

PC 机通过 USB 口与工业控制系统的通信实现

周立平¹,李 劼²,张文根²,丁凤其²

(1. 中南大学 信息科学与工程学院,湖南 长沙 410083;

2. 中南大学 冶金科学与工程学院,湖南 长沙 410083)

摘要: 工业领域中 PC 机与工业控制系统的通信方式一般通过串行口实现,鉴于通用串行总线 USB (Universal Serial Bus) 的优越性,设计了基于 USB 总线的数据采集系统。该系统是由增强型 P89C61X2 单片机、CAN 控制器 SJA1000 和 USB2.0 接口芯片 ISP1581 组成,工业控制网络为基于 CAN 总线的现场总线控制系统。阐述了 PC 机通过 USB 口与工业控制系统(CAN 网络)通信的实现方法和设计细节。结果表明,通过 PC 机的 USB 口可以对 CAN 总线控制网络上的数据和信息进行实时采集和处理。

关键词: USB 总线; 现场总线控制系统; 在系统编程; CAN 总线; 驱动程序; API 函数

中图分类号: TP 39

文献标识码: B

文章编号: 1006-6047(2005)09-0069-04

0 引言

通用串行总线 USB(Universal Serial Bus)协议能适应大多数外设的高速度和高通用性的需要,因此得到了广泛的推广和开发。目前 USB 端口已成为微机主板的标准端口,这种连接较以往普通的并行口和串行口而言,主要的优点是速度高、功耗低、支持即插即用和使用维护方便。

目前 USB 设备的开发主要体现在便携式辅助存储器设计如 U 盘及数码相机等的扩展存储器;再就是数据交换量少的一些外设如键盘等。也有部分数据采集系统的设计,但是一般仅仅局限于数据的采集,而忽略了客户端软件(高级应用程序)的设计,且数据采集速度较低(12 Mbit/s),没有真正实现上下位机的数据互相通信。在上述产品的开发过程中一般都采用集成主控制器的芯片,以 Cypress 公司的 EZ-USB FX2 系列为例,它们都把 USB 接口芯片与主控制器集成到一块,因此在实际应用中还必须设计合适的数据采集模块,这就要求考虑多方面的因素,主要有采样时序控制与数据采集器的设计(这与具体的应用场合关联较大),因此不能灵活满足实际配置要求,而且相对价格较贵,最终的产品通用性也相对较差。

1 USB 系统总体设计思想

本 USB 系统由单片机主控电路、USB 协议转换模块、CAN 协议转换和数据采集模块三部分组成。P89C61X2 作为主控制器,管理主机与 USB 设备间的数据传输,同时也负责管理 USB 资源,如带宽、总线

能量、传输方式切换等。工作时主控制器 P89C61X2 采集来自 CAN 的数据和信息,利用 ISP1581 进行协议转换,将采集的数据通过 USB 总线送往 PC 机,在 PC 机上用户根据需要设计高级应用程序以实现数据进一步处理,同时,可实现对 CAN 数据的发送。在设计过程中可以把该系统作为控制系统与 PC 机的 CAN-USB 适配卡;也可当作一个通过 CAN 总线连接着整个控制系统的 USB 设备。采用这种方案的优点是系统组成灵活,可根据不同系统需求,搭配不同的 MCU,具有较高性能价格比,避免了采用集成 MCU 芯片的不灵活性。

2 系统硬件设计

2.1 主要器件介绍

图 1 为整个系统的原理图。主要有主控制器 P89C61X2,USB 接口芯片 ISP1581,CAN 总线接口,串口驱动器 RS-232 等。

P89C61X2 单片机支持 ISP 操作(在系统编程),控制功能强大,而且有 64k 的 FLASH,可以方便调试工作。设计时通过 PC 机的串口,将调试好的程序用编程器固化在 FLASH 中,因此这过程也称为固件编程。系统运行后所有控制行为都通过固件程序进行实施。

ISP1581 芯片是 PHILIPS 公司的高速 USB2.0 设备控制器,完全符合 USB2.0 规范,既支持高速(480 Mbit/s)操作,又支持全速(12 Mbit/s)操作。ISP1581 没有内嵌微处理器,这样就为各种范围的控制器提供灵活的接口,符合设计的需要。在本系统中 ISP1581 与微控制器的通信是通过一个高速的通用的并行接口 P1,P3 实现。这些高速的微控制器接口极大地提高了系统的信息吞吐量,也解决了 USB1.1 速度低的问题。在设计时选用了控制传输、中断传

