



Part 3: IoT Development with NETPIE

by NETPIE.io

หัวข้อ Code Lab

>>> LAB01 : HTML Chat

สร้าง netpie device ในรูปแบบของไฟล์ HTML ไฟล์เดียว ซึ่งจะให้ส่งข้อความ chat หาดตัวเองผ่าน NETPIE

>>> LAB02 : HTML Chat with neighbor

สร้างไฟล์ HTML 2 ไฟล์ ให้ส่งข้อความ chat ถึงกัน ระหว่าง browser 2 หน้าต่าง ที่สามารถเปิดกันคนละเครื่องได้

>>> LAB03 : ESP8266 Chat

เราจะมาลองสร้าง hardware device ด้วย NodeMCU ให้ส่งข้อความ chat หาดตัวเองผ่าน NETPIE

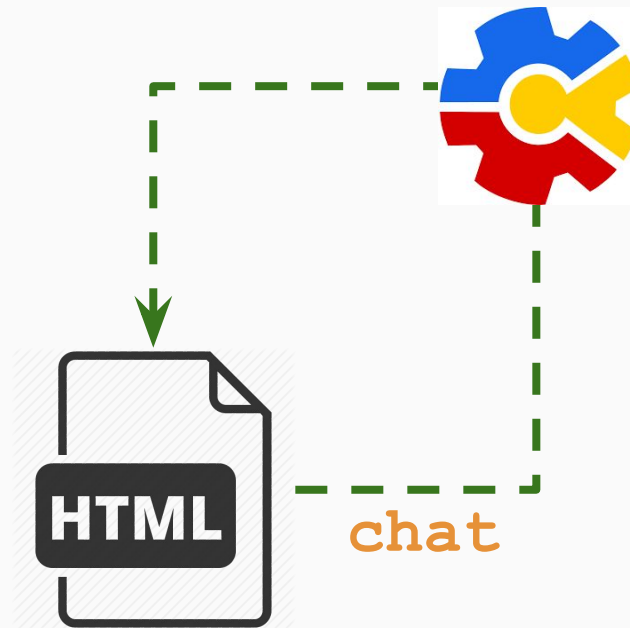
>>> LAB04 : ESP8266 Switch

ทำปุ่มบน NodeMCU ให้เป็น switch เปิดปิดไฟบน NodeMCU ของเพื่อนข้างๆ

>>> LAB05 : ESP8266 DHT

ทำ NodeMCU ให้เป็น sensor ตรวจสอบอุณหภูมิและความชื้น ส่งค่าออกไปขึ้นกระดาน

LAB 01 - HTML Chat



สร้าง netpie device ในรูปแบบของไฟล์ HTML
ไฟล์เดียว ซึ่งจะให้ส่งข้อความ chat หาตัวเองผ่าน
NETPIE

LAB 01 - HTML Chat

ไฟล์ basic.html

ส่วนที่ 1 ส่วนการเรียกใช้งาน Microgear library

```
<script src="https://cdn.netpie.io/microgear.js"></script>
```

ส่วนที่ 2 การประกาศตัวแปรที่จำเป็นสำหรับการเชื่อมต่อ NETPIE

```
<script type="text/javascript">  
  const APPID = "YOUR_APPID";  
  const KEY = "YOUR_KEY";  
  const SECRET = "YOUR_SECRET";  
  const ALIAS = "html5";  
  
  var microgear = Microgear.create({  
    key: KEY,  
    secret: SECRET,  
    alias : ALIAS  
  });  
</script>
```

LAB 01 - HTML Chat

ส่วนที่ 3 event listener

```
<script type="text/javascript">
    microgear.on('message',function(topic,msg) {
        document.getElementById("data").innerHTML += msg + "<br>";
    });

    microgear.on('connected', function() {
        microgear.setAlias(ALIAS);
        document.getElementById("data").innerHTML = "Now I am conn..
        ...
    });
</script>
```

LAB 01 - HTML Chat

ส่วนที่ 4 loop ส่งข้อความ

```
<script type="text/javascript">  
    ...  
    setInterval(function() {  
        k = k+1;  
        microgear.chat(ALIAS,"Chat to myself no: " + k);  
    },5000);  
</script>
```

ส่วนที่ 5 connect to NETPIE

```
microgear.connect(APPID);
```

ส่วนที่ 6 เป็น html div ที่เนื้อหาจะถูก append ถ้ามีข้อความส่งเข้ามา

```
<div id="data">_____</div>
```

LAB 01 - HTML Chat

```
<script src="https://cdn.netpie.io/microgear.js"></script>
<script type="text/javascript">
    var k = 0;
    var microgear = Microgear.create({
        key: YOUR_KEY,
        secret: YOUR_SECRET,
        alias : "html5"
    });

    microgear.on('message',function(topic,msg) {
        document.getElementById("data").innerHTML += msg + "<br>";
    });
    microgear.on('connected', function() {
        microgear.setAlias(ALIAS);
        document.getElementById("data").innerHTML = "Now I am connected..
        setInterval(function() {
            k = k+1;
            microgear.chat(ALIAS,"Chat to myself no: " + k);
        },1000);
    });

    microgear.connect(YOUR_APPID);
</script>
<div id="data">_____</div>
```

file:///Users/chavee/Workspace/NETPIE-Training/nstda-w

Now I am connected with netpie...

Chat to myself no: 1

Chat to myself no: 2

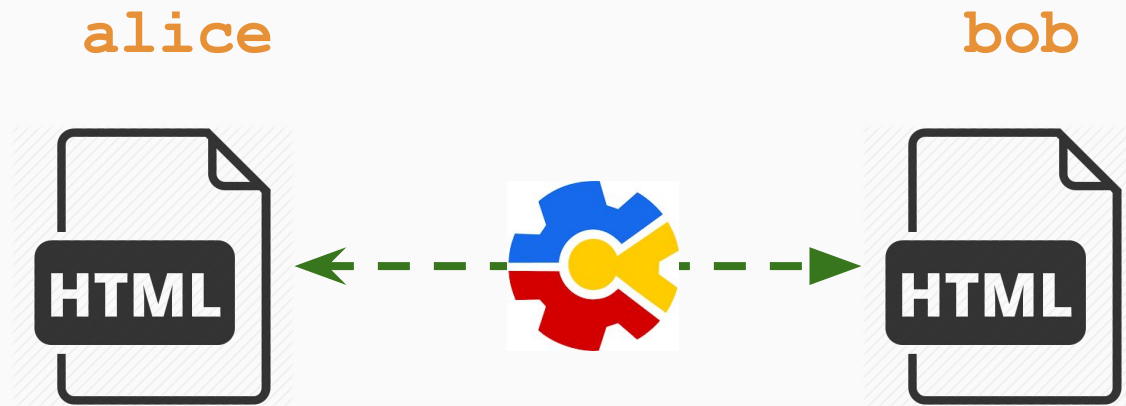
Chat to myself no: 3

Chat to myself no: 4

Chat to myself no: 5

Chat to myself no: 6

LAB 02 - HTML Chat with neighbor



`chat("bob", "Hi")` ----->

"Hi"

"Hello"

<-----

`chat("alice", "Hello")`

สร้างไฟล์ HTML 2 ไฟล์ ให้ส่งข้อความ chat ถึงกัน ระหว่าง browser 2 หน้าต่าง ที่สามารถเปิดกันคนละเครื่องได้

LAB 02 - HTML Chat with neighbor

ไฟล์ alice.html

ส่วนที่ 1 เรียกใช้ไฟล์ config ที่กำหนด APPID, KEY, SECRET

```
<script type="text/javascript" src="./config.js"></script>
```

ส่วนที่ 2 กำหนดชื่ออุปกรณ์ (ALIAS)

```
<script type="text/javascript">  
  const ALIAS = "alice";  
  const NEIGHBOR = "bob";  
</script>
```

ในส่วนการทำงาน Alice จะส่ง message หา NEIGHBOR ซึ่งก็คือ bob ทุก 1 วินาที

```
setInterval(function() {  
  k = k+1;  
  var msg = "Hello from "+ALIAS+" #"+k;  
  document.getElementById("data").innerHTML += "Sending...  
  microgear.chat(NEIGHBOR, msg);  
},1000);
```

LAB 02 - HTML Chat with neighbor

ไฟล์ config.js เป็นไฟล์เก็บ APPID, KEY และ SECRET ให้แก้ไขตรงกับ key ตัวเอง

```
const APPID = "YOUR_APPID";  
const KEY = "YOUR_KEY";  
const SECRET = "YOUR_SECRET";
```

LAB 02 - HTML Chat with neighbor

ไฟล์ bob.html

จะตอบกลับทุกครั้งเมื่อมีข้อความส่งเข้ามา การตอบกลับจะส่งไปยัง NEIGHBOR ซึ่งก็คือ alice

```
<script type="text/javascript">
```

```
    const ALIAS = "bob";
```

```
    const NEIGHBOR = "alice";
```

```
    microgear.on('message',function(topic,msg) {
```

```
        var reply = "Hi, I'm "+ALIAS+". I received: '"+msg+"'";
```

```
        document.getElementById("data").innerHTML = "Receiving --> " +
```

```
        document.getElementById("data").innerHTML += "Replying \"" +
```

```
        microgear.chat(NEIGHBOR,reply);
```

```
    });
```

```
</script>
```

LAB 02 - HTML Chat with neighbor

```
< > ↻ 🏠 ⓘ file:///Users/chavee/Workspace/NETPIE-Trai

I am : alice
Now I am connected with netpie...
Sending "Hello from alice #1" --->
Receiving --> Hi, I'm bob. I received: 'Hello from alice #1'
Sending "Hello from alice #2" --->
Receiving --> Hi, I'm bob. I received: 'Hello from alice #2'
```



