

แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต

IoT Approaches to Manufacturing System

1/4 - Arduino และ Arduino IDE


- การติดตั้ง Arduino IDE และการเพิ่ม ESP32 Board
- การโปรแกรมเพื่อควบคุม ESP-32 สั่งงาน อินพุต/เอาต์พุต
- การติดตั้งส่วนเสริมเพื่อเรียกใช้งาน สำหรับบอร์ดต่อขยาย อินพุต/เอาต์พุต
- คำถามท้ายบทเพื่อทดสอบความเข้าใจ

1/4. การติดตั้งโปรแกรม Arduino IDE และการเพิ่ม ESP32 Board

Arduino จะใช้โปรแกรมที่เรียกว่า Arduino IDE ในการเขียนโปรแกรม และคอมไพล์ลงบอร์ด โดยขนาดของโปรแกรม Arduino โดยปกติแล้วจะใหญ่กว่าโค้ด AVR ปกติเนื่องจากโค้ด AVR เป็นการเข้าถึงจากรีจิสเตอร์โดยตรง แต่โค้ด Arduino เข้าถึงผ่านฟังก์ชัน เพื่อให้สามารถเขียนโค้ดได้ง่ายมากกว่าการเขียนโค้ดแบบ AVR

1.1 การดาวน์โหลดโปรแกรม Arduino IDE

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




Arduino IDE 1.8.15

The open-source Arduino Software (IDE) makes it easy to write code and upload it to the board. This software can be used with any Arduino board.

DOWNLOAD OPTIONS

Windows Win 7 and newer

Windows ZIP file

Windows app Win 8.1 or 10 

Linux 32 bits

Linux 64 bits

จากนั้นจึงแสดงหน้าเชิญให้ร่วมบริจาค หากไม่ต้องการบริจาคสามารถคลิกปุ่ม JUST DOWNLOAD เพื่อเริ่มดาวน์โหลดโปรแกรมได้เลย

Support the Arduino IDE

Since the release 1.x release in March 2015, the Arduino IDE has been downloaded **52,893,985** times — impressive! Help its development with a donation.

\$3

\$5

\$10

\$25

\$50

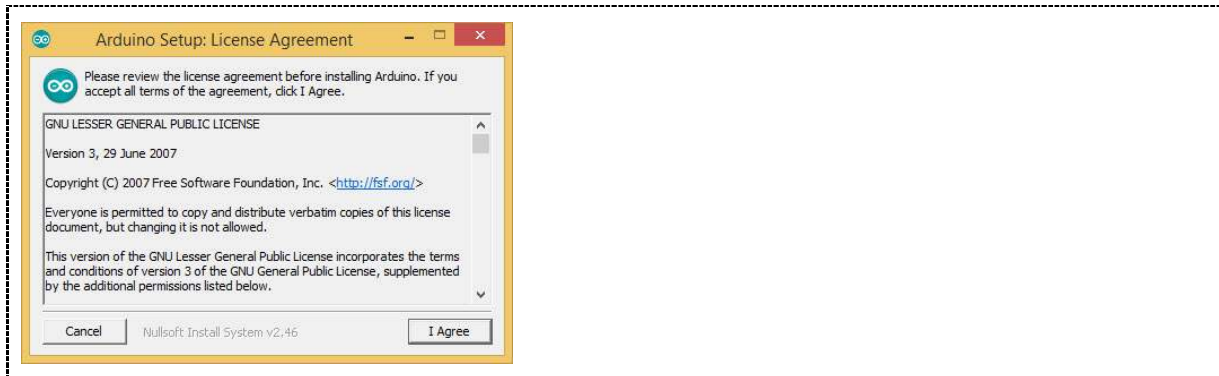
Other

JUST DOWNLOAD

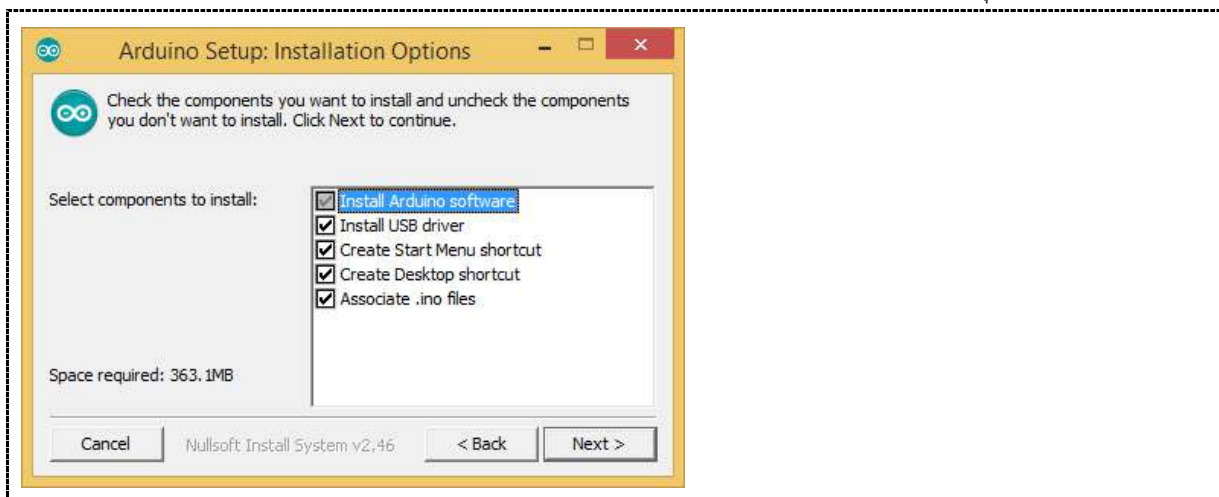
CONTRIBUTE & DOWNLOAD

1.2 การติดตั้งโปรแกรม Arduino IDE

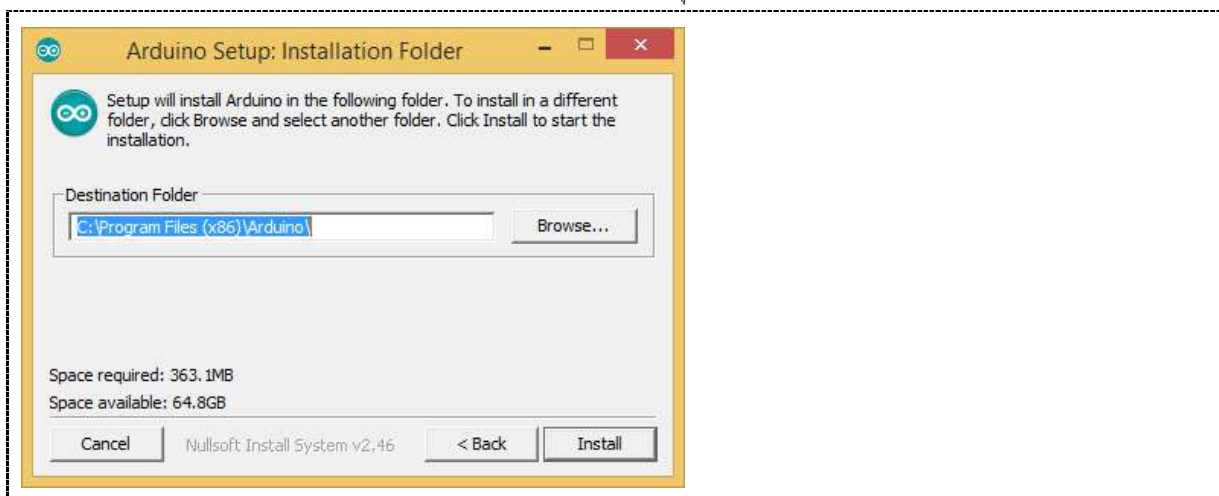
เมื่อดาวน์โหลดเสร็จแล้วให้เปิดไฟล์ติดตั้งขึ้นมาได้เลย กดปุ่ม I Agree ได้เลย



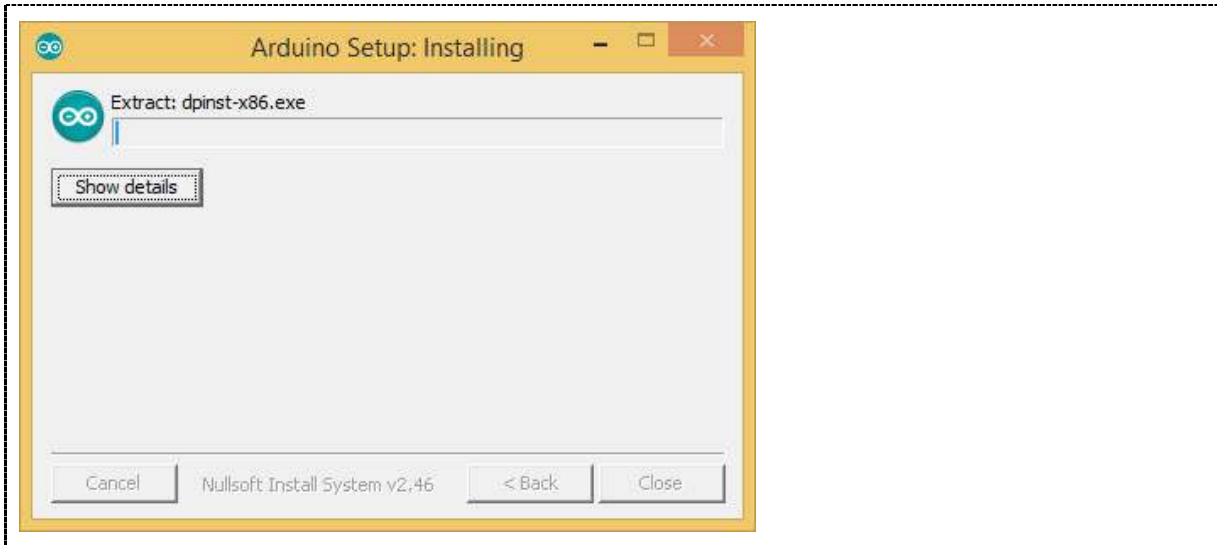
มีตัวเลือกให้เลือกติดตั้ง แนะนำให้เลือกทั้งหมด (ค่าเริ่มต้นคือเลือกทั้งหมด) แล้วคลิกปุ่ม Next >



