

Important Information

Thank you for choosing Freenove products!

Getting Started

First, please read the **Read Me First.pdf** document in the unzipped folder you created.

If you have not yet downloaded the zip file, associated with this kit, please do so now and unzip it.

Get Support and Offer Input

Freenove provides free and responsive product and technical support, including but not limited to:

- Product quality issues
- Product use and build issues
- Questions regarding the technology employed in our products for learning and education
- Your input and opinions are always welcome
- We also encourage your ideas and suggestions for new products and product improvements

For any of the above, you may send us an email to:

support@freenove.com

Safety and Precautions

Please follow the following safety precautions when using or storing this product:

- Keep this product out of the reach of children under 6 years old.
- This product should be **used only when there is adult supervision present** as young children lack necessary judgment regarding safety and the consequences of product misuse.
- This product contains small parts and parts, which are sharp. This product contains electrically conductive parts. **Use caution with electrically conductive parts near or around power supplies, batteries and powered (live) circuits.**
- When the product is turned ON, activated or tested, some parts will move or rotate. **To avoid injuries to hands and fingers keep them away from any moving parts!**
- It is possible that an improperly connected or shorted circuit may cause overheating. **Should this happen, immediately disconnect the power supply or remove the batteries and do not touch anything until it cools down!** When everything is safe and cool, review the product tutorial to identify the cause.
- Only operate the product in accordance with the instructions and guidelines of this tutorial, otherwise parts may be damaged or you could be injured.
- Store the product in a cool dry place and avoid exposing the product to direct sunlight.
- After use, always turn the power OFF and remove or unplug the batteries before storing.

Any concerns?  support@freenove.com

About Freenove

Freenove provides open source electronic products and services worldwide.

Freenove is committed to assist customers in their education of robotics, programming and electronic circuits so that they may transform their creative ideas into prototypes and new and innovative products. To this end, our services include but are not limited to:

- Educational and Entertaining Project Kits for Robots, Smart Cars and Drones
- Educational Kits to Learn Robotic Software Systems for Arduino, Raspberry Pi and micro: bit
- Electronic Component Assortments, Electronic Modules and Specialized Tools
- **Product Development and Customization Services**

You can find more about Freenove and get our latest news and updates through our website:

<http://www.freenove.com>

sale@freenove.com

Copyright

All the files, materials and instructional guides provided are released under [Creative Commons Attribution-NonCommercial-ShareAlike 3.0 Unported License](https://creativecommons.org/licenses/by-nc-sa/3.0/). A copy of this license can be found in the folder containing the Tutorial and software files associated with this product.



This means you can use these resource in your own derived works, in part or completely but **NOT for the intent or purpose of commercial use.**

Freenove brand and logo are copyright of Freenove Creative Technology Co., Ltd. and cannot be used without written permission.



Any concerns? [✉ support@freenove.com](mailto:support@freenove.com)

Contents

Important Information.....	1
Contents.....	1
Preface.....	1
ESP32-WROVER	2
CH340	6
Arduino Software	19
Environment Configuration	22
Notes for GPIO.....	26
Chapter 1 LED	28
Project 1.1 Blink	28
Chapter 2 Camera Web Server.....	33
Project 2.1 Camera Web Server	33
What's next?	42

Preface

ESP32 is a micro control unit with integrated Wi-Fi launched by Espressif, which features strong properties and integrates rich peripherals. It can be designed and studied as an ordinary Single Chip Microcontroller (SCM) chip, or connected to the Internet and used as an Internet of Things device.

ESP32 can be developed using the Arduino platform, which will definitely make it easier for people who have learned Arduino to master. Moreover, the code of ESP32 is completely open-source, so beginners can quickly learn how to develop and design IOT smart household products including smart curtains, fans, lamps and clocks.

Generally, ESP32 projects consist of code and circuits. Don't worry even if you've never learned code and circuits, because we will gradually introduce the basic knowledge of C programming language and electronic circuits, from easy to difficult. Our products contain all the electronic components and modules needed to complete these projects. It's especially suitable for beginners.

We divide each project into four parts, namely Component List, Component Knowledge, Circuit and Code. Component List helps you to prepare material for the experiment more quickly. Component Knowledge allows you to quickly understand new electronic modules or components, while Circuit helps you understand the operating principle of the circuit. And Code allows you to easily master the use of ESP32 and accessory kit. After finishing all the projects in this tutorial, you can also use these components and modules to make products such as smart household, smart cars and robots to transform your creative ideas into prototypes and new and innovative products.

In addition, if you have any difficulties or questions with this tutorial or toolkit, feel free to ask for our quick and free technical support through support@freenove.com

ESP32-WROVER

ESP32-WROVER has launched a total of two antenna packages, PCB on-board antenna and IPEX antenna respectively. The PCB on-board antenna is an integrated antenna in the chip module itself, so it is convenient to carry and design. The IPEX antenna is a metal antenna derived from the integrated antenna of the chip module itself, which is used to enhance the signal of the module.

PCB on-board antenna

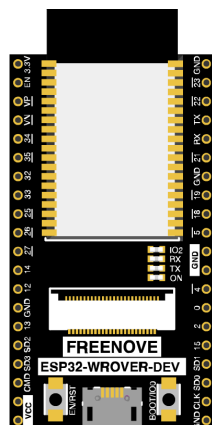


IPEX antenna

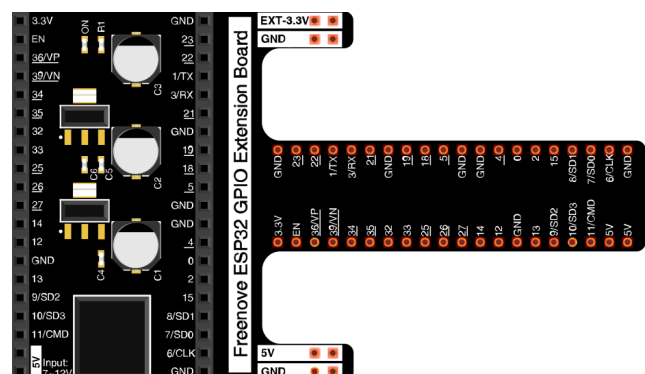


In this tutorial, the ESP32-WROVER is designed based on the IPEX antenna-packaged ESP32-WROVER module. And we also design an extension board, so that you can connect the wires with the toolkit more easily in accordance with the circuit diagram provided. The followings are their photos. All the projects in this tutorial are studied with this ESP32-WROVER.

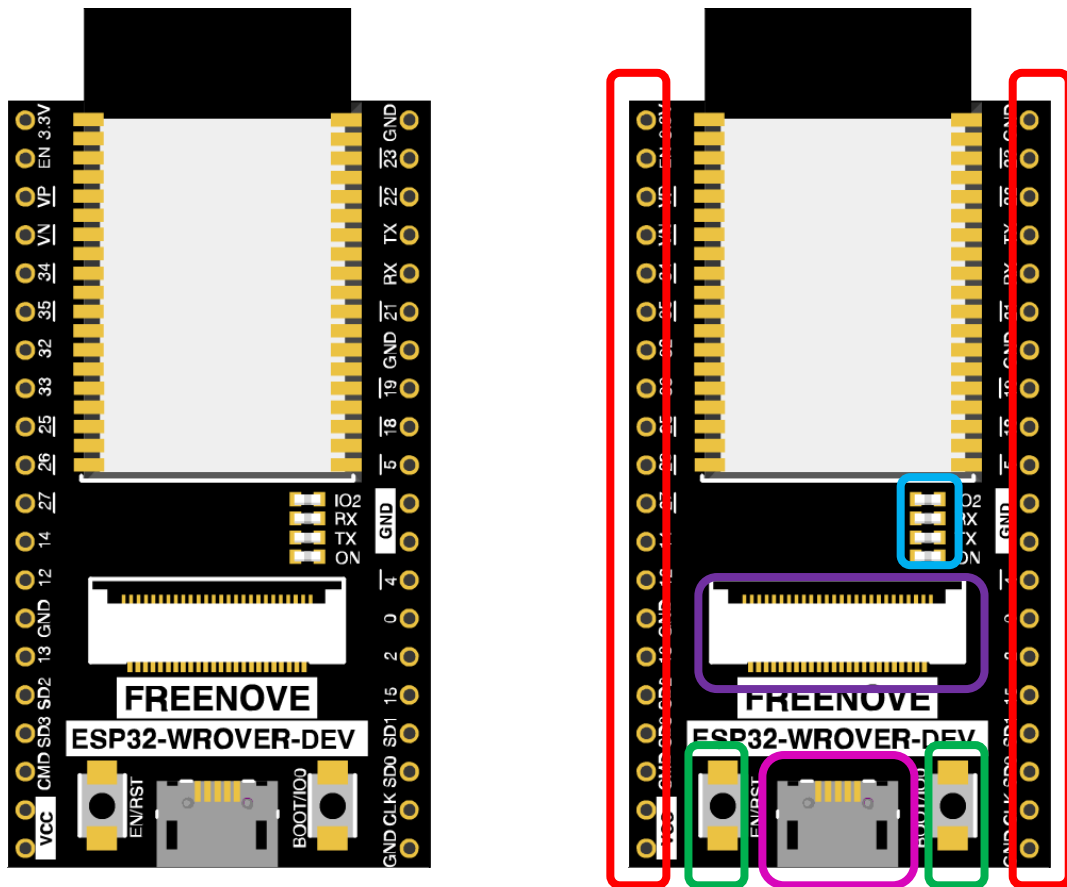
ESP32-WROVER








Extension board of the ESP32-WROVER



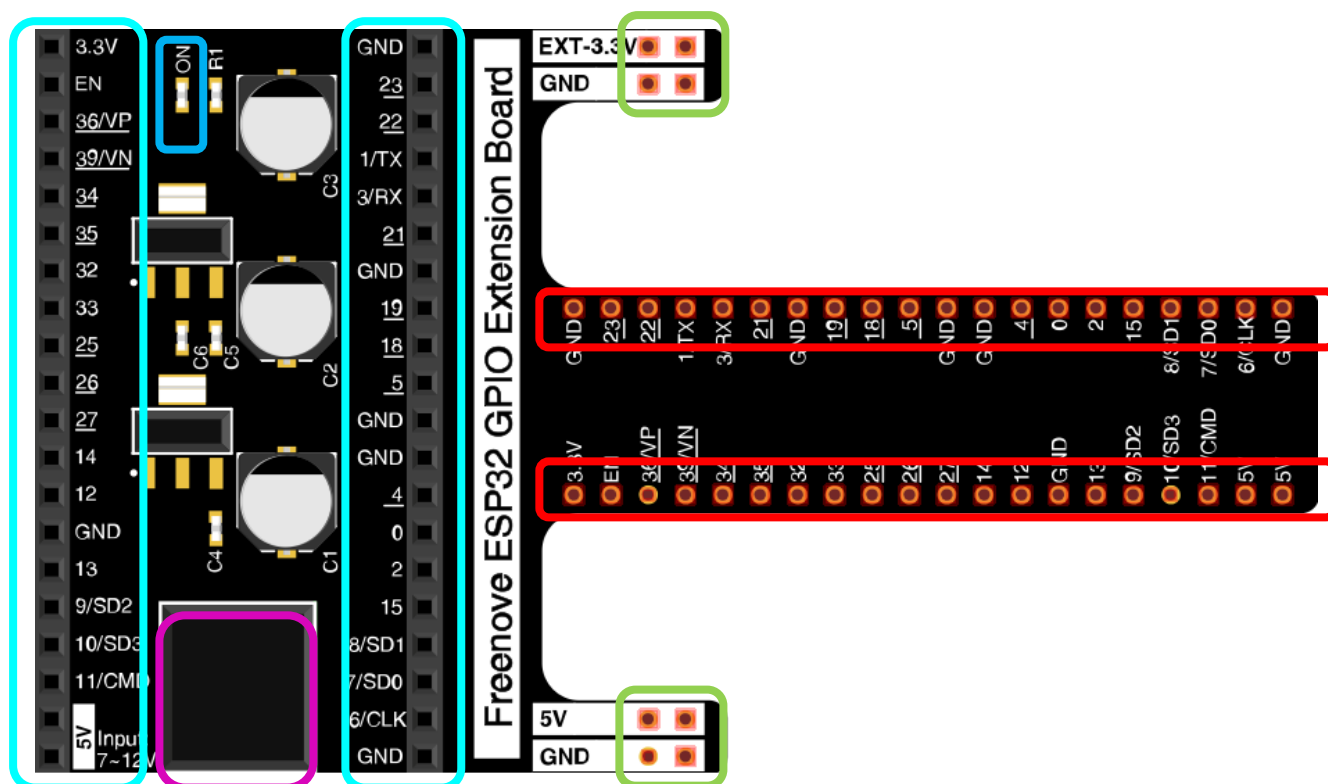
The hardware interfaces of ESP32-WROVER are distributed as follows:








Compare the left and right images. We've boxed off the resources on the ESP32-WROVER in different colors to facilitate your understanding of the ESP32-WROVER.

Box color	Corresponding resources introduction
	GPIO pin
	LED indicator
	Camera interface
	Reset button, Boot mode selection button
	USB port

The hardware interfaces of ESP32-WROVER are distributed as follows:



We've boxed off the resources on the ESP32-WROVER in different colors to facilitate your understanding of the ESP32-WROVER.

Box color	Corresponding resources introduction
	GPIO pin
	LED indicator
	GPIO interface of development board
	power supplied by the extension board
	External power supply

In ESP32, GPIO is an interface to control peripheral circuit. For beginners, it is necessary to learn the functions of each GPIO. The following is an introduction to the GPIO resources of the ESP32-WROVER development board.

In the following projects, we only use USB cable to power ESP32-WROVER by default.

In the whole tutorial, we don't use T extension to power ESP32-WROVER. So 5V and 3.3V (including EXT 3.3V) on the extension board are provided by ESP32-WROVER.

We can also use DC jack of extension board to power ESP32-WROVER. In this way, 5v and EXT 3.3v on extension board are provided by external power resource.

Name	No.	Type	Function
GND	1	P	Ground
3V3	2	P	Power supply
EN	3	I	Module-enable signal. Active high.
SENSOR_VP	4	I	GPIO36, ADC1_CH0, RTC_GPIO0
SENSOR_VN	5	I	GPIO39, ADC1_CH3, RTC_GPIO3
IO34	6	I	GPIO34, ADC1_CH6, RTC_GPIO4
IO35	7	I	GPIO35, ADC1_CH7, RTC_GPIO5
IO32	8	I/O	GPIO32, XTAL_32K_P (32.768 kHz crystal oscillator input), ADC1_CH4, TOUCH9, RTC_GPIO9
IO33	9	I/O	GPIO33, XTAL_32K_N (32.768 kHz crystal oscillator output), ADC1_CH5, TOUCH8, RTC_GPIO8
IO25	10	I/O	GPIO25, DAC_1, ADC2_CH8, RTC_GPIO6, EMAC_RXD0
IO26	11	I/O	GPIO26, DAC_2, ADC2_CH9, RTC_GPIO7, EMAC_RXD1
IO27	12	I/O	GPIO27, ADC2_CH7, TOUCH7, RTC_GPIO17, EMAC_RX_DV
IO14	13	I/O	GPIO14, ADC2_CH6, TOUCH6, RTC_GPIO16, MTMS, HSPICLK, HS2_CLK, SD_CLK, EMAC_TXD2
IO12 ¹	14	I/O	GPIO12, ADC2_CH5, TOUCH5, RTC_GPIO15, MTDI, HSPIQ, HS2_DATA2, SD_DATA2, EMAC_TXD3
GND	15	P	Ground
IO13	16	I/O	GPIO13, ADC2_CH4, TOUCH4, RTC_GPIO14, MTCK, HSPID, HS2_DATA3, SD_DATA3, EMAC_RX_ER
SHD/SD2 ²	17	I/O	GPIO9, SD_DATA2, SPIHD, HS1_DATA2, U1RXD
SWP/SD3 ²	18	I/O	GPIO10, SD_DATA3, SPIWP, HS1_DATA3, U1TXD
SCS/CMD ²	19	I/O	GPIO11, SD_CMD, SPICS0, HS1_CMD, U1RTS
SCK/CLK ²	20	I/O	GPIO6, SD_CLK, SPICLK, HS1_CLK, U1CTS
SDO/SD0 ²	21	I/O	GPIO7, SD_DATA0, SPIQ, HS1_DATA0, U2RTS
SDI/SD1 ²	22	I/O	GPIO8, SD_DATA1, SPID, HS1_DATA1, U2CTS
IO15	23	I/O	GPIO15, ADC2_CH3, TOUCH3, MTDO, HSPICS0, RTC_GPIO13, HS2_CMD, SD_CMD, EMAC_RXD3
IO2	24	I/O	GPIO2, ADC2_CH2, TOUCH2, RTC_GPIO12, HSPWP, HS2_DATA0, SD_DATA0
IO0	25	I/O	GPIO0, ADC2_CH1, TOUCH1, RTC_GPIO11, CLK_OUT1, EMAC_TX_CLK
IO4	26	I/O	GPIO4, ADC2_CH0, TOUCH0, RTC_GPIO10, HSPHD, HS2_DATA1, SD_DATA1, EMAC_TX_ER
NC1	27	-	-
NC2	28	-	-
IO5	29	I/O	GPIO5, VSPICS0, HS1_DATA6, EMAC_RX_CLK
IO18	30	I/O	GPIO18, VSPICLK, HS1_DATA7
IO19	31	I/O	GPIO19, VSPIQ, U0CTS, EMAC_TXD0
NC	32	-	-
IO21	33	I/O	GPIO21, VSPHD, EMAC_TX_EN
RXD0	34	I/O	GPIO3, U0RXD, CLK_OUT2
TXD0	35	I/O	GPIO1, U0TXD, CLK_OUT3, EMAC_RXD2
IO22	36	I/O	GPIO22, VSPWP, U0RTS, EMAC_TXD1
IO23	37	I/O	GPIO23, VSPID, HS1_STROBE
GND	38	P	Ground

Notice:

- GPIO12 is internally pulled high in the module and is not recommended for use as a touch pin.
- Pins SCK/CLK, SDO/SD0, SDI/SD1, SHD/SD2, SWP/SD3 and SCS/CMD, namely, GPIO6 to GPIO11 are connected to the SPI flash integrated on the module and are not recommended for other uses.

For more information, please visit: https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf

Any concerns? ✉ support@freenove.com

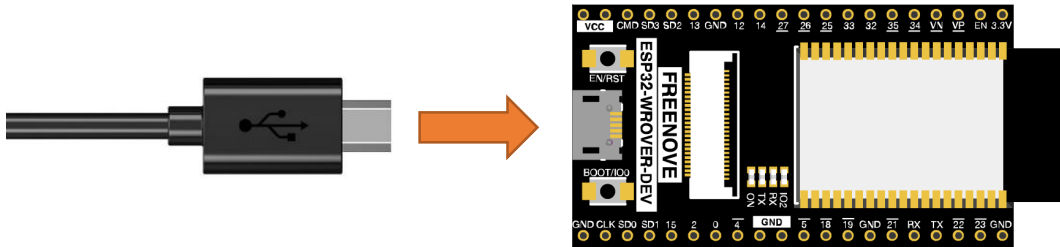
CH340

ESP32 uses CH340 to download codes. So before using it, we need to install CH340 driver in our computers.

