Qt+OpenCV环境搭建

1.在网上下载3个文件：

qt-opensource-windows-x86-mingw492-5.6.2.exe

cmake-3.6.3-win32-x86.zip

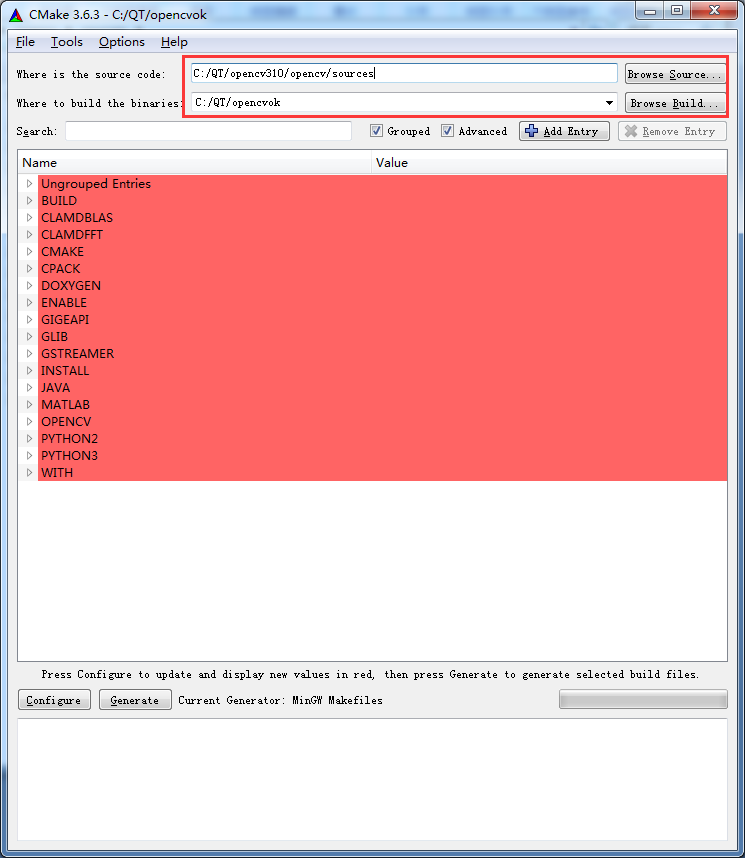
opencv-3.1.0.exe

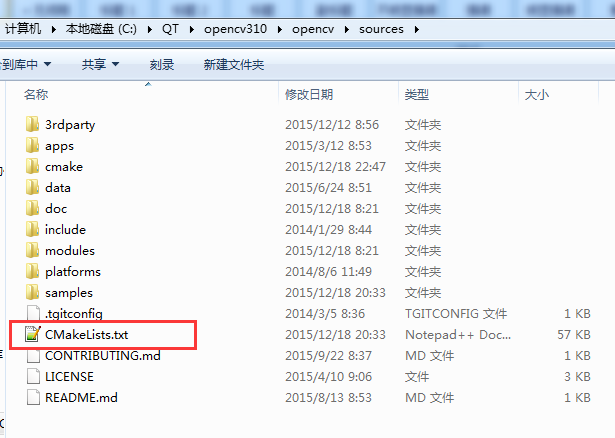
百度云共享地址：http://pan.baidu.com/s/1qXKNQJq

2.安装qt 5.6.2，截图省略

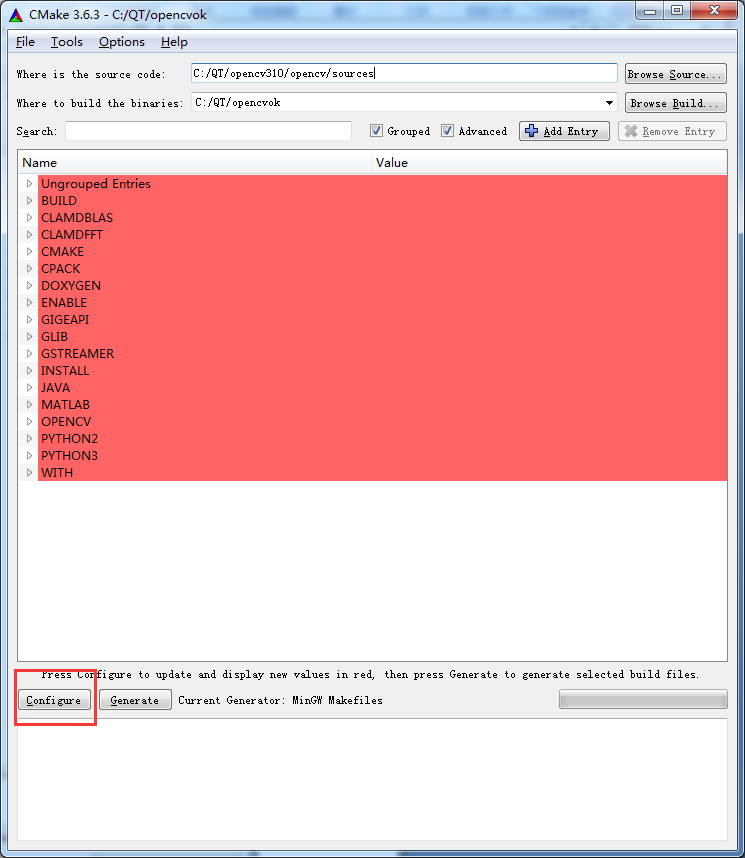
3.在cmake中配置opencv编译参数

1设好源(见再下一张图，要选CMakeLists.txt的目录)和目标(目标可以随意)

