51单片机与SD卡接口设计

Interface Design for SD Memory Card Based on 51 Single-Chip Microcomputer

姚放吾 曹木莲 卢昭材 丁福舜 (南京邮电大学计算机学院,江苏 南京 210003)

摘 要

介绍51单片机与SD卡的接口设计原理,以及FAT32文件系统的设计和应用程序控制SD卡读写操作的实现。

关键词:单片机,W86L388D,FAT32

Abstract

This paper introduces the design principle to interface single-chip microcomputer with SD memory card, as well as FAT32 file system design and implementation of applications to control SD card read-write operation.

Keywords:single-chip microcomputer,W86L388D,FAT32

近年来,SD存储卡在嵌入式产品中的应用越来越广泛,但SD卡接口一般仅集成在32位ARM处理器中,一般51单片机则由于资源限制没有该接口。因此,如何解决51单片机应用系统存取SD卡大容量数据就显得很有实际意义。本课题使用W86L388D作为单片机与SD卡的接口芯片,采用4线并行方式取代SPI串行方式读写数据,使SD卡的访问速度得到明显提高,并且设计了FAT32文件系统,提高了SD卡的存储效率和兼容性,因此特别适用于需要大容量存储功能的嵌入式仪表和设备中。

1 硬件设计

1.1 SD卡接口芯片简介

单片机与SD卡接口使用了华邦公司的W86L388D芯片。芯片的主要特性包括8/16位CPU总线接口、最高时钟率25MHz、支持SD卡1线或4线数据传输、支持DMA和中断传输模式。由于该芯片内部集成了SD卡操作的全部时序,因此用户只需通过初始化参数配置,然后给相应的寄存器写入控制命令,就可方便的控制SD卡的读写操作。每次命令操作结果的状态,可以通过查询内部的状态寄存器获知,以决定下一步的操作。W86L388D工作电压为3.3V,且采用48脚LQPF小型封装,所以特别适用于要求体积小、功耗低的嵌入式应用场合。

1.2 接口电路设计

电路中的W86L388D使用8位数据总线与单片机连接,它与处理器的接口采用异步模式(XTYP2引脚接地),芯片还支持另一种同步模式(XTYP2引脚接高电平)。在8位总线下,D15用作A0,与A1~A3共同作用选择内部寄存器。SD1~SD6引脚与SD卡的4根数据线,1根命令线和1根时钟线相连。W86L388D的5个GPIO引脚,一个用于SD卡是否插入的检测,一个用于控制MOS管的导通从而控制SD卡电源的接通与断开,两个分别控制两个发光二极管,可用来指示卡是否插入和对卡的写保护指示,一个检测写保护。32KB SRAM用于文件缓冲区,MAX3232是RS-232接口,用于和HOST机通信。

2 软件设计

软件设计主要包括三部分:对SD卡进行底层多块读写的驱动程序设计,中间层FAT32文件系统设计,以及跟上位机(HOST)依据传输协议的通信程序设计。整体软件框架结构如图1所示。

2.1 底层SD卡驱动程序

SD卡有两种总线协议,SD Bus协议和SPI协议。现在绝大部分微控制器都集成SPI接口,所以利用这种方式与SD卡通信相对简单方便,但SPI协议在数据交换时只允许1位数据串行传输,所以速度受到限制。在SD Bus协议下,允许1线到4线数据传输,因而提高了传输速度。但SD总线时序要求严格,如果用软件模拟不仅复杂烦琐,而且可靠性也不高,W86L388D支持SD Bus方式的4线数据传输,并且根据所收到的命令能自动产生相应的SD Bus时序,从而方便用户的使用,提高了系统的性能。

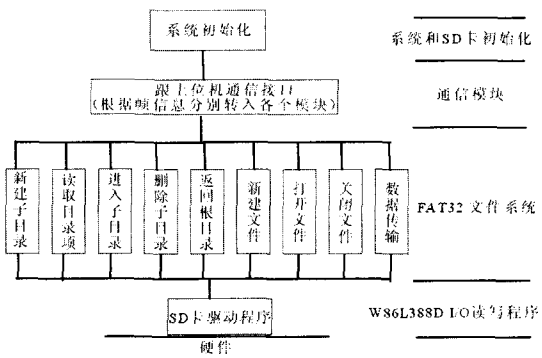

图1 SD卡软件框架结构

与SD卡通信的命令(CMD)和数据(DATA)由一个起始位同步,由一个结束位终止。发给SD卡的命令采用6字节的格式。命令由主机通过CMD线串行发给SD卡,标志SD卡的行为动作,部分命令要求卡返回一个应答信号,应答信号同样是通过CMD线由卡传送给主机的。当主机发送给卡的命令要求有数据交换时,DAT0~3线将进行相应的数据传输。

