**工作交接内容**

* Camera
* Recovery
* Windows驱动

**Camera**

这里没有什么好说的，都是高通平台的标准流程。

详细的调试内容包括驱动和效果；

驱动主要分两部分：

一：devicestree部分，分camera0和camera1也就是后摄和前摄，在驱动部分的配置要参考这里的配置。

以我们8916平台为例，刚说的dts文件指的是这个：

\kernel\arch\arm\boot\dts\qcom\msm8916-camera-sensor-qrd-skui.dtsi

二：驱动部分主要在hal层的代码添加，具体配置需要根据datasheet来，重点在I2C地址、上电以及mipi的配置。

以ov5670为例，这里指的配置文件是：

\vendor\qcom\proprietary\mm-camera\mm-camera2\media-controller\modules\sensors\sensor\_libs\ov5670\_q5v41b\ov5670\_q5v41b\_lib.c

效果部分：

目前我司没有效果调试的环境，目前主要的内容是由FAE提供参考效果代码，我们负责合入；如果客户有要求，需要具体调试的时候，这边是提case，由高通协助我们来调试，调试周期不确定，高通那边一个摄像头要完整调试效果的话，从头到尾一般需要1个月的时间来完成；主要时间的把控还在我们这边，看客户要求和项目进度。

效果的文件生成和优化，高通平台有专门的工具，叫” Chromatix”,这个工具可以去高通网站上下载，相关使用也有专门的介绍文档，我们也有这个文档，叫《80-NK872-2\_C\_Chromatix\_6\_Camera\_Tuning.pdf》

文档在服务器上：\\192.168.0.85\新产品部\软件部资料\技术分享\驱动领域\camera\msm8916\_camera\Multimedia Qcamera

关于camera部分的高通文档都在这个目录下。

效果、马达配置文件所在目录：

\vendor\qcom\proprietary\mm-camera\mm-camera2\media-controller\modules\sensors

其中actuator\_libs和actuators中存放的是马达配置

Chromatix目录存放的是效果代码

其余还有eeprom等。

添加一个camera主要的修改是下面几个地方

Make file

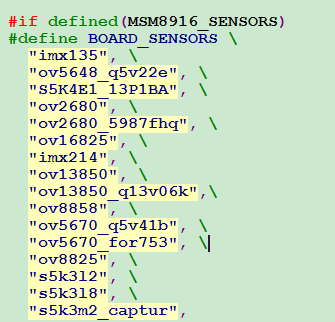
\vendor\qcom\proprietary\common\config\device-vendor.mk

效果、马达、camera的lib.c都在这个make file中添加

Camera ID添加

\vendor\qcom\proprietary\mm-camera\mm-camera2\media-controller\modules\sensors\module\sensor\_init.c

如下图



这里是8939的形式，8916略微有点不同，不过一看便知，也是同一个目录，同一个文件。

一般情况下FAE都会提供一套camera、马达和效果的参考代码，这些可以直接合入。我们这边开始需要做的是保证camera probe成功，如果失败可以从几个方向去排查：

