

<https://www.facebook.com/lamloeicom>

คอร์สอบรม

Node32Lite

Arduino IDE Basic

ขั้นพื้นฐาน



จุดประสงค์

- ผู้เข้าอบรมสามารถเขียน Arduino IDE ลงบน Node32Lite
- สามารถสร้างระบบนำมำใช้งานได้

เอกสารคอร์สอบรมนี้ สามารถดาวน์โหลดได้ที่

- <https://github.com/lamloei/present2>

*** คอร์สอบรมนี้ หมายความว่า สมกับผู้เริ่มต้น ***



กำหนดการ

- 28 กพ, 1, 2 มีค พื้นฐาน Arduino IDE Node32Lite



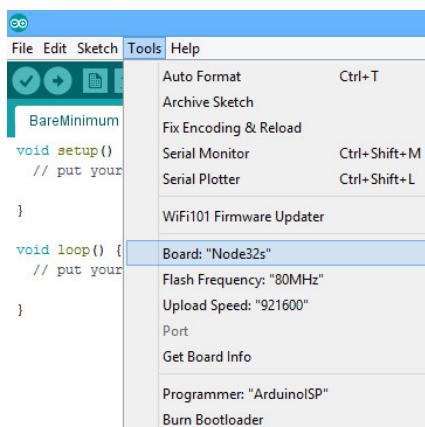
อุปกรณ์ และเครื่องมือ

- Node32Lite
- Micro USB Cable
- BreadBoard 830
- Blue LED
- Resistor
- Tact Switch
- Volume หรือ Potentiometer
- SHT30 เซ็นเซอร์วัดอุณหภูมิความชื้น
- สายต่อ



พื้นฐาน

Tools > Board: “Node32s”

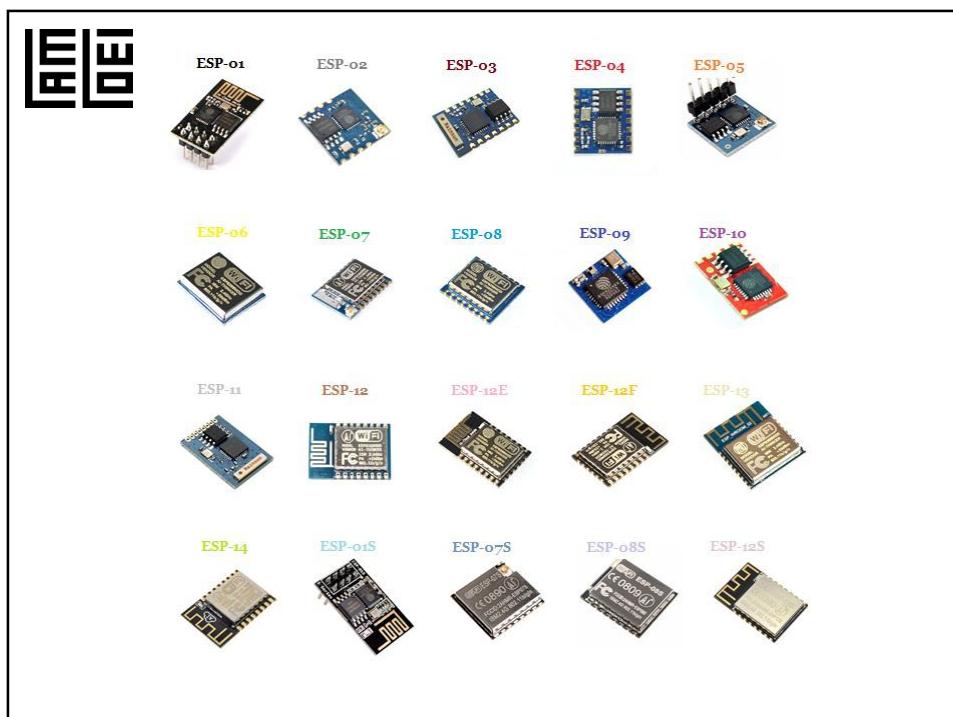


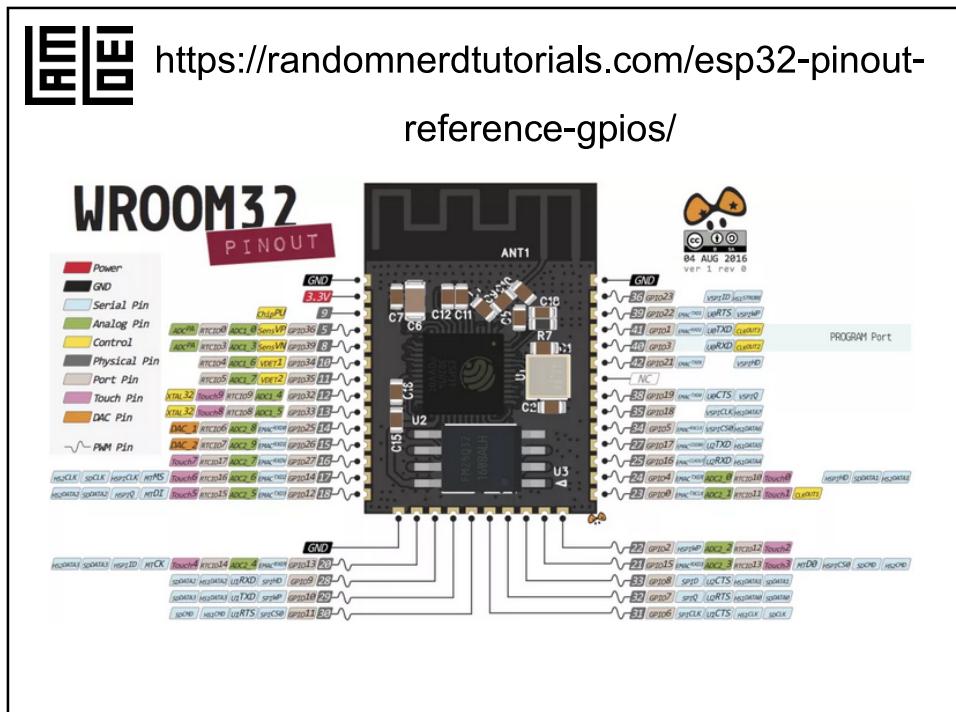
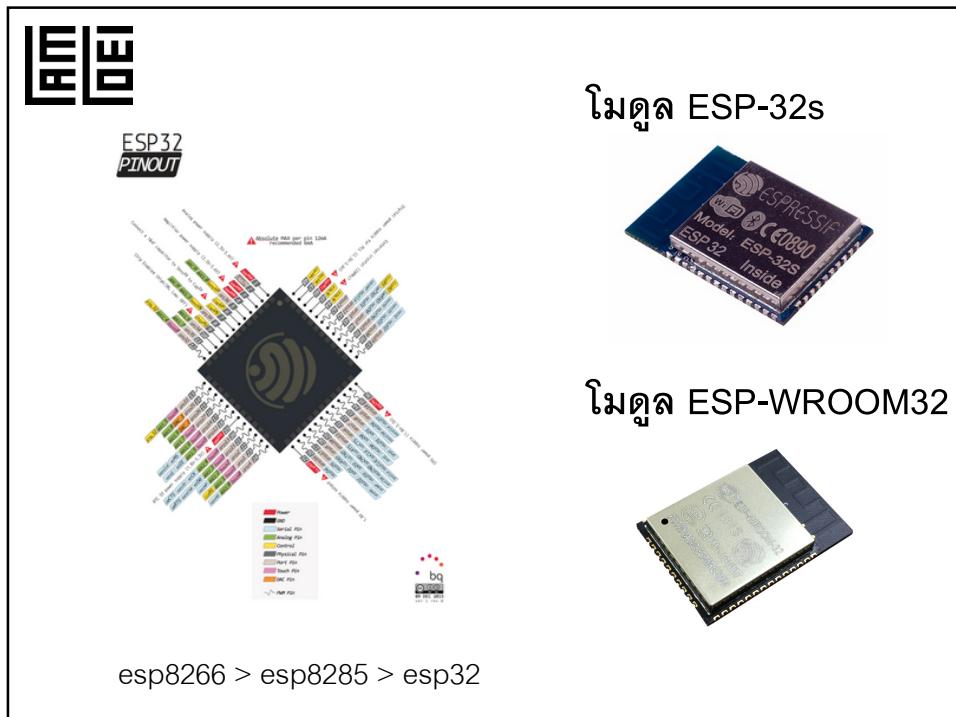
- * ต้องกดไฟมด้า
- เสียบสาย usb บอร์ด แล้วต่อเข้าคอม
- ต้องเห็น COM port ของบอร์ด
- ถ้ายังไม่ได้ติดตั้งโปรแกรมให้ข้ามไปติดตั้ง
- ที่ Arduino IDE ไปที่เมนู Tools

* จะต้องเห็น Board กับ Port

* Node32Lite ใช้ profile เดียวกับ Node32s

The screenshot shows the official website of Espressif Systems. At the top left is the Espressif logo, which consists of four stylized letters 'E' arranged in a 2x2 grid. To the right of the logo is the URL <https://espressif.com/>. The main header features the text 'ESP32 Wi-Fi + Bluetooth Combo Chip' and the subtitle 'An ultra low power and complete integration solution'. On the left side of the page is a vertical navigation menu with options: Products, Company, Ecosystem, Support, Documents, FAQ, and More. On the right side, there is a search bar and a 'Subscribe' button. Below the main header, there is a large image of a printed circuit board (PCB) with the ESP32 chip visible. A call-to-action button labeled 'Learn more >' is located at the bottom left of this image.





The screenshot shows the product page for the ESP32-DevKitC. At the top, there's a large image of the development board resting on a blue circuit board background. To the right of the image, the text "ESP32-DevKitC" is displayed in a large, bold font. Below the main image, there's a brief description: "Jump right into what matters. Start prototyping with our flagship SoC, ESP32. ESP32-DevKitC is a low-footprint, breadboard-friendly, minimum system development board which is powered by the ESP32-WROOM-32 module." The page includes standard navigation elements like a search bar and language selection.

The screenshot shows the Espressif website homepage with a banner announcing the achievement of shipping 100 million IoT chips. The banner features a large digital counter displaying "100 000,000" and the text "ESPRESSIF ships 100 million IoT chips". A "Learn more" button is visible at the bottom of the banner. On the left side, a vertical navigation menu is shown with options: Products, Company, Ecosystem, Support, Documents, and Contact Us. The background of the page is a dark image of Earth from space.

 longevity commitment

ESP8266 Series

- ESP8266EX - 12 years from January 1st, 2014
- ESP8266 modules - 12 years from January 1st, 2014
- ESP8266 dev kits - 12 years from January 1st, 2014
- ESP8285 - 10 years from January 1st, 2016



ESP8089 series

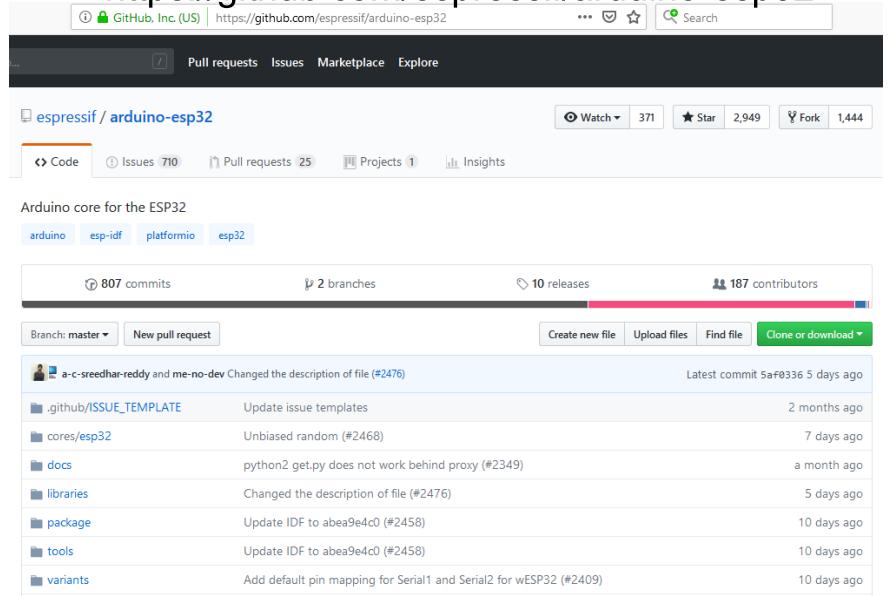
- ESP8089 - 12 years from January 1st, 2014

ESP32 Series

- ESP32 - 12 years from January 1st, 2016
- ESP32 modules - 12 years from January 1st, 2016
- ESP32 dev kits - 12 years from January 1st, 2016

<https://www.espressif.com/en/products/longevity-commitment>

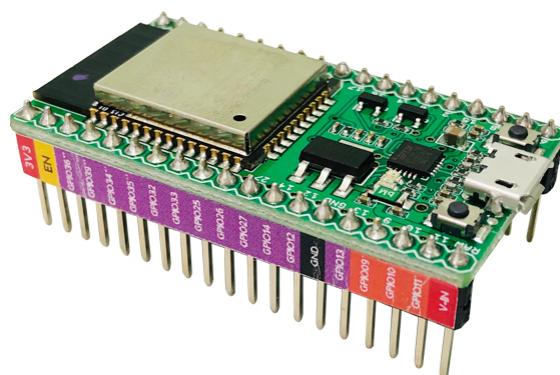
 <https://github.com/espressif/arduino-esp32>



The screenshot shows the GitHub repository page for the Arduino core for the ESP32. The repository has 807 commits, 2 branches, 10 releases, and 187 contributors. The latest commit was 5 days ago. The repository has 371 stars and 1,444 forks. The code tab is selected, showing tabs for arduino, esp-idf, platformio, and esp32. The issues tab shows 710 open issues. The pull requests tab shows 25 open pull requests. The projects tab shows 1 project. The insights tab is also visible.



