**目 录**

1. **AP篇**

**1.如何设置最大充电电流**

**2.如何修改camera预览、视频、图片的尺寸**

**3.GPIO键盘的修改**

**4.如何实现OTA功能**

**5.selinux作用域的修改**

1. **FOTA篇**

**1.KAIOS系统FOTA制作及修改**

**2.android系统的FOTA制作及修改**

**3.FOTA升级动画的制作及修改**

1. **MODEM篇**

**1.MODEM相关的RFC修改**

**2.实现工厂复写IMEI/ESN/BT的修改**

**3.QCN的制作及修改**

**4.golden efs的制作及修改**

**5.MBN的制作及修改**

**6.PLMN LOCK的设置**

**7.PLMN LOCK的清除与熔断机制**

1. **心得总结**

# AP篇

1. **如何设置最大(小)充电电流**

File Path:  [LINUX/android/kernel/drivers/power/qpnp-linear-charger.c](http://124.204.36.153:8081/" \l "/c/10866/1/LINUX/android/kernel/drivers/power/qpnp-linear-charger.c)

#define QPNP\_LBC\_IBATMAX\_MIN»       90 （此处定义最小充电电流）

#define QPNP\_LBC\_IBATMAX\_MAX»       720 （此处定义最大充电电流）

**2.如何修改camera预览、视频、图片的尺寸**

File path:

LINUX/android/vendor/qcom/proprietary/mm-camera/mm-camera2/media-controller/mct/pipeline/mct\_pipeline.c

static cam\_dimension\_t default\_preview\_sizes[] = {

static cam\_dimension\_t default\_picture\_sizes[] = {

static cam\_dimension\_t default\_liveshot\_sizes[] = {

static cam\_dimension\_t default\_video\_sizes[] = {

以上数组定义支持相关比例的图片，修改需要支持的照片resolution即可.

**3.GPIO键盘的修改**

File path:

[LINUX/android/device/qcom/msm8909/matrix\_keypad.kl](http://124.204.36.153:8081/" \l "/c/11234/1/LINUX/android/device/qcom/msm8909/matrix_keypad.kl)

[LINUX/android/kernel/arch/arm/boot/dts/qcom/F300B\_MSM8909\_512-skub.dtsi](http://124.204.36.153:8081/" \l "/c/11234/1/LINUX/android/kernel/arch/arm/boot/dts/qcom/F300B_MSM8909_512-skub.dtsi)

[LINUX/android/kernel/arch/arm/configs/F300B\_MSM8909\_512-perf\_defconfig](http://124.204.36.153:8081/" \l "/c/11234/1/LINUX/android/kernel/arch/arm/configs/F300B_MSM8909_512-perf_defconfig)

[LINUX/android/kernel/drivers/input/keyboard/aw9523.c](http://124.204.36.153:8081/" \l "/c/11234/1/LINUX/android/kernel/drivers/input/keyboard/aw9523.c)

1. 在matrix\_keypad.kl文件中定义了键盘所有按键映射表，需要自己根据实际的键盘情况去增加或者修改，在adb shell下用执行指令getevent，然后按每个按键，会显示出相应的键值。
2. 在F300B\_MSM8909\_512-skub.dtsi中定义了阵列式的键盘映射表，需要修改以下地方

&soc{

matrix\_keypad: matrix\_keypad@0{

row-gpios=<…> // 此为键盘所使用的行GPIO，需要修改

col-gpios=<…> // 此为键盘所使用的列GPIO，需要修改

linux,keymap=<…> //此为行列组合成的阵列对应的键值映射表，需要修改，数值为十六进制数，前两位表示列号，3、4位表示行号，后面4位表示键值

}

}

&tlmm\_pinmux{

tlmm\_gpio\_key{

qcom,pins = <&gp 52>, <&gp 56>, <&gp 7>,<&gp 99>,<&gp 6>; // 将行、列的GPIO都需要分开配置成这样

}

（3）在文件F300B\_MSM8909\_512-perf\_defconfig中，需要将GPIO键盘的驱动配置打开

CONFIG\_KEYBOARD\_MATRIX=y

（4）在文件aw9523.c中，此处修改根据特殊GPIO设定来判断区分是否为I2C键盘或者GPIO键盘,如果只存在一种键盘驱动，此处可以省略。

**4.如何实现OTA功能**

File path:

LINUX/android/device/qcom/msm8909\_512/system.prop

LINUX/android/device/qcom/common/rootdir/etc/init.qcom.sh

(1)在system.prop文件中，请检查是否有以下属性，若没有，需要添加以下修改

persist.radio.sw\_mbn\_update=0

persist.radio.start\_ota\_daemon=1

(2)在init.qcom.sh文件中，需要进行以下修改

将cp -r /firmware/image/modem\_pr/mbn\_ota/\* /data/misc/radio/modem\_config修改为

cp -r /firmware/image/modem\_pr/mcfg/configs/mcfg\_sw/generic/mbn\_ota/\* /data/misc/radio/modem\_config

在进行以上修改后，需要将NV71546设置成0X07

**5.selinux作用域的修改**

**File path:**

[LINUX/android/device/qcom/sepolicy/common/read\_sn.te](http://192.168.1.148:8080/" \l "/c/232/1/LINUX/android/device/qcom/sepolicy/common/read_sn.te)

LINUX/android/device/qcom/common/rootdir/etc/**init.qcom.rc**

LINUX/android/device/qcom/msm8909go/QC40A\_MSM8909go/**QC40A\_MSM8909go\_features.mk**

LINUX/android/device/qcom/sepolicy/common/**file\_contexts**

LINUX/android/device/qcom/sepolicy/common/**service.te**

LINUX/android/device/qcom/sepolicy/common/**service\_contexts**

LINUX/android/vendor/bluebank/read\_sn/**Android.mk**

LINUX/android/vendor/bluebank/read\_sn/**read\_sn.c**

read\_sn为一个新加的功能指令(此处源码并没有列出)，当执行这个功能时，发现很多权限没有，导致执行失败，如read,write,ioctl,open等权限，此时需要修改相应的权限,各个权限的修改可抓logcat,然后查看并依次打开需要的权限

read\_sn.te文件为新加的selinux作用域文件，目的是为了增加相关功能的权限,如下:

type read\_sn, domain; //此处必须定义一个作用域

type read\_sn\_exec, exec\_type,vendor\_file\_type,file\_type; //此处定义文件类型

init\_daemon\_domain(read\_sn) //初始化作用域

qmux\_socket(read\_sn); //可省略

//以下几条allow语句为允许相应的权限

allow read\_sn read\_sn\_exec:file {getattr execute};

allow read\_sn read\_sn\_exec:file entrypoint;

allow read\_sn self:socket {create bind read write ioctl setopt};

allow read\_sn self:capability dac\_override;

allow read\_sn diag\_device:chr\_file {read write open ioctl};

allow read\_sn vendor\_shell\_exec:file execute\_no\_trans;

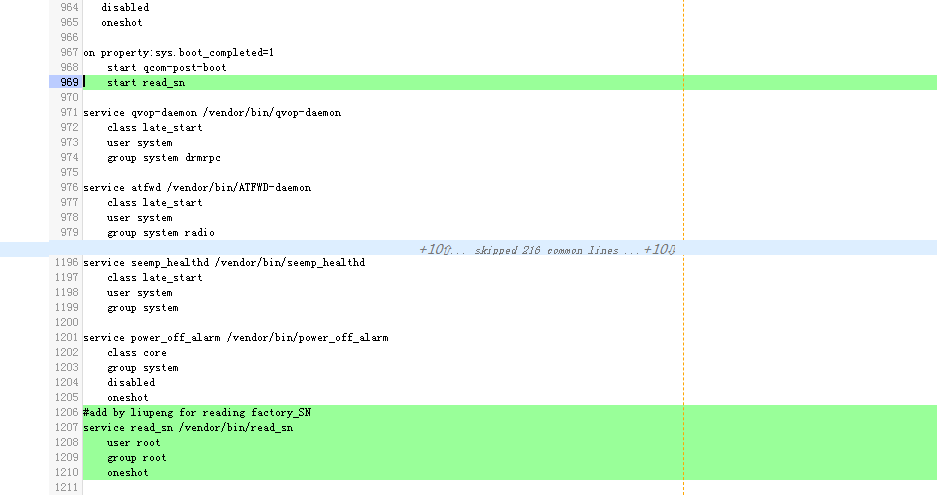
allow read\_sn vendor\_toolbox\_exec:file execute\_no\_trans;

allow read\_sn property\_socket:sock\_file {write};

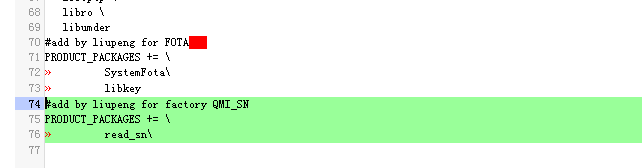
allow read\_sn init:unix\_stream\_socket connectto;

allow read\_sn radio\_prop:property\_service {set};

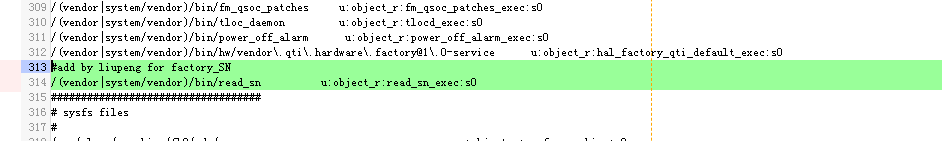
在**init.qcom.rc中增加开机自动启动读read\_sn的服务**



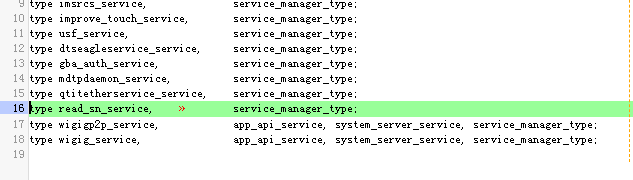
在**QC40A\_MSM8909go\_features.mk中将read\_sn指令编译进代码**



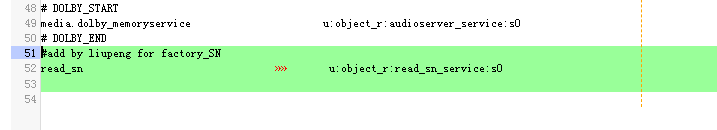
在**file\_contexts中增加read\_sn作用域上下文**



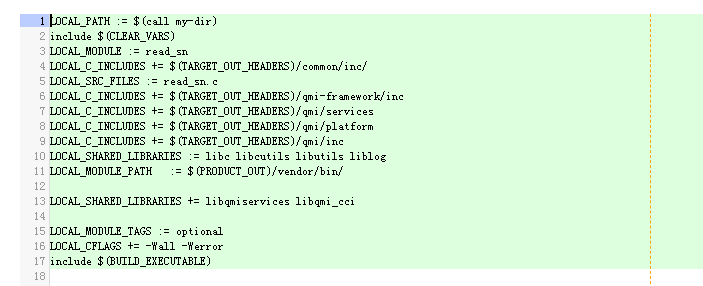
在**service.te中为read\_sn服务进行定义**



**在service\_contexts中为read\_sn定义服务的上下文**

****

**在Android.mk中，此文件为read\_sn增加的自定义编译文件**

****

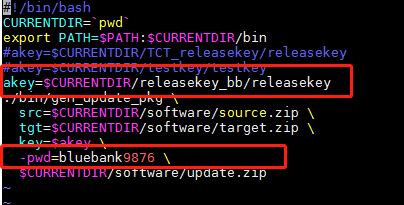
**在read\_sn.c中，此文件为read\_sn的源码文件，此处可查源代码**

# FOTA篇

1. **KAIOS系统FOTA制作及修改**

**FOTA环境搭建:**

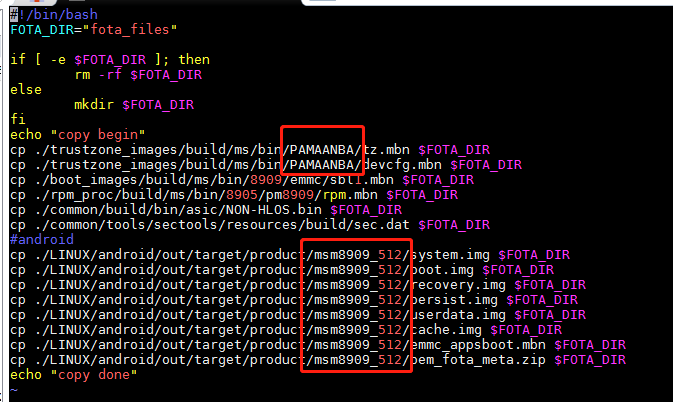
1. **获取KAIOS的FOTA工具:如FOTA\_1.5.6**
2. **由于项目采用block的方式，所以”cd FOTA\_1.5.6/fota\_tools/block”,执行./Generat\_tools.sh,可在上一层目录生成KaiOS-QC-Block-V1.5.6目录**
3. **切换到KaiOS-QC-Block-V1.5.6目录，在文件gen\_update\_pkg\_start.sh中修改项目的release\_key和pwd,(releasekey\_bb的文件夹需要手动添加到当前目录)**

****

**4.执行make,再执行make links,FOTA环境才算搭建成功**

**FOTA流程:**

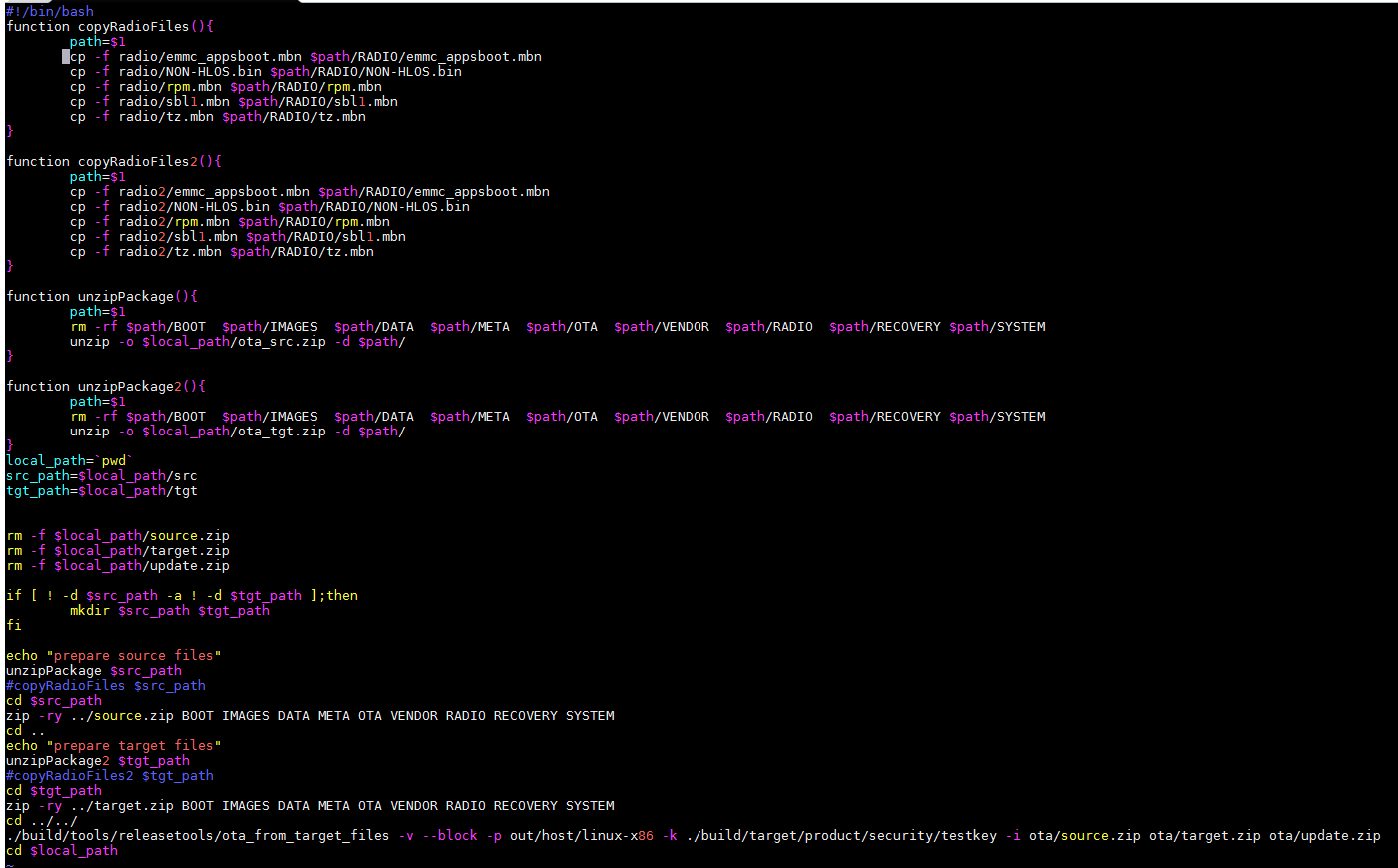
1. 保证制作FOTA的工具及其环境配置正确
2. 保证FOTA工具中的release\_key与版本里的一致
3. 编译基版本和目标版本 （他们之间的差异可以只改一个版本号测试）
4. 在LINUX环境下，执行2个版本下的get\_all\_fota\_files.sh,生成文件夹fota\_files,(其中圈出的地方可能因项目不同做相应目录名的修改和增删)



1. 将基础版本生成的fota\_files下的文件全部放入FOTA工具下的software/src/下，将目标版本生成的fota\_files下的文件全部放入FOTA工具下的software/tgt/下
2. 回到FOTA主目录，执行prepare\_pkg\_start.sh，再执行gen\_update\_pkg\_start.sh
3. 等待执行完成，即可在software/下生成update.zip文件，此包为FOTA升级包
4. 执行adb sideload update.zip进行测试，查看版本号是否从基版本变为目标版本号，或者执行模拟升级：
5. adb root
6. adb push update.zip /cache/
7. adb shell
8. cd /cache/recovery/
9. touch command
10. echo --update\_package=/cache/update.zip >command
11. adb reboot recovery

