



01076412

การออกแบบและพัฒนาระบบงานไมโครคอนโทรลเลอร์
Microcontroller Application Design and Development
(Internet of Things)

NodeMCU, Digital Input, Digital Output, External
Interrupt, ADC, PWM

NodeMCU 1.0



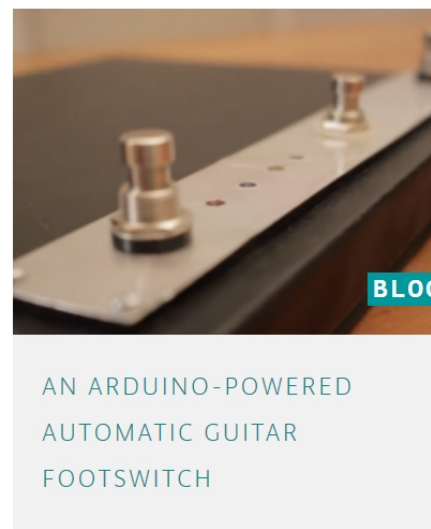
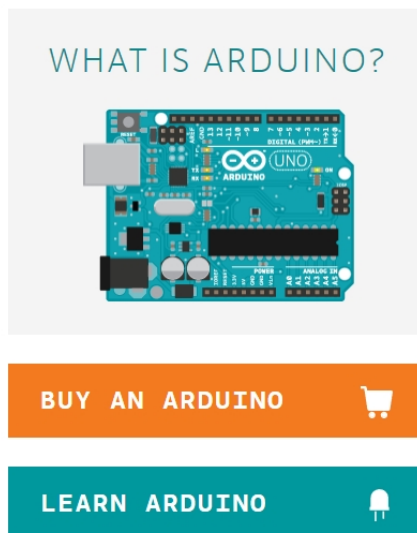
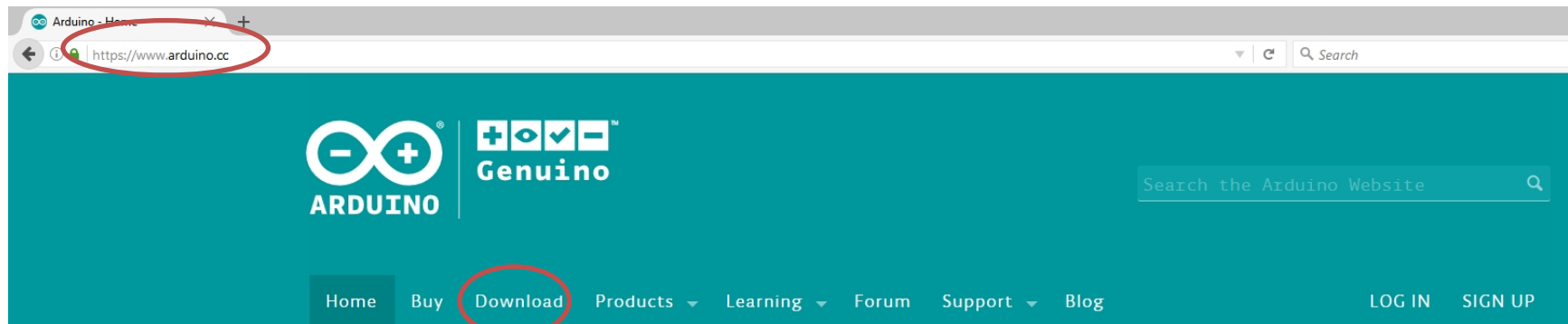
- ข้อมูลเชิงเทคนิค

- ใช้โมดูล ESP-12E (ESP8266 SoC chip) ของบริษัท AI Thinker
- ใช้ชิป Flash ความจุ 32Mbits (4MBytes)
- ใช้ชิป CP2102 ของ Silabs ทำหน้าที่เป็นส่วนเชื่อมต่อ USB-to-Serial
- มีขาสำหรับ SPI สำหรับต่อกับการ์ด SD
- ขา GPIO3/RXD0 และ GPIO1/TXD0 ต่อกับขา TXD และ RXD ของชิป CP2102 ตามลำดับ
- ขา GPIO13/RXD2 และ GPIO15/TXD2 (ใช้เป็นพอร์ต Serial เพิ่มอีกหนึ่งชุด)
- มีวงจรควบคุมแรงดัน 3.3V (@800mA max.) บนบอร์ด สามารถจ่ายแรงดันไฟเลี้ยง +5V จากภายนอกได้ (ต่อเข้าที่ขา VDD5V)
- ใช้คอนเนกเตอร์แบบ micro-USB สำหรับจ่ายแรงดันไฟเลี้ยง (VUSB) เท่ากับ +5V และสำหรับดาวน์โหลดเฟิร์มแวร์
- รับอินพุตแรงดันแบบแอนะล็อกสำหรับวงจร ADC (ขนาด 10 บิต) ที่อยู่ภายในชิป ผ่านวงจรแบ่งแรงดันด้วยตัวต้านทาน 100k / 220k (ลดแรงดันอินพุตจาก 0..3.3V ลงมาให้อยู่ในช่วง 0V..1V)

ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



- ดาวน์โหลด arduino จากเว็บ www.arduino.cc



ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



- ดาวน์โหลดตาม OS ที่ใช้งาน



ARDUINO 1.8.1

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. It runs on Windows, Mac OS X, and Linux. The environment is written in Java and based on Processing and other open-source software.

This software can be used with any Arduino board.
Refer to the [Getting Started](#) page for Installation instructions.

Windows Installer

Windows ZIP file for non admin install

Windows app

Mac OS X 10.7 Lion or newer

Linux 32 bits

Linux 64 bits

Linux ARM

[Release Notes](#)

[Source Code](#)

[Checksums \(sha512\)](#)

Connect, Collaborate, Create. [Learn more about the Create platform.](#)

**Try out the new
Arduino Web Editor**

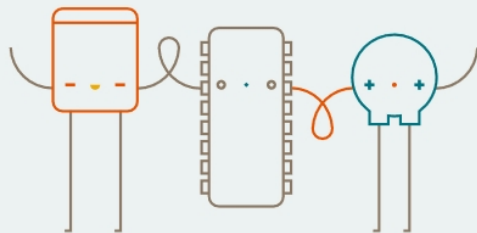
ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



- เลือก JUST DOWNLOAD และติดตั้ง

Support the Arduino Software

Consider supporting the Arduino Software by contributing to its development. (US tax payers, please note this contribution is not tax deductible). [Learn more](#) on how your contribution will be used.



