

การสร้าง MQTT Server บน Raspberry Pi เพื่อใช้งาน Chatbot LINE ในฟาร์มอัจฉริยะ Chatbot LINE from Raspberry Pi MQTT Server for Smart Farming

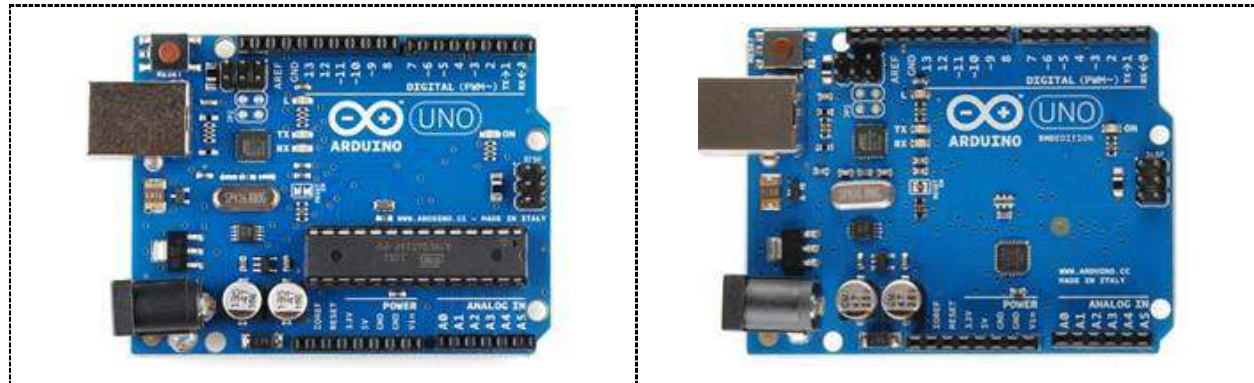
Arduino Microcontroller, การติดตั้ง Arduino IDE และเริ่มต้นใช้งาน

www.arduitronics.com/article/9/มาชมหน้าตาของบอร์ด-arduino-รุ่นต่างๆ-กันดีกว่า
<https://www.cmmakerclub.com/2015/06/esp8266/เริ่มต้นใช้งาน-esp8266-ผ่าน-arduino-ide/>
<https://www.arduinoall.com/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 ที่ผลิตออกมาจำหน่ายในปัจจุบันมีทั้งหมด 20 รุ่น มาลองดูกันคร่าวๆว่าแต่ละรุ่นมีความสามารถและการใช้งานต่างกันไป

1 Arduino Board (Official from Arduino.cc) มีหลากหลายรุ่นที่น่าสนใจ ได้แก่

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

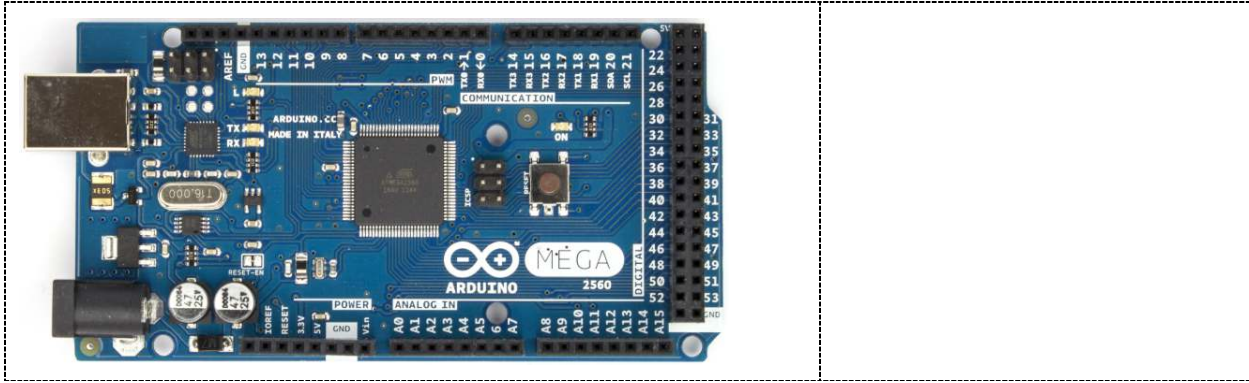


Arduino Uno R3

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)

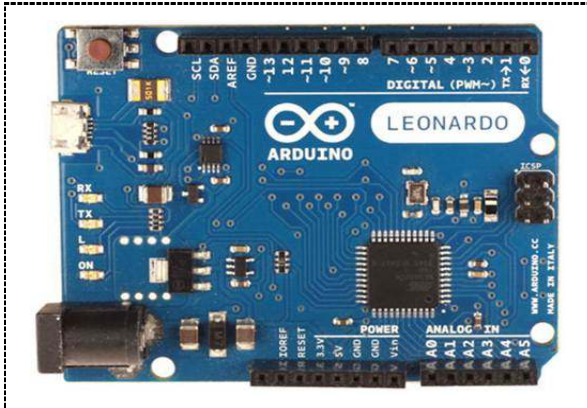
3. **Arduino Mega 2560 R3** เป็นบอร์ด Arduino ที่ออกแบบมาสำหรับงานที่ต้องใช้ I/O มากกว่า Arduino Uno R3 เช่น งานที่ต้องการรับสัญญาณจาก Sensor หรือควบคุมมอเตอร์ Servo หลายๆ ตัว ทำให้ Pin I/O ของบอร์ด Arduino Uno R3 ไม่สามารถรองรับได้ ทั้งนี้ บอร์ด Mega 2560 R3 ยังมีความหน่วยความจำแบบ Flash มากกว่า Arduino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



4. **Arduino Mega ADK** เป็นบอร์ดที่ออกแบบมาให้บอร์ด Mega 2560 R3 สามารถติดต่อกับอุปกรณ์ Android Device ผ่านพอร์ต USB Host ของบอร์ดได้ (ดูรายละเอียดสินค้า)



5. **Arduino Leonardo** การทำงานจะคล้ายกับบอร์ด Arduino Uno R3 แต่มีการเปลี่ยน MCU ตัวใหม่เป็น ATmega32U4 ซึ่งมีโมดูลพอร์ต USB มาด้วยบนชิป (แตกต่างจากบอร์ด Arduino UNO R3 หรือ Arduino Mega 2560 ที่ต้องใช้ชิป ATmega16U2 ร่วมกับ ATmega328 ในการเชื่อมต่อกับพอร์ต USB)



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

6. Arduino Pro Mini 328 3.3V เป็นบอร์ด Arduino ขนาดเล็ก ที่ใช้ MCU เบอร์ ATmega328 ซึ่งจะคล้ายกับบอร์ด Arduino Mini 05 แต่บนบอร์ดจะมี 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 แต่บนบอร์ดจะมี Regulator 5V ชุดเดียวเท่านั้น ระดับแรงดันไฟที่ขา I/O คือ 5V

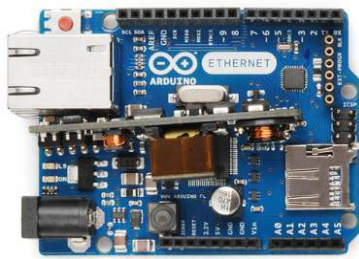


Arduino Pro Mini 328 3.3V

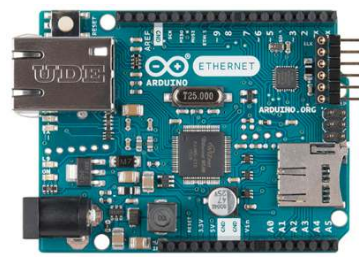


Arduino Pro Mini 328 5V

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



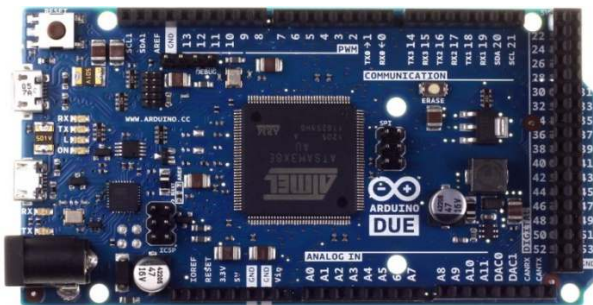
Arduino Ethernet with PoE module



Arduino Ethernet without PoE module

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 ที่เปลี่ยนชิป MCU ใหม่ ซึ่งจากเดิมเป็นตระกูล AVR เปลี่ยนเป็นเบอร์ AT91SAM3X8E(ตระกูล ARM Cortex-M3) แทน ทำให้การประมวลผลเร็วขึ้น แต่ยังคงรูปแบบโค้ดโปรแกรมของ Arduino ที่ง่ายอยู่



ข้อควรระวัง:เนื่องจาก MCU เป็นคนละเบอร์กับ Arduino Uno R3 อาจทำให้บอร์ด Shield บางตัวหรือ Library ใช้ร่วมกันกับบอร์ด Arduino Leonardo ไม่ได้ ผู้ใช้งานจำเป็นต้องตรวจสอบก่อนใช้งาน

2 การติดตั้งโปรแกรม Arduino IDE และการเพิ่ม ESP32 Board

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

2.1 การดาวน์โหลดโปรแกรม Arduino IDE

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

Downloads



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 [Get](#)
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

