

หนังสือประกอบการอบรมเชิงปฏิบัติการ  
หัวข้อการประยุกต์ใช้ IOT ขั้นสูง

โครงการ RSU START UP

จัดทำโดย  
หลักสูตรเทคโนโลยีสื่อสังคม  
วิทยาลัยเทคโนโลยีสารสนเทศและการสื่อสาร

มหาวิทยาลัยรังสิต

# สารบัญ

บทที่ 1 แนะนำ ESP32.....	4
ESP32.....	6
ซอฟต์แวร์ที่ใช้พัฒนา.....	8
ESP32 โมดูล.....	9
บอร์ดพัฒนา ESP32.....	12
เบรียบเทียบระหว่าง ESP8266 และ ESP32.....	16
การติดตั้ง ESP32 ใน Arduino IDE.....	20
บอร์ดพัฒนา ESP32.....	22
การเซ็ตค่าบอร์ด ESP32 ใน Arduino ide.....	33
ทดสอบการอัปโหลด sketch.....	34
บทที่ 2 การทำงานแบบมัลติทาสกิ้ง.....	38
EX2_MultiTasking_Milli_LED.....	41
EX3_MultiTasking_FreeRTOS_DHT22.....	43
EX4_MultiTasking_FreeRTOS_OLED_DHT22.....	46
เซ็นเซอร์สัมผัส.....	48
EX5_TouchSensor_Read.....	50
โหมดประหยัดพลังงาน.....	52
EX6_DeepSleep_Touchpad_Wakeup_DHT22_OLED.....	57
EX7_DeepSleep_Touchpad_Wakeup_DHT22_OLED.....	60
EX8_DeepSleep_Ext0_Wakeup_DHT22_OLED_Line.....	63
หน่วยความจำและพื้นที่เก็บข้อมูล.....	66
EX9_FlashMemory_NVS_BootCount.....	69
EX10_FlashMemory EEPROM.....	72
EX11_FlashMemory_SPIFFS.....	74
EX12_FlashMemory_SPIFFS_Line.....	79
บทที่ 3 ESP32 Framework.....	82
การรับพิกัดตำแหน่ง GPS.....	83
EX13_BlynkGPS.....	84
aREST Framework.....	87
EX14_aREST_Serial.....	94
EX15_aREST_WiFi_API.....	103
EX16_aREST_WiFi_UI_API.....	108
EX17_aREST_WiFi_UI_WEB_API.....	110

# สารบัญ

บทที่ 4 Blynk Cloud Server.....	112
การสร้าง Blynk Cloud บน DigitalOcean.....	113
การติดตั้ง Blynk.....	131
บทที่ 5 LoRa.....	144
EX18_LoRaReceiver.....	160
EX19_LoRaSender.....	162
EX20_LoRaReceiverInterrupt.....	164
EXT21_LoRaMultiCommunication_OLED_Client.....	166
EXT22_LoRaMultiCommunication_OLED_Base.....	168

# บทที่ 1 แนะนำ ESP32

รู้จัก ESP32 และเปรียบเทียบกับ ESP8266

# หัวข้อ

## ESP32

- แนะนำ ESP32
- ซอฟต์แวร์ที่ใช้พัฒนา
- ESP32 โมดูล
- บอร์ดพัฒนา ESP32
- เปรียบเทียบระหว่าง ESP8266 และ ESP32
- การติดตั้ง ESP32 ใน Arduino IDE
- การเขียนค่าบอร์ด ESP32 ใน Arduino ide
- ตัวอย่างการเขียนโค้ด
- ทดสอบการอัพโหลด sketch

# แนะนำ ESP32



## ESP32 คือ

ESP32 เป็นชิปของไอซีไมโครคอนโทรลเลอร์ที่รองรับการเชื่อมต่อ WiFi มีความสามารถในการเชื่อมต่อ Bluetooth Low-Energy (BLE, BT4.0, Bluetooth Smart) ผลิตโดยบริษัท Espressif จากประเทศไทย

ESP32 ได้แก้ไขจุดด้อยต่างๆของ esp8266 ไปจนหมด ไม่ว่าจะเป็นเรื่องของ I/O และ Analog input ที่มีไม่เพียงพอ กับการใช้งาน และ ปรับสเปคของ hardware ให้สูงขึ้น มีความเสถียรภาพสูง

## การพัฒนา ESP32

หลังจากที่บริษัท Espressif ได้ออกไอซี ESP8266 และได้รับความนิยมสูงสุด ก็ได้ออกไอซีรุ่นใหม่ ชื่อว่า ESP31B มีการพัฒนาชุดซอฟแวร์ ESP32\_RTOS\_SDK ไปพร้อมกัน แต่หลังจากนั้นไม่นาน บริษัท Espressif ได้ยกเลิกการใช้ชุดซอฟแวร์พัฒนาดังกล่าว และไปสร้างชุดพัฒนาใหม่ชื่อ ESP-IDF และได้ออกไอซี ESP32 เป็นครั้งแรก

บริษัท Espressif ได้ผลิตโมดูล ESP-WROOM-32 ออกมา จำนวนนี้ บริษัท AiThinker และ Seeedstudio ก็ได้ผลิตโมดูล ESP3212 ขึ้นมา แต่ด้วยปัญหาด้านการออกแบบจึงได้ยกเลิก การผลิตแล้วหันไปผลิต ESP32S แทน โดยมีลักษณะเหมือนกับ ESP-WROOM-32 ทุกอย่าง

ESP32 คือ

บริษัท Espressif ได้ทีมผู้พัฒนา Arduino core for ESP8266 WiFi chip และทีมผู้พัฒนา Arduino core for ESP31B WiFi chip มาร่วมงานช่วยพัฒนาชุดไลบารีและคอมไพล์เลอร์ สำหรับ Arduino ทำให้การพัฒนาเป็นไปอย่างรวดเร็ว

ESP-IDF ไม่มีโปรแกรม Editor แต่เป็นแกนพัฒนาหลัก เมื่อมีการเพิ่มฟีเจอร์ใหม่ๆ ถึงจะเพิ่มใน Arduino core for ESP32 WiFi chip ปัจจุบันเราสามารถใช้ Arduino IDE พัฒนา ESP32 ได้

## Key Features

- 240 MHz dual core Tensilica LX6 microcontroller with 600 DMIPS
  - Integrated 520 KB SRAM
  - Integrated 802.11 b/g/n HT40 Wi-Fi transceiver, baseband, stack and LwIP
  - Integrated dual mode Bluetooth (classic and BLE)
  - 16 MB flash, memory-mapped to the CPU code space
  - 2.3V to 3.6V operating voltage
  - -40°C to +125°C operating temperature
  - On-board PCB antenna / IPEX connector for external antenna

## Security Related

WEP, WPA/WPA2 PSK/Enterprise

## Hardware-accelerated encryption: AES/SHA2/Elliptical

## Curve Cryptography/RSA-4096

## Performance

Max data rate of 150 Mbps@11n HT40, 72 Mbps@11n HT20, 54

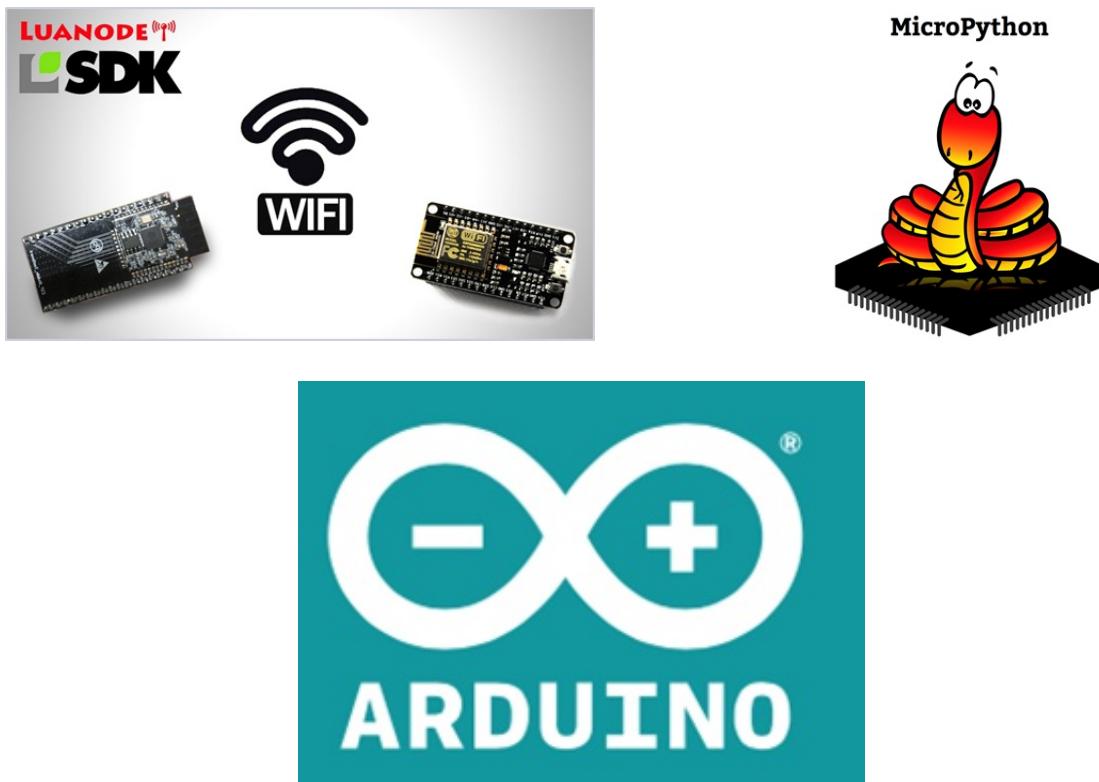
Maximum transmit power of 19.5 dBm@11b, 16.5 dBm@11g, 15.5 dBm@11n

Minimum receiver sensitivity of -97 dBm

5  $\mu$ A power consumption in Deep-sleep

# ซอฟต์แวร์ที่ใช้พัฒนา

## Software Development



The screenshot shows a graphical programming interface for the Espruino microcontroller. A script is displayed with the following logic:

```
watch BTN1 rising edge
do digitalWrite Pin LED1 Value true
wait 1 seconds
do digitalWrite Pin LED1 Value false
```

**The Graphical Editor lets you use Espruino even if you can't code!**

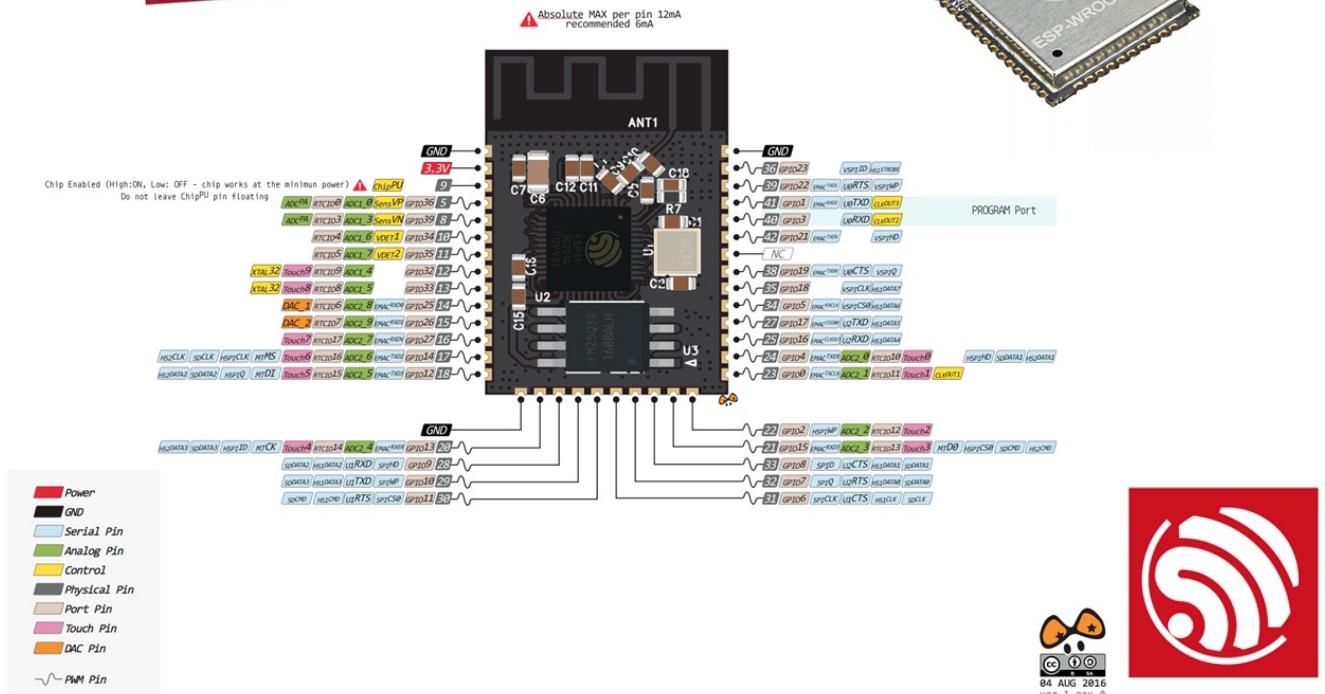
The optional graphical editor makes it quick and easy to create simple devices, even if you've never programmed before.

MicroPython-ESP32 รองรับการใช้งานพื้นฐานภาษา Python ส่วนใหญ่ได้ รองรับการจัดการ WiFi การเชื่อมต่อต่างๆ ของ ESP32 ส่วน Espruino on ESP32 ใช้ภาษา JavaScript ในการสั่งงานและรองรับการเขียนโปรแกรมแบบ Text และภาษาบล็อก (Block) และ LuaNod ก็รองรับคำสั่งที่ใช้บน Lua จริง ๆ แทบทุกคำสั่ง และรองรับการควบคุม WiFi ได้เต็มรูปแบบ

# ESP32 โมดูล

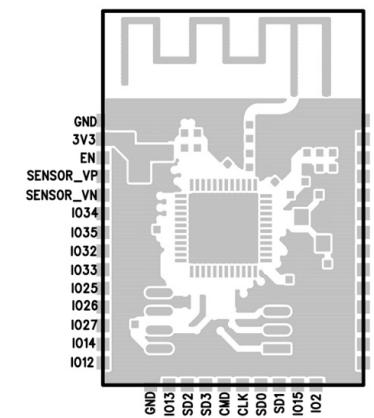
## ESP32 Module – WROOM-32

### WROOM32 PINOUT



ผลิตโดยบริษัท Espressif (บริษัทผู้ผลิตชิปไอซี ESP32) มีอยู่ด้วยกัน 2 เวอร์ชัน เวอร์ชันแรกจะแสดงเฉพาะโลโก้ของบริษัท Espressif และมีคำว่า ESP-WROOM-32 ส่วนเวอร์ชันที่สอง เกิดจากการที่ซิปได้ผ่านการทดสอบได้ในรับรองจากมาตรฐานต่าง จึงได้ออกรุ่นที่แสดงหมายเลขในรับรองบนซิปอุปกรณ์ แล้วได้ใช้งานมาถึงรุ่นปัจจุบัน ตัวโมดูลมาพร้อมกับเสาอากาศแบบ PCB บนตัว มีกรอบคุณอุปกรณ์ทั้งหมดบนโมดูลเพื่อลดสัญญาณรบกวนลง ซึ่งจะทำให้การทำงานโดยรวมมีเสถียรภาพมากยิ่งขึ้น ชิป ESP-WROOM-32 นี้ได้รับความนิยมเป็นอย่างมากในการใช้งานในด้านต่างๆ เนื่องจากสามารถผลิตได้ทันความต้องการและมีราคาถูกกว่าโมดูลที่ผลิตโดยผู้ผลิตอื่น ๆ มาก นอกจากนี้ ตำแหน่งขาต่างๆ ของ ESP-WROOM-32 ยังถือเป็นมาตรฐานที่ไม่แตกต่างจากผู้ผลิตอื่นต้องทำตามอีกด้วย

## ESP32 Module – ESP-32S



ESP-32S คือบอร์ดโมดูล (ESP32 chip + PCB board + cover box) ของบริษัท Ai-Thinker โดยมีขนาด ขาใช้งานและขนาดของตัวถัง ลักษณะรูต่างๆ เหมือนกับ ESP-WROOM-32 ของ Espressive และสามารถใช้ทดแทนกันได้ แต่หัวน้ำราคากลางของ ESP32S แพงกว่าโมดูล ESP-WROOM-32 และ ESP32-Bit ทำให้มีค่าอยู่สูงกว่า แต่ก็มีความสามารถในการนำไปใช้งานเท่าไหร่

## EESP32 Module – ESP32-Bit

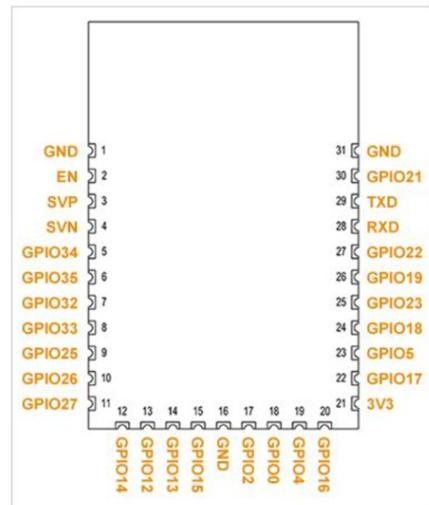
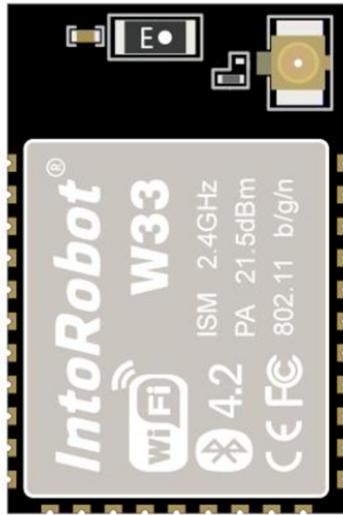
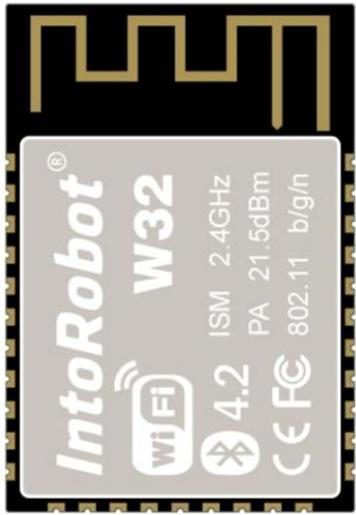


U1	
ESP32-Bit	
1	GND
2	VCC3V3
3	Sens_VP
4	GND
5	Sens_VN
6	GPIO21
7	CHIP_UP
8	TXDO
9	GPIO22
10	GPIO25
11	GPIO19
12	GPIO32
13	GPIO23
14	GPIO33
15	GPIO18
	GPIO10
	GPIO26
	GPIO17
	GPIO27
	GPIO16
	GPIO14
	GPIO12
	GPIO0/DL
	GPIO4
	GPIO13
	GPIO2
	GPIO15
	GPIO16

บริษัทผลิตบอร์ดไมโครคอมพิวเตอร์ในจีน eBox และ Widora ได้พัฒนาโมดูล ESP32-Bit ขึ้นดีคือมีขนาดเล็กกว่า ESP-WROOM-32 ทำให้ช่วยลดขนาดของบอร์ดพัฒนาลง สามารถเลือกใช้เสาอากาศภายในอกโดยใช้คอนเนคเตอร์แบบ IPLEX หรือเลือกใช้เสาอากาศแบบเซรามิกบนตัวโมดูล ส่วนข้อเสียคือมีการตัดขาดใช้งานบางส่วนออกไป ทำให้มีข้าใช้งานน้อยลง



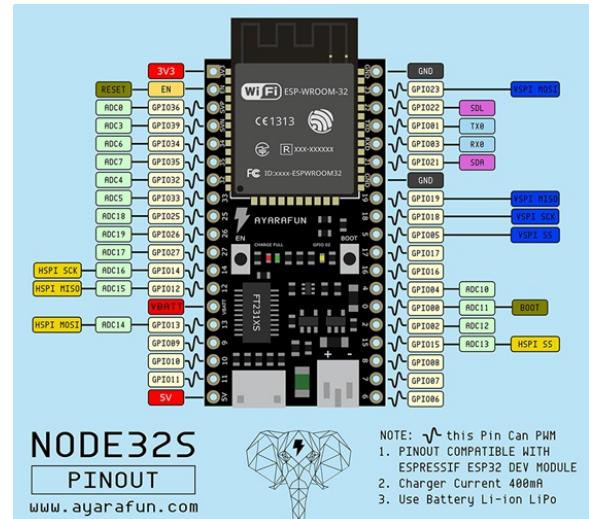
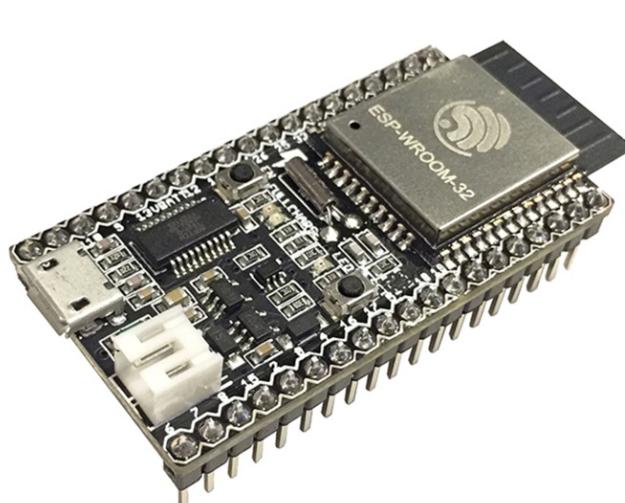
## ESP32 Module – IntoRobot-W32/W33



**IntoRobot**

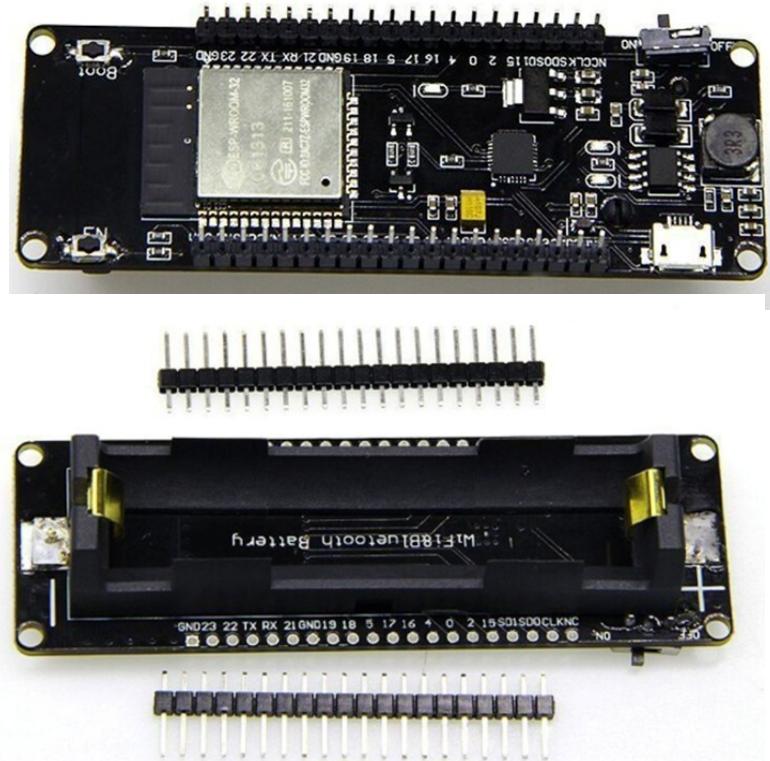
IntoRobot-W32/W33 ผลิตโดยบริษัท IntoRobot มีอยู่ด้วยกัน 2 รุ่น คือรุ่น W32 ที่ใช้เสาอากาศแบบ PCB ท่อสูญญากาศ และรุ่น W33 ที่มีเสาแบบเซรามิกเป็นเสาหลักในการรับสัญญาณและสามารถเปลี่ยนไปใช้คอนเนคเตอร์แบบ IPEX สำหรับต่อเสาอากาศภายนอกได้ ตำแหน่งขาและขนาดของโมดูลจะมีข้าต่อใช้งานทั้งหมด 31 ขา น้อยกว่า ESP-WROOM-32 / ESP32S แต่มีข้ามากกว่าโมดูล ESP32-Bit มีขนาดเล็กกว่า ESP-WROOM-32/ESP32S แต่มีขนาดใหญ่กว่าโมดูล ESP32-Bit

# บอร์ดพัฒนา ESP32



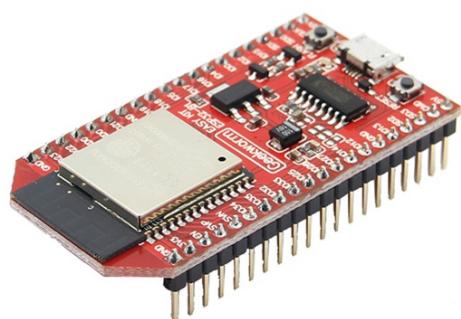
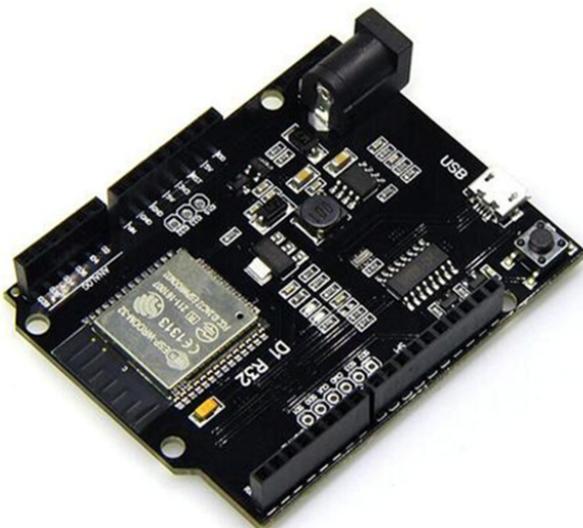
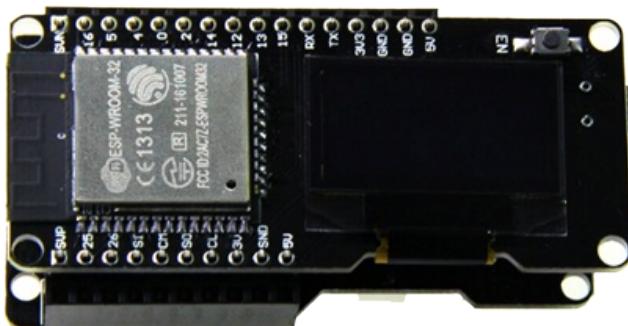
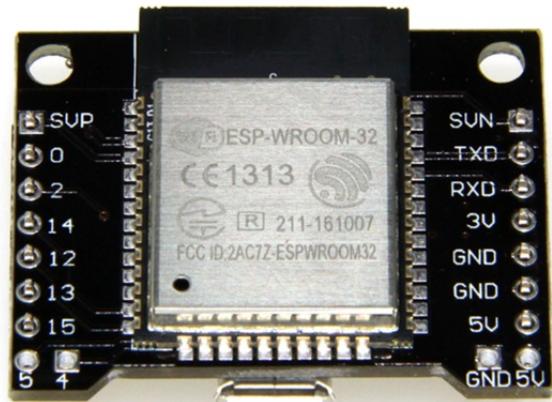
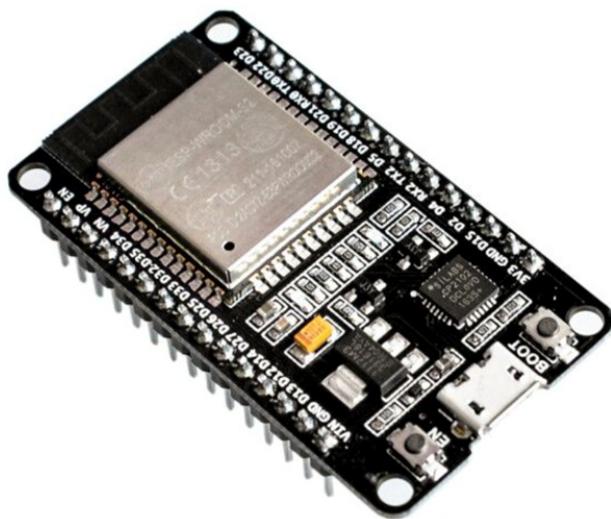
Ayarafun/LamLoei Node32S

"กูมีใจไทยอุตสาหกรรม กูมีใจไทยผลิต สนับสนุนผลงานไทย ชื่อดีอย่างภาคภูมิใจ"

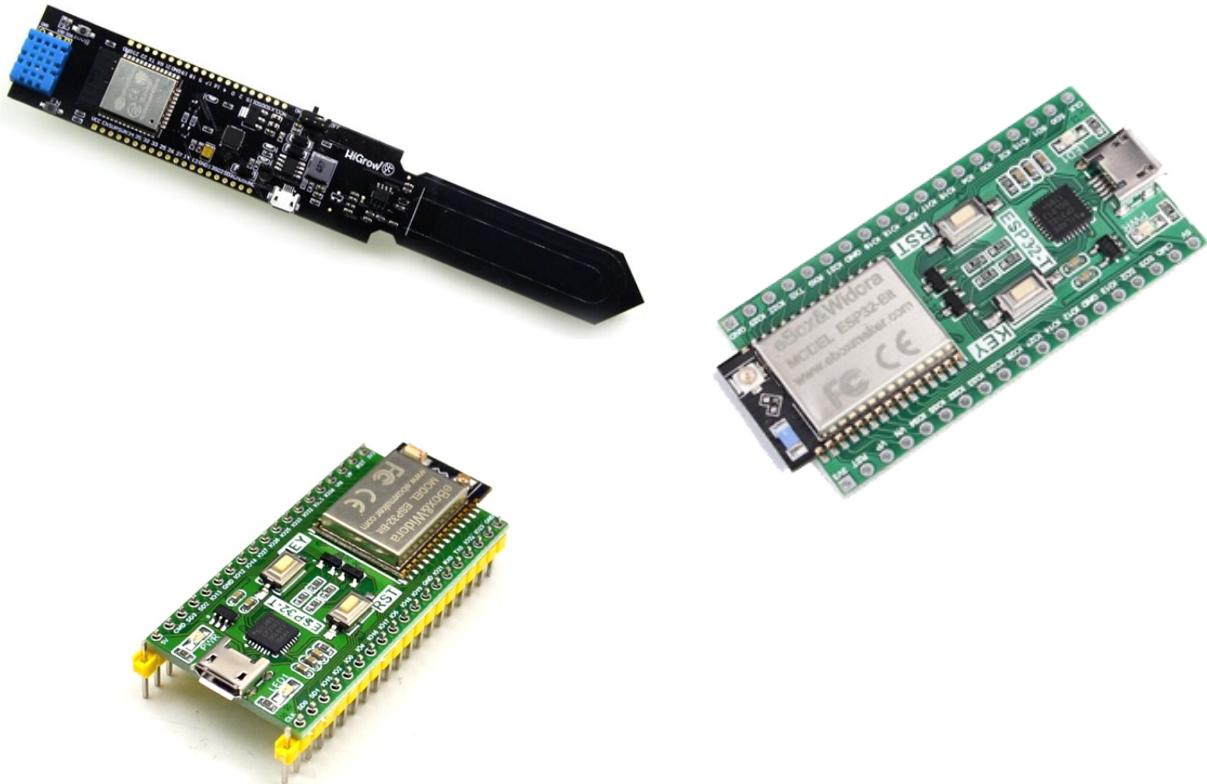


WeMos® WiFi + Bluetooth Battery ESP32 Development Tool

## บอร์ดพัฒนา ESP32

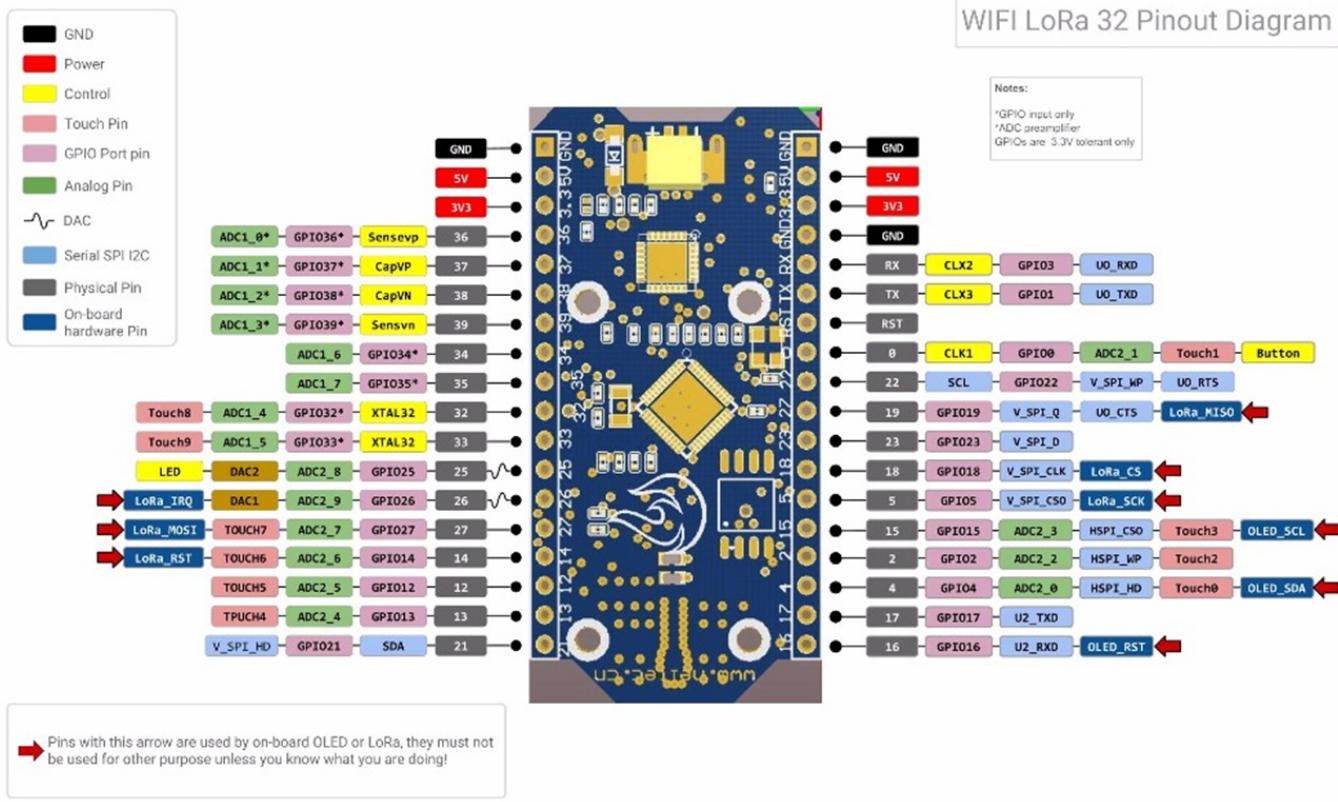


## บอร์ดพัฒนา ESP32



WiFi LoRa 32

## บอร์ดพัฒนา ESP32

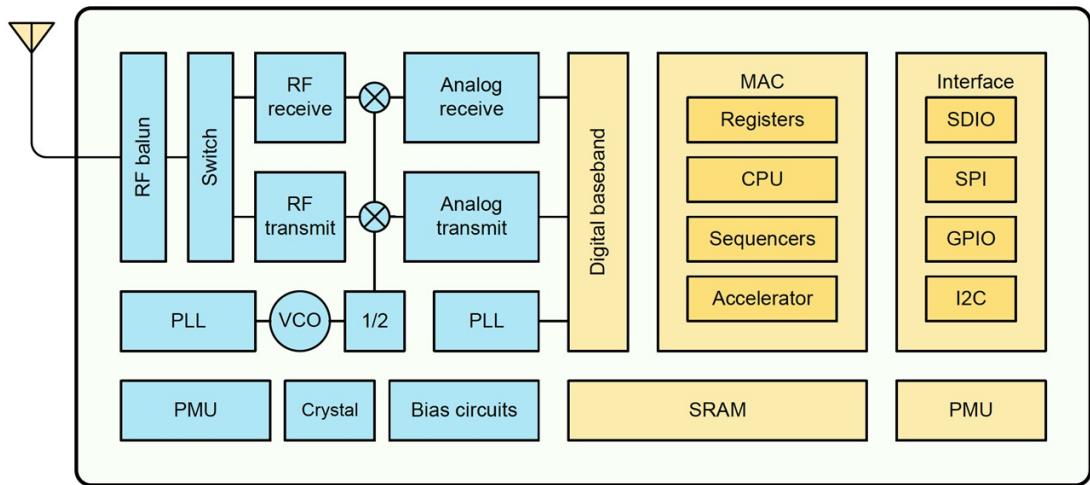


# เปรียบเทียบระหว่าง

## ESP8266 และ ESP32

	ESP8266	ESP32
	Tensilica L106 32-bit single core micro controller	Xtensa® Single-/Dual-core 32-bit LX6 microprocessor(s)
ROM	none	448 KB
RAM	160kB	520 KB
RTC RAM	768 Bytes	8kB slow + 8kB fast
QSPI flash/SRAM,	up to 1 x 16 MB	up to 4 x 16 MB
GPIOs	17	36
Wi-Fi	802.11 b/g/n/e/i	802.11 b/g/n/e/i
Bluetooth	no	yes
PWM Pins	8 Software	1 Hardware/16 software
ADC	10 bit	12 bit
DAC	no	2 x 8 bit
Touch sensor	no	10
Temperature Sensor	no	yes
Cryptographic engine	no	yes
Power supply:	2.5V to 3.6V	2.3V to 3.6V

## ESP8266 Block Diagram



แบ่งได้เป็น 2 ส่วน คือฝั่งด้านซ้าย Radio frequency (RF) part มีหน้าที่เชื่อมต่อและจัดการกับ WiFi และฝั่งด้านขวาคือ Logical RF ที่ประกอบด้วย MCU, SRAM, Interface เชื่อมต่อแบบต่างๆ สถาปัตยกรรมนี้มีขาใช้งานจำนวน 32 ขา

### ข้อจำกัด ESP8266

- MCU รักษาการเชื่อมต่อ WiFi และรันคำสั่งใน Sketch ในเวลาเดียวกัน

- wdt reset / “yield” command

- เหลือขาให้ใช้งานไม่มาก

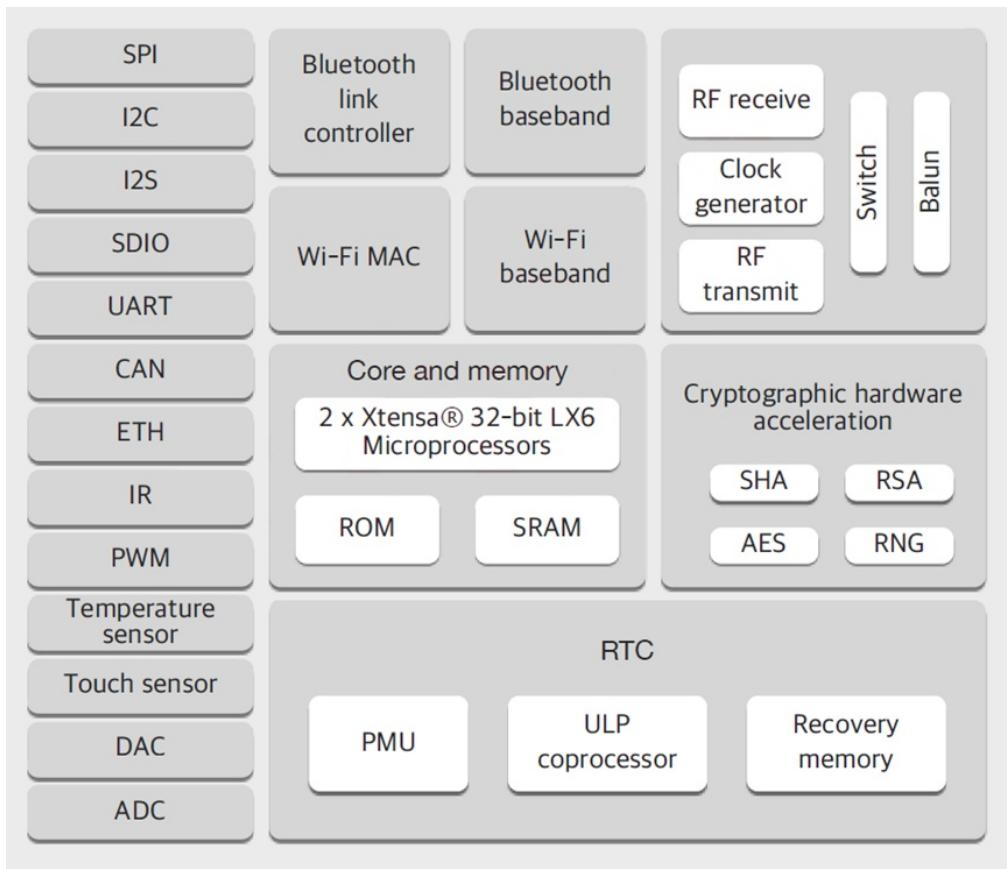
- มีหลายขาที่มีการเรียกใช้ต่อนบุต (GPIO0, GPIO2, GPIO16)
- ปัญหาระหว่างการบูต / มีขาใช้งานไม่เพียงพอ

- โหมดประหยัดพลังงาน

- รีเซ็ท signal เพื่อ wake-up
- สูญเสียการเชื่อมต่อ WiFi ใน
- ต้องเชื่อมต่อ WiFi ใหม่หลังจาก deep-sleep

วิธีรีเซ็ท signal เพื่อ wake-up ทำได้สองอย่างคือ สร้าง RTC ให้ ESP ขา GPIO16 หรือปลุกจาก signal ภายนอก

## ESP32 Block Diagram

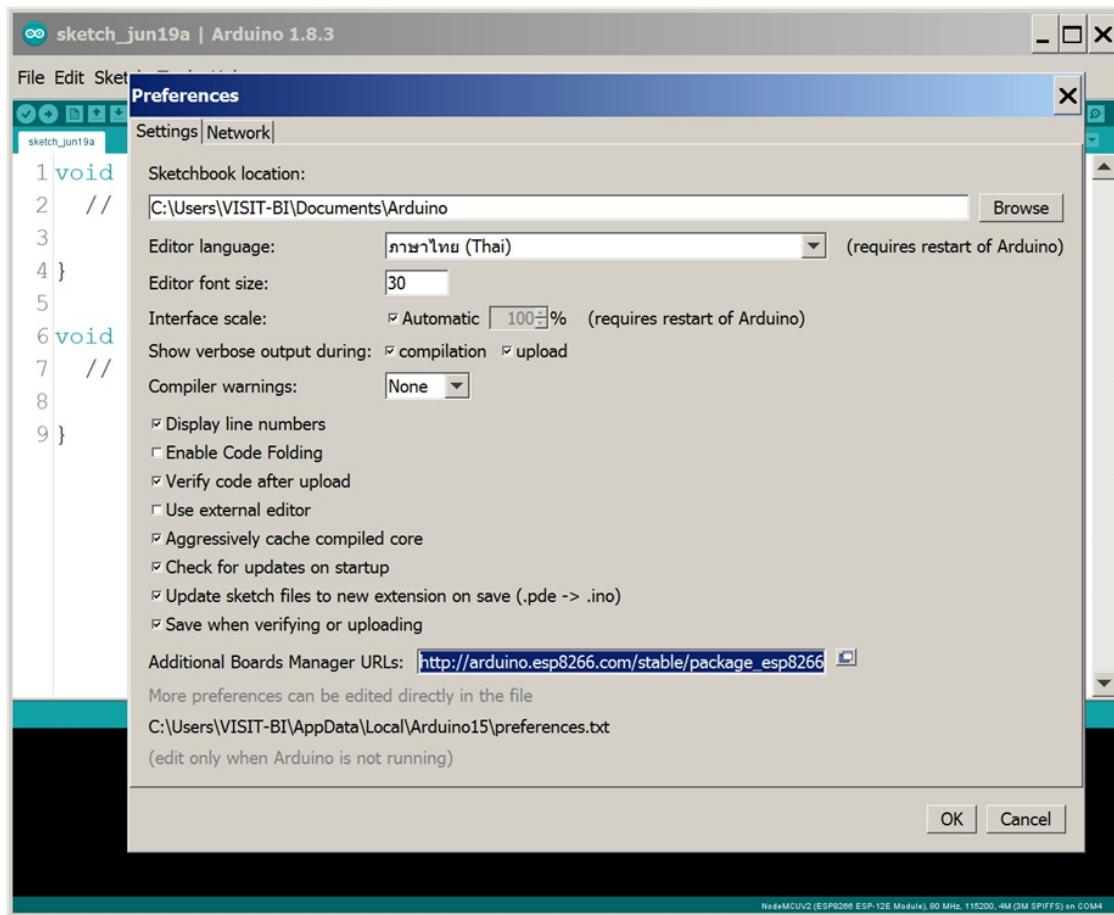


สถาปัตยกรรมนี้มี 48 pin ใช้งาน. ULP = ultra low power processor สามารถ deep sleep 2 core processor แต่ยังคงเชื่อมต่อ wifi ได้และปลูกต้นได้ ใช้กระแส 150 micro ampere. รองรับ Bluetooth 4.2 BLE Standard. ราคามาไม่แพงมากอนาคตจะถูกลง

## Advanced Peripheral Interfaces

- 12-bit SAR ADC up to 18 channels
- 2 × 8-bit D/A converters / 10 × touch sensors / Temperature sensor
- 4 × SPI / 2 × I2S / 2 × I2C / 3×UART
- 1 host (SD/eMMC/SDIO) / 1 slave (SDIO/SPI)
- Ethernet MAC interface with dedicated DMA and IEEE 1588 support
- CAN 2.0
- IR (TX/RX)
- Motor PWM
- LED PWM up to 16 channels
- Hall sensor
- Ultra low power analog pre-amplifier

## การติดตั้ง ESP8266 ใน Arduino IDE



### รูปแสดงการเพิ่มบอร์ด ESP8266 ใน Arduino IDE

การติดตั้ง ESP8266 ใน Arduino IDE ทำได้อย่างง่ายดาย โดยเพิ่มการจัดการบอร์ดจาก URL ได้เลย แต่สำหรับ ESP32 ยังไม่สามารถทำได้ เพราะยังอยู่ในช่วงพัฒนาอยู่ การติดตั้ง ESP32 จึงมีขั้นตอนที่มากกว่าและต้องทำติดตั้งเอง

# การติดตั้ง ESP32

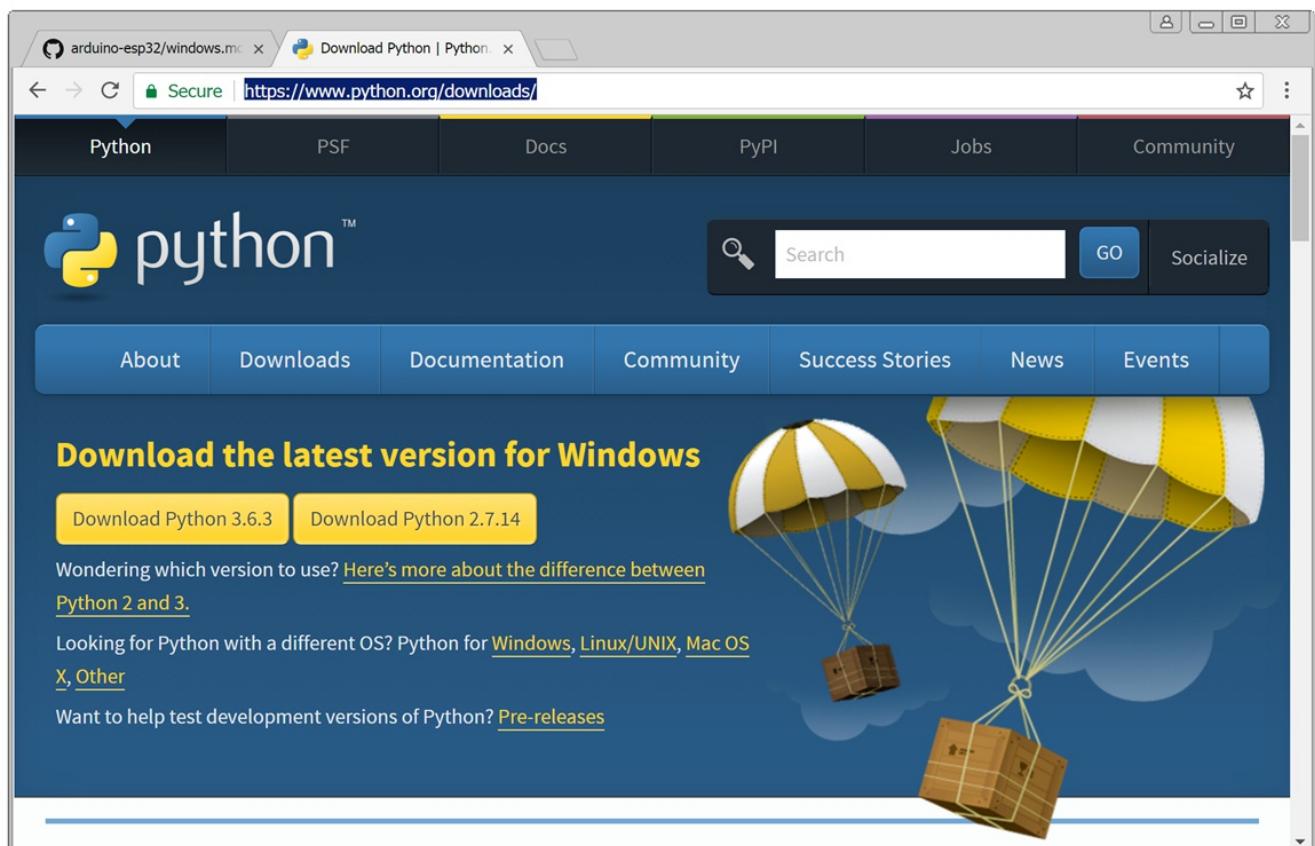
## ใน Arduino IDE

### ขั้นตอนในการติดตั้ง ESP32

1. ติดตั้ง Arduino IDE เวอร์ชันล่าสุด
2. ติดตั้ง python 2.7x
3. ติดตั้ง git
4. โคลน existing repository
5. รัน git.exe
6. ติดตั้ง USB driver (ถ้าติดตั้งแล้วให้ข้ามไป)