SINCE MARCH 2015, THE ARDUINO IDE HAS BEEN DOWNLOADED **12,774,655** TIMES. (IMPRESSIVE!) NO LONGER JUST FOR ARDUINO AND GENUINO BOARDS, HUNDREDS OF COMPANIES AROUND THE WORLD ARE USING THE IDE TO PROGRAM THEIR DEVICES, INCLUDING COMPATIBLES, CLONES, AND EVEN COUNTERFEITS. HELP ACCELERATE ITS DEVELOPMENT WITH A SMALL CONTRIBUTION! REMEMBER: OPEN SOURCE IS LOVE!

\$3

\$5

\$10

\$25

\$50

OTHER

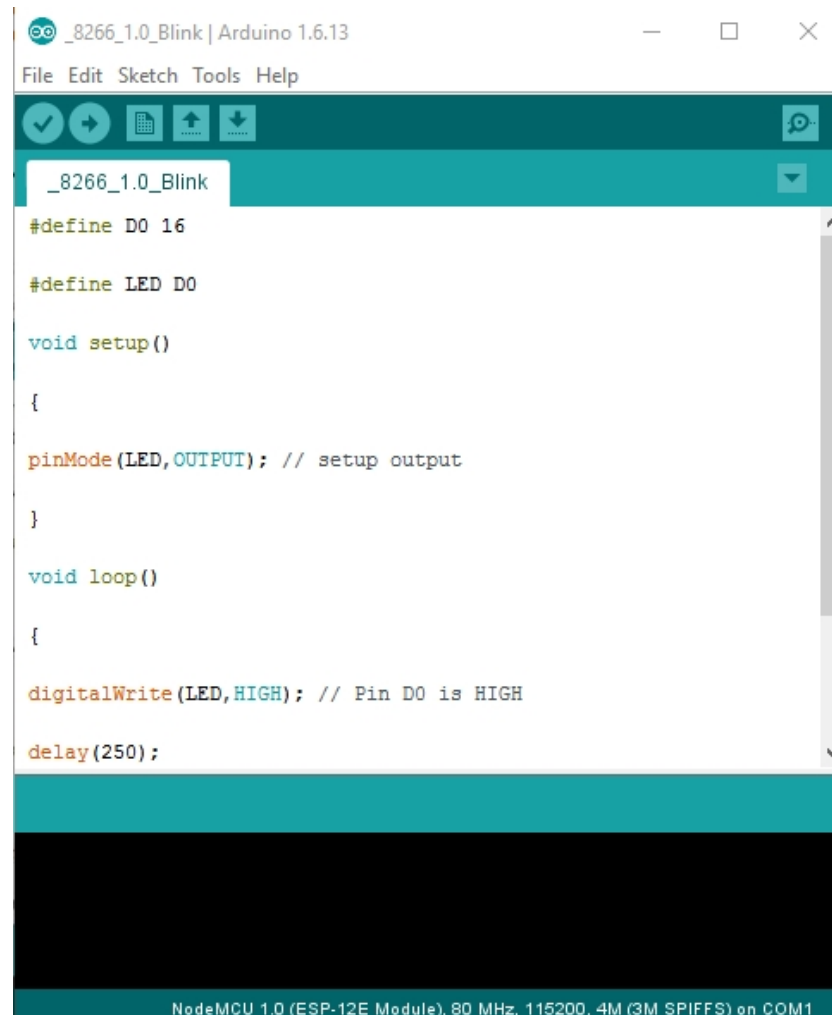
JUST DOWNLOAD

CONTRIBUTE & DOWNLOAD

ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



- เรียกขึ้นมาทำงาน



```
_8266_1.0_Blink | Arduino 1.6.13
File Edit Sketch Tools Help

[Icons]

_8266_1.0_Blink
#define D0 16

#define LED D0

void setup()
{
  pinMode(LED,OUTPUT); // setup output
}

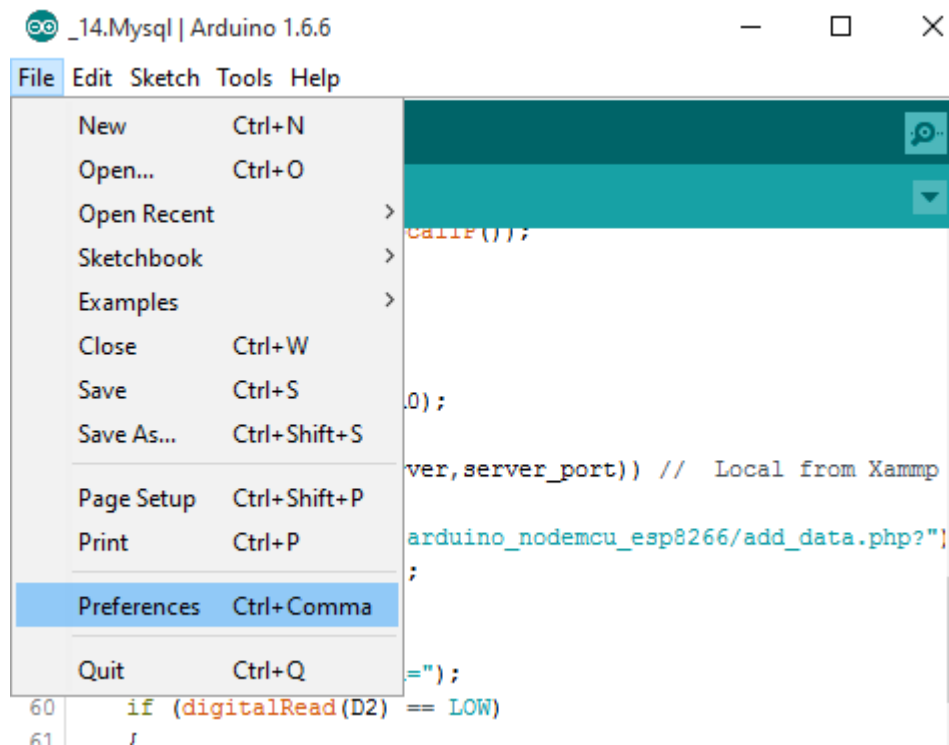
void loop()
{
  digitalWrite(LED,HIGH); // Pin D0 is HIGH
  delay(250);
}
```

