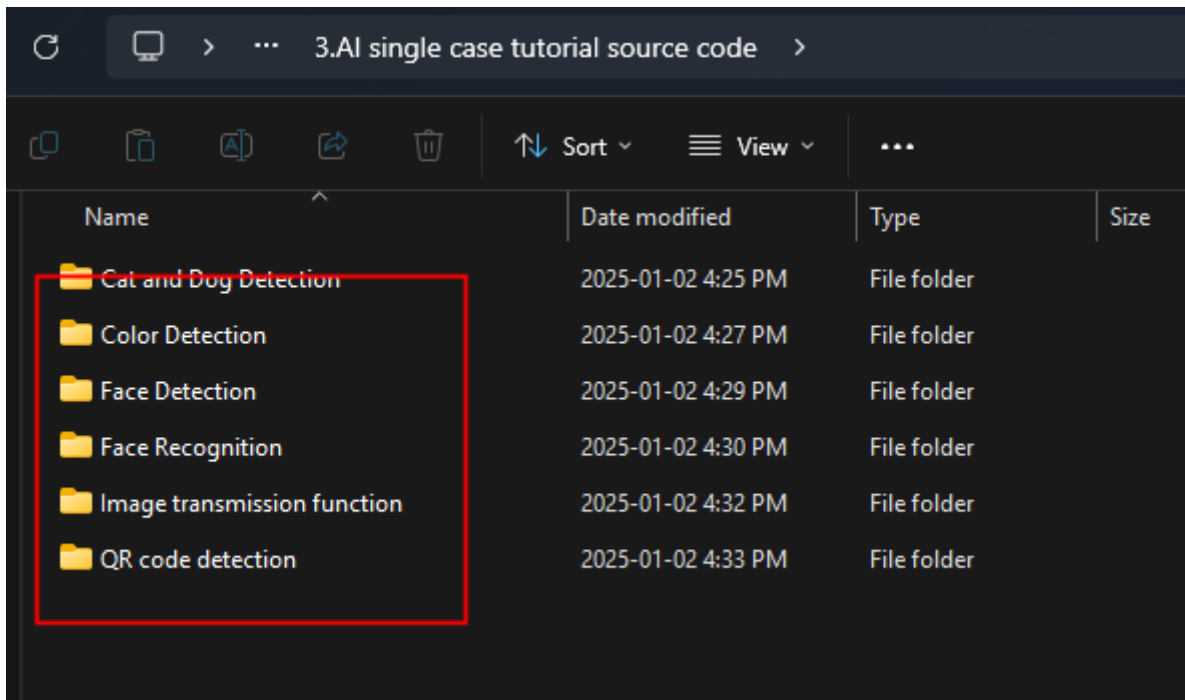


How to compile and use AI singleton tutorial

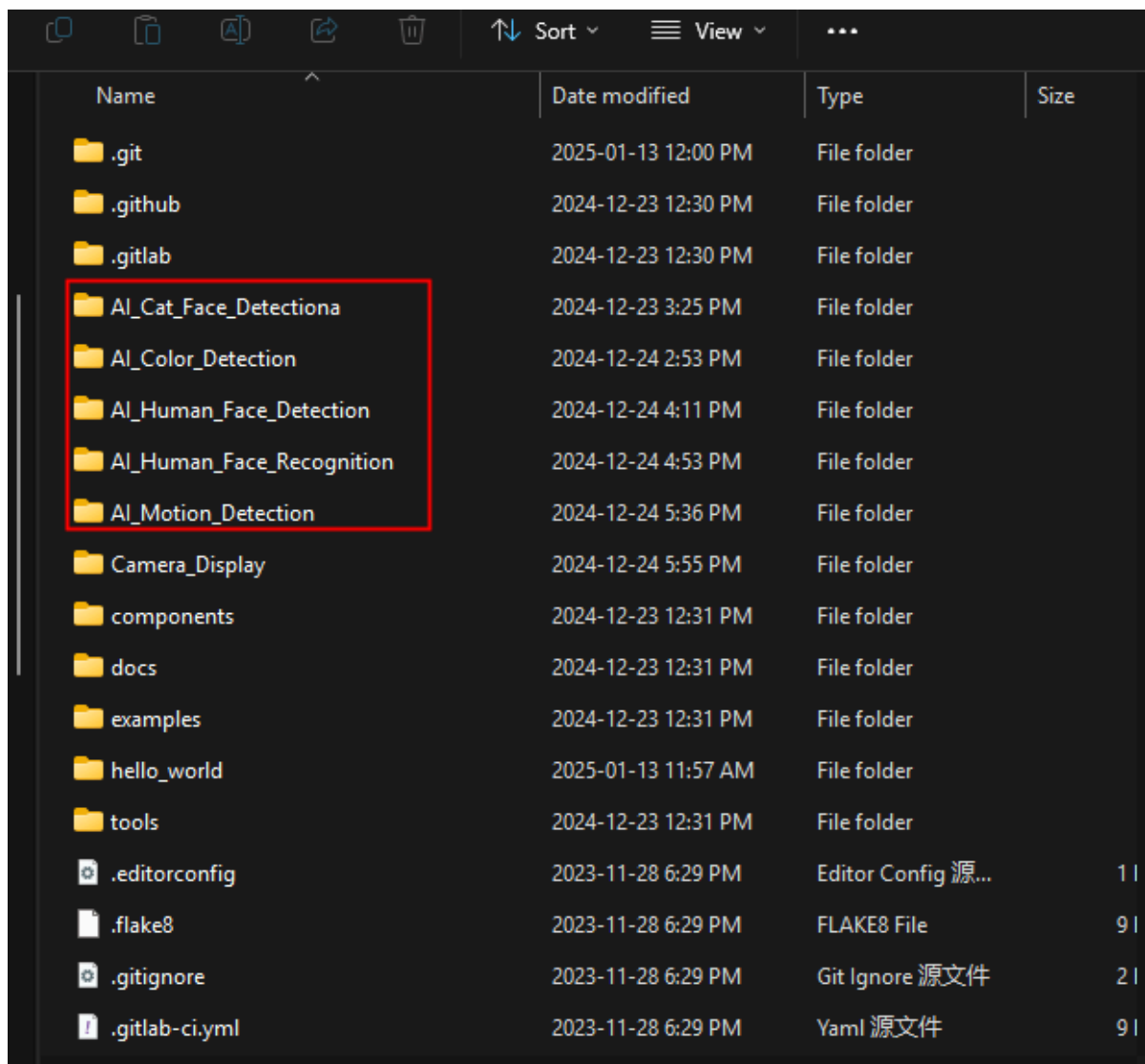
1. Configure environment burning

Open the AI singleton source code provided by the document,



Put the above source code in the path of esp-ide,

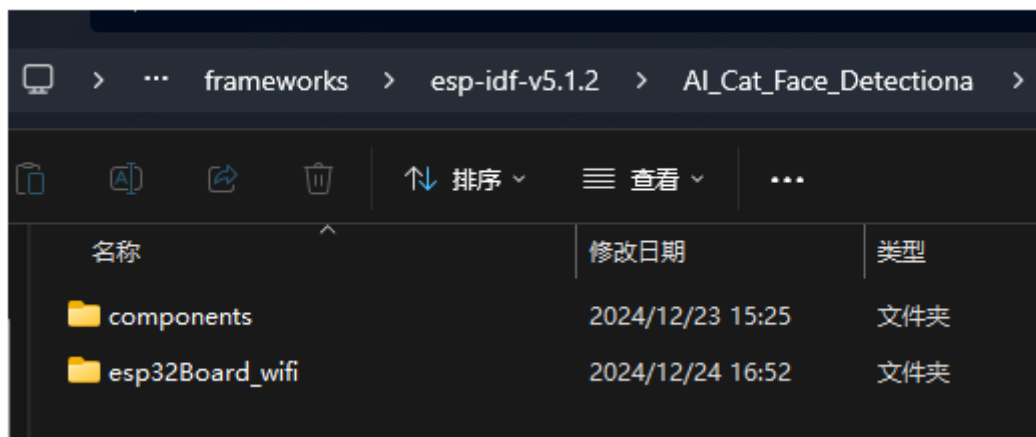
```
C:\Espressif\frameworks\esp-idf-v5.1.2>
```



Name	Date modified	Type	Size
.git	2025-01-13 12:00 PM	File folder	
.github	2024-12-23 12:30 PM	File folder	
.gitlab	2024-12-23 12:30 PM	File folder	
AI_Cat_Face_Detectionna	2024-12-23 3:25 PM	File folder	
AI_Color_Detection	2024-12-24 2:53 PM	File folder	
AI_Human_Face_Detection	2024-12-24 4:11 PM	File folder	
AI_Human_Face_Recognition	2024-12-24 4:53 PM	File folder	
AI_Motion_Detection	2024-12-24 5:36 PM	File folder	
Camera_Display	2024-12-24 5:55 PM	File folder	
components	2024-12-23 12:31 PM	File folder	
docs	2024-12-23 12:31 PM	File folder	
examples	2024-12-23 12:31 PM	File folder	
hello_world	2025-01-13 11:57 AM	File folder	
tools	2024-12-23 12:31 PM	File folder	
.editorconfig	2023-11-28 6:29 PM	Editor Config 源...	11
.flake8	2023-11-28 6:29 PM	FLAKE8 File	91
.gitignore	2023-11-28 6:29 PM	Git Ignore 源文件	21
.gitlab-ci.yml	2023-11-28 6:29 PM	Yaml 源文件	91

In this way, we can modify the singleton code directly in this environment, or burn the singleton code.

