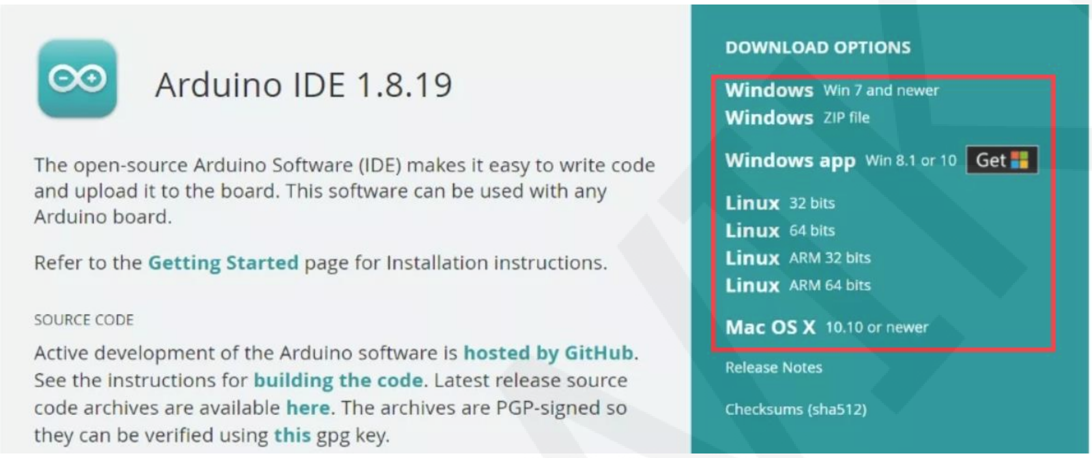
1. Installing the Arduino IDE

Download the installation package from the Arduino official website.

Download address:https://www.arduino.cc/en/software

Download the corresponding installation package according to your PC system, as shown in the following figure (the version in the picture may not be the latest version, and the download interface may not be the latest):



Arduino IDE 1.8.19

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.

Refer to the **Getting Started** page for Installation instructions.

SOURCE CODE

Active development of the Arduino software is **hosted by GitHub**. See the instructions for **building the code**. Latest release source code archives are available **here**. The archives are PGP-signed so they can be verified using **this** gpg key.

DOWNLOAD OPTIONS

Windows Win 7 and newer
Windows ZIP file
Windows app Win 8.1 or 10 Get
Linux 32 bits
Linux 64 bits
Linux ARM 32 bits
Linux ARM 64 bits
Mac OS X 10.10 or newer

Release Notes

Checksums (sha512)

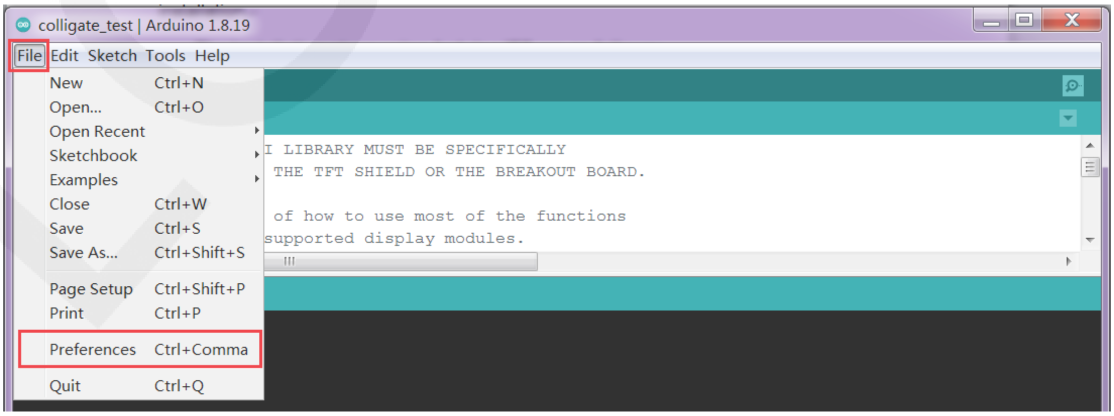
After the download is completed, unzip and click Install.

2. Install Arduino-ESP32 support library

There are two installation methods: using the Arduino IDE for installation and manual installation.

The installation steps using Arduino IED are as follows:

A. Open the Arduino IED software, click on the **File** ->**Preferences** button, as shown in the following figure:

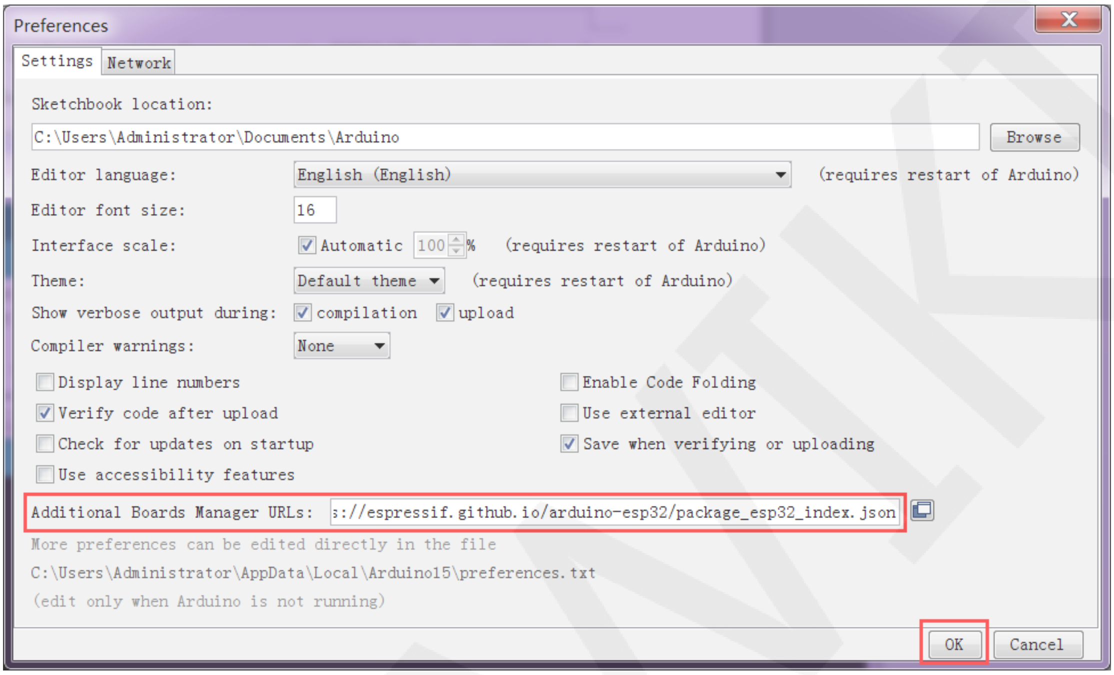


```
I LIBRARY MUST BE SPECIFICALLY
 THE TFT SHIELD OR THE BREAKOUT BOARD.

 of how to use most of the functions
supported display modules.
```

B. In the pop-up preferences interface, enter the ESP32 URL in the **Additional Boards Manager URLs**:

https://espressif.github.io/arduino-esp32/package_esp32_index.json

Then click the "**OK**" button to save and exit, as shown in the following figure:



Preferences

Settings Network

Sketchbook location:

C:\Users\Administrator\Documents\Arduino Browse

Editor language: English (English) (requires restart of Arduino)

Editor font size: 16

Interface scale: ☑ Automatic 100 % (requires restart of Arduino)

Theme: Default theme (requires restart of Arduino)

Show verbose output during: ☑ compilation ☑ upload

Compiler warnings: None

□ Display line numbers □ Enable Code Folding

☑ Verify code after upload □ Use external editor

□ Check for updates on startup ☑ Save when verifying or uploading

□ Use accessibility features

Additional Boards Manager URLs: s://espressif.github.io/arduino-esp32/package_esp32_index.json

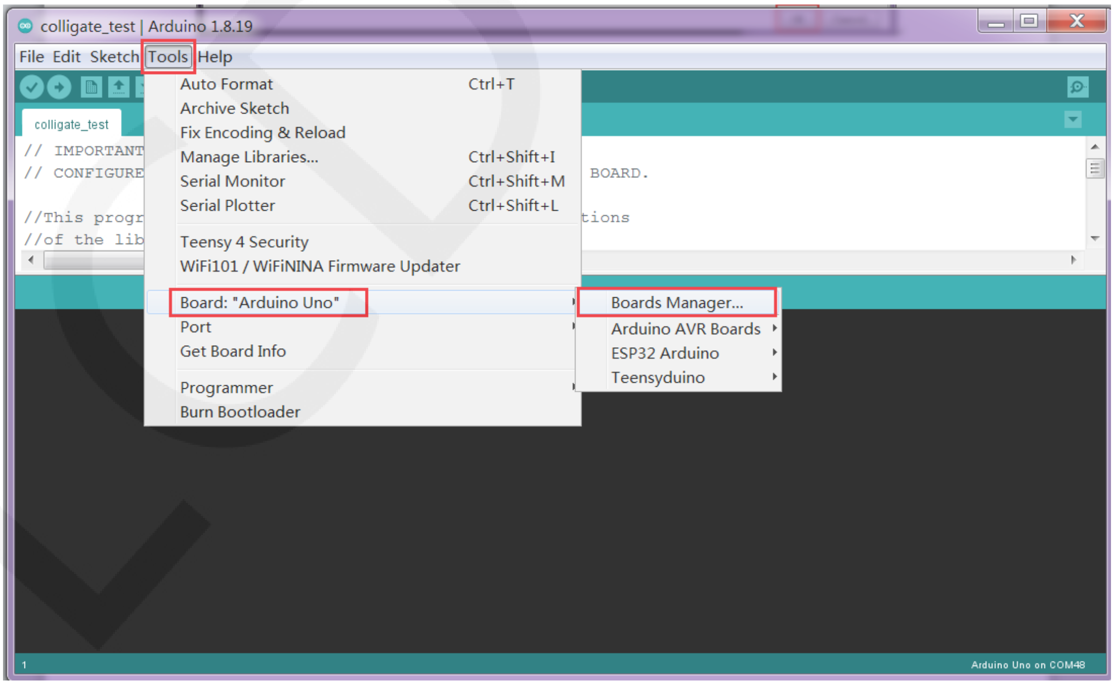
More preferences can be edited directly in the file

