STM32 单片机的 SD 卡存储器读写模块设计*

汤才刚，刘京京，沈瑞东

（上海舜华新能源系统有限公司，上海 201806）

摘要：为了满足氢气加气机在使用过程中大量数据的存储要求，设计了一种利用 SD 卡来扩展存储的电路模块。在兼顾稳定性与读写速度的基础上，利用 STM32 单片机与 SD 卡存储器件的接口技术，把 SD 卡的读写以及 FatFS 文件系统的读取写入移植到此单片机上。在读写的过程中，使用 Windows 中通用的文件系统使最终存储在 SD 卡的数据可以直接被 PC 机读取，使开发利用有更好的扩展性，取得了良好的效果。硬件设计方面利用单片机的 SPI 接口，只需要 4 个 I/O 口再加上几个上拉电阻就可实现与 SD 卡的接口电路，软件设计方面利用 IAR 嵌入式集成开发环境和移植 STM32 库函数，并使用开源的 FatFS 文件系统模块源代码实现单片机 SD 卡的读写编程。

关键词：STM32 单片机；SPI 接口；SD 卡；FatFS 文件系统

中图分类号：TP274　　**文献标识码**：A

Design of SD Card Read-write Module Based on STM32

Tang Caigang，Liu Jingjing，Shen Ruidong

（Shanghai Sunwise Energy Systems Co.，Ltd.，Shanghai 201806，China）

Abstract：In order to meet the large data storage requirements of the hydrogen dispenser in use，a circuit module that uses SD card to expand storage is designed. Based on the balance between stability and reading and writing speed，the interface technology of STM32 microcontroller and SD card storage device are used to transplant SD card read and write and FatFS file system read and write to this microcontroller. In the process of read and write，the general file system in Windows is used，so that the data finally stored in the SD card can be directly read by the PC，the development and utilization have better scalability and achieved good results. The hardware design uses the SPI interface of the single-chip microcomputer，only four I/O ports and a few pull-up resistors can be used to achieve the interface circuit with the SD card. The software design uses the IAR embedded integrated development environment and transplantation of STM32 library functions，and use the source code of the open source FatFS file system module to realize the programming of the read and write of the SD card of the single chip microcomputer.

Key words：STM32 microcontroller；SPI interface；SD card；FatFS file system

引　言

在氢气加气机使用过程中，经常需要存储大量的数据，并且要求在 PC 机上查看数据。由于受单片机自身硬件的限制，为了实现大容量数据的存储，必须要通过单片机的外部扩展来实现，现在通常使用的方法是在单片机外部扩展大容量 EEPROM 或 FLASH 芯片来实现。这种方法有两个缺点：①外部电路连接复杂，通用性差，对于不同的系统需要采用不同的电路；②存储的数据读取非常不便，需要通过单片机系统发给 PC 机来实现。综合比较后发现，最适合单片机系统的莫过于 SD 卡了，不仅容量大（32 GB 以上），而且支持 SPI 方式，方便移动，能满足不同应用的要求。只需要几个 I/O 口就可以外扩一个最大达 32 GB 以上的外部存储器，容量选择范围很大，更换也很方便，而且方便移动，编程也简单，是单片机大容量外部存储器的首选[1]。

为了方便管理 SD 卡中的文件、高效地读写数据，需要在 SD 卡中装载文件系统。因 FatFS 文件系统源码开源，资源占用少，兼容性好，使用范围较广，本设计利用 STM32 自带的 SPI 接口与 SD 卡连接通信，电路非常简单且易于实现，并成功实现了在 SD 卡中建立 FatFS 文件系统，使 STM32 单片机能够对 SD 卡中的文件进行读写等操作。

1　系统结构设计

SD 卡一般支持 2 种模式：SD 模式和 SPI 模式。主机

* 基金项目：上海市科学技术委员会科研计划项目（19DZ1206100）。

可以选择任意一种模式与 SD 卡通信，SD 模式允许 4 线的高速数据传输，SPI 模式允许简单地通过 SPI 接口与 SD 卡通信，这种模式较 SD 模式速度慢，不过利用 STM32 片内集成的 SPI 接口，最大通信速度可达 18 Mbps，每秒可传输数据 2 MB 以上，对于单片机系统一般的应用足够了[2]。

