

闪存设备在嵌入式 Linux 系统中的应用

李胜广, 张其善

(北京航空航天大学电子信息工程学院, 北京 100083)

摘要: 论述了多种闪存设备在嵌入式 Linux 系统应用过程中的关键问题的解决方法。分析了内核对不同闪存的支持和相关代码, 给出了文件系统的设计和使用方法。对嵌入式系统存储设备的设计具有借鉴意义。

关键词: 闪存; CF; DOM; DOC; 多媒体卡

Application of Flash Memory Devices in Embedded Linux System

LI Shengguang, ZHANG Qishan

(School of Electronic and Information Engineering, Beijing University of Aeronautics and Astronautics, Beijing 100083)

【Abstract】 This paper gives the solutions to pivotal problems of FLASH memory devices in embedded Linux system. Then it analyzes the support of Linux kernel to FLASH memory devices and programming codes, and gives the design and the application of file system. This scheme is a useful reference for the designs of storages of embedded system.

【Key words】 Flash memory; Compact flash(CF); Disk on module(DOM); Disk on chip(DOC); Multimedia card(MMC)

在以计算机、通信、消费电子 3C 技术合一的趋势下, 计算机工业的中心将从计算机产品转移到嵌入式信息产品。嵌入式系统的应用要求和成本因素决定了其软硬件资源都是精简高效的。存储设备一般不采用硬盘等高功耗、大体积设备, 而使用 Flash Memory 为存储载体的各种移动闪存设备。闪存具有功耗低、可靠性高、擦写方便等特点, 目前流行的闪存主要有 Flash 芯片、CF(Compact Flash)卡, DOM(Disk on Module)电子盘、DOC(Disk on Chip)电子盘、SD(Secure Digital)卡、MMC(Multimedia Card)卡等。闪存已经广泛地应用到防火墙、PDA、机顶盒、GPS 接收机等多种嵌入式设备中。

Linux 操作系统具有开放的源码、较小的系统内核等优点, 在嵌入式系统中得到广泛应用。本文对闪存在使用过程中需要解决的关键性软硬件问题进行了研究和总结。

1 使用 Flash 芯片

NOR 和 NAND 是现在主要的两种 FLASH 技术, NOR 和 NAND 之间在读写速度、接口、使用寿命等方面存在差别, 总体而言 NAND FLASH 适用于大量数据的存储, 而 NOR FLASH 主要用于高速读写要求的场合。

嵌入式 Linux 使用文件系统屏蔽了用户对闪存的使用, 实际是文件系统通过底层 MTD(memory technology device)驱动程序对 FLASH 进行访问。MTD 设备提供了抽象的接口层, MTD 层定义了一个抽象的 MTD 设备结构, 其中包含了存储设备的共同属性定义和操作函数定义。CFI 接口的 MTD 设备可分为 4 层: 设备节点, MTD 设备层, MTD 原始设备层和硬件驱动层。

(1)Flash 硬件驱动层: 硬件驱动层负责在 init 时驱动 FLASH 硬件, Linux MTD 设备的 NOR FLASH 芯片驱动遵循 CFI 接口标准, 其驱动程序位于 drivers/mtd/chips 子目录下。NAND 型 FLASH 的驱动程序则位于 drivers/mtd/nand 子目录下。

(2)MTD 原始设备: 原始设备层有两部分组成, 一部分是 MTD 原始设备的通用代码, 另一部分是各个特定的 Flash 的数据, 例如分区。

用于描述 MTD 原始设备的数据结构是 mtd_info, 这其中定义了大量的关于 MTD 的数据和操作函数。mtd_table(mtdcore.c)则是所有 MTD 原始设备的列表, mtd_part(mtd_part.c)是用于表示 MTD 原始设备分区的数据结构。

(3)MTD 设备层: 基于 MTD 原始设备, Linux 系统可定义 MTD 的块设备和字符设备。MTD 字符设备的定义在 mtdchar.c 中实现。MTD 块设备则是定义了一个描述 MTD 块设备的结构 mtdblk_dev, 并声明了一个名为 mtdblks 的指针数组, 该数组中的每一个 mtdblk_dev 和 mtd_table 中的每一个 mtd_info 相对应。

