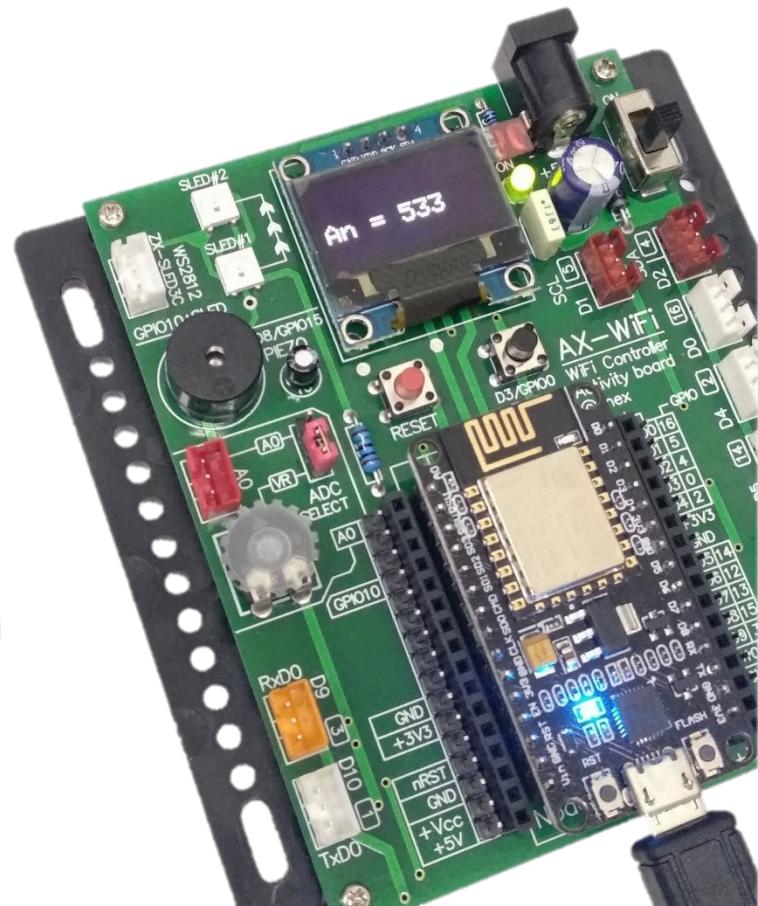
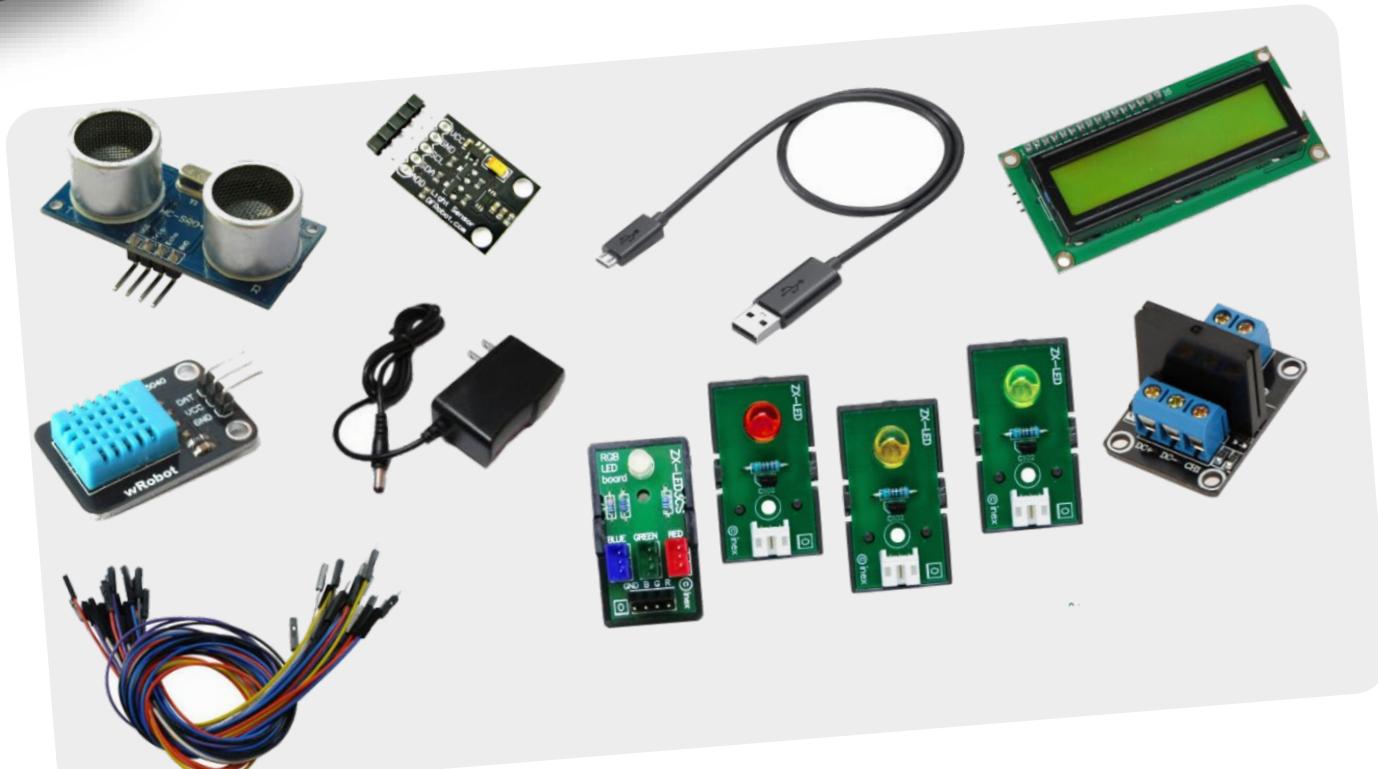
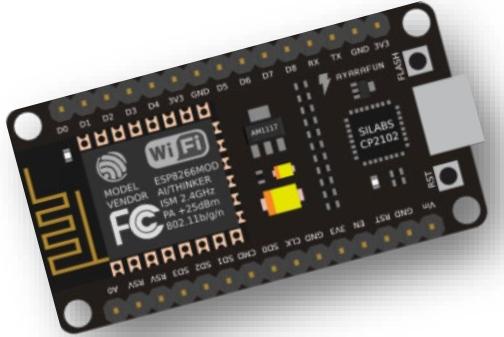
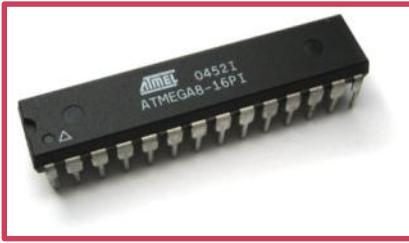


Start the Internet of Things (IoT)

with NodeMCU



ໃນໂຄຣຄອບໄໂສາເລອຣີກ່ຽວໜີ



หลักการระบบควบคุ้ม



อินพุต



ประมวลผล

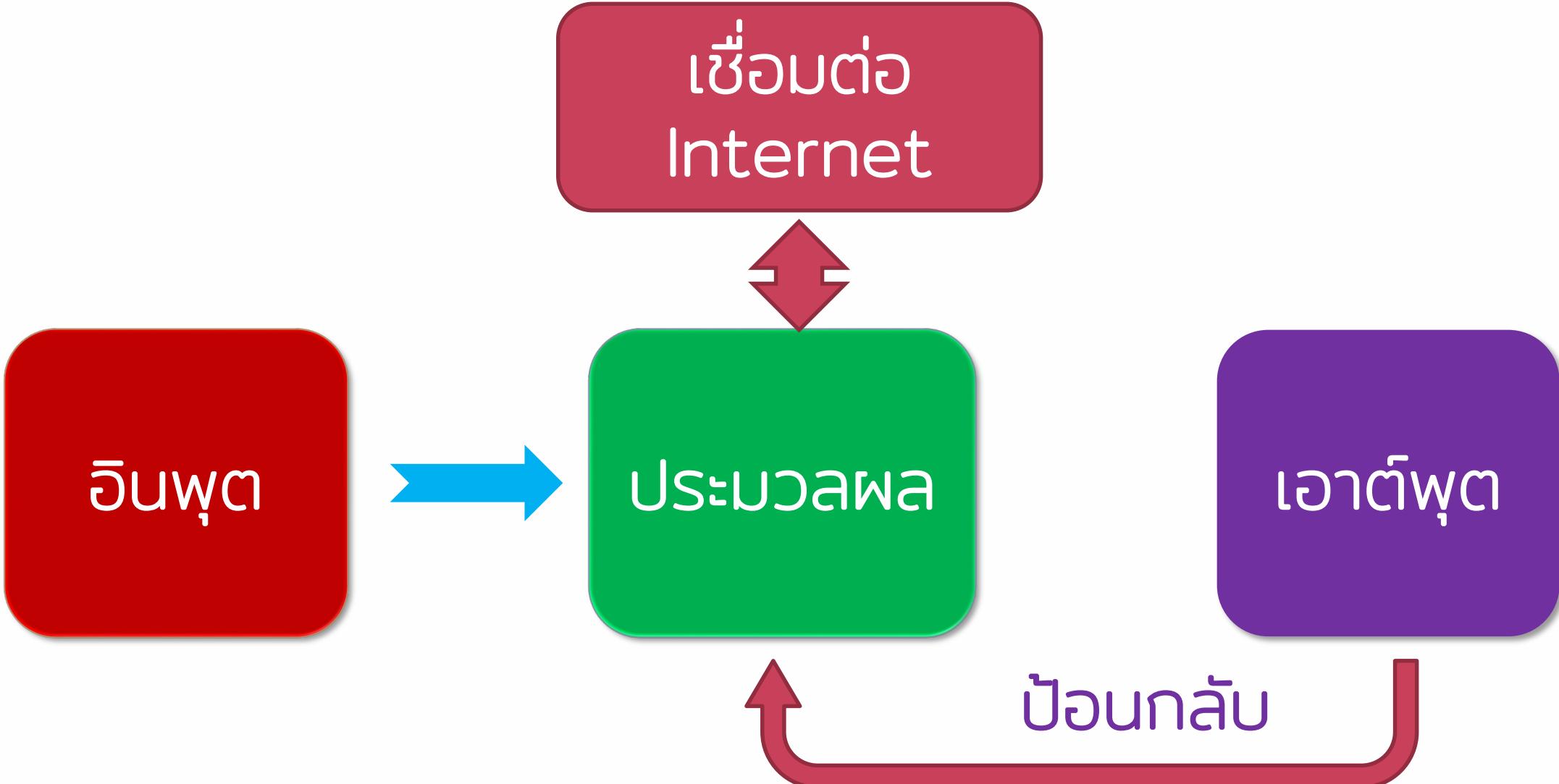


เอาต์พุต

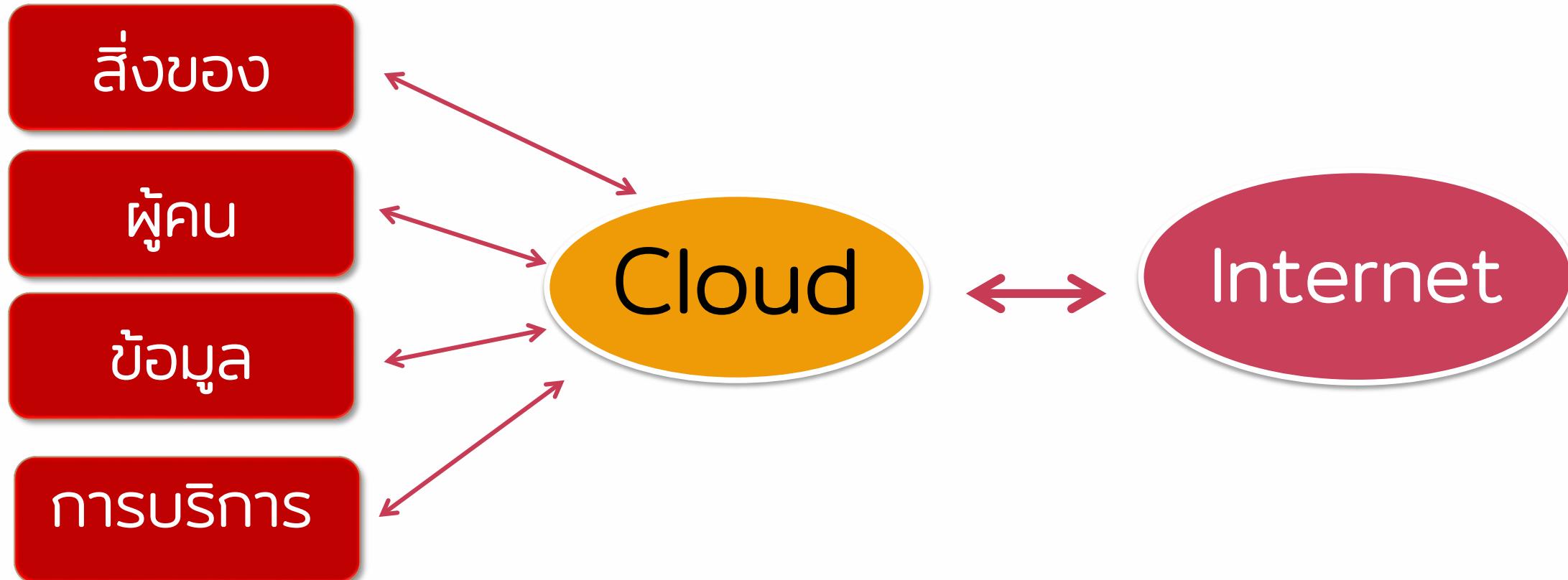


ป้อนกลับ

หลักการระบบควบคุม IoT



ความหมายของ IoT : Internet of Thing



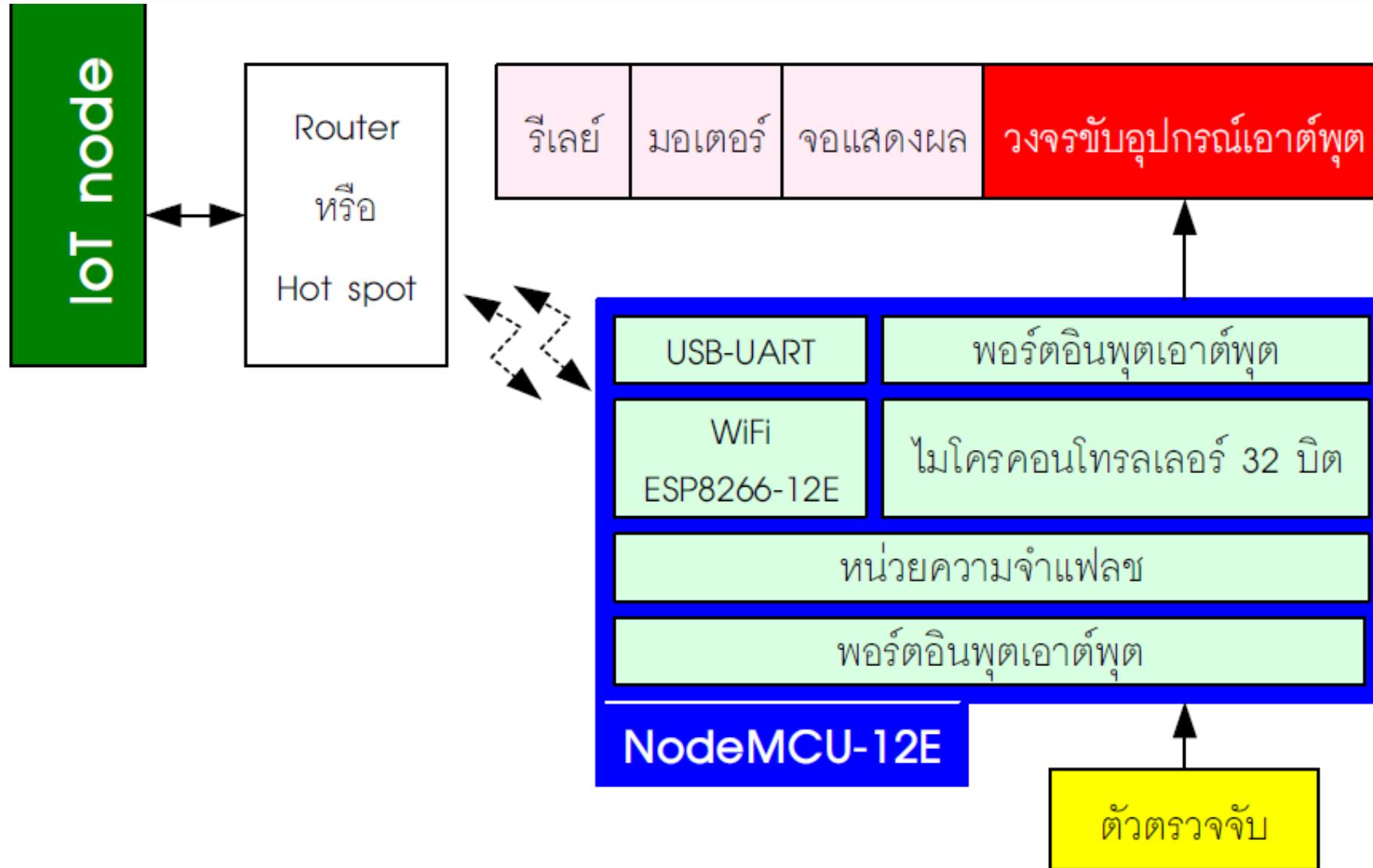
หมายถึง เทคโนโลยีที่ก่อให้เกิดการเชื่อมโยงกันของ สิ่งของ ผู้คน ข้อมูล และ การบริการเข้ากับเครือข่ายอินเทอร์เน็ต

ส่วนประกอบของ IoT

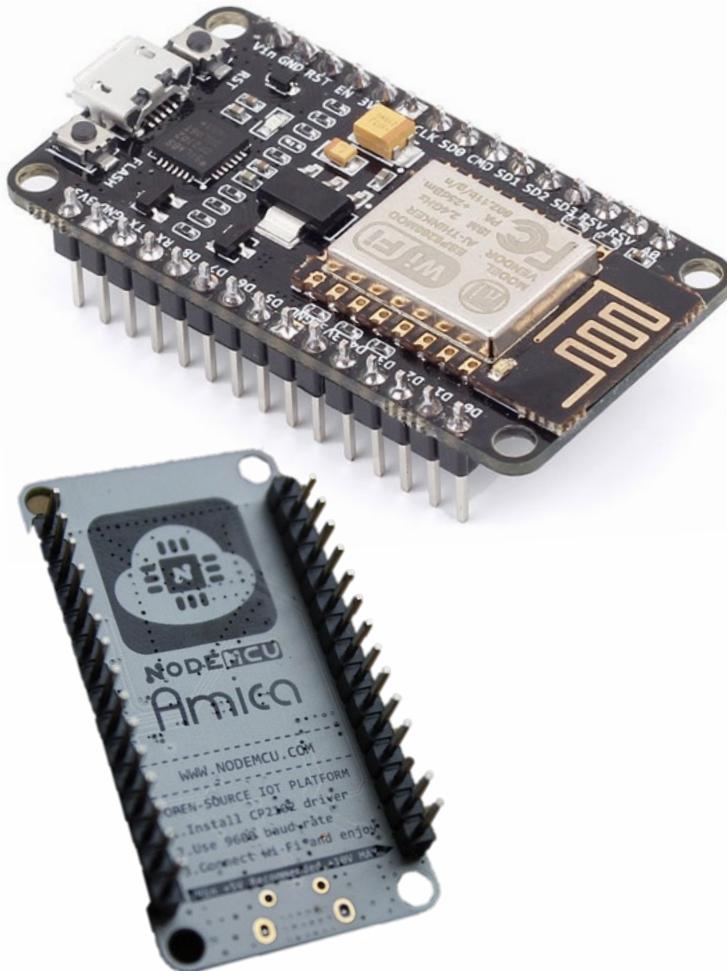
1. สิ่งของ (Thing)
2. ข้อมูล
3. ระบบเชื่อมต่ออินเทอร์เน็ต (แบบมีสายหรือไร้สาย)
4. ตัวควบคุม, ตัวตรวจสอบและอุปกรณ์ขับໂ Holden
5. ระบบจัดการฐานข้อมูล Cloud server



ແພນກາພເພື່ອໃຊ້ງານ NodeMCU



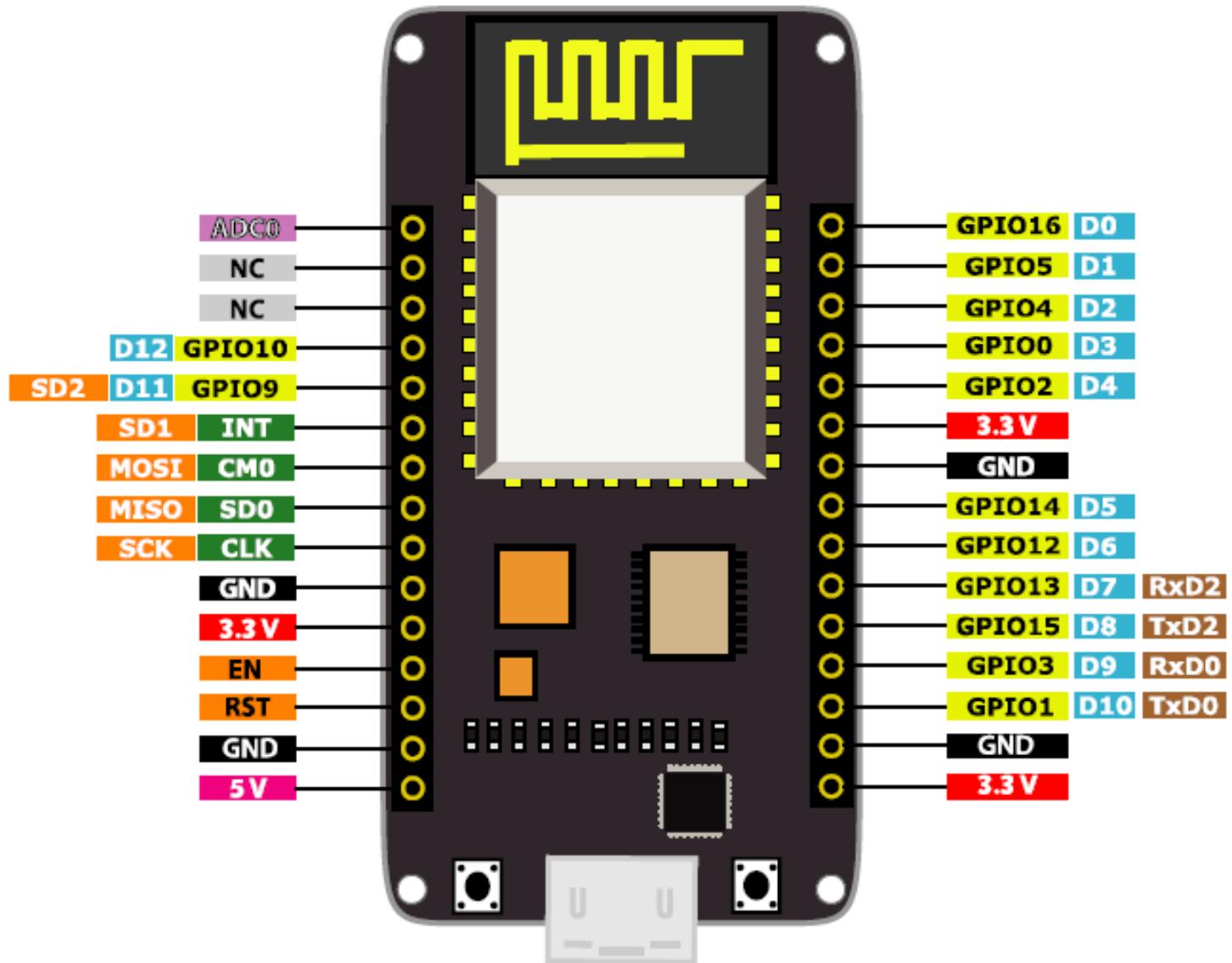
NodeMCU-12E



ESP8266-12E (ESP8266EX)

- MCU 32 BIT
- หน่วยความจำ 4 MByte (12E)
- RAM ประมาณ 112 kByte
- WIFI ในตัว
- USB 2 Serial สำหรับดาวน์โหลดไฟล์
- ใช้ไฟเลี้ยงผ่านพอร์ต USB ได้
- ดาวน์โหลดผ่าน USB
- GPIO 16 ขา
- ANALOG 1 ช่อง

NODE MCU Pinout



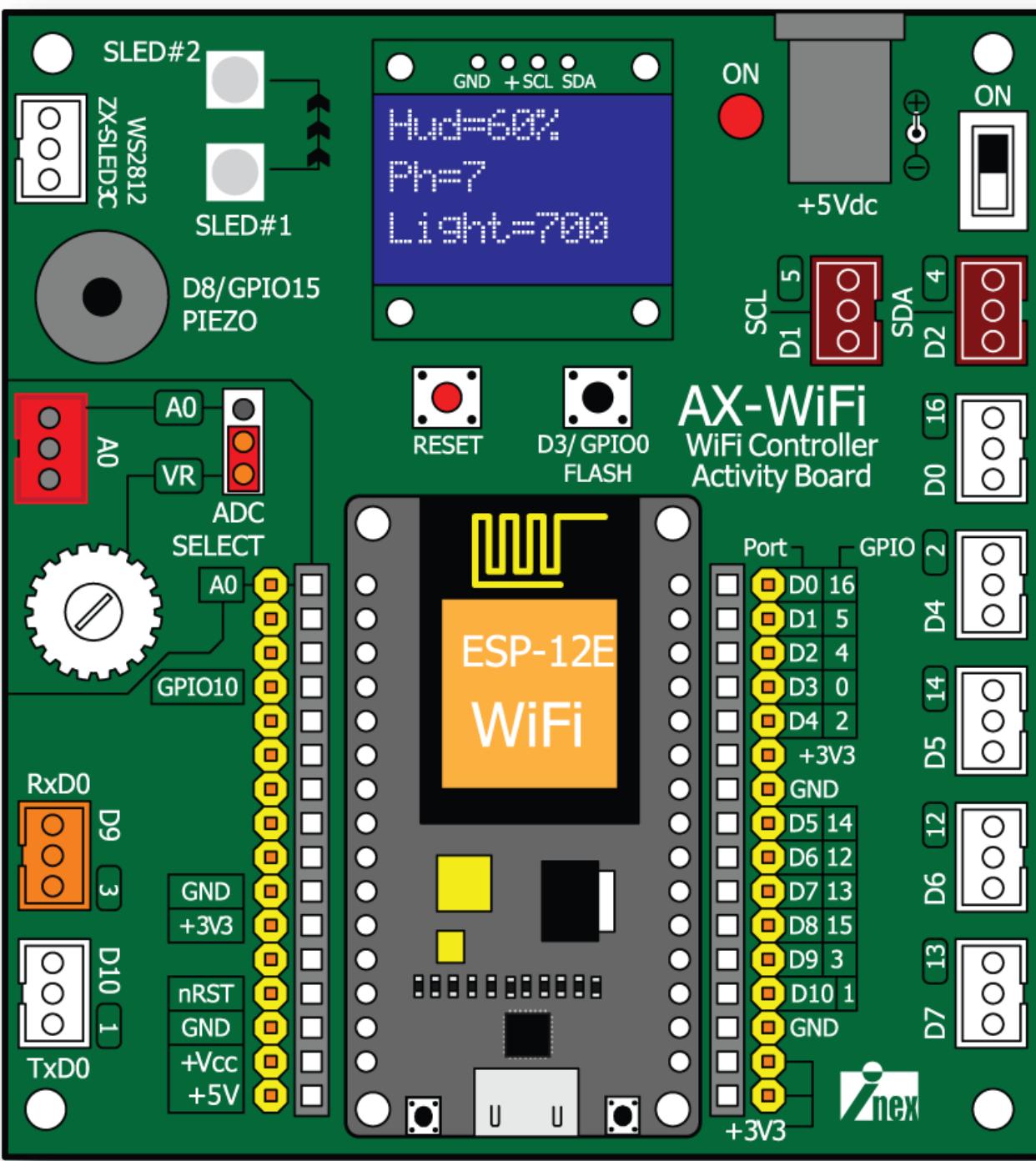
GPIO 3V

GPIO กระแสสูงสุด 15mA

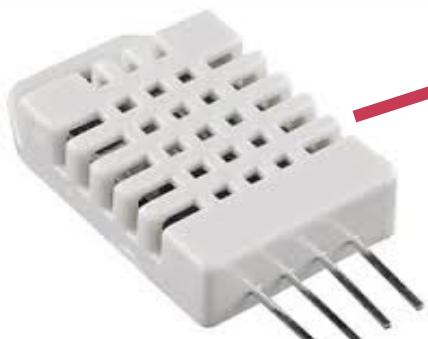
Analog รับแรงดัน 0-3.3V
(มีวงจรแบ่งแรงดัน)

AX-WiFi

บอร์ดเชื่อมต่อ I/O สำหรับ NodeMCU



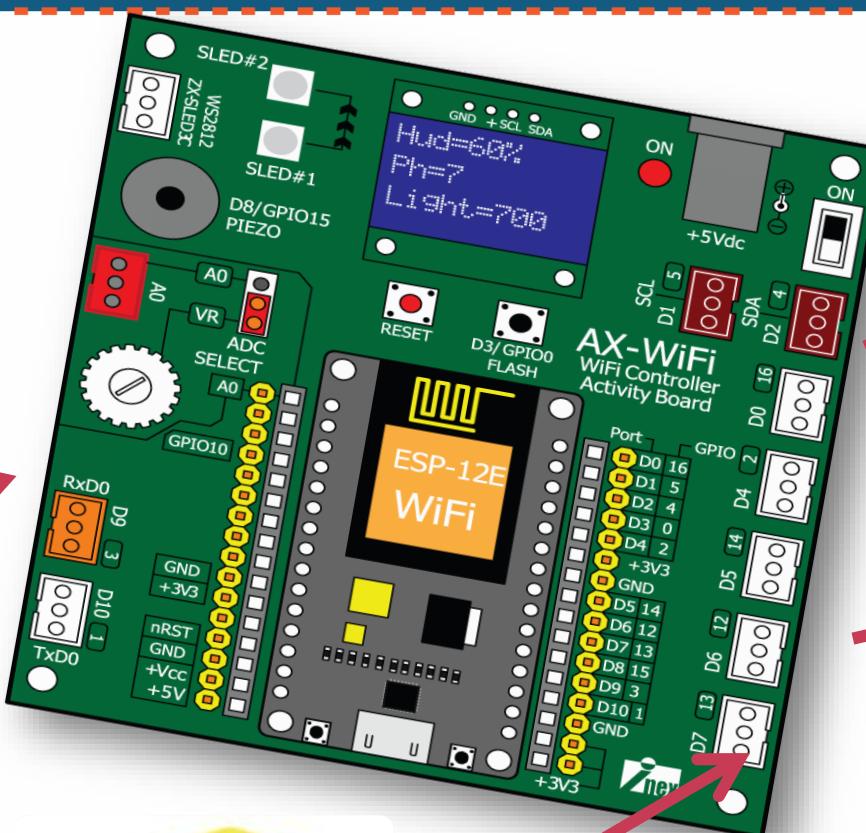
อุปกรณ์ต่อพ่วงต่างๆ ที่ใช้กดสอบ



DHT22



SWITCH

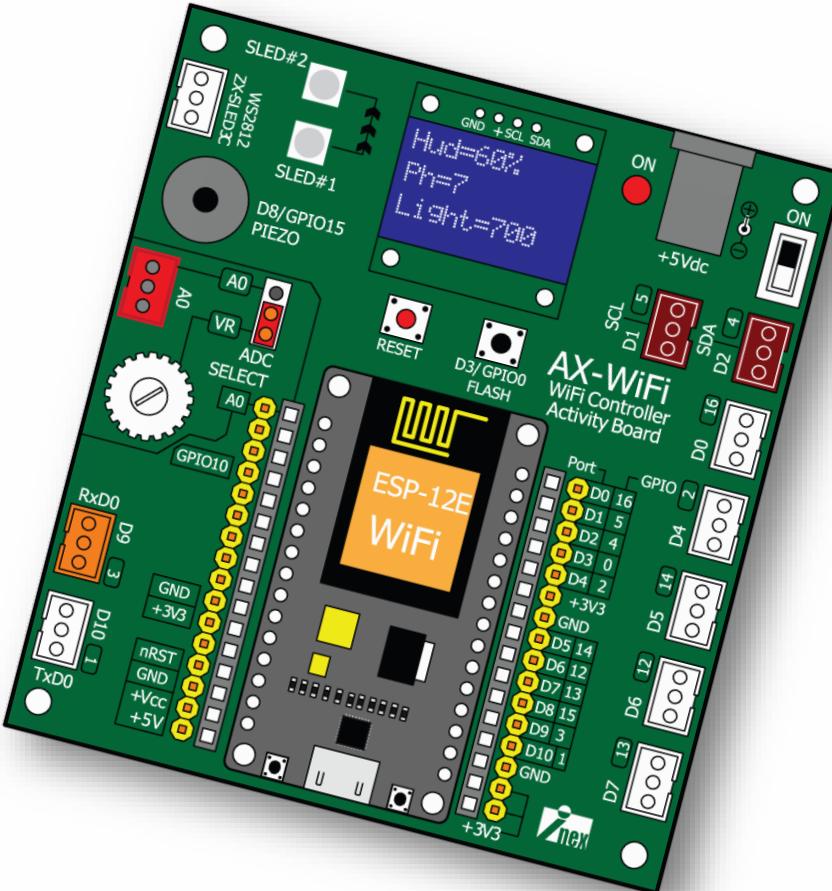


BH1750



ZX-LED

NodeMCU V2 กับ Arduino

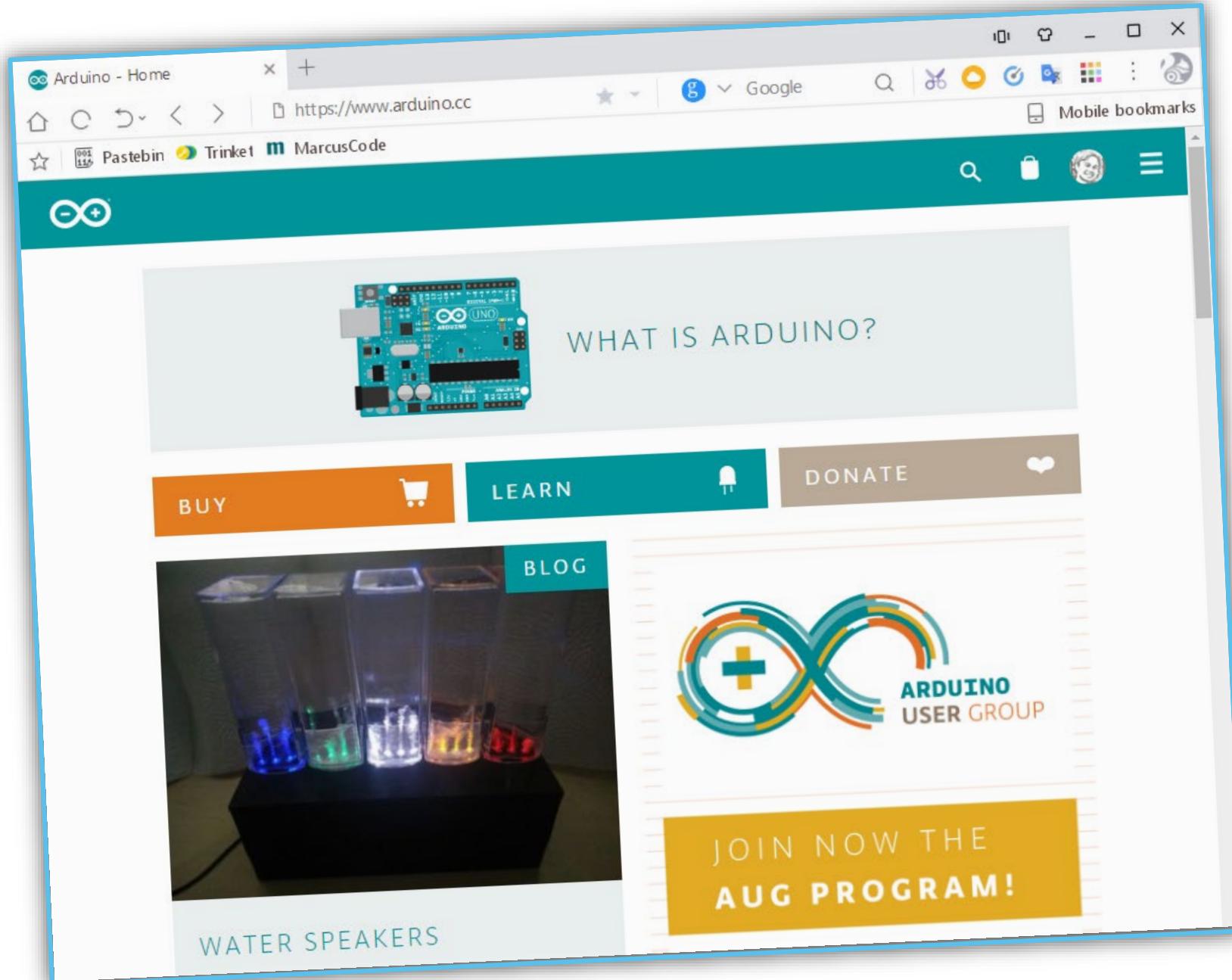


Arduino1.8.x

จุดเปลี่ยนของ ESP8266
คือมีคนทำให้เขียนโปรแกรมด้วย Arduino ได้

ต้นกำเนิด Arduino

- สร้างฮาร์ดแวร์
- ออกแบบซอฟต์แวร์
- ดาวน์โหลดได้โดยตรง
- OpenSource
- มี Editor ของตัวเอง
- มีตัวอย่างมากมาย



หน้าต่างช่วยเหลือครบทั้งหมด และคุณเคย

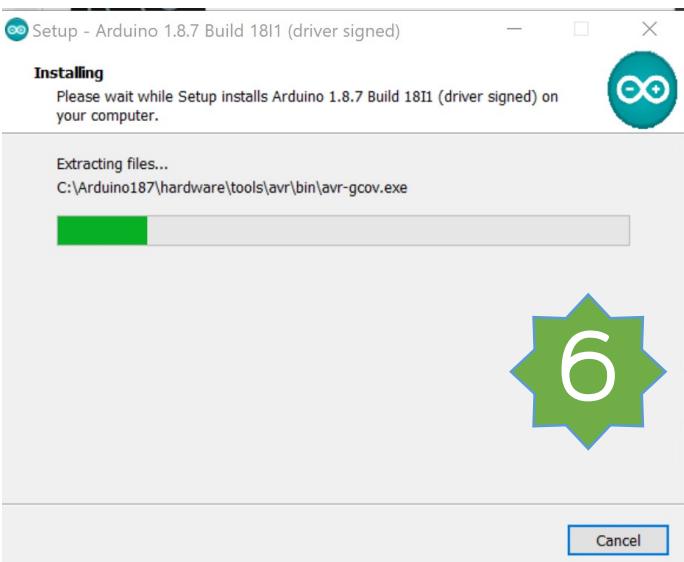
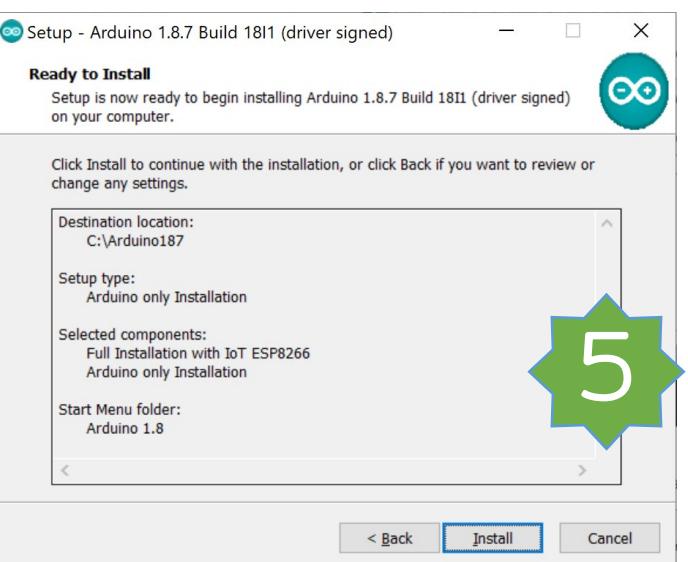
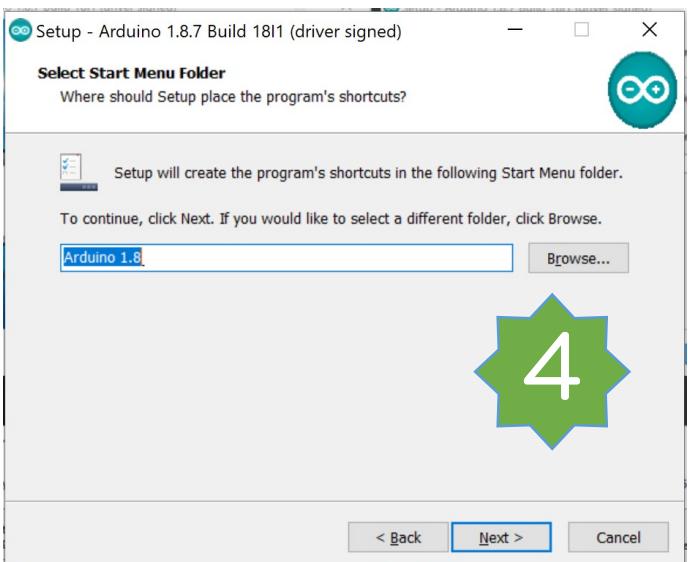
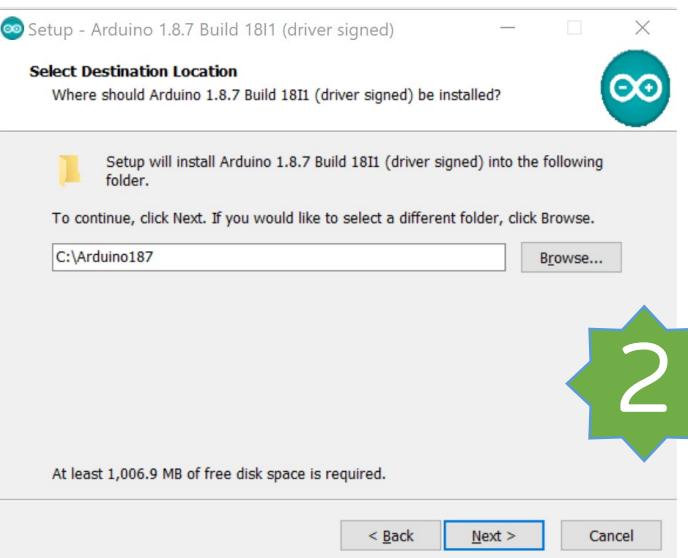
The screenshot shows a web browser displaying the Arduino Reference page at <https://www.arduino.cc/reference/en/>. The page title is "Language Reference". On the left, there's a sidebar with navigation links: LANGUAGE (selected), FUNCTIONS, VARIABLES, STRUCTURE, LIBRARIES, and GLOSSARY. Below these are two sections: "The Arduino Reference text is licensed under a Creative Commons Attribution-Share Alike 3.0 License." and "Find anything that can be improved? Suggest corrections and new documentation via GitHub.". The main content area starts with a paragraph about the Arduino programming language being divided into three parts: structure, values (variables and constants), and functions. Below this is a section titled "FUNCTIONS" with a sub-section for "Digital I/O" containing `digitalRead()`, `digitalWrite()`, and `pinMode()`; a "Math" section containing `abs()`, `constrain()`, `map()`, `max()`, `min()`, `pow()`, and `sq()`; and a "Random Numbers" section containing `random()` and `randomSeed()`. There's also a "Bits and Bytes" section with `bit()`, `bitClear()`, and `bitRead()`. At the bottom right of the page, there's a "100%" zoom indicator.

