

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

คอร์สอบรม

Node32Lite

Arduino IDE Basic

ขั้นพื้นฐาน



## จุดประสงค์

- ผู้เข้าอบรมสามารถเขียน Arduino IDE ลงบน Node32Lite
- สามารถสร้างระบบนำมายังงานได้

เอกสารคอร์สอ卜รมนี้ สามารถดาวน์โหลดได้ที่

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

\*\*\* คอร์สอ卜รมนี้เหมาะสมกับผู้เริ่มต้น \*\*\*



# กำหนดการ

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



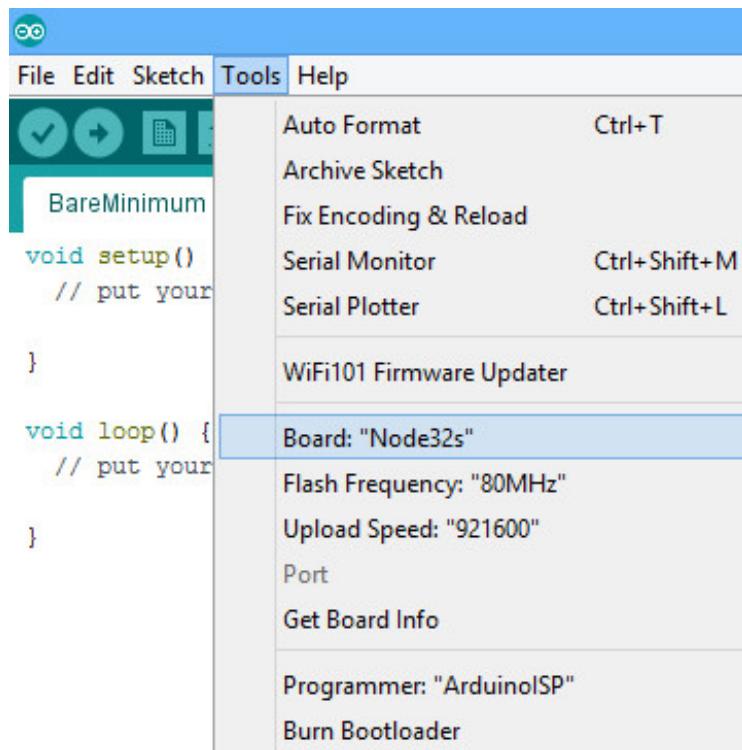
## อุปกรณ์ และเครื่องมือ

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



# พื้นฐาน

## Tools > Board: “Node32s”



1. \* ต้องถอดไฟมด้า
2. เสียบสาย usb บอร์ด แล้วต่อเข้าคอม
3. ต้องเห็น COM port ของบอร์ด
4. ถ้ายังไม่ได้ติดตั้งโปรแกรมให้ข้ามไปติดตั้ง
5. ที่ Arduino IDE ไปที่เมนู Tools

\*จะต้องเห็น Board กับ Port

\* Node32Lite ใช้ profile เดียวกับ Node32s



<https://espressif.com/>

The screenshot shows the official website for Espressif Systems. The header features the 'EE' logo, the word 'ESPRESSIF' in white, and a red Wi-Fi/BT icon. On the right are links for '中文' (Chinese), a search icon, and 'Subscribe'. A vertical navigation menu on the left includes 'Products', 'Company', 'Ecosystem', 'Support', 'Documents', 'FAQ', and 'More'. The main content area has a blue background with a blurred image of a circuit board. It highlights the 'ESP32 Wi-Fi + Bluetooth Combo Chip' and describes it as an 'ultra low power and complete integration solution'. Below this is a call-to-action button labeled 'Learn more >'. To the right, there's a diagram showing an ESP32 chip connected to a city skyline silhouette, with a small antenna and a lightbulb icon above it. A printed circuit board (PCB) with the text 'ESP32-D0WDQ6' and 'P0W255' is shown in the background.



ESP-01



ESP-02



ESP-03



ESP-04



ESP-05



ESP-06



ESP-07



ESP-08



ESP-09



ESP-10



ESP-11



ESP-12



ESP-12E



ESP-12F



ESP-13



ESP-14



ESP-01S



ESP-07S



ESP-08S

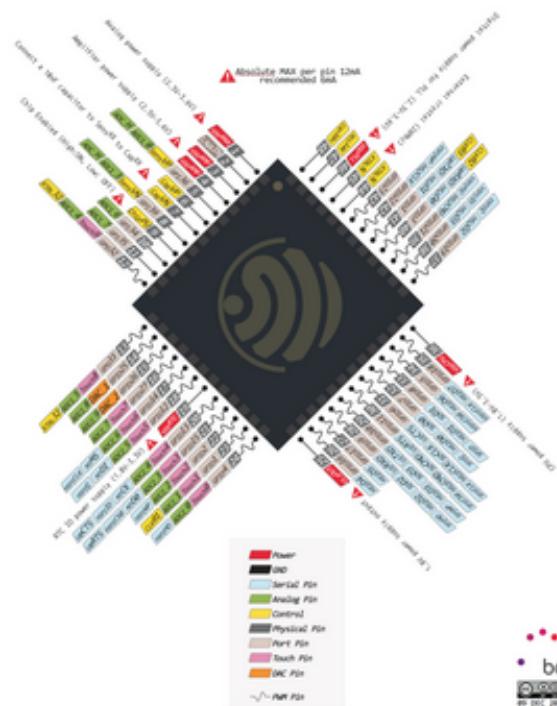


ESP-12S





ESP32  
PINOUT



# ມີດູລ ESP-32s



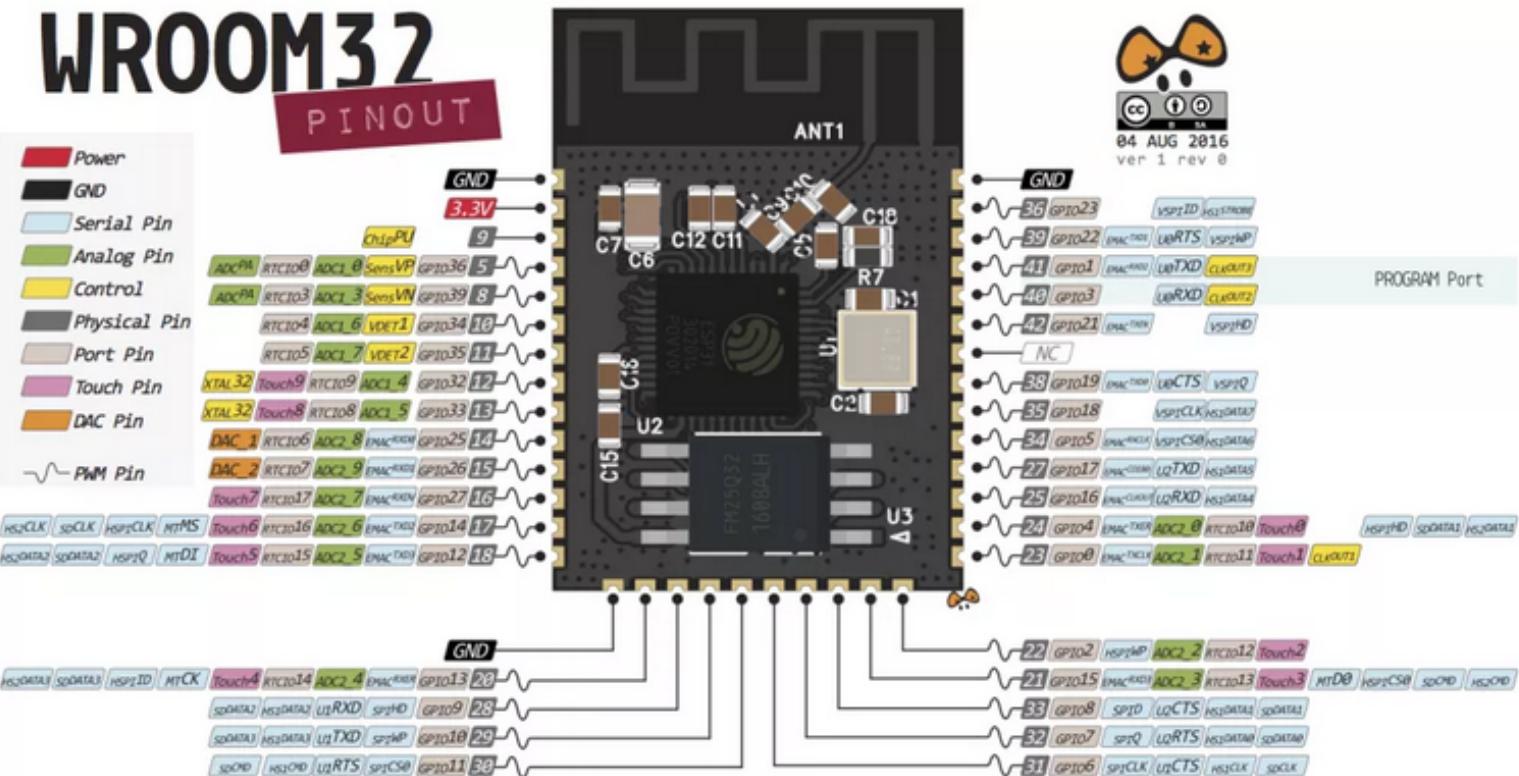
## ໂມຄູລ ESP-WROOM32



esp8266 > esp8285 > esp32



<https://randomnerdtutorials.com/esp32-pinout-reference-gpios/>





# ESP32-DevKitC

Home    https://www.espressif.com/en/products/hardware/esp32-devkitc/resources    ...    Search   

ESPRESSIF | Products / Hardware / ESP32-DevKitC / 中文       Subscribe

**ESP32-DevKitC**

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.



# 100 million

A screenshot of the Espressif website homepage. The URL https://www.espressif.com is visible in the browser bar. The page features a large digital Earth background. In the center, a large digital flip clock displays "100,000,000". Below it, the text "ESPRESSIF ships 100 million IoT chips" is displayed in yellow. A "Learn more" button is located at the bottom left. On the left side, there is a vertical navigation menu with icons and links: Products, Company, Ecosystem, Support, Documents, and Contact Us. At the top right, there are links for "中文" (Chinese), a search bar, and a "Subscribe" button.



## 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>

The screenshot shows the GitHub repository page for `https://github.com/espressif/arduino-esp32`. The repository is described as the "Arduino core for the ESP32". It has 807 commits, 2 branches, 10 releases, and 187 contributors. Recent commits include changes to issue templates, random cores, documentation, libraries, packages, tools, and variants.

**Code**

Issues 710 | Pull requests 25 | Projects 1 | Insights

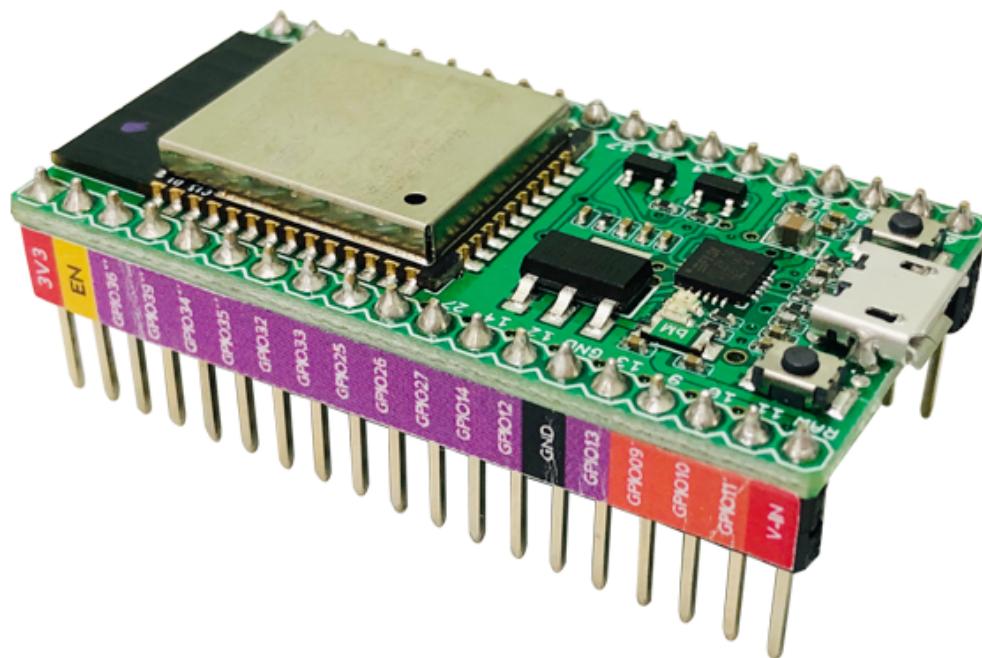
807 commits | 2 branches | 10 releases | 187 contributors

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

Commit	Message	Date
	a-c-sreedhar-reddy and me-no-dev Changed the description of file (#2476)	Latest commit 5af0336 5 days ago
	.github/ISSUE_TEMPLATE Update issue templates	2 months ago
	cores/esp32 Unbiased random (#2468)	7 days ago
	docs python2 get.py does not work behind proxy (#2349)	a month ago
	libraries Changed the description of file (#2476)	5 days ago
	package Update IDF to abea9e4c0 (#2458)	10 days ago
	tools Update IDF to abea9e4c0 (#2458)	10 days ago
	variants Add default pin mapping for Serial1 and Serial2 for wESP32 (#2409)	10 days ago



## Node32Lite



[https://www.gravitechthai.com/product\\_detail.php?d=3318](https://www.gravitechthai.com/product_detail.php?d=3318)

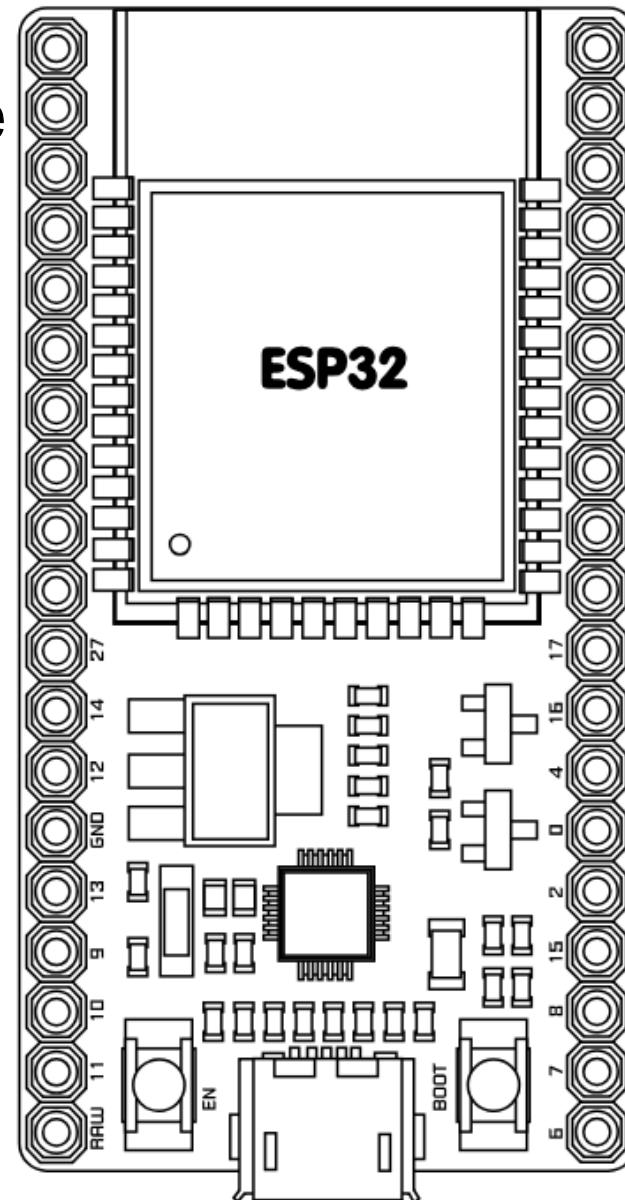
<https://www.cytron.io/p-node32-lite-wifi-and-amp;-bluetooth-development-kit>



