

LonWorks-USB 网关的设计与开发

郁汉琪¹, 胡秋林², 陆宝春²

(1. 南京工程学院工程训练与实验中心, 南京 210013; 2. 南京理工大学机械工程学院, 南京 210094)

摘要:介绍了用 USB 通信芯片 PDIUSB11 设计 USB 和 LonWorks 协议转换网关的方案。在分析了网关工作原理的基础上, 给出了硬件设计、USB 驱动程序设计、固件设计及 Neuron 芯片的通信程序设计, 并给出了一个用该网关实现列车门控系统的实例。

关键词: 网关; USB; LonWorks

Design and Development of LonWorks-USB Gateway

YU Hanqi¹, HU Qiulin², LU Baochun²

(1. Center of Training and Experiment, Nanjing Institute of Technology, Nanjing 210013;

2. School of Mechanism Engineering, Nanjing University of Science and Technology, Nanjing 210094)

【Abstract】 Based on communication chip PDIUSB11, this paper discusses the hardware and software design for LonWorks/USB protocol transition gateway. On the basis of analyzing the working theory of gateway, the design of hardware, USB driver, firmware and communication program of Neuron chip are presented. And an example of actualizing train gating system by gateway is given.

【Key words】 Gateway; USB; LonWorks

目前在微机上扩展的 LonWorks 总线设备接口一般采用 RS-232 串行总线。RS-232 虽然插拔方便, 但是传输速度太慢。而 USB 很好地解决了上述问题, 很容易实现便捷低成本的协议转换器, 并具有较高的传输速度, 即插即用, 是上下位机联系的一种新的方式。

在微控制器和 USB 接口的选择上有 2 种方式, 一种是采用具备 USB 通信功能的微处理器, 另一种是采用普通微处理器加上专用的 USB 通信芯片。第 1 种方法具有高集成度等特点, 但是由于这些芯片内部的微处理器具有 USB 接口, 因此它们与过去的开发系统是不兼容的, 需要购买新的开发系统, 投资较高。第 2 种方法的 USB 通用芯片只实现 USB 接口, 成本较低, 同时在外围微处理器的选择上可根据不同的系统需求, 选择自己熟悉的控制器进行开发, 使系统灵活, 简化了开发难度, 加快了系统的设计。因此, 我们选用了第 2 种设计方案, 设计了 LonWorks 到 USB 的转换网关。

1 系统工作原理

随着网络技术向控制领域的渗透, 企业管理与现场控制也迫切需要向网络化控制的方向发展。图 1 即为由 LonWorks 现场总线构建的车间网络化控制系统原理图。从图中可以看到, LonWorks-USB 网关是 LON 网与控制计算机的接口, 在通信过程中起着关键的作用, 它既完成了 LON 网与计算机的互联, 又实现了数据的交换。LonWorks-USB 网关作为 LON 网的一个特殊的网络节点——通信处理器, 能按照 LonTalk 协议与 LON 总线上所有其它的智能节点进行对等的通信, 起到网络桥梁的作用。当主控计算机要想控制底层的某个设备时, 它将命令从 USB 口发出, 网关将信息转化为 LonTalk 协议的报文发到 LON 网络上, 各个与从站计算机相连的 LonWorks-USB 网关把所有发送给它的报文信息接收下来, 重新打包装帧, 向下级计算机发送。下级计算机再通过网关将信息传到底层的 LON 网络。同样, 底层的设备也可以

通过这些过程, 将信息传给从站计算机和主控计算机。

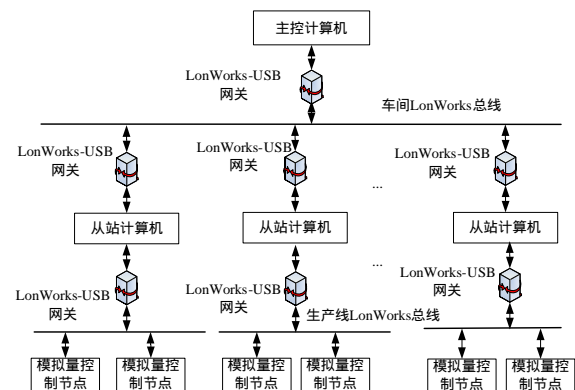


图 1 系统原理

2 硬件设计

LonWorks-USB 网关的硬件原理由图 2 所示, 它主要由 USB 通信芯片 PDIUSB11、8051 单片机、Neuron 芯片及它的一些外围芯片所组成。网关中, 神经元芯片选用的是可带外存储器的 MC143150。芯片内存储器的地址范围是 E800H ~ FFFFH, 包括 2KB 的 SRAM 和 512B 的 EEPROM。神经元芯片的晶振为 10MHz。由于有大量的数据缓冲区进行数据交换, 仅仅利用 MC143150 内部的存储空间是远远不够的, 因此扩展了外部存储器 FLASH 和 RAM。FLASH 选用 AT29C512, 其地址范围是 0000H ~ 7FFFH, RAM 选用 IS61C256AH-15N, 其地址范围是 8000H ~ DFFFH。Neuron 芯片与 LON 的网络介质的接口采用双绞线收发器 TPT/XF-1250, 传输速率为 1.25Mbps。Neuron 芯片有 11 个

作者简介: 郁汉琪(1961-), 男, 副教授, 主研方向: 自动化技术, PLC 技术, 数控技术等; 胡秋林, 硕士; 陆宝春, 副教授、博士
收稿日期: 2006-07-10 **E-mail:** hanqiyu2003@yahoo.com.cn

可编程的 I/O 引脚,并提供 4 类共 34 种 I/O 对象。通过引脚的不同配置,为外部硬件提供灵活的接口,实现不同的 I/O 对象。根据具体应用的要求,网关设计采用 Neuron 芯片预定义的并行 I/O 对象,允许以最高为 3.3Mbps 的速率进行双向数据传送,实现了高数据速率。并行口的工作方式有 3 种,即主模式、从 A 模式、从 B 模式。这里选择从 A 模式,在从 A 模式中,认为 Neuron 芯片为从 CPU,单片机 8051 为主 CPU,主 CPU 和从 CPU 之间的数据传输通过虚拟的写令牌传递协议得以实现。主 CPU 和从 CPU 交替地获得写令牌,只有拥有写令牌的一方可以写数据,或者不写任何数据传送一个空令牌。

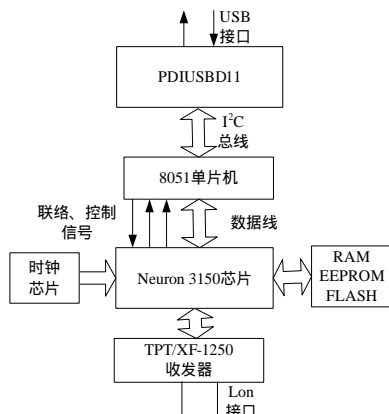


图 2 硬件原理

USB 外围芯片是 Philips 的 PDIUSB11,该芯片专门为外部的 MCU 提供了 I²C 接口,最高速度可达 1Mbps。其 SIE 接口包括 1 个中断和 1 个较大的寄存器,用于对 I²C 总线数据的存取。I²C 总线是一种串行总线,它以 2 根线实现了完善的全双工同步数据传送,可以方便地构成外围器件扩展系统。单片机 8051 与 PDIUSB11 芯片相连,因为 8051 单片机不带有 I²C 总线硬件接口,只能靠编写软件来实现 I²C 总线时序。串行 I²C 输入输出通过双向数据线 SDA 和时钟线 SCL 把设备连接到总线上,并且通过 SDA 和 SCL 在 8051 单片机与 PDIUSB11 芯片之间传送信息。

图 3 为各芯片之间的主要硬件连接。8051 单片机与 Neuron 芯片之间通过 8 位数据线并行口连接,8051 单片机与 PDIUSB11 之间通过 I²C 总线接口连接,I²C 总线的 SDA 和 SCL 都是双向 I/O 口线,通过上拉电阻相连,当总线空闲时,2 根线都是高电平。这里把 8051 单片机 P2.5、P2.6 管脚设为串行数据线(SDA)和串行时钟线(SCL),分别和 PDIUSB11 的 SDA 和 SCL 相连。

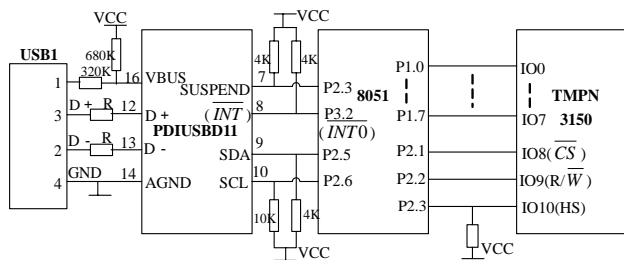


图 3 网关硬件连接

3 软件实现

软件的设计主要包括两部分:一是 PC 机端网关的驱动程序的编写和用户应用程序的编写;二是对 8051、

PDIUSB11 固件的编程和对辅 CPU3150 芯片的编程。

3.1 PC 机的编程

设备的驱动程序使用 WinDriver 组件和 Microsoft Visual C++ 来完成。利用它开发设备驱动程序不需要熟悉操作系统的内核,易于硬件设计人员直接进行驱动程序的开发,减少开发周期。在实际开发中采用的是用 Wizard 自动生成驱动程序的框架代码,根据实际需要修改代码,从而完成整个驱动程序的开发。设计步骤如下:

(1)安装 Visual C++ 和 DDK,再安装 WinDriver;

(2)装好后 Wizard 会首先检测附加在主机上的所有硬件设备,然后根据所选择的设备生成一个配置文件(.inf),包含设备的安装信息,同时生成一个库文件;

(3)在该类库的基础上修改和添加该网关的特定功能的函数,完成驱动程序的开发。

为了网关和主控计算机之间交换的数据能显示或利用,必须编写用户应用程序。开发的工具可使用任何一个支持 win32 函数 CreateFile()和 DeviceIOControl()的编译器,我们选择了 Visual C++ 开发工具。首先调用 CreateFile()函数,来取得访问设备驱动程序的句柄:

```
DeviceHandle=CreateFile(\\.\.\ezusb-0,GENERIC_WRITE,ILE_S
HARE_WRITE,NULL,OPEN_EXISTING,0,NULL);
```

然后调用 DeviceIOControl()函数来提交 I/O 控制码,并且为 CreateFile()函数返回的设备句柄设置 I/O 缓冲区。最后,调用 CloseHandle()函数关闭设备。其中,最重要的函数是 DeviceIOControl(),它的功能是完成应用程序与驱动程序之间数据的交换。

3.2 网关的软件设计

网关的软件设计包括两部分内容,对单片机 8051、PDIUSB11 固件的编程和对 Neuron 芯片的编程。

8051 通过 USB 通信芯片 PDIUSB11 与上位机进行通信。而 PDIUSB11 是专门的 I²C 接口芯片,所以 8051 必须通过软件实现 I²C 接口,完成与 PDIUSB11 的连接。当一个数据包来到时,PDIUSB11 将其中断线拉低,8051 单片机转入中断处理程序,首先读取 PDIUSB11 的中断状态寄存器,然后再读取其它寄存器的值,以获得数据包类型和包中数据。每次传送完一个数据包均会产生一次中断请求,中断处理程序监视控制序列并检测所有错误。

另一部分为 Neuron 芯片的软件程序的编写,用神经元的编程语言即 Neuron C 语言完成。Neuron 芯片主要完成将并口得到的报文解析,再利用 Neuron C 的消息传送机制,将解析的消息用网络报文传送到 LON 的网络上;同时,还将 LON 网络上其它节点以报文形式传送上来的数据或信息通过并口传送给单片机。

```
IO_0 parallel slave s_bus; //将管脚设为从 A 模式的并行输入/输出
#define BYTES 256 //设置单元数
when(io_in_ready(s_bus)) //输入数据就绪
when(io_out_ready(s_bus)) //输出数据就绪
io_in(s_bus,&p); //读取数据
io_out(s_bus,&p); //输出数据
msg_send(); //将数据发送到 LON 网络上
priority when(msg_arrives) //网络上的其它节点有消息传来
```

4 实例应用

笔者以某公司生产的列车自动门为研究对象,构建了基于 LonWorks 的列车控制网络,很好地验证了该网关的实用性。列车网络化监控系统主要由被监控设备、监控从站、监

控主站、LonWorks-USB 网关及门控器组成。这里采用了该公司的塞拉门为被监控对象；监控从站包括门控器及从站软件，负责监控本节车内的设备；监控主站包括主站计算机、主站软件。通过列车 LonWorks 网络监控主站可以将各种控制命令传达给监控从站，如要求某一个从站将塞拉门打开或者关闭；对应的监控从站接收到主站发来的命令后执行相关的动作或者将相应的数据信息传到主站，如从站执行主站发来的命令的同时将设备的状态信息发回给主站。图 4 为在列车上应用的网络化控制系统结构。

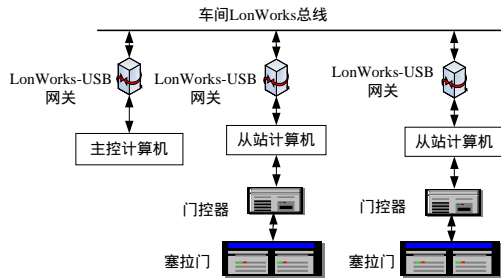


图 4 列车网络化控制系统结构

这样列车控制网络就将整个列车连成一个整体，列车总线通过通信节点与车厢控制器交换信息，完成对列车控制、检测和诊断等信息的传输。司机对整个列车的控制命令通过列车通信网络送到列车的各个车厢上，列车的各个车厢工作状态通过列车通信网络送到司机显示台上，从而使整个列车有效而安全地工作。

(上接第 242 页)