(4)设备节点: 通过 mknod 在 /dev 子目录下建立 MTD 字符设备节点和 MTD 块设备节点。

(5)根文件系统: 在 Bootloader 中将 JFFS(或 JFFS2)的文件系统映像 jffs.image(或 jffs2.img)烧到 FLASH 的某分区中, 在 /arch/arm/mach-*/arch.c 文件的 *_fixup 函数中将该分区作为根文件系统挂载。

(6)文件系统: 内核启动后, 通过 mount 命令可将 FLASH 中的其余分区作为文件系统挂载到挂载点上(见图 1)。

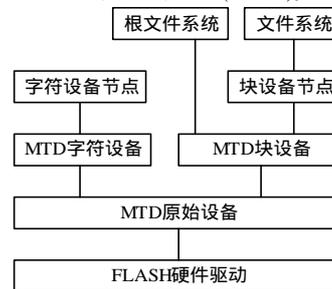


图 1 MTD 子系统层次

在 Linux 内核编译时, 配置内核选项:

在 Memory Technology Devices(MTD)选项, 设置 Memory Technology Devices support。该选项选择使用通用 MTD 驱动程序和 MTD 设备层。子项有:

作者简介: 李胜广(1977 -), 男, 博士生, 主研方向: 信息技术, 信息安全, 嵌入式系统; 张其善, 教授、博导

收稿日期: 2006-01-27 **E-mail:** lishengg@ee.buaa.edu.cn

```
<*> MTD partitioning support //对分区的支持
<*> Direct char device access to MTD devices //提供原始字符访问
<*> Caching block device access to MTD devices //提供原始块访问
进入 NAND FLASH Device drivers 子项, 加上<*> NAND Device support
```

对于原始的 Flash 芯片, 在嵌入式系统中可用 Uboot/vivi/Redboot 等 bootloader 进行分区和写入内核和文件系统。

2 使用 CF 卡和 DOM

20 世纪 90 年代初, SanDisk 和 KODAK、CASIO、Canon 制定了基于 RAM 和 ROM 技术的固态非易失 Compact FLASH 标准。CF 卡具有下面诸多特点: CF 几乎零热量; CF 卡比以往磁盘驱动器牢固耐用 5~10 倍多; CF 卡可反复擦写 100 万次, 数据至少可保存 100 年。

CF 卡分为 3 种寻址访问模式, 每一种模式在系统上反映为如何访问其地址空间。通过对 CF 卡的属性存储空间进行配置, 可得到不同的寻址访问模式, 前两种模式符合 PCMCIA PC 卡标准。CF 卡模式分类见图 2。

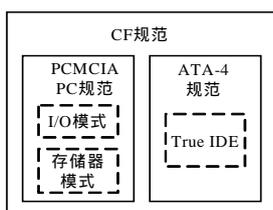


图 2 CF 卡模式分类

(1) PC IO 模式: 通过访问 CF 卡的内部 IO 寄存器所指 16B 数据块, 实现对 CF 卡的通用存储空间进行寻址访问, 一般利用系统提供的中断进行读写事务处理。

(2) PC 存储器模式: 将 CPU 和 CF 卡的地址线 and 数据线进行直连, 通过访问其通用存储空间的 16B 数据块来实现对 CF 卡的寻址访问。

(3) True IDE 模式: 该模式操作时序由 ATA-4 规范转换而来, 将标准 PC-ATA 磁盘 I/O 寻址空间 1F0h~1F7h、3F6h~3F7h 或者 170h~177h、376h~377h 作为寻址端口, 同时使用中断 IRQ14 进行事务处理。