- บุ๊ดคำสั่ง
- ไลบรารี
- ตัวแปร
- คำสั่งวน

ขั้นตอนการติดตั้งซอฟต์แวร์

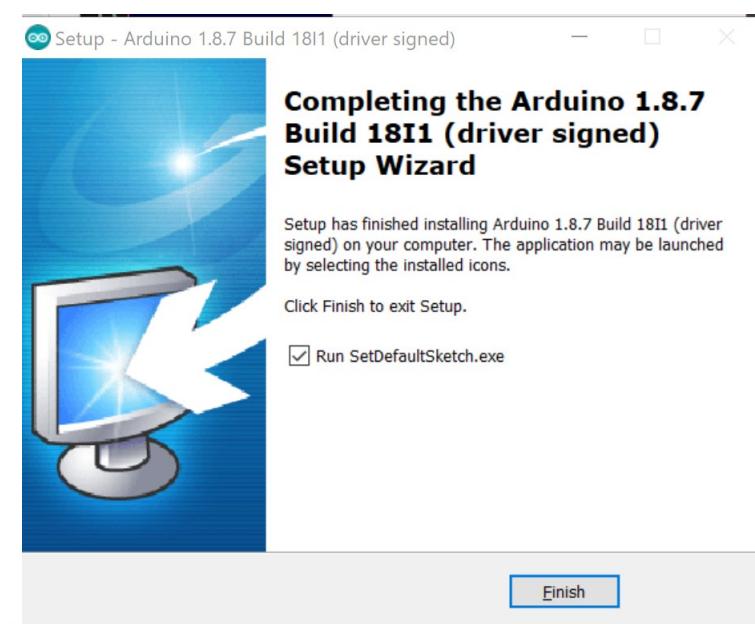
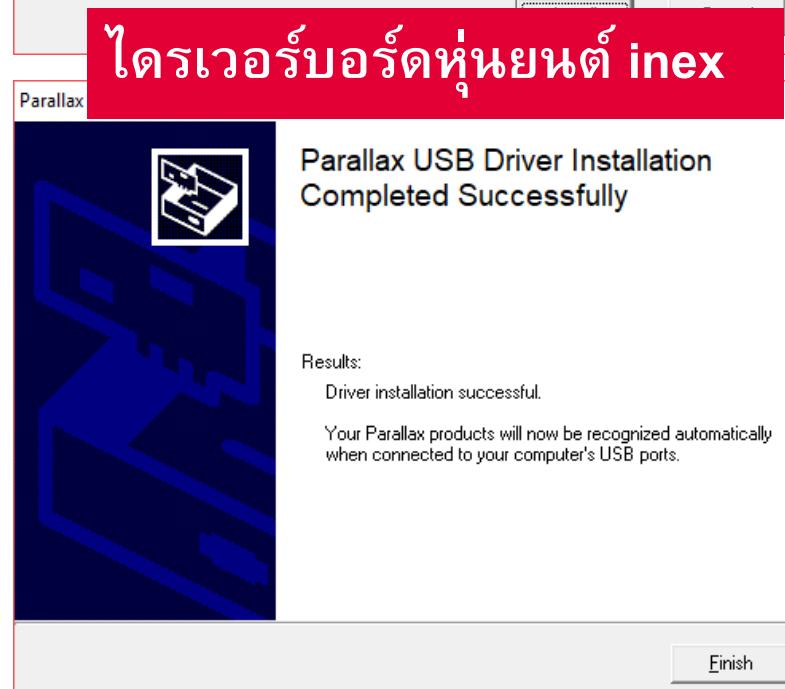
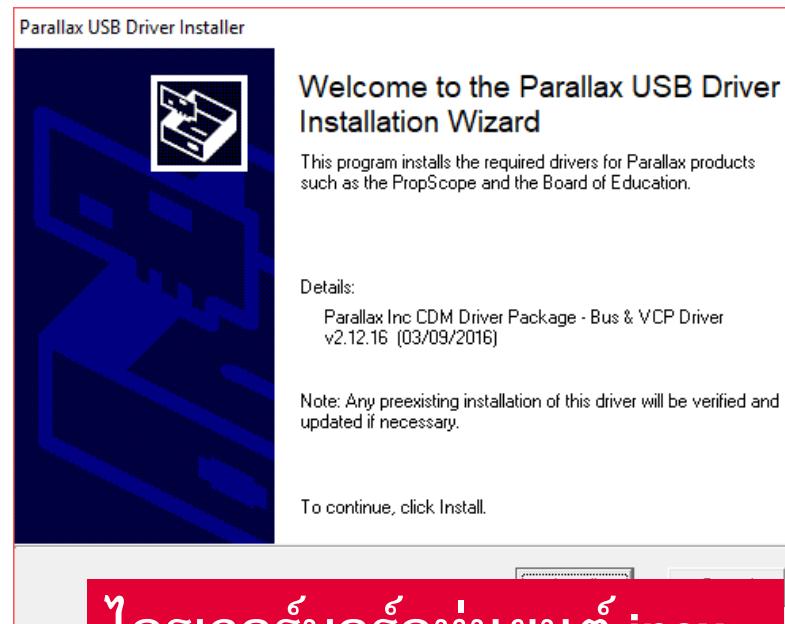
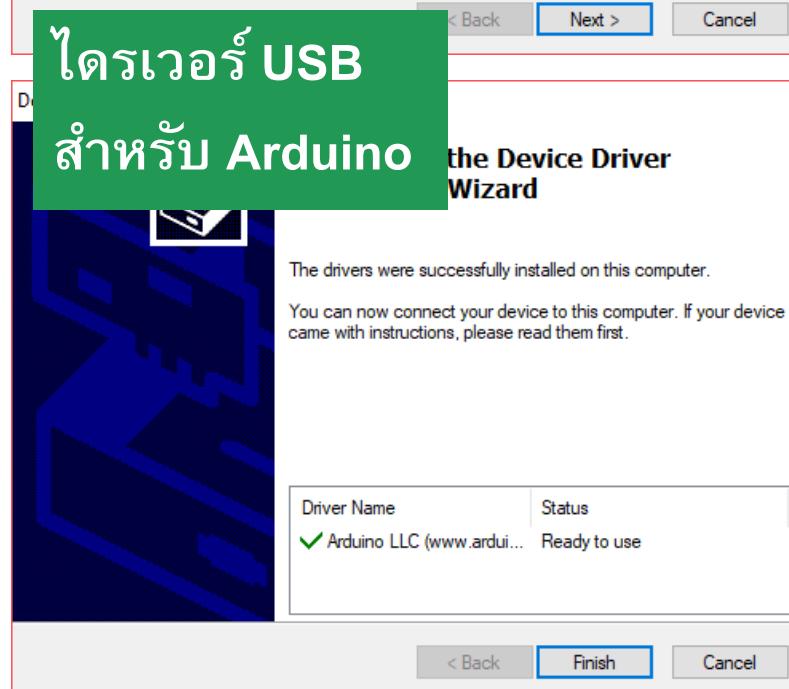
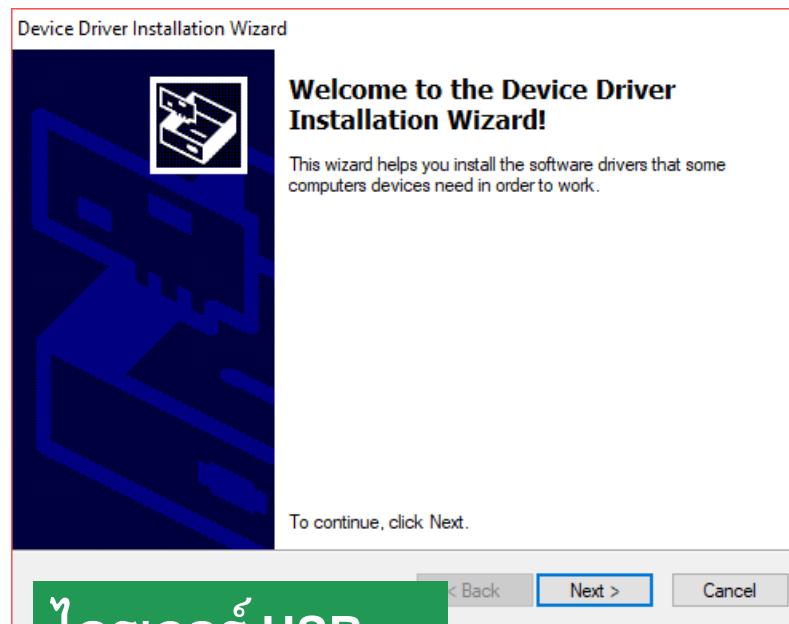


Arduino1.8.8_IoT_Setup181217.exe

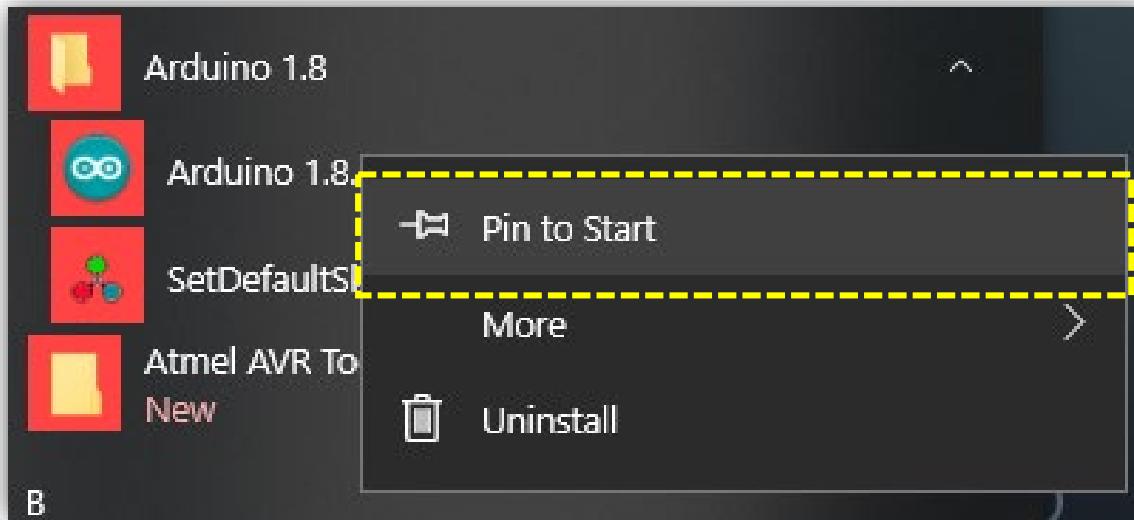
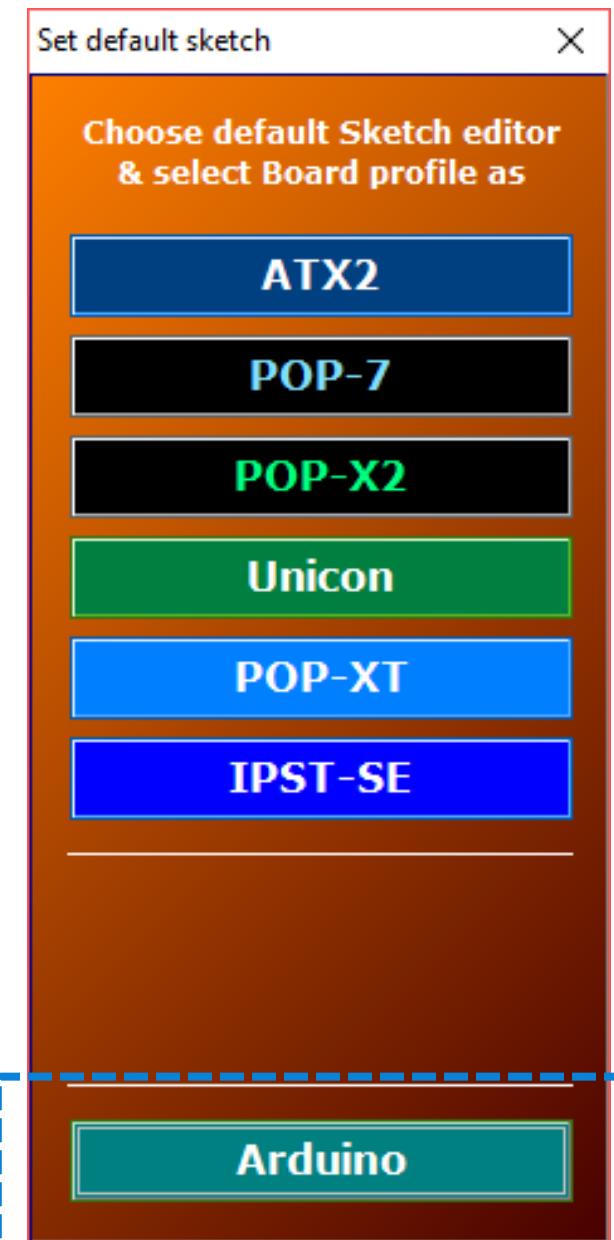


ขั้นตอนการติดตั้งไดรเวอร์

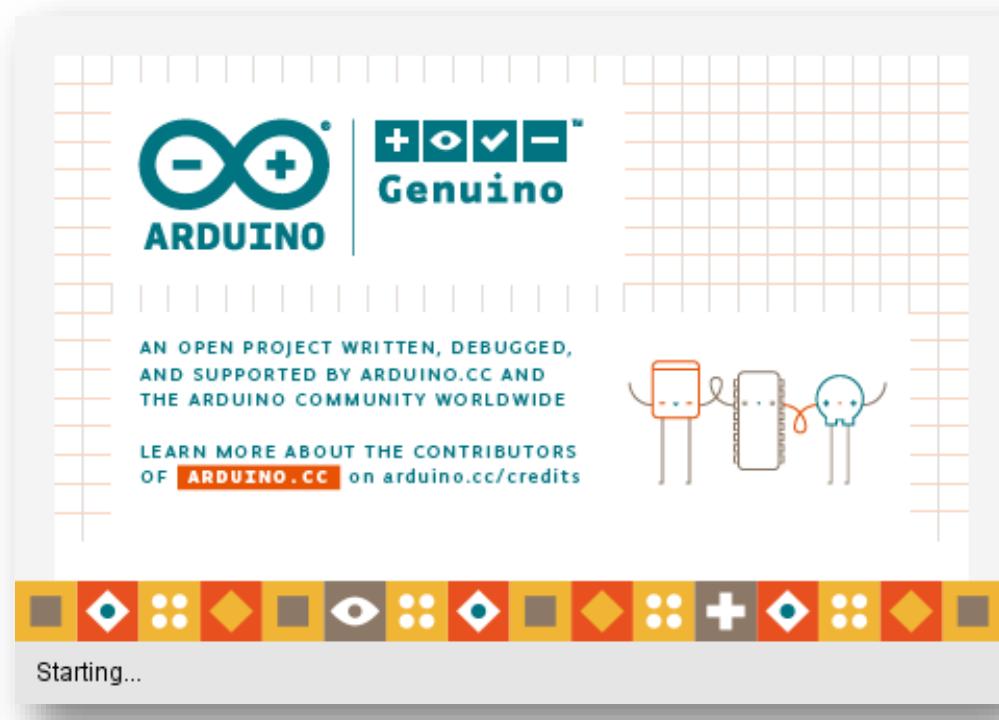
ไดรเวอร์สำหรับ NodeMCU คือ CP2102



เลือก Arduino สำหรับ และเพิ่ม Shortcut ไว้ที่ Start Menu



เปิดใช้งานครั้งแรก



sketch_apr29a | Arduino 1.8.8

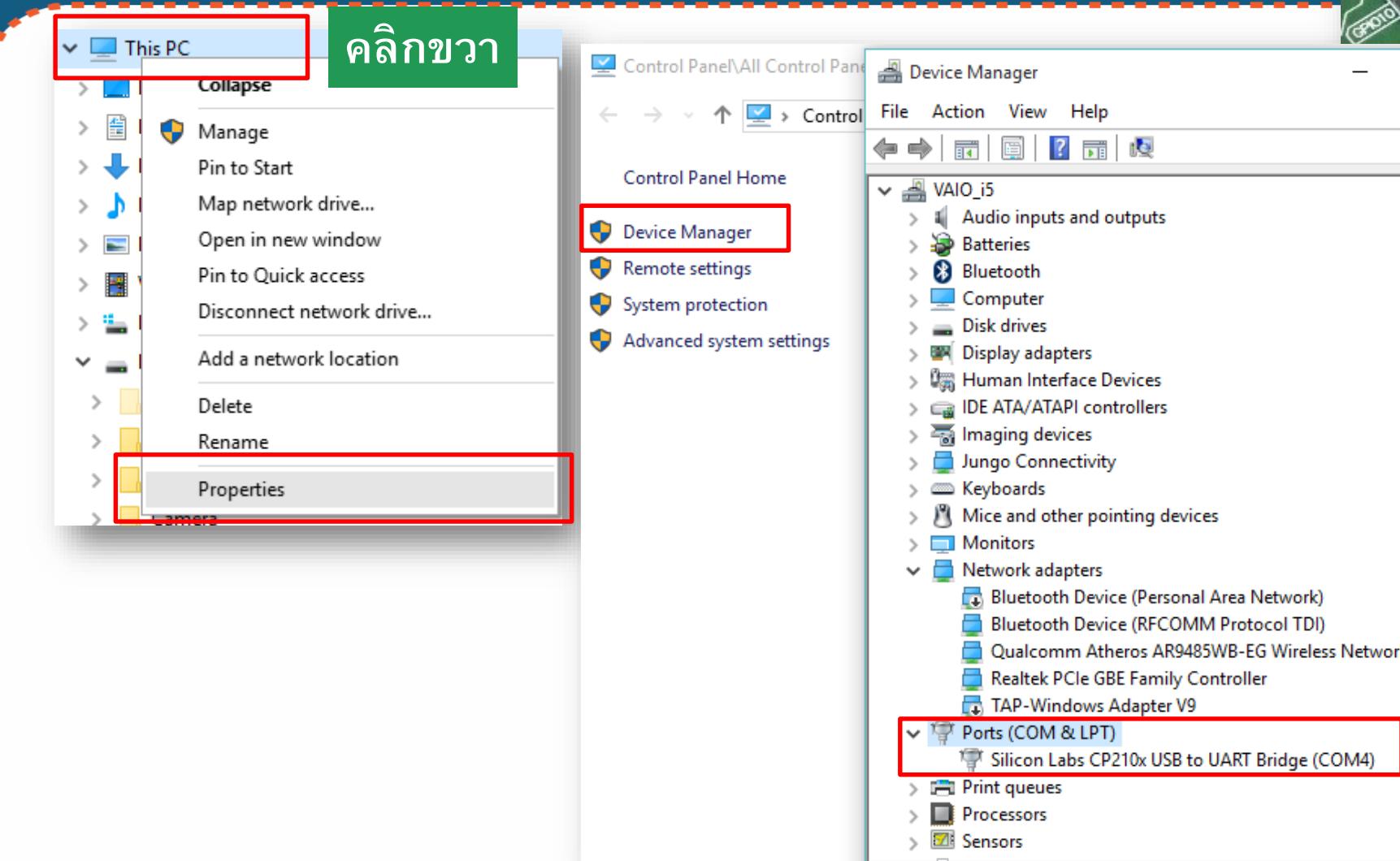
File Edit Sketch Tools Help

sketch_apr29a

```
1 void setup() {  
2  
3 }  
4  
5 void loop() {  
6  
7 }
```

Arduino/Genuino Uno on COM4

ตรวจสอบการเชื่อมต่อ



เปิดโปรแกรม Arduino เลือกบอร์ด



sketch_may12a | Arduino 1.8.8

File Edit Sketch Tools Help

Auto Format Ctrl+T

Archive Sketch

Fix Encoding & Reload

Manage Libraries... Ctrl+Shift+I

Serial Monitor Ctrl+Shift+M

Serial Plotter Ctrl+Shift+L

IPST-SE Block

WiFi101 / WiFiNINA Firmware Updater

Board: "NodeMCU 1.0 (ESP-12E Module)"

Upload Speed: "115200"

CPU Frequency: "80 MHz"

Flash Size: "4M (1M SPIFFS)"

Debug port: "Disabled"

Debug Level: "None"

Boards Manager...

Adafruit Feather HUZZAH ESP8266

ESPRESSO Lite 1.0

ESPRESSO Lite 2.0

Phoenix 1.0

Phoenix 2.0

NodeMCU 0.9 (ESP-12 Module)

● NodeMCU 1.0 (ESP-12E Module)

Olimex MOD-WIFI-ESP8266(-DEV)

SparkFun ESP8266 Thing

SparkFun ESP8266 Thing Dev

SweetPea ESP-210

WeMos D1 R2 & mini

WeMos D1 mini Pro

WeMos D1 mini Lite

WeMos D1 R1

เชื่อมต่อพอร์ตของ NodeMCU



The screenshot shows the Arduino IDE interface with the following details:

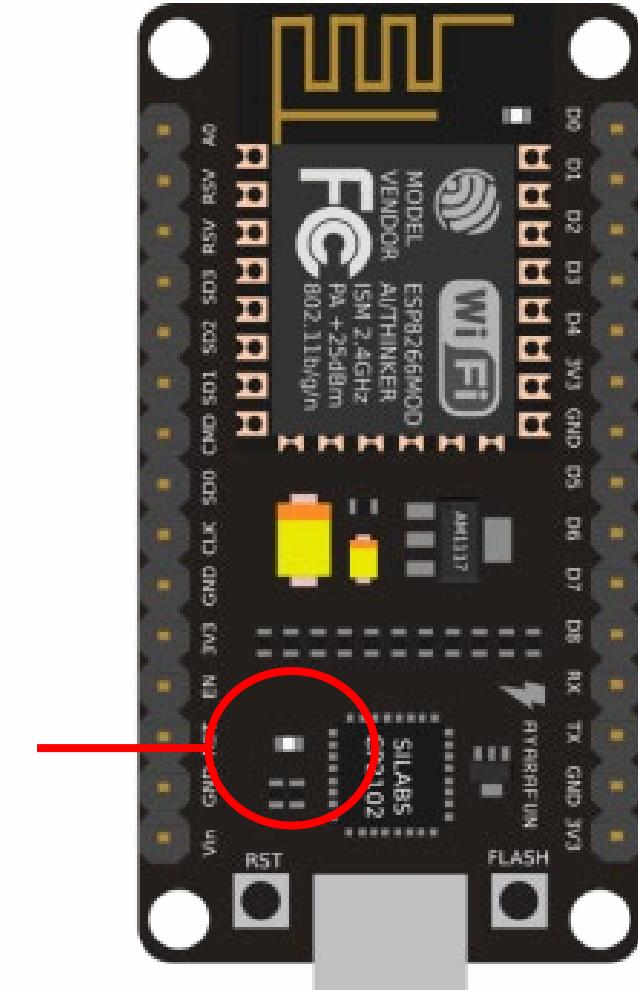
- File Edit Sketch Tools Help**: The Tools menu is currently selected.
- sketch_may12**: The current sketch name.
- Code Preview**: A vertical bar on the left showing the code structure.
- Tools Menu Options**:
 - Auto Format (Ctrl+T)
 - Archive Sketch
 - Fix Encoding & Reload
 - Manage Libraries...
 - Serial Monitor (Ctrl+Shift+M)
 - Serial Plotter (Ctrl+Shift+L)
 - IPST-SE Block
 - WiFi101 / WiFiNINA Firmware Updater
 - Board: "NodeMCU 1.0 (ESP-12E Module)"
 - Upload Speed: "115200"
 - CPU Frequency: "80 MHz"
 - Flash Size: "4M (1M SPIFFS)"
 - Debug port: "Disabled"
 - Debug Level: "None"
 - IwIP Variant: "v2 Lower Memory"
 - Erase Flash: "Only Sketch"
- Port Selection**:
 - Port: "COM48" (highlighted with a red box)
 - Get Board InfoA secondary dropdown menu is open under "Port:":
 - Serial ports
 - ✓ COM48This secondary menu is also highlighted with a red box.

ทดลองโปรแกรมแรก



```
void setup() {  
    pinMode(D0, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(D0, 1);  
    delay(500);  
    digitalWrite(D0, 0);  
    delay(500);  
}
```

LED ตัวเหลือง D0/16



Compile / Upload



อัปโหลดโปรแกรม

ตรวจสอบไฟกรณ์

→



```
sketch_nov21a.ino

void setup() {
  pinMode(D0,OUTPUT);
}

void loop() {
  digitalWrite(D0,1);
  delay(500);
  digitalWrite(D0,0);
  delay(500);
}
```

Done compiling.

```
Sketch uses 201,082 bytes (19%) of program storage space.
Maximum is 1,044,464 bytes.

Global variables use 44,592 bytes (54%) of dynamic memory,
leaving 37,328 bytes for local variables. Maximum is 81,920
bytes.
```

Compile / Upload



sketch_apr19a | Arduino 1.8.5

File Edit Sketch Tools Help

sketch_apr19a

```
1 void setup() {  
2     pinMode(D0, OUTPUT);  
3 }  
4 void loop() {  
5     digitalWrite(D0, 1);  
6     delay(500);  
7     digitalWrite(D0, 0);  
8     delay(500);  
9 }  
10
```

Uploading...

```
espcomm_send_command: receiving 2 bytes  
writing flash
```

E1Module), 80 MHz, 4M (1M SPIFFS), v2 Prebuilt (MSS=536), Disabled, None, 115200 on COM89

Sketch Tools Help

Verify / Compile

Ctrl+R

Upload

Ctrl+U

Ctrl+U

Done uploading.

รูปแบบ Arduino



```
void setup()
```

```
{
```

สำหรับกำหนดค่า เกิดขึ้นครั้งเดียว

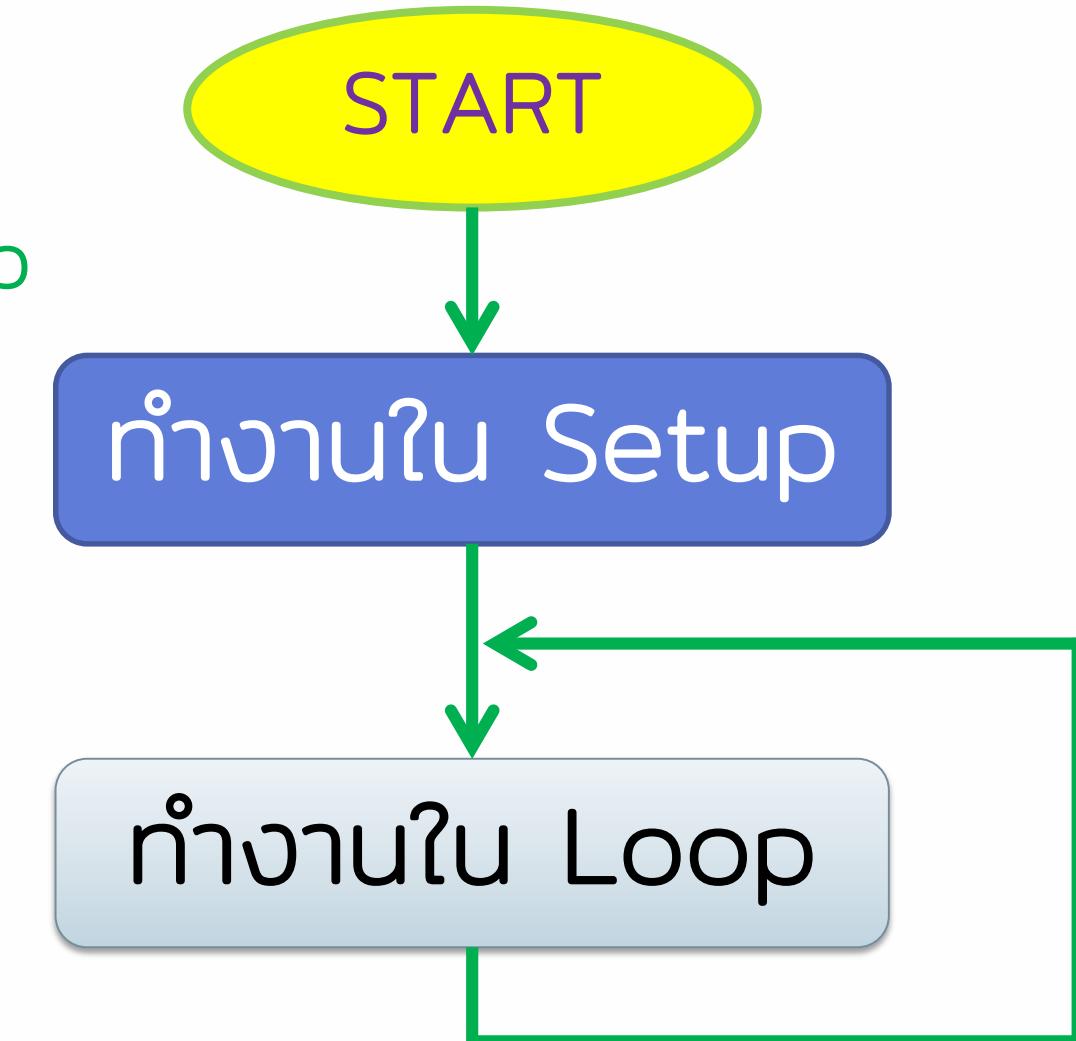
```
}
```

```
void loop()
```

```
{
```

โปรแกรมหลักทำงานต่อเนื่อง

```
}
```



กำหนดขาพอร์ตเป็นอินพุตหรือเอาต์พุต



pinMode (pin, Direction);

pin = ตำแหน่งขา D0-D10

Direction = OUTPUT , 1 กำหนดขาเป็นเอาต์พุต

Direction = INPUT , 0 กำหนดขาเป็นอินพุต

Direction = INPUT_PULLUP , 2 กำหนดขาเป็นอินพุตมีพูลอัปในตัว

pinMode (D0, 1); // กำหนดขาเป็นเอาต์พุต

รูปแบบ คำสั่ง digitalWrite



digitalWrite (pin , Logic) ;

pin = ตำแหน่งขา

Logic = HIGH , 1 เอาต์พุตlovจิก "1"

Logic = LOW , Ø เอาต์พุตlovจิก "Ø"

ตัวอย่าง

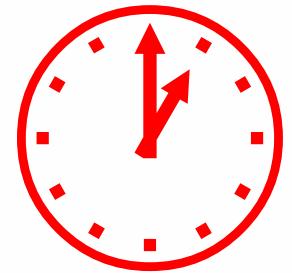
digitalWrite (D0 , HIGH) ;

รูปแบบ คำสั่ง delay



delay (DelayTime);

DelayTime ค่าเวลาหน่วงหน่วยเป็นมิลลิวินาที
ตัวอย่าง **delay (2000);** ค้างที่นี่นาน 2 วินาที



