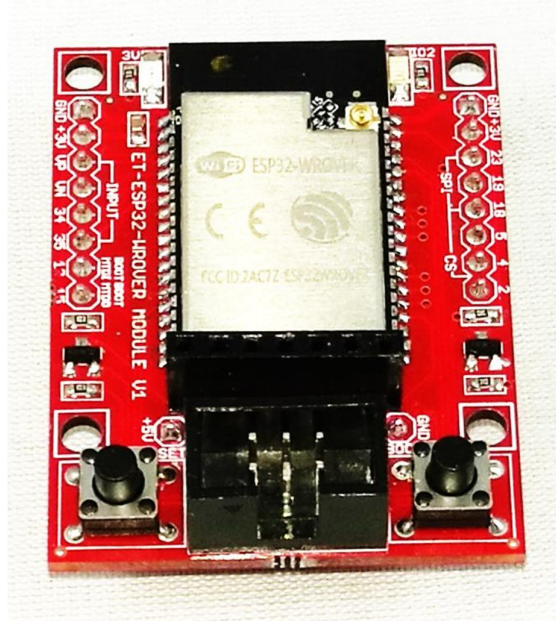


ET-ESP32 WROVER MODULE V1



ET-ESP32 WROVER MODULE V1 is Module MCU ESP32 of Espressif Systems that uses the Module model "ESP32 WROVER-I" to be MCU on board.

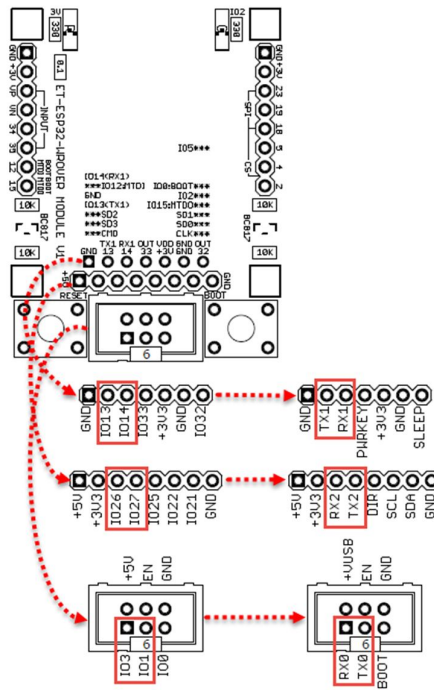
- **Memory**
 - 4MB SPI Flash(32Mbits SPI flash)/ 8MB PSRAM(64Mbits PSRAM)
- **Wi-Fi**
 - Frequency range: 2.4 - 2.5 GHz
 - Support Protocol 802.11 b/g/n (802.11n up to 150 Mbps)
 - A-MPDU and A-MSDU aggregation and 0.4 uS guard interval support
- **Bluetooth**
 - Support Protocol Bluetooth 4.2 BR/EDR and BLE
 - NZIF receiver with -98 dBm sensitivity
 - Class-1, class-2 and class-3 transmitter
 - AFH
 - CVSD and SBC
- **Hardware**
 - Module supports Interface of SD card, UART, SPI, SDIO, I2C, LED PWM, Motor PWM, I2S, I2C, IR
 - Hall Sensor and Temperature Sensor on Chip
 - 40 MHz Crystal on Board
 - Voltage in a range of 2.3 to 3.6 Volt
 - Average Current is 80mA
- **Software**
 - Support Wi-Fi on Mode of Station, SoftAP, SoftAP+Station, P2P
 - Support WPA/ WPA2/ WPA2-Enterprise/ WPS
 - Support Encoding of AES/ RSA/ ECC/ SHA
 - Upgrade Firmware UART/ OTA (via network)
 - Support Network of IPv4, IPv6, SSL, TCP/ UDP/ HTTP/ FTP/ MQTT

User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO ALLOCATE USART

There are 3 channels of USART Port Communication on Board ET-ESP32 RS485. The first channel (USART0) is used to download and debug the operation of system; the second channel (USART1) is used to connect to Module NB-IoT or other devices that are communicated through TTL; and the third channel(USART2) is used to connect with devices that are communicated through RS422/485 Communication. It allocates I/O Pins for USART Communication as follows;

- **USART0(USART-TTL)** is connected to ET-USB USART/TTL to upload Code and debug. It can show the operation or receive-data data as preferred.
 - IO3 is used as RX0.
 - IO1 is used as TX0.
- **USART1(USART-TTL)** is used to receive-send data according to the written firmware in order to control receiving-sending data.
 - IO14 is used as RX1.
 - IO13 is used as TX1.
- **USART2(RS422/485)** is used to receive-data data according to the written firmware in order to control receiving-sending data.
 - IO26 is used as RX2.
 - IO27 is used as TX2.
 - IO25 is used as Signal DIR (Direction).



In case of developing program by platform of Arduino, it has to setup I/O Pins to connect signals between Hardware USART of Module ESP32-WROVER and Board ET-ESP32 RS485 as follows;

User's Manual of Board ET-ESP32 WROVER MODULE V1

```
#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);

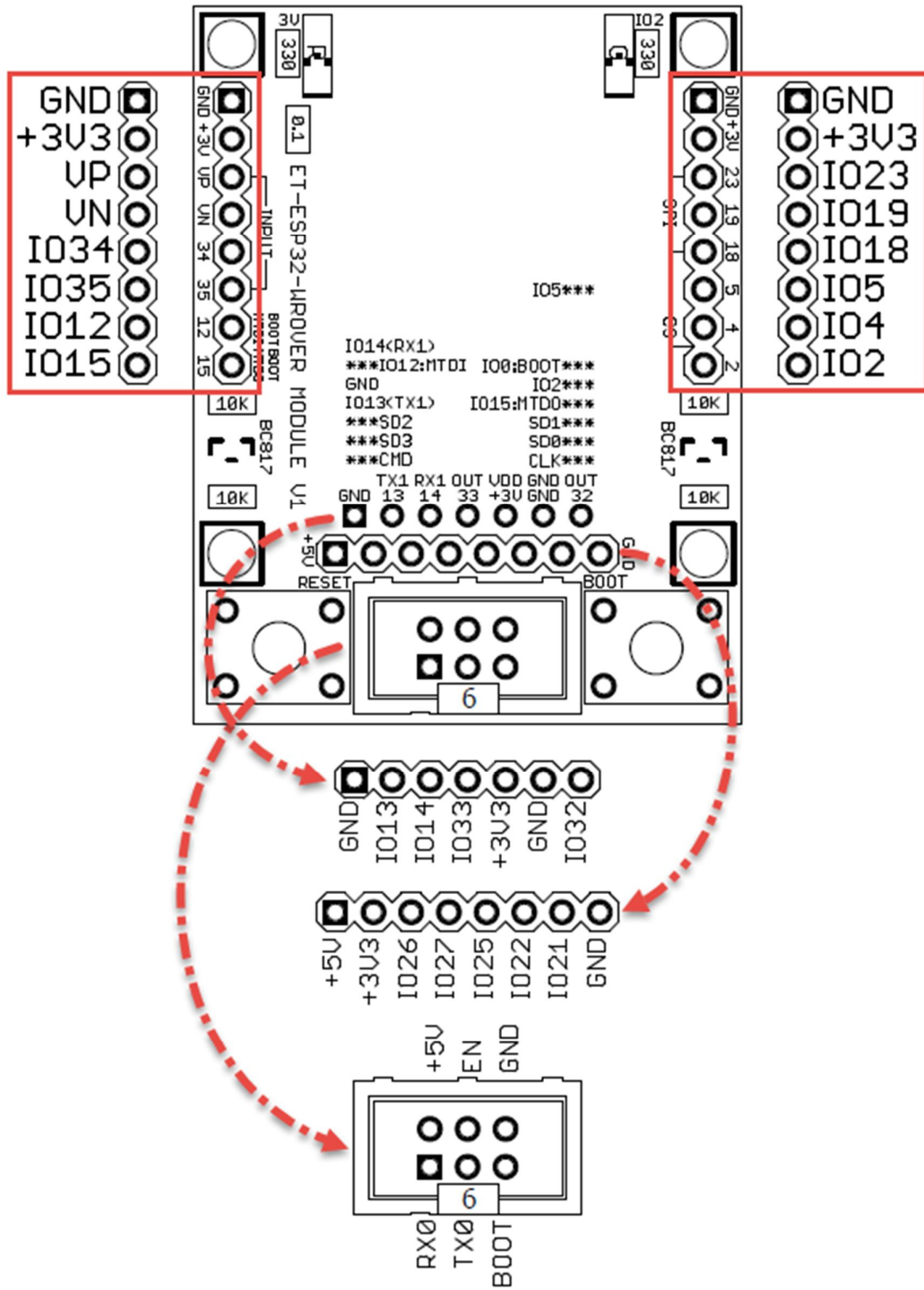
  while(!SerialRS485);

  ...
}

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

It shows an example of initialling USART of Board ET-ESP32-RS485.

User's Manual of Board ET-ESP32 WROVER MODULE V1

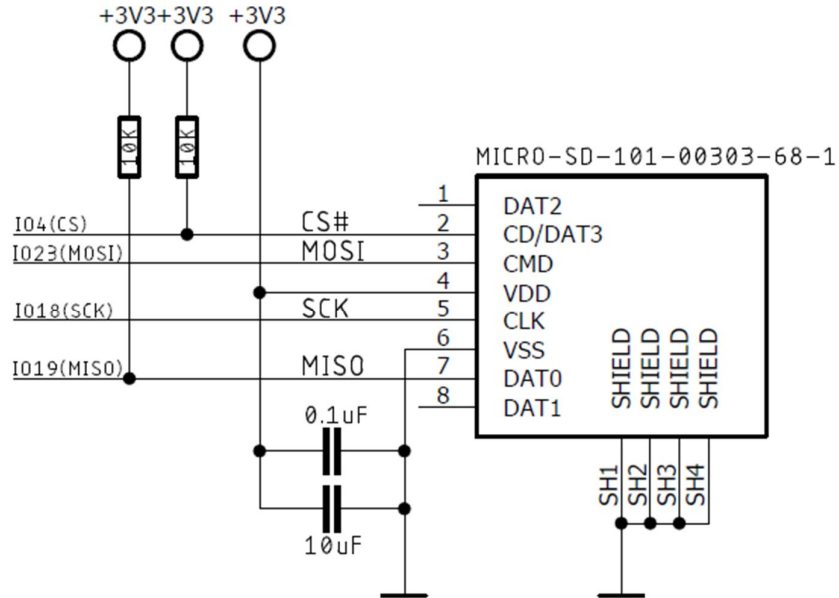


It shows how to arrange Signals of Board ET-ESP32 WROVER MODULE V1.

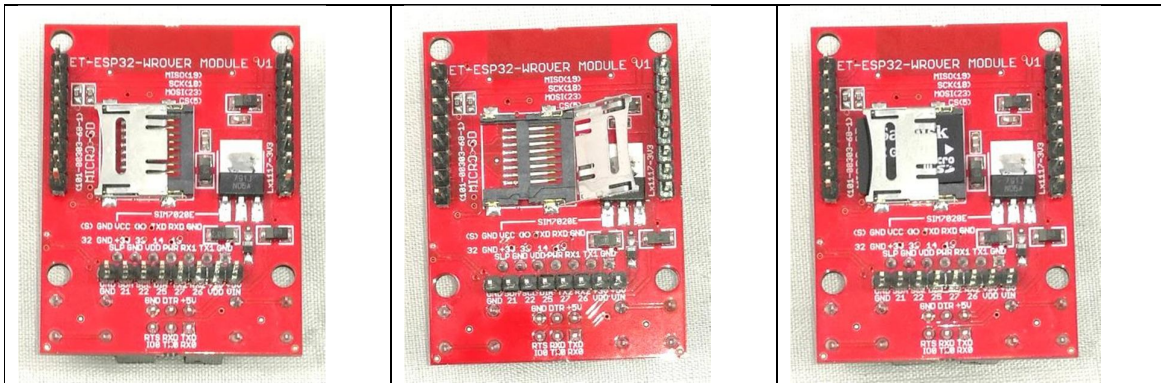
User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO INSTALL MICRO SD ON BOARD

There is Socket Micro SD underneath the breadboard of Board ET-ESP32 WROVER MODULE V1 to install Micro SD Card; it connects the circuit of ESP32 in a format of SPI Mode through Port SPI as shown below;



It shows the circuit that connects Micro SD by SPI Port.

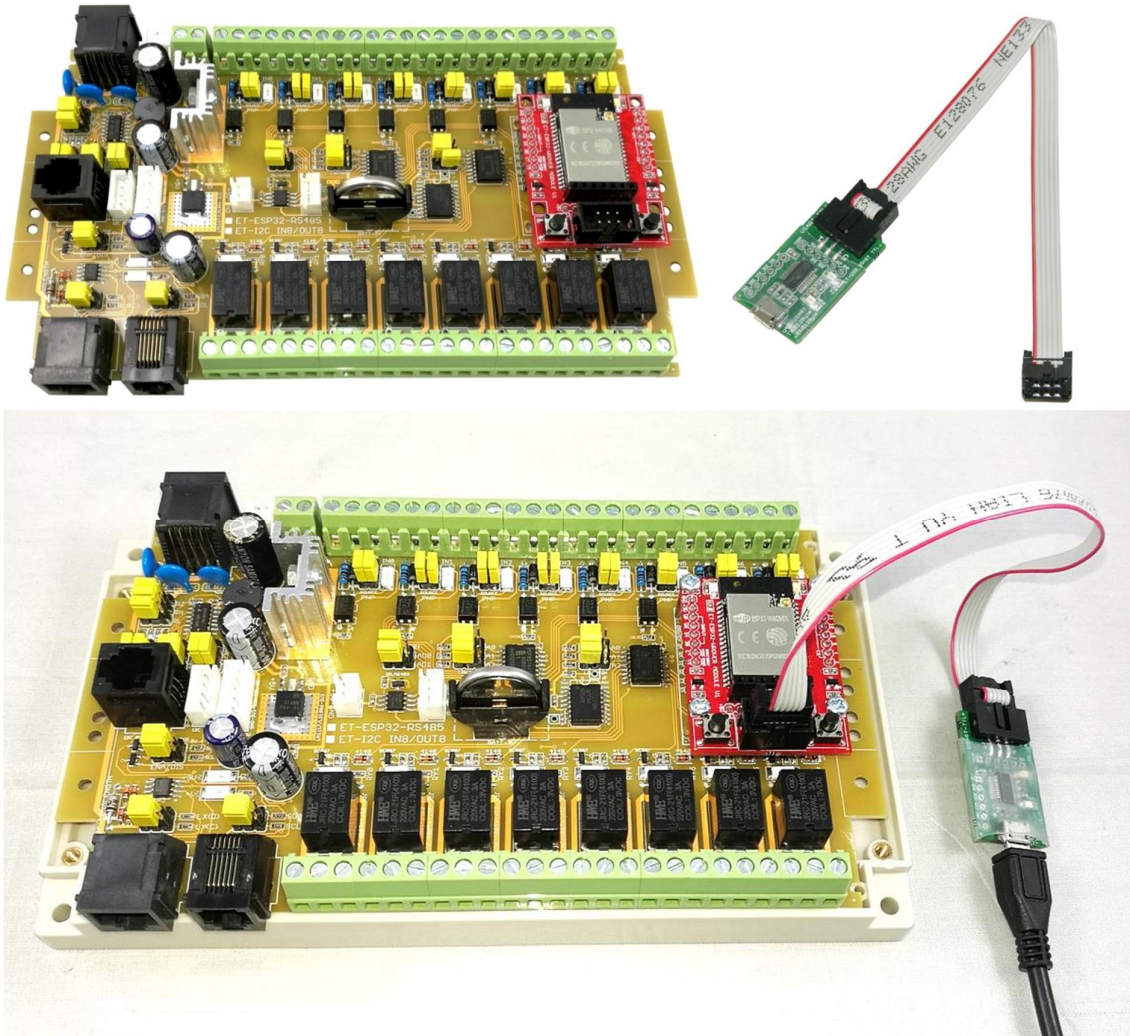


It shows how to insert Micro SD Card into the Socket.

User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO UPLOAD PROGRAM

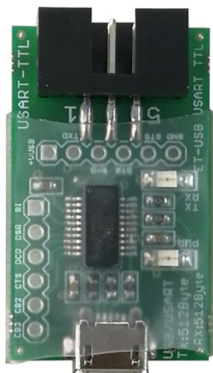
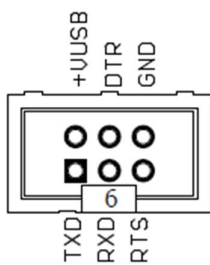
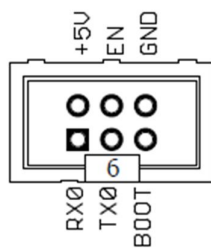
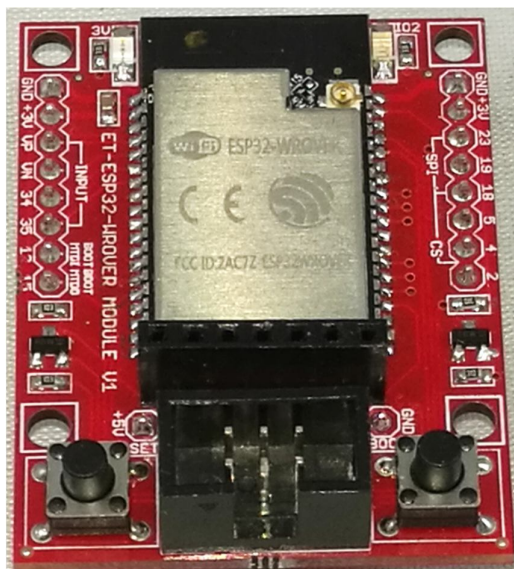
It can upload program into board through external device that is used to be USB/USART as TTL type and the Signal is connected through Connector 6PIN IDE by using Pair Cable as shown below;



NOTE

Be careful about using Board ET-ESP32 WROVER MODULE V1 together with Board ET-ESP32 RS485, PORT USB may be damaged because the Power Supply of USB has no enough Current to support devices internal circuit on board. So, it has to provide Power Supply into Board ET-ESP32 RS485 completely before connecting USB Cable.

User's Manual of Board ET-ESP32 WROVER MODULE V1



It shows how to connect Signal to upload program by using ET-USB USART/TTL.

User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO CHOOSE BETWEEN INTERNAL ANTENNA OR EXTERNAL ANTENNA

Normally, there is an internal antenna as PCB type on Module ESP32 WROVER-I completely; it can receive-send data with a Frequency of 2.4GHz. If it can't use any antenna internal board; for example, it has to install the work piece internal metal box but it blocks Signal Frequency, it can install external antenna on the Module as preferred. The circuit of Module that is designed and produced by Espressif has 0 Ohm Resistor that is used to be Jumper to choose either external antenna or internal antenna. For the Standard Module ESP32-WROVER-I, there is R-Jumper that has already been soldered in order to choose the external antenna; so, it can additionally connect external antenna that its own Cable is connected through Connector SMA. It is more convenient because it is soldered by ETT completely; it is easier to connect both internal antenna and external antenna to receive signal. Although practically it may not use both internal antenna and external antenna simultaneously, there is a problem with matching Signal incorrectly and it is less efficient to receive-send data. It should use only one type of antenna. If it requires using external antenna, it has to remove the connection between the soldering point and internal antenna as shown in the picture below;

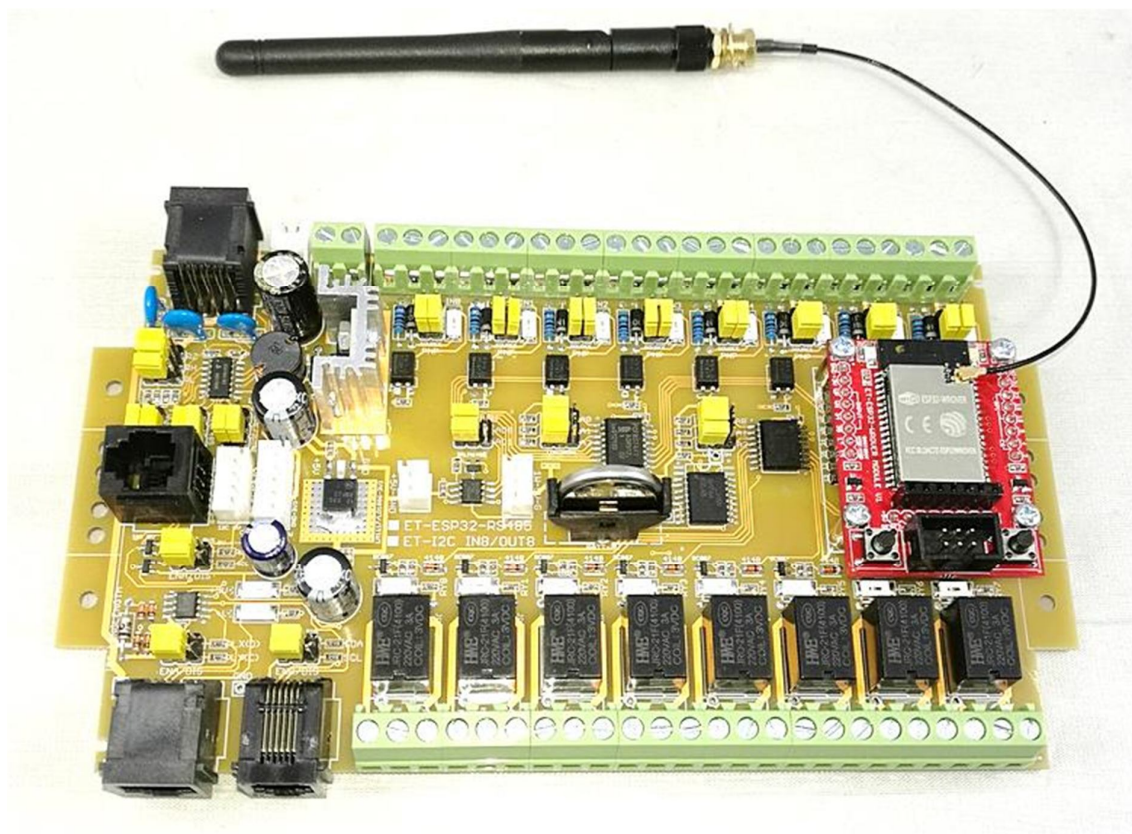


It shows how to solder Jumper(R 0 Ohm) to choose either internal antenna or external antenna.

User's Manual of Board ET-ESP32 WROVER MODULE V1



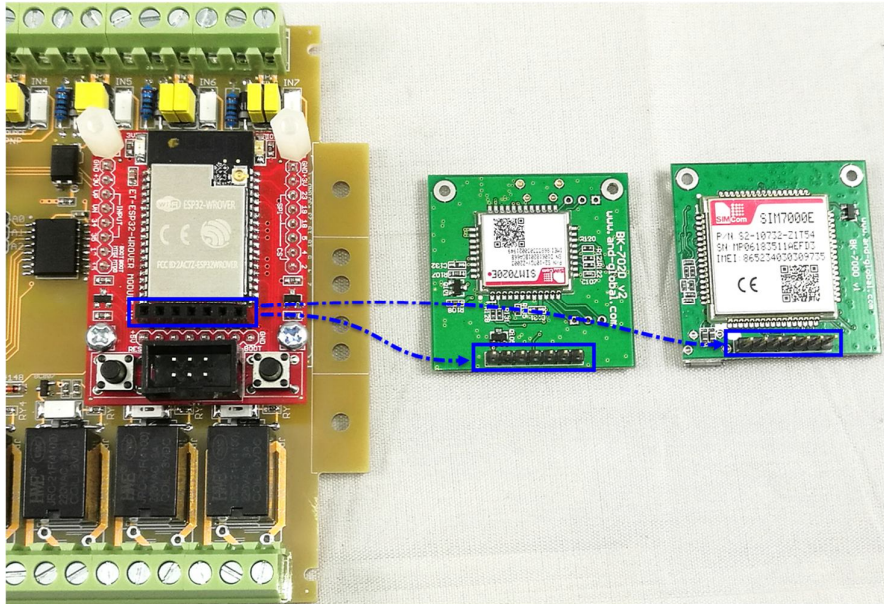
It shows an example of antenna "MT7681" for external use.



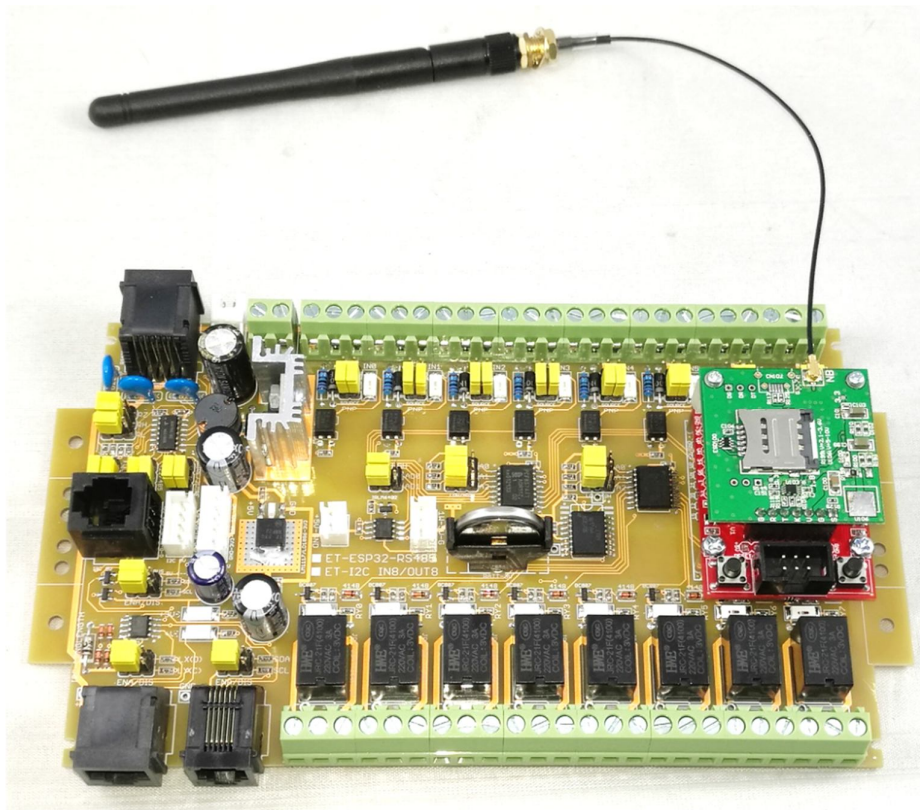
It shows how to install external antenna on the Module ET-ESP32 WROVER MODULE.

User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO CONNECT WITH MODULE NB-IoT



It shows an example of Module NB-IoT model "SIM7020E" and Model "SIM7000E" that can be installed on Board instantly.



