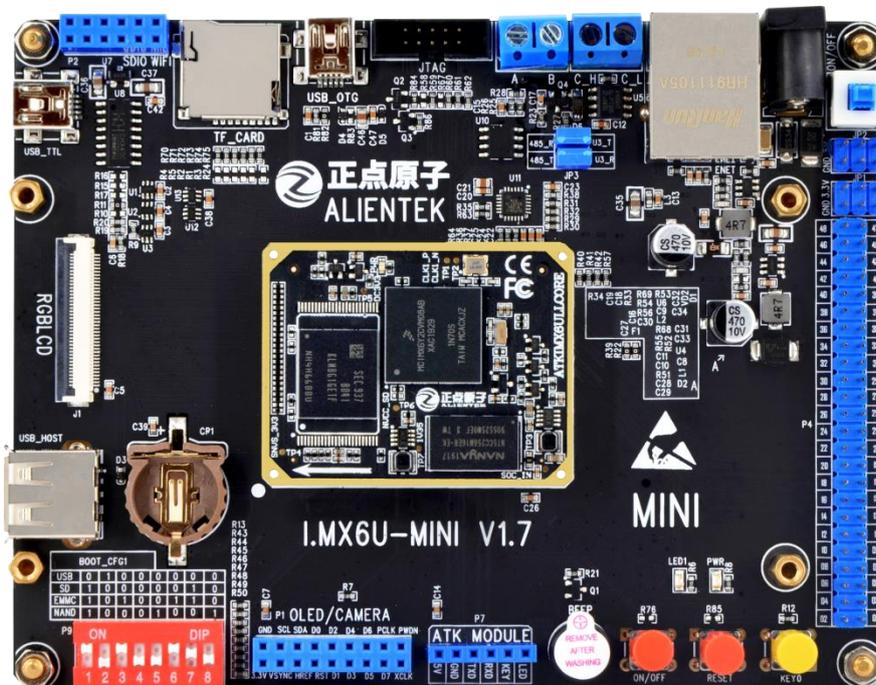
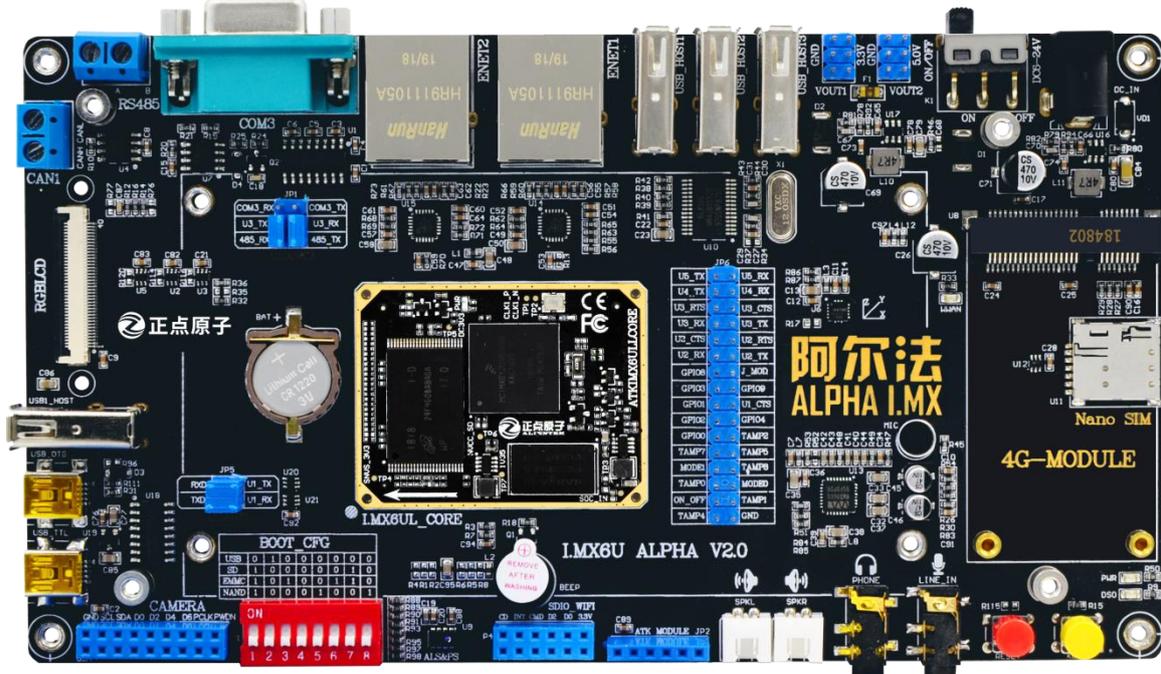