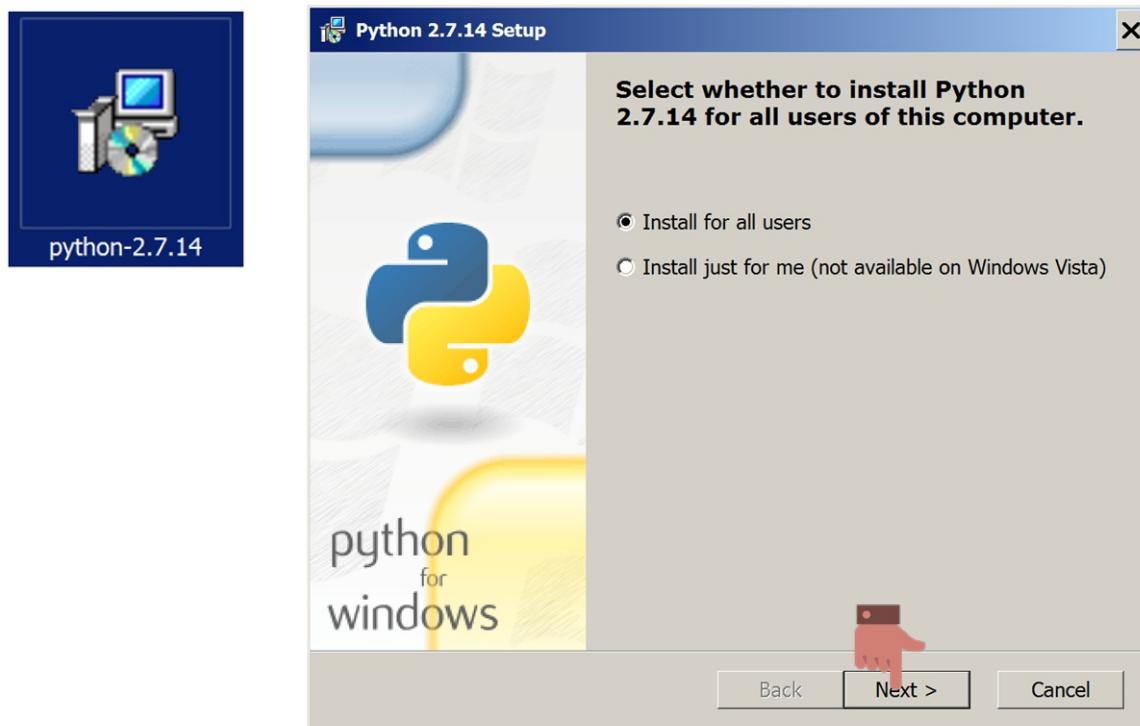
## ขั้นตอนที่ 1. ดาวน์โหลด python บน windows

<https://www.python.org/downloads/>



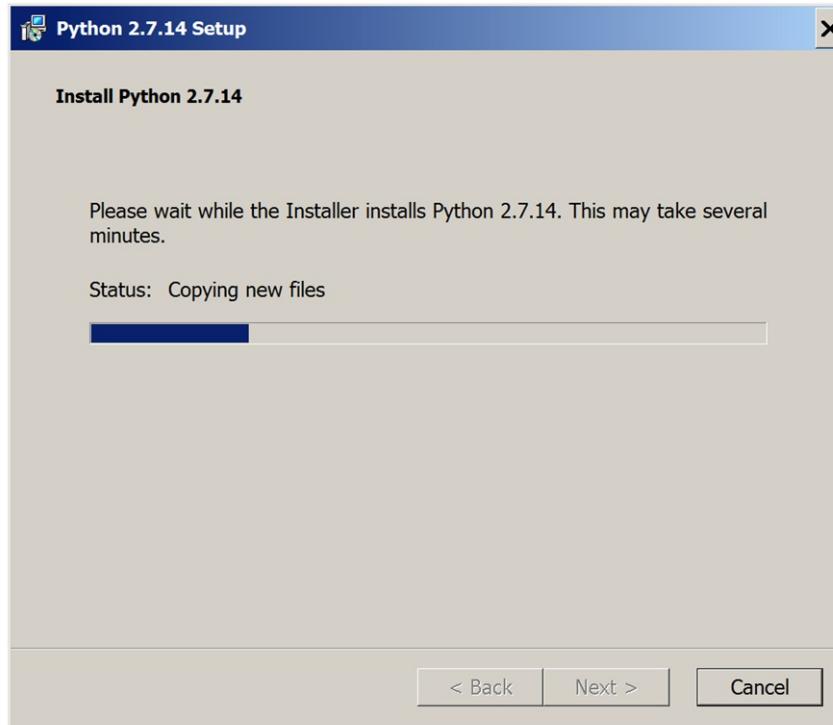
## การติดตั้ง ESP32 ใน Arduino IDE

### ขั้นตอนที่ 2. ติดตั้ง python บน windows



# การติดตั้ง ESP32 ใน Arduino IDE

## ขั้นตอนที่ 2. ติดตั้ง python บน windows



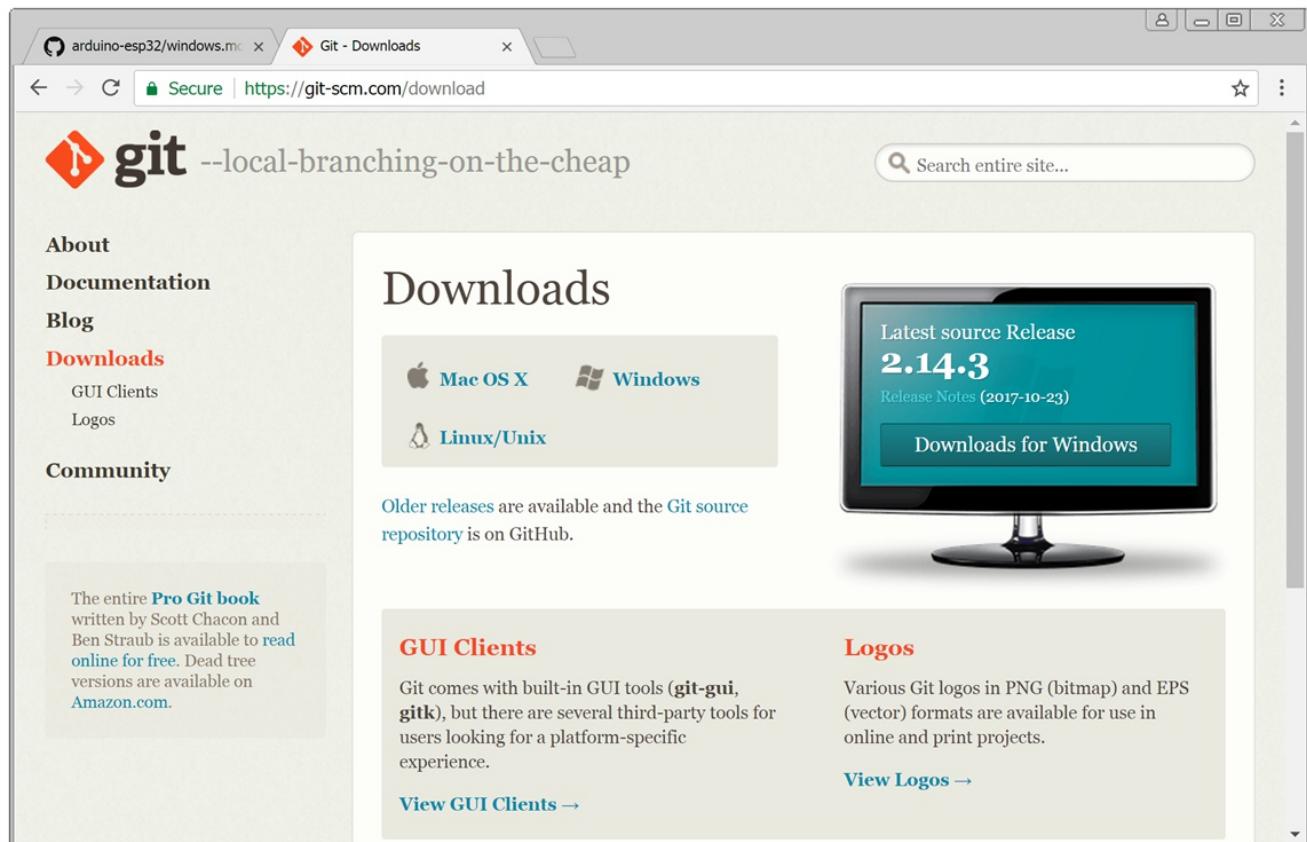
# การติดตั้ง ESP32 ใน Arduino IDE

## ขั้นตอนที่ 2. ติดตั้ง python บน windows

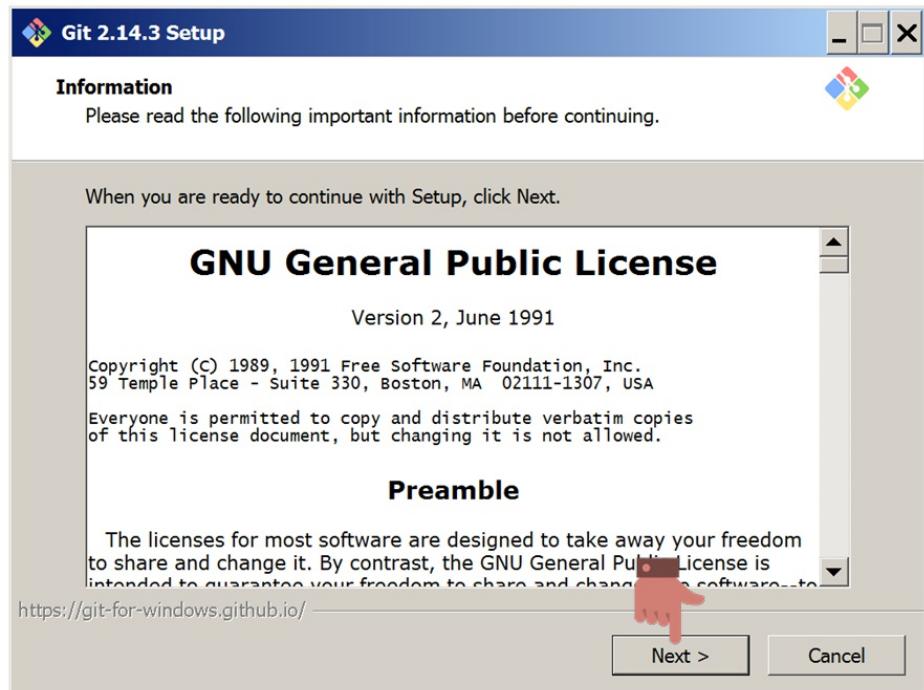


## ขั้นตอนที่ 3. ดาวน์โหลด git บน windows

<https://git-scm.com/download>

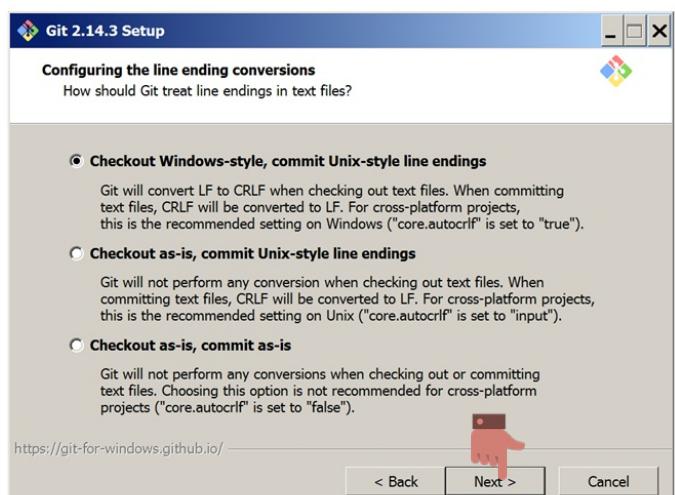
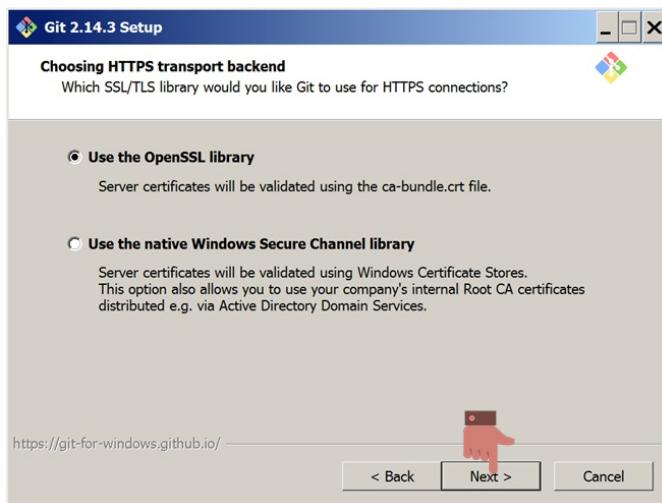
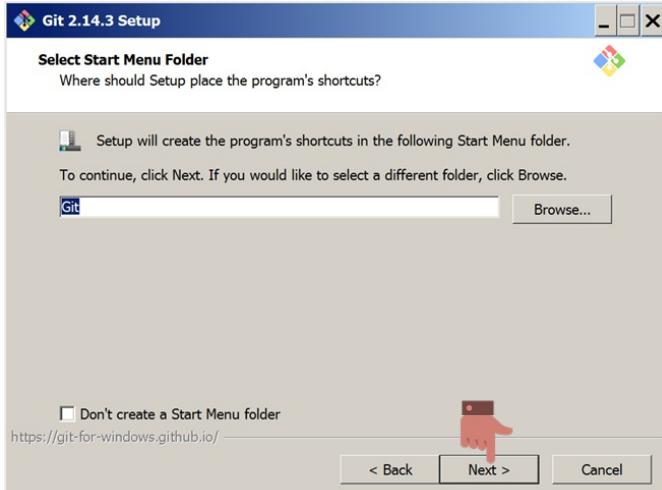


## ขั้นตอนที่ 4. ติดตั้ง git บน windows

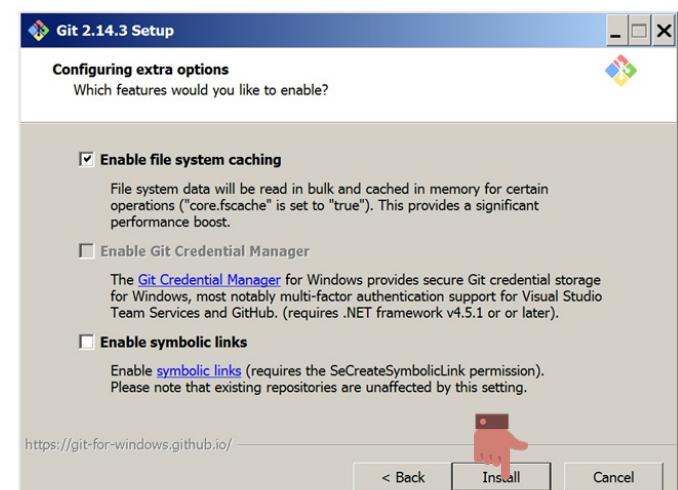
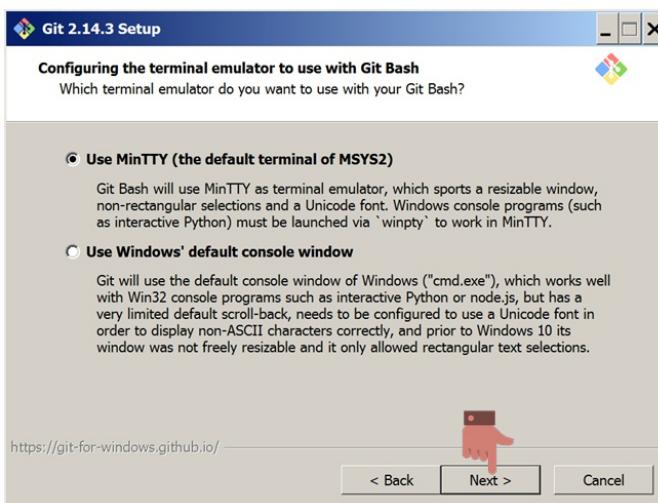
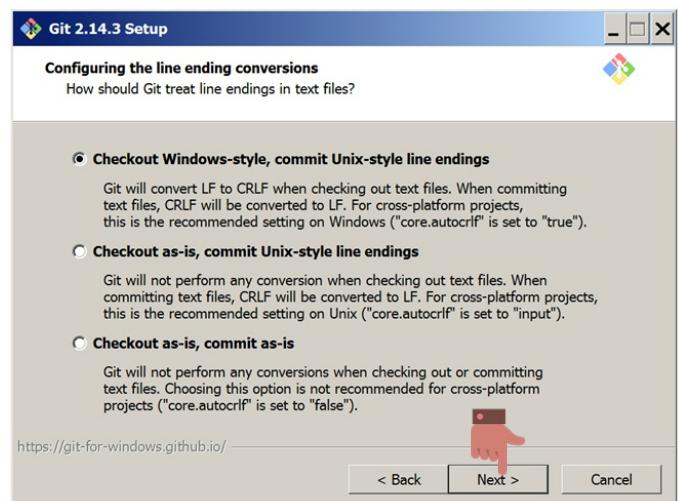
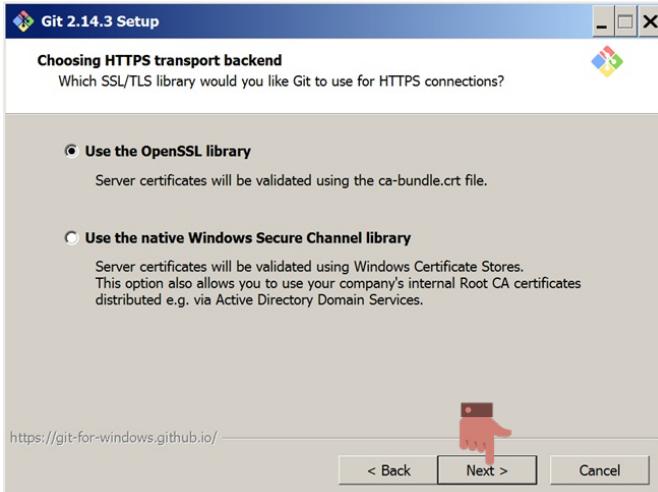


# การติดตั้ง ESP32 ใน Arduino IDE

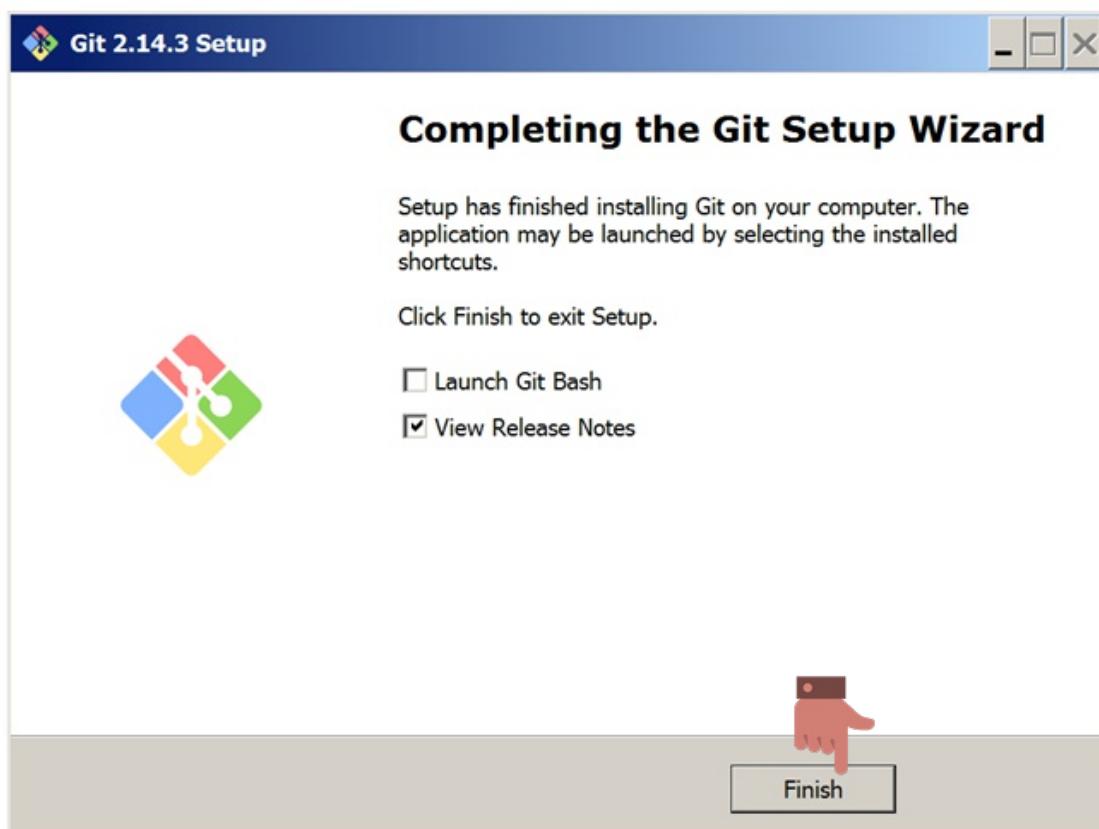
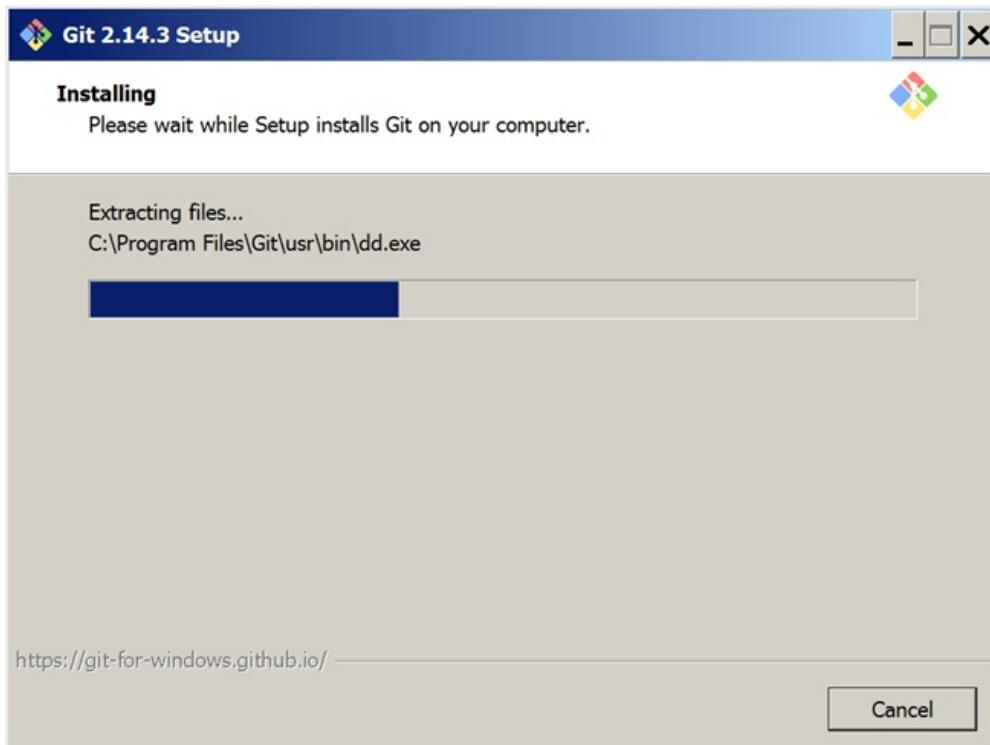
## ขั้นตอนที่ 4. ติดตั้ง git บน windows



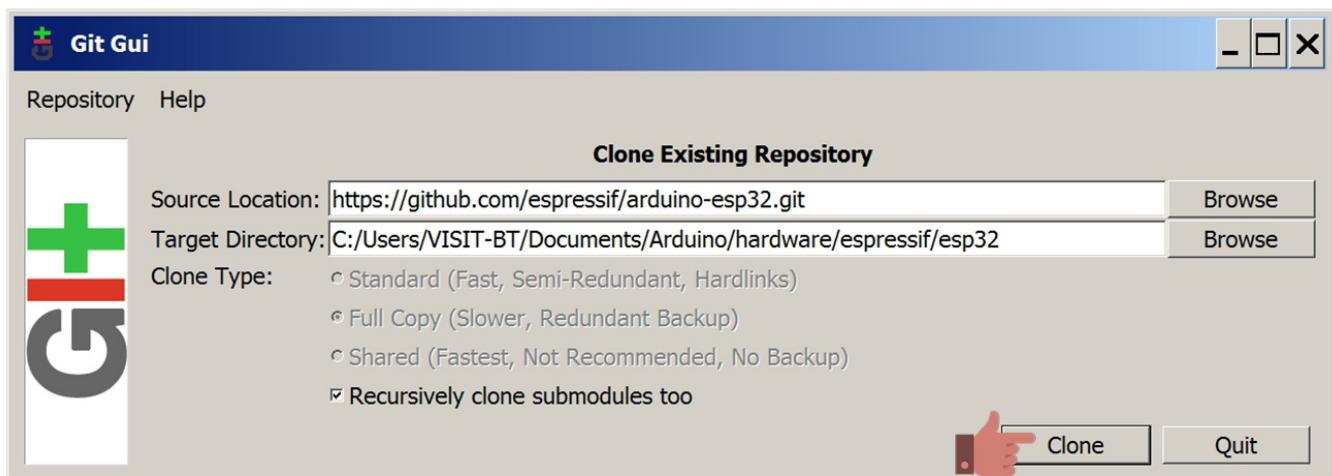
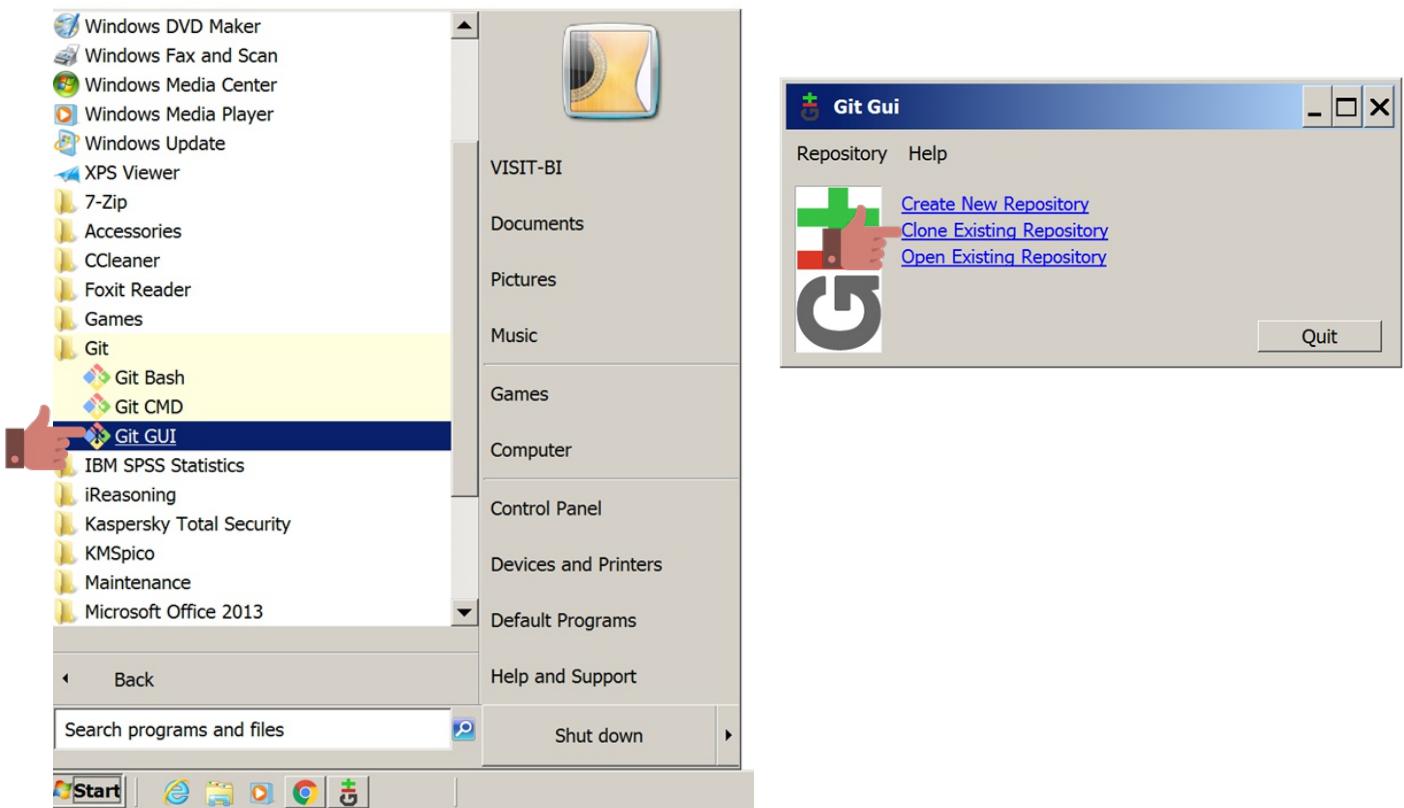
## ขั้นตอนที่ 4. ติดตั้ง git บน windows



## ขั้นตอนที่ 4. ติดตั้ง git บน windows



## ขั้นตอนที่ 5. เพิ่มบอร์ด ESP32 ด้วย Git GUI



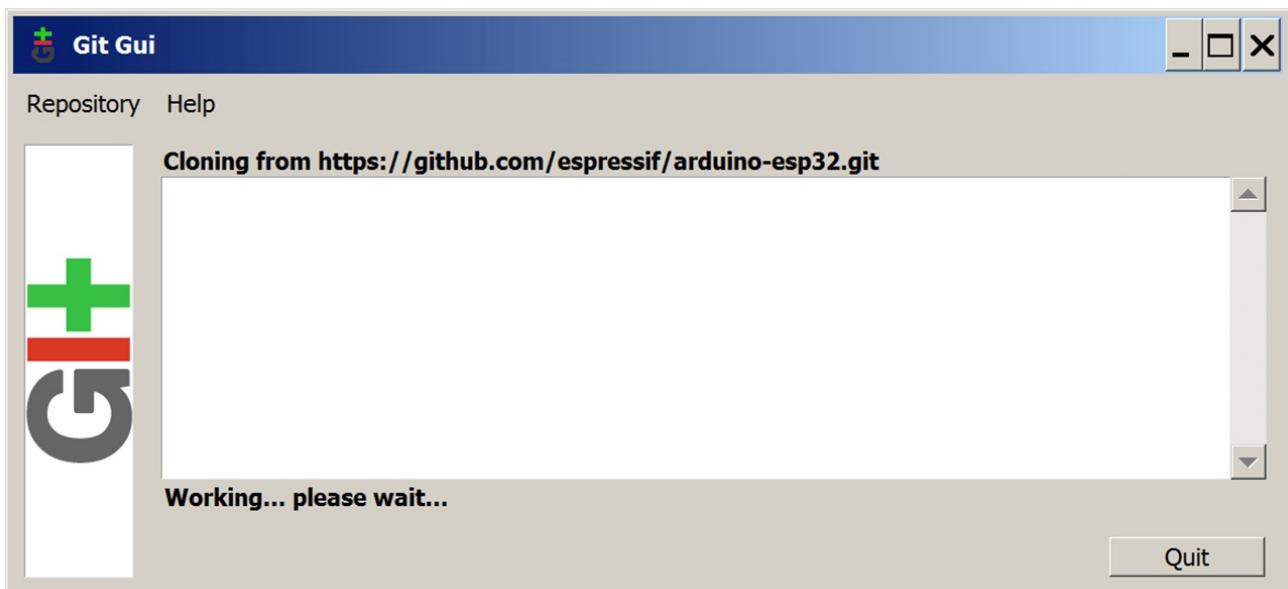
Source Location:

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

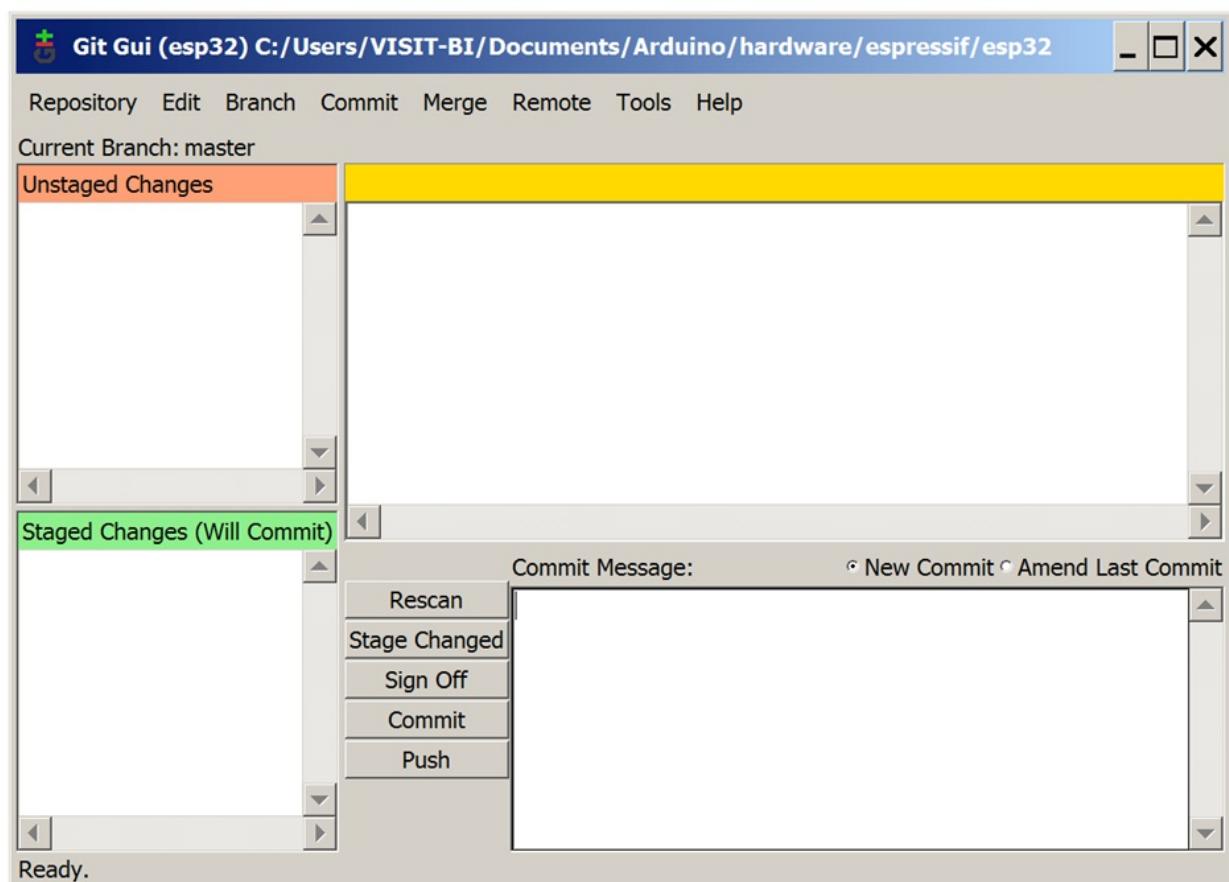
Target Directory:

C:/Users/YOUR\_USER\_NAME/Documents/Arduino/hardware/espressif/esp32

### ขั้นตอนที่ 5. เพิ่มบอร์ด ESP32 ด้วย Git GUI



รอสักครู่ Git กำลังดาวน์โหลดไฟล์



โหลดเสร็จแล้วให้ปิดหน้าต่างนี้

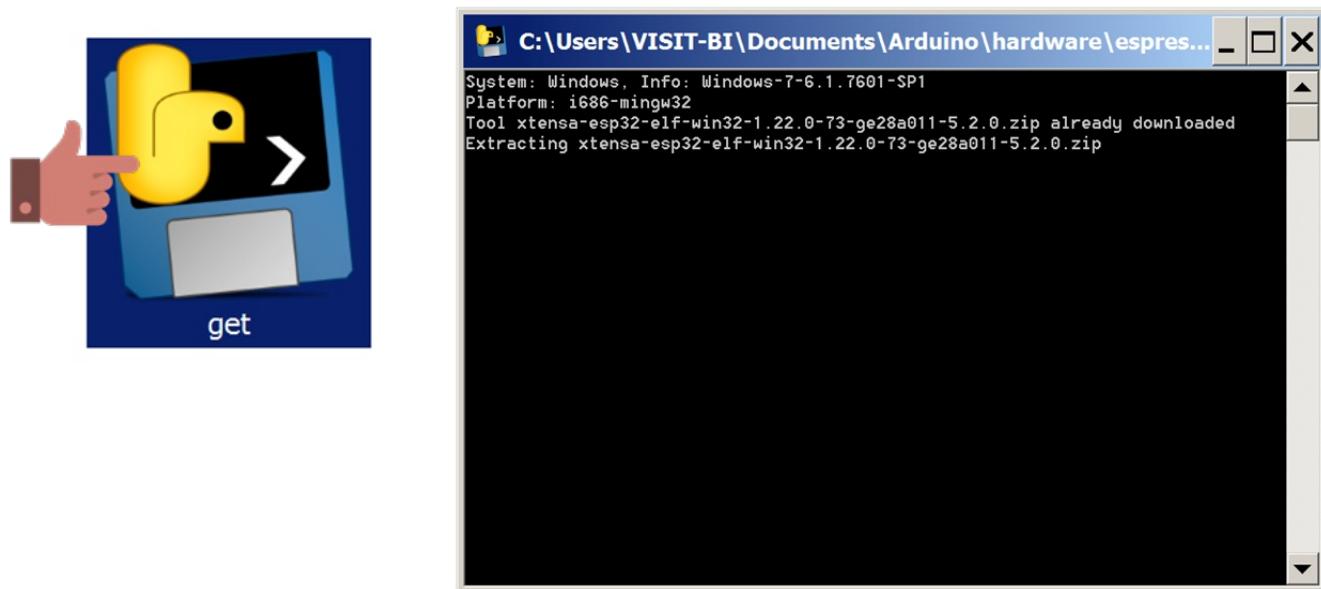
# การติดตั้ง ESP32 ใน Arduino IDE

## ขั้นตอนที่ 5. เพิ่มบอร์ด ESP32 ด้วย Git GUI

(1) เข้าไปที่

C:/Users/YOUR\_USER\_NAME/Documents/Arduino/hardware/espressif/esp32/tools

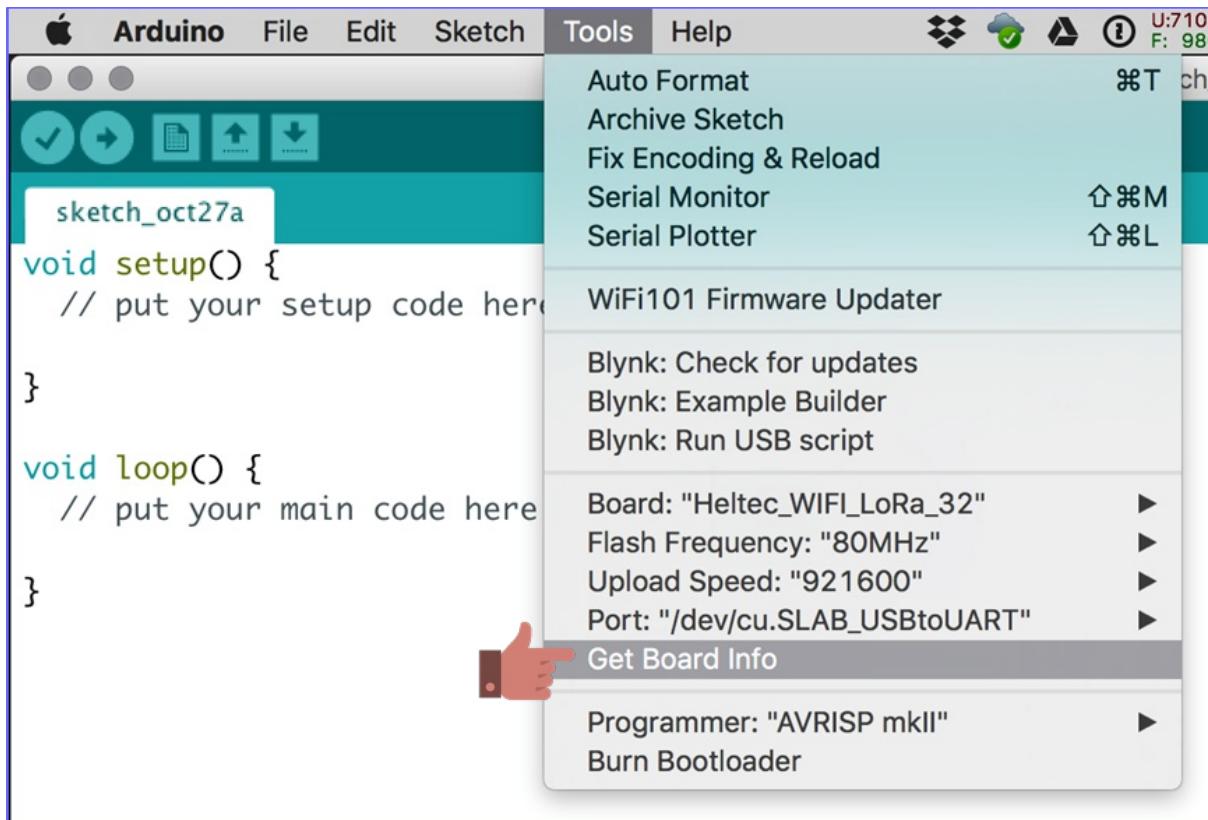
(2) แล้ว double-click get.exe จะแสดงหน้าต่างอ กมา ถ้าเสร็จแล้วหน้าต่างจะปิดเอง



หลังจากหน้าต่างปิดแล้วจะมีโฟล์เดอร์ xtensa-esp32-elf เพิ่มเข้ามา

Name	Date modified	Type	Size
dist	27-Oct-17 7:34 PM	File folder	
mkspiffs	27-Oct-17 7:34 PM	File folder	
partitions	27-Oct-17 7:31 PM	File folder	
sdk	27-Oct-17 7:31 PM	File folder	
xtensa-esp32-elf	27-Oct-17 7:33 PM	File folder	
build	27-Oct-17 7:31 PM	PY File	5 KB
common	27-Oct-17 7:31 PM	SH File	3 KB
espota	27-Oct-17 7:31 PM	Application	3,936 KB
espota	27-Oct-17 7:31 PM	PY File	10 KB
esptool	27-Oct-17 7:34 PM	Application	3,382 KB
esptool	27-Oct-17 7:31 PM	PY File	113 KB
gen_esp32part	27-Oct-17 7:31 PM	Application	3,260 KB
gen_esp32part	27-Oct-17 7:31 PM	PY File	15 KB
get	27-Oct-17 7:31 PM	Application	5,090 KB
get	27-Oct-17 7:31 PM	PY File	5 KB
platformio-build	27-Oct-17 7:31 PM	PY File	7 KB

# การเซ็ตค่าบอร์ด ESP32 ใน Arduino ide



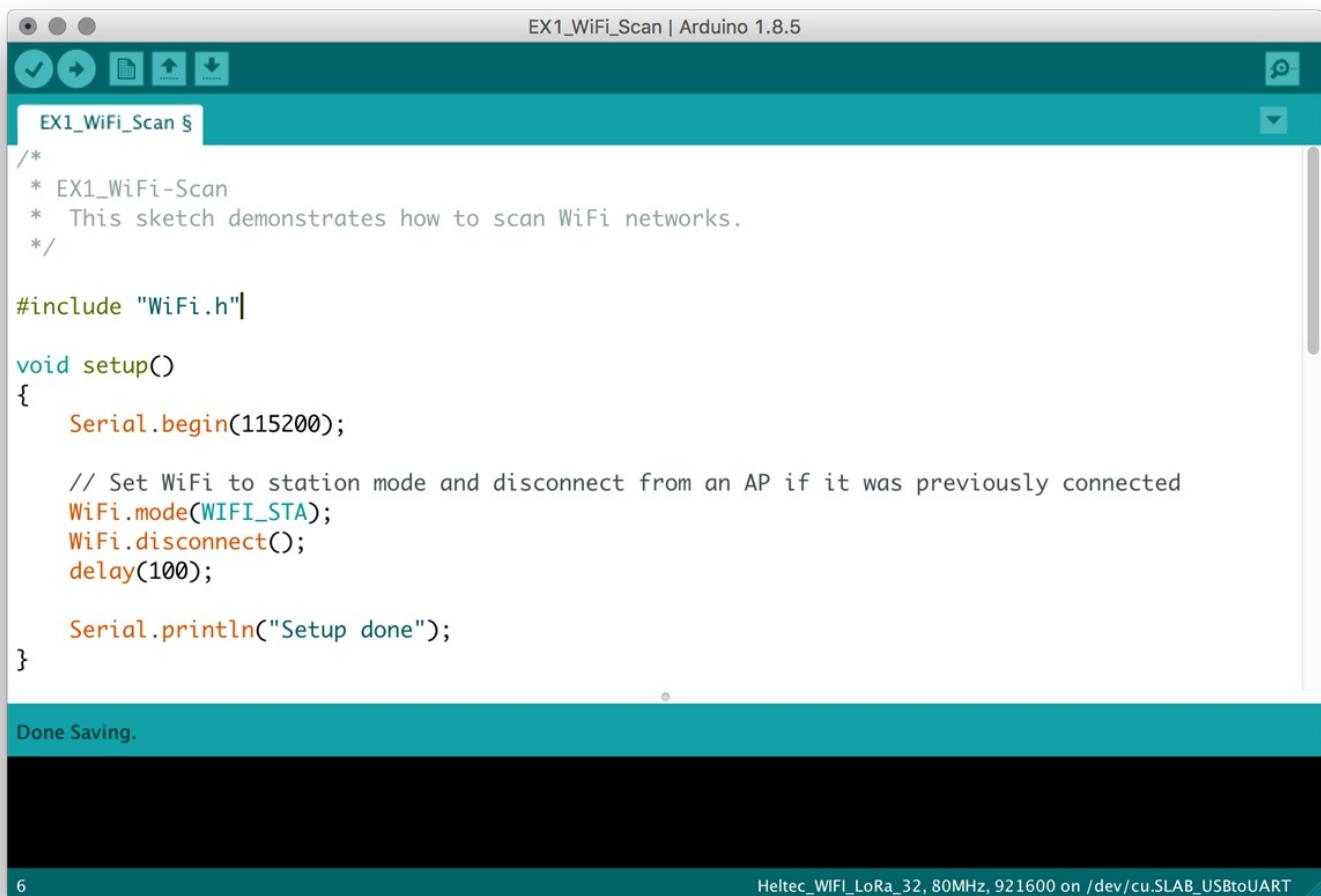
ทดสอบการเชื่อมต่อกับบอร์ด Heltec WiFi LoRa 32



# ทดสอบการอัพโหลด

## sketch

### EX1\_WiFi\_Scan\_ทดสอบการอัพโหลด sketch



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

- Title Bar:** EX1\_WiFi\_Scan | Arduino 1.8.5
- Toolbar:** Includes icons for file operations (New, Open, Save, Upload, Download) and a gear icon.
- Sketch Area:** Displays the code for the EX1\_WiFi\_Scan sketch. The code includes comments explaining the purpose of the sketch and its setup function, which initializes WiFi mode and prints "Setup done".
- Status Bar:** Shows the message "Done Saving." and the port information "Heltec\_WIFI\_LoRa\_32, 80MHz, 921600 on /dev/cu.SLAB\_USBtoUART".

### ทดสอบการอัพโหลด Sketch

## EX1\_WiFi\_Scan\_ทดสอบการอัพโหลด sketch

The screenshot shows the Arduino IDE interface with the title bar "EX1\_WiFi\_Scan | Arduino 1.8.5". The code editor contains the following text:

```
/*  
Done uploading.  
Writing at 0x00034000... (52 %)  
Writing at 0x00038000... (57 %)  
Writing at 0x0003c000... (63 %)  
Writing at 0x00040000... (68 %)  
Writing at 0x00044000... (73 %)  
Writing at 0x00048000... (78 %)  
Writing at 0x0004c000... (84 %)  
Writing at 0x00050000... (89 %)  
Writing at 0x00054000... (94 %)  
Writing at 0x00058000... (100 %)  
Wrote 463328 bytes (299602 compressed) at 0x00010000 in 5.9 seconds (effective 627.4 kbit/s)...  
Hash of data verified.  
Compressed 3072 bytes to 122...  
  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (122 compressed) at 0x00008000 in 0.0 seconds (effective 2594.0 kbit/s)...  
Hash of data verified.  
  
Leaving...  
Hard resetting...
```

The status bar at the bottom right shows "Heltec\_WIFI\_LoRa\_32, 80MHz, 921600 on /dev/cu.SLAB\_USBtoUART".

