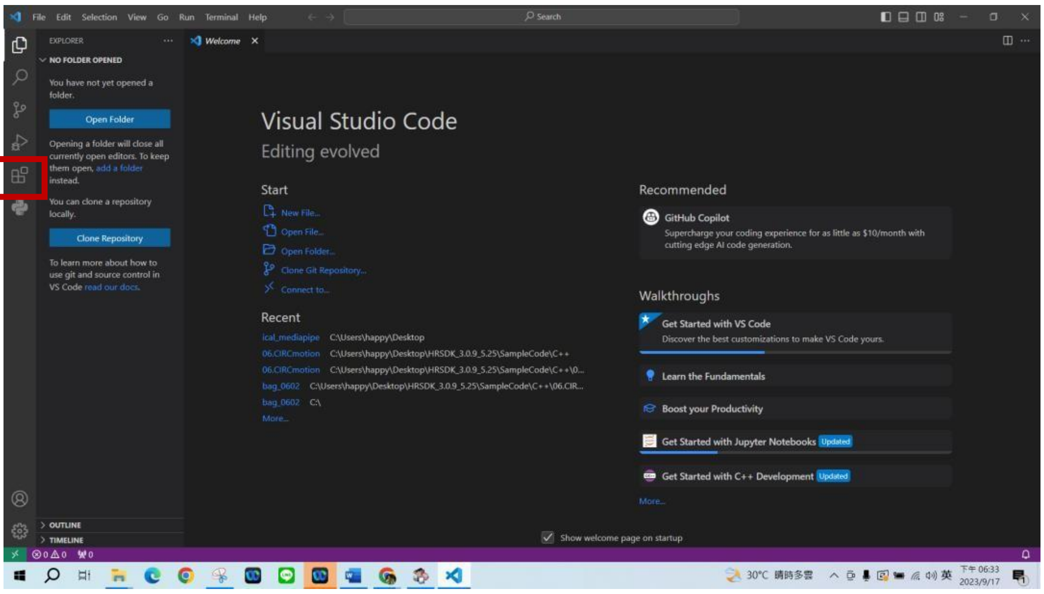
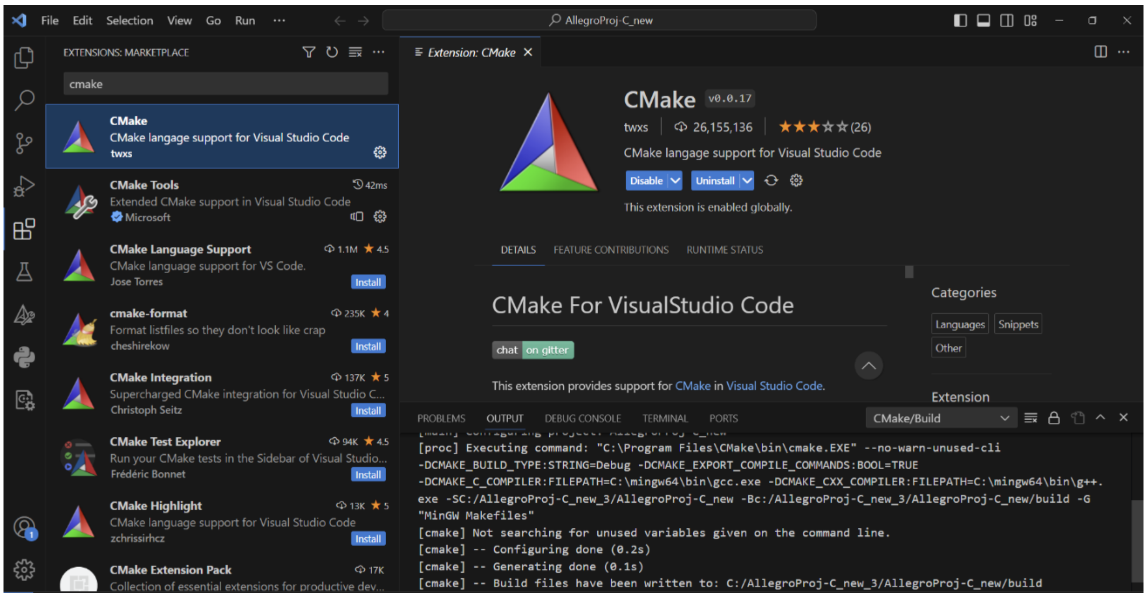
1120 Allegro VSCode 環境安裝