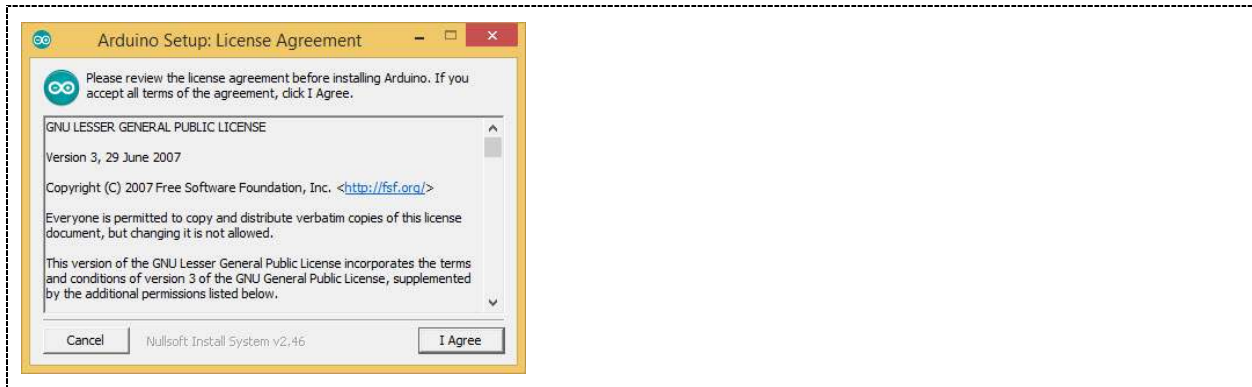
Other

JUST DOWNLOAD

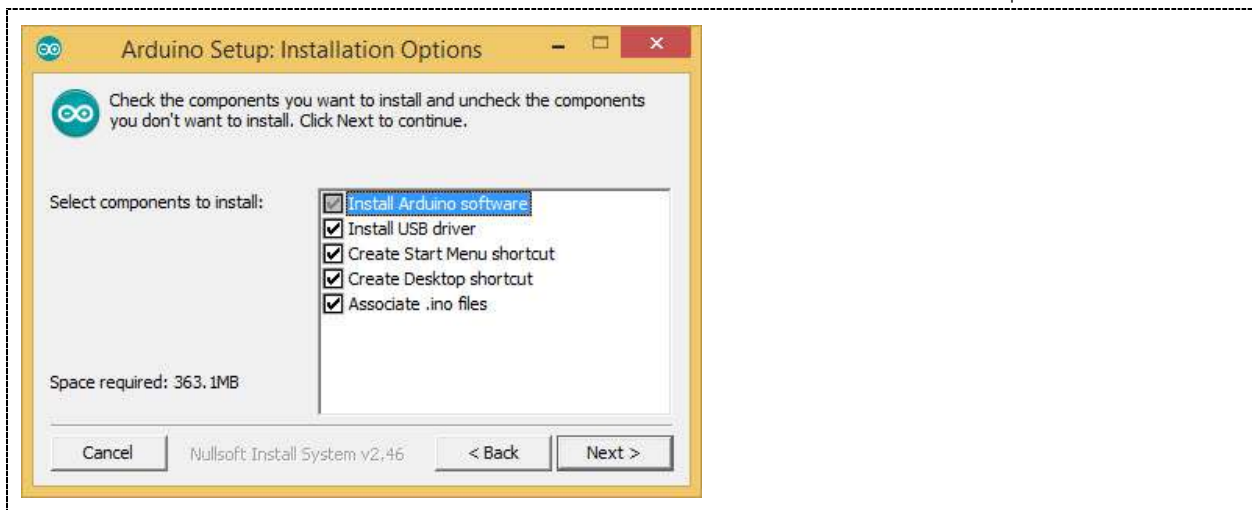
CONTRIBUTE & DOWNLOAD

2.2 การติดตั้งโปรแกรม Arduino IDE

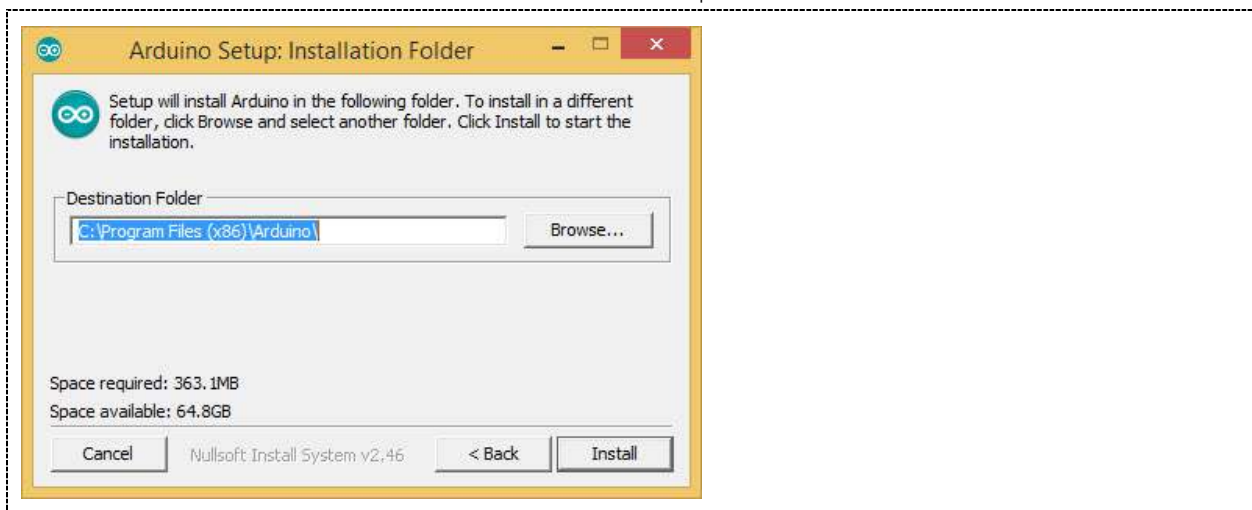
เมื่อดาวน์โหลดเสร็จแล้วให้เปิดไฟล์ติดตั้งขึ้นมาได้เลย กดปุ่ม I Agree ได้เลย



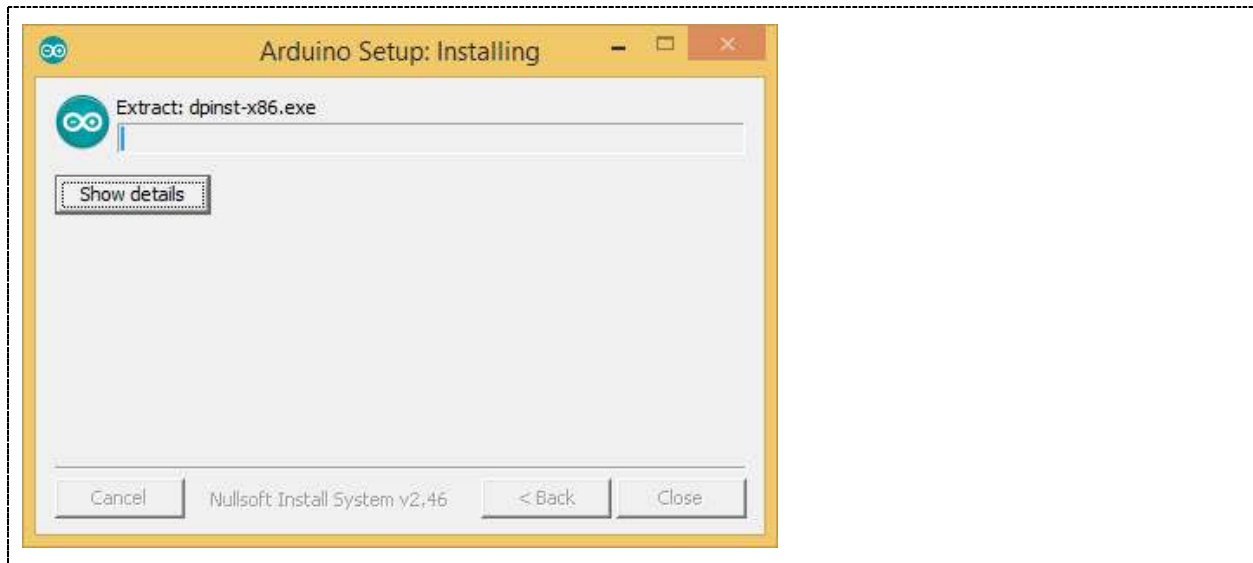
มีตัวเลือกให้เลือกติดตั้ง แนะนำให้เลือกทั้งหมด (ค่าเริ่มต้นคือเลือกทั้งหมด) แล้วคลิกปุ่ม Next >



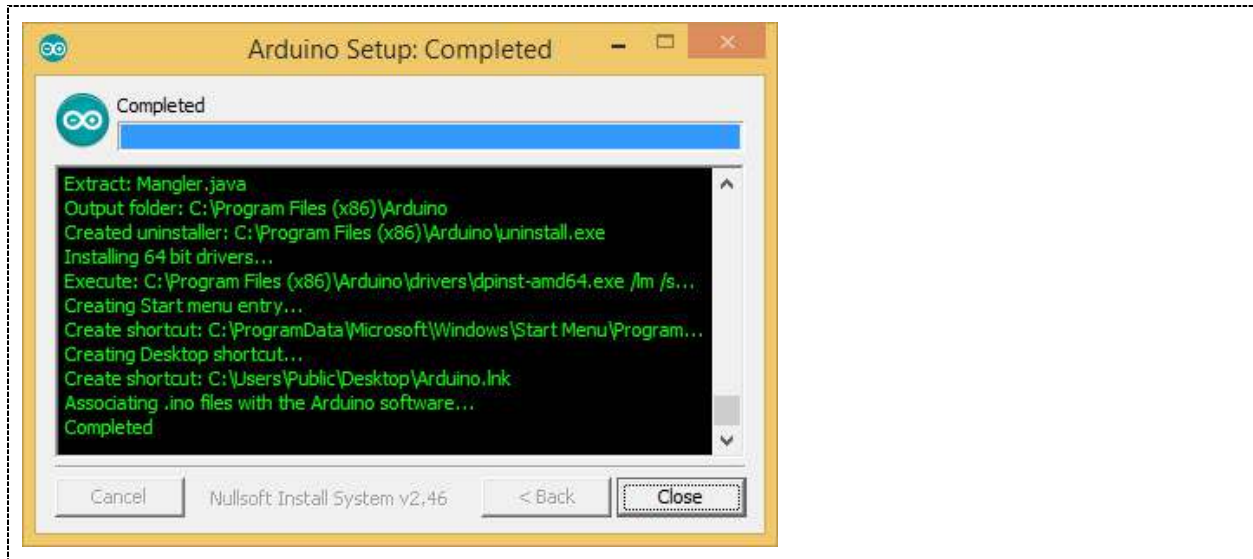
เลือกโฟลเดอร์ติดตั้งโปรแกรม หากไม่ต้องการแก้ไขคลิกปุ่ม Install ได้เลย



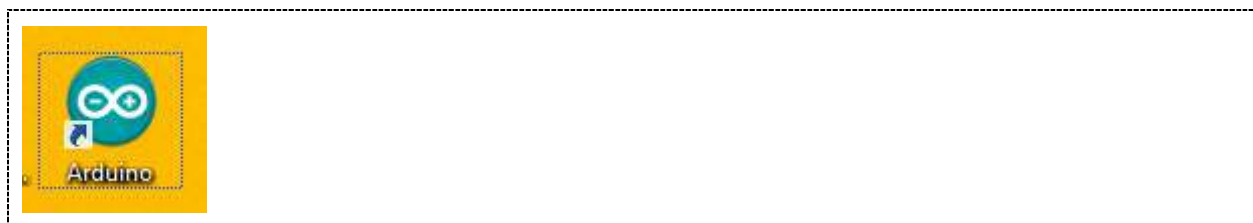
รอจนกว่าโปรแกรมจะติดตั้งเสร็จสิ้น



เมื่อขึ้นคำว่า Completed หมายถึงการติดตั้งเสร็จสมบูรณ์แล้ว
คลิกปุ่ม Close เพื่อปิดโปรแกรมลงไปได้เลย