## Node32Lite

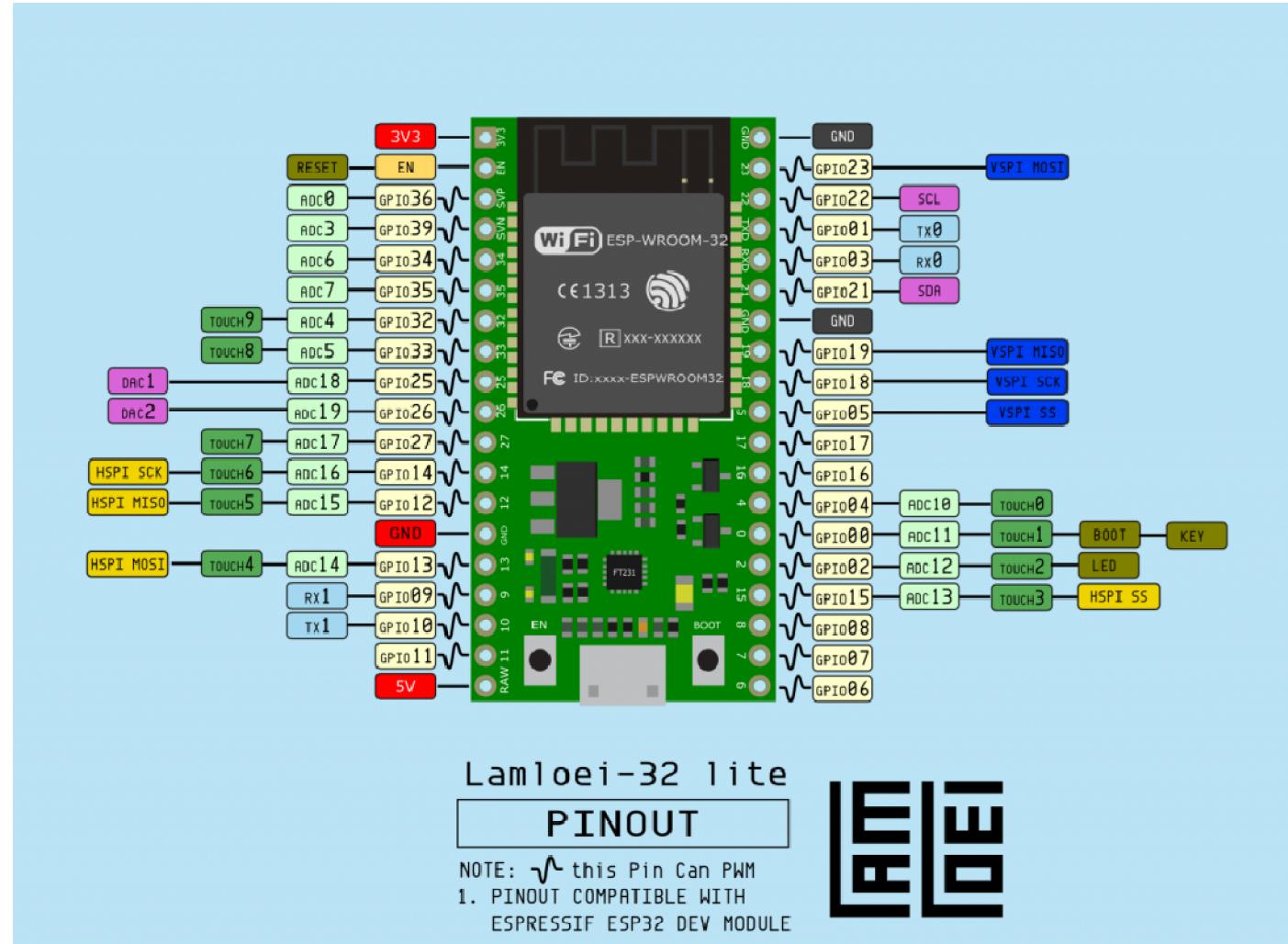
### Feature

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 Pinout





## Reverse Pin

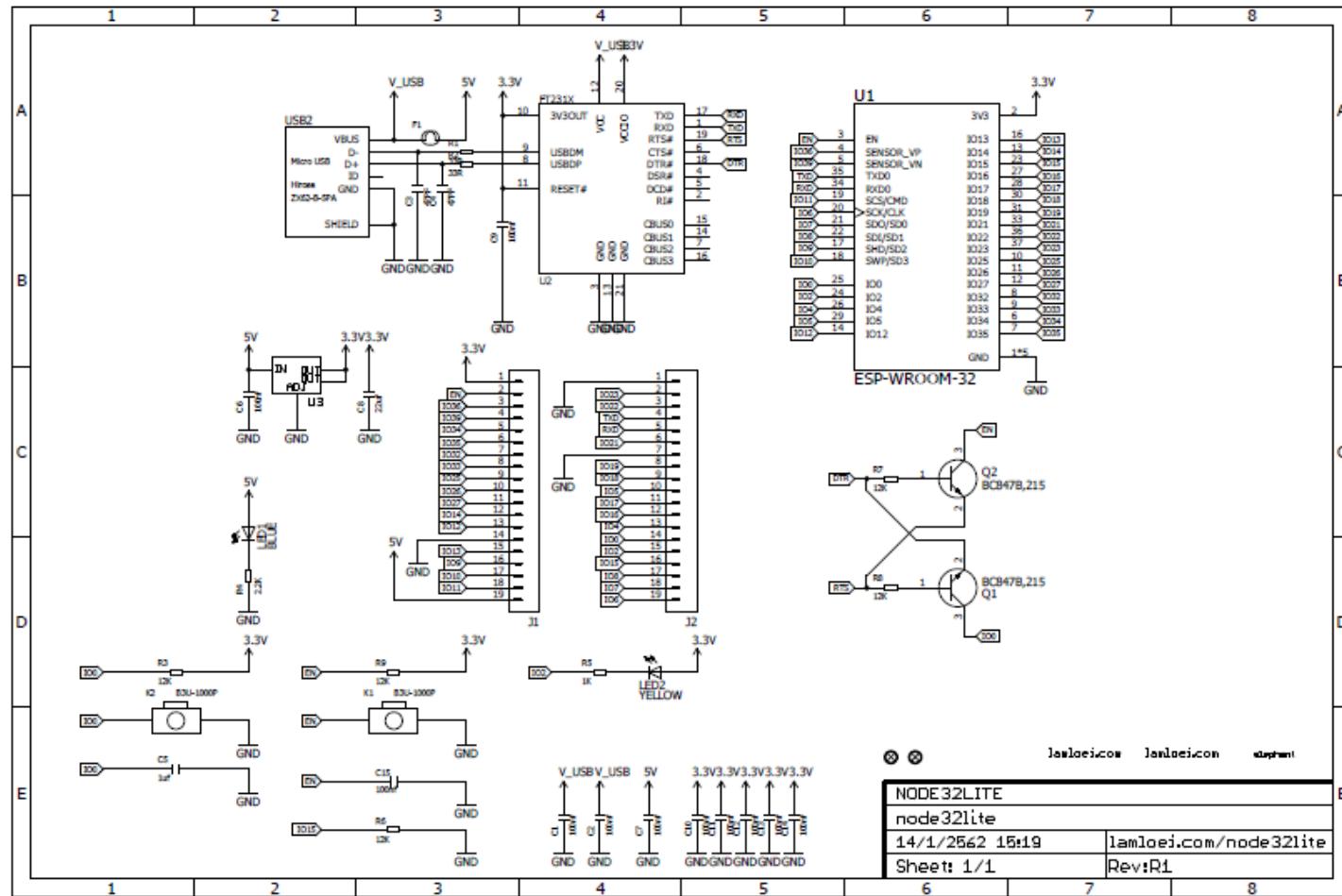
- Input only – 36, 39, 34, 35
- Not Used – 6, 7, 8, 9, 10, 11
- IO Special – 0, 2, 5, 12, 15

กรณีเสียบไฟแล้วไม่ทำงานเลย

1. กดปุ่ม Reset 1 ครั้ง
2. หรือ ต่อ io0 เข้ากับ 3v3
3. หรือ ต่อ Capacitor 1uF หรือมากกว่า ระหว่าง EN กับ GND

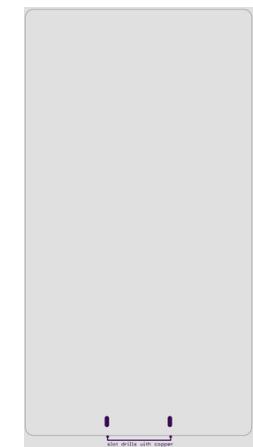
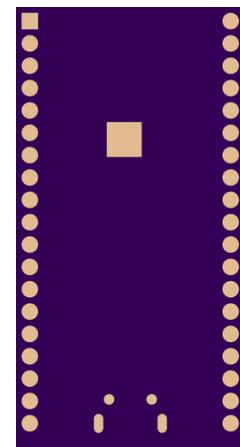
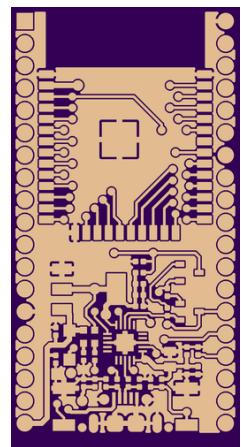
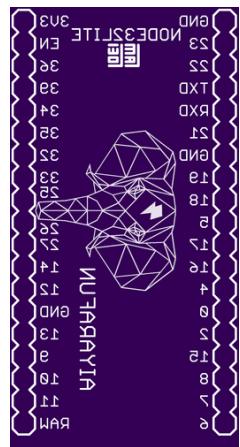
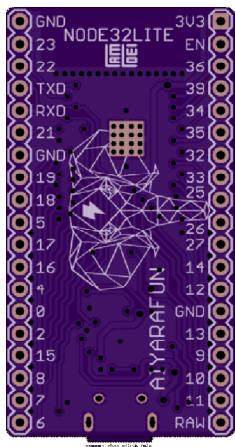
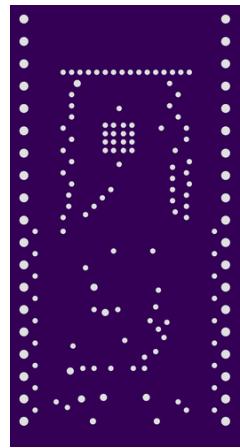
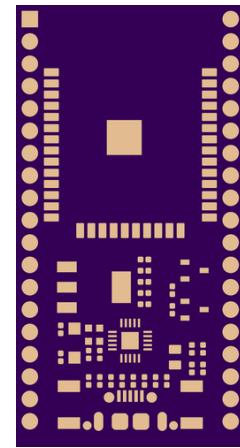
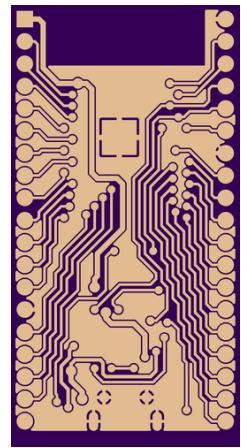
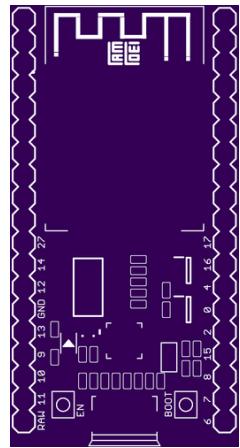
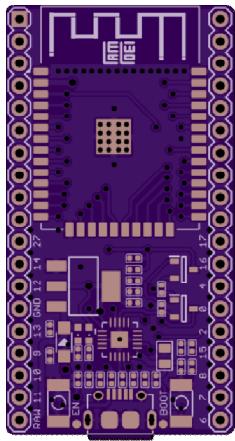


