



INTERNET OF THING

WITH NETPIE



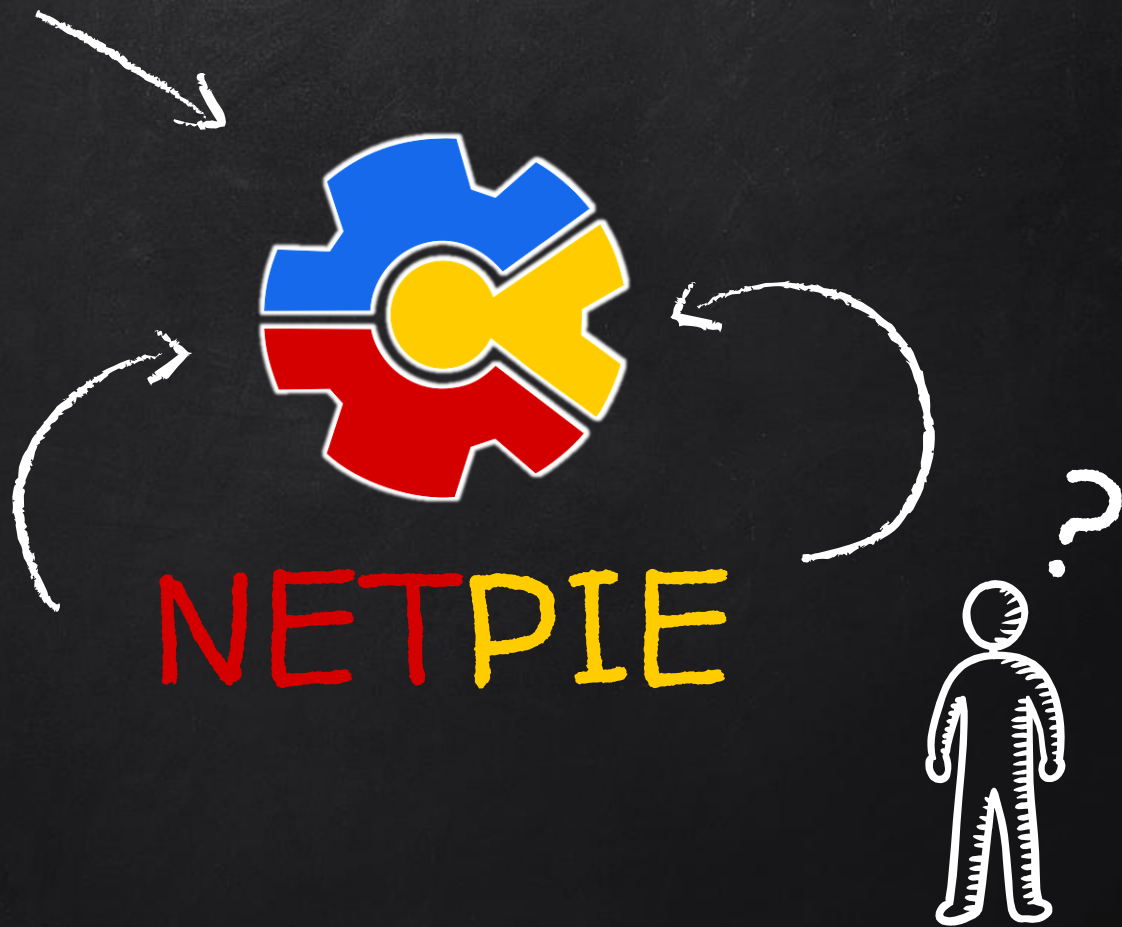
Anun Panya

Research Assistant

anun.panya@nectec.or.th

Internet Innovation Laboratory

National Electronics and Computer Technology Center (NECTEC)





