

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

คอร์สอบรม

Node32Pico Plus Netpie

Arduino IDE Basic

ขั้นพื้นฐาน



จุดประสงค์

- ผู้เข้าอบรมสามารถเขียน Arduino IDE ลงบน Node32Pico
- สามารถเชื่อมต่อ netpie ได้

เอกสารคอร์สอบรมนี้ สามารถดาวน์โหลดได้ที่

- <https://github.com/lamloei/present/20180218>

*** คอร์สอบรมนี้ หมายความว่า สมกับผู้เริ่มต้น ***



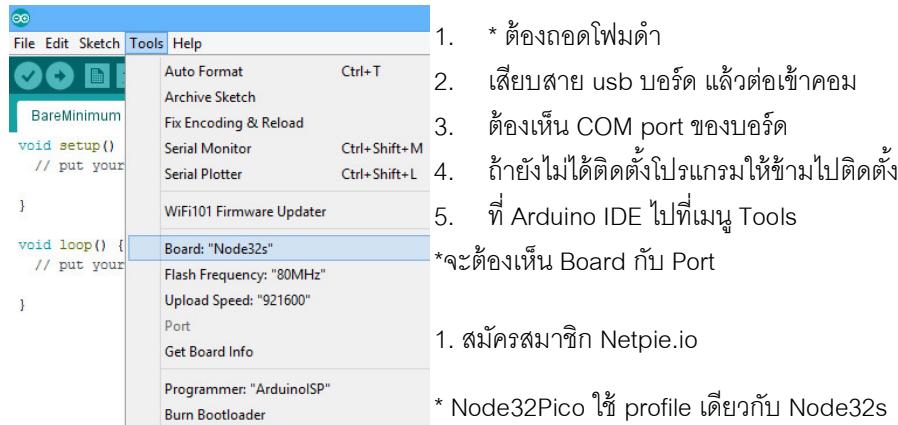
กำหนดการ

- 11 – intro & install
- 12 – basic input output
- 13 – lunch break
- 14 – adv IO touch
- 15 – sensor
- 16 – ble simple & connect wifi & temp & humi
- 17 – connect netpie

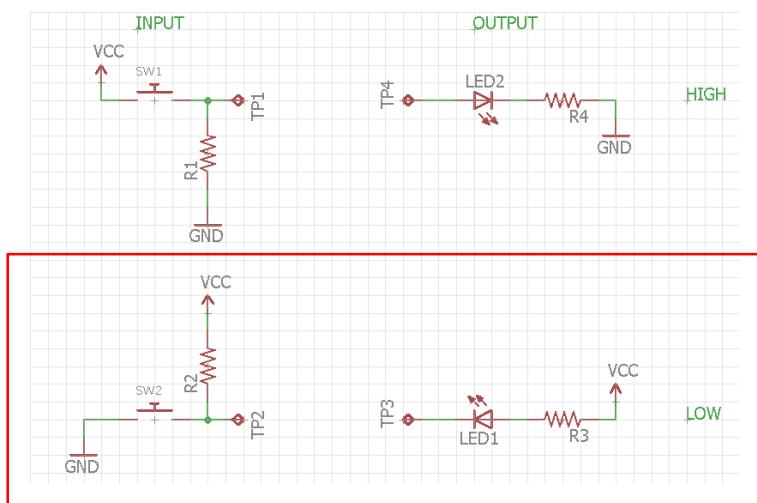


พื้นฐาน

Tools > Board: “Node32s”



Active High Active Low



Active Low จะใช้แหล่งจ่ายภายนอก ทำให้มีเป็นภาระของ MCU



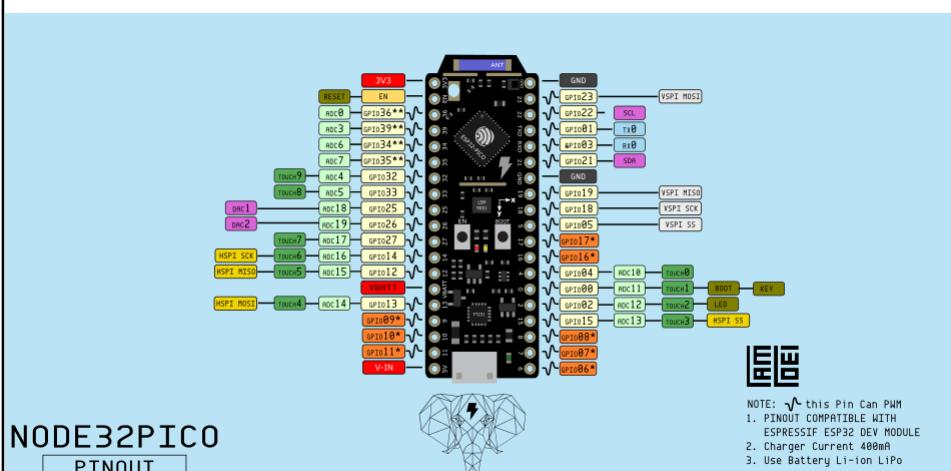
Node32Pico

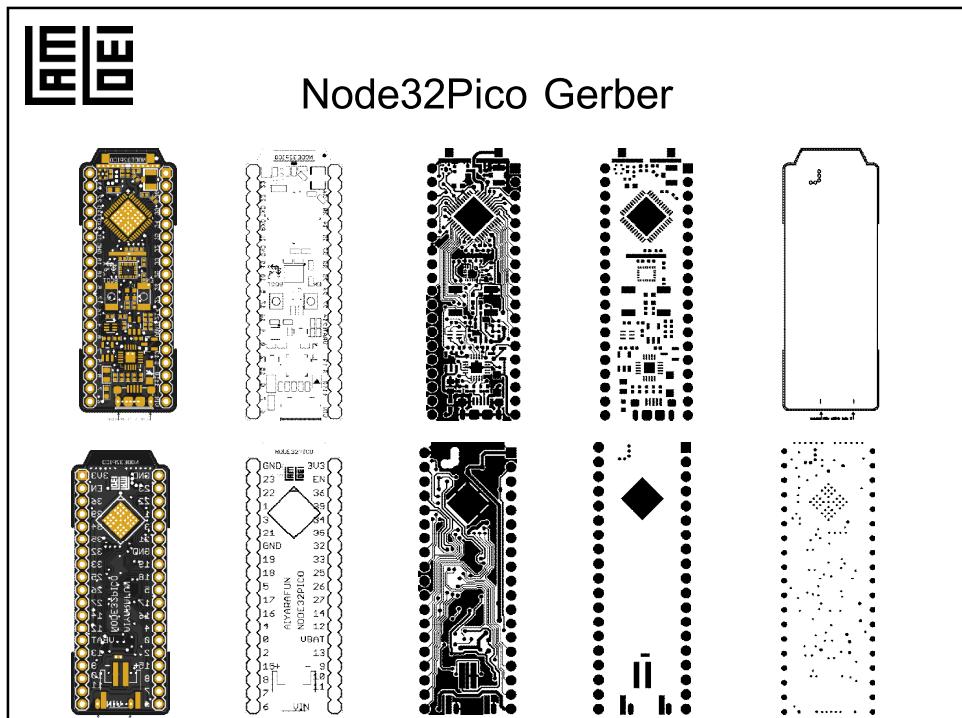
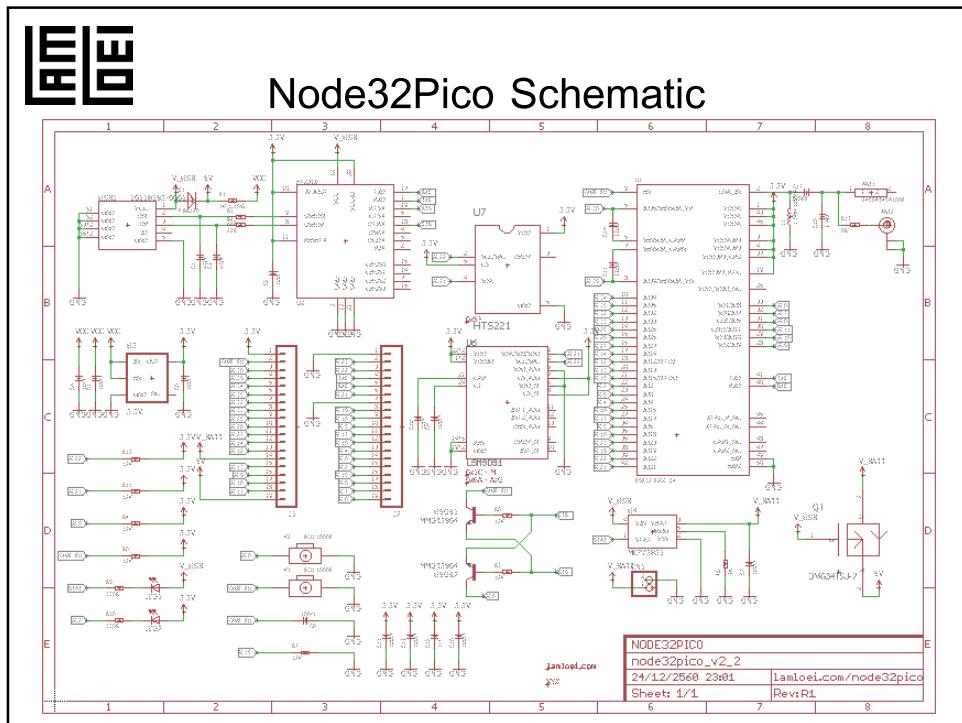


https://www.gravitechthai.com/product_detail.php?d=2273



Node32Pico Pinout







Node32Pico BOM

Node32Pico				v2.2	24/12/2017		
Item	Quantity	Reference	Part	Footprint	Mfg	Mfg P/N	Vendor
1	1	ANT1	ANTENNA CHIP UWB 2.3 - 2.7 GHZ		Johanson	2500AT44M0400E	Mouser
2	1	ANT2	U.FL				
3	2	C1, C2	CAP CER 47PF 50V COG 2%	0402	Murata	GRM1555C1H470GA01D	Mouser
4	9	C3, C5, C6, C7, C8, C9, C11, C12, C16	CAP CER 0.1UF 16V X7R	0402	Murata	GRM155R71C104KA88D	Mouser



Node32Pico Testrun

```

COM29

11: IUVINTEL (*~72)
12: NSTDA-WIFI (-72)
13: eduroam (-73)
14: NSTDA-GUEST (-74)
15: NSTDA-WIFI (-88)

***CHIP ID***
ESP32 Chip ID = 940E001DA0DE
Chip Revision (official version): 0

***I2C SCANNING***
I2C device found at address 0x1C !
I2C device found at address 0x5F !
I2C device found at address 0x6A !
done

***HALL SENSOR***
sensor = 9

***TIME TEMPERATURE SENSOR***
scan start: 00:00:00 Temp onBoard 167°F 75.00°C

***HTS221 SENSOR***
Humidity : 20.00 %
Temperature: 38.60 celsius

***LISMD51 SENSOR***
G: -0.10, -2.93, -0.39 deg/s
A: -0.03, 0.03, 0.98 g
M: 0.61, -0.74, -0.82 gauss
Pitch, Roll: 1.54, 1.55
Heading: 137.85

***WIFI SCAN***
scan done

```

Autoscroll No line ending 115200 baud Clear output



วิธีติดตั้ง

- Install Arduino IDE
- Install Git SCM
- Git GUI, source, target > clone
- Double click get.exe



ดาวน์โหลดและติดตั้งไฟล์ที่

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

HOME BUY SOFTWARE PRODUCTS LEARNING COMMUNITY SUPPORT

Download the Arduino IDE



ARDUINO 1.8.5

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](#) Get

[Mac OS X](#) 10.7 Lion or newer

[Linux 32 bits](#)
[Linux 64 bits](#)
[Linux ARM](#)

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



- Git Gui
- Select Clone Existing Repository
- Source: <https://github.com/espressif/arduino-esp32.git>
- Target: C:/Users/[YOUR_USER_NAME]/Documents/
Arduino/hardware/espressif/esp32
- Click Clone
- Open C:/Users/[YOUR_USER_NAME]/Documents/
Arduino/hardware/espressif/esp32/tools and double-click get.exe

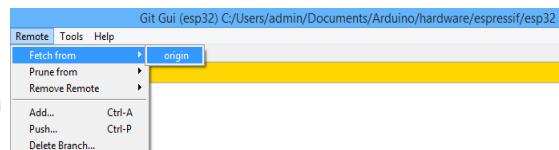


Updated

```
C:\Users\admin>cd C:\Users\admin\Documents\Arduino\hardware\espressif\esp32
C:\Users\admin\Documents\Arduino\hardware\espressif\esp32>git pull origin master
From https://github.com/espressif/arduino-esp32
 * branch            master       -> FETCH_HEAD
Already up-to-date.

C:\Users\admin\Documents\Arduino\hardware\espressif\esp32>
```