NodeMCU 1.0 (ESP-12E Module), 80 MHz, 115200, 4M (3M SPIFFS) on COM1

ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



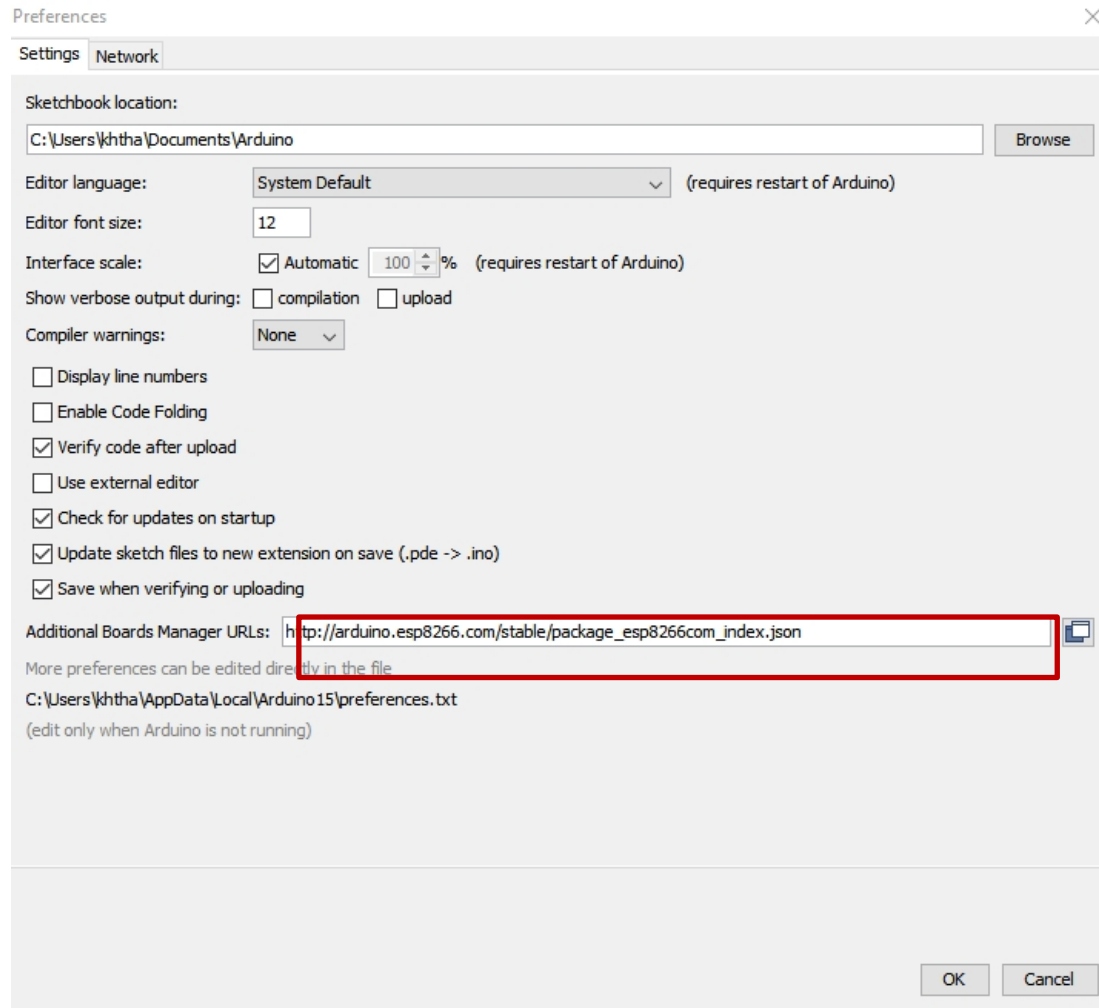
- เลือก File -> Preference



ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



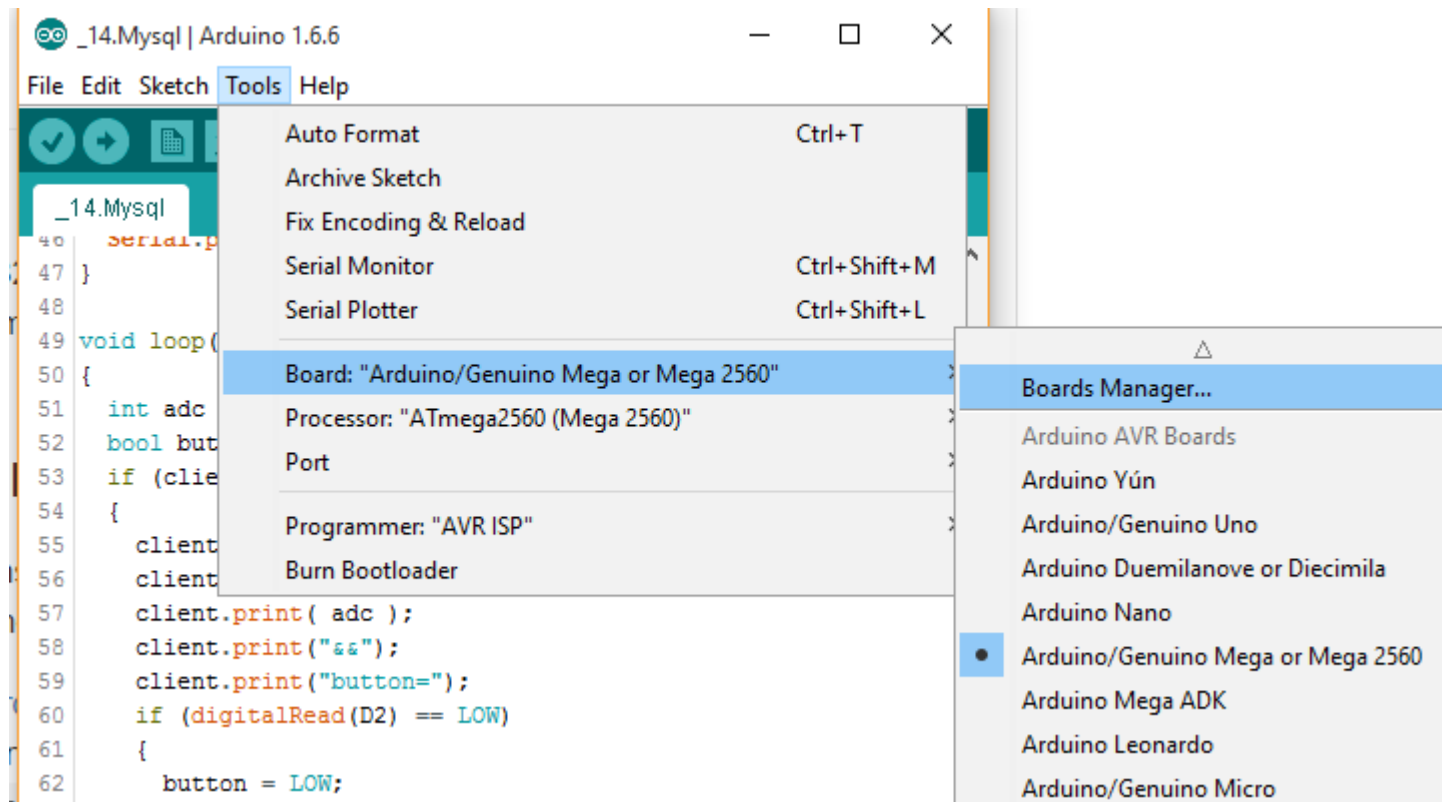
- http://arduino.esp8266.com/stable/package_esp8266com_index.json



ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



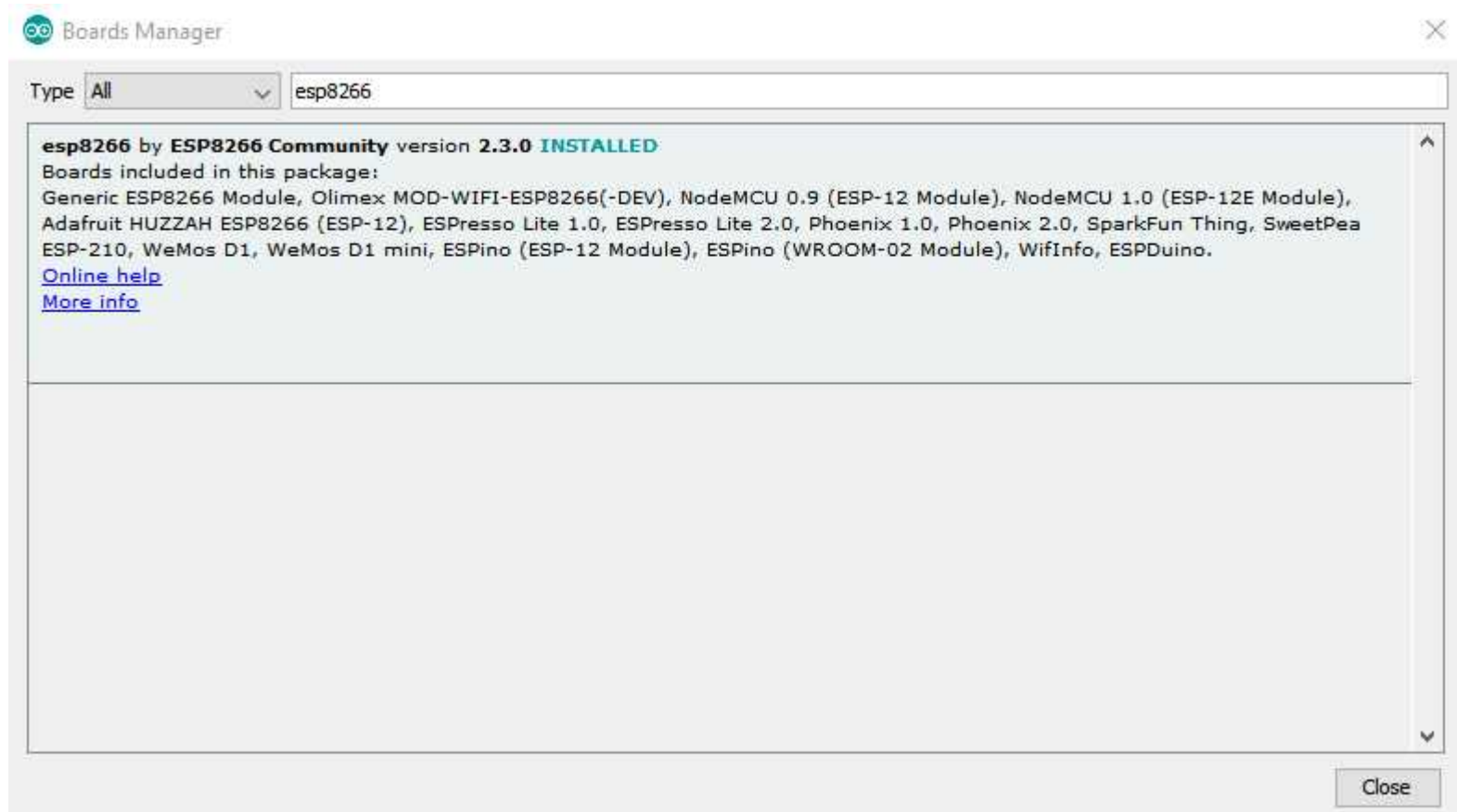
- เรียก Board Manager



ติดตั้ง Arduino เพื่อใช้กับ NodeMCU



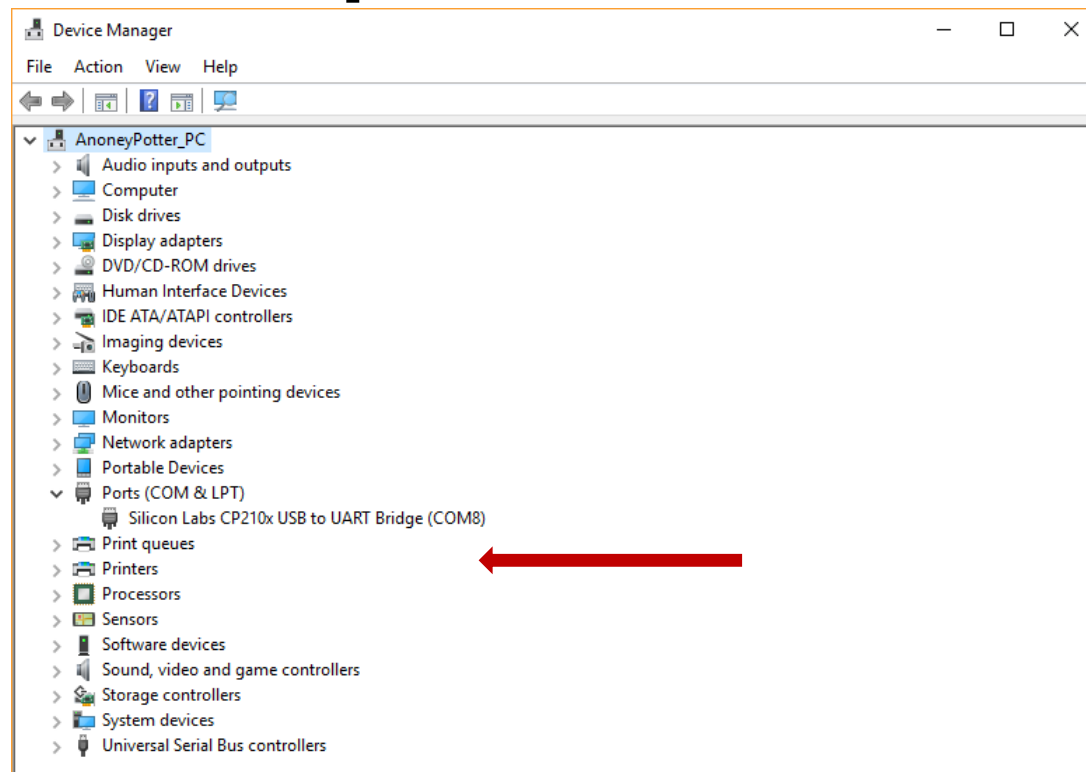
- ป้อนตัวกรอง esp8266 แล้วติดตั้ง
- ถ้าแสดงดังรูป แปลว่าติดตั้งเรียบร้อยแล้ว





