

การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร M2M - Intelligence Machine Control

1/4 - Arduino and Arduino IDE

- Industrial 4.0, IoTs, IIoTs and Disruptive Technology
 - Arduino Microcontroller การติดตั้ง Arduino IDE และเริ่มต้นใช้งาน
 - การโปรแกรมเพื่อควบคุมล็อกงาน อินพุต/เอาต์พุต ของ ESP-32
 - การโปรแกรมเพื่อใช้งานแบบ All Over IP
 - การโปรแกรมเพื่อใช้งานแบบ IoTs ผ่าน MQTT โปรโตคอล
 - คำถามท้ายบทเพื่อทดสอบความเข้าใจ

1/6 -- Industrial 4.0, IoTs, IIoTs and Disruptive Technology

< เอกสารประกอบ “M2M-D11 -- Industrial 4.0, IoTs and IIoTs.pdf” >

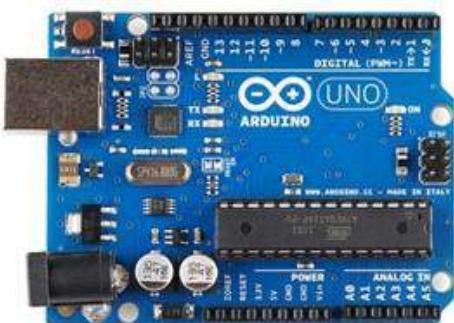
2/6 -- Arduino Microcontroller, การติดตั้ง Arduino IDE และเริ่มต้นใช้งาน

www.arduinotronics.com/article/9/มาตรฐานหน้าต่างของบอร์ด-arduino-rุ่นทางๆ-กันน้ำกีว่า
<https://www.cmmakerclub.com/2015/06/esp8266/เริ่มต้นใช้งาน-esp8266-ผ่าน-arduino-ide/>
m/product/1057/wemos-d1-wifi-nodemcu-arduino-wifi-uno-board-esp8266-arduino-ide

บอร์ด Arduino ถือว่าเป็น Open Hardware Platform ที่ถูกพัฒนาขึ้นโดยมี Micro-controller ของ Atmel เป็นหัวใจหลัก บอร์ด Arduino ที่ผลิตออกมากำลังน้ำย่อยในปัจจุบันมีทั้งหมด 20 รุ่น มาลองศึกษาดูครับว่าแต่ละรุ่นมีความสามารถและการใช้งานต่างกันไป

2.1 Arduino Board (Official from Arduino.cc) มีหลักหลายรุ่นที่นำเสนอใน ได้แก่

1. Arduino Uno R3 เป็นบอร์ด Arduino ที่ได้รับความนิยมมากที่สุด เนื่องจากราคาไม่แพง ส่วนใหญ่โปรเจคและ Library ต่างๆ ที่พัฒนาขึ้นมา Support จะอ้างอิงกับบอร์ดนี้เป็นหลัก และข้อดีอีกอย่างคือ กรณีที่ MCU เสีย ผู้ใช้งานสามารถซื้อมาเปลี่ยนเองได้ง่าย



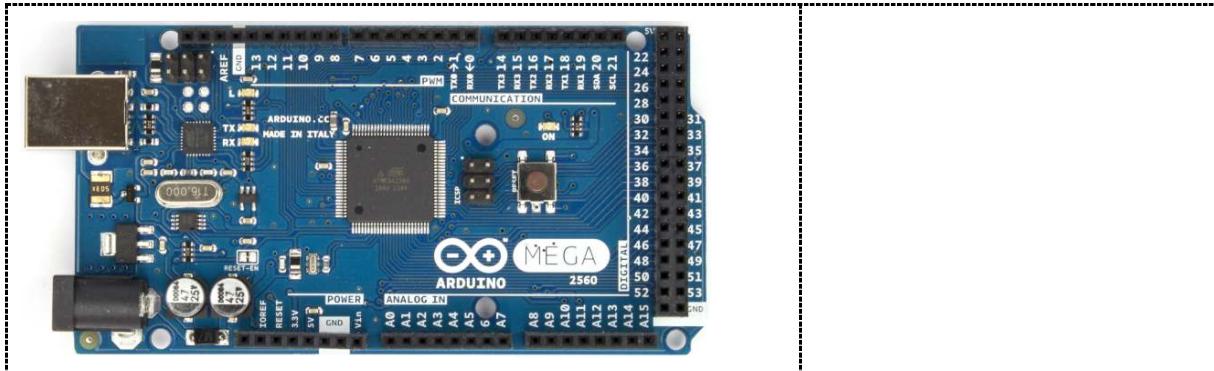
Arduino Uno R3



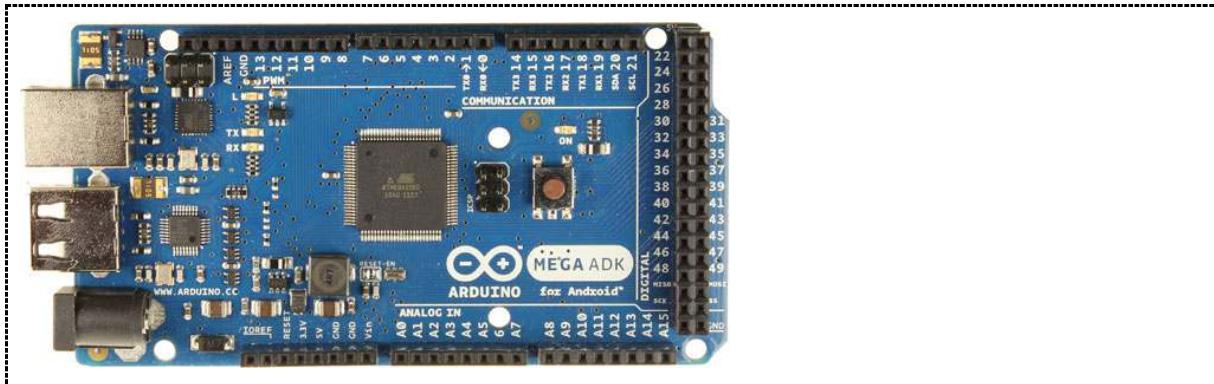
Arduino Uno SMD

2. Arduino Uno SMD เป็นบอร์ดที่มีคุณสมบัติและการทำงานเหมือนกับบอร์ด Arduino UNO R3 ทุกประการ แต่จะแตกต่างกับที่ Package ของ MCU ซึ่งบอร์ดนี้จะมี MCU ที่เป็น Package SMD (Arduino UNO R3 มี MCU ที่เป็น Package DIP)

3. Arduino Mega 2560 R3 เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino Uno R3 ทำให้สามารถเขียนโค้ดโปรแกรมเข้าไปได้มากกว่า ในความเร็วของ MCU ที่เท่ากัน



4. Arduino Mega ADK เป็นบอร์ดที่ออกแบบมาให้บอร์ด Mega 2560 R3 สามารถติดต่อ กับอุปกรณ์ Android Deviceผ่านพอร์ต USB Host ของบอร์ดได้ (ดูรายละเอียดลิ้นค่า)



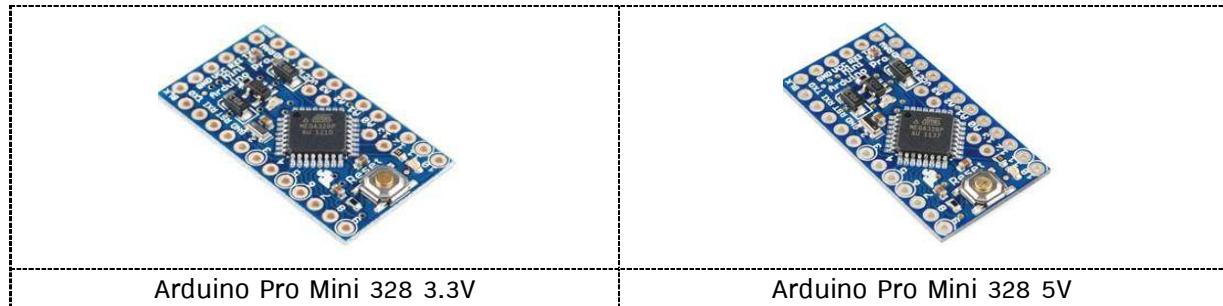
5. Arduino Leonardo การทำงานจะคล้ายกับบอร์ด Arduino Uno R3 แต่มีการเปลี่ยน MCU ตัวใหม่เป็น ATmega32U4 ซึ่งมีโมดูลพอร์ต USB มาด้วยบนชิป (แตกต่างจากบอร์ด Arduino UNO R3 หรือ Arduino Mega 2560 ที่ต้องใช้ชิป ATmega16U2 รวมกับ Atmega328 ในการเชื่อมต่อกับพอร์ต USB)



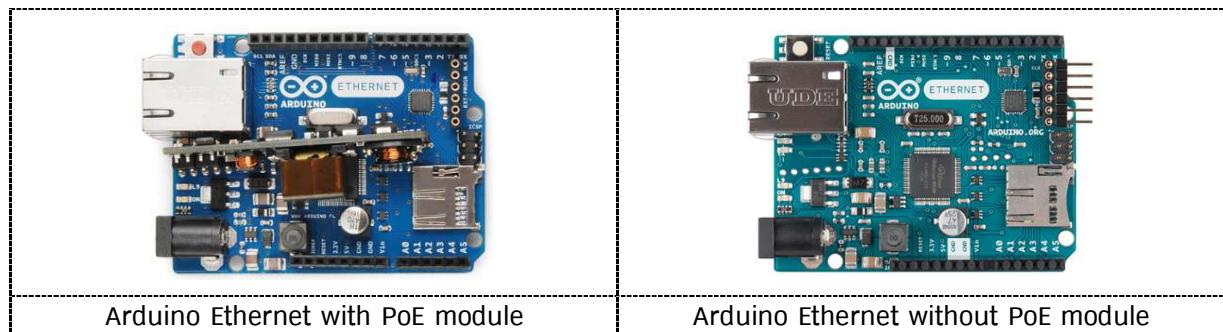
ข้อควรระวัง: เนื่องจาก MCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจจะทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ด Arduino Leonardo ไม่ได้ ซึ่งงานจำเป็นต้องตรวจสอบก่อนใช้งาน

6. Arduino Pro Mini 328 3.3V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 ซึ่งจะคล้ายกับบอร์ด Arduino Mini 05 แต่บอร์ดจะมี Regulator 3.3 V ชุดเดียวเท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 3.3V

7. Arduino Pro Mini 328 5V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 เช่นเดียวกับบอร์ด Arduino Mini 05 แต่บอร์ดจะมี Regulator 5V ชุดเดียวเท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 5V

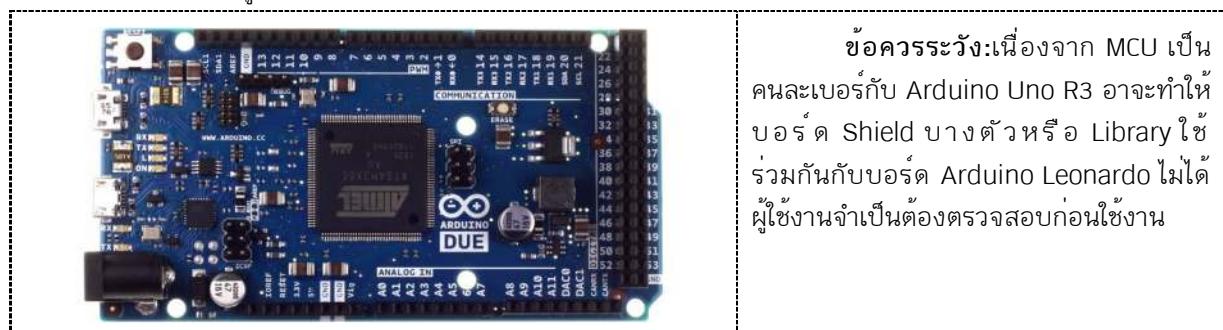


8. Arduino Ethernet with PoE module เป็นบอร์ด Arduino ที่ใช้ MCU เบอร์เดียวกับ Arduino Uno SMD ในบอร์ดมีชิป Ethernet และช่องสำหรับเสียบ SD Card รวมทั้งโมดูล POE ทำให้บอร์ดนี้สามารถใช้แหล่งจ่ายไฟจากสาย LAN ได้โดยตรง โดยไม่ต้องต่อ Adapter เพิ่ม แต่บอร์ด Arduino Ethernet with PoE module นี้จะไม่มีพอร์ต USB ทำให้เวลาโปรแกรมต้องต่อบอร์ด USB toSerial Converter เพิ่มเติม



9. Arduino Ethernet without PoE module บอร์ดนี้จะตัดโมดูล POE ออกไป ต้องใช้ไฟจากพอร์ต Power Jack เท่านั้น คุณสมบัติอื่นๆ จะเหมือนกับบอร์ด Arduino Ethernet with PoE module

10. Arduino Due เป็นบอร์ด Arduino ที่เปลี่ยนอิปิส MCU ใหม่ ซึ่งมาจากเดิมเป็นตระกูล AVR เป็นตระกูล ARM Cortex-M3) แทน ทำให้การประมวลผลเร็วขึ้น แต่ยังคงรูปแบบโค้ดโปรแกรมของ Arduino ที่ง่ายอยู่



2.2 การติดตั้งโปรแกรม Arduino IDE และการเพิ่ม ESP32 Board

Arduino จะใช้โปรแกรมที่เรียกว่า Arduino IDE ในการเขียนโปรแกรม และคอมไฟล์ลงบอร์ด โดยขนาดของ โปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่ โค้ด Arduino เข้าถึงผ่านพังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR

2.2.1 การดาวน์โหลดโปรแกรม Arduino IDE

ดาวน์โหลดไฟล์โปรแกรมได้จากเว็บไซต์ <http://www.arduino.cc/en/Main/Software> เลือกระบบปฏิบัติการ ที่ต้องการจะติดตั้ง (ตัวอย่างผมใช้ Windows จึงเลือก Windows Installer) @20210701 – Ver 1.18.15



จากหน้าจัดแสดงหน้าเขินให้ร่วมบริจาค หากไม่ต้องการบริจาคสามารถคลิกปุ่ม JUST DOWNLOAD เพื่อเริ่ม ดาวน์โหลดโปรแกรมได้เลย

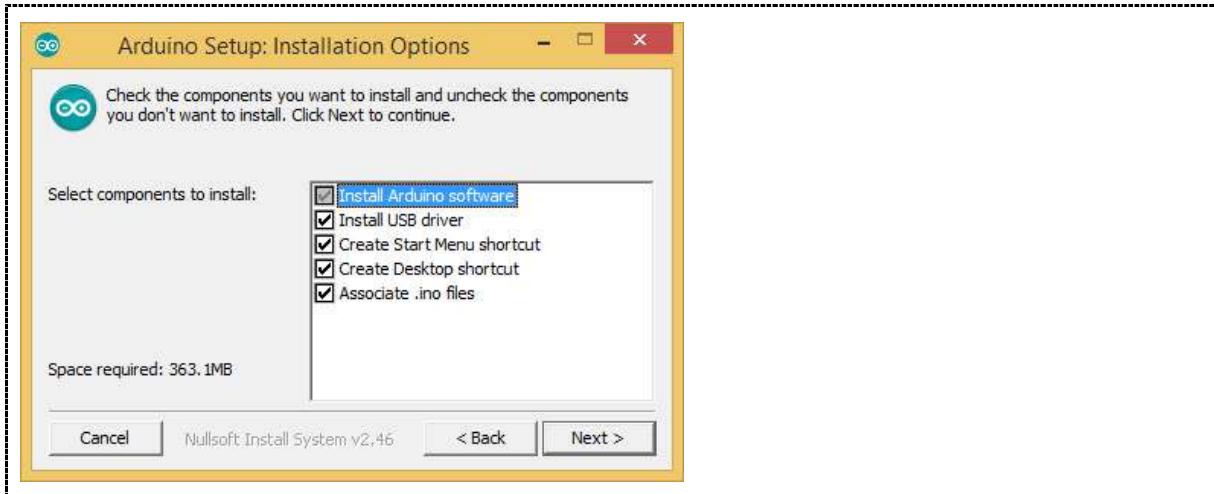


2.2.2 การติดตั้งโปรแกรม Arduino IDE

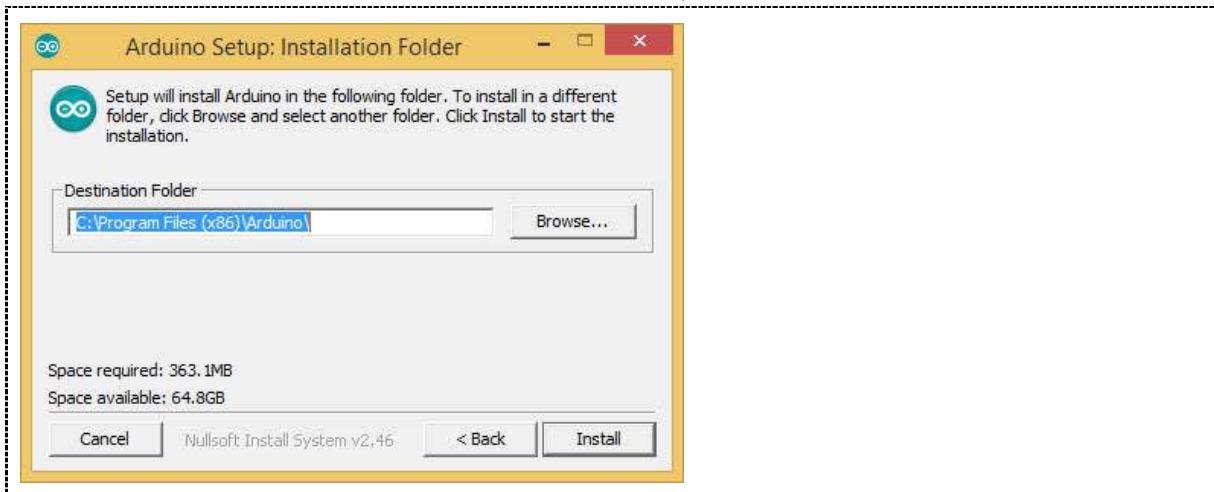
เมื่อดาวน์โหลดเสร็จแล้วให้เปิดไฟล์ติดตั้งขึ้นมาได้เลย กดปุ่ม I Agree ได้เลย



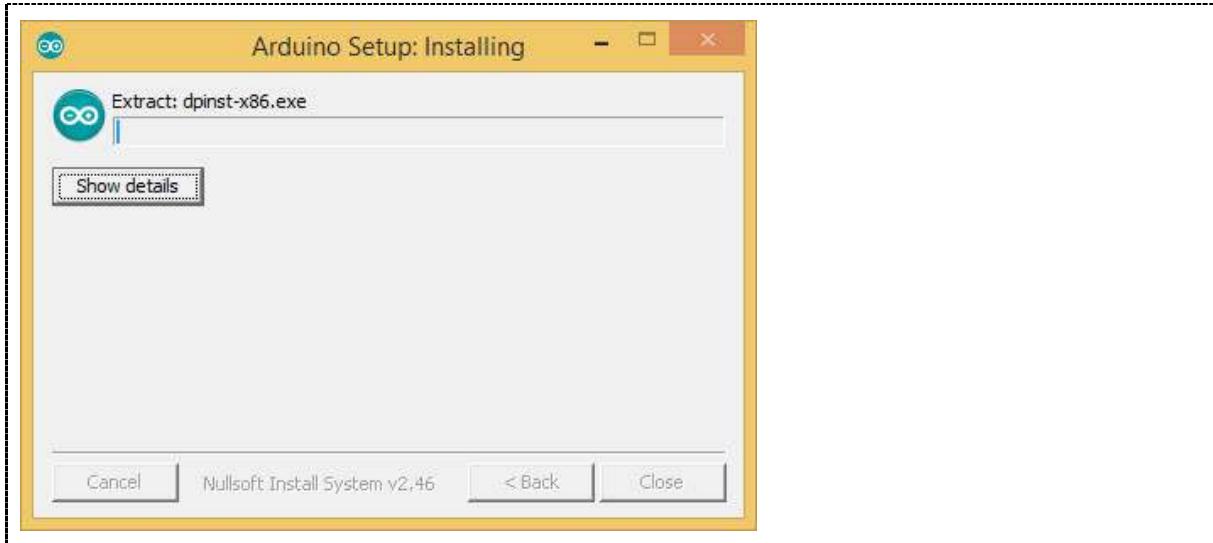
มีตัวเลือกให้เลือกติดตั้ง แนะนำให้เลือกทั้งหมด (ค่าเริมต้นคือเลือกทั้งหมด) และคลิกปุ่ม Next >



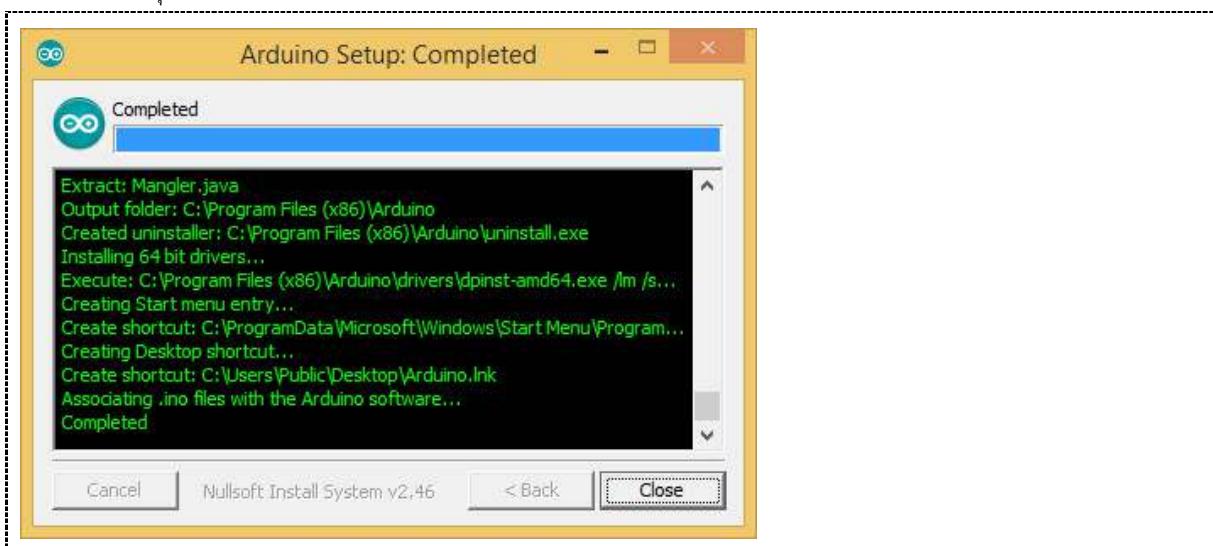
เลือกไฟล์เดอร์ติดตั้งโปรแกรม หากไม่ต้องการแก้ไขคลิกปุ่ม Install ได้เลย



รอจนกว่าโปรแกรมจะติดตั้งเสร็จล้วน



เมื่อขึ้นคำว่า Completed หมายถึงการติดตั้งเสร็จสมบูรณ์แล้ว
คลิกปุ่ม Close เพื่อปิดโปรแกรมลงไปได้เลย

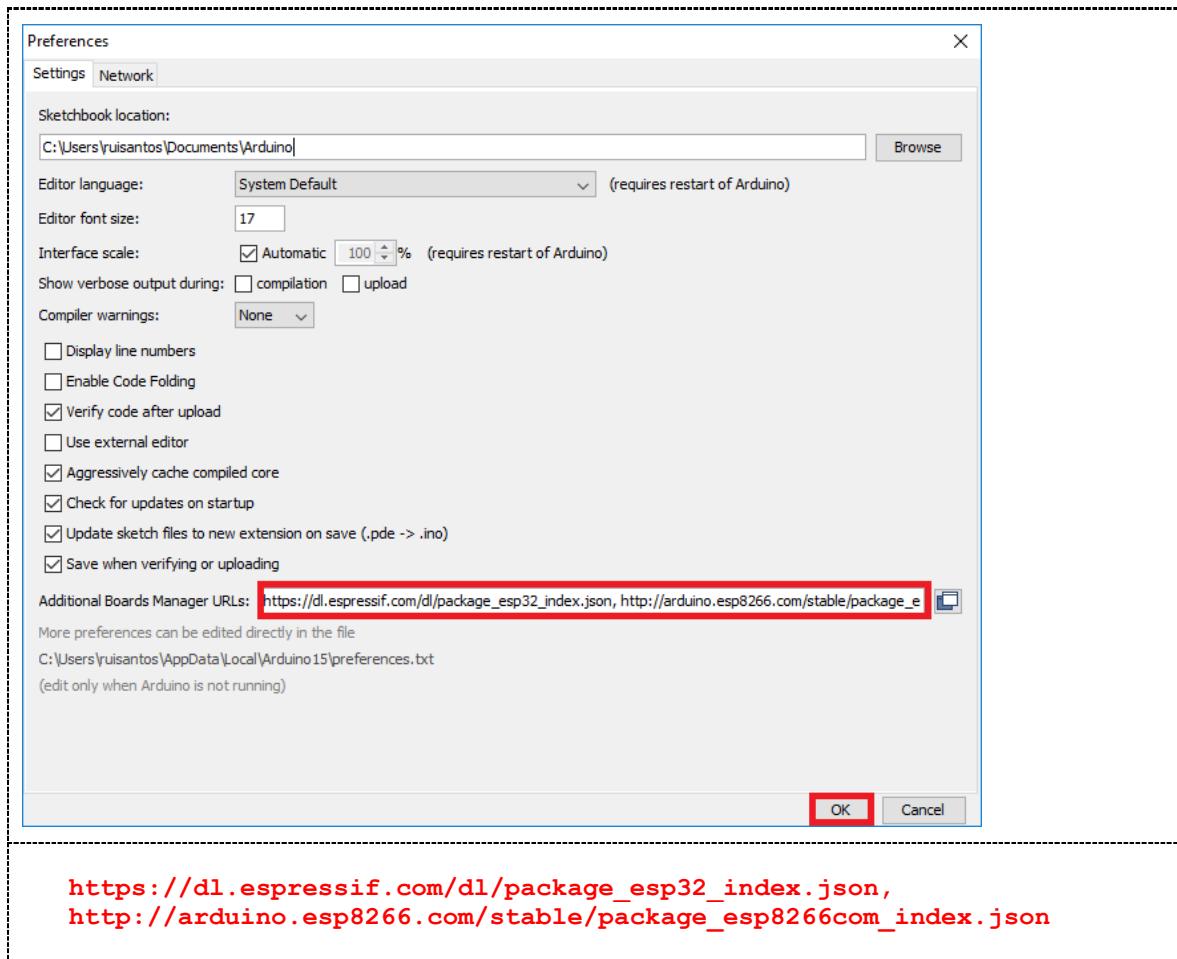


หน้าเดสท็อปจะมีไอคอนโปรแกรม Arduino ขึ้นมาแล้ว

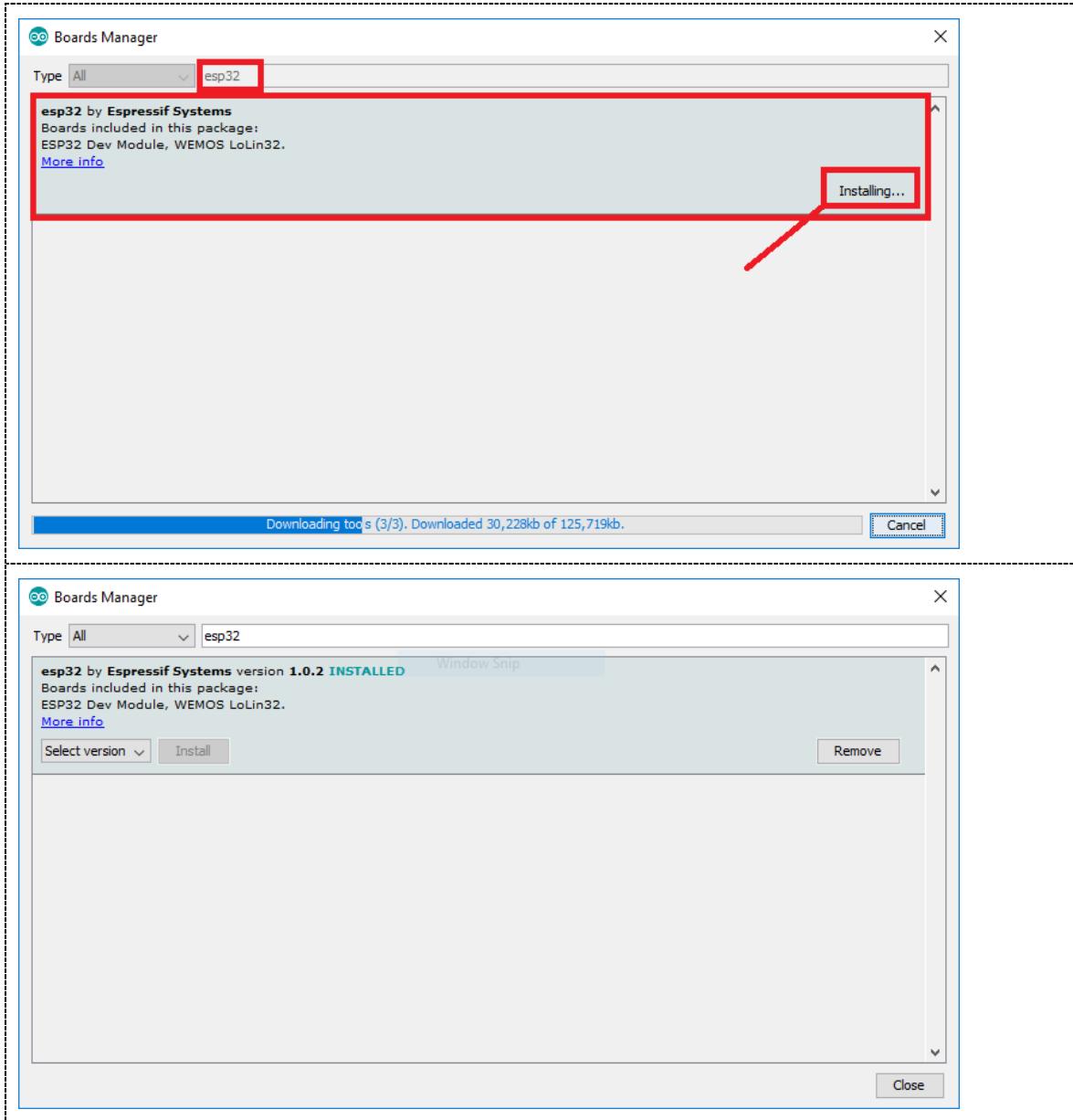


2.2.3 Add ESP32 Board ตามขั้นตอนดังนี้

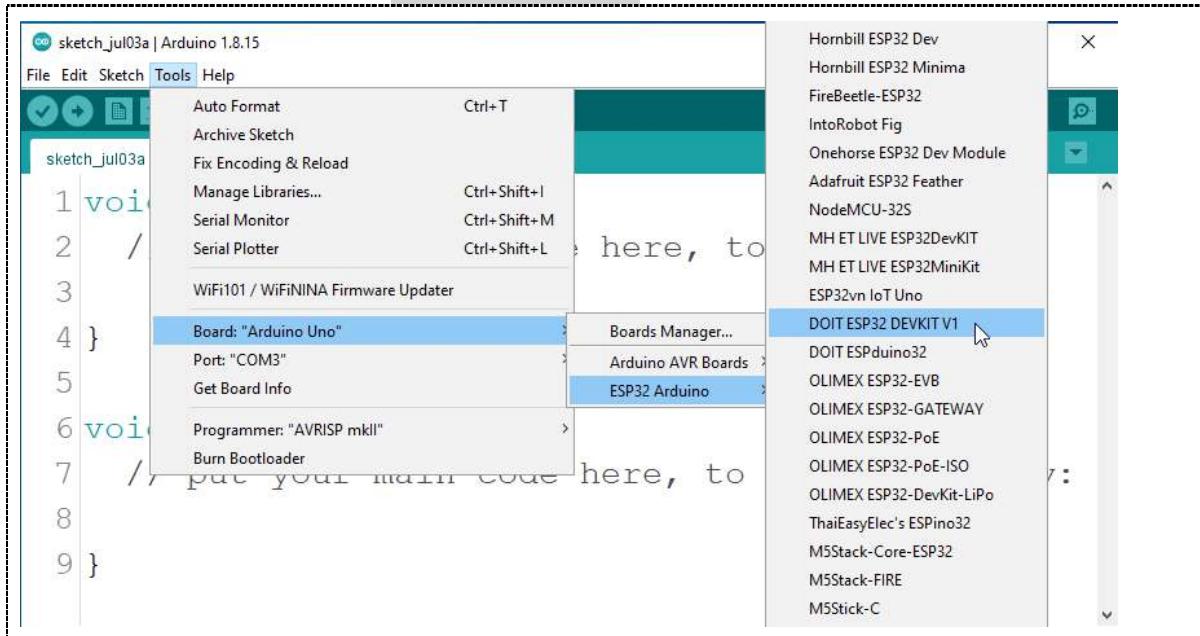
- <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>
- เข้าเมนู File >> Preferences จะขึ้นหน้าต่าง Preferences ให้สังเกตในช่อง Additional Board Manager URLs:
- เลือก URL >> ลงใน Additional Board Manager URLs: ดังนี้
https://dl.espressif.com/dl/package_esp32_index.json
- แล้วกด OK



