

การฝึกอบรมหลักสูตร

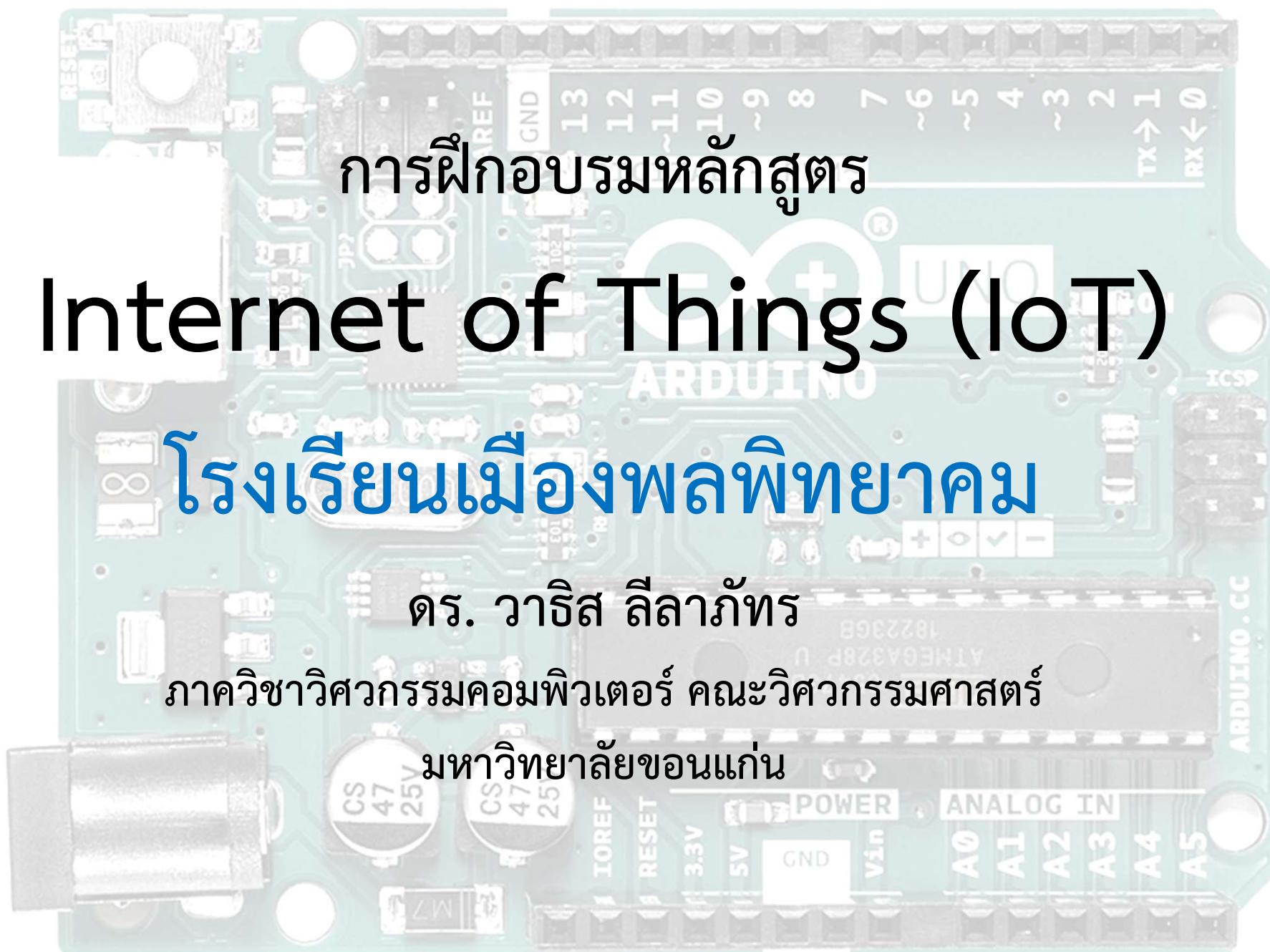
# Internet of Things (IoT)

## โรงเรียนเมืองพลพิทยาคม

ดร. วาริส ลีลาภัทร

ภาควิชาวิศวกรรมคอมพิวเตอร์ คณะวิศวกรรมศาสตร์

มหาวิทยาลัยขอนแก่น



# IoT คืออะไร?



IoT (Internet of Things) อินเตอร์เน็ตของสรรพสิ่ง เป็นระบบประกอบด้วย อุปกรณ์ที่สามารถสื่อสารข้อมูลผ่านอินเตอร์เน็ตได้ โดยอุปกรณ์เหล่านี้จะ ประกอบด้วย

- \* ตัวตรวจวัดหรือเซ็นเซอร์
- \* โปรแกรมหรือซอฟต์แวร์ควบคุมการทำงาน

# ตัวอย่างอุปกรณ์ IoT



# ตัวอย่างอุปกรณ์ IoT

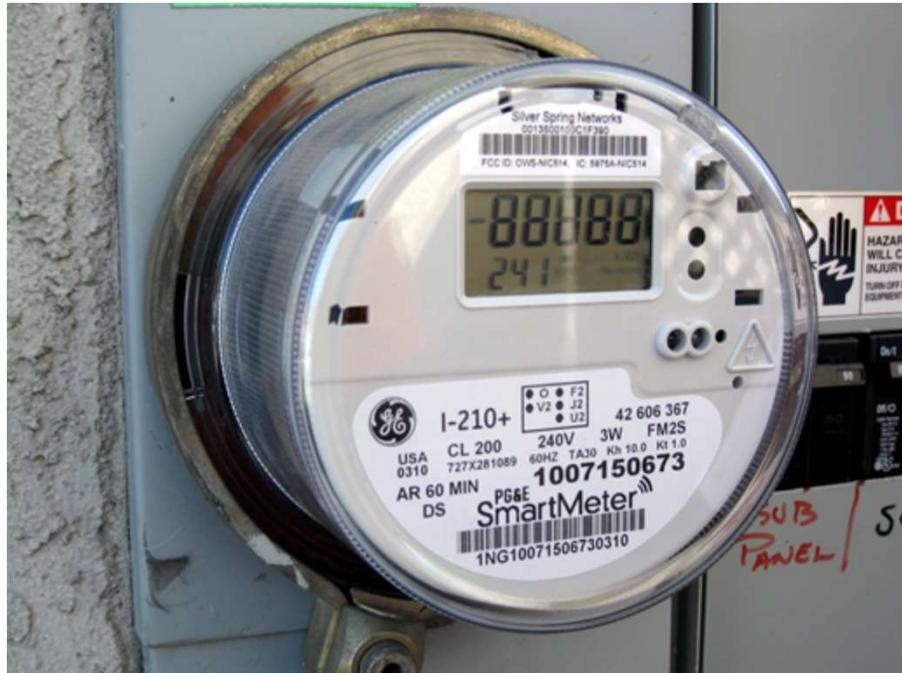


Smart washer



Smart refrigerator

# ตัวอย่างอุปกรณ์ IoT

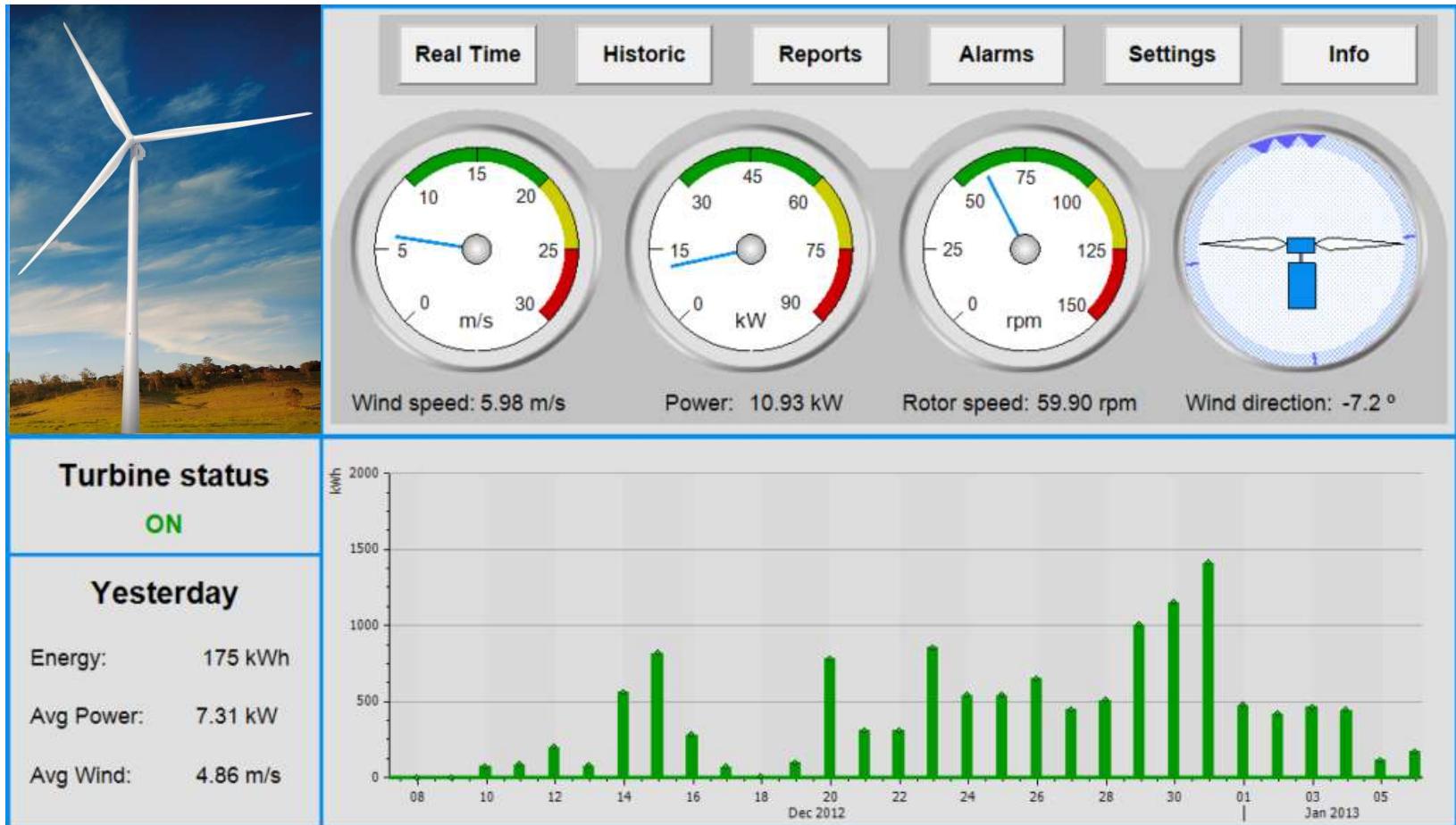


Smart energy meter



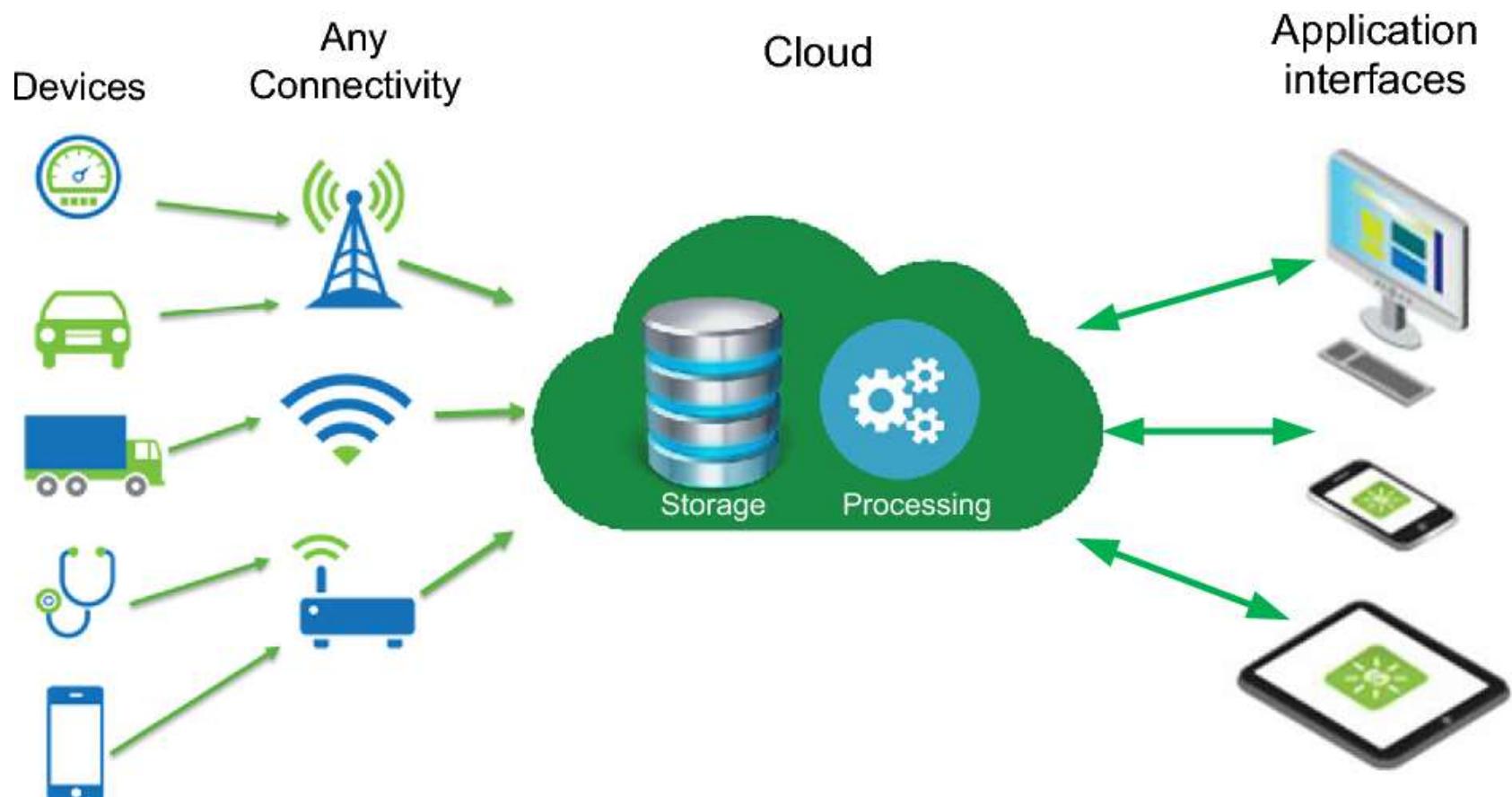
Smart water meter

# ตัวอย่างอุปกรณ์ IoT



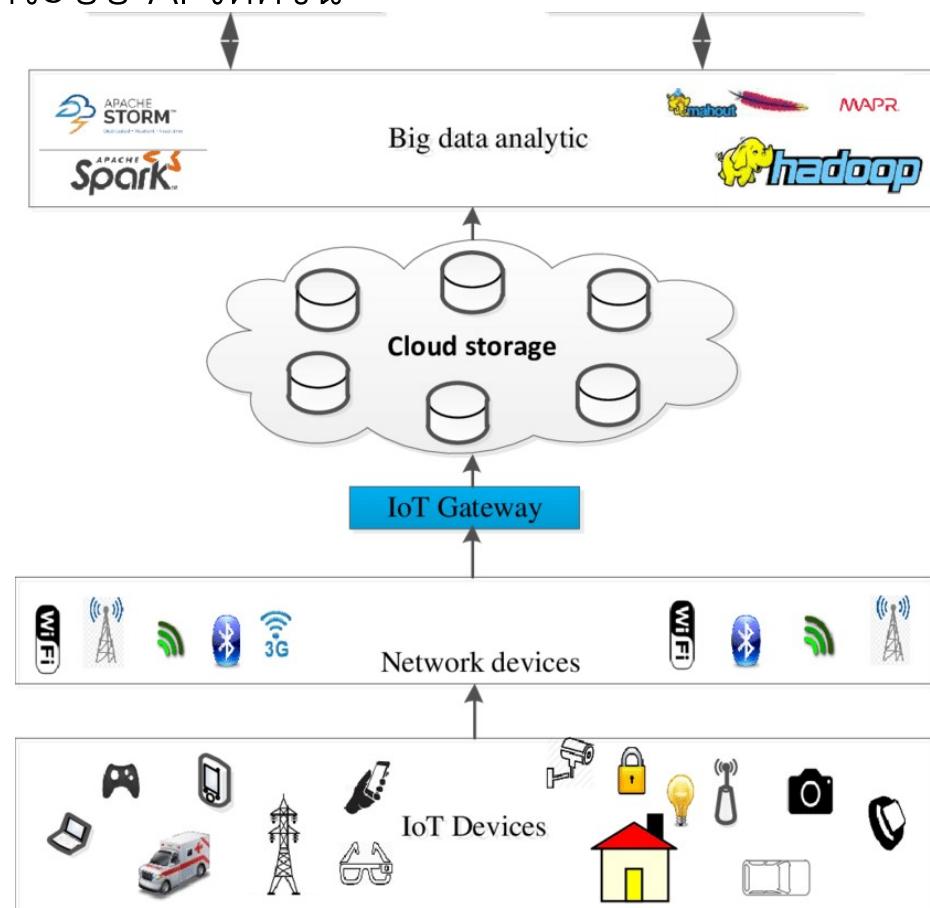
Wind turbine monitor

# ส่วนประกอบของระบบ IoT



# ประโยชน์ของ IoT

- ทำให้อุปกรณ์เครื่องใช้หรือเครื่องจักรสามารถติดต่อสื่อสารกันได้โดยตรง
- มนูษย์ใช้งานอุปกรณ์ต่างได้สะดวกขึ้น
- มีการเก็บข้อมูลการใช้งานตลอดเวลาโดยอัตโนมัติ
- ช่วยพัฒนาระบบ AI ให้ดีขึ้น



