

การพัฒนา AX-WiFi

บน

Platform Arduino



ฝึกอบรมหลักสูตรระยะสั้น

ฐานสมาร์ตนาฬุ่นที่ 3

Education to Employment :

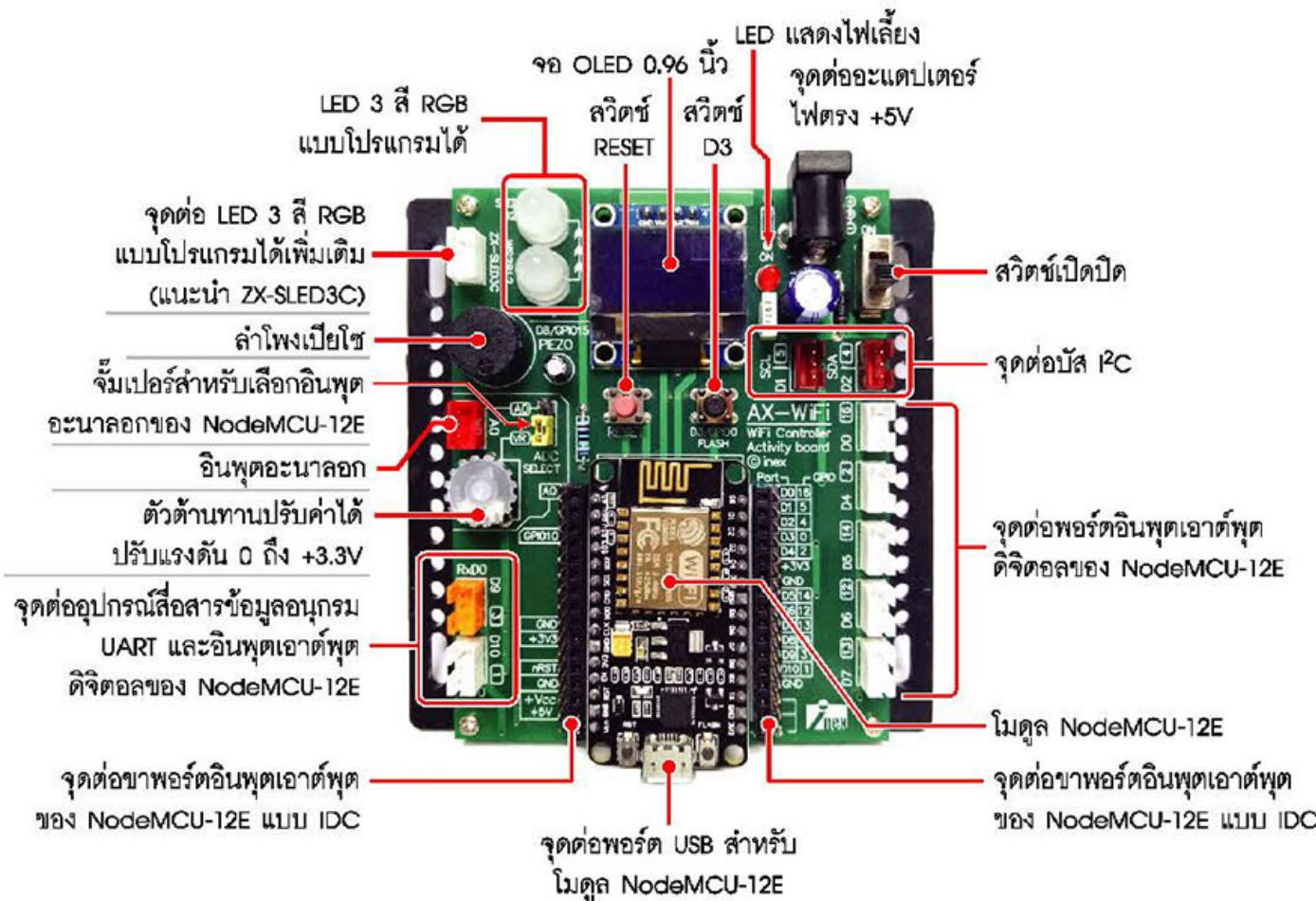
Vocational Boot Camp ^{3rd}

หลักสูตร

ติดตั้งระบบควบคุมอุปกรณ์ไฟฟ้าภายในบ้านด้วยเทคโนโลยี

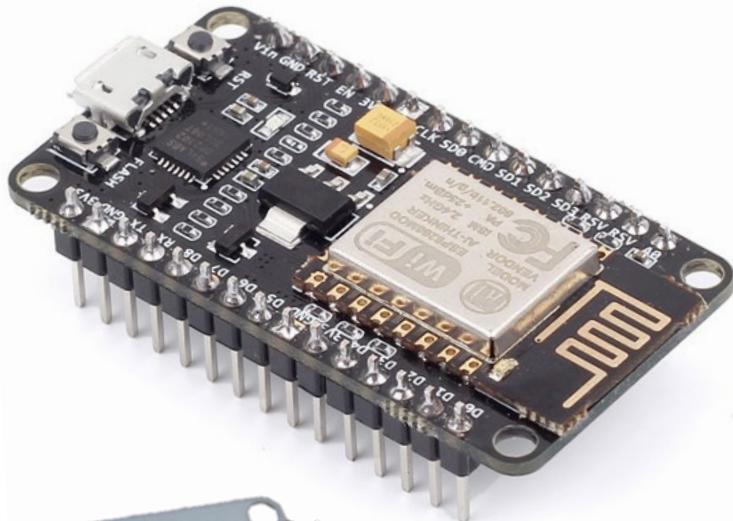
Internet of Things

AX-WiFi



NodeMCU-12E

ESP8266-12E (ESP8266EX)



MCU 32 BIT

หน่วยความจำ 4 MByte (12E)

RAM ประมาณ 112 kByte

WIFI ในตัว

USB 2 Serial สำหรับดาวน์โหลด
ไฟล์เลี้ยงผ่านพอร์ต USB ได้
ดาวน์โหลดผ่าน USB

GPIO 16 ขา

ANALOG 1 ช่อง

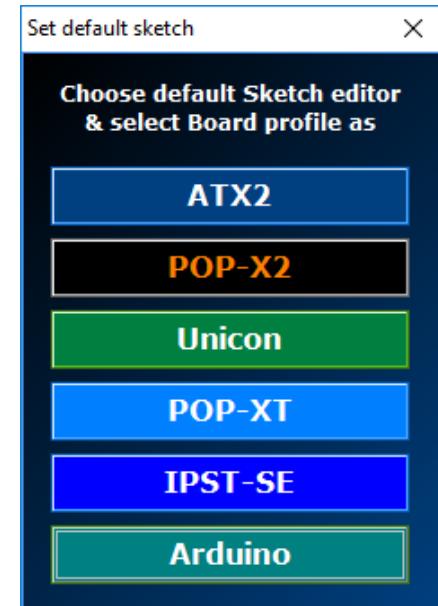
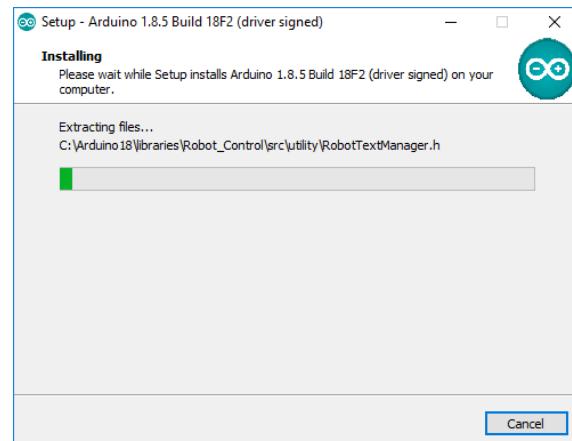
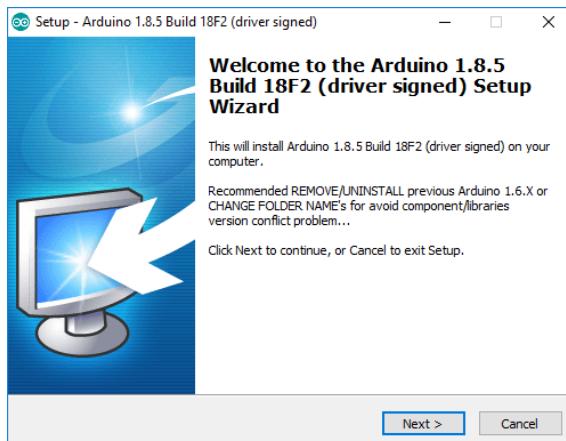
Install Arduino



ติดตั้งโปรแกรม Arduino 1.8.x ขึ้นไป

สามารถดาวน์โหลดได้ที่

http://www.inex.co.th/store/software/Arduino1.8.5_IoT_Setup170714.exe

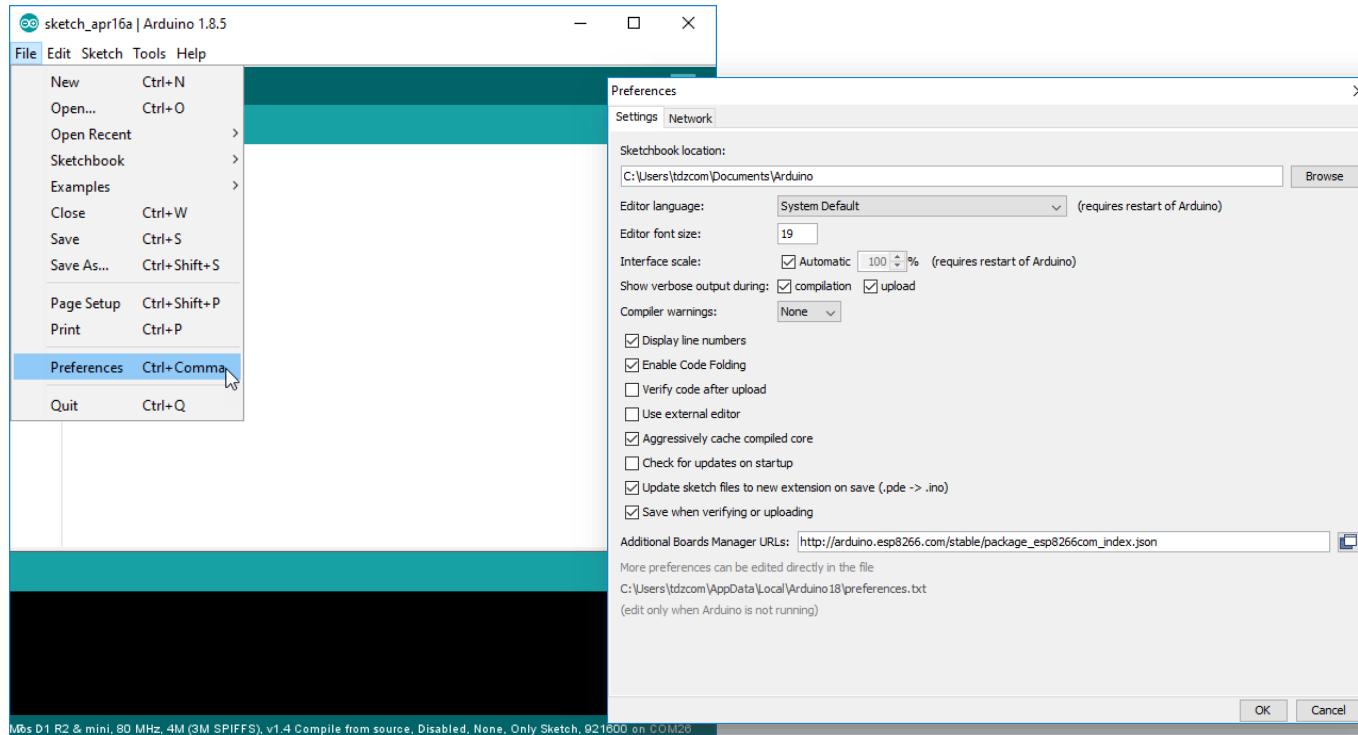


ทางบริษัท INEX ได้ทำการเพิ่มบอร์ดของบริษัทเข้าไปในตัวโปรแกรม

ติดตั้งบอร์ด NodeMCU ลงบนโปรแกรม Arduino

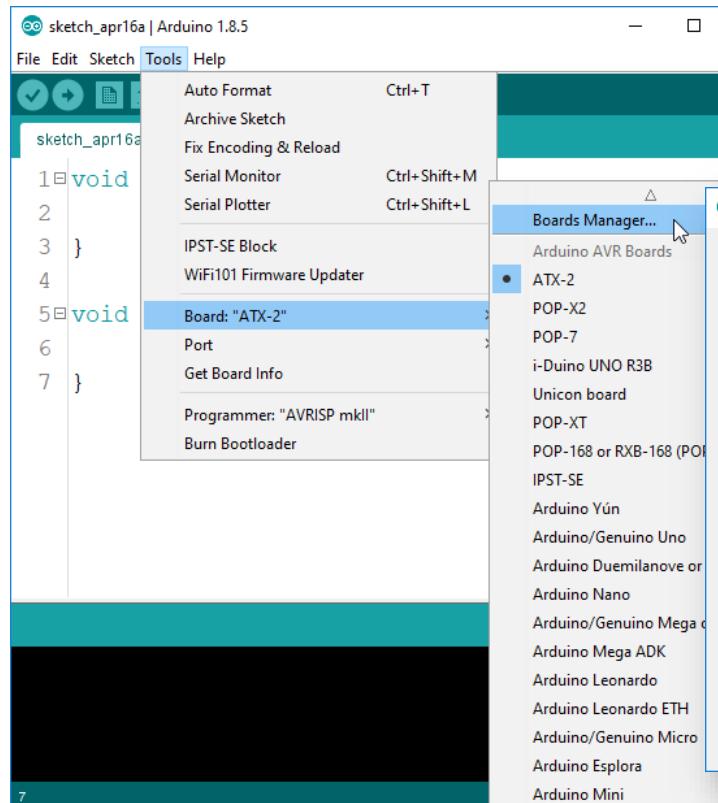
เพิ่ม URL ในช่อง Additional Boards Manager URLs:

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

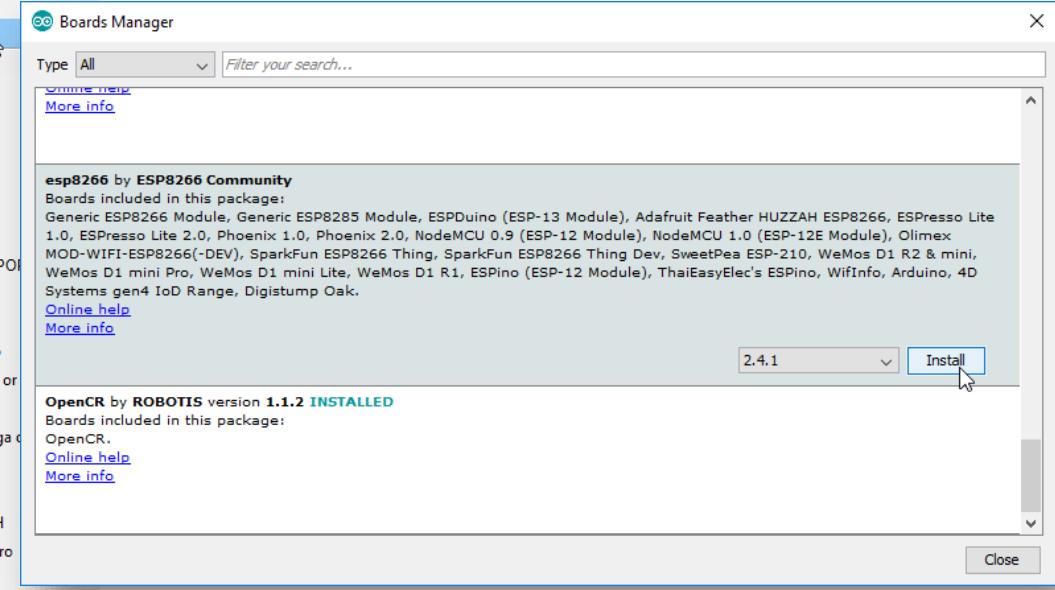


ติดตั้งบอร์ด AX-WiFi ลงบนโปรแกรม Arduino

ไปที่ Board Manager เพื่อติดตั้ง ขั้นตอนนี้จะต้องต่ออินเตอร์เน็ต



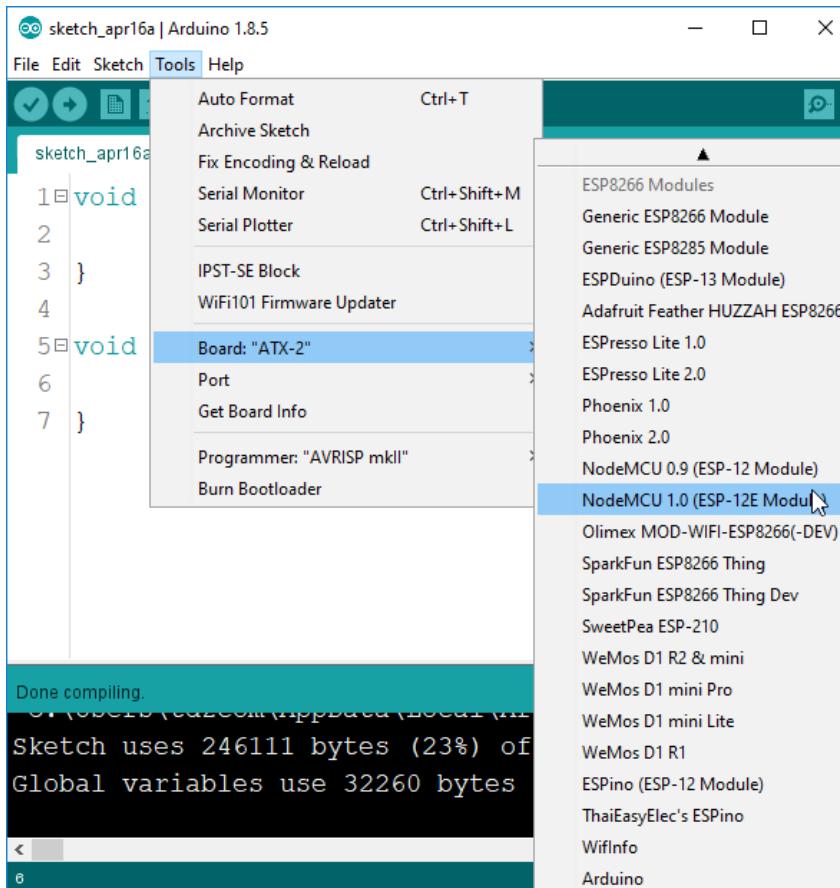
เลือกเวอร์ชันที่ต้องการ แล้วคลิก Install



