



# ESP32系列教程

## 环境搭建\_Windows

Wireless-Tag



版本1.0.0

启明云端

版权所有©2023



## 目录

前言 .....	3
1. 安装git .....	3
2. 安装Python .....	8
3. 获取ESP-IDF .....	11

Wireless-tag



## 前言

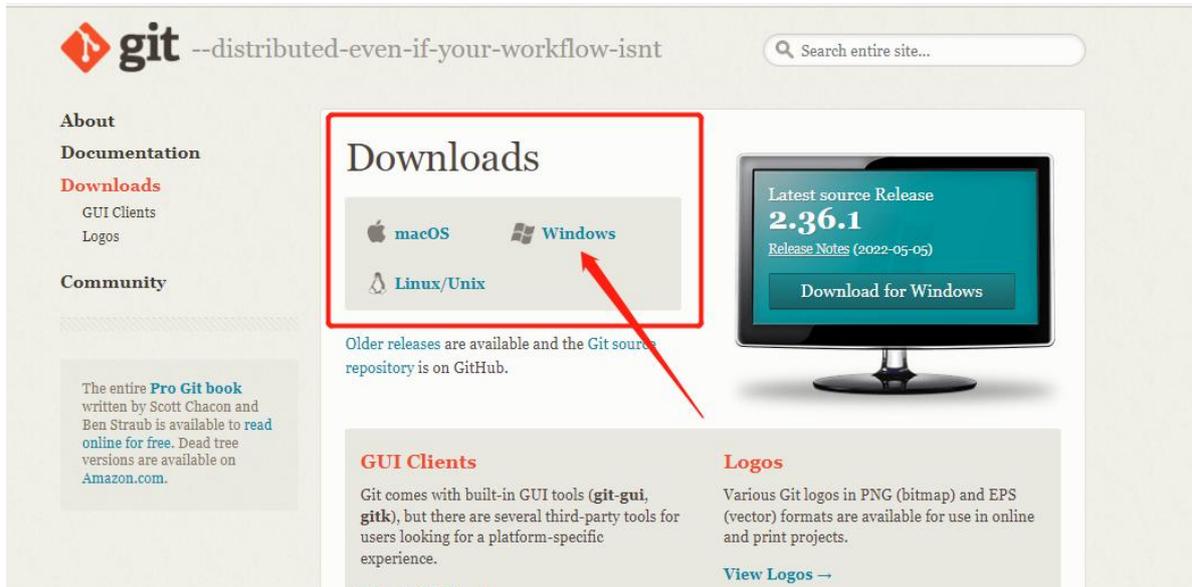
本文档介绍了在Windows下的ESP-IDF开发环境搭建的步骤，并通过一个简单的示例来演示项目编译、烧录和运行的过程。