C:\Users\Administrator\AppData\Local\Arduino15\preferences.txt

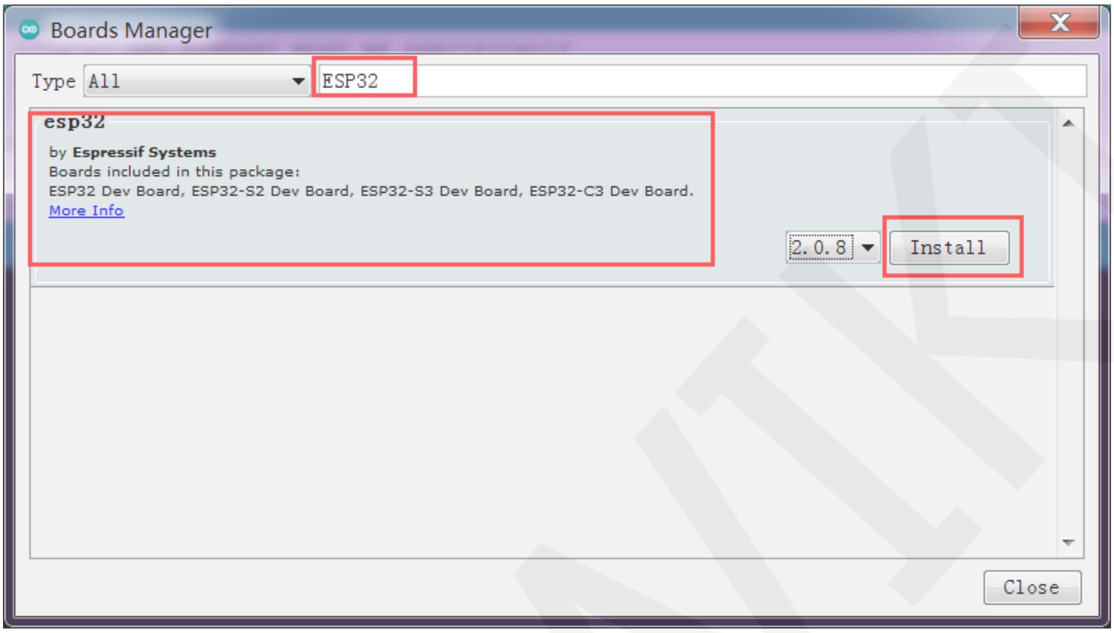
(edit only when Arduino is not running)

OK Cancel

C. Click the **Tools** button, then select the **Board**->**Boards Manager** option, as shown in the following figure:

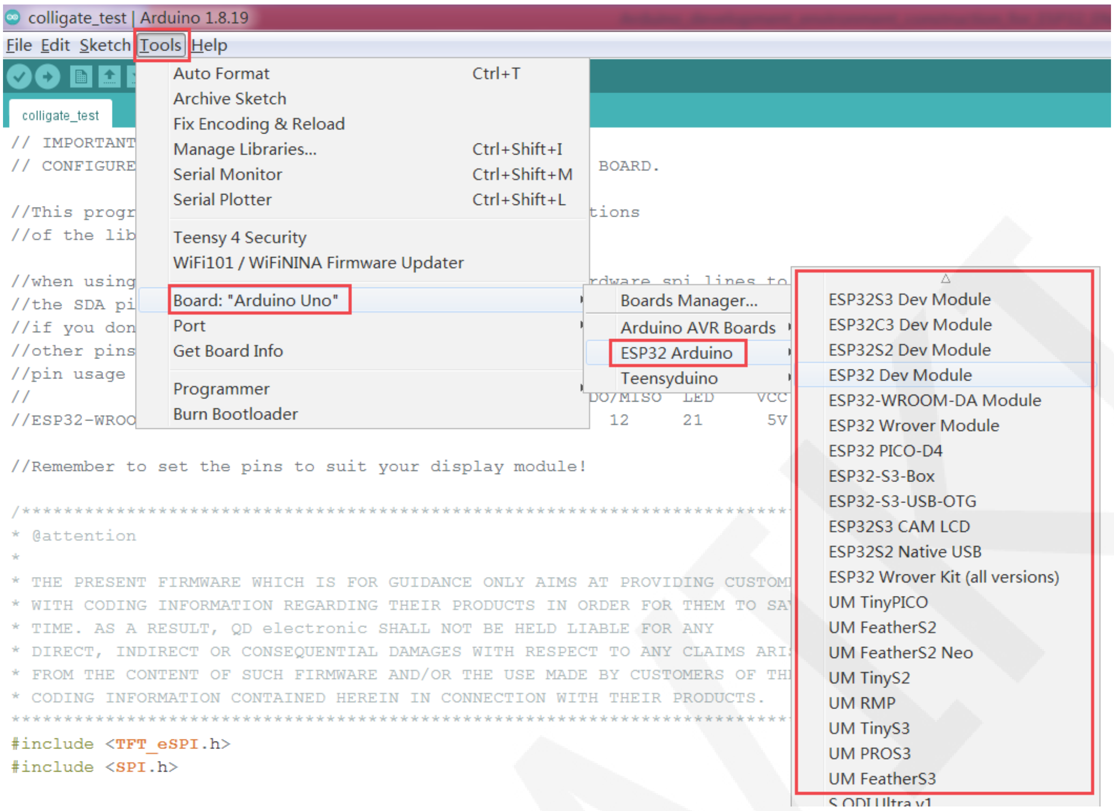


D. Click the **Boards Manager** button to enter the Boards Manager interface. After the interface is loaded, enter "**ESP32**" in the search bar to search for the Arduino-ESP32 support library. After the search is completed, click the "**Install**" button to install, as shown in the following figure:



esp32
by **Espressif Systems**
Boards included in this package:
ESP32 Dev Board, ESP32-S2 Dev Board, ESP32-S3 Dev Board, ESP32-C3 Dev Board.
More Info

E. The installation time is relatively long. If there is an error during the installation process, try it multiple times. After the installation is completed, click the "**Tools**" button and select the **Board** option. You will see the **ESP32 Arduino** option. After selecting this option, various ESP32 development board models will appear, as shown in the following figure:



```
// IMPORTANT                                                  BOARD.
// CONFIGURE

//This progr                                                  tions
//of the lib

//when using                                                  rdware spi lines to
//the SDA pi
//if you don
//other pins
//pin usage
//                                                            DO/MISO   LED    VCC
//ESP32-WROO                                                  12        21     5V

//Remember to set the pins to suit your display module!

/**********************************************************************************
* @attention
*
* THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOM
* WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SA
* TIME. AS A RESULT, QD electronic SHALL NOT BE HELD LIABLE FOR ANY
* DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARI
* FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF TH
* CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.
**********************************************************************************
#include <TFT_eSPI.h>
#include <SPI.h>
```

If Arduino IEDs cannot be successfully installed (possibly due to poor network environment, resulting in continuous download errors), then manual installation is necessary.