แสดงการอัพโหลดแล้วเสร็จสมบูรณ์

## ทดสอบการอัพโหลด sketch

### EX1\_WiFi\_Scan\_ทดสอบการอัพโหลด sketch

```
scan done
1 networks found
1: plutter (-80)*

scan start
scan done
2 networks found
1: plutter (-80)*
2: 0AA (-93)*

scan start
scan done
1 networks found
1: plutter (-80)*

scan start
```

Autoscroll      Both NL & CR      115200 baud      Clear output

แสดงผลลัพธ์ในคอนโซล



# บทที่ 2 การทำงานแบบมัลติทาสกิ้ง

Multitasking on ESP32

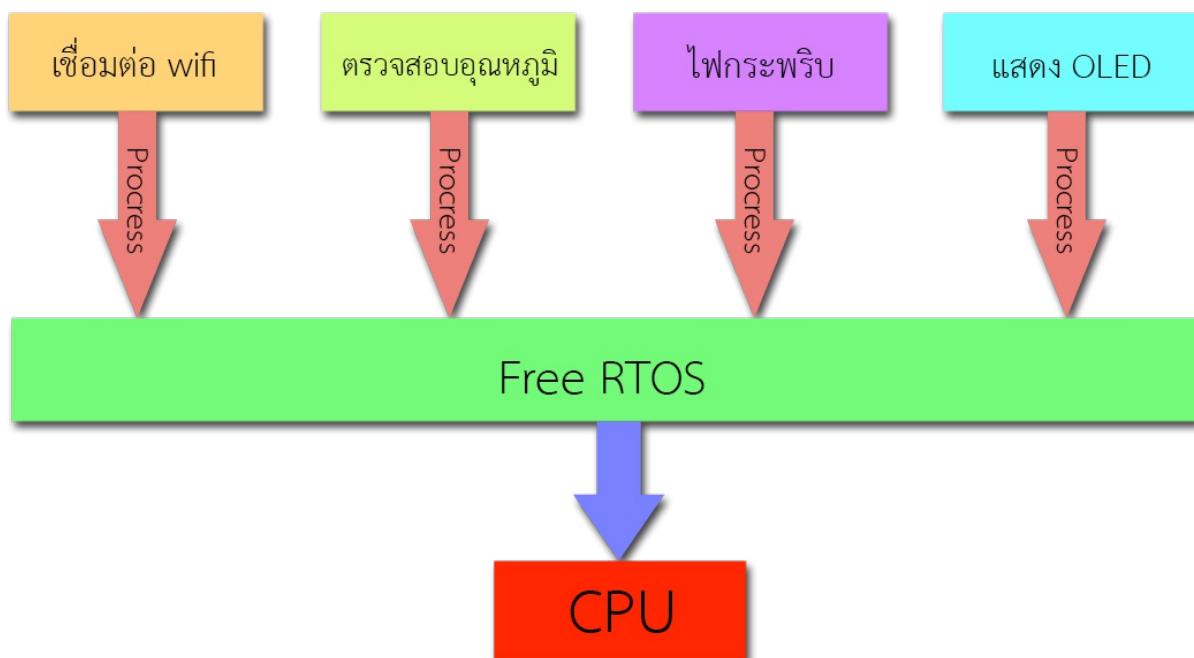
# หัวข้อ

## การทำงานของ ESP32

- การทำงานแบบมัลติทาสกิ้ง
- โหมดประทัยดพลังงาน
- หน่วยความจำและพื้นที่เก็บข้อมูล

## มัลติทาสกิ้ง (multitasking) ใน ESP32 คือ

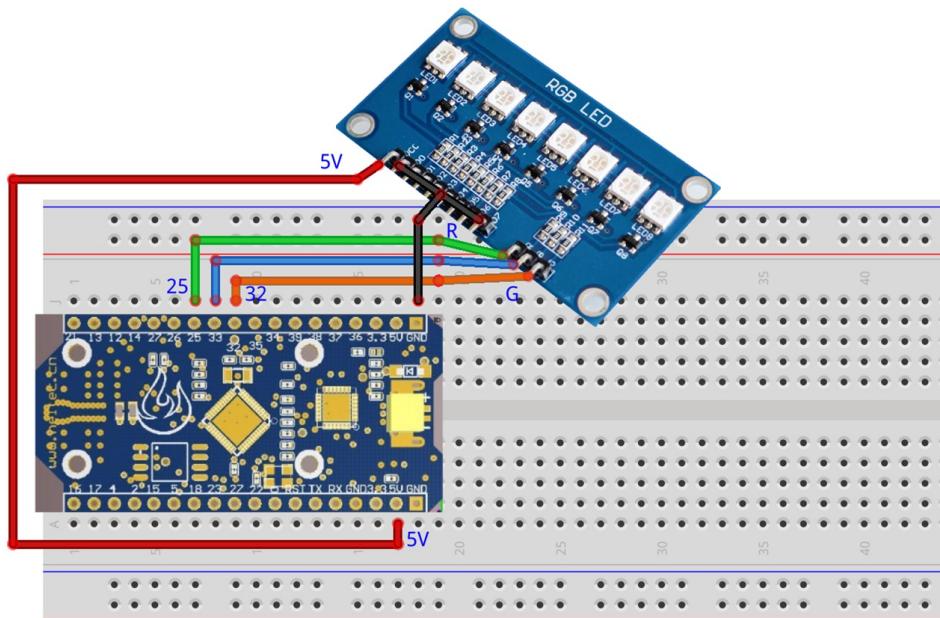
การทำงานหลายส่วนขนานกันไปอาทิเช่นการตรวจสอบสถานะของแสงเพื่อเปิดปิดไฟในขณะที่ต้องตรวจสอบสถานะการกดปุ่มสวิตซ์ไปด้วยในเวลาเดียวกัน หรือการทำไฟกระพริบเป็นจังหวะเพื่อแสดงให้เห็นว่าอุปกรณ์กำลังทำงานในขณะที่ต้องอยู่วนตรวจสอบข้อมูลที่มาจากเซนเซอร์วัดอุณหภูมิความชื้น เป็นต้น



## EX2\_MultiTasking\_Milli\_LED

ตัวอย่าง แสดงการควบคุมหลอดไฟ LED กระพริบคนละจังหวะ หลอดไฟ LED หลอดแรกจะกระพริบทุก 0.7 วินาทีหลอดที่สองจะกระพริบทุก 0.3 วินาที และหลอดที่สามจะกระพริบทุก 0.1 วินาที

ฟังก์ชัน `milli()` ส่งค่ากลับมาเป็น เวลาเป็นมิลลิวินาทีนับตั้งแต่บอร์ด Arduino เริ่มใช้งานโปรแกรมปัจจุบัน ตัวเลขที่ส่งกลับมาจะเกิดการ overflow (กลับค่าไปเป็น 0) หลังจากเริ่มนับประมาณ 50 วัน



```
EX2_MultiTasking_Milli_LED | Arduino 1.8.5

/*
 * EX2_MultiTasking_Milli_LED
 * ควบคุมหลอดไฟ LED กระพริบคนละจังหวะ
 */

//-----ตั้งค่าที่นี่-----///

#define LED1 32
#define LED2 33
#define LED3 25 //ไฟ LED บนบอร์ด

//-----///

unsigned long last1, last2, last3;

void setup() {
  pinMode(LED1, OUTPUT);
  pinMode(LED2, OUTPUT);
  pinMode(LED3, OUTPUT);
}

Done uploading.
Hash of data verified.

Leaving...
Hard resetting...

10 - 8
```

## MultiTasking FreeRTOS

เป็นการทำงานแบบหลายกระบวนการในเวลาเดียวกัน และยังสามารถกำหนดระดับความสำคัญ (Priority) ได้ ความสามารถนี้ ESP32 จะใช้ระบบ FreeRTOS เข้ามาช่วยจัดการ โดยทำการแบ่งแยกคอร์ของ MCU ควบคุมการทำงานของแต่ละ task เรียกว่าการทำงานในลักษณะนี้ว่ามัลติทาสกิ้ง โดยแต่ละ Task จะแยกการทำงานออกจากกันอย่างอิสระ

### ฟังก์ชัน xTaskCreate() สำหรับการสร้าง task

```
/* Create the task, storing the handle. */
xReturned = xTaskCreate(
    vTaskCode,          /* Function that implements the task. */
    "NAME",            /* Text name for the task. */
    STACK_SIZE,         /* Stack size in words, not bytes. */
    ( void * ) 1,       /* Parameter passed into the task. */
    tskIDLE_PRIORITY,  /* Priority at which the task is created. */
    &xHandle );        /* Used to pass out the created task's handle. */
```

## EX3\_MultiTasking\_FreeRTOS\_DHT22

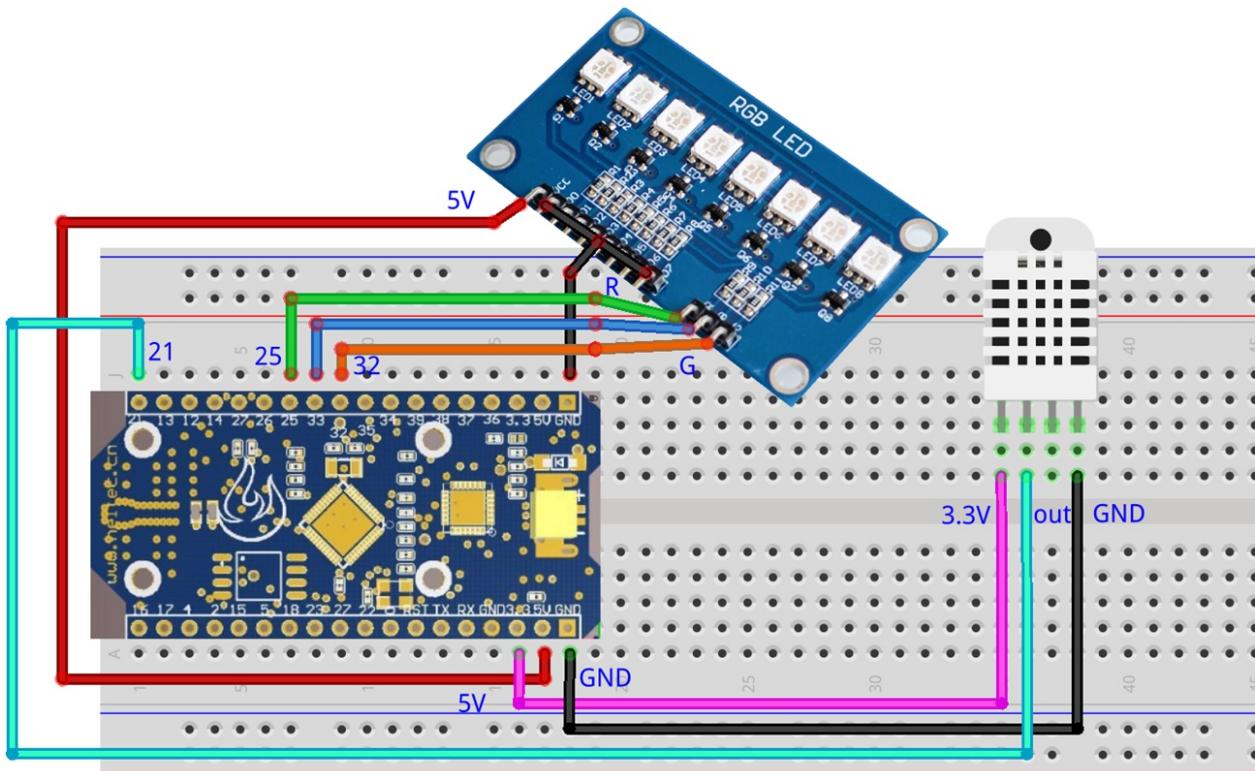
การทำงานแบบหลายกระบวนการในเวลาเดียวกัน

Task1 จะควบคุมหลอดไฟให้กระพริบทุก 0.7 วินาที

Task2 จะควบคุมหลอดไฟให้กระพริบทุก 0.3 วินาที

Task3 จะควบคุมหลอดไฟให้กระพริบทุก 0.1 วินาที

Task4 จะอ่านค่าอุณหภูมิและความชื้นจากเซ็นเซอร์ DHT22 มาแสดงผลในคอนโซล



## การทำงานแบบมัลติทาสกิ้ง EX3\_MultiTasking\_FreeRTOS\_DHT22

EX3\_MultiTasking\_FreeRTOS\_DHT22 | Arduino 1.8.5

```
EX3_MultiTasking_FreeRTOS_DHT22

/*
 * EX3_MultiTasking_FreeRTOS_DHT22
 * การแบ่ง task ทำงาน โดยควบคุมหลอดไฟ LED กระพริบคันละจังหวะ
 * และอ่านค่าอุณหภูมิและความชื้นจากเซ็นเซอร์ DHT22 มาแสดงผลในคอนโซล
 */

//-----ตั้งค่าที่นี่-----///

#define LED1 32
#define LED2 33
#define LED3 25 //build-in LED on board
#define DHTTYPE dht11
#define DHTPIN 21

//-----///

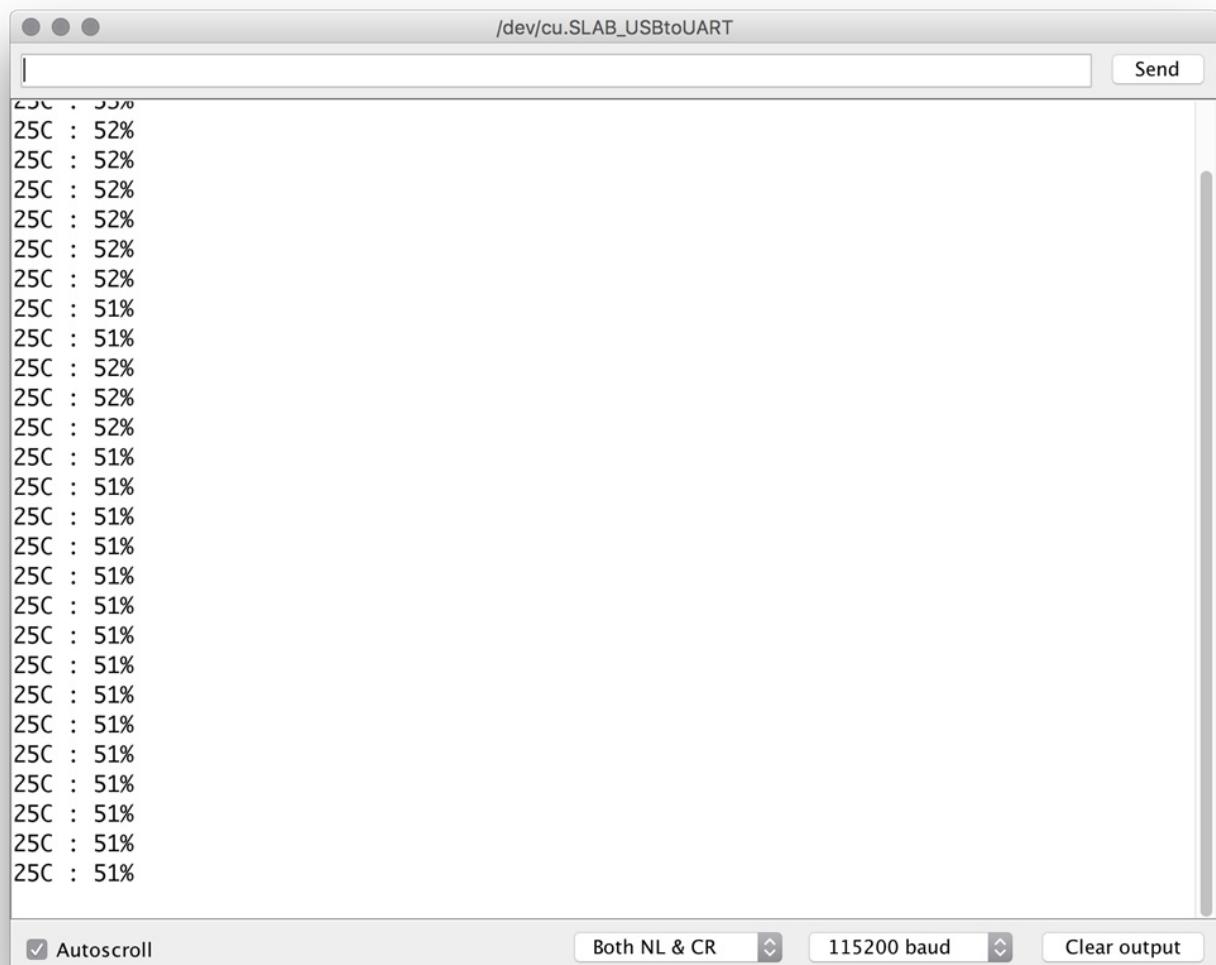
#include <SimpleDHT.h>          //โหลดไลบารี DHT11
SimpleDHT11 DHTTYPE;           //ระบุรุ่นเซ็นเซอร์ DHT11
byte temperature = 0;           //กำหนดตัวแปรเก็บค่าอุณหภูมิ

Done uploading.

Hash of data verified.

Leaving...
Hard resetting...
```

การทำงานแบบมัลติทาสกิ้ง  
EX3\_MultiTasking\_FreeRTOS\_DHT22



## EX4\_MultiTasking\_FreeRTOS\_OLED\_DHT22

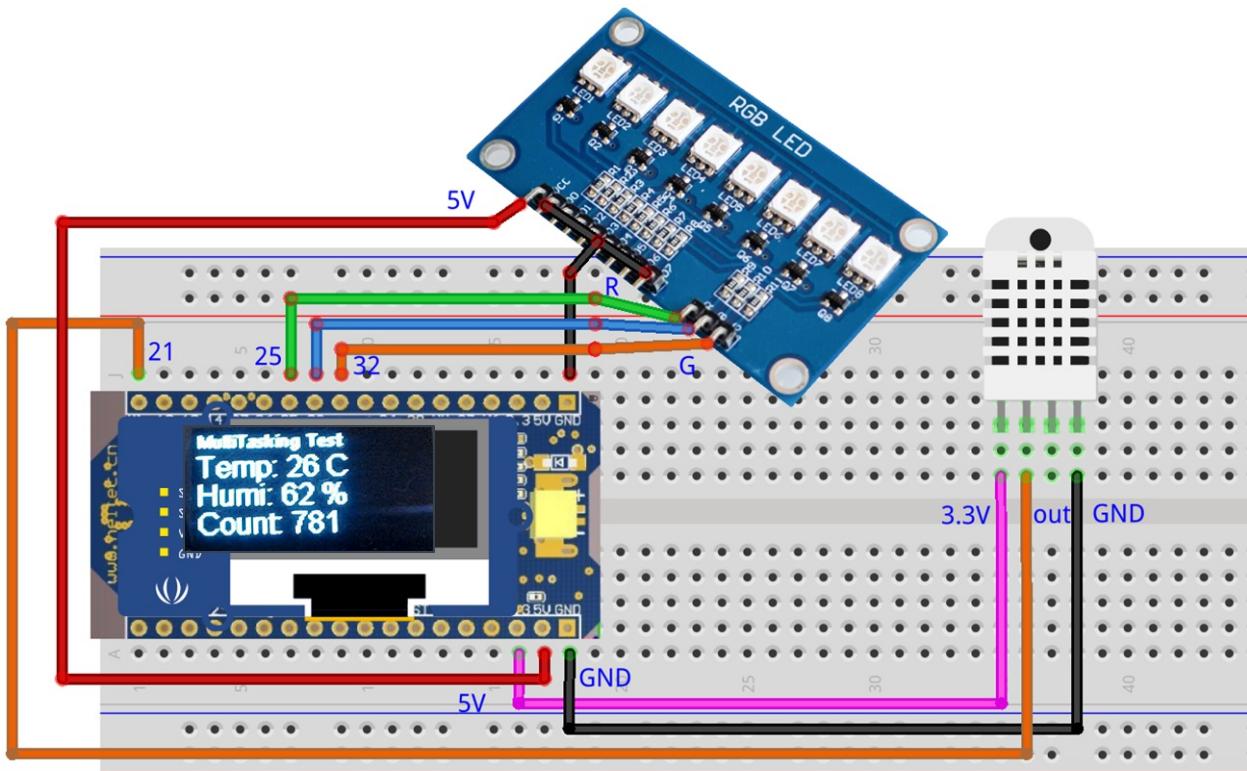
การทำงานแบบหลายกระบวนการในเวลาเดียวกัน

Task1 จะควบคุมหลอดไฟให้กระพริบทุก 0.7 วินาที

Task2 จะควบคุมหลอดไฟให้กระพริบทุก 0.3 วินาที

Task3 จะควบคุมหลอดไฟให้พริบทุก 0.1 วินาที

Task4 จะอ่านค่าอุณหภูมิและความชื้นจากเซ็นเซอร์ DHT22 มาแสดงผลในคอนโซล



## การทำงานแบบมัลติทาสกิ้ง EX4\_MultiTasking\_FreeRTOS\_OLED\_

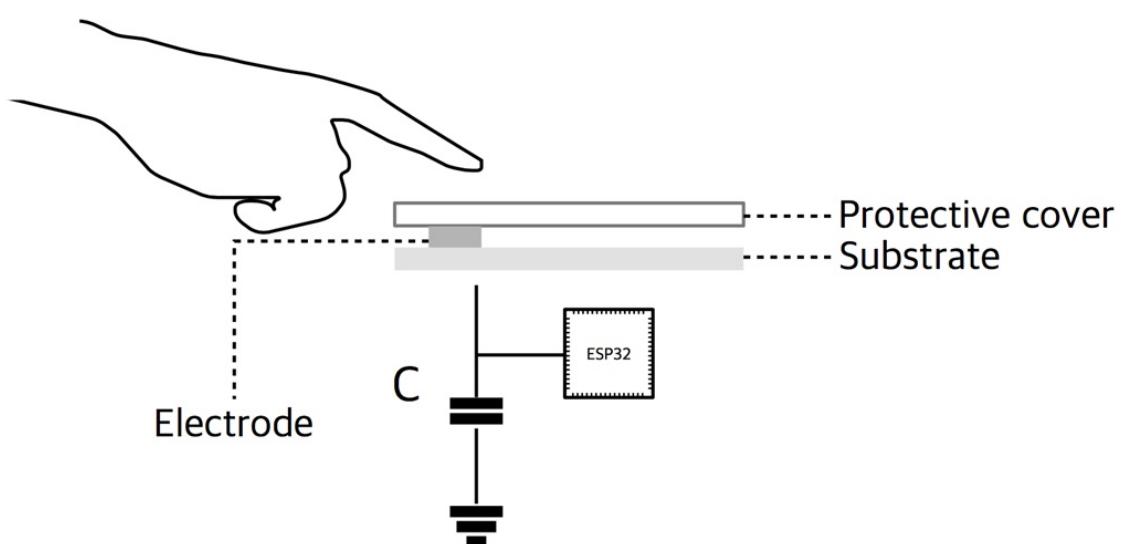
```
/*  
 * EX4_MultiTasking_FreeRTOS_DHT22_OLED  
 * การทำงานโดยควบคุมหลอดไฟ LED กระพริบคนละจังหวะ  
 * อ่านค่าอุณหภูมิและความชื้นจากเซนเซอร์ DHT22 มาแสดงผลในคอนโซล  
 * และแสดงค่าในจอ OLED  
 */  
  
///-----ตั้งค่าที่นี่-----///  
  
#define LED1 32  
#define LED2 33  
#define LED3 25 //build-in LED on board  
#define DHTTYPE dht11  
#define DHTPIN 21  
  
///-----///  
  
#include <Wire.h>  
#include "SSD1306.h"  
SSD1306 display(0x3c, 4, 15);  
  
Done Saving.  
Leaving...  
Hard resetting...  
  
14 - 10          Heltec_WIFI_LoRa_32, 80MHz, 921600 on /dev/cu.SLAB_USBtoUART
```

# ເຫັນເຊວ່າສົມຜັສ

Touch Sensor on ESP32

# Touch sensor

A touch sensor system is built on a substrate which carries electrodes and relevant connections with a flat protective surface. When a user touches the surface, the capacitance variation is triggered, and a binary signal is generated to indicate whether the touch is valid.



## EX5\_TouchSensor\_Read

ทดสอบการอ่านค่าเซ็นเซอร์สัมผัสที่มากับ ESP32 และแสดงผลในคอนโซล ข้าที่ต่อทดสอบ คือ

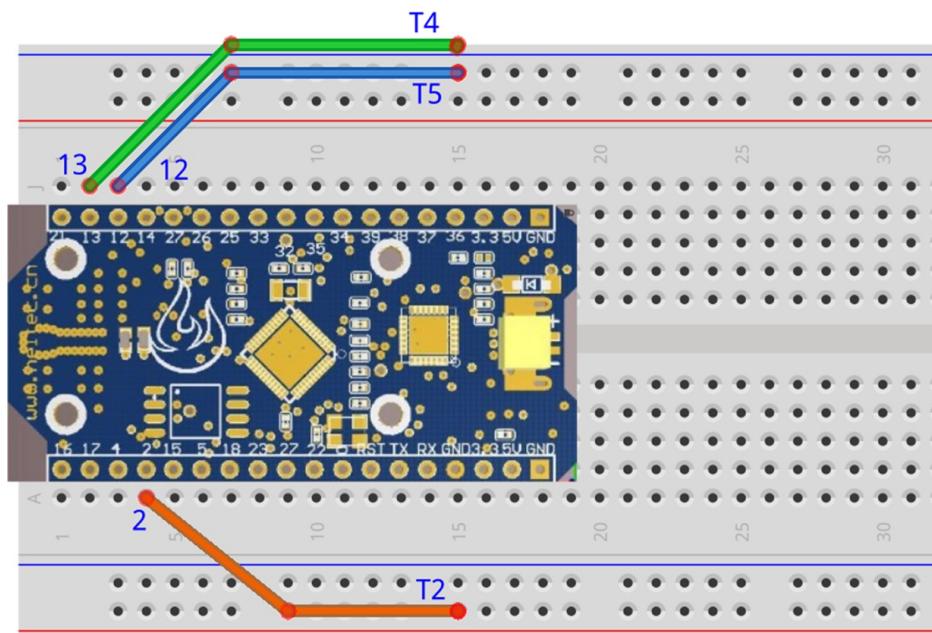
TOUCH4 (GPIO13)

TOUCH5 (GPIO12)

## TOUCH2 (GPIO2)

## ฟังก์ชันใช้งานที่เกี่ยวข้อง

```
// ถ้า T4 มีค่าเกิน 40 ให้ไปที่ฟังก์ชัน gotTouch1() ที่สร้างไว้  
touchAttachInterrupt(T4, gotTouch1, 40);  
  
esp_deep_sleep_enable_touchpad_wakeup(); // กำหนดการตื่นด้วยเซ็นเซอร์สัมผัส  
esp_deep_sleep_start(); // เข้าสู่โหมด Deep Sleep  
Serial.println(touchRead(T4)); // อ่านค่า sensor T4 และแสดงผลใน
```



## EX5\_TouchSensor\_Read

```
/*  
 * EX5_TouchSensor_Read  
 * รอการเปลี่ยนสถานะของเซ็นเซอร์สัมผัส T4 T5 T2 และแสดงผลในコンโซล  
 */  
  
void gotTouch1(){  
    Serial.print("Touch sensor T4 detected : ");  
    Serial.println(touchRead(T4));  
}  
  
void gotTouch2(){  
    Serial.print("Touch sensor T5 detected : ");  
    Serial.println(touchRead(T5));  
}  
  
void gotTouch3(){  
    Serial.print("Touch sensor T2 detected : ");  
    Serial.println(touchRead(T2));  
}  
  
void setup() {  
    Serial.begin(115200);  
  
    Serial.println("ESP32 Touch Interrupt Test");  
}
```

Done Saving.

```
| /dev/cu.SLAB_USBtoUART Send  
Touch sensor T4 detected : 51  
Touch sensor T4 detected : 13  
Touch sensor T4 detected : 12  
Touch sensor T4 detected : 12  
Touch sensor T4 detected : 12  
Touch sensor T4 detected : 15  
Touch sensor T2 detected : 38  
Touch sensor T4 detected : 75  
Touch sensor T2 detected : 13  
Touch sensor T2 detected : 13  
Touch sensor T2 detected : 42  
Touch sensor T2 detected : 13  
Touch sensor T2 detected : 16  
Touch sensor T2 detected : 13  
Touch sensor T2 detected : 13  
Touch sensor T2 detected : 69  
Touch sensor T4 detected : 13  
Touch sensor T4 detected : 13  
Touch sensor T4 detected : 12  
Touch sensor T4 detected : 14  
Touch sensor T4 detected : 75  
Touch sensor T5 detected : 30  
Touch sensor T5 detected : 18  
Touch sensor T5 detected : 19  
Touch sensor T5 detected : 19  
Touch sensor T5 detected : 32
```

# ໂທນົດປະຫວັດພລັງງານ

Deep Sleep on ESP32

Table: Functionalities Depending on the Power Modes

Power mode	Active	Modem-sleep	Light-sleep	Deep-sleep	Hibernation
Sleep pattern		Association sleep pattern		ULP sensor-monitored pattern	-
CPU	ON	PAUSE	ON	OFF	OFF
Wi-Fi/BT base-band and radio	ON	OFF	OFF	OFF	OFF
RTC	ON	ON	ON	ON	OFF
ULP co-processor	ON	ON	ON	ON/OFF	OFF

## Sleep Pattern

- Associationsleepattern: The power mode switches between the active mode and Modem-sleep/Light- sleep mode during this sleep pattern. The CPU, Wi-Fi, Bluetooth, and radio are woken up at predetermined intervals to keep Wi-Fi/BT connections alive.
- ULP sensor-monitored pattern: The main CPU is in the Deep-sleep mode. The ULP co-processor does sensor measurements and wakes up the main system, based on the measured data from sensors.

## Power mode

- Active mode: The chip radio is powered on. The chip can receive, transmit, or listen.
- Modem-sleep mode: The CPU is operational and the clock is configurable. The Wi-Fi/Bluetooth base- band and radio are disabled.
- Light-sleep mode: The CPU is paused. The RTC and ULP-coprocessor are running. Any wake-up events (MAC, host, RTC timer, or external interrupts) will wake up the chip.
- Deep-sleep mode: Only RTC is powered on. Wi-Fi and Bluetooth connection data are stored in RTC memory. The ULP-coprocessor can work.
- Hibernation mode: The internal 8MHz oscillator and ULP-coprocessor are disabled. The RTC recovery memory are power-down. Only one RTC timer on the slow clock and some RTC GPIOs are active. The RTC timer or the RTC GPIOs can wake up the chip from the Hibernation mode.

Table: Power Consumption by Power Modes

Power mode	Description	Power consumption
Active (RF working)	Wi-Fi Tx packet 13 dBm ~ 21 dBm	160 ~ 260 mA
	Wi-Fi / BT Tx packet 0 dBm	120 mA
	Wi-Fi / BT Rx and listening	80 ~ 90 mA
	Association sleep pattern (by Light-sleep)	0.9 mA@DTIM3, 1.2 mA@DTIM1
Modem-sleep	The CPU is powered on.	Max speed: 20 mA
		Normal speed: 5 ~ 10 mA
		Slow speed: 3 mA
Light-sleep	-	0.8 mA
Deep-sleep	The ULP co-processor is powered on.	0.5 mA
	ULP sensor-monitored pattern	25 $\mu$ A @1% duty
	RTC timer + RTC memory	10 $\mu$ A
Hibernation	RTC timer only	2.5 $\mu$ A

Table: RF Power Consumption Specifications

Mode	Min	Typ	Max	Unit
Transmit 802.11b, DSSS 1 Mbps, POUT = +19.5 dBm	-	225	-	mA
Transmit 802.11b, CCK 11 Mbps, POUT = +18.5 dBm	-	205	-	mA
Transmit 802.11g, OFDM 54 Mbps, POUT = +16 dBm	-	160	-	mA
Transmit 802.11n, MCS7, POUT = +14 dBm	-	152	-	mA
Receive 802.11b, packet length = 1024 bytes, -80 dBm	-	85	-	mA
Receive 802.11g, packet length = 1024 bytes, -70 dBm	-	85	-	mA
Receive 802.11n, packet length = 1024 bytes, -65 dBm	-	80	-	mA
Receive 802.11n HT40, packet length = 1024 bytes, -65 dBm	-	80	-	mA

The current consumption measurements are conducted with 3.0 V supply and 25°C ambient, at antenna port. All the transmitters' measurements are based on 90% duty cycle and continuous transmit mode.

## ความหมายของหน่วยทางไฟฟ้า

mV มิลลิโวลต์ คือหน่วยที่ใช้เรียกขนาดของแรงดันไฟฟ้าหนึ่งในพันของโวลต์ เช่น 1mV หมายถึงขนาดของแรงดันไฟฟ้า 1 มิลลิโวลต์ เท่ากับ 0.001 โวลต์ ขนาดของแรงดันไฟฟ้า 1000mV มีค่าเท่ากับ 1 โวลต์

V (Volt) โวลต์ คือหน่วยที่ใช้เรียกขนาดของแรงดันไฟฟ้า เช่น 220V หมายถึงขนาดของแรงดันไฟฟ้า 220 โวลต์

mA มิลลิแอมป์ คือหน่วยที่ใช้เรียกปริมาณของกระแสไฟฟ้าที่ไหลผ่านหนึ่งในพันของแอมป์ เช่น 1mA หมายถึงปริมาณของกระแสไฟฟ้าที่ไหลผ่านมีขนาด 1 มิลลิแอมป์ เท่ากับ 0.001 แอมป์ ปริมาณของกระแสไฟฟ้าที่ไหลผ่าน 1000mA มีค่าเท่ากับ 1 แอมป์

A (Ampere) แอมป์ หรือ แอมเปร คือหน่วยที่ใช้เรียกปริมาณของกระแสไฟฟ้าที่ไหลผ่าน เช่น 10A หมายถึงปริมาณกระแสไฟฟ้าที่ไหลผ่าน 10 แอมป์

W วัตต์ คือหน่วยที่ใช้เรียกขนาดของกำลังไฟฟ้า เช่น 1W หมายถึงใช้กำลังไฟฟ้า 1 วัตต์

## ตารางแปลง มิลลิวินาที เป็น วินาที

1 มิลลิวินาที = 0.001 วินาที	10 มิลลิวินาที = 0.01 วินาที	2500 มิลลิวินาที = 2.5 วินาที
2 มิลลิวินาที = 0.002 วินาที	20 มิลลิวินาที = 0.02 วินาที	5000 มิลลิวินาที = 5 วินาที
3 มิลลิวินาที = 0.003 วินาที	30 มิลลิวินาที = 0.03 วินาที	10000 มิลลิวินาที = 10 วินาที
4 มิลลิวินาที = 0.004 วินาที	40 มิลลิวินาที = 0.04 วินาที	25000 มิลลิวินาที = 25 วินาที
5 มิลลิวินาที = 0.005 วินาที	50 มิลลิวินาที = 0.05 วินาที	50000 มิลลิวินาที = 50 วินาที
6 มิลลิวินาที = 0.006 วินาที	100 มิลลิวินาที = 0.1 วินาที	100000 มิลลิวินาที = 100 วินาที
7 มิลลิวินาที = 0.007 วินาที	250 มิลลิวินาที = 0.25 วินาที	250000 มิลลิวินาที = 250 วินาที
8 มิลลิวินาที = 0.008 วินาที	500 มิลลิวินาที = 0.5 วินาที	500000 มิลลิวินาที = 500 วินาที
9 มิลลิวินาที = 0.009 วินาที	1000 มิลลิวินาที = 1 วินาที	1000000 มิลลิวินาที = 1000 วินาที

## ກາຣອອກຈາກໂທມດ deep sleep ຂອງ ESP8266 ແລະ ESP32

### ESP8266 Wake-Up Source:

- Reset the chip from an external source.
- Connect GPIO16 to the reset pin for a timed wake-up.
- With the timer, we were only able to sleep for about one hour.

### The ESP32 has four different wake-up sources:

- The touch pads.
- The timer.
- Two possibilities of external Wake-up.
- ULP coprocessor wake-up.

## EX6\_DeepSleep\_Touchpad\_Wakeup\_DHT22\_OLED

ກຳນົດໃຫ້ ESP32 ອູ່ໃນໂທມດ deep sleep ເພື່ອປະຫຍັດພລັງງານແລະອອກຈາກໂທມດ deep sleep ດ້ວຍເຫັນເຂົ້າສຳເນົາ

ຄ້າມີການສັນຜັກເຫັນເຂົ້າສຳເນົາ T4 ທີ່ໄດ້ຮັບມາໃຫ້ມາແລ້ວລັບມາກຳນົດໃຫ້ມາໃໝ່  
ຈະໄປອ່ານຄ່າຂໍ້ມູນອຸນຫຼມມີຄວາມເຊື້ນ DHT22 ມາແສດງຜລໃນຈອ OLED ແລະ Console ເປັນເວລາ 15 ວິນາທີ  
ຈາກນັ້ນກີ່ຈະເຂົ້າສູ້ໂທມດ deep sleep ເພື່ອປະຫຍັດພລັງງານຕ່ອງໄປ ມີການເກີບຂໍ້ມູນຈຳນວນທີ່ບູຕໄວ້ໃນໜ່ວຍ  
ຄວາມຈຳຂອງ RTC ດ້ວຍ

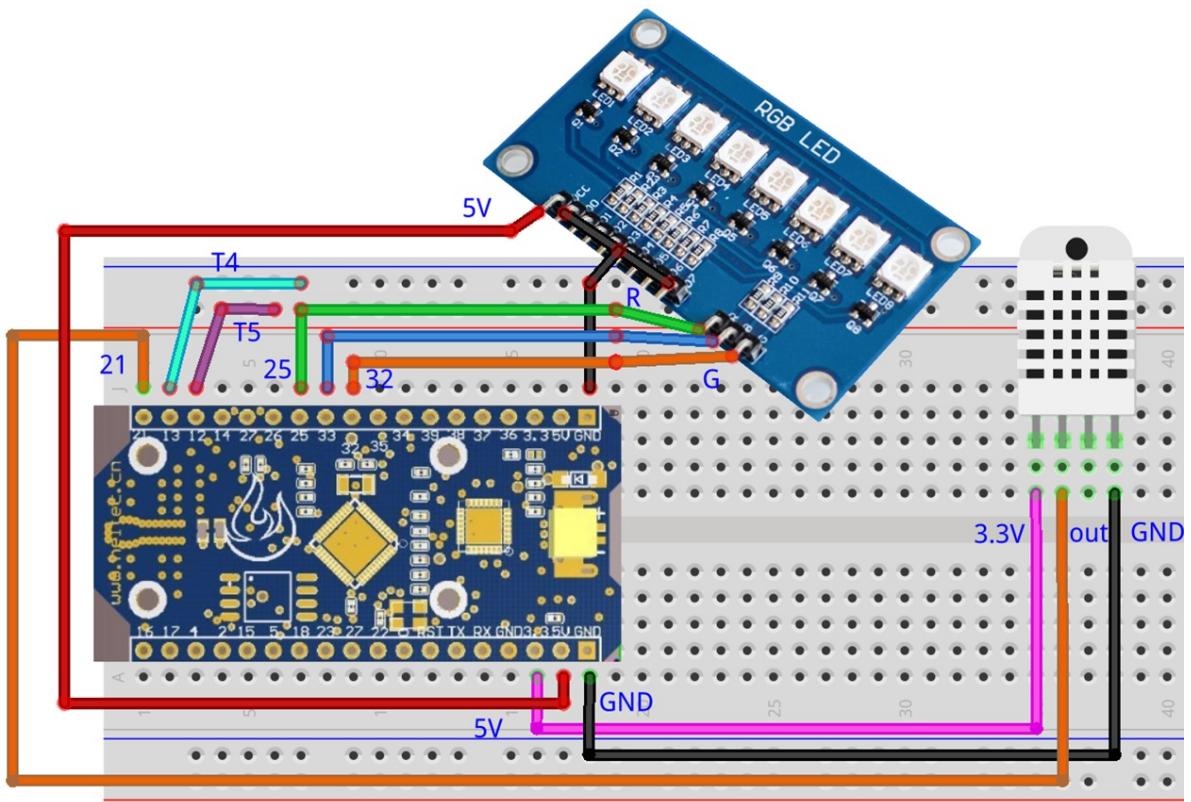
### ຝຶກ໌ໜີໃຊ້ງານທີ່ເກີຍວ່າຂອງ

```
touchAttachInterrupt(T4, gotTouch1, 40);  
// ຄ້າ T4 ມີຄ່າເກີນ 40 ໃຫ້ໄປທີ່ຝຶກ໌ໜີ gotTouch1() ທີ່ສ້າງໄວ້
```

```
esp_deep_sleep_enable_touchpad_wakeup(); // ກຳນົດກາຣຕື່ນດ້ວຍເຫັນເຂົ້າສຳເນົາ  
esp_deep_sleep_start(); // ເຂົ້າສູ້ໂທມດ Deep Sleep  
Serial.println(touchRead(T4)); // ອ່ານຄ່າ sensor T4 ແລ້ວແສດງຜລໃນ console  
RTC_DATA_ATTR int bootCount = 0; // ເກີບຈຳນວນກາຣບູຕແຕ່ລະຄົ້ງ
```

# ໄທມດປະຫຍດພລັງງານ

## EX6\_DeepSleep\_Touchpad\_Wakeup\_DHT22\_OLED



```
EX6_DeepSleep_Touchpad_Wakeup_DHT22_OLED | Arduino 1.8.5

/*
 * EX6_DeepSleep_Touchpad_Wakeup_DHT22_OLED
 * ອໍານວຍມີແລກວິເຄາະຂຶ້ນຈາກເຊື່ອຮຸ່ນ DHT22 ມາສະດົງຜລໃນຄອນໂຮລ
 * ແລະແສດງຄາໃນຈອ OLED ແລ້ວ deep sleep
 */

-----ຕົງຕ່າທີ່ນີ້-----///

#define LED1 32
#define LED2 33
#define LED3 25      //build-in LED on board
#define DHTTYPE dht11 //ຮະບູຮຸ່ນເຊື່ອຮຸ່ນ DHT11
#define DHTPIN 21

-----///

#include "esp_deep_sleep.h"
#include <Wire.h>
#include "SSD1306.h"
SSD1306 display(0x3c, 4, 15);

#include <SimpleDHT.h>           //ໂຫດໄລບາຣີ DHT11
SimpleDHT11 DHTTYPE;            //ຮະບູຮຸ່ນເຊື່ອຮຸ່ນ DHT11
byte temperature = 0;           //ກຳນົດດ້ວຍແປຣເກີນຄ່າອຸ່ນກຸມ
byte humidity = 0;              //ກຳນົດດ້ວຍແປຣເກີນຄ່າວິເຄາະຂຶ້ນສັນລັກ
```

## ໂທມດປະຫຍດພລັງງານ

### EX6\_DeepSleep\_Touchpad\_Wakeup\_DHT22\_OLED

The screenshot shows a terminal window titled "/dev/cu.SLAB\_USBtoUART" displaying serial port output. The output includes sensor readings (Temperature and Humidity) and a deep sleep boot sequence.

```
Temperature : 28 C Humidity : 64 %
Sleep
ets Jun  8 2016 00:22:57

rst:0x5 (DEEPSLEEP_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:812
load:0x40078000,len:0
load:0x40078000,len:10212
entry 0x40078a00
Boot number: 3
Temperature : 28 C Humidity : 63 %
Temperature : 28 C Humidity : 64 %
Temperature : 28 C Humidity : 63 %
Sleep
```

At the bottom of the window, there are several configuration buttons: "Autoscroll" (checked), "Both NL & CR", "115200 baud", and "Clear output".

## EX7\_DeepSleep\_Touchpad\_Wakeup\_DHT22\_OLED

ກຳທັນດີໃຫ້ ESP32 ອູ່ໃນໂທມດ deep sleep ເພື່ອປະຫຍັດພລັງງານ ແລ້ວກຳທັນດໍາ GPIO2 ເປັນຕົວຮະຕຸນໃຫ້ອອກຈາກໂທມດ deep sleep ຄໍາມີການເປີ່ຍືນສະຖານະເປັນ HIGH

ກາຣທົດສອບນີ້ຈະໃໝ່ໄຟ 3.3v ຈ່າຍເຂົ້າໄປໂດຍໃໝ່ switch ຂັ້ນໄວ້ ຄໍາມີກາຣກົດ switch ຈະມີກາຣຈ່າຍໄຟເຂົ້າໄປທີ່ຂາ GPIO2 ຜຶ້ງຈະທຳໃຫ້ສະຖານະເປີ່ຍືນເປັນ HIGH ແລ້ວອອກຈາກໂທມດ deep sleep

ESP32 ຈະໄປອ່ານຄ່າຂໍ້ມູນອຸນຫຼວມຄວາມชິ້ນ DHT22 ມາແສດງຜລໃນຈອ OLED ແລ້ວ Console ເປັນເວລາ 5 ວິນາທີ ຈາກນັ້ນກີ່ຈະເຂົ້າສູ້ໂທມດ deep sleep ເພື່ອປະຫຍັດພລັງງານຜ່ອໄປ ມີກາຣເກີບຂໍ້ມູນຈຳນວນທີ່ບຸຕິໄວ້ໃນໜ່ວຍຄວາມຈຳຂອງ RTC ດ້ວຍ

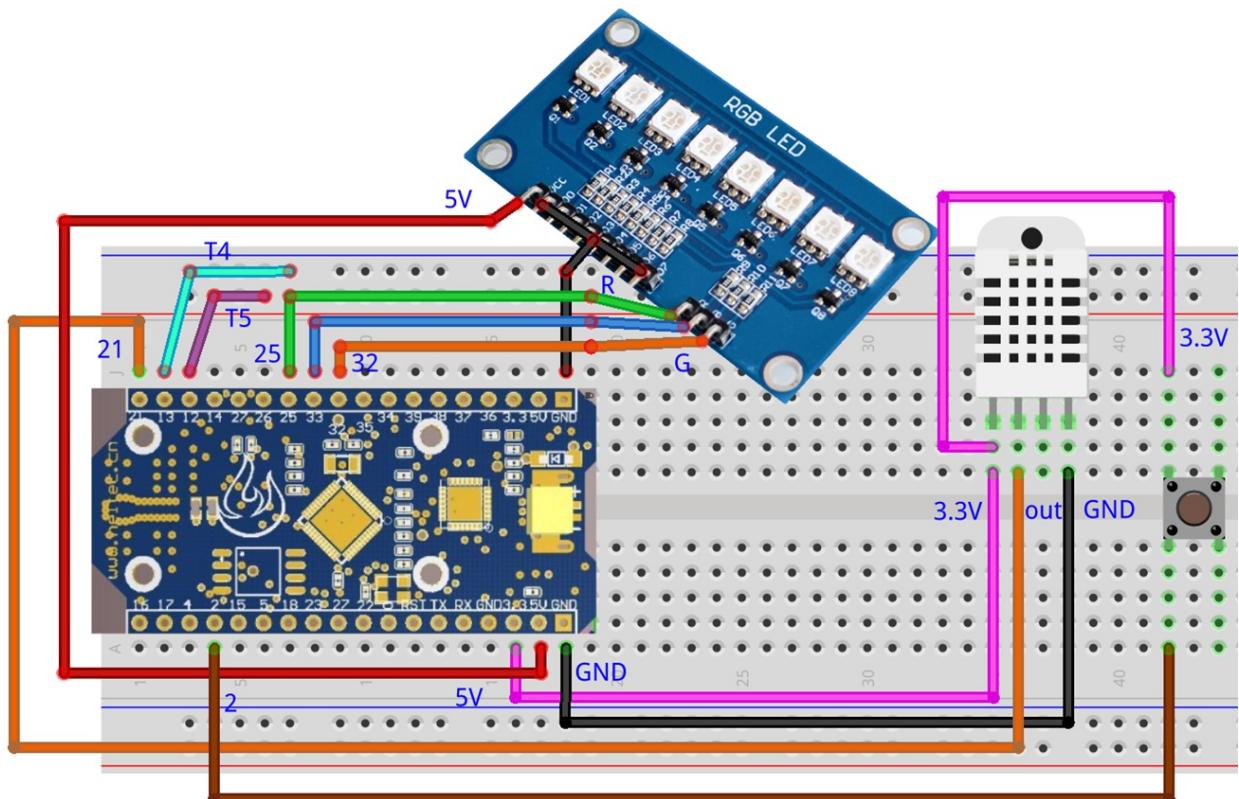
### ຝຶ່ງກໍ່ສັນໃຊ້ງານທີ່ເກີ່ວຂຶ້ອງ

```
esp_deep_sleep_enable_ext0_wakeup((gpio_num_t)2, HIGH);
// ກຳທັນດໍາ GPIO2 ເປັນຕົວຮະຕຸນໃຫ້ອອກຈາກໂທມດ deep sleep ຄໍາມີການເປີ່ຍືນສະຖານະເປັນ HIGH ກາຣທົດສອບນີ້ຈະໃໝ່ໄຟ 3.3v ຈ່າຍເຂົ້າໄປໂດຍໃໝ່ switch ຂັ້ນໄວ້
```

```
esp_deep_sleep_enable_touchpad_wakeup(); // ກຳທັນດກາຣຕື່ນດ້ວຍເຫັນເຂອຮ້
esp_deep_sleep_start(); // ເຂົ້າສູ້ໂທມດ Deep Sleep
Serial.println(touchRead(T4)); // ອ່ານຄ່າ sensor T4 ແລ້ວແສດງຜລໃນ console
RTC_DATA_ATTR int bootCount = 0; // ເກີບຈຳນວນກາຣບຸຕິແຕ່ລະຄັ້ງ
```

ໂທມດປະຫຍດພລັງງານ

## EX7\_DeepSleep\_Touchpad\_Wakeup\_DHT22\_OLED



```
EX7_DeepSleep_Ext0_Wakeup_DHT22_OLED | Arduino 1.8.5

/*
 * EX7_DeepSleep_Ext0_Wakeup_DHT22_OLED
 * อ่านค่าอุณหภูมิและความชื้นจากเซ็นเซอร์ DHT22 มาแสดงผลในคอนโซล
 * และแสดงค่าในจอ OLED 5 วินาที และเข้าໂທມດ deep sleep
 */
-----ตั้งค่าที่นี่-----
#define LED1 32
#define LED2 33
#define LED3 25 //build-in LED on board
#define DHTTYPE dht11
#define DHTPIN 21
-----///
#include "esp_deep_sleep.h"
#include <Wire.h>
#include "SSD1306.h"
SSD1306 display(0x3c, 4, 15);

#include <SimpleDHT.h>           //โหลดไลบรารี่ DHT11
SimpleDHT11 DHTTYPE;            //ระบุรุ่นเซ็นเซอร์รุ่น DHT11
byte temperature = 0;           //กำหนดตัวแปรเก็บค่าอุณหภูมิ
byte humidity = 0;              //กำหนดตัวแปรเก็บค่าความชื้นล้มลัง
```

Done Saving.  
Leaving...  
Hard resetting...

## ໂທມດປະຫຍດພລັງງານ

### EX7\_DeepSleep\_Touchpad\_Wakeup\_DHT22\_OLED

The screenshot shows a terminal window titled "/dev/cu.SLAB\_USBtoUART". The window displays the following text:

```
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:812
load:0x40078000,len:0
load:0x40078000,len:10212
entry 0x40078a00
Wakeup!!
Boot number: 2
Temperature : 28 C Humidity : 70 %
Sleep.....
ets Jun  8 2016 00:22:57

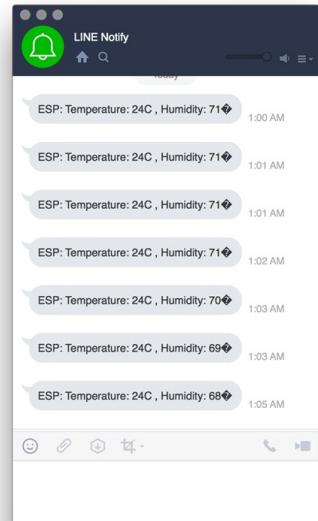
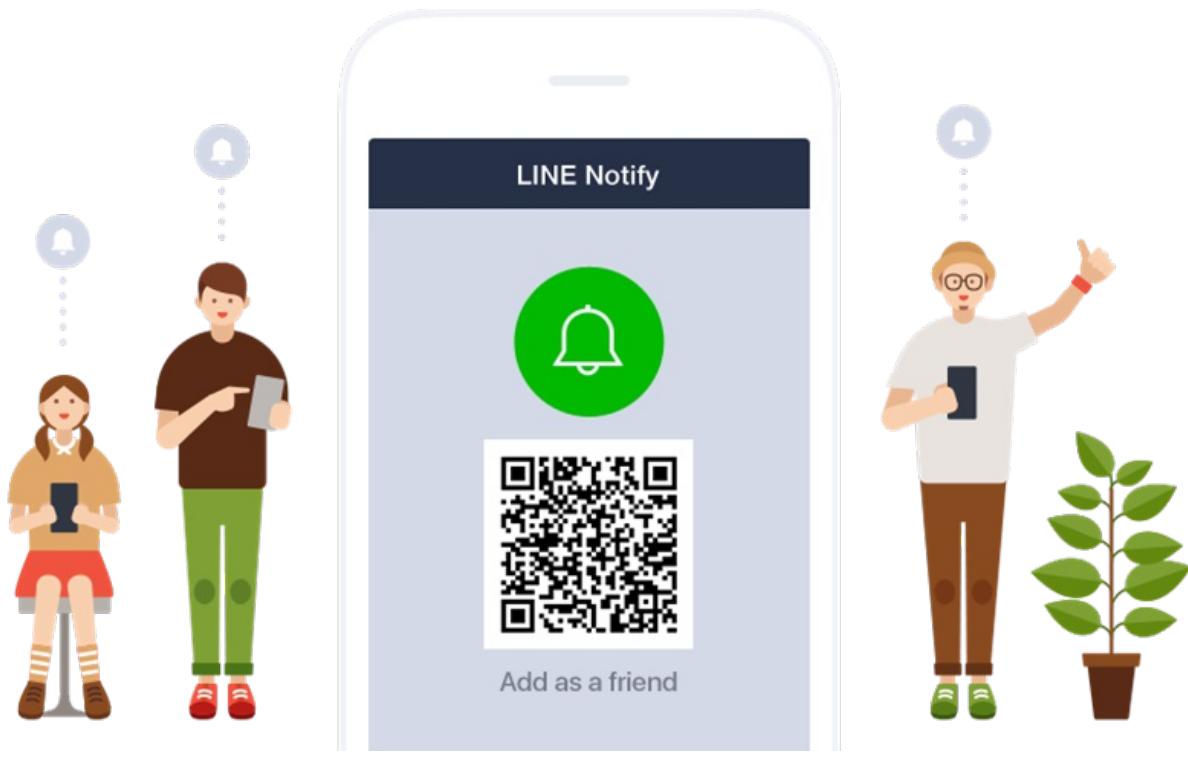
rst:0x5 (DEEPSLEEP_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
configSIP: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:812
load:0x40078000,len:0
load:0x40078000,len:10212
entry 0x40078a00
Wakeup!!
Boot number: 3
Temperature : 28 C Humidity : 69 %
Sleep.....
```

At the bottom of the terminal window, there are three buttons: "Autoscroll" (checked), "Both NL & CR", "115200 baud", and "Clear output".