# มข. กับการพัฒนา IoT

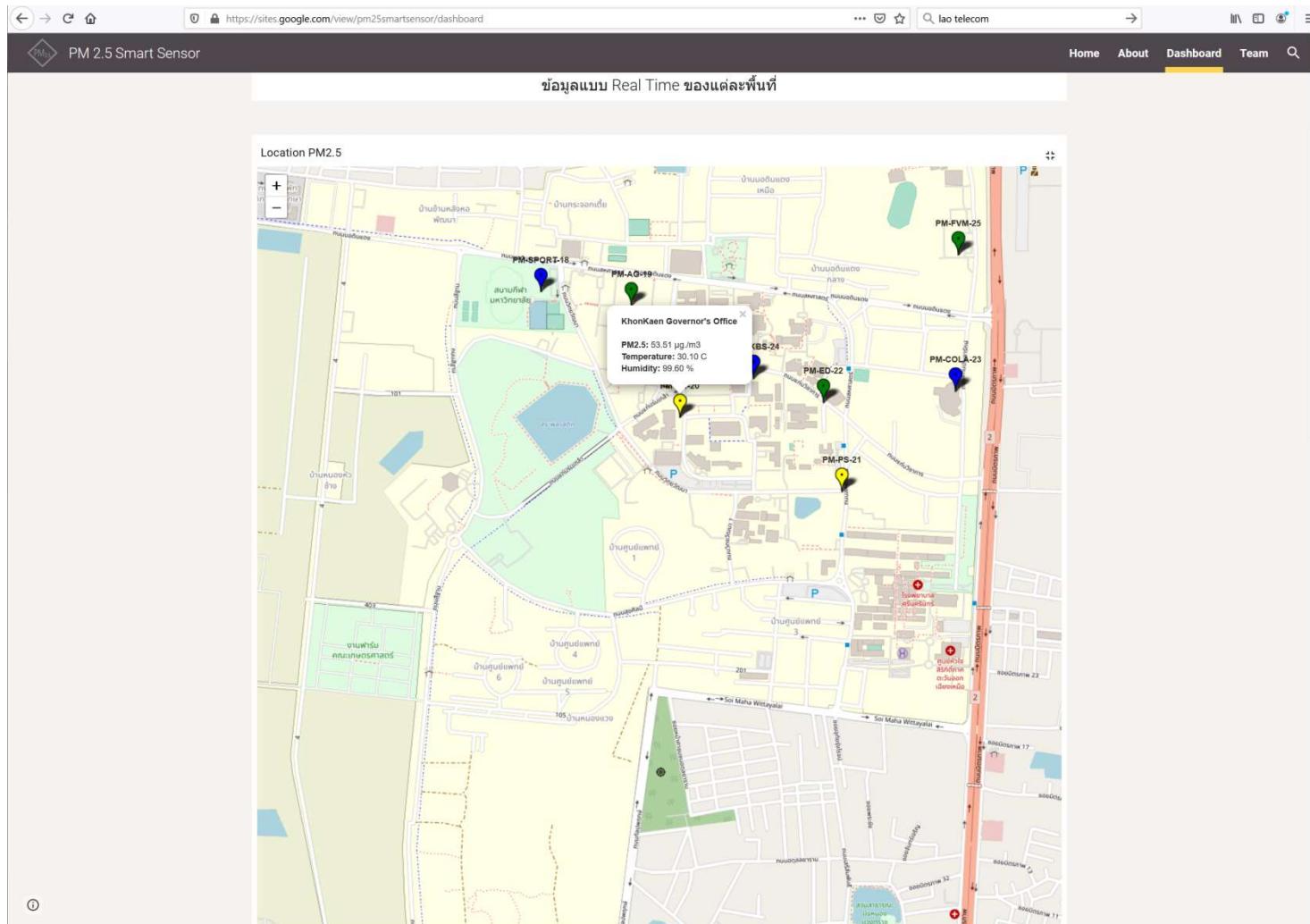
## Smart City

- ระบบตรวจจับการฝ่าฝืนสัญญาณไฟจราจรอัตโนมัติ
- ระบบตรวจจับไม่สวมหมวกนิรภัยอัตโนมัติ
- ระบบควบคุมสัญญาณไฟจราจรตามปริมาณรถ
- ถังขยะอัจฉริยะ (Smart bin)
- ลานจอดรถอัจฉริยะ (Smart parking)
- เซ็นเซอร์ตรวจสอบมลพิษและสภาพอากาศ PM2.5

## Agriculture

- แปลงปลูกผักอัจฉริยะ (Smart farm)

# เซ็นเซอร์ตรวจสอบมลพิษและสภาพอากาศ (PM2.5)



<https://sites.google.com/view/pm25smartsensor/dashboard>

# ເໜື້ນເຊົ່ວໝາຍສອບມລພິ່ງແລະສກາພອກາສ (PM2.5)

The screenshot shows a web-based dashboard titled "PM 2.5 Smart Sensor". The page URL is https://sites.google.com/view/pm25smartsensor/dashboard. The dashboard displays a table of data for nine different locations, each equipped with a PM2.5 smart sensor. The columns in the table include:

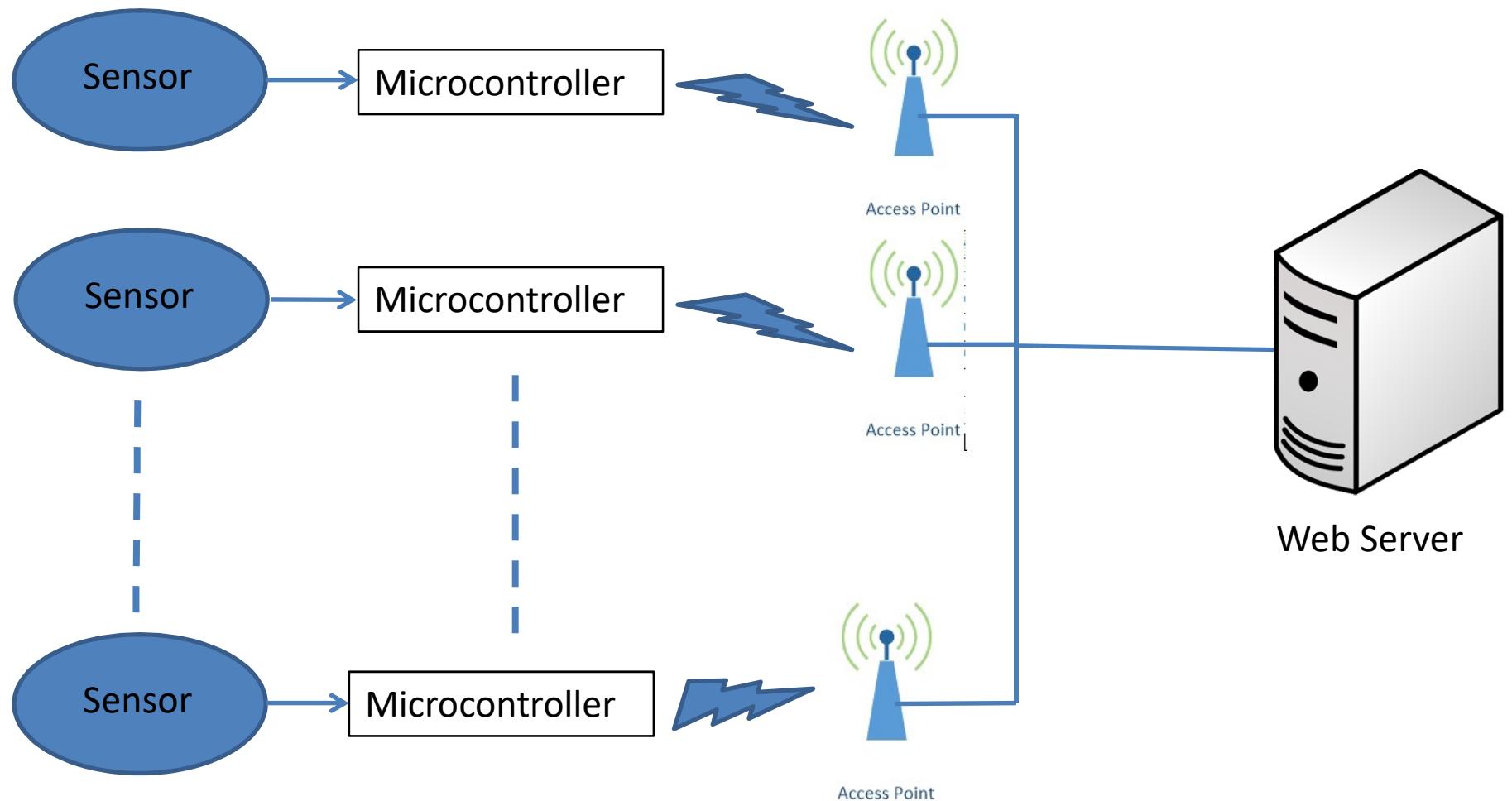
- Location Name (in Lao and English)
- Air quality
- PM2.5 concentration ( $\mu\text{g}/\text{m}^3$ )
- Temperature ( $^{\circ}\text{C}$ )
- Humidity (%)

The data is as follows:

| Location  | Air quality | PM2.5 ( $\mu\text{g}/\text{m}^3$ ) | Temperature ( $^{\circ}\text{C}$ ) | Humidity (%) |
|---|-------------|------------------------------------|------------------------------------|--------------|
| ກອງການກີ່ມິ້າ<br>PM-SPORT-WIFI                  | Good        | 5                                  | 34                                 | 65           |
| ຄະນະ<br>ເກະດວກສາສົ່ງ<br>PM-AG-WIFI              | Good        | 29                                 | 31                                 | 59           |
| ຄະນະ<br>ວິສວະກົມສາສົ່ງ<br>PM-EN-WIFI            | Moderate    | 54                                 | 30                                 | 100          |
| ຄະນະ<br>ເກສັ້ນສາສົ່ງ<br>PM-PS-WIFI              | Moderate    | 65                                 | 25                                 | 96           |
| ຄະນະຕຶກຂ່າ<br>ສາສົ່ງ<br>PM-ED-WIFI              | Good        | 33                                 | 27                                 | 95           |
| ໂທ<br>ວິທາລີຍາຮ<br>ປັກຮອງຫອງດີນ<br>PM-COLA-WIFI | Good        | 9                                  | 27                                 | 100          |
| ຄະນະບໍ່ທາງ<br>ຊູຮກົງແລະກາຮ<br>PM-HHSS-WIFI      | Good        | 5                                  | 22                                 | 100          |
| ຄະນະ<br>ສັດວະພາຫຍ<br>ສາສົ່ງ<br>PM-FVM-WIFI      | Good        | 44                                 | 17                                 | 83           |
| ນິ້ນສີຮານ                                       | Air quality | PM2.5                              | Temperature                        | Humidity     |

# เซ็นเซอร์ตรวจสอบมลพิษและสภาพอากาศ (PM2.5)

- ใช้เทคโนโลยี Wi-Fi



# ແນະນຳໄມໂຄຣຄອນໂທຣລເລ່ວ໌

ໄມໂຄຣຄອນໂທຣລເລ່ວ໌ເປັນຄອມພິວເຕອີ່ນາດເລື້ກບຣຈຸອູຍູໃນແພງວົງຈຽວມ (IC) ປະກອບດ້ວຍ

- ຜ່າຍປະມວລຜລ (CPU)
- ຜ່າຍຄວາມຈຳ (RAM / ROM / Flash)
- ຜ່ອງສໍ້ຜູານຂໍອມູລເຂົ້າ-ອອກ (I/O ports)
- ວຈຈະປະກອບອື່ນໆ ເຊັ່ນ USB, I2C, SPI, ADC

ໄມໂຄຣຄອນໂທຣລເລ່ວ໌ມີໜາຍຝູ້ຜົດລືດ

- Atmel: AVR
- ARM
- Tensilica (ESP8266, ESP32)

ມີການນຳມາໃໝ່ງານທີ່ຫລາກຫລາຍ ອາທີ:

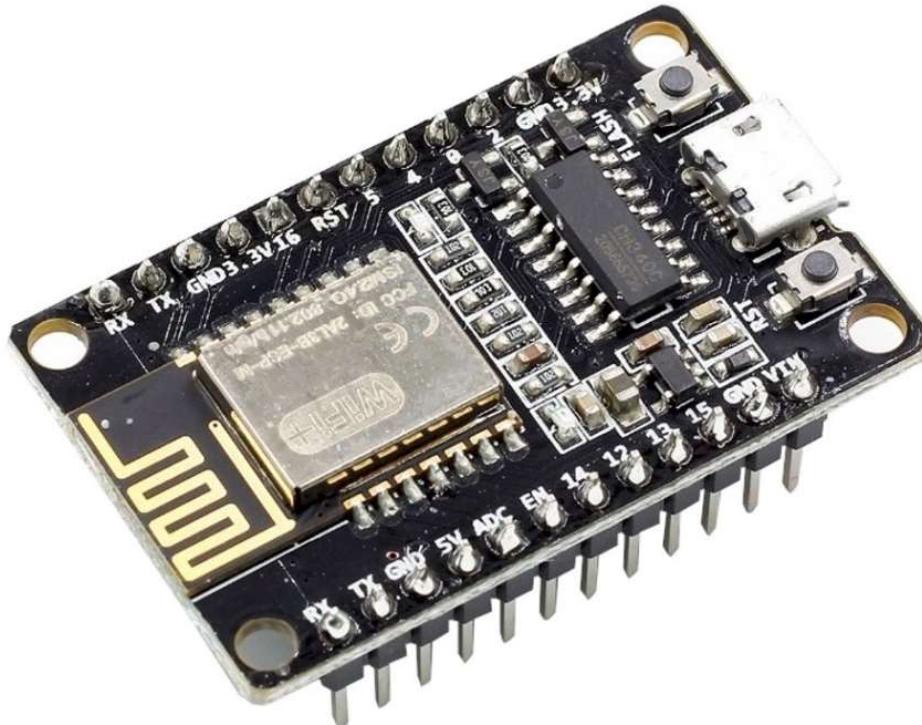
- ເຄື່ອງໃໝ່ໄຟຟ້າຕ່າງໆ
- ໃນຮຽນຕໍ່
- ທຸ່ນຍົນຕໍ່ແລະຮະບບຄວບຄຸມໃນໂຮງງານ



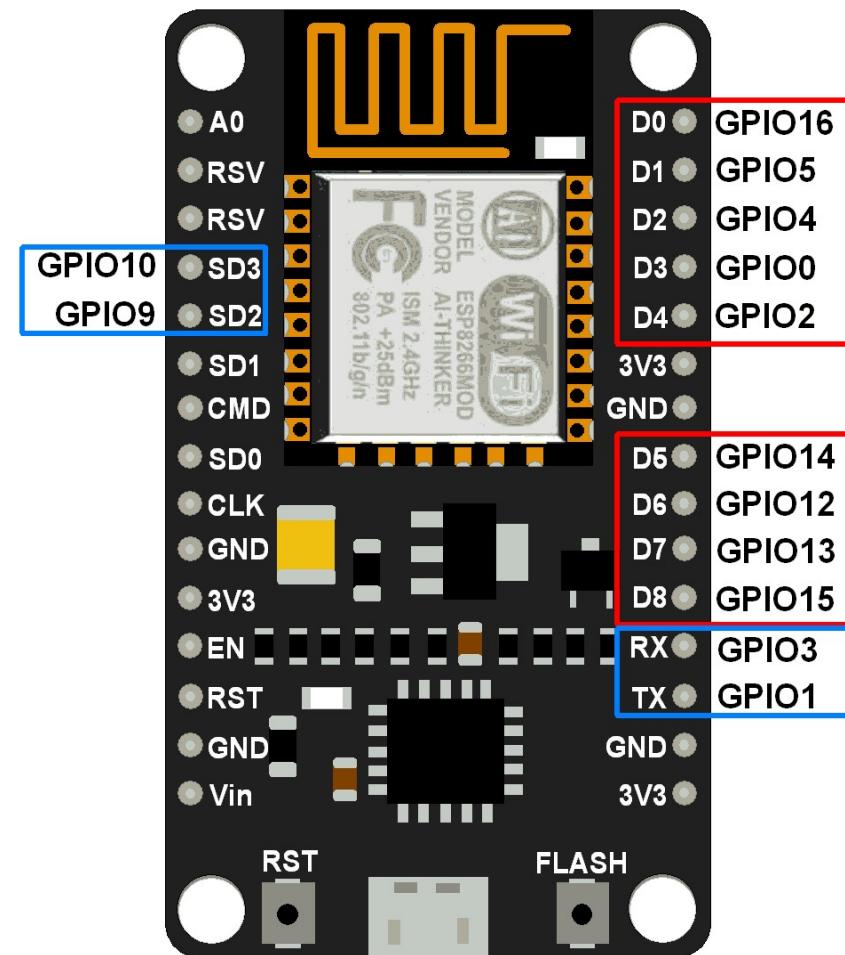
[www.TheEngineeringProjects.com](http://www.TheEngineeringProjects.com)

# ESP8266 (NodeMCU)

- เป็นไมโครคอนโทรลเลอร์สำหรับการพัฒนาระบบ IoT
- เขียนโปรแกรมโดยใช้ภาษา C/C++ หรือ Python
- ใช้งานร่วมกับ Arduino IDE ได้