Alice

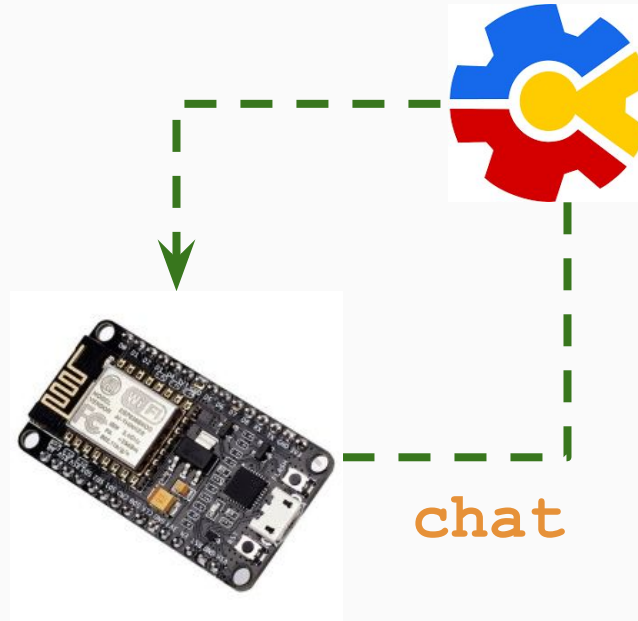


Bob

```
⬅ ⓘ file:///Users/chavee/Workspace/NETPIE-Training/nstda-workshop/02-N

I am : bob
Receiving --> Hello from alice #2
Replying "Hi, I'm bob. I received: 'Hello from alice #2'" --->
```

LAB 03 - ESP8266 Chat



เราจะมาลองสร้าง hardware device ด้วย NodeMCU ให้
ส่งข้อความ chat หาดตัวเองผ่าน NETPIE

LAB 03 - ESP8266 Chat

ไฟล์ Basic.ino

ส่วนที่ 1 ทำการ include library และประกาศตัวแปรต่างๆ

```
#include <ESP8266WiFi.h>
#include <MicroGear.h>
```

```
// ----- แก่ค่า config 6 ค่าข้างล่างนี้ -----
const char* ssid      = "YOUR_WIFI_SSID";          // ชื่อ ssid
const char* password  = "YOUR_WIFI_PASSWORD";      // รหัสผ่าน wifi

#define APPID          "APPID"                     // ให้แทนที่ด้วย AppID รวม
#define KEY            "KEY"                       // ให้แทนที่ด้วย Key รวม
#define SECRET         "SECRET"                   // ให้แทนที่ด้วย Secret รวม
#define ALIAS          "YOUR_UNIQUE_ALIAS"         // แทนที่ด้วยหมายเลขของท่าน เช่น "A01"
// -----
```

```
WiFiClient client;
```

```
MicroGear microgear(client);
```

LAB 03 - ESP8266 Chat

ส่วนที่ 2 ส่วนของฟังก์ชัน event

2.1 onMsghandler เป็นฟังก์ชันสำหรับรับข้อความ

```
void onMsghandler(char *topic, uint8_t* msg, unsigned int msglen)
{
    Serial.print("Incoming message --> ");
    msg[msglen] = '\0';
    Serial.println((char *)msg);
}
```

2.2 onConnected เป็นฟังก์ชันเมื่อเชื่อมต่อ NETPIE สำเร็จ

```
void onConnected(char *attribute, uint8_t* msg, unsigned int msglen)
{
    Serial.println("Connected to NETPIE...");
    microgear.setAlias(ALIAS);
}
```

LAB 03 - ESP8266 Chat

ส่วนที่ 3 `setup()` เป็นส่วนที่จะทำการกำหนดค่าเริ่มต้นให้อุปกรณ์

- `microgear.on` กำหนด callback function ให้ event
Event:

MESSAGE	ทำงานเมื่อมีข้อความส่งเข้ามา
CONNECTED	ทำงานเมื่อเชื่อมต่อ NETPIE สำเร็จ
PRESENT	ทำงานเมื่ออุปกรณ์ Online/Offline
ERROR	ทำงานเมื่อ Error เกิดขึ้น

- `microgear.init` ทำการ initiate ตัวแปร `microgear`
- `microgear.connect()` สั่งให้ `microgear` เชื่อมต่อไปยัง NETPIE

```
void setup() {  
    microgear.on(MESSAGE, onMsghandler) ;  
    microgear.on(CONNECTED, onConnected) ;  
  
    microgear.init(KEY, SECRET, ALIAS) ;  
    microgear.connect(APPID) ;  
}
```