## EX8\_DeepSleep\_Timer\_Wakeup\_DHT22\_OLED\_Line

ກຳທັນດີໃໝ່ ESP32 ອູ່ໃນໂທມດ deep sleep ເພື່ອປະຫຍັດພລັງງານເປັນເວລາ 1 ນາທີ ອອກຈາກໂທມດ deep sleep

ພອຄືງເວລາທີ່ກຳທັນດີ ມີການກົດສວິຕົ້ນທີ່ຕ່ອງໄວ້ ESP32 ຈະອອກຈາກໂທມດ deep sleep ຈາກນັ້ນຈະທຳການເຂື່ອມຕ່ອງ WiFi ແລະ ອຳນາຄ່າຂໍ້ອມລອຸນໜູມີແລະຄວາມຊື້ນ (DHT22) ໄປແສດງຜລໃນຈອ OLED, Console ແລະສັງເຈັ້ງເຕືອນໃນ Line Notify ລັ້ງຈາກເວລາ 30 ວິນາທີທີ່ຕ່ັງໄວ້ ກີ່ຈະເຂົ້າສູ່ໂທມດ deep sleep ເພື່ອປະຫຍັດພລັງງານຕ່ອງໄປ ມີການເກີບຂໍ້ອມລົງຈານວນທີ່ບູຕູໄວ້ໃນໜ່ວຍຄວາມຈຳຂອງ RTC ດ້ວຍ



# ໂຄມດປະຫຍດພລັງງານ

## EX8\_DeepSleep\_Timer\_Wakeup\_DHT22\_OLED\_Line

### ພັກໜີ້ໃຊ້ຈຳກັດທີ່ເກື່ອງຂອງ

// ກຳທັນດາ GPIO2 ເປັນຕົວກະຮຸ້ນໃຫ້ອອກຈາກໂຄມດ deep sleep ຄໍາມີກາຣເປີ່ມສະພາະເປັນ HIGH ກາຣທົດສອບນີ້ຈະໃຊ້ໄຟ 3.3v ຈ່າຍເຂົ້າໄປໂດຍໃ້ switch ຫັນໄວ້

```
esp_deep_sleep_enable_ext0_wakeup((gpio_num_t)2, HIGH);
```

//ກຳທັນດໃຫ້ຕື່ນເມື່ອຄຣບ 1 ນາທີຂ່າງໜ້າ

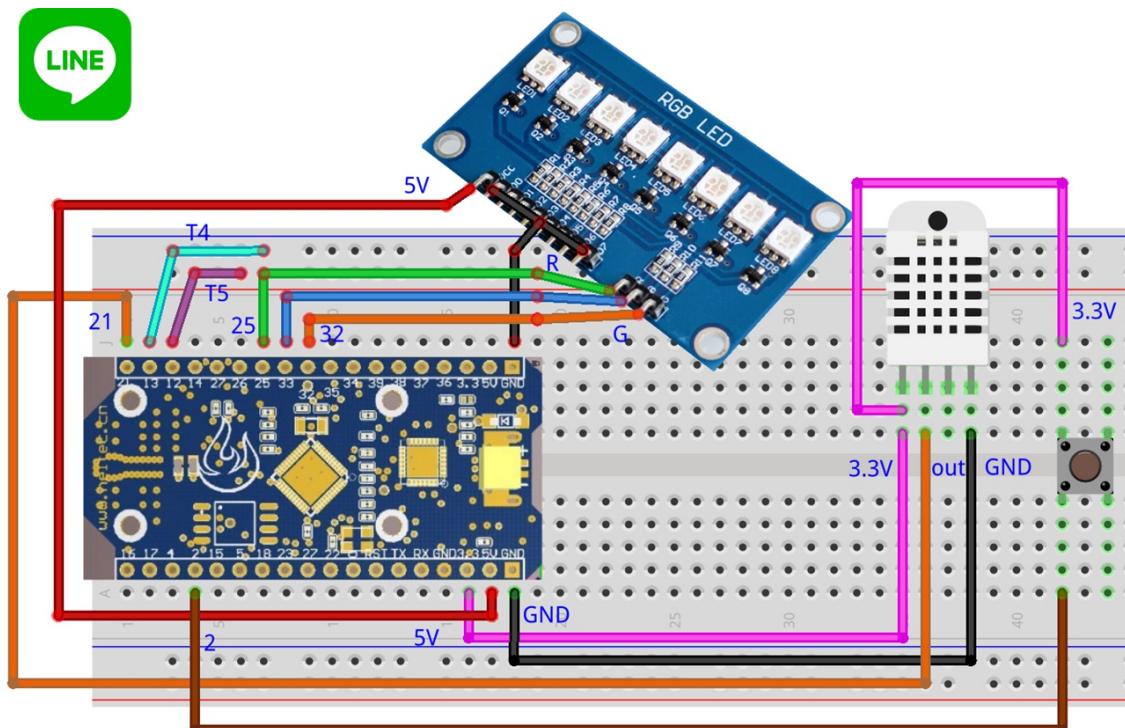
```
esp_deep_sleep_enable_timer_wakeup(1 * 60 * 1000 * 1000);
```

esp\_deep\_sleep\_enable\_touchpad\_wakeup(); // ກຳທັນດກາຣຕື່ນດ້ວຍເຂົ້ນເຊື່ອຮ້

```
esp_deep_sleep_start(); // ເຂົ້າສູ່ໂຄມດ Deep Sleep
```

```
Serial.println(touchRead(T4)); //ອ່ານຄ່າ sensor T4 ແລ້ວແສດງຜລໃນ console
```

```
RTC_DATA_ATTR int bootCount = 0; // ເກີບຈຳນວນກາຣບຸຕແຕ່ລະຄຮັງ
```



## ໂທມດປະຫຍດພລັງງານ

### EX8\_DeepSleep\_Timer\_Wakeup\_DHT22\_OLED\_Lin

```
/*  
 * EX8_DeepSleep_Timer_Wakeup_DHT22_OLED  
 * ອ່ານຄໍາອຸ່ນກູມແລະຄວາມຊື່ຈາກເຊັນເຊົ່ວ໌ DHT22 ມາແສດງຜລໃນຄອນໂສລ  
 * ແສດງຄາໃຈຈອ OLED ແລ້ວສັງຂອມລືໄປຍັງ line ກອນເຫັນສູ່ deep sleep ອີກຮອນ  
 */  
  
-----ຕັ້ງຄ່າທີ່ນີ້-----  
  
#define TokenLine "yff3kysyIvgkgbNbSeHapzHaCW9jK8lzV5hxvAK1k9s" //ຈົກສເໜືອມຕ້ອ Line  
const char* ssid = "plutter"; // your network SSID  
const char* password = "visit012"; // your network password  
const char* server = "notify-api.line.me"; //Line Server URL  
  
#define LED1 32  
#define LED2 33  
#define LED3 25 //build-in LED on board  
#define DHTTYPE dht11  
#define DHTPIN 21  
  
-----  
  
#include <WiFiClientSecure.h>  
#include "esp_deep_sleep.h"  
#include <Wire.h>  
#include "SSD1306.h"  
SSD1306 dispI2C(0x3c, 4, 15);  
  
Done Saving.  
  
Leaving...  
Hard resetting...  
  
10 Heltec_WiFi_LoRa_32, 80MHz, 921600 on /dev/cu.SLAB_USBtoUART
```

configSIP: 0, SPIWP:0xee  
clk\_drv:0x00,q\_drv:0x00,d\_drv:0x00,cs0\_drv:0x00,hd\_drv:0x00,wp\_drv:0x00  
mode:DIO, clock div:1  
load:0x3fff0018,len:4  
load:0x3fff001c,len:812  
load:0x40078000,len:0  
load:0x40078000,len:10212  
entry 0x40078a00  
DHT22 read failed!  
Temperature: 24C , Humidity: 68%  
Boot number: 2  
Attempting to connect to SSID: plutter  
....Connected to plutter  
  
Starting connection to server...  
Connected to server!  
POST /api/notify HTTP/1.1  
Host: notify-api.line.me  
Authorization: Bearer yff3kysyIvgkgbNbSeHapzHaCW9jK8lzV5hxvAK1k9s  
Cache-Control: no-cache  
User-Agent: ESP8266  
Content-Type: application/x-www-form-urlencoded  
Content-Length: 40  
  
message=Temperature: 24C , Humidity: 68%  
headers received

LINE Notify

ESP: Temperature: 24C , Humidity: 71 1:00 AM  
ESP: Temperature: 24C , Humidity: 71 1:01 AM  
ESP: Temperature: 24C , Humidity: 71 1:01 AM  
ESP: Temperature: 24C , Humidity: 71 1:02 AM  
ESP: Temperature: 24C , Humidity: 71 1:03 AM  
ESP: Temperature: 24C , Humidity: 69 1:03 AM  
ESP: Temperature: 24C , Humidity: 68 1:05 AM

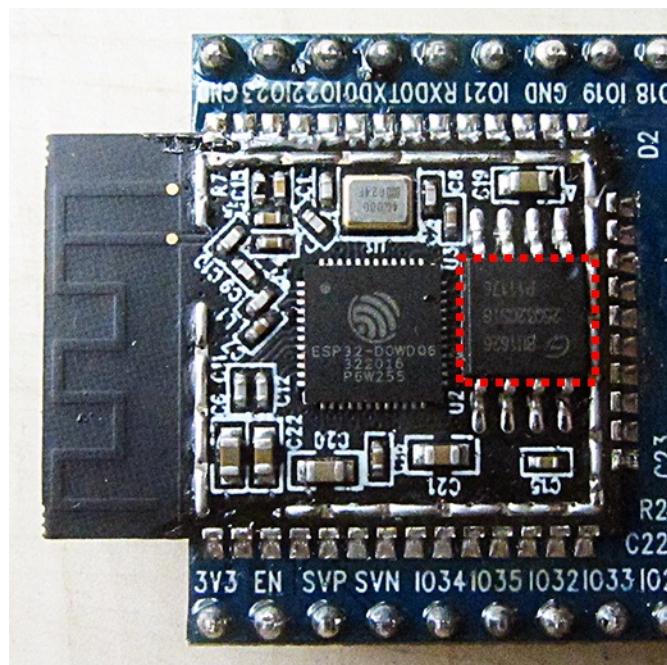
# หน่วยความจำและพื้นที่เก็บข้อมูล

ESP32's Flash memory

# หน่วยความจำและพื้นที่เก็บข้อมูล

ชิป ESP32 จะใช้ Flash memory ภายในในการเก็บโปรแกรม ข้อมูล คอนฟิก และค่าใช้งานต่างๆ การทำงานจะผ่าน SPI bus มีขนาดสูงสุด 16Mb สำหรับโมดูล ESP-WROOM-32 ให้ flash memory มา 4Mb

Vendor	Name	Flash memory (MiB)
Espressif	ESP-WROOM-03	4
	ESP-WROOM-32	4
	ESP32-WROVER	4
Ai-Thinker	ESP-32S	4
AnalogLamb	ESP-32S-ALB	4
	ALB-WROOM	16
	ALB32-WROVER	4
Pycom <sup>[20]</sup>	W01	8
	L01	8
	S01	8
	G01	8



## หน่วยความจำและพื้นที่เก็บข้อมูล

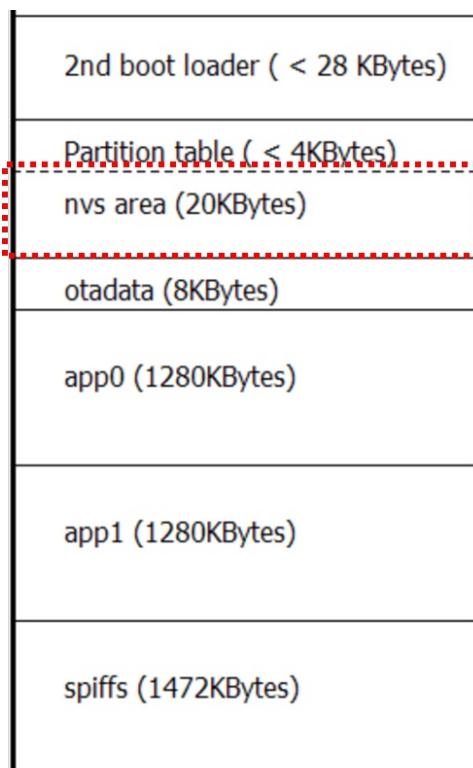
Table: Flash memory Partition Table

# Name	Type	SubType	Offset	Size	KBytes
nvs	data	nvs	0x9000	0x5000	20
otadata	data	ota	0xe000	0x2000	8
app0	app	ota_0	0x10000	0x140000	1280
app1	app	ota_1	0x150000	0x140000	1280
eeprom	data	0x99	0x290000	0x1000	4
spiffs	data	spiffs	0x291000	0x16F000	1468

พาทิชันที่สามารถเขียนโปรแกรมเก็บแบบถาวร ได้แก่ NVS มีขนาด 20KB EEPROM มีขนาด 4KB และ SPIFFS มีขนาด 1.3 MB ส่วนอกนั้นเป็นพื้นที่ของระบบที่ต้องใช้

### NVS (Nonvolatile Storage)

เป็น SPI flash memory ของโมดูล ESP32 มีขนาด 20 Kbytes เอาไว้เก็บข้อมูลระบบ เช่น ชื่อ SSID MC Address และอื่นๆ ข้อมูลที่เก็บจะเป็นแบบถาวรไม่สูญหายแม้เขียน sketch ทับลงไป การใช้งานให้นึกเสมอว่าไฟล์จะอยู่ร่วมกับไฟล์ระบบอื่นๆ ระมัดระวังการใช้งาน แนะนำให้เก็บข้อมูลที่ไม่ซับซ้อน เช่น รหัสผ่าน ตัวเลขบันทึกจำนวนการบูต หรือข้อความสั้นๆ

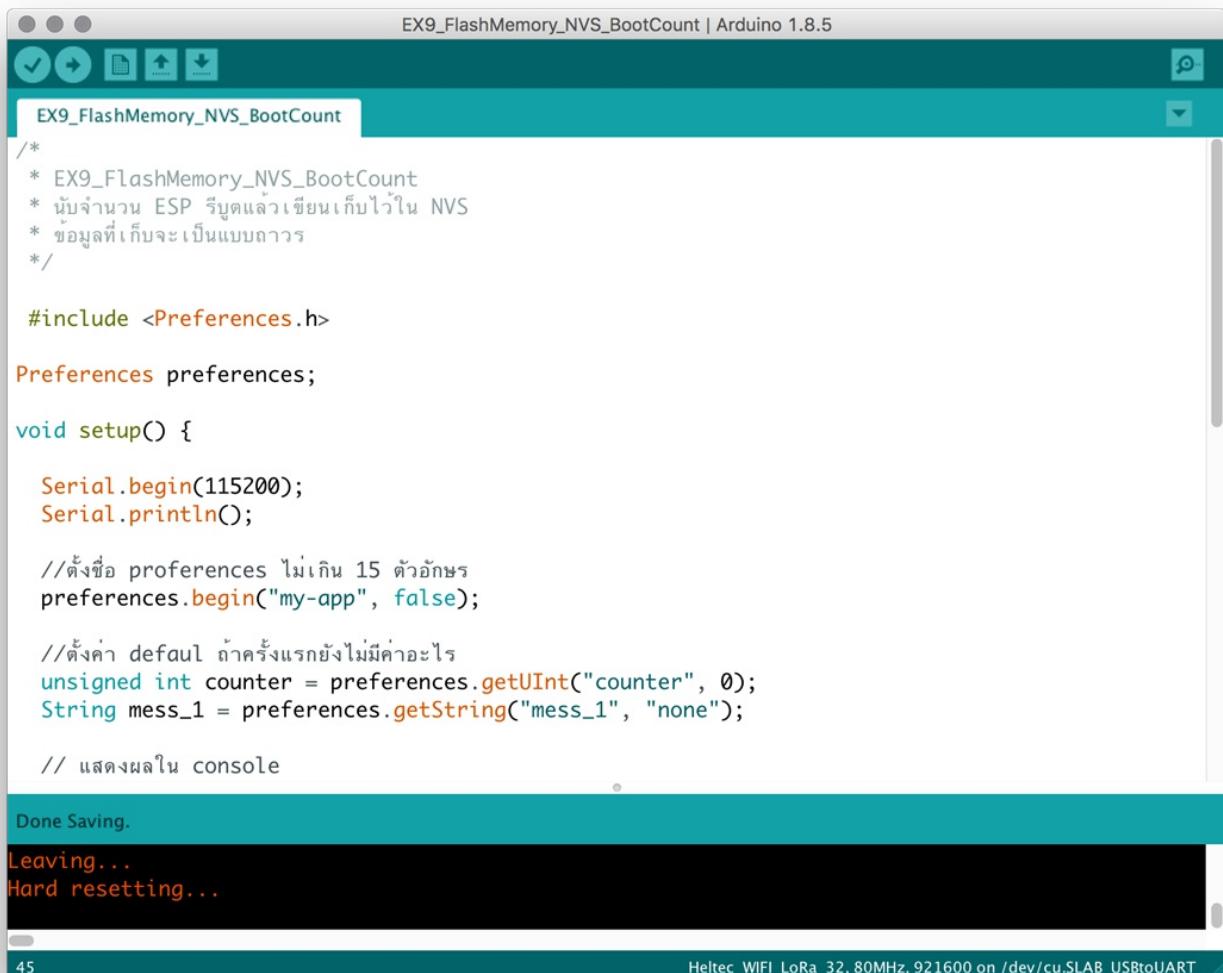


## หน่วยความจำและพื้นที่เก็บข้อมูล

## EX9\_FlashMemory\_NVS\_BootCount

นับจำนวน ESP รีบูตแล้วเขียนเก็บไว้ใน SPI flash memory - NVS (Nonvolatile Storage) ของโมดูล ESP32 ข้อมูลที่เก็บจะเป็นแบบถาวร ไม่สูญหายแม้มีการเขียน Sketch ทับลงไป

## หน่วยความจำและพื้นที่เก็บข้อมูล EX9\_FlashMemory\_NVS\_BootCount



```
EX9_FlashMemory_NVS_BootCount | Arduino 1.8.5

EX9_FlashMemory_NVS_BootCount

/*
 * EX9_FlashMemory_NVS_BootCount
 * นับจำนวน ESP รีบูตแล้วเขียนเก็บไว้ใน NVS
 * ข้อมูลที่เก็บจะเป็นแบบถาวร
 */

#include <Preferences.h>

Preferences preferences;

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

    //ตั้งชื่อ preferences ไม่เกิน 15 ตัวอักษร
    preferences.begin("my-app", false);

    //ตั้งค่า default ถ้าครั้งแรกซึ่งไม่มีค่าอะไร
    unsigned int counter = preferences.getInt("counter", 0);
    String mess_1 = preferences.getString("mess_1", "none");

    // แสดงผลใน console
}

Done Saving.

Leaving...
Hard resetting...

45
```

Heltec\_WIFI\_LoRa\_32, 80MHz, 921600 on /dev/cu.SLAB\_USBtoUART

หน่วยความจำและพื้นที่เก็บข้อมูล  
EX9\_FlashMemory\_NVS\_BootCount

```
comittstp. v, SPRINT.0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:812
load:0x40078000,len:0
load:0x40078000,len:10212
entry 0x40078a00

Current counter value: 80
Current text : TextNumber-80
Restarting in 10 seconds...
ets Jun  8 2016 00:22:57

rst:0xc (SW_CPU_RESET),boot:0x17 (SPI_FAST_FLASH_BOOT)
configip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:812
load:0x40078000,len:0
load:0x40078000,len:10212
entry 0x40078a00

Current counter value: 81
Current text : TextNumber-81
Restarting in 10 seconds...
```

Autoscroll

Both NL & CR

115200 baud

Clear output

## หน่วยความจำและพื้นที่เก็บข้อมูล

## EX10\_FlashMemory\_EEPROM

EEPROM เป็นหน่วยความจำแบบถาวร ข้อมูลไม่สูญหายแม้ว่าไม่มีไฟเลี้ยง การอ่านเขียนข้อมูลทำได้ครั้งละ byte เท่านั้น เขียนข้อมูลได้สูงสุดประมาณ 100,000 ครั้ง

ทดสอบเขียนข้อมูลลงบน SPI flash memory ของโมดูล ESP32 ใน EEPROM ที่มีขนาด 4KB เป็นการเก็บข้อมูลแบบถาวร อ่านค่าจาก EEPROM แปลง char array เป็น String มาแสดงผลใน Console

```
EX10_FlashMemory_EEPROM | Arduino 1.8.5
Serial Monitor

EX10_FlashMemory_EEPROM

/*
 * EX10_FlashMemory_EEPROM
 * ทดสอบเขียนข้อมูลลงใน SPI flash memory ของโมดูล ESP32
 * ใน EEPROM ที่มีขนาด 4KB เป็นการเก็บข้อมูลแบบถาวร
 * อ่านค่าจาก EEPROM แปลง char array เป็น String มาแสดงผลใน Console
 */

#include "EEPROM.h"

char msg[] = "RSUwifi,2.4GHz,CH6,20MHz,rsu01,12345678"; //ข้อมูลที่ต้องการเขียนลงใน EEPROM
int msgSize = sizeof(msg); //นับจำนวนตัวอักษร

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

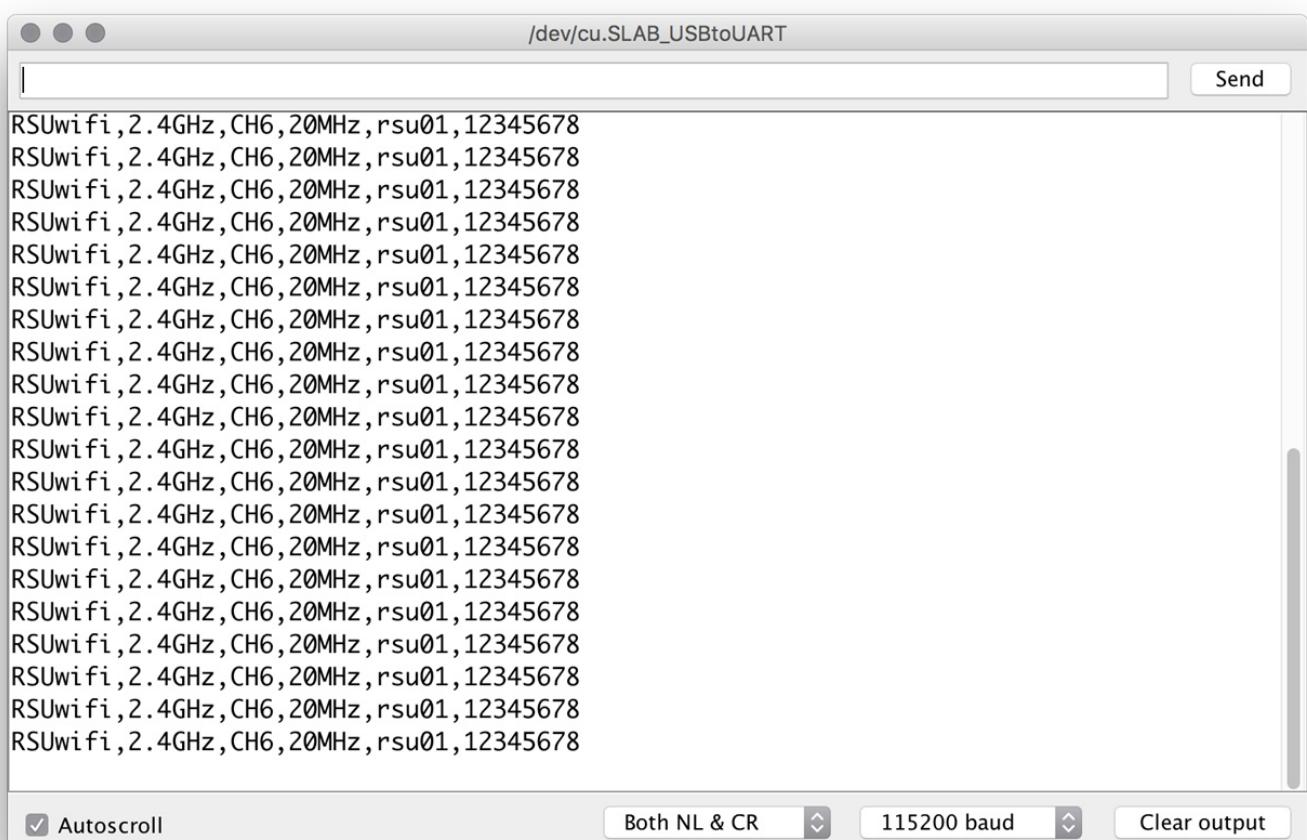
    //เขียนข้อมูลลง EEPROM เริ่มที่ตำแหน่งที่ 0
    for (int i=0; i < msgSize; i++) {
        EEPROM.write(i, msg[i]);
    }
    EEPROM.commit();
}

void loop() {
    //อ่านค่าจาก EEPROM แปลง char array เป็น String
    String txt;
    for (int i = 0; i < msgSize; i++) {

Done uploading.

Leaving...
Hard resetting...
```

หน่วยความจำและพื้นที่เก็บข้อมูล  
EX10\_FlashMemory EEPROM



The screenshot shows a terminal window titled "/dev/cu.SLAB\_USBtoUART". The window displays a series of identical lines of text: "RSUwifi,2.4GHz,CH6,20MHz,rsu01,12345678". This pattern repeats 15 times. At the bottom of the window, there are three buttons: "Autoscroll" (checked), "Both NL & CR", and "115200 baud". To the right of these buttons is a "Clear output" button.

```
RSUwifi,2.4GHz,CH6,20MHz,rsu01,12345678
```

## หน่วยความจำและพื้นที่เก็บข้อมูล

### EX11\_FlashMemory\_SPIFFS

SPIFFS (SPI Flash File System) จะอยู่ใน flash memory of ESP-WROOM-32 (ESP32) เป็นพื้นที่เก็บข้อมูลที่เราจะสามารถอัปโหลดไฟล์ไปเก็บไว้ได้ เช่นเก็บบันทึกข้อมูลอุณหภูมิและความชื้นของเซ็นเซอร์ DHT22 ค่าคอนฟิก เป็นต้น มีพื้นที่เก็บข้อมูลขนาด 1.5M

### การอัปโหลดไฟล์ไปเก็บไว้ใน SPIFFS

ก่อนอื่นต้องทำการติดตั้ง tool ชื่อ “ESP32FS” ไว้ในโฟล์เดอร์

MAC:

/Users/visit01/Documents/Arduino/tools

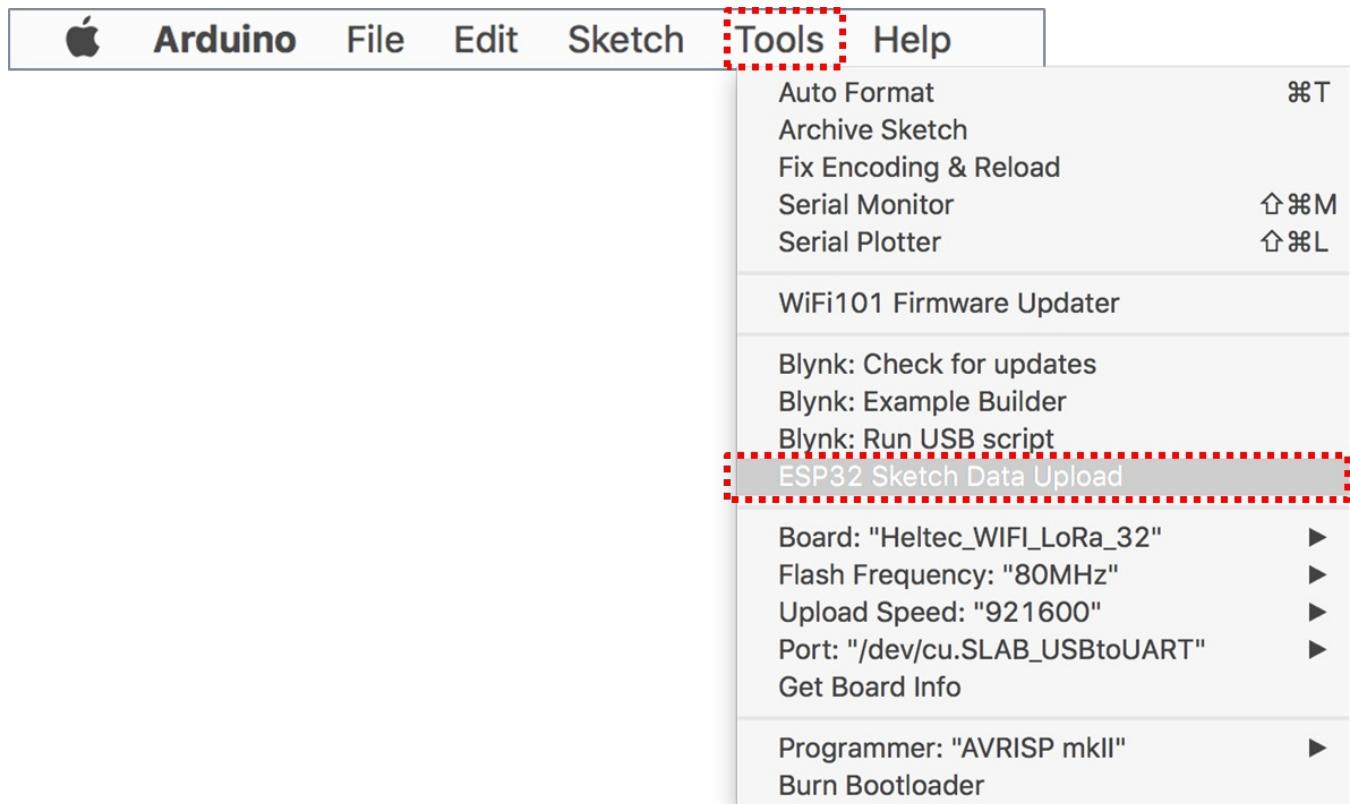
Windows:

C:/Documents/Arduino/tools

Name	Date Modified	Size	Kind
► BlynkExampleBuilder	Apr 4, 2017, 23:57	--	Folder
► BlynkUpdater	Apr 9, 2017, 15:23	--	Folder
► BlynkUsbScript	Apr 9, 2017, 15:23	--	Folder
► ESP32FS	Today, 10:56	--	Folder
► tool	Sep 30, 2017, 19:03	--	Folder
esp32fs.jar	Sep 30, 2017, 19:03	8 KB	Java JAR file

# หน่วยความจำและพื้นที่เก็บข้อมูล EX11\_FlashMemory\_SPIFFS

## การอัปโหลดไฟล์ไปเก็บไว้ใน SPIFFS



ให้เก็บไฟล์ที่ต้องการอัปโหลดเก็บไว้ในไฟล์เดอร์ (ไม่เกิน 1.1MB)

MAC

/Users/visit01/Documents/Arduino/TRAIN-IOT-ADVANCE/EX11\_FlashMemory\_SPIFFS/data

Windows

C:/Documents/Arduino/TRAIN-IOT-ADVANCE/EX11\_FlashMemory\_SPIFFS/data

Name	Date Modified	Size	Kind
pic1.zip	Today, 09:55	576 KB	ZIP archive
pic2.png	Yesterday, 01:21	53 KB	PNG image
pic3.png	Yesterday, 01:21	53 KB	PNG image
pic4.jpg	Yesterday, 17:58	383 KB	JPEG image

## หน่วยความจำและพื้นที่เก็บข้อมูล EX11\_FlashMemory\_SPIFFS



```
/*
 * EX11_FlashMemory_SPIFFS
 * ดูรายละเอียดของ SPI flash memory ของโมดูล ESP32
 * SPIFFS สามารถเก็บข้อมูลแบบถาวร มีขนาดความจุที่สามารถอัปโหลดไฟล์ได้ 1.1MB
 */

#include "FS.h"
#include <SPIFFS.h>

void setup() {
    Serial.begin(115200);
    delay(2000);
    Serial.println("-----");

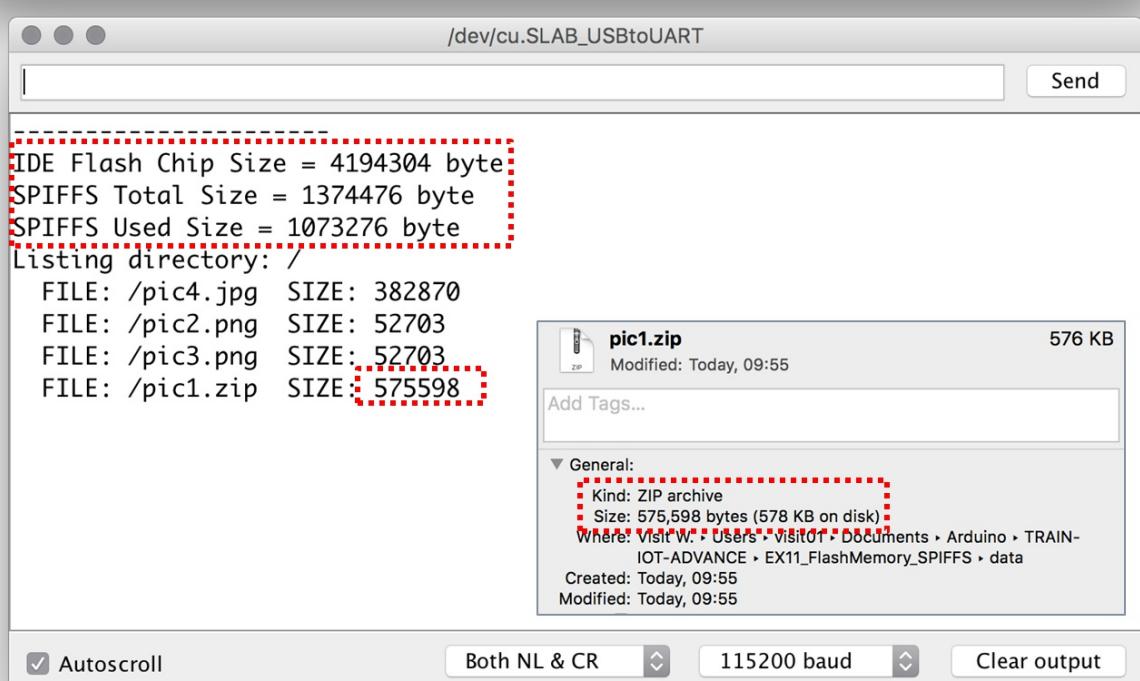
    if (!SPIFFS.begin()) {
        Serial.println("SPIFFS Mount Failed");
        return;
    }

    Serial.printf("IDE Flash Chip Size = %d byte\r\n", ESP.getFlashChipSize());
    Serial.printf("SPIFFS Total Size = %d byte\r\n", SPIFFS.totalBytes());
    Serial.printf("SPIFFS Used Size = %d byte\r\n", SPIFFS.usedBytes());

    delay(100);
}

Done Saving.

Leaving...
Hard resetting...
```



```
/dev/cu.SLAB_USBtoUART
| IDE Flash Chip Size = 4194304 byte
| SPIFFS Total Size = 1374476 byte
| SPIFFS Used Size = 1073276 byte
Listing directory: /
FILE: /pic4.jpg SIZE: 382870
FILE: /pic2.png SIZE: 52703
FILE: /pic3.png SIZE: 52703
FILE: /pic1.zip SIZE: 575598
```

pic1.zip  
Modified: Today, 09:55  
576 KB  
Add Tags...  
General:  
Kind: ZIP archive  
Size: 575,598 bytes (578 KB on disk)  
Where: Windows\Users\VISIT01\Documents\Arduino\TRAIN-IOT-ADVANCE\EX11\_FlashMemory\_SPIFFS\data  
Created: Today, 09:55  
Modified: Today, 09:55

Autoscroll Both NL & CR 115200 baud Clear output

## การขยายพาร์ทิชัน SPIFFS

เราสามารถขยายพื้นที่ของ SPIFFS เพื่อเก็บข้อมูลได้มากขึ้น Arduino IDE จะเก็บข้อมูลการแบ่งพาร์ทิชันไว้ที่ไฟล์ชื่อ “default.csv” ให้ทำด้วยความระมัดระวัง !

MAC

/Users/visit01/Documents/Arduino/hardware/espressif/esp32/tools/partitions

Windows

C:/Documents/Arduino/hardware/espressif/esp32/tools/partitions

----- พาร์ทิชัน default ESP32 -----

```
# Name, Type, SubType, Offset, Size, Flags
    nvs,     data, nvs,    0x9000, 0x5000,
    otadata, data, ota,   0xe000, 0x2000,
    app0,    app, ota_0,  0x10000, 0x140000,
    app1,    app, ota_1,  0x150000, 0x140000,
    eeprom,   data, 0x99,  0x290000, 0x1000,
    spiffs,   data, spiffs, 0x291000, 0x16F000,
```

----- พาร์ทิชันหลังจากปรับแต่งแก้ไข -----

```
# Name, Type, SubType, Offset, Size, Flags
    nvs,     data, nvs,    0x9000, 0x5000,
    otadata, data, ota,   0xe000, 0x2000,
    app0,    app, ota_0,  0x10000, 0xF0000,
    app1,    app, ota_1,  0x100000, 0xF0000,
    eeprom,   data, 0x99,  0x1F0000, 0x1000,
    spiffs,   data, spiffs, 0x1F1000, 0x20F000,
```

## หน่วยความจำและพื้นที่เก็บข้อมูล

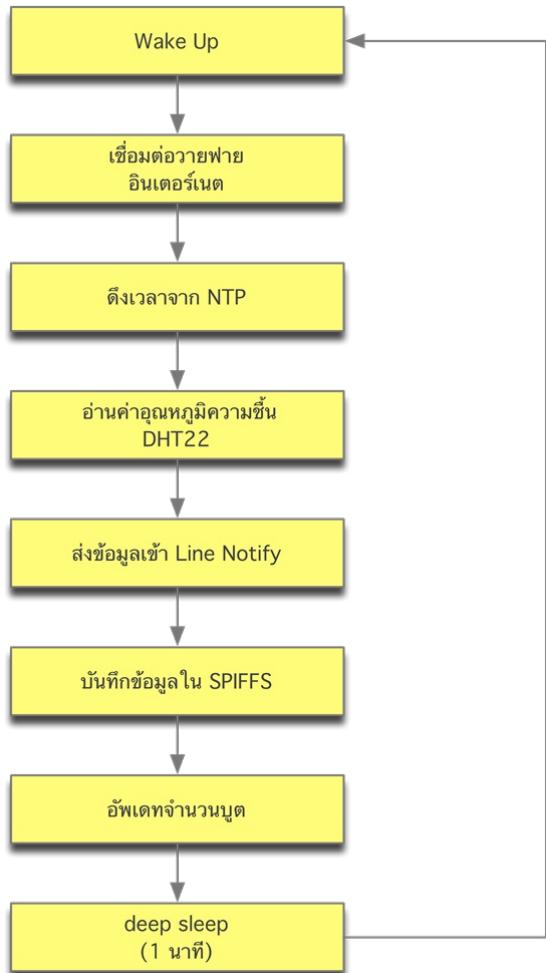
----- พาร์ทิชัน default ESP32 -----

```
|-----|  
| |-----|  
| IDE Flash Chip Size = 4194304 byte  
| SPIFFS Total Size = 1374476 byte  
| SPIFFS Used Size = 1073276 byte  
| Listing directory: /  
| FILE: /pic4.jpg SIZE: 382870  
| FILE: /pic2.png SIZE: 52703  
| FILE: /pic3.png SIZE: 52703  
| FILE: /pic1.zip SIZE: 575598  
|-----|  
  
 Autoscroll      Both
```

----- พาร์ทิชันหลังจากรับแต่งแก้ไข -----

```
|-----|  
| |-----|  
| IDE Flash Chip Size = 4194304 byte  
| SPIFFS Total Size = 1976876 byte  
| SPIFFS Used Size = 1073276 byte  
| Listing directory: /  
| FILE: /pic1.zip SIZE: 575598  
| FILE: /pic4.jpg SIZE: 382870  
| FILE: /pic2.png SIZE: 52703  
| FILE: /pic3.png SIZE: 52703  
|-----|  
  
 Autoscroll      Both
```

## EX12\_FlashMemory\_SPIFFS\_Line



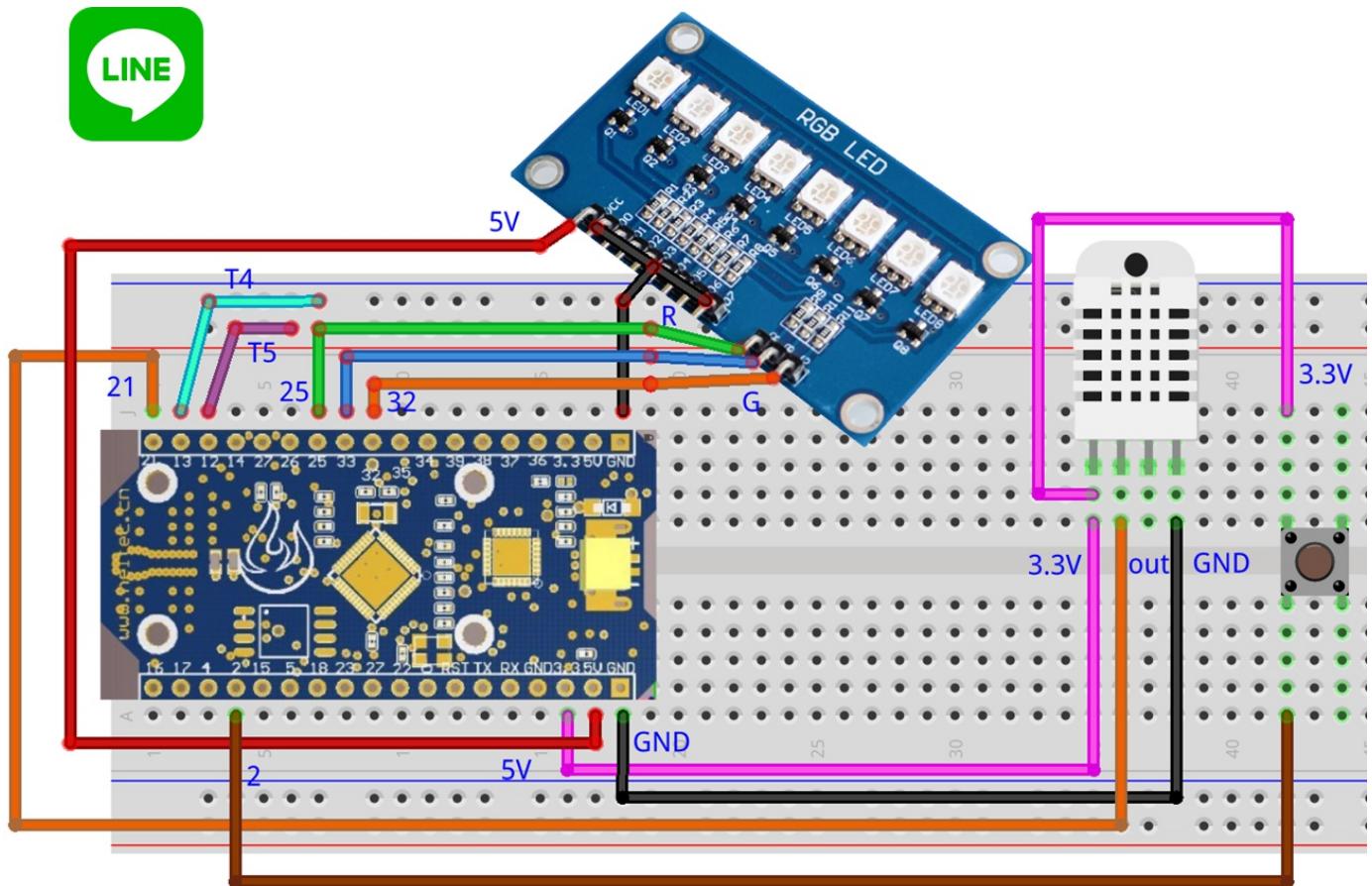
ออกจากการdeep sleep แบบอัตโนมัติตามเวลาที่กำหนดหรือกดปุ่มสวิตซ์ แล้ว เชื่อมต่อ WiFi และดึงค่าเวลาจาก NTP server ผ่านอินเตอร์เน็ต อ่านค่าอุณหภูมิและความชื้นจากเซนเซอร์ DHT22 มาแสดงผลในคอนโซล และเก็บบันทึกข้อมูลไว้ใน SPIFFS พร้อมกับเวลาปัจจุบันจากนั้นรายงานค่าอุณหภูมิและความชื้นเข้า Line Notify แสดงขั้นตอนทำงานบนจอ OLED เป็นระยะ รวมถึงเก็บจำนวนการบูตไว้ใน RTC ก่อนเข้าสู่ deep sleep ที่ตั้งเวลาได้

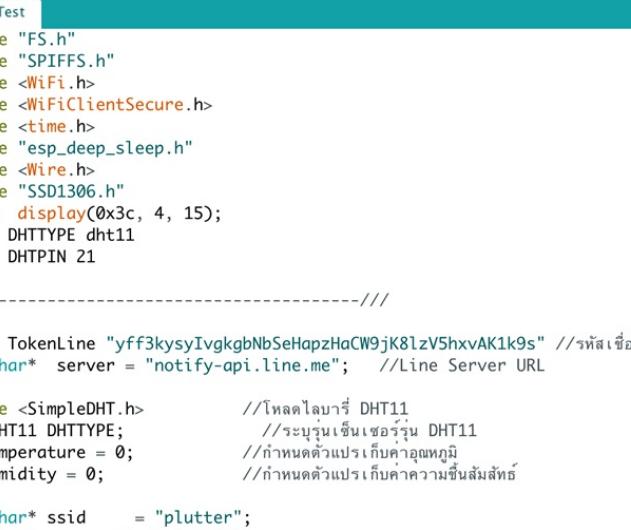
### ตัวอย่างฟังก์ชันที่สร้างขึ้นใช้งาน

```

listDir(SPIFFS, "/", 0); //แสดงโฟล์เดอร์ทั้งหมด
writeFile(SPIFFS, "/hello.txt", "Hello "); //เขียนทั้งไฟล์
appendFile(SPIFFS, "/hello.txt", "World!\n"); //เขียนต่อไฟล์
readFile(SPIFFS, "/hello.txt"); //อ่านไฟล์
deleteFile(SPIFFS, "/foo.txt"); //ลบไฟล์
renameFile(SPIFFS, "/hello.txt", "/foo.txt"); //เปลี่ยนชื่อไฟล์
testFileIO(SPIFFS, "/test.txt"); //ทดสอบไฟล์
Showdisplay("Wake UP:"); //แสดงข้อความบนจอ OLED
Gettemp(); อ่านค่าอุณหภูมิและความชื้น
linenotify(); ส่ง Line notify
  
```

## หน่วยความจำและพื้นที่เก็บข้อมูล EX12\_FlashMemory\_SPIFFS\_Line





The screenshot shows the Arduino IDE interface with the title "SPIFFS\_Test | Arduino 1.8.5". The code editor contains the following C++ code:

```
#include "FS.h"
#include "SPIFFS.h"
#include <WiFi.h>
#include <WiFiClientSecure.h>
#include <time.h>
#include "esp_deep_sleep.h"
#include <Wire.h>
#include "SSD1306.h"
SSD1306 display(0x3c, 4, 15);
#define DHTTYPE dht11
#define DHTPIN 21

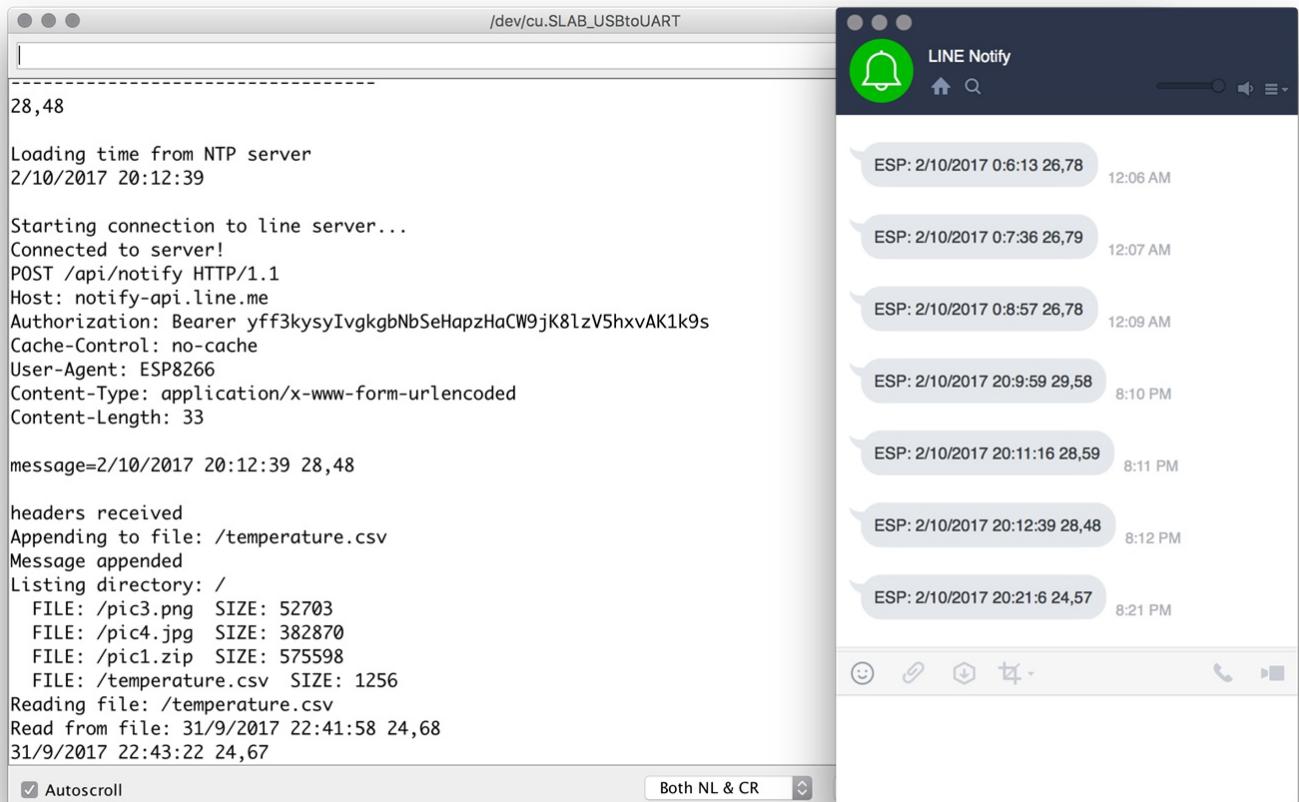
//-----
#define TokenLine "yff3kysyIvgkgbNbSeHapzHaCW9jK8lzVShxvAK1k9s" //รหัสเชื่อมต่อ Line
const char* server = "notify-api.line.me"; //Line Server URL

#include <SimpleDHT.h>          //โหลดไลบรารี DHT11
SimpleDHT11 DHTTYPE;           //ระบุรุ่นเซ็นเซอร์รุ่น DHT11
byte temperature = 0;           //กำหนดค่าแปรเบื้องต้นค่าอุณหภูมิ
byte humidity = 0;              //กำหนดค่าแปรเบื้องต้นค่าความชื้นสัมลักษณ์

const char* ssid      = "plutter";
const char* password = "visit012";

int timezone = 7 * 3600;        //ตั้งเวลา TimeZone ตามเวลาประเทศไทย
```