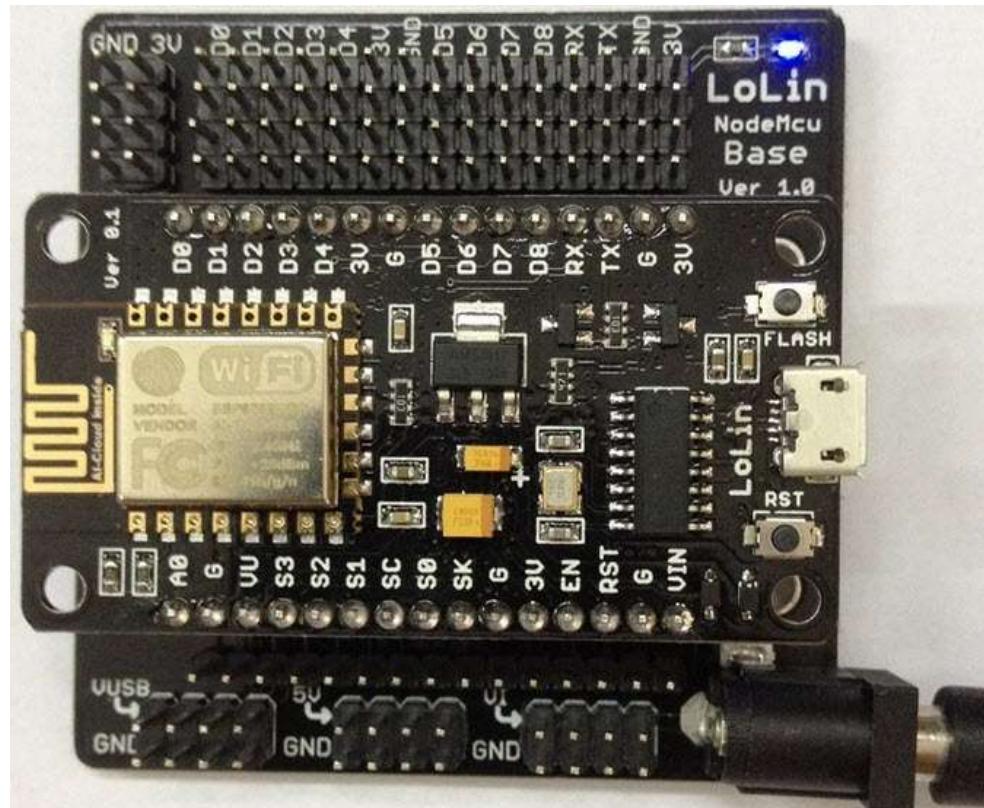


# การจัดขาของ NodeMCU



# NodeMCU Base

- เพิ่มจุดเชื่อมต่อสายไฟไปข้างๆ ของ Node MCU และ External Power Supply
- ใช้งานกับ Node MCU V3 เท่านั้น

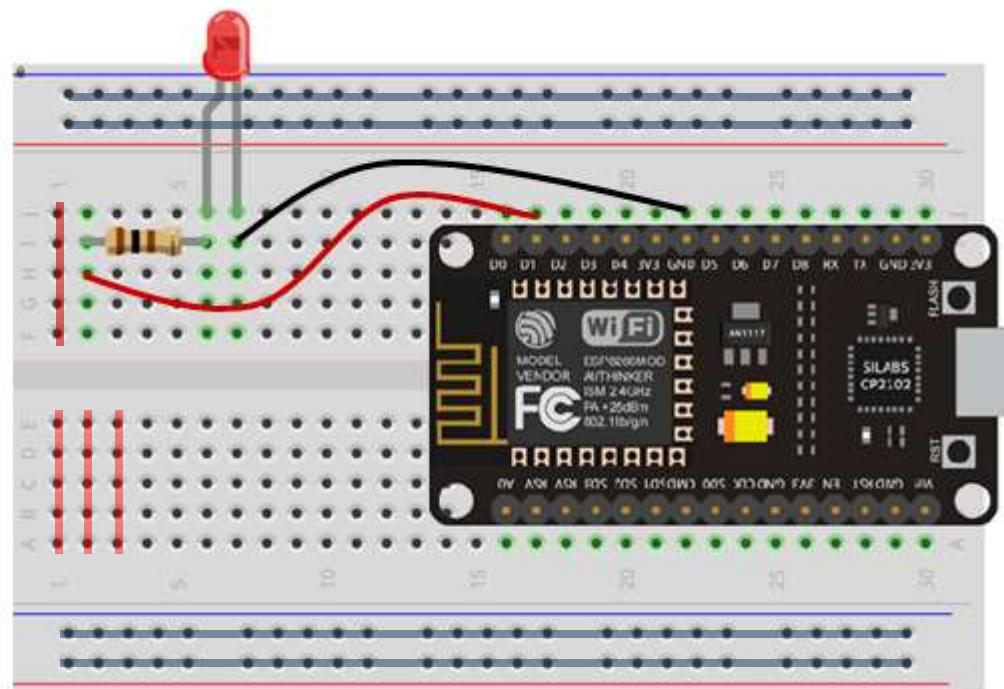


# คำอธิบายขาต่อสำหรับใช้งาน NodeMCU

- 5V & 3.3V เป็นขาที่จ่ายแรงดัน 5V และ 3.3V ใช้สำหรับจ่ายไฟให้กับวงจรภายนอก
- GND เป็นขาแรงดันอ้างอิง (ระดับแรงดัน 0V)
- Reset ขาเรียกตัวการทำงานของบอร์ด
- Vin เป็นขาที่จ่ายแรงดันไฟเลี้ยงเข้ามาให้บอร์ด
- RXD และ TXD เป็นชารับและส่งข้อมูลระหว่างบอร์ดและคอมพิวเตอร์
- A0 เป็นชารับสัญญาณอะโนดลอกจากภายนอก
- D0, D1, ..., D8 เป็นชารับส่งสัญญาณดิจิทัล

# การต่อวงจรเพื่อใช้งาน Node MCU

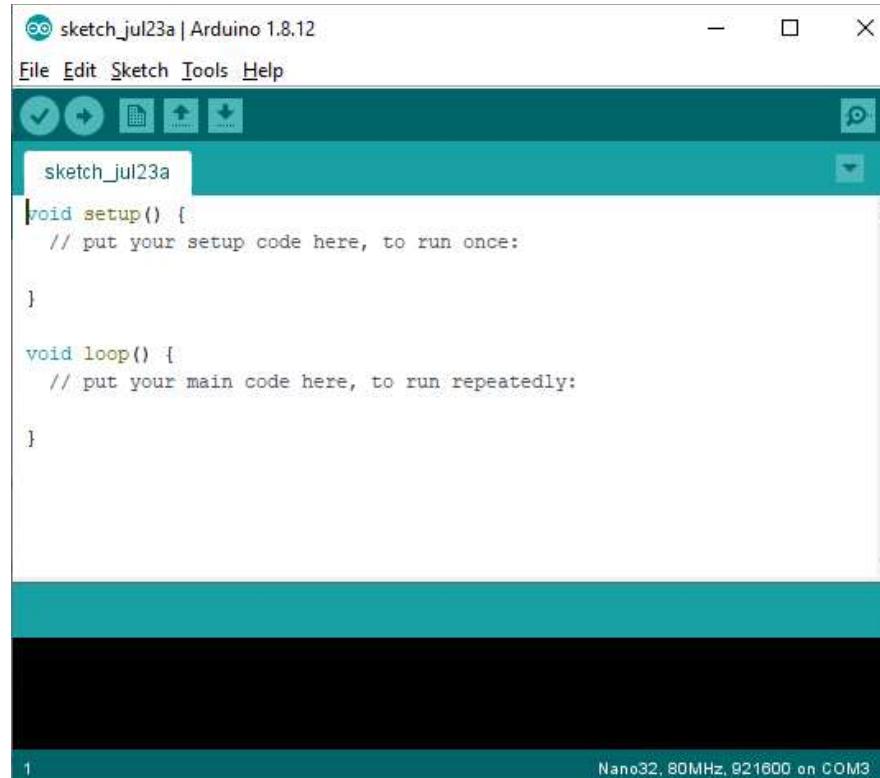
- ต่อวงจรบน Prototyping Board
- ภายใน prototyping board จะมีตัวนำเชื่อมจุดต่างๆ ไว้ให้แล้ว
- ด้านบนและล่างอย่างละ 2 แฉว เชื่อมต่อตามแนวโนน
- ตรงกลางเชื่อมต่อตามแนวตั้ง



# ซอฟต์แวร์สำหรับพัฒนา

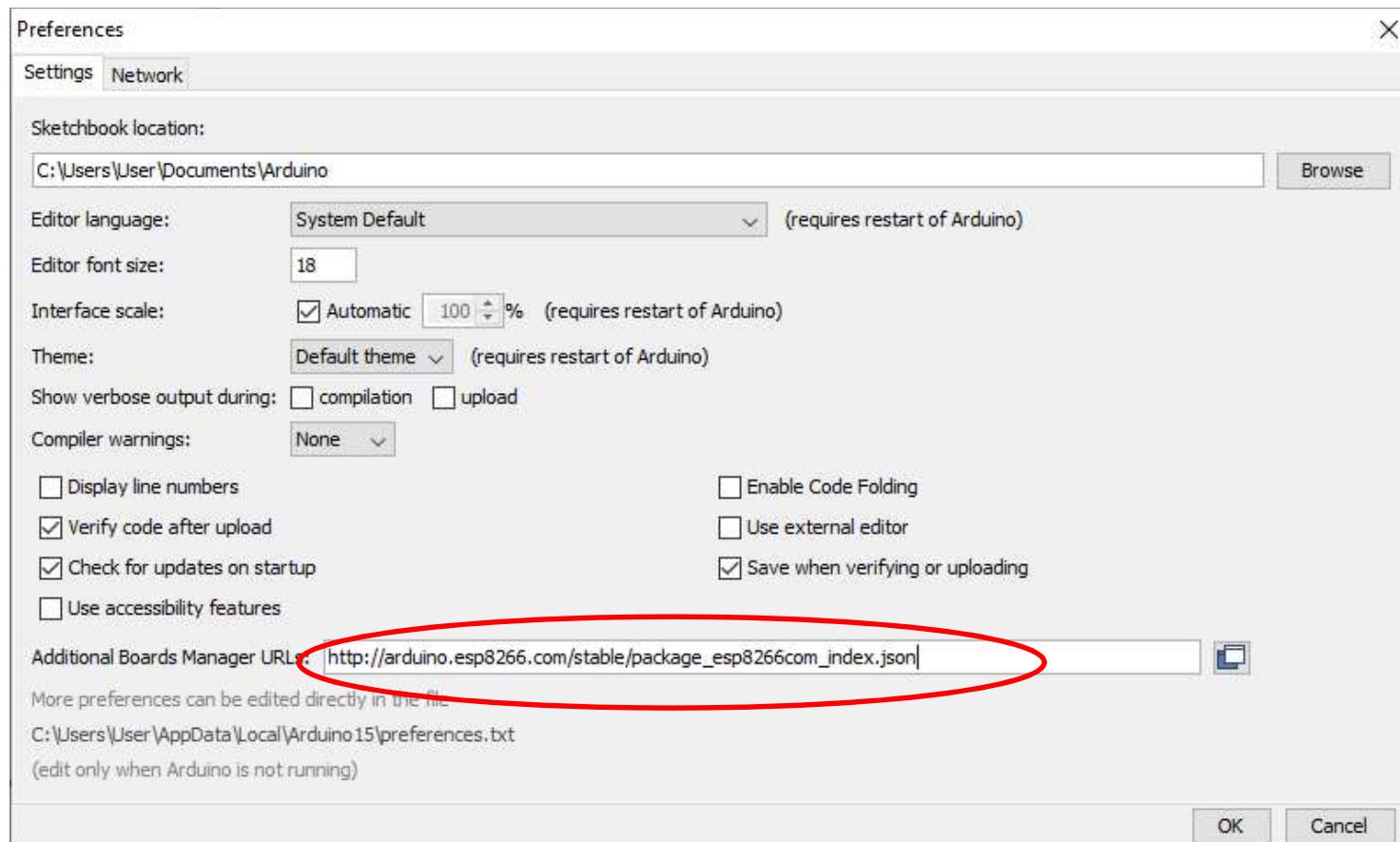
## Arduino IDE

- ใช้สำหรับการเขียนและแปลงโปรแกรมภาษา C และเขียนลง Node MCU
- ดาวน์โหลดได้จาก [arduino.cc/en/Main/Software](https://arduino.cc/en/Main/Software)



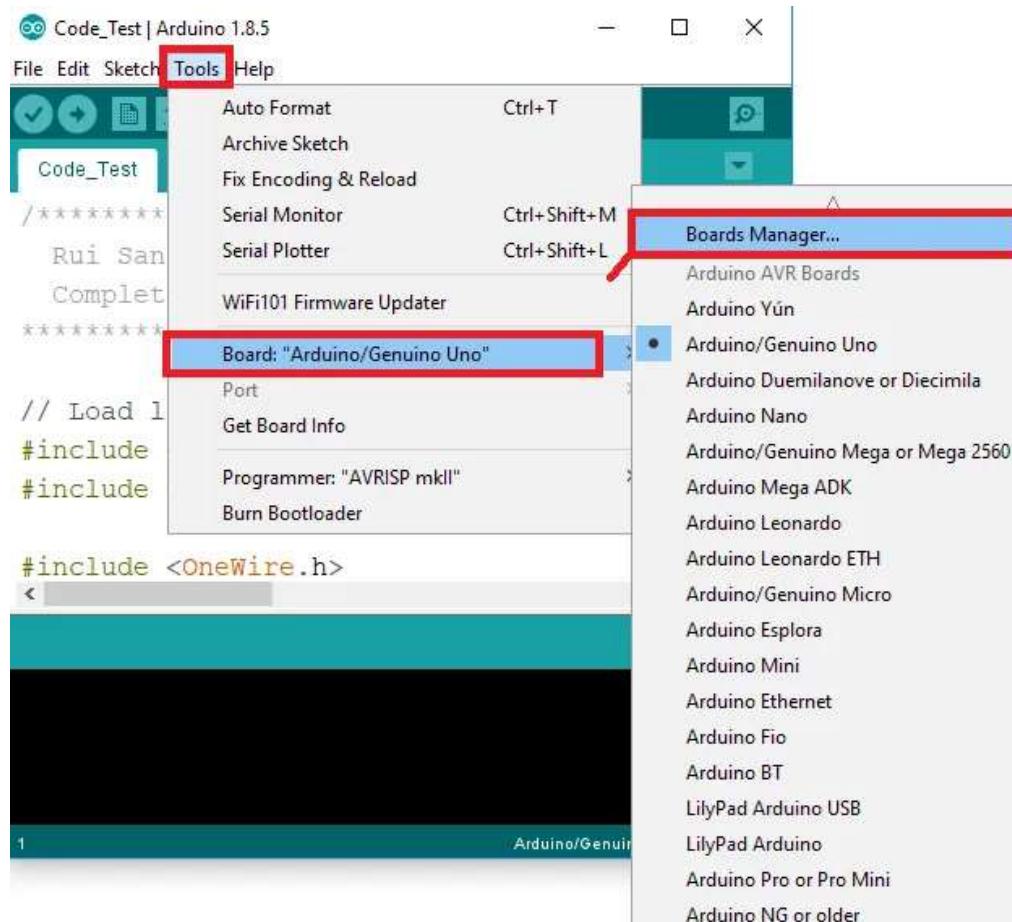
# วิธีเพิ่มบอร์ด Node MCU ลงใน Arduino IDE

- เปิดโปรแกรม Arduino IDE และไปที่เมนู File → Preferences
- พิมพ์ `http://arduino.esp8266.com/stable/package_esp8266com_index.json` ลงในช่อง Additional Boards Manager URLs ดังภาพ



# วิธีเพิ่ม NodeMCU ลงใน Arduino IDE

ไปที่เมนู Tools > Board > Boards Manager...



