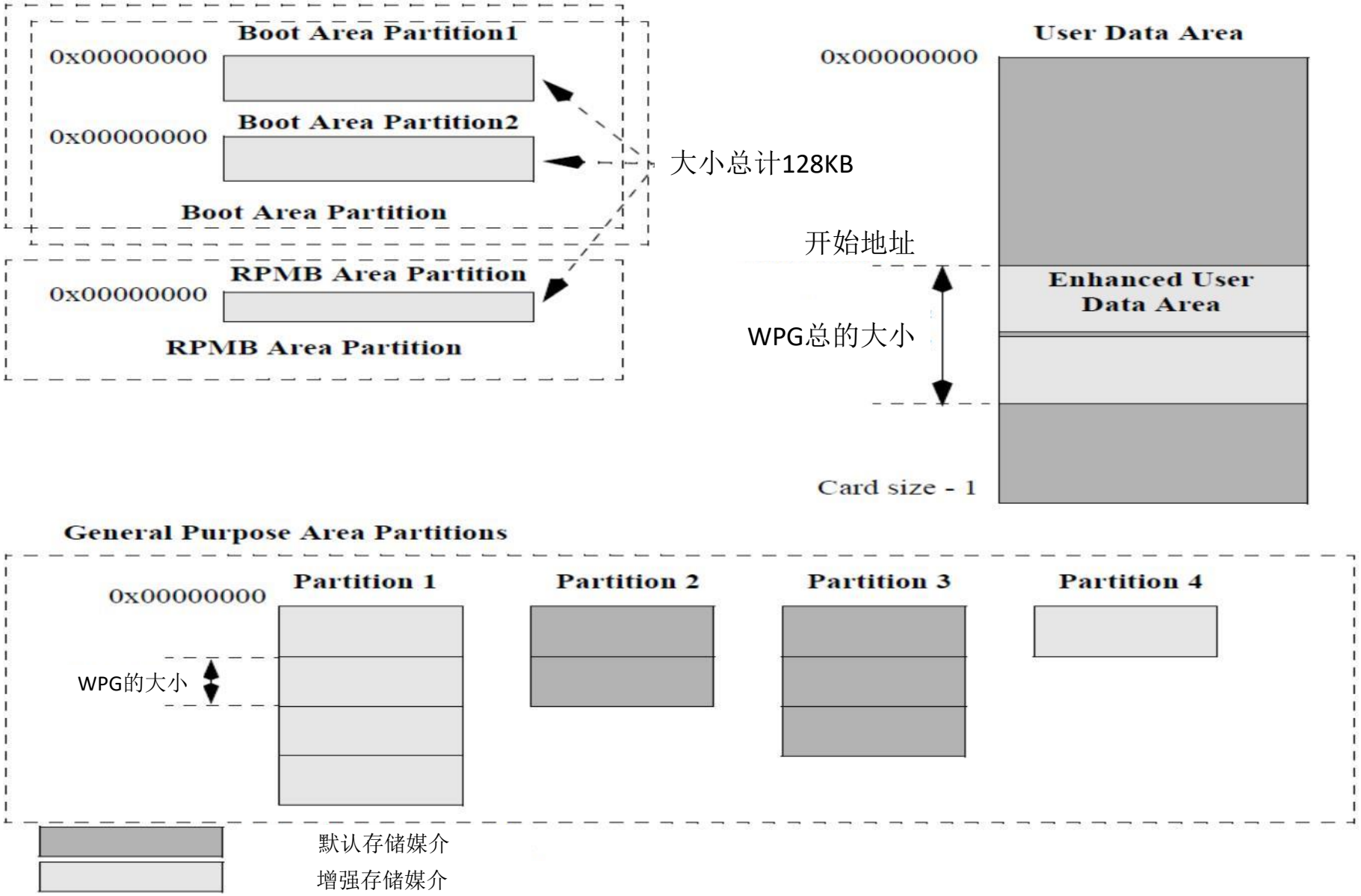
eMMC RPMB, Enhance and GP

Update On 4/2/2019

Biyong Sun

eMMC 布局

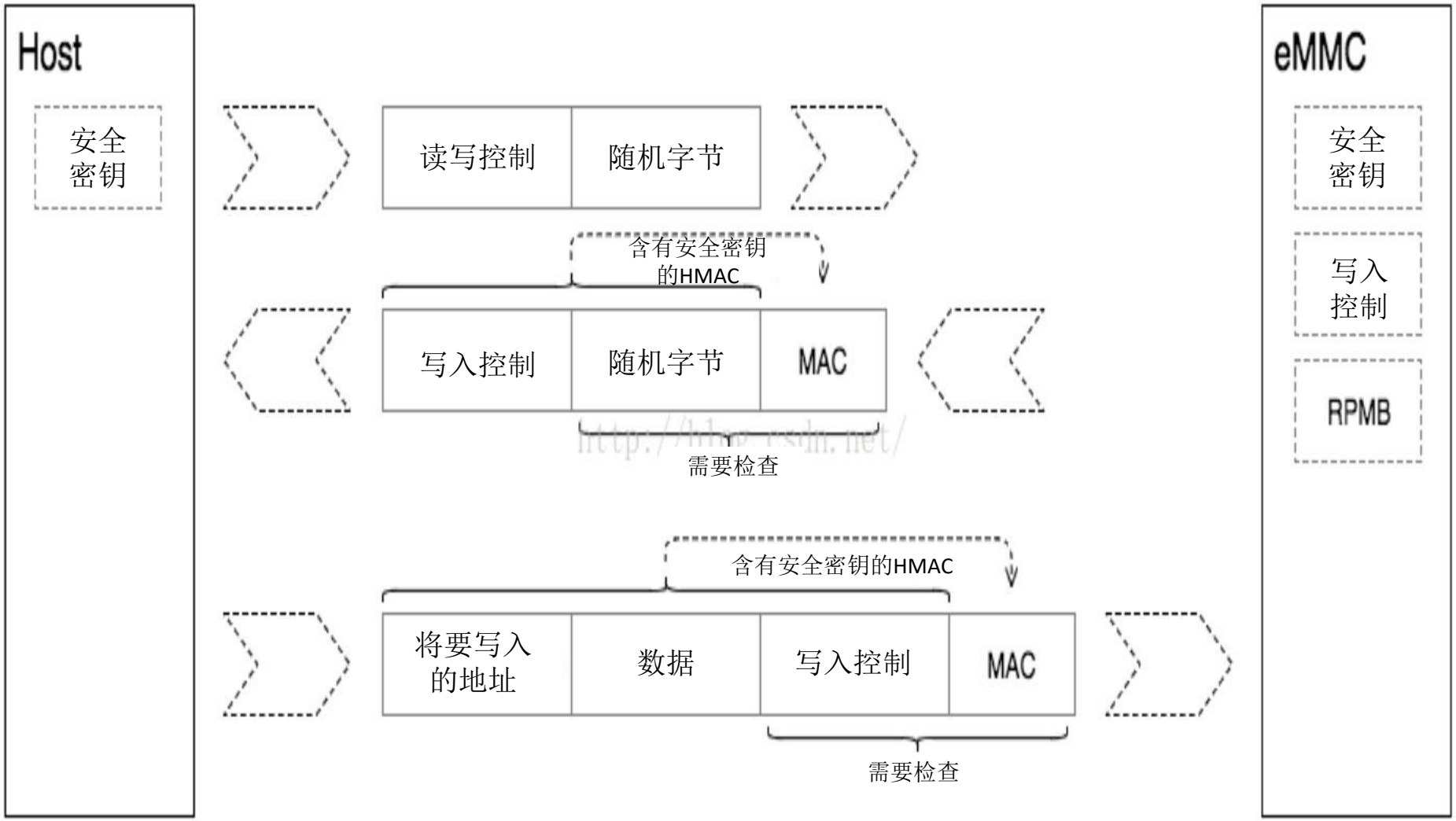

Boot Area Partition1
0x00000000
Boot Area Partition2
0x00000000
Boot Area Partition
RPMB Area Partition
0x00000000
RPMB Area Partition
大小总计128KB
User Data Area
0x00000000
开始地址
Enhanced User Data Area
WPG总的大小
Card size - 1
General Purpose Area Partitions
0x00000000
Partition 1
Partition 2
Partition 3
Partition 4
WPG的大小
默认存储媒介
增强存储媒介

