

文章编号: 1671-1041(2005)01-0052-02

# USB 上位机程序开发与设计

童洪洁, 李宝华

(吉林大学 电子科学与工程学院, 长春 130061)

**摘要:** 本文介绍了在 VC++ 环境下 USB 上位机程序的开发和设计, 且利用 Windows API(Application Program Interface) 实现 USB 接口通信的方法。给出了多线程客户应用程序软件的实例。

**关键词:** USB;API;VC++;线程

**中图分类号:** TP311 **文献标识码:** B

## Develop and design of the host computer procedure on usb

TONG Hong-jie, LI Bao-hua

(Electronic Science and Engineering Department of Jilin University, Changchun 130061)

**Abstract:** This paper introduces the development and design of the host computer procedure on USB under the environment of VC ++, making use of Windows API (Application Program Interface) to realize USB interface communication method. Provide the instance of many thread customer's application program software.

**Key words:** USB;API;VC++;THREAD

USB(通用串行总线)现在逐渐成为各种新型设备的通用标准总线,并且大有取代传统的 RS-232 趋势。USB 在速度有 RS-232 无法匹敌的优势,同时随着 USB 的发展,从 1995 年提出 USB 标准,经历了 USB1.0、USB1.1、USB2.0,到现在 USBOTG 规范,USB 的传输速度也得到迅猛的提升。从 USB1.1 设备的传输速度 12Mbps/s 到 2.0 设备的 480Mbps/s 的发展,同时 USB 设备即插即用的特点,使其应用更加广泛并且超过了目前很多工业接口总线标准,因而它的应用越发重要。

然而 USB 在应用上给我们带来极大方便时,在软件处理上也给我们造成了不小麻烦。由于 USB 设备是基于通信协议的接口设备,使得我们不能在电脑上像处理 RS-232 那样来处理 USB。USB 设备需要驱动程序的支持,而 RS-232 则不需要,而且 USB 设备是从设备,任何的 USB 传输都必须从电脑里的 USB 主机开始发送指令,虽然绝大部分的 USB 设备厂商都给出了驱动程序但是应用接口还需要客户自行设计,从而使得上位机的编程变得尤为重要。在 VC++ 环境下,传统的 RS-232 可以直接使用相应的控件直接与外设进行通信,而 USB 不具备相应的控件,不过我们可以利用 VC 的环境下的 Windows API 函数把 USB 接口当作文件来操作,从而轻而易举的实现 USB 接口通信。

### 1 USB 接口通信

目前大多数驱动程序都采用了 WDM(Window Driver Model),从而我们可以采用两种方式打开设备:(1)GUID 接口方式;(2)符号连接名方式。在采用 GUID 接口方式之前,一定要获取此设备的 GUID,在实际开发中可以从驱动程序或者利用 Windows API 函数获取。以下为主要步骤:

#### 1.1 获取设备信息集

```
SetupDiGetClassDevs(
```

```
pGuid, // 指向一个安装类或接口类的 GUID
NULL, // 指向字符串,过滤返回的设备
NULL, // 返回顶层窗口句柄
DIGCF_INTERFACEDEVICE|DIGCF_PRESENT
// 给出构造设备信息集的控制操作
)
```

SetupDiGetClassDevs 函数是一个设备信息集的处理程序,它包含了指定参数匹配的所有已安装的设备。

#### 1.2 识别接口信息

```
SetupDiEnumDeviceInterfaces(
hardwareDeviceInfo, // 指向接口设备信息集
0, // 指向一个 SP_DEVINFO_DATA 结构
pGuid, // 指向一个 GUID,它标识了请求接口的设备接口类
i, // 设备接口表中基于零的索引
&deviceInfoData) // 分配给调用程序的缓冲区
```

SetupDiEnumDeviceInterfaces 函数返回设备信息集的一个设备接口元素的环境结构,每次调用该函数返回一个设备接口的信息。

#### 1.3 得设备路径名

```
SetupDiGetDeviceInterfaceDetail(
HardwareDeviceInfo, // 指向设备信息集
DeviceInfoData, // 指向标识设备的接口
functionClassDeviceData, // 指向接受接口信息
predictedLength, // 接受缓冲区大小
NULL) // 指向缓冲区接受请求接口的设备信息
```

#### 1.4 设备设置

在应用程序和驱动程序中,最常用的 API 函数为 CreateFile, ReadFile, WriteFile, DeviceIoControl, CloseFile 等等,其中 CreateFile, CloseFile 函数是必须用到的,其他的根据具体需要而定。

```
hFile=CreateFile(
functionClassDeviceData->DevicePath, // 打开设备名
GENERIC_READ | GENERIC_WRITE, // 允许读写访问文件
FILE_SHARE_READ|FILE_SHARE_WRITE, // 读写共享模式
NULL,
OPEN_EXISTING, // 打开而不是创建
FILE_ATTRIBUTE_NORMAL|FILE_FLAG_OVERLAPPED, // 以异步
方式打开文件
NULL);
```

如果上述函数返回成功,这时我们就可以进行读写操作。一般情况下我们采用 ReadFile, WriteFile 就足够了,如果我们需要在驱动中加入控制部分的时候,就需要使用 DeviceIoControl 函数。通常我们都采用同步方式打开文件,但是当我们的程序需要处理多个事务时,以同步方式打开接口就会造成接口函数的阻塞,只有等到驱动程序完成相应的数据传输操作时才能往下执行。由于目前市面上比较流行的 USB 设备芯片都是多端点通信,但是他们本身只能准双工通信,在数据传输的协调方面至关重要。因此当我们采用异步通信就可以很好的协调程序事务。