For example, if we want to download cat and dog recognition, we click on the AI_Cat_Face_Detectionna folder to enter,



名称	修改日期	类型
components	2024/12/23 15:25	文件夹
esp32Board_wifi	2024/12/24 16:52	文件夹

Then click on the esp32Board_wifi folder,

esp-idf-v5.1.2 > AI_Cat_Face_Detection > esp32Board_wifi

Name	Date modified	Type	Size
bin	2025-01-13 2:31 PM	File folder	
build	2024-12-24 6:32 PM	File folder	
main	2024-12-23 3:25 PM	File folder	
CMakeLists.txt	2024-12-24 2:46 PM	Text Document	1 KB
dependencies.lock	2024-12-20 2:17 PM	LOCK File	1 KB
partitions.csv	2024-12-20 2:17 PM	XLS 工作表	1 KB
sdkconfig	2024-12-20 2:17 PM	File	61 KB
sdkconfig.defaults	2024-12-20 2:17 PM	DEFAULTS File	2 KB
sdkconfig.defaults.esp32s3	2024-12-20 2:17 PM	ESP32S3 File	2 KB
sdkconfig.old	2024-12-20 2:17 PM	OLD File	54 KB

In this path, you can directly compile the code and directly cd in the tool To the current path,

```
C:\Espressif\frameworks\esp-idf-v5.1.2>cd C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi
```

```
C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi>
```

At this time, enter the command to compile

```
idf.py build
```

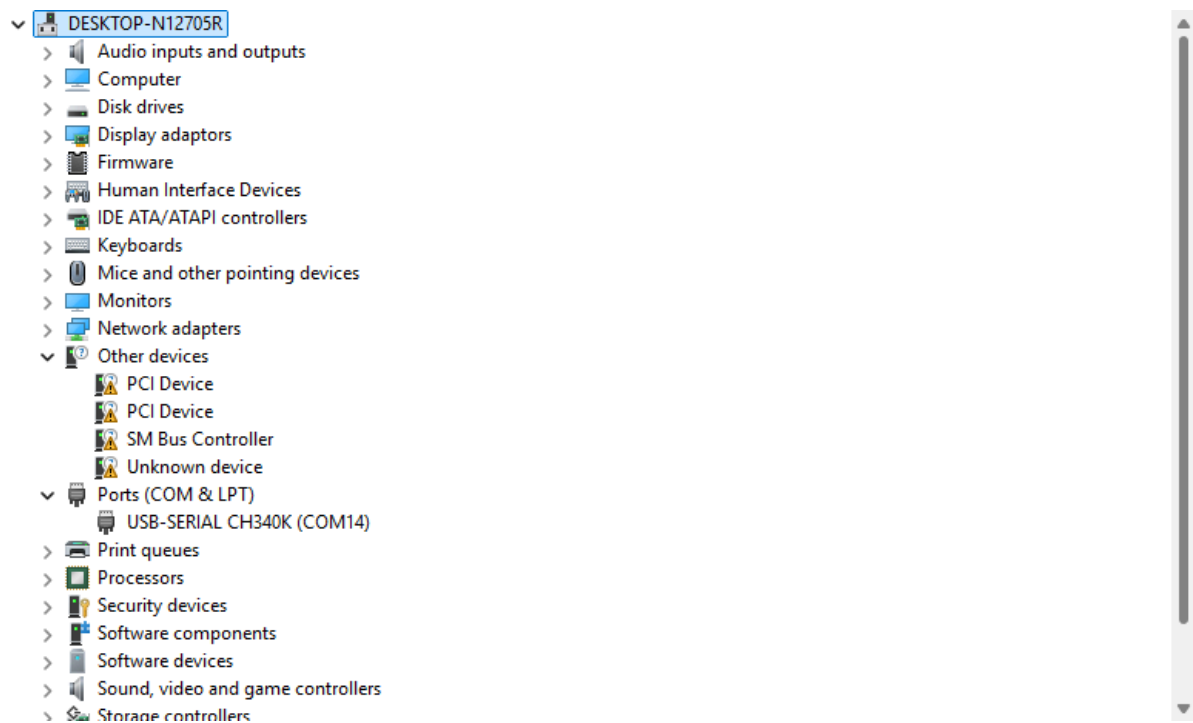
When this interface appears, the compilation is successful,

```
C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi> idf.py build
Executing action: all (aliases: build)
Running ninja in directory C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi\build
Executing "ninja all"...
[1/4] cmd.exe /C "cd /D C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi\build\AI_Cat_Face_Detection\bin"
AI_Cat_Face_Detection.bin binary size 0x1726f0 bytes. Smallest app partition is 0x3b6000 bytes. 0x243910 bytes (61%) free.
[1/1] cmd.exe /C "cd /D C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi\build\bootloader\bin"
bootloader binary size 0x3b80 bytes. 0x4480 bytes (54%) free.

Project build complete. To flash, run this command:
C:\Espressif\python_env\idf5.1_py3.11_env\Scripts\python.exe ..\..\components\esptool_py\esptool\esptool.py -p (PORT) -b
460800 --before default_reset --after hard_reset --chip esp32s3 --no-stub write_flash --flash_mode dio --flash_size 4MB
--flash_freq 80m 0x0 build\bootloader\bootloader.bin 0x8000 build\partition_table\partition-table.bin 0x10000 build\AI_
Cat_Face_Detection.bin
or run 'idf.py -p (PORT) flash'
C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi>
```