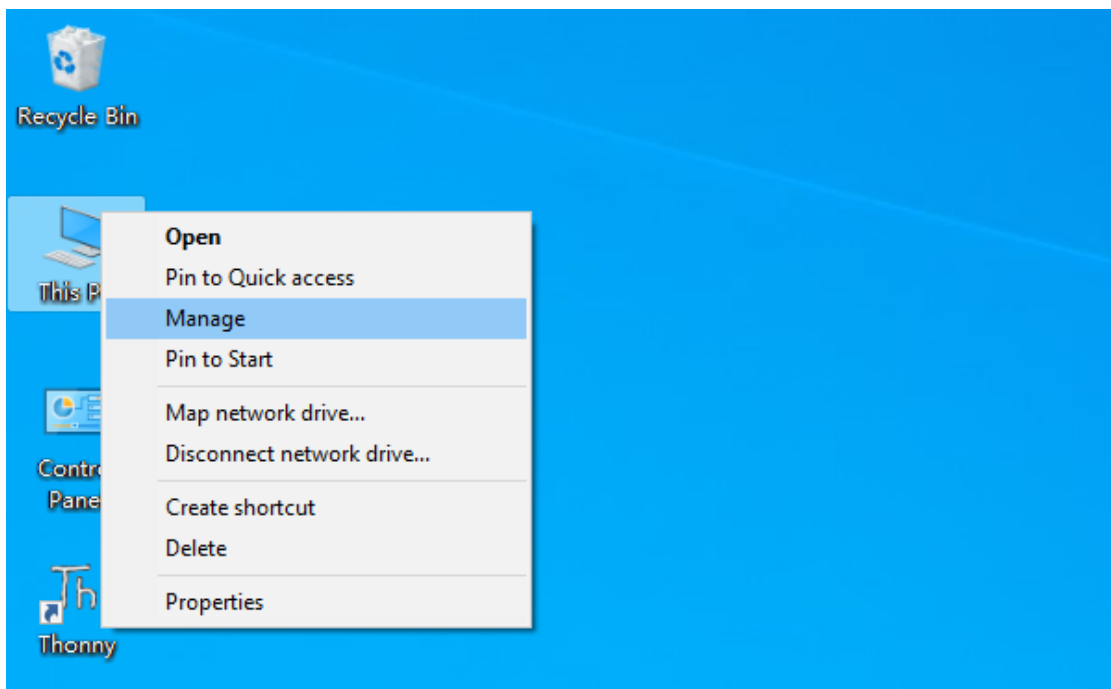
Windows

Check whether CH340 has been installed

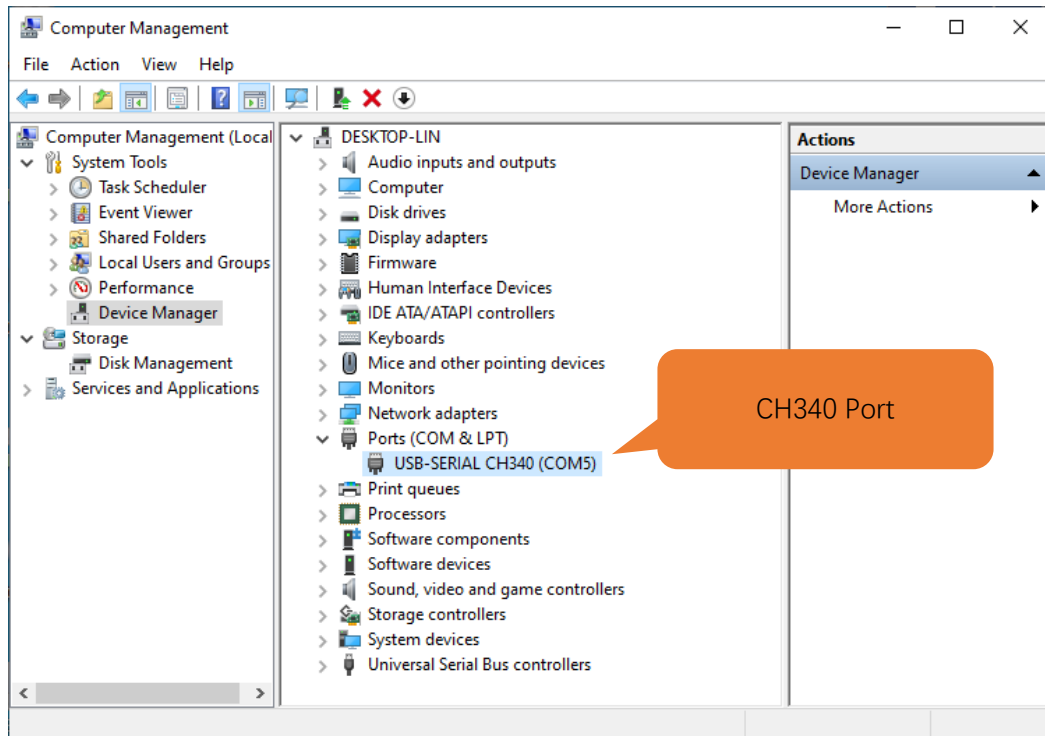
1. Connect your computer and ESP32 with a USB cable.



2. Turn to the main interface of your computer, select "This PC" and right-click to select "Manage".



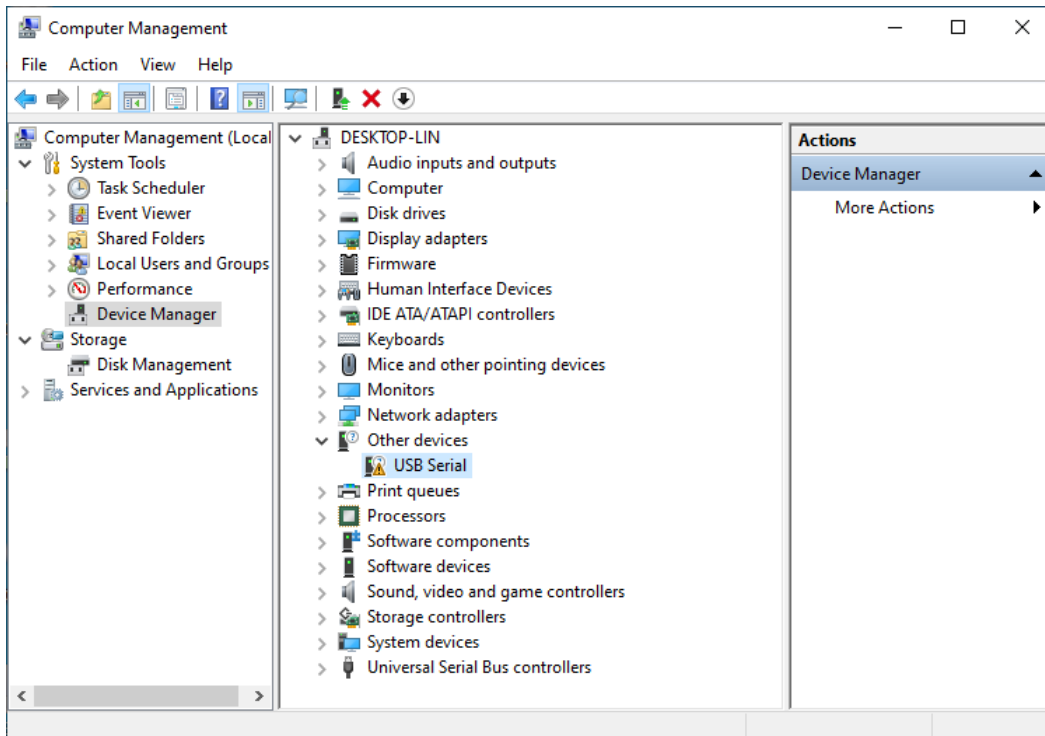
- Click "Device Manager". If your computer has installed CH340, you can see "USB-SERIAL CH340 (COMx)". And you can click [here](#) to move to the next step.



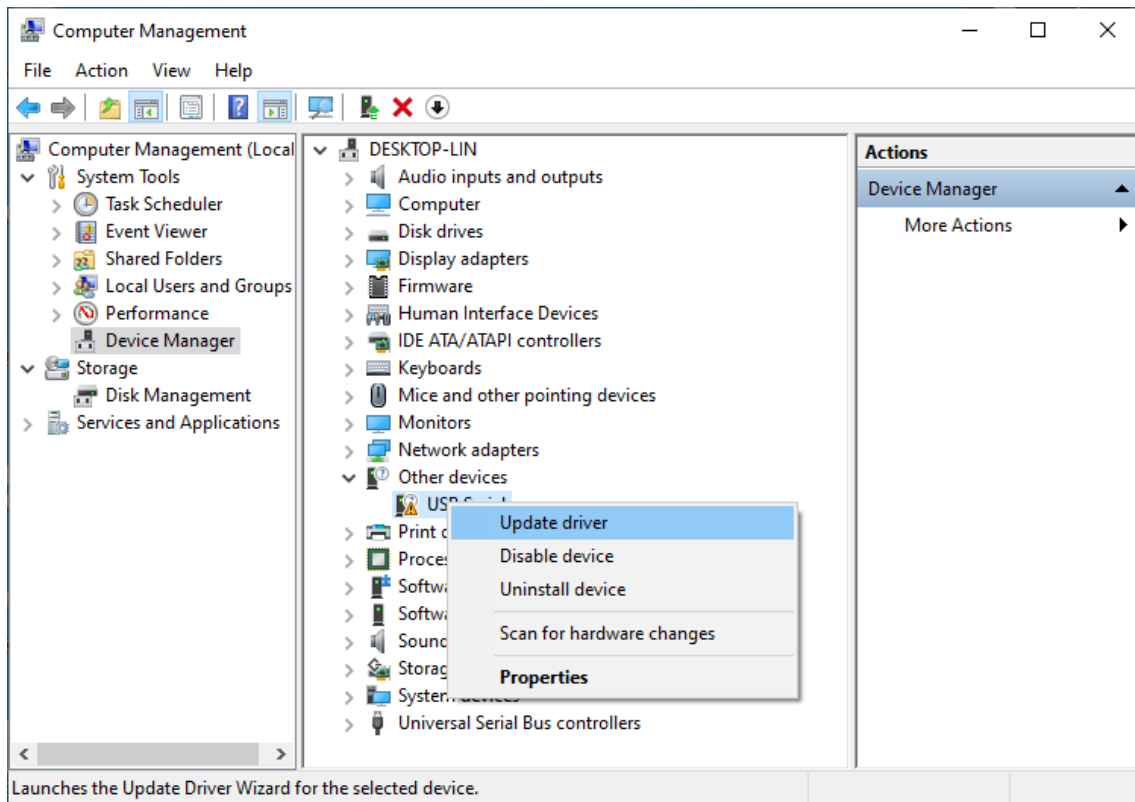
Installing CH340

Method1

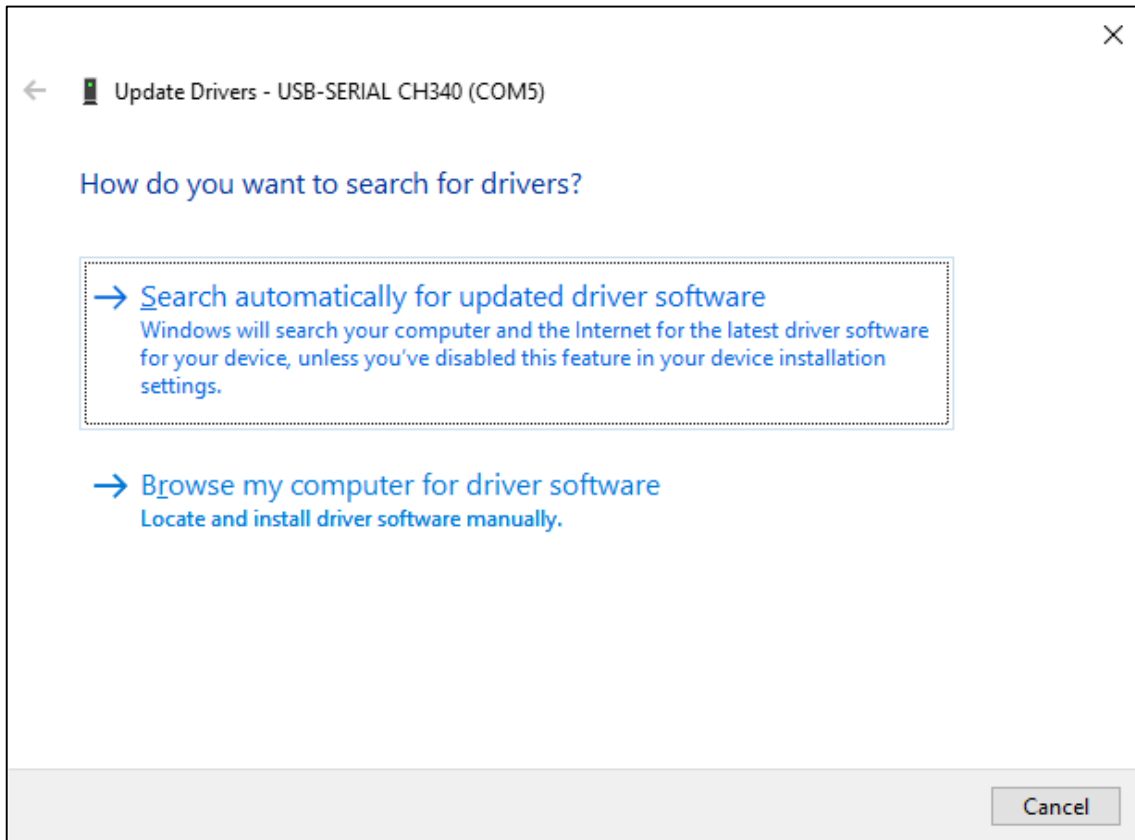
- If you have not yet installed CH340, you'll see the following interface.



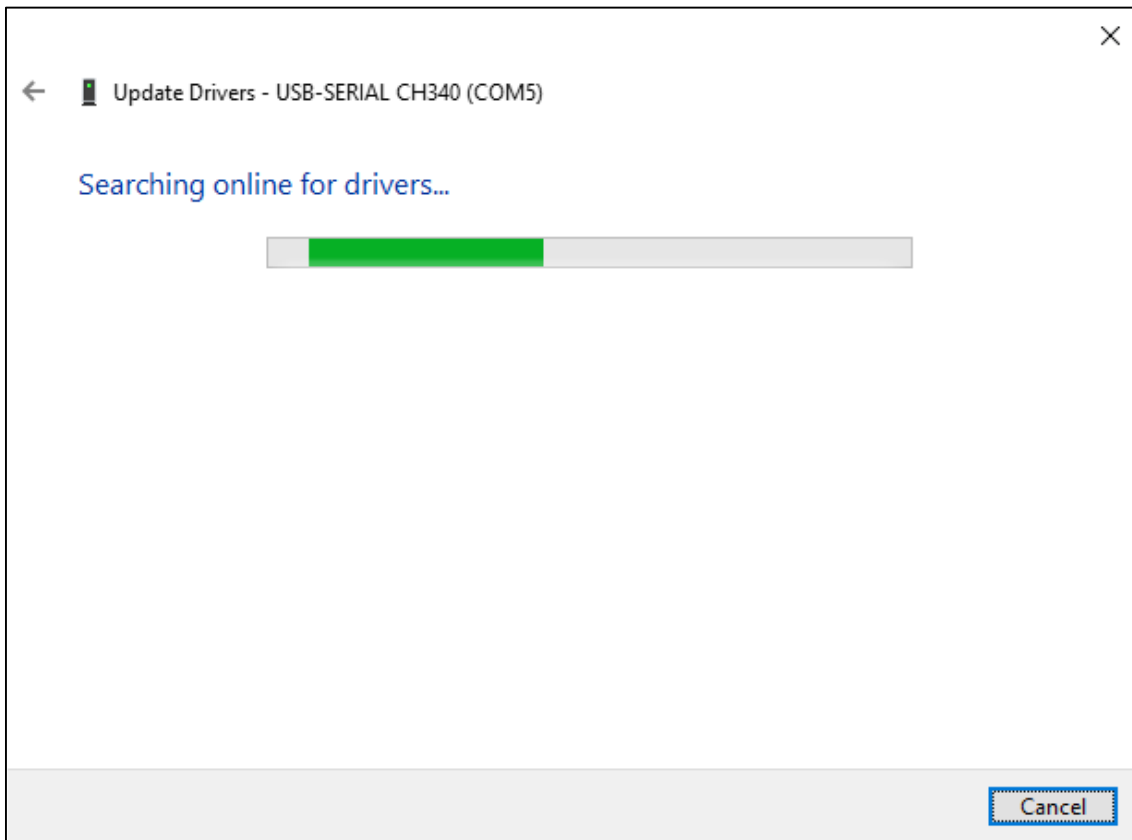
- Click “USB Serial” and right-click to select “Update driver”.



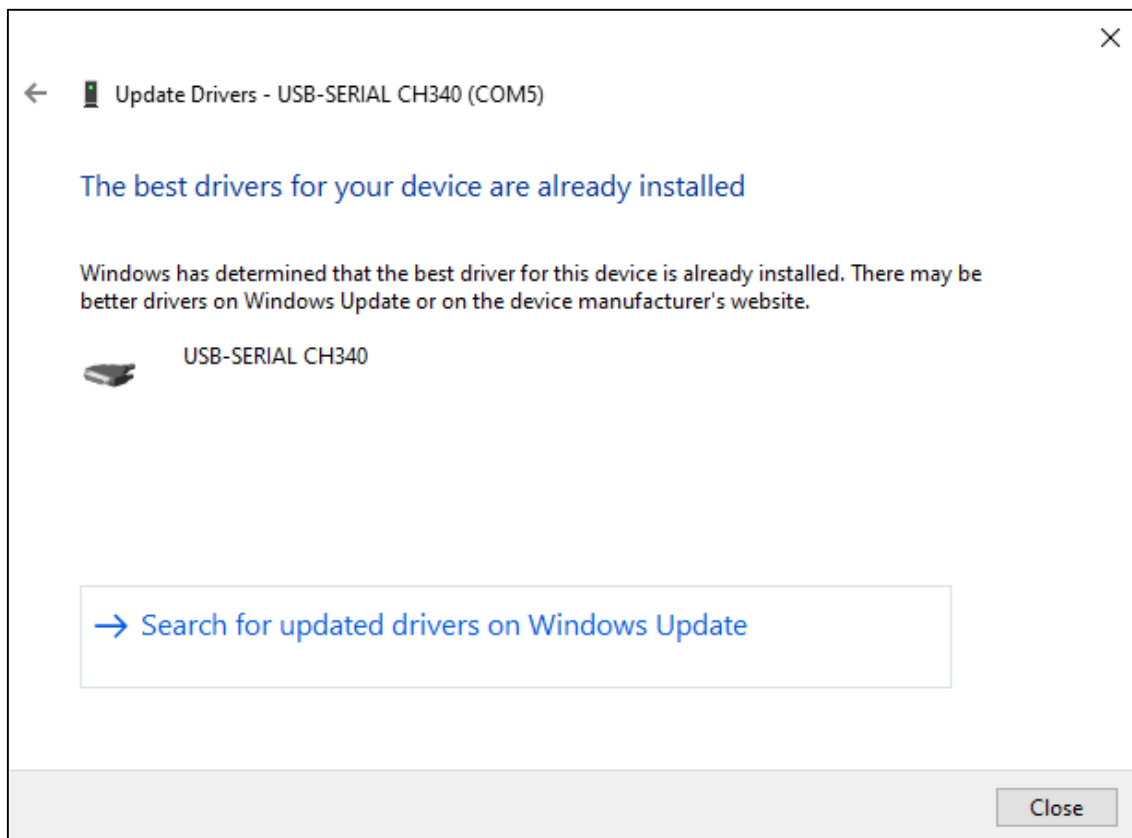
- Click “Search automatically for updated driver software”.