ติดตั้งบอร์ด AX-WiFi ลงบนโปรแกรม Arduino

ตรวจสอบการติดตั้ง ไปที่ tools>Board เลื่อนลงจะเห็นกลุ่ม

ESP8266 Modules แล้วเลือก **NodeMCU 1.0**



ติดตั้งบอร์ด AX-WiFi ลงบนโปรแกรม Arduino

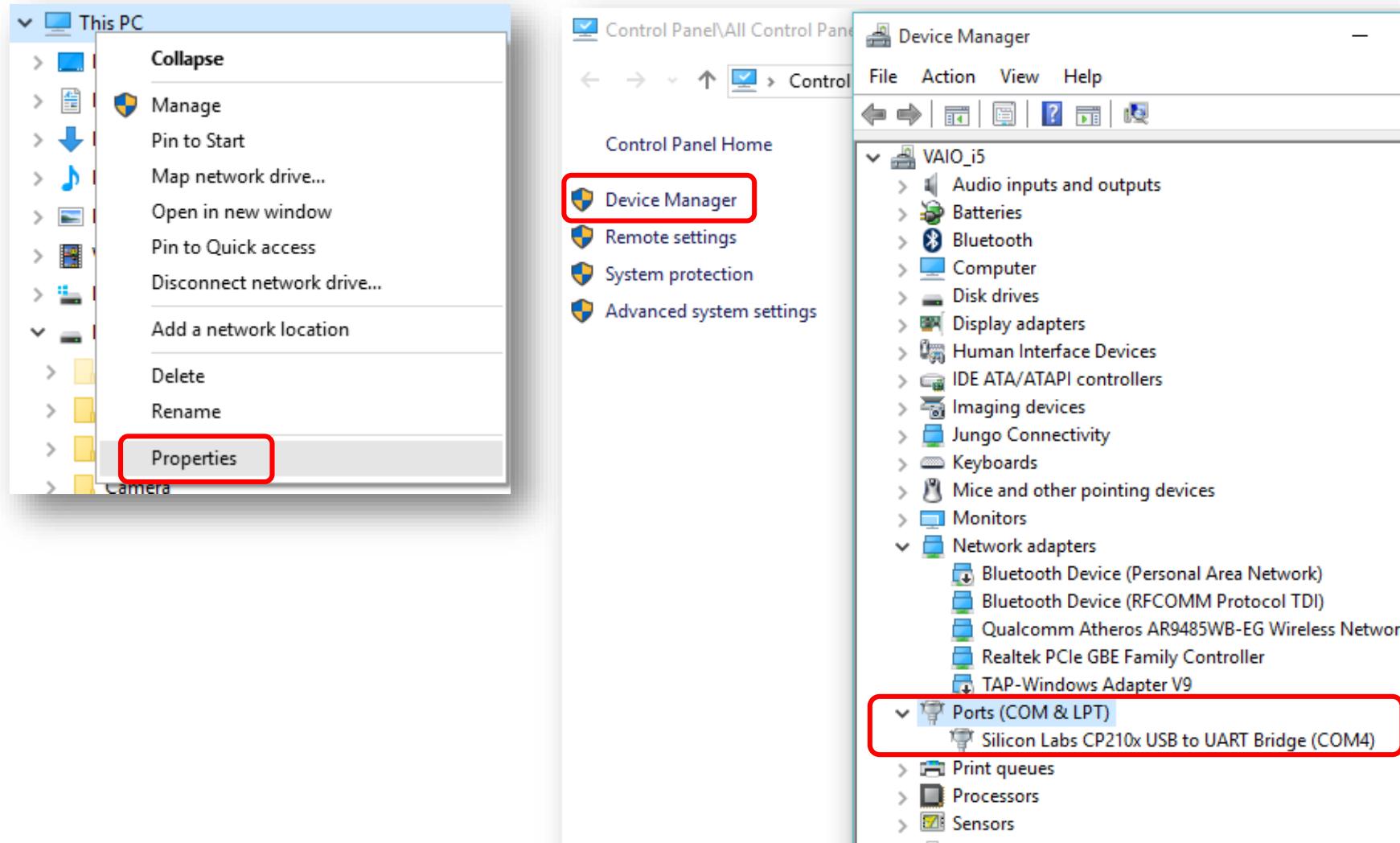
ตรวจสอบการติดตั้ง ด้วยการ **Compile** โดยที่ไม่ต้องเขียนโปรแกรม หากไม่มีอะไรผิดพลาดจะขึ้นข้อความ **Done compiling.**

```
sketch_apr16a | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_apr16a
1 void setup() {
2
3 }
4
5 void loop() {
6
7 }
```

Done compiling.
Sketch uses 246111 bytes (23%) of program storage space. Global variables use 32260 bytes (39%) of dynamic memory.

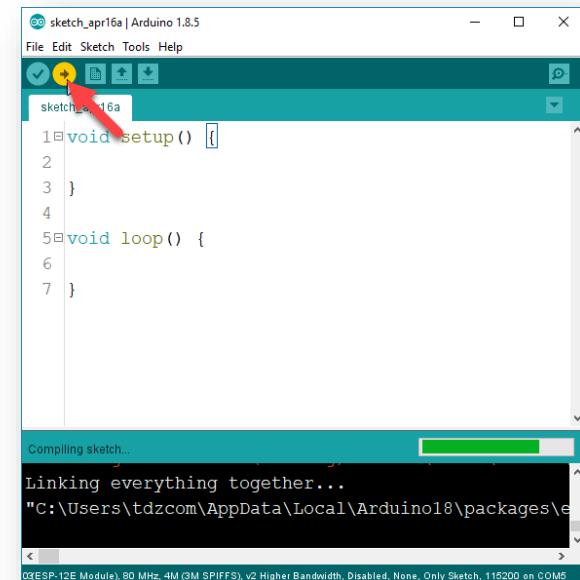
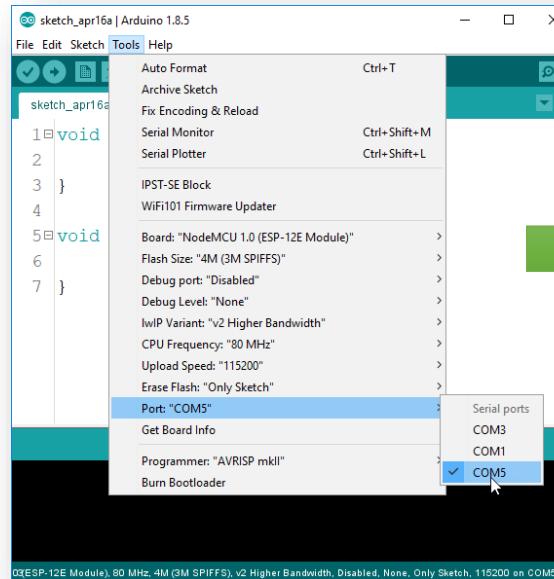
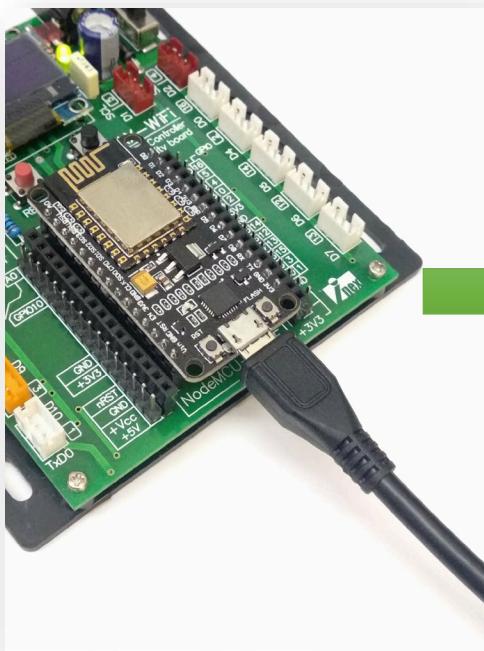
0.0 1.0 (ESP-12E Module), 80 MHz, 4M (1M SPIFFS), v2 Lower Memory, Disabled, None, Only Sketch, 115200 on COM26

ตรวจสอบการเชื่อมต่อ



ติดตั้งบอร์ด AX-WiFi ลงบนโปรแกรม Arduino

ตรวจสอบการโปรแกรมลงบอร์ดต่อสาย NodeMCU กับคอมพิวเตอร์ด้วยสายmicroUSB กำหนด Port ที่เป็นของ NodeMCU ให้เรียบร้อย



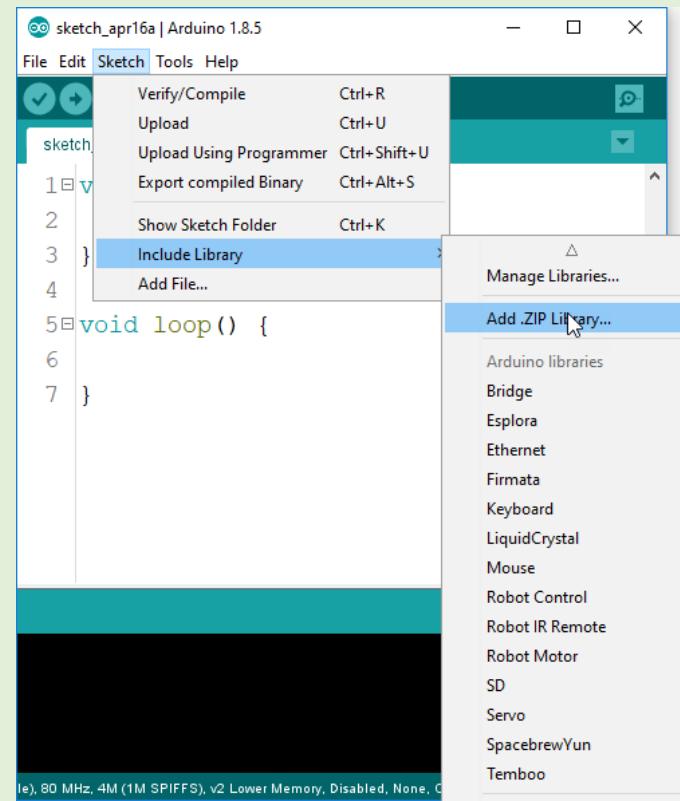
ติดตั้งไลบรารี AX-WiFi

ไปที่ <https://github.com/inexglobal/AXWIFI/archive/master.zip>

เพื่อดownload ไลบรารี



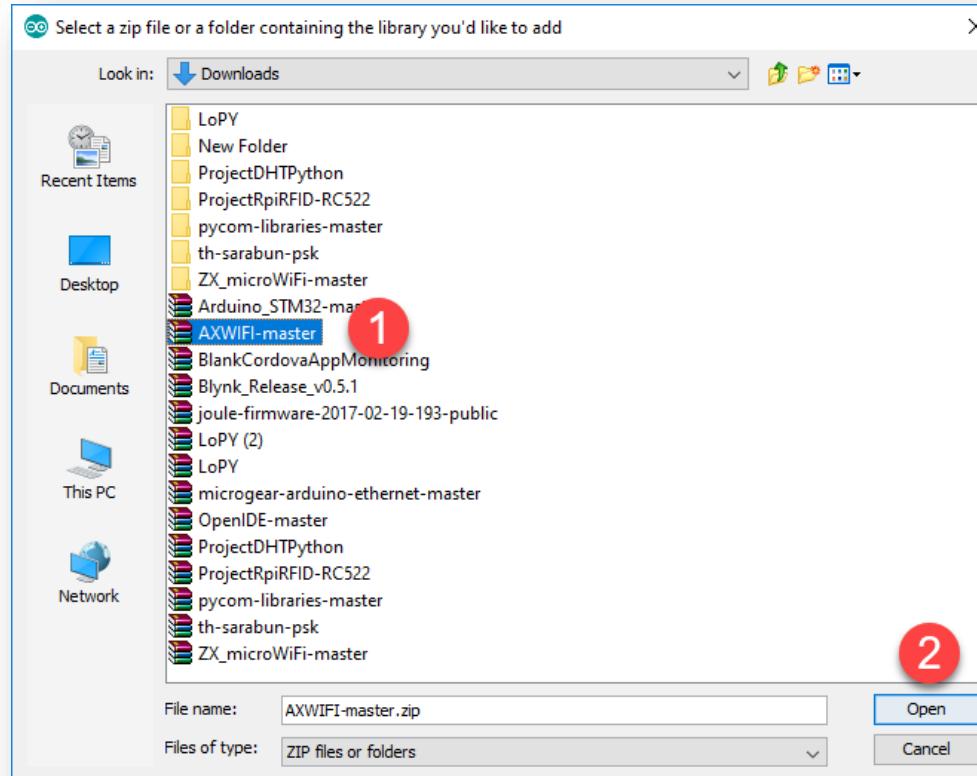
ไปที่ Sketch > Include Library > Add.Zip Library...



ติดตั้งไลบรารี AX-WiFi



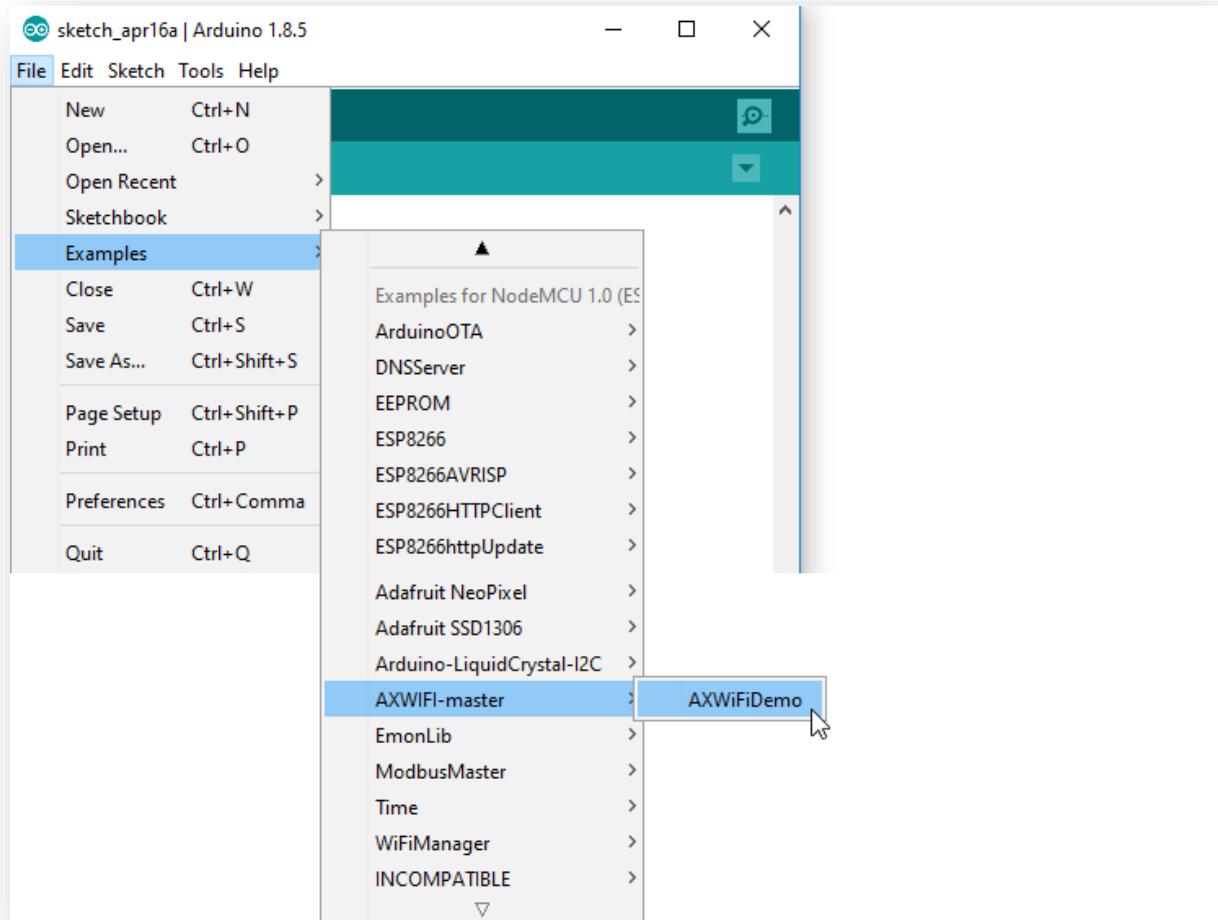
เลือกไฟล์ที่ AXWIFI-master จากนั้นคลิกปุ่ม Open



ทดสอบโปรแกรมลง AX-WiFi



ไปที่แถบเมนู File > Examples > AX-WIFI > AXWIFIDemo



ทดสอบโปรแกรมลง AX-WiFi



กดปุ่ม Upload ถ้าไม่มีอะไรผิดพลาดจะขึ้น Done uploading

AXWiFiDemo | Arduino 1.8.5

File Edit Sketch Tools Help

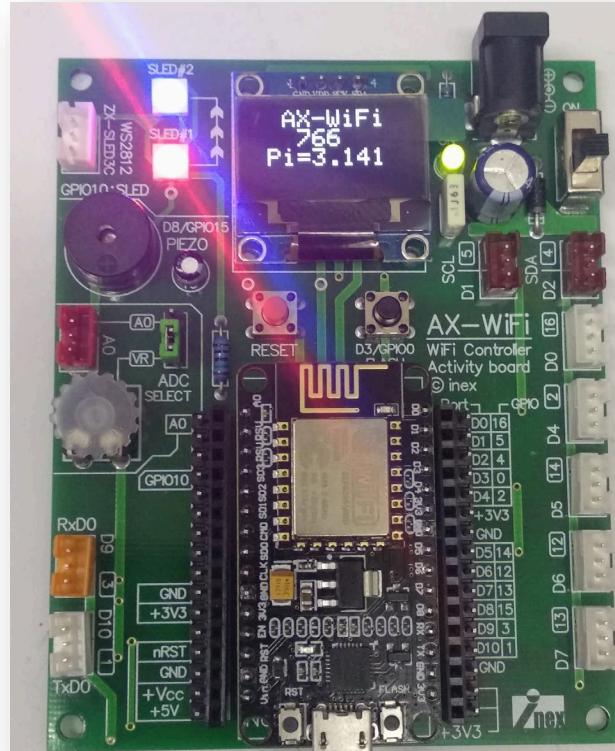
AXWiFiDemo

```
1 #include "AXWIFI.h"
2 int i=0;
3 void setup() {
4     ax.begin();
5     // ax.SledTypeRGB();
6     ax.SledTypeGRB();
7     ax.beep();
8     pinMode(D3, INPUT);
9 }
10
11 void loop() {
12     ax.SledShow(0, 32, 0, 0);
}
```

Done uploading.

```
        setting serial port timeouts to 1000
        flush complete
```

(0), 80 MHz, 4M (3M SPIFFS), v2 Higher Bandwidth, Disabled, None, Only Sketch, 115200 on COM5



ไลบรารี AXWIFI

```
#include <AXWIFI.h>
```

อนุญาติให้ใช้ได้กับทุกคน

ฟังก์ชันที่เกี่ยวกับระบบ:

```
ax.begin();
```

ฟังก์ชันเริ่มต้นการใช้งานไลบรารี AX-WiFi

ໄລບຣაີ່ AX-WiFi ພັກໍ້ຫັນທີ່ເກື່ອງກັບຈອແສດງຜລ

ຈອແສດງຜລທີ່ AX-WiFi ໃຊ້ມື້ນາດ 128×64

ax.OledSetTextSize(int size) ພັກໍ້ຫັນກຳໜດຂນາດຕົວອັກໜຣ
ພາຮາມີເຕຼອຮ

size - ຂນາດຕົວອັກໜຣ

ຄ້າຫາກກຳໜດຂນາດ 1 , 1 ຕົວອັກໜຣຈະໃໝ່ 6×8 ຈຸດ

ຕົວຢ່າງ

ax.OledSetTextSize(1) ;

ໄລບຣაີ່ AX-WiFi ພັກໜັນທີເກື່ອງກັບຈອແສດງຜລ

ax.OledSetText(uint8_t l, uint8_t c,char *fmt, ...)

ພາຮາມີເຕືອ້ຮ

l – ຄືອຕຳແໜ່ງບຣທິດ

c – ຄືອຕຳແໜ່ງຕົວອັກ່າຍ

fmt – ຂໍອຄວາມທີ່ຕ້ອງກາຣແສດງຮົມຄິງຮ້າສທີ່ໃຊ້ກໍານົດຮູປແບບ

ພິເສະໜ

ຕົວອຢ່າງ

ax.OledSetText(0,2,"AX-WiFi");

ax.OledSetText(2,1,"Pi=% .3f",3.141);

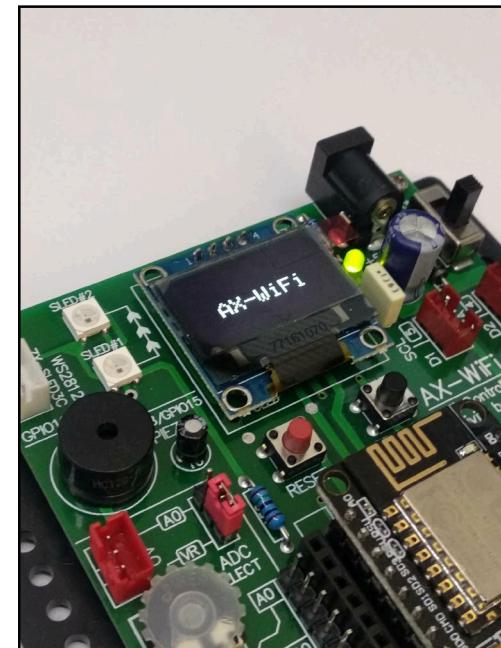
ໄລບຣაີ່ AX-WiFi ພຶກໜັນທີເກື່ອງກັບຈອແສດງຜລ

ax.OledShow()

ພຶກໜັນທີໃຊ້ແສດງຜລອອກທາງໜ້າຈອເມື່ອກຳຫນດຂໍ້ຄວາມເສົ້າຈແລ້ວ

ຕ້ວອຍ່າງ

```
#include "AXWIFI.h"
void setup() {
    ax.begin();
    ax.OledSetTextSize(2);
    ax.OledSetText(2,2,"AX-WiFi");
    ax.OledShow();
}
void loop() {
```



ໄລບຣາີ່ AX-WiFi ພຶກໜັນທີ່ເກີຍກັບ LED

`ax.SledTypeGRB()`

ພຶກໜັນກຳຫນດຮູບແບບສື່ເຮືອງຈາກ ສື່ເຂີຍວ, ສື່ແດງ, ສິນໍ້າເຈີນ

`ax.SledTypeRGB()`

ພຶກໜັນກຳຫນດຮູບແບບສື່ເຮືອງຈາກ ສື່ແດງ, ສື່ເຂີຍວ, ສິນໍ້າເຈີນ

`ax.SledShow(uint16_t n, uint8_t r, uint8_t g, uint8_t b)`

ພຶກໜັນກຳຫນດແລະແສດງສື່ອອກ LED

ພາຣາມີເຕອຮ້

`n` – ລຳດັບ LED

`r , g , b` – ຄ່າສື່ ແດງ,ເຂີຍວ,ນໍ້າເຈີນ 0-255

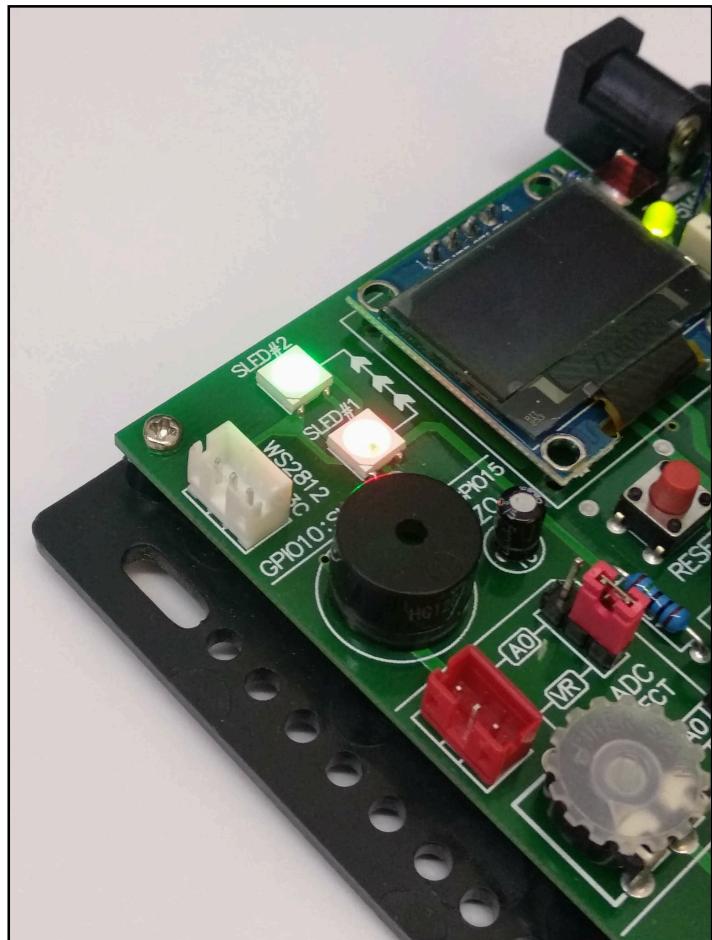
ໄລບຣາີ່ AX-WiFi ພຶກໜັນທີເກີຍກັບ LED

ຕົວຢ່າງ

LED ດວງທີ 1 ແສດງສື່ແດງ 12 (5 %)

LED ດວງທີ 2 ແສດງສື່ເຂີຍວ 12 (5 %)

```
#include "AXWIFI.h"
void setup() {
    ax.begin();
    ax.SledTypeGRB();
    ax.SledShow(0,12,0,0);
    ax.SledShow(1,0,12,0);
}
void loop() {
}
```

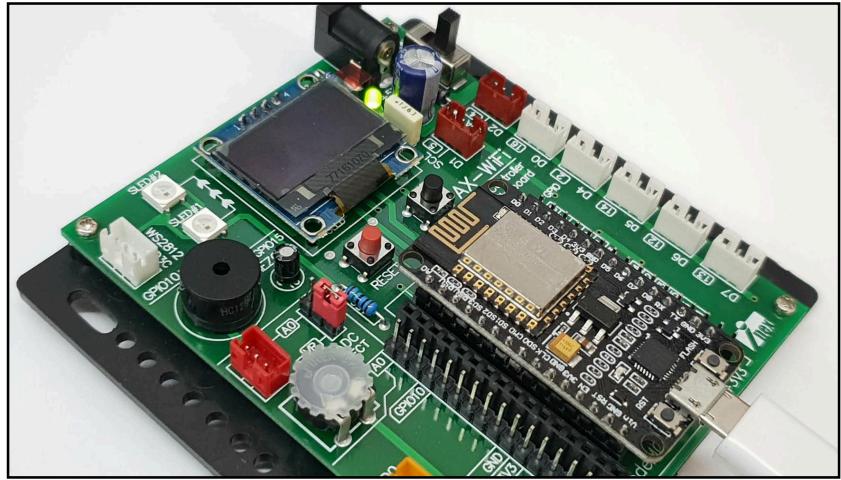


ໄລບຣაີ່ AX-WiFi ພັກໍ້ຈັນທີ່ເກື່ອງຂອງລຳໂພງ

ax.beep();

ພັກໍ້ຈັນກຳຫນດເສີຍອອກລຳໂພງ

```
#include "AXWIFI.h"
void setup() {
    ax.begin();
    pinMode(D3, INPUT);
}
void loop() {
    if(!digitalRead(D3)) {
        ax.beep();
    }
}
```



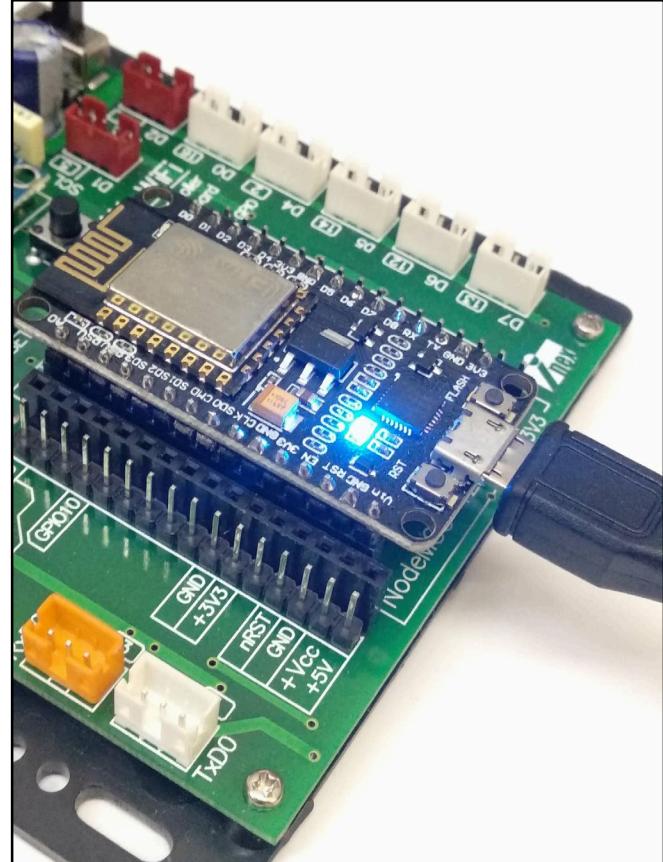
ກົດສວິຕ໌ສື່ດຳ ບນບອົດ AX-WiFi ຈະມີ
ເສີຍດັ່ງອອກຈາກລຳໂພງ

โปรแกรมไฟกระพริบ

```
#include "AXWIFI.h"
void setup() {
    pinMode(D0,OUTPUT) ;
    ax.begin();
}
void loop() {
    digitalWrite(D0,1) ;
    delay(500) ;
    digitalWrite(D0,0) ;
    delay(500) ;
}
```

เมื่อโปรแกรมเริ่มทำงาน LED บน

NodeMCU กระพริบ



LED ตัวแหน่ง D0/16

รูปแบบ Arduino

```
void setup()
```

```
{
```

สำหรับกำหนดค่า เกิดขึ้นครั้งเดียว

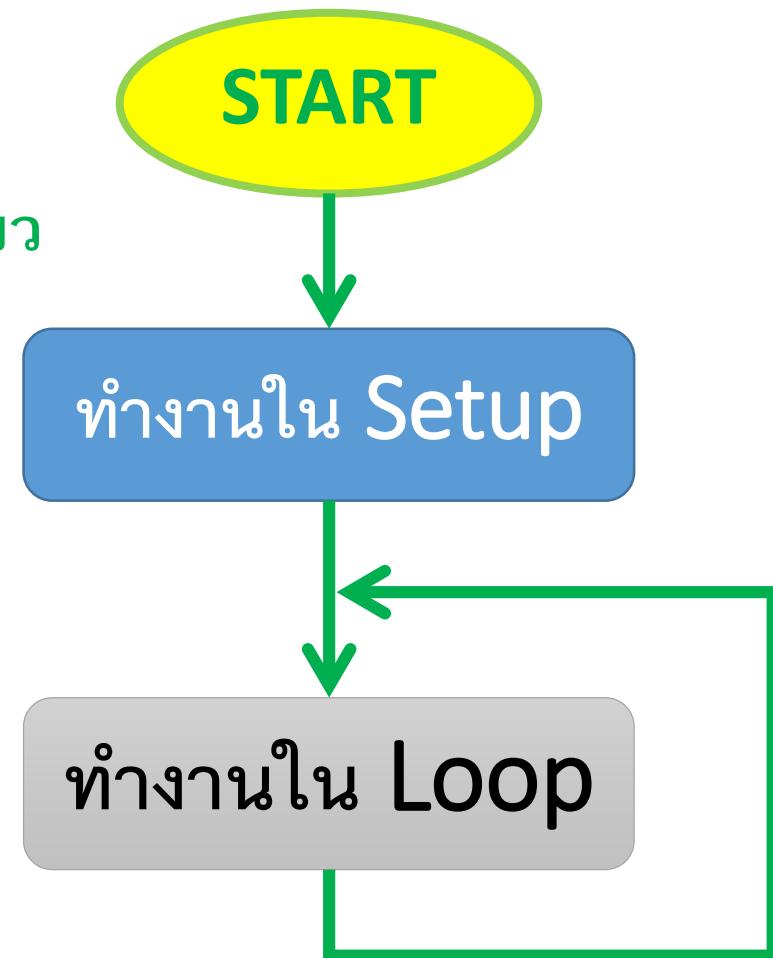
```
}
```

```
void loop()
```

```
{
```

โปรแกรมหลักทำงานต่อเนื่อง

```
}
```



กำหนดขาพอร์ตเป็นอินพุตหรือเอาต์พุต

pinMode (pin , Direction) ;

pin = ตำแหน่งขา D0-D10

Direction = OUTPUT , กำหนดขาเป็นเอาต์พุต

Direction = INPUT , กำหนดขาเป็นอินพุต

Direction = INPUT_PULLUP , กำหนดขาเป็นอินพุต มีตัว
ต้านทานพูลอัพในตัว

pinMode (D0 , 1) ; // กำหนดขาเป็นเอาต์พุต

รูปแบบ คำสั่ง digitalWrite

digitalWrite (pin , Logic) ;

pin = ตำแหน่งขา

Logic = HIGH , 1 เอาต์พุตlogic "1"

Logic = LOW , 0 เอาต์พุตlogic "0"

ตัวอย่าง

digitalWrite (D0 , HIGH) ;

รูปแบบ คำสั่ง delay

คำสั่งนั่งเวลาค้างที่บรรทัดนี้นานเท่าค่า DelayTime

delay (DelayTime) ;

DelayTime ค่าเวลาหน่วงน่วยเป็นมิลลิวินาที

ตัวอย่าง

delay (2000) ;

ค้างที่นี่นาน 2 วินาที

ตัวแปรที่ใช้ใน Arduino

byte	0-255 (unsigned char)
word	0-65535 (unsigned int)
boolean	0-1 True False
int	-32768 ถึง 32767
char	-128 ถึง 127
float	-3.4 x 10 ³⁸ ถึง 3.4 x 10 ³⁸

