

# Windows下CUDA开发环境搭建

By yongjian.li

## 概述

本文主要介绍如何在Windows10下配置CUDA的开发环境以方便进行CUDA开发工作，windows下CUDA开发一般都是选择Visual Studio进行开发，需要先安装Visual Studio（一定要先安装），然后安装CUDA（CUDA会自动集成到Visual Studio中），之后就可以直接在Visual Studio中新建CUDA项目进行开发。但是鉴于平时开发用VScode居多，所以本文将介绍如何在VScode上进行CUDA C开发，包括**VScode安装**，**C/C++程序编译**，**CUDA安装**，**CUDA C程序编译**四部分内容。开发平台的操作系统为Win10，使用的IDE是**VScode**，CUDA版本为**CUDA11.1**，完成配置后将可以实现cu文件的编译和执行（需保证GPU支持CUDA）。

## 1. VScode安装

1. Visual Studio Code（简称VS Code）是一款由微软开发且跨平台的免费源代码编辑器。该软件支持语法高亮、代码自动补全（又称IntelliSense）、代码重构、查看定义功能，并且内置了命令行工具和Git版本控制系统。用户可以更改主题和键盘快捷方式实现个性化设置，也可以通过内置的扩展程序商店安装扩展以拓展软件功能。可以去VScode官网[下载安装包](#)，安装比较简单，一路点击NEXT就可以完成安装；
2. VScode的使用可以去官网查看[官方英文说明文档](#)，下面将介绍几个常用的快捷键和命令：

ctrl + shift + p	#显示Command Palette
ctrl + `	#调出terminal
ctrl + p	#快速打开文件
ctrl + ,	#打开settings(UI)全局配置界面
code .	#在当前文件夹下打开VScode
ctrl+shift+B	#run build task

3. 安装好VScode后还需要安装插件，安装插件只是方便编写程序包括自动补全等功能，但是插件并不包含编译器，所以需要单独下载编译器，以下为推荐下载安装的插件：

C++: C/C++ (必须安装)

CUDA C: Nsight Visual Studio Code Edition (必须安装)

Cmake: CMake、CMakeTools

远程开发: Remote - SSH、Remote - SSH: Editing Configuration Files

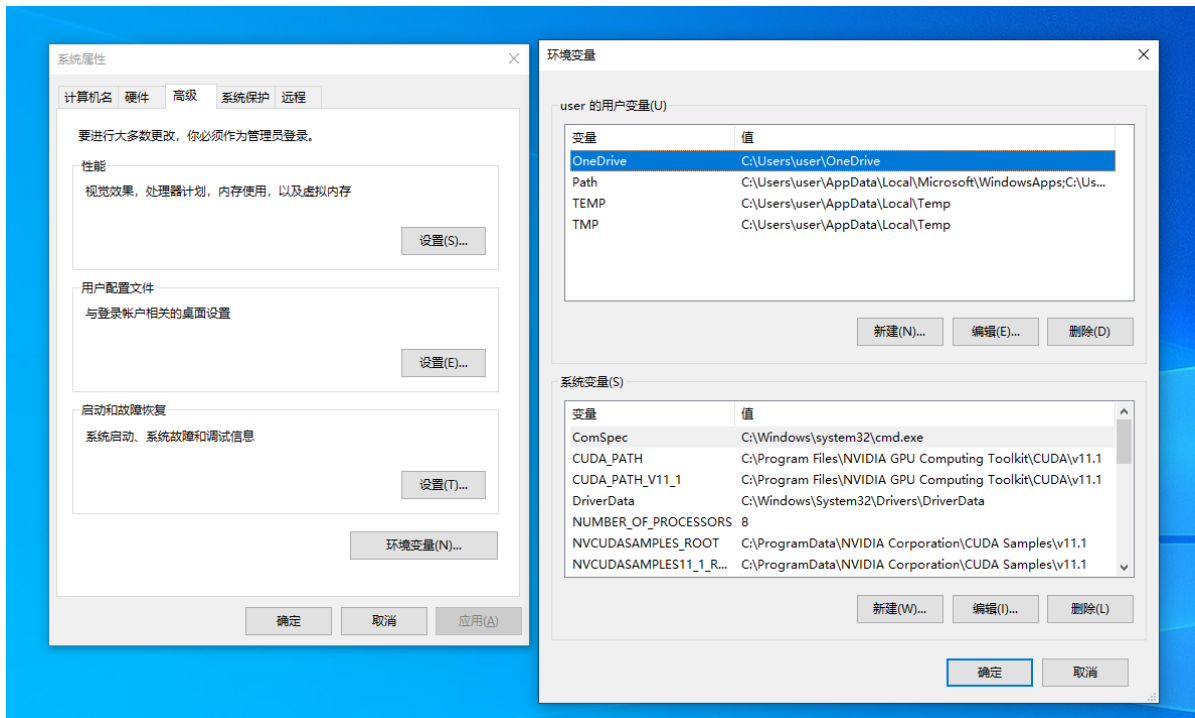
通用插件: Better Comments、GitHistory、psioniq File Header 用于自动生成文件头

