



# อินเทอร์เน็ตของสรรพสิ่ง (Internet of Things)

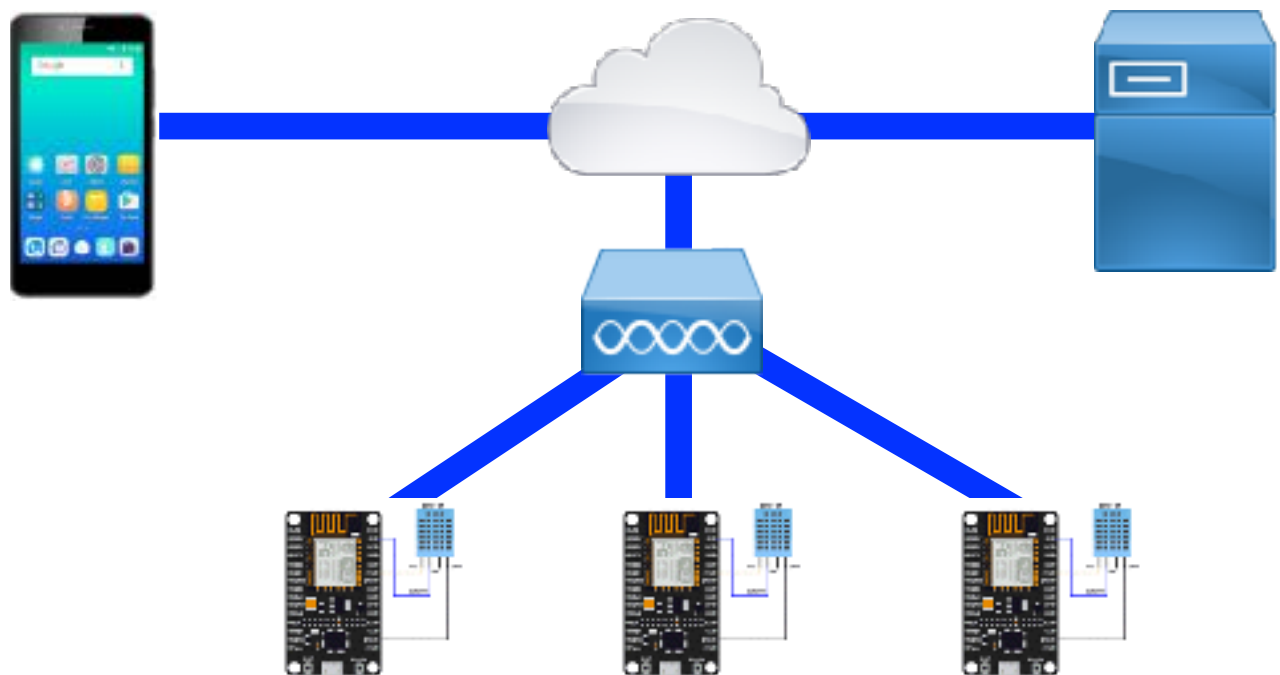
ชนันท์กรณ์ จันแดง และ ทีมวิทยากร  
สาขาวิชาเทคโนโลยีสารสนเทศ  
สำนักวิชาสารสนเทศศาสตร์ มหาวิทยาลัยวลัยลักษณ์



<https://github.com/cjundang/IoTWorkshop6Hr>



# สถาปัตยกรรมระบบ



- User Interface (UI) เป็นส่วนติดต่อผู้ใช้ ซึ่งอาจจะพัฒนาด้วยเทคโนโลยีต่างๆ เช่น เว็บ มือถือแอนดรอยด์ ไอโฟน หรือ โปรแกรมใดๆ
- Server เป็นเครื่องแม่ข่าย สำหรับเก็บข้อมูล อาจจะสร้างขึ้นมาเอง โดยใช้ระบบปฏิบัติการต่างๆ หรือ ใช้ระบบคลาวด์ที่ให้บริการบนอินเทอร์เน็ต
- IoT Gateway เป็นอุปกรณ์ที่เชื่อมต่ออินเทอร์เน็ตและกระจายอินเทอร์เน็ตให้แก่อุปกรณ์อื่นๆ ผ่านสายพาย (WiFi) อาจจะเป็น ADSL Modem
- Sensor Node เป็นอุปกรณ์ฝังตัวที่เชื่อมต่อกับเซนเซอร์ ดิจิทัลและแอนาล็อก เพื่อวัดข้อมูลสิ่งแวดล้อมและควบคุม รวมถึงเชื่อมต่อเครือข่าย

# การประยุกต์อินเทอร์เน็ตของสรรพสิ่ง



## เทคโนโลยี SMART HOME

ควบคุมบ้านอัจฉริยะผ่านอินเทอร์เน็ต

**Appliance Control**

ควบคุมเครื่องใช้ไฟฟ้า ทีวี แอร์ และเครื่องเรือนการทำงานในสมาร์ตโฟน

**Media Center**

ระบบเล่นดนตรี คอมพิวเตอร์ต่างๆ ไร้สาย

**Smart Kitchen**

ห้องครัวสามารถสื่อสารกับคุณได้โดยตรง ภายใต้แนวคิด Internet of Things

**Landscape Control**

เปิด-ปิด รดน้ำต้นไม้เมื่อถึงเวลา ชวนหมาไปเดินเล่น

**Home Monitoring**

สอดส่องดูแลความเคลื่อนไหวภายในบ้านตลอด 24 ชั่วโมง

**Home Control**

ควบคุมระบบไฟฟ้า ภูมิอากาศ อุปกรณ์ทุกชนิดภายในห้อง

ANANDA DEVELOPMENT

UPPER-LEVEL SOLUTIONS

ANANDA\_CULTURE/ANANDA\_DEVELOPMENT

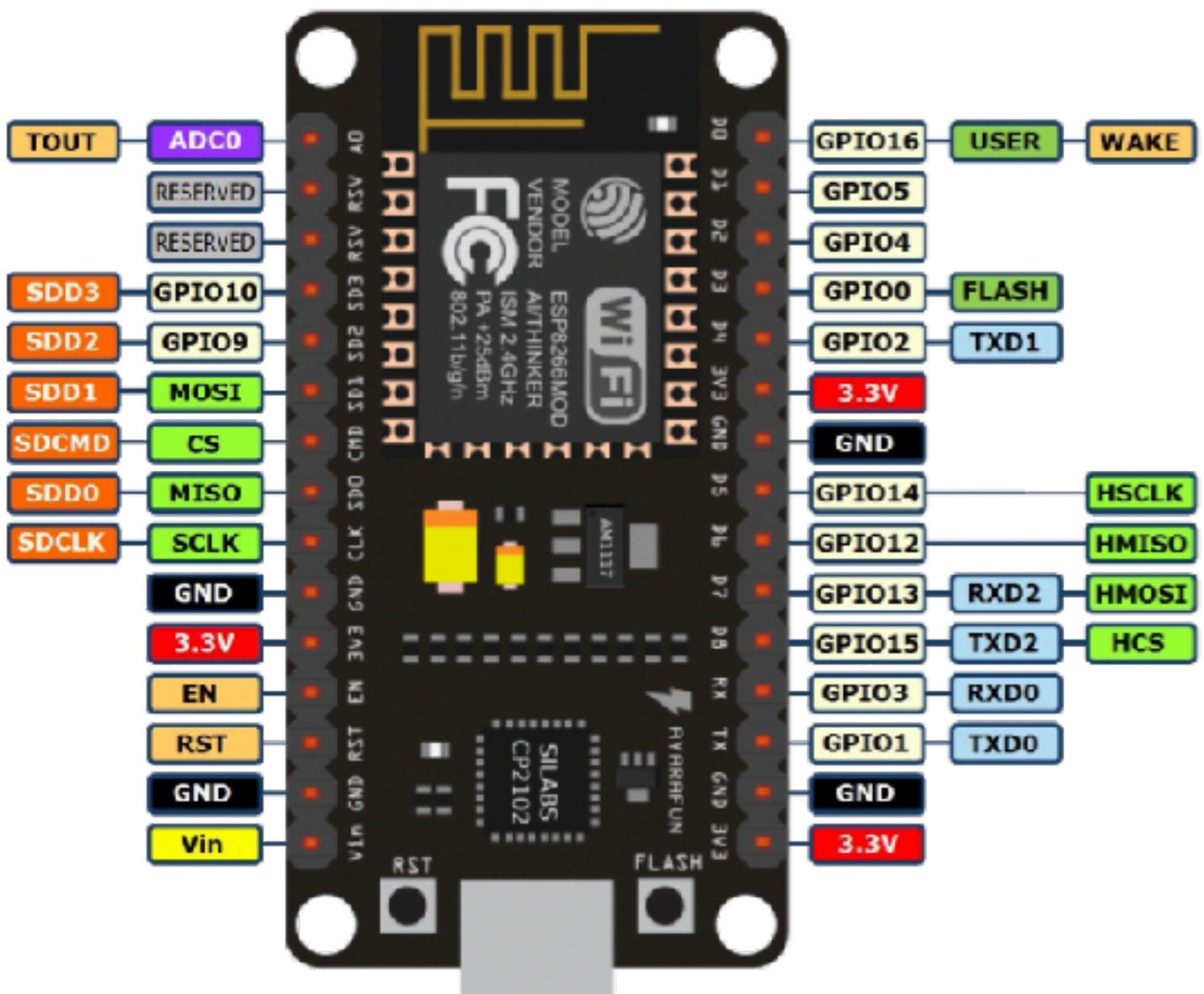
# การประยุกต์อินเทอร์เน็ตของสรรพสิ่ง





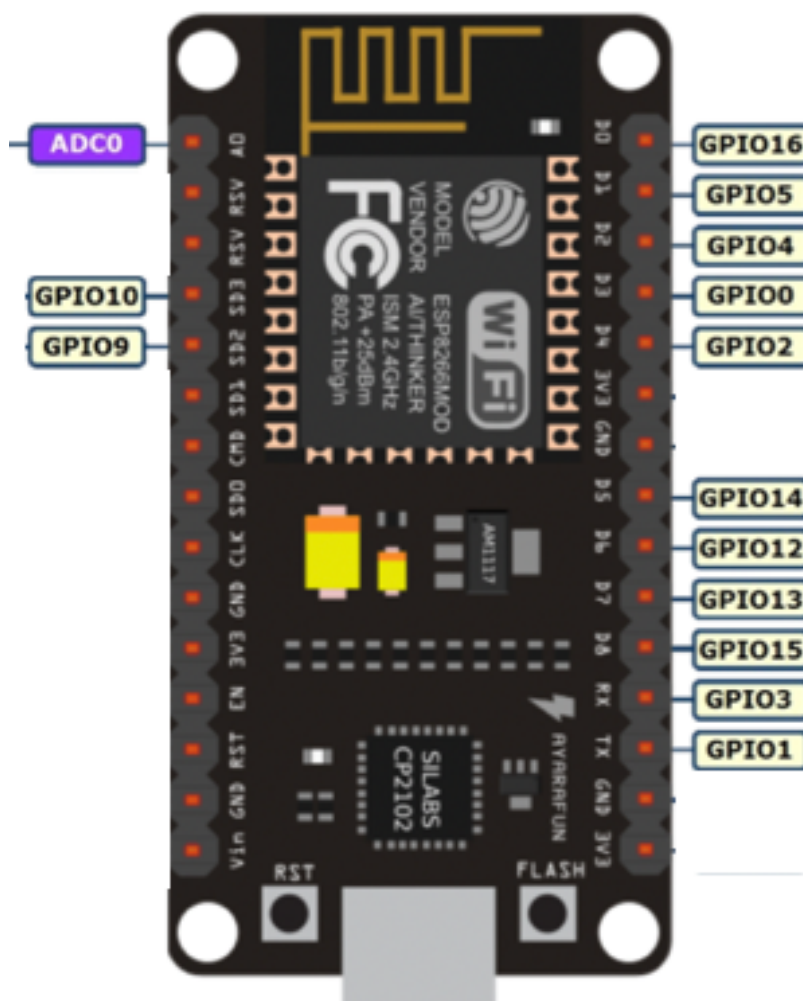
# ESP8266-NodeMCU

- ESP8266 เป็นโมดูลสำหรับเชื่อมต่อเครือข่ายวายฟาย ซึ่งมีพิน (pin) สำหรับเชื่อมต่อกับโมดูลอื่นๆ
- NodeMCU เป็นโมดูลส่วนขยายที่เชื่อมต่อกับ ESP8266 เพื่ออำนวยความสะดวกในการพัฒนาระบบ



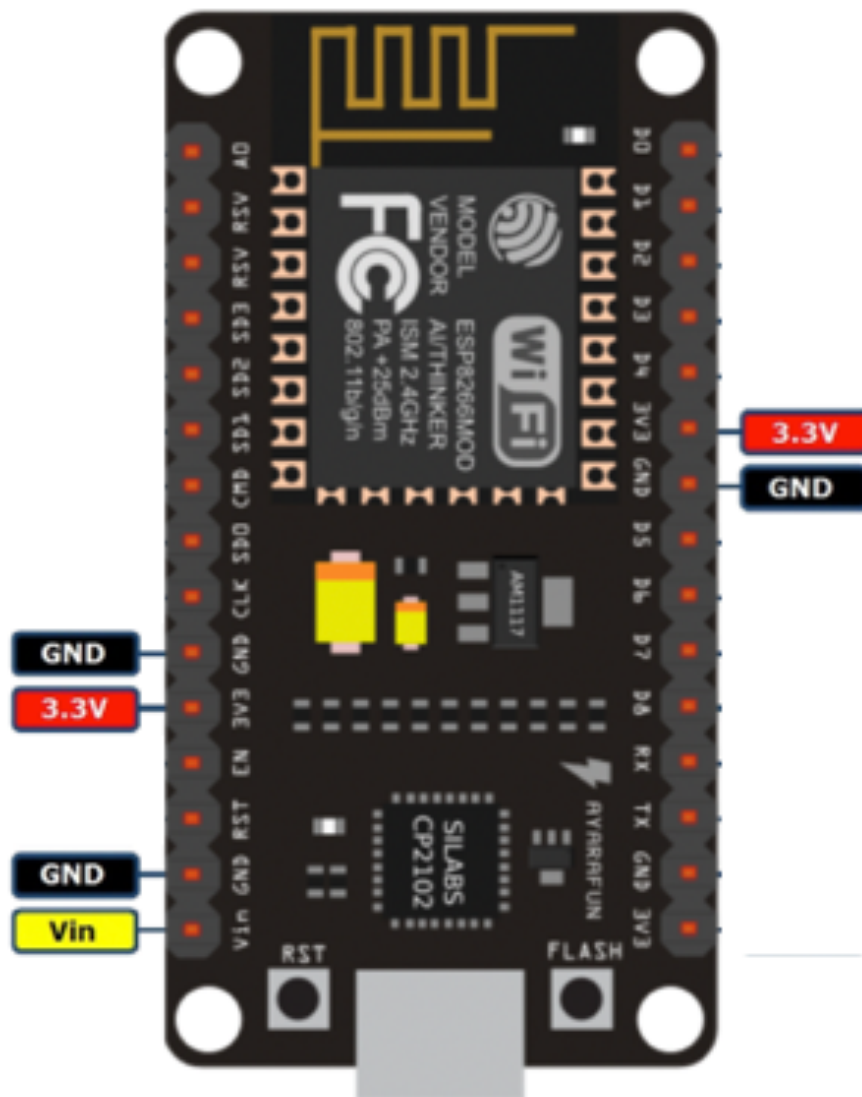
# Pin

- GPIOx เป็นพินที่ใช้เชื่อมต่อกับอุปกรณ์ทั่วไป ถูกกำหนดหน้าที่เมื่อเริ่มใช้งาน ไม่มีหน้าที่จำเพาะเจาะจง
  - อาจจะเรียกแทนด้วยหมายเลขเดี่ยวๆ ได้ เช่น GPIO9 ตอนโปรแกรมแทนด้วย 9
  - หรือมีชื่อเรียกพิเศษตามบอร์ด เช่น GPIO16 อาจจะเรียกว่า D0
  - เป็นการสื่อสารแบบดิจิทัล
- ยกเว้น A0 เป็นการสื่อสารแบบแอนาล็อก