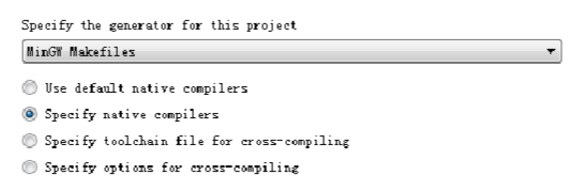




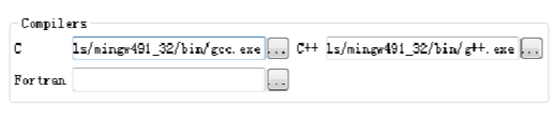
2.点Configure



3.选择编译器→MinGW Makefiles→Specify native compilers



4.设置编译器路径



C为：C:/QT/Qt5.6.2/Tools/mingw492\_32/bin/gcc.exe (请根据自己的目录进行选择)

C++为：C:/QT/Qt5.6.2/Tools/mingw492\_32/bin/g++.exe (请根据自己的目录进行选择)

5.报错，说opencv\_ffmpeg.dll下载失败

将opencv\build\bin目录中的

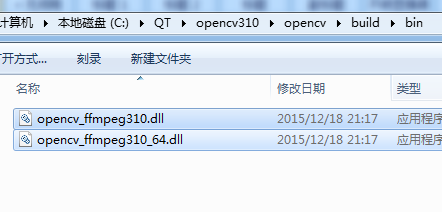
opencv\_ffmpeg310.dll

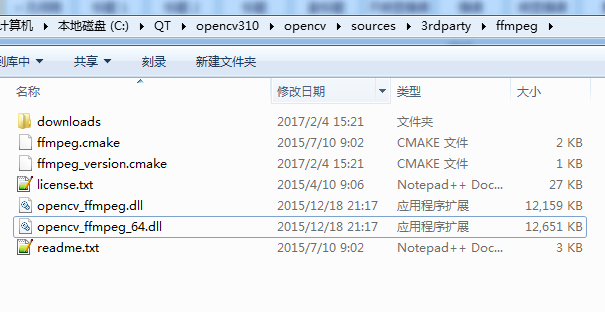
opencv\_ffmpeg310\_64.dll

复制到opencv\sources\3rdparty\ffmpeg目录，并改名为

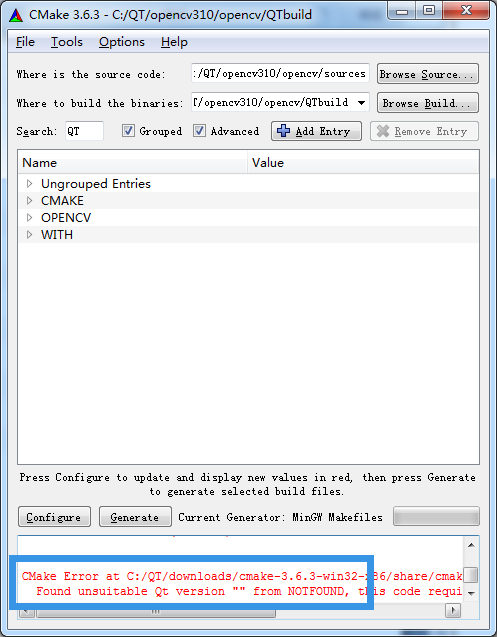
opencv\_ffmpeg.dll

opencv\_ffmpeg\_64.dll

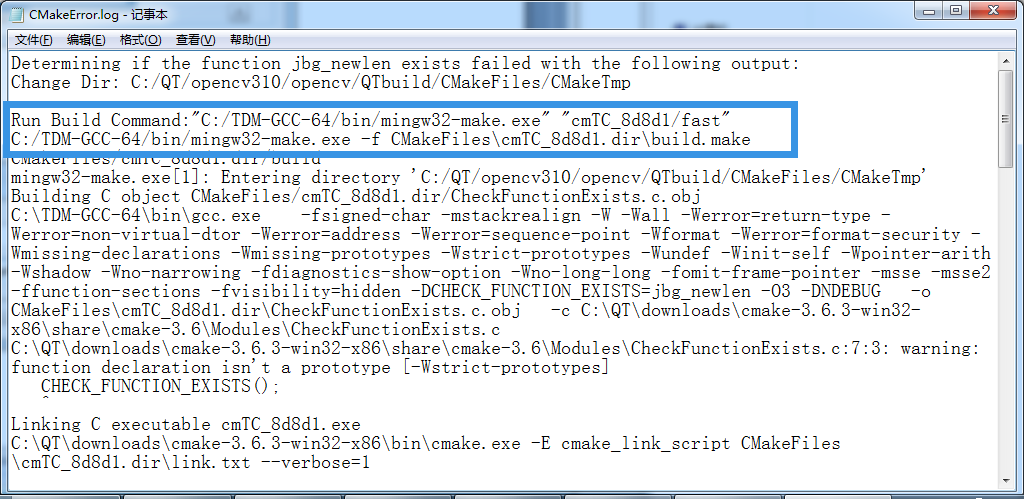




