1. Software environment configuration

Build the development environment of Raspberry Pi Pico 2 under Windows

Preparation

ARM GCC compiler.exe file

CMake download msi file

Build Tools for Visual Studio 2019

Python 3 download exe (Windows installer)

Git download Git for Windows downloads

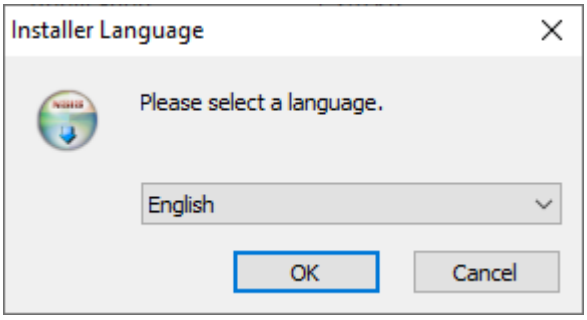
Download the executable installers of these five packages, and then install all five packages on Windows

Install ARM GCC COMPLIER

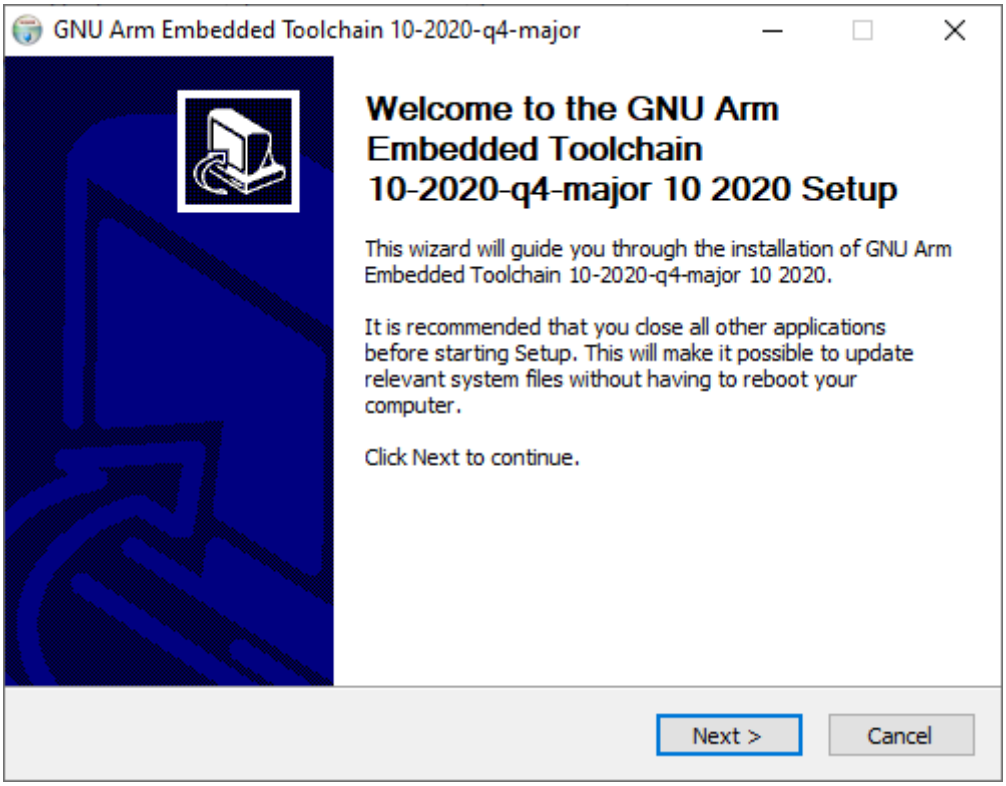
You can download the version you need from the ARM GCC complier official website, or get it from the attachment of the document

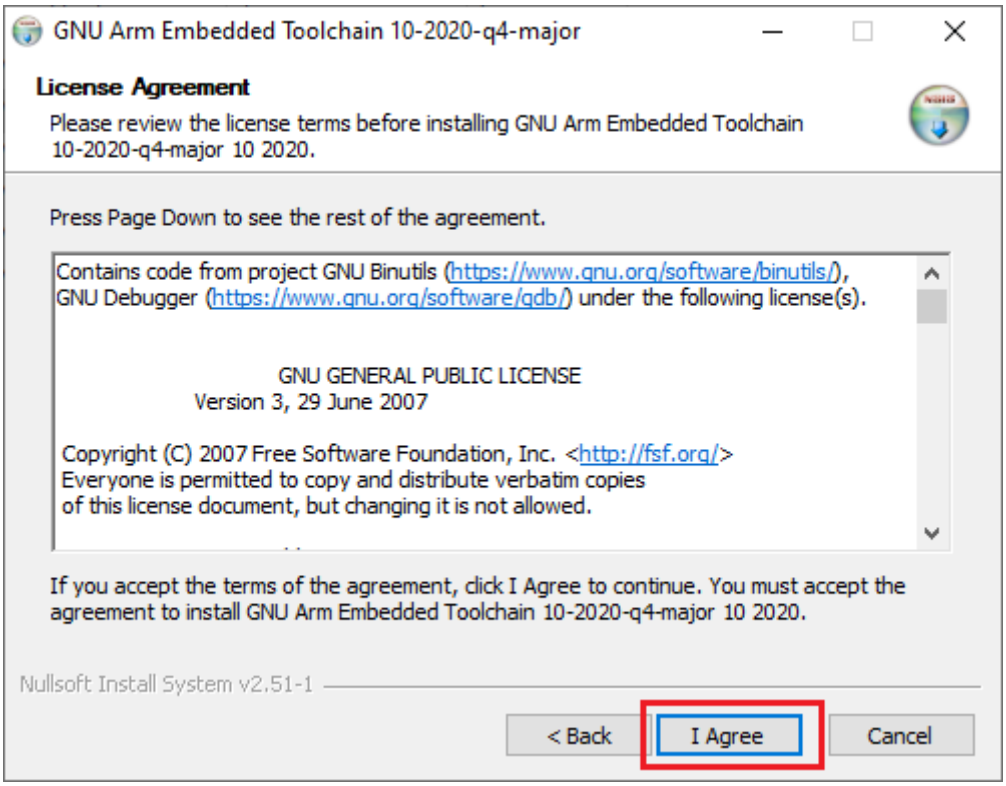
https://developer.arm.com/tools-and-software/open-source-software/developer-tools/gnu-toolchain/gnu-rm/downloads

Select the installation language as Chinese

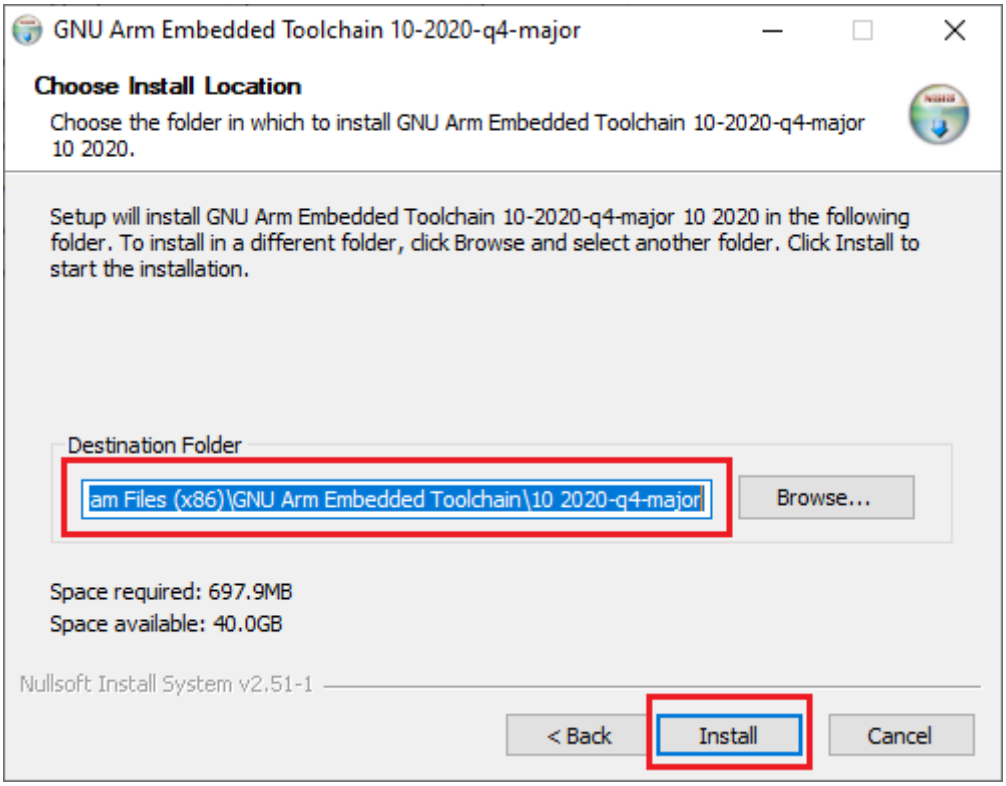


Follow the prompts to install

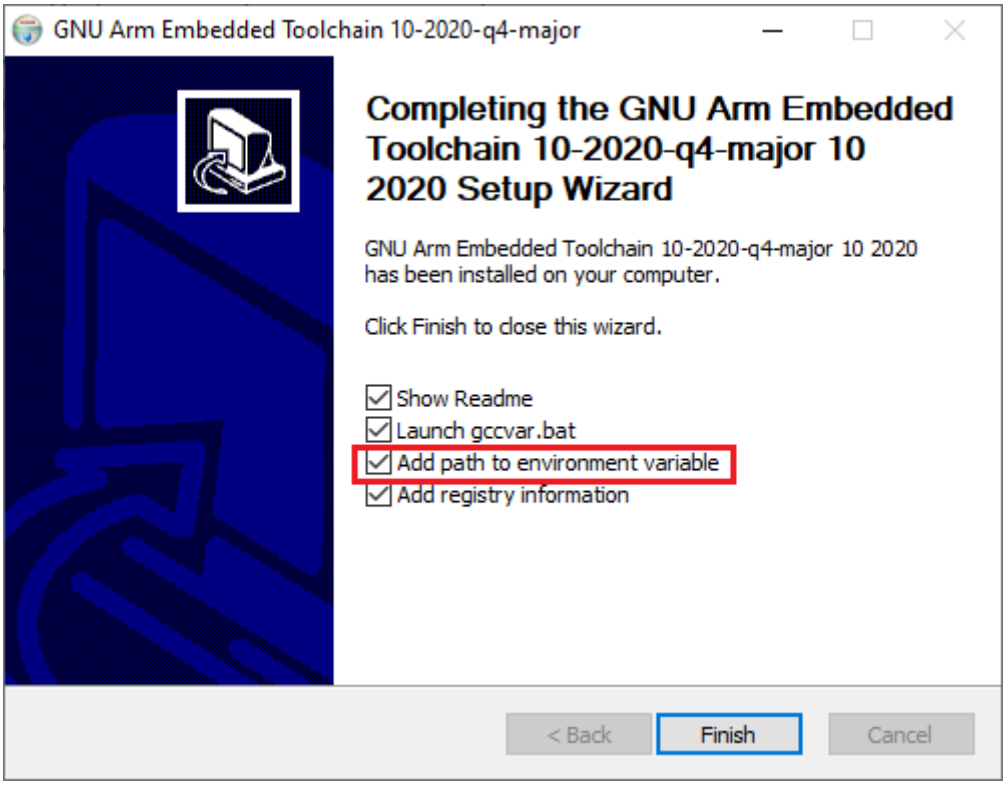




Select the installation path



During the installation process, when prompted, check the box to request the path to the ARM compiler as an environment variable in the Windows shell