If everything is normal, a .bin file will be generated after the compilation is completed.

Connect the device to the computer, and you can directly use the COM port to burn. Open the computer's device manager to check what port is recognized.



In the terminal, enter the following command to write the program in.

```
idf.py -p com14 flash
```

```
Writing at 0x0017ec00... (99 %)
Writing at 0x0017f000... (99 %)
Writing at 0x0017f400... (99 %)
Writing at 0x0017f800... (99 %)
Writing at 0x0017fc00... (99 %)
Writing at 0x00180000... (99 %)
Writing at 0x00180400... (99 %)
Writing at 0x00180800... (99 %)
Writing at 0x00180c00... (99 %)
Writing at 0x00181000... (99 %)
Writing at 0x00181400... (99 %)
Writing at 0x00181800... (99 %)
Writing at 0x00181c00... (99 %)
Writing at 0x00182000... (99 %)
Writing at 0x00182400... (100 %)
Wrote 1517568 bytes at 0x00010000 in 47.0 seconds (258.2 kbit/s)...
Hash of data verified.
Erasing flash...
Took 0.07s to erase flash block
Writing at 0x00008000... (33 %)
Writing at 0x00008400... (66 %)
Writing at 0x00008800... (100 %)
Wrote 3072 bytes at 0x00008000 in 0.1 seconds (271.6 kbit/s)...
Hash of data verified.
Leaving...
Hard resetting via RTS pin...
Done
C:\Espressif\frameworks\esp-idf-v5.1.2\AI_Cat_Face_Detection\esp32Board_wifi>
```

If the above log appears, the download is successful.

2. Modify the code

If you want to modify the code for secondary development, you can open the main folder,

文件夹	.vscode	2024/12/23 15:25	文件夹
文件夹	ai	2024/12/24 14:30	文件夹
文件夹	include	2024/12/24 14:31	文件夹
文件夹	src	2024/12/23 19:36	文件夹
C++ 源文件	app_main.cpp	2024/12/24 14:27	C++ 源文件
文本文档	CMakeLists.txt	2024/12/20 14:17	文本文档

The `src` folder and `app_main.cpp` are both programs that can be modified, one is equivalent to the main function, and the other is the source program called by the main function.

A screenshot of the ESP32 IDE's file explorer. The breadcrumb path at the top is 'esp32Board_wifi > main > src'. The file list shows the following items: a folder named '.vscode' (last modified 2024/12/23 15:25), and several C++ source files: 'app_myhttpd.cpp' (2024/12/20 14:17), 'app_mymdns.c' (2024/12/20 14:17), 'app_myperipherals.c' (2024/12/20 14:17), 'app_mywifi.c' (2024/12/23 19:39), 'my_usart.c' (2024/12/24 14:28), 'my_usart1_user.c' (2024/12/24 14:23), 'my_user_iic.c' (2024/12/24 14:24), 'mykey.c' (2024/12/24 14:27), and 'yahboom_camera.c' (2024/12/20 14:17). The 'app_mywifi.c' file is currently selected and highlighted.

After the modification, repeat the above compilation steps. Burn after compilation.

