文章编号:1001-2486(2001)03-0115-05

基于 ISA 总线的 CAN 通讯卡的设计与实现*

陈杨,龙志强,吕治国

(国防科技大学机电工程与自动化学院,湖南 长沙 410073)

摘 要:为实现 PC 机在控制局域网络(CAN)中的应用,开发设计了一种 PC 机与 CAN 的接口——CAN 通讯卡。该通讯卡采用微控制器完成底层的 CAN 通讯任务,并利用双口 RAM 与 PC 机进行数据交换;通过基于 Win9x 的 PC 机设备驱动程序实现上层应用程序对通讯卡的操作。该系统结构紧凑、可靠性高,已在磁悬浮列车控制系统中得到了成功应用。

关键词:控制局域网络;ISA;虚拟设备驱动程序

中图分类号:TP334 文献标识码:A

The Design and Implementation of CAN Board Based on ISA Bus

CHEN Yang, LONG Zhi-qiang, LU Zhi-guo

(College of Mechatronics Engineering and Automation, National Univ. of Defense Technology, Changsha 410073, China)

Abstract: A CAN board acted as the interface between PC and controller area network is designed and its virtual device driver under Windows9X OS is developed, which ensure that PC can be successfully used in CAN. Micro controller on board accomplishes underlying communicating tasks, and exchanges data with PC through dual-port RAM. This system, which characterizes simple structure and good reliability, has been applied to maglev successfully.

Key words: controller area network(CAN); ISA; Virtual Device Driver(VxD)

现场总线是应用在微机化测量控制设备之间实现双向串行多节点数字通信的系统,也被称为开放式、数字化、多点通信的底层控制网络。在武器装备自动化、制造业、流程工业、交通、楼宇、测控等方面的自动化系统中具有广泛的应用前景。

CAN 是目前比较流行的现场总线中的一种。它最初是由德国 Bosch 公司在 20 世纪 80 年代初期,为了解决现代汽车中的控制与测试仪器之间的数据交换而开发的。CAN 总线具有以下优点:支持多个主节点,各节点通过总线仲裁获得总线使用权;通信速率高,最快可达到 1Mbps;可靠性高,总线协议具有完善的错误处理机制。因此,CAN 总线在工业生产的各个领域得到了广泛应用。1993 年,ISO 组织为 CAN 制定了 ISO 标准。CAN 总线目前主要有两种技术规范:2.0 Part A 和 2.0 Part B。

PC 机具有强大的数据处理能力和友好的人机交互界面,将 PC 机作为上位机引入 CAN 网络,可以大大提高网络的整体性能。一方面可为整个控制过程建立数据库,通过 CAN 网络采集数据,根据过程变化采用相应的控制策略,再经 CAN 网络向控制对象发出控制指令。另一方面也可通过 PC 机了解系统的运行状况,对异常情况及时干预。然而,PC 机上是没有 CAN 接口的,所以必须制作一块 PC 通讯卡。

1 通讯卡的硬件设计与调试

本文所设计的 CAN 通讯卡由三个系统构成:CAN 总线接口部分,控制部分,PC 机接口部分。系统框图如图 1 所示。控制部分是通讯卡的核心,CAN 总线过来的数据经控制器预处理后,暂时存储在卡上的片外 RAM 中,PC 机定时访问数据。一旦 PC 机发出指令,通讯卡上的控制器将被硬件中断触发,马上对指令做出反应。

