第20卷 第12期
2013年12月

电 光 与 控 制
Electronics Optics & Control

Vol. 20 No. 12
Dec. 2013

doi:10.3969/j.issn.1671-637X.2013.12.019

基于 VxWorks 的 FLASH 之文件系统 TFFS 分析

石改辉， 武 静， 李 兵
（中国航空工业集团公司洛阳电光设备研究所，河南 洛阳 471009）

摘 要：FLASH 存储器在工程应用中非常广泛，已成为嵌入式系统的重要组成部分。在 FLASH 上开发文件系统能方便用户对 FLASH 进行拷贝、删除以及创建等操作。详细介绍了 VxWorks 操作系统下 FLASH TureFFS 的架构，以 W78M32VP 芯片为例，以实际工程中遇到的文件系统无法格式化问题为契机，讨论如何在 FLASH 上实现文件系统。

关键词：VxWorks；FLASH；W78M32VP；TFFS
中图分类号：V247.1 **文献标志码**：A **文章编号**：1671-637X(2013)12-0080-04

Analysis of FLASH TFFS Based on VxWorks

SHI Gaihui, WU Jing, LI Bing
(Luoyang Institute of Electro-Optical Equipment, AVIC, Luoyang 471009, China)

Abstract: The FLASH memorizer is an important part of the embedded systems, which has found wide application in engineering. It is very convenient for the users to use the FLASH memorizer by developing the FLASH file system, such as copying, deleting and creating. The TFFS configuration of FLASH under VxWorks operation system is discussed in detail, and how to realize the file system on FLASH memorizer is presented by taking W78M32VP chip as an example for working out the practical problem that the file system can be not formatted.

Key words: VxWorks; FLASH; W78M32VP; TFFS

0 引言

VxWorks 是 Wind River System 开发的一种嵌入式实时操作系统[1]，在实际型号研发中得到了广泛的应用。FLASH 存储器具有容量大、可靠性高、抗震动、低功耗以及掉电非易失性等特点[2-3]，广泛用于工程领域，但它的使用寿命有限[4]，为了延长闪存的使用寿命，行之有效的办法是在 FLASH 上实现 TFFS。TFFS 文件系统是为 VxWorks 操作系统提供的定制实现，在 FLASH 上建立 TFFS 文件系统，既能延长 FLASH 的使用寿命，提高使用效率，又能使开发者像使用 M-DOS 文件一样方便地进行拷贝、删除、创建等操作，使基于 FLASH 的软件开发更加便捷。

本文以实际工程应用中，W78M32V FLASH 芯片架构发生变化导致文件系统无法格式化为研究契机，详细描述了 TFFS 文件架构，对 FLASH 上实现 TFFS 进行了深入的研究，并解决了实际工程中的问题。

收稿日期：2013-01-04 **修回日期**：2013-01-31

作者简介：石改辉(1981—)，女，河南洛阳人，硕士，工程师，研究方向为嵌入式系统研究。

1 TFFS 文件系统架构

TFFS(True Flash File System)是 M-systems 公司推出的一种文件系统[5]，它需要在 TFFS 上加载 DOS 文件系统才能使用。TFFS 屏蔽了存储介质底层的硬件差异，为上层应用提供了统一的接口，使得应用程序可以方便地对 FLASH 进行操作。TFFS 架构如图 1 所示。

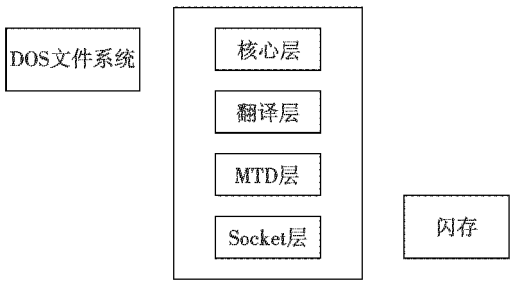

图 1 TFFS 分层结构

Fig. 1 Hiberarchy of TFFS

TFFS 组成包括：翻译层、MTD 层(Memory Technoilogy Driver)、Socket 层、核心层[6]。

1）翻译层具有 TFFS 和 DOS 文件系统之间的高级交互功能。常见的有 3 种不同的翻译层模块，选择哪一种模块要由所使用的 FLASH 技术类型决定。目前

有 3 种 FLASH 技术：NOR 技术、NAND 技术和 SSFDC 技术。

2）MTD 层主要包含对具体的 FLASH 进行读、写、擦、ID 识别、映射等驱动，并设置了一些与 FLASH 密切相关的参数。

3）Socket 层提供了 TFFS 和硬件之间的接口服务。

4）核心层主要功能是将其他 3 层有机结合起来，处理一些全局问题。

目前，VxWorks 提供的核心层和翻译层驱动是经过编译后的二进制格式，只需直接应用，而不能修改。要实现基于某个特定型号 FLASH 的 TFFS 功能，只需对 MTD 层和 Socket 层的模块进行修改或者重新编写。