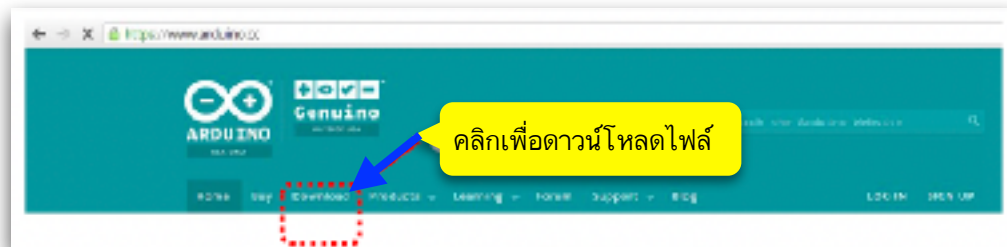
# Pin

- GND เป็นพิน Ground เป็นแรงดันไฟ 0 สำหรับไว้อ้างอิงกันพินอื่นๆ
  - 3.3V เป็นแหล่งจ่ายไฟบนบอร์ดขนาด 3 โวลต์
  - Vin เป็นแหล่งจ่ายไฟบนบอร์ดขนาด 5 โวลต์



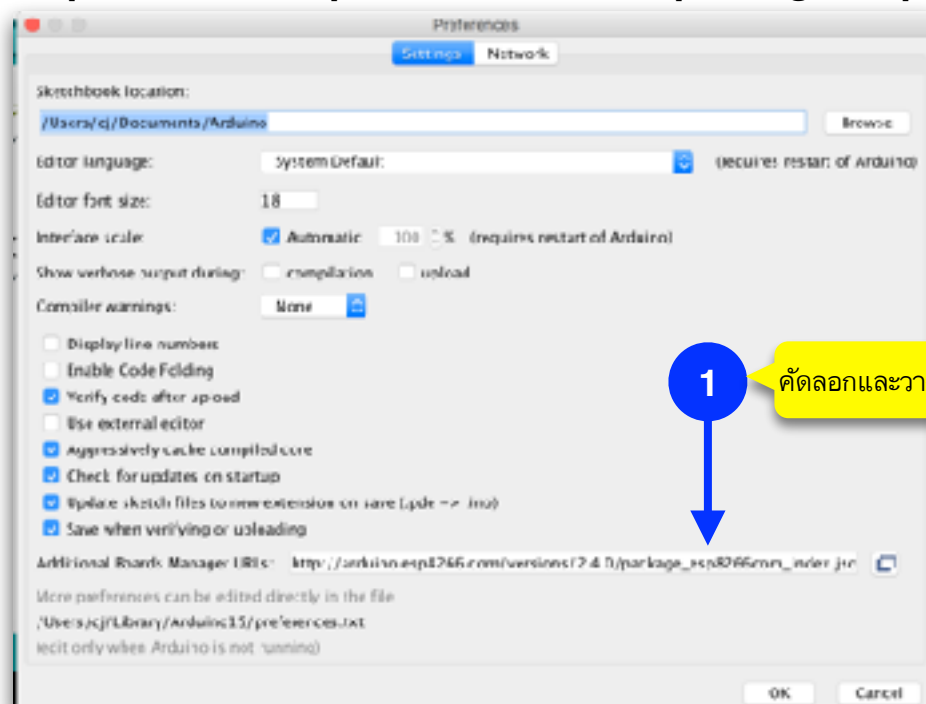
# โปรแกรม Arduino IDE

- Download Arduino IDE จาก [www.arduino.cc](http://www.arduino.cc) เลือกตามสถาปัตยกรรมของเครื่อง



- โปรแกรม ที่ flash ในบอร์ด ประกอบด้วย
  - โปรแกรมส่วนที่ผู้พัฒนาเขียนมา - ฝึก ใน workshop นี้
  - Firmware ของบอร์ดนั้นๆ
    - ตั้งค่า โดย เลือกเมนู File > Preference หลังจากนั้นพิมพ์ต่อไปนี้ ใน Additional Boards Manager URLs แล้วกด OK

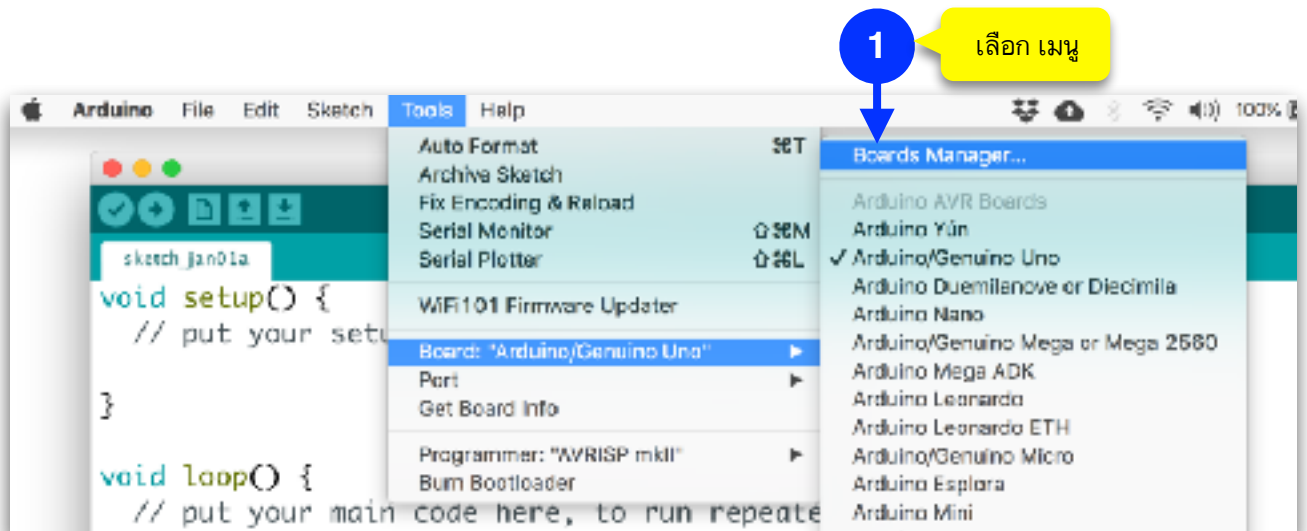
[http://arduino.esp8266.com/stable/package\\_esp8266com\\_index.json](http://arduino.esp8266.com/stable/package_esp8266com_index.json)





# โปรแกรม Arduino IDE

- เลือกเมนู Tools > Boards > Boards Manager

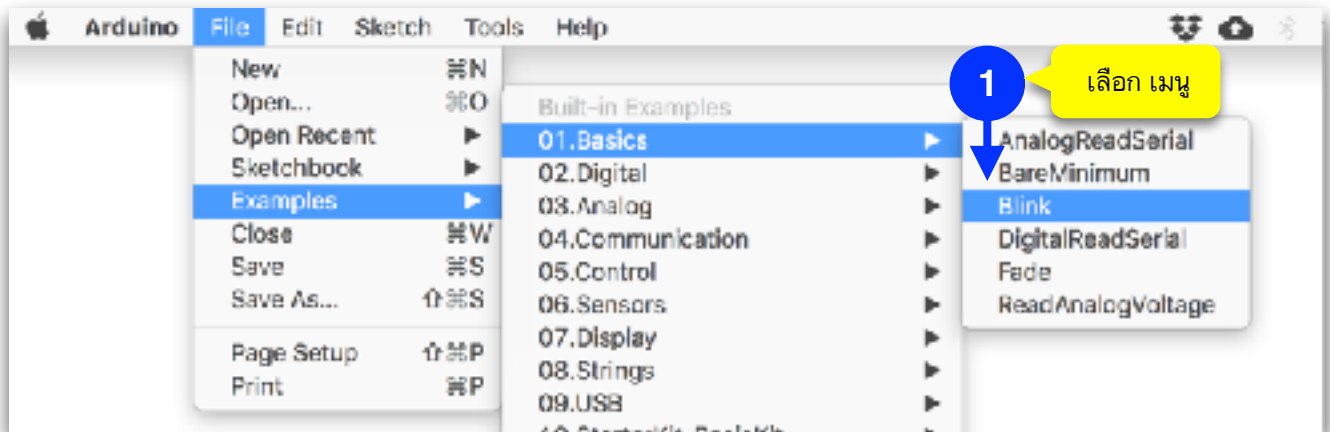


- แล้วพิมพ์ **ESP8266** เพื่อค้นหา Package จากนั้น กด Install

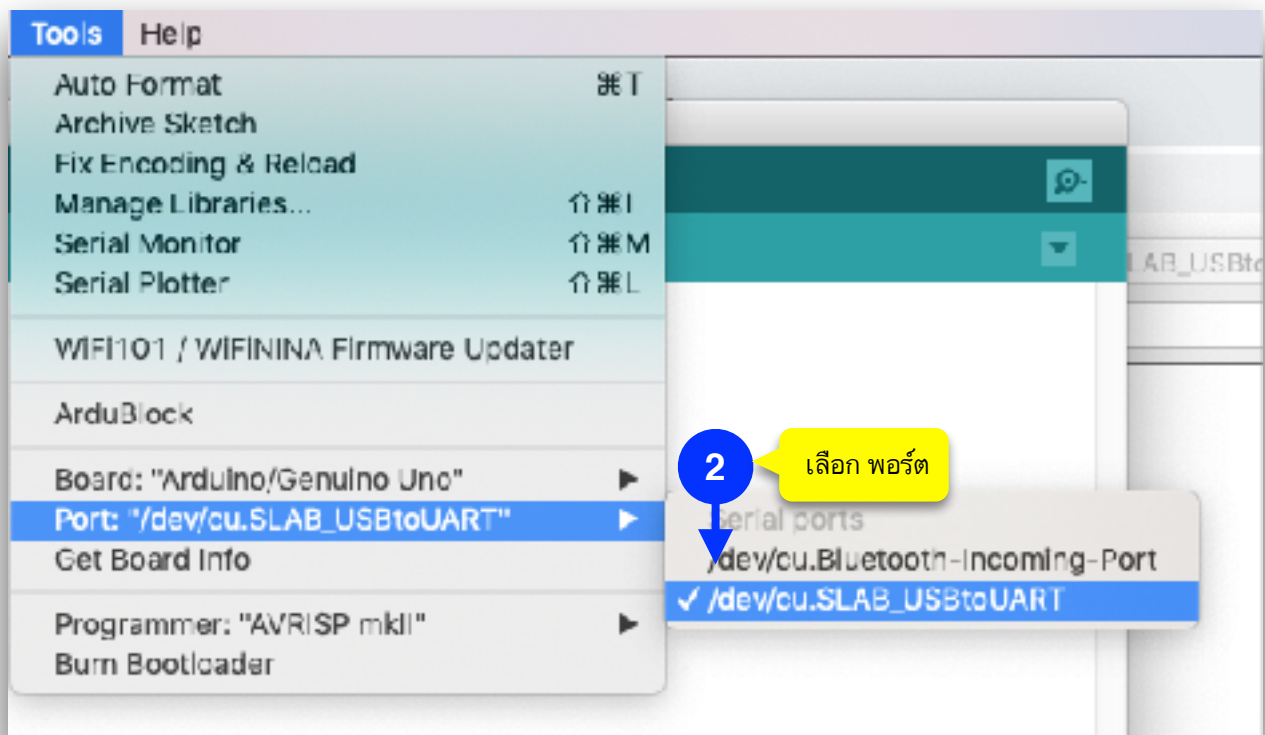


# การ Flash โปรแกรมเข้าบอร์ด

- เลือก File > Examples > 01.Basics > Blink เพื่อสร้างโปรแกรมตัวอย่าง



- ให้ต่อสาย USB กับ NodeMCU และเครื่องคอมพิวเตอร์ หลังจากนั้นก็ตรวจสอบชนิดของพอร์ต ซึ่งอาจจะเป็น COM3, COM4, หรืออื่นๆ



# การตรวจสอบชื่อ port ใน MSWindow10

- เลือก File > Examples > 01.Basics > Blink เพื่อสร้างโปรแกรมตัวอย่าง

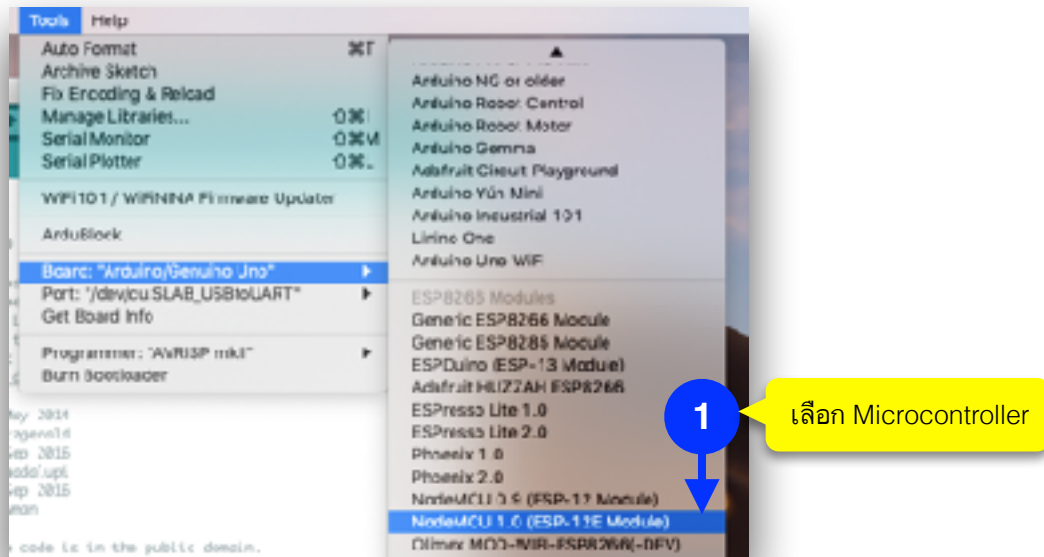


- ให้ต่อสาย USB กับ NodeMCU และเครื่องคอมพิวเตอร์ หลังจากนั้นก็ตรวจสอบชนิดของพอร์ต ซึ่งอาจจะเป็น COM3, COM4, หรืออื่นๆ