1.1 CAN 总线接口部分

CAN 总线协议的实现必须依赖 CAN 控制器。这里选用了 Philips 公司的产品 SJA1000。该芯片支持

* 收稿日期:2000-10-10
作者简介:陈杨(1978-),男,博士生。

2.0B 协议。

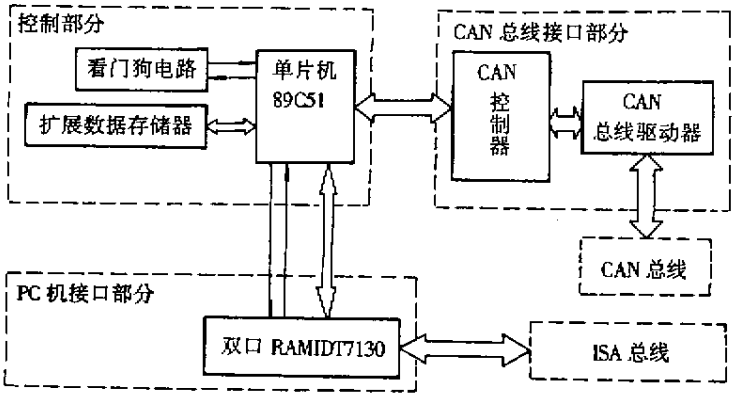

图 1 通讯卡硬件结构框图

Fig.1 Structure of CAN board

SJA1000 有两种工作模式:基本模式和增强模式。两种模式的硬件电路都是相同的,差别主要存在于对 SJA1000 的软件编程方面。不同模式下,SJA1000 具有不同的寄存器组结构。在基本模式下,SJA1000 只可收发标准数据帧(标准数据帧的标识符为 11 位),且错误报警的极限值不能修改;在增强模式下,SJA1000 既可接收标准数据帧,也可接收扩展数据帧(扩展数据帧的标识符为 29 位),可修改错误报警的极限值,并且 SJA1000 具有更加灵活的滤波方式,能够根据数据帧的标识符有选择地接收一些数据帧。另外,增强模式下的 SJA1000 能够进行自检,即可通过自发自收一组报文来判断该控制节点是否正常地挂在 CAN 总线上。使用者所要做的主要工作只是 SJA1000 的初始化,收发报文的处理以及对节点脱离总线的检测与处理。

CAN 控制器必须通过总线驱动器 82C250 才能接在 CAN 总线上,数据流经总线驱动器转变为差分信号。信号的差分传输有效地提高了总线的抗干扰性。具体电路如图 2 所示。图中,6N137 为高速光电耦合器(延迟时间 $t < 75ns$)。通讯卡的电源取自 ISA 总线。为了保护计算机,电源应通过 DC—DC 变换后再供给光耦隔离后的 CAN 总线接口部分使用。若该 CAN 控制节点挂在 CAN 总线末端,则总线驱动器 82C250 的 CANH 与 CANL 两端应加一个阻值为 120 欧姆的终端电阻进行阻抗匹配。

1.2 通讯卡控制部分

通讯卡控制部分核心是单片机 89C51,外加一些附属电路,如看门狗 MAX813、扩展的数据存储器、译码电路 GAL16V8 等。

看门狗 MAX813 使用方便,无需编程。它具有四项功能:上电复位、掉电检测、手动复位、看门狗计时。MAX813 复位信号的输出受两个端口信号影响:PC 机发出的复位命令信号;看门狗计时器溢出引起的复位请求信号。使用硬件看门狗有效地提高了系统的可靠性,避免了网络的瘫痪。采用 GAL16V8 译码产生片选信号,不仅修改方便,而且保密性高。

1.3 PC 总线接口部分

通讯卡采用了 ISA 总线,设计具体电路前需要对 I/O 端口地址,存储器地址空间,中断号,DMA 号等四大总线资源进行分配。

通讯卡采用 1K 的双口 RAM IDT7130 为单片机与 PC 机交换数据服务。双口 RAM 具有两套独立的地址总线,数据总线,片选信号线、读写控制线以及中断线,一方通过对某一特定存储单元进行写操作,向对方发出中断信号。双口 RAM 内部的地址比较电路,避免两者对同一存储单元进行访问。PC 机通过定时查询的方式访问双口 RAM,不必给通讯卡分配中断号,另外传输的数据量不是很大,所以,也无需 DMA 号。接下来的问题便是 PC 机是以端口映射的方式,还是以存储器空间映射的方式访问双口 RAM。