# Node32Lite Schematic



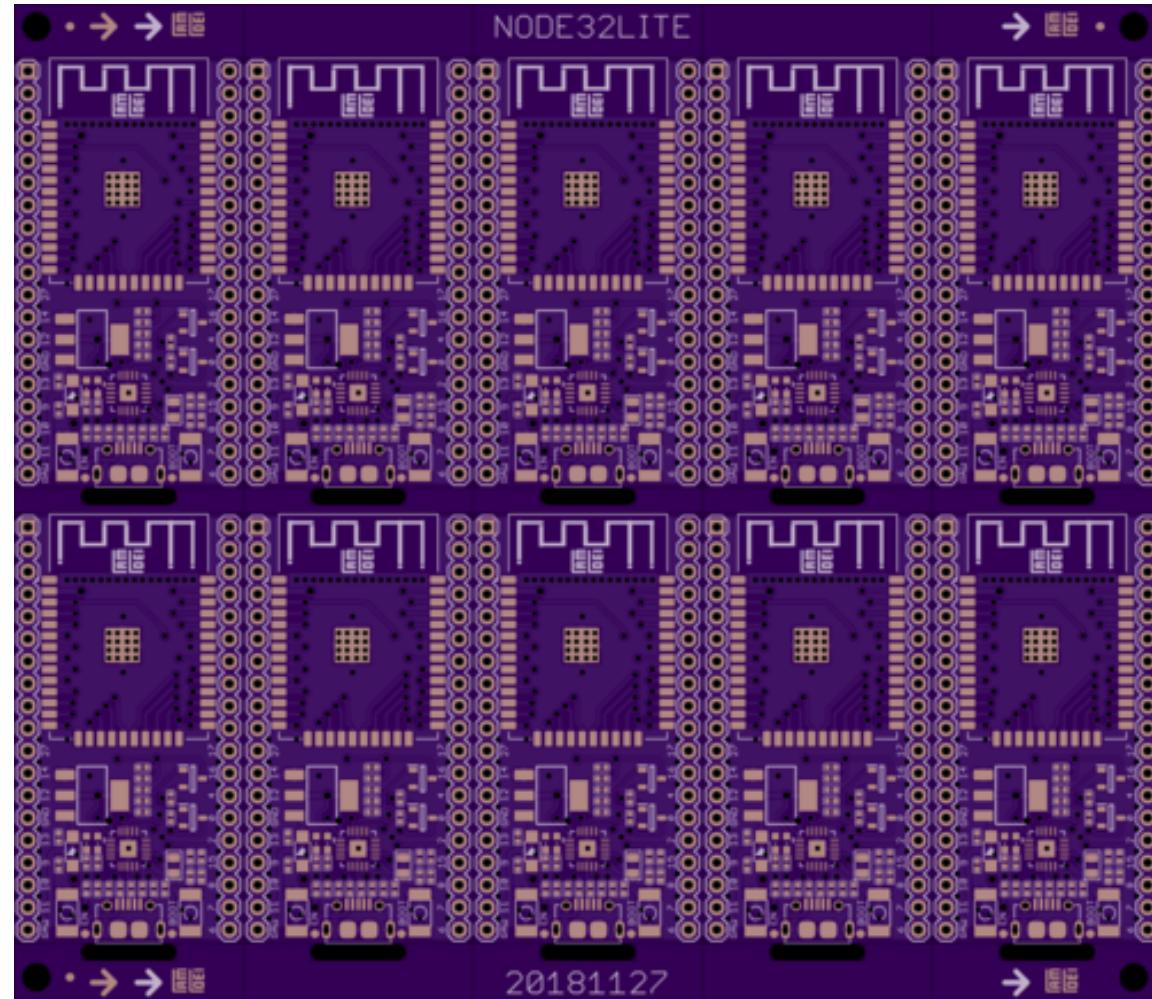


# Node32Lite Gerber





## Node32Lite Gerber Panel

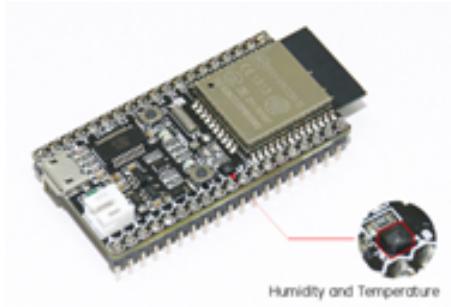




## Node32 Family



Node32Pico



Node32s Plus



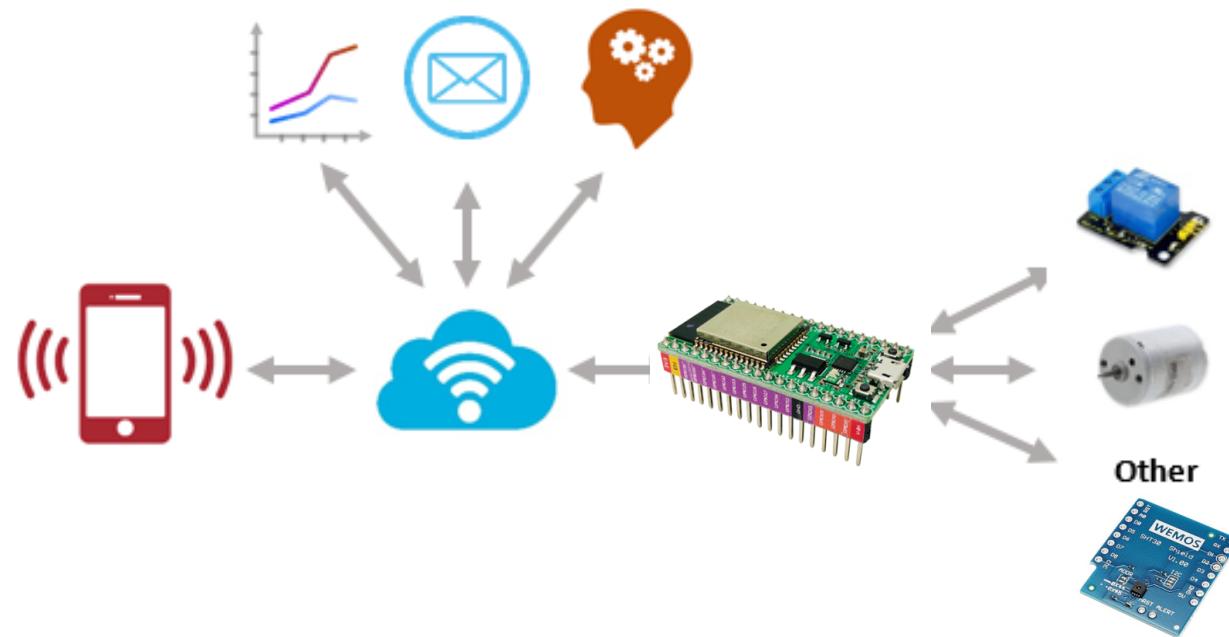
Node32s



NodeWiFi

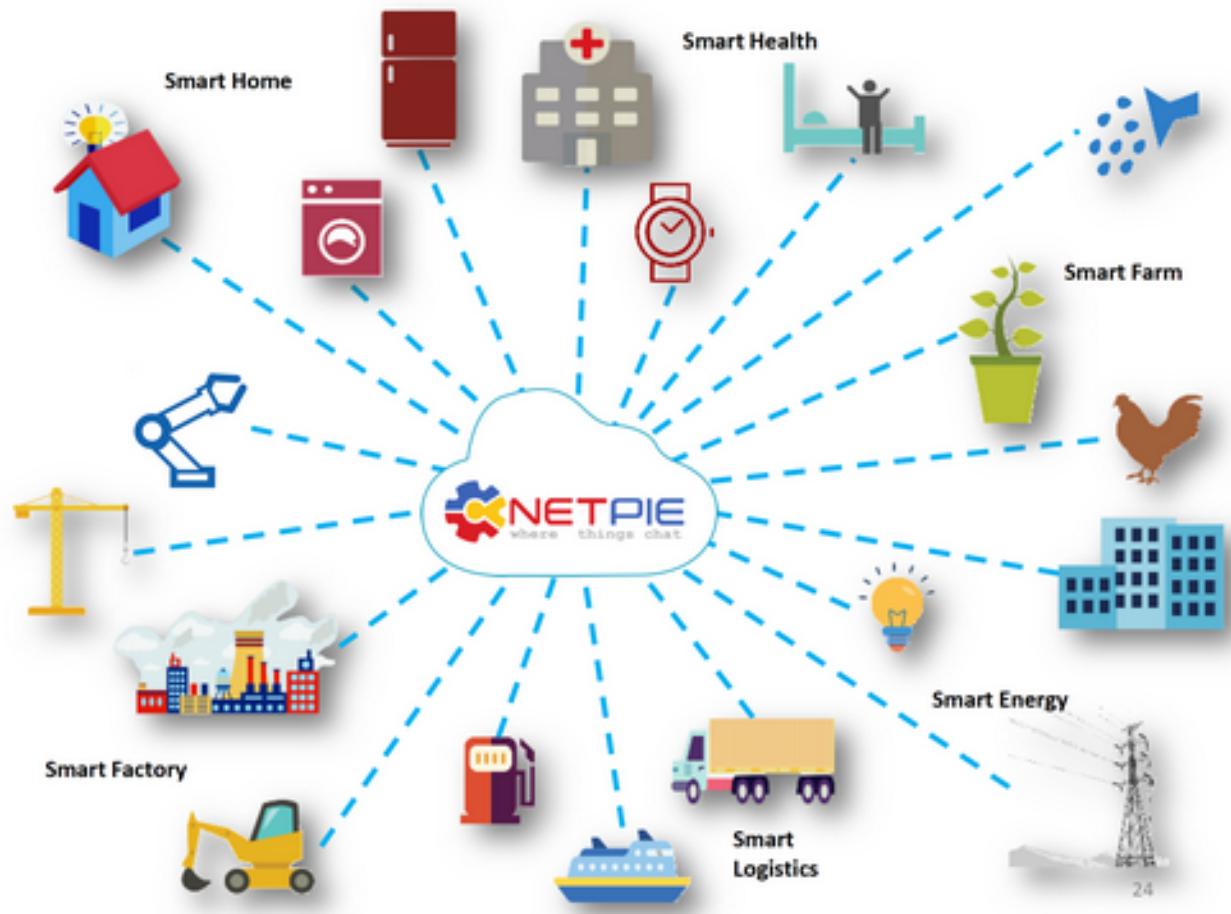


# เปลี่ยน ค่ากायภาพ > ตัวเลข > สร้างกราฟ อีเมล แจ้งเตือน วิเคราะห์ ผ่านมือถือ





# Smart System





## แบบฝึกหัดที่ 1 - Smart System

1. ชื่อระบบ
2. อธิบายระบบ
3. ค่าที่ต้องการวัด – ควบคุม



# ดาวน์โหลดและติดตั้งไฟล์

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

The screenshot shows the Arduino website's navigation bar. The 'SOFTWARE' menu item is highlighted in white, indicating it is the current section. Other menu items include HOME, STORE, EDU, RESOURCES, COMMUNITY, and HELP. The top right corner features a search bar with a magnifying glass icon and the word 'Search'.

## Download the Arduino IDE

The screenshot displays the Arduino 1.8.8 software download page. On the left, there is a large teal circular icon containing a white infinity symbol with a minus sign on the left and a plus sign on the right. To the right of the icon, the text 'ARDUINO 1.8.8' is displayed in bold. Below this, a paragraph describes the Arduino Software (IDE) as open-source, Java-based software for writing and uploading code to Arduino boards, compatible with Windows, Mac OS X, and Linux. It also mentions that the software can be used with any Arduino board and provides a link to the 'Getting Started' page for installation instructions. On the right side of the page, there is a teal sidebar with download links for various operating systems: 'Windows Installer, for Windows XP and up', 'Windows ZIP file for non admin install', 'Windows app' (with a 'Get' button), 'Mac OS X 10.8 Mountain Lion or newer', 'Linux 32 bits', 'Linux 64 bits', 'Linux ARM', and links for 'Release Notes', 'Source Code', and 'Checksums (sha512)'.



# Arduino IDE

The screenshot shows the Arduino IDE interface. The title bar reads "sketch\_feb25a | Arduino 1.8.8". The menu bar includes File, Edit, Sketch, Tools, and Help. The toolbar has icons for upload, download, and other functions. The code editor window displays the following sketch:

```
sketch_feb25a
File Edit Sketch Tools Help
sketch_feb25a
void setup() {
  // put your setup code here, to run once:
}

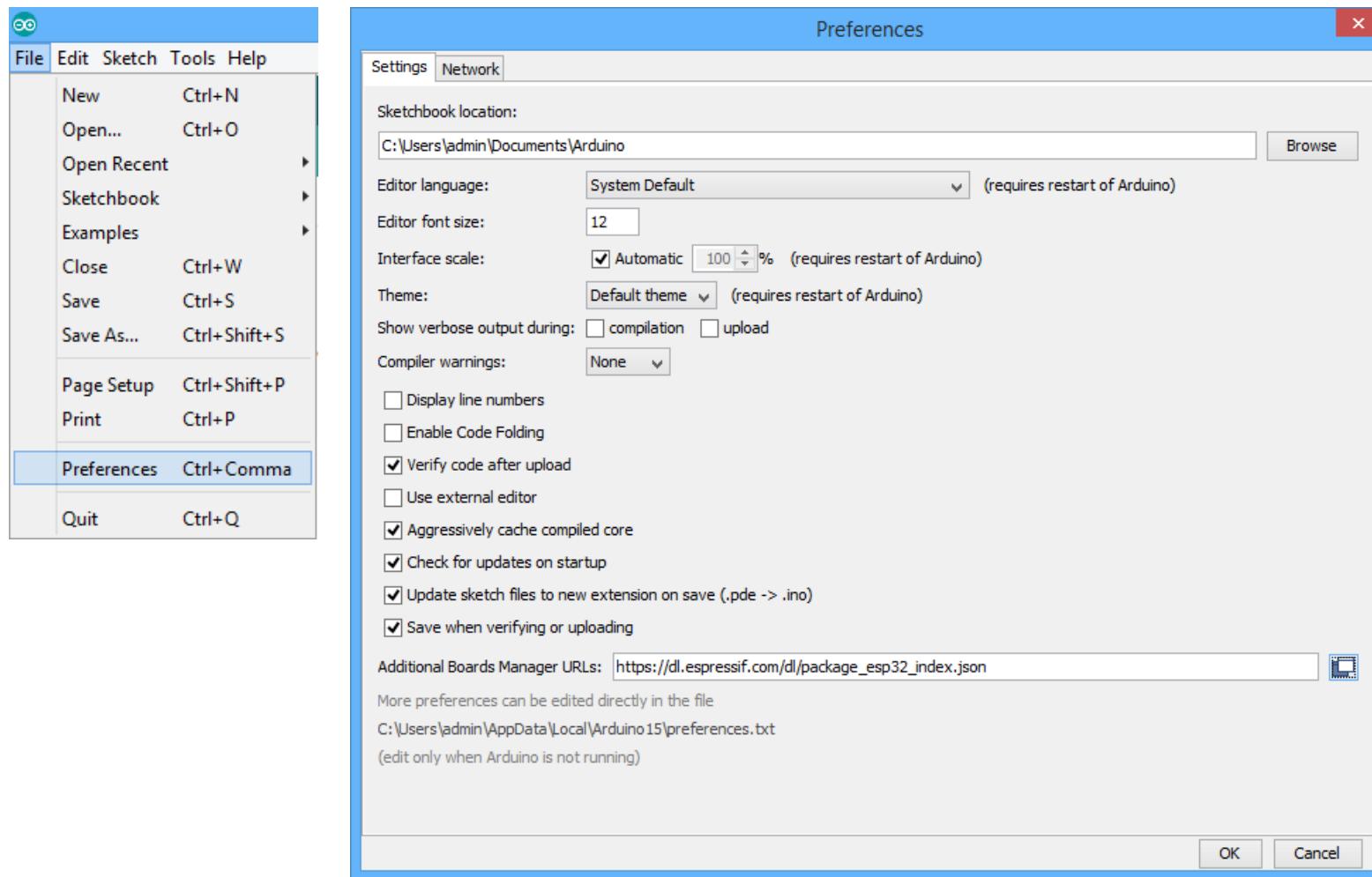
void loop() {
  // put your main code here, to run repeatedly:
}
```

The bottom status bar indicates "Arduino/Genuine Uno".



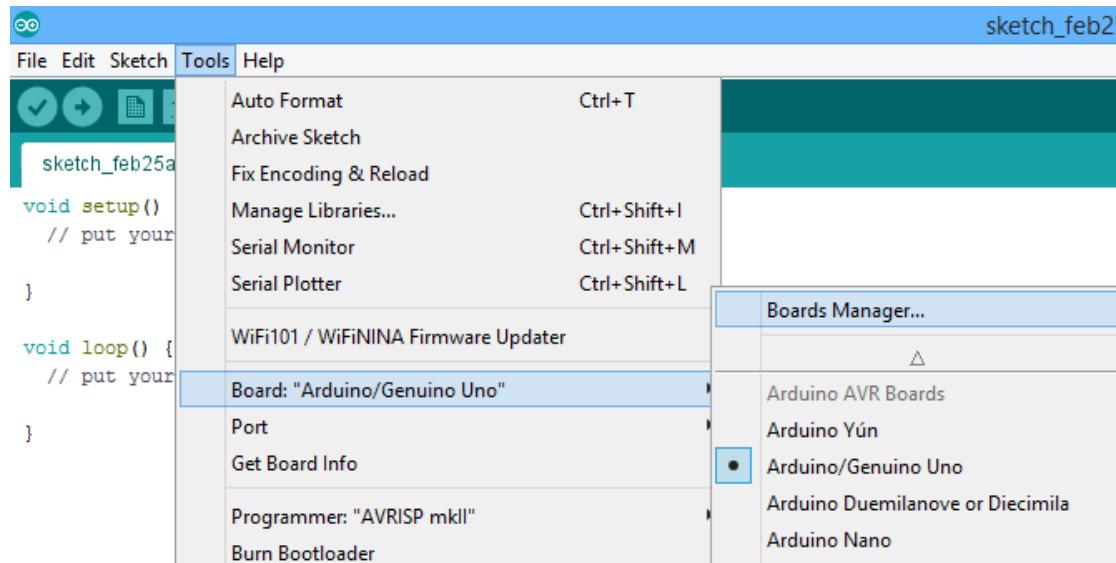
## File > Preferences

[https://dl.espressif.com/dl/package\\_esp32\\_index.json](https://dl.espressif.com/dl/package_esp32_index.json)