Node32Lite



https://www.gravitechthai.com/product_detail.php?d=3318

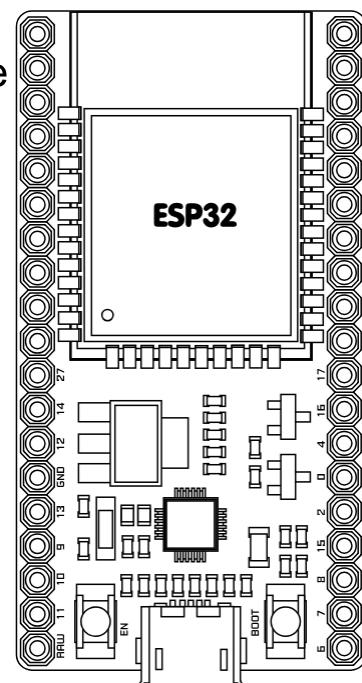
<https://www.cytron.io/p-node32-lite-wifi-and-amp;-bluetooth-development-kit>

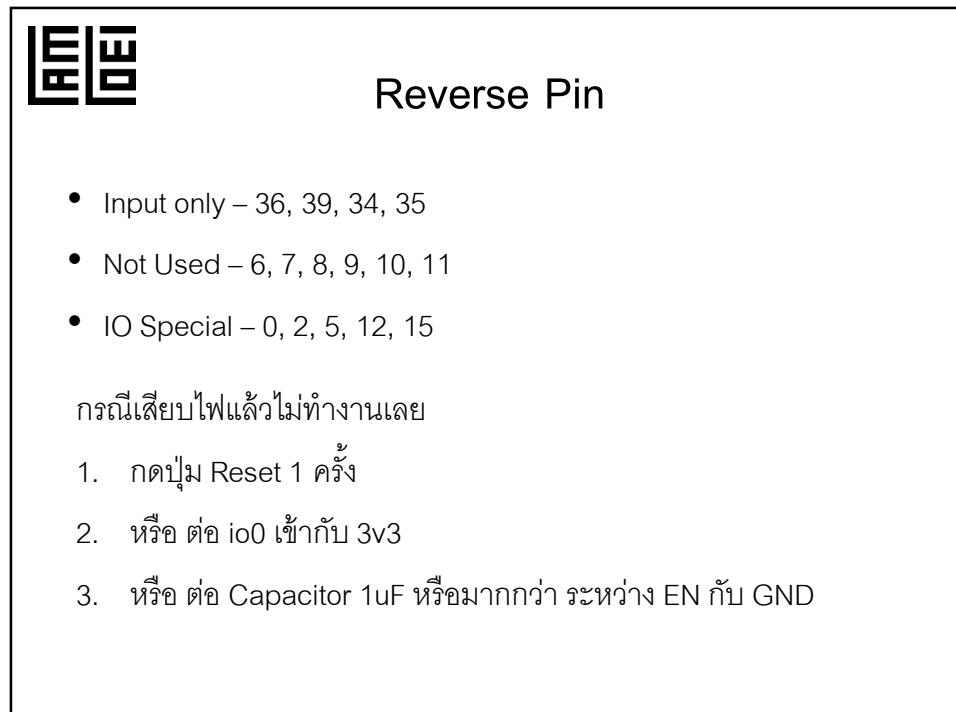
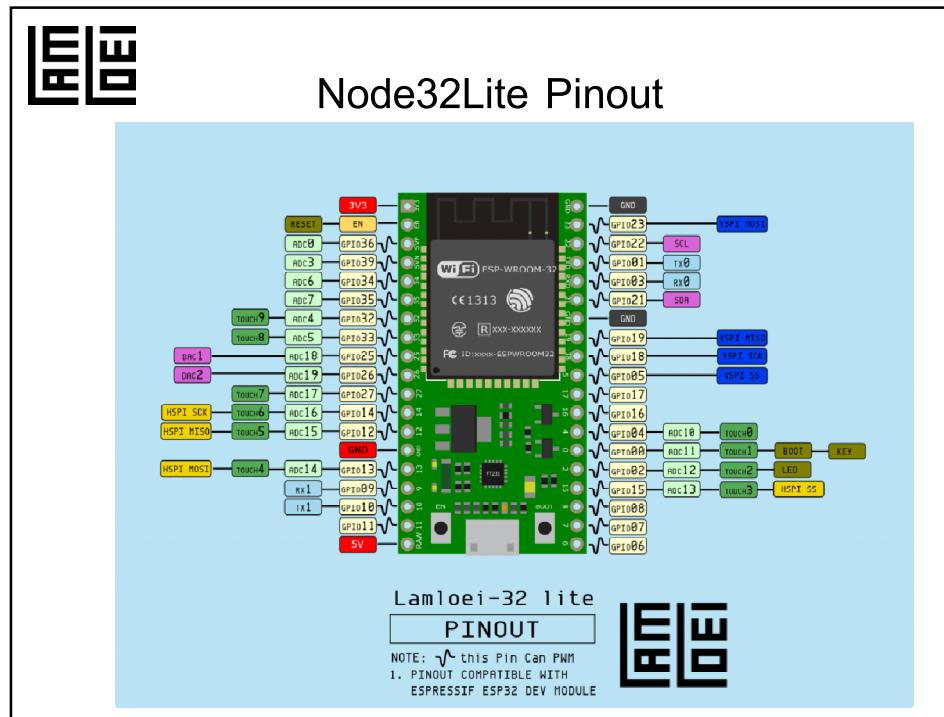


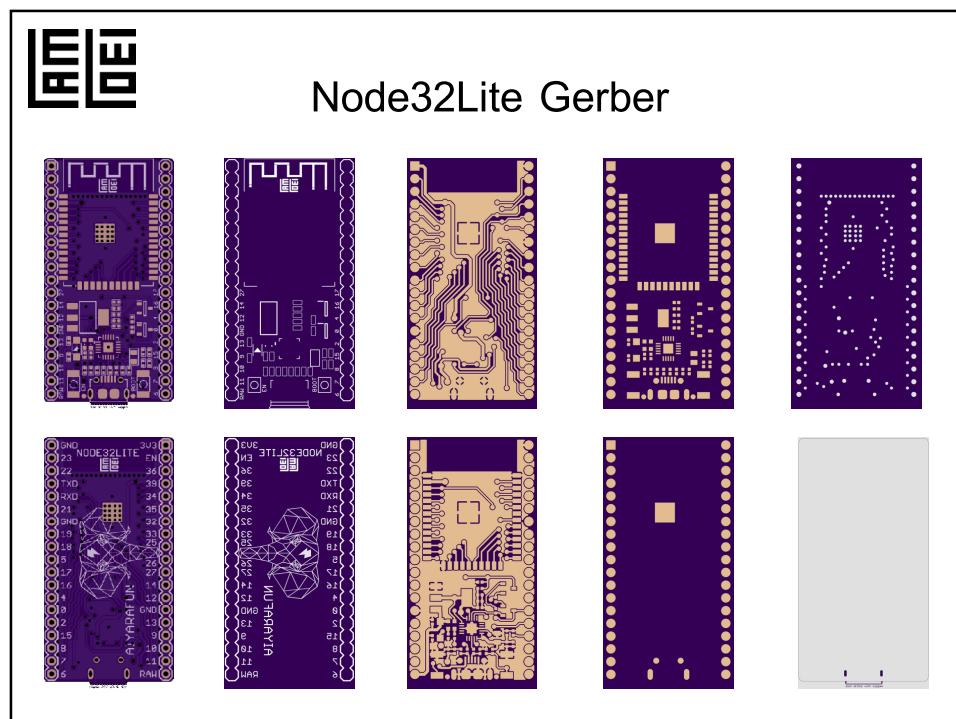
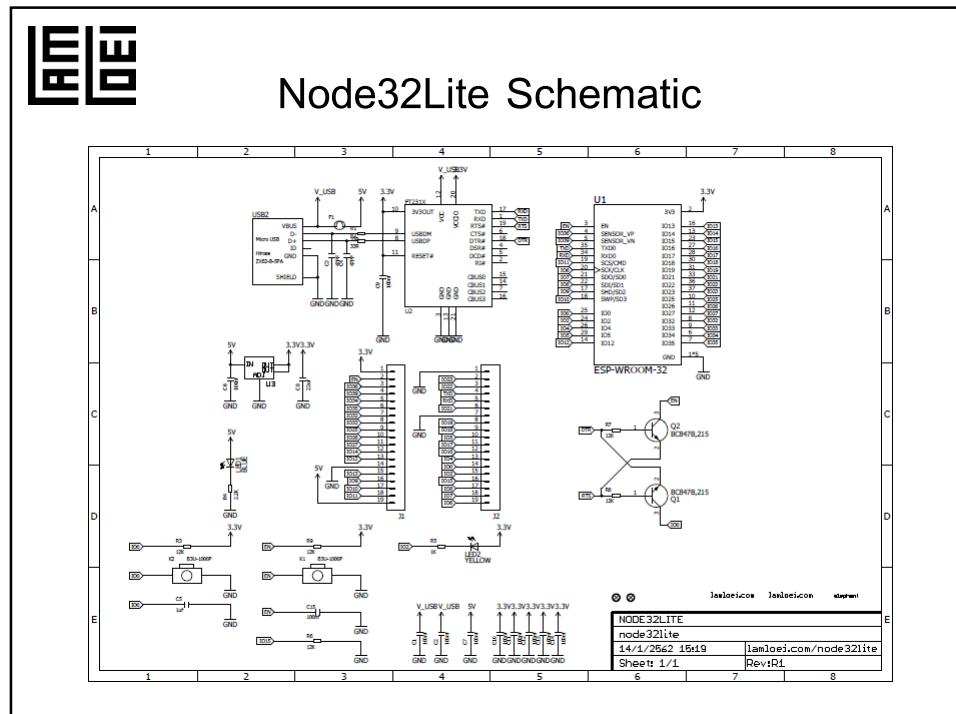
Node32Lite

Feature

1. ESP32-WROOM-32
2. FT231XQ-R
3. PTC FUSE 500mA
4. LM1117MPX-3.3
5. Micro USB
6. 0.9" * 19P Pin Header
7. Switch EN & BOOT
8. Led Blue Power
9. Led Yellow IO2

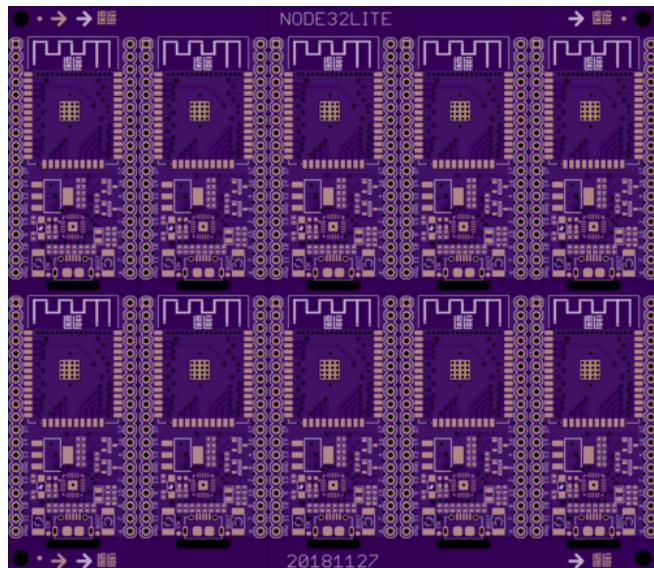








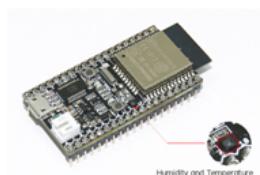
Node32Lite Gerber Panel



Node32 Family



Node32Pico



Node32s Plus



Node32s

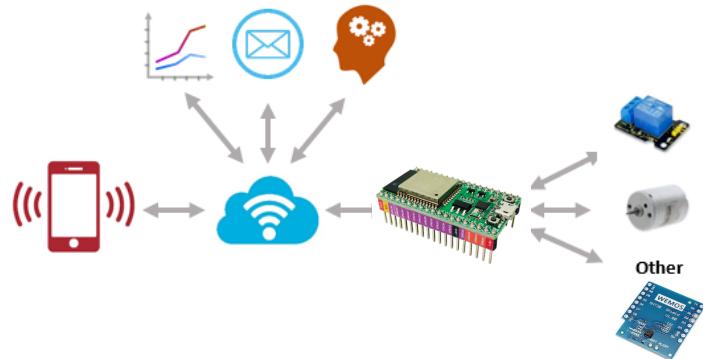


NodeWiFi

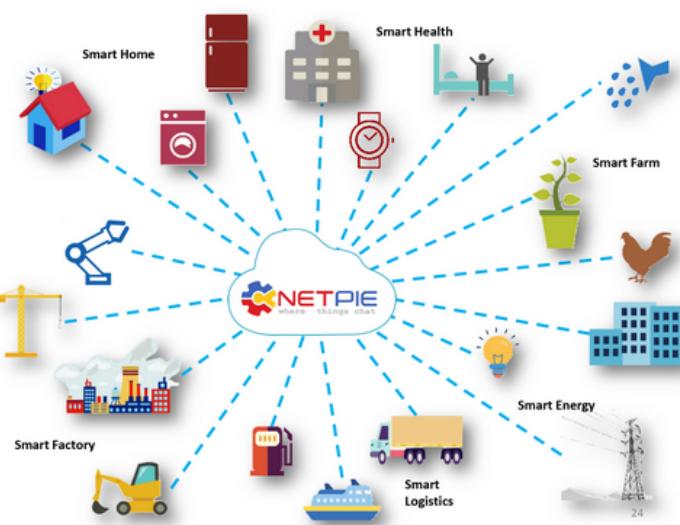


เปลี่ยน ค่าภายในภาพ > ตัวเลข > สร้างกราฟ

อีเมล แจ้งเตือน วิเคราะห์ ผ่านมือถือ



Smart System





แบบฝึกหัดที่ 1 - Smart System

1. ชื่อระบบ
2. อธิบายระบบ
3. ค่าที่ต้องการวัด – ควบคุม



ดาวน์โหลดและติดตั้งไฟล์ที่

<https://www.arduino.cc/en/Main/Software>

(ล็อก https://www.arduino.cc/en/Main/Software)

HOME STORE SOFTWARE EDU RESOURCES COMMUNITY HELP

Download the Arduino IDE



ARDUINO 1.8.8

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.