字段有：目标 MCU 类型、Flash 开始/结束地址、RAM 开始地址、整体擦除程序代码、页擦除程序代码和页写入程序代码。其中，3 个程序代码字段的值可通过对用户装载的 S19 文件处理后直接获得。用户在选择擦除、写入等操作前需将与当前目标 MCU 对应的参数发送给基板。

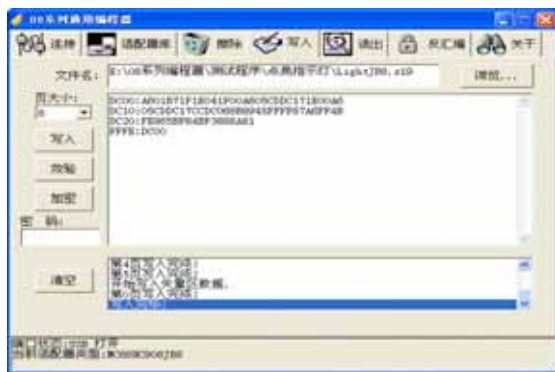


图 3 单击“写入”选项卡后显示界面

单击“擦除”选项卡，实现对目标芯片的整体擦除或页擦除，同时提供了查空功能。在“写入”标签下，读取用户选择的待写入目标芯片的 S19 文件内容，对其进行排序处理，然后将数据进行分页，分批发送给基板。写入完毕后，可将 Flash 存储器中的内容按页读出，并与 PC 机端的数据进行比较，若发现有字节不同，则表明写入错误，并立即停止比较。

单击“读出”选项卡后，用户输入 4 位十六进制开始地址和结束地址，单击“开始读”按钮后，程序发送命令字 102、

5 结论

多年来，PC 外设与 PC 的通信问题一直是开发人员关心的问题，USB 的高速率、易扩展、热插拔等特性比较好地解决了这个问题，为 PC 外设性能的大幅度提高提供了可能性。本文设计的 LonWorks-USB 网关为计算机与 LonWorks 之间的直接通信提供了一种有效的方法，为今后 LonWorks 网络的开发提供了一种新的方法；在结构上采用双 CPU 设计，用 I²C 总线作为芯片间的串行数据传输总线，具有结构简单、小型化、速度快的特点，非常适合用于测控网络系统。

参考文献

- 1 周琦, 袁学文, 肖文华等. LonWorks-USB 互连适配器的设计开发[J]. 电子技术应用, 2003, 29(12): 38-40.
- 2 马建国, 刘桂华. 并行 I/O 口模拟串行 I²C 总线技术[J]. 西南工学院学报, 1999, 14(1): 7-12.
- 3 杨育红. LON 网络控制技术及应用[M]. 西安: 西安电子科技大学出版社, 1999.
- 4 Echelon Corporation. Neuron C Programmer's Guide(Revision 4)[Z]. 1995.
- 5 石磊, 王学林, 陈慧等. USB-CAN 总线通信协议转换器[J]. 自动化技术与应用, 2004, 23(6): 34-36.
- 6 郁汉琪, 陆宝春, 徐德宏. 基于 PROFIBUS-DP 总线的冗余技术的实施策略[J]. 计算机工程, 2003, 29(14).
- 7 郁汉琪. 电气控制与可编程序控制器应用技术[M]. 南京: 东南大学出版社, 2004-06.

待读数据个数 N ($N = \text{结束地址} - \text{开始地址} + 1$)、开始地址高字节以及开始地址低字节，随后等待从基板接收 N 个数据，并按一定格式输出。

5 结束语

本文设计的通用编程器具有使用方便、扩展性强等特点，在进行软件设计时，尽量将芯片参数存放于高端 PC 机上。若要对新开芯片进行编程，仅需要根据该芯片进入监控的条件设计一个适配头，并将该芯片的参数添加到数据库中即可。目前，该通用编程器已经在小范围内试用，得到了用户的一致好评。

参考文献

- 1 王宜怀. 嵌入式应用在线编程开发系统的研制[J]. 计算机工程, 2002, 28(12).
- 2 Motorola.MC68HC908JB8 HCMOS Micro-controller Unit Technical Data[Z]. 1999.
- 3 Motorola.MC68HC908JB8 HCMOS Micro-controller Unit Technical Data[Z]. 1999.
- 4 Motorola.MC68HC908JL8 HCMOS Micro-controller Unit Technical Data[Z]. 1999.
- 5 Motorola.MC68HC908GP32 HCMOS Micro-controller Unit Technical Data[Z]. 1999.
- 6 Anderson D. USB 系统体系(第 2 版)[M]. 北京: 中国电力出版社, 2003.
- 7 Compaq, Intel, Microsoft, NEC. Universal Serial Bus Specification Revision 1.1[Z]. 1998.