The steps for manual installation are as follows:

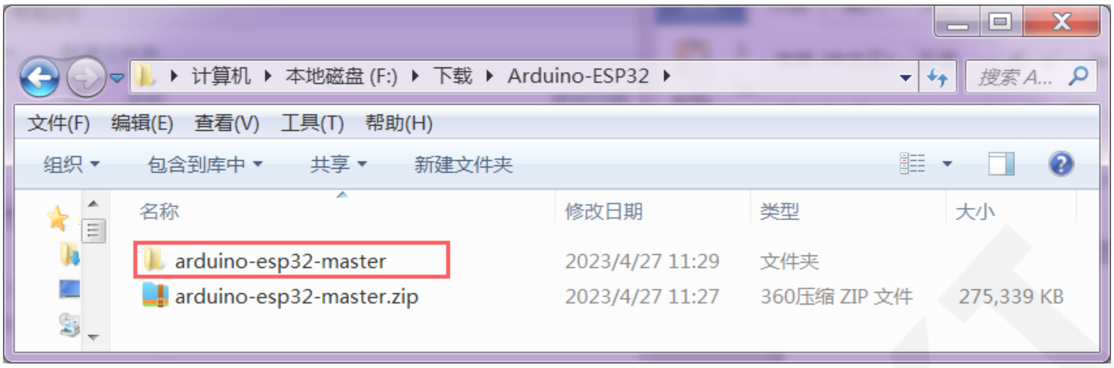
A. Download the Arduino-ESP32 support library at the following address:

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

If the above address cannot be downloaded, it is recommended to use the following download address:

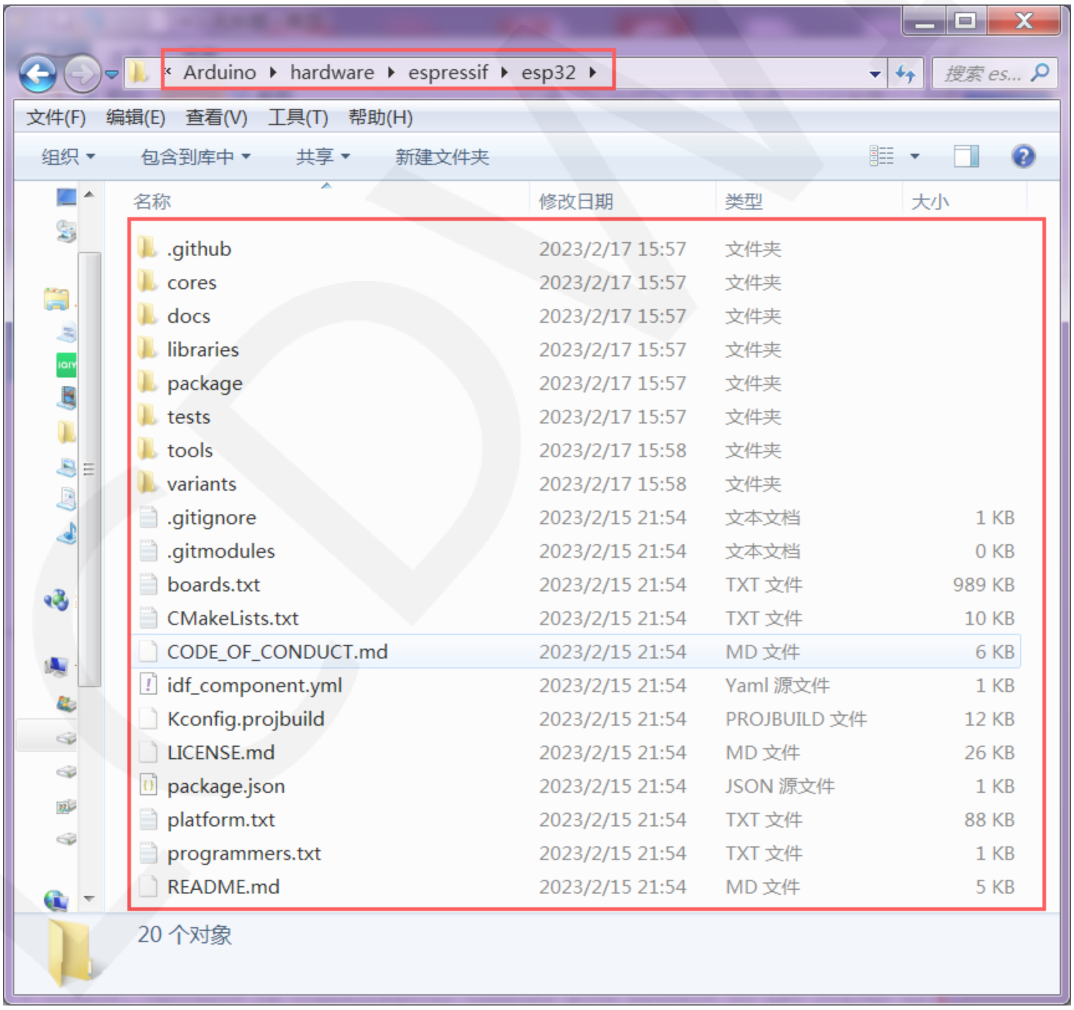
https://gitcode.net/mirrors/espressif/arduino-esp32

B. After successfully downloading the support library, unzip the file as shown in the following figure:


计算机 ▸ 本地磁盘 (F:) ▸ 下载 ▸ Arduino-ESP32 ▸
搜索 A...
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 ▾ 包含到库中 ▾ 共享 ▾ 新建文件夹

名称	修改日期	类型	大小
arduino-esp32-master	2023/4/27 11:29	文件夹	
arduino-esp32-master.zip	2023/4/27 11:27	360压缩 ZIP 文件	275,339 KB

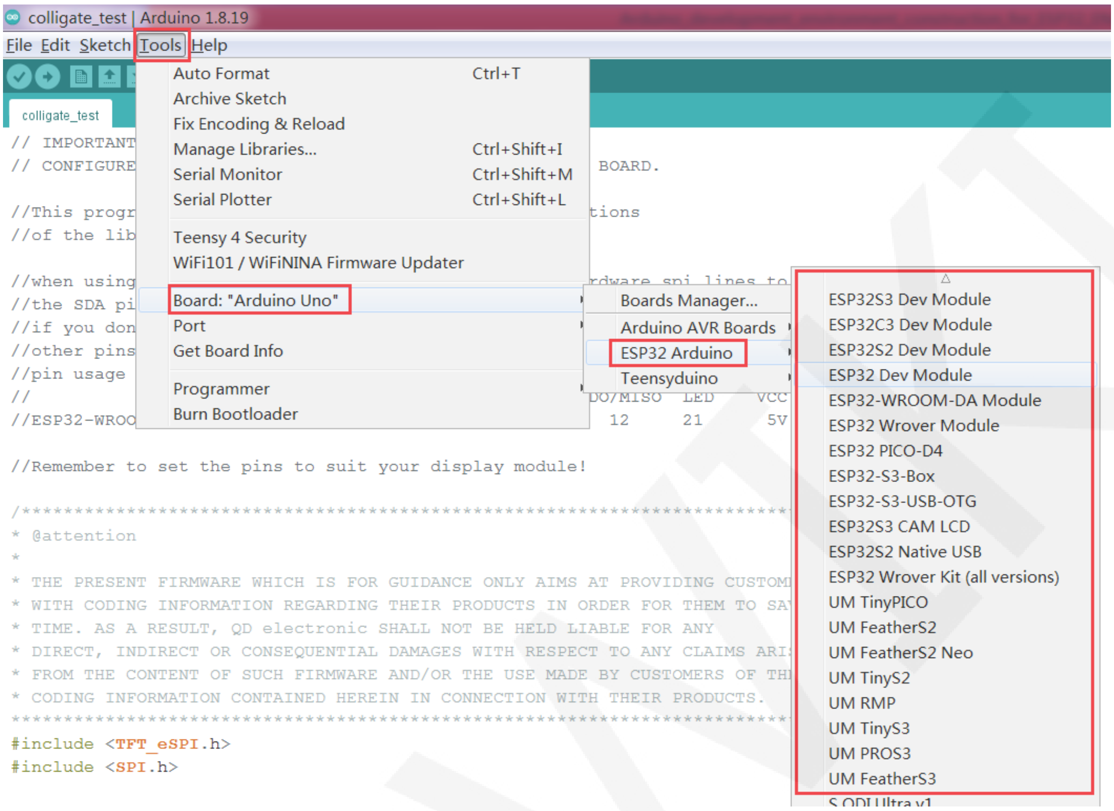
C. Open the Arduino IED installation directory, create a new **espressif** directory in the **hardware** directory, and then create an **esp32** directory in the **espressif** directory. Finally, copy all the support library files extracted in the previous step (i.e. all files in the **arduino-esp32-master** directory) to the **esp32** directory, as shown in the following figure:


« Arduino ▸ hardware ▸ espressif ▸ esp32 ▸
搜索 es...
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 ▾ 包含到库中 ▾ 共享 ▾ 新建文件夹

名称	修改日期	类型	大小
.github	2023/2/17 15:57	文件夹	
cores	2023/2/17 15:57	文件夹	
docs	2023/2/17 15:57	文件夹	
libraries	2023/2/17 15:57	文件夹	
package	2023/2/17 15:57	文件夹	
tests	2023/2/17 15:57	文件夹	
tools	2023/2/17 15:58	文件夹	
variants	2023/2/17 15:58	文件夹	
.gitignore	2023/2/15 21:54	文本文档	1 KB
.gitmodules	2023/2/15 21:54	文本文档	0 KB
boards.txt	2023/2/15 21:54	TXT 文件	989 KB
CMakeLists.txt	2023/2/15 21:54	TXT 文件	10 KB
CODE_OF_CONDUCT.md	2023/2/15 21:54	MD 文件	6 KB
idf_component.yml	2023/2/15 21:54	Yaml 源文件	1 KB
Kconfig.projbuild	2023/2/15 21:54	PROJBUILD 文件	12 KB
LICENSE.md	2023/2/15 21:54	MD 文件	26 KB
package.json	2023/2/15 21:54	JSON 源文件	1 KB
platform.txt	2023/2/15 21:54	TXT 文件	88 KB
programmers.txt	2023/2/15 21:54	TXT 文件	1 KB
README.md	2023/2/15 21:54	MD 文件	5 KB

