

文章编号:1001-5485(2009)12-0110-05

基于 STM32 的励磁系统录波存储器的实现

张云,熊杰,宋宴明

(武汉市陆水自动控制技术有限公司,武汉 430010)

摘要:传统使用 PC 机现场对励磁系统进行录波,调试正常后一般就不再进行后续录波,对于后期事故的排除及产品的更新都不利。介绍了一种采用基于 ARM Cortex-M3 内核的 STM32 微控制器与 SD 半导体快速闪记忆器(SD 卡)的微机录波存储器的设计方法。并从硬件和软件两方面具体论述了该存储器的实现。该系统实现了自动对励磁系统的数据进行快速采集、分析和存储的长时间录波。

关键词:ARM Cortex-M3;STM32;SD 卡;实时录波;串口通讯;励磁系统

中图分类号:TV734.4 **文献标识码:**A

1 概述

励磁控制器是同步发电机励磁系统的重要组成部分,它对同步发电机乃至电力系统的安全稳定运行有着重要影响。随着电力系统规模的不断增大,系统结构和运行方式的日趋复杂,对同步电机励磁控制器运行的可靠性就有更严格的要求,也就提高了对励磁系统的数据采集及储存性能要求。励磁系统的可靠性与历史数据有着密切的联系,通过对历史数据的研究,能更好地了解励磁系统的工作情况,从而设计新的方案提高励磁系统的稳定性和可靠性。本文介绍采用基于 ARM Cortex-M3 内核的 STM32 微控制器作为主控嵌入式芯片的微机励磁系统录波存储器的设计方法。

2 硬件设计

数据采集和存储以一片 STM32F103RC 为主体,通过一片 MAX3232 完成串口数据的通讯。STM32 微处理器采用 SPI 的模式与 SD 卡连接,实现对 SD 卡的读写。

2.1 STM32F103RC

STM32 系列 32 位闪存微控制器基于突破性的 ARM CortexTM-M3 内核,这是一款专为嵌入式应用而开发的内核。STM32 系列产品得益于 Cortex-M3 在架构上进行的多项改进,包括提升性能的同时又

提高了代码密度的 Thumb-2 指令集,大幅度提高中断响应,而且所有新功能都同时具有业界最优的功耗水平。STM32 系列产品为 MCU 用户提供了新的自由度。它提供了一个完整的 32 位产品系列,在结合了高性能、低功耗和低电压特性的同时,保持了高度的集成性能和简易的开发特性。

STM32F103RC 是属于 STM32F103xx 系列中的一种微控制芯片。片上带 256 kB Flash ROM 和 48 kB SRAM,拥有一个实时时钟、3 个 SPI 接口、SDIO 高速接口、5 个 USART 接口和多个计时器。

2.2 串口通讯芯片 ST3232

MAX3232 作为一款串口通讯芯片,具有低功耗、高数据速率、增强型 ESD 数据保护等特性,在现代电子技术中有着很广泛的应用。MAX3232 包括 2 个发送器和 2 个接收器,能同时与两路调节器进行通讯。它和 STM32 接口电路如图 1。

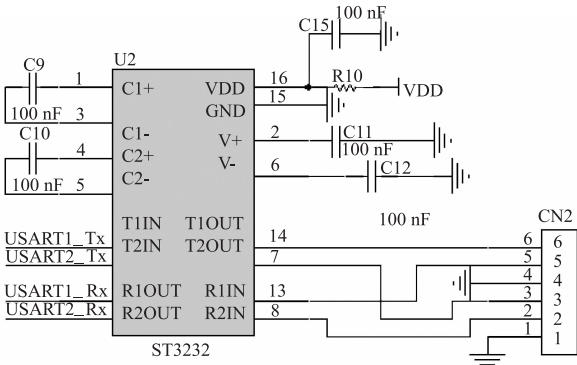


图1 MAX3232 接口电路图
Fig. 1 MAX 3232 Interface circuit

2.3 SD 卡接口

SD 卡连接器使用 SPI 总线与 STM32 处理器连接,如图 2 所示。

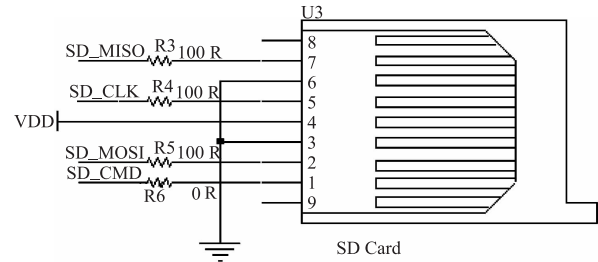


图 2 SD 卡接口电路图
Fig. 2 SD card interface circuit

SD 卡主要引脚和功能如下：

CLK,时钟信号,每个时钟周期传输一个命令或数据位,频率可在 0 ~ 25 MHz 之间变化,SD 卡的总线管理器可以不受任何限制的自由产生 0 ~ 25 MHz 的频率；