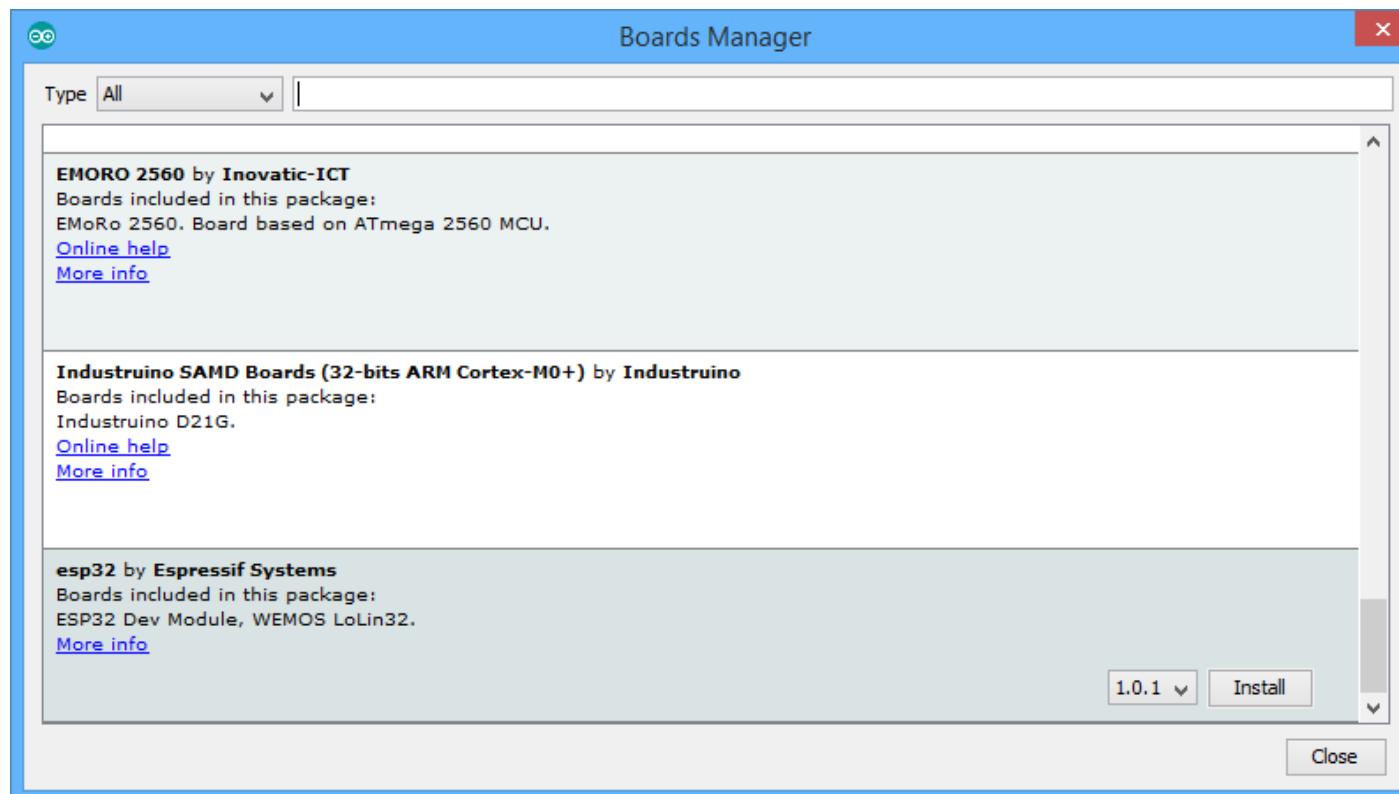


## Tools > Board > Board Manager...





คลิก Install ที่ esp32

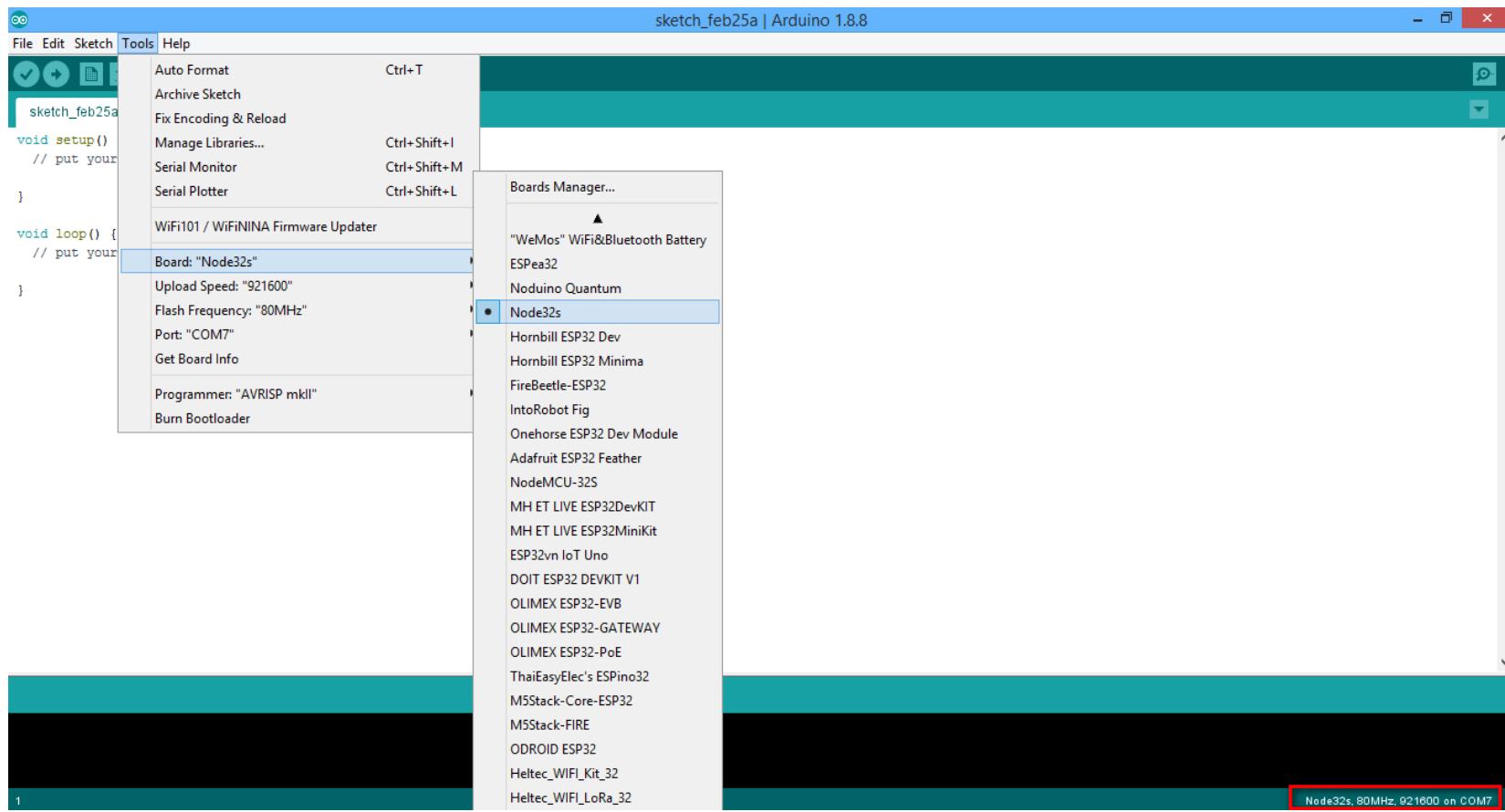




# ເສື່ອບັບອົບ Node32Lite

Tools > Board > Node32s

ເລືອກ Port





## ไฟล์เดอร์

- โฟลเดอร์ Arduino

C:\Users\[YOUR\_USER\_NAME]\Documents\Arduino

- ไฟล์เดอร์ Arduino IDE

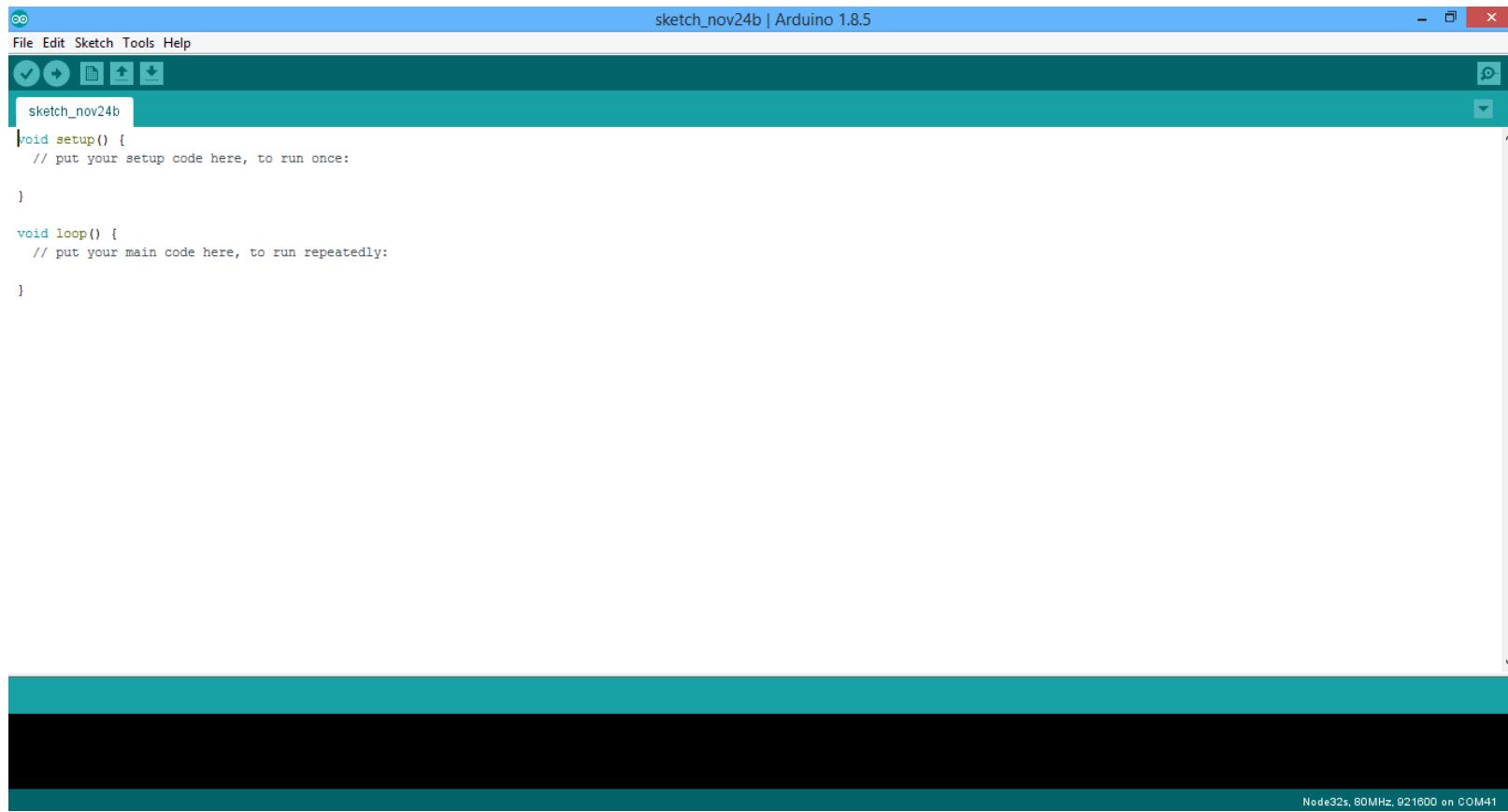
C:\Program Files (x86)\Arduino

C:\Users\[YOUR\_USER\_NAME]\AppData\Local\Arduino15



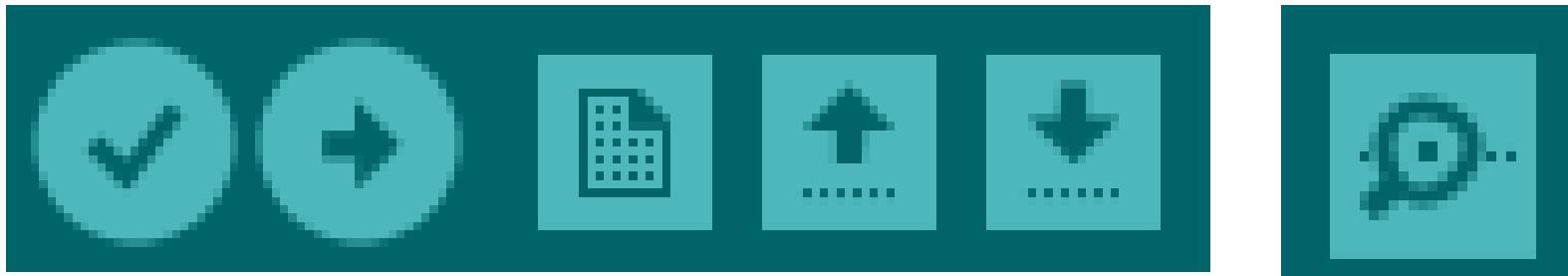
# หน้าจอ Arduino IDE

เปิดโปรแกรม Arduino IDE





## เมนูลัด



Verify

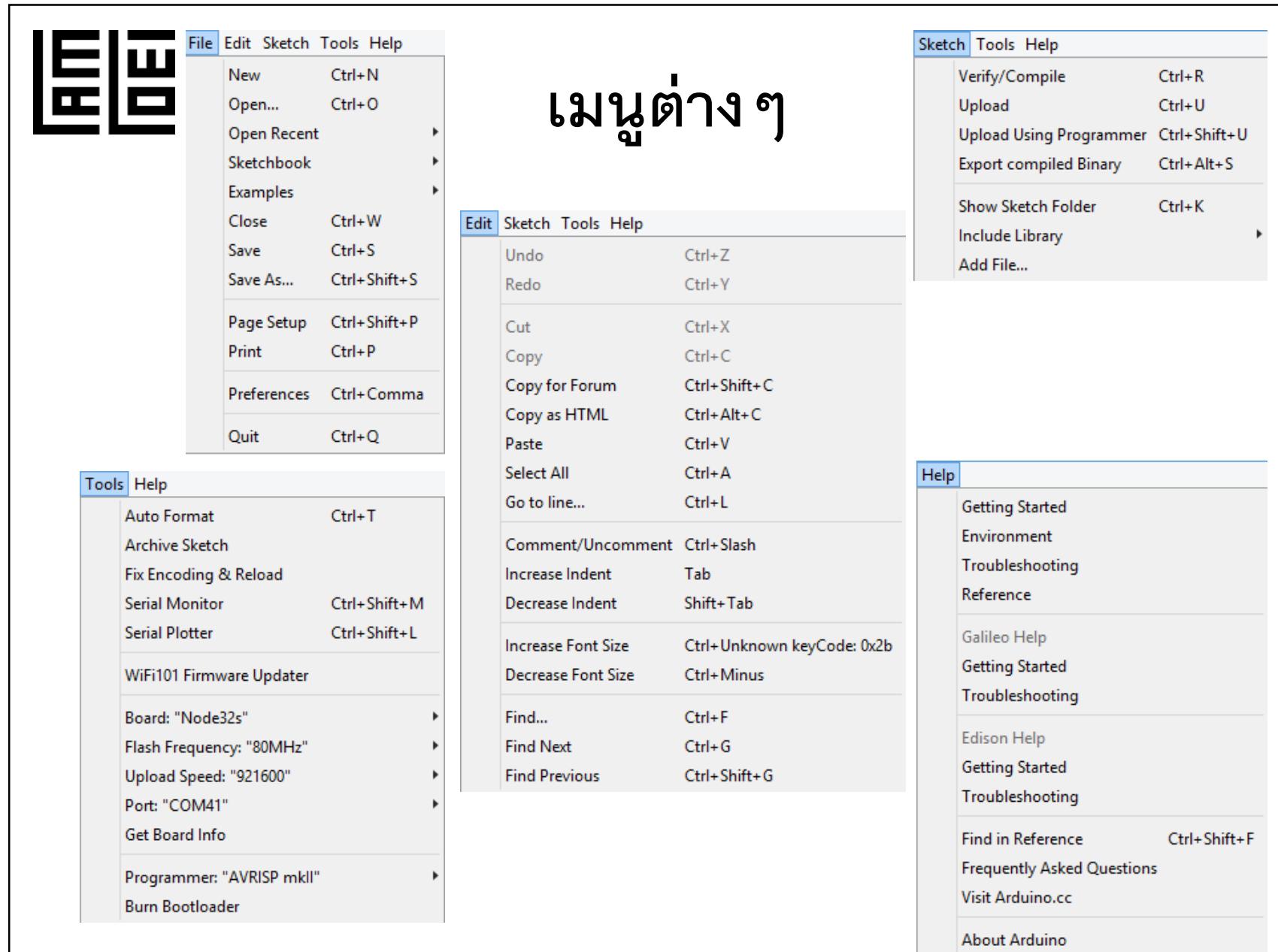
Upload

New

Open

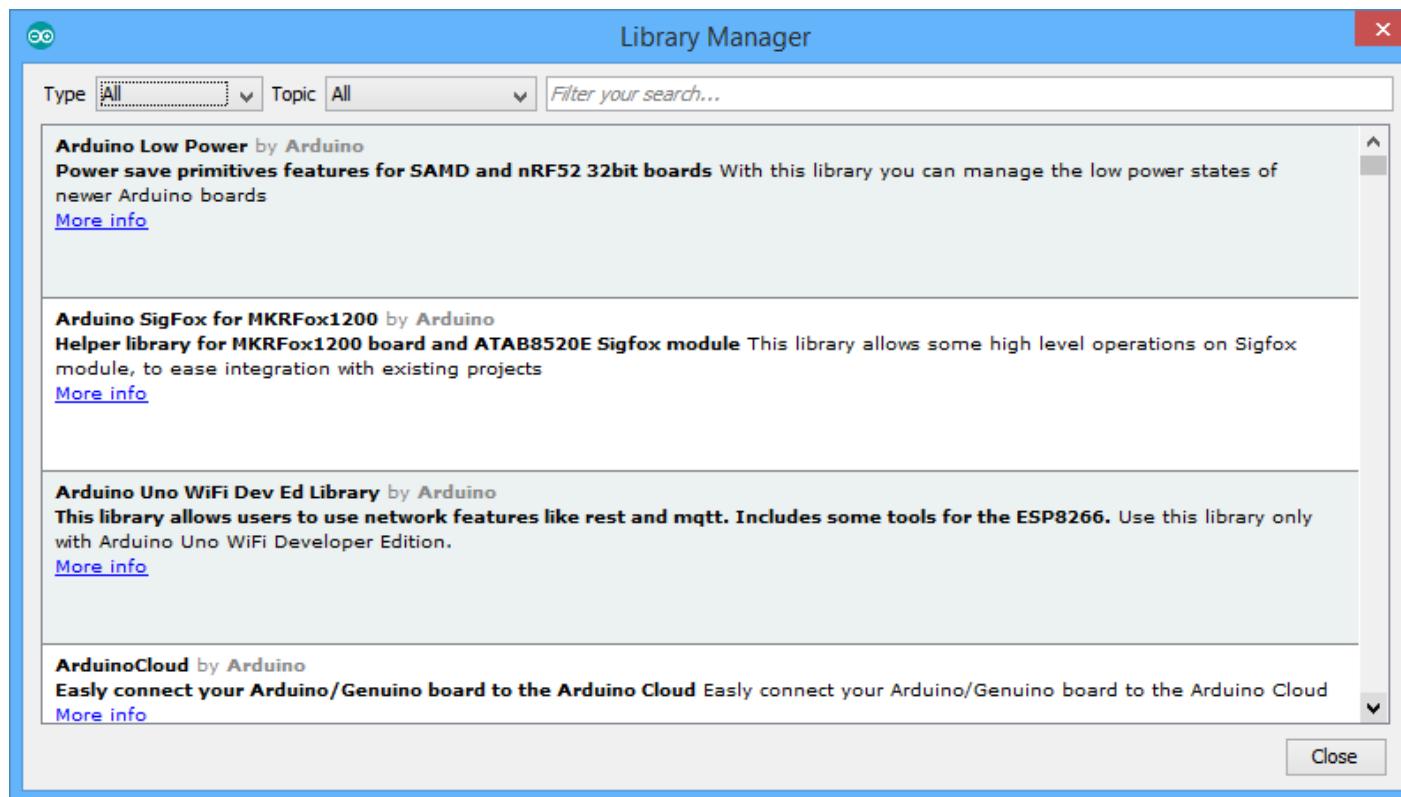
Save

Serial Monitor



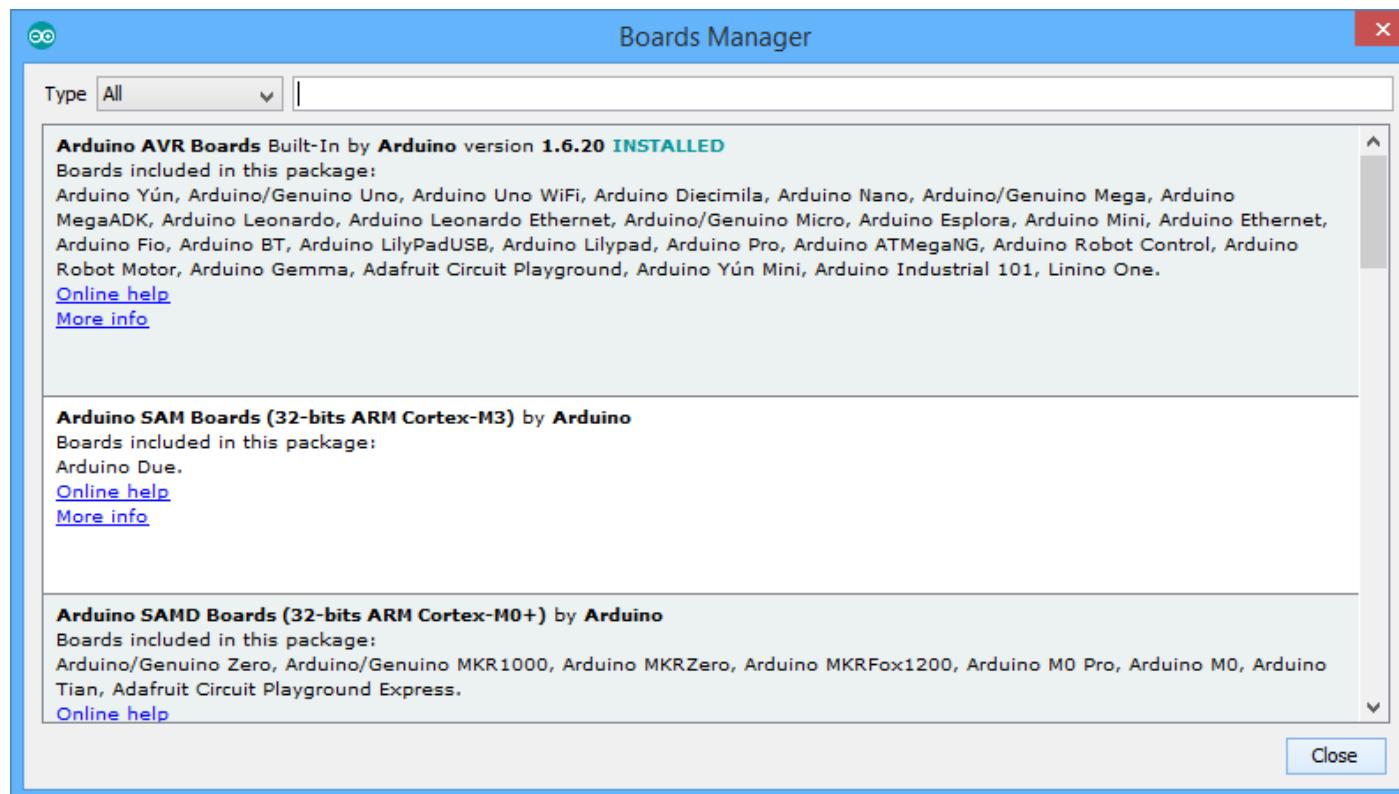