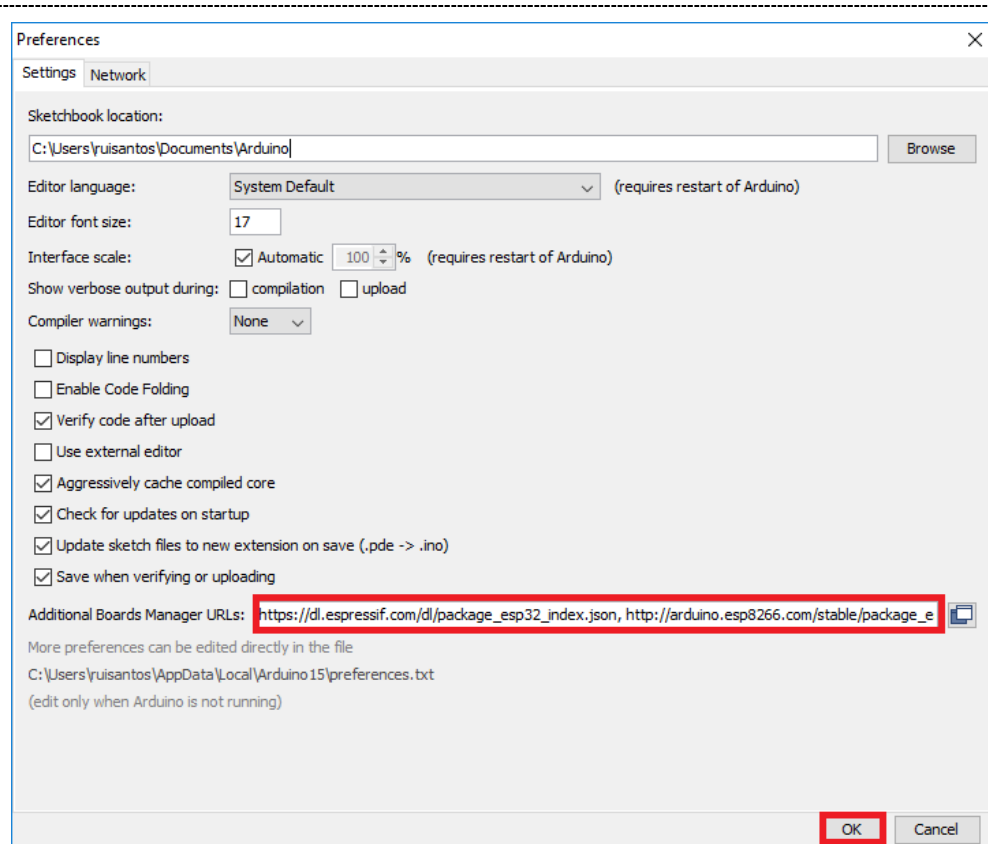


หน้าเดสก์ทอปก็จะมีไอคอนโปรแกรม Arduino ขึ้นมาแล้ว



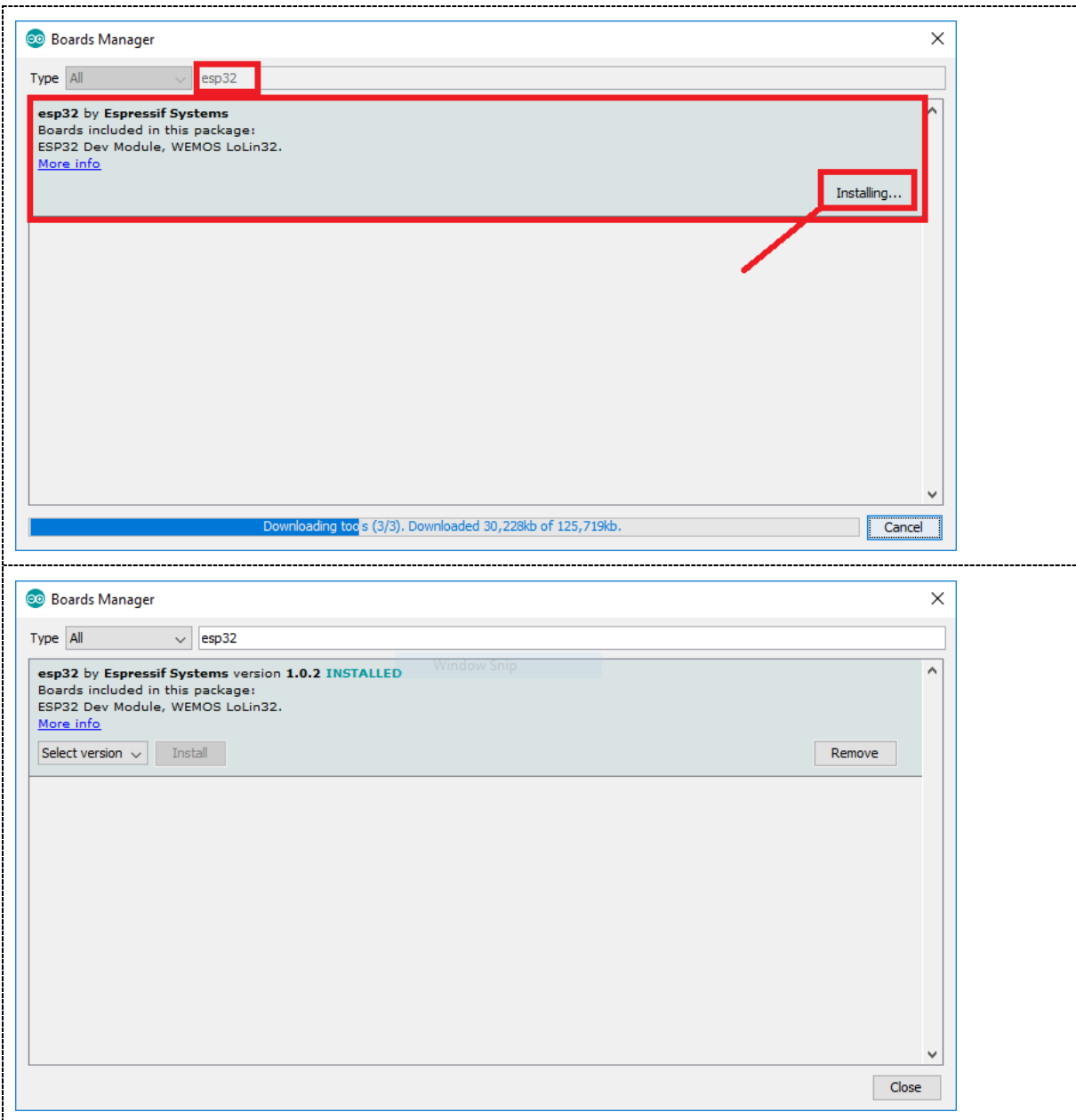
2.3 Add ESP32 Board ตามขั้นตอนดังนี้

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

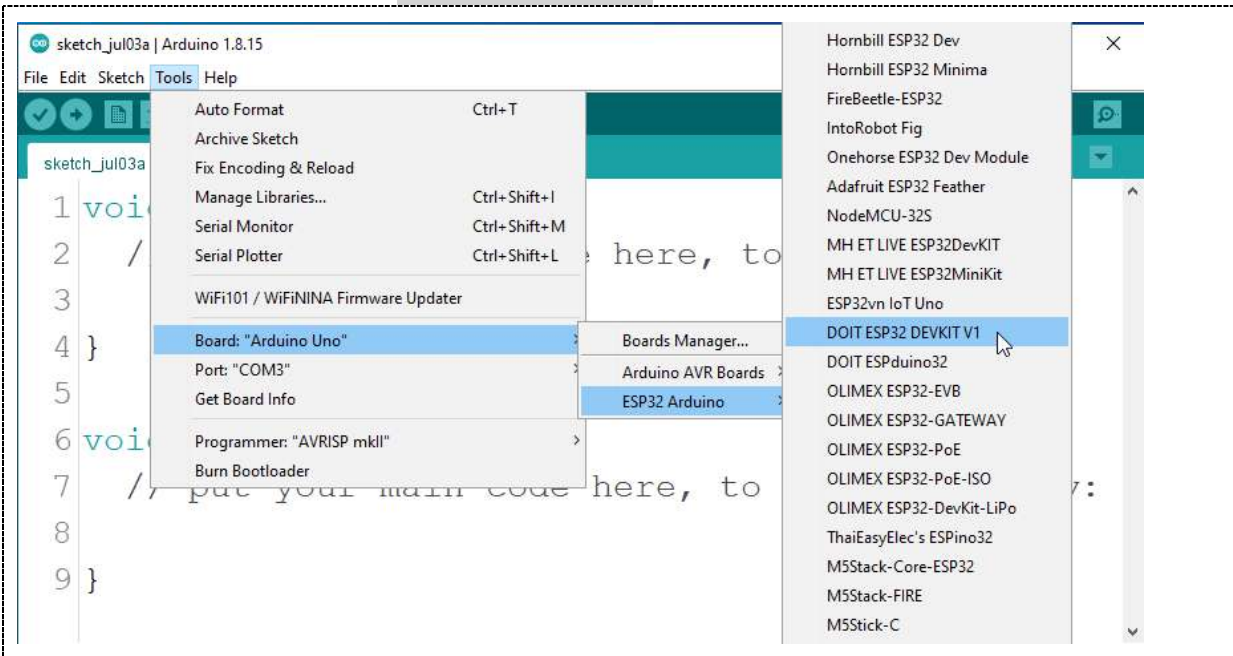


https://dl.espressif.com/dl/package_esp32_index.json,
http://arduino.esp8266.com/stable/package_esp8266com_index.json

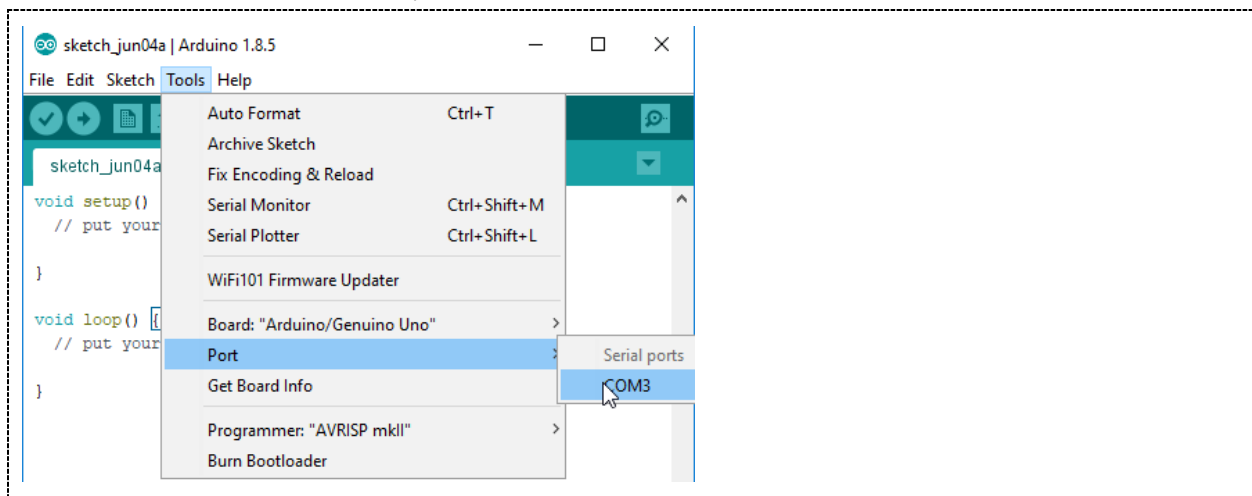
- ไปที่ Menu Tools → Board : “xxxxx” → Board Manager
- เลือกบอร์ด esp32 ของ ESP32 by Espressif Systems → กด Install