4. windows下常用的C/C++编译器包括MSCV编译器和Mingw编译器，但在[windows下开发CUDA时只支持MSCV编译器](#)，所以我们应该下载和安装MSCV编译器。通常有两种方式安装MSCV编译器，第一种是直接安装Visual Studio，优点是不需要自己配置环境变量，但是会占用较大的空间；第二种是[只安装Build Tools for Visual Studio](#)，可以节省空间，但是需要自己配置windows系统的环境变量。如果选择第一种方式可以直接看第二大部分C/C++环境搭建与开发，否则需要进行环境配置。
5. 安装完Build Tools for Visual Studio后，如果想不配置环境变量也进行C/C++开发，只能在Developer Command Prompt终端中打开VScode：如下图所示找到该应用，然后在该终端中切换到项目文件夹输入 `code .` 即可打开VScode，但是这样比较不方便，所以可以自行配置环境变量，直接打开VScode进行C/C++开发，详见步骤6。



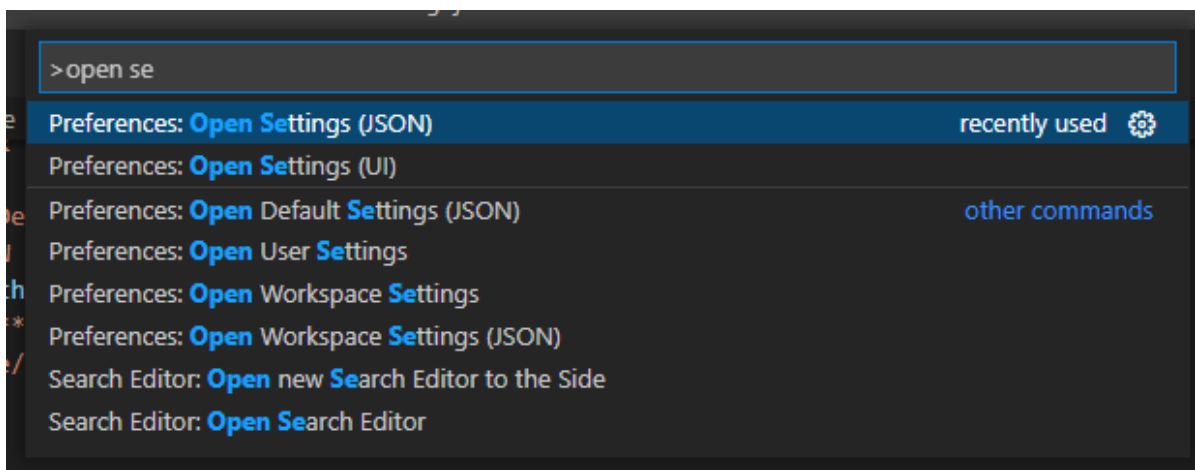
6. 配置环境变量的目的是为了let windows自身的终端找到MSCV编译器即cl.exe和依赖以实现编译。按快捷键win+R后，输入“sysdm.cpl”，然后回车，找到“高级”，点击右下方的环