CLOUD PLATFORM

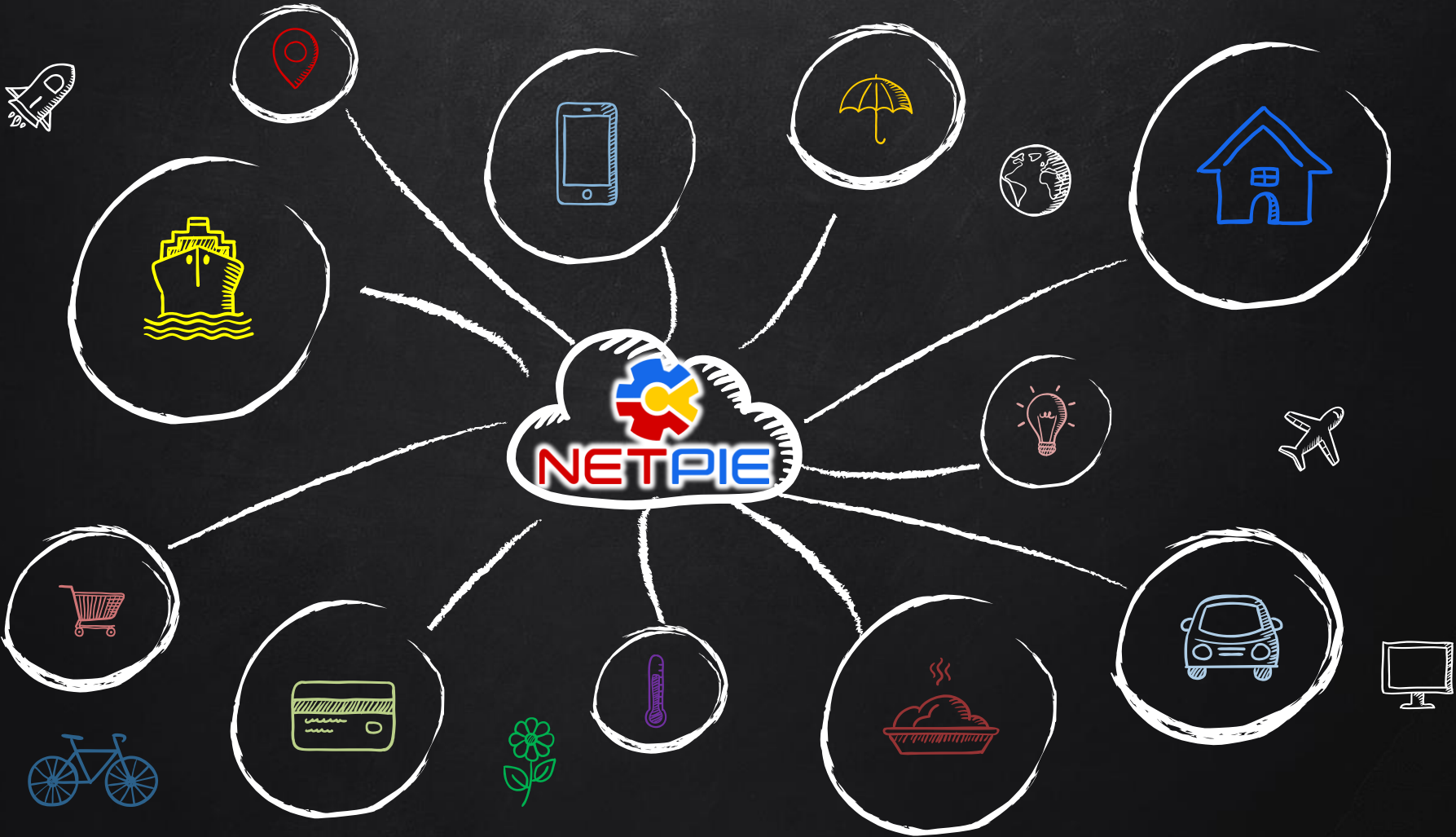
+



MICROGEAR LIBRARY

=

NETPIE





BENEFITS OF NETPIE



REDUCES THE USE OF CONNECTION RESOURCES



REDUCES DATA SECURITY BURDEN



FLEXIBLE TO EXTEND SYSTEM





FEATURES OF MICROGEAR LIBRARY



COMMUNICATION



AUTHENTICATION



AUTHORIZATION



COORDINATION



good
job



SUPPORT

PROGRAMING LANGUAGE



{ REST APIs }





SUPPORT

OPERATING SYSTEM & HARDWARE



HOW TO USE MICROGEAR LIBRARY



FUNCTIONS FOR USE



Functions

- create();
- init();
- connect();
- setAlias();
- chat();
- publish();
- subscribe();
- unsubscribe();
- useTLS();
- writeFeed();



