

Implementing Serial Communication in Windows CE

在 Windows CE 下实现 串口通信

摘要: 本文首先简单介绍了在 Windows CE 下设备驱动程序的访问方式, 然后详细介绍了在 Windows CE 下实现串口通信的方法和过程。本文对于在 Windows CE 下开发串口通信应用有一定参考价值。

关键词: Windows CE 串口通信

1 引言

Windows CE 是一个 32 位、多任务、多线程, 也是争论最多的一个非常有趣的 Microsoft Windows 操作系统。它是专门为体积小、资源要求低、便携式的机器而设计的。Windows CE 作为目前掌上电脑的主流操作系统, 它只是具有 Win32 子集 API, 但是其中处理通信的 API 却很丰富。这对于在 Windows CE 下开发通信程序提供了便利。在 Windows CE 下, 串口通信作为一种重要的通信方式, 应用比较广泛。串口不仅仅是用来进行数据同步的。因为它是标准的, 所以, 它还可以接串口打印机、串口 Modem 等, 甚至是用户自己开发的基于标准串口的任意外设! 下面将详细介绍在 Windows CE 下实现串口通信的过程。

2 Windows CE 下访问设备驱动程序简介

2.1 设备驱动程序简介

在研究串行驱动程序前, 先简单地介绍一下 Windows CE 在一般情况下是如何处理驱动程序的。Windows CE 将设备驱动程序分为两个主要的组: 本地设备驱动程序和流设备驱动程序。

本地驱动程序是一些硬件所必需的, 并且是在 Windows CE 硬件被设计好以后由原始设备商创建的。具有本地驱动程序的设备有键盘、触摸屏等。流接口设备驱动程序可以由第三方生产商提供, 以支持添加到系统中的硬件。因为 Windows CE 系统通常没有像 ISA 和 PCI 总线那样的用于附加插卡的总线, 附加硬件通常通过 PCMCIA 来安装。在这种情况下, 设备驱动程序将使用由低级 PCMCIA 驱动程序提供的功能来访问 PCMCIA 中的插卡。

此外, 可能会编写一个设备驱动程序来扩充现有的驱动程序的功能。例如: 可能会编写一个驱动程序来提供通过串行链接传输的压缩或加密数据流。在这种情况下, 应用程序能访问加密的驱动程序, 然后该驱动程序使用串行驱动程序来访问串行硬件。

流接口设备驱动程序用三个字符名后跟一位数字来表示的。下面是一些三字符名称的例子:

- COM 串行驱动程序;
- ACM 音频压缩管理器;
- WAV 音频 Wave 驱动程序;
- CON 控制台驱动程序。

2.2 设备驱动程序的访问方法

Windows CE 下的应用程序是通过文件 I/O 函数 CreateFile, ReadFile, WriteFile 和 CloseHandle 访问设备驱动程序的。用 CreateFile 打开设备并使用设备名称作为五字符驱动程序名称 (3 个字符加 1 位数字再加上一个冒号)。驱动程序可以用所有不同访问权限打开: 只读、只写、读/写、或不可读写。

一旦设备被打开, 就可以用 WriteFile 向设备发送数据, 也可用 ReadFile 从设备读出数据。对文件进行操作时, 在 Windows CE 下的设备不支持重叠的 I/O。可以用函数 DeviceIoControl 向驱动程序发送控制字符。

3 Windows CE 下串口通信实现过程

下面介绍一个简单的点对点的对话程序, 它使用基本串行通信方式来连接两个 Windows CE 设备。窗口的大部分由接收文本窗口所占据。从另一个设备接收的文本将显示在这里。屏幕的下面是发送文本窗口, 在这里键入字符, 然后单击 ENTER 或点击 SEND 按钮, 则文本将被发送到另一个设备上。

3.1 基本串行通信

串行设备的接口是常规I/O驱动程序调用和与通信相关的具体函数组合。串行设备被视为用于打开、关闭、读和写串行端口的常规、可安装的流设备。为了配置端口，Win32 API支持一组通信函数，Windows CE支持大多数Windows NT或Windows 98所支持的通信函数。