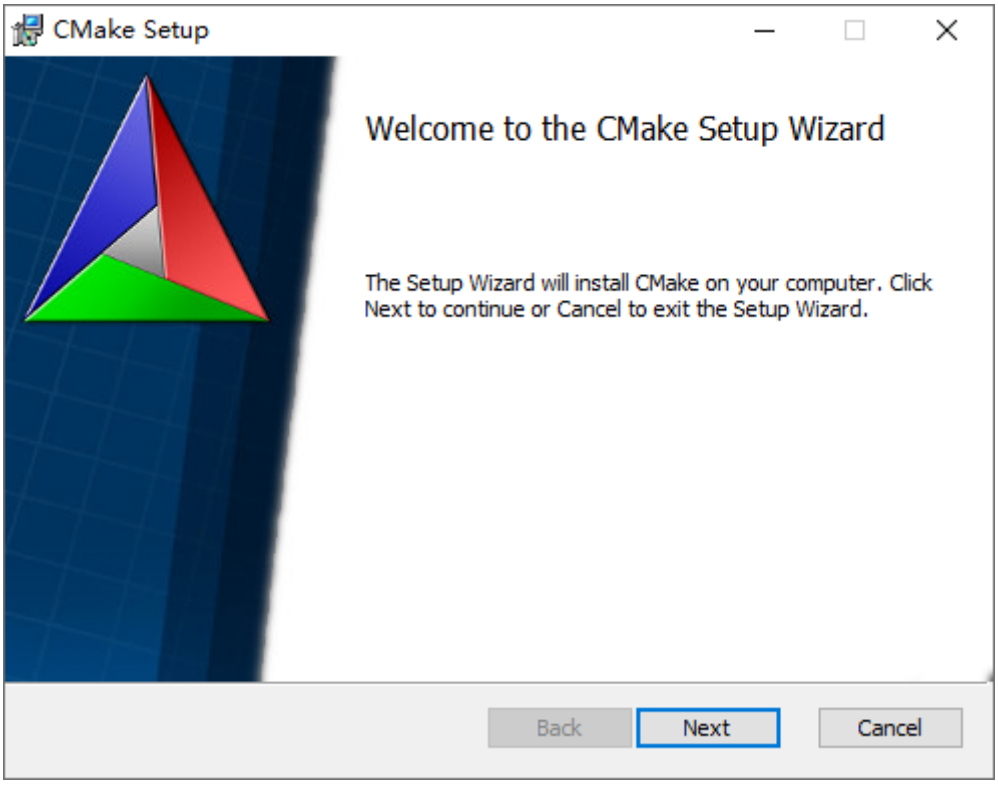


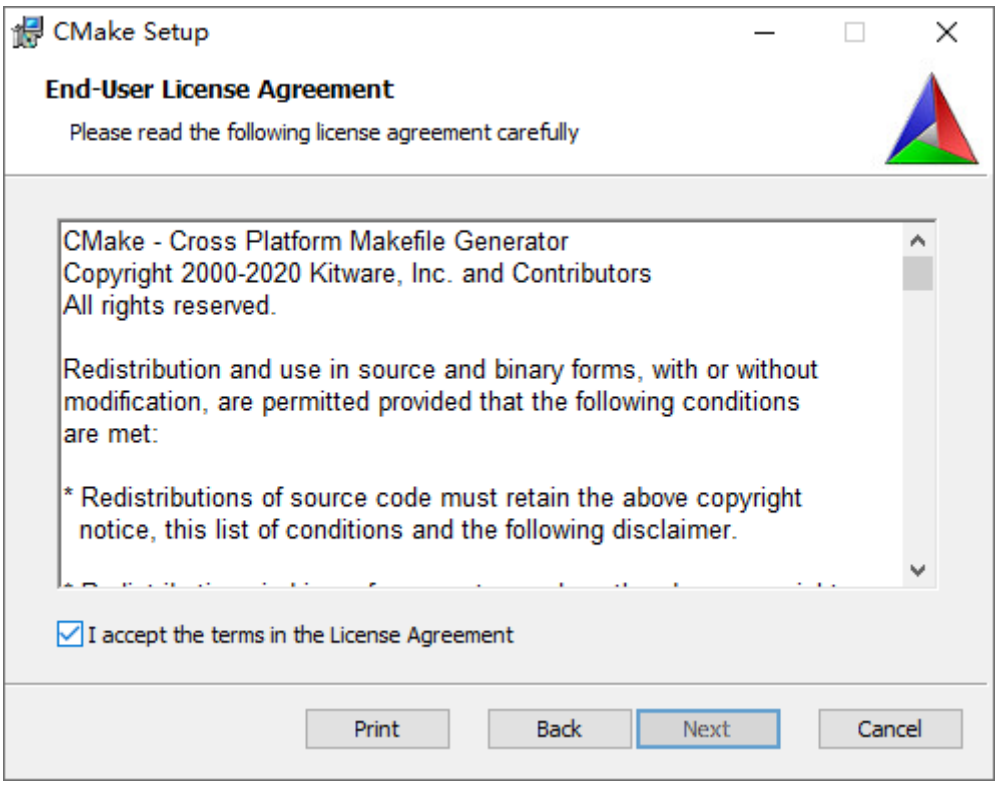
Install CMAKE

cmake official website: https://cmake.org/download/

You can also get the installation package from the attachment of the document

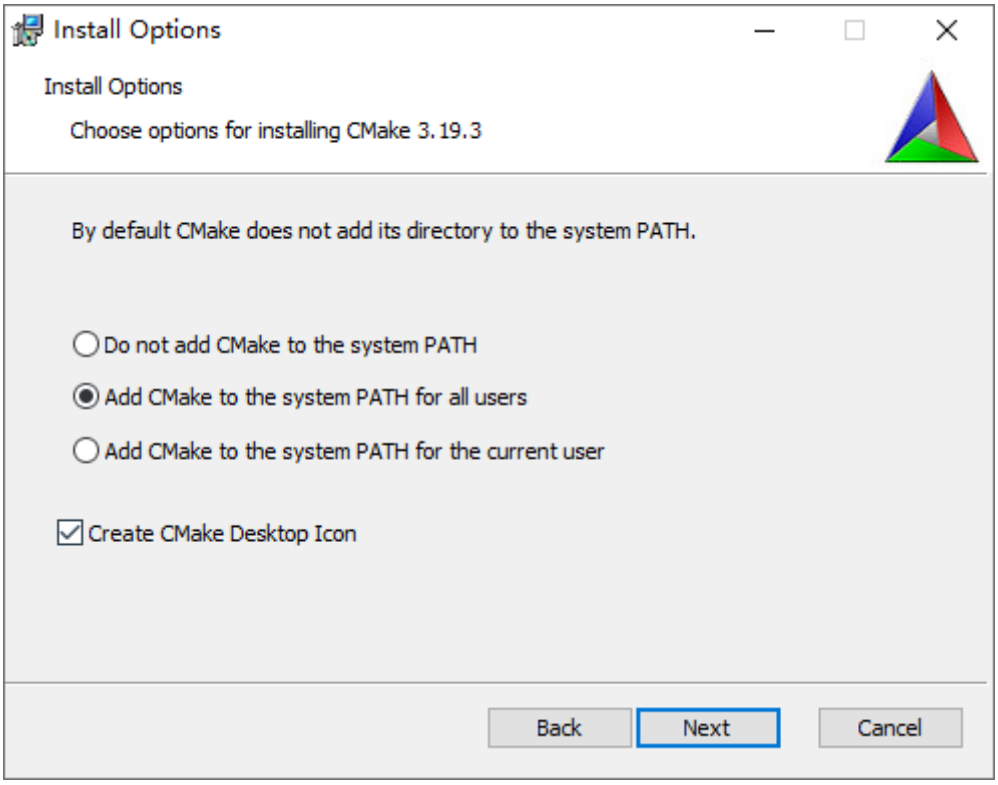
Click Next according to the prompt. It is recommended to turn off the antivirus software before installation



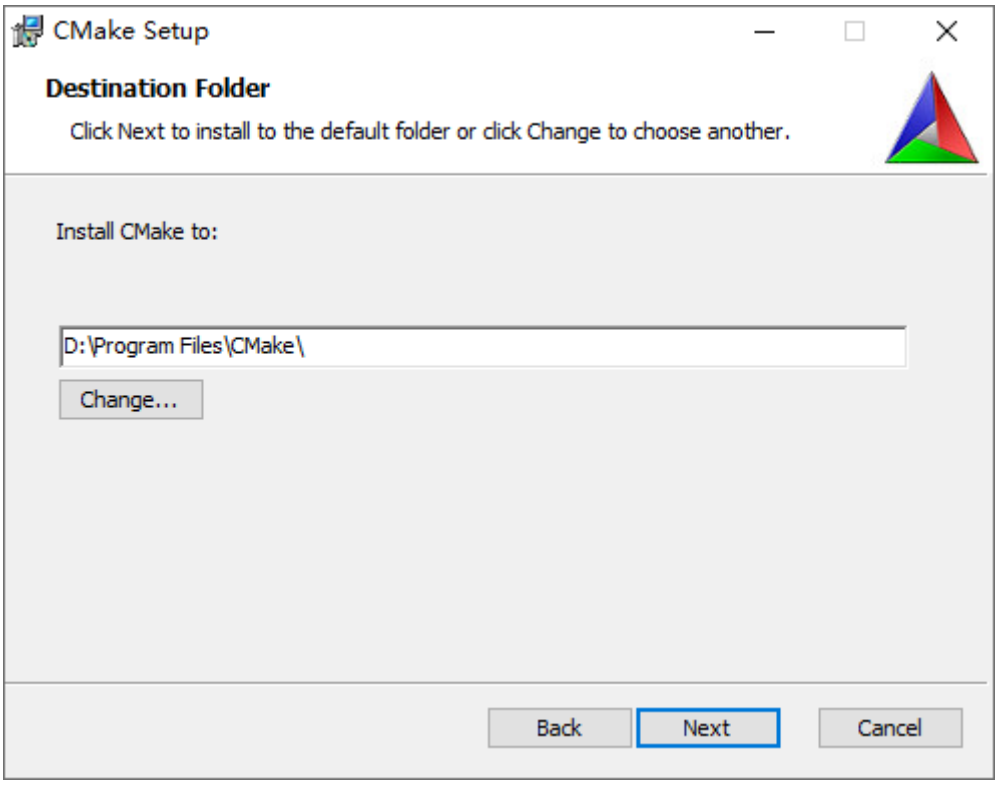


During the installation process, the installer will prompt to add CMake to the system path of all users