本设计的 SD 卡接口电路硬件设计主要由以 STM32F103VCT6 为核心的处理器、电源和 SD 卡及连接器组成。系统整体结构如图 1 所示。

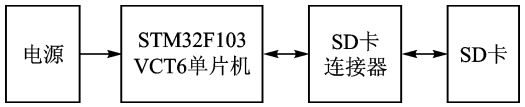

图 1　系统整体结构

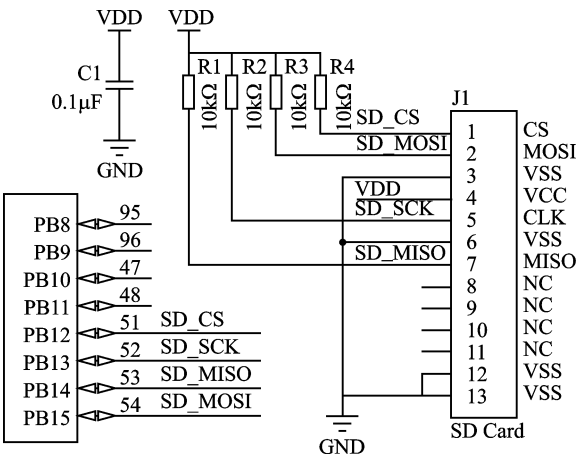

图 2　SD 卡与 STM32 连接电路图

2　硬件电路设计

2.1　STM32 单片机简介

本设计使用的主控制芯片为 STM32F103VCT6 单片机，主要实现 SPI 模块功能，控制 SD 卡的读写。该单片机是 ST 公司推出的基于 ARM Cortex－M3 内核的高性能的微处理器芯片，该单片机功能强大，处理速度快。在软件开发上应用 IAR Embedded Workbench for ARM 嵌入式集成开发平台进行产品研发，使得整个产品的研发周期大大缩短并使整个程序移植性强，方便易懂。

STM32F103VCT6 单片机工作主频最高可达 72 MHz，封装形式为 LQFP100，内置 256 KB FLASH 和 48 KB RAM，包含 10 个定时器、2 个 12 位的 ADC、5 个异步通信串行口 USART、3 个同步通信串行口 SPI、2 个 I^2C 接口、80 个 I/O 口等[4-5]。

2.2　SD 卡接口电路设计

本设计中使用 STM32F103VCT6 单片机 PB12、PB13、PB14、PB15 这 4 个 I/O 口所组成的片内集成 SPI2 接口与 SD 卡进行通信，单片机为主设备，SD 卡为从设备。所对应的信号线分别为 SD_CS 为片选控制端；SD_SCK 为主设备时钟输出端口，与 SD 卡时钟输入端口相连；SD_MISO 为主设备输入端口，与 SD 卡数据输出端口相连；SD_MOSI 为主设备输出端口，与 SD 卡数据输入端口相连。这 4 根信号线必须接上拉电阻[3]，连接电路图如图 2 所示。

3　软件设计

3.1　FatFS 文件系统应用

FatFS 是一个开源的、为小型嵌入式系统设计的通用 FAT 文件系统模块，支持 FAT12、FAT16 与 FAT32，支持多种存储媒介，有独立的缓冲区，可对多个文件进行读写，可裁剪的文件系统。FatFS 的编写遵循 ANSI C，并且完全与磁盘 I/O 层分开。因此，它独立（不依赖）于硬件架构，可以被嵌入到低成本的微控制器中（如 AVR、8051、PIC、ARM、Z80、68K 等），而不需要做任何修改。FatFS 文件系统在嵌入式软件开发中的应用图如图 3 所示，应用层使用 FatFS 提供的 API 函数与 FatFS 模块通信，FatFS 模块使用 FatFS 提供的底层存储介质接口函数对存储设备进行配置，只有这样才能正常使用 FatFS 文件系统[4]。