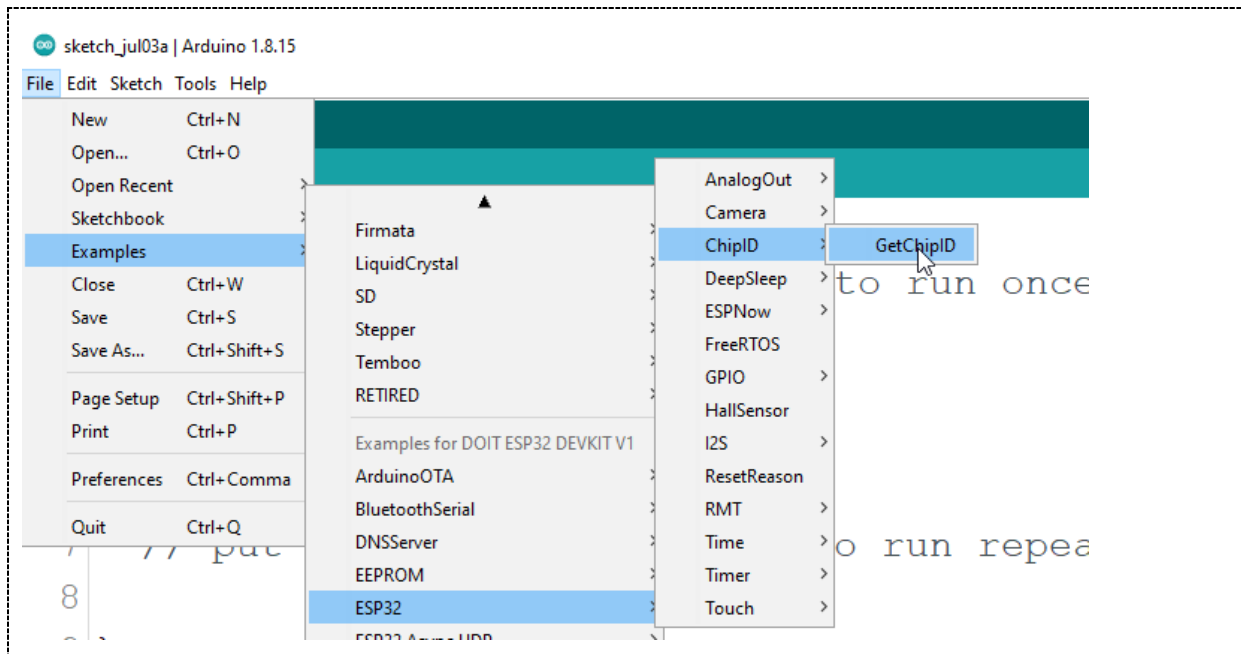
- เมื่อติดตั้งเสร็จ เข้าไปที่ Tools->Boards ลองเช็คดูว่า Arduino IDE รองรับการใช้งานร่วมกับ ESP32 แล้วหรือไม่ ถ้าเจอหัวข้อ **DOIT ESP32 DEVKIT1** แสดงว่าสามารถใช้งานได้ ESP32 ได้แล้ว



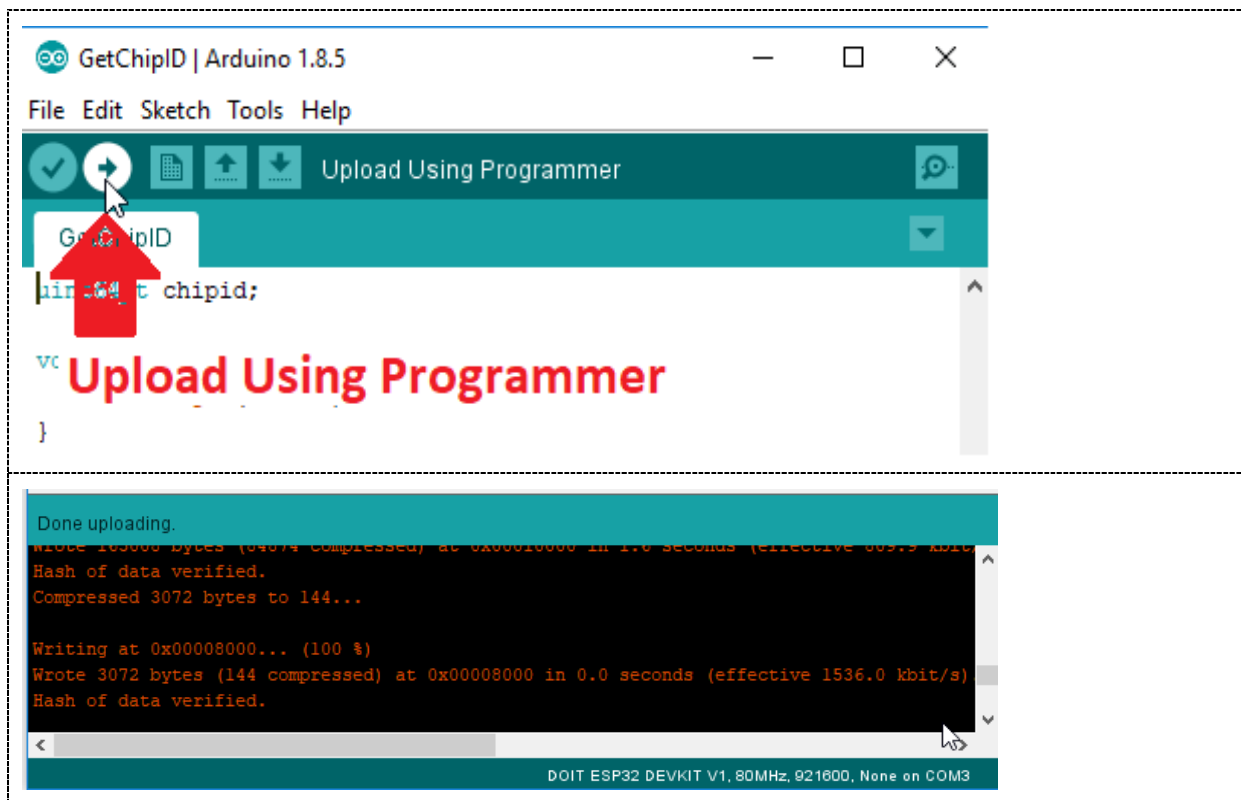
- ต่อบอร์ด ESP32 เข้ากับ PC ผ่าน USB Cable
- เลือกบอร์ดเป็น DOIT ESP32 DEVKIT1
- เลือก Communication port



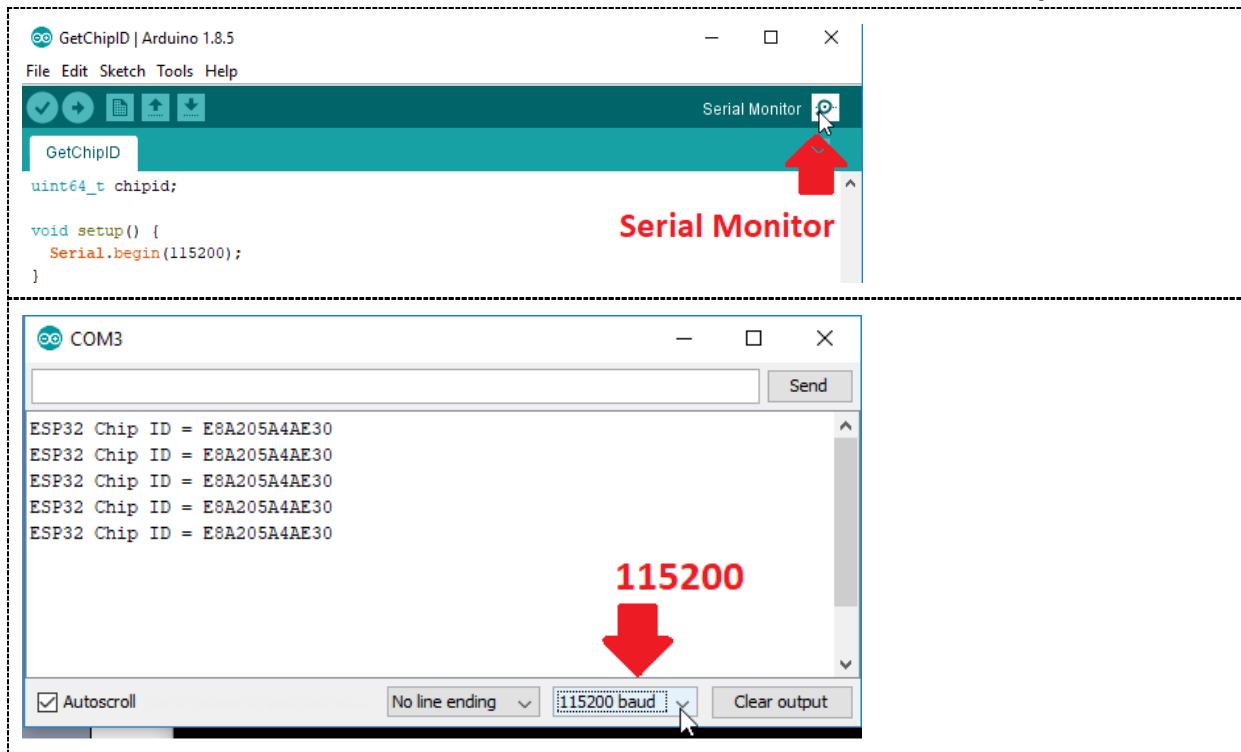
- ทดสอบโปรแกรม File → Example → ESP32 → ChipID → "GetChipID"



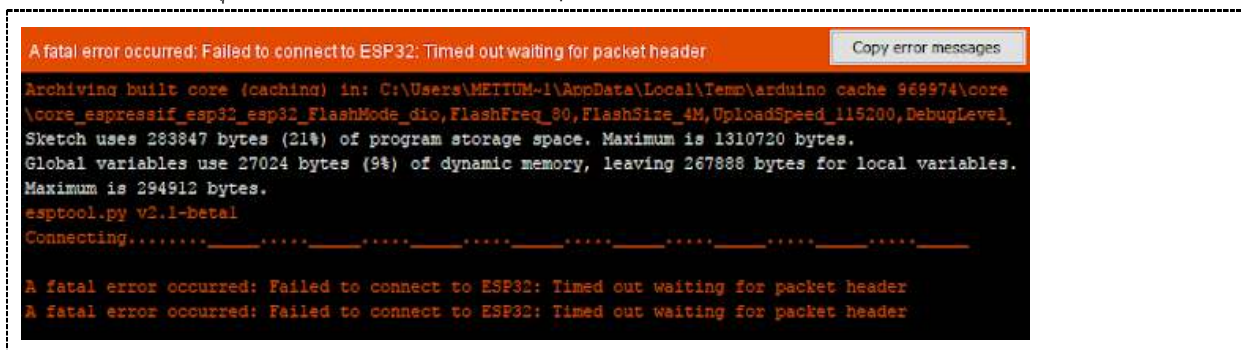
- กด Upload โปรแกรม
- เมื่อโปรแกรมเสร็จเรียบร้อยแล้วจะขึ้นข้อความแสดงในกรอบสีเหลือง



