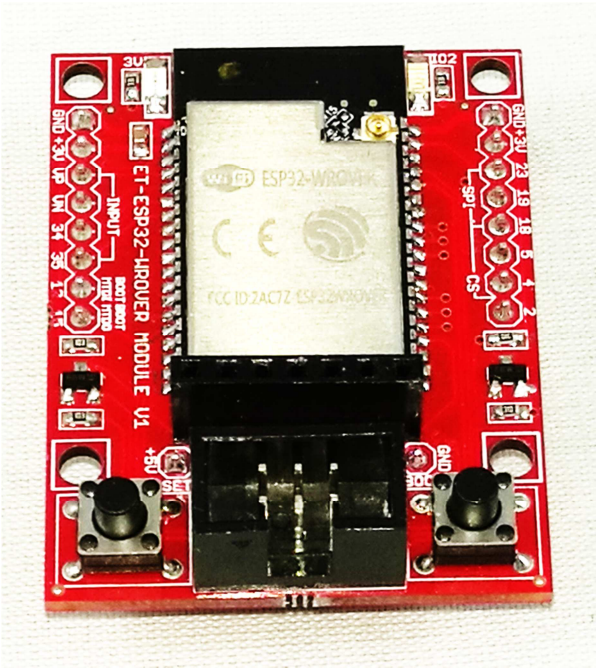


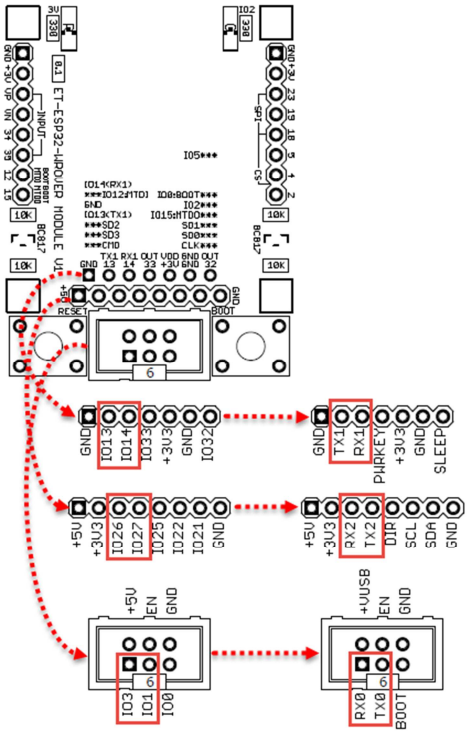
ET-ESP32 WROVER MODUE V1 เป็นโมดูล MCU ตระกูล ESP32 ของ Espressif Systems โดยเลือกใช้โมดูล รุ่น ESP32 WROVER-I เป็นชุด MCU ประจำบอร์ด

- **Memory**
  - 4MB SPI Flash(32Mbits SPI flash) / 8MB PSRAM(64Mbits PSRAM)
- **Wi-Fi**
  - ช่วงความถี่การทำงาน 2.4 – 2.5 GHz
  - รองรับโพรโตคอล 802.11 b/g/n (802.11n up to 150 Mbps)
  - A-MPDU and A-MSDU aggregation and 0.4 uS guard interval support
- **Bluetooth**
  - รองรับโพรโตคอล Bluetooth 4.2 BR/EDR และ BLE
  - NZIF receiver with -98 dBm sensitivity
  - Class-1, class-2 and class-3 transmitter
  - AFH
  - CVSD and SBC
- **Hardware**
  - โมดูลรองรับอินเตอร์เฟส SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, I2C, IR
  - Hall Sensor และ Temperature Sensor บนชิพ
  - 40 MHz crystal บนบอร์ด
  - แรงดันการทำงาน 2.3 ถึง 3.6 โวลต์
  - กระแสไฟฟ้าเฉลี่ยในการทำงาน 80mA
- **Software**
  - รองรับ Wi-Fi โหมด Station, SoftAP, SoftAP+Station, P2P
  - รองรับ WPA / WPA2 / WPA2-Enterprise / WPS
  - รองรับการเข้ารหัส AES / RSA / ECC / SHA
  - ช่องทางการอัพเกรดเฟิร์มแวร์ UART / OTA (via network)
  - รองรับเครือข่าย IPv4, IPv6, SSL, TCP / UDP / HTTP / FTP / MQTT

การจัดสรรใช้งาน **USART**

บอร์ด ET-ESP32 RS485 ออกแบบวงจรให้มีส่วนของพอร์ตสื่อสารอนุกรม USART จำนวน 3ช่อง โดยใช้สำหรับ Download และ Debug การทำงานของระบบ 1ช่อง(USART0) และใช้เชื่อมต่อกับโมดูล NB-IoT หรืออุปกรณ์อื่นๆที่สื่อสารผ่าน USART แบบ TTL ได้(USART1) และเชื่อมต่อกับอุปกรณ์ที่ติดต่อสื่อสารแบบอนุกรม RS422/485 (USART2) โดยเลือกจัดสรรขาสัญญาณ I/O สำหรับเชื่อมต่อกับ USART ดังนี้

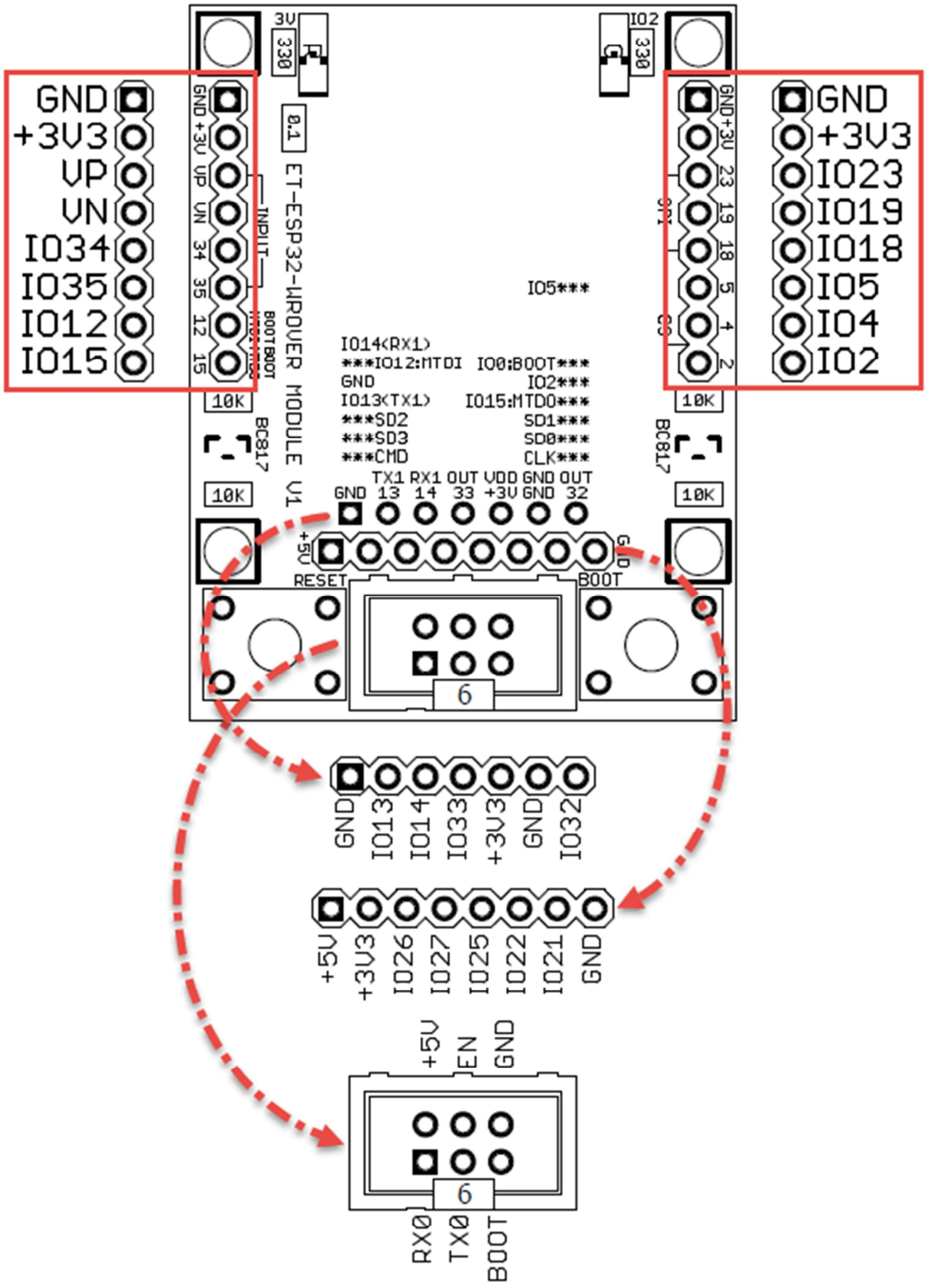
- USART0(USART-TTL) ใช้เชื่อมต่อกับ ET-USB USART/TTL สำหรับ Upload Code และ Debug แสดงการทำงาน หรือ รับส่งข้อมูลได้ตามต้องการ
    - IO3 เป็น RX0
    - IO1 เป็น TX0
- USART1(USART-TTL) ใช้สำหรับรับส่งข้อมูลตามการเขียน firmware ควบคุมการรับส่ง
    - IO14 เป็น RX1
    - IO13 เป็น TX1
- USART2(RS422/485) ใช้สำหรับรับส่งข้อมูลตามการเขียน firmware ควบคุมการรับส่ง
    - IO26 เป็น RX2
    - IO27 เป็น TX2
    - IO25 เป็นสัญญาณ DIR(Direction)