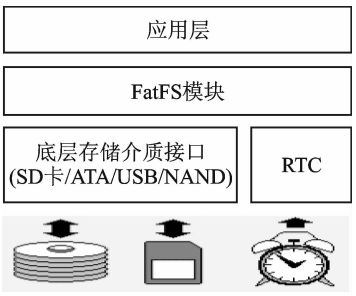

图 3　FatFS 文件系统应用图

3.2　程序流程图

主程序总体设计思路：系统上电后单片机内部进行复位，接着对系统时钟、GPIO 口、SD 卡接口硬件进行配置。GPIO 口初始化时需要使能 SPI2 时钟，否则无法开启单片机的 SPI 模式。接下来配置 FatFS 底层存储介质接口函数，将 SD 驱动有关的操作放在文件系统中，再调用 API 接口函数，将 SD 卡写入数据或读取 SD 卡的数据，最后在 PC 机上对数据进行验证[5]。主程序设计流程如图 4 所示。

3.3　SD 卡 SPI 模式下的命令

SPI 模式下 SD 卡和单片机的通信采用发送应答机制。首先主机端（单片机）发送命令 command，SD 卡应答 response。SD 卡的命令格式如下：

字节 1			字节 2～5	字节 6	
7	6	5　0	31　0	7　1	0
0	1	command	命令参数	CRC	1

SD 卡的指令由 6 个字节组成，字节 1 的最高 2 位固

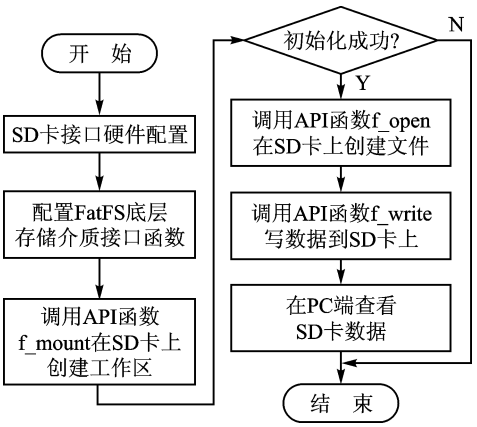

图 4 程序流程图

定为 01，低 6 位为命令号，比如 CMD17，为 1 0001 即 16 进制 0x11，完整的 CMD17，第一个字节为 0101 0001，即 0x11＋0x40；字节 2～5 位为命令参数，有些命令没有参数；字节 6 的高 7 位为 CRC 值，最低位固定为 1。

SD 卡的命令总共有 12 类，下面是几个重要的命令，如表 1 所列。

表 1 SD 卡部分操作指令

命 令	参 数	回 应	描 述
CMD0(0x00)	NONE	R1	复位 SD 卡
CMD8(0x08)	VHS＋Check Pattern	R7	发送接口状态命令
CMD9(0x09)	NONE	R1	读取卡特定数据寄存器
CMD10(0x0A)	NONE	R1	读取卡标志数据寄存器
CMD16(0x10)	按大小	R1	设置块大小(字节数)
CMD17(0x11)	地址	R1	读取一个块的数据
CMD24(0x18)	地址	R1	写入一个块的数据
CMD41(0x29)	NONE	R3	发送给主机容量支持信息和激活卡初始化过程
CMD55(0x37)	NONE	R1	告诉 SD 卡，下一个是特定应用命令
CMD58(0x3A)	NONE	R3	读取 OCR 寄存器

单片机每发送一条命令，SD 卡都会给出一个应答，以告知主机该命令的执行情况，或者返回主机需要获取的数据。SPI 模式下，SD 卡针对不同的命令，应答可以是 R1～R7，其中 R1 的应答最多，其格式如下：

BIT7	BIT6	BIT5	BIT4	BIT3	BIT2	BIT1	BIT0
0	参数错误	地址错误	连续擦除错误	命令 CRC 错误	非法命令	擦除错误	IDLE 状态

3.4 SD 卡初始化

对 SD 卡进行初始化操作，是 SD 卡进行正常数据读写的前提。SD 卡接入后，默认进入 SD 模式，等待电压稳定需上电延时 250 ms 即等待至少 74 个时钟周期。将时钟周期频率设置为 100～400 kHz，拉低片选信号 CS，发送 CMD0。如果收到应答信号 01H，则表示 SD 卡进入 SPI 模式[6]。

SD 卡的初始化过程如下：①初始化与 SD 卡连接的硬件配置(CPU 的 SPI 配置，I/O 口配置)；②上电延时(大于 74 个 CLK)；③复位卡(CMD0)，进入 IDLE 状态；④发送 CMD8，检查是否支持 2.0 协议；⑤根据不同协议检查 SD 卡并判断卡类型(命令包括 CMD1、CMD55、CMD41 和 CMD58 等)；⑥取消片选，发送 8 个 CLK 后，结束初始化。