- จากนั้นลองเปิด Serial Monitor ขึ้น ESP32 จะทำการปรี้น ChipID แสดงขึ้นมา ดังรูป



- บอร์ดบางล็อตอาจขึ้น **connecting.....-----.....-----.....-----.....**
- ให้กดปุ่ม BOOT ค้างไว้จนกว่าจะเริ่ม Upload



3 การทดลอง

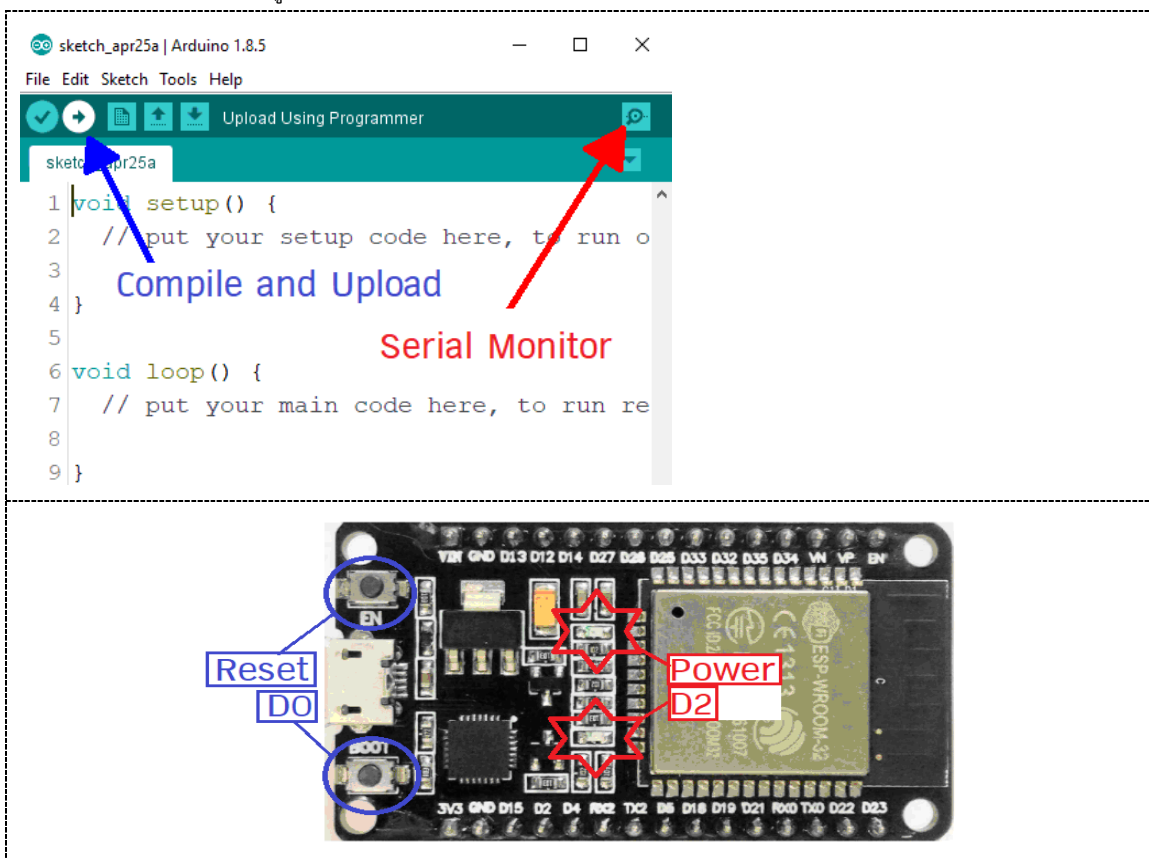
การทดลองที่ 1 ทดสอบLED Output

```
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
- เลี้ยวขวา เพื่อ Compile and Upload
- แฉ่นขยาย เพื่อดูหรือสั่งงานผ่าน Serial Monitor



การทดลองที่ 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 3 ทำความเข้าใจรูปแบบข้อมูลของ 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 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 5 ทดสอบการรับเลขทศนิยมผ่าน 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

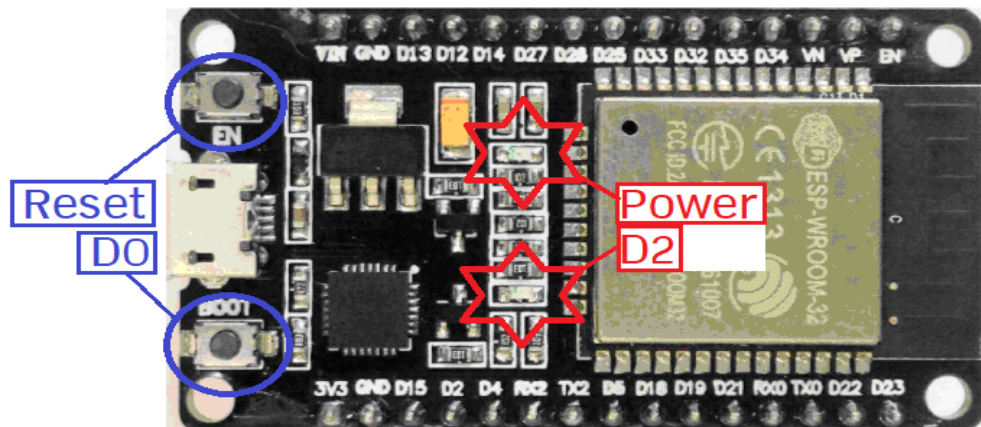
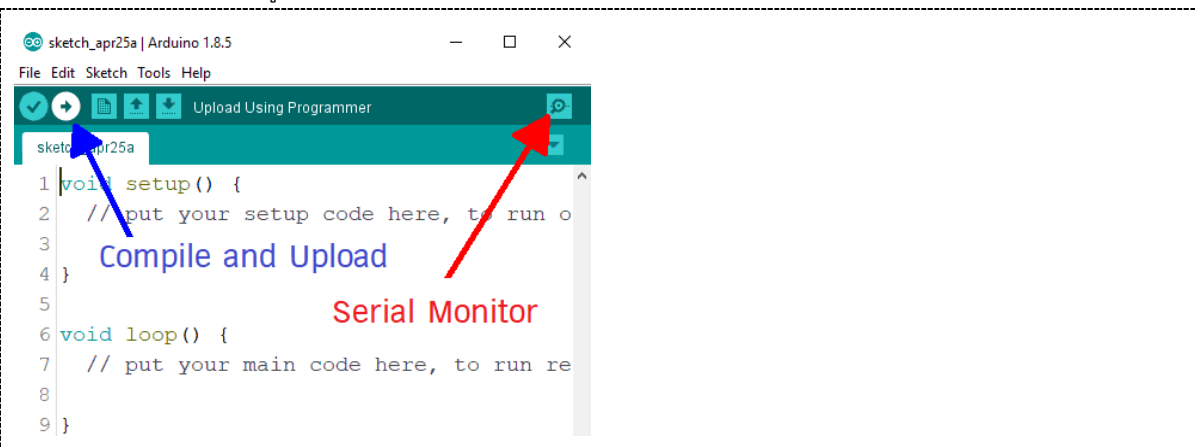
การทดลองที่ 6 LED Blink

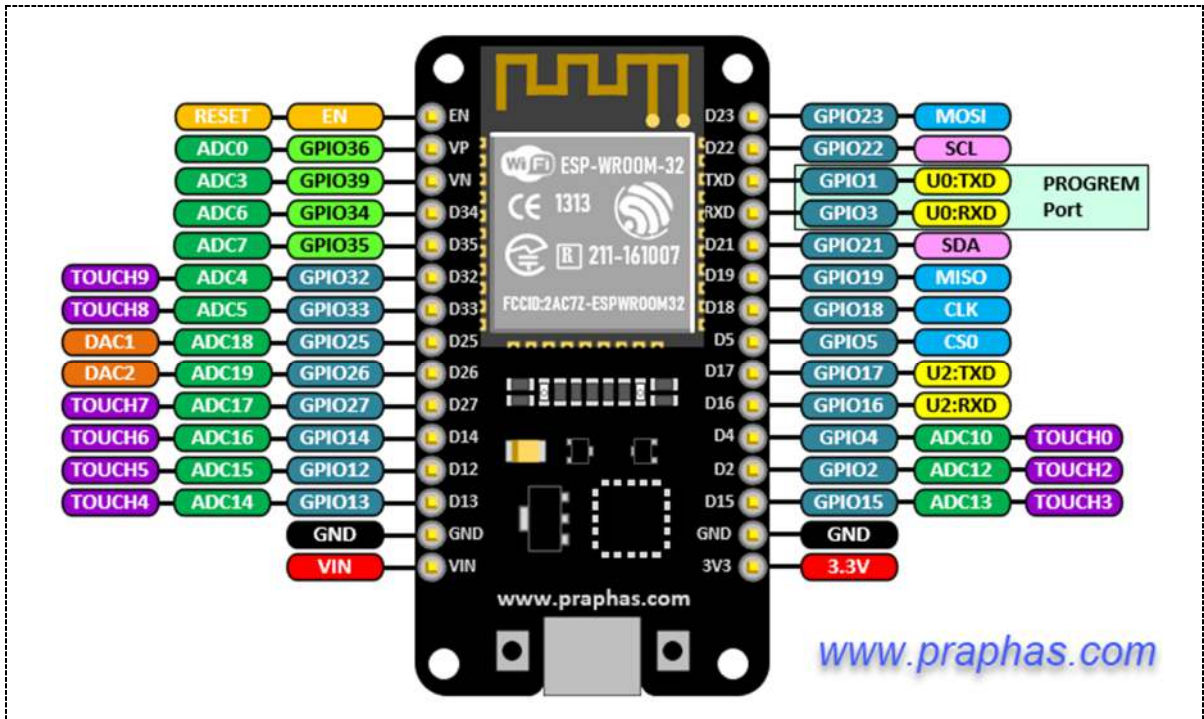
- Tools → Board → DOIT ESP32 DEVKIT V1
- Tools → Port → COM xx
- ทดสอบการทำงานของโปรแกรมไฟกระพริบ File → Example → Basic → Blink