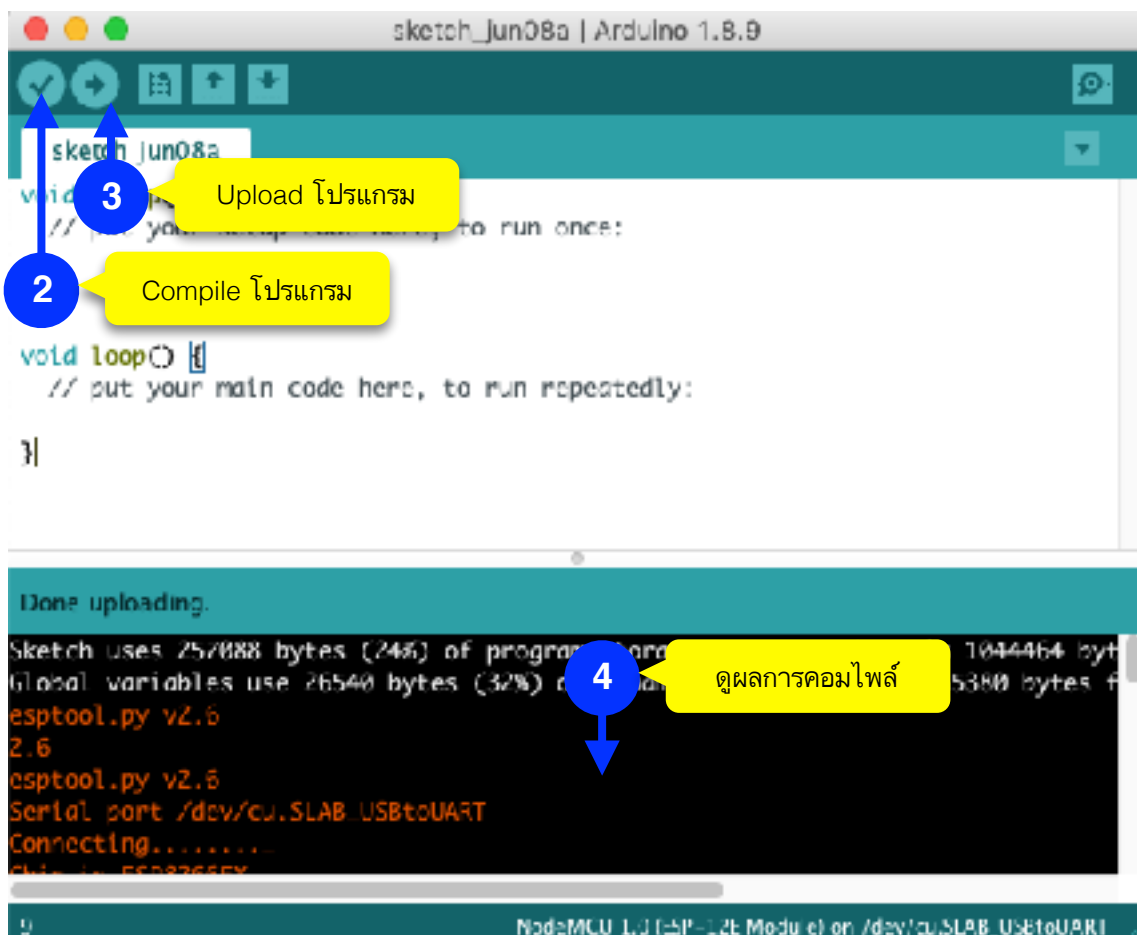


# การ Flash โปรแกรมเข้าบอร์ด

- เลือก Microcontroller เป็น NodeMCU 1.0



- คอมไพล์โปรแกรม > เขียนโปรแกรมลง node > คอยจนกว่าจะเขียนครบ 100%

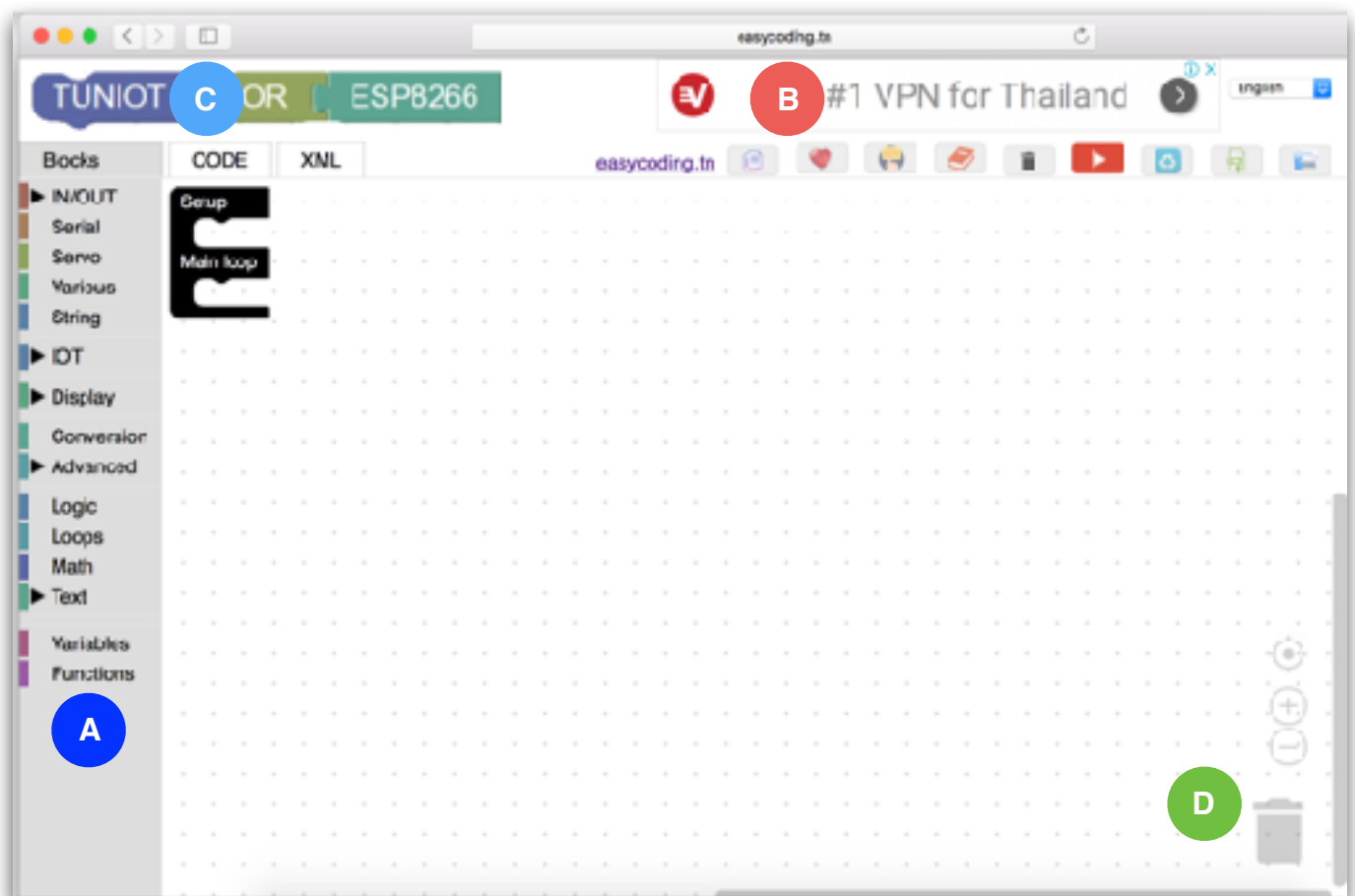




# การเขียนโปรแกรมด้วย BlockCode

- สร้าง Block Code ด้วยเว็บไซต์

<http://easycoding.tn/tuniot/demos/code/>



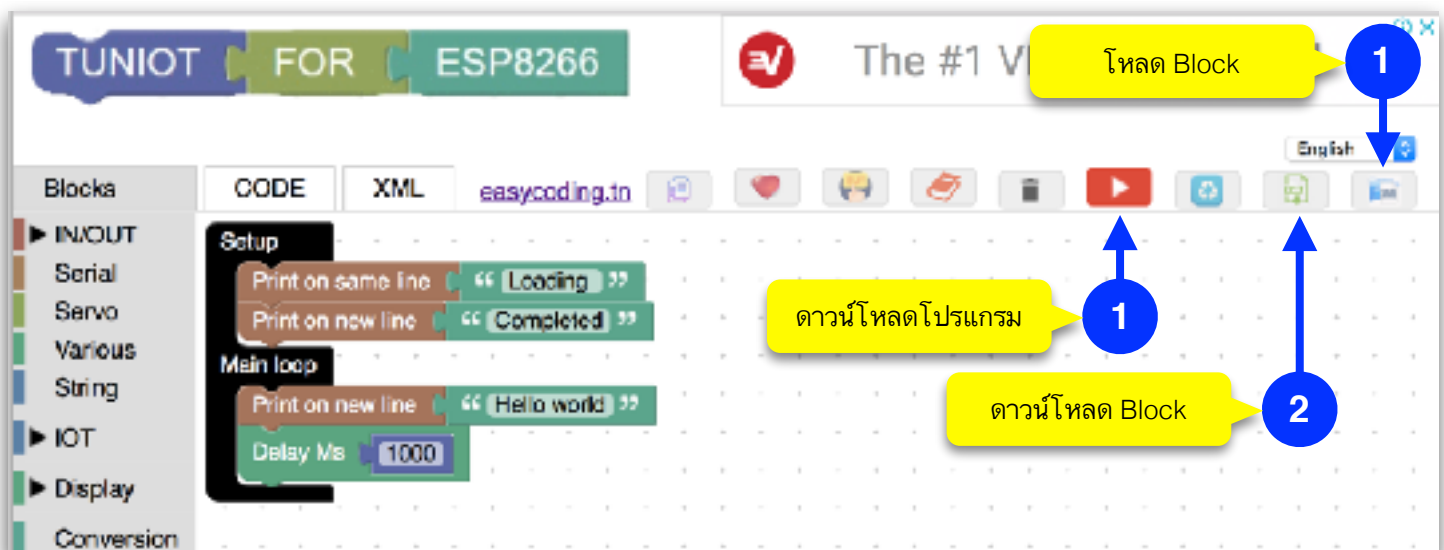
- A ส่วนของ Block Code เป็นการสร้าง code อัตโนมัติ
- B เมนูย่อย สำหรับลบ, ดาวน์โหลด Code, ดาวน์โหลด XML เป็นต้น
- C เปลี่ยน workspace เป็น Block, Code และ XML
- D ลบ Block บางตัว หรือซุมหน้าจอ

# Ex1- Hello World

- โปรแกรมแรกไม่มีการเชื่อมต่อวงจร
- สร้างโปรแกรม Hello World ด้วย Block Code

เมนูหลัก	สัญลักษณ์	หน้าที่
Serial	Print on same line "Loading"	แสดงข้อความแล้วไม่ขึ้นบรรทัดใหม่
Serial	Print on new line "Completed"	แสดงข้อความแล้วขึ้นบรรทัดใหม่
Various	Delay Ms 1000	หน่วงเวลา 1000 ms

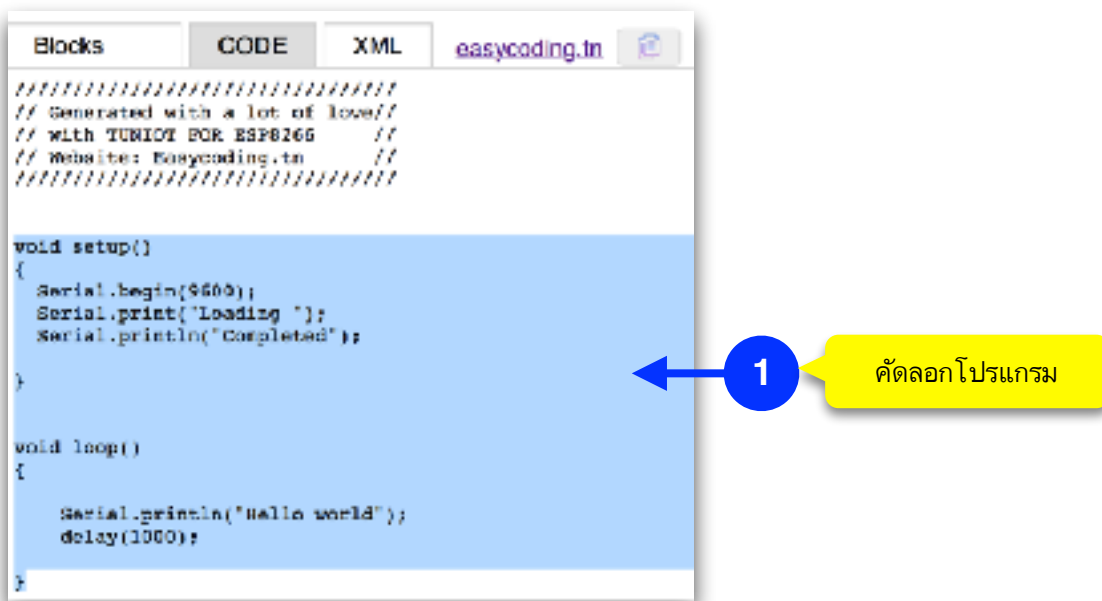
- วาง Block ตามดังภาพ



- 1 ดาวน์โหลดโปรแกรมเพื่อคอมไพล์ด้วยโปรแกรม Arduino
- 2 ดาวน์โหลด Block เพื่อเก็บไว้
- 3 โหลด Block ที่บันทึกไว้

# Ex1- Hello World

- การคอมไพล์โปรแกรม ทำได้ 2 วิธี
  - ดาวน์โหลดไฟล์โปรแกรม > คอมไพล์ด้วย Arduino IDE
  - เปิดหน้าจอ Code > คัดลอกโปรแกรม > วางในโปรแกรม