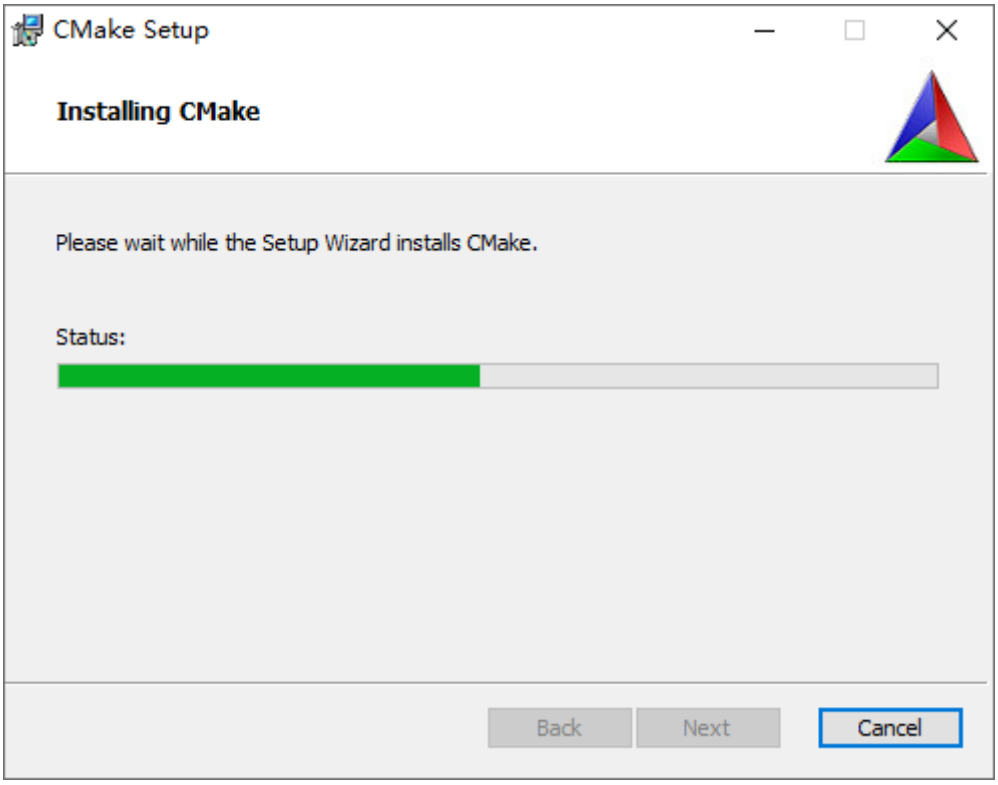
Check 1 and 2



Select the installation path

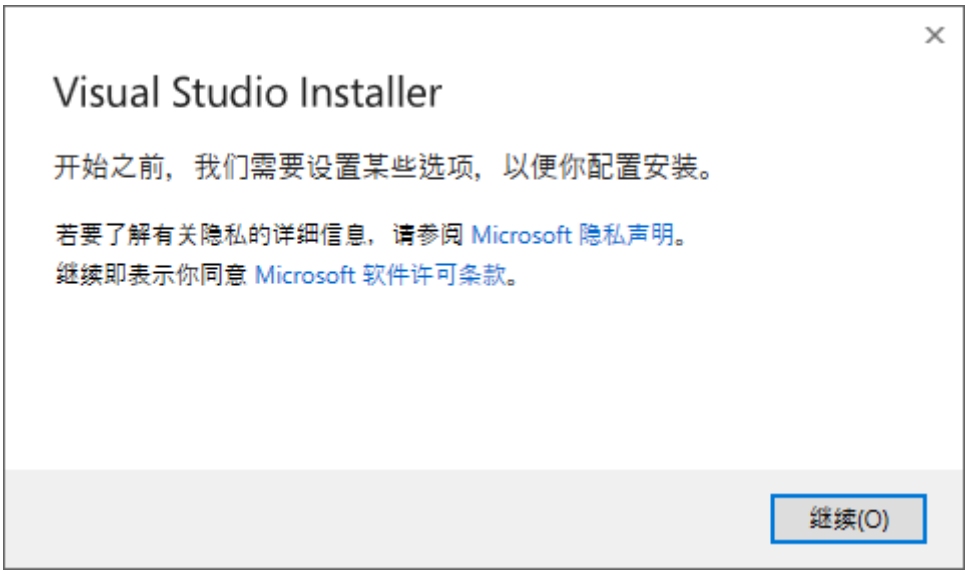


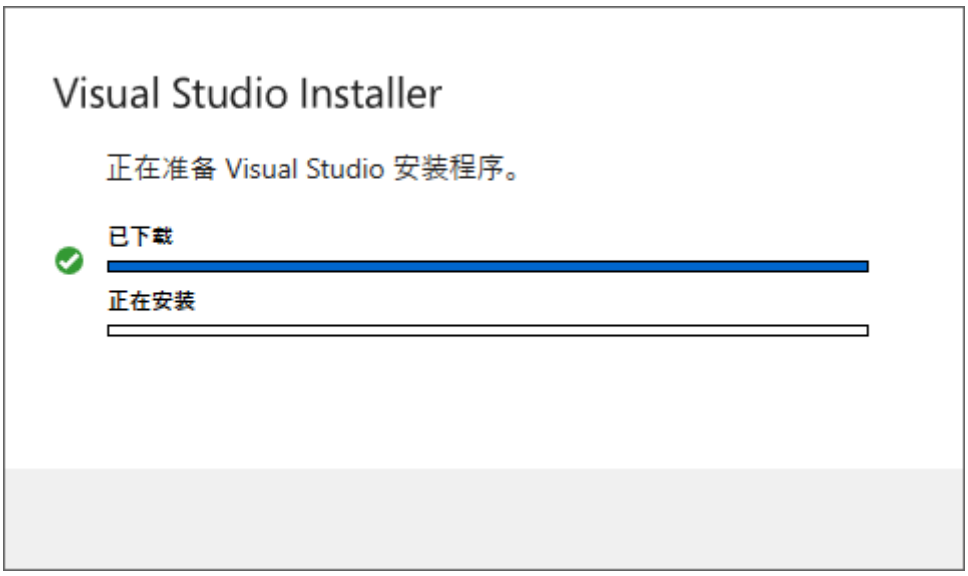
Installation completed, restart to take effect

