

EMMC 量产方法说明

MStar Confidential for Customer Internal Use Only

MStar Confidential for Customer Internal Use Only

© 2015 MStar Semiconductor, Inc. All rights reserved.

MStar Semiconductor makes no representations or warranties including, for example but not limited to, warranties of merchantability, fitness for a particular purpose, infringement of any intellectual property right or the accuracy or completeness of this document, and reserves the right to make changes without further notice to any products herein to improve reliability, function or design. No responsibility is assumed by MStar Semiconductor arising out of the application or user of any product or circuit described herein; neither does it convey any license under its patent rights, nor the rights of others.

MStar is a trademark of MStar Semiconductor, Inc. Other trademarks or names herein are only for identification purposes only and owned by their respective owners.

REVISION HISTORY

Revision No.	Description	Date
V1.0	• Initial release	10/17/2013
V1.1	• 增加烧录器的 ECSD 设定	10/24/2013
V1.2	• 完善 emmcbin.ini 规则的说明	07/15/2014
V1.3	• 增加 reliable write 和 hw reset 相关 ecscd 设置说明	02/02/2015

MStar Confidential for Customer Internal Use Only

TABLE OF CONTENTS

EMMC 量产方法说明.....	1
1. 用母片复制的方法	1
1.1. 简介.....	1
1.2. 烧录 boot1 区	1
1.3. 烧录 user 区	1
1.3.1. 从母片 dump user 区数据	1
1.3.2. 替换 emmc.bin 里的 MBOOT 分区数据	2
1.3.3. 用烧录器烧写 user 区	3
2. 用 Emmcnize 打包镜像的方法	4
2.1. 简介.....	4
2.2. 烧写 boot1 区	4
2.3. 烧写 user 区	4
2.2.1. 生成 mpool.bin.....	4
2.2.2. 用 Emmcnize 打包镜像	5
2.2.3. 用烧录器烧写 user 区	6
2.2.4. 用烧录器烧写 mboot 后 USB 升级主程序	7

1. 用母片复制的方法

1.1. 简介

母片复制,即把母片 emmc flash 里的数据 dump 出来,再用烧录器把 dump 出来的数据烧到空的 emmc flash 里。从而实现 emmc flash 的复制。

我们使用了 emmc flash 里的 boot1 区和 user 区。所以烧录 emmc flash 的时候,分两种情况:

一,如果烧录器不支持同时烧录这两个分区的话,需要分别烧录这 2 个区。

烧写时,需要设置烧录器的 extended CSD register:

1 首先要设置 reliable write, 即

```
ecsd[175] = 0x1
ecsd[167] = 0x1f
ecsd[155] = 0x1
```

设置完成后对 emmc 断电再重新上电

2 烧写 boot1 区时,需要设置 ECSD 179 = 0x49, ECSD 177 = 0x0a

3 烧写 user 区时,需要设置 ECSD 179 = 0x48, ECSD 162 = 0x1

二,如果烧录器支持同时烧录 boot 和 user 区。

烧写时,需要设置烧录器的 extended CSD register:

1 首先要设置 reliable write, 即

```
ecsd[175] = 0x1
ecsd[167] = 0x1f
ecsd[155] = 0x1
```

设置完成后对 emmc 断电再重新上电

2 需要设置 ECSD 179 = 0x48, ECSD 177 = 0x0a, ECSD 162 = 0x1

1.2. 烧录 boot1 区

boot1 区里存放的是 rom_emmc_boot.bin。由于 rom_emmc_boot.bin 里的 hash0 可能会被 efuse 里的 unique key 加密,所以需要直接烧录原始的 rom_emmc_boot.bin 到 boot1 区。

1.3. 烧录 user 区

1.3.1. 从母片 dump user 区数据

烧录 user 区,需要在 mboot 命令行里把母片 emmc flash 里的数据 dump 到 U 盘里。由于 FAT 4G 文件大小的限制,所以这里分 2 种情况介绍 emmc.bin 的生成。

1. 总容量小于 4G 的 emmc flash 步骤

1) 确定 U 盘的利用的 port number

插上 U 盘(格式化为 FAT32 格式,容量应不小于 EMMC 总容量)后,用命令确定 USB port

```
#usb start [port number]
```