True IDE 模式与 IDE 硬盘兼容。CF 卡通过 OE 上的电平决定其是否是 True IDE 模式还是其它模式。设置成 True IDE 模式后, PCMCIA 协议和配置被禁止, 具体指令可参考 ATA 指令集。DOM 的接口是 IDE, 可直接插在主板的 IDE 接口上, 操作与普通硬盘类似。DiskOnModule 电子盘有 40 针 IDE 接口和含电源的 44 针 IDE 接口两种方式。

对于 DOM 和 CF 卡的使用与普通硬盘相同, 编译内核时不用加入 MTD 支持, 只要包含 ATA/IDE/MFM/RLL support 的支持即可, 这与驱动 IDE 硬盘没有特殊区别。

3 使用 DOC 设备(DiskOnChip2000)

以色列 M-system 公司推出的 DiskOnChip2000 系列闪存为 32DIP 封装, 和标准的 E²PROM 完全兼容, 可直接引导系统。控制器和存储器封装在同一晶片上而不需要任何总线、插槽或者连接器。DiskOnChip2000 结构主要由系统接口单元、引导模块、闪存控制单元以及 FLASH 存储介质等组成(见图 3)。

2.4 以上版本的 Linux 对 DOC 支持都比较好, 编译内核时, 选择 Memory Technology Device(MTD) support 支持, 子项中选中<*>NFTL(NAND FLASH Translation Layer)。然后进入其中的 Self-contained MTD devices drivers, 选中与使用的 M-system 设备相同的项。最好将这几项都加入内核中, 便于从 DOC 上启动系统。

建立设备节点文件, 根据 nftl 的源码得知设备节点号为 93:

```
mknod /dev/msys/fla b 93 0
```

普通的 Grub 和 Lilo 无法让 DOC 上的 Linux 内核启动, 可从 M-System 的官方网站下载 lilo-22.4.1.tar.gz, 进入目录“lilo/lilo-bin.22/lilo-bin”, 将 doc.b 拷贝到 DOC 的 /mnt/doc/boot/目录下, 把 doc-lilo 拷贝到“/sbin/”下, 修改一下 DOC 中“/mnt/doc/etc/”的“lilo.conf”。然后运行 doc-lilo -r /mnt/doc, 设置 BIOS 使得 DOC 可作为启动盘。根据其说明修改 lilo.conf 即可。

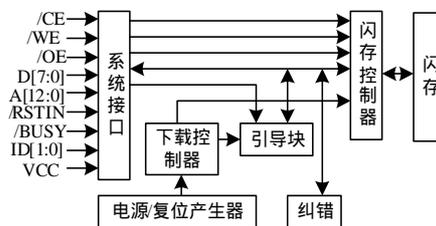


图 3 DiskOnChip 内部结构

4 使用 MMC 卡和 SD 卡

MMC 卡由西门子公司和 SanDisk 于 1997 年推出, 尺寸如 SIM 卡, 重量只有 1.5g。MMC 也是把存储单元和控制器一同做到卡上, 智能的控制器使得 MMC 保证兼容性和灵活性。SD 卡是由日本松下、东芝和美国 SanDisk 公司共同开发的, 具有大容量、高性能, 尤其是安全等多种特点的多功能存储卡。它比 MMC 卡多了一个进行数据著作权保护的暗号认证功能。

MMC 和 SD 存储卡的接口是兼容的。SD 接口除了保留 MMC 的 7 针外, 还在两边加多了 2 针数据线。采用了 NAND 型 FLASH Memory, 平均数据传输率能达到 2Mbps。

MMC/SD 卡的标准定义了 2 种命令: ACMD 和 GCMD, 标准中详细地定义了所支持的所有命令。所有的 ACMD 命令在发送前必须发送 CMD55 命令, 以标志后面的命令是 ACMD 命令。普通命令 GCMD 则可直接发送。