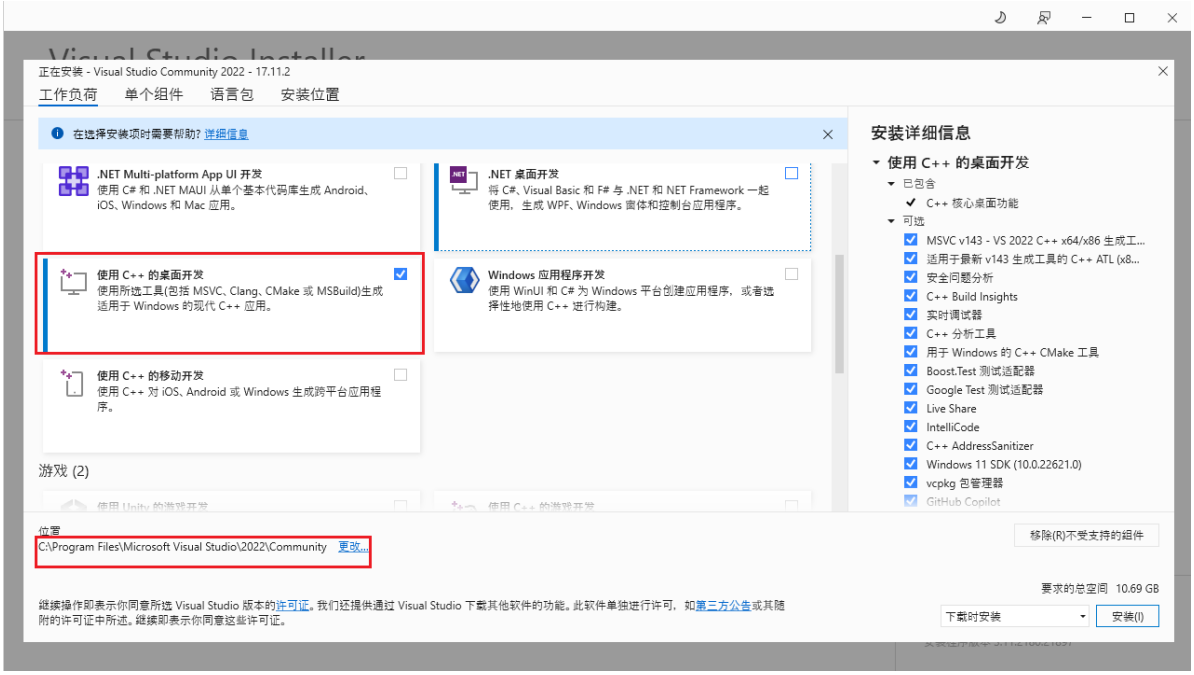


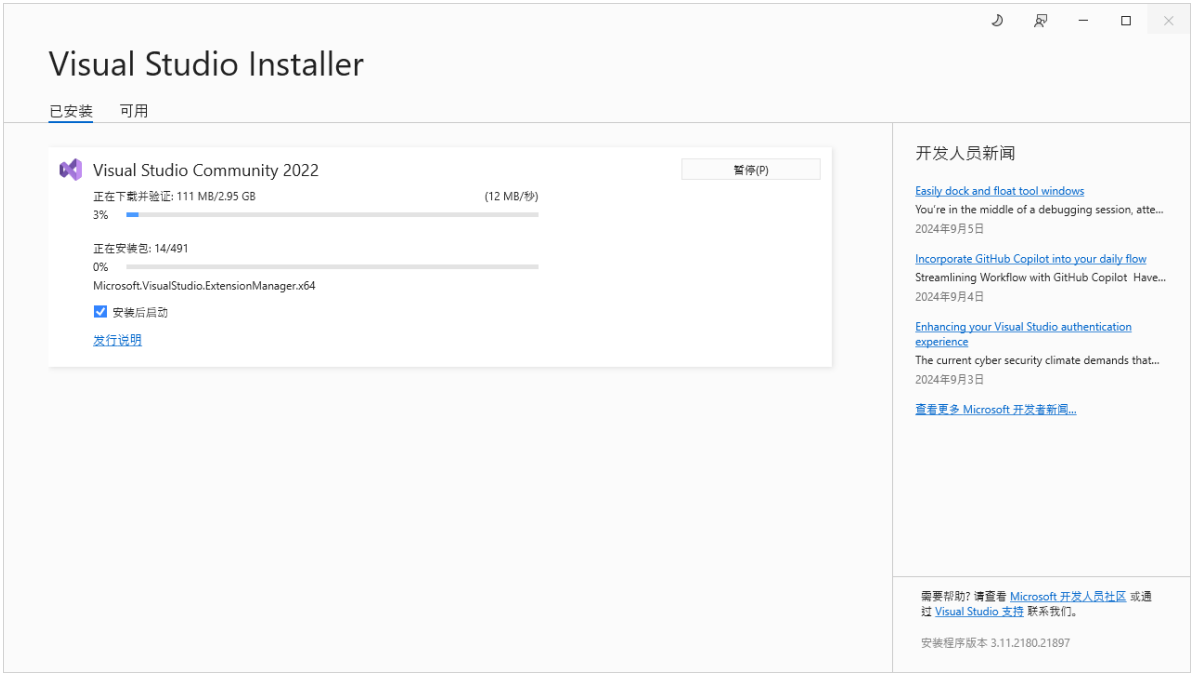
Install Visual Studio

Visual Studio official website: https://visualstudio.microsoft.com/zh-hans/downloads/, you can also get the installation software in the attachment of the document, here take VS2022 as an example.




Visual Studio Installer
正在准备 Visual Studio 安装程序。
已下载
正在安装

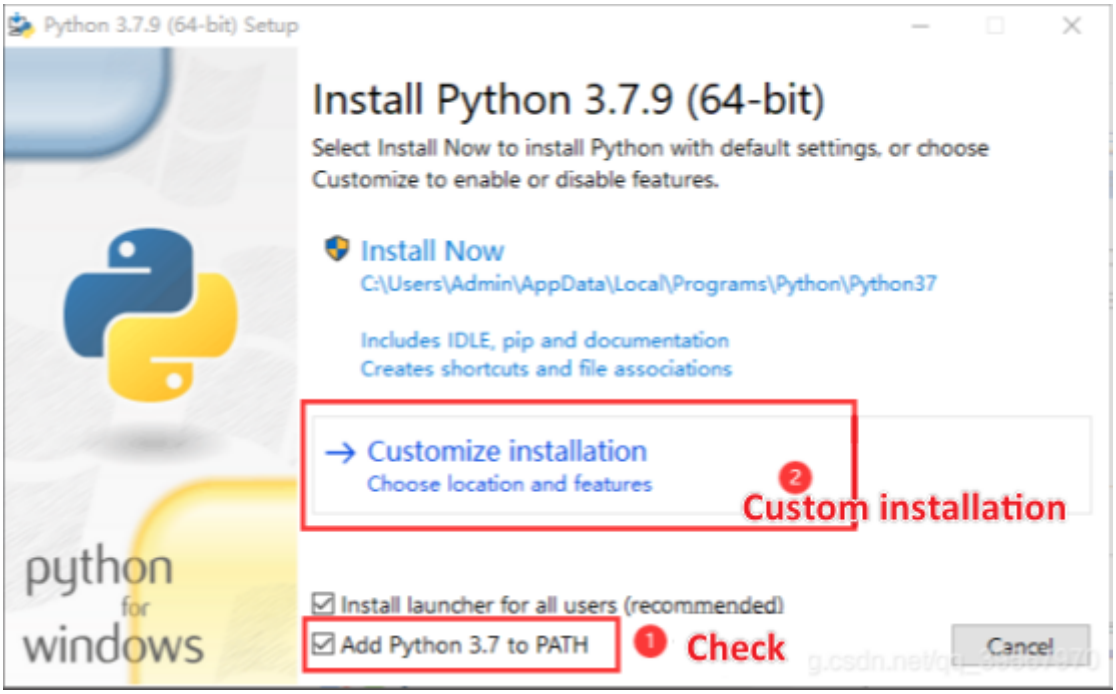

正在安装 - Visual Studio Community 2022 - 17.11.2
工作负荷
单个组件
语言包
安装位置
在选择安装项时需要帮助? 详细信息
.NET Multi-platform App UI 开发
使用 C# 和 .NET MAUI 从单个基本代码库生成 Android、iOS、Windows 和 Mac 应用。
.NET 桌面开发
将 C#、Visual Basic 和 F# 与 .NET 和 .NET Framework 一起使用，生成 WPF、Windows 窗体和控制台应用程序。
使用 C++ 的桌面开发
使用所选工具(包括 MSVC、Clang、CMake 或 MSBuild)生成适用于 Windows 的现代 C++ 应用。
Windows 应用程序开发
使用 WinUI 和 C# 为 Windows 平台创建应用程序，或者选择性地使用 C++ 进行构建。
使用 C++ 的移动开发
使用 C++ 对 iOS、Android 或 Windows 生成跨平台应用程序。
游戏 (2)
安装详细信息
使用 C++ 的桌面开发
已包含
C++ 核心桌面功能
可选
MSVC v143 - VS 2022 C++ x64/x86 生成工...
适用于最新 v143 生成工具的 C++ ATL (x8...
安全问题分析
C++ Build Insights
实时调试器
C++ 分析工具
用于 Windows 的 C++ CMake 工具
Boost.Test 测试适配器
Google Test 测试适配器
Live Share
IntelliCode
C++ AddressSanitizer
Windows 11 SDK (10.0.22621.0)
vcpkg 包管理器
GitHub Copilot
位置
C:\Program Files\Microsoft Visual Studio\2022\Community
更改...
移除(R)不受支持的组件
要求的总空间 10.69 GB
继续操作即表示你同意所选 Visual Studio 版本的许可证。我们还提供通过 Visual Studio 下载其他软件的功能。此软件单独进行许可，如第三方公告或其随附的许可证中所述。继续即表示你同意这些许可证。
下载时安装
安装(I)

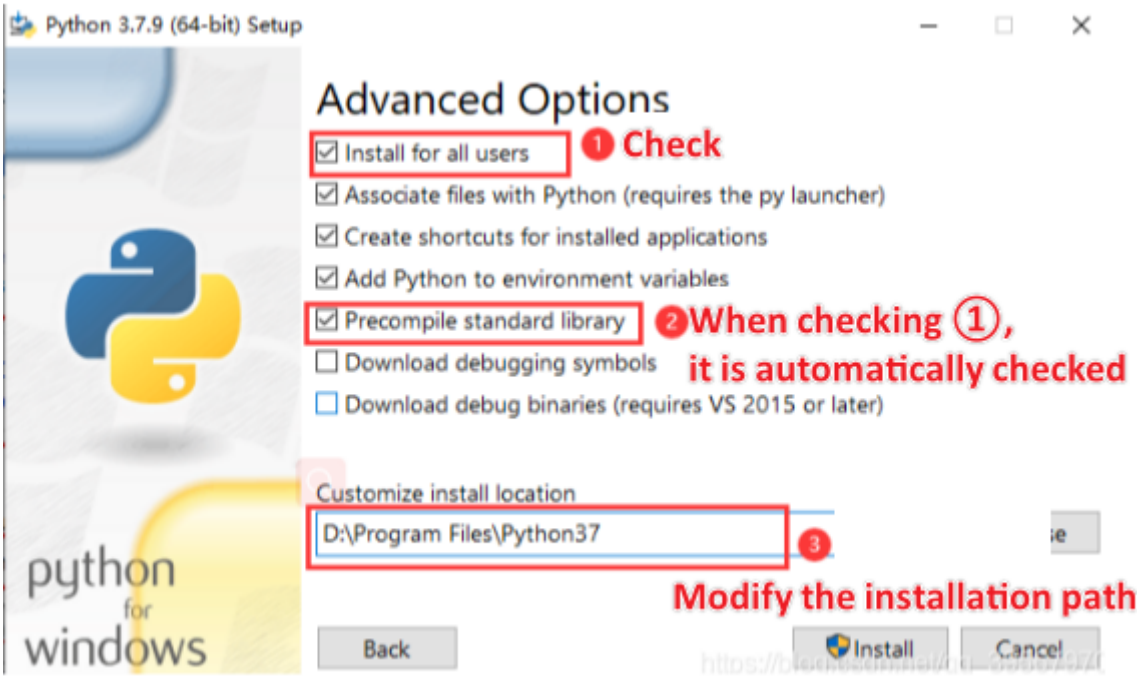

Visual Studio Installer
已安装
可用
Visual Studio Community 2022
暂停(P)
正在下载并验证: 111 MB/2.95 GB
(12 MB/秒)
3%
正在安装包: 14/491
0%
Microsoft.VisualStudio.ExtensionManager.x64
安装后启动
发行说明
开发人员新闻
Easily dock and float tool windows
You're in the middle of a debugging session, atte...
2024年9月5日
Incorporate GitHub Copilot into your daily flow
Streamlining Workflow with GitHub Copilot Have...
2024年9月4日
Enhancing your Visual Studio authentication experience
The current cyber security climate demands that...
2024年9月3日
查看更多 Microsoft 开发者新闻...
需要帮助? 请查看 Microsoft 开发人员社区 或通过 Visual Studio 支持 联系我们。
安装程序版本 3.11.2180.21897