# Sketch > Include Library > Manage Libraries...



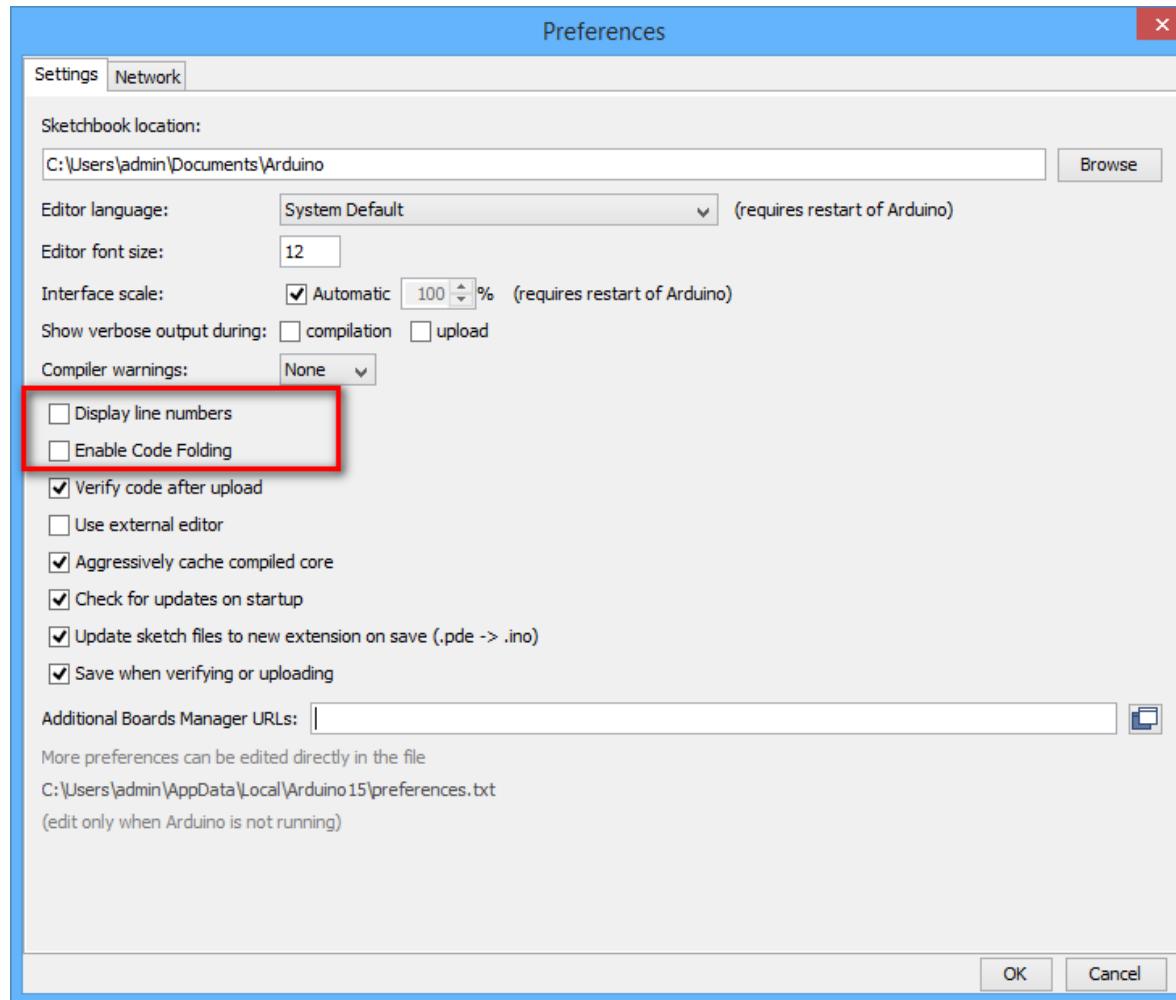


## Tools > Board: > Board Manager...





## File > Preferences





# waiting for download

เปิด Serial Monitor

กดปุ่ม EN ค้างไว้

กดปุ่ม BOOT ค้างไว้

ปล่อยปุ่ม EN

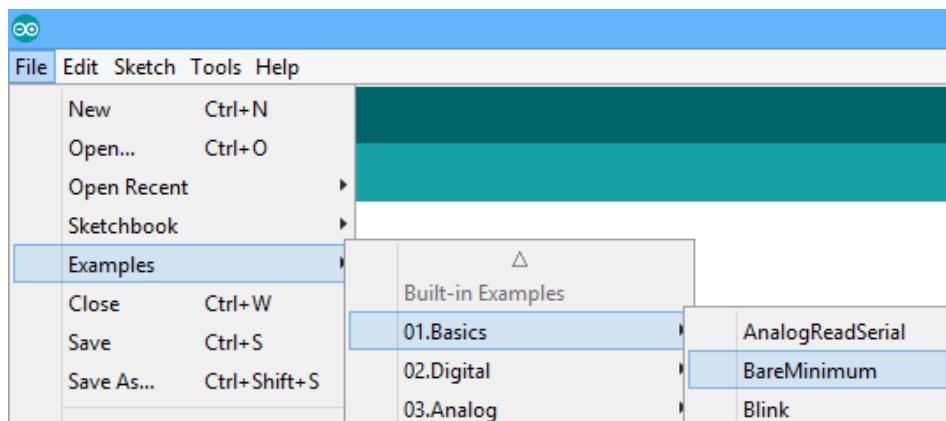
ปล่อยปุ่ม BOOT

...ขึ้นข้อความ waiting for download

กดแล้วปล่อยปุ่ม EN



# File > Examples > 01.Basics > BareMinimum



```
void setup() {
  // put your setup code here, to run once:

}

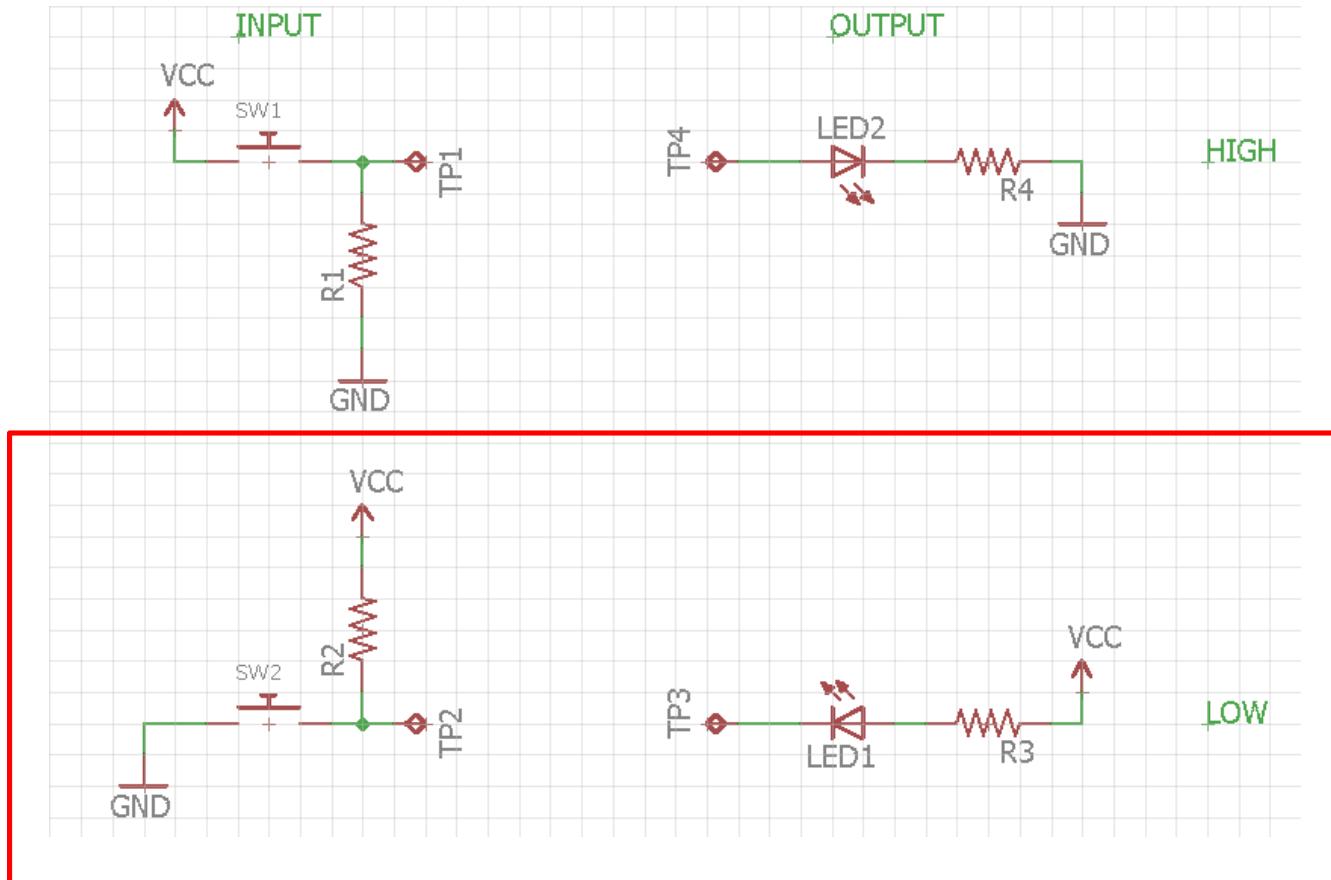
void loop() {
  // put your main code here, to run repeatedly:
}
```

void setup() {} - ทำงานครั้งเดียว

void loop() {} - ทำงานวนลูป



# Active High Active Low

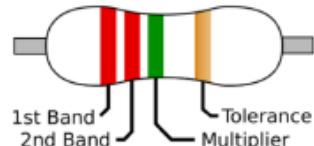
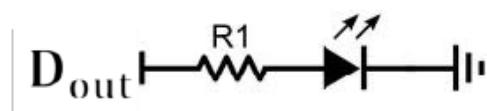


