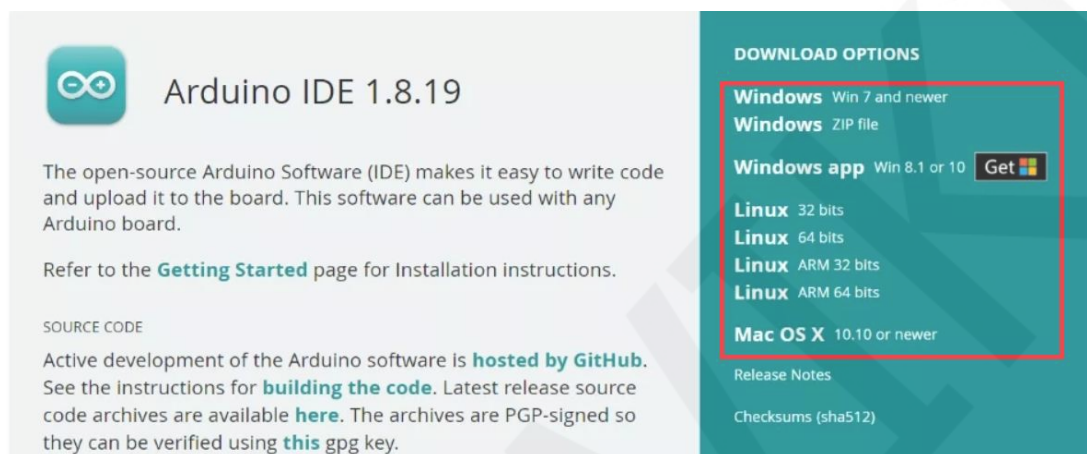


1、安装 Arduino IDE

从 arduino 官网下载安装包。

下载地址: <https://www.arduino.cc/en/software>

根据自己的 PC 机系统下载相应的安装包, 如下图所示(图片中的版本不一定是最新版的, 下载界面也不一定是最新界面):



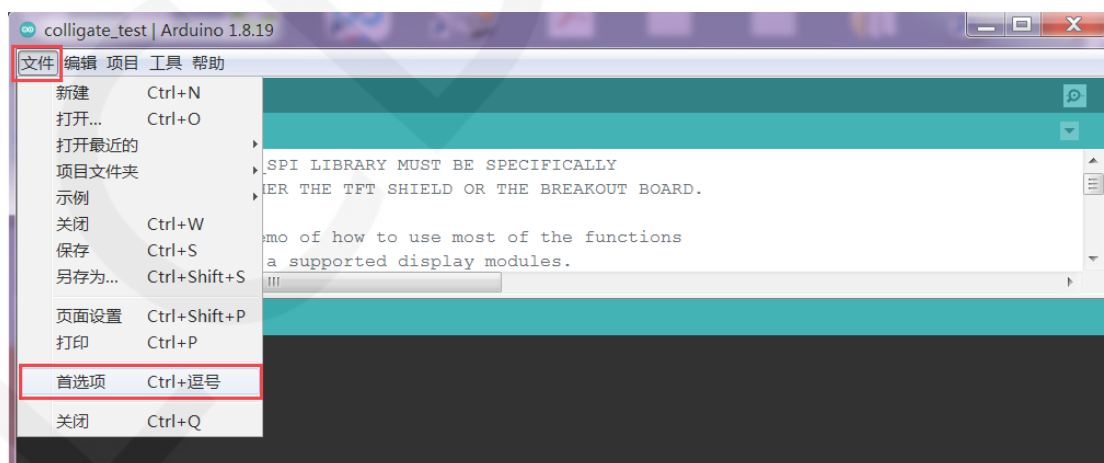
下载完成后, 解压并点击安装即可。

2、安装 Arduino-ESP32 支撑库

有两种安装方法: 借助 Arduino IDE 安装和手动安装。

借助 Arduino IDE 安装步骤如下:

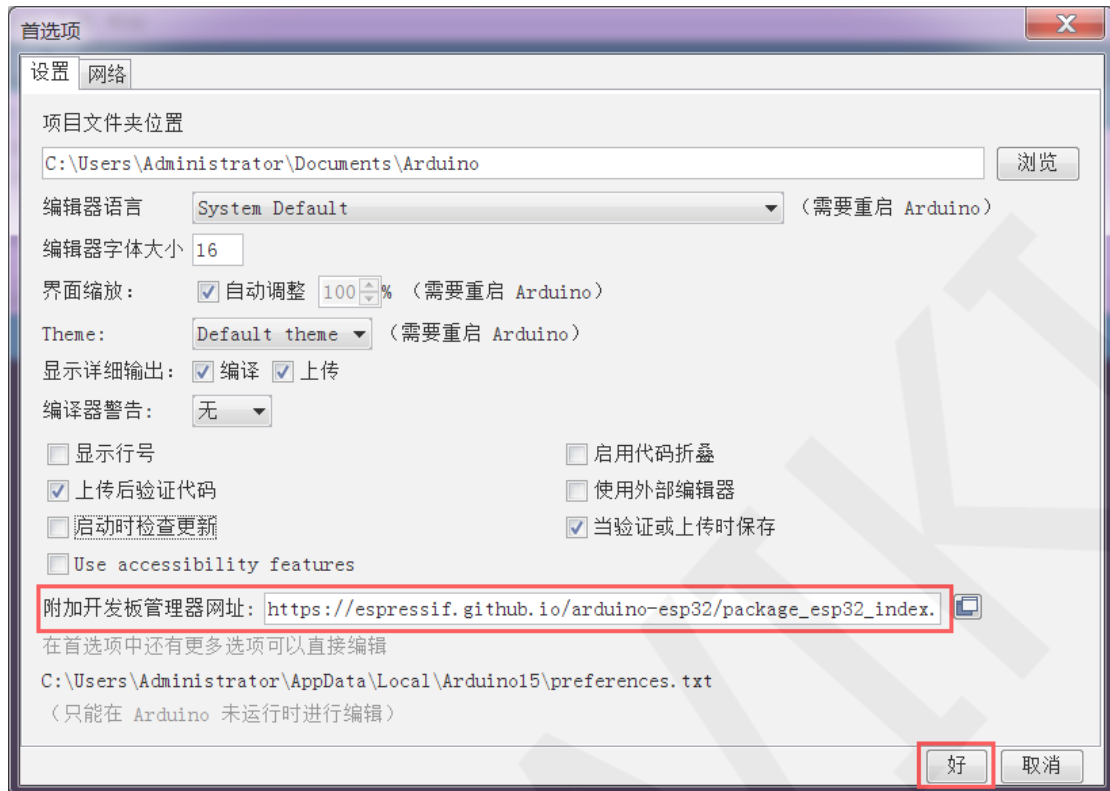
A、打开 Arduino IDE 软件, 点击文件 -> 首选项按钮, 如下图所示:



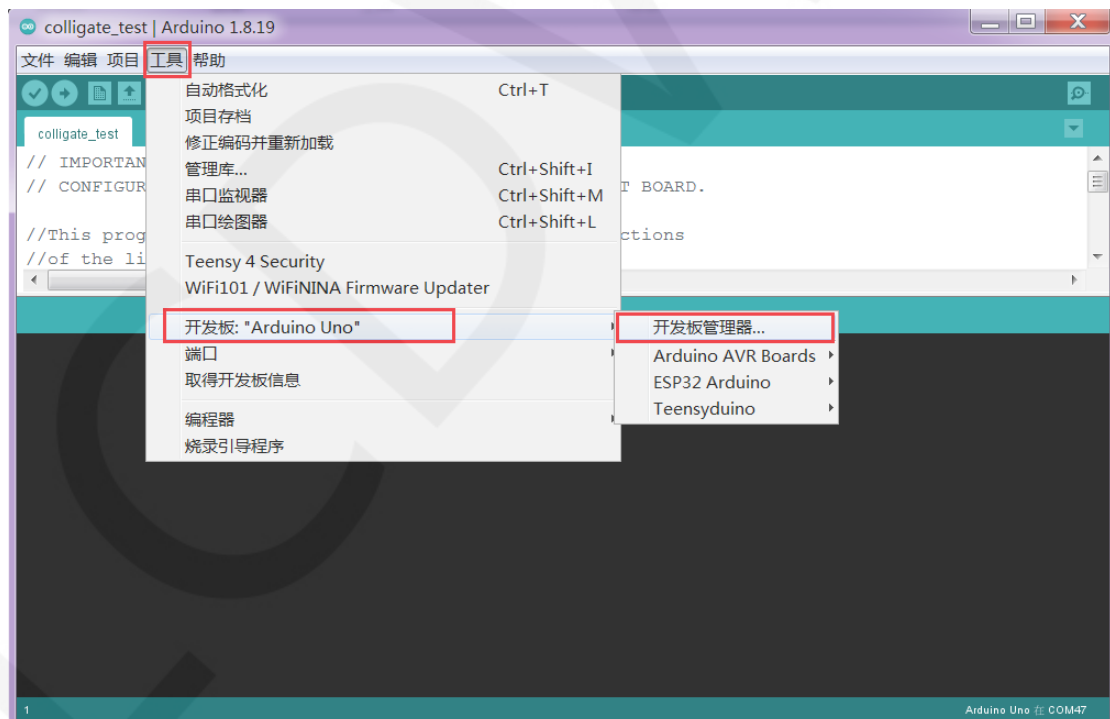
B、在弹出的首选项界面的附加开发板管理器网址里输入 ESP32 网址:

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

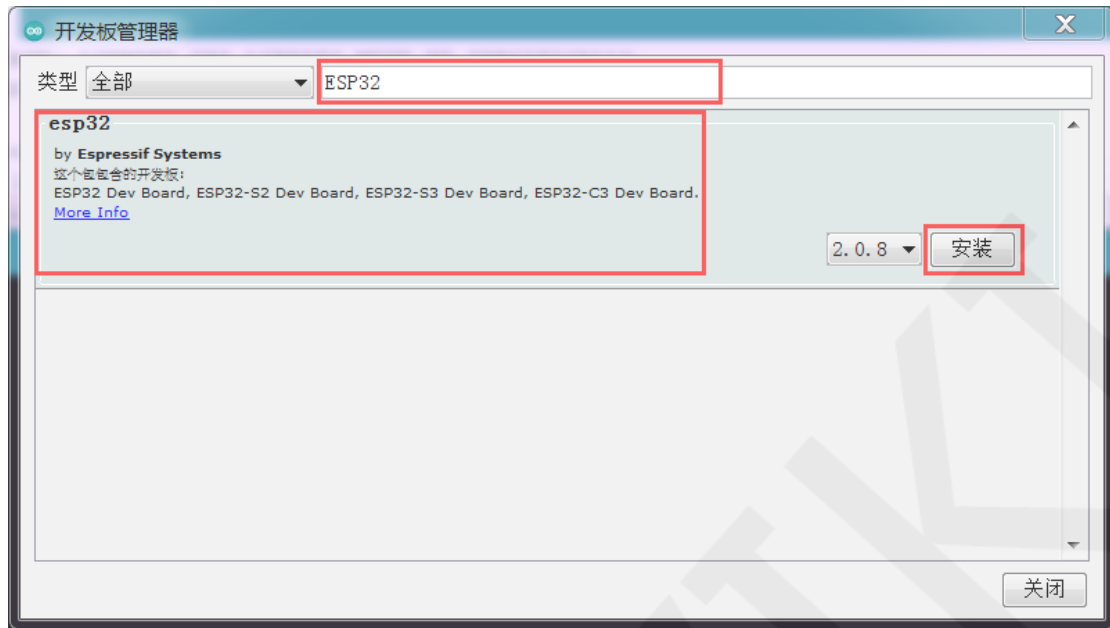
然后点击“好”按钮保存退出, 如下图所示:



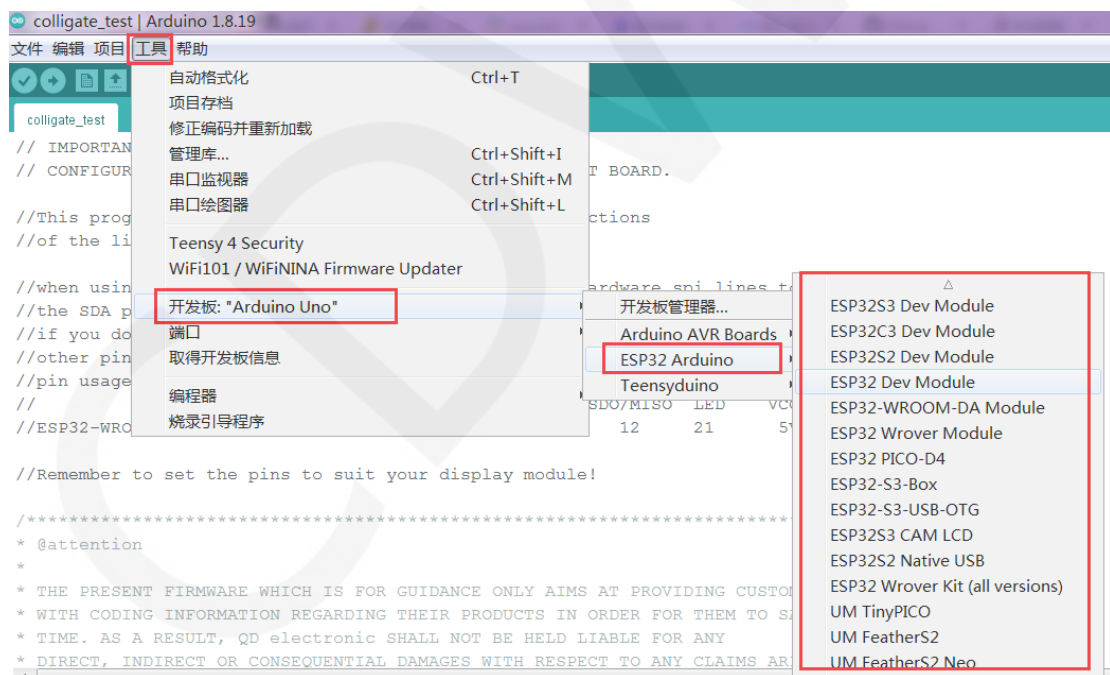
C、点击工具按钮，然后选择开发板->开发板管理器选项，如下图所示：



D、点击开发板管理器按钮，进入开发板管理器界面，待界面加载完成后，在搜索栏里输入“ESP32”，搜索 Arduino-ESP32 支撑库，搜索完成后，点击“安装”按钮进行安装，如下图所示：



E、安装时间比较长，如果安装过程出错，就多试几次。安装完成后，点击“工具”按钮，选择开发板选项，就会看到出现了 **ESP32 Arduino** 选项，选择该项后，会出现各种 ESP32 开发板型号，如下图所示：