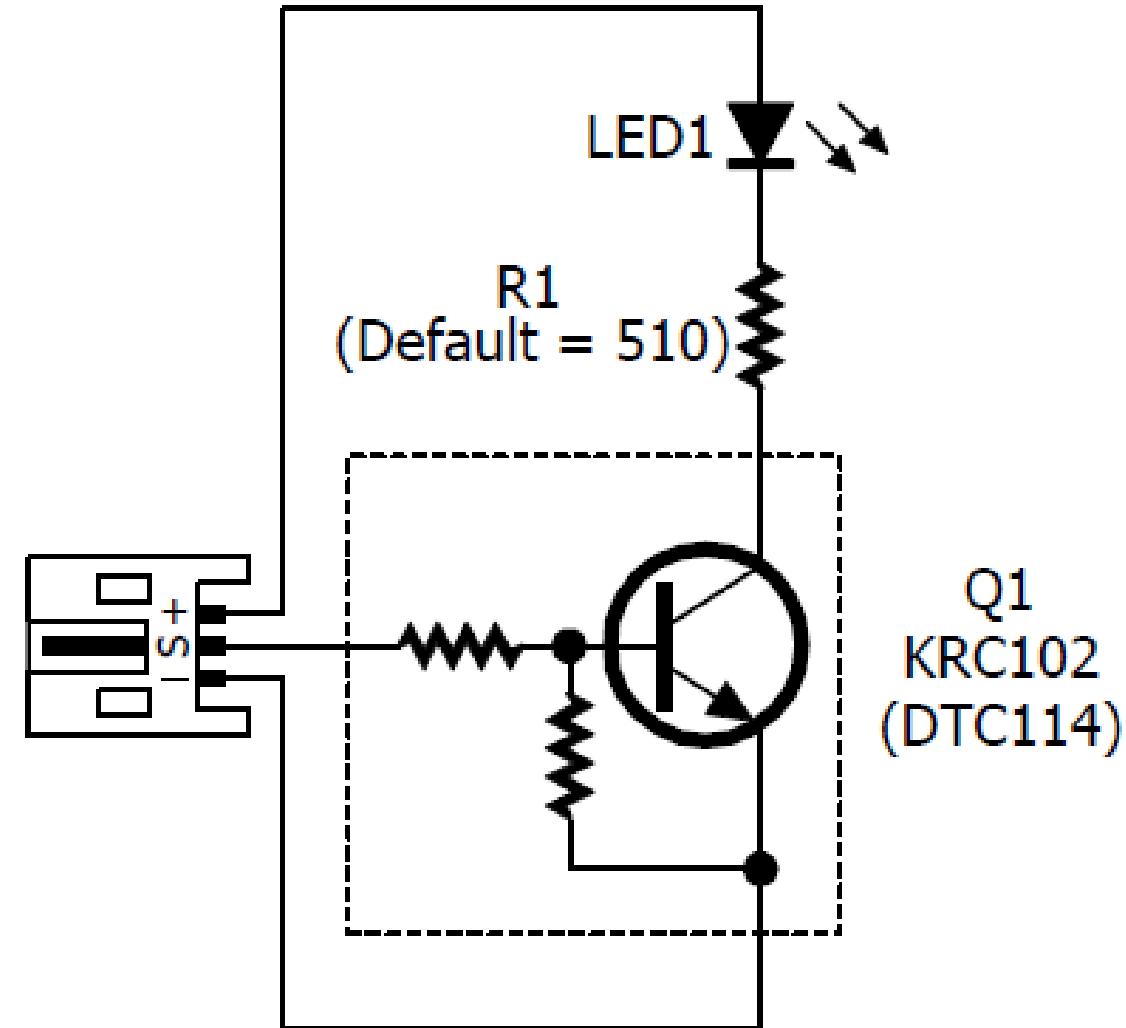
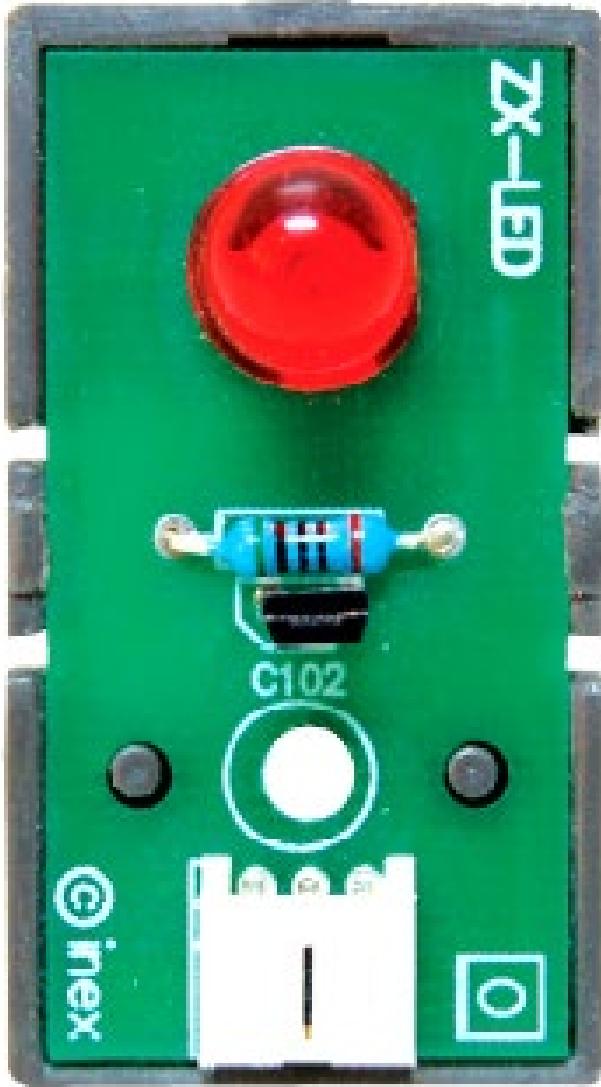
delayMicroseconds (time)

time ค่าเวลาหน่วงหน่วยเป็นไมโครวินาที

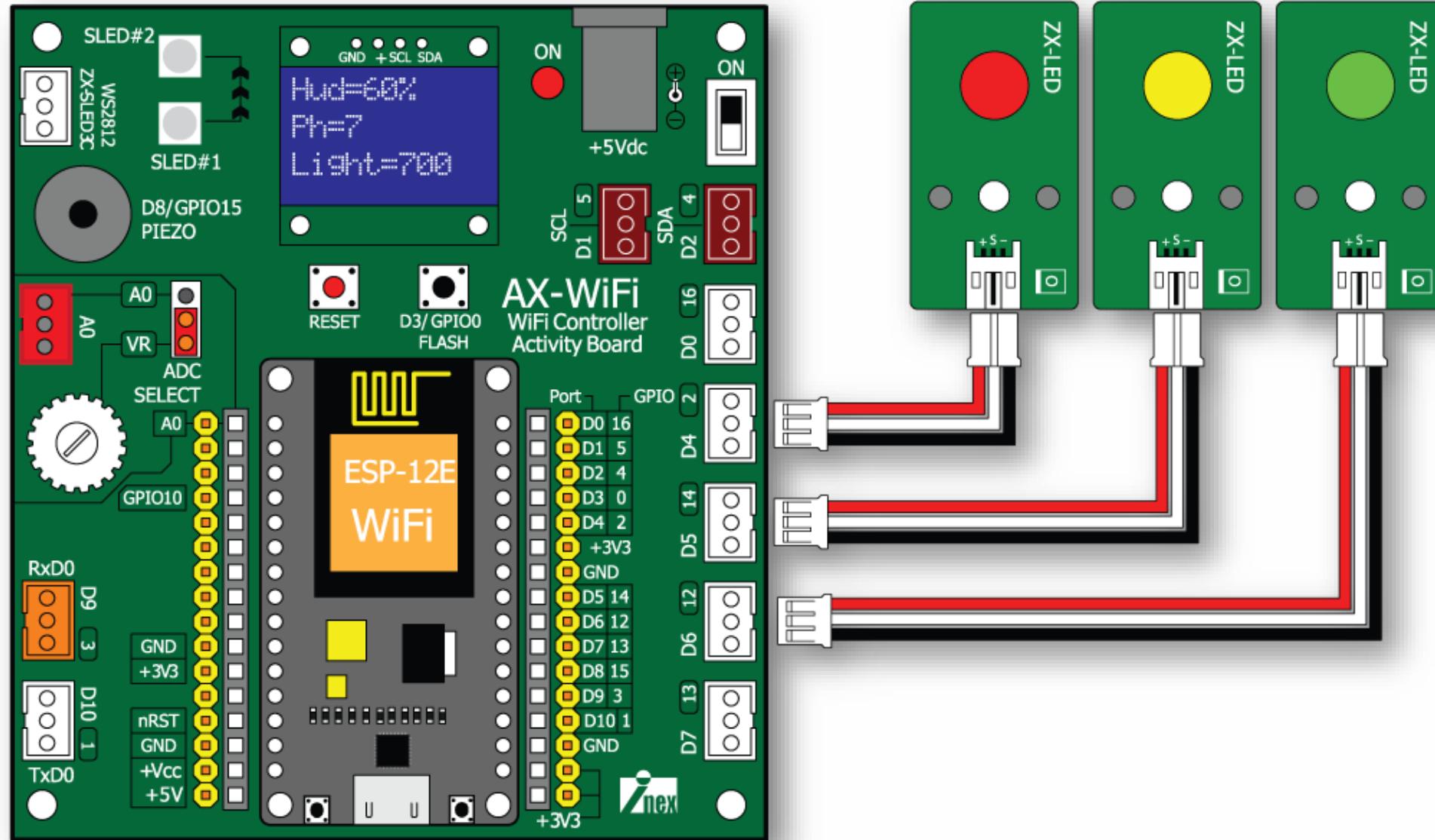
ตัวอย่าง **delayMicroseconds (300);**

ค้างที่นี่ 300 ไมโครวินาที

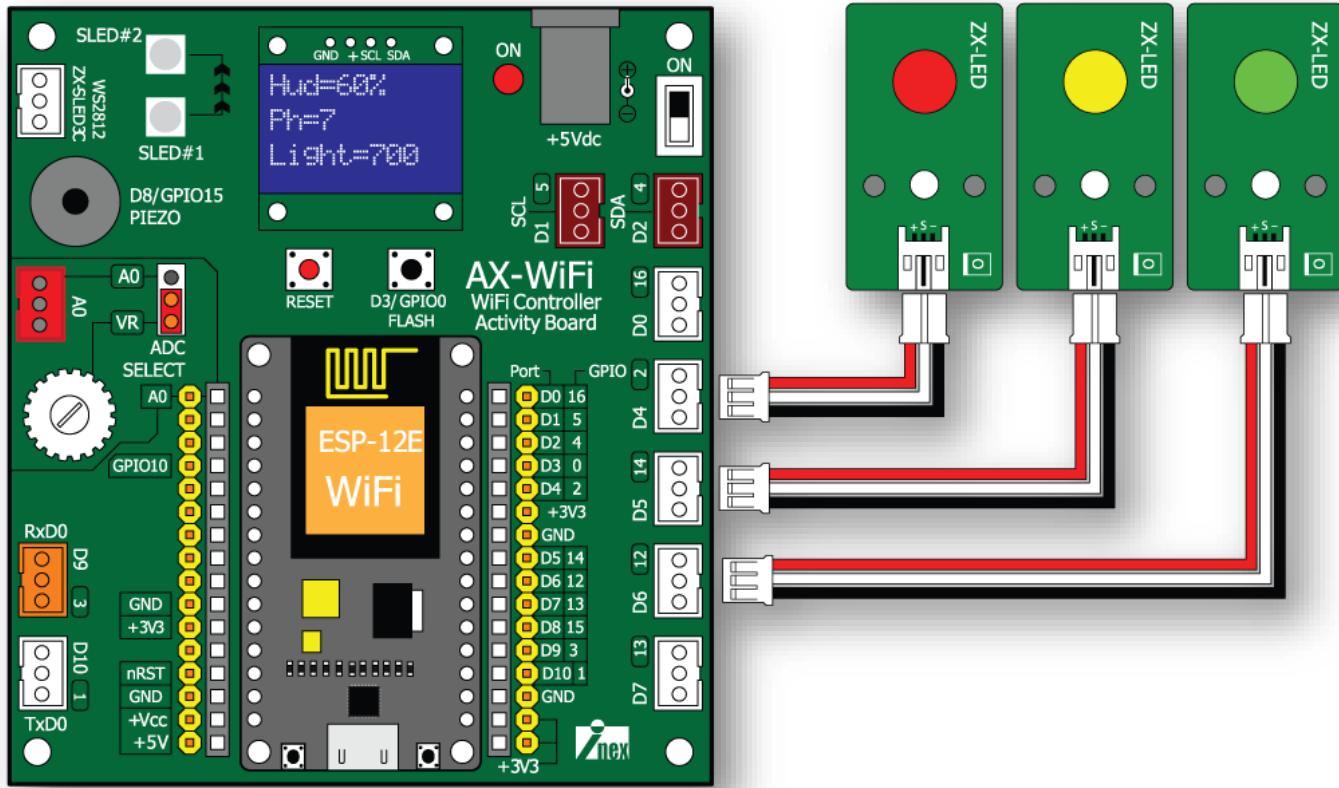
ZX-LED



ต่อ LED เข้ากับช่อง D4 , D5 และ D6



ໄວ່ງ 4 ດົກ ແລະ ພົບສຸດ



```
void setup() {  
    pinMode(D4, OUTPUT);  
    pinMode(D5, OUTPUT);  
    pinMode(D6, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(D4, 1);  
    digitalWrite(D5, 0);  
    digitalWrite(D6, 0);  
    delay(300);  
    digitalWrite(D4, 0);  
    digitalWrite(D5, 1);  
    digitalWrite(D6, 0);  
    delay(300);  
    digitalWrite(D4, 0);  
    digitalWrite(D5, 0);  
    digitalWrite(D6, 1);  
    delay(300);  
}
```

คำสั่งตรวจสอบเงื่อนไข for



รูปแบบ

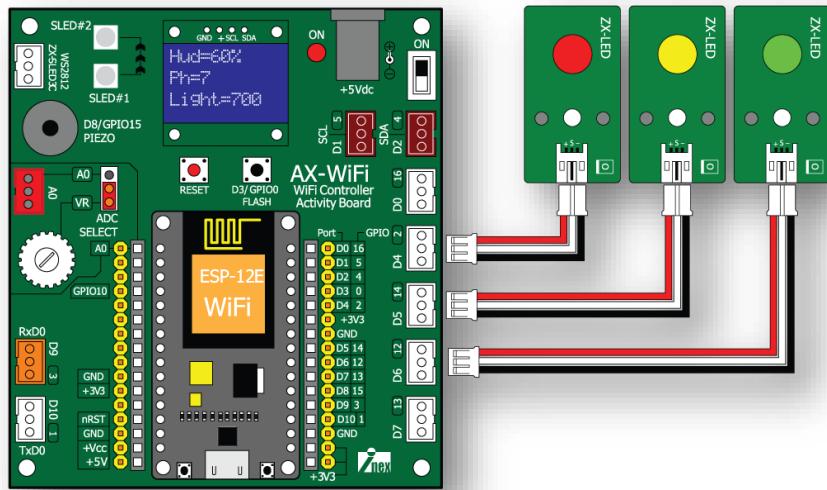
for (ค่าเริ่มต้น; เงื่อนไข ;ส่วนเพิ่มค่า)

{

 คำสั่ง วนครับเงื่อนไข;

}

ໄວ່ງ 3 ດາວ ແບບປັບປຸງ

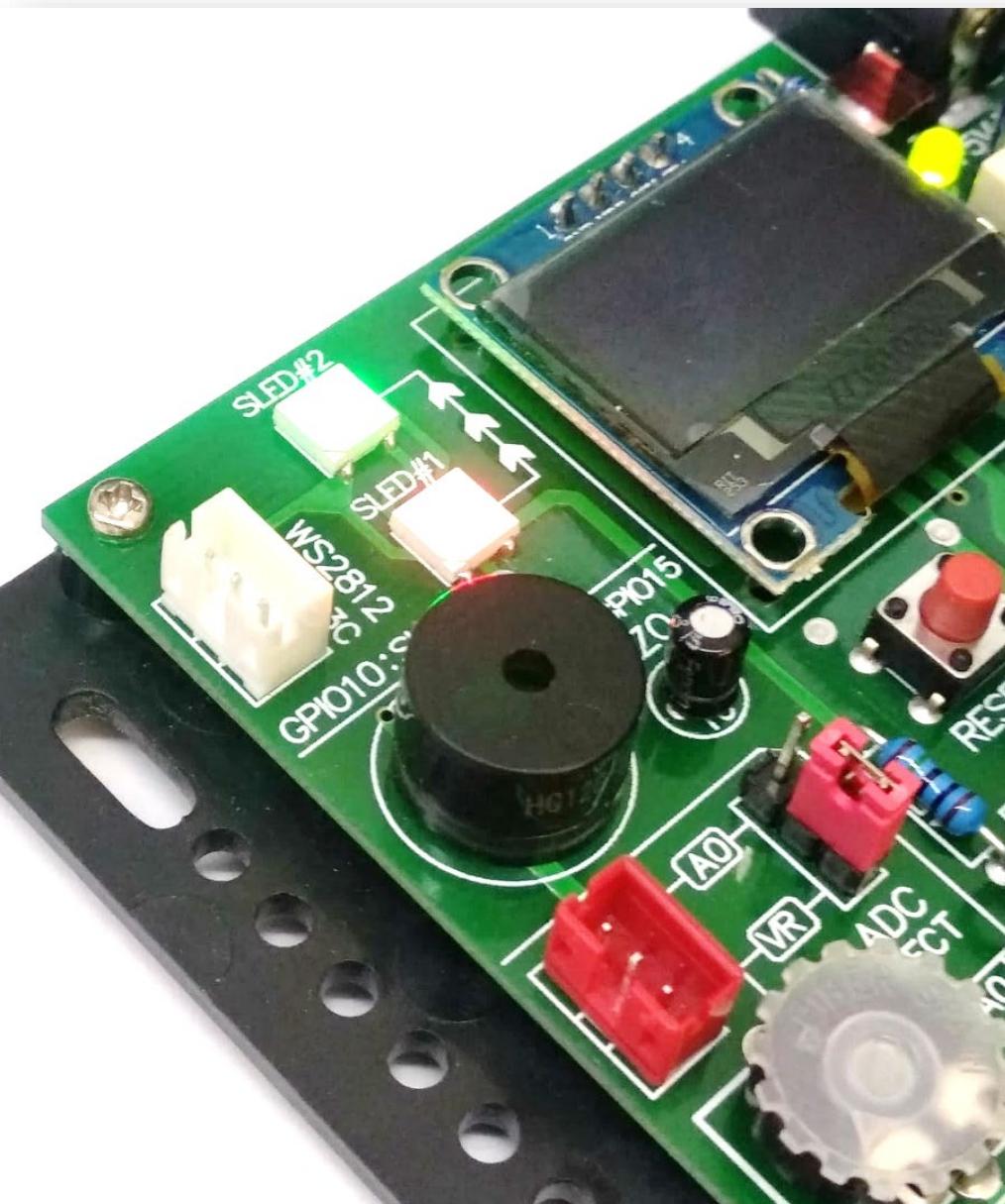


```
int pins []={D4 ,D5 ,D6} ;  
int i;  
void setup () {  
    for (i=0 ;i<3 ;i++) {  
        pinMode (pins [i] ,OUTPUT) ;  
    }  
}  
void loop () {  
    for (i=0 ;i<3 ;i++) {  
        digitalWrite (pins [i] ,HIGH) ;  
        delay (300) ;  
        digitalWrite (pins [i] ,LOW) ;  
        delay (300) ;  
    }  
}
```

ໄວ່ງ 3 ດົກ ແບບແກບ

```
int pins []={D4 ,D5 ,D6} ;  
int i;  
void setup () {  
    for (i=0;i<3;i++) {pinMode (pins [i] ,1) ; }  
}  
void loop () {  
    for (i=0;i<3;i++) {  
        digitalWrite (pins [i] ,1) ;delay (200) ;  
    }  
    for (i=0;i<3;i++) {  
        digitalWrite (pins [i] ,0) ;delay (200) ;  
    }  
}
```

ลำโพงเปียโซ



ติดตั้งกีต้าแม่น D8

- ใช้ลำโพงเปียโซ มีอัมพีแบนซ์ 32W
- มีค่าความถี่ย่าน 300Hz ถึง 3000 Hz

สร้างเสียง

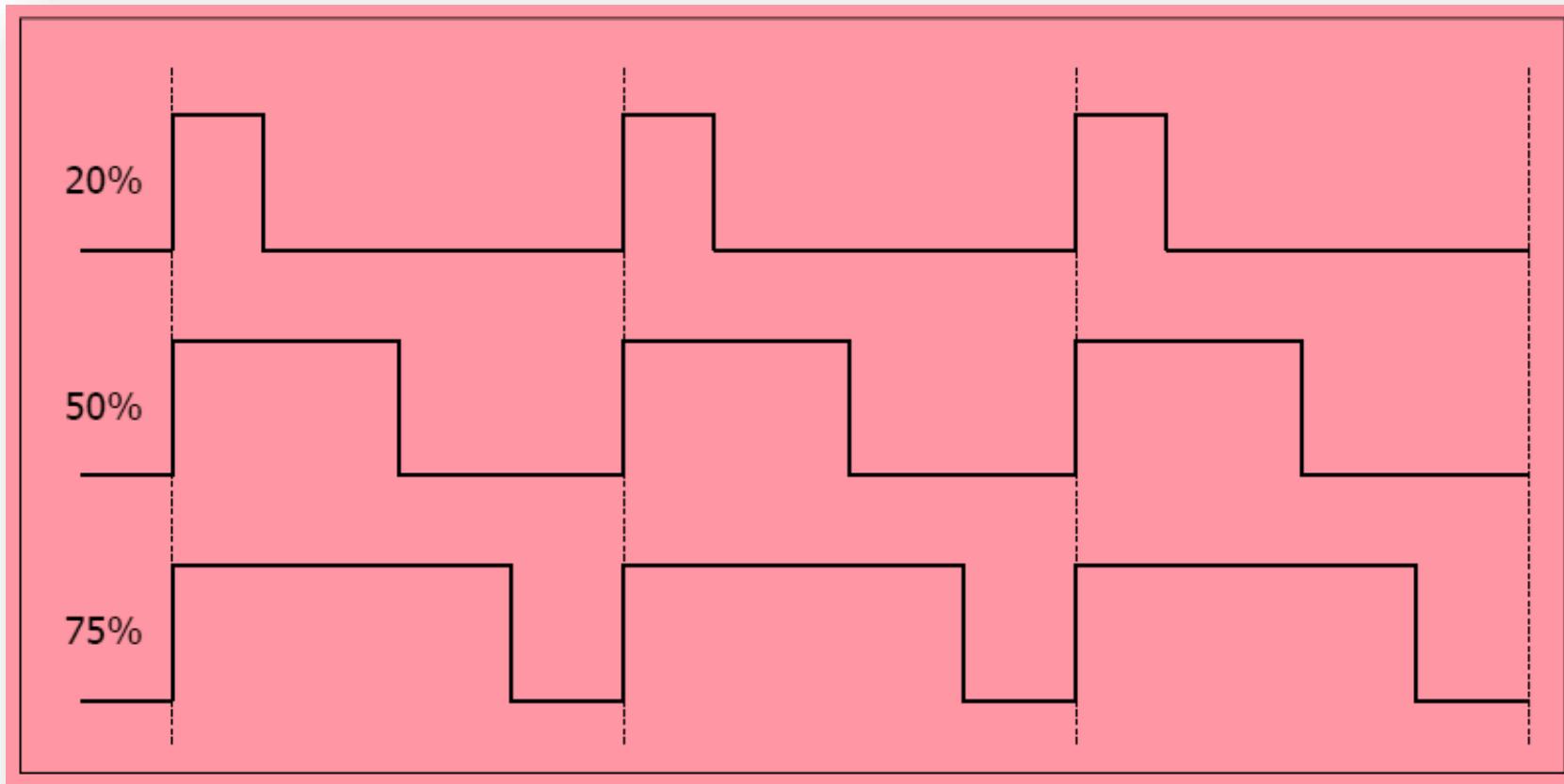


```
void setup() {  
    pinMode(D8, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(D8, 1);  
    delayMicroseconds(300);  
    digitalWrite(D8, 0);  
    delayMicroseconds(300);  
}
```

$f=1/600 \text{ usec}$

$f=1666.6 \text{ Hz}$

PWM : Pulse Width Modulator



`analogWrite(pin,DutyCycle);`

ความถี่ Fix 1kHz

NodeMCU PWM ทุกขา

ส่งค่าออกเข้าต์พุตแบบ อะนาลอก

analogWrite (pin , PWM) ;

pin = ตำแหน่งขา D0-D10

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

ความถี่ PWM ประมาณ 1kHz

ปรับค่าความถี่

analogWriteFreq (Freq) ;

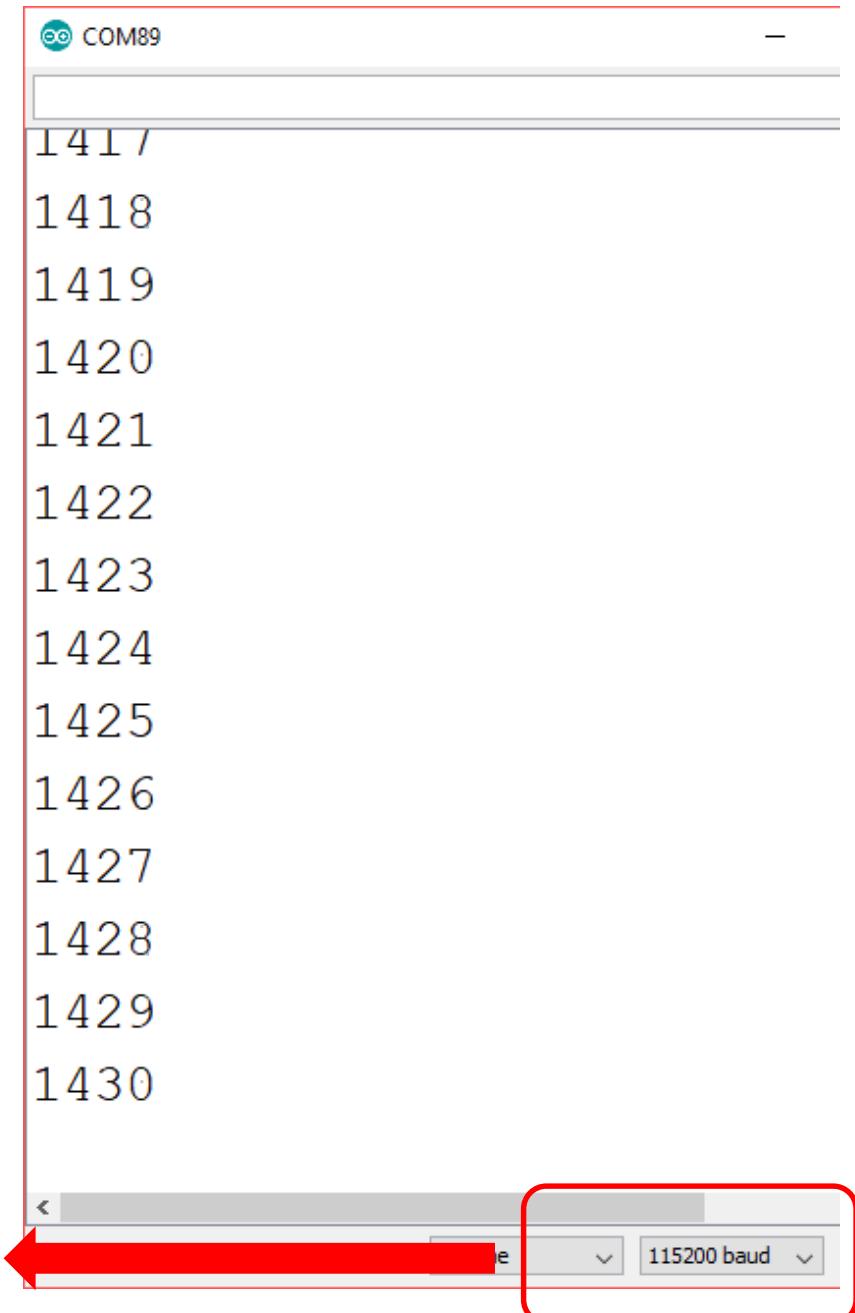
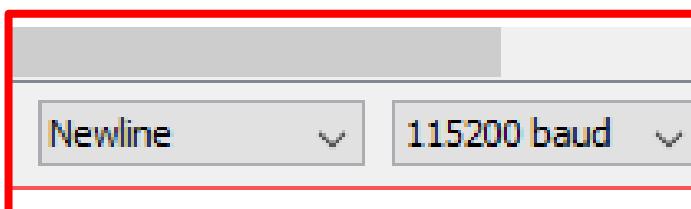
Freq = 18-60,000 Hz

ค่าดิวตี้ไซเก็ล 50%

```
void setup() {  
    analogWrite(D8, 512);  
}  
void loop() {}
```

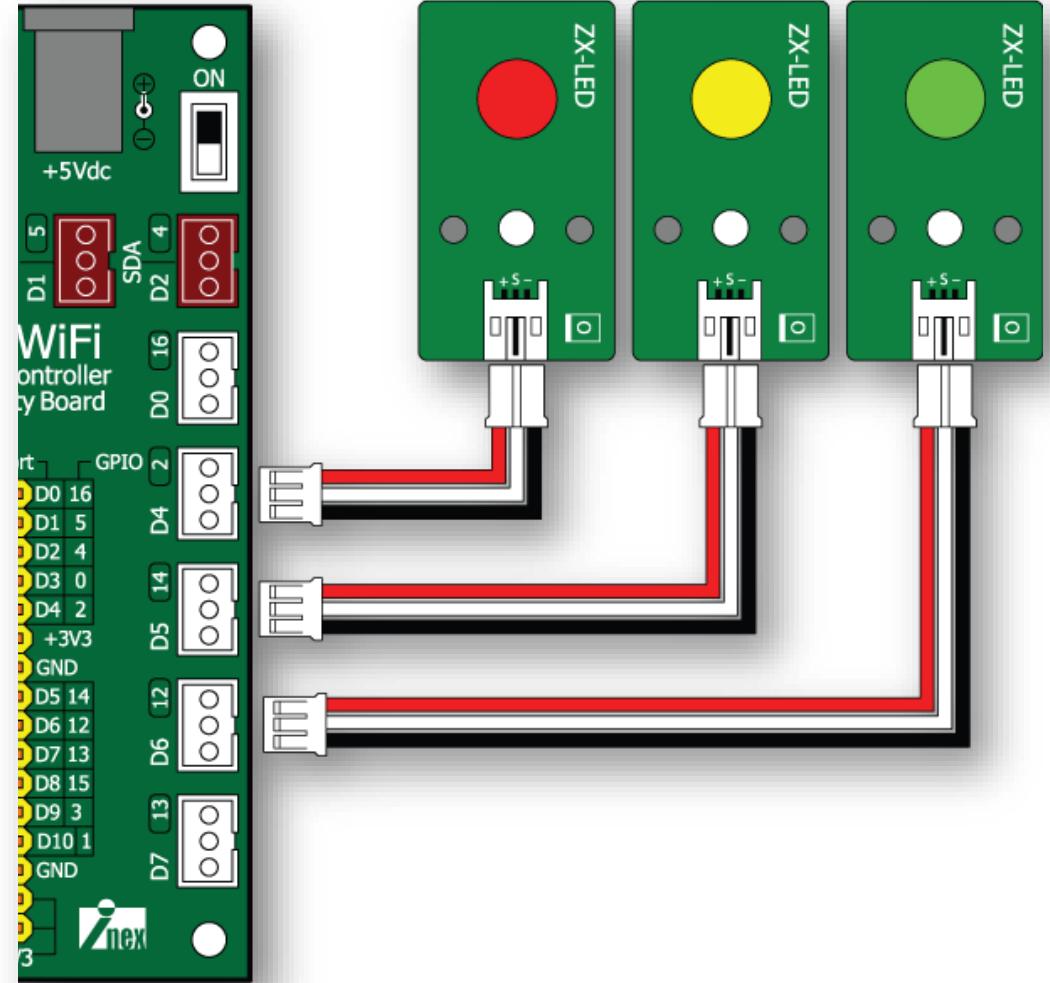
ทดลองเปลี่ยนความถี่

```
int i;  
void setup() {  
    Serial.begin(115200);  
    analogWrite(D8, 512);  
}  
  
void loop() {  
    for (i=100;i<20000;i++) {  
        analogWriteFreq(i);  
        Serial.println(i);  
        delay(100);  
    }  
}
```



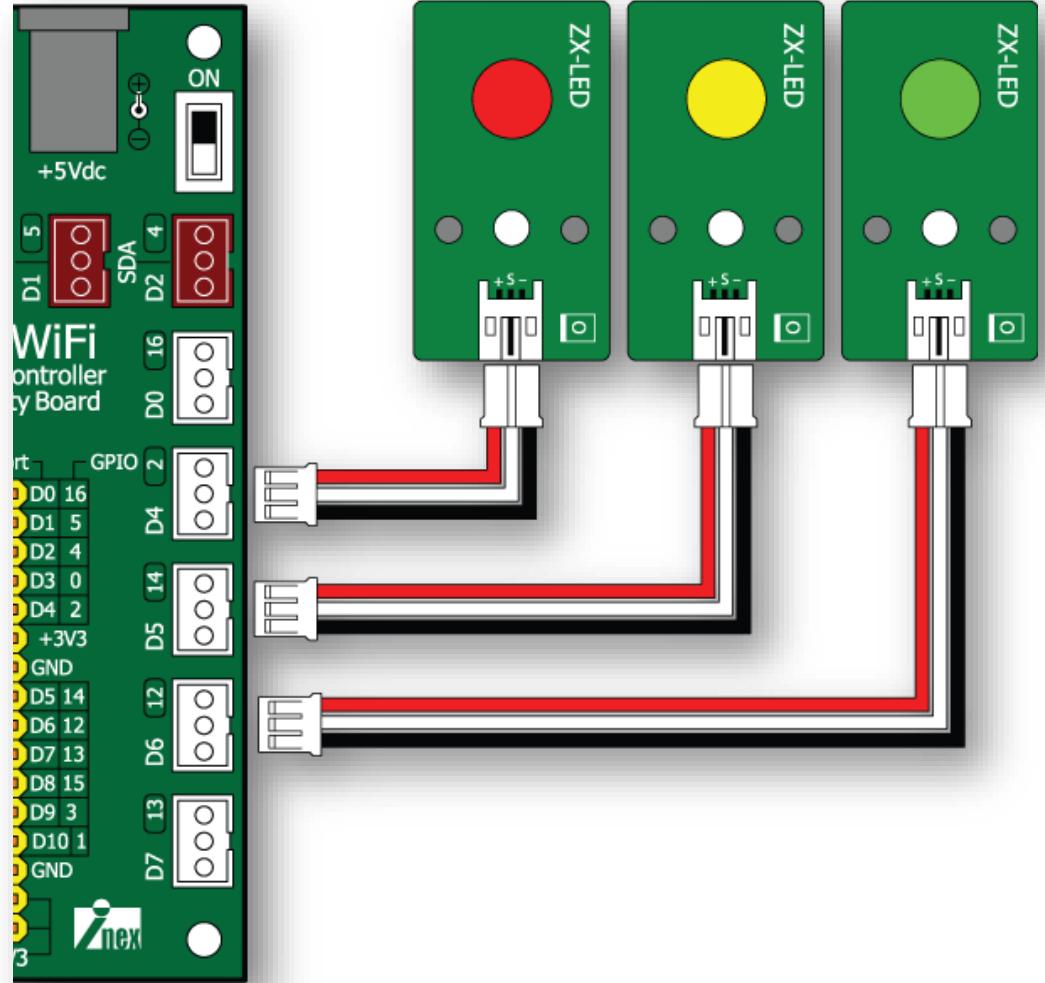
ทดลองเปลี่ยนดิวตี้ไซเกิล

```
int i;  
void setup() {  
    Serial.begin(115200);  
    analogWrite(D6, 512);  
}  
  
void loop() {  
    for (i=0; i<1023; i++) {  
        analogWrite(D6, i);  
        Serial.println(i);  
        delay(10);  
    }  
}
```



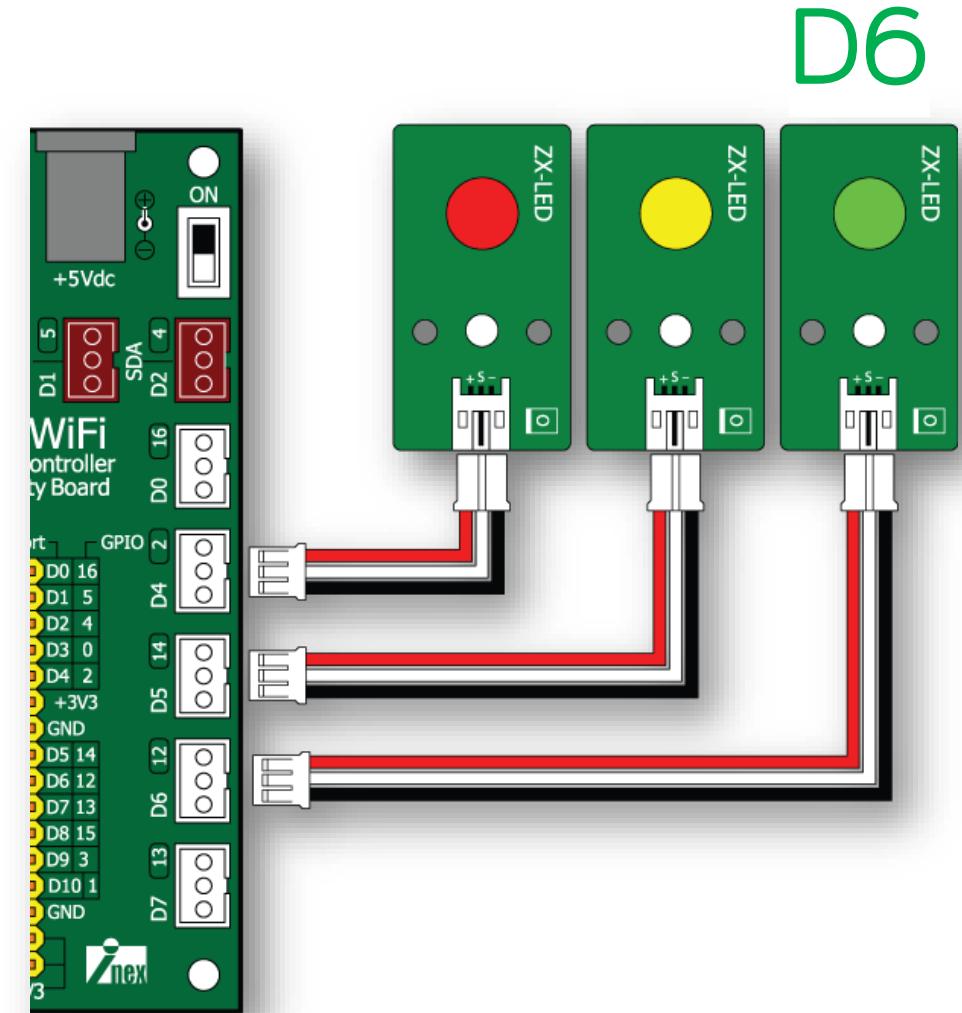
ทดลองเปลี่ยนดิวตี้ไซเกิล

```
int i;  
void setup() {  
    Serial.begin(115200);  
    analogWrite(D6, 512);  
}  
  
void loop() {  
    for (i=0; i<1023; i++) {  
        analogWrite(D6, i);  
        Serial.println(i);  
        delay(10);  
    }  
}
```



