周立功单片机公司策划

ARM嵌入式系统软件开发实例

（一）

周立功　等编著

北京航空航天大学出版社

内容简介

本书详细介绍当前几大热点 ARM 嵌入式系统软件模块的原理及其在 AM7 上的实现。分为 5 章，每章介绍一种模块。第 1 章介绍 FAT 文件系统的基础知识，以及兼容 FAT12、FAT16 和 FAT32 的文件系统模块 ZLG/FS 的源码分析。第 2 章介绍 USB 模块驱动程序的设计思想及实现过程。第 3 章详细介绍 CF 卡和 IDE 硬盘及相应软件模块 ZLG/CF 的设计思想及实现过程。第 4 章详细介绍 TCP/IP 及相应软件模块 ZLG/IP 的设计思想及实现过程。第 5 章介绍 GUI 的基础知识及 GUI 模块 ZLG/GUI 的设计思想和实现过程。

这些模块是在 PHILIPS 公司的通用 ARM7 微控制器 LPC2200 系列上调式通过的，可以很容易地移植到基于其他处理器核的嵌入式系统上。

本书可作为《ARM 嵌入式系统系列教程》的配套参考资料，可用作高等院校相关专业的 ARM 嵌入式系统课程的参考书，也可作为从事 ARM 嵌入式系统开发应用工程技术人员的参考资料。

图书在版编目(CIP)数据

ARM 嵌入式系统软件开发实例. 1/周立功等编著.
北京：北京航空航天大学出版社，2005. 1
ISBN 7 - 81077 - 583 - 9

Ⅰ. A… Ⅱ. 周… Ⅲ. 微处理器，ARM—系统设计
Ⅳ. TP332

中国版本图书馆 CIP 数据核字(2004)第 1332214 号

ARM 嵌入式系统软件开发实例(一)

周立功 等编著

责任编辑 朱伟锋 王 鸿

*

北京航空航天大学出版社出版发行
北京市海淀区学院路 37 号(100083) 发行部电话：(010)82317024 传真：(010)82328026
http://www.buaapress.com.cn E-mail:bhpress@263.net
涿州市新华印刷有限公司印装 各地书店经销

*

开本：787×960 1/16 印张：41.25 字数：924 千字
2004 年 12 月第 1 版 2004 年 12 月第 1 次印刷 印数：5 000 册
ISBN 7 - 81077 - 583 - 9 定价：56.00 元

前　言

本书自成体系，也可作为《ARM 嵌入式系统系列教程》的配套参考资料。

互联网为何发展如此迅速？因为沟通的魅力无限。沟通可以极大地促进社会发展，这不是相加的关系，而是相乘甚至是指数的关系。设想一下，远古时代的类人猿一个一个地在森林中生活，“老死不相往来”，地球上还会出现现代文明社会吗？

早期的嵌入式系统是一个个孤立的系统，与人的沟通——人机界面也很简单。这是由当时的技术水平以及当时嵌入式系统的应用场合决定的。与早期的电脑类似，早期的嵌入式系统价格昂贵，数量也很少，普通人不可能接触，只有专家才能操作。尽管如此，早期嵌入式系统的性能却很低，只能用于解决实际的问题。由于成本昂贵，性能较低，设计嵌入式系统主要考虑如何充分发挥其性能，其他方面只好割爱；因此造成只有“专家”才能使用嵌入式系统。

时过境迁，嵌入式系统已经发生了巨大变化。如今，“福特出售的‘计算能力’已超过了IBM”（福特公司的高级经理语。这里的‘计算能力’泛指通用计算机和嵌入式系统的计算能力，事实上，福特公司不生产通用计算机，只生产汽车，内嵌许多嵌入式系统），嵌入式系统已深入到社会的方方面面。嵌入式系统的广泛使用使嵌入式系统之间互相沟通变得十分重要。孤立的嵌入式系统必须事必亲为，结果任何事都不能做得最好。而嵌入式系统增多后，每个嵌入式系统可以专注于一件事，可以做得最好。同样对于一个公司，尽管每个员工都做得最好，但若员工之间没有互相沟通与协调，公司也运作不下去。因此，各个嵌入式系统之间须相互协调，甚至还需要与整个系统的外部交换数据。

本书介绍的几个软件模块都是用于嵌入式系统之间、嵌入式系统与其他系统之间、嵌入式系统与人之间的互相沟通和交换数据。鉴于 ARM 核在嵌入式系统中的地位，这些模块首先是在 PHILIPS 公司的通用 ARM7 微控制器 LPC2200 系列上调试通过的，但可以很容易移植到基于其它处理器核的嵌入式系统上。