It shows an example of installing Module NB-IoT on Board ET-ESP32 WROVER V1.

User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO INSTALL ESP32 TOOLS FOR PROGRAM ARDUINO IDE

If it hasn't installed any Program Arduino IDE yet, it has to install all new programs. There are various Operating Systems that can be used with this Program Arduino IDE; in this case, it suggests how to install Program Arduino IDE under the Operating System of Windows only. For other Operating Systems, it is slightly different method to install program; please look at **Installation Instructions** and click Link of the preferable Operating System to install program. In this case, please go to **Development Status** -> **Installation Instructions** and then choose **Instructions for Windows**.

Please click the Link below, it describes how to install Program esp32 tools for using with Arduino IDE; it includes various Operating Systems and including Windows.

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

Please look at the picture below, it is an example of installing program; "**Development Status** -> **Installation Instructions**".

Development Status

Most of the framework is implemented. Most noticeable 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

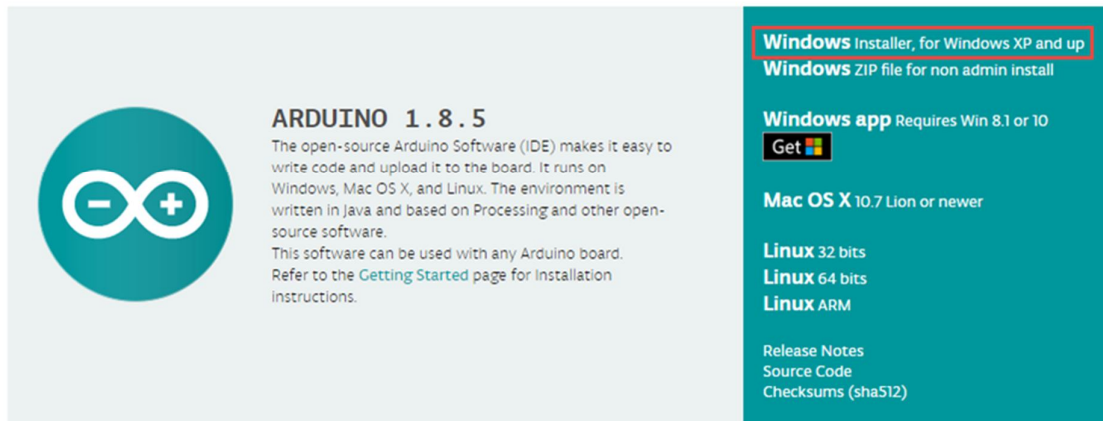
You can use [EspExceptionDecoder](#) to get meaningful call trace.

User's Manual of Board ET-ESP32 WROVER MODULE V1

HOW TO INSTALL ARDUINO IDE

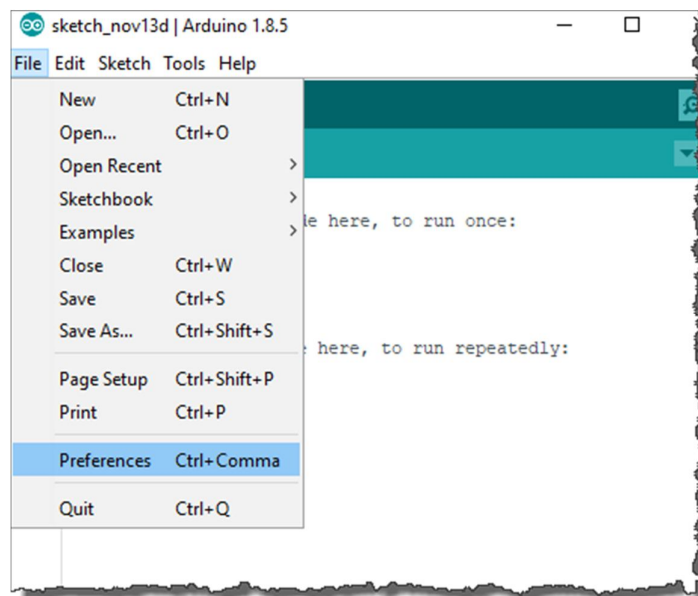
If it hasn't installed any Program Arduino IDE yet, it has to download and install Program Aduino IDE from URL: <https://www.arduino.cc> first. In this case, it auto-runs Program Install of Arduino IDE to install Tools according to the standard of installing program.

Download the Arduino IDE

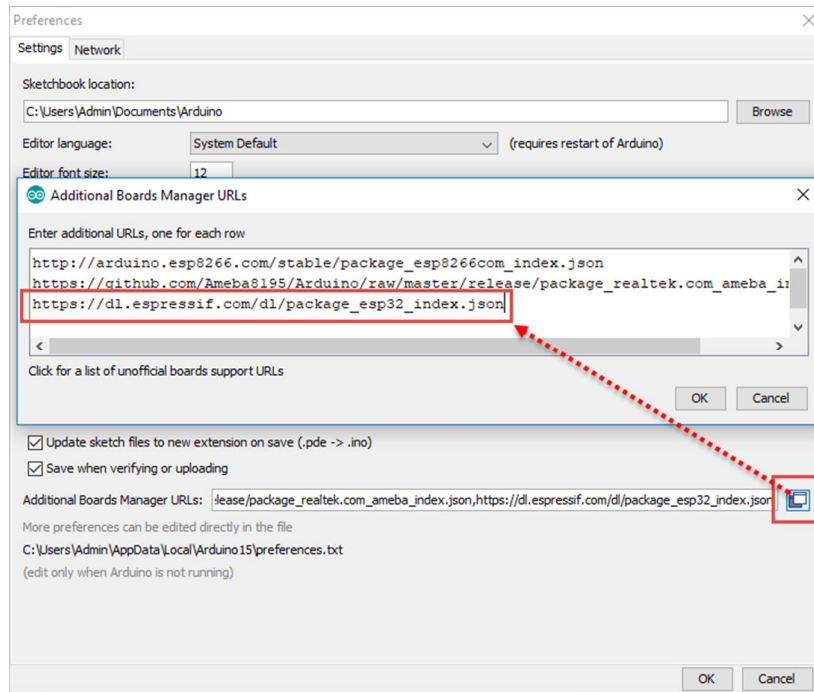