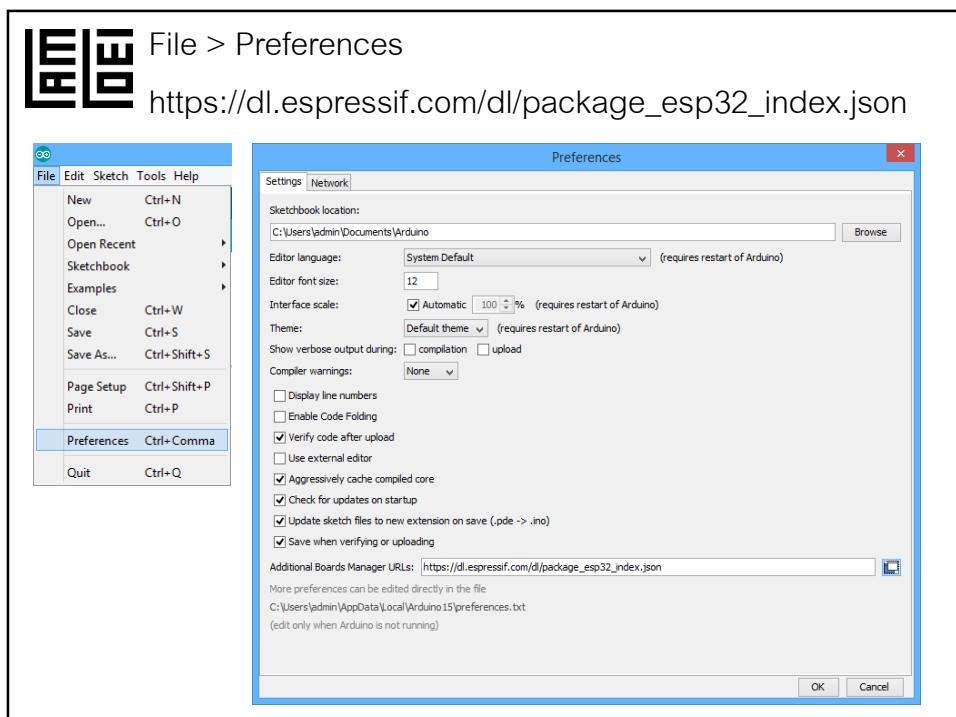
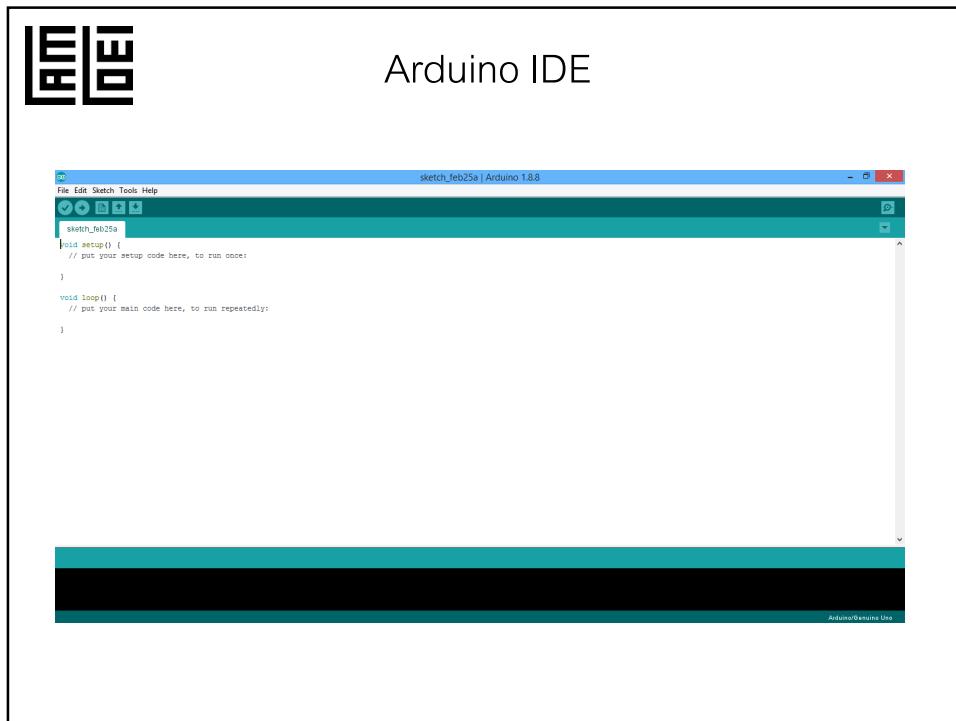
Windows Installer, for Windows XP and up
Windows ZIP file for non admin install

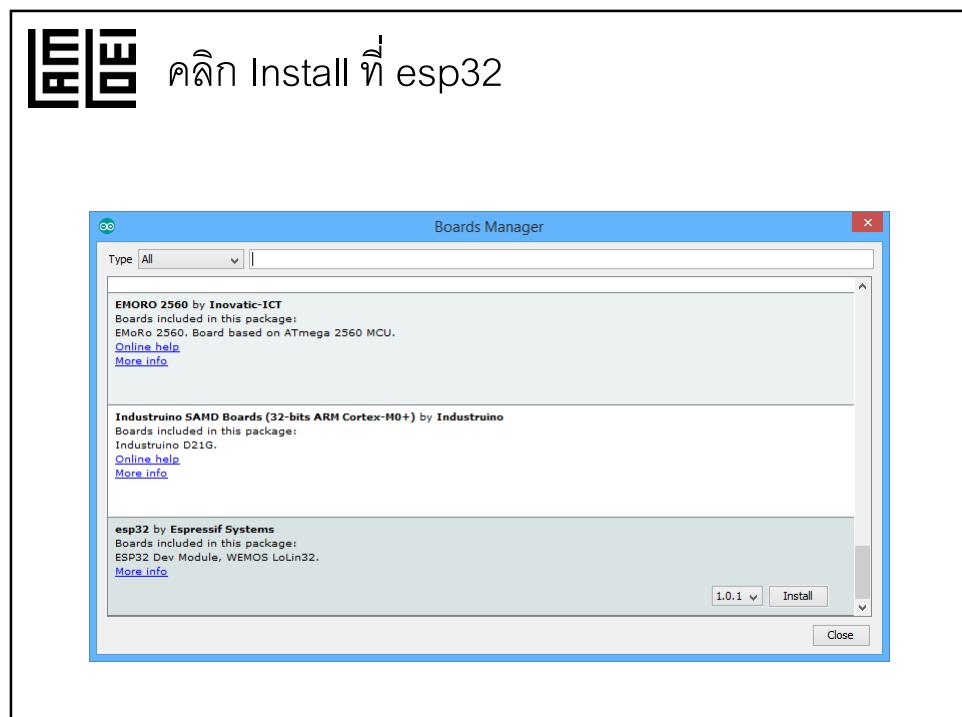
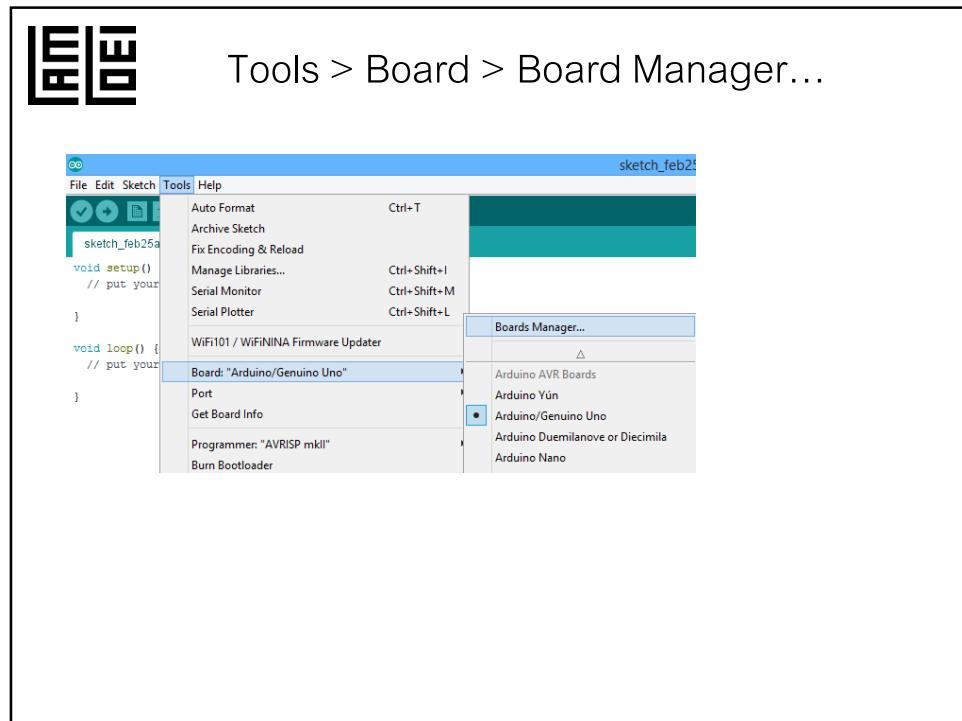
Windows app Requires Win 8.1 or 10
[Get](#)

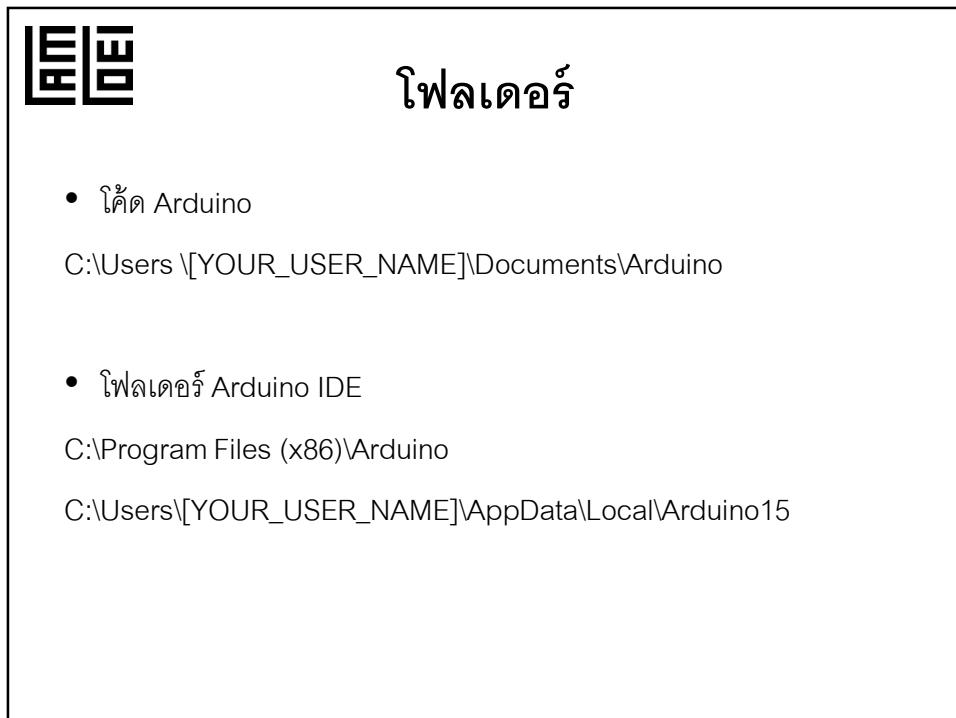
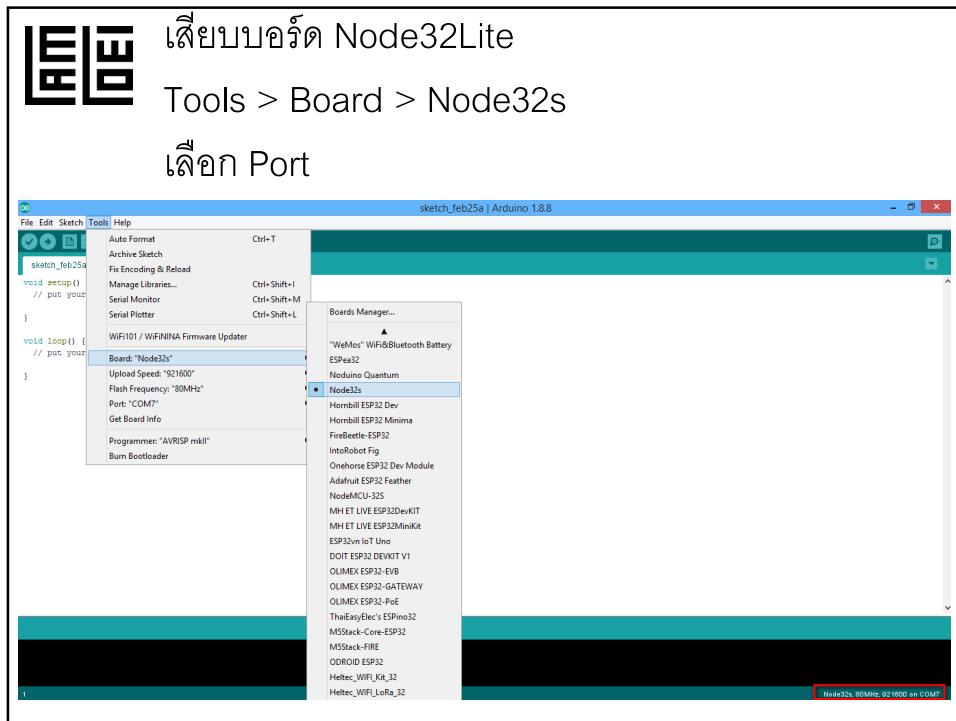
Mac OS X 10.8 Mountain Lion or newer

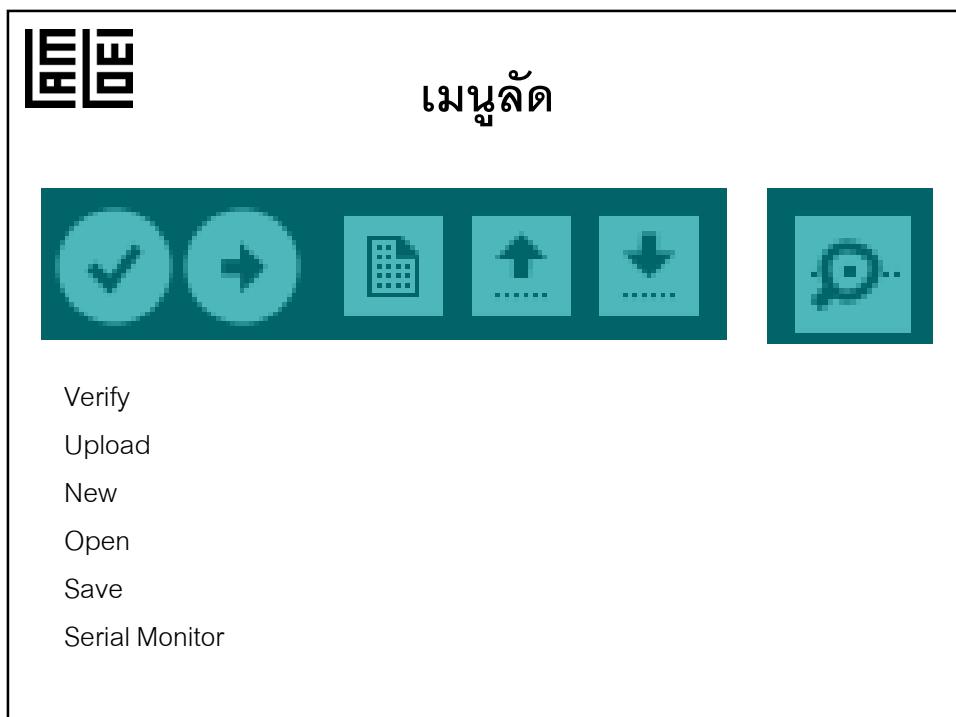
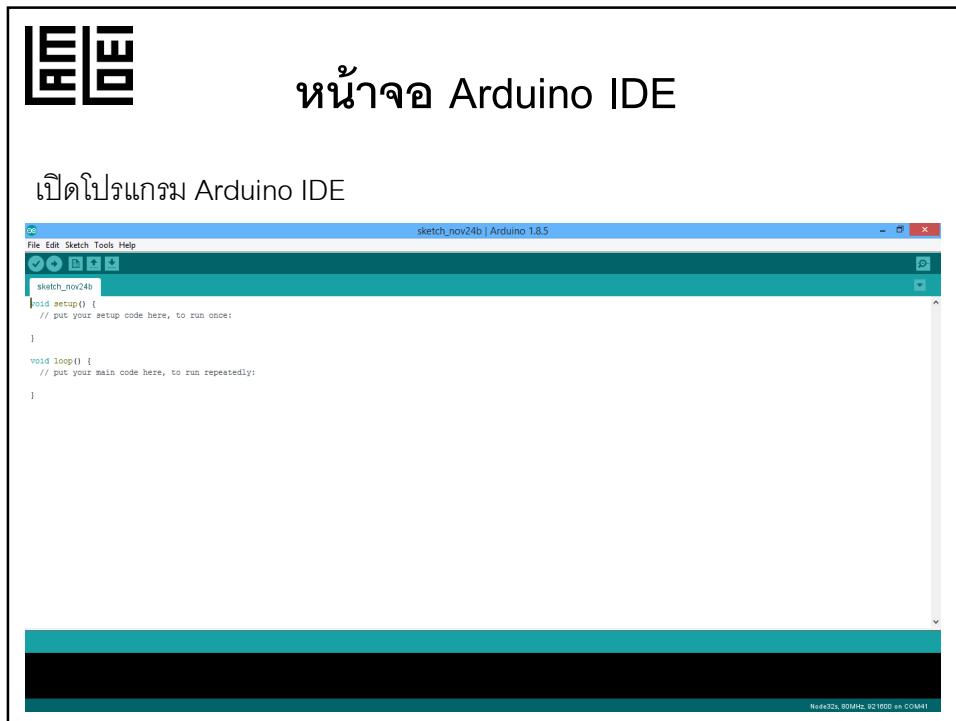
Linux 32 bits
Linux 64 bits
Linux ARM

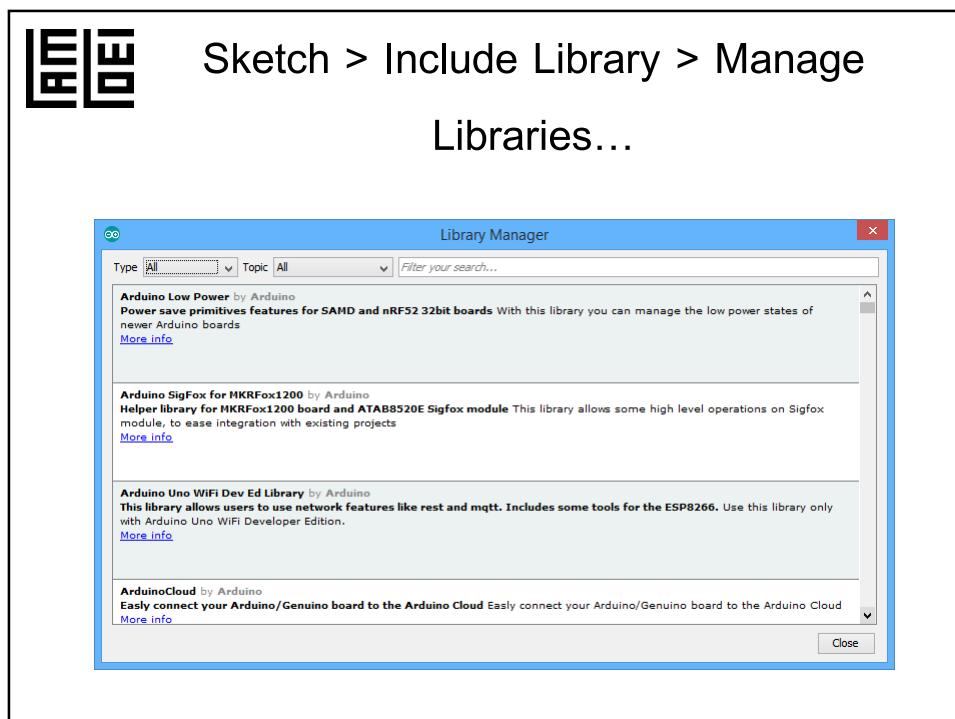
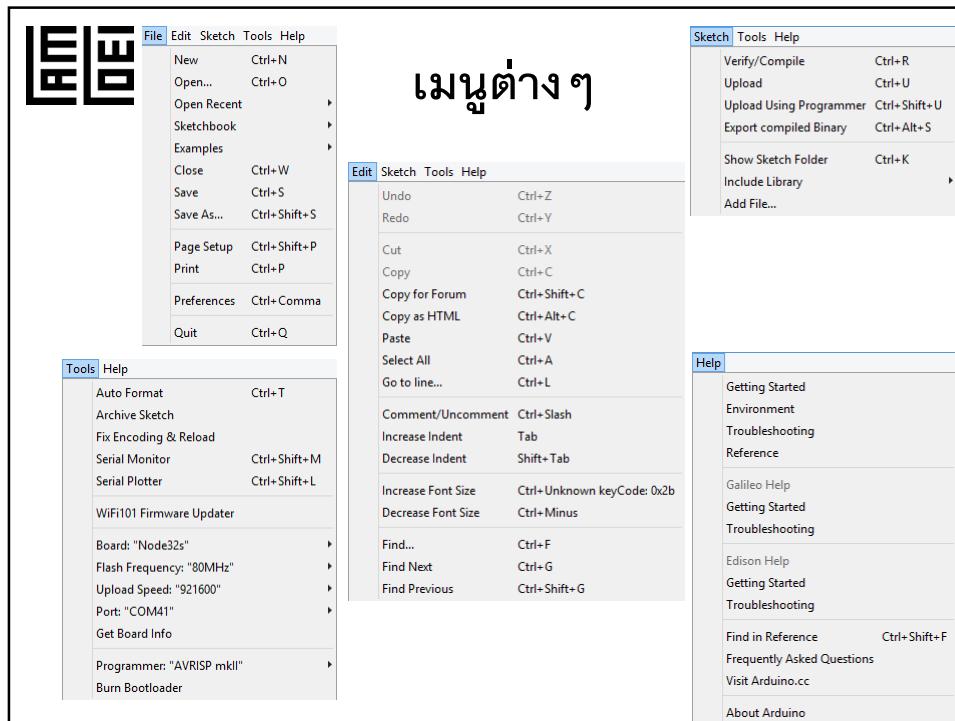
[Release Notes](#)
[Source Code](#)
[Checksums \(sha512\)](#)

