

文章编号:1001-9081(2007)01-0254-03

## 基于 ARM 的 USB 接口数据记录器的设计

王 安<sup>1</sup>, 钱晓亮<sup>1</sup>, 樊文侠<sup>2</sup>

(1. 西北工业大学 自动化学院, 陕西 西安 710072;

2. 西安工业大学 电子信息工程学院, 陕西 西安 710032)

(qxl\_sunshine@163.com)

**摘 要:**以 PHILIPS 公司的 ARM7TDMI 处理器 LPC2214 为核心,设计并实现了一种容量为 128MB 的数据记录器。该记录器是一种用于记录工控设备之间交换数据的自动记录设备,其目的是为了辅助分析工控系统的工作情况。实际使用表明,该记录器设计方案可行,且与传统的数据记录器相比:高达 128MB 的容量,使得通过 RS-422 连续记录数据的时间大幅提高;使用 USB 接口,则突破了以往数据记录器使用串口上传数据速度慢的限制,缩短了传输时间。

**关键词:**数据记录器;工控系统;USB;RS-422

**中图分类号:** TP333 **文献标识码:** A

## Design of USB interface for data recorder based on ARM

WANG An<sup>1</sup>, QIAN Xiao-liang<sup>1</sup>, FAN Wen-xia<sup>2</sup>

(1. College of Automation, Northwestern Polytechnical University, Xi'an Shaanxi 710072, China;

2. College of Electronic Information Engineering, Xi'an Technological University, Xi'an Shaanxi 710032, China)

**Abstract:** A kind of design and application for a 128MB memory data recorder based on LPC2214 of Philips was introduced, which could record the exchange data between the industrial control equipments and help the engineers to analyze the working state of the industrial control system. Practical application indicates the feasibility of this design scheme. Compared with the traditional data recorder, its 128MB memory make the recording time through RS-422 increase greatly. Instead of serial port, using USB interface can decrease the transmission time evidently.

**Key words:** data recorder; industrial control system; USB; RS-422

一些工控系统各组件之间通过 RS-422 来交换数据,通过监测 RS-422 通信的数据可以帮助分析工控系统的工作情况,这为工作人员排除系统故障提供了一个依据。使用 PC 机并安装串口调试助手这样的软件虽然也可以达到上述要求,但在某些工业现场,因为空间较小、工作环境恶劣等因素,不适合工作人员长时间驻留,所以设计一个体积小、密封的 RS-422 接口数据记录器来自动记录工控设备之间的交换数据具有重要意义。

本文所论述的数据记录器通过 RS-422 接口接收工控设备的通讯数据,再通过 USB 接口把记录器中的数据上传到 PC 中并以二进制文件的形式保存,该二进制文件即可作为辅助分析系统故障所用。

### 1 仪器的总体框架

如图 1 所示,一个完整的数据记录器需要由 PC 机中的 USB 驱动程序及应用软件和密封在 120cm × 80cm × 55cm 大小铝盒中的数据记录器共同组成。记录器的工作过程如下:

(1) 通过 RS-422 接口接受数据。此时要求外接 +5V 电源,USB 接口断开,ARM 每接收到一定量的数据就把它存入 FLASH 中。

(2) 通过 USB 接口上传数据。USB 总线供电,RS-422 接口断开,使用 USB 的上位机管理软件在 PC 中操作。首先,向下位机发“上传”命令使 ARM 将 FLASH 中的数据通过 USB 接口电路全部上传。其次,打开本次上传的数据文件,或是在当前目录下单独打开某个数据文件,进行数据分析。最后,向