# หน่วยความจำและพื้นที่เก็บข้อมูล EX12\_FlashMemory\_SPIFFS\_Line

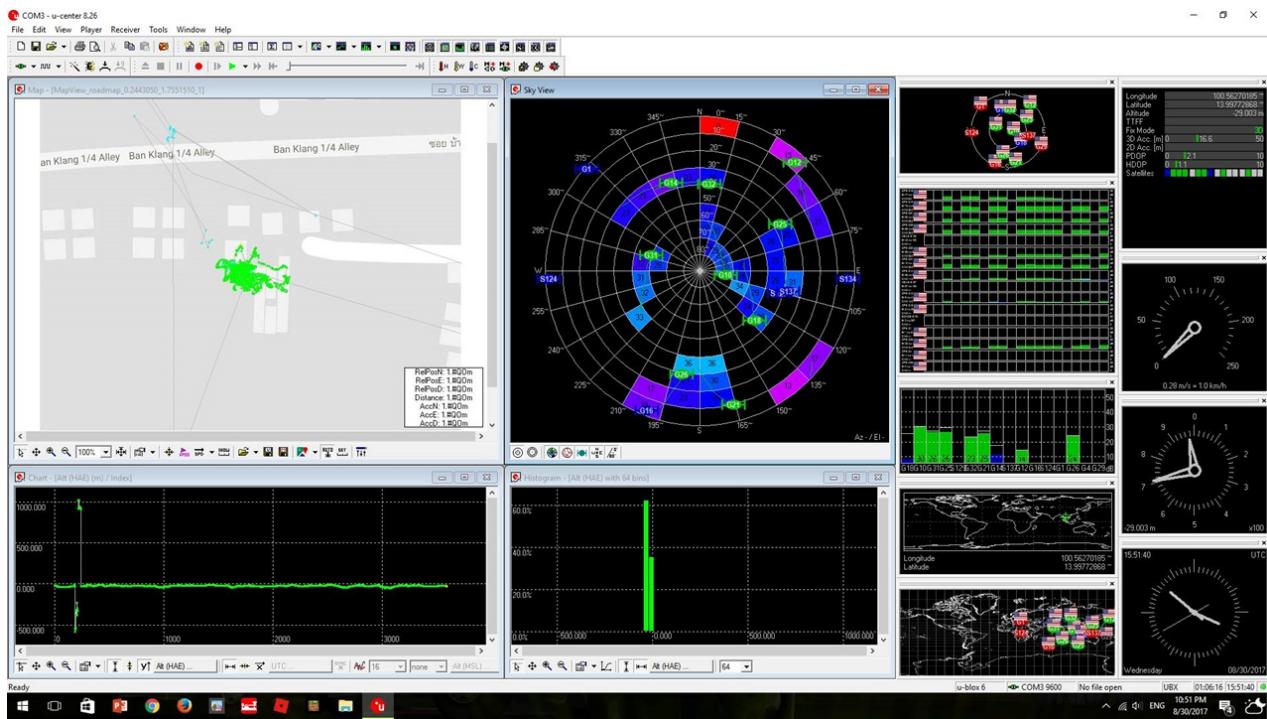


# บทที่ 3 ESP32 Framework

- การรับพิกัดตำแหน่ง GPS
- RESTful Framework

# การรับพิกัดตำแหน่ง GPS

## GPS on ESP32



u-center Windows

GNSS evaluation software for Windows

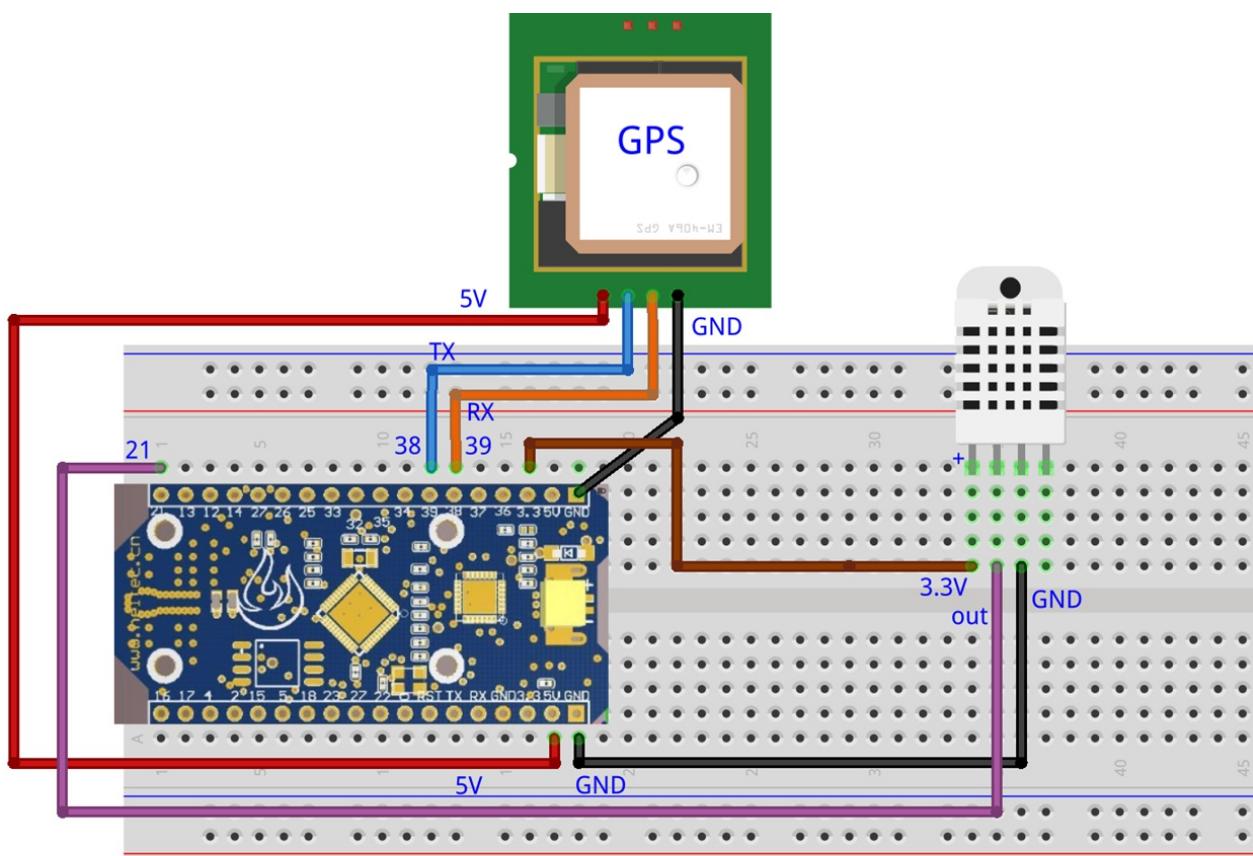
<https://www.u-blox.com/en/product/u-center-windows>

## EX13\_BlynkGPS

รับค่า GPS และวัดอุณหภูมิและความชื้นจาก DHT22 จากนั้นนำไปแสดงตำแหน่งบนแผนที่และค่าพิกัดใน LCD บน Blynk Application

กำหนดการเข้มต่อ

1. Sensor DHT22 วัดอุณหภูมิและความชื้น ต่อขา 21
2. Sensor Analog ต่อขา A6, A7 //ขา GPIO34, 35
3. GPS ต่อขา RXPin = 38, TXPin = 39 , VCC เข้า 5V และ GND เข้า GND
4. Analog ขา A6(GPIO 34), A7 (GPIO35) ถ้ามี



## การรับพิกัดตำแหน่ง GPS

### EX13\_BlynkGPS

```
#include <SimpleDHT.h>
#include <TinyGPS++.h>
#include <Wire.h>
#include "SSD1306.h"
SSD1306 display(0x3c, 4, 15);

//-----
#define DHTTYPE dht11
#define DHTPIN 21

/*ค่อนพิก token เข้า Blynk server เป็นยังไงก็ได้*/
char auth[] = "5848e29f6cd14f41b9b5a9c556aa4479";
char ssid[] = "plutter";
char pass[] = "visit012";

const double Home_LAT = 14.005691; // กำหนดตำแหน่งต้นทางที่จะวัดระยะห่าง Latitude
const double Home_LNG = 100.562408; // กำหนดตำแหน่งต้นทางที่จะวัดระยะห่าง Longitude

static const int RXPin = 38, TXPin = 39; // กำหนดขา Ublox
static const uint32_t GPSBaud = 9600; // กำหนด Ublox GPS Rate 9600

//-----
TinyCDCplus_001.ino // สำหรับอินเทล GPS

Done Saving.
Hash of data verified.

Leaving...
Hard resetting...

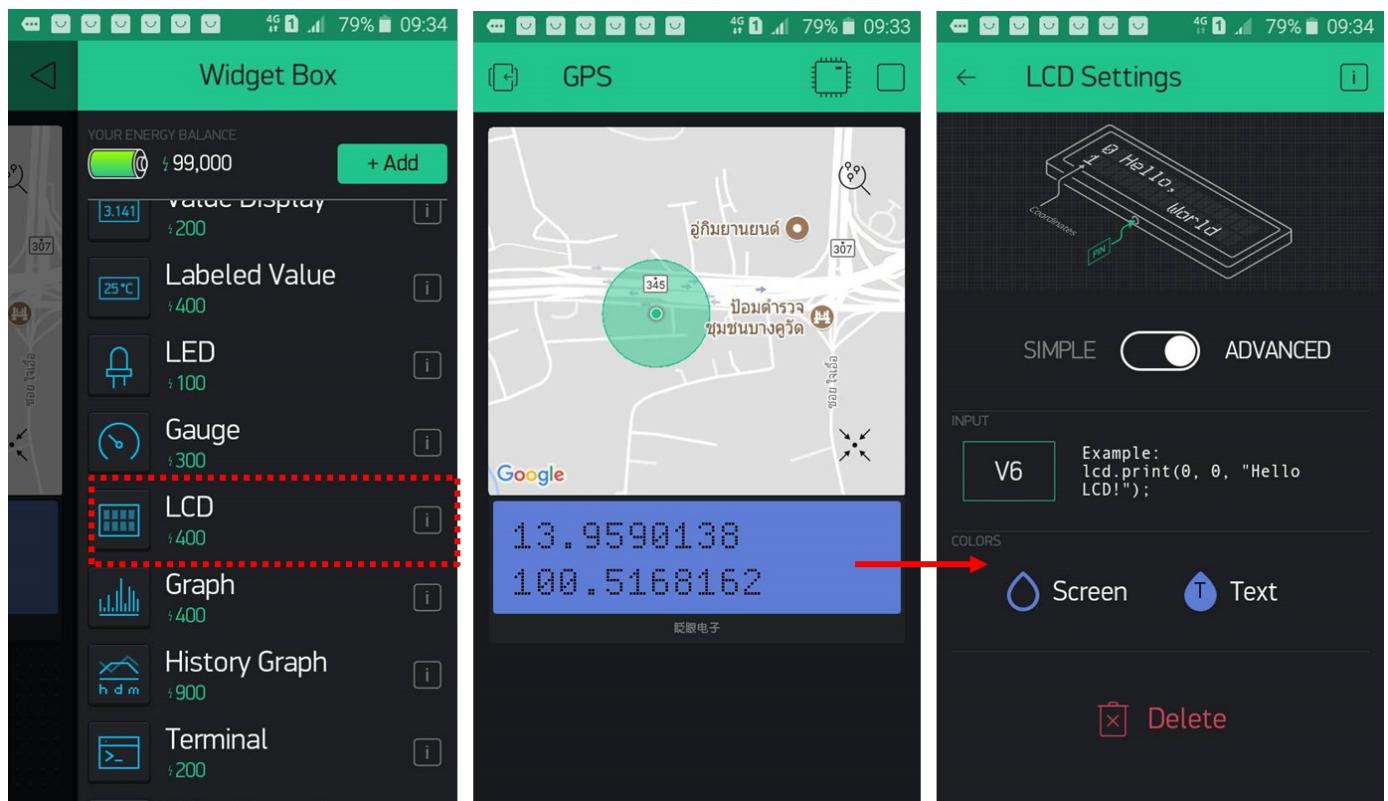
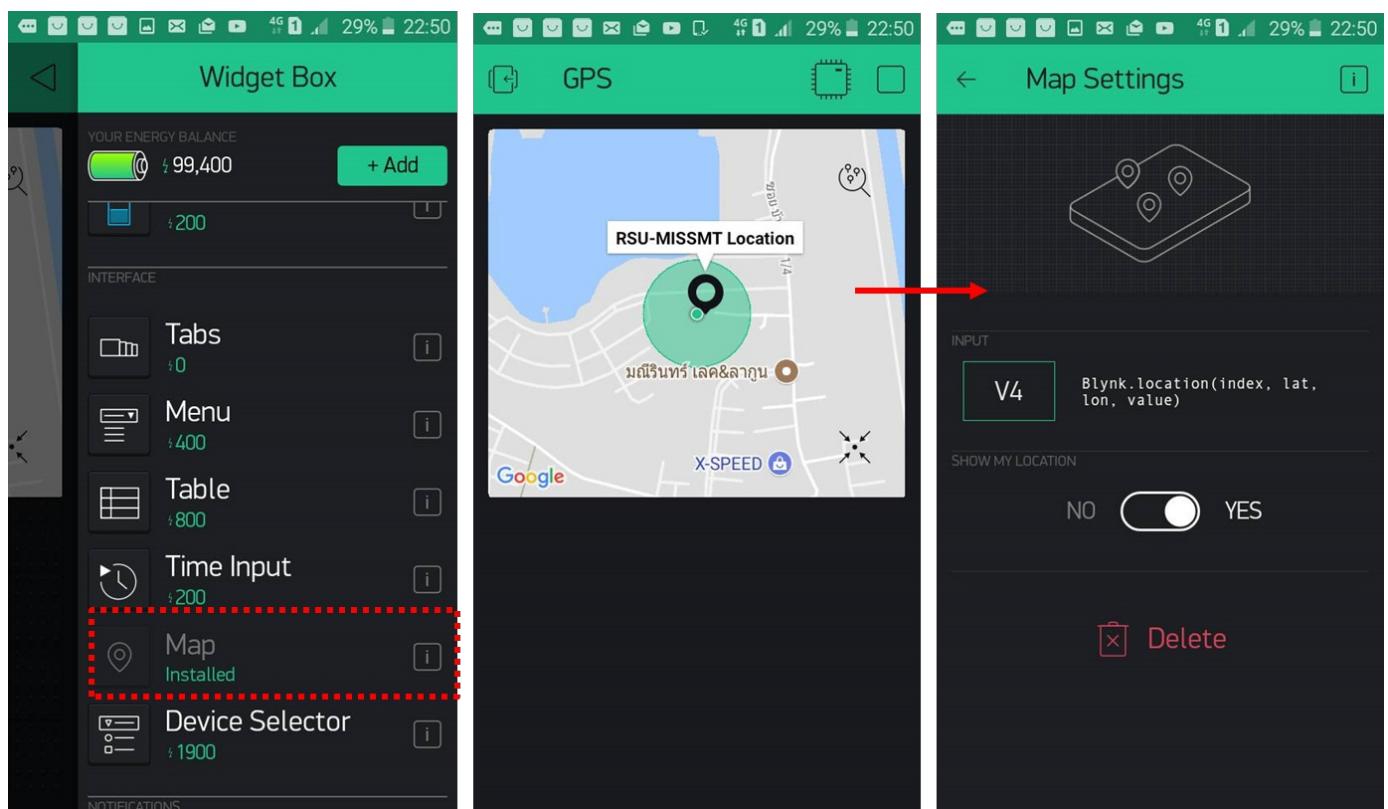
60
```

```
Latitude : 13.958925
Longitude : 100.516749
Satellites: 10
Elevation : 24.70m
Time UTC : 1:54:6
Heading : 0.00
Speed : 0.06
Temperature = 25
Humidity = 80
Analog Sensor1 = 304
Analog Sensor2 = 528
Hall sensor: 4
M to Home: 7165

Latitude : 13.958940
Longitude : 100.516753
Satellites: 10
Elevation : 19.40m
Time UTC : 1:54:11
Heading : 0.00
Speed : 0.05
Temperature = 25
Humidity = 80
Analog Sensor1 = 337
Analog Sensor2 = 544
Hall sensor: -10
M to Home: 7163

Autoscroll Both NL & CR 115200 baud Clear output
```

## การรับพิกัดตำแหน่ง GPS EX13\_BlynkGPS



# aREST

## Framework on ESP32

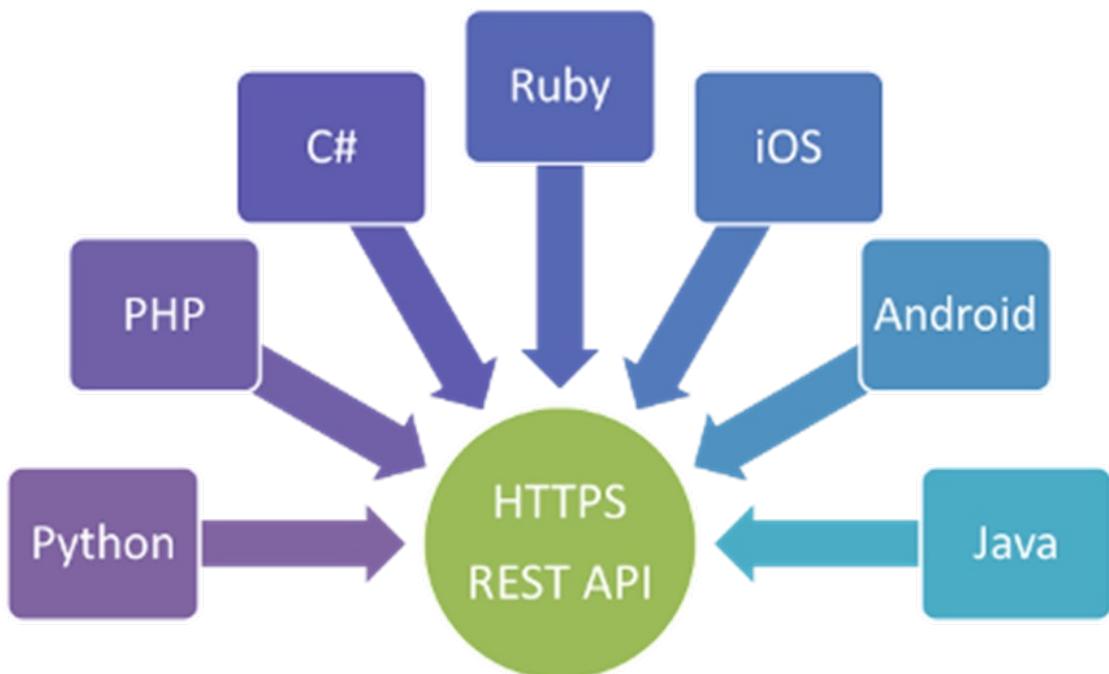
### แนะนำ aREST Framework

aREST เป็น framework ที่สามารถควบคุมการทำงานของ ESP32, ESP8266, Arduino, Raspberry หรืออื่นๆ ได้โดยตรงผ่านทาง Serial console ใน Arduino IDE หรือผ่านเครือข่าย IP

การทำงานจะใช้ API - JSON ในการแลกเปลี่ยนข้อมูล รองรับการทำงานแบบ Server และ Client ถ้าอยู่ในวงแ伦เดียวกัน ผู้ใช้สามารถใช้ web browser เข้าถึงบอร์ดและสั่งการผ่าน API ได้โดยตรง

### RESTful (Representational State Transfer)

คือ Web Service ที่ใช้ HTTP Method (GET, POST, PUT, DELETE) แลกเปลี่ยนข้อมูลในรูปแบบของ JSON หรือ XML



## RESTful (Representational State Transfer)

<https://maps.googleapis.com/maps/api/geocode/json?address=rangsit>

```

▼{
  ▼"results": [
    ▼{
      ▼"address_components": [
        ▼{
          "long_name": "Rangsit",
          "short_name": "Rangsit",
          ▼"types": [
            "locality",
            "political"
          ]
        },
        ▼{
          "long_name": "Prachathipat",
          "short_name": "Prachathipat",
          ▼"types": [
            "political",
            "sublocality",
            "sublocality_level_1"
          ]
        },
        ▼{
          "long_name": "Thanyaburi District",
          "short_name": "Thanyaburi District",
          ▼"types": [
            "administrative_area_level_2",
            "political"
          ]
        },
        ▼{
          "long_name": "Pathum Thani",
          "short_name": "จ.ปทุมธานี",
          ▼"types": [
            "administrative_area_level_1",
            "political"
          ]
        },
        ▼{
          "long_name": "Thanyaburi",
          "short_name": "Thanyaburi",
          ▼"types": [
            "locality"
          ]
        }
      ]
    }
  ]
}
  
```

## JSON (JavaScript Object Notation)

เจสันจัดเก็บในรูปแบบที่สามารถอ่านและเข้าใจได้ง่าย เปรียบเสมือนรูปแบบของ อาเรย์ (Array) ชนิดหนึ่งที่ใช้รับส่งแลกเปลี่ยนข้อมูลข้ามแพตฟอร์ม

```
{  
    "firstName": "รัชชานนท์",  
    "lastName": "เวียงนาค",  
    "address":  
    {  
        "City": "กรุงเทพฯ",  
        "Country": "ประเทศไทย",  
    },  
    "phoneNumbers":  
    [  
        "099-9999999",  
        "02-6690001"  
    ]  
}
```

## JSON Text to a JavaScript Object

```
1  <!DOCTYPE html>
2  <html>
3  <body>
4
5  <h2>Create Object from JSON String</h2>
6
7  <p id="demo"></p>
8
9  <script>
10
11 var text = '{"employees":[' +
12   '{"firstName":"John","lastName":"Doe"},' +
13   '{"firstName":"Anna","lastName":"Smith"},' +
14   '{"firstName":"Peter","lastName":"Jones"}]';
15
16 obj = JSON.parse(text);
17 document.getElementById("demo").innerHTML =
18 obj.employees[1].firstName + " " + obj.employees[1].lastName;
19
20 </script>
21
22 </body>
23 </html>
```

ผลลัพธ์ที่ได้ >>

### Create Object from JSON String

Anna Smith

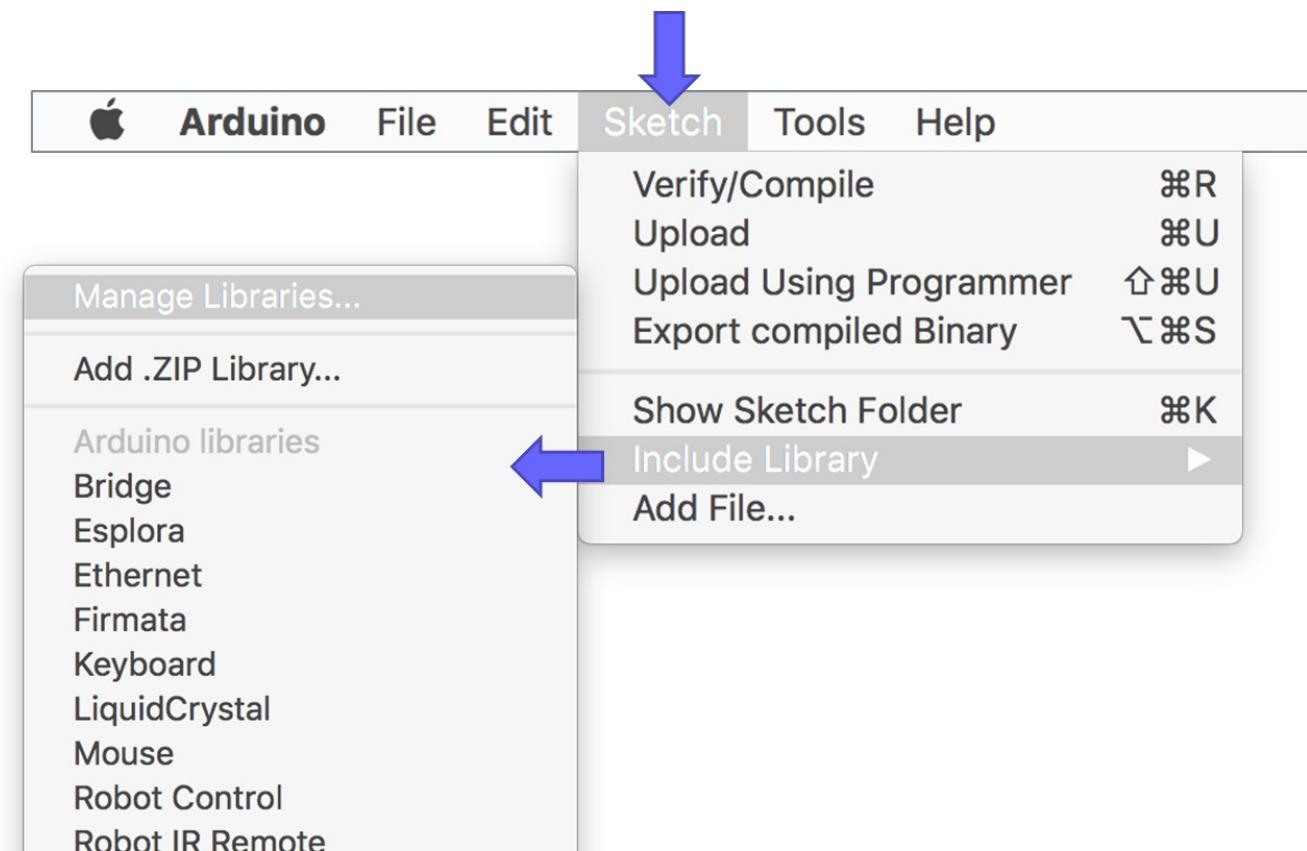
## JSON Text to a JavaScript Object

```
1 <?php  
2  
3 $myObj->name = "John";  
4 $myObj->age = 30;  
5 $myObj->city = "New York";  
6  
7 $myJSON = json_encode($myObj);  
8  
9 echo $myJSON;  
10  
11 ?>
```

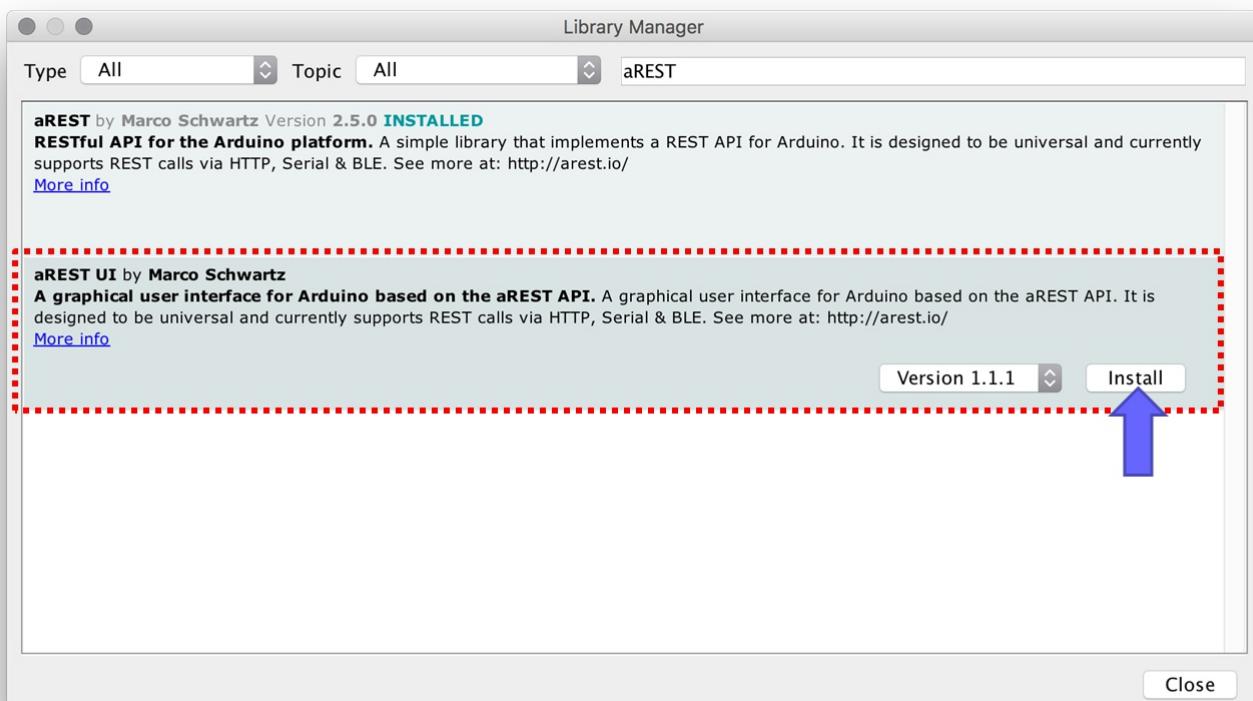
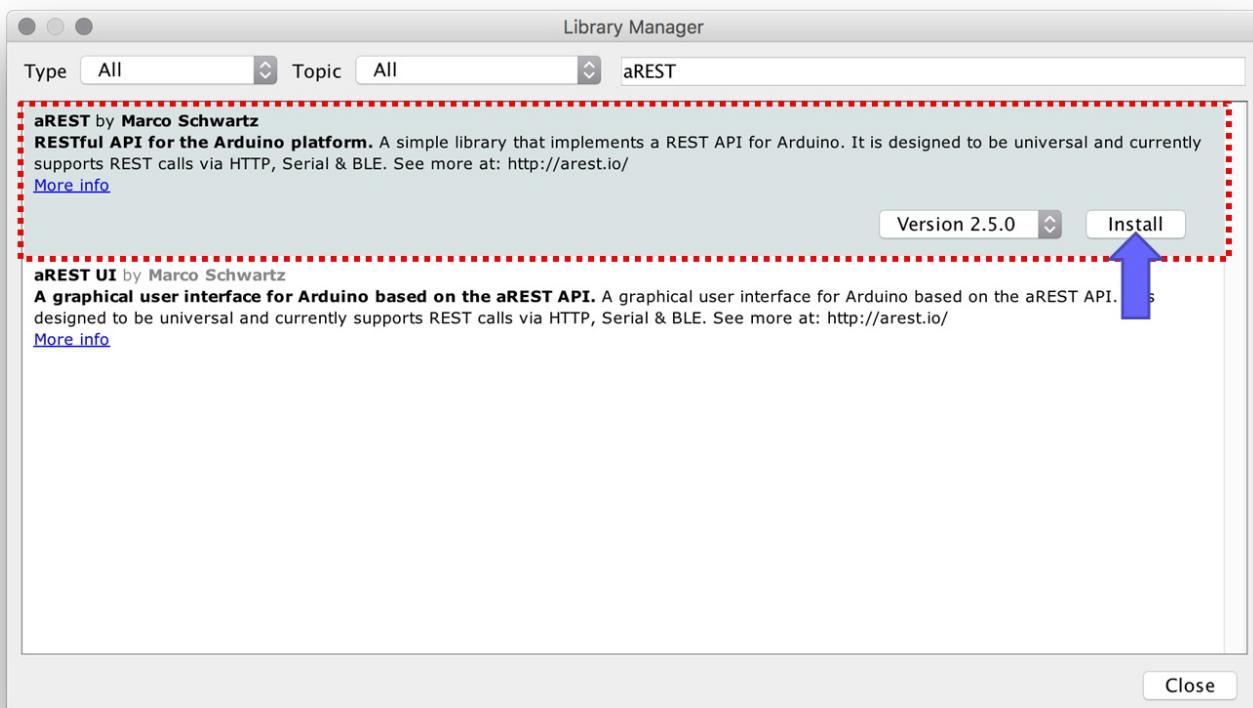
ผลลัพธ์ที่ได้ &gt;&gt;

{ "name": "John", "age": 30, "city": "New York" }

## ติดตั้ง aREST Library



## ติดตั้ง aREST Library



## EX14\_aREST\_Serial

ควบคุมบอร์ดผ่านทาง Serial โดยใช้ aREST Framework API เป็นตัวจัดการ

คำสั่งที่เกี่ยวข้อง

```
/* กำหนดให้ขา 25 เป็น OUTPUT */
```

```
/mode/25/o
```

```
/* กำหนดสถานะให้ขา 25 เป็น HIGH */
```

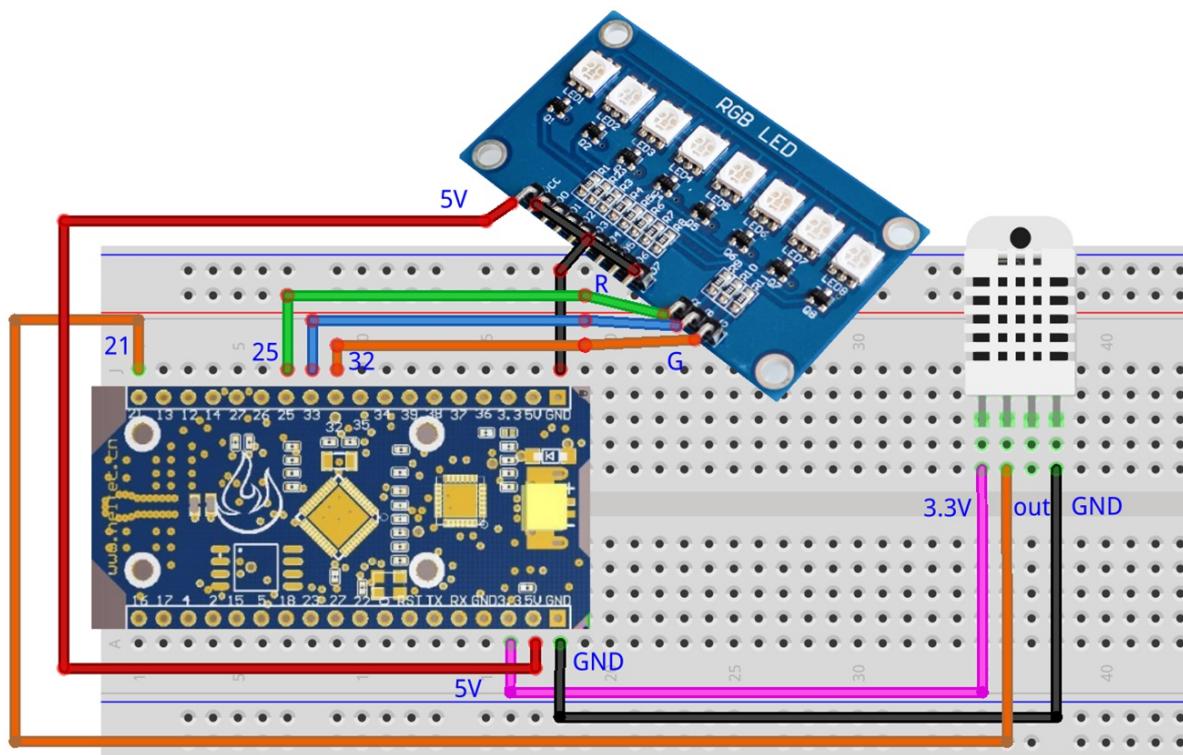
```
/digital/25/1
```

```
/* กำหนดสถานะให้ขา 25 เป็น LOW */
```

```
/digital/25/0
```

```
/* เช็คสถานะให้ขา 25 */
```

```
/digital/25
```



AREST Framework  
EX14\_aREST\_Serial

```
/* กำหนดให้ขา 25 เป็น OUTPUT */
/mode/25/o

/* กำหนดให้ขา 32 เป็น OUTPUT */
/mode/32/o

/* กำหนดให้ขา 33 เป็น OUTPUT */
/mode/33/o
```

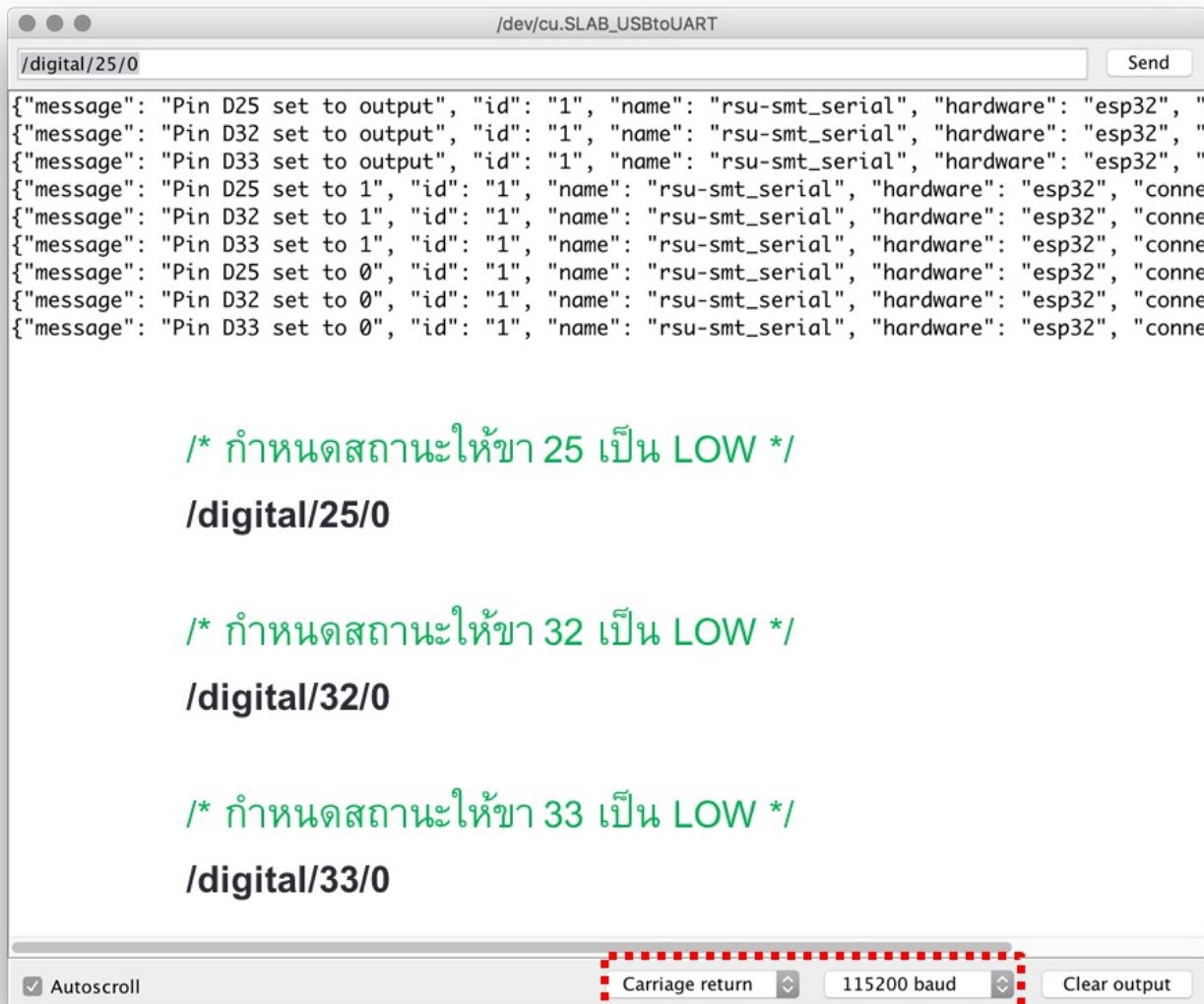
# AREST Framework

## EX14\_aREST\_Serial

```
/* กำหนดสถานะขา 25 เป็น HIGH */
/digital/25/1

/* กำหนดสถานะขา 32 เป็น HIGH */
/digital/32/1

/* กำหนดสถานะขา 33 เป็น HIGH */
/digital/33/1
```



```
/* กำหนดสถานะขา 25 เป็น LOW */
/digital/25/0

/* กำหนดสถานะขา 32 เป็น LOW */
/digital/32/0

/* กำหนดสถานะขา 33 เป็น LOW */
/digital/33/0
```

## AREST Framework EX14\_aREST\_Serial

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

- Title Bar:** 14\_aREST\_Serial | Arduino 1.8.5
- Code Editor:** The code for "14\_aREST\_Serial" is displayed. It includes comments in Thai and English, and defines an aREST instance named "rest" which handles Serial communication.
- Serial Monitor:** Shows the message "Done uploading." followed by "Leaving..." and "Hard resetting...".
- Status Bar:** Displays the device information: Heltec\_WIFI\_LoRa\_32, 80MHz, 921600 on /dev/cu.SLAB\_USBtoUART.

```
/dev/cu.SLAB_USBtoUART
/digital/25
Send

{"message": "Pin D25 set to output", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "con
 {"message": "Pin D32 set to output", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "con
 {"message": "Pin D33 set to output", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "con
 {"message": "Pin D25 set to 1", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connecte
 {"message": "Pin D32 set to 1", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connecte
 {"message": "Pin D33 set to 1", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connecte
 {"message": "Pin D25 set to 0", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connecte
 {"message": "Pin D32 set to 0", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connecte
 {"message": "Pin D33 set to 0", "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connecte
 {"return_value": 0, "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connected": true}
 {"return_value": 0, "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connected": true}
 {"return_value": 0, "id": "1", "name": "rsu-smt_serial", "hardware": "esp32", "connected": true}

/* เช็คสถานะให้ขา 25 */

/digital/25/0

/* เช็คสถานะให้ขา 32 เป็น */

/digital/32/0

/* เช็คสถานะให้ขา 33 */

/digital/33/0
```

Autoscroll      Carriage return      115200 baud      Clear output

## API documentation

The API currently supports five type of commands: digital, analog, and mode, variables, and user-defined functions.

### Digital

Digital is to write or read on digital pins on the Arduino. For example:

```
/* sets pin number 25 to a low state */
```

```
/digital/25/0
```

```
/* sets pin number 25 to a high state */
```

```
/digital/25/1
```

```
/* reads value from pin number 25 in JSON format (note that for compatibility reasons, /  
digital/25/r produces the same result) */
```

```
/digital/25
```

The API currently supports five type of commands: digital, analog, and mode, variables, and user-defined functions.

## Analog

Analog is to write or read on analog pins on the Arduino. Note that you can only write on PWM pins for the Arduino Uno, and only read analog values from analog pins 0 to 5. For example:

```
/* sets pin number 6 to 123 using PWM */  
/* ESP32 ยังไม่รองรับคำสั่ง AnalogWrite */  
/analog/6/123
```

```
/* returns analog value from pin number A0 in JSON format (note that for compatibility  
reasons, /analog/0/r produces the same result) */  
/analog/0
```

The API currently supports five type of commands: digital, analog, and mode, variables, and user-defined functions.

## Mode

Mode is to change the mode on a pin. For example:

```
/* sets pin number 32 as an output */  
/mode/32/o
```

```
/* sets pin number 32 as an input */  
/mode/32/i
```

The API currently supports five type of commands: digital, analog, and mode, variables, and user-defined functions.

## Variables

You can also directly call variables that are defined in your sketch. Integer variables are supported by the library.

To access a variable in your sketch, you have to declare it first, and then call it from with a REST call. For example, if your aREST instance is called "rest" and the variable "temperature":

```
/* declares the temperature in the Arduino sketch */  
rest.variable("temperature", &temperature);  
  
/* returns the value of the variable in JSON format */  
/temperature
```

The API currently supports five type of commands: digital, analog, and mode, variables, and user-defined functions.

### Get data about the board

You can also access a description of all the variables that were declared on the board with a single command. This is useful to automatically build graphical interfaces based on the variables exposed to the API. This can be done via the following calls:

**/ or /id**

The names & types of the variables will then be stored in the variables field of the returned JSON object

## EX15\_aREST\_WiFi\_API

ควบคุมบอร์ดผ่านทาง IP โดยเชื่อมต่อวายฟายและใช้ aREST Framework API เป็นตัวจัดการ

### ตัวอย่างคำสั่ง

<http://192.168.1.108/digital/25/1>

<http://192.168.1.108/digital/32/1>

<http://192.168.1.108/digital/33/1>

<http://192.168.1.108>

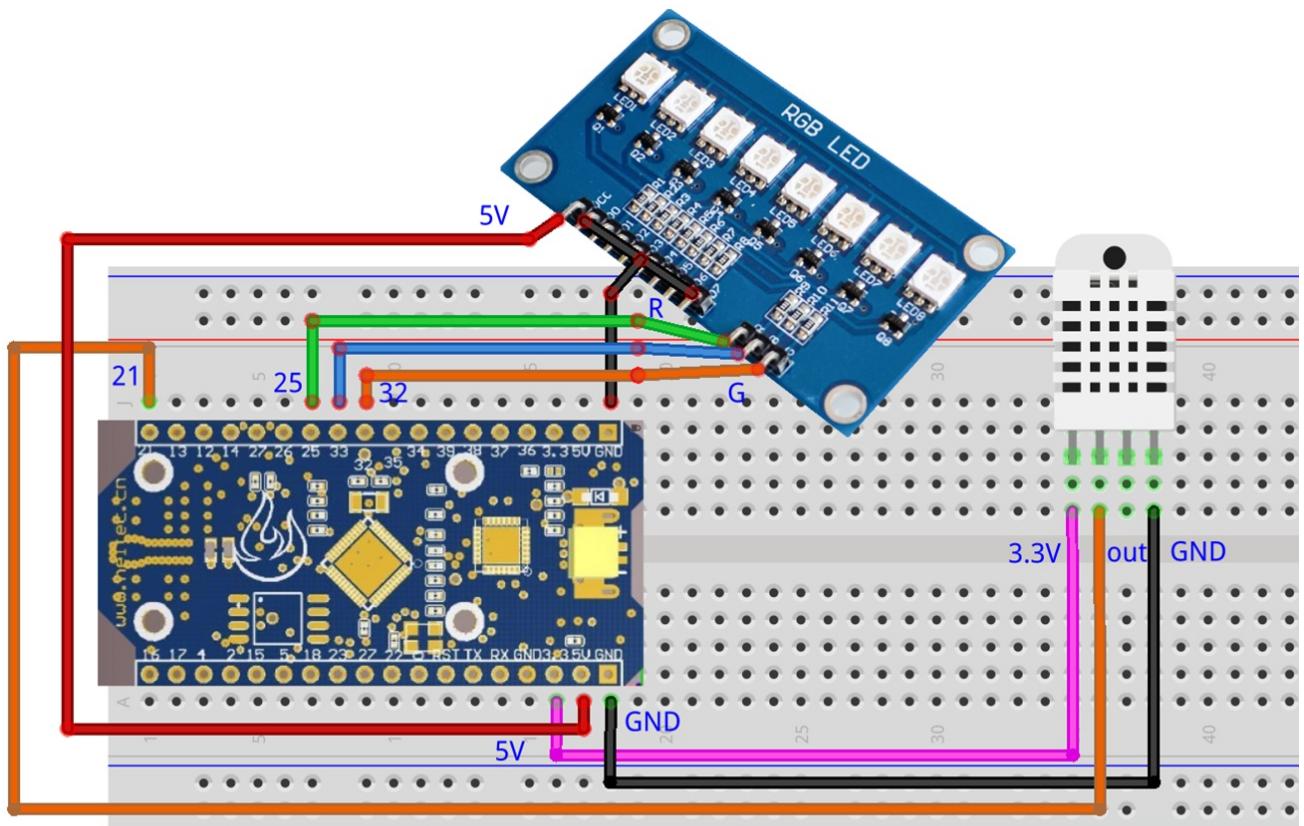
<http://192.168.1.108/temperature>

<http://192.168.1.108/humidity>

<http://192.168.1.108/red?params=5>

<http://192.168.1.108/blue?params=5>

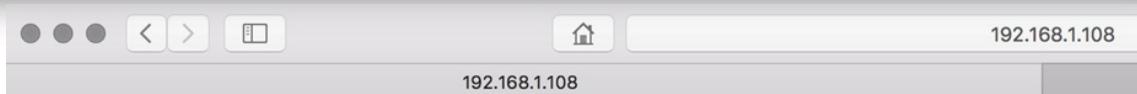
<http://192.168.1.108/green?params=5>



# AREST Framework

## EX15\_aREST\_WiFi\_API

```
/*  
 * EX15_aREST WIFI API  
 * គាលកុំណរត់ ESP32 ទូទាត់ទំនាក់ទំនង IP ដោយ aREST API  
 */  
  
// Import required libraries  
#include "WiFi.h"  
#include <aREST.h>  
  
#include <Wire.h>  
#include "SSD1306.h"  
SSD1306 display(0x3c, 4, 15);  
  
//-----//  
  
#define DHTTYPE DHT22  
#define DHTPIN 21  
  
#include <SimpleDHT.h> //ໂលតិតាហារី DHT22  
SimpleDHT22 DHTTYPE; //របៀបរួមទិន្នន័យទូទាត់ DHT22  
float temperature = 0; //ការណែនាំតុរោគកំណត់តុរោគ  
float humidity = 0; //ការណែនាំអេឡិចត្រូនិកការពារការពារ  
  
//-----//  
  
// Create aREST instance  
aREST rest = aREST();  
  
Done Saving.  
Leaving...  
Hard resetting...  
  
19 Heltec_WIFI_LoRa_32, 80MHz, 921600 on /dev/cu.SLAB_USBtoUART
```



```
▼ {  
    "connected": true,  
    "hardware": "esp32",  
    "id": "1",  
    "name": "rsusmt_esp32",  
    ▼ "variables": {  
        "humidity": 56.8,  
        "temperature": 26.1  
    }  
}
```

AREST Framework  
EX15\_aREST\_WiFi\_API

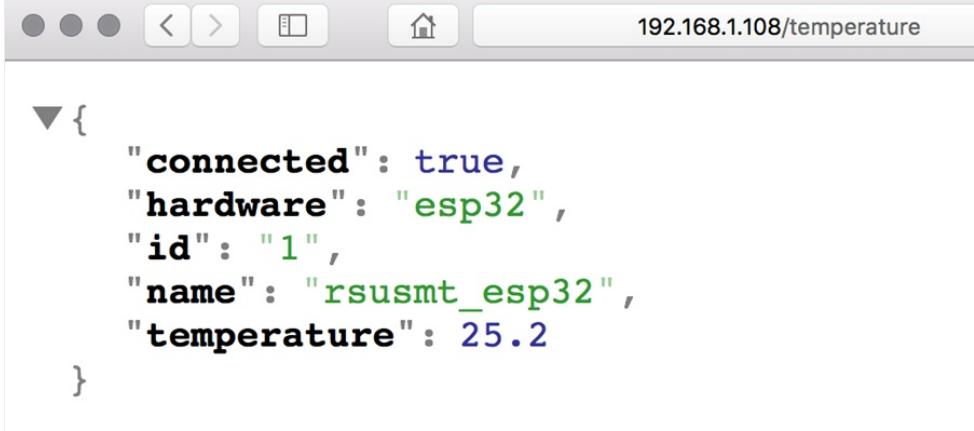
```
192.168.1.108/digital/25/1
▼ {
  "connected": true,
  "hardware": "esp32",
  "id": "1",
  "message": "Pin D25 set to 1",
  "name": "rsusmt_esp32"
}

192.168.1.108/digital/32/1
▼ {
  "connected": true,
  "hardware": "esp32",
  "id": "1",
  "message": "Pin D32 set to 1",
  "name": "rsusmt_esp32"
}

192.168.1.108/digital/33/1
▼ {
  "connected": true,
  "hardware": "esp32",
  "id": "1",
  "message": "Pin D33 set to 1",
  "name": "rsusmt_esp32"
}

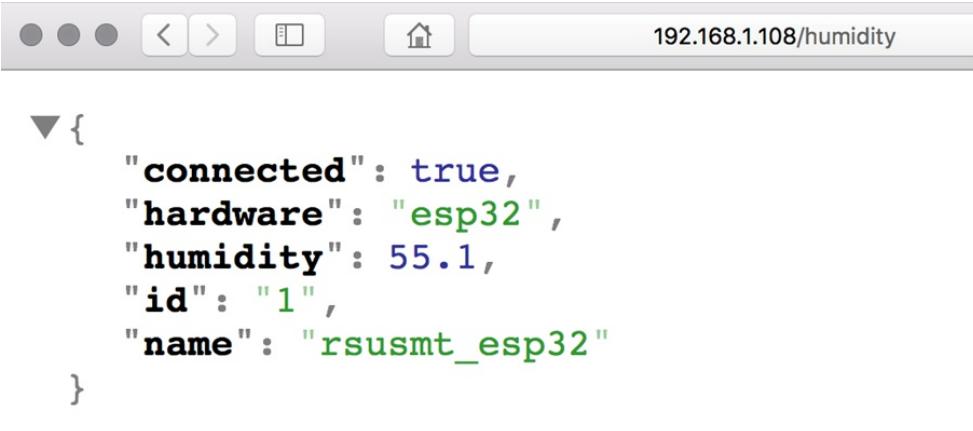
192.168.1.108/id
▼ {
  "connected": true,
  "hardware": "esp32",
  "id": "1",
  "name": "rsusmt_esp32"
}
```