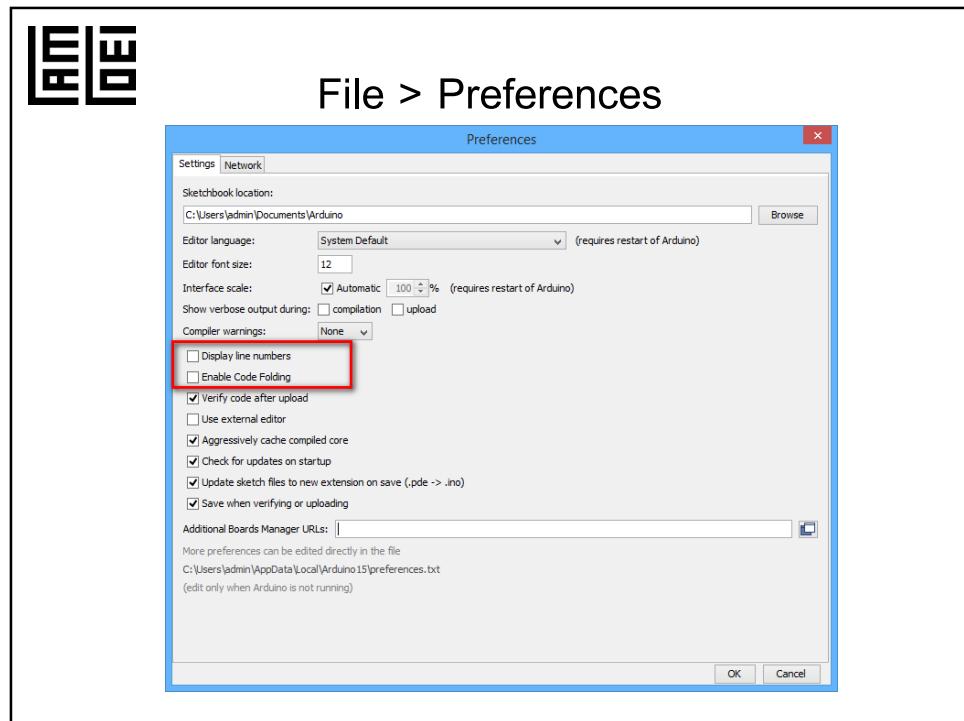
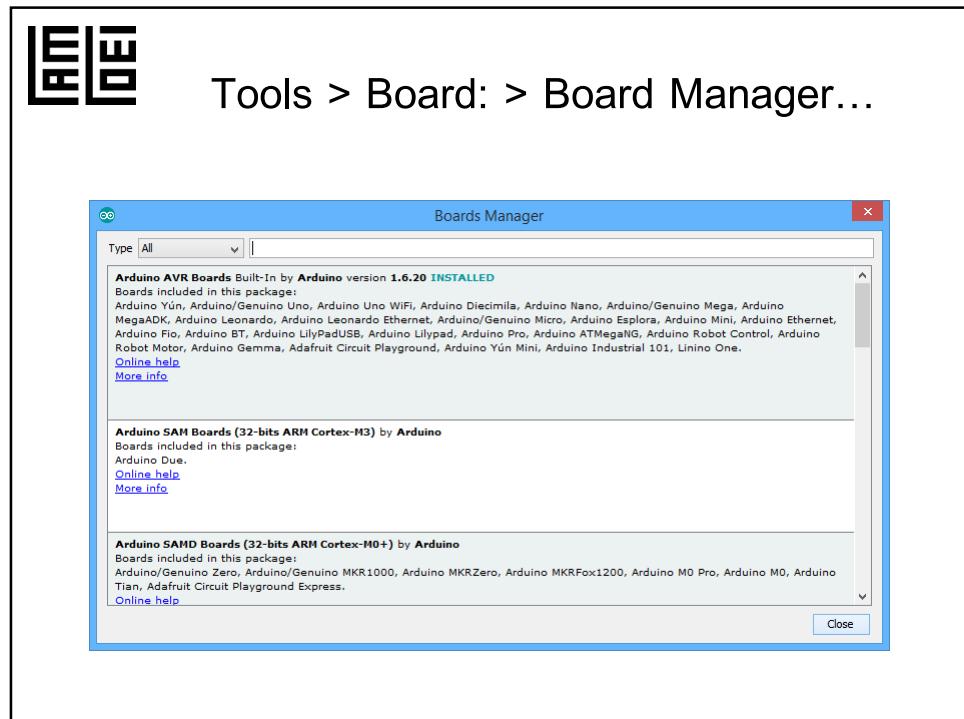


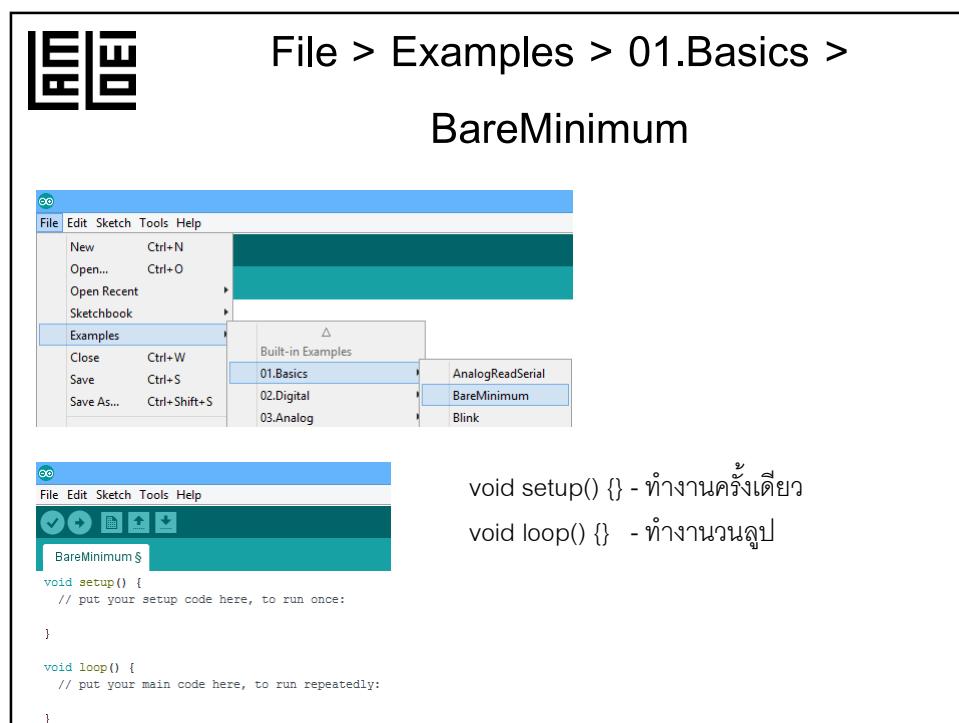
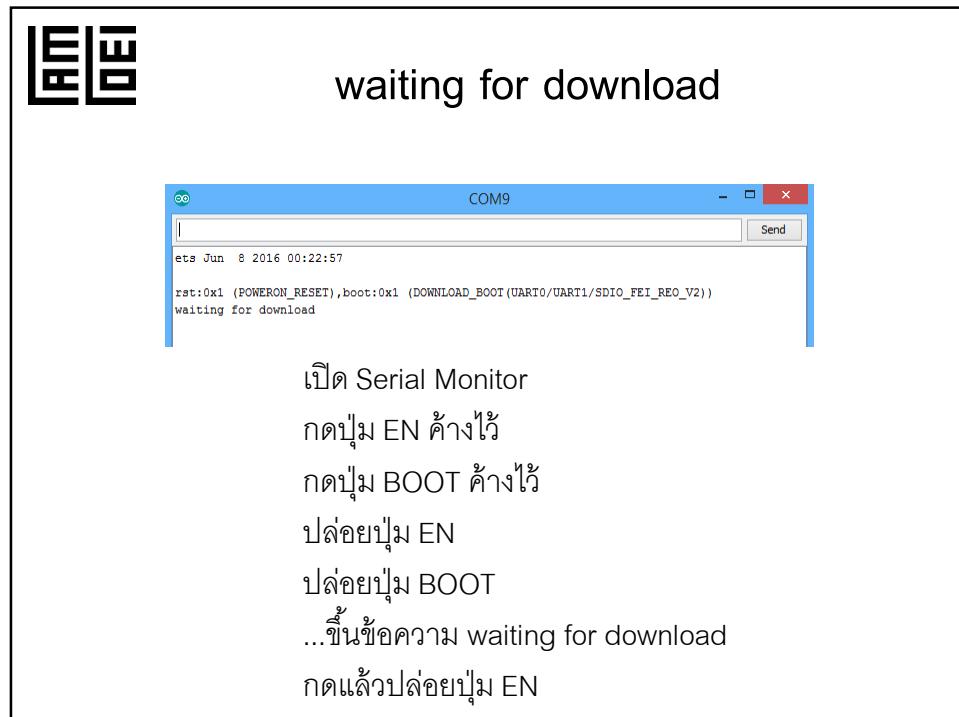






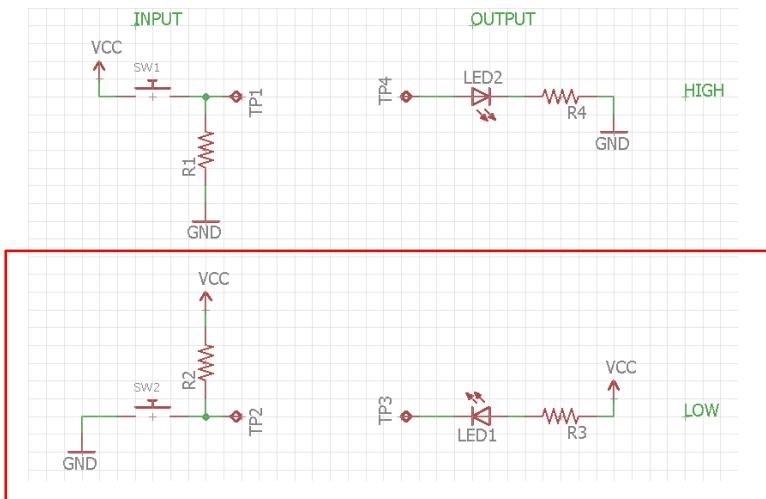








Active High Active Low

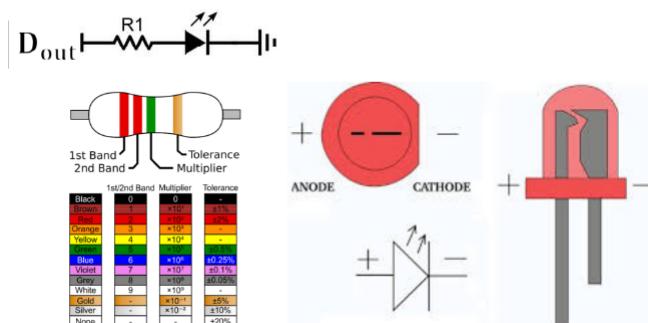


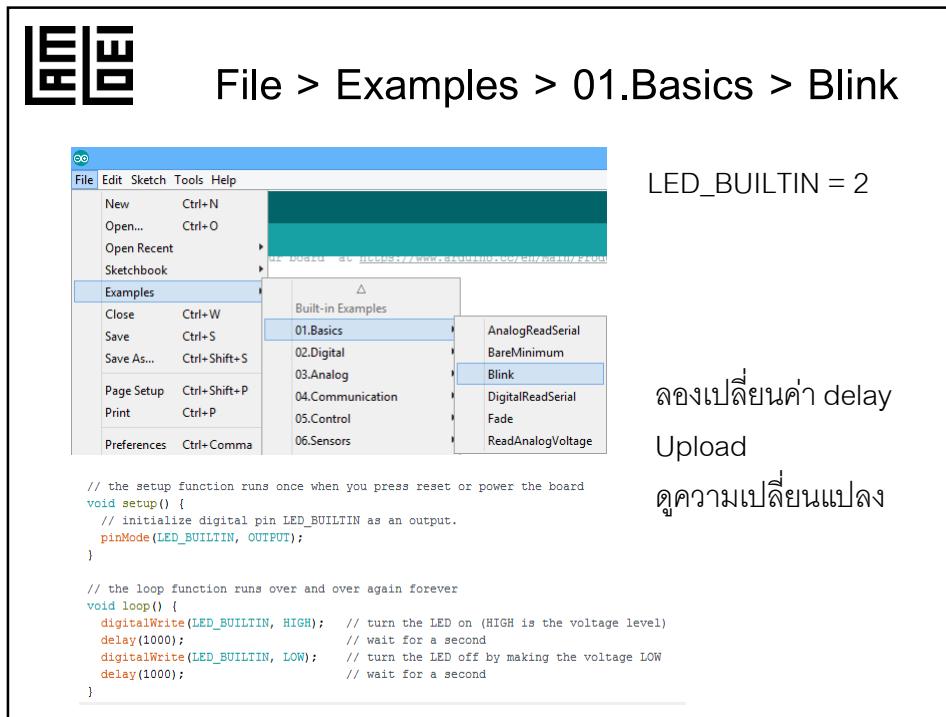
Active Low จะใช้แหล่งจ่ายภายนอก ทำให้ไม่เป็นภาระของ MCU



LED & Resistor

LED & resistor





```

#ifndef Pins_Arduino_h
#define Pins_Arduino_h

#include <stdint.h>

#define EXTERNAL_NUM_INTERRUPTS 16
#define NUM_DIGITAL_PINS 40
#define NUM_ANALOG_INPUTS 16

#define analogInputToDigitalPin(p) (((p)<20)?(esp32_adc2gpio[(p)]):-1)
#define digitalPinToInterrupt(p) (((p)<40)?(p):-1)
#define digitalPinHasPWM(p) ((p < 39))

static const uint8_t LED_BUILTIN = 2;
#define BUILTIN_LED LED_BUILTIN // backward compatibility

static const uint8_t KEY_BUILTIN = 0;

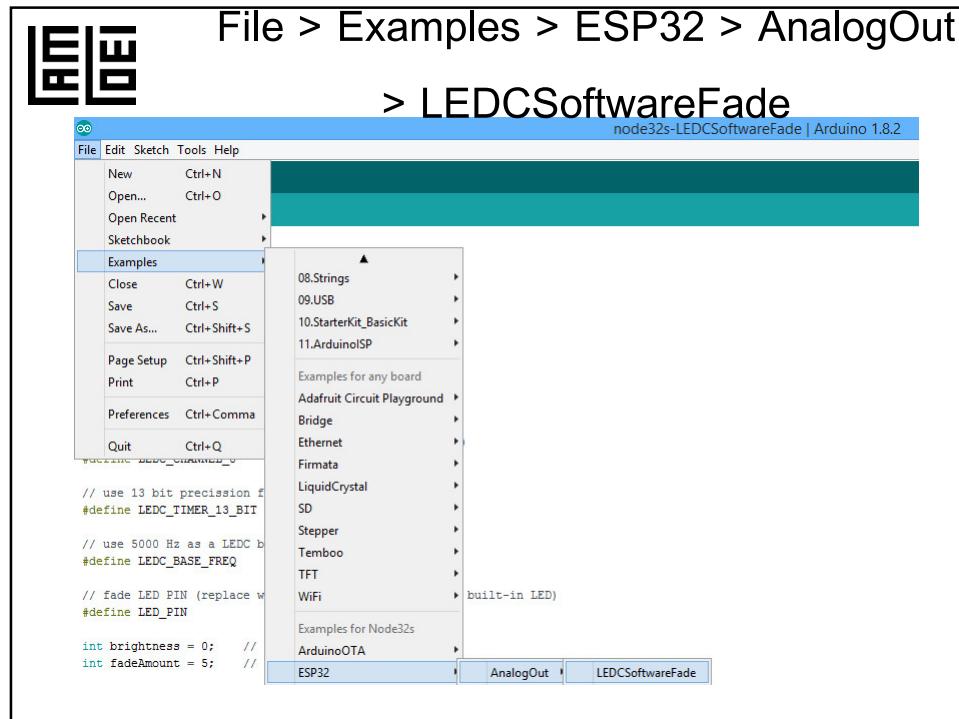
static const uint8_t TX = 1;
static const uint8_t RX = 3;

static const uint8_t SDA = 21;
static const uint8_t SCL = 22;

static const uint8_t SS = 5;
static const uint8_t MOSI = 23;
static const uint8_t MISO = 19;
static const uint8_t SCK = 18;

static const uint8_t A0 = 36;
static const uint8_t A3 = 39;
static const uint8_t A4 = 32;
static const uint8_t A5 = 33;
static const uint8_t A6 = 34;
static const uint8_t A7 = 35;
static const uint8_t A10 = 4;
static const uint8_t A11 = 0;