ทดสอบการทำงาน

- เสียบ NodeMCU กับ USB ของเครื่อง PC
- เรียกดู Device Manager
- ถ้า Com Port ไม่ขึ้นตามรูป ให้ติดตั้ง Driver

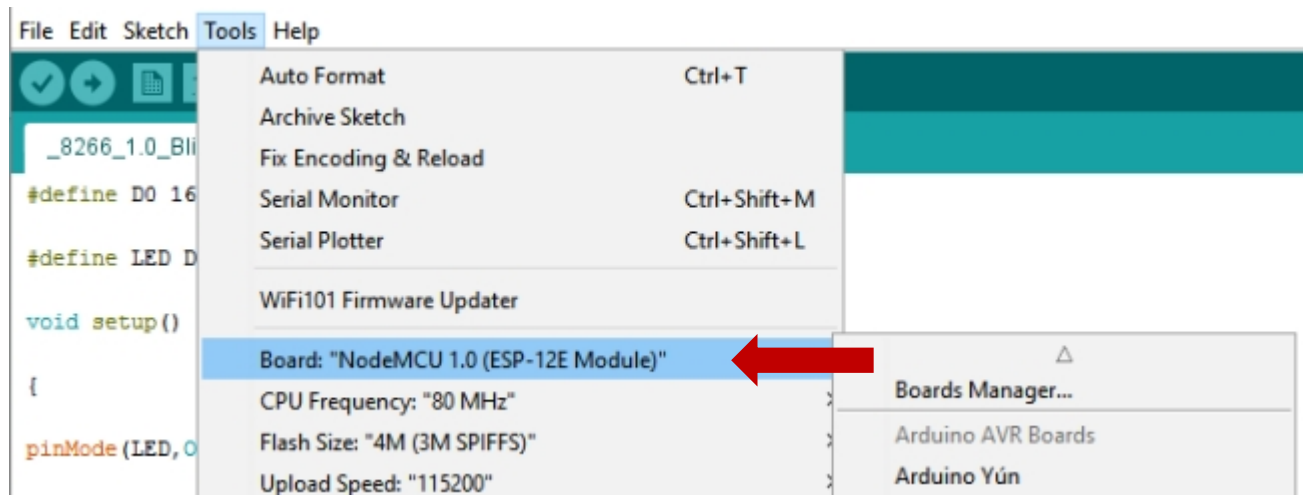


<https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>

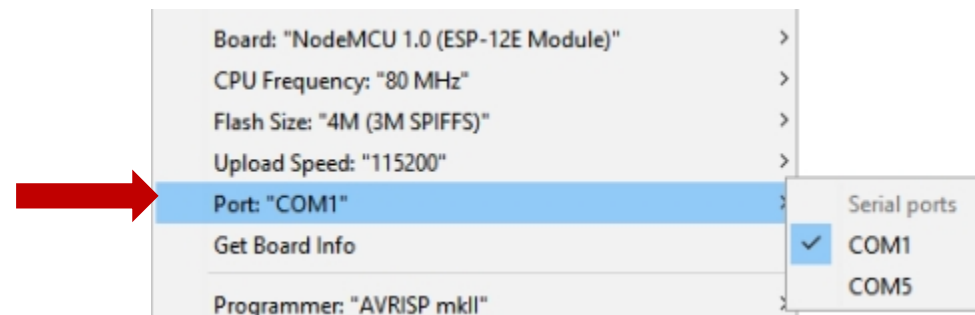
กำหนด Board และ Port



- ไปที่ Tools -> Board และเลือกบอร์ดเป็น NodeMCU 1.0



- เลือก Port เป็น Port ที่เพิ่มเข้ามาใหม่

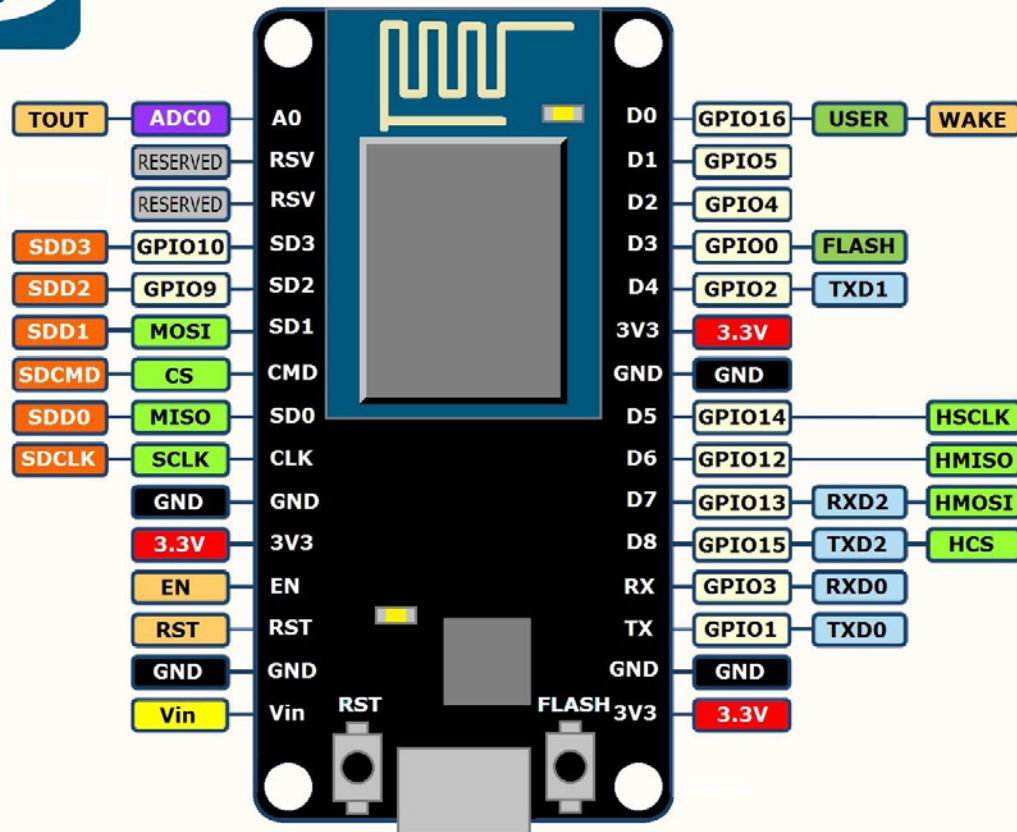


Pinout



NodeMCU ESP-12 development kit V1.0

PIN DEFINITION





PinMode

Syntax:

```
pinMode(pin, mode)
```

Parameter:

pin: the number of the pin whose mode you wish to set

mode: INPUT, OUTPUT or INPUT_PULLUP.