สำหรับในกรณีที่พัฒนาโปรแกรมด้วยแพลตฟอร์มของ Arduino นั้น การกำหนด Pin I/O สำหรับใช้เชื่อมต่อกับ Hardware USART ของโมดูล ESP32-WROVER กับบอร์ด ET-ESP32 RS485 ทำได้ดังนี้

```
#include <HardwareSerial.h>


#define SerialDebug  Serial                  // USB Serial(Serial0)
#define SerialNBIOT_RX_PIN    14
#define SerialNBIOT_TX_PIN    13
#define SerialNBIOT  Serial1                 // Serial1(IO13=TXD,IO14=RXD)
#define SerialRS485_RX_PIN    26
#define SerialRS485_TX_PIN    27
#define SerialRS485  Serial2                 // Serial2(IO27=TXD,IO26=RXD)


void setup()
{
  SerialDebug.begin(115200);
  while(!SerialDebug);

  SerialNBIOT.begin(115200, SERIAL_8N1, SerialNBIOT_RX_PIN, SerialNBIOT_TX_PIN);
  while(!SerialNBIOT);

  SerialRS485.begin(115200, SERIAL_8N1, SerialRS485_RX_PIN, SerialRS485_TX_PIN);
  while(!SerialRS485);

  ...
}

void loop()
{
  ...
}
```

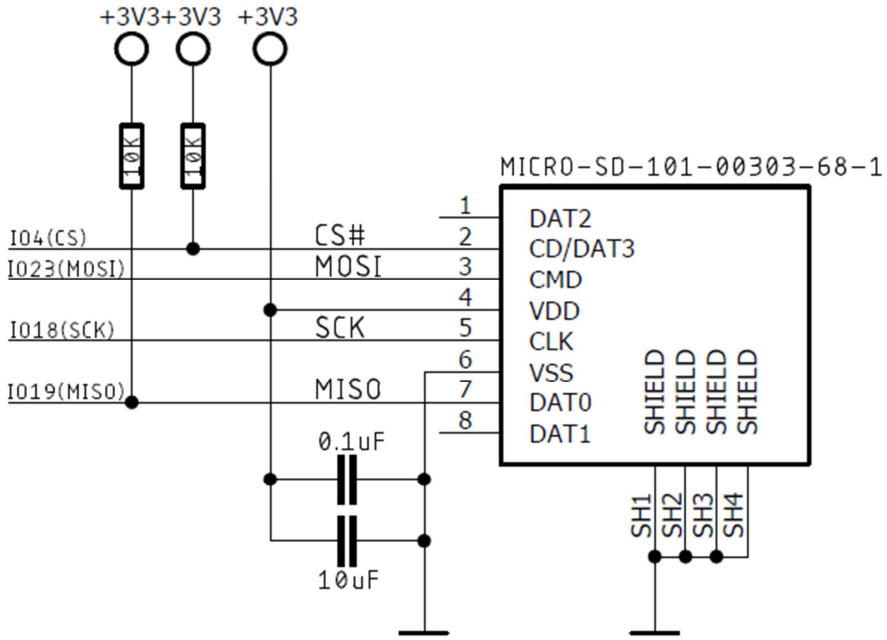
แสดงตัวอย่างการ Initial USART ของบอร์ด ET-ESP32-RS485



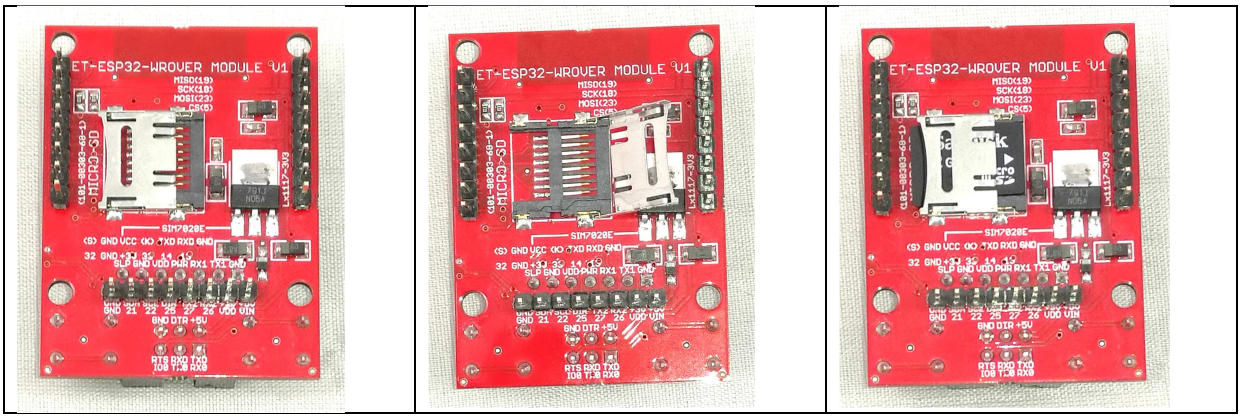
รูปแสดง การจัดเรียงสัญญาณ ของบอร์ด ET-ESP32 WROVER MODULE V1

การติดตั้ง Micro SD กับบอร์ด

บอร์ด ET-ESP32 WROVER MODULE V1 ได้ออกแบบให้มี Socket Micro SD สำหรับติดตั้งใช้งานกับการ์ดหน่วยความจำแบบ Micro SD Card ไว้ใต้แผงวงจรของบอร์ด โดยเชื่อมต่อวงจรของ ESP32 แบบ SPI Mode ผ่านทางพอร์ต SPI ดังรูป



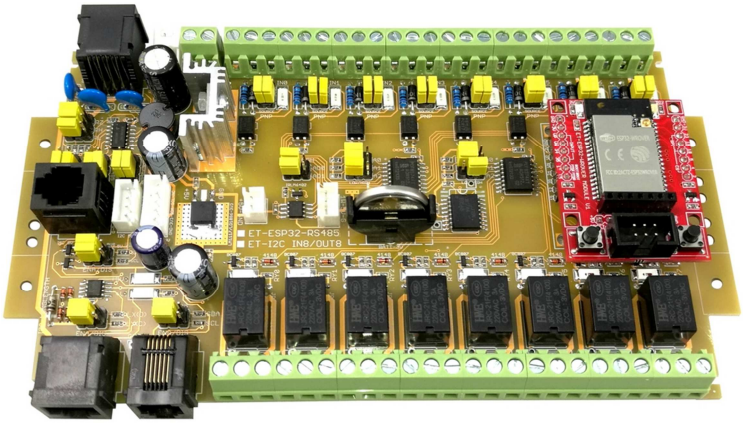
รูปแสดง วงจรการเชื่อมต่อกับ **Micro SD** ด้วย **SPI** พอร์ต