After downloaded and installed Program Arduino IDE successfully; next, it has to install ESP32 TOOLS into Arduino IDE that can do through Menu **Board Manager...** of Arduino IDE conveniently. Remember, it has to do this process while computer PC is connecting to internet or ONLINE only, otherwise, it can't do. Please follow these instructions;

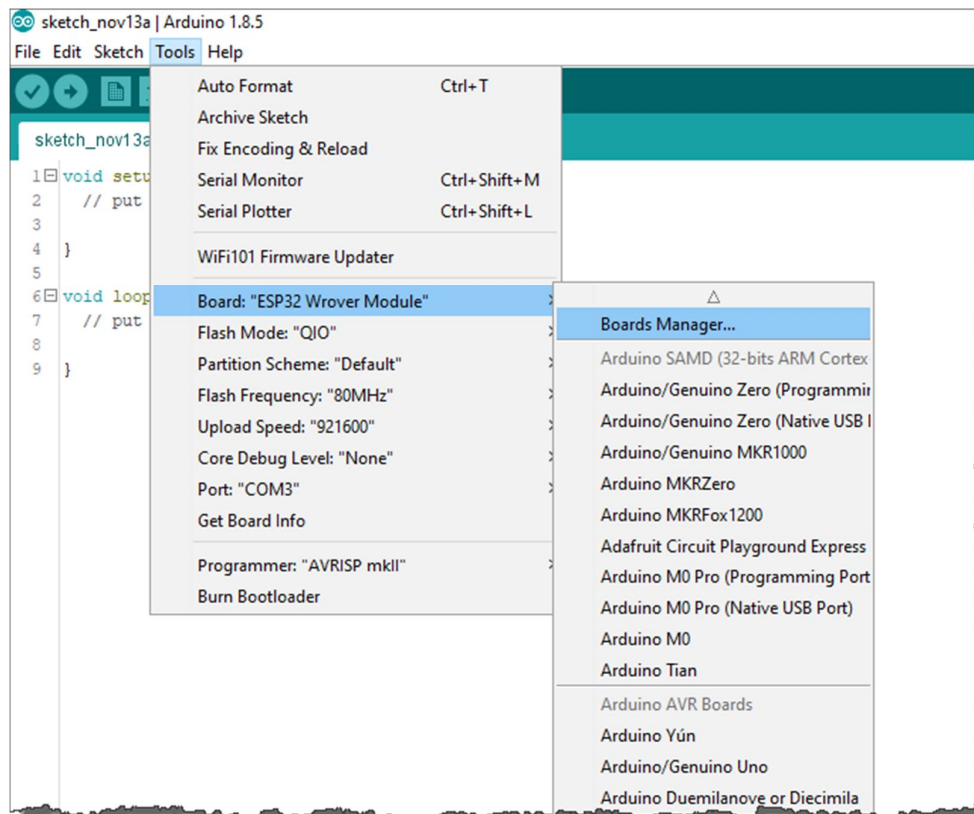
1. Click Menu **File** → **Preferences** → **Additional Boards Manager URLs:** and add URL "https://dl.espressif.com/dl/package_esp32_index.json" to store file of esp32 Tools as shown below;



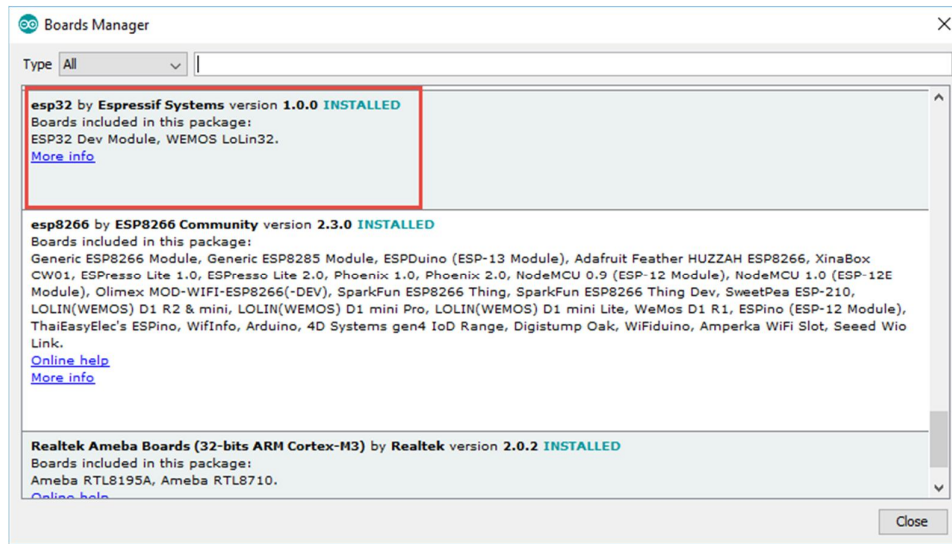
User's Manual of Board ET-ESP32 WROVER MODULE V1



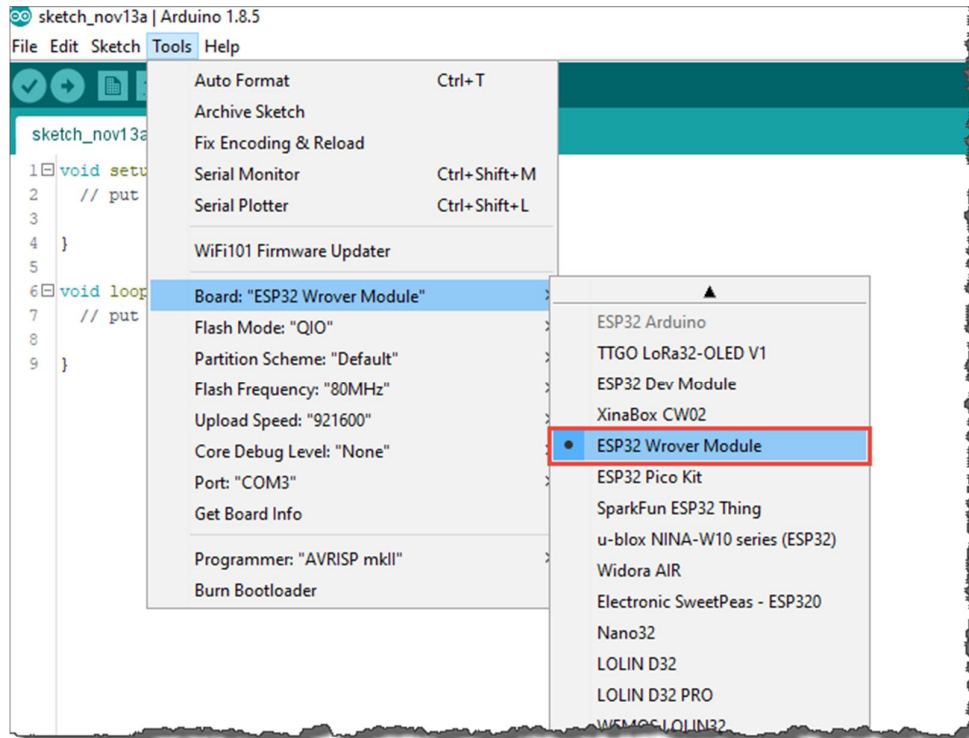
2. Click **Menu Tools** → **Board** → **Boards Manager...**, search and choose Tools for **esp32** of **Espressif** → **esp32 by Espressif System version 1.0.0**; and then choose **Install**.



User's Manual of Board ET-ESP32 WROVER MODULE V1



3. After installed Tools of esp32 through Boards Manager... successfully; click **Tools** → **Board** to test the operation. If everything is correct, it shows alternatives of Board esp32; in case of Board ET-ESP32 RS485, please choose Board of ESP32 Wrover Module as shown below;

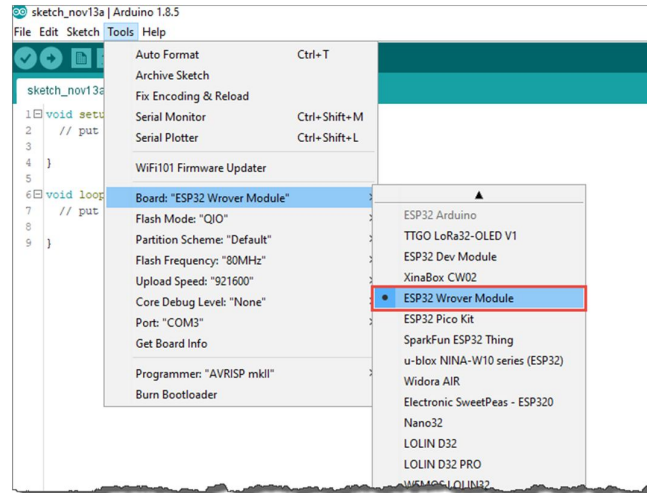


User's Manual of Board ET-ESP32 WROVER MODULE V1

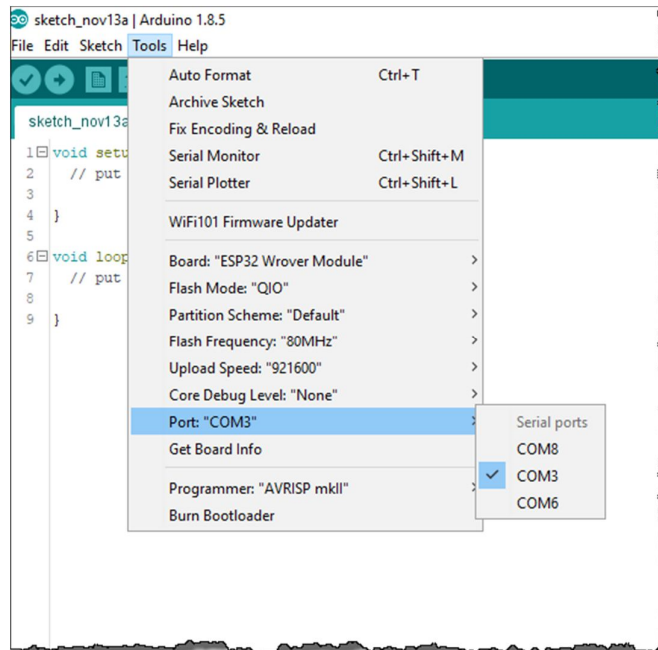
HOW TO USE ESP32 TOGETHER WITH ARDUINO IDE

1. Run Arduino IDE and choose **Menu Tools -> Board:**, it shows versions of board in a family of ESP32; in this case, choose **ESP32 Wrover Module**. For **Menu Port**, please choose the COM Port number that is allocated by Windows Driver as shown below;

Tools -> Board: ESP32 Wrover Module



Tools -> Port: COM Port No. of Board



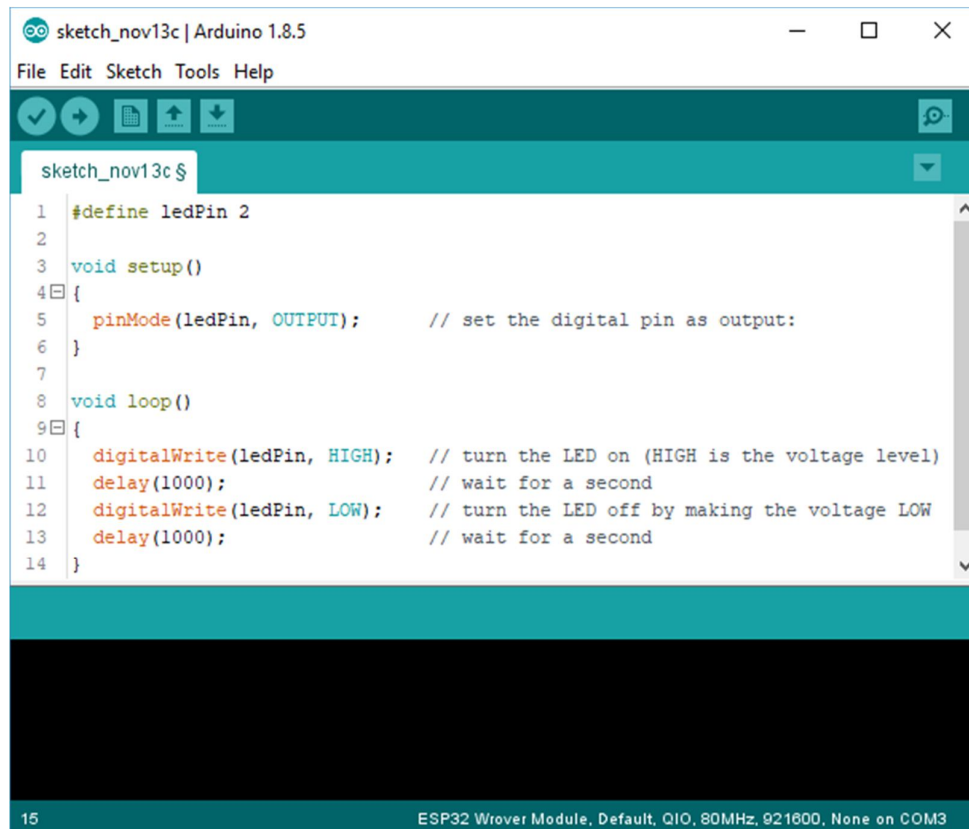
User's Manual of Board ET-ESP32 WROVER MODULE V1

2. Choose **Menu File New**, it shows a blank page for writing program; please type; please write Code Program to test the operation of board and then save the Code as shown below;