LAB 03 - ESP8266 Chat

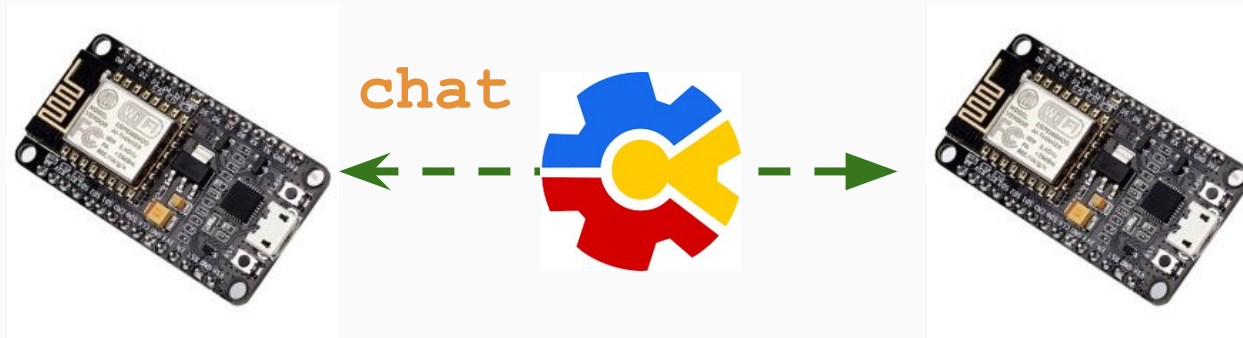
ส่วนที่ 4 `loop()` เป็นที่ทำงานอยู่ตลอดเวลา ใน `loop` จะทำการส่งข้อความทุกๆ 1 วินาที ด้วยคำสั่ง `microgear.chat()` ส่งให้ตัวเอง (ALIAS)

```
void loop() {
    if (microgear.connected()) {
        // เรียก function นี้เป็นระยะๆ ไม่เช่นนั้น connection จะหลุด
        microgear.loop();

        if (millis() - last_chat_time >= 1000) {
            Serial.println("Send chat message >>>");

            // chat ทั่วโลก device ชื่อ ALIAS ซึ่งก็คือชื่อของตัวเอง
            microgear.chat(ALIAS, "Hello..");
            last_chat_time = millis();
        }
    }
    else {
        Serial.println("connection lost, reconnect...");
        microgear.connect(APPID);
    }
}
```

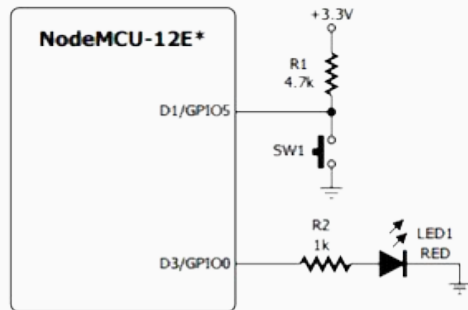
LAB 04 - ESP8266 Switch



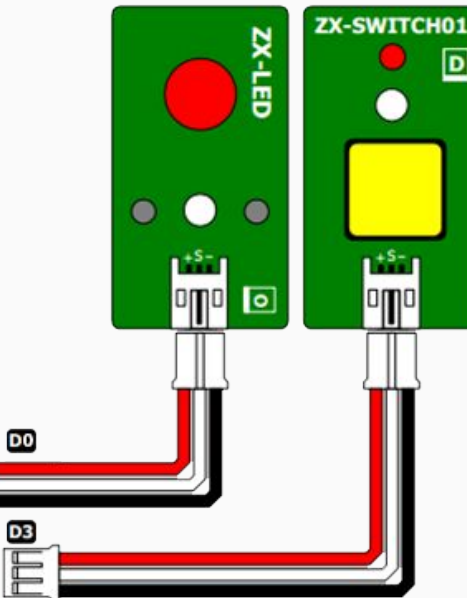
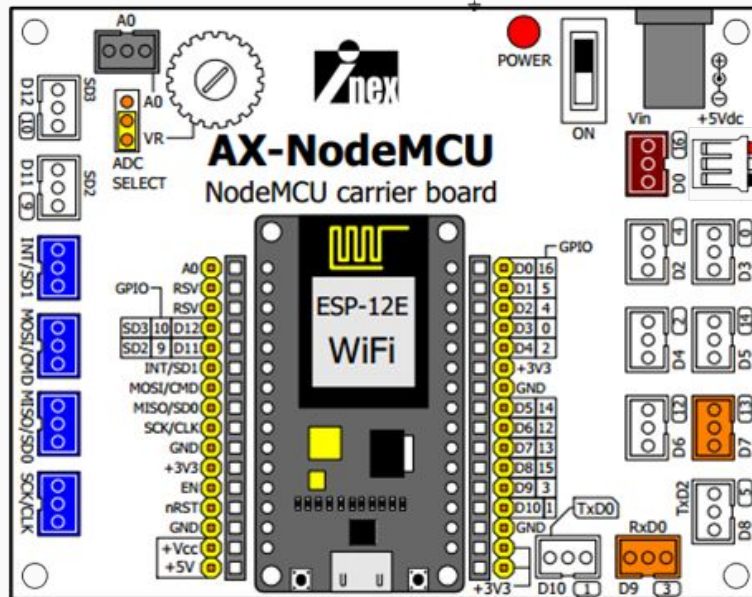
ทำปุ่มบน NodeMCU ให้เป็น switch เปิดไฟบน NodeMCU ของเพื่อนข้างๆ