■ 開啟 Visual Studio Code 進入 Extension

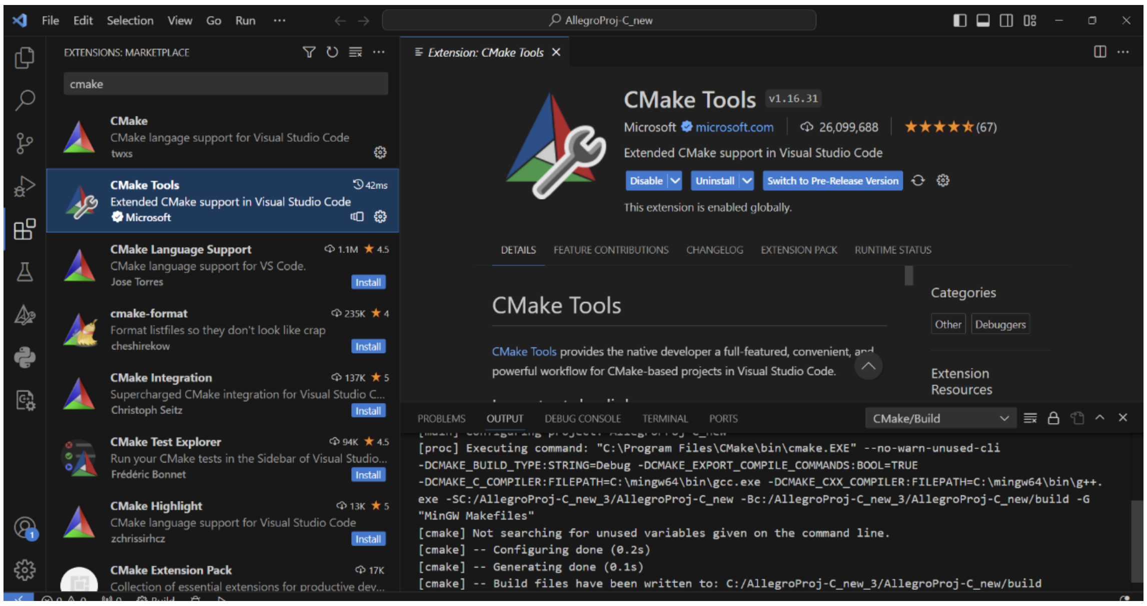


■ 安裝 CMake([自行參閱CMake詳細教學])



```
[proc] Executing command: "C:\Program Files\CMake\bin\cmake.EXE" --no-warn-unused-cli
-DCMAKE_BUILD_TYPE:STRING=Debug -DCMAKE_EXPORT_COMPILE_COMMANDS:BOOL=TRUE
-DCMAKE_C_COMPILER:FILEPATH=C:\mingw64\bin\gcc.exe -DCMAKE_CXX_COMPILER:FILEPATH=C:\mingw64\bin\g++.
exe -SC:/AllegroProj-C_new_3/AllegroProj-C_new -Bc:/AllegroProj-C_new_3/AllegroProj-C_new/build -G
"MinGW Makefiles"
[cmake] Not searching for unused variables given on the command line.
[cmake] -- Configuring done (0.2s)
[cmake] -- Generating done (0.1s)
[cmake] -- Build files have been written to: C:/AllegroProj-C_new_3/AllegroProj-C_new/build
```

■ 安裝 CMake Tools



CMake
CMake langage support for Visual Studio Code
twxs

CMake Tools 42ms
Extended CMake support in Visual Studio Code
Microsoft

CMake Language Support 1.1M ★ 4.5
CMake language support for VS Code.
Jose Torres

cmake-format 235K ★ 4
Format listfiles so they don't look like crap
cheshirekow

CMake Integration 137K ★ 5
Supercharged CMake integration for Visual Studio C...
Christoph Seitz

CMake Test Explorer 94K ★ 4.5
Run your CMake tests in the Sidebar of Visual Studio...
Frédéric Bonnet

CMake Highlight 13K ★ 5
CMake language support for Visual Studio Code
zchrissirhcz

CMake Extension Pack 17K
Collection of essential extensions for productive dev...

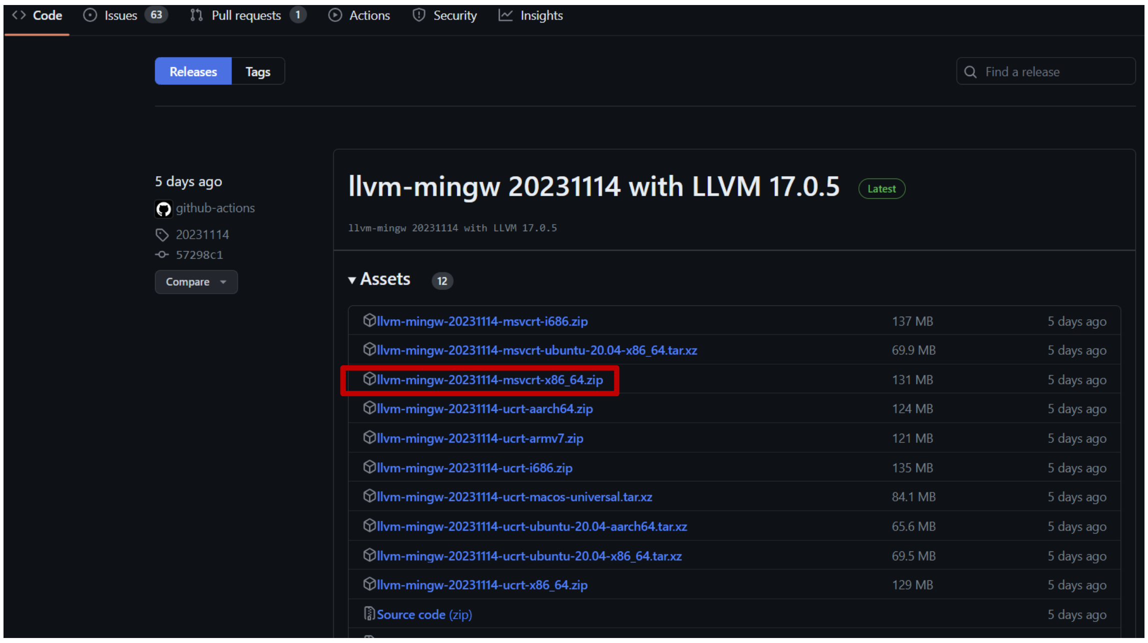
CMake Tools v1.16.31
Microsoft microsoft.com 26,099,688 ★★★★★(67)
Extended CMake support in Visual Studio Code
This extension is enabled globally.

CMake Tools

CMake Tools provides the native developer a full-featured, convenient, and powerful workflow for CMake-based projects in Visual Studio Code.

```
[proc] Executing command: "C:\Program Files\CMake\bin\cmake.EXE" --no-warn-unused-cli
-DCMAKE_BUILD_TYPE:STRING=Debug -DCMAKE_EXPORT_COMPILE_COMMANDS:BOOL=TRUE
-DCMAKE_C_COMPILER:FILEPATH=C:\mingw64\bin\gcc.exe -DCMAKE_CXX_COMPILER:FILEPATH=C:\mingw64\bin\g++.
exe -SC:/AllegroProj-C_new_3/AllegroProj-C_new -Bc:/AllegroProj-C_new_3/AllegroProj-C_new/build -G
"MinGW Makefiles"
[cmake] Not searching for unused variables given on the command line.
[cmake] -- Configuring done (0.2s)
[cmake] -- Generating done (0.1s)
[cmake] -- Build files have been written to: C:/AllegroProj-C_new_3/AllegroProj-C_new/build
```