**2 .android系统的FOTA制作及修改**

1. 请参考例子文件(walabot\_FOTA\_readme\_text.docx)
2. 其中LINUX/android/ota/run.sh文件为自定义添加的，方便制作FOTA升级包，可参照自行修改(此项目由于不需要更新RADIO，所以屏蔽掉)

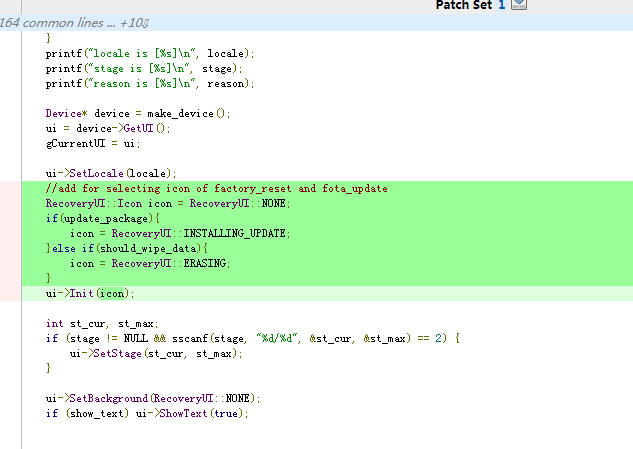


**3.FOTA升级动画的制作及修改**

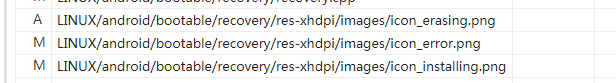
**(1)安装环境，根据<https://blog.csdn.net/cursem/article/details/53132797>**

**(2)向客户请求需要尺寸的一套FOTA动画，如320X240图片，然后按照（1）提供的方法进行动画合成**

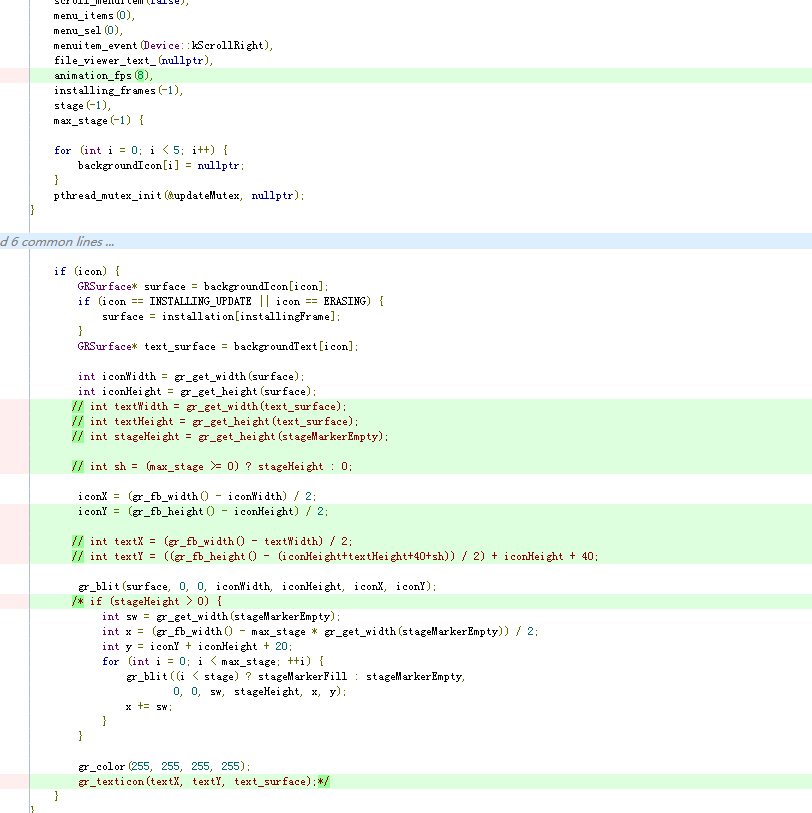
**(3)** 修改文件LINUX/android/bootable/recovery/recovery.cpp

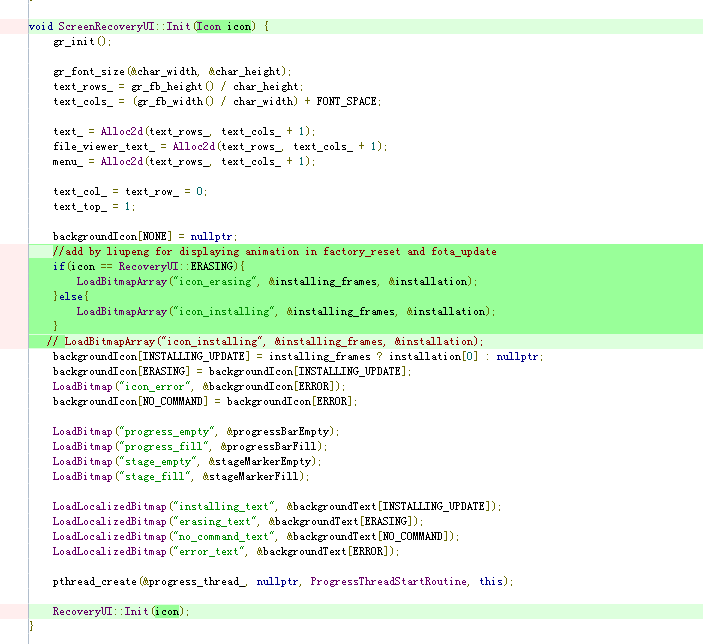
****

**替换图片文件**

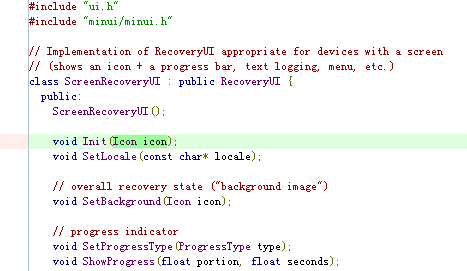
****

**修改文件**LINUX/android/bootable/recovery/screen\_ui.cpp

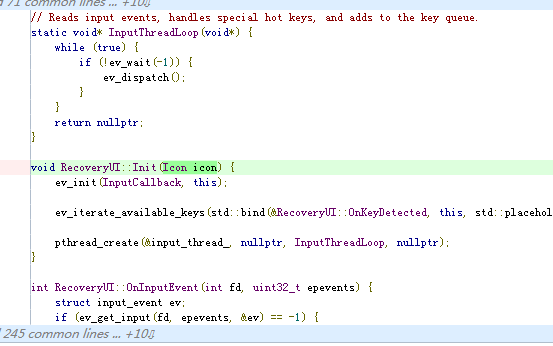




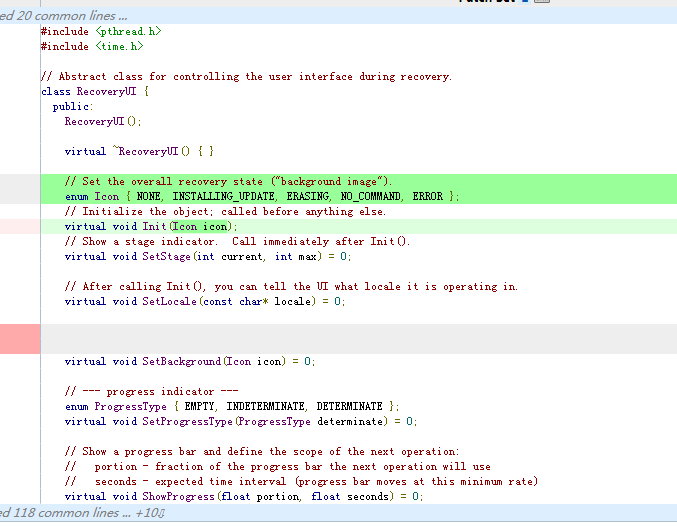
修改LINUX/android/bootable/recovery/screen\_ui.h



修改LINUX/android/bootable/recovery/ui.cpp



修改LINUX/android/bootable/recovery/ui.h



# MODEM篇

## 1.MODEM相关的RFC修改

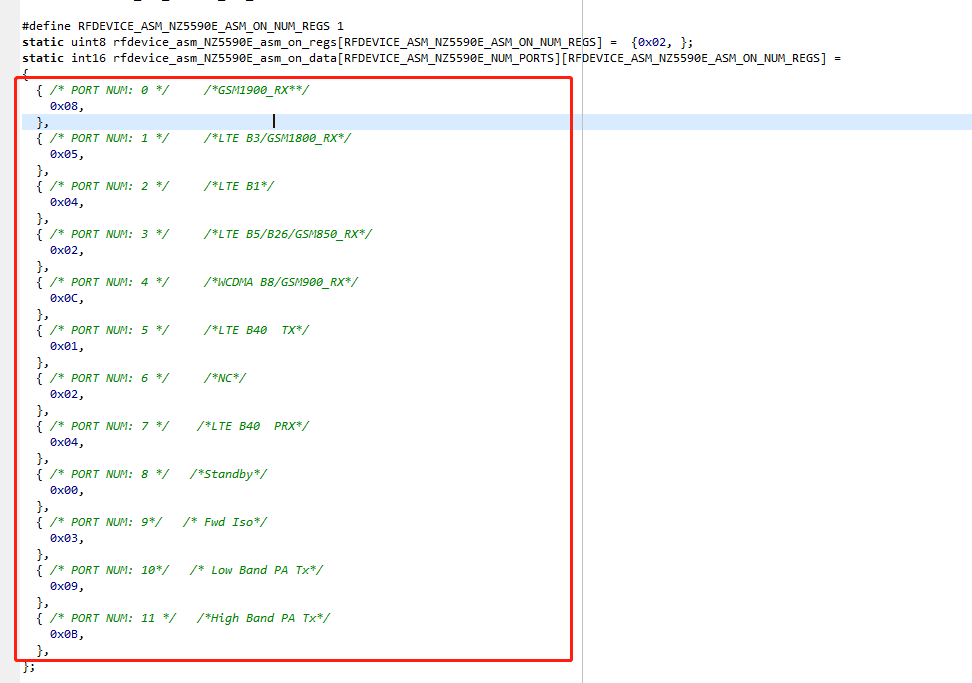
File path:

Modem\_proc/rfc\_jolokia/

Modem\_proc/rfdevice\_asm/

Modem\_proc/rfdevice\_pa/

文件夹rfdevice\_asm和rfdevice\_pa为相关使用的PA器件的定义和asm开关定义，此部分代码由PA厂修改，我们只需要利用其中的一些定义即可,如：TXRX修改时，需要参考以下文件的定义，选择正确的PORT，可根据硬件射频要求，对照硬件datasheet



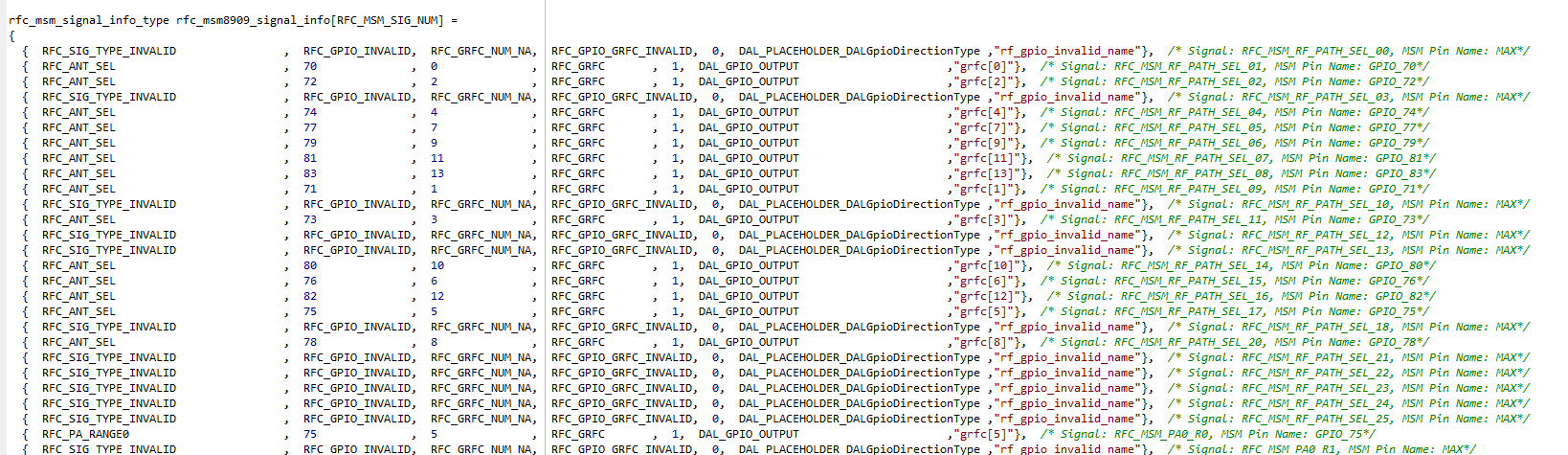
在文件rfc\_jolokia中，此文件为RFC文件，定义了驱动及相关band通路的设定等功能。

File path:

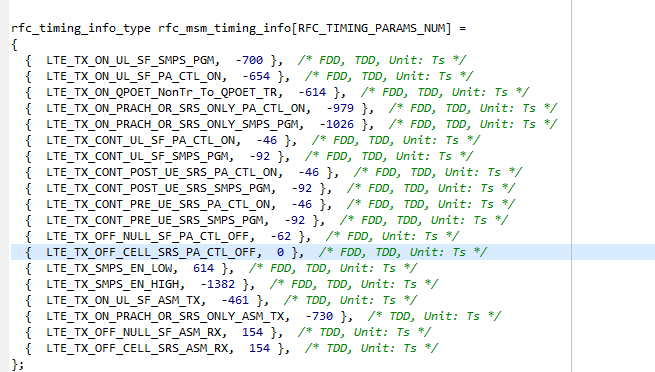
modem\_proc\rfc\_jolokia\target\msm8909\src\ rfc\_msm\_signal\_info\_ag.c

在rfc\_msm\_signal\_info\_ag.c中，定义了所有的信号与GPIO的关系，以及相关的PA时序，在后面设置RXTX时会用到相关的信号定义

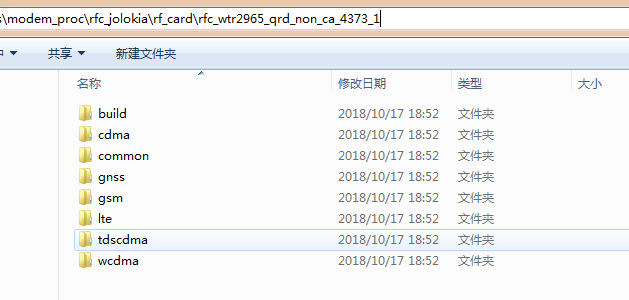
信号定义:



PA时序设定:



在如下的目录中，分别定义了各种网络制式的驱动及band设定,common文件为共有的设定



下面举例一种制式说明:LTE和common文件的修改

Common目录：

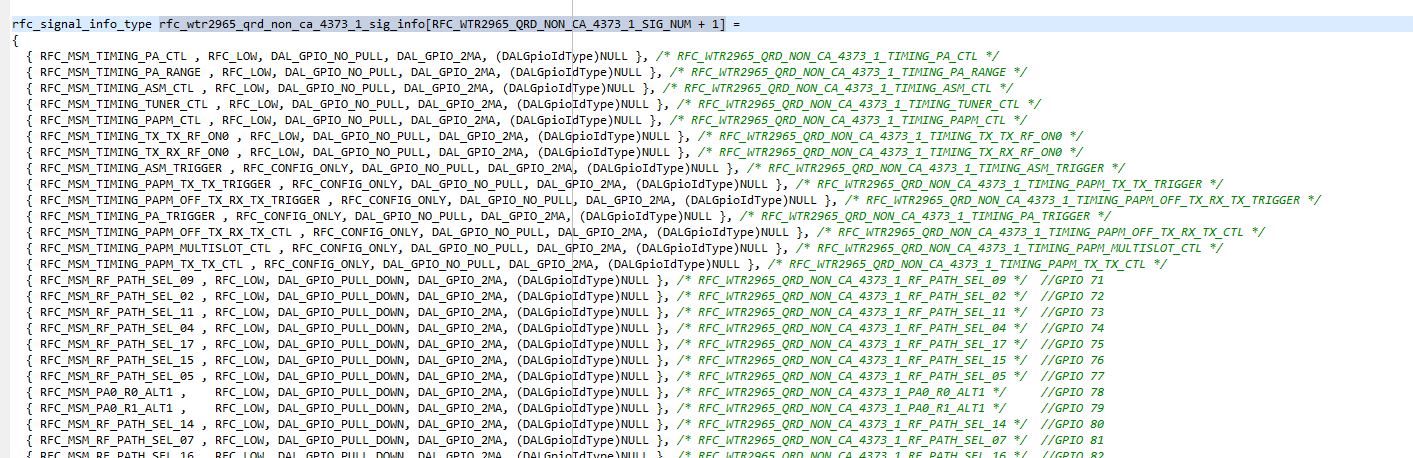
common\src\ rfc\_wtr2965\_qrd\_non\_ca\_4373\_1\_cmn\_ag.cpp

此.cpp文件定义了所使用的物理设备及其相关主集/分集所使用的band.

首先在rfc\_wtr2965\_qrd\_non\_ca\_4373\_1\_phy\_devices\_list[]中定义了PA,ASM的相关信息，值得注意的是红圈部分的数字，如果有多个设备，请一定要按顺序从1-2-3-4的设置，逐次加1的顺序，否则可能会造成机器开机无限重启的问题.

