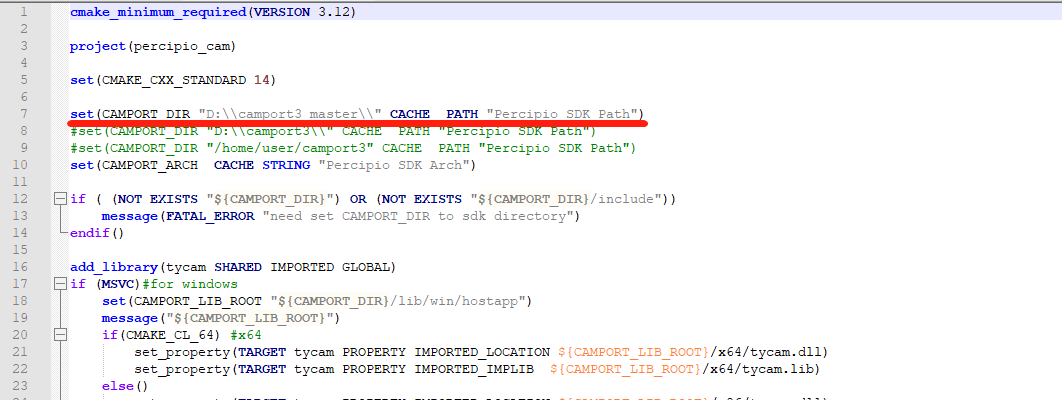
# Win:

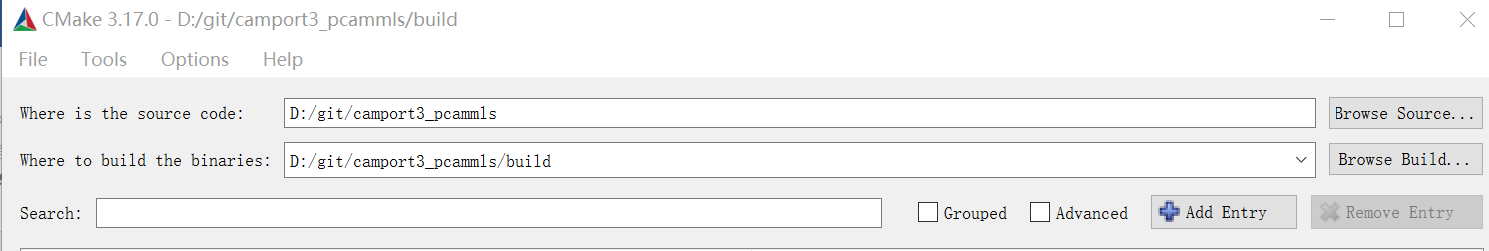
## PYTHON:

1. 下载图漾相机SDK保存至本地，下载地址: <https://github.com/percipioxyz/camport3>
2. 打开此SDK目录下CMakeLists.txt,修改文件指定camport3路径并保存退出，如下图所示：