下位机发“清除”命令使 ARM 将 FLASH 全片擦除,准备下一次使用。

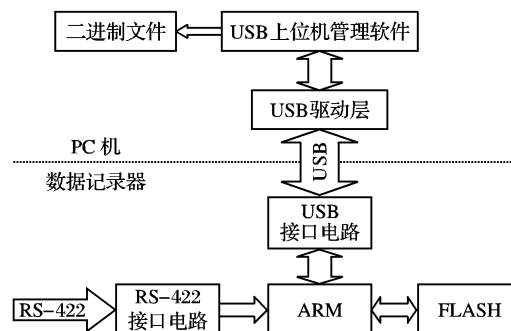


图 1 记录器的总框图

### 2 硬件层设计

#### 2.1 仪器主要器件的选型

处理器: 选用 PHILIPS 公司的 ARM7TDMI 处理器 LPC2214,主要优点为:① 运行速度快;② 功耗低;③ 多达 76 个 GPIO;④ 内置串行通信接口;⑤ 片内有 16kB 的 SRAM 和 256kB 的 FLASH;⑥ 工作温度范围为 -40℃ ~ 125℃,满足数据记录器的工作条件;⑦ 相关资料丰富,使用 C 语言编程十分方便,易于开发<sup>[1]</sup>。

FLASH:由于记录器对存储容量要求较高,从而选用 SAMSUNG 公司的 NAND FLASH K9F1G08U0M-PIB0,其实际的存储容量高达 128MB + 4MB,其中的 4MB 为备用空间;可

收稿日期:2006-07-26;修订日期:2006-09-24

作者简介:王安(1959-),男,山东蓬莱人,副教授,主要研究方向:智能仪器; 钱晓亮(1982-),男,河南孟州人,硕士研究生,主要研究方向:嵌入式系统; 樊文侠(1956-),女,山西运城人,教授,主要研究方向:计算机测控、智能仪器、线性系统。



靠性好,可以编程/擦除 10 万次,数据可以保存 10 年;3.3V 供电,功耗低,可以和 LPC2214 等芯片使用同一个 3.3V 稳压源,简化了电源电路的设计;  $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$  的工作温度范围满足数据记录器的工作条件。

RS-422 接口:由于记录器要求的 RS-422 输入波特率较高,故选用 MAXIM 公司的 MAX3485ESA,其支持最高传输速率 10Mbps,同时对低速无限制,而且它的半双工通讯(记录器只接受不发送)也符合设计要求。此外,它是 3.3V 供电,与 ARM 供电兼容,工作温度范围为  $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ ,满足数据记录器的工作条件。

USB 接口: 选用 PHILIPS 公司的 USB 设备端接口芯片 PDIUSB12。PDIUSB12 的片缘好、相关的开发资料较多是选用它的一个重要因素。此外, 它还具有以下一些特性:

- ① 符合 USB1.1 版规范;
- ② 可与 ARM 按照总线方式连接;
- ③ 主端点具有双缓冲配置增加了数据吞吐量;
- ④ 可通过软件控制与 USB 的连接;
- ⑤ 可支持 3.3V 和 5V 两种电源供电;
- ⑥ 工作温度范围为  $-40^{\circ}\text{C} \sim 85^{\circ}\text{C}$ , 满足数据记录器的工作条件;
- ⑦ 具有良好 EMI 特性的总线供电能力<sup>[5]</sup>。

## 2.2 硬件电路设计

数据记录器的硬件部分结构如图 1 的下位机部分所示, 主要由 ARM 最小系统, 用于存放记录数据的片外 FLASH 存储电路、RS-422 接口电路、USB 接口电路这四部分组成。ARM 最小系统是标准设计, RS-422 接口电路比较简单, 本文重点论述一下 FLASH 和 USB 部分的电路。

### (1) FLASH 存储电路

图 2 所示即为 K9F1G08U0M 与 LPC2214 的连接电路图。对电路图的几点说明:① FLASH 的片选不与 ARM 的片选直接相连,而是通过 I/O 口控制,上电后直接将其拉低,使其一直有效;② 不使用写保护和上电自动读取功能;③ 通过对 7 脚外接上拉电阻使 FLASH 平时一直处于就绪状态,并通过 I/O 口来查询 FLASH 的忙状态。