Wireless-tag

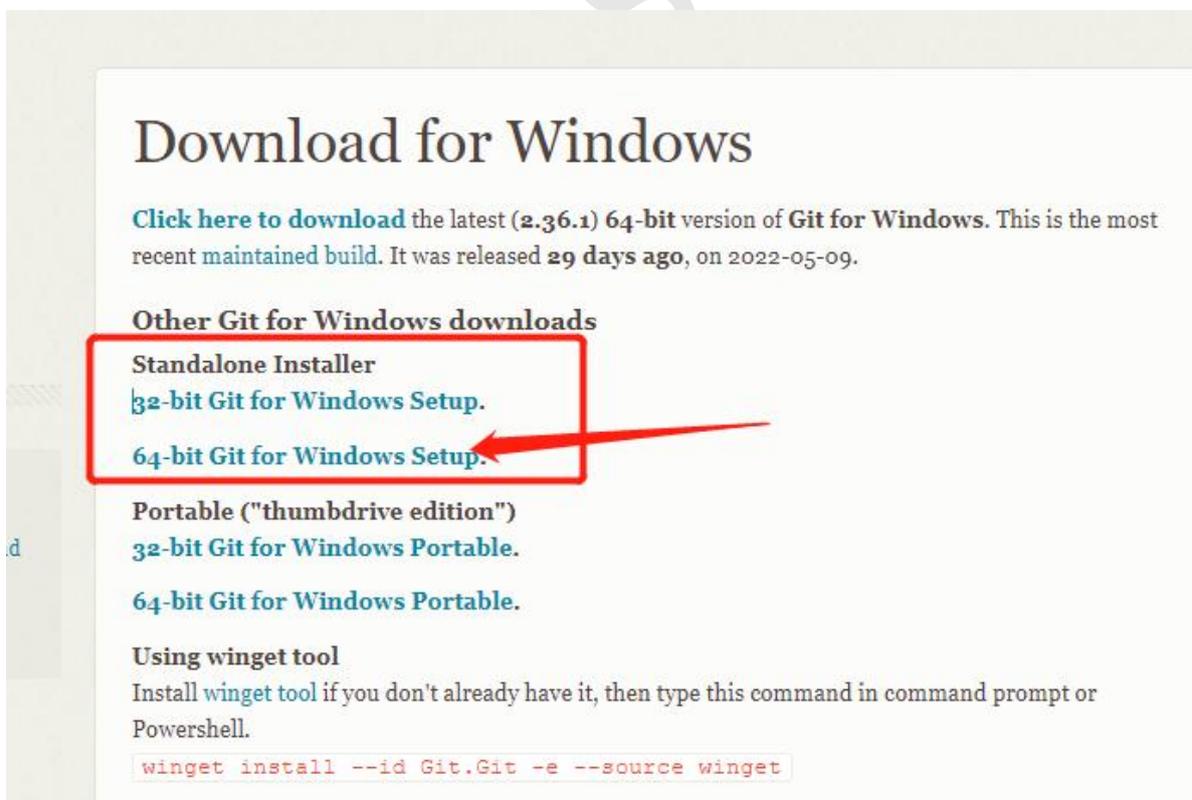
## 1. 安装git

在安装ESP\_IDF环境前，需要先安装git工具，打开下面的链接，下载适合的git。

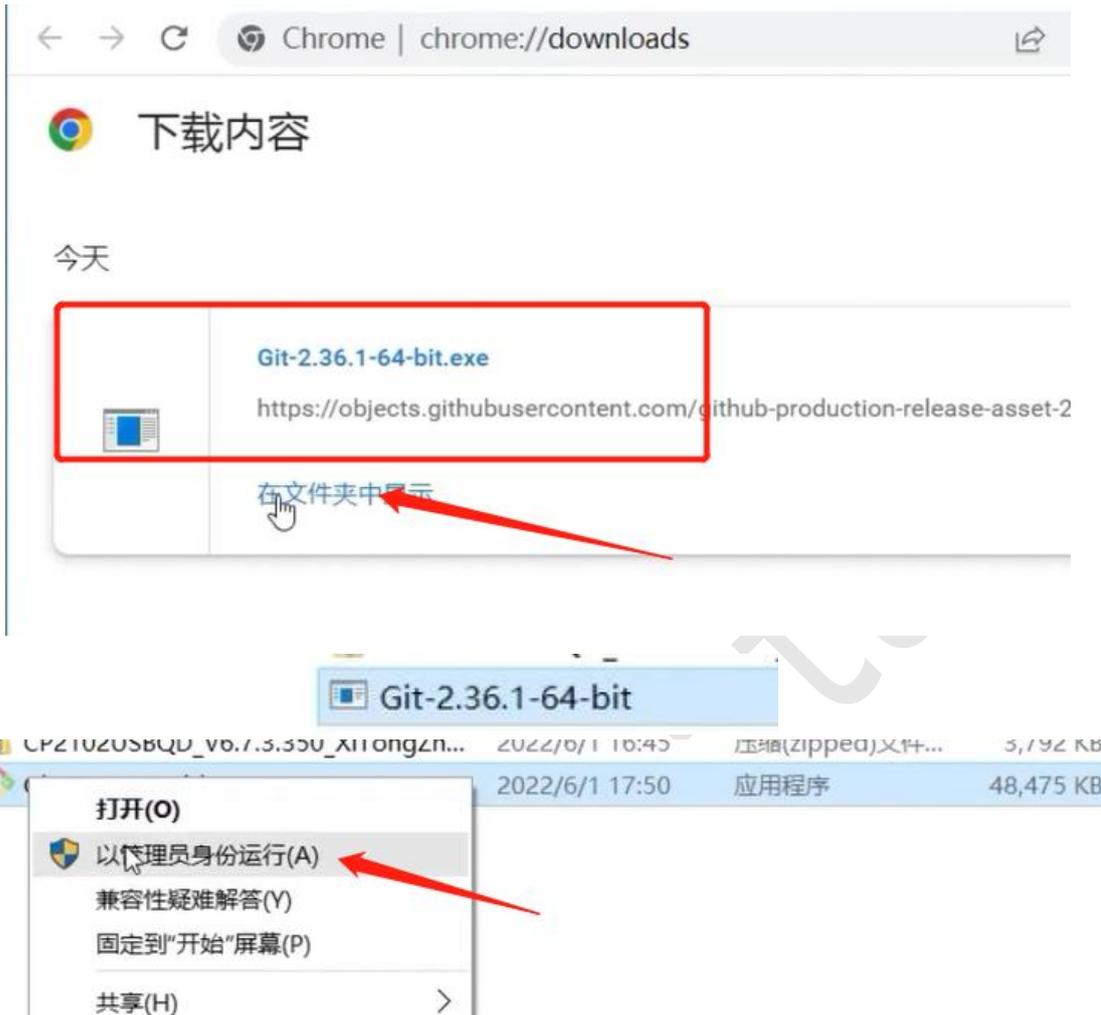
git下载地址：<https://git-scm.com/downloads>



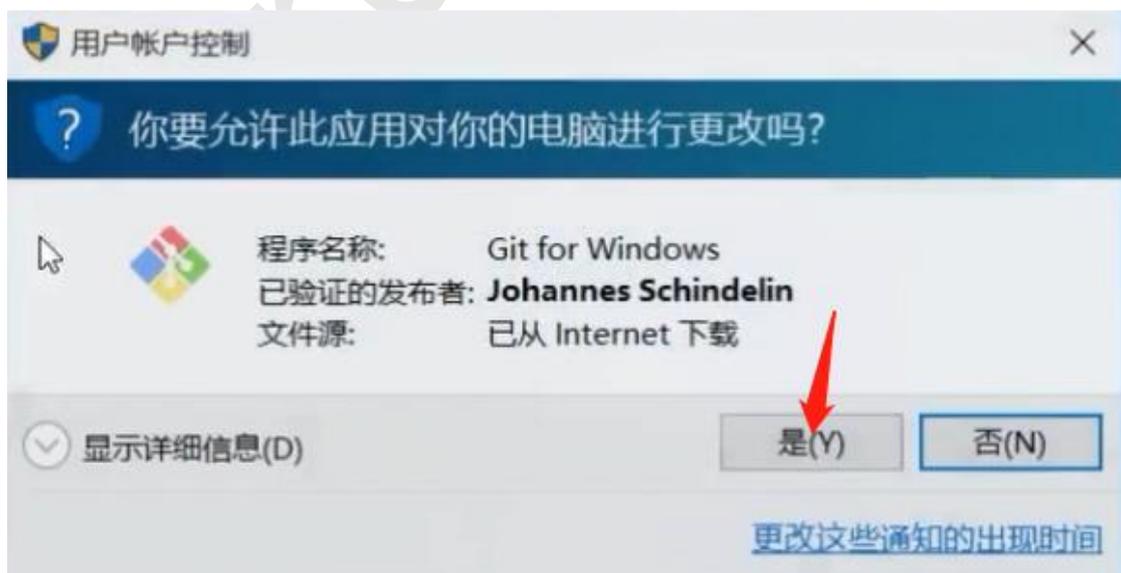
在红框内选择git的安装系统，上面有三个操作系统的选择，本次教程使用的是Windows。