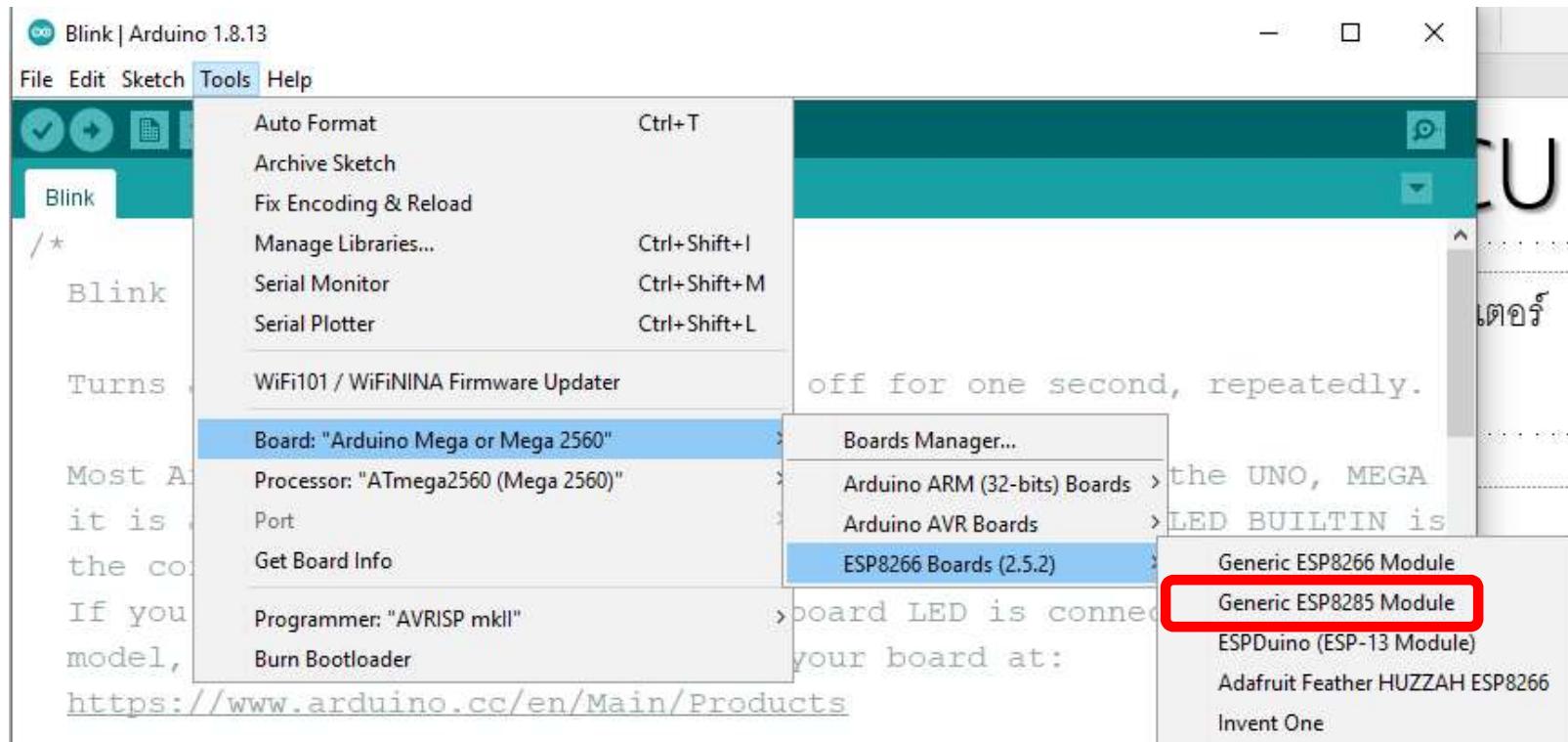
# วิธีเพิ่ม NodeMCU ลงใน Arduino IDE

- พิมพ์ esp8266 ในช่องค้นหา กดปุ่ม ENTER
- หน้าจอจะแสดงตัวเลือก 8266 by ESP8266 Community เลือก INSTALL



# การเชื่อมต่อ NodeMCU กับ PC

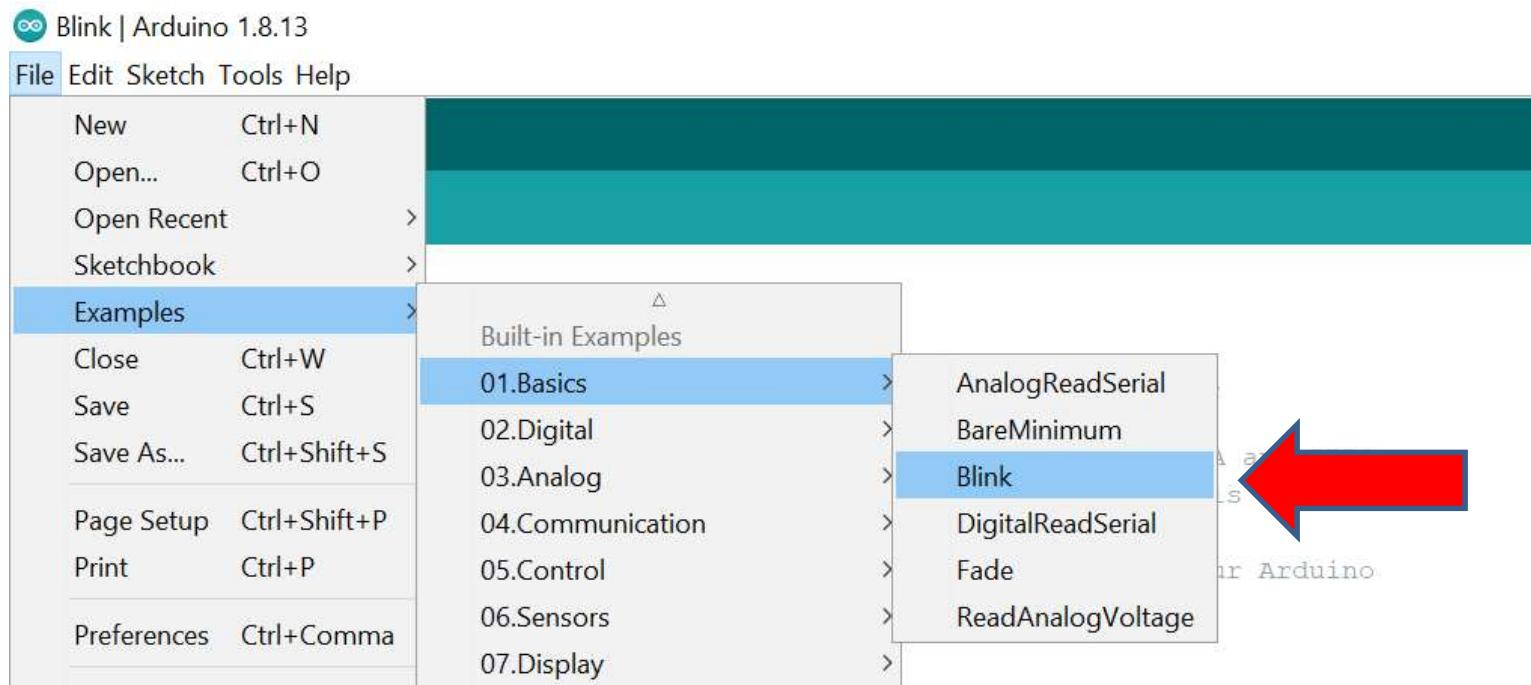
- ต่อสาย USB เข้ากับ Node MCU และ USB port ของคอมพิวเตอร์
- คอมพิวเตอร์จะติดตั้ง Driver ให้โดยอัตโนมัติ
- ไปที่เมนู Tools → Board → ESP8266 Boards → Generic ESP8266 Module



# การเชื่อมต่อ NodeMCU กับ PC

ทดสอบการเชื่อมต่อโดย

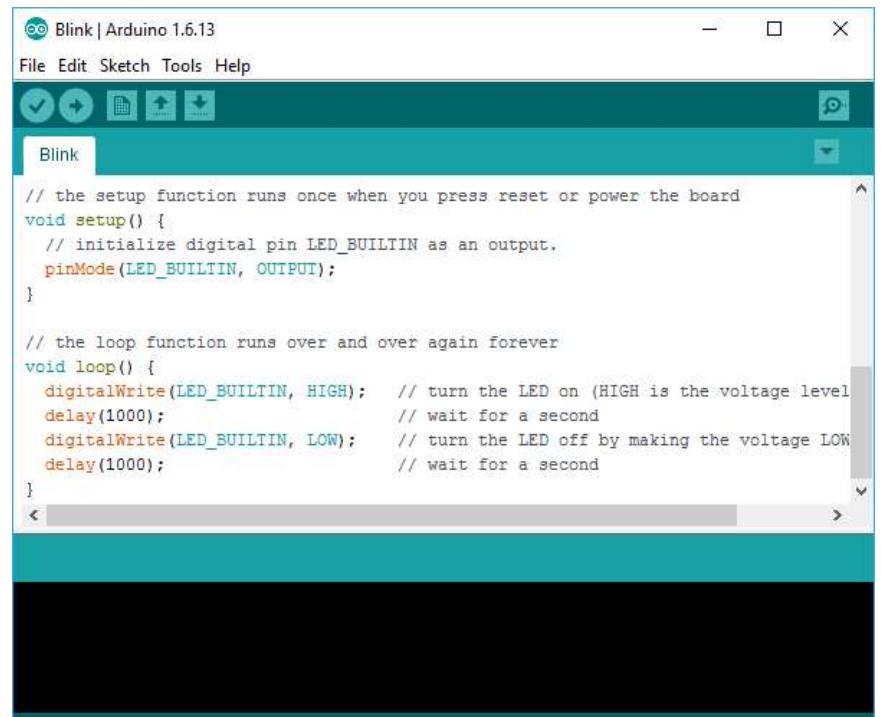
- ไปที่เมนู File → Examples → 01.Basics → Blink
- คลิกที่สัญลักษณ์ Download ➡
- รอให้โปรแกรมทำการแปลภาษาและดาวน์โหลด จะเห็นไฟกระพริบบนบอร์ด Node MCU



# การเขียนโปรแกรมด้วย Arduino IDE

## หน้าที่ของ Arduino IDE

- เขียนและแก้ไขโปรแกรม
- แปลงโปรแกรมให้เป็นภาษาเครื่อง
- ส่งโปรแกรมที่แปลงแล้วเข้าบอร์ด



The screenshot shows the Arduino IDE interface with the title bar "Blink | Arduino 1.6.13". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for file operations like Open, Save, and Print. The main workspace is titled "Blink" and contains the following code:

```
// the setup function runs once when you press reset or power the board
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
  digitalWrite(LED_BUILTIN, HIGH);    // turn the LED on (HIGH is the voltage level
  delay(1000);                      // wait for a second
  digitalWrite(LED_BUILTIN, LOW);     // turn the LED off by making the voltage LOW
  delay(1000);                      // wait for a second
}
```

At the bottom right, it says "Polarlu A-Star 32U4 on COM4".

# การเขียนโปรแกรมด้วย Arduino IDE

- ใช้ Arduino IDE เพื่อเขียนโปรแกรม หรือ source code เรียกว่า ‘sketch’
- Sketch จะใช้ไวยากรณ์ของภาษา C และ C++
- โครงสร้างเริ่มต้นของ sketch มีดังนี้:

```
void setup() {  
    // คำสั่งในนี้จะทำงาน 1 ครั้ง  
}  
  
void loop() {  
    // คำสั่งในนี้จะทำงานวนซ้ำ  
}
```

# การเขียนโปรแกรมด้วย Arduino IDE

## ตัวอย่างโปรแกรม

```
// บรรทัดหมายเหตุ
void setup() {
    // ใช้ย่อหน้าแสดงลำดับชั้นของโครงสร้างโปรแกรม
    pinMode(LED_BUILTIN, OUTPUT);
}

// the loop function runs over and over again forever
void loop() {
    digitalWrite(LED_BUILTIN, HIGH);      // turn the LED on (HIGH is the voltage level)
    delay(1000);                      // wait for a second
    digitalWrite(LED_BUILTIN, LOW);       // turn the LED off by making the voltage LOW
    delay(1000);                      // wait for a second
}
```

# ดิจิทัลพอร์ท (Digital Port)

- ดิจิทัลพอร์ททำหน้าที่รับหรือส่งสัญญาณดิจิทัล (“0” หรือ “1”) หรือระดับแรงดัน 0 โวลท์ และ 5 โวลท์
- ใช้คำสั่ง `pinMode()` เพื่อกำหนดว่าจะให้พอร์ททำหน้าที่รับหรือส่ง

## คำอธิบาย

กำหนดหน้าที่ของดิจิทัลพอร์ท โดย `input` จะเป็นการรับ และ `output` จะเป็นการส่ง

## รูปแบบ

`pinMode(ขา, mode)`

## Parameters

ขา: ขาสัญญาณของ Node MCU

mode: INPUT, OUTPUT

## ค่าส่งกลับ

ไม่มี

# การเขียน (ส่ง) ข้อมูลอุปกรณ์ดิจิทัลพอร์ท

## `digitalWrite()`

### คำอธิบาย

เป็นการส่งข้อมูลอุปกรณ์ดิจิทัลพอร์ทที่กำหนด

### รูปแบบ

`digitalWrite(ขา, value)`

### Parameters

ขา: หมายเลขขาที่ต้องการส่งข้อมูล (หมายเลข GPIO)

value: HIGH (ปล่อยแรงดัน 5V) หรือ LOW (ปล่อยแรงดัน 0V)

### ค่าส่งกลับ

ไม่มี

# การอ่าน (รับ) ข้อมูลเข้าทางดิจิทัลพอร์ท

## `digitalRead()`

คำอธิบาย

เป็นการรับข้อมูลเข้าทางดิจิทัลพอร์ทที่กำหนดรูปแบบ

`digitalRead(ขา)`

Parameters

ขา: หมายเลขขาที่ต้องการส่งข้อมูล (หมายเลข GPIO)

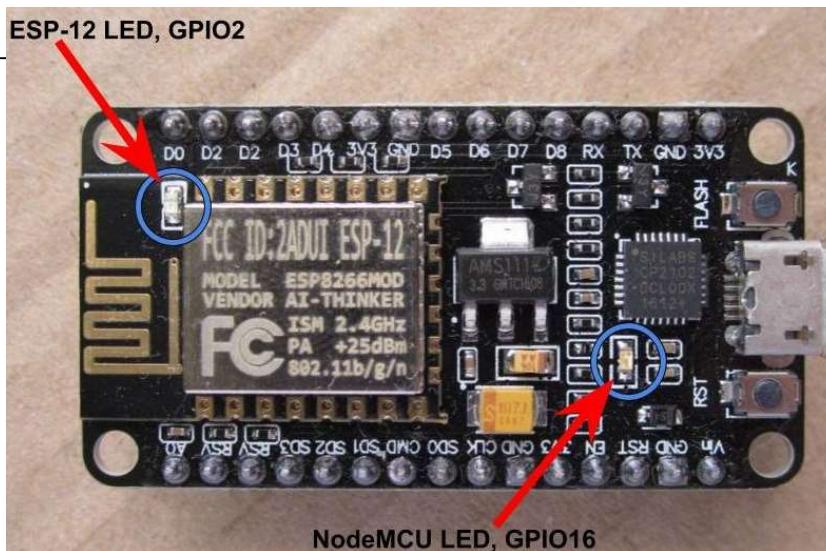
ค่าส่งกลับ

HIGH หรือ 1 ถ้ามีแรงดัน 5V ที่ขา หรือ  
LOW หรือ 0 ถ้ามีแรงดัน 0V ที่ขา

# ตัวอย่างการใช้งานดิจิทัลพอร์ท

โปรแกรมไฟกระพริบโดยใช้ LED บน Node MCU

```
int LED = 2;  
void setup() {  
    pinMode(LED, OUTPUT);  
}  
  
void loop() {  
    digitalWrite(LED, LOW);  
    delay(1000);  
    digitalWrite(LED, HIGH);  
    delay(1000);  
}
```



# การสื่อสารกับ PC

คำสั่ง **Serial.print()** และ **Serial.println()**

คำอธิบาย

ใช้ส่งข้อมูล (ข้อความหรือตัวเลข) จาก Node MCU มาแสดงผลใน Serial monitor ของ Arduino IDE จะต้องใช้ร่วมกับคำสั่ง **Serial.begin()** เช่น

รูปแบบ

**Serial.begin(9600)**

**Serial.println(ข้อมูล)** หรือ **Serial.print(ข้อมูล)**

ตัวอย่าง

```
void setup() {  
    Serial.begin(9600);  
    Serial.println();  
    Serial.print("Hello!, ");  
    Serial.println("world");  
    Serial.print("3x7 = ");  
    Serial.println(3*7);  
}  
void loop() {  
}
```

# การสื่อสารกับ PC

คำสั่ง **Serial.read()**

คำอธิบาย

ใช้รับข้อมูลจาก Node MCU โดยจะต้องใช้ร่วมกับคำสั่ง **Serial.available()** เช่น  
รูปแบบ

**ตัวแปร = Serial.read()**

ตัวอย่าง

```
int x;  
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    if(Serial.available() > 0) {  
        x = Serial.read();  
        Serial.print("You sent ");  
        Serial.println(x, DEC);  
    }  
}
```

# โครงสร้างโปรแกรม

**if**

คำริบาย

ใช้เลือกการทำงานตามเงื่อนไขที่กำหนด โดยเงื่อนไขจะมีค่าเป็น จริง หรือ เท็จ

รูปแบบ

```
if (เงื่อนไข) {  
    คำสั่ง  
}
```

ตัวอย่าง

```
if (Pin7 == HIGH) {  
    digitalWrite(LEDpin1, HIGH);  
}  
else {  
    digitalWrite(LEDpin1, LOW);  
}
```

```
if (เงื่อนไข1) {  
    คำสั่ง1  
}  
else if (เงื่อนไข2) {  
    คำสั่ง2  
}  
else {  
    คำสั่ง3  
}
```

# โครงสร้างโปรแกรม

## การกำหนดเงื่อนไข

- เงื่อนไขจะต้องมีค่าเป็น จริง หรือ เท็จ เท่านั้น
- ลักษณะของเงื่อนไขจะเป็นการเปรียบเทียบว่า
  - เท่ากับ ใช้สัญลักษณ์  $=$
  - ไม่เท่ากับ ใช้สัญลักษณ์  $\neq$
  - มากกว่า ใช้สัญลักษณ์  $>$
  - มากกว่าหรือเท่ากับ ใช้สัญลักษณ์  $\geq$
  - น้อยกว่า ใช้สัญลักษณ์  $<$
  - น้อยกว่าหรือเท่ากับ ใช้สัญลักษณ์  $\leq$
- เงื่อนไขสามารถมีได้มากกว่า 1 โดยใช้ตัวเชื่อม และ  $\&\&$ , หรือ  $\|$

ตัวอย่าง

```
if (i  $\geq$  0)
    digitalWrite(LEDpin1, HIGH);
else
    digitalWrite(LEDpin1, LOW);
```

```
if(x  $\leq$  10  $\&\&$  x  $>$  0)
```

```
if(number == -1  $\|$  status != 1)
```

```
if(a == 1  $\&\&$  b == 2  $\&\&$  c == 3  $\|$  c == 4)
```

# โครงสร้างโปรแกรม

## for

คำอธิบาย

เป็นคำสั่งเพื่อให้กลุ่มคำสั่งที่กำหนดถูกทำงานจำนวนซ้ำตามจำนวนครั้งที่กำหนด

รูปแบบ

```
for (ค่าเริ่มต้น; เงื่อนไข; การเพิ่มค่า) {  
    คำสั่ง  
}
```

ตัวอย่าง

```
int Count = 0;  
int i;  
void setup() {  
    Serial.begin(9600);  
    for (i = 0; i < 5; i++) {  
        Serial.println(Count);  
        Count = Count+2;  
    }  
}  
void loop() {  
}
```

# โครงสร้างโปรแกรม

จากตัวอย่าง

```
for (i = 0; i < 5; i++) {  
    Serial.println(Count);  
    Count = Count+2;  
}
```