```



The screenshot shows the Arduino IDE during the upload process. The title bar reads "node32s-LEDCSoftwareFade | Ar". The status bar at the bottom shows "Done uploading.", "Hash of data verified.", "Leaving...", and "Board resetting...". The code editor displays the same sketch as the previous screenshot, with the line "#define LED_PIN 2" highlighted by a red box.

```

#define LEDC_TIMER_13_BIT 13

#define LEDC_BASE_FREQ 15000

#define LED_PIN 2

int brightness = 0; // how bright the LED is
int fadeAmount = 5; // how many points to fade the LED by
  
```

#define LEDC_BASE_FREQ 15000
#define LED_PIN 2



File > Examples > 02.Digital > BlinkWithoutDelay

```

node32lite_BlinkWithoutDelay | Arduino 1.8.8

File Edit Sketch Tools Help
(node32lite_BlinkWithoutDelay)
node32lite_BlinkWithoutDelay

void setup() {
  // set the digital pin as output:
  pinMode (ledPin, OUTPUT);
}

void loop() {
  // here is where you'd put code that needs to be running all the time.

  // check to see if it's time to blink the LED; that is, if the difference
  // between the current time and last time you blinked the LED is bigger than
  // the interval at which you want to blink the LED.
  unsigned long currentMillis = millis();

  if (currentMillis - previousMillis >= interval) {
    // save the last time you blinked the LED
    previousMillis = currentMillis;

    // if the LED is off turn it on and vice-versa:
    if (ledState == LOW) {
      ledState = HIGH;
    } else {
      ledState = LOW;
    }

    // set the LED with the ledState of the variable:
    digitalWrite(ledPin, ledState);
  }
}

Done uploading.

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

```



แบบฝึกหัดที่ 2 – ลองต่อ LED

1. ต่อ LED ที่ขาของ Node32Lite ให้ LED สว่าง ดับ ตามลำดับ

ARDUINO CHEAT SHEET
JEROEN DOGGEN, AP UNIVERSITY COLLEGE ANTWERP

This image shows a comprehensive Arduino cheat sheet. It includes sections on Structure, Control Structures, Further Syntax, General Operators, Pointer Access, Bitwise Operators, Compound Operators, Constants, Data Types, Arrays, Strings, Conversion, Qualifiers, Interrupts, Advanced I/O, Time, Math, Pseudo Random Numbers, I/O Pins, Analog I/O, Digital I/O, Serial Communication, and Websites. Each section contains relevant code snippets and descriptions. A photograph of an Arduino Uno board is also included.

<https://forum.arduino.cc/index.php?topic=173843.0>

File > Examples > 02.Digital > Button

The screenshot shows the Arduino IDE interface with the 'Button' example sketch selected under 'File > Examples > 02.Digital > Button'. The code for the sketch is displayed in the main window:

```
const int buttonPin = 0;
const int ledPin = 2;
```

Below the code, there is a message in Thai: "กดปุ่มไฟสว่าง ปล่อยปุ่มไฟดับ".



แบบฝึกหัดที่ 3

- ให้ กดปุ่มไฟดับ ปล่อยปุ่มไฟสว่าง
- กดปุ่มไฟดับ กดปุ่มไฟสว่าง กดปุ่มไฟดับ กดปุ่มไฟสว่าง



The screenshot shows the Arduino IDE interface with the title "node32s-Serial | Arduino 1.8.5". The code in the sketch is:

```
node32s-Serial
void setup() {
  Serial.begin(115200);
}

void loop() {
  Serial.println("Hello Node32s");
}
```

The Serial Monitor window titled "COM41" displays the repeated message "Hello Node32s". The baud rate dropdown in the monitor window is set to "115200 baud".

พิมพ์คำสั่ง

ปรับค่า Serial 115200 ให้ตรงกัน



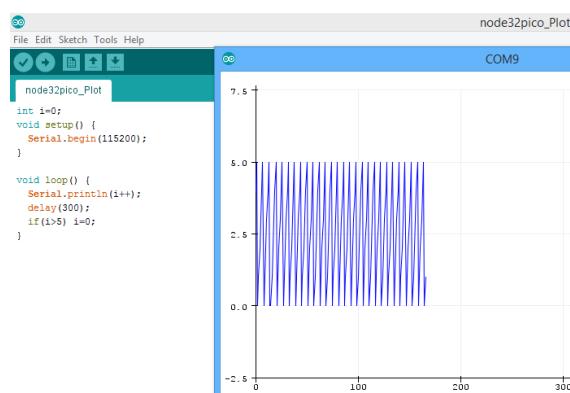
แบบฝึกหัดที่ 4

ให้แสดงที่ Serial Monitor

```
*  
**  
***  
****  
*****
```



Tools > Serial Plotter



พิมพ์ค่าสั่ง

เมนู Tools > Serial Plotter

ปรับค่า Serial 115200 ให้ตรงกัน



สรุปขั้นตอน

- File > Examples
- ปรับค่า
- Upload Program
- ดูเกตความเปลี่ยนแปลง



File > Examples > ESP32 > ChipID > GetChipID

Arduino 1.8.5 | GetChipID

```

File Edit Sketch Tools Help
GetChipID
uint64_t chipid;

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

void loop() {
    chipid=ESP.getEfuseMac(); //The chip ID is essentially its MAC address(length: 6 bytes).
    Serial.printf("ESP32 Chip ID = %04X", (uint16_t)(chipid>>32)); //print High 2 bytes
    Serial.printf("%08X\n", (uint32_t)chipid); //print Low 4bytes.

    delay(3000);
}

```

COM41

```

| Send
ESP32 Chip ID = 8CCE03C40A24
ESP32 Chip ID = 8CCE03C40A24
ESP32 Chip ID = 8CCE03C40A24

```



File > Examples > ESP32 > Touch > TouchRead

TouchRead | Arduino 1.8.5

```

File Edit Sketch Tools Help
TouchRead
// ESP32 Touch Test
// Just test touch pin - Touch0 is T0 which is on
void setup()
{
    Serial.begin(115200);
    delay(1000); // give me time to bring up serial
    Serial.println("ESP32 Touch Test");
}

void loop()
{
    Serial.println(touchRead(T0)); // get value using
    delay(1000);
}

```

COM41

61
0
61
61
13
15
61
18
62
62
62
62

Autoscroll No line ending 115200 baud



File > Examples > ESP32 > DeepSleep > TimerWakeUp

TimerWakeUp | Arduino 1.8.5

```

File Edit Sketch Tools Help
TimerWakeUp
Author:
Pranav Cherukupalli <cherukupallip@gmail.com>
*/
#define uS_TO_S_FACTOR 1000000 /* Conversion factor */
#define TIME_TO_SLEEP 5 /* Time ESP32 will sleep */

RTC_DATA_ATTR int bootCount = 0;

/*
Method to print the reason by which ESP32
has been awoken from sleep
*/
void print_wakeup_reason() {
    esp_sleep_wakeups_t wakeup_reason;
    wakeup_reason = esp_sleep_get_wakeup_cause();

    switch(wakeup_reason) {
        case 1 : Serial.println("Wakeup caused by external signal using RTC_IO"); break;
        case 2 : Serial.println("Wakeup caused by external signal using RTC_CNTL"); break;
        case 3 : Serial.println("Wakeup caused by timer"); break;
        case 4 : Serial.println("Wakeup caused by touchpad"); break;
        case 5 : Serial.println("Wakeup caused by UIP program"); break;
        default : Serial.println("Wakeup was not caused by deep sleep"); break;
    }
}

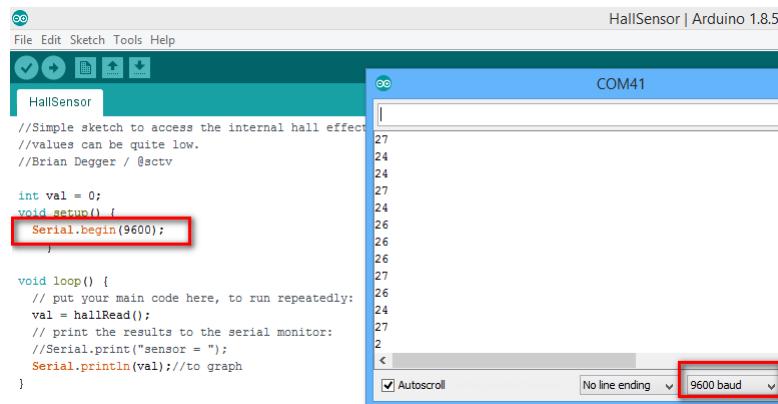
Setup ESP32 to sleep for every 5 Seconds
Going to sleep now
Boot number: 2
Wakeup caused by timer
Setup ESP32 to sleep for every 5 Seconds
Going to sleep now
Boot number: 3
Wakeup caused by timer
Setup ESP32 to sleep for every 5 Seconds
Going to sleep now

```

COM41

Autoscroll No line ending 115200 baud Clear output

File > Example > ESP32 > HallSensor



```
//Simple sketch to access the internal hall effect
//values can be quite low.
//Brian Degger / @sctv

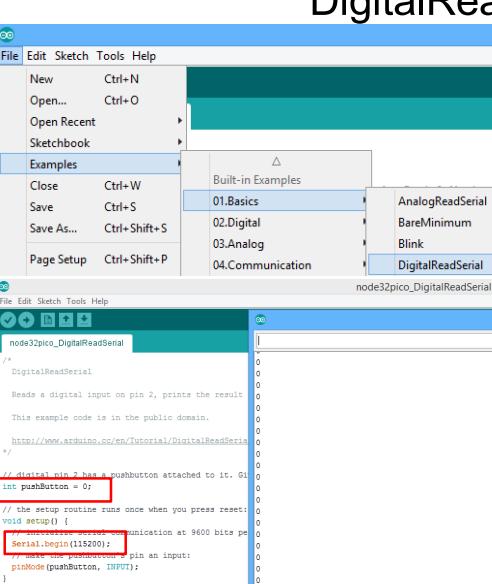
int val = 0;
void setup() {
  Serial.begin(9600);
}

void loop() {
  // put your main code here, to run repeatedly:
  val = hallRead();
  // print the results to the serial monitor:
  //Serial.print("sensor = ");
  Serial.println(val); //to graph
}
```

ปรับ Serial 9600

File > Examples > 01.Basics >

DigitalReadSerial



```
int pushButton = 0;
Serial.begin(115200);

/*
DigitalReadSerial
Reads a digital input on pin 2, prints the result
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/DigitalReadSerial
*/
//Digital pin 2 has a pushbutton attached to it. G1
int pushButton = 0;

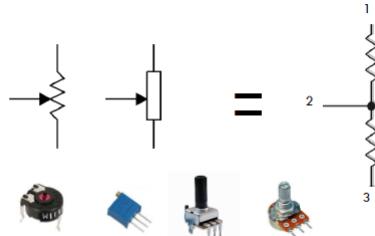
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(115200);
  // make the pushbutton a pin input:
  pinMode(pushButton, INPUT);
}
```