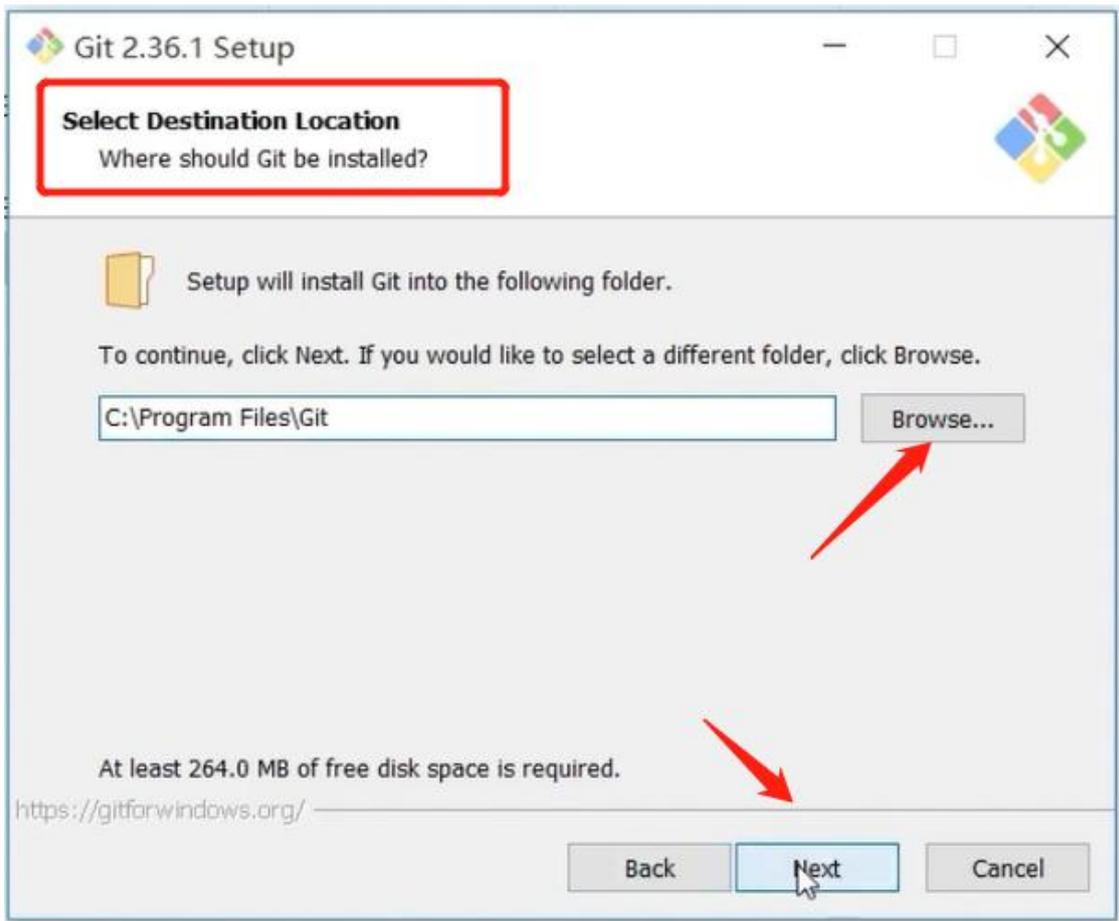
在这里，选择“Standalone Installer”独立安装程序，64位。



下载好后，点击“在文件夹中显示”，在弹出的文件夹中找到“git安装程序”，选择“以管理员身份运行”。



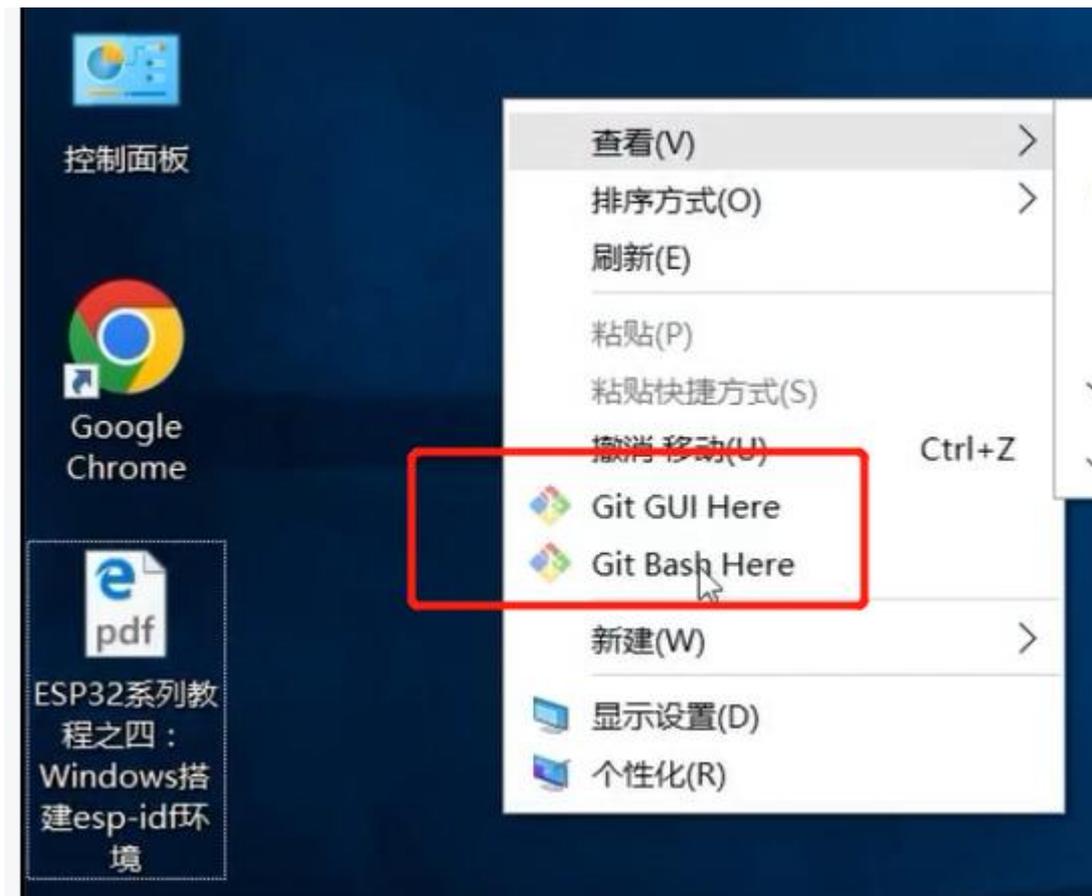
选择“是”，一直点击“next”，直到“安装地址”的选择。



这里是安装路径的设置，点击“Browse”选择想要安装git的地方（也可以用默认地址），然后点击“next”下一步。



一直next，直到安装完成，点击“finish”完成。



回到桌面，点击鼠标右键，看是否有红框内的选项，有即为安装成功。

## 2. 安装Python

点击下方的连接下载python

下载链接: <https://www.python.org/downloads>

寻找特定版本？  
Python 按版本号发布：

发布版本	发布日期		点击了解更多
蟒蛇 3.9.13	2022年5月17日	下载	发行说明
Python 3.10.4	2022年3月24日	下载	发行说明
蟒蛇 3.9.12	2022年3月23日	下载	发行说明
Python 3.10.3	2022年3月16日	下载	发行说明
蟒蛇 3.9.11	2022年3月16日	下载	发行说明
蟒蛇 3.8.13	2022年3月16日	下载	发行说明
蟒蛇 3.7.13	2022年3月16日	下载	发行说明

