

Uniflash 烧写脚本的修改和 SPI 烧写的支持

在上一篇文章中，我们介绍了如何使用 Uniflash 进行 NAND 的烧写，主要列举了如何烧写 Linux 相关 image 的过程，收到了不少朋友的反馈。根据大家的反馈，我在本文中，会对如何修改烧写脚本源文件 debrick.txt 进行介绍，并对 SPI 烧写的支持进行介绍。

SPI 的烧写方法，和之前 NAND 烧写流程基本一致，主要的工作是在于 Uniflash 的烧写脚本源文件 debrick.txt 的修改。这里以 debrick-spi.txt 脚本源文件作为参考，讲解一下如何修改烧写脚本源文件。（NAND flash 源文件的修改，与此也相差无几，请重点参考下面第 4 节。）

我们首先回顾一下 Uniflash 的工作流程：当物理连接、启动模式、主机环境等一切设置 OK 后，Uniflash 烧写的第一步会将 u-boot-spl-restore.bin, u-boot-restore.img 下载到板子里，启动板子进入到 UBOOT stage；（特别注意：u-boot-spl-restore.bin, u-boot-restore.img 这两个文件制作时使用的配置选项：如果是使用网口烧写，请使用 [am335x evm restore flash](#)；如果是使用 USB 烧写，请使用 [am335x evm restore flash usbspi](#)）进入到 UBOOT 后，Uniflash 会下载 debrick.scr 脚本文件到板子（debrick.scr 文件决定了烧写方式：NAND 烧写的源文件为 debrick-nand.txt，SPI 烧写的源文件为 debrick-spi.txt，内容中还包含了要烧写文件的个数、名称信息、大小信息、以及烧写到板子 flash 的地址区域信息，DDR 地址缓存定义），板子之后的操作就是根据这个 debrick.scr 文件来进行的。下面我们以烧写 MLO.byteswap 和 u-boot.img 两个文件为例，介绍如何修改 debrick.scr 脚本的源文件 debrick.txt：

1. 使用源文件 flash_cat_util.c，生成 flash_cat_util.out 工具。

```
steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$ ls
flash_cat_util.c
steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$ sudo gcc -W flash_cat_util.c -o flash_cat_util.out
steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$ ls
flash_cat_util.c flash_cat_util.out
```

2. 将需要烧写进 spi flash 的文件考入到当前的目录，本例中为 MLO.byteswap 和 u-boot.img。并使用命令 `./flash_cat_util.out MLO.byteswap u-boot.img` 生成 flash-image-data 和 flash-image.out 文件。

```
steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$ ls
flash_cat_util.c flash_cat_util.out MLO.byteswap u-boot.img
steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$ sudo ./flash_cat_util.out MLO.byteswap u-boot.img

Opening Input File MLO.byteswap

len of MLO.byteswap is 100549
leftover 197 offset 102400

Closing Input File MLO.byteswap

Opening Input File u-boot.img

len of u-boot.img is 364152
leftover 1656 offset 466944

Closing Input File u-boot.img

File concatenation is complete

steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$ ls
flash_cat_util.c flash_cat_util.out flash-image-data flash-image.out MLO.byteswap u-boot.img
steven@WorkStation:~/workspace/EZSDK_06_00_00_00/board-support/u-boot-2013.01.01-psp06.00.00.00/fl
ash_script_test/packet$
```

3. 查看 flash-image-data 文件，这里的 Length 即为要修改到 debrick 脚本文件中的文件长度。
PS: 这里文件的大小是十进制表示的，拷贝到 debrick 脚本中需要做 16 进制的转化。

```
3 Texas Instruments Flash Image Concatenation tool - Copyright 2013
4
5
6 Image Name, offset into DDR Image will be after transfer and Length of the Image
7 Place in the Debrick text file
8
9 Image MLO.byteswap Offset 0 Length 100549
10 Image u-boot.img Offset 102400 Length 364152
```