对于其它的驱动程序编写要根据硬件电路的设计, 如果使用 USB 口的 MMC/SD 读卡器, 则使用非常简单, 可参见下一部分。如果用处理器的 IO 口直连 MMC/SD 卡的管脚, 2.4 版本的 Linux 下则使用块设备驱动编程方法, 根据其时序对其进行驱动的编程。Linux 内核版本 2.6 里面已经实现了 MMC/SD 卡的驱动, 在 Linux 2.6 下通过下面方式对其挂载使用:

```
mknod /dev/mmcblk0 b 254 1
mount -t vfat -o sync /dev/mmcblk0p1 /mnt/mmc
```

5 使用 USB 设备

Linux 不同于 Windows, 它把所有的分区都当成设备对待, 使用前挂载, 使用后卸载, USB 移动存储器(移动硬盘/U 盘)、USB 口的读卡器也不例外。不同的 Linux 系统对 USB 设备的使用方法是不同的, 通常设备名一般为 SDA*等。在 RedHat 上面使用 USB 移动存储器步骤:

(1) 插上 USB 设备。

(2) 确保载入相应的驱动:

```
modprobe usb-storage
```

usbcore.o 和 usb-uhci.o 或 usb-ohci 也是不可缺少的。如果系统中 SCSI 支持也是模块方式, 则加载 scsi_mod 和 sd_mod 模块。

(3) 挂载 USB 移动存储器:

```
mount -t vfat /dev/sda1 /mnt/usb
```

(4) 使用结束后卸载, umount /mnt/usb。

6 闪存设备文件系统

Flash 的寿命以擦除次数表示,一般是 100 万次。闪存设备的某块区域一旦超过使用寿命而损坏,即便其它区域是好的也将无法使用。对于原始 Flash 芯片,常用的文件系统为 cramfs+jffs/jffs2/yaffs/yaffs2+ramfs。而对于 CF 卡、DOC 或 DOM,则用 ext2/jffs2+ ramfs。选择合适的文件系统对于提高它们的寿命是非常关键的。

下面以 CF 卡为例,说明文件系统的安排。由于/var 目录存放系统经常变化的文件比如日志等,因此不能将 var 里面的文件放在 CF 卡中。内存文件系统是在 Linux 中将一部分内存 mount 作分区来使用,通常称之为 RamDisk。将内存文件系统挂载到 var 目录下,里面的文件操作都是在内存中完成。

第 1 种就是传统意义上的 RamDisk,可格式化,然后加载。其不足之处是大小固定。不建议使用。

在内核 2.4 下可用更简单的方法实现,即内存文件系统(Ramfs)。Ramfs 处于虚拟文件系统层,而不像 RamDisk 那样基于虚拟在内存中的其它文件系统。无须格式化,只要内存足够可创建多个,并在创建时可指定其最大能用的内存大小。

如果 Linux 已经将 Ramfs 编译进内核,则很容易使用 Ramfs。创建一个目录,加载 Ramfs 到该目录即可:

```
# mkdir /RamDk
# mount -t ramfs none / RamDk
```

缺省情况下,Ramfs 被限制最多可使用内存大小的一半。可通过 maxsize 选项来改变。

由于 init 进程根据 inittab 里面的设置首先执行 rc.sysinit 脚本,并且该脚本里面执行别的脚本,需要在/var 目录下记录日志或者产生锁定文件。因此必须在使用 var 前挂载该目录,不然就会出现系统错误。在合适的地方修改/etc/rc.d/rc.sysinit 文件,添加如下语句:

(上接第 190 页)

应用软件的流程如图 4 所示。软件首先完成系统的初始化,并根据需要设置相应的视频记录参数,主要参数如表 2 所示,这些参数决定了最终记录的视频数据的格式。软件还必须使 W99200F 通过 I2C 总线对视频解码器 SAA7113 进行配置。

表 2 主要寄存器定义

寄存器	注 释
Video_format	选取制式:NTSC、PAL
Vframe_pattern	选取“m”和“n”的值
Vbit_rate_m, Vbit_rate_l	选取比特率
Vbv_size	选取 VBV 大小
Vbv_initial	选取 VBV 初始化的充满度
Vwork_mode	比特[4:0]=0x0
Vgop_header	选取 GOP 头和序列头频率

