

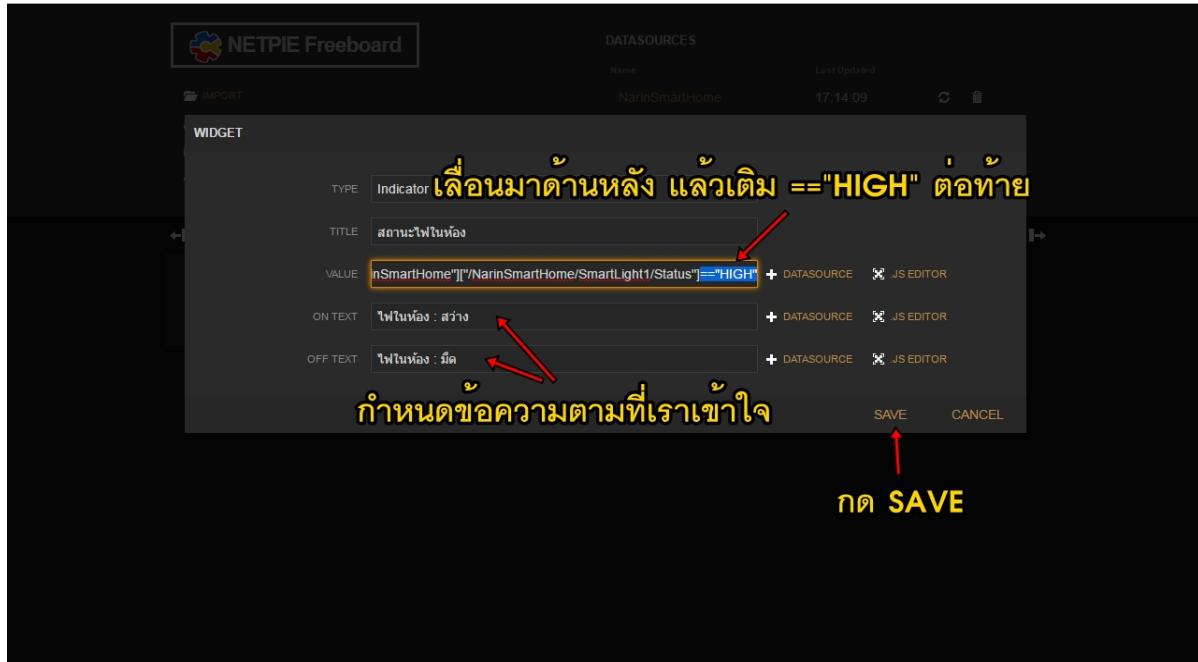


<https://blog.netpie.io/archives/570/cats-28>

จากนั้นผูกจะทำการ Add Pane และเพิ่ม Widget ลงไป
โดยอันแรกจะเป็น Indicator Light นะครับ ก็ให้ทำตามนี้เลย

