

# 【CookBook】QCA如何打包image文件

2017年7月3日  
14:35

## 一、IPQ40xx image烧写过程

```
set ipaddr 192.168.1.11
set serverip 192.168.1.xx (This must be the address of the TFTP server)
ping ${serverip}
tftpboot 0x84000000 xxxx-ipq40xx-single.img
imgaddr=0x84000000 && source $imgaddr:script
```

1、IPQ40xx的image文件头包换一个烧写脚本，source为uboot命令，执行烧写脚本进行升级。

## 2、IPQ4018 DK01的分区内容

```
mtd0: 00040000 00010000 "0:SBL1"
mtd1: 00020000 00010000 "0:MIBIB"
mtd2: 00060000 00010000 "0:QSEE"
mtd3: 00010000 00010000 "0:CDT"
mtd4: 00010000 00010000 "0:DDPARAMS"
mtd5: 00010000 00010000 "0:APPSBLENV"
mtd6: 00080000 00010000 "0:APPSBL"
mtd7: 00010000 00010000 "0:ART"
mtd8: 00400000 00010000 "0:HLOS"
mtd9: 01600000 00010000 "rootfs"
mtd10: 00780000 00010000 "rootfs_data"
```

在完整的烧写脚本下，除了ART分区，其他分区均会被重新烧写。

## 二、内核uimage打包过程

当前uimage的打包方式分成两种，一种是legacy传统的，一种是Flat模式的新方式。IPQ40xx使用的便是Flat这种新方式。Flat Image是为了实现unify kernel的梦想，将bsp相关配置从内核分离，一个内核可以拥有很多配置，在启动过程中使用正确配置来适应设备，此功能在ARM处理器上率先被使用。

名词熟悉：

FIT: Flat Image Tree  
FDT: Flattened Device Tree  
DTS: Device Tree Source  
DTB: Device Tree Blob  
ITS: Image Tree Source  
ITB: Image Tree Blob

打包过程:

```
tqm5200.its
+
vmlinux.bin.gz      mkimage      xfer to target
eldk-4.2-ramdisk  -----> tqm5200.itb -----> bootm
tqm5200.dtb          /|\
...                  |
```

```
'new uImage'
```

参考内核文档“source\_file\_format.txt”

对照IPQ40xx工程:

Dts/dtsi文件位于内核arch/arm/boot/dts目录下, 用于表示bsp相关配置, 为4018添加GPIO灯与GPIO按钮便在此处实现。DTB编译成是Dts/dtsi的二进制文件。

its文件位于openwrtqsdk/build\_dir/target-arm\_cortex-a7\_uClibc-1.0.14\_eabi/linux-ipq806x目录下, 用于表达Flat image格式的布局, 里面包含kernel在主机路径、dtb在主机下的路径和config配置。config对应dtb, 当跑在不同板子中的时候, 可以需要相应的config配置。

内核Image.gz在qsdk/build\_dir/target-arm\_cortex-a7\_uClibc-1.0.14\_eabi/linux-ipq806x/Image.gz

### 三、IPQ40xx image打包过程

IPQ40xx image打包同样使用Flat image, 只是头加上了烧写脚本flash.src, Flat Image打包过程如同上述。

mkimage命令使用时必须参考its文件, 来进行img文件的构建。

its文件位于IPQ4019.ILQ.5.0/common/build/ipq/flash.its, 但这个文件通过脚本生成的, 脚本参考位于meta-scripts/ipq40xx\_xxx目录下xxxx.conf文件。

flash.scr是烧写脚本, 位于IPQ4019.ILQ.5.0/common/build/ipq/flash.scr。

flash.its和flash.scr是动态生成的, 修改它们是无效的, 并且因为一次可能要产生好几个Image, 存下来只是最后的样子。

SBL1全称为Secondary boot loader, 第二阶段的BootLoader, 用于启动uboot, 文件名为sbl1\_xxx.mbn, 存放着BOOT.BF.3.1.1/boot\_images/build/ms/bin/40xx/xxx, xxx为nor、nand、nor-plus-emmc、nor-plus-nand, 表示各种启动方式。

MIBIB存放着分区表信息, 放在BOOT.BF.3.1.1/boot\_images/build/ms/bin/40xx/xxx下。

QSEE全称是Qualcomm Secure Execution Environment, 提供一个硬件TrustZone (安全区), 来应对安全威胁, 具体可参考文档“80\_Y8950\_22\_IPQ40XX\_SECURITY\_DESIGN\_USER\_GUIDE.pdf”。文件名为tz.mbn, 位于TZ.BF.2.7/trustzone\_images/build/ms/bin/MAZAANAA。

CDT文件存储着platform ID和DDR配置, 位于BOOT.BF.3.1.1/boot\_images/build/ms/bin/40xx/misc/cdt目录下, 均二进制文件, 使用时拷贝到IPQ4019.ILQ.5.0/common/build/ipq/目录下。

APPSBL存放的是Uboot, APPSBLENV存放着uboot的环境变量。

B00T.BF.3.1.1\boot\_images\build\ms\bin\40xx\misc\tools\config目录下有每个设备型号的cdt, partison、smem等编译需要的配置文件, 但实际上编译版本时, 是不会重新生成bin文件再拷贝到IPQ4019.ILQ.5.0/common/build/ipq/目录下。B00T.BF.3.1.1\boot\_images\build\ms\bin\40xx\misc\tools有如何从配置文件生成cdt、partition、smem二进制文件的工具。

make\_img.sh会调用update\_common\_info.py, 拷贝分区所需要的各个文件到IPQ4019.ILQ.5.0/common/build/ipq相关目录下, 打包IPQ40xx image所有用到的文件均在其中, 但是这些文件都是被拷贝或生成文件, 修改他们不会有任何作用。

#### TIPS:

1、参考内核文档目录“uImage.FIT”, 里面介绍都是FIT相关内容。