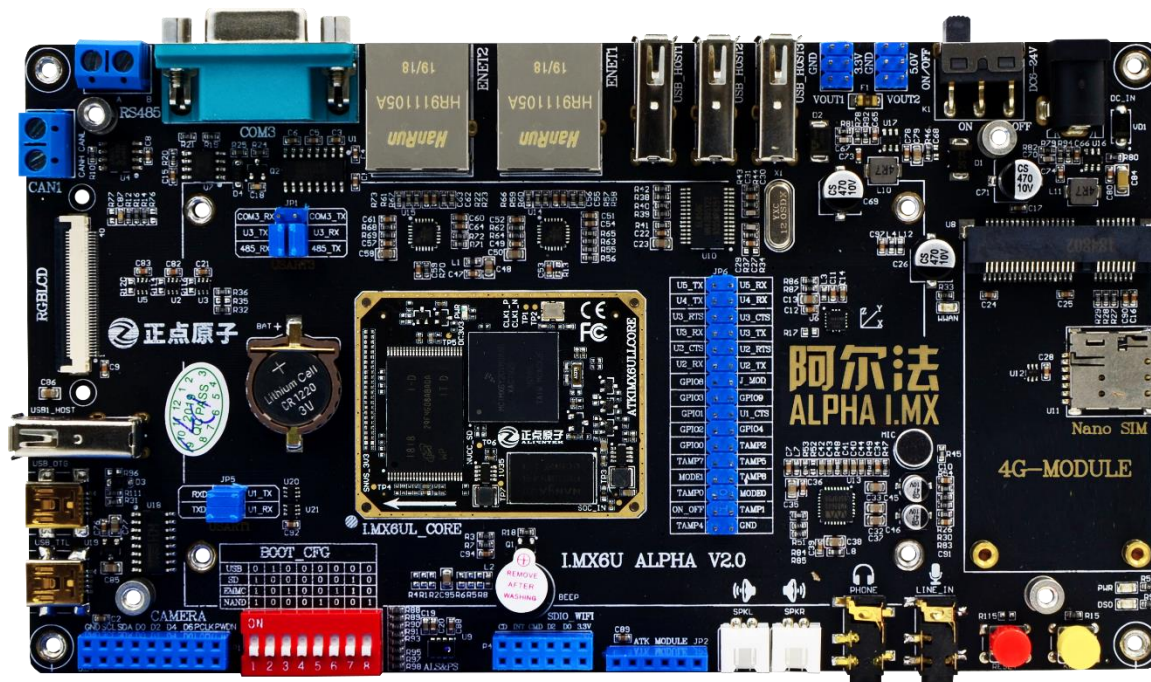


I.MX6U 移植 OpenCV V1.0



正点原子广州市星翼电子科技有限公司

淘宝店铺 1: <http://eboard.taobao.com>

淘宝店铺 2: <http://openedv.taobao.com>

技术支持论坛 (开源电子网) : www.openedv.com

原子哥在线教学: www.yuanzige.com

官方网站: www.alientek.com

最新资料下载链接: <http://www.openedv.com/posts/list/13912.htm>

E-mail: 389063473@qq.com QQ: [389063473](https://www.qq.com/389063473)

咨询电话: [020-38271790](tel:020-38271790)

传真号码: [020-36773971](tel:020-36773971)

团队: [正点原子团队](#)

正点原子, 做最全面、最优秀的嵌入式开发平台软硬件供应商。

友 情 提 示

如果您想及时免费获取“正点原子”最新资讯, 敬请关注正点原子微信公众平台, 我们将及时给您发布最新消息和重要资料。



关注方法:

- (1) 微信“扫一扫”, 扫描右侧二维码, 添加关注
- (2) 微信→添加朋友→公众号→输入“正点原子”→关注
- (3) 微信→添加朋友→输入“alientek_stm32”→关注



文档更新说明

版本	版本更新说明	负责人	校审	发布日期
V1.0	初稿:	正点原子 linux 团队		2020.2.21

目录

前言	V
1、安装交叉编译工具	VI
2、下载 OpenCV 源码	7
3、搭建 OpenCV 的编译环境	7
4、编译 OpenCV 源码	14
5、Qt 例程使用 OpenCV	18
5.1 编译 QOpenCV 工程	18
5.2 测试 QOpenCV 工程	20

前言

本文将介绍 OpenCV 的移植方法，及结合 Qt 例程去进一步学习 OpenCV。网上的移植方法也很多，但是有些过于老旧等。本文将与大家一起学习移植 OpenCV。编译过程十分简单，写这个文档也是为了大家少走弯路。下面带大家一起移植 OpenCV 到正点原子 I.mx6ULL 开发板 ALPHA 上。

- Ubuntu 版本: Ubuntu14.04 64bits 、 Ubuntu16.04 64bits。其它版本 Ubuntu 没测试过。
- 交叉编译工具: arm-linux-gnueabi- 64bit

1、安装交叉编译工具

注意，这里不能使用 NXP yocto 所使用的交叉编译器 arm-poky-linux-gnueab，具体原因未知。本次使用的是《【正点原子】I.MX6U 嵌入式 Linux 驱动开发指南 V1.x》的第 4.3 小节第三方编译器 arm-linux-gnueabihf。

由于作者的 Ubuntu 是 64 位的，所以选择的是 64 位的编译工具链。请参照《【正点原子】I.MX6U 嵌入式 Linux 驱动开发指南 V1.x》的第 4.3 小节配置好交叉编译工具链。

2、下载 OpenCV 源码

OpenCV 官方源码下载链接为 <https://opencv.org/releases/> 可以看到有很多版本的 OpenCV 源码下载。本次选择的是 opencv-3.4.1 版本源码，其他版本请自行选择编译，不确保源码编译不会出错，出错请自行解决。

由于 opencv-3.4.1 源码压缩包在官网下载的慢，而且经常是下载失败。大家理解下，是国外网站。作者本人将源码下载下来上传到自己的百度云。请将下面地址复制到浏览器打开。

链接: https://pan.baidu.com/s/1eeUWWOMf6G6xOe8F4c9_kQ

提取码: kym5

如链接失效，请联系 QQ 1252699831

3、搭建 OpenCV 的编译环境

从第 2 节下载好源码后。在 Ubuntu 新建一个 opencv 文件夹。拷贝源码到 opencv 这个目录。如下图

```
alientek@ubuntu:~/opencv$ ls
opencv-3.4.1.tar.gz
alientek@ubuntu:~/opencv$
```

图 3.1 把 opencv 源码拷贝到 opencv 目录下

使用 tar 指令解压，得到一个 opencv-3.4.1 文件夹。

```
alientek@ubuntu:~/opencv$ tar xf opencv-3.4.1.tar.gz
alientek@ubuntu:~/opencv$ ls
opencv-3.4.1  opencv-3.4.1.tar.gz
alientek@ubuntu:~/opencv$
```

图 3.2 解压 opencv 源码

在 opencv 文件夹下创建 build、install 文件夹。

build 文件夹: 构建目录

install 文件夹: 安装目录

```
alientek@ubuntu:~/opencv$ mkdir build install
alientek@ubuntu:~/opencv$ ls
build  install  opencv-3.4.1  opencv-3.4.1.tar.gz
alientek@ubuntu:~/opencv$
```