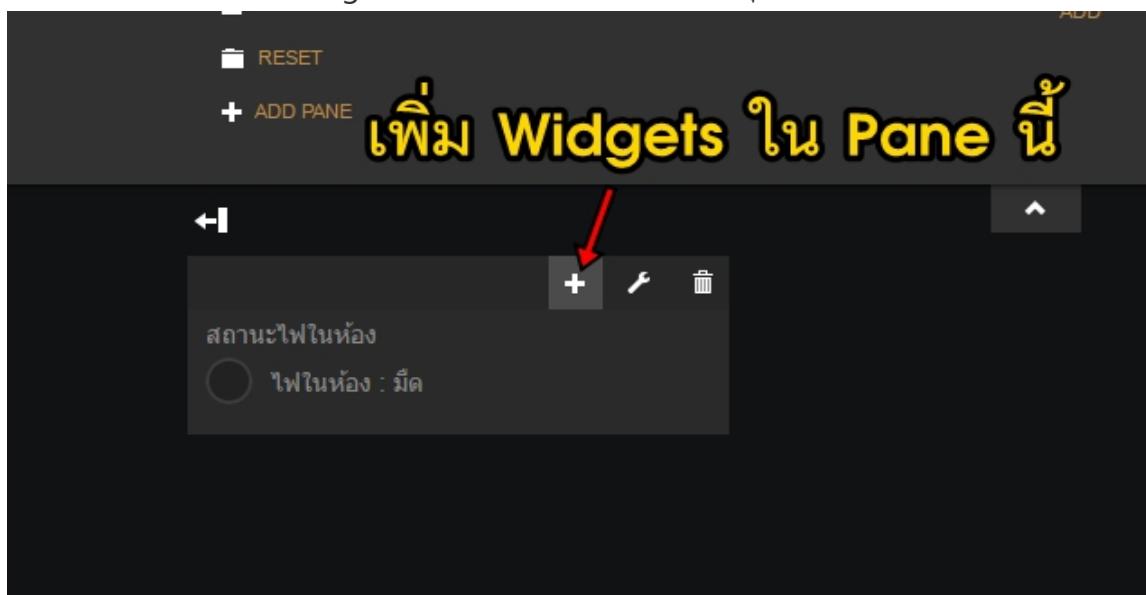


<https://blog.netpie.io/archives/570/cats-29>

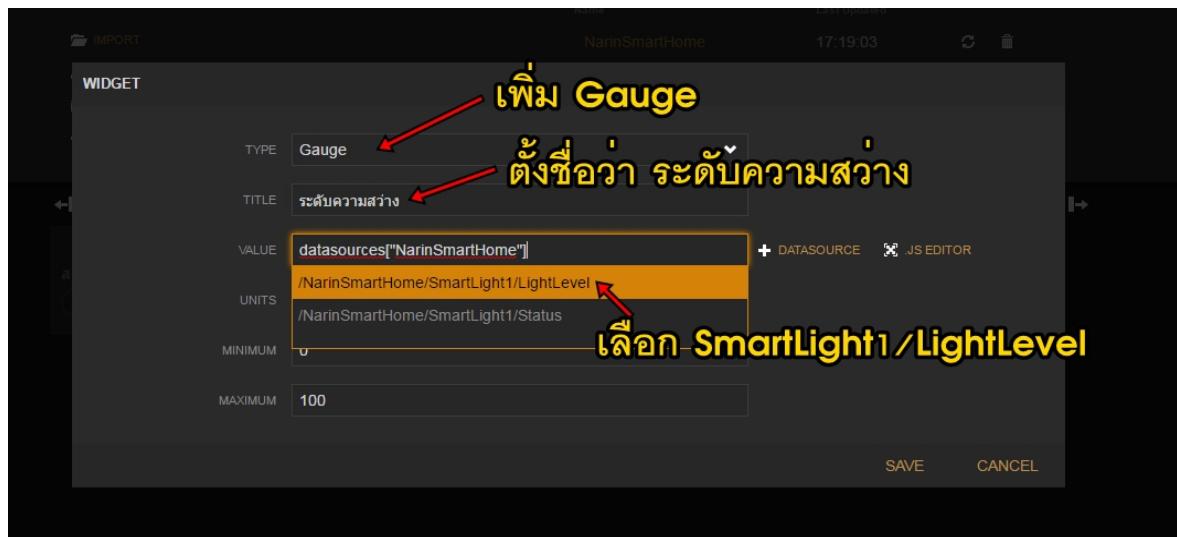


<https://blog.netpie.io/archives/570/cats-30>

