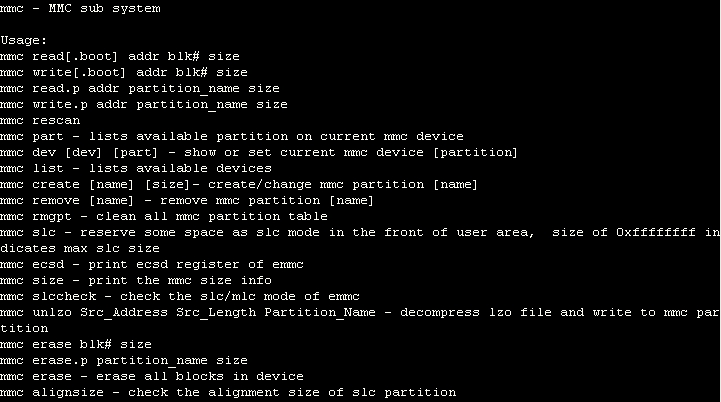
Mboot mmc/ext4相关命令手册

PART I: mmc command.

目前Mboot支持以下mmc命令：



1) mmc read/write.

读写命令。Addr = 内存地址， blk# = 起始block数， size = 需要读写的大小。

比如：#mmc read 0x50000000 0x400b 0x400000, 表示从第0x400b块(0x400b\*0x200 = 0x801600)处开始读0x400000 bytes大小数据到内存0x50000000。 返回打印OK说明读成功。

#mmc write 0x50000000 0x400b 0x400000, 表示将内存起始处0x50000000开始算的数据共0x400000 bytes写到第0x400b块起始处。返回打印OK说明写成功。

Mmc write最新的code在末尾加了一个参数1，格式如下：

#mmc write 0x50000000 0x400b 0x400000 1

表示本次写操作跳过空白无效数据段不写，默认不加的话则不区分将所有数据写进emmc.

2) mmc read.p/write.p/write.p.continue.

也是读写命令，与mmc read/write 不同的是第二个参数不是blk#, 而是分区名partition\_name，这样省去升级时再手动计算各分区便宜。比如：

#mmc read.p/write.p 0x50000000 boot 0x400000

mmc write.p.continue命令是将数据写到指定分区偏移处。比如：

#mmc write.p.continue 0x50000000 userdata 0x6400000 0xa00000,将0x50000000保存的c长度为0xa00000的数据写到userdata分区0x6400000偏移处。

同样返回OK说明读/写成功。同样 write.p/write.p.continue在末尾加上了参数1，与mmc write命令同样效果。

3) mmc rescan

初始化命令，因为在每个命令前都做过，所以用处不多。

4) mmc part

打印当前分区信息。

5) mmc dev

显示当前分区以及分区间切换，这里的分区是mmc硬件分区，并不是mmc part列出来的软件分区。用法：#mmc dev 0 1, 如果当前在硬件分区0, 该命令可以切换至硬件分区1。

6) mmc list

列出当前所用mmc设备命令。

7) mmc create

创建mmc分区，用法：

#mmc create boot 0x400000, 即创建分区名为boot, 大小0x400000 bytes的分区。需要注意的是分区需要依顺序创建，也就是说如果当前已经有2个分区，再敲这个命令的话boot会算为第3个分区。

8) mmc remove

移除mmc分区，用法：

#mmc remove boot，即移除boot分区。

该命令尽量少用，如果想update分区表的话请配合mmc rmgpt和mmc create使用。

9) mmc rmgpt

移除当前所有分区。

10) mmc slc size reliable\_write

设置slc分区大小，size代表要设置的slc分区字节数。

size为0代表不设置slc mode；软件会将size修改为对齐并向上取整，确保实际设置的size不小于用户输入的size；

reliable\_write为0代表不使能reliable write功能，反之则打开reliable write；

如果想将全部可转化空间都转换为slc mode，可以使用mmc slc 0xffffffff；

开发调试用。

11) mmc ecsd

打印ecsd寄存器信息，开发调试用。

12) bin2emmc

将Fat32 U盘中的mmc bin文件烧进mmc中。CL540869支持。

13) mmc setecsd num mask value

修改ecsd寄存器，可以单个bit修改。比如：

mmc setecsd B3 0x40 0x40

表示对ecsd的B3寄存器的bit6置1，而：

mmc setecsd B3 0x40 0x0

则表示对ecsd的B3寄存器的bit6置0

开发调试用，一般情况下禁用。

14) mmc size

显示当前emmc的可用容量，块大小及块总数

显示已经设置的slc size，slc分区的最大size，emmc对reliable write的支持和设置状况。

15) mmc slcchk

检查当前的slc/mlc配置，相关的状态保存在ENV中。

16) mmc relwrchk

检查当前的reliable write配置，相关的状态保存在ENV中。

17) mmc slcrelwrchk

检查当前的slc/mlc和reliable write配置，相关的状态保存在ENV中。

18) mmc unlzo

解压缩lzo分区镜像并写进mmc分区，如将0x50000000地址保存的大小100M的镜像解压写入system分区：

#mmc unlzo 0x50000000 0x6400000 system

19) mmc erase

mmc erase blk#/partition\_name size, 如：

#mmc erase 0 0x400000 => 从0号block处擦除4M大小

#mmc erase boot 0x400000 =>擦除boot分区开始往后4M数据

#mmc erase =>整个chip擦除

20) mmc dd

mmc/usb之间互相dump raw data, 会损坏u盘文件系统，使用前请先备份U盘数据。默认用usb port 0。

#mmc dd mmc2usb =>dump mmc中数据到usb

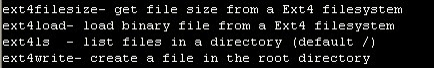
#mmc dd usb2mmc =>dump usb中数据到mmc

21) mmc alignsize

查询SLC分区大小的对齐单位。

PART II: ext4 command.

目前Mboot支持一下ext4 command（CL540514支持）



1) ext4filesize

获取mmc 某个分区下的文件大小，用法：

#ext4filesize mmc 0:7 setenv，即获取mmc设备0，第7个分区下的名为setenv文件大小。

2) ext4load

读ext4文件命令，用法：

#extload mmc 0:7 0x50000000 setenv 0x400，即将mmc设备0，第7个分区下名为setenv文件读出0x400 bytes到内存地址0x50000000处，如果大小不指定的话则是将整个文件读进内存中。

3) ext4ls

列出文件夹/分区下所有信息，用法：

#ext4ls mmc 0:7 /Customer, 即列出mmc设备0,第7个分区下Customer文件夹内信息。

#ext4ls mmc 0:7, 则是列出mmc设备0,第7个分区下的信息。

4) ext4write

写ext4文件到mmc命令，用法：

#ext4write mmc 0:2 /boot/uImage 0x30007fc0 6183120, 将保存在内存起始地址0x30007fc0，大小6183120 bytes数据写到mmc设备0，第2个分区的/boot/uImage文件中，如果没有该文件则自动新建。