图 3.3 新建构建目录、安装目录

安装 cmake 和 cmake-gui 工具

Ubuntu# `sudo apt-get install cmake cmake-qt-gui cmake-curses-gui`

```
alientek@ubuntu:~$ sudo apt-get install cmake cmake-qt-gui cmake-curses-gui
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  cmake-data libjsoncpp1
Suggested packages:
  codeblocks eclipse ninja-build
The following NEW packages will be installed:
  cmake cmake-curses-gui cmake-data cmake-qt-gui libjsoncpp1
0 upgraded, 5 newly installed, 0 to remove and 448 not upgraded.
Need to get 6,485 kB of archives.
After this operation, 29.5 MB of additional disk space will be used.
Do you want to continue? [Y/n] y
Get:1 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 cmake-data all 3.5.1-1ubuntu3 [1,121 kB]
Get:2 http://mirrors.aliyun.com/ubuntu xenial/main amd64 libjsoncpp1 amd64 1.7.2-1 [73.0 kB]
Get:3 http://mirrors.aliyun.com/ubuntu xenial-updates/main amd64 cmake amd64 3.5.1-1ubuntu3 [2,623 kB]
Get:4 http://mirrors.aliyun.com/ubuntu xenial-updates/universe amd64 cmake-curses-gui amd64 3.5.1-1ubuntu3 [1,276 kB]
Get:5 http://mirrors.aliyun.com/ubuntu xenial-updates/universe amd64 cmake-qt-gui amd64 3.5.1-1ubuntu3 [1,393 kB]
Fetched 6,485 kB in 16s (394 kB/s)
Selecting previously unselected package cmake-data.
(Reading database ... 179855 files and directories currently installed.)
Preparing to unpack .../cmake-data_3.5.1-1ubuntu3_all.deb ...
Unpacking cmake-data (3.5.1-1ubuntu3) ...
Selecting previously unselected package libjsoncpp1:amd64.
Preparing to unpack .../libjsoncpp1_1.7.2-1_amd64.deb ...
Unpacking libjsoncpp1:amd64 (1.7.2-1) ...
Selecting previously unselected package cmake.
Preparing to unpack .../cmake_3.5.1-1ubuntu3_amd64.deb ...
Unpacking cmake (3.5.1-1ubuntu3) ...
Selecting previously unselected package cmake-curses-gui.
```