Active Low จะใช้แหล่งจ่ายภายนอก ทำให้ไม่เป็นภาระของ MCU

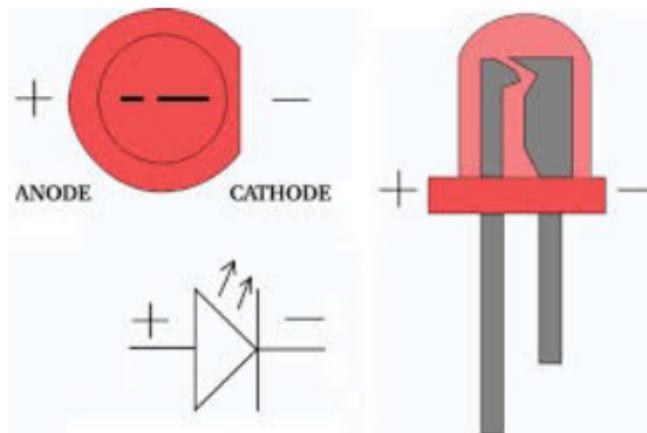


# LED & Resistor

LED & resistor

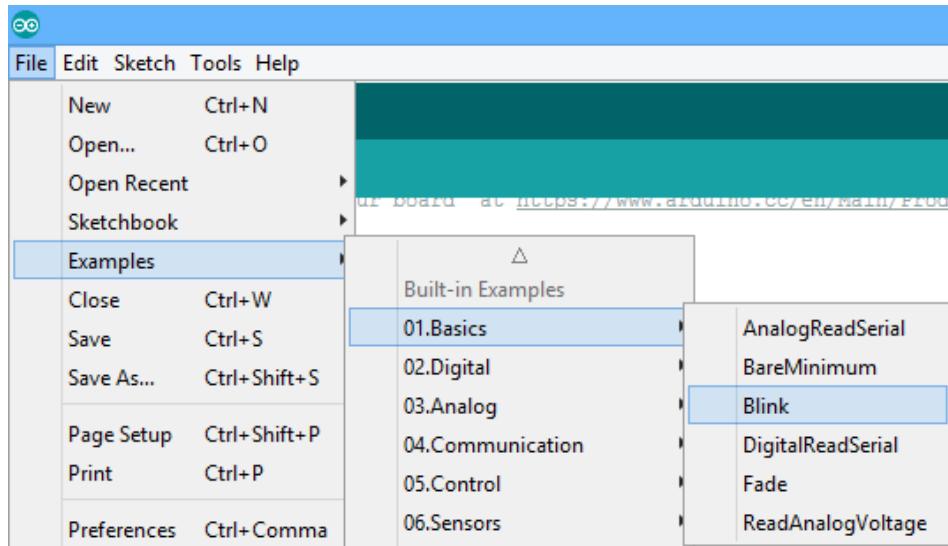


	1st/2nd Band	Multiplier	Tolerance
Black	0	$\times 10^0$	$\pm 1\%$
Brown	1	$\times 10^1$	-
Red	2	$\times 10^2$	$\pm 2\%$
Orange	3	$\times 10^3$	-
Yellow	4	$\times 10^4$	$\pm 0.5\%$
Green	5	$\times 10^5$	$\pm 0.25\%$
Blue	6	$\times 10^6$	$\pm 0.1\%$
Violet	7	$\times 10^7$	$\pm 0.05\%$
Grey	8	$\times 10^8$	-
White	9	$\times 10^9$	$\pm 5\%$
Gold	-	$\times 10^{-1}$	$\pm 10\%$
Silver	-	$\times 10^{-2}$	$\pm 20\%$
None	-	-	-





## File > Examples > 01.Basics > Blink



```
// 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
}
```

`LED_BUILTIN = 2`

ลองเปลี่ยนค่า `delay`

Upload

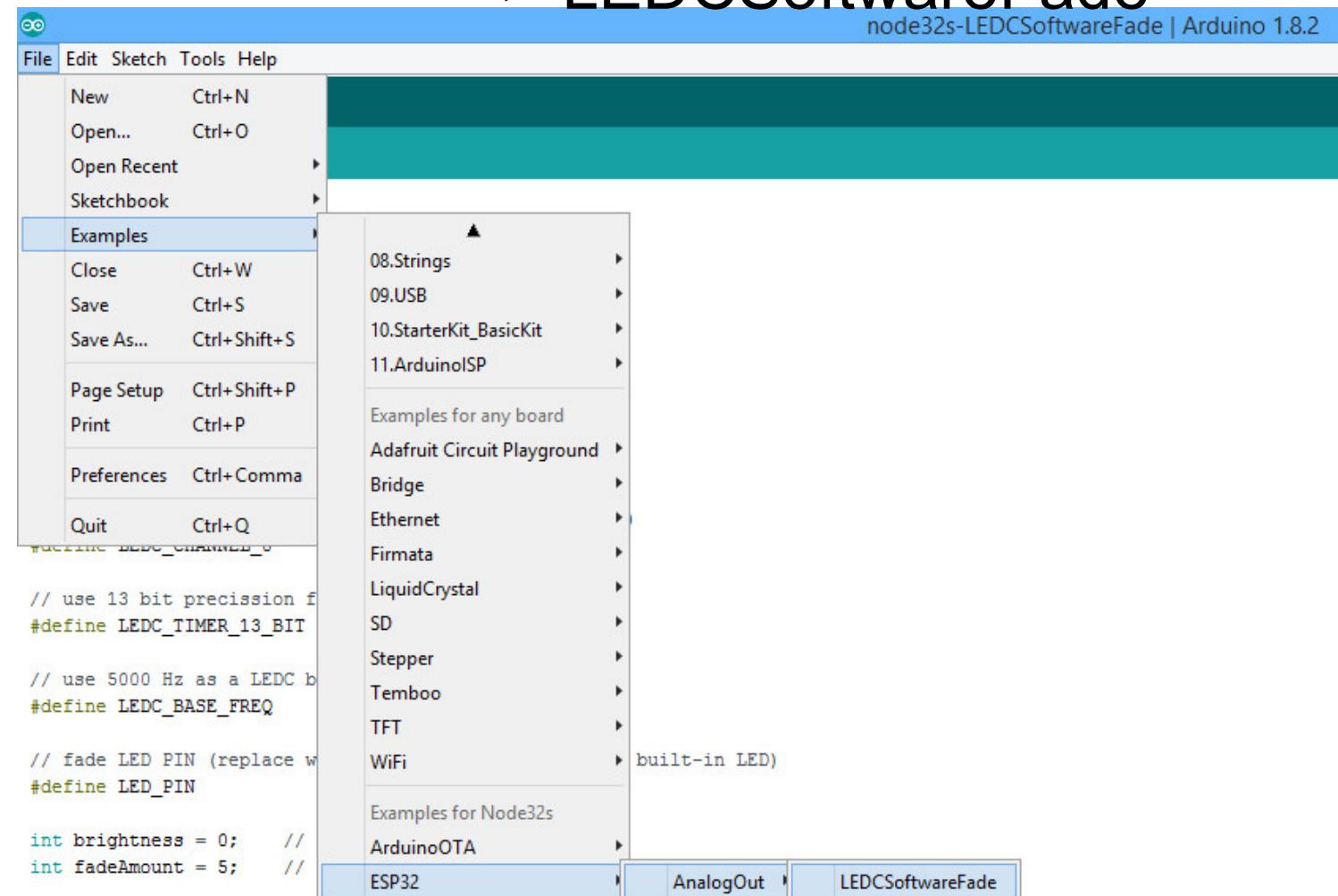
ดูความเปลี่ยนแปลง

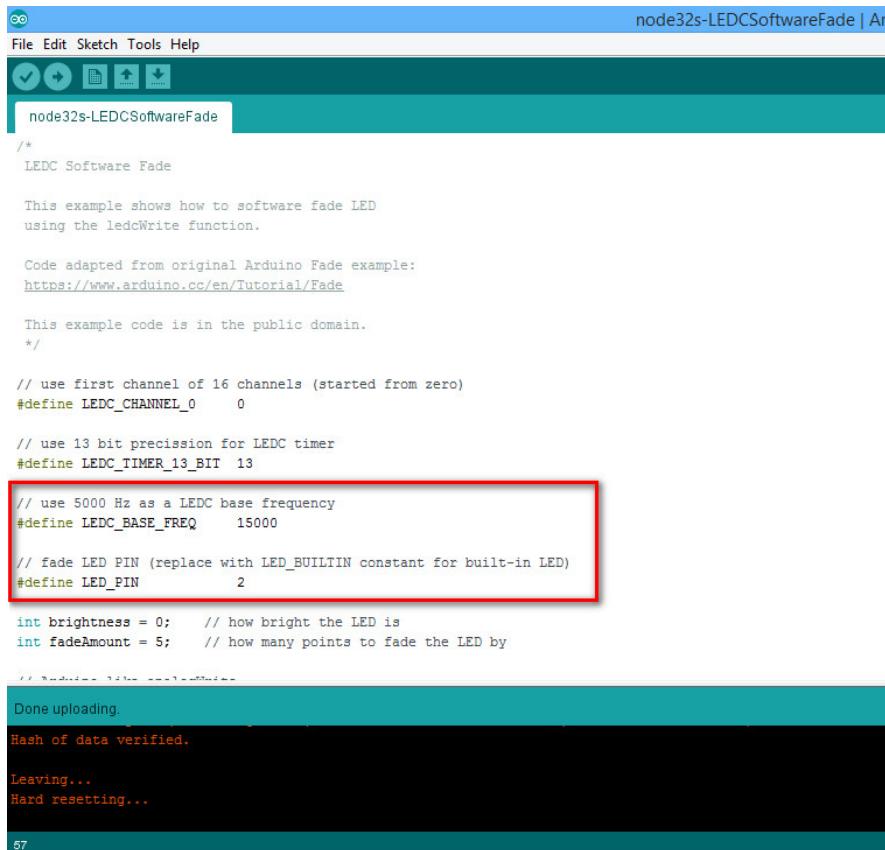


# Documents\Arduino\hardware\espressif\ esp32\variants\node32s\pins\_arduino.h

```
pins_arduino.h x
1 ifndef Pins_Arduino_h
2 define Pins_Arduino_h
3
4 include <stdint.h>
5
6 define EXTERNAL_NUM_INTERRUPTS 16
7 define NUM_DIGITAL_PINS 40
8 define NUM_ANALOG_INPUTS 16
9
10 define analogInputToDigitalPin(p) (((p)<20)?(esp32_adc2gpio[(p)]):-1)
11 define digitalPinToInterrupt(p) (((p)<40)?(p):-1)
12 define digitalPinHasPWM(p) (p < 34)
13
14 static const uint8_t LED_BUILTIN = 2;
15 define BUILTIN_LED LED_BUILTIN // backward compatibility
16
17 static const uint8_t KEY_BUILTIN = 0;
18
19 static const uint8_t TX = 1;
20 static const uint8_t RX = 3;
21
22 static const uint8_t SDA = 21;
23 static const uint8_t SCL = 22;
24
25 static const uint8_t SS = 5;
26 static const uint8_t MOSI = 23;
27 static const uint8_t MISO = 19;
28 static const uint8_t SCK = 18;
29
30 static const uint8_t A0 = 36;
31 static const uint8_t A3 = 39;
32 static const uint8_t A4 = 32;
33 static const uint8_t A5 = 33;
34 static const uint8_t A6 = 34;
35 static const uint8_t A7 = 35;
36 static const uint8_t A10 = 4;
37 static const uint8_t A11 = 0;
```

# File > Examples > ESP32 > AnalogOut > LEDCSoftwareFade





```
File Edit Sketch Tools Help
node32s-LEDSoftwareFade | Ar
node32s-LEDSoftwareFade
/*
LED Software Fade

This example shows how to software fade LED
using the ledcWrite function.

Code adapted from original Arduino Fade example:
https://www.arduino.cc/en/Tutorial/Fade

This example code is in the public domain.
*/
#define LEDC_CHANNEL_0      0
#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

Done uploading.
Hash of data verified.
Leaving...
Hard resetting...
57
```

```
#define LEDC_BASE_FREQ 15000
```

```
#define LED_PIN 2
```



# File > Examples > 02.Digital > BlinkWithoutDelay

The screenshot shows the Arduino IDE interface with the title bar "node32lite\_BlinkWithoutDelay | Arduino 1.8.8". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, refresh, and other functions. The code editor window contains the "node32lite\_BlinkWithoutDelay" sketch. The code is as follows:

```
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);
    }
}

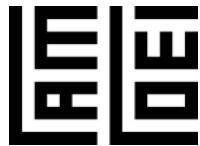
Done uploading.

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



## แบบฝึกหัดที่ 2 – ลองต่อ LED

1. ต่อ LED ที่ขาของ Node32Lite ให้ LED สว่าง ดับ ตามลำดับ



### Structure

```
void setup()
void loop()
```

### Control Structures

```
if (x<5) {}
for(int i = 0; i < 255; i++) {}
while((x < 6) {}
```

### Further Syntax

```
// Single line comment
/* ... */ Multi line comment
#define ANSWER 42
#include <myLib.h>
```

### General Operators

=	assignment
+, -	addition, subtraction
*, /	multiplication, division
%	modulo
==	equal to
!=	not equal to
<	less than
<=	less than or equal to

### Pointer Access

&	reference operator
*	dereference operator

### Bitwise Operators

&	bitwise AND
	bitwise OR
^	bitwise XOR
~	bitwise NOT

### Compound Operators

++	Increment
--	Decrement
+=	Compound addition
&=	Compound bitwise AND

### Constants

HIGH, LOW	
INPUT, OUTPUT	
true, false	
53 : Decimal	
B11010101: Binary	
0x5BA4: Hexadecimal	

### Data Types

void	0, 1, false, true
boolean	e.g. 'a' -128 → 127
char	
unsigned char	0 → 255
int	-32.768 → 32.767
unsigned int	0 → 65535
long	-2.147.483.648 → 2.147.483.647
float	-3.4028235E+38 → 3.4028235E+38
sizeof (myint)	returns 2 bytes

### Arrays

```
int myInts[6];
int myPins[] = {2,4,8,5,6};
int myVals[6] = {2,-4,9,3,5};
```

### Strings

```
char S1[15];
char S2[8] = "A','r','d','u','i','n','o';
char S3[8] = "A','r','d','u','i','n','o','\0';
char S4[8] = "Arduino";
char S5[8] = "Arduino";
char S6[15] = "Arduino";
```

### Conversion

char()	int0	long0
byte()	word0	float0

### Qualifiers

static	Persist between calls
volatile	Use RAM (nice for ISR)
const	Mark read-only
PROGMEM	Use flash memory

## ARDUINO CHEAT SHEET

JEROEN DOGGEN, AP UNIVERSITY COLLEGE ANTWERP



ARTESIS PLANTIJN  
HOGESCHOOL ANTWERPEN

### Interrupts

attachInterrupt(interrupt, function, type)	
detachInterrupt(interrupt)	
boolean(interrupt)	
interrupts()	
noInterrupts()	

### Advanced I/O

tone(pin, freqhz)	
tone(pin, freqhz, duration_ms)	
noTone(pin)	
shiftOut(dataPin, clockPin, how, value)	
unsigned long pulseIn(pin, [HIGH,LOW])	

### Time

unsigned long millis()	50 days overflow
unsigned long micros()	70 min overflow
delay(ms)	
delayMicroseconds(us)	

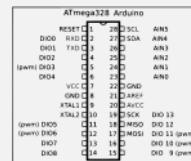
### Math

min(x,y)	max(x,y)	abs(x)
sin(rad)	cos(rad)	tan(rad)
pow(base, exponent)		
map(val, fromL, fromH, toL, toH)		
constrain(val, fromL, toH)		

### Pseudo Random Numbers

randomSeed(seed)	
long random(max)	
long random(min, max)	

### ATmega328 Pinout



### I/O Pins

	Uno	Mega
# of IO	14 + 6	54 + 11
Serial Pins 3	0 → RX, 1 → TX	RX1 → RX4 2,3 → TX1 → TX4 5,6 → RX5 → RX8 10 → TX5 → TX8
Interrupts	2,3	
PWM Pins	5,6 → 9,10 → 3,11	0 → 13
SPI (MOSI, MISO, SCK)	10 → 13	50 → 53
I2C (SDA, SCL)	A4, A5	20,21

### Analog I/O

analogReference(EXTERNAL, INTERNAL)	
analogRead(pin)	
analogWrite(pin, value)	

### Digital I/O

pinMode(pin, [INPUT, OUTPUT])	
digitalRead(pin)	
digitalWrite(pin, value)	

### Serial Communication

Serial.begin(speed)	
Serial.print("Text")	
Serial.println("Text")	

### Websites

forum.arduino.cc	
playground.arduino.cc	
arduino.cc/en/Reference	

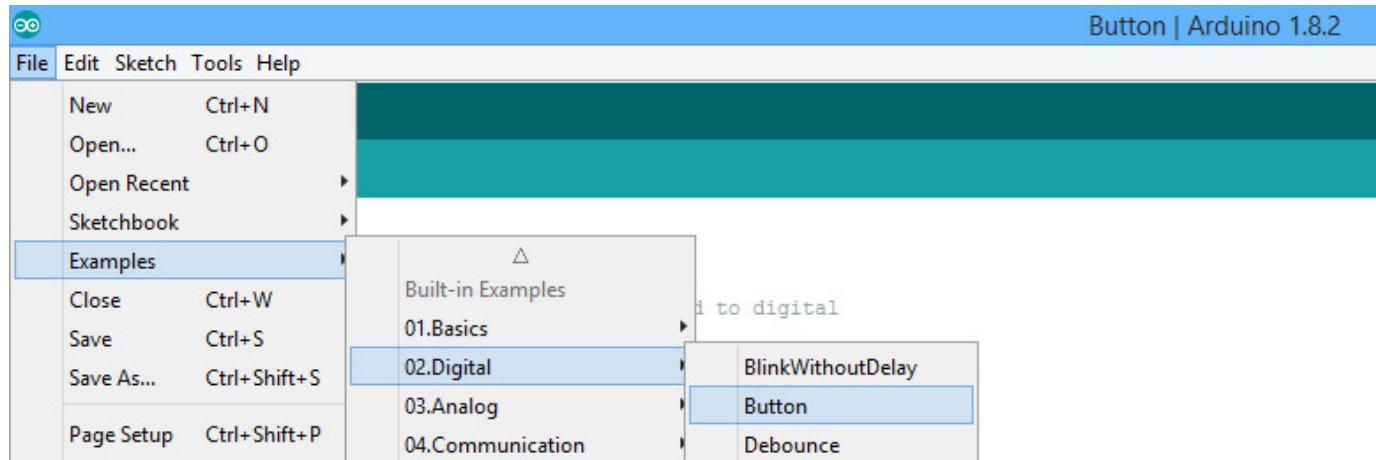
### Arduino Uno Board



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



## File > Examples > 02.Digital > Button