AREST Framework  
EX15\_aREST\_WiFi\_API



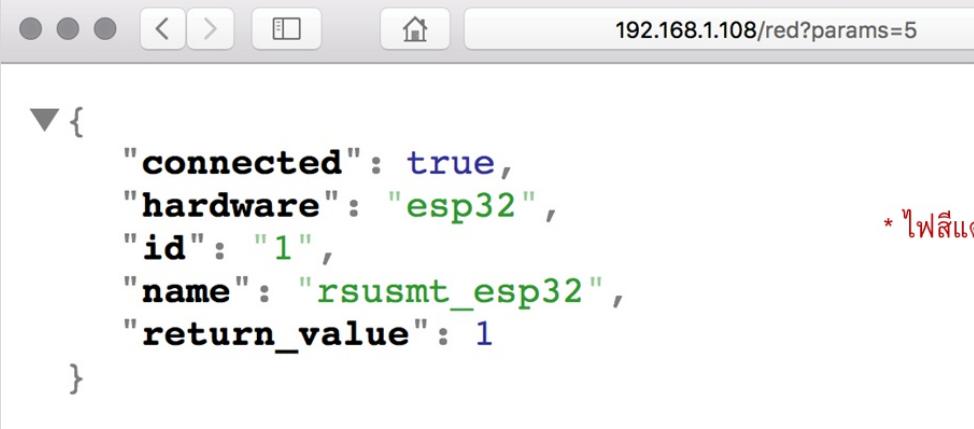
192.168.1.108/temperature

```
▼ {  
  "connected": true,  
  "hardware": "esp32",  
  "id": "1",  
  "name": "rsusmt_esp32",  
  "temperature": 25.2  
}
```



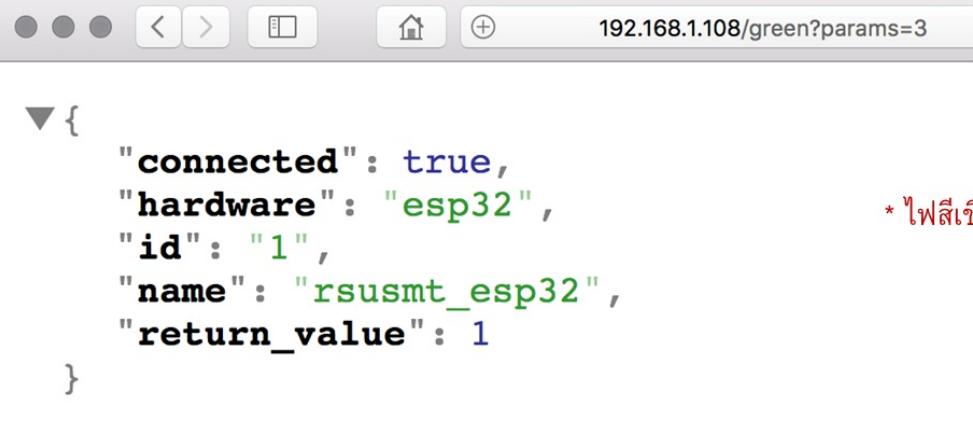
192.168.1.108/humidity

```
▼ {  
  "connected": true,  
  "hardware": "esp32",  
  "humidity": 55.1,  
  "id": "1",  
  "name": "rsusmt_esp32"  
}
```



192.168.1.108/red?params=5

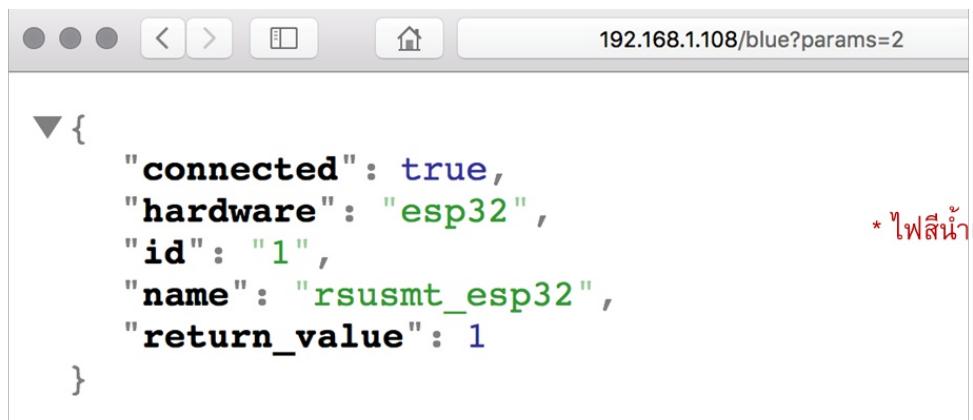
```
▼ {  
  "connected": true,  
  "hardware": "esp32",  
  "id": "1",  
  "name": "rsusmt_esp32",  
  "return_value": 1  
} *ไฟสีแดงกระพริบ 5 ครั้ง
```



192.168.1.108/green?params=3

```
▼ {  
  "connected": true,  
  "hardware": "esp32",  
  "id": "1",  
  "name": "rsusmt_esp32",  
  "return_value": 1  
} *ไฟสีเขียวกระพริบ 3 ครั้ง
```

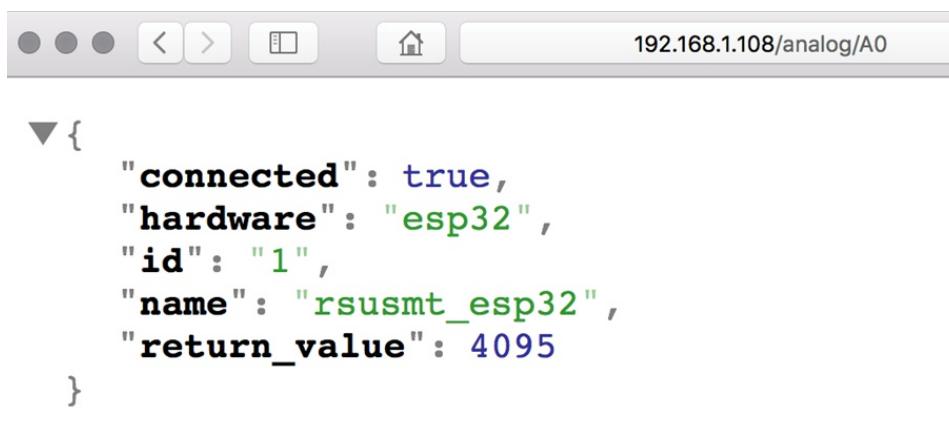
AREST Framework  
EX15\_aREST\_WiFi\_API



192.168.1.108/blue?params=2

```
▼ {
  "connected": true,
  "hardware": "esp32",
  "id": "1",
  "name": "rsusmt_esp32",
  "return_value": 1
}
```

\* ไฟสีน้ำเงินกระพริบ 2 ครั้ง

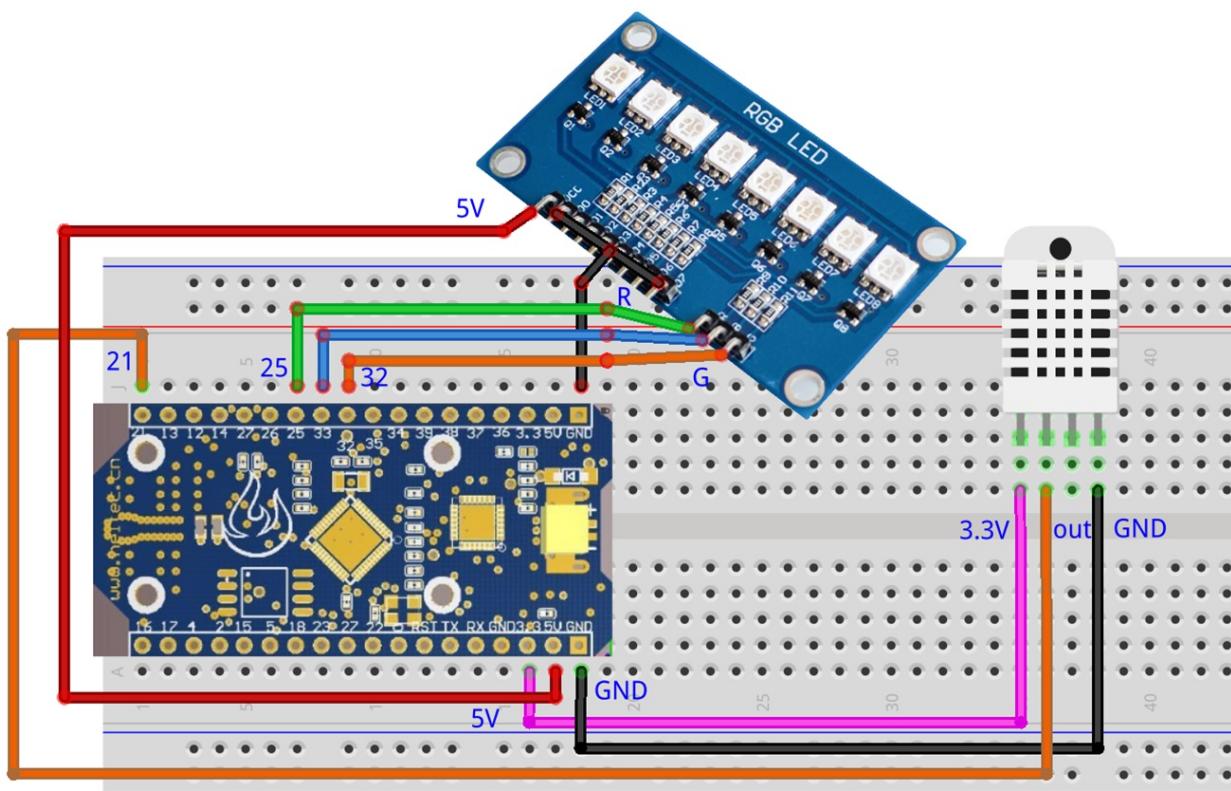
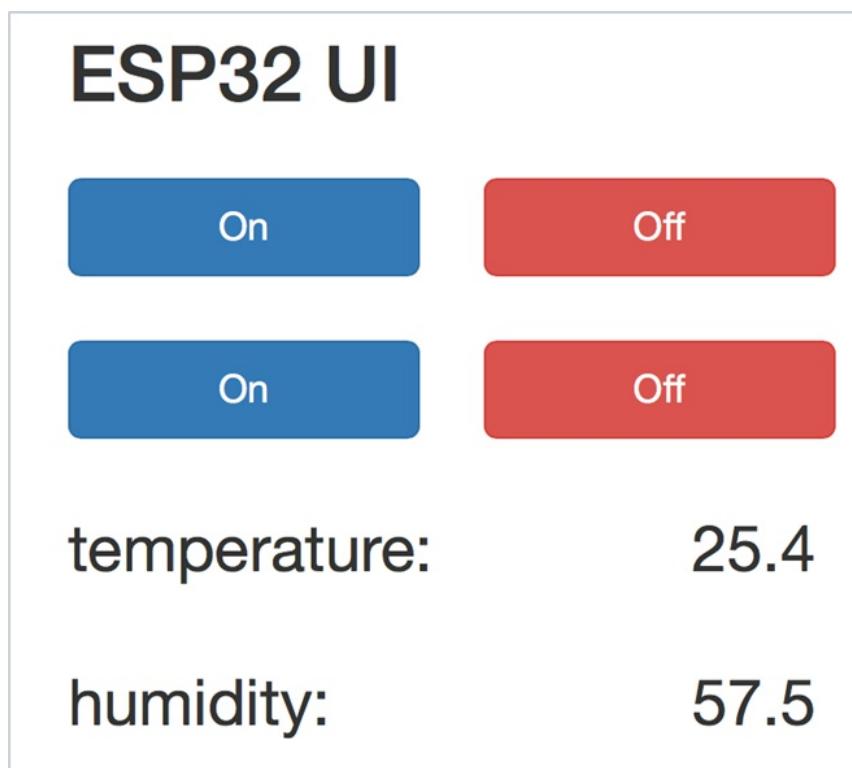


192.168.1.108/analog/A0

```
▼ {
  "connected": true,
  "hardware": "esp32",
  "id": "1",
  "name": "rsusmt_esp32",
  "return_value": 4095
}
```

## EX16\_aREST\_WiFi\_UI\_API

ควบคุมบอร์ดผ่านทาง IP โดยเชื่อมต่อวายฟายและใช้ aREST Framework API เป็นตัวจัดการ แสดงผล user interface บน web browser

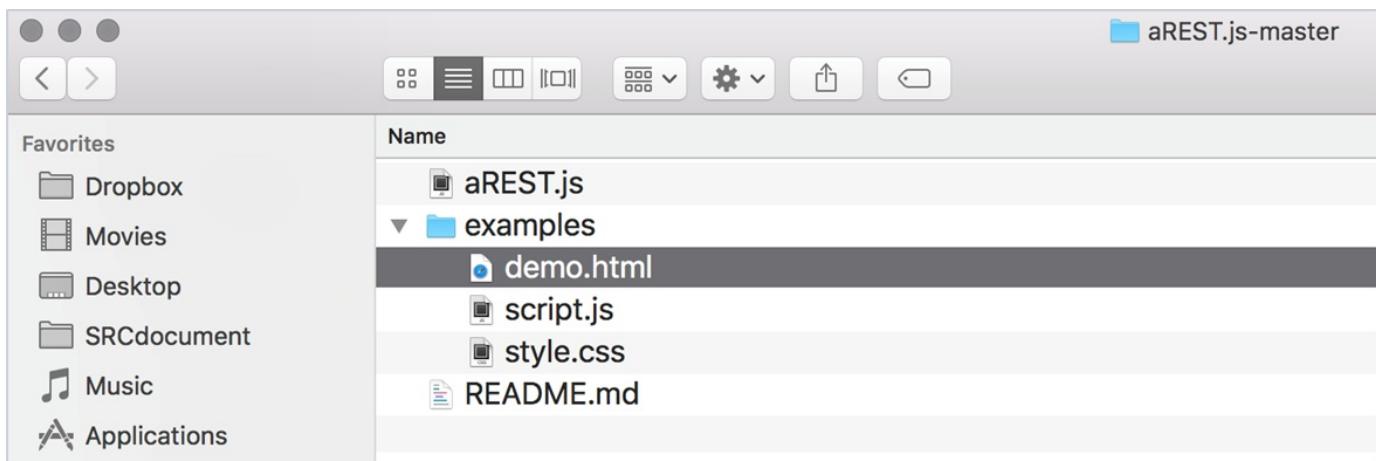


## AREST Framework EX16\_aREST\_WiFi\_API

```
/*  
 * EX2_MultiTasking_Milli_LED  
 * ควบคุมบอร์ด ESP32 โดยตรงผ่าน IP ด้วย aREST API  
 */  
  
// Import required libraries  
#include "WiFi.h"  
#include <aREST.h>  
  
#include <Wire.h>  
#include "SSD1306.h"  
SSD1306 display(0x3c, 4, 15);  
  
///-----///  
  
#define DHTTYPE dht11  
#define DHTPIN 21  
  
#include <SimpleDHT.h>          //ไฟล์ไลบรารี DHT11  
SimpleDHT11 DHTTYPE;           //ระบุรุ่นเซ็นเซอร์รุ่น DHT11  
byte temperature = 0;          //กำหนดตัวแปรเก็บค่าอุณหภูมิ  
byte humidity = 0;             //กำหนดตัวแปรเก็บค่าความชื้นสัมสัทธิ์  
int t;  
int h;  
  
///-----///  
  
Done Saving.  
  
Writing at 0x00008000... (100 %)  
Wrote 3072 bytes (121 compressed) at 0x00008000 in 0.0 seconds (effective 2377.2 kbit/s)  
26
```

## EX17\_aREST\_WiFi\_UI\_WEB\_API

ควบคุมบอร์ดผ่านทาง IP โดยเชื่อมต่อวายฟายและใช้ aREST Framework API เป็นตัวจัดการ แสดงผล user interface บน web browser

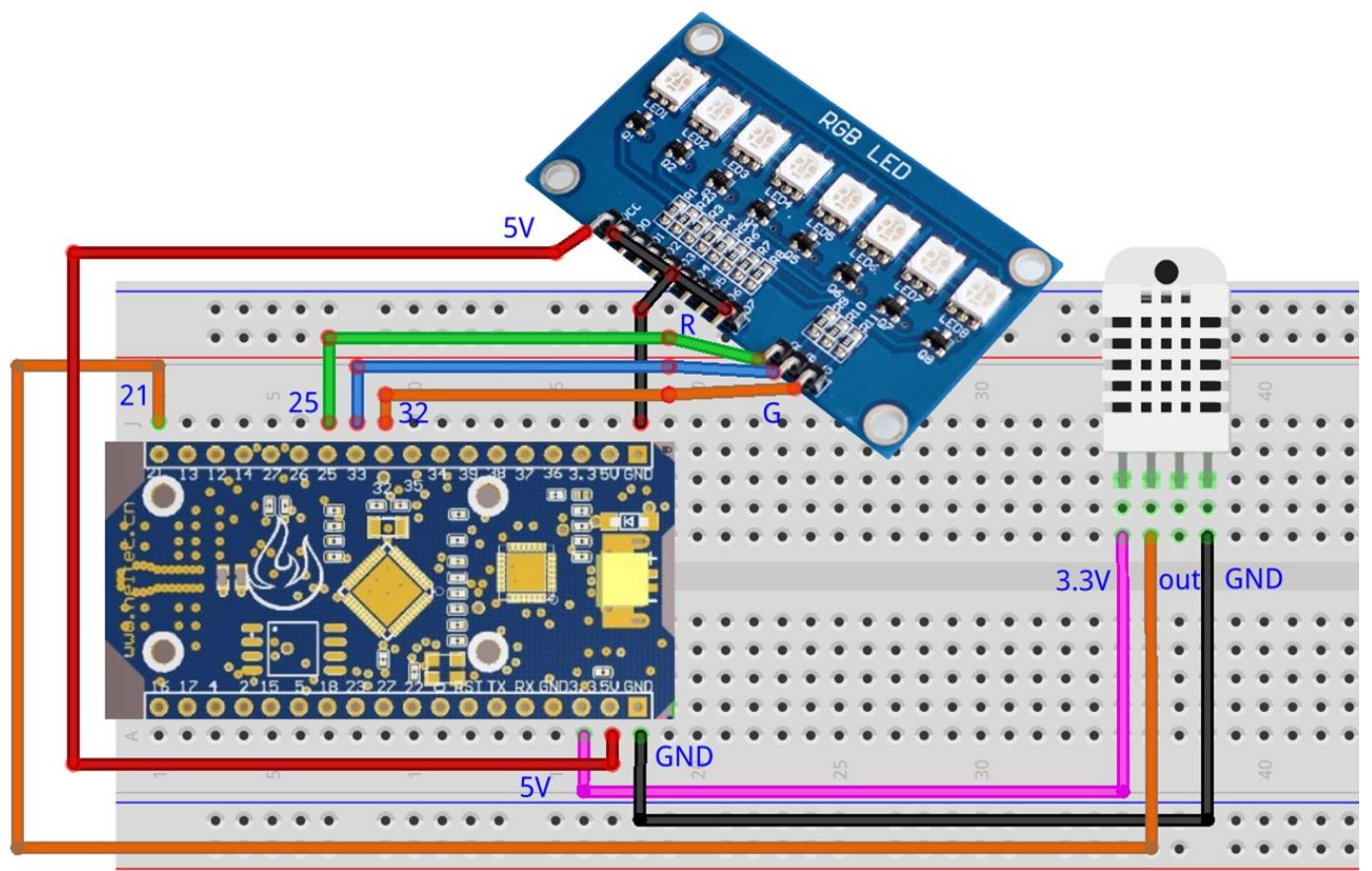


## ESP32 - WebControl

Device IP address:  Connect

RED (PIN25)	<button>On</button>	<button>Off</button>
GREEN (PIN32):	<button>On</button>	<button>Off</button>
BLUE (PIN33):	<button>On</button>	<button>Off</button>
RED Blink:	<button>3 Times</button>	<button>5 Times</button>
Analog pin A0: 4095	Digital pin 25: 1	
Temperature: 25.4	Humidity: 55.3	

AREST Framework  
EX17\_aREST\_WiFi\_UI\_WEB\_API



## บทที่ 4 Blynk Cloud Server

- ติดตั้ง Blynk Cloud Server บน digitalocean
  - สร้าง firewall ให้ Blynk Server

# Blynk Cloud Server

ติดตั้ง บน digitalocean



## ขั้นตอนการสร้าง Blynk Cloud บน DigitalOcean

1. ติดตั้ง Linux Server บน DigitalOcean
2. ติดตั้ง JAVA บน UBUNTU server
3. ติดตั้ง Blynk Server
4. ตั้งค่าคอนฟิก Blynk Server
5. ทดสอบเชื่อมต่อ Blynk Server
6. ทดสอบเข้าสู่หน้าจัดการ Administrator
7. สร้าง firewall ให้ Blynk Server

## Blynk could server

### 1. ติดตั้ง Linux Server บน DigitalOcean

The screenshot shows the DigitalOcean Droplets dashboard. At the top, there are navigation links: Droplets, Spaces, Images, Networking, Monitoring, API, and Support. A green 'Create' button is on the right. On the left, a sidebar has a 'Create' button with a dropdown menu containing options like 'Droplets', 'Volumes', 'Domains/DNS', 'Cloud Firewalls', 'Floating IPs', 'Load Balancers', 'Alert Policies', and 'Spaces'. The main area is titled 'Droplets' and shows a table with four rows of server details. The columns are 'Name', 'IP Address', 'Created', and 'Tags'. The first row's 'Name' column shows a thumbnail of a server icon with the text 'BlynkServer'. The second row's 'Name' column shows a thumbnail of a server icon with the text 'BlynkServer2'. The third row's 'Name' column shows a thumbnail of a server icon with the text 'BlynkServer3'. The fourth row's 'Name' column shows a thumbnail of a server icon with the text 'BlynkServer4'. The 'Created' column shows dates: '4 months ago', '10 months ago', '1 year ago', and '1 year ago'. The 'IP Address' column shows IP addresses: '52.199.204.107', '52.199.107.41', '52.199.172.249', and '52.199.246.173'. The 'Tags' column is empty for all rows.

The screenshot shows the 'Create Droplets' wizard, Step 1: Choose an image. The top navigation bar includes 'Distributions' and 'One-click apps'. Below, there are six boxes for different operating systems: Ubuntu, FreeBSD, Fedora, Debian, CoreOS, and CentOS. The 'Ubuntu' box is highlighted with a red dashed border and shows the version '16.04.3 x64' with a dropdown arrow. The other five boxes show their respective icons and names: 'FreeBSD', 'Fedora', 'Debian', 'CoreOS', and 'CentOS', each with a 'Select version' dropdown arrow below them.

## Blynk could server

### ຕິດຕັ້ງ Linux Ubuntu Server ບນ DigitalOcean

Choose a size

Standard   High Memory   High CPU

Enough RAM, CPU, and storage space needed to get applications off the ground.

\$5/mo \$0.007/hour  512 MB / 1 CPU 20 GB SSD disk 1000 GB transfer	\$10/mo \$0.015/hour  1GB / 1 CPU 30 GB SSD disk 2 TB transfer	\$20/mo \$0.030/hour  2 GB / 2 CPUs 40 GB SSD disk 3 TB transfer	\$40/mo \$0.060/hour  4 GB / 2 CPUs 60 GB SSD disk 4 TB transfer	\$80/mo \$0.119/hour  8 GB / 4 CPUs 80 GB SSD disk 5 TB transfer	\$160/mo \$0.238/hour  16 GB / 8 CPUs 160 GB SSD disk 6 TB transfer
\$320/mo \$0.476/hour  32 GB / 12 CPUs 320 GB SSD disk 7 TB transfer	\$480/mo \$0.714/hour  48 GB / 16 CPUs 480 GB SSD disk 8 TB transfer	\$640/mo \$0.952/hour  64 GB / 20 CPUs 640 GB SSD disk 9 TB transfer			

Droplets   Spaces   Images   Networking   Monitoring   API   Support

Create Droplets

Choose an image ?

Distributions   One-click apps

Ubuntu  16.04.3 x64 Select version	FreeBSD Select version	Fedor Select version	Debian Select version	CoreOS Select version	CentOS Select version
---	---------------------------	-------------------------	--------------------------	--------------------------	--------------------------

Choose a size

## Blynk could server

### ຕິດຕັ້ງ Linux Ubuntu Server ບນ DigitalOcean

The screenshot shows the 'Add block storage' section of the DigitalOcean interface. At the top, there is a message: 'Currently only available in BLR1, FRA1, LON1, NYC1, NYC3, SFO2, SGP1 and TOR1.' Below this is a 'Add Volume' button. The main area is titled 'Choose a datacenter region' and displays a grid of datacenter options:

New York	San Francisco	Amsterdam	Singapore	London	Frankfurt
1 2 3	1 2	2 3	1	1	1
Toronto	Bangalore				
1	1				

The 'Singapore' datacenter is highlighted with a red dashed border.

The screenshot shows the 'Select additional options' and 'Finalize and create' sections of the DigitalOcean interface.

**Select additional options:** This section includes checkboxes for Private networking, Backups, IPv6, User data, and Monitoring.

**Add your SSH keys:** This section includes a 'New SSH Key' button and a checkbox for 'My MacBook'.

**Finalize and create:** This section includes fields for 'How many Droplets?' (set to 1) and 'Choose a hostname' (set to 'rsusmtBlynkServer'). It also has an 'Add Tags' button and a large green 'Create' button with a purple arrow pointing right.

## Blynk could server

### ຕິດຕັ້ງ Linux Ubuntu Server ບນ DigitalOcean

DigitalOcean Droplets interface showing a list of servers. The first server, 'rsusmtBlynkServer', is highlighted with a blue selection bar. It has an IP address of [REDACTED] and was created 4 months ago.

Name	IP Address	Created	Tags
rsusmtBlynkServer	[REDACTED]	4 months ago	
[REDACTED]	[REDACTED]	10 months ago	
[REDACTED]	[REDACTED]	1 year ago	
[REDACTED]	[REDACTED]	1 year ago	

DigitalOcean Droplets interface showing a list of servers. The first server, 'rsusmtBlynkServer', is highlighted with a red dashed box around its row. It has an IP address of 128.199.117.92 and the tag 'Go change the world!'. The rest of the list remains the same as the previous screenshot.

Name	IP Address	Created	Tags
rsusmtBlynkServer	128.199.117.92	4 months ago	Go change the world!
[REDACTED]	[REDACTED]	10 months ago	
[REDACTED]	[REDACTED]	1 year ago	
[REDACTED]	[REDACTED]	1 year ago	

Blynk could server

## ติดตั้ง Linux Ubuntu Server บน DigitalOcean

Your New Droplet: rsusmtBlynkServer ก่อตั้งจดหมาย X

DigitalOcean <support@support.digitalocean.com>  
ดึง ฉัน ▾

Your new Droplet is all set to go! You can access it using the following credentials:

Droplet Name: rsusmtBlynkServer  
IP Address: 128.199.117.92  
Username: root  
Password: 5eadfd268f172a69a5f8bddf82

For security reasons, you will be required to change this Droplet's root password when you login. You should choose a password that is easy for you to remember, but hard for a computer to guess. You might try creating an alpha-numerical phrase from a childhood memory ("I was in a spelling bee at age 7," might become "lwm#1sbaa7"). Random strings of common words, such as "Mousetrap Sausage", are also good choices, as well, too.

As an added security measure, we also strongly recommend adding an SSH key to your account. You can do this by visiting [https://cloud.digitalocean.com/settings/security?i=c39171](#)

Once added, you can select your SSH key and use it when creating future Droplets. This eliminates the need for you to remember the password for your Droplets much less vulnerable to attack.

Happy Coding,  
Team DigitalOcean

## Upgrade OS: sudo apt-get update

```
visit01 — root@rsusmtBlynkServer: ~ — ssh — 115x32
* Documentation: https://help.ubuntu.com
* Management: https://landscape.canonical.com
* Support: https://ubuntu.com/advantage

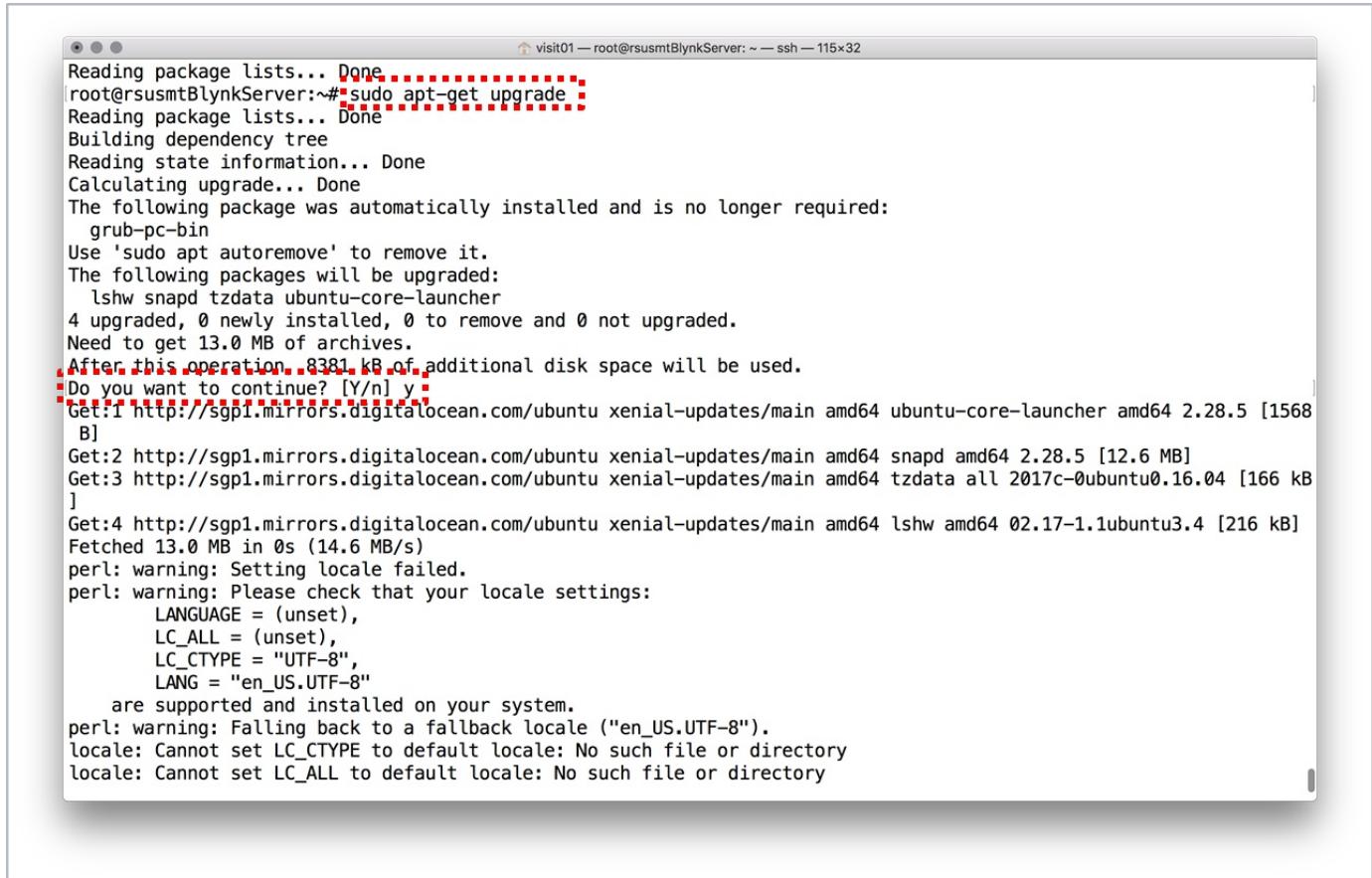
Get cloud support with Ubuntu Advantage Cloud Guest:
http://www.ubuntu.com/business/services/cloud

0 packages can be updated.
0 updates are security updates.

Last login: Mon Nov 13 02:15:48 2017 from 182.232.237.239
root@rsusmtBlynkServer:~# sudo apt-get update
Get:1 http://security.ubuntu.com/ubuntu xenial-security InRelease [102 kB]
Hit:2 http://sgp1.mirrors.digitalocean.com/ubuntu xenial InRelease
Get:3 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates InRelease [102 kB]
Get:4 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-backports InRelease [102 kB]
Get:5 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/main Sources [868 kB]
Get:6 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/restricted Sources [4808 B]
Get:7 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/universe Sources [7728 kB]
Get:8 http://security.ubuntu.com/ubuntu xenial-security/main Sources [99.4 kB]
Get:9 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/multiverse Sources [179 kB]
Get:10 http://security.ubuntu.com/ubuntu xenial-security/restricted Sources [2600 B]
Get:11 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/universe amd64 Packages [7532 kB]
Get:12 http://security.ubuntu.com/ubuntu xenial-security/universe Sources [44.3 kB]
Get:14 http://security.ubuntu.com/ubuntu xenial-security/multiverse Sources [1140 B]
Get:16 http://security.ubuntu.com/ubuntu xenial-security/main amd64 Packages [382 kB]
Get:13 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/universe Translation-en [4354 kB]
Get:15 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/multiverse amd64 Packages [144 kB]
Get:17 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/multiverse Translation-en [106 kB]
Get:18 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates/main Sources [280 kB]
83% [7 Sources store 0 B] [Waiting for headers] [Waiting for headers]
```

## ติดตั้ง Linux Ubuntu Server บน DigitalOcean

### Upgrade OS: sudo apt-get upgrade

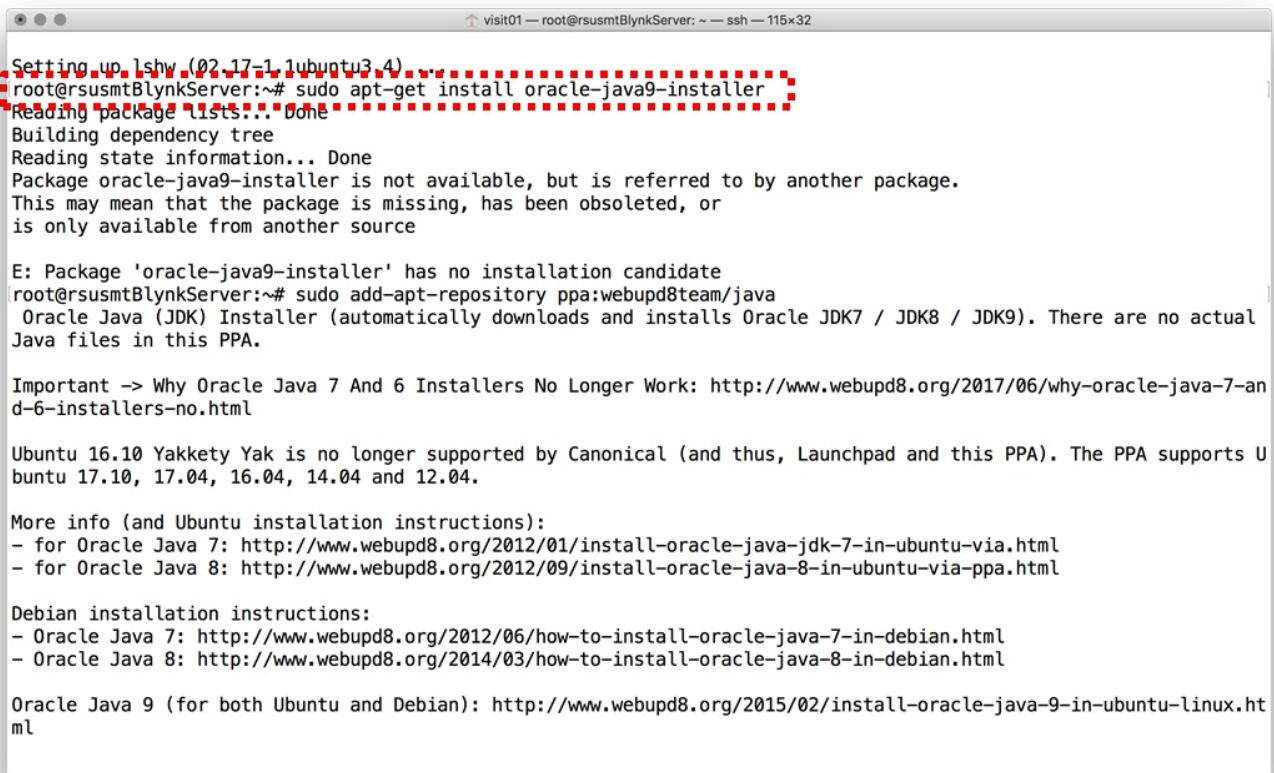


```
visit01 — root@rsusmtBlynkServer: ~ — ssh — 115x32
Reading package lists... Done
root@rsusmtBlynkServer:~# sudo apt-get upgrade
Reading package lists... Done
Building dependency tree
Reading state information... Done
Calculating upgrade... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following packages will be upgraded:
  lshw snapd tzdata ubuntu-core-launcher
4 upgraded, 0 newly installed, 0 to remove and 0 not upgraded.
Need to get 13.0 MB of archives.
After this operation, 8381 kB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates/main amd64 ubuntu-core-launcher amd64 2.28.5 [1568 B]
Get:2 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates/main amd64 snapd amd64 2.28.5 [12.6 MB]
Get:3 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates/main amd64 tzdata all 2017c-0ubuntu0.16.04 [166 kB]
Get:4 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates/main amd64 lshw amd64 02.17-1.1ubuntu3.4 [216 kB]
Fetched 13.0 MB in 0s (14.6 MB/s)
perl: warning: Setting locale failed.
perl: warning: Please check that your locale settings:
  LANGUAGE = (unset),
  LC_ALL = (unset),
  LC_CTYPE = "UTF-8",
  LANG = "en_US.UTF-8"
are supported and installed on your system.
perl: warning: Falling back to a fallback locale ("en_US.UTF-8").
locale: Cannot set LC_CTYPE to default locale: No such file or directory
locale: Cannot set LC_ALL to default locale: No such file or directory
```

## 2. ติดตั้ง JAVA บน UBUNTU Server

Installing the Oracle JDK 9 :

```
sudo add-apt-repository ppa:webupd8team/java
```



```
visit01 — root@rsusmtBlynkServer: ~ — ssh — 115x32
Setting up lshw (02.17-1.1ubuntu3.4)
root@rsusmtBlynkServer:~# sudo apt-get install oracle-java9-installer
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package oracle-java9-installer is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source

E: Package 'oracle-java9-installer' has no installation candidate
root@rsusmtBlynkServer:~# sudo add-apt-repository ppa:webupd8team/java
  Oracle Java (JDK) Installer (automatically downloads and installs Oracle JDK7 / JDK8 / JDK9). There are no actual Java files in this PPA.

Important -> Why Oracle Java 7 And 6 Installers No Longer Work: http://www.webupd8.org/2017/06/why-oracle-java-7-and-6-installers-no.html

Ubuntu 16.10 Yakkety Yak is no longer supported by Canonical (and thus, Launchpad and this PPA). The PPA supports Ubuntu 17.10, 17.04, 16.04, 14.04 and 12.04.

More info (and Ubuntu installation instructions):
- for Oracle Java 7: http://www.webupd8.org/2012/01/install-oracle-java-jdk-7-in-ubuntu-via.html
- for Oracle Java 8: http://www.webupd8.org/2012/09/install-oracle-java-8-in-ubuntu-via-ppa.html

Debian installation instructions:
- Oracle Java 7: http://www.webupd8.org/2012/06/how-to-install-oracle-java-7-in-debian.html
- Oracle Java 8: http://www.webupd8.org/2014/03/how-to-install-oracle-java-8-in-debian.html

Oracle Java 9 (for both Ubuntu and Debian): http://www.webupd8.org/2015/02/install-oracle-java-9-in-ubuntu-linux.html
```

Installing the Oracle JDK 9 :

```
sudo apt-get update
```

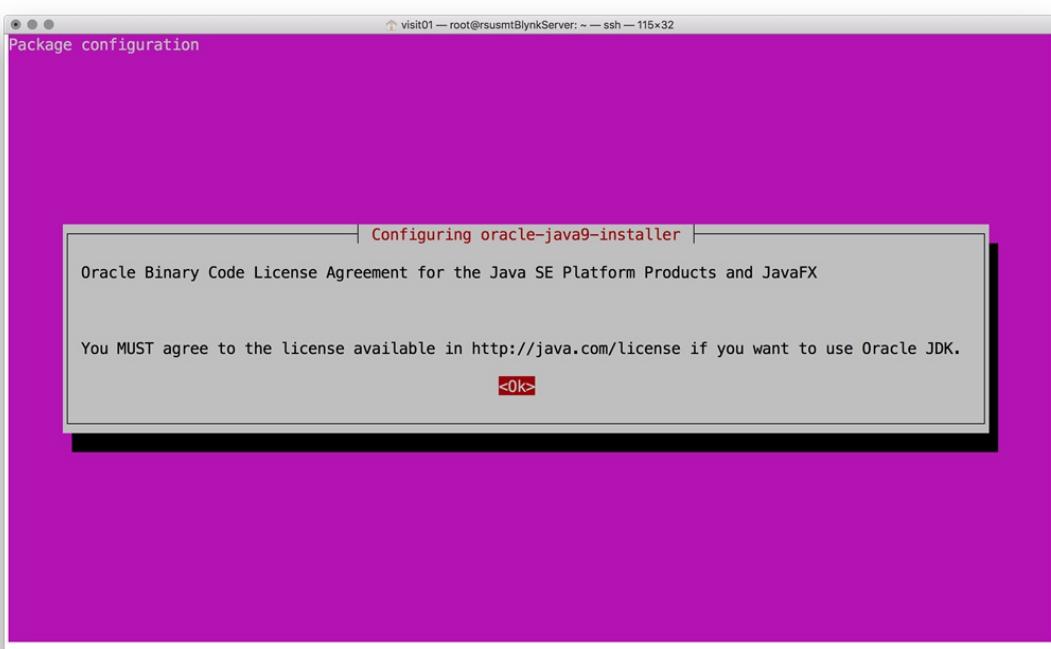
```
root@rsusmtBlynkServer:~# sudo apt-get update
Hit:1 http://mirrors.digitalocean.com/ubuntu xenial InRelease
Hit:2 http://mirrors.digitalocean.com/ubuntu xenial-updates InRelease
Hit:3 http://mirrors.digitalocean.com/ubuntu xenial-backports InRelease
Get:4 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial InRelease [17.5 kB]
Hit:5 http://security.ubuntu.com/ubuntu xenial-security InRelease
Get:6 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main amd64 Packages [2912 B]
Get:7 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main Translation-en [1260 B]
Fetched 21.7 kB in 1s (14.4 kB/s)
Reading package lists... Done
root@rsusmtBlynkServer:~#
```

## ติดตั้ง JAVA บน UBUNTU Server

Installing the Oracle JDK 9 :

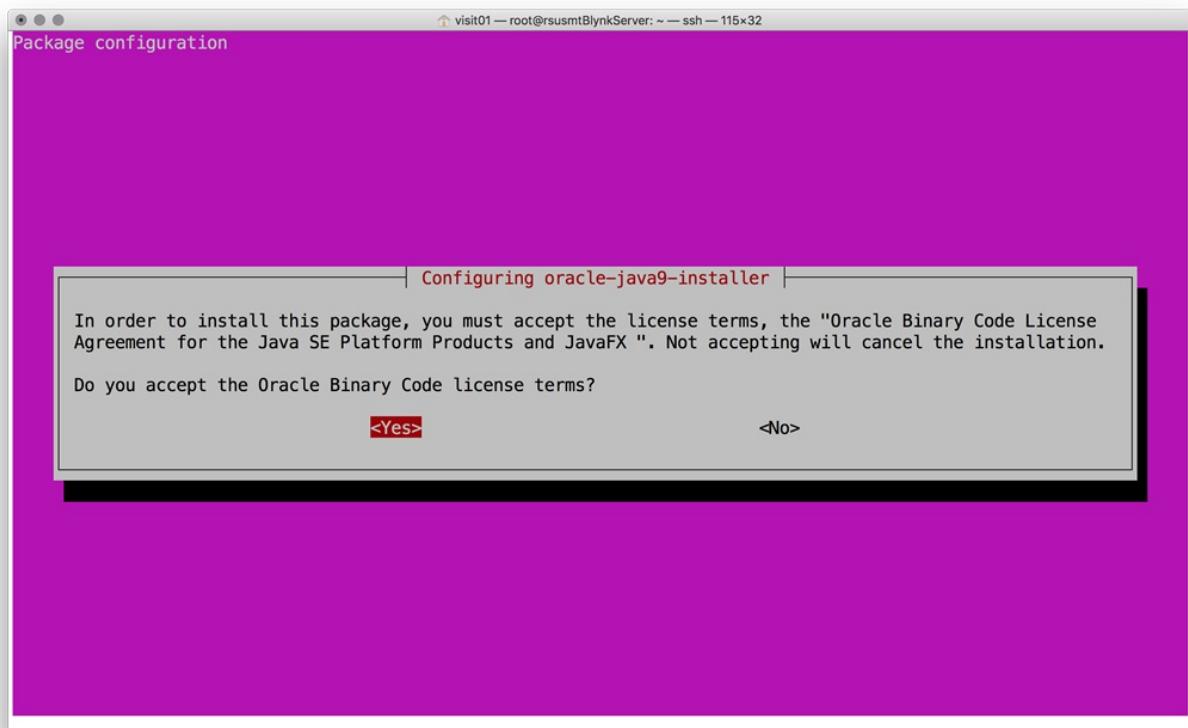
```
sudo apt-get install oracle-java9-installer
```

```
visit01 — root@rsusmtBlynkServer: ~ — ssh — 115x32
root@rsusmtBlynkServer:~# sudo apt-get install oracle-java9-installer
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following package was automatically installed and is no longer required:
  grub-pc-bin
Use 'sudo apt autoremove' to remove it.
The following additional packages will be installed:
  binutils gsfonts gsfonts-x11 java-common libfontenc1 libxfont1 oracle-java9-set-default x11-common
  xfonts-encodings xfonts-utils
Suggested packages:
  binutils-doc binfmt-support visualvm ttf-baekmuk | ttf-unfonts | ttf-unfonts-core ttf-kochi-gothic
  | ttf-sazanami-gothic ttf-kochi-mincho | ttf-sazanami-mincho ttf-arpthic-uming firefox | firefox-2 | iceweasel
  | mozilla-firefox | iceape-browser | mozilla-browser | epiphany-gecko | epiphany-webkit | epiphany-browser
  | galeon | midbrowser | moblin-web-browser | xulrunner | xulrunner-1.9 | konqueror | chromium-browser | midori
  | google-chrome
The following NEW packages will be installed:
  binutils gsfonts gsfonts-x11 java-common libfontenc1 libxfont1 oracle-java9-installer oracle-java9-set-default
  x11-common xfonts-encodings xfonts-utils
0 upgraded, 11 newly installed, 0 to remove and 0 not upgraded.
Need to get 6514 kB of archives.
After this operation, 20.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main amd64 oracle-java9-installer amd64 9.0.1-1~webup
d8~0 [30.8 kB]
Get:2 http://sgp1.mirrors.digitalocean.com/ubuntu xenial-updates/main amd64 binutils amd64 2.26.1-1ubuntu1~16.04.5
[2311 kB]
Get:3 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/main amd64 java-common all 0.56ubuntu2 [7742 B]
Get:4 http://sgp1.mirrors.digitalocean.com/ubuntu xenial/main amd64 gsfonts all 1:8.11+urwcyr1.0.7~pre44-4.2ubuntu1
[3374 kB]
Get:5 http://ppa.launchpad.net/webupd8team/java/ubuntu xenial/main amd64 oracle-java9-set-default amd64 9.0.1-1~web
upd8~0 [4442 B]
```



Installing the Oracle JDK 9 :

```
sudo apt-get install oracle-java9-installer
```



Installing the Oracle JDK 9 :

```
Verify installation: java -version
```

```
root@rsusmtBlynkServer:~# java -version
java version "9.0.1"
Java(TM) SE Runtime Environment (build 9.0.1+11)
Java HotSpot(TM) 64-Bit Server VM (build 9.0.1+11, mixed mode)
root@rsusmtBlynkServer:~#
```

### 3.ติดตั้ง Blynk Server

สร้าง folder ชื่อ BlynkServer

```
mkdir /BlynkServer
```

เข้าไปใน folder BlynkServer

```
cd /BlynkServer
```

ดาวน์โหลดไฟล์ติดตั้ง Server

```
wget https://github.com/blynkkk/blynk-server/releases/download/v0.28.5/server-0.28.5-java8.jar
```

ทดสอบรัน server

```
java -jar /BlynkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk &
```

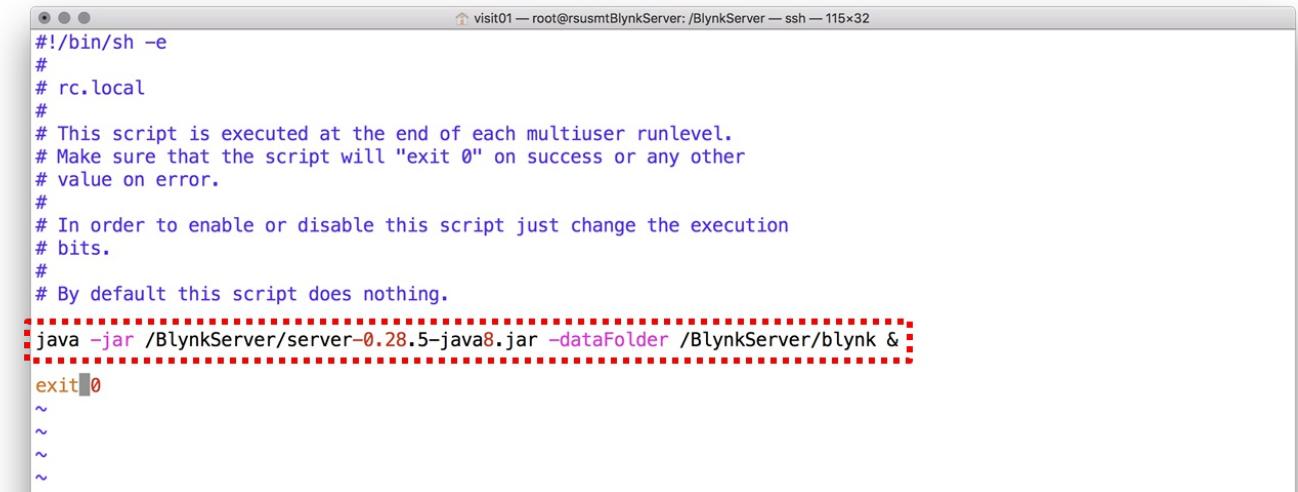
```
root@rsusmtBlynkServer:/BlynkServer# java -jar /BlynkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk
&
[1] 5175
root@rsusmtBlynkServer:/BlynkServer#
Blynk Server 0.29.0-SNAPSHOT successfully started.
All server output is stored in folder '/BlynkServer/logs' file.
Your Admin login email is admin@blynk.cc
Your Admin password is admin
```

## Blynk could server ติดตั้ง Blynk Server

ตั้งค่าให้ server รันอัตโนมัติหลังการบูต

แก้ไขไฟล์ /etc/rc.local เพิ่มบรรทัดด้านล่างเข้าไป

```
java -jar /BlynkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk &
```