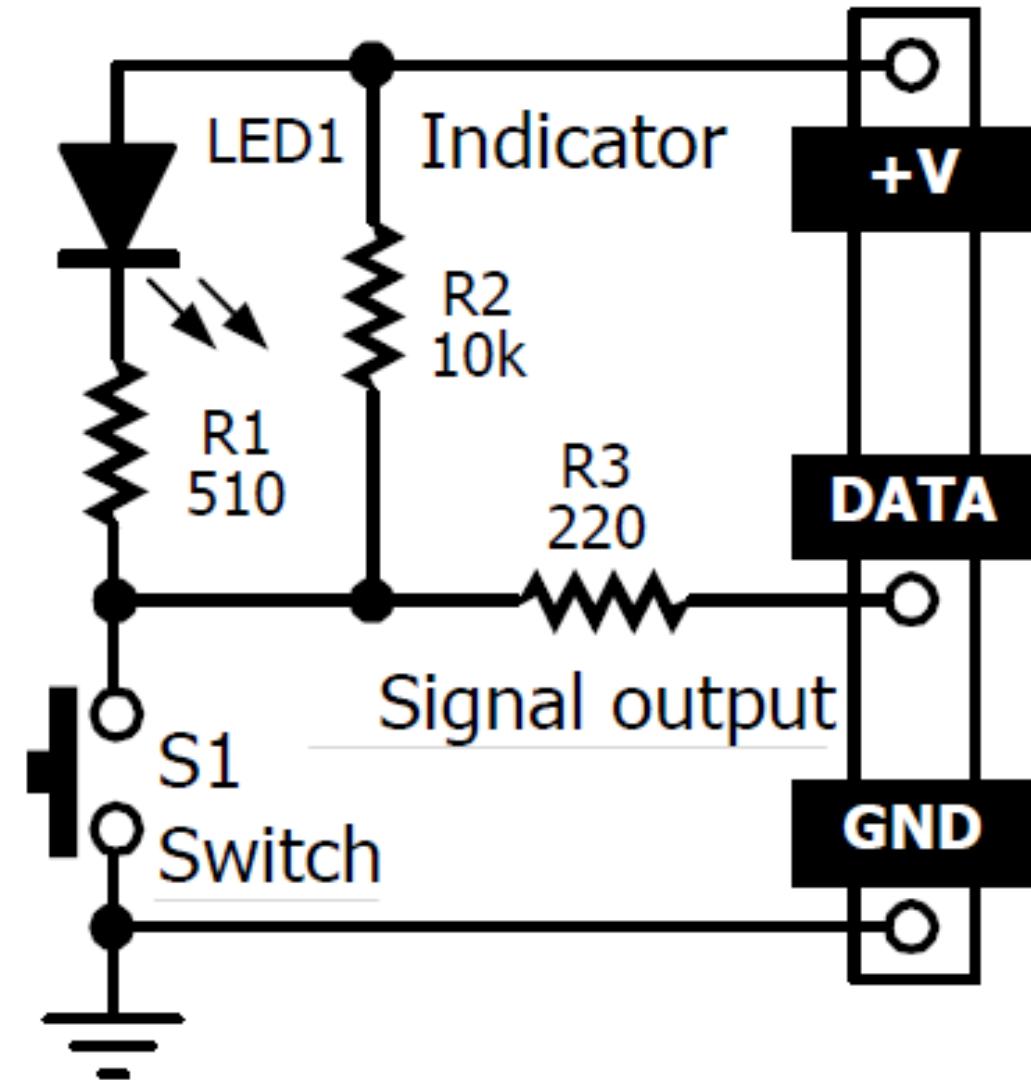
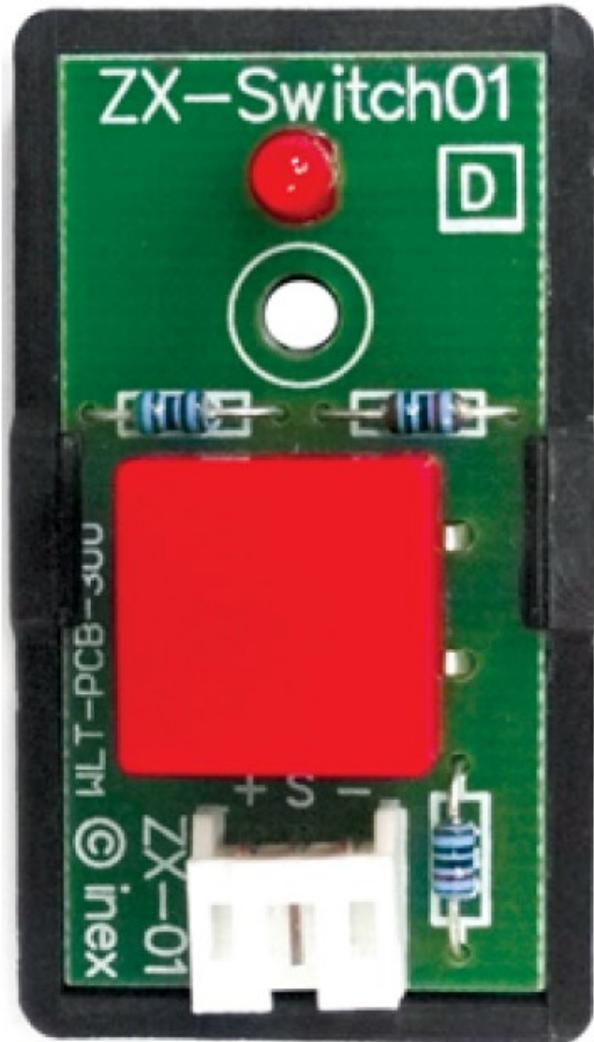
ໄວ້ງູທີ່ຂາ D6

```
int i;  
void setup() {}  
void loop() {  
    for(i=0; i<=1023; i+=5) {  
        analogWrite(D6, i);  
        delay(3);  
    }  
    for(i=1023; i>0; i-=5) {  
        analogWrite(D6, i);  
        delay(3);  
    }  
}
```



D6

សុចាប ZX-Switch01



กำหนดขาพอร์ตเป็นอินพุตหรือเอาต์พุต

pinMode (pin , Direction) ;

pin = ตำแหน่งขา D0-D10

Direction = OUTPUT , 1 กำหนดขาเป็นเอาต์พุต

Direction = INPUT , 0 กำหนดขาเป็นอินพุต

Direction = INPUT_PULLUP , 2 กำหนดขาเป็นอินพุต มีพูลอัปในตัว

pinMode (D0 , 0) ; // กำหนดขาเป็นอินพุต

ຕົວແປຣກີ່ໃຊ້ໃນ Arduino

byte 0-255 (unsigned char)

word 0-65535 (unsigned int)

boolean 0-1 **True** **False**

int -32768 ກຶ່ງ 32767

char -128 ກຶ່ງ 127

float -3.4×10^{38} ກຶ່ງ 3.4×10^{38}

າຂັ້ວມູລເພີ່ມເຕີມຈາກ reference

รูปแบบ คำสั่ง digitalRead

x=digitalRead(pin);

x = ตัวแปร Boolean กี่นำมารับค่า

pin = ขาอินพุตดิจิตอล

ตัวอย่าง

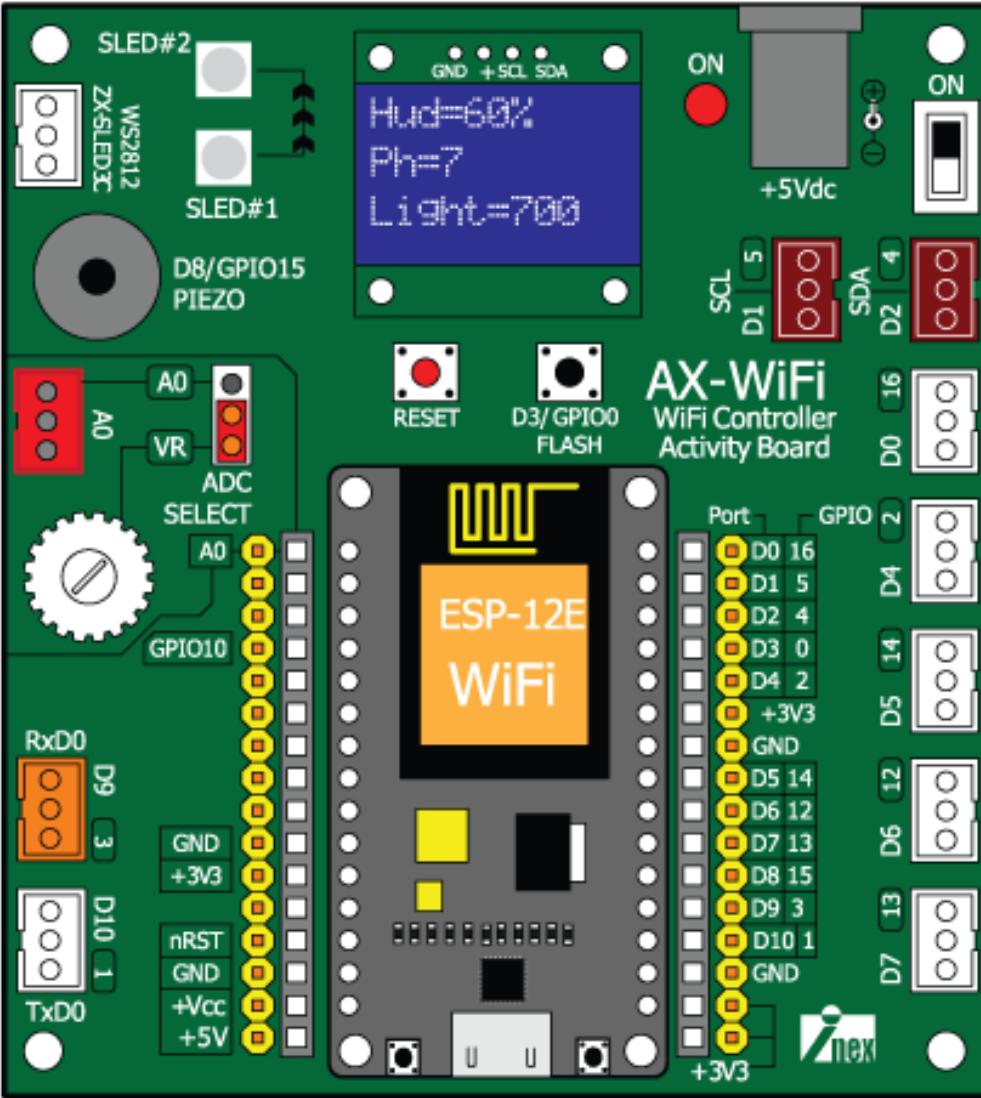
```
x=digitalRead(D3);
```

```
if (x==0) {
```

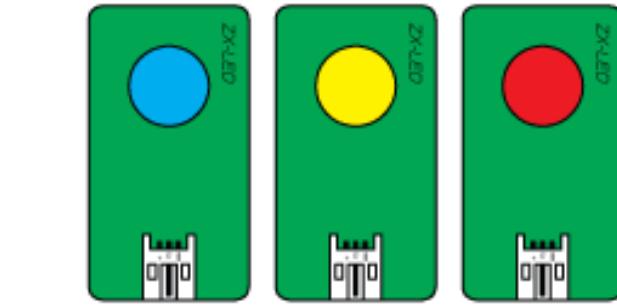
```
    digitalWrite(D5, HIGH)
```

```
}
```

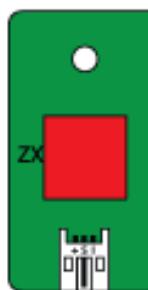
เชื่อมต่อ LED และ Switch



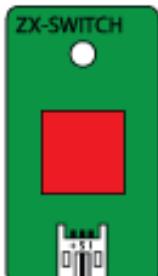
D0 D4 D5



D6



D7



ตัวอย่าง digitalRead()

```
void setup() {  
    pinMode(D4, OUTPUT);  
    pinMode(D6, INPUT);  
}  
  
void loop() {  
    if(digitalRead(D6)) {  
        digitalWrite(D4, HIGH);  
    }  
    else{  
        digitalWrite(D4, LOW);  
    }  
}
```

คำสั่งตรวจสอบเงื่อนไข if-else

รูปแบบ

```
if (เงื่อนไข)
{
    คำสั่ง1 ทำเมื่อเงื่อนไขเป็นจริง;
}
else
{
    คำสั่ง2 ทำเมื่อเงื่อนไขเป็นเท็จ;
}
```

សុពល កណ្តាល/កណ្តប់ (ខ្សោ Delay)

```
boolean st=0;
void setup() {
    pinMode(D1, INPUT);
    pinMode(D4, OUTPUT);
}
void loop() {
    if(!digitalRead(D1)) {
        delay(200);
        if(st) {
            digitalWrite(D4, HIGH); st=0;
        }
        else{
            digitalWrite(D4, LOW); st=1;
        }
    }
}
```

คำสั่งตรวจสอบเงื่อนไข while

รูปแบบ

while(เงื่อนไข)

{

 คำสั่งวนเงื่อนไขเป็นเก็จ;

}

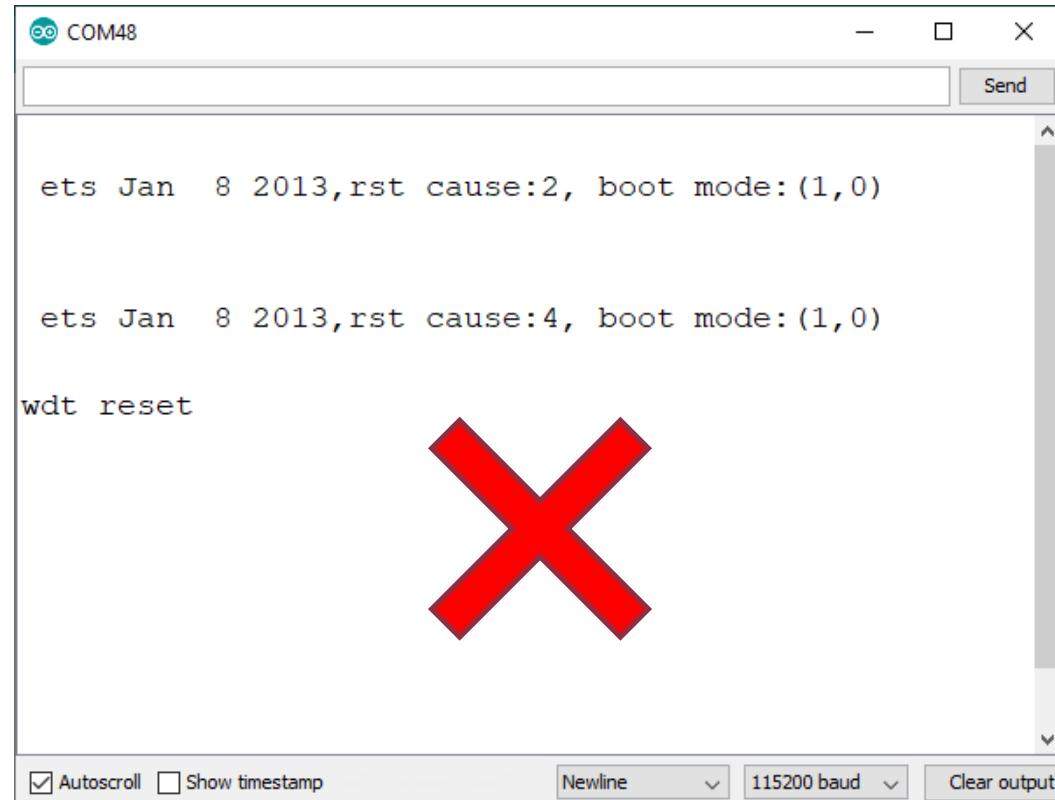
การใช้งาน While หลุดออกด้วย break

```
while(1)
{
    คำสั่งที่ 1;
    คำสั่งที่ 2;
    if(เงื่อนไขเป้าหมาย)
    {
        break;
    }
}
```

ปกติโปรแกรมจะวนคำสั่ง 1 และ 2 รวมถึงตรวจสอบเงื่อนไข if อย่างต่อเนื่อง จนกระทั่งพบว่า เงื่อนไขเป้าหมายเป็นจริง จึงจะหลุดออกจากลูป while ด้วยคำสั่ง **break**

while(1); // โปรแกรมค้างที่บันกัดคำสั่งนี้ตลอดกาล

สวิตซ์ กดติด/กดดับ (ใช้ while)



WDT รีเซ็ต เนื่องจากค้างที่ while นาน

```
boolean st=0;  
void setup() {  
    pinMode(D7, INPUT);  
    pinMode(D4, OUTPUT);  
}  
void loop() {  
    if(!digitalRead(D7)){  
        while(!digitalRead(D7));  
        delay(200);  
        if(st){  
            digitalWrite(D4, HIGH); st=0;  
        }  
        else{  
            digitalWrite(D4, LOW); st=1;  
        }  
    }  
}
```

การรับส่งข้อมูล Serial กับคอมพิวเตอร์

`Serial.begin(BAUD)` เริ่มต้นใช้งานการสื่อสาร BAUD คือบอเดตที่ใช้ในการสื่อสาร

`Serial.write(1Byte)` ส่งข้อมูล 1 ไบต์, หรือข้อความง่ายๆ

`Serial.print(Text)` ส่งข้อความพร้อมพารามิเตอร์ต่างๆ

`Serial.println()` ส่งข้อความพร้อมพารามิเตอร์ต่างๆ 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";

`Serial.available()`

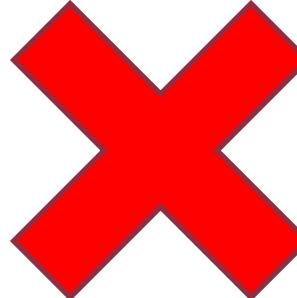
ตรวจสอบว่ามีข้อมูลหรือไม่

`Serial.read()`

อ่านข้อมูล

ສົວຕະໜີ ກາດຕິດ/ກາດດັບ (ໃຊ້ while)

```
boolean st=0;
void setup() {
    Serial.begin(115200);
    pinMode(D7, INPUT);
    pinMode(D5, OUTPUT);
    Serial.println("Hello");
}
void loop() {
    Serial.println("x");
    delay(200);
    if(!digitalRead(D7)) {
        while(!digitalRead(D7));
        if(st) {
            digitalWrite(D5, HIGH); st=0;
        }
        else{
            digitalWrite(D5, LOW); st=1;
        }
    }
}
```



COM48

|

x

x

x

x

x

x

Soft WDT reset

ctx: cont

sp: 3ffef530 end: 3ffef720 offset: 01b0

>>>stack>>>

3ffef6e0: 40201356 000000c8 000000c8 401068d2
3ffef6f0: 00000000 00000000 3ffee6e4 402020b0
3ffef700: 3ffffdad0 00000000 3ffee6e4 40202478
3ffef710: feeffeffe feeffeffe 3ffee700 40100710

<<<stack<<<

ets Jan 8 2013,rst cause:2, boot mode:(1,6)

ets Jan 8 2013,rst cause:4, boot mode:(1,6)

wdt reset

Autoscroll Show timestamp

Newline 115200 baud Clear output

การแก้สวิตซ์ กดติด/กดดับ (ใช้ while)

```
boolean st=0;  
void setup() {  
    Serial.begin(115200);  
    pinMode(D1, INPUT);  
    pinMode(D4, OUTPUT);  
    Serial.println("Hello");  
}  
void loop() {  
    Serial.println("x"); delay(200);  
    if(!digitalRead(D1)) {  
        while(!digitalRead(D1)) {delay(1);}  
        if(st) {  
            digitalWrite(D4, HIGH); st=0;  
        }  
        else {  
            digitalWrite(D4, LOW); st=1;  
        }  
    }  
}
```

เพิ่มคำสั่ง



ຝຶກຂັ້ນ millis()

millis ຈະເພີ່ມຄ່າຂັ້ນທີລະ 1 ຖຸກ ຣັງ 1 ມີລສິວນາກີ

millis ຈະກຳຈານວຍໆເບື້ອງຮັບດ້ວຍ Interrupt Timer

millis ໃຊ້ຕັວແປຣ Long ໃນກາຣເກີບຄ່າ (4 ໄບຕົ)

4,294,967,296 ມີລສິວນາກີ ກ່ອນເຄລີຍຮົບືນ 0

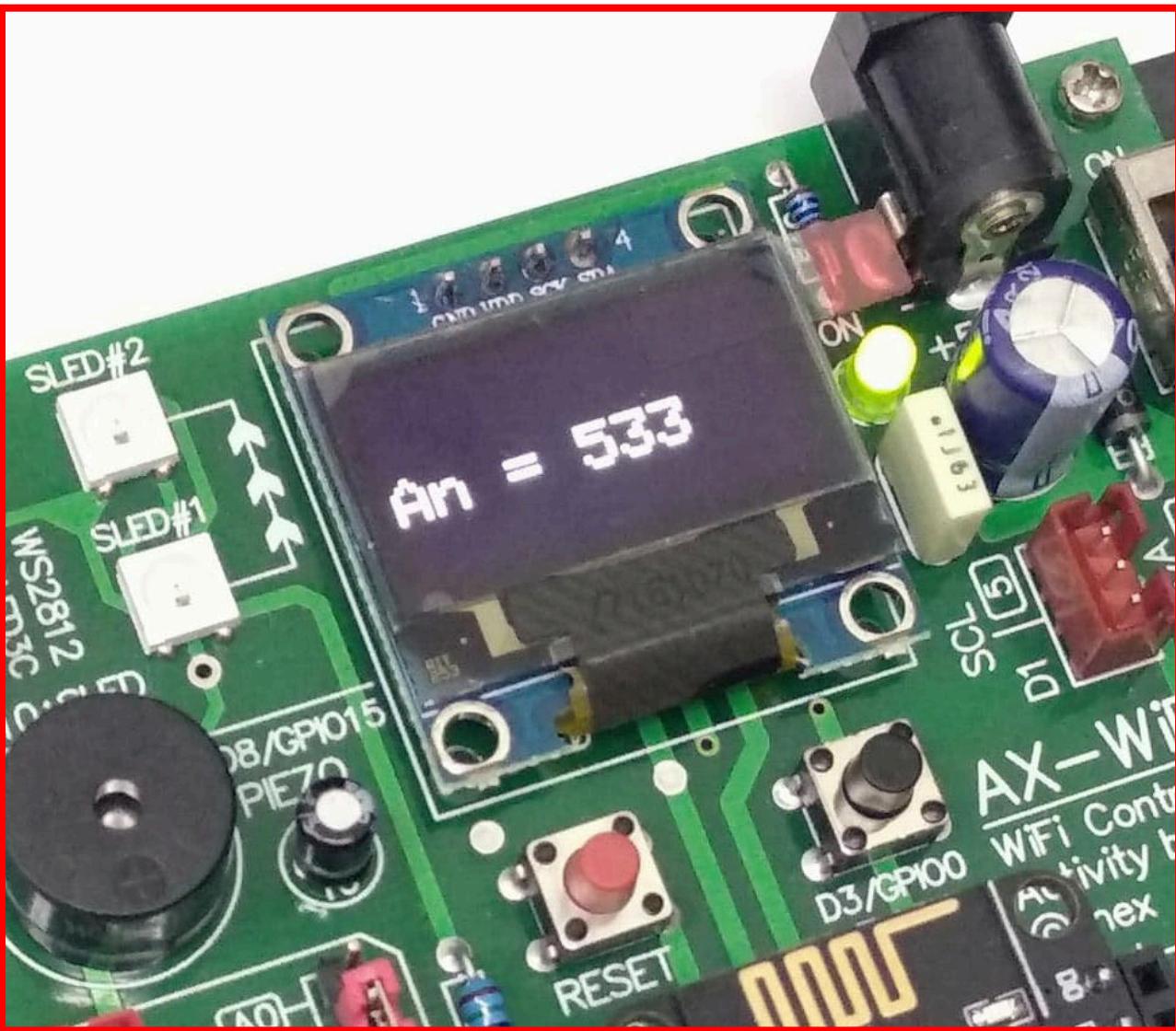
ตัวอย่างการใช้ millis()

```
int interval1,interval2;
bool S1=0,S2=0,S3=0;
void setup() {
    pinMode(D4,OUTPUT);
    pinMode(D5,OUTPUT);
    interval1=millis();
    interval2=millis();
}
void loop() {
    if (millis()>interval1+350){
        interval1=millis();
        if(S1){digitalWrite(D4,HIGH);S1=0;} 
        else{digitalWrite(D4,LOW);S1=1;}
    }
    if (millis()>interval2+400){
        interval2=millis();
        if(S2){digitalWrite(D5,HIGH);S2=0;} 
        else{digitalWrite(D5,LOW);S2=1;}
    }
}
```

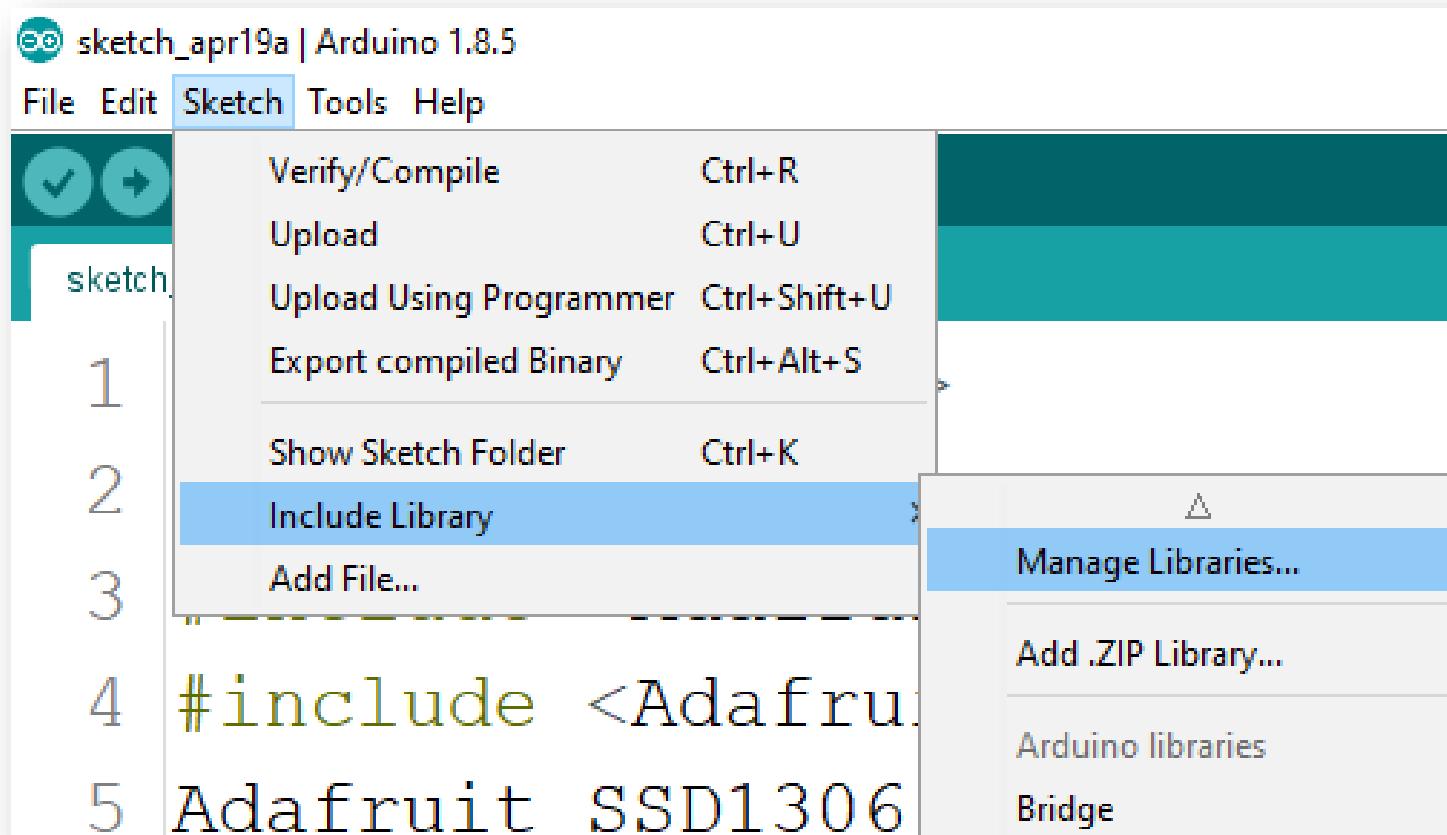
ไฟกระพริบไม่วร้อนกัน

จอแสดงผล OLED 128x64 จุด

- จอแสดงผล Ø.96 นิ้ว
- ใช้ชิปเบอร์ SSD1306
- ความละเอียด 128 × 64 จุด
- ให้แสงสีขาวหรือฟ้า
- ติดต่อผ่านระบบบัส I2C
- ไฟเลี้ยง +3.3V 20mA
- มีวงจรกวีแรงดัน +4.5V เพื่อขับ LED

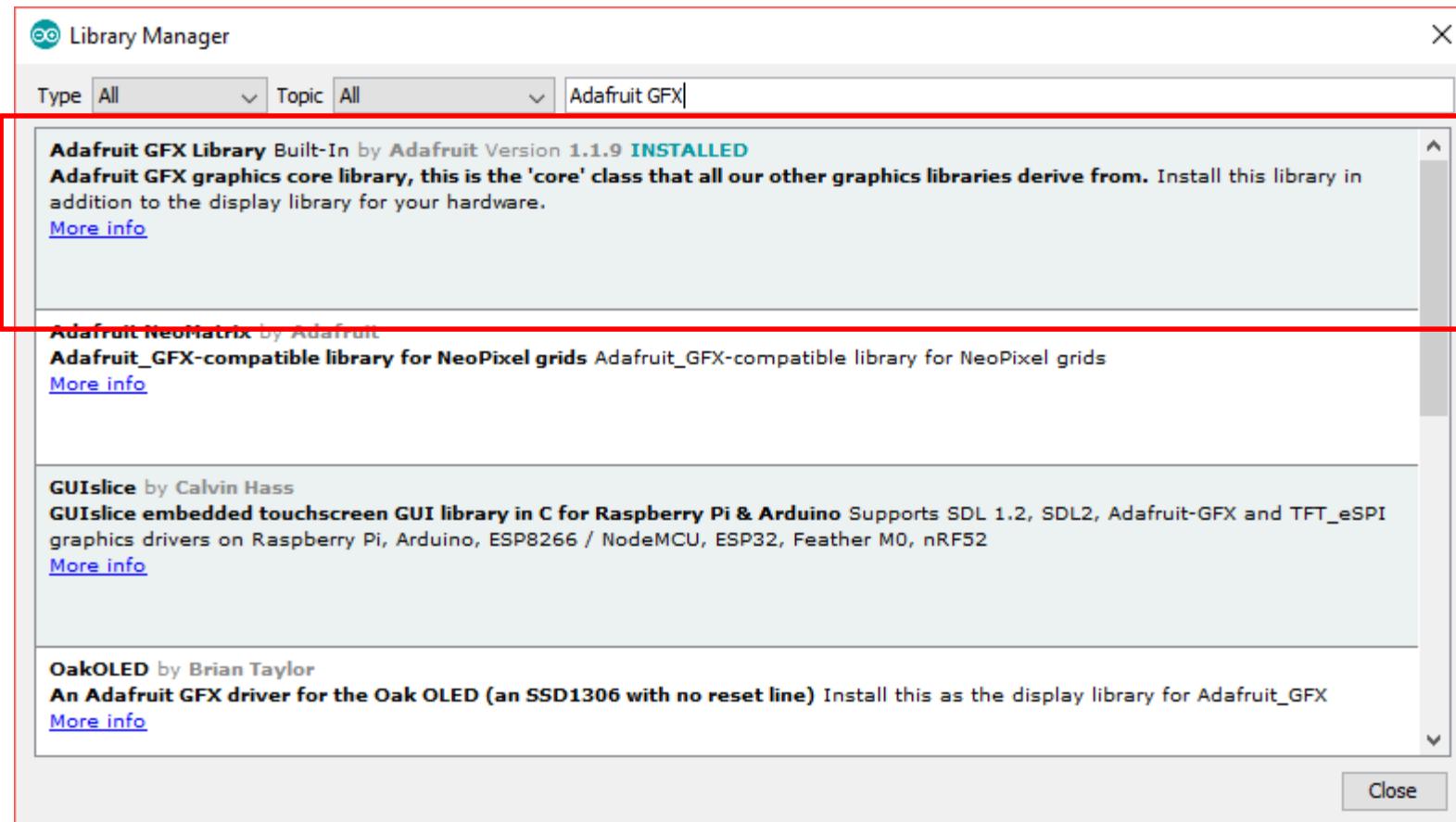


1. ผนวกไลบรารีเพื่อติดต่อ GLCD Adafruit_GFX.h



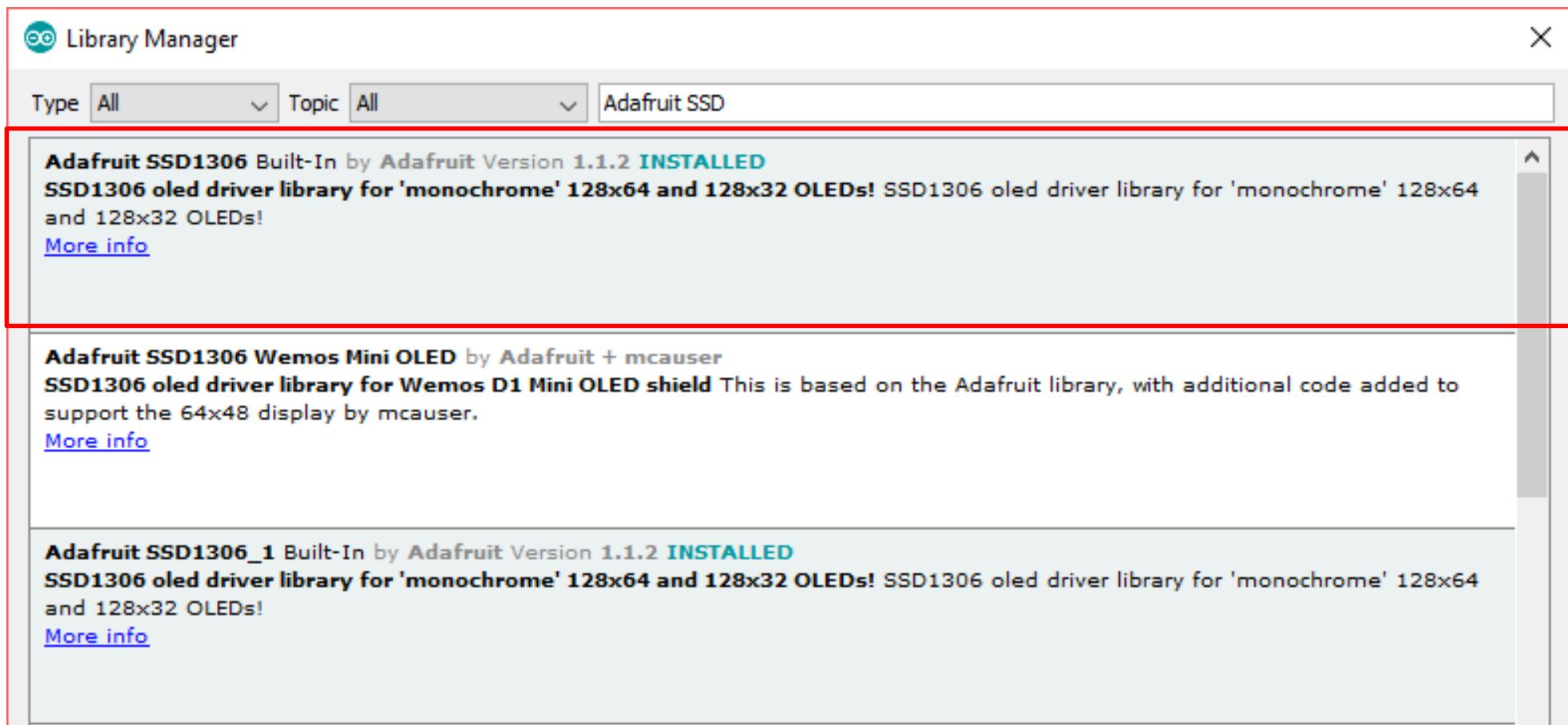
เชื่อมต่อ Internet และเข้าที่ Manage Libraries...