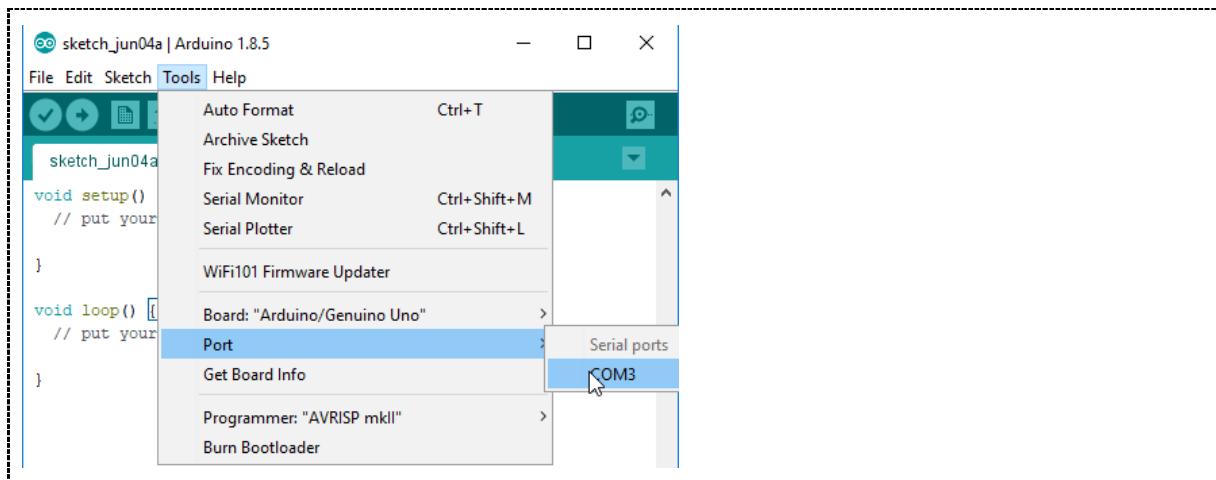
- ไปที่ Menu Tools → Board : “xxxxx” → Board Manager
- เลือกบอร์ด esp32 ของ ESP32 by Espressif Systems → กด Install



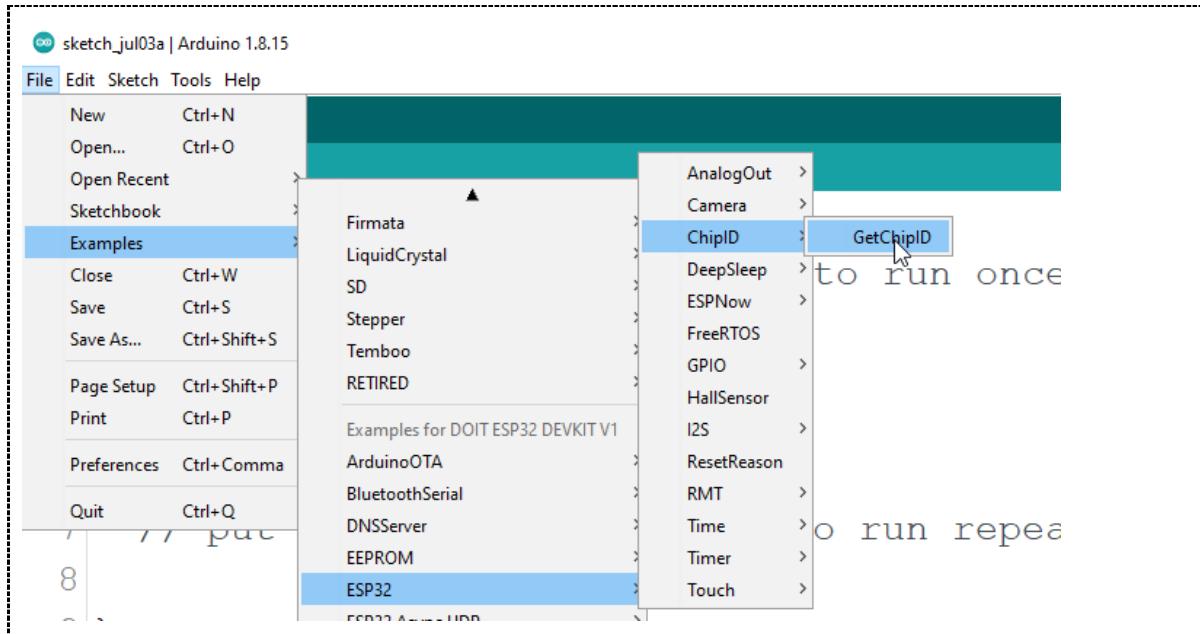
- เมื่อติดตั้งเสร็จ เข้าไปที่ Tools->Boards ลองเช็คดูว่า Arduino IDE รองรับการใช้งานร่วมกับ ESP32 แล้วหรือไม่ ถ้าเจอหัวข้อ DOIT ESP32 DEVKIT1 แสดงว่าสามารถใช้งานได้ ESP32 ได้แล้ว



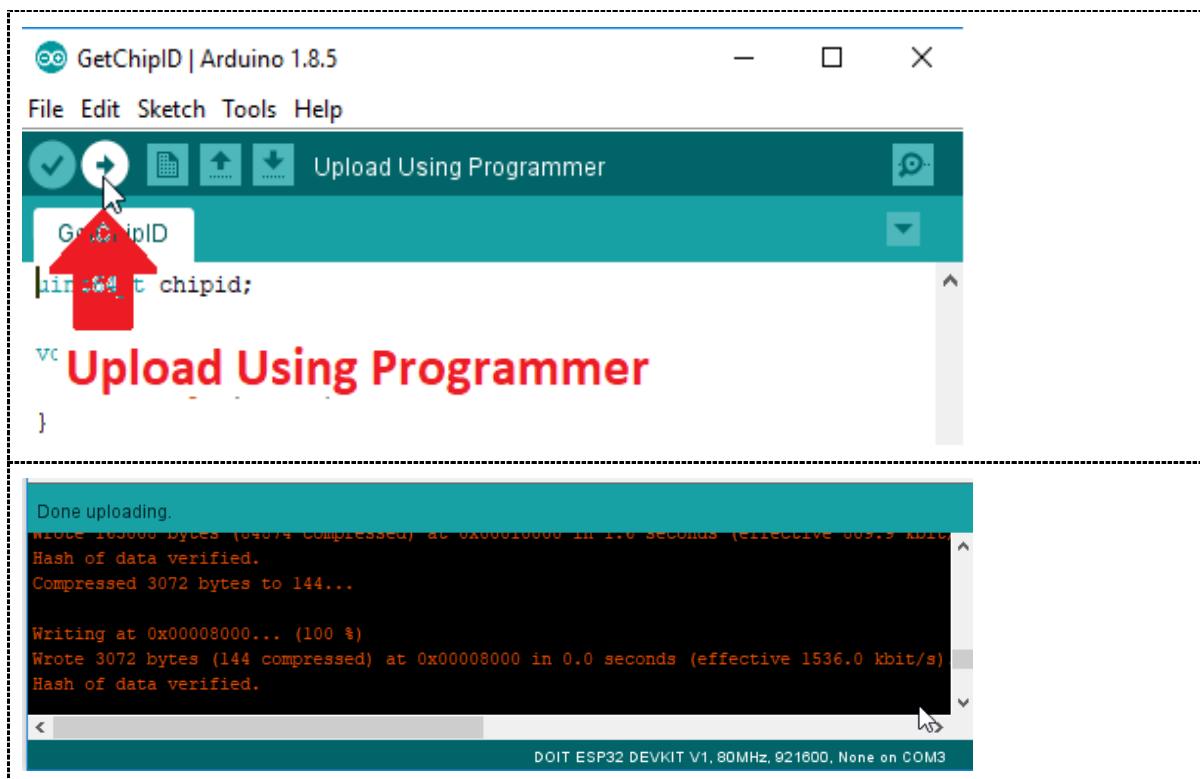
- ต่อบอร์ด ESP32 เข้ากับ PC ผ่าน USB Cable
- เลือกบอร์ดเป็น DOIT ESP32 DEVKIT1
- เลือก Communication port



- ทดสอบโปรแกรม File → Example → ESP32 → ChipID → "GetChipID"



- กด Upload โปรแกรม
- เมื่อโปรแกรมเสร็จเรียบร้อยจะขึ้นข้อความแสดงในกรอบลีเหลือง



- จากนั้นลองเปิด Serial Monitor ขึ้น ESP32 จะทำการปริ้น ChipID และแสดงขึ้นมา ดังรูป

GetChipID | Arduino 1.8.5

File Edit Sketch Tools Help

Serial Monitor

GetChipID

```
uint64_t chipid;

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

Serial Monitor

COM3

Send

ESP32 Chip ID = E8A205A4AE30
ESP32 Chip ID = E8A205A4AE30

115200

Autoscroll No line ending 115200 baud Clear output

- บอร์ดบางล็อตอาจขึ้น Connecting.....
- ให้กดปุ่ม BOOT ค้างไว้จนกว่าจะเริ่ม Upload

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

Archiving built core (caching) in: C:\Users\METTUM~1\AppData\Local\Temp\arduino cache 969974\core\core_espressif_esp32_esp32_FlashMode_dio,FlashFreq_80,FlashSize_4M,UploadSpeed_115200,DebugLevel. Sketch uses 283847 bytes (21%) of program storage space. Maximum is 1310720 bytes. Global variables use 27024 bytes (9%) of dynamic memory, leaving 267888 bytes for local variables. Maximum is 294912 bytes. esptool.py v2.1-beta1

Connecting.....

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

A fatal error occurred: Failed to connect to ESP32: Timed out waiting for packet header

การทดลองที่ 2.1 ทดสอบ LED Output

```
int led = 13;

// the setup routine runs once when you press reset:
void setup() {
  pinMode(led, OUTPUT);
}

// the loop routine runs over and over again forever:
void loop() {
  digitalWrite(led, HIGH);
  delay(1000);
  digitalWrite(led, LOW);
  delay(1000);
}
```

- Tools → Board → DOIT ESP32 DEVKIT V1
- Tools → Port → COM xx
- File → Example → Basic → Blink
- เลือกขวา เพื่อ Compile and Upload
- เว้นขยาย เพื่อต่อหรือสั่งงานผ่าน Serial Monitor

sketch_apr25a | Arduino 1.8.5

File Edit Sketch Tools Help

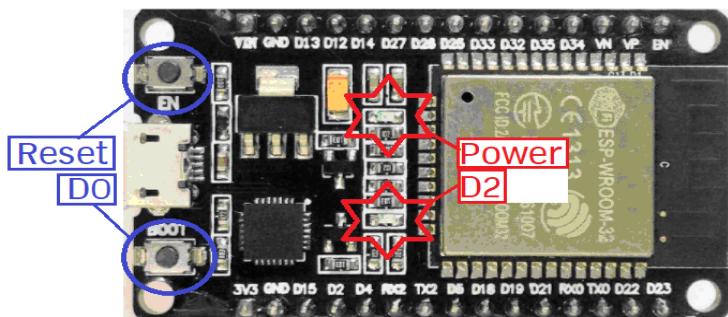
Upload Using Programmer

sketch_apr25a

void setup() {
 // put your setup code here, to run o
} **Compile and Upload**

Serial Monitor

void loop() {
 // put your main code here, to run re
}



การทดลองที่ 2.2 ทดสอบการรับส่งข้อมูลด้วย RS232

```

int ReadByte = 0; // for incoming serial data

void setup()
{
    Serial.begin(19200);
    Serial.flush();
    Serial.println("Demo : RS232 Receive & Transmit Data");
    Serial.println("Press Anykey for test...");
    Serial.print(">");
}

void loop()
{
    if (Serial.available() > 0)
    {
        ReadByte = Serial.read();
        Serial.print((char)ReadByte);
        if(ReadByte == 0x0D)
        {
            Serial.println();
            Serial.print(">");
        }
    }
}

```

การทดลองที่ 2.3 ทำความสะอาดขาเข้า/ขาออกแบบข้อมูลของ RS232

```

int ReadByte = 0; // for incoming serial data

void setup()
{
    Serial.begin(19200);
    Serial.flush();
    Serial.println("Demo : RS232 Command & Format Type");
    Serial.print("Press Anykey for test...");
}

void loop()
{
    if (Serial.available() > 0)
    {
        ReadByte = Serial.read();
        Serial.println((char)ReadByte);
        Serial.println();
        Serial.print("Display In DEC = ");
        Serial.println(ReadByte,DEC);
        Serial.print("Display In HEX = ");
        Serial.println(ReadByte,HEX);
        Serial.print("Display In OCT = ");
        Serial.println(ReadByte,OCT);
        Serial.print("Display In BIN = ");
        Serial.println(ReadByte,BIN);
        Serial.print("Display In BYTE = ");
        Serial.println((char)ReadByte);
        Serial.println();
        Serial.print("Press Anykey for test...");
    }
}

```

การทดลองที่ 2.4 ทดสอบการรับค่าจำนวนเต็มผ่าน RS232

```

int i,.ReadByte = 0; // for incoming serial data

void setup()
{ Serial.begin(19200);
  Serial.flush();
  Serial.println("Demo : RS232 Get Integer Data");
  Serial.print(">");
}

void loop()
{ if (Serial.available() > 0)
  { ReadByte = Serial.parseInt();
    Serial.print(ReadByte);
    Serial.print(" - ");
    for(i=0; i<.ReadByte; i++) Serial.print("X");
    Serial.println();
    Serial.print(">");
  }
}

```

- ทดลองป้อน 2
- ทดลองป้อน 10
- ทดลองป้อน 2.5

การทดลองที่ 2.5 ทดสอบการรับเลขคณิตผ่าน RS232

```

float ReadByte; // for incoming serial data

void setup()
{ Serial.begin(19200);
  Serial.flush();
  Serial.println("Demo : RS232 Get Float Data");
  Serial.print(">");
}

void loop()
{ if (Serial.available() > 0)
  { ReadByte = Serial.parseFloat();
    Serial.print(" Variable X = ");
    Serial.print(ReadByte,4);
    Serial.print(" > Power of X = ");
    Serial.println(ReadByte*ReadByte,4);
    Serial.print(">");
  }
}

```

- ทดลองป้อน 2
- ทดลองป้อน 2.5
- บรรทัด
Serial.print(ReadByte,4);
เลข 4 คืออะไร

3/6 -- การโปรแกรมเพื่อควบคุมสั่งงาน อินพุต/เอาต์พุต ของ ESP-32

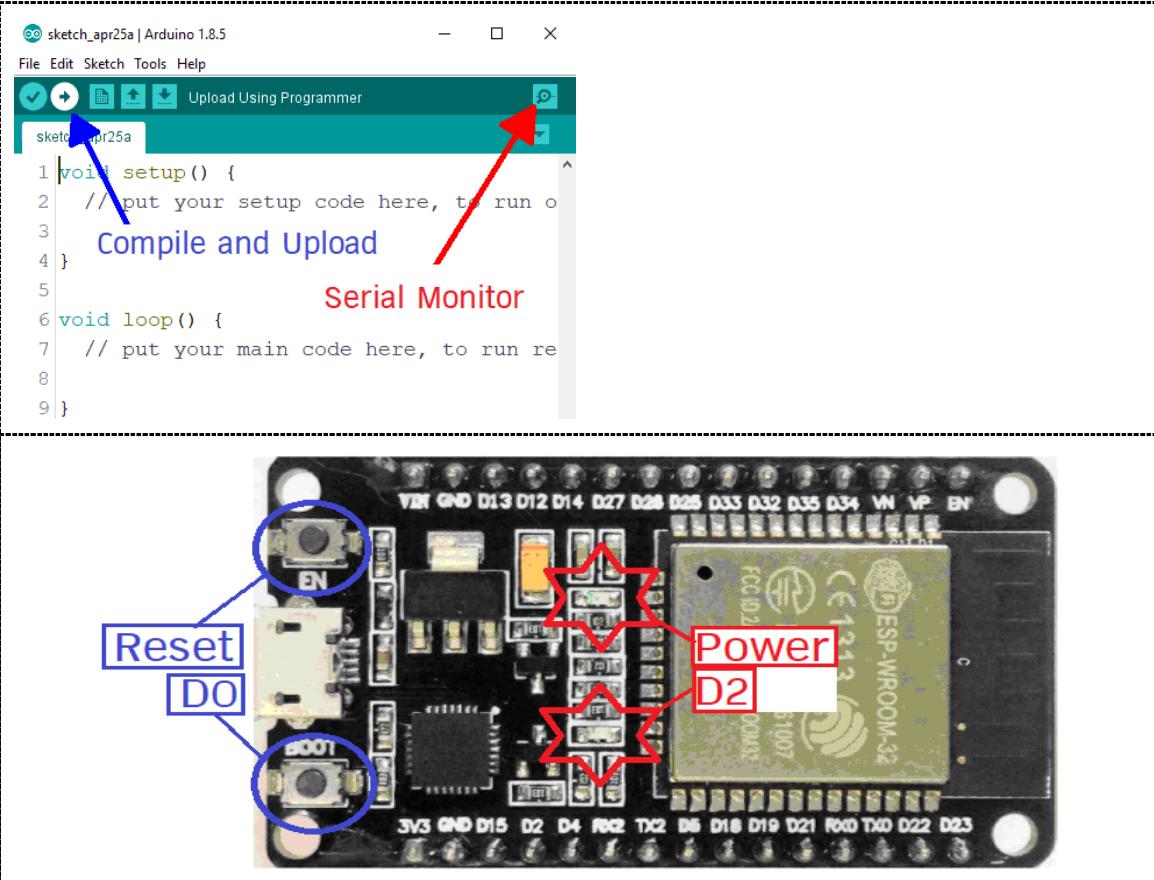
การทดลองที่ 3.1 – Blink

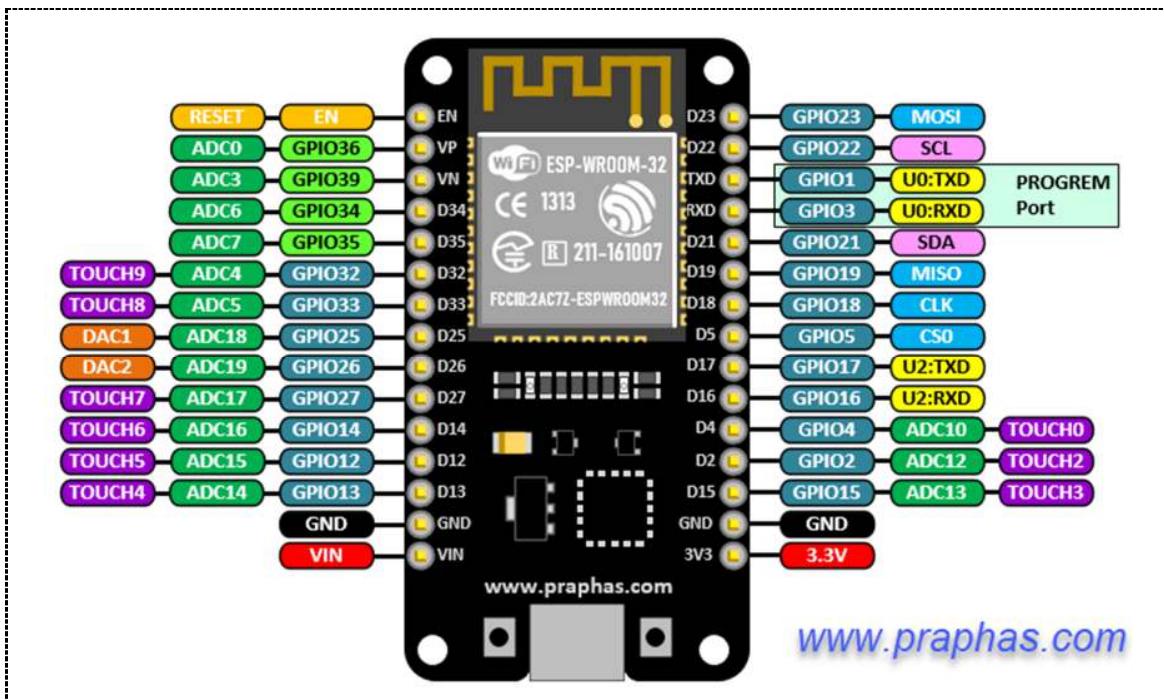
- Tools → Board → DOIT ESP32 DEVKIT V1
- Tools → Port → COM xx
- ทดสอบการทำงานโปรแกรมไฟกระพริบ File → Example → Basic → Blink

```
// Example_101 - Blink
void setup() {
    pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
    digitalWrite(LED_BUILTIN, HIGH); // turn the LED on (HIGH is the voltage level)
    delay(1000); // wait for a second
    digitalWrite(LED_BUILTIN, LOW); // turn the LED off by making the voltage LOW
    delay(1000); // wait for a second
}
```

- เลือกขวา เพื่อ Compile and Upload
- วนขยาย เพื่อดูหรือสั่งงานผ่าน Serial Monitor





การทดลองที่ 3.2 – Single LED Display

```

// Example_102a – Single LED fix pin
void setup() {
  pinMode(2, OUTPUT);
}

void loop() {
  digitalWrite(2, HIGH);      delay(1000);
  digitalWrite(2, LOW);       delay(1000);
}

// Example_102b – Single LED variable pin
int LED_test = 2;

void setup() {
  pinMode(LED_test, OUTPUT);
}

void loop() {
  digitalWrite(LED_test, HIGH);      delay(1000);
  digitalWrite(LED_test, LOW);       delay(1000);
}

// Example_102c – Single LED define pin
#define LED_test 2

void setup() {
  pinMode(LED_test, OUTPUT);
}

void loop() {
  digitalWrite(LED_test, HIGH);      delay(1000);
  digitalWrite(LED_test, LOW);       delay(1000);
}

// Example_102d – Single LED with array
int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0};

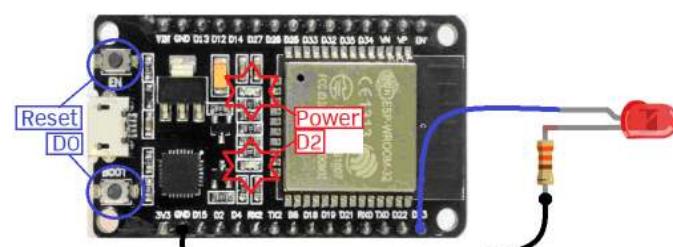
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  for (int i = 0; i < nloop; i++)
  { digitalWrite(LED_BUILTIN, DispBuff[i]);    delay(120);
    digitalWrite(LED_BUILTIN, LOW);           delay(120);
  }
}

```

คำแนะนำ

หากให้ Pin D23 ต่อ LED แล้ว
ปรับโปรแกรมทั้ง 102a, 102b, 102c,
102d เพื่อแสดงผลต้องทำอย่างไร

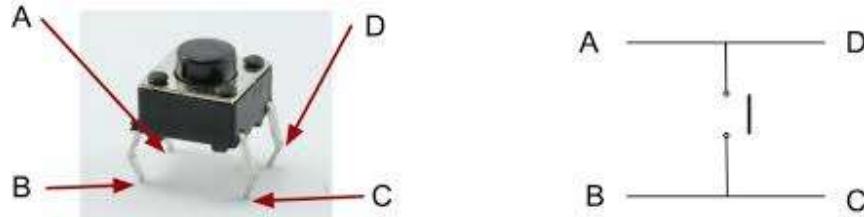
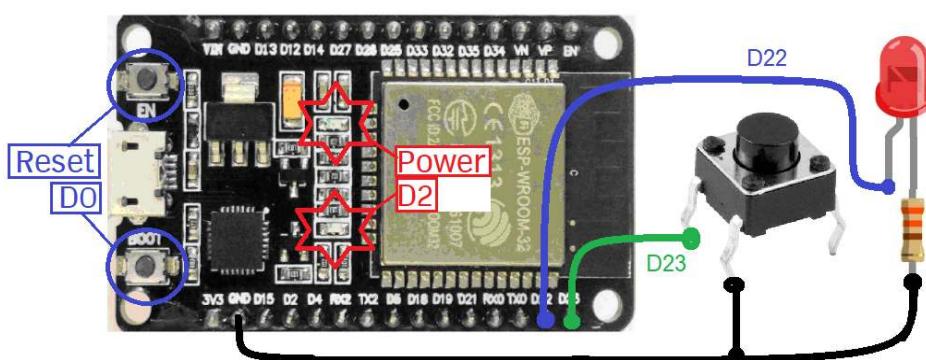
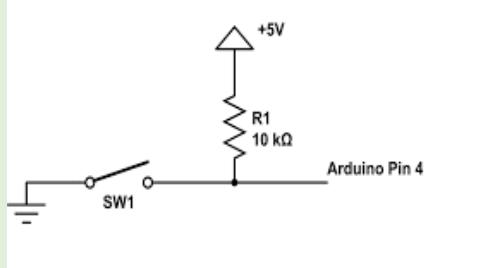


การทดลองที่ 3.3a – Input Switch

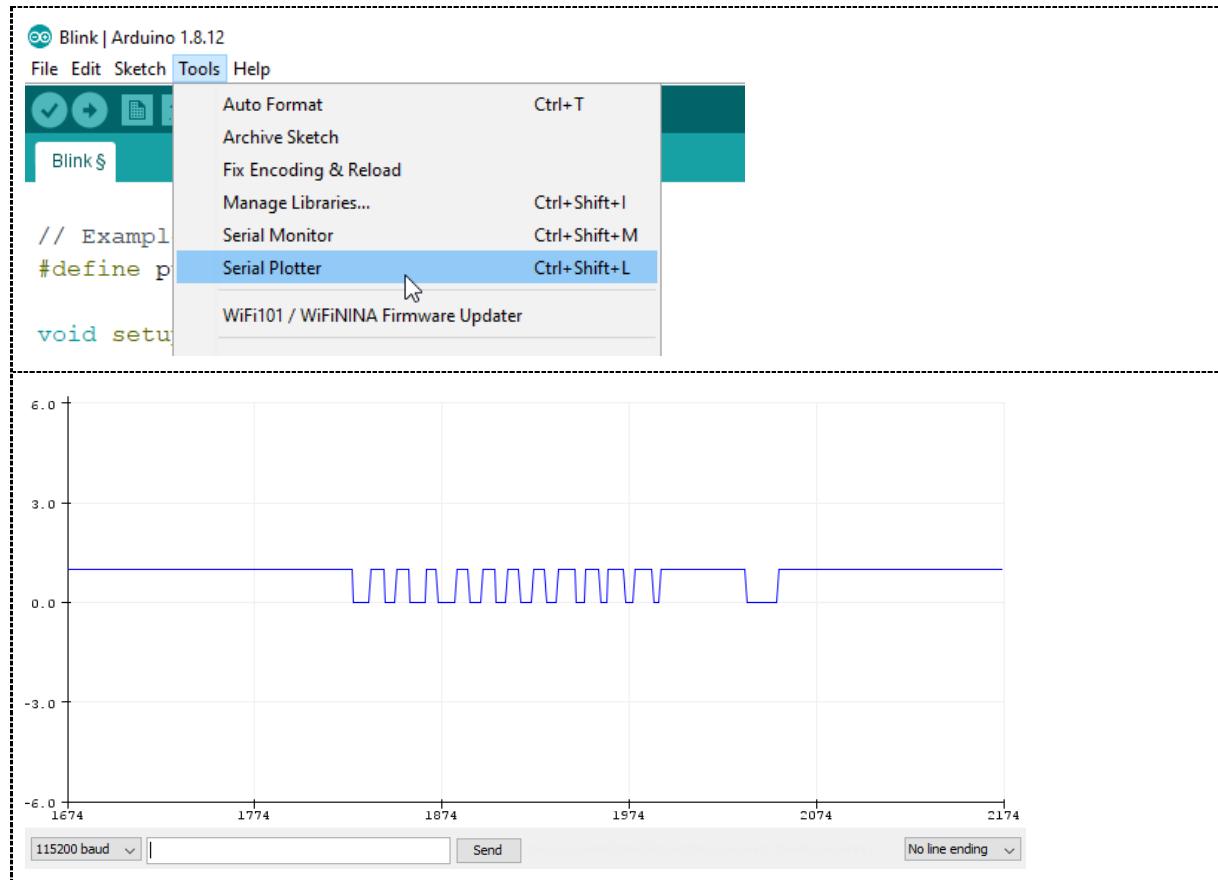
```
// Example_103a – Input Switch
#define pushButton 0

void setup() {
    Serial.begin(115200);
    pinMode(pushButton, INPUT_PULLUP);
}

void loop() {
    int swRead = digitalRead(pushButton);
    Serial.println(swRead);
    delay(10);
}
```



ดูการทำงานเมื่อกดสวิตซ์ที่ Tools → Serial Plotter



การทดลองที่ 3.3b – Input Switch กดติด ปล่อยดับ

```
// Example_103b – Input Switch
#define pushButton 23
#define LEDPin 22

void setup() {
  Serial.begin(115200);
  pinMode(pushButton, INPUT_PULLUP);
  pinMode(LEDPin, OUTPUT);
}

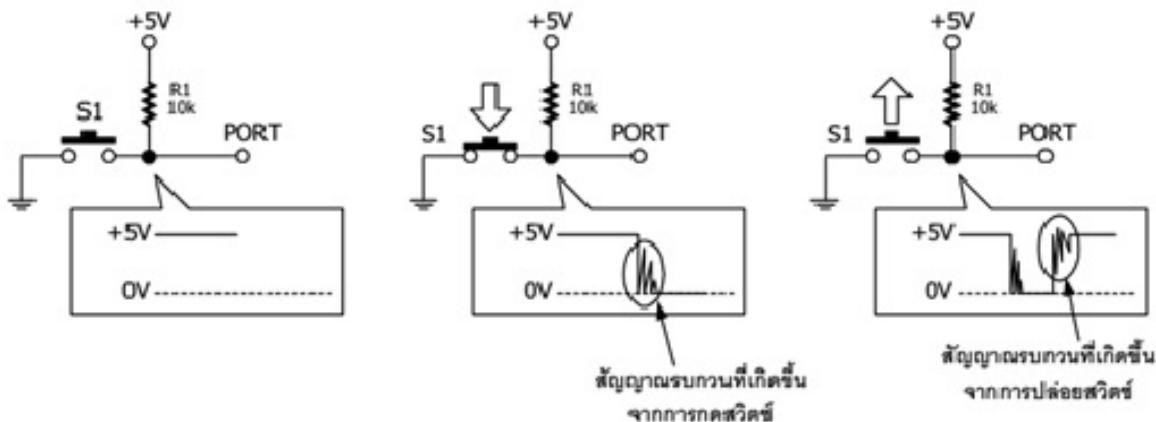
void loop() {
  int buttonState = digitalRead(pushButton);
  digitalWrite(LEDPin,buttonState);
  Serial.println(buttonState);
  delay(1);
}
```

การทดลองที่ 3.4 – Input Switch กดแล้วนับจำนวนครั้งการกด