20 个对象

D. Open the **Arduino IED** software, click the "**Tools**" button, select the **board** option, and you will see the **ESP32 Arduino** option. After selecting this option, various ESP32 development board models will appear, as shown in the following figure:



```
// IMPORTANT
// CONFIGURE                                         BOARD.

//This progr                                         tions
//of the lib

//when using                                         rdware spi lines to
//the SDA pi
//if you don
//other pins
//pin usage
//                                                   DO/MISO  LED    VCC
//ESP32-WROO                                           12      21     5V

//Remember to set the pins to suit your display module!

/**********************************************************************
* @attention
*
* THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMI
* WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SA
* TIME. AS A RESULT, QD electronic SHALL NOT BE HELD LIABLE FOR ANY
* DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARI
* FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF TH
* CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.
***********************************************************************
#include <TFT_eSPI.h>
#include <SPI.h>
```

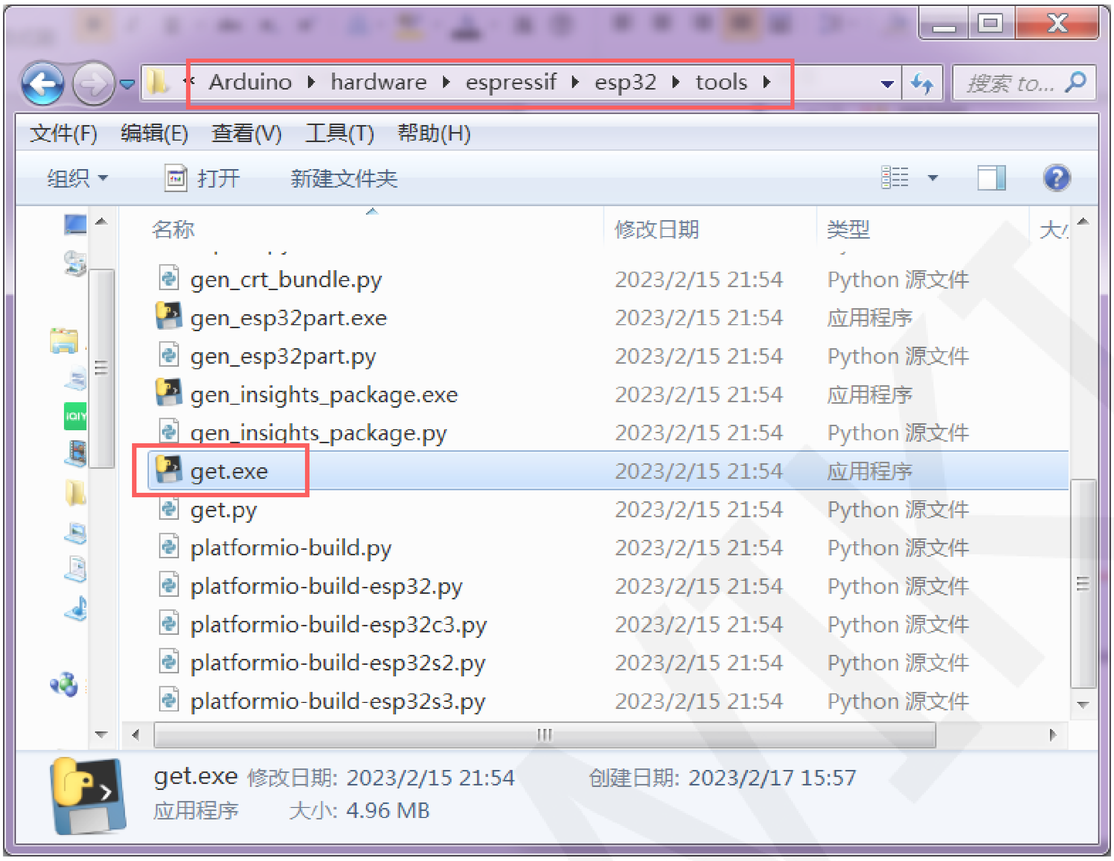
3. Install Compilation Tools

This step is only for installing the Arduino-ESP32 support library using manual methods. If it is installed using the Arduino IDE method, this step is not necessary.

There are two methods for installing compilation tools: e**xecuting script installation** and **manual installation**.

The installation steps for executing the script are as follows:

A. Open the "**hardware\espressif\esp32\tools**" directory under the Arduino IED installation directory and locate the "**get. exe**" file, as shown in the following figure:


Arduino ▸ hardware ▸ espressif ▸ esp32 ▸ tools ▸
搜索 to...
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 ▾ 打开 新建文件夹

名称	修改日期	类型
gen_crt_bundle.py	2023/2/15 21:54	Python 源文件
gen_esp32part.exe	2023/2/15 21:54	应用程序
gen_esp32part.py	2023/2/15 21:54	Python 源文件
gen_insights_package.exe	2023/2/15 21:54	应用程序
gen_insights_package.py	2023/2/15 21:54	Python 源文件
get.exe	2023/2/15 21:54	应用程序
get.py	2023/2/15 21:54	Python 源文件
platformio-build.py	2023/2/15 21:54	Python 源文件
platformio-build-esp32.py	2023/2/15 21:54	Python 源文件
platformio-build-esp32c3.py	2023/2/15 21:54	Python 源文件
platformio-build-esp32s2.py	2023/2/15 21:54	Python 源文件
platformio-build-esp32s3.py	2023/2/15 21:54	Python 源文件

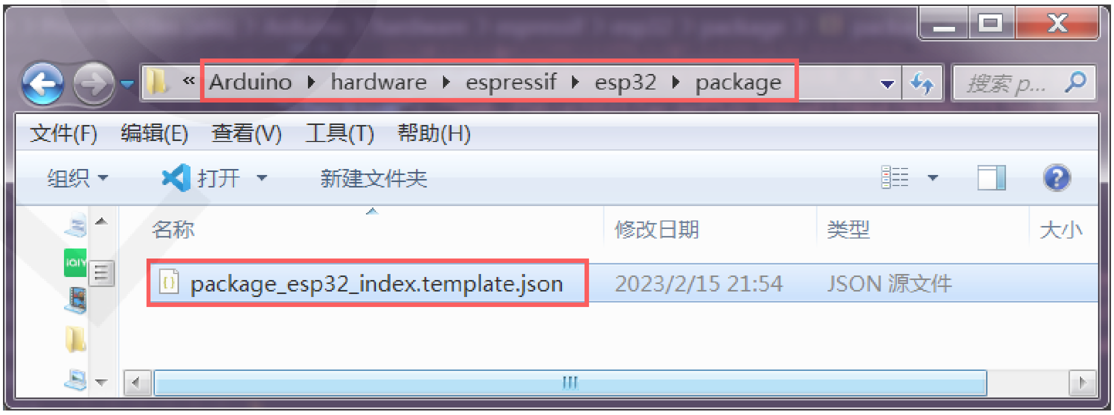
get.exe 修改日期: 2023/2/15 21:54 创建日期: 2023/2/17 15:57
应用程序 大小: 4.96 MB

B. Double click on the '**get. exe**' file to download and install the compilation tool, which takes a bit longer to execute.

If the get.exe program cannot install the compilation tool, then manual installation is required.

The manual installation steps are as follows:

A. Open the "**hardware\espressif\esp32\package**" directory under the Arduino IED installation directory and locate the "**package_esp32_index. template. json**" file, as shown in the following figure:


« Arduino ▸ hardware ▸ espressif ▸ esp32 ▸ package
搜索 p...
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 ▾ 打开 ▾ 新建文件夹

名称	修改日期	类型	大小
package_esp32_index.template.json	2023/2/15 21:54	JSON 源文件	

B. Open ”**package_esp32_index.template.json**” file, Find the download links for the three files: ”**xtensa-esp32-elf-gcc** “,” **esptool_py** “, and” **mkspiffs**”. Wherein:

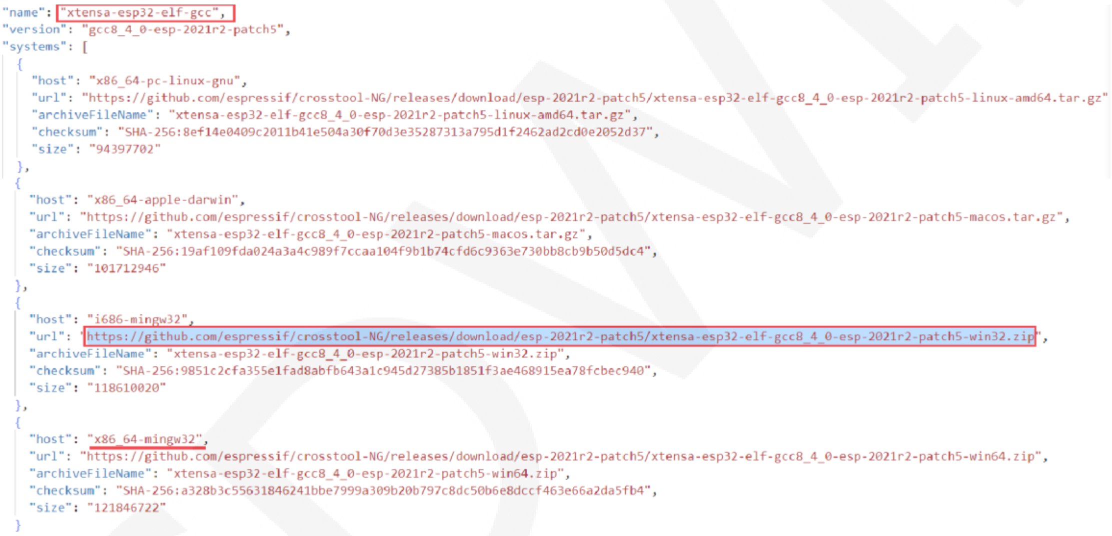
xtensa-esp32-elf-gcc is the ESP32 cross compilation toolchain

esptool_py is a tool for burning (downloading) ESP32 software

mkspiffs is the file system used on SPI NOR flash devices

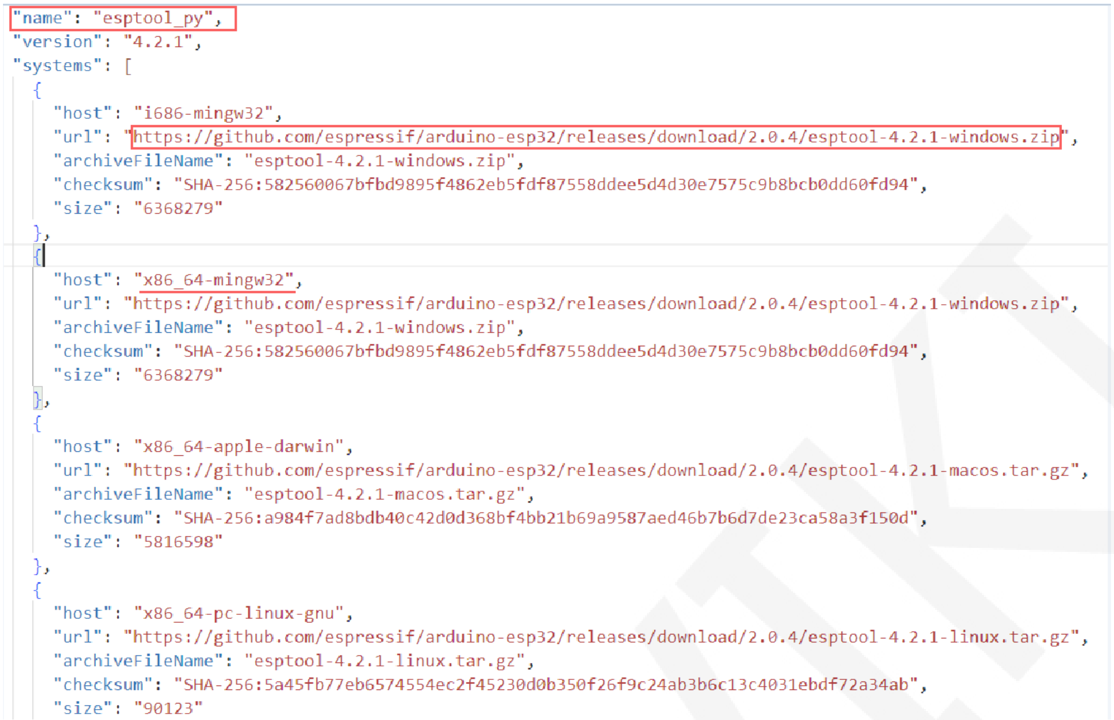
Each file corresponds to download links for multiple platforms, such as Linux, Apple, etc. Here, only the i686-mingw32 platform, which is the 32-bit Windows platform, is required. Of course, if you are using a 64 bit Windows system, you can also use x86_ Link to the 64 mingw32 platform. The three file links are obtained as shown in the following figure:

xtensa-esp32-elf-gcc:



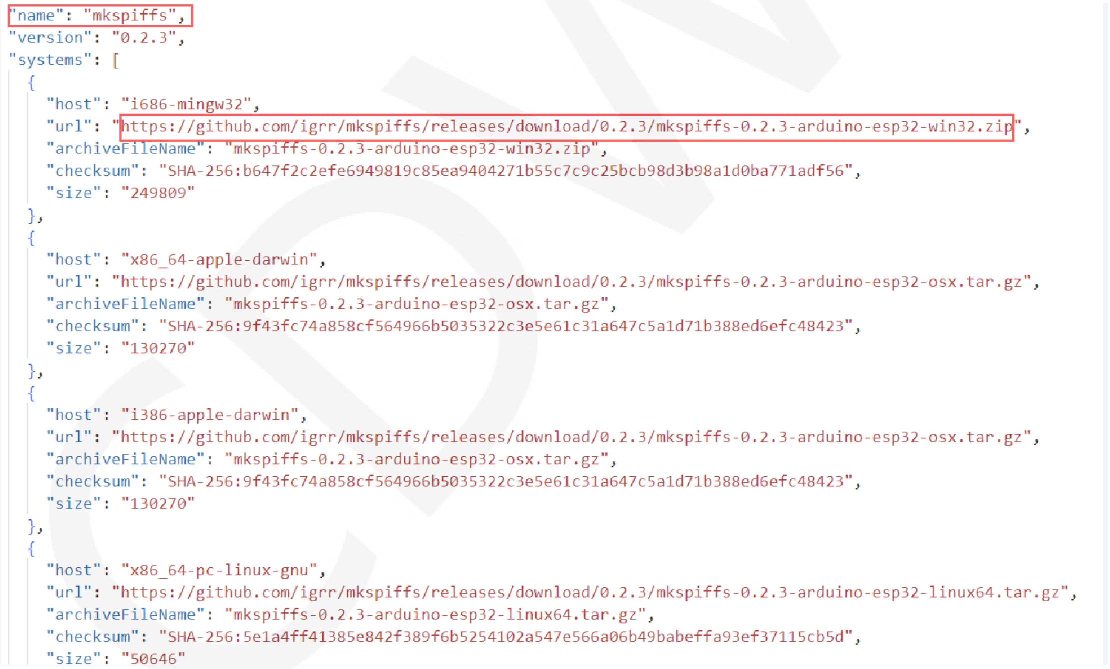
```
"name": "xtensa-esp32-elf-gcc",
"version": "gcc8_4_0-esp-2021r2-patch5",
"systems": [
  {
    "host": "x86_64-pc-linux-gnu",
    "url": "https://github.com/espressif/crosstool-NG/releases/download/esp-2021r2-patch5/xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-linux-amd64.tar.gz"
    "archiveFileName": "xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-linux-amd64.tar.gz",
    "checksum": "SHA-256:8ef14e0409c2011b41e504a30f70d3e35287313a795d1f2462ad2cd0e2052d37",
    "size": "94397702"
  },
  {
    "host": "x86_64-apple-darwin",
    "url": "https://github.com/espressif/crosstool-NG/releases/download/esp-2021r2-patch5/xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-macos.tar.gz",
    "archiveFileName": "xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-macos.tar.gz",
    "checksum": "SHA-256:19af109fda024a3a4c989f7ccaa104f9b1b74cfd6c9363e730bb8cb9b50d5dc4",
    "size": "101712946"
  },
  {
    "host": "i686-mingw32",
    "url": "https://github.com/espressif/crosstool-NG/releases/download/esp-2021r2-patch5/xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win32.zip",
    "archiveFileName": "xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win32.zip",
    "checksum": "SHA-256:9851c2cfa355e1fad8abfb643a1c945d27385b1851f3ae468915ea78fcbec940",
    "size": "118610020"
  },
  {
    "host": "x86_64-mingw32",
    "url": "https://github.com/espressif/crosstool-NG/releases/download/esp-2021r2-patch5/xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win64.zip",
    "archiveFileName": "xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win64.zip",
    "checksum": "SHA-256:a328b3c55631846241bbe7999a309b20b797c8dc50b6e8dccf463e66a2da5fb4",
    "size": "121846722"
  }
```

esptool_py:



```
"name": "esptool_py",
"version": "4.2.1",
"systems": [
  {
    "host": "i686-mingw32",
    "url": "https://github.com/espressif/arduino-esp32/releases/download/2.0.4/esptool-4.2.1-windows.zip",
    "archiveFileName": "esptool-4.2.1-windows.zip",
    "checksum": "SHA-256:582560067bfbd9895f4862eb5fdf87558ddee5d4d30e7575c9b8bcb0dd60fd94",
    "size": "6368279"
  },
  {
    "host": "x86_64-mingw32",
    "url": "https://github.com/espressif/arduino-esp32/releases/download/2.0.4/esptool-4.2.1-windows.zip",
    "archiveFileName": "esptool-4.2.1-windows.zip",
    "checksum": "SHA-256:582560067bfbd9895f4862eb5fdf87558ddee5d4d30e7575c9b8bcb0dd60fd94",
    "size": "6368279"
  },
  {
    "host": "x86_64-apple-darwin",
    "url": "https://github.com/espressif/arduino-esp32/releases/download/2.0.4/esptool-4.2.1-macos.tar.gz",
    "archiveFileName": "esptool-4.2.1-macos.tar.gz",
    "checksum": "SHA-256:a984f7ad8bdb40c42d0d368bf4bb21b69a9587aed46b7b6d7de23ca58a3f150d",
    "size": "5816598"
  },
  {
    "host": "x86_64-pc-linux-gnu",
    "url": "https://github.com/espressif/arduino-esp32/releases/download/2.0.4/esptool-4.2.1-linux.tar.gz",
    "archiveFileName": "esptool-4.2.1-linux.tar.gz",
    "checksum": "SHA-256:5a45fb77eb6574554ec2f45230d0b350f26f9c24ab3b6c13c4031ebdf72a34ab",
    "size": "90123"
```