境变量，可以看到如下界面，编辑下方的系统环境变量，一共需要三个环境变量，如果没有就需要新建，已经有的需要编辑增加路径，具体设置如下所示：



```
Path:
C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\VC\Tools\MSVC\14.29.30037\bin\Hostx64\x64
Include:
C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\VC\Tools\MSVC\14.29.30037\include;
C:\Program Files (x86)\Windows Kits\10\Include\10.0.19041.0\winrt;
C:\Program Files (x86)\Windows Kits\10\Include\10.0.19041.0\shared;
C:\Program Files (x86)\Windows Kits\10\Include\10.0.19041.0\ucrt;
C:\Program Files (x86)\Windows Kits\10\Include\10.0.19041.0\um
LIB:
C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\ucrt\x64;
C:\Program Files (x86)\Windows Kits\10\Lib\10.0.19041.0\um\x64;
C:\Program Files (x86)\Microsoft Visual Studio\2019\BuildTools\VC\Tools\MSVC\14.29.30037\lib\x64
```

7. 由于vscode默认使用“utf-8”编码，而在Windows控制台默认使用的是“gbk”编码，如果直接新建文件编译会出错，为此先对vscode做一些配置修改，确保新建的C/C++文件使用“gbk”编码。打开VScode，按下 `ctrl+shift+p`，调出Command Palette，输入 `open settings`，下拉列表中选择 `Preferences:Open Setting(JSON)`，增加对应代码即可。

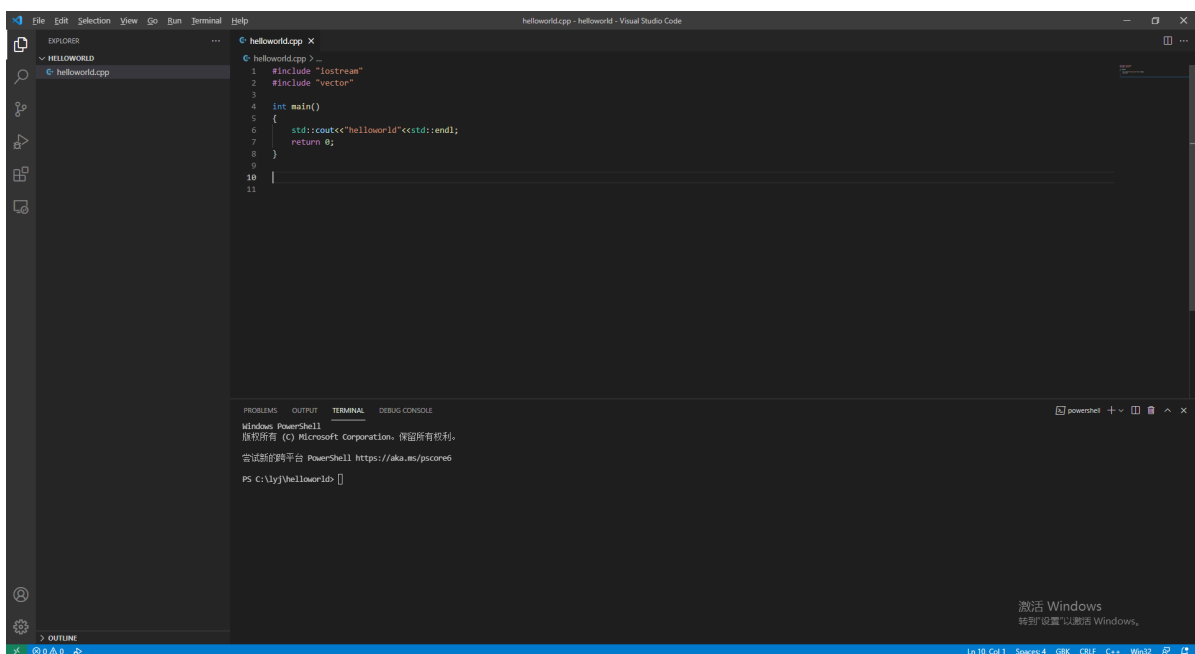


```
"[c]":{
  "files.encoding": "gbk"
},
"[cpp]": {
  "files.encoding": "gbk"
},
```