如果借助 Arduino IDE 无法安装成功(可能的原因是网络环境较差,导致下载一直出错),那么就需要手动安装。

手动安装的步骤如下：

A、下载 Arduino-ESP32 支撑库，下载地址如下：

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

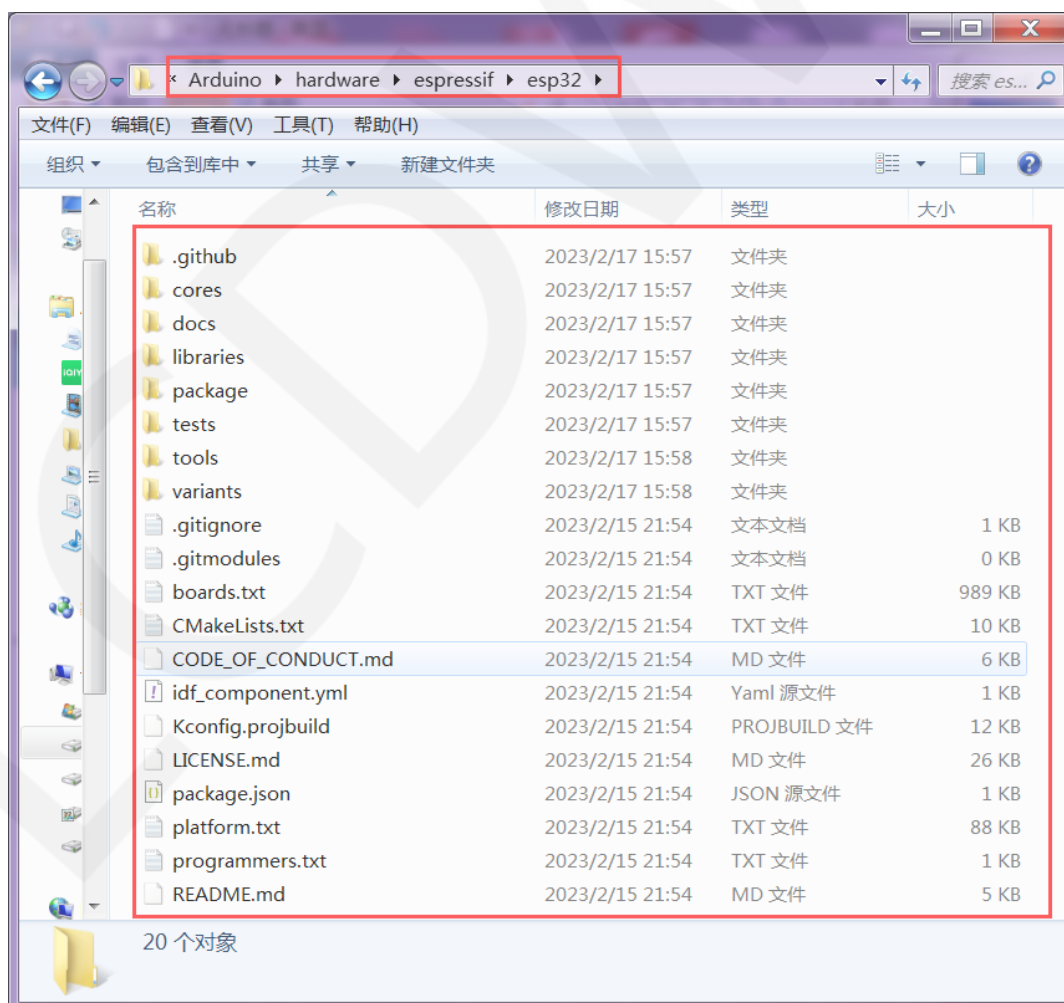
如果上面的地址不能下载，那推荐使用下面的下载地址：

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

B、支撑库下载成功后，将文件进行解压，如下图所示：

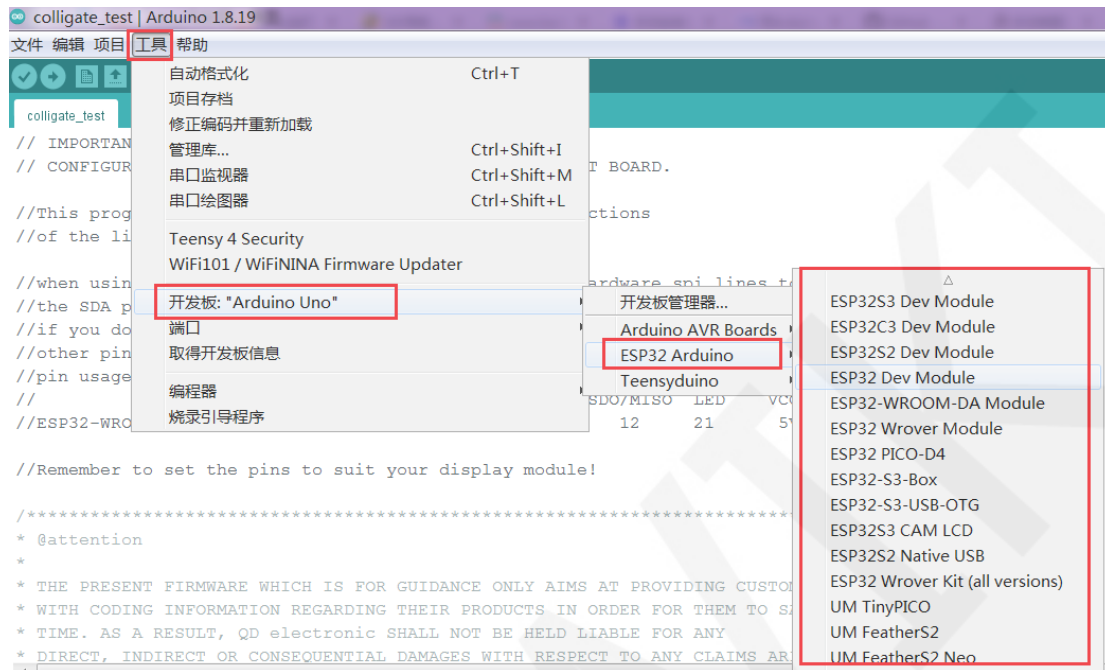


C、打开 Arduino IED 安装目录，在 **hardware** 目录下新建 **espressif** 目录，再在 **espressif** 目录下新建 **esp32** 目录，最后将上一步解压的支撑库文件（就是 **arduino-esp32-master** 目录下的所有文件）全部拷贝到 **esp32** 目录，如下图所示：