The screenshot shows a terminal window titled "visit01 — root@rsusmtBlynkServer: /BlynkServer — ssh — 115x32". The window displays a portion of the /etc/rc.local script. A red dashed rectangle highlights the line "java -jar /BlynkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk &". Below this line, the command "exit 0" is shown, also enclosed in a red dashed rectangle. The terminal window has a light gray background and black text.

```
#!/bin/sh -e
#
# rc.local
#
# This script is executed at the end of each multiuser runlevel.
# Make sure that the script will "exit 0" on success or any other
# value on error.
#
# In order to enable or disable this script just change the execution
# bits.
#
# By default this script does nothing.

java -jar /BlynkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk &

exit 0
~
```

## 4.ตั้งค่าคอนฟิก Blynk Server

### ตั้งค่าคอนฟิก server

ให้ดาวน์โหลดไฟล์ชื่อ “server.properties” ที่เตรียมไว้ให้

```
cd /BlynkServer
```

```
wget https://www.dropbox.com/s/0r3dgsh0ix8yrdm/server.properties
```

ดาวน์โหลดไฟล์ชื่อ “server.properties”

```
root@rsusmtBlynkServer:/BlynkServer# cd /BlynkServer/
root@rsusmtBlynkServer:/BlynkServer# wget https://www.dropbox.com/s/0r3dgsh0ix8yrdm/server.properties
--2017-11-13 04:19:27-- https://www.dropbox.com/s/0r3dgsh0ix8yrdm/server.properties
Resolving www.dropbox.com (www.dropbox.com)... 162.125.81.1, 2620:100:6031:1::a27d:5101
Connecting to www.dropbox.com (www.dropbox.com)|162.125.81.1|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://dl.dropboxusercontent.com/content_link/7fiihtaUkDHLc0ZFgjhj8cm0J3kB0gsBBfoQmcSEaffRMJgUy6AYvR0eokENq9hLp/file [following]
--2017-11-13 04:19:27-- https://dl.dropboxusercontent.com/content_link/7fiihtaUkDHLc0ZFgjhj8cm0J3kB0gsBBfoQmcSEaffRMJgUy6AYvR0eokENq9hLp/file
Resolving dl.dropboxusercontent.com (dl.dropboxusercontent.com)... 162.125.81.6, 2620:100:6031:6::a27d:5106
Connecting to dl.dropboxusercontent.com (dl.dropboxusercontent.com)|162.125.81.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 4978 (4.9K) [text/plain]
Saving to: 'server.properties'

server.properties          100%[=====] 4.86K --.-KB/s   in 0s

2017-11-13 04:19:28 (37.0 MB/s) - 'server.properties' saved [4978/4978]

root@rsusmtBlynkServer:/BlynkServer#
```

แก้ไขไฟล์ชื่อ “server.properties” ด้วย text editor nano หรือ vi

```
root@rsusmtBlynkServer:/BlynkServer# ls
backup blynk clone deleted logs server-0.28.5-java8.jar server.properties static
root@rsusmtBlynkServer:/BlynkServer# nano server.properties
```

## ตั้งค่าคอนฟิก server

แก้ไขไฟล์ชื่อ “server.properties”

```

1 #hardware mqtt port
2 hardware.mqtt.port=8440
3
4 #hardware ssl port
5 hardware.ssl.port=8441
6
7 #hardware plain tcp/ip port
8 hardware.default.port=8442
9
10 #http and web sockets port
11 http.port=8080
12
13 #https and web sockets port
14 https.port=9443
15
16 #application ssl port
17 app.ssl.port=8443
18 |
19 #address to bind to. by default bounded to all interfaces
20 listen.address=
21
22 #by default server uses embedded in jar cert to simplify local server installation.
23 #WARNING DO NOT USE THIS CERTIFICATES ON PRODUCTION OR IN WHERE ENVIRONMENTS REAL SECURITY REQUIRED.
24 #provide either full path to files either use '.' for specifying current directory. For instance "./my
25 server.ssl.cert=
26 server.ssl.key=
27 server.ssl.key.pass=
28
29 #by default System.getProperty("java.io.tmpdir")/blynk used
30 data.folder=
31
32 #folder for logs.
33 logs.folder=./logs
34
35 #log debug level. trace|debug|info|error. Defines how precise logging will be.
36 log.level=info
37
38 #maximum number of devices allowed per account
39 user.devices.limit=25
40
41 #maximum number of tags allowed per account
42 user.tags.limit=100
43
44 #defines maximum allowed number of user dashboards. Needed to limit possible number of tokens.
45 user.dashboard.max.limit=100
46
47 #defines maximum allowed widget size in KBs as json string.
48 user.widget.max.size.limit=20
49
50 #user is limited with 100 messages per second.
51 user.message.quota.limit=100
52
53 #maximum allowed number of notification queue. Queue responsible for processing email, pushes, twits
54 #Because of performance issue - those queue is processed in separate thread, this is required due
55 #to blocking nature of all above operations. Usually limit shouldn't be reached.
56 notifications.queue.limit=5000
57
58 #Number of threads for performing blocking operations - push, twits, emails, db queries.
59 #Recommended to hold this value low unless you have to perform a lot of blocking operations.
60 blocking.processor.thread.pool.limit=6
61

```

## ตั้งค่าคอนฟิก server

แก้ไขไฟล์ชื่อ “server.properties”

```

62 #this setting defines how often we can send mail/tweet/push or any other notification. Specified in se
63 notifications.frequency.user.quota.limit=15
64
65 #this setting defines how often we can send webhooks. Specified in miliseconds
66 webhooks.frequency.user.quota.limit=1000
67
68 #this setting defines how big could be response for webhook GET request. Specified in kbs
69 webhooks.response.size.limit=72
70
71 #maximum size of user profile in kb's
72 user.profile.max.size=128
73
74 #number of strings to store in terminal widget
75 terminal.strings.pool.size=25
76
77 #number of strings to store in map widget
78 map.strings.pool.size=25
79
80 #number of strings to store in lcd widget
81 lcd.strings.pool.size=6
82
83 #maximum number of rows allowed
84 table.rows.pool.size=100
85
86 #period in millis for saving all user DB to disk.
87 profile.save.worker.period=60000
88
89 #period in millis for saving stats to disk.
90 stats.print.worker.period=60000
91
92 #specifies maximum period of time when hardware socket could be idle. After which
93 #socket will be closed due to non activity. In seconds. Default value 15 if not provided.
94 #leave it empty for infinity timeout
95 hard.socket.idle.timeout=15
96
97 #Enables native socket transport for Linux using JNI. Should be turned on only if you 100% sure.
98 #may not work on some environments. Used to increase server performance. Performance boost is ~20-40%.
99 enable.native.epoll.transport=false
100
101 #Enabled native openSSL support for SSL handlers. Should be turned on only if you 100% sure.
102 #may not work on some environments. Used to increase server performance. Performance boost is ~16%.
103 #For more details see - http://netty.io/wiki/forked-tomcat-native.html
104 enable.native.openssl=false
105
106 #enable DB
107 enable.db=false
108
109 #enable raw data storage to DB
110 enable.raw.db.data.store=false
111
112 #size of async logger ring buffer. should be increased for loads >2-3k req/sec
113 async.logger.ring.buffer.size=2048
114
115 #initial amount of energy
116 initial.energy=100000
117
118 #ADMINISTRATION SECTION
119
120 admin.rootPath=/admin

```

## Blynk could server

### ตั้งค่าคอนฟิก Blynk Server

#### ตั้งค่าคอนฟิก server

แก้ไขไฟล์ชื่อ “server.properties”

```
121
122 #used for reset password page and certificate generation.
123 #by default current server IP is taken. could be replaced with more friendly hostname.
124 #it is recommended to override this property with your server IP to avoid possible problems of host re
125 #server.host=test.blynk.cc
126
127 #email used for certificate registration, could be omitted in case you already specified it in mail.pr
128 contact.email=rsu.missmt@gmail.com
129
130 #network interface to determine server's current IP.
131 #only the first characters of the interface's name are needed.
132 #the default setting eth will use the first ethX interface found (i.e. eth0)
133 net.interface=eth
134
135 #comma separated list of administrator IPs. allow access to admin UI only for those IPs.
136 #you may set it for 0.0.0.0/0 to allow access for all.
137 #you may use CIDR notation. For instance, 192.168.0.53/24
138 allowed.administrator.ips=0.0.0.0/0
139
140 # default admin name and password, that will be created on initial server start
141 admin.email=rsu.missmt@gmail.com
142 admin.pass=xxxxxx
143
144 #comma separated list of users allowed to create accounts. leave it empty if no restriction required.
145 allowed.users.list=
```

#### ตั้งค่าคอนฟิกอีเมล์แจ้งข่าวสาร

ให้ดาวน์โหลดไฟล์ชื่อ “mail.properties” ที่เตรียมไว้ให้

```
cd /BlynkServer
```

```
wget https://www.dropbox.com/s/1hlrm6q078qo5sj/mail.properties
```

## Blynk could server

### ตั้งค่าคอนฟิก Blynk Server

#### ตั้งค่าคอนฟิก server

ดาวน์โหลดไฟล์ชื่อ “mail.properties”

```
[root@rsusmtBlynkServer:/BlynkServer# cd /BlynkServer/
[root@rsusmtBlynkServer:/BlynkServer# wget https://www.dropbox.com/s/1hlrm6q078qo5sj/mail.properties
--2017-11-13 04:26:00-- https://www.dropbox.com/s/1hlrm6q078qo5sj/mail.properties
Resolving www.dropbox.com (www.dropbox.com)... 162.125.81.1, 2620:100:6031:1::a27d:5101
Connecting to www.dropbox.com (www.dropbox.com)|162.125.81.1|:443... connected.
HTTP request sent, awaiting response... 302 Found
Location: https://dl.dropboxusercontent.com/content_link/cTYUStGWykPuCG4VtAG8uczqC0nnbvnubLV1kd44PvPqE6h8rDbg5rsWxpqo7se/file [following]
--2017-11-13 04:26:02-- https://dl.dropboxusercontent.com/content_link/cTYUStGWykPuCG4VtAG8uczqC0nnbvnubLV1kd44PvPqE6h8rDbg5rsWxpqo7se/file
Resolving dl.dropboxusercontent.com (dl.dropboxusercontent.com)... 162.125.81.6, 2620:100:6031:6::a27d:5106
Connecting to dl.dropboxusercontent.com (dl.dropboxusercontent.com)|162.125.81.6|:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 166 [text/plain]
Saving to: 'mail.properties'

mail.properties          100%[=====]      166  --.-KB/s    in 0s

2017-11-13 04:26:02 (15.8 MB/s) - 'mail.properties' saved [166/166]
root@rsusmtBlynkServer:/BlynkServer# ]
```

แก้ไขไฟล์ชื่อ “mail.properties” ด้วย text editor nano หรือ vi

```
[root@rsusmtBlynkServer:/BlynkServer# ls
backup blynk clone deleted logs mail.properties server-0.28.5-java8.jar server.properties static
[root@rsusmtBlynkServer:/BlynkServer# nano mail.properties ]
```

แก้ไขไฟล์ชื่อ “mail.properties”

ในที่นี้จะใช้ gmail เป็น mail server ส่งข้อความ ให้ใส่ username และ password เพื่อ login ให้ถูกต้อง  
แนะนำให้สร้าง account ใหม่

```
mail.smtp.auth=true
mail.smtp.starttls.enable=true
mail.smtp.host=smtp.gmail.com
mail.smtp.port=587
mail.smtp.username=rsu.missmt@gmail.com
mail.smtp.password=xxxxxxx
```

## Blynk could server

### ตั้งค่าคอนฟิก Blynk Server

ทดสอบการรัน server

เช็คก่อนว่ามีการรัน server ไว้หรือไม่ ถ้ามีให้ kill process ก่อน

```
root@rsusmtBlynkServer:/BlynkServer# ps aux | grep java
root      5175  1.6 25.7 2278828 128620 pts/3  Sl   04:04   0:28 java -jar
/BlynkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk
root      5364  0.0  0.1 12916    960 pts/4    S+   04:33   0:00 grep --co
lor=auto java
root@rsusmtBlynkServer:/BlynkServer# kill 5175
root@rsusmtBlynkServer:/BlynkServer# ps aux | grep java
root      5372  0.0  0.2 12916   1020 pts/4    S+   04:33   0:00 grep --co
lor=auto java
root@rsusmtBlynkServer:/BlynkServer# -
```

เช็ค server ว่าสามารถทำงานได้หรือไม่

```
root@rsusmtBlynkServer:/BlynkServer# ps aux | grep java
root      5434  9.5 27.9 2278812 139772 pts/4  Sl   04:39   0:11 java -jar /Bly
nkServer/server-0.28.5-java8.jar -dataFolder /BlynkServer/blynk
root      5481  0.0  0.2 12916   1012 pts/4    S+   04:41   0:00 grep --color=a
uto java
root@rsusmtBlynkServer:/BlynkServer# -
```

ตั้งค่าคอนฟิกข้อความในอีเมล์แจ้งข่าวสาร

ให้ดาวน์โหลดไฟล์ชื่อ “single\_token\_mail\_body.txt” ที่เตรียมไว้ให้

cd /BlynkServer

wget [https://www.dropbox.com/s/bwzp7kw4svrz150/single\\_token\\_mail\\_body.txt](https://www.dropbox.com/s/bwzp7kw4svrz150/single_token_mail_body.txt)

Welcome to RSU MISSMT internet of things platform.

-

Thank you for using on our cloud server.

We hope you are enjoying your IoT service.

-

<https://www.facebook.com/RSU.IT.SMT/>

<https://sites.google.com/site/myrsusmt/contact>

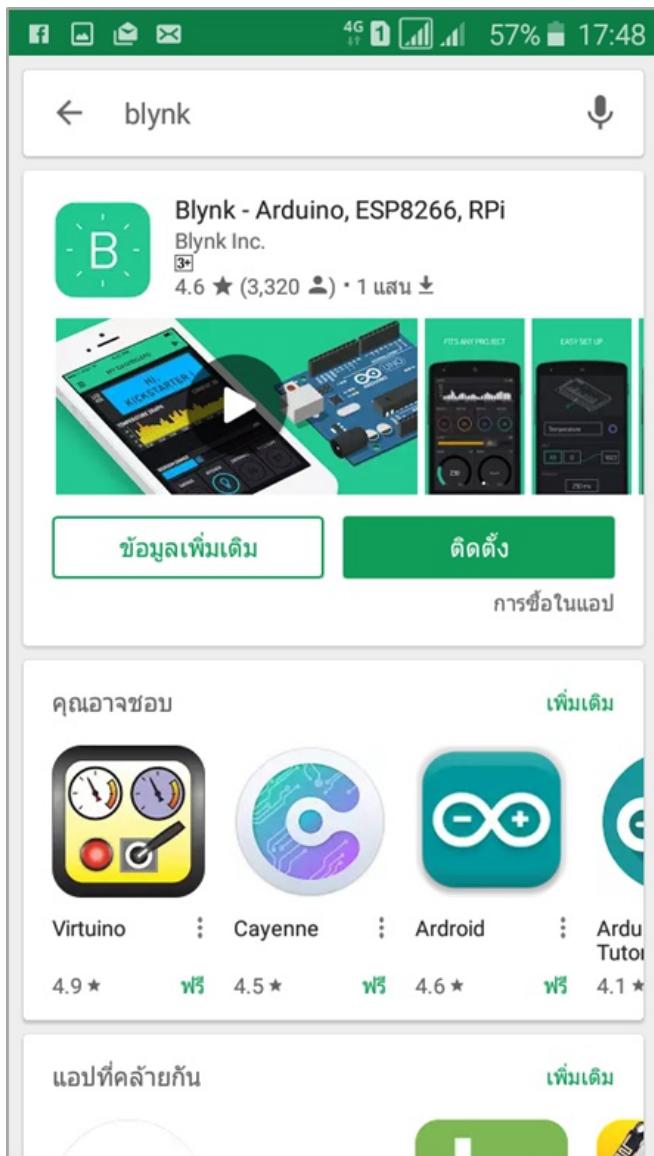
<https://www2.rsu.ac.th>

# การติดตั้ง Blynk

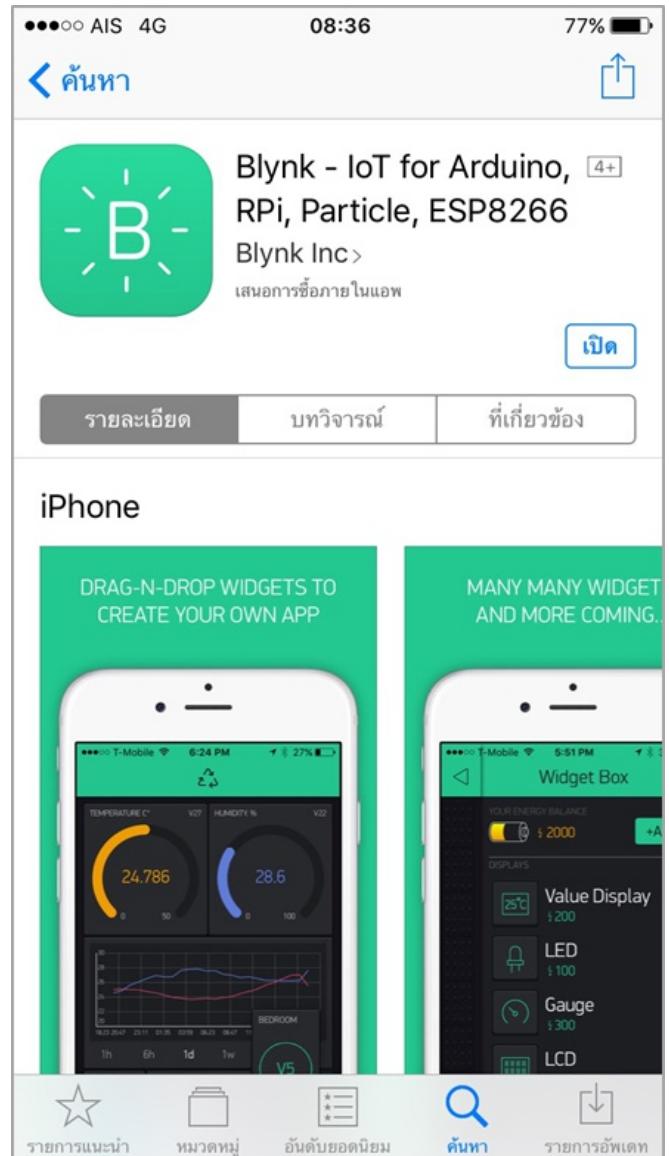
รู้จัก blynk app และการใช้งาน

## 5.ทดสอบเข้ามายัง Blynk Server

ดาวน์โหลด Blynk บน smart phone

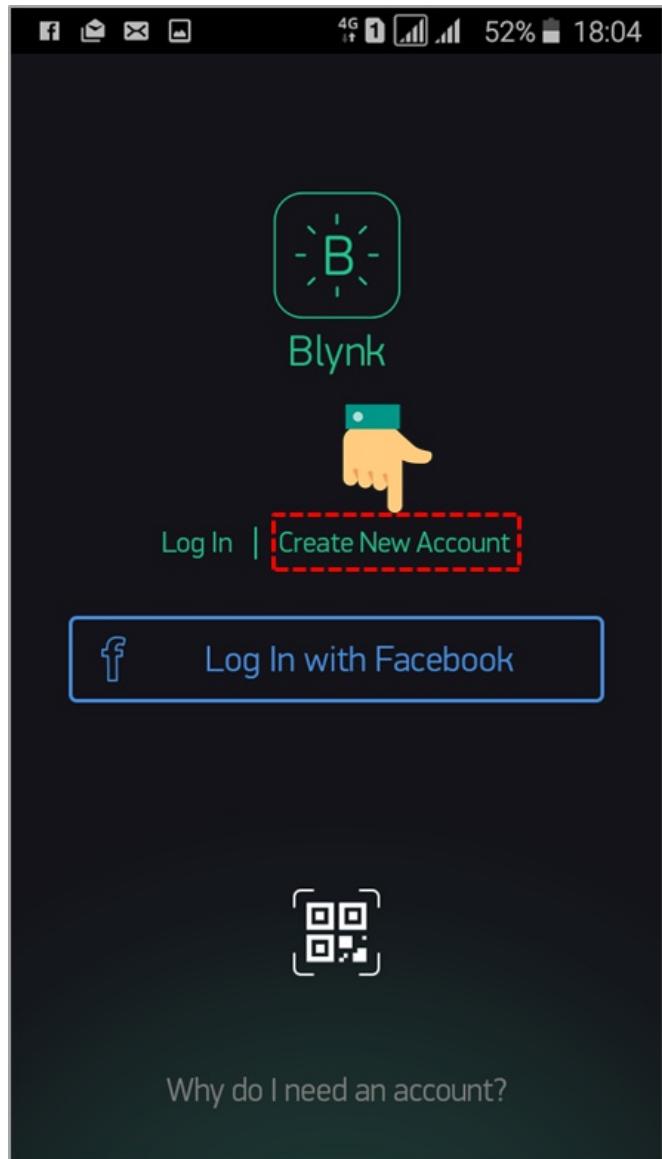


Android OS

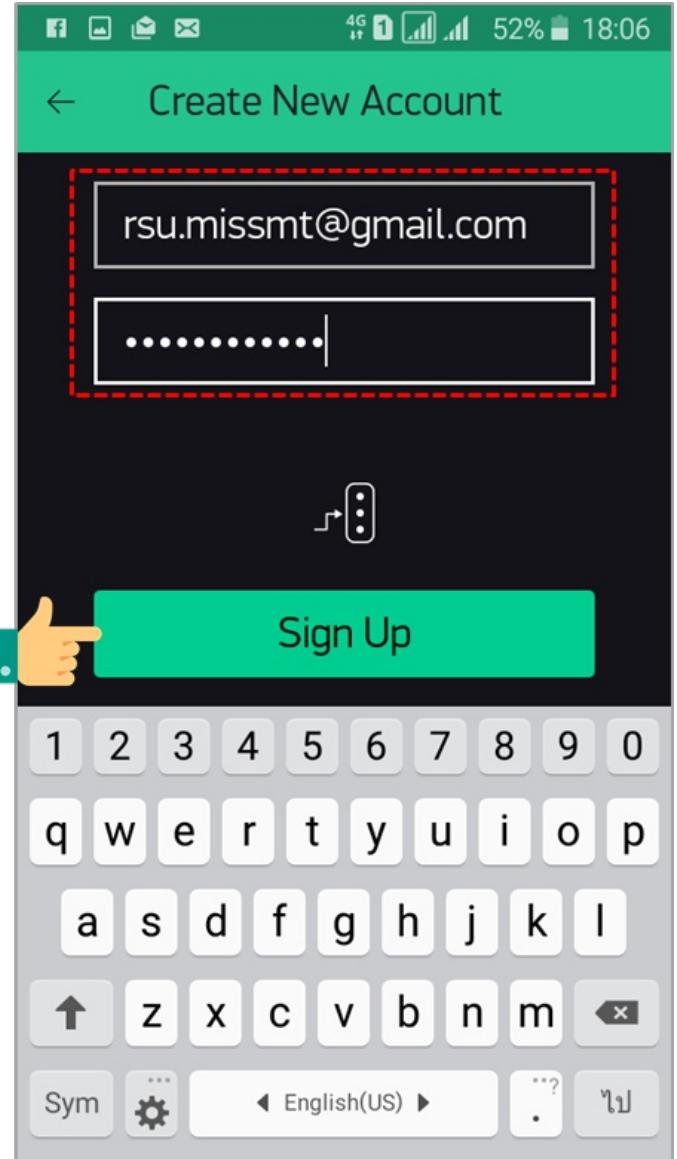


iOS

ลงทะเบียนเข้าใช้งานบน blynk server

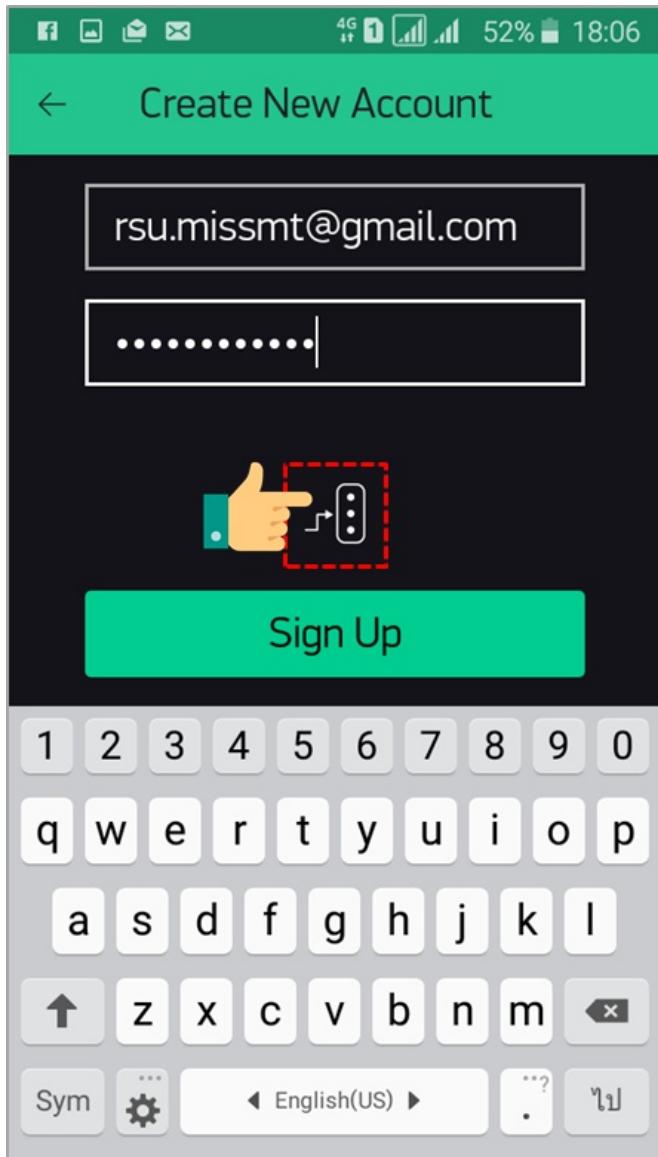


สร้างผู้ใช้งานใหม่

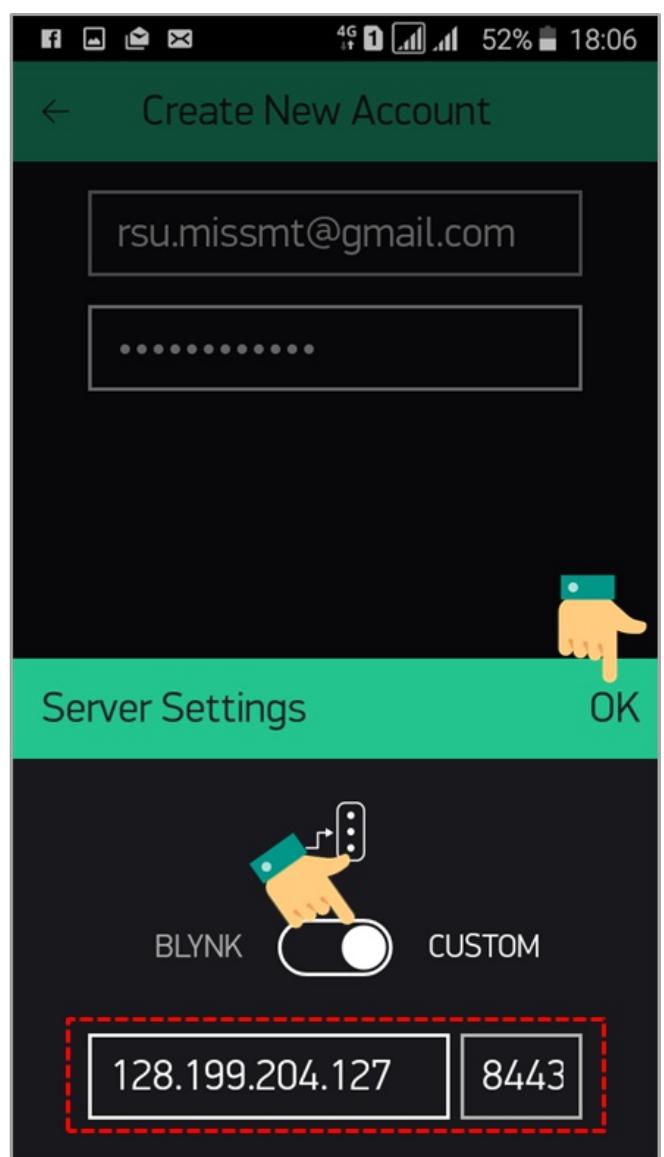


ระบุที่อยู่อีเมล์และรหัสผ่าน

## เลือก Cloud SERVER ที่สร้างขึ้น

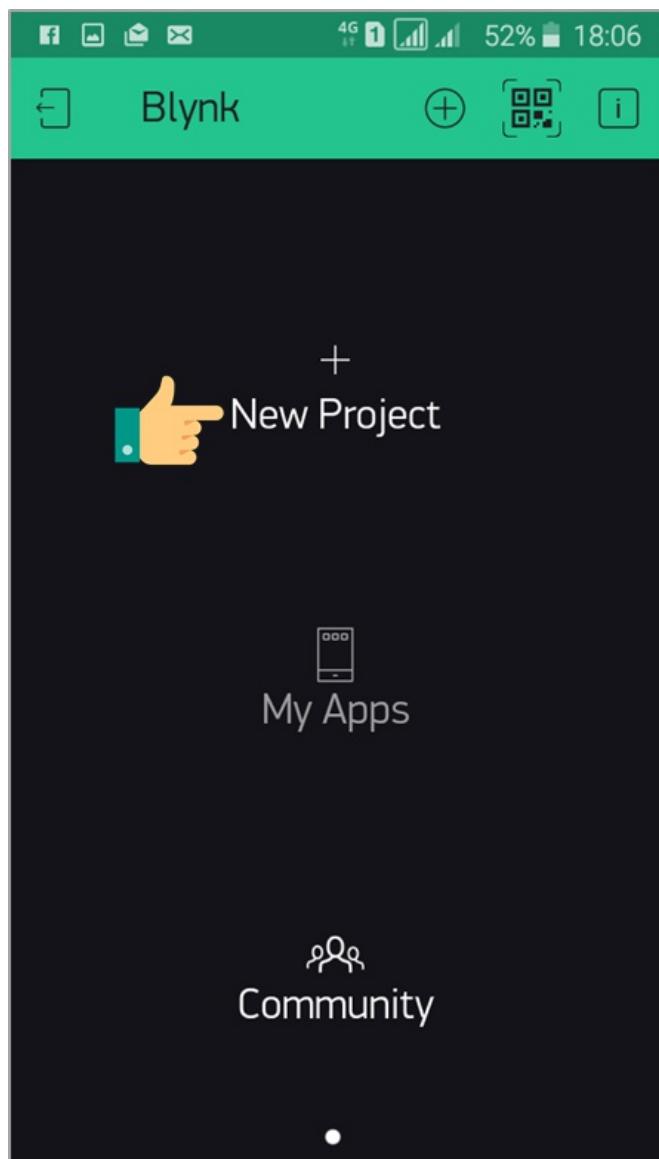


เลือกตั้งค่า Server

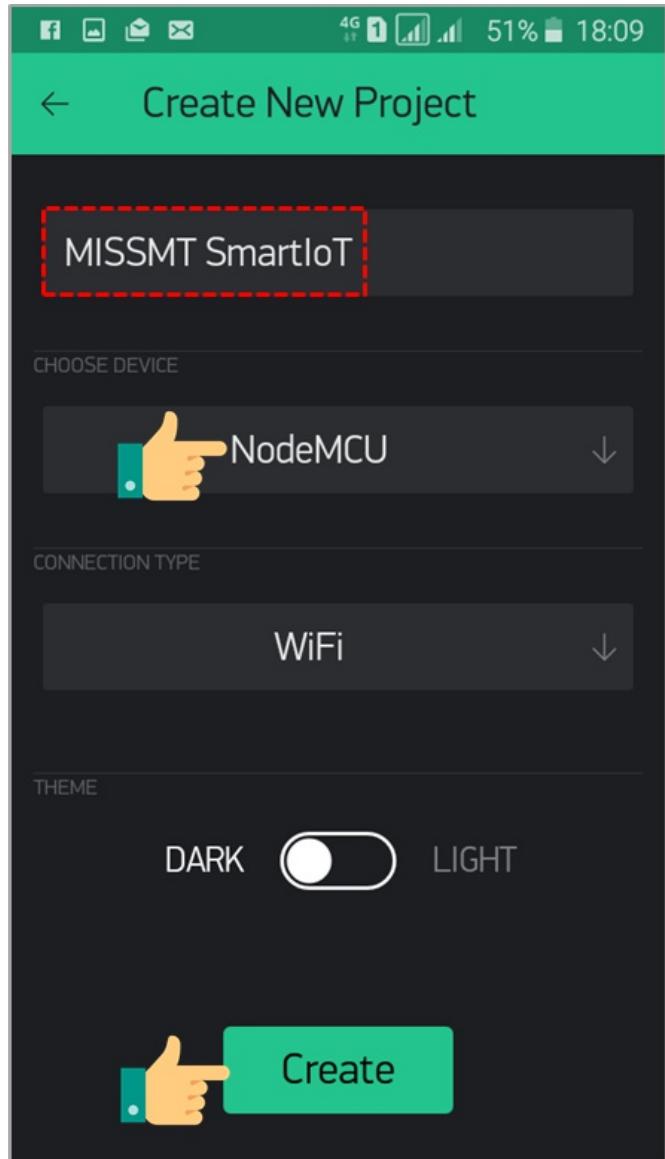


ระบุ IP Server ที่สร้างขึ้นใหม่

## เข้าสู่หน้าแรก ทดลองสร้างโปรเจค

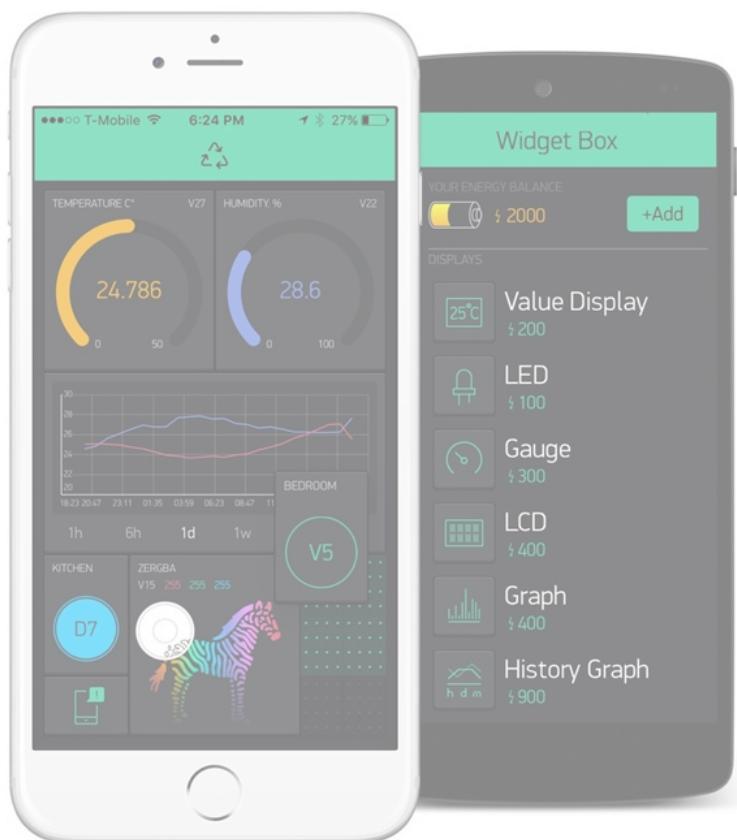
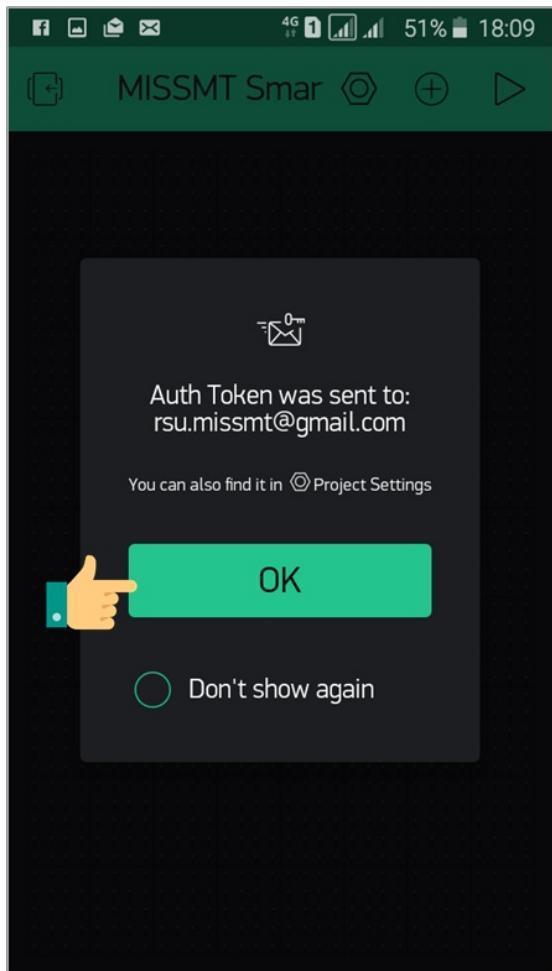


สร้างโปรเจคใหม่



สร้างโปรเจค

## ลงทะเบียนเข้าใช้งานบน blynk server



ระบบส่งเมล์รหัส TOKEN ไปให้

## รหัสลงทะเบียนเข้าใช้งานบน blynk server

The screenshot shows a Gmail inbox with the following details:

- Subject:** Auth Token for MISSMT SmartIoT project and device  
MISSMT SmartIoT
- From:** rsu.missmt@gmail.com
- Date:** 18:09 (0 นาทีที่ผ่านมา)
- Body:**

Welcome to RSU IMSSMT internet of things platform.

Thank you for using on our cloud server.  
We hope you are enjoying your IoT service.

<https://www.facebook.com/RSU.IT.SMT/>  
<https://sites.google.com/site/myrsusmt/contact>  
<https://www2.rsu.ac.th>

**Red box highlights:** The 'Auth Token' value: 8720f918efce45c5882436ca50d52ee1

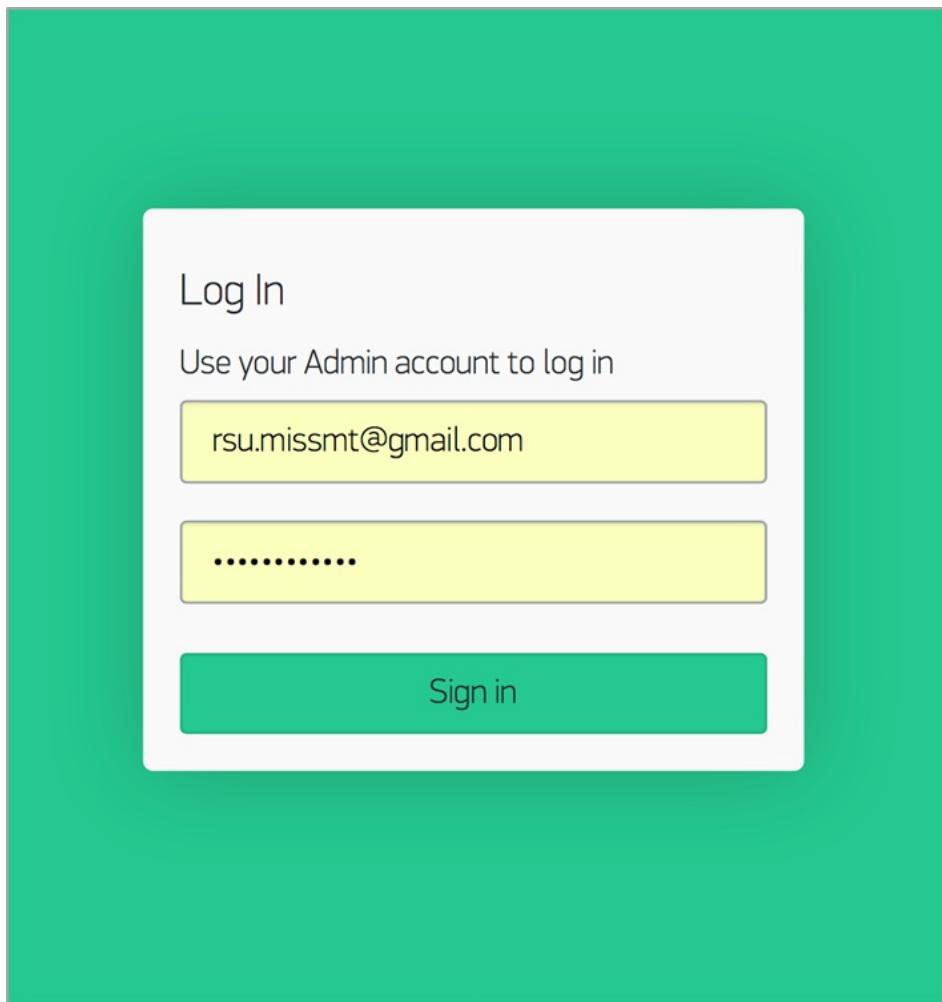
**Bottom of the email:** Click here to [ตอบกลับ](#) or [ส่งต่อ](#)

**Left sidebar:** Includes 'เขียน', 'กสติ์', 'ติดดาว', 'จดหมายที่ส่งแล้ว', 'จดหมายร่าง', 'เพิ่มเติม', 'smt', and a '+' button.

## 6. ทดสอบเข้าสู่หน้าจัดการ Administrator

Blynk administrator

<https://128.199.117.92:9443/admin>



Blynk Administration

- Users
- Stats
- Realtime
- Request per user
- Messages
- Board types
- Login types
- Widgets
- Projects per user
- Cells per project
- Size of user profile
- Webhook hosts
- IPs

- Hardware Info
- Library versions
- CPU types
- Hardware boards

## Board Types

Board Name	Count
NodeMCU	295
ESP8266	33
ESP32 Dev Board	4
Microduino CoreUSB	1
Raspberry Pi 3 B	1

1 - 5 on 5

Users

Stats

Realtime

Request per user

Messages

Board types

Login types

Widgets

Projects per user

Cells per project

Size of user profile

Webhook hosts

IPs

Hardware Info

Library versions

CPU types

Hardware boards

Connection types

## Edit user "visit01@gmail.com"

Email visit01@gmail.com

Name visit01@gmail.com

Pass ..... ✓

LastModifiedTs 1509928614995

Energy 99000

AppName Blynk

Region local

LastLoggedIP 49.229.72.164

Profile Dashboards

Id	952049623	
Name	GPS	
CreatedAt	1509636231032	
Theme	Blynk	

List Delete

## 7. สร้าง firewall ให้ Blynk Server

### Blynk Cloud Server firewall

#### สร้าง firewall

The screenshot shows the DigitalOcean Networking interface. At the top, there are navigation links: Droplets, Spaces, Images, Networking (which is highlighted in blue), Monitoring, API, and Support. To the right of these are a green 'Create' button and a user profile icon. Below the navigation, the word 'Networking' is displayed in a large, dark font. Underneath it, there are five tabs: Domains, Floating IPs, Load Balancers, Firewalls (which is underlined in blue to indicate it is active), and PTR records. In the bottom right corner of the main content area, there is a large blue 'Create Firewall' button.

#### ตั้งชื่อ firewall

The screenshot shows the 'Create Firewall' form. At the top, it has the same navigation bar as the previous screenshot: Droplets, Spaces, Images, Networking, Monitoring, API, and Support, with Networking selected. Below the navigation, there is a back arrow labeled '← Firewalls'. The main title is 'Create Firewall'. A 'Name' field contains the value 'myFirewall', which is highlighted with a blue border and has a green checkmark icon to its right. Under the heading 'Inbound Rules', there is a sub-instruction: 'Set the Firewall rules for incoming traffic. Only the specified ports will accept inbound connections. All other traffic will be dropped.' Below this, there is a table with four columns: Type, Protocol, Port Range, and Sources. The first row shows a dropdown menu set to 'SSH', 'TCP' as the protocol, '22' as the port range, and a 'Sources' field containing 'All IPv4'. To the right of this row is a red 'Delete' link.

## กำหนดพอร์ตใช้งานขาเข้า

### Inbound Rules

Set the Firewall rules for incoming traffic. Only the specified ports will accept inbound connections. All other traffic will be dropped.

Type	Protocol	Port Range	Sources	
SSH	TCP	22	All IPv4	Delete
Custom	TCP	Ports 8080	All IPv4	Delete
Custom	TCP	Ports 8440	All IPv4	Delete
Custom	TCP	Ports 8441	All IPv4	Delete
Custom	TCP	Ports 8442	All IPv4	Delete
Custom	TCP	Ports 8443	All IPv4	Delete
Custom	TCP	Ports 9443	All IPv4	Delete

## กำหนดพอร์ตใช้งานขาออก

### Outbound Rules

Set the Firewall rules for outbound traffic. Outbound traffic will only be allowed to the specified ports. All other traffic will be blocked.

Type	Protocol	Port Range	Destinations	
All TCP	TCP	All ports	All IPv4 All IPv6	Delete
All UDP	UDP	All ports	All IPv4 All IPv6	Delete
New rule				

### Apply to Droplets

Select Droplets to apply your Firewall rules to.

Search for a Droplet or a tag

Create Firewall

## Blynk Cloud Server firewall

ตั้งค่า firewall

Droplets Spaces Images Networking Monitoring API Support Create ▾ 

## Networking

Domains Floating IPs Load Balancers Firewalls PTR records

Create Firewall

Name	Droplets	Rules	Created	More ▾
 myFirewall	0	9	1 minute ago	<a href="#">More ▾</a>
 IoTfirewall	1	9	3 months ago	<a href="#">More ▾</a>

 **myFirewall**  
9 Rules / 0 Droplets

Rules Droplets Destroy



Choose Droplets  
Your Firewall isn't applied to any Droplet. Select to which ones it should apply or use tags to select groups of Droplets.

Add Droplets

## Blynk Cloud Server firewall

กำหนด firewall ใช้งานใน server

The screenshot shows the 'myFirewall' section with 9 Rules. A modal window titled 'Add Droplet' is open, featuring a search bar with the placeholder 'Search for a Droplet or a tag' and a blue 'Add Droplet' button. Below the search bar is a decorative graphic of a water droplet surrounded by concentric circles. The main area is labeled 'Choose Droplets' with the instruction: 'Your Firewall isn't applied to any Droplet. Select to which ones it should apply or use tags to select groups of Droplets.' A blue 'Add Droplets' button is located at the bottom of this section.

rsuBlynkServer ใช้งาน firewall เรียบร้อย

The screenshot shows the 'Networking' section with a green 'Create' button and a user icon. Below it, the 'Firewalls' section is shown with 'myFirewall' containing 9 Rules / 1 Droplet. The 'Droplets' tab is selected, showing a table with one entry:

Name	IP Address	State	Added	More
rsuBlynkServer 512 MB / 20 GB / SGP1	128.199.117.92	Updating...	Just now	More

# บทที่ 5 LoRa

แนะนำ LoRa

รับส่งข้อมูลผ่าน LoRa บนคลื่นความถี่ 433MHz

# LoRa

Long range, low power  
wireless platform



## What is LoRa?

LoRa: Long range, low power wireless platform is the prevailing technology choice for building IoT networks worldwide.



LoRa: Long range, low power wireless platform is the prevailing technology choice for building IoT networks worldwide.

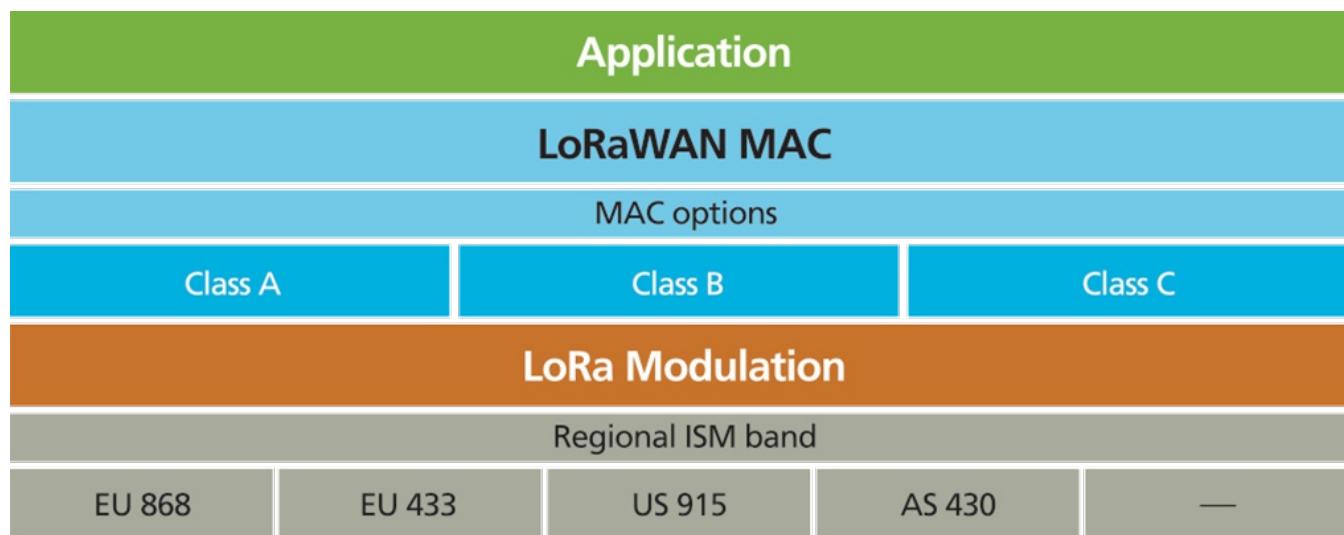
Smart IoT applications have improved the way we interact and are addressing some of the biggest challenges facing cities and communities: climate change, pollution control, early warning of natural disasters, and saving lives. Businesses are benefiting too, through improvements in operations and efficiencies as well as reduction in costs.

This wireless RF technology is being integrated into cars, street lights, manufacturing equipment, home appliances, wearable devices – anything, really. LoRa Technology is making our world a Smart Planet.



## LoRa Technology

LoRa Technology offers a very compelling mix of long range, low power consumption and secure data transmission

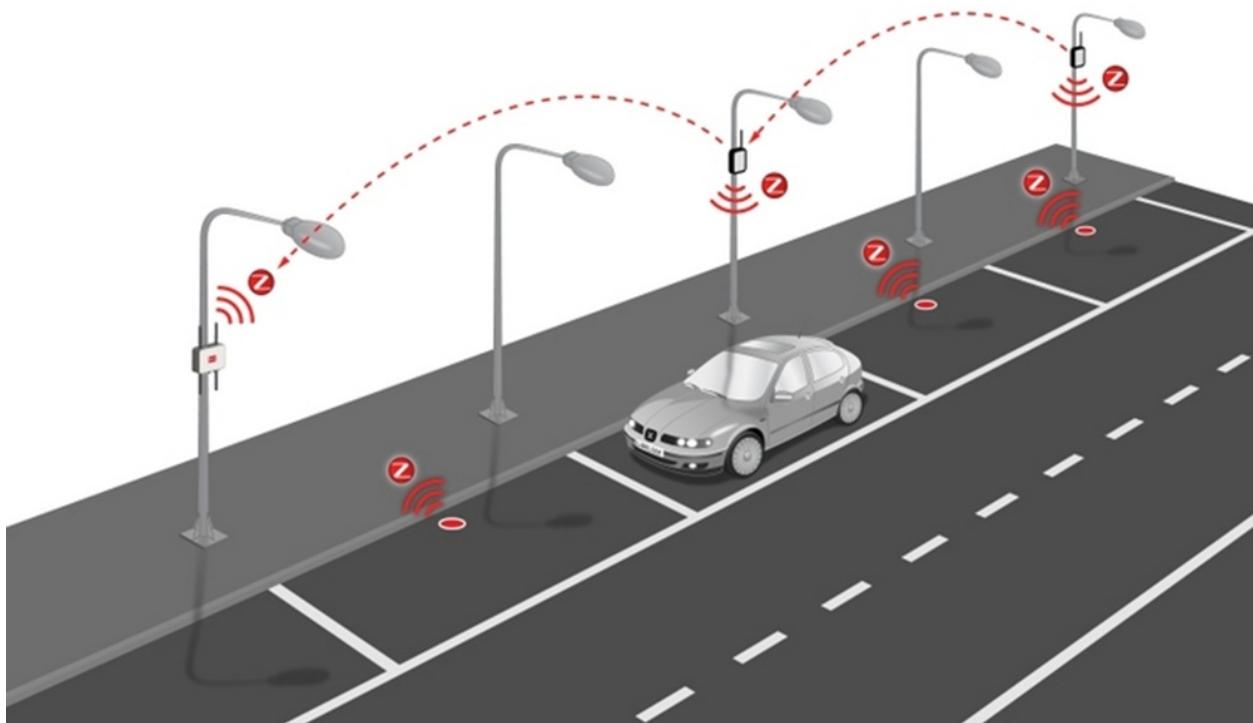


A Semtech innovation, LoRa Technology offers a very compelling mix of long range, low power consumption and secure data transmission. Public and private networks using this technology can provide coverage that is greater in range compared to that of existing cellular networks. It is easy to plug into the existing infrastructure and offers a solution to serve battery-operated IoT applications. Semtech builds LoRa Technology into its chipsets. These chipsets are then built into the products offered by our vast network of IoT partners and integrated into LPWANs from mobile network operators worldwide.