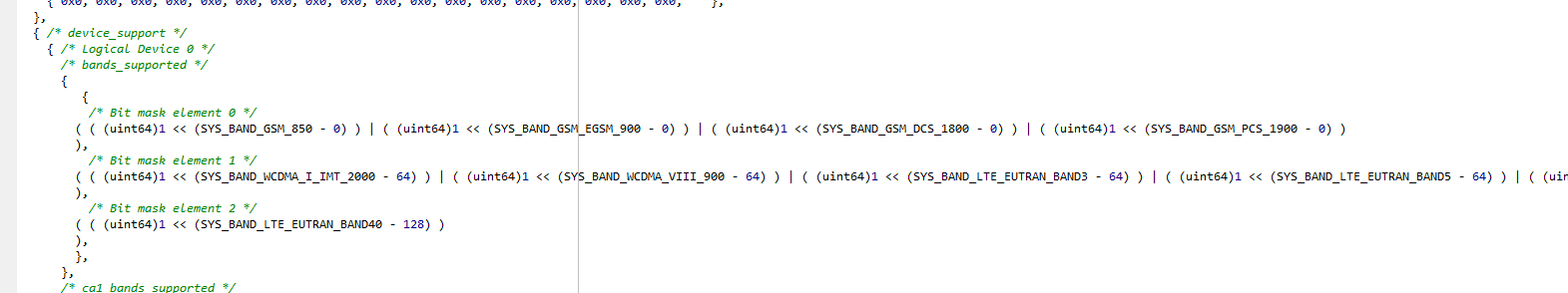


在rfc\_wtr2965\_qrd\_non\_ca\_4373\_1\_sig\_info[RFC\_WTR2965\_QRD\_NON\_CA\_4373\_1\_SIG\_NUM + 1]列表中，需要注意一个GPIO是否被2种信号重复定义，若重复定义可能会造成GPIO在某些band中无效

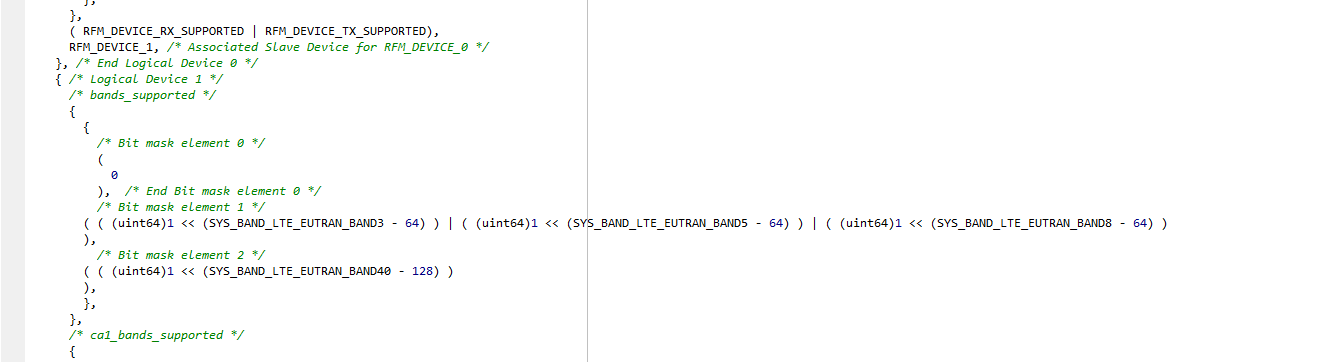


在rfc\_wtr2965\_qrd\_non\_ca\_4373\_1\_logical\_device\_properties中，硬件所需要的band必须加入到相应的主集(rx0)和分集(rx1)中,否则会造成切换band会死掉

主集band的bit mask设定:



分集band的bit mask设定:



LTE目录:

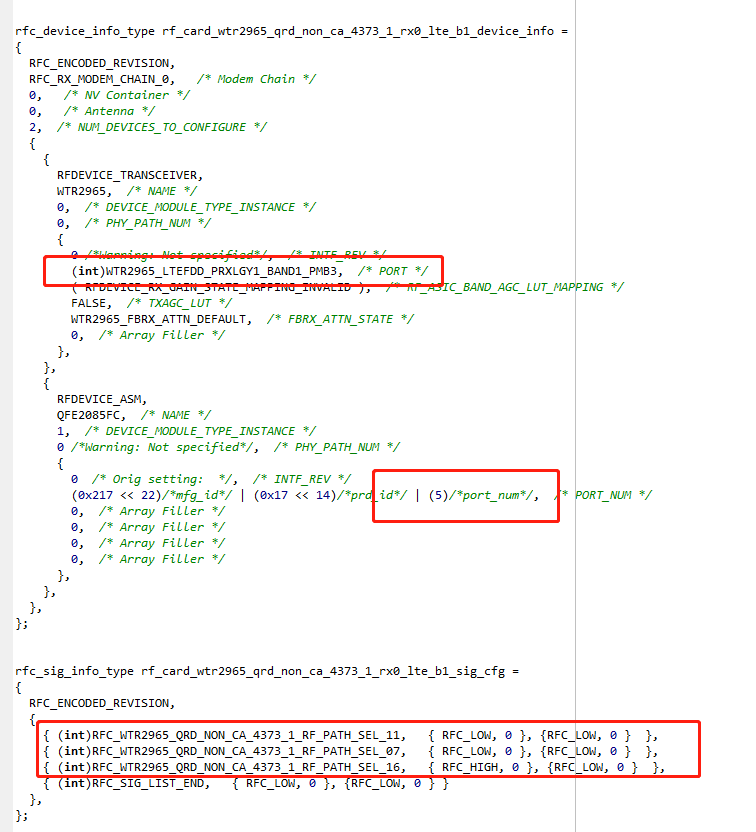
lte\src\ rfc\_wtr2965\_qrd\_non\_ca\_4373\_1\_lte\_config\_ag.cpp

lte\src\ rfc\_wtr2965\_qrd\_non\_ca\_4373\_1\_lte\_config\_data\_ag.c

以上2个文件，举例其中一个band1说明，其他类似.

在.cpp文件中定义了band1使用的的信号，设备，时序，参照修改即可.

在.c文件中,需要找硬件给通路表及原理图，对照修改rx0的通路、端口和GPIO设定

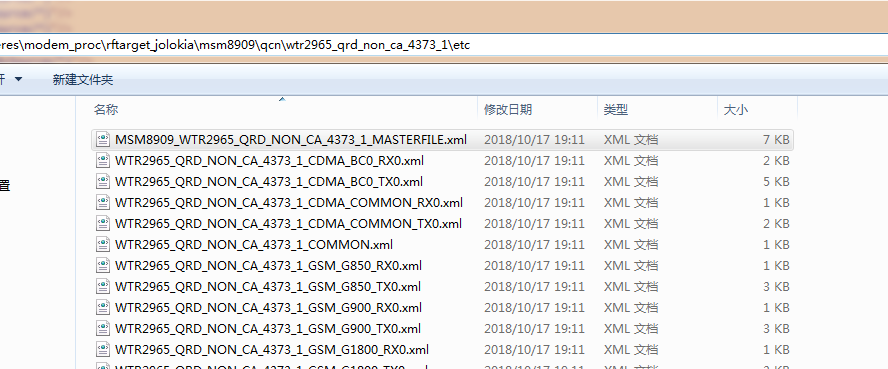


Rx1和TX0同理修改.

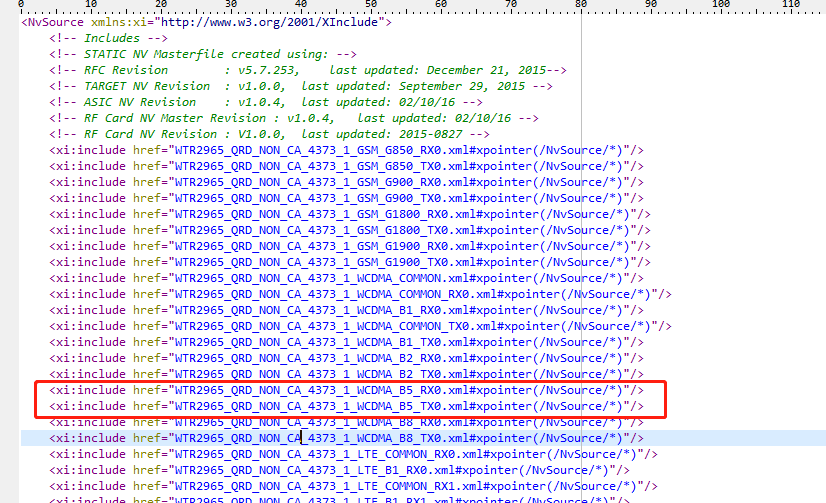
**2.实现工厂复写IMEI/ESN/BT的修改**

**请参考文档(高通平台IMEI\_MEID\_BT\_ESN复写经验分享.docx)**

**3.QCN的制作及修改**



如图，在上图路径中找到MSM8909\_WTR2965\_QRD\_NON\_CA\_4373\_1\_MASTERFILE.xml文件（如要添加band,需要从其他项目复制相关的TX0,RX0,RX1文件到此，并修改此MASTERFILE的文件）



如图BAND5一样添加相应的BAND。

然后利用QRCT3将MSM8909\_WTR2965\_QRD\_NON\_CA\_4373\_1\_MASTERFILE.xml转化为QCN，就可以得到基本的QCN。