- Git CMD
- cd C:\Users\[YOUR_USER_NAME]\Documents\Arduino\hardware\espressif\esp32
- git pull origin master
- หรือ Fetch from > origin

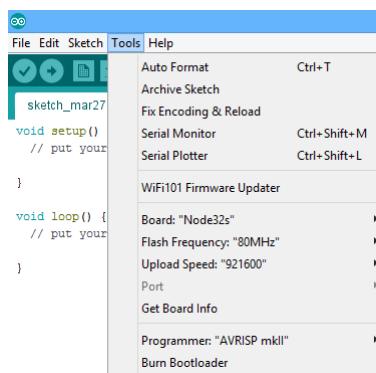


ไฟล์เดอร์

- งานเอกสาร Arduino
C:\Users\[YOUR_USER_NAME]\Documents\Arduino
- ไฟล์เดอร์ Arduino IDE
C:\Program Files (x86)\Arduino
C:\Users\[YOUR_USER_NAME]\AppData\Local\Arduino15



Tools > Board: “Node32s”

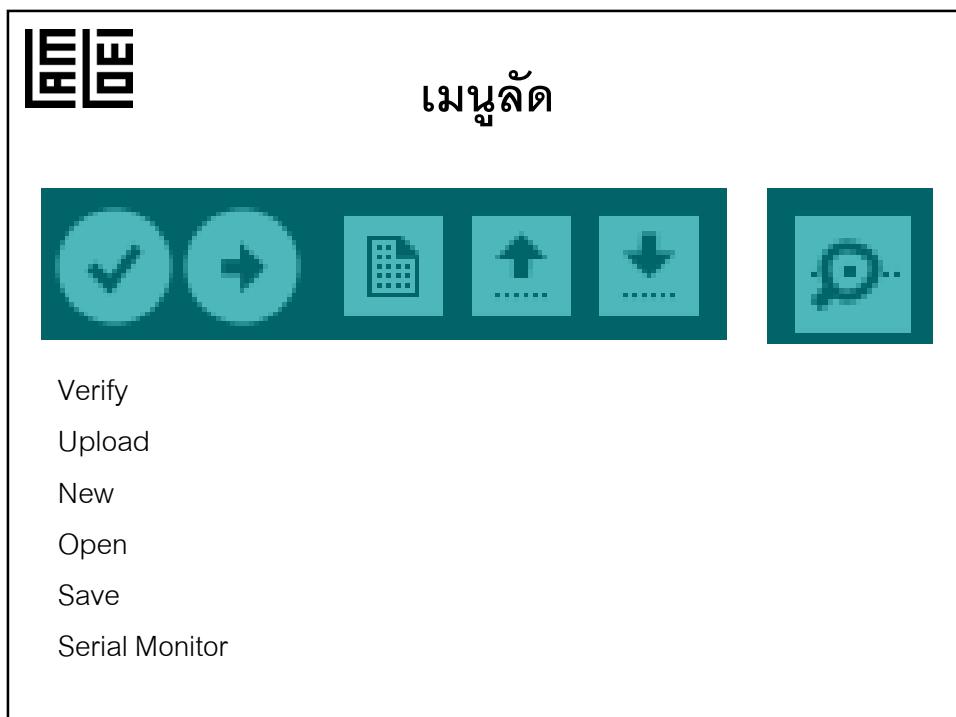
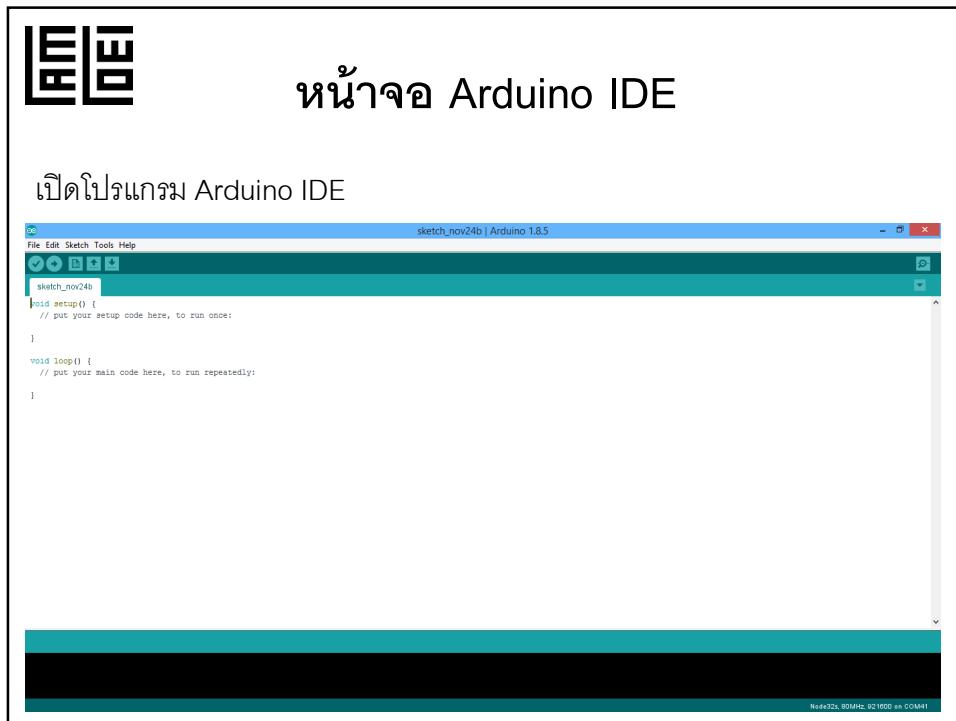


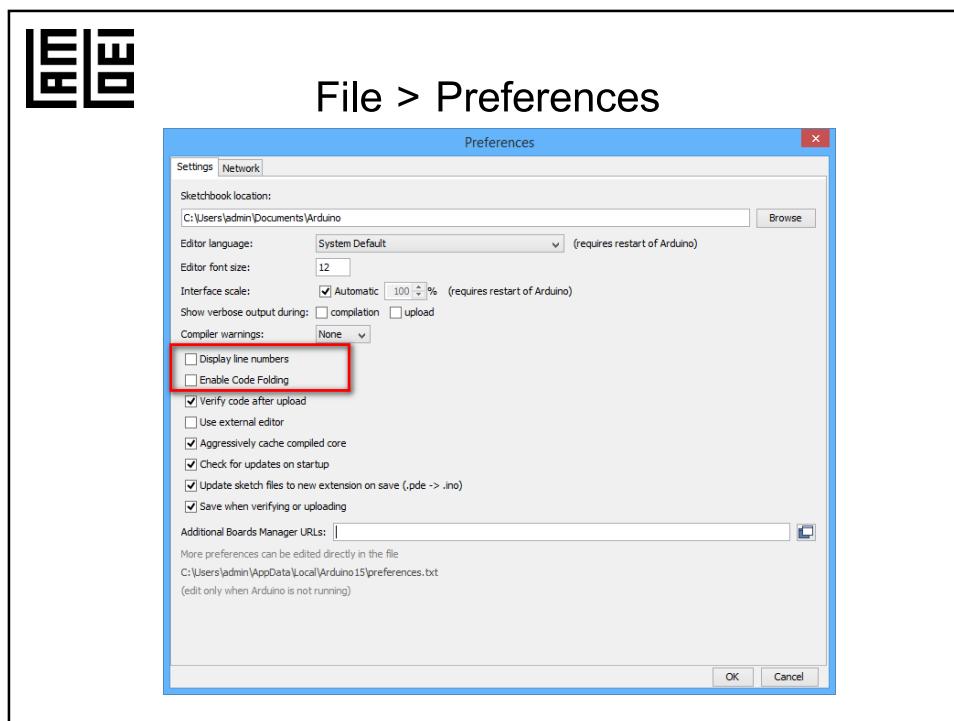
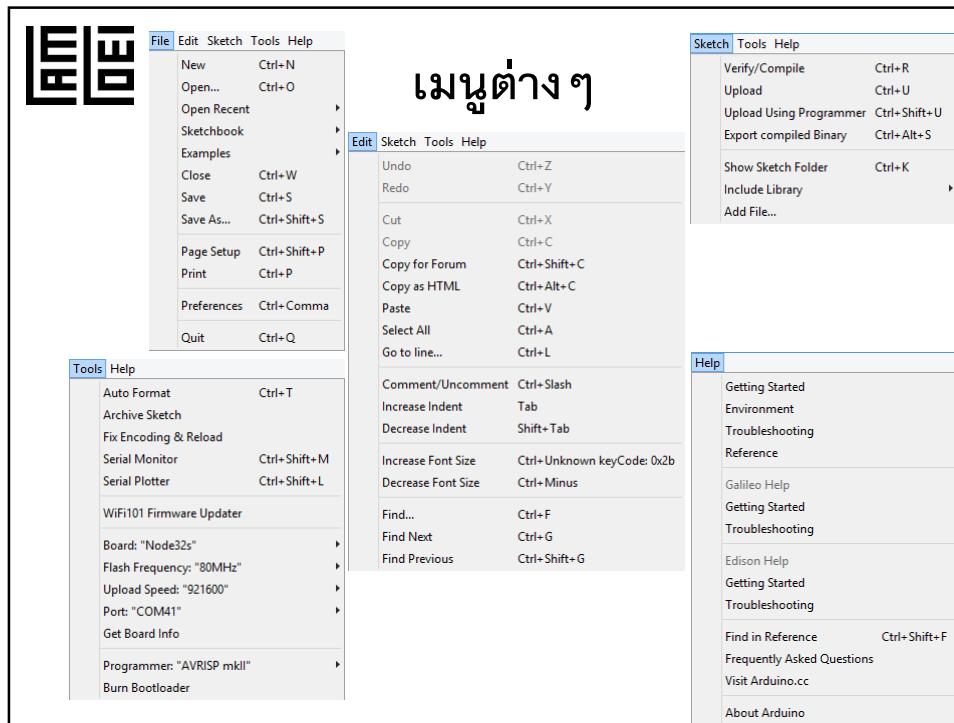
ຈະຕີອັນເຫັນທັງ Board ແລະ Port



Break

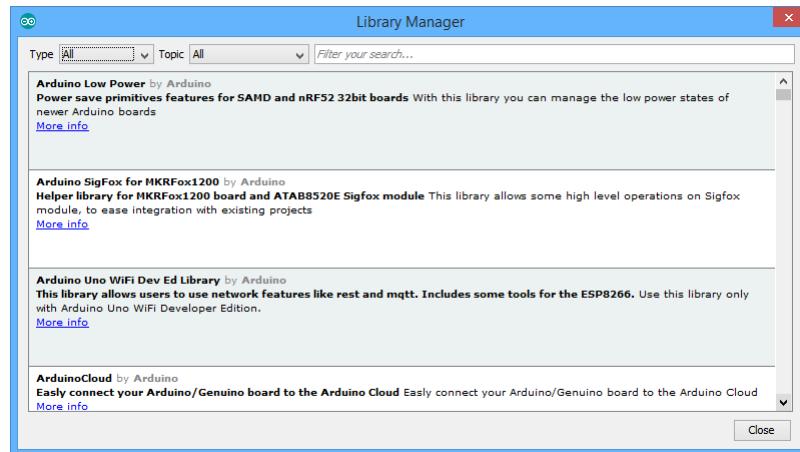
- ພັກ 15 ນາທີ
- ແນະນຳດັວ
- ຫົ້ອ
- ຄວາມຊອບ
- ຄວາມຄາດຫວັງຂອງຄອຮົສນີ້
- ຄອຮົສທີ່ອຍາກຈະໃໝ່ເປີດໃນອນາຄຕ



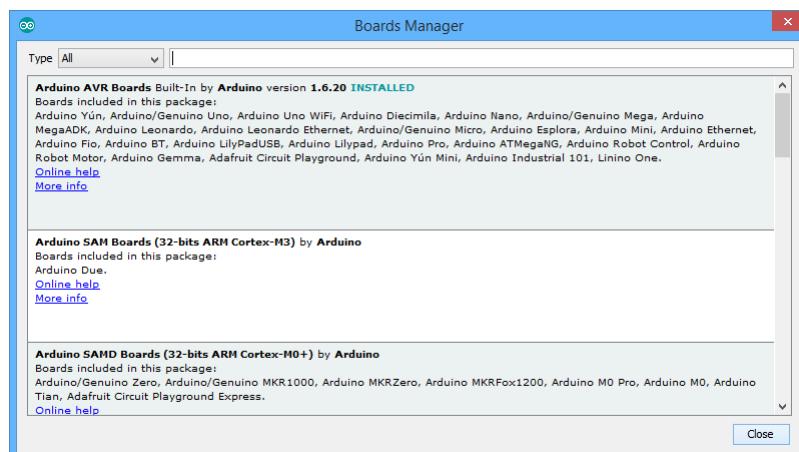




Sketch > Include Library > Manage Libraries...