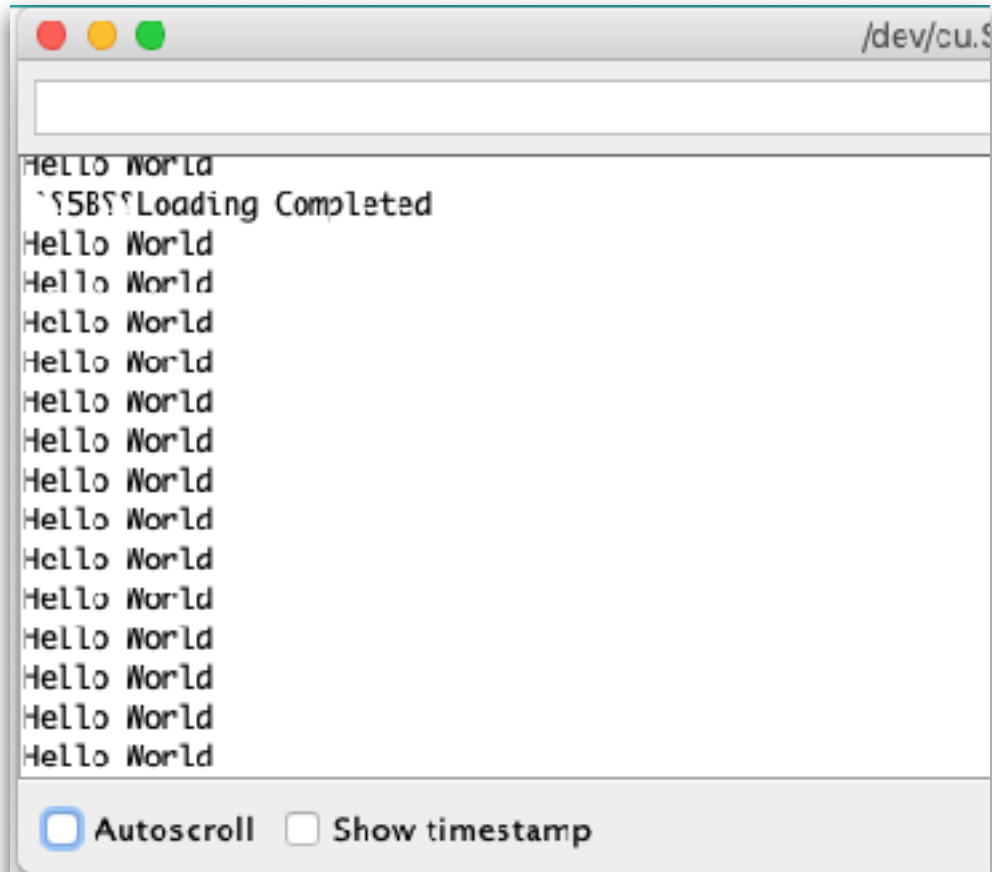


- คอมไพล์โปรแกรม



# Ex1- Hello World

- ผลการรันโปรแกรม



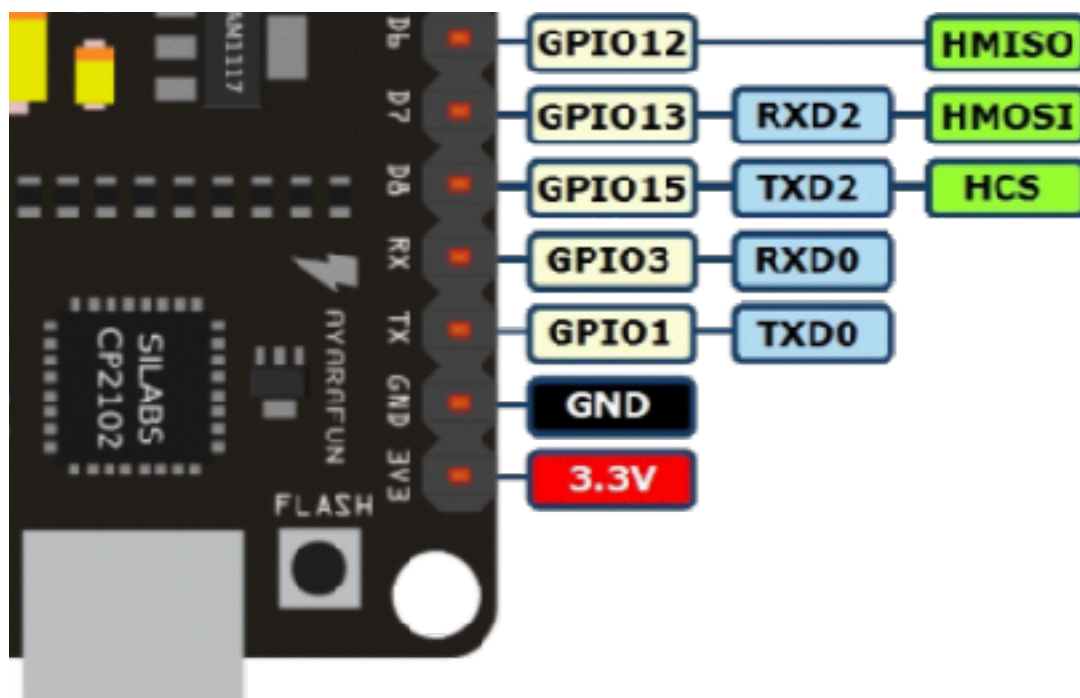
```
/dev/cu.S  
Hello World  
`5B? Loading Completed  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World  
Hello World
```

☒ Autoscroll ☐ Show timestamp







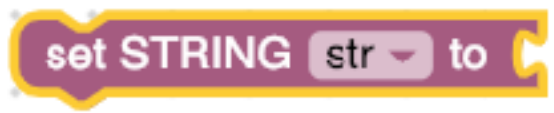


# Serial Port

- Node MCU มีพอร์ตสำหรับการสื่อสารแบบอนุกรม (Serial) จำนวน 3 พอร์ต
  - ผ่าน USB
  - ผ่าน PIN TX/RX
- การใช้งานต้องกำหนดความเร็วในการสื่อสาร
  - 9600bps, ....
- อ่านหรือเขียนข้อมูลผ่านพอร์ต
  - `Serial.println()`, `Serial.readln()`



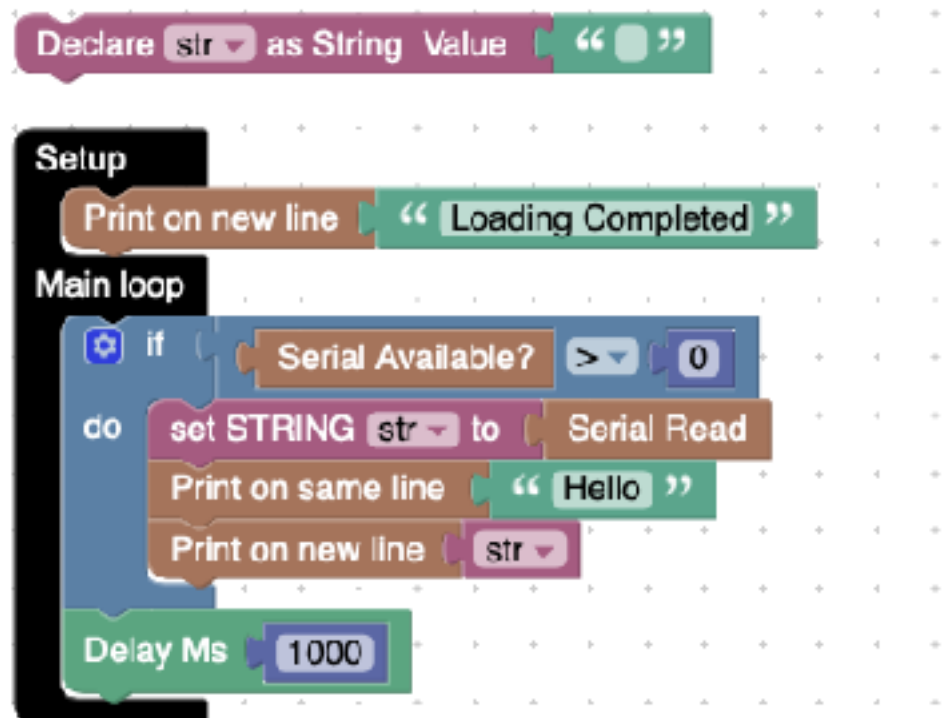
## Ex2- Hello Me

- โปรแกรมนี้ ไม่มีการต่อวงจร
- การใช้งาน Block Code

เมนูหลัก	สัญลักษณ์	หน้าที่
String		ประกาศตัวแปร ชื่อ str เพื่อเก็บข้อความ
Text		กำหนดค่า ให้แก่ String
Various		หน่วงเวลา 1000 ms
Logic		การเปรียบเทียบระหว่างซ้ายและขวา
Logic		กำหนดเงื่อนไข ถ้า....แล้ว
Math		กำหนดค่าตัวเลขให้ตัวแปร
String		กำหนดค่าให้ตัวแปร str แบบ String
Text		กำหนดค่าเริ่มต้น ให้แก่ข้อความ
Serial		ตรวจสอบสถานะของ Serial Port หากมากกว่า 0 ถือว่าพร้อมอ่านข้อมูล
Serial		อ่านค่าจาก Serial Port ครั้ง 1 ตัวอักษร

# Ex2- Hello Me

- วาง Block Code



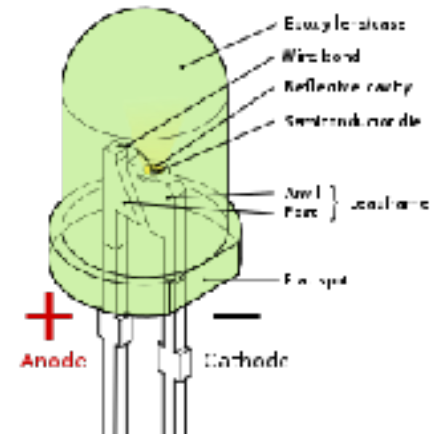
- คัดลอกโปรแกรม และแก้ไขโปรแกรม

```
String str;
void setup()
{
  str = "";
  Serial.begin(9600);
  Serial.println("Loading Completed");
}
void loop()
{
  if (Serial.available() > 0) {
    // str = Serial.read();
    // str = Serial.readString();
    str = Serial.readStringUntil('\n');
    Serial.print("Hello ");
    Serial.println(str);
  }
  delay(1000);
}
```

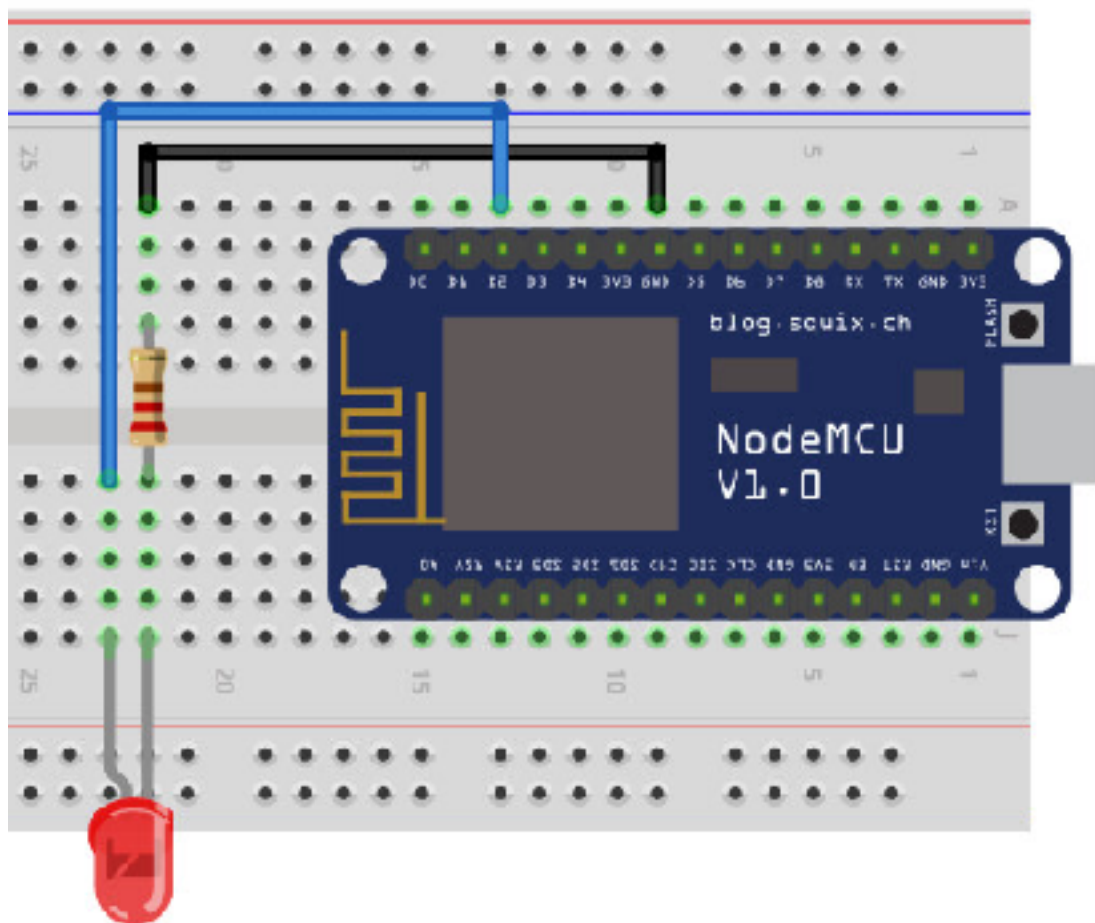
- คอมไพล์โปรแกรม และสังเกตผล

# LED

- โครงสร้างของ LED



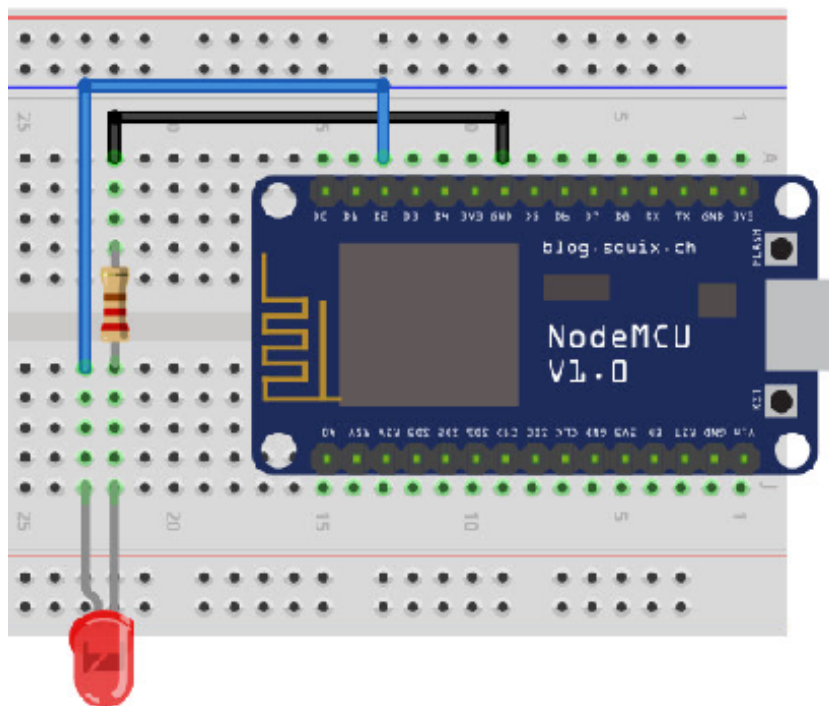
- การเชื่อมต่อ LED ต้องเชื่อมต่อตัวต้านทาน ทางขั้วบวก