```
const int buttonPin = 0;
```

```
const int ledPin = 2;
```

กดปุ่มไฟสว่าง ปล่อยปุ่มไฟดับ



## แบบฝึกหัดที่ 3

- ให้ กดปุ่มไฟดับ ปล่อยปุ่มไฟสว่าง
- กดปุ่มไฟดับ กดปุ่มไฟสว่าง กดปุ่มไฟดับ กดปุ่มไฟสว่าง





## แบบฝึกหัดที่ 4

ให้แสดงที่ Serial Monitor

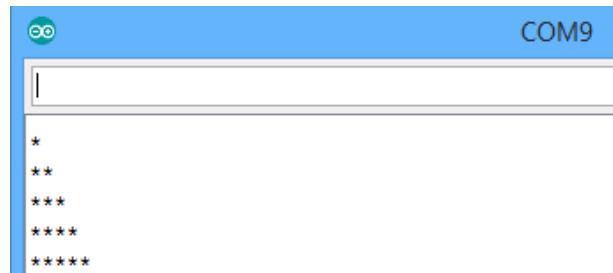
\*

\*\*

\*\*\*

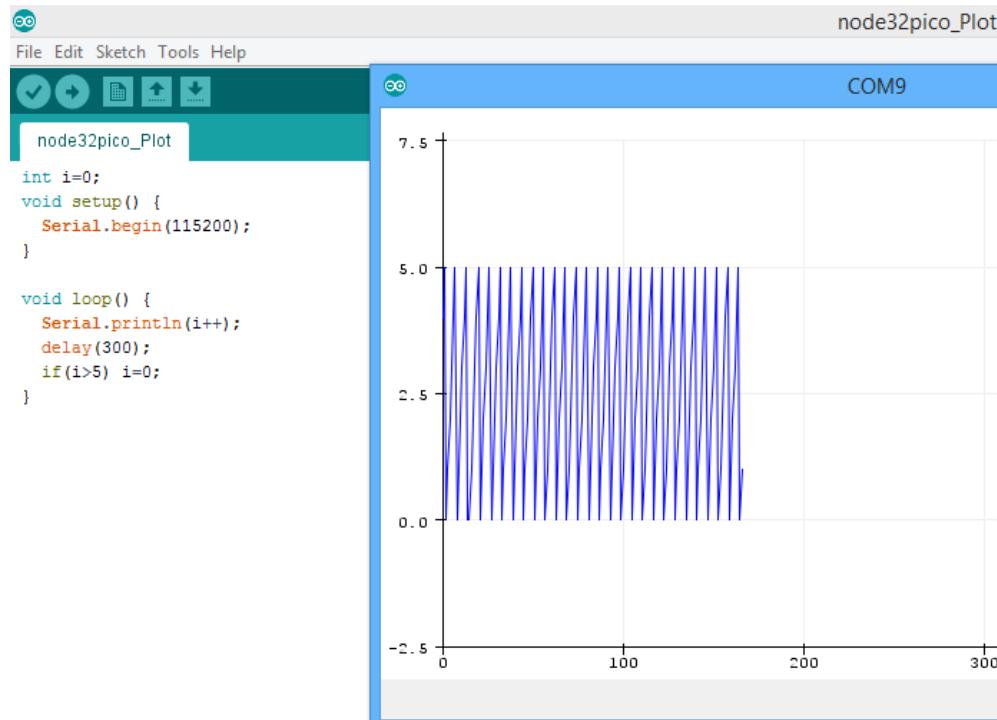
\*\*\*\*

\*\*\*\*\*





## Tools > Serial Plotter



พิมพ์ค่าสั่ง

เมนู Tools > Serial Plotter

ปรับค่า Serial 115200 ให้ตรงกัน



## สรุปขั้นตอน

- File > Examples
- ปรับค่า
- Upload Program
- สังเกตความเปลี่ยนแปลง



## File > Examples > ESP32 > ChipID > GetChipID

The screenshot shows the Arduino IDE interface. The top bar displays the title "GetChipID | Arduino 1.8.5". The menu bar includes File, Edit, Sketch, Tools, and Help. Below the menu is a toolbar with icons for upload, refresh, and other functions. The main code editor window contains the following sketch:

```
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);
}
```

Below the code editor is the Serial Monitor window titled "COM41". It shows the output of the sketch, which repeats the message "ESP32 Chip ID = 8CCE03C40A24" three times. The monitor has a text input field and a "Send" button.

```
ESP32 Chip ID = 8CCE03C40A24
ESP32 Chip ID = 8CCE03C40A24
ESP32 Chip ID = 8CCE03C40A24
```



## File > Examples > ESP32 > Touch > TouchRead

The screenshot shows the Arduino IDE interface with the 'TouchRead' sketch open. The code is as follows:

```
// ESP32 Touch Test
// Just test touch pin - Touch0 is T0 which is on
// digital pin 13
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);
}
```

The serial monitor window is titled 'TouchRead | Arduino 1.8.5' and shows the output from the sketch. The output consists of a series of values: 61, 0, 61, 61, 61, 13, 15, 61, 18, 62, 62, 62, 62. The baud rate is set to 115200.

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



## File > Examples > ESP32 > DeepSleep > TimerWakeUp

```
File Edit Sketch Tools Help
TimerWakeUp | Arduino 1.8.5
COM41
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

Send
Autoscroll No line ending 115200 baud Clear output
```

```
/*
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_wakeup_cause_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 ULP program"); break;
        default : Serial.println("Wakeup was not caused by deep sleep"); break;
    }
}
```



## File > Example > ESP32 > HallSensor

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

- Sketch Name:** HallSensor
- IDE Version:** HallSensor | Arduino 1.8.5
- Serial Monitor:** COM41
- Sketch Content (approximate code):**

```
//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
}
```

- Serial Monitor Output:** The output shows alternating values of 27 and 24, followed by a single value of 2.
- Serial Monitor Baud Rate:** 9600 baud (highlighted with a red box).

ปรับ Serial 9600



# File > Examples > 01.Basics > DigitalReadSerial

```

File Edit Sketch Tools Help
New Ctrl+N
Open... Ctrl+O
Open Recent
Sketchbook
Examples
Close Ctrl+W
Save Ctrl+S
Save As... Ctrl+Shift+S
Page Setup Ctrl+Shift+P
File Edit Sketch Tools Help
node32pico_DigitalReadSerial
/*
  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. Give it a name.
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's pin an input:
  pinMode(pushButton, INPUT);
}

int pushButton = 0;
Serial.begin(115200);
pinMode(pushButton, INPUT);
}

```

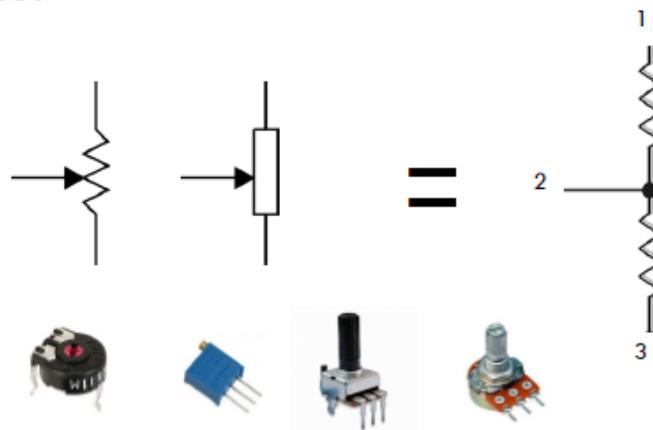
int pushButton = 0;  
Serial.begin(115200);

กดปุ่ม BOOT  
ดูความเปลี่ยนแปลง



# Volume

Potentiometer





# File > Examples > 01.Basics > AnalogReadSerial

The screenshot shows the Arduino IDE interface with the sketch `node32lite_AnalogReadSerial` loaded. The code reads analog input from pin A0 and prints the value to the Serial Monitor. The monitor window shows the value 4095 repeated multiple times, followed by a single value of 40.

```
/*
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(115200);
}

// 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 > EEPROM > eeprom\_write

The screenshot shows the Arduino IDE interface. The left pane displays the code for the `eeprom_write` example. The right pane shows the Serial Monitor window titled "COM9". The monitor displays several lines of binary data, each preceded by a timestamp and a message indicating 64 bytes written to Flash and their corresponding values.

```
// 0 to 1023 and each byte of the EEPROM can on
// value from 0 to 255.
// int val = analogRead(10) / 4;
int val = byte(random(10020));
// write the value to the appropriate byte of t
// these values will remain there when the boar
// turned off.
EEPROM.write(addr, val);
Serial.print(val); Serial.print(" ");
// advance to the next address. there are 512
// the EEPROM, so go back to 0 when we hit 512.
// save all changes to the flash.
addr = addr + 1;
if (addr == EEPROM_SIZE)
{
    Serial.println();
    addr = 0;
    EEPROM.commit();
```

node32pico\_eeprom\_write | Arduino 1.8.5

File Edit Sketch Tools Help

COM9

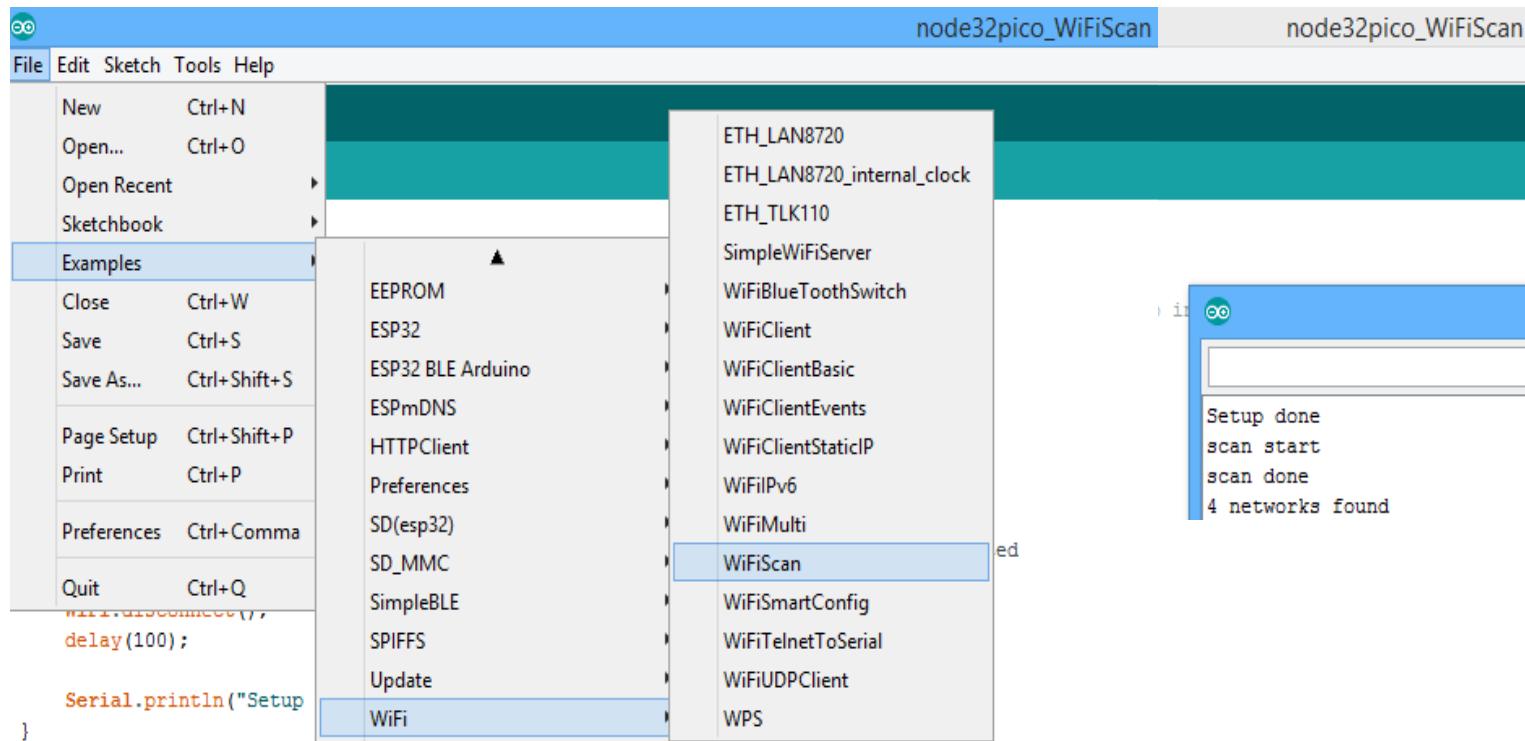
60 19 219 79 70 86 181 243  
64 bytes written on Flash . Values are:  
75 74 95 73 144 119 192 54 221 49 122 208 170 49 121 198 248 39 124  
-----  
150 230 240 216 233 126 130 241 82 196 11 20 194 87 252 139 116 252  
64 bytes written on Flash . Values are:  
150 230 240 216 233 126 130 241 82 196 11 20 194 87 252 139 116 252  
-----  
184 24 88 41 29 18 44 145 60 134 254 38 80 55 68 165 27 43 32 194 19  
64 bytes written on Flash . Values are:  
184 24 88 41 29 18 44 145 60 134 254 38 80 55 68 165 27 43 32 194 19  
-----  
120 167 223 193 61 212 37 18 229 99 8 104 68 4 109 7 232 126 81 14 5

< >

Autoscroll      No line ending      115200 baud      Clear output



# File > Examples > WiFi > WiFiScan





# Node32Pico SoftAP

The screenshot displays three windows related to the Node32Pico SoftAP project:

- Code Editor:** Shows the Arduino IDE code for the `node32pico_softap` sketch. The code initializes a WiFi soft access point with SSID "Node32Pico" and password "123456789". It then prints the IP address to the serial port.
- Serial Monitor:** A terminal window titled "COM9" showing the output of the sketch. It displays "IP address: 192.168.4.1".
- WiFi Settings:** A dark-themed WiFi configuration interface. It shows a connection to "lamloeiWIFI" (Connected) and a new entry for "Node32Pico". It includes sections for "Networks", "Airplane mode" (set to Off), "Connections", and "Wi-Fi" (set to On).