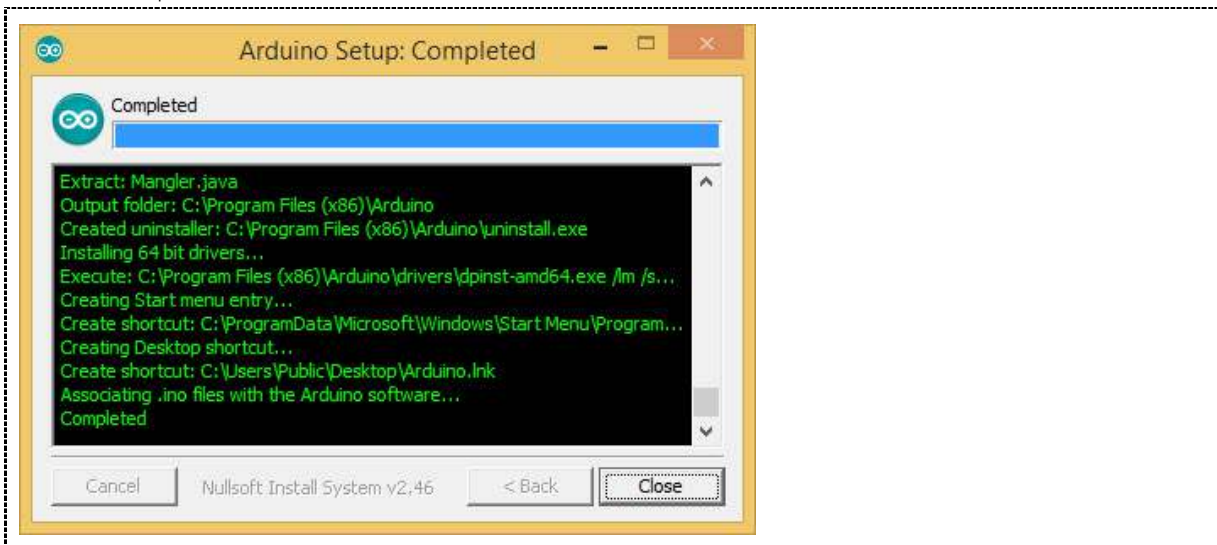
เลือกโฟลเดอร์ติดตั้งโปรแกรม หากไม่ต้องการแก้ไขคลิกปุ่ม Install ได้เลย



รอจนกว่าโปรแกรมจะติดตั้งเสร็จสิ้น



เมื่อขึ้นคำว่า Completed หมายถึงการติดตั้งเสร็จสมบูรณ์แล้ว
คลิกปุ่ม Close เพื่อปิดโปรแกรมลงไปได้เลย

