FAT32 文件系统在 Cortex-M3 音乐播放器中的应用

何 谐

（江阴职业技术学院，江阴 214400）

摘要：主要介绍了 FAT32 文件系统原理及其在音乐播放器中的实现方法，并介绍了一种 Cortex-M3 内核的音乐播放器的硬件电路和软件系统。该音乐播放器以 STM32F103RB 芯片为主控器，连接 VS1003 解码芯片，使用 SD 卡存储音乐文件，通过 FAT32 文件系统识别音乐文件。实验证明，该系统稳定实用。

关键词：FAT32 文件系统；音乐播放器；Cortex-M3

中图分类号：TN912.2　　**文献标识码**：A

Application of FAT32 File System in Cortex-M3 Music Player

He Xie

(Jiangyin Polytechnic College, Jiangyin 214400, China)

Abstract: This paper introduces the principle of FAT32 file system and an implementation in Cortex-M3 music player, also the hardware and software design of the music player is introduced. The music player takes STM32F103RB chip as main controller and connects it to VS1003 decoding chip. Music files are stored in SD card and identified by FAT32 file system. Experimental test shows the system is stable and practical.

Key words: FAT32 file system; music player; Cortex-M3

引 言

FAT 类文件系统是 Windows 操作系统在磁盘文件管理上最常用的一种文件系统。而 FAT32 文件系统是微软 FAT 类文件系统中的最高版本，微软从 Windows95 版本开始，后续的操作系统均支持 FAT32 文件系统，是现今 Windows 下最常用的硬盘文件系统。当人们使用 SD 卡等存储装置从计算机上拷取文件时，存储器中的文件管理即符合 FAT32 文件系统的管理原则，因此，使用嵌入式芯片设计音乐播放器时，为了能自动识别 SD 卡上的音乐文件，关键是 FAT32 文件系统在嵌入式芯片上的实现，本文研究了 FAT32 文件系统在 Cortex-M3 内核的音乐播放器上的应用，并设计了基于 FAT32 的音乐播放器。

1 系统硬件设计

Cortex-M3 内核是 ARM 公司推出的基于 ARMv7 体系架构的处理器核，具有高性能、低成本、低功耗的特点，而意法半导体公司的 STM32 系列芯片 STM32F103RB 则是在 Cortex-M3 内核的基础上扩展了高性能的外围设备。该芯片工作频率为 72 MHz，内置高速存储器（128 KB 的闪存和 20 KB 的 SRAM）。由于其丰富的外设和优异的性价比，音乐播放器选择其作为主控制器芯片，利用其自带的 2 组 SPI 接口读取 SD 卡音频文件以及缓存，并将读取的音频数据流送至音频解码器 VS1003 进行解码；同时主控制器还负责人机交互，连接 TFT 屏幕显示歌曲名称以及键盘用以选择曲目。

本音乐播放器选择 SD 卡作为存储装置。SD 卡使用前应通过读卡器连接至计算机，格式化为 FAT32 文件格式，同时将 *.mp3、*.wav、*.wma 格式的音乐文件复制到 SD 卡中。一般来说，SD 卡支持两种工作模式：SD 模式和 SPI 模式。由于 STM32F103RB 本身具有 2 个 SPI 接口，因此 SD 卡以 SPI 模式连接 STM32F103RB 的 SPI1 口。

VS1003 是荷兰 VLSI 公司出品的一款单芯片 MP3/WMA/MIDI/WAV 音频解码和 ADPCM 编码芯片，通过 SPI 控制。该系统中，VS1003 作为主控芯片的从机使用，STM32F103RB 通过它的 SPI2 口向 VS1003 不断输出音频数据流，VS1003 自动解码，并连接外部功放和喇叭，就可以听到所播放的音乐了。