这样就完成了对 SD 卡的初始化，注意末尾发送的 8 个 CLK 是提供 SD 卡额外的时钟，完成某些操作。通过 SD 卡初始化可以知道 SD 卡的类型(V1、V2、V2HC 或 MMC)，在完成初始化之后，就可以开始读写数据了。

3.5 SD 卡读写操作

SD 卡的数据读写以块为单位，一个块的最大长度为 512 字节，在初始化中进行设置。单片机发送 CMD17 或 CMD18 进行 SD 卡的单个块或多个块的读操作；单片机发送 CMD24 或 CMD25 进行 SD 卡的单个块或多个块的写操作。下面以读写单个块为例，介绍具体操作过程，读写多个块的过程与此类似，命令为 CMD18 和 CMD25。

读 SD 卡单个块数据，具体过程如下：①发送 CMD17；②接收卡响应 R1；③接收数据起始令牌 0xFE；④接收数据；⑤接收 2 个字节的 CRC，如果不使用 CRC，这两个字节在读取后可以丢掉；⑥禁止片选之后，发 8 个 CLK。

写 SD 卡单个块数据，具体过程如下：①发送 CMD24；②接收卡响应 R1；③发送写数据起始令牌 0xFE；④发送数据；⑤发送 2 字节的伪 CRC；⑥禁止片选之后，发 8 个 CLK[7]。

SD 卡数据的读写基本单位是块，如需大数据量的读写操作，可以有两种方法实现：单块的多次读写和多块读写的命令。在速度要求不高的场合可以使用单块的多次读写，在高速的数据读写场合，必须使用多块读写的命令，效率比单块的多次读写高很多。

3.6 FatFS 文件系统模块移植

在 STM32 程序工程中需要新建两个文件夹，FatFS 用于存放 FatFS 源文件，User 文件夹下 SPI_SD_Driver.c 文件用于存放 SPI 的底层驱动文件，这个文件是 SD 卡初始化和读写相关的函数。这些函数是在 diskio.c 文件中的 5 个函数所调用的：函数 disk_initialize，SD 卡的初始化，调用底层的 SD_Init()函数；函数 disk_status，获取 SD 卡的状态，这里可以不用管；函数 disk_read，从 SD 卡中读取数据，含读 SD 卡单块数据函数和多块数据函数；函数 disk_write，将数据写入 SD 卡，含写 SD 卡单块数据函数

和多块数据函数，若该文件系统为只读文件系统则不用实现该函数，含写单块函数和多块函数；函数 disk_ioctl，获取 SD 卡文件系统相关信息[8]。

4 SD 卡文件读写实现

FatFS 底层存储介质接口函数已修改完成，下一步就需要使用 API 函数实现文件的读写[9]。具体实现代码如下：

```
unsigned charTxFileBuffer[]=" FatFS System test is OK ! \r\n";
res=f_mount(&fs," 0:" ,1);
                    /* 创建一个工作区，调用初始化函数 */
if(res !=FR_OK)
  printf(" mount ERROR \n\t");
else
  printf(" mount SUCCESS \n\t");
res=f_open(&FileSystemDst," 0:/Demo1. txt",FA_CREATE_
ALWAYS|FA_WRITE); /* 在刚刚开辟的工作区的盘符 0 下打开一个名为 Demo1. txt 的文件，没有则创建文件 */
if(res==FR_OK ){
  printf(" File Open SUCCESS !\n\t");
  res = f_write(&FileSystemDst, TxFileBuffer, sizeof(TxFile-
Buffer), &bw); /* 将缓冲区的数组变量 TxFileBuffer 的内容写到刚刚打开的 Demo1. txt 文件中 */
  if(res)
    printf(" File Write ERROR !\n\t");
  else
    printf(" File Write SUCCESS !\n\t");
  f_close(&FileSystemDst); /* 关闭文件 */
}
else if(res == FR_EXIST)
  printf(" File is already exist \n");
else
  printf(" Don't know the error !\r\n");
```