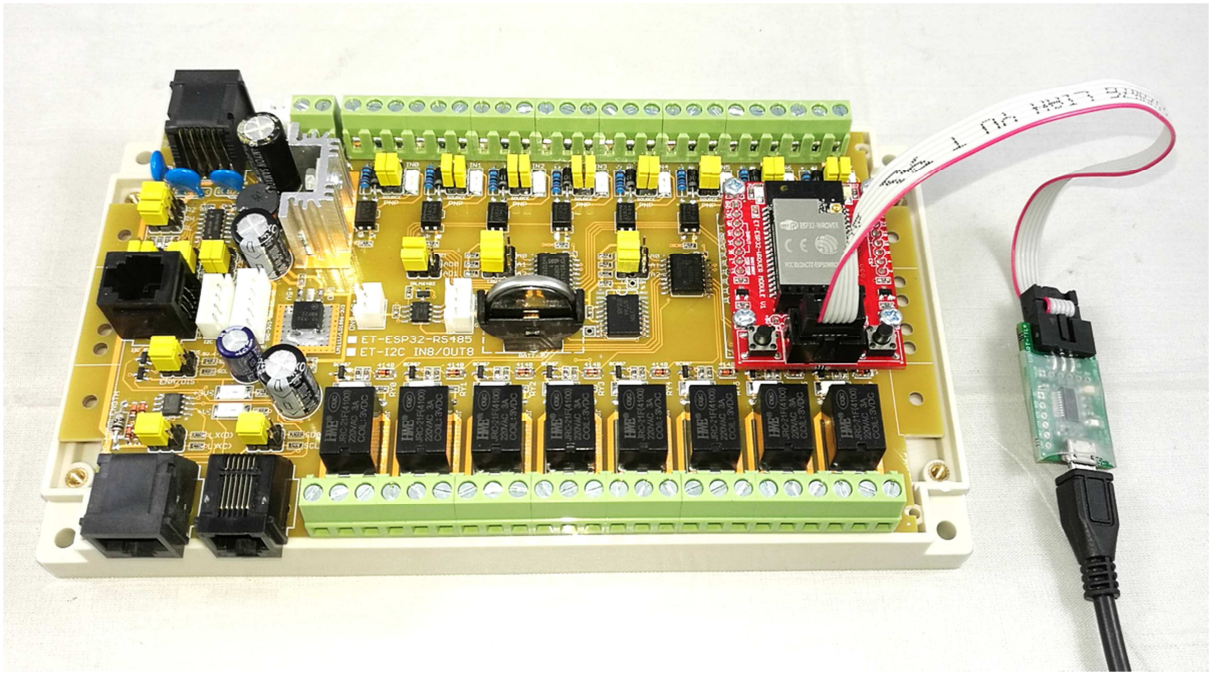


รูปแสดงการติดตั้ง การ์ด **Micro SD** กับ **Socket**

การ Upload Program

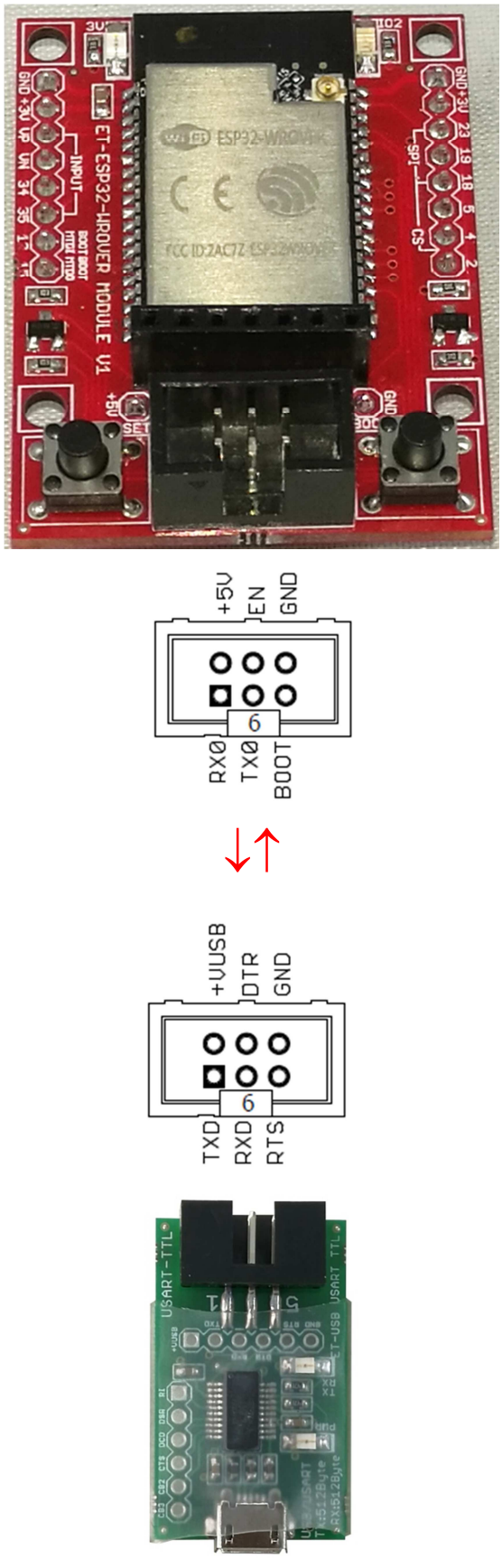
ในการ **Upload Program** ให้บอร์ดจะต้องกระทำผ่านอุปกรณ์ภายนอกที่ทำหน้าที่เป็น **USB/USART** แบบ **TLL** โดยเชื่อมต่อสัญญาณผ่านทางขั้ว **6PIN IDE** ด้วยสายแพร์ดังตัวอย่าง





*** หมายเหตุ ***

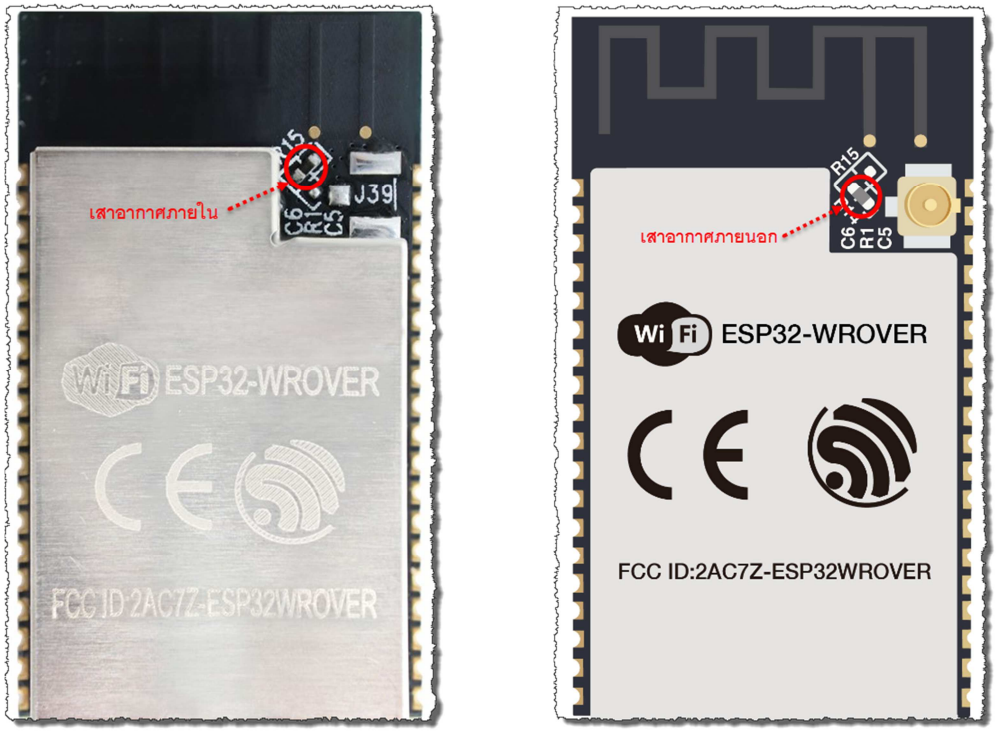
ในกรณีติดตั้งบอร์ด **ET-ESP32 WROVER MODULE V1** ใช้งานร่วมกับบอร์ด **ET-ESP32 RS485** ผู้ใช้ต้องทำการจ่ายไฟเลี้ยงวงจรให้กับบอร์ด **ET-ESP32 RS485** ให้เรียบร้อยก่อนที่จะเสียบสาย **USB** เพราะกระแสไฟจากแหล่งจ่ายไฟของ **USB** มีไม่มากพอที่จะเลี้ยงอุปกรณ์ในวงจรของบอร์ดได้เพียงพอ อาจทำให้พอร์ต **USB** เสียหายได้



