

win11 安装opencv c++ GNU 在vscode上可用、可调试

1.cmake3.30.1

安装windowx86-64.msi版本

Latest Release (3.30.1)

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gzipped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gzipped tar files of the install tree. The OS-machine.tar.Z files are compressed tar files of the install tree. The tar file distributions can be untared in any directory. They are prefixed by the version of CMake. For example, the linux-x86_64 tar file is all under the directory cmake-linux-x86_64. This prefix can be removed as long as the share, bin, man and doc directories are moved relative to each other. To build the source distributions, unpack them with zip or tar and follow the instructions in README.rst at the top of the source tree. See also the [CMake 3.30 Release Notes](#).

Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.30.1.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.30.1.zip

Binary distributions:

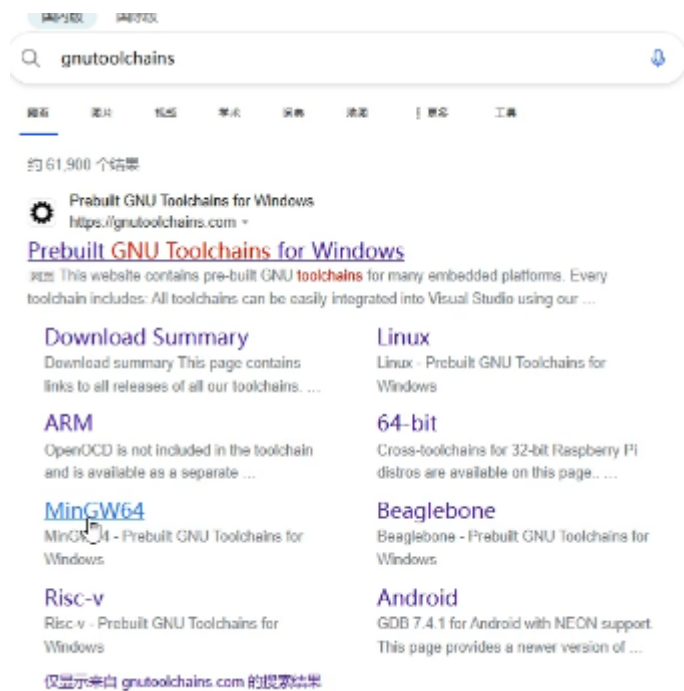
Platform	Files
Windows x64 Installer:	cmake-3.30.1-windows-x86_64.msi
Windows x64 ZIP	cmake-3.30.1-windows-x86_64.zip
Windows i386 Installer:	cmake-3.30.1-windows-i386.msi
Windows i386 ZIP	cmake-3.30.1-windows-i386.zip
Windows ARM64 Installer:	cmake-3.30.1-windows-arm64.msi
Windows ARM64 ZIP	cmake-3.30.1-windows-arm64.zip

安装

添加桌面图标

路径改为：E:\CMake

2.搜：gnutoolchains--Mingw64



mingw64-gcc12.2.0.exe

SYSPROGS

PRODUCTS DOWNLOADS TUTORIALS SUPPORT CONTACT

Toolchains

- Android (GDB only)
- ARM
- AVR
- Beaglebone
- Blackfin
- Cubieboard
- ESP32
- ESP8266
- Nvidia Jetson
- Kendryte

Prebuilt Windows Toolchain for MinGW64

The following toolchain releases are available:

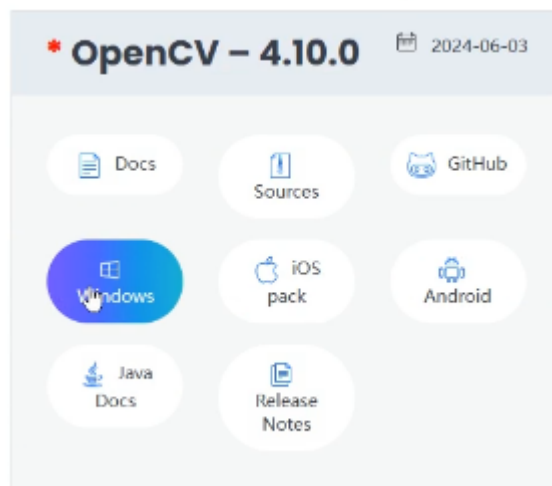
Binutils	GCC	Newlib	GDB	Download link
2.39	12.2.0	msvcrt	12.1	mingw64-gcc12.2.0.exe (79 MB)
2.30	9.1.0	msvcrt	8.3	mingw64-gcc9.1.0.exe (69 MB)
2.30	8.1.0	msvcrt	8.1	mingw64-gcc8.1.0.exe (46 MB)
2.23	4.7.1		7.5.0	mingw64-gcc4.7.1.exe (31 MB)