```
// Example_104 – Counter Switch Press
#define pushButton 23
#define LEDPin 22
int Counter = 0;

void setup() {
  Serial.begin(115200);
  pinMode(pushButton, INPUT_PULLUP);
  pinMode(LEDPin, OUTPUT);
}

void loop() {
  if (digitalRead(pushButton) == LOW) {
    delay(20);
    Counter++;
    Serial.println(Counter);
    while (digitalRead(pushButton) == LOW);
    delay(20);
  }
}
```



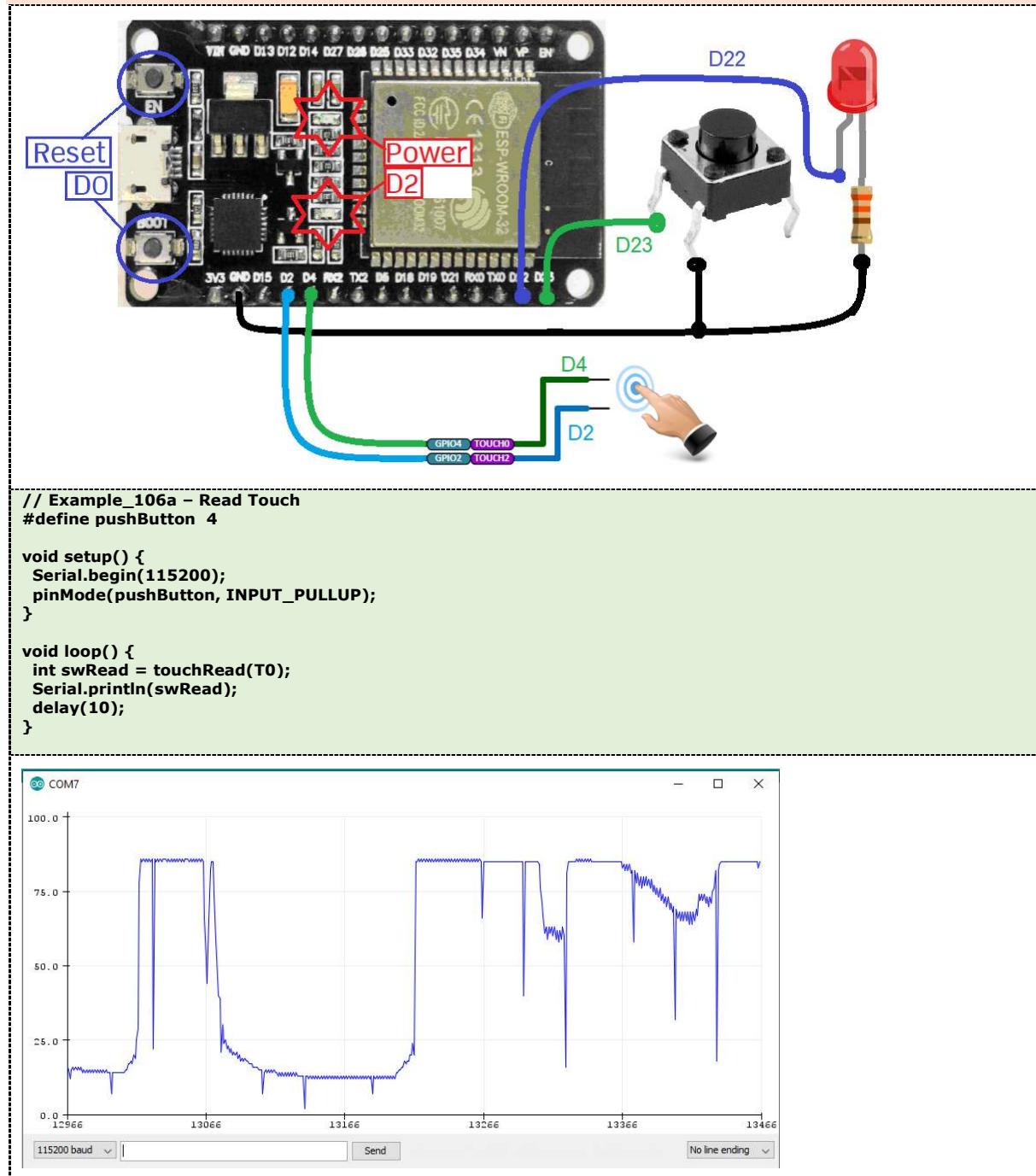
การทดลองที่ 3.5 – Input Switch กดติด กดดับ

```
// Example_105 – Push On/ Push Off
#define pushButton 23
#define LEDPin 22
int buttonState = 0;

void setup() {
  Serial.begin(115200);
  pinMode(pushButton, INPUT_PULLUP);
  pinMode(LEDPin, OUTPUT);
}

void loop() {
  if (digitalRead(pushButton) == LOW) {
    delay(20);
    buttonState = 1 - buttonState;
    digitalWrite(LEDPin, buttonState);
    while (digitalRead(pushButton) == LOW);
    delay(20);
  }
}
```

การทดลองที่ 3.6 – Input Touch Switch กดติด กดดับ



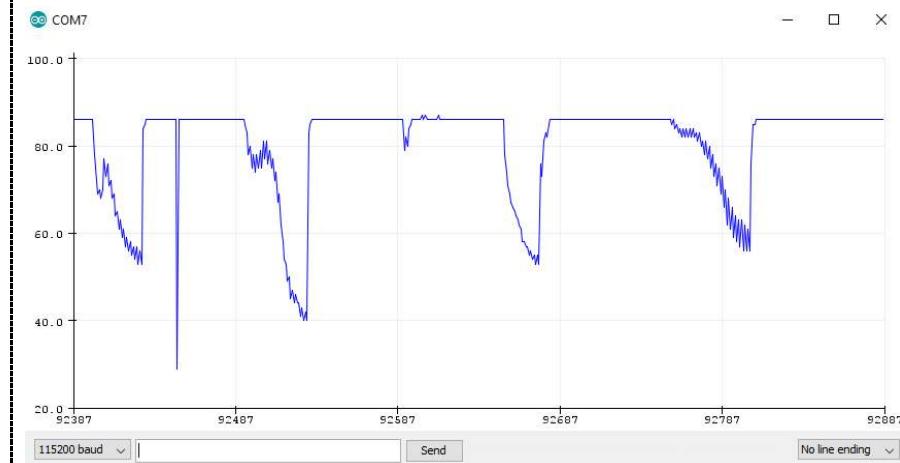
`touchRead(Touch pin);`

ฟังก์ชันอ่านค่าจากเซนเซอร์สัมผัส Touch pin: หมายเลขอาร์ทของเซอร์เซอร์สัมผัสที่ต้องการอ่านค่า เช่น `x=touchRead(TOUCH0);` หมายถึงอ่านค่าจากขา TOUCH0 เก็บไว้ในตัวแปร `x`

```
// Example_106B – Read Touch Control On/Off
#define LEDPin 22 // (or D2 BuiltIn LED)
#define pushButton 4
#define RefTouchSw 75
int Count, buttonState = 1;

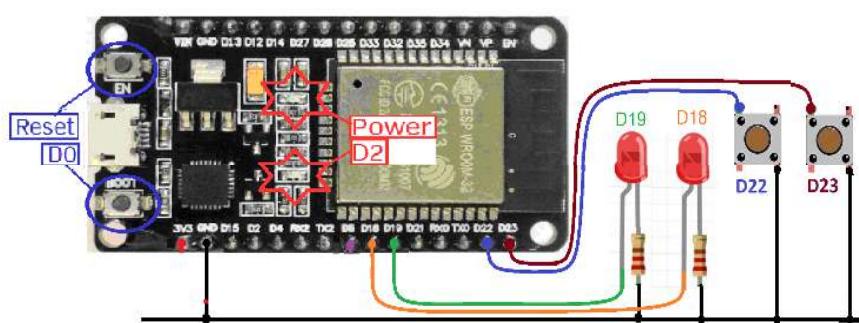
void setup() {
  Serial.begin(115200);
  pinMode(pushButton, INPUT_PULLUP);
  pinMode(LEDPin, OUTPUT);
}

void loop() {
  int touchReadT0 = touchRead(T0);
  Serial.println(touchReadT0);
  delay(10);
  if (touchReadT0 > RefTouchSw) Count = 15;
  else Count--;
  if (Count < 0) {
    buttonState = 1 - buttonState;
    digitalWrite(LEDPin, buttonState);
    while (touchRead(T0) < RefTouchSw);
  }
}
```



Quiz_101 – กดติด กดดับ 2 ชุด

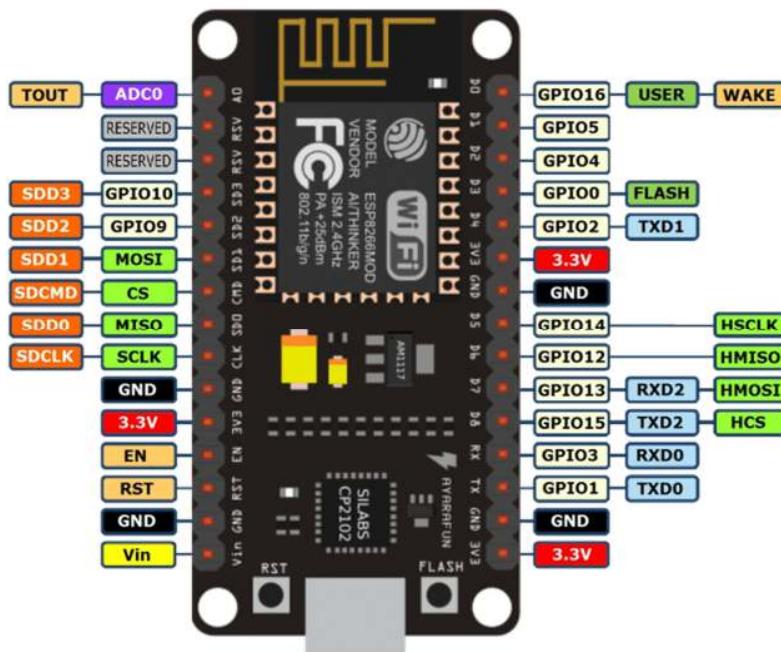
- หากต้องการใช้ 1 สวิตซ์ ควบคุม 1 LED แบบกดติด-กดดับ จำนวน 2 ดวงจะต้องใช้และเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}



4/6 -- การโปรแกรมเพื่อใช้งานแบบ All Over IP

4.1 ESP-8266, ESP-32 and M5-Stack

4.1.1 NodeMCU - ESP8266

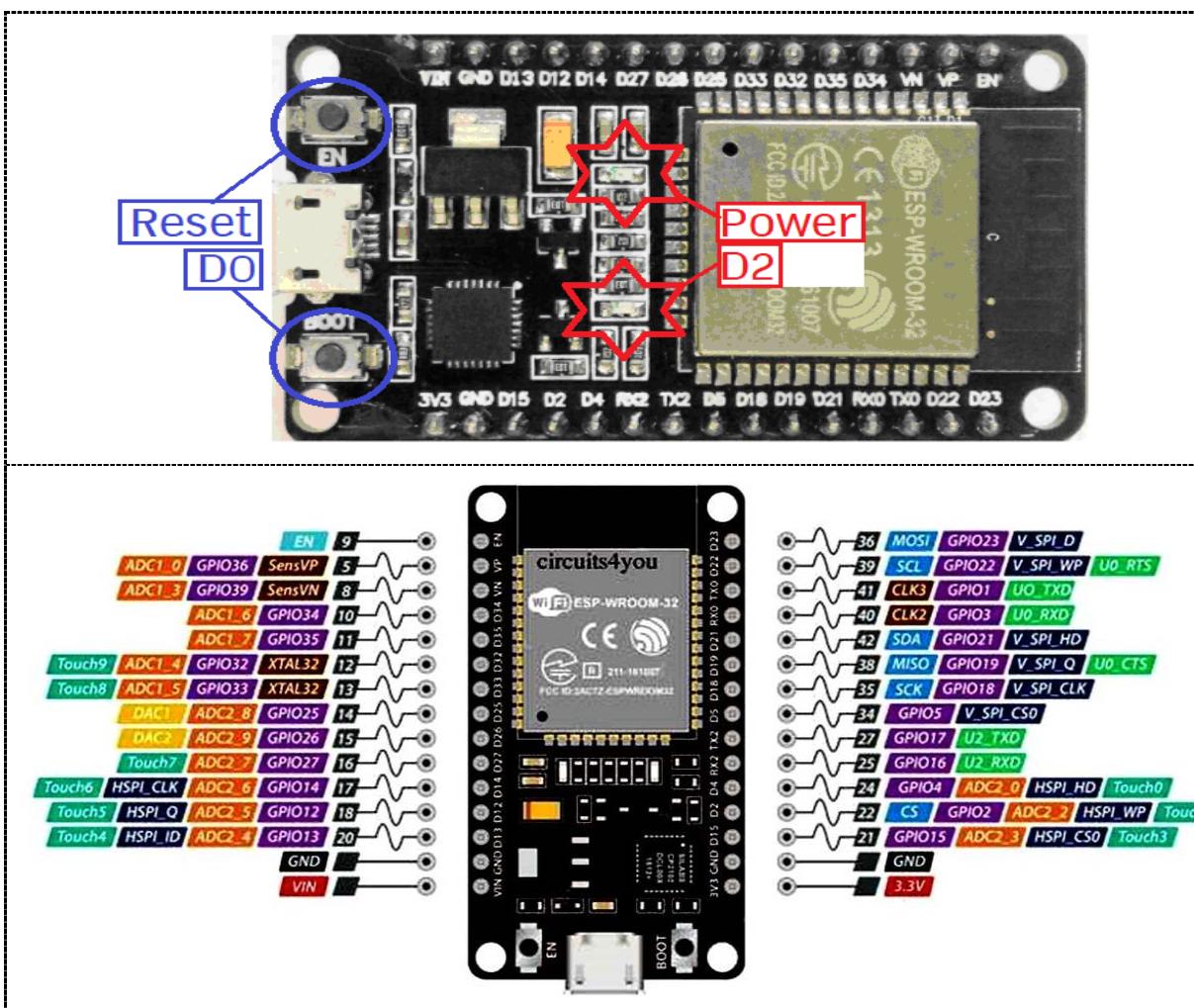


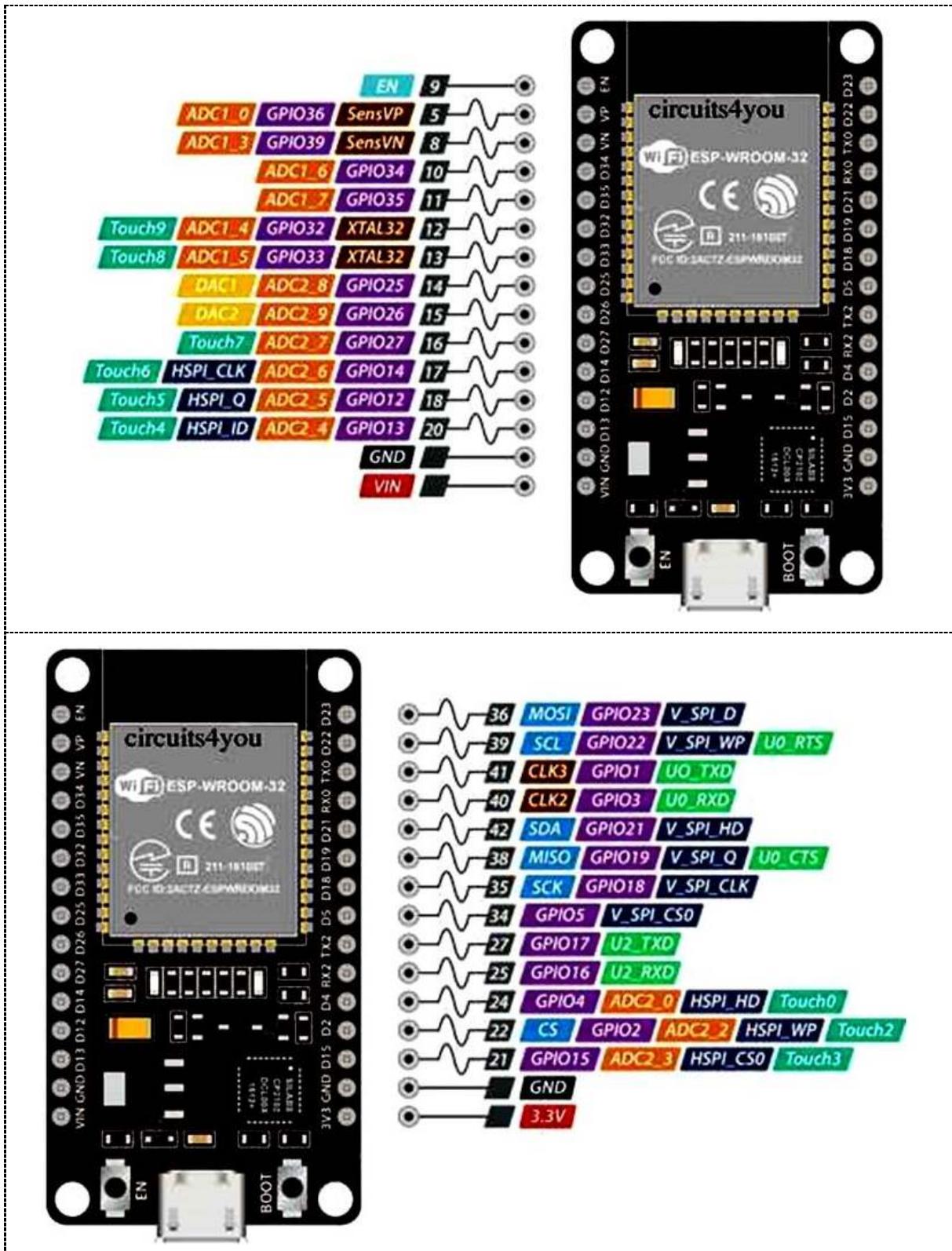
NodeMCU V0.9 ESP-12 Module	NodeMCU V1.0 ESP-12E Module	NodeMCU V3.0 ESP-12E Module
Arduino IDE = Node0.9	Arduino IDE = Node1.0	Arduino IDE = Node1.0
<ul style="list-style-type: none"> USB-SERIAL CH340 ใช้ Serial LED ที่ GPIO1 ได้แต่ต้องไม่ใช้พร้อม Serial Communication มี LED Built in ที่ GPIO16 "BUILTIN_LED" 	<ul style="list-style-type: none"> Silicon Labs CP210x USB to UART Bridge ใช้ Serial LED ที่ GPIO2 ได้แต่ต้องไม่ใช้พร้อม Serial Communication มี LED Built in ที่ GPIO16 "BUILTIN_LED" 	<ul style="list-style-type: none"> USB-SERIAL CH340 ใช้ Serial LED ที่ GPIO2 ได้แต่ต้องไม่ใช้พร้อม Serial Communication

4.1.2 ESP-32 Dev Kit V1 Board

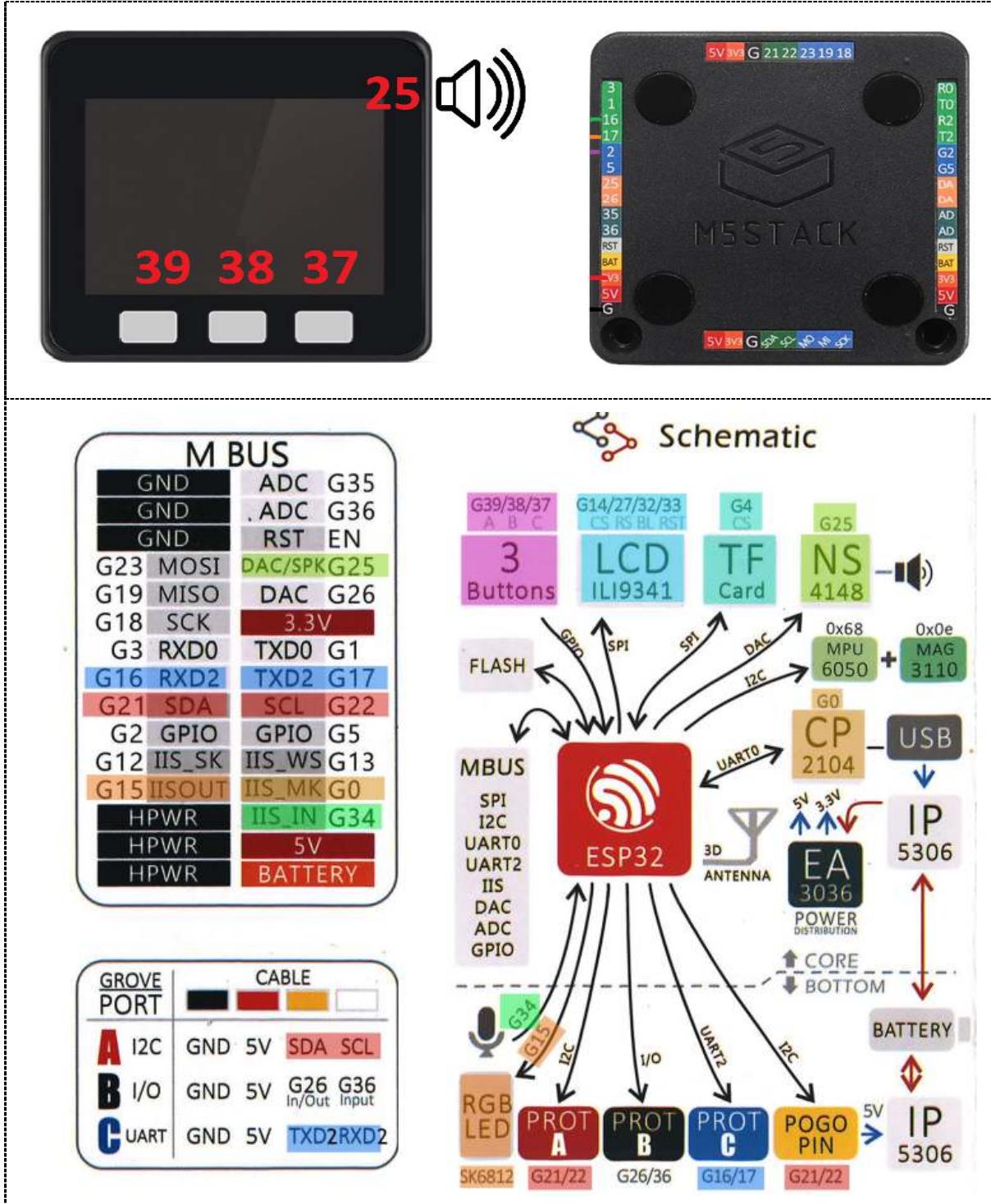
ก่อนหน้านี้มีการใช้งาน NodeMCU V2 ซึ่งเป็น ESP8266 อย่างแพร่หลาย แต่ด้วยเลี่ยงลือเสียงเล่าข่าวเรื่องความสามารถของ ESP32 ที่พัฒนาความสามารถเพิ่มมาแก้จุดด้อยของ ESP8266 ทั้งรองรับการเชื่อมต่อแบบ Hybrid ทั้ง WiFi และ Bluetooth มีพอร์ตรองรับ I/O ได้เพิ่มขึ้น รองรับ touch sensor มี hardware เข้ารหัสสำหรับ HTTPS และอีกมากมาย ด้วยเหตุผลที่ว่าไปแล้วและราคาที่ไม่แพง รอบนี้เลยได้ ESP32 Development Board ที่มีชื่อเต็มคือ DOIT ESP32 DevKit V1 ใช้โมดูล ESP-WROOM-32 มาทำการทดสอบ

รายละเอียดเพิ่มเติมของ DOIT ESP32 DevKit V1 ลองเข้าไปดูใน SmartArduino (<https://github.com/SmartArduino/SZDOITWiKi/wiki/ESP8266---ESP32>) หน้าตากล้าย ESP32 DevKit C V2 ของ Espressif และ Development Board ตระกูลเดียวกันกับเจ้าอื่นๆเลย มีเจาะรูสีมุ่งมาด้วย แต่ pinout ไม่เหมือนกัน





4.1.3 M5-Stack



FIRE
DEVELOPMENT KIT
V1.0

ESP32 | Wi-Fi | BLE | FLASH | 9DOF | 3D-Antenna
2 inch LCD @320X240 | TYPE-C USB | TF-Reader
1W Speaker | MEMS MIC | RGB LED x 10 | 3 Buttons
550mAh Battery | GROVE I2C | GROVE I/O | GROVE UART
MicroPython | LEGO Compatible | Arduino Compatible

Operation

- POWER ON: single press when using battery, and RESET.
- POWER OFF: double press

Type C: Power Supply / Charge / UART/Upload

GROVE PH2.0-4: SCL SDA 5V GND

MicroSD Card (TF-Card): 16GB

CORE Module: S454 X12.6mm

MSGO Bottom Magnet Paste: 34x540mm

Battery Inside

M5STACK

Internal View

M BUS

GND	ADC	G35
GND	ADC	G36
GND	RST	EN
G23	MOSI	DAC/SPKG25
G19	MISO	DAC G26
G18	SCK	3.3V
G3	RXD0	TXD0 G1
G16	RXD2	TXD2 G17
G21	SDA	SCL G22
G2	GPIO	GPIO G5
G12	IIS_SK	IIS_WS G13
G15	IISOUT	IIS_MK G0
HPWR	IIS_IN	G34
HPWR	5V	
HPWR	BATTERY	

GROVE PORT

A	I2C	GND	5V	SDA	SCL
B	I/O	GND	5V	G26	G36
C	UART	GND	5V	TXD	RXD

PORT A

PORT B

PORT C

RESET/POWER BUTTON

TYPE-C USB

9DOF SENSOR

TF Card Reader

IP5306

BATTERY SOCKET

SPEAKER

2X15 PIN BUS

ESP32 + FLASH

3D-ANTENNA

CP2104

NS4148

EA3036

2' Color LCD

BUTTON x 3

RGB BAR

PORT A

PORT B

PORT C

RESET/POWER BUTTON

TYPE-C USB

550mAh Battery Inside

POGO PIN

M3 HOLE

MAGNET X4

LEGO HOLE

1W SPEAKER

RGB BAR

CHG. LED

TF-CARD MIC

QR Code

www.M5stack.com
<https://github.com/m5stack>

5Stack ESP32 คอมพิวเตอร์จิ๋วที่ใช้สร้างงานต้นแบบที่มาพร้อมกับหน้าจอ 2 นิ้ว ความละเอียด 320x240 pixel และมีหน้ากากให้เปลี่ยนเป็นพิมพ์ได้ 3 แบบ ตามการใช้งานที่ออกแบบ Keyboard panel, Gameboy panel และ Number Panel พร้อมกับฐานชาติและแบต LiPo ขนาด 650mAh

บอร์ดด้านหลังจะเป็น M5 Faces ซึ่งทำหน้าที่ต่อเข้ากับแบนพิมพ์ Panel ต่างๆ และยังมีแบตเตอรี่ LiPO ขนาด 650mAh ที่สามารถชาร์จผ่าน charging Base ได้

4.1.4 อ่านเพิ่มเติม

- NodeMCU GitHub: <https://github.com/nodemcu>
- NodeMCU Driver: <https://www.silabs.com/products/mcu/Pages/USBtoUARTBridgeVCPDrivers.aspx>
- ThaiEasyElect: <http://www.thaieasyelec.com/products/internet-of-things/nodemcu-development-kit-v2-detail.html?gclid=Cj0KEQjwl-e4BRCwqeWkv8TWqOoBEiQAMocbPytm40atWOSYlaQI7V0O0p-7asSWryeJ9tCQJNxnpoaAk2-8P8HAQ>
- AiyaraFun: <http://www/ayarafun.com/2015/08/introduction-arduino-esp8266-nodemcu/>
- Firmware Build and Example: <http://nodemcu-build.com/>
- Read This <https://playelek.com/doit-esp32-devkit-v1/>
- Read This <http://esp32.net/>
- Read This <https://www.arduintronics.com/product/1329/doit-esp32-development-board-esp-wroom-32-wifibluetooth-esp-32s>
- Read This <https://www.mcucity.com/product/1144/doit-esp32-wifibluetooth-ultra-low-power-consumption-dual-core-esp-32-esp-32s-esp-32-similar-esp8266>
- M5Stack - <https://github.com/m5stack/M5Stack>
- M5Stack - <http://forum.m5stack.com/topic/360/m5stack-fire-pinout-leaflet>

การทดลองที่ 4.1 –Start with Arduino IDE in Hello World

1. Install Arduino IDE and Add ESP32 Board
2. Test EX00_Blink

```
// ESP-32

int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0};

void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}

void loop() {
  for (int i = 0; i < nloop; i++) {
    digitalWrite(LED_BUILTIN, DispBuff[i]);
    delay(120);
    digitalWrite(LED_BUILTIN, LOW);
    delay(120);
  }
}
```

```
#include <M5Stack.h>
int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0};

void setup() {
  M5.begin();
  M5.Lcd.printf("M5Stack Speaker");
}

void loop() {
  for (int i = 0; i < nloop; i++) {
    if (DispBuff[i] == 1)
      M5.Speaker.beep(); //beep;
    else
      M5.Speaker.mute(); //no beep;
    delay(100);
    M5.Speaker.mute(); //no beep;
    delay(100);
  }
}
```

การทดลองที่ 4.2 – My MAC Address

3. โหลดโปรแกรม My MAC = _____

```
uint64_t chipid;

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

void loop() {
  chipid = ESP.getEfuseMac(); //The chip ID is essentially its MAC address(length: 6 bytes).
  Serial.printf("ESP32 Chip ID = ");
  Serial.printf("%02X:", (uint8_t)(chipid >> 0)); //print 1 bytes
  Serial.printf("%02X:", (uint8_t)(chipid >> 8));- //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 16)); //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 24)); //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 32)); //print 1 bytes.
  Serial.printf("%02X:", (uint8_t)(chipid >> 40)); //print 1 bytes.
  Serial.println();
  delay(3000);
}
```

4. ทดสอบ File → Example → WiFi → WiFiScan หรือโปรแกรมต่อไปนี้

```
#include "WiFi.h"

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

  // Set WiFi to station mode and disconnect from an AP if it was previously connected
  WiFi.mode(WIFI_STA);
  WiFi.disconnect();
  delay(100);
  Serial.println("Setup done");
}

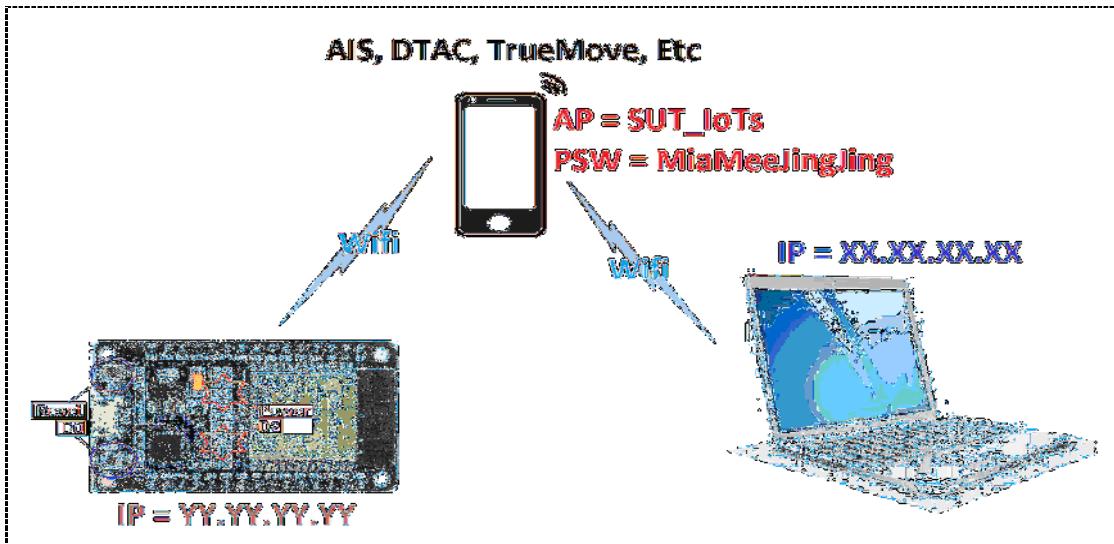
void loop()
{
  Serial.println("scan start");

  // WiFi.scanNetworks will return the number of networks found
  int n = WiFi.scanNetworks();
  Serial.println("scan done");
  if (n == 0) {
    Serial.println("no networks found");
  } else {
    Serial.print(n);
    Serial.println(" networks found");
    for (int i = 0; i < n; ++i) {
      // Print SSID and RSSI for each network found
      Serial.print(i + 1);
      Serial.print(": ");
      Serial.print(WiFi.SSID(i));
      Serial.print(" (");
      Serial.print(WiFi.RSSI(i));
      Serial.print(")");
      Serial.println((WiFi.encryptionType(i) == WIFI_AUTH_OPEN)? ":" "*");
      delay(10);
    }
  }
  Serial.println("");

  // Wait a bit before scanning again
  delay(5000);
}
```

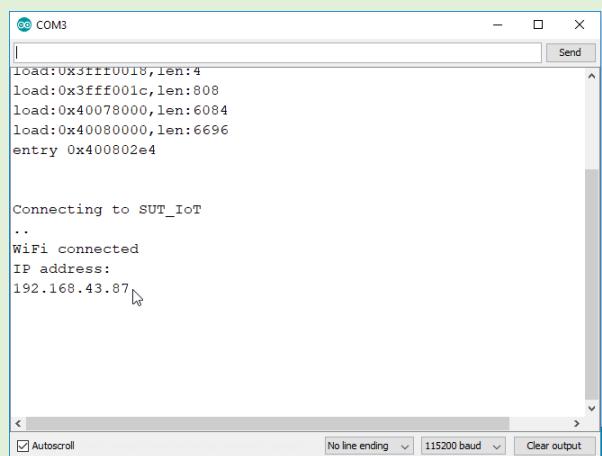
การทดลองที่ 4.3 –Connect to Network

5. โหลดโปรแกรมและ อย่าลืมแก้เป็นชื่อ SSID และ Password เป็นของตัวเอง



6. ทดสอบการเชื่อมต่อด้วยคำสั่ง Ping ในหน้าต่าง command

```
#include <WiFi.h>
const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
void setup() {
  Serial.begin(115200);
  delay(10);
  Serial.println(); Serial.printin();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  { delay(500);
    Serial.print(".");
  }
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}
void loop() {}
```



```
C:\ Select Command Prompt
Microsoft Windows [Version 10.0.16299.547]
(c) 2017 Microsoft Corporation. All rights reserved.