Library Manager ค้นหา Adafruit_GFX.h



พิมพ์ค้นหา Adafruit GFX (ถ้าติดตั้งจาก inex จะมีอยู่แล้ว)
ถ้าไม่มีให้กด Install เพื่อติดตั้ง

ไฟล์ライบรารี Adafruit_SSD1306.h



พิมพ์ค้นหา Adafruit SSD (ถ้าติดตั้งจาก inex จะมีอยู่แล้ว)

ถ้าไม่มีให้กด Install เพื่อติดตั้ง

พิงก์ชิ้นใช้งาน OLED

OLED.begin(MODE,ADDR)

กำหนดค่าเตรียมความพร้อม

MODE ซึ่ปกี่ใช้เลือกเป็น SSD1306_SWITCHCAPVCC

ADDR แอดเดรสของโมดูล OLED ตัวนี้คือ 0x3C

clearDisplay()

เคลียร์หน้าจอ

setCursor(x,y)

ระบุตำแหน่งพิกัด X,Y กี่จะแสดง

X พิกัดแนวนอน 0-127

Y พิกัดแนวตั้ง 0-63

พังก์ชั่นใช้งาน OLED

setTextSize(s) กำหนดขนาดตัวอักษร

S ขนาดตัวอักษรเป็นจำนวนเท่า 1,2,3

กำหนดขนาด 1 ใช้พื้นที่ 6x8 พิกเซล

display() เริ่มแสดงข้อความที่กำหนดไว้ก่อนหน้า

drawBitmap (x,y,*bitmap,w,h,color)

แสดงจุดภาพที่เก็บอยู่ในรูปแบบอาร์ray *bitmap

X พิกัดแนวอน 0-127

w ความกว้างภาพ

Y พิกัดแนวตั้ง 0-63

h ความสูงภาพ

color สีของภาพ

ຝຶກໜີໃຊ້ງານ OLED

println() ສັ່ງພິມພົບຂ້ອຄວາມຕັ້ງອັກເປຣຂຶ້ນບຣກັດໃໝ່

print() ສັ່ງພິມພົບຂ້ອຄວາມຕັ້ງອັກເປຣ

ໂດຍຂ້ອຄວາມແສດງໄດ້ກັ້ງຕັ້ງເລບແລະຕັ້ງອັກເປຣ

setTextColor(color)

setTextColor(color,background) ແສດງສືບອົງຕັ້ງອັກເປຣ ແລະ ແສດງພື້ນໜັງ

color ສືບອົງຕັ້ງອັກເປຣເປັນໄດ້ແຄ່ ວHITE ມີເຊື່ອ BLACK

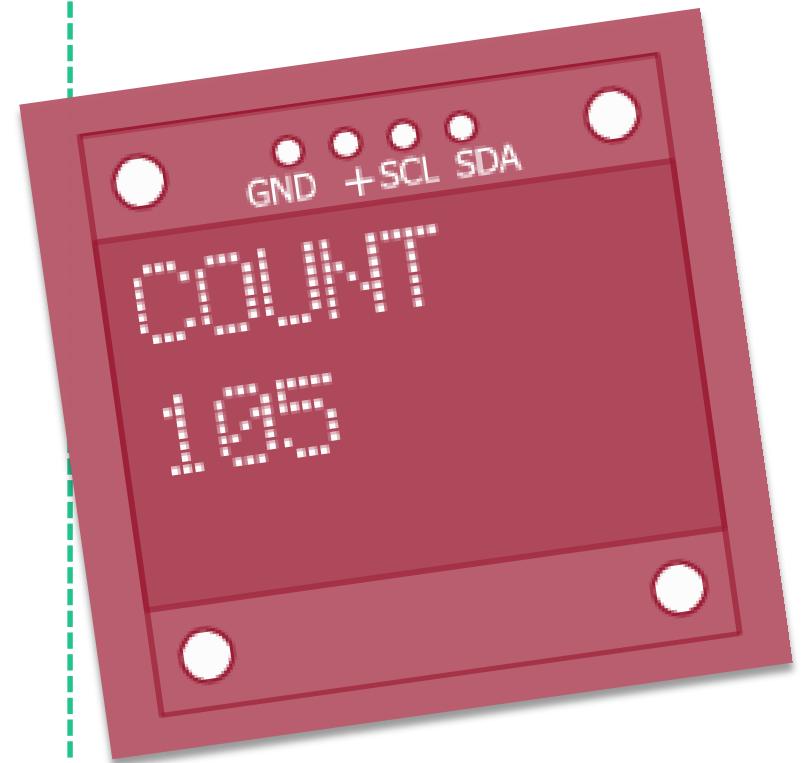
background ສືບອົງພື້ນໜັງຕັ້ງອັກເປຣເປັນໄດ້ແຄ່ WHITE ມີເຊື່ອ BLACK

ຕົວຢ່າງກາຮແສດງພລນ້າຈວ

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 OLED(-1);
void setup() {
    OLED.begin(SSD1306_SWITCHCAPVCC, 0x3C);
//initialize
}
void loop() {
    OLED.clearDisplay();
    OLED.setTextColor(WHITE);
    OLED.setCursor(0, 0);
    OLED.setTextSize(1);
    OLED.println("Innovative");
    OLED.setCursor(0, 10);
    OLED.setTextSize(2);
    OLED.println("Experiment");
    OLED.display(); //
}
```

ตัวอย่างการแสดงผลตัวเลข

```
#include <Wire.h>
#include <SPI.h>
#include <Adafruit_GFX.h>
#include <Adafruit_SSD1306.h>
Adafruit_SSD1306 OLED(-1);
int x=0;
void setup() {
    OLED.begin(SSD1306_SWITCHCAPVCC, 0x3C); //initialize
}
void loop() {
    OLED.clearDisplay();
    OLED.setTextColor(WHITE);
    OLED.setCursor(0, 0);
    OLED.setTextSize(3);
    OLED.println("COUNT");
    OLED.setCursor(0, 27);
    OLED.println(x,DEC);
    OLED.display(); //
    delay(300);
    x++;
}
```



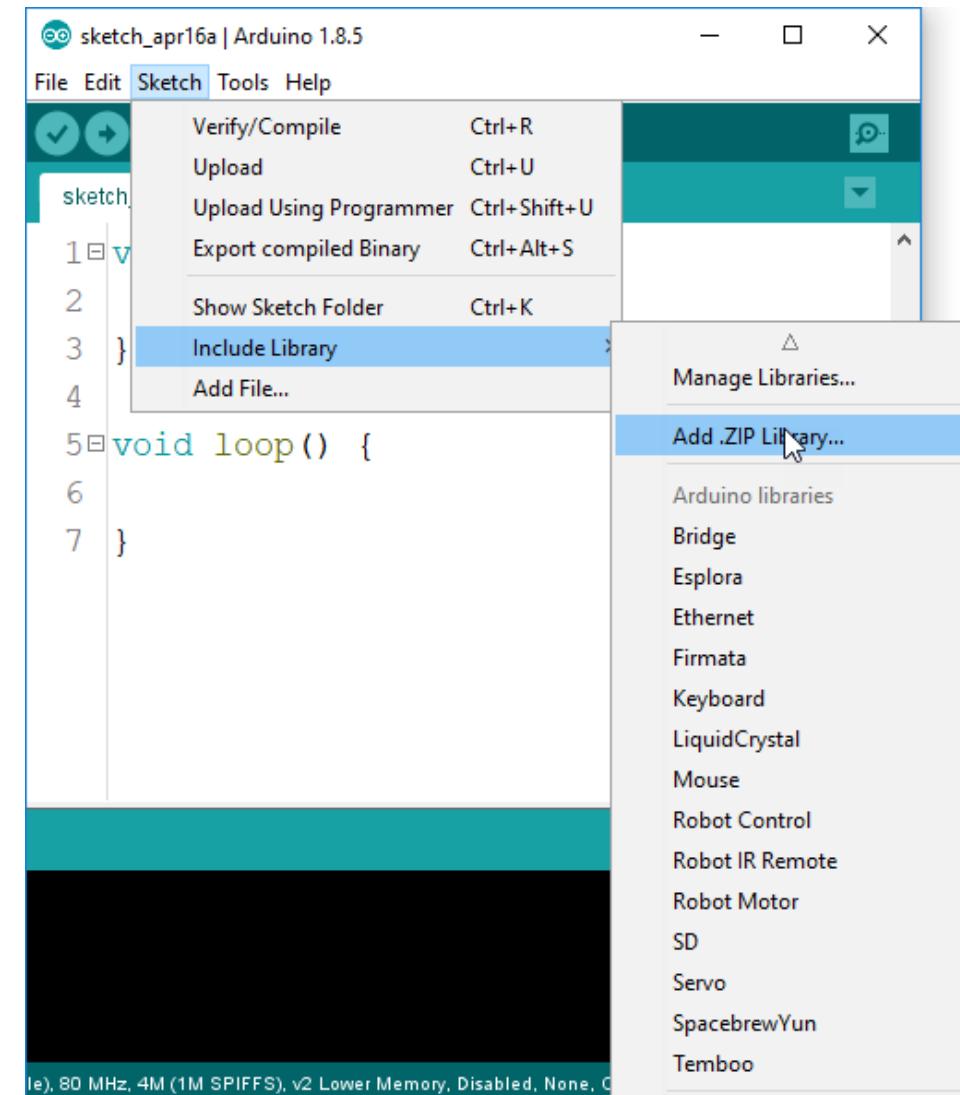
การติดตั้งไลบรารี AX-WIFI

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

เพื่อดาวน์โหลดไลบรารี

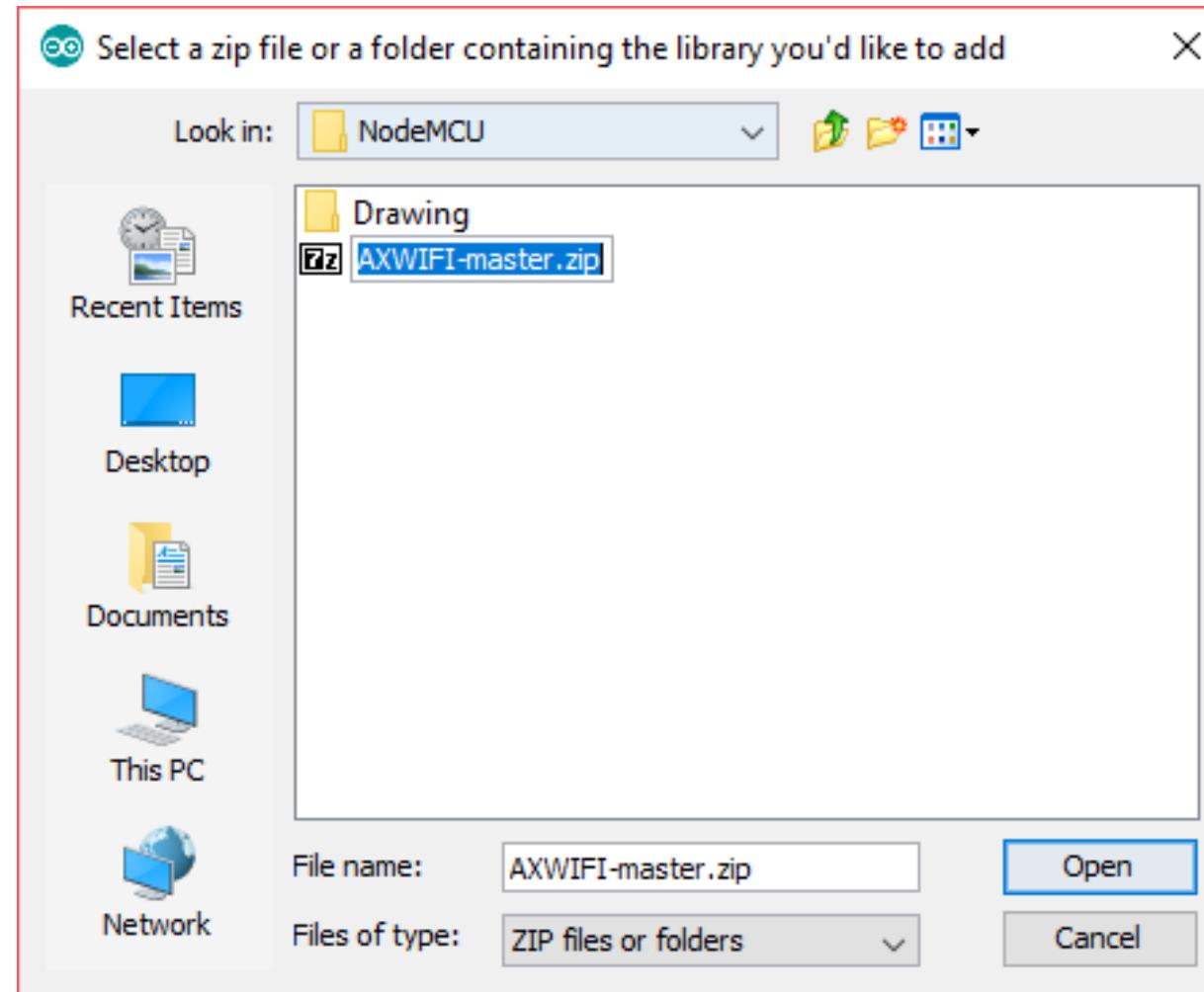


2. เพิ่มไลบรารีเข้าไปยัง Arduino



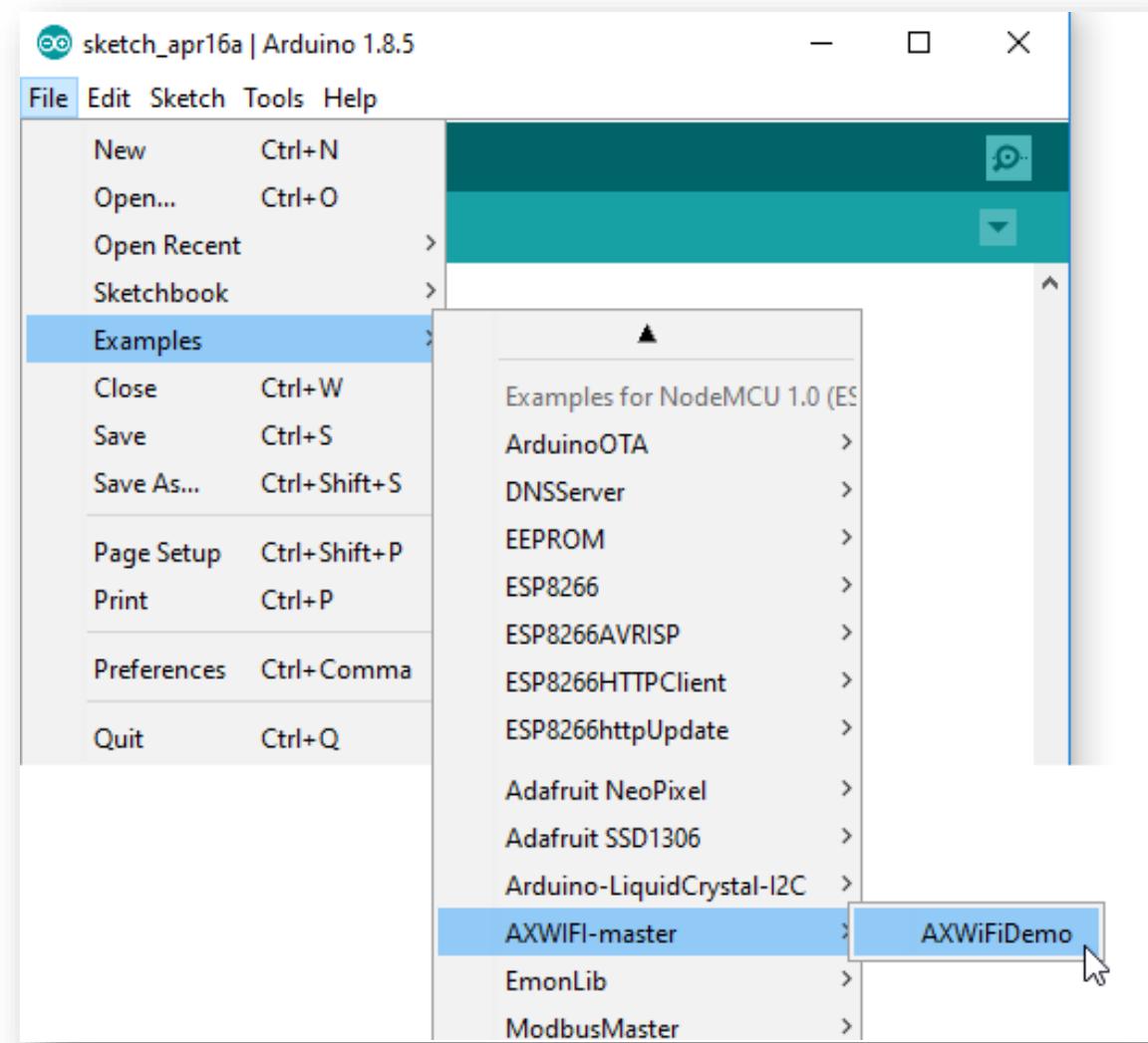
การติดตั้งไลบรารี AX-WIFI

3. เลือกซื้อไลบรารีที่ดาวน์โหลดไว้ก่อนหน้าบันทึก Open



ทดสอบโปรแกรมลง AX-WiFi

ไปที่เดบเมนู File > Examples > AX-WIFI > AXWIFIDemo



ไลบรารี AXWIFI.h

#include <AXWIFI.h>	publik ไลบรารี AXWIFI เข้ากับโปรแกรม
ax.begin();	ฟังก์ชันเริ่มต้นการใช้งานไลบรารี AX-WiFi
ax.beep();	สร้างเสียงออกลำโพงเปีย佐
ax.SledTypeGRB();	ฟังก์ชันกำหนดรูปแบบสีเรืองจาก สีเขียว, สีแดง, สีน้ำเงิน
ax.SledTypeRGB();	ฟังก์ชันกำหนดรูปแบบสีเรืองจาก สีแดง, สีเขียว, สีน้ำเงิน
ax.SledShow(Num, R, G, B)	ฟังก์ชันกำหนดและแสดงสีออก SLED Num - ตำแหน่งของ SLED R , G , B - ค่าสี แดง, เขียว, น้ำเงิน ระหว่าง 0-255

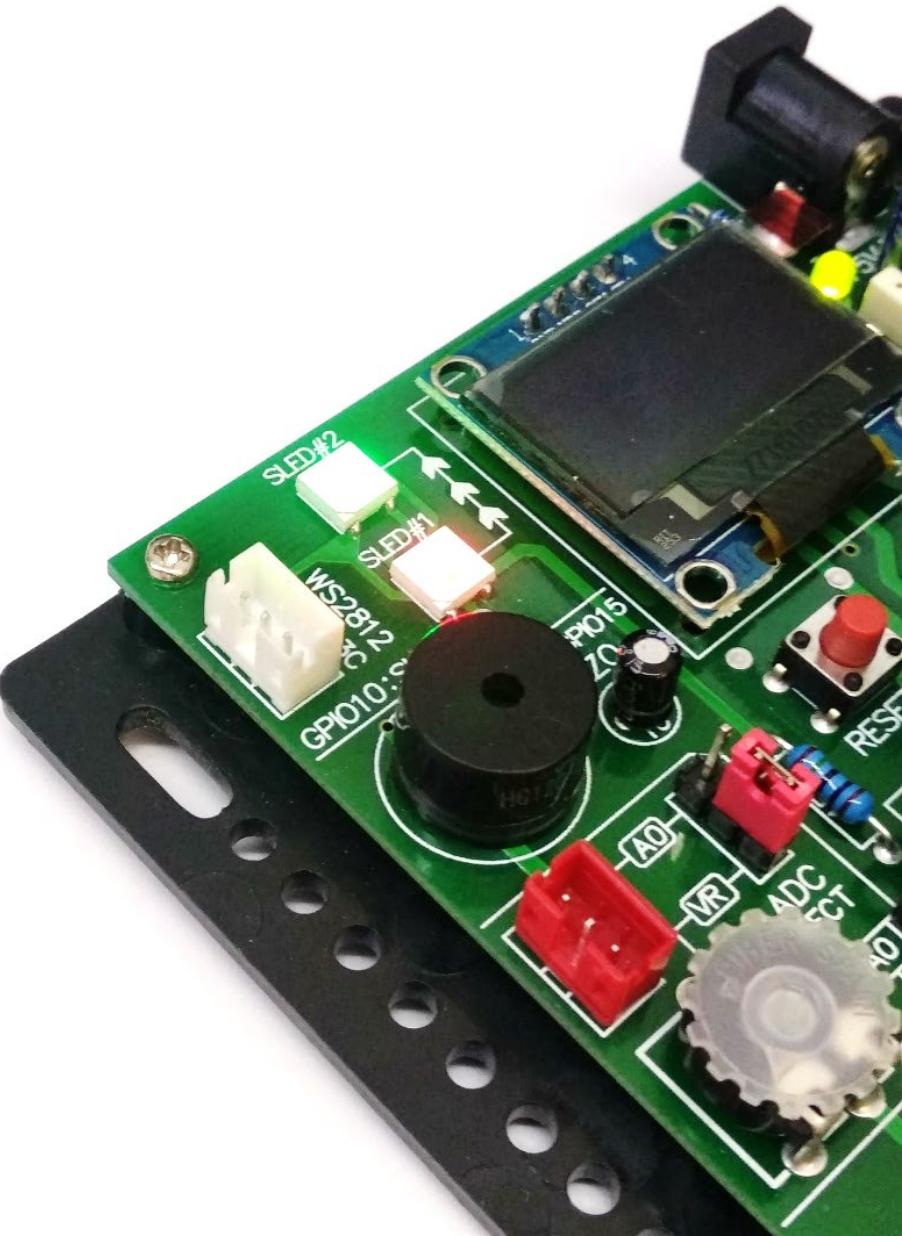
ทดลองสี LED

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

ตัวอย่าง

LED ดวงที่ 1 แสดงสีแดง 12 (5 %)

LED ดวงที่ 2 แสดงสีเขียว 12 (5 %)



ໄລຍະໄຣ AXWIFI.h ເກີຍວກັບ OLED

ax.OledSetText(X, Y, char *p, ...)

X – ຄື່ອຕຳແໜນ່ງປຣກັດ

Y – ຄື່ອຕຳແໜນ່ງຕົວວັກເຊ

*p – ຄື່ອຂ້ອຄວາມກໍ່ຕ້ອງການແສດງ

ຄ່າພິເສດ

%d ແສດງເລກຈຳນວນເຕີມ

%f ແສດງເລກຈຳນວນຈົງ (ເລກທຄບີຍມ)

ຕົວຢ່າງ

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

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

ໄລບຣາີ່ AXWIFI.h ເກື່ອງກັບ OLED

ax.OledSetTextSize(Size) ພັງກົບກຳນົດຂາດຕັວອັກປຣ
size - ຂາດຕັວອັກປຣເປັນຈຳນວນເກົ່າ ດ້ວຍເຫັນຕົວ 1 ເກົ່າ 6x8 ຈຸດ
ຕັວຢ່າງ **ax.OledSetTextSize(1) ;**

ເນື້ອກຳນົດຂ້ອຄວາມເສຮື່ອແລ້ວຕ້ອງໃຊ້ຄຳສັ່ງ

ax.OledShow() ;

ເພື່ອແສດງຜລ

ทดลองโปรแกรมแสดงผลอ่าน OLED

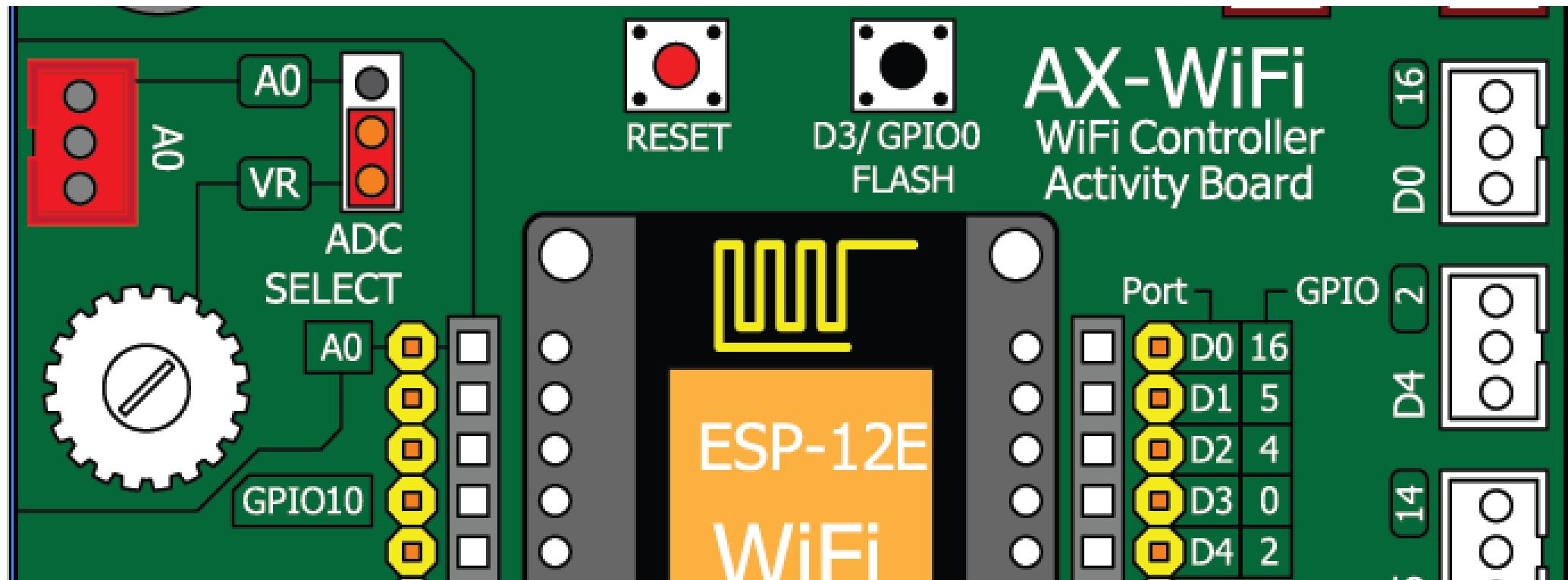
```
#include "AXWIFI.h"
void setup() {
    ax.begin();
    ax.OledSetTextSize(2);
    ax.OledSetText(2,2,"AX-WiFi");
    ax.OledShow();
}
void loop() {
}
```

การอ่านค่าอะนาลอก

x=analogRead (pin) ;

x = ตัวแปร int เพื่อบันทึกค่า

pin = ขาอินพุตอะนาลอก A0 เก่าบีบ

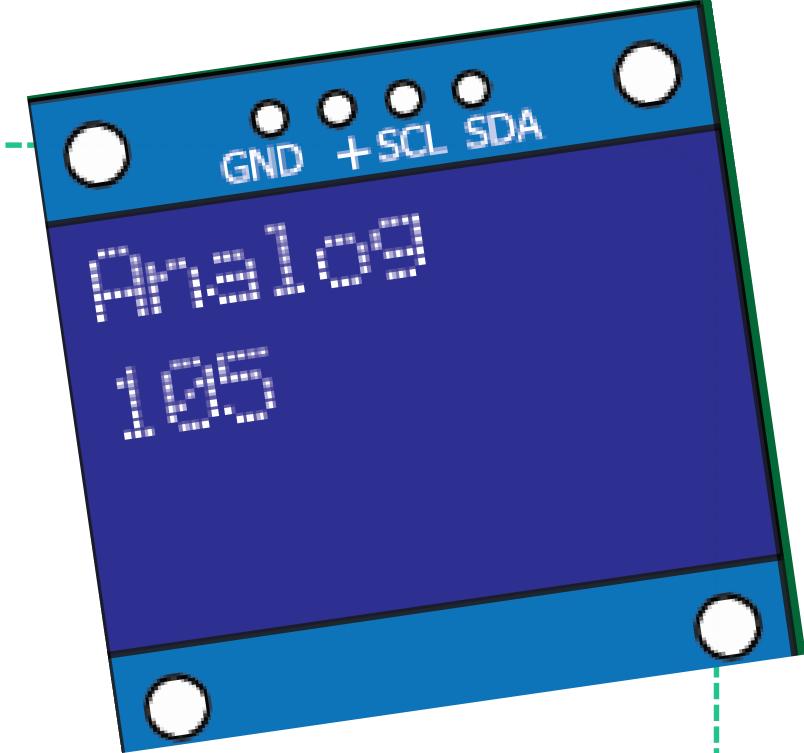


ตัวอย่างอ่านค่า analog และแสดงที่ OLED

```
#include "AXWIFI.h"

void setup() {
    ax.begin();
    ax.OledSetTextSize(2);
    ax.OledSetText(0,0,"Analog");
    ax.OledShow();
}

void loop() {
    ax.OledSetText(1,0,"%d ",analogRead(A0));
    ax.OledShow();
}
```



การใช้งานฟังก์ชัน map

map(value, fromLow, fromHigh, toLow, toHigh)

ตัวอย่าง

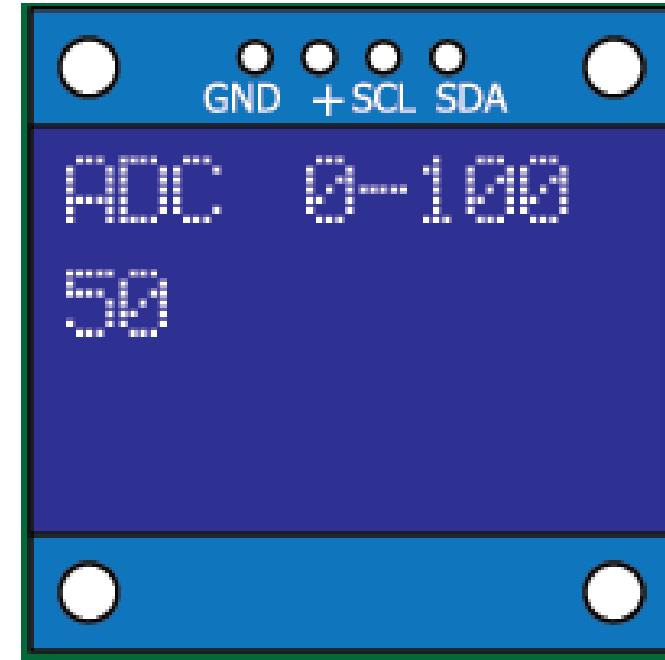
```
y = map(x, 1, 50, 50, 1);
```

ตัวอย่าง

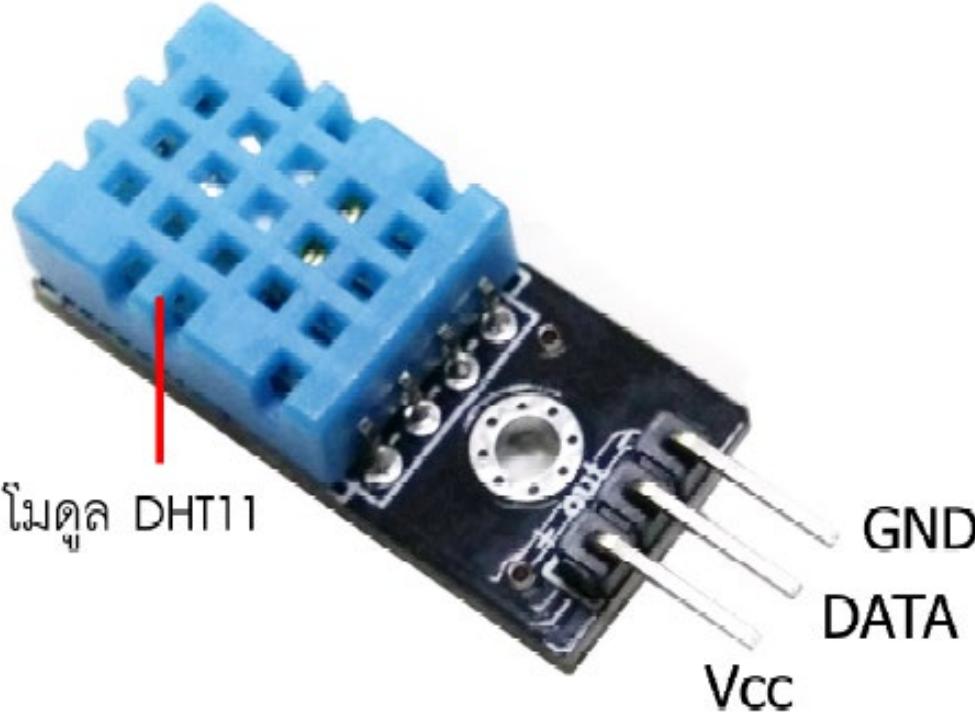
```
y = map(x, 1, 50, 50, -100);
```

ตัวอย่างอ่านค่าพร้อมใช้คำสั่ง map

```
#include "AXWIFI.h"
int x;
void setup() {
    ax.begin();
    ax.OledSetTextSize(2);
    ax.OledSetText(0,0,"ADC 0-100");
    ax.OledShow();
}
void loop() {
    x=map(analogRead(A0),0,1024,0,100);
    ax.OledSetText(1,0,"%d ",x);
    ax.OledShow();
}
```



ZX-DHT11 วัดความชื้นและอุณหภูมิ



วัดความชื้นสัมพักร์

20 ถึง 80%RH

ความผิดพลาด 5%RH

ความละเอียด 1 %

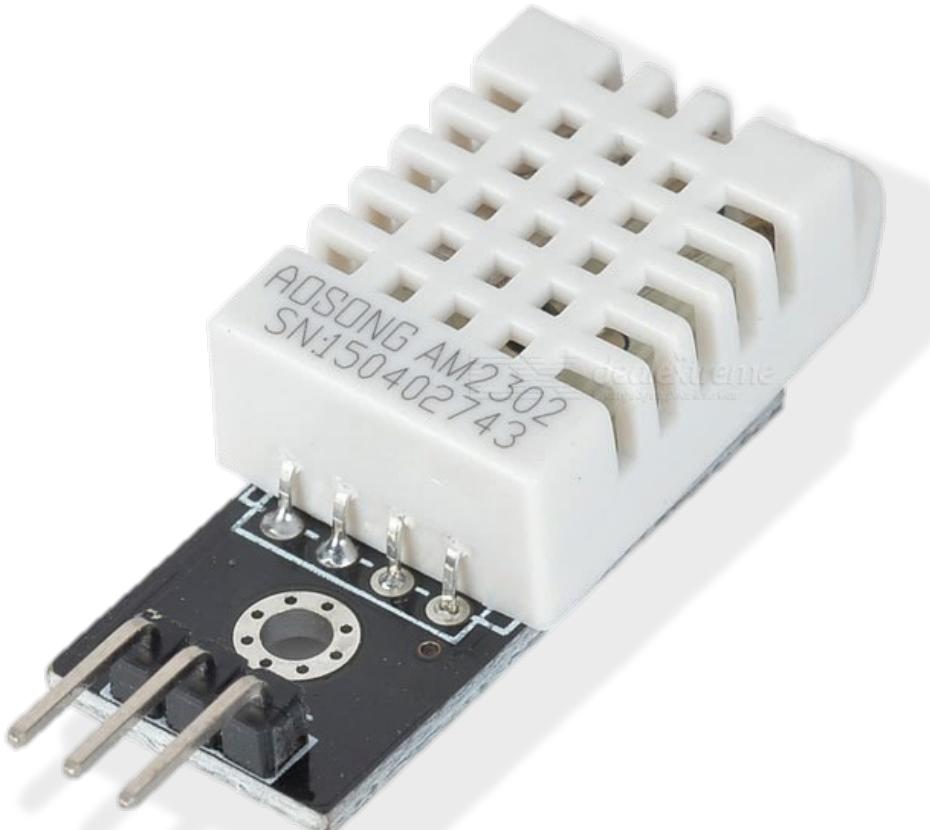
วัดอุณหภูมิ

0 ถึง 50 องศาเซลเซียส

ความผิดพลาด 2 องศาเซลเซียส

ความละเอียด 1 องศาเซลเซียส

ZX-DHT22 วัดความชื้นและอุณหภูมิ



วัดความชื้นสัมพักร์

0 ถึง 100%RH

ความผิดพลาด 2-5%RH

ความละเอียด 0.1 %

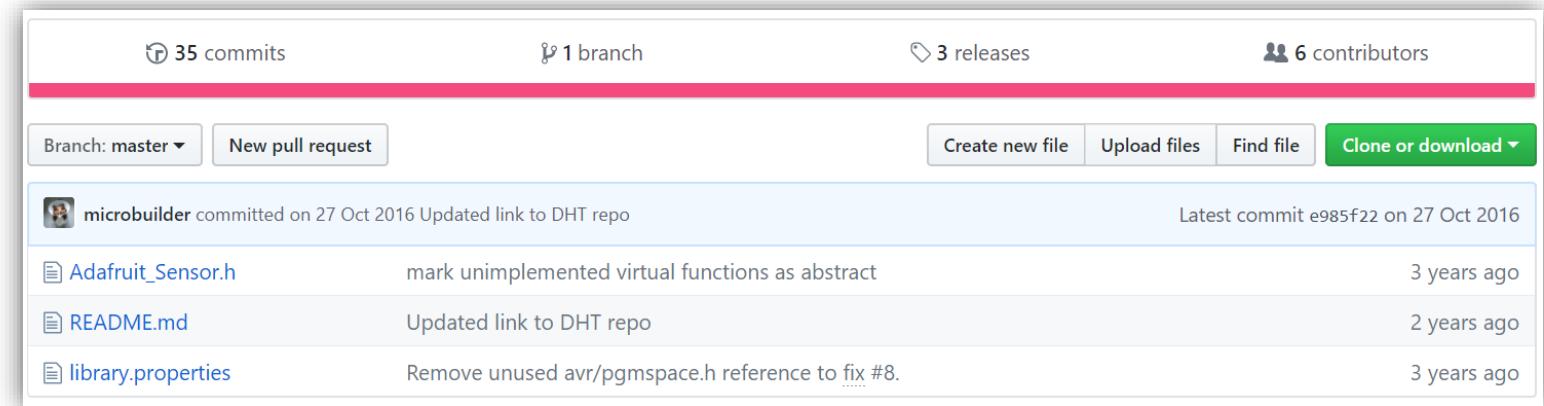
วัดอุณหภูมิ

-40 ถึง 80 องศาเซลเซียส

ความผิดพลาด +- 0.5 องศาเซลเซียส

ความละเอียด 0.1 องศาเซลเซียส

ดาวน์โหลดไลบรารี DHT22



Branch: master ▾ New pull request Create new file Upload files Find file Clone or download ▾

microbuilder committed on 27 Oct 2016 Updated link to DHT repo Latest commit e985f22 on 27 Oct 2016

Adafruit_Sensor.h mark unimplemented virtual functions as abstract 3 years ago

README.md Updated link to DHT repo 2 years ago

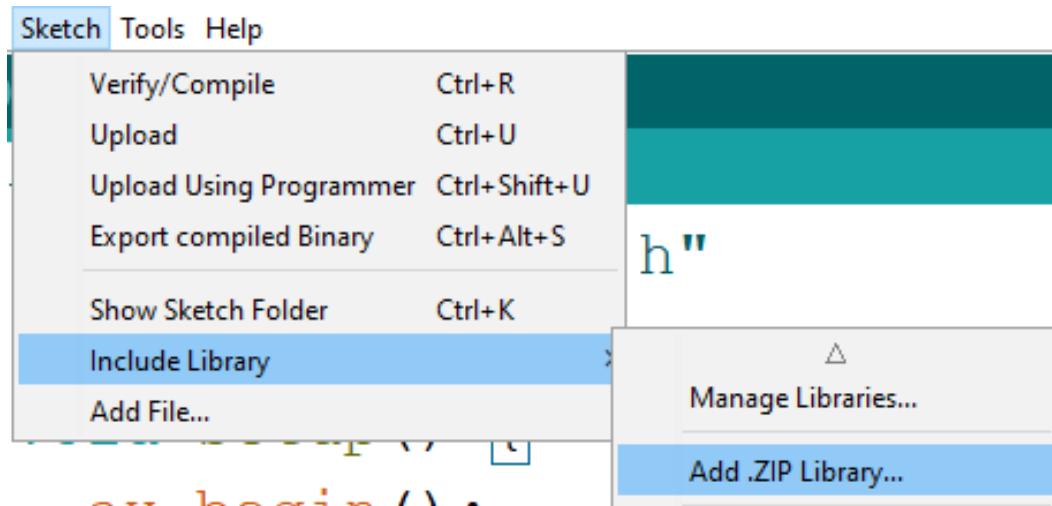
library.properties Remove unused avr/pgmspace.h reference to fix #8. 3 years ago

สามารถดาวน์โหลดได้ที่

https://github.com/adafruit/Adafruit_Sensor

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

แตกไฟล์ไปยังโฟลเดอร์ libraries ที่ได้ติดตั้งไว้ในเครื่อง หรือติดตั้งผ่าน



Sketch>Include Library>Add.ZIP Library

การใช้งานผ่านไลบรารี DHT.h

เพิ่ม Library ในโปรแกรมด้วยคำสั่ง `#include "DHT.h"`

เรียกใช้ DHT โดยระบุ `DHT dht(D5, DHT22, 15);`

D5 ขาที่เชื่อมต่อ DHT22 รุ่นที่เลือกใช้

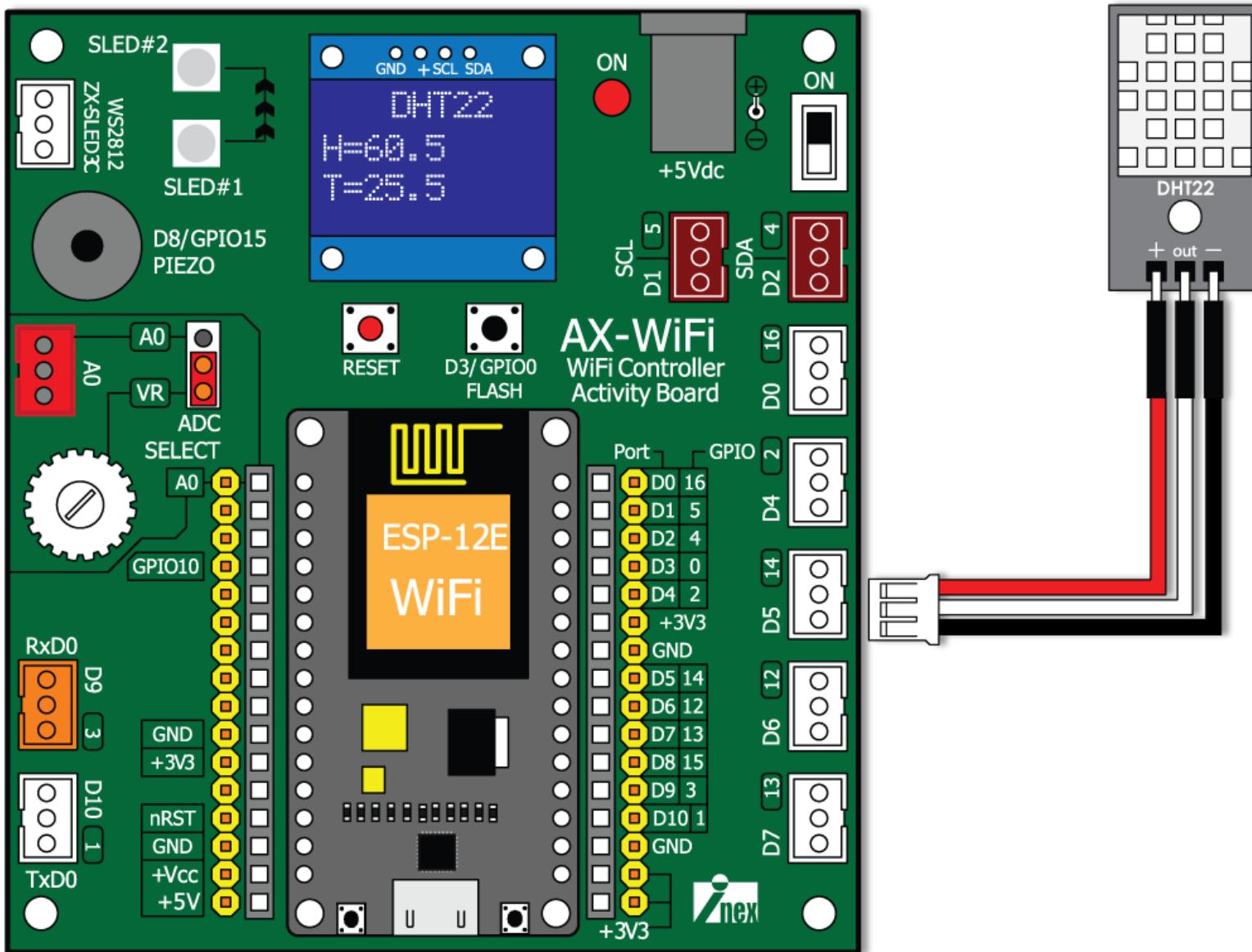
เริ่มต้นใช้งานด้วยคำสั่ง `dht.begin()`

ตรวจสอบอ่านค่าด้วย `dht.read()` True อ่านค่าได้ False อ่านค่าไม่ได้

อ่านค่าความชื้นสัมพัทธ์ `dht.readHumidity()`

อ่านค่าอุณหภูมิองศาเซลเซียส `dht.readTemperature()`

การเชื่อมต่อ DHT-11 กับ AX-WiFi

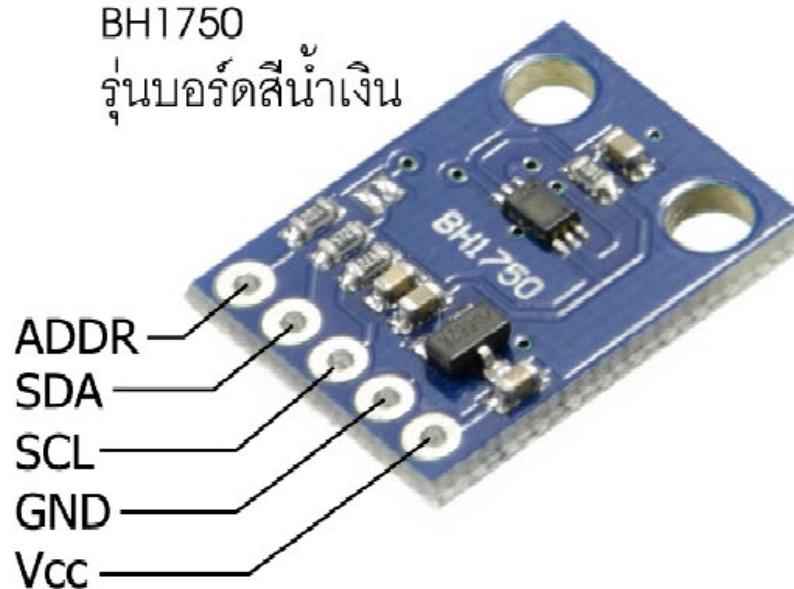