Events

- connected
- closed
- error
- message
- present
- **absent**





Function create





เป็นฟังก์ชันสำหรับสร้าง object microgear เพื่อเริ่มต้นใช้งาน (ไม่ใช่สำหรับ Arduino)

Microgear.create(config);

arguments : **config** เป็น json object ที่มี attribute ดังนี้

 **key** – เป็น key สำหรับ device

 **secret** – เป็น secret ของ key ซึ่งจะใช้ประกอบในกระบวนการยืนยันตัวตน

 **alias** – เป็นการตั้งชื่อเรียก จะใส่ที่นี้หรือ setAlias() ที่หลังก็ได้



```
var microgear = MicroGear.create({  
  key : "sXfqDcXHzbFXiLk",  
  secret : "DNonzg2ivwS8ceksykGntrfQjxbL98",  
  alias : "myhtml"  
});
```




Function init




เป็นฟังก์ชัน Initial library (ใช้สำหรับ **Arduino**)

```
microgear.init(key, secret, alias);
```

arguments:

 **key** – เป็น key สำหรับ device

 **secret** – เป็น secret ของ key ซึ่งจะใช้ประกอบในกระบวนการยืนยันตัวตน

 **alias** – เป็นการตั้งชื่อของ device



```
microgear.init("sXfqDcXHzbFXiLk", "DNonzg2ivwS8ceksykGntrfQjxbL98", "myplant");
```



Function connect



เป็นฟังก์ชันเชื่อมต่อไปที่ NETPIE โดยระบุ appid เป้าหมาย

```
microgear.connect(appid);
```

arguments:

 **appid** – คือ application ที่ microgear จะทำการเชื่อมต่อ



```
microgear.connect("happyfarm");
```




Function setAlias



เป็นฟังก์ชันที่ **microgear** สามารถตั้งชื่อตัวเองได้ ซึ่งสามารถใช้เป็นชื่อเรียกในการใช้ฟังก์ชัน `chat()`

```
microgear.setAlias(alias);
```

arguments:

 **alias** – ชื่อของ **microgear** นี้



```
microgear.setAlias("plant");
```



Function chat



เป็นฟังก์ชันสำหรับส่งข้อความแบบเจาะจงผู้รับ

```
microgear.chat(gearalias, message);
```

arguments :

 **gearalias** – ชื่อของ microgear ที่ต้องการจะส่งข้อความไปถึง

 **message** – ข้อความที่ต้องการส่ง



```
microgear.chat("valve","I need water");
```



Function publish



เป็นฟังก์ชันสำหรับในกรณีที่ต้องการส่งข้อความแบบไม่เจาะจงผู้รับ สามารถใช้ฟังก์ชัน `publish` ไปยัง `topic` ที่กำหนดได้ ซึ่งจะมีแต่ `microgear` ที่ `subscribe topic` นี้เท่านั้น ที่จะได้รับข้อความ

```
microgear.publish(topic, message);
```

arguments:

 `topic` - ชื่อของ `topic` ที่ต้องการจะส่งข้อความไปถึง

 `message` - ข้อความที่ต้องการส่ง



```
microgear.publish("/outdoor/temp","28.5");
```



Function subscribe



เป็นฟังก์ชันสำหรับในกรณีที่ **microgear** อาจจะมีความสนใจใน **topic** ใดเป็นการเฉพาะ สามารถใช้ฟังก์ชัน **subscribe()** ในการบอกรับ **message** ของ **topic** นั้นได้

```
microgear.subscribe(topic);
```

arguments:

 **topic** – ชื่อของ **topic** ที่ต้องการจะรับข้อความ



```
microgear.subscribe("/outdoor/temp");
```



Function unsubscribe



เป็นฟังก์ชันสำหรับในกรณีที่ microgear ต้องการยกเลิกการ subscribe

```
microgear.unsubscribe(topic);
```

arguments :