C:\Users\Pk007_Bit32>ping 192.168.43.87

Pinging 192.168.43.87 with 32 bytes of data:
Reply from 192.168.43.87: bytes=32 time=780ms TTL=255
Reply from 192.168.43.87: bytes=32 time=377ms TTL=255
Reply from 192.168.43.87: bytes=32 time=386ms TTL=255
Reply from 192.168.43.87: bytes=32 time=395ms TTL=255

Ping statistics for 192.168.43.87:
  Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
  Approximate round trip times in milli-seconds:
    Minimum = 377ms, Maximum = 780ms, Average = 484ms

C:\Users\Pk007_Bit32>
```

การทดลองที่ 4.4 –Web Server-Command

7. Test Ex20_WebServer_Cmd

```
#include <WiFi.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
int pinTest = 2;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(pinTest, OUTPUT);           // set the LED pin mode
  delay(10);
  Serial.print("\n\nConnecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected.");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());

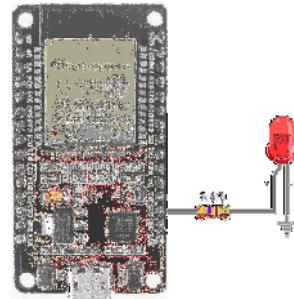
  server.begin();
}

int value = 0;
bool LED_Status = LOW;

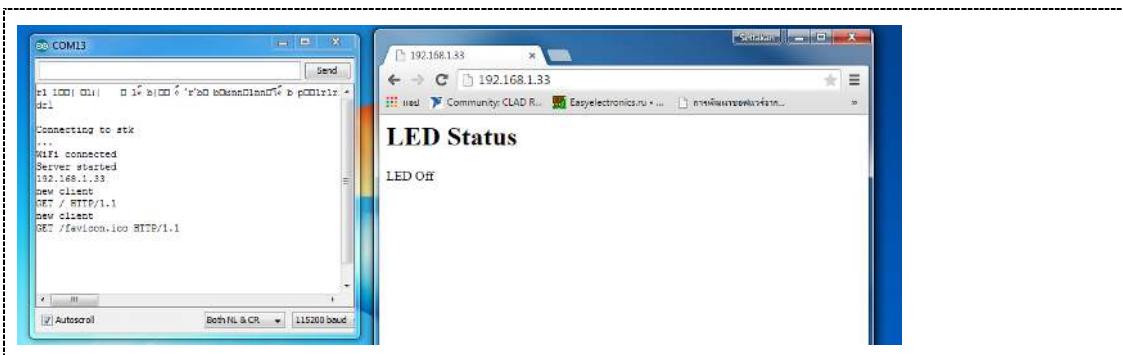
void loop() {
  digitalWrite(pinTest, LED_Status);
  WiFiClient client = server.available(); // listen for incoming clients

  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = "";
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        if (c == '\n') { // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");
            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED On");
            else
              client.println("LED Off");
            client.println("</p>");
            client.println("</body>");
            client.println("</html>");
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
        if (currentLine.endsWith("GET /ledon")) LED_Status = HIGH;
        if (currentLine.endsWith("GET /ledoff")) LED_Status = LOW;
      }
    }
    client.stop(); // close the connection
    Serial.println("Client Disconnected.");
  }
}
```

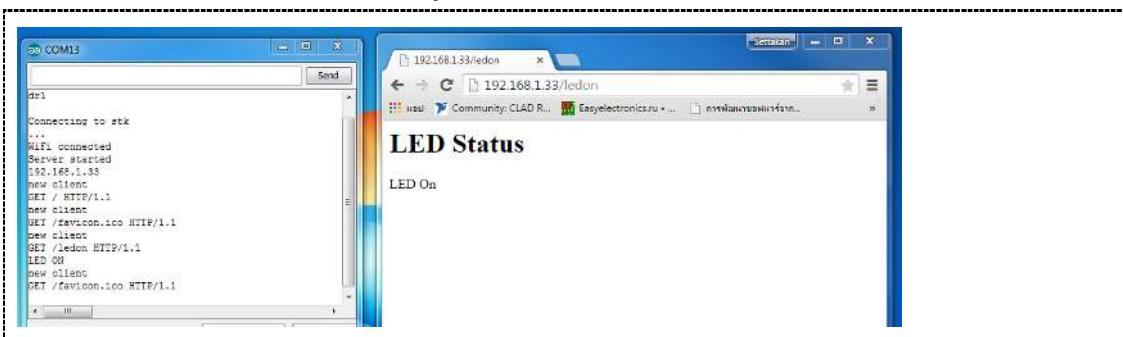
การทดลองนี้เป็นการนำเอา ESP-32 มาสร้างเป็น Web Server โดยเมื่อมีการร้องขอหน้าเว็บใช้ตัวมาร์กให้หลอด LED ติดดับ ต่อวงจร โหลดโปรแกรมและอย่าลืมแก้เป็นชื่อ SSID และ Password เป็นของตัวเอง



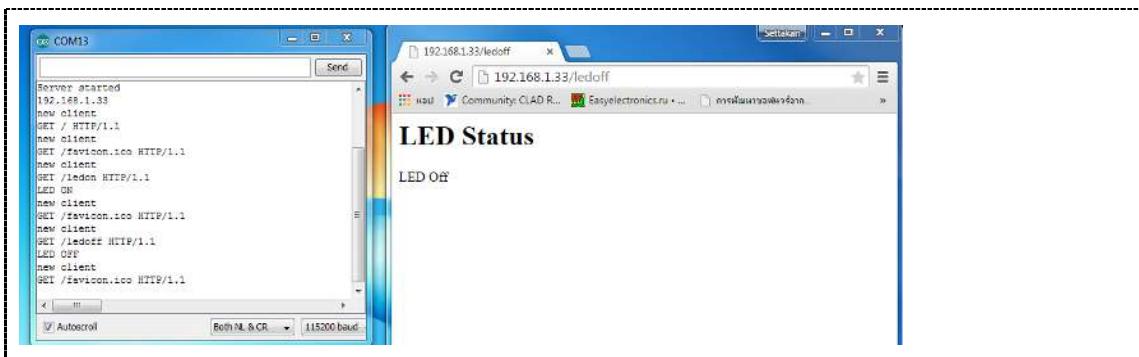
เปิด Web Browser และกำหนด url ไปที่ IP ของ ESP-32 → YY.YY.YY.YY



เรียกหน้าเว็บไปที่ YY.YY.YY.YY/ledon สังเกต >> หลอด LED จะติด



เรียกหน้าเว็บไปที่ YY.YY.YY.YY/ledoff สังเกต >> หลอด LED จะดับ



การทดลองที่ 4.5 –Web Server-Button

8. Test Ex21_WebServer_Button

```
#include <WiFi.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
int pinTest = 2;

WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(pinTest, OUTPUT); // set the LED pin mode
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}

int value = 0;
bool LED_Status = LOW;

void loop() {
  digitalWrite(pinTest, LED_Status);
  WiFiClient client = server.available(); // listen for incoming clients

  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        if (c == '\n') { // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");

            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED On");
            else
              client.println("LED Off");

            client.println("<p>");
            client.println("<a href=\"/ledon\"><button>LED On</button></a>");
            // client.println("<a href=\"/ledon\"><button style = \"background-color: #f44336;\">LED On</button></a>");
            client.println("</p>");
            client.println("<a href=\"/ledoff\"><button>LED Off</button></a>");
            // client.println("<a href=\"/ledoff\"><button style = \"background-color: #008CBA;\">LED Off</button></a>");

            client.println("<body>");
            client.println("<html>");
            break;
          } else {
            currentLine = "";
          }
        } else if (c != '\r') {
          currentLine += c;
        }
        if (currentLine.endsWith("GET /ledon")) LED_Status = HIGH;
        if (currentLine.endsWith("GET /ledoff")) LED_Status = LOW;
      }
    }
    client.stop(); // close the connection
    Serial.println("Client Disconnected.");
  }
}
```

- ควบคุมการทำงานของ LED D1 โดยต่อวงจรตามรูป

- ลองปรับแก้จากพอร์ต 80 เป็นพอร์ตอื่น เช่น 3456
- การเรียกดูผ่าน Web Browser ก็ต้องระบุพอร์ตด้วย เช่น **192.168.0.3:3456**

```
บรรทัดที่ 6
จาก WiFiServer server(80); // กำหนดใช้งาน TCP Server ที่ Port 80
เป็น WiFiServer server(3456);
```

การทดลองที่ 4.6 – Web Server-Digital Read

9. Test Ex30_WebServer_DigitalRead

```
#include <WiFi.h>
#define pinTest 2
#define SW_Test 39

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  pinMode(SW_Test, INPUT_PULLUP);
  pinMode(pinTest, OUTPUT);
  delay(10);
  Serial.print("\n\nConnecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);

  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }

  Serial.println("");
  Serial.println("WiFi connected."); Serial.println("IP address: ");
  Serial.println(WiFi.localIP()); server.begin();
}

int value = 0;
bool LED_Status = LOW;

void loop() {
  digitalWrite(pinTest, LED_Status);
  WiFiClient client = server.available(); // listen for incoming clients

  if (client) { // if you get a client,
    Serial.println("New Client."); // print a message out the serial port
    String currentLine = ""; // make a String to hold incoming data from the client
    while (client.connected()) { // loop while the client's connected
      if (client.available()) { // if there's bytes to read from the client,
        char c = client.read(); // read a byte, then
        Serial.write(c); // print it out the serial monitor
        if (c == '\n') { // if the byte is a newline character
          if (currentLine.length() == 0) {
            client.println("HTTP/1.1 200 OK");
            client.println("Content-type:text/html");
            client.println();
            client.println("<html>");
            client.println("<body>");
            client.println("<h1>LED Status</h1>");

            client.println("<p>");
            if (LED_Status == HIGH)
              client.println("LED On");
            else
              client.println("LED Off");

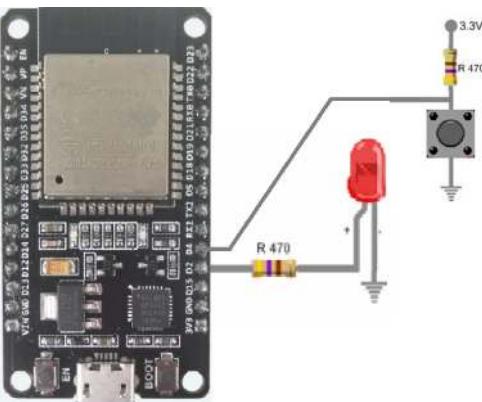
            client.println("</p>");
            client.println("<a href=\"/ledon\"><button>LED On</button></a>");
            client.println("<a href=\"/ledoff\"><button>LED Off</button></a>");
            client.println("</p>");

            client.println("<h1>Read Switch</h1>");
            client.println("<style>");
            client.println(".circle-green,.circle-red");
            client.println("{width: 100px; height: 100px; border-radius: 50%;}");
            client.println(".circle-green {background-color: green}");
            client.println(".circle-red {background-color: red}");
            client.println("</style>");

            client.println("<meta http-equiv=\"refresh\" content=\"1\">");
            client.println("<p>");
            if (digitalRead(SW_Test) == HIGH)
              client.println("<div class=\"circle-red\"></div>");
            client.println("<p>SW = 1</p>");
          }
        }
      }
    }
  }
}
```

```
    currentLine = "";
}
} else if (c != '\r') {
    currentLine += c;
}
if (currentLine.endsWith("GET /ledon")) LED_Status = HIGH;
if (currentLine.endsWith("GET /ledoff")) LED_Status = LOW;
}
client.stop(); // close the connection:
Serial.println("Client Disconnected.");
}
```

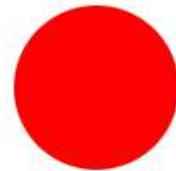
- วงศ์ตามรูป



LED Status

LED On

Read Switch



SW = 1

- เปิด Web Browser และเรียกหน้า Page ไปยัง IP ของ Node MCU

• เมื่อไม่กด Switch

• ทดลองกด Switch

Read Switch



SW = 1

Read Switch



SW = 0

การทดลองที่ 4.7 – Web Server-Sensor

10. Test Ex40_DHT22 Sensor ทดสอบโปรแกรมนี้

- Add Library ด้วยคำสั่ง Sketch → Include Library → Manage
- ค้นหา DHT22 เลือก DHT-22 library ของ [beegee_tokyo Ver 1.17.0](#)

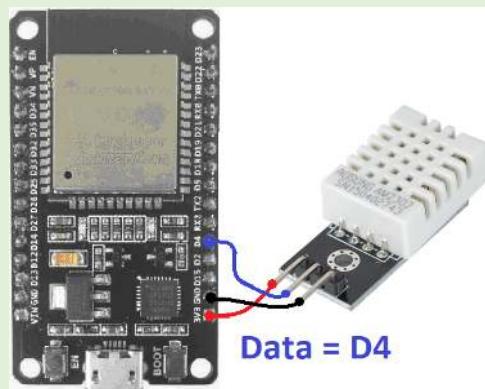
DHT sensor library for ESPx
by beegee_tokyo Version 1.17.0 INSTALLED
Arduino ESP library for DHT11, DHT22, etc Temp & Humidity Sensors Optimized lib
changes: Reduce CPU usage and add decimal part for DHT11
[More info](#)

● DHT-22 Test Code

```
#include "DHTesp.h"
DHTesp dht;

void setup()
{
  Serial.begin(115200);
  Serial.println();
  Serial.println("Status\tTemperature
(C)\tHumidity (%)");
  dht.setup(4, DHTesp::DHT22); // DHT_Pin D4,
  DHT22
}

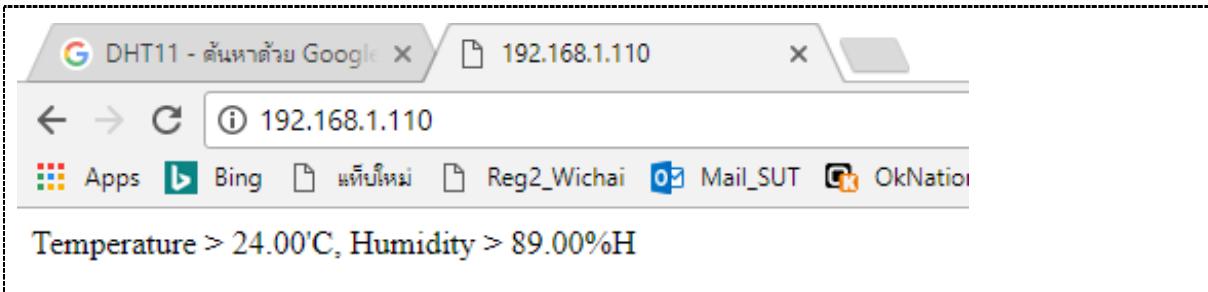
void loop()
{
  delay(2000);
  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();
  Serial.println();
  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(temperature, 1);
  Serial.print("\t\t");
  Serial.print(humidity, 1);
}
```



Status	Temperature (C)	Humidity (%)
OK	27.6	46.2
OK	27.6	46.3
OK	27.6	46.3
OK	27.5	45.9
OK	27.5	45.4
OK	27.5	45.1
OK	27.5	45.2
OK	27.5	45.4

11. Test Ex41_WebServer_DHT22 Sensor

- ผลของการทดสอบการทำงาน



12. ทดสอบ WebServer Test Code

```
#include <WiFi.h>
#include "DHTesp.h"
#define DHT_Pin 4
const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

DHTesp dht;
WiFiServer server(80);

void setup() {
  Serial.begin(115200);
  Serial.print("\n\nConnecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED)
  { delay(500); Serial.print(".");
  }
  Serial.println();
  Serial.println("WiFi connected");
  server.begin();
  Serial.println("Server started");
  Serial.println(WiFi.localIP());
  dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
}

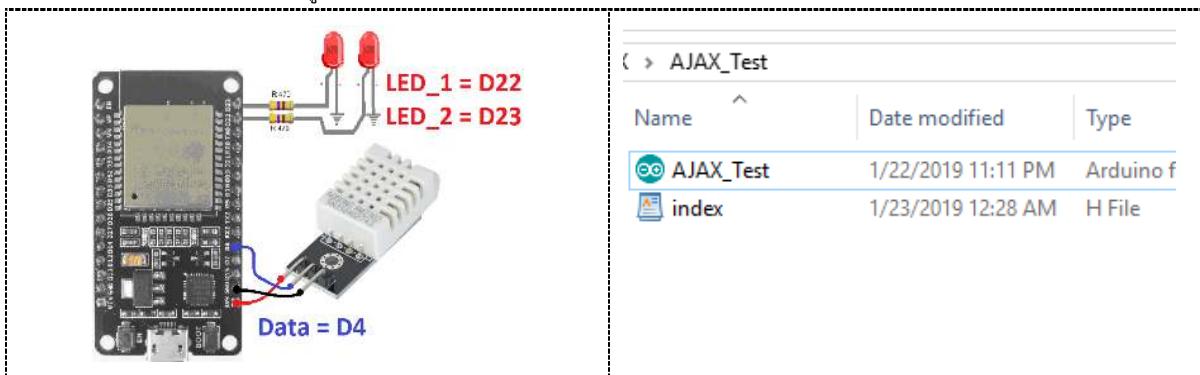
void loop() {
  WiFiClient client = server.available(); // wait Client request
  if (client) {
    Serial.println("new client"); // an http request ends with a blank line
    Serial.println("Requesting temperatures...");
    Serial.print("Temperature is: ");
    float Humid = dht.getHumidity();
    float cTemp = dht.getTemperature();
    Serial.print(cTemp, 2); Serial.print("C, ");
    Serial.print(Humid, 2); Serial.println("%H");
    boolean currentLineIsBlank = true;
    while (client.connected()) {
      if (client.available()) {
        char c = client.read();
        Serial.write(c);
        if (c == '\n' && currentLineIsBlank) // send a standard http response header
        { client.println("HTTP/1.1 200 OK");
          client.println("Content-Type: text/html");
          client.println("Connection: close"); // the connection will be closed after completion of the response
          client.println("Refresh: 5"); // refresh the page automatically every 5 sec
          client.println();
          client.println("<!DOCTYPE HTML>");
          client.println("<html>");
          client.print("Temperature > "); client.print(cTemp, 2);
          client.print("C, Humidity > "); client.print(Humid, 2);
          client.print("%H");
          client.println("<br />");
          client.println("</html>");
          break;
        }
        if (c == '\n') // you're starting a new line
        { currentLineIsBlank = true;
        }
        else if (c != '\r') // you've gotten a character on the current line
        { currentLineIsBlank = false;
        }
      }
    }
    delay(1); // give the web browser time to receive the data
    client.stop(); // close the connection:
    Serial.println("client disconnected");
  }
}
```

การทดลองที่ 4.7 – Web Server Monitor by AJAX

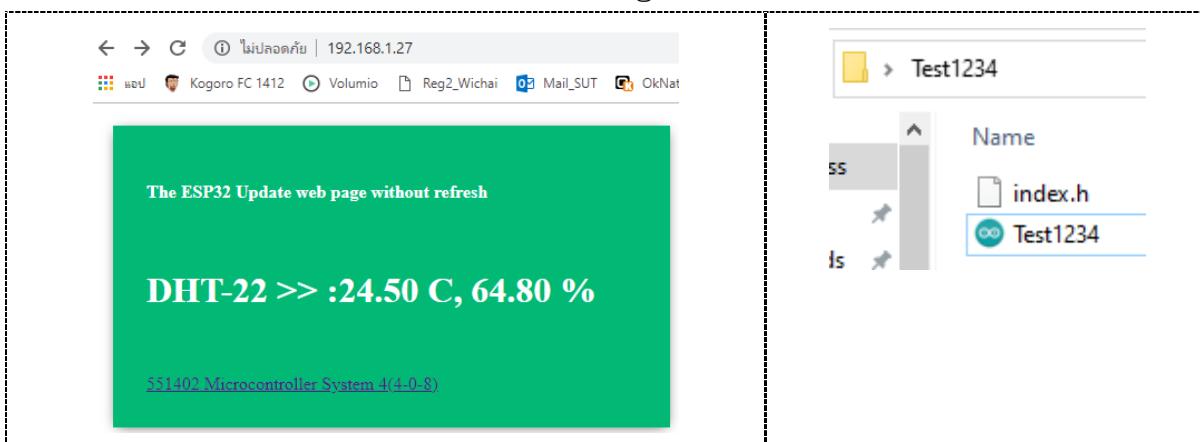
13. Test Ex50_WebServer AJAX_Monitor by AJAX

ในตัวอย่างก่อนหน้านี้ การ Update สถานะของการกด Button จะใช้การ refresh หน้า web ทั้งหมดทุกๆ 1 วินาที ทำให้ทั้งหน้ากระพริบ และ เป็นการรับ/ส่ง Data ที่ค่อนข้างลื่นเปลือย เนื่องจากในบางส่วนไม่ได้มีการเปลี่ยนแปลงแต่เราต้อง update ทั้งหน้า ในการทดลองนี้เราได้นำเอา Ajax เข้ามาช่วยให้สามารถ update ข้อมูลมาแสดงเฉพาะส่วนที่มีการเปลี่ยนแปลง ทำให้ไม่ต้อง refresh ทั้งหน้า และลดภาระของ Web server ให้ทำงานน้อยลง

- Read More <https://circuits4you.com/2018/11/20/web-server-on-esp32-how-to-update-and-display-sensor-values/>
- Read More <https://circuits4you.com/2018/02/04/esp8266-ajax-update-part-of-web-page-without-refreshing/>
- ดูวงจรดังรูป ดังนี้



- Create Program “Test_AJAX_01.ino”
- Open Notepad, Create “index.h” file and save to Test_AJAX_01 folder
- Upload “Test_AJAX_01.ino” to ESP-32
- เปิด Web Browser และเรียกหน้า Page ไปยัง IP ของ ESP-32



14. Code Web Server Monitor by AJAX

- Main Code → “Test_AJAX_01.ino”

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "index.h" //Web page header file

#include "DHTesp.h"
#define DHT_Pin 4 // DHT-11 Pin D4

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

DHTesp dht;
WebServer server(80);

void handleRoot() {
    String s = MAIN_page; //Read HTML contents
    server.send(200, "text/html", s); //Send web page
}

void handleADC() {
    float h = dht.getHumidity();
    float t = dht.getTemperature();
    String Value = String(t) + " C, ";
    Value += String(h) + "%";
    Serial.print("DHT-22 >> ");
    Serial.println(Value);
    Serial.println(Value);
    server.send(200, "text/plain", Value);
}

void setup(void) {
    Serial.begin(115200);
    Serial.println("\nBooting Sketch...");
    WiFi.mode(WIFI_STA); //Connect to your wifi
    WiFi.begin(ssid, password);
    Serial.println("Connecting to ");
    Serial.print(ssid);
}

while (WiFi.waitForConnectResult() != WL_CONNECTED) {
    Serial.print(".");
    delay(10);
}

Serial.print("\nConnected to ");
Serial.println(ssid);
Serial.print("IP address: ");
Serial.println(WiFi.localIP()); //IP address assigned to your ESP

server.on("/", handleRoot); //This is display page
server.on("/readADC", handleADC); //To get update of ADC Value only
server.begin(); //Start server
Serial.println("HTTP server started");
dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
}

void loop(void) {
    server.handleClient();
    delay(1);
}
```

- Include → “index.h”

```

const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<style>
.card{
  max-width: 500px;
  min-height: 250px;
  background: #02b875;
  padding: 30px;
  box-sizing: border-box;
  color: #FFF;
  margin:20px;
  box-shadow: 0px 2px 18px -4px rgba(0,0,0,0.75);
}
</style>
<body>

<div class="card">
<h4>The ESP32 Update web page without refresh</h4><br>
<h1>DHT-22 > :<span id="ADCValue">0</span></h1><br>
<br><a href="https://www.facebook.com/groups/311747285898180/">551402 Microcontroller System 4(4-0-8)</a>
</div>
<script>

setInterval(function() {
  // Call a function repetatively with 2 Second interval
  getData();
}, 2000); //2000mSeconds update rate

function getData() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("ADCValue").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "readADC", true);
  xhttp.send();
}
</script>
</body>
</html>
)=====";

```

การทดลองที่ 4.8 – Web Server Monitor and Control by AJAX

16. Test Ex51_WebServer AJAX_Control Monitor by AJAX

- Create Program “Test_AJAX_02.ino”
- Open Notepad, Create “index.h” file and save to Test_AJAX_02 folder
- Upload “Test_AJAX_02.ino” to ESP-32
- เปิด Web Browser และเรียกหน้า Page ไปยัง IP ของ ESP-32

The screenshot shows a web browser window with the following details:

- Address bar: ไม่ปลอดภัย | 192.168.160.22
- Toolbar icons: Back, Forward, Stop, Refresh, Home, and several other tabs like Kogoro FC 1412, Volumio, Reg2_Wichai, Mail_SUT, OkNation, and สำนักข่าวอิศรา.
- Main content area:
 - The ESP-32 Update web page without refresh**
 - Two sets of buttons for LED control:
 - LED1 ON (red)
 - LED1 OFF (blue)
 - Another set of buttons:
 - LED2 ON (red)
 - LED2 OFF (blue)
 - Text: State of [LED1, LED2] is >> NA
 - Text: DHT-22 sensor : Temp = 29.10 C, Humidity = 53.20 %
 - Text: [551402 Microcontroller System 4\(4-0-8\)](#)

- Code “Test_AJAX_02.ino” – WebServer AJAX Monitor by AJAX

```
#include <WiFi.h>
#include <WiFiClient.h>
#include <WebServer.h>
#include "DHTesp.h"

#include "index.h" //Our HTML webpage contents with javascripts
#define DHT_Pin 4
#define testLED1 22
#define testLED2 23

//SSID and Password of your WiFi router
const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";

WebServer server(80); //Server on port 80
DHTesp dht;

String ledState1 = "NA";
String ledState2 = "NA";

//=====
// This routine is executed when you open its IP in browser
//=====
void handleRoot() {
    String s = MAIN_page; //Read HTML contents
    server.send(200, "text/html", s); //Send web page
}

void handleADC() {
    float h = dht.getHumidity();
    float t = dht.getTemperature();
    String tmpValue = "Temp = ";
    tmpValue += String(t) + " C, Humidity = ";
    tmpValue += String(h) + " %";
    server.send(200, "text/plain", tmpValue); //Send value to client ajax request
}

void handleLED() {
    String t_state = server.arg("LEDstate"); //Refer xhttp.open("GET", "setLED?LEDstate="+led, true);
    Serial.println(t_state);
    if (t_state == "11") { digitalWrite(testLED1, HIGH); ledState1 = "ON"; } //Feedback parameter
    if (t_state == "10") { digitalWrite(testLED1, LOW); ledState1 = "OFF"; } //Feedback parameter
    if (t_state == "21") { digitalWrite(testLED2, HIGH); ledState2 = "ON"; } //Feedback parameter
    if (t_state == "20") { digitalWrite(testLED2, LOW); ledState2 = "OFF"; } //Feedback parameter
    server.send(200, "text/plain", ledState1+", "+ledState2); //Send web page
}

void setup(void) {
    Serial.begin(115200);
    dht.setup(DHT_Pin, DHTesp::DHT22); // DHT_Pin D4, DHT22
    pinMode(testLED1, OUTPUT);
    pinMode(testLED2, OUTPUT);
    Serial.print("\n\nConnect to ");
    Serial.println(ssid);
    WiFi.begin(ssid, password);
    while (WiFi.status() != WL_CONNECTED) {
        delay(500); Serial.print(".");
    }
    Serial.print("\nConnected "); Serial.println(ssid);
    Serial.print("IP address: "); Serial.println(WiFi.localIP());

    server.on("/", handleRoot);
    server.on("/setLED", handleLED);
    server.on("/readADC", handleADC);

    server.begin();
    Serial.println("HTTP server started");
}

void loop(void) {
    server.handleClient(); //Handle client requests
}
```

- Test_AJAX_02 = "index.h"

```

const char MAIN_page[] PROGMEM = R"=====(
<!DOCTYPE html>
<html>
<body>

<div id="demo">
<h1>The ESP-32 Update web page without refresh</h1>
<button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60);">LED1 ON</button>
<button type="button" onclick="sendData(10)" style="background: rgb(100,116,255);">LED1 OFF</button><br><br>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60);">LED2 ON</button>
<button type="button" onclick="sendData(20)" style="background: rgb(100,116,255);">LED2 OFF</button><br><br>
State of [LED1, LED2] is >> <span id="LEDState">NA</span><br>
</div>

<div>
<br>DHT-22 sensor : <span id="ADCValue">0</span><br>
</div>

<script>
function sendData(led) {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("LEDState").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "setLED?LEDstate="+led, true);
  xhttp.send();
}

setInterval(function() {
  // Call a function repetatively with 2 Second interval
  getData();
}, 2000); //2000mSeconds update rate

function getData() {
  var xhttp = new XMLHttpRequest();
  xhttp.onreadystatechange = function() {
    if (this.readyState == 4 && this.status == 200) {
      document.getElementById("ADCValue").innerHTML =
      this.responseText;
    }
  };
  xhttp.open("GET", "readADC", true);
  xhttp.send();
}
</script>

<br><a href="https://www.facebook.com/groups/557681561647621/">551402 Microcontroller System 4(4-0-8)</a>

</body>
</html>
)=====";

```

Quiz_202 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากมีกด Link ไปที่หน้า FB ของตัวเอง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2OjLGk

← → C ⓘ Not secure | 192.168.43.237

The ESP-32 Update web page without refresh

LED1 ON
LED2 ON
LED3 ON
LED4 ON

LED1 OFF
LED2 OFF
LED3 OFF
LED4 OFF

State of [LED1, LED2, LED3, LED4] is >> ON, OFF, OFF, ON

DHT-22 sensor : Temp = 28.10 C, Humidity = 43.90 %

[By Wichai Srisuruk](#)

```

58     ledState2 = "OFF";
59 }
60 if (t_state == "31") {
61   digitalWrite(testLED1, HIGH); //Feedback parameter
62   ledState3 = "ON";
63 }
64 if (t_state == "30") {
65   digitalWrite(testLED1, LOW); //Feedback parameter
66   ledState3 = "OFF";
67 }
68 if (t_state == "41") {
69   digitalWrite(testLED1, HIGH); //Feedback parameter
70   ledState4 = "ON";
71 }
72 if (t_state == "40") {
73   digitalWrite(testLED1, LOW); //Feedback parameter
74   ledState4 = "OFF";
75 }
76 server.send(200, "text/plain", ledState1 + ", " + ledState2 + ", " + ledState3 + ", " + ledState4); //Send web page
77 }
78

```

```

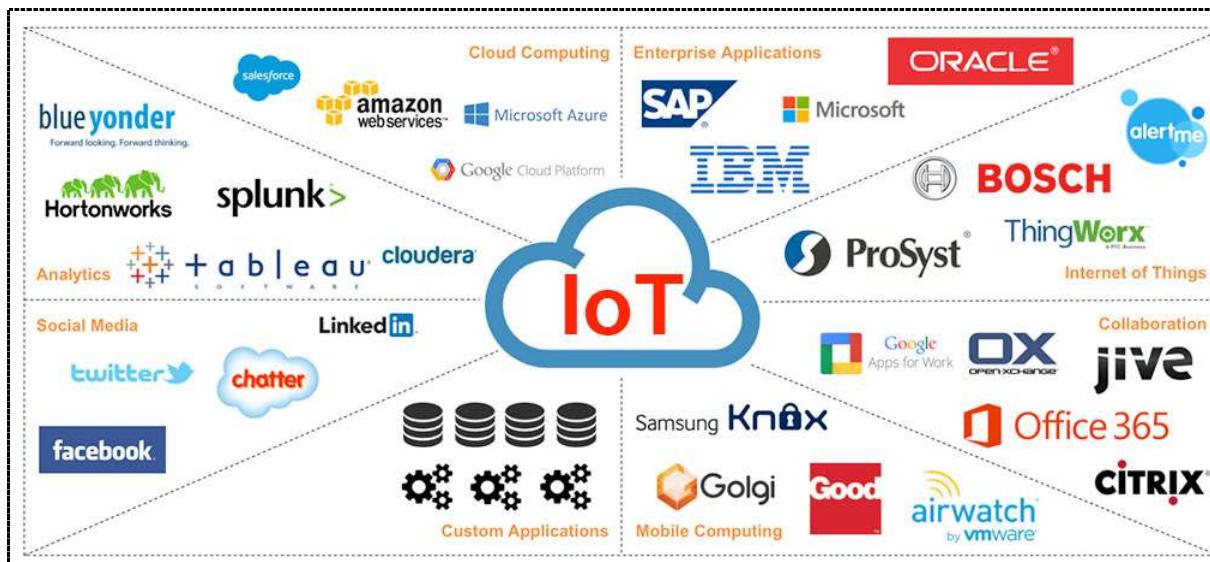
<div id="demo">
<h1>The ESP-32 Update web page without refresh</h1>
<button type="button" onclick="sendData(11)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED1 ON</button>
<button type="button" onclick="sendData(21)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED2 ON</button>
<button type="button" onclick="sendData(31)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED3 ON</button>
<button type="button" onclick="sendData(41)" style="background: rgb(202, 60, 60); height: 40px; width: 100px">LED4 ON</button><br><br>
<button type="button" onclick="sendData(10)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED1 OFF</button>
<button type="button" onclick="sendData(20)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED2 OFF</button>
<button type="button" onclick="sendData(30)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED3 OFF</button>
<button type="button" onclick="sendData(40)" style="background: rgb(100,116,255); height: 40px; width: 100px">LED4 OFF</button><br><br>
State of [LED1, LED2, LED3, LED4] is >> <span id="LEDState">NA</span><br>
</div>

```

5/6 -- การโปรแกรมเพื่อใช้งานแบบ IoTs ผ่าน MQTT proto-col

<https://mtonlia.com/10-free-public-private-mqtt-brokers-for-testing-prototyping/>

<https://www.hivemq.com/blog/mqtt-toolbox-mqtt-lens/>

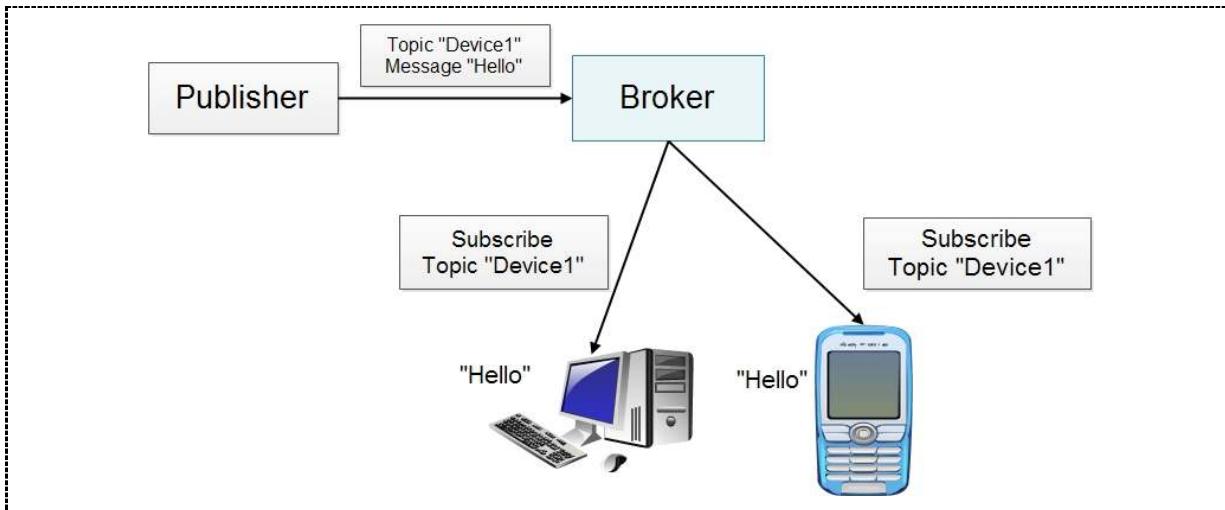


5.1 IoT Concept

ปัจจุบันเทคโนโลยีที่กำลังมาแรงสำหรับนักพัฒนาด้าน Embedded (ไม่ได้แค่เฉพาะ Embedded อย่างเดียวครับ) เป็นเทคโนโลยีที่กล่าวกันมากคงจะหนีไม่พ้น IOT ซึ่งเป็นเทคโนโลยีใหม่ในยุคนี้เลยก็ว่าได้ แรงขนาดที่ว่า Microsoft เอง ก็ยังพอร์ต Windows 10 มาไว้ลงบน Raspberry Pi แฉมยังใจดีติด IOT มาให้ด้วย ซึ่งผู้ผลิตไม่ได้ตามลงไปดูว่าใช้ Broker ตัวไหน และมี Library ให้ใช้งานมากด้วยหรือไม่? หรือไม่ผูกกีเข้าใจผิดเกี่ยวกับมันครับถ้าผิดพลาดก็ขออภัยมานะที่นี้ด้วยครับ.