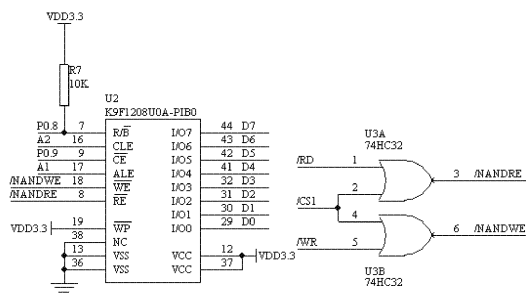


图 2 FLASH 存储电路

## (2) USB 接口电路

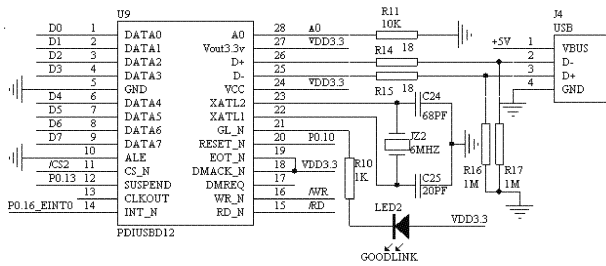


图3 USB 接口电路

如图 3 所示, PDIUSBD12 的硬件接口和外部存储器接口很相似。DATA0 ~ DATA7 与 LPC2214 的 D0 ~ D7 直接相连, 读写引脚也是如此, 片选引脚与 LPC2214 的 BANK2 片选引脚相连, A0 与 LPC2214 的地址线 A0 相连, 用于区分数据地址和命令地址。这样, PDIUSBD12 就相当于 LPC2214 的一片外部 RAM 了, 还有不同于一般外部存储器的中断输出信号

以及复位输入和挂起输出信号,此外,D+/D-线上要串接 $18\Omega$ 的匹配电阻<sup>[4]</sup>。

### 3 软件层设计

软件层设计包含上位机的 USB 驱动及应用程序设计和下位机的硬件程序设计两部分。其中 USB 的固件编程是整个软件设计的核心。

### 3.1 下位机主程序流程

如图4所示,由于RS-422串口通信和USB通信全部采用中断方式处理,所以下位机的主程序主要由两个循环组成,分别用于等待串口中断和USB中断。上电后,先进入RS-422中断等待循环,准备接收数据,直到有USB的setup包出现才退出循环,转入USB中断等待循环,同时轮询上位机发送的命令标志,并做相应处理。

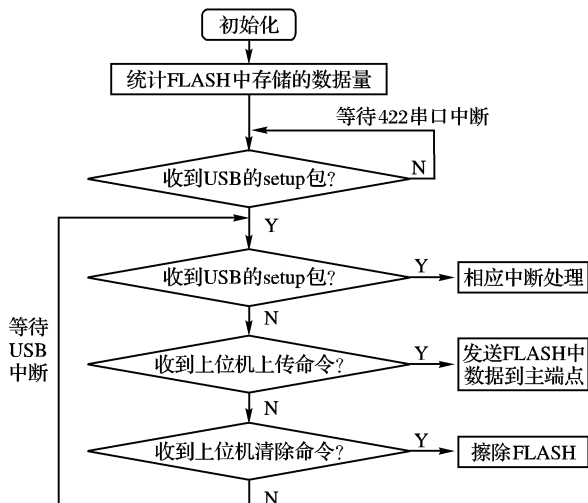


图4 下位机主程序流程

### 3.2 USB 的固件编程

固件设计的目标就是使 PDIUSB12 在 USB 上达到最大的传输速率。外围设备都可使用 PDIUSB12 在 USB 上传输数据,这些设备的 CPU 要忙于处理许多设备控制和数据以及图像处理等任务。PDIUSB12 的固件设计成完全的中断驱动,当 CPU 处理前台任务时,USB 的传输可在后台进行,这就确保了最佳的传输速率和更好的软件结构,同时简化了编程和调试<sup>[5]</sup>。

