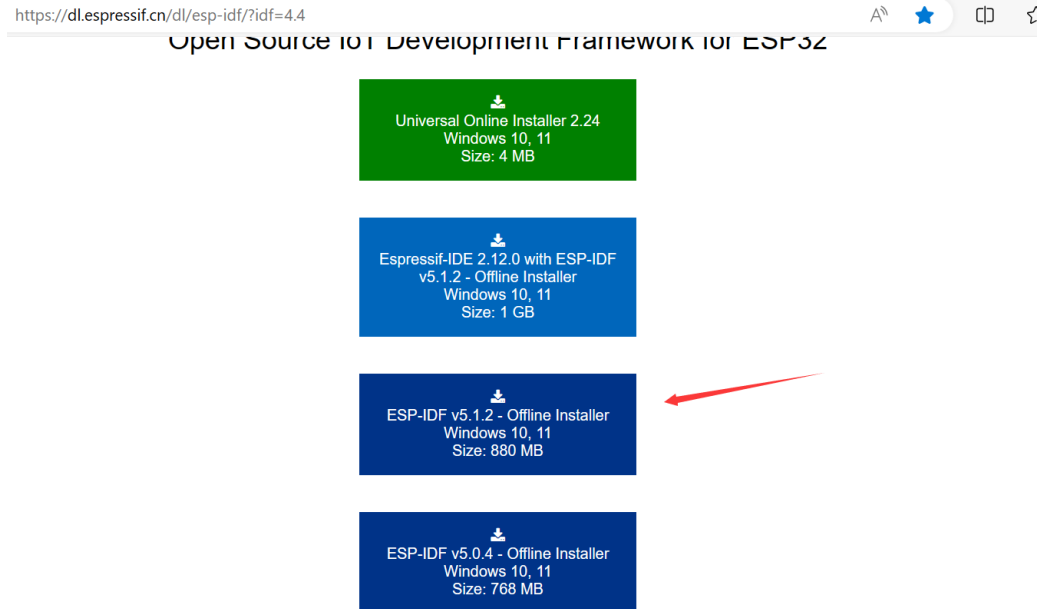


第一步：IDF5.1 下载

下载网址：<https://dl.espressif.com/dl/esp-idf/?idf=4.4> 打开网址后如下图：



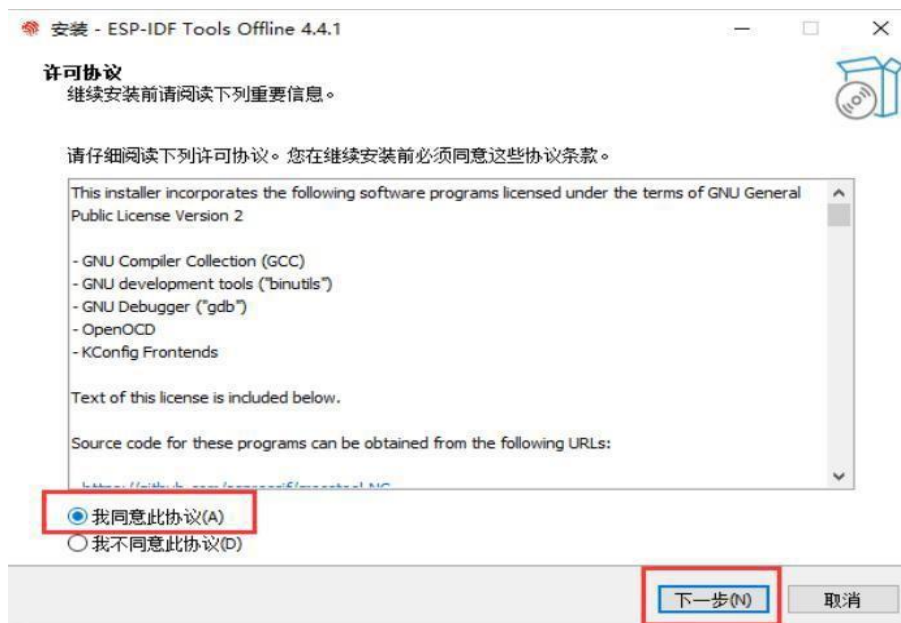
第二步：离线 idf5.1 安装

注意：安装前关闭 360 等杀毒软件

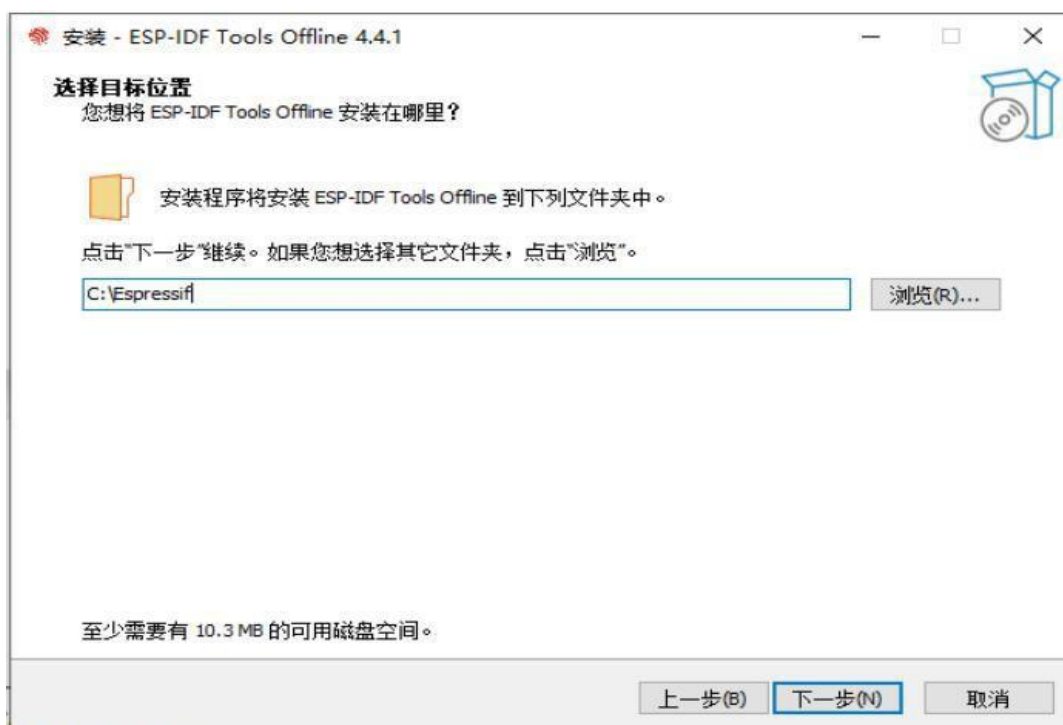
（一）下载好的离线的安装包，双击安装。



(三) 同意协议，然后选择下一步，其他的都是默认选项，选择下一步即可：



(四) 安装目录的选择，可以修改，然后下一步：

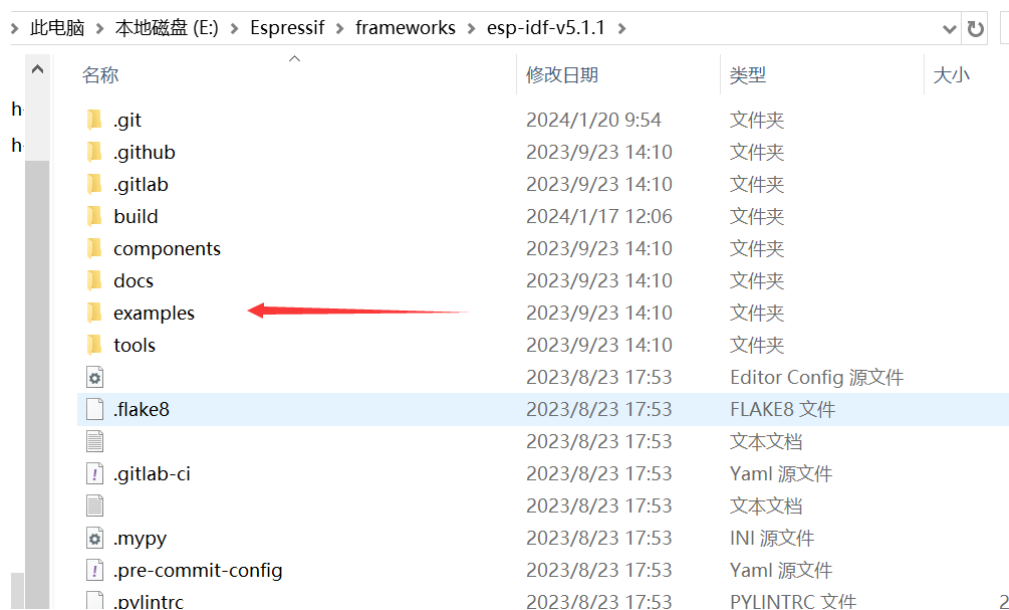


(五) 安装完成。

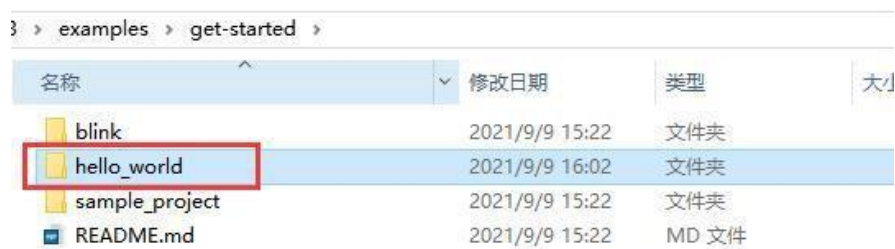


第三步：编译和使用（方法一）

经过第二步的安装，我们的 idf 开发环境已经安装好了，打开我们安装的目录：如下图：

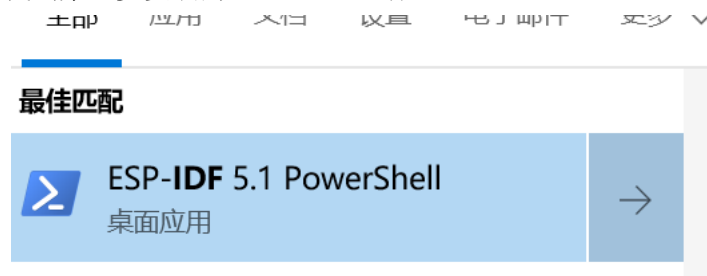


其中目录“[example](#)”就是官方提供的一些参考代码。我们下面尝试编译以下目录的“[hello_world](#)”工程：



注意：关 掉360 等杀毒软件。

(一) 在开始菜单找到第二步安装的 ESP-IDF，运行。



运行后，我们看到末尾的提示：[go to the project directory and run: idf.py build](#)，也就是 告诉我们，进入对应的工程目录，使用命令“[idf.py build](#)”就可以编译了

(二) 通过 cd 命令进入相应的例程

```
Checking if Python packages are up to date...
Python requirements from I:\ESP32\esp-idf-v4.3\requirements.txt are satisfied.

Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

    idf.py build

I:\ESP32\esp-idf-v4.3>cd .\examples\get-started\hello_world_ 进入编译的工程
```