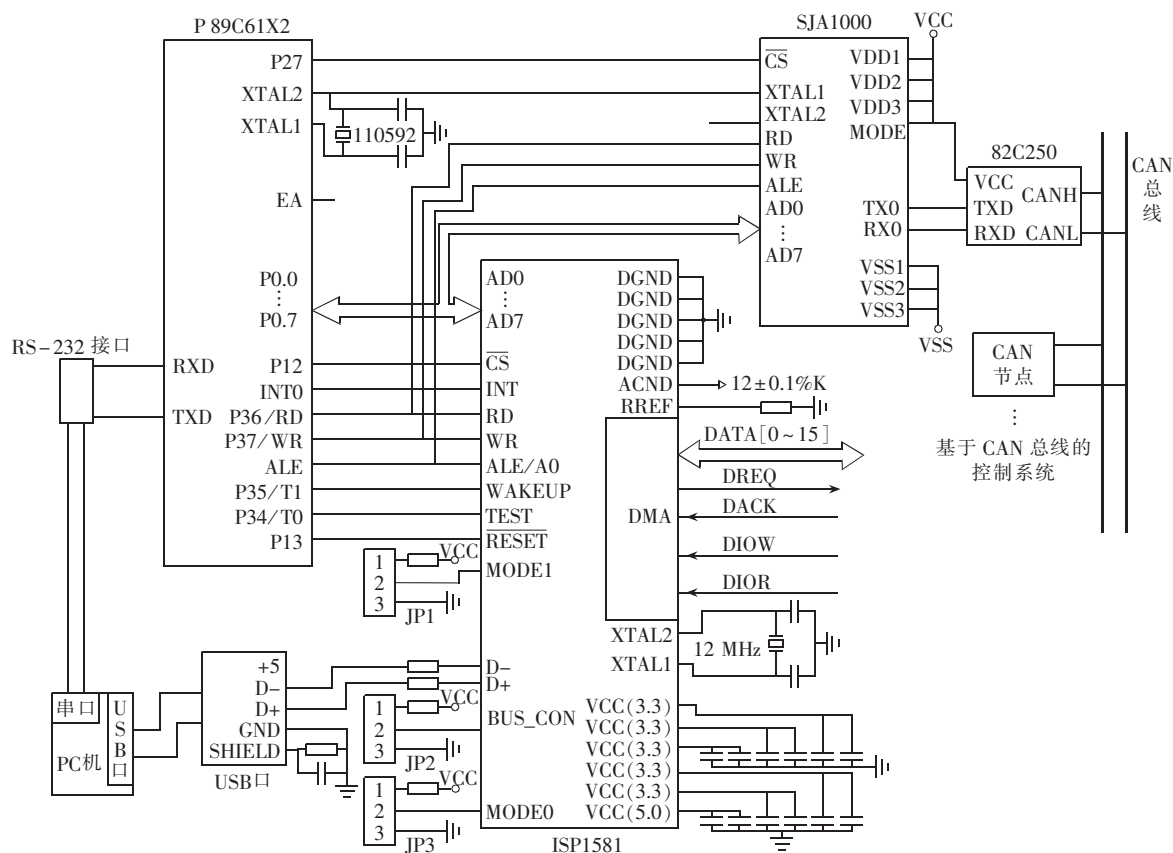


图 1 系统原理图

Fig.1 System schematic diagram

输和批量传输。控制传输用来传递控制信息,固定使用端点 0;中断传输使用端点 1,用来传送 CAN 网络状态信息;批量传输用来实现主机和 CAN 网络节点之间的数据传送,使用端点 2。在上电时,通过配置 BUS_CONF/DA0,MODE1,MODE0/DA1 引脚电平可以适应绝大多数的微处理器接口类型。例如,通过 BUS_CONF/DA0 引脚,总线配置可以选择普通处理器模式(generic processor mode)或者分割总线模式(split bus mode);在普通处理器模式下,通过 MODE0/DA1 引脚可以选择读写选通信号为 8051 风格或者 Motorola 风格。在此处选用普通处理器模式,采用 8051 风格的读写选通信号。