■ 安裝 MinGW 64



5 days ago
github-actions
20231114
57298c1

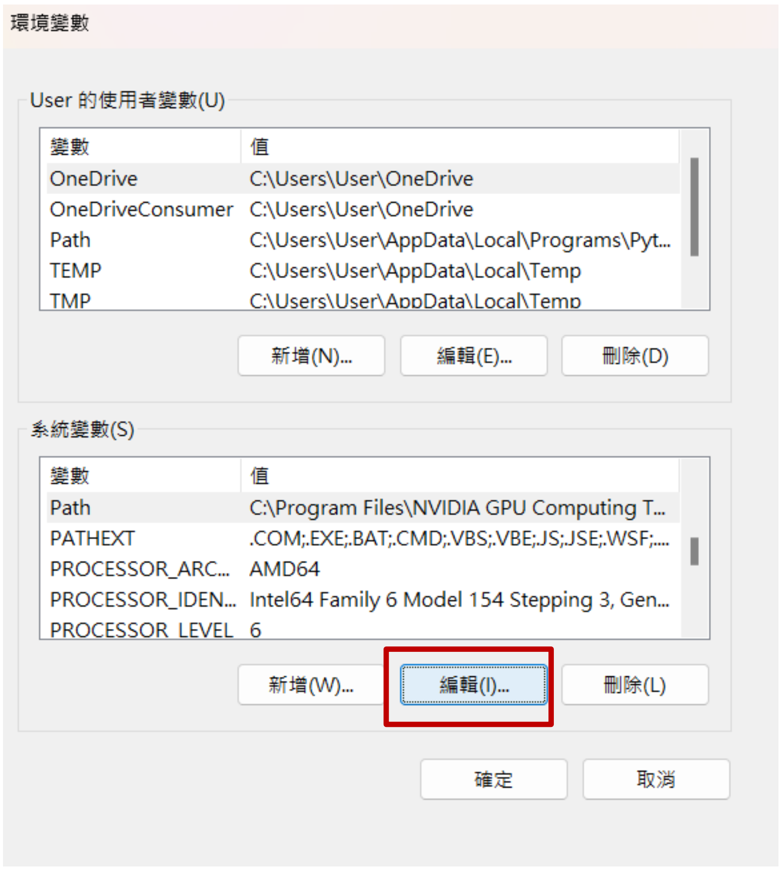
llvm-mingw 20231114 with LLVM 17.0.5

llvm-mingw 20231114 with LLVM 17.0.5

Assets 12

llvm-mingw-20231114-msvcrt-i686.zip	137 MB	5 days ago
llvm-mingw-20231114-msvcrt-ubuntu-20.04-x86_64.tar.xz	69.9 MB	5 days ago
llvm-mingw-20231114-msvcrt-x86_64.zip	131 MB	5 days ago
llvm-mingw-20231114-ucrt-aarch64.zip	124 MB	5 days ago
llvm-mingw-20231114-ucrt-armv7.zip	121 MB	5 days ago
llvm-mingw-20231114-ucrt-i686.zip	135 MB	5 days ago
llvm-mingw-20231114-ucrt-macos-universal.tar.xz	84.1 MB	5 days ago
llvm-mingw-20231114-ucrt-ubuntu-20.04-aarch64.tar.xz	65.6 MB	5 days ago
llvm-mingw-20231114-ucrt-ubuntu-20.04-x86_64.tar.xz	69.5 MB	5 days ago
llvm-mingw-20231114-ucrt-x86_64.zip	129 MB	5 days ago
Source code (zip)		5 days ago

■ 選系統變數下的Path，按下編輯


環境變數

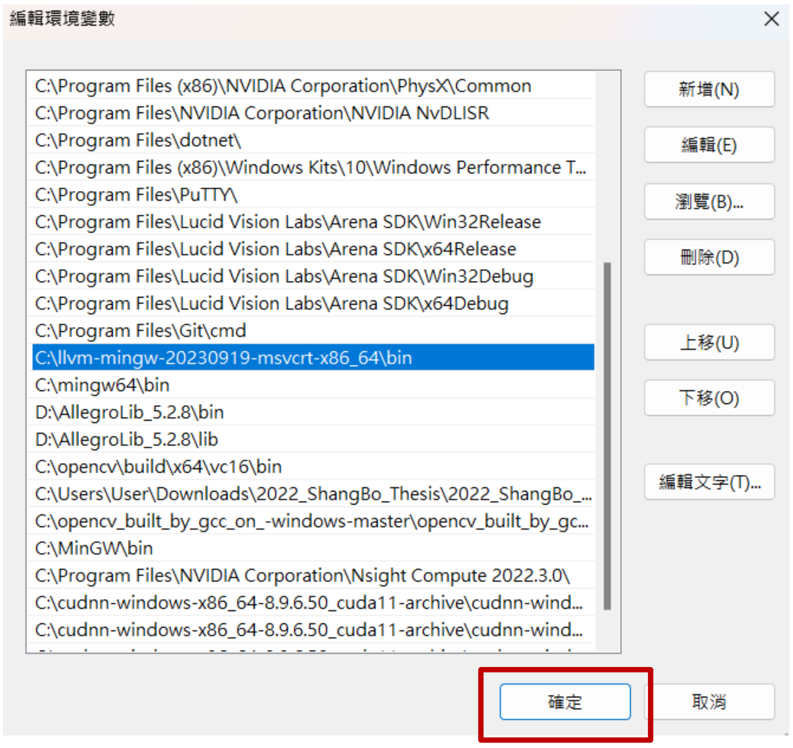
User 的使用者變數(U)

變數	值
OneDrive	C:\Users\User\OneDrive
OneDriveConsumer	C:\Users\User\OneDrive
Path	C:\Users\User\AppData\Local\Programs\Pyt...
TEMP	C:\Users\User\AppData\Local\Temp
TMP	C:\Users\User\AppData\Local\Temp