比如确认是否插在 port 1, 输入

```
#usb start 1
```

- 如果打印发现设备，则当前确用 port 1。
- 2) 使用 `emmcbin` 命令将 EMMC 中数据复制到 U 盘文件中。

`#emmcbin [port number]`

比如上面使用 port 1, 输入

`#emmcbin 1`

等待进度条完成并提示 dump 完成后，会在 U 盘下生成一个名为 `emmc.bin`，大小跟 EMMC 容量一致的文件（emmc 容量可通过 `mmc size` 命令查询）。

2. 总容量大于 4G 的 emmc flash 步骤

- 1) 插上 U 盘(格式化为 FAT32 格式，容量应不小于 EMMC 总容量)后，如果不确定插哪个 usb port, 可以先用

`#usb start [port number]`

命令确认，比如确认是否插在 port 1, 输入

`#usb start 1`

如果打印发现设备则当前确用 port 1。

- 2) 使用 `mmc dd` 命令将 EMMC 数据复制到 U 盘中。

`#mmc dd mmc2usb [port number]`

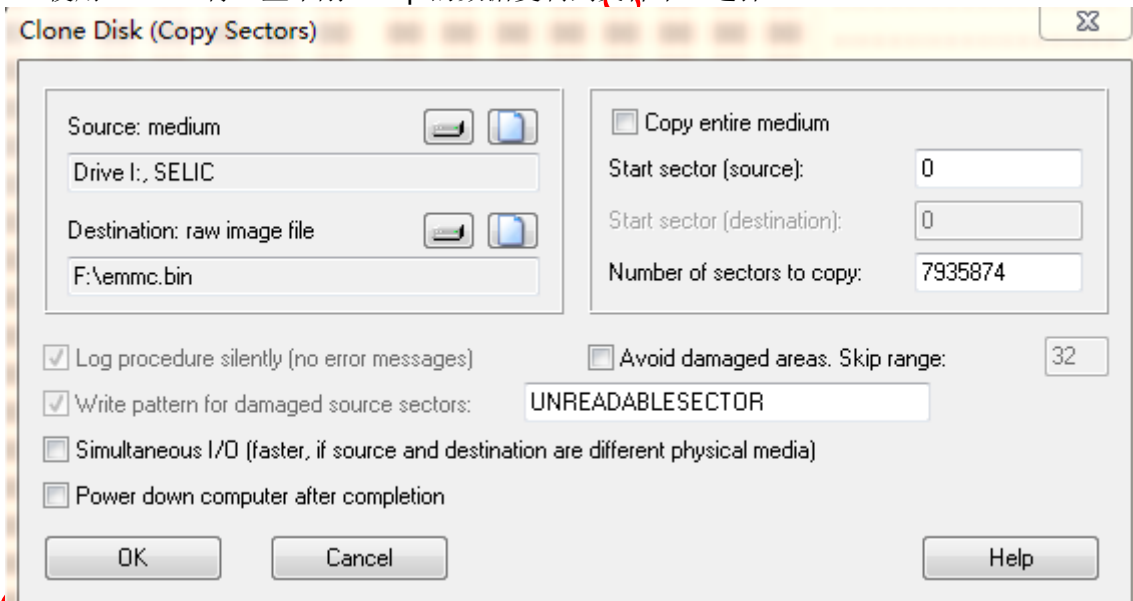
比如上面使用 port 1, 输入

`#mmc dd mmc2usb 1`

等待复制完成(提示 100%)后，记住 dump 下来的总大小 (block 数)，其实也就是 EMMC 总容量，会在打印的最后一行给出来：

Dump Block: XXX

- 3) 在 PC 上使用 winhex 将 u 盘中刚 dump 的数据复制到文件中。选择 Tools->Disk tools->Clone disk。



Source 选 U 盘，Destination 选择新创建目标文件目录。Start sector 填 0，即从 U 盘开始处读数据，注意这里的 Number of sectors to copy 需要填 2) 中 “Dump Block: ” 后跟的大小，然后点 OK 等待制作完成。

注：参考 rui.wang 写的《Emmc_MLC_bin 制作方法.doc》

1.3.2. 替换 emmc.bin 里的 MBOOT 分区数据

从母片 dump 出来的 `emmc.bin` 里 MBOOT 分区数据，实质是 `mboot.bin`。但是 `mboot.bin` 里的 CustomerKeyBank 被母片对应的 efuse 里 unique key 加密了，所以需要把 `emmc.bin` 里 MBOOT 分