本书各个章节内容均由各个嵌入式软件模块的编写者完成，详细地介绍了相应嵌入式软件模块的实现思想和方法。

各个章节内容安排如下：

第 1 章——ZLG/FS 原理与应用。介绍与 FAT12、FAT16 和 FAT32 兼容的文件系统的原理，并通过对 ZLG/FS 软件包的源码分析，分层次介绍如何在嵌入式系统中支持基于 FAT12、FAT16 和 FAT32 的卷，即分析 ZLG/FS 的实现。

FAT 文件系统是被通用电脑广泛支持的文件系统,也是嵌入式系统常用的文件系统,使用该文件系统,可以使用简单的方法与最广泛的系统交换数据,并使这些数据发挥最大效益。

第 2 章——USB 驱动程序开发。以 PDIUSBD12 为例,详细介绍如何使用 PHILIPS 公司的 LPC2200 ARM 微控制器开发基于 μC/OS - II 的 USB 驱动程序。通过这个例子,用户可深入了解如何较规范地编写基于 μC/OS - II 操作系统的 USB 驱动程序。

USB 是当前 PC 机流行的外设接口总线,USB 的从设备和主设备也愈来愈多。嵌入式系统如果作为 USB 从设备,与 PC 机通信就比较简单,如果再与 FAT 文件系统相结合,则可将嵌入式系统设计成 U 盘,不需要驱动程序即可与 PC 很方便地交换数据。

第 3 章——CF 卡及 IDE 接口实现与编程。详细地介绍 CF 卡(Compact Flash Card)在 True IDE 模式下的应用技术,以及 CF 卡驱动的开发方法和实例。

CF 卡是使用广泛的电子存储设备,很多数码相机、PDA 都使用它来存储数据,甚至一些手机也支持 CF 卡。CF 卡比其他电子存储设备便宜,嵌入式系统如果使用它作为存储设备,成本较低,又容易与其他设备交换数据(但须与 FAT 文件系统相结合)。如果与 USB 结合,可形成读卡器,交换数据就更加方便。

第 4 章——ZLG/IP 的原理及应用。从网络结构的角度分析 TCP/IP 协议的组成,并通过解剖 ZLG/IP 分析 TCP/IP 协议栈的实现过程。

前面几章都是介绍近距离的沟通,而用本章的软件模块可实现远距离沟通。目前互联网遍布全球,通过该软件模块可实现在全世界范围内沟通。

第 5 章——GUI 图形用户界面基础。详细介绍嵌入式系统简易的图形用户界面 ZLG/GUI 的原理,分析 Bresenham 画直线、圆和椭圆等算法,并提供实现的代码和应用例子代码。

现在嵌入式系统功能越来越强,越来越复杂,操作却要求越来越简单。这时就需要 GUI 图形用户界面来显身手。漂亮的人机界面还会提高用户的购买欲。

本书介绍的嵌入式软件模块均由广州周立功单片机发展有限公司资深工程师设计,并会不断地升级软件,力求软件越来越完善。

参与本书编写和工作的主要人员有陈明计、黄邵斌、戚军、叶皓贵、周立山、郑明远、刘英斌、岳宪臣和朱旻等。全书由周立功负责规划、内容安排、定稿与修改。

由于作者水平有限,书中难免有疏忽、不恰当甚至错误的地方,恳请各位老师及同行指正。

感谢北京航空航天大学出版社的大力支持,使本书得以快速出版;感谢 PHILIPS 美国半导体公司 CK Phua 先生几年来一如既往的支持和关心。

周立功

2004 年 11 月

目　录

第 1 章　ZLG/FS 原理与应用

第1章 ZLG/FS 原理与应用

1.1 概 述

1.1.1 ZLG/FS 简介

ZLG/FS 是广州周立功单片机发展有限公司开发的面向嵌入式系统的小型文件系统，是 ZLG 系列中间件的重要成员之一。它是与 FAT12、FAT16 和 FAT32 高度兼容的文件系统，可以直接与个人电脑交换文件。它是可移植的、可固化的文件系统，可以用于前、后台系统，也可用于多任务环境。目前，ZLG/FS 的最新版本为 1.0。