安装

路径改为：E:\SysGCC

接受

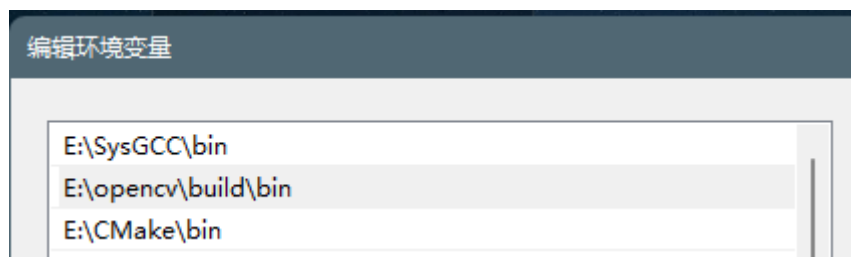
3.opencv-4.10.0



解压

路径改为: E:\

4.环境变量--path



如果

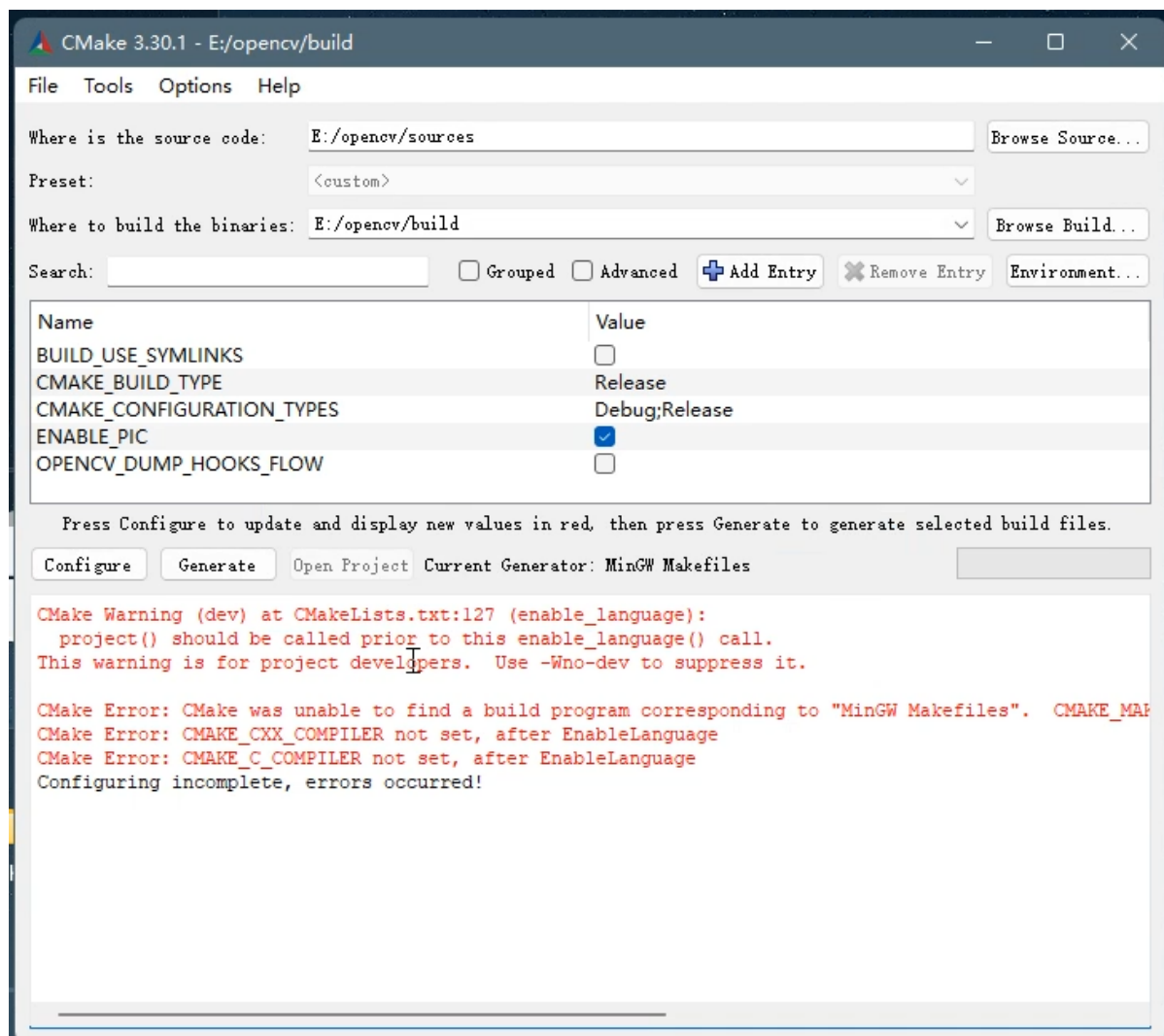
`gcc --version`

查看是比较新的版本，我这里是12.2.0

5.打开cmake-gui

source code: E:/opencv/sources

build binaries: E:/opencv/build



configure

Specify the generator for this project

MinGW Makefiles

- ☒ Use default native compilers
- ☐ Specify native compilers
- ☐ Specify toolchain file for cross-compiling
- ☐ Specify options for cross-compiling

失败则重启windows重试

成功则点generate

6.网盘下载generate.exe

复制到E:\opencv\build\bin下

7.在opencv/build下

mingw32-make

```
[ 70%] Built target opencv_test_videoio
[ 70%] Built target opencv_perf_videoio
[ 73%] Built target opencv_calib3d
[ 75%] Built target opencv_test_calib3d
[ 76%] Built target opencv_perf_calib3d
[ 76%] Built target opencv_test_highgui
[ 77%] Built target opencv_objdetect
[ 78%] Built target opencv_test_objdetect
[ 78%] Built target opencv_perf_objdetect
[ 79%] Built target opencv_stitching
[ 80%] Built target opencv_test_stitching
[ 80%] Built target opencv_perf_stitching
[ 82%] Built target opencv_video
[ 83%] Built target opencv_test_video
[ 84%] Built target opencv_perf_video
[ 85%] Built target ade
[ 93%] Built target opencv_gapi