รูปแสดงการเชื่อมต่อสัญญาณเพื่อ Upload Program โดยใช้ ET-USB USART/TTL

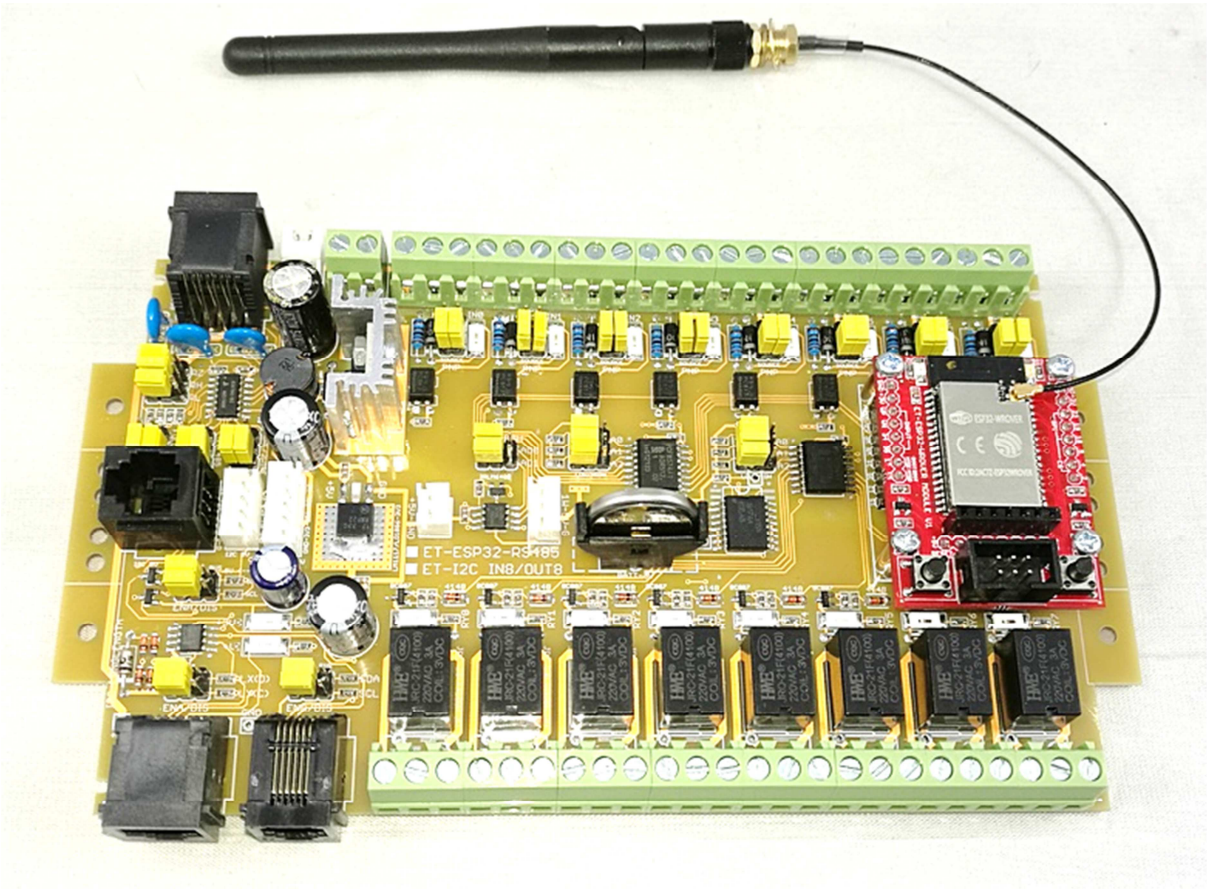
การเลือกใช้ เสาอากาศ ภายใน หรือ ภายนอก

ตามปรกติดแล้ว โมดูล ESP32 WROVER-I จะมีเสาอากาศภายในแบบ PCB มาให้ด้วยแล้ว และสามารถรับส่งข้อมูลผ่านคลื่นความถี่ย่าน 2.4GHz ได้อยู่แล้ว แต่ในกรณีที่ไม่สามารถใช้เสาอากาศภายในบอร์ดได้ เช่น ชิ้นงานมีความจำเป็นต้องติดตั้งในกล่องโลหะซึ่งปิดกั้นสัญญาณความถี่ ผู้ใช้ก็สามารถทำการต่อเสาอากาศจากภายนอกให้กับโมดูลได้ โดยทาง Espressif ผู้ออกแบบและผลิตโมดูลได้ออกแบบวงจรไว้โดยใช้ตัวต้านทาน 0โอห์ม ทำหน้าที่เป็น Jumper สำหรับเลือกใช้เสาอากาศภายนอกหรือภายในเตรียมไว้ให้ โดยโมดูลรุ่น ESP32-WROVER-I มาตรฐานจะทำการบัดกรี R-Jumper เพื่อเลือกเสาอากาศภายนอกไว้ให้โดยผู้ใช้ต้องจัดหาเสาอากาศแบบมีสายต่อผ่านขั้ว SMA มาติดตั้งเพิ่มเติมจึงจะสามารถใช้งานได้ แต่เพื่อความสะดวกทาง อีทีที ได้ทำการบัดกรีเพื่อเชื่อต่อเสาอากาศภายในเตรียมไว้ให้แล้วและก็สามารถต่อเสาอากาศภายนอก สำหรับรับสัญญาณได้ด้วย เพียงแต่ในทางเทคนิคแล้วการใช้เสาอากาศมากกว่า 1เสา อาจเกิดปัญหาการ matching สัญญาณผิดเพี้ยนทำให้ประสิทธิภาพการรับส่งผิดเพี้ยนไปได้จึงควรเลือกจุดต่อเสาอากาศเพียงแบบใดแบบหนึ่งเท่านั้น ถ้าต้องการใช้เสาอากาศภายนอกก็ควรปลดจุดบัดกรีที่ต่อกับเสาอากาศภายในออก ดังรูป



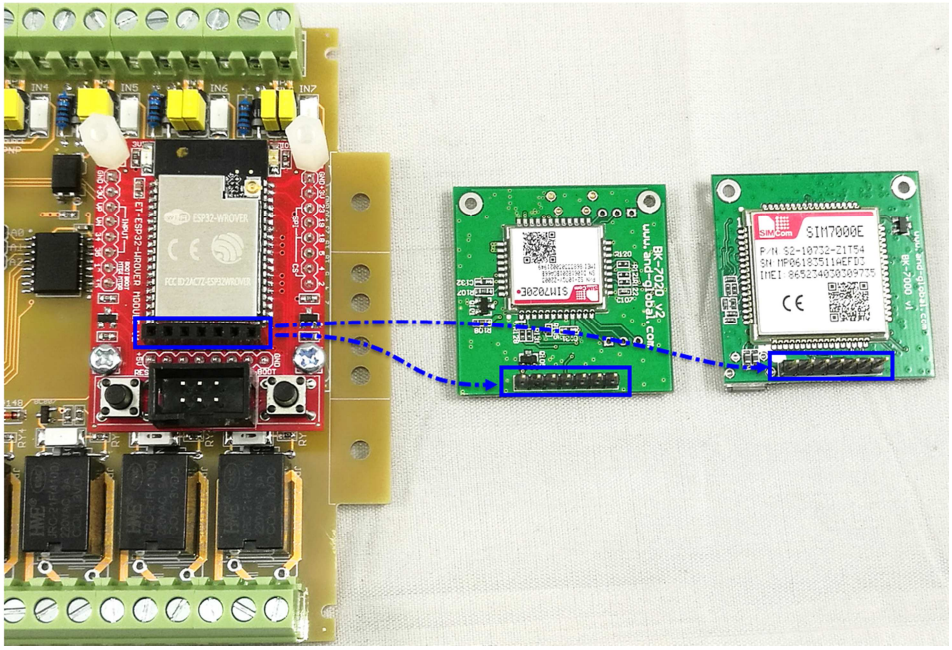
รูปแสดง การบัดกรี Jumper(R 0 โอห์ม) เพื่อเลือกใช้เสาอากาศภายในและภายนอก