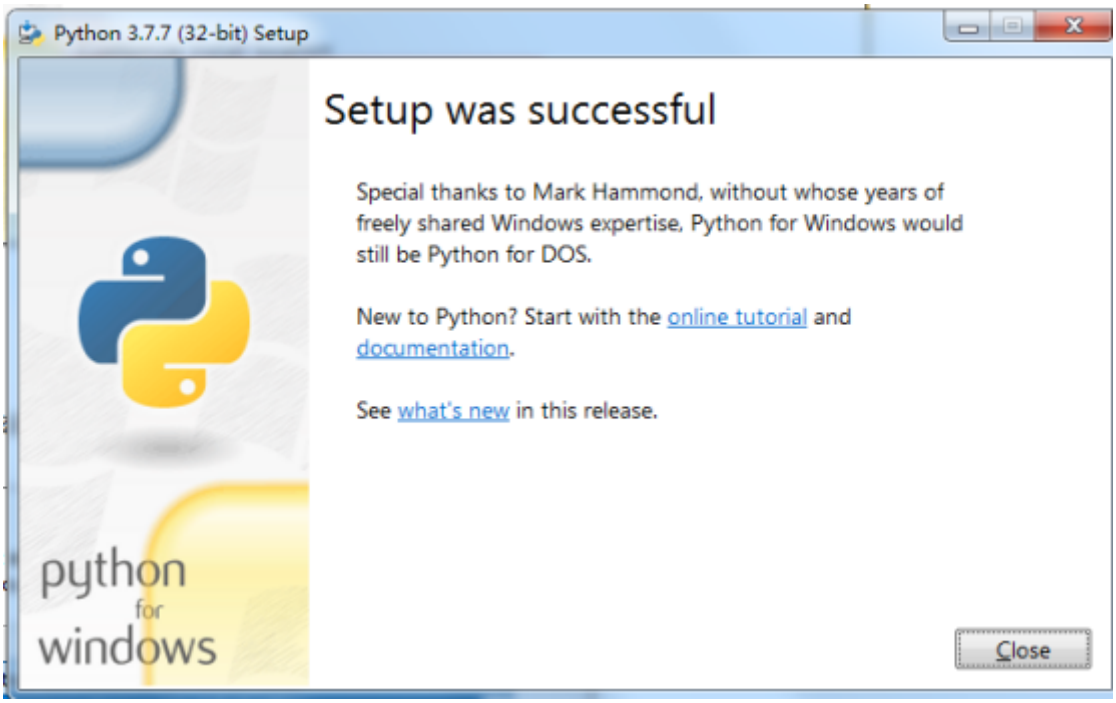
INSTALL PYTHON3

During the installation process, follow the prompts of the installer to add Python 3.7 to the system path for all users.

Additionally, when installing Python, select Customize installation, click Optional Features, and then select Install for all users under Advanced Features. You should also disable the maximum path length when prompted at the end of the Python installation.

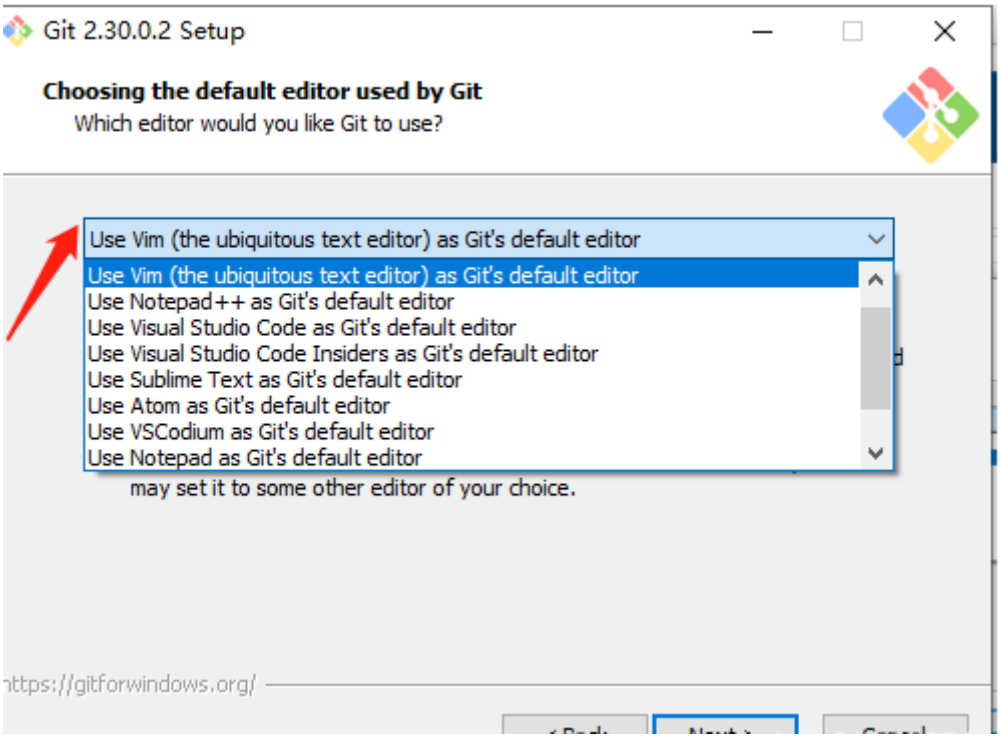






Install GIT

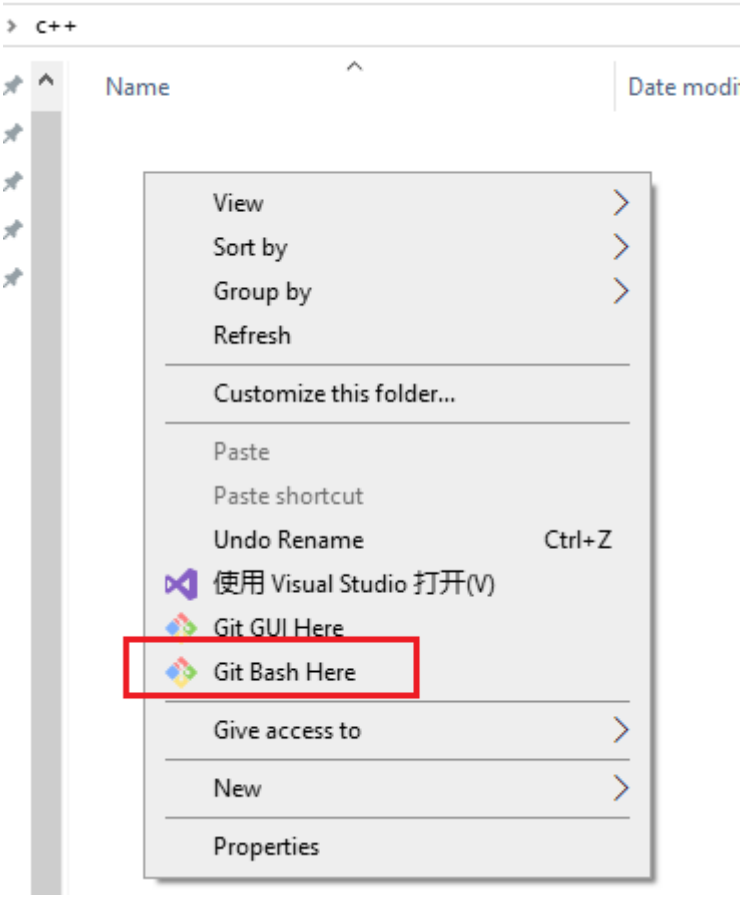
When installing Git, you should make sure to change the default editor from vim, see the image below. The default configuration is fine



Get the SDK and examples for Pico 2

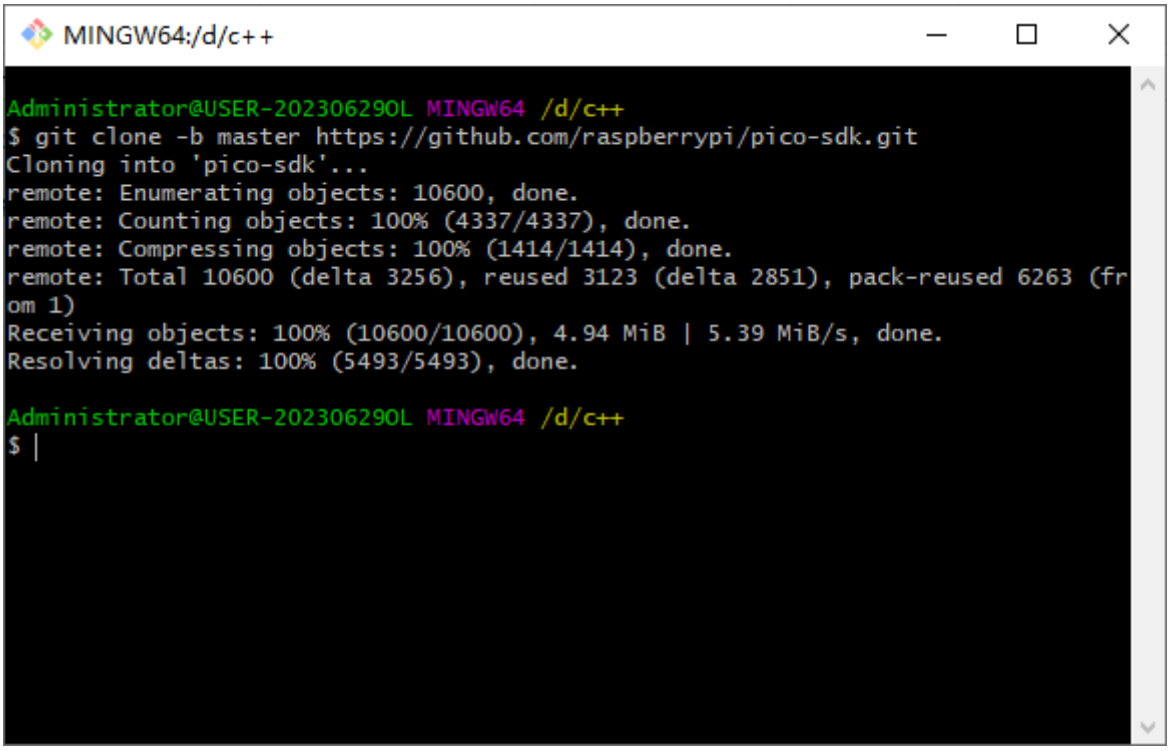
The following source code can be obtained in the provided materials.

Create a folder, right-click it and select Git bash Here



In the command box that appears, enter

```
git clone -b master https://github.com/raspberrypi/pico-sdk.git
cd pico-sdk
git submodule update --init
cd ..
git clone -b master https://github.com/raspberrypi/pico-examples.git
```



Open the pico-sdk folder and proceed to the next step (this step takes a long time)

```
remote: Counting objects: 100% (4347/4347), done.
remote: Compressing objects: 100% (1415/1415), done.
remote: Total 10600 (delta 3212), reused 3131 (delta 2860), pack-reused 6253 (fr
om 1)
Receiving objects: 100% (10600/10600), 4.79 MiB | 167.00 KiB/s, done.
Resolving deltas: 100% (5489/5489), done.

Administrator@USER-20230629OL MINGW64 /d/c++
$ cd pico-sdk

Administrator@USER-20230629OL MINGW64 /d/c++/pico-sdk (master)
$ git submodule update --init
Submodule 'lib/btstack' (https://github.com/bluekitchen/btstack.git) registered
for path 'lib/btstack'
Submodule 'lib/cyw43-driver' (https://github.com/georgerobotics/cyw43-driver.git
) registered for path 'lib/cyw43-driver'
Submodule 'lib/lwip' (https://github.com/lwip-tcpip/lwip.git) registered for pat
h 'lib/lwip'
Submodule 'lib/mbedtls' (https://github.com/Mbed-TLS/mbedtls.git) registered for
 path 'lib/mbedtls'
Submodule 'tinyusb' (https://github.com/hathach/tinyusb.git) registered for path
 'lib/tinyusb'
Cloning into 'D:/c++/pico-sdk/lib/btstack'...
```

Installation routine

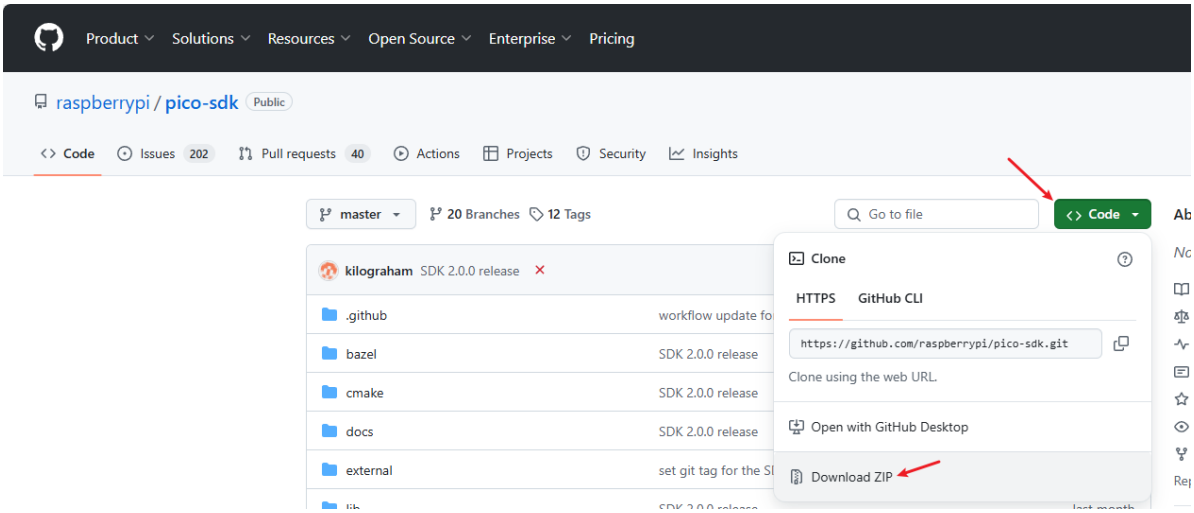
```
Administrator@USER-20230629OL MINGW64 /d/c++/pico-sdk (master)
$ git clone -b master https://github.com/raspberrypi/pico-examples.git
Cloning into 'pico-examples'...
remote: Enumerating objects: 3379, done.
remote: Counting objects: 100% (1195/1195), done.
remote: Compressing objects: 100% (635/635), done.
remote: Total 3379 (delta 701), reused 854 (delta 557), pack-reused 2184 (from 1
)
Receiving objects: 100% (3379/3379), 8.32 MiB | 7.34 MiB/s, done.
Resolving deltas: 100% (1679/1679), done.

Administrator@USER-20230629OL MINGW64 /d/c++/pico-sdk (master)
$
```

Extension: You can download it manually from Github. The link is as follows (this method is said to be feasible online, but it has not been verified and is only for reference)

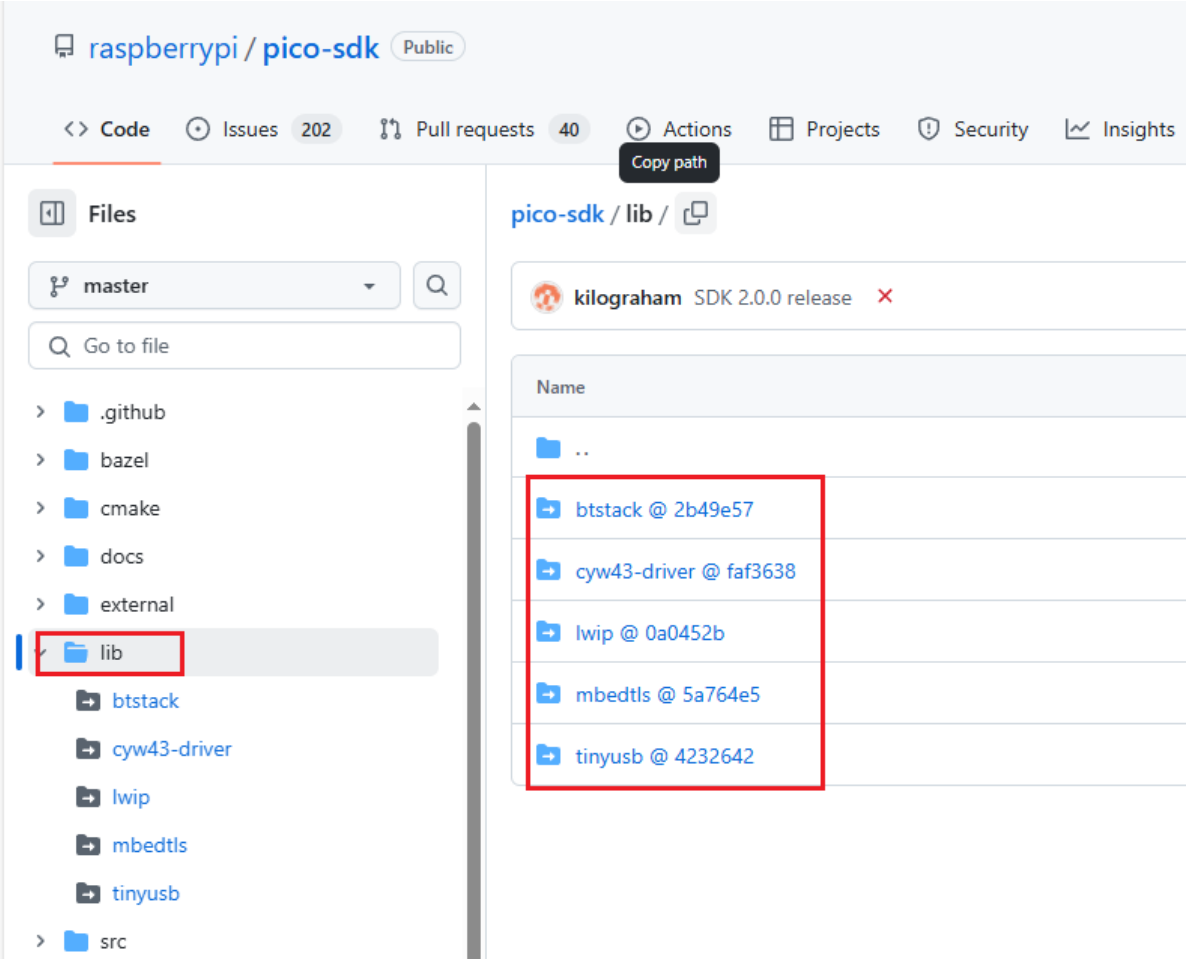
https://github.com/raspberrypi/pico-sdk

https://github.com/raspberrypi/pico-examples



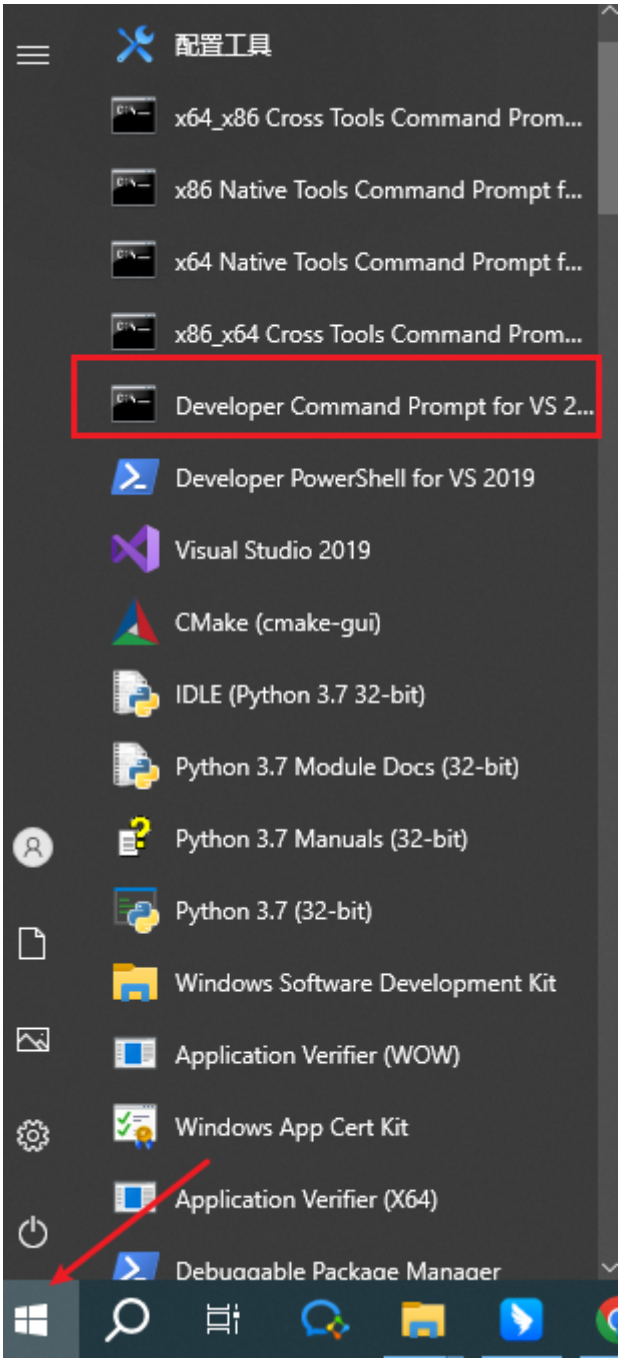
After downloading and unzipping, delete the -master after the folder name

Note that the previous git submodule update The --init command clones a tinyusb file to the lib path of the SDK, so if you download it yourself, you also need to download tinyusb and put it in this path.