## 2. C/C++程序编译

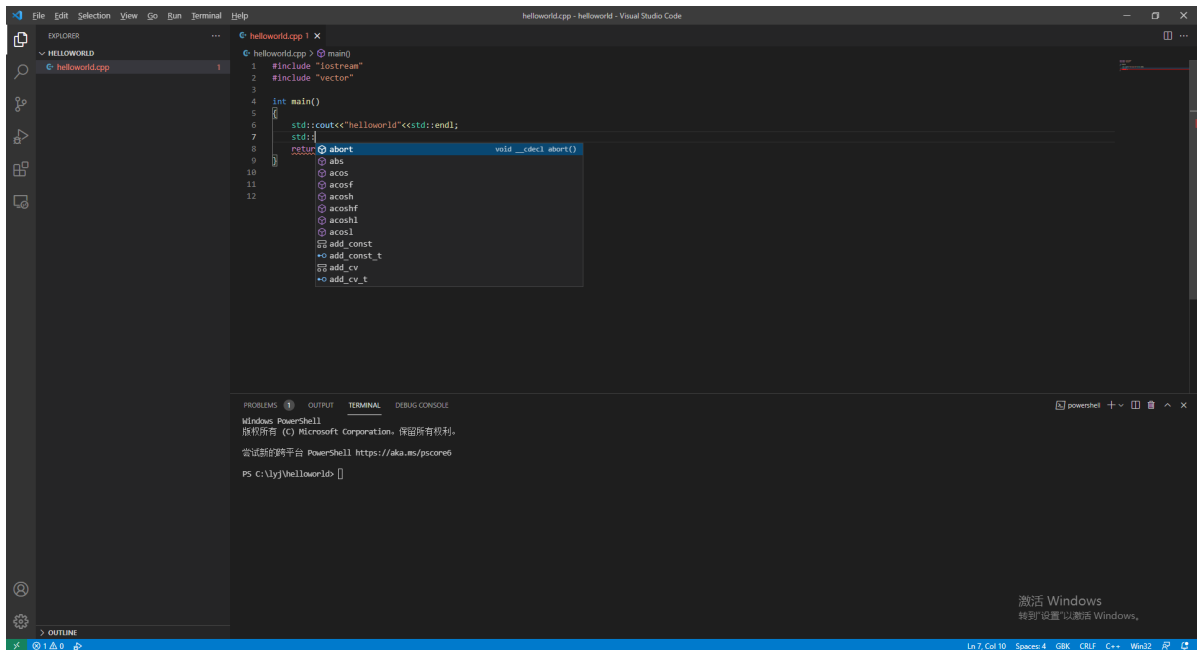
### 1. Create Hello World

先选择一个文件夹作为workspace，在windows终端中切换到该文件夹下然后输入code .，就可以在该文件夹下打开VScode，然后新建helloworld.cpp文件。