# LED

- ต่อดวงจรดึงภาพ

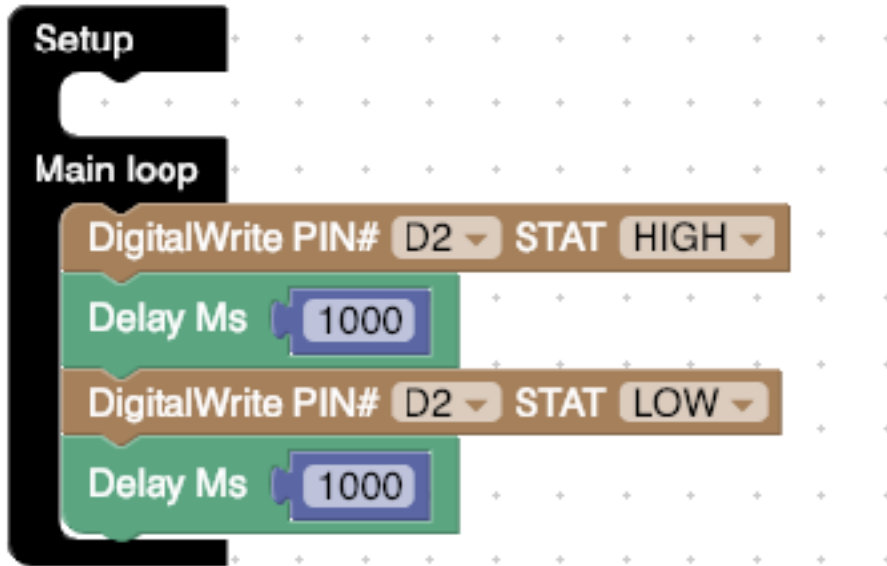


- การใช้งาน Block Code

เมนูหลัก	สัญลักษณ์	หน้าที่
IN/OUT - Digital	DigitalWrite PIN# D0 STAT LOW	เขียนสถานะไฟไปยังพินที่กำหนด
Various	Delay Ms 1000	หน่วงเวลา 1000 ms

# Ex3 - LED Blink

- วาง Block Code



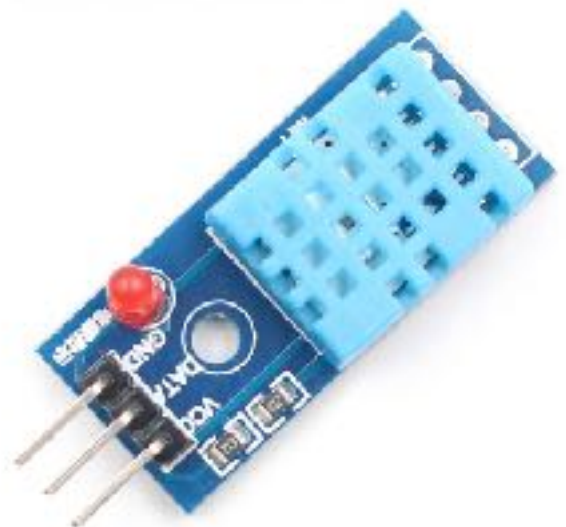
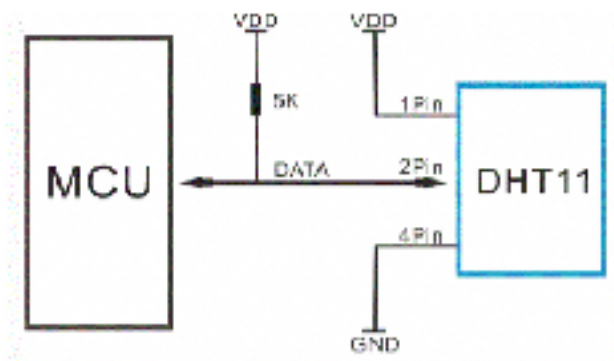
- คัดลอกโปรแกรม คอมไพล์โปรแกรม และสังเกตผล

```
void setup()
{
  pinMode(16, OUTPUT);
}

void loop()
{
  digitalWrite(16, HIGH);
  delay(1000);
  digitalWrite(16, LOW);
  delay(1000);
}
```

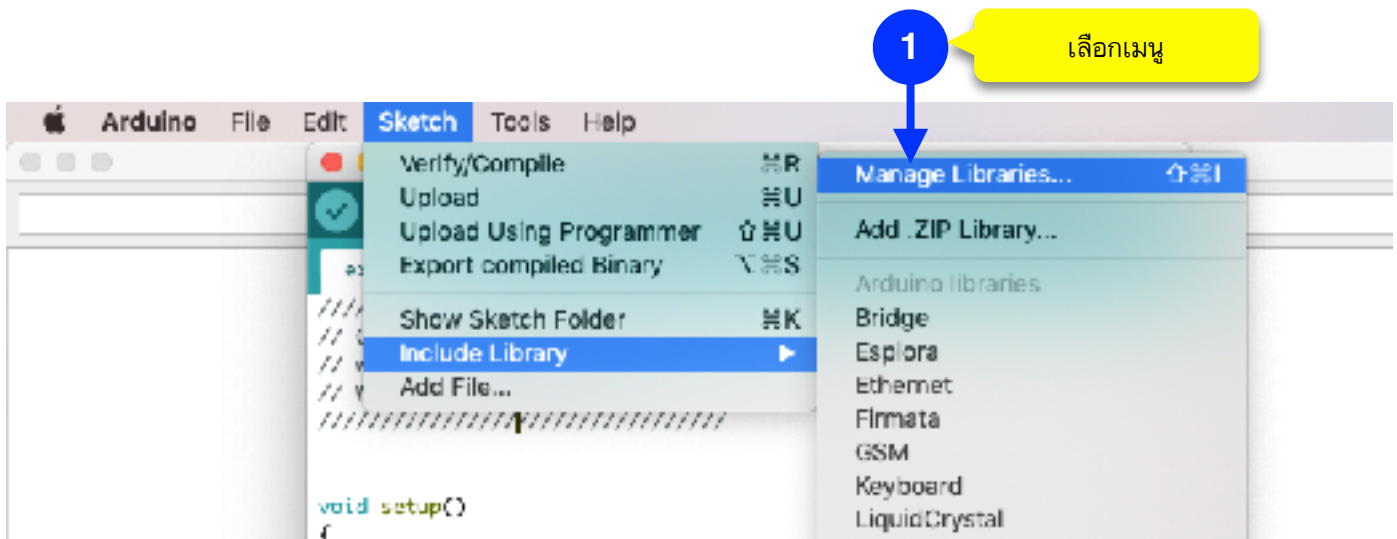
# เซนเซอร์วัดความชื้นและอุณหภูมิ

- DHT11: Digital Temperature and Humidity Sensor
  - โมดูลวัดอุณหภูมิ (Temperature) และความชื้นสัมพัทธ์ (Humidity) โดยใช้ชิพ DHT11
  - ให้ Output ออกมาเป็นแบบดิจิทัล
  - ใช้ไฟ DC ขนาด 3.5 – 5.5 โวลต์
  - เหมาะสำหรับวัดอุณหภูมิ (Temperature) ในช่วง 0-60° c (+/- 2%)
  - ความชื้นสัมพัทธ์ (Humidity) ในช่วง 20-90 % RH. (+/- 5%)
  - สามารถต่อกับบอร์ด Arduino ใช้งานได้ทันที

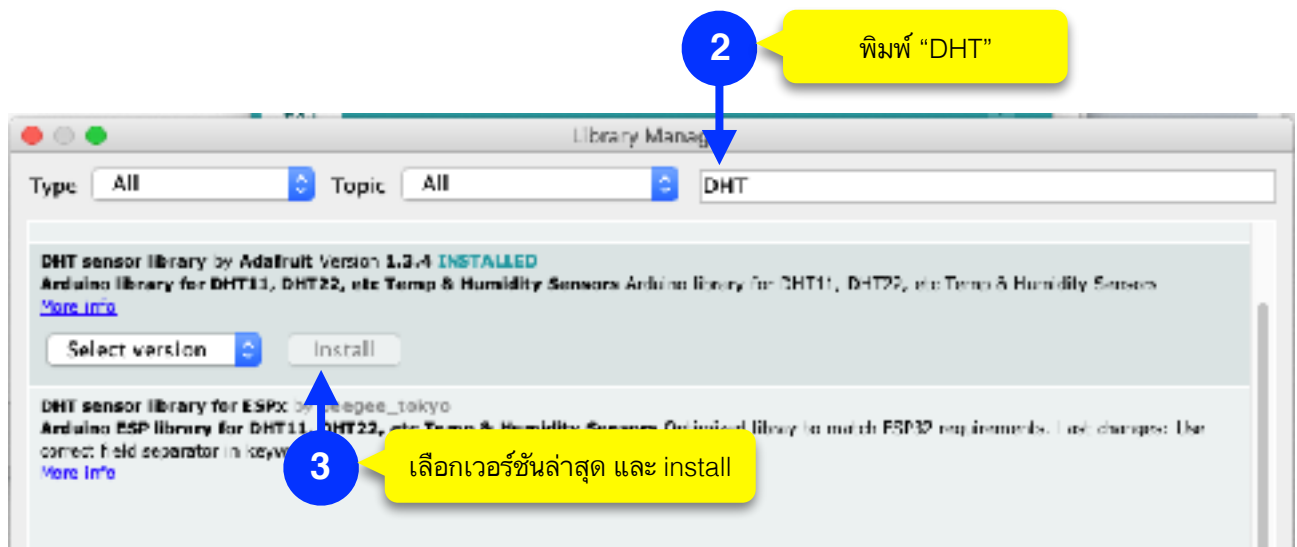


# เซนเซอร์วัดความชื้นและอุณหภูมิ

- ต้องเพิ่มไลบรารี DHT11 ก่อนพัฒนาโปรแกรม



- พิมพ์ “DHT” เพื่อค้นหา หลังจากนั้น ติดตั้งไลบรารี

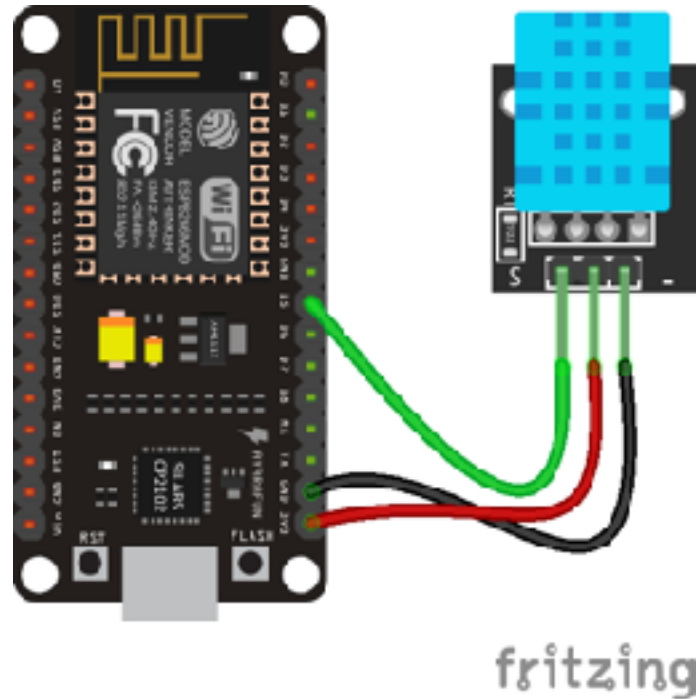




# Ex4 - Temperature/Humidity Sensor

- ต่อดังภาพ

Arduino	Module
3.3V	VCC
GND	GND
D5	DATA



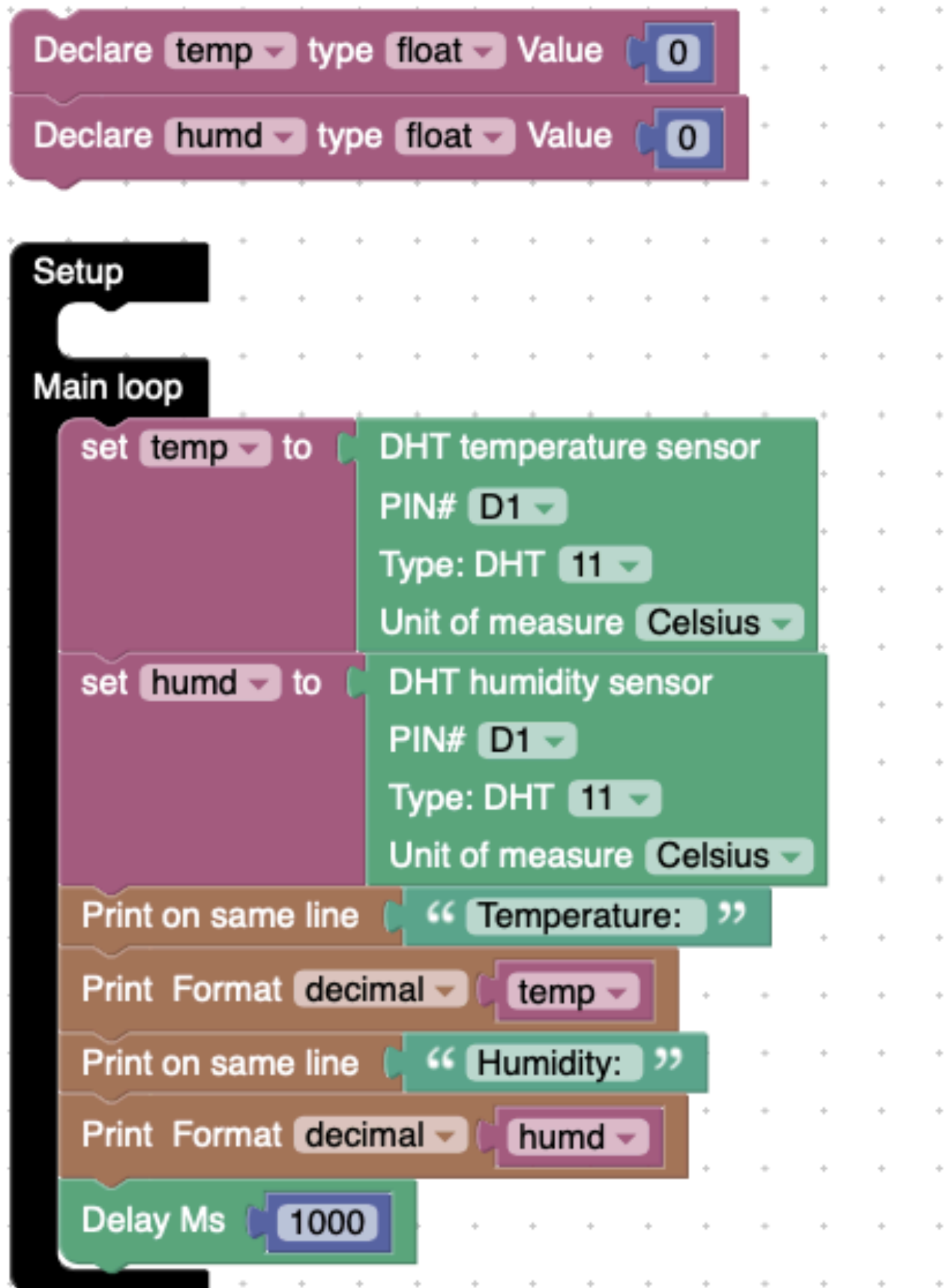
# Ex4 - Temperature/Humidity Sensor

- การใช้งาน Block Code

เมนูหลัก	สัญลักษณ์	หน้าที่
Variable		ประกาศตัวแปรชื่อ i ชนิดข้อมูลเป็น long
Math		กำหนดค่าเริ่มต้นจำนวนเต็ม
Variable		กำหนดค่าให้แก่ตัวแปร i
Various		อ่านข้อมูลจากเซนเซอร์ DHT11 อ่านค่าอุณหภูมิซึ่งต่อพิน D0
Various		อ่านข้อมูลจากเซนเซอร์ DHT11 อ่านค่าความชื้นซึ่งต่อพิน D0
Serial		แสดงผลทางจอ console
Serial		แสดงผลทางจอ console แบบทศนิยม