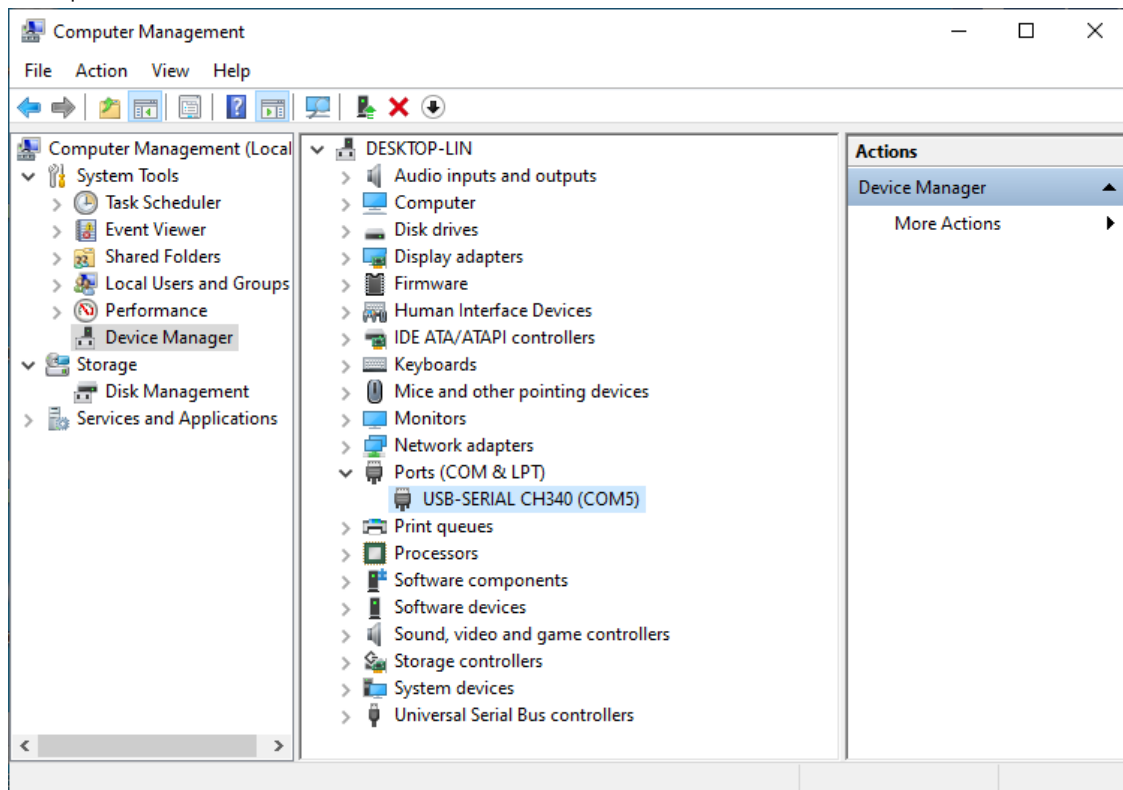
4. Wait for CH340 to finish installation.



5. When you see the following interface, it indicates that CH340 has been installed to your computer. You can close the interface.



6. When ESP32 is connected to computer, you can see the following interface. Click here to move to the next step.



Method2

- First, download CH340 driver, click <http://www.wch-ic.com/search?q=CH340&t=downloads> to download the appropriate one based on your operating system.

index / search / search CH340

All (14)

Downloads (7)

Products (4)

Application (2)

Video (1)

News (0)

keyword CH340

Downloads(7)

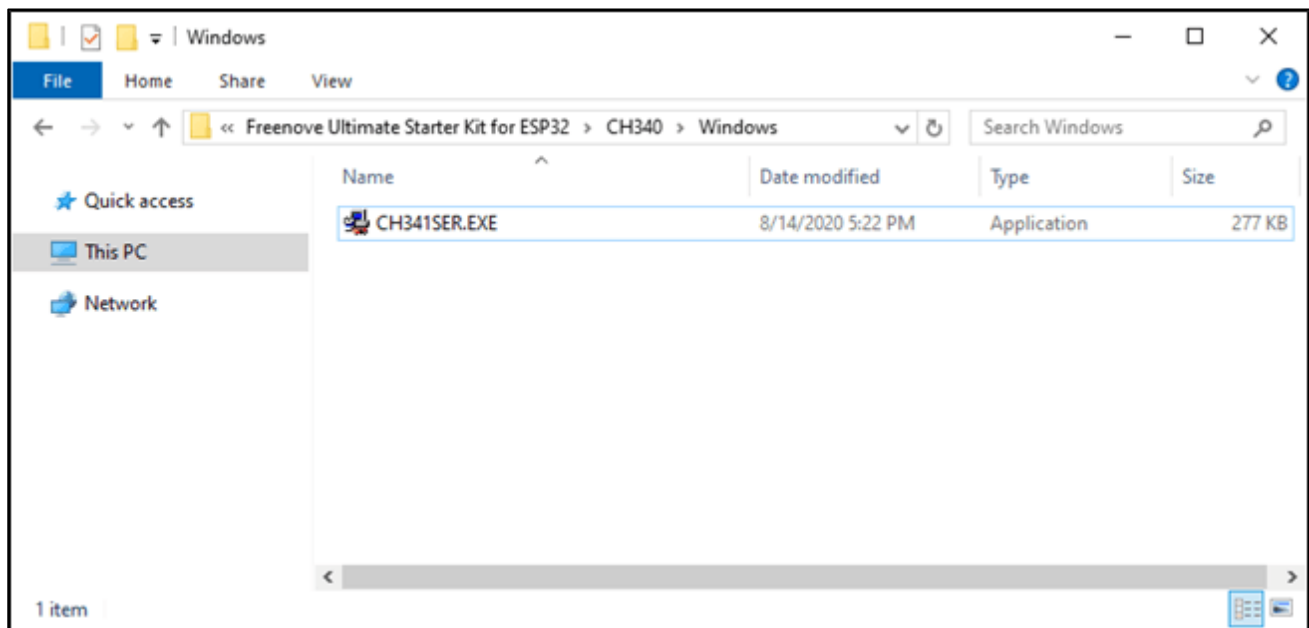
file category	file content	version	upload time
Driver&Tools			
CH341SER.EXE	CH340/CH341 USB to serial port Windows driver, supports 32/64-bit Windows 10/8.1/8/7/VISTA/XP, Server 2016/2012/2008/2003, 2000/ME/98	3.5	2019-03-18
CH341SER.ZIP	CH340/CH341 USB to serial port Windows driver, includes DLL dynamic library and non-standard baud rate settings and other instructions. Supports 32/64-bit Windows 10/8.1/8/7/VISTA/XP, Server 2016/2012/2008/2003, 2000/ME/98	3.5	2019-03-05
CH341SER_ANDROID...	CH340/CH341 USB to serial port Android free drive application library, for Android OS 3.1 and above version which supports USB Host mode already, no need to load Android kernel driver, no root privileges. Contains apk, lib library, file (Host Driver), App Demo Example (USB to UART Demo)	1.6	2019-04-19
CH341SER_LINUX...	CH340/CH341 USB to serial port LINUX driver	1.5	2018-03-18
CH341SER_MAC.ZI...	CH340/CH341 USB to serial port MAC OS driver	1.5	2018-07-05
Others			
PRODUCT_GUIDE.P...	Electronic selection of product selection manual, please refer to related product technical manual for more technical information.	1.4	2018-12-29
InstallNoteOn64...	Instructions for the driver after 18 years of August cannot be installed under some 64-bit WIN7 (English)	1.0	2019-01-10

If you would not like to download the installation package, you can open

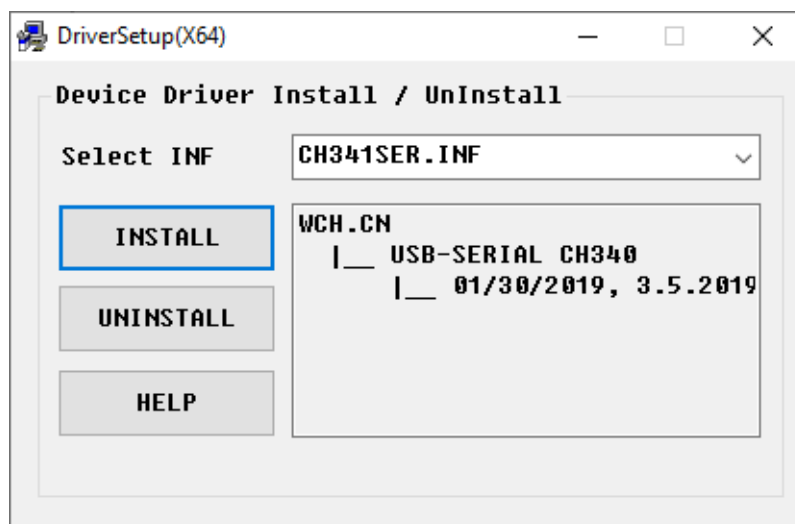
“Freenove_ESP32_WROVER_Starter_Kit_for_ESP32/CH340”, we have prepared the installation package.

Name	Date modified	Type	Size
Linux	8/14/2020 5:24 PM	File folder	
MAC	8/14/2020 5:23 PM	File folder	
Windows	8/14/2020 5:23 PM	File folder	

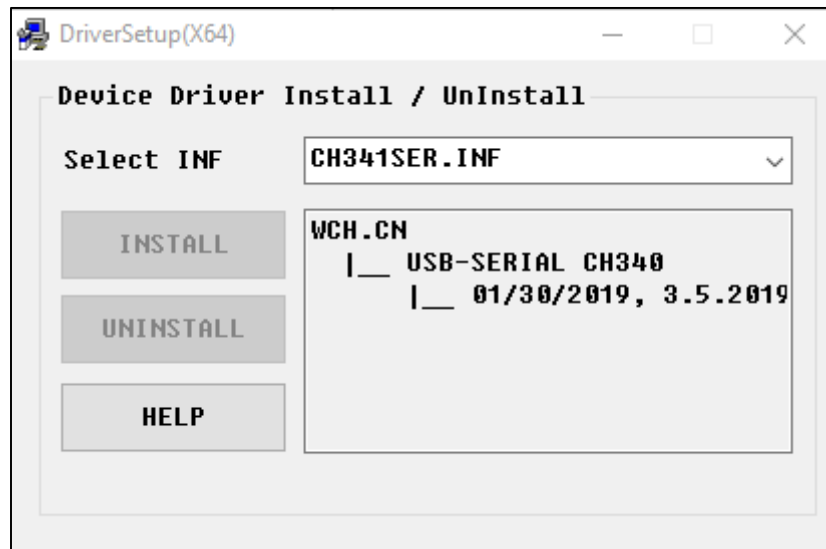
- Open the folder "Freenove_ESP32_WROVER_Starter_Kit_for_ESP32/CH340/Windows/"



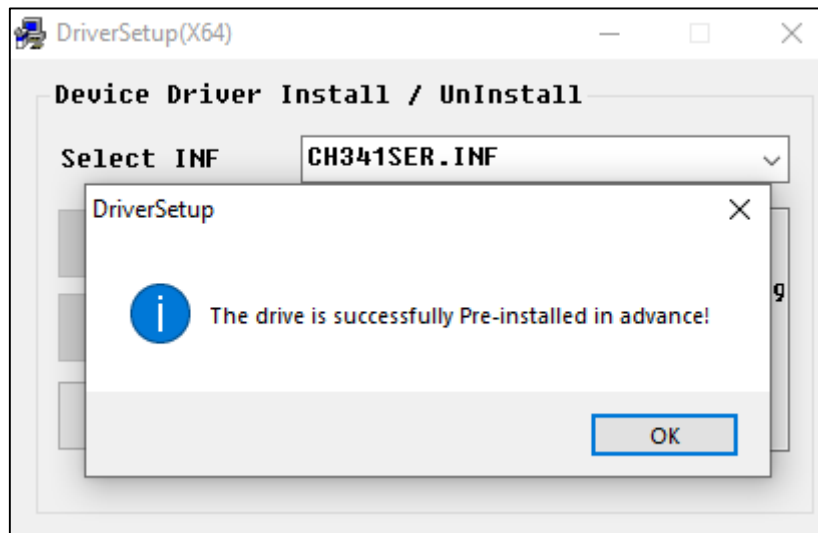
- Double click "CH341SER.EXE".



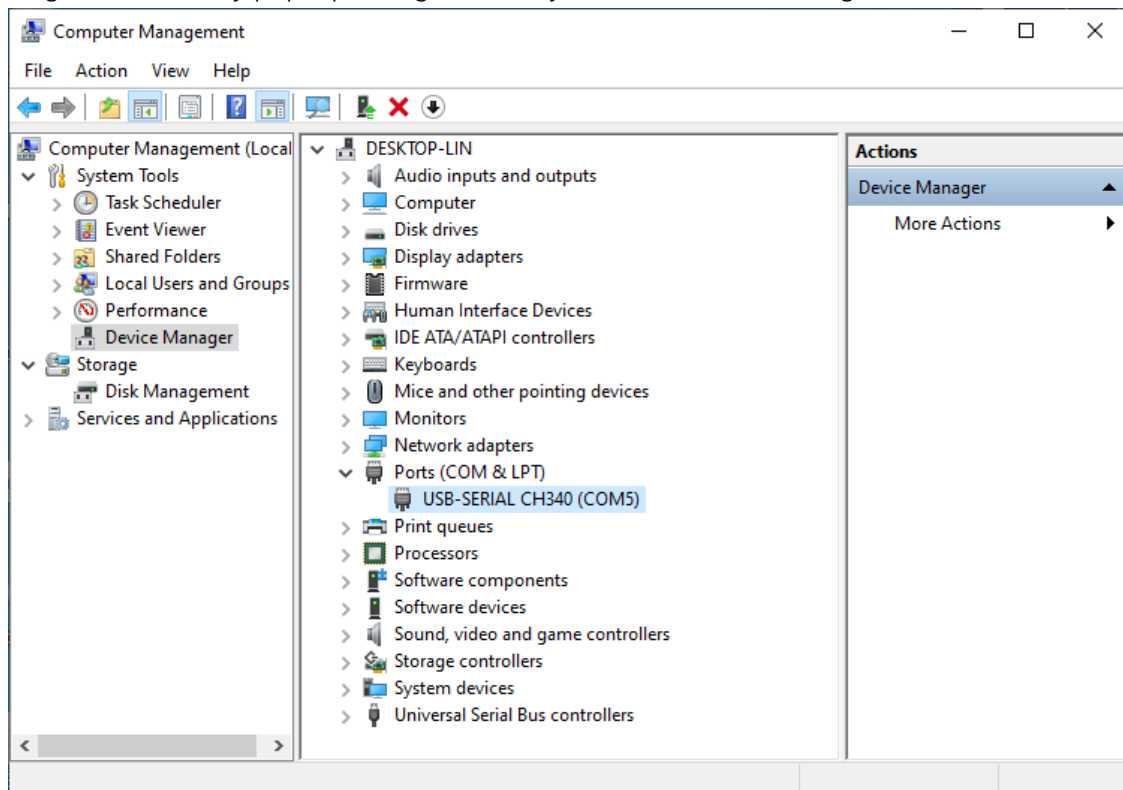
- Click "INSTALL" and wait for the installation to complete.



- Install successfully. Close all interfaces.



6. When ESP32 is connected to computer, select “This PC”, right-click to select “Manage” and click “Device Manager” in the newly pop-up dialog box, and you can see the following interface.



7. So far, CH340 has been installed successfully. Close all dialog boxes.

MAC

First, download CH340 driver, click <http://www.wch-ic.com/search?q=CH340&t=downloads> to download the appropriate one based on your operating system.

The screenshot shows the WCH website with a search bar containing 'keyword ch340'. The left sidebar lists categories: All (14), Downloads (7), Products (4), Application (2), Video (1), and News (0). The main content area shows 'Downloads (7)' with a table of results. An orange callout box labeled 'Windows' points to the 'CH341SER.EXE' entry. Another orange callout box labeled 'Linux' points to the 'CH341SER_LINUX...' entry. A third orange callout box labeled 'MAC' points to the 'CH341SER_MAC.ZI...' entry.

file category	file content	version	upload time
Driver&Tools			
CH341SER.EXE	CH340/CH341 USB to serial port Windows driver, supports 32/64-bit Windows 10/8.1/8/7/VISTA/XP, Server 2016/2012/2008/2003, 2000/ME/98	3.5	2019-03-18
CH341SER.ZIP	CH340/CH341 USB to serial port Windows driver, includes DLL dynamic library and non-standard baud rate settings and other instructions. Supports 32/64-bit Windows 10/8.1/8/7/VISTA/XP, Server 2016/2012/2008/2003, 2000/ME/98	3.5	2019-03-05
CH341SER_ANDROID...	CH340/CH341 USB to serial port Android free drive application library, for Android OS 3.1 and above version which supports USB Host mode already, no need to load Android kernel driver, no root privileges. Contains apk, lib library file (Java Driver), App Demo Example, AT Demo SDK).	1.6	2019-04-19
CH341SER_LINUX...	CH340/CH341 USB to serial port LINUX driver	1.5	2018-03-18
CH341SER_MAC.ZI...	CH340/CH341 USB to serial port MAC OS driver	1.5	2018-07-05

If you would not like to download the installation package, you can open

“Freenove_ESP32_WROVER_Starter_Kit_for_ESP32/CH340”, we have prepared the installation package.

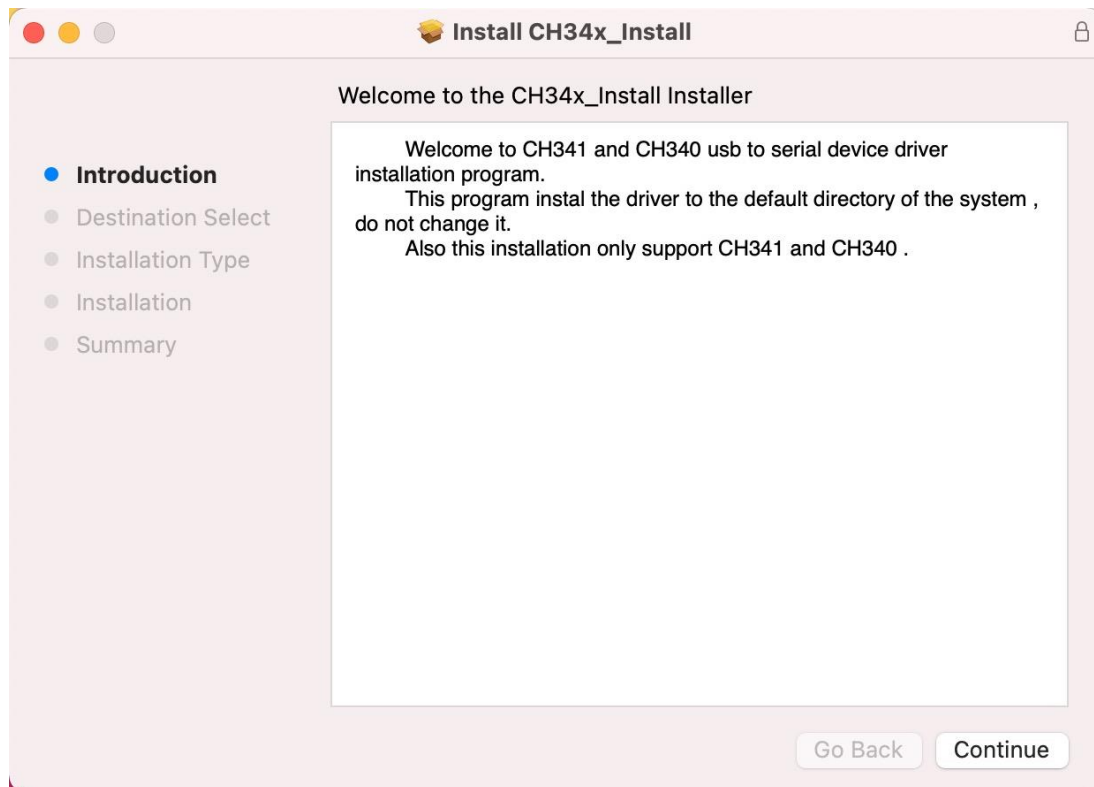
Second, open the folder “Freenove_ESP32_WROVER_Starter_Kit_for_ESP32/CH340/MAC/”

The screenshot shows a Mac file explorer window titled 'MAC'. The left sidebar shows the 'Favorites' section with 'Downloads' selected. The main pane shows a list of files: 'CH34x_Install_V1.5.pkg' (26 KB, Installer package) and 'ReadMe.pdf' (146 KB, Adobe...ocument). An orange callout box labeled 'Run it.' points to the 'CH34x_Install_V1.5.pkg' file.

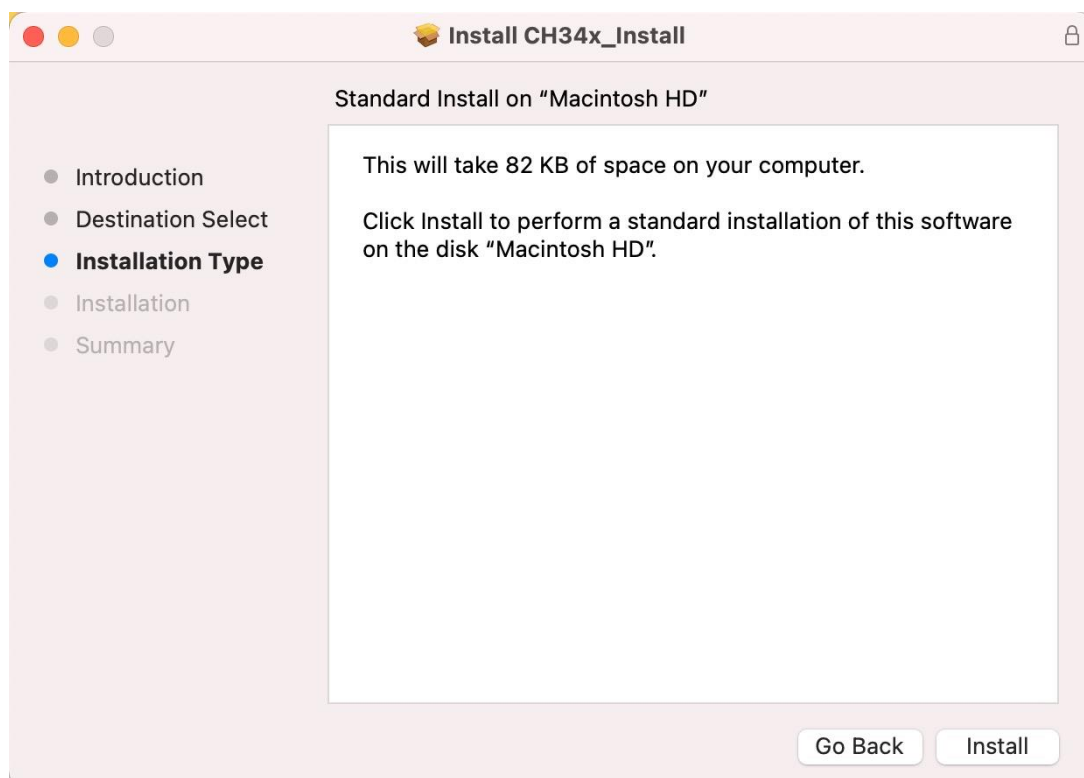
Name	Modified	Size	Kind
CH34x_Install_V1.5.pkg	2018 at 2:00 PM	26 KB	Installer package
ReadMe.pdf	Dec 8, 2016 at 4:09 PM	146 KB	Adobe...ocument

Any concerns? [✉ support@freenove.com](mailto:support@freenove.com)

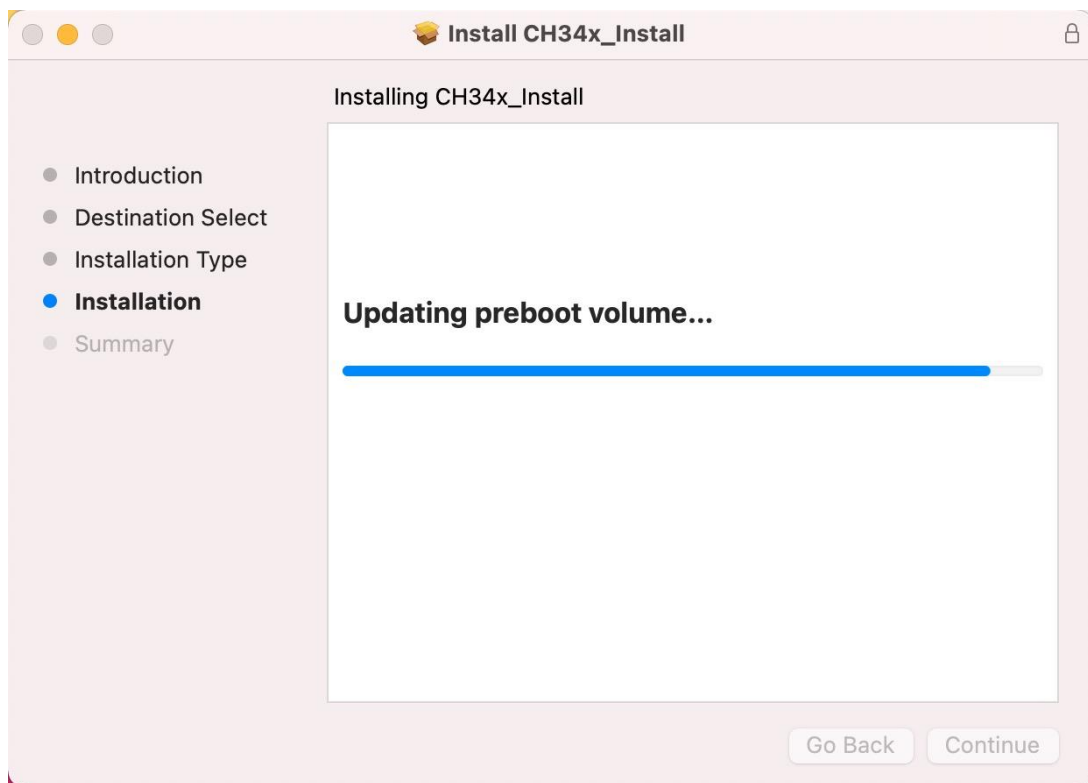
Third, click Continue.



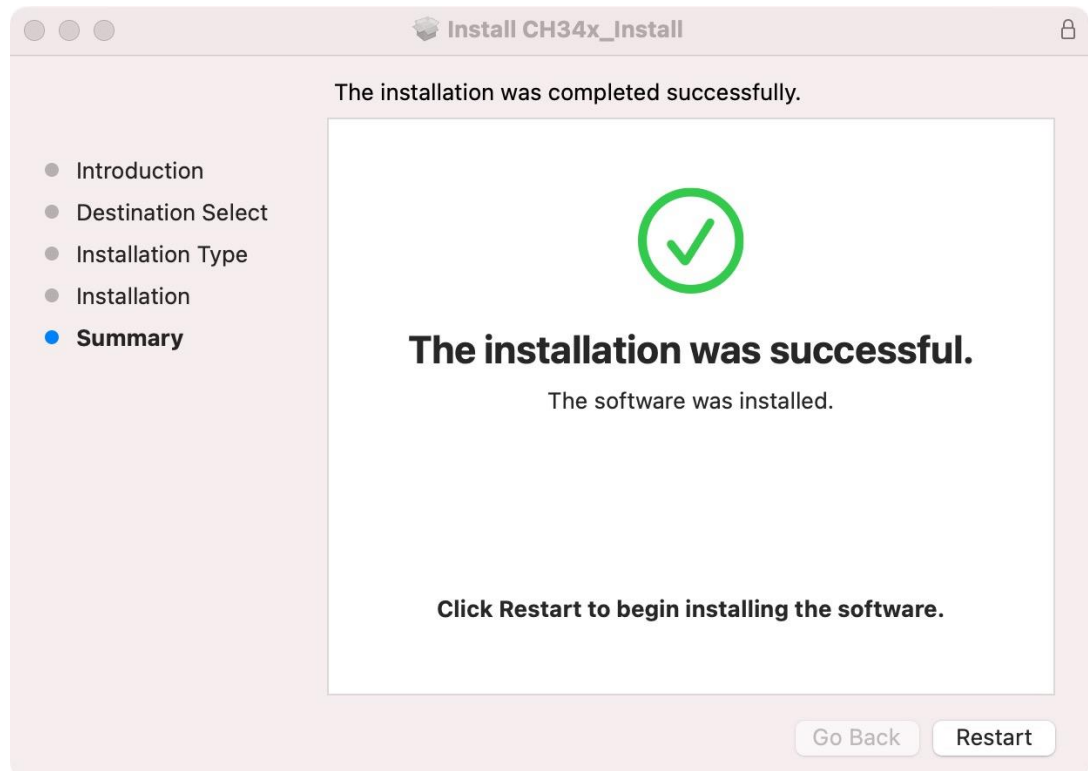
Fourth, click Install.



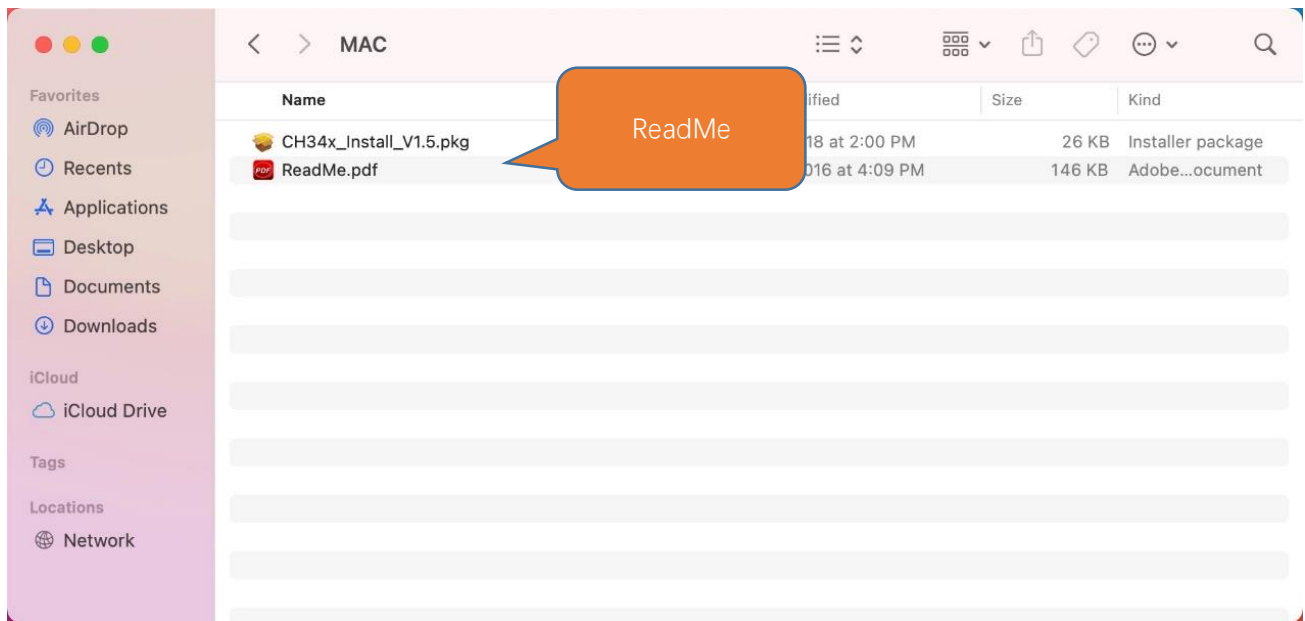
Then, waiting Finish.



Finally, restart your PC.



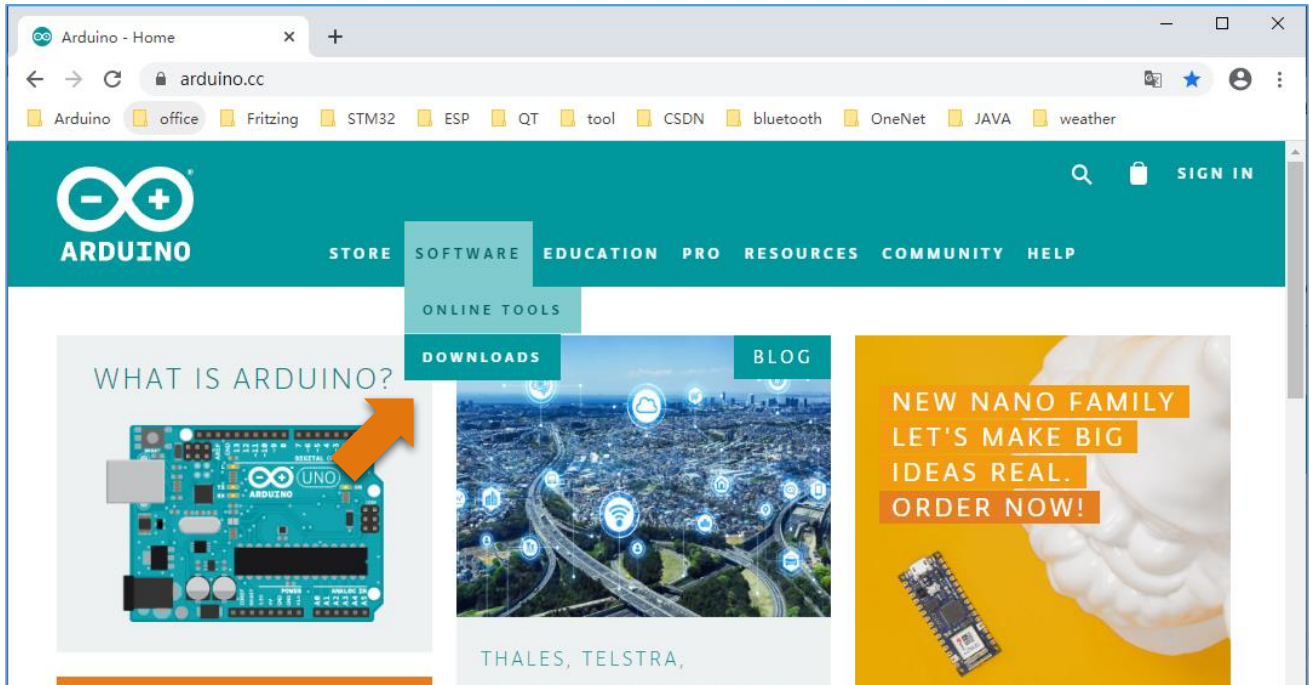
If you still haven't installed the CH340 by following the steps above, you can view readme.pdf to install it.



Arduino Software

Arduino Software (IDE) is used to write and upload the code for Arduino Board.

First, install Arduino Software (IDE): visit <https://www.arduino.cc>, click "Download" to enter the download page.



Select and download corresponding installer according to your operating system. If you are a windows user, please select the "Windows Installer" to download to install the driver correctly.