1.1.2 ZLG/FS 的特点

- 高度兼容 FAT12、FAT16 和 FAT32。ZLG/FS 可以正确访问由 Windows98 建立的 FAT12、FAT16 和 FAT32 逻辑盘，ZLG/FS 建立的逻辑盘也可以被 Windows98 正确访问。
- 可移植。全部代码由 ANSI C 编写，并且与目标处理器的存储器结构无关(即无论存储器是大端结构还是小端结构均不影响程序的执行)，方便用户移植到自己的目标系统中。
- 可固化。ZLG/FS 为嵌入系统设计，如果用户有固化手段，那么它可以嵌入到用户产品中成为产品的一部分。
- 支持多任务操作系统。提供 ZLG/FS 在 μC/OS - II 使用的接口代码，用户参考这些代码就可以很方便地在别的多任务环境下使用 ZLG/FS。
- 兼容多种介质。ZLG/FS 提供一个底层驱动程序的接口，用户只需要提供相应介质的扇区访问代码就可以在相应的介质上使用 ZLG/FS。
- 提供源代码。需要购买源码的用户可以与广州周立功单片机发展有限公司联系。
- 可配置。得到源码的用户可以对一些参数进行配置。

1.1.3 已实现的特性

- 支持多个逻辑盘;
- 支持多种介质同时使用;
- 支持树型目录结构,子目录层数不受限制;
- 支持以 FAT12、FAT16 和 FAT32 格式化的逻辑盘;
- 支持 8.3 文件名格式;
- 支持文件读/写和目录操作;
- 提供格式化(Format)范例代码。

1.1.4 暂时未实现的特性

- 对文件、目录名的限制不够严格。这一版本的 ZLG/FS 有一些 FAT 规范限制使用的字符没有过滤掉。
- 忽略文件(目录)属性中的时间相关属性。因为嵌入式系统常常没有系统时钟,而且即使有,也没有统一的标准,所以这个版本的 ZLG/FS 没有处理文件、目录与时间相关的特性。
- 忽略文件(目录)属性中的只读、存档、隐含和系统属性。这一版本的 ZLG/FS 没有对带有几个属性的文件、目录进行特殊处理。
- 忽略除第一个 FAT 表以外的所有 FAT 表。如果逻辑盘有几个 FAT 表,则这一版本的 ZLG/FS 只操作第一个 FAT 表,其他 FAT 表保持不变。这几个 FAT 表就不同步。
- 忽略长文件名。这一版本的 ZLG/FS 不支持创建和访问长文件名的文件。但具有长文件名的文件还是可以通过短文件名访问。

ZLG/FS 的升级版将逐步解决上述问题。

1.2 使 用

1.2.1 使用示例

程序清单 1.1 是一个简单的 ZLG/FS 使用范例。它首先初始化文件系统(程序清单 1.1(2)~(4));再建立一个目录(程序清单 1.1(5)~(7));然后读/写一个文件(程序清单 1.1(8)~(17)),接着删除这个文件(程序清单 1.1(18))和目录(程序清单 1.1(19)、(20));最后关闭文件系统(程序清单 1.1(21)、(22))。

程序清单 1.1　ZLG/FS 使用示例

```
#include "config.h"                                                    (1)

int main(void)
{
    HANDLE FHandle;
    unit8 buf[10];
    char *S  =  "a.txt";

    DiskInit();                                                        (2)
    AddFileDriver(FloppyCammand);                                      (3)
    FileInit();                                                        (4)

    ChangeDrive("a:");                                                 (5)
    MakeDir("dir2.dir");                                               (6)
    ChangeDir("a:\\dir2");                                             (7)

    FHandle = FileOpen("a.txt", "w");                                  (8)
    if  (FHandle != Not_Open_FILE)                                     (9)
    {
        FileSeek(FHandle, 0, SEEK_END);                                (10)
        FileWrite(S, 6, FHandle);                                      (11)
        FileClose(FHandle);                                            (12)
    }

    FHandle = FileOpen("a.txt", "r");                                  (13)
    if  (FHandle != Not_Open_FILE)                                     (14)
    {
        FileSeek(FHandle, 0, SEEK_SET);                                (15)
        FileRead(buf, 6, FHandle2);                                    (16)
        FileClose(FHandle);                                            (17)
    }

    RemoveFile(S);                                                     (18)

    ChangeDir("a:\\");                                                 (19)
    RemoveDir("dir2");                                                 (20)
```

```
    RemoveFileDriver(GetDrive("a"));                                    (21)
    return 0;                                                           (22)
}
```

1.2.2 Config.h 和 fat.h

程序清单 1.1 中的第一句就是 include "config.h",并且 config.h 是惟一包含的头文件,其他头文件均包含在这个头文件中。读者可能会注意到,本书所有的 *.c 文件都是这样。一般情况下,用户的 *.c 也应该这样做,这样,用户就不必在工程项目中每个 *.c 都考虑应该包含什么头文件。这样的代价是增加了编译时间,但现代计算机速度相当快,增加的时间并不多。这样做还有一个很重要的好处,即增加了可移植性。如果将程序由一种 CPU 移植到另一种 CPU,就不必每一个 *.c 文件都修改所包含的文件了。config.h 还可以保存用户自己的设置。