 **topic** – ชื่อของ topic ที่ต้องการจะยกเลิกรับข้อความ



```
microgear.unsubscribe("/outdoor/temp");
```




Function useTLS



เป็นฟังก์ชันสำหรับ enable หรือ disable TLS ซึ่งสำหรับ HTML5 microgear จะใช้ TLS เป็นค่า default

```
microgear.useTLS(tlsmode);
```

arguments:

 **tlsmode** – เป็น true หมายถึง ใช้ TLS (เป็นค่า default)
false หมายถึง ไม่ใช้ TLS



```
microgear.useTLS(false);
```







Function writeFeed



เป็นฟังก์ชันเขียนข้อมูลลง feed storage

```
microgear.writeFeed(feedid, datajson [, apikey]);
```

arguments :

-  **feedid** – ชื่อของ feed ที่ต้องการจะเขียนข้อมูล
-  **datajson** – ข้อมูลที่จะบันทึก ในรูปแบบ json
-  **apikey** – apikey สำหรับตรวจสอบสิทธิ์ หากไม่กำหนด จะใช้ default apikey ของ feed ที่ให้สิทธิ์ไว้กับ AppID



```
microgear.writeFeed("homesensor",{temp:25.7,humid:62.8,light:8.5});
```

or

```
microgear.writeFeed("homesensor",{temp:25.7,humid:62.8,light:8.5}, "df56v4rt89gaer8");
```



Events



application ที่รันบน microgear จะมีการทำงานในแบบ event driven คือเป็นการทำงานตอบสนองต่อ event ต่างๆ ด้วยการเขียน callback function ขึ้นมารองรับในลักษณะนี้

```
microgear.on(event, callback);
```

arguments:



event – ชื่อ event



callback – callback function



Events



event 'connected' เกิดขึ้นเมื่อ microgear library เชื่อมต่อกับ platform สำเร็จ

```
microgear.on("connected", function() {  
    console.log("connected");  
});
```

event 'closed' เกิดขึ้นเมื่อ microgear library ตัดการเชื่อมต่อกับ platform

```
microgear.on("closed", function() {  
    console.log("closed");  
});
```





Events



event 'error' เกิดขึ้นเมื่อมี error ขึ้นภายใน microgear

```
microgear.on("error", function(err) {  
    console.log("Error: "+err);  
});
```

event 'message' เมื่อมี message เข้ามา จะเกิด event นี้ขึ้น พร้อมกับส่งผ่านข้อมูลเกี่ยวกับ message นั้นมาทาง argument ของ callback function

```
microgear.on("message", function(topic, msg) {  
    console.log("Incoming message: "+msg);  
});
```





Events



event 'present' เกิดขึ้นเมื่อมี microgear ใน appid เดียวกัน online เข้ามาเชื่อมต่อ netpie

```
microgear.on("present", function(event) {  
    console.log(event);  
});
```

event 'absent' เกิดขึ้นเมื่อมี microgear ใน appid เดียวกัน offline หายไป

```
microgear.on("absent", function(event) {  
    console.log(event);  
});
```





WORKSHOP

Let's start !



เตรียมพร้อมการใช้งาน

- ✔ ติดตั้ง [Arduino IDE](#) และ [USB serial TTL Driver](#)
- ✔ ติดตั้ง [Package ESP8266](#) บน Arduino IDE
- ✔ ติดตั้ง [Microgear library](#) และ [DHT Sensor library](#) (v.1.2.3)
- ✔ สมัครสมาชิก NETPIE (<https://netpie.io>)




ESP8266

connect with NETPIE.



Basic microgaer

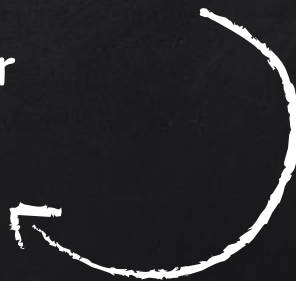
เปิดโปรแกรม Arduino IDE 

คลิกที่ File

- L Examples

- L ESP8266 Microgear

- L Basic



“

แก้ไข APPID, KEY และ SECRET

...

```
const char* ssid  = "<WIFI SSID>";    // ชื่อ ssid  
const char* password = "<WIFI KEY>";  // รหัสผ่าน wifi
```

```
#define APPID "<APPID>"                // ให้แทนที่ด้วย AppID  
#define KEY   "<KEY>"                  // ให้แทนที่ด้วย Key  
#define SECRET "<SECRET>"            // ให้แทนที่ด้วย Secret
```

...

