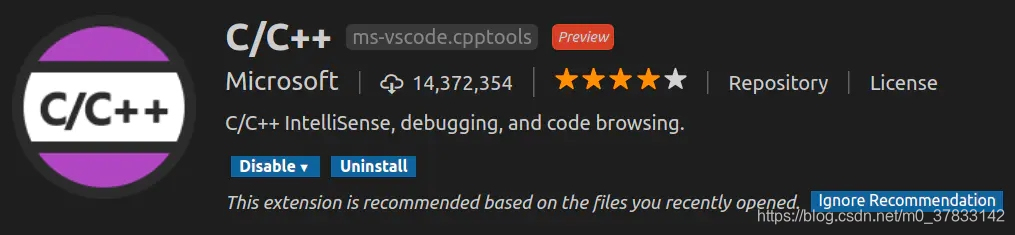
# 1,安装vscode，这个本节简单，略。

# 2.安装TDM-GCC, [下载地址](https://link.juejin.cn?target=https%3A%2F%2Fjmeubank.github.io%2Ftdm-gcc%2Fdownload%2F)

这里很多网站的帖子都建议下载minGW ，*但是不少Min-GW编译器不支持C++版本11或更高版本。OpenCV中的线程需要这个版本*。所以我建议使用TDM—GCC，exe直接安装就行了，还有一个优点就是他会自己你配置环境变量。很方便

# 3.安装vscode的c++扩展

## 搭建vscode的扩展按钮，输入c，然后选择下面这个搭建安装

C/C++编译工具  
  
**完成后，我们就可以编辑一段C++代码并调试！**

## **编写一段c++代码，如下**

|  |
| --- |
|  |

## 这时我们只是完成了源码编辑，按**F5调试会提示创建launch.json脚本文件**，或者在**左侧调试栏创建launch.json文件**（父目录：.vscode自动创建） 在这里插入图片描述

## 我们写的是C++文件，所以选择g++(gdb)选项–>选择"g++.exe build active file"，然后就会生成一个task.json,内容如下

|  |
| --- |
| {      "tasks": [          {              "type": "cppbuild",              "label": "g++.exe build active file",              "command": "d:\\programs\\TDM-GCC-64\\bin\\g++.exe",              "args": [                  "-fdiagnostics-color=always",                  "-g",                  "${file}",                  "-o",                  "${fileDirname}\\${fileBasenameNoExtension}.exe"              ],              "options": {                  "cwd": "d:\\programs\\TDM-GCC-64\\bin"              },              "problemMatcher": [                  "$gcc"              ],              "group": {                  "kind": "build",                  "isDefault": true              },              "detail": "调试器生成的任务。"          }      ],      "version": "2.0.0"  } |

## 然后我们需要编写一个lauch.josn文件，内容如下

|  |
| --- |
| {      "configurations": [          {              "name": "C/C++: g++.exe 生成和调试活动文件",              "type": "cppdbg",              "request": "launch",              "program": "${fileDirname}\\${fileBasenameNoExtension}.exe",              "args": [],              "stopAtEntry": false,              "cwd": "${workspaceFolder}",              "environment": [],              "externalConsole": true,              "MIMode": "gdb",              "miDebuggerPath": "D:\\programs\\TDM-GCC-64\\bin\\gdb.exe",              "setupCommands": [                  {                      "description": "为gdb启用整齐打印",                      "text": "-enable-pretty-printing",                      "ignoreFailures": true                  }              ],              "preLaunchTask": "g++.exe build active file"          }      ]  } |

# 4.配置opencv

## 1）需要先用TDM-GCC把opencv源码编译为TDM-GCC版本的库，使用的是minGW项目一类型，需要先用cmke配置，可以参考：[clion+tdm-gcc+opencv+contrib开发环境搭建](clion+tdm-gcc+opencv+contrib开发环境搭建.docx)，我们把编译好的opencv改名opencvlib放置d盘的programs目录里面

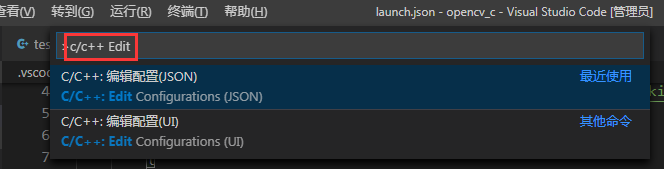
|  |
| --- |
|  |

## 2）然后我们把上面的代码修改如下：

|  |
| --- |
| #include <opencv2/opencv.hpp>  #include <opencv2/highgui.hpp>  using namespace cv;  int main()  {      Mat img = imread("./girls.png");      imshow("beauty", img);      waitKey(0);      destroyAllWindows();      return 0;  } |

### 注意：此时会报错，因为我们还没有配置opencv

## 3) 按Ctrl+Shift+P打开命令面板，输入>c/c++ Edit，选择Edit Configuration(JSON)即可创建c\_cpp\_properties.json文件。

  
在c\_cpp\_properties.json中添加opencv头文件路径：

"D:/programs/opencvlib/include",

"D:/programs/opencvlib/include/opencv",

"D:/programs/opencvlib/include/opencv2"

c\_cpp\_properties.json完整的内容如下：

|  |
| --- |
| {      "configurations": [          {              "name": "Win32",              "includePath": [                  "${workspaceFolder}/\*\*",                  "D:/programs/opencvlib/include",                  "D:/programs/opencvlib/include/opencv",                  "D:/programs/opencvlib/include/opencv2"              ],              "defines": [                  "\_DEBUG",                  "UNICODE",                  "\_UNICODE"              ],              "windowsSdkVersion": "10.0.22621.0",              "compilerPath": "d:\\programs\\TDM-GCC-64\\bin\\g++.exe",              "cStandard": "c17",              "cppStandard": "c++17",              "intelliSenseMode": "clang-x64"          }      ],      "version": 4  } |

4)修改一下task.json文件，添加一下库的路径，完整代码如下

|  |
| --- |
| {      "tasks": [          {              "type": "cppbuild",              "label": "g++.exe build active file",              "command": "d:\\programs\\TDM-GCC-64\\bin\\g++.exe",              "args": [                  "-fdiagnostics-color=always",                  "-g",                  "${file}",                  "-o",                  "${fileDirname}\\${fileBasenameNoExtension}.exe",                  "-I",                  "D:/programs/opencvlib/include",                  "-I",                  "D:/programs/opencvlib/include/opencv",                  "-I",                  "D:/programs/opencvlib/include/opencv2",                  "-L",                  "D:/programs/opencvlib/x64/mingw/lib",                  "-l",                  "libopencv\_img\_hash341",                  "-l",                  "libopencv\_world341"              ],              "options": {                  "cwd": "d:\\programs\\TDM-GCC-64\\bin"              },              "problemMatcher": [                  "$gcc"              ],              "group": {                  "kind": "build",                  "isDefault": true              },              "detail": "调试器生成的任务。"          }      ],      "version": "2.0.0"  } |

### 点击右上角的调试按钮，效果如下

|  |
| --- |
|  |

# 恭喜，vscode+tdm-gcc+opencv开发环境搭建成功