测试环境

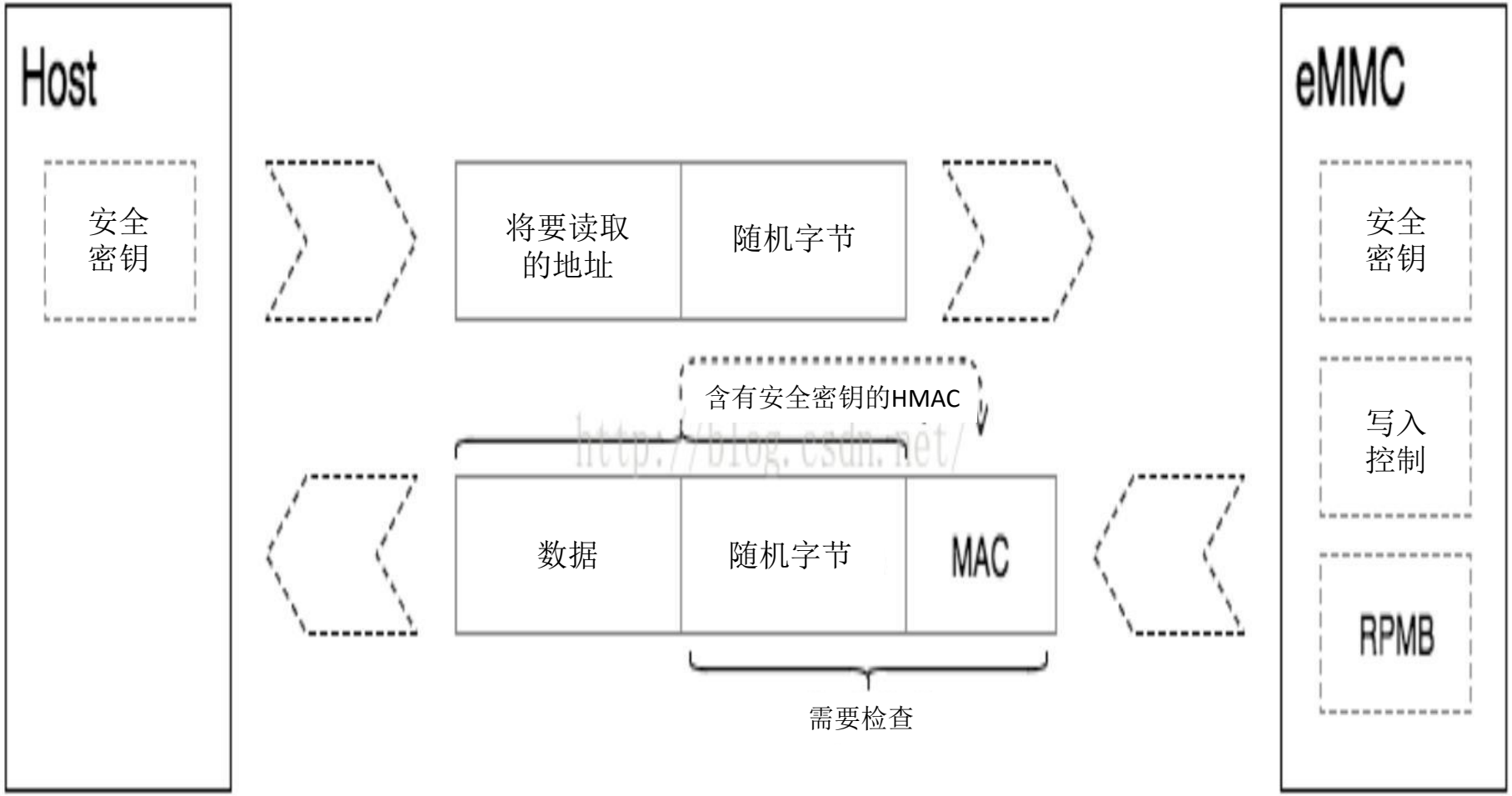
HW: i.MX6Q SDB
SW: L4.1.15_2.0.0

eMMC RPMB
(Replay Protected Memory Block)

RPMB 的写入


Host
安全
密钥
读写控制
随机字节
含有安全密钥
的HMAC
写入控制
随机字节
MAC
需要检查
含有安全密钥的HMAC
将要写入
的地址
数据
写入控制
MAC
需要检查
eMMC
安全
密钥
写入
控制
RPMB