图 3.4 安装 cmake 等工具

进入 build 目录下, 执行指令 cmake-gui, 开始配置 OpenCV 的环境。

Ubuntu# cmake-gui

```
alientek@ubuntu:~/opencv/build$ cmake-gui
```

图 3.5 运行 cmake-gui 图形化工具

执行完成后会出现图形化工具 cmake-gui。

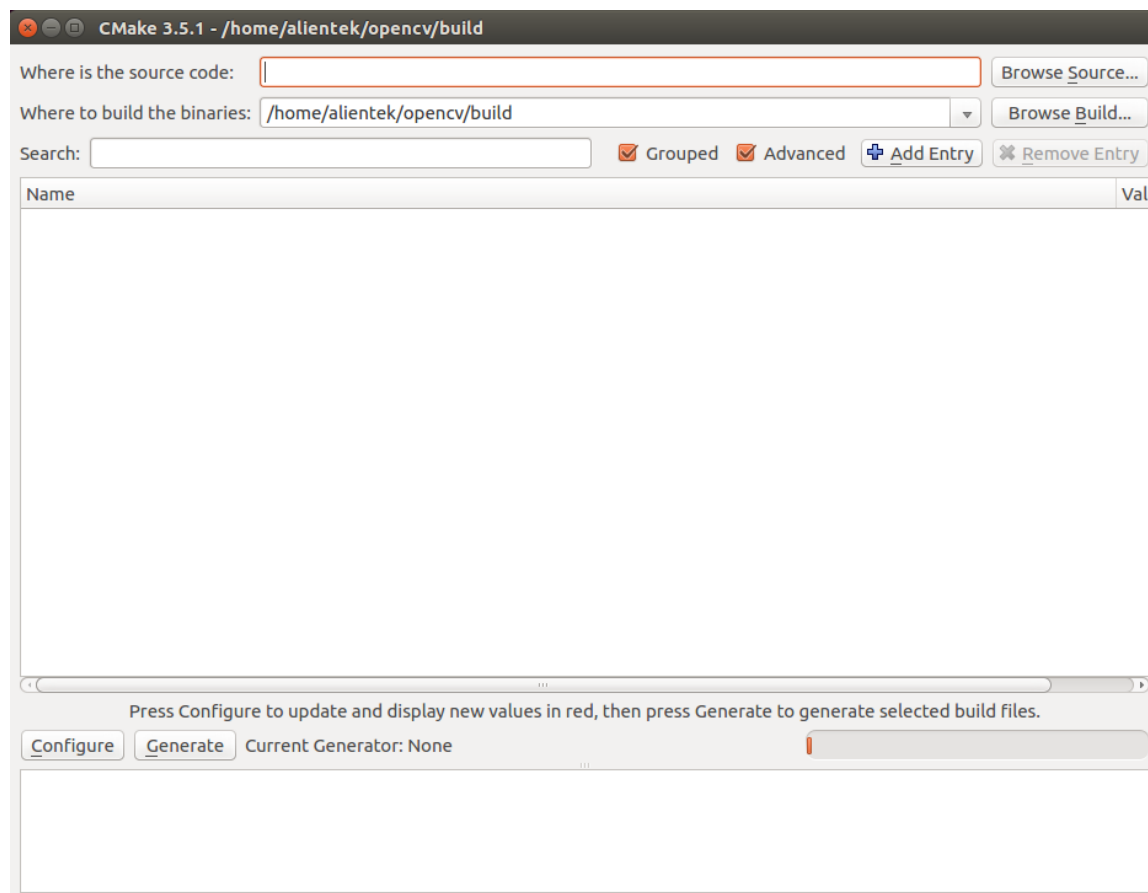


图 3.6 图形化工具 cmake-gui

指定我们源码的所在路径和构建目录。勾选 Advanced，再点击 Configure 或者 Generate。

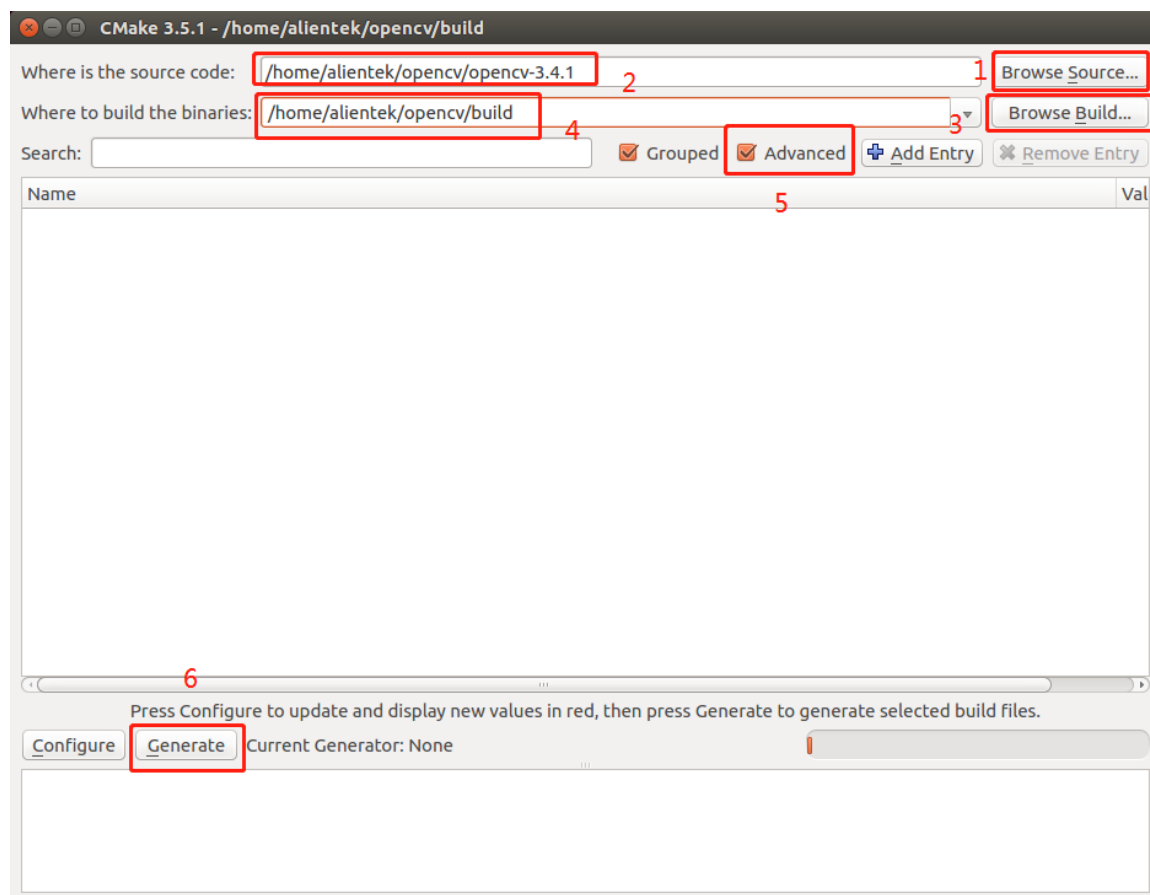


图 3.7 指定源码目录及构建目录

选择 Unix Makefiles，然后选择 Specify options for cross-compiling，再点击 Next。

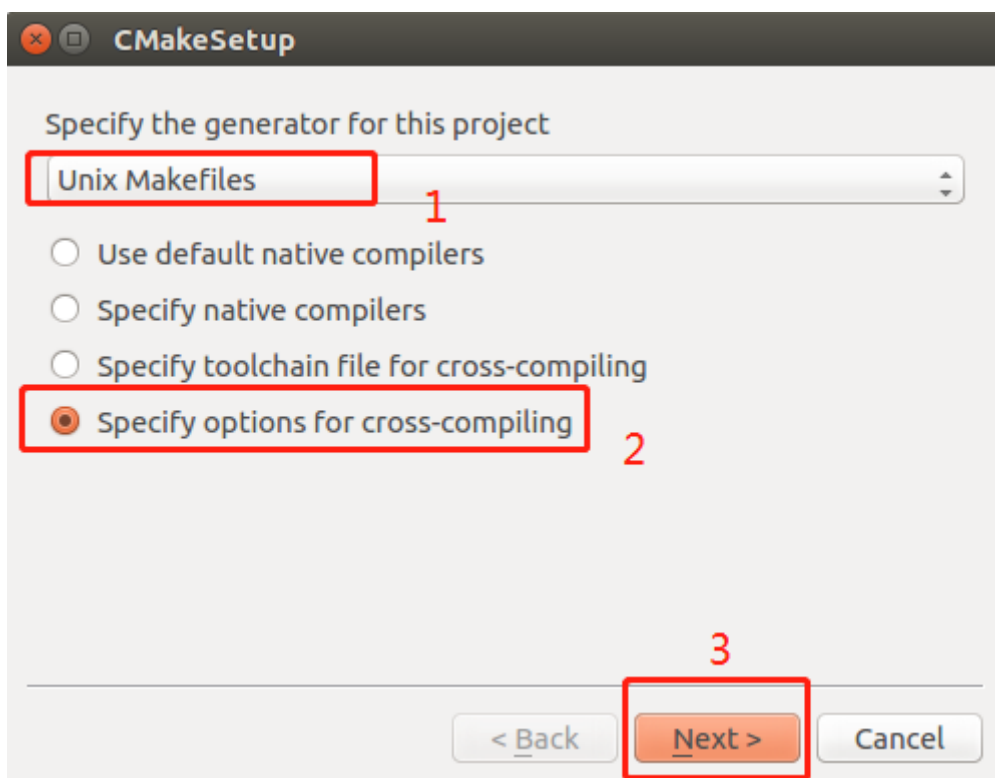


图 3.8 选择 Unix 类型 Makefiles 及指定具体交叉编译器路径

填写交叉编译器的路径，注意红色部分请填写自己的交叉编译器路径。

Operation System: linux

Compilers C: /home/alientek/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/f/bin/arm-linux-gnueabi/f-gcc

Compilers C++: /home/alientek/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/f/bin/arm-linux-gnueabi/f-g++

Target Root: /home/alientek/gcc-linaro-4.9.4-2017.01-x86_64_arm-linux-gnueabi/f/bin