```
#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
}
```



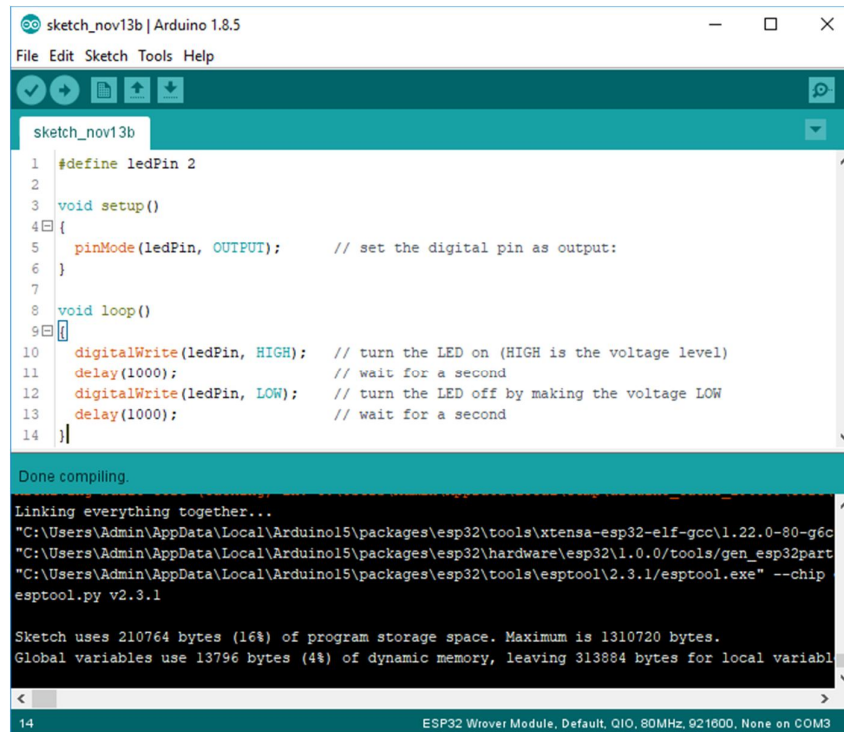
The screenshot shows the Arduino IDE interface. The title bar reads "sketch_nov13c | Arduino 1.8.5". The menu bar includes "File", "Edit", "Sketch", "Tools", and "Help". Below the menu bar is a toolbar with icons for check, run, upload, and download. The main editor area shows the code from the previous block, with line numbers 1 through 14. The status bar at the bottom indicates "15" and "ESP32 Wrover Module, Default, QIO, 80MHz, 921600, None on COM3".

```
sketch_nov13c $
1  #define ledPin 2
2
3  void setup()
4  {
5    pinMode(ledPin, OUTPUT);    // set the digital pin as output:
6  }
7
8  void loop()
9  {
10   digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
11   delay(1000);                // wait for a second
12   digitalWrite(ledPin, LOW);  // turn the LED off by making the voltage LOW
13   delay(1000);                // wait for a second
14 }
```

15 ESP32 Wrover Module, Default, QIO, 80MHz, 921600, None on COM3

User's Manual of Board ET-ESP32 WROVER MODULE V1

3. Save and Compile to test the operation.



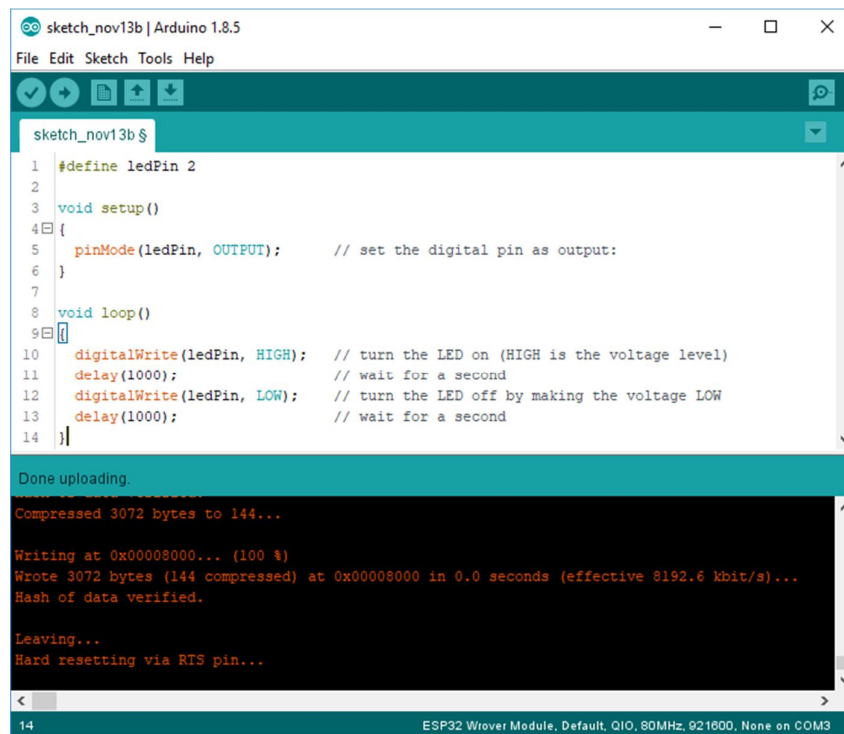
```
sketch_nov13b | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_nov13b
1 #define ledPin 2
2
3 void setup()
4 {
5   pinMode(ledPin, OUTPUT); // set the digital pin as output:
6 }
7
8 void loop()
9 {
10  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
11  delay(1000); // wait for a second
12  digitalWrite(ledPin, LOW); // turn the LED off by making the voltage LOW
13  delay(1000); // wait for a second
14 }