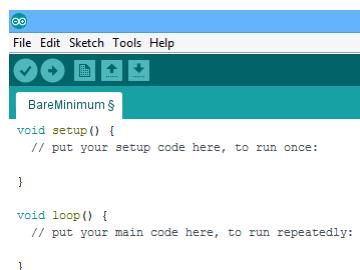
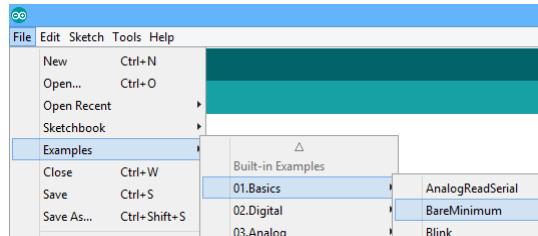


Tools > Board: > Board Manager...





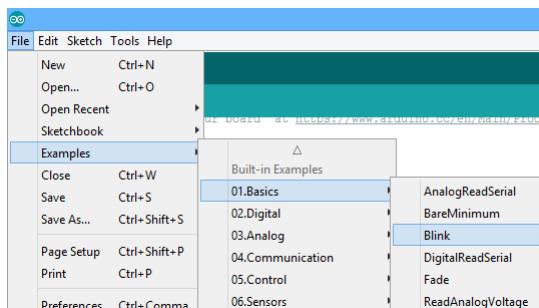
File > Examples > 01.Basics > BareMinimum



void setup() {} - ทำงานครั้งเดียว
void loop() {} - ทำงานวนลูป



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
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 SCK = 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 A1  = 39;
32 static const uint8_t A4  = 32;
33 static const uint8_t A5  = 30;
34 static const uint8_t A6  = 31;
35 static const uint8_t A7  = 35;
36 static const uint8_t A10 = 4;
37 static const uint8_t A11 = 0;

```



waiting for download

```

COM9
|
ets Jun  8 2016 00:22:57

rst:0x1 (POWERON_RESET),boot:0x1 (DOWNLOAD_BOOT(UART0/UART1/SDIO_FEI_REO_V2))
waiting for download

```

กดปุ่ม EN ค้างไว้
 กดปุ่ม BOOT ค้างไว้
 ปล่อยปุ่ม EN
 ปล่อยปุ่ม BOOT
 ...ขึ้นข้อความ waiting for download
 กดแล้วปล่อยปุ่ม EN

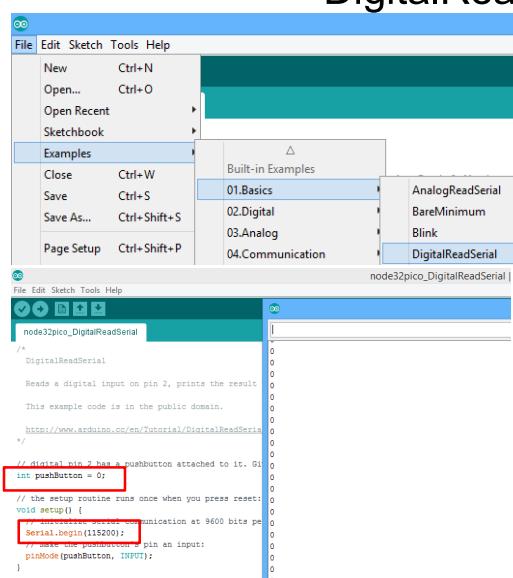


Break

- Lunch 60 minute



File > Examples > 01.Basics > DigitalReadSerial



```

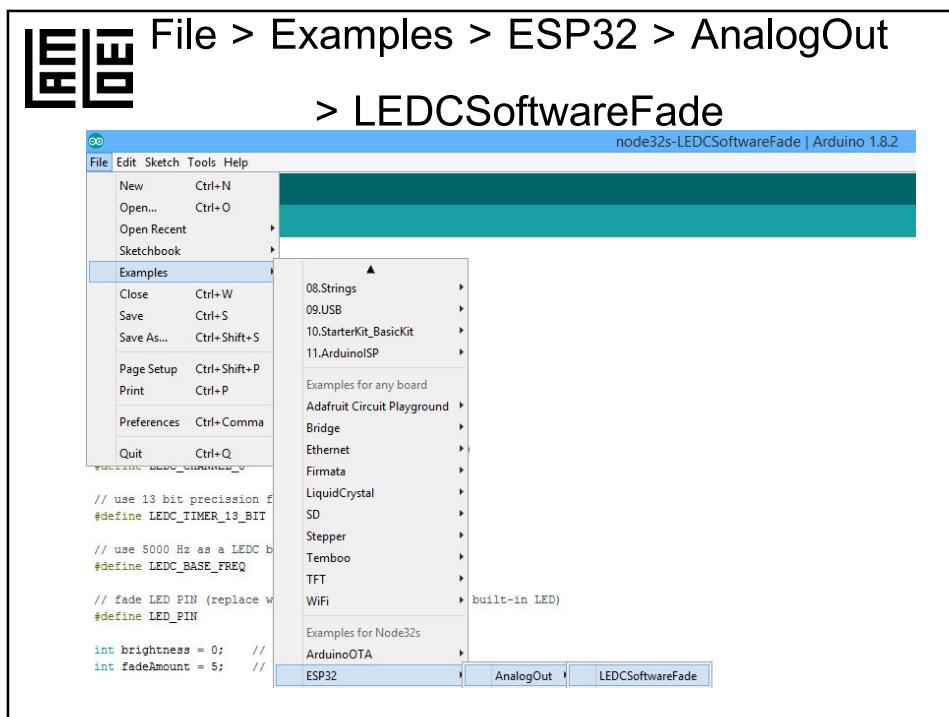
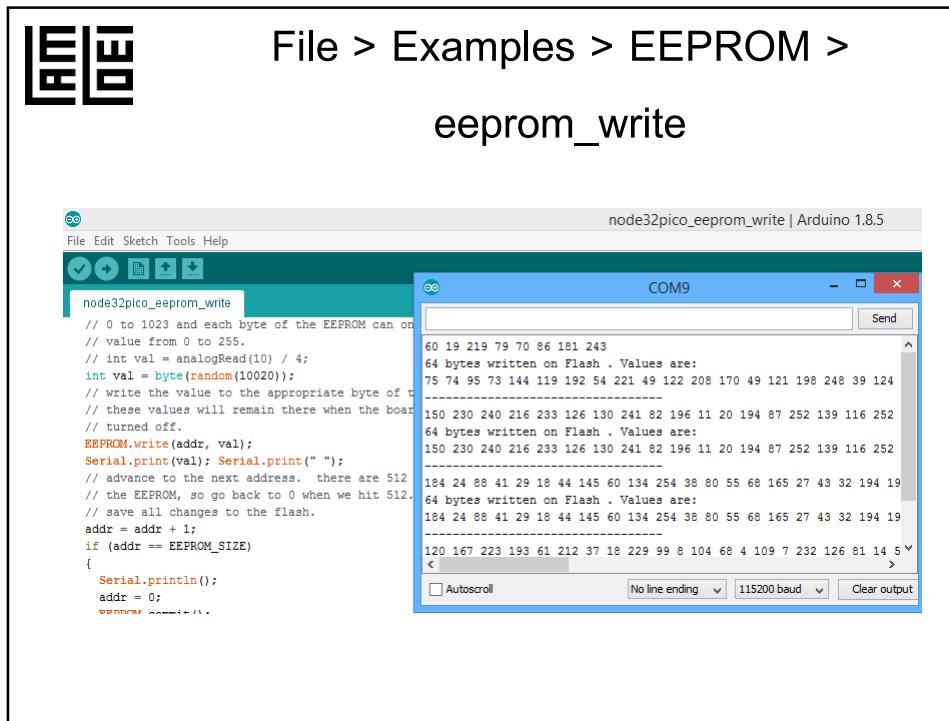
int pushButton = 0;
Serial.begin(115200);

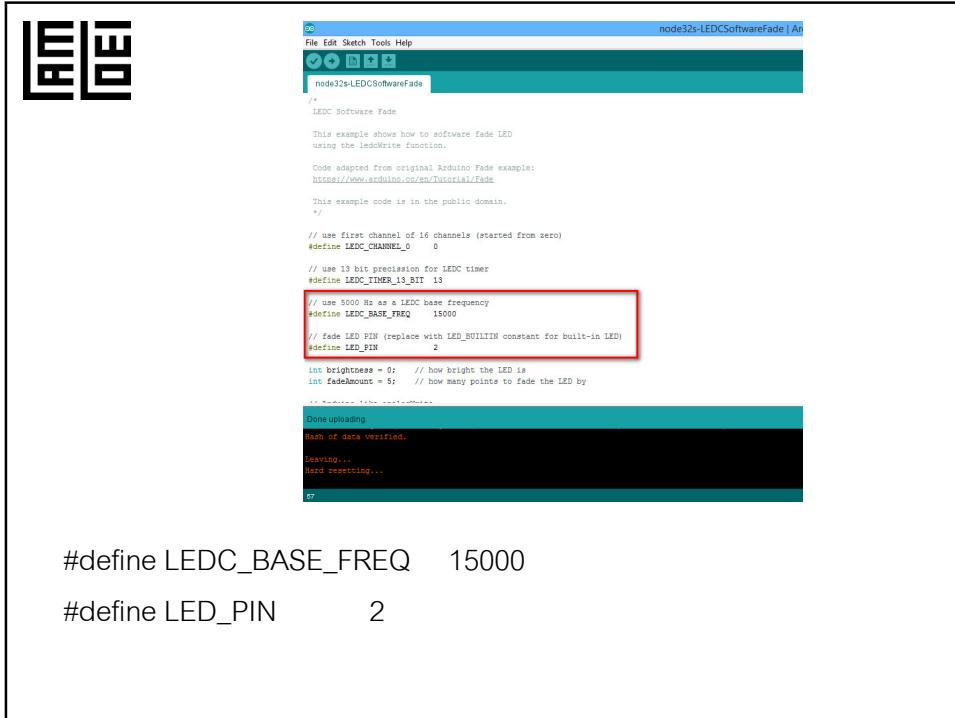
/*
DigitalReadSerial
Reads a digital input on pin 2, prints the result
This example code is in the public domain.
http://www.arduino.cc/en/Tutorial/DigitalReadSerial
*/
// digital pin 2 has a pushbutton attached to it. GND
int pushButton = 0;

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

```

กดปุ่ม BOOT
ดูความเปลี่ยนแปลง



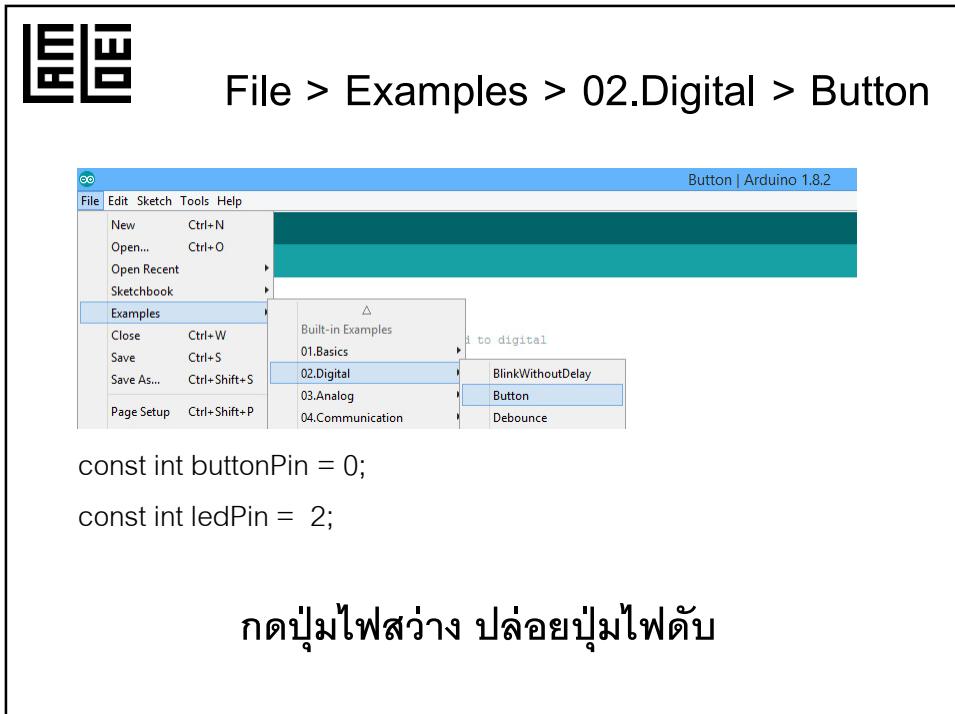