Digital Output

Syntax:

```
digitalWrite(pin, logic)
```

Parameter:

pin: the number of the pin whose mode you wish to set
logic : HIGH or LOW.

Blinky (ไฟกระพริบ)



```
#define D0 16 //Map NodeMCU pin to ESP8266 Module
#define LED D0 // Define LED pin
void setup()
{
    pinMode(LED,OUTPUT); // setup output
}
void loop()
{
    digitalWrite(LED,HIGH); // Pin D0 is HIGH
    delay(250);
    digitalWrite(LED,LOW); // Pin D0 is LOW
    delay(250);
}
```




Excercise

- ให้เขียนโปรแกรมไฟกระพริบ 1 วินาทีต่อครั้ง 3 ครั้ง และ 0.5 วินาทีต่อครั้ง 3 ครั้ง สลับกันไป



Digital Input

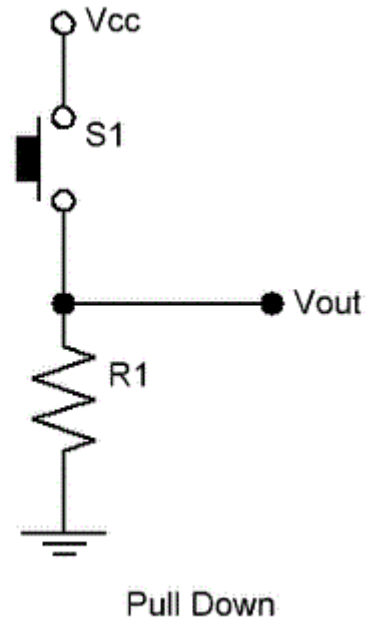
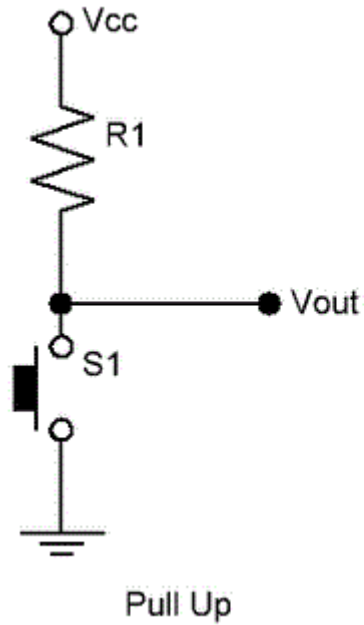
Syntax:

```
pinMode(pin, mode)
```

Parameter:

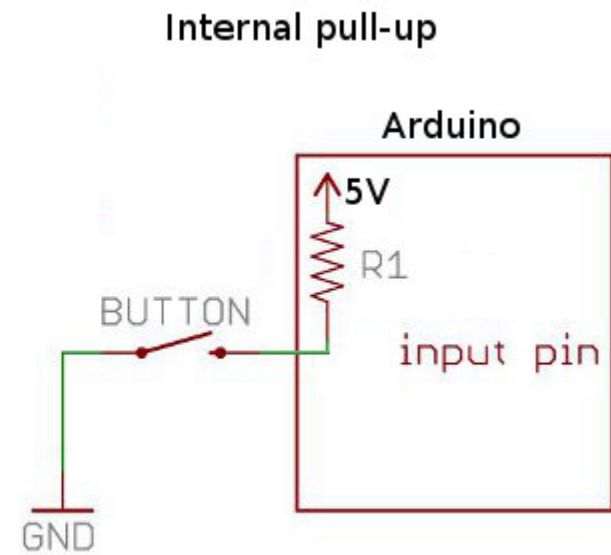
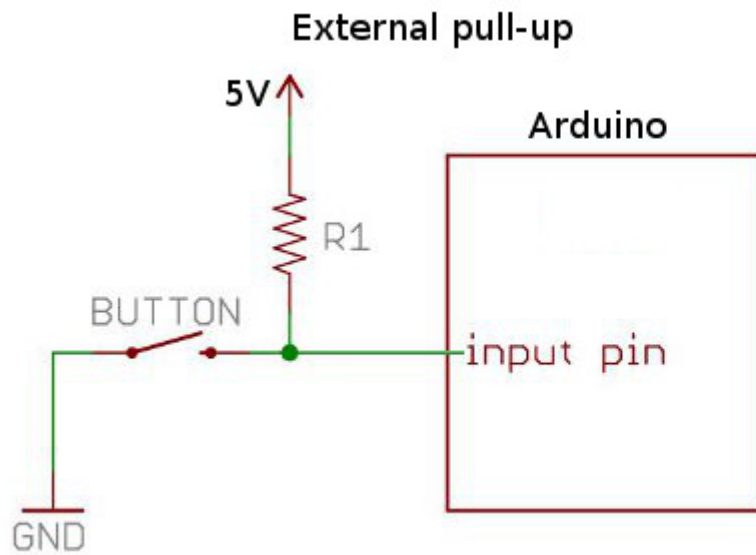
pin: the number of the pin whose mode you wish to set
mode: INPUT, OUTPUT or INPUT_PULLUP.

Pull Up or Pull Down





Input Pull Up





Digital Read

Syntax:

```
digitalRead(pin)
```

Parameter:

pin: the number of the pin whose mode you wish to set

Return:

HIGH: when the logic is HIGH

LOW: when the logic is LOW



Button Read

```
#define D2 4           // Map NodeMCU to ESP8266 pin
#define button D2
#define pressed LOW

void setup()
{
    Serial.begin(9600); pinMode(button, INPUT_PULLUP);
}

void loop()
{
    bool ReadSwitch = digitalRead(button);

    if(ReadSwitch == pressed)
    {
        Serial.println("Pressed Switch.");
        delay(500);
    }
}
```



Excercise

- เขียนโปรแกรมอ่านข้อมูลจากสวิตช์และแสดงใน Serial Monitor หากไม่มีสวิตช์ให้ใช้สายไฟจัมที่ Low แทน

การใช้งาน External Interrupt



Syntax:

```
attachInterrupt(interrupt, ISR, mode)
```

Parameter:

interrupt: attached to any GPIO pin, except GPIO16.

ISR (interrupt service routine) : call when the interrupt occurs; this function must take no parameters and return nothing.