Done compiling.
Linking everything together...
"C:\Users\Admin\AppData\Local\Arduino15\packages\esp32\tools\xtensa-esp32-elf-gcc\1.22.0-80-g6c
"C:\Users\Admin\AppData\Local\Arduino15\packages\esp32\hardware\esp32\1.0.0/tools/gen_esp32part
"C:\Users\Admin\AppData\Local\Arduino15\packages\esp32\tools\esptool\2.3.1/esptool.exe" --chip
esptool.py v2.3.1

Sketch uses 210764 bytes (16%) of program storage space. Maximum is 1310720 bytes.
Global variables use 13796 bytes (4%) of dynamic memory, leaving 313884 bytes for local variabl

ESP32 Wrover Module, Default, QIO, 80MHz, 921600, None on COM3
```

4. Upload and see the operation of board.



```
sketch_nov13b | Arduino 1.8.5
File Edit Sketch Tools Help
sketch_nov13b $
1 #define ledPin 2
2
3 void setup()
4 {
5   pinMode(ledPin, OUTPUT); // set the digital pin as output:
6 }
7
8 void loop()
9 {
10  digitalWrite(ledPin, HIGH); // turn the LED on (HIGH is the voltage level)
11  delay(1000); // wait for a second
12  digitalWrite(ledPin, LOW); // turn the LED off by making the voltage LOW
13  delay(1000); // wait for a second
14 }

Done uploading.
Compressed 3072 bytes to 144...

Writing at 0x00008000... (100 %)
Wrote 3072 bytes (144 compressed) at 0x00008000 in 0.0 seconds (effective 8192.6 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin...

ESP32 Wrover Module, Default, QIO, 80MHz, 921600, None on COM3
```