[ 98%] Built target opencv_test_gapi
[ 98%] Built target opencv_perf_gapi
[ 98%] Built target opencv_annotation
[ 98%] Built target opencv_visualisation
[100%] Built target opencv_interactive-calibration
[100%] Built target opencv_version
[100%] Built target opencv_version_win32
[100%] Built target opencv_model_diagnostics
```

8.有可能要重装powershell和重启，酌情做

9.终端-generate

generate.exe E:/opencv

```

PS C:\Users\75954> generate
请提供文件路径作为参数
generate.exe opencv_floder_path
generate.exe default -> generate.exe D:/opencv
PS C:\Users\75954> generate.exe E:/opencv
***** .vscode/c_cpp_properties.json *****
*****
F1 -> C/C++:Edit Configurations(UI) -> Include path(ap
pend)

E:/opencv/build/include/**

***** .vscode/tasks.json *****
*****

"-IE:\\opencv\\build\\include", "-LE:\\opencv\\build\\bin", "-
lopencv_calib3d4100", "-lopencv_core4100", "-lopencv_dnn4100
", "-lopencv_features2d4100", "-lopencv_flann4100", "-lopenc
v_gapi4100", "-lopencv_highgui4100", "-lopencv_imgcodecs4100
", "-lopencv_imgproc4100", "-lopencv_ml4100", "-lopencv_objd
etect4100", "-lopencv_photo4100", "-lopencv_stitching4100",
"-lopencv_video4100", "-lopencv_videoio4100", "-lopencv_vide
oio_ffmpeg4100_64",

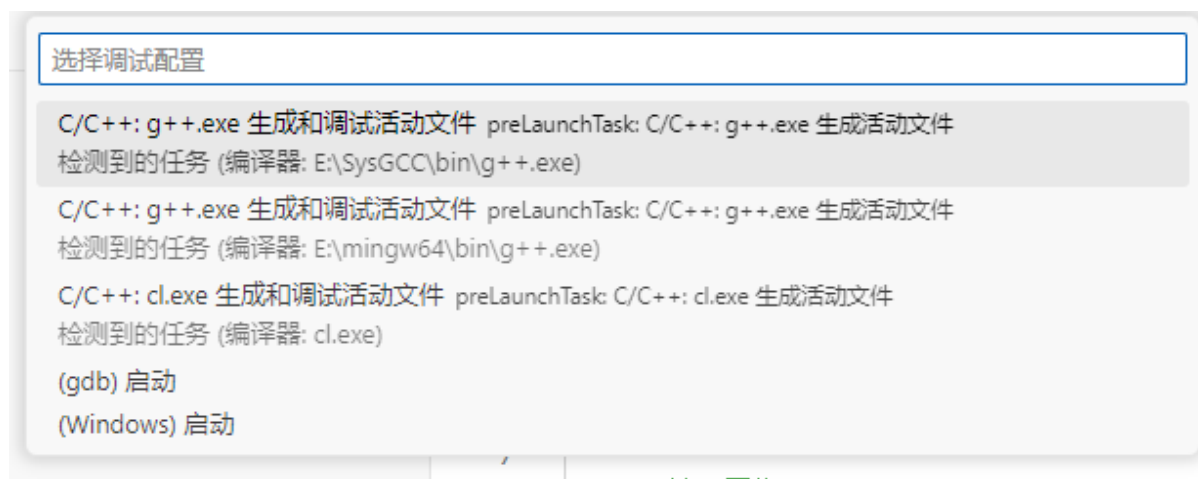
```