ข้อมูลเพิ่มเติมจาก reference

ส่งค่าออกเอาต์พุตแบบ อะนาลอก

analogWrite (pin , PWM) ;

pin = ตำแหน่งขา D0-D10

PWM = ค่า DutyCycle 0-1023 แทน 0-100 %

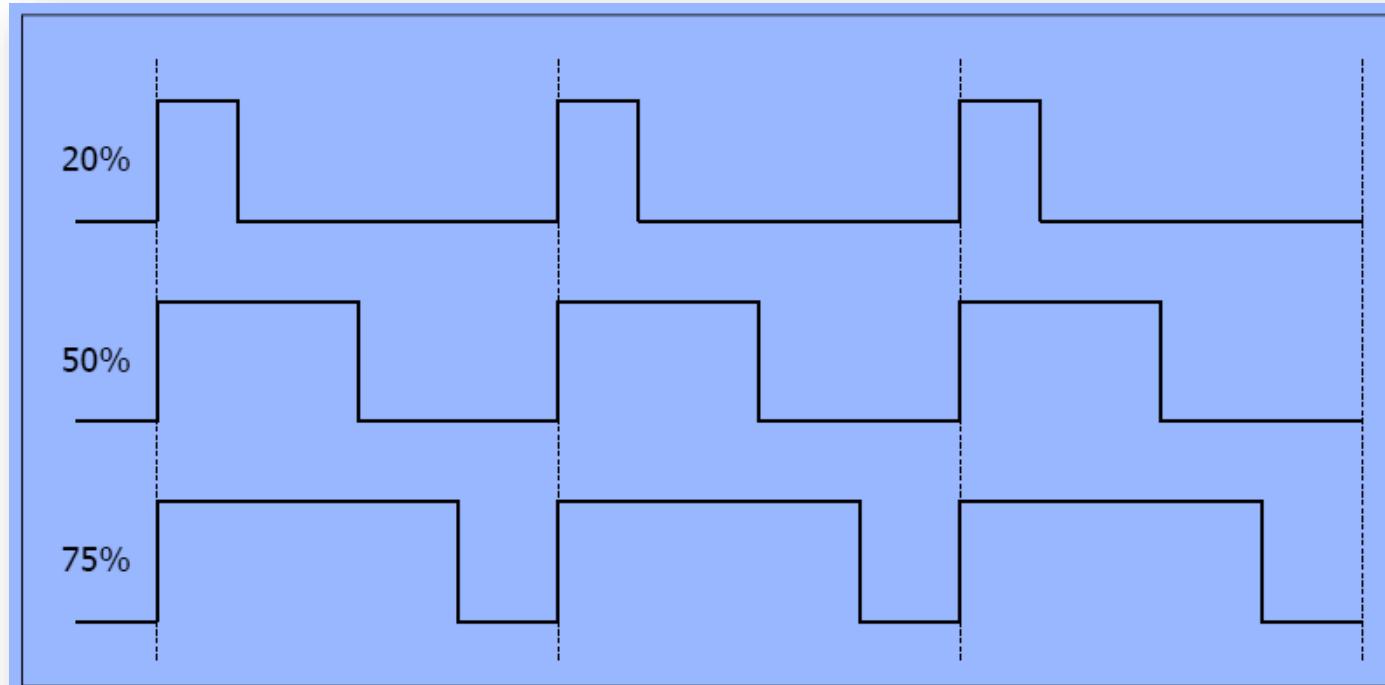
ความถี่ PWM ประมาณ 1kHz

ปรับค่าความถี่

analogWriteFreq (Freq) ;

Freq = 18-60,000 Hz

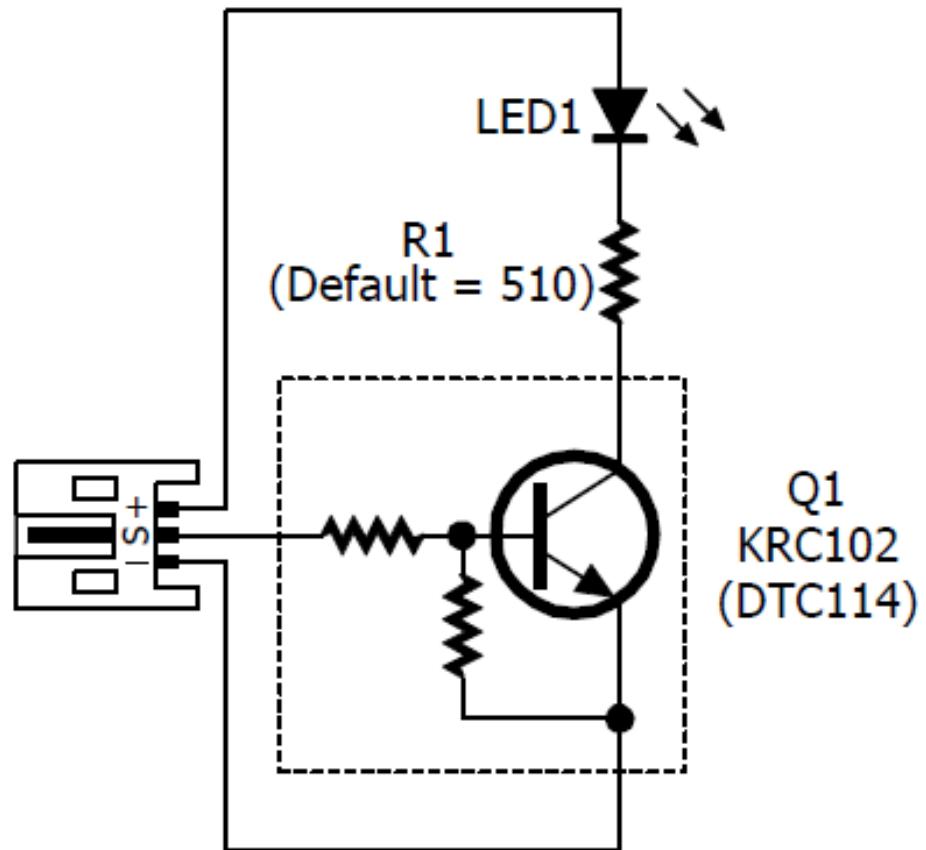
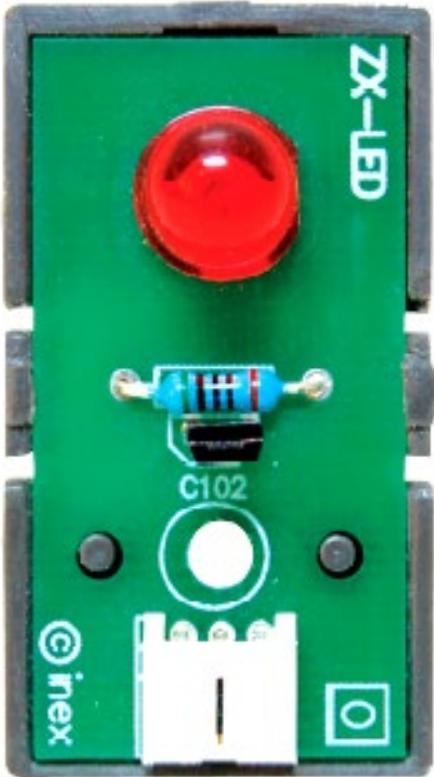
PWM : Pulse Width Modulator



analogWrite (pin , DutyCycle) ;
ความถี่ Fix 1kHz

NodeMCU PWM ทุกขา

ZX-LED

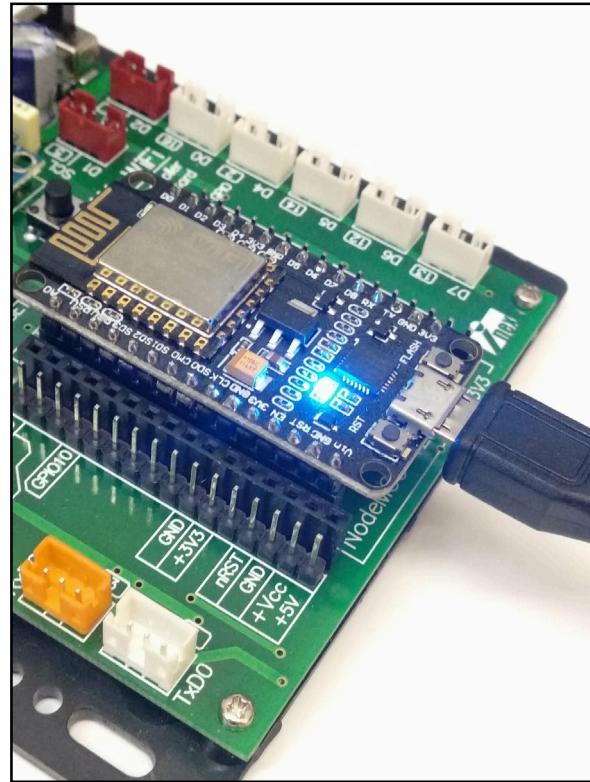


ทดสอบเปลี่ยนดิวตี้ไซเกิล

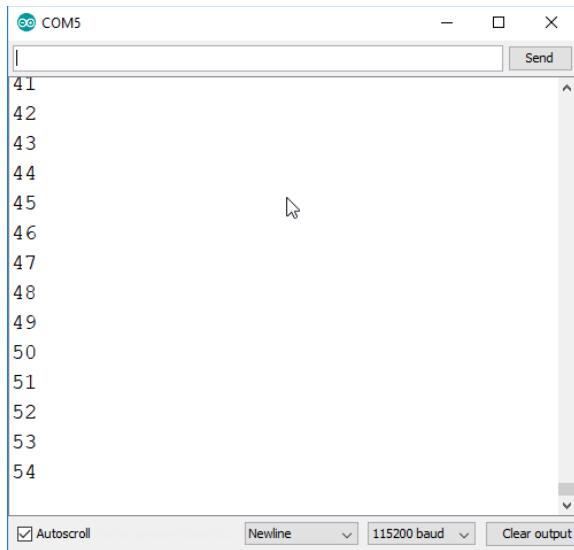
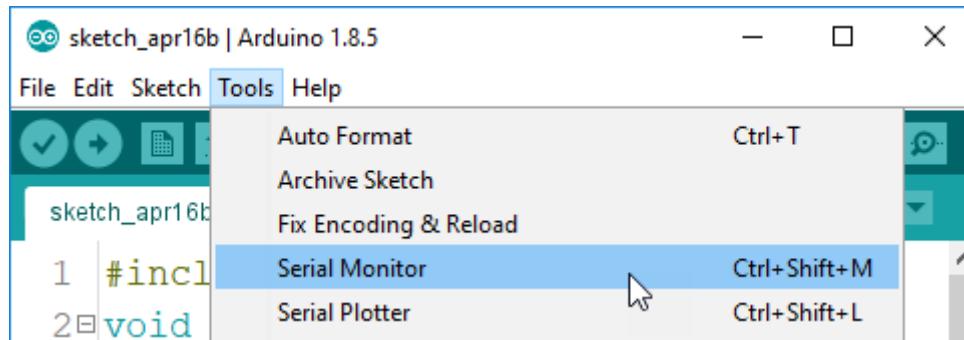
```
#include "AXWIFI.h"

void setup() {
    ax.begin();
    Serial.begin(115200);
    analogWrite(D0, 127);
}

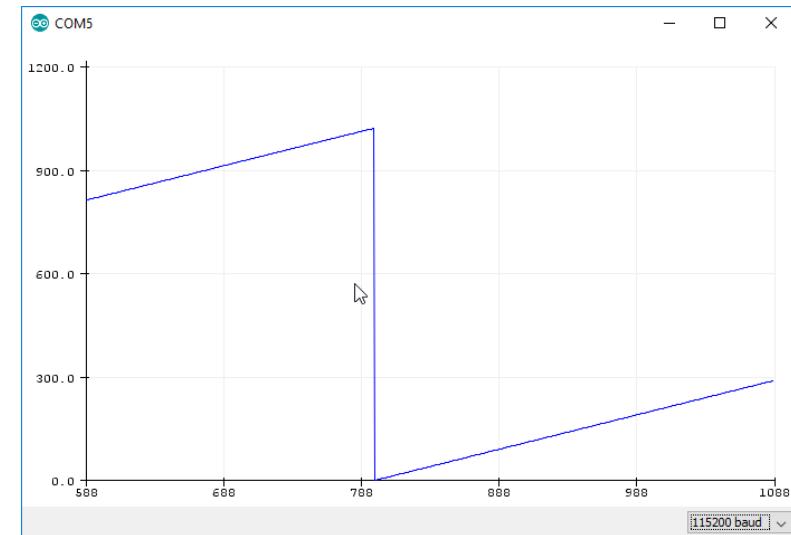
void loop() {
    for (int i=0;i<1023;i++){
        analogWrite(D0,i);
        Serial.println(i);
        delay(10);
    }
}
```



ผลลัพธ์เปลี่ยน divisor ใช้เกล



Serial Monitor



Serial Plotter

การรับส่งข้อมูล Serial กับคอมพิวเตอร์

Serial.begin(BAUD) เริ่มต้นใช้งานการสื่อสาร
BAUD คือบอเดตที่ใช้ในการสื่อสาร

Serial.write(1Byte) ส่งข้อมูล 1 ไบต์, หรือข้อความง่าย ๆ

Serial.print(BAUD) ส่งข้อความพร้อมพารามิเตอร์ต่าง ๆ

Serial.println(BAUD) ส่งข้อความพร้อมพารามิเตอร์ต่าง ๆ และขึ้นบรรทัดใหม่

Serial.print(78, BIN) gives "1001110"

Serial.print(78, OCT) gives "116"

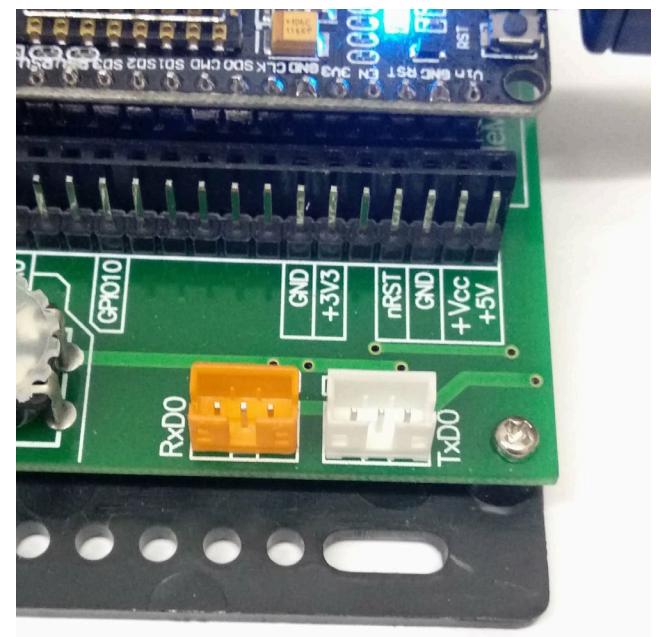
Serial.print(78, DEC) gives "78"

Serial.print(78, HEX) gives "4E"

Serial.println(1.23456, 0) gives "1"

Serial.println(1.23456, 2) gives "1.23"

Serial.println(1.23456, 4) gives "1.2346";



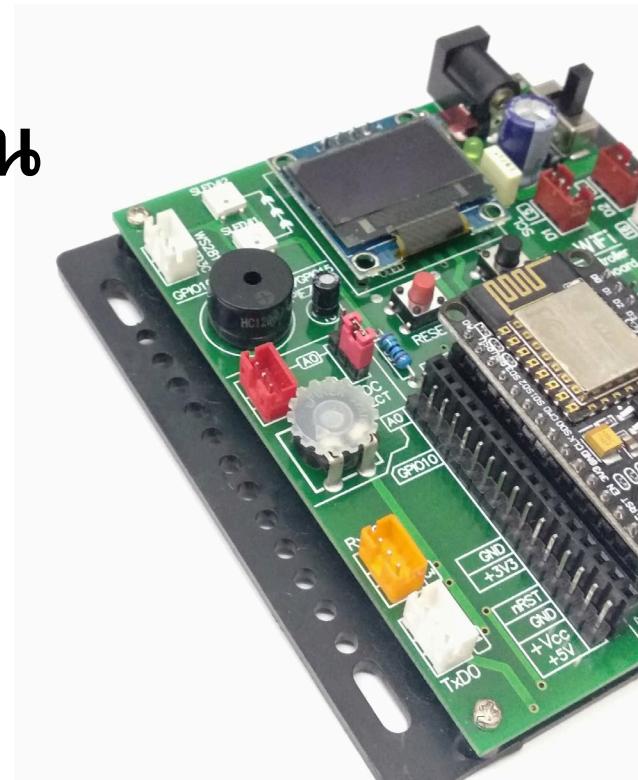
การอ่านค่าอ่อนาล็อก

x=analogRead (pin);

x = ตัวแปร int เพื่อนำมารับค่า

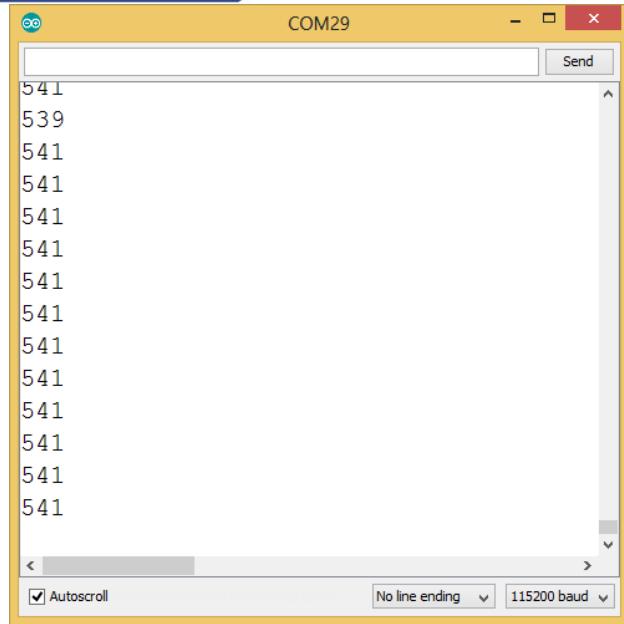
0-1023 (10 bit)

pin = ขาอินพุตอ่อนาล็อก A0 เท่านั้น



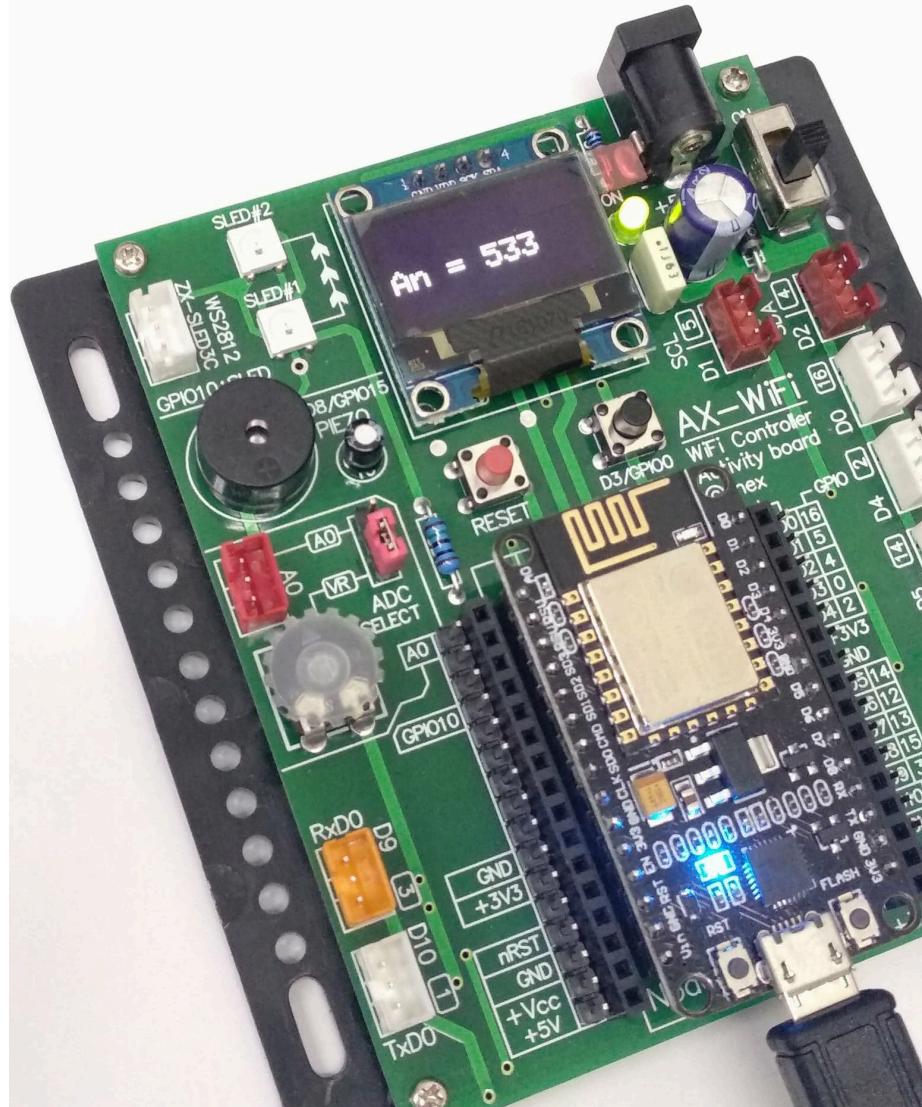
ทดสอบเปลี่ยนความตื้นใช้เคลื่อนที่ด้วย VR

```
void setup0{  
    Serial.begin(115200);  
}  
  
void loop0{  
    int anVal = analogRead(A0);  
    analogWrite(D0,anVal);  
    Serial.println(anVal);  
    delay(10);  
}
```