IOT มันคืออะไร พอดีคุณดูมีหลายสิ่งอย่างที่ไม่แน่ใจ เช่น Internet of Things เมื่อคอมพิวเตอร์เริ่มคุยกันเองได้, โลกแห่ง IOT มาถึงแล้ว, IOT เทคโนโลยีที่ธุรกิจต้องรู้. ลองนึกภาพดูครับว่าถ้าหากอุปกรณ์สามารถสื่อสารกันได้ผ่าน www ไม่ว่าจะเป็น PC, Smart Phone หรือแม้แต่อุปกรณ์ขนาดเล็กพวก Micro-Controller, PLC, HUB, Switch หรืออะไรก็แล้วแต่ที่มันสามารถต่อระบบ Network ไม่ว่ามันจะอยู่ที่บ้าน ที่ทำงาน ไร่ นา ฟาร์มโรงเรือน โรงงานอุตสาหกรรมหรือที่อื่นๆที่มีระบบเน็ตเวอร์กที่เข้าถึง www ได้เราจะสามารถควบคุมมันได้ทั้งหมดที่ไหนก็ได้ในโลกใบนี้

องค์ประกอบหลักของ IoT จะมี 3 ส่วนคือ Broker, Publisher และ Subscriber. ซึ่งการรับและส่งข้อมูลนั้น มันจะส่งข้อมูลไปมหาภันนั่นจะส่งผ่านตัวกลางนั้นก็คือ Broker Server โดยตัวส่งนี้จะเรียกว่า Publisher ส่งข้อมูลเข้าไปยัง Broker พร้อมระบุหัวข้อ (Topic) ที่ต้องการส่งข้อมูลอีกไป จากนั้นตัวรับซึ่งเรียกว่า Subscriber ถ้าหากตัวรับต้องการรับข้อมูลจากตัวส่งจะต้องทำการ Subscribe หัวข้อ Topic ของ Publisher นั้นๆ ผ่าน Broker เช่นกัน



จากรูปภาพด้านบนจะมีด้วย Publisher ทำการ Publish ข้อความ “Hello” ใน Topic Device1 เมื่อและถ้าหากมีคอมพิวเตอร์ หรืออุปกรณ์อื่นๆทำการ Subscribe หัวข้อ Topic Device1 เมื่อ Publisher ทำการส่งข้อมูลไปยัง Topic อุปกรณ์ Subscribe จะได้ข้อความ “Hello” เชนเดียวกัน. ก็คล้ายๆกับที่ใช้งานไลน์ที่คุยกันเป็นกลุ่มนั้นเลยครับ. ซึ่งจะเห็นข้อความ “Hello” ในเวลาเดียวกันนั้นหมายความว่าอุปกรณ์ใดๆที่ทำการ Subscribe Topic เดียวกันก็จะได้ข้อความเดียวกันครับ

5.2 MQTT-Message Queue Telemetry Transport

โปรโตคอลที่ใช้สำหรับรับและส่งข้อมูลนั้นคือ MQTT ปัจจุบันถึง Version 3.1 ในที่นี้จะมาทำการทดลองส่งข้อมูลกันตัว Server จะมีอยู่ด้วยกันหลายค่ายครับสำหรับที่ลิสตามาด้านล่างนี้ครับ

Open Source MQTT Broker Server

- Mosquitto
- RSMB
- ActiveMQ
- Apollo
- Moquette
- Mosca
- RabbitMQ

Client

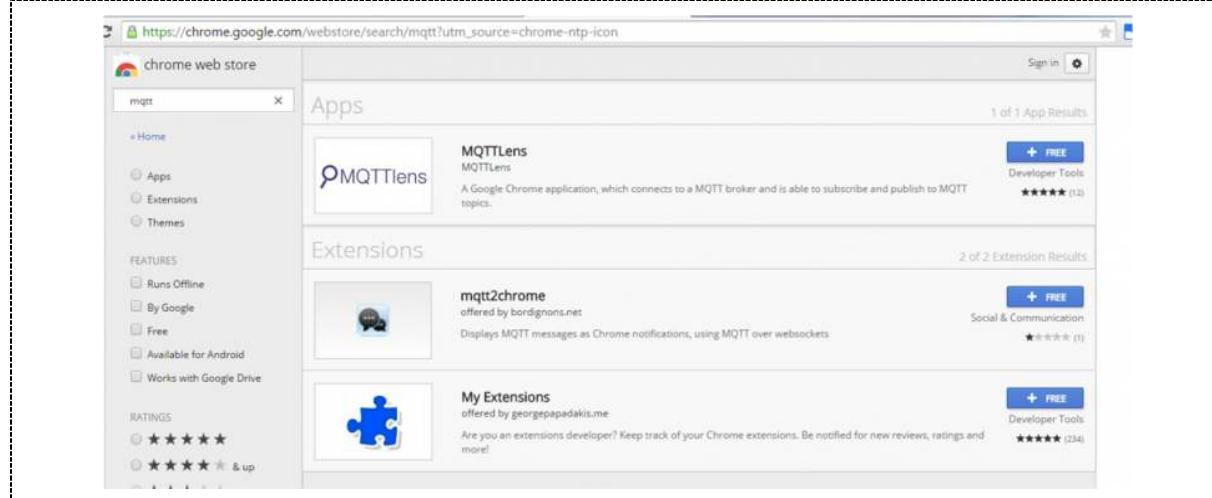
- Paho
- Xenqtt
- mqtt.js
- node_mqtt_client
- Ascoltatori
- Arduino MQTT Client

สำหรับ MQTT Broker Server ฟรีที่ผมพอกันได้ก็มีดังนี้ครับ

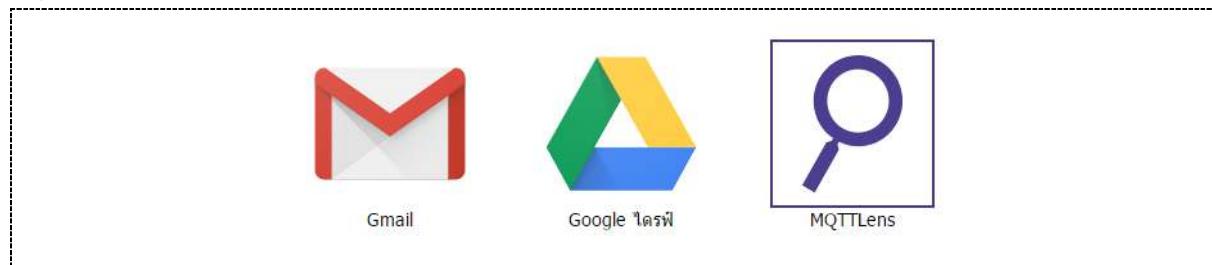
- test.mosquitto.org
- mqtt.eclipse.org
- broker.mqttdashboard.com

Step1a/3 กำหนดตัว Subscribe

สำหรับเครื่องมีสำหรับทดสอบที่จะทำการส่งข้อมูล(pub) และรับข้อมูล(sub) ก็มีอยู่ด้วยกันหลายตัวครับ เช่น แต่ละเลือกมาใช้งานลักษณะนี้ ในที่นี้ผมเลือกเป็น plugin สำหรับ chrome คือ MQTTLens



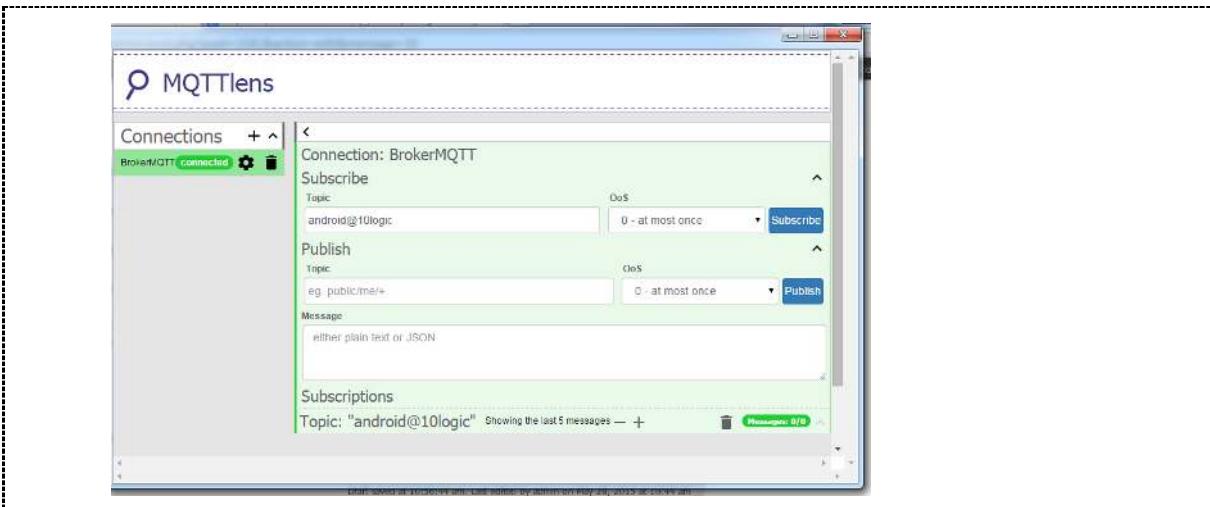
Mqttlens



mqttlens

เปิด MQTTLens ขึ้นมาจากหน้าป้อนรายละเอียด เมื่อป้อนรายละเอียดครบให้คลิกที่ CREATE CONNECTION

- Connection Name: test_MQTT ← อะไรก็ได้
- Hostname: test.mosquitto.org ← default = 1883
- Port: 1883 ← ตามที่ MQTTLens ให้มา
- Client ID: RXL77Nb



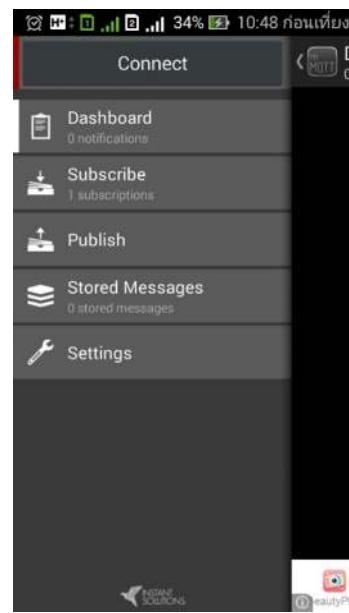
mqttlens

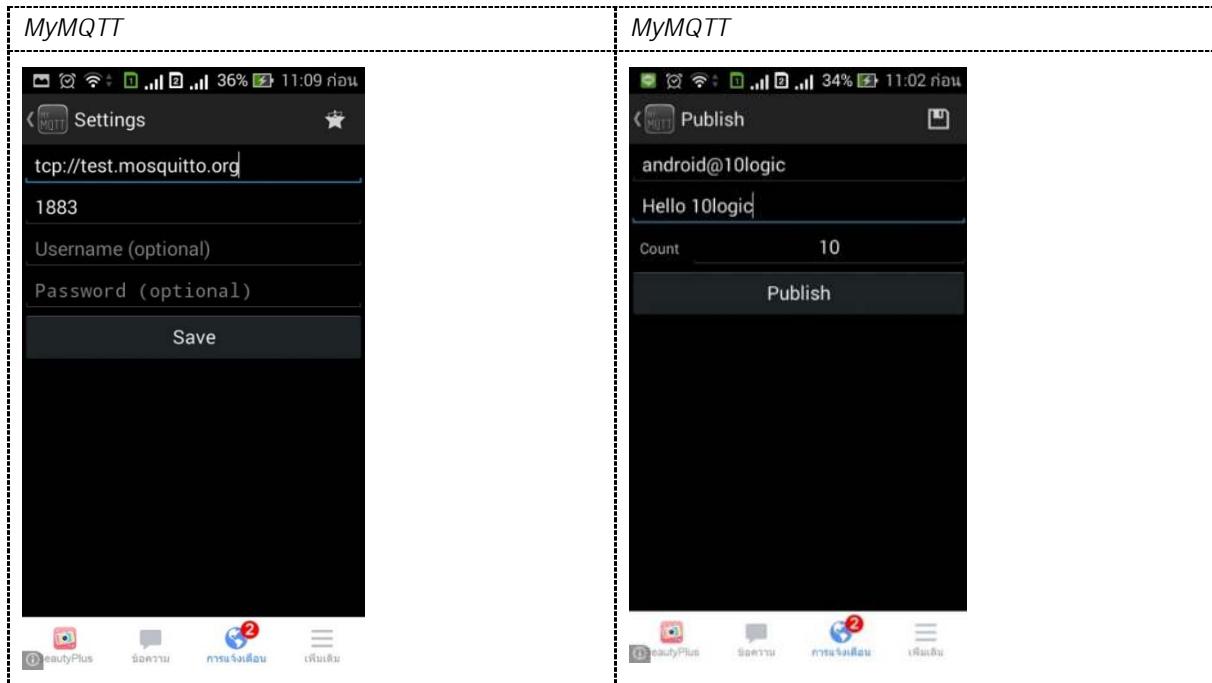
- Subscribe: android@10logic
ในที่นี่ผมจะทำการ Subscriber ที่ Topic ชื่อว่า android@10logic

Step2a/3: กำหนดตัว Publisher

Publisher ซึ่งเป็น App สำหรับ Android ทำการ Public ข้อความ Hello 10logic ไปยัง Topic android@10logic เข้าไปใน play store และค้นคำว่า MyMQTT แล้วติดตั้งลงบน Smart Phone ของเราครับ

MyMQTT
เปิด MyMQTT และเข้าไปยังเมนู Settings

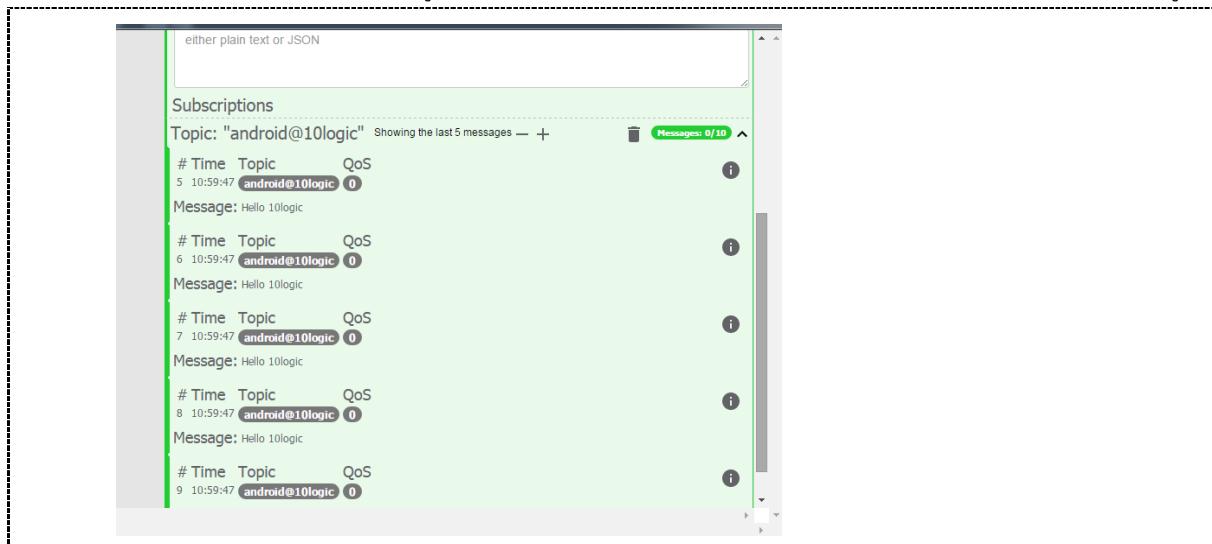




Step3a/3: ทดสอบการทำงาน

MyMQTT

ເລື່ອມໃຊ້ Smart Phone ທີ່ນີ້ຍຸ້ງການ Public ຂໍ້ຄວາມ “Hello 10logic” ພານ MyMQTT ແສດງຜລຕາມຮູບ



mqttlens

ຈະເຫັນວ່າສາມາຄຣັບຂໍ້ຄວາມ Hello 10logic ໄດ້ຕາມຕ້ວຍຍ່າງດັ່ງການ ເຮັນກາພກັນແລ້ວໃໝ່ເໜີມຄຣັບ ທີ່ນີ້ເລື່ອນັກພັດນາຕ້ອງກາຮັດສັງຂໍ້ມູນຈາກອຸປະນົມ embedded ສາມາຄຣັດສັງຂໍ້ມູນຫຼື້ນມາດໍາເຫັນກັນ

5.3 IoT และ MQTT คือ อะไร?

MQTT ย่อมาจาก Message Queue Telemetry Transport เป็นโปรโตคอลประยุกต์ที่ใช้โปรโตคอล TCP เป็นฐาน ออกแบบมาสำหรับงานที่ต้องการ ฯ สื่อสารแบบเรียลไทม์แบบไม่จำกัดแพลตฟอร์ม หมายถึงอุปกรณ์ทุกชิ้นสามารถสื่อสารกันได้ผ่าน MQTT

MQTT จะแบ่งเป็น 2 ฝั่ง คือผู้ส่งเชิร์ฟเวอร์มักจะเรียกว่า MQTT Broker ส่วนผู้รับผู้ใช้งานจะเรียกว่า MQTT Client ในการใช้งานด้าน IOT จะเกี่ยวข้องกับ MQTT Client เป็นหลัก โดยจะมี MQTT Broker ทึ้งแบบพรี และเสียเงินไว้รองรับอยู่แล้ว ทำให้การสื่อสารข้อมูลผ่าน MQTT จะใช้เชิร์ฟเวอร์พรี หรือ MQTT Broker พรี เหล่านี้เป็นตัวกลาง

ลักษณะการใช้งาน MQTT อาจจะเปรียบเสมือนได้กับการใช้งานห้องแชท Line สำหรับอุปกรณ์แต่ละตัวจะมีชื่อเป็นของตนเอง มี Username Password เป็นของตัวเอง และอาจจะมีห้องลับเฉพาะของตนเอง ตั้งนั้น การใช้งาน MQTT ผู้เขียนเชิญจะขอยกตัวอย่างของ MQTT เทียบกับห้องแชทได้ดังนี้

กลุ่มผู้ใช้ (User)

ใน MQTT จะแบ่งกลุ่มของผู้ใช้งานออกเป็น 2 ระดับ คือ

- ระดับสูงสุด – สามารถที่จะรับ-ส่งข้อมูลกับอุปกรณ์ หรือซองทางได้ ๆ ก็ได้ในระบบ หรือเปรียบได้กับแอดมินที่สามารถเข้าไปดูข้อความได้ทุกห้องแม้จะเป็นห้องลับก็ตาม
- ระดับทั่วไป – สามารถรับ-ส่งข้อมูลกับอุปกรณ์หรือซองทางที่กำหนดไว้เฉพาะเท่านั้น เปรียบได้กับผู้ใช้งาน Line ที่สามารถแชทในห้องที่ตัวเองสร้างได้ หรือเป็นสมาชิกในห้อง แต่ไม่สามารถเข้าไปแชทในห้องที่ไม่ได้เป็นสมาชิก

ในการใช้งานจริง ในอุปกรณ์ต่าง ๆ ควรจะใช้งานในระดับทั่วไป เพื่อความปลอดภัยกรณีอุปกรณ์เหล่านั้นถูกแฮกแล้วไม่สร้างความเสียหายไปยังอุปกรณ์อื่น ๆ ที่อยู่ในซองทางเฉพาะของแต่ละอุปกรณ์

เส้นทาง (Topic)

เส้นทาง เปรียบเหมือนกับหัวข้อ หรือห้องแชทที่ต้องการจะคุย และการคุยกันจะมีเฉพาะอุปกรณ์ที่อยู่ในห้องนั้น ๆ (Subscribe) ถึงจะสามารถได้รับข้อมูลที่มีการส่งไปในห้องนั้น ๆ ที่ถูกเรียกว่าเส้นทางเนื่องจากการใช้งานส่งข้อมูลและรับข้อมูลจะเหมือนกับเส้นทางในระบบไฟล์ เช่น /Room1/LED ซึ่งระบบเส้นทางนี้นอกจากอุปกรณ์จะสามารถรับการสนทนainห้องตามเส้นทาง /Room1/LED ได้แล้ว ยังสามารถรับสนทนาระหว่างห้อง /Room1 ได้ด้วย หากเป็นการรับฟังในเส้นทาง(Subscribe) /Room1 จะหมายถึงการส่งข้อมูลได้ ๆ ที่นำหน้าด้วย /Room1 เช่น /Room1/LED , /Room1/Value ผู้ที่รอฟัง (Subscribe) /Room1 อยู่จะได้รับข้อมูลเหล่านั้นด้วย

คุณภาพข้อมูล (QoS)

แบ่งออกเป็น 3 ระดับดังนี้

- QOS0 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่
- QOS1 – ส่งข้อมูลเพียงครั้งเดียว ไม่สนใจว่าผู้รับจะได้รับหรือไม่ แต่ให้จำค่าที่ส่งล่าสุดไว้ เมื่อมีการเชื่อมต่อใหม่จะได้รับข้อมูลครั้งล่าสุดอีกครั้ง
- QOS2 – ส่งข้อมูลหลาย ๆ ครั้งจนกว่าปลายทางจะได้รับข้อมูล มีข้อเสียที่สามารถทำงานได้ช้ากว่า QOS0 และ QOS1

การส่งข้อมูล (Publish)

การส่งข้อมูลในแต่ละครั้งจะต้องประกอบไปด้วยเล็กน้อย ทางข้อมูล และคุณภาพข้อมูล ซึ่งการส่งข้อมูลจะเรียกว่า Publish

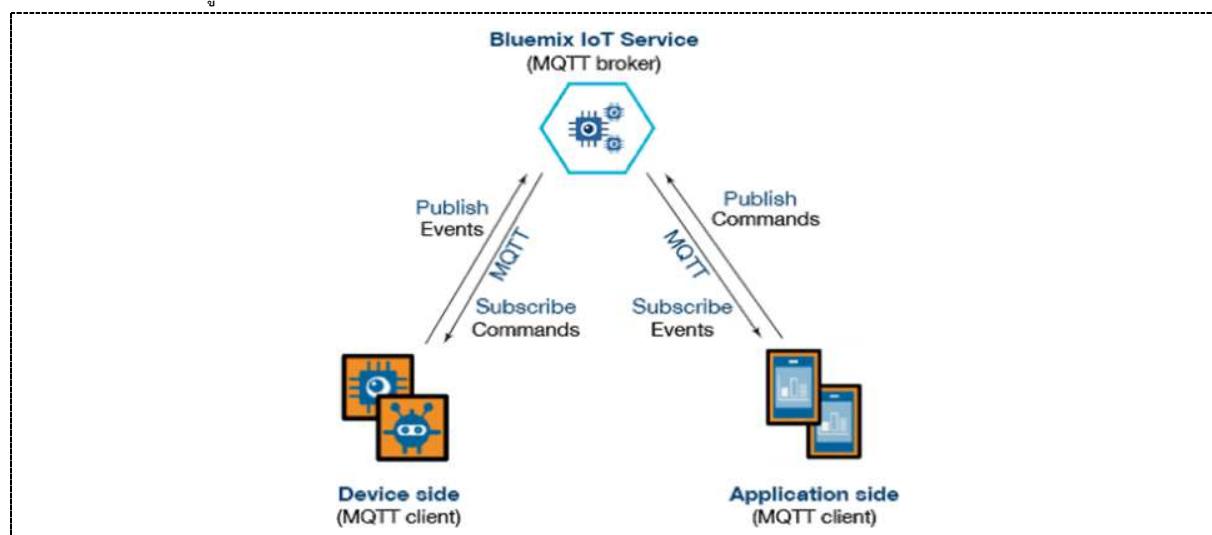
การรับข้อมูล (Subscribe)

การรับข้อมูลในระบบ MQTT จะรับข้อมูลได้เฉพาะเมื่อมีการเรียกใช้การ Subscribe ไปยัง Topic ที่กำหนด อาจเบรียบได้กับการ Subscribe คือการเข้าไปนั่งรอเพื่อนในกลุ่ม Line ส่งแซทมาหา เมื่อมีการส่งข้อมูลเข้ามาจะเกิด สิ่งที่เรียกว่าเหตุการณ์ (Event) ให้เรากดเข้าไปดูข้อความที่เพื่อน ๆ ส่งเข้ามา

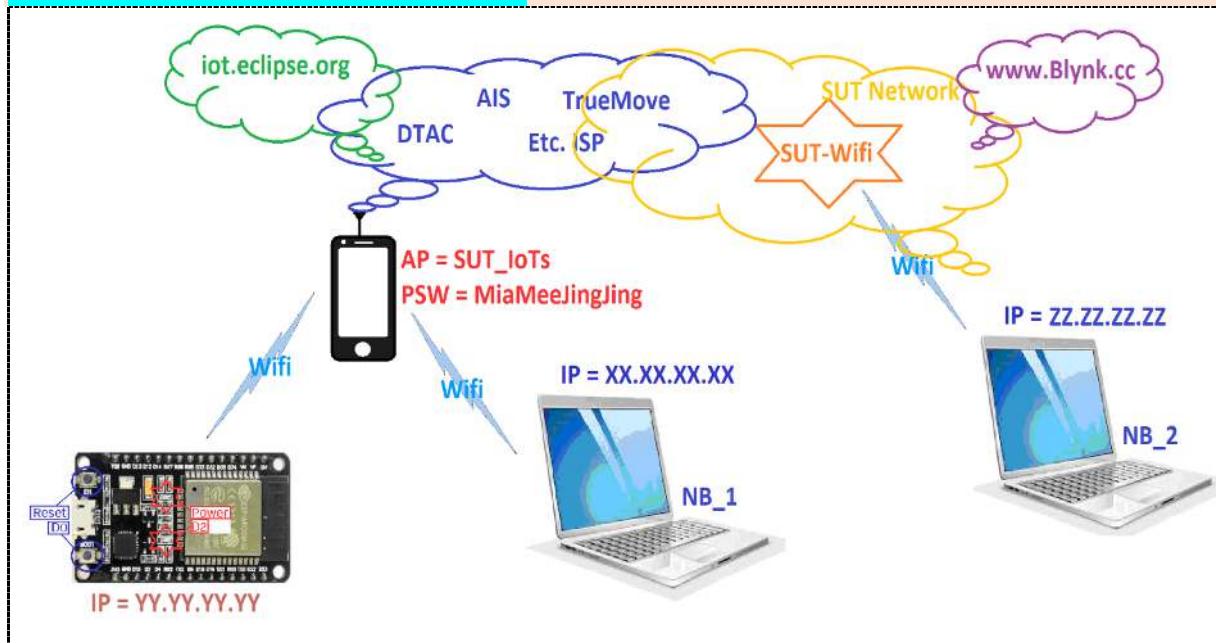
จะเห็นได้ว่า MQTT ก็เปลี่ยนแปลงห้องแซทของอุปกรณ์ที่จะสนทนากันแลกเปลี่ยนข้อมูลกันแบบเรียลไทม์ผ่าน เครือข่ายอินเตอร์เน็ต

5.4 IoT มีวิธีการทำงานอย่างไร?

องค์ประกอบหลักของ IoT จะมี 3 ส่วนคือ Broker, Publisher และ Subscriber. ซึ่งการรับและส่งข้อมูลนั้น มักจะส่งข้อมูลไปมากกันนั้นจะส่งผ่านตัวกลางนั้นก็คือ Broker Server โดยตัวส่งนี้จะเรียกว่า Publisher ส่งข้อมูลเข้าไปยัง Broker พร้อมระบุหัวข้อ (Topic) ที่ต้องการส่งข้อมูลไป จากนั้นตัวรับซึ่งเรียกว่า Subscriber ถ้าหากตัวรับ ต้องการรับข้อมูลจากตัวส่งจะต้องทำการ Subscribe หัวข้อ Topic ของ Publisher นั้นๆ ผ่าน Broker เช่นกัน ลองดู ความลับพันธ์ ตามรูป



5.5 ข้อแตกต่างระหว่าง IoT กับ Over Internet



การทดลองก่อนหน้าเป็นการควบคุมผ่านอินเตอร์เน็ต จำเป็นต้องรู้ IP ของอุปกรณ์ปลายทาง และระบบต้องอยู่ในวงเครือข่ายเดียวกัน เช่น จากรูปเราไม่สามารถใช้ NB_2 เข้ามาควบคุม ESP32 ได้ โดยตรง เพราะ IP=ZZ.ZZ.ZZ.ZZ และ IP=YY.YY.YY.YY อยู่คนละเครือข่าย หากต้องการสามารถกำหนดเส้นทางจาก NB_2 ผ่านเครือข่ายของมหาวิทยาลัย ไปยังผู้ให้บริการมือถือ วนมาที่มือถือ เข้ามายัง ESP32 การควบคุมสิ่งการแบบนี้จำเป็นต้องรู้เลขปลายทางซึ่งเป็นไอพีของอุปกรณ์

กรณีของ IOTS กระบวนการข้างต้นจะปรับใหม่ คือ ไก่จำเป็นต้องรู้เลขไอพีของอุปกรณ์ปลายทางแต่ให้อุปกรณ์วิ่งไปรับคำสั่งที่ตัวกลาง (Broker) แทน จากรูป NB_2 จะส่งคำสั่ง (Publish) ไปยังตัวกลาง ตัวอุปกรณ์ปลายทางต้องแจ้งรับข้อมูล (Subscribe) จากตัวกลาง เมื่อมีคำสั่งเข้ามาและตัวอุปกรณ์ปลายทางข้ามารับข้อมูลอุปกรณ์ปลายทางค่อยทำงานตามคำสั่งที่ได้รับ

เห็นได้ว่าแบบแรกจำเป็นต้องเข้าให้ถึง ESP32 แต่แบบหลังใช้วิธีนัดรับข้อมูลที่ตัวกลางที่ทั้งสองฝ่ายตกลงกันไว้