รูปแสดง ตัวอย่างเสาอากาศสำหรับใช้ภายนอก รุ่น MT7681

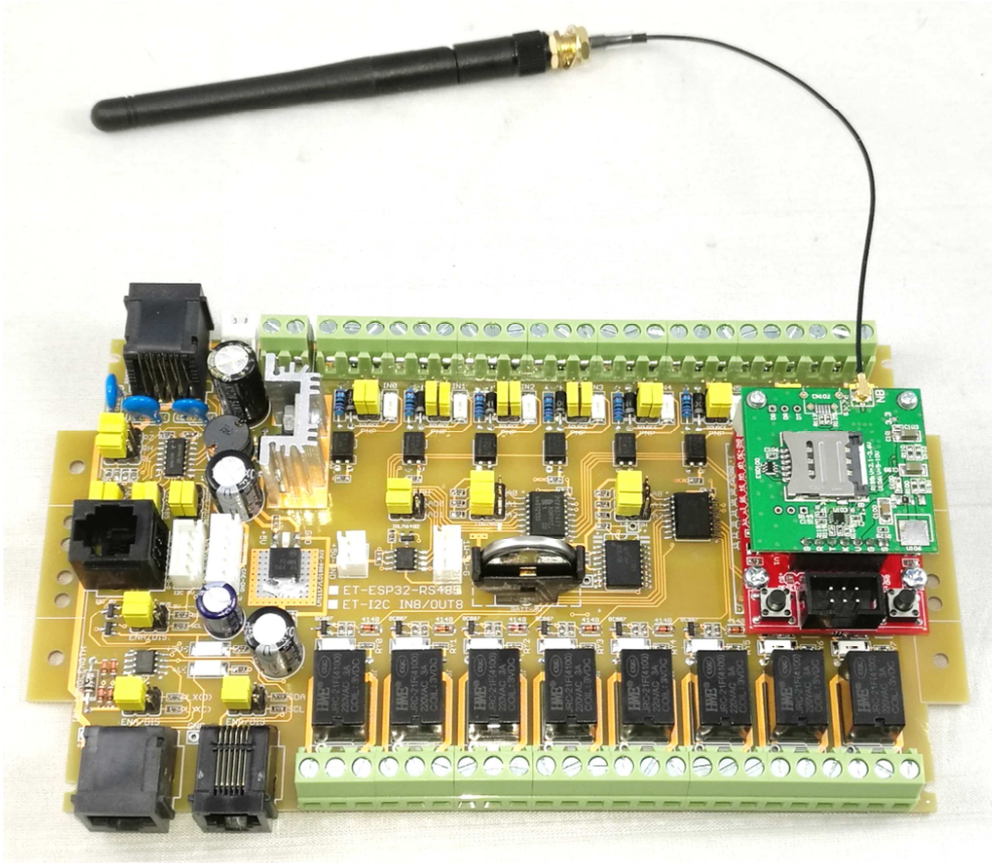


รูปแสดง การติดตั้งโมดูล ET-ESP32 WROVER MODUE กับ เสาอากาศภายนอก

การเชื่อมต่อกับโมดูล **NB-IoT**



ตัวอย่างโมดูล **NB-IoT** รุ่น **SIM7020E** และ **SIM7000E** ที่สามารถติดตั้งกับบอร์ดได้ทันที



ตัวอย่างการติดตั้งโมดูล **NB-IoT** ใช้งานร่วมกับบอร์ด **ET-ESP32 WROVER V1**

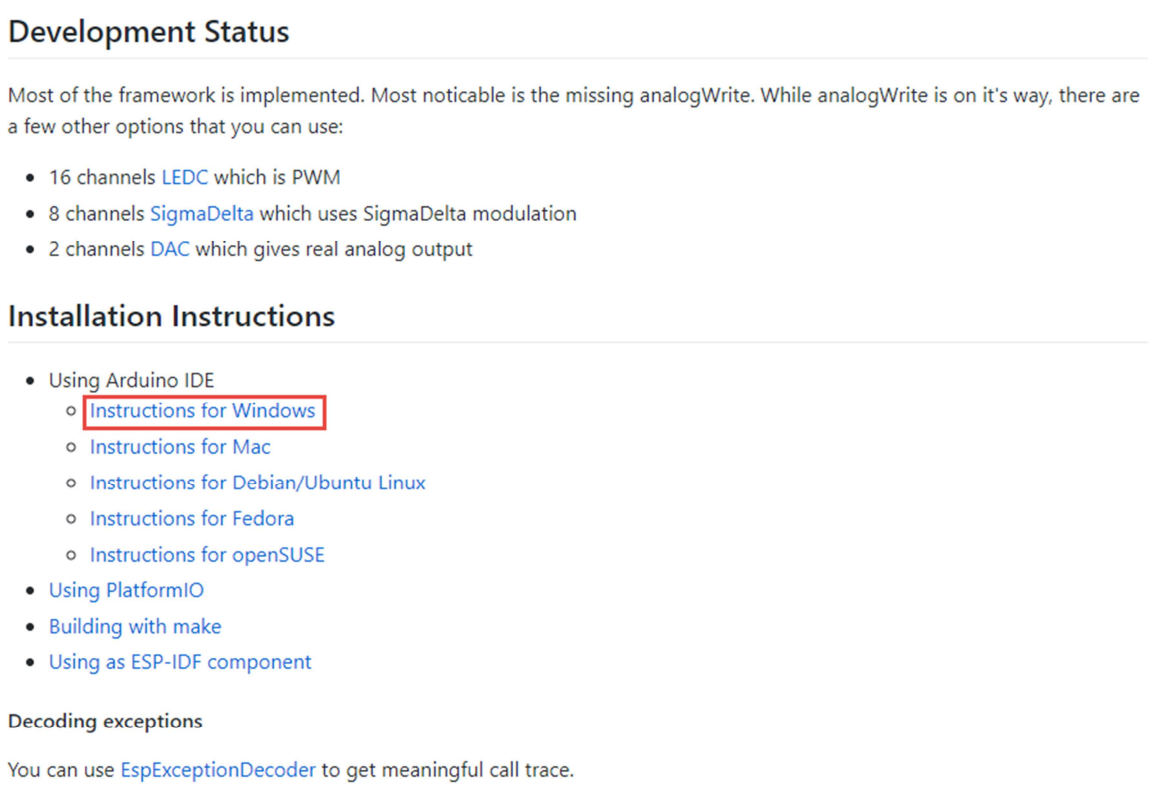
การติดตั้ง esp32 tools เพื่อใช้งานกับโปรแกรม Arduino IDE

ในกรณีที่ยังไม่ได้ติดตั้งใช้งานโปรแกรม Arduino IDE ไว้เลย ผู้ใช้จะต้องเริ่มต้นทำการติดตั้งโปรแกรมใหม่ทั้งหมด ซึ่งโปรแกรม Arduino IDE สามารถทำการติดตั้งใช้งานได้กับระบบปฏิบัติการ ที่หลากหลาย ซึ่งวิธีการอาจมีความแตกต่างกันแต่ในที่นี้จะขอแนะนำขั้นตอนการติดตั้งกับระบบปฏิบัติการของ Windows เท่านั้น สำหรับระบบปฏิบัติการอื่นผู้ใช้สามารถเข้าไปศึกษาขั้นตอนวิธีการได้ในหัวข้อ Installation Instructions ซึ่งจะมี Link ของหัวข้อสำหรับวิธีการติดตั้งในแต่ละระบบปฏิบัติการให้เลือกในที่นี้ให้ไปที่หัวข้อ Development Status -> Installation Instructions ให้เลือก Instructions for Windows

ซึ่งวิธีการและขั้นตอนการติดตั้งโปรแกรม esp32 tools เพื่อใช้งานกับ Arduino IDE แบบต่างๆ ในหลายๆระบบปฏิบัติการ รวมทั้งของ Windows จะอยู่ที่

https://github.com/espressif/arduino-esp32

ซึ่งผู้ใช้สามารถอ้างอิงได้จากหัวข้อ Development Status -> Installation Instructions ดังรูป



Development Status

Most of the framework is implemented. Most noticable is the missing analogWrite. While analogWrite is on it's way, there are a few other options that you can use:

- 16 channels LEDC which is PWM
- 8 channels SigmaDelta which uses SigmaDelta modulation
- 2 channels DAC which gives real analog output