区的数据还原为原始的 **mboot.bin**。

在 linux 下用程序 mbootnize 处理即可：

```
#!/mbootnize mboot.bin emmc.bin new_emmc.bin
```

参数说明：

mboot.bin: 原始的 mboot.bin。

emmc.bin: 从母片里 dump 出来的数据。

new_emmc.bin: 替换 MBOOT 分区数据后的 emmc.bin。

1.3.3. 用烧录器烧写 user 区

用烧录器把生成的 new_emmc.bin 烧到 user 区即可。

MStar Confidential for Customer Internal Use Only

2. 用 EMMCNIZE 打包镜像的方法

2.1. 简介

用 Emmcnize 打包镜像是指根据原始的分区镜像文件，生成烧录 user 区的 emmc.bin。和从母片复制方法的不同之处是，不需要从母片 dump 数据，可以完全根据原始的各个分区的 image 来制作 emmc.bin。同样，也需要烧 boot1 区和 user 区。

我们使用了 emmc flash 里的 boot1 区和 user 区。所以烧录 emmc flash 的时候，分两种情况：

一，如果烧录器不支持同时烧录这两个分区的话，需要分别烧录这 2 个区。

烧写时，需要设置烧录器的 extended CSD register:

1 首先要设置 reliable write，即

```
ecsd[175] = 0x1
ecsd[167] = 0x1f
ecsd[155] = 0x1
```

设置完成后对 emmc 断电再重新上电

2 烧写 boot1 区时，需要设置 ECSD 179=0x49, ECSD 177=0x0a

3 烧写 user 区时，需要设置 ECSD 179 = 0x48

二，如果烧录器支持同时烧录 boot 和 user 区。

烧写时，需要设置烧录器的 extended CSD register:

1 首先要设置 reliable write，即

```
ecsd[175] = 0x1
ecsd[167] = 0x1f
ecsd[155] = 0x1
```

设置完成后对 emmc 断电再重新上电

2 需要设置 ECSD 179 = 0x48, ECSD 177=0x0a

2.2. 烧写 boot1 区

直接用烧录器把 rom_emmc_boot.bin 烧到 boot1 区

2.3. 烧写 user 区

2.2.1. 生成 mpool.bin

如果 mboot secure boot 有打开(比如 GTV 的案子)，则需要按下面步骤生成 mpool.bin。其他未打开 secure boot，此步骤可直接跳过，在 2.2.2 节里面直接给 mpool 分区填空值即可(emmcnize 找不到分区对应的镜像就会在分区对应位置填空值)。

MPOOL 分区里面存放了 boot、recovery、tee 对应的 secureinfo 信息和 nuttx_config.bin，所以需要根 据 secure_info_boot.bin、secure_info_recovery.bin、secure_info_tee.bin、nuttx_config.bin 来生成 mpool.bin。

文件夹 generator_mpool 里的文件：

└── image


```

|   |—— nuttx_config.bin
|   |—— secure_info_boot.bin
|   |—— secure_info_recovery.bin
|   |—— secure_info_tee.bin
|—— generator_mpool.sh
|—— util
    |—— align
    |—— crc
    |—— MergeBin.exe
    |—— SeparateBin.exe

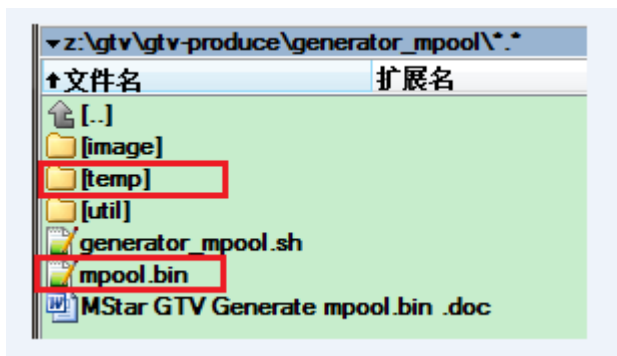
```

1) 准备 image

把 secure_info_boot.bin、secure_info_recovery.bin、secure_info_tee.bin 和 nuttx_config.bin 拷贝到 image 文件夹里。