STM32F103RB 与 SD 卡和 VS1003 的连接图如图 1 所示。

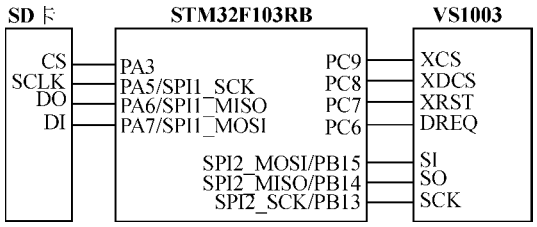

图 1　STM32F103RB 与 SD 卡和 VS1003 的连接图

2　FAT32 文件系统基本原理

FAT 类文件系统对文件的存储空间是按簇进行划分和管理的，即一个文件总是占用若干个整簇，即文件最后一簇剩余的空间将不再使用。不同的簇采用簇号来区分，FAT32 文件系统使用 32 位二进制来表示簇号，适用于空间大于 512 MB 的磁盘。每簇对应的扇区数将决定磁盘的容量，扇区是仅小于簇的存储单位，一般每扇区大小为 512 字节。为了提高对整个磁盘空间的利用率，“簇”不宜大，也不宜小。对于小容量的磁盘，FAT32 系统实际每个簇为 4 KB，也就是 8 个扇区；当磁盘容量超过 8 GB 时，每个簇为 8 KB，32 GB 以上每簇为 32 KB。

当 SD 卡被格式化为 FAT32 文件格式时，SD 卡中的逻辑盘空间就被划分为三大部分：保留区、FAT 表区（文件分配表区）、DATA 区。其中保留区和 FAT 表区又合称为系统区，文件的真正存储从 DATA 区开始。

2.1　保留区

保留区包含了 FAT32 的重要数据结构——MBR（主引导记录）和 DBR（系统引导扇区）。MBR 如果存在，则绝对 0 扇区就是 MBR；如不存在，此扇区就是 DBR。MBR 的前面 446 字节为引导程序，随后的 64 字节是 4 条记录，称为 DPT（磁盘分区表），每条记录均包含了分区的开始磁头、开始柱面与分区类型、分区的结束头、结束柱面与扇区、分区的第一个扇区、总扇区数等信息。由于 SD 卡容量不是很大，通常不作分区，后面三条分区记录均为 0。

通过 DPT 可以找到 DBR 扇区，在 DBR 中记录了分区的很多重要的信息，这些重要信息都保存在 BPB（BIOS 参数块）这个区域。这些重要信息包含了 FAT32 的大部分全局参数，例如：每扇区的字节数、每簇扇区数、保留扇区数、FAT 表数、FAT 区前隐扇区数、FAT 表所占扇区数、第一个目录的簇号等。

2.2　FAT 表区

DBR 扇区之后是保留扇区，再接着便是 FAT 表区了，FAT32 中有两个 FAT 表，第二个是第一个的备份，通过对 BPB 的解析可知 FAT 表的开始扇区和大小。FAT 表是一个链式结构，每 4 个字节（一个 32 位二进制）为一个 FAT 表项，每个 FAT 表项对应一个簇，00 簇和 01 簇被系统保留，因此前 2 个 FAT 表项为特殊字符。从 02 簇开始，每个簇都依次对应一个 FAT 表项内容。如果该簇未使用或已回收，相应 FAT 表项内容写零，坏簇以 0FFFFFF7H 标识；如果该簇是文件的最后一簇，FAT 表项值为 0FFFFFFFH；如果该簇不是文件的最后一簇，FAT 表项值为该文件占用的下一个簇的簇号，文件占用的各簇构成一个簇链，保存在 FAT 表中。因此，只要知道文件的起始簇号，就可以根据该链式结构找到整个文件。当新建文件时，如果新建的文件只占用一个簇，为其分配的簇对应的 FAT 表项将会写入结束标记，如不只占用一个簇，则在 FAT 表项中记录下一簇簇号直至结束；当删除文件时，文件所对应的 FAT 表项将被设置为 0，以表示其对应的簇处于未分配状态。