如图，

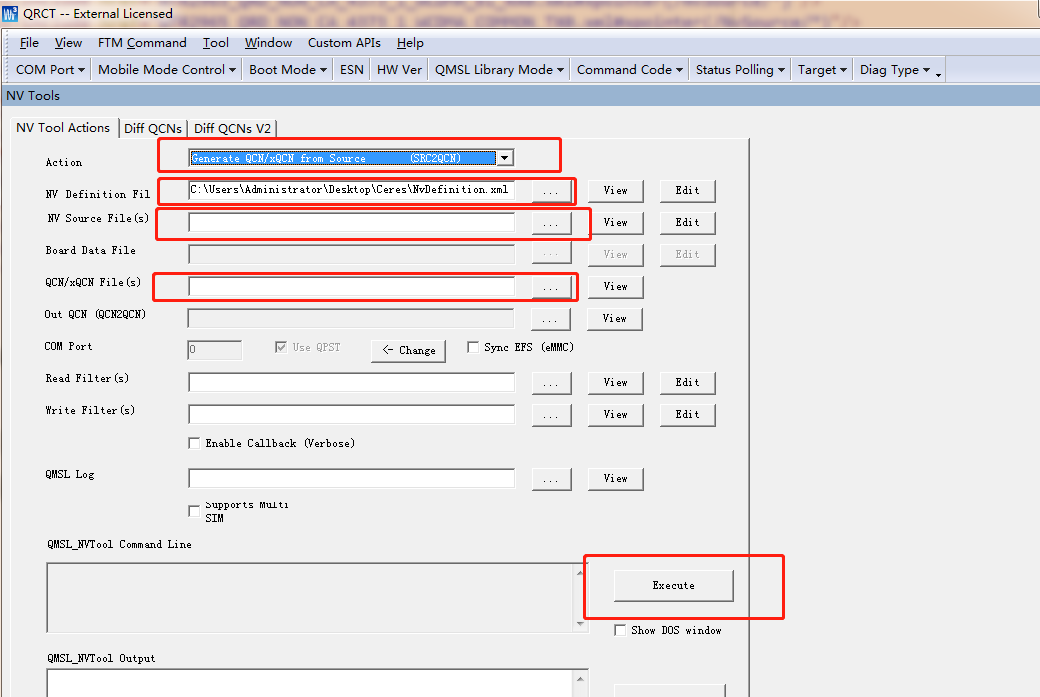
Action:此项可以选择将xml转化为QCN（或QCN转化XML）

NV Definition File:选择NV的定义文件（file path: modem\_proc\rfnv\etc\NvDefinition.xml）

NV Source File(s):选择需要转化的XML

QCN/xQCN File(s):要转化的QCN（名字自定义）

最后，点击Execute生成QCN



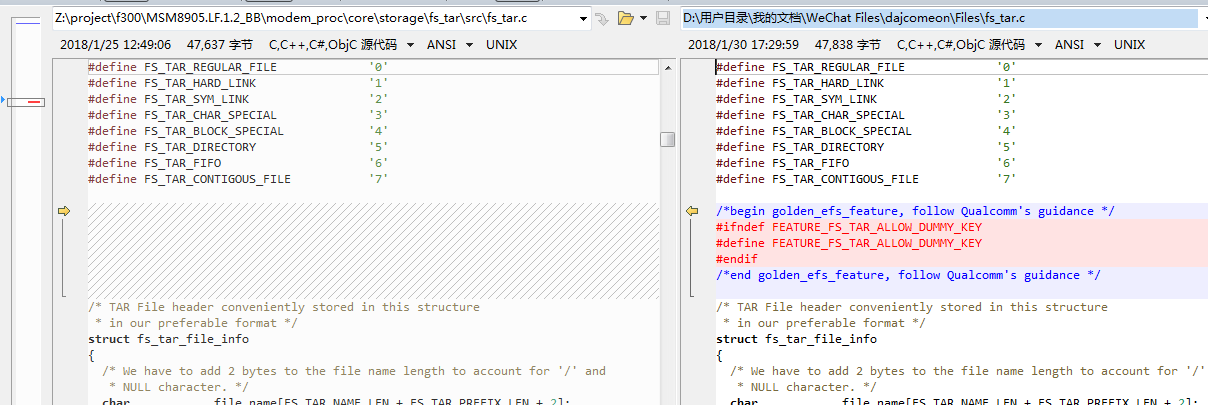
**4.golden efs的制作及修改**

file path:

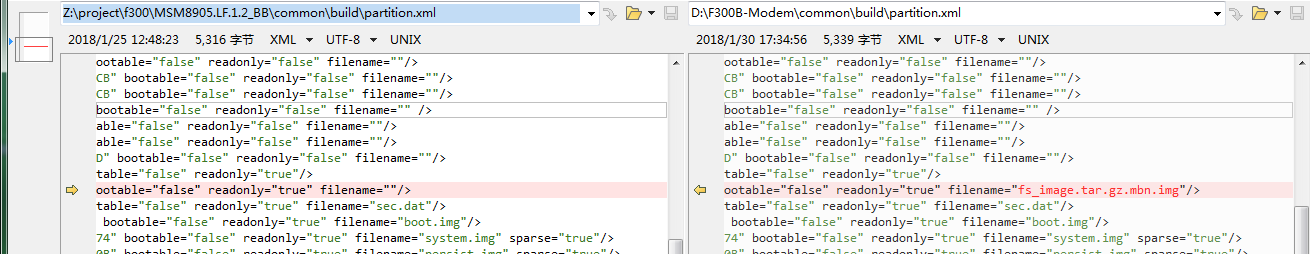
common/build/partition.xml

modem\_proc/core/storage/fs\_tar/src/fs\_tar.c

在.c中修改如图

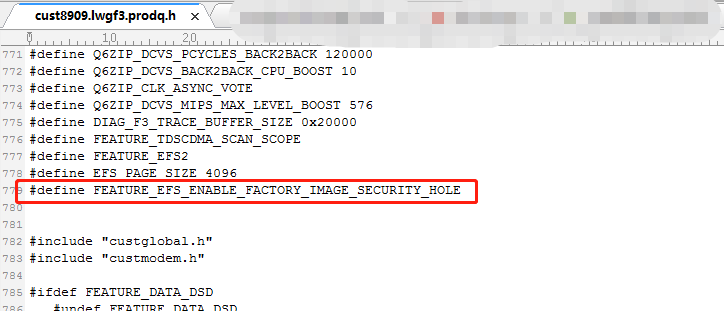


在XML中修改如图,注意必须在第一次的fs\_image.tar.gz.mbn.img制作好了才能修改



在文件modem\_proc\build\ms\cust8909.lwg.prodq.h中增加宏开关（下划线部分是按项目名字区分）

#define FEATURE\_EFS\_ENABLE\_FACTORY\_IMAGE\_SECURITY\_HOLE



保证以上修改后，重新编译MODEM的MPSS,然后取一台手机，刷最新版本后执行以下步骤:

1. 擦除modemst1/modemst2分区 （没有做fsg的情况，不需要执行这一步）
2. 用QPST备份QCN
3. 全擦除fsg/modemst1/modemst2
4. 还原QCN
5. 利用QXDM修改需要的NV值 （若无修改，则忽略此步骤）