mkspiffs:



```
"name": "mkspiffs",
"version": "0.2.3",
"systems": [
  {
    "host": "i686-mingw32",
    "url": "https://github.com/igrr/mkspiffs/releases/download/0.2.3/mkspiffs-0.2.3-arduino-esp32-win32.zip",
    "archiveFileName": "mkspiffs-0.2.3-arduino-esp32-win32.zip",
    "checksum": "SHA-256:b647f2c2efe6949819c85ea9404271b55c7c9c25bcb98d3b98a1d0ba771adf56",
    "size": "249809"
  },
  {
    "host": "x86_64-apple-darwin",
    "url": "https://github.com/igrr/mkspiffs/releases/download/0.2.3/mkspiffs-0.2.3-arduino-esp32-osx.tar.gz",
    "archiveFileName": "mkspiffs-0.2.3-arduino-esp32-osx.tar.gz",
    "checksum": "SHA-256:9f43fc74a858cf564966b5035322c3e5e61c31a647c5a1d71b388ed6efc48423",
    "size": "130270"
  },
  {
    "host": "i386-apple-darwin",
    "url": "https://github.com/igrr/mkspiffs/releases/download/0.2.3/mkspiffs-0.2.3-arduino-esp32-osx.tar.gz",
    "archiveFileName": "mkspiffs-0.2.3-arduino-esp32-osx.tar.gz",
    "checksum": "SHA-256:9f43fc74a858cf564966b5035322c3e5e61c31a647c5a1d71b388ed6efc48423",
    "size": "130270"
  },
  {
    "host": "x86_64-pc-linux-gnu",
    "url": "https://github.com/igrr/mkspiffs/releases/download/0.2.3/mkspiffs-0.2.3-arduino-esp32-linux64.tar.gz",
    "archiveFileName": "mkspiffs-0.2.3-arduino-esp32-linux64.tar.gz",
    "checksum": "SHA-256:5e1a4ff41385e842f389f6b5254102a547e566a06b49babeffa93ef37115cb5d",
    "size": "50646"
```

The three extracted links are as follows:

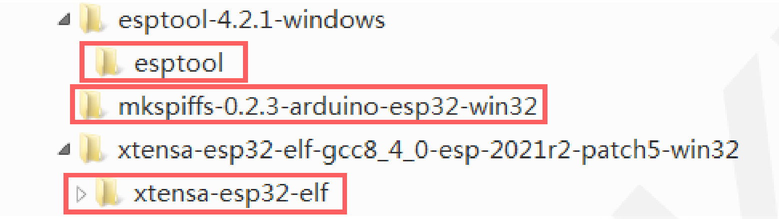
https://github.com/espressif/crosstool-NG/releases/download/esp-2021r2-patch5/xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win32.zip

https://github.com/espressif/arduino-esp32/releases/download/2.0.4/esptool-4.2.1-windows.zip

https://github.com/igrr/mkspiffs/releases/download/0.2.3/mkspiffs-0.2.3-arduino-esp32-win32.zip

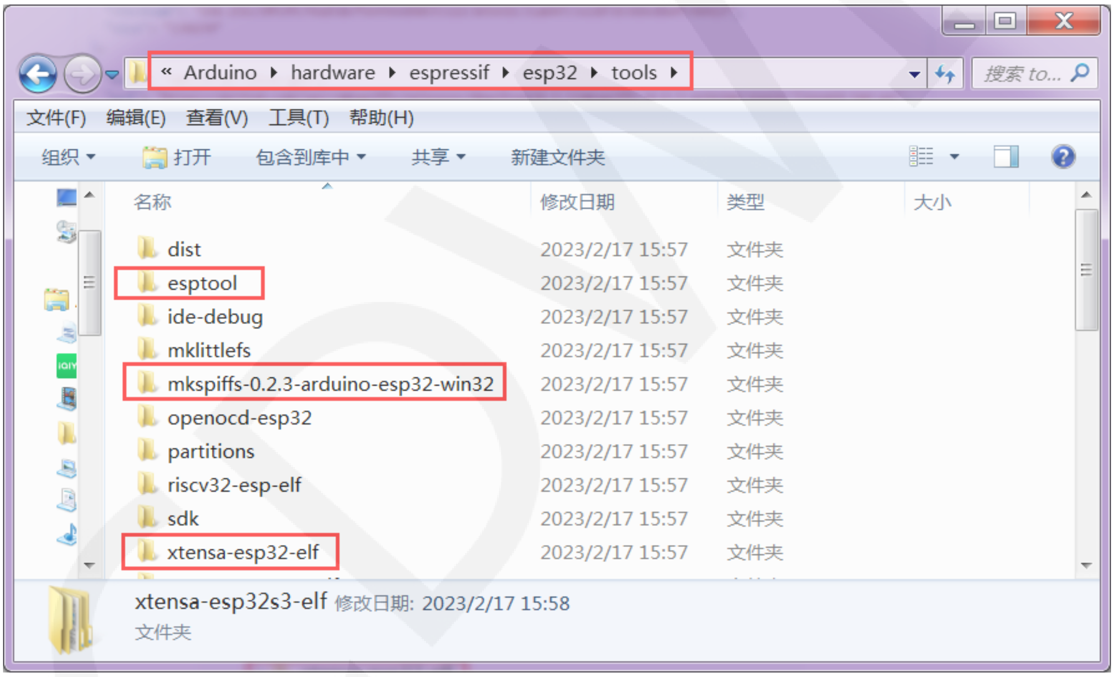
You can directly click to download, or if the speed is slow, you can copy it to a browser or downloader to download.

C. After downloading three files, unzip them, and the unzipped folder is shown in the following figure:



esptool-4.2.1-windows
esptool
mkspiffs-0.2.3-arduino-esp32-win32
xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win32
xtensa-esp32-elf

D. Then copy the extracted folder to the "**hardware\espressif\esp32\tools**" directory under the Arduino IED installation directory, as shown in the following figure:



« Arduino ▸ hardware ▸ espressif ▸ esp32 ▸ tools ▸

搜索 to...
文件(F) 编辑(E) 查看(V) 工具(T) 帮助(H)
组织 ▾ 打开 包含到库中 ▾ 共享 ▾ 新建文件夹

名称	修改日期	类型	大小
dist	2023/2/17 15:57	文件夹	
esptool	2023/2/17 15:57	文件夹	
ide-debug	2023/2/17 15:57	文件夹	
mklittlefs	2023/2/17 15:57	文件夹	
mkspiffs-0.2.3-arduino-esp32-win32	2023/2/17 15:57	文件夹	
openocd-esp32	2023/2/17 15:57	文件夹	
partitions	2023/2/17 15:57	文件夹	
riscv32-esp-elf	2023/2/17 15:57	文件夹	
sdk	2023/2/17 15:57	文件夹	
xtensa-esp32-elf	2023/2/17 15:57	文件夹	

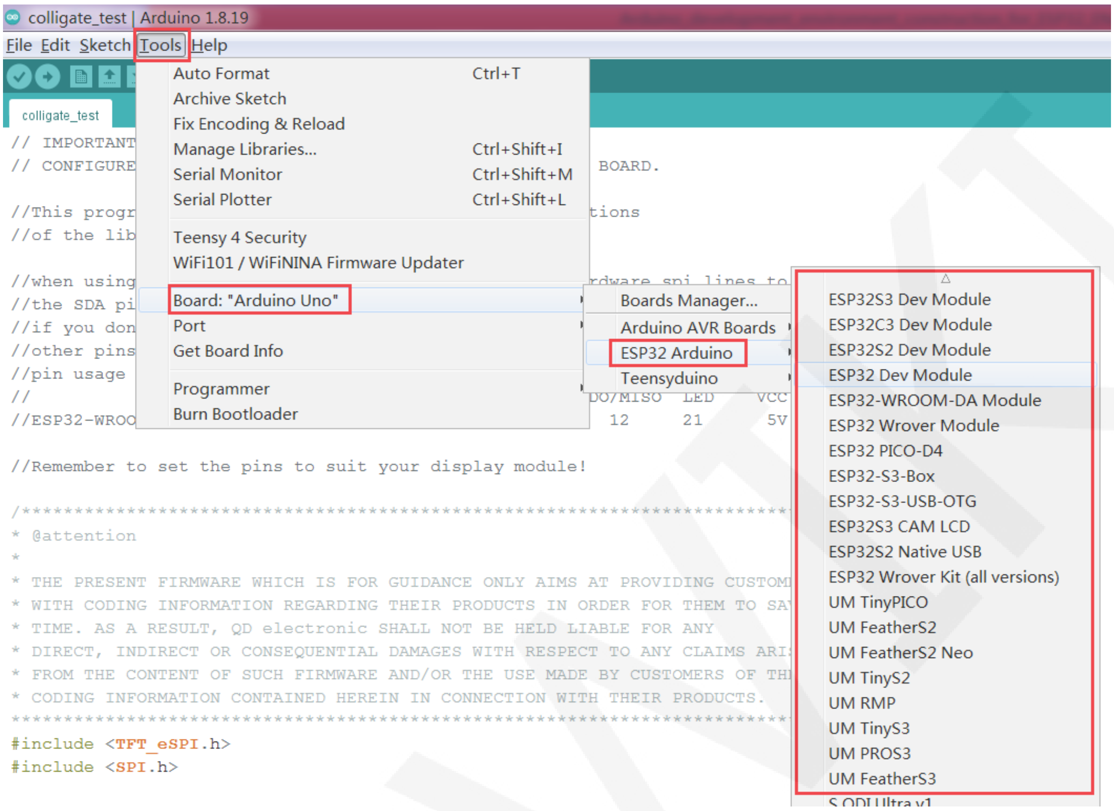
xtensa-esp32s3-elf 修改日期: 2023/2/17 15:58
文件夹

4. Compile and download the program

After the development environment is set up, it is necessary to compile and download the program to verify whether the environment is set up correctly.