### 2 客户软件设计

收稿日期: 2004-11-12

当我们按照前面的流程设计好,就可以考虑应用程序了。通常程序除了保存传输而来的数据以外,更多的是数据的处理和加工。为了使各种事务的协调我们采用多线程技术,所谓线程就是操作系统分配处理器时间的最基本单元。利用 MFC 框架结构,我们能够加快程序的开发周期。

## 2.1 向端口读数据

```
ReadFile(hFile,          //USB 端口句柄
threadParam->pcIoBuffer, // 数据缓冲区
threadParam->uiLength,   // 数据的长度
&nBytes,                // 实际返回数据字节数
&ol)                    // 一个指向 OVERLAPPED 结构变量采用异步通信
```

## 2.2 创建线程

```
CwinThread *ReadDoc=AfxBeginThread(
ReadPipe,          // 指向工人线程的控制函数指针
&m_ConRead         // 传递给线程的控制函数参数
0,                 // 默认线程优先级
0,                 // 默认线程堆栈大小
0,                 // 控制线程的附加标志
0)                 // 默认安全属性
```

## 2.3 线程间通信与同步

为了保证线程间的数据能正确传递,通常采用全局变量、CEvent 类、自定义消息来联系。同时要保证各线程间的数据处理是相互协调,我们不能在同一时间访问同一数据时,既往里面写数据,同时又读数据。所谓的同步就是保持在同一个进程内的线程工作协调一致。常用的线程同步方法有 Critical Section、互斥对象(Mutex)、信号量(Semaphore)。

## 2.4 异步控制函数

```
UINT ReadPipe(void * pParam)
{
    for(;;)
    {
        criticalSection.Lock(); // 锁定共享数据区 threadParam->pcIoBuffer
        bResult = ReadFile(hFile, threadParam->pcIoBuffer,--- &ol);
        criticalSection.Unlock(); // 开锁,允许其他线程访问共享数据区
        if(bResult!=TRUE) // 判断是否打开文件
```

```
{
if(GetLastError()!=ERROR_IO_PENDING)
{
    ----- // 出错返回
}
while(WaitForSingleObject(ol.hEvent,100)==WAIT_TIMEOUT)
{
    -----// 超时处理
}
if(!::GetOverlappedResult(hFile,lpOverlapped,&nBytes,FALSE))
{
    continue;
}
Ep2Update.SetEvent();
// 通过 CEvent 类对象 Ep2Update 通知其他正在等待数据的线程
-----}
-----}
-----}
```

## 3 总结

上述只给出了上位机软件设计主要思路与流程,并重点介绍了接口设置和线程的调用,有了这些我们基本上能满足绝大部分客户软件的后台需求,我们可以很容易的实现从 USB 口读出数据,而后进行自己所需要的数据处理。笔者利用 VC++ 环境下的 MFC 应用程序框架,开发了虚拟示波器软件。它能够在接受数据的同时,对数据进行分析,运算同时波形显示,最大化的利用了 USB 给我们带来的方便与快捷。●

### 参考文献

- [1] 周立功等. PDIUSB12 USB 固件编程与驱动开发. 北京:北京航空航天大学出版社, 2003.
- [2] 许永和等. EZ-USB FX 系列单片机 USB 外围设备设计与应用. 北京:北京航空航天大学出版社, 2002.
- [3] 武安河等. Windows 2000/XP WDM 设备驱动程序开发. 北京:电子工业出版社, 2003.
- [4] 同志工作室. Visual C++6.0 开发技巧与实例教程. 北京:人民邮电出版社, 2000.
- [5] Jon Bates, Tim Tompkins 著. 何健辉等译. 实用 Visual C++ 6.0 教程. 北京:清华大学出版社, 2000.

作者简介:童洪浩,吉林大学电子科学与工程学院研究生,主要研究方向为计算机接口通信。通信地址:吉林大朝阳校区第四公寓,邮编 130061  
作者声明:自愿将本文稿酬捐为“仪器仪表用户杂志爱心助学基金”

文章编号: 1671-1041(2005)01-0053-03

# PC 在 Linux 下与 DSP 的异步串行通讯

周新林, 张继和, 余永立

(大连交通大学 电气信息学院, 辽宁 大连 116028)

**摘要:** 为了实现数据高速采集,开发了 PC 与 DSP 的串行通讯系统。采用 RS232 标准进行串行通讯,是由于它具有稳定的性能和有广泛的应用。DSP 的速度快,这有助于提高数据采集的速度;并且 DSP 内部集成了大量的外设,例如异步串行口,这使 DSP 与 PC 之间添加简单的接口就能实现异步串行通讯。Linux 操作系统有许多优点,在工业领域前景广阔。本文介绍了:RS232 串行通讯的特点;DSP 与 PC 的接口的实现;DSP 中异步串行口的设置及程序流程;在 Linux 操作系统中串口编程。

**关键词:** RS232, 串行通讯, 异步串行口, DSP, Linux 操作系统  
**中图分类号:** TP29 **文献标识码:** B

## Asynchronous serial communication between DSP and PC under Linux

收稿日期: 2004-09-22

ZHOU Xin-lin,ZHANG Ji-he,YU Yong-li  
(Dalian jiaotong university,Dalian 116028)

**Abstract:** In order to realize high speed data sampling ,developing serial communication between PC and DSP.Using the standards of RS232 because of its steady performance and widely application.DSP has high speed,it is useful for high speed data sampling; there are many peripherals in DSP chips,such as asynchronous serial port,it is easy to add simply interface between PC and DSP for asynchronous serial communication. Linux operating system has many merits so that it has wide prospect in industry.This paper refer to:the characteristics of RS232 serial communication; realizing interface between PC and DSP ;setting asynchronous serial port in DSP and promgram flow of DSP; programing for serial port in Linux operating system.

**Keywords:** RS232;serial communication;asynchronous serial port; DSP;Linux operating system