系統變數(S)

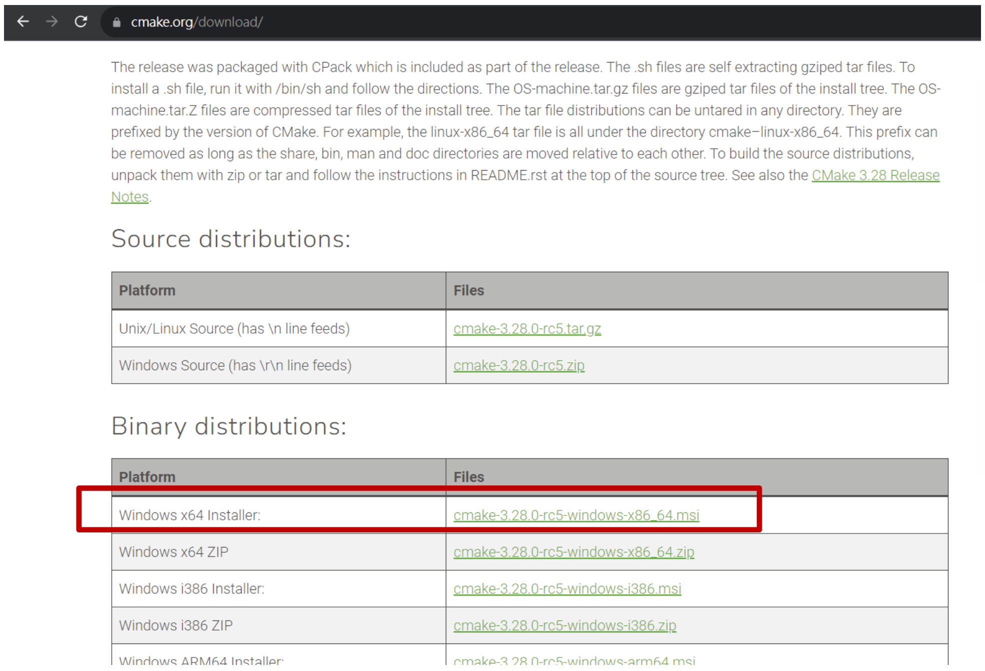
變數	值
Path	C:\Program Files\NVIDIA GPU Computing T...
PATHEXT	.COM;.EXE;.BAT;.CMD;.VBS;.VBE;.JS;.JSE;.WSF;....
PROCESSOR_ARC...	AMD64
PROCESSOR_IDEN...	Intel64 Family 6 Model 154 Stepping 3, Gen...
PROCESSOR_LEVEL	6

■ 新增路徑C:\llvm-mingw-20230919-msvcrt-x86_64\bin， 完成後點選確定


編輯環境變數

```
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\dotnet\
C:\Program Files (x86)\Windows Kits\10\Windows Performance T...
C:\Program Files\PuTTY\
C:\Program Files\Lucid Vision Labs\Arena SDK\Win32Release
C:\Program Files\Lucid Vision Labs\Arena SDK\x64Release
C:\Program Files\Lucid Vision Labs\Arena SDK\Win32Debug
C:\Program Files\Lucid Vision Labs\Arena SDK\x64Debug
C:\Program Files\Git\cmd
C:\llvm-mingw-20230919-msvcrt-x86_64\bin
C:\mingw64\bin
D:\AllegroLib_5.2.8\bin
D:\AllegroLib_5.2.8\lib
C:\opencv\build\x64\vc16\bin
C:\Users\User\Downloads\2022_ShangBo_Thesis\2022_ShangBo_...
C:\opencv_built_by_gcc_on_-windows-master\opencv_built_by_gc...
C:\MinGW\bin
C:\Program Files\NVIDIA Corporation\Nsight Compute 2022.3.0\
C:\cudnn-windows-x86_64-8.9.6.50_cuda11-archive\cudnn-wind...
C:\cudnn-windows-x86_64-8.9.6.50_cuda11-archive\cudnn-wind...
```

■ [安裝Cmake]



cmake.org/download/

The release was packaged with CPack which is included as part of the release. The .sh files are self extracting gziped tar files. To install a .sh file, run it with /bin/sh and follow the directions. The OS-machine.tar.gz files are gziped 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 3.28 Release Notes.

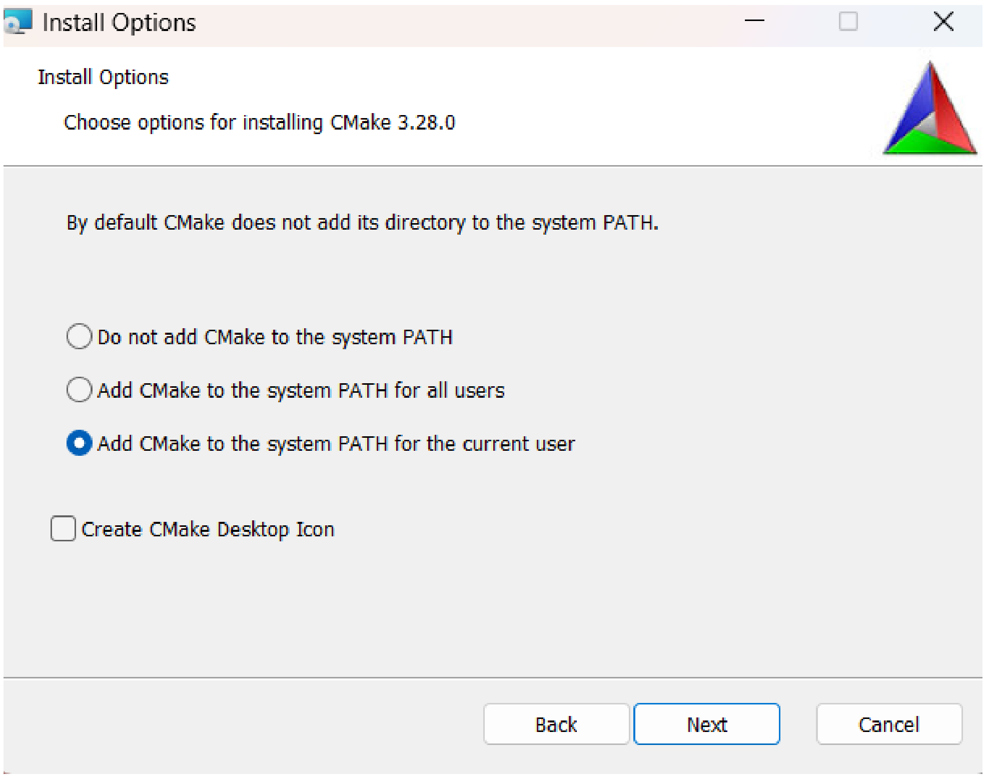
Source distributions:

Platform	Files
Unix/Linux Source (has \n line feeds)	cmake-3.28.0-rc5.tar.gz
Windows Source (has \r\n line feeds)	cmake-3.28.0-rc5.zip

Binary distributions:

Platform	Files
Windows x64 Installer:	cmake-3.28.0-rc5-windows-x86_64.msi
Windows x64 ZIP	cmake-3.28.0-rc5-windows-x86_64.zip
Windows i386 Installer:	cmake-3.28.0-rc5-windows-i386.msi
Windows i386 ZIP	cmake-3.28.0-rc5-windows-i386.zip
Windows ARM64 Installer:	cmake-3.28.0-rc5-windows-arm64.msi

■ 選擇 Add to current user 並點擊 Next 直至完成



Install Options

Install Options

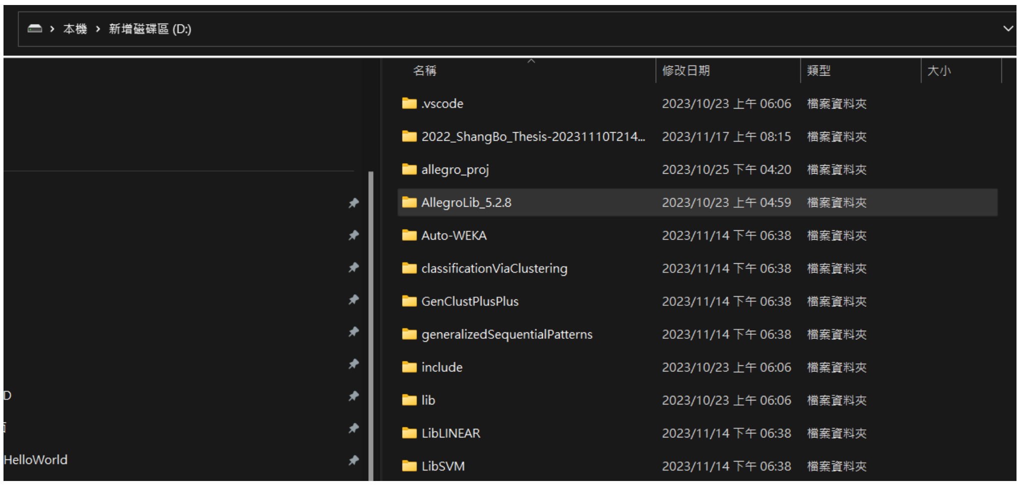
Choose options for installing CMake 3.28.0

By default CMake does not add its directory to the system PATH.

- ○ Do not add CMake to the system PATH
- ○ Add CMake to the system PATH for all users
- [✓ Add CMake to the system PATH for the current user]