```
// 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
- แนวนอน ขยาย เพื่อดูหรือสั่งงานผ่าน Serial Monitor





การทดลองที่ 7 – 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(2, OUTPUT);
}
```

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

```
// Example_102c – Single LED definid 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
```

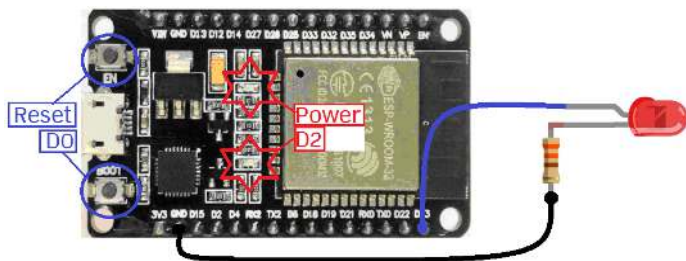
```
int nloop = 24;
char DispBuff[] = {1, 1, 0, 0, 1, 1, 1, 0, 1, 1, 0, 0, 1, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0};
```

```
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

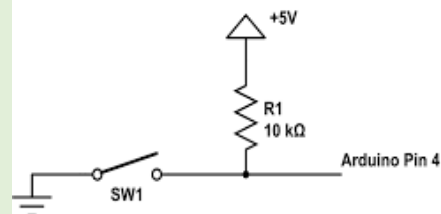


การทดลองที่ 8a – Input Switch

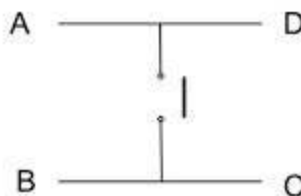
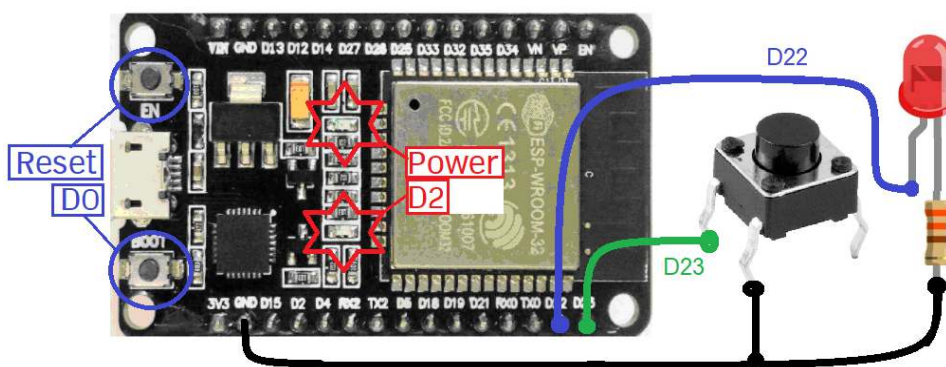
```
// Example_103a - Input Switch
#define pushButton 0

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

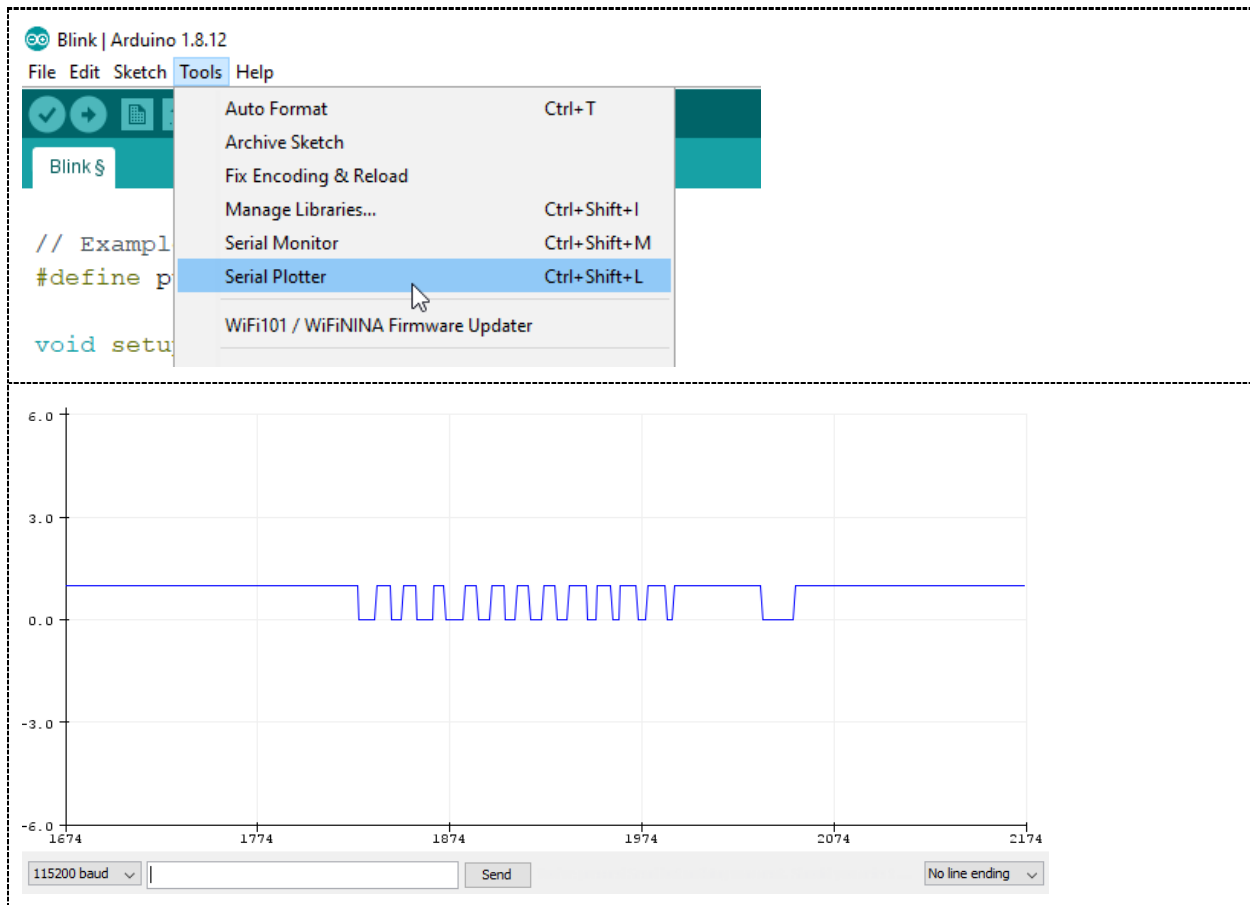
void loop() {
  int swRead = digitalRead(pushButton);
  Serial.println(swRead);
  delay(10);
}
```



- External Pull up
- Internal Pull up



ดูการทำงานเมื่อกดสวิตซ์ที่ Tools → Serial Plotter



การทดลองที่ 8b – Input Switch 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);
}
```

การทดลองที่ 9 – 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++;

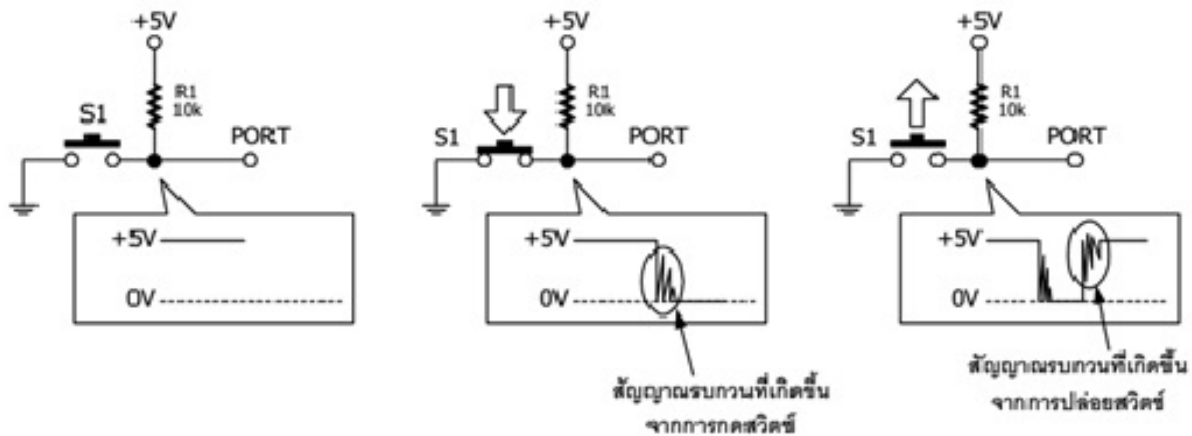
Serial.println(Counter);

while (digitalRead(pushButton) == LOW);

delay(20);

}

}

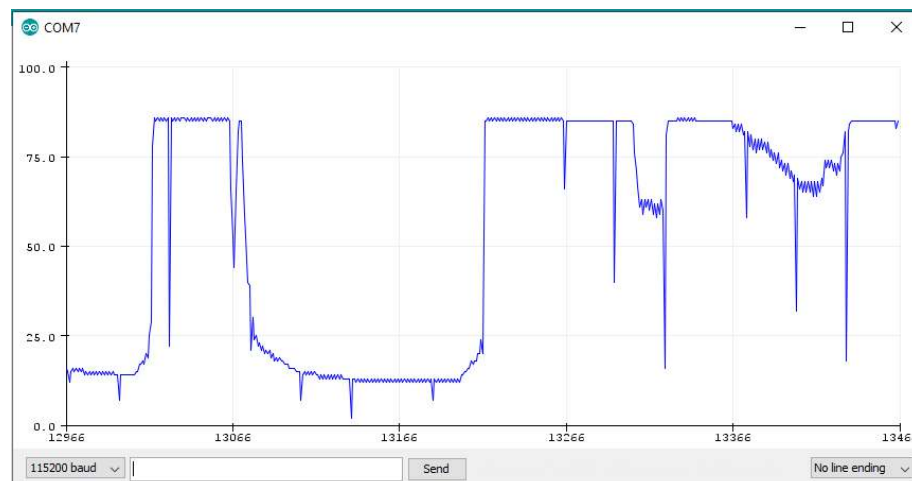
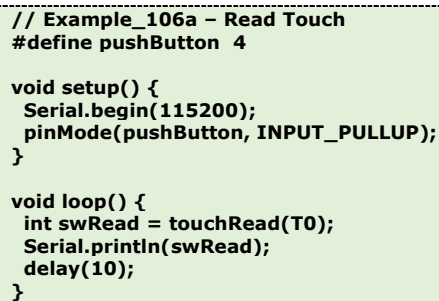


การทดลองที่ 10 – Input Switch กดติด กดดับ

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

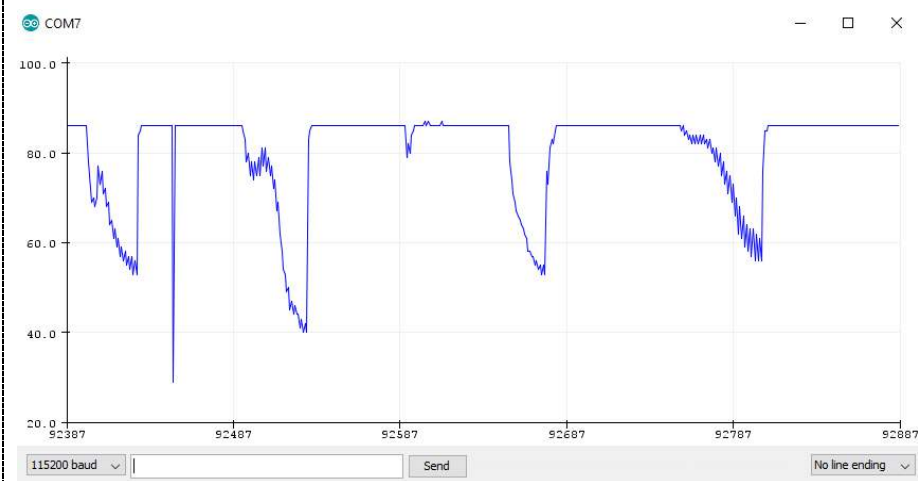


ฟังก์ชันอ่านค่าจากเซนเซอร์สัมผัส Touch pin: หมายเลขขาพอร์ตของเซอร์เซอร์สัมผัสที่ต้องการอ่านค่า เช่น x=touchRead(T0); หมายถึงอ่านค่าจากขา TOUCH0 เก็บไว้ในตัวแปร x

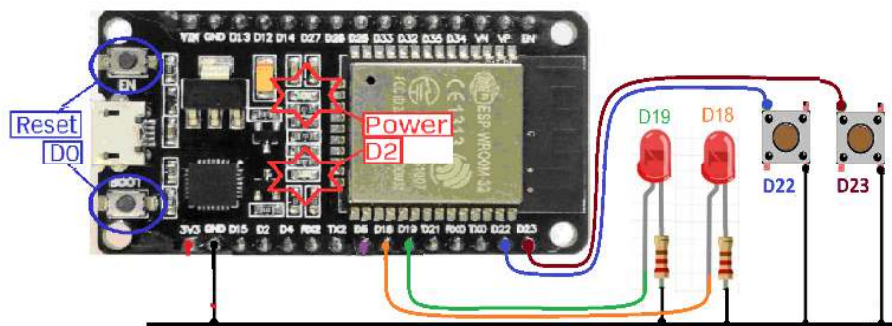
```
// Example_106B - Read Touch Control On/Off
#define LEDPin 22 //(or D2 BuiltIn LED)
#define pushButton 4
#define RefTouchSw 75
int Count, buttonState = 1;