2.3　DATA 区

DATA 区从 02 簇开始，该区根据存放的内容，可分为根目录区和文件数据区。根目录区存放根目录文件，通过 BPB 解析可知 02 簇的开始扇区，该扇区也称根目录簇开始扇区。从这个扇区开始依次存放文件目录项，每个文件目录项占用 32 字节，根目录文件的大小随文件目录项个数增加而增加。除了第一个 32 字节为文件卷标，其后的每个文件目录项均描述了和文件相关的大部分信息，如文件名、文件创建时间、访问时间、文件大小、文件起始簇号等等。解析出这些文件信息，特别是文件起始簇号，嵌入式芯片就可以根据 FAT 表访问任意文件内容了。根目录项中的内容也可以是子目录文件的信息用同样方法也可以找出子目录下的所有文件的信息。

3　FAT32 文件系统的实现

FAT32 在 Cortex-M3 内核的嵌入式芯片上的实现依赖于一些必要的结构体，这些结构体包括了 FAT32 文件系统的重要区域信息，通过对这些结构体的解析，可以方便计算出目标文件的重要信息。因此，FAT32 文件系统首先定义这些结构体，包括 MBR 结构体 struct PartSector、DPT 结构体 struct PartRecord、BPB 结构体 struct FAT32_BPB、文件目录项结构体 struct direntry 和文件信息结构体 struct FileInfo。

3.1　FAT32 与 SD 卡的接口

FAT32 文件系统本身就是用来管理和控制对扇区数据的读和写，因此，需要构建一个 FAT32_ReadSector() 函数，用来读取存储设备的扇区，与 FAT32_WriteSector() 一起构成其他 FAT32 函数的底层函数。由于系统的存储设备是 SD 卡，SD 卡函数 SD_ReadDisk() 可从指定物理扇区号读取 1 个扇区的数据，放入全局变量数据缓冲区

FAT32_Buffer[512]中，因此，使用 SD_ReadDisk()来作为这个底层函数。

3.2 初始化 FAT32 文件系统

初始化 FAT32 文件系统包括以下几个函数。

寻找 DBR 函数 FAT32_Find_DBR()：此函数用 FAT32_ReadSector()读入 SD 卡 0 扇区数据，强制类型转换为 MBR 结构体 struct PartSector，解析出其中的 DPT 结构体 struct PartRecord，返回 DBR 所在扇区号。

FAT32_Init()函数：将 DBR 扇区数据读入缓冲区，强制类型转化为 BPB 结构体，解析 struct FAT32_BPB，计算出 FAT32 的重要信息：每扇区字节数、每簇扇区数、第一个 FAT 表扇区号、FAT 表占用的扇区数、根目录簇号、根目录簇开始扇区、根目录占用扇区、第一个数据扇区，作为全局变量参数。

ClustToSector()函数：输入参数是任意簇号，返回该簇所对应的起始扇区号。

GetNextCluster()函数：根据文件当前簇号，在 FAT 表中找到文件下一簇号返回，如没有后继簇，则返回 0x0ffffff8，表示文件结束。

3.3 打开文件后获取文件信息

Get_File_Info()：该函数输入参数为根目录簇号 dir_clust、全局参数文件信息结构体 struct FileInfo、音乐文件类型、音乐文件序号 count。该函数首先根据根目录簇号找到该簇起始扇区号，连续读取扇区，直到 FAT 表中簇链结束。对读取的每个扇区数据以 32 字节为单位强制转换为文件目录项结构体 struct direntry，对每个合法的文件目录项结构体 struct direntry 进行解析。若音乐文件类型符合输入参数，文件索引号自增 1；若该文件索引号即为音乐文件序号时，表示找到目标音乐文件。将该目标文件的 struct direntry 的 32 字节内容转化到全局变量文件信息结构体 struct FileInfo，提取文件信息详情。struct FileInfo 结构体如下：

```
typedef struct{
    unsigned char File_ShortName[8]; //8 字节的短文件名
    unsigned char File_LongName[80];//80 字节的长文件名
    unsigned long File_StartCluster;    //文件起始簇号
    unsigned long File_Size;            //文件字节大小
    unsigned char File_Attr;            //文件属性
    unsigned long File_Type;            //文件类型
    unsigned short File_CreateTime;     //文件创建时间
    unsigned short File_CreateDate;     //文件创建日期
    unsigned long File_CurClust;        //文件当前簇号
}FileInfo;
```