2　TFFS 的工作原理

1）损耗均衡（wear-leveling）。

为了延长闪存的使用寿命，TFFS 采用了一种损耗均衡策略[6]。该方法的原理是平衡使用所有的存储块，避免对某一块过度使用，从而延长闪存的使用寿命。

2）碎片回收。

在 FLASH 的实际使用中会产生许多碎片，如果不对这些碎片进行回收，FLASH 很快就会变成只读状态。TFFS 使用碎片回收机制来解决这个问题，通过拷贝有效数据到另一个擦出单元，然后更新映射表，从而实现碎片回收。

3）块分配和关联数据集结。

为了提高闪存的读写速率，TFFS 将关联的数据集中存放。尽量在同一个擦除单元内维持一个由多个物理上连续的自由块组成的存储池，从而加快了文件访问速度，并减少碎片的产生。

4）错误恢复。

为了解决写 FLASH 失败导致数据丢失的问题，TFFS 采取了一种“先写后擦”的策略。只有在写操作完成并且新数据校验成功后，新扇区的数据才有效，否则老扇区的数据有效，这样就能保证已经写到 FLASH 上的数据的稳定性。

5）TFFS 和引导映像共享 FLASH。

VxWorks 允许 TFFS 和引导映像共享同一块 FLASH。在同一块 FLASH 上申请两个不同的空间，实现了既能安全存放引导映像，又能对 FLASH 进行任意的读、写、擦等操作。

3　TFFS 的设计与实现

FLASH 上实现基于 VxWorks 的 TFFS，设计流程如图 2 所示。

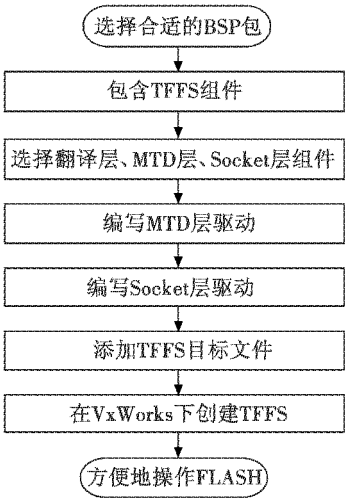

图 2　TFFS 设计流程

Fig. 2　Flow chart of TFFS design

以 FLASH W78M32V 为例，在一块可移动 FLASH 上实现 TFFS。FLASH 大小为 256 MB，前边 12 MB 用作 Bootrom 和 VxWroks 映像区域，剩余地址空间用作 TFFS。开发工具为 PC 机，开发环境采用 Tornado2.2。

3.1　在 VxWorks 下配置 TFFS

1）要在 VxWorks 下实现 TFFS，就必须在 VxWorks 下包含 TFFS 组件，首先要在 config.h 文件中进行一些宏定义[7]，这样才能在 Tornado 下初始化 TFFS。

2）选择翻译层模块、MTD 层模块以及 Socket 层模块。

WindRiver 为翻译层提供了基于 NAND 和 NOR 的 FLASH 技术，根据 FLASH 技术类型选择合适的翻译层模块，W78M32V 是基于 NOR 技术的 FLASH。

VxWroks 自带了一些 MTD 层组件模块，包括基于 AMD、三星等品牌 FLASH 的 MTD 模块，从中选择合适的模块，必要时定义新的 MTD 层模块。

默认情况下，BSP 目录里没有 sysTffs.c 这个文件，可以从 Tornado 安装目录..\target\src\drv\tffs\sockets 下拷贝一个 sysTffs.c 文件到自己的 BSP 目录下，在这个文件里格式化 TFFS 函数 sysTffsFormat()，当 TFFS 挂接到一个 FLASH 设备上以后，就需要用这个函数来格式化 TFFS 空间。

3）编写 MTD 层驱动函数。

默认情况下，BSP 目录里没有 MTD 组件驱动，可以从 Tornado 安装目录..\target\src\drv\tffs 下拷贝一个接近的 MTD 组件驱动文件到自己的 BSP 目录下，必要时，重新编写 MTD 层组件驱动。该文件定义了 TFFS 每个扇区的大小/访问宽度，同时也定义了 MTD 识别函数、执行读、写、擦除函数所需的指令及函数原