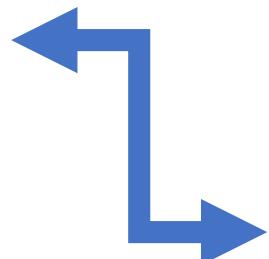
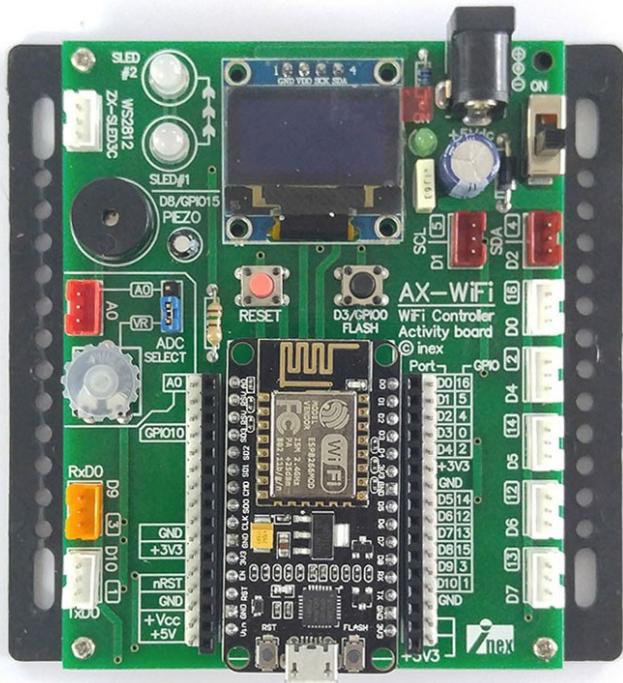
แสดงออกบนหน้าจอ OLED

แบบทดสอบ



```
#include "AXWIFI.h"
void setup() {
ax.begin();
ax.OledSetTextSize(2);
Serial.begin(115200);
}
void loop() {
int anVal = analogRead(A0);
ax.OledSetText(2, 0, "An = %d ", anVal);
ax.OledShow();
analogWrite(D0, anVal);
Serial.println(anVal);
delay(10)
}
```

การใช้งาน AX-WiFi เชื่อมต่อ App Blynk



App Blynk



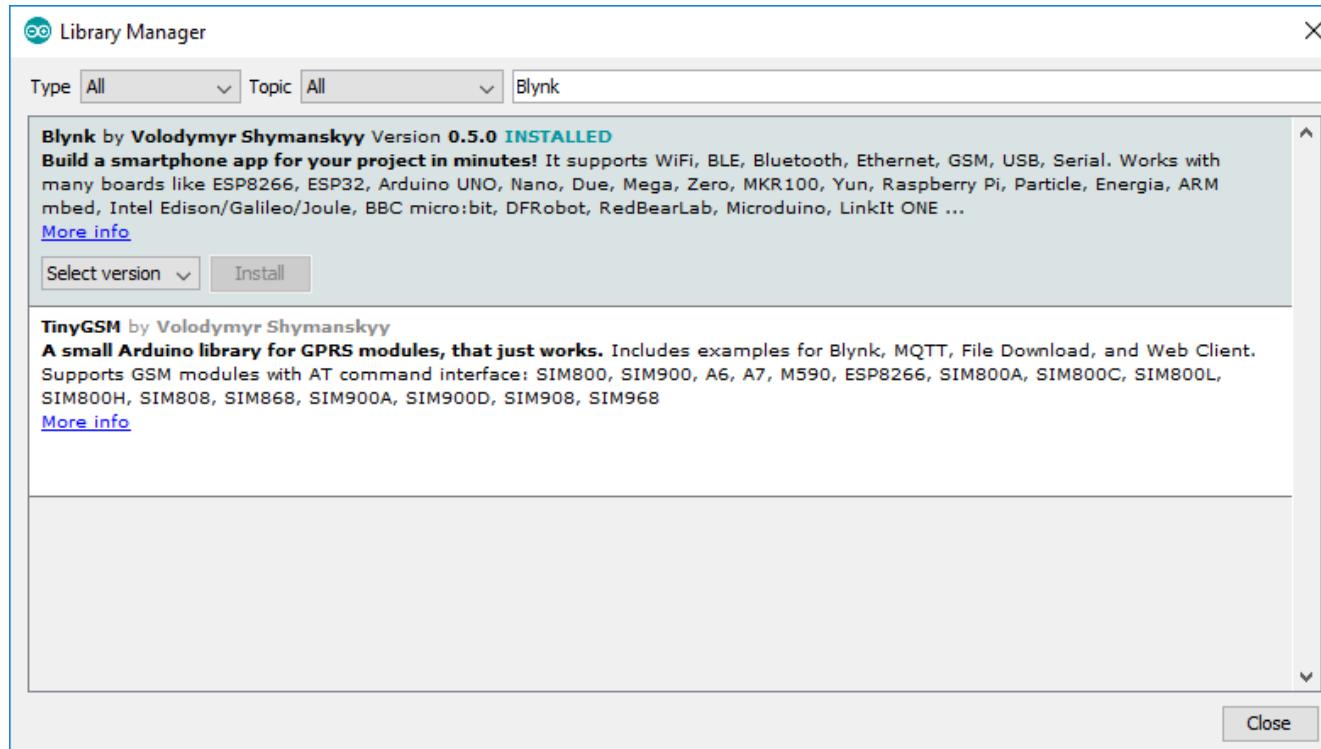
Blynk เป็นอีกหนึ่งผู้ที่ให้บริการเกี่ยวกับ IoT ที่มาพร้อมแอปพลิเคชันบนสมาร์ทโฟนสามารถสร้างหน้าควบคุมได้ง่าย ให้ใช้ฟรี 2000 พลังงานจะลดลงเลื่อย ๆ เมื่อเรียกใช้ Widget แต่ละ Widget จะใช้พลังงานไม่เท่ากัน



ติดตั้งไลบรารี Blynk



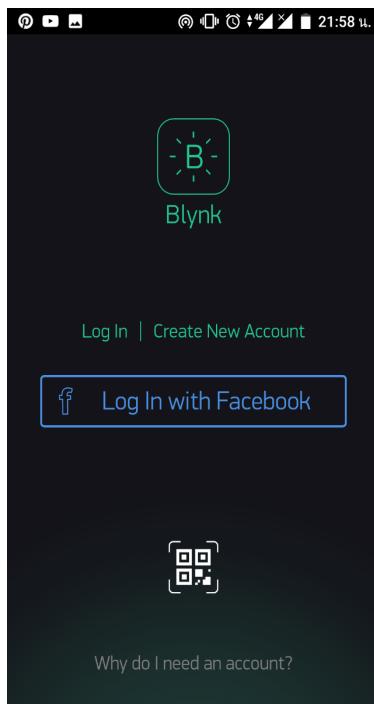
เปิดโปรแกรม Arduino ไปที่ແນ້ນ Sketch > Include Library > Manager Libraries... พิมพ์ข้อความว่า Blynk ดังรูปจากนั้นกดปุ่ม Install เพื่อติดตั้งไลบรารี



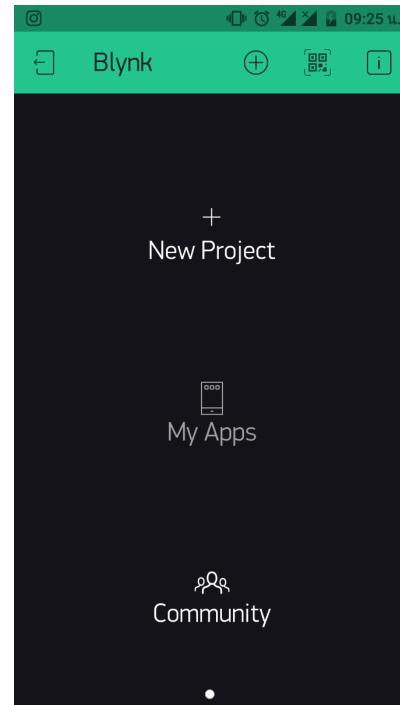
ขั้นตอนการลงทะเบียนใช้งาน Blynk



เปิด App Blynk เพื่อลงทะเบียนเข้า
ใช้งาน ด้วย Facebook หรือ Email
ผู้ลงทะเบียนจำเป็นจะต้องมี Email



เลือก New Project



ขั้นตอนการลงทะเปลี่ยนใช้งาน Blynk



3. ตั้งค่า Project เมื่อเสร็จแล้วให้คลิกปุ่ม Create

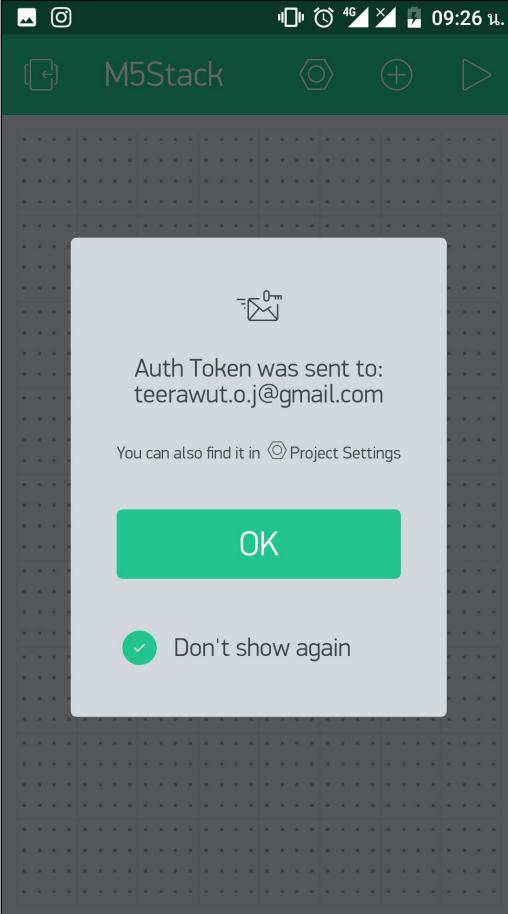
ตั้งชื่อ Project = IoT (ตามต้องการ)

Choose Device = ESP8266

Connection Type = WiFi

Theme = (Dark:เข้ม,Light:สว่าง)

ขั้นตอนการลงทะเบียนใช้งาน Blynk



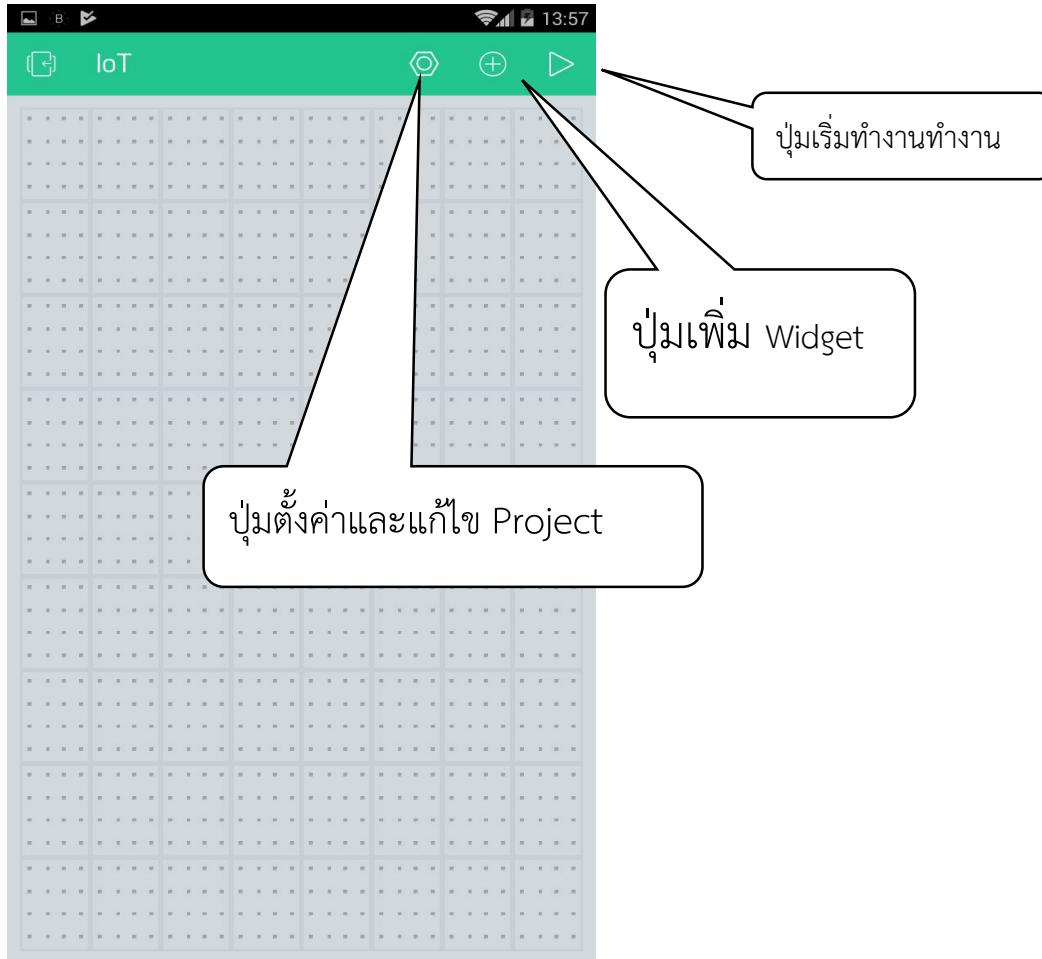
A screenshot of the M5Stack mobile application interface. At the top, there are standard Android status icons (signal strength, battery, time). Below that, the title "M5Stack" is displayed next to a back arrow, a gear icon for settings, a plus sign for new projects, and a right-pointing arrow. The main content area shows a light gray pop-up window. Inside the window, there is a small envelope icon with a '0' next to it. Below the icon, the text "Auth Token was sent to:" is followed by an email address "teerawut.o.j@gmail.com". Underneath this, smaller text says "You can also find it in Project Settings". At the bottom of the window is a large green button labeled "OK". Below the "OK" button is a checkbox with a checkmark and the text "Don't show again". The background of the app shows a dark grid pattern, likely the workspace for project development.