4 音乐播放器系统软件设计

音乐播放器可循环播放 SD 卡上的音乐格式文件，只要文件格式为 *.mp3、*.wav、*.wma，就可顺序播放，TFT 屏幕显示正在播放的音乐文件名称，通过对按键的操作可实现歌曲的暂停/播放、选择上一首和下一首。该音乐播放器软件系统设计除 FAT32 文件系统 FAT32.c 之外，还包括以下几大模块：歌曲播放模块 music_play.c 为软件的最高层应用层；SD 卡模块 sd.c、VS1003 模块 vs1003.c 为 music_play.c 提供设备驱动；而 SPI 模块spi.c 为最底层的物理数据交换程序，为 sd.c 和 vs1003.c 提供支持。软件系统结构如图 2 所示。

应用层	music_play.c
文件系统层	FAT32 .c
设备驱动层	sd.c vs1003.c
物理层	spi.c

图 2 软件系统结构

应用层程序 music_play.c 在初始化之后首先计算出根目录下音乐格式歌曲的数目，随后判断按键状态，当选择前一首、后一首或一首歌播放完毕时，改变播放歌曲的索引号，播放该索引号的歌曲。

播放某歌曲时，首先利用 FAT32 函数 Get_File_Info()，根据歌曲索引号和文件类型来构造该文件的文件信息结构体 struct FileInfo，并获取这首歌的文件起始簇号、歌曲名称等信息。根据文件开始簇号，调用 FAT32 函数 FAT32_ReadSector() 读取 1 扇区数据，存入 STM32F103RB 的 512 字节的数据缓冲区，缓冲区数据随后通过 SPI 函数 SPI2_ReadWriteByte()发送给 VS1003 播放。当该簇所有扇区播放完毕后，一簇结束，利用 FAT32 函数 GetNextCluster()在 FAT 表中继续寻找下一簇簇号，循环上述过程，直到簇链结束，歌曲也播放完毕。

歌曲播放主流程如图 3 所示。

系统读取 SD 卡时使用 STM32F103RB 的 SPI1 口，设置为高速模式；发送数据到 VS1003 时使用 SPI2 口，设置为低速模式。这样在歌曲的播放过程中，数据的读取速度永远超过数据的发送速度，可以得到比较好的音质，不会出现声音的抖动。

结 语

本文结合 FAT32 文件系统提出嵌入式系统的音乐播放器设计方案，研究了 FAT32 文件系统在 Cortex-M3

```
}
```

(2) 中断向量的映射和初始化函数

```
void __irq TIMER2(void){
  rI_ISPC=BIT_TIMER2;   //清除 TIMER2 的中断请求位
}
void Isr_Init(void){
  U32 i;
  for(i=_RAM_STARTADDRESS;i<(_RAM_STARTADDRESS+0x20);i+=4){
    *((volatile unsigned*)i)=0xEA000000+0x1FFE;
                          //中断向量加载到 0x0c008000 处
  }
  rINTCON=0x5;//非向量中断模式,使能 IRQ 中断
  rINTMOD=0x0;//中断全部为 IRQ 模式
  pISR_EINT0=(unsigned)KEYIN0;
        //把中断函数给指针 pISR_xxx,链接到
        //"_ISR_STARTADDRESS+某中断的偏移地址"后
  pISR_TIMER2=(unsigned)TIMER2;
  pISR_TIMER0=(unsigned)TIMER0;
  rINTMSK=~(BIT_GLOBAL|BIT_EINT0|BIT_TIMER2|BIT_TIMER0);
        //打开外部中断 0,定时器中断 2 和定时器中断 0
}
```