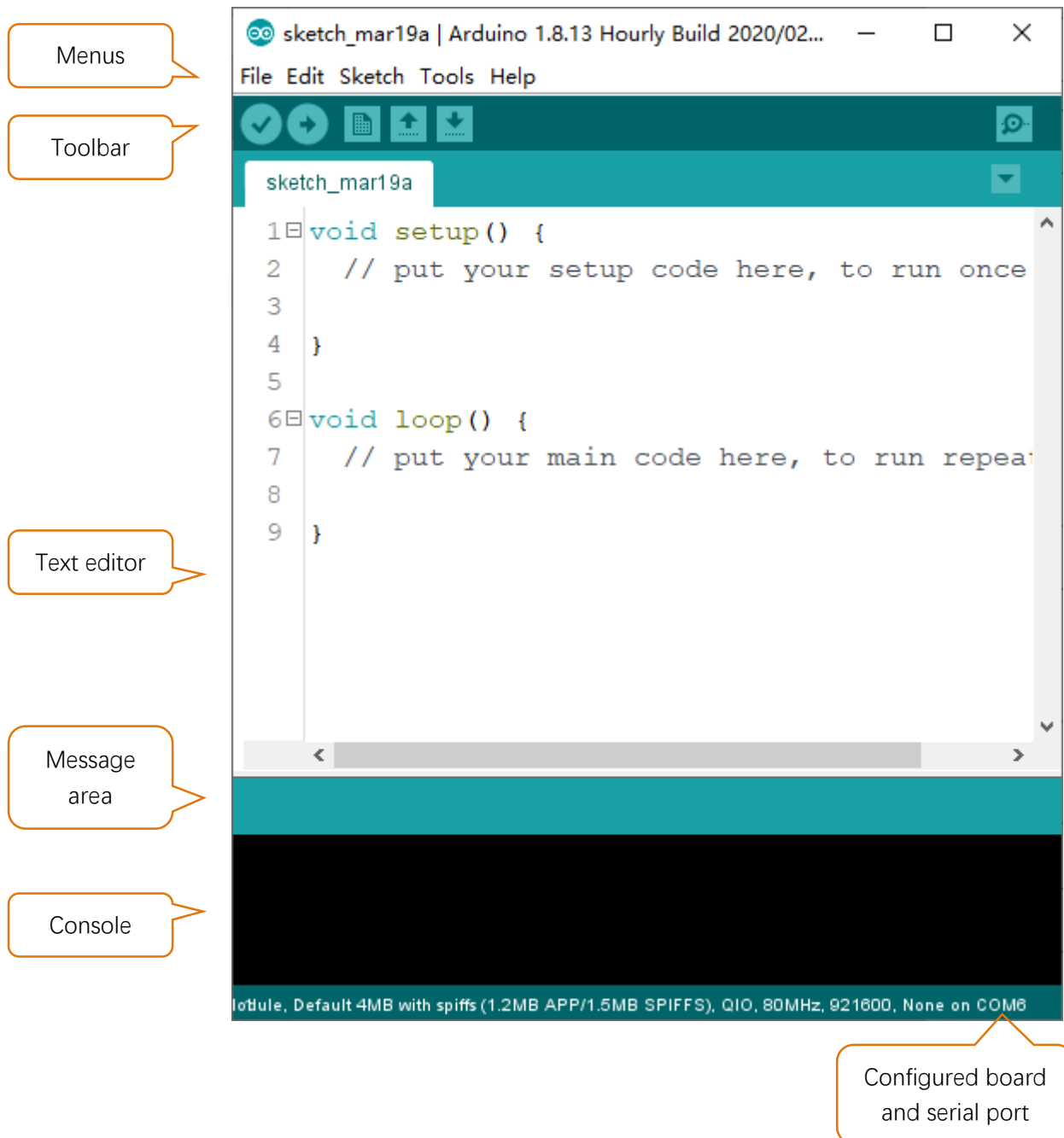
After the download completes, run the installer. For Windows users, there may pop up an installation dialog box of driver during the installation process. When it popes up, please allow the installation.

After installation is complete, an Arduino Software shortcut will be generated in the desktop. Run the Arduino Software.

Any concerns? ✉ support@freenove.com



The interface of Arduino Software is as follows:



Programs written with Arduino Software (IDE) are called **sketches**. These sketches are written in the text editor and saved with the file extension **.ino**. The editor has features for cutting/pasting and searching/replacing text. The message area gives feedback while saving and exporting and also displays errors. The console displays text output by the Arduino Software (IDE), including complete error messages and other information. The bottom right-hand corner of the window displays the configured board and serial port. The toolbar buttons allow you to verify and upload programs, create, open, and save sketches, and open the serial monitor.



Verify

Check your code for compile errors .



Upload

Compile your code and upload them to the configured board.



New

Create a new sketch.



Open

Present a menu of all the sketches in your sketchbook. Clicking one will open it within the current window and overwrite its content.



Save

Save your sketch.



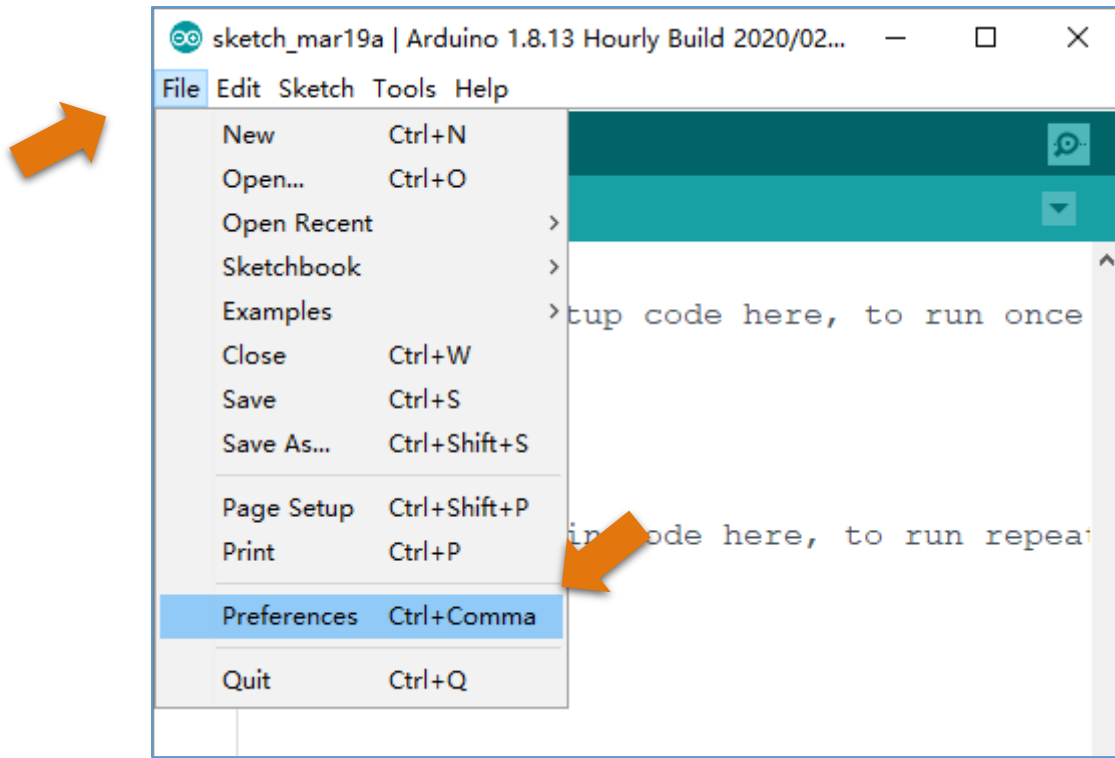
Serial Monitor

Open the serial monitor.

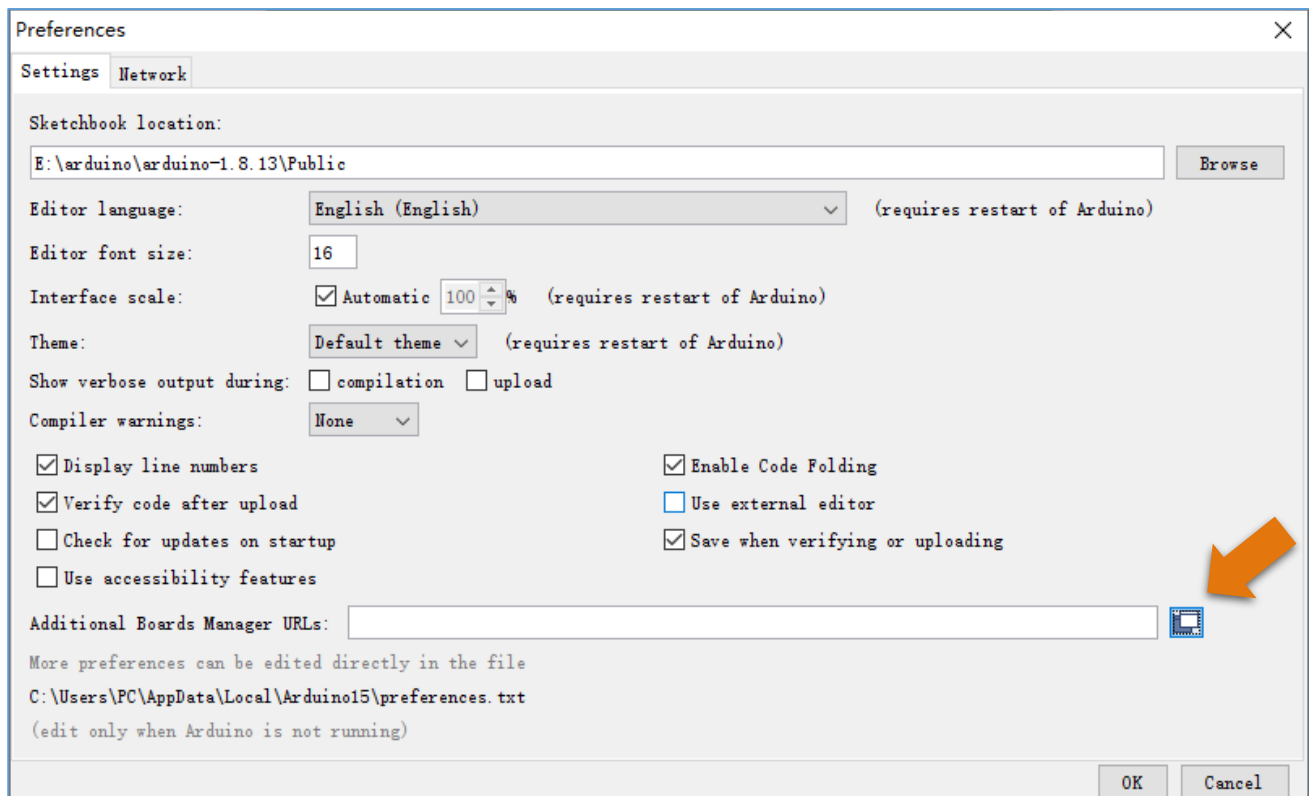
Additional commands are found within the five menus: File, Edit, Sketch, Tools, Help. The menus are context sensitive, which means only those items relevant to the work currently being carried out are available.

Environment Configuration

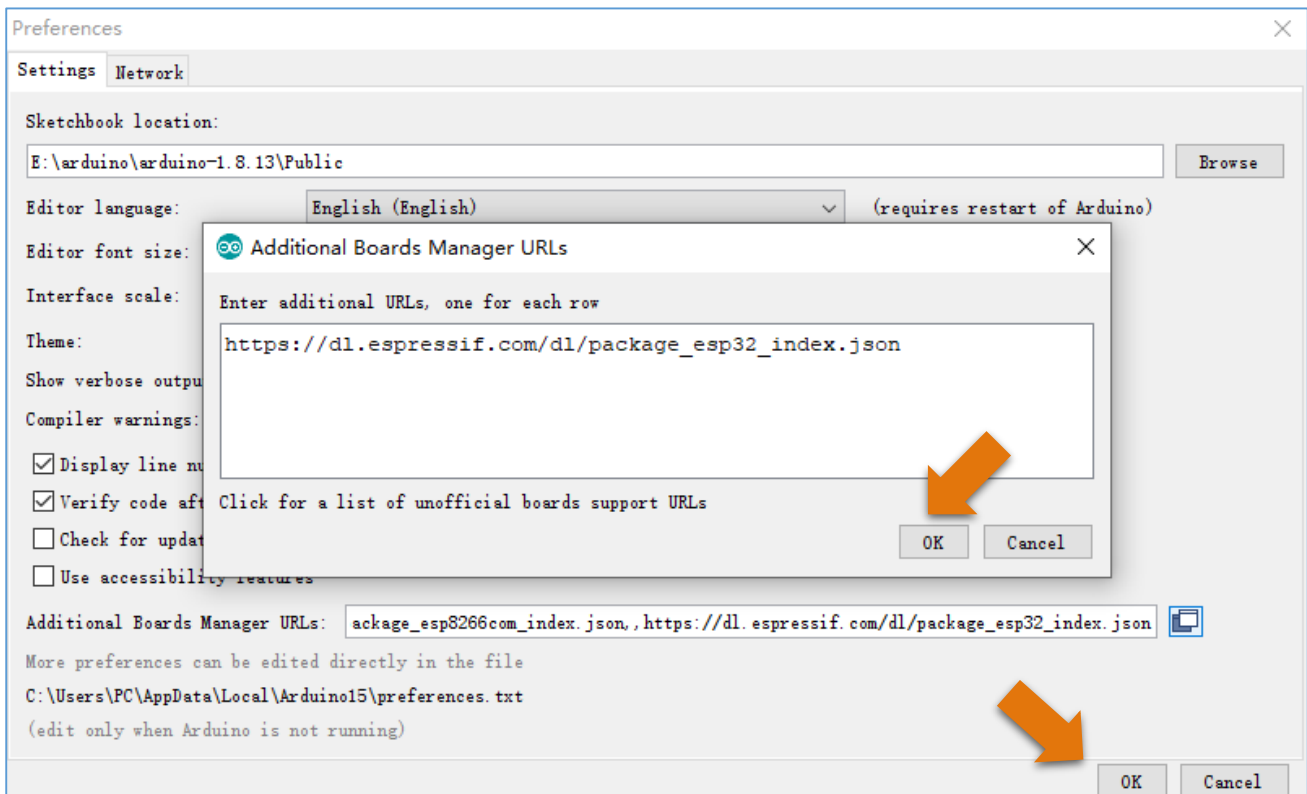
First, open the software platform arduino, and then click File in Menus and select Preferences.



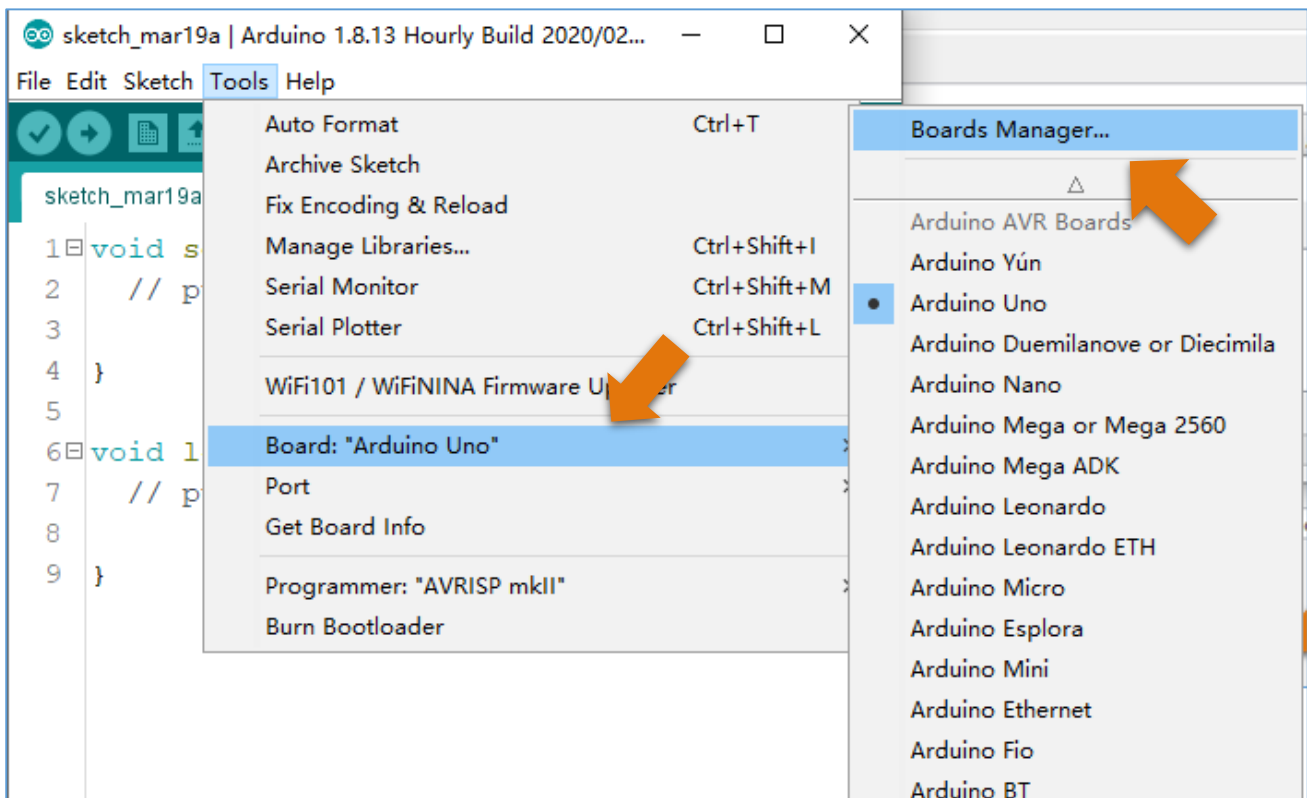
Second, click on the symbol behind "Additional Boards Manager URLs"



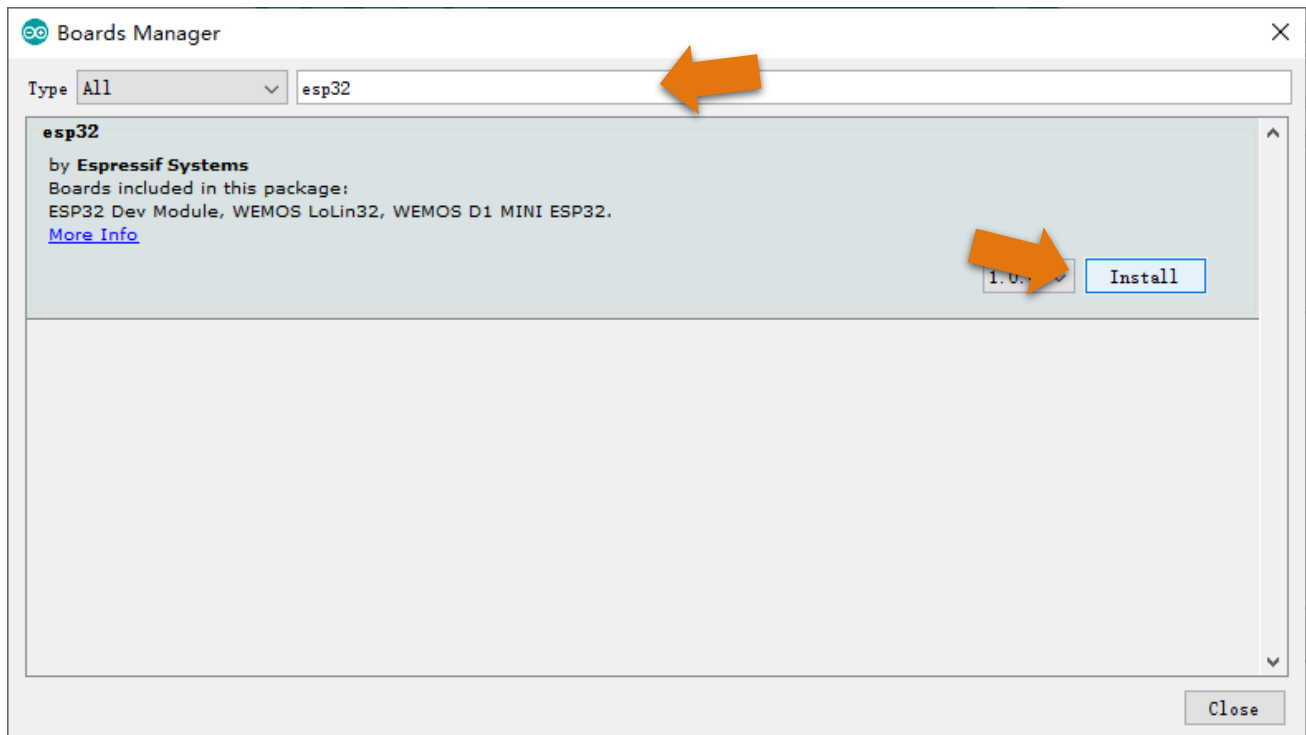
Third, fill in https://dl.espressif.com/dl/package_esp32_index.json in the new window, click OK, and click OK on the Preferences window again.