查看旧版本

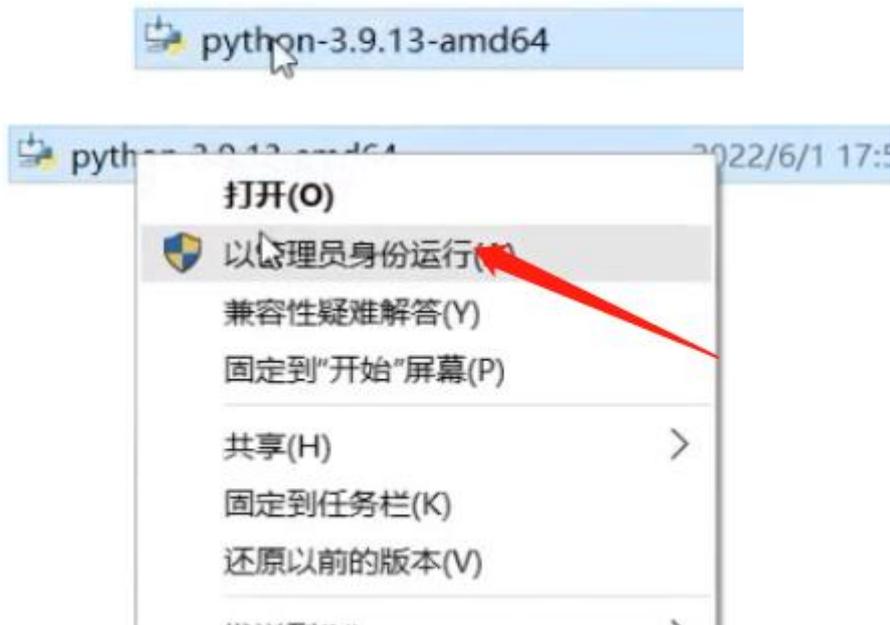
直接选择最新版，点击“下载”。

https://www.python.org/downloads/release/python-3913/

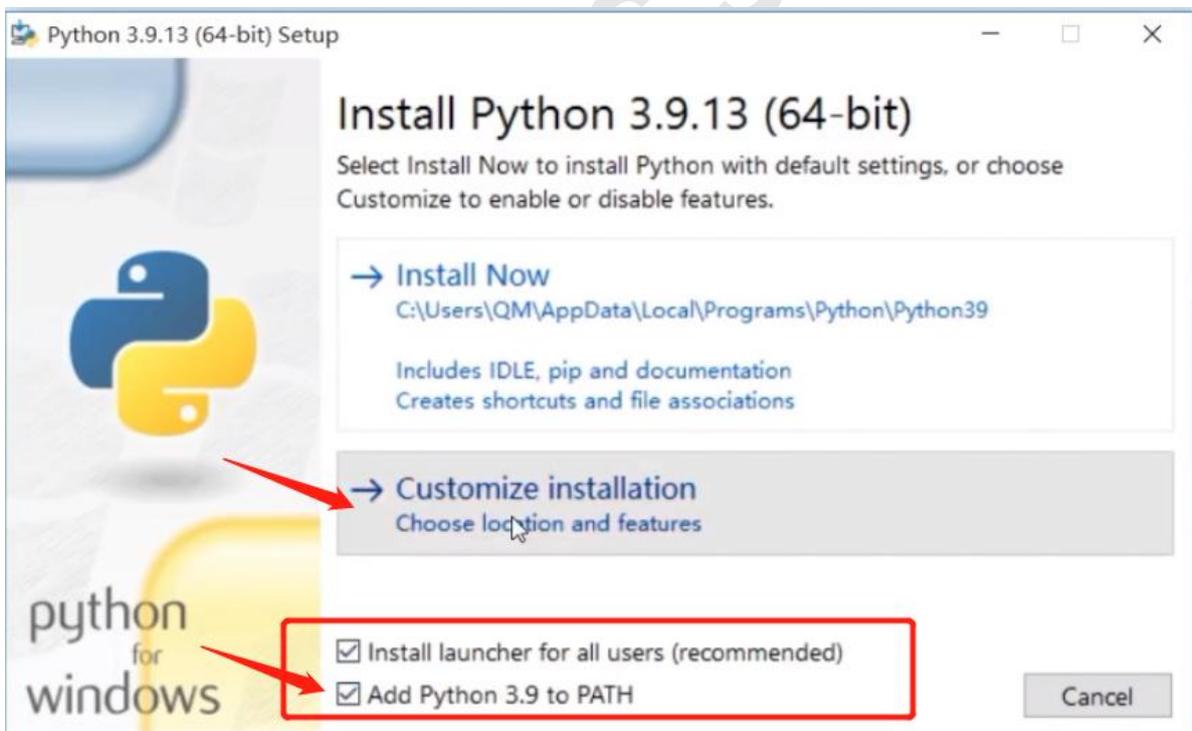
### Files

Version	Operating System	Description	MDS Sum	File Size	GPG
<a href="#">Gzipped source tarball</a>	Source release		eafda83543bad127cade4d288fdab87	26355887	SIG
<a href="#">XZ compressed source tarball</a>	Source release		5e2411217b0060828d5f923eb422a3b8	19754368	SIG
<a href="#">macOS 64-bit Intel-only installer</a>	macOS	for macOS 10.9 and later, deprecated	671848930809decf27f586ddf98c6e9b	30997161	SIG
<a href="#">macOS 64-bit universal2 installer</a>	macOS	for macOS 10.9 and later	76b63cf623e32cdf27c5033434bd69ce	38821163	SIG
<a href="#">Windows embeddable package (32-bit)</a>	Windows		fec0bc06857502a56dd1aaeea6488ef8	7729405	SIG
<a href="#">Windows embeddable package (64-bit)</a>	Windows		57731cf80b1c429a0be7133266d7d7cf	8570740	SIG
<a href="#">Windows help file</a>	Windows		c86feba059b340a1de2a9d2ee7059a6d	8953644	SIG
<a href="#">Windows installer (32-bit)</a>	Windows		46c35b0a2a4325c275b2ed3187b08ac4	28096840	SIG
<a href="#">Windows installer (64-bit)</a>	Windows	Recommended	e7062b85c3624af82079794729618eca	29235432	SIG

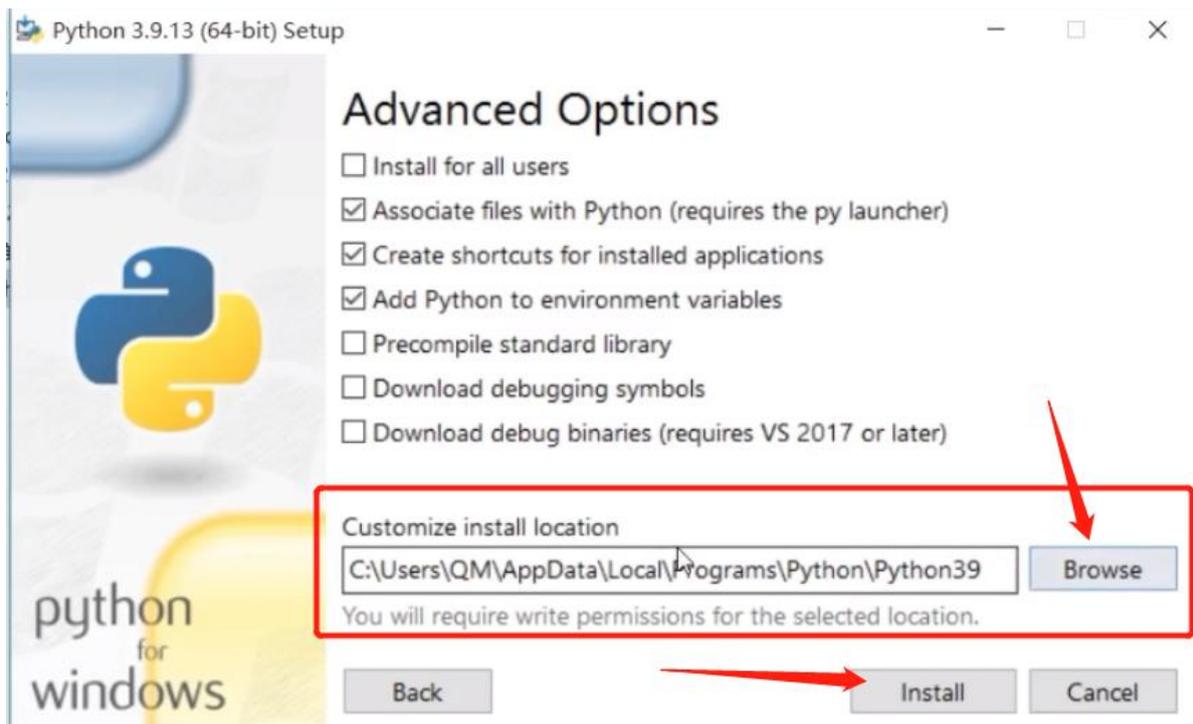
找到红框箭头处——windows64位的安装包，点击即可下载python安装包。