LAB 04 - ESP8266 Switch

การต่ออุปกรณ์



(A) วงจรสำหรับต่อวงจรทดลองบนบอร์ด



(B) ต่อวงจรด้วยอุปกรณ์ในชุด
IoT Education Kit

LAB 04 - ESP8266 Switch

ไฟล์ ESP8266_Switch.ino

ส่วนที่ 1 ทำการ include library และประกาศตัวแปรต่างๆ

```
// ----- แก้ค่า config 7 ค่าข้างล่างนี้ -----
const char* ssid      = "YOUR_WIFI_SSID";      // ชื่อ ssid
const char* password  = "YOUR_WIFI_PASSWORD";  // รหัสผ่าน wifi

#define APPID          "APPID"                  // ให้แทนที่ด้วย AppID รวม
#define KEY             "KEY"                    // ให้แทนที่ด้วย Key รวม
#define SECRET         "SECRET"                 // ให้แทนที่ด้วย Secret รวม

#define ALIAS          "YOUR_UNIQUE_ALIAS"       // แทนที่ด้วยหมายเลขของท่าน เช่น "A01"
#define NEIGHBOR       "NEIGHBOR_ALIAS"         // ชื่ออุปกรณ์ของเพื่อน เช่น "A02"
// -----

#define BUTTONPIN      D3                        // pin ที่ต่อกับปุ่ม Flash บน NodeMCU
#define LEDPIN         LED_BUILTIN              // pin ที่ต่อกับไฟ LED บน NodeMCU
```

LAB 04 - ESP8266 Switch

ส่วนที่ 2 ฟังก์ชัน updateLED(int state)

```
void updateLED(int state) {  
    // ไฟ LED บน NodeMCU จะติดก็ต่อเมื่อส่งค่า LOW ไปให้ LEDPIN  
    if(state==1 && currentLEDState == 0){  
        currentLEDState = 1;  
        digitalWrite(LEDPIN, LOW);    //LED ON  
    }  
    else if (state==0 && currentLEDState == 1) {  
        currentLEDState = 0;  
        digitalWrite(LEDPIN, HIGH);    //LED OFF  
    }  
}
```

LAB 04 - ESP8266 Switch

ส่วนที่ 3 กำหนดเงื่อนไขเปลี่ยนสถานะ LED เมื่อมีข้อความส่งค่าสถานะเข้ามา
ถ้าส่งเลข 0 LED จะดับ, เลข 1 LED จะติด

```
void onMsgghandler(char *topic, uint8_t* msg, unsigned int  
msglen) {  
    Serial.print("Incoming message --> ");  
    msg[msglen] = '\\0';  
    Serial.println((char *)msg);  
  
    if (*(char *)msg == '0') updateLED(0);  
    else if (*(char *)msg == '1') updateLED(1);  
}
```

LAB 04 - ESP8266 Switch

ส่วนที่ 4 กำหนด Pin Mode ใน void setup()

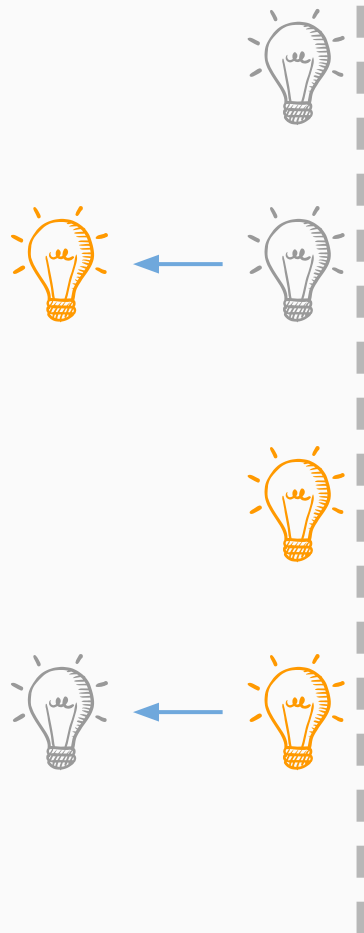
```
void setup() {  
    pinMode(LEDPIN, OUTPUT);    // ตั้ง LEDPIN เป็น OUTPUT mode  
    pinMode(BUTTONPIN, INPUT);  // ตั้ง BUTTONPIN เป็น INPUT mode  
    updateLED(currentLEDState);  
    ...  
}
```