CMD,双向命令和回复线,命令是一次主机到从卡操作的开始,命令可以是主机到单卡寻址,也可以是到所有卡,回复是对之前命令的回答,回复可以来自单卡或所有卡；

DAT0-3,数据线,数据可以从卡传向主机也可以从主机传向卡。

2.4 系统结构框图

STM32 微控制器起到连接 MAX3232 和 SD 卡的作用,从而实现励磁调机器串口数据向 SD 存储的目的。励磁调节器的实时数据以数据包的形式通过串口发送出来,其中包括包头、数据和数据包编号。STM32 微处理器采用 DMA 的形式,并通过 DMA 中断实现对接收缓冲区的管理,从而实现快速对串口数据接收。STM32 同时对缓冲区的数据通过 SPI 总线传送到 SD 卡上,SD 卡实现了对历史数据的存储。数据采集和存储的结构框图见图 3。

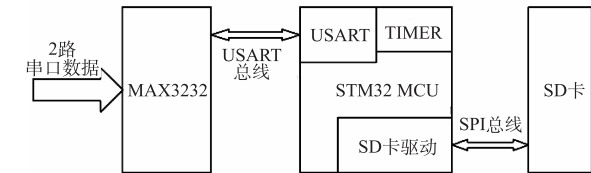


图 3 数据采集和存储的结构框图
Fig. 3 Data collection and storage block diagram

3 软件设计

在本系统中,软件设计分为 4 个部分:固件设计、SD 卡驱动设计、文件系统设计、实时时钟设计。

3.1 固件设计

固件需要完成的任务包括 2 个方面,一是与串口的数据交互,二是数据缓冲区的管理。

(1) 与串口数据的交互:STM32 处理器通过 MAX3232 与励磁调节器的串口进行数据交互。此系统外接 2 个励磁调节器,每个励磁调节器的实时数据通过串口以数据包的形式发送过来。由于励磁调节器分为主用和备用,因此只需对主用励磁调节器的数据进行采集整理和存储。在一开始需对串口数据进行分析,判断主用和备用。串口通讯的波特率为 115 200,8 位数据位,无奇偶校验位和 1 位停止位,无硬件流控制。

(2) 数据缓冲区的管理:在本系统中开辟了 2 个数据缓冲区,这 2 个缓冲区为串口接收缓冲区,同时也是 SD 卡数据存储数据的来源。由于不能同时对同一缓冲区进行读写,因此串口和 SD 卡只能在不同的时间访问缓冲区。在串口接收数据到一个缓冲区时,将另一个缓冲区的数据写入到 SD 卡上。当然写 SD 卡的时间要小于串口接收数据直到数据缓冲区满的时间。这就需要有一个对缓冲区进行管理的机制。设计的算法大概如下:①配置缓冲区 1 为串口接收缓冲区,设置缓冲区 1 满中断;②处理缓冲区 1 的满中断,配置缓冲区 2 为串口接收缓冲区,设置缓冲区 2 满中断;③将缓冲区 1 的数据进行处理并及时写入 SD 卡上;④处理缓冲区 2 的满中断,配置缓冲区 1 为串口接收缓冲区,设置缓冲区 2 满中断;⑤将缓冲区②的数据进行处理并及时写入 SD 卡上;⑥返回 2 步骤。如此达到对数据的高效采集整理并存储。

3.2 SD 卡驱动设计

由于 SD 卡是以命令形式来控制读写等操作的,也可根据命令对多块或单块进行读写操作。在 SPI 模式下其命令由 6 个字节构成,其中高位在前。SD 卡命令的格式如表 1 所示。

表 1 SD 卡命令的格式
Table 1 SD card command format

Byte 1				Byte2-5			Byte 6	
7	6	5	0	31	0	7	0	
0	1	Command		Command		Argument	CRC	1