6.再点一次Configure，继续报错。提示没有合适的Qt版本



查看日志



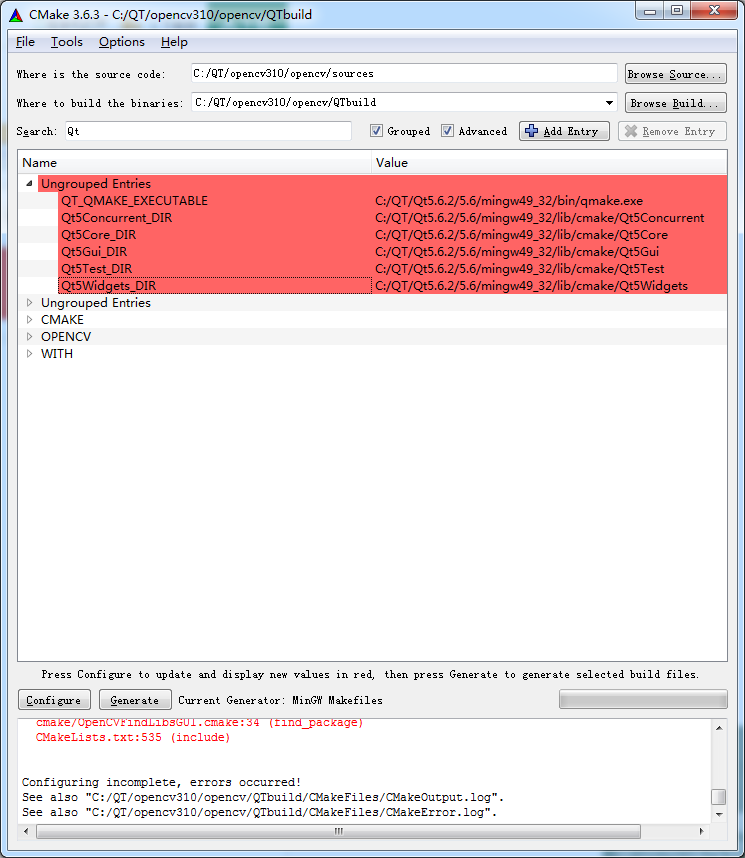
查看系统变量(控制面板→系统→高级系统设置→高级→环境变量)居然是TDM-GCC在前.....

所以将Qt的移前



不过还是不行，手工指定再试一次

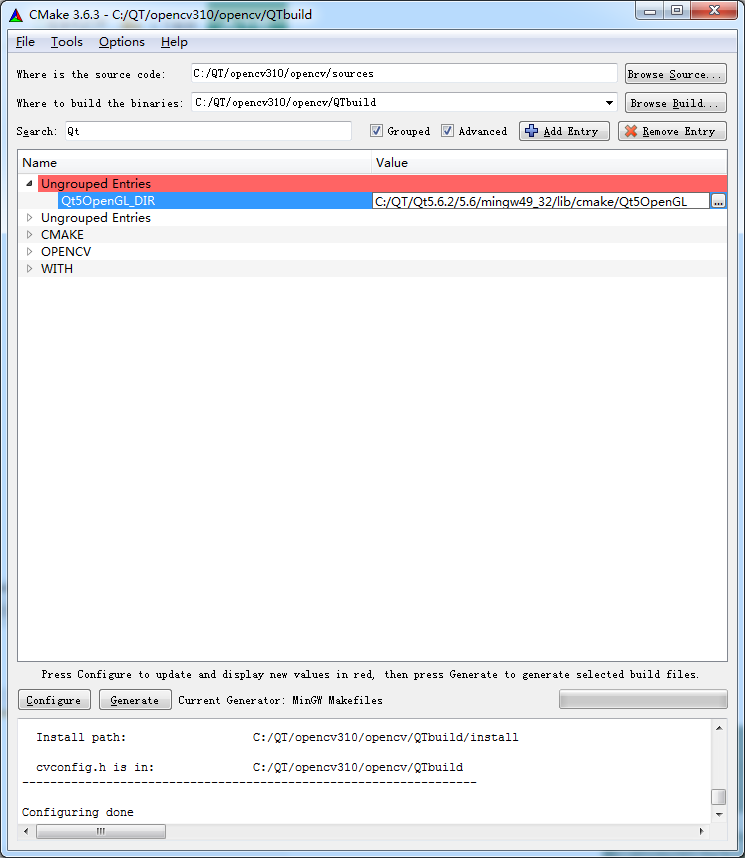
|  |  |
| --- | --- |
| QT\_QMAKE\_EXECUTABLE | C:/QT/Qt5.6.2/5.6/mingw49\_32/bin/qmake.exe |
| Qt5Concurrent\_DIR | C:/QT/Qt5.6.2/5.6/mingw49\_32/lib/cmake/Qt5Concurrent |
| Qt5Core\_DIR | C:/QT/Qt5.6.2/5.6/mingw49\_32/lib/cmake/Qt5Core |
| Qt5Gui\_DIR | C:/QT/Qt5.6.2/5.6/mingw49\_32/lib/cmake/Qt5Gui |
| Qt5Test\_DIR | C:/QT/Qt5.6.2/5.6/mingw49\_32/lib/cmake/Qt5Test |
| Qt5Widgets\_DIR | C:/QT/Qt5.6.2/5.6/mingw49\_32/lib/cmake/Qt5Widgets |