4. คลิกที่ปุ่ม OK เพื่อส่ง Auth Token ไปยัง Email
Auth Token จะต้องนำไปใส่ไว้ในโปรแกรมที่เราได้
เขียนบนโปรแกรม Arduino

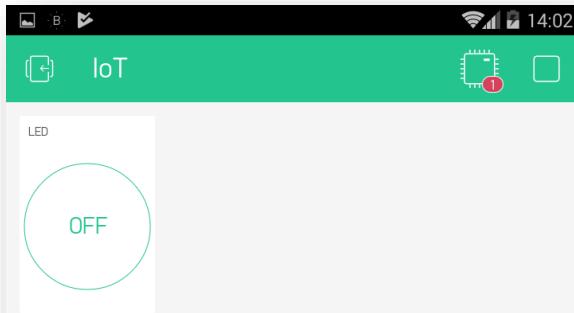
ขั้นตอนการลงทะเปลี่ยนใช้งาน Blynk



5.รายละเอียดปุ่มที่สำคัญต่อการสร้างหน้าควบคุม

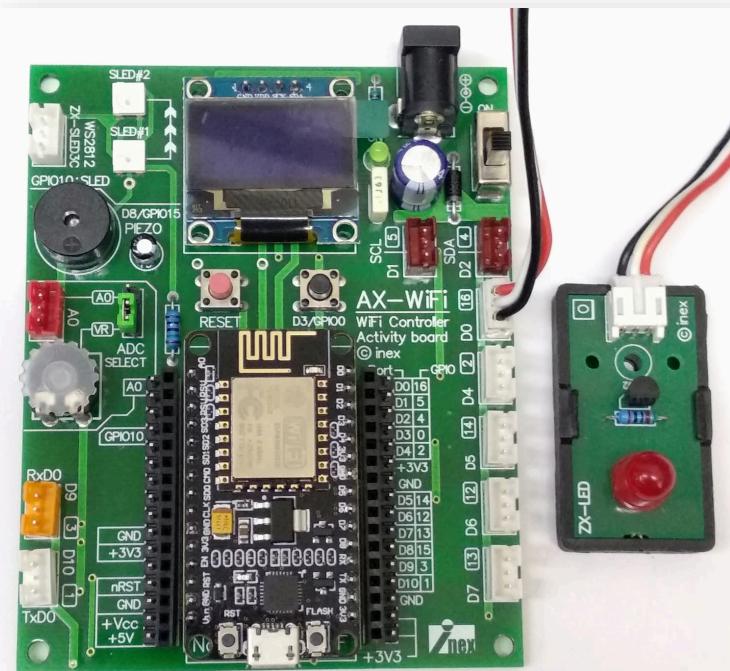


ตัวอย่างการเปิด-ปิด LED ด้วย Blynk



การต่อวงจร

ต่อ ZX-LED เข้ากับ AX-WiFi ให้ต่อสายสัญญาณ
เข้ากับขา GPIO16 หรือ D0

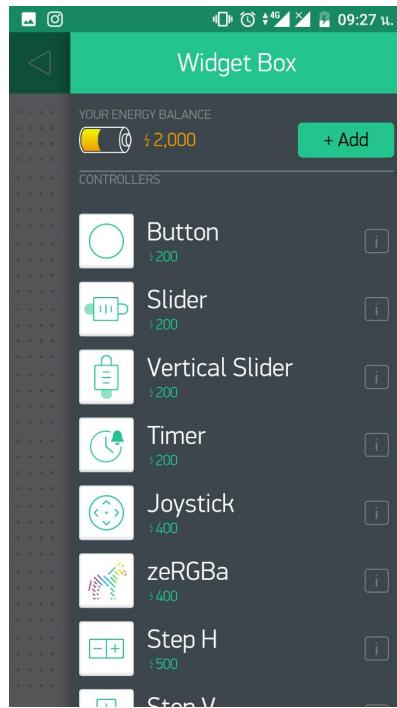


ตัวอย่างการเปิด-ปิด LED ด้วย Blynk

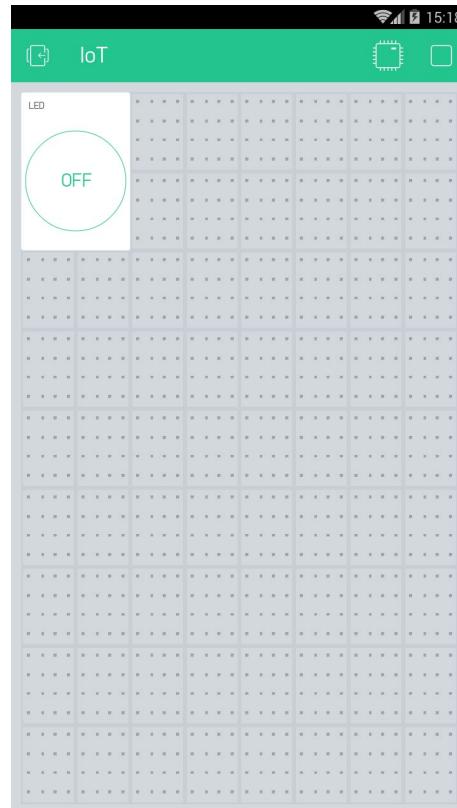


การออกแบบหน้าควบคุม Blynk

1. คลิกที่ปุ่ม(เพิ่ม : +) เพื่อเลือก Widget
จากนั้นเลือก Button



2. จากนั้นคลิกที่ Button เพื่อตั้งค่า



ตัวอย่างการเปิด-ปิด LED ด้วย Blynk



การออกแบบหน้าควบคุมบน Blynk

The screenshot shows the 'Button Settings' screen in the Blynk app. It displays a digital button component labeled 'BUTTON'. Below it, there is an 'LED' section with a green switch icon. Under 'OUTPUT', the pin 'GP16' is selected. A logic level switch is set to '0' (LOW). Under 'MODE', the switch is set to 'PUSH'. In the 'ON/OFF LABELS' section, the 'ON' label is 'ON' and the 'OFF' label is 'OFF'. At the bottom right, there is a red 'Delete' button.

3. กำหนดตำแหน่งขาที่ต้องการ On - Off

ตั้งชื่อปุ่ม : ตามต้องการ

Output : GP16

Mode : PUSH

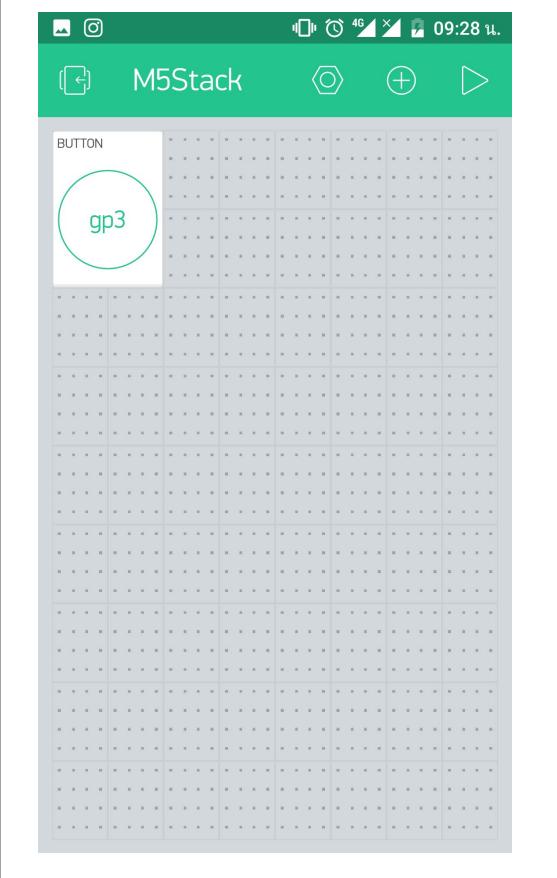
จากนั้นกดปุ่มกลับ

ตัวอย่างการเปิด-ปิด LED ด้วย Blynk

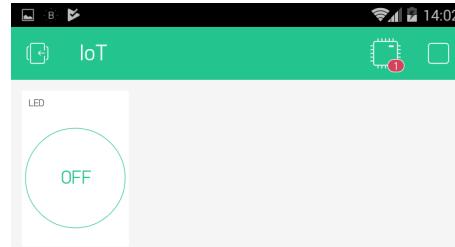


การออกแบบหน้าควบคุมบน Blynk

4. กดปุ่ม Run เพื่อทดสอบ



เมื่อ RUN จะเห็นว่าส่วนแถบด้านบนที่เป็นรูปชิฟจะขึ้น Wasn't online yet เพราะว่า อุปกรณ์ยังไม่ได้เชื่อมต่อกับ Server Blynk



ตัวอย่างการเปิด-ปิด LED ด้วย Blynk

ตัวอย่างโปรแกรม NodeMCUBlynk.ino

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"

char ssid[] = "INEX01_2.4GHz";
char pass[] = "123456789-0";
char auth[] = "Token";

void setup() {
    Serial.begin(9600);
    Blynk.begin(auth, ssid, pass);
    ax.begin();
}

void loop() {
    Blynk.run();
}
```

ตัวอย่างการเปิด-ปิด LED ด้วย Blynk

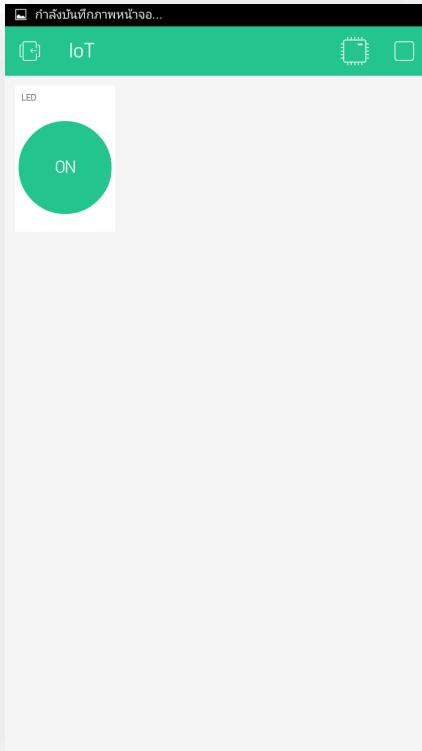
อธิบายเพิ่มเติม

- เรียกใช้ไลบรารี Blynk ใช้คำสั่ง #include <BlynkSimpleEsp8266.h>
- เชื่อมต่อ Wi-Fi และ Server Blynk ใช้คำสั่ง
Blynk.begin(auth,ssid,pass);
- auth คือ token ที่ได้มาจากการสร้าง Project ใช้ยืนยันอุปกรณ์กับ
Server Blynk ให้มาจาก Email
- ปรับปรุงตัวแปรหรือสถานะทั้งหมดของ ESP8266 ใช้คำสั่ง Blynk.run();

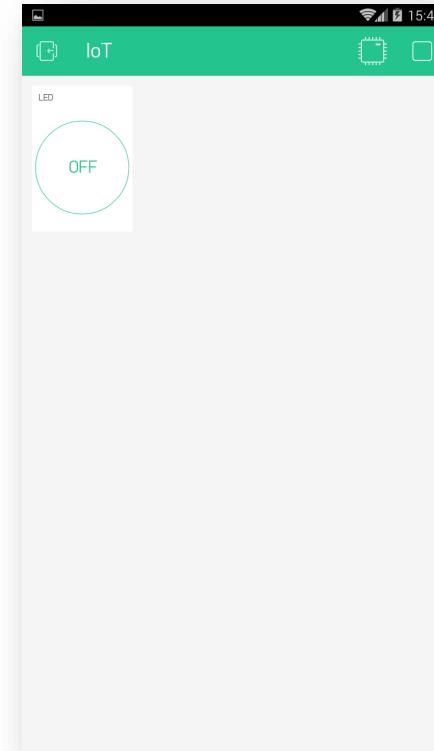
ตัวอย่างการเปิด-ปิด LED ด้วย Blynk

ทดสอบบนสมาร์ทโฟน

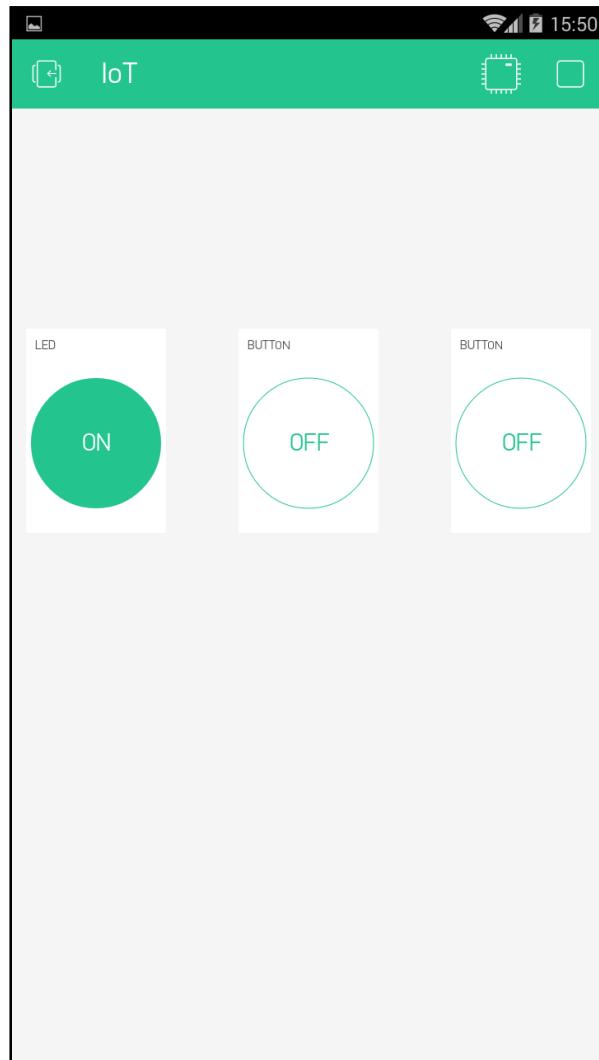
เมื่อกดปุ่มเพื่อ ON สั่งเปิดไฟ



เมื่อกดปุ่มเพื่อ OFF สั่งปิดไฟ



แบบทดสอบเปิด-ปิด LED 3 ดวง ด้วย Blynk



คุณสมบัติของ Blynk



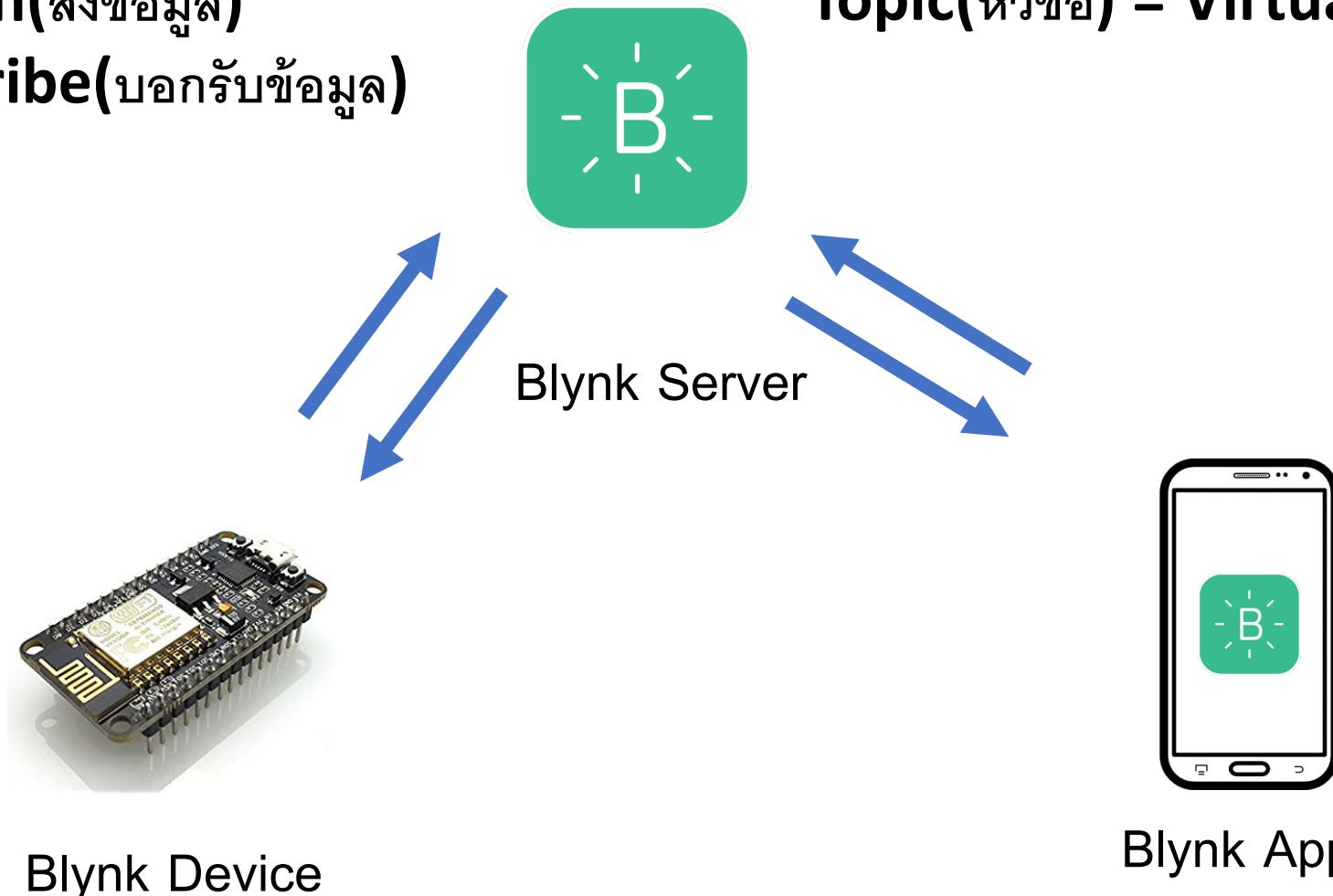
- ใช้proto協定 mqtt
- เชื่อมต่อได้สูงสุด 20 อุปกรณ์(ฟรี)
- Blynk มี topic หรือที่เรียกว่า Virtual pin ให้ใช้ 128 หัวข้อเท่านั้น
- Digital pin จะขึ้นอยู่กับอุปกรณ์ที่ใช้
- Blynk App สามารถเข้าใช้งานพร้อมกันได้มากกว่า 1 เครื่อง
- Blynk App ให้พลังงาน (Energy) 2000 ให้เริ่มต้นการใช้ App (ฟรี)

หลักการทำงานของ Blynk mqtt

Publish(ส่งข้อมูล)

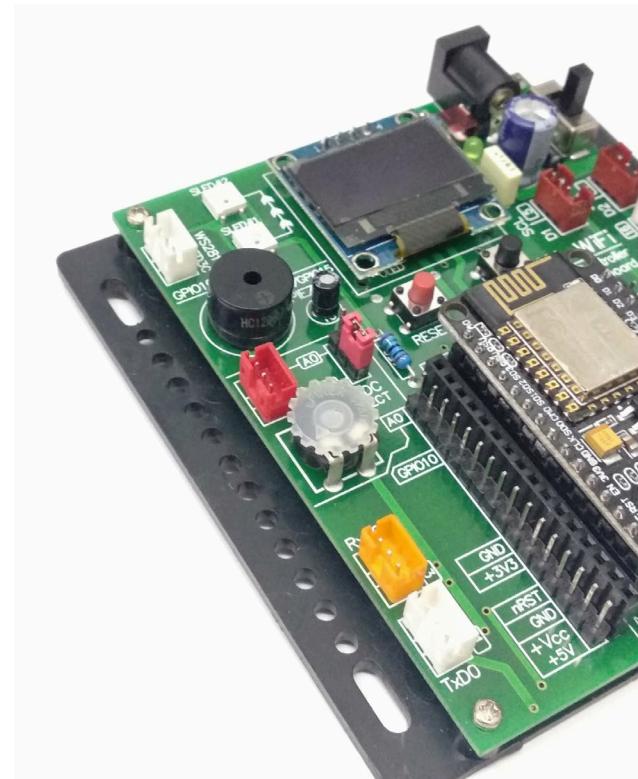
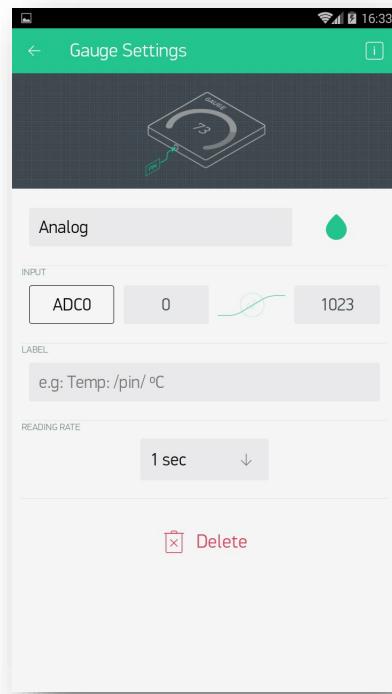
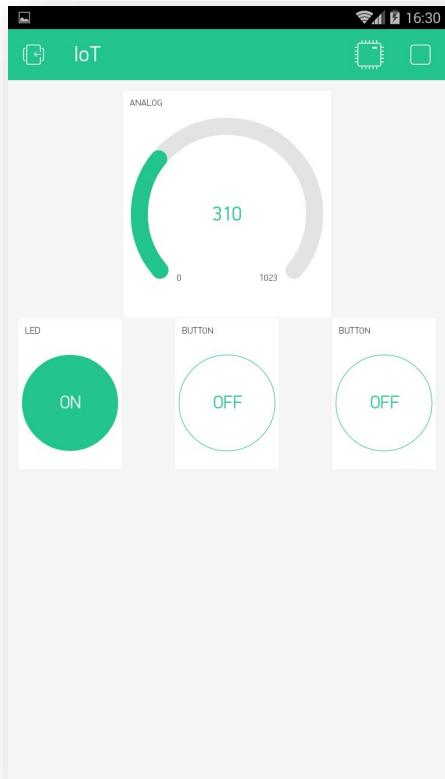
Subscribe(บอกรับข้อมูล)

Topic(หัวข้อ) = Virtual pin



สร้างตัวแสดงผลค่าอุณหภูมิ

เพิ่ม Gauge



ZX-DHT22 วัดความชื้นและอุณหภูมิ



วัดความชื้นสัมพัทธ์

0 ถึง 100%RH

ความผิดพลาด 2-5%RH

ความละเอียด 0.1 %

วัดอุณหภูมิ

-40 ถึง 80 องศาเซลเซียส

ความผิดพลาด +- 0.5 องศาเซลเซียส

ความละเอียด 0.1 องศาเซลเซียส

ดาวน์โหลดไลบรารี DHT22

สามารถดาวน์โหลดได้ที่

https://github.com/adafruit/Adafruit_Sensor

<https://github.com/adafruit/DHT-sensor-library>

จากนั้นแตกไฟล์นามาโฟลเดอร์ที่ได้ไปเก็บไว้ยัง โฟลเดอร์
libraries ที่ได้ติดตั้งไว้ในเครื่อง

การใช้งาน library DHT.h

เพิ่ม Library ในโปรแกรมด้วยคำสั่ง

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

สร้างตัวแปรมาองรับ object DHT ที่ชื่อว่า dht

