从SD卡启动系统

2018年03月06日　　下午 05:52

1,让uboot支持保存环境变量到SD卡上:

修改uboot源码中bootloader/u-boot-2009.08/include/configs/mx28_evk.h文件的第269行:

把:

```
//#define CONFIG_FSL_ENV_IN_MMC
#undef CONFIG_FSL_ENV_IN_MMC
#define CONFIG_FSL_ENV_IN_NAND
```

改为:

```
#define CONFIG_FSL_ENV_IN_MMC
//#undef CONFIG_FSL_ENV_IN_MMC
//#define CONFIG_FSL_ENV_IN_NAND
```

重新编译uboot,生成新的imx28_ivt_uboot.sb文件:

```
# make ARCH=arm CROSS_COMPILE=arm-fsl-linux-gnueabi- distclean
# make ARCH=arm CROSS_COMPILE=arm-fsl-linux-gnueabi- mx28_evk_config
# make ARCH=arm CROSS_COMPILE=arm-fsl-linux-gnueabi-
# cp -av u-boot ../imx-bootlets-src-10.12.01/
# cd ../imx-bootlets-src-10.12.01/
# make CROSS_COMPILE=arm-fsl-linux-gnueabi- BOARD=iMX28_EVK
# cp -av ./imx28_ivt_uboot.sb $(TFTPBOOT) -fr
```

2,使用cfimager.exe工具把新的imx28_ivt_uboot.sb烧写到sd卡上:

烧写脚本文件:

```
@echo off
echo.
echo EasyARM-iMX28x 制作SD卡启动程序
echo.
echo 请输入SD卡盘符:
set /p diskpath=
set cmdpath=%~dp0
echo 注意:
echo 文件会被烧写在%diskpath% 盘
echo.
%cmdpath%cfimager.exe -f %cmdpath%imx28_ivt_uboot.sb -d %diskpath%
echo 烧写完毕，按键退出
echo.
pause>nul
```

关于cfimager:

```
>cfimager.exe -h
Usage: cfimager.exe [options]
Options:
  -h, -?, -help            Show this help
  -v, -version             Display the version
  -f, -file <path>           Input firmware file
  -d, -drive <char>           Drive letter (no colon, case does not matter)
  -a, -always-format          Always reformat entire drive
  -x, -extra <int>           Extra kilobytes in firmware region
  -e, -extra-image <path>       Extra (3rd) partition image file
  -n, -no-format             Do not format the FAT32 partition
  -i, -info                Show info about the drive, do not format
  -TA3                   Format drive for TA3 ROM only
  -TA4                   Format drive for TA4+ ROM only
  -imx51                  Format driver for i.MX51 ROM only
  -imx53                  Format driver for i.MX53 ROM only
  -img, -bin               Preformatted image is provided, dump it on the device starting at block 0
  -redundant_boot, -dual_boot,
  -secondary_boot             Format the drive for secondary boot with primary and secondary image and config block
  -bcb, -BCB               Format the drive for BCB boot
  -raw                   Write Image to physical location
  -offset                                    physical location offset, must be used with -raw
```

```
 -skip                    skip how many byte of firmware image, must be used with -raw
                        typical usage for linux uboot.bin is
                          cfimager -raw -offset 0x400 -skip 0x400 -f uboot.bin -d G

The -f and -d options are required. All other options are optional. By default,
the tool will not reformat the entire drive. If it cannot place the firmware
without reformatting, it will display a message. You can then run the tool
again using the -always-format option.


The -e option used created 3rd partition with extra image file such ext3 for linuxboot image create
By default, the drive will be formatted so that it can work on both TA3 and
TA4 (and above) devices. Using the -TA3 or -TA4 switch will cause the drive
to be formatted so that it works with only that ROM version.

Based on the -imx51 option and the name of the nb0 file, the address to flash
the i.MX51 file on the card is determined automatically by cfimager

The -no-format option causes the tool to skip writing the FAT partition, but
still writes the corresponding partition entry in the MBR. This lets you use
the operating system's native formatter.
```

~~3,为SD卡重新分区:~~

把TF卡插入ubuntu电脑,使用Gparted软件给TF卡分区.我们看到TF卡里有三个分区:一个FAT分区,两个1M大小的未知分区,还有一些未用存储空间(大约7.5M).