以上代码实现的功能是在 SD 卡中新建一个文件名为 Demo1. txt 的文本文件，将数据缓冲区 TxFileBuffer 中的内容写入这个文件中。

```
res=f_open(&FileSystemDst," 0:/Demo1. txt", FA_OPEN_
EXISTING | FA_READ);
if(res)
  printf(" File Open ERROR !\n\t");
else
  printf(" File Open SUCCESS !\n\t");
br=1;
for(i=0;i<512;i++)
  FileRxBuffer[i]=0;
res = f_read(&FileSystemDst, FileRxBuffer, sizeof(FileRx-
Buffer),&br);
if(res)
  printf(" File Read ERROR !\n\t");
else
  printf(" File Read SUCCESS !\n\t");
printf(" \r\n %s",FileRxBuffer);
f_close(&FileSystemDst);
```

以上代码实现的功能是读取 SD 卡中文件名为 Demo1. txt 的文本文件，将读取到的数据放在数据缓冲区 FileRxBuffer 中。

5 数据读写验证

把 STM32 单片机写过数据的 SD 卡插入 PC 的 SD 卡插槽查看数据，与单片机写入的数据一致，PC 机端数据如图 5 所示。

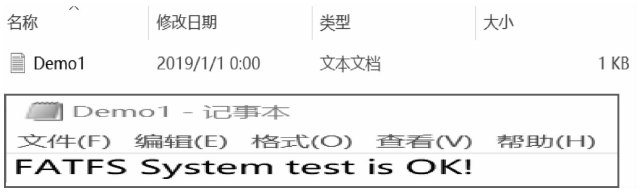

图 5 PC 机端数据

单片机读取 SD 卡的数据放在数据缓冲区 FileRxBuffer 中，通过 IAR 平台查看缓冲区数据如图 6 所示，其数据与文本文件中的数据一致[10]。

图 6 缓冲区数据

结 语

该设计以 STM32 微处理器为控制核心，根据 FatFS 文件系统规范成功实现了基于 SD 卡的一系列操作（如创建、删除、读写等），经测试，该系统稳定可靠，并已成功应用于氢气加气机样机的数据存储系统中。该系统硬件电路简单，软件的可移植性强，非常适用于高速、大容量的数据存储场合。它支持热插拔及数据写保护功能，能正确读写 SanDisk、Kingston、SAMSUNG 等厂商多种容量的 SD 卡，最高读写速度可达 1 MB/s，可满足小型嵌入式系统的应用需求，具有较高的应用价值。

参考文献

[1] ST Microelectronics. STM32F10xxx in - application programming using the SPI[EB/OL]. [2020 - 06]. http://www.st.com.

[2] 刘波文. ARM Cortex - M3 应用开发实例详解[M]. 北京：电子工业出版社，2011.

电子科技大学,2018.
[8] 唐思超. RISC-V 处理器的 C 语言启动代码设计方法[J]. 单片机与嵌入式系统应用,2020(4):14-17.
[9] 谷涛. 轻松学 C#[M]. 北京:电子工业出版社,2013:1-324.
[10] 缑文博. 基于 AVR 单片机的电动汽车蓄电池监测系统的设计[J]. 电子世界,2020(7):198.
[11] 高淑婷. 基于 AVR 单片机的汽车空调控制系统设计[J]. 机械装备研发,2020(11):130.
[12] 刘晓,陈广凯,赵汉青,等. 一种基于单片机串口通信的数据缓存处理方法[J]. 信息通信,2020(4):103-104.
[13] 陈旭辉,杨红云. USB 接口的虚拟多串口通信设备设计[J]. 单片机与嵌入式系统应用,2020(4):18-21.
[14] 黄才权,毕增军,张辉. ZYNQ 的超低时延视频编解码系统设计[J]. 单片机与嵌入式系统应用,2020(2):27-30.