(三) 设置工程支持的芯片

编译程序前需要设置工程支持对应的芯片（[ESP32](#)、[ESP32-C3](#)、[ESP32-S2](#) 和 [ESP32-S3](#)），通过命令“[idf.py set-target \(型号\)](#)”操作。配套提供的工程已经全部配置过芯片，这步可以不用操作。

```
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>idf.py set-target
Usage: idf.py set-target [OPTIONS] (esp32|esp32s2|esp32c3|esp32s3|linux)
Try 'idf.py set-target --help' for help.

Error: Missing argument '(esp32|esp32s2|esp32c3|esp32s3|linux)'. Choose from:
    esp32,
    esp32s2,
    esp32c3,
    esp32s3,
    linux

I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>idf.py set-target esp32
Adding 'set-target's dependency 'fullclean' to list of commands with default set of options.
Executing action: fullclean
```

不带参数表示查询，注意空格

IDF支持的芯片

设置工程支持ESP32芯片

ESP32 芯片配置命令: `idf.py set-target esp32`
ESP32 C3 芯片配置命令: `idf.py set-target esp32c3`
ESP32 S2 芯片配置命令: `idf.py set-target esp32s2`
ESP32 S3 芯片配置命令: `idf.py set-target esp32s3`

还可以通过命令“`idf.py menuconfig`”配置工程选项，如flash 大小，log 输出等级、波特率。

(四) 工程编译

```
Done! You can now compile ESP-IDF projects.
Go to the project directory and run:

idf.py build

I:\ESP32\esp-idf-v4.3>cd ..\examples\get-started\hello_world
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>idf.py build
Executing action: all (aliases: build)
Running ninja in directory i:\esp32\esp-idf-v4.3\examples\get-started\hello_world\build
Executing "ninja all"...
```

输入编译命令并回车，开始编译

```
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>idf.py build
Executing action: all (aliases: build)
Running ninja in directory i:\esp32\esp-idf-v4.3\examples\get-started\hello_world\build
Executing "ninja all"...
[1/46] Performing build step for 'bootloader'
ninja: no work to do.
[44/44] Generating binary image from built executable
esptool.py v3.1-dev
Merged 1 ELF section
Generated I:/ESP32/esp-idf-v4.3/examples/get-started/hello_world/build/hello-world.bin

Project build complete. To flash, run this command:
C:\Users\Administrator\espressif\python_env\idf4.3_py3.8_env\Scripts\python.exe ..\..\components\esptool_py\esptool\
esptool.py -p <PORT> -b 460800 --before default_reset --after hard_reset --chip esp32 write_flash --flash_mode dio --fl
ash_size detect --flash_freq 40m 0x1000 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition-table.bin
0x10000 build\hello-world.bin
or run 'idf.py -p <PORT> flash'
```

编译完成，并提示下载程序的命令

(五) 程序下载 编译完成后，已经告诉了我们下载的命令，它的完整的命令如下

:

`idf.py -p (PORT) [-b BAUD] flash`

使用串口 COM3 和波特率 115200 下载，命令如: `idf.py -p com3 -b 115200 flash`,注意空格。

(PORT):表示开发板生成的串口号，比如我们板子生成的是 com3.

[-b BAUD]:这个参数可有可无。如果有，表示指定波特率下载。

```
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>
I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world>idf.py -p com3 -b 460800 flash
Executing action: flash
Running ninja in directory I:\ESP32\esp-idf-v4.3\examples\get-started\hello_world\build
Executing "ninja flash"...
[1/4] Performing build step for 'bootloader'
ninja: no work to do.
[1/2] cmd.exe /C "cd /D I:\ESP32\esp-idf-v4.3\components\esptool.py && C:\Users\Administrator\espressif\tools\cmake\3.16.4\bin\cmake.exe -D IDF_PATH="I:\ESP32\
esptool.py esp32 -p com3 -b 460800 --before-default_reset --after-hard_reset write_flash --flash_mode dio --flash_freq 40n --flash_size 2MB 0x8000 partition_t
esptool.py v3.1-dev
Serial port com3
Connecting....
Chip is ESP32-D0WD (revision 1)
Features: WiFi, BT, Dual Core, 240MHz, VRef calibration in efuse, Coding Scheme None
Crystal is 40MHz
MAC: a4:c5:7c:49:f5:78
Uploading stub...
Running stub...
Stub running...
Changing baud rate to 460800
Changed.
Configuring flash size...
Flash will be erased from 0x00008000 to 0x00008fff...
Flash will be erased from 0x00001000 to 0x00007fff...
Flash will be erased from 0x00010000 to 0x00037fff...
Compressed 3072 bytes to 103...
Writing at 0x00008000... (100 %)
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.0 seconds (effective 527.1 kbit/s)...
Hash of data verified.
Compressed 25008 bytes to 15381...
Writing at 0x00001000... (100 %)
Wrote 25008 bytes (15381 compressed) at 0x00001000 in 0.8 seconds (effective 246.7 kbit/s)...
Hash of data verified.
Compressed 161200 bytes to 84529...
Writing at 0x00010000... (16 %)
Writing at 0x0001aa3e... (33 %)
Writing at 0x000201e2... (50 %)
Writing at 0x00025abe... (66 %)
Writing at 0x0002f113... (83 %)
Writing at 0x000364e7... (100 %)
Wrote 161200 bytes (84529 compressed) at 0x00010000 in 2.2 seconds (effective 592.3 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done
```