กดปุ่ม BOOT
ดูความเปลี่ยนแปลง



Volume

Potentiometer



File > Examples > 01.Basics > AnalogReadSerial

```

node32lite_AnalogReadSerial | Arduino 1.8.8
File Edit Sketch Tools Help
node32lite_AnalogReadSerial
/*
AnalogReadSerial

Reads an analog input on pin 0, prints the result to the Serial Monitor.
Graphical representation is available using Serial Plotter (Tools > Serial Plotter)
Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and GND.

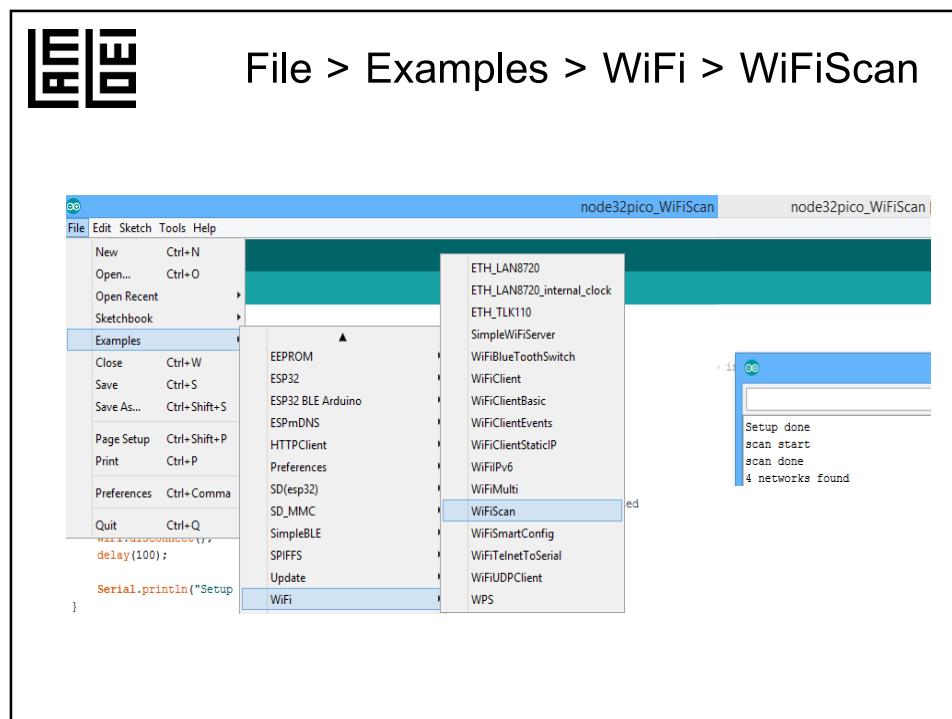
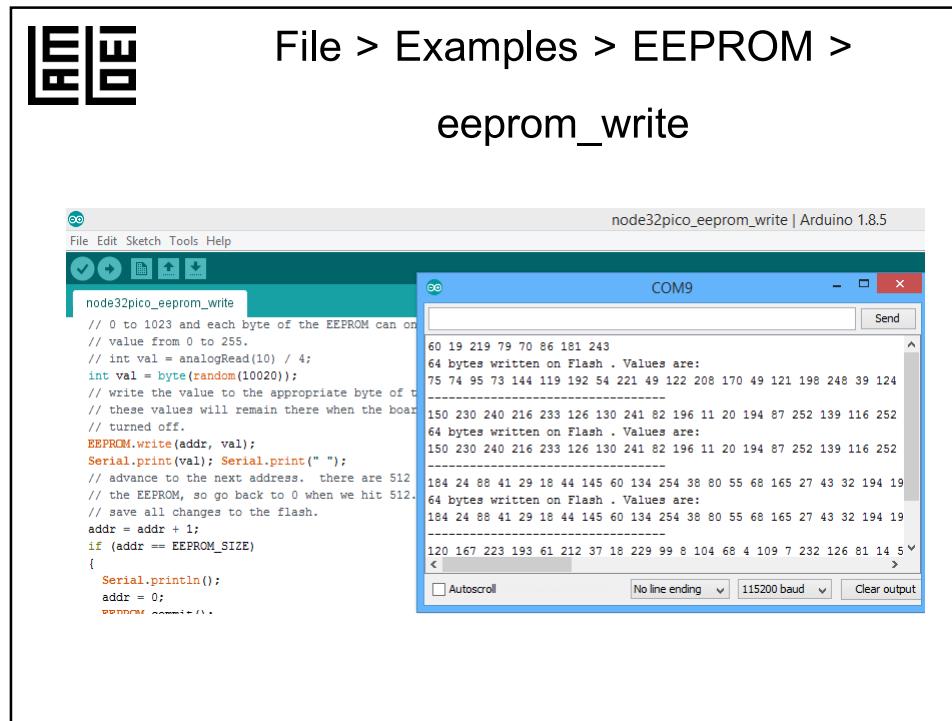
This example code is in the public domain.

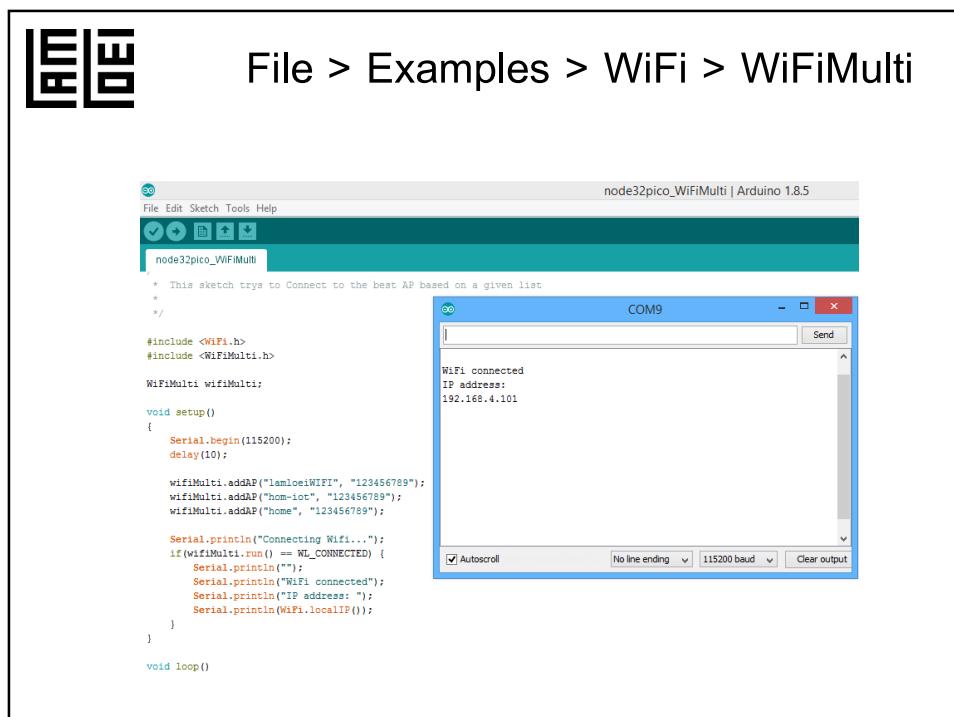
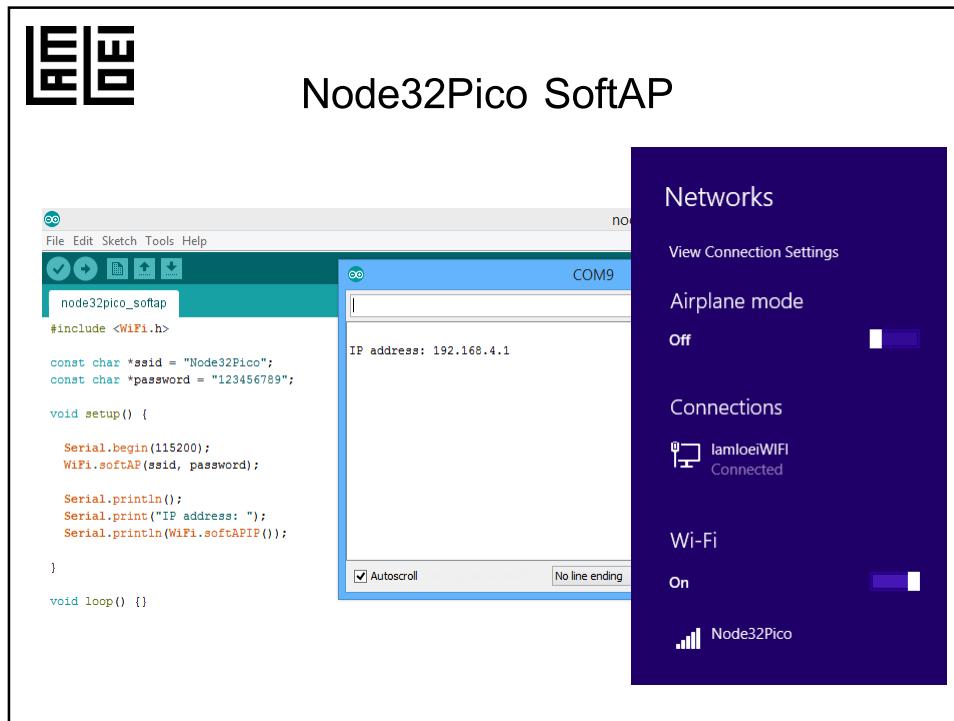
http://www.arduino.cc/en/Tutorial/AnalogReadSerial
*/
// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1); // delay in between reads for stability
}

```

ต่อ Volume เข้ากับขา A0 (io36)

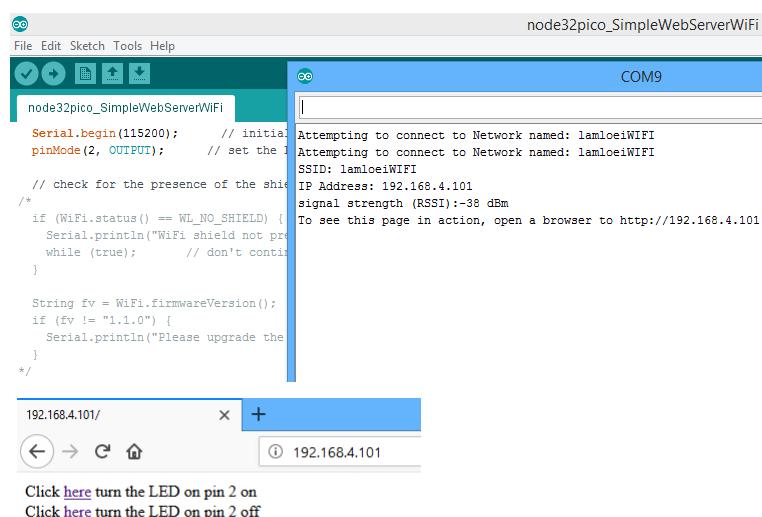






File > Examples > WiFi >

SimpleWebServerWiFi



```

File Edit Sketch Tools Help
node32pico_SimpleWebServerWiFi
Serial.begin(115200); // initial
pinMode(2, OUTPUT); // set the i
/*
// check for the presence of the shi
if (WiFi.status() == WL_NO_SHIELD) {
  Serial.println("WiFi shield not pr
  while (true); // don't contin
}

String fv = WiFi.firmwareVersion();
if (fv != "1.1.0") {
  Serial.println("Please upgrade the
}
*/

```

Attempting to connect to Network named: lamloeiWIFI
 Attempting to connect to Network named: lamloeiWIFI
 SSID: lamloeiWIFI
 IP Address: 192.168.4.101
 signal strength (RSSI):-38 dBm
 To see this page in action, open a browser to http://192.168.4.101

192.168.4.101 +
 ↻ → ⌂ ⌂ ① 192.168.4.101

Click [here](#) turn the LED on pin 2 on
 Click [here](#) turn the LED on pin 2 off



לְאֵלָה

```

const char* ssid    = "lamloeiWIFI";
const char* password = "123456789";
Serial.begin(115200);
pinMode(2, OUTPUT);
/*
if (WiFi.status() == WL_NO_SHIELD) {
  Serial.println("WiFi shield not present");
  while (true); // don't continue
}
String fv = WiFi.firmwareVersion();
if (fv != "1.1.0") {
  Serial.println("Please upgrade the firmware");
}
*/

```



```

client.print("Click <a href=\"/H\">here</a> turn the LED on pin 2 on<br>");
client.print("Click <a href=\"/L\">here</a> turn the LED on pin 2 off<br>");

if (currentLine.endsWith("GET /H")) {
    digitalWrite(2, LOW);           // GET /H turns the LED on
}

if (currentLine.endsWith("GET /L")) {
    digitalWrite(2, HIGH);          // GET /L turns the LED off
}