王君(工程师),主要研究方向为预警装备保障。

(责任编辑:薛士然　收稿日期:2020-07-07)

[3] 史胜伟,潘冀宁,孙慧洋. 基于 STM32 的 MicroSD 卡 Fat 文件系统快速实现[J]. 通讯世界,2016(17):81-83.
[4] 李鸿征. 高性能 Micro SD 卡读写器的设计与开发[J]. 焦作大学学报,2017,31(4):69-72.
[5] 颜秋男,胡毅. STM32F103VB 的 SD 卡在应用编程设计[J]. 单片机与嵌入式系统应用,2012(2):36-38,46.
[6] 李敏,侯亚玲,刘颖. 基于 SD 卡的 FAT32 文件系统设计与实现[J]. 物联网技术,2017,7(7):96-98,102.
[7] 王坤,丁红胜. 基于 STM32 的图像编码与采集系统[J]. 电子设计工程,2018,26(5):179-183.
[8] 徐涛,陈赫,卢少微,等. 基于 STM32 的碳纳米纸传感器信号采集系统设计[J]. 仪表技术与传感器,2019(9).
[9] 孙海英,朱晔,罗春,等. 基于 STM32 控制器的设备运行时间自动统计装置设计[J]. 电子测试,2020(2):20-21.
[10] 梁菲惜. 基于 STM32 和 DGUS 液晶屏的随机键盘设计[J]. 电子制作,2019(1):10-11.

汤才刚(工程师),主要研究方向为单片机与嵌入式系统应用。

(责任编辑:薛士然　收稿日期:2020-06-28)

Xilinx 为 5G 无线电大规模部署推出突破性 Zynq RFSoC DFE

赛灵思公司(Xilinx, Inc.)宣布推出 Zynq RFSoC DFE,这是一类全新的具有突破性意义的自适应无线电平台,旨在满足不断演进的 5G NR 无线应用标准。Zynq RFSoC DFE 将硬化的数字前端(DFE)模块与灵活应变的可编程逻辑相结合,为涵盖低、中、高频段频谱的广泛用例打造了高性能、低功耗且经济高效的 5G NR 无线电解决方案。Zynq RFSoC DFE 在采用硬化模块的 ASIC 的成本效益与可编程和自适应 SoC 的灵活性、可扩展性及上市时间优势之间,实现了最佳的技术平衡。

5G 无线电所需的解决方案,不仅要满足广泛部署所提出的带宽、功耗和成本挑战,还必须适应三大关键 5G 用例:增强型移动宽带(eMBB)、大规模机器类通信(mMTC)以及超可靠低时延通信(URLLC)。此外,解决方案必须能够随不断演进的 5G 标准进行扩展,如 OpenRAN (O-RAN)、全新的颠覆性 5G 商业模式。Zynq RFSoC DFE 集成了针对 5G NR 性能与节电要求而硬化的 DFE 应用专用模块,同时还提供了结合可编程自适应逻辑的灵活性,从而为日益发展的 5G 3GPP 和 O-RAN 无线电架构提供了面向未来的解决方案。

赛灵思执行副总裁兼有线与无线业务部总经理 Liam Madden 表示:"为满足 5G 的特殊需求,赛灵思史上首次推出这样一款硬化应用专用 IP 多于自适应逻辑的无线电平台。随着 5G 相关市场需求日益演进,集成式 RF 解决方案也需不断适应未来标准。Zynq RFSoC DFE 在灵活应变能力与固定功能 IP 之间提供了最佳平衡。"

与上一代产品相比,Zynq RFSoC DFE 将单位功耗性能提升高达两倍,并且能够从小蜂窝扩展至大规模 MIMO (mMIMO)宏蜂窝。该解决方案是一款独特的直接 RF 采样平台,能够在所有 FR1 频带和新兴频带(最高可达 7.125 GHz)内实现载波聚合/共享、多模式、多频带 400 MHz 瞬时带宽。当用作毫米波中频收发器时,Zynq RFSoC DFE 可提供高达 1600 MHz 的瞬时带宽。Zynq RFSoC DFE 的架构支持客户绕过或定制硬化的 IP 模块。例如,客户既可以利用支持现有和新兴 GaN Pas 的赛灵思经现场验证的 DPD,也可以插入其自有的独特 DPD IP。

ABI Research 5G 高级研究总监 Dimitris Mavrakis 表示:"随着 5G 商业部署和新用例持续演进,对于整个供应链而言,如何为供应商提供灵活的组件以创建具有成本效益、适应性强且面向未来的设备十分关键。尤其是 OpenRAN,其在这方面的要求更高,灵活的设计对其成功至关重要。Zynq RFSoC DFE 实现了硬化和自适应可编程逻辑之间的平衡,是一款兼具通常 ASIC 才具有的成本优势以及 FPGA 才拥有的设计灵活性和定制化优势的独特产品。"