后台 ISR(中断服务程序)和前台主程序循环之间的数据交换通过事件标志和数据缓冲区来实现。以控制端点在数据包处理时的情况为例:ISR 接收和保存数据缓冲区中的控制传输,并设置相应的标志寄存器,主循环向协议处理程序发出请求。由于所有的标准器件、级别和厂商请求都是在协议处理程序中进行处理,ISR 得以保持它的效率,而且一旦增加新的请求只需要在协议层进行修改<sup>[5]</sup>。

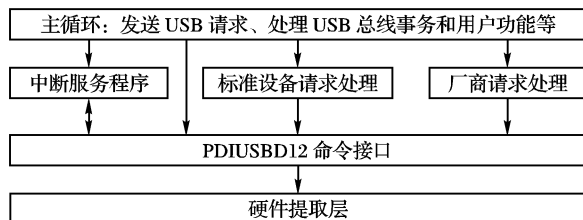


图5 固件结构 and 数据流向

图 5 为 USB 固件程序的结构框架。

1) 硬件提取层:对微处理器的 I/O 口、数据总线等硬件进行操作;

2) PDIUSB12 命令接口:对 PDIUSB12 器件进行操作的模块子程序集;



3) 中断服务程序:当 PDIUSB12 向微控制器发出中断请求时,读取 PDIUSB12 的中断传输来的数据,设定事件标志,交由主循环程序来处理;

4) 标准请求处理程序:对 USB 的标准设备请求进行处理;

5) 厂商请求处理程序:对用户添加的厂商请求进行处理;

6) 主循环程序:发送 USB 请求、处理 USB 总线事件和用户功能处理等。

### 3.3 上位机程序设计

上位机的程序主要包含 USB 的驱动程序及应用程序两部分。其中驱动程序部分直接使用周立功单片机公司提供的驱动程序,在 Windows 2000 或 Windows XP 下安装后即可使用。应用程序部分则调用周立功单片机公司提供的动态链接库 EasyUSB.dll 来读写端口的数据<sup>[5]</sup>。

如图 6 所示,点击“上传”按钮,PC 在与下位机握手后就会读取端口 2(对应 D12 的主端口)的数据,并以二进制文件的形式保存,接受完后会弹出对话框提示接受完毕,否则会提示握手不正确。点击“清除”按钮,向下位机发送清除命令,下位机完成后会回复一个清除完毕命令,PC 接收到这个命令后会弹出对话框提示清除完毕,否则会提示清除失败。点击“打开文件”按钮,会把本次上传的数据文件打开。

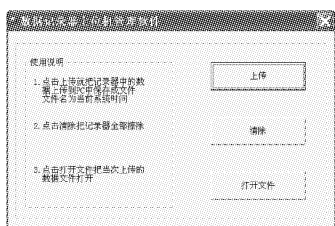


图 6 记录器上位机管理软件界面

## 4 结语

在设计并实现了数据记录器后,按照设计要求在实验室对它进行了测试,步骤如下:

1) 将一个文件通过 RS-422 串口下载到数据记录器中保存;

2) 通过 USB 接口将记录器中的数据通过数据记录器上位机管理软件,上传到 PC 并以文件的形式保存;

3) 设计一个文件比较软件,将源文件和上传的二进制数据文件进行比较,比较结束后如果完全正确会弹出对话框,告诉你两个文件完全一样,否则提示文件不同,并计算误码率。

按照以上步骤,对各种大小的文件,RS-422 使用多种波特率,对记录器进行了大量测试,测试结果表明上传文件全部正确,误码率为零,从而证明数据记录器设计方案正确和可行。现在该记录器已经在工业现场中实际使用,完全满足设计要求、使用方便。

### 参考文献:

- [1] 周立功. LPC2210 ARM 微控制器数据手册 1[Z]. 广州: 广州周立功单片机发展有限公司, 2004.
- [2] 王田苗. 嵌入式系统设计与实例开发[M]. 北京: 清华大学出版社, 2002.
- [3] 周立功. ARM 嵌入式系统基础教程[M]. 北京: 北京航空航天大学出版社, 2005.
- [4] 周立功. ARM 嵌入式系统实验教程(一)[M]. 北京: 北京航空航天大学出版社, 2004.
- [5] 周立功. PDIUSB12 USB 固件编程与驱动开发[M]. 北京: 北京航空航天大学出版社, 2003.