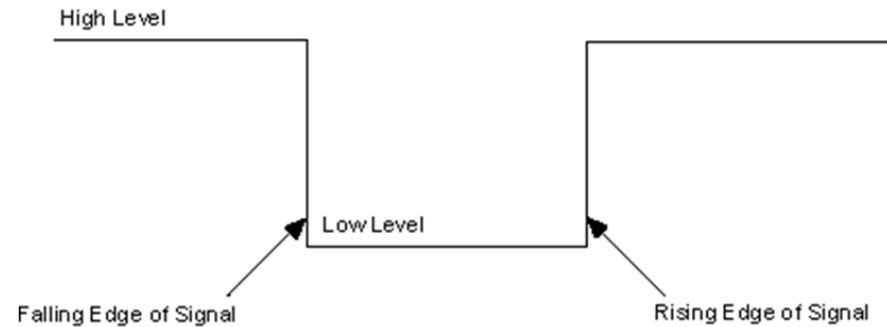
Mode: defines when the interrupt should be triggered.

LOW to trigger the interrupt whenever the pin is low.

CHANGE to trigger the interrupt whenever the pin changes value.

RISING to trigger when the pin goes from low to high.

FALLING for when the pin goes from high to low.



Program attach interrupt



```
// set pin numbers:
#define D0 16 // Map NodeMCU Pin to ESP8266 Pin
#define D2 4
#define buttonPin D2 // pushbutton pin
#define ledPin D0 // LED pin

void setup()
{
    Serial.begin(115200); // initialize serial:
    pinMode(ledPin, OUTPUT);
    pinMode(buttonPin, INPUT_PULLUP);
    attachInterrupt(buttonPin, EXTI2_ISR, CHANGE);
}

void loop()
{
    delay(250);
}

void EXTI2_ISR()
{
    digitalWrite(ledPin, !digitalRead(ledPin));
    Serial.println("Interrupt from pin D2");
}
```



Exercise

- ใช้วงจรจาก Exercise ที่ผ่านมา แต่ให้ใช้ Interrupt แทน
- โดยกดปุ่ม 1 ครั้งให้ดับ กดอีกครั้งให้ติด



Analog to Digital Converter

Syntax:

```
analogRead(pin)
```

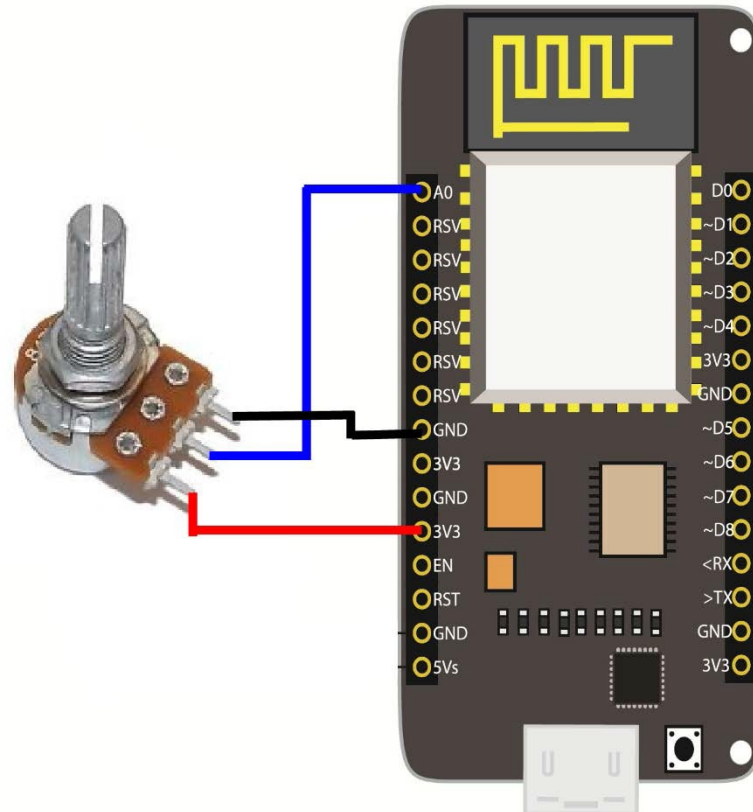
Parameter:

pin : the number of the analog input pin (A0)
in ESP8266 has only **A0** pin

Return:

int (0 to 1023) 10 bits

Example





Program : Analog Read

```
void setup()
{
    Serial.begin(115200);
}

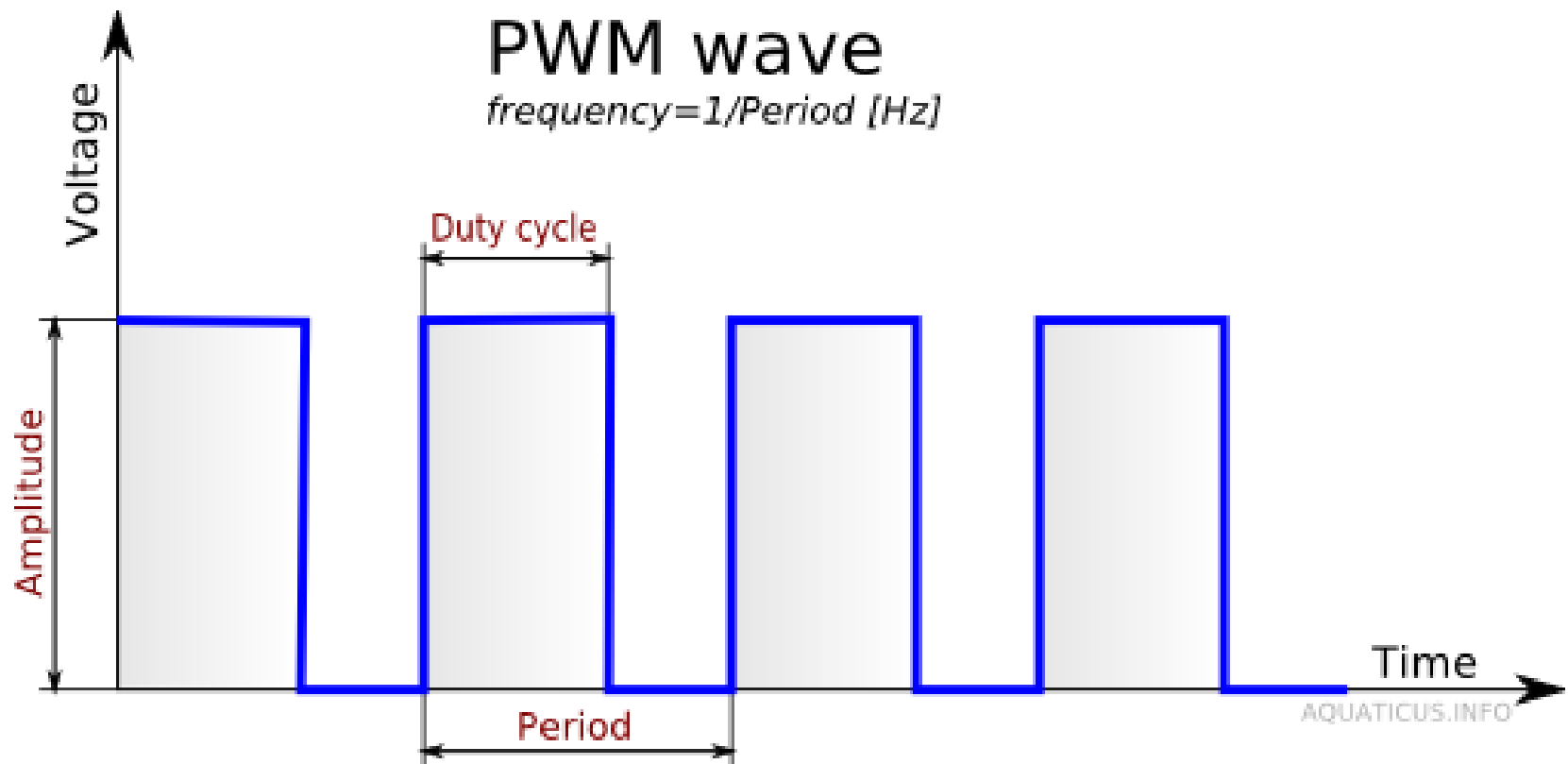
void loop()
{
    // read the input on analog pin 0:
    int sensorValue = analogRead(A0);

    Serial.print("ADC 10 bit = ");
    Serial.print(sensorValue);           // print out the value you read:

    float volts = 3.30*(float)sensorValue/1023.00;
    Serial.print(" , Voltage = "); Serial.print(volts,2);
    Serial.println(" V");
    delay(1);    // delay in between reads for stability
}
```