☐ Create CMake Desktop Icon

■ 安裝 AllegroLib_5.2.8 至 D:\AllegroLib_5.2.8

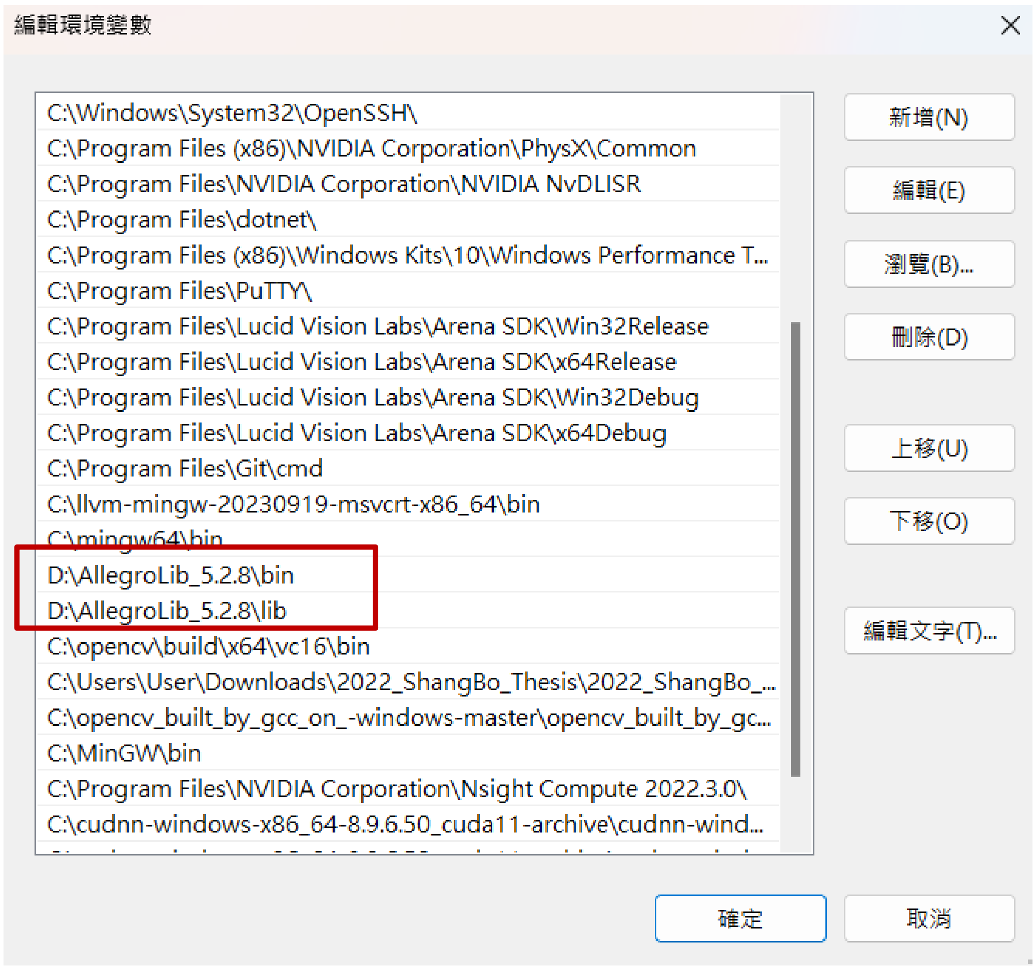


本機 › 新增磁碟區 (D:)

名稱	修改日期	類型	大小
.vscode	2023/10/23 上午 06:06	檔案資料夾	
2022_ShangBo_Thesis-20231110T214...	2023/11/17 上午 08:15	檔案資料夾	
allegro_proj	2023/10/25 下午 04:20	檔案資料夾	
AllegroLib_5.2.8	2023/10/23 上午 04:59	檔案資料夾	
Auto-WEKA	2023/11/14 下午 06:38	檔案資料夾	
classificationViaClustering	2023/11/14 下午 06:38	檔案資料夾	
GenClustPlusPlus	2023/11/14 下午 06:38	檔案資料夾	
generalizedSequentialPatterns	2023/11/14 下午 06:38	檔案資料夾	
include	2023/10/23 上午 06:06	檔案資料夾	
lib	2023/10/23 上午 06:06	檔案資料夾	
LibLINEAR	2023/11/14 下午 06:38	檔案資料夾	
LibSVM	2023/11/14 下午 06:38	檔案資料夾	

HelloWorld

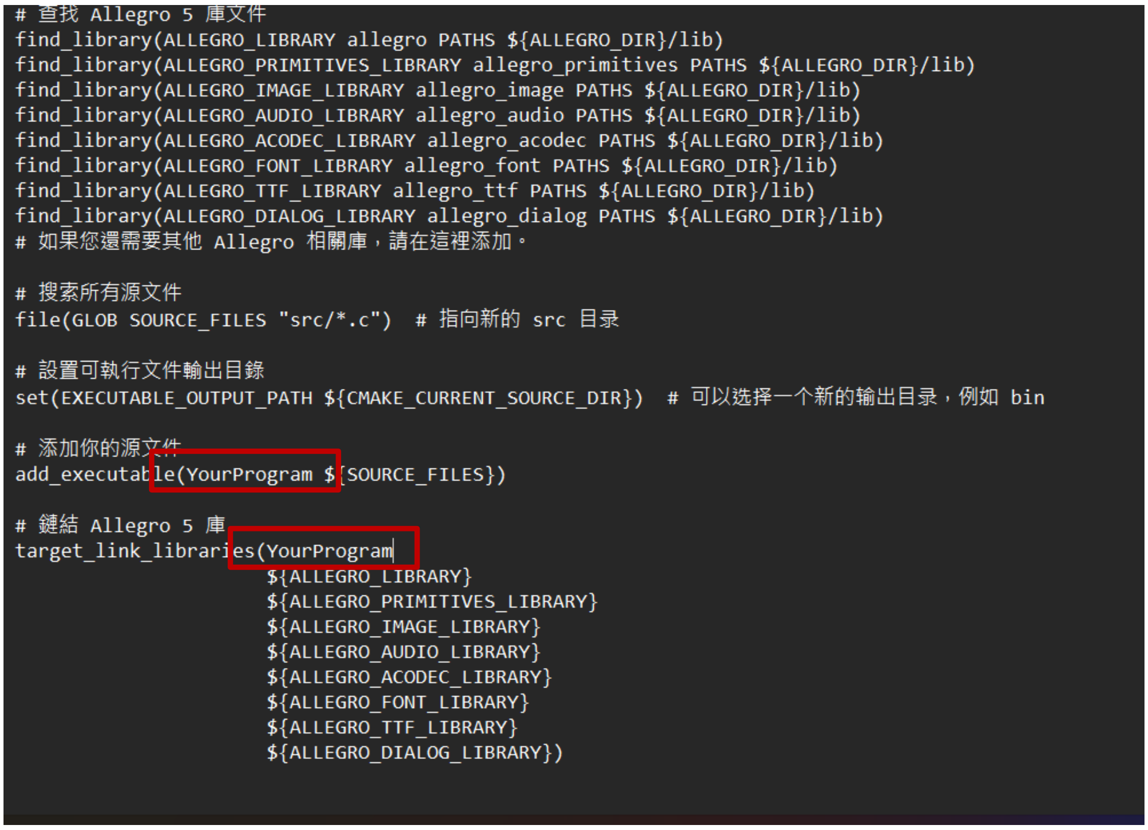
■ 新增路徑 D:\AllegroLib_5.2.8\bin 和 D:\AllegroLib_5.2.8\lib，完成後點選確定