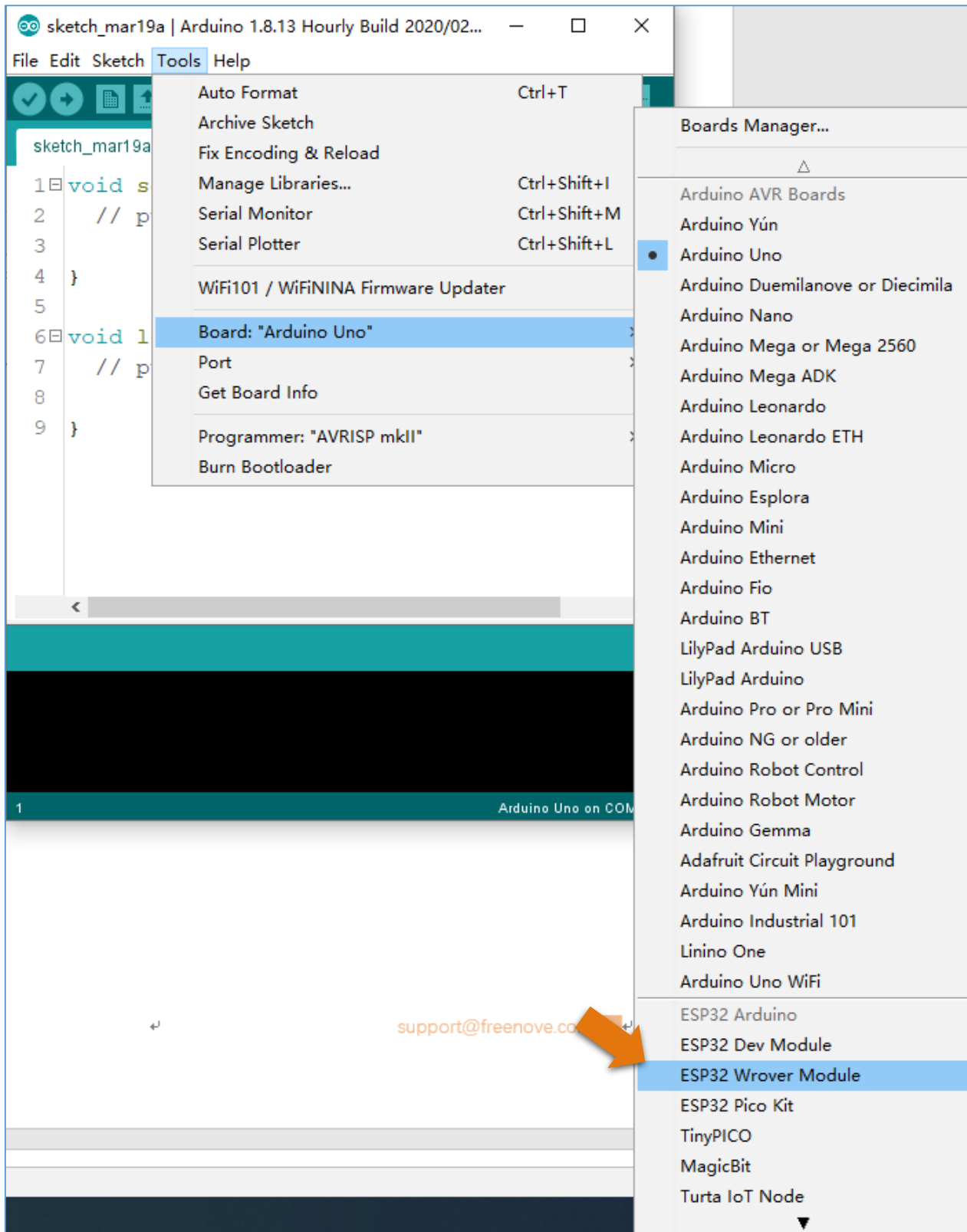
Fourth, click Tools in Menus, select Board:"ArduinoUno", and then select "Boards Manager".



Fifth, input "esp32" in the window below, and press Enter. click "Install" to install.



When finishing installation, click Tools in the Menus again and select Board: "Arduino Uno", and then you can see information of ESP32-WROVER. click "ESP32-WROVER" so that the ESP32 programming development environment is configured.



Notes for GPIO

Strapping Pin

There are five Strapping pins for ESP32: MTDI、GPIO0、GPIO2、MTDO、GPIO5。

With the release of the chip's system reset (power-on reset, RTC watchdog reset, undervoltage reset), the strapping pins sample the level and store it in the latch as "0" or "1" , and keep it until the chip is powered off or turned off.

Each Strapping pin is connecting to internal pull-up/pull-down. Connecting to high-impedance external circuit or without an external connection, a strapping pin's default value of input level will be determined by internal weak pull-up/pull-down. To change the value of the Strapping, users can apply an external pull-down/pull-up resistor, or use the GPIO of the host MCU to control the level of the strapping pin when the ESP32's power on reset is released.

When releasing the reset, the strapping pin has the same function as a normal pin.

The followings are default configurations of these five strapping pins at power-on and their functions under the corresponding configuration.

Voltage of Internal LDO (VDD_SDIO)					
Pin	Default	3.3 V		1.8 V	
MTDI	Pull-down	0		1	
Bootling Mode					
Pin	Default	SPI Boot		Download Boot	
GPIO0	Pull-up	1		0	
GPIO2	Pull-down	Don't-care		0	
Enabling/Disabling Debugging Log Print over U0TXD During Bootling					
Pin	Default	U0TXD Active		U0TXD Silent	
MTDO	Pull-up	1		0	
Timing of SDIO Slave					
Pin	Default	Falling-edge Sampling Falling-edge Output	Falling-edge Sampling Rising-edge Output	Rising-edge Sampling Falling-edge Output	Rising-edge Sampling Rising-edge Output
MTDO	Pull-up	0	0	1	1
GPIO5	Pull-up	0	1	0	1

Note:

- Firmware can configure register bits to change the settings of "Voltage of Internal LDO (VDD_SDIO)" and "Timing of SDIO Slave" after bootling.
- The MTDI is internally pulled high in the module, as the flash and SRAM in ESP32-WROVER only support a power voltage of 1.8 V (output by VDD_SDIO).

If you have any questions about the information of GPIO, you can click [here](#) to go back to ESP32-WROVER to view specific information about GPIO.

If you have any difficulties or questions with this tutorial or toolkit, feel free to ask for our quick and free technical support through support@freenove.com at any time.

or check: https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf

Any concerns? ✉ support@freenove.com

Flash Pin

GPIO6-11 has been used to connect the integrated SPI flash on the module, and is used when GPIO 0 is power on and at high level. Flash is related to the operation of the whole chip, so the external pin GPIO6-11 cannot be used as an experimental pin for external circuits, otherwise it may cause errors in the operation of the program.

GPIO16-17 has been used to connect the integrated PSRAM on the module.

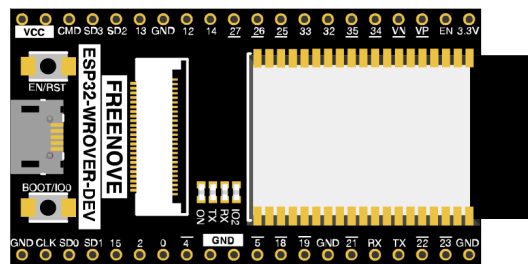
Because of external pull-up, MTDI pin is not suggested to be used as a touch sensor. For details, please refer to Peripheral Interface and Sensor chapter in "ESP32 Data_Sheet".

For more relevant information, please check:

https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf.

Cam Pin

When using the camera of our ESP32-WROVER, please check the pins of it. Pins with underlined numbers are used by the camera function, if you want to use other functions besides it, please avoid using them.



CAM_Pin	GPIO_pin
I2C_SDA	GPIO26
I2C_SCL	GPIO27
CSI_VYSNC	GPIO25
CSI_HREF	GPIO23
CSI_Y9	GPIO35
XCLK	GPIO21
CSI_Y8	GPIO34
CSI_Y7	GPIO39
CSI_PCLK	GPIO22
CSI_Y6	GPIO36
CSI_Y2	GPIO4
CSI_Y5	GPIO19
CSI_Y3	GPIO5
CSI_Y4	GPIO18

If you have any questions about the information of GPIO, you can click [here](#) to go back to ESP32-WROVER to view specific information about GPIO.

or check: https://www.espressif.com/sites/default/files/documentation/esp32-wrover_datasheet_en.pdf.

Chapter 1 LED

This chapter is the Start Point in the journey to build and explore ESP32 electronic projects. We will start with simple “Blink” project.

Project 1.1 Blink

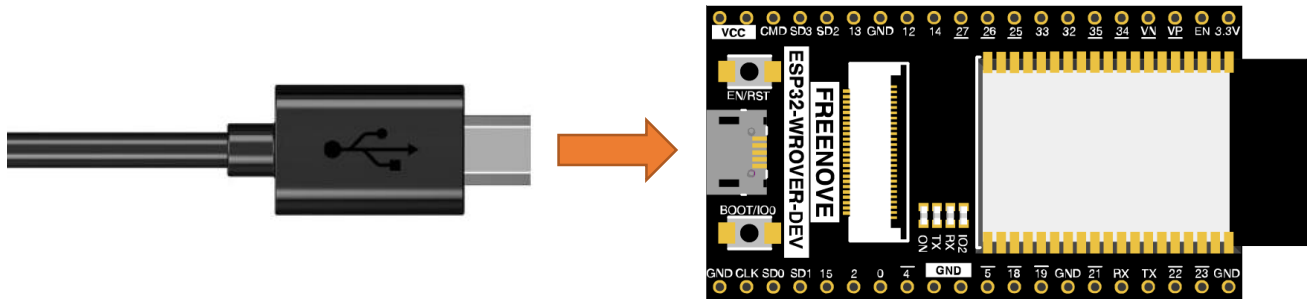
In this project, we will use ESP32 to control blinking a common LED.

Component List

ESP32-WROVER x1	USB cable
-----------------	-----------

Power

ESP32-WROVER needs 5v power supply. In this tutorial, we need connect ESP32-WROVER to computer via USB cable to power it and program it. We can also use other 5v power source to power it.



In the following projects, we only use USB cable to power ESP32-WROVER by default.

In the whole tutorial, we don't use T extension to power ESP32-WROVER. So 5V and 3.3V (including EXT 3.3V) on the extension board are provided by ESP32-WROVER.

We can also use DC jack of extension board to power ESP32-WROVER. In this way, 5v and EXT 3.3v on extension board are provided by external power resource.

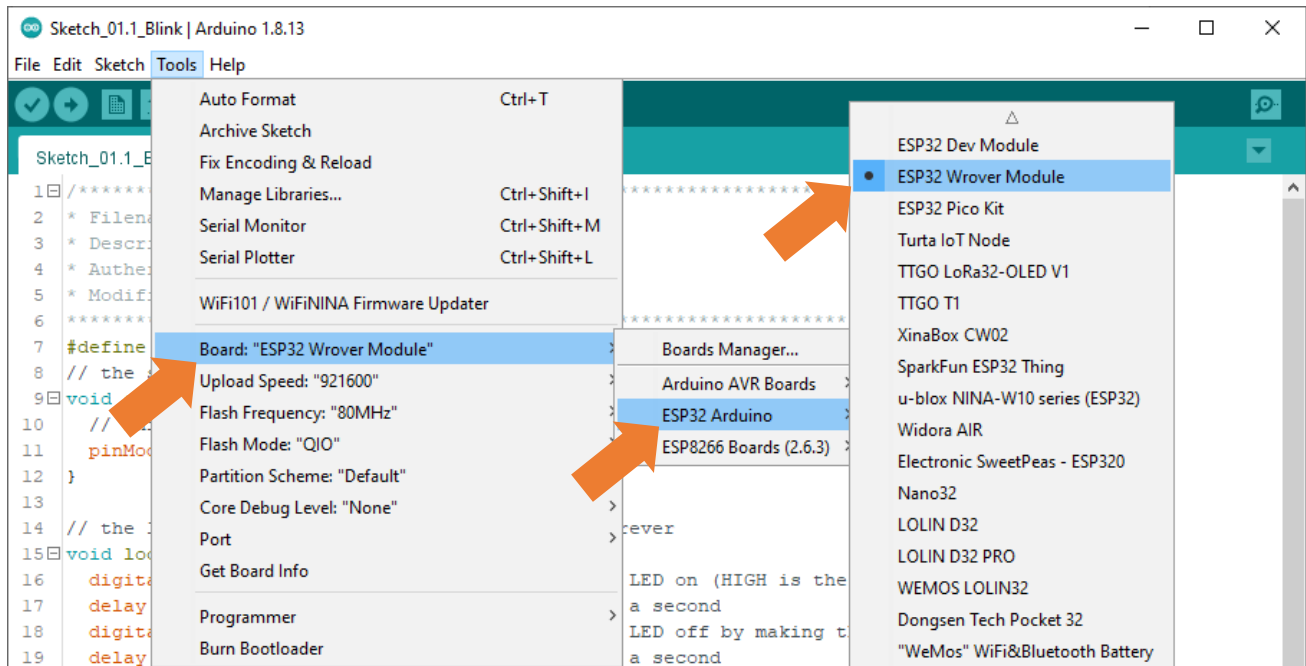
Sketch

According to the circuit, when the GPIO2 of ESP32-WROVER output level is high, the LED turns ON. Conversely, when the GPIO2 ESP32-WROVER output level is low, the LED turns OFF. Therefore, we can let GPIO2 circularly output high and low level to make the LED blink.

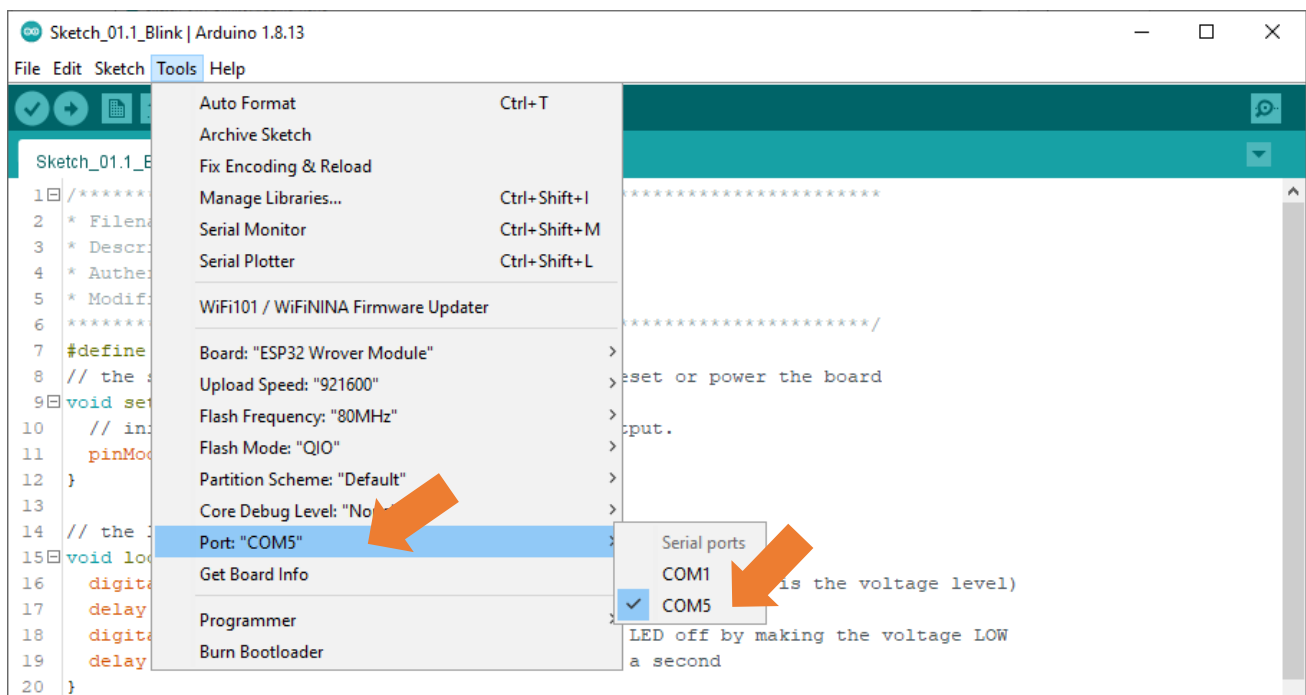
Upload the following Sketch:

Freenove_ESP32_WROVER_Starter_Kit_for_ESP32\Sketches\Sketch_01.1_Blink.