# Ex4 - Temperature/Humidity Sensor

- วาง Block Code



# Ex4 - Temperature/Humidity Sensor

- คัดลอกโปรแกรม วางในโปรแกรม Arduino IDE และคอมไพล์โปรแกรม

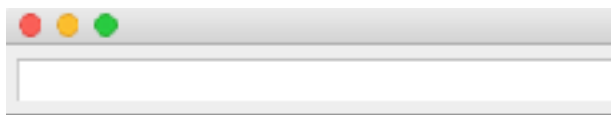
```
#include "DHT.h"

float temp;
float humd;
DHT dht14(14,DHT11);

void setup()
{
    temp = 0;
    humd = 0;
    Serial.begin(9600);
}

void loop()
{
    humd = (dht14.readHumidity( ));
    temp = (dht14.readTemperature( ));
    Serial.print("Temperature : ");
    Serial.println(temp);
    Serial.print("Humidity : ");
    Serial.println(humd);
    delay(1000);
}
```

- ทดสอบโปรแกรม



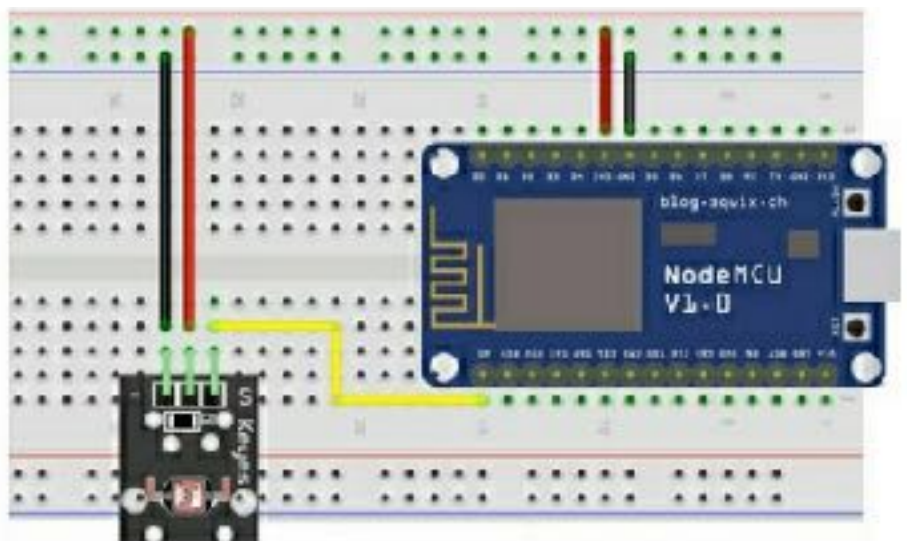
```
Humidity : 56.00
Temperature : 28.70
Humidity : 55.00
Temperature : 28.70
Humidity : 55.00
Temperature : 28.70
Humidity : 55.00
Temperature : 28.70
Humidity : 55.00
Temperature : 28.70
Humidity : 54.00
Temperature : 28.70
Humidity : 54.00
```

# Light Dependent Resistor - LDR

- LDR คือ ตัวต้านปรับค่าได้ตามแสง ซึ่งความต้านทานที่เปลี่ยนไปทำให้โวลต์ที่ไหลผ่านตัวต้านทานเปลี่ยนไปด้วย เราสามารถวัดโวลต์และอ่านค่าเป็นความสว่างของแสงจาก LDR ได้
- สัญญาณที่ได้เป็นแบบ Analog ดังนั้นจึงทำได้โดยผ่านทางขา A0 ของ NodeMCU โดยมีค่าระหว่าง 0 - 1023



- การต่อวงจร



Arduino	Module
3.3V	VCC
GND	GND
A0	S



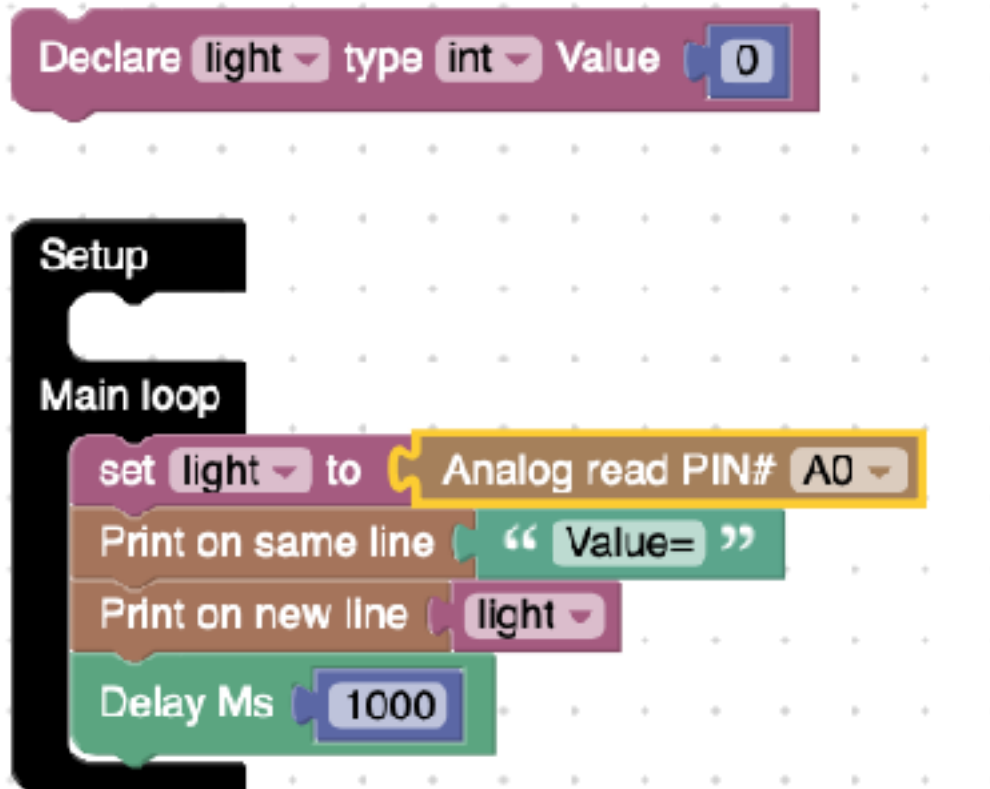
# Ex5 - LDR Sensor

- การใช้งาน Block Code

เมนูหลัก	สัญลักษณ์	หน้าที่
Variable		ประกาศตัวแปรชื่อ light ชนิดข้อมูลเป็น int
Math		กำหนดค่าเริ่มต้นจำนวนเต็ม
Variable		กำหนดค่าให้แก่ตัวแปร light
Various		อ่านข้อมูลจากเซนเซอร์ แอนาล็อก
Variable		ตัวแปรชื่อ light
Serial		แสดงผลทางจอ console
Various		หน่วงเวลา 1000 ms

# Ex5 - LDR Sensor

- การเชื่อมต่อ Block Code



- ส่วนของโปรแกรม

```
int light;
void setup()
{
  light = 0;
  Serial.begin(9600);
}
void loop()
{
  light = analogRead(A0);
  Serial.print("Value =");
  Serial.println(light);
  delay(1000);
}
```

# Exercise

- ทดสอบโปรแกรม
  - เมื่อแสงน้อย ค่า สูง หรือ ต่ำ
- ผู้เข้าอบรม ทดลอง
  - หาก แสงน้อย ให้ LED ติด
  - หาก แสงมาก ให้ LED ดับ

# Ex6- Exercise Solution

- Block Code

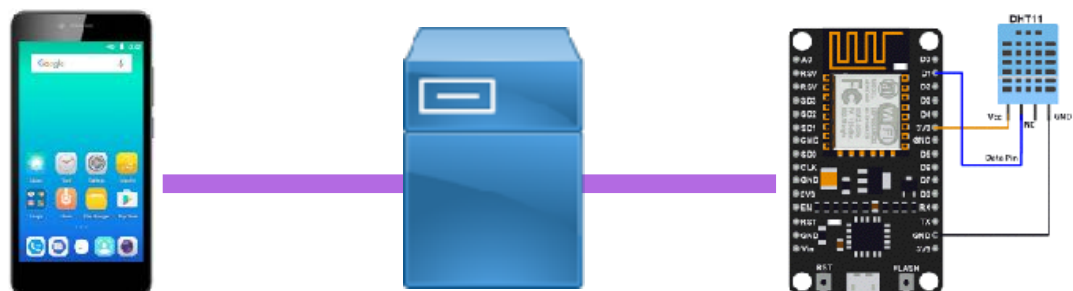
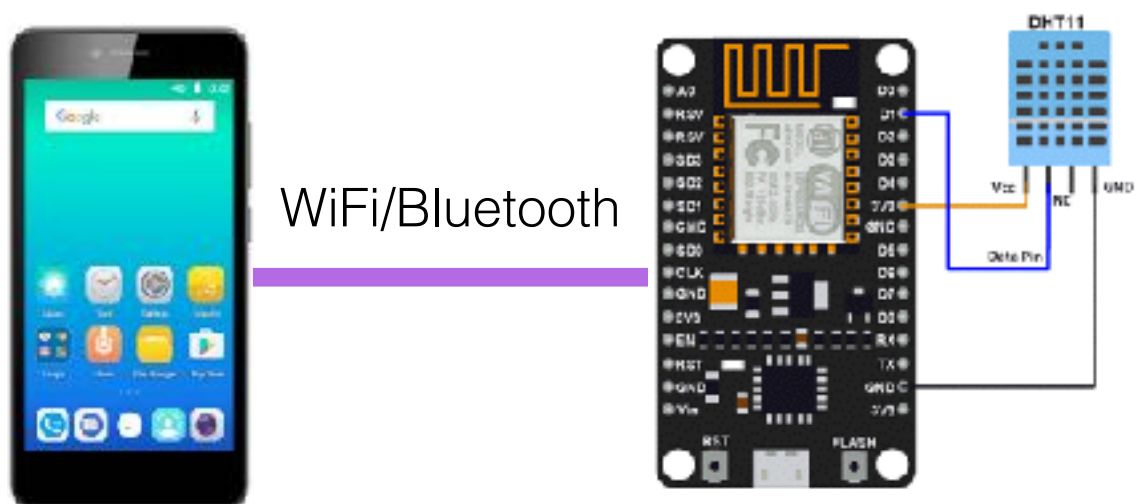


```
float light;
void setup()
{
    light = 0;
    pinMode(16, OUTPUT);
    Serial.begin(9600);
}

void loop()
{
    light = analogRead(A0);
    if (light > 512) {
        digitalWrite(16,HIGH);
    } else {
        digitalWrite(16,LOW);
    }
    Serial.print("Value = ");
    Serial.println(light);
    delay(1000);
}
```

# สถาปัตยกรรมเครือข่ายสำหรับ IoT

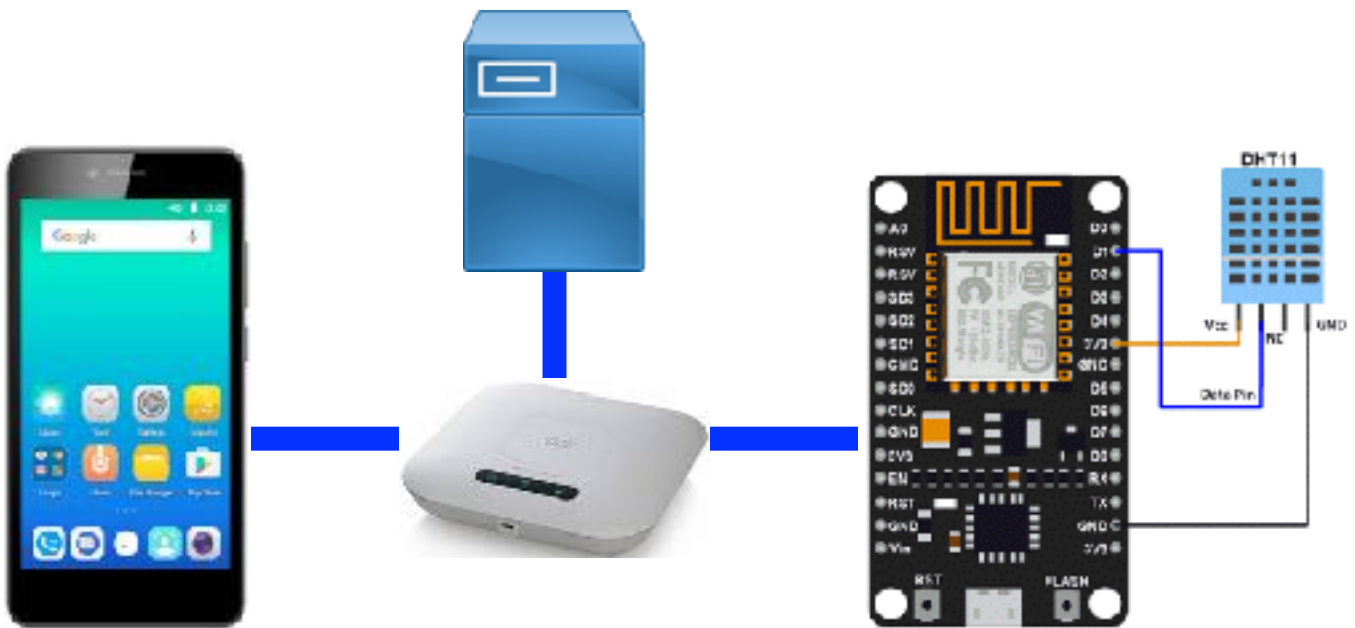
- การใช้งานระบบ IoT เพื่อการเฝ้าระวังและควบคุมการทำงาน ต้องการการเชื่อมต่อผ่านเครือข่าย อาจจะเป็น WiFi หรือ Bluetooth
- สถาปัตยกรรมเครือข่าย มีหลายรูปแบบ ได้แก่
  - Node - Mobile
  - Node - Server - Mobile





# NodeMCU - Monitoring

- เป็นการออกแบบระบบอย่างง่าย โดยให้ NodeMCU แจกจ่าย IP ผ่าน WiFi ให้แก่มือถือ หลังจากนั้น การควบคุมการทำงานทั้งหมด สามารถสั่งผ่านมือถือได้



- ขั้นตอนการโปรแกรม
  - ให้ node เชื่อมต่อ Access Point
  - อ่านข้อมูลจากเซนเซอร์ ส่งไปเก็บไว้ที่ Cloud
  - ให้มือถือเชื่อมต่อ Access Point และ อ่านข้อมูลจาก Cloud

# ลงทะเบียนใช้งาน Thingspeak

- เข้าเว็บไซต์ <http://www.thingspeak.com> เพื่อลงทะเบียน



- ป้อนข้อมูลเพื่อลงทะเบียน

Create MathWorks Account

**Email Address**

**Location**

**First Name**

**Last Name**

**Continue**

**Cancel**

Already have a ThingSpeak account?  
[Sign In](#)

Annotations:

- 2: อีเมลล์สำหรับสมัคร (Email for registration)
- 3: เลือกประเทศ Thailand (Select country Thailand)
- 4: ชื่อ (First Name)
- 5: นามสกุล (Last Name)
- 6: ยืนยันการสมัคร (Confirm registration)

# ลงทะเบียนใช้งาน Thingspeak

- เข้าเว็บไซต์ <http://www.thingspeak.com> เพื่อลงทะเบียน

## Personal Email Detected

To use your organization's MATLAB, enter your work or university email

### Email Address

ejundang@gmail.com

☒ Use this email for my MathWorks Account

Continue

Cancel

7

ยืนยันการเลือกใช้อีเมลนี้

8

ยืนยันการสมัคร

- เช็กเมลล์ และ กด Verify your email

**MathWorks**

**Important MathWorks Account Information**

**Thank you for registering with MathWorks!**

Verify your email address by clicking this link:

**Verify your email**

If you are unable to click the link, copy and paste this link into the address bar on your browser.