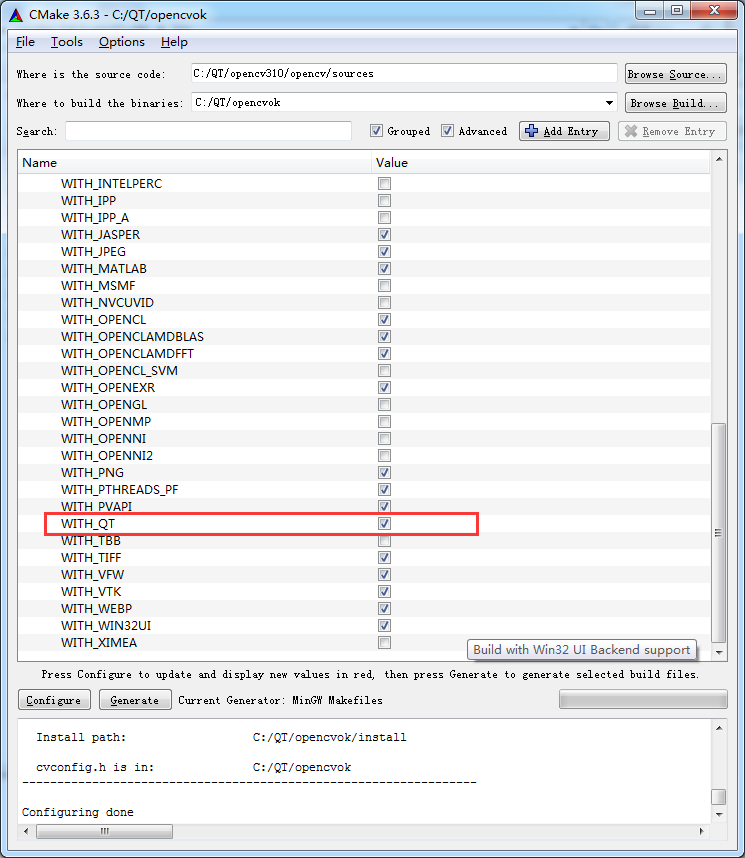
再点Configure，继续报错

同样还要设

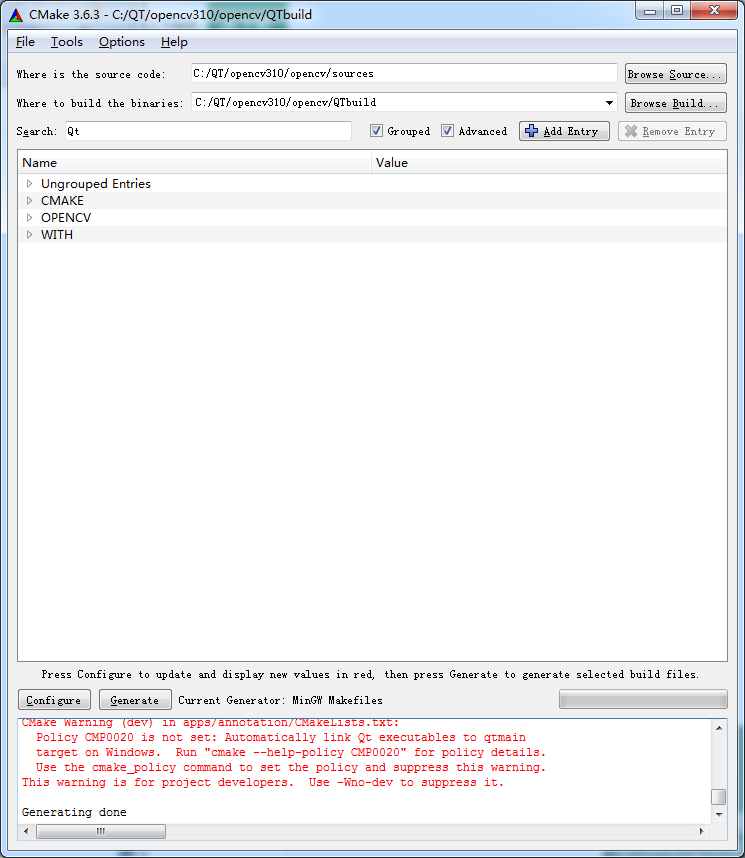
|  |  |
| --- | --- |
| Qt5OpenGL\_DIR | C:/QT/Qt5.6.2/5.6/mingw49\_32/lib/cmake/Qt5OpenGL |



再点Configure，上面己经没有红色，再次检查是否己勾选“WITH\_QT”



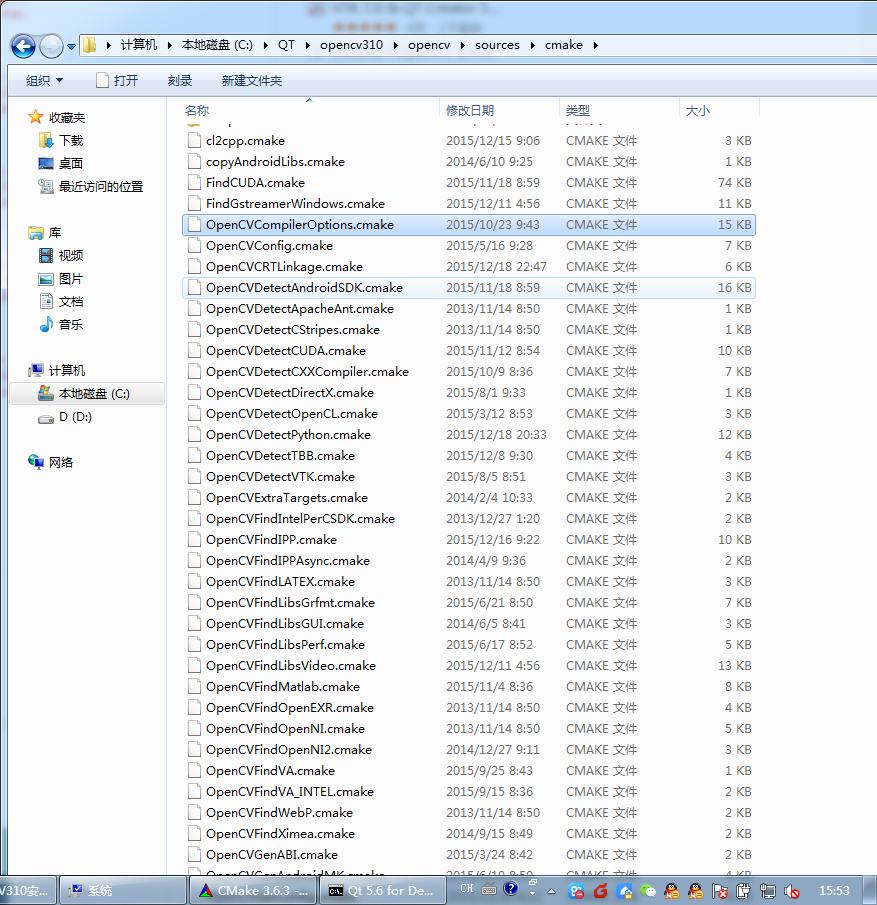
7.再点Configure如果没有红色，就可以点Generate。不要理会报错，看到最后的“Generating done”