下载命令，串口是com3,波特率是460800，注意加空格

有可能需要通过按键进入下载过程

下载完成

(六) 其他控制台命令说明

idf.py menuconfig //进入配置菜单

idf.py -p PORT [-b BAUD] flash //可以设置波特率的下载命令

idf.py -p COM3 -b 460800 flash //通过串口 3 下载程序

idf.py -p COM3 monitor //打开串口 com3 监视

idf.py -p PORT flash monitor //程序下载完成后，打开串口监视

idf.py build //编译

idf.py fullclean //清除

idf.py set-target esp32 //设置支持的芯片，

第四步：使用 vscode 编译（方法二）

注意：建议关闭 360 之类的杀毒软件，否则会出现莫名其妙的错误。

vscode 编译，必须在第二步的离线安装 IDF 基础上完成。

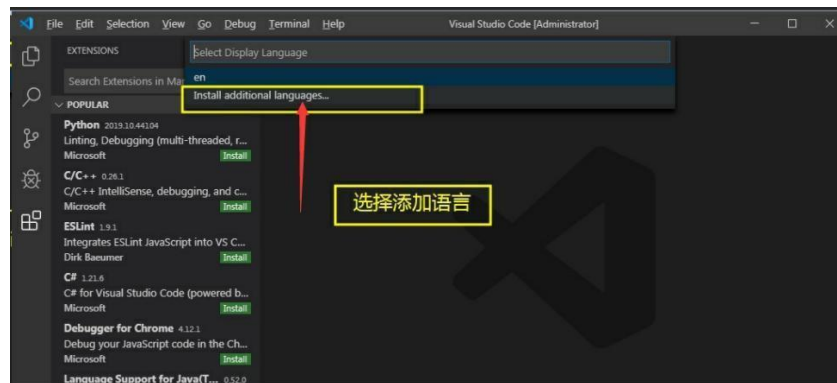
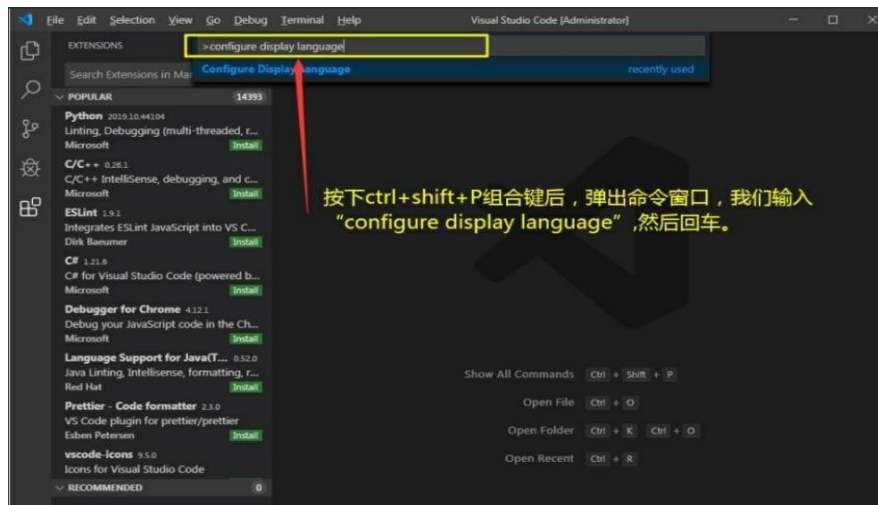
(一) 需要先安装 Visual Studio Code 软件

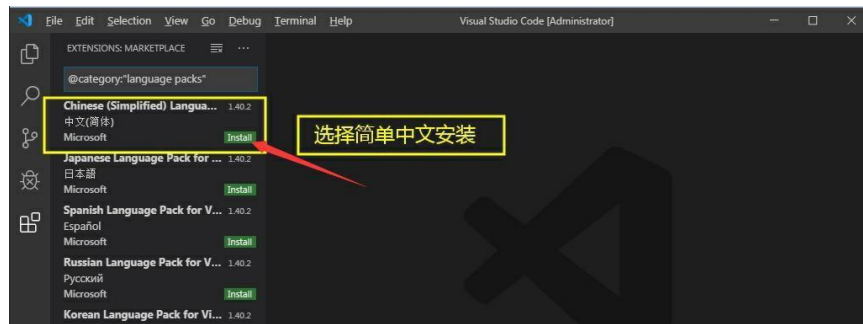
注意：如果已经安装，可以跳到下一步。 下载网址：

<https://code.visualstudio.com/Download>

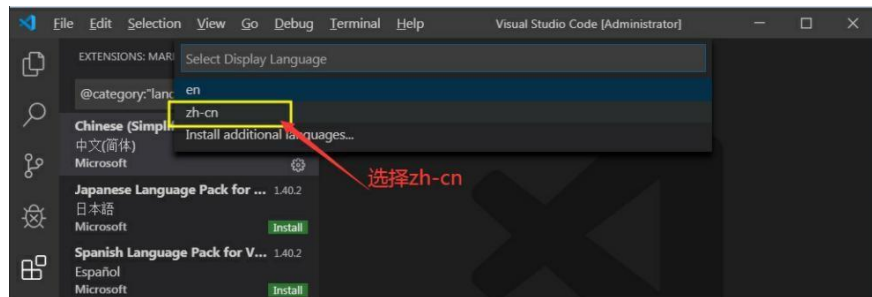
管理员运行后，接受许可后，一路点下一步，直到安装完成即可。 可以考虑按以下方法汉化。

启动 Visual Studio Code 软件，按下组合键“**ctrl+shift+p**”，输入命令“**configure display language**”，回车确认。





再一次按下组合键“**ctrl+shift+p**”，输入命令“**configuredisplay language**”，回车确认，选择“zh-cn”，然后重启vscode 软件即可。