指定交叉编译工具的具体路径，按个人路径指定，如下图。其他默认即可。

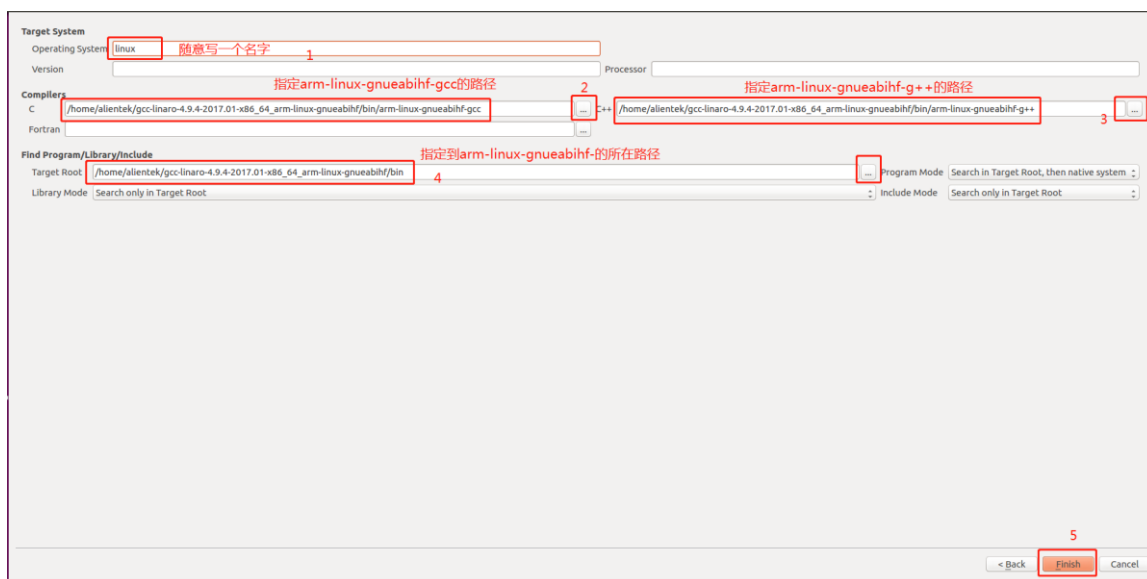


图 3.9 指定交叉编译工具的具体路径

下面就是您配置的信息

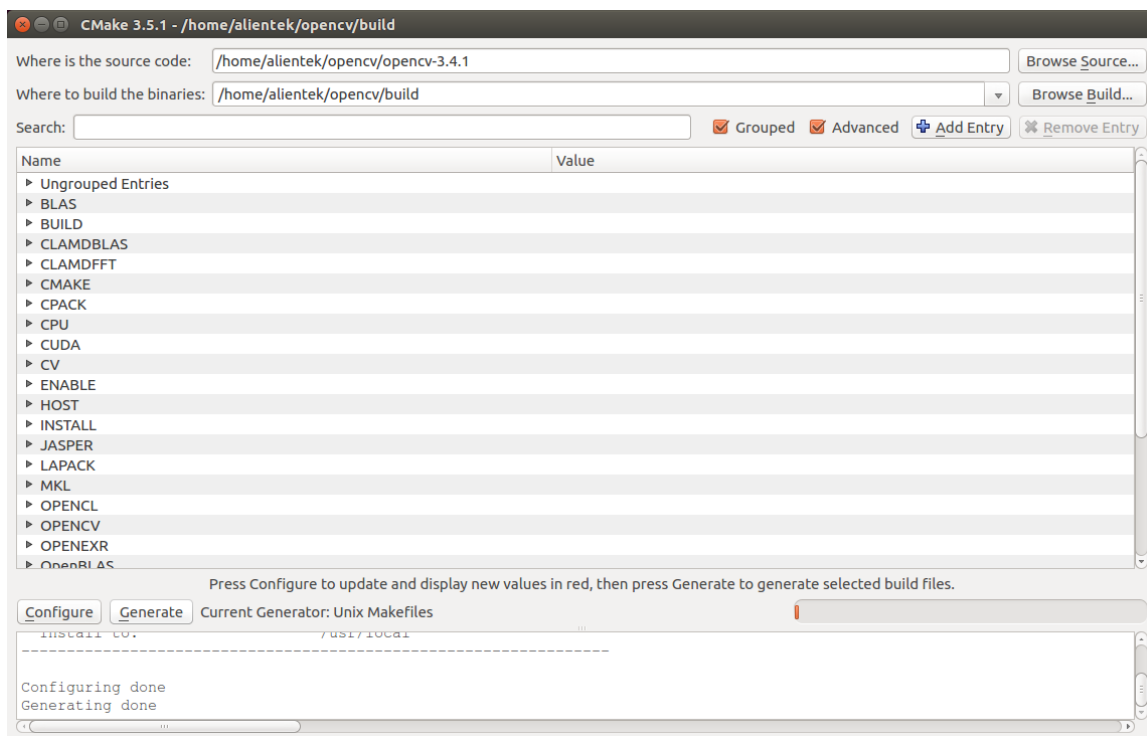


图 3.10 配置信息

我们还需要配置一些信息，点击如下图 CMAKE 处。

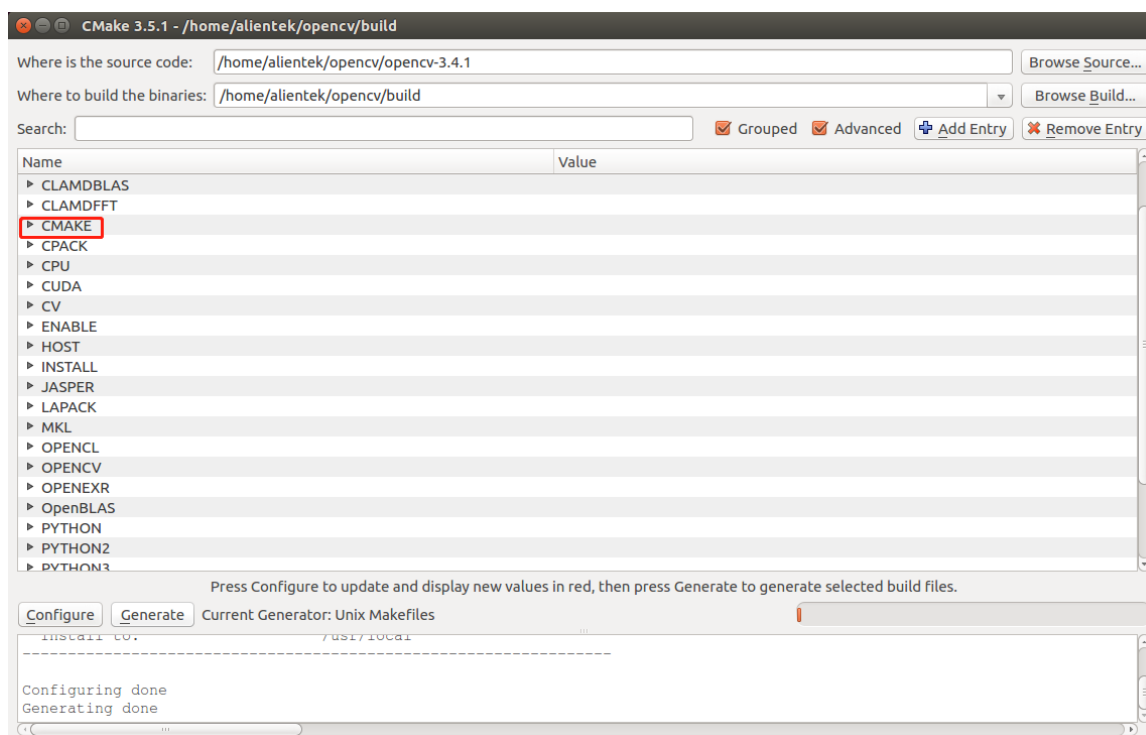


图 3.11 点击 CMAKE

在 CMAKE_EXE_LINKER_FLAGS 处添加上 -lpthread -lrt -ldl。添加这些是指定依赖库的链接参数。

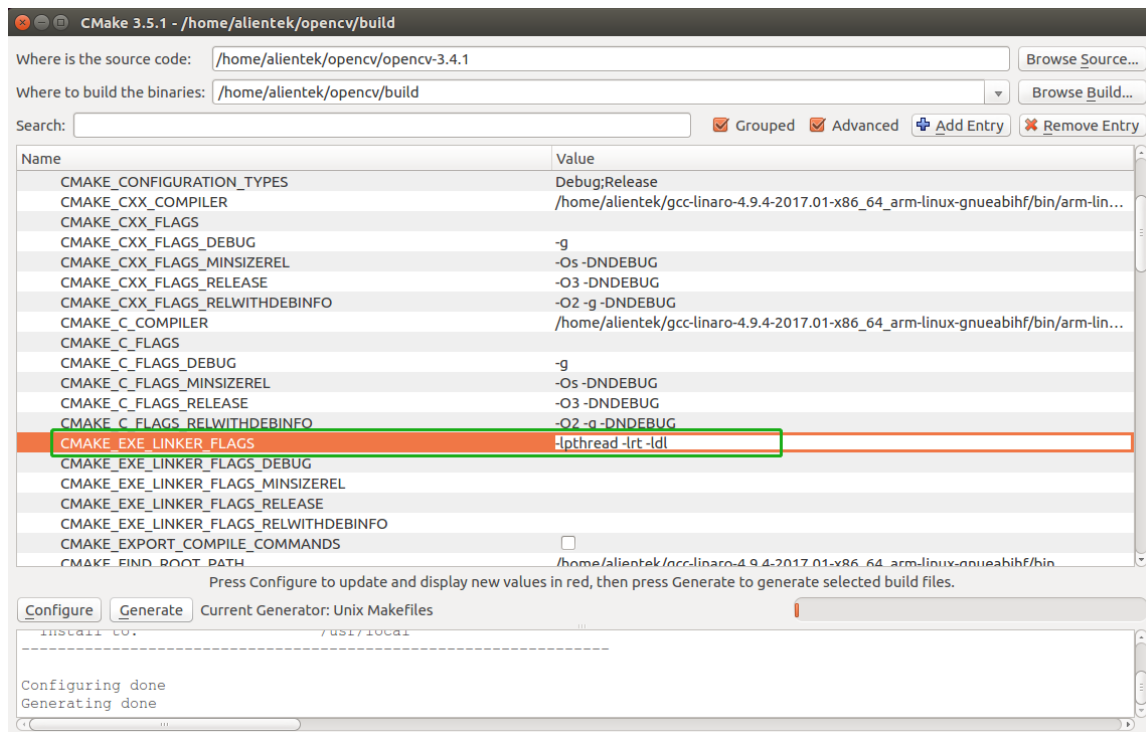


图 3.12 配置依赖库参数

再在 CMAKE_INSTALL_PREFIX 处指定安装目录，我们在上面已经新建了 install 安装目录。我们直接指定到该目录即可。如果不指定，它会默认安装到 Ubuntu 系统目录/usr/local 下。这样