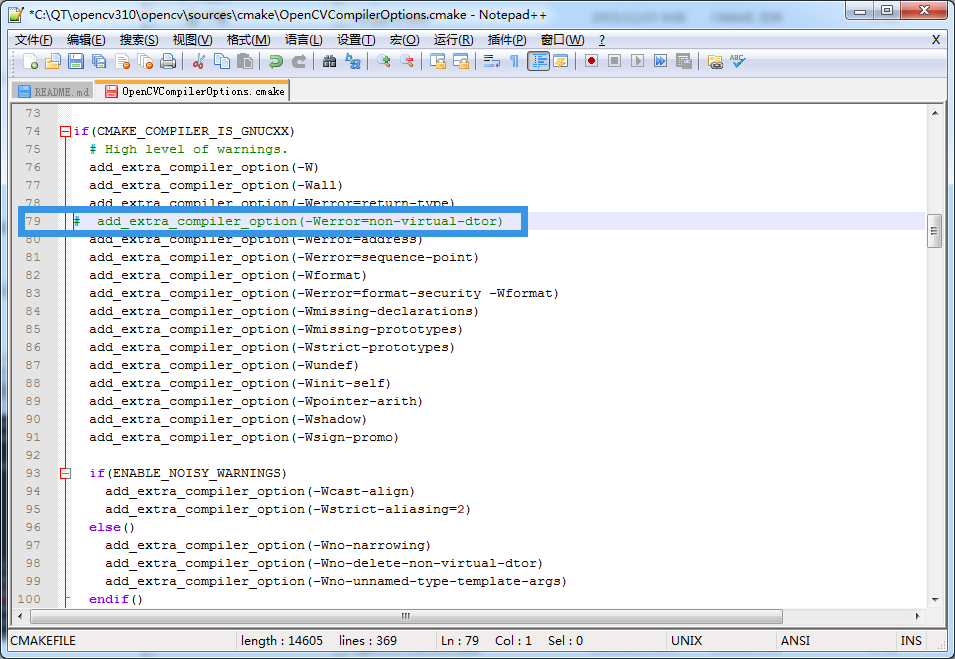
光是来到这一步己经这么坑，前前后后折腾cmake就已经用了2、3个晚上。

4.编译前做准备

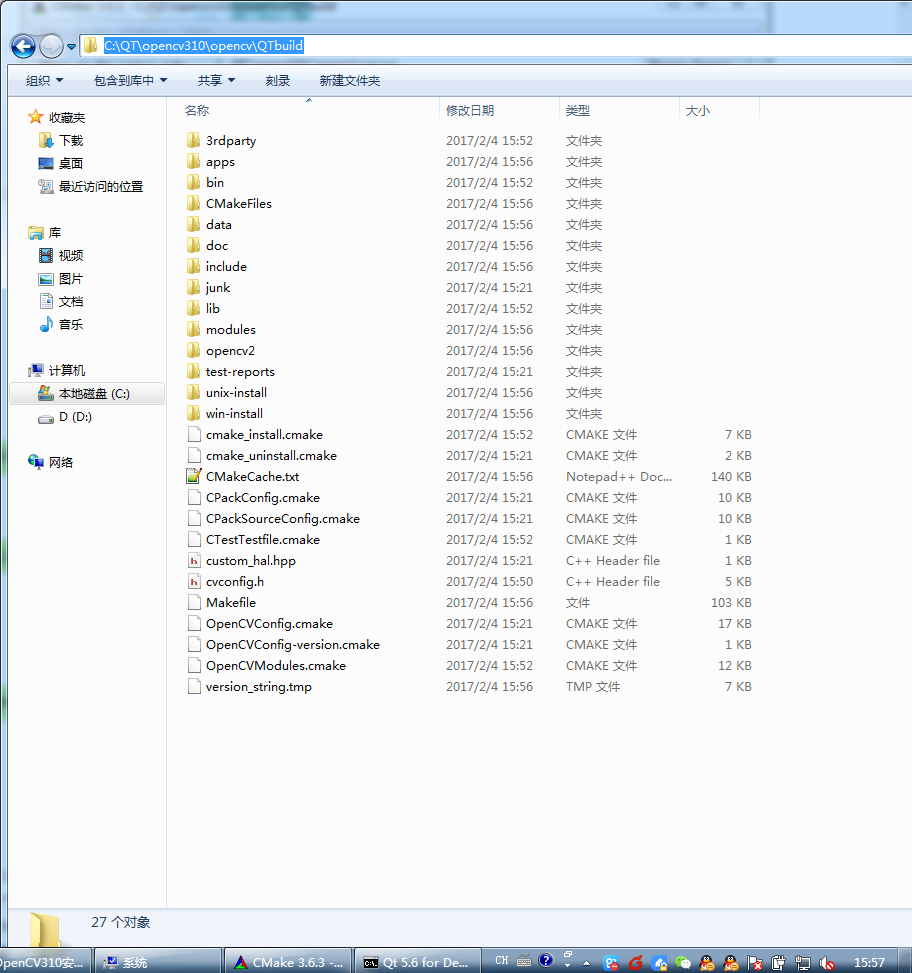
1.先到opencv\sources\cmake找到OpenCVCompilerOptions.cmake