SD 卡上电后会自动初始化,通过给卡发送 CMD0 也可以复位卡。卡收到 SEND_RCA(CMD3)命令后即进入数据传输模式。STM32 处理器识别到卡后也进入此模式。进入数据传输模式后,主机先不停地发送 SEND_CSD(CMD9)命令获取卡的 CSD 信息。SET_DSR(CMD4)用于设置卡的 DSR


```
BLOCK) 来读取一个数据块 */
MSD_SendCmd ( MSD_READ_SINGLE_BLOCK,
ReadAddr, 0xFF);
if (! MSD_GetResponse ( MSD_RESPONSE_NO_
ERROR)) /* 检测 SD 卡是否响应读命令 */
{
/* 寻找数据的起始标识 */
if (! MSD_GetResponse ( MSD_START_DATA_
SINGLE_BLOCK_READ))
{
for (i = 0; i < NumByteToRead; i + +) /*
SD 卡读数据块的数据 */
{
* pBuffer = MSD_ReadByte(); /* 保存接
收到得数据 */
pBuffer + +; /* 指针指向下一个要读数据
的地址 */
}
MSD_ReadByte(); /* 获得 CRC 校验字节 */
MSD_ReadByte(); /* 度数据成功 */
rvalue = MSD_RESPONSE_NO_ERROR;
}
}
MSD_CS_HIGH(); /* 将 SD 卡选择信号拉高
*/
MSD_WriteByte(DUMMY); /* 发送一个 dummy
字节: 延迟 8 时钟脉冲 */
return rvalue; /* 返回 SD 卡的响应 */
}
```

3.3 文件系统的设计

因为要用 SPI 操作 SD 卡,所以移植了一个免费开源的 FAT 文件系统, FatFs, 实现了读写文件的操作。

其中底层的 SPI 总线对 SD 卡的操作在 SPI_SD_driver. c/h 中,而 FatFs 的移植文件 diskio. c 中对磁盘的操作函数中将调用底层的操作函数。下面是一些底层操作函数(具体如何实现就不再详细说明):

```
u8 MSD_Init( void); /* SD 卡初始化
u8 MSD_WriteBlock ( const u8 * pBuffer, u32
WriteAddr, u16 NumByteToWrite); /*写一个扇区
u8 MSD_ReadBlock( u8 * pBuffer, u32 ReadAd-
dr, u16 NumByteToRead); /*读一个扇区
u8 MSD_WriteBuffer ( const u8 * pBuffer, u32
WriteAddr, u32 NumByteToWrite); /*SD 卡写数据
u8 MSD_ReadBuffer ( u8 * pBuffer, u32
```

```
ReadAddr, u32 NumByteToRead); /*读多个扇区
u8 MSD_GetCSDRegister ( sMSD_CSD * MSD_
csd); /*读 SD 卡 CSD
u8 MSD_GetCIDRegister ( sMSD_CID * MSD_
cid); /*读 SD 卡 CID
void Get_Medium_Characteristics ( void); /* 取
SD 卡容量
void MSD_SendCmd ( u8 Cmd, u32 Arg, u8
Crc); /*SD 卡发送一个命令
void MSD_WriteByte(u8 byte); /*SPI 总线写一
个字节
u8 MSD_ReadByte( void); /*SPI 总线读一个字
节
void SPI_Config(u16 BaudRatePrescaler); /*SPI
配置
```

在 disk 初始化中,我们调用 MSD_SD_driver. c 中的 SD 卡初始化函数。经测试,在 STM32 的 36MSPI 时钟下,读文件的速度在每秒 300K 以上,写文件也有 280 多 K 的速度。

3.4 实时时钟系统的设计

STM32 有一个带校准功能的 32.768 kHz RTC 振荡器,使用外部备用电池为 RTC 和后备寄存器供电。在系统掉电的时候,仍然保持时钟功能和后备寄存器的使用。第一次使用时,在后备寄存器中写入 0x5A5A,这样再次启动时,先读入后备寄存器中的数据,如果为 0X5A5A,就不需要对时钟进行重新配置了。实时时钟具有一组连续运行的计数器,可以通过适当的软件提供日历时钟功能。RTC 中保存的时间格式,是 UNIX 时间戳格式的,即一个 32 bit 的变量。这里采用的是 tm 数据类型。结构如下:

```
struct tm
{
int tm_sec; /* 秒,0 ~ 60
int tm_min; /* 分,0 ~ 59
int tm_hour; /* 时, 0 ~ 23
int tm_mday; /* 日, 1 ~ 31
int tm_mon; /* 月, 0 ~ 11
int tm_year; /* 年,从 1900 年开始计算
int tm_wday; /* 星期, 0 ~ 6
...
}
```