```
#define DHTTYPE DHT22  
#define DHTPIN D5  
DHT dht(DHTPIN, DHTTYPE, 15);
```

การใช้งาน Methods library DHT.h

เริ่มต้นการทำงาน `dht.begin()`

`dht.read()` สถานะของการอ่านค่า
การคืนพารามิเตอร์

True = อ่านค่าได้

False = อ่านค่าไม่ได้

การใช้งาน Methods library DHT.h

dht.readHumidity()

อ่านค่าความชื้นสัมพัทธ์

การคืนค่าparametr

ค่าความชื้นสัมพัทธ์ % RH

การใช้งาน Methods library DHT.h

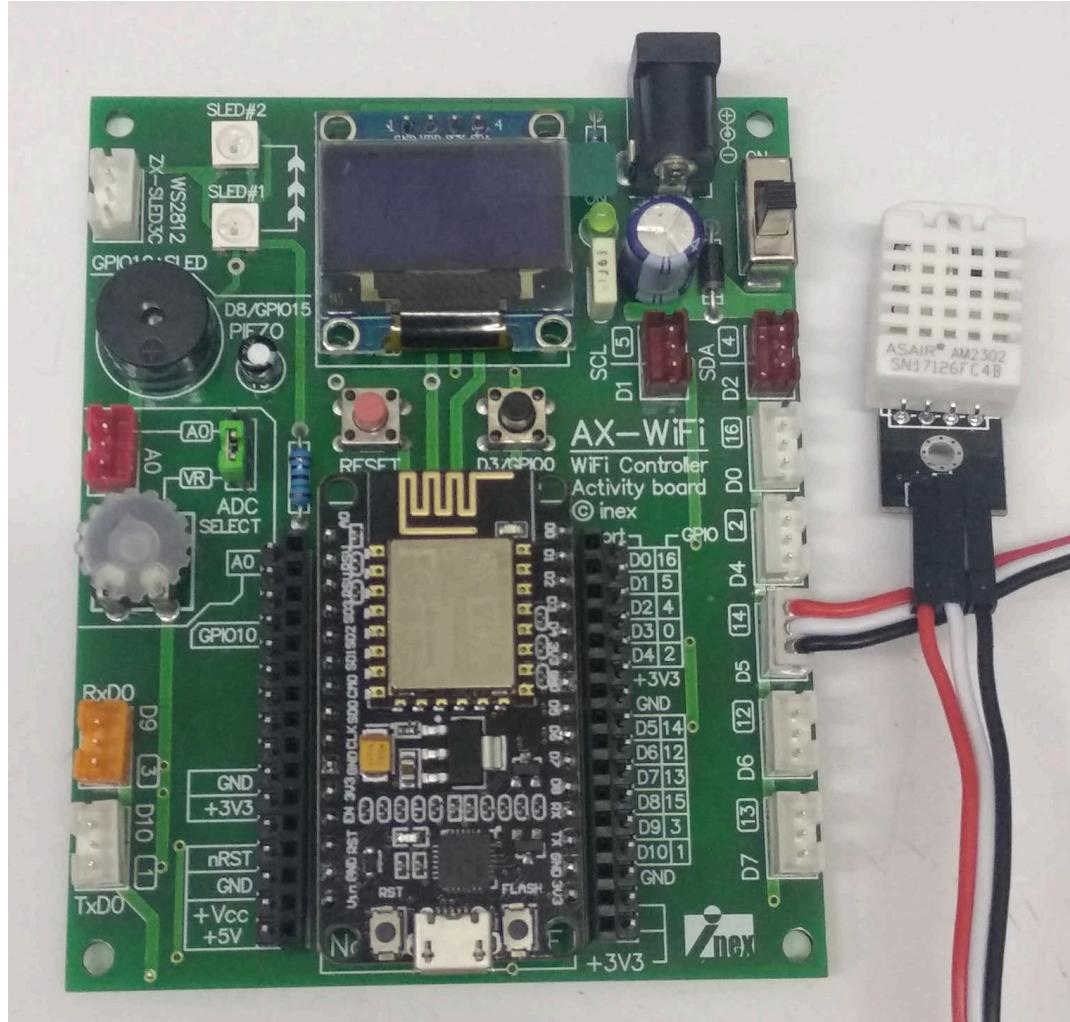
dht.readTemperature()

อ่านค่าอุณหภูมิให้หน่วยเป็นองศาเซลเซียส

การคืนค่าพารามิเตอร์

ค่าอุณหภูมิให้หน่วยเป็นองศาเซลเซียส

ต่อใช้งาน DHT22



ตัวอย่างการใช้งาน DHT22

```
#include "AXWIFI.h"
#include "DHT.h"
DHT dht(D5,DHT22,15);
void Sensor(){
    float h = dht.readHumidity();
    float t = dht.readTemperature();
    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
    Serial.printf("H = %.1f : T = %.1f",h,t);
    Serial.println();
}
void setup(){
    dht.begin();
    ax.begin();
    Serial.begin(115200);
}
void loop(){
    delay(2000);
    Sensor();
}
```

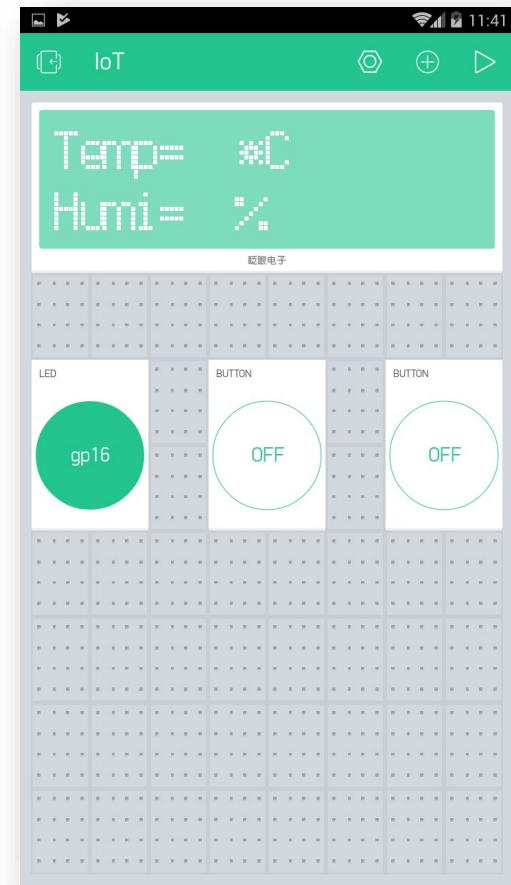
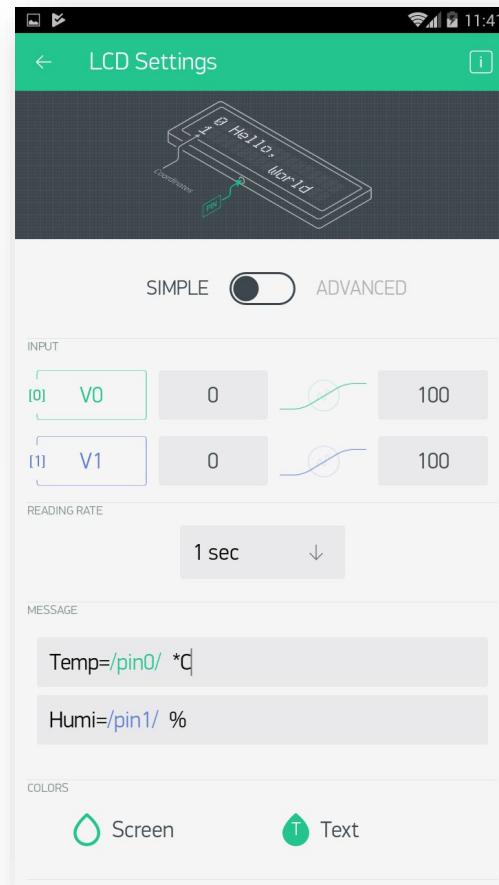
ส่งค่าจาก DHT22 ไปยัง Blynk App

เพิ่ม LCD

Temp=58.500 °C
Humi=25.900 %

V0 = Temp

V1 = Humi



คำสั่งที่ใช้กำหนดรอบเวลาการส่งข้อมูล

ฟังก์ชันที่ใช้กำหนดการทำงานแบบรอรอบเวลา จากตัวอย่างทำงานทุก ๆ 2 วินาที กระโดดไปทำงานที่ฟังก์ชัน CallFuntion

```
BlynkTimer timer;
```

```
timer.setInterval(2000L,CallFuntion);  
timer.run();
```

คำสั่งที่ใช้ส่งข้อมูล

ฟังก์ชันที่ใช้ส่งข้อมูล (publish) ไปยัง Virtual Pin

```
Blynk.virtualWrite(v0, h);  
Blynk.virtualWrite(v1, t);
```

ตัวอย่างส่งค่า DHT22

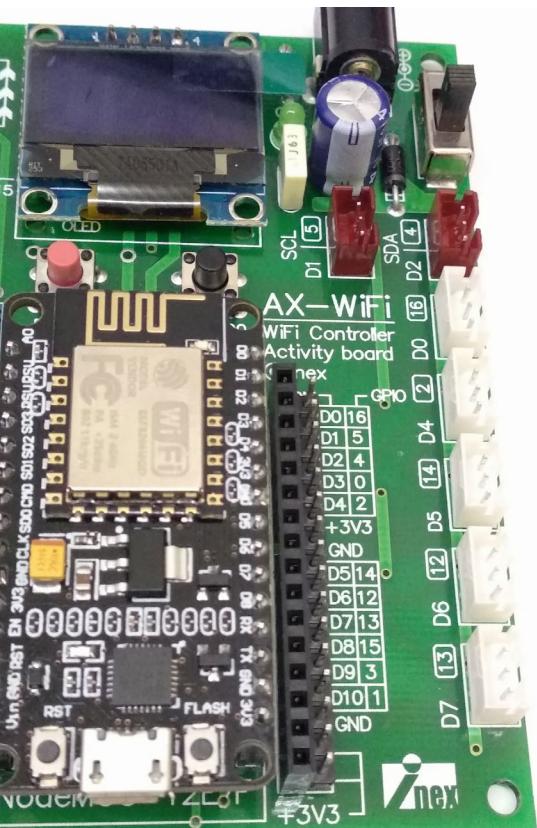
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"
#include "DHT.h"
DHT dht(D5, DHT22, 15);
BlynkTimer timer;
char ssid[] = "INEX01_2.4GHz";
char pass[] = "123456789-0";
char auth[] = "a3bd1c2e7a0246c18e44cc35bb15d30a";

void Sensor() {
    float h = dht.readHumidity();
    float t = dht.readTemperature(); // or
dht.readTemperature(true) for Fahrenheit
    if (isnan(h) || isnan(t)) {
        Serial.println("Failed to read from DHT sensor!");
        return;
    }
    Serial.printf("H = %.1f : T = %.1f ", h, t);
    Serial.println();
    Blynk.virtualWrite(V0, h);
    Blynk.virtualWrite(V1, t);
}
void setup() {
    dht.begin();
    ax.begin();
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(2000L, Sensor);
}
void loop() {
    Blynk.run();
    timer.run();
}
```

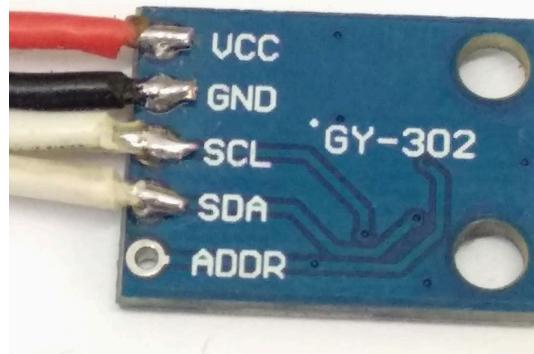
การวัดค่าความเข้มแสง

BH1750

- ให้หน่วยอุปกรณ์เป็น LUX
- สื่อสารแบบ I2C



ด้านหลัง



AX-WiFi	BH1750
D1(GPIO 5)	SCL
D2(GPIO 4)	SDA

ด้านหน้า

การต่อใช้งาน

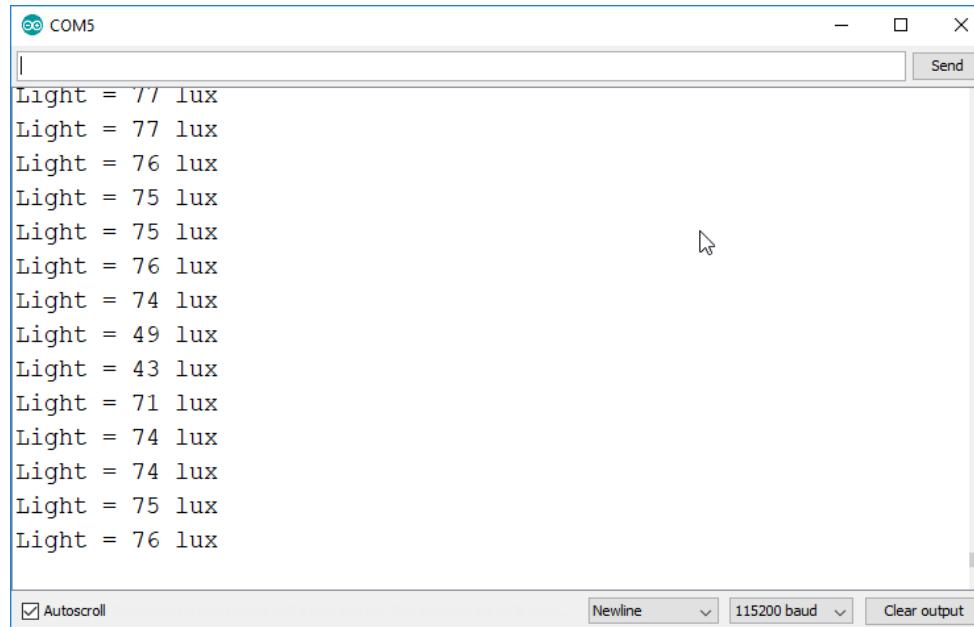
โปรแกรมอ่านค่าความเข้มแสง

```
#include <Wire.h>
int BH1750address = 0x23;
byte buff[2];
void setup() {
    Wire.begin();
    Serial.begin(115200);
}
void loop() {
    int val = BH1750_Read(BH1750address);
    if (val > 0 ){      // Read and check data from BH1750
        Serial.print("Light = ");
        Serial.print(val);
        Serial.println(" lux");
    }else{
        Serial.println("Failed to read from BH1750 sensor!");
    }
    delay(150);
}
```

โปรแกรมอ่านค่าความเข้มแสง (ต่อ)

```
int BH1750_Read(int address) {
    int i = 0;
    float val = 0;
    Wire.beginTransmission(address);
    Wire.write(0x10);
    Wire.endTransmission();
    delay(200);
    Wire.beginTransmission(address);
    Wire.requestFrom(address, 2);
    while (Wire.available()) {
        buff[i] = Wire.read();
        i++;
    }
    Wire.endTransmission();
    if (2 == i) {
        val = ((buff[0] << 8) | buff[1])/1.2;
        return val;
    }else{
        return -1;
    }
}
```

ผลลัพธ์



The screenshot shows a terminal window titled "COM5". The window displays a series of light sensor readings in lux. The text area contains the following output:

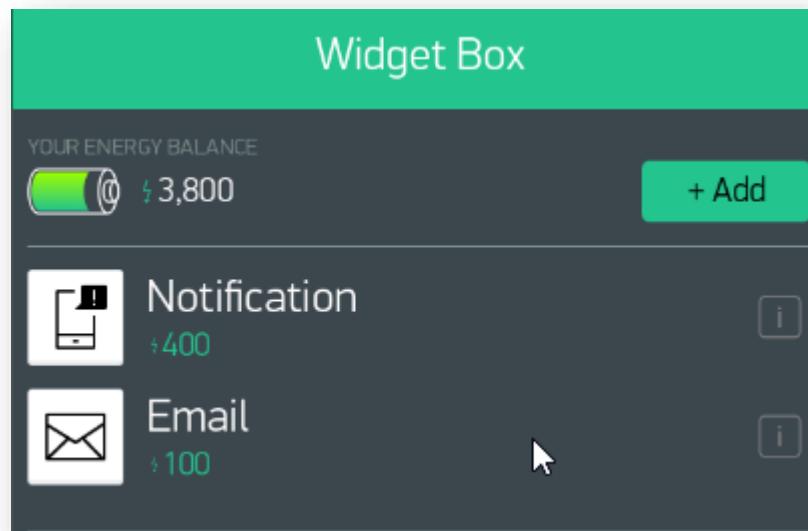
```
Light = 77 lux
Light = 77 lux
Light = 76 lux
Light = 75 lux
Light = 75 lux
Light = 76 lux
Light = 74 lux
Light = 49 lux
Light = 43 lux
Light = 71 lux
Light = 74 lux
Light = 74 lux
Light = 75 lux
Light = 76 lux
```

At the bottom of the window, there are three buttons: "Autoscroll" (checked), "Newline" (dropdown menu), "115200 baud" (dropdown menu), and "Clear output".