```

File Edit Sketch Tools Help
node32s-LEDCSoftwareFade | Ar
File Edit Sketch Tools Help
node32s-LEDCSoftwareFade
LEDC 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.
*/
// use first channel of 16 channels (started from zero)
#define LEDC_CHANNEL_0    0
// use 12 bit precision for LEDC timer
#define LEDC_TIMER_13_BIT 13
// use 5000 Hz as a LEDC base frequency
#define LEDC_BASE_FREQ    15000
// fade LED PIN (replace with LED_BUILTIN constant for built-in LED)
#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...
Board resetting...
57

```

#define LEDC_BASE_FREQ 15000
#define LED_PIN 2



```

File Edit Sketch Tools Help
File Examples > 02.Digital > Button | Arduino 1.8.2
File Edit Sketch Tools Help
File Examples > 02.Digital > Button
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
Built-in Examples △
01.Basics
02.Digital
03.Analog
04.Communication
BlinkWithoutDelay
Button
Debounce

```

```

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

```

กดปุ่มไฟสว่าง ปล่อยปุ่มไฟดับ



แบบฝึกหัด (15นาที)

- ให้ กดปุ่มไฟดับ ปล่อยปุ่มไฟสว่าง



Serial Monitor

The screenshot shows the Arduino IDE Serial Monitor window. The title bar says "node32pico_Serial | Ar". The main area displays the following text:

```

6: Hello Node32Pico
7: Hello Node32Pico
8: Hello Node32Pico
9: Hello Node32Pico
10: Hello Node32Pico
11: Hello Node32Pico
12: Hello Node32Pico
13: Hello Node32Pico
14: Hello Node32Pico
15: Hello Node32Pico
16: Hello Node32Pico
17: Hello Node32Pico
18: Hello Node32Pico
19: Hello Node32Pico
20: Hello Node32Pico

```

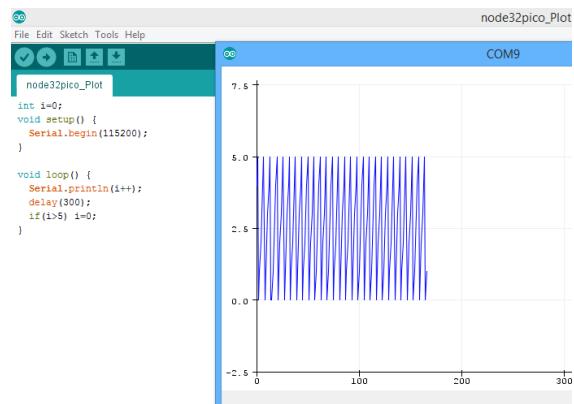
พิมพ์ค่าสั่ง

ปรับค่า Serial 115200 ให้ตรงกัน

<input checked="" type="checkbox"/> Autoscroll	No line ending	115200 baud	Clear output
--	----------------	-------------	--------------



Tools > Serial Plotter

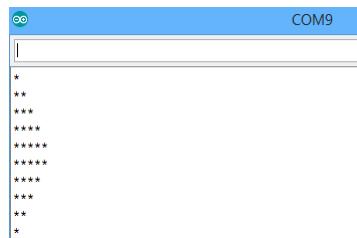


แบบฝึกหัด (15นาที)





แบบฝึกหัด (15นาที)



สรุปขั้นตอน

- File > Examples
- ปรับค่า
- Upload Program
- สังเกตความเปลี่ยนแปลง



แบบฝึกหัด (15นาที)

- เลือก File > Example
- ปรับค่า และสังเกตความเปลี่ยนแปลง
- จงอธิบายตัวอย่างนั้น



File > Examples > ESP32 > ChipID >
GetChipID

```

node32pico_GetChipID
File Edit Sketch Tools Help
node32pico_GetChipID
COM9
ESP32 Chip ID = A417001DA0D8
ESP32 Chip ID = A417001DA0D8
ESP32 Chip ID = A417001DA0D8

uint64_t chipid;
void setup() {
  Serial.begin(115200);
}

void loop() {
  chipid=ESP.getEfuseMac(); //The chip ID is essential
  Serial.printf("ESP32 Chip ID = %04X", (uint16_t)(chipid));
  Serial.printf("%08X\n", (uint32_t)chipid); //print L

  delay(3000);
}

```



File > Examples > ESP32 > DeepSleep

> TimerWakeUp

```

File Edit Sketch Tools Help
File > Examples > ESP32 > DeepSleep
> TimerWakeUp
node32pico_TimerWakeUp

/*
Simple Deep Sleep with Timer Wake Up
=====
ESP32 offers a deep sleep mode for effective power
saving as power is an important factor for IoT
applications. In this mode CPUs, most of the RAM,
and all the digital peripherals which are clocked
from APB_CLK are powered off. The only parts of
the chip which can still be powered on are:
RTC controller, RTC peripherals ,and RTC memories

This code displays the most basic deep sleep with
a timer to wake it up and how to store data in
RTC memory to use it over reboots

This code is under Public Domain License.

Author:
Pranav Cherukuballu <cherukuballu@gmail.com>

```

node32pico_TimerWakeUp

Boot number: 1
Wakeup was not caused by deep sleep
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



File > Examples > ESP32 > Touch >

TouchRead

```

File Edit Sketch Tools Help
File > Examples > ESP32 > Touch >
> TouchRead
node32pico_TouchRead

// ESP32 Touch Test
// Just test touch pin - Touch0 is T0 which is on GPIO4

void setup()
{
    Serial.begin(115200);
    delay(1000); // give me time to bring up serial monitor
    Serial.println("ESP32 Touch Test");
}

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

```

node32pico_TouchRead

ESP32 Touch Test
79
89
89
88
89
55
79
10
79
15
1

- ต่อสายไฟยาวประมาณ 10cm ที่ขา GPIO4
- ใช้มือจับที่สายไฟ จะเห็นว่าค่าที่ได้จะน้อยลง

File > Example > ESP32 >

HallSensor

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, and Help. The title bar says "node32pico_HallSensor". The code editor contains the following sketch:

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

int val = 0;
void setup() {
  Serial.begin(115200);
}

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
}
```

The Serial Monitor window is titled "node32pico_HallSensor" and "COM9". It displays the following output:

```
11
14
12
11
10
10
13
12
11
12
11
11
10
13
10
10
13
11
```

Below the IDE, there is Thai text explaining the Hall Sensor's behavior:

ปรับ Serial 115200
 ♦ N เป็น ลบ (ค่าน้อยลง)
 ♦ S เป็น บวก (ค่าเพิ่มขึ้น)

Temperature

The screenshot shows the Arduino IDE interface. The top menu bar includes File, Edit, Sketch, Tools, Help. The title bar says "node32pico_Temperature". The code editor contains the following sketch:

```
#ifdef __cplusplus
extern "C" {
#endif
uint8_t temperature_sens_read();
#ifdef __cplusplus
}
#endif
uint8_t temperature_sens_read();

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

void loop() {
  Serial.print("Temperature: ");

  // Convert raw temperature in F to Celsius degrees
  Serial.print((temperature_sens_read() - 32) / 1.8);
  Serial.println(" C");
  delay(5000);
}
```

The Serial Monitor window is titled "node32pico_Temperature" and "COM9". It displays the following output:

```
Temperature: 46.11 C
Temperature: 46.11 C
Temperature: 46.67 C
Temperature: 46.67 C
Temperature: 46.67 C
Temperature: 46.67 C
```

At the bottom of the Serial Monitor window, there is a checked checkbox labeled "Autoscroll".



Break

- 15 นาที



เพิ่ม Code จากภายนอก

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

คลิก Get Code

`Serial.begin(115200);`

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

Search

HOME BUY SOFTWARE PRODUCTS LEARNING COMMUNITY SUPPORT

```

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]



Node32pico_I2cScanner

The screenshot shows the Arduino IDE interface. The sketch window contains the following code:

```

// Version 6, November 27, 2015.
// Added waiting for the Leonardo serial communication.
//
//
// This sketch tests the standard 7-bit addresses
// Devices with higher bit address might not be seen properly.
//

#include <Wire.h>

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

void loop()
{
}

```

The serial monitor window shows the output of the I2C scanner:

```

I2C Scanner
Scanning...
I2C device found at address 0x1C !
I2C device found at address 0x5F !
I2C device found at address 0x6A !
done

```



เพิ่ม library HTS221 จากภายนอก

- เปิดเบราว์เซอร์ไปที่ <http://lambda-board.com/main/lambda-plus/>
- คลิก Humidity & Temperature เพื่อดาวน์โหลดไฟล์
- เดก้าไฟล์ไปที่ Arduino/libraries
- ปิดแล้วเปิด Arduino IDE

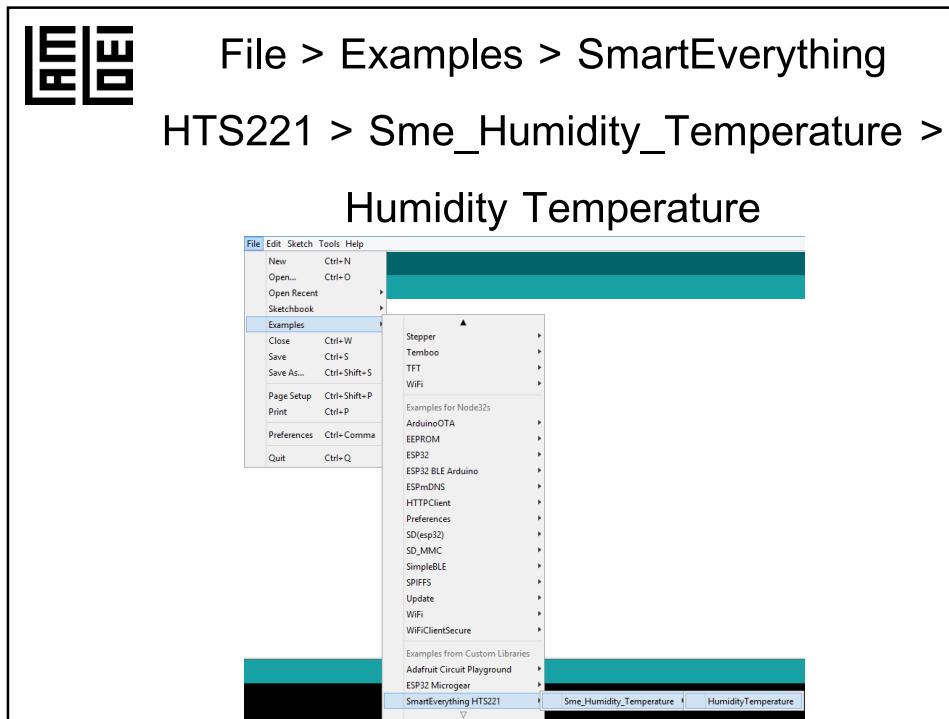


Libraries

- 9-DOF
- Barometer
- Humidity & Temperature

The screenshot shows the Windows File Explorer displaying the contents of the 'libraries' folder within the Arduino directory. The folder contains the following files and folders:

Name	Date modified	Type	Size
Adafruit_Circuit_Playground	22/11/2560 14:26	File folder	
ESP32_Micogear-master	24/11/2560 0:29	File folder	
Firmata	22/11/2560 14:26	File folder	
sme-hts221-library-master	22/11/2560 18:30	File folder	
readme.txt	22/11/2560 13:41	TXT File	1 KB



Node32pico hts221 Humidity Temperature

```

node32pico_hts221_HumidityTemperature
File Edit Sketch Tools Help
node32pico_hts221_HumidityTemperature
void setup() {
    //Initiate the Wire library and join the
    Wire.begin();
    pinMode(2, OUTPUT);
    smeHumidity.begin();
    Serial.begin(115200);
}

// the loop function runs over and over again
void loop() {
    double data = 0;

    data = smeHumidity.readHumidity();
    Serial.print("Humidity : ");
    Serial.print(data);
    Serial.println("%");

    data = smeHumidity.readTemperature();
    Serial.print("Temperature: ");
    Serial.print(data);
    Serial.println(" celsius");

    digitalWrite(2, HIGH); // turn the LED on
    delay(500); // wait for a sec
    digitalWrite(2, LOW); // turn the LED off
    delay(500); // wait for a sec
}

```

Serial Monitor Output:

```

Humidity : 86.00 %
Temperature: 24.88 celsius
Humidity : 72.00 %
Temperature: 34.69 celsius
Humidity : 72.00 %
Temperature: 34.74 celsius
Humidity : 72.00 %
Temperature: 34.74 celsius
Humidity : 72.00 %
Temperature: 34.71 celsius
Humidity : 72.00 %
Temperature: 34.73 celsius
Humidity : 72.00 %
Temperature: 34.71 celsius
Humidity : 72.00 %
Temperature: 34.69 celsius

```

Output from the serial monitor:

```

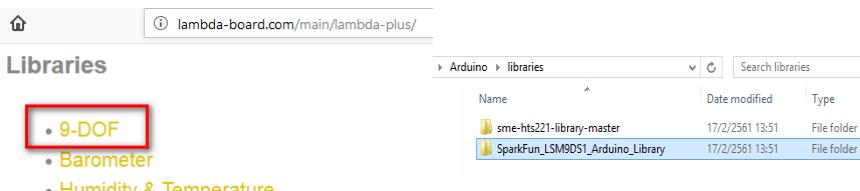
pinMode(2, OUTPUT);
Serial.begin(115200);
digitalWrite(2, HIGH);
digitalWrite(2, LOW);

```



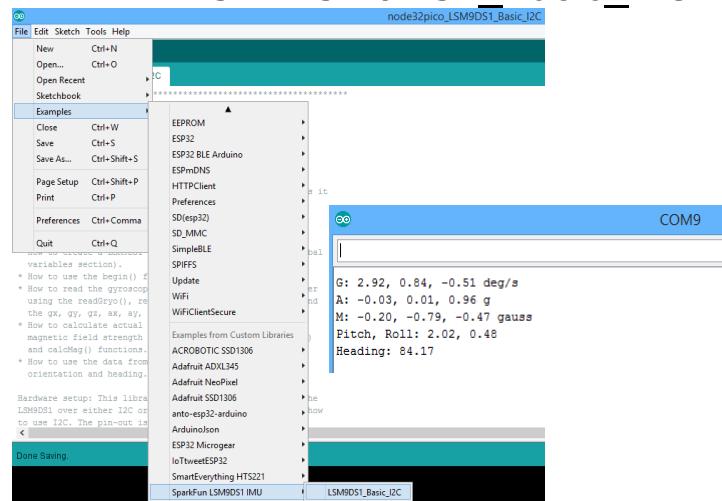
เพิ่ม library LSM9DS1 จากภายนอก

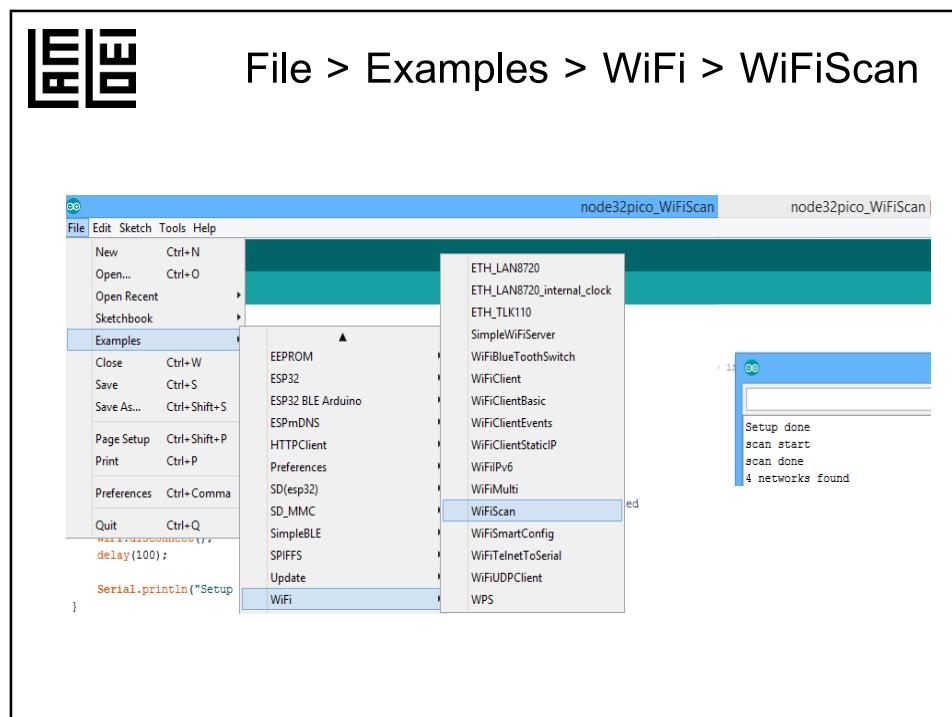
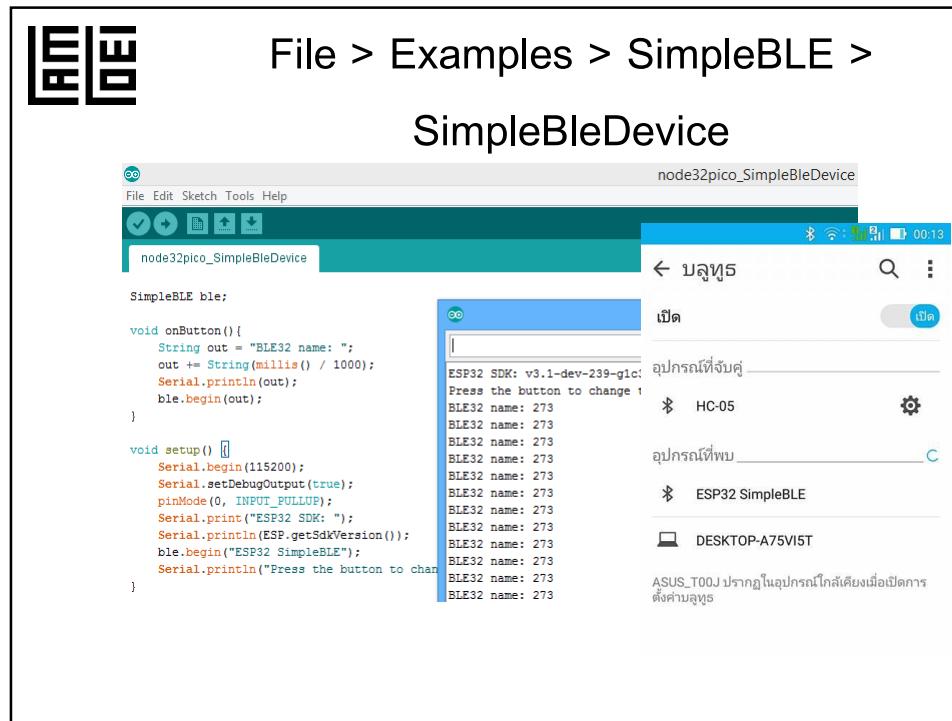
1. เปิดเบราว์เซอร์ไปที่ <http://lambda-board.com/main/lambda-plus/>
2. คลิก 9-DOF เพื่อดาวน์โหลดไฟล์
3. แตกไฟล์ไปที่ Arduino/libraries
4. ปิดแล้วเปิด Arduino IDE

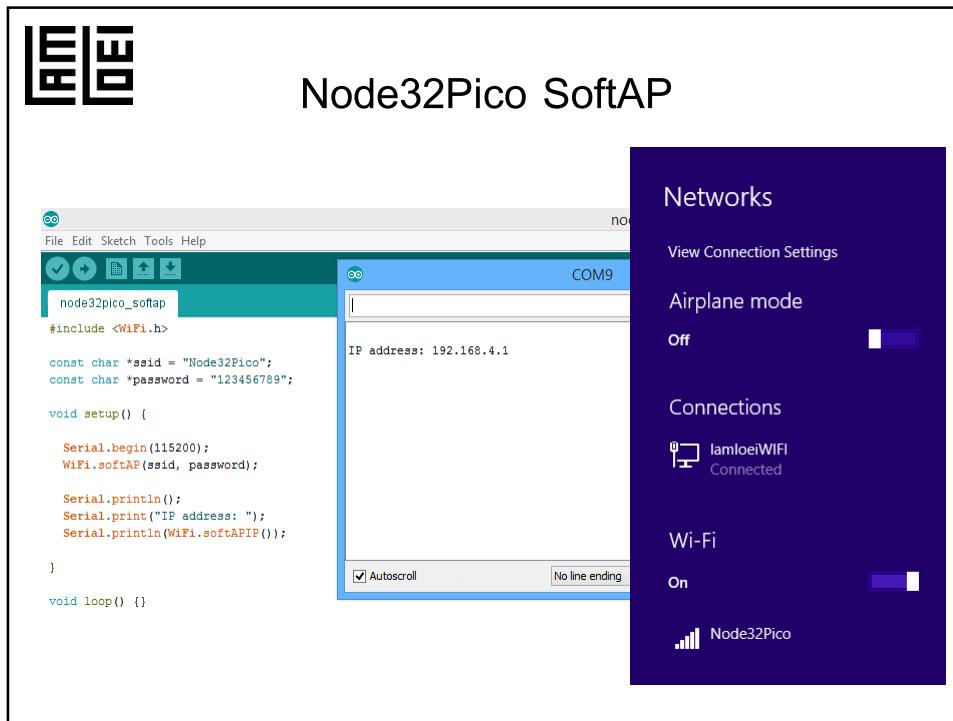


File > Examples > Sparkfun LSM9DS1

IMU > LSM9DS1_Basic_I2C









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



عنوان