D、打开 Arduino IED 软件，点击“工具”按钮，选择开发板选项，就会看到出现了

ESP32 Arduino 选项，选择该项后，会出现各种 ESP32 开发板型号，如下图所示：



3、安装编译工具

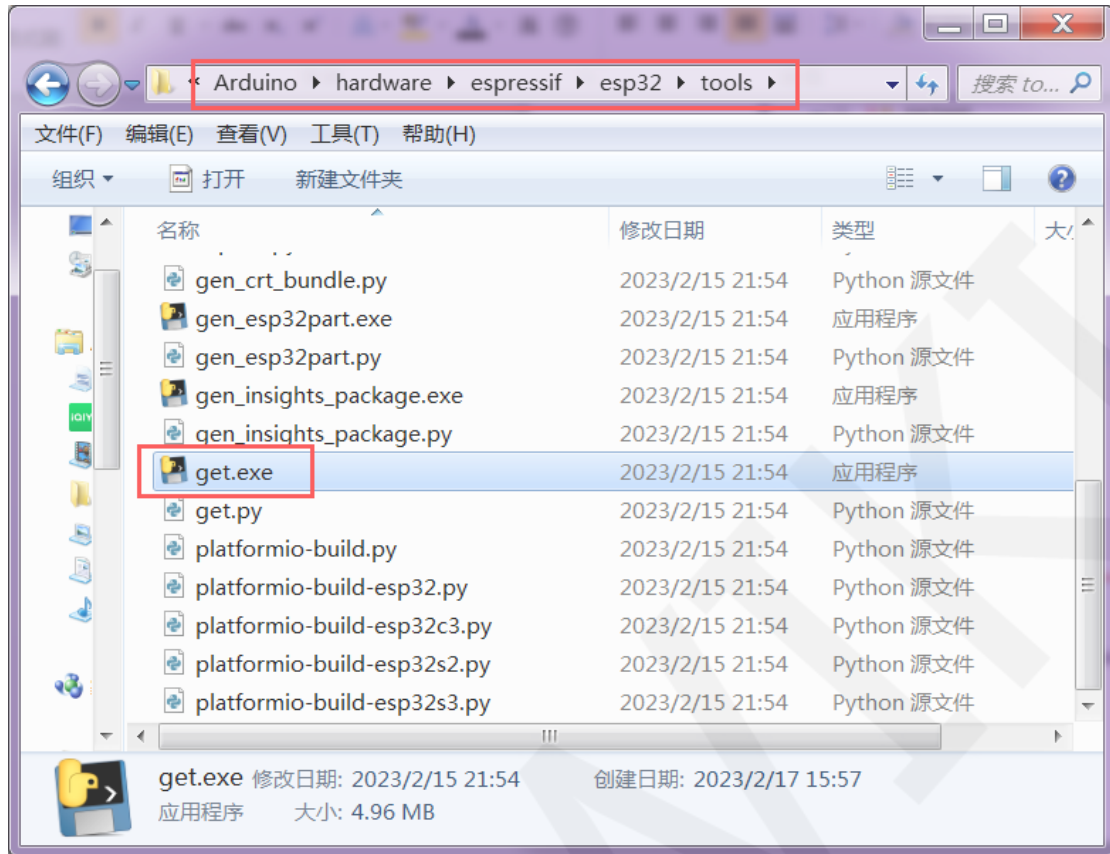
这步操作只针对使用手动方法安装 Arduino-ESP32 支撑库，如果是借助 Arduino IDE 的方法安装的，则不需要进行这步操作。

安装编译工具有两种方法：**执行脚本安装**和**手动安装**。

执行脚本安装步骤如下：

A、打开 Arduino IDE 安装目录下的“**hardware\espressif\esp32\tools**”目录，找到“**get.exe**”

文件，如下图所示：

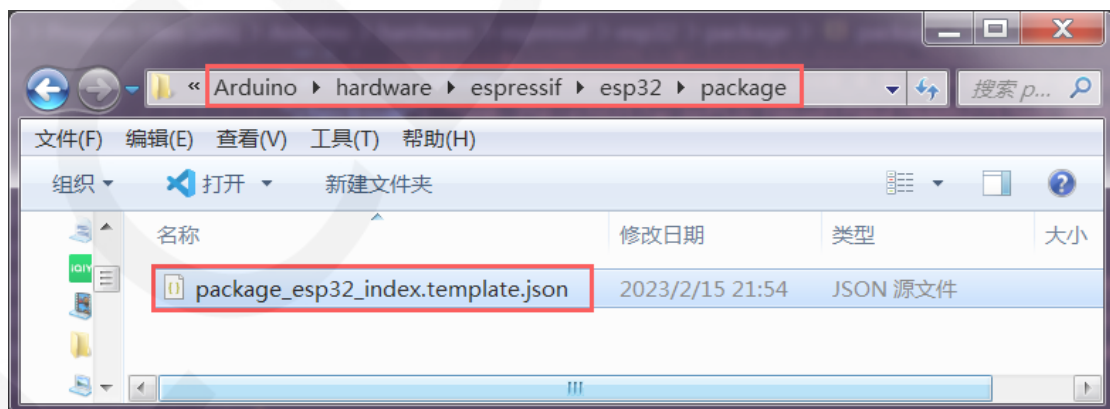


B、双击“get.exe”文件执行编译工具下载和安装，执行时间稍微有点长。

如果执行 get.exe 程序无法安装编译工具，那就需要手动安装。

手动安装步骤如下：

A、打开 Arduino IED 安装目录下的“hardware\espressif\esp32\package”目录，找到“package_esp32_index.template.json”文件，如下图所示：



B、打开“package_esp32_index.template.json”文件，找到“xtensa-esp32-elf-gcc”、“esptool_py”、“mkspiffs”这三个文件的下载链接。其中：

xtensa-esp32-elf-gcc 为 ESP32 交叉编译工具链

esptool_py 为 ESP32 软件烧录（下载）工具

mkspiffs 为用于 SPI NOR 闪存设备上的文件系统

每个文件对应应有多个平台的下载链接，例如 linux、apple 等，这里只需要 **i686-mingw32** 平台的，也就是 32 位 windows 平台的。当然如果使用的是系统是 64 位 windows，也可以使用 **x86_64-mingw32** 平台的链接。三个文件链接获取如下图所示：