Installation Instructions

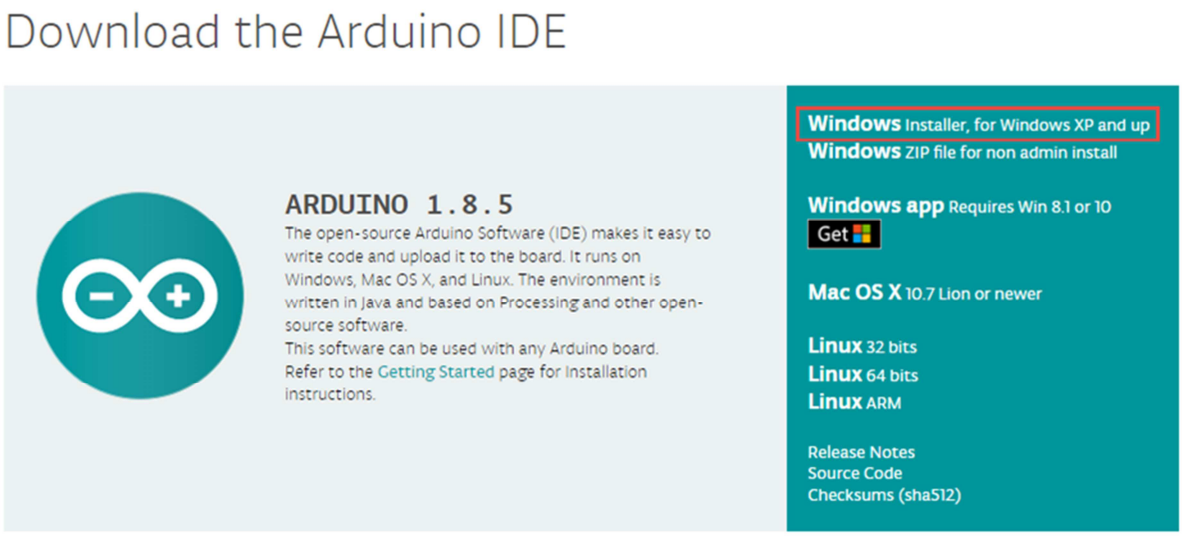
- Using Arduino IDE
  - Instructions for Windows
  - Instructions for Mac
  - Instructions for Debian/Ubuntu Linux
  - Instructions for Fedora
  - Instructions for openSUSE
- Using PlatformIO
- Building with make
- Using as ESP-IDF component

Decoding exceptions

You can use EspExceptionDecoder to get meaningful call trace.

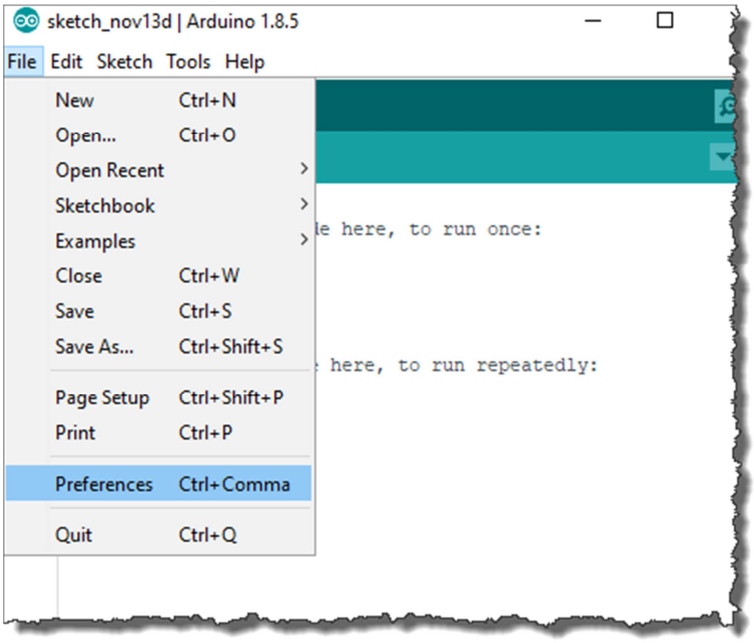
การติดตั้ง Arduino IDE

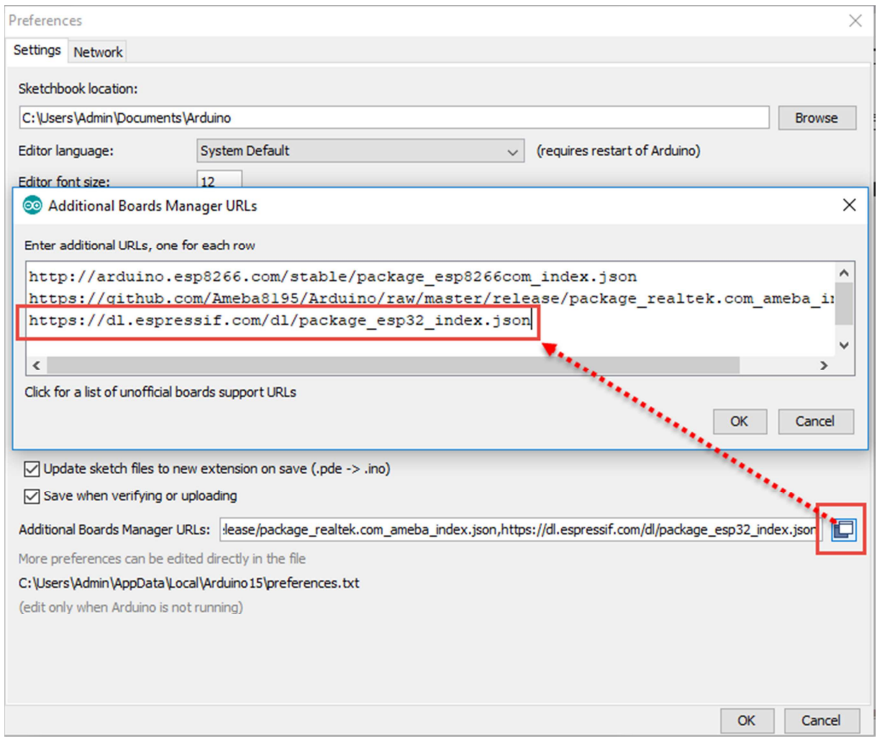
ในกรณีที่ยังไม่เคยติดตั้งโปรแกรม Arduino IDE ไว้ก่อนเลย ผู้ใช้ต้องทำการ download และติดตั้งโปรแกรม Arduino IDE จาก url ชื่อ https://www.arduino.cc ให้เรียบร้อย ซึ่งผู้ใช้สามารถสั่ง Run โปรแกรม Install และ ให้โปรแกรม Install ของ Arduino IDE ทำการติดตั้ง tools ต่างๆให้เองโดยอัตโนมัติตามขั้นตอนมาตรฐานของการ Install โปรแกรม



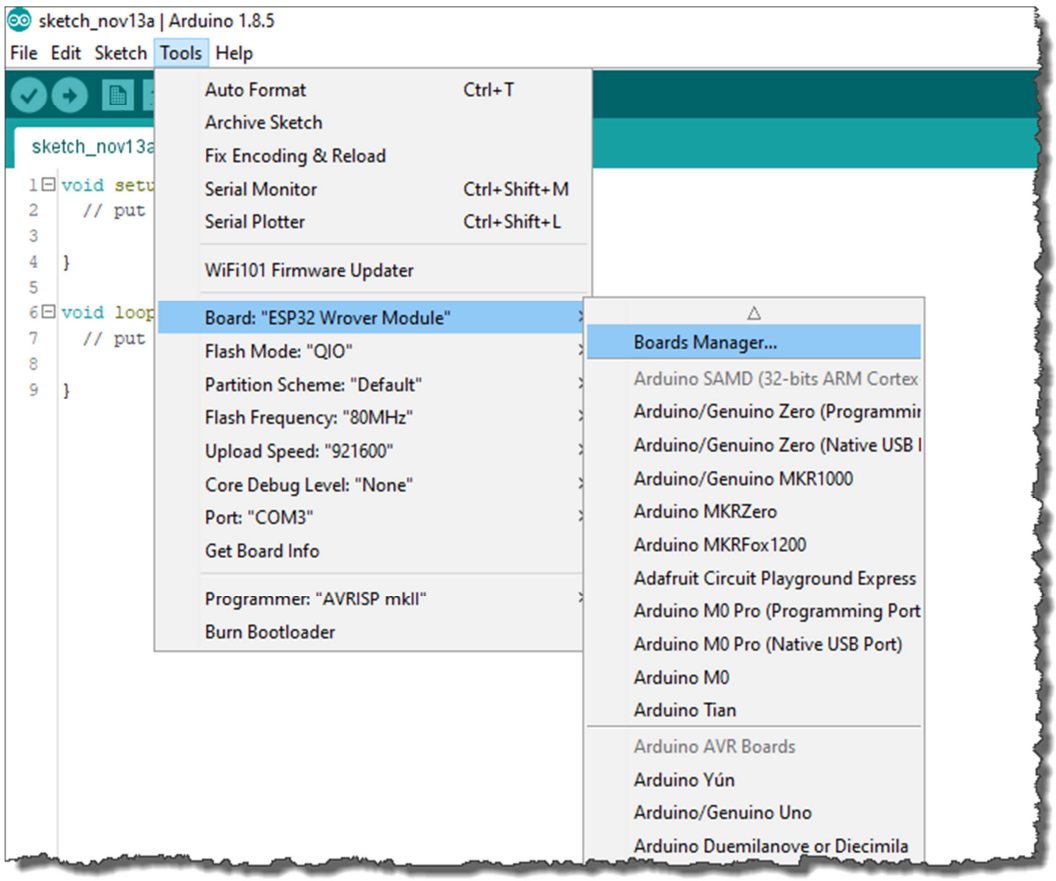
หลังจากทำการดาวน์โหลดและติดตั้งโปรแกรม Arduino IDE เสร็จเรียบร้อยแล้ว ขั้นตอนต่อไปจะเป็นการติดตั้ง esp32 tools ลงใน Arduino IDE เพื่อใช้งานร่วมกัน ซึ่งในขั้นตอนนี้ ผู้ใช้สามารถกระทำผ่านทางเมนู Board Manager... ของ Arduino IDE ได้โดยสะดวก แต่ต้องทำในขณะที่คอมพิวเตอร์ PC เชื่อมต่อ online กับ Internet ไว้แล้วเท่านั้น ถ้าไม่ได้เชื่อมต่อ Internet ไว้จะไม่สามารถกระทำได้ โดยมีวิธีการดังนี้