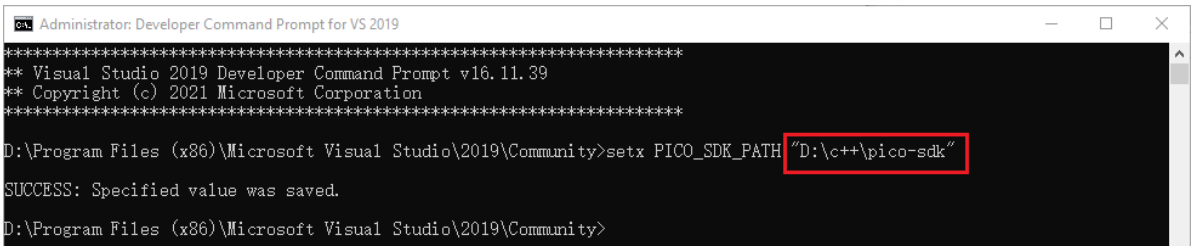


Raspberry Pi Pico 2 Windows Development Environment - Compile and Build Projects in Command Line

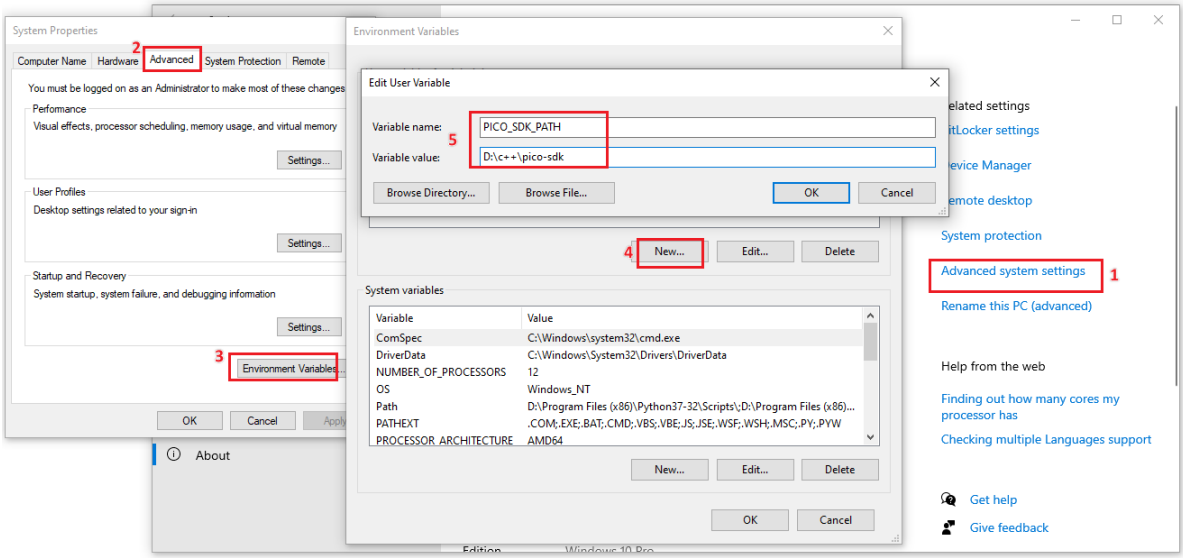
Configure SDK Path
Open a Developer Command Prompt window from the Windows menu by selecting Windows > Visual Studio
2022 > Developer Command Prompt Menu.



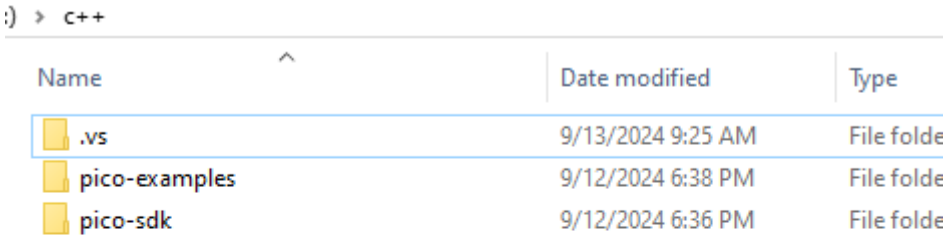
Then set the SDK path as follows: setx PICO_SDK_PATH "...pico-sdk" Close the window after successful setting