- ตัวแปร **i = 0** เป็นค่าเริ่มต้น
- **i <= 10** เป็นเงื่อนไข
- **i++** เป็นการเพิ่มค่าให้กับตัวแปร **i** โดย **i++ → i = i+1**
- ตัวแปร **i** จะมีค่าเป็น **0, 1, 2, 3, 4**
- ตัวแปร **Count** จะมีค่าเป็น **0, 2, 4, 6, 8**

# แบบฝึกหัด

ตัวอย่าง

โปรแกรมสร้างสูตรคูณแม่ 2

```
int i;  
void setup() {  
    Serial.begin(9600);  
    Serial.println();  
    for(i = 1; i <= 12; i++) {  
        Serial.print(i); Serial.print(" "); Serial.println(i*2);  
    }  
}  
void loop() {  
}
```

# แบบฝึกหัด

ตัวอย่าง

โปรแกรมคำนวณ  $n!$

```
int k;  
int n = 7;  
int nfact = 1;  
void setup() {  
    Serial.begin(9600);  
    Serial.println();  
    for(k = 1; k <= n; k++) {  
        nfact = nfact*k;  
    }  
    Serial.print(n); Serial.print("! = "); Serial.print(nfact);  
}  
void loop() {  
}
```

# แบบฝึกหัด

1. เขียนโปรแกรมเพื่อให้หลอดไฟกระพริบ 10 ครั้งแล้วดับ
2. เขียนโปรแกรมเพื่อให้หลอดไฟกระพริบ 100 ครั้งแล้วดับ พร้อมทั้งแสดงจำนวนครั้งที่กระพริบผ่านทาง Serial terminal

```
int k;
int n = 7;
int nfact = 1;
void setup() {
    Serial.begin(9600);
    Serial.println();
    for(k = 1; k <= n; k++) {
        nfact = nfact*k;
    }
    Serial.print(n); Serial.print("! = ");
    Serial.print(nfact);
}
void loop() {
```

# โครงสร้างโปรแกรม

while

คำอธิบาย

เป็นการวนซ้ำจนกว่าเงื่อนไขที่กำหนดจะเป็นเท็จ

รูปแบบ

```
while (เงื่อนไข) {  
    คำสั่ง  
}
```

ตัวอย่าง

```
int k;  
void setup() {  
    Serial.begin(9600);  
    while (k <= 10) {  
        Serial.println(k);  
        k = k+1;  
    }  
}  
void loop() {  
}
```

# แบบฝึกหัด

ตัวอย่าง

โปรแกรมแสดงจำนวนที่ 13 หารลงตัว จาก 0-100

```
int i;  
int n = 13;
```

```
void setup() {  
    Serial.begin(9600);  
    Serial.println();  
    i = 1;  
    while(i*n < 100) {  
        Serial.println(i*n);  
        i = i+1;  
    }  
    Serial.print("There are "); Serial.print(i-1); Serial.println(" numbers");  
}  
void loop() {  
}
```

# ชนิดข้อมูล

|               |   |
|---------------|---|
| boolean       | ‘true’, ‘false’   |
| char          | ขนาด 8-bit มีค่าระหว่าง -128 ถึง 127                      |
| unsigned char | ขนาด 8-bit มีค่าระหว่าง 0 ถึง 255                         |
| byte          | เหมือน <i>unsigned char</i>                               |
| int           | ขนาด 16-bit มีค่าระหว่าง -32,768 ถึง 32,767               |
| unsigned int  | ขนาด 16-bit มีค่าระหว่าง 0 ถึง 65535                      |
| word          | เหมือน <i>unsigned int</i>                                |
| long          | ขนาด 32-bit มีค่าระหว่าง -2,147,483,648 ถึง 2,147,483,647 |
| unsigned long | ขนาด 32-bit มีค่าระหว่าง 0 ถึง 4,294,967,295              |
| short         | เหมือน <i>int</i>   |
| float         | ขนาด 32-bit มีค่าระหว่าง -3.4028235E+38 ถึง 3.4028235E+38 |
| double        | เหมือน <i>float</i>                                       |
| string        | ลำดับของตัวอักษร  |

# แบบฝึกหัด

ตัวอย่าง

โปรแกรมสร้างตารางแปลงค่าอุณหภูมิจาก °C เป็น °F

```
float C, F;
```

```
void setup() {
    Serial.begin(9600);
    Serial.println();
    Serial.println("-----");
    Serial.println(" C      F ");
    Serial.println("-----");
    for(C = -10; C <= 40; C++) {
        // C/5 = (F-32)/9
        F = (9.0/5.0)*C + 32.0;
        Serial.print(C,2); Serial.print(" "); Serial.println(F,2);
    }
}
void loop() {
}
```

# คำสั่งใน Arduino IDE

ดิจิทัล

pinMode()

digitalWrite()

digitalRead()

แอนะล็อก

analogReference()

analogRead()

analogWrite() - PWM

รับส่งข้อมูลอื่นๆ

tone()

noTone()

shiftOut()

shiftIn()

pulseIn()

คณิตศาสตร์

min()

max()

abs()

constrain()

map()

pow()

sqrt()

สื่อสาร

Serial

Stream

เวลา

millis()

micros()

delay()

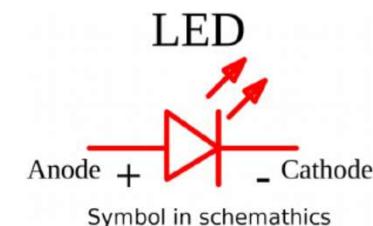
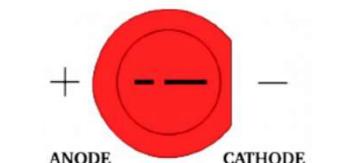
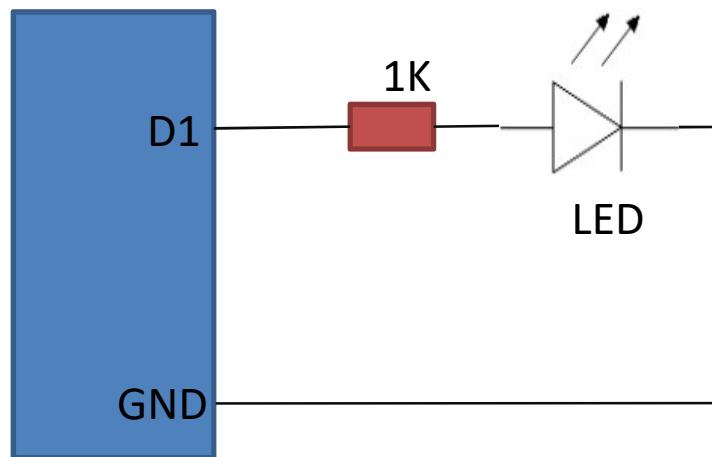
delayMicroseconds()

ไปที่เมนู Help → Reference

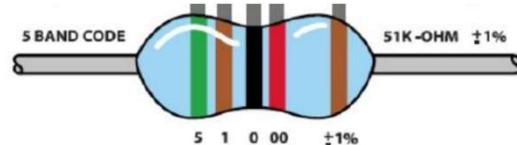
# การต่อ LED ภายนอก

- ต้องจรตามภาพด้านล่าง
- ไปที่เมนู File → Examples → 01. Basics → Blink
- แก้ไขโปรแกรม โดยเปลี่ยน `LED_BUILTIN` ให้เป็น 5
- ทดสอบการทำงาน

Node MCU



Symbol in schematics



## รหัสสี

แดง แดง ดำ  
น้ำตาล ดำ แดง  
น้ำตาล ดำ ส้ม

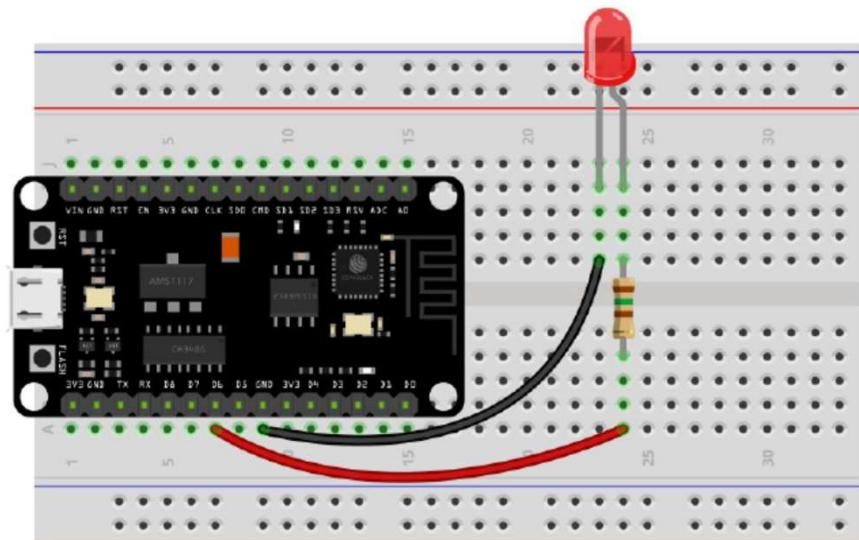
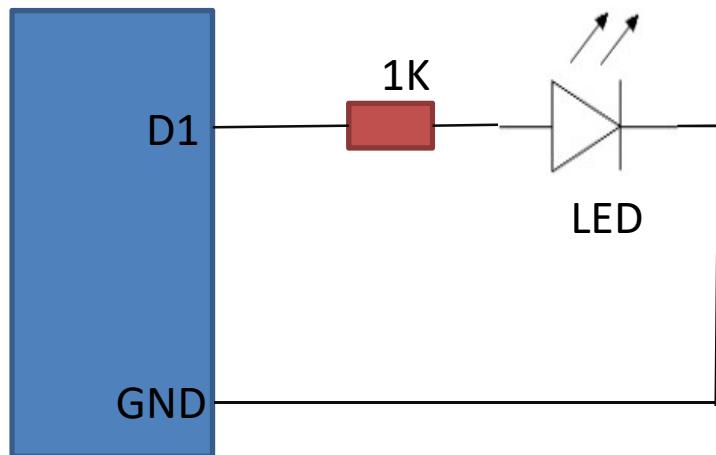
## ค่าความต้านทาน

220 Ohm  
1K Ohm  
10K Ohm

# การต่อ LED ภายนอก

- ต่อวงจรตามภาพด้านล่าง
- ไปที่เมนู *File* → *Examples* → *01. Basics* → *Blink*
- แก้ไขโปรแกรม โดยเปลี่ยน **LED\_BUILTIN** ให้เป็น 5
- ทดสอบการทำงาน

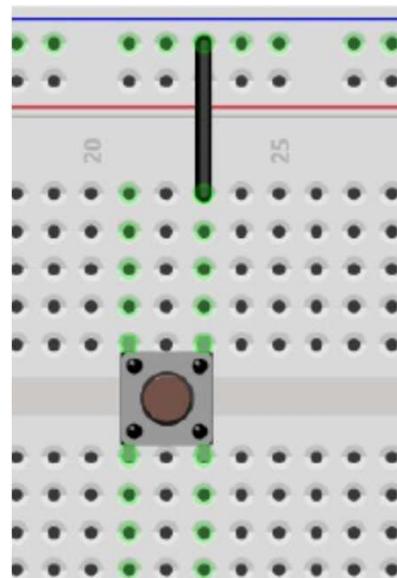
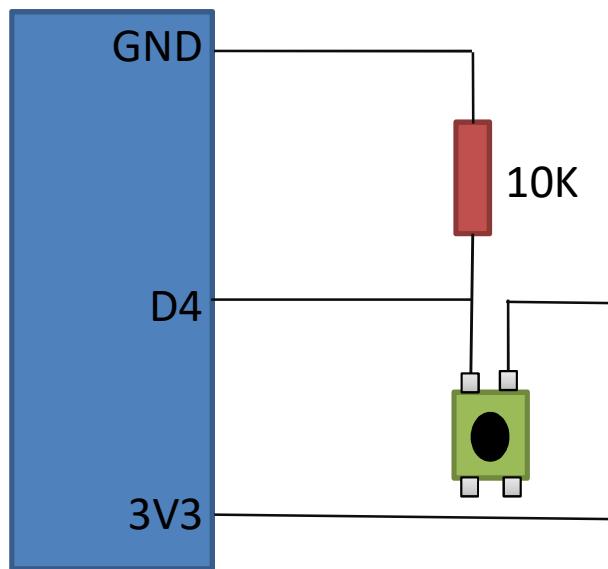
Node MCU



# การต่อสวิตซ์

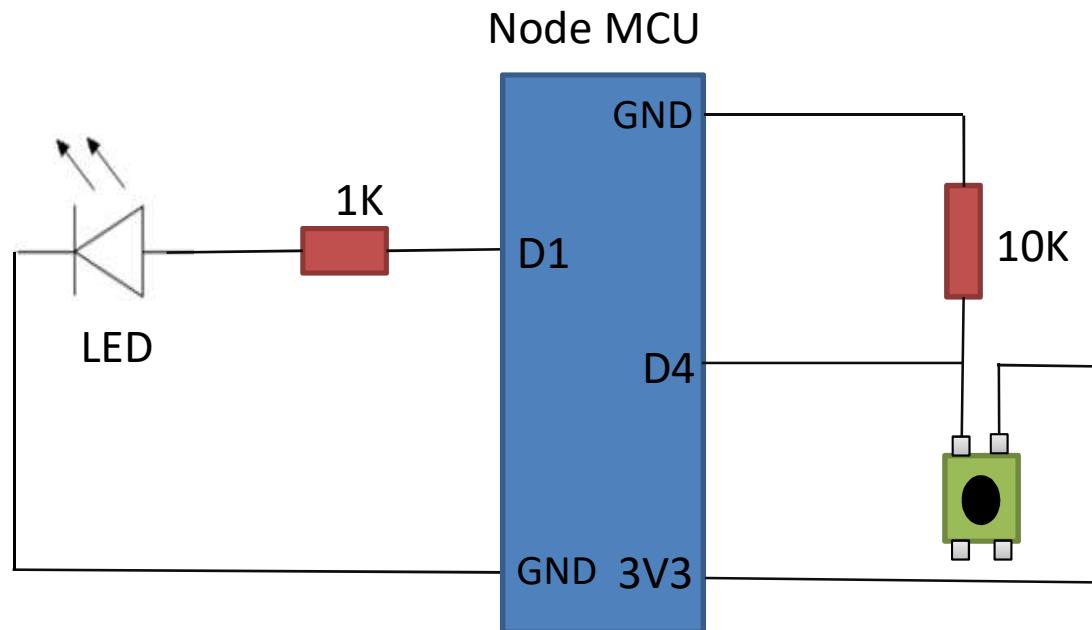
- ไปที่เมนู *File* → *Examples* → *01. Basics* → *DigitalReadSerial*
- เพิ่มวงจรจากข้อที่แล้วให้เป็นตามรูปด้านล่าง

Node MCU



# แบบฝึกหัดการใช้งานสวิตซ์และ LED

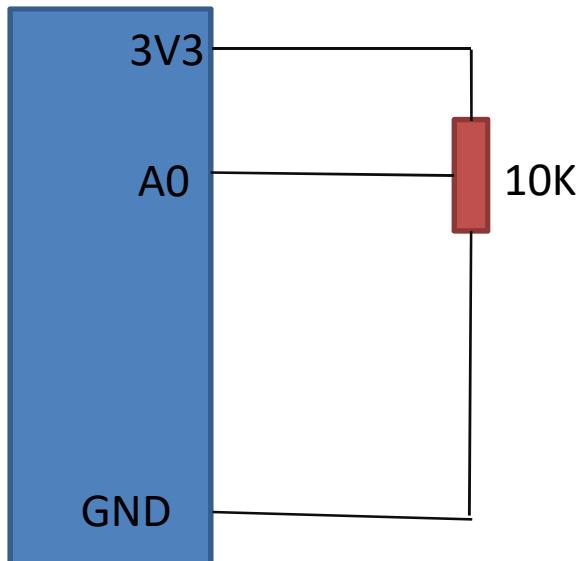
1. เขียนโปรแกรมให้กดปุ่มแล้ว LED สว่าง เมื่อปล่อยปุ่ม LED ดับ
2. เขียนโปรแกรมให้กดปุ่มแล้ว LED ดับ เมื่อปล่อยปุ่ม LED สว่าง
3. เขียนโปรแกรมให้กดปุ่มแล้ว LED กระพริบ 5 ครั้ง



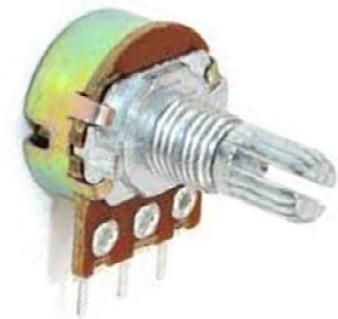
# การรับสัญญาณเอนะลอก

- ใช้ตัวต้านทานปรับค่าได้ทำหน้าที่เป็นแหล่งกำเนิดสัญญาณเอนะลอก
- สัญญาณเอนะลอกซึ่งมีระดับแรงดัน 0 – 3.3V ให้เป็นจำนวนเต็มที่มีค่า 0 – 1023
- ต้องจรวจตามรูป และทดสอบการทำงานโดยใช้โปรแกรมดังต่อไปนี้

Node MCU

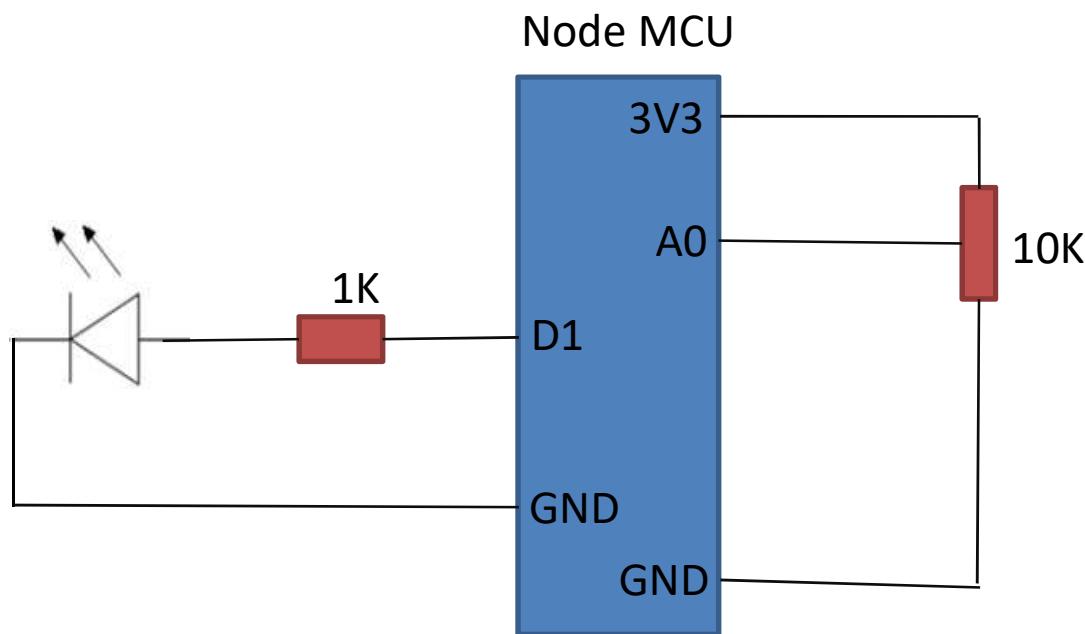


```
int AnalogValue = 0;  
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    AnalogValue = analogRead(A0);  
    Serial.println(AnalogValue);  
    delay(500);  
}
```



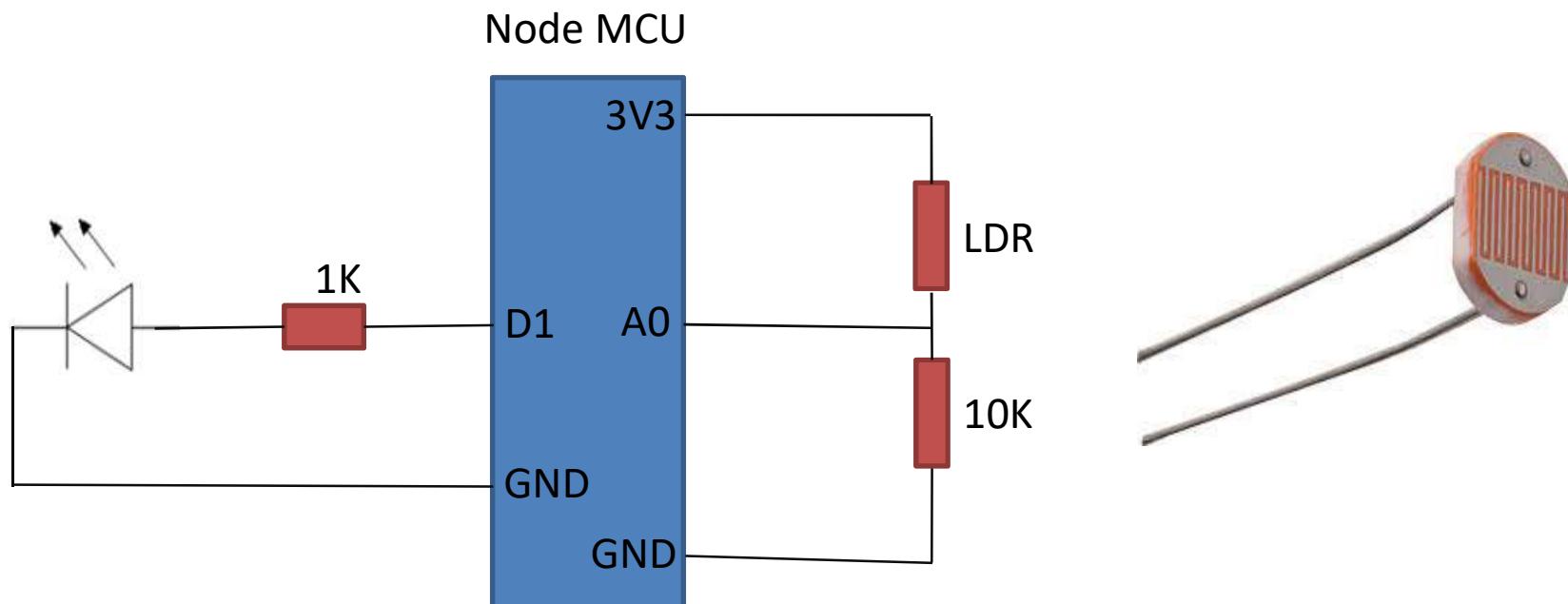
# แบบฝึกหัด

เขียนโปรแกรมควบคุมไฟกระพริบที่สามารถปรับอัตราเร็วของการกระพริบได้โดยใช้ตัวต้านทานปรับค่าได้



# ตัวต้านทานเปลี่ยนค่าตามแสง

- ตัวต้านทานเปลี่ยนค่าตามแสง Light-dependent Resistor (LDR) จะมีความต้านทานเปลี่ยนไปตามความเข้มแสงที่ต่ำกระหบ
- ความเข้มแสงสูงความต้านทานจะต่ำ ความเข้มแสงต่ำความต้านทานจะสูง



# แบบฝึกหัด

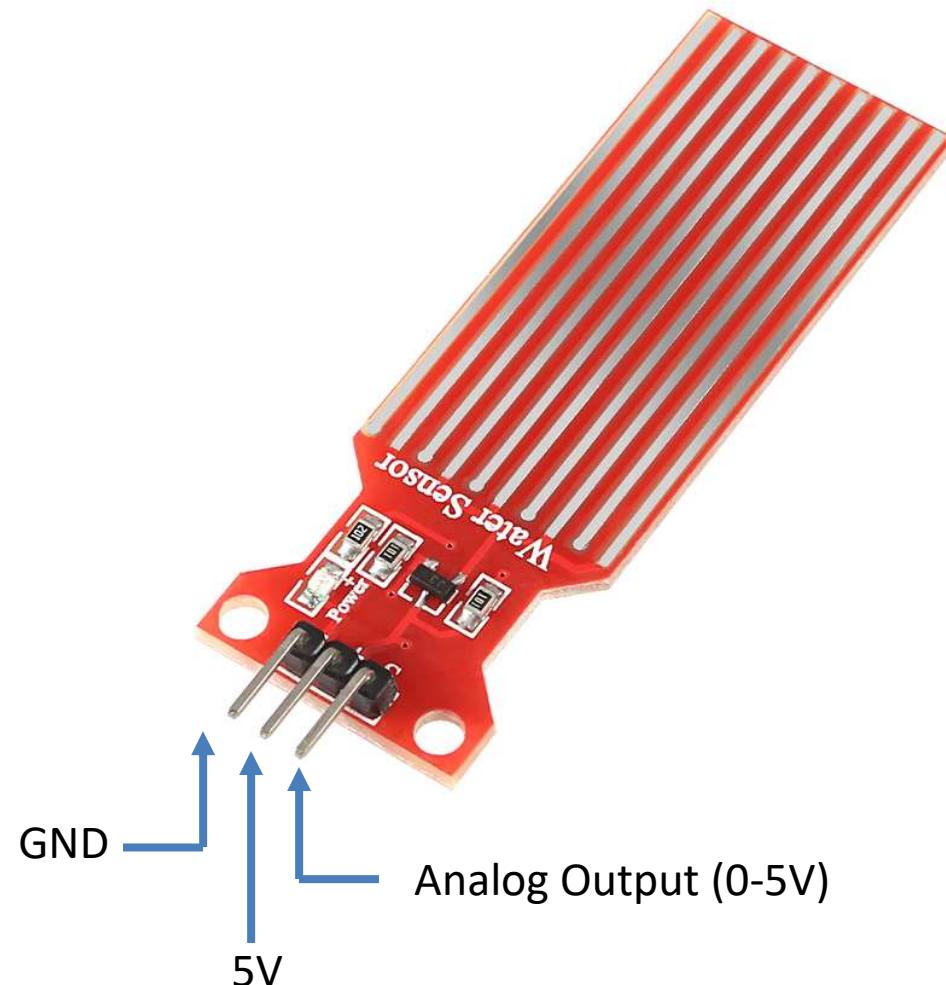
ตัวอย่าง

โปรแกรมแสดงค่าระดับแสงที่ต่อกลไก LDR

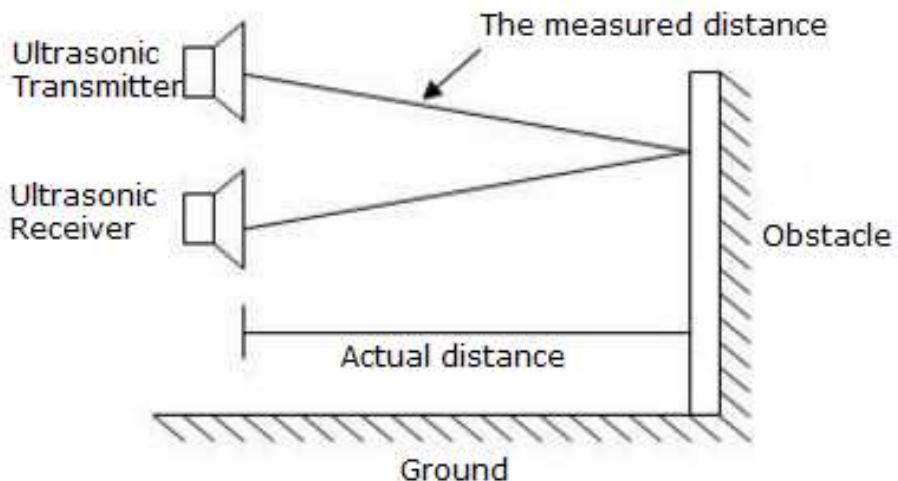
```
int LDRValue = 0;  
void setup() {  
    Serial.begin(9600);  
}  
void loop() {  
    LDRValue = analogRead(A0);  
    Serial.println(LDRValue);  
    delay(500);  
}
```

# การใช้งาน Water sensor

- ใช้หลักการการนำไฟฟ้าของน้ำ
- น้ำมาก → นำไฟฟ้าได้มาก → Higher output level



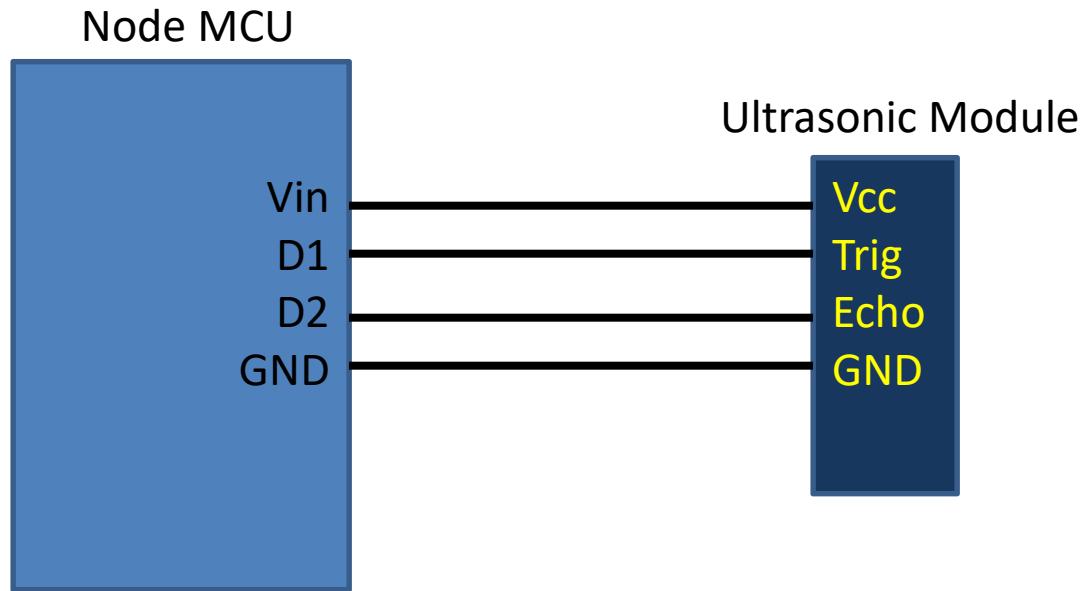
# การใช้งานอัลตราโซนิกเซนเซอร์



## Specifications:

- Working voltage: DC 5 V
- Static current: 3 mA
- Working temperature: 0 ~ + 70
- Output way: GPIO
- **Induction Angle: Less than 15**
- **Detection range: 2 cm to 4 m**
- **Detecting precision: 0.3 cm + 1%**
- Sensor size: Approx. 45 x 20 x 1.6mm

# Ultrasonic Module



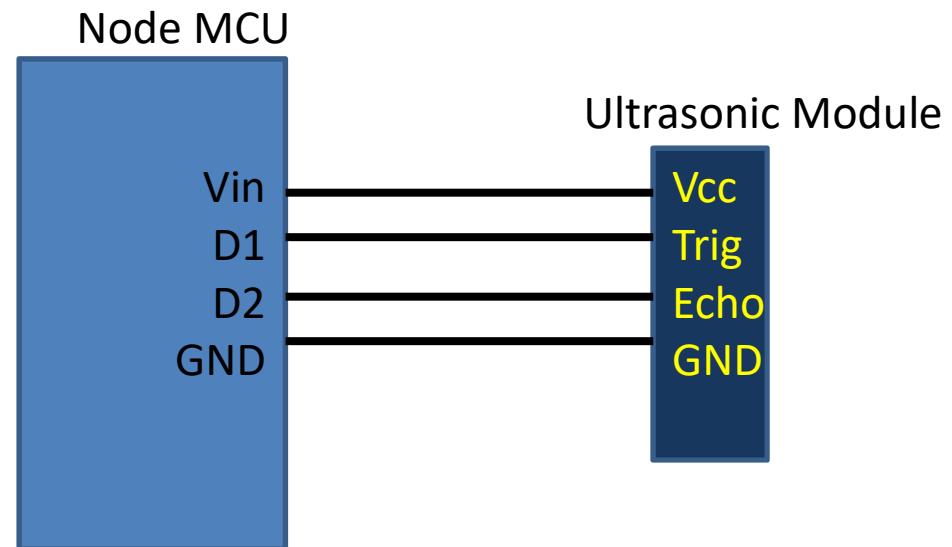
<https://github.com/watisleelapatra/Arduino>  
ไฟล์ชื่อ ultrasonic01.ino

```

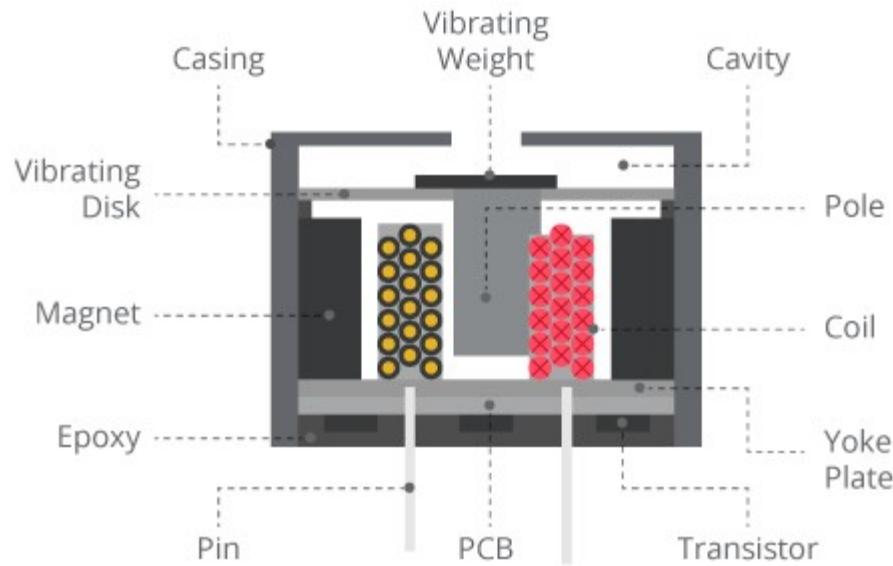
int pingPin = 5;
int inPin = 4;
void setup() {
    Serial.begin(9600);
}
void loop() {
    long duration, cm;
    pinMode(pingPin, OUTPUT);
    digitalWrite(pingPin, LOW);
    delayMicroseconds(2);
    digitalWrite(pingPin, HIGH);
    delayMicroseconds(5);
    digitalWrite(pingPin, LOW);
    pinMode(inPin, INPUT);
    duration = pulseIn(inPin, HIGH);
    cm = microsecondsToCentimeters(duration);
    Serial.print(cm);
    Serial.print("cm");
    Serial.println();
    delay(500);
}
long microsecondsToCentimeters(long microseconds)
{
    return microseconds / 29 / 2;
}