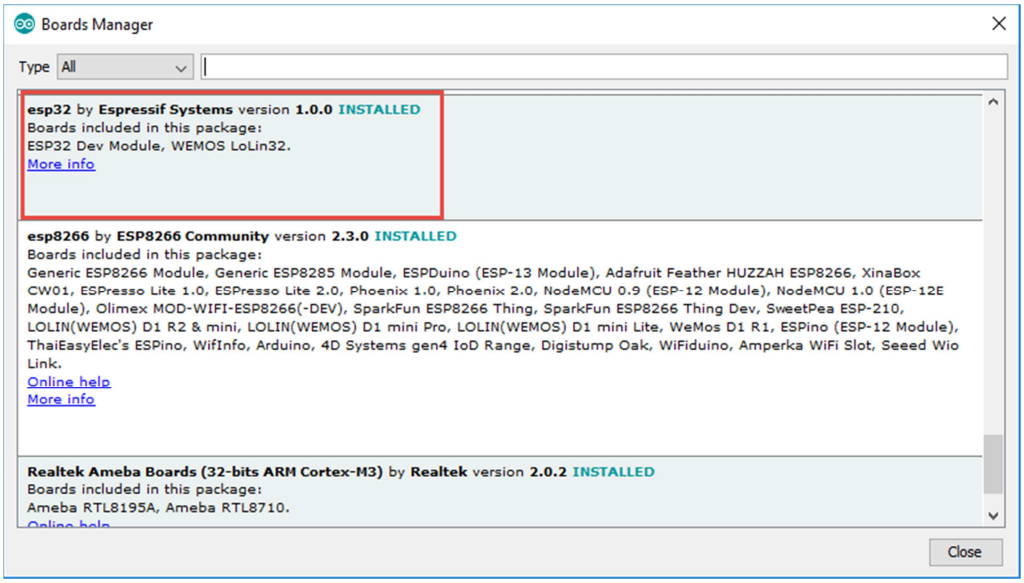
1. เลือกเมนู File → Preferences → Additional Boards Manager URLs: แล้วเพิ่ม URL ที่เก็บไฟล์ของ esp32 Tools ไว้ คือ “https://dl.espressif.com/dl/package_esp32_index.json” ดังตัวอย่าง



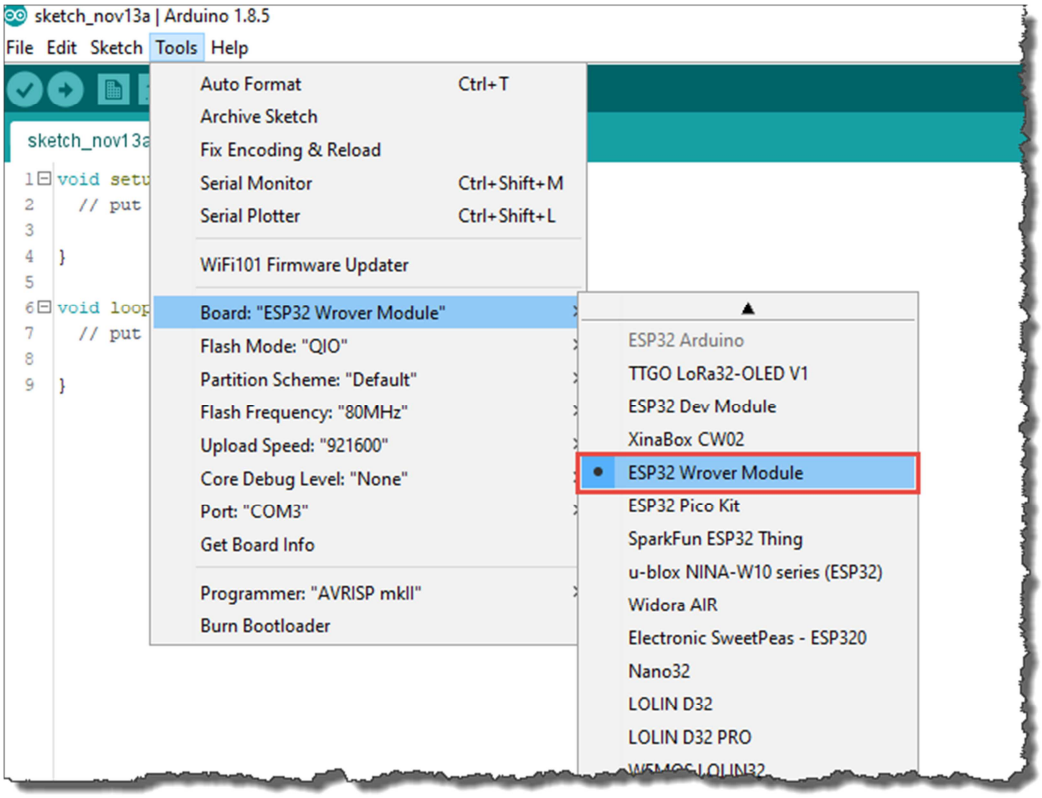


2. เลือกเมนู Tools → Board → Boards Manager…แล้วค้นหาและเลือกชุด Tools สำหรับ esp32 ของ Espressif → esp32 by Espressif System version 1.0.0 แล้วเลือก Install





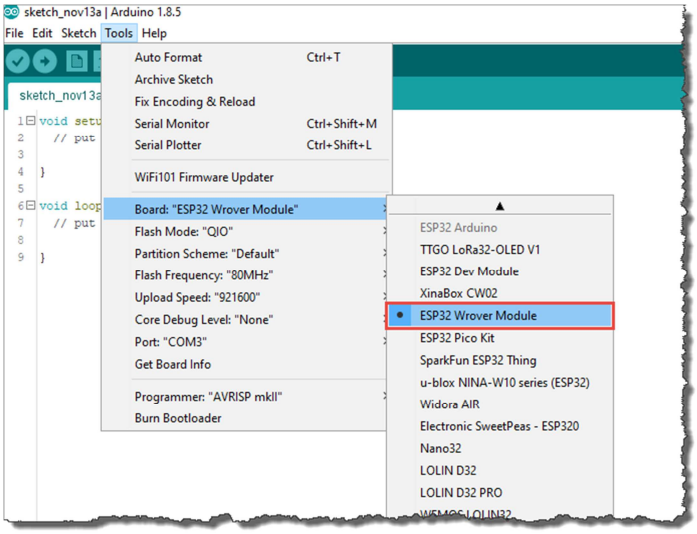
3. หลังจากทำการ Install Tools ของ esp32 ผ่าน Boards Manager... เสร็จเรียบร้อยแล้ว ให้ทดลองเลือกที่เมนู Tools → Board ถ้าถูกต้องจะมีตัวเลือกบอร์ดของ esp32 ให้เลือกใช้ ในกรณีของบอร์ด ET-ESP32 RS485 ให้เลือกชุดบอร์ดของ ESP32 Wrover Module ดังรูป



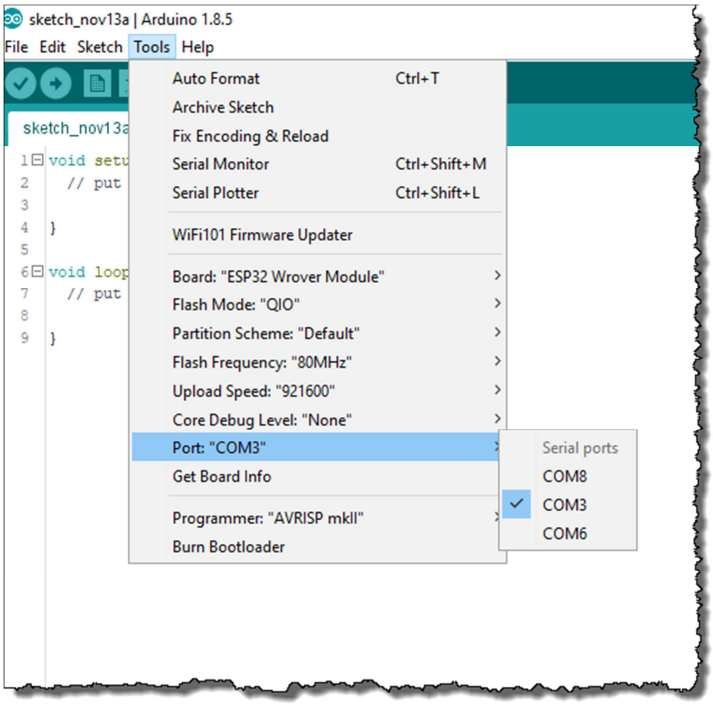
การใช้งาน esp32 กับ Arduino IDE

1. สั่ง Run Arduino IDE แล้วเลือกที่เมนู Tools -> Board: ซึ่งจะมีเมนูรุ่นของบอร์ดในตระกูล ESP32 ให้เลือกกำหนดในการใช้งาน ในที่นี้ให้เลือก ESP32 Wrover Module และที่เมนู Port ให้เลือกหมายเลขพอร์ตเป็น COM Port ของบอร์ดที่ได้จากการจัดสรรของ Windows Driver ดังตัวอย่าง

Tools -> Board: ESP32 Wrover Module

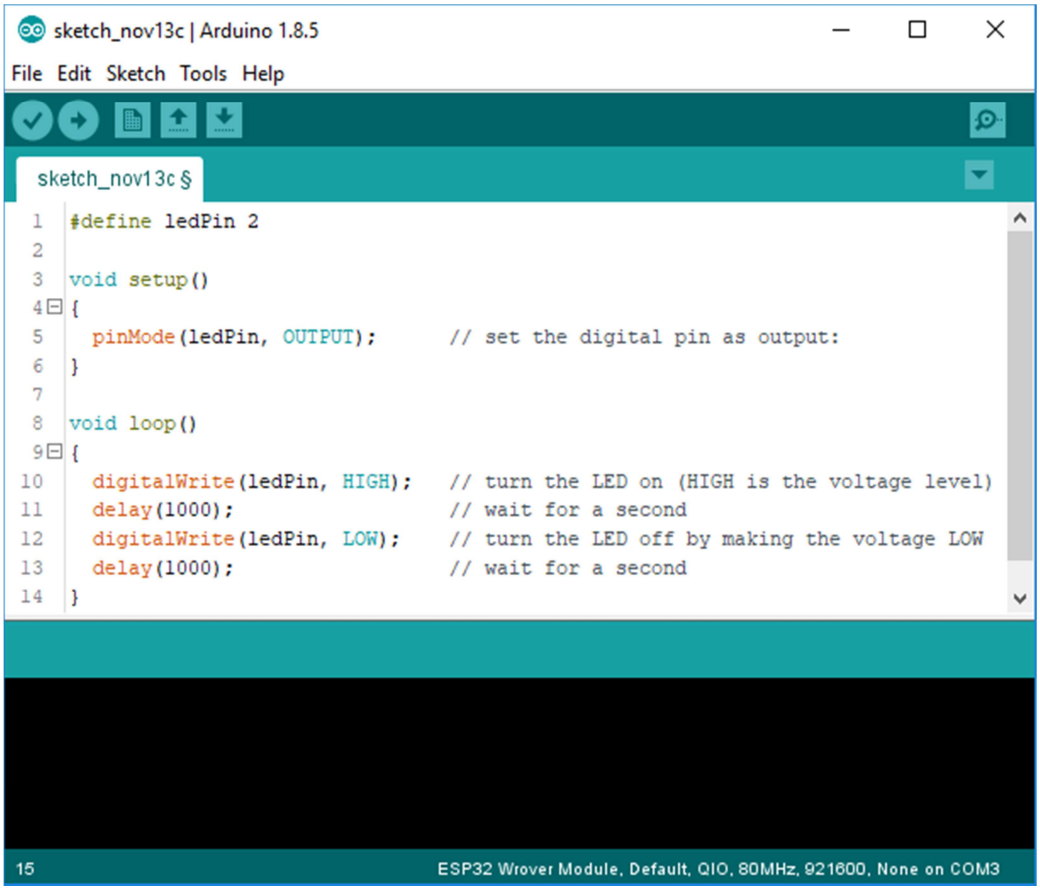


Tools -> Port:หมายเลข **COM Port** ของบอร์ด

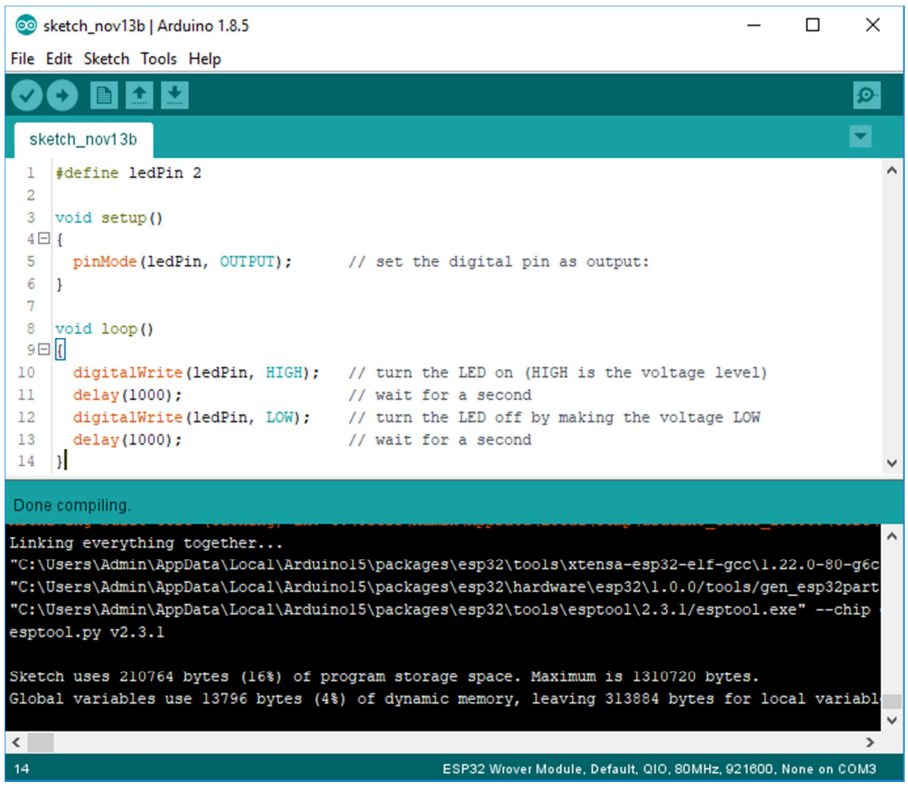


2. ทำการเลือกที่เมนู File New ซึ่งจะได้หน้ากระดาษสำหรับเริ่มต้นเขียนโปรแกรมมาให้ ให้ทำการพิมพ์โค๊ดโปรแกรมสำหรับทดสอบการทำงานของบอร์ดแล้วสั่งบันทึกไว้ดังตัวอย่าง

```
#define ledPin 2


void setup()
{
  pinMode(ledPin, OUTPUT);      // set the digital pin as output:
}


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



3. สั่งบันทึกและทดสอบ Compile



4. ทดสอบสั่ง Upload และดูการทำงานของบอร์ด