初始化结束后,利用Linux系统提供的多线程机制,软件同时启动 6 路视频压缩卡进行工作,系统在运行时同时启动了 10 个线程,其中每路压缩卡的记录各开一个单独的线程,保证了它们能够实时的读取FIFO中的数据;音频数据的记录用另外一个线程;还有另外的两个线程分别用来根据切换标志周期性地创建文件夹和监听串口传来的指令信息并进行相应的处理。线程之间的通信与同步采用了POSIX^[5]提供的信号量(Semaphores)和互斥量(Mutex)机制。

系统软件实现了与外界通过串口进行有效的通信,以获取射击信号及事件标志,在所记录的文件中予以标记。最后,

```
mount -t ramfs none /var
mkdir -p /var/log #建立必要目录
...
```

这样系统启动后/var 目录下的文件均存在于内存中,减少了系统对 CF 卡的频繁读写,提高了使用寿命。

对于普通 Flash 芯片,利用工具软件将文件系统先做成 cramfs/jffs2/yaffs2 映像文件,烧入 Flash 芯片中,挂载 ramfs 系统到 tmp 目录,进行相应的暂存,对实际要保存的文件写入 jffs2/yaffs2 文件格式的系统中。

7 结束语

本文主要介绍了多种闪存设备的结构和特点、着重说明它们如何在嵌入式 Linux 系统中被内核支持和文件系统的设计。嵌入式系统在工业控制、信息产品中的不断应用,闪存设备的适用范围也将越来越广。

参考文献

- 1 Bovet D P, Cesati M. Understanding the Linux Kernel[M]. O'Reilly, 2002.
- 2 Embedding Redhat Linux in a DiskOnChip HOWTO[EB/OL]. 2005. <http://www.prosig.com/protor/kbase/DiskOnChipDevelopment-HOWTO.pdf>.
- 3 Intel Corporation. Understanding the Flash Translation Layer (FTL) Specification[Z]. 1998.
- 4 CompactFlash Specification[Z]. 2005. http://www.compactFlash.org/cfspc3_0.pdf.
- 5 Brake C, Sutherland J. Flash Filesystems for Embedded Linux Systems[J/OL]. 2005. <http://www.linuxjournal.com/article/4678>.
- 6 Malik V. The Linux MTD, JFFS HOWTO[EB/OL]. 2005. <http://ftp.linux.org.uk/pub/people/dwmw2/mtd/cvs/mtd/mtd-jffs-HOWTO.txt>.

系统从串口接收关机信号,由主线程停止各个记录线程和监听线程,然后自己退出线程,完成视频记录工作。

3 结论

本机载视频记录系统基于龙芯 2 号处理器 PC104 Plus 模块构建,采用移植后的 Linux 操作系统,通过 W99200F 进行实时视频压缩,实现了对 6 路视频和 1 路音频的实时记录,所有记录数据均符合多媒体技术的相关标准,同时由于采用了国产高性能 CPU 和开放源代码的操作系统,也实现了高安全性和高可靠性。本系统成功完成了各项功能指标验证,准备进行环境实验,以便装配于某型主战飞机。它同时也为我军机载设备采用国产 CPU 芯片提供了一个先例。

参考文献

- 1 PC104 Plus Specification Version 2.0[EB/OL]. 2003-11. <http://www.pc104.org>.
- 2 中国科学院计算技术研究所. 龙芯 2 号用户手册[R]. 2004.
- 3 Winbond Electronics Corp. W99200F MPEG1 Video Encoder Data Sheet[EB/OL]. 1999. <http://www.winbond.com.tw>.
- 4 高翔. Godson 主板 BIOS 的开发[D]. 合肥:中国科学技术大学, 2002.
- 5 张立冬,程晓宇. 某型机载数字音视频记录系统的软件设计与开发[J]. 计算机工程, 2005, 31(6): 213-215.
- 6 Matthew N, Stones R. Linux 程序设计[M]. 北京:机械工业出版社, 2002.