型,这些驱动函数是实际工作中对 TFFS 进行磁盘操作调用的最底层函数。对扇区大小的定义非常重要,TFFS 在 MTD 层的所有驱动函数执行操作的最小单元是扇区,这里定义的扇区大小可以是 FLASH 实际扇区大小的 n 次幂($n=1,2,4,8,\cdots$),最小和实际扇区大小一样[8],否则,MTD 层驱动将无法正常工作,也就无法对 TFFS 进行格式化。

4) 编写 Socket 层驱动函数。

有关 Socket 组件的驱动位于 sysTffs.c 文件里,一般情况下,这些函数都满足需要,不需要进行修改。

5) 添加 TFFS 的目标文件。

在 makefile 文件里添加目标文件。

6) 为 VxWorks 创建 TFFS。

在对 TFFS 的几个架构层分别进行配置之后,就可以为 VxWorks 创建 TFFS。

首先调用 trrsDrv( )函数,为 VxWorks 初始化 TFFS,为目标机上所有的 FLASH 设备注册 Socket 组件的驱动程序。

其次调用 tffsDiskInit( )函数,该函数为 VxWorks 在 FLASH 上配置一个 TFFS,并给 TFFS 设备取名,在这里取名为"/tffs0"。

最后调用 tffsDrvFormat( )函数,格式化 FLASH 上的 TFFS 区域供使用。格式化完成后,返回 0 表示格式化成功;返回 -1 表示格式化失败。

格式化成功之后,调用 usrTffsConfig( ),创建 TFFS 块设备,并把 DOS 文件系统挂接到 TFFS 上,至此,TFFS 配置完成。在串口下,能看到一个名为"/tffs0"的设备,而且可以像操作其他磁盘一样对此设备进行创建、拷贝、删除等。

3.2 实例

在实际项目中,出现过 CPU 板采用了同一型号新批次的 FLASH 芯片(型号为 W78M32VP),文件系统无法格式化,而老批次 FLASH 可以正常格式化。参照上文 TFFS 开发过程,为解决这个问题,首先对比芯片资料,新批次 FLASH 和老批次 FLASH 同为 NOR FLASH,翻译层驱动没有问题,对比扇区架构,老批次 FLASH 的扇区架构为[9]:

- Bank A (16 Mb):4 Kw×8 and 32 Kw×31
- Bank B (48 Mb):32 Kw×96
- Bank C (48 Mb):32 Kw×96
- Bank D (16 Mb):4 Kw×8 and 3 Kw×31

新批次 FLASH 扇区架构为 64 Kw×128[10],查看 MTD 层驱动,定义的 TFFS 扇区操作的单位是是 32 Kw,由于在 TFFS 开发过程中,驱动函数操作的最小单元是一个扇区,如果这个扇区定义小于 FLASH 芯片手册上规定的物理扇区大小,将导致驱动函数执行失败,修改 MTD 层驱动 S29GLMtd.c 中对 TFFS 操作扇区大小的宏定义[11],修改如下:

```
#if
#define S29GL_MTD_SECTOR_SIZE   0x00008000   /*32K*/
#else
#define S29GL_MTD_SECTOR_SIZE   0x00010000   /*64K*/
#endif
```

重新编译操作系统,文件系统格式化成功,如图 3 所示。

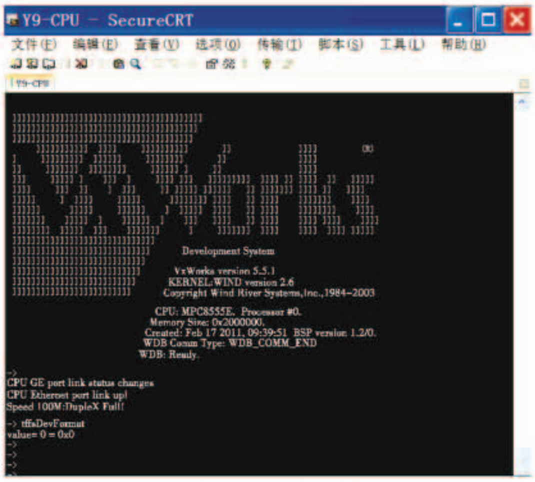

图 3 文件系统格式化界面

Fig. 3 Formatting interface of file system

3.3 性能分析

实现 TFFS 可以延长 FLASH 的使用寿命,以 W78M32VP 为例,可以擦除 10 万次。如果要写入 100 个小于 64 KB 的文件,FAT 表至少要更新 100 次,而 FAT 表处于某一个固定的扇区内,该扇区被擦除 100 次,1000 天这样的操作就会使该扇区被擦除 10 万次,达到了使用寿命的极限;如果每天写 2 次这样的文件,只用 500 天(约 1.5 年)就会使 FLASH 器件报废。如果在该器件上实现了 TFFS 文件系统,损耗均衡算法不允许 FAT 表位于固定的扇区内,该算法使每个扇区得到均衡使用,这样,FLASH 的使用寿命为

$$\text{期望寿命}=(\text{容量}*\text{总擦除次数}*0.7)/\text{每天写入字节数} \quad (1)$$

式中,0.7 表示文件系统和 TFFS 管理结构的额外消耗系数,该 FLASH 的使用寿命为 $(32\ \mathrm{M}*100000*0.7)/(64\ \mathrm{K}*2)=17920000$,(约为 49095 年),大大延长了 FLASH 的使用寿命。

4 结语

本文对 VxWorks 下 TFFS 的架构以及工作原理进行了详细介绍,并在 W78M32VP FLASH 上实现了 TFFS,解决了实际工程中 W78M32VP TFFS 格式化失败导致文件系统无法使用的问题,并在多个项目上得

到验证，在实际工程中得到了成功的应用。

参 考 文 献

[1] 王仁勇，愈建新. 基于 VxWorks 的 TrueFFS 分析与实现[J]. 计算机工程，2007，12：68-70.

[2] BEZ R, CAMERLENGHI E, MODELLI A, et al. Introduction to flash memory [C]//Proceedings of the IEEE, 2003, 91：489-502.

[3] CORP T. NAND vs NOR flash memory technology overview [EB/OL]. (2006-4-25)[2013-01-05]. http://www.toshiba.com/taec/components/Generic/Memory.Resources/NAND vs NOR.pdf.

[4] Samsung Company. Electronics S[EB/OL]. http://www.samsung.com/global/business/sem iconductor/products/Flash/Products.NAND Flash html.

[5] 朱汉德，梁杰申. 通过 TFFS 加载和升级 VxWorks 映像[J]. 微计算机信息，2008(14)：36-38.

[6] Technical Note. TureFFS wear-leveling mechanism[EB/OL]. (2002-05) http://m-sys.com/Dmitry Shmidt.

[7] 薛原. 基于 VxWorks 的文件系统的研究与实现[J]. 电子设计工程，2009(8)：107-109.

[8] WOODHOUSE D. JFFS：The journalling flash file system [EB/OL]. [2001-10-10]. http://sources.rdhat.com/Jffs2/jffs1.pdf.

[9] White Company. White electronic designs W78M32V-XBX. [EB/OL]. [2013-01-05]. http://www.whiteedc.com.

[10] White Company. White electronic designs W78M32VP-XBX[EB/OL]. [2013-01-05]. http://www.whiteedc.com.

[11] STMicroelectronics. UM0116 STR7 family flash programming user manual Rev4 [EB/OL]. (2006-09-20). http://www.st.com/stonline/books/pdf/docs/11130.pdf.

(上接第 13 页)

[11] 王景奇，范奎武，张最良. 机动目标对搜索的最优规避[J]. 军事系统工程，2001，11(1)：4-9.

[12] 汤智胤，何琳. 基于直升机声纳探测的潜艇被发现概率仿真[J]. 系统仿真学报，2008，20(17)：4751-4755.

[13] 屈也频，廖英. 潜艇位置散布规律与搜潜效能评估模型研究[J]. 系统仿真学报，2008，20(12)：3280-3283.

[14] PATRICIA A T. Some priorities for a target probability area ADA085057[R]. 2001.

[15] 盛骤，谢式千，潘承毅. 概率论与数理统计[M]. 北京：高等教育出版社，2001：56-62.

[16] 陆光宇，董志荣，惠小霞. 被动浮标目标运动分析及其仿真计算[J]. 指挥控制与仿真，2006，28(6)：31-34.

[17] 徐钟济. 蒙特卡罗法[M]. 上海：上海科学技术出版社，1985：24-55.

下　期　要　目

多子阵平板天线峰值旁瓣优化
空空导弹武器系统作战效能评估系统设计
车载平视显示技术
空空导弹用于舰载反导拦截的导引策略
多无人机时序到达协同控制方法
面向不确定性环境的多无人机协同防碰撞
节点崩溃条件下信息系统安全风险传播
对地观测相机像移速度矢量建模
直升机侧翼护航倾斜三角形搜潜方法及其仿真
基于 Snake 模型的波门自适应跟踪算法