編輯環境變數

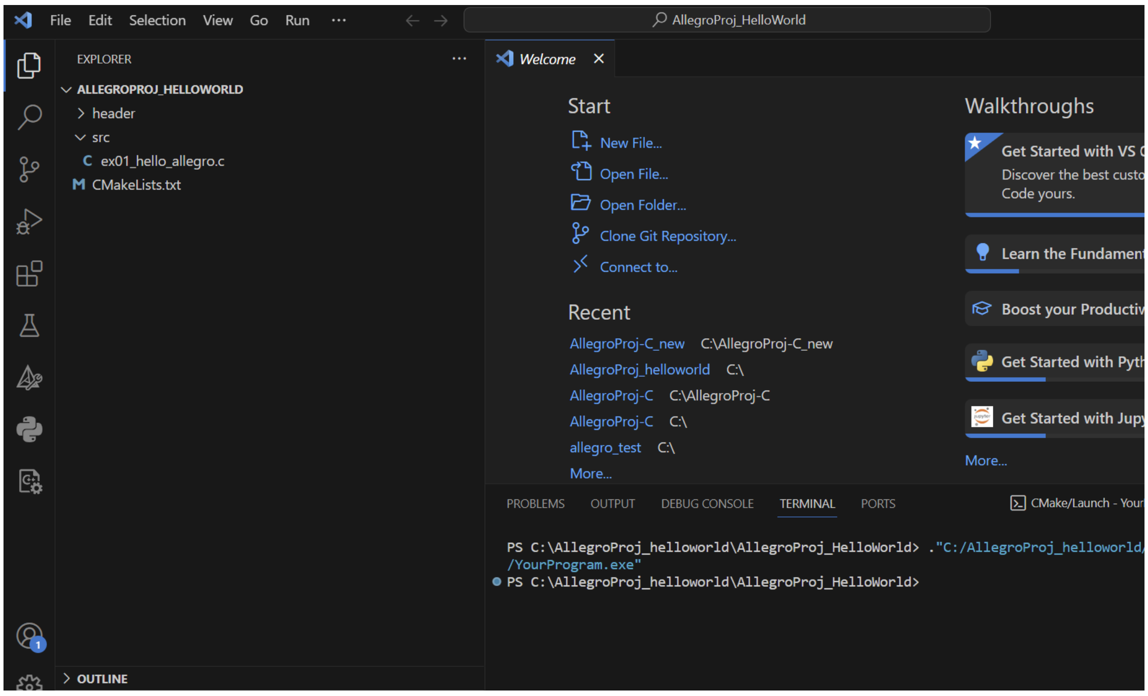
C:\Windows\System32\OpenSSH\
C:\Program Files (x86)\NVIDIA Corporation\PhysX\Common
C:\Program Files\NVIDIA Corporation\NVIDIA NvDLISR
C:\Program Files\dotnet\
C:\Program Files (x86)\Windows Kits\10\Windows Performance T...
C:\Program Files\PuTTY\
C:\Program Files\Lucid Vision Labs\Arena SDK\Win32Release
C:\Program Files\Lucid Vision Labs\Arena SDK\x64Release
C:\Program Files\Lucid Vision Labs\Arena SDK\Win32Debug
C:\Program Files\Lucid Vision Labs\Arena SDK\x64Debug
C:\Program Files\Git\cmd
C:\llvm-mingw-20230919-msvcrt-x86_64\bin
C:\mingw64\bin
D:\AllegroLib_5.2.8\bin
D:\AllegroLib_5.2.8\lib
C:\opencv\build\x64\vc16\bin
C:\Users\User\Downloads\2022_ShangBo_Thesis\2022_ShangBo_...
C:\opencv_built_by_gcc_on_-windows-master\opencv_built_by_gc...
C:\MinGW\bin
C:\Program Files\NVIDIA Corporation\Nsight Compute 2022.3.0\
C:\cudnn-windows-x86_64-8.9.6.50_cuda11-archive\cudnn-wind...

- 打開CMakelists，可自行更改其專案名稱(須兩個同樣名稱)



```
# 查找 Allegro 5 庫文件
find_library(ALLEGRO_LIBRARY allegro PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_PRIMITIVES_LIBRARY allegro_primitives PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_IMAGE_LIBRARY allegro_image PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_AUDIO_LIBRARY allegro_audio PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_ACODEC_LIBRARY allegro_acodec PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_FONT_LIBRARY allegro_font PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_TTF_LIBRARY allegro_ttf PATHS ${ALLEGRO_DIR}/lib)
find_library(ALLEGRO_DIALOG_LIBRARY allegro_dialog PATHS ${ALLEGRO_DIR}/lib)
# 如果您還需要其他 Allegro 相關庫，請在這裡添加。

# 搜索所有源文件
file(GLOB SOURCE_FILES "src/*.c")  # 指向新的 src 目录

# 設置可執行文件輸出目錄
set(EXECUTABLE_OUTPUT_PATH ${CMAKE_CURRENT_SOURCE_DIR})  # 可以选择一个新的输出目录，例如 bin

# 添加你的源文件
add_executable(YourProgram ${SOURCE_FILES})

# 鏈結 Allegro 5 庫
target_link_libraries(YourProgram
                      ${ALLEGRO_LIBRARY}
                      ${ALLEGRO_PRIMITIVES_LIBRARY}
                      ${ALLEGRO_IMAGE_LIBRARY}
                      ${ALLEGRO_AUDIO_LIBRARY}
                      ${ALLEGRO_ACODEC_LIBRARY}
                      ${ALLEGRO_FONT_LIBRARY}
                      ${ALLEGRO_TTF_LIBRARY}
                      ${ALLEGRO_DIALOG_LIBRARY})
```

- 打開AllegroProj_HelloWorld專案資料夾，其中包含header、src 和 CMakelist.txt



```
ALLEGROPROJ_HELLOWORLD
  > header
  ∨ src
      C ex01_hello_allegro.c
  M CMakeLists.txt
```

Get Started with VS C
Discover the best custo
Code yours.