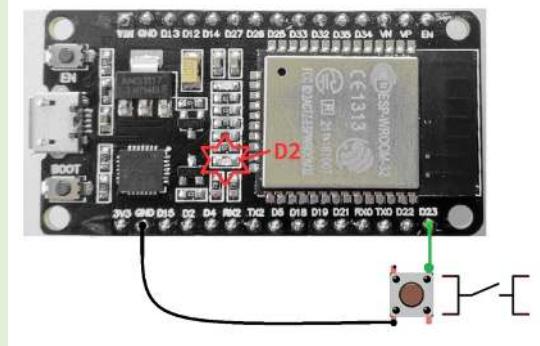
การทดลองที่ 5.1 – Publish

1. ทดสอบโปรแกรมนับจำนวนครั้งการกดสวิตช์ และแสดงผลที่ Serial Monitor ด้วย baud rate 115200

```
int pushButton = 23;
int LED_Monitor = 2;
int Counter = 0;

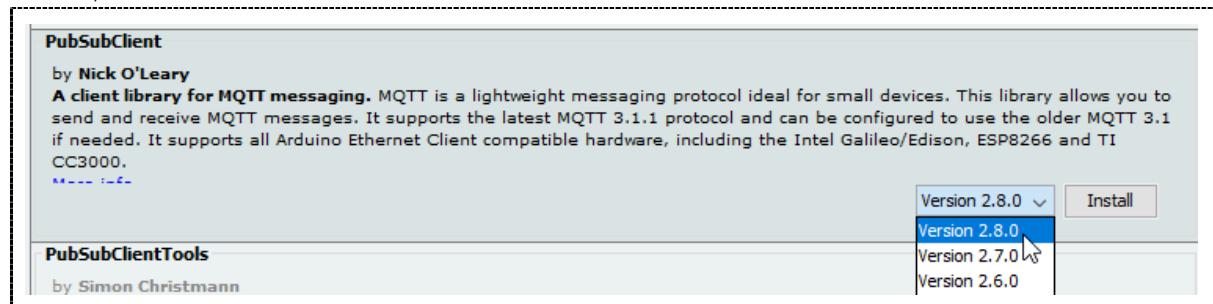
void setup() {
  Serial.begin(115200);
  pinMode(LED_Monitor, OUTPUT);
  pinMode(pushButton, INPUT_PULLUP);
  Serial.print(" Counter = ");
  Serial.println(Counter);
  digitalWrite(LED_Monitor, Counter % 2);
}

void loop() {
  if (digitalRead(pushButton) == 0)
  { Counter++;
    Serial.print(" Counter = ");
    Serial.println(Counter);
    digitalWrite(LED_Monitor, Counter % 2);
    while (digitalRead(pushButton) == 0);
    delay(100);
  }
}
```



```
COM7
Counter = 1
Counter = 2
Counter = 3
```

2. Add Library → การใช้งาน MQTT บน ESP32 จะใช้งานผ่านไฟล์ PubSubClient.h จะต้องติดตั้งเพิ่มเติม โดยใช้ Library Manager ค้นหาคำว่า PubSubClient เลือก PubSubClient Version 2.8.0 และสามารถกดปุ่ม Install เพื่อติดตั้ง



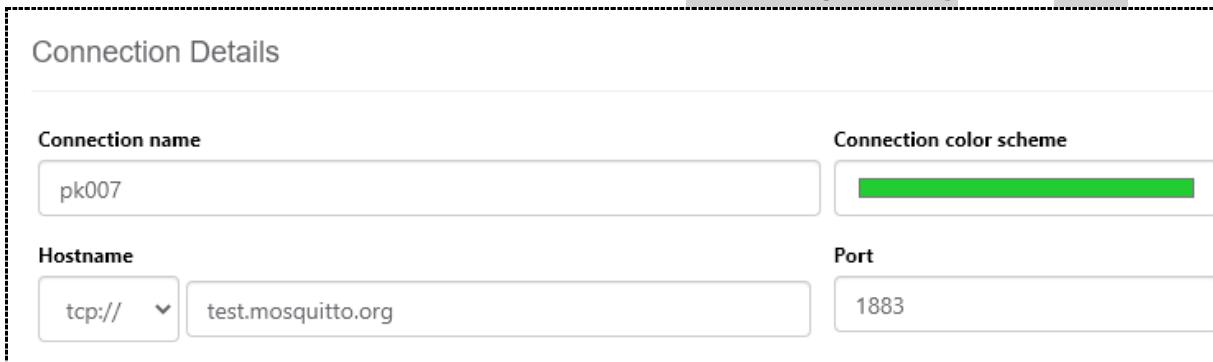
3. โปรแกรมจะ Publish ไปยัง **test.mosquitto.org** ในหัวข้อ **myHome1234**

- Test Broker = **test.mosquitto.org**
- Port = **1883**
- Topic, Subscribe, Publish = **myHome1234**



MQTTLens

4. คำแนะนำการติดตั้งใช้งานตามนี้ <http://www.steves-internet-guide.com/using-mqtt-lens/>
5. เปิด MQTT Lens ตั้งค่าเบรคเกอร์เป็น → Test Broker = **test.mosquitto.org**, Port = **1883**,



6. กำหนด Topic → Topic, Subscribe, Publish = **myHome1234**

Connections + ^ | Pk007

Subscribe
Topic: myHome1234 QoS: 0 - at most once SUBSCRIBE

Publish
Topic: myHome1234 QoS: 0 - at most once Retained PUBLISH
Message: hello

Subscriptions
Topic: "myHome1234" Showing the last 5 messages — +
Time Topic QoS
0 7:00:19 myHome1234 0
Message: hello

Messages: 0/1

ลีเดง - ยังไม่พร้อมใช้งาน | สีเขียว - พร้อมใช้งาน

Connections + ^ | Pk007

Connections + ^ | Pk007

7. โปรแกรมทดสอบ Publish ไปยัง **test.mosquitto.org** ในหัวข้อ **myHome1234**

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

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "myHome1234";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];
int value = 0;

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void reconnect()
{
  while (!client.connected()) // Loop until we're reconnected
  {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX); // Create a random client ID
    if (client.connect(clientId.c_str())) // Attempt to connect
    {
      Serial.println("connected"); // Once connected, publish an announcement...
      client.publish(topic1, "Hello World Pk007"); // ... and resubscribe
      client.subscribe(topic1);
    }
    else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup()
{
  Serial.begin(115200);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}

void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
  long now = millis();
  if (now - lastMsg > 2000)
  {
    lastMsg = now;
    ++value;
    sprintf(msg, 75, "hello world #%d", value);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(topic1, msg);
  }
}
```

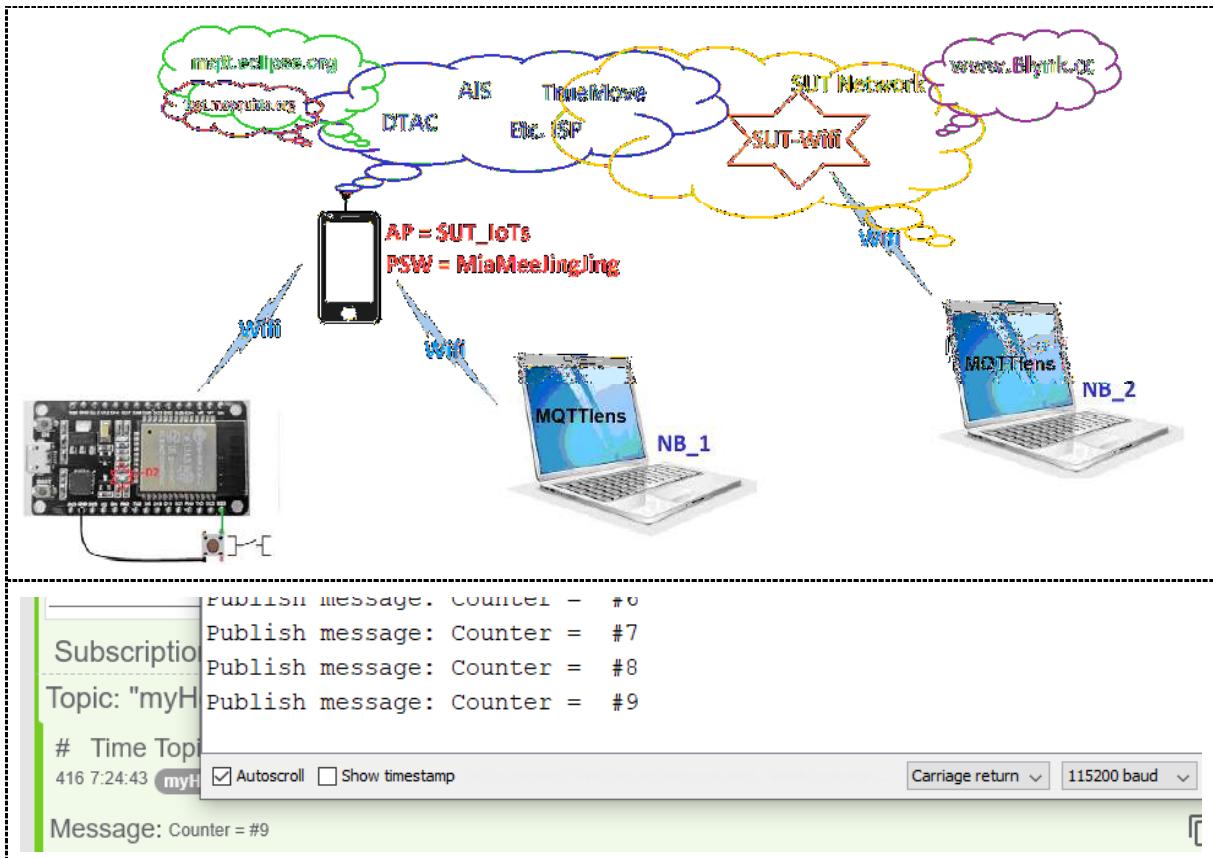
8. ผลการทำงานดูที่ Serial Monitor ว่าส่งอะไรออกไป และดูที่ MQTTLens ว่าได้รับอะไรมาบ้าง

```

.....
WiFi connected
IP address:
192.168.1.30
Attempting MQTT connection...connected
Publish message: hello world #1
Publish message: hello world #2
Publish message: hello world #3
Publish message: hello world #4
Publish message: hello world #5
Publish message: hello world #6
Publish message: hello world #7
Publish message: hello world #8
Publish message: hello world #9

Subscriptions
Topic: "myHome1234" Showing the last 1 messages — +
# Time Topic QoS
10 7:09:33 myHome1234 0
Message: hello world #9
  
```

9. ทดลองโปรแกรมนับจำนวนครั้งการกดสวิตช์ ส่งค่าไปยัง MQTT Broker



การทดลองที่ 5.2 – Switch Counter and Publish

11. โปรแกรมนับจำนวนครั้งการกดสวิตช์ ส่งค่าไปยัง MQTT Broker

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

const char* ssid = "SUT_IoT's";
const char* password = "MaiMeeJingJing";
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "myHome1234";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];

int pushButton = 23;
int LED_Monitor = 2;
int Counter = 0;

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to "); Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500); Serial.print(".");
  }
  randomSeed(micros());
  Serial.println(""); Serial.println("WiFi connected");
  Serial.println("IP address: "); Serial.println(WiFi.localIP());
}

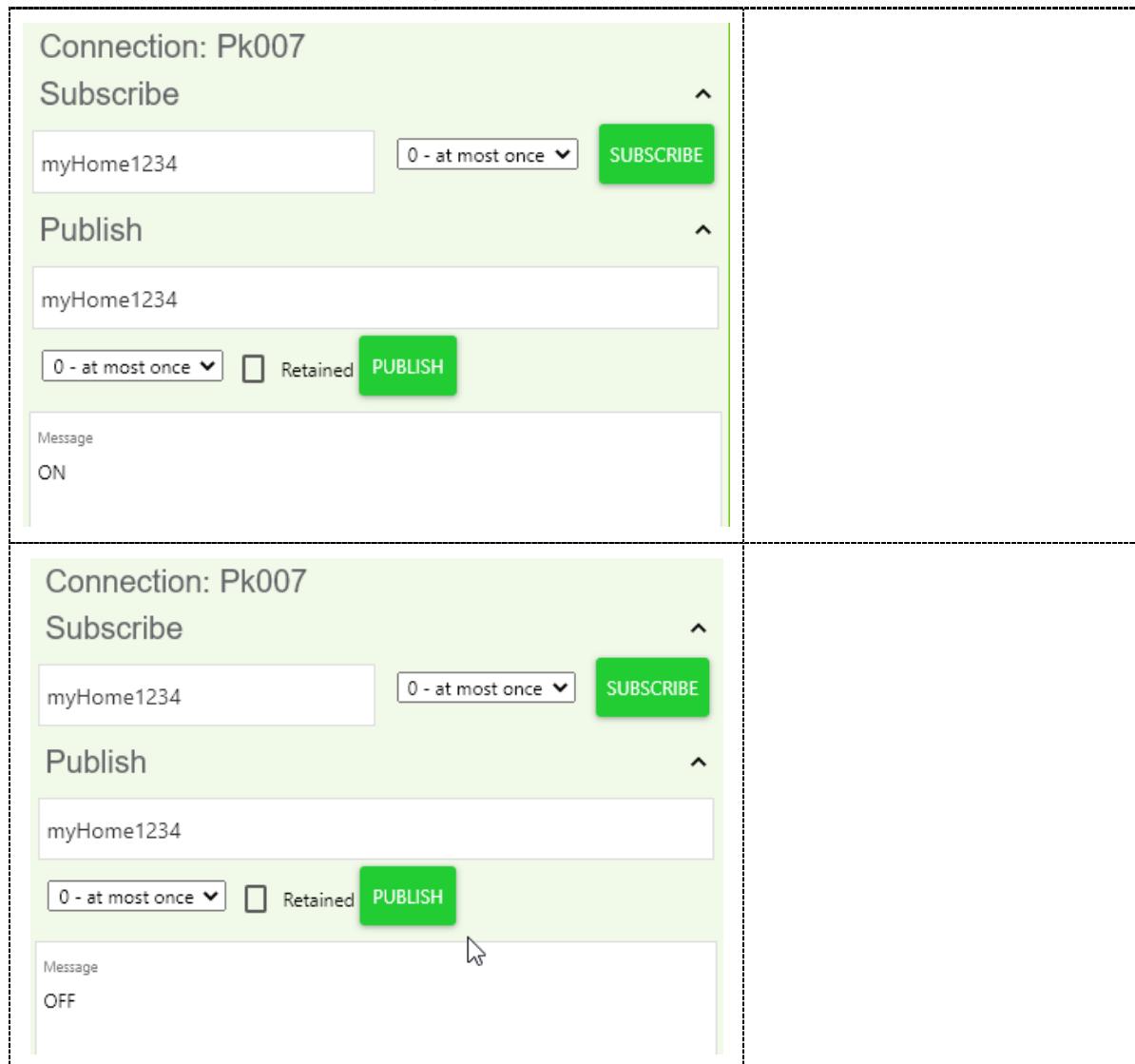
void reconnect()
{
  while (!client.connected()) // Loop until we're reconnected
  {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX); // Create a random client ID
    if (client.connect(clientId.c_str())) // Attempt to connect
    {
      Serial.println("connected"); // Once connected, publish an announcement...
      client.publish(topic1, "Hello World Pk007"); // ... and resubscribe
      client.subscribe(topic1);
    } else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

void setup()
{
  Serial.begin(115200);
  pinMode(LED_Monitor, OUTPUT);
  pinMode(pushButton, INPUT_PULLUP);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
}

void loop()
{
  if (digitalRead(pushButton) == 0)
  {
    Counter++;
    if (!client.connected()) reconnect();
    client.loop();
    sprintf(msg, 75, "Counter = #%ld", Counter);
    Serial.print("Publish message: ");
    Serial.println(msg);
    client.publish(topic1, msg);
    digitalWrite(LED_Monitor, Counter % 2);
    while (digitalRead(pushButton) == 0);
    delay(100);
  }
}
```

การทดลองที่ 5.3 – Publish and Subscribe

12. เมื่อโหลดโปรแกรมแล้วทำการทดสอบ publish “OFF” → Off LED และ “ON” → On LED



13. โปรแกรมการ Publish และ Subscribe หน่วงเวลาครึ่งละ 5 วินาที

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

const char* ssid = "SUT_IoT";
const char* password = "MaiMeeJingJing";
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "myHome1234";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];

int TestLED = 2;
int value = 0;

void setup_wifi() {
  delay(10);
  Serial.println();
}
```

```

Serial.print("Connecting to ");
Serial.println(ssid);
WiFi.begin(ssid, password);
while (WiFi.status() != WL_CONNECTED) {
  delay(500);  Serial.print(".");
}
randomSeed(micros());
Serial.println("");
Serial.println("WiFi connected");
Serial.println("IP address: ");
Serial.println(WiFi.localIP());
pinMode(TestLED, OUTPUT);
}

void callback(char* topic, byte* payload, unsigned int length)
{ char myPayLoad[50];
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("]");
  for (int i = 0; i < length; i++)
  { Serial.print((char)payload[i]);
    myPayLoad[i] = payload[i];
    myPayLoad[i + 1] = '\0'; // End of String
  }
  Serial.print("\n--> ");
  Serial.println(myPayLoad);
  myPayLoad[4] = '\0'; // String lessthan 4 Charector
  if ((String)myPayLoad == "ON") digitalWrite(TestLED, HIGH);
  if ((String)myPayLoad == "OFF") digitalWrite(TestLED, LOW);
}

void reconnect()
{ while (!client.connected())           // Loop until we're reconnected
{ Serial.print("Attempting MQTT connection...");
  String clientId = "ESP8266Client-";
  clientId += String(random(0xffff), HEX); // Create a random client ID
  if (client.connect(clientId.c_str())) // Attempt to connect
  { Serial.println("connected");        // Once connected, publish an announcement...
    client.publish(topic1, "Hello World Pk007"); // ... and resubscribe
    client.subscribe(topic1);
  } else
  { Serial.print("failed, rc=");
    Serial.print(client.state());
    Serial.println(" try again in 5 seconds");
    delay(5000);
  }
}
}

void setup()
{ Serial.begin(115200);
setup_wifi();
client.setServer(mqtt_server, 1883);
client.setCallback(callback);
pinMode(TestLED, OUTPUT);
}

void loop()
{ if (!client.connected()) reconnect();
client.loop();
long now = millis();
if (now - lastMsg > 2000)
{ lastMsg = now;
++value;
sprintf(msg, "hello world #%d", value);
Serial.print("Publish message: ");
Serial.println(msg);
client.publish(topic1, msg);
}
}

```

การทดลองที่ 5.4 – Switch Counter, Publish and Subscribe

14. ทดสอบโปรแกรมนับจำนวนครั้งการกดสวิตช์ ส่งค่าไปยัง MQTT Broker และควบคุมการเปิด-ปิด LED

```
// DOI ESP-32

#include <WiFi.h>
#include <PubSubClient.h>

const char* ssid = "SUT_IoTs";
const char* password = "MaiMeeJingJing";
const char* mqtt_server = "test.mosquitto.org";
const char* topic1 = "myHome1234";

WiFiClient espClient;
PubSubClient client(espClient);
long lastMsg = 0;
char msg[50];

int TestLED = 2;
int pushButton = 23;
int Counter = 0;

void setup_wifi() {
  delay(10);
  Serial.println();
  Serial.print("Connecting to ");
  Serial.println(ssid);
  WiFi.begin(ssid, password);
  while (WiFi.status() != WL_CONNECTED) {
    delay(500);
    Serial.print(".");
  }

  randomSeed(micros());
  Serial.println("");
  Serial.println("WiFi connected");
  Serial.println("IP address: ");
  Serial.println(WiFi.localIP());
}

void callback(char* topic, byte* payload, unsigned int length)
{
  char myPayLoad[50];
  Serial.print("Message arrived [");
  Serial.print(topic);
  Serial.print("] ");
  for (int i = 0; i < length; i++)
  {
    Serial.print((char)payload[i]);
    myPayLoad[i] = payload[i];
    myPayLoad[i + 1] = '\0'; // End of String
  }
  Serial.print("\n--> ");
  Serial.println(myPayLoad);
  myPayLoad[4] = '\0'; // String less than 4 Charector
  if ((String)myPayLoad == "ON") digitalWrite(TestLED, HIGH);
  if ((String)myPayLoad == "OFF") digitalWrite(TestLED, LOW);
}

void reconnect()
{
  while (!client.connected()) // Loop until we're reconnected
  {
    Serial.print("Attempting MQTT connection...");
    String clientId = "ESP32 Client-";
    clientId += String(random(0xffff), HEX); // Create a random client ID
    if (client.connect(clientId.c_str())) // Attempt to connect
    {
      Serial.println("connected"); // Once connected, publish an announcement...
      client.publish(topic1, "Hello World Pk007"); // ... and resubscribe
      client.subscribe(topic1);
    }
    else
    {
      Serial.print("failed, rc=");
      Serial.print(client.state());
      Serial.println(" try again in 5 seconds");
      delay(5000);
    }
  }
}

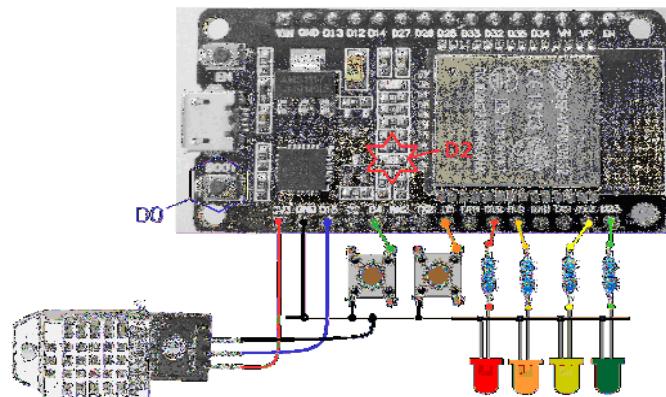
void setup()
{
  Serial.begin(115200);
  pinMode(pushButton, INPUT_PULLUP);
  pinMode(TestLED, OUTPUT);
  setup_wifi();
  client.setServer(mqtt_server, 1883);
  client.setCallback(callback);
}

void loop()
{
  if (!client.connected()) reconnect();
  client.loop();
}
```

```
if (digitalRead(pushButton) == 0)
{ Counter++;
  client.loop();
  sprintf (msg, "Count = %d", Counter);
  Serial.print("Publish message: ");
  Serial.println(msg);
  client.publish(topic1, msg);
  while (digitalRead(pushButton) == 0);
  delay(100);
}
```

Quiz_103 – Pub/Sub Data from (DHT22 + 4 LED + 2 Switch)

- อ่านค่า DHT-22 และส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
- กำหนดให้ใช้ mqtt.eclipse.org เป็น Broker
- ควบคุมการเปิดปิด 4 LED
- รับคำสั่งที่กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm

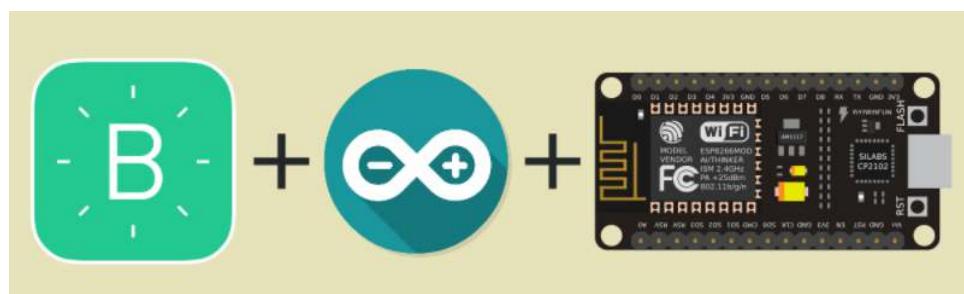


5.6 Blynk IoT Platform

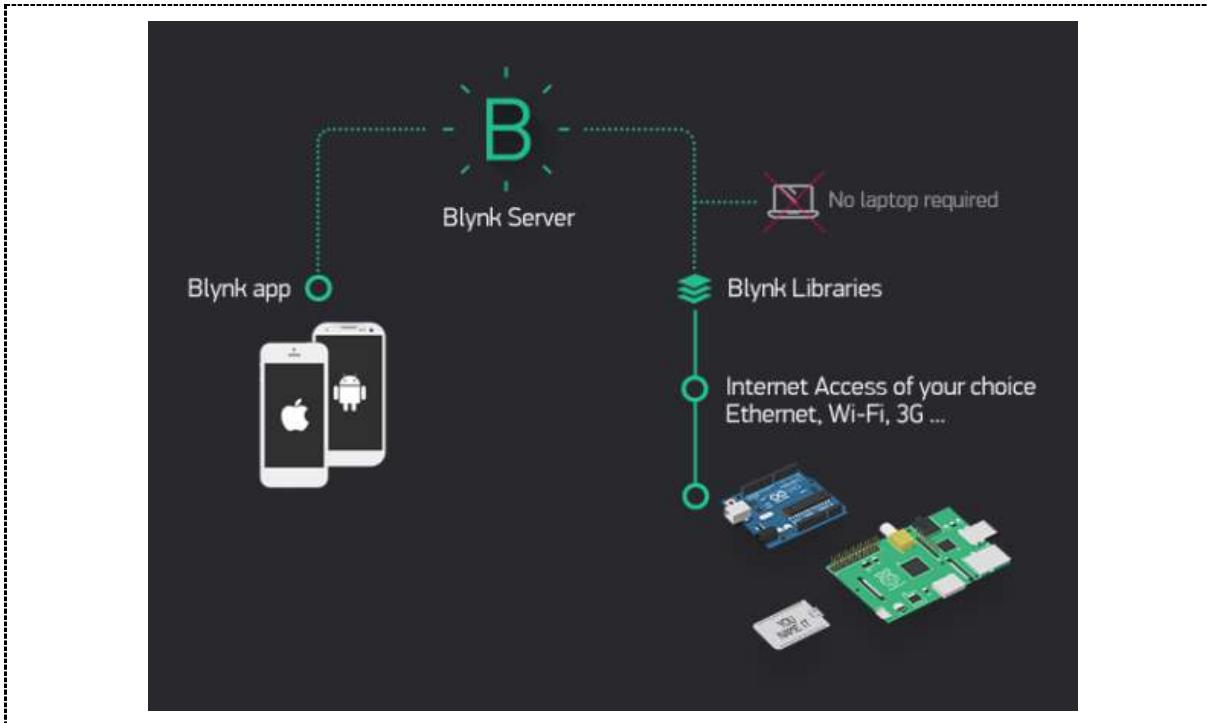
<http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>



Blynk เป็น cloud platform ที่ให้บริการฟรี สำหรับ IOT, Blynk เป็น Application สำหรับระบบงาน IOT มีความน่าสนใจคือการเขียนโปรแกรมที่ง่าย ไม่ต้องเขียน App เองสามารถใช้งานได้อย่าง Real time สามารถเชื่อมต่อ Device ต่างๆเข้ากับ Internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, Esp8266, Esp32, Nodemcu, Raspberry pi นำมาแสดงบน Application ได้อย่างง่ายดาย และที่สำคัญ Application Blynk ยังฟรี และ รองรับในระบบ IOS และ Android อีกด้วย



Blynk สามารถเชื่อมต่ออุปกรณ์ Device ของเราเข้ากับ internet ได้อย่างง่ายดาย ไม่ว่าจะเป็น Arduino, ESP8266, Raspberry pi หรือแม้แต่เครื่องที่รวมเอา widget ต่างๆมาควบคุมแทนการเขียน code ยากๆ ไม่เพียงเท่านั้น ทางเลือกในการเชื่อมต่อเข้ากับ Blynk server เราสามารถใช้ได้ทั้ง WiFi และเครือข่ายมือถือ โดยสามารถ Download application ฟรีได้ทั้งระบบ IOS และ Android



อ่านเพิ่มเติม

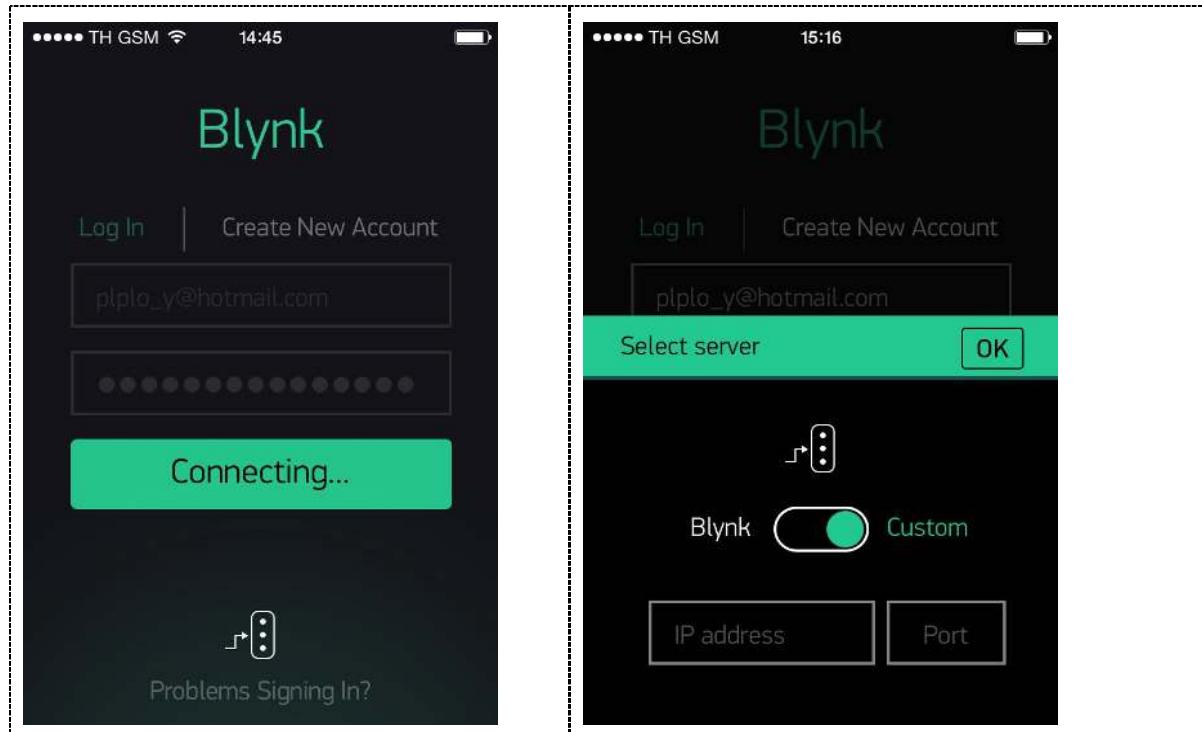
- <https://www.blynk.cc/>, <http://docs.blynk.cc/>
- <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- <https://www.9arduino.com/article/59/app-สำเร็จรูป-blynk-nodemcu-esp8266-ตอนที่-1-blynk-คืออะไร>
- <http://www.ayarafun.com/2015/08/easy-iot-play-with-blynk/>
- <https://github.com/blynkkk/blynk-server>
- <http://thaiopensource.org/มาเรียน-blynk-กับ-esp8266-กัน/>

การทดลองที่ 5.5 – Blynk LED Control

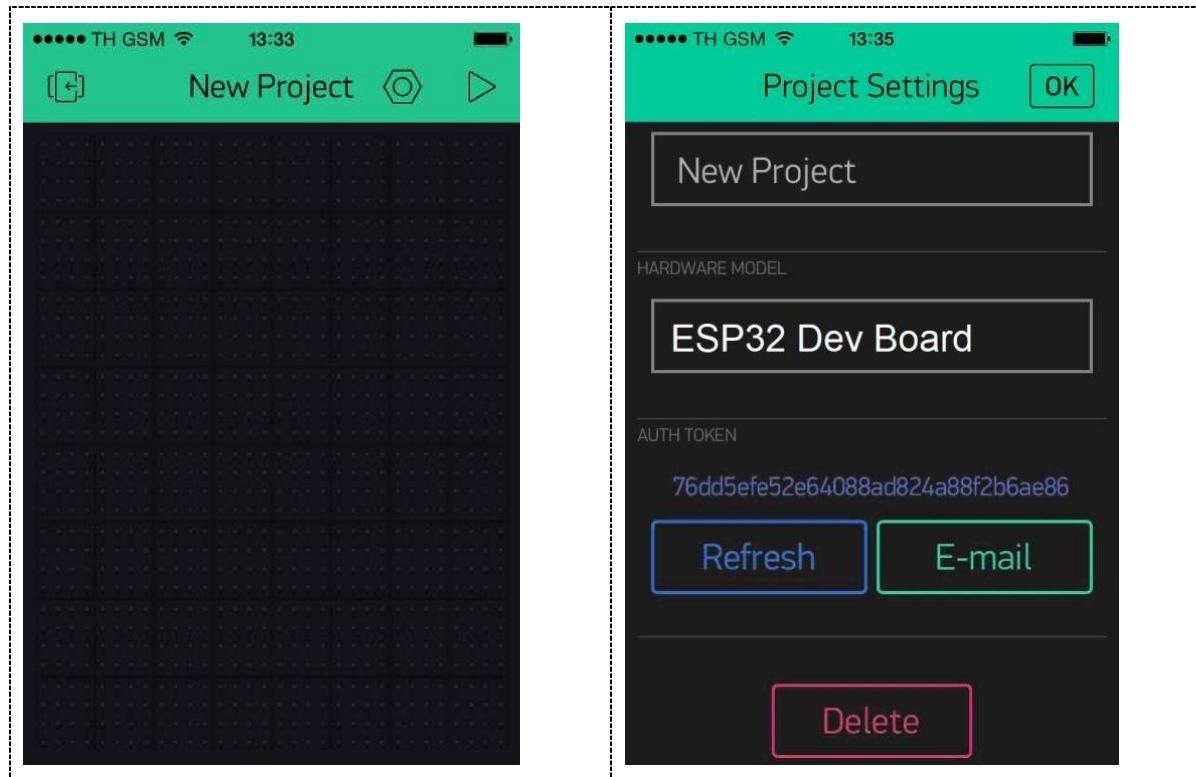
1. ติดตั้ง Blynk Application บนมือถือ
2. ติดตั้ง library Blynk เพื่อใช้งานกับ ESP32 บน Arduino IDE เลือกใช้ Version 0.6.1 หรือติดตั้งจาก <https://github.com/blynkkk/blynk-library/archive/master.zip>



3. เริ่มต้นใช้งาน Blynk เราต้องสมัคร ลงทะเบียน เพื่อใช้งานก่อน ให้เลือกที่คำว่า Create New account เพื่อสร้างการเชื่อมต่อ กับ application กับ Email ของผู้ใช้งาน
4. Login เข้า Blynk
5. เลือก Connect เข้ากับ server ของ Blynk (Blynk ให้ผู้ใช้เลือก Custom Server ได้โดยใส่ IP Address ของ Server เราเอง โดยกดที่รูป Problems Signing In และเลื่อน scroll จาก Blynk ไป Custom)



6. ต่อมาเป็นการสร้าง Project ของเราด้วย Blynk ให้กดที่ลัญลักษณ์หกเหลี่ยมมุมขวาบนเพื่อตั้งค่า
7. โดยในหน้านี้เราจะสามารถตั้งชื่อ Project ของเราและเลือกรูปแบบ Hardware ที่เราจะใช้ได้ Hardware ให้เลือก ESP32 Dev Board



- โดยทุกครั้งที่เริ่มสร้างโปรเจคใหม่ AUTH TOKEN จะถูกเปลี่ยนใหม่เสมอ ซึ่ง KEY นี้เองที่เป็นเสมือนกุญแจสำหรับเขื่อมต่อ โดยที่เราไม่ต้องใช้ user, password เราสามารถกดที่คำว่า "E-mail" เพื่อส่ง KEY นี้เข้าเมลเราได้

Auth Token for New Project project and device New Device คลังจดหมาย X

dispatcher@blynk.io บันทึกการสมัคร
ถึง สัปดาห์

อังกฤษ > ไทย แปลชื่อความ

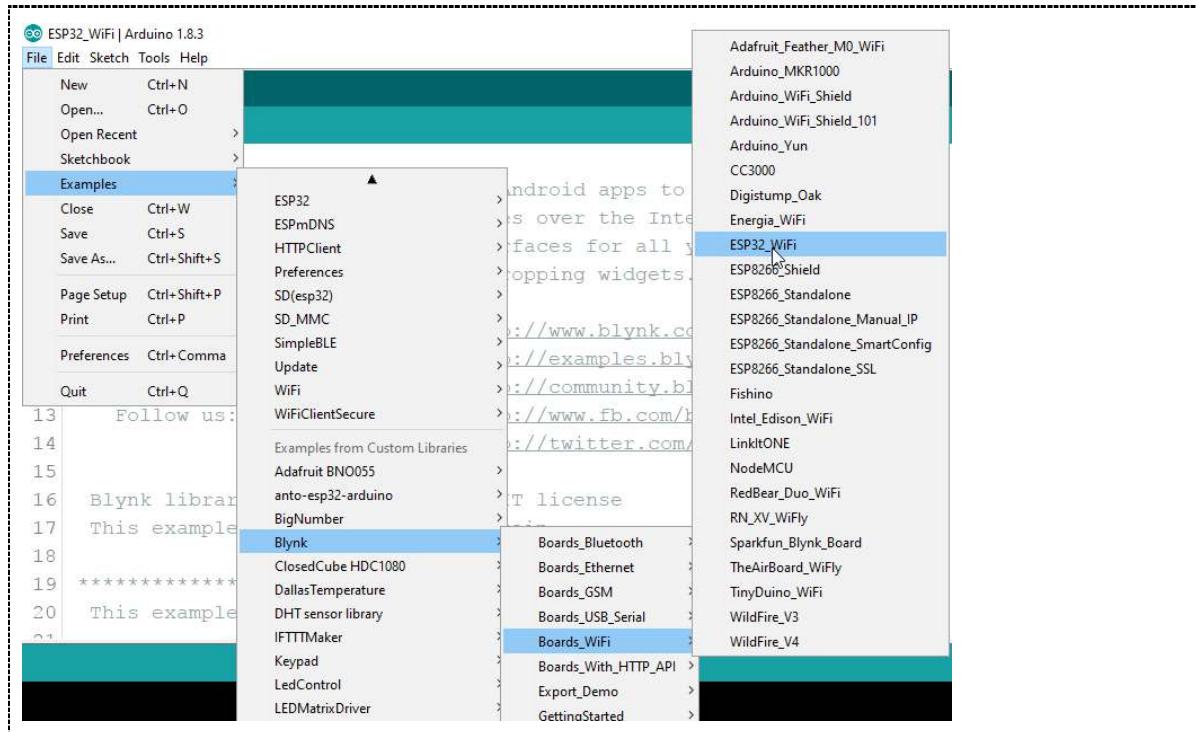
Auth Token : d172e0defc664664afc11def54aa45e4

Happy Blynking!

- Getting Started Guide -> <https://www.blynk.cc/getting-started>
Documentation -> <http://docs.blynk.cc/>
Sketch generator -> <https://examples.blynk.cc/>

9. การใช้งานบน Arduino IDE

หลังจากเราลง Library Blynk เรียบร้อยแล้วจะมี Example ที่ติดมาด้วยจำนวนมาก ซึ่งในบทความนี้จะขอแนะนำตัวอย่างง่ายๆ เริ่มต้นไปกับ File → Example → Blynk → Board_Wifi → ESP32_Wifi



10. ให้โค้ดของเราจะเห็นว่ารูปแบบการสั่งงานใน void loop() คือเพียง Blynk.run(); เพื่อสั่งงานจากภายนอก

<pre>ESP32_WIFI:~</pre> <pre> 1 2 #define BLYNK_PRINT Serial 3 4 #include <WiFi.h> 5 #include <WiFiClient.h> 6 #include <BlynkSimpleEsp32.h> 7 8 char auth[] = "YourAuthToken"; 9 char ssid[] = "YourNetworkName"; 10 char pass[] = "YourPassword"; 11 12 void setup() 13 { Serial.begin(9600); 14 Blynk.begin(auth, ssid, pass); 15 } 16 17 void loop() 18 { Blynk.run(); 19 }</pre>	<pre>#define BLYNK_PRINT Serial</pre> <pre>#include <WiFi.h> #include <WiFiClient.h> #include <BlynkSimpleEsp32.h></pre> <pre>char auth[] = "YourAuthToken"; char ssid[] = "YourNetworkName"; char pass[] = "YourPassword";</pre> <pre>void setup() { Serial.begin(115200); Blynk.begin(auth, ssid, pass); }</pre> <pre>void loop() { Blynk.run(); }</pre>
---	--

และในบรรทัด char auth[] = "YourAuthToken"; ให้ไป copy key ของ Auth Token จากใน Email ที่เราได้รับมาใส่