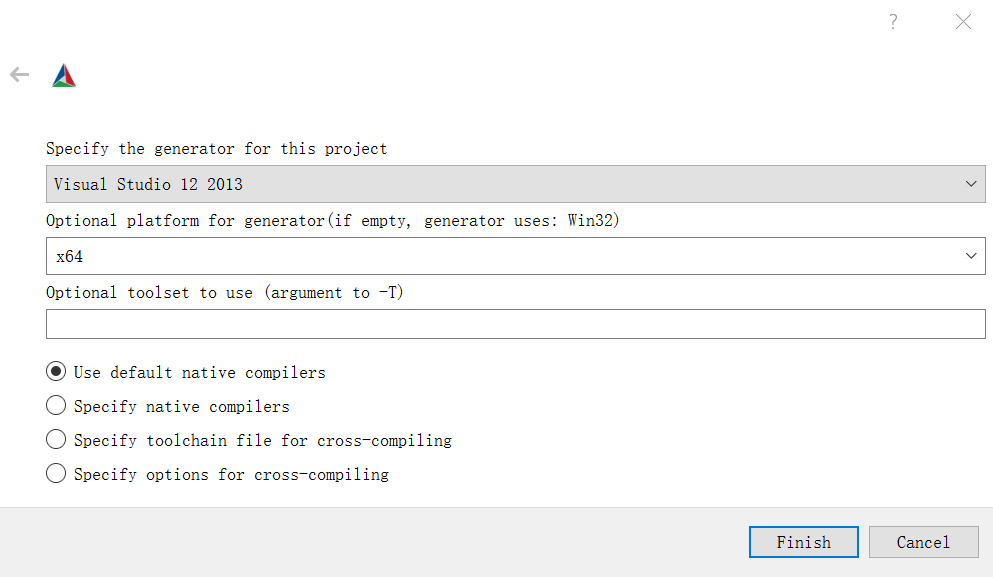


划红线处分别设置了本地camport3路径

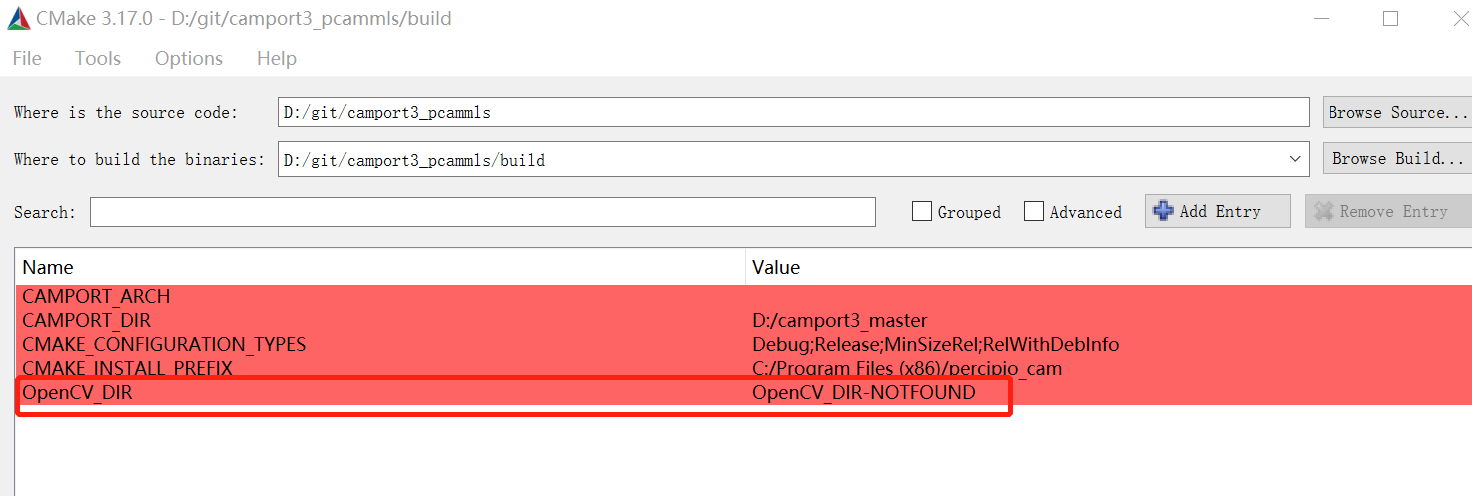
1. 打开cmake-gui，打开cmakelists.txt，并指定生成项目路径，如下图所示：

