

通过虚拟驱动程序 (VxD) 实现稳定的串口通信

乔锐 虞钢

(中国科学院力学研究所 北京 100080)

摘要: 本文主要介绍在 Windows95/98 环境下, 利用 VxD 开发工业激光器通信程序, 以实现稳定的通信。同时还介绍了 Win32 应用程序与底层 VxD 交换数据的方法。文中给出了开发的程序以及所实现的功能。

关键词: VxD 串口通信 实时时钟 工业激光系统

1 问题的背景

激光加工技术目前已越来越广泛地应用于现代工业生产之中。在我们开发的集成化柔性激光加工系统中, 采用了框架式五轴机器人结构和全数控的大功率工业脉冲激光器系统^[1]。其中激光器系统和机器人系统均采用了上下位机控制。上位机可以是一台普通 PC 机, 与位于控制柜中的下位机通过串口通信。由于我们所采用的工业激光器工作于近红外波段, 属不可见光范围, 在计算机控制过程中如果通信中断可能会造成严重的后果。因此为确保正常工作和操作安全, 通信的稳定性至关重要。控制激光器的串口通信采用了软握手方式, 即激光器与控制计算机在没有数据交换时, 以 10ms 左右的间隔互相发送特定的字符作为握手信号, 以确认通信的正常, 激光器如果连续发送 3 个握手信号未接到计算机的应答就自动进行停机保护, 因此对激光器的应答反应时间必须小于 30ms。由于采用了基于 Windows 95/98 的控制程序, 考虑到 Windows 多任务引起的进程非实时性以及加工机器人在加工中以 20Hz 左右的频率通过另一个串口不断传回当前机器人的位置与姿态数据, 因此要想在同一个控制程序中通过两个串口对激光器系统和机器人系统进行控制, 必须首先实现通信的稳定和互不干扰。

2 技术方案

由于全数字工业激光器系统对通信的要求比较苛刻, 因此我们采用了虚拟设备驱动 VxD (Virtual Device Driver) 作为系统底层控制。VxD 是 Windows95/98 操作系统下各种虚拟设备驱动程序的统称, 它是管理系统软硬件资源的可执行文件, 运行于处理器 ring 0 上的动态连接库(DLL), 具有最高的优先级。我们把与激光器通信有关的功能放于 VxD 中, 利用实时时钟(RTC)硬件中断控制相应串口发送数据。Windows 在处理硬件中断时, 首先由虚拟可编程中断控制器 VPICD(Virtual Programmable Interrupt Controller Device) 处理, 如果有某个

本文于 2000-05-17 收到, 2000-06-21 收到修改稿。

VxD事先注册了这个硬件中断的处理程序，那么VPICD就把硬件中断交给这个VxD处理；如果没有VxD注册这个硬件中断的处理程序，那么VPICD就把硬件中断交给有相应TSR的虚拟机(VM)处理^[2]。在实时时钟硬件中断处理函数中向激光器发送握手信号或控制命令，中断频率可以通过在system.ini加入如下两行：[rtcint] FreqIndex=n 加以调整，通过改变n值，就可以设置新的中断频率，n的选择如表1所示。

在本程序中选择的频率为1024Hz，即以1ms为单位进行扫描，确定是否需要发送握手信号或控制字符。选择这个频率主要是考虑到由于采用的是事件驱动的方式读取数据，为了可及时应答，应选择较高的频率。1ms的间隔可满足程序的要求，又不会过多占用CPU时间。

对串口的监测以事件驱动I/O方式进行，即在VxD中预先指定所响应的串口事件，例如接收到一个字符并放入输入缓冲区中，这样当有对应的串口事件产生时，Windows就会自动通知VxD。通过使用事件，应用程序不需要为接收数据而不断地检测串口，从而节约了宝贵的CPU时间^[3]。VxD对接收到的数据，只检查数据的完整性以确保通信正常，而不对数据进行具体分析，这部分工作放到Win32应用程序中完成，这反映了在VxD设计中的一个基本思想——把最关键部分放到VxD中完成，一般和外围的工作则由常规程序来做，以保证VxD以尽可能快的速度和相对较小的CPU占用时间运行。