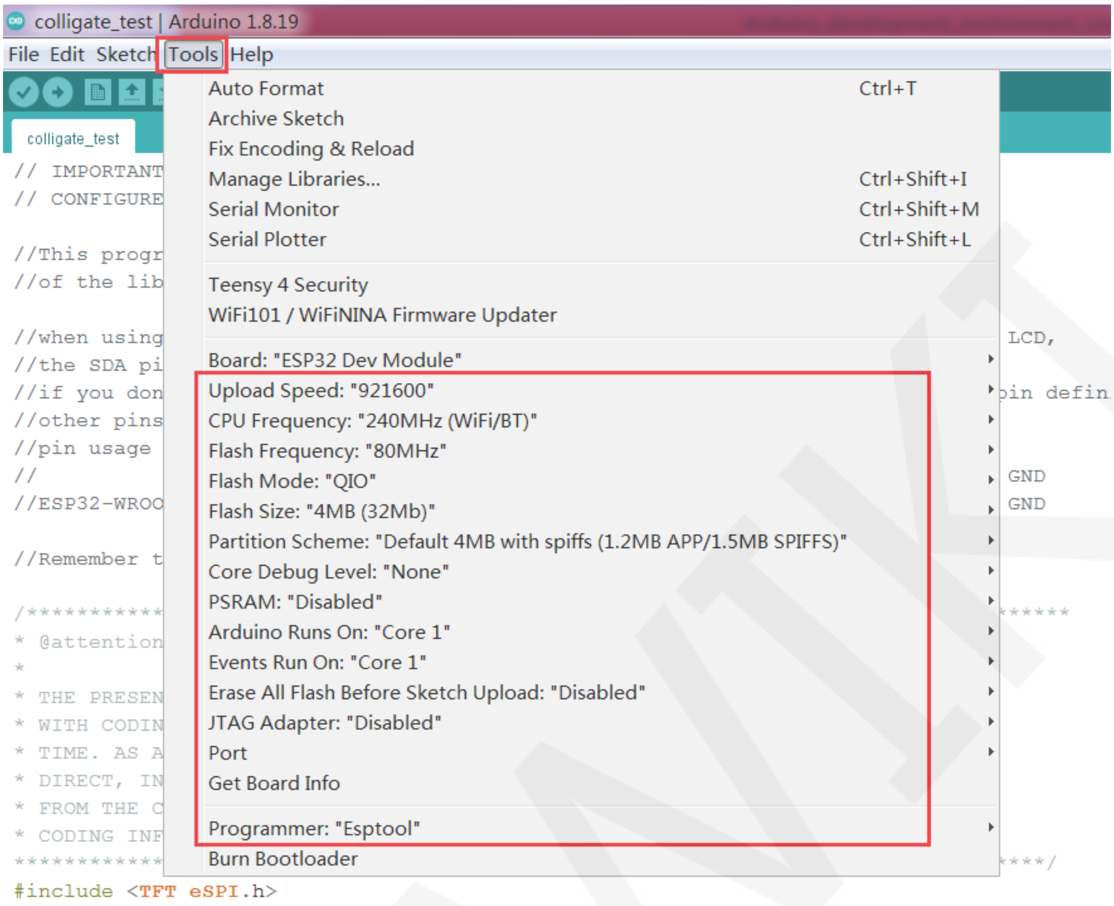
A. Connect the ESP32 development board to the PC and power up the development board.

B. Open the Arduino IDE software, click the **Tools** button, and then select **Board ->ESP32 Arduino ->ESP32 Dev Module**, as shown in the following figure:



```
// IMPORTANT
// CONFIGURE                                                  BOARD.

//This progr                                                 tions
//of the lib

//when using                                                 rdware spi lines to
//the SDA pi
//if you don
//other pins
//pin usage
//                                                           DO/MISO   LED    VCC
//ESP32-WROO                                                   12      21     5V

//Remember to set the pins to suit your display module!

/**************************************************************************
* @attention
*
* THE PRESENT FIRMWARE WHICH IS FOR GUIDANCE ONLY AIMS AT PROVIDING CUSTOMI
* WITH CODING INFORMATION REGARDING THEIR PRODUCTS IN ORDER FOR THEM TO SAV
* TIME. AS A RESULT, QD electronic SHALL NOT BE HELD LIABLE FOR ANY
* DIRECT, INDIRECT OR CONSEQUENTIAL DAMAGES WITH RESPECT TO ANY CLAIMS ARI
* FROM THE CONTENT OF SUCH FIRMWARE AND/OR THE USE MADE BY CUSTOMERS OF THI
* CODING INFORMATION CONTAINED HEREIN IN CONNECTION WITH THEIR PRODUCTS.
***************************************************************************
#include <TFT_eSPI.h>
#include <SPI.h>
```

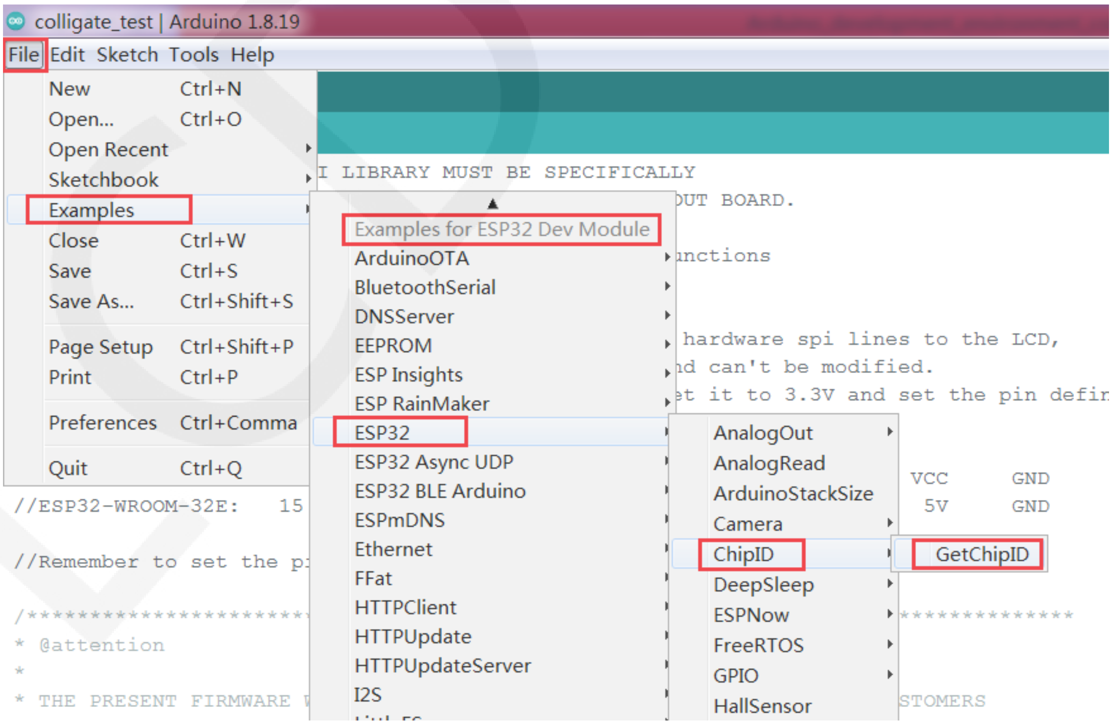
C. Click the **Tools** button again, and you can see that the **Board** has become **ESP32**. Next, according to the configuration of the ESP32 module you are using, set the **Upload Speed**, **Flash**, **PSRAM**, **Port**, and **other option parameters**, as shown in the following figure:


colligate_test | Arduino 1.8.19
File Edit Sketch Tools Help
colligate_test
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
Teensy 4 Security
WiFi101 / WiFiNINA Firmware Updater
Port
Get Board Info
Burn Bootloader