然后下载for\_make\_fsg\_env.rar文件，并解压

然后cd到for\_make\_fsg\_env目录下，并在此路径下打开cmd,同时必须打开连接QXDM

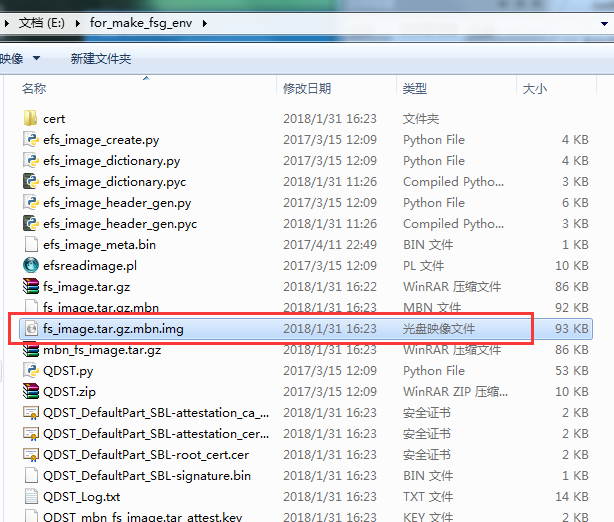
然后依次执行以下指令

1) perl efsreadimage.pl -z

2) python QDSTMBN.py fs\_image.tar.gz

3)efs\_image\_create.py efs\_image\_meta.bin fs\_image.tar.gz.mbn

最后生成fsg文件, 将fs\_image.tar.gz.mbn.img复制到源码中的common/build/下，提交

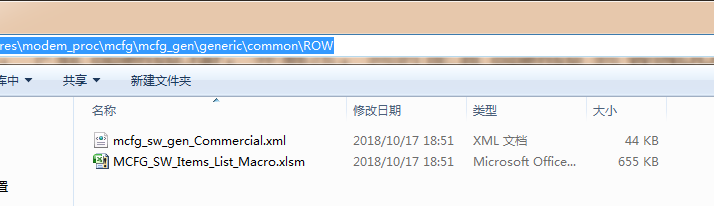


若制作过程有错误，下载openssl.rar，并解压，同时配置openssl到WINDOW全局环境变量中

## 5.MBN的制作及修改

#### 一般MBN的制作和修改:

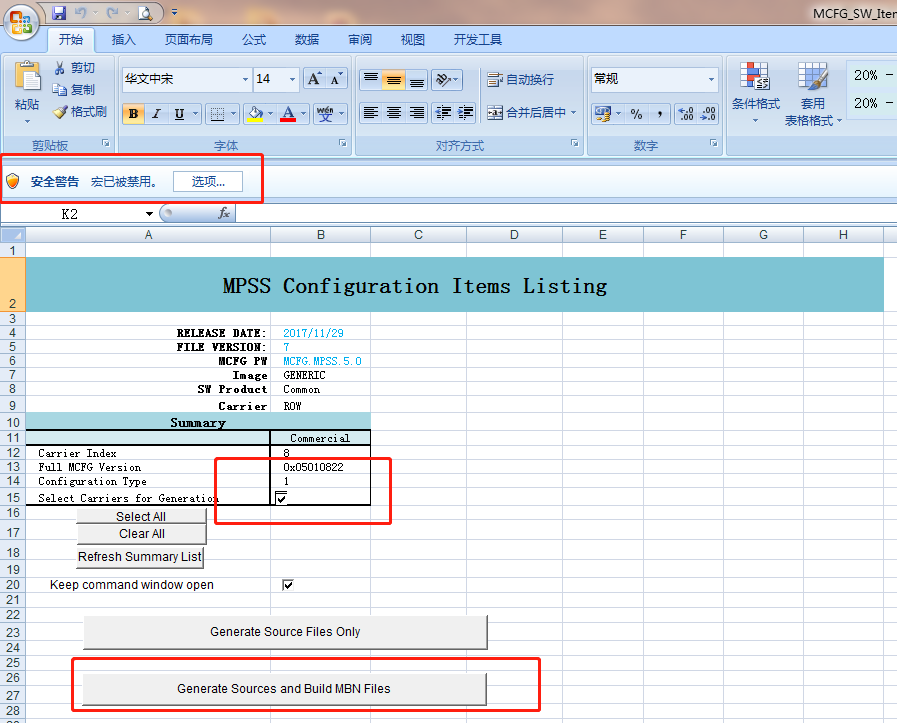
下面以row.mbn为例说明，其他类似



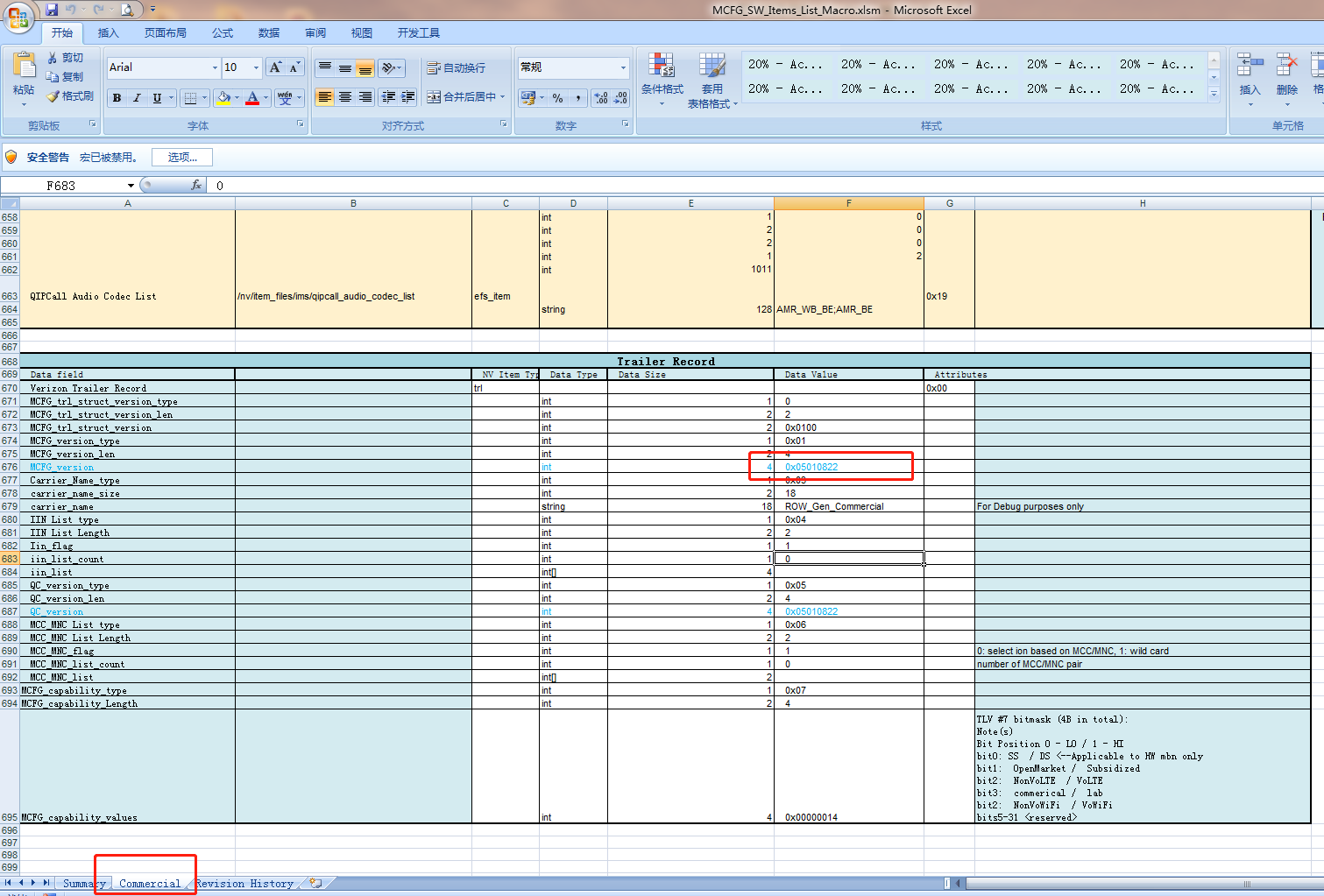
环境要求:

1. 需要安装Microsoft Office 2007或者以上的版本
2. 将MODEM代码拉到本地某个目录下

如上图目录，打开.xlsm表格文件,点击红圈“选项”的地方，选择启用宏,然后保证勾选了MBN，就可以点击”Generate Sources and Build MBN files”,会自动更新XML和MBN,新生成的MBN在” modem\_proc\mcfg\configs\mcfg\_sw\generic\common\ROW\Commercial\”



MBN的修改要点:

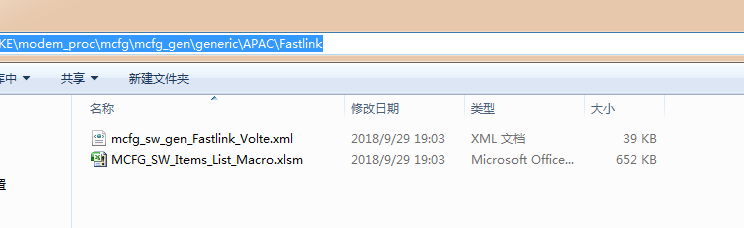


如图，选择相应的MBN相应的表格文件，然后此表中包含了整个MBN的NV定义，同时也可以根据要求修改相应的NV值，值得注意的是:”MCFG version”为版本号，每次修改后一定要加1，配合OTA更新MBN的方式，可以实现只刷机就能更新MBN.

#### 自定义MBN的制作:

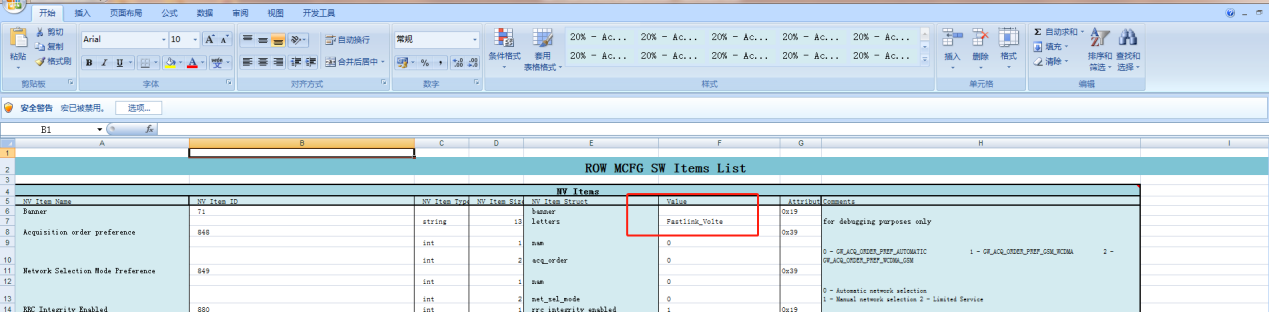
**在做项目过程中，会遇到有些MBN没有的情况，这时候找客户要:运营商NV参数、IIN以及所需求的国家的MCC和MNC。**