运行于处理器ring3上的Win32应用程序完成对激光器数据的处理和分析，以及与机器人的通信。VxD与Win32应用程序的通信采用了异步过程调用方式APC(Asynchronous Procedure Calls)。通过W32_DEVICEIOCONTROL接口由Win32应用程序向VxD注册一个回调函数，然后VxD在适当的时候主动调用这个函数，从而达到相互通信的目的^[4]。由于激光器发送过来的数据长度不确定，因此VxD采用了动态分配内存的方式，当收到一个数据时，申请一块全局内存，把激光器数据存入其中，并把指向这块内存的指针通过APC调用传到Win32应用程序中。应用程序使用完之后，通知VxD删除这块内存。由于VxD运行于系统的底层，因此它申请的全局内存可以直接被Win32应用程序访问。这里应该注意的是，在使用异步过程调用时，Win32应用程序只有进入可惊醒的睡眠状态(Alertable Sleep State)时才可以响应调用，因此可以在程序中加入对WM_TIMER的响应，定时进入睡眠状态以响应可能的APC调用。

3 编程实现

VxD的开发采用Vireo公司设计的VxD开发工具箱VtoolsD，利用VtoolsD中的开发工具QuickVxD生成一个可动态加载的、基于C语言的VxD源程序框架。动态加载的VxD可以在应用程序运行时载入，程序退出时自动卸载，具有较大的灵活性，下面给出其流程。

(1) 在Win32应用程序中加载VxD并设定APC回调地址。

```
//加载与激光器通信的CommCtrl.VXD，并获得其句柄
hDevice=CreateFile("\\\\.\\CommCtrl.VXD",0,0,0,CREATE_NEW,FILE_FLAG_DELETE_ON_CLOSE,0);
```

表1 实时时钟中断频率设置范围

n	频率/Hz	n	频率/Hz
1	256	7	512
2	128	10	64
3	8 192	11	32
4	4 096	12	16
5	2 048	13	8
6	1 024	14	4

```
//通过DeviceIoControl函数把APC回调函数的地址传给VxD
DeviceIoControl(hDevice, APCX_REGISTER, &CommCtrlApc, sizeof(PVOID), &RetInfo,
    sizeof(RetInfo), &nBytesReturned, NULL)
```

(2) VxD 初始化工作是在函数 OnSysDynamicDeviceInit() 中实现的，该函数为消息 SYS_DYNAMIC_DEVICE_INIT 的处理函数，在 VxD 动态加载后由操作系统自动调用。

```
BOOL OnSysDynamicDeviceInit()
{
    .....
    struct VPICD_IRQ_Descriptor IRQdesc; //传给VPICD_Virtualize_IRQ的结构
    IRQdesc.VID_IRQ_Number=8; //响应实时中断IRQ 8
    IRQdesc.VID_Hw_Int_Proc=(DWORD)VPICD_Thunk_HWInt(RTCInt_Handler,
        &RTCInt_Thunk); //设定中断处理函数地址
    RTCIRQHandle=VPICD_Virtualize_IRQ(&IRQdesc); //注册中断
```

(3) 在 VxD 中初始化串口并保存 APC 调用地址。

```
HANDLE hPort=COMM_OpenComm("COM2", (VMHANDLE)-1); //打开串口2
DWORD OnW32Deviceiocontrol(PIOCTLPARAMS p) {
    //Win32应用程序调用DeviceIoControl函数后相应的VxD接收控制消息W32_DEVICEIOCONTROL,
    //调用消息响应函数OnW32Deviceiocontrol,PIOCTLPARAMS结构中保存了APC调用的参数
    switch(p->dioc_IOCTLCode) {
        case DIOC_OPEN: //Win32应用程序调用CreateFile时调用
            break;
        case DIOC_CLOSEHANDLE: //Win32应用程序退出时调用
            break;
        case COMMCTRL_ApcAddress: //Win32应用程序调用DeviceIoControl时保存回调函数APC的地址
            CommCtrlApc=(PVOID)p->dioc_InBuf;
    }
}
```