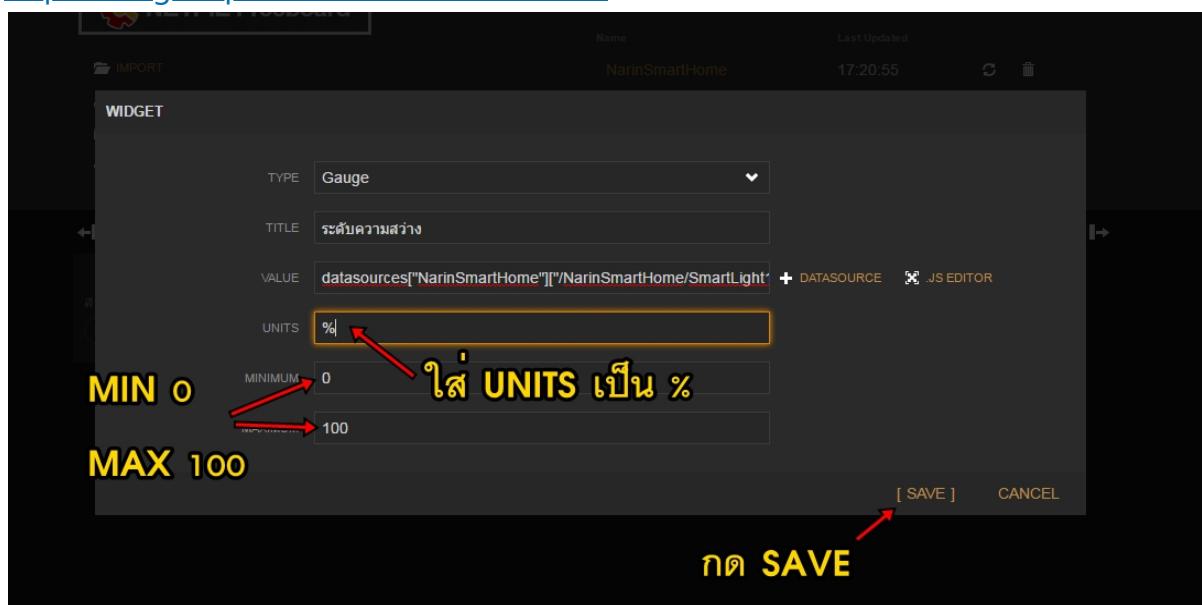
จากนั้นก็เพิ่มอันต่อๆไปครับ แต่ผมจะไม่สร้าง PANE ใหม่ จะเพิ่มลงใน PANE เดียวกันนี้เลย โดยต่อไปผมจะเพิ่ม Gauge เพื่อบอกระดับความสว่างปัจจุบันเป็น % ลงไปครับ ทำตามนี้เลย