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-256:19af109fda024a3a4c989f7c9aa104f9b1b74cf6dc9363e730bb8cb9b50d5dc4",
    "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:a328b3c55631846241bbe7999a309b20b797c8dc50b6e8dccc463e66a2da5fb4",
    "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-256:582560067bfb9895f4862eb5fdf87558dde5d4d30e7575c9b8bcb0dd60fd94",
    "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-256:582560067bfb9895f4862eb5fdf87558dde5d4d30e7575c9b8bcb0dd60fd94",
    "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:b647f2c2efe6949819c85ea9404271b55c7c9c25bc98d3b98a1d0ba771adf56",
      "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"
    }
  ]
}
```

提取出来的三个链接如下:

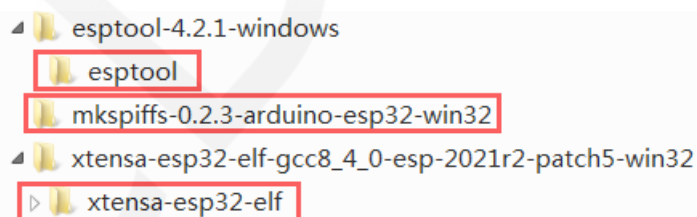
https://github.com/espressif/crostoool-NG/releases/download/esp-2021r2-patch5/xtensa-esp32-elf-gcc8_4_0-esp-2021r2-patch5-win32.zip

<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>

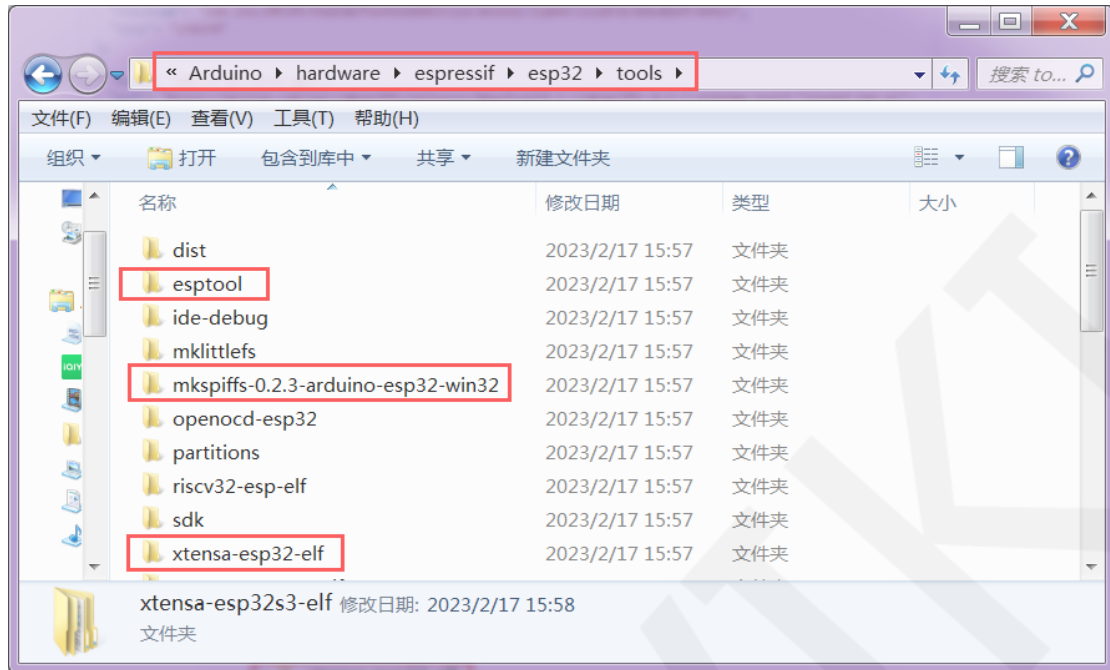
可以直接点击下载, 如果速度慢, 也可以拷贝到浏览器或者下载器里下载。

C、三个文件下后, 进行解压, 解压后的文件夹如下图所示:



D、然后将解压后的文件夹拷贝到 Arduino IED 安装目录下的

“hardware\espressif\esp32\tools” 目录, 如下图所示:



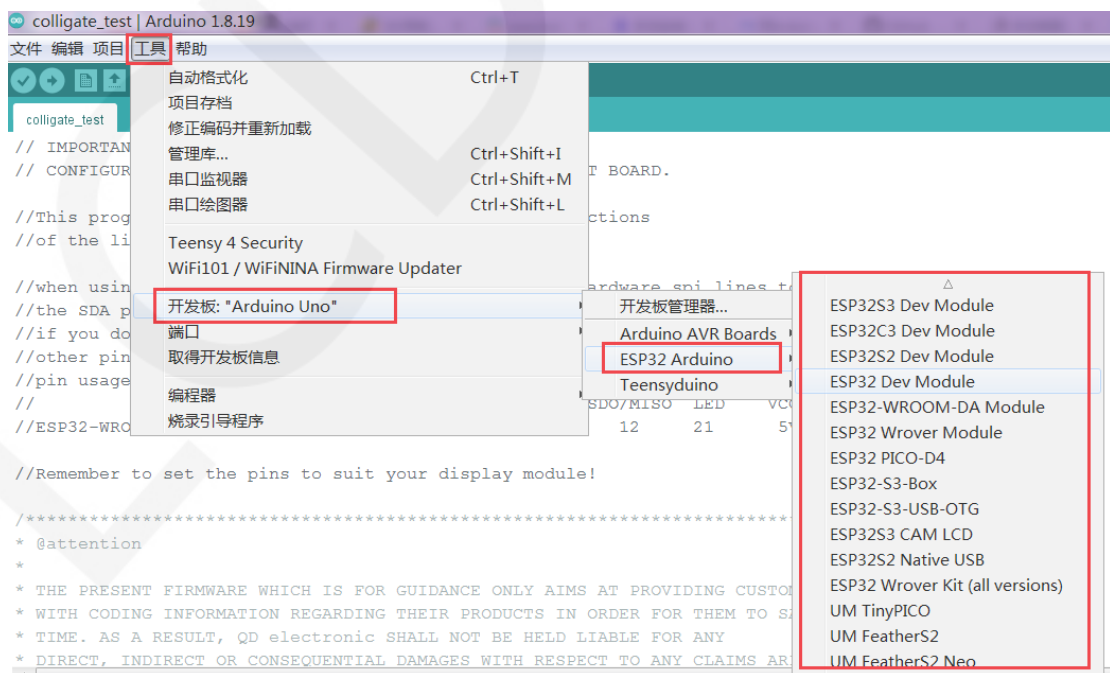
4、编译并下载程序

开发环境搭建好之后，需要编译并下载程序，来验证环境是否搭建正确。