```

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  
}
```



Break

- 15 ນາທີ



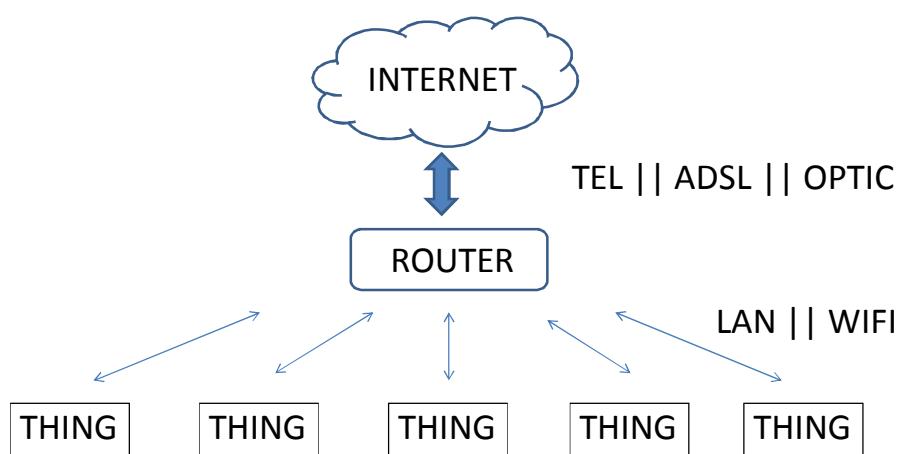
สมัคร Netpie

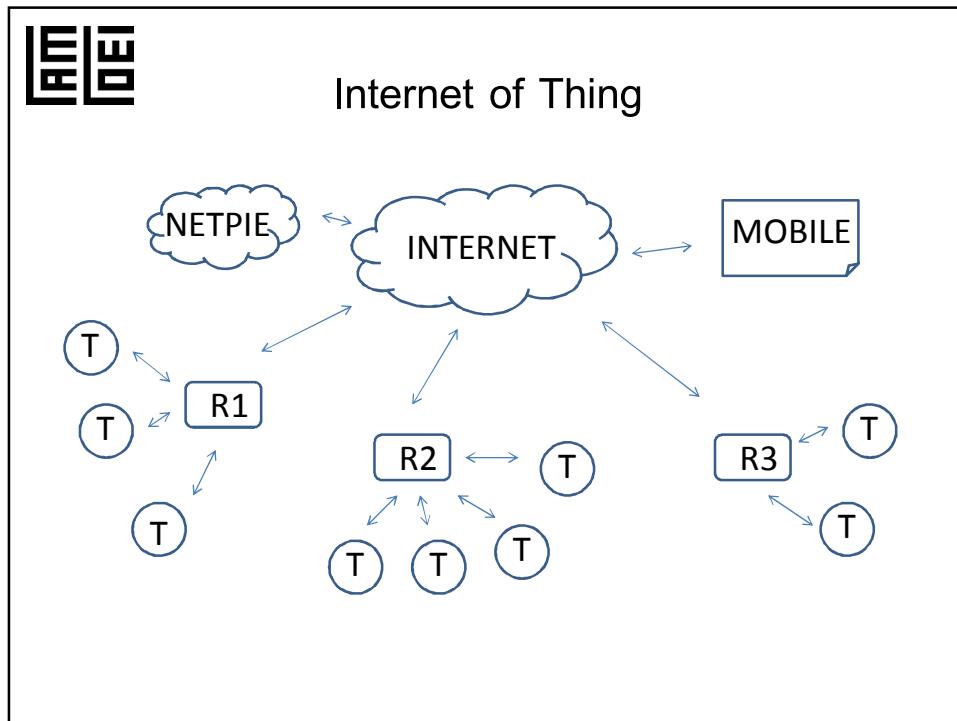
จะได้

1. User
2. Password - สำหรับเข้าหน้าเว็บ
3. Appid
4. Appkey
5. Appsecret - สำหรับให้ Thing เข้าใช้งาน



Internet of Thing

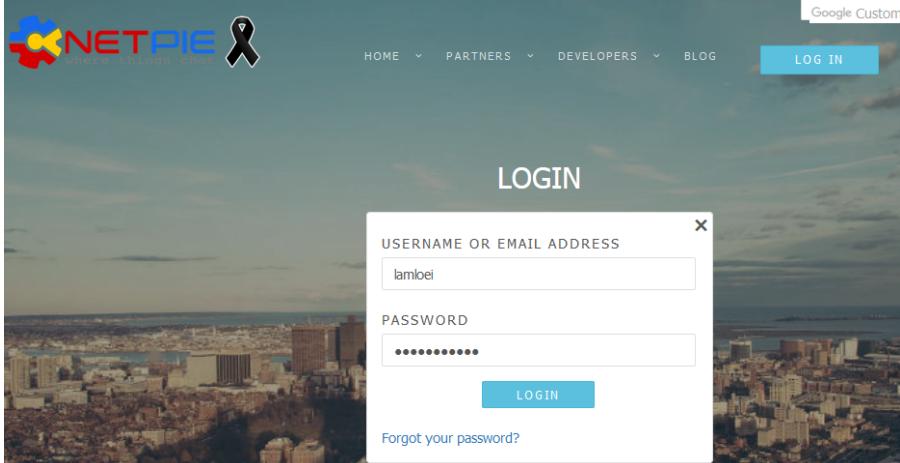




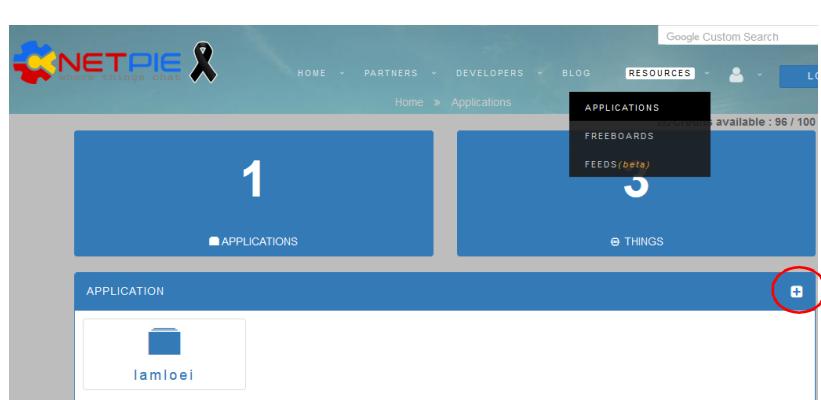
สมัคร Netpie

<https://netpie.io/>

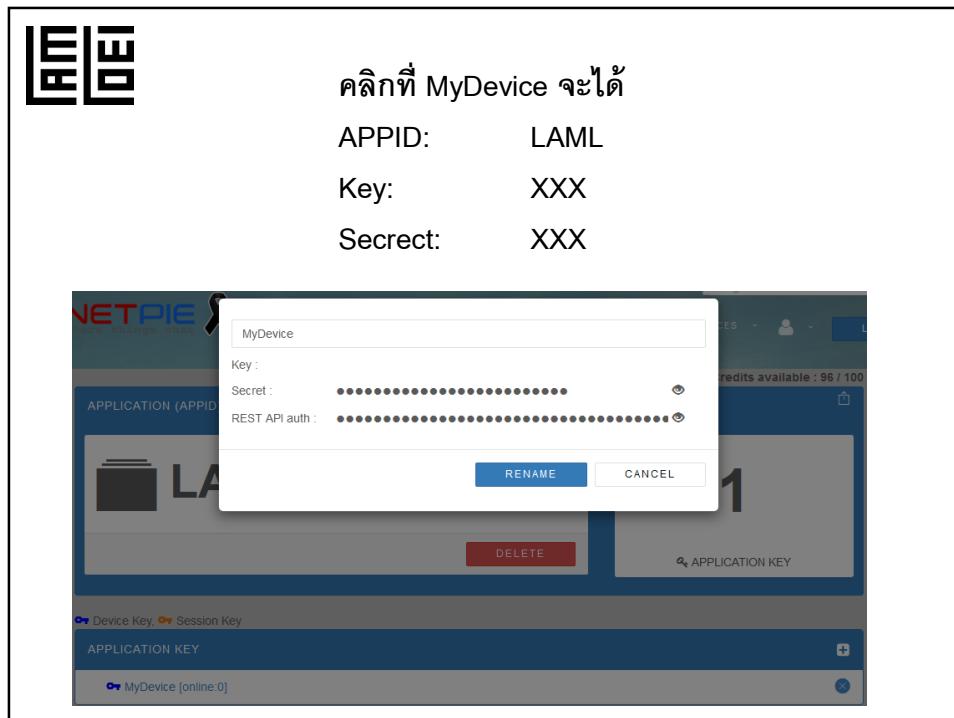
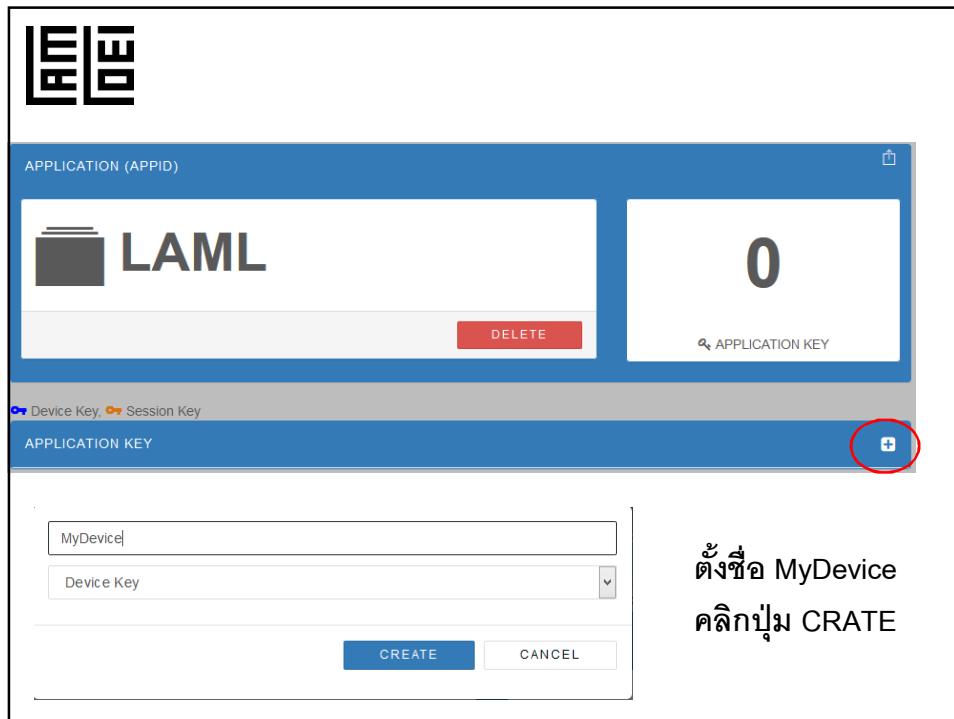
SIGN UP FREE ແລ້ວທຳຕາມຂໍ້ເຕອນ



The screenshot shows the NETPIE login page. At the top left is the NETPIE logo with a red and blue gear icon. To its right is the word "NETPIE" in a bold, sans-serif font, with "Where Design Meets" in smaller text below it. A black ribbon graphic is positioned next to the logo. The top navigation bar includes links for "HOME", "PARTNERS", "DEVELOPERS", "BLOG", and a prominent blue "LOG IN" button. A "Google Custom" search bar is located in the top right corner. The main content area features a large, scenic background image of a city skyline under a cloudy sky. Overlaid on this is a white login form with the word "LOGIN" at the top. It contains fields for "USERNAME OR EMAIL ADDRESS" (with "lamloei" typed in) and "PASSWORD" (with a series of dots). Below these fields is a blue "LOGIN" button. At the bottom of the form is a link "Forgot your password?".