<https://blog.netpie.io/archives/570/cats-31>

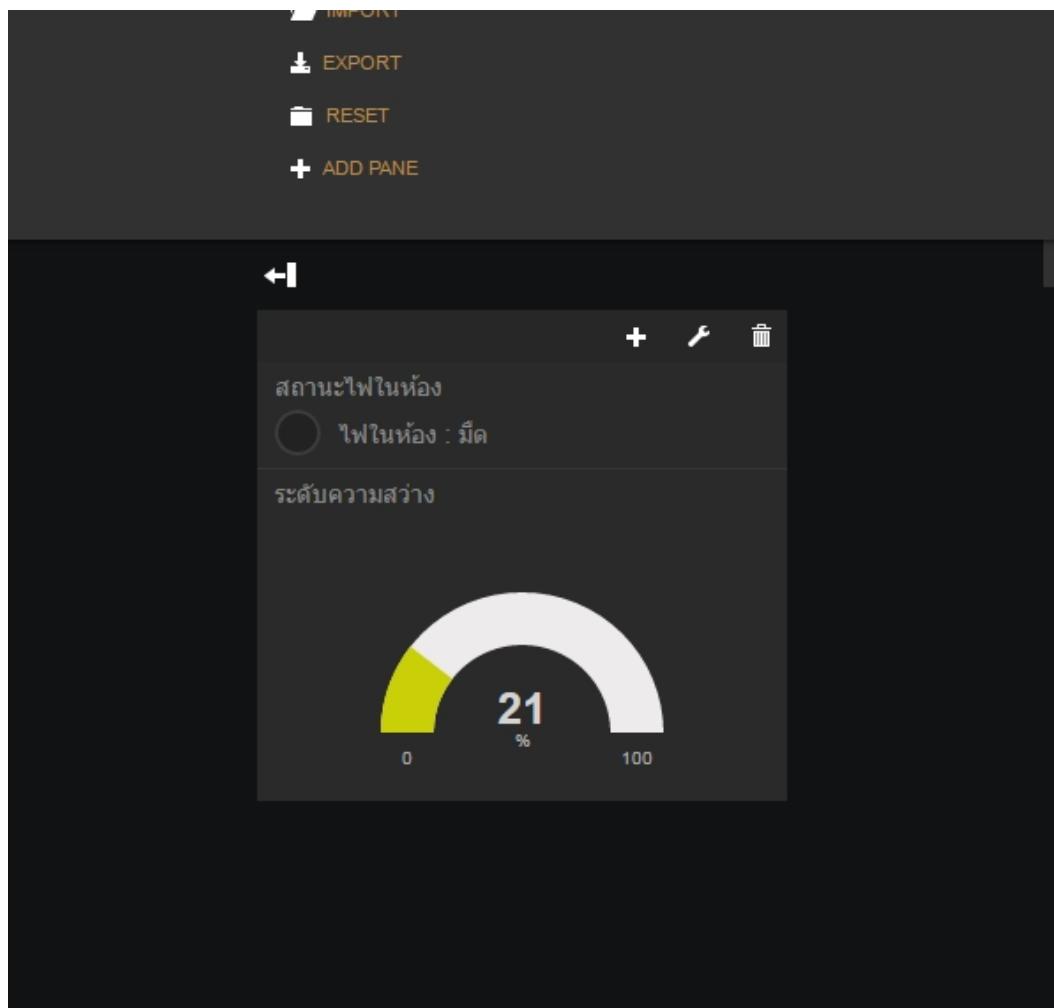


<https://blog.netpie.io/archives/570/cats-32>



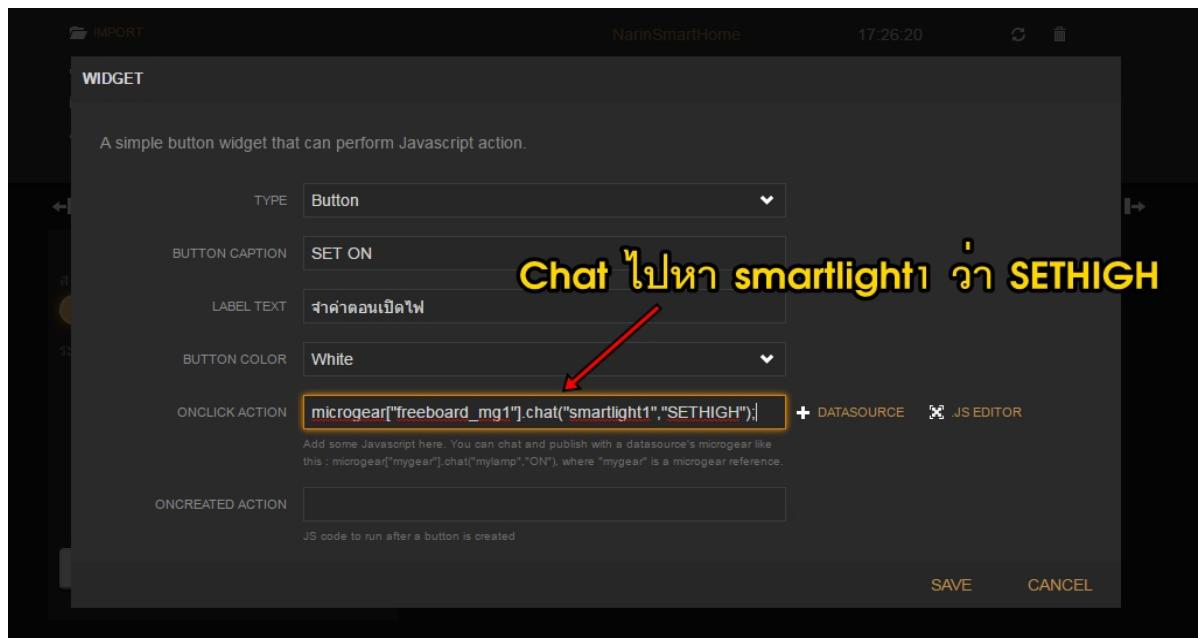
<https://blog.netpie.io/archives/570/cats-33>

ที่นี่เราจะได้ Gauge สำหรับแสดงค่าความสว่างมาแล้วครับ



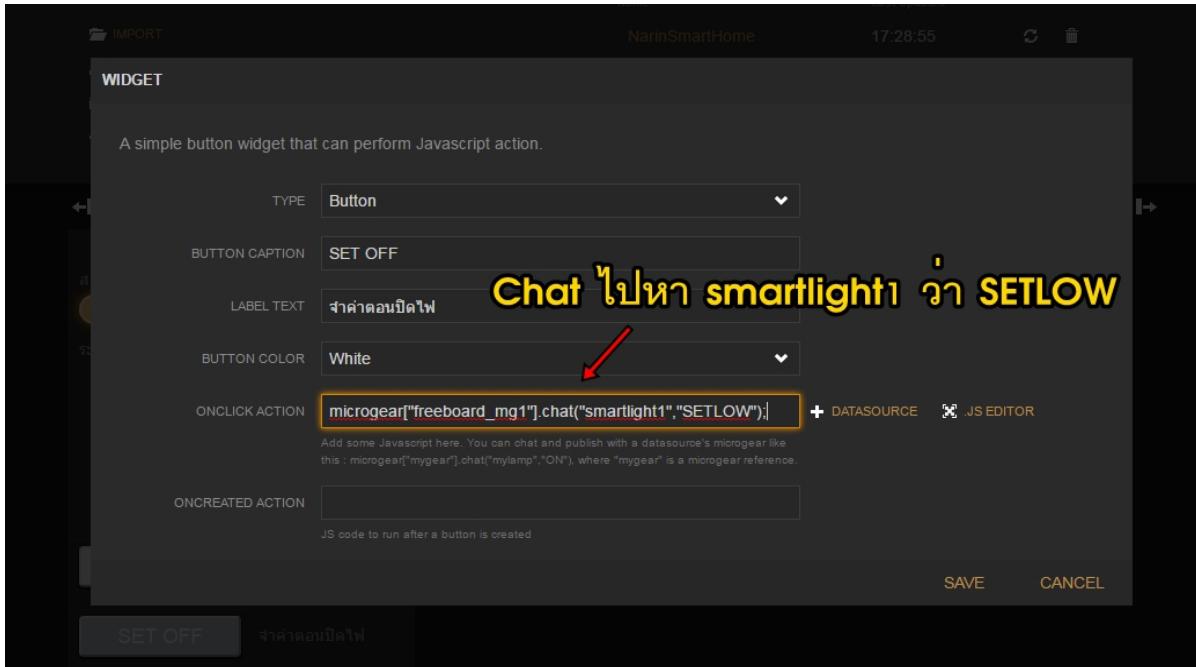
<https://blog.netpie.io/archives/570/cats-34>

ต่อไปก็จะเป็นการสร้างปุ่มจำค่าความสว่างสูงสุดต่ำสุดกันนะครับ
ก็ให้สร้างปุ่มตามนี้เลยครับ เริ่มจาก ปุ่มจำค่าตอนเปิดไฟนะครับ



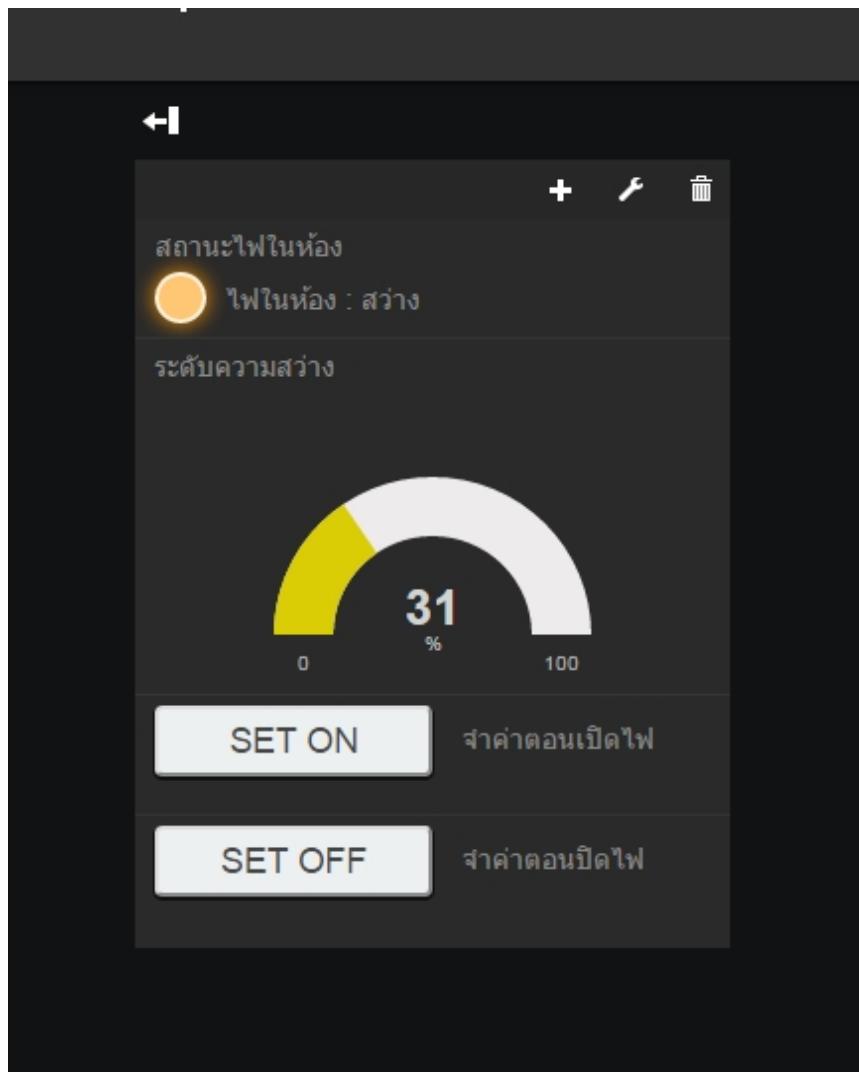
<https://blog.netpie.io/archives/570/cats-35>

ตามด้วยปุ่มจำค่าตอนปิดไฟครับ



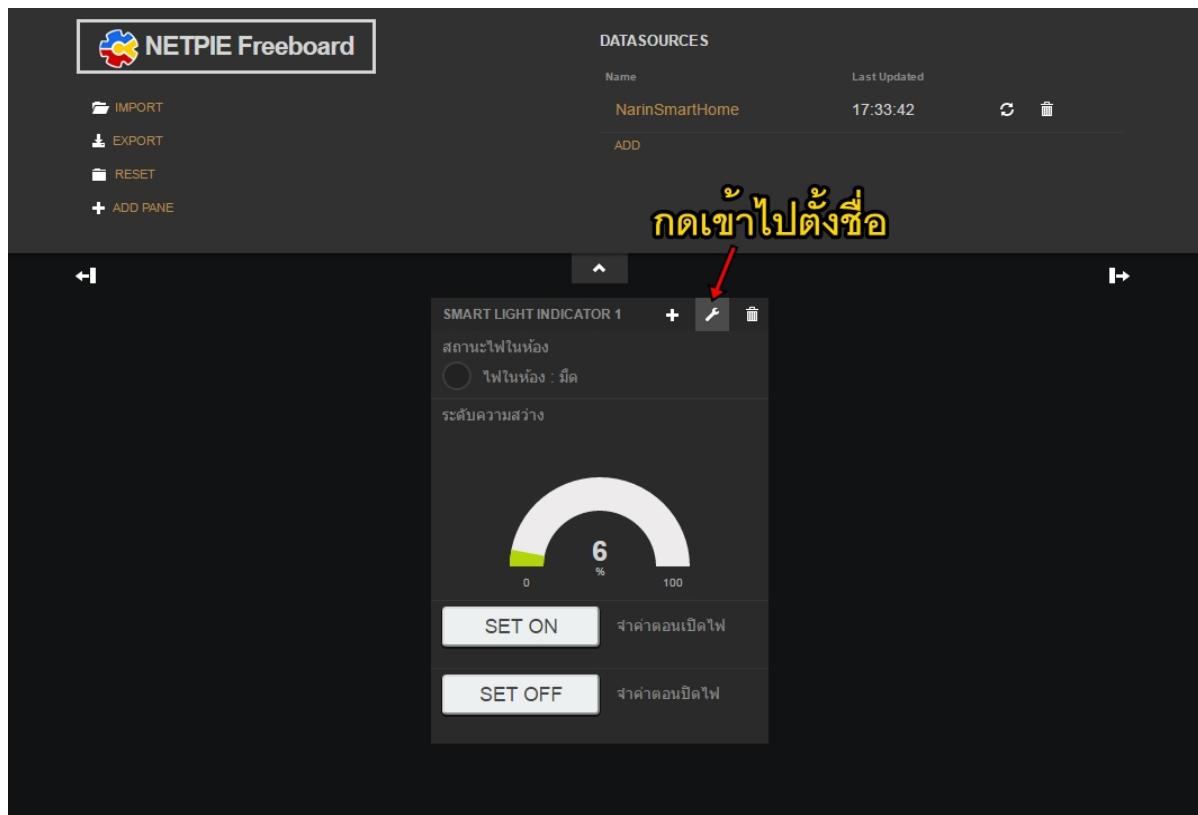
<https://blog.netpie.io/archives/570/cats-36>

ที่นี่ก็จะได้ครบแล้วครับ สำหรับ PANE ของ Smart Light Indicator



<https://blog.netpie.io/archives/570/cats-37>

ที่นี่ก็ ตั้งชื่อ Pane ใหม่ และย้ายไปวางตรงกลาง ก็จะสวยงามแล้วครับ ^_^



<https://blog.netpie.io/archives/570/cats-38>

แคนน์ก์เสร็จแล้วรับ กับ Smart Light Indicator พร้อม Freeboard ควบคุม

STEP 6 : รีวิวการทำงานของ Smart Light Indicator 1