```
#include "AXWIFI.h"
#include "DHT.h"
DHT dht(D5,DHT22,15);
void setup(){
    dht.begin();
    ax.begin();
    ax.OledSetTextSize(2);ax.OledSetText(0,2,"DHT22");
}
void loop(){
    delay(1000);
    if (dht.read()) {
        float h = dht.readHumidity();
        float t = dht.readTemperature();
        ax.OledSetText(1,0,"H=%1f",h);
        ax.OledSetText(2,0,"T=%1f",t);
        ax.OledShow();
    }
    else{
        ax.clear();
        ax.OledSetText(0,0,"Error Reading");
        ax.OledShow(); while(1);
    }
}
```

BH1750 บอร์ดวัดความเข้มแสงติดต่อผ่านบัส I²C



BH1750
รุ่นบอร์ดสีดำ



BH1750
รุ่นบอร์ดสีน้ำเงิน

ย่างบัดความเข้มแสง 1 ถึง 65,535 ลักซ์ มีค่าความผิดพลาด 20%

กบต่อการรบกวนจากแสงอิบพราเด

ต่อพร้อมกันได้ 2 ตัวกำหนดผ่านขา ADDR

ตำแหน่งแอดเดรสของ I²C อยู่ที่ 0x23

BH1750 บอร์ดวัดความเข้มแสงติดต่อผ่านบัส I²C

ข้อมูลค่าความเข้มแสงในสภาพแวดล้อม

กลางคืน

มีค่าความเข้มแสง 0.001 ถึง 0.02 ลักซ์

คืนพระจันทร์เต็มดวง

มีค่าความเข้มแสง 0.02 ถึง 0.3 ลักซ์

พื้นที่ในร่มยามมีเมฆ

มีค่าความเข้มแสง 5 ถึง 50 ลักซ์

พื้นที่กลางแจ้งยามมีเมฆ

มีความเข้มแสง 50 ถึง 500 ลักซ์

ในร่มที่มีไฟสว่างมาก

มีความเข้มแสง 100 ถึง 1,000 ลักซ์

แสงไฟอ่านหนังสือ

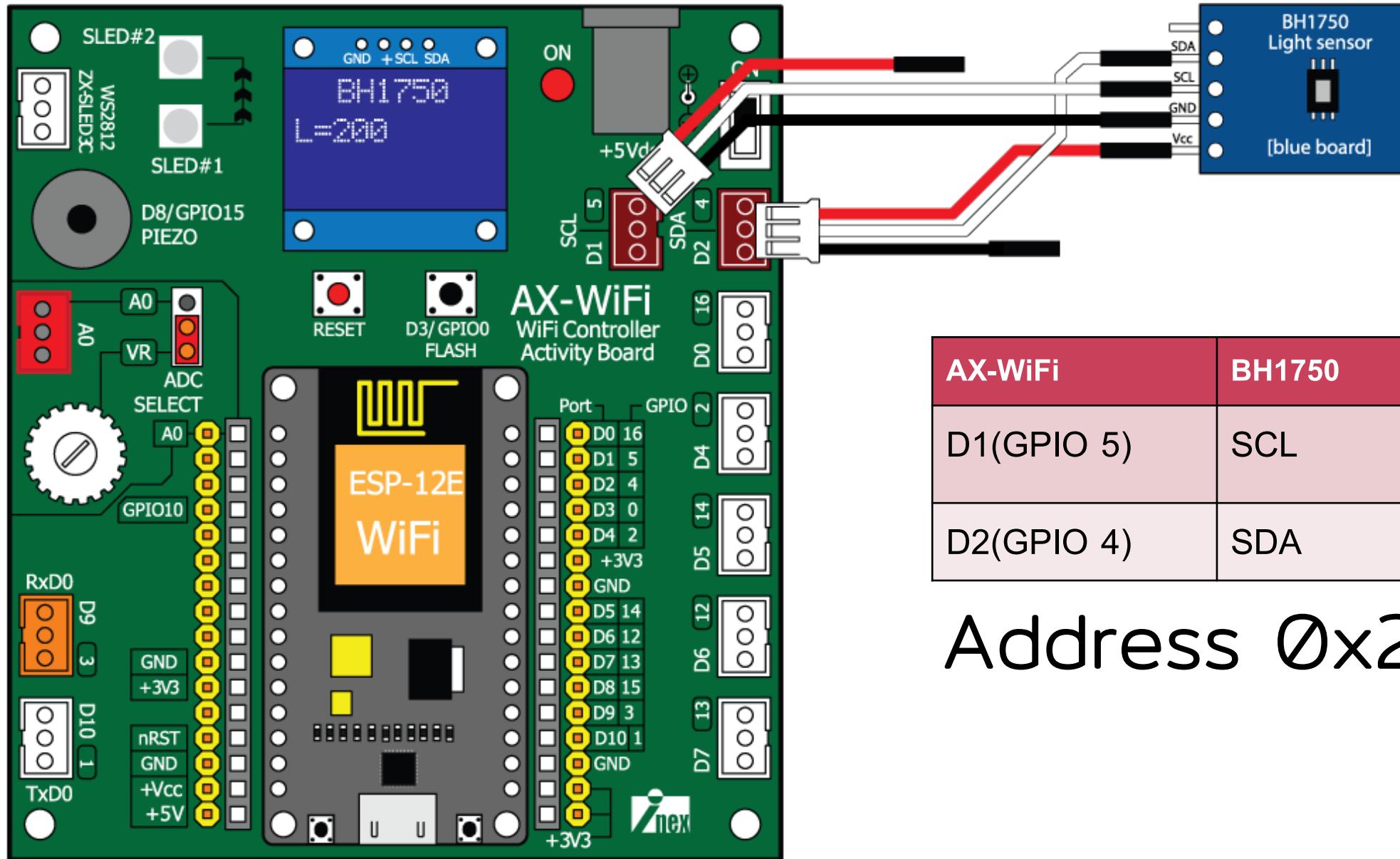
มีความเข้มแสง 50 ถึง 60 ลักซ์

หน้าจอภาพของไฮมวิเดโອินบ้าน

มีความเข้มแสง 1400 ลักซ์

ข้อมูลนี้เป็นค่าตัวเลขโดยประมาณนำมาใช้ประกอบการอธิบายและให้เห็นในภาพรวมเท่านั้น

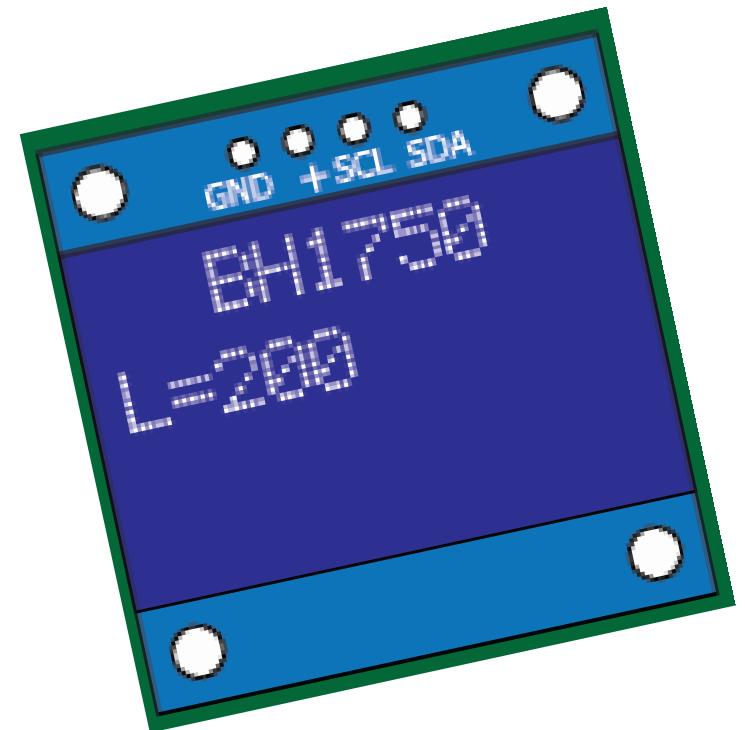
เชื่อมต่อ BH1750 กับ AX-WiFi



Address $\textcircled{0}x23$

การอ่านค่าจาก BH1750 และแสดงที่ OLED

```
#include <Wire.h>
#include <AXWIFI.h>
byte buff[2];
int L;
void setup() {
    Wire.begin();
    ax.begin();
    ax.OledSetTextSize(2);
    ax.OledSetText(0,1,"BH1750");
}
void loop() {
    L=BH1750_Read(0x23); ← เรียกใช้ฟังก์ชัน
    ax.OledSetText(1,0,"L=%d ",L);
    ax.OledShow();
}
```



ฟังก์ชันอ่านค่าจากโมดูล BH1750

```
int BH1750_Read(int address){  
    int i=0, val;  
    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 (i==2) {  
        val = ((buff[0] << 8) | buff[1]);  
        return val;  
    }else{return -1;}  
}
```

รูปแบบการอ่านค่า

`attachInterrupt(pin, ISR, mode)`

`pin` ตำแหน่งขาที่รับอินเตอร์รัปต์

`ISR` ตำแหน่งฟังก์ชันที่เรียกบริการ Interrupt

`mode` รูปแบบการกระตุ้น Interrupt

`LOW` กระตุ้นเมื่อขาเป็น 0

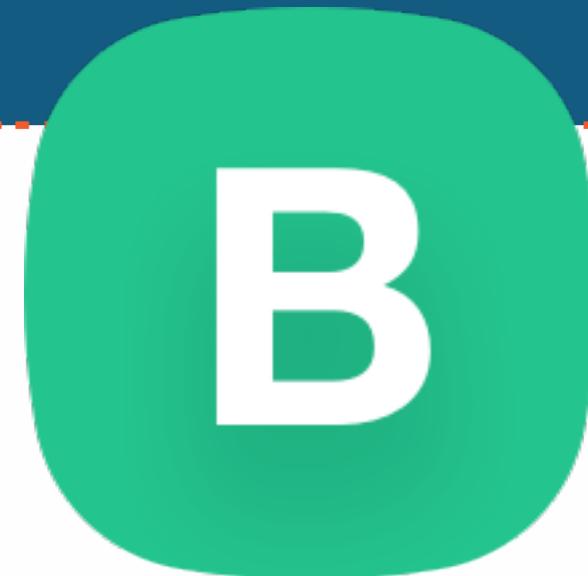
`CHANGE` กระตุ้นเมื่อมีการเปลี่ยนค่า

`RISING` กระตุ้นที่ขอบขาขึ้น

`FALLING` กระตุ้นที่ขอบขาลง

ตัวอย่างการใช้งาน Interrupt

```
void setup() {  
    pinMode(D2, OUTPUT);  
    pinMode(D3, OUTPUT);  
    pinMode(D5, INPUT);  
    pinMode(D4, INPUT);  
    attachInterrupt(D5,pinD5_ISR,CHANGE);  
    attachInterrupt(D4,pinD4_ISR,CHANGE);  
}  
  
void loop() {  
    digitalWrite(D3,1);delay(100);  
    digitalWrite(D3,0);delay(100);  
}  
  