(3) 系统主函数

```
void Main(void){
  rSYSCFG=SYSCFG_8KB;//使用 8 KB Cache 允许写缓存操作
  Port_Init();
  Timer0_Pwm();
  Timer2_Pwm(100,50);
  Isr_Init();
  while(1){
    rTCON=0x999909;
          //定时器初始化,让 PE7、6、5、4、3 输出 PWM 信号
  }
}
```

结 语

以 ARM 微处理器及其外围电路实现的电子凸轮可以开发为专有的凸轮控制器,应用在全自动枕式包装机及其他相关产品的控制系统中。本文设计了一种电子凸轮的速度曲线,并在此基础上提出了一 种加速度值的计算方法。通过简单的硬件系统配置,依据上述算法结构编写 C 语言程序,S3C44B0X 微处理器 I/O 口最终输出的脉冲频率驱动伺服电机,实现了速度无零点、无跳变的柔性变速运转。

参考文献

[1] 三星公司. S3C44B0X 英文数据手册,2007.

[2] 张勇. ARM 原理与 C 程序设计[M]. 西安:西安电子科技大学出版社,2009.

[3] 尹章伟,刘全香,马桃林. 包装概论[M]. 北京:化学工业出版社,2006.

[4] 黄义萍,魏军会. 电子凸轮及其在自动化生产中的应用[J]. 淮北煤师院学报,2001,22(1).

[5] 王程,贺炜. 基于单片机的电子凸轮系统研究[J]. 机械设计与制造,2006(4).

[6] 周高峰,曹巨江. 电子凸轮通电脉冲频率的推导及其控制程序[C]//全国印刷、包装机械凸轮、连杆机构学术研讨会论文集,2005:26-28.

(责任编辑:梅栾芳 收稿日期:2013-02-27)

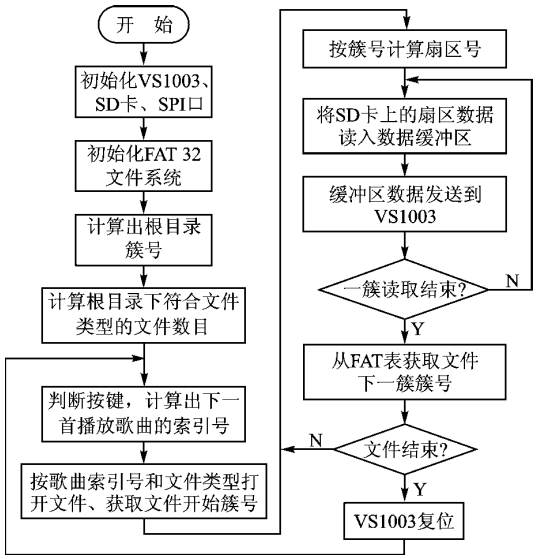

图 3 歌曲播放流程图

内核的音乐播放器中的实现方法,对便携式多媒体播放器的研发具有一定的指导意义。

参考文献

[1] 何立民. 单片机高级教程[M]. 北京:北京航空航天大学出版社, 2007.

[2] 戴士剑,涂彦辉. 数据恢复技术[M]. 2 版. 北京:电子工业出版社,2007.

[3] 刘伟. 数据恢复技术深度揭秘[M]. 北京:电子工业出版社, 2010.

[4] 唐继贤. 51 单片机应用系统开发实例精解[M]. 上海:科学技术出版社,2012.

[5] 刘军. 例说 STM32 [M]. 北京:北京航空航天大学出版社, 2011.

何谐(讲师),研究方向为单片机与嵌入式应用。

(责任编辑:梅栾芳 收稿日期:2013-01-29)