其中 wday,yday 可以自动产生,软件直接读取,mon 的取值为 0 ~ 11,tm_year 在 time. h 库中定义为 1900 年起的年份,即 2008 年应表示为 2008 - 1900 = 108。这种表示方法对用户来说不是十分友好,与

现实有较大差异。所以在本系统中,屏蔽了这种差异。即外部调用本文件的函数时,tm 结构体类型的日期,tm_year 即为 2008。

成员函数说明:

```
struct tm Time_ConvUnixToCalendar(time_t t);
```

输入一个 Unix 时间戳 (time_t), 返回 Calendar 格式日期

```
time_t Time_ConvCalendarToUnix(struct tm t);
```

输入一个 Calendar 格式日期, 返回 Unix 时间戳 (time_t)

```
time_t Time_GetUnixTime(void);
```

从 RTC 取当前时间的 Unix 时间戳值

```
struct tm Time_GetCalendarTime(void);
```

从 RTC 取当前时间的日历时间

```
void Time_SetUnixTime(time_t);
```

输入 UNIX 时间戳格式时间, 设置为当前 RTC 时间

```
void Time_SetCalendarTime(struct tm t);
```

输入 Calendar 格式时间, 设置为当前 RTC 时间

外部调用时, 只需设置一个 Calendar 格式的日期变量即可。其他功能以参考 ansi - c 库相关资料

等, 这里就不做详细说明。

4 结 语

经过长期试验表明, 该系统基本实现了对励磁系统实时数据的快速采集、分析和存储的长时间录波功能。在 2G 的 SD 卡上总是保留了最新的 1 000 个文件, 每个文件的大小为 1.6 M。每 1 min 创建一个文件, 也就是说 SD 卡上保留了最近 16 h 的录波数据。通过读卡器即可方便地在 PC 机上读出 SD 卡上的数据。该方案虽是对微机励磁系统进行的实时数据录波, 亦可用于其他需要对实时数据进行快速采集、分析、存储和长时间录波的微机控制系统。

参考文献:

[1] 李 宁. 基于 MDK 的 STM32 处理器开发应用[M]. 北京: 北京航空航天大学出版社, 2008.

[2] 王永红, 徐 炜, 郝立平. STM32 系列 ARM Cortex-M3 微控制器原理与实践[M]. 北京: 北京航空航天大学出版社, 2008.

(编辑: 周晓雁)

Recording and Memory of Excitation System
Based on STM32 MCU

ZHANG Yun, XIONG Jie, SONG Yan-ming

(Wuhan Lushui Automatic Control Technology Limited Liability Company, Wuhan 430010, China)

Abstract: The PC to carry on recording and memory is used for the excitation system records in tradition methods, and after debugged in good order, the following records are no longer carried on , it is disadvantage to the later accident elimination and the product renewal. This system can conduct the recording automatically, gathering, analysis and memory . This article elaborates this memory's realization from both hardware and software designs. The system enables automatically rapid collection, analysis and storage of excitation system data in long time.

Key words: ARM Cortex-M3; STM32; SD card; real-time record; com communication; excitation system



2009 年第 26 卷, 第 11 期, P60, 表 1 中的 c, φ 值有误, 现具体更正如下:

原表

表 1 膨胀土物理力学参数

Table 1 Physical and mechanical parameters of an expansive soil

密度 ρ / ($\text{kg} \cdot \text{m}^{-3}$)	变形 模量 E / MPa	泊松 比 μ	未风化层 抗剪强度		风化层天然 抗剪强度		风化层饱和 抗剪强度	
			φ / ($^{\circ}$)	c / kPa	φ / ($^{\circ}$)	c / kPa	φ / ($^{\circ}$)	c / kPa
2 300	23	0.33	84.9	45.0	48.7	27.2	13.7	22.2

更正后的表

表 1 膨胀土物理力学参数

Table 1 Physical and mechanical parameters of an expansive soil

密度 ρ / ($\text{kg} \cdot \text{m}^{-3}$)	变形 模量 E / MPa	泊松 比 μ	未风化层 抗剪强度		风化层天然 抗剪强度		风化层饱和 抗剪强度	
			c / kPa	φ / ($^{\circ}$)	c / kPa	φ / ($^{\circ}$)	c / kPa	φ / ($^{\circ}$)
2 300	23	0.33	84.9	45.0	48.7	27.2	13.7	22.2