2) 运行 generator_mpool.sh 脚本，生成 mpool.bin 文件：

selic.cheng@tc-002:~/gtv/gtv-produce/generator_mpool \$./ generator_mpool.sh
这时会生成 mpool.bin，temp 目录里面生成一些中间文件。



3) 把 mpool.bin 拷贝出来，后面 Emmcnize 打包镜像时要用到

注意事项：

开机的时候，MPOOL 分区里的 ENV 是空的（做 mpool.bin 的时候，没有把 ENV 写进去），mboot 会执行命令 envload，从 tvservice/config/set_env 里 load ENV，所以要确保 tvservice.img 里 set_env 的正确性。

2.2.2. 用 Emmcnize 打包镜像

Emmcnize 是将多个分区镜像（mboot.bin，mpool.bin，recovery.img，boot.img，system.img 等等）打包成一个单独的镜像，并在该镜像头部添加 Emmc 分区表的工具。使用步骤：

1. 将 Emmcnize_V2 和所有待打包分区镜像、ini 配置文件放在同文件夹下；
2. 执行 “./emmcnize_V2 XX.ini XX.bin”，等待打包完成。其中 XX.ini 是配置文件名，XX.bin 是打包后的单独镜像名，比如配置文件名为 emmcbin.ini，需要打包到单独镜像 emmc.bin，则敲 “./emmcnize_V2 emmcbin.ini emmc.bin”。
3. 生成的 emmc.bin 就可以直接拿烧录器烧到 emmc 中了。
4. 如果想生成只包含分区表的 bin，执行 “./emmcnize_V2 XX.ini XX.bin -p”，即在末尾加 -p 参数。

Emmcnize通过读取ini配置文件来建立分区表。ini配置文件格式如下：

```
[MBOOT-volume]
image=mboot.bin
vol_id=1
vol_size=0x300000
vol_name=MBOOT

[MPOOL-volume]
image=mpool.bin
vol_id=2
vol_size=0x200000
vol_name=MPOOL

[misc-volume]
image=misc.img
vol_id=3
vol_size=0x80000
vol_name=misc

[recovery-volume]
image=recovery.img.aes
vol_id=4
vol_size=0x800000
vol_name=recovery
...
```

其中，

- 用户在Board config里面设置每个分区的顺序和大小，每个分区的size都要保证512Byte对齐，也就是说最小单位是0x200；
 - 编译android后会生成set_partition文件，set_partition里面各个分区的顺序与size要与emmcbin.ini中各个分区的顺序和size保持一致，每个分区的size务必保证512Byte对齐；
 - [XX-volume] 是每个分区信息的头，Emmcnize通过这个头信息来识别整个分区信息；
 - image=XX.img 定义分区镜像名，必须与已有镜像名相同。如果需要划分空白分区，也就是说没有对应的分区镜像的话，image可设成一个不存在的镜像名，但必须有值；
 - vol_id=X 每个分区都有一个单独的vol_id，不可相同。vol_id编号从1开始，依次递增（+1）顺序排列，emmcbin_V2最大支持63个分区，并且分区数据从7M偏移处开始；
 - vol_size=XX 定义分区大小，vol_size必须不小于分区镜像大小，最好与set_partition里面的数值保持一致，并且保证512Byte对齐；
 - vol_name=XX 定义emmc分区名，各分区名不可相同。尽量与烧入分区的镜像名统一以便于区分。
- 以上每个分区的各个参数必须赋值，否则会出错。

2.2.3. 用烧录器烧写 user 区

用烧录器把 emmc.bin 烧到 user 区即可

2.2.4. 用烧录器烧写 mboot 后 USB 升级主程序

如果mboot 随主程序一起烧录的话，按以上步骤进行。

如果先烧录mboot，然后用usb 升级主程序的话，则需要以下步骤：

- 按 2.2 步骤用烧录器把rom_emmc_boot.bin 烧到emmc boot1 区，
- 把 mboot.bin 用烧录器烧到emmc user 区的0x200000-0x500000 位置处，
- 贴片重启后把 2.2.2 节制作出来emmc.bin 用u 盘升级烧写进emmc(user 区)。

其中用烧录器烧录 emmc 时，ecsd 设置请参考 2.1 节。

MStar Confidential for Customer Internal Use Only