在访问SD卡的过程中,所有的操作都由主机发起,主机发起的命令(CMD)有两种类型:广播命令和点对点命令。广播命令对系统中所有的卡都有效,只有被选定的卡才能接受点对点命令。主机对SD卡的操作都是经过以下两种模式:卡识别模式和数据传输模式。

卡识别模式:在这个模式中,主机复位系统中所有卡,得到卡的工作电压、系统分配给卡的地址(RCA)等信息。这些操作对系统中所有的卡都有效,且所有的数据传输都是通过CMD线进行的。系统初始化包括:W86L388D的初始化和SD卡的初始化。

W86L388D的初始化:①设置CPU访问W86L388D的总线宽度为8位,且W86L388D工作在模式1;②进行软件复位;③设置系统时钟,使系统工作于低速状态;④检测是否有卡插入,如果没有则关闭卡电源,如果有,则进行下一步卡的初始化。

SD 卡的初始化：①W86L388D 中断允许设置；②利用 ACMD41 命令获取卡操作条件寄存器 OCR 的内容（使用 ACMD41 命令之前先使用 CMD55 命令），获取此卡工作允许的电压范围；③利用 CMD2 命令获取卡识别寄存器 CID 的内容，获取卡的厂商 ID、卡的名称、卡的版本等信息；④利用 CMD 3 命令获取系统分配给卡的地址 RCA；⑤设置系统时钟，使系统工作于高速状态；⑥如果系统中有多张 SD 卡，利用 CMD7 命令选定其中一张卡进行下一步的操作。

数据传输模式：在这个模式中，可以对 SD 卡进行读、写和擦除等操作，包括：单块的读操作，单块的写操作，多块的读操作，多块的写操作。

2.2 文件系统的实现

考虑到目前 SD 卡容量都较大，因此文件系统采用 FAT32 格式设计。

(1)获取指定逻辑盘信息

调用底层驱动程序从第 0 块开始，每次读取一个扇区(512 字节)，如果发现最后结束标志的两个字节是 55AA，并且第 446 (从 0 开始计数)个字节的值是 0X80，则表示此扇区是主引导扇区(MBR)。读出主引导扇区后，根据主引导扇区的第 454~457 (从 0 开始编号)的连续四个字节的值(小端模式存储)，就可以根据这个数值找到系统分区引导记录 BPB 扇区的地址了。在实际操作 SD 卡的过程中，发现用这种方法有时读不出个别 SD 卡的容量和类型信息，具体原因不明。所以修改的算法如下：从零号扇区开始每次读取一个扇区进行搜索，如果发现最后结束标志的两个字节是 55AA，并且此扇区[0]的内容是 0xEB 或 0xE9，则此扇区就是 BPB，否则就读取此扇区[454~457]内容的值，根据这个数值找到系统分区引导记录 BPB 扇区的地址。从而得到逻辑盘的相关信息。具体函数实现如下：

```
Disk_Info *GetDiskInfo()
{      .....
       if(ReceiveMul[0][510]==0x55&&ReceiveMul[0][511]==0xAA){
                  /* 每扇区字节数 */
DiskInfo->BytsPerSec=(ReceiveMul [0][11])|((uint16)ReceiveMul[0]
[12]<<8);
    DiskInfo->SecPerClus = ReceiveMul[0][13]; /* 每簇扇区数 */
      /* FAT 开始扇区号 */
DiskInfo ->FATStartSec =temp +(ReceiveMul [0] [14]) |((uint16)Re-
ceiveMul[0][15]<<8);
     DiskInfo->NumFATs = ReceiveMul[0][16]; /* FAT 表个数 */
                  /* 逻辑盘(卷)占扇区数 */
   DiskInfo ->SecPerDisk =ReceiveMul [0] [32]  | ((uint16)Re-
ceiveMul[0][33] <<8);
   DiskInfo->SecPerDiskl=((uint32)ReceiveMul[0][34] << 16);
    DiskInfo->SecPerDiskl=((uint32)ReceiveMul[0][35] << 24);
                  /* FAT 表占用扇区数 */
DiskInfo ->FATSecCnt =ReceiveMul [0] [36] |((uint16)ReceiveMul [0]
[37]<<8);
DiskInfo ->FATSecCnt |=(((uint32)ReceiveMul [0] [38] <<16) |(uint32)
ReceiveMul[0][39] << 24);
                  /* 根目录开始扇区号 */
   DiskInfo->RootClus =ReceiveMul [0][44]  |(ReceiveMul [0][45]
<< 8);
  DiskInfo ->RootClus |=((uint32)ReceiveMul [0] [46] <<16) |(uint32)
(ReceiveMul[0][47] <<24));
                  /* 数据区开始扇区号 */
DiskInfo->DataStartSec=DiskInfo->FATStartSec +DiskInfo->FAT-
SecCnt*DiskInfo->NumFATs;
temp1 = DiskInfo->SecPerDisk - DiskInfo->DataStartSec;
temp1 = temp1 / DiskInfo->SecPerClus;
DiskInfo->ClusPerData = temp1;
return DiskInfo;
}
.....
}
```

(2)目录相关操作实现

当接收到上位机的信息帧，根据帧格式可判定对目录的操作包括：新建子目录、读目录、进入子目录、删除子目录、返回到根目录。

当操作为新建子目录时，根据目录名，首先在父目录项下依次搜索子目录项，如果发现要创建的目录已经存在，则向上位机返回子目录已存在的信息，返回空指针结束。否则搜索 FAT1 寻找一空闲簇项，将该簇的数据区清空作为该新建子目录的 FDT 表，然后在父目录的 FDT 表项中找到一空闲的 FDT 表项，将子目录的目录名，属性，首簇号等相关目录项信息写入到相应的位置。

当操作为读目录时，首先获取父目录的首簇号，将该簇对应的父目录的 FDT 表项内容发送给上位机，接着将父目录的首簇号的 FAT 链表对应的簇的 FDT 表项内容发送给上位机，直到链表结束。在此期间，每次发送一帧信息都要检测应答帧有无错误，如有错误则将缓冲区的数据重发给上位机，如果超过三次还有错误则退出操作失败。

进入子目录和删除子目录操作，都要先在父目录下的 FDT 表项中搜索，如果子目录不存在则向上位机返回操作失败信息。如果找到子目录项，对于进入子目录操作而言只要将当前的父目录项的首簇号变为子目录项对应的首簇号即可，对于删除操作，则获得子目录对应的首簇号，判定子目录是否为空，如果为空，则将父目录项下在其对应的 FDT 表的第一个字节变为 0XE5 即可。如果要删除的子目录不空则不允许删除，返回失败信息。

(3)文件相关操作实现

文件操作包括：新建文件、打开文件、关闭文件和数据传输。

进入新建文件操作，首先在父目录下搜索文件是否存在，如果不存在，则在 FAT 表中找一空闲簇号作为该文件的首簇号并作为一个全局变量先保存，在接收完上位机发送的数据帧之后，只有接收到关闭文件帧，才更新 FAT 表和 FDT 表。

进入打开文件操作，首先接收上位机传送的控制帧信息，从中获取要打开文件的文件名，然后在文件目录表中搜索该文件是否存在。如果文件不存在则出错，函数返回空指针操作结束。如果文件存在，则从 FDT 表中获得文件的首簇号和文件大小等相关信息。根据簇链依次读取每个簇的数据到缓冲区，当缓冲一定存储容量后将数据发送给上位机。如此循环操作直至把全部数据发送到上位机。

关闭文件操作表示上位机的数据已全部发送完毕，此时要更新 FAT 表和 FDT 表中的相关内容。先查询文件目录表是否已满，如果没有剩余的目录项，则向上位机发送 SD 卡容量已满的出错信息。否则将该文件的文件名，大小等相关信息写入空闲的文件目录项，接着再用储存在数据缓冲区的簇链更新 FAT 表。

数据传输操作时，先将接收到的数据存放在 SRAM 缓冲区，缓冲区的大小用宏定义实现，这样可以根据 SRAM 的容量随时更改缓冲区的大小以便于移植。当数据超过缓冲区容量时，停止接收上位机数据，将缓冲区中的数据写入 SD 卡，这样循环反复直到文件数据全部写入 SD 卡。

(下转第 68 页)

来完成。模糊控制程序分为以下四步进行。①根据测量的去污力，计算去污力的偏差和偏差的变化；②根据去污力的偏差和偏差的变化，模糊化；③查询模糊控制表，可得出控制的输出量；④将控制输出量乘以比例因子，施与控制对象。程序流程如图5所示。

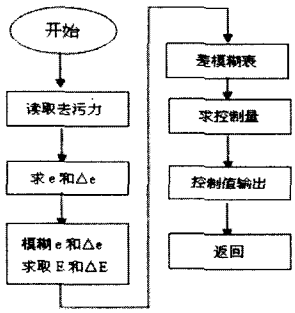

图5 程序流程图

全部的控制规则由表1给出，模糊控制规则一共十五条规则。

表1 模糊控制规则

去污力 / 材料 / 污渍量	面料	纱线	纤维原料
很多	很大	很大	大
多	大	大	适中
普通	适中	适中	适中
少	小	小	很小
很少	小	很小	很小

污渍量的隶属函数如图6所示。

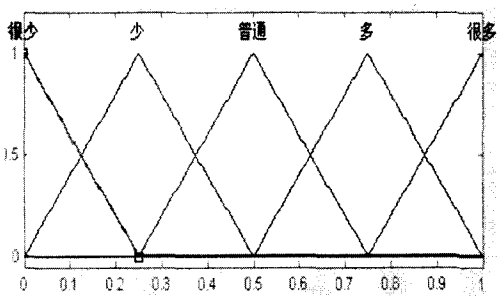

图6 污渍量的隶属函数

衣服材料的隶属函数如图7所示。

实验仿真图如图8所示。

3 结束语

利用Matlab来实现模糊控制器的仿真分析非常简便，为研究模糊控制理论、设计模糊控制器提供了有力的工具。近年来，模糊控制系统的研究取得了很大的进展，特别是模糊控制器的结构分析，模糊系统的万能逼近特性，模糊状态方程及稳定性分析，软计算技术等；同时，模糊逻辑在软件硬件方面也取得了飞速的发展。但模糊系统理论仍存在一定的问题，主要有以下不足之处：尽管模糊系统的万能逼近特性已被证明，但只是一个存在性定理。实际中，对于一般的未知系统，如何找到一个合理的模糊逼近器，尚无确定的方法。这些问题都有待于进一步研究。

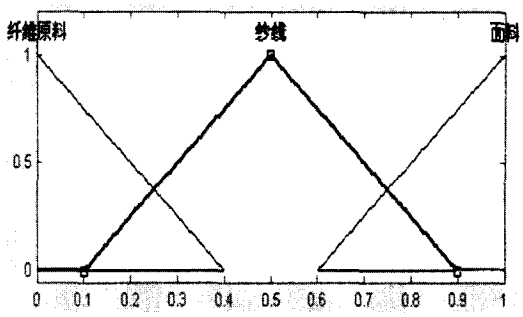

图7 衣服材料的隶属函数

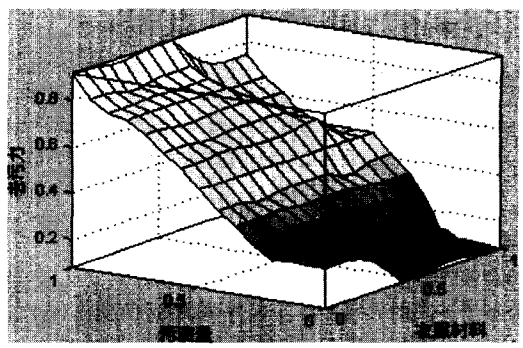

图8 实验仿真图

参考文献

[1]李士勇.模糊控制神经理论和智能控制论[M].哈尔滨：哈尔滨工业大学出版社，1996：283-294

[2]ProcykT J,Mamdani E H.A linguistic self_organizing process controller.Automaqica,1979,5(1)：15-30

[3]Maurlcio Figueiredo,Femando Comide.Design of FuzzySystems Using Neuro_fuzzy Networks [J].IEEE Transactions On Neural Networks,1999 10(4)：815-827

[4]梁海瑛，张泾周.规则自适应模糊PID数字电压调节器设计[J].计算机工程与设计，2008，29(3)：665-686

[收稿日期：2008.5.22]

(上接第61页)

2.3 与上位机(HOST)的通信协议规范

HOST机通过串口与嵌入式SD卡读写器连接，通信程序的主要功能是使用约定的通信协议与应用系统通信，完成数据的可靠双向传送，并实现对SD卡的读写操作。通信协议使用了控制帧、数据帧、应答帧方式，数据/控制帧长为515Byte，其中，头两个字节定义帧的功能，数据512Byte，最后一个字节为累加校验和。应答帧长度为8Byte，无校验，表示应答状态。

作为HOST机，只要编写一个API函数，按照约定的帧格式发送命令和数据，即可实现对SD卡的读写操作。

3 结束语

本设计通过桥接芯片采用SD Bus模式访问SD卡，不但访问速度显著提高，而且读写过程稳定可靠，是低端嵌入式系统更新与升级较为方便的设计方案。由于程序使用C语言编写，因此具有很好的移植性，可以广泛地应用于嵌入式产品的开发中。

参考文献

[1]Microsoft Extensible Firmware Initiative FAT32 File System Specification.Version 1.03,December 6,2000.Microsoft Corporation

[2]W86L388D Winbond Host Interface SD/MMC Memory Card Bridge.Version 1.0,May 17,2005

[3]W86L388 Programming Information

[4]周立功，等.ARM嵌入式系统软件开发实例(一)[M].北京：北京航空航天大学出版社，2005

[5]宋群生，宋亚琼.硬盘扇区读写技术-修复硬盘与恢复文件[M].北京：机械工业出版社，2004

[收稿日期：2008.4.16]