การแจ้งเตือนบน Blynk App



บริการ Notification สามารถส่งแจ้งเตือนได้ 15 วินาทีต่อ 1 ครั้ง และสามารถส่งได้ข้อความสูงสุด 120 ตัวอักษร ใช้พลังงานไป 400

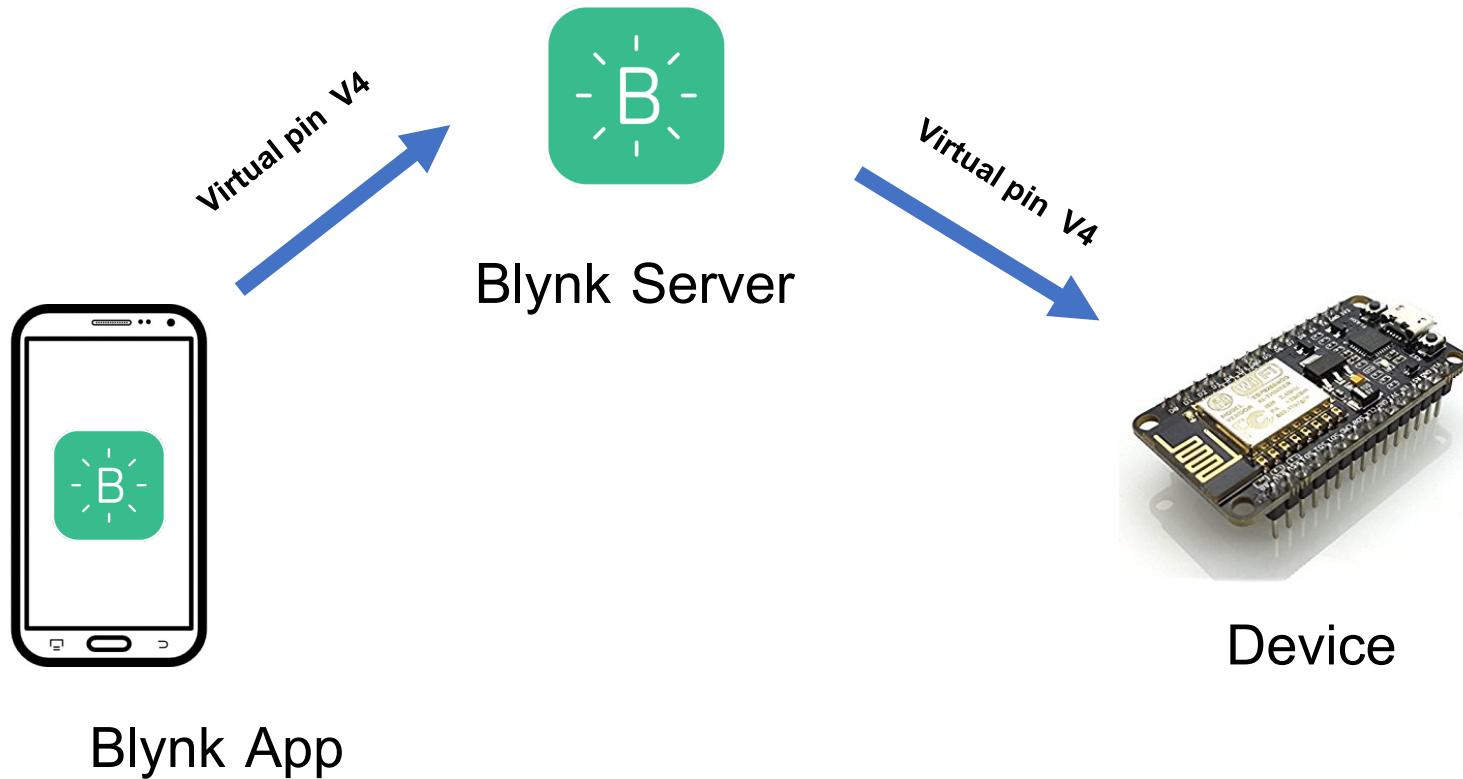


คำสั่ง

Blynk.notify("Alarm !");

การบอกรับค่า(Subscribe) จาก Blynk

Blynk มีหัวข้อการส่งข้อมูล(topic) หรือ Virtual pin ให้ใช้ทั้งหมด 128 หัวข้อ NodeMCU สามารถบอกรับหัวข้อเพื่อนำข้อมูลนั้นมาใช้งานได้



รูปแบบชุดคำสั่ง

```
BLYNK_WRITE(V4) { // Subscribe Virtual pin 4
    int pinValue = param.asInt();
    Serial.print("V4 Value is : ");
    Serial.println(pinValue);
}
```

V4 คือ หัวข้อที่ต้องการรับข้อมูล

ตัวอย่าง

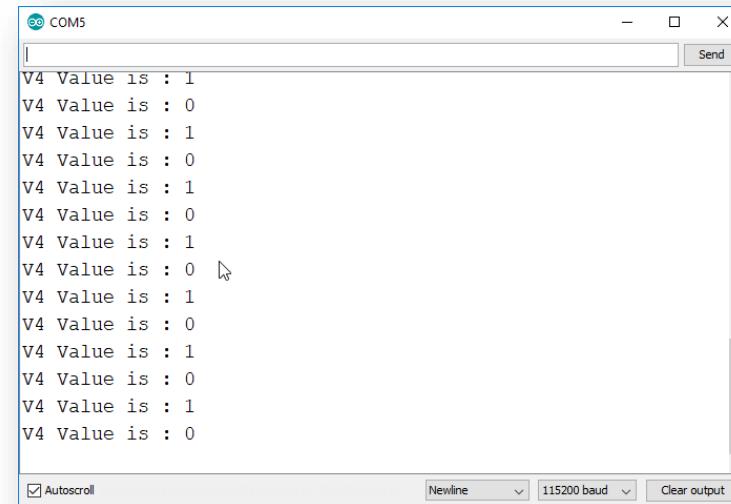
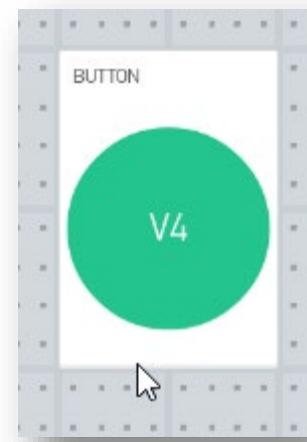
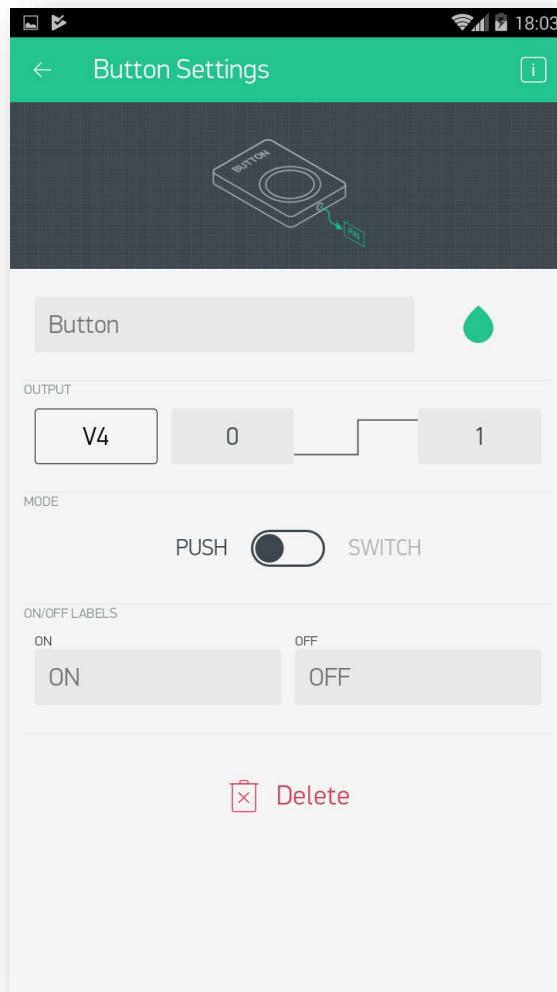
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"

char ssid[] = "INEX01_2.4GHz";
char pass[] = "123456789-0";
char auth[] = "a3bd1c2e7a0246c18e44cc35bb15d30a";
BLYNK_WRITE(V4){ // Subscribe Virtual pin 4
    int pinValue = param.asInt();
    Serial.print("V4 Value is : ");
    Serial.println(pinValue);
}

void setup() {
    ax.begin();
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);
}

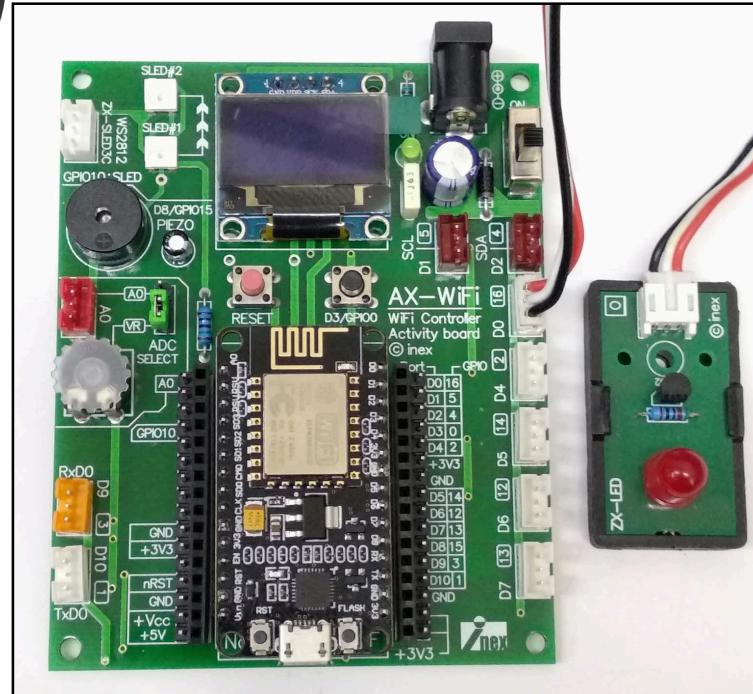
void loop() {
    Blynk.run();
}
```

ตั้งค่าบัน Blynk App



สร้างระบบเปิด-ปิด ไฟ 2 ทาง

เปิด-ปิด ด้วยสวิตซ์ D3



เปิด-ปิด ปุ่มบน Blynk App

ตัวอย่างโปรแกรม

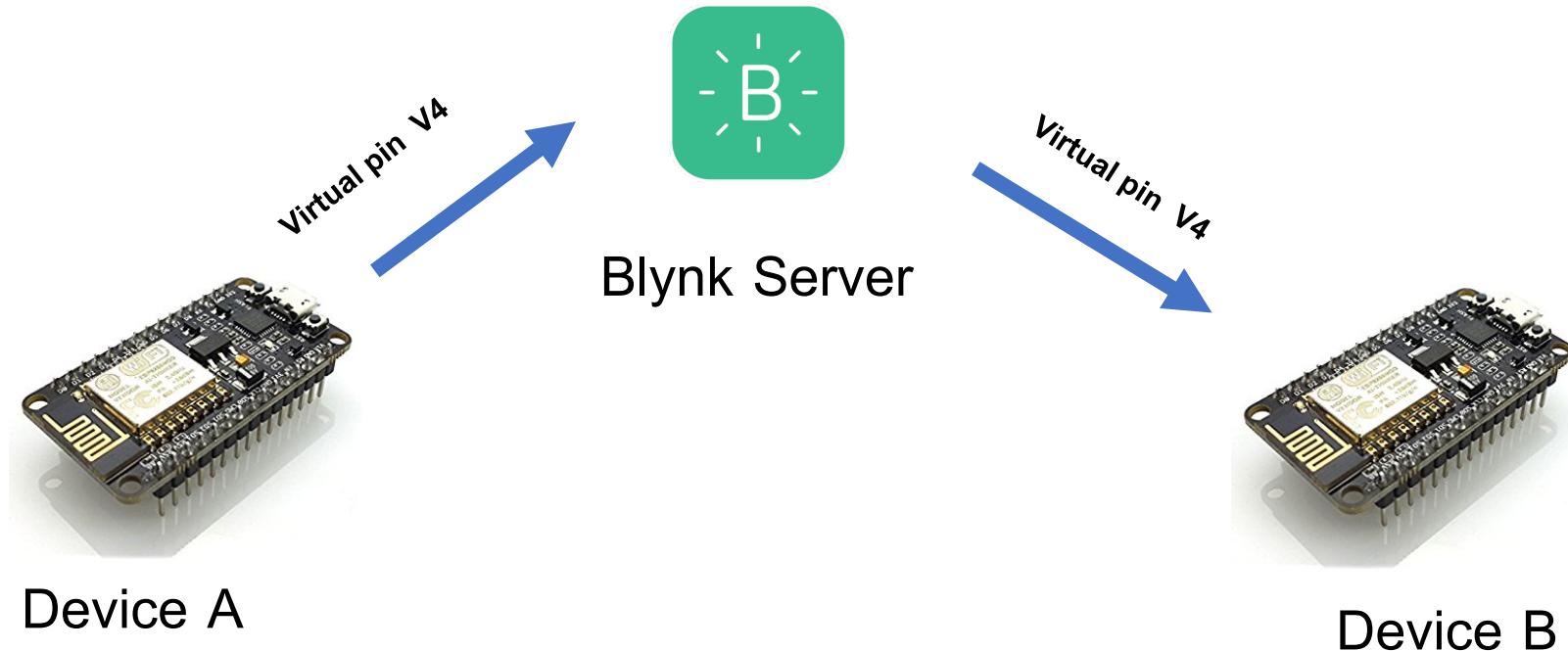
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"
char ssid[] = "INEX01_2.4GHz";
char pass[] = "123456789-0";
char auth[] = "a3bd1c2e7a0246c18e44cc35bb15d30a";
void setup() {
    pinMode(LEDpin,OUTPUT);
    pinMode(SWpin,INPUT);
    ax.begin();
    Serial.begin(115200);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(2000L,SendStatus);
}
void loop() {
    if(!digitalRead(SWpin)){
        digitalWrite(LEDpin,!digitalRead(LEDpin));
        delay(1000);
    }
    Blynk.run();
    timer.run();
}
```

ตัวอย่างโปรแกรม (ต่อ)

```
int LEDpin = D0 ;
int SWpin = D3 ;
BlynkTimer timer;
BLYNK_WRITE(V4) { // Subscribe Virtual pin 4
    int pinValue = param.asInt();
    digitalWrite(LEDpin,pinValue);
}
void SendStatus() {
    Blynk.virtualWrite(V4,digitalRead(LEDpin));
}
```

การบอกรับค่า(Subscribe) จาก Blynk

Blynk มีหัวข้อการส่งข้อมูล(topic) หรือ Virtual pin ให้ใช้ทั้งหมด 128 หัวข้อ NodeMCU สามารถบอกรับหัวข้อเพื่อนำข้อมูลนั้นมาใช้งานได้



Device A

Device B

อุปกรณ์ A ส่งข้อมูลในหัวข้อ V4 และในขณะนั้นอุปกรณ์ B บอกรับหัวข้อ V4 ด้วย จะต้องใช้ Bridge โดยไม่จำเป็นต้องใช้ Blynk App

ແນະນຳຕ້ວ

ຊື່ອ : ຕີරຸຫ ຈິຕພຣມມາ

Name : Teerawut Chitphomma

ຕຳແໜ່ງງານ : Engineer

E-mail : teerawut@inex.co.th

ບຣັຈັກ ອິນໄນເວຕີຟ ເໝັກເພອວິເມັນດີ ຈຳກັດ
ເລຂທີ່ 108 ຊອຍສຸຂຸມວິຖ 101/2 ດັນນສຸຂຸມວິຖ
ແຂວງບາງນາ ເຂດບາງນາ ກຽງເທິພາ 10260
ເລຂປະຈຳຕ້ວຜູ້ເສີຍການ : 01055 4 0042483

ໂທຮສທ໌ : 0-2747-7001-4 ສ້ອງ 081-920-9477

ໂທຮສາຣ : 0-2747-7005