# File > Examples > WiFi > WiFiMulti

The screenshot shows the Arduino IDE interface with the sketch `node32pico_WiFiMulti` open. The code implements the WiFiMulti library to connect to a list of Wi-Fi networks. It includes a setup where it prints "Connecting Wifi..." and then runs a loop to check for connection. Upon successful connection, it prints "WiFi connected" and its IP address.

```
* This sketch tries to Connect to the best AP based on a given list
*/
#include <WiFi.h>
#include <WiFiMulti.h>

WiFiMulti wifiMulti;

void setup()
{
    Serial.begin(115200);
    delay(10);

    wifiMulti.addAP("lamloeiWIFI", "123456789");
    wifiMulti.addAP("hom-iot", "123456789");
    wifiMulti.addAP("home", "123456789");

    Serial.println("Connecting Wifi...");
    if(wifiMulti.run() == WL_CONNECTED) {
        Serial.println("");
        Serial.println("WiFi connected");
        Serial.println("IP address: ");
        Serial.println(WiFi.localIP());
    }
}

void loop()
```

The Serial Monitor window titled "COM9" shows the output:

```
WiFi connected
IP address:
192.168.4.101
```



# File > Examples > WiFi > SimpleWebServerWiFi

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

- Sketch Window:** Displays the code for `node32pico_SimpleWebServerWiFi`. The code initializes the serial port at 115200 baud, sets pin 2 as an output, and attempts to connect to a WiFi network named "lamloeiWIFI". It also prints the IP address (192.168.4.101), signal strength (RSSI), and provides instructions to view the web server in a browser.
- Serial Monitor:** Titled "node32pico\_SimpleWebServerWiFi" and connected to "COM9". It shows the serial output from the sketch, including the connection attempt and the provided instructions.
- Browser Preview:** A small window showing a web page at `192.168.4.101/`. The page contains two links:
  - "Click [here](#) turn the LED on pin 2 on"
  - "Click [here](#) turn the LED on pin 2 off"

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



ปรับ

```
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

File Edit Sketch Tools Help

node32lite\_ESP\_NBNST

```
#include <WiFi.h>
#include <NetBIOS.h>

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

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

  // Connect to WiFi network
  WiFi.mode(WIFI_STA);
  WiFi.begin(ssid, password);
  Serial.println("");

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

  NBNS.begin("ESP");
}

..
```

Connected to lamloeiWIFI  
IP address: 192.168.4.112

Autoscroll  Show timestamp

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=115ms 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=38ms 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 = 21ms, Maximum = 115ms, Average = 50ms

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=115ms 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 = 19ms, Maximum = 115ms, Average = 51ms

C:\Users\admin>
```



# File > Examples > SimpleBLE > SimpleBleDevice

The screenshot shows the Arduino IDE with the sketch `node32pico_SimpleBleDevice` open. The code implements a simple BLE device with a button that prints its name and a timestamp to the Serial port. The Serial monitor output shows repeated entries of "BLE32 name: 273".

```
SimpleBLE ble;

void onButton() {
    String out = "BLE32 name: ";
    out += String(millis() / 1000);
    Serial.println(out);
    ble.begin(out);
}

void setup() {
    Serial.begin(115200);
    Serial.setDebugOutput(true);
    pinMode(0, INPUT_PULLUP);
    Serial.print("ESP32 SDK: ");
    Serial.println(ESP.getSdkVersion());
    ble.begin("ESP32 SimpleBLE");
    Serial.println("Press the button to change name");
}
```

On the right, the BLE Scanner application is running on a mobile device. It lists nearby BLE devices:

- HC-05 (Bluetooth)
- ESP32 SimpleBLE (Bluetooth)
- DESKTOP-A75VI5T (Computer)

A note at the bottom of the screen states: "ASUS\_T00J ปราบภัยในอุปกรณ์โกลเดี้ยงเมื่อเปิดการตั้งค่าบลูทูธ"



# File > Examples > BluetoothSerial > SerialToSerialBT

node32lite\_SerialToSerialBT | Arduino 1.8.8

```
// This example code is in the Public Domain (or CC0 licensed, at your option.)
// By Evandro Copercini - 2018
//
// This example creates a bridge between Serial and Classical Bluetooth (SPP)
// and also demonstrate that SerialBT have the same functionalities of a normal Serial
// port.

#include "BluetoothSerial.h"

#if !defined(CONFIG_BT_ENABLED) || !defined(CONFIG_BLUEDROID_ENABLED)
#error Bluetooth is not enabled! Please run `make menuconfig` to and enable it
#endif

BluetoothSerial SerialBT;

void setup() {
    Serial.begin(115200);
    SerialBT.begin("ESP32test"); //Bluetooth device name
    Serial.println("The device started, now you can pair it with bluetooth!");
}

void loop() {
    if (Serial.available()) {
        SerialBT.write(Serial.read());
    }
    if (SerialBT.available()) {
        Serial.write(SerialBT.read());
    }
    delay(20);
}
```

The device started, now you can pair it with bluetooth!

-Upload เสร็จให้กดปุ่ม EN 1 ครั้ง  
โหลดโปรแกรม Android Bluetooth Serial ที่มีอยู่อ



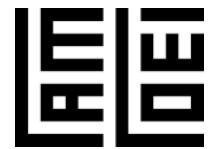
# เพิ่ม Code จากภายนอก

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

The screenshot shows a web browser window with the URL <https://playground.arduino.cc/Main/I2cScanner> in the address bar. The page content is a code editor displaying the following Arduino sketch:

```
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. }
```

In the bottom right corner of the code editor, there is a blue button labeled "[Get Code]" which is enclosed in a red rectangular box.



## 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 image shows the Arduino IDE interface with two windows. The left window is titled "node32lite\_i2c\_scanner" and contains the following code:

```
#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;
            Serial.print("I2C device found at address ");
            Serial.print(address, HEX);
            Serial.println(" !");
            nDevices++;
        }
    }
    if (nDevices == 0)
        Serial.println("No I2C devices found");
}
```

The right window is the "Serial Monitor" showing the output of the code. It displays three instances of the message "I2C device found at address 0x45 !", followed by "done".

```
I2C device found at address 0x45 !
done

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

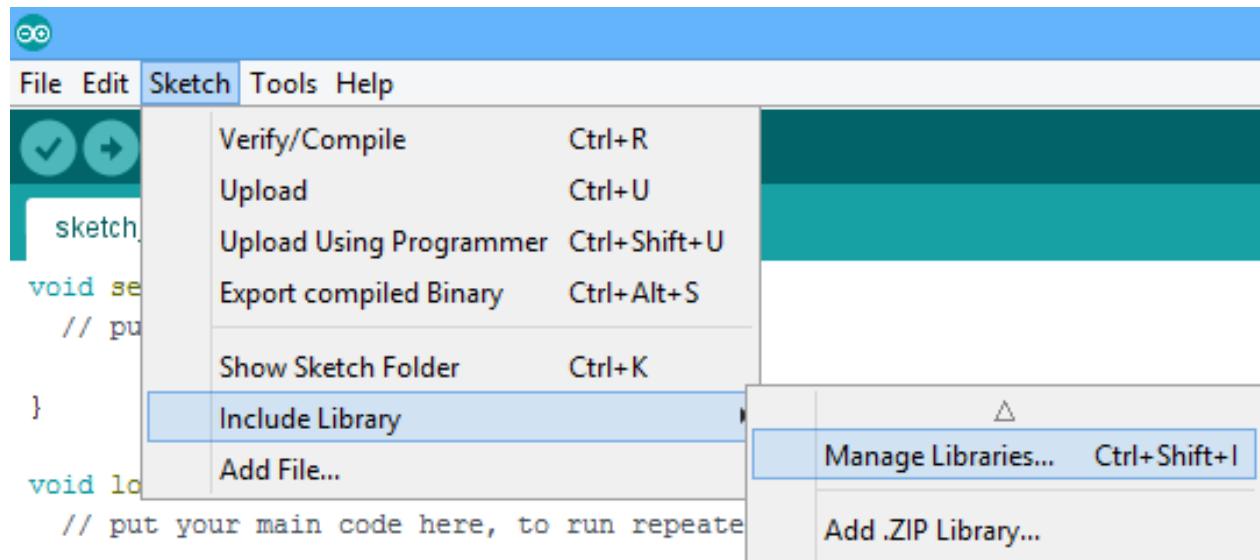
Scanning...
I2C device found at address 0x45 !
done

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



## อ่านค่า อุณหภูมิ ความชื้น เพิ่ม library

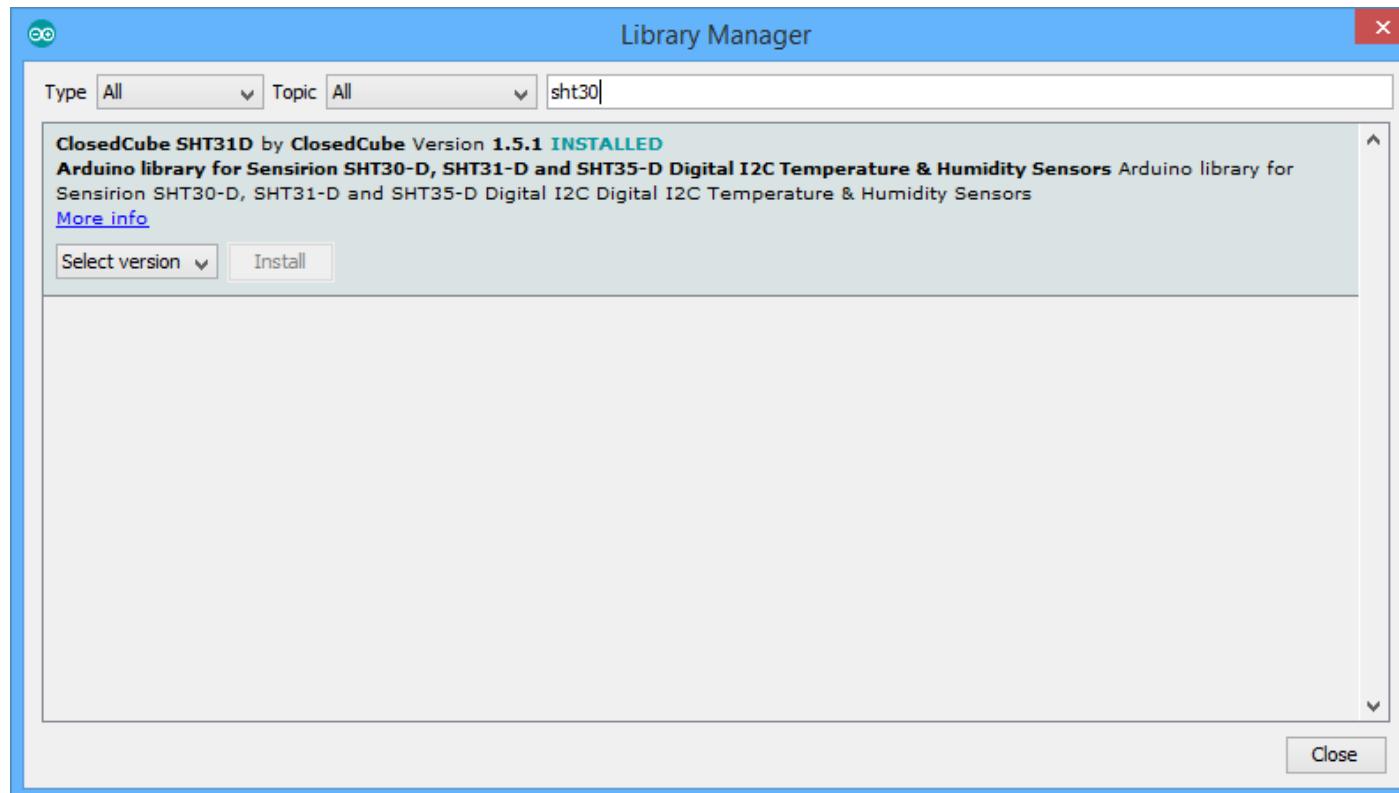
Sketch > Include Library > Manage Libraries...





พิมพ์ SHT30 ในช่อง Search

Install ClosedCube SHT31D





# File > Examples > ClosedCube SHT31D > periodicmode

The image shows the Arduino IDE interface. The top bar displays "periodicmode | Arduino 1.8.8". The left pane shows the code for the "periodicmode" sketch:

```
File Edit Sketch Tools Help
periodicmode $ periodicmode | Arduino 1.8.8
void setup()
{
    Wire.begin();

    Serial.begin(115200);
    Serial.println("ClosedCube SHT3X-D Periodic Mode Example");
    Serial.println("supports SHT30-D, SHT31-D and SHT35-D");

    sht3xd.begin(0x45); // I2C address: 0x44 or 0x45

    Serial.print("Serial #");
    Serial.println(sht3xd.readSerialNumber());

    if (sht3xd.periodicStart(SHT3XD_REPEATABILITY_HIGH, SHT3XD_FREQUENCY))
        Serial.println("[ERROR] Cannot start periodic mode");
}

void loop()
{
    printResult("Periodic Mode", sht3xd.periodicFetchData());
    delay(250);
}
```

The right pane shows the Serial Monitor window with the following output:

```
Periodic Mode: T=24.37C, RH=37.78%
Periodic Mode: T=24.38C, RH=37.75%
Periodic Mode: T=24.37C, RH=37.81%
Periodic Mode: T=24.40C, RH=37.78%
Periodic Mode: T=24.38C, RH=37.76%
Periodic Mode: T=24.38C, RH=37.76%
Periodic Mode: T=24.38C, RH=37.75%
Periodic Mode: T=24.37C, RH=37.72%
Periodic Mode: T=24.38C, RH=37.70%
Periodic Mode: T=24.37C, RH=37.70%
Periodic Mode: T=24.37C, RH=37.70%
Periodic Mode: T=24.36C, RH=37.65%
```

At the bottom of the Serial Monitor window, there are two checkboxes: "Autoscroll" (checked) and "Show timestamp" (unchecked).



## แบบฝึกหัดที่ 5

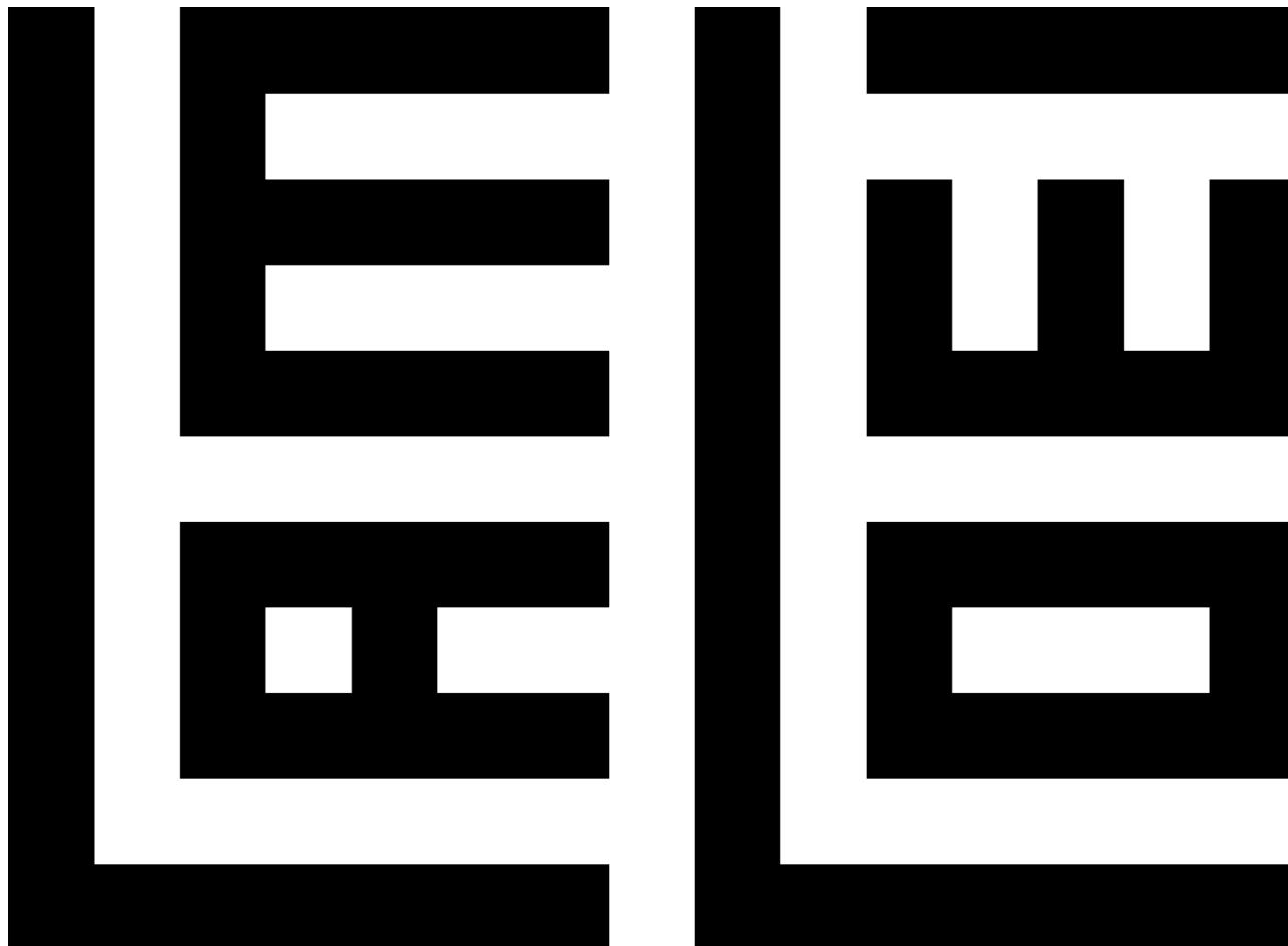
ให้เลือกตัวอย่าง File > Example แล้วอธิบาย



## แบบฝึกหัดที่ 6

สร้างระบบด้วย

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



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