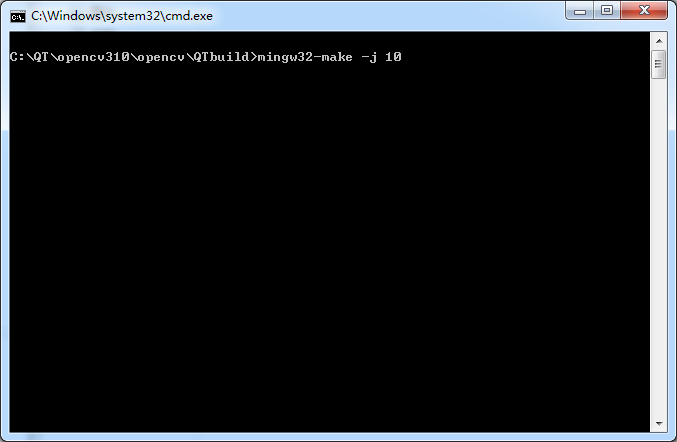
2.将add\_extra\_compiler\_option(-Werror=non-virtual-dtor)注释掉(在第79行的位置)



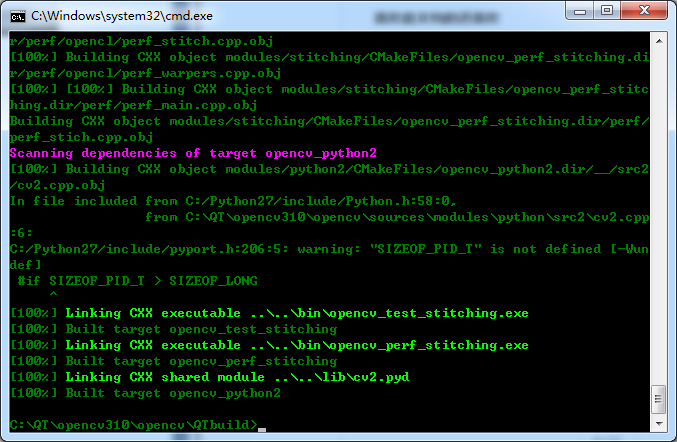
3.去opencv\QTbuild目录打开命令行进行编译



mingw32-make -j 10

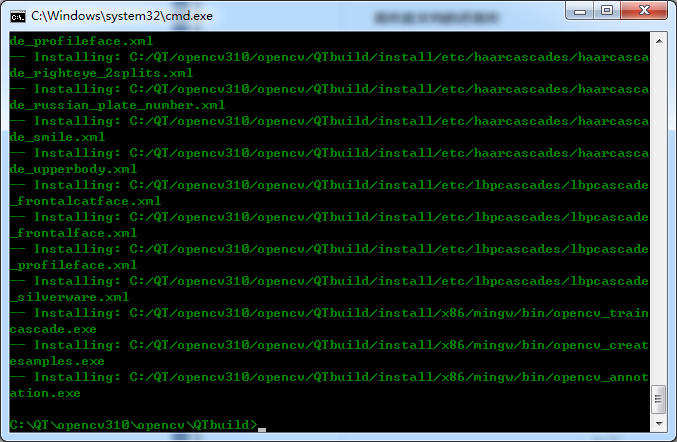


狂等直到100%(耗时比较长)