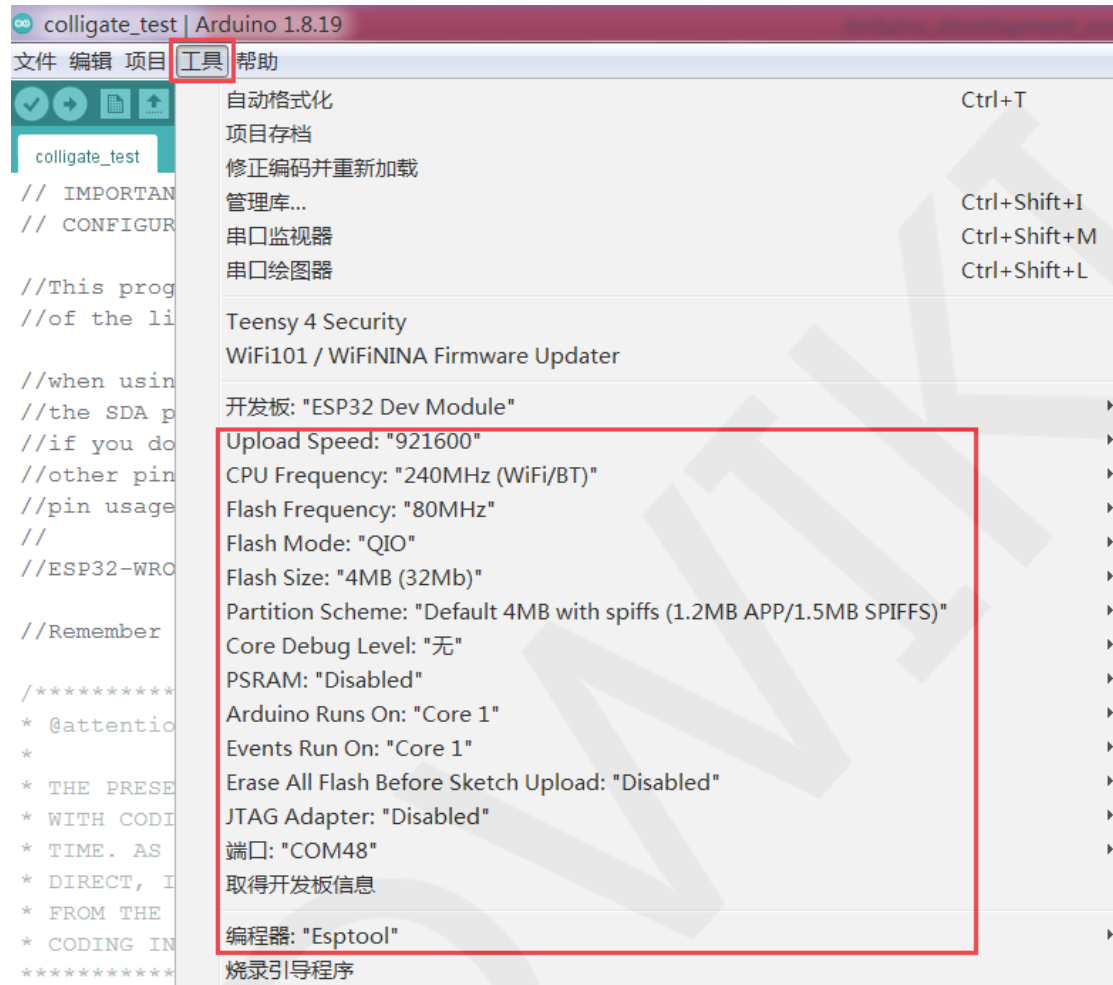
A、将 ESP32 开发板和 PC 机连接，给开发板上电。

B、打开 Arduino IDE 软件，单击工具按钮，然后选择开发板 → **ESP32 Arduino** →

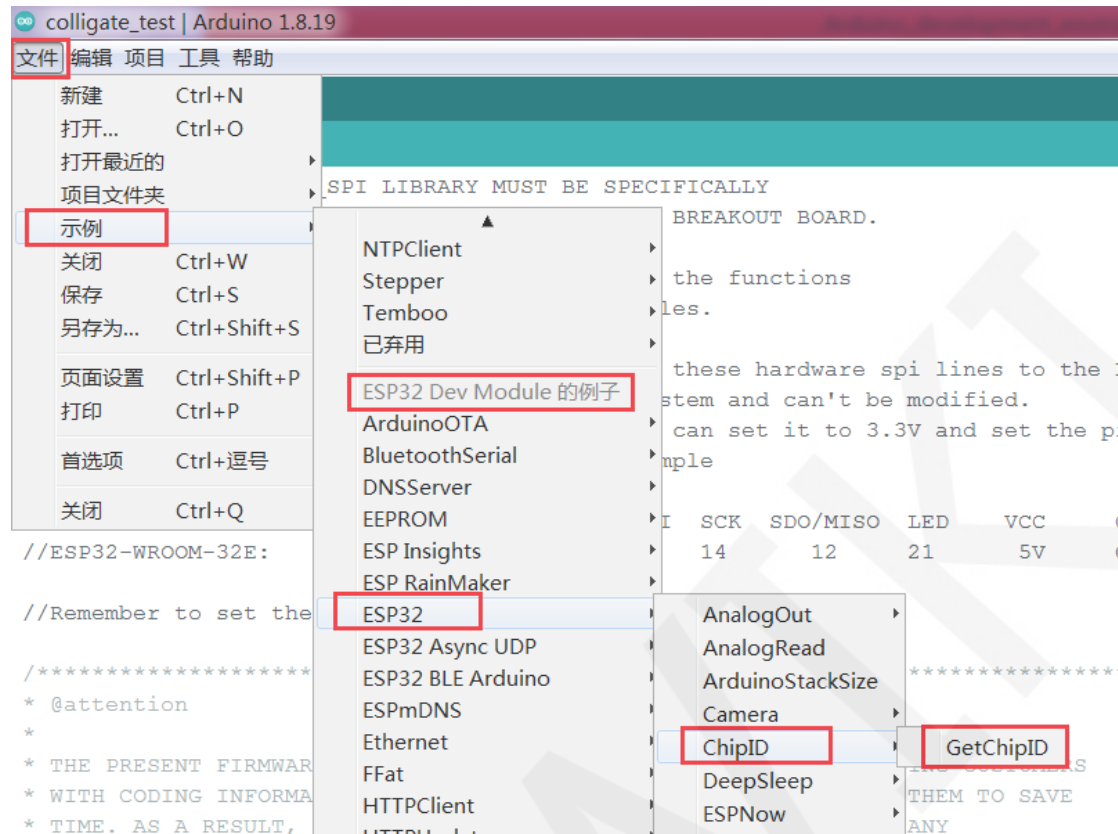
ESP32 Dev Module，如下图所示：



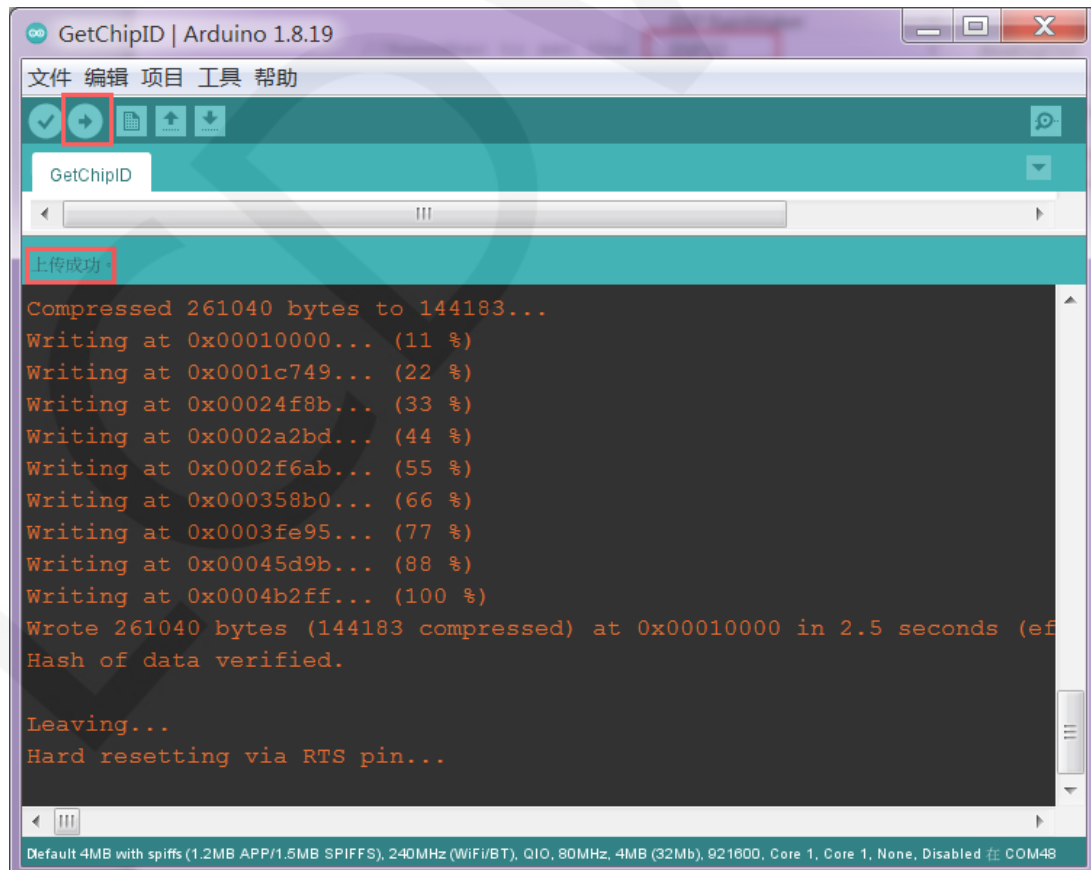
C、再点击**工具**按钮，可以看到开发板已经变成 ESP32，接下来根据自己所用的 ESP32 模组的配置，设置**下载速率**、**Flash**、**PSRAM**、**端口**等选项参数，如下图所示：



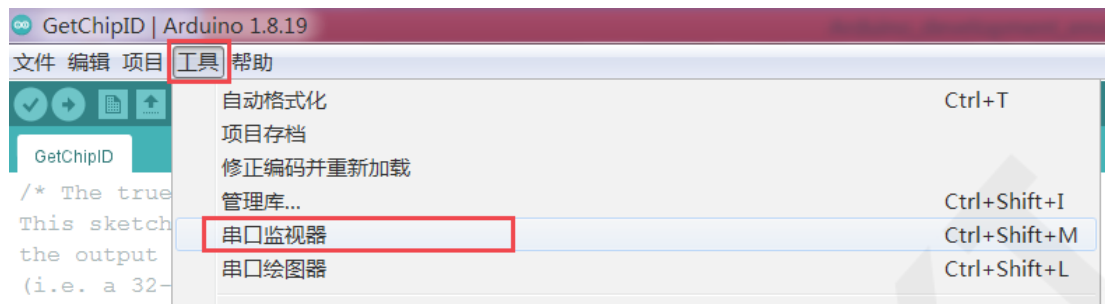
D、单击**文件**按钮，然后选择**示例**选项，再选择一个 ESP32 示例，如下图所示（这里选择 GetChipID 示例）：



E、单击上传按钮，此时会进行程序编译和烧录，待出现“上传成功”提示，则说明程序编译且烧录成功，如下图所示：



F、点击工具->串口监视器，就会出现串口信息界面，如下图所示：



串口信息界面有 Chip ID 等相关信息显示出来，就说明程序运行成功，环境搭建成功，如下图所示：