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);
  }
}
```

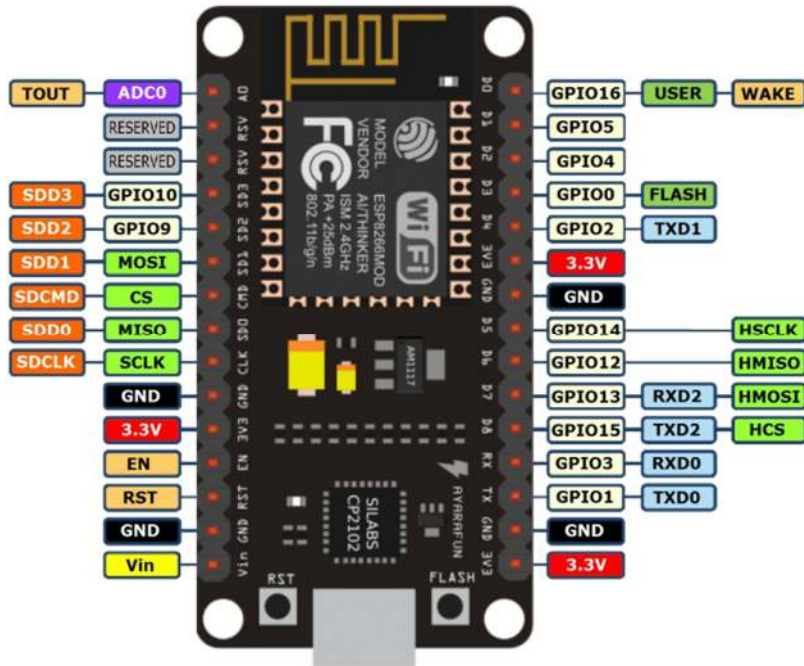
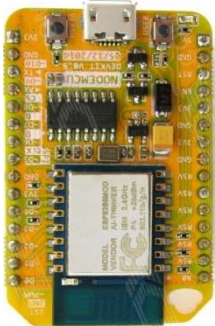




- หากต้องการให้ใช้ 1 สวิตช์ ควบคุม 1 LED แบบกดติด-กดดับ จำนวน 2 วงจรจะต้องวางจระและเขียนโปรแกรมอย่างไร {SW-D22 -- LED-D19, SW-D23 -- LED-D18}



4. ESP-8266, ESP-32 and M5-Stack

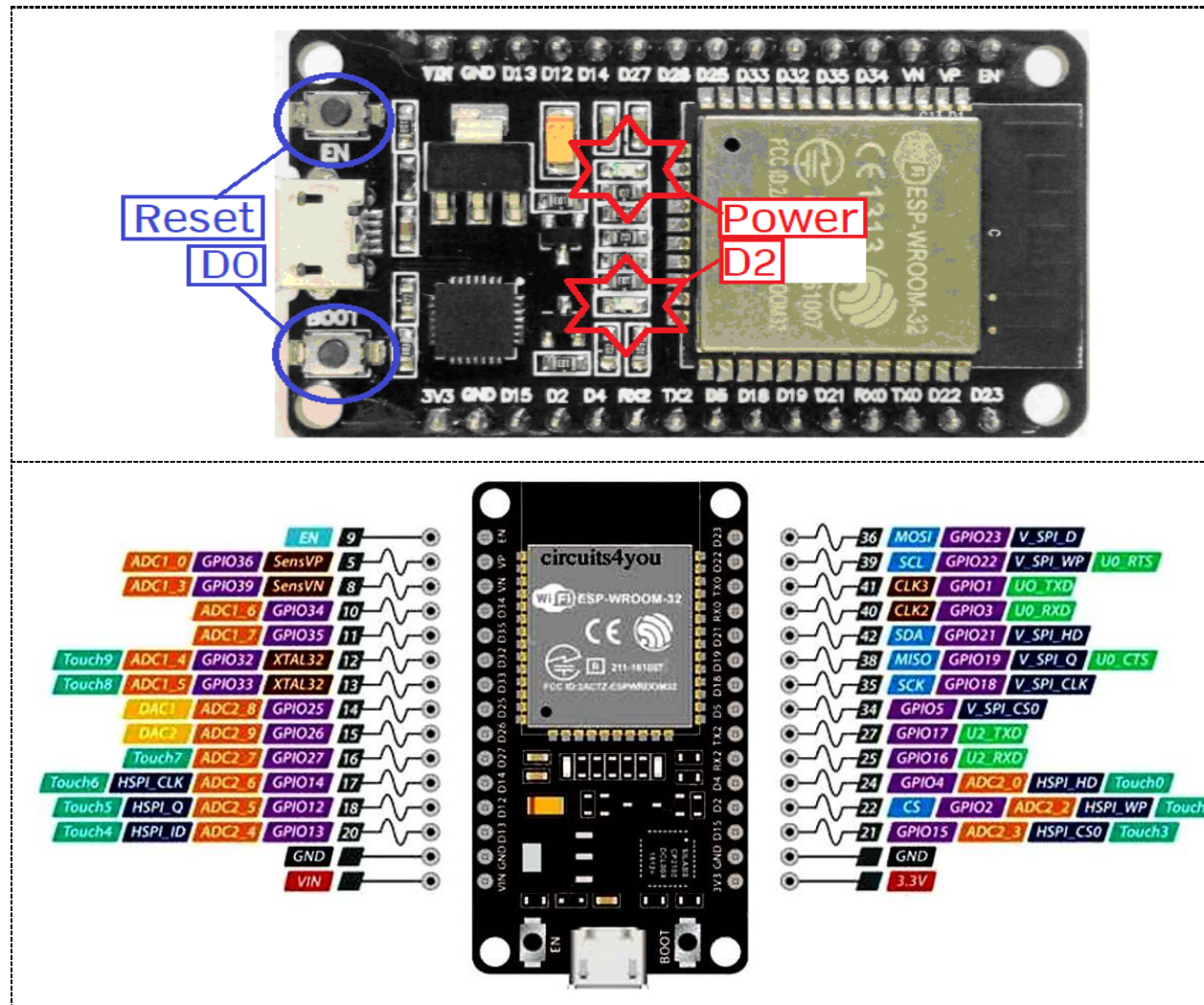
4.1 NodeMCU - ESP8266

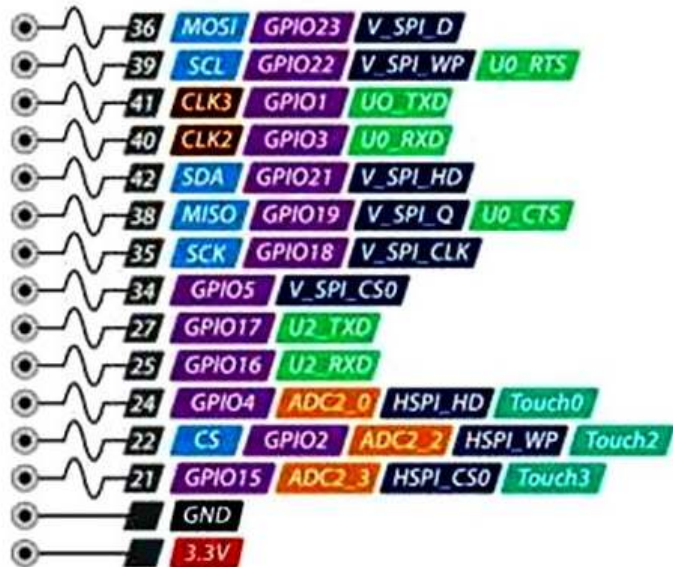
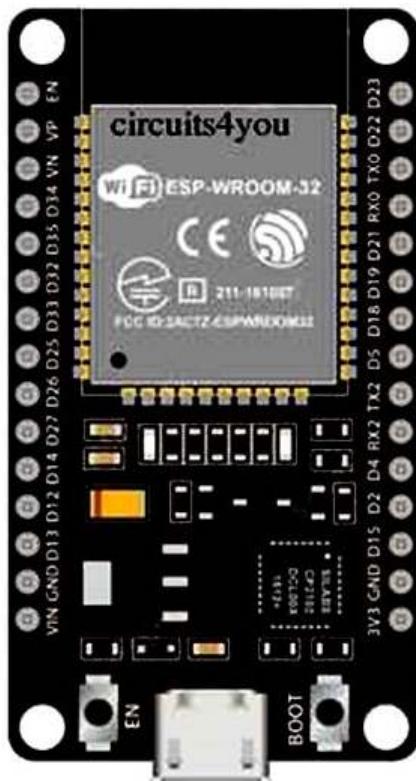
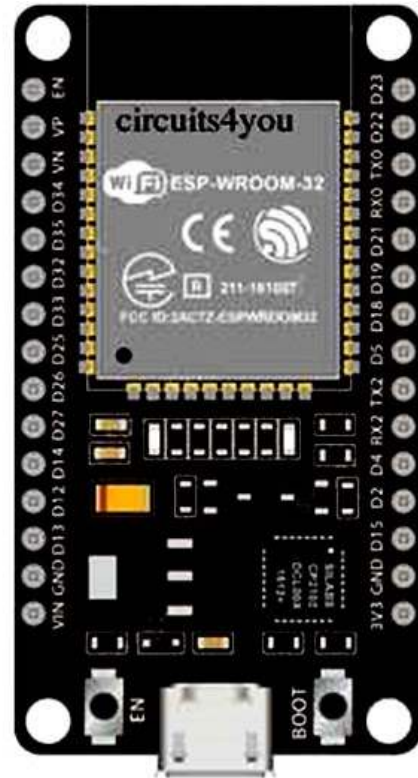
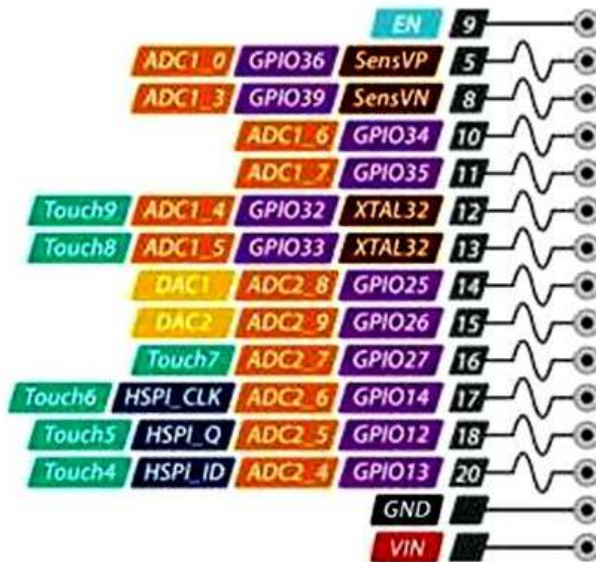
		
NodeMCU V0.9	NodeMCU V1.0	NodeMCU V3.0
ESP-12 Module	ESP-12E Module	ESP-12E Module
Arduino IDE = Node0.9	Arduino IDE = Node1.0	Arduino IDE = Node1.0
		
<ul style="list-style-type: none"> • USB-SERIAL CH340 • ใช้ Serial LED ที่ GPIO1 ได้ แต่ต้องไม่ใช้พร้อม Serial Communication • มี LED Buid in ที่ GPIO16“BUILTIN_LED”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

4.2 ESP-32 Dev Kit V1 Board

ก่อนหน้านี้มีการใช้งาน NodeMCU V2 ซึ่งเป็น ESP8266 อย่างแพร่หลาย แต่ด้วยเสียงลือเสียงเล่าอ้างเรื่องความสามารถของ 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 ใช้ไมโคร ESP-WROOM-32 มาทำการทดสอบ

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