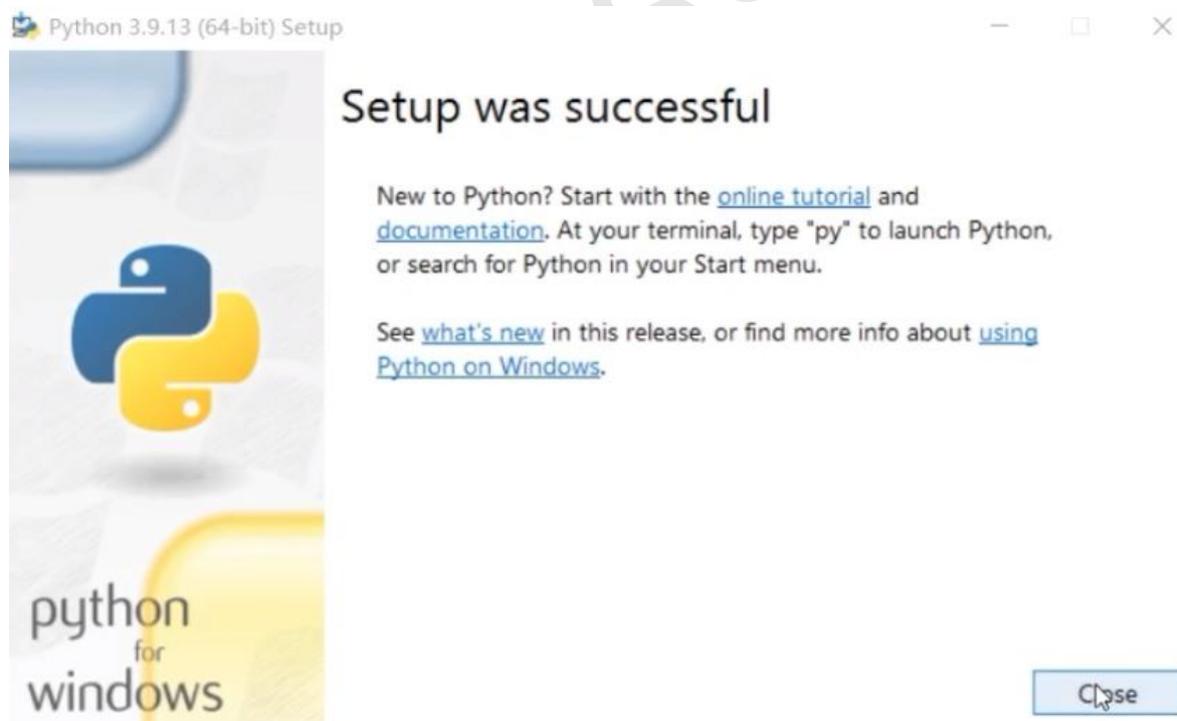
下载好后，选择python安装包，鼠标右键点击，选择“以管理员身份运行”，即可开始安装python，下面设置一些小配置。



打开安装包，下面红框中的两个都要勾选上，默认只勾选其中一个。然后点击“Customize installation”自定义安装，一直“next”，直到安装路径的选择。



在这个界面选择安装python的路径（也可以用默认地址），点击“Browse”浏览选择路径，然后点击“install”开始下载。



显示这个界面就表示安装完成。

### 3. 获取ESP-IDF

Windows上获取ESP-IDF的方式有几种

方法1: 打开git, 在git里面将esp-idf克隆到你的目录下进行安装

方法2: 到乐鑫的官网下载esp-idf压缩包, 建议解压后改名为esp-idf

方法2下载地址: [SDK和演示|乐鑫科技\(espresif.com.cn\)](http://SDK和演示|乐鑫科技(espresif.com.cn))