SJA1000 是 CAN 控制器,它实现了 CAN 2.0 的协议,具体可参阅文献[1]。

RS-232 驱动器实现 P89C61X2 的在系统编程。通过串口可边编程边调试,串口的功能也仅仅在此,当完成调试工作后,可以不接串口。

PC 机作为 USB HOST(主机)。

系统的工作电源通过 USB 总线提供,当外设功耗大时,也可以设计成外部供电(当 USB HOST 为掌上电脑时考虑采用外置电源)。

2.2 工作原理

系统的主要工作过程:主控制器协调 SJA1000 从基于 CAN 总线的控制系统采集数据信息,然后 P89C61X2 控制 ISP1581 对 SJA1000 的缓冲区进行

读取,ISP1581 的 16 个端点公用一个 8 k 的片内 FIFO 存储器,P89C61X2 还有双数据指针,能满足各种传输速度的需要(低速 1.5 Mbit/s,全速 12 Mbit/s,高速 480 Mbit/s),至于具体的数据协议转换和数据缓冲都在各部件中实现。当然在设备启动过程中 P89C61X2 单片机还要协调 PC 机进行 USB 设备枚举和判断 ISP1581 设备状态(传输方式、传输管道选择、速度匹配等,可参考 ISP1581 的说明^{①②}),传输状态切换等工作。当 ISP1581 在缓冲区满后能自动启动数据传输,此时 PC 机发起通信,P89C61X2 配合 PC 机通过 USB 口读取数据。PC 往下发送数据的通信是首先从 PC 机发起,别的过程正好相反。考虑到通常发送的数据量较少且要求高实时性,采用中断方式发送。在编程过程中,可以把 SJA1000 当作一个数据缓冲区,它只是定时刷新,再配合 ISP1581 读取或发送数据和信息。

2.3 主要控制信号

SJA1000 的片选为 P 27,时钟、读、写信号都由 P89C61X2 提供,数据/地址线与 P89C61X2 直接相连。

ISP1581 的片选为 P12,由于 P89C61X2 为 8 位,此处 ISP1581 工作于断开总线模式,此时 AD[0~7]作数据/地址复用线,DATA[0~15]仅作 DMA 总

① ISP1581Hi-Speed USB Interface Device Data Sheet (Rev.5).Philips Semiconductors, 2003.

② ISP1581Programming Guide(Rev.3).Philips Semiconductors, 2003.

线。注意当 ISP1581 在通用模式时 AD[0~7]仅作地址线,DATA[0~15]作系统总线和 DMA 总线。它的工作模式和主要信号可通过 JP 1,JP 2,JP 3 进行配置。

3 系统软件设计

系统软件部分由固件编程、计算机端设备驱动和应用程序三部分组成。

3.1 微控制器程序设计

微控制器程序设计的主要目的是在 USB 总线上的传输速度达到最大以及快速响应主机提出的各种请求。当然在 CAN 网络上即使近距离传输最大速度也仅为 1 Mbit/s,而 USB2.0 的最大传输速度可以达到 480 Mbit/s,因此可以很方便地在工程中扩展尽量多的模块(通过 USB HUB 可以最多挂接 127 个 USB 设备,每个 USB 设备挂接一个子控制系统)。

编程时采用模块化思想,每个模块实现一定功能,编程工具采用 Keil 公司的 C51 开发软件 Keil uVision2,程序分为以下几块:主函数 main.c 启动设备和系统的工作,负责跟踪 USB 事件,当事件发生时引导执行相应程序,启动后它主要是监视 ISP1581 的端点状态的变化;isr.c 是中断服务程序,在这里可视为后台处理,而 main.c 相当于前台处理,当前台检测到状态变化就跳到相应地方进行相关程序的处理;init.C 负责初始化单片机、SJA1000 及 ISP1581 芯片的所有端口、存储缓冲区、定时器等,把 SJA1000 的读写数据过程也当作初始化的一部分,与一般初始化的不同之处是它要通过 P89C61X2 的控制对缓冲区不断地刷新;vender.c 主要是厂商定义命令,处理厂商请求;chap_9.c 包含主要的 USB 协议,在主机和设备之间建立一个连接;cansingal.c 负责把控制网络的信号和数据采集上来(编程把采集速度设置为 200 k bit/s,需要的话还可以加大,ISP1581 在采集数据时采用批量传输,往低层发送数据时考虑实时需要采用同步方式)。

值得注意的是:微控制器在上电后完成所需要的初始化工作后,微控制器将重新连接 USB,包括将模式寄存器的 Soft Connect 设置为 1,这样确保了在微控制器准备好服务 ISP1581 之前,ISP1581 不会进行操作。中断服务包括 SJA1000 的中断服务程序和 ISP1581 的数据中断程序。主循环程序流程如图 2 所示,图 3 为 ISP1581 中断服务流程图。

3.2 驱动程序开发

驱动程序负责对底层硬件的访问,因此与硬件关联较大。图 4 所示为一个典型的 USB 系统的通信层次结构:它通过 USB 主机和 USB 设备互连组成,其中实箭头表示主机上的实际通信,USB 主机和 USB 设备之间的通信最终发生在物理线上。USB 设备上的相应接口根据不同的具体应用可能有较大的

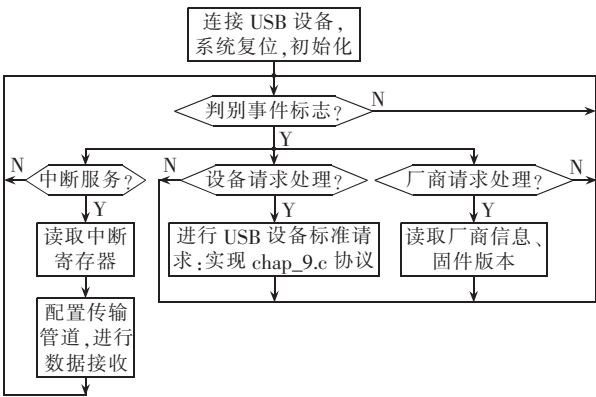


图 2 微控制器程序整体流程图

Fig.2 MCU program flowchart

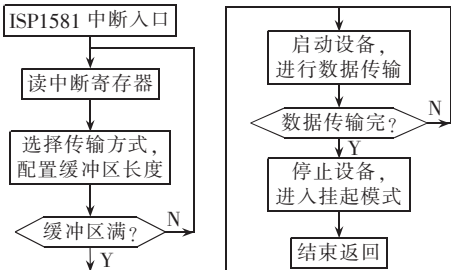


图 3 ISP1581 中断服务流程图

Fig.3 ISP1581 interruption service flowchart

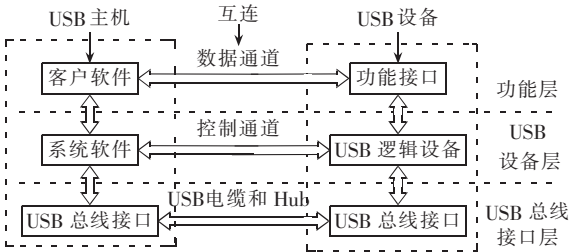


图 4 Windows USB 通信层次结构图

Fig.4 The communication hierarchical structure of Windows USB

差异,然而在每一水平层之间存在着逻辑接口,如主机中客户程序软件与 USB 设备的功能接口间的通信代表了设备需求与设备能力之间的约定。USB 总线接口处理电气层与协议层的互连[2]。

针对该通信模式,可以将整个 USB 系统分成三个基本组件:主控制器驱动程序(HCD),USB 总线驱动程序(USBD)和主机软件。在本驱动程序开发中,使用的开发工具是 Jungo 公司的 WinDriverV6.03。WinDriver 的优点是支持多种操作系统、用户可以不了解操作系统内部的具体工作机理,也不需要了解各个系统 DDK(Developing or Debugging in Kernel)的开发工具。只要在 WinDriver 提供的开发平台,即可完成驱动程序的设计工作,剩下的底层细节由 WinDriver 内核统一处理。降低了对开发者编程能力的要求,同时也大大缩短了开发周期。而且在编写驱动程序过程中,用户只须考虑控制设备本身,Windows 能将各种资源如内存分配,I/O 端口配置,DMA 申请全部自动完成[2,3]。

下面就使用 WinDriver 开发驱动程序步骤和过程作一个简要说明^[4~8]:

- a. 启动 WinDriver 的 Driver Wizard 工具,检测硬件是否正常;
- b. 在 Driver Wizard 中选择所使用的开发环境,这里使用 VC++6.0 开发环境,并生成驱动程序代码;
- c. 对生成的代码进行修改,使其符合目标系统的需要;
- d. 在 WinDriver 环境的用户模式下,调试驱动程序。