The screenshot shows the NETPIE "RESOURCES > APPLICATIONS +" section. The top navigation bar is identical to the login page, with the "APPLICATIONS" tab being the active one. A dropdown menu for "APPLICATIONS" is open, showing options like "FREEBOARDS" and "FEEDS (beta)". On the left, there's a large blue box with the number "1" and the text "APPLICATION". Inside this box is a thumbnail for an application named "lamloei". To the right of this box is another blue box labeled "THINGS". At the bottom of the "APPLICATION" box is a blue button with a white plus sign (+), which is circled in red. Below the "APPLICATION" box is a search bar containing the text "LAML". At the very bottom are two buttons: "CREATE" in blue and "CANCEL" in white.



ดาวน์โหลด Library

- ดาวน์โหลดไฟล์ https://github.com/lamloei/ESP32_Microgear
- แตกไฟล์ลงโฟลเดอร์ Arduino/libraries
- ปิดและเปิด Arduino IDE ใหม่

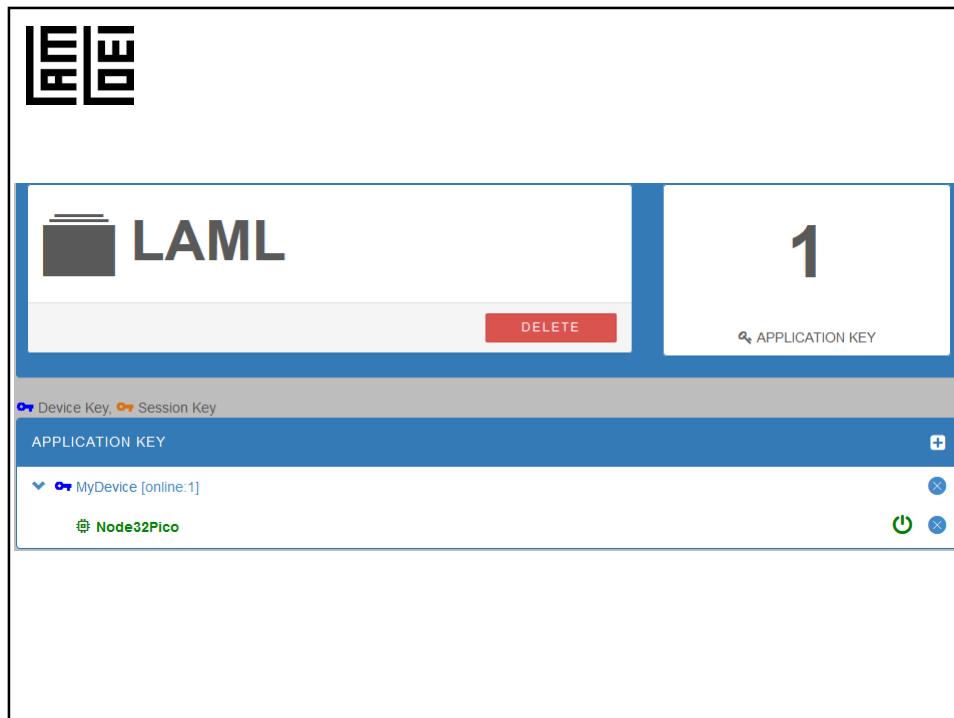
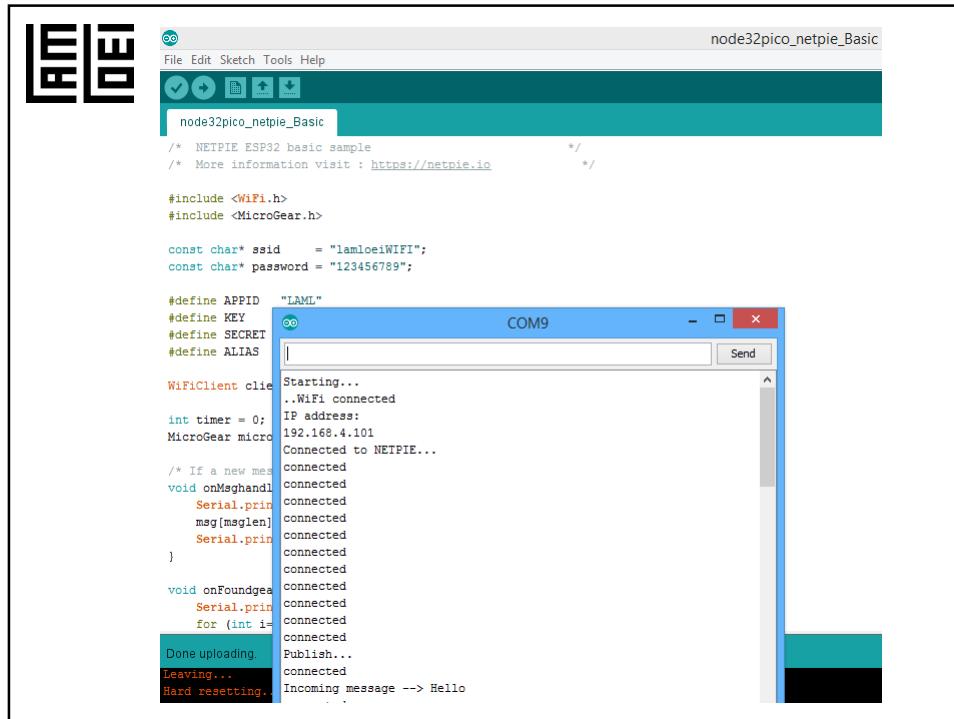
This PC > Documents > Arduino > libraries

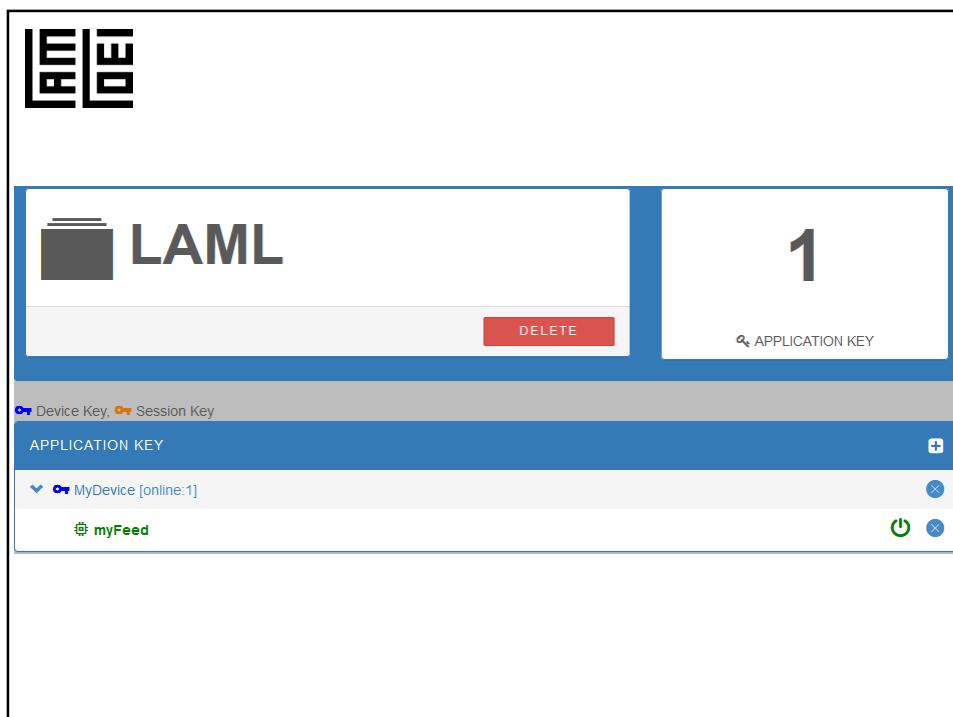
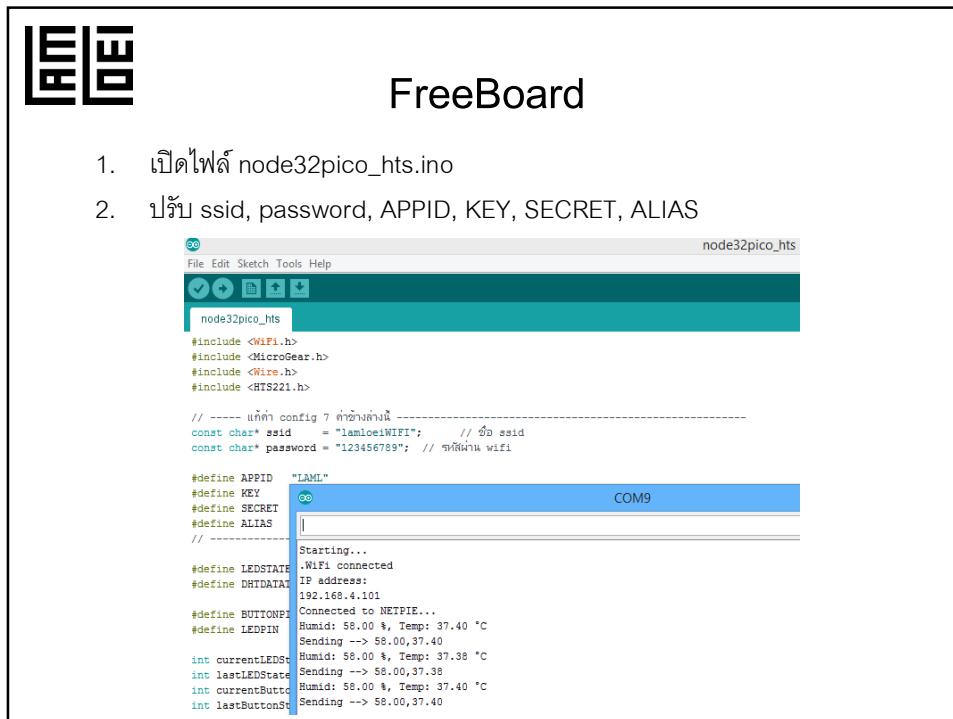
Name	Date modified	Type	Size
ESP32_Microgear-master	17/2/2561 15:52	File folder	
sme-hts221-library-master	17/2/2561 13:51	File folder	
SparkFun_LSM9DS1_Arduino_Library	17/2/2561 13:51	File folder	

File > Examples > ESP32 Microgear > Basic