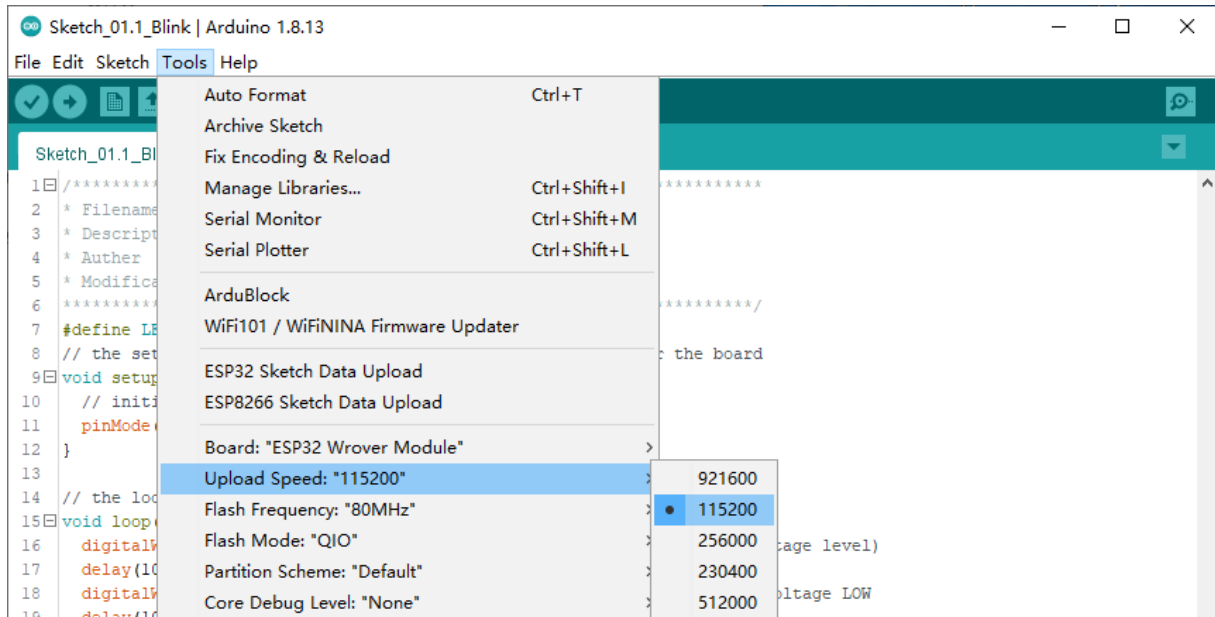
Before uploading the code, click "Tools", "Board" and select "ESP32 Wrover Module".



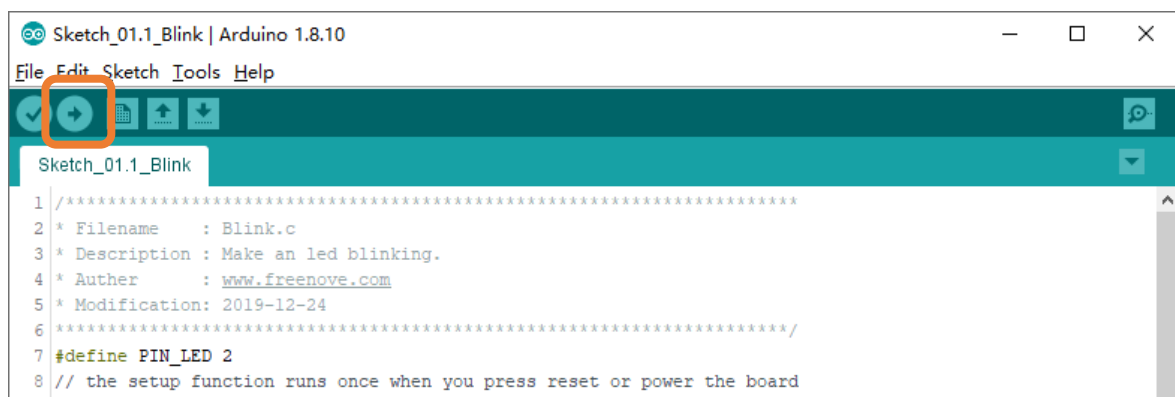
Select the serial port.



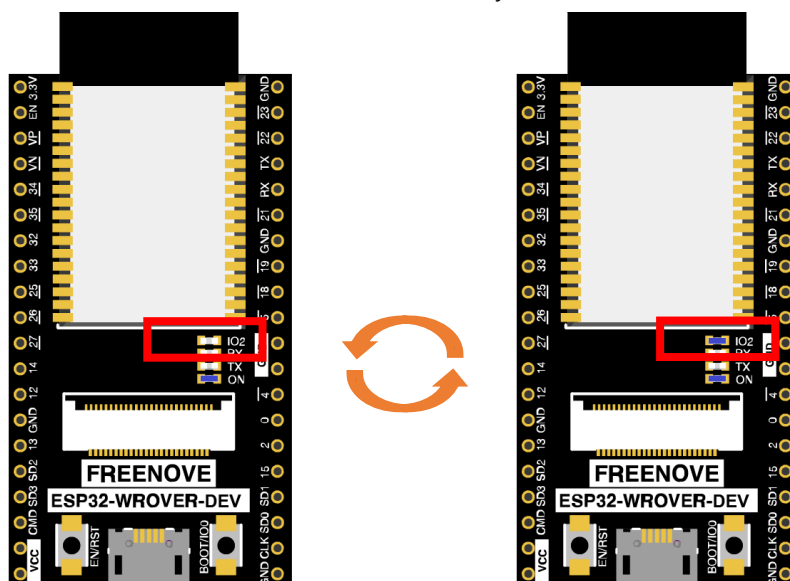
Note: For macOS users, if the uploading fails, please set the baud rate to 115200 before clicking “Upload Using Programmer”.



Sketch_01.1_Blink



Click “Upload”, Download the code to ESP32-WROVER and your LED in the circuit starts Blink.



If you have any concerns, please contact us via: support@freenove.com

Any concerns? [✉ support@freenove.com](mailto:support@freenove.com)

The following is the program code:

```

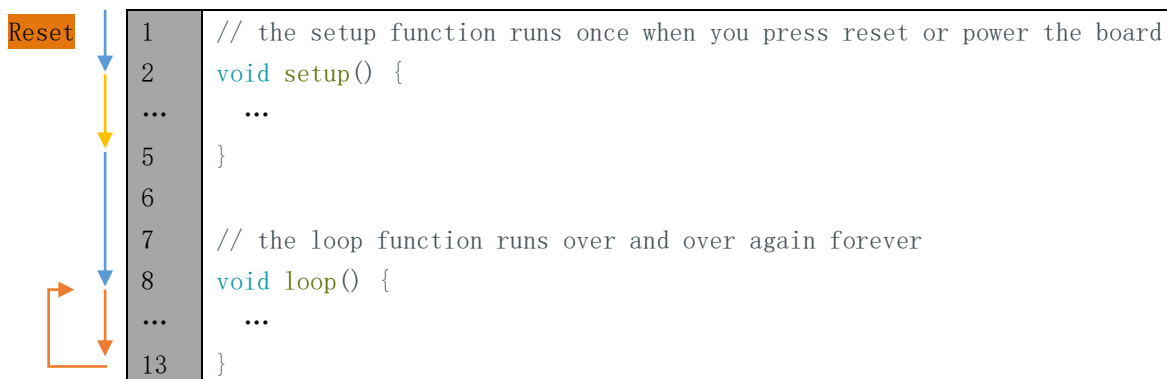
1  #define PIN_LED 2
2  // the setup function runs once when you press reset or power the board
3  void setup() {
4      // initialize digital pin LED_BUILTIN as an output.
5      pinMode(PIN_LED, OUTPUT);
6  }
7
8  // the loop function runs over and over again forever
9  void loop() {
10     digitalWrite(PIN_LED, HIGH); // turn the LED on (HIGH is the voltage level)
11     delay(1000);                 // wait for a second
12     digitalWrite(PIN_LED, LOW);  // turn the LED off by making the voltage LOW
13     delay(1000);                 // wait for a second
14 }

```

The Arduino IDE code usually contains two basic functions: void setup() and void loop().

After the board is reset, the setup() function will be executed firstly, and then the loop() function.

setup() function is generally used to write code to initialize the hardware. And loop() function is used to write code to achieve certain functions. loop() function is executed repeatedly. When the execution reaches the end of loop(), it will jump to the beginning of loop() to run again.



Reset

Reset operation will lead the code to be executed from the beginning. Switching on the power, finishing uploading the code and pressing the reset button will trigger reset operation.

In the circuit, ESP32-WROVER's GPIO2 is connected to the LED, so the LED pin is defined as 2.

```
1  #define PIN_LED 2
```

This means that after this line of code, all PIN_LED will be treated as 2.

In the setup () function, first, we set the PIN_LED as output mode, which can make the port output high level or low level.

```

4  // initialize digital pin PIN_LED as an output.
5  pinMode(PIN_LED, OUTPUT);

```

Then, in the loop () function, set the PIN_LED to output high level to make LED light up.

```
10 digitalWrite(PIN_LED, HIGH); // turn the LED on (HIGH is the voltage level)
```

Any concerns? [✉ support@freenove.com](mailto:support@freenove.com)

Wait for 1000ms, that is 1s. Delay () function is used to make control board wait for a moment before executing the next statement. The parameter indicates the number of milliseconds to wait for.

```
11    delay(1000);           // wait for a second
```

Then set the PIN_LED to output low level, and LED light off. One second later, the execution of loop () function will be completed.

```
12    digitalWrite(PIN_LED, LOW); // turn the LED off by making the voltage LOW
13    delay(1000);           // wait for a second
```

The loop() function is constantly being executed, so LED will keep blinking.

Reference

```
void pinMode(int pin, int mode);
```

Configures the specified pin to behave either as an input or an output.

Parameters

pin: the pin number to set the mode of.

mode: INPUT, OUTPUT, INPUT_PULLDOWN, or INPUT_PULLUP.

```
void digitalWrite (int pin, int value);
```

Writes the value HIGH or LOW (1 or 0) to the given pin which must have been previously set as an output.

For more related functions, please refer to <https://www.arduino.cc/reference/en/>

Chapter 2 Camera Web Server

In this section, we'll use ESP32's video function as an example to study.

Project 2.1 Camera Web Server

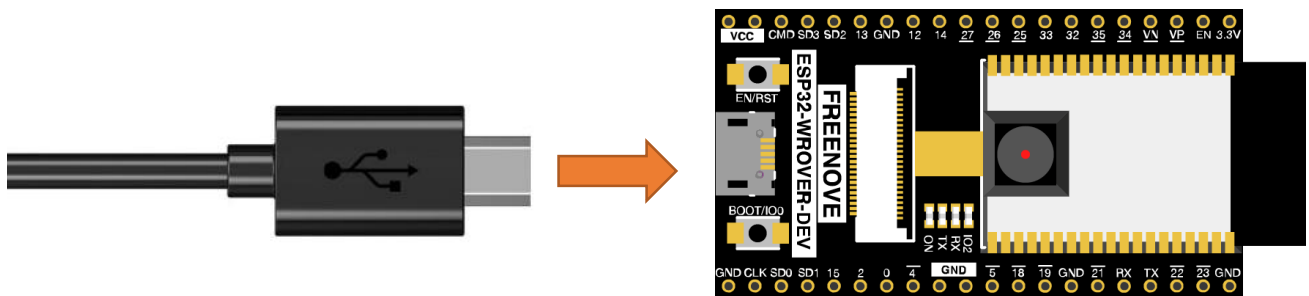
Connect ESP32 using USB and check its IP address through serial monitor. Use web page to access IP address to obtain video and image data.

Component List

Micro USB Wire x1	ESP32-WROVER x1
-------------------	-----------------

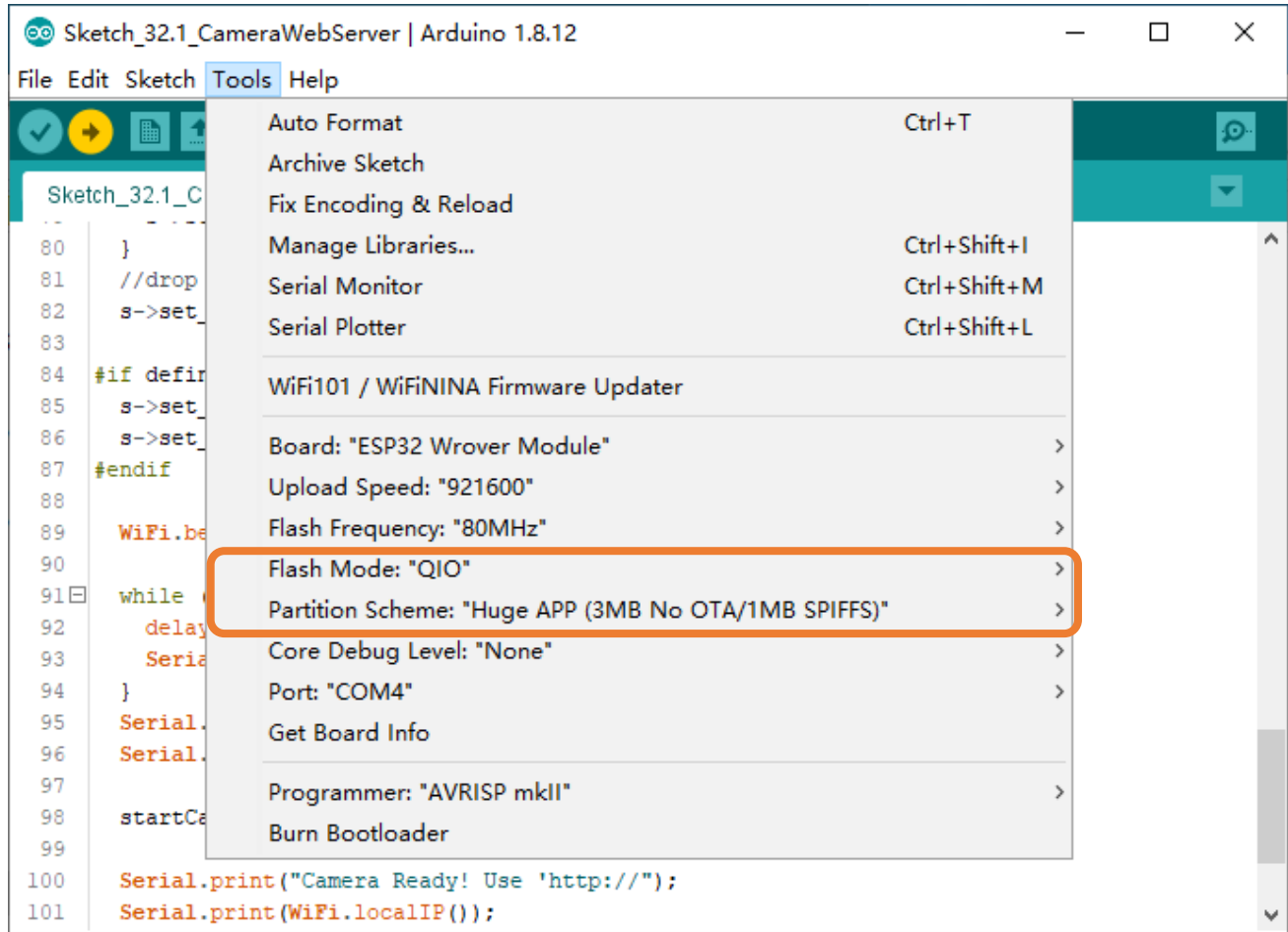
Circuit

Connect Freenove ESP32 to the computer using USB cable.



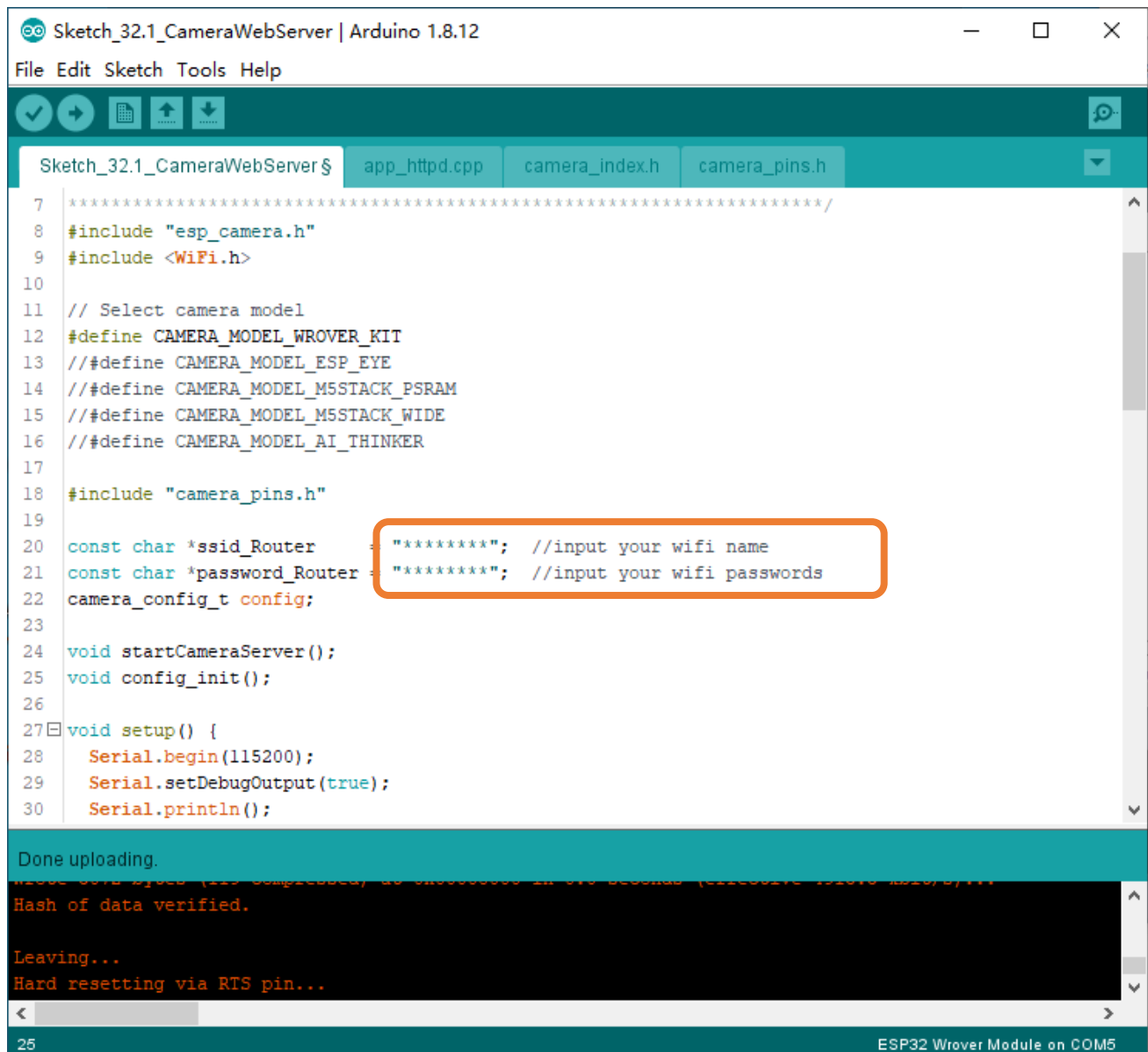
Sketch

Click Tools in the menu bar, select Flash Mode: "QIO", and select Partition Scheme: "Huge APP (3MB No OTA/1MB SPIFFS)", as shown in the figure below.



The reason for this configuration is that the Sketch occupies about 2MB of program storage, while in the previous default mode, it was about 1.2M. Therefore, we need to choose a more appropriate partitioning scheme for ESP32.