**注意：所有自定义的MBN都是以row.mbn为基础修改的，所以只需要复制row.mbn的.xlsm表格修改即可.**

****

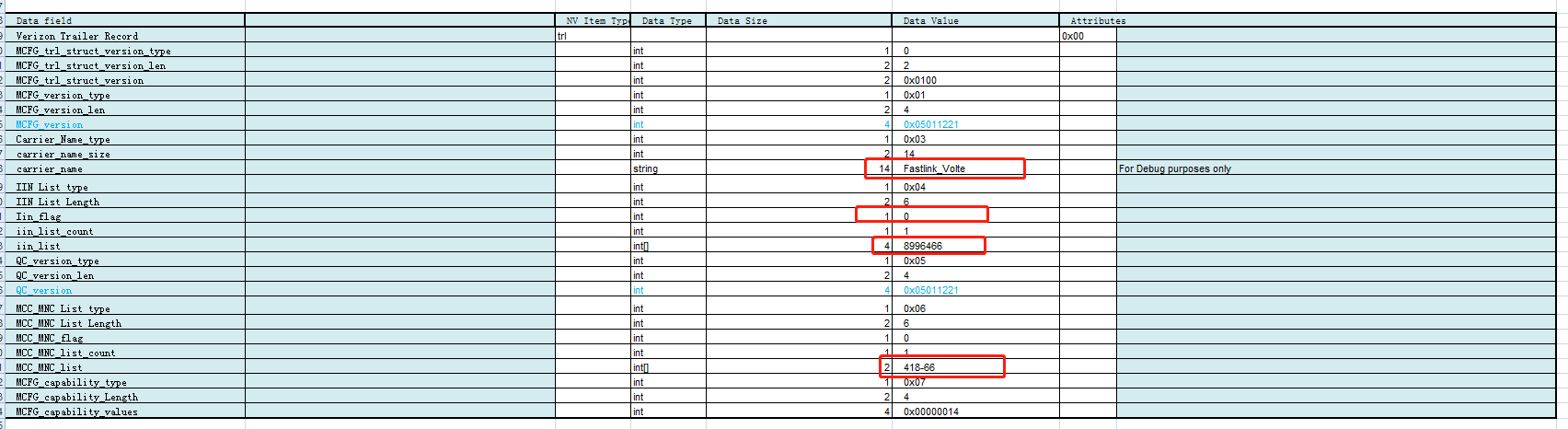
**如图，为自定义增加的一个MBN，自己创建Fastlink目录，复制row.mbn的.xlsm，并修改名字和相关参数定义**

**下面详细说明此自定义MBN的修改细节**

****

**图1**

**如上图，这个名字自己定义，并修改,同时其中需求的NV参数，逐一去查找，然后对比其他的NV参数的修改格式进行修改、增加和删除.**

****

**图2**

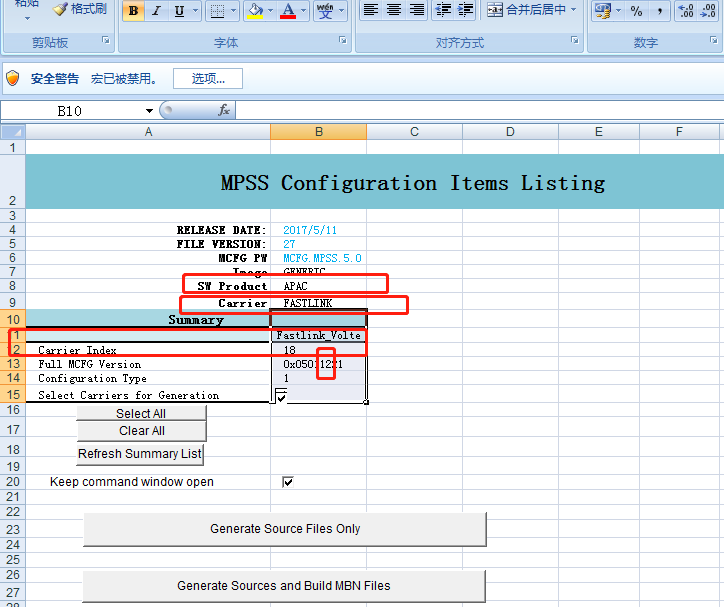
**如上图，**

**carrier\_name: 和图1中的名字保持一致**

**IN\_flag: 改为0，只有通用的row.mbn才为1.**

**Iin\_list: 此表可以添加很多个运营商的IIN，一个运营商对于一个IIN。**

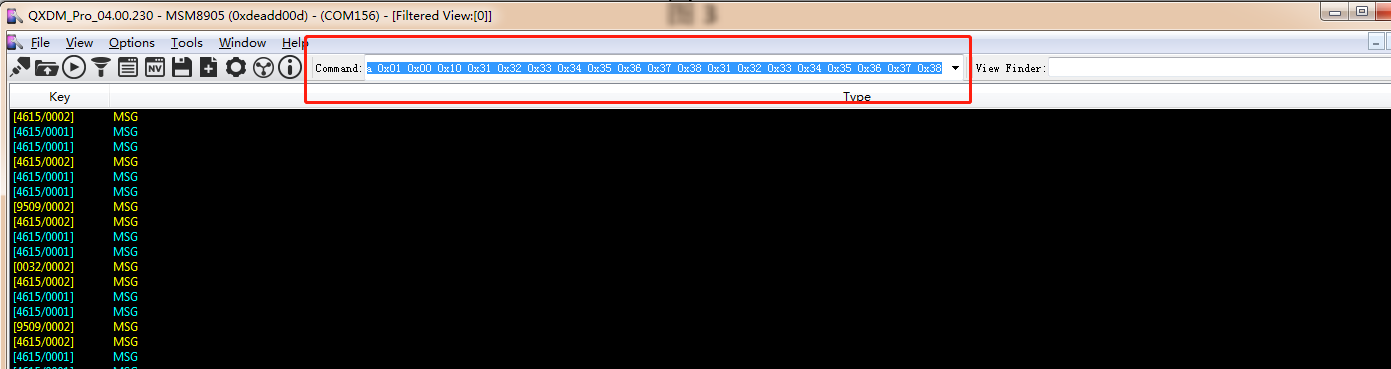
**MCC\_MNC\_list: 此表也可以添加多个MCC-MNC对，保证每个MCC-MNC对都应该有一个IIN**

****

**图3**

**如图，SW\_Product修改为新增文件目录的上层目录,Carrier为运营商名。Carrier\_Index为MCFG version的第5、6位转化成的十进制数字（如0X12=1x16+2=18）,注意此Carrier Index一定不要和所用到的其他MBN的数值一样，可自定义修改成不一样的数值，否则造成无法读取MBN或者MBN名字覆盖等现象.**

## 6. PLMN LOCK的设置



设置PLMN LOCK，需要连接QXDM，然后输入一些command。

下面举例说明这些command的规则要求:

PLMN LOCK COMMAND:

1.send\_data 0x80 0x21 0x60 0xea 0x01 0x00

上面字节中，一般只修改0x01,这个设置锁单卡还是双卡，单卡为0X01，双卡为0X0A。

2.send\_data 0x80 0x21 0x62 0xea 0x01 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0xA0 0x86 0x01 0x00 0x10 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x03 0x34 0x36 0x30 0x02 0x30 0x31 **0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00** 0x34 0x36 0x30 0x02 0x30 0x30 **0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00** 0x35 0x31 0x30 0x02 0x31 0x30 **0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00 0x00**

**步骤2中，**

**前12个字节:只需要修改0X01那位为单卡还是双卡**

**13-16字节:这4个字节为设置迭代次数,不用修改**

**17字节:0X10设置CK长度，一般是8位或者16位，即可为0x08或0X10.**

**18-33字节:这里是16位CK码,取每个字节的个位数，然后可以得到CK=1234567812345678**

**34字节:0X03设置需要的PLMN LOCK个数，这里为三个**

**后面的字节:每20个字节为一组PLMN LOCK。**

**例：取第一组PLMN LOCK。**

0x34 0x36 0x30:前三字节，取个位数，即是MCC=460.

0x02:此为MNC的位数，这里为2位,三位数即为0x03。

0x30 0x31 0x00: 这三位为MNC=01，只有2位的情况，第三位用0x00补。

后面13位字节：用0x00补.

PLMN UNLOCK command:

send\_data 0x80 0x21 0x63 0xea 0x01 0x00 0x10 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38 0x31 0x32 0x33 0x34 0x35 0x36 0x37 0x38

此为PMN LOCK解锁的指令，字节说明与上面一样，唯一的区别就是第三个字节，0X63为解锁字节，0X62为加锁字节，0x60为策略设置字节

7.**PLMN LOCK的清除与熔断机制**

请参考文件(高通平台清除PLMN\_LOCK经验分享.docx)

此方法为了解决:

当还原能上网的QCN还是无法注网的问题，可能就是PLMN LOCK导致的，同时需要查看NV00453是否为0

# 心得总结

“天生我材必有用”，上天生我下来，就一定要做一个对他人，对社会有用的人。我们不需要多牛逼，多伟大，只需要能帮到他人、公司、社会就好。所以请尽量多做事，多体现你的价值。

“学会慢慢长大，而不是慢慢变老”，善于学新的东西，新的挑战，不仅仅只做自己擅长的事，更要尝试不擅长的事，善于虚心向他人请教。

“千里之行，始于足下”，善于做笔记，不管是账号密码，还是有用的网站地址，记录难于记忆的。

“心若不安定，到哪儿都是流浪”，少一些浮躁和抱怨，静下心做事，不管结果是否满意，尽力尽心就行。

。。。。。。未完待续