```
//ใช้ได้ค่าตัวแปลง
const char* ssid    = "lamloeiWIFI";
const char* password = "123456789";

#define APPID  <APPID>
#define KEY    <APPKEY>
#define SECRET <APPSECRET>
#define ALIAS   "Node32Pico"
```





NETPIE RESOURCES > FREEBOARDS

CLICK +
ตั้งชื่อ แล้วคลิกปุ่ม CREATE

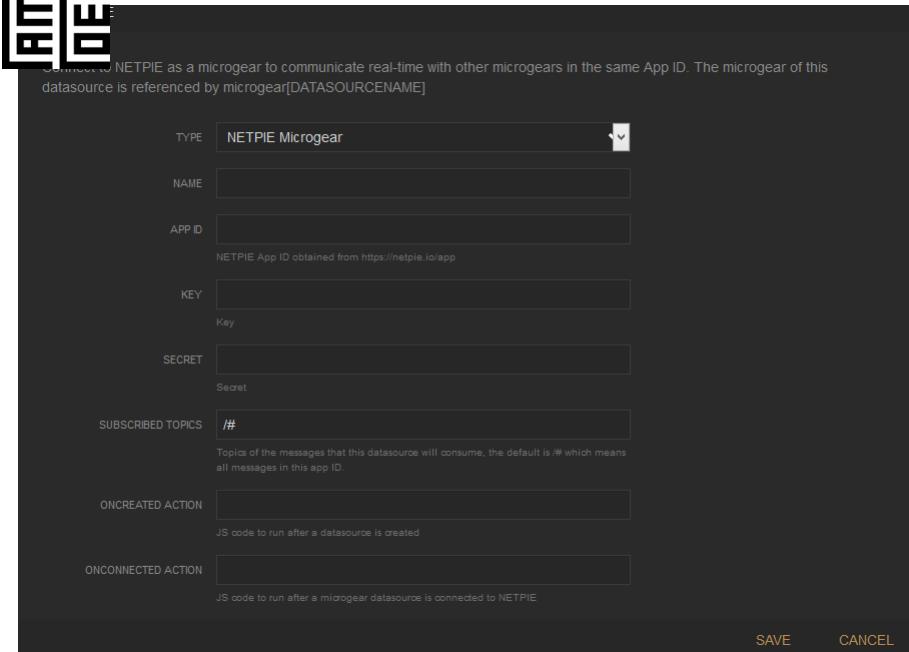
NETPIE lamloeiFreeboard

CLICK ที่ชื่อที่ตั้งขึ้นใหม่
คลิกปุ่ม ADD

NETPIE Freeboard

DATASOURCES

ADD



Connect to NETPIE as a microgear to communicate real-time with other microgears in the same App ID. The microgear of this datasource is referenced by microgear[DATASOURCENAME]

TYPE	NETPIE Microgear
NAME	<input type="text"/>
APP ID	<input type="text"/>
NETPIE App ID obtained from https://netpie.io/app	
KEY	<input type="text"/>
Key	
SECRET	<input type="text"/>
Secret	
SUBSCRIBED TOPICS	/#
Topics of the messages that this datasource will consume, the default is /# which means all messages in this app ID.	
ONCREATED ACTION	<input type="text"/>
JS code to run after a datasource is created	
ONCONNECTED ACTION	<input type="text"/>
JS code to run after a microgear datasource is connected to NETPIE	
SAVE CANCEL	

DataSource myFreeboard

- TYPE NETPIE Mirogear
- ตั้งชื่อ NAME (myFreeboard)
- ใส่ค่า APP ID
- KEY
- SECRET
- แล้วคลิกปุ่ม SAVE

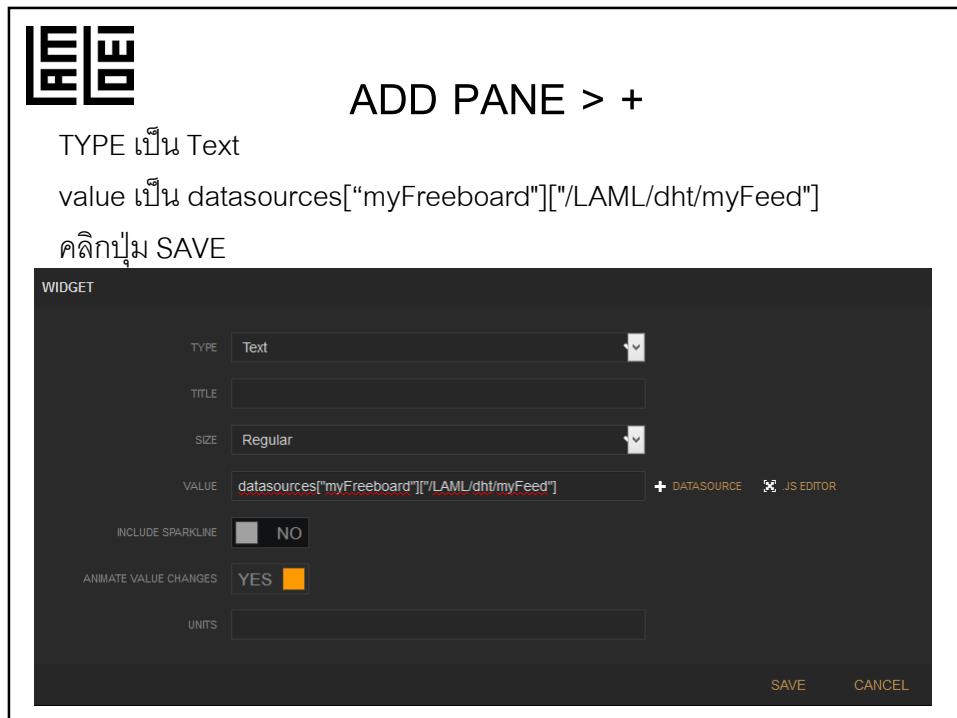
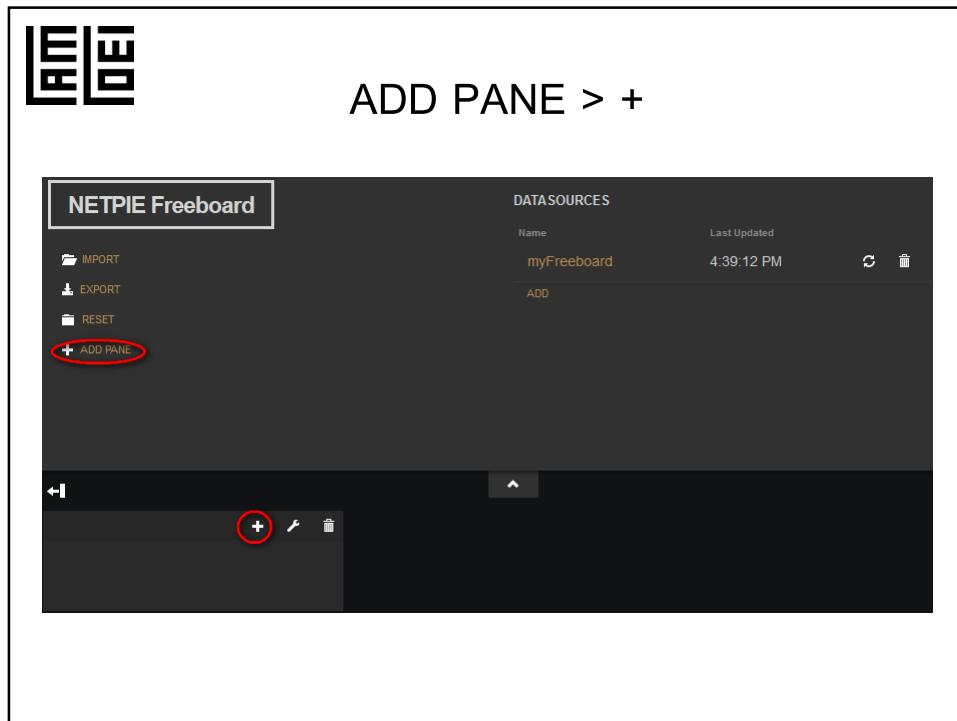
ถ้าเขียนต่อได้จะขึ้นชื่อ Name ที่ตั้งไว้ กับค่า Last Updated

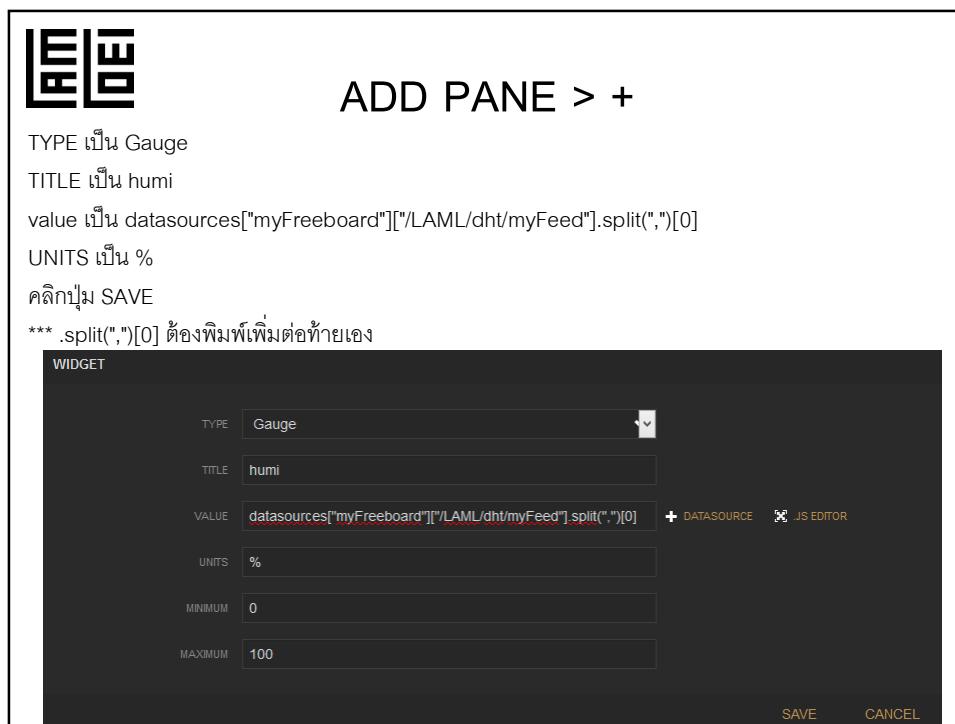
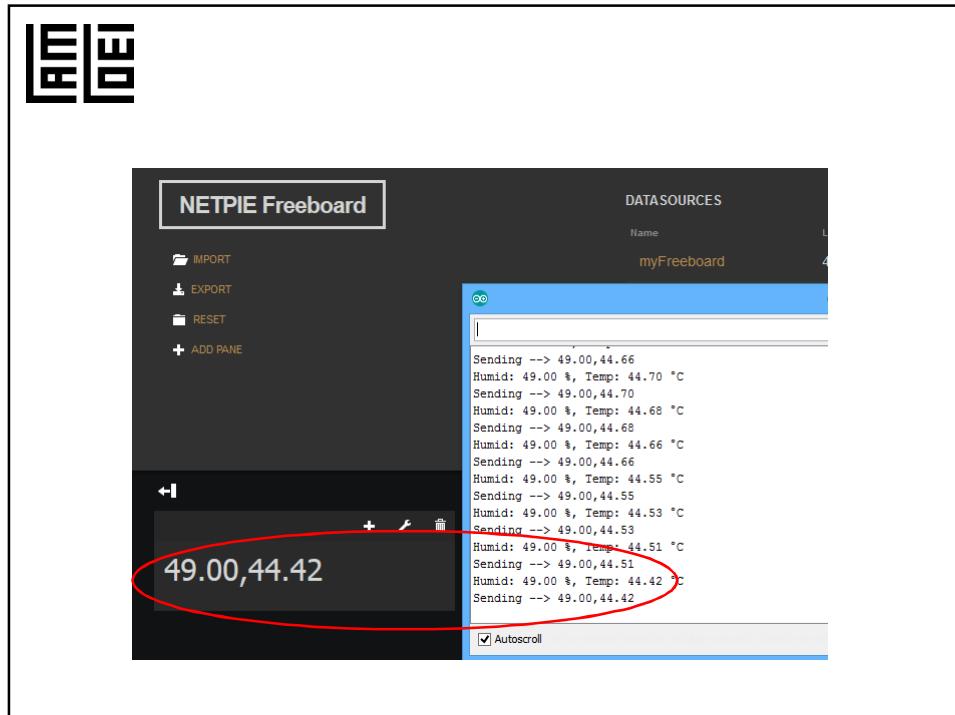
NETPIE Freeboard

- IMPORT
- EXPORT
- RESET
- + ADD PANE

DATASOURCES

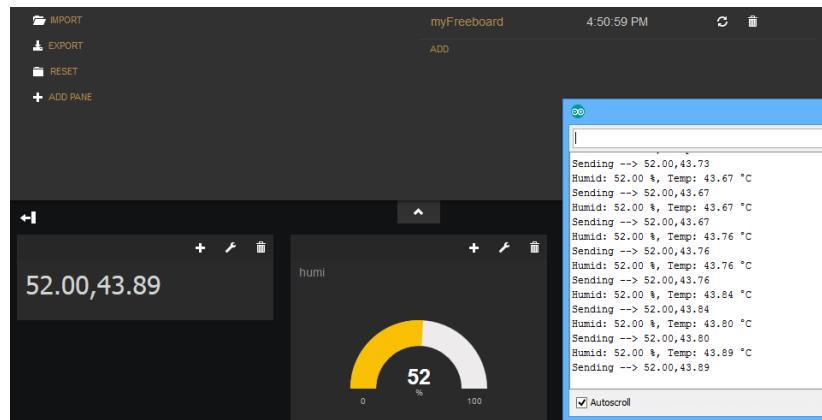
Name	Last Updated
myFreeboard	4:36:48 PM
ADD	







สามารถขยับ panel ได้



ADD PANE > +

TYPE เป็น Gauge

TITLE เป็น ชื่อหน่วย

value เป็น datasources["myFreeboard"]["/LAML/dht/myFeed"].split(",")[1]

UNITS เป็น C

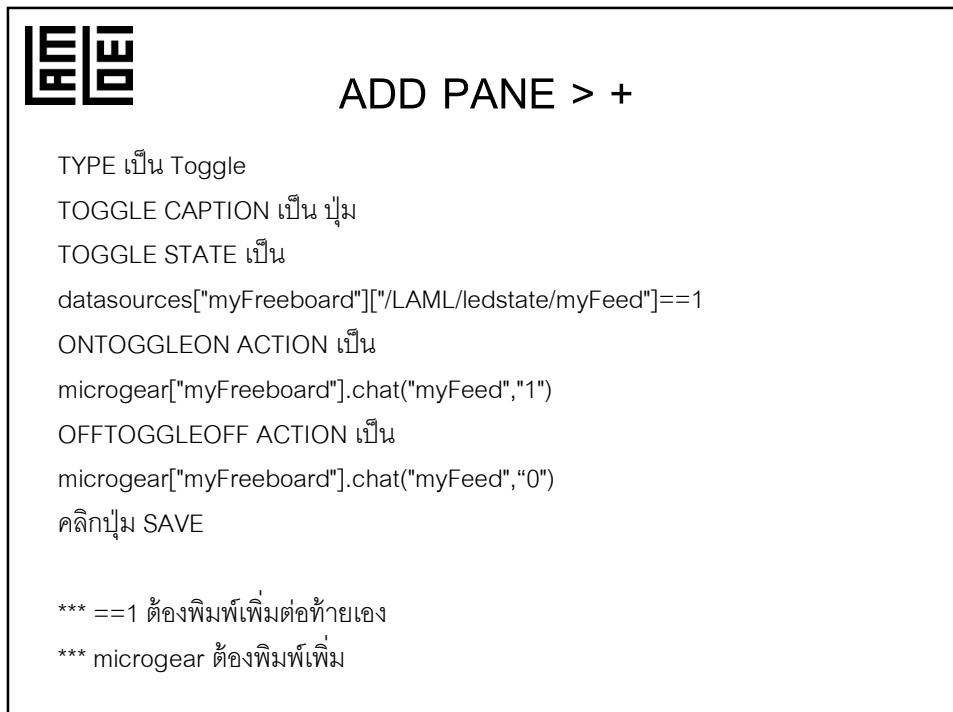
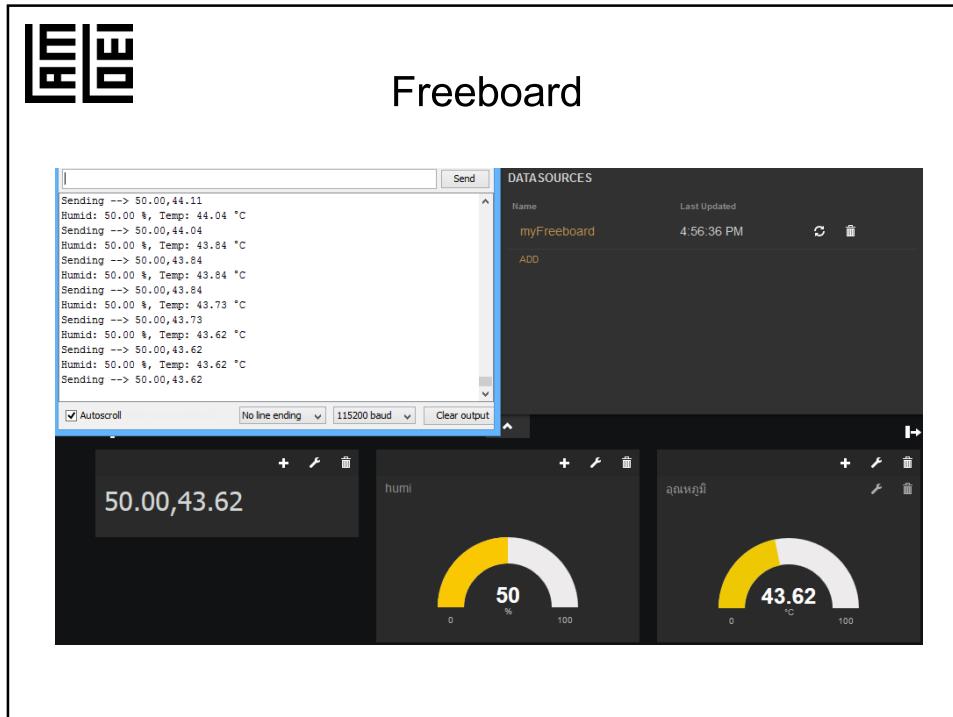
คลิกปุ่ม SAVE

*** .split(",")[1] ต้องพิมพ์เพิ่มต่อท้ายเอง

The dialog box has the following fields:

- TYPE: Gauge
- TITLE: ชื่อหน่วย
- VALUE: `datasources["myFreeboard"]["/LAML/dht/myFeed"].split(",")[1]`
- UNITS: °C
- MINIMUM: 0
- MAXIMUM: 100

At the bottom are 'SAVE' and 'CANCEL' buttons.



The screenshot shows a configuration dialog for a 'Widget'. The title bar says 'WIDGET'. The main area is titled 'WIDGET' and contains the following fields:

- TYPE:** Toggle
- TOGGLE CAPTION:** 1μ
- TOGGLE STATE:** datasources["myFreeboard"]"/LAMI/ledstate/myFeed"]==1
Add a condition to switch a toggle state here. Otherwise it just toggle by click.
- ON TEXT:** ON
- OFF TEXT:** OFF
- ONTOGGLEON ACTION:** microgear["myFreeboard"].chat("myFeed","1")
JS code to run when a toggle is switched to ON
- ONTOGGLEOFF ACTION:** microgear["myFreeboard"].chat("myFeed","0")
JS code to run when a toggle is switched to OFF
- ONCREATED ACTION:** (empty)

At the bottom right are 'SAVE' and 'CANCEL' buttons.

The screenshot shows a Freeboard dashboard titled 'Freeboard'. At the top, there is a terminal window displaying the following output:

```
Incoming message --> 1
Incoming message --> 0
Humid: 48.00 %, Temp: 45.99 °C
Sending --> 48.00,45.99
```

Below the terminal are three data visualization components:

- A toggle switch labeled 'ON'.
- A gauge chart labeled 'humid' showing a value of 48%.
- A gauge chart labeled 'Temp' showing a value of 44.8 °C.

The entire dashboard is contained within a frame with a logo in the top-left corner.



THE END

- Q & A