【正点原子】I.MX6U

出厂系统 OpenCV 使用说明 V1.1





正点原子公司名称 : 广州市星翼电子科技有限公司

原子哥在线教学平台 : www.yuanzige.com

开源电子网 / 论坛 : <http://www.openedv.com/forum.php>

正点原子淘宝店铺 : <https://openedv.taobao.com>

正点原子官方网站 : www.alientek.com

正点原子 B 站视频 : <https://space.bilibili.com/394620890>

电话: 020-38271790 传真: 020-36773971

请关注正点原子公众号, 资料发布更新我们会通知。

请下载原子哥 APP, 数千讲视频免费学习, 更快更流畅。



扫码关注正点原子公众号



扫码下载“原子哥”APP

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	正点原子 Linux 团队	2021.03.09
V1.1	1.修改文档格式及第二页联系方式。	正点原子 Linux 团队	正点原子 Linux 团队	2021.06.19

目录

前言.....	5
第一章 交叉编译环境搭建.....	6
1.1 安装出厂系统所使用的交叉编译器	7
第二章 编译 OPENCV 程序.....	9
2.1 编译简单的 OPENCV 程序	10
2.2 如何在 QT 项目中加入 OPENCV	12
附录-A	12

前言

正点原子 I.MX6U 出厂系统自带的 OpenCV 版本是 3.1 版本,如果您使用的系统没有带 OpenCV,请到正点原子论坛 www.openedv.com,找到正点原子资料下载总链接,然后找到 Linux,在 I.MX6U 资料网盘上下载最新的出厂系统即可!

如果您觉得 OpenCV 版本低,不适用,可以参考光盘资料里的【正点原子】I.MX6U 移植 OpenCV V1. x. pdf 文档自行进行移植高版本的 OpenCV 即可!但是相对于学习和基本的使用 OpenCV 3.1 版本已经足够了。

本档需要用户有 RGB LCD 屏显示的,因为 OpenCV 是做图像处理的,如果没有屏的用户请不要进行操作,因为下面的例子都是要显示在屏上面的。

注意:本档并不介绍如何学习 OpenCV,不属于我们讨论范围内。同时本档是适用于正点原子的出厂文件系统,不适用于教程所定制的 buildroot 或者 busybox 文件系统,因为上面两个文件系统是不含 OpenCV 与 Qt 的。

本档所使用的环境:

-  Ubuntu 16.04 64 bits。
-  要求读者会使用 FileZilla、WinSCP 及 Windows Git 进行 Ubuntu 与 Windows 间互传文件的方法。
-  要求读者会开发板与 ubuntu 传输文件,请参考【正点原子】I.MX6U 开发板文件拷贝及固件更新参考手册 V1. x. pdf。

第一章 交叉编译环境搭建

正点原子 I.MX6U 出厂系统的由 Yocto 项目编译得到，所以交叉编译工具需要使用出厂配套的交叉编译工具。

1.1 安装出厂系统所使用的交叉编译器

如果已经在【正点原子】I.MX6U 用户快速体验 V1.x.x.pdf 第四章 4.2 小节安装出厂系统所用的 Poky 交叉编译，那么就可以直接到下一章节直接开始使用出厂系统的 OpenCV 吧。如果还没有安装出厂系统所使用的交叉编译器请按如下步骤进行安装。

把开发板光盘 A-基础资料->5、开发工具->1、交叉编译器->fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh 拷贝到 Ubuntu 虚拟机
如下图本文已经把交叉编译工具拷贝到了 Ubuntu 虚拟机。

```
allentek@ubuntu:~$ ls
fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
allentek@ubuntu:~$
```

图 1.1 1 拷贝 sdk 工具包到 ubuntu 系统里

执行下面的指令修改脚本的权限，修改权限后可以看到此脚本颜色显示改变，说明修改成功。

```
chmod u+x fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
```

直接执行脚本安装交叉编译工具，连续敲下两次回车键确认，再输入用户密码即可。本次安装的目录为脚本所指定的默认安装的目录，后面的内核编译环境的交叉编译都是按这个安装目录去操作，所以建议用户也是默认安装到/opt/fsl-imx-x11/4.1.15-2.1.0 这个默认目录。

```
allentek@ubuntu:~$ ./fsl-imx-x11-glibc-x86_64-meta-toolchain-qt5-cortexa7hf-neon-toolchain-4.1.15-2.1.0.sh
Freescale i.MX Release Distro SDK installer version 4.1.15-2.1.0
=====
Enter target directory for SDK (default: /opt/fsl-imx-x11/4.1.15-2.1.0):
You are about to install the SDK to "/opt/fsl-imx-x11/4.1.15-2.1.0". Proceed[Y/n]?
[sudo] password for allentek:
Extracting SDK.....done
Setting it up...done
SDK has been successfully set up and is ready to be used.
Each time you wish to use the SDK in a new shell session, you need to source the environment setup script e.g.
$ . /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allentek@ubuntu:~$
```