(上接第 253 页)

I/O 信息、网络信息、进程信息、服务信息、用户信息、应用活动状态信息和系统日志信息等。

表 2 WIN32 数据提供者映射的常见类

类名称	属性	对应的性能数据
win32_Processor	MaxClockSpeed	CPU 的主频
Win32_LoginMemory Configuration	TotalPagefileSpace	内存的虚拟值
Win32_PerfRawData_PerfOS_Memory	AvailableMbytes	内存的可用值
Win32_diskDrive	Size	硬盘的总量
Win32_LogicalDisk	FreeSpace	逻辑硬盘的空闲值
Win32_PerfRawData_Tcpip_NetworkInterface	CurrentBandwidth	网络的带宽
	BytesTotalPersec	网络的流量

某些性能数据并不能从相关的类中直接获取,需要进行一定的计算,如 CPU 使用率、内存空闲率、硬盘空闲率、硬盘碎片空间等等,下面以内存空闲率为例,简要介绍性能数据获取的实现原理。

```
Set collItems = objWMIService.ExecQuery_("Select * From Win32_PerfRawData_PerfOS_Memory") '查询内存使用值类
For Each objItem in collItems
    avmem = objItem.AvailableMbytes '获取可用内存(kB)
Next
set wbemObjectSet = objWMIService.ExecQuery_("Select * From Win32_LogicalMemoryConfiguration") '查询虚拟内存类
For Each wbemObject In wbemObjectSet
    tomem = wbemObject.TotalPhysicalMemory/1024 '获取物理内存(KB)
Next
data_mi = INT(avmem * 100 / tomem) '计算空闲内存率
```

### 3.3 预警处理机制

预警处理程序启动后,首先从“故障预警表”中读取所有

记录,并与采集程序中传递的性能信息进行比较,对超出阈值的性能数据,根据预警方式的不同调用不同的预警接口程序,发送相应的警报信息。对于原因确定的故障,可运行预先设计好的修复程序自动排除。其中,短信通知是最常用预警措施,本文采用东方软件公司开发的 SMIAS(短信网关系统)<sup>[4]</sup>提供的 SP 接口,实现短信预警,鉴于篇幅,仅给出关键代码。

```
cmpp_connect_to_ismg("134.130.51.68", 7890, &conn);
//向短信网关 134.130.51.68 发送连接请求
cmpp_login(&conn, &cl); //向短信网关登录注册用户
cmpp_submit(&conn, &cs); //发送短信到短信网关
cmpp_recv(&conn, &cp, 0);
//接收短信网关返回的处理信息
```

## 4 结语

本文利用 WMI 标准和消息中间件通信技术,实现了跨操作平台、跨设备的性能监控管理系统,提供故障发现、预警以及自动修复等多项功能,并在温州建行得到成功应用。实践证明,该模型使用价值高,通用性强,尤其适合于金融、电信、电力、民航和铁路等信息系统集中化程度高、高可靠性运行环境的应用场合。

### 参考文献:

- [1] MATTHEW L, ASHLEY M. Windows Management Instrumentation (WMI)[M]. Indianapolis: New Riders Publishing, 2001.
- [2] 徐晶, 许炜. 消息中间件综述[J]. 计算机工程, 2005, 31(16): 73-76.
- [3] 周宇, 赵成栋, 康建初. 应用性能管理中的相关标准和技术[J]. 计算机工程, 2004, 30(18): 101-102.
- [4] 刘荣辉, 刘光昌. 短消息代理网关系统的设计与实现[J]. 电信科学, 2005, (3): 23-26.