```

# Ultrasonic Module

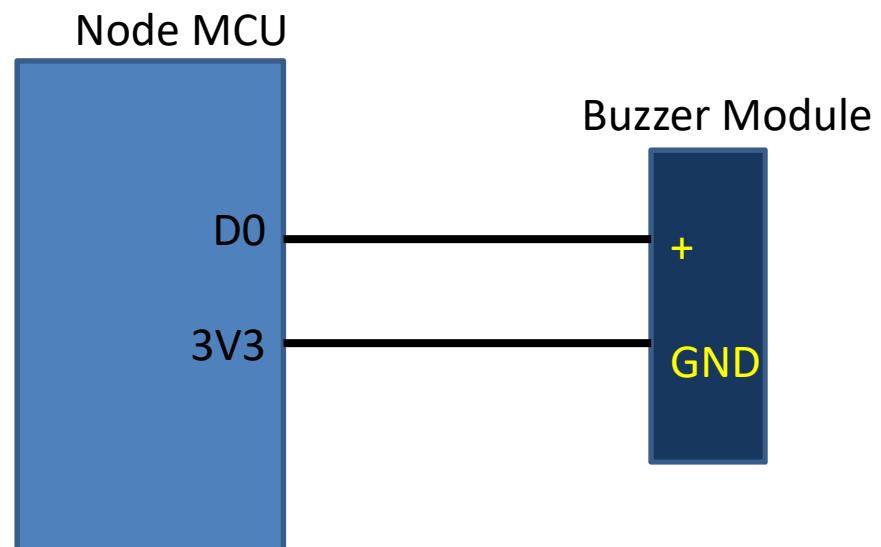


# การใช้งานลำโพง buzzer



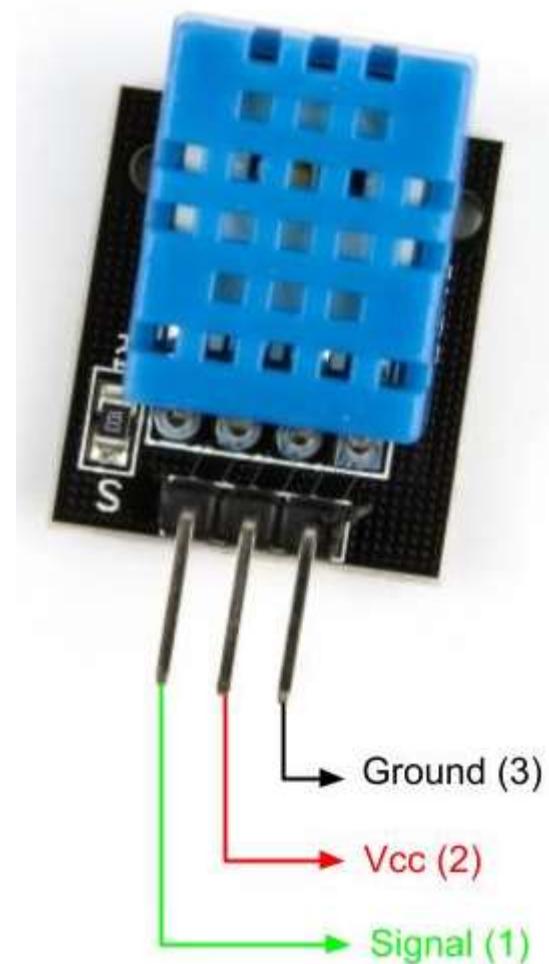
# การใช้งานลำโพง buzzer

```
int pinTone = 16;  
void setup() {  
    pinMode(pinTone, OUTPUT);  
}  
  
void loop() {  
    tone(pinTone, 1000);  
    delay(1000);  
    tone(pinTone, 2000);  
    delay(1000);  
}
```

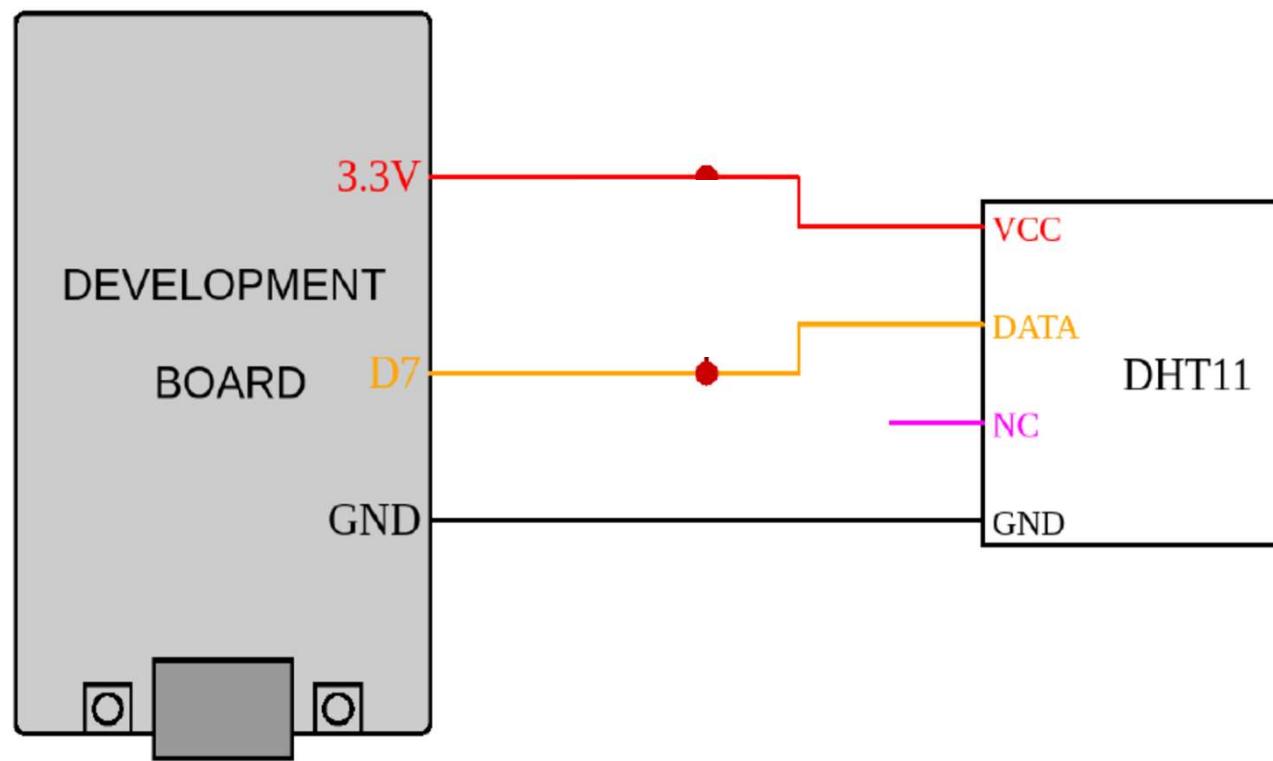


# เซนเซอร์อุณหภูมิและความชื้น (DHT11)

- วัดอุณหภูมิในช่วง 0-50 °C
- วัดความชื้นในช่วง 20-95%
- ทำงานที่แรงดัน 3.3V-5V

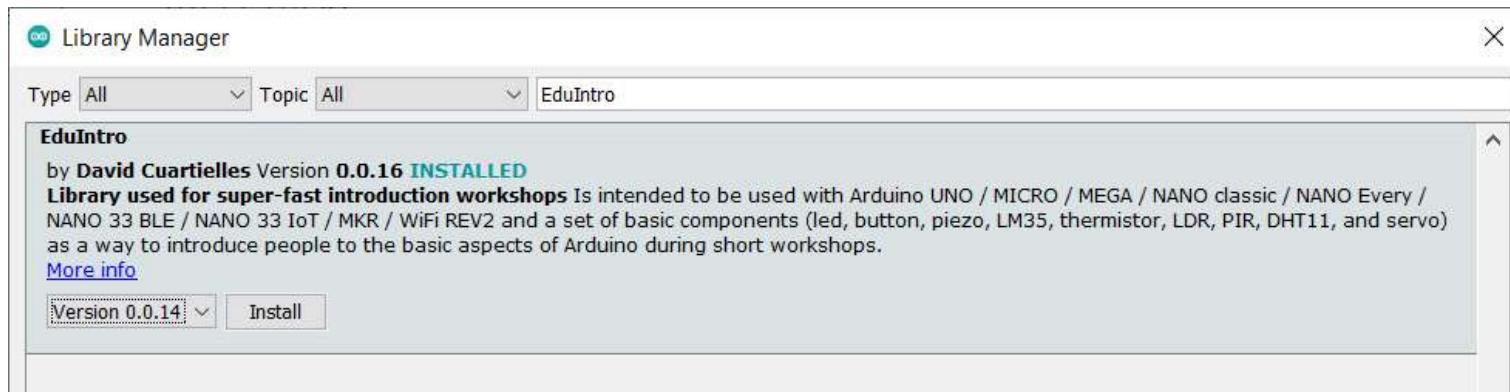


# เซนเซอร์อุณหภูมิและความชื้น (DHT11)



# เซนเซอร์อุณหภูมิและความชื้น (DHT11)

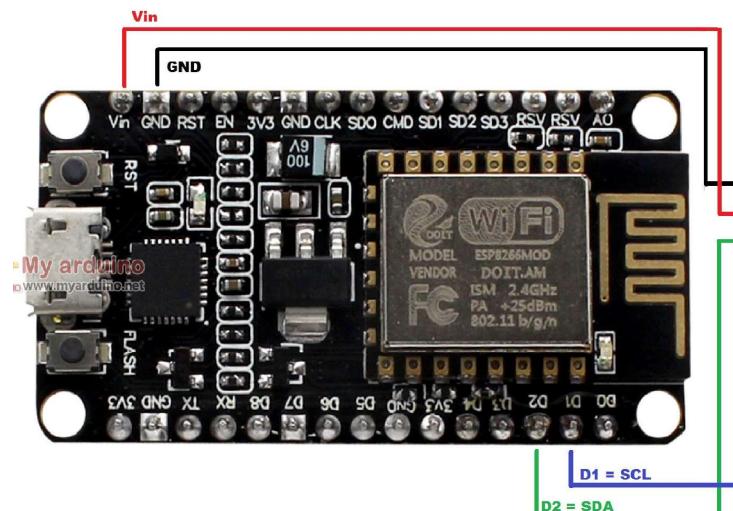
- ติดตั้ง Library ชื่อ EduIntro
- ไปที่เมนู Sketch → Include Library → Manage Libraries...
- พิมพ์ EduIntro ในช่องค้นหา จากนั้นกดปุ่มติดตั้ง



<https://github.com/watisleelapatra/Arduino>  
ไฟล์ชื่อ EduIntro\_DHT11.ino

# Liquid Crystal Display (LCD)

- แสดงตัวอักษรและตัวเลข (Alphanumeric) 2 บรรทัด x 16 ตัวอักษร
- เชื่อมต่อผ่าน I<sup>2</sup>C (SCL, SDA)
- ทำงานที่แรงดัน 5V (V<sub>USB</sub>)
- ต้องติดตั้ง Library (LiquidCrystal\_I2C) ก่อนจะใช้งาน



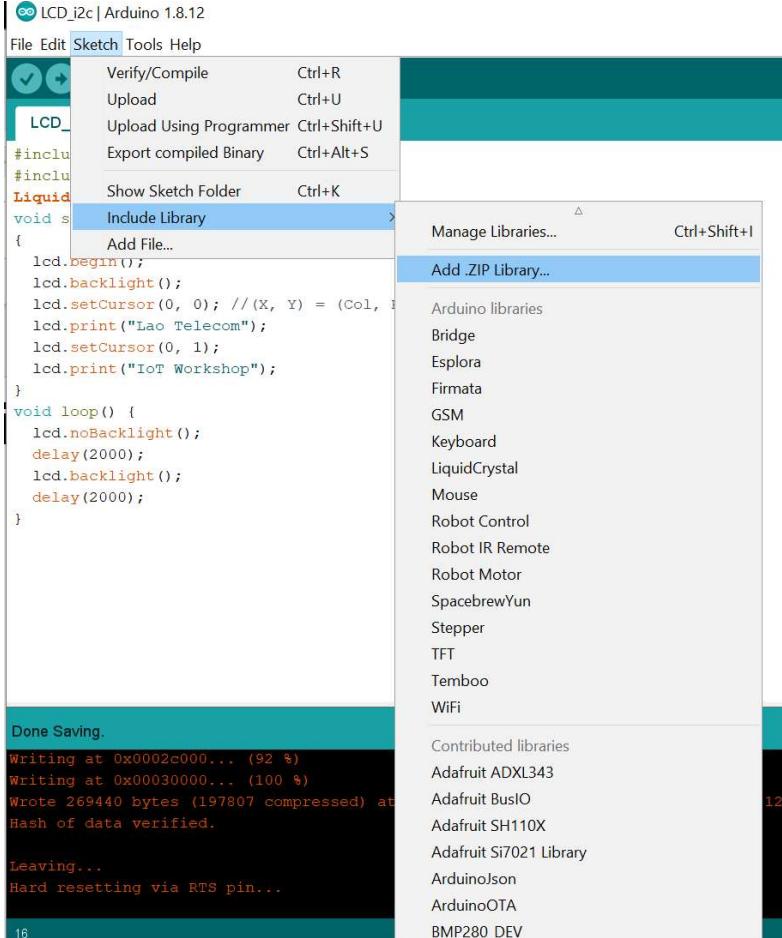
| LCD | NodeMCU          |
|-----|------------------|
| GND | GND              |
| Vcc | V <sub>USB</sub> |
| SDA | D2               |
| SCL | D1               |



# Liquid Crystal Display (LCD)

- ไปที่เมนู Sketch → Include Library → Add .ZIP Library...
- เลือก LiquidCrystal\_i2c.zip
- ตัวอย่าง LCD\_i2c.ino
- API อยู่ใน file

## LiquidCrystal\_I2C.cpp



```
LCD_i2c | Arduino 1.8.12
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
lcd.begin();
lcd.backlight();
lcd.setCursor(0, 0); // (X, Y) = (Col, Row)
lcd.print("Lao Telecom");
lcd.setCursor(0, 1);
lcd.print("IoT Workshop");
}
void loop() {
lcd.noBacklight();
delay(2000);
lcd.backlight();
delay(2000);
}

Done Saving.
Writing at 0x0002c000... (92 %)
Writing at 0x00030000... (100 %)
Wrote 269440 bytes (197807 compressed) at
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

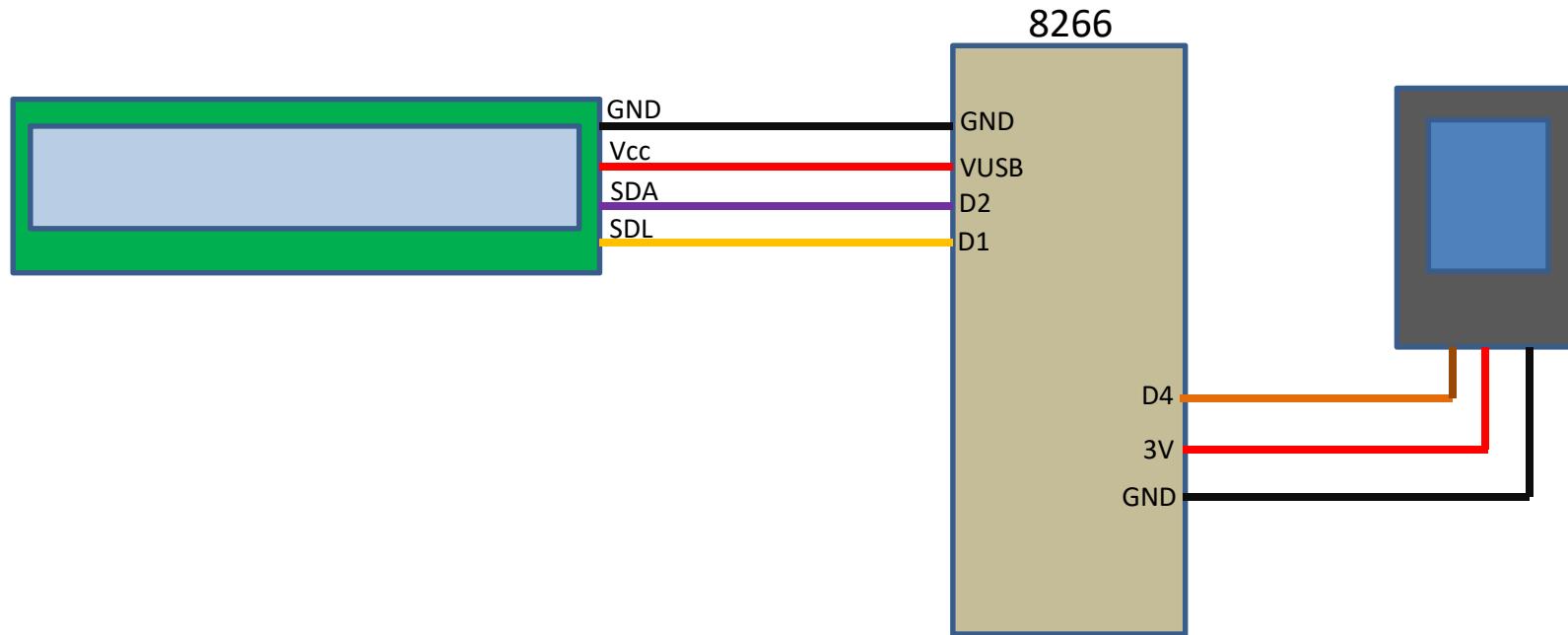
16
```

The screenshot shows the Arduino IDE interface with the sketch titled "LCD\_i2c". The "Sketch" menu is open, and the "Include Library" option is highlighted. A dropdown menu is displayed, with "Add .ZIP Library..." being the selected item. The main code area shows a simple LCD control program. The status bar at the bottom indicates "Done Saving." and "Hash of data verified." The page number "16" is visible at the bottom left.