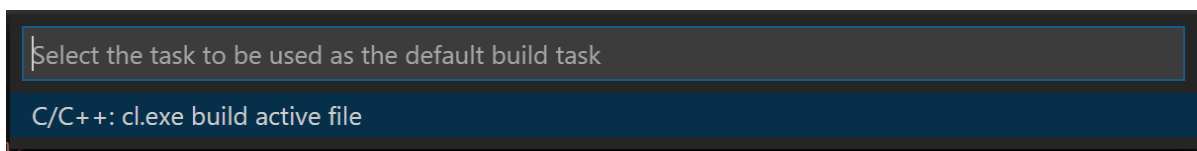
## 2. Explore IntelliSense

检查自动补全功能，输入std::，查看是否会有代码提示。



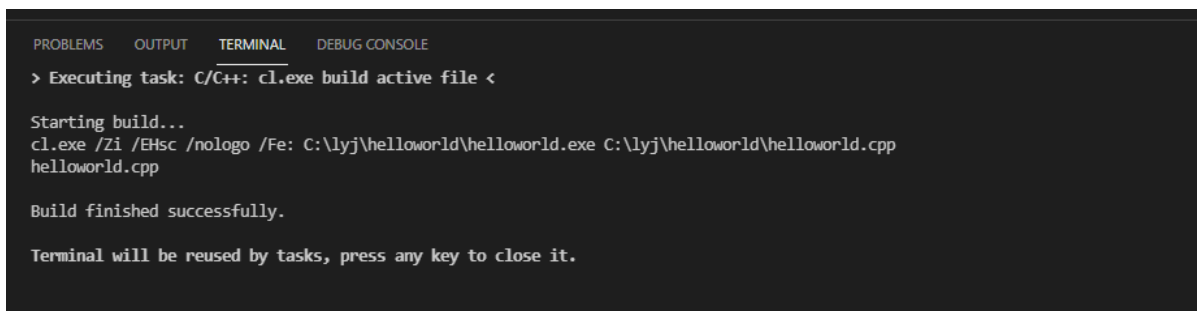
## 3. Build helloworld.cpp

选择 Terminal > Configure Default Build Task，选择 cl.exe。



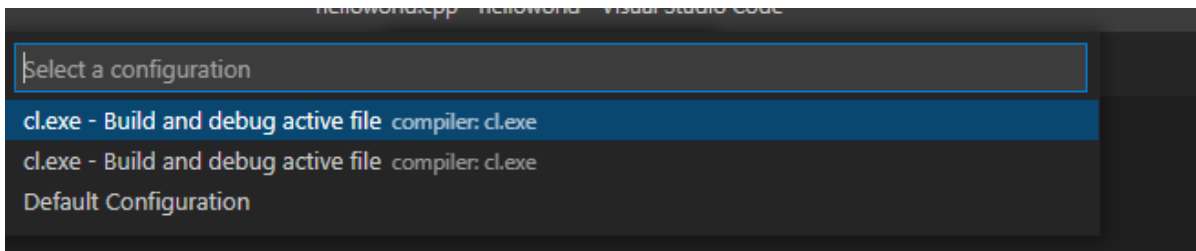
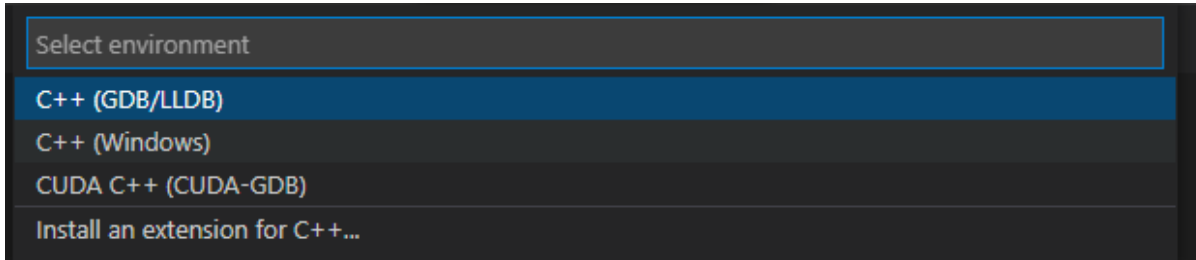
## 4. Running the build

回到 helloworld.cpp，按下 Ctrl+Shift+B，就会开始build，结果如下，代表build成功。



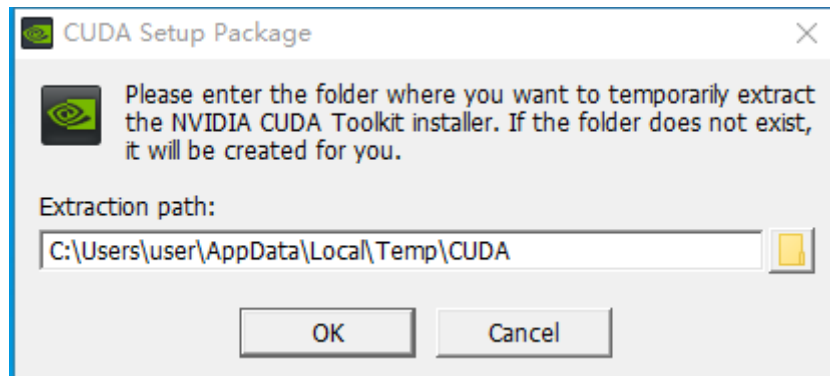
## 5. Debug helloworld.cpp

6. 回到 helloworld.cpp , 选择 Run > Add Configuration 并选择 C++ (Windows) , 然后选择 cl.exe build and debug active file , 再点击F5, 就会出现运行结果, 当然也可以设置断点。