会直接替换我们/usr/local 下的库。

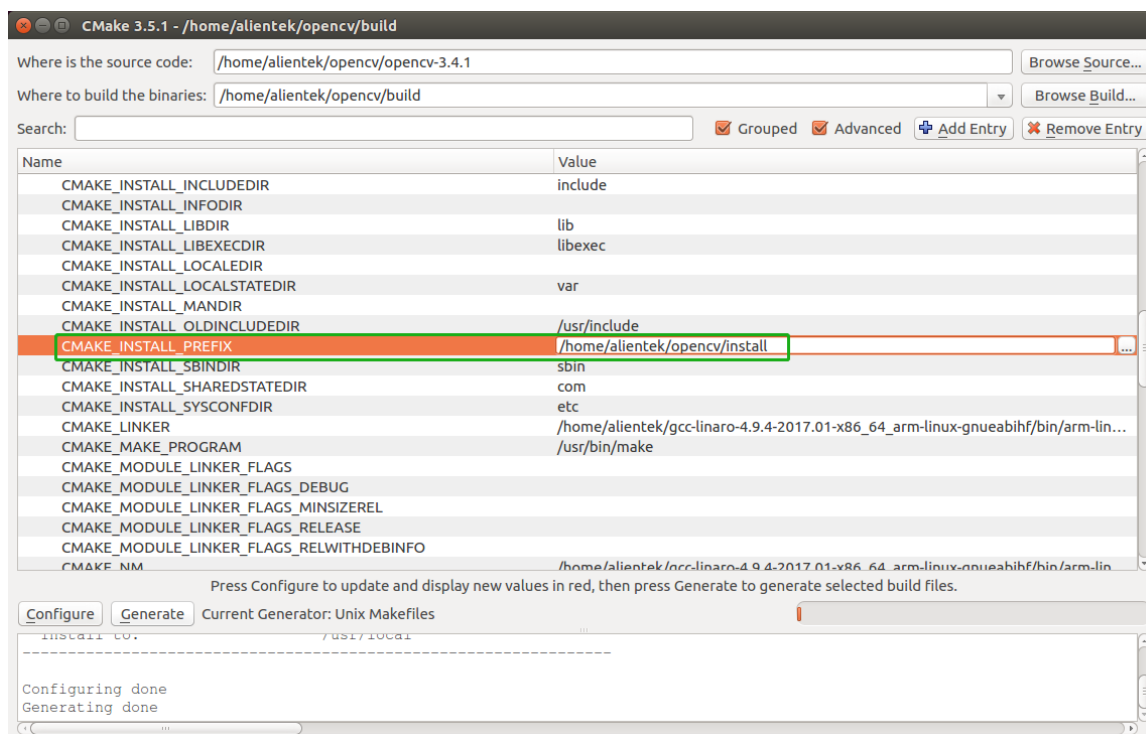


图 3.13 指定安装路径

至此我们配置完，现在我们需要生成 Makefile 等文件，我们先点击 Configure，再点击 Generate 就可以生成了。

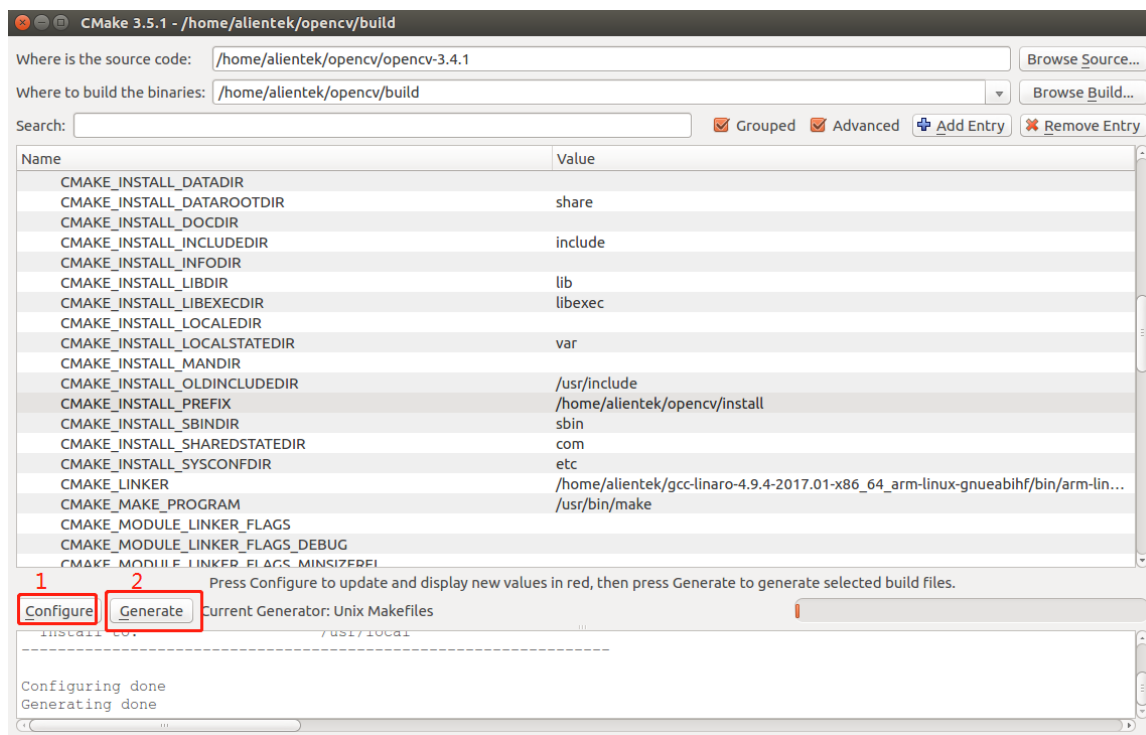


图 3.14 配置生成 Makefile 等文件

如下图，点击了 Generate 后看到信息如下图一样，表明生成成功，若出现报错，请根据对话

框里的内容指定的文件去自行去检查错误！一般都是交叉编译器路径不对造成的。配置完成生成文件后，我们点击左上角关闭这个图形化配置界面即可。

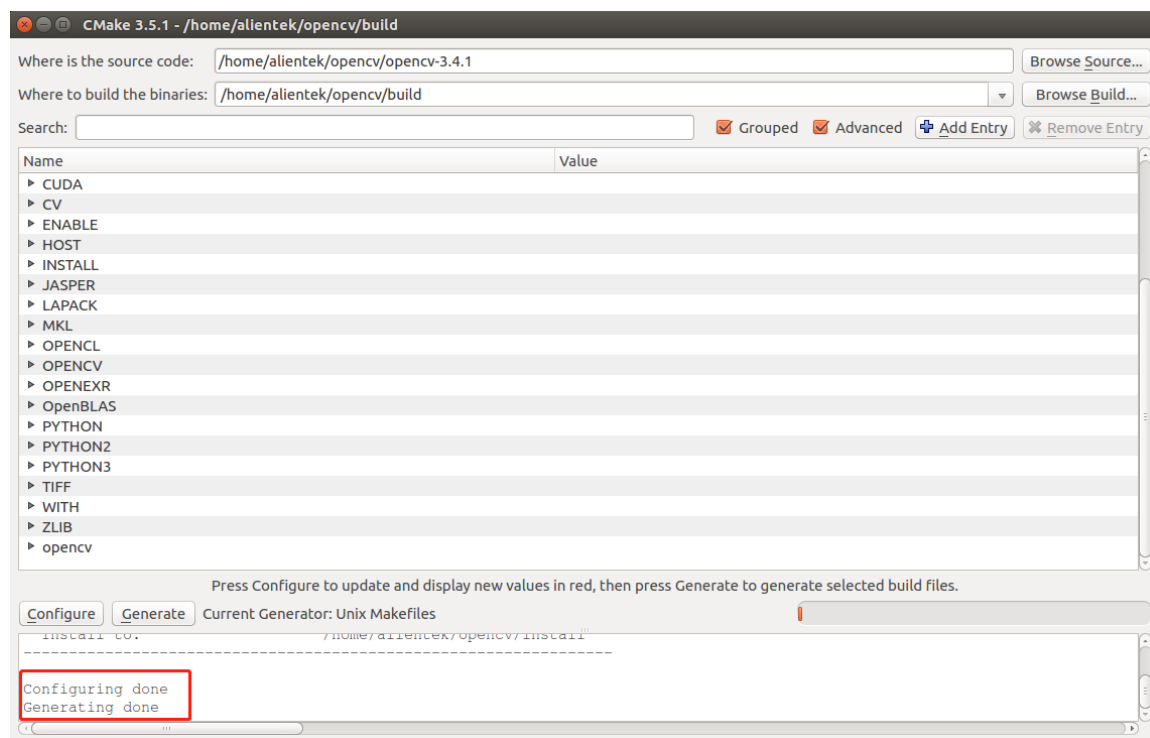


图 3.15 检查配置成功与否

4、编译 OpenCV 源码

在第 3 节配置完成后，先不要急着输入 make。首先在源码目录 opencv-3.4.1/3rdparty/protobuf/src/google/protobuf/stubs/common.cc 这个文件下添加#define HAVE_PTHREAD 宏定义才可以编译的过。具体原因是 HAVE_PTHREAD 宏定义了 pthread 库。在如下位置添加即可。

```

16 // contributors may be used to endorse or promote products derived from
17 // this software without specific prior written permission.
18 //
19 // THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS
20 // "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT
21 // LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR
22 // A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT
23 // OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL,
24 // SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT
25 // LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE,
26 // DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY
27 // THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT
28 // (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE
29 // OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.
30
31 // Author: kenton@google.com (Kenton Varda)
32
33 #define HAVE_PTHREAD
34 #include <google/protobuf/message_lite.h> // TODO(gerbens) ideally remove this.
35 #include <google/protobuf/stubs/common.h>
36 #include <google/protobuf/stubs/once.h>
37 #include <google/protobuf/stubs/status.h>
38 #include <google/protobuf/stubs/stringpiece.h>
39 #include <google/protobuf/stubs/strutil.h>
40 #include <google/protobuf/stubs/int128.h>
41 #include <errno.h>
42 #include <sstream>
43 #include <stdio.h>
44 #include <vector>
45

```

图 4.1 修改 common.cc 文件

修改完成后，返回于是 build 目录下，可以看到 build 目录下已经准备了构建文件。我们直接输入 make 构建即可。

```

allientek@ubuntu:~/opencv/build$ ls
3rdparty      CMakeDownloadLog.txt  CMakeVars.txt      CPackSourceConfig.cmake  cv_cpu_config.h      junk      opencv2      OpenCVModules.cmake
apps          CMakeFiles            configured          CTestTestfile.cmake     data                 lib        OpenCVConfig.cmake  opencv_tests_config.hpp
bin           cmake_install.cmake   CopyOfCMakeCache.txt  custom_hal.hpp           doc                  Makefile   OpenCVConfig-version.cmake  unix-install
CMakeCache.txt  cmake_uninstall.cmake  CPackConfig.cmake   cvconfig.h              include              modules    OpenCVGenPkgConfig.info.cmake  version_string.tnp
allientek@ubuntu:~/opencv/build$

```

图 4.2 build 目录准备用于构建的文件

输入 make 开始编译

Ubuntu# make -j 16

```

allientek@ubuntu:~/opencv/build$ make -j 16
Scanning dependencies of target zlib
Scanning dependencies of target libjpeg
Scanning dependencies of target libprotobuf
Scanning dependencies of target gen-pkgconfig
Scanning dependencies of target libjasper
Scanning dependencies of target libwebp
[ 0%] Generate opencv.pc
[ 0%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/compress.c.obj
[ 0%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/adler32.c.obj
[ 0%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/crc32.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/deflate.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/gzclose.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/gzlib.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/gzread.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/gzwrite.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/inflate.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/infrees.c.obj
[ 1%] Building C object 3rdparty/zlib/CMakeFiles/zlib.dir/inflate.c.obj
[ 1%] Building C object 3rdparty/libjpeg/CMakeFiles/libjpeg.dir/jccolor.c.obj
[ 1%] Building C object 3rdparty/libjasper/CMakeFiles/libjasper.dir/jas_version.c.obj

```

图 4.3 开始构建

编译完成如下图，如果有出错，先删除 build 目录下的所有文件，按第 3 节重新再来一次！cmake-gui 尽量一次配置成功！

```
[ 95%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_undistort_badarg.cpp.obj
[ 95%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_posit.cpp.obj
[ 95%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_cornerssubpix.cpp.obj
[ 96%] Linking CXX executable ../../bin/opencv_traincascade
[ 96%] Linking CXX executable ../../bin/opencv_interactive-calibration
[ 96%] Built target opencv_traincascade
[ 96%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_decompose_projection.cpp.obj
[ 96%] Linking CXX executable ../../bin/opencv_test_videostab
[ 96%] Built target opencv_interactive-calibration
[ 97%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_main.cpp.obj
[ 97%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_affine2d_estimator.cpp.obj
[ 97%] Built target opencv_test_videostab
[ 97%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_modeltest.cpp.obj
[ 97%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_chesscorners_badarg.cpp.obj
[ 98%] Linking CXX shared library ../../lib/libopencv_stitching.so
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_chesscorners_timing.cpp.obj
[ 98%] Built target opencv_stitching
Scanning dependencies of target opencv_perf_stitching
[ 98%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf/perf_estimators.cpp.obj
[ 98%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf/perf_main.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_compose_rt.cpp.obj
Scanning dependencies of target opencv_test_stitching
[ 98%] Building CXX object modules/stitching/CMakeFiles/opencv_test_stitching.dir/test/test_blenders.cpp.obj
[ 98%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf/perf_main.cpp.obj
[ 98%] Building CXX object modules/stitching/CMakeFiles/opencv_test_stitching.dir/test/test_matchers.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_undistort_points.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/opencvcl/test_stereobm.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_undistort.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_fundam.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_chesscorners.cpp.obj
[ 98%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf/perf_stich.cpp.obj
[ 98%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_camera_calibration_badarg.cpp.obj
[ 99%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_homography.cpp.obj
[ 99%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_affine3d_estimator.cpp.obj
[ 99%] Building CXX object modules/calib3d/CMakeFiles/opencv_test_calib3d.dir/test/test_affine_partial2d_estimator.cpp.obj
[ 99%] Building CXX object modules/stitching/CMakeFiles/opencv_test_stitching.dir/test/ocl/test_warpers.cpp.obj
[ 100%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf/opencvcl/perf_stitch.cpp.obj
[ 100%] Building CXX object modules/stitching/CMakeFiles/opencv_perf_stitching.dir/perf/opencvcl/perf_warpers.cpp.obj
[ 100%] Building CXX object modules/stitching/CMakeFiles/opencv_test_stitching.dir/test/test_blenders.cuda.cpp.obj
[ 100%] Linking CXX executable ../../bin/opencv_test_stitching
[ 100%] Linking CXX executable ../../bin/opencv_test_calib3d
[ 100%] Built target opencv_test_stitching
[ 100%] Built target opencv_test_calib3d
[ 100%] Linking CXX executable ../../bin/opencv_test_core
[ 100%] Built target opencv_test_core
[ 100%] Linking CXX executable ../../bin/opencv_perf_stitching
[ 100%] Built target opencv_perf_stitching
[ 100%] Linking CXX executable ../../bin/opencv_perf_core
[ 100%] Built target opencv_perf_core
allientek@ubuntu:~/opencv/build$
```

图 4.4 构建成功

输入 `make install` 安装到指定目录

Ubuntu# `make install`


```

alientek@ubuntu:~/opencv/build$ make install
[ 0%] Built target gen-pkgconfig
[ 2%] Built target zlib
[ 6%] Built target libjpeg
[ 9%] Built target libtiff
[ 19%] Built target libwebp
[ 22%] Built target libjasper
[ 23%] Built target libpng
[ 29%] Built target IlmImf
[ 36%] Built target libprotobuf
[ 42%] Built target opencv_core
[ 47%] Built target opencv_imgproc
[ 48%] Built target opencv_imgcodecs
[ 48%] Built target opencv_videoio
[ 49%] Built target opencv_highgui
[ 49%] Built target opencv_ts
[ 51%] Built target opencv_test_core
[ 54%] Built target opencv_perf_core
[ 54%] Built target opencv_flann
[ 54%] Built target opencv_test_flann
[ 57%] Built target opencv_perf_imgproc
[ 61%] Built target opencv_test_imgproc
[ 62%] Built target opencv_ml
[ 63%] Built target opencv_test_ml
[ 64%] Built target opencv_obidetect

```

图 4.5 开始安装

安装成功如下图

```

-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/outlier_rejection.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/deblurring.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/inpainting.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/global_motion.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/wobble_suppression.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/optical_flow.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/ring_buffer.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/motion_stabilizing.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/fast_marching.hpp
-- Installing: /home/alientek/opencv/install/include/opencv2/videostab/log.hpp
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_eye.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_eye_tree_eyeglasses.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_righteye_2splits.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_fullbody.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_frontalface_alt_tree.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_frontalface_alt2.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_lowerbody.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_russian_plate_number.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_frontalcatface.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_frontalface_alt.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_frontalcatface_extended.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_licence_plate_rus_16stages.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_lefteye_2splits.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_upperbody.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_smile.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_frontalface_default.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/haarcascades/haarcascade_profileface.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/lbpcascades/lbpcascade_profileface.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/lbpcascades/lbpcascade_frontalcatface.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/lbpcascades/lbpcascade_frontalface.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/lbpcascades/lbpcascade_frontalface_improved.xml
-- Installing: /home/alientek/opencv/install/share/OpenCV/lbpcascades/lbpcascade_silverware.xml
-- Installing: /home/alientek/opencv/install/bin/opencv_traincascade
-- Installing: /home/alientek/opencv/install/bin/opencv_createsamples
-- Installing: /home/alientek/opencv/install/bin/opencv_annotation
-- Installing: /home/alientek/opencv/install/bin/opencv_visualisation
-- Installing: /home/alientek/opencv/install/bin/opencv_interactive-calibration
-- Installing: /home/alientek/opencv/install/bin/opencv_version
alientek@ubuntu:~/opencv/build$

```

图 4.6 安装成功

进入 install 目录下有如下几个目录表明成功。到此完成编译！

```
alientek@ubuntu:~/opencv/install$ ls
bin  include  lib  LICENSE  share
alientek@ubuntu:~/opencv/install$
```

图 4.7 检查 install 目录下的文件

5、Qt 例程使用 OpenCV

5.1 编译 QOpenCV 工程

正点原子提供 OpenCV Qt 例程，例程作者已经上传到百度云。请在复制链接到浏览器打开！

链接：<https://pan.baidu.com/s/1z-CdhbcmgdpPZMzo9WpZUA>

提取码：6rn0，如链接失效请联系作者 QQ 1252699831

下载完成后，把 QOpenCV 文件夹放到 install 文件夹下。

```
alientek@ubuntu:~/opencv/install$ ls
bin  include  lib  LICENSE  QOpenCV  share
alientek@ubuntu:~/opencv/install$
```

图 5.1.1 将 QOpenCV 放于 install 文件夹下

为什么要放到 install 文件夹下呢，这是因为我们在工程 QOpenCV.pro 文件里定义了 lib 库的路径是在 QOpenCV.pro 的 ../../lib 下。我们打开 QOpenCV.pro 工程来看。

进入 QOpenCV/opencvdemo 目录下。

Ubuntu# vi QOpenCV.pro

注意 INCLUDEPATH 需要修改为自己 OpenCV 安装文件夹下的 include 路径！“#”号的是注释，不编译。

```
25
26 SOURCES += main.cpp\
27             mainwindow.cpp
28
29 HEADERS += mainwindow.h
30
31 INCLUDEPATH += /home/alientek/opencv/install/include
32 LIBS += ../../lib/libopencv_core.so \
33          ../../lib/libopencv_highgui.so \
34          ../../lib/libopencv_imgproc.so \
35          ../../lib/libopencv_videoio.so \
36          ../../lib/libopencv_imgcodecs.so \
37          -lpthread
38 #LIBS += /home/alientek/opencv/install/lib/libopencv_core.so \
39 #        /home/alientek/opencv/install/lib/libopencv_highgui.so \
40 #        /home/alientek/opencv/install/lib/libopencv_imgproc.so \
41 #        /home/alientek/opencv/install/lib/libopencv_videoio.so \
42 #        /home/alientek/opencv/install/lib/libopencv_imgcodecs.so \
43 #        -lpthread
44
45 RESOURCES += \
46             iconres.qrc
47
```

图 5.1.2 查看 QOpenCV.pro 文件

我们修改好后，将开始编译这个 Qt 工程。

请根据【正点原子】IMX6U 用户快速体验 V1.x.pdf 第四章来搭建 Qt 工程的交叉编译工具链。

搭建完成后，在 Ubuntu 终端执行下面的指令来使能 Qt 交叉编译的环境变量。

```
Ubuntu# source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
```

在 OOpenCV.pro 同目录下，执行 qmake 生成 Makefile，用于编译 Qt 工程。

```
allentek@ubuntu:~/opencv/Install/00opencv/opencvdemo5$ sudo /opt/fsl-lmx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allentek@ubuntu:~/opencv/Install/00opencv/opencvdemo5$ ls
allentek@ubuntu:~/opencv/Install/00opencv/opencvdemo5$ iconres.qrc main.cpp mainwindow.cpp mainwindow.h QOpenCV.pro resources
allentek@ubuntu:~/opencv/Install/00opencv/opencvdemo5$ qmake
```

图 5.1.3 使用 qmake 生成 Makefile

再执行 make 编译 Qt 工程

```
alientek@ubuntu:~/opencv/install/Q0penCV/opencvdemo$ qmake
alientek@ubuntu:~/opencv/install/Q0penCV/opencvdemo$ make -j 16
```

图 5.1.4 执行 make

编译完成如下图

[illegible]

图 5.1.5 编译完成

编译完成后在当前目录生成可执行文件 `QOpenCV`，我们需要把它和 `resources` 文件夹拷贝到文件系统里。

```
allentek@ubuntu:~/opencv/install/00opencv/opencvdemo$ ls
lconres.qrc main.cpp main.o mainmainwindow.cpp mainmainwindow.h mainmainwindow.o Makefile moc_mainmainwindow.cpp moc_mainmainwindow.o 00opencv 00opencv.pro qrc_lconres.cpp qrc_lconres.o resource
allentek@ubuntu:~/opencv/install/00opencv/opencvdemo$
```

图 5.1.6 拷贝 QOpenCV 和 resources 文件夹到开发板的/home/root 目录下

补充说明:

如果我们不是把 QOpenCV 这个工程放置于 lib 同级目录下, QOpenCV.pro 就会编译找不到路径。而且库必须是位于 ../../lib 下, 就算我们指定 lib 具体路径, 如果不在 ../../lib 就会编译报如下警告! 这是因为 ../../lib/libopencv_core.so 和 ../../lib/libopencv_imgproc.so 这两个库它自带路径 ../../lib。

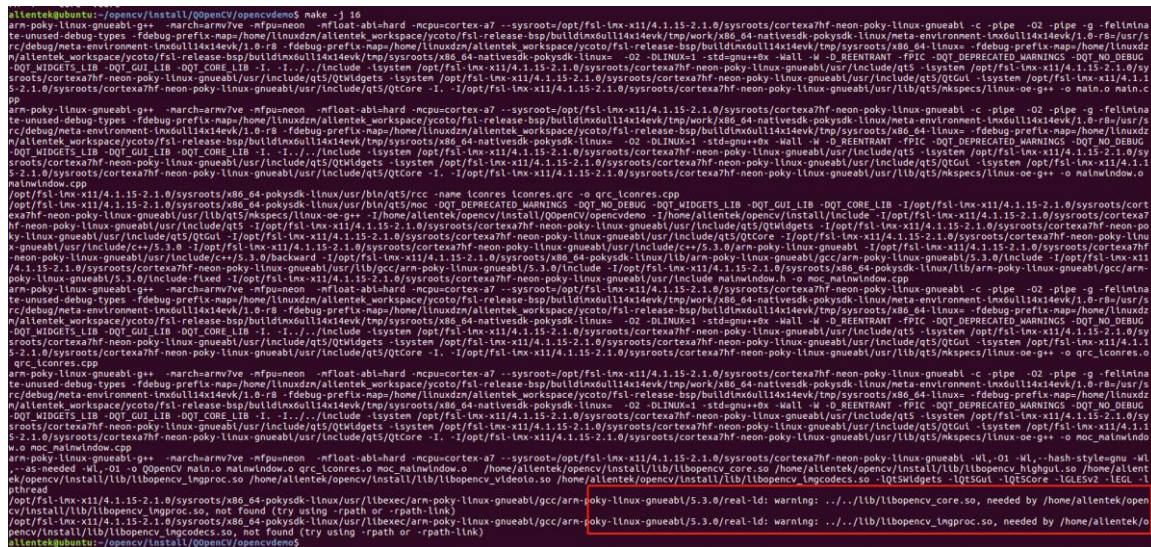


图 5.1.7 编译报警告的情况

为了避免这个警告，和编译出来的程序可能找不到路径，所以才在工程设置成下面这样的。

```
INCLUDEPATH += /home/alientek/opencv/install/include
LIBS += ../../lib/libopencv_core.so \
        ../../lib/libopencv_highgui.so \
        ../../lib/libopencv_imgproc.so \
        ../../lib/libopencv_videoio.so \
        ../../lib/libopencv_imgcodecs.so \
        -lpthread
```

图 5.1.8 解决方法

5.2 测试 QOpenCV 工程

请使用正点原子出厂的文件系统！自己编译的 busybox 或 buildroot 文件系统是不具备 Qt 库的，Qt 应用程序是运行不了的。

首先在串口终端里执行

USER# /etc/init.d/psplash.sh //关闭原来的桌面应用程序

我们需要拷贝第 4 节 install/lib 文件夹下的 opencv 相关库拷贝到开发板的 /lib 目录下。（这里一定是 lib 目录下，其它目录不行，即使设置了环境变量。具体原因未知！）