“

Verify/Compile เป็นการตรวจสอบโค้ดที่เขียนถูกต้องหรือไม่

กด Sketch -> **Verify/Compile**

กด **Ctrl + R**

กด



เลือก Board ให้ถูกต้อง
ก่อน Verify/Compile
อยู่ในเมนู Tools



Upload เป็นการอัปโหลดหรือเขียนโค้ดลงในหน่วยความจำของอุปกรณ์

กด Sketch -> **Upload**

กด **Ctrl + U**

กด



เลือก Board และ Port ให้ถูกต้อง
ก่อน Upload
อยู่ในเมนู Tools



“

Serial Monitor เป็นดูสถานะการทำงานของอุปกรณ์ ซึ่งต้องระบุ Baud rate ให้ตรงกับที่ระบุในโค้ด เช่น `Serial.begin(<baud rate>);`

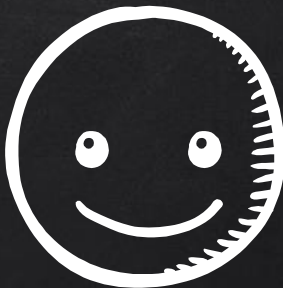
กด Tools → Serial Monitor

กด **Ctrl + Shift + M**

กด



เลือก Baud rate
ให้ตรงล่ะ!!!



เลือก Port ให้ถูกต้องด้วยนะ
อยู่ในเมนู Tools





Result

.....WiFi connected

เชื่อมต่อ WiFi สำเร็จ

IP address:

172.18.45.229

Connected to NETPIE...

เชื่อมต่อ NETPIE สำเร็จ

connected

Publish...

ทำการส่งข้อความ

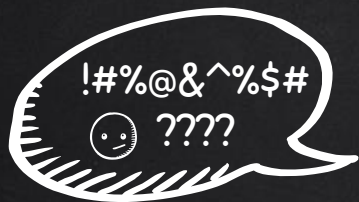
connected

connected

Incoming message --> Hello

มีข้อความส่งมา

connected





ESP8266 Switch



ดาวน์โหลดโค้ด  <https://goo.gl/zexisA>

เมื่อโหลดเสร็จจะได้ไฟล์ .zip  และแตกไฟล์ .zip

เข้าไปที่โฟลเดอร์ code -> **part2-netpie**



HTML5



Arduino



66

ESP8266 SWITCH

...

```
const char* ssid    = "<WIFI SSID>";           // ชื่อ ssid
const char* password = "<WIFI KEY>";           // รหัสผ่าน wifi
```

```
#define APPID "<APPID>"           // ให้แทนที่ด้วย AppID
#define KEY   "<KEY>"             // ให้แทนที่ด้วย Key
#define SECRET "<SECRET>"        // ให้แทนที่ด้วย Secret
#define ALIAS "switch"           // ชื่ออุปกรณ์
```

```
#define DeviceControl "html5"      // ชื่ออุปกรณ์ที่ต้องการส่งข้อความไปให้
```

...

66

ESP8266 SWITCH

```
void setup() {  
  microgear.on(MESSAGE,onMsghandler);  
  microgear.on(CONNECTED,onConnected);  
  
  Serial.begin(115200);  
  Serial.println("Starting...");  
  
  pinMode(LED_BUILTIN, OUTPUT);  
  digitalWrite(LED_BUILTIN, HIGH);  
  
  ...  
}
```

66

ESP8266 SWITCH

```
void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen) {  
    ...  
    if(*(char *)msg == '1'){  
        digitalWrite(LED_BUILTIN, LOW); // LED on  
        microgear.chat(DeviceControl, "1");  
    }else{  
        digitalWrite(LED_BUILTIN, HIGH); // LED off  
        microgear.chat(DeviceControl, "0");  
    }  
    ...  
}
```




Result

.....WiFi connected

IP address:

172.18.45.229

Connected to NETPIE...