实际上,两种方式都行。若采用端口映射的方式,由于 x86 系统借用 10 条低位地址线($A_0 \sim A_9$)译

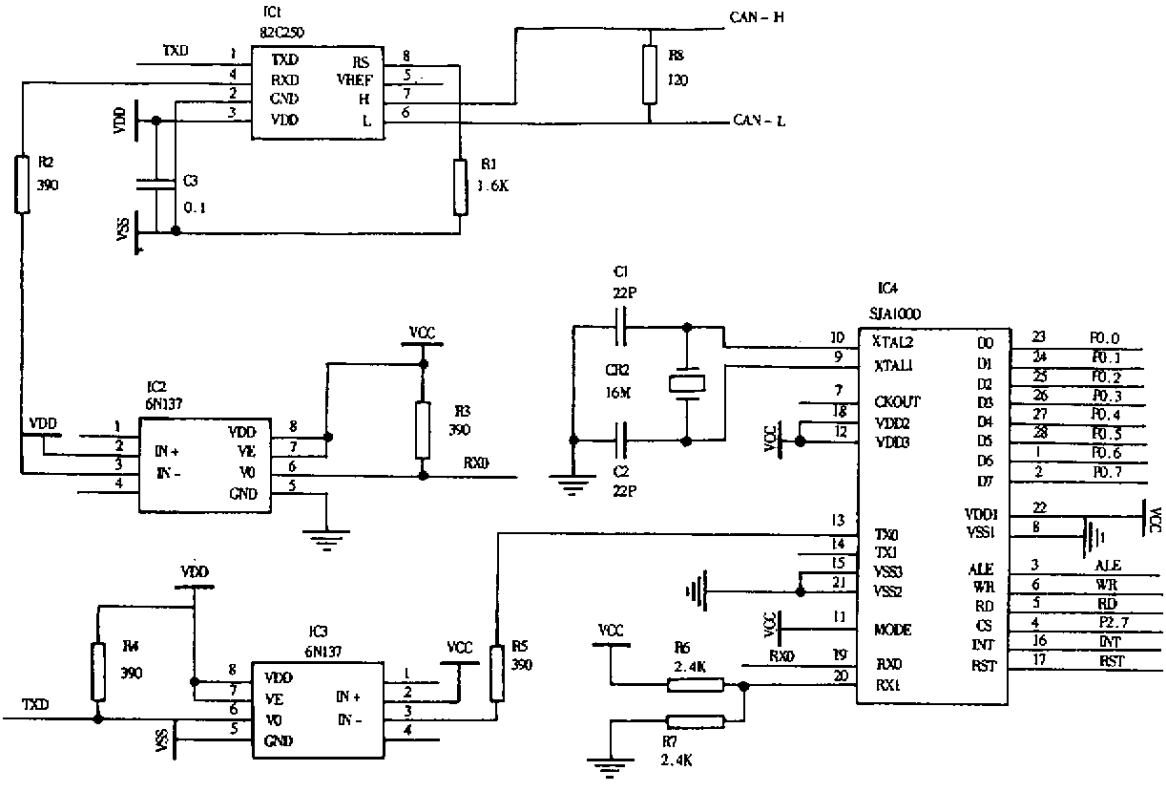

图 2　CAN 总线接口电路

Fig.2　Interface circuit of CAN bus

码产生端口的片选信号，并没有提供 1K 的端口地址资源可供利用，必须借用高位地址线帮助解决该问题。保持 A9 至 A4 这六条地址线不变，用来译码产生 IDT7130 的片选信号，变换 A15 至 A10，A3 至 A0 这十条地址线来访问不同端口。这种译码方法所得到的 1K 端口地址空间是不连续的。需注意的是，端口译码时，需添加表示 DMA 操作正在进行的 AEN 之反向信号和 IOR、IOW 信号来限定。

若采用存储器空间映射的方式，则比较简单。x86 系统已为主板上的插卡留下了 256K 的地址空间（0A0000H ~ 0DFFFFH）。通过通讯卡上的跳线，改变通讯卡的基地址，为该通讯卡找到一段空闲的地址空间。设计采用了第二种方式。

1.4　硬件调试

硬件调试分两步进行。第一步，把通讯卡作为一个单片机控制的 CAN 节点挂在 CAN 总线上，与其它 CAN 节点通信，以此确定通讯卡上的 CAN 接口部分正确无误。第二步，测试通讯卡的 ISA 总线接口部分，调试工具为 Debug。在将通讯卡插入 ISA 插槽之前，需用 Debug 查找到一段空闲的地址空间，然后完成通讯卡上基地址跳线。具体查找方法是用 Debug 的 D 命令显示 0A0000H ~ 0DFFFFH 范围内某段空间的内容，如果全是 0FFH，并且用 Debug 的 E 命令无法修改其内容，则这段空间即可作为 IDT7130 的映射地址。

调试时，用 Debug 的 E 命令修改 IDT7130 某一存储单元内容。通讯卡上的控制器配合，将该单元的值读出再通过 CAN 网络发往其它 CAN 节点，最后把修改的值显示出来，以此来判断对通讯卡的写操作无误。

同理，也可从其它地方输入值，传到通讯卡上，接着控制器将其写入 IDT7130，再用 Debug 的 D 命令显示 IDT7130 的内容，以此来检验对通讯卡的读操作。

2　通讯卡的软件设计

该通讯卡的设备驱动程序是为操作系统 Windows98 开发设计的，用于驱动连接到计算机的硬件设

备。在 Windows98 下,各硬件设备都有相应的虚拟设备驱动程序 VxD。Windows98 操作系统包括以下几部分:虚拟机管理器 VMM(Virtual Machine Manager)和虚拟设备驱动程序 VxD (Virtual Device Driver),16位和32位的应用程序,16位和32位的 DLL。系统核心是由 VMM 与 VxD 的集合组成。