<https://www.mathworks.com/mwaccount/widgets/embedded/profiles/verify/trabSSv?email=chanchanan.ja%40mail.wu.ac.th>

Sincerely,  
MathWorks Customer Service Team

**Verify Your MathWorks Account**

To finish creating your account, complete the following steps:

1. Go to your inbox for ejundang@gmail.com.
2. Click the link in the email we sent you.
3. Click **Continue**.

**Didn't receive the email?**

- Check your spam folder.
- Send me the email again.
- If you still have not received the email, Contact Customer Support.

9

ตรวจสอบตัวตนผ่านเมลล์

- เลือก Continue เพื่อไปต่อ

เลือกเพื่อไปต่อ

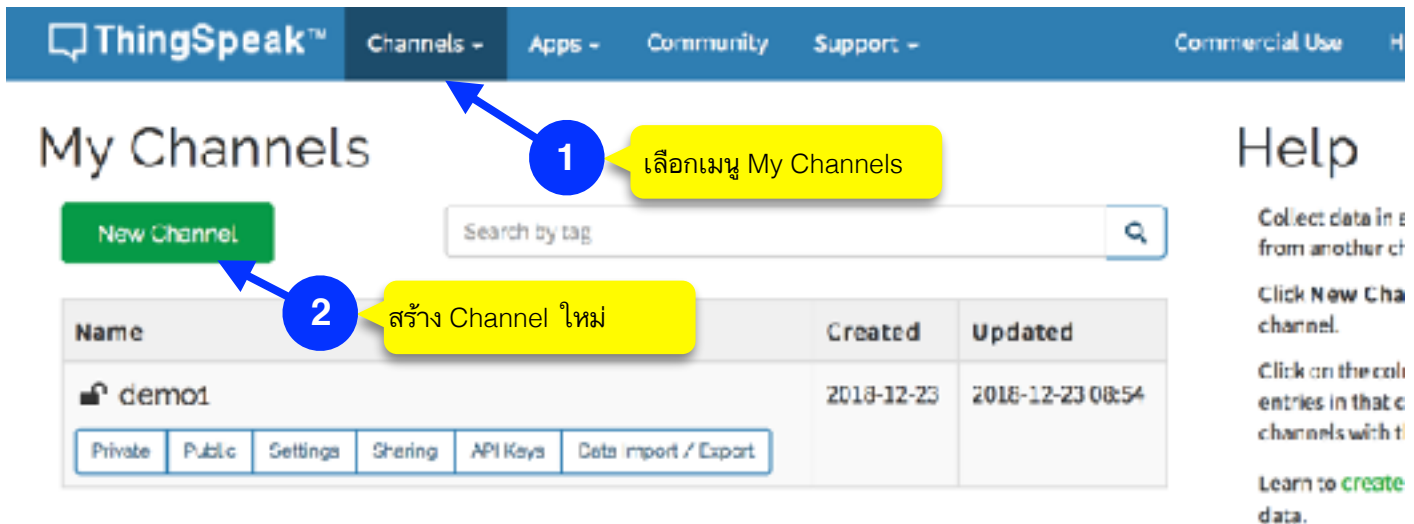
10

Continue

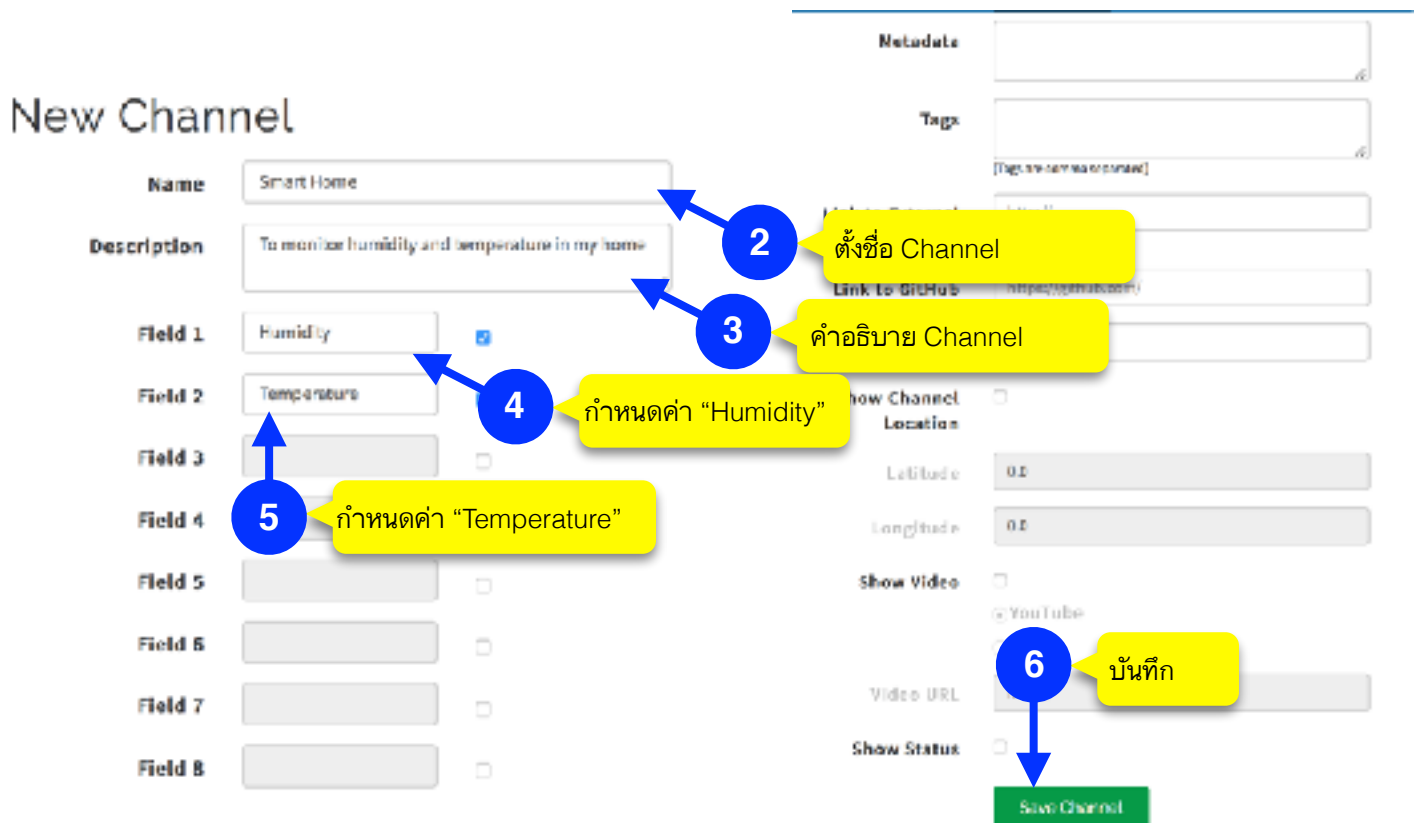
Cancel

# ใช้งาน Thingspeak

- หลังจากนั้น Login เข้าระบบ
- สร้าง Channel ใหม่



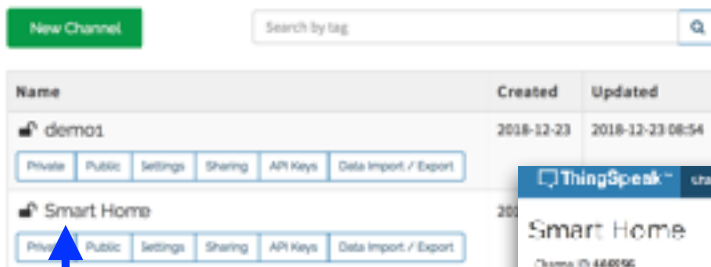
- สร้าง Channel เพื่อเก็บข้อมูล โดยกำหนดให้เก็บข้อมูล 2 ชนิด คือความชื้น และ อุณหภูมิ



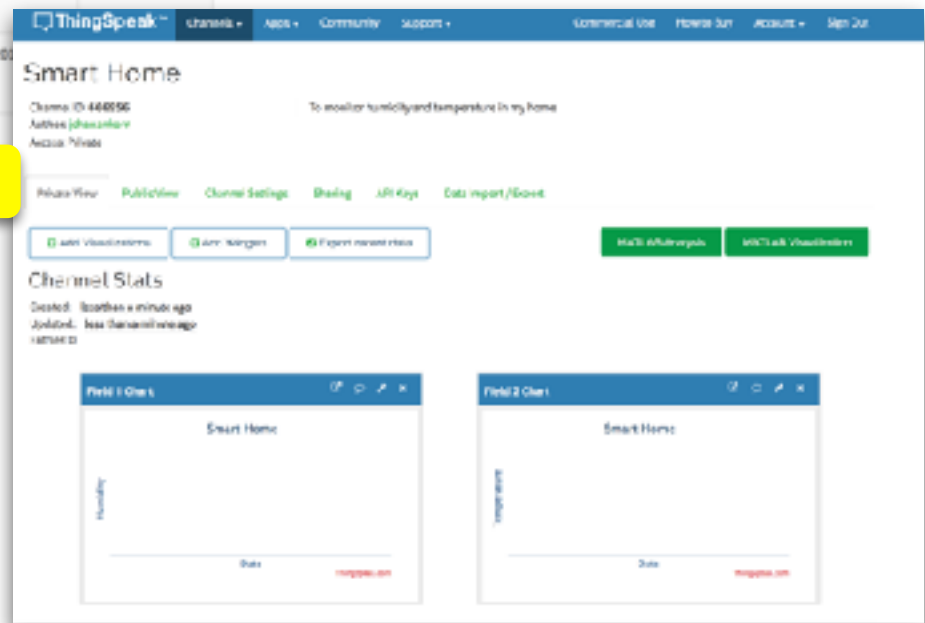
# การใช้งาน Thingspeak

- หน้า Dashboard ของ Channel

## My Channels



เลือกเพื่อเข้าหน้า Dashboard



- ปรับสิทธิ์การเข้าถึงเป็น Public

## Smart Home

Channel ID: 666956  
Author: jchanankorn  
Access: Private

To monitor humidity and temperature in my home

Private View Public View Channel Settings **Sharing** API Keys Data Import / Export

### Channel Sharing Settings

- ☐ Keep channel view private
- ☒ Share channel view with everyone
- Share channel view only with the following users:

เลือก Share channel view with everyone

เลือก Sharing

### Help

Thingspeak allows you to control who can view the data in your channel. Irrespective of settings on this tab, reading data from or writing data to the fields of a channel requires the appropriate API key for the channel.

### Channel Sharing Settings

- **Keep channel view private:** Selecting this option keeps your channel private. Only you will be able to see the channel view.
- **Share channel view with everyone:** Selecting this option makes the public view of your channel viewable by anyone browsing the ThingSpeak website.
- **Share channel view only with the following users:** Selecting this option shares the private view of your channel only with specific ThingSpeak users.

# การใช้งานใช้งาน Thingspeak

- แสดงผลข้อมูลในรูปแบบกราฟ

## Smart Home

[Watch](#) [Tweet](#) [Like](#) [Share](#)

Channel ID: 865556  
Author: drchanankorn  
Access: Public

**3**

เข้าสู่โหมด Public View

[Private View](#) [Public View](#) [Channel Settings](#) [Sharing](#) [API Keys](#) [Data Import / Export](#)[Add Visualization](#)[Add Widgets](#)[Export recent data](#)[MATLAB Analysis](#)[MATLAB Visualization](#)