图 1.1 2 安装 sdk 工具到默认的目录

使用方法也十分简单，根据上面打印出来的提示，直接使能环境变量就可以了。但是在不同终端或者切换用户时需要重新使能环境变量方可使用。

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

```
allentek@ubuntu:~$ source /opt/fsl-imx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allentek@ubuntu:~$
```

图 1.1 3 使能环境变量

使能环境变量后可以使用 env 指令查看生效的环境变量，下图为部分截图，可以看出使能了这个环境变量后 gcc 已经配置好编译时所用的参数，如硬浮点参数-mfpu=neon -mfloat-abi=hard。使用硬浮点交叉编译，可以使用 CPU 自带 FPU。下图为环境变量部分截图。

env

```
LESSOPEN=| /usr/bin/lesspipe %s
ARCH=arm
RANLIB=arm-poky-linux-gnueabi-ranlib
OE_QMAKE_CFLAGS=
INSTANCE=Unity
UPSTART_JOB=unity-settings-daemon
CROSS_COMPILE=arm-poky-linux-gnueabi-
CC=arm-poky-linux-gnueabi-gcc -march=armv7ve -mfpu=neon -mfloat-abi=hard -mcpu=cortex-a7 --sysroot=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi
XDC_RUNTIME_DIR=/run/user/1000
DISPLAY=:0
XDC_CURRENT_DESKTOP=Unity
GTK_MODULES=ibus
OE_QMAKE_LIBDIR_QT=/opt/fsl-imx-x11/4.1.15-2.1.0/sysroots/cortexa7hf-neon-poky-linux-gnueabi/usr/lib
OBJDUMP=arm-poky-linux-gnueabi-objdump
```

图 1.1 4 查看使能后的环境变量

使用 `arm-poky-linux-gnueabi-gcc -v` 指令可以查看 gcc 版本，表明环境变量已经生效。

```
arm-poky-linux-gnueabi-gcc --version
```

```
allentek@ubuntu:~$ arm-poky-linux-gnueabi-gcc --version
arm-poky-linux-gnueabi-gcc (GCC) 5.3.0
Copyright (C) 2015 Free Software Foundation, Inc.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.
allentek@ubuntu:~$
```

图 1.1 5 查看 gcc 版本信息

第二章 编译 OpenCV 程序

下面开始介绍如何使用正点原子出厂系统的 OpenCV。介绍的内容有如下：

- (1) 如何编写 OpenCV 应用程序，怎么编译？
- (2) 在 Qt 里如何加入 OpenCV？

2.1 编译简单的 OpenCV 程序

下面开始编写一个简单的 OpenCV 例子，然后使用第一章里安装的交叉编译器进行编译这个简单的例子，拷贝编译出来的可执行文件到开发板上执行即可！

OpenCV 是使用 C++ 编程语言，可以说 C++ 语言是 C 语言的超集。所以我们编程需要使用 C++ 语言。C++ 语法这里我们不分析，这些都是很基本的，现在我们开始写一个简单的 OpenCV 程序吧。

编辑一个 `opencv_test.cpp` 文件。注意 C++ 程序一般后缀是使用 `cpp`。

```
vi opencv_test.cpp
```

在终端下插入下面的代码。

```
#include <opencv2/opencv.hpp>
#include <iostream>

using namespace std;
using namespace cv;

int main(int argc, char **argv)
{
    Mat img = imread("/opt/src/images/美女.jpg");
    if (img.empty()) {
        cout << "打开图像失败！" << endl;
        return -1;
    }
    namedWindow("image", CV_WINDOW_AUTOSIZE);
    imshow("image", img);
    waitKey();

    return 0;
}
```

使能环境变量，执行下面的指令会使能编译器，目的是为了告诉系统我们的编译器的位置在哪里。

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

使用下面的指令开始编译。

```
$CXX opencv_test.cpp -o opencv_test -lopencv_core -lopencv_highgui -lopencv_imgproc -lopencv_videoio -lopencv_imgcodecs
```

指令解释：

- `$CXX`：指的是取终端里 `CXX` 的值，出厂系统提供的交叉编译工具链编译 C++ 就使用 `$CXX`，如果编译 C，请使用 `$CC`。
- `opencv_test.cpp`：程序源文件。
- `-o`：指定编译输出的文件名。
- `opencv_test`：编译后的目标文件名称。

- -l: 这个是-l, 不是一, 是 L 的小写 l。指定链接库。上面指定了许多链接库。但是实际上可能不用那么多。请根据自己的项目自行添加或者裁减! 写多了不会错, 写少了到编译时可能会报错! 具体需要看自己用到的 API 在哪个库里。

编译完成后, 使用 ls 指令查看编译完成的目标文件 opencv_test。

```
ls opencv_test
```

```
allentek@ubuntu:~$ vi opencv_test.cpp
allentek@ubuntu:~$ source /opt/fsl-inx-x11/4.1.15-2.1.0/environment-setup-cortexa7hf-neon-poky-linux-gnueabi
allentek@ubuntu:~$ $CXX opencv_test.cpp -o opencv_test -lopencv_core -lopencv_highgui -lopencv_imgproc -lopencv_videoio -lopencv_imgcodecs
allentek@ubuntu:~$ ls opencv_test
opencv_test
allentek@ubuntu:~$
```

图 2.1 1 编译 opencv 应用程序

拷贝到开发板上执行。注意需要看出厂系统的文件系统版本。正点原子只有 2.0 版本以上的出厂文件系统才支持 OpenCV 程序显示。

查看出厂文件系统版本, 使用下面的指令查看。看到是 v2.0 版本以上才支持 opencv 程序直接显示。如果不是, 请到正点原子论坛找资料下载, 更新到最新的出厂系统即可。

```
cat /etc/version
```

```
root@ATK-IMX6U:~# cat /etc/version
Author: 正点原子@qq1252699831

Version v1.0 - Version v1.9
Note:
请参阅正点原子最新快速体验手册第1.2.2小节

Version v2.0 (Latest stable version)
Date: 2021.02.23
Note:
1. 移除mysql
2. 其他优化

root@ATK-IMX6U:~#
```

图 2.1 2 查看文件系统版本

拷贝 opencv_test 到开发板文件系统目录, 如下。

```
root@ATK-IMX6U:~# ls opencv_test
opencv_test
root@ATK-IMX6U:~#
```

图 2.1 3 拷贝 opencv_test 到开发板

在执行 opencv_test 之前, 首先我们需要使用 xinit 指令初始化。Xinit 是作用是对 X-Window 系统的初始化程序, 主要完成 X 服务器的初始化设置。在高版本的文件系统使用 wayland 的方式。使用 Xinit 初始化, 包括服务端, 可以接收鼠标键盘的输入等。简单的来说要想直接使用 OpenCV 显示在屏幕上, 是依赖 X 服务的, 需要开启桌面窗口, 然后 OpenCV 程序在桌面窗口里运行。

出厂系统默认使用的 Qt 是不依赖 X 服务器。默认出厂系统已经把 X 服务器相关关掉, 所以我们需要执行下面的执行进行开启。

执行下面指令时请先退出出厂系统默认的 Qt 界面。在 Qt 界面“设置”里点击退出即可。

```
xinit /etc/X11/Xsession >/var/log/Xsession.log 2>&1 &
```

屏幕立刻变黑, 看不到任何东西。实际上 X 服务已经运行了。

执行下面的指令运行我们编译出来的 OpenCV 程序。

```
root@ATK-IMX6U:~# xinit /etc/X11/Xsession >/var/log/Xsession.log 2>&1 &
[2] 1380
root@ATK-IMX6U:~# ./opencv_test
```

图 2.1 4 开启 X 服务

上面的 `opencv_test` 程序会读取我们文件系统下的 `/opt/src/images/美女.jpg` 图片, 并显示在屏幕上, 如下图。



图 2.1 5 程序运行在 LCD 屏上的现象

如果用户有摄像头, 编写相关的 `opencv` 应用程序一样可以按上面这种方法, 直接显示于屏幕上。

2.2 如何在 Qt 项目中加入 OpenCV

与上一小节编译 `OpenCV` 程序相似, 我们只需要在 `Qt` 的 `pro` 项目添加相关的 `opencv` 相关设置。在项目文件 `*.pro` 添加以下语句即可!

```
unix{
    CONFIG += link_pkgconfig
    PKGCONFIG += opencv
}
```

然后参考【正点原子】I.MX6U 出厂系统 Qt 交叉编译环境搭建 V1.x.pdf, `x >= 5` 的文档配置好 `Qt Creator` 的交叉编译环境, 添加 `opencv` 相关头文件, 然后编写项目, 编译项目即可!

附录-A