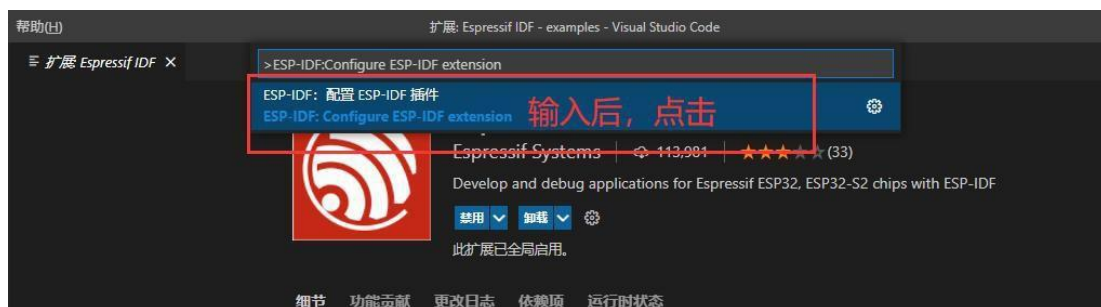


（二）安装 esp-idf 插件

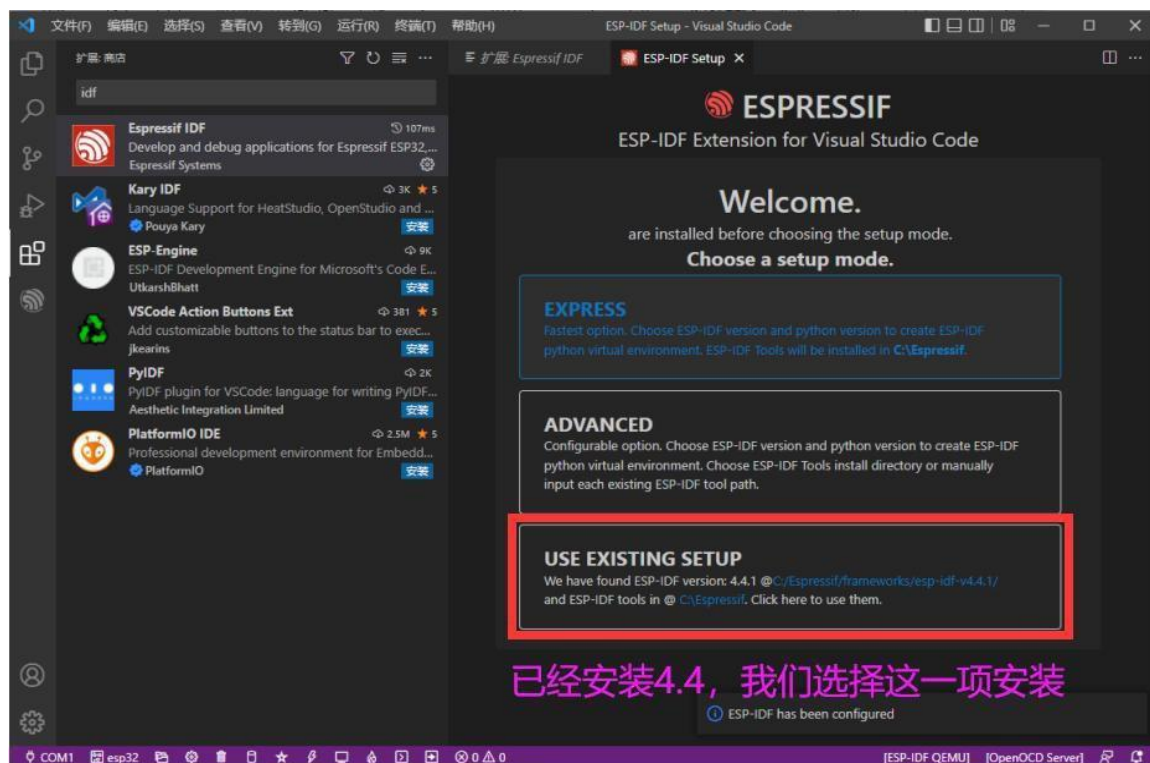
在扩展搜索idf，即可看到 Espressif IDF，点击安装



IDF 安装完成后，按下F1 键，输入“ESP-IDF:Configure ESP-IDF extension”，打开配置页面。



在以下界面，选择“ADVANCED”或者“USE EXISTINGSETUP”都可以，我们选择后者。



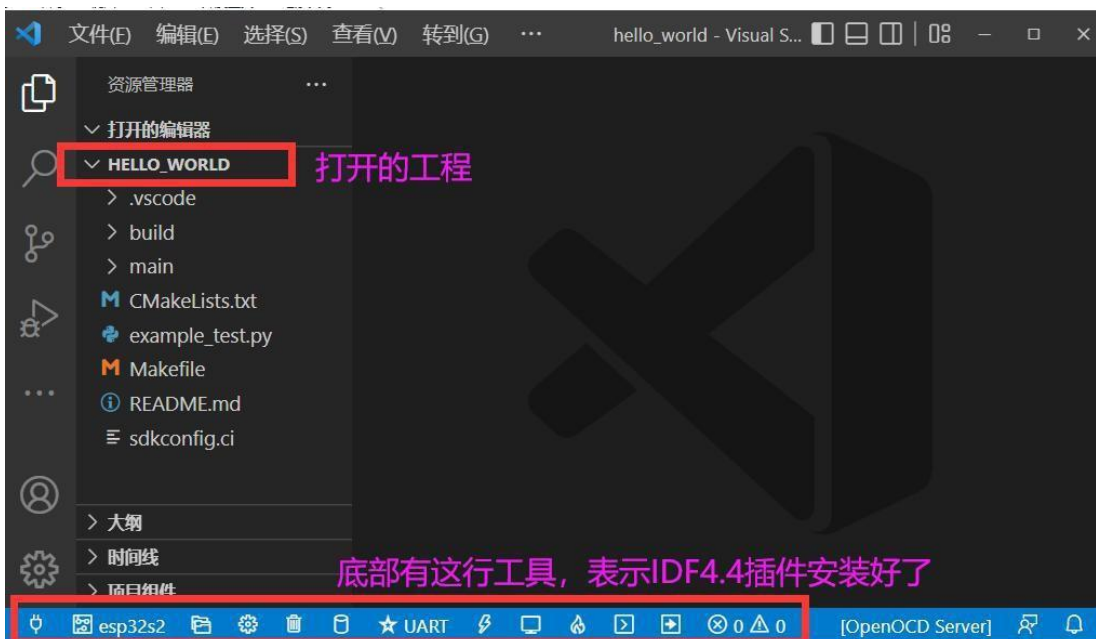
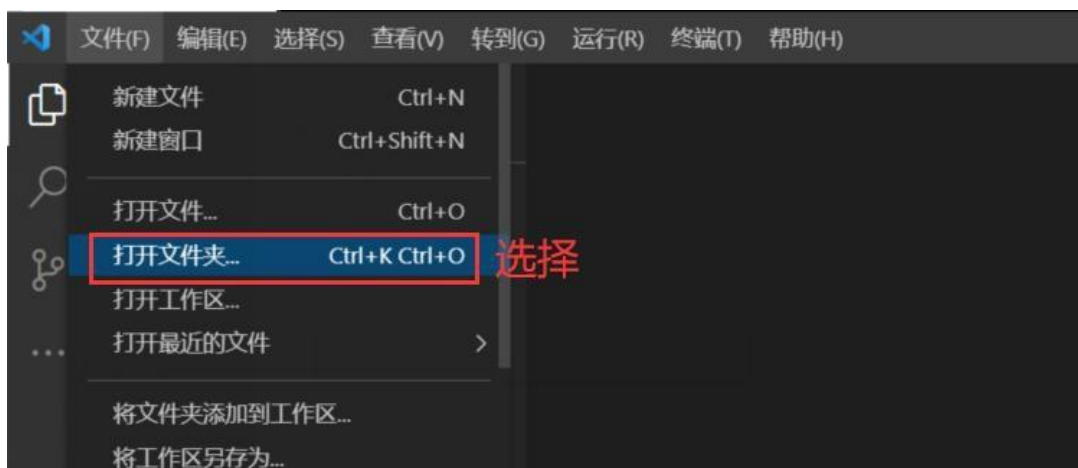
安装完成:



（三）打开工程和编译

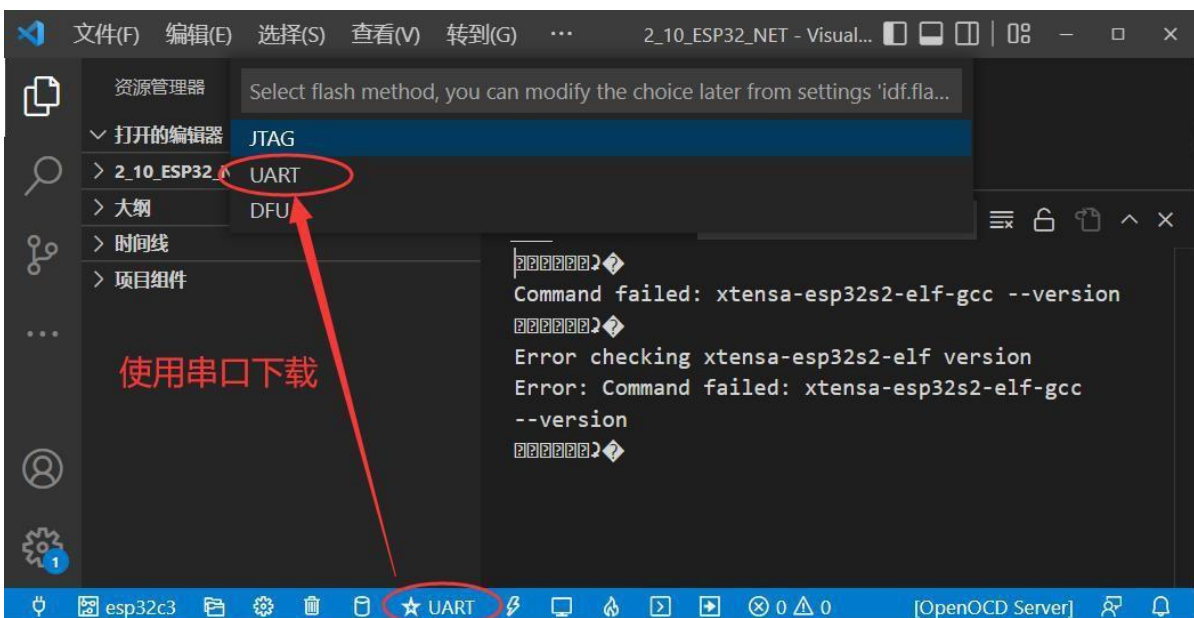
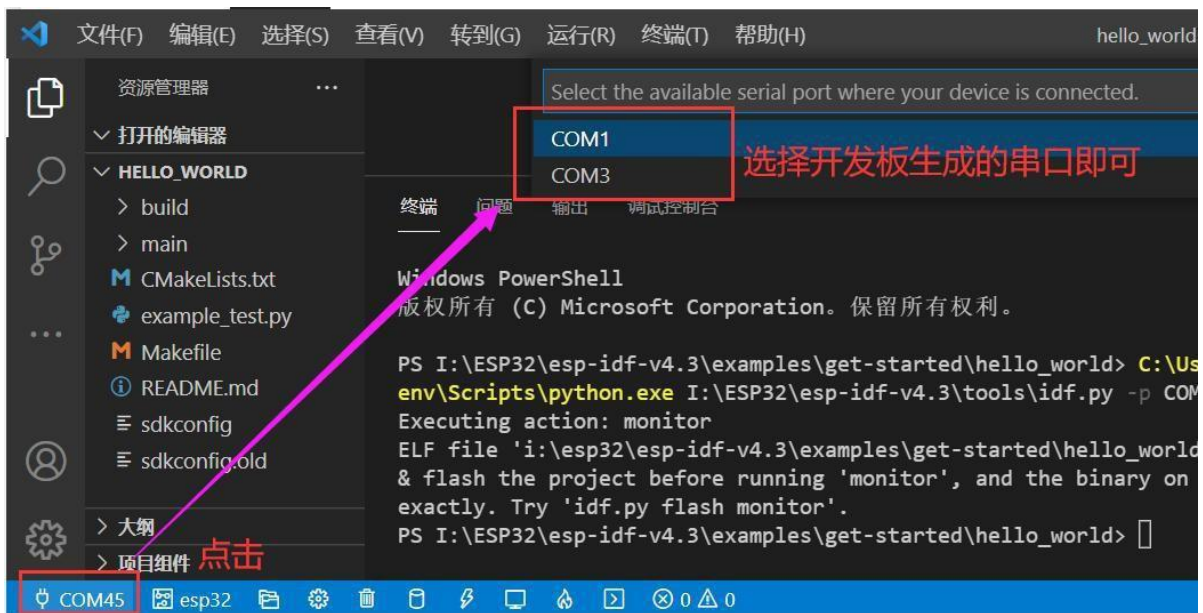
注意：必须关闭 360 之类的杀毒软件，否则会出现各种未知问题。