方法3: 使用ESP-IDF工具安装器安装

ESP-IDF工具安装器下载地址: <https://dl.espressif.cn/dl/esp-idf/>

**本次教程使用方法3来演示**

打开ESP-IDF工具安装器下载地址。

## ESP-IDF Windows Installer Download

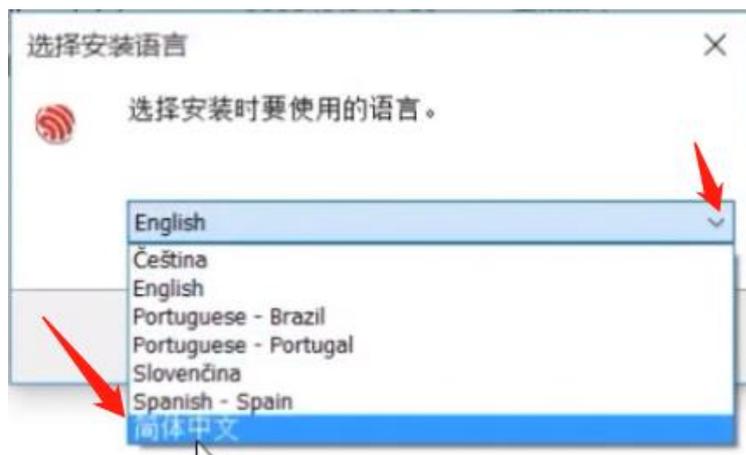
面向 ESP32 的开源物联网开发框架



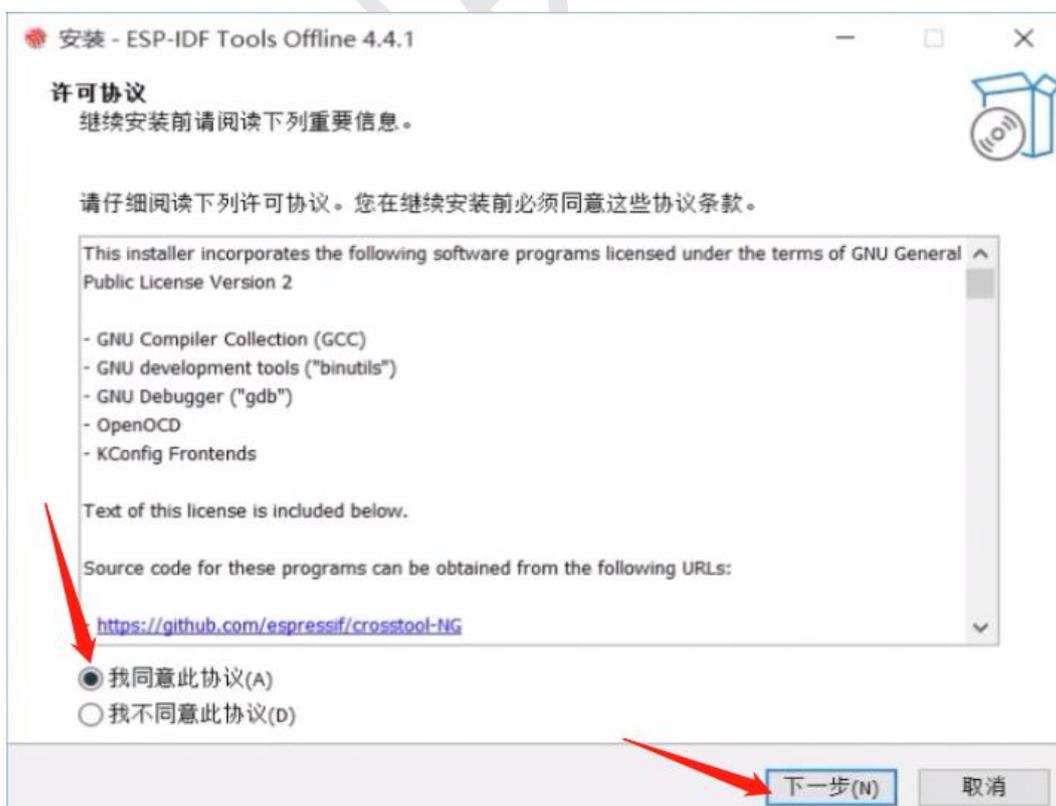
这里选择ESP-IDF v4.4.1-脱机安装程序，点击即开始下载。



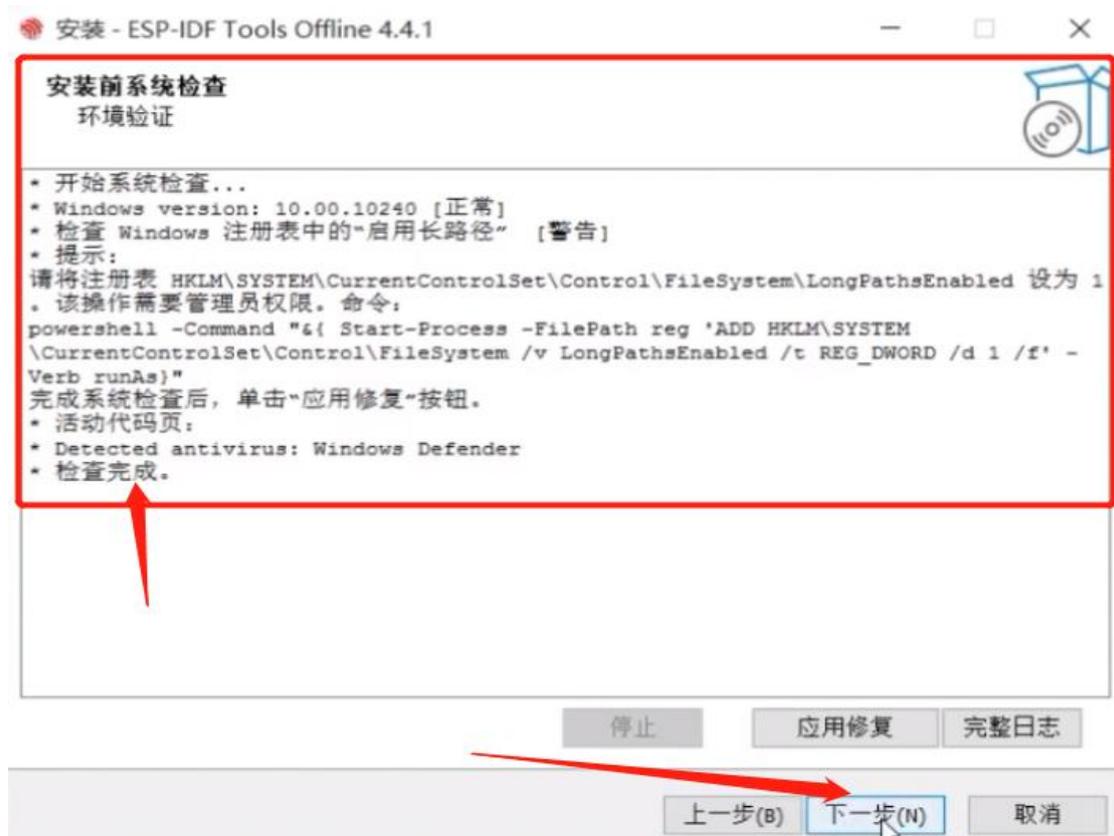