ส่วนที่ 5 อ่านค่าปุ่มสวิตช์ และส่งข้อความเปลี่ยนสถานะ LED บน NodeMCU ของเพื่อน

```
void loop() {  
  
    if (digitalRead(BUTTONPIN)==HIGH) currentButtonState = 0;  
    else currentButtonState = 1;  
  
    if(currentButtonState != lastButtonState){  
        microgear.chat(NEIGHBOR, currentButtonState);  
        lastButtonState = currentButtonState;  
    }  
}
```

LAB 04 - ESP8266 Switch

A



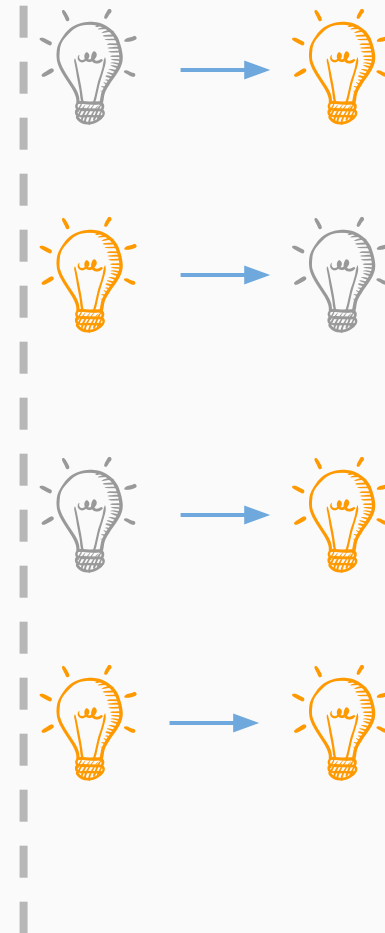
A Chat 1 ---> 1

1 <--- B Chat 1
A Chat 0 ---> 0

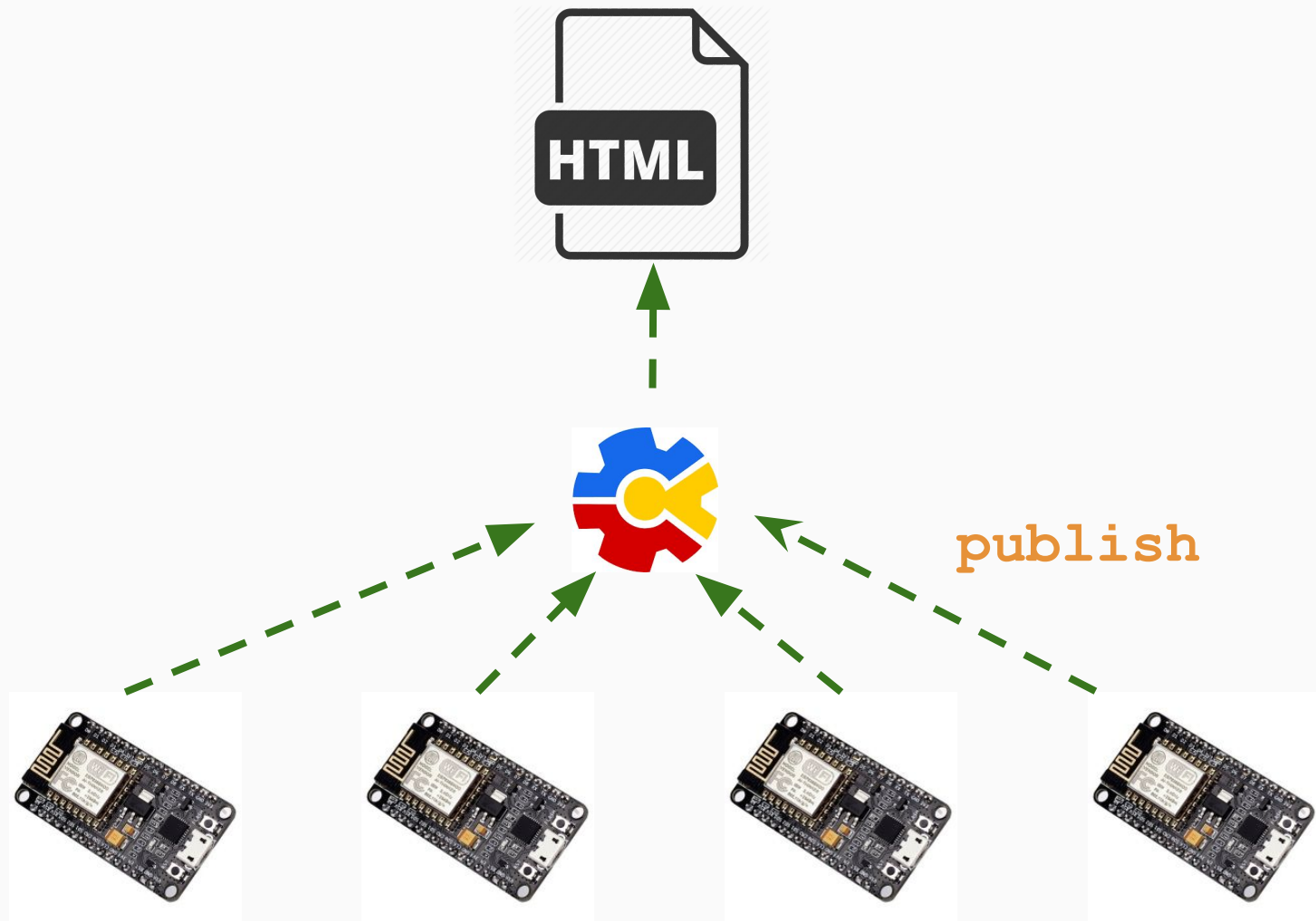
A Chat 1 ---> 1

0 <--- B Chat 0
A Chat 1 ---> 1

B

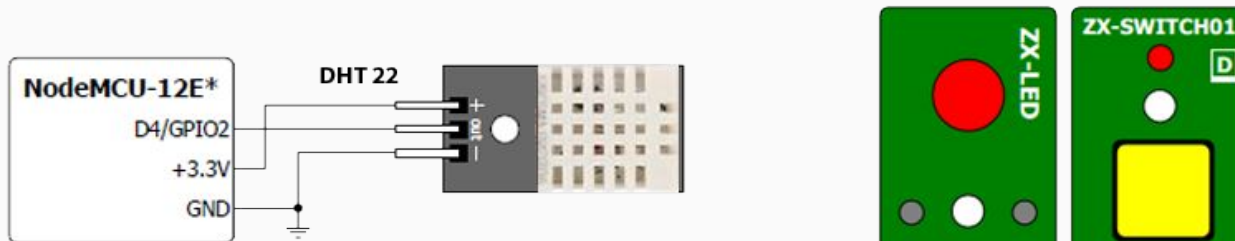


LAB 05 - ESP8266 DHT

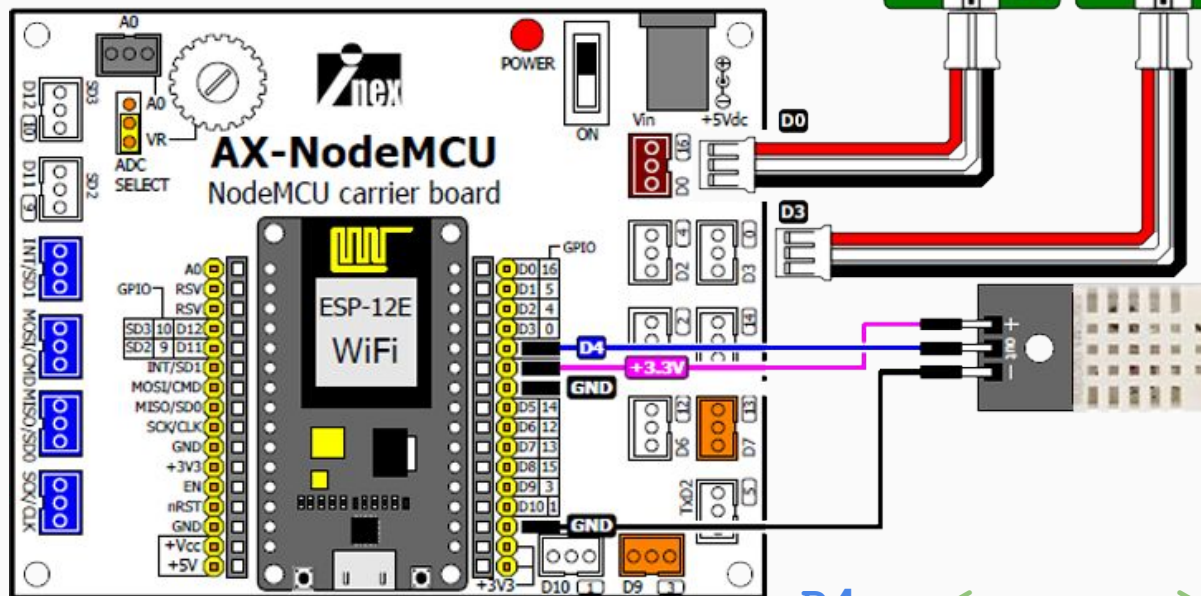


LAB 05 - ESP8266 DHT

การต่ออุปกรณ์



(A) วงจรสำหรับต่อวงจรทดลองบนบอร์ด



D4	<----->	out (DATA)
3v3	<----->	+ (VCC)
GND	<----->	- (GND)