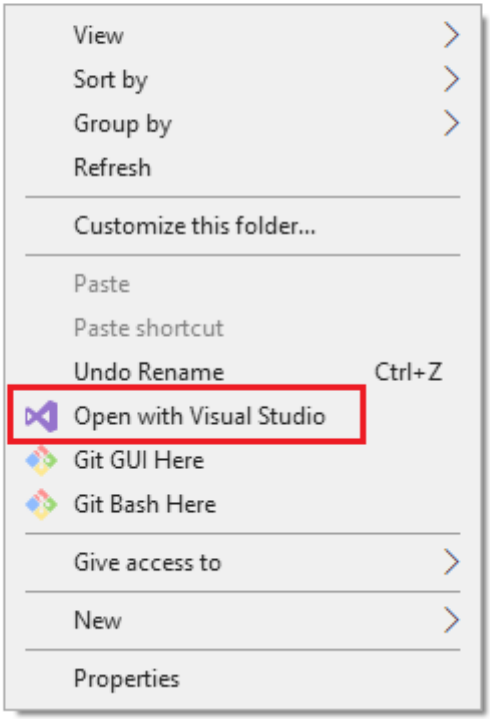


You can also set environment variables through this step

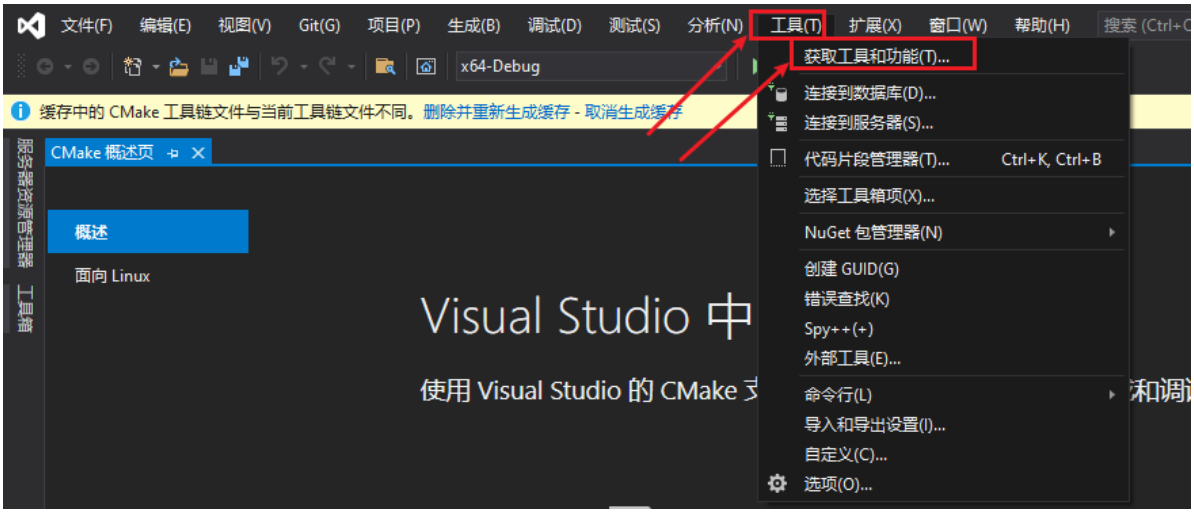


Use Visual Studio 2022 in the path where Pico SDK is saved Open

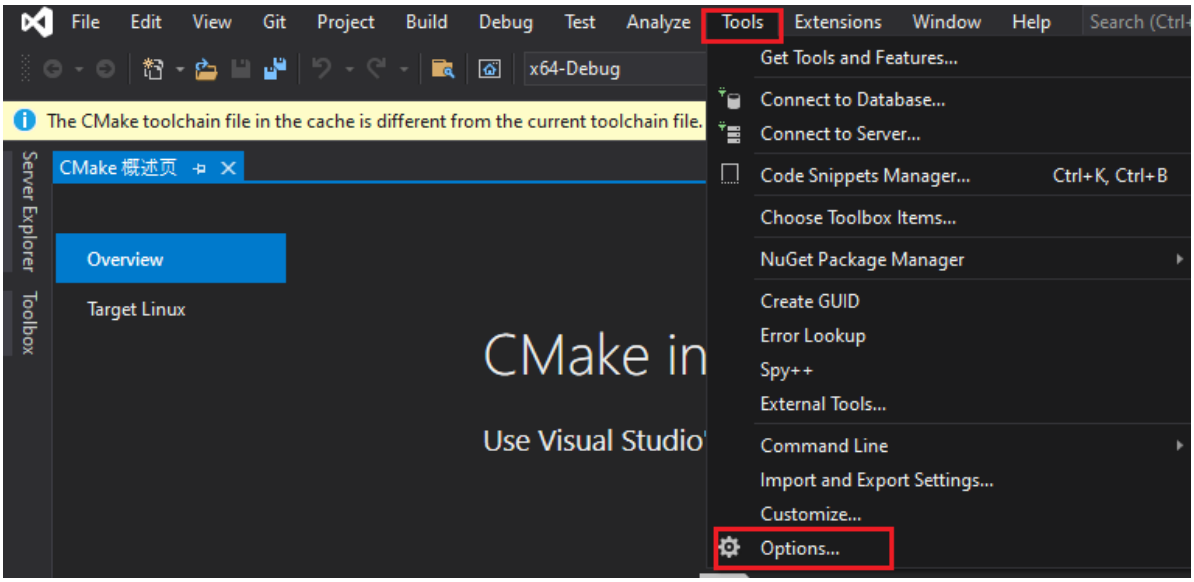


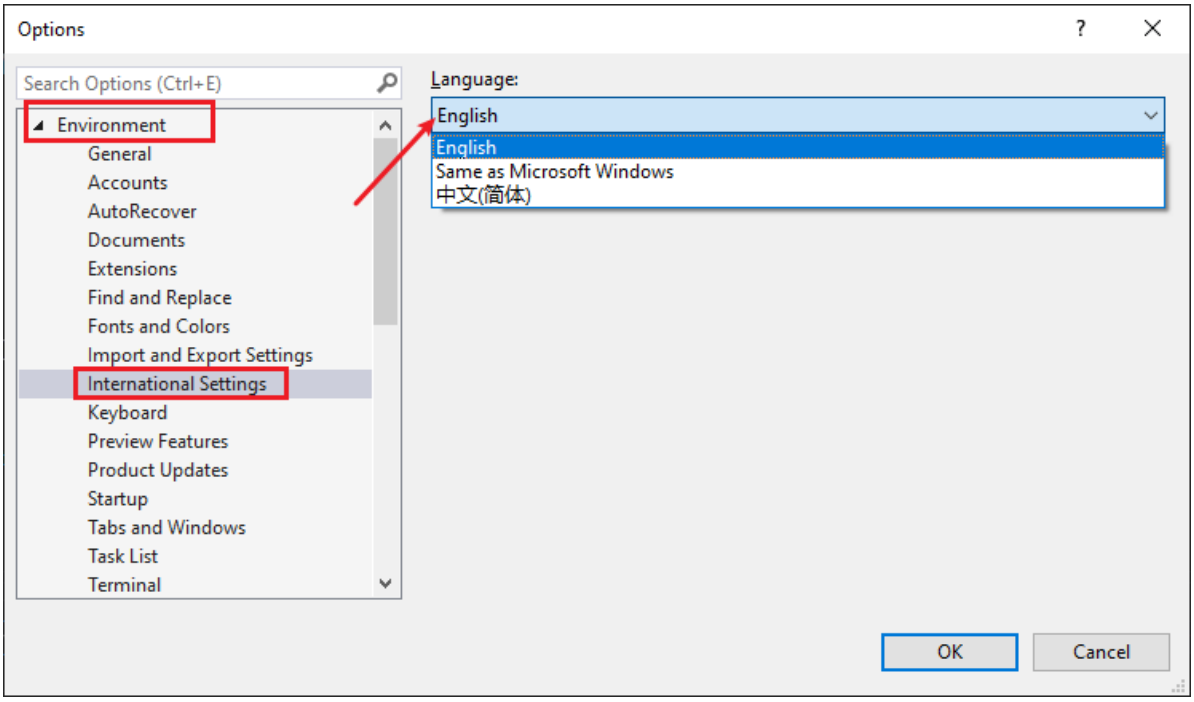


You can go to Tools(工具)->Get tools and features(获取工具和功能)->Language pack(语言包)->Select English(英语), then click Modify(修改).


文件(F) 编辑(E) 视图(V) Git(G) 项目(P) 生成(B) 调试(D) 测试(S) 分析(N) 工具(T) 扩展(X) 窗口(W) 帮助(H)
x64-Debug
缓存中的 CMake 工具链文件与当前工具链文件不同。删除并重新生成缓存 - 取消生成缓存
CMake 概述页
概述
面向 Linux
Visual Studio 中
使用 Visual Studio 的 CMake
获取工具和功能(T)...
连接到数据库(D)...
连接到服务器(S)...
代码片段管理器(T)... Ctrl+K, Ctrl+B
选择工具箱项(X)...
NuGet 包管理器(N)
创建 GUID(G)
错误查找(K)
Spy++(+)
外部工具(E)...
命令行(L)
导入和导出设置(I)...
自定义(C)...
选项(O)...

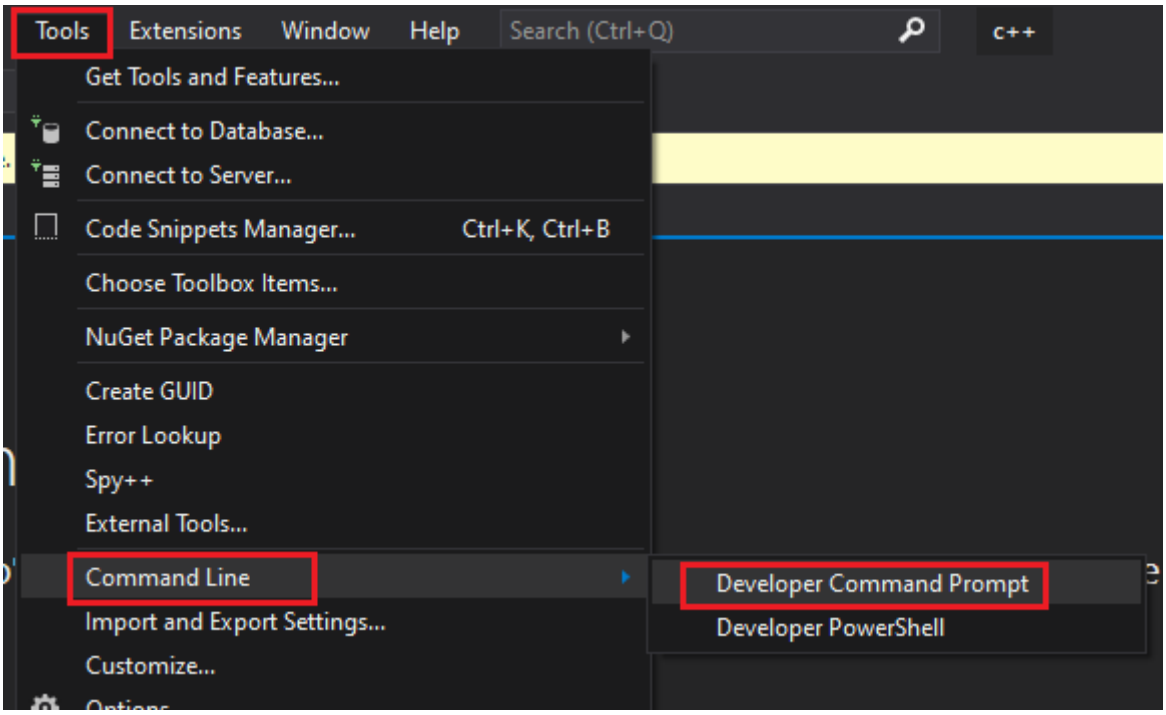

正在修改 - Visual Studio Community 2019 - 16.11.39
工作负荷 单个组件 语言包 安装位置
可以将其他语言包添加到 Visual Studio 安装。
中文(简体)
中文(繁体)
俄语
土耳其语
德语
意大利语
捷克语
日语
法语
波兰语
英语
葡萄牙语(巴西)
西班牙语
韩语
安装详细信息
Visual Studio 核心编辑器
使用 C++ 的桌面开发
已包含
C++ 核心桌面功能
可选
MSVC v142 - VS 2019 C++ x64/x86 生成工...
Windows 10 SDK (10.0.19041.0)
实时调试器
C++ 分析工具
用于 Windows 的 C++ CMake 工具
适用于最新 v142 生成工具的 C++ ATL (x8...
Boost.Test 测试适配器
Google Test 测试适配器
Live Share
IntelliCode
C++ AddressSanitizer
MSVC v142 - VS 2019 C++ ARM64 生成工...
适用于最新 v142 生成工具的 C++ MFC (x8...
位置
D:\Program Files (x86)\Microsoft Visual Studio\2019\Community
移除(R)不受支持的组件
系统驱动器(C) 75 MB
其他驱动器 95 MB
要求的总空间 169 MB
下载时安装
修改(M)


File Edit View Git Project Build Debug Test Analyze Tools Extensions Window Help
x64-Debug
The CMake toolchain file in the cache is different from the current toolchain file.
CMake 概述页
Overview
Target Linux
CMake in
Use Visual Studio
Get Tools and Features...
Connect to Database...
Connect to Server...
Code Snippets Manager... Ctrl+K, Ctrl+B
Choose Toolbox Items...
NuGet Package Manager
Create GUID
Error Lookup
Spy++
External Tools...
Command Line
Import and Export Settings...
Customize...
Options...



Click OK and restart the software.

Find the command line in the tool, open the developer command prompt, if the output prompts an error, ignore it first, and try the following steps to see if it works

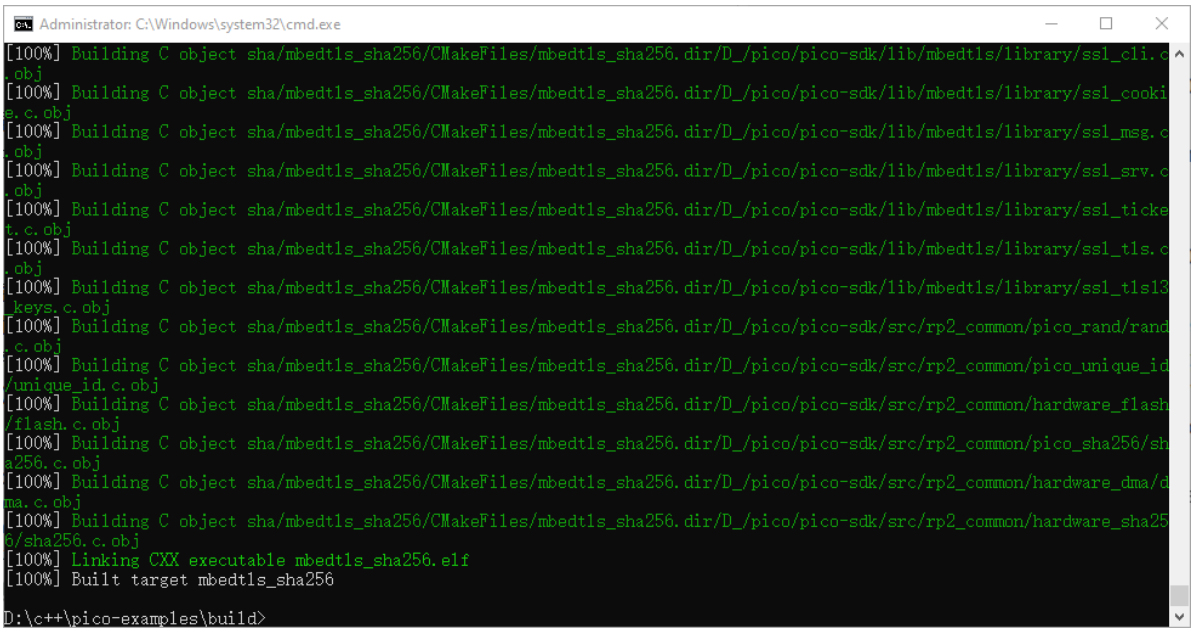


Enter the following commands in the opened command line

```
cd pico-examples
mkdir build
cd build
cmake -DPICO_BOARD=pico2 -DPICO_PLATFORM=rp2350 -G "NMake Makefiles" ..
#If it shows that the SDK path cannot be found, run the following command
../pico-sdk is the actual path of your SDK
cmake -DPICO_BOARD=pico2 -DPICO_PLATFORM=rp2350 -DPICO_SDK_PATH=../pico-sdk -G
"NMake Makefiles" ..

nmake
```

Wait for the compilation to complete



After the compilation is completed, find the corresponding compiled files bin elf u2f in the bulid folder. Drag the u2f in the above file into the disk recognized by Pico 2 (Note: When burning for the first time, it is an empty code. Pico 2 can directly recognize the disk when connected to USB. When there is an executable program in it, you need to press and hold the BOOTSEL button and then connect USB) After dragging in, the disk is disconnected and execution begins (the blink compiled file used here has the LED on the board flashing)