### 3. CUDA安装

1. CUDA (Compute Unified Device Architecture, 统一计算架构) 是由英伟达NVIDIA所推出的一种集成技术, 是该公司对于GPGPU的正式名称。透过这个技术, 用户可利用NVIDIA的GeForce 8以后的GPU和较新的Quadro GPU进行计算。亦是首次可以利用GPU作为C-编译器的开发环境。可以去NVIDIA官网[下载CUDA11.1](#)的安装包, 下载完成后点击安装, 会弹出以下界面, 选择文件夹用来临时存放CUDA的安装程序, 一般默认不修改, **点击OK**, 等extract进度达到100%会就启动CUDA安装程序;



2. 安装程序启动后, 会首先启动系统兼容性检查, 会检查电脑的GPU卡是否支持CUDA等, 如果支持的话会显示下图, 接下来**点击同意并继续**;



3. 选择自定义安装并点击下一步；



4. 自定义安装是需要自己选择安装哪些文档，默认是全勾选的，出问题的可能性小，但是会占用更大的空间，这里我们选择全部勾选安装并**点击下一步**；

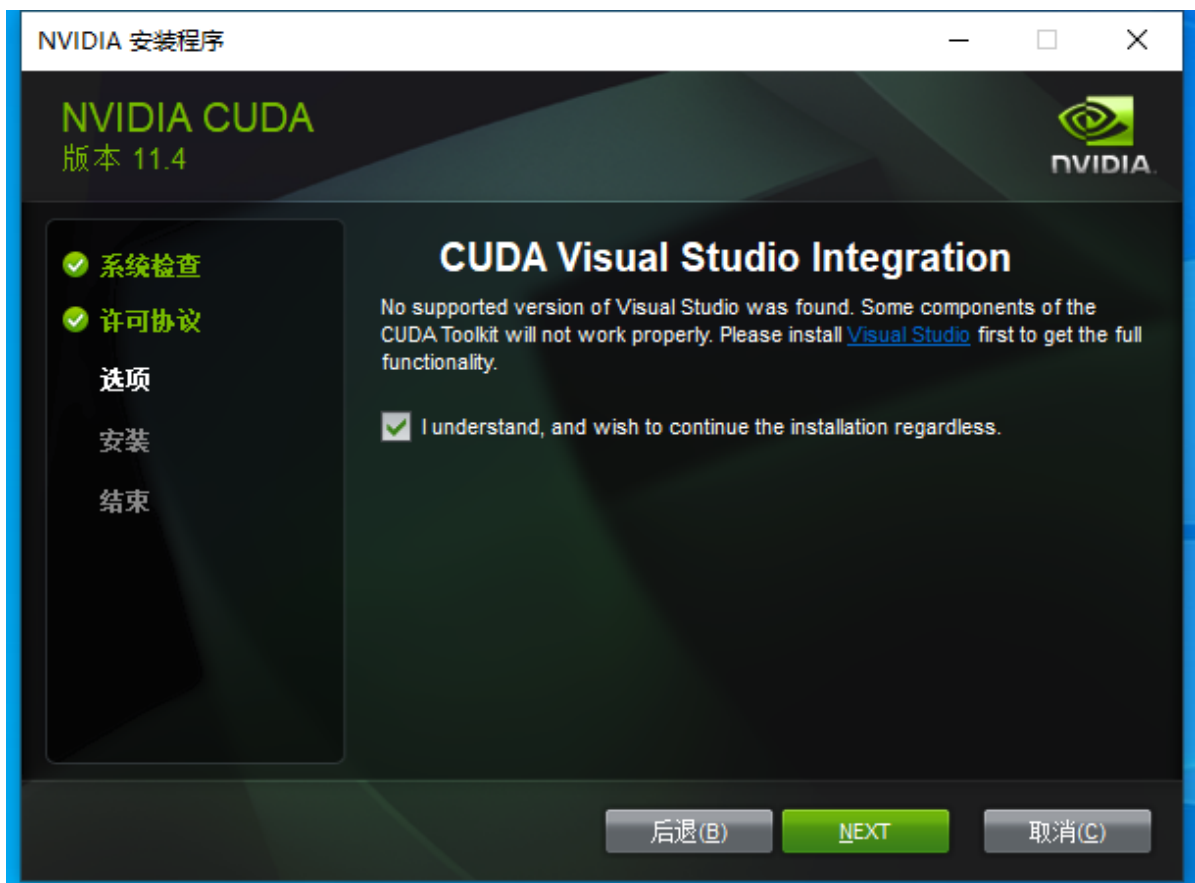


5. 弹出安装的位置，最好默认这个路径不要修改，**点击下一步**；

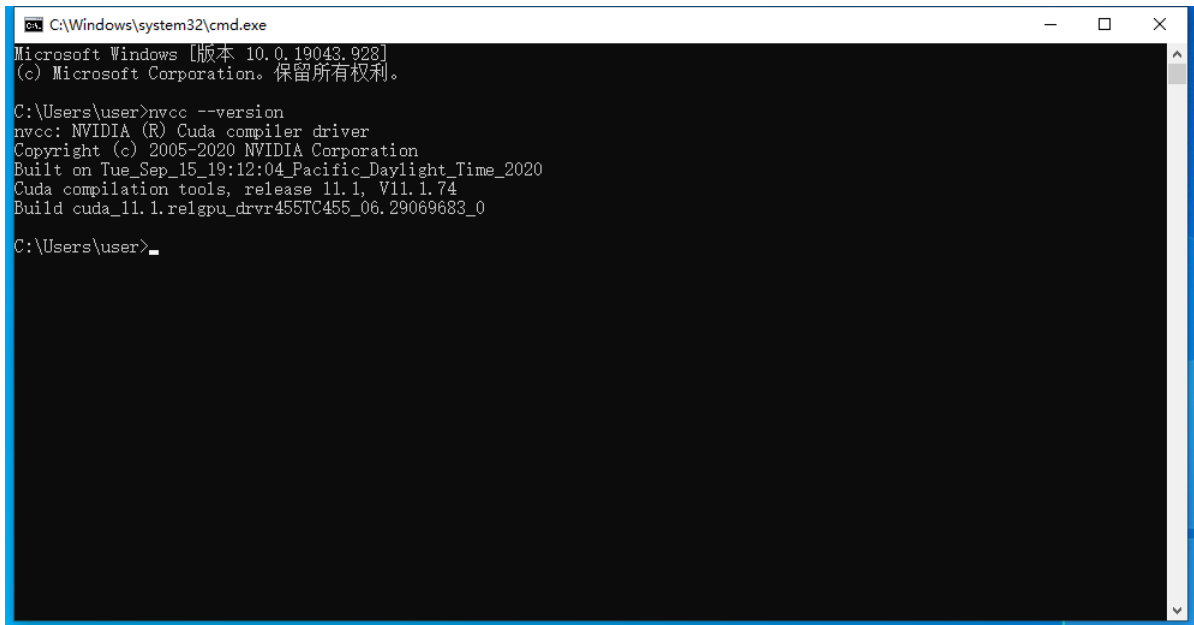




6. 如果你的电脑没有安装visual studio, 那么会出现如下提示, 我们勾选I understand并点击NEXT即可;



7. 等待安装完成以后**点击下一步**，再**点击完成**，这样CUDA11.1就安装好了；
8. 验证是否安装成功可以按住windows键+r，输入cmd，打开命令终端，输入**nvcc --version**，如果正确输出nvcc编译器版本则证明安装正确。