3.3 应用程序设计及思想

USB 的客户程序实际上是对客户端设备映像的操作,这些映像由 USB D 或 HCD 产生,在 Windows DDK 中存在着一组叫做 USB D Interface(USB D I)函数的 API 函数集。该函数集包括了传输函数管道 Pipe 函数、设备配置函数及其他函数。应用这些函数可编写支持任何 USB 兼容设备的 USB 驱动程序和客户程序。其中管道 Pipe 是 USB 设备驱动程序建立的逻辑通信通道。

该系统与 PC 机之间的通信主要通过 USB D I 的 API 函数实现对 USB 口的读写,这几个函数在设计驱动程序过程中实现。其中主要有:从 USB 设备读取数据 ExReadData(int pipenum,unsigned char* recbuff,int len,int waittime);向 USB 设备写入数据 ExWriteData(int pipenum,unsigned char* recbuff,int len,int waittime);在 PC 机上采用 Visual C++ 设计一个具体过程的监控界面^[5],通过界面的控件启动 API 函数就可以通过 USB 口把数据从 CAN 网络读上来和把数据送下去。

对于采上来的数据可以再通过开发一些应用程序对之进行处理,这也是该系统设计的主要目的。用户应用程序设计只需通过上面的 API 函数进行简单的引导,而不必再去考虑具体的设备实现过程。在系统设计中主要是用采集的数据生成报表和数据备份,同时在分析的基础上生成控制数据对工业系统进行控制,这因具体情况而定。

4 结论

该系统充分利用了 USB 总线的优越性,突破了控制系统与 PC 机之间通过以 RS-232 为基础的异步串行通信方式,很好地解决了通过 PC 机 USB 口与工业底层控制系统的接口和通信问题,方便了特别是长时间运行设备数据的采集。

参考文献:

- [1] 饶运涛,邹建军,郑勇芸,等. 现场总线 CAN 原理与应用技术[M]. 北京:北京航空航天大学出版社,2003.
- [2] CANT C. Windows WDM 设备驱动程序开发指南[M]. 孙义译. 北京:机械工业出版社,2000.
- [3] 周立功. DIUSB D I2 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.
- [4] 梁鸿翔,王润田. 基于 USB 2.0 的同步高速数据采集器的设计[J]. 电子应用技术,2004,(10):15.
LIANG Hong-xiang,WANG Run-tian. Designing a synchronous high-speed data collector based on USB 2.0[J]. **Electronic Application Technology**,2004,(10):15.
- [5] 汪晓平. Visual C++网络通信协议分析与应用实现[M]. 北京:人民邮电出版社,2003.
- [6] 王田苗. 嵌入式系统设计与实例开发[M]. 第 2 版. 北京:清华大学出版社,2003.
- [7] 张海峰,苏涛,张登福. USB 在数据采集中的应用[J]. 电力自动化设备,2004,24(3):63-65.
ZHANG Hai-feng,SU Tao,ZHANG Deng-fu. Application of USB in data collection[J]. **Electric Power Automation Equipment**,2004,24(3):63-65.
- [8] 廖忠. USB 在远程数据采集中的应用[J]. 计算机与现代化,2002,77(1):4-6.
LIAO Zhong. The application of USB in remote data acquisition system[J]. **Computer and Modernization**,2002,77(1):4-6.

(责任编辑:戴绪云)

作者简介:

周立平(1977-),男,湖南安仁人,硕士研究生,研究方向为现场总线技术与嵌入式系统(E-mail:lipian_zhou@126.com);
李劫(1963-),男,湖南汨罗人,教授,博士研究生导师,从事现场总线技术、冶金过程仿真与智能控制研究。

Communication through USB between PC and industrial control system

ZHOU Li-ping¹, LI Jie², ZHANG Wen-gen², DING Feng-qi²

(1. Information Science and Engineering School, Central South University, Changsha 410083, China;

2. Metallurgy Science and Engineering School, Central South University, Changsha 410083, China)

Abstract: The serial port is normally used in the communication between PC and industrial control system. Due to the high performances of USB(Universal Serial Bus), a data acquisition system based on USB is designed. It consists of enhanced P89C61X2(single-chip computer), SJA1000(CAN controller) and ISP1581(USB 2.0 interface chip). The industrial control system is based on FCS of CAN bus. The design and implementation of communication between PC and industrial control system through USB is expounded. The results show that it is practical to real-timely acquire and process the data and information from CAN through UBS communication.

Key words: USB; FCS; in system programming; CAN bus; driver program; API function