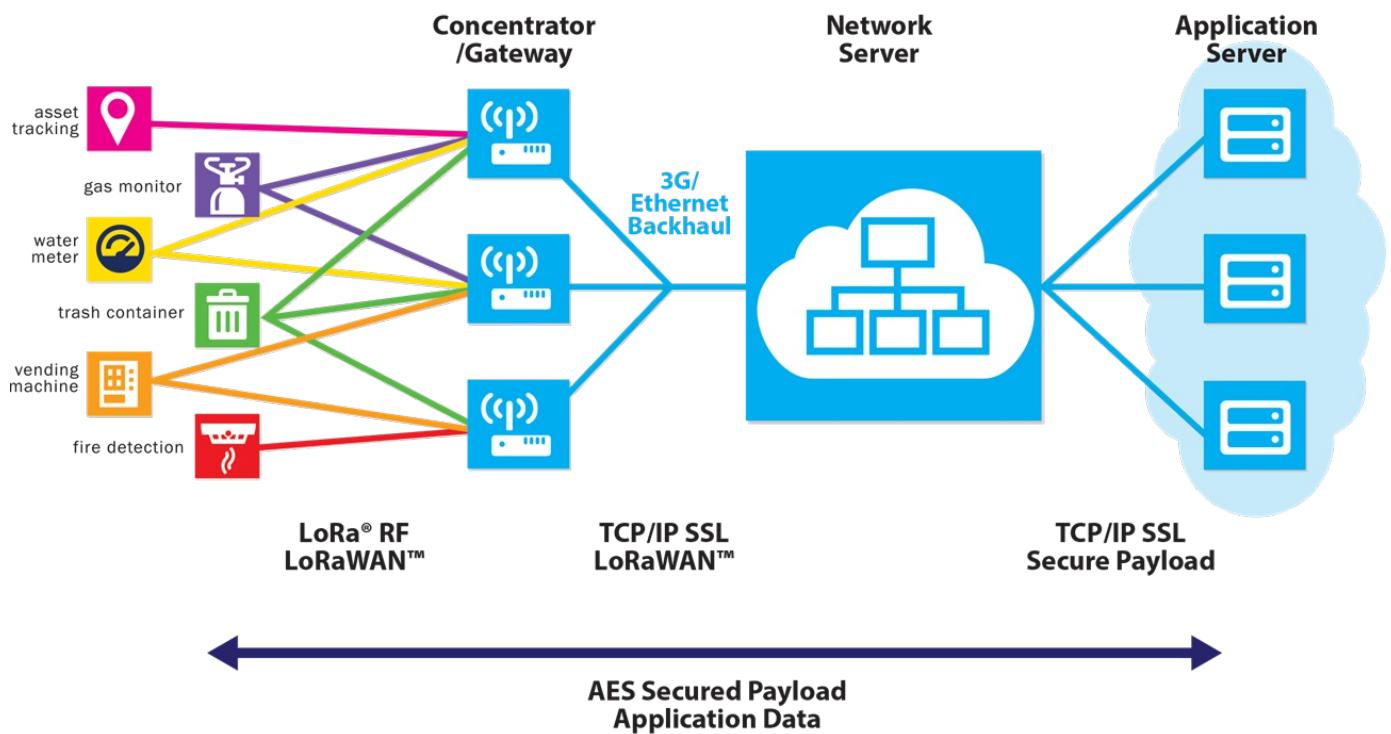
## LoRaWAN Protocol

Wide area communication between remote sensors and gateways connected to the



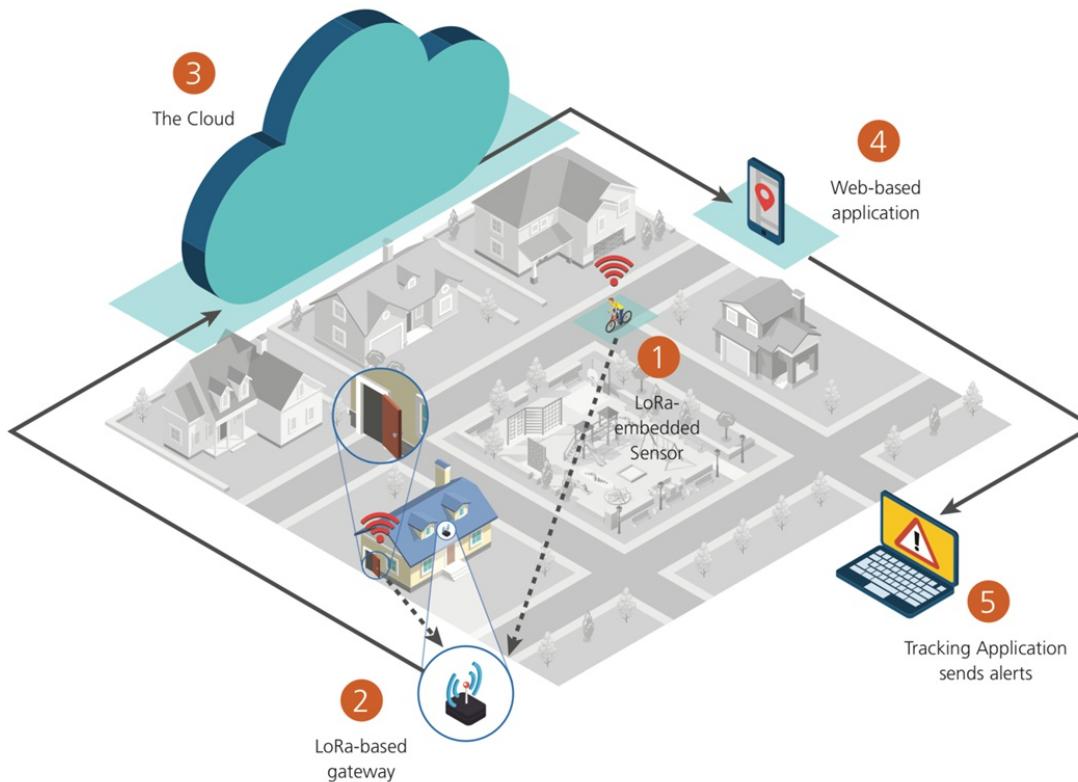
LoRaWAN is a protocol specification built on top of the LoRa technology developed by the LoRa Alliance. It uses unlicensed radio spectrum in the Industrial, Scientific and Medical (ISM) bands to enable low power, wide area communication between remote sensors and gateways connected to the network. This standards-based approach to building a LPWAN allows for quick set up of public or private IoT networks anywhere using hardware and software that is bi-directionally secure, interoperable and mobile, provides accurate localization, and works the way you expect. The specification is available for free to download from the LoRa Alliance website today.

## Network Diagram



## LoRa Applications

### Smart Home Asset Tracking

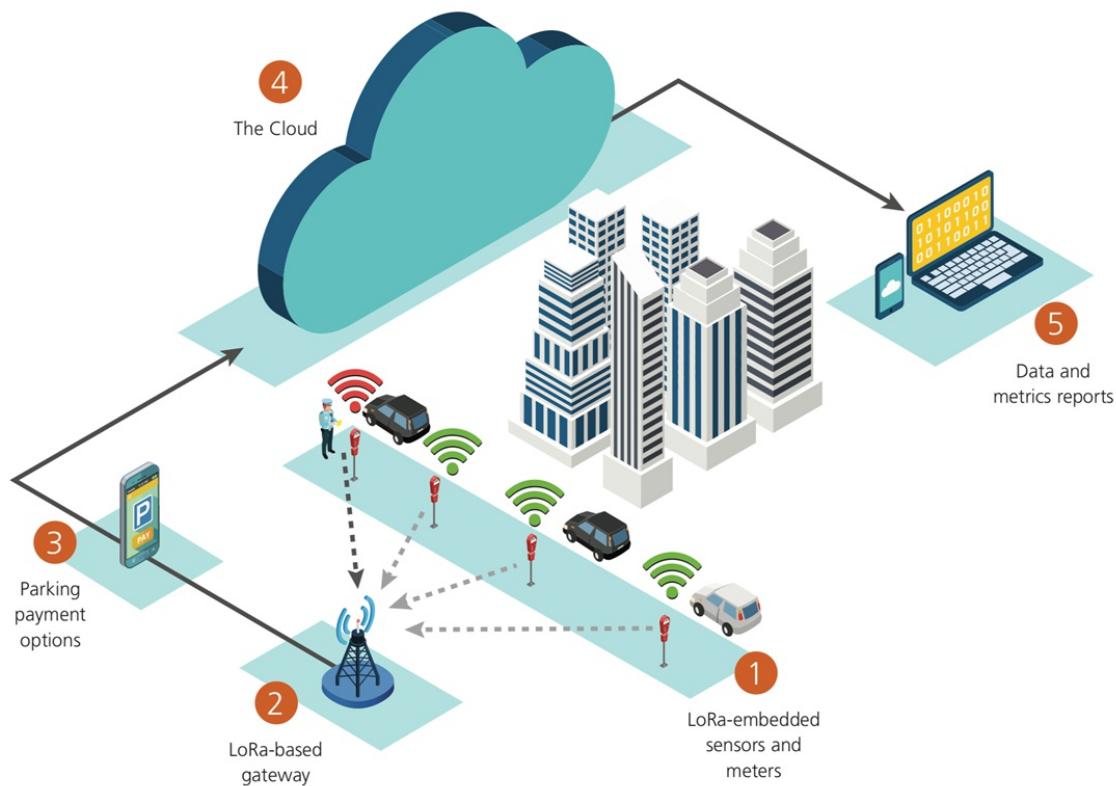


## DESCRIPTION

LoRa® devices and wireless RF technology (LoRa Technology) is making it easy and economical for consumers to dive into the Internet of Things (IoT). With affordable sensors, gateways and tracking devices, families can also experience the same IoT benefits, including asset tracking and real-time monitoring, as enterprises and global organizations. LoRa Technology enables smart home devices to be interoperable and coordinate across an expansive distance, enabling better asset monitoring and smarter grid behaviors. Using typical wireless technologies for this type of application would increase bandwidth and power costs, but with LoRa-based devices, low amounts of power and wireless bandwidth are utilized.

## LoRa Applications

### Smart Parking

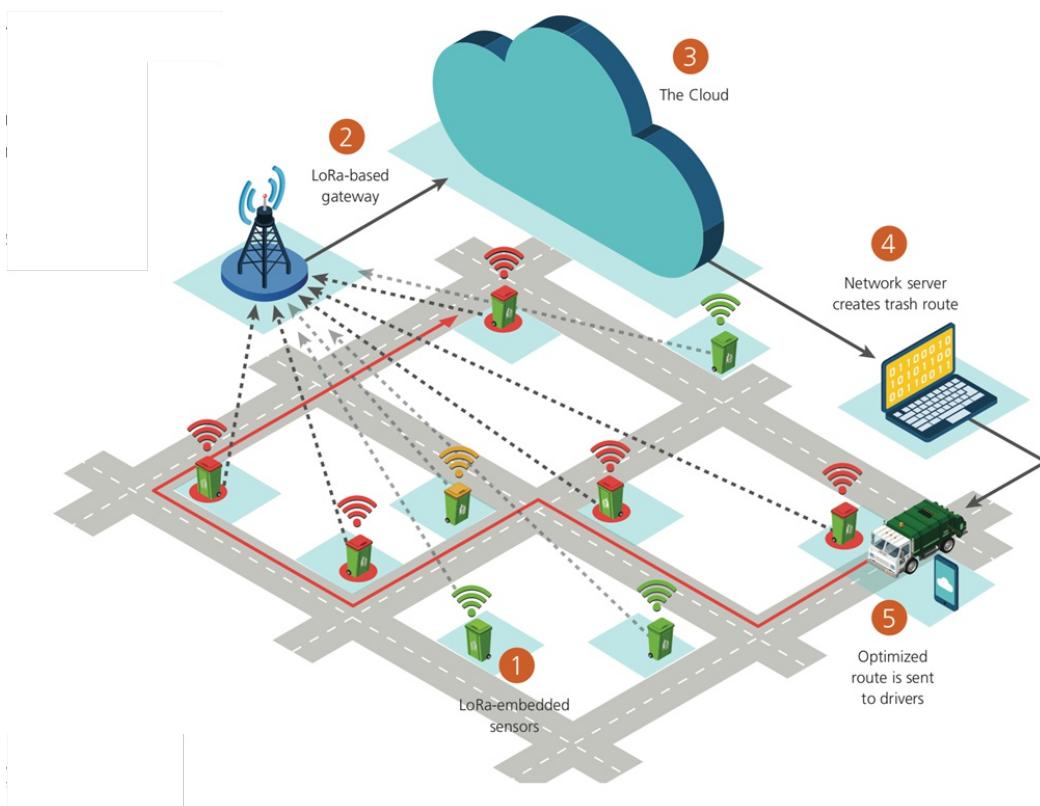


### DESCRIPTION

LoRa® devices and wireless RF technology (LoRa Technology) is making it easy and economical for smart cities to improve public services. From better usage of city street lights to smarter parking meters, LoRa Technology enables the most advanced and affordable solutions for urban planning and management. While traditional urban monitoring systems have been prohibitively expensive for medium and small sized municipalities, LoRa Technology provides an affordable solution that can increase municipal revenues and improve city services. City governments are always in search of better ways to optimize spending and increase revenue. With LoRa Technology, municipalities are able to both lower the overall cost of infrastructure while advancing long-term technological goals.

## LoRa Applications

### Waste Management



### DESCRIPTION

Traditionally, waste collection has been performed on a fixed schedule. However, regularly scheduled waste collection is not optimal as it does not account for different individuals or businesses filling up their waste bins at different rates. This means collection trucks must stop at each point on their route to empty each waste bin regardless of whether they are full or not. It also leads to situations where some waste bins overflow before the next collection schedule. By implementing sensors embedded with LoRa Technology into waste bins and using an intelligent low-power wide area network based on the LoRaWAN™ protocol, cities can significantly reduce their operational costs by streamlining their waste collection routes and deploying trash bins only where they are necessary.

## LoRa Applications

### Fall Detection



### DESCRIPTION

For the elderly, falling and not being able to get up or summon help is a very scary prospect and happens frequently enough that it is a public safety concern in certain communities. Fear of falling can keep elderly from participating in activities, accelerate depression and decrease social activities all of which leads to a reduced quality of life. IoT-based fall detection systems alleviate the fear of not being discovered after a fall which can help elderly to live an active, normal life. By implementing a IoT-based fall detection solution comprised of sensors and gateways embedded with LoRa Technology and an intelligent low power wide area network based on the LoRaWAN™ protocol, elderly people can live more full lives.



## LoRa Applications



**Asset Tracking**  
[Download Application Brief »](#)



**Cattle Tracking**  
[Download Application Brief »](#)



**Fleet Management**  
[Download Application Brief »](#)



**Smart Home Asset Tracking**  
[Download Application Brief »](#)



**Smart Parking**  
[Download Application Brief »](#)



**Air Pollution Monitoring**  
[Download Application Brief »](#)



**Agriculture Processing**  
[Download Application Brief »](#)



**Energy Management & Sustainability**  
[Download Application Brief »](#)



**Fall Detection**



**Fire Detection**



**Fleet Tracking**



**Home Security**



**Indoor Air Quality**  
[Download Application Brief »](#)



**Industrial Temp**  
[Download Application Brief »](#)



**Liquid Presence Detection**  
[Download Application Brief »](#)



**Locating Stolen Vehicles & Cargo**  
[Download Application Brief »](#)



**Medical Fridge**  
[Download Application Brief »](#)



**Precision Farming**  
[Download Application Brief »](#)



**Predictive Maintenance**  
[Download Application Brief »](#)



**Radiation Leak Detection**  
[Download Application Brief »](#)



**Shipment Quality**



**Smart Lighting**



**Tank Flow Monitoring**



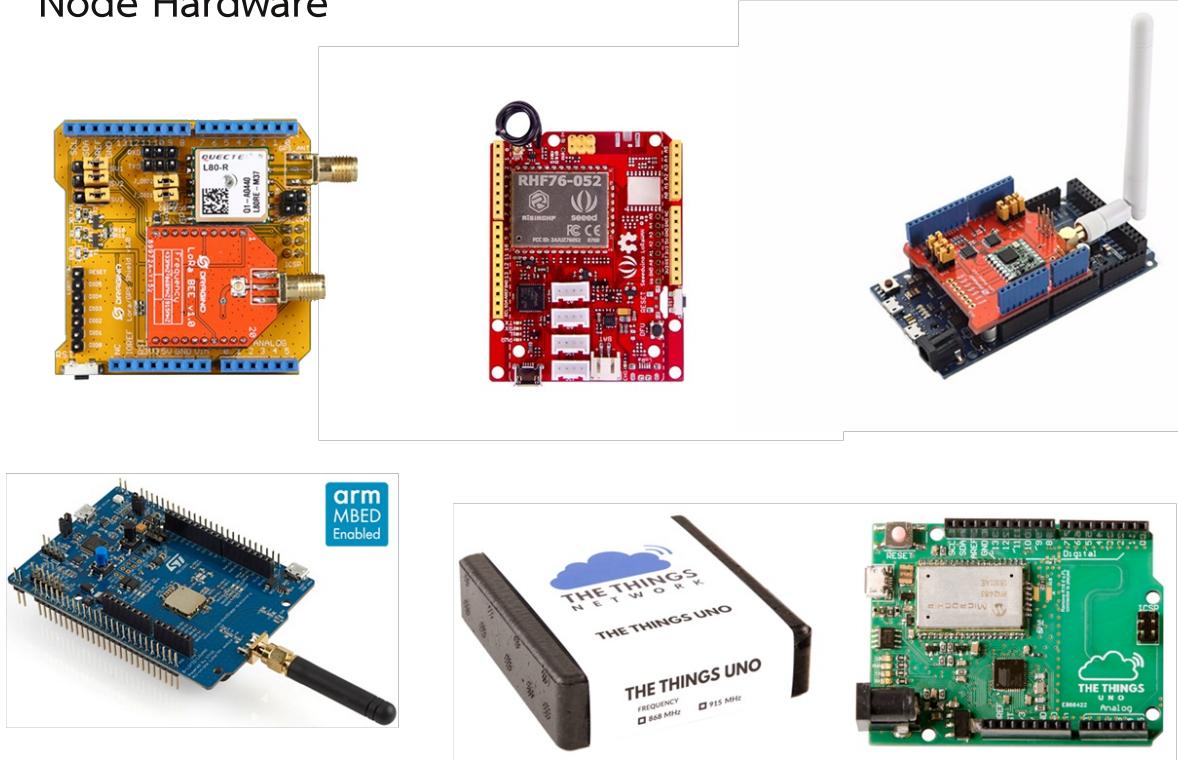
**Waste Management**



## Gateway Hardware



## Node Hardware

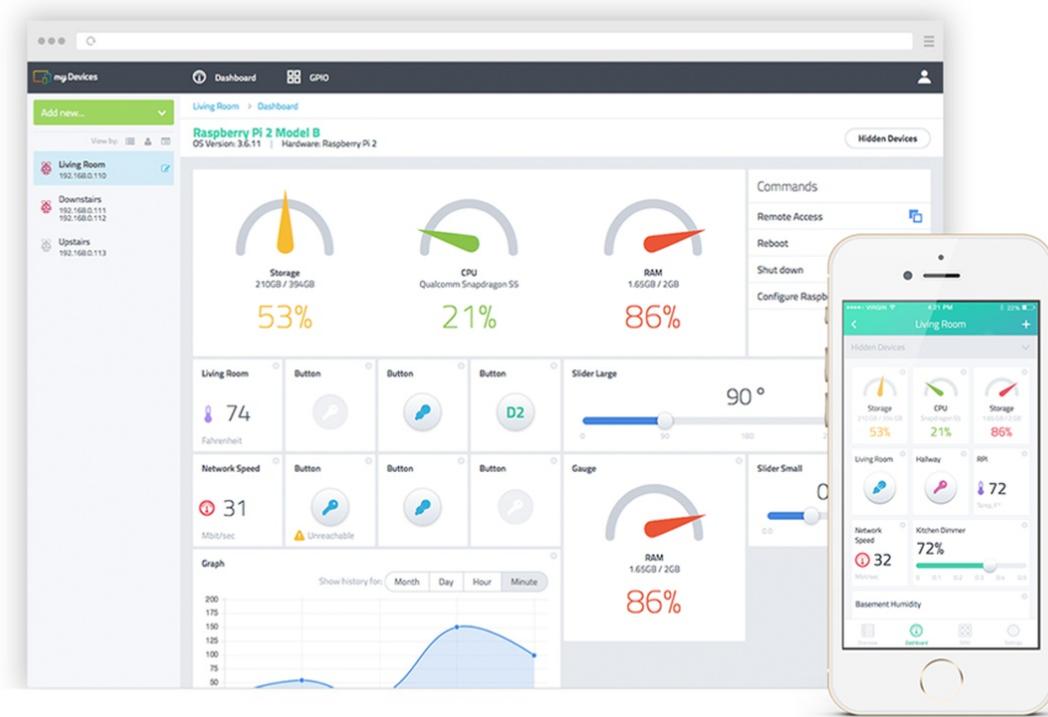


# The Things Network

The screenshot shows the The Things Network Console interface. At the top, there's a navigation bar with 'Person 2' and various icons. Below it, the main header reads 'THE THINGS NETWORK CONSOLE COMMUNITY EDITION'. The left sidebar shows a breadcrumb path: Applications > evt\_tnn\_integration > Devices > sodaqexplorerlondonoffice > Settings. The main content area is divided into two columns: 'DEVICE SETTINGS' (left) and 'SETTINGS' (right). The 'DEVICE SETTINGS' column has tabs for General and Location, with 'General' selected. The 'SETTINGS' column contains fields for Description (a text input field), Device EUI (a text input field containing '00 CC BB ED 78 79 26 CD'), Application EUI (a text input field containing '70 B3 D5 7E F0 00 64 68'), and Activation Method (a button group with 'OTAA' and 'ABP' options, where 'OTAA' is highlighted). There are also some small status indicators like '8 bytes' and a green checkmark.

The Things Network is building a network for the Internet of Things by creating abundant data connectivity, so applications and businesses can flourish. The technology we use is called LoRaWAN and it allows for things to talk to the internet without 3G or WiFi. So no WiFi codes and no mobile subscriptions. It features low battery usage, long range and low bandwidth. Perfect for the internet of things.

## myDevices Cayenne



Cayenne is the world's first drag and drop IoT project builder that empowers developers, designers and engineers to quickly prototype and share their connected device projects. Cayenne was designed to help users create Internet of Things prototypes and then bring them to production.

There are several major components in the platform:

**Cayenne Mobile Apps** – Remotely monitor and control your IoT projects from the Android or iOS Apps.

**Cayenne Online Dashboard** – Use customizable widgets to visualize data, set up rules, schedule events and more.

# LoRA

Point to point link on 433MHz

(ร่าง) ประกาศ กสทช. เรื่อง มาตรฐานทางเทคนิคของเครื่องโทรคมนาคม และอุปกรณ์ สำหรับเครื่องวิทยุคมนาคม ประเภท Radio Frequency Identification: RFID



30

The National Broadcasting and Telecommunications Commission

## สรุปของสาระสำคัญของร่างประกาศ

ประกาศ กทช. (ฉบับปัจจุบัน)		ร่าง ประกาศ กสทช. (ฉบับปรับปรุงแก้ไข)	
กทช. มท. 1010 – 2550		กสทช. มท. 1010 – 25xx	
2.1.4 ย่านความถี่วิทยุ 920 - 925 MHz		2.1.2 ย่านความถี่วิทยุ 920 - 925 MHz	
กำลังส่งสูงสุด	เงื่อนไข	กำลังส่งสูงสุด	เงื่อนไข
≤ 0.5 mW (e.i.r.p.)	ได้รับยกเว้นใบอนุญาตให้มีใช้ นำออก ซึ่งเครื่องวิทยุคมนาคม และตั้งสถานีวิทยุคมนาคม	≤ 0.5 mW (e.i.r.p.)	ได้รับยกเว้นใบอนุญาตให้มีใช้ นำออก ซึ่งเครื่องวิทยุคมนาคม และตั้งสถานีวิทยุคมนาคม
≤ 4 W (e.i.r.p.)	ต้องได้รับใบอนุญาตวิทยุ คมนาคมที่เกี่ยวข้อง	≤ 4 W (e.i.r.p.)	ต้องได้รับใบอนุญาตวิทยุ คมนาคมที่เกี่ยวข้อง
FCC Part 15.247		FCC Part 15.247	
FCC Part 15.249		FCC Part 15.249	

(ร่าง) ประกาศ กสทช. เรื่อง มาตรฐานทางเทคนิคของเครื่องโทรคมนาคม และอุปกรณ์ สำหรับเครื่องวิทยุคมนาคม ประเภท Radio Frequency Identification: RFID



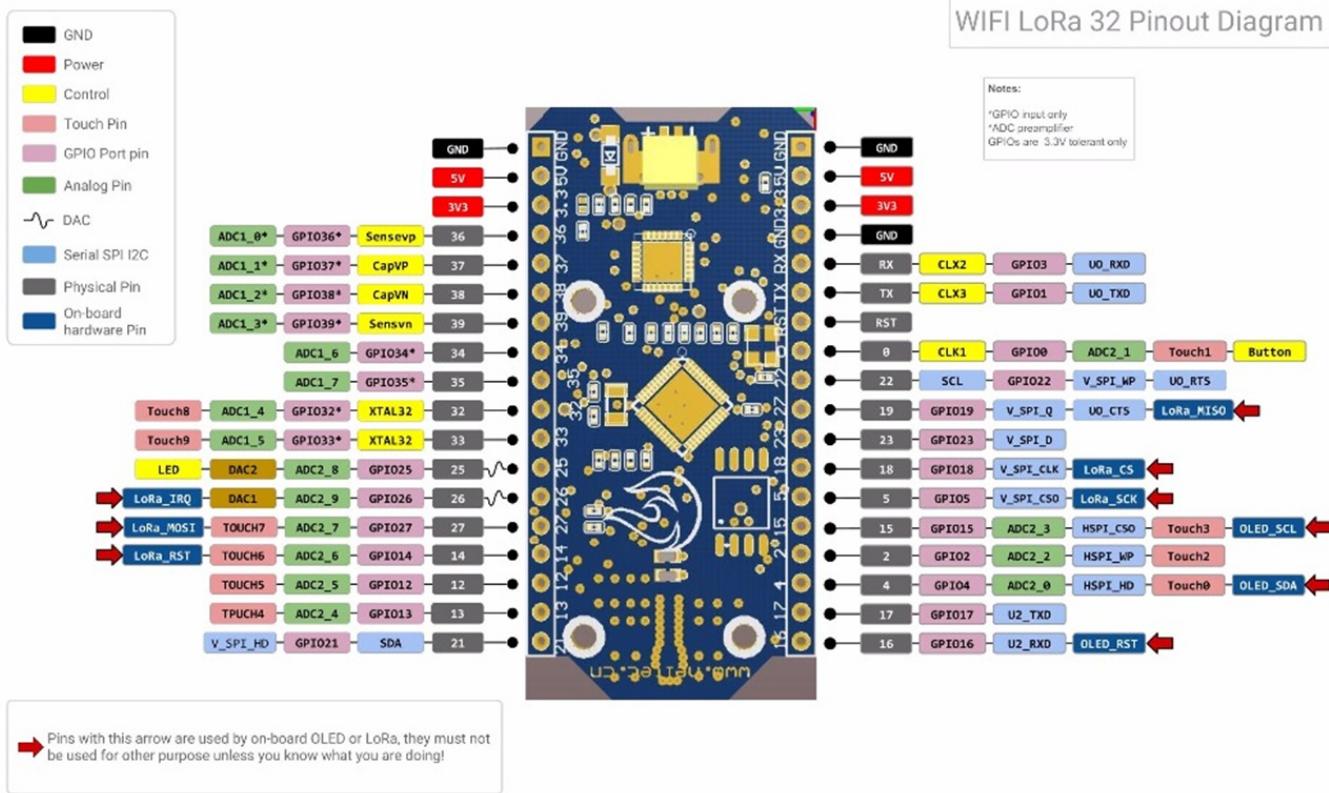
The National Broadcasting and Telecommunications Commission

## สรุปของสาระสำคัญของร่างประกาศ

ประกาศ กทช. (ฉบับปัจจุบัน)		ร่าง ประกาศ กสทช. (ฉบับปรับปรุงแก้ไข)	
กทช. มท. 1010 – 2550		กสทช. มท. 1010 – 25xx	
2.1.3 ย่านความถี่วิทยุ 433.05 – 434.79 MHz		2.1.3 ย่านความถี่วิทยุ 433.05 – 434.79 MHz	
กำลังส่งสูงสุด	เงื่อนไข	กำลังส่งสูงสุด	เงื่อนไข
≤10 mW(e.i.r.p.)	ได้รับยกเว้นใบอนุญาตให้ทำ มี ใช้ นำเข้า นำออก หรือ ค้าซึ่ง เครื่องวิทยุคมนาคม และตั้ง สถานีวิทยุคมนาคม	≤10 mW(e.i.r.p.)	ได้รับยกเว้นใบอนุญาตให้ทำ มี ใช้ นำเข้า นำออก หรือ ค้าซึ่ง เครื่องวิทยุคมนาคม และตั้ง สถานีวิทยุคมนาคม
ETSI EN 300 220-1		ETSI EN 300 220-1	

<b>Channel</b>	<b>Frequency (MHz)</b>	<b>Channel</b>	<b>Frequency (MHz)</b>	<b>Channel</b>	<b>Frequency (MHz)</b>
<b>1</b>	433.075	<b>24</b>	433.650	<b>47</b>	434.225
<b>2</b>	433.100	<b>25</b>	433.675	<b>48</b>	434.250
<b>3</b>	433.125	<b>26</b>	433.700	<b>49</b>	434.275
<b>4</b>	433.150	<b>27</b>	433.725	<b>50</b>	434.300
<b>5</b>	433.175	<b>28</b>	433.750	<b>51</b>	434.325
<b>6</b>	433.200	<b>29</b>	433.775	<b>52</b>	434.350
<b>7</b>	433.225	<b>30</b>	433.800	<b>53</b>	434.375
<b>8</b>	433.250	<b>31</b>	433.825	<b>54</b>	434.400
<b>9</b>	433.275	<b>32</b>	433.850	<b>55</b>	434.425
<b>10</b>	433.300	<b>33</b>	433.875	<b>56</b>	434.450
<b>11</b>	433.325	<b>34</b>	433.900	<b>57</b>	434.475
<b>12</b>	433.350	<b>35</b>	433.925	<b>58</b>	434.500
<b>13</b>	433.375	<b>36</b>	433.950	<b>59</b>	434.525
<b>14</b>	433.400	<b>37</b>	433.975	<b>60</b>	434.550
<b>15</b>	433.425	<b>38</b>	434.000	<b>61</b>	434.575
<b>16</b>	433.450	<b>39</b>	434.025	<b>62</b>	434.600
<b>17</b>	433.475	<b>40</b>	434.050	<b>63</b>	434.625
<b>18</b>	433.500	<b>41</b>	434.075	<b>64</b>	434.650
<b>19</b>	433.525	<b>42</b>	434.100	<b>65</b>	434.675
<b>20</b>	433.550	<b>43</b>	434.125	<b>66</b>	434.700
<b>21</b>	433.575	<b>44</b>	434.150	<b>67</b>	434.725
<b>22</b>	433.600	<b>45</b>	434.175	<b>68</b>	434.750
<b>23</b>	433.625	<b>46</b>	434.200	<b>69</b>	434.775

## Pinout



# WIFI\_LoRa\_32 ports

GPIO5 -- SX1278's SCK

## GPIO19 -- SX1278's MISO

GPIO27 -- SX1278's MOSI

GPIO18 -- SX1278's CS

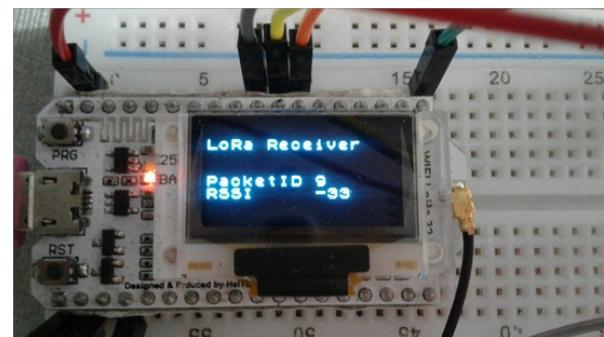
GPIO14 -- SX1278's RESET

GPIO26 -- SX1278's IRQ(Interrupt Request)

## EX18\_LoRaReceiver

ทดสอบรับข้อมูลผ่านเครือข่าย LoRa บนคลื่นความถี่ 433MHz แบบ point to point พิรุ่งแสดงผลค่า RSSI บน serial console

ถ้าส่ง package สำเร็จหลอดไฟ LED บนบอร์ดที่ต่อ กับขา GPIO25 จะกระพริบ



```
EX18_LoRaReceiver | Arduino 1.8.5

EX18_LoRaReceiver
#include <U8x8lib.h>
#include <LoRa.h>

String receivedText;
String receivedRssi;

// WIFI_LoRa_32 ports
// GPIO5 -- SX1278's SCK
// GPIO19 -- SX1278's MISO
// GPIO27 -- SX1278's MOSI
// GPIO18 -- SX1278's CS
// GPIO14 -- SX1278's RESET
// GPIO26 -- SX1278's IRQ(Interrupt Request)

#define SS      18
#define RST     14
#define DI0     26
#define BAND   433125000

// the OLED used
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(15,4,16);

void setup() {
    pinMode(25,OUTPUT); //Send success, LED will bright 1 second
    SPI.begin(5, 19, 27, 18);
}

Done uploading.
Leaving...
Hard resetting...

63
```

Heltec\_WIFI\_LoRa\_32, 80MHz, 921600 on /dev/cu.SLAB\_USBtoUART

## EX18\_LoRaReceiver

The screenshot shows a terminal window titled "/dev/cu.SLAB\_USBtoUART". The window displays a series of received LoRa packets, each consisting of a string of characters followed by "with RSSI" and a numerical value. The packets are numbered from 20 to 49. The terminal interface includes a text input field at the top, a "Send" button, and several configuration options at the bottom: "Autoscroll" (checked), "Both NL & CR" (selected), "115200 baud" (selected), and a "Clear output" button.

```
Received packet 'hello 20' with RSSI -28
Received packet 'hello 21' with RSSI -28
Received packet 'hello 22' with RSSI -29
Received packet 'hello 23' with RSSI -29
Received packet 'hello 24' with RSSI -28
Received packet 'hello 25' with RSSI -27
Received packet 'hello 26' with RSSI -28
Received packet 'hello 27' with RSSI -28
Received packet 'hello 28' with RSSI -28
Received packet 'hello 29' with RSSI -28
Received packet 'hello 30' with RSSI -39
Received packet 'hello 31' with RSSI -29
Received packet 'hello 32' with RSSI -29
Received packet 'hello 33' with RSSI -30
Received packet 'hello 34' with RSSI -28
Received packet 'hello 35' with RSSI -29
Received packet 'hello 36' with RSSI -27
Received packet 'hello 37' with RSSI -28
Received packet 'hello 38' with RSSI -27
Received packet 'hello 39' with RSSI -29
Received packet 'hello 40' with RSSI -29
Received packet 'hello 41' with RSSI -29
Received packet 'hello 42' with RSSI -29
Received packet 'hello 43' with RSSI -29
Received packet 'hello 44' with RSSI -28
Received packet 'hello 45' with RSSI -28
Received packet 'hello 46' with RSSI -28
Received packet 'hello 47' with RSSI -38
Received packet 'hello 48' with RSSI -28
Received packet 'hello 49' with RSSI -39
```

Autoscroll      Both NL & CR      115200 baud      Clear output

## EX19\_LoRaSender

ทดสอบส่งข้อมูลผ่านเครือข่าย LoRa บนคลื่นความถี่ 433MHz แบบ point to point พร้อมแสดงผลค่า RSSI บน serial console

ถ้าส่ง package สำเร็จหลอดไฟ LED บนบอร์ดที่ต่อ กับขา GPIO25 จะกระพริบ



```
EX19_LoRaSender | Arduino 1.8.5

EX19_LoRaSender
/*
 * EX19_LoRaSender
 * ทดสอบส่งข้อมูลผ่านเครือข่าย LoRa พร้อมแสดงค่า RSSI
 * บนคลื่นความถี่ 433MHz
 */

#include <U8x8lib.h>
#include <LoRa.h>

String receivedText;
String receivedRssi;

// WIFI_LoRa_32 ports

// GPIO5 -- SX1278's SCK
// GPIO19 -- SX1278's MISO
// GPIO27 -- SX1278's MOSI
// GPIO18 -- SX1278's CS
// GPIO14 -- SX1278's RESET
// GPIO26 -- SX1278's IRQ(Interrupt Request)

#define SS      18
#define RST    14
#define DI0    26
#define BAND  433.125E6
#define PABOOST true|


Done Saving.
Leaving...
Hard resetting...

26
```

The screenshot shows a terminal window titled "/dev/cu.SLAB\_USBtoUART". The window displays a series of log messages indicating the transmission of 28 LoRa packets. The messages are as follows:

```
LoRa Initial OK!
Sending packet: 0
Sending packet: 1
Sending packet: 2
Sending packet: 3
Sending packet: 4
Sending packet: 5
Sending packet: 6
Sending packet: 7
Sending packet: 8
Sending packet: 9
Sending packet: 10
Sending packet: 11
Sending packet: 12
Sending packet: 13
Sending packet: 14
Sending packet: 15
Sending packet: 16
Sending packet: 17
Sending packet: 18
Sending packet: 19
Sending packet: 20
Sending packet: 21
Sending packet: 22
Sending packet: 23
Sending packet: 24
Sending packet: 25
Sending packet: 26
Sending packet: 27
```

At the bottom of the terminal window, there are several control buttons: "Autoscroll" (checked), "Both NL & CR", "115200 baud", and "Clear output".

## EX20\_LoRaReceiverInterrupt

ทดสอบส่งข้อมูลผ่านเครือข่าย LoRa บนคลื่นความถี่ 433MHz แบบ point to point พร้อมแสดงผลค่า RSSI บน serial console

ถ้าส่ง package สำเร็จหลอดไฟ LED บนบอร์ดที่ต่อ กับขา GPIO25 จะกระพริบ หลังจากที่ได้รับ package



```
EX20_LoRaReceiverInterrupt | Arduino 1.8.5

/*
 * EX20_LoRaReceiverInterrupt
 * ทดสอบรับข้อมูลผ่านเครือข่าย LoRa พร้อมแสดงค่า RSSI
 * บนคลื่นความถี่ 433MHz
 */

#include <LoRa.h>
#include <U8x8lib.h>

String receivedText;
String receivedRssi;

// Pin definition of WIFI LoRa 32
#define SCK 5 // GPIO5 -- SX127x's SCK
#define MISO 19 // GPIO19 -- SX127x's MISO
#define MOSI 27 // GPIO27 -- SX127x's MOSI
#define SS 18 // GPIO18 -- SX127x's CS
#define RST 14 // GPIO14 -- SX127x's RESET
#define DI0 26 // GPIO26 -- SX127x's IRQ(Interrupt Request)

#define BAND 433.125E6 //you can set band here directly,e.g. 868E6,915E6
#define PABOOST true

// the OLED used
U8X8_SSD1306_128X64_NONAME_SW_I2C u8x8(15,4,16);

void setup() {
  Done Saving.
  Leaving...
  Hard resetting...
}

8
Heltec_WIFI_LoRa_32, 80MHz, 921600 on /dev/cu.SLAB_USBtoUART
```

## EX20\_LoRaReceiverInterrupt

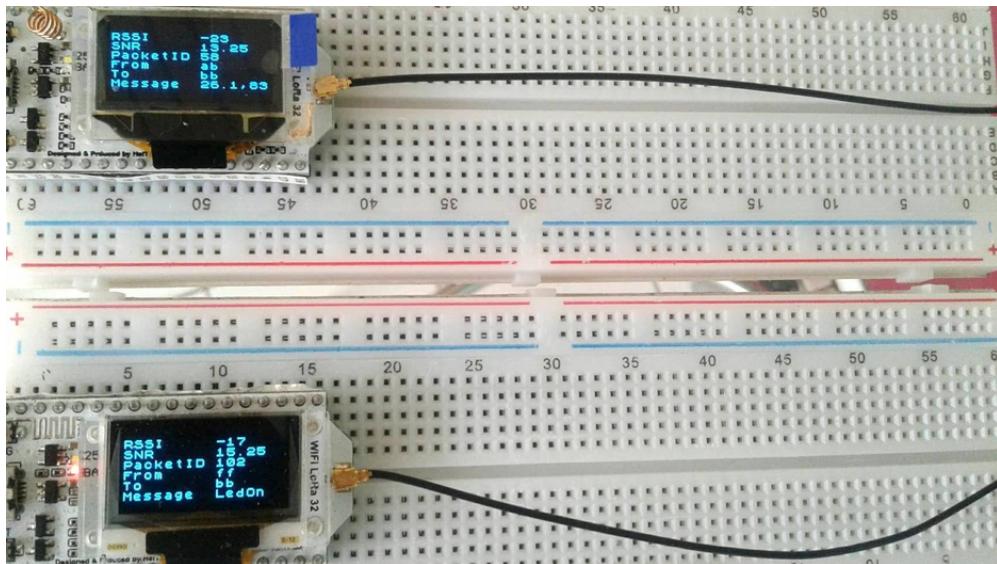
The screenshot shows a terminal window titled '/dev/cu.SLAB\_USBtoUART'. The window displays a continuous stream of received LoRa packets, each consisting of the text 'Received packet' followed by a string like 'hello 315' and an RSSI value like '-26'. The packets are numbered from 315 to 343. The terminal interface includes a 'Send' button at the top right, and at the bottom, there are checkboxes for 'Autoscroll' and 'Clear output', and dropdown menus for 'Both NL & CR' and '115200 baud'.

```
Received packet 'hello 315' with RSSI -26
Received packet 'hello 316' with RSSI -26
Received packet 'hello 317' with RSSI -26
Received packet 'hello 318' with RSSI -27
Received packet 'hello 319' with RSSI -26
Received packet 'hello 320' with RSSI -26
Received packet 'hello 321' with RSSI -26
Received packet 'hello 322' with RSSI -27
Received packet 'hello 323' with RSSI -26
Received packet 'hello 324' with RSSI -27
Received packet 'hello 325' with RSSI -26
Received packet 'hello 326' with RSSI -26
Received packet 'hello 327' with RSSI -27
Received packet 'hello 328' with RSSI -27
Received packet 'hello 329' with RSSI -27
Received packet 'hello 330' with RSSI -27
Received packet 'hello 331' with RSSI -27
Received packet 'hello 332' with RSSI -27
Received packet 'hello 333' with RSSI -26
Received packet 'hello 334' with RSSI -27
Received packet 'hello 335' with RSSI -26
Received packet 'hello 336' with RSSI -26
Received packet 'hello 337' with RSSI -27
Received packet 'hello 338' with RSSI -26
Received packet 'hello 339' with RSSI -27
Received packet 'hello 340' with RSSI -26
Received packet 'hello 341' with RSSI -26
Received packet 'hello 342' with RSSI -26
Received packet 'hello 343' with RSSI -27
```

Autoscroll      Both NL & CR      115200 baud      Clear output

## EXT21\_LoRaMultiCommunication\_OLED\_Client

การรับและส่งข้อมูลผ่านเครือข่าย LoRa แบบ Point to MultiPoint ที่สามารถระบุต้นทางได้ อุปกรณ์ทุกตัวต้องใช้ความถี่เดียวกัน



```
EXT21_LoRaMultipleCommunication_OLED_Client | Arduino 1.8.5

EXT21_LoRaMultipleCommunication_OLED_Client §

/*
 * EXT21_LoRaMultiCommunication_OLED_Client
 * LoRa Multiple Communication
 *
 * การรับและส่งข้อมูลผ่านเครือข่าย LoRa และระบุต้นทางและปลายทาง
 * สามารถส่งแบบ broadcast ไปยัง device ทุกด้วยระบุปลายทางเป็น 0xFF
 * อุปกรณ์ทุกตัวต้องใช้ความถี่เดียวกัน
 */

#include <U8x8lib.h>
#include <LoRa.h>

String receivedText;
String receivedRssi;

// Pin definition of WIFI LoRa 32
// Heltec AutoMation 2017 support@heltec.cn
#define SCK    5 // GPIO5 -- SX1278's SCK
#define MISO   19 // GPIO19 -- SX1278's MISO
#define MOSI   27 // GPIO27 -- SX1278's MOSI
#define SS     18 // GPIO18 -- SX1278's CS
#define RST   14 // GPIO14 -- SX1278's RESET
#define DI0   26 // GPIO26 -- SX1278's IRQ(Interrupt Request)

#define BAND  433E6 //you can set band here directly,e.g. 868E6,915E6
#define PABOOST true

Done uploading.

Leaving...
Hard resetting...

14
```

## EXT21\_LoRaMultiCommunication\_OLED\_Client

```
RSSI: -17
Snr: 6.50

Sending 26.1,83
Received from: 0xbb
Sent to: 0xff
Message ID: 168
Message length: 5
Message: LedOn
RSSI: -17
Snr: 14.25

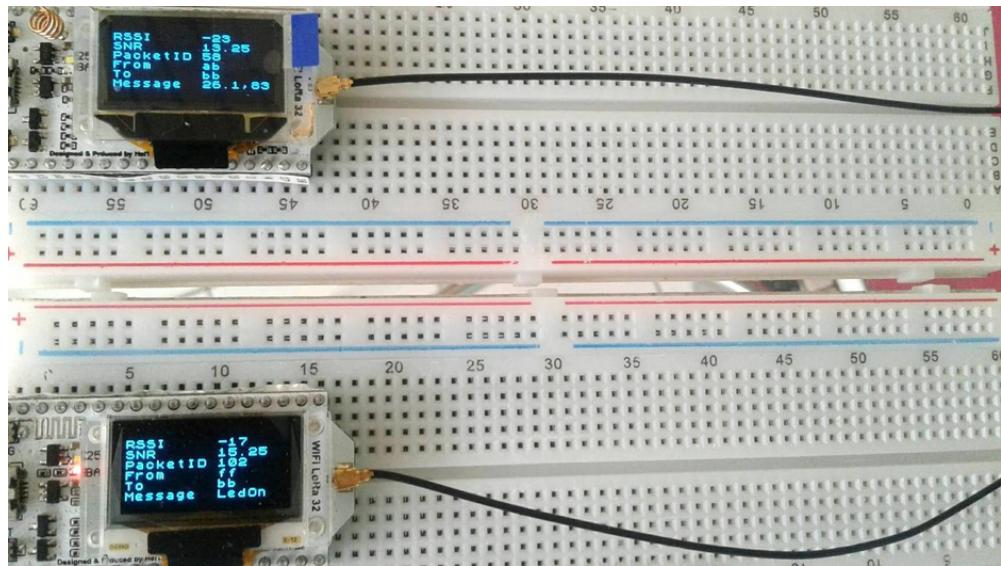
Sending 26.1,83
Received from: 0xbb
Sent to: 0xff
Message ID: 169
Message length: 5
Message: LedOn
RSSI: -18
Snr: 15.25

Sending 26.1,83
Received from: 0xbb
Sent to: 0xff
Message ID: 170
Message length: 5
Message: LedOn
RSSI: -17
Snr: 14.50

 Autoscroll      Both NL & CR      115200 baud      Clear output
```

## EXT22\_LoRaMultiCommunication\_OLED\_Base

การรับและส่งข้อมูลผ่านเครือข่าย LoRa แบบ Point to MultiPoint ที่สามารถระบุต้นทางและปลายทาง สามารถส่งแบบ broadcast ไปยัง device ทุกตัวได้โดยระบุปลายทางเป็น 0xFF อุปกรณ์ทุกตัวต้องใช้ความถี่เดียวกัน



```
EXT22_LoRaMultipleCommunication_OLED_Base | Arduino 1.8.5

EXT22_LoRaMultipleCommunication_OLED_Base §

/*
 * EXT22_LoRaMultipleCommunication_OLED_Base |
 * LoRa Multiple Communication
 *
 * การรับและส่งข้อมูลผ่านเครือข่าย LoRa แบบระบุต้นทางและปลายทาง
 * สามารถส่งแบบ broadcast ไปยัง device ทุกตัวได้โดยระบุปลายทางเป็น 0xFF
 * อุปกรณ์ทุกตัวต้องใช้ความถี่เดียวกัน
 */

#include <U8x8lib.h>
#include <LoRa.h>

String receivedText;
String receivedRssi;

// Pin definition of WIFI LoRa 32
// HelTec AutoMation 2017 support@heltec.cn
#define SCK      5    // GPIO5 -- SX1278's SCK
#define MISO     19   // GPIO19 -- SX1278's MISO
#define MOSI     27   // GPIO27 -- SX1278's MOSI
#define SS       18    // GPIO18 -- SX1278's CS
#define RST      14    // GPIO14 -- SX1278's RESET
#define DI0      26    // GPIO26 -- SX1278's IRQ(Interrupt Request)

#define BAND    433E6 //you can set band here directly,e.g. 868E6,915E6
#define PABOOST true

Done Saving.
```

## EXT22\_LoRaMultiCommunication\_OLED\_Base

The screenshot shows a terminal window titled '/dev/cu.SLAB\_USBtoUART' displaying a series of received LoRa messages. The window includes a text input field at the top, a scrollable message list below, and a control panel at the bottom with checkboxes for 'Autoscroll', baud rate selection, and a 'Clear output' button.

```
Message ID: 46
Message length: 7
Message: 26.1,83
RSSI: -27
Snr: 9.75

Received from: 0xab
Sent to: 0xbb
Message ID: 47
Message length: 7
Message: 26.1,83
RSSI: -26
Snr: 10.00

Received from: 0xab
Sent to: 0xbb
Message ID: 48
Message length: 7
Message: 26.1,83
RSSI: -27
Snr: 10.00

Received from: 0xab
Sent to: 0xbb
Message ID: 49
Message length: 7
Message: 26.1,83
RSSI: -27
Snr: 9.75
```

Autoscroll      Both NL & CR      115200 baud      Clear output