# Liquid Crystal Display (LCD)

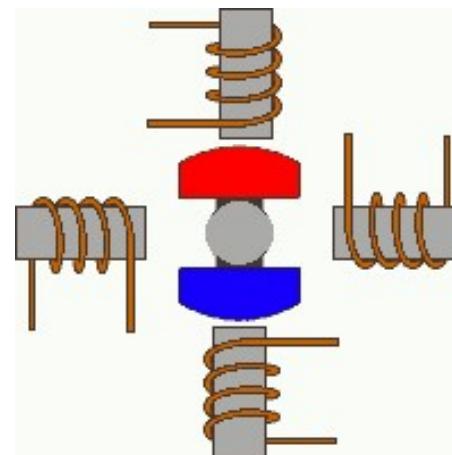
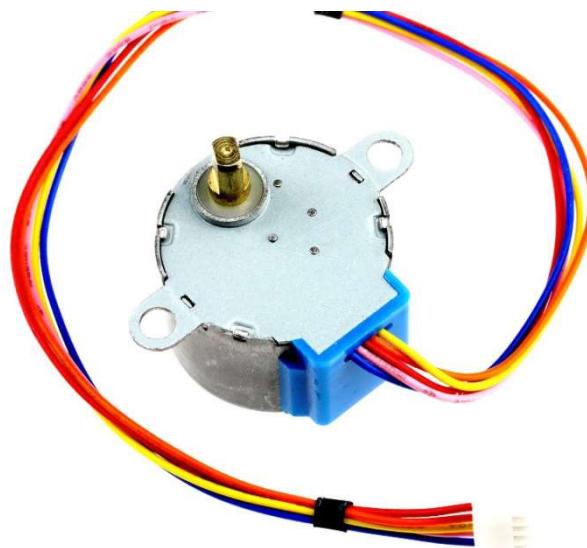
- ตัวอย่าง EduIntro\_LCD\_Temp.ino

| DHT11 | NodeMCU | LCD | NodeMCU |
|-------|---------|-----|---------|
| GND   | GND     | GND | GND     |
| Vcc   | 3V3     | Vcc | VUSB    |
| OUT   | D4      | SDA | D2      |
|       |         | SCL | D1      |



# Stepping Motor

- มอเตอร์ที่สามารถควบคุมการหมุนเป็นองศาได้
- $5.625^\circ$  /step หรือ 64 steps/รอบ (Model 28BYJ-48)
- ต้องมีวงจรควบคุมพิเศษ



# Stepping Motor

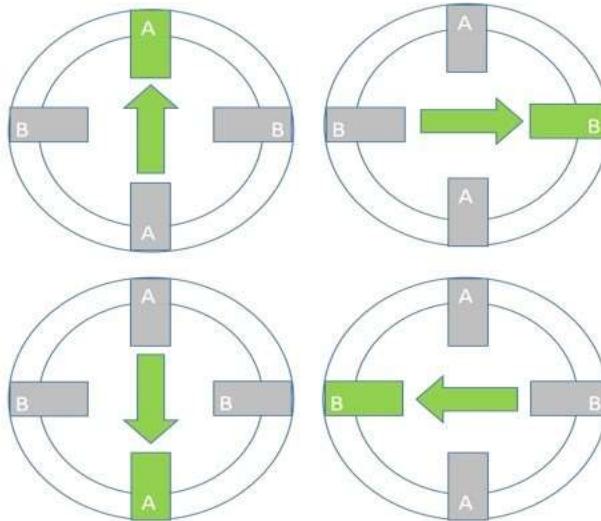


Fig 1 – One phase on – full step

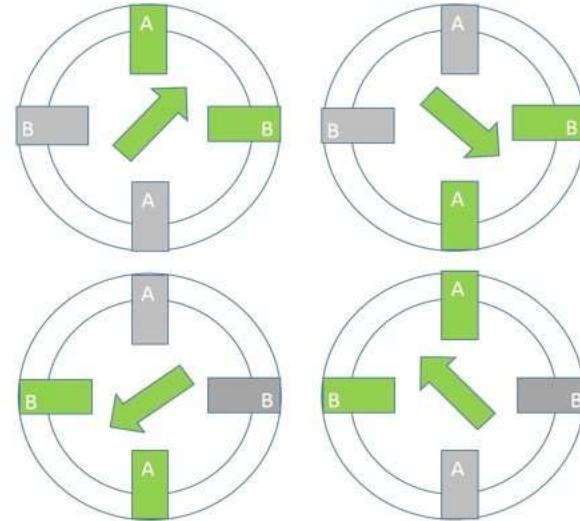


Fig2 – Two phase on – full step



SteppingMotor-1Phase.ino



SteppingMotor-2Phase.ino

# Stepping Motor

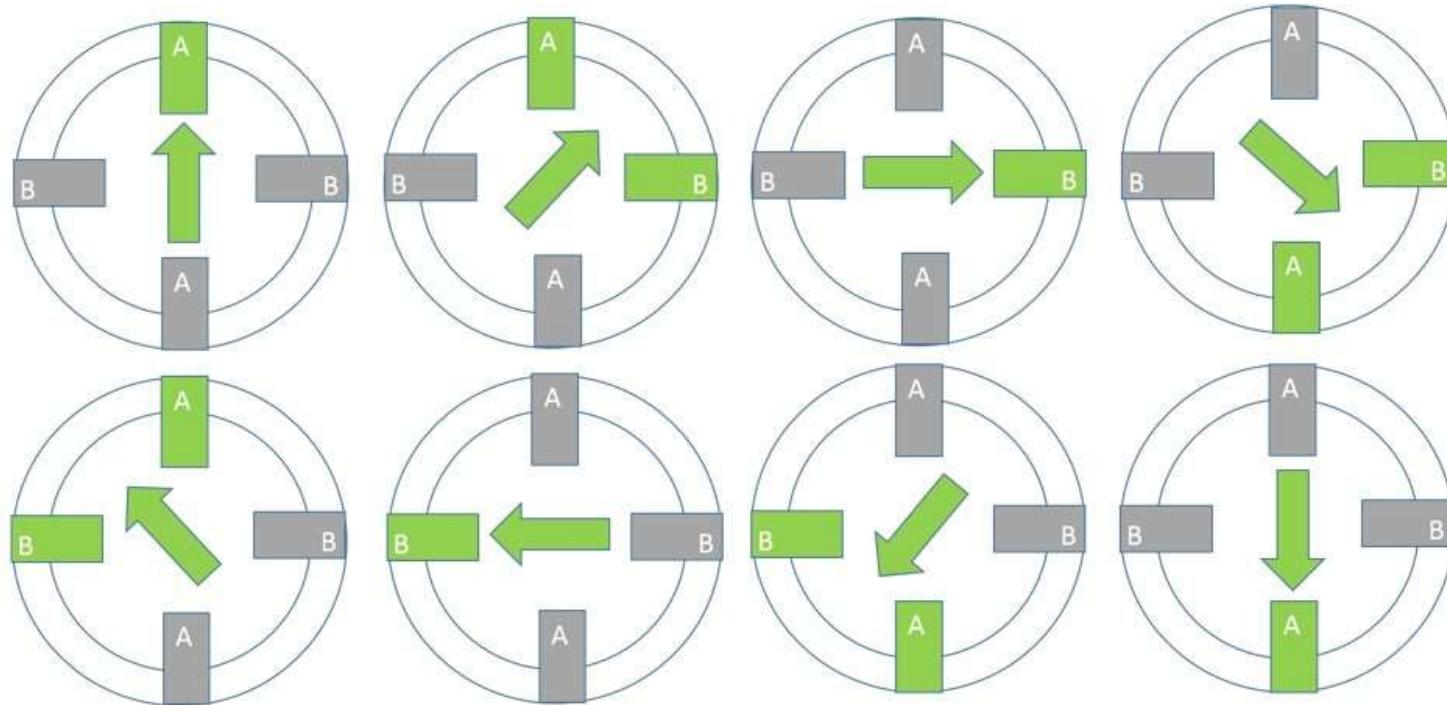


Fig3 - One-two phase on - half step



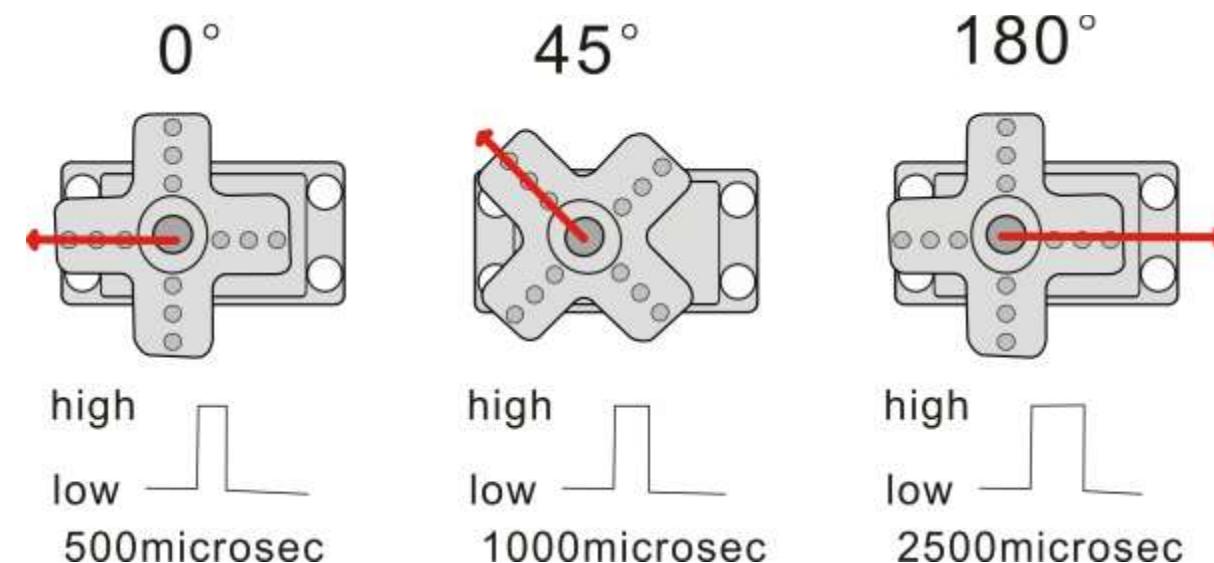
SteppingMotor-HalfStep.ino

# Servo Motor

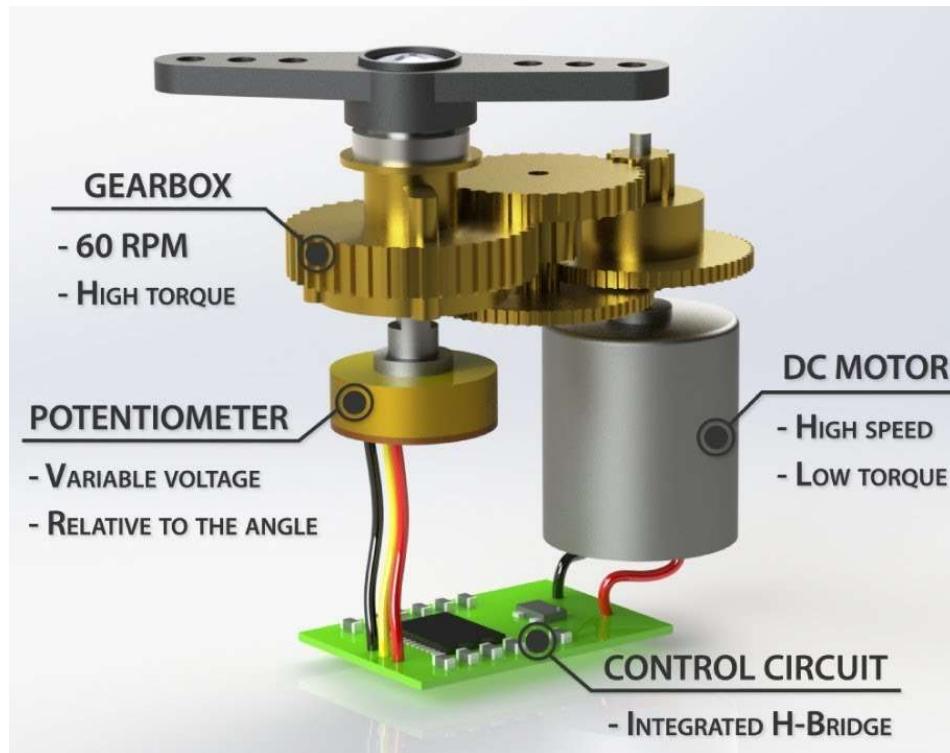


## Specification:

- No load speed: 0.12 seconds / 60 degrees (4.8V)
- Stall Torque: 1.6 kg·cm (0.15 Nm) @4.8V
- Operating temperature: -30 ~ +60 degrees Celsius
- Operating voltage: 4.8V-6V
- Working current: less than 500mA
- Weight: 9 grams

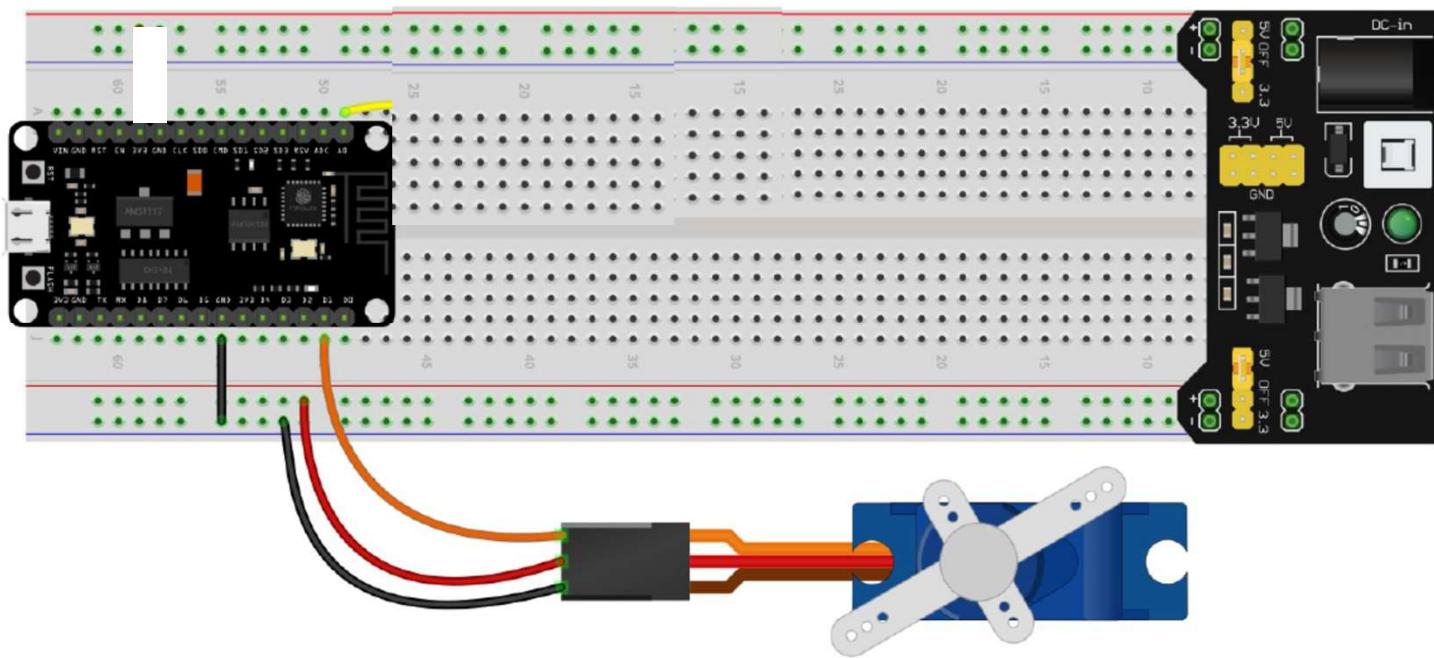
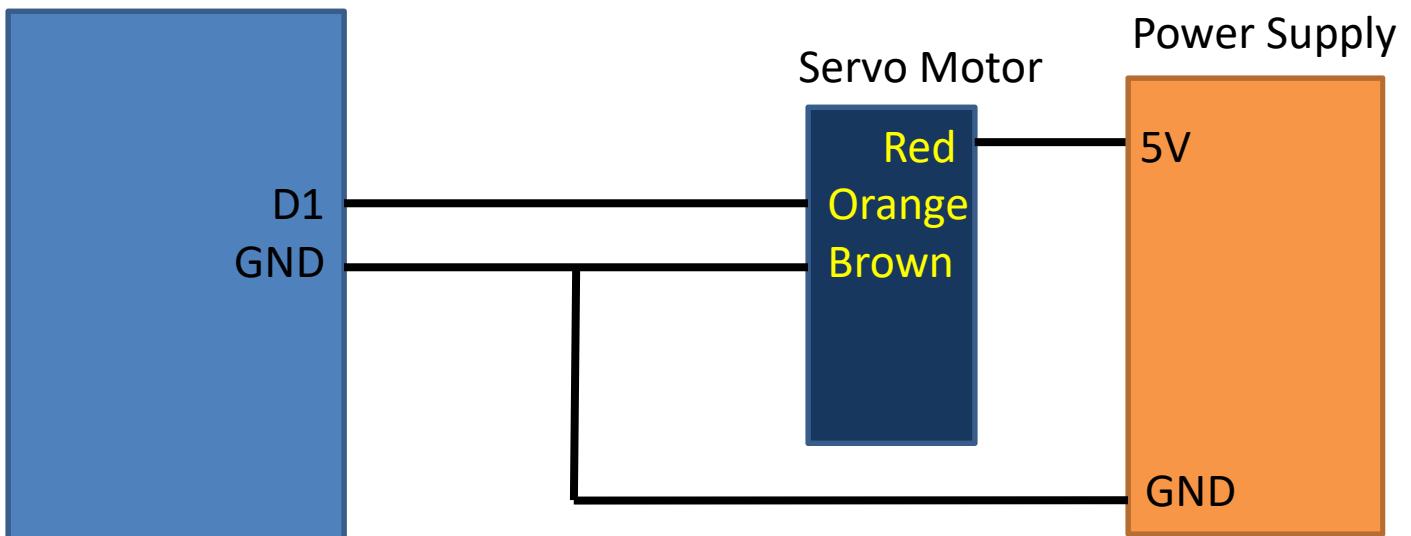


# Servo Motor



# Servo Motor

Arduino



# Servo Motor

```
#include <Servo.h>
Servo myservo;
void setup()
{
    myservo.attach(5);
}
void loop()
{
    myservo.write(0);
    delay(1000);
    myservo.write(90);
    delay(1000);
    myservo.write(300);
    delay(1000);
}
```

# DC Motor