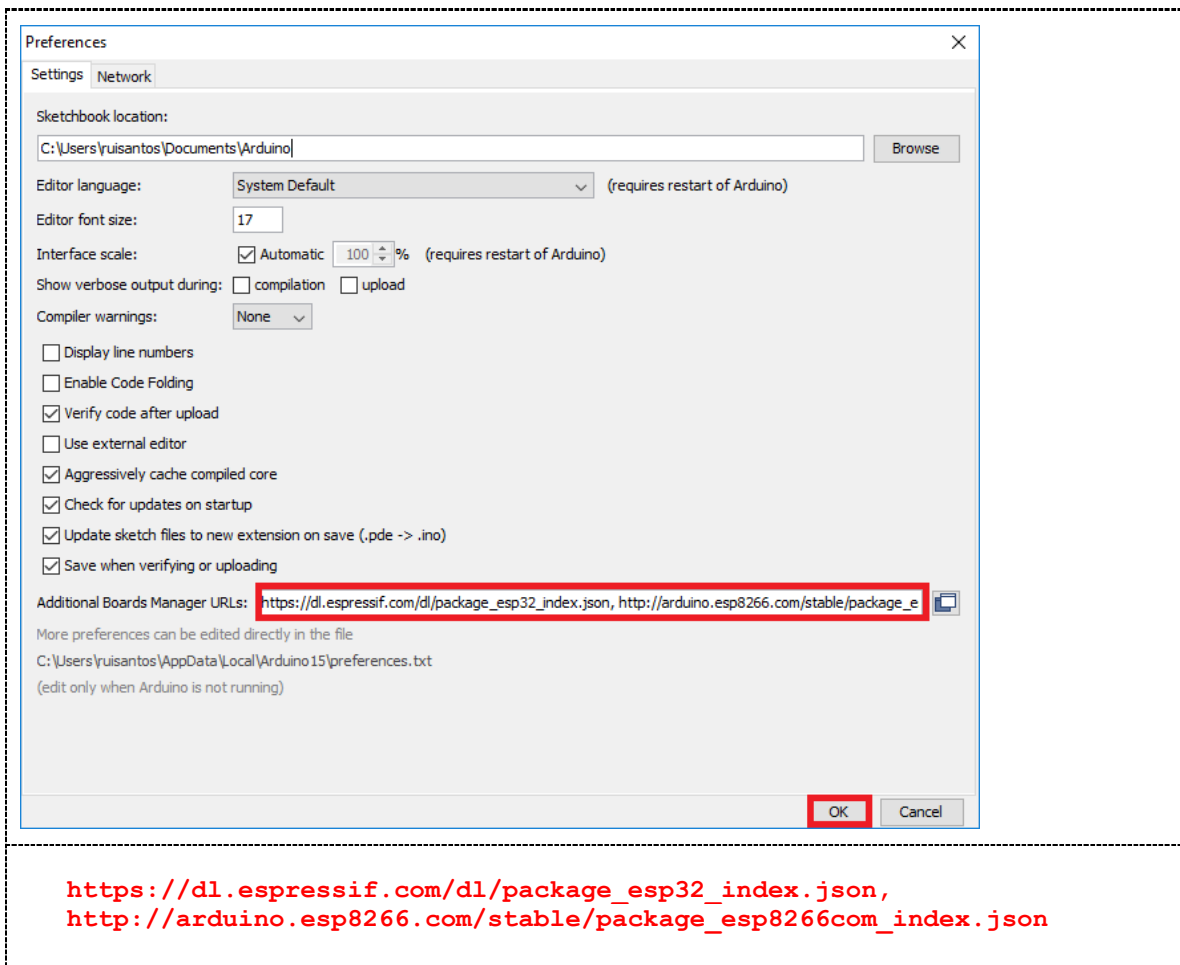


หน้าเดสท็อปก็น่าจะมีไอคอนโปรแกรม Arduino ขึ้นมาแล้ว

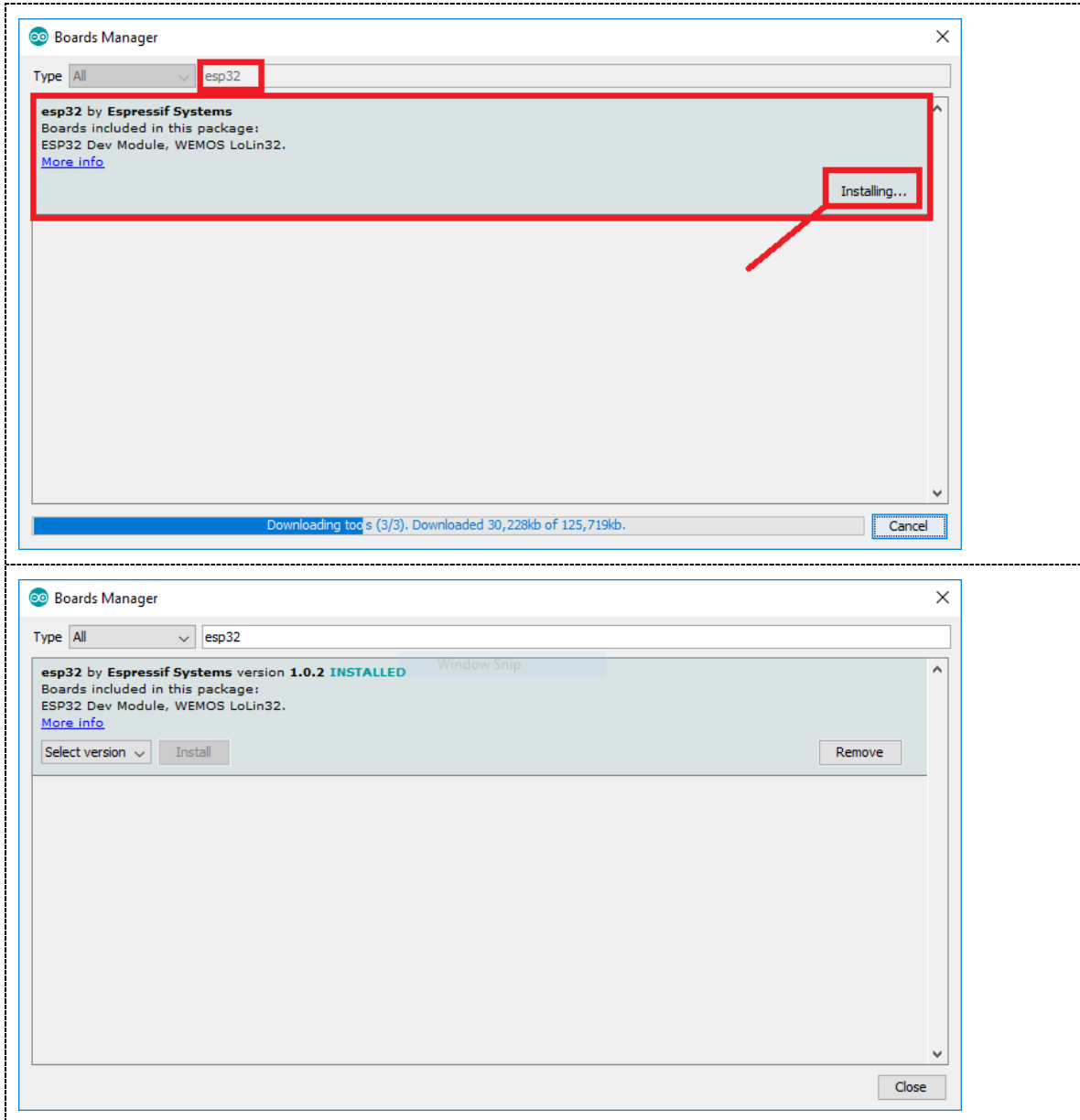


1.3 Add ESP32 Board ตามขั้นตอนดังนี้

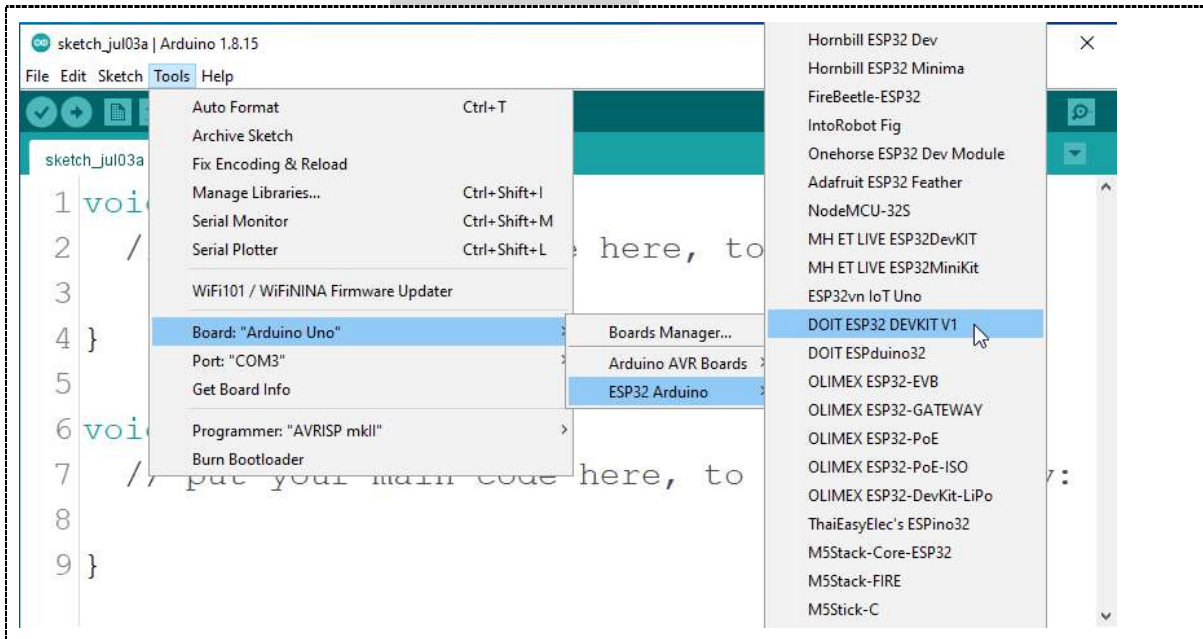
- <https://randomnerdtutorials.com/installing-the-esp32-board-in-arduino-ide-windows-instructions/>
- เข้าเมนู File >> Preferences จะขึ้นหน้าต่าง Preferences ให้สังเกตในช่อง Additional Board Manager URLs:
- ใส่ URL >> ลงใน Additional Board Manager URLs: ดังนี้
https://dl.espressif.com/dl/package_esp32_index.json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



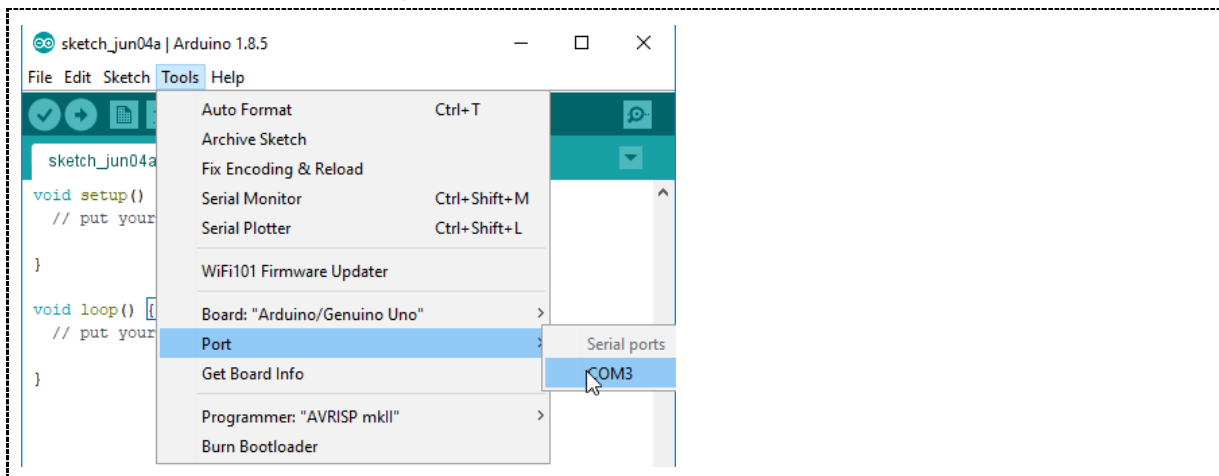
- ไปที่ Menu Tools → Board : “xxxxx” → Board Manager
- เลือกบอร์ด esp32 ของ ESP32 by Espressif Systems → กด Install



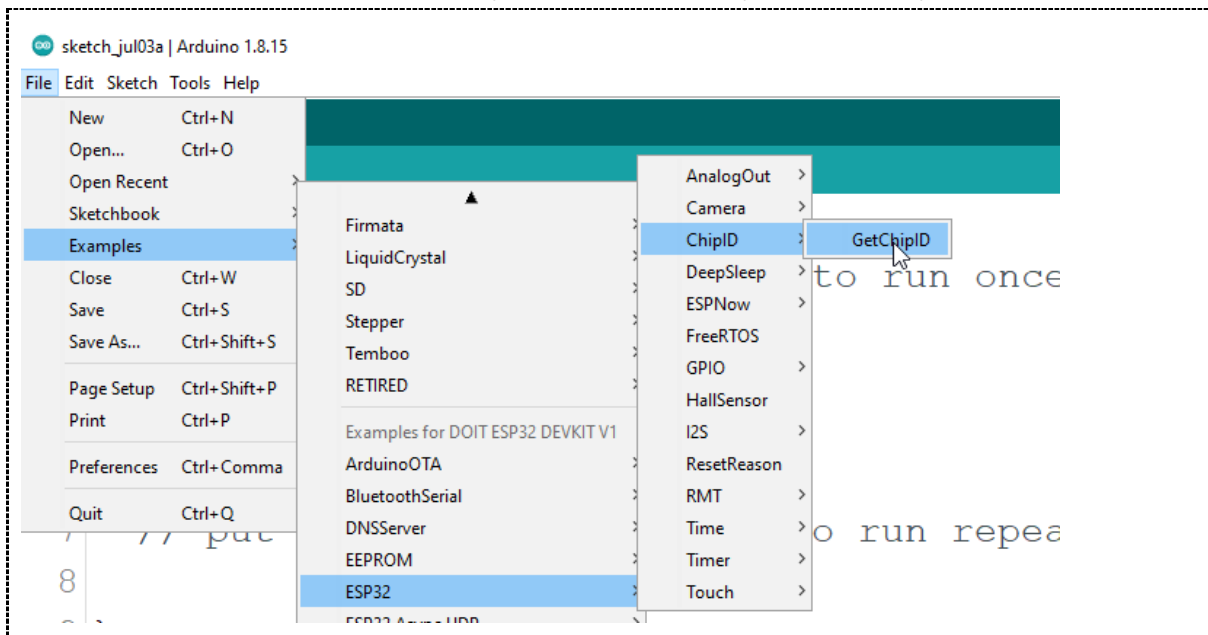
- เมื่อติดตั้งเสร็จ เข้าไปที่ Tools->Boards ลองเช็คดูว่า Arduino IDE รองรับการใช้งานร่วมกับ ESP32 แล้วหรือไม่ ถ้าเจอหัวข้อ **DOIT ESP32 DEVKIT1** แสดงว่าสามารถใช้งานได้ ESP32 ได้แล้ว



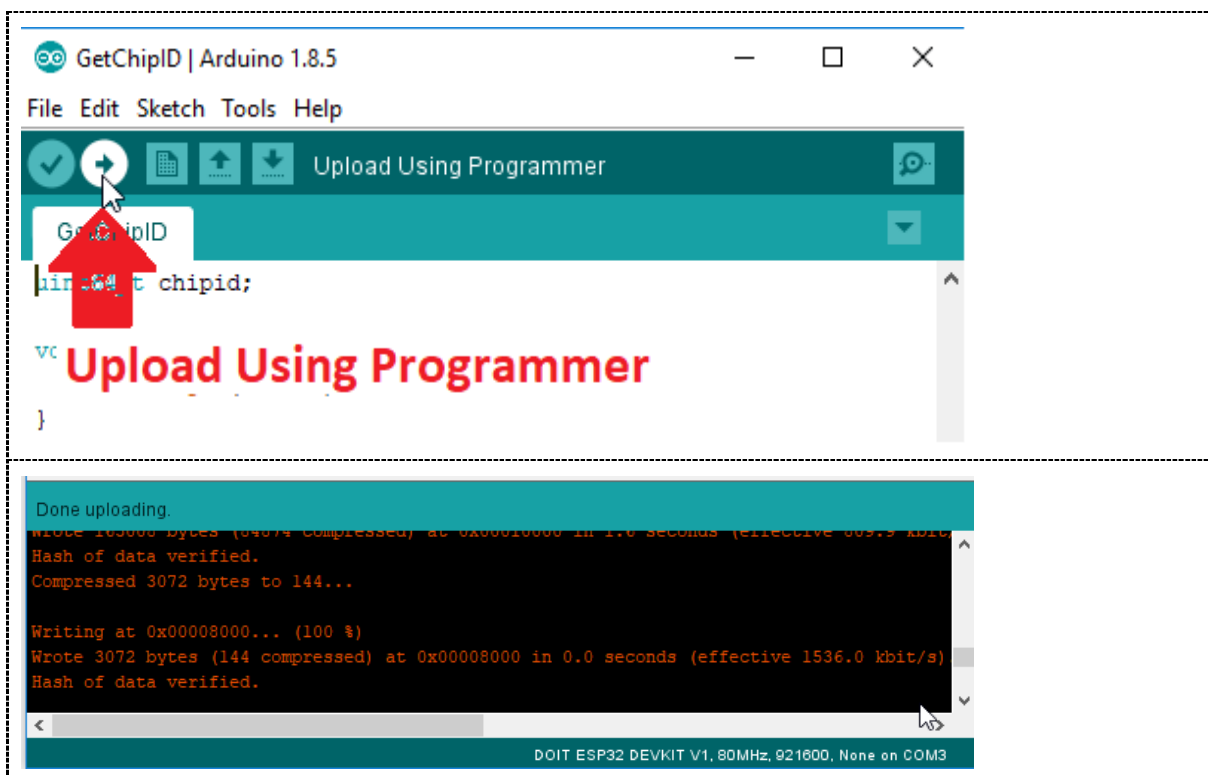
- ต่อบอร์ด ESP32 เข้ากับ PC ผ่าน USB Cable
- เลือกบอร์ดเป็น DOIT ESP32 DEVKIT1
- เลือก Communication port