```



File > Examples > NetBIOS > ESP_NBNST

node32lite_ESP_NBNST | Arduino 1.8.8

The screenshot shows the Arduino IDE with the 'node32lite_ESP_NBNST' sketch open. The code includes #include <WiFi.h> and #include <NetBIOS.h>. It defines WiFi credentials and sets up the WiFi connection. It then enters a loop where it waits for connection, prints connection status, and prints the IP address. Finally, it initializes NBNS and begins ESP operations. In the background, a Command Prompt window shows two ping operations: one to 192.168.4.112 and one to 'ESP'. Both pings show 0% loss and low round-trip times.

```

File Edit Sketch Tools Help
File Edit Sketch Tools Help
node32lite_ESP_NBNST | Arduino 1.8.8
File Edit Sketch Tools Help
File Edit Sketch Tools Help
node32lite_ESP_NBNST
#include <WiFi.h>
#include <NetBIOS.h>

const char* ssid = "lanloeiWIFI";
const char* password = "123456789";

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

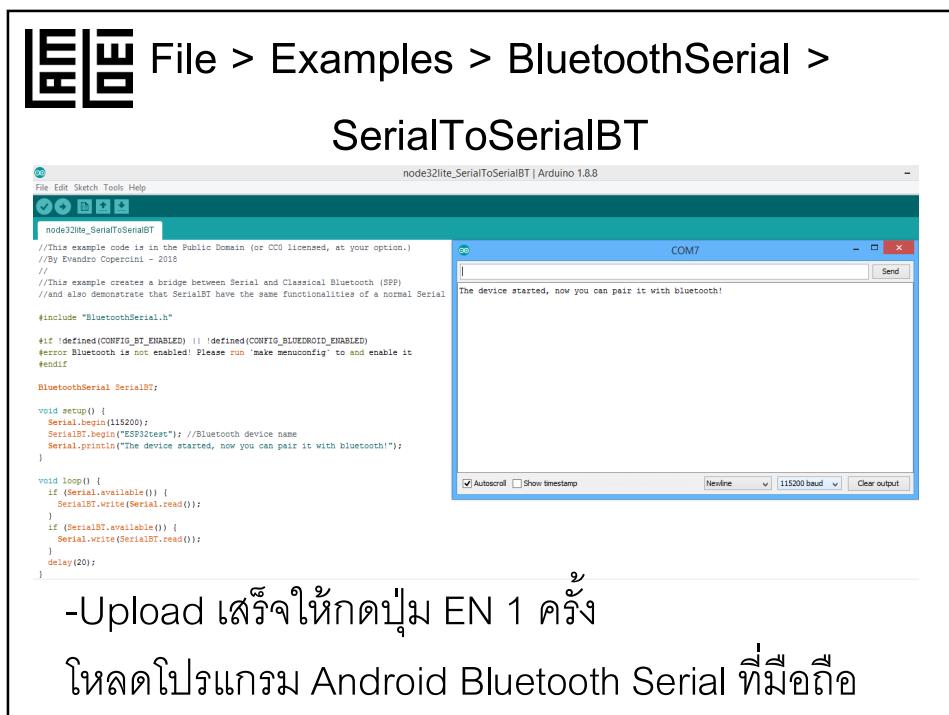
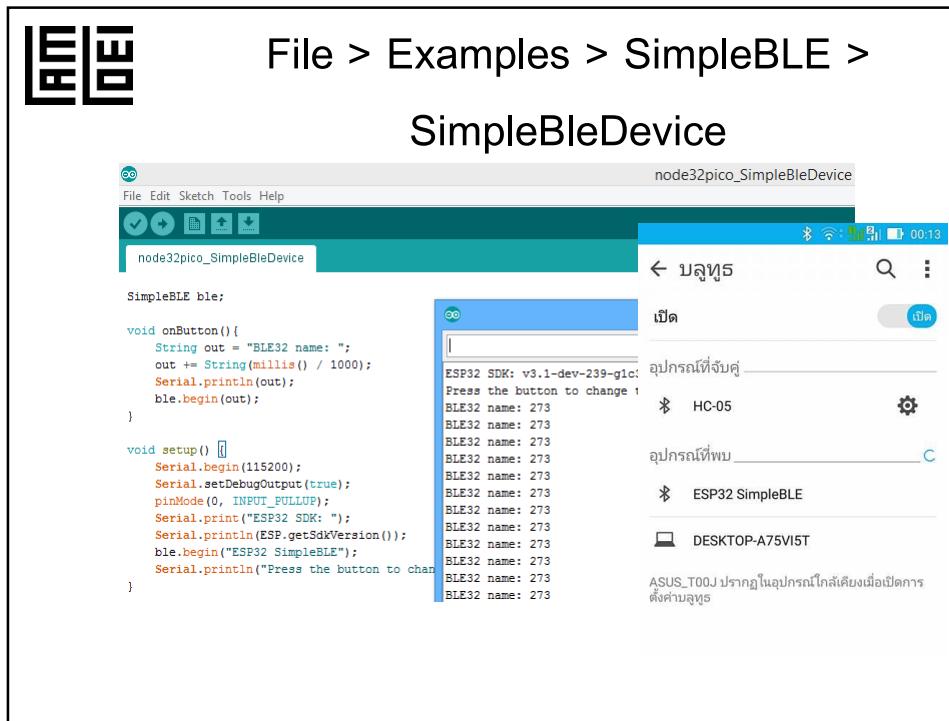
    // Connect to WiFi network
    WiFi.mode(WIFI_STA);
    WiFi.begin(ssid, password);
    Serial.println("Connected to lanloeiWIFI");
    IR address: 192.168.4.112

    // Wait for connection
    while (WiFi.status() != WL_CONNECTED) {
        delay(500);
        Serial.print(".");
    }
    Serial.println("");
    Serial.println("Connected to ");
    Serial.println(ssid);
    Serial.print("IP address: ");
    Serial.println(WiFi.localIP());

    NBNS.begin("ESP");
}

Command Prompt
C:\Users\admin>ping 192.168.4.112
Pinging 192.168.4.112 with 32 bytes of data:
Reply from 192.168.4.112: bytes=32 time=15ms TTL=255
Reply from 192.168.4.112: bytes=32 time=21ms TTL=255
Reply from 192.168.4.112: bytes=32 time=28ms TTL=255
Reply from 192.168.4.112: bytes=32 time=30ms TTL=255
Ping statistics for 192.168.4.112:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 15ms, Maximum = 30ms, Average = 21ms
C:\Users\admin>ping ESP
Pinging ESP (192.168.4.112) with 32 bytes of data:
Reply from 192.168.4.112: bytes=32 time=15ms TTL=255
Reply from 192.168.4.112: bytes=32 time=19ms TTL=255
Reply from 192.168.4.112: bytes=32 time=31ms TTL=255
Reply from 192.168.4.112: bytes=32 time=40ms TTL=255
Ping statistics for 192.168.4.112:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 15ms, Maximum = 31ms, Average = 25ms
C:\Users\admin>

```





เพิ่ม Code จากภายนอก

เปิดเบราว์เซอร์ไปที่ <https://playground.arduino.cc/Main/I2cScanner>

คลิก Get Code

```

73. Serial.print("0");
74. Serial.println(address,HEX);
75. }
76. }
77. if (nDevices == 0)
78. Serial.println("No I2C devices found\n");
79. else
80. Serial.println("done\n");
81.
82. delay(5000); // wait 5 seconds for next scan
83. }

```

[Get Code]



ต่อ SHT30 เข้ากับ Node32Lite

Node32Lite	SHT30
1. GND	GND
2. 3V3	3V3
3. IO22	D1
4. IO21	D2



แสดง Address ของ SHT30 ที่ 0x45

The screenshot shows the Arduino IDE interface with the title bar "node32lite_i2c_scanner | Arduino 1.8.8". The code in the editor is as follows:

```

File Edit Sketch Tools Help
node32lite_i2c_scanner
#include <Wire.h>

void setup()
{
    Wire.begin();
    Serial.begin(115200);
    while (!Serial); // Leonardo: wait for serial monitor
    Serial.println("\nI2C Scanner");
}

void loop()
{
    byte error, address;
    int nDevices;
    Serial.println("Scanning...");

    for (int i = 1; i <= 127; i++)
    {
       Wire.beginTransmission(i);
        error = Wire.endTransmission();
        if (error == 0)
        {
            address = i;
            nDevices++;
        }
        else
        {
            Serial.print("I2C device found at address ");
            Serial.print(i);
            Serial.println(" !");
        }
    }
    Serial.print("done\n");
}

```

The Serial Monitor window displays the output of the I2C scan:

```

I2C device found at address 0x45 !
done

Scanning...
I2C device found at address 0x45 !
done

Scanning...
I2C device found at address 0x45 !
done

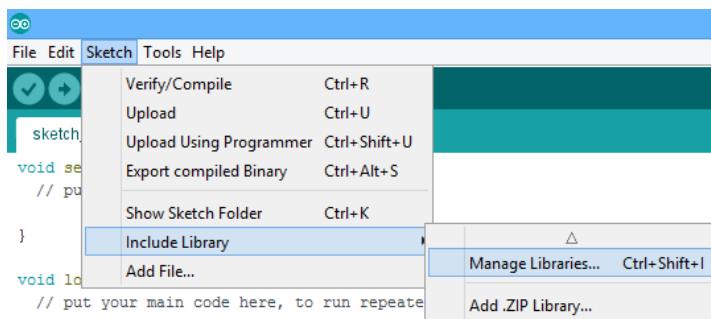
Scanning...
I2C device found at address 0x45 !
done

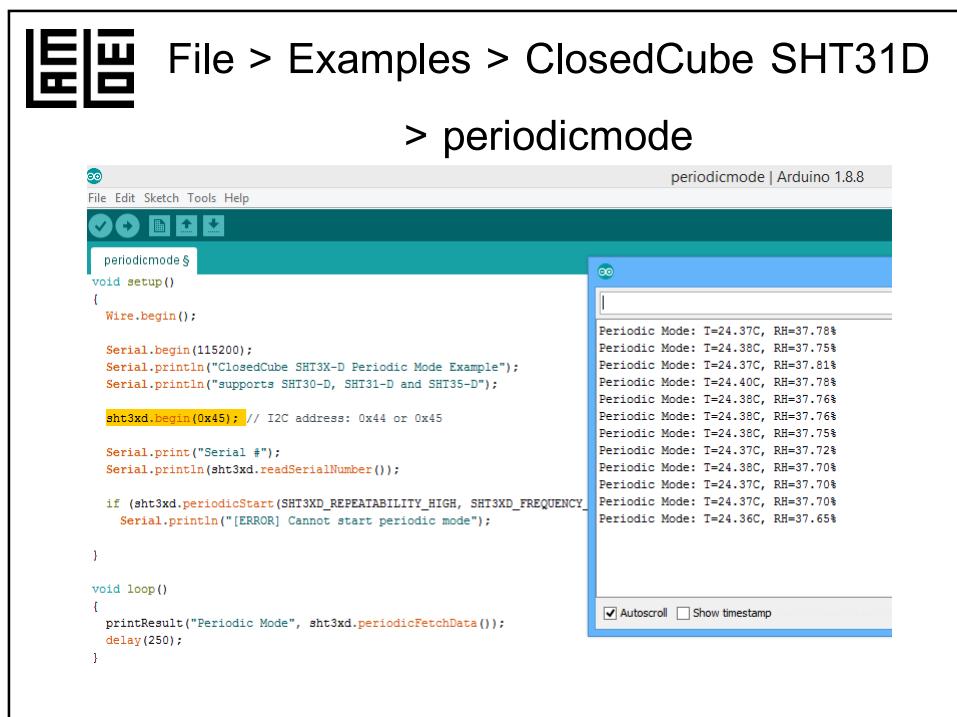
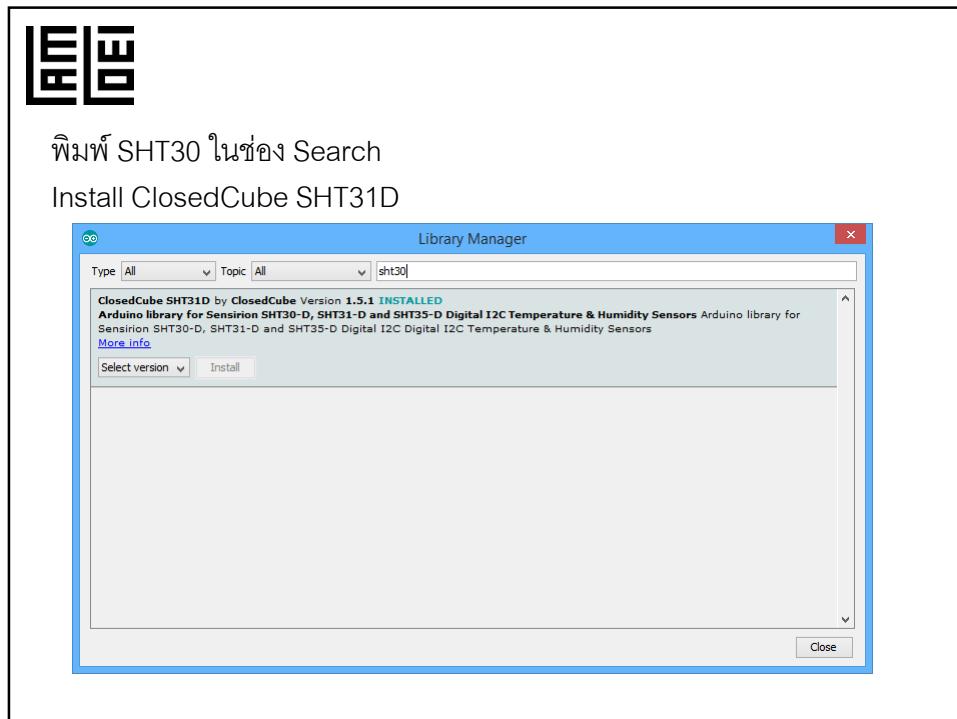
```



อ่านค่า อุณหภูมิ ความชื้น เพิ่ม library

Sketch > Include Library > Manage Libraries...







แบบฝึกหัดที่ 5

ให้เลือกตัวอย่าง File > Example และอธิบาย



แบบฝึกหัดที่ 6

สร้างระบบด้วย

1. เซ็นเซอร์ (ค่าภายใน เป็นตัวเลข)
2. LED (Digital Output)
3. Tact Switch (Digital Input)
4. Volume (Analog Input)



<https://www.facebook.com/lamloeicom>