## Channel Stats

Created: about a minute ago  
Updated: less than a minute ago  
Entries: 0





# การใช้งาน Thingspeak

- ข้อมูลสำหรับการเชื่อมต่อ Channel

## Smart Home

Channel ID: 666956

Author: jchanankorn

Access: Public

3

คัดลอก Channel ID

monitor humidity and tempe

เลือก API Keys

1

Private View

Public View

Channel Settings

Sharing

API Keys

Data

## Write API Key

Key

L7DAOIF7G9WBYYLF

คัดลอก API Keys

2

Generate New Write API Key

## Read API Keys

Key

LTJFM7ZHVU5XALF7

Note

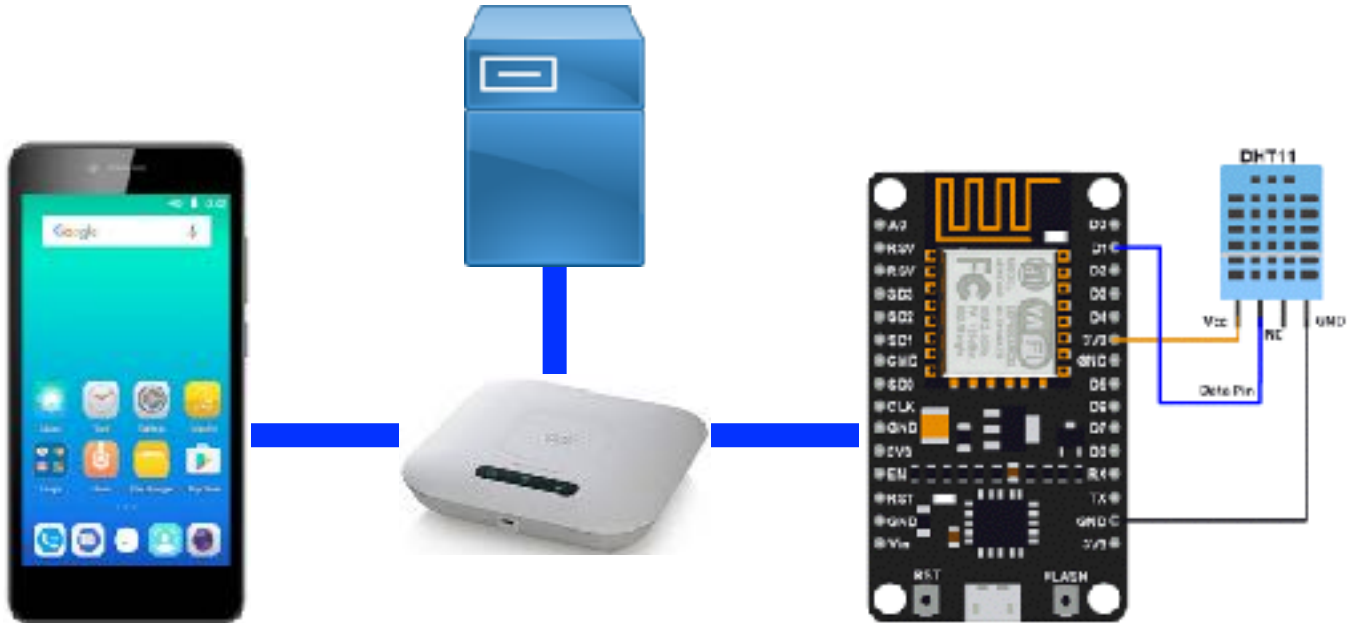
Save Note

Delete API Key

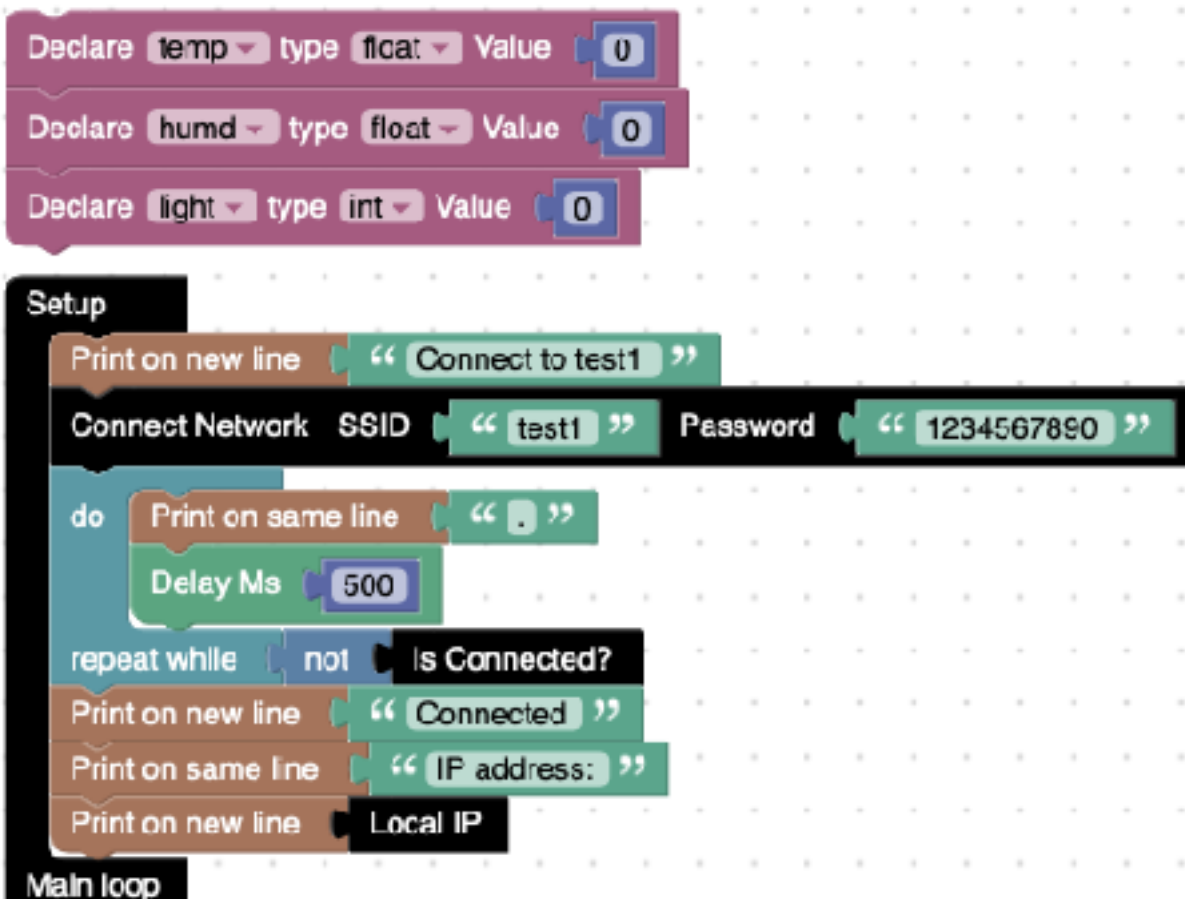
Generate New Read API Key

# NodeMCU to Thingspeak

- เชื่อมต่อวงจร

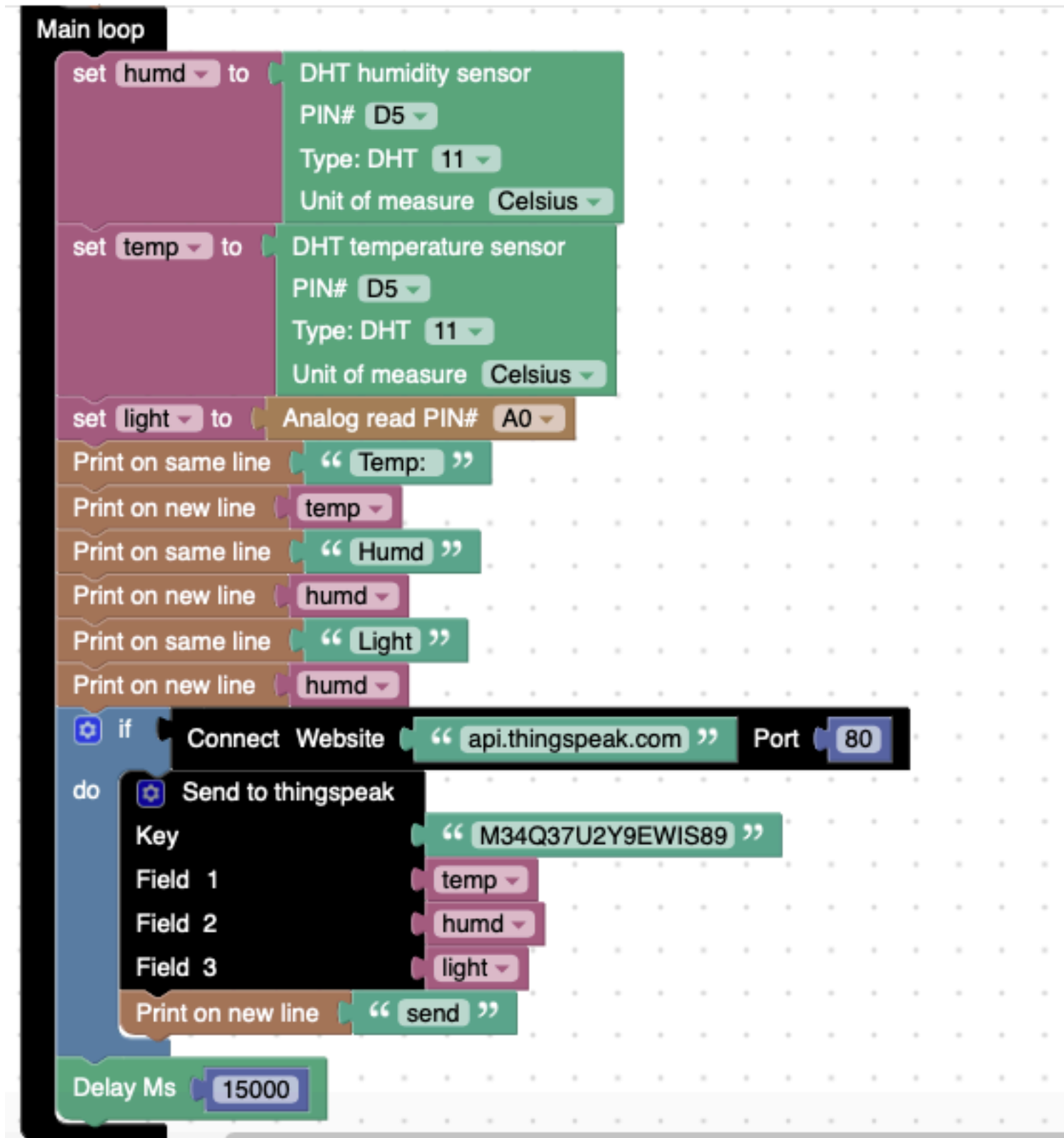


- การใช้งาน Block Code เพื่อเชื่อมต่อ WiFi



# NodeMCU to Thingspeak

- การใช้งาน Block Code เพื่อเชื่อมต่อ Thingspeak

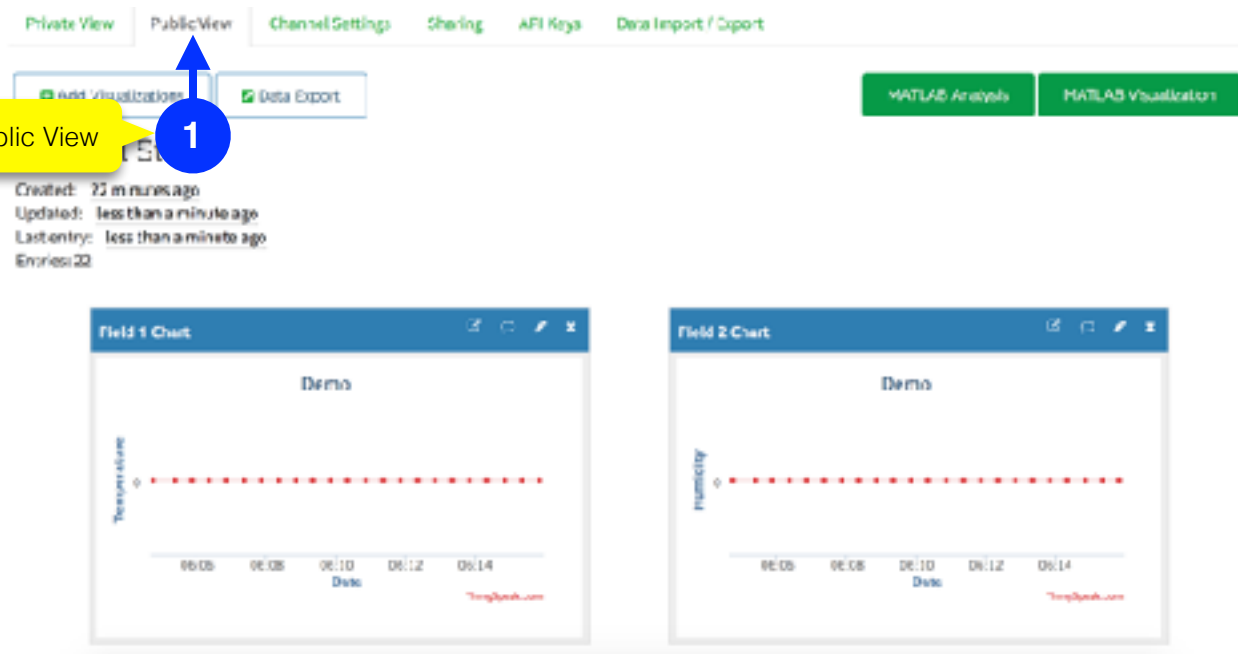


The image shows a Scratch-style block code for a NodeMCU microcontroller. The code is organized into a 'Main loop' structure. It starts with two 'set' blocks for 'humd' and 'temp' variables, both pointing to 'DHT humidity sensor' and 'DHT temperature sensor' respectively. Both sensors are configured with 'PIN# D5', 'Type: DHT 11', and 'Unit of measure Celsius'. The 'light' variable is set to 'Analog read PIN# A0'. The code then prints the values of 'temp', 'humd', and 'light' on the serial monitor, alternating between 'Print on same line' and 'Print on new line' for each variable. After printing, the code enters an 'if' block that checks if the 'Connect Website' is 'api.thingspeak.com' and the 'Port' is '80'. If true, it enters a 'do' block where it sends data to Thingspeak. The 'Send to thingspeak' block is configured with 'Key' 'M34Q37U2Y9EWIS89', 'Field 1' 'temp', 'Field 2' 'humd', and 'Field 3' 'light'. It also includes a 'Print on new line' block with the text 'send'. Finally, the code ends with a 'Delay Ms' block set to '15000'.

```
Main loop
  set humd to DHT humidity sensor
  PIN# D5
  Type: DHT 11
  Unit of measure Celsius
  set temp to DHT temperature sensor
  PIN# D5
  Type: DHT 11
  Unit of measure Celsius
  set light to Analog read PIN# A0
  Print on same line "Temp: "
  Print on new line temp
  Print on same line "Humd "
  Print on new line humd
  Print on same line "Light "
  Print on new line humd
  if Connect Website "api.thingspeak.com" Port 80
  do
    Send to thingspeak
    Key "M34Q37U2Y9EWIS89"
    Field 1 temp
    Field 2 humd
    Field 3 light
    Print on new line "send"
  Delay Ms 15000
```

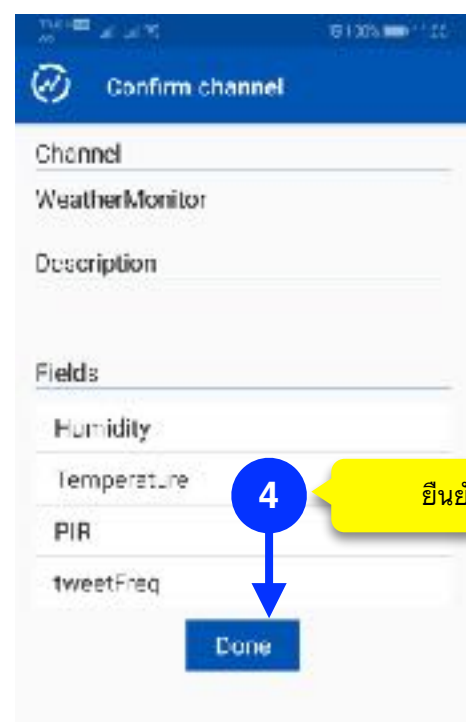
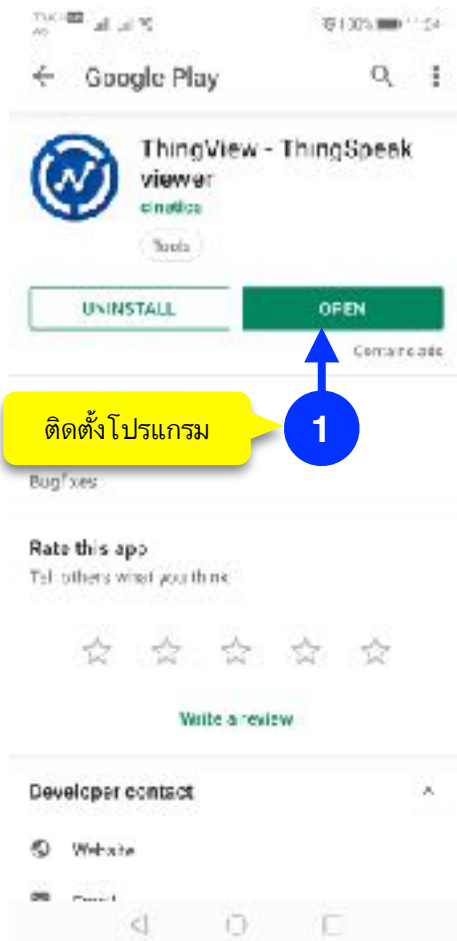
# NodeMCU to Thingspeak

- ดูผลการทำงานผ่าน Thingspeak



# NodeMCU to Thingspeak

- การแสดงผลผ่านโปรแกรมบนมือถือ ซึ่งโปรแกรม Thingspeak มีโปรแกรมเชื่อม Thingsview ให้ดาวน์โหลดและติดตั้งโปรแกรม



# NodeMCU to Thingspeak

- การแสดงผลผ่านโปรแกรมบนมือถือ ซึ่งโปรแกรม Thingspeak มีโปรแกรมเชื่อม Thingsview ให้ดาวน์โหลดและติดตั้งโปรแกรม