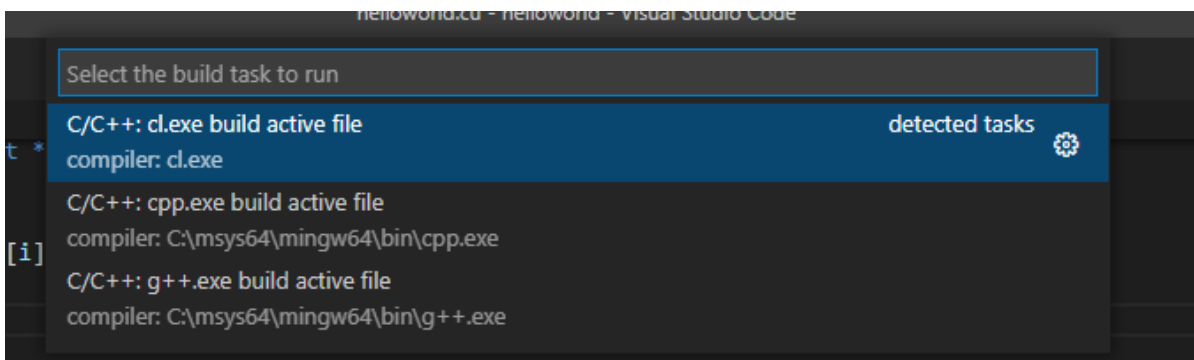
```
C:\Windows\system32\cmd.exe
Microsoft Windows [版本 10.0.19043.928]
(c) Microsoft Corporation. 保留所有权利。

C:\Users\user>nvcc --version
nvcc: NVIDIA (R) Cuda compiler driver
Copyright (c) 2005-2020 NVIDIA Corporation
Built on Tue_Sep_15_19:12:04_Pacific_Daylight_Time_2020
Cuda compilation tools, release 11.1, V11.1.74
Build cuda_11.1.relgpu_drvr455TC455_06.29069683_0

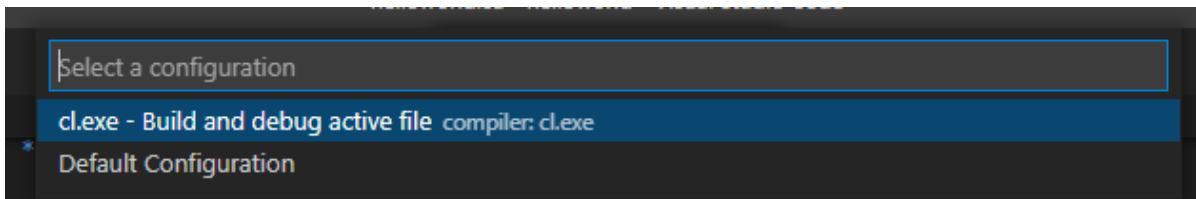
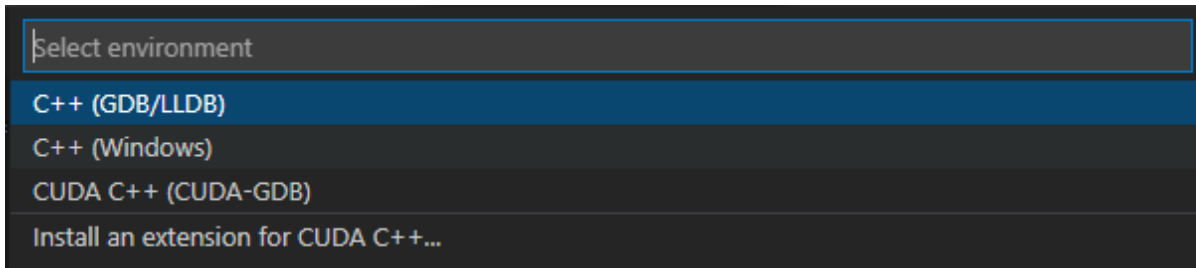
C:\Users\user>
```

## 4. CUDA C程序编译

1. 同样新建文件夹，然后在该文件夹下新建helloworld.cu文件，编写一个简单的CUDA程序，编写过程中注意是否有自动补全功能，如果没有则有可能是因为没有安装Nsight Visual Studio Code Edition插件；
2. 写完程序后点击 terminal>configure default build task ，选择  
cl.exe build active file ，修改tasks.json文件，主要修改  
"command": "nvcc"和"args": ["-g", "-o", "\${fileDirname}\\\${fileBasenameNoExtension}.exe", "\${file}"]  
然后重新回到cu文件，按 ctrl+shift+B ，就可以进行build。



3. 对于Windows是无法调用CUDA-gdb的，但是可以debug cpp代码，回到cu文件，点击 F5，选择c++ (windows)， cl.exe， 会生成launch.json， 然后重新回到cu文件， 点击 F5， 就可以debug cpp代码。



## Reference

1. <https://code.visualstudio.com/docs/cpp/config-msvc>
2. <https://stackoverflow.com/questions/5888908/cuda-with-mingw-updated>
3. <https://code.visualstudio.com/docs/cpp/config-msvc>
4. <https://marketplace.visualstudio.com/items?itemName=NVIDIA.nsght-vscode-edition>
5. <https://ericzhng.github.io/eric-blogs/2018/10/11/vs-code-cuda/>