10.打开项目

把刚刚指示的路径都复制进去，也可以直接复制这里的整个文件

常规方法

c++文件，点击运行



失败，并出现一个.vscode/tasks.json

将

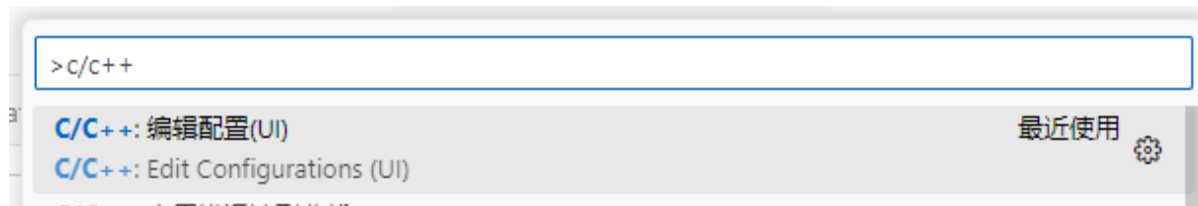
```
"-IE:\\opencv\\build\\include","-LE:\\opencv\\build\\bin","-lopencv_calib3d4100",
"-lopencv_core4100", "-lopencv_dnn4100", "-lopencv_features2d4100", "-
lopencv_flann4100", "-lopencv_gapi4100", "-lopencv_highgui4100", "-
lopencv_imgcodecs4100", "-lopencv_imgproc4100", "-lopencv_ml4100", "-
lopencv_objdetect4100", "-lopencv_photo4100", "-lopencv_stitching4100", "-
lopencv_video4100", "-lopencv_videoio4100", "-lopencv_videoio_ffmpeg4100_64",
```

复制到"\${file}",的下一行

选中并格式化该行内容

shift+ctrl+p 召唤c/c++

选编辑配置



编译器路径:

E:/SysGCC/bin/g++.exe

IntelliSense 模式:

\${default}

包含路径: 加一个

E:/opencv/build/include/**

直接复制

.vscode/c_cpp_properties.json

```
{
  "configurations": [
    {
      "name": "Win32",
      "includePath": [
        "${workspaceFolder}/**",
        "E:/opencv/build/include/**"
      ],
      "defines": [
        "_DEBUG",
        "UNICODE",
        "_UNICODE"
      ],
      "windowsSdkVersion": "10.0.22621.0",
      "compilerPath": "E:/SysGCC/bin/g++.exe",
      "cStandard": "c17",
      "cppStandard": "c++17",
      "intelliSenseMode": "${default}"
    }
  ]
}
```

```

    }
  ],
  "version": 4
}

```

.vscode/tasks.json

```

{
  "tasks": [
    {
      "type": "cppbuild",
      "label": "C/C++: g++.exe 生成活动文件",
      "command": "E:/SysGCC/bin/g++.exe",
      "args": [
        "-fdiagnostics-color=always",
        "-g",
        "${file}",
        "-Ie:\\opencv\\build\\include",
        "-Le:\\opencv\\build\\bin",
        "-lopencv_calib3d4100",
        "-lopencv_core4100",
        "-lopencv_dnn4100",
        "-lopencv_features2d4100",
        "-lopencv_flann4100",
        "-lopencv_gapi4100",
        "-lopencv_highgui4100",
        "-lopencv_imgcodecs4100",
        "-lopencv_imgproc4100",
        "-lopencv_ml4100",
        "-lopencv_objdetect4100",
        "-lopencv_photo4100",
        "-lopencv_stitching4100",
        "-lopencv_video4100",
        "-lopencv_videoio4100",
        "-lopencv_videoio_ffmpeg4100_64",
        "-o",
        "${fileDirname}\\${fileBasenameNoExtension}.exe"
      ],
      "options": {
        "cwd": "E:/SysGCC/bin"
      },
      "problemMatcher": [
        "$gcc"
      ],
      "group": {
        "kind": "build",
        "isDefault": true
      },
      "detail": "调试器生成的任务。"
    }
  ],
  "version": "2.0.0"
}

```