1.上电是否正确，PWDM、RESET、CLK、I2C硬件上是否ok

2.I2C地址是否正确

3.MIPI是否配置正确，比如2line、4line或1line，是否跟datasheet匹配

4.前后摄配置是否跟dts里匹配

以上4点，除第一点外，其余的修改都在对应camera的lib.c中进行，或者排查。

**Recovery**

Recovery升级包打包环境需要放在ubuntu服务器运行，打包工具和环境在make\_update\_package文件夹中，将整个文件放在服务器即可。

另外：该升级包打包完成后是一个.zip文件，可以选择adb导入方式和制作成cache镜像烧入设备中两种方式来导入到设备。

将.zip制作成cache镜像，也有工具，在”\工作交接\recovery相关\打包yaffs2和解压yaffs2工具”,其中有使用说明《使用方法.txt》。

一般使用的触发recovery升级也有两种方法：

一是使用command文件来实现，二是写misc分区来实现。

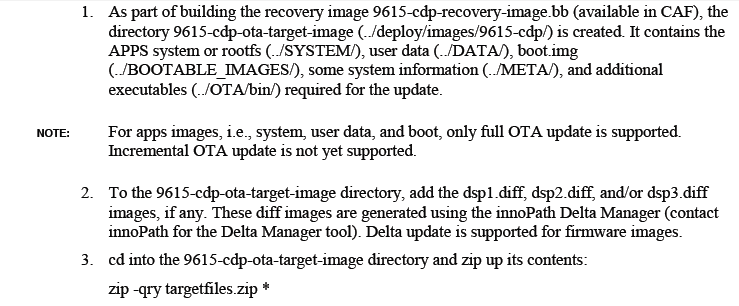
这两种办法我都尝试过，对应需要的command文件和misc镜像可以参考” \工作交接\recovery相关\recovery修改”目录下的”command”和” misc.yaffs”；都可以用UE编辑器打开查看的。

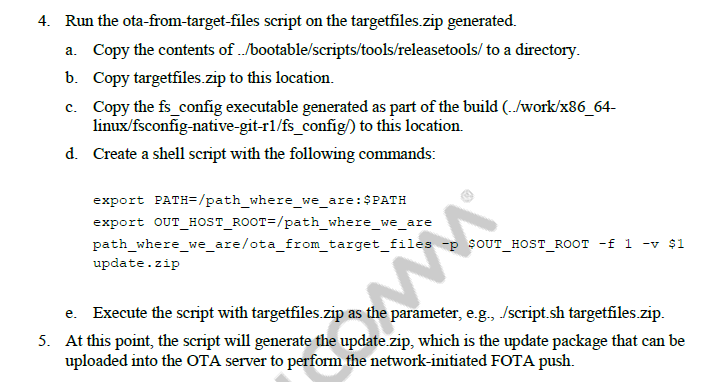
Command文件需要放在设备的固定路径：/cache/recovery/

Misc.yaffs可以用fastboot命令来烧入,例如：fastboot flash misc d:/misc.yaffs

附上一个介绍，参看如下说明：

**9x15 制作升级包**





准备好上文提到到的directory（比如ota\_diff\_package），该文件中存放了打包的targetfiles.zip；和copy过来的一些文件。

在directory同级目录放着创建的script.sh

文件内容是

export PATH=./ota\_diff\_package:$PATH

export OUT\_HOST\_ROOT=./ota\_diff\_package

./ota\_diff\_package/ota\_from\_target\_files -p $OUT\_HOST\_ROOT -f 1 -v $1 update.zip

然后cd到该目录

执行 ./script.sh ./ota\_diff\_package/targetfiles.zip

之后会在该目录生成最终要使用的update.zip

该文件就是用来升级的升级包。

**Windows驱动**

Windows驱动不是一两句可以说清楚的，具体点的工作内容，这里主要包括两部分

一：动的编译

编译部分有一个文档，参见《Windows驱动编译.docx》，介绍了编译环境的准备。

二：驱动打包

高通提供的驱动是非exe的，我们需要将他打包成exe安装文件：

打包所用的工具是innosetup，需要安装该工具以提供打包脚本运行所需要的环境，innosetup有文档说明，详见《Inno+Setup中文帮助.pdf》。

目前已有的环境在windows-forge文件中

打包需运行USBDriver.iss脚本来进行，具体可以打开该脚本文件进行参考。

驱动logo是该文件中的logo.ico文件。

USBDriver.iss脚本运行成功后会在\windows-forge\cdrom目录下生成setup.exe文件

驱动散包需要放在USBDriver.iss脚本中写好的地方，目前的脚本要求

adb驱动放在：\windows-forge\extern\9615\_adb\_driver

windows驱动放在：\windows-forge\extern\usb\_driver