Sketch_02.1_As_CameraWebServer



```
Sketch_32.1_CameraWebServer | Arduino 1.8.12
File Edit Sketch Tools Help

Sketch_32.1_CameraWebServer$ app_httpd.cpp camera_index.h camera_pins.h

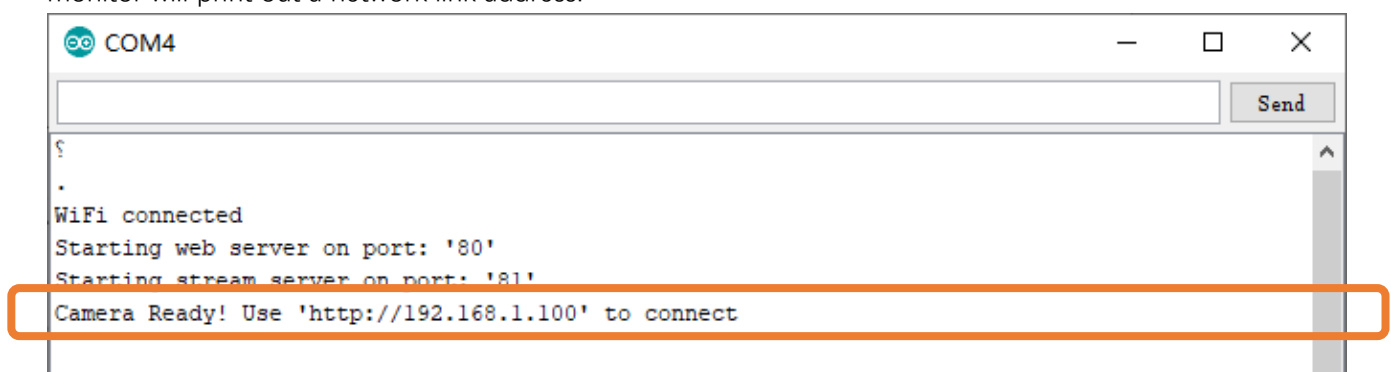
7  *****/
8  #include "esp_camera.h"
9  #include <WiFi.h>
10
11 // Select camera model
12 #define CAMERA_MODEL_WROVER_KIT
13 //#define CAMERA_MODEL_ESP_EYE
14 //#define CAMERA_MODEL_M5STACK_PSRAM
15 //#define CAMERA_MODEL_M5STACK_WIDE
16 //#define CAMERA_MODEL_AI_THINKER
17
18 #include "camera_pins.h"
19
20 const char *ssid_Router = "*****"; //input your wifi name
21 const char *password_Router = "*****"; //input your wifi passwords
22 camera_config_t config;
23
24 void startCameraServer();
25 void config_init();
26
27 void setup() {
28     Serial.begin(115200);
29     Serial.setDebugOutput(true);
30     Serial.println();
31 }

Done uploading.
Hash of data verified.
Leaving...
Hard resetting via RTS pin...

25 ESP32 Wrover Module on COM5
```

Before running the program, please modify your router's name and password in the box shown in the illustration above to make sure that your Sketch can compile and work successfully.

Compile and upload codes to ESP32, open the serial monitor and set the baud rate to 115200, and the serial monitor will print out a network link address.



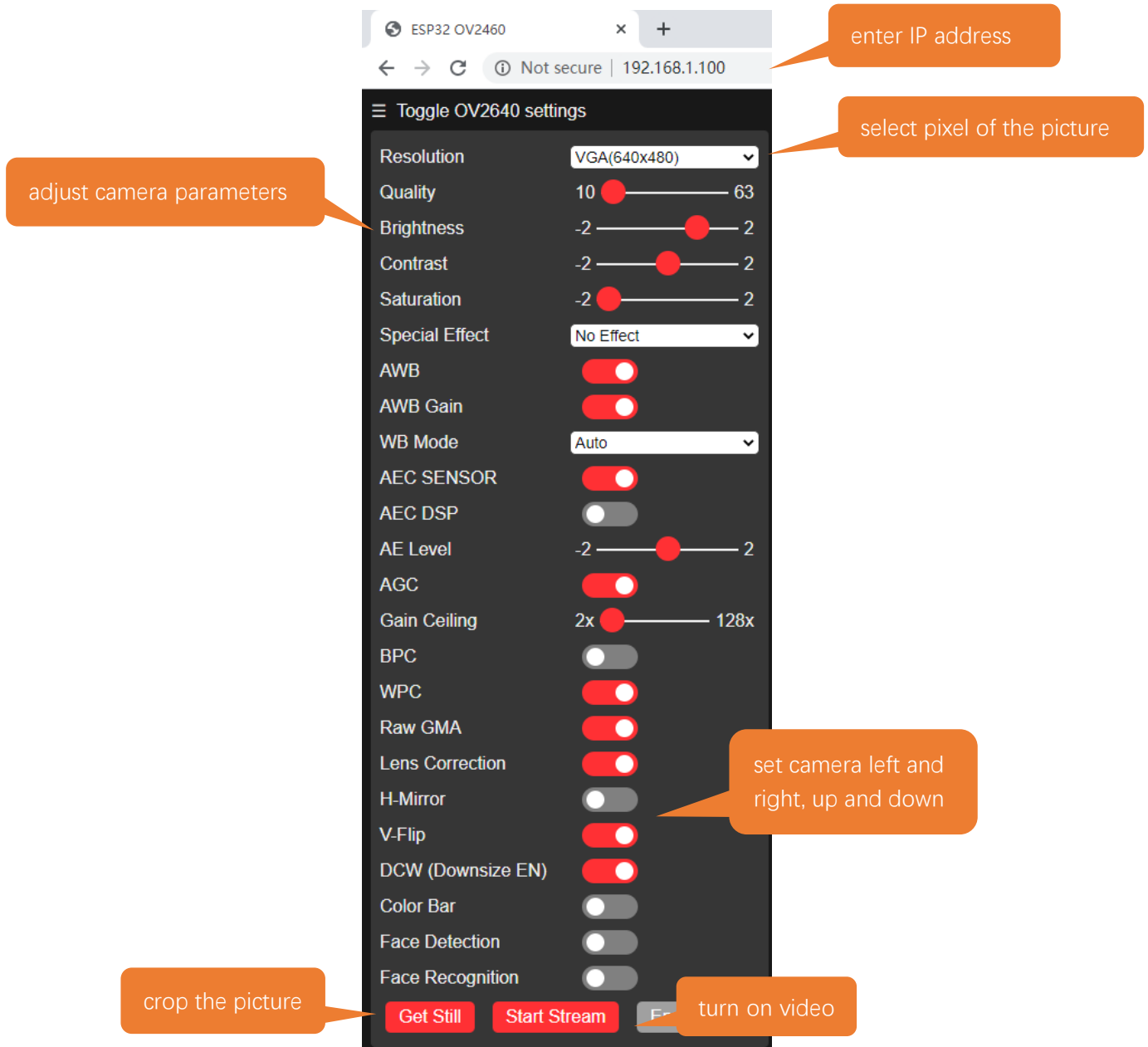
```
COM4

?
.
WiFi connected
Starting web server on port: '80'
Starting stream server on port: '81'
Camera Ready! Use 'http://192.168.1.100' to connect
```

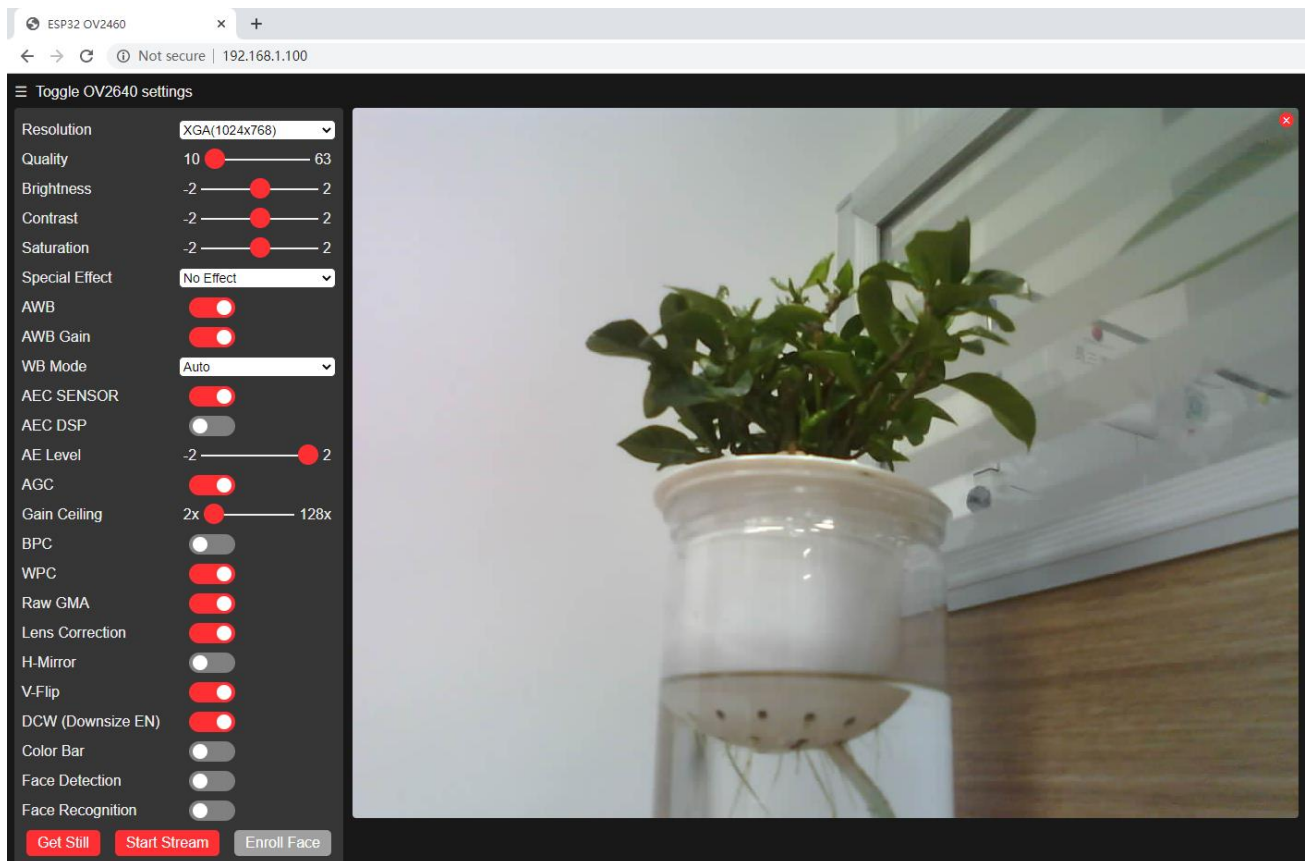
Any concerns? [✉ support@freenove.com](mailto:support@freenove.com)

If your ESP32 has been in the process of connecting to router, but the information above has not been printed out, please re-check whether the router name and password have been entered correctly and press the reset key on ESP32-WROVER to wait for a successful connection prompt.

Open a web browser, enter the IP address printed by the serial monitor in the address bar, and access it. Taking the Google browser as an example, here's what the browser prints out after successful access to ESP32's IP.



Click on Start Stream. The effect is shown in the image below.



The following is the main program code. You need include other code files in the same folder when write your own code.

```

1  #include "esp_camera.h"
2  #include <WiFi.h>
3
4  // Select camera model
5  #define CAMERA_MODEL_WROVER_KIT
6  //#define CAMERA_MODEL_ESP_EYE
7  //#define CAMERA_MODEL_M5STACK_PSRAM
8  //#define CAMERA_MODEL_M5STACK_WIDE
9  //#define CAMERA_MODEL_AI_THINKER
10
11 #include "camera_pins.h"
12
13 const char *ssid_Router    = "*****"; //input your wifi name
14 const char *password_Router = "*****"; //input your wifi passwords
15 camera_config_t config;
16
17 void startCameraServer();
18 void config_init();
19
20 void setup() {

```

```
21 Serial.begin(115200);
22 Serial.setDebugOutput(true);
23 Serial.println();
24
25 config_init();
26 config.frame_size = FRAMESIZE_VGA;
27 config.jpeg_quality = 10;
28
29 // camera init
30 esp_err_t err = esp_camera_init(&config);
31 if (err != ESP_OK) {
32     Serial.printf("Camera init failed with error 0x%x", err);
33     return;
34 }
35
36 sensor_t * s = esp_camera_sensor_get();
37 s->set_vflip(s, 1);           //flip it back
38 s->set_brightness(s, 1);      //up the blightness just a bit
39 s->set_saturation(s, -1);     //lower the saturation
40
41 WiFi.begin(ssid_Router, password_Router);
42 while (WiFi.status() != WL_CONNECTED) {
43     delay(500);
44     Serial.print(".");
45 }
46 Serial.println("");
47 Serial.println("WiFi connected");
48
49 startCameraServer();
50
51 Serial.print("Camera Ready! Use 'http://");
52 Serial.print(WiFi.localIP());
53 Serial.println("' to connect");
54 }
55
56 void loop() {
57     ;
58 }
59
60 void config_init() {
61     config.ledc_channel = LEDC_CHANNEL_0;
62     config.ledc_timer = LEDC_TIMER_0;
63     config.pin_d0 = Y2_GPIO_NUM;
64     config.pin_d1 = Y3_GPIO_NUM;
```

```

65  config.pin_d2 = Y4_GPIO_NUM;
66  config.pin_d3 = Y5_GPIO_NUM;
67  config.pin_d4 = Y6_GPIO_NUM;
68  config.pin_d5 = Y7_GPIO_NUM;
69  config.pin_d6 = Y8_GPIO_NUM;
70  config.pin_d7 = Y9_GPIO_NUM;
71  config.pin_xclk = XCLK_GPIO_NUM;
72  config.pin_pclk = PCLK_GPIO_NUM;
73  config.pin_vsync = VSYNC_GPIO_NUM;
74  config.pin_href = HREF_GPIO_NUM;
75  config.pin_sscb_sda = SIOD_GPIO_NUM;
76  config.pin_sscb_scl = SIOC_GPIO_NUM;
77  config.pin_pwdn = PWDN_GPIO_NUM;
78  config.pin_reset = RESET_GPIO_NUM;
79  config.xclk_freq_hz = 20000000;
80  config.pixel_format = PIXFORMAT_JPEG;
81  config.fb_count = 1;
82  }

```

Add procedure files and API interface files related to ESP32 camera.

```

1  #include "esp_camera.h"
2  #include <WiFi.h>
3
4  // Select camera model
5  #define CAMERA_MODEL_WROVER_KIT
6  //#define CAMERA_MODEL_ESP_EYE
7  //#define CAMERA_MODEL_M5STACK_PSRAM
8  //#define CAMERA_MODEL_M5STACK_WIDE
9  //#define CAMERA_MODEL_AI_THINKER
10
11 #include "camera_pins.h"

```

Enter the name and password of the router

```

13  const char *ssid_Router    = "*****"; //input your wifi name
14  const char *password_Router = "*****"; //input your wifi passwords

```

Initialize serial port, set baud rate to 115200; open the debug and output function of the serial.

```

21  Serial.begin(115200);
22  Serial.setDebugOutput(true);
23  Serial.println();

```

Configure parameters including interface pins of the camera. Note: It is generally not recommended to change them.

```

60  void config_init() {
61      config.ledc_channel = LEDC_CHANNEL_0;
62      config.ledc_timer = LEDC_TIMER_0;
63      config.pin_d0 = Y2_GPIO_NUM;
64      config.pin_d1 = Y3_GPIO_NUM;

```

```

65 config.pin_d2 = Y4_GPIO_NUM;
66 config.pin_d3 = Y5_GPIO_NUM;
67 config.pin_d4 = Y6_GPIO_NUM;
68 config.pin_d5 = Y7_GPIO_NUM;
69 config.pin_d6 = Y8_GPIO_NUM;
70 config.pin_d7 = Y9_GPIO_NUM;
71 config.pin_xclk = XCLK_GPIO_NUM;
72 config.pin_pclk = PCLK_GPIO_NUM;
73 config.pin_vsync = VSYNC_GPIO_NUM;
74 config.pin_href = HREF_GPIO_NUM;
75 config.pin_sscb_sda = SIOD_GPIO_NUM;
76 config.pin_sscb_scl = SIOC_GPIO_NUM;
77 config.pin_pwdn = PWDN_GPIO_NUM;
78 config.pin_reset = RESET_GPIO_NUM;
79 config.xclk_freq_hz = 20000000;
80 config.pixel_format = PIXFORMAT_JPEG;
81 config.fb_count = 1;
82 }

```

ESP32 connects to the router and prints a successful connection prompt. If it has not been successfully connected, press the reset key on the ESP32-WROVER.

```

41 WiFi.begin(ssid_Router, password_Router);
42 while (WiFi.status() != WL_CONNECTED) {
43     delay(500);
44     Serial.print(".");
45 }
46 Serial.println("");
47 Serial.println("WiFi connected");

```

Open the video streams server function of the camera and print its IP address via serial port.

```

49 startCameraServer();
50
51 Serial.print("Camera Ready! Use 'http://");
52 Serial.print(WiFi.localIP());
53 Serial.println("' to connect");

```

Configure the display image information of the camera.

The `set_vflip()` function sets whether the image is flipped 180°, with 0 for no flip and 1 for flip 180°.

The `set_brightness()` function sets the brightness of the image, with values ranging from -2 to 2.

The `set_saturation()` function sets the color saturation of the image, with values ranging from -2 to 2.

```

36 sensor_t * s = esp_camera_sensor_get();
37 s->set_vflip(s, 1); //flip it back
38 s->set_brightness(s, 1); //up the blightness just a bit
39 s->set_saturation(s, -1); //lower the saturation

```

Modify the resolution and sharpness of the images captured by the camera. The sharpness ranges from 10 to 63, and the smaller the number, the sharper the picture. The larger the number, the blurrier the picture. Please refer to the table below.

26	<code>config.frame_size = FRAMESIZE_VGA;</code>
27	<code>config.jpeg_quality = 10;</code>

Reference

Image resolution	Sharpness	Image resolution	Sharpness
FRAMESIZE_QQVGA	160x120	FRAMESIZE_VGA	640x480
FRAMESIZE_QQVGA2	128x160	FRAMESIZE_SVGA	800x600
FRAMESIZE_QCIF	176x144	FRAMESIZE_XGA	1024x768
FRAMESIZE_HQVGA	240x176	FRAMESIZE_SXGA	1280x1024
FRAMESIZE_QVGA	320x240	FRAMESIZE_UXGA	1600x1200
FRAMESIZE_CIF	400x296	FRAMESIZE_QXGA	2048x1536

What's next?

Thanks for your reading.

This tutorial is all over here. If you find any mistakes, omissions or you have other ideas and questions about contents of this tutorial or the kit and etc., please feel free to contact us: support@freenove.com
We will check and correct it as soon as possible.

If you are interesting in processing, you can learn the Processing.pdf in the unzipped folder.

If you want to learn more about Arduino, Raspberry Pi, smart cars, robots and other interesting products in science and technology, please continue to focus on our website. We will continue to launch cost-effective, innovative and exciting products.

<http://www.freenove.com/>

Thank you again for choosing Freenove products.