下载完成后，鼠标右击，选择“以管理员身份运行”。



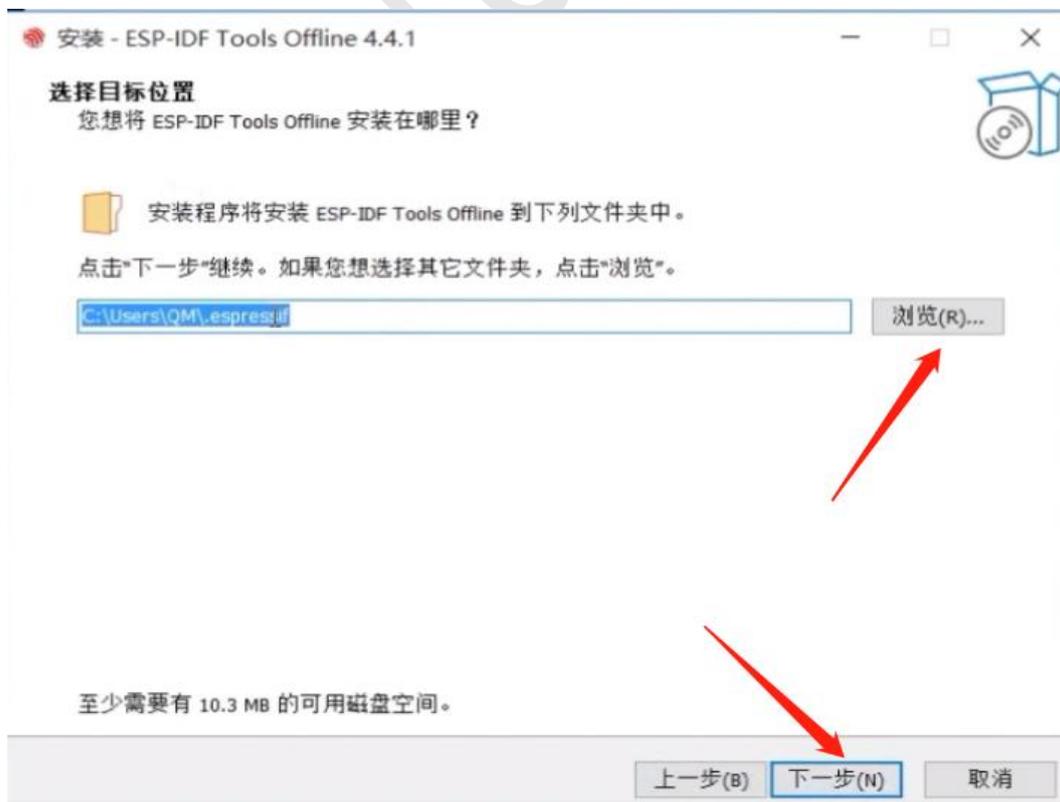
选择合适的语言，然后点击“确定”。



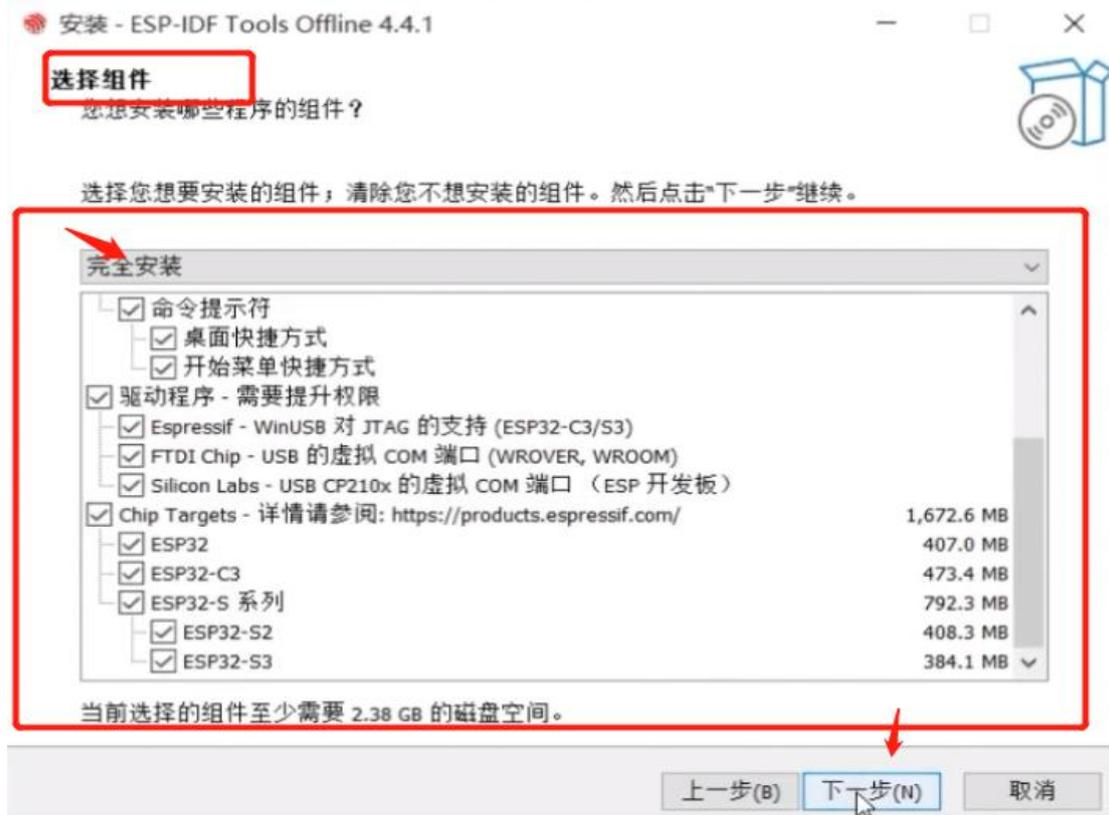
选中“我同意此协议”，然后进入“下一步”。



这里会有一个系统检查，显示“检查完成”后，再点击“下一步”。



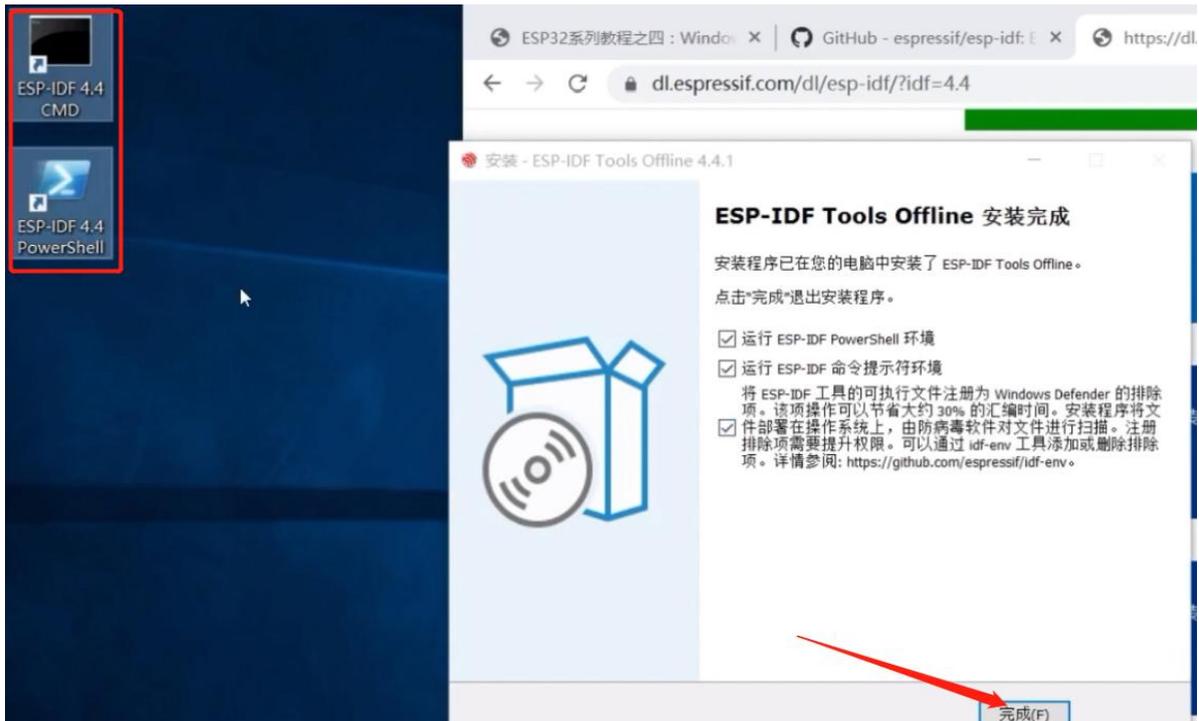
点击“浏览”选择ESP-IDF的安装路径，然后点击“下一步”。



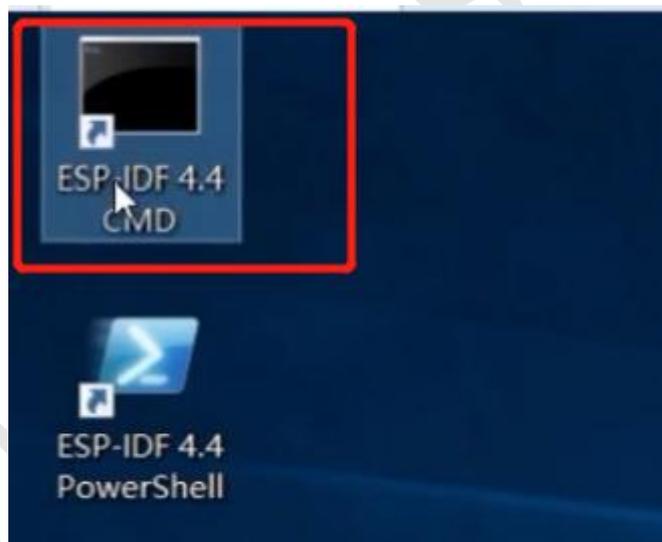
选择下载包含的组件，这里建议选择“完全安装”，然后点击“下一步”，直到安装完成即可。



