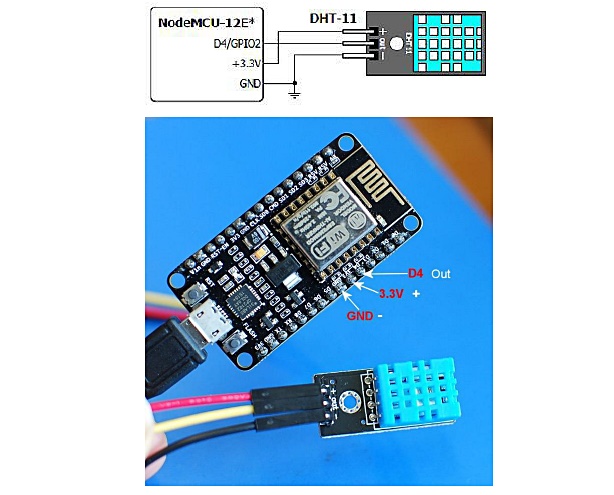
**Lab 1**

**การแสดงผลข้อมูลจากอุปกรณ์เซนเซอร์บน NETPIE Freeboard**

Lab 1 เป็นการสร้างแอปพลิเคชั่นเซนเซอร์วัดอุณหภูมิและความชื้นด้วย NodeMCU และเซนเซอร์โมดูล DHT11 (อุปกรณ์ ZX-DHT11 ใน IoT Kit) ในขั้นแรกผู้ใช้ต้องติดตั้ง Library ที่เกี่ยวข้องกับเซนเซอร์ก่อน ซึ่งในที่นี้คือ DHT.h ซึ่งสามารถดาวน์โหลดได้ที่ <https://github.com/adafruit/DHT-sensor-library> ให้ผู้ใช้งานคัดลอกไปวางไว้ใน Folder Arduino\libraries ในภาพด้านล่างแสดงการเชื่อมต่อ เซนเซอร์ DHT11 กับ NodeMCU



หลังจากเชื่อมต่อเสร็จ ให้ทาตามขั้นตอนดังนี้

1.ฝั่ง NodeMCU สามารถเขียนโค้ดสร้างไฟล์ piedht.ino ตามข้างล่างและ Upload ไฟล์เข้า NodeMCU

piedht.ino

#include <DHT.h>

#include <ESP8266WiFi.h>

#include <MicroGear.h>

const char\* ssid = "SSID";

const char\* password = "PASSWORD";

#define APPID "YOUR\_APPID"

#define KEY "YOUR\_KEY"

#define SECRET "YOUR\_SECRET"

#define ALIAS "piedht"

WiFiClient client;

int timer = 0;

char str[32];

#define DHTTYPE DHT11 //Define sensor type

#define DHTPIN D4 // Define sensor pin

DHT dht(DHTPIN, DHTTYPE, 15); //Initialize DHT sensor

int humid;

int temp;

MicroGear microgear(client);

void onMsghandler(char \*topic, uint8\_t\* msg, unsigned int msglen) {

Serial.print("Incoming message -->");

msg[msglen] = '\0';

Serial.println((char \*)msg);

}

void onConnected(char \*attribute, uint8\_t\* msg, unsigned int msglen) {

Serial.println("Connected to NETPIE...");

microgear.setAlias(ALIAS);

}

void setup(){

dht.begin();

microgear.on(MESSAGE,onMsghandler);

microgear.on(CONNECTED,onConnected);

Serial.begin(115200);

Serial.println("Starting...");

if (WiFi.begin(ssid, password)) {

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

}

Serial.println("WiFi connected");

Serial.println("IP address: ");

Serial.println(WiFi.localIP());

microgear.init(KEY,SECRET,ALIAS);

microgear.connect(APPID);

}

void loop(){

if (microgear.connected()) {

Serial.println("connected");

microgear.loop();

if (timer >= 1000) {

humid = dht.readHumidity();

temp = dht.readTemperature();

sprintf(str,"%d,%d",humid,temp);

Serial.println(str);

Serial.print("Sending -->");

microgear.publish("/dht",str);

timer = 0;

}

else timer += 100;

}

else {

Serial.println("connection lost, reconnect...");

if (timer >= 5000) {

microgear.connect(APPID);

timer = 0;

}

else timer += 100;

}

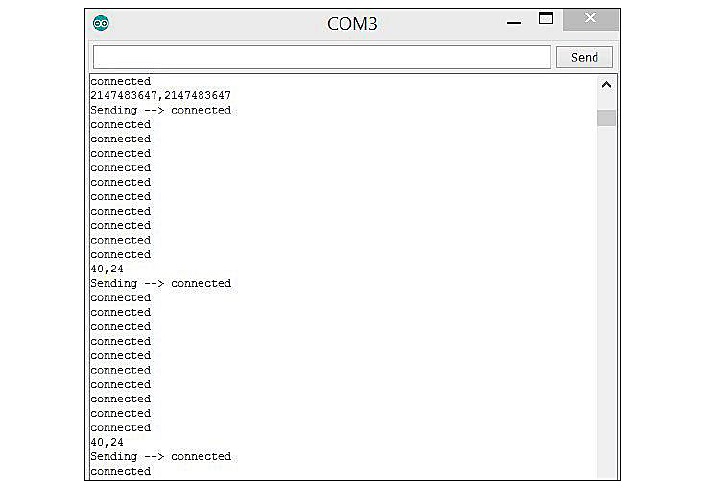
delay(100);

}

ในตัวอย่างนี้ คำสั่ง microgear.publish(“/dht”,str) คือการ Publish ข้อความ (ในที่นี้เป็น Message String str) ไปยัง Topic ที่ระบุคือ /dht ซึ่งข้อความคือค่าของอุณหภูมิและความชื้นที่วัดได้จากเซนเซอร์ DHT

2.เปิด Console โดยไปที่ Tools -> Serial Monitor เพื่อตรวจสอบว่าเซนเซอร์ทางานได้ปกติหรือไม่

รูปแสดงคอนโซลตรวจสอบการทางานของเซนเซอร์



3.ฝั่ง NETPIE Freeboard คลิกที่ Datasource ที่สร้างขึ้นก่อนหน้านี้ และแก้ไขในช่อง SUBSCRIBED TOPICS ให้เป็น /dht

4.กด + (ADD PANE) เพื่อสร้าง Widget ชนิด Gauge ใหม่ 2 Widget เพื่อแยกแสดงอุณหภูมิและความชื้น โดยกรอกตั้งค่าแต่ละ Widget ดังนี้

Widget 1:

TITLE : Humidity

VALUE : datasources["YourDatasourceName"]["/YourAppID/dht"].split(",")[0]

UNIT : %

MINIMUM : 0

MAXIMUM : 100

Widget 2:

TITLE : Temperature

VALUE : datasources["YourDatasourceName"]["/YourAppID/dht"].split(",")[1]

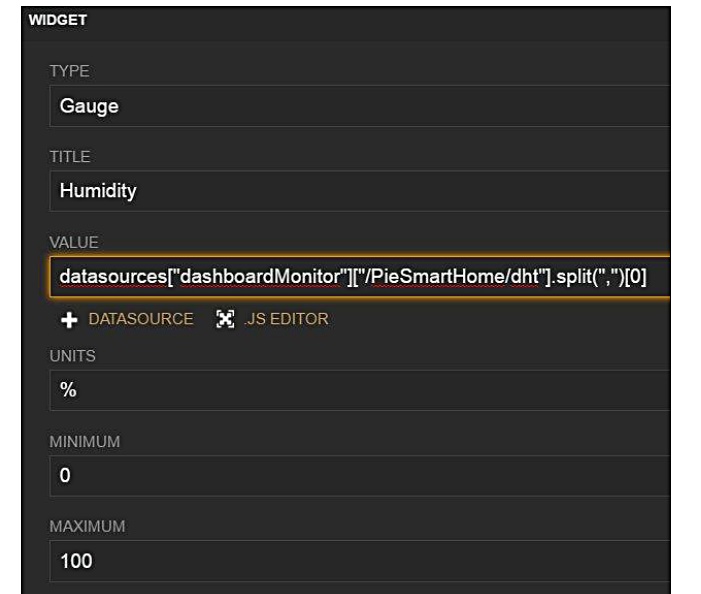
UNIT : C

MINIMUM : 0

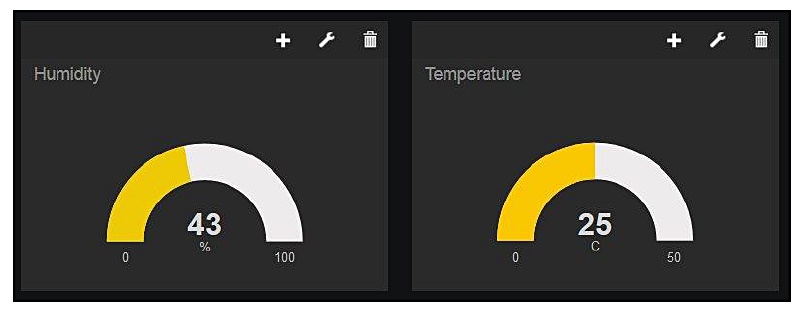
MAXIMUM : 50

ภาพด้านล่างแสดงตัวอย่างการกรอกข้อมูลเพื่อสร้าง Widget แสดงค่าความชื้น (Humidity)

รูปแสดงตัวอย่างหน้าจอตั้งค่า Widget แสดงค่าความชื้น



รูปแสดง Widget แสดงค่าอุณหภูมิและความชื้นที่วัดได้จากเซนเซอร์



คำอธิบายเพิ่มเติม

เนื่องจาก DHT ส่งค่ามาในรูปแบบ "Humidity, Temperature" เช่น " 40,24 " เวลาเรารับค่าเข้ามา จึงต้องทาการแยกออกเป็น Array โดยใช้เครื่องหมาย Comma “,” เป็นตัวแบ่ง จากนั้นก็อ้างอิงถึงช่องใน Array ของ Message String ที่ Publish ใน Topic /dht เช่น Index \[0\] หมายถึงค่าแรก และ Index \[1\] หมายถึงค่าถัดมา

**Lab 5.2**

**การควบคุมอุปกรณ์ด้วย NETPIE Freeboard**

Lab 5.2 แสดงการประยุกต์ NETPIE Freeboard ในการควบคุมอุปกรณ์ โดยในเบื้องต้นนี้ เราจะควบคุมไฟ LED บนบอร์ด NodeMCU ซึ่งใช้หลักการทางานบนพื้นฐานของการ Subscribe ข้อความจาก Topic หรือหัวข้อที่ระบุ และการกาหนดตรรกะของการควบคุมทั้งในส่วนของ Datasource และส่วนของ Widget ที่ใช้ควบคุม โดยมีขั้นตอนดังต่อไปนี้

1.แก้ไขไฟล์ pieled2.ino โดยระบุข้อมูลการเข้าถึงเครือข่าย Wifi ข้อมูล APPID, KEY และ SECRET ตามโค้ดข้างล่าง และทาการ Upload ไฟล์เข้า NodeMCUให้เชื่อมต่อกับ NETPIE pieled2.ino

#include <ESP8266WiFi.h>

#include <MicroGear.h>

const char\* ssid = "SSID";

const char\* password = "PASSWORD";

#define APPID "YOUR\_APPID"

#define KEY "YOUR\_KEY"

#define SECRET "YOUR\_SECRET"

#define ALIAS "pieled"

WiFiClient client;

char state = 0;

char stateOutdated = 0;

char buff[16];

MicroGear microgear(client);

void sendState(){

if (state==0)

microgear.publish("/pieled/state","0");

else

microgear.publish("/pieled/state","1");

Serial.println("send state..");

stateOutdated = 0;

}

void updateIO(){

if (state >= 1) {

digitalWrite(LED\_BUILTIN, LOW);

}

else {

state = 0;

digitalWrite(LED\_BUILTIN, HIGH);

}

}

void onMsghandler(char \*topic, uint8\_t\* msg, unsigned int msglen) {

char m = \*(char \*)msg;

Serial.print("Incoming message -->");

msg[msglen] = '\0';

Serial.println((char \*)msg);

if (m == '0' || m == '1') {

state = m=='0'?0:1;

updateIO();

}

if (m == '0' || m == '1' || m == '?') stateOutdated = 1;

}

void onConnected(char \*attribute, uint8\_t\* msg, unsigned int msglen) {

Serial.println("Connected to NETPIE...");

microgear.setAlias(ALIAS);

stateOutdated = 1;

}

void setup(){

Serial.begin(115200);

Serial.println("Starting...");

pinMode(LED\_BUILTIN, OUTPUT);

if (WiFi.begin(ssid, password)) {

while (WiFi.status() != WL\_CONNECTED) {

delay(500);

Serial.print(".");

}

}

microgear.on(MESSAGE,onMsghandler);

microgear.on(CONNECTED,onConnected);

microgear.init(KEY,SECRET,ALIAS);

microgear.connect(APPID);

}

void loop(){

if (microgear.connected()) {

if (stateOutdated) sendState();

microgear.loop();

}

else {

Serial.println("connection lost, reconnect...");

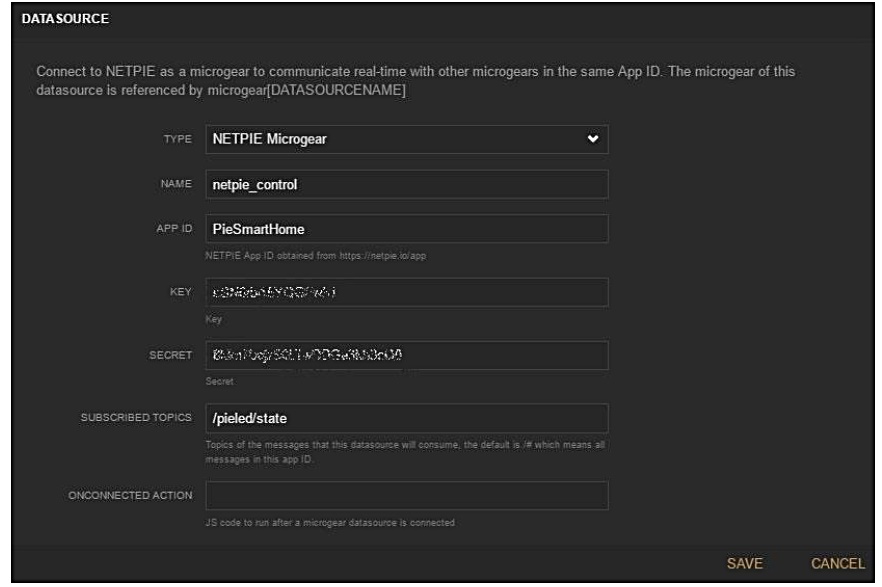
microgear.connect(APPID);

}

}

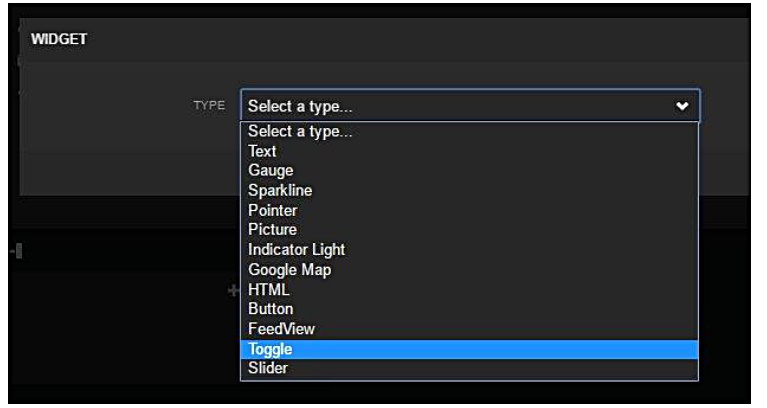
2.ในหน้า NETPIE Freeboard คลิกเพิ่ม Datasource ที่สร้างขึ้นก่อนหน้านี้เพื่อแก้ไข ตั้งชื่อ Datasource ใส่ค่า KEY และ SECRET และในช่อง SUBSCRIBED TOPICS ให้ใส่ /pieled/state หรือ Topic ที่ท่านระบุไว้สาหรับการ publish ในไฟล์ pieled2.ino ดังแสดงในภาพด้านล่าง และกด Save

รูปแสดงหน้าต่างตั้งค่า Datasource ใน NETPIE Freeboard สาหรับควบคุม LED



3.สร้าง Widget ขึ้นมาใหม่โดยกด + (ADD PANE) และเลือกชนิดใน Drop Down Box เป็นแบบ Toggle

รูปแสดงหน้าจอเลือกชนิด Widget ให้เป็นแบบ Toggle



จากนั้นตั้งค่า Widget ดังนี้โดยหน้าจอการตั้งค่าแสดงดังรูป

TOGGLE CAPTION : ตั้งชื่อปุ่ม Toggle (ในตัวอย่างตั้งเป็น PIE\_LED)

TOGGLE STATE : ใส่ข้อมูลตามชื่อของ Datasource และ Topic เช่น  
datasources["netpie\_control"]["/PieSmartHome/pieled/state"]==1

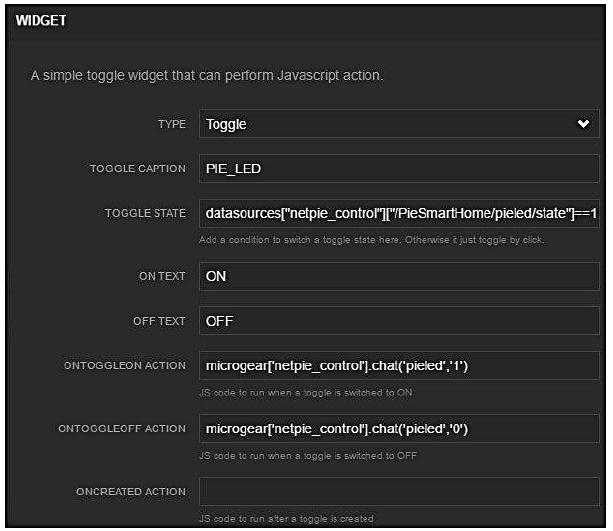
ON TEXT : ON

OFF TEXT : OFF

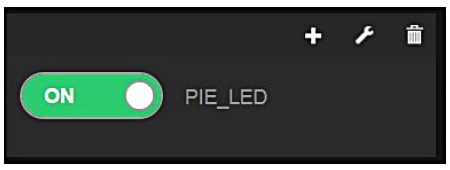
ONTOGGLEON ACTION : microgear['netpie\_control'].chat('pieled','1')

ONTOGGLEOFF ACTION : microgear['netpie\_control'].chat('pieled','0')

รูปแสดงหน้าจอการตั้งค่า Widget ชนิด Toggle



แล้วกด Save จะได้ Widget ที่มีปุ่มควบคุมดังแสดงในภาพ เพื่อทดสอบเปิดปิด LED บน NodeMCU



คำอธิบายเพิ่มเติม

+

TOGGLE STATE เป็นสถานะ On/Off ซึ่งสามารถผูกกับตรรกะของ Datasource ในที่นี่เราตั้งค่าให้ Toggle เปลี่ยนสถานะตามค่าที่ส่งมาใน Topic ชื่อ /pieled/state

ONTOGGLEON และ ONTOGGLEOFF เป็นคาสั่งที่จะถูกเรียก เมื่อ Toggle เปลี่ยนสถานะไปเป็น ON และ OFF ตามลาดับ

ในหน้า Datasources ตรงช่อง SUBSCRIBED TOPICS นั้น นอกจากจะสามารถระบุค่าแบบเจาะจงเป็น topic /pieled/state แล้ว ยังสามารถระบุค่าเป็น /pieled/+ ก็ได้ โดยใช้เครื่องหมาย (+) ซึ่งเป็น Single-Level Wildcard เพื่อรับค่าของ State

| เราสามารถใช้ Wildcard เพื่อช่วยในการ Subscribe Topic ต่างๆ เช่น หากต้องการ Subscribe Topic ตามที่ระบุแบบเจาะจงดังนี้: "/home/kitchen/temp", "/home/bedroom/temp", และ "/home/livingroom/temp" ก็สามารถยุบเหลือ 1 Topic คือ "/home/+/temp" นอกจาก + แล้วยังใช้เครื่องหมาย # ได้ด้วย โดยที่เครื่องหมาย + จะแทนคาอะไรก็ได้ระดับชั้นเดียว ส่วน # จะแทนคาอะไรก็ได้ในระดับชั้นยาวต่อไปเท่าไหร่ก็ได้ เช่นจะให้ match 3 Topic ข้างต้น ก็อาจจะเขียนระบุเป็น Topic คือ "/home/#" |
| --- |