(4) 处理中断，与激光器通信，与 Win32 应用程序交换数据。

```
//RTCInt_Handler响应IRQ8,处理中断
BOOL_stdcall RTCInt_Handler(VMHANDLE hVM, IRQHANDLE hIRQ)
    { //在这里发送握手信号及控制命令 }

HANDLE TheThread=Get_Cur_Thread_Handle(); //获得进入睡眠状态的Win32应用程序的线程句柄
VWIN32_QueueUserApc(CommCtrlApc, 0, TheThread);
    //VxD通过VWIN32_QueueUserApc服务来调用Win32应用程序注册回调函数
    //CommCtrlApc:Win32 app的回调函数APC的地址
```

(5) 退出。

```
VOID OnSystemExit(VMHANDLE hVM) {
    .....
    VPICD_Physically_Mask(RTCIRQHandle);
    //注销在虚拟可编程中断控制器(VPICD)中
    //注册的中断
    VPICD_Force_Default_Behavior(RTCIRQHandle);
    VCOMM_CloseComm(hPort); //关闭串口}
```

4 实验结果及结束语

结合中科院重大仪器设备研制项目^[1]，本程序经过长时间实际运行，工作稳定、可靠，未出现过因通信中断而停机的现象。在保持激光器稳定控制的同时，可同步接收五轴机器人以

20Hz 左右的频率发回的位姿数据。VxD 对系统占用小，利用 Windows 自带的程序“系统资源监视器”观察系统资源占用小于 5%，本程序运行中不受运行其他程序的影响，可在操作中同时启动多个其他的程序。

将实时性要求很高的通信功能放到 VxD 中，可以使其他部分程序的编制相对独立，从而避免了它们对通信产生的干扰。VxD 是一种十分有效的程序，它可以解决常规应用程序所无法解决的问题，代码短小，效率高。但应注意的是，由于 VxD 位于操作系统的底层，因此编写和调试过程均难度较大，同时由于它拥有与操作系统一样的权限，在运行中对操作系统可能产生的破坏也比常规的应用程序大得多。因此除非十分必要，应尽量避免编写这种底层程序。

参 考 文 献

- 1 虞钢. 一种具有柔性传输和多轴联动的装置. 中国发明专利, 98101217.5. 1999-06-4.
- 2 Mitt Pictrek 著, 米东, 王森译. Windows95 系统编程奥秘. 北京: 电子工业出版社, 1996.
- 3 Charles A. Mirho 等著, 贺军等译. Windows95 通信编程. 北京: 清华大学出版社, 1997.
- 4 孙喜明. VxD 技术内幕. 中国计算机报, 1999 年 4 月.

简 报 利用 POWERPOINT 制作超媒体教学幻灯片

超文本是由节点和表达节点之间关系的链组成的网络。这种平面结构和直线结构不同。从学习的路径、使用角度、信息的表现形式来看，在超文本结构中，用户可以根据学习水平和特点很方便地提取相关知识进行学习，可在整个网络中进行浏览、查询、注释、切换等操作。在超文本的节点中加入多媒体信息，如声音、图形、图像、动画等，就构成了超媒体。

制作超媒体教学幻灯片时，把每张幻灯片看作是超媒体结构中的一个节点。首先，根据教学内容和教学目标制作出每张幻灯片，幻灯片的次序可任意，然后设定关键字，关键字可以是文字、图形、图像等，并确定这些关键字分别和哪些幻灯片进行链接；最后按上述设定的关键字选中后通过“动作设置”或“超级链接功能”链接到相应的幻灯片上。此外，被链接的幻灯片还应有返回链接处的链接。

制作超媒体教学幻灯片应考虑四个方面：选题（该题目是否适合于使用超文本结构）、节点设计（节点内容不应过大或过小，过大则内容杂乱，过小则链接困难）、节点内容的呈现方式（可根据界面一致性和连续性的原则，利用母版将幻灯片划分为多个小区，或作为教学内容，或作为动画显示，或作为标题等）和链接方式（可在教学内容中选取相应关键字，在学生容易出现分歧或迷惑的地方设立导航键或帮助键建立链接）。

林慧君（广东省湛江教育学院 524037）

摘要 00049：浅谈 ActiveX 控件及创建

本文简要介绍了 ActiveX 技术、ActiveX 控件，并以一个监测数据库中某一字段的 ActiveX 控件为实例，讲述 ActiveX 控件在 Delphi5 中的创建和使用。

作者：余洁，蔡翔云，姜麟（昆明理工大学 昆明 650093）