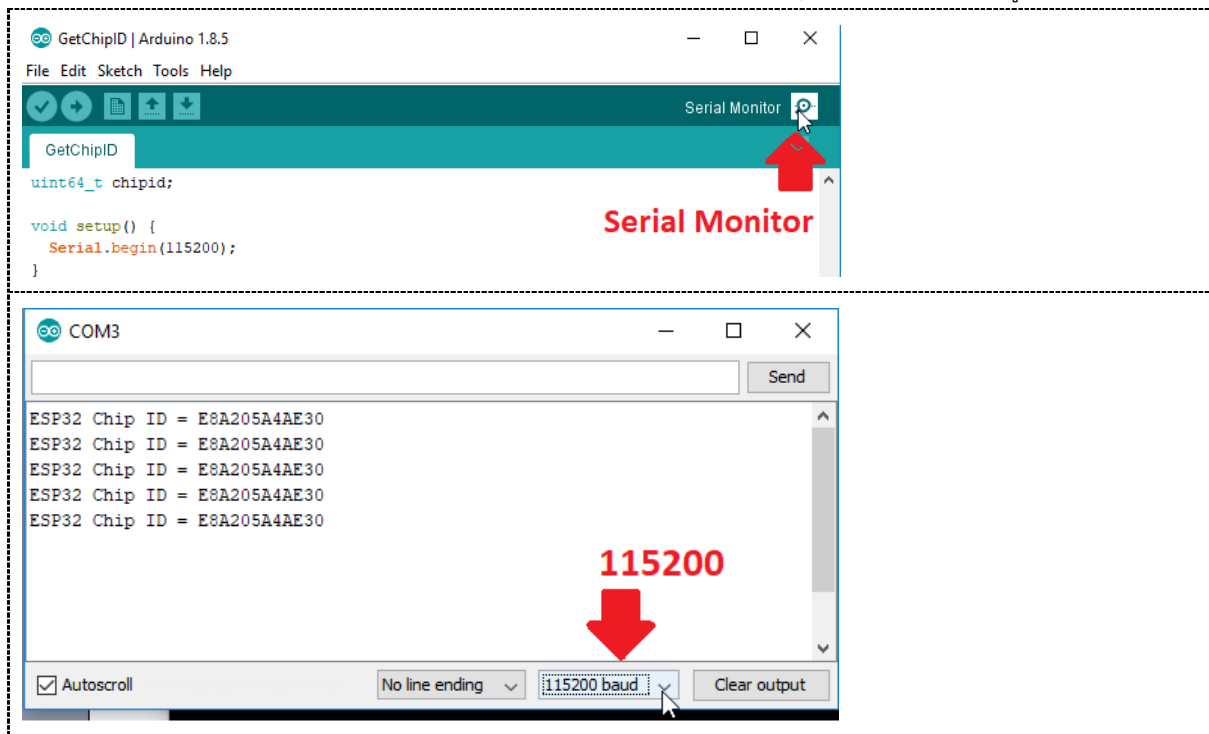
- ทดสอบโปรแกรม File → Example → ESP32 → ChipID → "GetChipID"



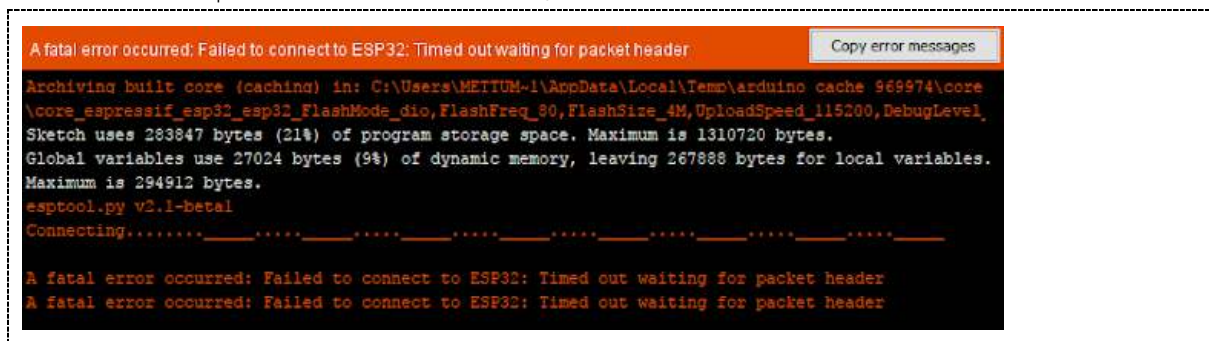
- กด Upload โปรแกรม
- เมื่อโปรแกรมเสร็จเรียบร้อยแล้วจะขึ้นข้อความแสดงในกรอบสีเหลือง



- จากนั้นลองเปิด Serial Monitor ขึ้น ESP32 จะทำการป้อน ChipID แสดงขึ้นมา ดังรูป



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



2/4. การโปรแกรมเพื่อควบคุม ESP-32 สั่งงาน อินพุต/เอาต์พุต

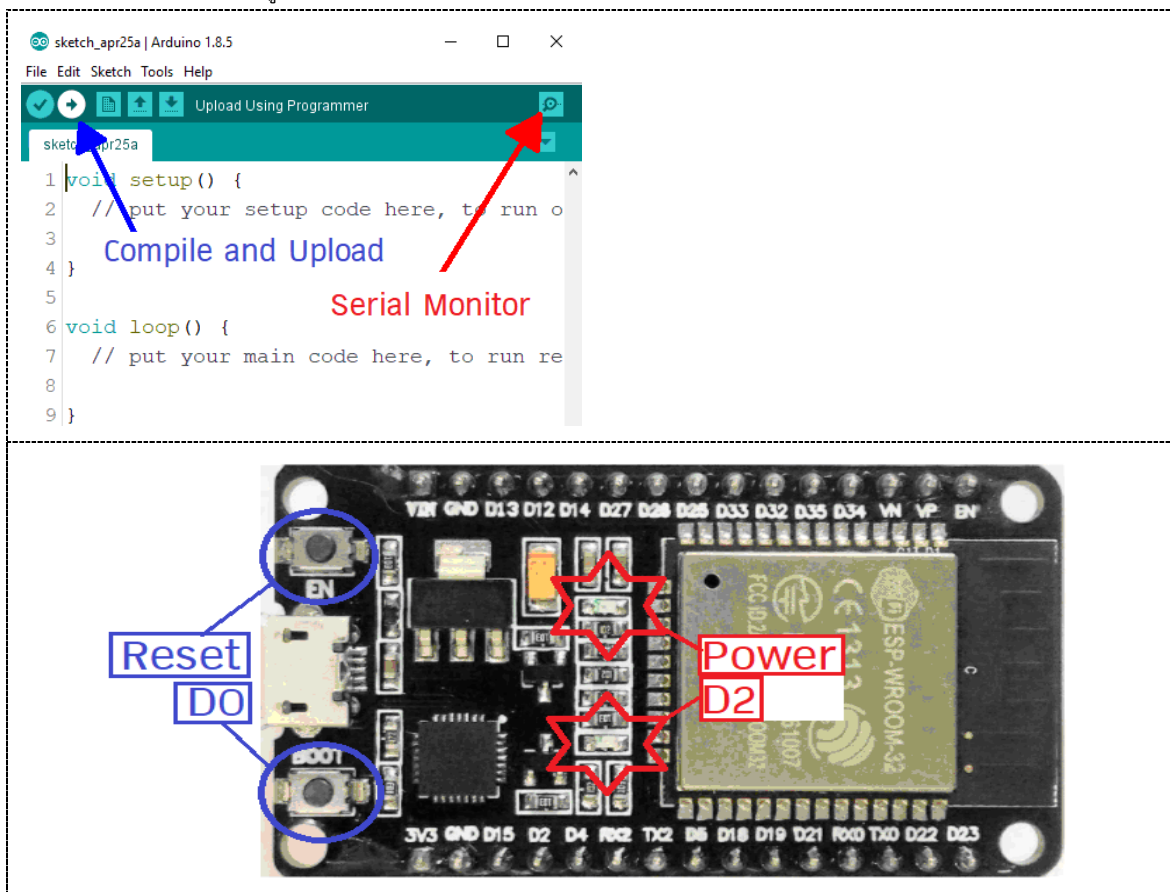
Example_101 - Blink

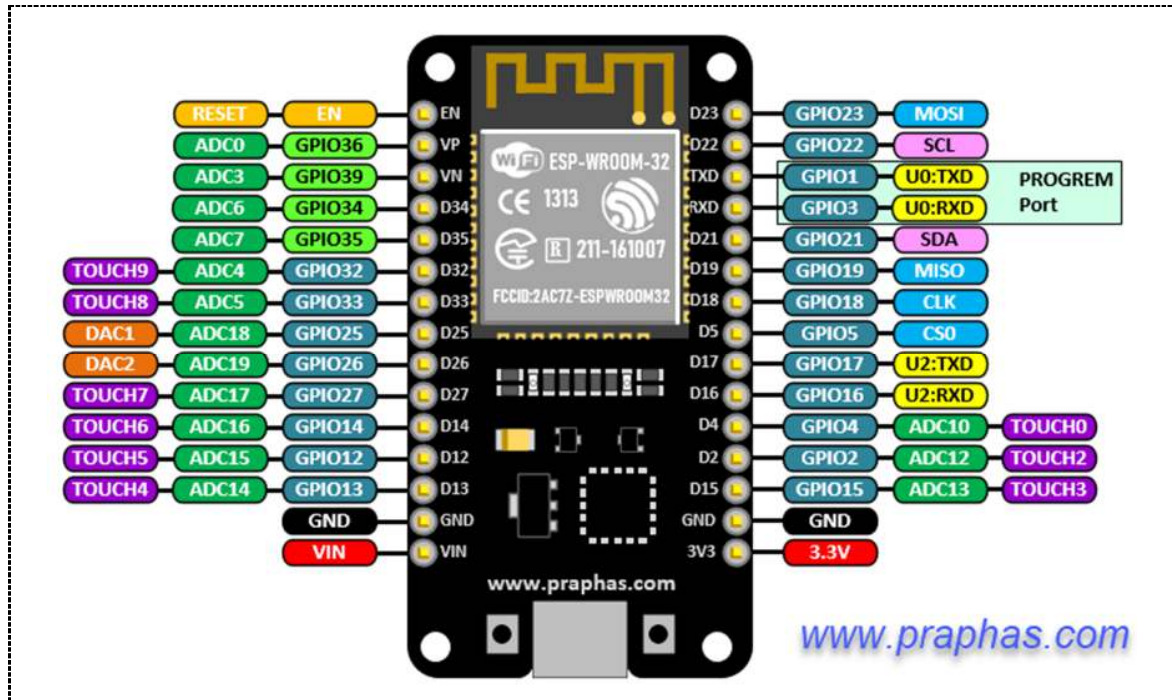
- Tools → Board → DOIT ESP32 DEVKIT V1
- Tools → Port → COM xx
- ทดสอบการทำงานของโปรแกรมไฟกระพริบ File → Example → Basic → Blink

```
// 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
}
```

- เลี้ยวขวา เพื่อ Compile and Upload
- แนวนอนขยาย เพื่อดูหรือสั่งงานผ่าน Serial Monitor





Example_102 - Display Single LED

// 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(2, OUTPUT);
}

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

// Example_102c – Single LED definind pin

```
#define LED_test 2

void setup() {
  pinMode(2, OUTPUT);
}

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

// Example_102d – Single LED with array

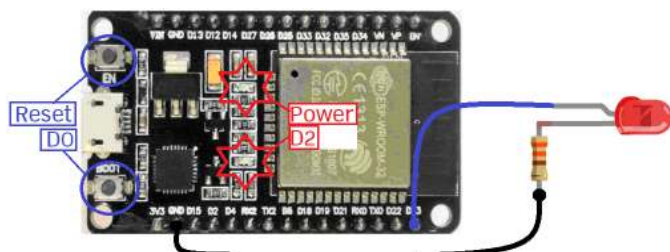
```
int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0};

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);
  }
}
```

คำถาม

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

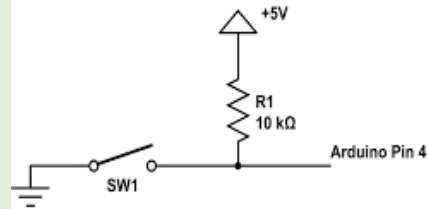


Example_103a - Input SW

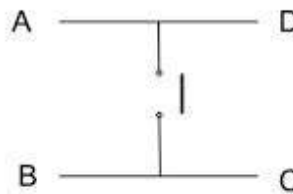
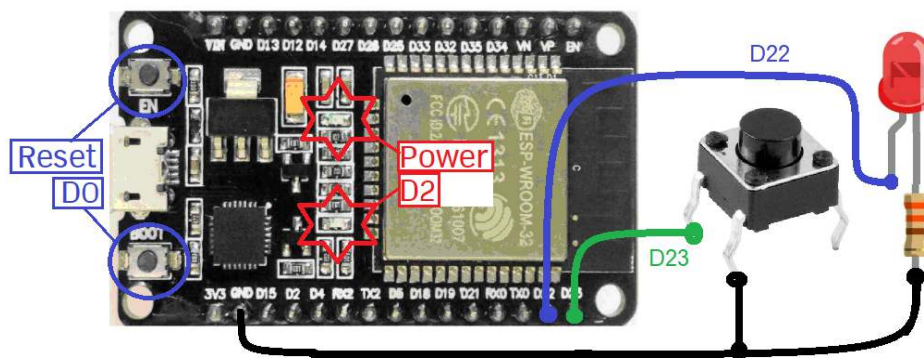
```
// Example_103a - Input Switch
#define pushButton 0

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

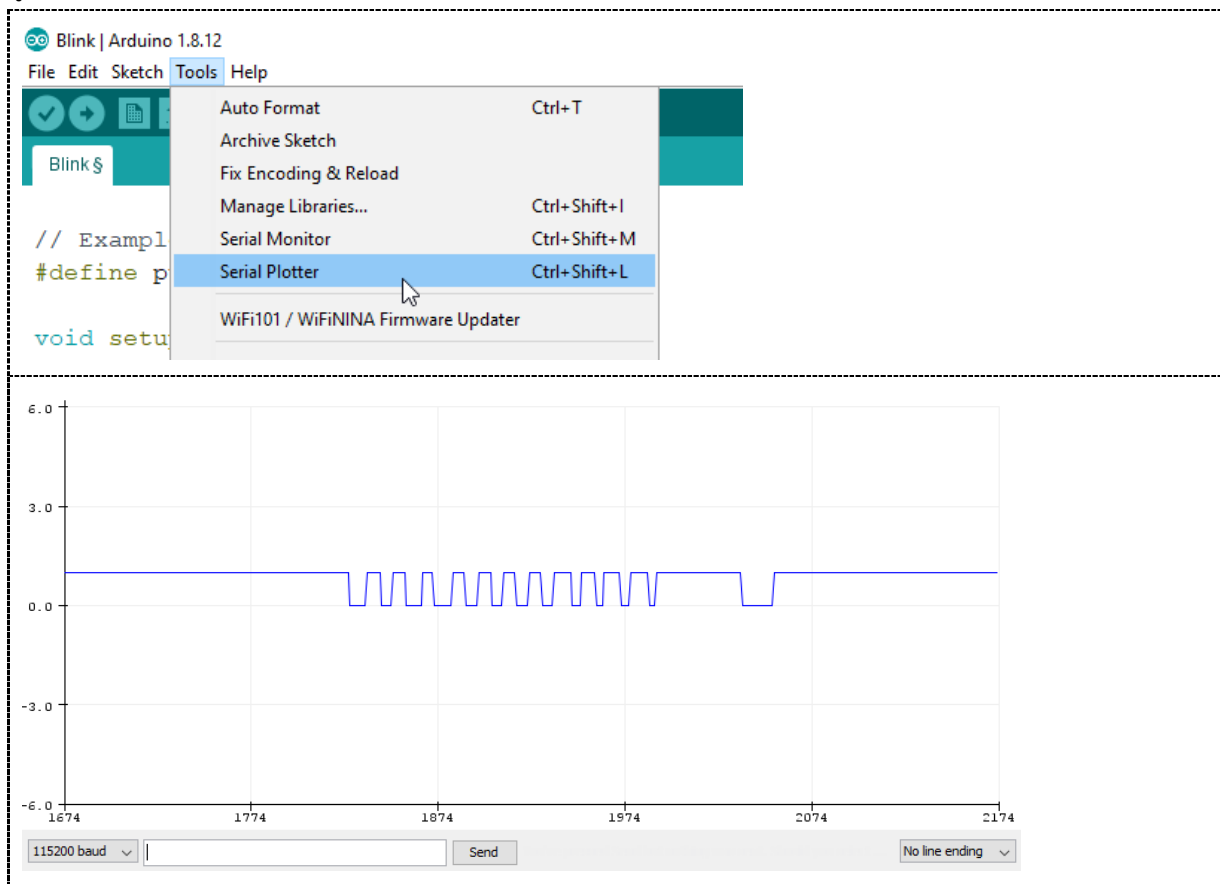
void loop() {
  int swRead = digitalRead(pushButton);
  Serial.println(swRead);
  delay(10);
}
```



- External Pull up
- Internal Pull up



ดูการทำงานเมื่อกดสวิตช์ที่ Tools → Serial Plotter



Example_103b - Input SW – กดติด ปลอยดับ

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

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

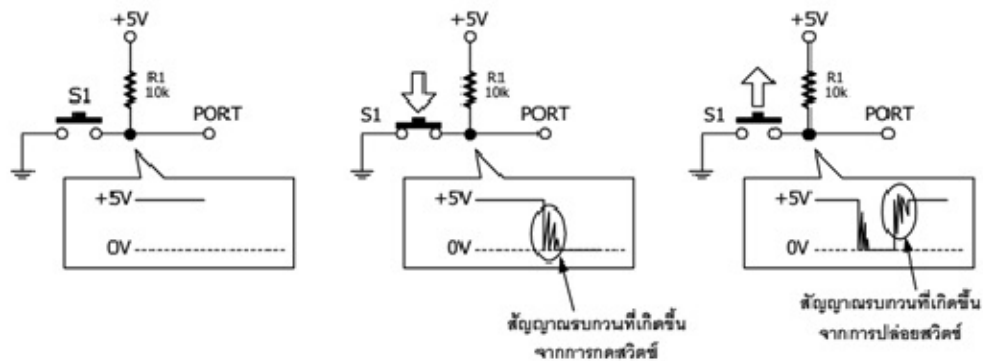
void loop() {
  int buttonState = digitalRead(pushButton);
  digitalWrite(LEDPin, buttonState);
  Serial.println(buttonState);
  delay(1);
}
```

Example_104 - Input SW – 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);
  }
}
```



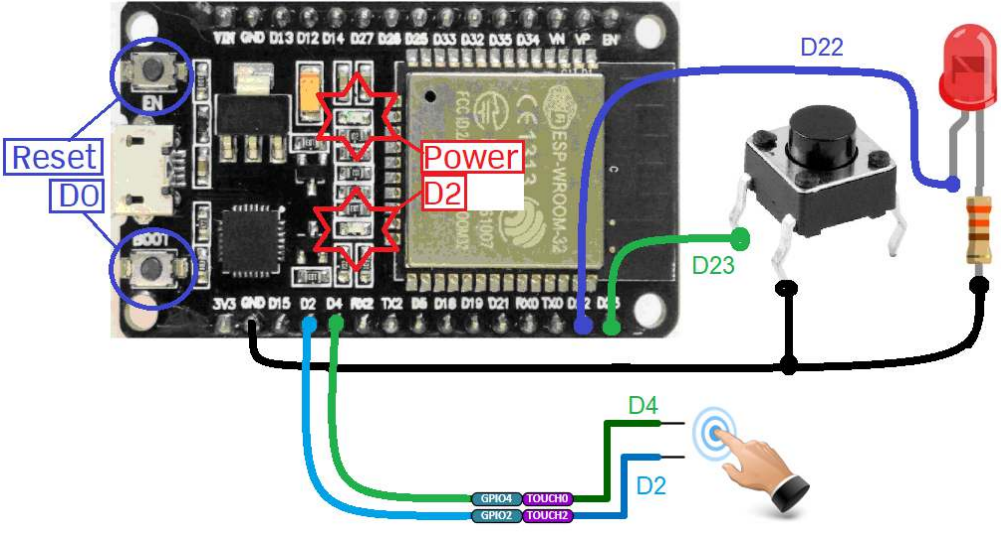
Example_105 - Input SW – กดติด กดดับ

```
// 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);
  }
}
```

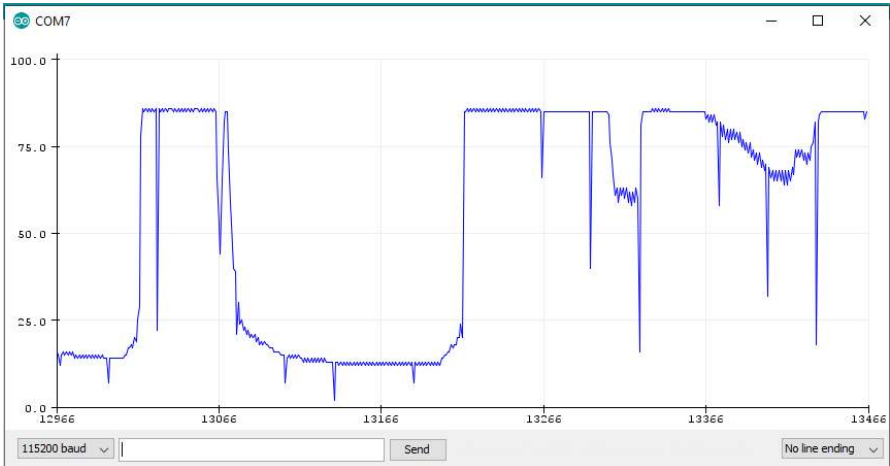
Example_106 – Touch Switch – กดติด กดดับ



```
// Example_106a – Read Touch
#define pushButton 4

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

void loop() {
  int swRead = touchRead(T0);
  Serial.println(swRead);
  delay(10);
}
```



touchRead(Touch pin);

ฟังก์ชันอ่านค่าจากเซนเซอร์สัมผัส Touch pin: หมายเลขขาพอร์ทของเซอร์เซอร์สัมผัสที่ต้องการอ่านค่า เช่น x=touchRead(T0); 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;

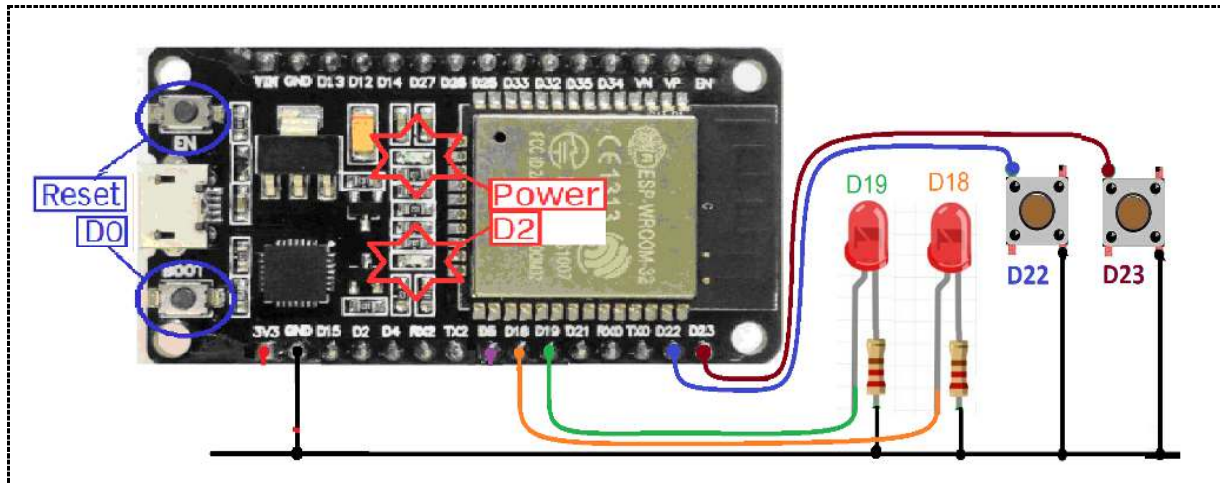
void setup() {
  Serial.begin(115200);
  pinMode(pushButton, INPUT_PULLUP);
  pinMode(LEDPin, OUTPUT);
}

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

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



3/4. การติดตั้งส่วนเสริมเพื่อเรียกใช้งาน สำหรับบอร์ดต่อขยาย อินพุต/เอาต์พุต

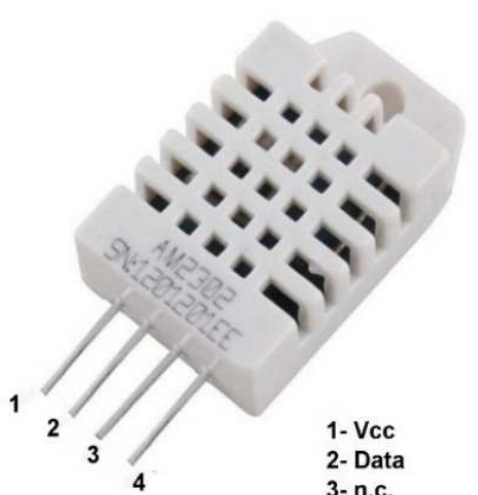
Example_201 - การใช้งาน DHT-22

DHT-22 (Humidity and Temperature Sensor) คือ เซ็นเซอร์ที่ใช้ในการวัดอุณหภูมิกับความชื้นในอากาศ โดยมีคุณสมบัติ ดังนี้

- ย่านวัดความชื้น 0-100% RH โดยมีค่าความแม่นยำ $\pm 5\%$ RH
- ความละเอียดในการวัด 2-5% แสดงผลแบบ 8 บิต
- ย่านวัดอุณหภูมิ -40 ถึง 80 องศาเซลเซียส โดยมีค่าความแม่นยำ ± 0.5 องศาเซลเซียส
- ความละเอียดในการวัด 1 องศาเซลเซียส แสดงผลแบบ 8 บิต
- กินกระแส 0.5 – 2.5 mA (ขณะทำการวัดค่า ที่ระดับแรงดัน 3 – 5.5 VDC)
- อ่านค่าสัญญาณ (Sample Rate) ทุก 0.5 วินาที
- Read This
 - <https://www.arduitronics.com/article/13/การใช้งาน-dht11-humidty-and-temperature-sensor-กับบอร์ด-arduino>
 - http://www.thaieasyelec.com/information_support/EADN057/EADN057-i%20Module%20-%20SHT11%20Humidity%20and%20Temp%20Sensor%20Kit%20V1.0-QuickStart%20Guide.pdf
 - http://www.sbt.ac.th/new/sites/default/files/TNP_Unit_10.pdf

DHT22 Temperature-Humidity Sensor

- 3.3 to 6V power and I/O
- 1.5mA max current use during conversion
- 0-100% humidity readings with 2-5% accuracy
- 40 to 80°C temperature readings $\pm 0.5^{\circ}\text{C}$ accuracy
- Up to 0.5 Hz sampling rate (once every 2 seconds)
- 4 pins, 0.1" spacing

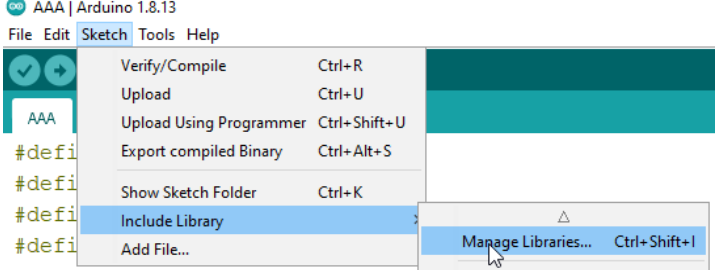
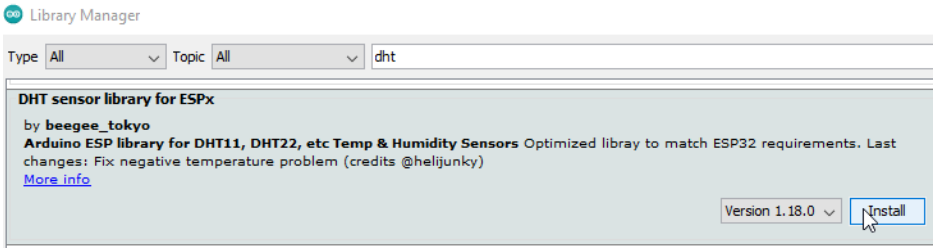
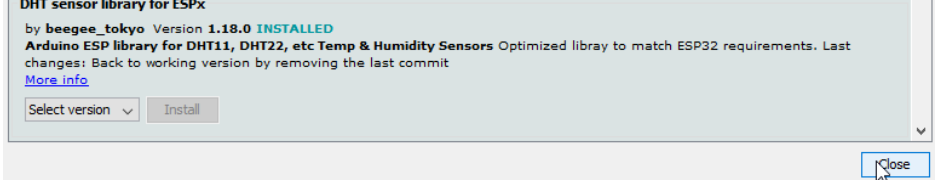


1) VCC
 2) DATA (digital I/O)
 3) Not Connected (N.C)
 4) GND

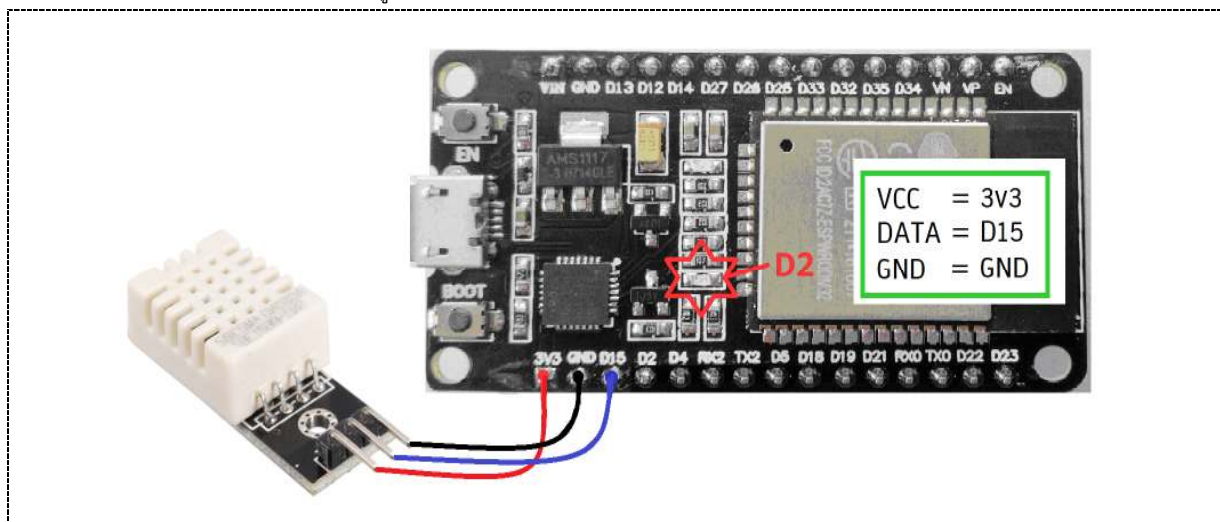
1- Vcc
 2- Data
 3- n.c.
 4- GND

การทดสอบ

1. Add DHT-11 Sensor Library เริ่มจาก Sketch → Inc Library → Manage
2. Filter ด้วยข้อความ **DHT**, เลือก **DHT Sensor library for ESPx V1.17.0** แล้วทำการติดตั้ง

	Sketch → Include Lib → Manage ...
	Filter → dht22 เลือก DHT sensor library for ESPx Version 1.18.0
	เมื่อสำเร็จจะแสดง 1.18.0 INSTALLED . Close

3. การต่อวงจรภายในเป็น ดังรูป



4. ทดสอบการทำงาน ด้วยโปรแกรมดังต่อไปนี้ และผลการทำงานด้วย Serial Monitor @Baud Rate = 115200

```

#define DHT22_Pin 15
#include "DHTesp.h"

DHTesp dht;

void setup() {
  Serial.begin(115200);
  Serial.println();
  Serial.println("Status\tHumidity (%)\tTemperature (C)\t(F)\tHeatIndex (C)\t(F)");
  dht.setup(DHT22_Pin, DHTesp::DHT22); // Connect DHT sensor to GPIO 15
}

void loop() {
  delay(dht.getMinimumSamplingPeriod());

  float humidity = dht.getHumidity();
  float temperature = dht.getTemperature();

  Serial.print(dht.getStatusString());
  Serial.print("\t");
  Serial.print(humidity, 1);
  Serial.print("\t");
  Serial.print(temperature, 1);
  Serial.print("\t");
  Serial.print(dht.toFahrenheit(temperature), 1);
  Serial.print("\t");
  Serial.print(dht.computeHeatIndex(temperature, humidity, false), 1);
  Serial.print("\t");
  Serial.println(dht.computeHeatIndex(dht.toFahrenheit(temperature), humidity, true), 1);
  delay(2000);
}

```

```

configsip: 0, SPIWP:0xee
clk_drv:0x00,q_drv:0x00,d_drv:0x00,cs0_drv:0x00,hd_drv:0x00,wp_drv:0x00
mode:DIO, clock div:1
load:0x3fff0018,len:4
load:0x3fff001c,len:1044
load:0x40078000,len:8896
load:0x40080400,len:5816
entry 0x400806ac

Status Humidity (%) Temperature (C) (F) HeatIndex (C) (F)
OK 78.4 25.7 78.3 27.1 80.7
OK 78.4 25.7 78.3 27.1 80.7
OK 78.3 25.7 78.3 27.1 80.7
OK 78.3 25.7 78.3 27.1 80.7
OK 77.9 25.7 78.3 27.1 80.7

```

Quiz_102 – ปรับการแสดงผลที่ Serial Monitor เป็นดังนี้

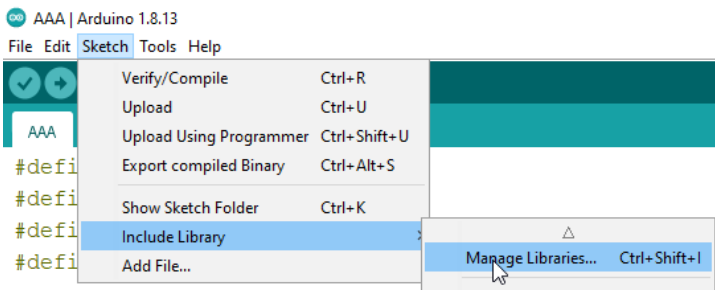
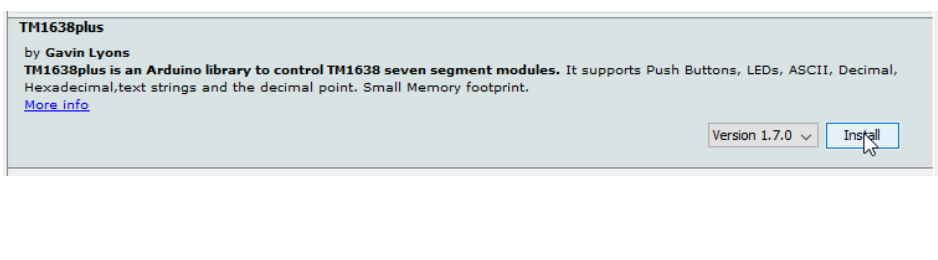
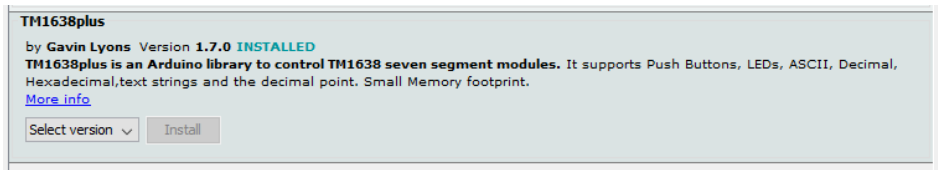
Temperature: 23.0C / 74.7F. Humidity: 24.9%

Temperature: 23.0C / 74.7F. Humidity: 24.9%

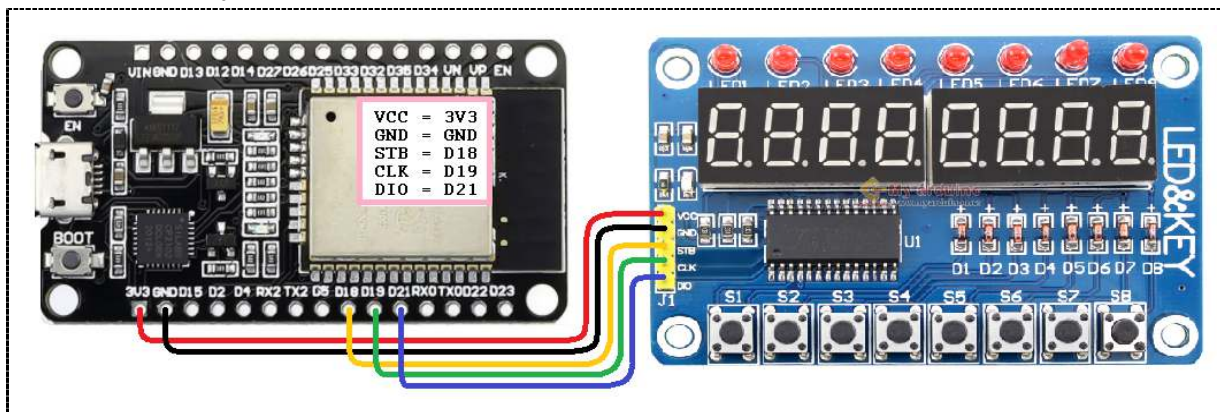
Temperature: 23.0C / 74.7F. Humidity: 24.9%

Example_202 - การใช้งาน TM1638

1. Read <https://blog.3d-logic.com/2015/01/10/using-a-tm1638-based-board-with-arduino/>
2. เพิ่มไลบรารีของ TM1638 ให้กับ Arduino IDE ตามลำดับ

	Sketck → Include Lib → Manage ...
	Filter → tm1638 เลือก TM1638plus Version 1.7.0
	เมื่อสำเร็จจะแสดง 1.7.0 INSTALLED

3. ต่อวงจรดังรูป



4. ทดสอบการทำงานของโปรแกรม-5A

```

#include <TM1638plus.h>

#define Brd_STB 18 // strobe = GPIO connected to strobe line of module
#define Brd_CLK 19 // clock = GPIO connected to clock line of module
#define Brd_DIO 21 // data = GPIO connected to data line of module
bool high_freq = true; //default false,, If using a high freq CPU > ~100 MHZ set to true.

TM1638plus tm(Brd_STB, Brd_CLK , Brd_DIO, high_freq);
#define myTestDelay 1000

void setup()
{ tm.displayBegin();
  delay(myTestDelay);
}

void loop() {
  tm.displayHex(0, 1);
  tm.displayHex(1, 2);
  tm.displayHex(2, 3);
  tm.displayHex(3, 4);
  tm.displayHex(4, 5);
  tm.displayHex(5, 6);
  tm.displayHex(6, 7);
  tm.displayHex(7, 8);
  delay(myTestDelay); // display 12345678

  tm.displayHex(0, 8);
  tm.displayHex(1, 9);
  tm.displayHex(2, 10);
  tm.displayHex(3, 11);
  tm.displayHex(4, 12);
  tm.displayHex(5, 13);
  tm.displayHex(6, 14);
  tm.displayHex(7, 15);
  delay(myTestDelay); // display 89ABCDEF
}

```

5. การเรียงตำแหน่งของ 7_Segment จากซ้ายไปขวาเรียงอย่างไร

6. ทดสอบการทำงานของโปรแกรม-5B

```

#include <TM1638plus.h>

#define Brd_STB 18 // strobe = GPIO connected to strobe line of module
#define Brd_CLK 19 // clock = GPIO connected to clock line of module
#define Brd_DIO 21 // data = GPIO connected to data line of module
bool high_freq = true; //default false,, If using a high freq CPU > ~100 MHZ set to true.