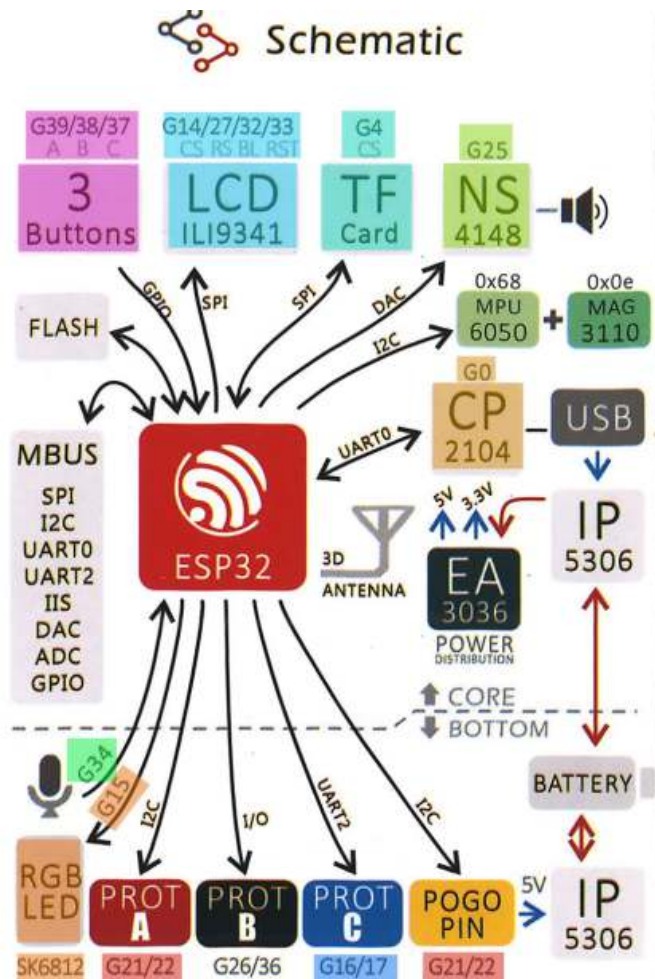


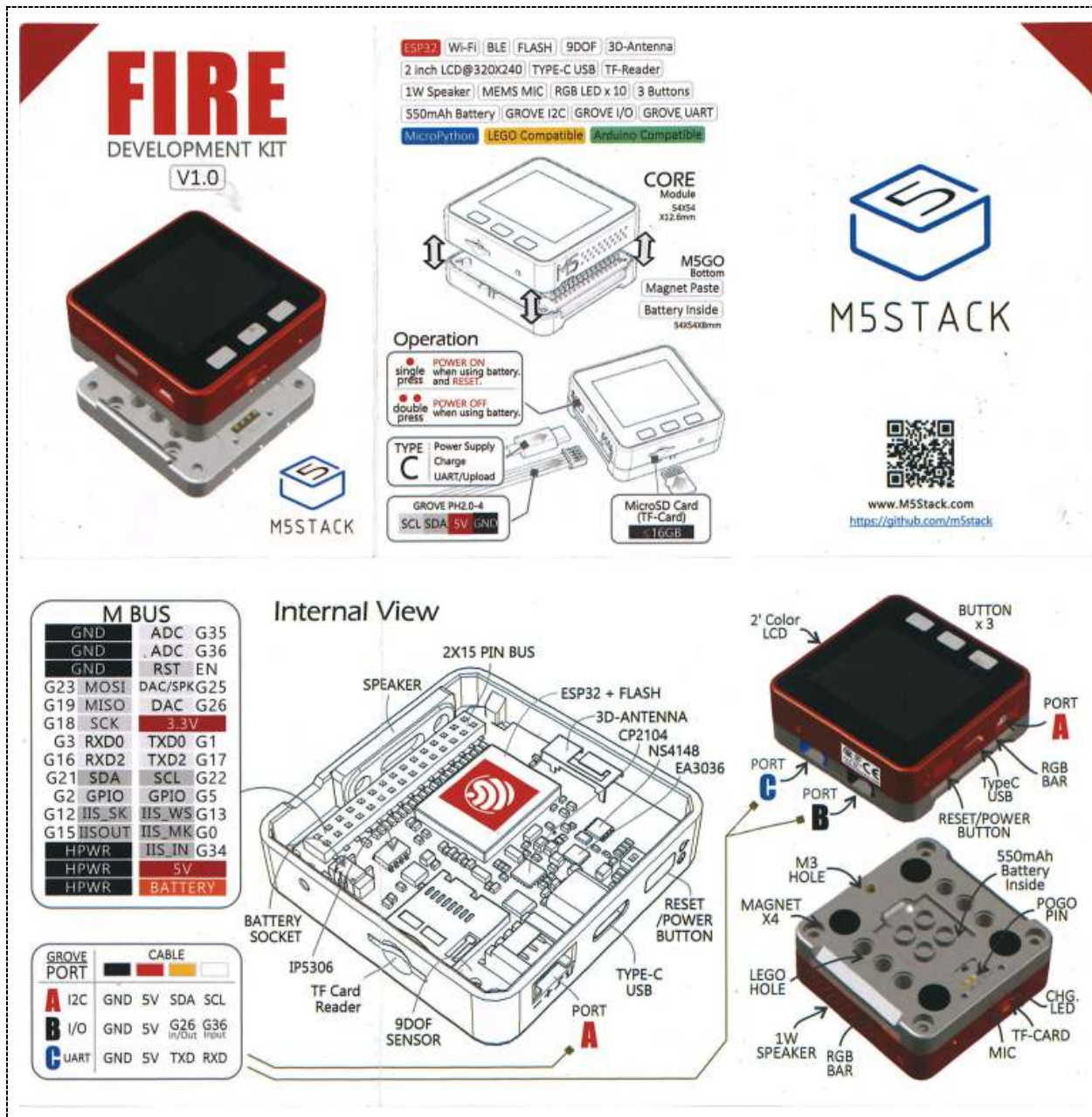
4.3 M5-Stack



M BUS			
GND	ADC	G35	
GND	ADC	G36	
GND	RST	EN	
G23	MOSI	DAC/SPK	G25
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		CABLE	
A	I2C	GND 5V	SDA SCL
B	I/O	GND 5V	G26 G36 In/Out Input
C	UART	GND 5V	TXD2RXD2





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

M5Stack คืออุปกรณ์ที่ใช้สร้างตัวต้นแบบของอุปกรณ์ WiFi+Bluetooth โดยใช้ชิป ESP32 ของ Espressif โปรแกรมได้ทั้ง Arduino, Micro-Python หรือ Web-IDE ตัวบอร์ดหลัก M5 Core จะอยู่ด้านบนติดกับจอ LCD จะมีเสาอากาศแบบ 3D มาให้ (3D Antenna), Grove connector สำหรับ I2C, microSD card socket, JST battery socket, สวิตช์ เปิด/ปิด/reset, 3ปุ่มกดบนหน้าจอ

บอร์ดด้านหลังจะเป็น M5 Faces ซึ่งทำหน้าที่ต่อเข้ากับแป้นพิมพ์ Panel ต่างๆ และยังมีแบตเตอรี่ LiPo ขนาด 650mAh ที่สามารถชาร์ตผ่าน charging Base ได้

4.4 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-e4BRCwqeWkv8TWqOoBEiQAMocbPytjm40atWOSYlaQI7V0O0p-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.arduitronics.com/product/1329/doit-esp32-development-board-esp-wroom-32-wifiblueetooth-esp-32s>
- Read This <https://www.mcucity.com/product/1144/doit-esp32-wifiblueetooth-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>