LAB 05 - ESP8266 DHT

ไฟล์ ESP8266_DHT_LED.ino

ส่วนที่ 1 ประกาศตัวแปรต่างๆ

```
#include <ESP8266WiFi.h>
#include <MicroGear.h>
#include "DHT.h"
// ----- แก้ค่า config 7 ค่าข้างล่างนี้ -----
const char* ssid      = "YOUR_WIFI_SSID";      // ชื่อ ssid
const char* password = "YOUR_WIFI_PASSWORD"; // รหัสผ่าน wifi

#define APPID    "APPID"                // ให้แทนที่ด้วย AppID รวม
#define KEY      "KEY"                  // ให้แทนที่ด้วย Key รวม
#define SECRET   "SECRET"               // ให้แทนที่ด้วย Secret รวม

#define ALIAS     "YOUR_UNIQUE_ALIAS"      // แทนที่ด้วย alias ของท่านเช่น "A01"
#define NEIGHBOR "NEIGHBOR_ALIAS"         // ชื่ออุปกรณ์ของเพื่อน เช่น "A02"
// -----

#define LEDSTATETOPIC "/ledstate/" ALIAS // topic สำหรับ publish สถานะ led
#define DHTDATATOPIC  "/dht/" ALIAS      // topic สำหรับ publish ส่งข้อมูล dht
#define DHTPIN        D4                 // GPIO2 ขาที่ต่อเข้ากับขา DATA (บางโมดูลใช้คำว่า OUT)
#define DHTTYPE        DHT22             // e.g. DHT11, DHT21, DHT22
DHT dht(DHTPIN, DHTTYPE);
```

LAB 05 - ESP8266 DHT

ส่วนที่ 2 initial ตัวแปร dht

```
void setup() {  
    ...  
    dht.begin(); // setup ตัวแปรสำหรับอ่านค่า DHT Sensor  
    ...  
}
```

ส่วนที่ 3 อ่านค่าปุ่มสวิตช์ ถ้ามีการเปลี่ยนแปลง ให้ publish สถานะใหม่ออกไป

```
void loop() {  
    ...  
    if (digitalRead(BUTTONPIN)==HIGH) currentButtonState = 0;  
    else currentButtonState = 1;  
  
    if (currentButtonState != lastButtonState){  
        microgear.chat(NEIGHBOR, currentButtonState);  
        lastButtonState = currentButtonState;  
    }  
    ...  
}
```

LAB 05 - ESP8266 DHT

ส่วนที่ 4 รอรับ message ถ้ามีใครส่งเข้ามาเป็น 0 สั่งปิด LED, 1 สั่งเปิด LED

```
void onMsgghandler(char *topic, uint8_t* msg, unsigned int msglen)
{
    Serial.print("Incoming message --> ");
    msg[msglen] = '\0';
    Serial.println((char *)msg);

    if (*(char *)msg == '0') updateLED(0);
    else if (*(char *)msg == '1') updateLED(1);
}
```

LAB 05 - ESP8266 DHT

ส่วนที่ 5 อ่านค่าอุณหภูมิ ความชื้น และ publish ค่าออกไป (มี HTML หน้ากระดาน subscribe อยู่)

```
void loop() {  
    ...  
    if(millis() - lastDHTRead > 2000){           // ทำทุก 2 วินาที  
        float humid = dht.readHumidity();        // อ่านค่าความชื้น  
        float temp  = dht.readTemperature();    // อ่านค่าอุณหภูมิ  
        lastDHTRead = millis();  
  
        Serial.print("Humid: "); Serial.print(humid); Serial.print(" %, ");  
        Serial.print("Temp: "); Serial.print(temp); Serial.println(" °C ");  
  
        if (isnan(humid) || isnan(temp)) { // ตรวจสอบค่า humid และ temp เป็นตัวเลขหรือไม่  
            Serial.println("Failed to read from DHT sensor!");  
        }  
        else{  
            // เตรียมสตริงในรูปแบบนี้ --> "{humid},{temp}"  
            String datastring = (String)humid+","+ (String)temp;  
            Serial.print("Sending --> ");  
            Serial.println(datastring);  
            microgear.publish(DHTDATATOPIC,datastring);  
        }  
    }  
    ...  
}
```

LAB 05 - ESP8266 DHT

ตัวอย่าง

Alias : A01

Topic ที่ Publish ส่งไปจะเป็น /dht/A01

ผลลัพธ์บน serial monitor

```
Connected to NETPIE...
Humidity: 58.90 %RH , Temperature: 24.50 *C
Sending --> 58.90,24.50,0
Humidity: 58.90 %RH , Temperature: 24.50 *C
Sending --> 58.90,24.50,0
Humidity: 59.40 %RH , Temperature: 24.50 *C
Sending --> 59.40,24.50,0
Humidity: 59.40 %RH , Temperature: 24.50 *C
Sending --> 59.40,24.50,0
Humidity: 59.40 %RH , Temperature: 24.50 *C
Sending --> 59.40,24.50,0
```