TM1638plus tm(Brd_STB, Brd_CLK, Brd_DIO, high_freq);
#define myTestDelay 1000

void setup()
{ Serial.begin(115200);
  tm.displayBegin();
  delay(myTestDelay);
}

void loop() {
  int SwInput = tm.readButtons();
  Serial.print("Switch Input = ");
  Serial.println(SwInput, HEX);

  tm.display7Seg(0, 0x40); >>> 0100 0000 >> tgfe dcba
  tm.display7Seg(1, 0x40);
  tm.display7Seg(2, 0x40);
  tm.display7Seg(3, 0x40);
  tm.display7Seg(4, 0x40);
  tm.display7Seg(5, 0x40);
  tm.display7Seg(6, 0x40);
  tm.display7Seg(7, 0x40);

  if (SwInput == 0x01) tm.displayHex(0, 1);
  if (SwInput == 0x02) tm.displayHex(1, 2);
  if (SwInput == 0x04) tm.displayHex(2, 3);
  if (SwInput == 0x08) tm.displayHex(3, 4);
  if (SwInput == 0x10) tm.displayHex(4, 5);
  if (SwInput == 0x20) tm.displayHex(5, 6);
  if (SwInput == 0x40) tm.displayHex(6, 7);
  if (SwInput == 0x80) tm.displayHex(7, 8);

  delay(100);
}

```

7. คำสั่งใดใช้เพื่ออ่านค่าการกดสวิตช์
8. การเรียงตำแหน่งของ Switch จากซ้ายไปขวา เมื่อกดสวิตช์ จะให้ค่าอย่างไร
9. คำสั่ง . displayHex, . display7Seg คือคำสั่งอะไร ใช้งานต่างกันอย่างไร
10. หาก 8 บรรทัดก่อนสุดท้าย เริ่มจากบรรทัด if (SwInput == 0x01) tm.displayHex(0, 1); แก้ไขจาก == เป็น = การทำงานยังทำงานได้เหมือนเดิมหรือไม่ ☒ ทำไม่ได้ ☐ ทำได้
11. สัญลักษณ์ = และ == ใช้งานต่างกันอย่างไร

12. ทดสอบการทำงานของโปรแกรม-5C

```

#include <TM1638plus.h>

#define Brd_STB 18 // strobe = GPIO connected to strobe line of module
#define Brd_CLK 19 // clock = GPIO connected to clock line of module
#define Brd_DIO 21 // data = GPIO connected to data line of module
bool high_freq = true; //default false,, If using a high freq CPU > ~100 MHZ set to true.

TM1638plus tm(Brd_STB, Brd_CLK , Brd_DIO, high_freq);
#define myTestDelay 1000

void setup()
{ Serial.begin(115200);
  tm.displayBegin();
  delay(myTestDelay);
  tm.display7Seg(0, 0x40);
  tm.display7Seg(1, 0x40);
  tm.display7Seg(2, 0x40);
  tm.display7Seg(3, 0x40);
  tm.display7Seg(4, 0x40);
  tm.display7Seg(5, 0x40);
  tm.display7Seg(6, 0x40);
  tm.display7Seg(7, 0x40);
}

void loop() {
  int SwInput = tm.readButtons();
  Serial.print("Switch Input = ");
  Serial.println(SwInput, HEX);
  tm.setLED(0, 0);
  tm.setLED(1, 0);
  tm.setLED(2, 0);
  tm.setLED(3, 0);
  tm.setLED(4, 0);
  tm.setLED(5, 0);
  tm.setLED(6, 0);
  tm.setLED(7, 0);

  if (SwInput == 0x01)      tm.setLED(0, 1);
  if (SwInput == 0x02)      tm.setLED(1, 1);
  if (SwInput == 0x04)      tm.setLED(2, 1);
  if (SwInput == 0x08)      tm.setLED(3, 1);
  if (SwInput == 0x10)      tm.setLED(4, 1);
  if (SwInput == 0x20)      tm.setLED(5, 1);
  if (SwInput == 0x40)      tm.setLED(6, 1);
  if (SwInput == 0x80)      tm.setLED(7, 1);
  delay(100);
}

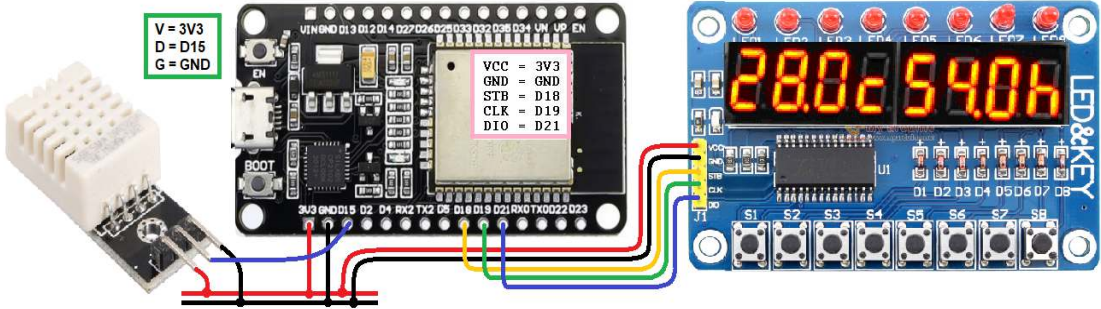
```

13. คำสั่ง `tm.setLED(5, 0);` เลข 5 หมายถึงอะไร ค่าต่ำสุดเท่าไร ค่าสูงสุดเท่าไร

14. คำสั่ง `tm.setLED(5, 0);` เลข 0 หมายถึงอะไร ค่าต่ำสุดเท่าไร ค่าสูงสุดเท่าไร

Quiz_103 – Read Temp and Show

- ต้องวงจรเพิ่มเติม ทดสอบการทำงานด้วยโปรแกรมต่อไปนี้ และปรับแก้ให้ถูกต้อง



```
#include <TM1638plus.h>
#include "DHTesp.h"

#define Pin_DHT22 15 // D15
#define Brd_STB 18 // strobe = GPIO connected to strobe line of module
#define Brd_CLK 19 // clock = GPIO connected to clock line of module
#define Brd_DIO 21 // data = GPIO connected to data line of module
bool high_freq = true; //default false,, If using a high freq CPU > ~100 MHZ set to true.

DHTesp dht;
TM1638plus tm(Brd_STB, Brd_CLK, Brd_DIO, high_freq);

void setup() {
  Serial.begin(115200);
  dht.setup(Pin_DHT22, DHTesp::DHT22);
  tm.displayBegin();
}

void loop() {
  float h = dht.getHumidity();
  float t = dht.getTemperature();
  Serial.print("Temperature: ");
  Serial.print(t); Serial.print(" *C\t");
  Serial.print("Humidity: ");
  Serial.print(h); Serial.print(" %\n");

  int Tempp2 = 7; int Tempp1 = 6; int Tempp0 = 5;
  int Humi2 = 4; int Humi1 = 3; int Humi0 = 2;

  tm.displayHex(0, Tempp2);
  tm.displayASCIllwDot(1, Tempp1 + '0'); // turn on dot
  tm.displayHex(2, Tempp0);
  tm.display7Seg(3, B01011000); // Code=tgfedcba
  tm.displayHex(4, Humi2);
  tm.displayASCIllwDot(5, Humi1 + '0'); // turn on dot
  tm.displayHex(6, Humi0);
  tm.display7Seg(7, B01110100); // Code=tgfedcba

  delay(2000);
}
```

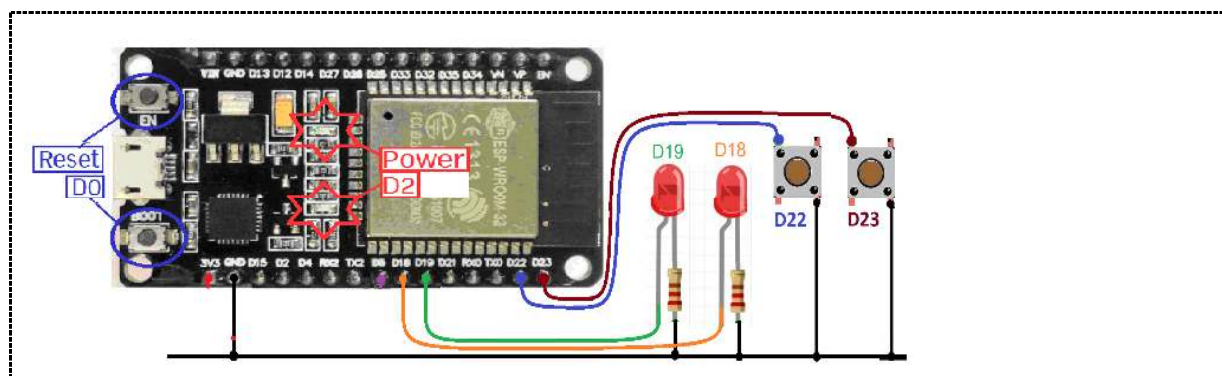
แนวทางการใช้งานอินเทอร์เน็ตของสรรพสิ่งในระบบการผลิต
IoT Approaches to Manufacturing System

ชื่อ-สกุล :

4/4. คำถามท้ายบทเพื่อทดสอบความเข้าใจ

Quiz_101 – กดติด กดดับ 2 ชุด

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



< Test Code >

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

Quiz_102 – ปรับการแสดงผลที่ Serial Monitor เป็นดังนี้

Temperature: 23.0C / 74.7F. Humidity: 24.9% Temperature: 23.0C / 74.7F. Humidity: 24.9% Temperature: 23.0C / 74.7F. Humidity: 24.9%

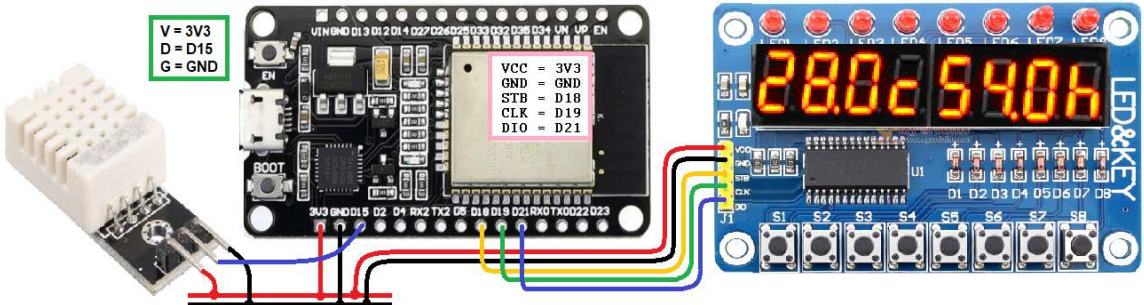
< Test Code >

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2

Quiz_103 – Read Sensor and Show

- ต้องวงจรเพิ่มเติม ทดสอบการทำงานด้วยโปรแกรมต่อไปนี้ และปรับแก้ให้ถูกต้อง



< Test Code >

รูปการต่อวงจร – 1

รูปการต่อวงจร – 2