1）删除FAT分区。不要修改移动两个未知分区，里面放置的是uboot，改动后就无法使用uboot启动。

2）删除FAT后，在未用分区上，开始分区.分区参考：

a, 从0扇区开始20M空余出来不使用，未来用于放置uboot的环境变量和uImage文件

b, 20M之后可以留100M做FAT16分区

c, 把其余的空间都可以分区成ext2格式分区, 用于存放rootfs文件系统（如果格式为ext3,ext4, 开发板会无法加载为根文件系统,原因未知）

3）保存更改.

~~4,烧写uImage文件和rootfs.tar.bz2到SD卡:~~

sd卡从0扇区开始的20M空间用于存放uboot环境变量和uImage镜像,为了给uboot环境变量保留足够的空间,我们从**第2048(0x800)扇区[即第2048*(扇区数)**512*(每个扇区512个字节)*=1048576(0x100000)字节]**开始存放uImage镜像,给环境变量预留1M的空间.命令如下:

```
# dd if=uImage of=/dev/sdb seek=2048 //seek表示偏移量
```

假设sd卡的ext2分区(rootfs分区)挂载在系统的"/media/rootfs"目录下,写入rootfs文件系统的的命令如下:

```
# tar -xjf rootfs.tar.bz2 –C /media/rootfs
```

说明:

*cfimager.exe -a -f imx28_ivt_uboot.sb **-e rootfs.ext2.img** -d %diskpath%*

使用cfimager工具烧写uboot镜像的时候,可以附加上文件系统镜像,cfimager工具会自动为我们分配合理的分区(一个FAT32分区,一个uboot分区,一个rootfs分区).

不需要且不建议再自己手动更改分区,如果自己手动分区会破坏原有的分区表信息,造成有些分区不可用.

文件系统镜像制作方法:创建一个**虚拟磁盘**,格式化,然后把文件系统解压到这个虚拟分区中,最后使用**dd**命令把这个虚拟分区刻录成文件系统分区镜像.

5,设置uboot的环境变量:

把制作好的sd卡插入开发板的sd卡插槽中,并把sd卡启动引脚短接,上电后系统会从sd卡启动.

按任意键进入uboot shell,设置环境变量以便从sd卡加载内核和文件系统,命令如下:

```
# setenv bootargs 'gpmi=g console=ttyAM0,115200n8 root=/dev/mmcblk0p4 rw rootwait rootfstype=ext2 init=/sbin/init
fec_mac= ethact mem=64M'
# setenv sdcard_boot 'mmc read 0 $(loadaddr) 800 3000;bootm'
# setenv bootcmd 'run sdcard_boot'
# saveenv
    Saving Environment to MMC...
    Writing to MMC(0)... done
```

mmc read命令用法:

Usage:

mmc read <device_no> load_addr blk_no cnt

device_no	*mmc设备号,0x00*
load_addr	*把uImage文件读取到内存中指定的地址处(十六进制,0x41600000)*
blk_no	*读取的起始扇区号(十六进制,0x800)*
Cnt	*读取的扇区数(十六进制,0x3000)*

```
MMC read: dev # 0, block # 2048,count 12288 ...
12288 blocks read: OK
## Booting kernel from Legacy Image at 41600000 ...
   Image Name:   Linux-2.6.35.3-571-gcca29a0
   Image Type:   ARM Linux Kernel Image (uncompressed)
   Data Size:    2365944 Bytes =  2.3 MB
   Load Address: 40008000
   Entry Point:  40008000
   Verifying Checksum ... OK
   Loading Kernel Image ... OK
OK

Starting kernel ...
```

http://maker.zlgmcu.com/forum.php?mod=viewthread&tid=394977&extra=page%3D2

SD_TF卡启动系统Eas...

EasyArm_IMX280A ...

EasyArm_IMX280A ...

easyarm_sdboot