把下面这些库拷贝到文件系统路径 /lib 下。

```
alientek@ubuntu:~/opencv/install/lib$ ls
libopencv_calib3d.so  libopencv_features2d.so  libopencv_imgcodecs.so  libopencv_objdetect.so  libopencv_stitching.so  libopencv_videoio.so
libopencv_core.so    libopencv_flann.so       libopencv_imgproc.so    libopencv_photo.so      libopencv_superres.so  libopencv_videostab.so
libopencv_dnn.so     libopencv_highgui.so     libopencv_ml.so         libopencv_shape.so      libopencv_video.so
alientek@ubuntu:~/opencv/install/lib$
```

图 5.2.1 OpenCV 库

如下图作者已经拷贝到开发板的 /lib 目录下了

```
root@ATK-IMX6U:~# ls /lib/libopencv.*
/lib/libopencv_calib3d.so  /lib/libopencv_features2d.so  /lib/libopencv_imgcodecs.so  /lib/libopencv_objdetect.so  /lib/libopencv_stitching.so  /lib/libopencv_videoio.so
/lib/libopencv_core.so    /lib/libopencv_flann.so       /lib/libopencv_imgproc.so    /lib/libopencv_photo.so      /lib/libopencv_superres.so  /lib/libopencv_videostab.so
/lib/libopencv_dnn.so     /lib/libopencv_highgui.so     /lib/libopencv_ml.so         /lib/libopencv_shape.so      /lib/libopencv_video.so
root@ATK-IMX6U:~#
```

图 5.2.2 拷贝 OpenCV 库到开发板的 /lib 目录下

同时拷贝了 5.1 小节编译出来 Qt opencv 应用程序及 resouces 资源文件到/home/root 目录下。如下图。

```
root@ATK-IMX6U:~# ls
QOpenCV driver resources shell
root@ATK-IMX6U:~#
```

图 5.2.3 拷贝 QOpenCV 应用程序及资源文件到开发板的/home/root 目录下
在串口终端里直接执行 QOpenCV Qt 应用程序

USER# ./QOpenCv

```
root@ATK-IMX6U:~# ./QOpenCV
```

图 5.2.4 执行 QOpenCV 应用程序

lcd 显示屏上已经出现应用程序的示例。因为应用程序是设定 800*480 的分辨率的，不做兼容。如果不符合您的屏的尺寸大小，请自行修改。
如下图，点击原图，应用程序则不对图像处理。

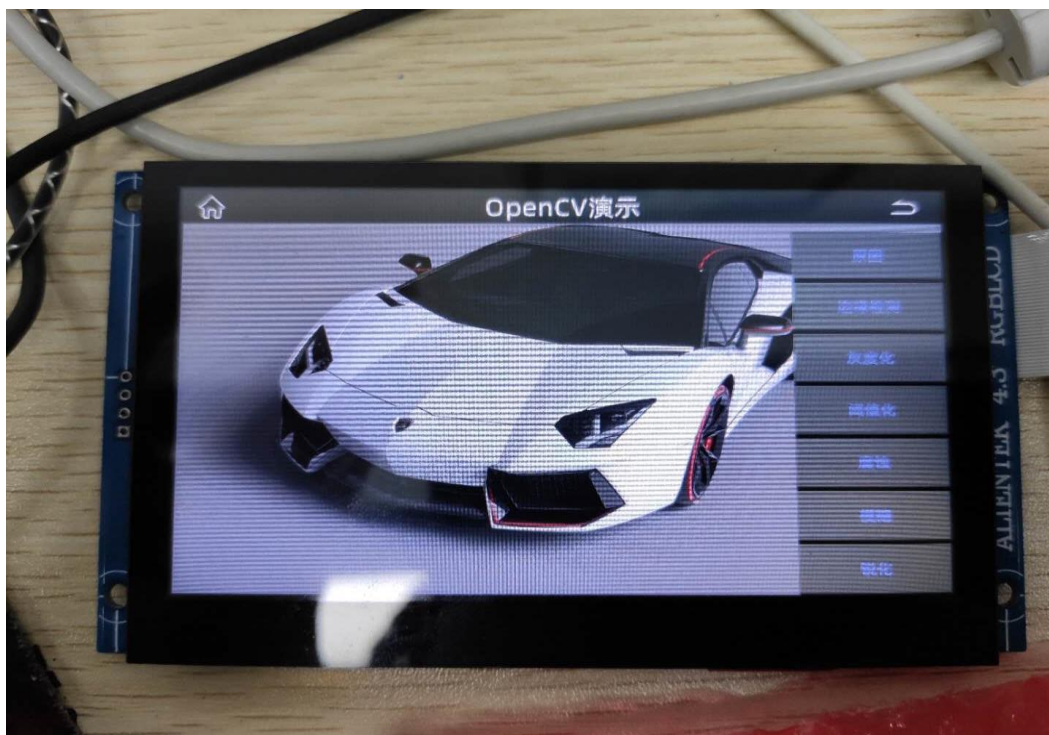


图 5.2.5 QOpenCV 示例上显示的原图

点击边缘检测，可以看到这辆汽车已经被线条勾勒了。此外还有灰度化、阈值化、腐蚀、模糊、锐化等。这里就不贴图了，图片处理速度非常快。效果还是不错的

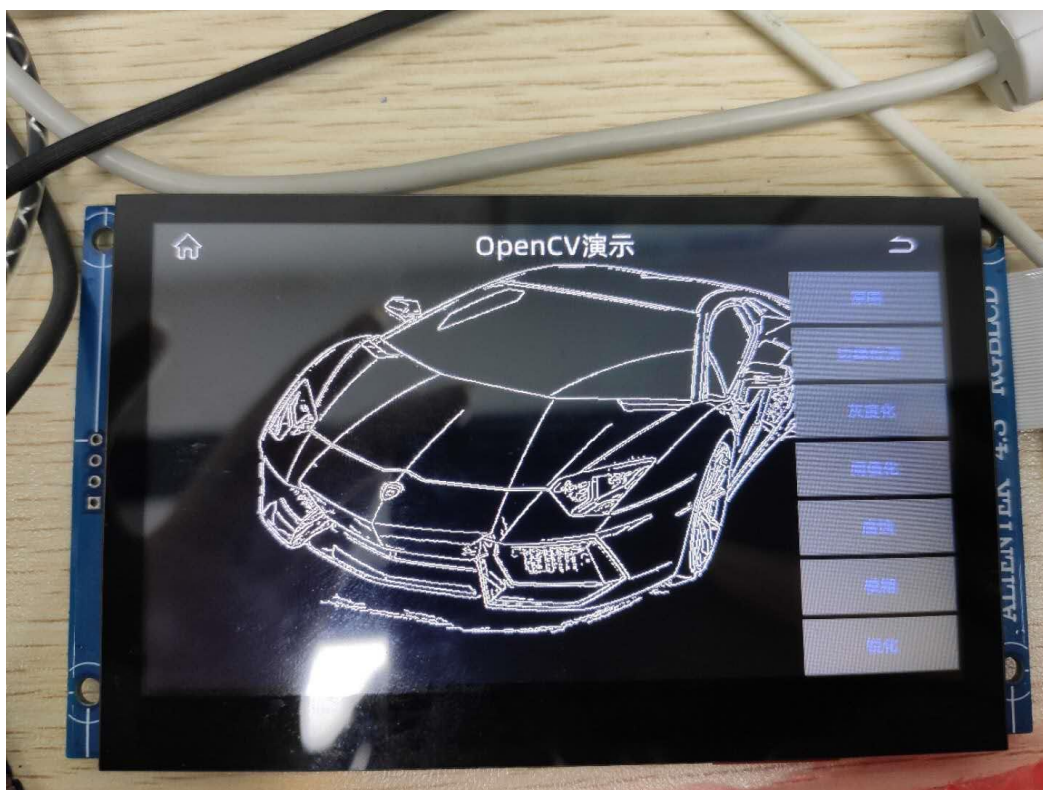


图 5.2.6 点击边缘检测，图像处理的效果如上图

至此移植及应用示例已经全部结束。若文档中有错漏或者需要补充的请到正点原子 imx6 版块发贴讨论。