void pinD5_ISR() {  
    digitalWrite(D2, HIGH);  
}  
void pinD4_ISR() {  
    digitalWrite(D2, LOW);  
}
```



Blynk

ทำ Internet of Things เป็นเรื่องง่ายสำหรับทุกคน

<https://blynk.io/>

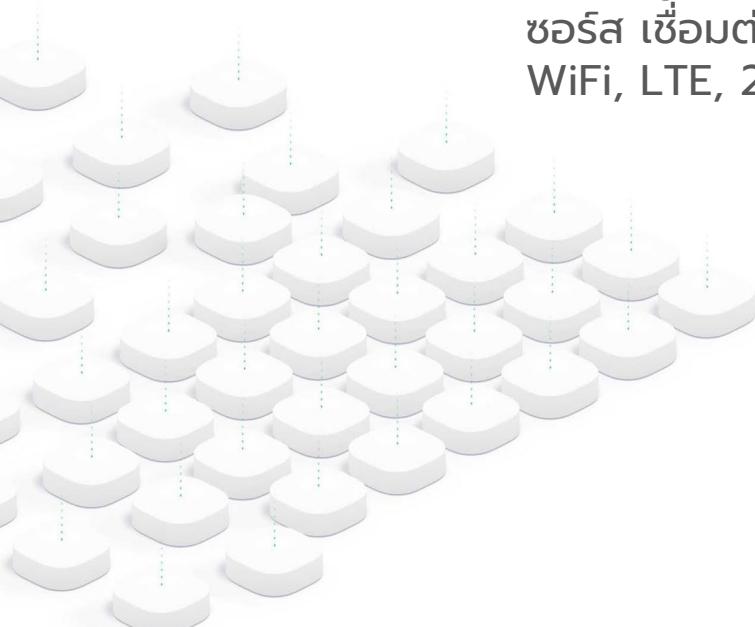
B

Blynk

แพลตฟอร์ม IoT ที่ได้รับความนิยมสูงสุด เพื่อเชื่อมต่ออุปกรณ์ของเรากับคลาวด์ และพัฒนาแบบเพื่อควบคุมและจัดการฮาร์ดแวร์ต่างๆ ได้อย่าง halfway



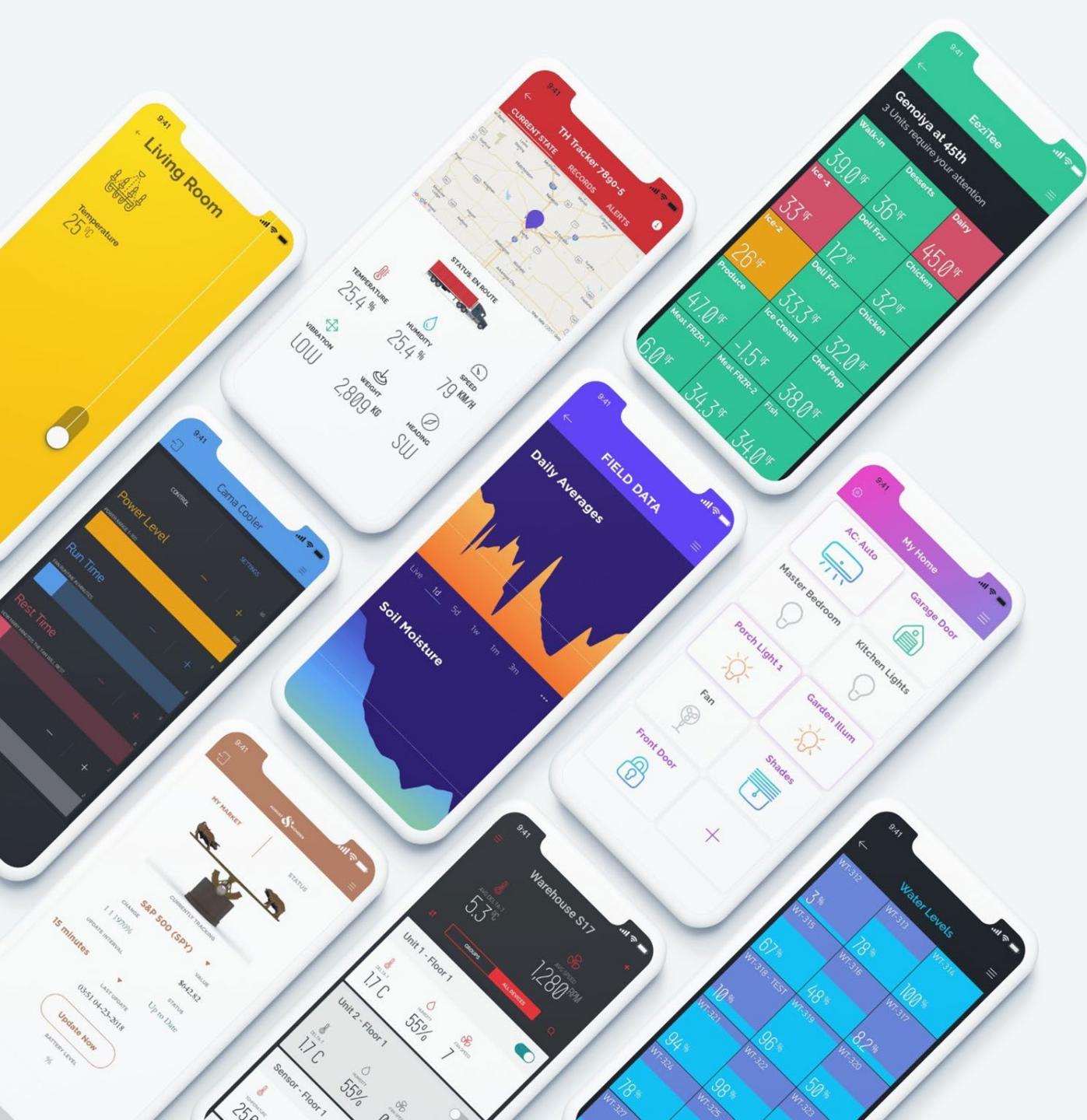
ฮาร์ดแวร์กว่า 400 แบบที่ใช้งานได้กันที่ด้วย Blynk Cloud ที่ร่วดเร็วและโถเพ่นชอร์ส เชื่อมต่อได้อย่างง่ายดายผ่าน WiFi, LTE, 2G-4G หรือ Ethernet



วิดเจ็ตที่ออกแบบไว้หลากหลายเพื่อสร้างแอป iOS และ Android ควบคุมอุปกรณ์อิเล็กทรอนิกส์ ตรวจสอบข้อมูลเซ็นเซอร์ รับการแจ้งเตือนและอีกมากมาย



ควบคุมฮาร์ดแวร์ของคุณจากทุกที่ในโลก!



สิ่งที่ต้องมีเมื่อ

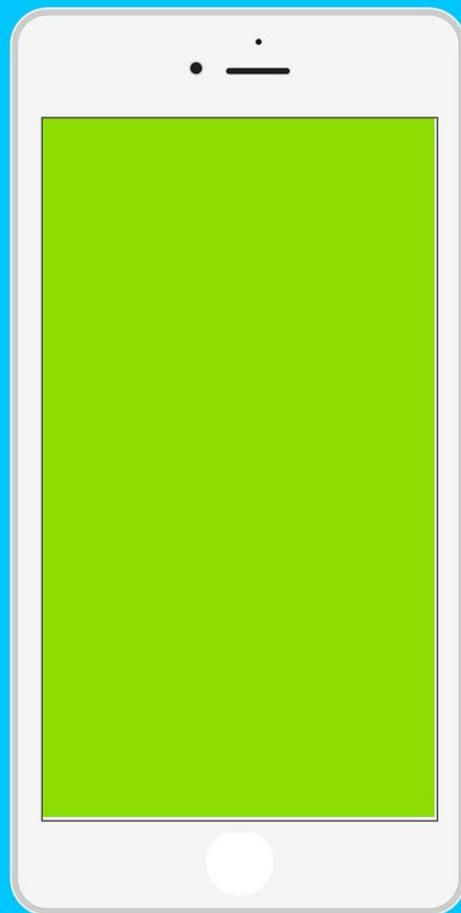
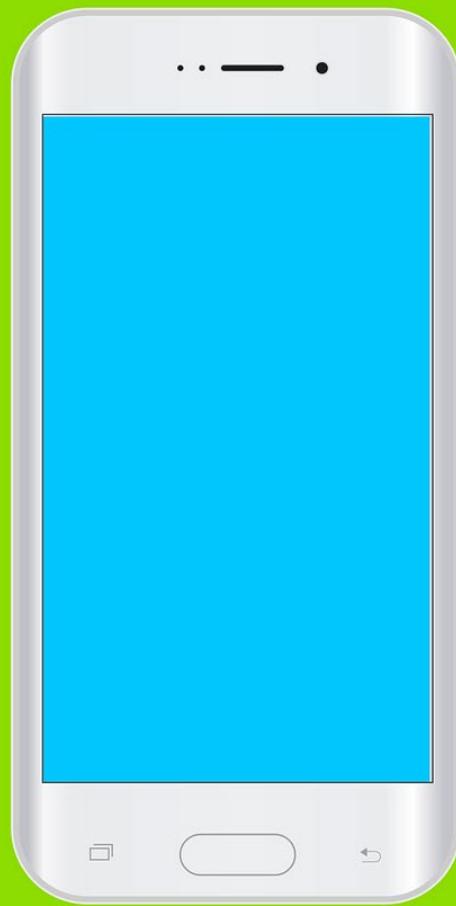


Blynk

1. มี Smartphone

2. มี Hardware IoT

3. มี Internet



1. มี Smartphone

Download on the
App Store

GET IT ON
Google Play

Android OS Version 4.2 ขึ้นไป

iOS Version 9 ขึ้นไป



ทำงานบนคอมพิวเตอร์ไม่ได้ ทำได้
ผ่านซอฟต์แวร์พวงกэмล้องการ
ทำงานเท่านั้น

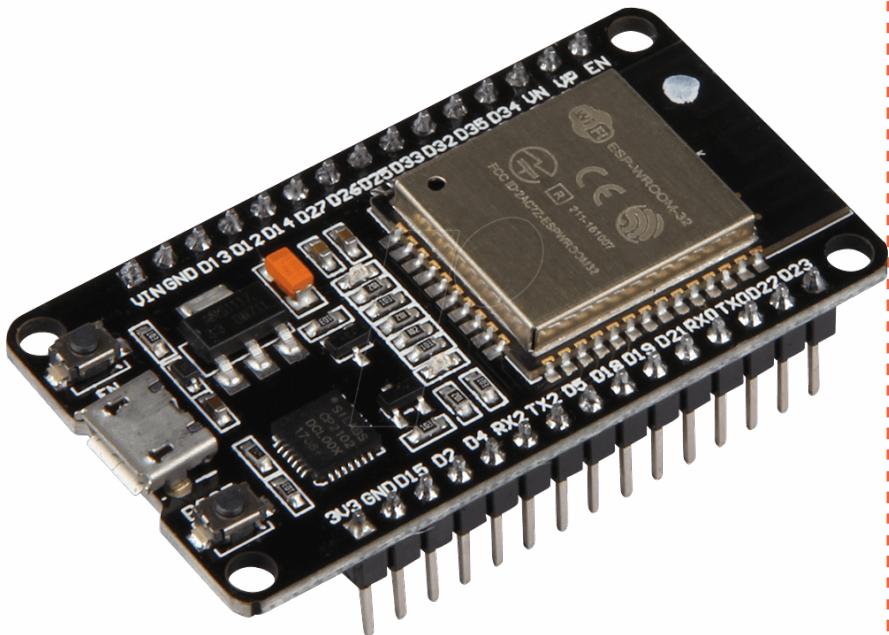
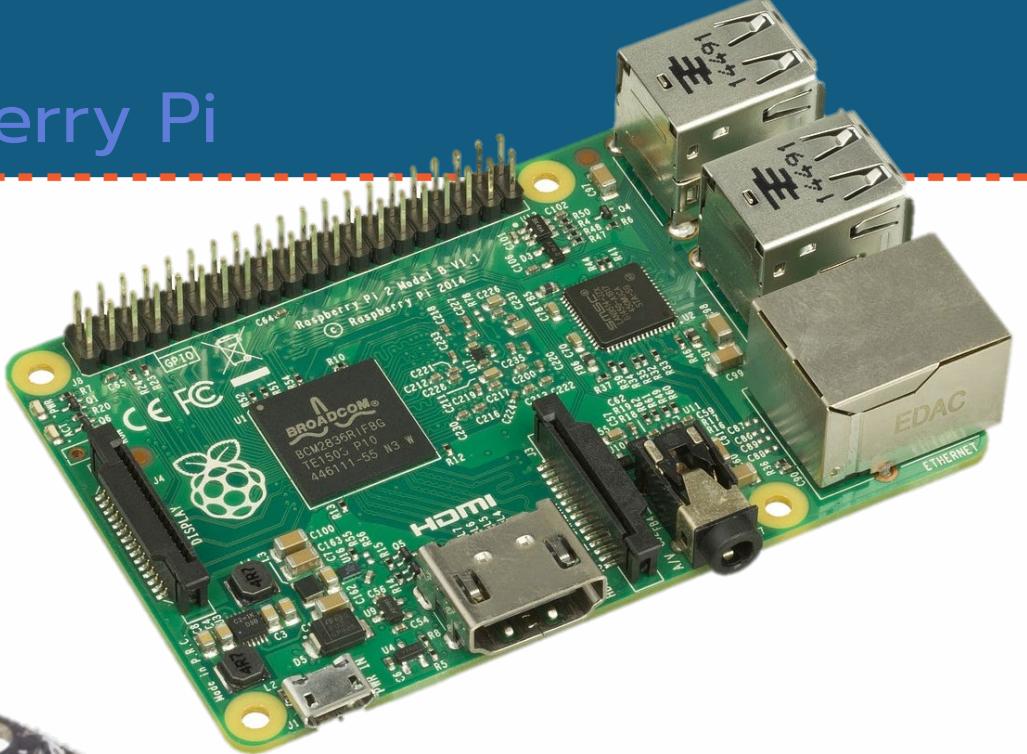
Raspberry Pi



2. І̄ Hardware IoT



ESP8266



ESP32

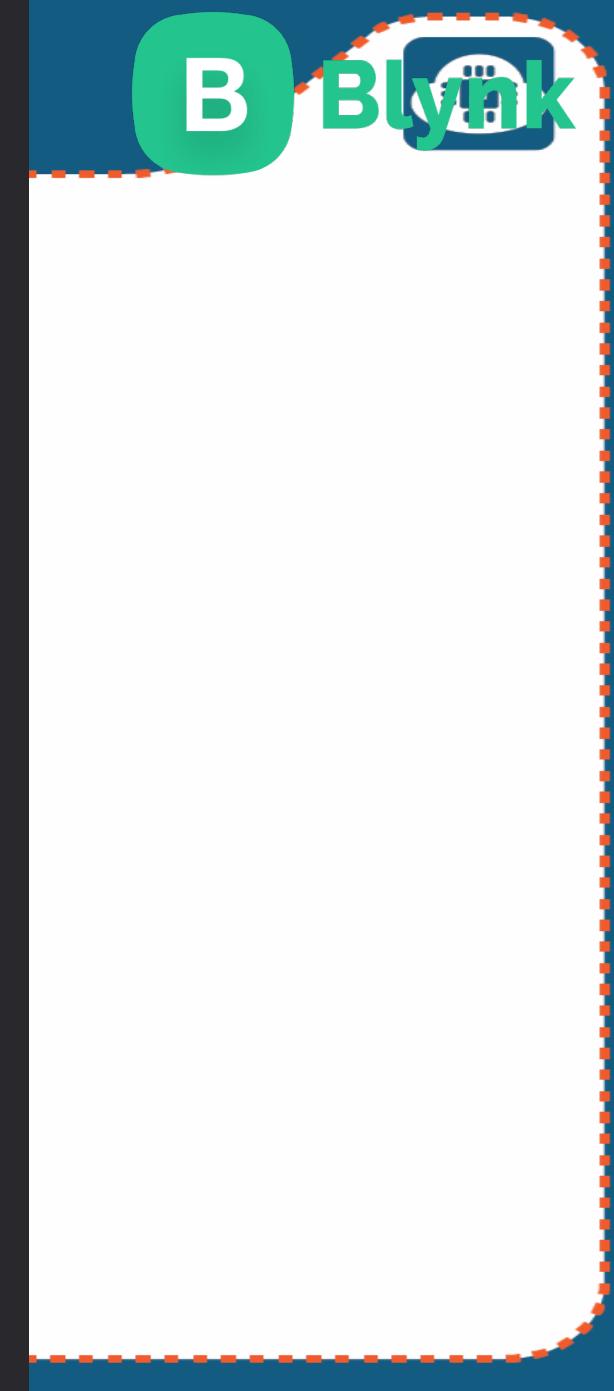
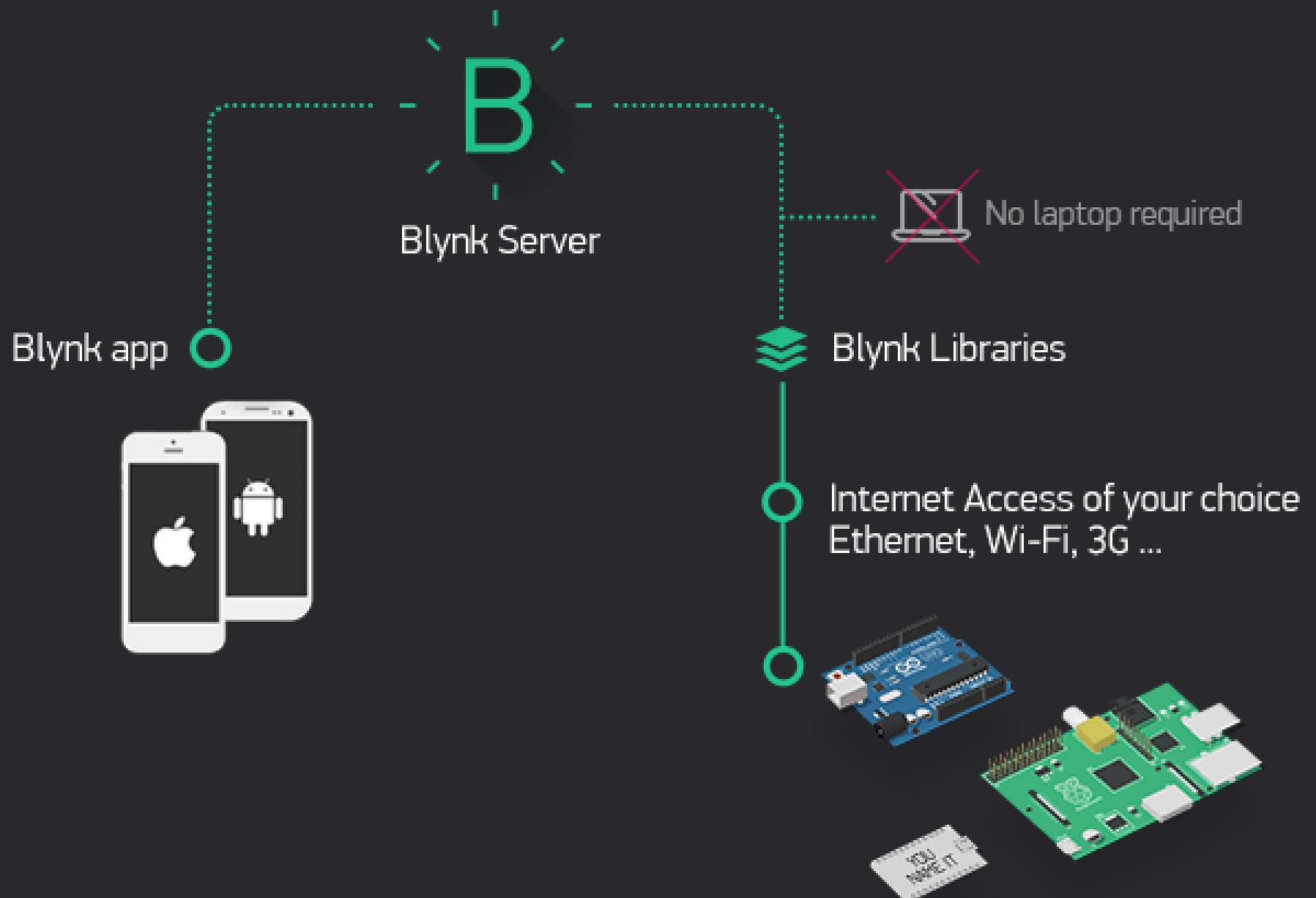
3. มี Internet



ฮาร์ดแวร์ที่ใช้งานต้องเชื่อมต่ออินเทอร์เน็ต

- WiFi
- Ethernet
- Cellular (GSM, 2g, 3g, 4g, LTE)
- Serial
- USB via your PC
- Bluetooth (BETA)





1

ចាប់តាម App Blynk : 1 នាត់



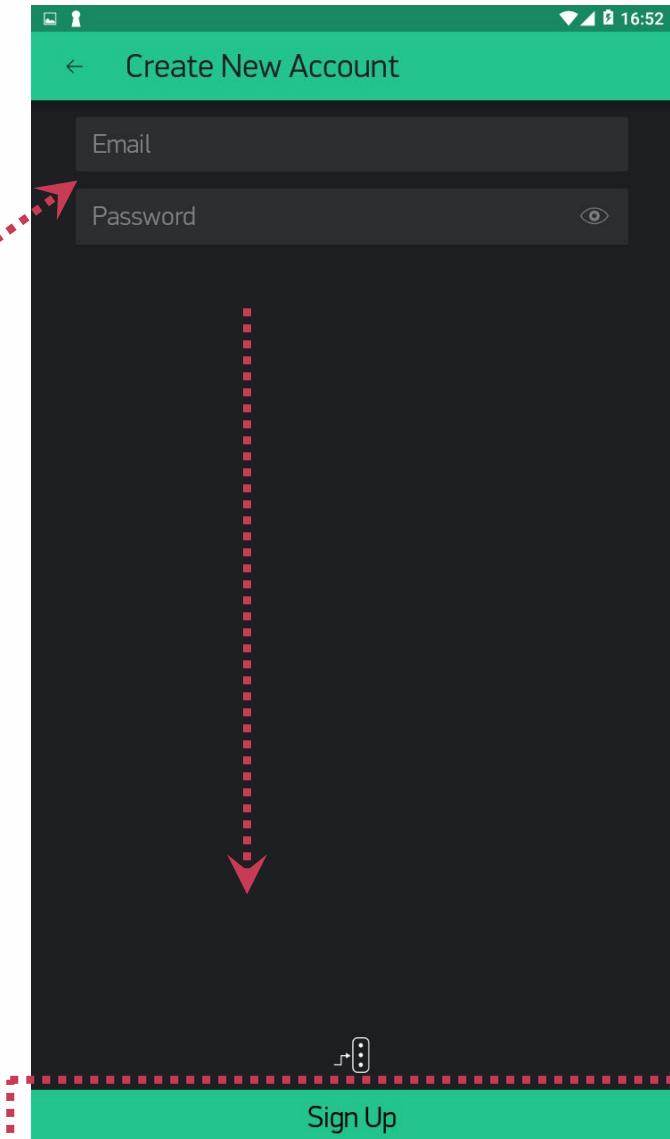
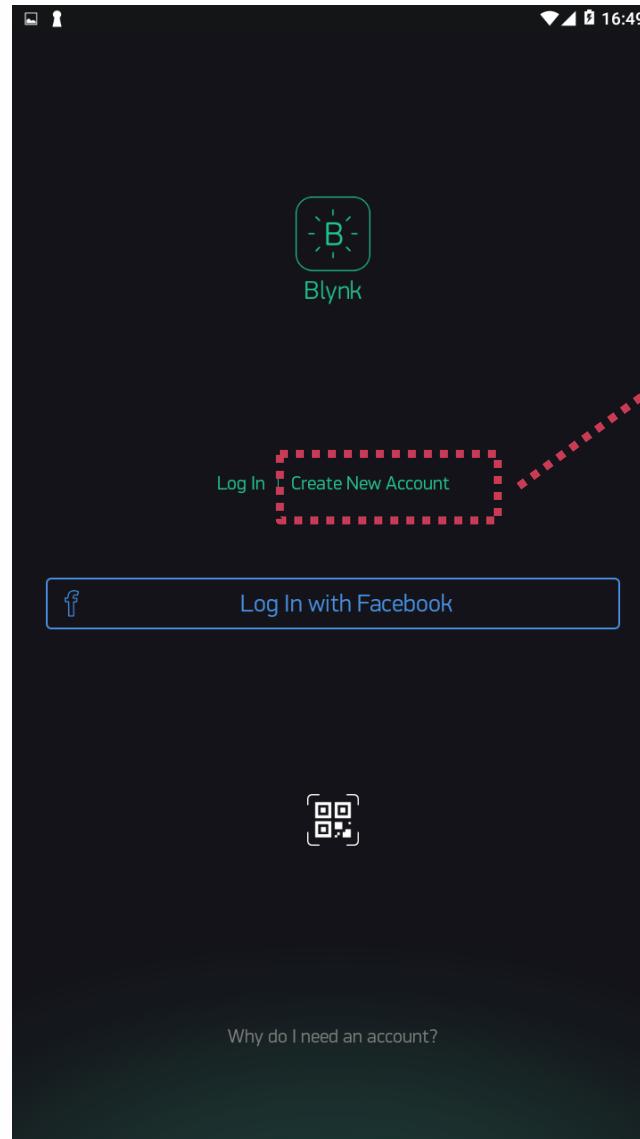
2

ลงทะเบียน

หรือ Login ผ่าน Facebook

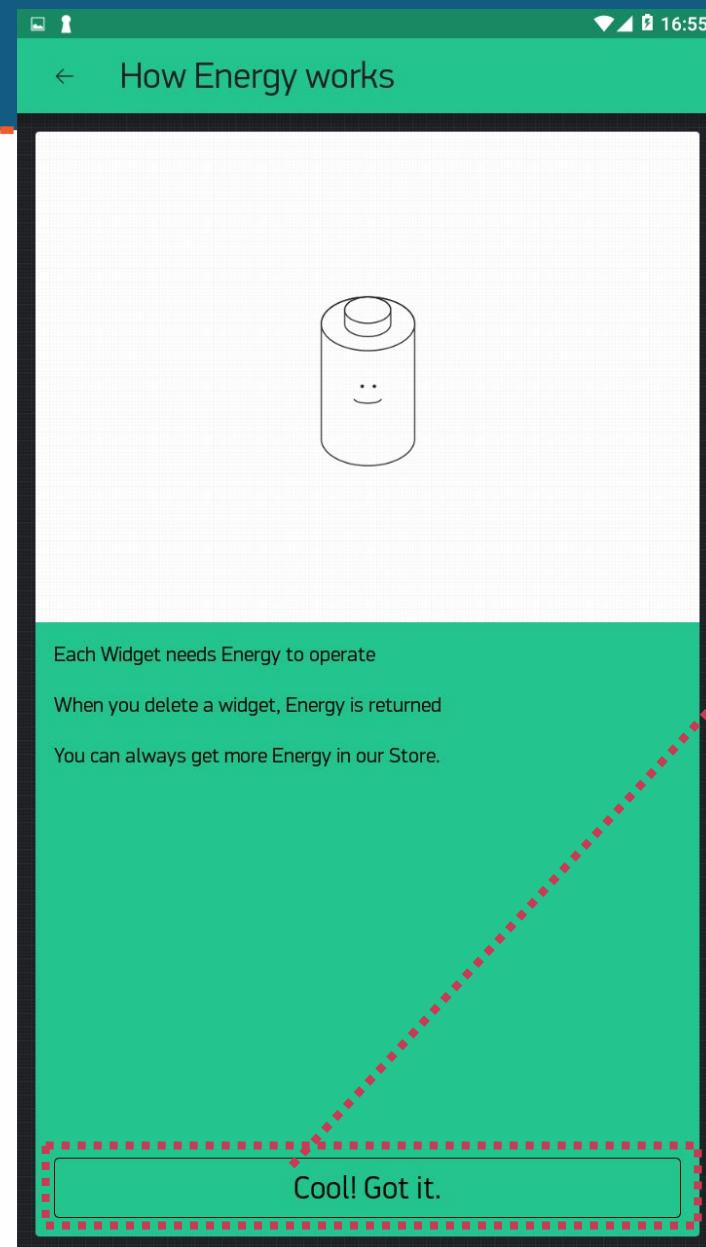
ใช้ชื่อคุณอีเมล

เพื่อรับคีย์ที่ส่งมา



3 สร้าง Project

Blynk จะแจ้งเรื่องพลังงานให้ทราบ
คือถ้าเราสร้างวิดเก็ต เราจะเสีย
พลังงานไป แต่ถ้าลบออก พลังงานจะ^{กลับมา} อย่างได้เพิ่ม ไปซื้อที่ **ร้านค้า**
ของ Blynk ได้เลย



B Blynk



ทำ Internet of Things เป็นเรื่องง่ายสำหรับทุกคน

<https://blynk.io/>

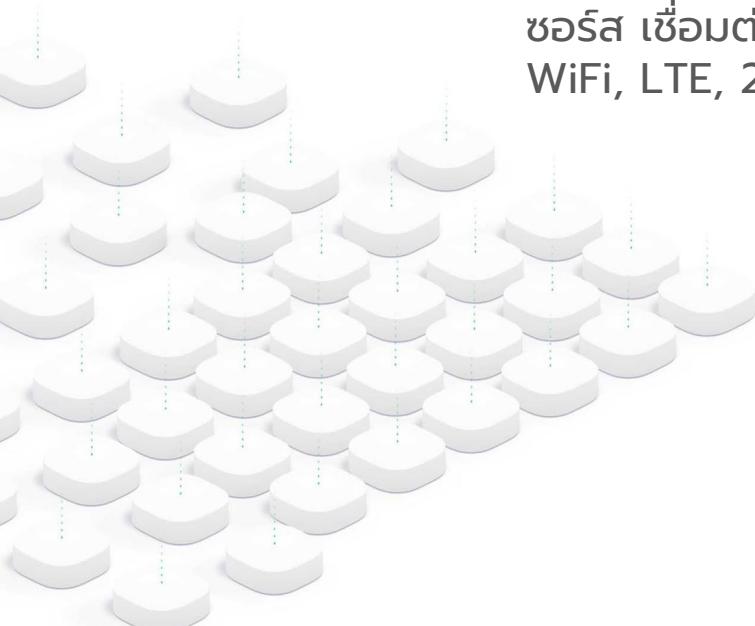
B

Blynk

แพลตฟอร์ม IoT ที่ได้รับความนิยมสูงสุด เพื่อเชื่อมต่ออุปกรณ์ของเรากับคลาวด์ และถูกออกแบบเพื่อควบคุมและจัดการฮาร์ดแวร์ต่างๆ ได้อย่าง halfway



ฮาร์ดแวร์กว่า 400 แบบที่ใช้งานได้กันที่ด้วย Blynk Cloud ที่ร่วดเร็วและโถเพ่นชอร์ส เชื่อมต่อได้อย่างง่ายดายผ่าน WiFi, LTE, 2G-4G หรือ Ethernet



วิดเจ็ตที่ออกแบบไว้หลากหลายเพื่อสร้างแอพ iOS และ Android ควบคุมอุปกรณ์อิเล็กทรอนิกส์ ตรวจสอบข้อมูลเซ็นเซอร์ รับการแจ้งเตือนและอีกมากมาย



ควบคุมฮาร์ดแวร์ของคุณจากทุกที่ในโลก!

สิ่งที่ต้องมีเมื่อใช้



Blynk



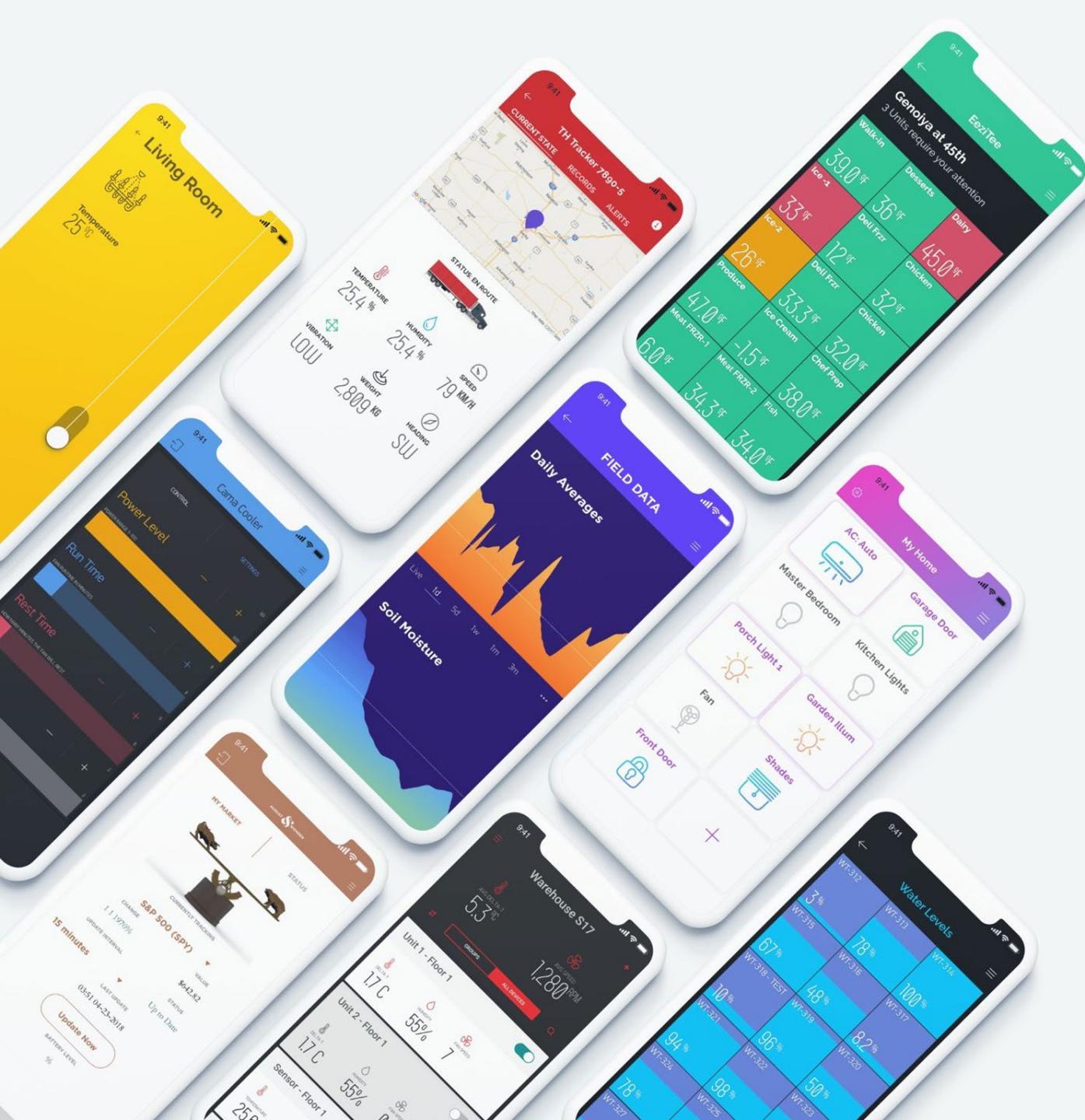
1. มี Smartphone



2. มี Hardware IoT



3. มี Internet





1. มี Smartphone

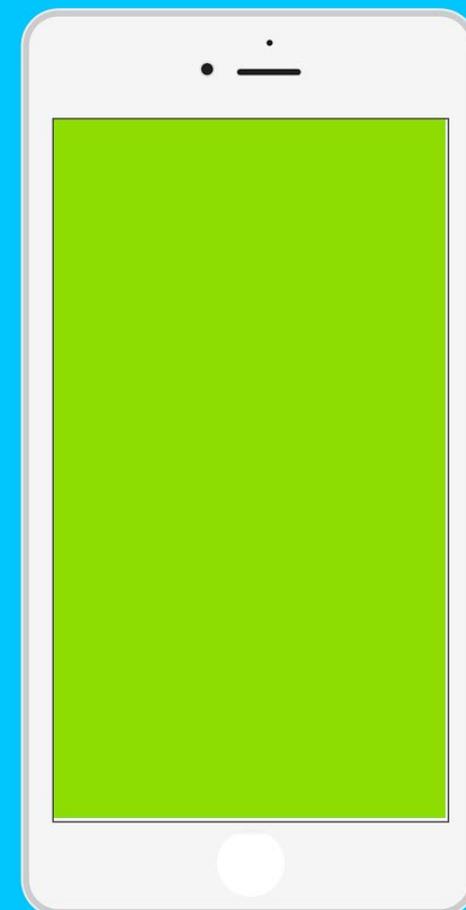
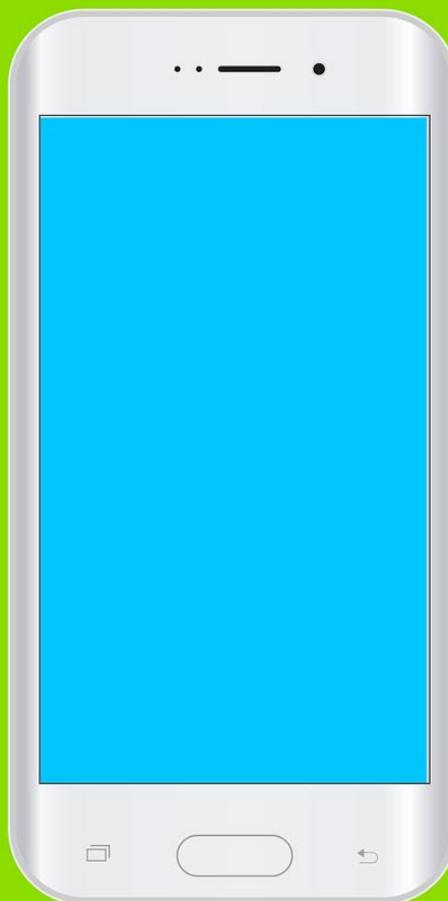


Android OS Version 4.2 ขึ้นไป

iOS Version 9 ขึ้นไป



ทำงานบนคอมพิวเตอร์ไม่ได้ ทำได้
ผ่านซอฟต์แวร์พวงกэмล่องการ
ทำงานเท่านั้น

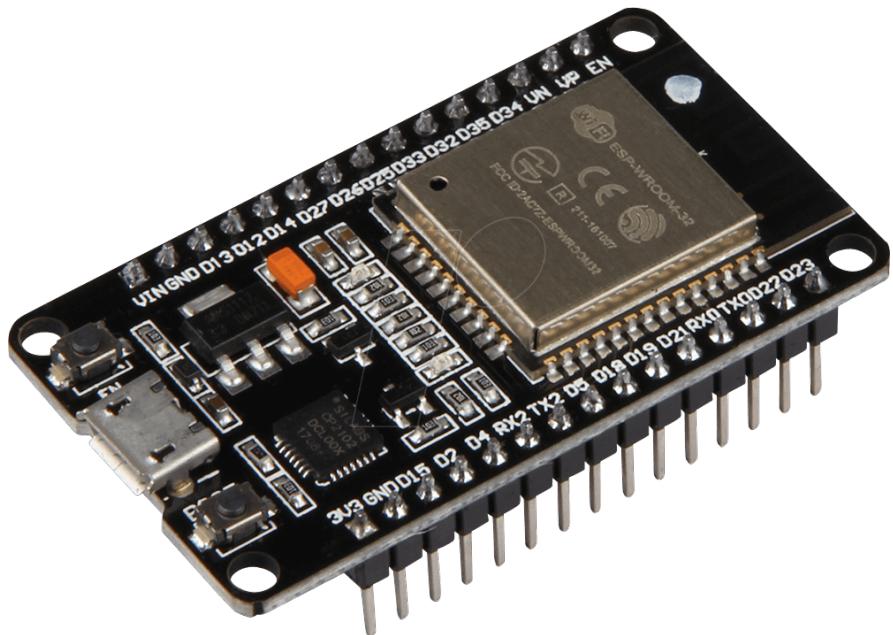


Raspberry Pi



ESP8266

2. І̄ Hardware IoT



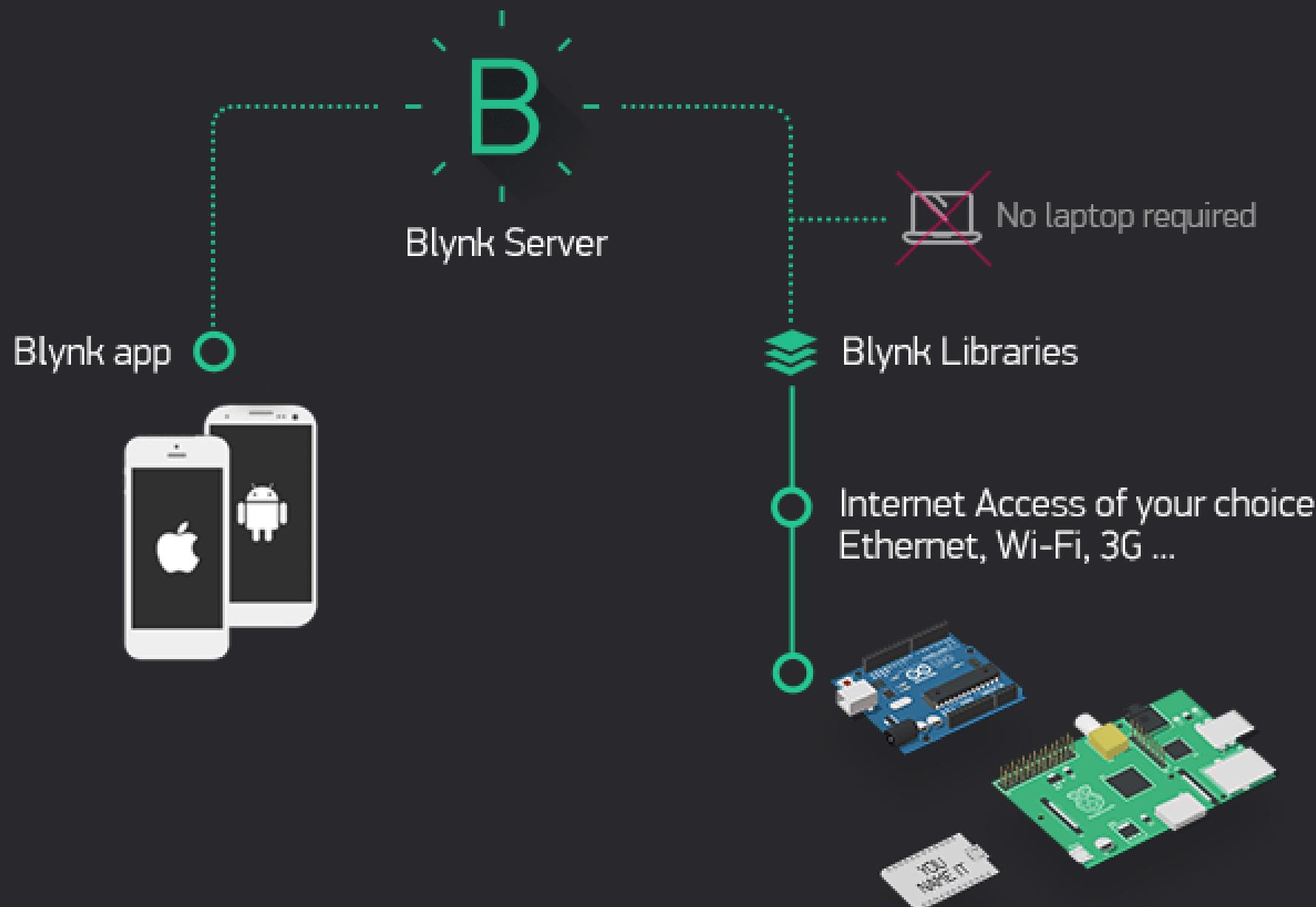
ESP32

3. มี Internet

ฮาร์ดแวร์ที่ใช้งานต้องเชื่อมต่ออินเทอร์เน็ต

- WiFi
- Ethernet
- Cellular (GSM, 2g, 3g, 4g, LTE)
- Serial
- USB via your PC
- Bluetooth (BETA)





1

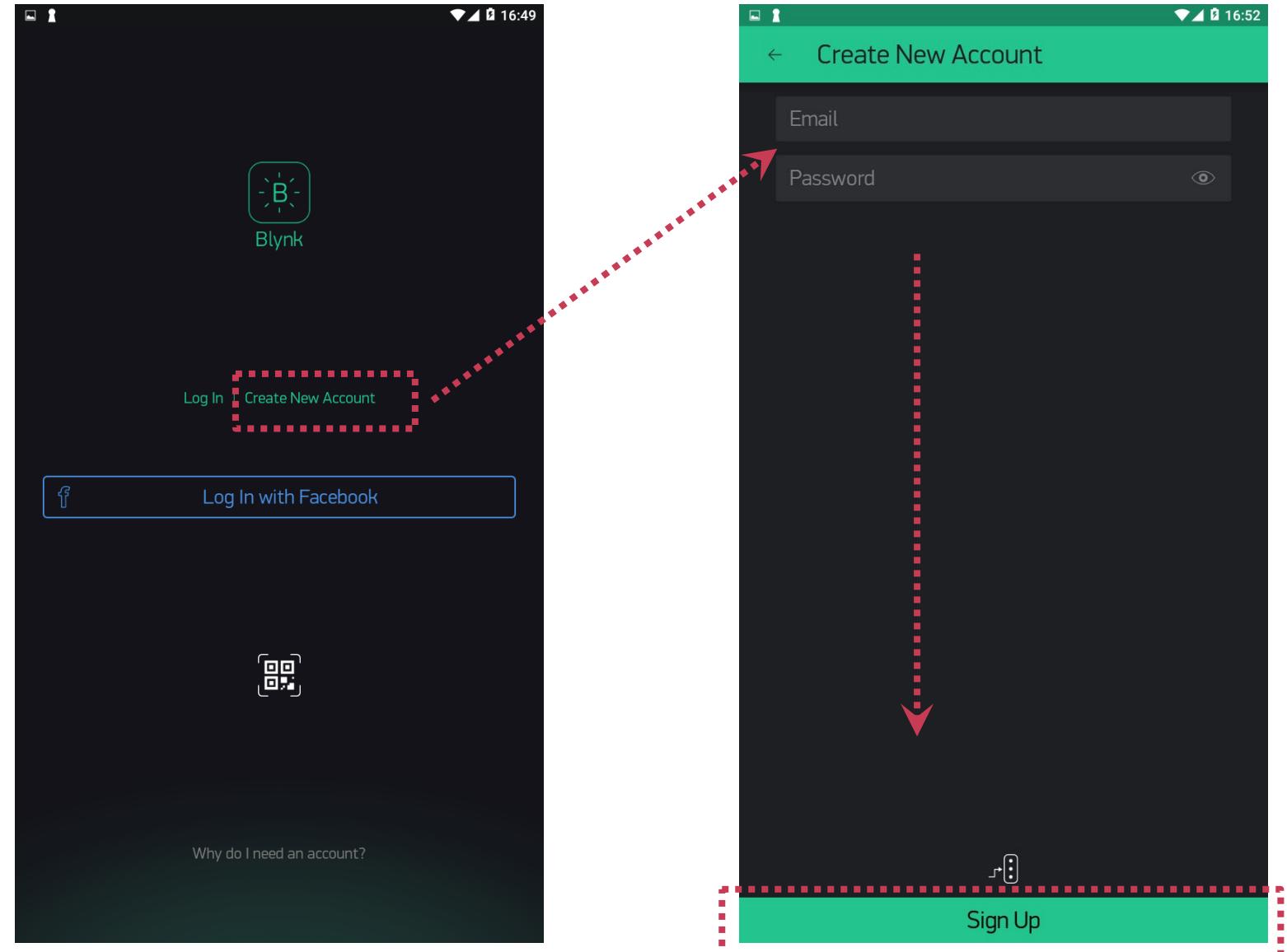
ຕາວໂນໄກສດ App Blynk : 1 ໃຫ້



2

ลงทะเบียน

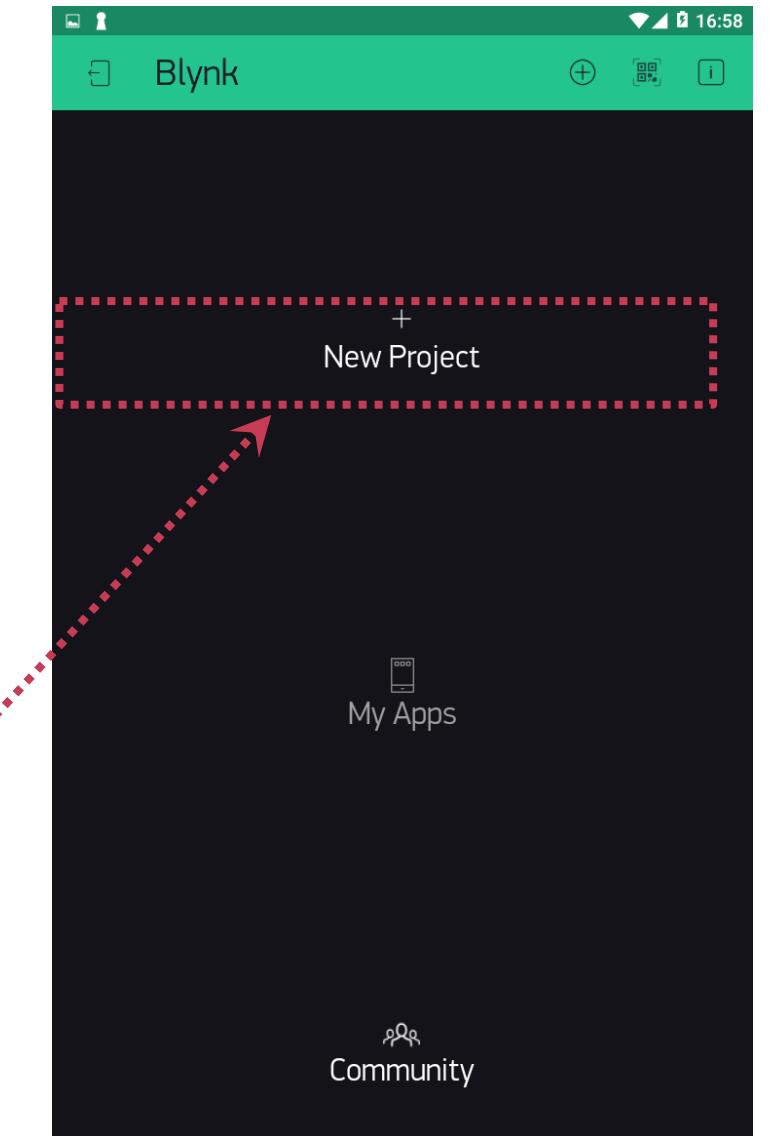
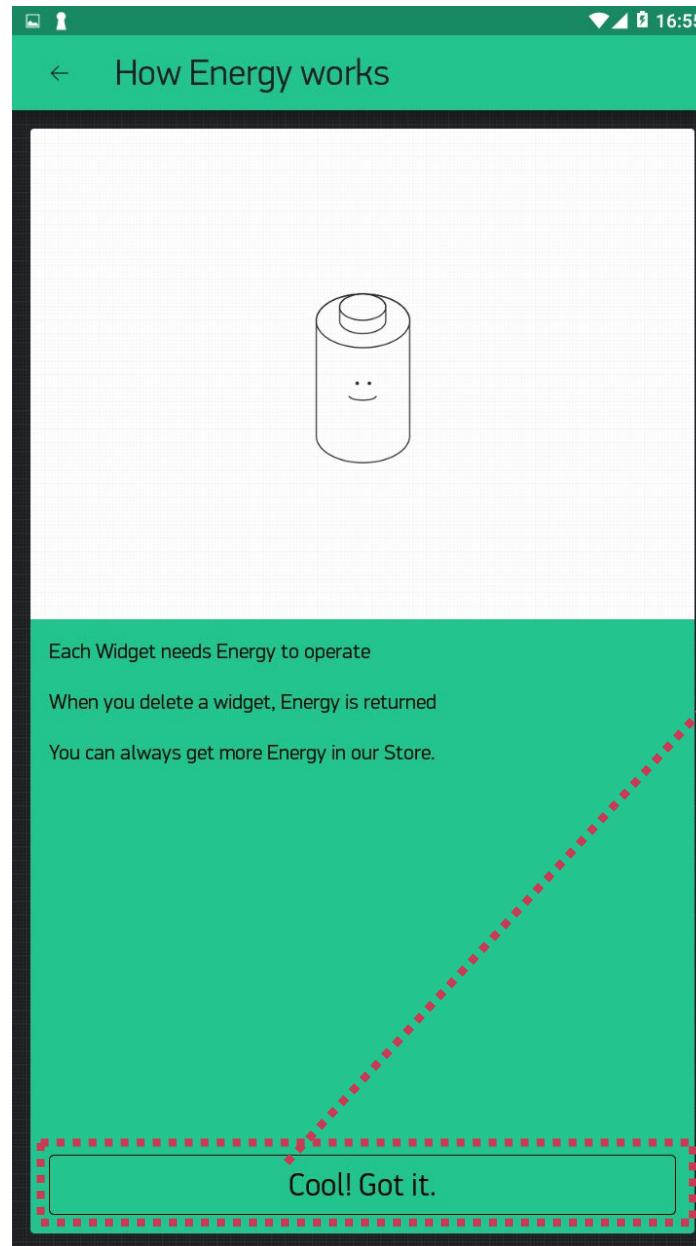
หรือ **Login** ผ่าน **Facebook**
หรือ **สมัคร** ด้วย **Email**
เพื่อรับคีย์ที่ส่งมา



3

สร้าง Project

Blynk จะแจ้งเรื่องพลังงานให้ทราบ
คือถ้าเราสร้างวิดเก็ต เราจะเสีย
พลังงานไป แต่ถ้าลบออก พลังงานจะ
กลับมา อย่างไรเพิ่ม ไปซื้อที่ **ร้านค้า**
ของ Blynk ได้เลย

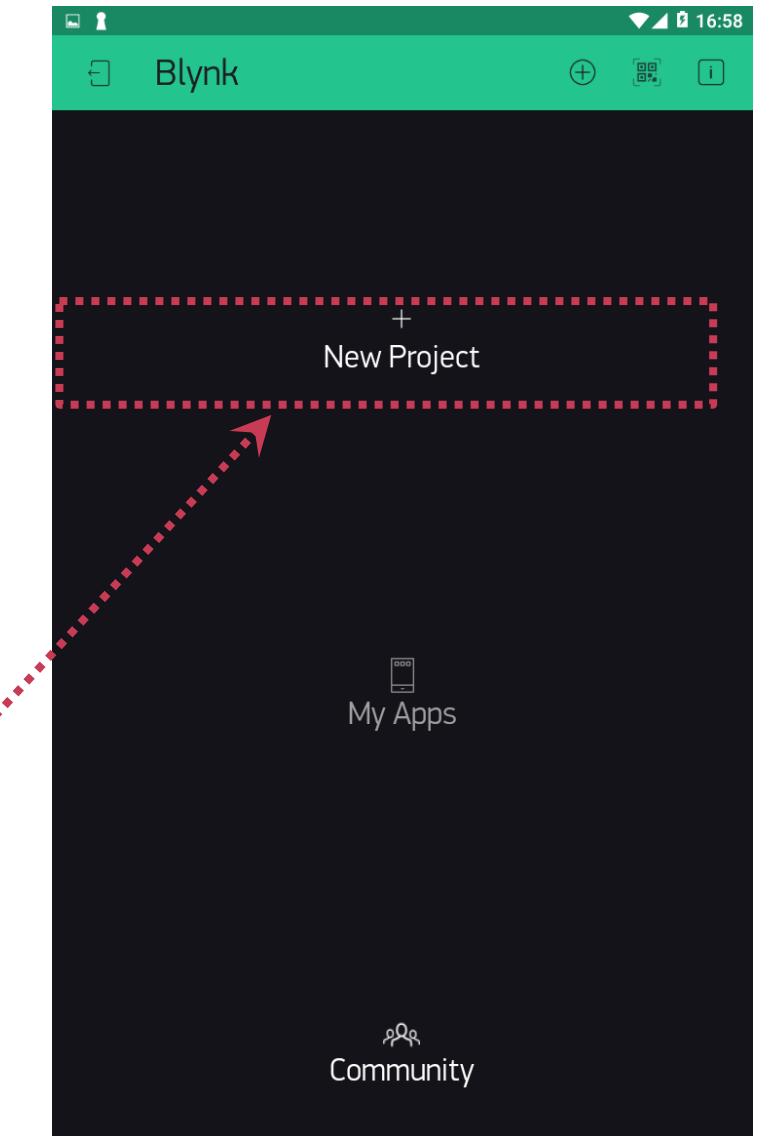
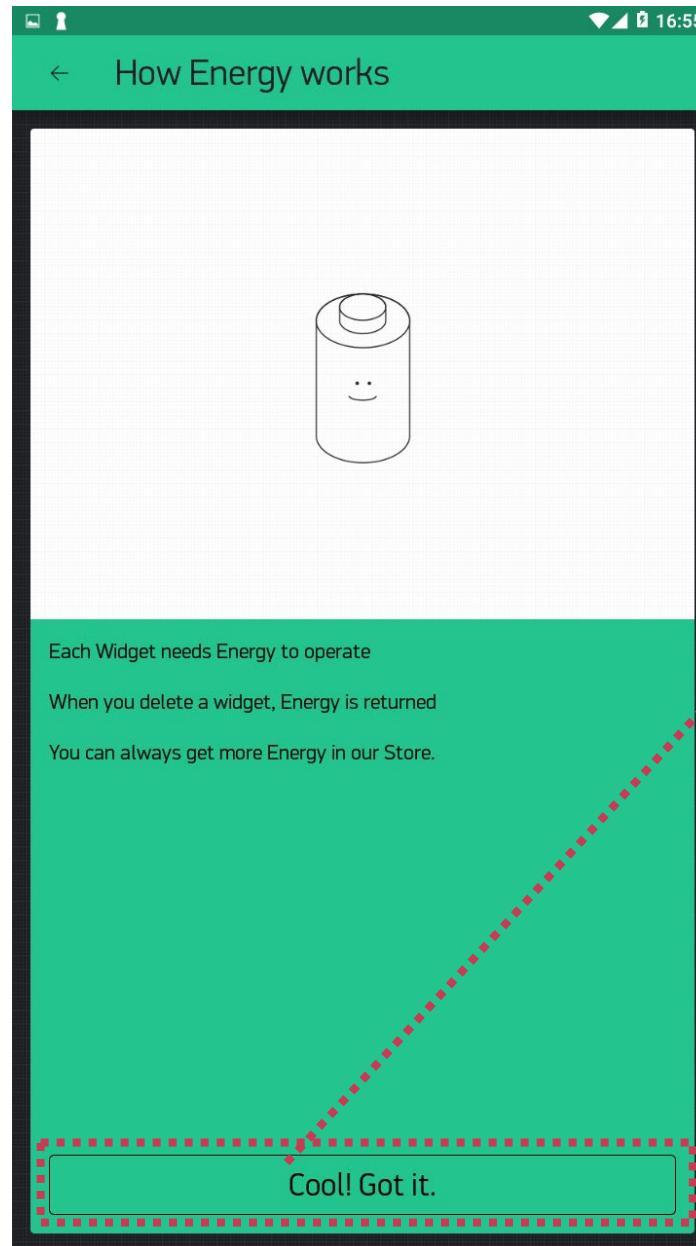


ดัดจากนั้นก็เป็นการสร้าง Project
ใหม่ขึ้นมา โดยเลือกที่ **New Project**

3

สร้าง Project

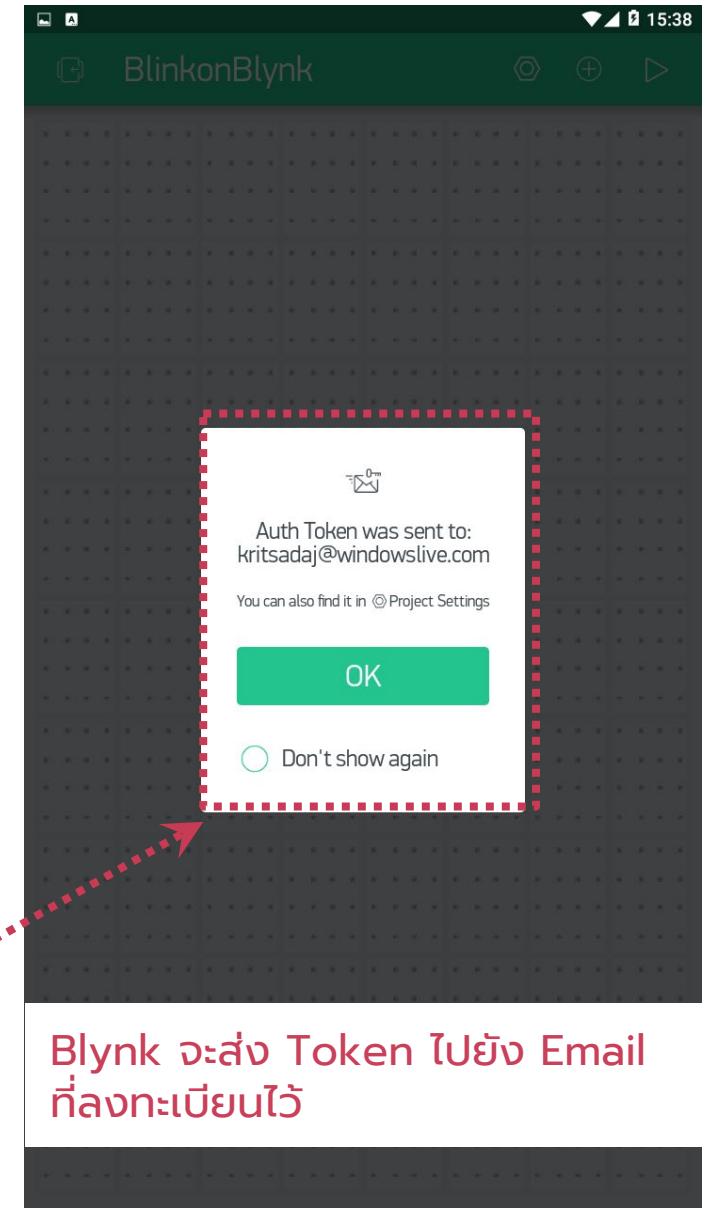
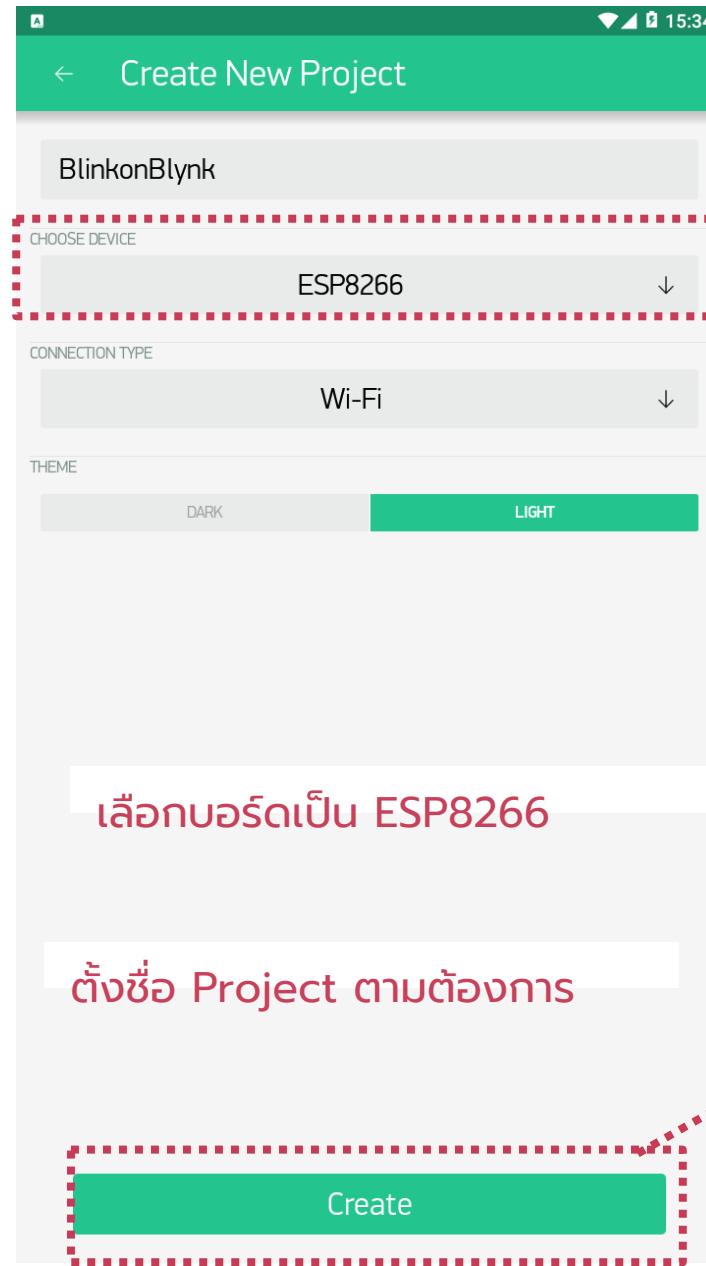
Blynk จะแจ้งเรื่องพลังงานให้ทราบ
คือถ้าเราสร้างวิดเก็ต เราจะเสีย
พลังงานไป แต่ถ้าลบออก พลังงานจะ
กลับมา อยากรู้เพิ่ม ไปซื้อที่ **ร้านค้า**
ของ Blynk ได้เลย



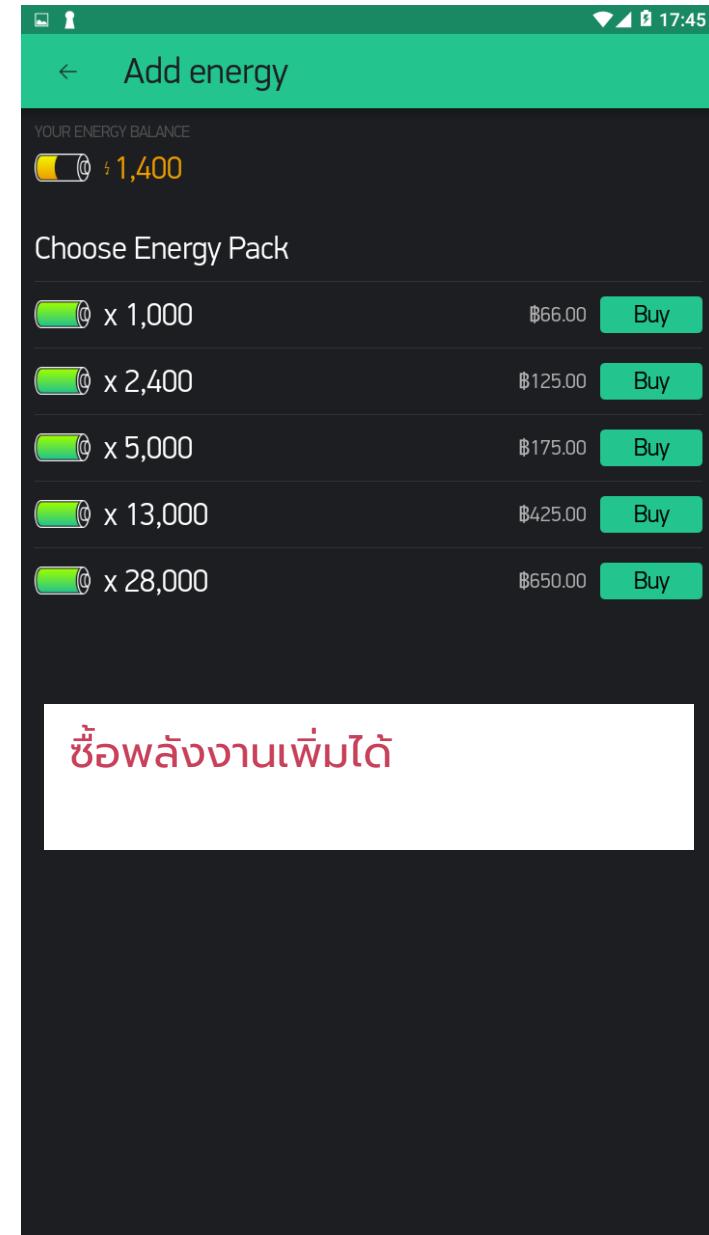
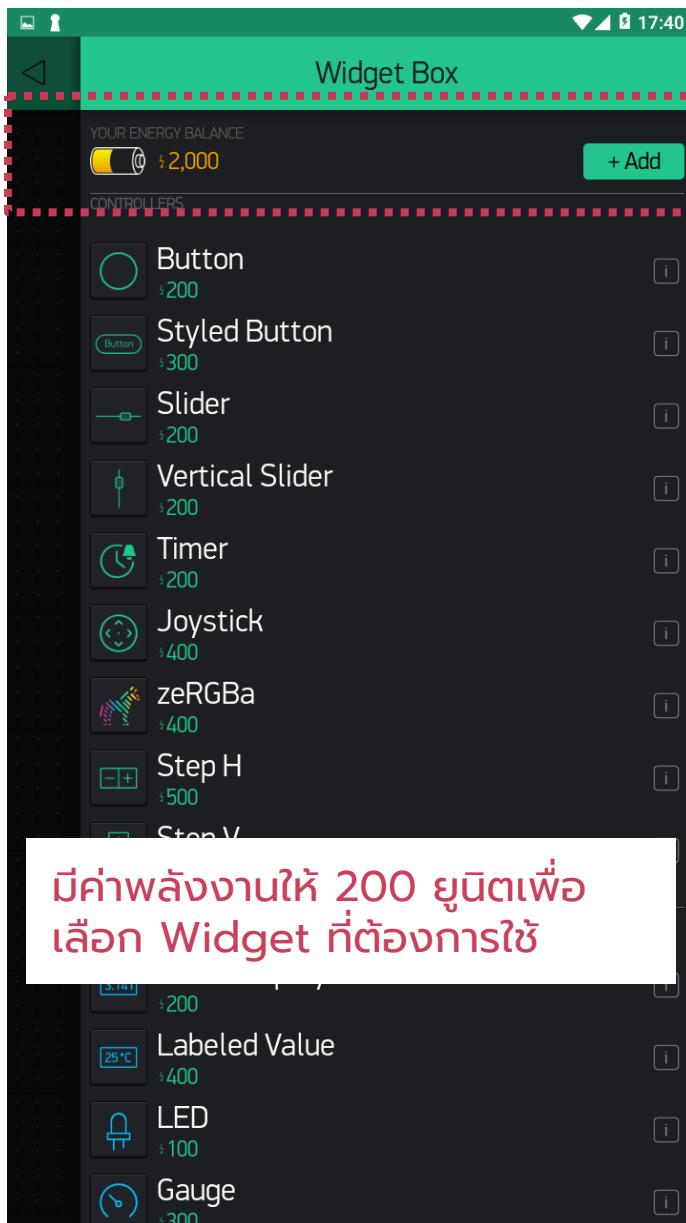
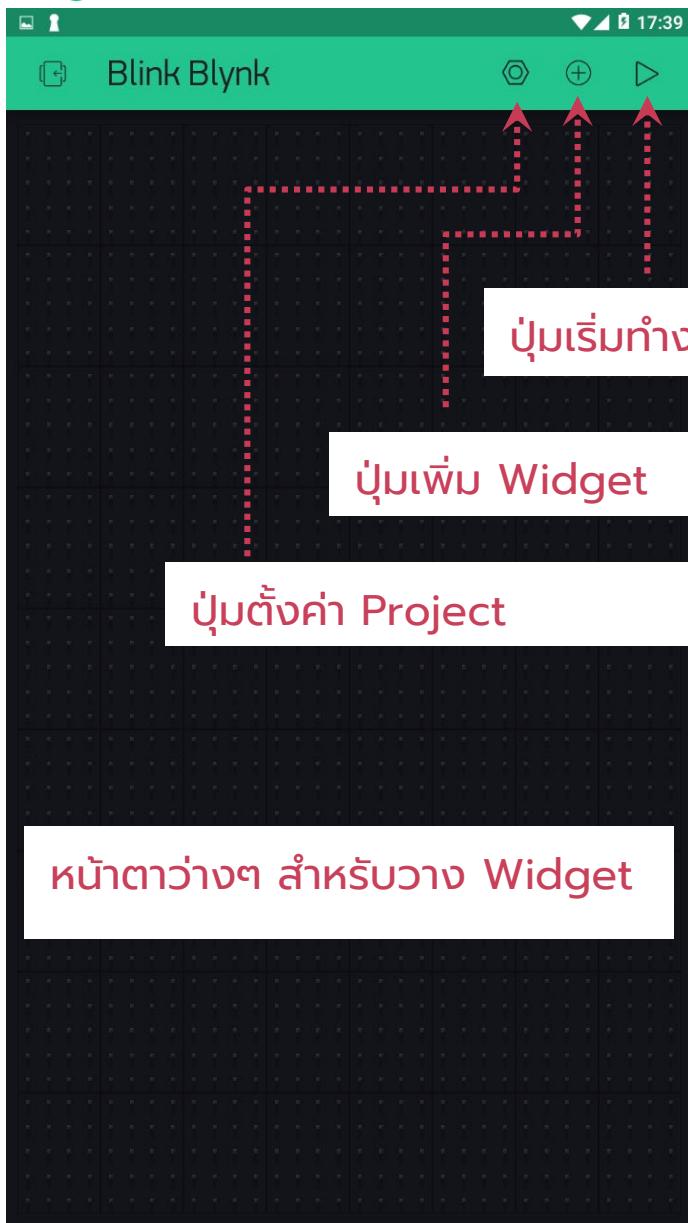
ดัดจากนั้นก็เป็นการสร้าง Project
ใหม่ขึ้นมา โดยเลือกที่ **New Project**

4

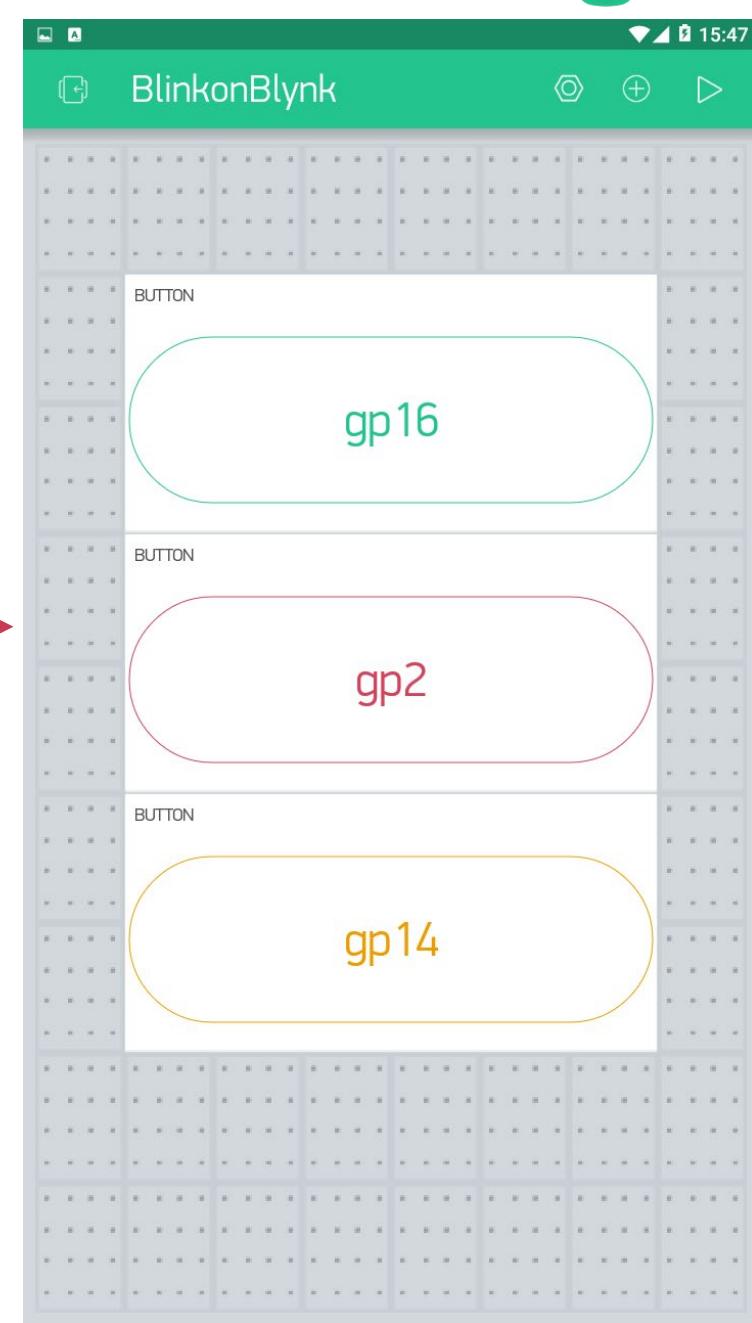
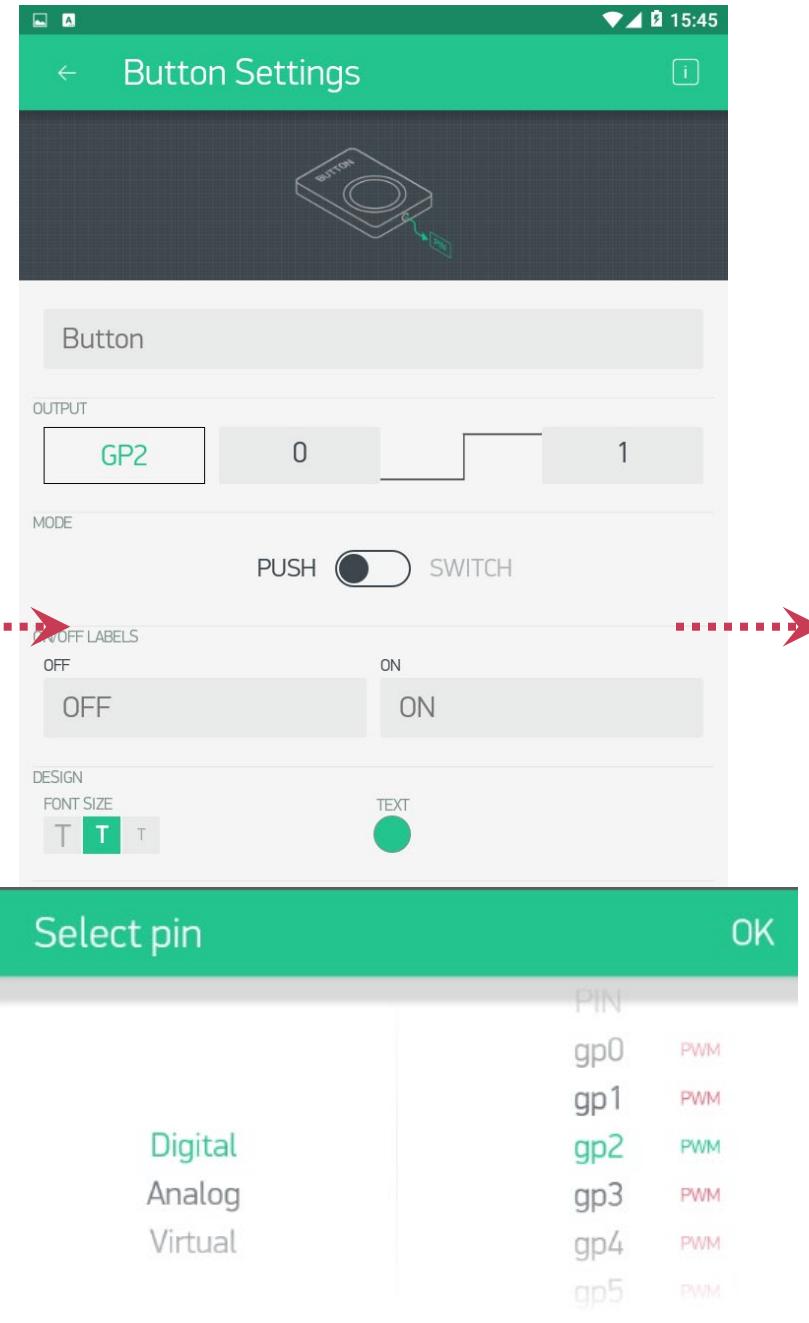
เลือกชาร์ดแวร์



หน้าจอ และ Widget

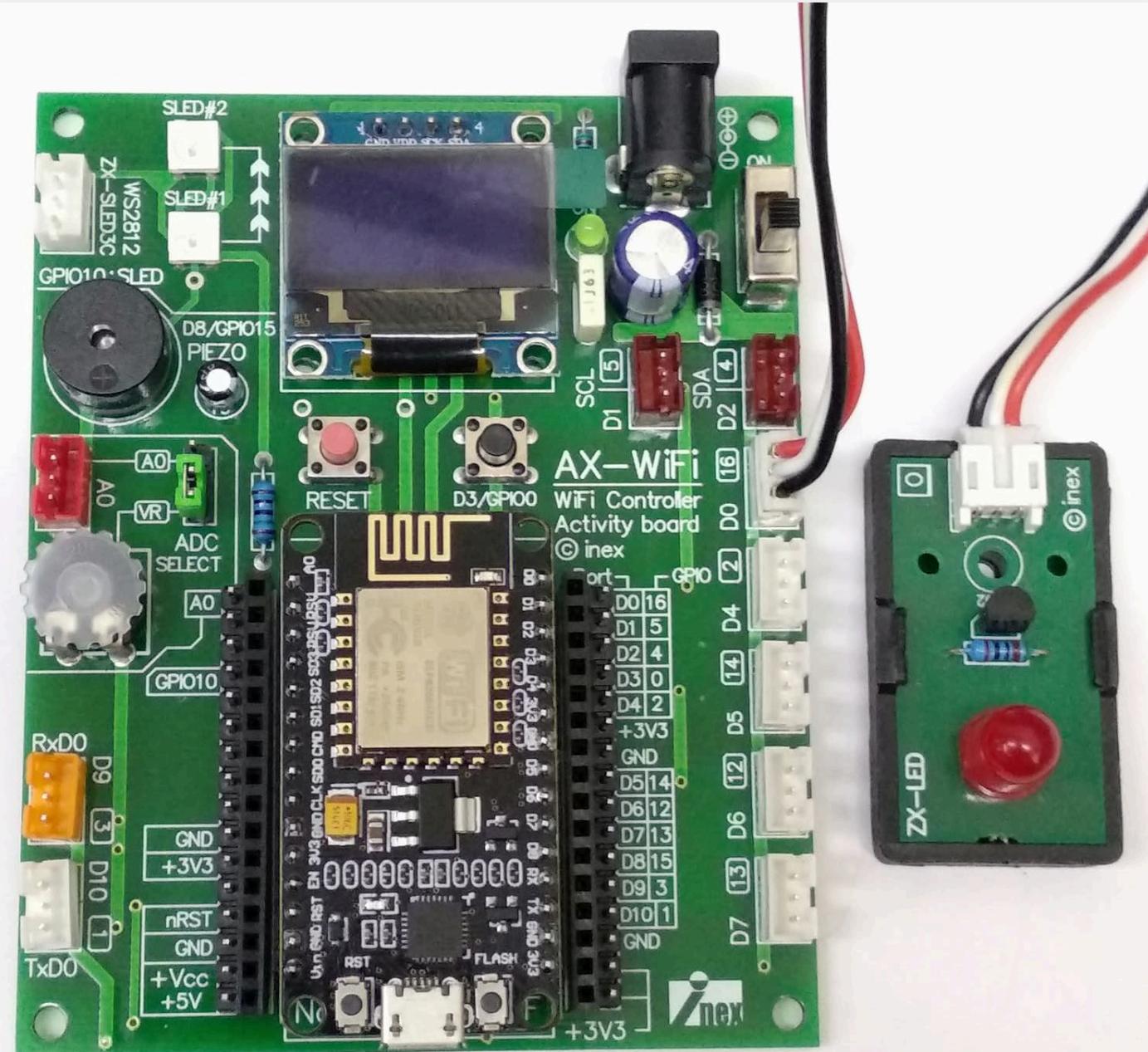


ວາງ widget

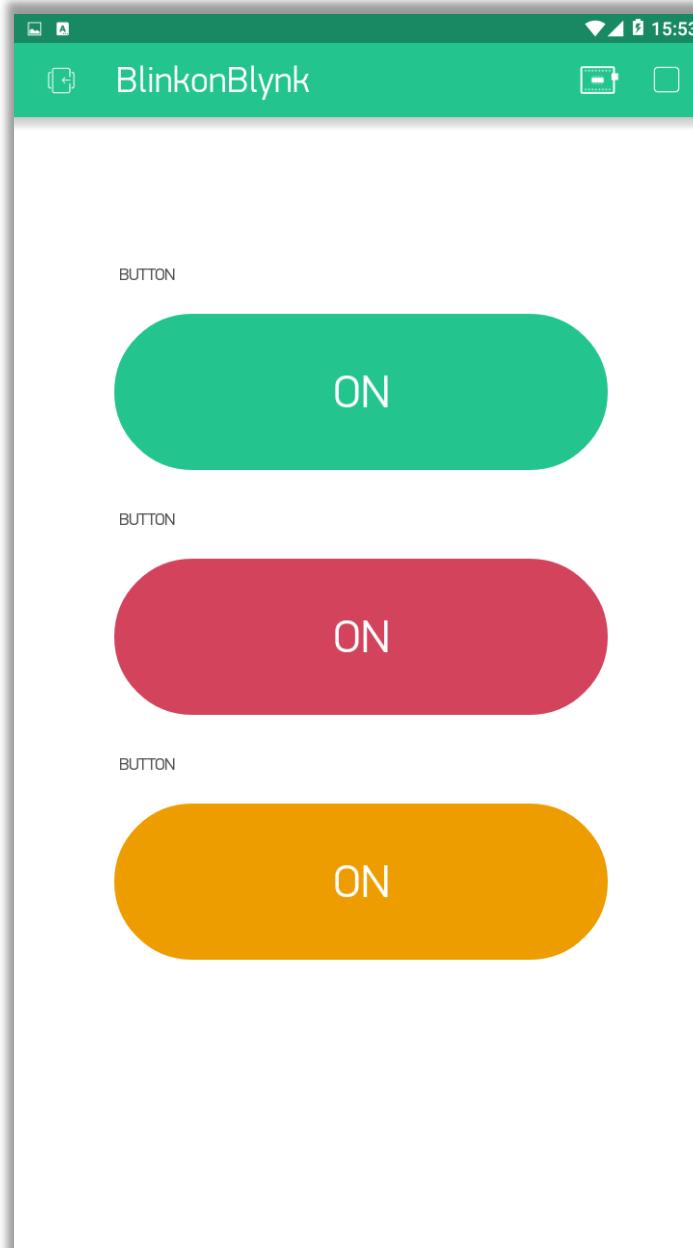


ដែង NodeMCU

តែ ZX-LED ខ្លាក់ខា
GPIO16 (D0)
GPIO2 (D4)
និង GPIO14 (D5)



โค้ดผึ้ง NodeMCU



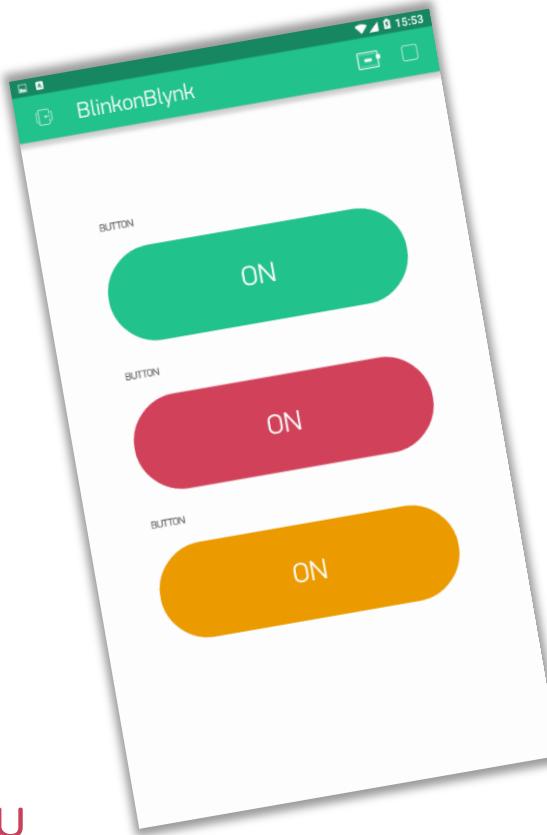
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"

char ssid[] = "Kritsada9";
char pass[] = "0895878268";
char auth[] = "c2253a54ee9a4472bba169b2533a6f48";

void setup() {
    Serial.begin(15200);
    Blynk.begin(auth, ssid, pass);
    ax.begin();
}

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

อธิบายการทำงาน



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

char ssid[] = "Kritsada9";
char pass[] = "0895878268";
char auth[] = "c2253a54ee9a4472bba169b2533a6f48";

void setup() {
    Serial.begin(15200);
    Blynk.begin(auth, ssid, pass);
    ax.begin();
}

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

อธิบายเพิ่มเติม

- เรียกใช้ไลบรารี่ 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
- ปรับปรุงตัวแปรหรือสถานะกึ่งหนดของ ESP8266 ใช้คำสั่ง `Blynk.run();`

การแสดงค่า display

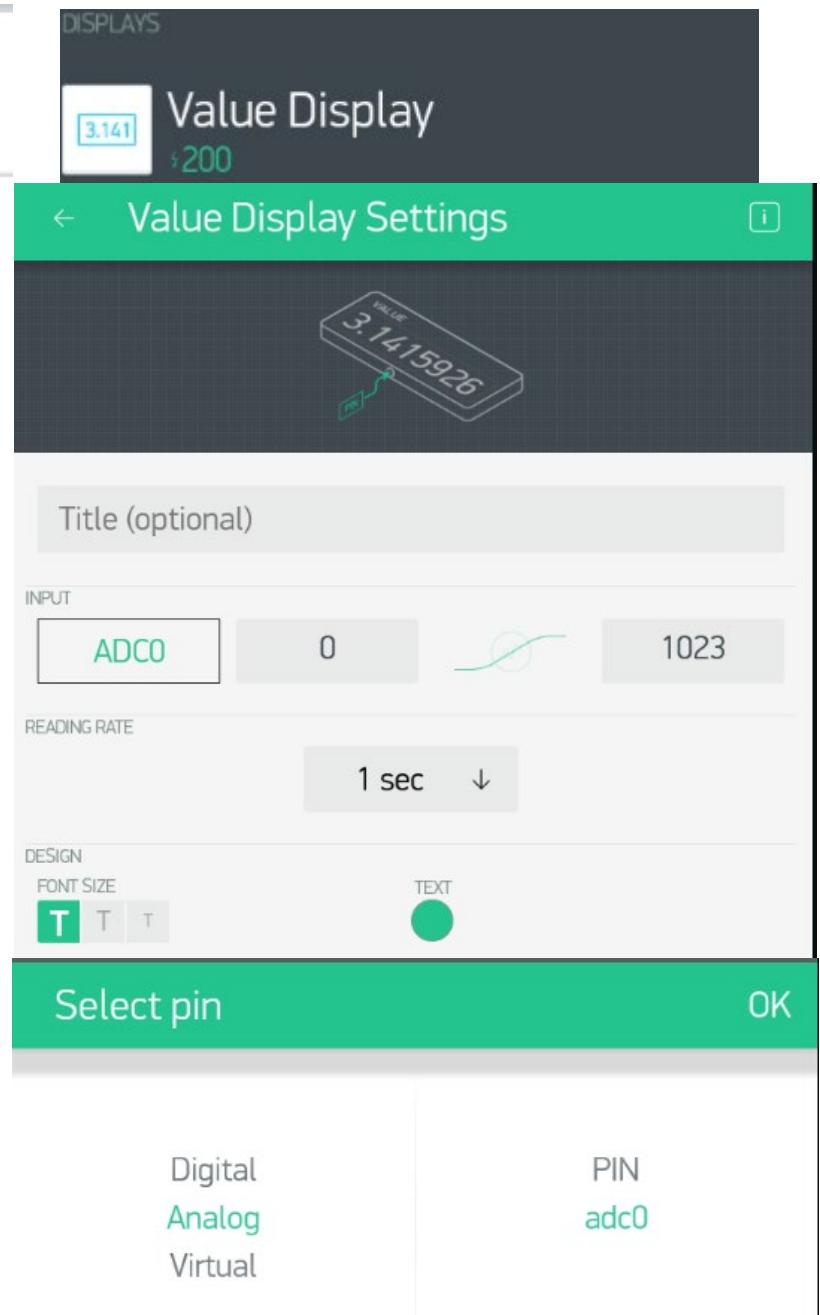
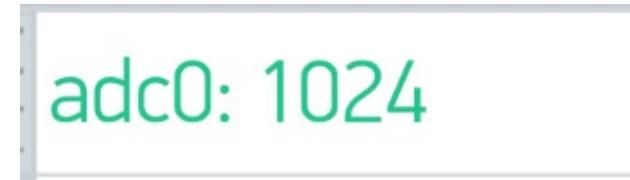
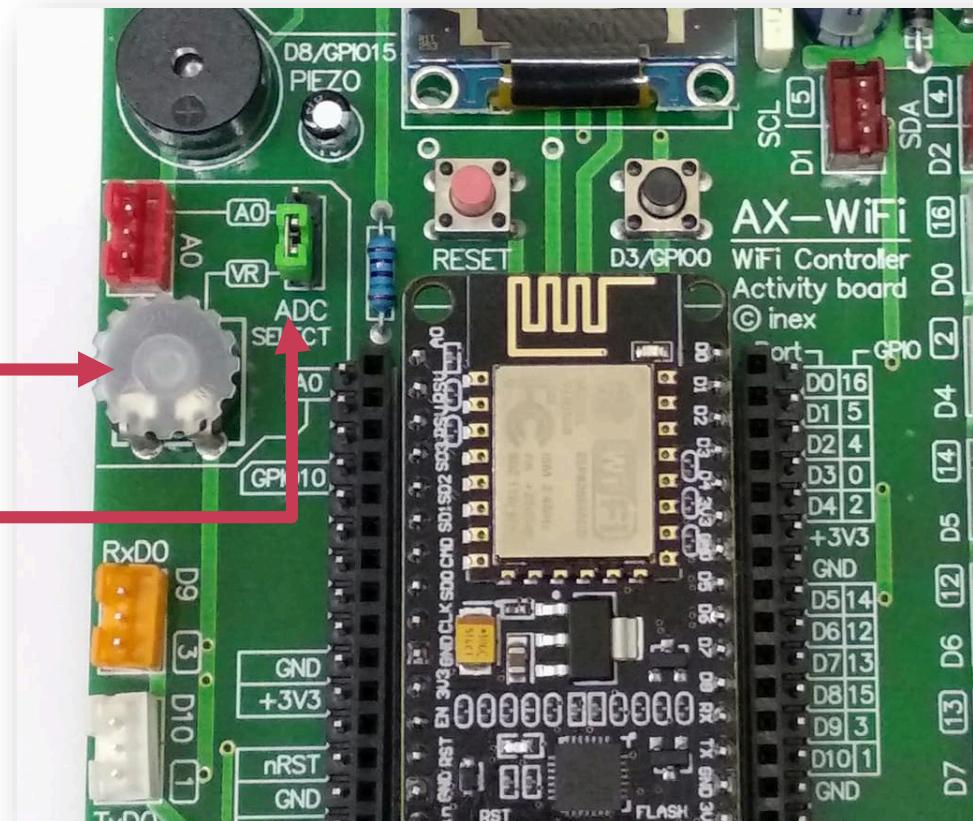
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"
char ssid[] = "Kritsada9";
char pass[] = "0895878268";
char auth[] = "c2253a54ee9a4472bba169b2533a6f48";
void setup() {
    Serial.begin(15200);
    Blynk.begin(auth, ssid, pass);
    ax.begin();
}
void loop() {
    Blynk.run();
}
```