RPMB read


Host
安全
密钥
将要读取
的地址
随机字节
含有安全密钥的HMAC
数据
随机字节
MAC
需要检查
eMMC
安全
密钥
写入
控制
RPMB

mmc-utils中与PRMB有关的指令

mmc rpmb write-key <rpmb device> <key file>

mmc rpmb write-block <rpmb device> <address> <256 byte data file> <key file>
mmc rpmb read-block <rpmb device> <address> <blocks count> <output file> [key file]

请查看mmc的帮助以获得更多信息

使用 RPMB

- 设置密钥(OTP)
- 将密钥和错误密钥写入RPMB
- 从RPMB中读取密钥和错误密钥

设置密钥(OTP)

```
echo 'Authkeymustbe32byteslength_0000' > keyfile.txt

mmc rpmb write-key /dev/mmcblk3rpmb keyfile.txt
```

注意！这是只可烧写一次的(不可逆的)

需要重启电源

将密钥和错误密钥写入RPMB

```
echo
'256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile
.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile
.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile
.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile
.' > data.txt

echo 'Authkeymustbe32byteslength_0000' > keyfile.txt
mmc rpmb write-key /dev/mmcblk3rpmb keyfile.txt

echo 'Authkeymustbe32byteslength_1111' > Wrongkeyfile.txt
mmc rpmb write-block  /dev/mmcblk3rpmb 0  data.txt
Wrongkeyfile.txt
RPMB operation failed, retcode 0x0002
```

从RPMB中读取密钥和错误密钥

mmc rpmb read-block /dev/mmcblk3rpmb 0 1 out.txt
mmc rpmb read-block /dev/mmcblk3rpmb 0 1 out.txt keyfile.txt
cat out.txt
256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.256bytedatafile.

mmc rpmb read-block /dev/mmcblk3rpmb 0 1 out.txt Wrongkeyfile.txt
RPMB MAC mismatch

使用密钥/ MAC（消息身份验证代码）时，请确保数据已认证。不能是通过黑客入侵伪造的数据。

Enhanced User Data Area (pseudoSLC Mode)

什么是Enhanced User Data Area

- 简单而言就是伪造的SLC
- 使该区域更加可靠
- 启用后容量会变小（从MLC到SLC）
- pSLC模式可以提高写入速度

Enable the enhanced user area

- 检 查 MAX_ENH_SIZE_MULT
  mmc extcsd read /dev/mmcblk3 | grep MAX_ENH_SIZE_MULT -A 1
  Max Enhanced Area Size [MAX_ENH_SIZE_MULT]: 0x0000ea
  i.e. 3833856 KiB

- 启用所有区域以增强命令：

mmc enh_area set <-y|-n|-c> <start KiB> <length KiB> <device>

mmc enh_area set **-n** 0 3833856 /dev/mmcblk3
在真正使用**-y**之前，请使用**-n**进行检查

mmc enh_area set -y 0 3833856 /dev/mmcblk3
Enhanced User Data Area Size [ENH_SIZE_MULT]: 0x0000ea
i.e. 3833856 KiB
Max Enhanced Area Size [MAX_ENH_SIZE_MULT]: 0x0000ea
i.e. 3833856 KiB

注意!这 是 只 可 编 程 一 次 的 (不 可 逆 的) 更 改
需要重启电源

GP
(General Purpose Partition)

GP(General Purpose Partition)

- 高达4个GP可用于创建物理分区
- GP具有增强属性

创建GP

- 指令
  mmc gp create <-y|-n|-c> <length KiB> <partition> <**enh_attr**> <ext_attr> <device>
- 检查MAX_ENH_SIZE_MULT
  mmc extcsd read /dev/mmcblk3 | grep MAX_ENH_SIZE_MULT -A 1
  Max Enhanced Area Size [MAX_ENH_SIZE_MULT]: 0x0000ea
  i.e. 3833856 KiB
- 创建2个GP

创建 gp2
mmc gp create **-n** 93888 2 **1** 0 /dev/mmcblk3
Enhanced GP1 Partition Size [GP_SIZE_MULT_1]: 0x00000b
i.e. 90112 KiB
Max Enhanced Area Size [MAX_ENH_SIZE_MULT]: 0x0000ea
i.e. 3833856 KiB
注意：如果并非创建最后一个**gp**，**-n** 仅用来检查和设置**eMMC**寄存器