虚拟机 VM(Virtual Machine)是一个可运行的任务,包含应用程序支撑软件,内存和 CPU 寄存器,Windows98 下有 DOS 虚拟机和系统虚拟机两种。虚拟机管理器主要任务便是创建、运行、监控和终止虚拟机,它工作在 Ring 0 层,不可重入。

VxD 也工作在 Ring 0 层。它作为操作系统信任的一部分而存在,所以必须精心设计。VMM 捕获权限不为0级的应用程序对底层硬件操作的请求,然后调用 VxD 的服务。对底层硬件的操作是由 VxD 完成的,VxD 也可调用其它 VxD 的服务。在使用 VxD 的服务前,必须先将 VxD 加载。Windows 提供了三种途径加载 VxD:途径一,在 system.ini 中登记条目,方法是 device = × × × .vxd;途径二,在注册表中声明,这是大部分 VxD 使用的方法,声明的位置在 HKEY_LOCAL_MACHINE \ System \ Current Control Set \ Services \ VxD;途径三:应用程序动态加载,使用 API 函数 CreatFile,此时 VxD 必须放在系统根目录下 C:\ Windows \ System。

由于通讯卡采用了内存映射的方式,所以存在着地址变换问题,Windows 利用'分页'技术来管理存储器,所谓'分页'技术,即是把线性地址空间的一个区域通过页表映射到物理空间的某个区域(RAM 或硬盘上)。现在已知一块物理空间区域,如何获得它对应的线性空间区域的指针是问题的关键。这个工作将由 VxD 来完成,一旦获得这个指针,那么应用程序中对 IDT7130 的读写将转换为对指针的操作。

目前,开发 VxD 常见工具有 Microsoft 公司的 DDK,Vireo software 公司的 VtoolsD。VtoolsD 引入了类的概念,并针对各类硬件设备驱动程序提供相应的 C++ 类库,因而开发效率较高。以下是应用程序对 VxD 调用的部分代码:

```
void main()
 {
     HANDLE hDevice;
     INT Val;
     PVOID inBuf[1];      // buffer for passing handle to VxD
     DWORD RetInfo [2];      // buffer to receive data from VxD
     DWORD cbBytesReturned;      // count of bytes returned from VxD
...
     hDevice = CreateFile("\\\\.\\mapdev.vxd", 0,0,0,
                              CREAT_NEW, FILE_FLAG_DELETE_ON_CLOSE, 0);
                    //获取 Vxd 的句柄
     if (hDevice == INVALID_HANDLE_VALUE)
      {
          fprintf(stderr, "Cannot load VxD, error = %08lx \n",
                         GetLastError());
          exit(1);
     }
     cprintf("%d", *pVAL);
     inBuf[0] = (PVOID)&Val;
     if ( !DeviceIoControl(hDevice, MDR_SERVICE_MAP,
                         inBuf,(DWORD)sizeof(inBuf), NULL,0,&cbBytesReturned,NULL)
       )
     fprintf(stderr, "Cannot control, error = %08lx \n",
                         GetLastError());
```

```
//应用程序通过 DeviceIoControl(…)向 VxD 发命令,命令代码为 MDR_SERVICE_MAP
…
}
```

为了方便使用者使用,可将上述对 VxD 服务的调用封装成 DLL,使 VxD 对使用者透明。DLL 提供了对通讯卡的初始化函数,状态查询函数及对数据缓冲区的读写函数,使用起来很方便。根据实际要求,自行开发计算机外设并定制其设备驱动程序,对 WINDOWS 操作系统进行模块化扩展。这得益于 WINDOWS 操作系统本身的模块化设计。Windriver,Vtoolsd 等一批优秀开发包的出现,已在工程界掀起了对虚拟设备驱动程序研究的热潮。

3 结束语

利用所设计的通讯卡开发出的分布式测控系统已成功应用于磁悬浮列车系统中。半年多的实验表明,该系统具有高速、可靠、实时性好,易于维护且工程造价低等优点。它很好地解决了集中式测控系统应用于大型设备控制时所产生的诸多问题。

参考文献:

[1] 杨强,李秋堂. Win9X 虚拟设备驱动程序编程指南[M]. 北京:清华大学出版社,1999.
[2] 刘其峰. WIN95 下虚拟设备驱动程序设计开发[J]. 电子技术应用,2000,15(4).
[3] 陈杨,龙志强,刘曙生. 基于 CAN 总线的数据通信系统研究[J]. 测控技术,2000(10).