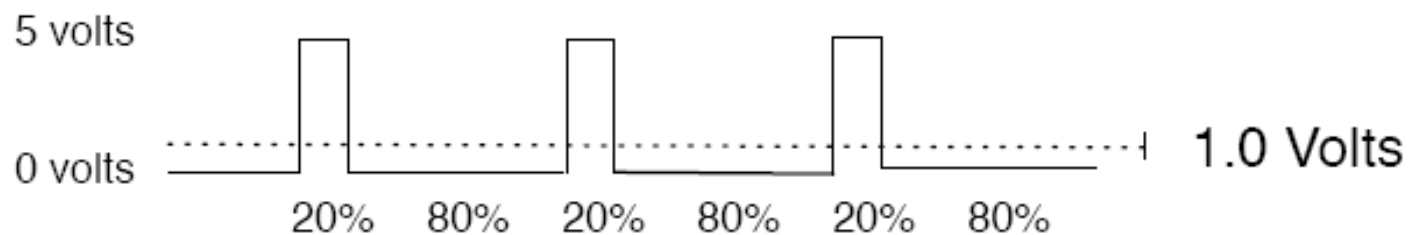
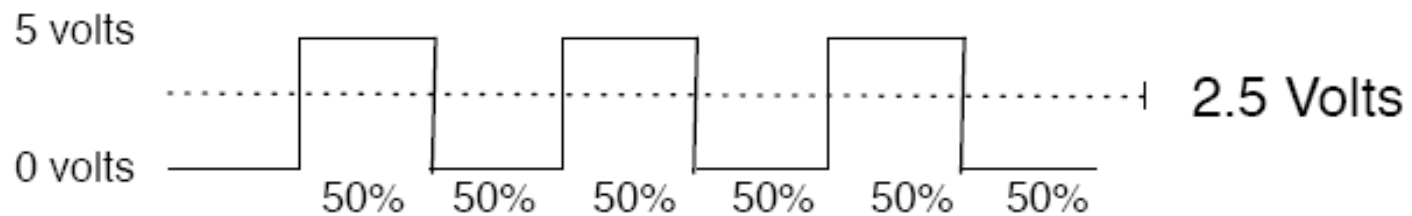
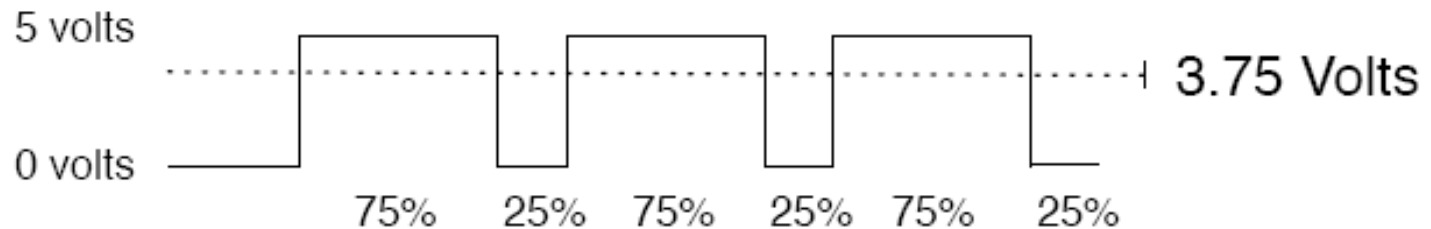


Pulse Width Modulation (PWM)





PWM – Adjust duty cycle





PWM

Syntax:

```
analogWrite(pin, duty)
```

Parameter:

pin: the pin to write to.

duty: duty cycle in 10 bits(0-1023) at default 1KHz.



PWM - Example

```
#define PWM_Pin 16 // LED on NODEMCU
```

```
#define PWM_off LOW
```

```
int duty = 512
```

```
void setup()
```

```
{
```

```
    pinMode(PWM_Pin,OUTPUT);
```

```
    digitalWrite(PWM_Pin,PWM_off);
```

```
}
```

```
void loop()
```

```
{
```

```
    analogWrite(PWM_Pin,duty);
```

```
}
```



PWM – Adjust Frequency

Syntax:

```
analogWriteFreq(freq)
```

Parameter:

freq : to change the frequency.



Exercise

- ให้แก้ไขโปรแกรม ใช้ PWM ให้ค่อยหรี่ LED จนดับ จากนั้นเพิ่มความสว่างจน LED สว่างเต็มที่ สลับกันไปเรื่อยๆ



PWM – Adjust Frequency

```
#define D0 16 // USER LED Wake
```

```
void setup()
```

```
{  
    pinMode(D0, OUTPUT);  
    digitalWrite(D0, LOW);  
    analogWrite(D0, 512);  
}
```

```
void loop()
```

```
{  
    for (int f = 250; f <= 1000; f++)  
    {  
        analogWriteFreq(f);  
        delay(5);  
    }  
}
```



Exercise

- ให้แก้ไขโปรแกรม ใช้ PWM โดยเปลี่ยน Frequency จาก 250 HZ ค่อยๆเพิ่มขึ้นไปถึง 1KHz และจาก 1 KHz ค่อยๆลดลงมาที่ 250 Hz



For your attention