```

8 char auth[] = "d172e0defc664664afc11def54aa45e4";
9 char ssid[] = "testVirus";
10 char pass[] = "1510031510";
11

```

บรรทัด Blynk.begin(auth,"ssid","pass"); ให้เลือก Wifi ที่เราใช้เชื่อมต่อ กับ app ใน ssid และ password wifi ใน pass

และในกรณีเราเปลี่ยนรูปแบบการเชื่อมต่อ Wifi เป็นแบบ IP Address เราสามารถเขียนโค้ดได้ว่า

```

Blynk.begin(auth, SSID, pass);
to
Blynk.begin(auth, SSID, pass, "your_host");
or to
Blynk.begin(auth, SSID, pass, IPAddress(XXX,XXX,XXX,XXX));

```

11. จากนั้นให้ทำการ Upload ลง board และเปิด Serial Monitor จนกระทั่งมีข้อความขึ้นแบบนี้

```

4 #include <WiFi.h>
5 #include <WiFiClient.h>
6 #include <Blynk>
7
8 char auth[] = "d172e0defc664664afc11def54aa45e4";
9 char ssid[] = "testVirus";
10 char pass[] = "1510031510";
11
12 void setup()
13 {
14     Serial.begin(115200);
15     Blynk.begin(auth, ssid, pass);
16 }
17 void loop()
18 {
19     Blynk.run();
20 }

```

Serial Monitor output:

```

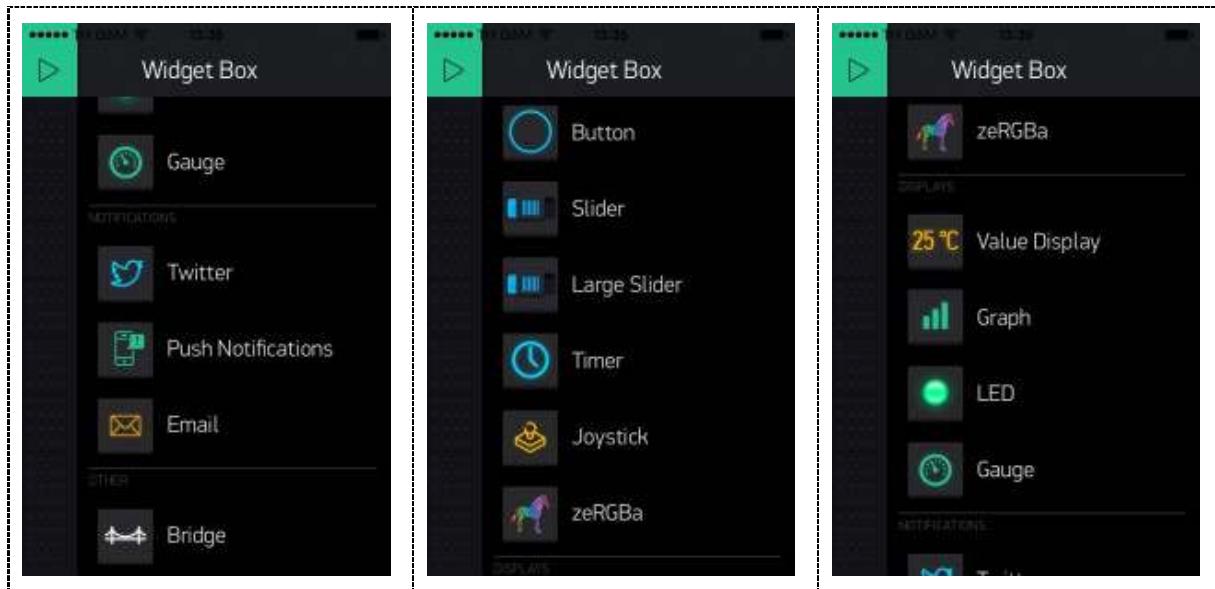
entry 0x400806ac
[33] Connecting to SUT_IoTs
[619] Connected to WiFi
[619] IP: 192.168.43.237
[619]

/v _ ) / / _ \ / _ / / / / \ / ' _ /
/ _ / / \ , / / / / \ \ \
/ _ / v0.6.1 on ESP32

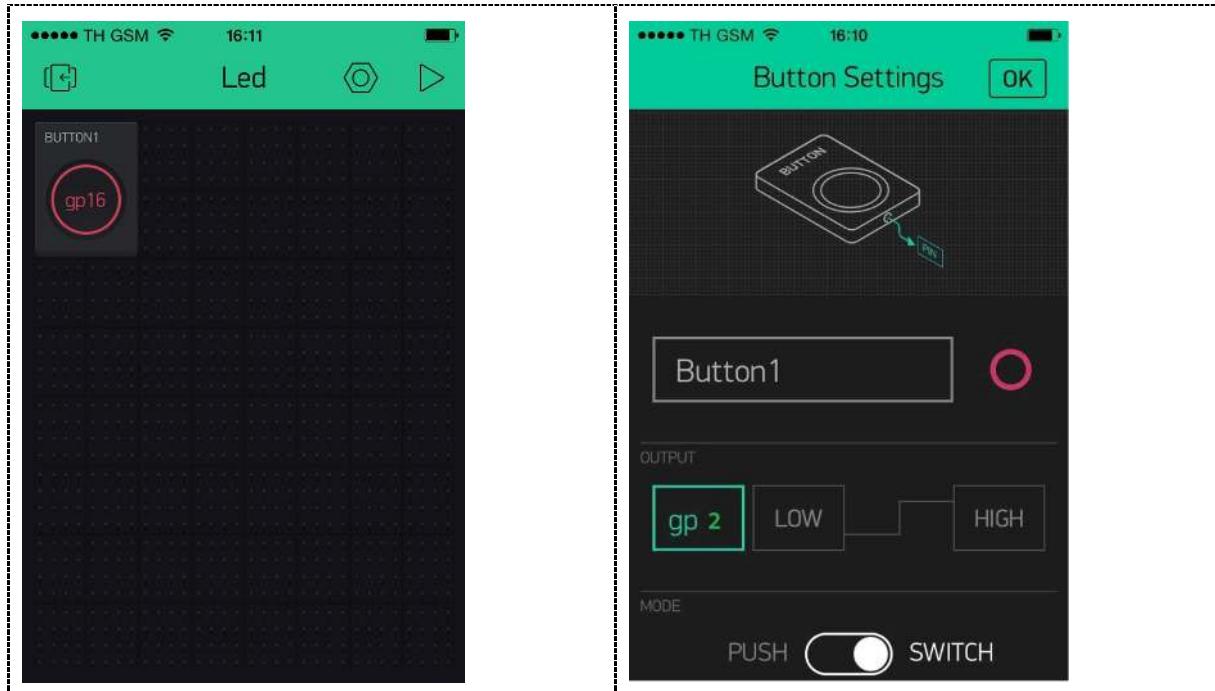
[625] Connecting to blynk-cloud.com:80
[1138] Ready (ping: 204ms).

```

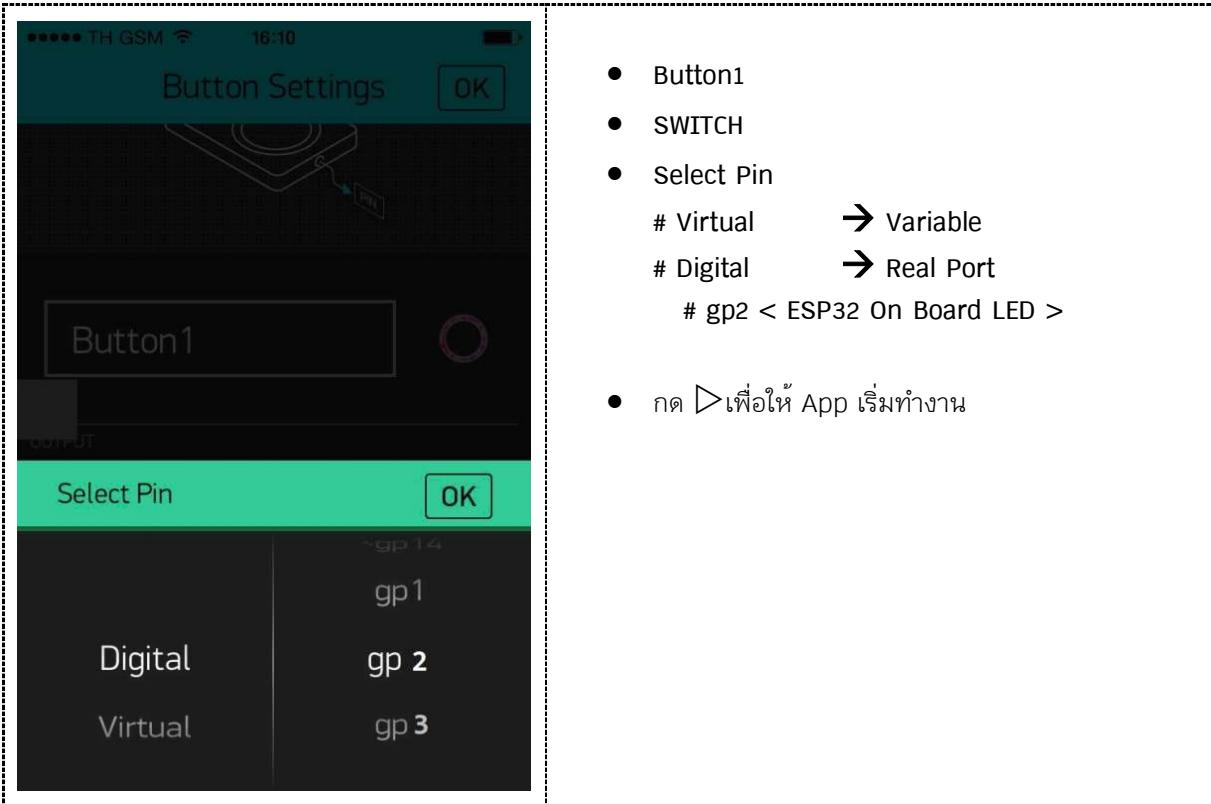
12. สร้างโปรเจคของเราน Blynk โดยการกดที่พื้นที่ว่างเปล่าตรงไหนก็ได้ในหน้า New project จะปรากฏหน้าต่างของ Widget ให้เราเลือกขึ้นมา



13. ในบทความรู้บันี้เราจะลองใช้ดูอะไรที่ง่ายๆ ก่อน ให้ลองเลือก Button widget มาลงบนพื้นที่ว่างเปล่ามา 1 อัน จากนั้นเราจะมาตั้งค่าการใช้งานปุ่ม Button กันโดยกดไปที่รูป Button ที่เราเลือกจะปรากฏหน้าต่างแบบนี้ซึ่งในหน้านี้เราจะสามารถเปลี่ยนชื่อปุ่มได้ และเลือกโหมด output pin ที่ต่อ กับ อุปกรณ์จาก board ของเราได้



14. เลือกรูปแบบ pin จะให้เป็นขา Digital หรือ Virtual ก็ได้ ซึ่งรูปแบบ Virtual จะไม่ใช่การรับค่าจากขาต่างๆ เป็นเหมือนการสร้างตัวแปรมาเก็บค่าอีกที และเลือกขา GPIO ให้ตรงกับ อุปกรณ์ที่เราจะต่อ เมื่อเสร็จแล้วกด ▶ เพื่อให้ App เริ่มทำงาน



15. เมื่อแก้ไขโปรแกรมแล้วจะได้ ดังนี้

```
#define BLYNK_PRINT Serial

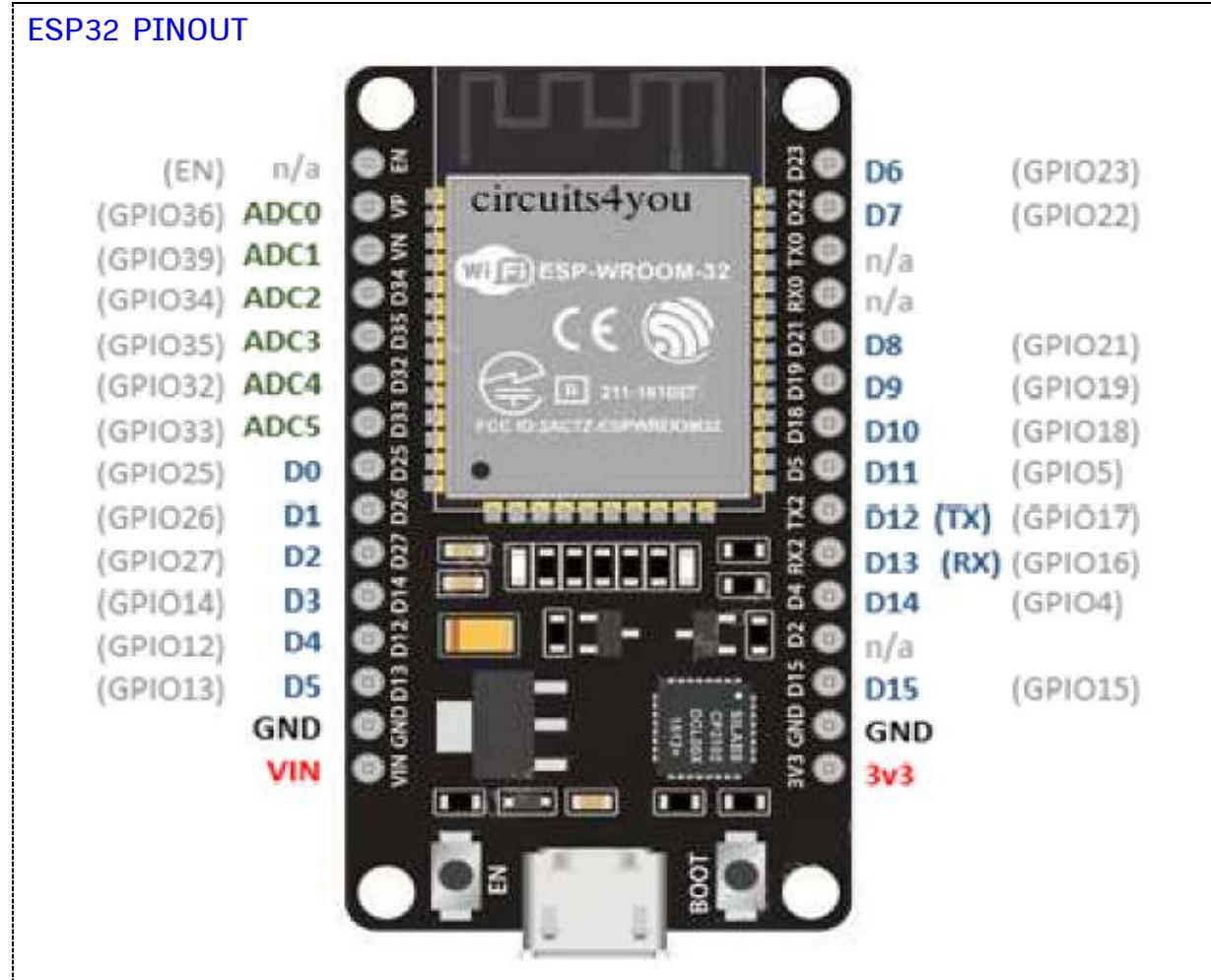
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

char auth[] = "f311e6ae4aa9421cb0f29b531b305b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

void setup()
{
  Serial.begin(115200);
  Blynk.begin(auth, ssid, pass);
}

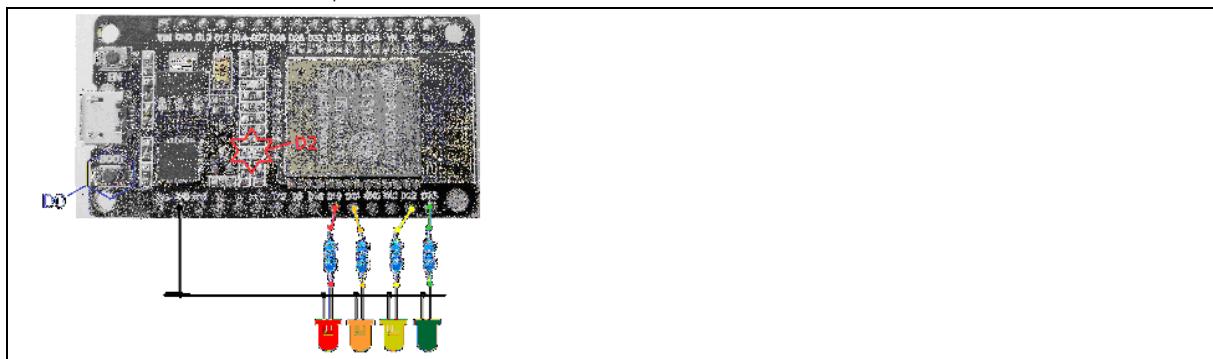
void loop()
{
  Blynk.run();
}
```

ESP32 PINOUT



16. ทดสอบการทำงานโดยการคุม LED GPIO2 หรือ On Board DOIT ESP32 Kit Ver1

17. ให้ทำการทดสอบควบคุม 4 External LED ด้วย Blynk



การทดลองที่ 5.6 – Blynk LED Control and Monitor

19. การทดสอบอ่านสวิตช์ DO และแสดงผลที่ LED Port V5 บน Blynk

- จาก Web <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- จาก Web <https://community.blynk.cc/t/how-to-turn-on-widget-leds/643>
- ทดสอบโปรแกรม ให้แก้ไข (1/3)Auth, (2/3)SSID และ (3/3>Password

```
// ESP 32
#define BLYNK_PRINT Serial

#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>

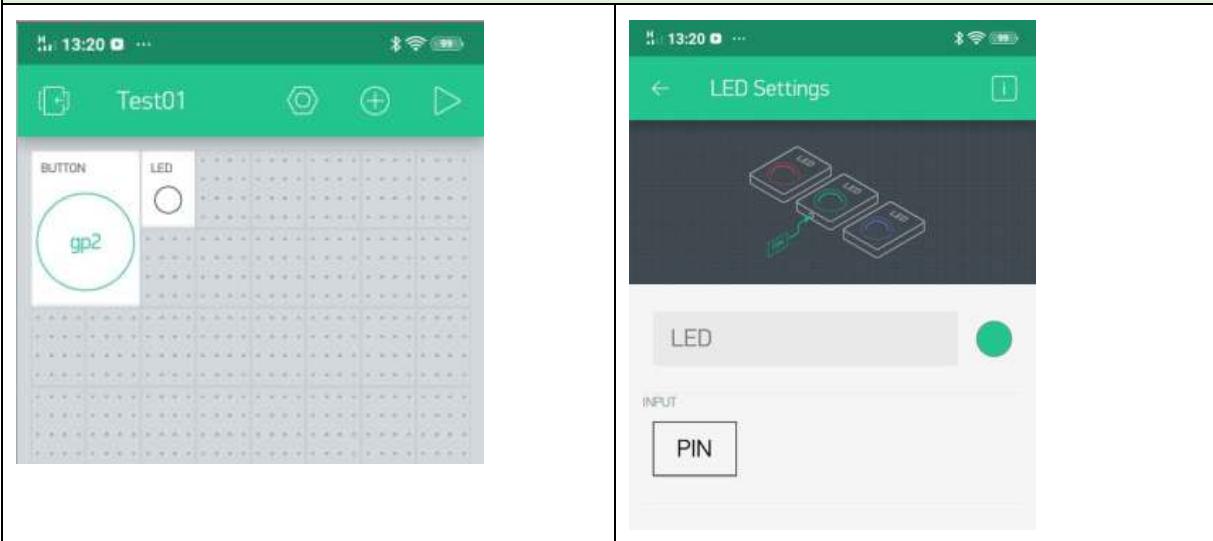
const int btnPin = 0; // DO
boolean btnState = false;
WidgetLED blynk_LED(V5);
BlynkTimer timer; // Announcing the timer

char auth[] = "f311e6ae4aa94210f29b53061b305b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

void myTimerEvent()
{
    boolean isPressed = (digitalRead(btnPin) == LOW);
    if (isPressed != btnState)
    {
        if (isPressed)
            blynk_LED.on();
        else
            blynk_LED.off();
        btnState = isPressed;
        Serial.print(" LED Status = ");
        Serial.println(btnState);
    }
}

void setup()
{
    Serial.begin(115200);
    pinMode(btnPin, INPUT_PULLUP);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(250L, myTimerEvent);
}

void loop()
{
    Blynk.run();
    timer.run(); // running timer every 250ms
}
```



LED Settings

LED

PIN

Select pin

OK

V2

V3

V4

V5

V6

V7

V8

LED Settings

LED

INPUT

V5

Delete

```
/ _ ) / _ 
/ _ / / / / _ \ \ ' _ /
/ _ / / \ _ , / / / / \ \ \
/ _ / v0.6.1 on ESP32
```

[4678] Connecting to blynk-cl
[4892] Ready (ping: 177ms).
LED Status = 1
LED Status = 0
LED Status = 1
LED Status = 0
[37614] Heartbeat timeout
[39617] Connecting to blynk-cl
[42631] Login timeout
[44631] Connecting to blynk-cl
[46263] Ready (ping: 1621ms).

Reset (D0)

EN

Power

D2

100%

LED

การทดลองที่ 5.7 – Blynk LED Control and Sensor Monitor

21. การทดสอบอ่านอุณหภูมิด้วย DHT-22 และแสดงผลที่ Blynk

- จาก Web <http://help.blynk.cc/getting-started-library-auth-token-code-examples/blynk-basics/how-to-display-any-sensor-data-in-blynk-app>
- Install **DHT22 Library** เลือก **DHT Sensor library for ESPx V1.17.0** และทำการติดตั้ง



22. ต่อวงจร DHT-22 เข้าที่ขา D15 และทดสอบการทำงานของโปรแกรม

+ DG

```
#include "DHTesp.h"
DHTesp dht;
const int pinDHT_22 = 15;

void setup()
{ Serial.begin(115200);
  dht.setup(pinDHT_22, DHTesp::DHT22);
}

void loop()
{ float temperature = dht.getTemperature();
  float humidity = dht.getHumidity();
  Serial.print(" Temp('C) >> "); Serial.print(temperature, 1);
  Serial.print(", Humidity(%) >> "); Serial.println(humidity, 1);
  delay(2000);
}
```

23. ทดสอบการทำงานของโปรแกรม ให้แก่ไฟ (1/3)Auth, (2/3)SSID และ (3/3)Password

```
// ESP-32

#define BLYNK_PRINT Serial
#include <WiFi.h>
#include <WiFiClient.h>
#include <BlynkSimpleEsp32.h>
#include "DHTesp.h"

BlynkTimer timer; // Announcing the timer
DHTesp dht;

char auth[] = "f311e6ae4aa9421cb0f29b53061b3b";
char ssid[] = "SUT_IoTs";
char pass[] = "MaiMeeJingJing";

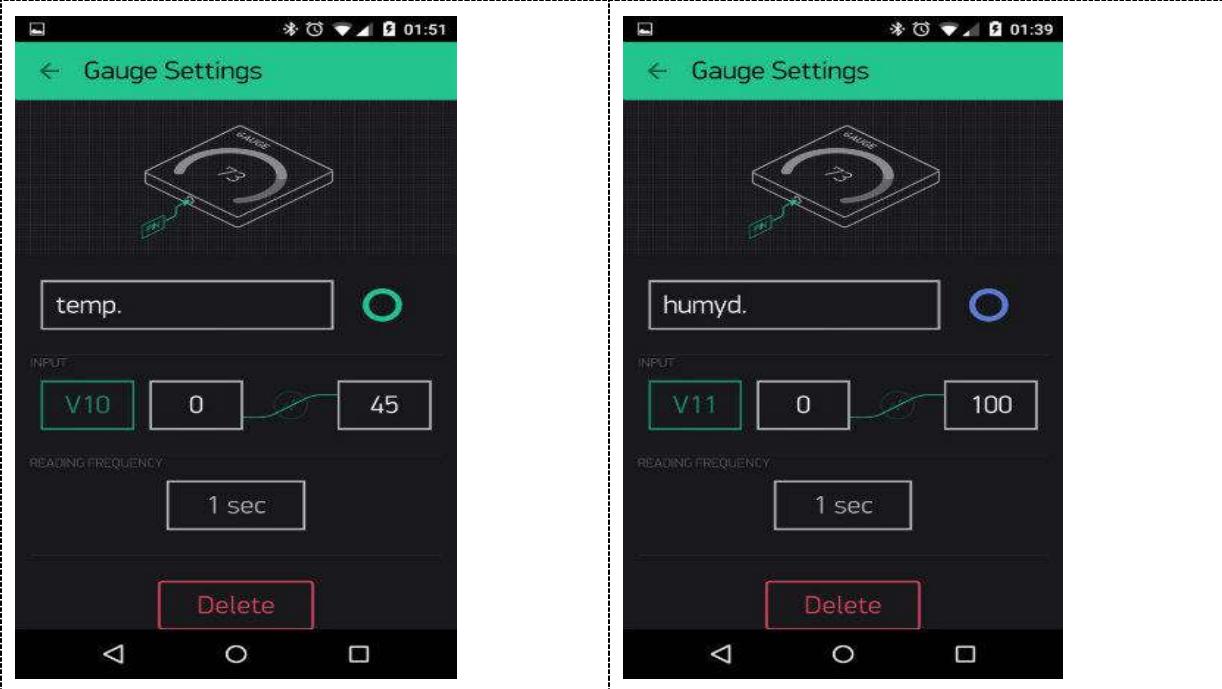
const int pinDHT_22 = 15; // D15
float temperature = 12.34, humidity = 56.78;