每一个 config.h 文件中均包括如程序清单 1.2 的定义。

程序清单 1.2　fat.h

```
#include "fat.h"
```

fat.h 定义了 ZLG/FS 的接口函数及一些 ZLG/FS 使用到的宏和数据结构。如果用户需要使用 ZLG/FS,则必须包含这个文件。

1.2.3 与编译器无关的数据类型

因为标准 C 语言没有规定 short、int、long 等数据类型的字长,所以不同的 C 语言编译器根据自己所对应的 CPU 的字长随意定义,隐含不可移植性。针对这种情况,ZLG/FS 要求用户定义与编译器无关的数据类型,以保证可移植性。LPC2200 的这部分代码见程序清单 1.3。

程序清单 1.3　定义与编译器无关的数据类型

```
typedef unsigned char     uint8;        /*无符号 8 位整型变量*/
typedef signed char       int8;         /*有符号 8 位整型变量*/
typedef unsigned short    uint16;       /*无符号 16 位整型变量*/
typedef signed short      int16;        /*有符号 16 位整型变量*/
typedef unsigned int      uint32;       /*无符号 32 位整型变量*/
typedef signed int        int32;        /*有符号 32 位整型变量*/
typedef float             fp32;         /*单精度浮点数(32 位长度)*/
typedef double            fp64;         /*双精度浮点数(64 位长度)*/
```

这样,假设用户在 32 位系统中使用 ZLG/FS,在程序中定义了一种数据变量如 uint32,它

是一个 32 位无符号整数，范围是 0～0xFFFFFFFF；然后又把它移植到 16 位系统中。此时 unsigned int 已经是 16 位了，但 uint32 仍然是 32 位，只是将 uint32 的定义改为 unsigned long int。

1.2.4 初始化 ZLG/FS

在使用 ZLG/FS 之前，首先需要初始化它。初始化 ZLG/FS 主要包括以下几个步骤：

① 初始化磁盘信息管理系统(程序清单 1.1(2))。

② 加载逻辑盘驱动程序(程序清单 1.1(3))，可以加载多个逻辑盘驱动程序。

③ 初始化文件管理系统(程序清单 1.1(4))。

执行上述 3 个步骤后即可正常使用 ZLG/FS。

1.2.5 目录相关操作

ZLG/FS 提供很多函数用来操作逻辑盘的目录，其主要部分见表 1.1。函数的具体使用请参考 1.9 节和 1.11 节。

表 1.1 ZLG/FS 目录操作函数列表

函数名	功 能
GetDrive	获取逻辑盘索引号
ChangeDrive	改变当前逻辑盘
MakeDir	建立目录
RemoveDir	删除空目录
ChangeDir	改变当前目录

1.2.6 文件相关操作

ZLG/FS 提供很多函数用来操作逻辑盘的文件，其主要部分见表 1.2。函数的具体使用请参考 1.10 节和 1.11 节。

表 1.2 ZLG/FS 文件操作函数列表

函数名	功 能
RemoveFile	删除文件
FileOpen	打开文件
FileClose	关闭文件
FileGetCh	从文件中读取 1 个字节

续表 1.2

函数名	功　能
FileRead	从文件中读取数据块
FilePutCh	写1个字节到文件
FileWrite	写1个数据块到文件
FileCloseAll	关闭所有文件
FileEof	判断文件是否读/写到文件尾
FileSeek	移动文件读/写位置

1.2.7 关闭 ZLG/FS

当因为某种原因机器需要停机或重启时，可以关闭 ZLG/FS。关闭 ZLG/FS 时，只需要卸载所有已加载的逻辑盘驱动程序(程序清单 1.1(21))即可。此时，用户可以停机或重启机器，或是加载新的逻辑盘驱动程序。

1.2.8 在多任务环境下使用 ZLG/FS

前面介绍的是在前、后台系统下 ZLG/FS 的使用，在多任务环境下其使用略有不同。程序清单 1.4 为 μC/OS-II 中使用的例子，它与程序清单 1.1 的功能完全一样。

程序清单 1.4　在 μC/OS-II 中使用 ZLG/FS

```
#include "config.h"

OS_STK        TaskStk[1024];
OS_STK        TaskStartStk[1024];

void  TaskStart (void * pdata);
int main(void)
{
        OSInit();
        PC_DOSSaveReturn();
        PC_VectSet(uCOS, OSCtxSw);
       OSTaskCreate(TaskStart, (void *)0, &TaskStartStk[1023], 9);
        OSTaskCreate(OSFileTask, (void *)0, &TaskStk[1023], 8);                    (1)
        OSStart();
        return 0;
}
```