เชื่อมต่อ WiFi สำเร็จ

เชื่อมต่อ NETPIE สำเร็จ

ผังอุปกรณ์ทำเสร็จแล้ว ไปทำตัวควบคุมกันต่อเลย



“

HTML SWITCH

แก้ไขไฟล์ config.json เป็นไฟล์ที่เก็บ APPID, KEY และ SECRET สำหรับเชื่อมต่อ NETPIE

```
var config = '{ "appid": "<APPID>", "key": "<KEY>", "secret": "<SECRET>" }';
```

“

HTML SWITCH

```
function changeState(){  
  
    var stateON = document.getElementById('toggle').checked;  
    if(stateON){  
        microgear.chat("switch","1");  
    }else{  
        microgear.chat("switch","0");  
    }  
  
    document.getElementById('toggle').checked = !stateON;  
  
}
```

“

HTML SWITCH

```
microgear.on('message',function(topic,msg) {  
  
  if(msg==1){  
    document.getElementById('state').innerHTML = "ON";  
    document.getElementById('toggle').checked = true;  
  }else{  
    document.getElementById('state').innerHTML = "OFF";  
    document.getElementById('toggle').checked = false;  
  }  
  
});
```

เชื่อมต่อ WiFi สำเร็จ



Result

.....WiFi connected

IP address:
172.18.45.229

เชื่อมต่อ NETPIE สำเร็จ

Connected to NETPIE...

Incoming message --> 1

Incoming message --> 0

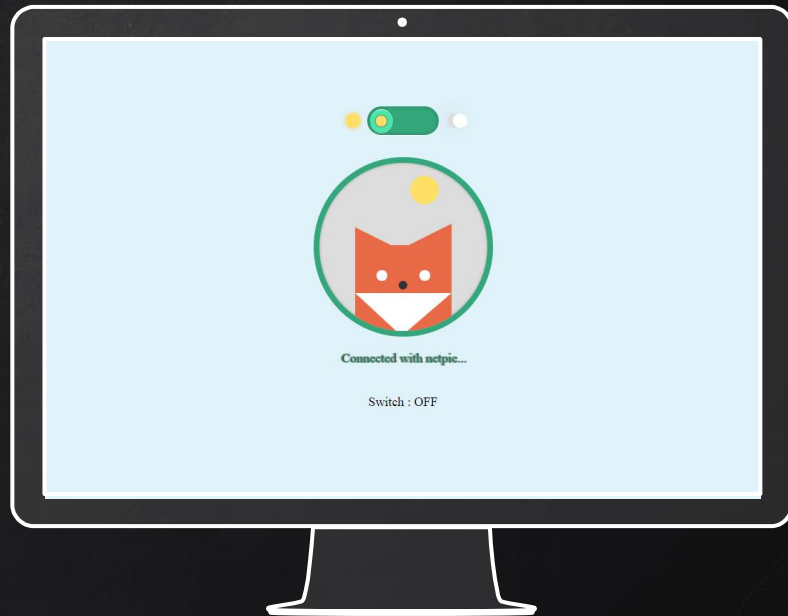
0

=



1

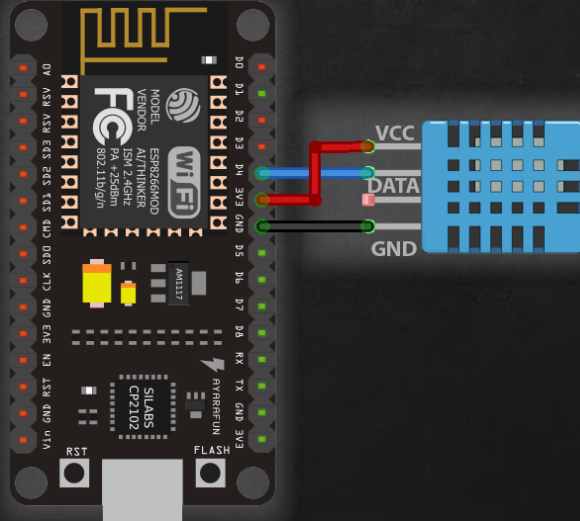
=







ESP8266 DHT Sensor



66

ESP8266 DHT SENSOR

```
#include "DHT.h"
```

```
...
```

```
#define DHTPIN D4
```

```
#define DHTTYPE DHT11
```

```
DHT dht(DHTPIN, DHTTYPE);
```

```
char str[32];
```

```
int humid = 0;
```

```
int temp = 0;
```

```
unsigned long lastTimeDHT = 0;
```

```
...
```