5.安装

mingw32-make install



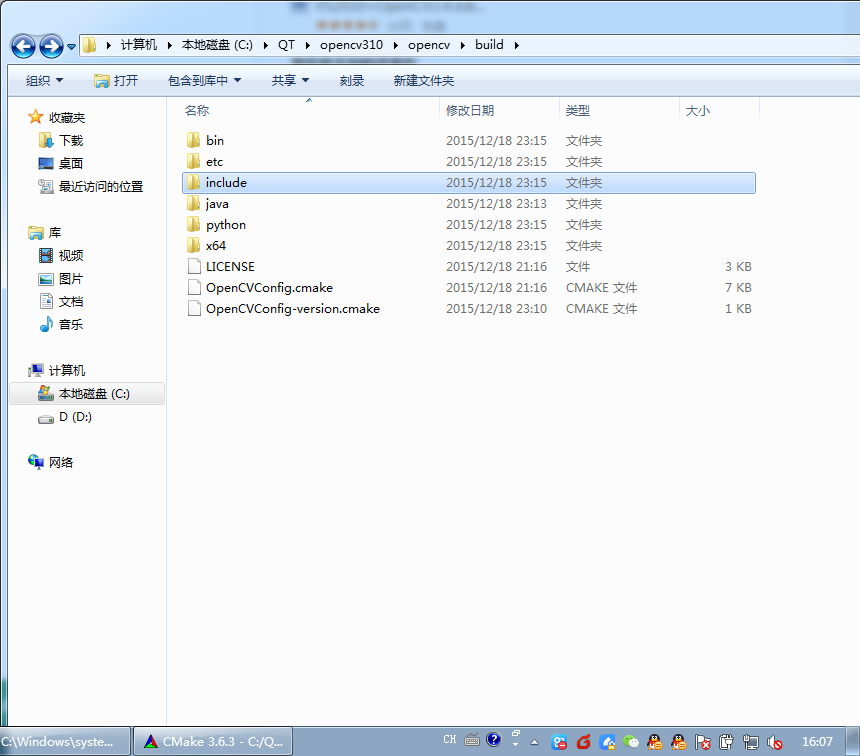
6.打包编译好的文件

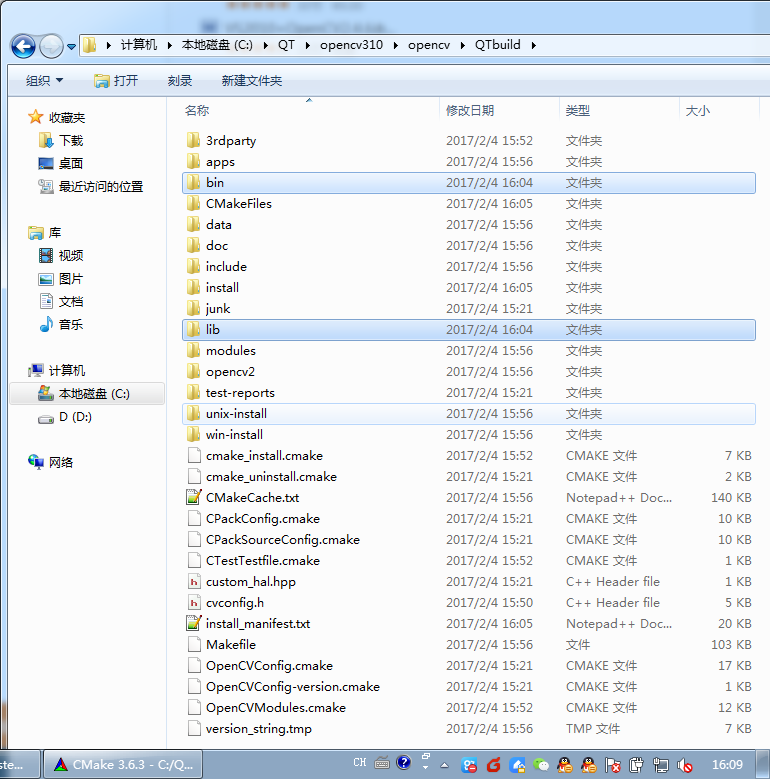
build中的include目录

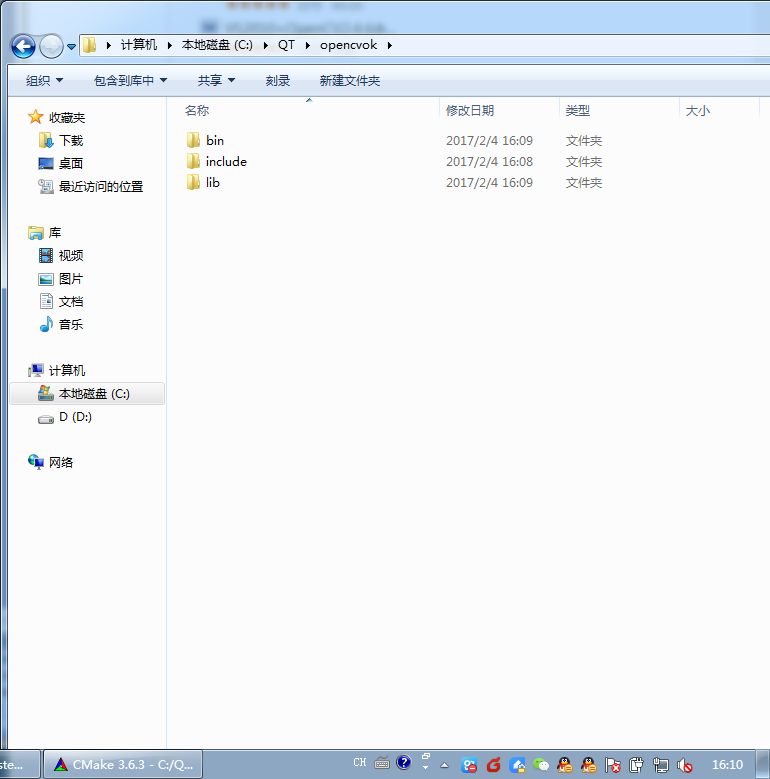
和

CMake目标路径 下的bin和lib文件夹

复制到一个新的目录





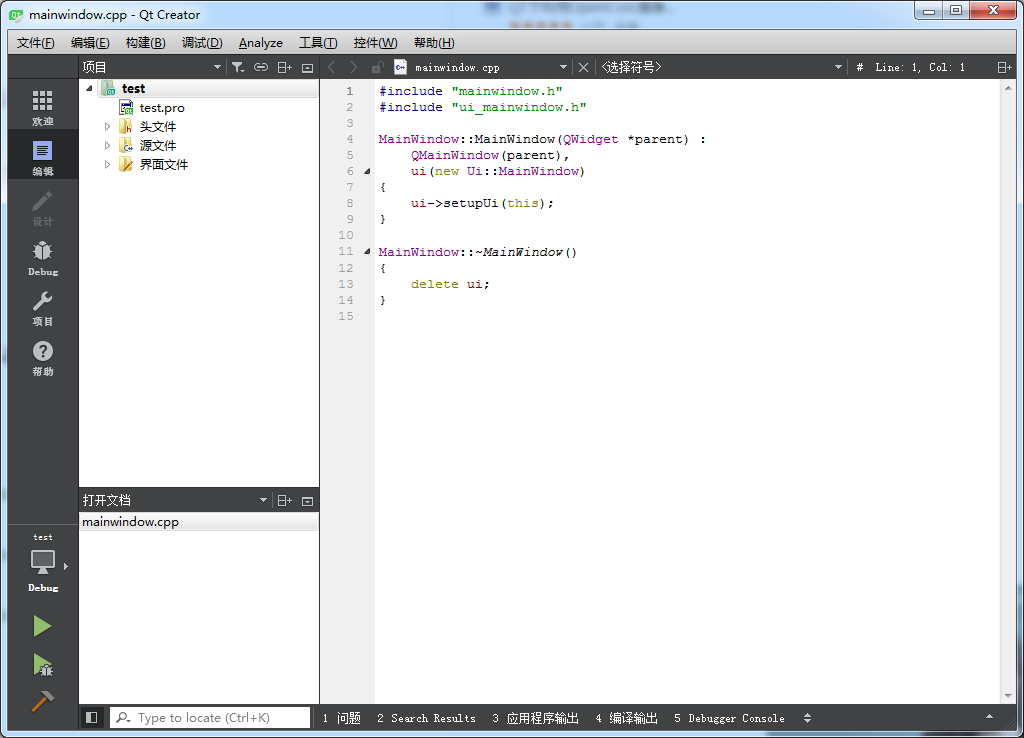


7.在path变量中添加上面的bin路径



8.测试Qt+OpenCV运行正常

1.Qt中新建一个工程，名为test



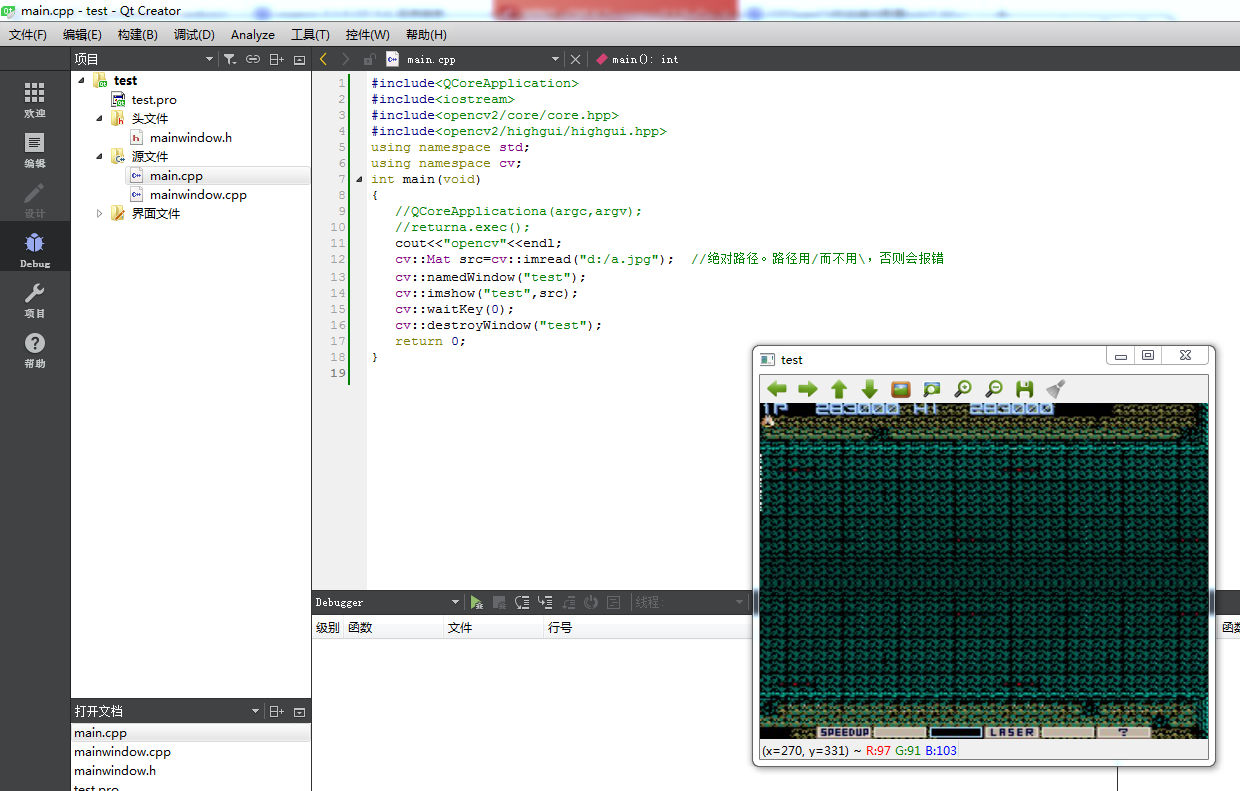
2.在test.pro中添加(路径请自己改)

|  |
| --- |
| INCLUDEPATH += C:\QT\opencvok\include\opencv\                 C:\QT\opencvok\include\opencv2\                 C:\QT\opencvok\include  LIBS += -L C:\QT\opencvok\lib\libopencv\_\*.a |

3.修改main.cpp为

|  |
| --- |
| #include<QCoreApplication>  #include<iostream>  #include<opencv2/core/core.hpp>  #include<opencv2/highgui/highgui.hpp>  using namespace std;  using namespace cv;  int main(void)  {  //QCoreApplicationa(argc,argv);  //returna.exec();  cout<<"opencv"<<endl;  cv::Mat src=cv::imread("d:/a.jpg"); //绝对路径。路径用/而不用\，否则会报错  cv::namedWindow("test");  cv::imshow("test",src);  cv::waitKey(0);  cv::destroyWindow("test");  return 0;  } |

4.测试运行



至此Qt+OpenCV环境搭建完成