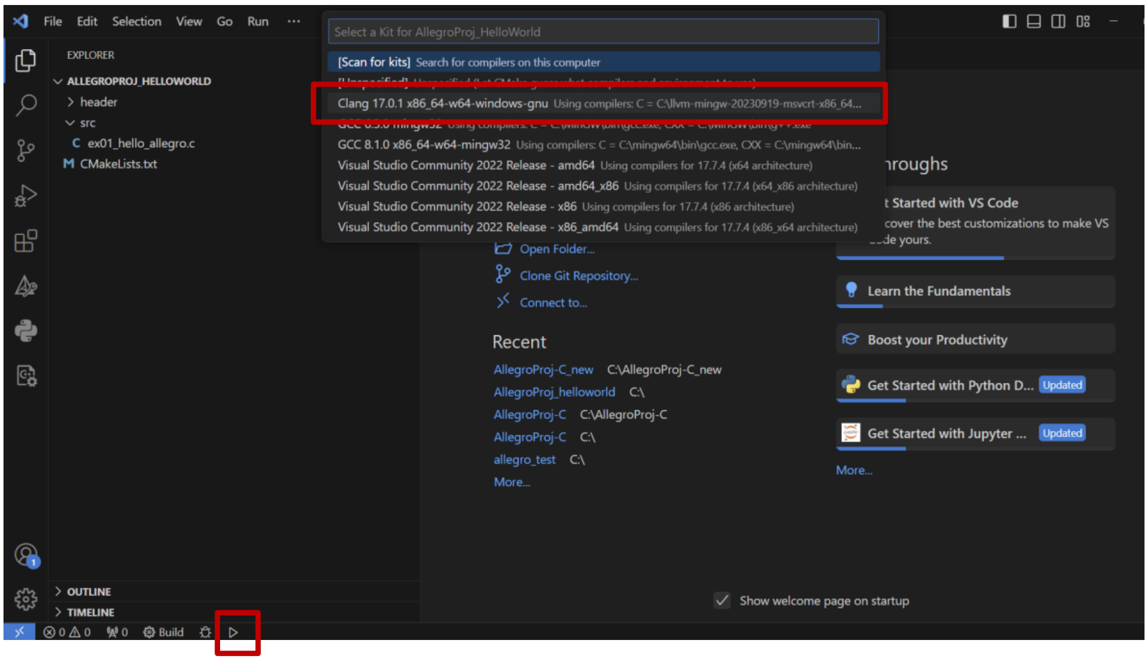
Recent

AllegroProj-C_new	C:\AllegroProj-C_new
AllegroProj_helloworld	C:\
AllegroProj-C	C:\AllegroProj-C
AllegroProj-C	C:\
allegro_test	C:\

More...

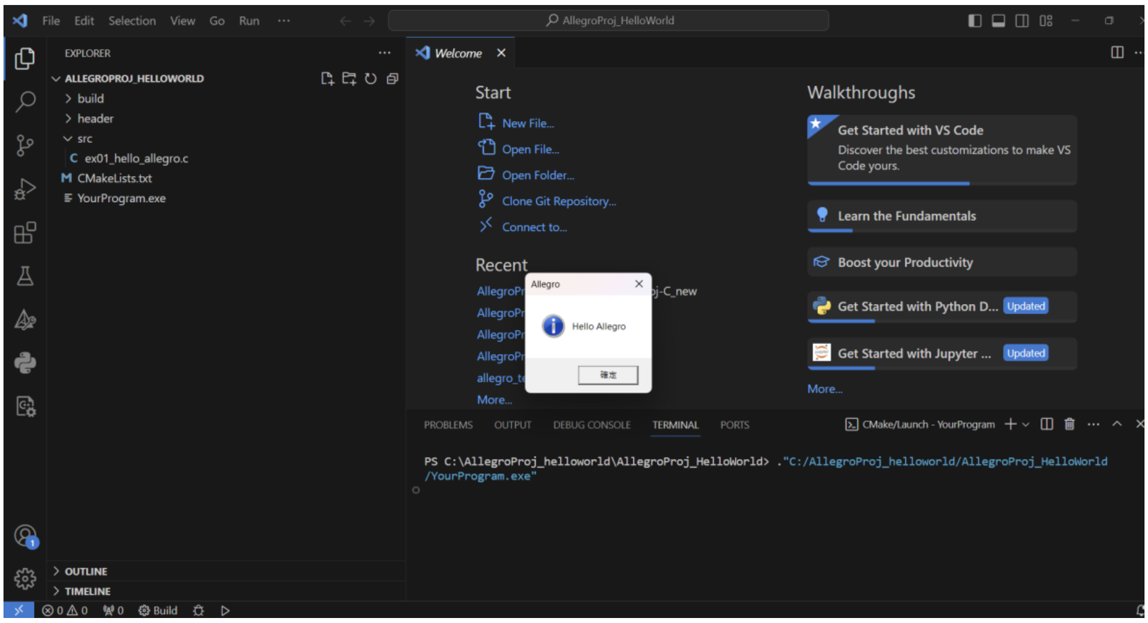
```
PS C:\AllegroProj_helloworld\AllegroProj_HelloWorld> ."C:/AllegroProj_helloworld/
/YourProgram.exe"
PS C:\AllegroProj_helloworld\AllegroProj_HelloWorld>
```

■ 點選底下三角形圖示(Cmake Tools)，選擇64位元編譯器 Clang 17.0.1



```
Select a Kit for AllegroProj_HelloWorld
[Scan for kits] Search for compilers on this computer
Clang 17.0.1 x86_64-w64-windows-gnu Using compilers: C = C:\llvm-mingw-20230919-msvcrt-x86_64...
GCC 8.1.0 x86_64-w64-mingw32 Using compilers: C = C:\mingw64\bin\gcc.exe, CXX = C:\mingw64\bin...
Visual Studio Community 2022 Release - amd64 Using compilers for 17.7.4 (x64 architecture)
Visual Studio Community 2022 Release - amd64_x86 Using compilers for 17.7.4 (x64_x86 architecture)
Visual Studio Community 2022 Release - x86 Using compilers for 17.7.4 (x86 architecture)
Visual Studio Community 2022 Release - x86_amd64 Using compilers for 17.7.4 (x86_x64 architecture)
```

■ 生成 build 資料夾與對應的 makefile 項目，並生成YourPrograme.exe，顯示 Hello Allegro視窗



```
PS C:\AllegroProj_helloworld\AllegroProj_HelloWorld> ."C:/AllegroProj_helloworld/AllegroProj_HelloWorld
/YourProgram.exe"
```