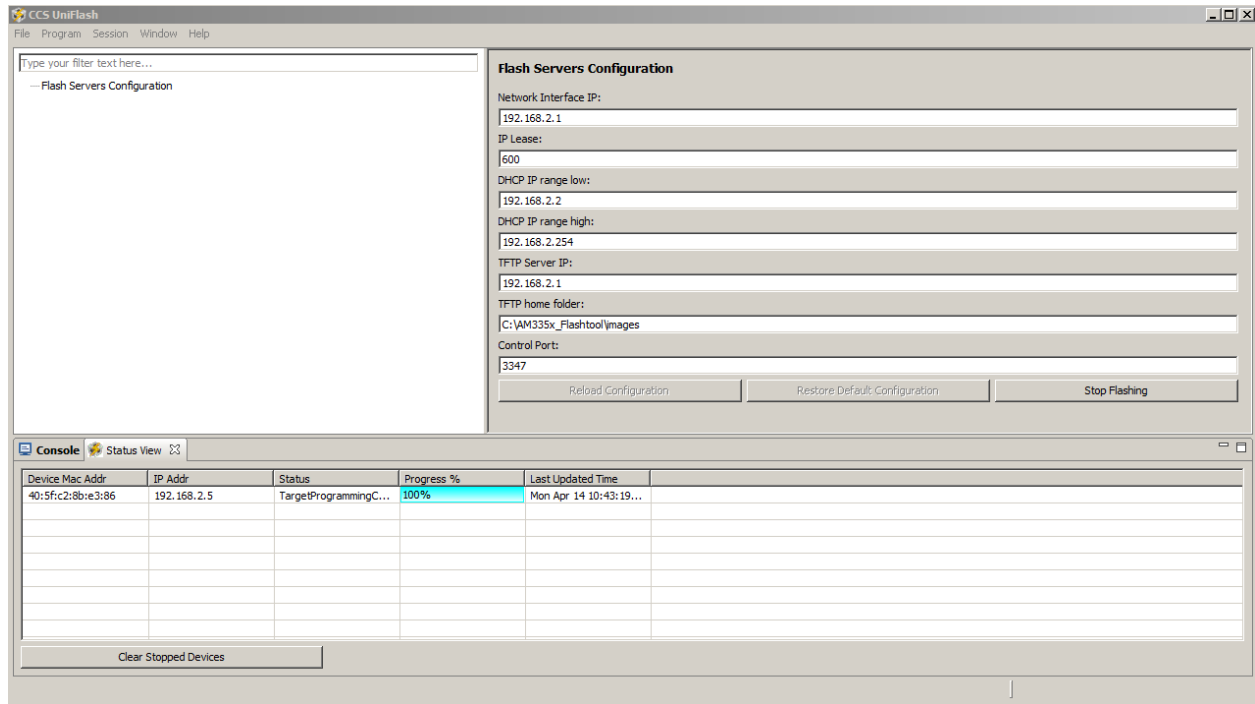
4. 修改 debrick-spi.txt 文件，示例文件中为烧写 MLO.byteswap, u-boot.img, ulmge 三个文件，我们这里仅烧写 MLO.byteswap, u-boot.img 两个文件，所以对应的修改如下：
 - a. 45 行的 IMAGE_CNT: 需要烧写的文件个数。原始为 3，本例中我们修改为 2。
 - b. 50-53 行的 Image_Name: 定义烧写文件的名字。本例中我们只需要注释掉 Image3_Name 即可。
 - c. 55-58 行的 Image_SPIFLASH_Offset: 定义烧写到 SPI flash 的地址。这里和你的 SPI flash 的 layout 相关。本例中，我们使用的默认值，只需要将 Image3_SPIFLASH_Offset 注释即可。
 - d. 63-66 行 Image_DDR_ADDR: 定义 TFTP 时 image load 的地址。首先 image 会被 load 到这里，再烧写到 SPI flash 中。这里需要注意，DDR load 的地址足够放下你的 image 即可。本例不做修改，只需要将 Image3_DDR_ADDR 注释即可。
 - e. 69-71 行 Image_Length: 这里是定义了 image 的大小，多数情况下，都是要修改的，而且这个数值是根据 flash-image-data 文件中的结果得来的。本例中，我们根据文中第 3 步中的结果，修改 Image1_Length 为 0x188C5，Image2_Length 为 0x58E78，再注释 Image3_Length 即可。
 - f. 最后，在 163-169 行，这段代码为 Image 3 的烧写代码，本例中并不需要，注释掉即可。

通过以上 a.b.c.d.e.f 的步骤，完成了 debrick-spi.flash 的修改，再使用以下命令生成 debrick.scr 即可：
`./mkimage -A arm -O U-Boot -C none -T script -d debrick-spi.txt debrick.scr`

如果你需要烧写更多的文件（比如 4 个、5 个），需要增加的步骤同上。

5. 至此，我们已完成了所有关于烧写的文件修改设置。只需要将相应的文件考入到烧写目录，即可进行 SPI flash tool 的烧写。

SPI flash 烧写成功画面:



SPI flash 烧写的 Uniflash 相关文件夹的:

