**一、交叉编译器**

gcc-linaro-4.9-2016.02-x86\_64\_arm-linux-gnueabihf.tar.xz

1、配置环境变量

sudo gedit ～/.bashrc

export PATH=$PATH:/home/s106/arm/arm-linux-gnueabihf/bin

export CROSS\_COMPILE=arm-linux-gnueabihf-

2、使环境变量生效

source ~/.bashrc

3、运行：arm-linux-gnueabihf-gcc -v出来版本信息，即构建成功

**二、camke2.8.12.2,使用低版本**

1、./configure

2、make

3、make install

**三、Opencv+ffmpeg的交叉移植**

1、使用opencv的Videoio库函数需要ffmpeg的支持

2、OpenCV交叉编译时各个库的依赖关系:

OpenCV |--------zlib

|--------jpeg

|--------libpng

|--------zlib

|--------tiff

|--------zlib

|--------ffmpeg

|--------x264

|--------xvidcore

3、依赖库的配置以及注意事项如下:

参考<https://www.eefocus.com/beiyangguangdian/blog/14-03/302119_52e49.html> 3.1、首先建立一个路径：

export ZYNQ\_CV\_INSTALL=/opt/zedboard/opencv\_zed/opencv-lib #OpenCV第三方库交叉编译库的安装路径

3.2、zlib-1.2.7

export CC=arm-xilinx-linux-gnueabi-gcc

./configure --prefix=$ZYNQ\_CV\_INSTALL --shared

make

make install

3.3、x264

./configure --host=arm-linux --cross-prefix=arm-xilinx-linux-gnueabi- --enable-shared --disable-asm --prefix=$ZYNQ\_CV\_INSTALL

make

make install

3.4、xvidcore

cd build/generic

./configure --prefix=$ZYNQ\_CV\_INSTALL --host=arm-xilinx-linux-gnueabi --disable-assembly

make

make install

注意:xvidcore的编译中可能会调用arm指令集,但是在ZEDBoard上的ARM v7指令集的CortexA9核心只能支持Thumb/Thumb2,因此这里需要禁用汇编选项--disable-assembly

3.5、ffmpeg

./configure --prefix=$ZYNQ\_CV\_INSTALL --enable-shared --disable-static --enable-gpl --enable-cross-compile --arch=arm --disable-stripping --target-os=linux --enable-libx264 --enable-libxvid --cc=arm-xilinx-linux-gnueabi-gcc --enable-swscale --extra-cflags=-I$ZYNQ\_CV\_INSTALL/include --extra-ldflags=-L$ZYNQ\_CV\_INSTALL/lib --disable-asm

make

make install

注意:ffmpeg编译过程中可能会用到x264,xvid,zlib等之类的库,因此我们需要添加额外的LDFLAGS和CFLAGS变量,更多的信息可以通过./configure --help进行参考,另外这里的--disable-asm与xvidcore中的--disable-assembly的作用是一样的,否则也是不能编译通过的

3.6、opencv3.4.4

3.6.1、mkdir build

3.6.2、cd build，建立toolchain.cmake

3.6.3、cmake -DBUILD\_EXAMPLES=1 -DBUILD\_ZLIB=1 -DBUILD\_JPEG=1 -DCMAKE\_TOOLCHAIN\_FILE=./xilinx.cmake -DWITH\_FFMPEG=1 -DWITH\_PNG=0 -DWITH\_1394=0 -DWITH\_GTK=0 -DWITH\_GTK\_2\_X=0 -DWITH\_GSTREAMER=0 -DWITH\_GSTREAMER\_0\_10=0 ..

如果cmake找不到ffmpeg的错误：

FFMPEG: NO

avcodec: NO

avformat: NO

avutil: NO

swscale: NO

avresample: NO

执行export PKG\_CONFIG\_LIBDIR=/opt/zedboard/opencv\_zed/opencv-lib/lib/pkgconfig（测试有效） 参考<https://blog.csdn.net/Webster_WXH/article/details/80965475>

将pkgconfig文件夹放入/usr/include/x86\_64-linux-gnu目录下。（注意看一下路径）

路径参考[https://blog.csdn.net/woainannanta/article/details/78260419。](https://blog.csdn.net/woainannanta/article/details/78260419%E3%80%82)

重启，使环境变量生效

3.6.4、make

如果过程中出现错误 ../../lib/libopencv\_videoio.so.3.4.0: undefined reference to av\_init\_packet@LIBAVCODEC\_58' ../../lib/libopencv\_videoio.so.3.4.0: undefined reference to avformat\_get\_riff\_video\_tags@LIBAVFORMAT\_58' ../../lib/libopencv\_videoio.so.3.4.0: undefined reference to avformat\_get\_mov\_video\_tags@LIBAVFORMAT\_58' ../../lib/libopencv\_videoio.so.3.4.0: undefined reference to avcodec\_find\_decoder@LIBAVCODEC\_58'

应该是搜索路径还有问题，暂时先把ffmpeg交叉编译生成文件复制到交叉编译器的usr路径下， 即/home/s106/arm//home/s106/arm/arm-linux-gnueabihf/arm-linux-gnueabihf/libc/usr/lib， 就不会报错了。（有效） 参考<https://blog.csdn.net/sinat_39391990/article/details/87957379>

3.6.5、make install

**四、将生成的opencv库移植到开发板**

1、将生成的/include、/lib复制到板载linux的系统路径，如/usr/local/include、/usr/local/lib和/usr/include、/usr/lib等 2、将生成的OpenCV第三方库交叉编译库，路径ZYNQ\_CV\_INSTALL=/opt/zedboard/opencv\_zed/opencv-lib 复制到板载linux系统的/usr/lib。（有效） 因为板载编译器可能会出现找不到依赖库的错误。

**五、添加环境变量**

sudo vim opencv.config

i

/usr/local/lib

Esc

:wq

sudo ldconfig