创建gp1
mmc gp create **-y** 524288 1 **1** 0 /dev/mmcblk3
Enhanced GP1 Partition Size [GP_SIZE_MULT_1]: 0x000040
i.e. 524288 KiB
Max Enhanced Area Size [MAX_ENH_SIZE_MULT]: 0x0000ea
i.e. 3833856 KiB

注意!这是只可编程一次的(不可逆的)更改
需要重启电源

使用 GP

ls /dev/mmcblk3*
mmcblk3 mmcblk3boot0 mmcblk3boot1 **mmcblk3gp0** **mmcblk3gp1**
mmcblk3rpmb

fdisk -l
Disk /dev/mmcblk3: **6** GiB, 6476005376 bytes, 12648448 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk **/dev/mmcblk3gp1**: 88 MiB, 92274688 bytes, 180224 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

Disk **/dev/mmcblk3gp0**: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes

注意：eMMC消耗大约2G来获得增强的88MiB + 512MiB = 600MiB
总容量从大约6.6G增加到大约8G

使用 GP(Cont.)

fdisk /dev/mmcblk3gp0

```
Command (m for help): p
Disk /dev/mmcblk3gp0: 512 MiB, 536870912 bytes, 1048576 sectors
Units: sectors of 1 * 512 = 512 bytes
Sector size (logical/physical): 512 bytes / 512 bytes
I/O size (minimum/optimal): 512 bytes / 512 bytes
Disklabel type: dos
Disk identifier: 0xd0d48a7b

Device            Boot  Start     End Sectors  Size Id Type
/dev/mmcblk3gp0p1        2048  264191  262144  128M  c W95 FAT32 (LBA)
/dev/mmcblk3gp0p2      264192 1048575  784384  383M 83 Linux
```

```
mkfs.vfat -F 32   /dev/mmcblk3gp0p1
mkfs.ext3          /dev/mmcblk3gp0p2
```

Miscellaneous

设备可靠性

mmc write_reliability set <-y|-n|-c> <partition> <device>

Enable write reliability per partition for the <device>.
Dry-run only unless -y or -c is passed.
Use -c if more partitioning settings are still to come.

注意!这是只可编程一次的(不可逆的)更改。

设置快速启动

mmc bootbus set <boot_mode> <reset_boot_bus_conditions> <boot_bus_width> <device>

Set Boot Bus Conditions.

<boot_mode> must be "single_backward|single_hs|dual"

<reset_boot_bus_conditions> must be "x1|retain"

<boot_bus_width> must be "x1|x4|x8“

将总线设置为8位ddr模式

mmc bootbus set dual retain x8 /dev/mmcblk3

设置快速启动(Cont.)

7.4.65 BOOT_BUS_CONDITIONS [177]

This register defines the bus width for boot operation.

Table 135 — Boot bus configuration

Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0
Reserved			BOOT_MODE		RESET_BOOT_BUS_CONDITIONS	BOOT_BUS_WIDTH	

Bit[7:5] : Reserved

Bit [4:3] : BOOT_MODE (non-volatile)

0x0 : Use single data rate + backward compatible timings in boot operation (default)

0x1 : Use single data rate + High Speed timings in boot operation mode

0x2 : Use dual data rate in boot operation

0x3 : Reserved

NOTE HS200 & HS400 is not supported during BOOT operation.

set bus to 8 bit ddr mode

mmc bootbus set <boot_mode> <reset_boot_bus_conditions> <boot_bus_width> <device>

mmc bootbus set **dual** **retain** **x8** /dev/mmcblk3

74.65 BOOT_BUS_CONDITIONS [177] (cont'd)

Bit [2]: RESET_BOOT_BUS_CONDITIONS (non-volatile)

0x0 : Reset bus width to x1, single data rate and backward compatible timings after boot operation (default)

0x1 : Retain BOOT_BUS_WIDTH and BOOT_MODE values after boot operation. This is relevant to Push-pull mode operation only.

Bit[1:0] : BOOT_BUS_WIDTH (non-volatile)

0x0 : x1 (sdr) or x4 (ddr) bus width in boot operation mode (default)

0x1 : x4 (sdr/ddr) bus width in boot operation mode

0x2 : x8 (sdr/ddr) bus width in boot operation mode

0x3 : Reserved

原文链接：https://community.nxp.com/docs/DOC-343116

由Biyong Sun于2019-4-12上传的文件