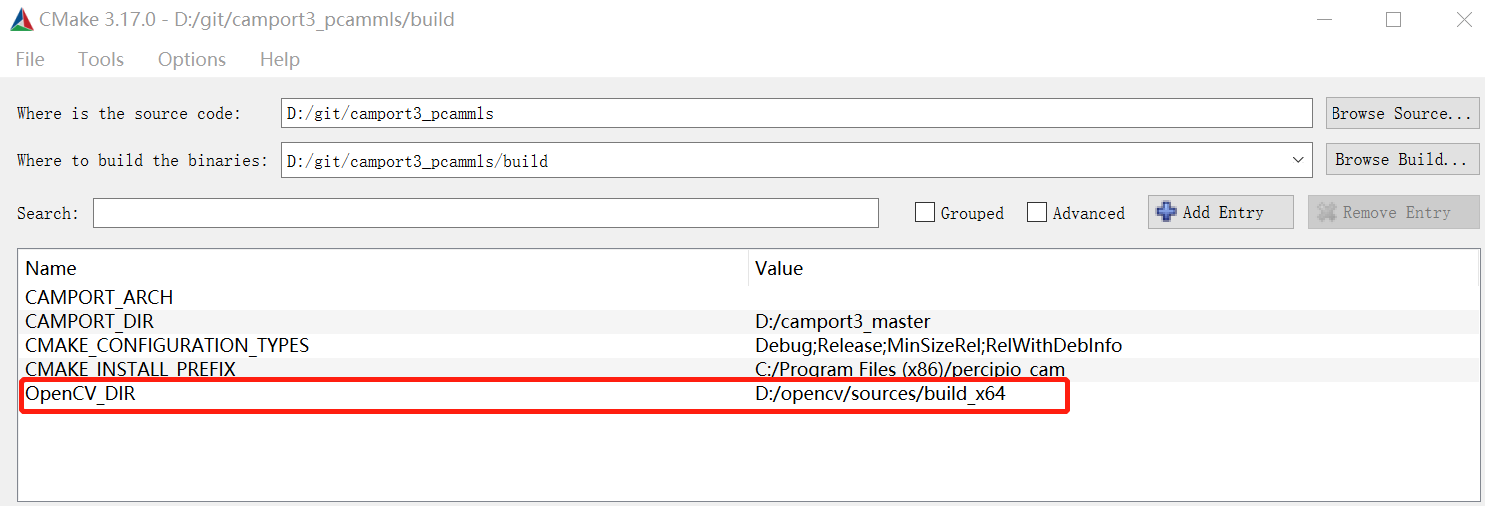


1. 点击Configure,设置生成的目标项目版本，包括VS版本及X86/X64版本设置：



1. 依次点击Generate/Open Project，生成并打开VS项目，选择Debug/Release后，完成编译(如果没有配置本地opencv环境，会提示找不到opencv，需要手动指定opencv本地路径，如下两图所示，分别为未指定opencv路径报错和手动指定了opencv路径)。





1. 使用cmake-gui生成本地VS项目之后，使用VS打开此项目，编译之后会生成动态库percipio\_cam.dll，将生成的动态库percipio\_cam.dll和SDK camport3中对应平台的动态库tycam.dll拷贝至同一路径下，同时将此SDK中的测试demo文件percipio\_test.py拷贝至此路径下，即可使用python打开percipio\_test.py。

注:python sdk sample程序用到了numpy和opencv，若本地没有安装，需要使用pip安装

## CSHARP:

流程和PYTHON基本一致，只是在CMAKE生成编译项目之前，可以打开CMakeLists.txt，将CSHARP\_CAM\_DEMO改成ON，这样在cmake生成项目的时候会生成csharp的测试demo程序: percipio\_test，完全编译完成之后，将SDK动态库tycam.dll以及生成的动态库percipio\_cam.dll以及生成的csharp demo程序 percipio\_test.exe复制至同一目录下(如果系统没有设置opencv环境变量，同样需要将opencv动态库拷贝至此目录)，即可运行程序打开相机。

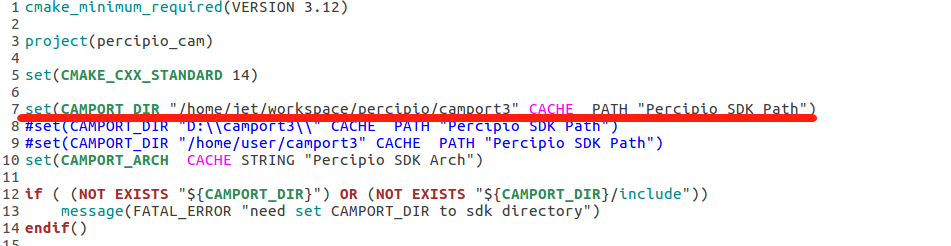
# Linux:

## PYHON:

1. 下载图漾相机SDK保存至本地，下载地址: <https://github.com/percipioxyz/camport3，也可以使用git工具将项目clone下来>:

git clone <https://github.com/percipioxyz/camport3.git>

1. 切换至此SDK目录，打开CMakeLists.txt文件，修改文件指定camport3路径后保存并退出，如下图所示：



划红线处分别设置了本地camport3路径

1. 在此SDK目录下依次执行以下命令：

mkdir build

cd build

cmake ..

make

将生成的动态库libpercipio\_cam.so 和SDK camport3下的对应平台动态库libtycam.so 复制至/usr/lib 目录下

1. 使用python打开percipio\_test.py(USB设备需要注意设备权限，可以通过sudo运行程序或者直接编写Linux下设备权限规则).

注:python sdk sample程序用到了numpy和opencv，若本地没有安装，需要使用pip安装