66

ESP8266 DHT SENSOR

```
void setup() {
```

```
...
```

```
  dht.begin();
```

```
...
```

```
}
```

```
void loop() {
```

```
...
```

```
  int h = dht.readHumidity();
```

```
  int t = dht.readTemperature();
```

```
...
```

```
}
```



Result

.....WiFi connected

IP address:

172.18.45.229

Connected to NETPIE...

Humidity: 60 %

Humidity: 60 %

Humidity: 59 %

Temperature: 30 *C Sending --> 60,30

Temperature: 30 *C Sending --> 60,30

Temperature: 30 *C Sending --> 59,30

เชื่อมต่อ WiFi สำเร็จ

เชื่อมต่อ NETPIE สำเร็จ



ส่งข้อความ

อุปกรณ์ทำเสร็จแล้ว ไปทำหน้าที่แสดงผลกันต่อเลย



“

HTML SWITCH & DHT

เพิ่มโค้ดส่วนการเช็ค topic dht ที่ส่ง message เข้ามา

```
microgear.on('message',function(topic,msg) {  
    ...  
    if(topic==("/"+APPID+"/dht")){  
        var data = msg.split(',');  
        if(data.length==2){  
            document.getElementById('humid').innerHTML = data[0];  
            document.getElementById('temp').innerHTML = data[1];  
        }  
    }  
}
```

“

HTML SWITCH & DHT

เพิ่มโค้ดส่วนการ subscribe topic สำหรับรับ message

```
microgear.on('connected', function() {  
  ...  
  microgear.subscribe("/dht");  
  ...  
});
```


HTML SWITCH

[illegible]

เชื่อมต่อ WiFi สำเร็จ



Result

.....WiFi connected

IP address:
172.18.45.229

เชื่อมต่อ NETPIE สำเร็จ

Connected to NETPIE...

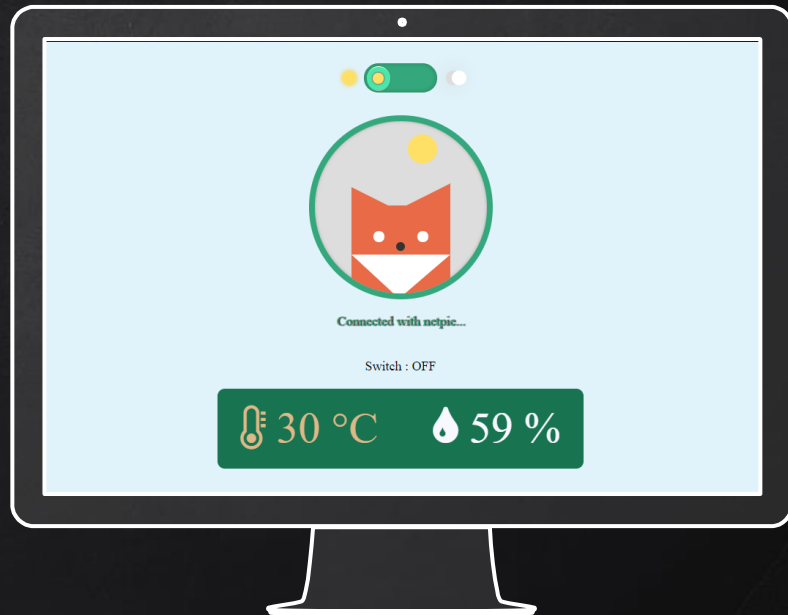
Sending --> 60,30

Incoming message --> 1

Sending --> 59,30

Incoming message --> 0

0 =  1 = 







THANKS!

Any questions?

Join us

<https://netpie.io>

You can find me at

Line & Facebook : [anun.panya](#)

anun.panya@nectec.or.th