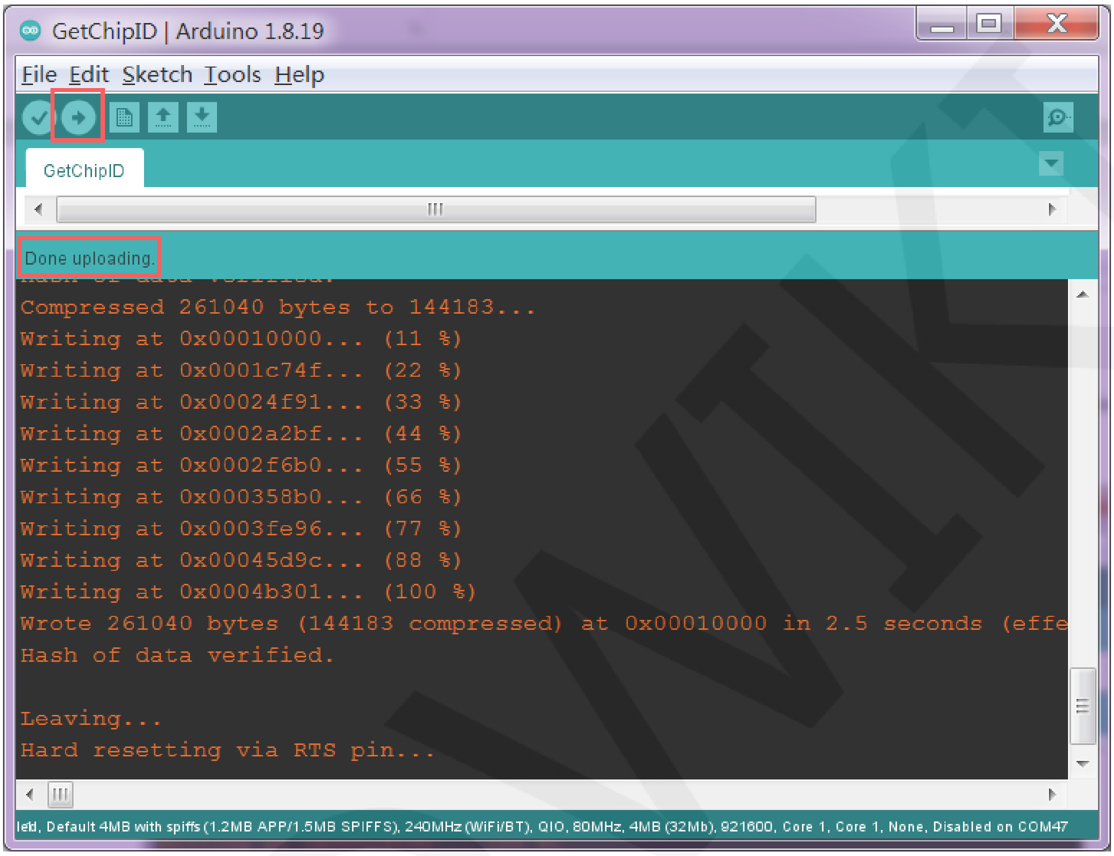
Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"
Flash Frequency: "80MHz"
Flash Mode: "QIO"
Flash Size: "4MB (32Mb)"
Partition Scheme: "Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS)"
Core Debug Level: "None"
PSRAM: "Disabled"
Arduino Runs On: "Core 1"
Events Run On: "Core 1"
Erase All Flash Before Sketch Upload: "Disabled"
JTAG Adapter: "Disabled"
Programmer: "Esptool"

```
#include <TFT_eSPI.h>
```

D. Click the **File** button, then select the **Examples** option, and then select an ESP32 example, as shown in the following figure (select **GetChipID** example here):



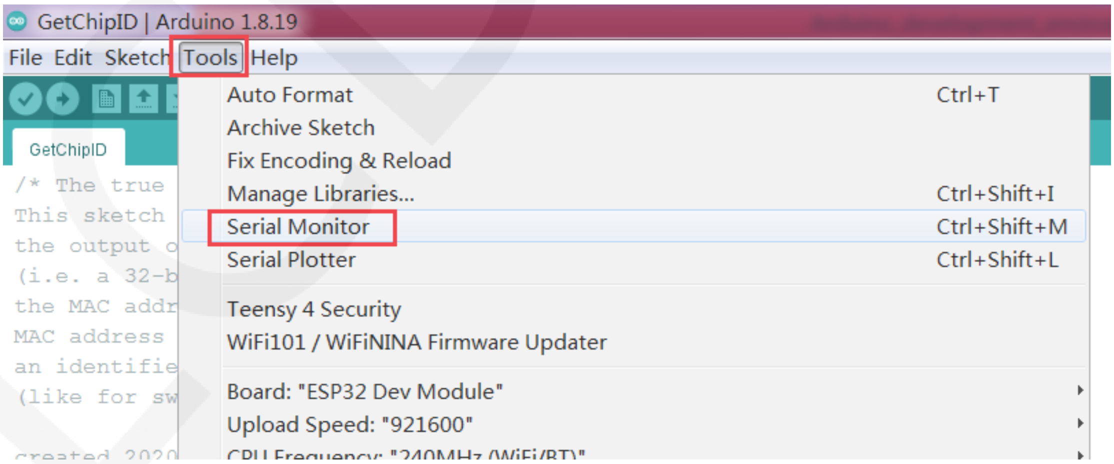
E. Click the **Upload** button, and the program will be compiled and burned. When the "**Done uploading**" prompt appears, it indicates that the program has been compiled and burned successfully, as shown in the following figure:



```
Compressed 261040 bytes to 144183...
Writing at 0x00010000... (11 %)
Writing at 0x0001c74f... (22 %)
Writing at 0x00024f91... (33 %)
Writing at 0x0002a2bf... (44 %)
Writing at 0x0002f6b0... (55 %)
Writing at 0x000358b0... (66 %)
Writing at 0x0003fe96... (77 %)
Writing at 0x00045d9c... (88 %)
Writing at 0x0004b301... (100 %)
Wrote 261040 bytes (144183 compressed) at 0x00010000 in 2.5 seconds (effe
Hash of data verified.

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

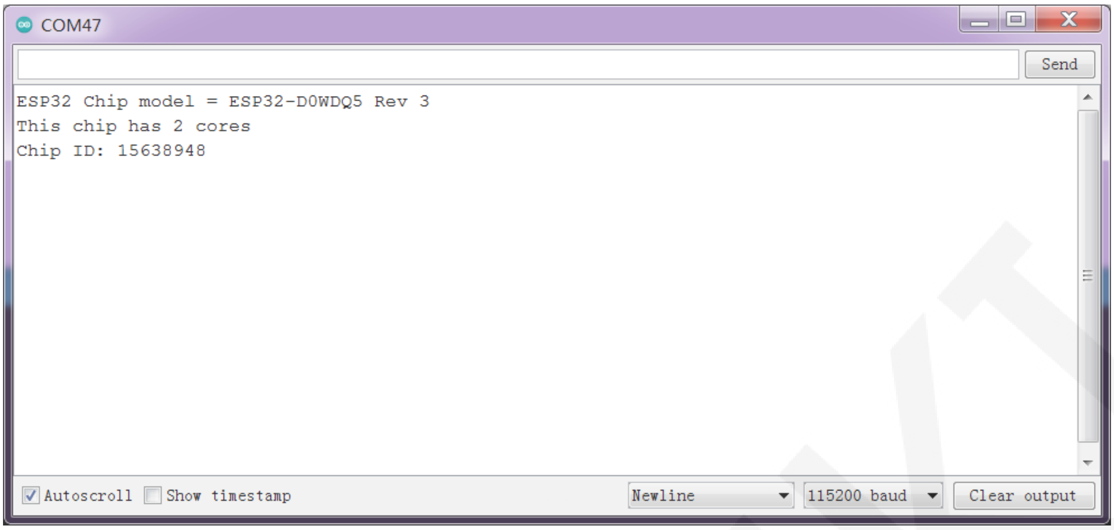
led, Default 4MB with spiffs (1.2MB APP/1.5MB SPIFFS), 240MHz (WiFi/BT), QIO, 80MHz, 4MB (32Mb), 921600, Core 1, Core 1, None, Disabled on COM47

F. Click on **Tools** ->**Serial Monitor**, and the serial port information interface will appear, as shown in the following figure:


GetChipID | Arduino 1.8.19
File Edit Sketch Tools Help
GetChipID
Auto Format Ctrl+T
Archive Sketch
Fix Encoding & Reload
Manage Libraries... Ctrl+Shift+I
Serial Monitor Ctrl+Shift+M
Serial Plotter Ctrl+Shift+L
Teensy 4 Security
WiFi101 / WiFiNINA Firmware Updater
Board: "ESP32 Dev Module"
Upload Speed: "921600"
CPU Frequency: "240MHz (WiFi/BT)"

```
/* The true
This sketch
the output o
(i.e. a 32-b
the MAC addr
MAC address
an identifie
(like for sw

created 2020
```

The serial information interface displays relevant information such as Chip ID, indicating successful program operation and successful environment setup, as shown in the following figure:



```
ESP32 Chip model = ESP32-D0WDQ5 Rev 3
This chip has 2 cores
Chip ID: 15638948
```