VSCODE 安装完成 esp-idf 插件后，我们打开一个工程：

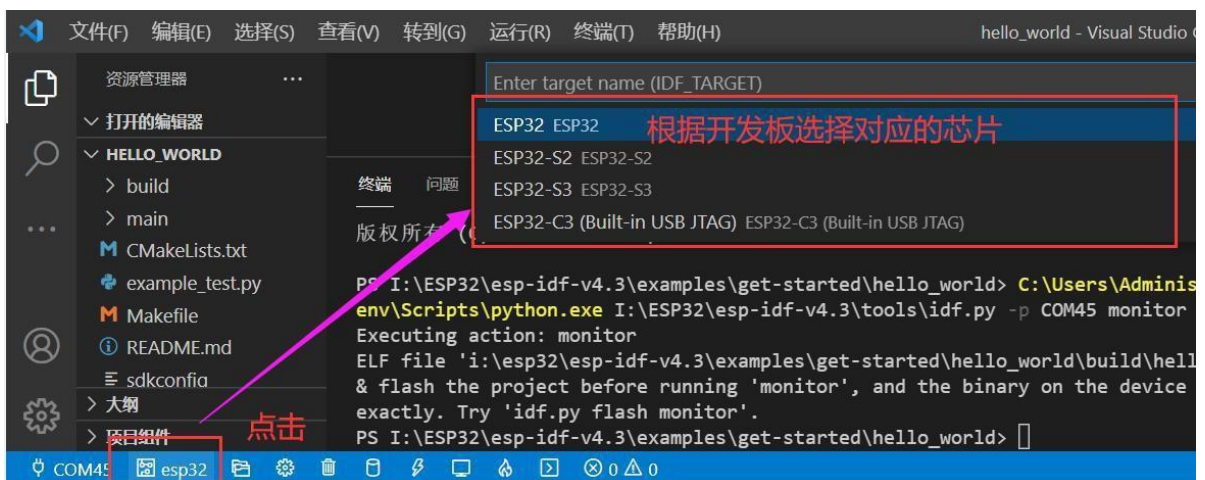


界面底部工具，把鼠标放到对应的图标上，会有提示，说明如下：

① 串口选择

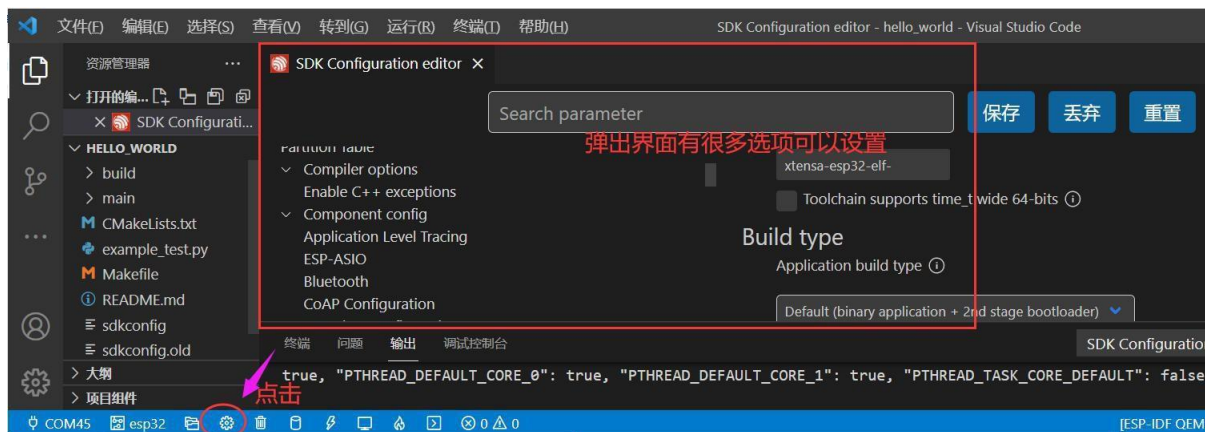


② 芯片选择

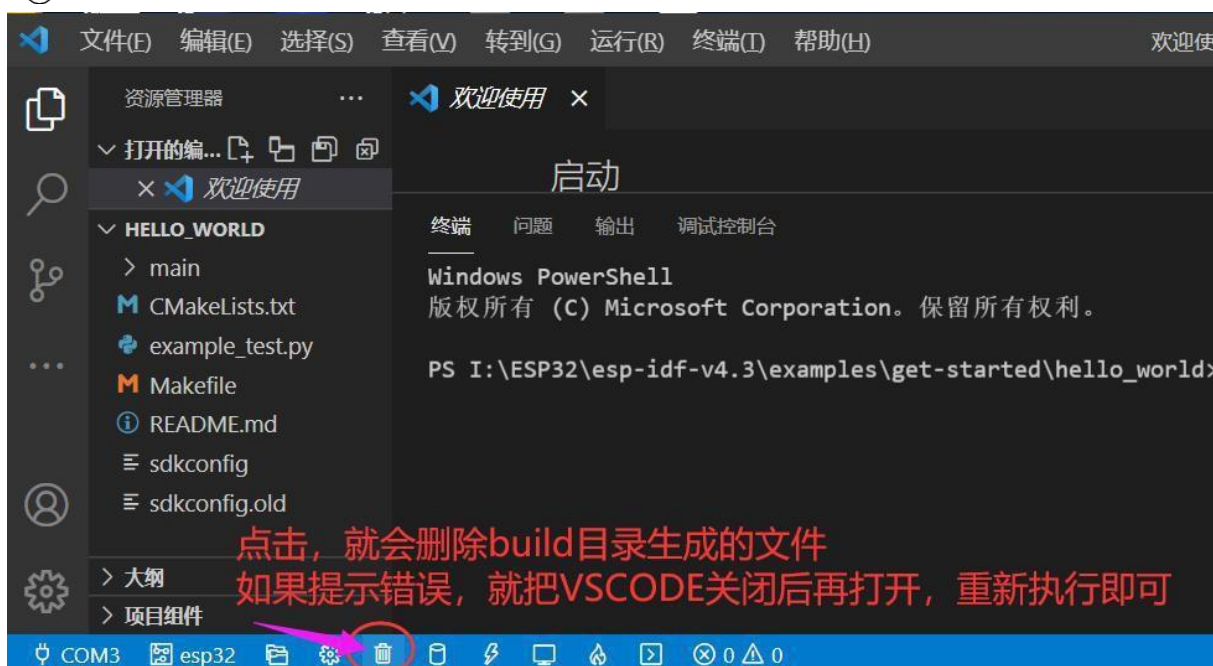


③ 工程配置

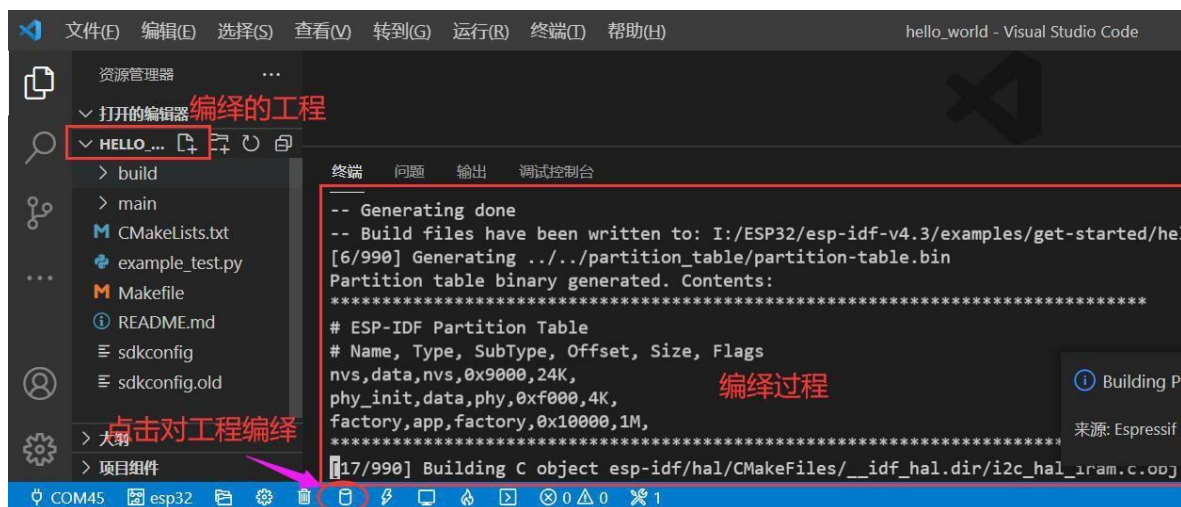
可以通过菜单配置 CPU 频率, flash 大小, LOG 输出等级, LOG 输出波特率等。



④ 清除编译结果



⑤ 编译程序



```
终端 问题 输出 调试控制台
[85/86] Linking C executable bootloader.elf
[86/86] Generating binary image from built executable
esptool.py v3.1-dev
Merged 1 ELF section
Generated I:/ESP32/esp-idf-v4.3/examples/get-started/hello_world/build/bootloader/bootloader.bin
[990/990] Generating binary image from built executable
esptool.py v3.1-dev
Merged 1 ELF section
Generated I:/ESP32/esp-idf-v4.3/examples/get-started/hello_world/build/hello-world.bin

终端将被任务重用，按任意键关闭。 编译完成
```

⑥ 下载程序

文件(F) 编辑(E) 选择(S) 查看(V) 转到(G) 运行(R) 终端(T) 帮助(H) hello_world - Visual Studio Code

资源管理器 ...

- 打开的编辑器
- HELLO_WORLD
 - build
 - main
 - CMakeLists.txt
 - example_test.py
 - Makefile
 - README.md
 - sdkconfig
 - sdkconfig.old
- 大纲
- 项目组件

终端 问题 输出 调试控制台 提示：可能需要通过按键，进入下载过程

```
Compressed 25024 bytes to 15385...
Wrote 25024 bytes (15385 compressed) at 0x00001000 in 0.7 seconds (effective 297.2 kbit/s)...
Hash of data verified.
Compressed 3072 bytes to 103... 下载过程
Wrote 3072 bytes (103 compressed) at 0x00008000 in 0.1 seconds (effective 423.5 kbit/s)...
Hash of data verified.

Leaving...
Hard resetting via RTS pin... 下载完成

终端将被任务重用，按任意键关闭。
```

COM3 esp32 下载