安装的途中，可能会弹出这一个提示，勾选“信任”，点击“安装”即可。



安装完成后，桌面会显示两个ESP-IDF，是正常现象，直接点击“完成”即可。



双击打开“ESP-IDF4.4CMD”。

```

选择ESP-IDF 4.4 CMD
Using Git in C:\Users\QM\.espressif\tools\idf-git\2.34.2\cmd\
git version 2.34.1.windows.1
Setting IDF_PATH: C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1

Adding ESP-IDF tools to PATH...
C:\Users\QM\.espressif\tools\xtensa-esp32-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32-elf\bin
C:\Users\QM\.espressif\tools\xtensa-esp32s2-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s2-elf\bin
C:\Users\QM\.espressif\tools\xtensa-esp32s3-elf\esp-2021r2-patch3-8.4.0\xtensa-esp32s3-elf\bin
C:\Users\QM\.espressif\tools\riscv32-esp-elf\esp-2021r2-patch3-8.4.0\riscv32-esp-elf\bin
C:\Users\QM\.espressif\tools\esp32ulp-elf\2.28.51-esp-20191205\esp32ulp-elf\binutils\bin
C:\Users\QM\.espressif\tools\esp32s2ulp-elf\2.28.51-esp-20191205\esp32s2ulp-elf\binutils\bin
C:\Users\QM\.espressif\tools\cmake\3.20.3\bin
C:\Users\QM\.espressif\tools\openocd-esp32\v0.11.0-esp32-20211220\openocd-esp32\bin
C:\Users\QM\.espressif\tools\ninja\1.10.2\
C:\Users\QM\.espressif\tools\idf-exe\1.0.3\
C:\Users\QM\.espressif\tools\ccache\4.3\ccache-4.3-windows-64
C:\Users\QM\.espressif\tools\dfu-util\0.9\dfu-util-0.9-win64
C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\tools

Checking if Python packages are up to date...
Python requirements from C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\requirements.txt are satisfied.

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

    idf.py build

C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1>
    
```

如果打开后，显示的界面时如上图这样，就表示成功安装ESP-IDF，尝试编译一下ESP-IDF的示例，建议先把开发板接上。

- cd examples/get-started/hello-world      进入hello-world示例目录
- idf.py set-target esp32                    选择芯片类型（如果是C3就输入esp32c3）
- idf.py build                                编译项目
- idf.py flash (-b 115200 -p COM3)        烧录固件到开发板中（-b波特率-p设备端口）
- idf.py monitor                            监视串口

**快捷：** idf.py flash (-b 115200 -p COM3) monitor      一键三连（编译、烧录、监视）

```

C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started>cd hello_world
C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started\hello_world>idf.py build
Executing action: all (aliases: build)
Running cmake in directory c:\users\qm\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started\hello_world
Executing "cmake -G Ninja -DPYTHON_DEPS_CHECKED=1 -DESP_PLATFORM=1 -DCCACHE_ENABLE=1 c:\users\qm\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started\hello_world"...
-- IDF_TARGET not set, using default target: esp32
-- Found Git: C:/Users/QM/.espressif/tools/idf-git/2.34.2/cmd/git.exe (found version "2.34.1.windows.1")
    
```

```

C:\Users\QM\.espressif\python_env\idf4.4_py3.8_env\scripts\python.exe ... (components)
--port (PORT) --baudrate 460800 --before default_reset --after hard_reset --chip esp32 write_flash --flash
--flash_freq 40m 0x1000 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition_table.bin
or run 'idf.py -p (PORT) flash'
C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started\hello_world>
    
```

当显示“idf.py -p (PORT) flash”，即为编译成功，可以进行烧录，可以看到flash上面有一个端口选择的参数，需要找到开发板的设备端口。（如果只有一个设备连入，可以不用填写这个参数）



打开电脑的“设备管理器”，点击“端口”，红框的这个就是设备端口COM3。如果有很多个设备，分不清是哪一个，就停留在这个页面，拔插一下自己的开发板，看看前后变化就可以确定了。

```
C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started\hello_world>idf.py -p COM3 flash -b 115200 monitor
Executing action: flash
Running ninja in directory c:\users\qm\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started\hello_world\build
Executing "ninja flash"...
[1/5] cmd.exe /C "cd /D C:\Users\QM\.espressif\frameworks\...1/examples/get-started/hello_world/build/hello_world.bin"
hello_world.bin binary size 0x297f0 bytes. Smallest app partition is 0x100000 bytes. 0xd6810 bytes (84%) free.
[2/5] Performing build step for 'bootloader'
[1/1] cmd.exe /C "cd /D C:\Users\QM\.espressif\frameworks\esp-idf-v4.4.1\examples\get-started/hello_world/build/bootload
er\esp-idf\esptool_py && C:\Users\QM\.espressif\python_env\idf4.4_py3.8_env\Scripts\python.exe C:\Users\QM\.espressif/fr
ameworks/esp-idf-v4.4.1/components/partition_table/check_sizes.py --offset 0x8000 bootloader 0x1000 C:\Users\QM\.espress
if/frameworks/esp-idf-v4.4.1/examples/get-started/hello_world/build/bootloader/bootloader.bin"
Bootloader binary size 0x62d0 bytes. 0xd30 bytes (12%) free.
[2/3] cmd.exe /C "cd /D C:\Users\QM\.espressif\frameworks\...sp-idf-v4.4.1/components/esptool_py/run_serial_tool.cmake"
esptool.py esp32 -p COM3 -b 115200 --before=default_reset --after=hard_reset write_flash --flash_mode dio --flash_freq 4
0m --flash_size 2MB 0x1000 bootloader/bootloader.bin 0x10000 hello_world.bin 0x8000 partition_table/partition-table.bin
esptool.py v3.3-dev
```

这里为了方便，直接将烧录和监视一次性执行，（编译也可以通过这个一键三连的命令执行）

```
Binary image header.  
I (307) cpu_start: Starting scheduler on PRO CPU.  
I (0) cpu_start: Starting scheduler on APP CPU.  
Hello world!  
This is esp32 chip with 2 CPU core(s), WiFi/BT/BLE, silicon revision 1, 2MB external flash  
Minimum free heap size: 295364 bytes  
Restarting in 10 seconds...  
Restarting in 9 seconds...  
Restarting in 8 seconds...  
Restarting in 7 seconds...
```

运行成功!

效果为：显示一个“Hello world! ”，并开始10秒的倒计时，时间到系统自动重启。

Wireless-tag