โค้ดเดิม

ปรับ Knob



เลือก Jumper



การแสดงค่า display

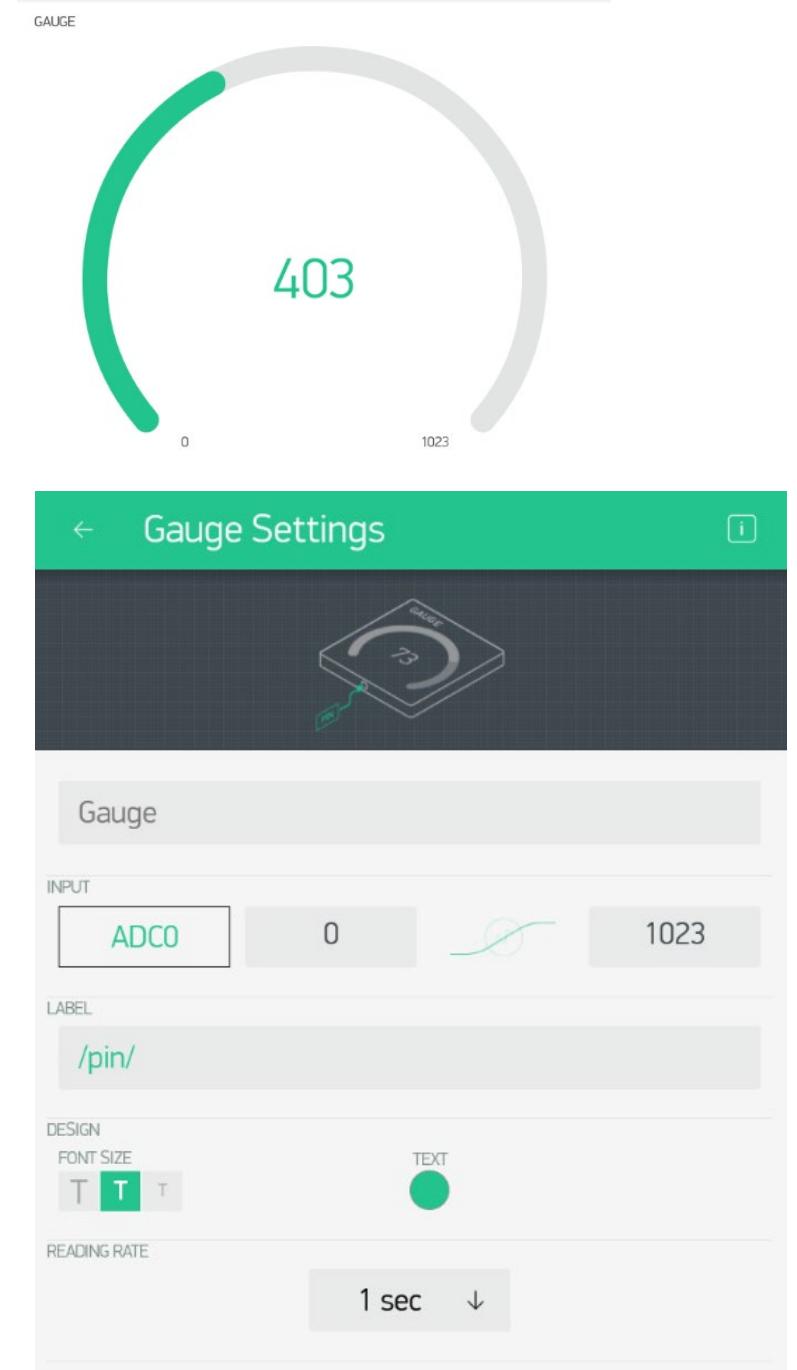
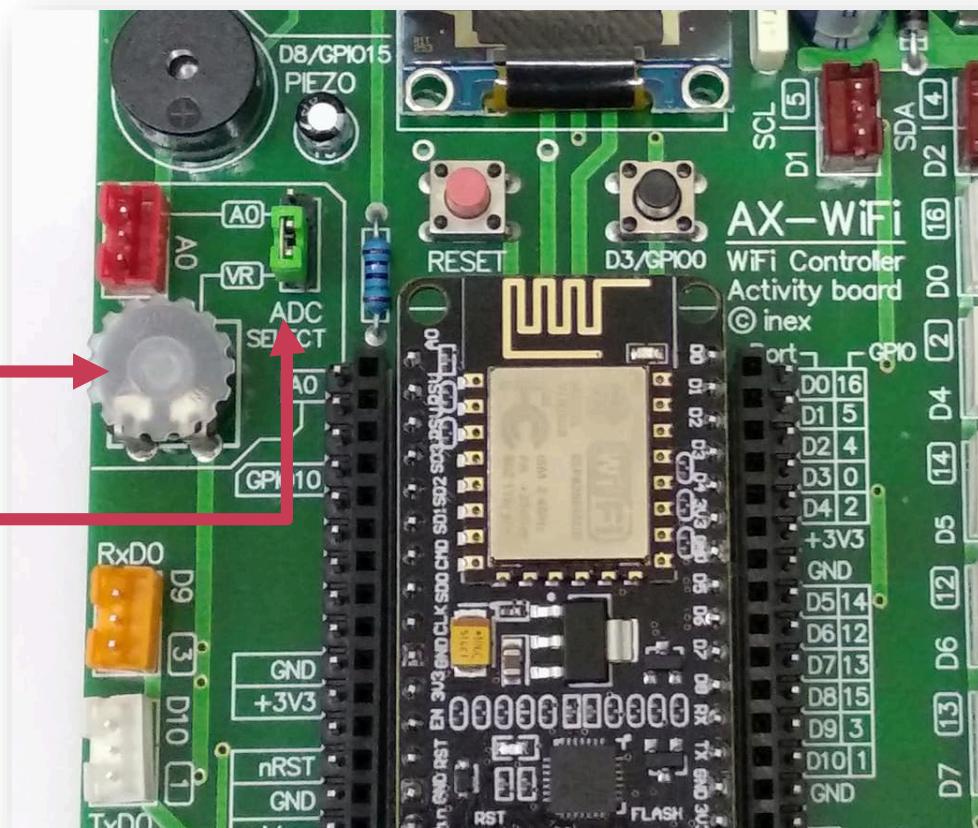
```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"
char ssid[] = "Kritsada9";
char pass[] = "0895878268";
char auth[] = "c2253a54ee9a4472bba169b2533a6f48";
void setup() {
    Serial.begin(15200);
    Blynk.begin(auth, ssid, pass);
    ax.begin();
}
void loop() {
    Blynk.run();
}
```

โค้ดเดิม

ปรับ Knob



เลือก Jumper



การแสดงค่า display

```
#define BLYNK_PRINT Serial
#include <ESP8266WiFi.h>
#include <BlynkSimpleEsp8266.h>
#include "AXWIFI.h"
char ssid[] = "Kritsada9";
char pass[] = "0895878268";
char auth[] = "c2253a54ee9a4472bba169b2533a6f48";
void setup() {
    Serial.begin(15200);
    Blynk.begin(auth, ssid, pass);
    ax.begin();
}
void loop() {
    Blynk.run();
}
```

โค้ดเดิม

ปรับ Knob



เลือก Jumper