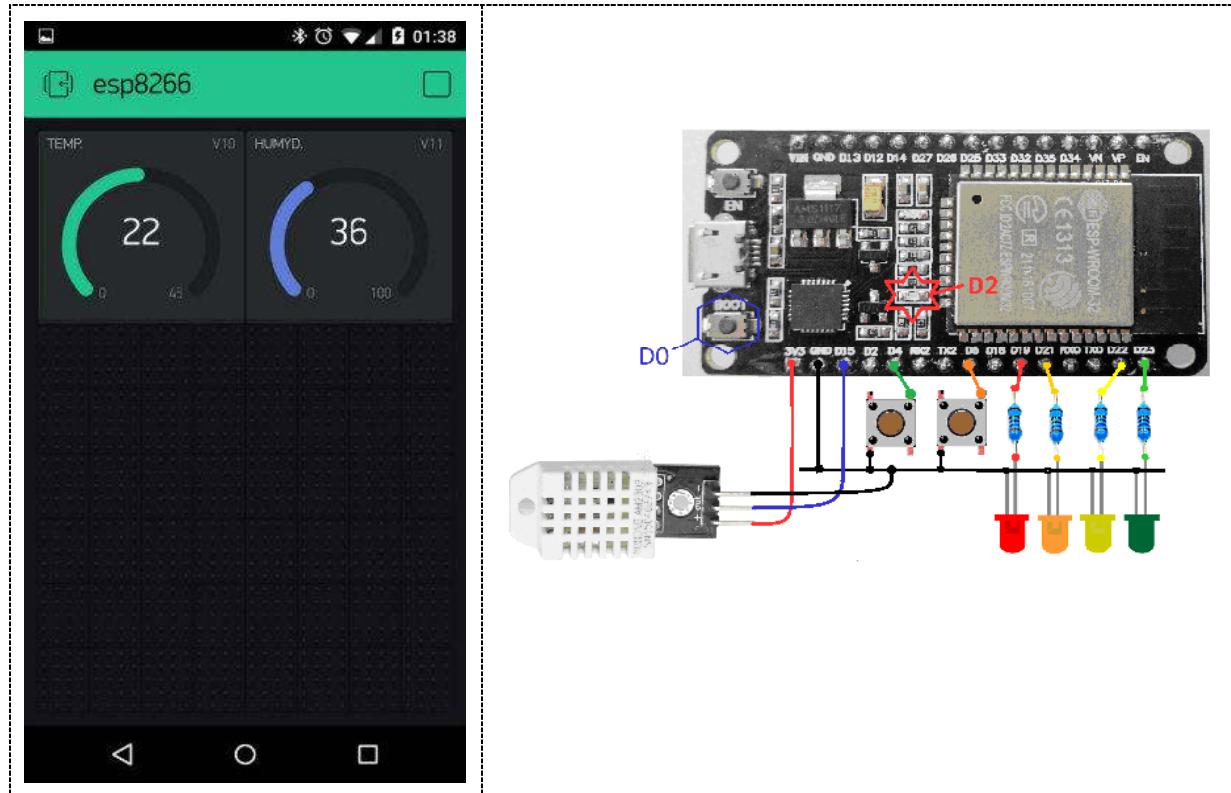
void myTimerEvent()
{
    temperature = dht.getTemperature();
    humidity = dht.getHumidity();
    Blynk.virtualWrite(V10, temperature);
    Blynk.virtualWrite(V11, humidity);
    Serial.print(" Temp('C') >> ");
    Serial.print(temperature, 1);
    Serial.print(", Humidity(%) >> ");
    Serial.println(humidity, 1);
}

void setup()
{
    Serial.begin(115200);
    dht.setup(pinDHT_22, DHTesp::DHT22);
    Blynk.begin(auth, ssid, pass);
    timer.setInterval(1000L, myTimerEvent);
}

void loop()
{
    Blynk.run();
    timer.run(); // running timer every 250ms
}
```

24. ที่ Blynk ให้ใช้ Gauge และ Port V10 และ Port V11





Quiz_104 – Blynk and LINE from (DHT22 + 4 LED + 2 Switch)

ให้ทำการทดสอบควบคุม 4 LED และตรวจสอบค่าจาก DHT22, 2-Switch พร้อมทั้งบันทึก ค่าอุณหภูมิและความชื้อลง GoogleSheet ด้วย Blynk พร้อมทั้งแจ้งเตือนไปยัง LINE Notify

- ควบคุมการปิดเปิด 4 LED
- อ่านค่า DHT-22 และส่งไปยัง Blynk ทุกๆ 5 วินาที
- บันทึกค่าไปยัง Google Sheet
- หากอุณหภูมิเกิน 28°C ให้แจ้งไปยัง LINE
- รับค่าสวิตซ์กำหนด SW1 และ Overheat Alarm, SW2 และ Intruders Alarm ไปยัง LINE
- Read More <https://community.blynk.cc/t/connect-to-google-sheets/42822>

5.6 ESP-32 Direct to LINE Notify

การทดลองที่ 5.8 – ESP-32 Direct to LINE Notify

https://github.com/TridentTD/TridentTD_LineNotify

<https://www.phraphas.com/forum/index.php?topic=356.0>

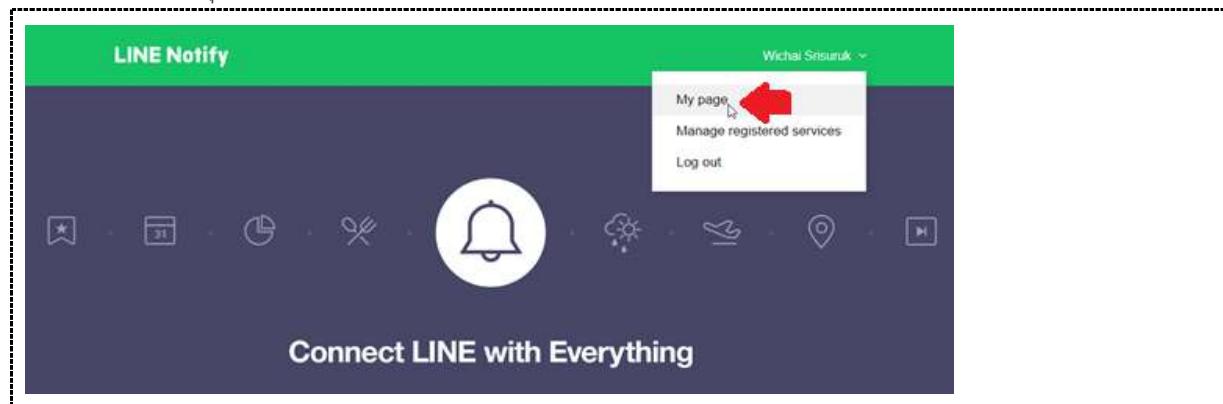
1. Add Library

- Sketch → Inc Lib → Manage
- Filter ด้วยข้อความ LINE notify เพิ่มแลบรารี [TridentTD_LineNotify Ver 3.0.3](#)



2. Create LINE Token

- ทำการ Log in ที่ <https://notify-bot.line.me/en/>
- จากนั้น เข้าไปที่ <https://notify-bot.line.me/my/> เพื่อทำการ สร้าง Line Token ขึ้นมา ให้กดปุ่ม Generate token



Generate access token (For developers)

By using personal access tokens, you can configure notifications without having to add a web service.

[Generate token](#)

Generate Token

[LINE Notify API Document](#)

- ตั้งชื่อ Token อะไรก็ได้ พร้อมทั้งเลือก กลุ่มใน Line ที่ต้องการให้มีการแจ้งเตือนไปยังกลุ่ม

The screenshot shows two side-by-side windows for generating a token.

Left Window: Generate token

- Name:** myTest1234 (highlighted with a red arrow)
- Select a chat to send notifications to:**
 - Search bar: Search by group name
 - List:
 - 1-on-1 chat with LINE Notify (highlighted with a red arrow labeled "LINE to")
 - BLUE CHARA
 - Opt.Chem.
 - The gangster
 - ...
- Note:** Revealing your personal access token can allow a third party to obtain the names of your connected chats as well as your profile name.
- Create** button (highlighted with a red arrow)

Right Window: Your token is:

- Your token is:** 94H0Lqvi4ybwTlb74hIR8v0bzYcZyv81DvDyMc
- Note:** If you leave this page, you will not be able to view your newly generated token again. Please copy the token before leaving this page.
- Copy** button (highlighted with a red arrow)
- Close** button

3. Code for Test

```
#include <TridentTD_LineNotify.h>

#define SSID          "Test1234"
#define PASSWORD      "MaiMeeJingJing"
#define LINE_TOKEN    "xxxxxxxxxxxxxxxxxxxxxx"

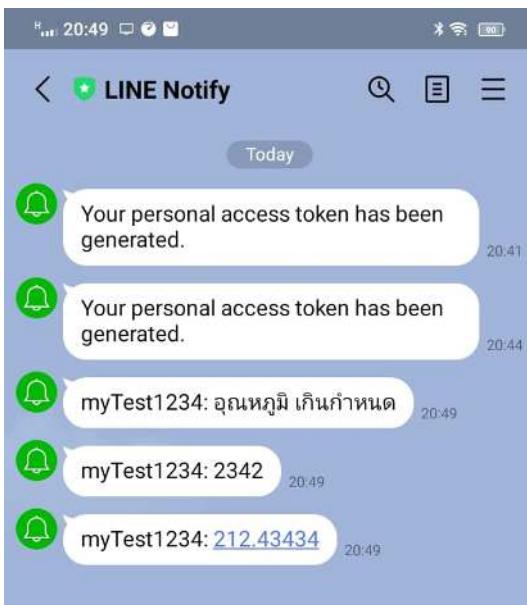
void setup() {
  Serial.begin(115200); Serial.println();
  Serial.println(LINE.getVersion());

  WiFi.begin(SSID, PASSWORD);
  Serial.printf("WiFi connecting to %s\n", SSID);
  while(WiFi.status() != WL_CONNECTED) { Serial.print("."); delay(400); }
  Serial.printf("\nWiFi connected\nIP : ");
  Serial.println(WiFi.localIP());

  LINE.setToken(LINE_TOKEN);

  LINE.notify("อุณหภูมิ เกินกำหนด");
  LINE.notify(2342); // จำนวนเต็ม
  LINE.notify(212.43434,5); // จำนวนจริง และต้อง 5 หลัก

  void loop() {
    delay(1);
  }
}
```



5.7 Pushing Data to Google Sheet

<https://community.blynk.cc/t/pushing-data-to-google-docs/7580>

<https://www.youtube.com/watch?v=3Bv1CbPyDPs>

การทดลองที่ 5.9 – ESP-32 Direct to Google Sheet

1/3 Setting up the Google Sheet as server.

1. Log in to your Google account and open a blank Google Spread Sheet.
2. Change the document name and the sheet name as preferred and create the required columns In this example:
 - Document name: ESP32_Data_log
 - Sheet name: Sensor_Data
 - Columns: Timestamp, Sensor1, Sensor2

3. Tools -> Script editor. This will open up a new window to write the script. Copy and paste the code in the [GScriptCode.txt](#) file to the Google Script editor. Save the script with a preferred name. (Example: ESP32_Data_Log_Script).

- Click [here](#) to find more information regarding Google Script.

Apps Script **ESP32_Data_Log_Script**

ไฟล์ **Code.gs** **ไฟล์ร่าง** **บริการ**

```

function doGet(e){
  // open the spreadsheet
  var ss = SpreadsheetApp.getActive();

  // use the 'id' parameter to differentiate between sheets
  var sheet = ss.getSheetByName(e.parameter["id"]);

  // extract headers
  // getRange accepts row, col, number_of_rows and num_of_cols as argument
  // getLastColumn returns the position of the last column that has content
  var headers = sheet.getRange(1, 1, 1, sheet.getLastColumn()).getValues()[0];

  // store the position of the last row
  var lastRow = sheet.getLastRow();

  var cell = sheet.getRange('a1');
  var col = 0;
  var d = new Date();

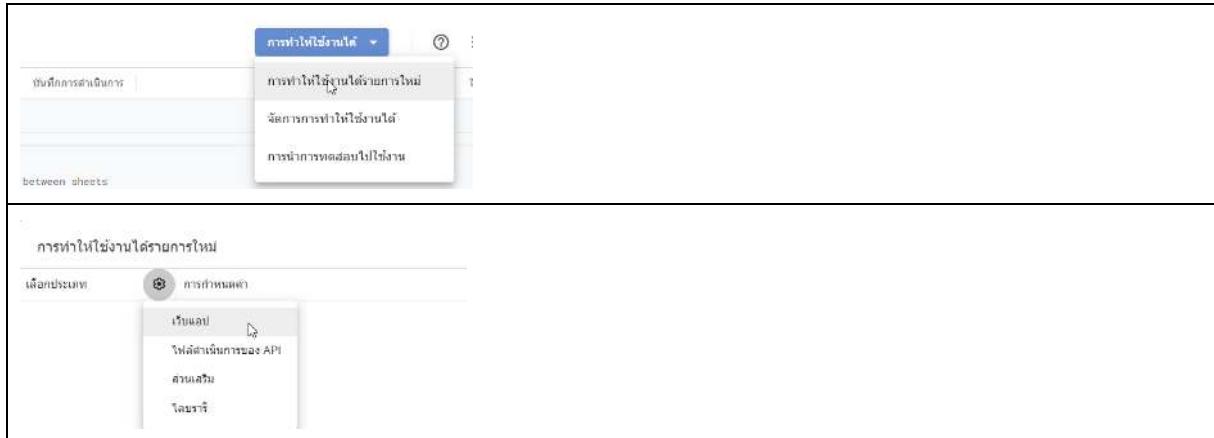
  for (i in headers){
    // loop through the headers and if a parameter name matches the header name insert the value
    if (headers[i] == "Timestamp"){
      val = d.toDateString() + ", " + d.toLocaleTimeString();
    }
    else{
      val = e.parameter[headers[i]];
    }

    // append data to the last row
    cell.offset(lastRow, col).setValue(val);
    col++;
  }

  return ContentService.createTextOutput('success');
}

```

4. Go to Publish -> Deploy as web app.



<p>การทำให้ใช้งานได้รายการใหม่</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;">เลือกมีเดีย</td> <td style="width: 15%; text-align: center;">@</td> <td style="width: 70%;">การกำหนดค่า</td> <td style="width: 10%; text-align: right; vertical-align: bottom;">?</td> </tr> <tr> <td>รีบเมลล์</td> <td></td> <td>รายละเอียด</td> <td></td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">ตรวจสอบรายการใหม่</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">ผู้รับเมลล์</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">อีเมลล์ อีเมลล์ที่ต้องการรับ อีเมลล์ (pk007.sut@gmail.com)</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">ชื่อบรรษัทและชื่อผู้รับเมลล์ที่ต้องการรับ อีเมลล์ (บริษัทจราจร)</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">ผู้รับเมลล์ชื่อ รับจากนี้</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;">ส่วนของข้อความที่ต้องการให้เมลล์ติดตาม ดูข้อมูลเพิ่มเติม</td> </tr> <tr> <td colspan="4" style="text-align: center; padding: 5px;"> <input type="button" value="ยกเลิก"/> <input type="button" value="การดำเนินการต่อ"/> </td> </tr> </table>	เลือกมีเดีย	@	การกำหนดค่า	?	รีบเมลล์		รายละเอียด		ตรวจสอบรายการใหม่				ผู้รับเมลล์				อีเมลล์ อีเมลล์ที่ต้องการรับ อีเมลล์ (pk007.sut@gmail.com)				ชื่อบรรษัทและชื่อผู้รับเมลล์ที่ต้องการรับ อีเมลล์ (บริษัทจราจร)				ผู้รับเมลล์ชื่อ รับจากนี้				ส่วนของข้อความที่ต้องการให้เมลล์ติดตาม ดูข้อมูลเพิ่มเติม				<input type="button" value="ยกเลิก"/> <input type="button" value="การดำเนินการต่อ"/>			
เลือกมีเดีย	@	การกำหนดค่า	?																																	
รีบเมลล์		รายละเอียด																																		
ตรวจสอบรายการใหม่																																				
ผู้รับเมลล์																																				
อีเมลล์ อีเมลล์ที่ต้องการรับ อีเมลล์ (pk007.sut@gmail.com)																																				
ชื่อบรรษัทและชื่อผู้รับเมลล์ที่ต้องการรับ อีเมลล์ (บริษัทจราจร)																																				
ผู้รับเมลล์ชื่อ รับจากนี้																																				
ส่วนของข้อความที่ต้องการให้เมลล์ติดตาม ดูข้อมูลเพิ่มเติม																																				
<input type="button" value="ยกเลิก"/> <input type="button" value="การดำเนินการต่อ"/>																																				
<p>การทำให้ใช้งานได้รายการใหม่</p> <p>รีบແນປຕ່ອງການສິ່ງເຂົ້າເຊົ່າງຂ່ອງມູນຂອງຄຸນ</p> <p>ໃຫ້ສິ່ງເຂົ້າເຊົ່າງ</p>																																				
<p>ເລືອກນັ້ນຂຶ້ນ</p> <p>ເພື່ອໄປຢັ້ງ ESP32_Data_Log_Script</p> <table border="1" style="width: 100%; border-collapse: collapse;"> <tr> <td style="width: 15%;"></td> <td style="width: 15%; text-align: center;">Wichai Srisuruk</td> <td style="width: 70%; text-align: right;">pk007.sut@gmail.com</td> </tr> <tr> <td colspan="3" style="text-align: center; padding: 5px;">@ ໃຫ້ນັ້ນຂຶ້ນອື່ນ</td> </tr> </table>		Wichai Srisuruk	pk007.sut@gmail.com	@ ໃຫ້ນັ້ນຂຶ້ນອື່ນ																																
	Wichai Srisuruk	pk007.sut@gmail.com																																		
@ ໃຫ້ນັ້ນຂຶ້ນອື່ນ																																				
<p></p> <p>Google ຍັງໄນ້ໄດ້ຍືນຍັນແລ້ວ</p> <p>ແລ້ວກ່າລັງຂອບເຂົ້າເຊົ່າງຂ່ອງມູນທີ່ລະເຮືອດອນໃນນັ້ນຂຶ້ນ Google ຂອງຄຸນ ຄຸນໄນ້ຄວຮໃຫ້ແນປຕ່ອງເຈັບກວ້ານັກພັດນາແລ້ວ ປັດນາແລ້ວ (pk007.sut@gmail.com) ຈະຍືນຍັນແລ້ວນີ້ກັບ Google</p> <p>ຍືນຍັນສູງ ກລັບຜູ້ຄວາມປົດກັນ</p> <p>ຖາເນີນການຕ່ອເຈພະເນື້ອຄຸນເຂົ້າໃຈຄວາມເສື່ອແລະເຫື່ອເກີ້ວັກພັດນາແລ້ວ (pk007.sut@gmail.com) ເທົ່ານັ້ນ</p> <p>ໄປທີ ESP32_Data_Log_Script (ໄປໂປໂອັນດູນ)</p>																																				

Google ลงชื่อเข้าใช้ด้วย Google

ESP32_Data_Log_Script ต้องการ
เข้าถึงบัญชี Google ของคุณ

pk007.sut@gmail.com

ชื่นจะอนญาตให้ [ESP32_Data_Log_Script](#) ดำเนินการ
ต่อไปนี้

● ดู แก้ไข สร้าง และลบสิ่งที่คุณสนใจในบัญชี Google ของคุณ

ตรวจสอบว่าคุณเชื่อถือ [ESP32_Data_Log_Script](#) ได้
คุณอาจกำลังแบ่งปันข้อมูลที่มีความละเอียดลึกซึ้งกับเว็บไซต์หรือ
แอปนี้ คุณเดาหรือยกเลิกสิทธิ์เข้าถึงได้ทุกเมื่อในบัญชี Google

Learn how Google helps you [share data safely](#).

ดูว่าที่ [ESP32_Data_Log_Script](#) จะจัดการข้อมูลของคุณโดย
อ่านข้อกำหนดในการให้บริการและนโยบายความเป็นส่วนตัว
ของแอป

ยกเลิก [อนุญาต](#)

การทำให้ใช้งานได้รายการใหม่

บันเด็ดการทำให้ใช้งานได้เรียบร้อยแล้ว

เวลาชั้น 1 วันที่ 6 ก.ค. เวลา 21:41

รหัสกราฟทำให้ใช้งานได้

AKfycbzqLhrYKsjIBWLrVdunHKmy9k_u6vmC_OwmEPp9LCPUwncDUuMPUup1dINqx7GX7J7OdQ

ตัดลอก

รีบบล็อก

URL

https://script.google.com/macros/s/AKfycbzqLhrYKsjIBWLrVdunHKmy9k_u6vmC_OwmEPp9LCPUwncDUuMPUup1dINqx7GX7J7OdQ

ตัดลอก

[เลือกอีก](#)

AKfycbzqLhrYKsjIBWLrVdunHKmy9k_u6vmC_OwmEPp9LCPUwncDUuMPUup1dINqx7GX7J7OdQ

https://script.google.com/macros/s/AKfycbzqLhrYKsjIBWLrVdunHKmy9k_u6vmC_OwmEPp9LCPUwncDUuMPUup1dINqx7GX7J7OdQ/exec

5. Copy and save the URL and click OK .

6. Your URL will look something like this:

<https://script.google.com/macros/s/{yourkey}/exec>

2/3 Test Google Sheet via Web browser

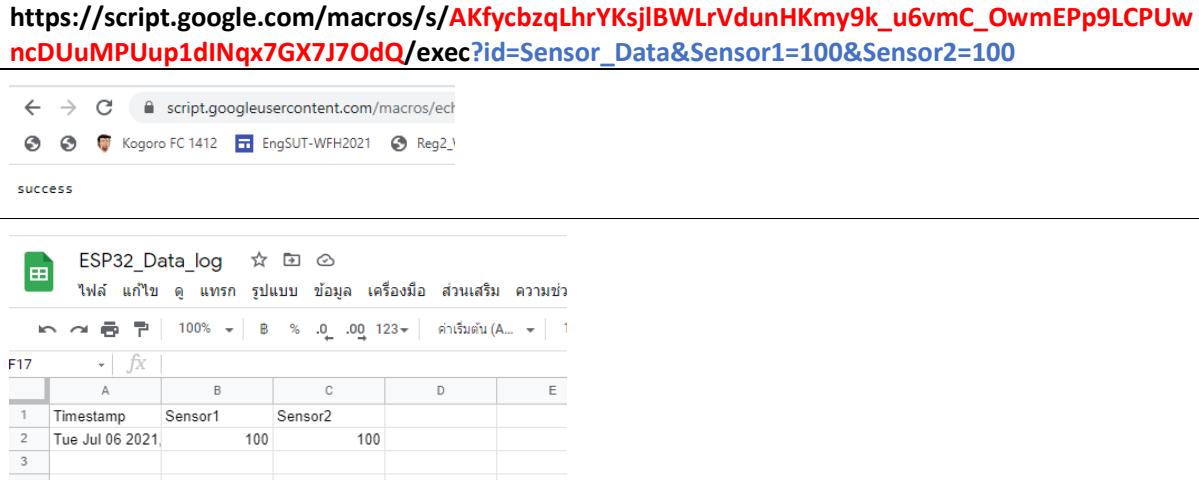
7. To test the server open a browser and enter the URL below.

```
https://script.google.com/macros/s/{your-key}/exec?id=Sensor_Data&Sensor1=100&Sensor2=100
```

If everything is correct, the word 'success' will be shown on your web page.

Now if you go back to your google sheet you will be able to see your data (sensor1 = 100, sensor2 = 100) been logged.

https://script.google.com/macros/s/AKfycbzqLhrYKsjLBWLrVdunHKmy9k_u6vmC_OwmEPp9LCPUwncDUuMPUup1dINqx7GX7J7OdQ/exec?id=Sensor_Data&Sensor1=100&Sensor2=100



	Timestamp	Sensor1	Sensor2
1	Tue Jul 06 2021,	100	100
2			
3			

Congratulations you have successfully setup your Google sheet as the server!

2/3 Test Google Sheet via ESP-32

8. Arduino Code Test

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

String t;
const char* host = "script.google.com";
const int httpsPort = 443;

const char* ssid = "Mue.009MbX";
const char* pass = "1212312121";
String GAS_ID = "AKfycbzqLhrYKsjLBWLrVdunHKmy9k_u6vmC_OwmEPp9LCPUwncDUuMPUup1dINqx7GX7J7OdQ";
String GAS_Sheet = "Sensor_Data";

WiFiClientSecure client;

long now = millis();
long lastMeasure = 0;

void setup() {
  Serial.begin(115200); delay(500);
  WiFi.begin(ssid, pass);
  Serial.print("\nConnecting");
  while (WiFi.status() != WL_CONNECTED) {
    Serial.print(".");
  }
  Serial.print("\nSuccessfully connected to : "); Serial.println(ssid);
  Serial.print("\nIP address: "); Serial.println(WiFi.localIP());
  Serial.println();
  client.setInsecure();
}

void loop() {
  now = millis();
  if (now - lastMeasure > 5000) {
    lastMeasure = now;
    float ValueSensor1 = random(2000, 4000) / 100.0;
    float ValueSensor2 = random(6000, 8000) / 100.0;
    Serial.println();
  }
}
```

```

Serial.print("\nSensor1 = "); Serial.print(ValueSensor1, 1);
Serial.print("\tSensor2 = "); Serial.print(ValueSensor2, 1);
Serial.println();
sendData(ValueSensor1, ValueSensor2);
}

void sendData(float SValue1, float SValue2) {
Serial.println("=====");
Serial.print("connecting to "); Serial.println(host);

//---- Connect to Google host
if (!client.connect(host, httpsPort)) {
Serial.println("connection failed");
return;
}

//---- Post Data
String url;
url += "/macros/s/" + GAS_ID + "/exec?";
url += "id=" + String(GAS_Sheet);
url += "&Sensor1=" + String(SValue1,2);
url += "&Sensor2=" + String(SValue2,2);
Serial.print("requesting URL: "); Serial.println(url);

client.print(String("GET ") + url + " HTTP/1.1\r\n" +
"Host: " + host + "\r\n" +
"User-Agent: BuildFailureDetectorESP8266\r\n" +
"Connection: close\r\n\r\n");

Serial.println("request sent");

//---- Wait Echo
while (client.connected()) {
String line = client.readStringUntil('\n');
if (line == "\r") {
Serial.println("headers received");
break;
}
}
String line = client.readStringUntil('\n');
if (line.startsWith("{\"state\":\"success\"")) {
Serial.println("ESP-32/Arduin CI successfull!");
} else {
Serial.println("ESP-32/Arduin CI has failed");
}

Serial.print("reply was : ");
Serial.println(line);
Serial.println("closing connection");
Serial.println("=====");
Serial.println();
}
}

```

<https://techtutorialsx.com/2017/11/18/esp32-arduino-https-get-request/>
<http://www.iotsharing.com/2017/08/how-to-use-https-in-arduino-esp32.html>
https://www.tutorialspoint.com/esp32_for_iot/esp32_for_iot_transmitting_data_over_wifi_using_https.htm

<https://www.youtube.com/watch?v=bFskgdfvFEc>

https://script.google.com/macros/s/AKfycbzqLhrYKsjLBWLrVdunHKmy9k_u6vmc_OwmEPp9LCPUwnncDUuMPUup1dINqx7GX7J70dQ/exec?id=Sensor_Data&Sensor1=23.37&Sensor2=61.23

https://script.google.com/macros/s/AKfycbwLfmIdx60P4kCHaK15pmSLyjw3Wv6B6mmJvPZzaIqR7uX6EbFCFGNOI-hKGwuIWxDz6g/exec?id=Sensor_Data&Sensor1=23.37&Sensor2=61.23

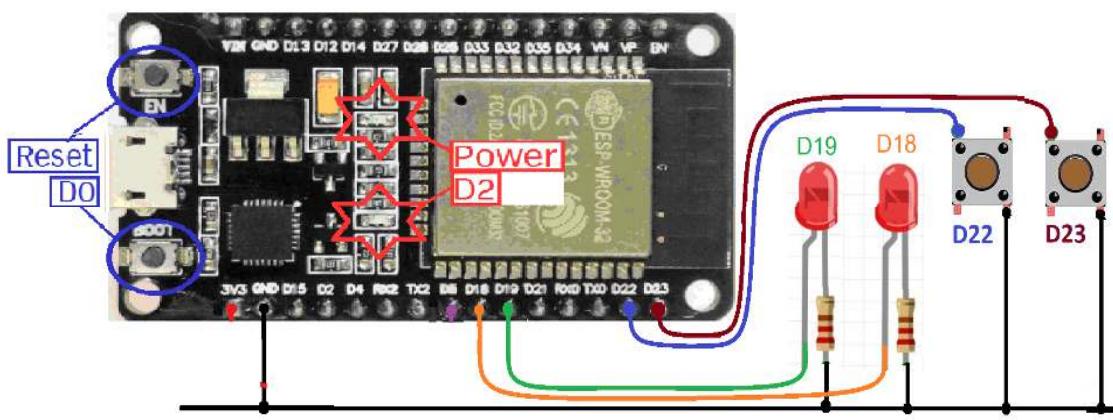
การควบคุมเครื่องจักรอัจฉริยะโดยใช้การสื่อสารระหว่างเครื่องจักรกับเครื่องจักร
M2M - Intelligence Machine Control

ข้อ-สกุล :

6/6 -- คำถ้ามห้ายบทเพื่อทดสอบความเข้าใจ

Quiz_101 – กดติด กดดับ 2 ชุด

- หากต้องการให้ใช้ 1 สวิตซ์ ควบคุม 1 LED แบบกดติด-กดดับ จำนวน 2 ดวงจะต้องจะและเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}



โปรแกรมที่ใช้ทดสอบ

รูปการทดสอบ 1

รูปการทดสอบ 2

Quiz_102 – Web Control 4 LED and Monitor Humid/Temperature

- เพิ่มเติมจาก Q202 อยากได้ปุ่มสำหรับคุมปิด-เปิด หลอดไฟ LED 4 ดวง
- อยากรีกด Link ไปที่หน้า FB ของตัวเอง
- https://www.colorhexa.com/008cba?fbclid=IwAR3dIZ_gRgDWmREmnzuknLbMxV3pOHy4YIPuLEz8-ZzTOX2VhWxcH2QjLGk

← → ⌂ Not secure | 192.168.43.237

The ESP-32 Update web page without refresh

[LED1 ON](#) [LED2 ON](#) [LED3 ON](#) [LED4 ON](#)

[LED1 OFF](#) [LED2 OFF](#) [LED3 OFF](#) [LED4 OFF](#)

State of [LED1, LED2, LED3, LED4] is >> ON, OFF, OFF, ON

DHT-22 sensor : Temp = 28.10 C, Humidity = 43.90 %

[By Wichai Srisuruk](#)

โปรแกรมที่ใช้ทดสอบ

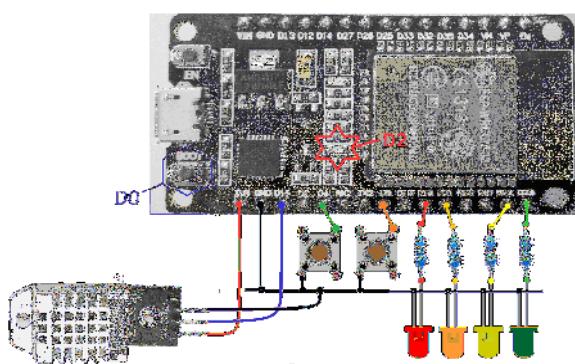
รูปถ่ายหน้า Web Browser

รูปการทดสอบ 1

รูปการทดสอบ 2

Quiz_103 – Pub/Sub Data from (DHT22 + 4 LED + 2 Switch)

- อ่านค่า DHT-22 และส่งไปยัง MQTT Broker ทุกๆ 5 วินาที
 - กำหนดให้ใช้ mqtt.eclipse.org เป็น Broker
 - ควบคุมการปิดเปิด 4 LED
 - รับค่าสวิตซ์กำหนด SW1 แจ้ง Overheat Alarm, SW2 แจ้ง Intruders Alarm



โปรแกรมที่ใช้ทดสอบ

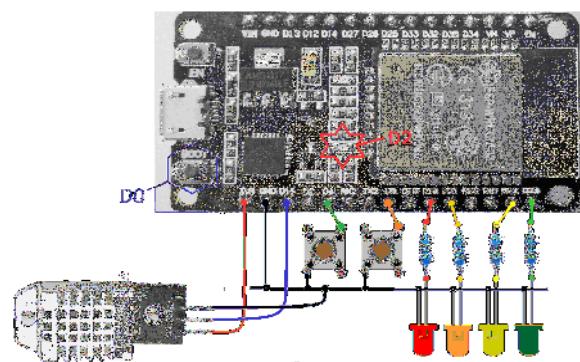
รูปหน้าจอ MQTT Lens

ຮຽນຮັບຮັດການ – 1

ຮູບກາຣຕ້ອງຈາກ - 2

Quiz_104 – Blynk and LINE from (DHT22 + 4 LED + 2 Switch)

- ควบคุมการเปิดปิด 4 LED
- อ่านค่า DHT-22 และส่งไปยัง Blynk ทุกๆ 5 วินาที
- บันทึกค่าไปยัง Google Sheet
- หากอุณหภูมิเกิน 28°C ให้แจ้งไปยัง LINE
- รับค่าสวิตช์กำหนด SW1 และ Overheat Alarm, SW2 แจ้ง Intruders Alarm ไปยัง LINE



โปรแกรมที่ใช้ทดสอบ

รูปหน้าจอ Blynk

รูปหน้าจอ LINE

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2