3.2 打开和关闭串行端口

在所有的流设备驱动程序中，均使用CreateFile来打开串行端口设备。所使用的名称要遵循前面提到的准则，即COM后接要打开的COM端口号再加上一个冒号。冒号是Windows CE所必需的，它是为了区别于在Windows NT中用于设备驱动程序的命名规则。下面的程序行以读写方式打开COM1端口：

```
hSer=CreatFile(TEXT("COM1:"),  
GENERIC_READ|GENERIC_WRITE,0,NULL,  
OPEN_EXISTING,0,NULL)
```

必需将0传递到共享参数以及CreateFile的安全属性和模板文件参数中，Windows CE不支持设备的重叠I/O，因此不能在参数dwFlagAndAttributes中传递FILE_FLAG_OVERLAPPED标志。返回的句柄或者是已打开的串行端口的句柄，或者是INVALID_HANDLE_VALUE，如果在打开失败时，Create File不会返回0。调用CloseHandle可以关闭一个串行端口，方法如下：

```
CloseHandle(hSer);
```

3.3 读写串行端口

我们可以使用ReadFile和WriteFile函数写串行端口，如：

```
int rc;  
DWORD cBytes;  
BYTE ch;  
rc = ReadFile(hSer,&ch,1,&cBytes,NULL);
```

如果调用成功，变量ch将读入一个字节。

cBytes被设置为读取字节的数量。

写串行端口也很简单。调用过程如下：

```
int rc;  
DWORD cBytes;  
BYTE ch;  
ch = TEXT('a');  
rc = WriteFile(hSer, &ch, 1, &cBytes, NULL);
```

上面的代码将字母a写入以前已经打开的串行端口。如果成功，则ReadFile和WriteFile都将返回TRUE。由于Windows CE不支持重叠I/O，应该注意，不要试图从主线程或从已经创建了窗口的任意线程读或写大量的串行数据，因为这些线程还负责处理用于它们窗口的消息队列，不能阻塞它们，让它们等待相对较慢的串行读或串行写操作。实际上，应当用单独的线程来读或写串行端口。

还可以用下面的函数来传送单个字符：

```
BOOL TransmitCommChar(HANDLE hFile,  
char cChar);
```

TransmitCommChar和WriteFile的不同之处在于TransmitCommChar将要传输的字符放在传输队列的前面。当调用WriteFile时，字符将被排在还没有被串行驱动程序传输的任何字符之后，TransmitCommChar允许不用等待队列为空而在流中快速地插入控制字符。

3.4 异步串行I/O

虽然Windows CE不支持重叠I/O，但可以使用多个线程来执行同样类型的重叠操作，当主线程正忙时，需要做的就是运行单独的线程来处理同步I/O操作。在这个例子中，演示如何在读和写串行端口时使用单独的线程，要等待一个事件，首先用SetCommMask来设置事件掩码，可以用GetCommMask来查询当前事件掩码。

3.5 配置串行端口

读和写串行端口比较简单，但还必须对端



口配置好正确的波特率、字符长度等。这里介绍一种简单的方法。

可以用 SetCommState、GetCommState 两个函数来配置串行端口。这两个函数都包括已打开的串行端口的句柄和指向 DCB 结构的指针。用 SetCommState 能设置多种状态，因为 DCB 结构中的元素比较多，最好不要从头填写整个结构，而应该用修改串行端口的最好方法，即调用 GetCommState 来填充 DCB 结构，修改必须的区域，然后调用 SetCommState 来配置串行端口。

3.6 设置端口超时值

SetCommState 能调整串行驱动程序几乎所有的细微操作。然而仍须再进一步，设置端口的超时值。超时是在 ReadFile 或 WriteFile 自动返回前 WindowsCE 等待读或写操作的时间长度。控制串口超时的函数是：

```
BOOL GetCommTimeouts(HANDLE hFile,
LPCOMMTIMEOUTS lpCommTimeouts)和
BOOL SetCommT
```

```
imeouts(HANDLE hFile, LPCOMMTIME
OUTS lpCommTimeouts)
```

这两个函数的参数是已打开串行设备的句柄和指向 COMMTIMEOUT 结构指针。COMMTIMEOUT 结构定义如下：

```
typedef struct _COMMTIMEOUTS{
    DWORD ReadIntervalTimeout;
    DWORD ReadTotalTimeoutMultiplier;
    DWORD ReadTotalTimeoutConstant;
    DWORD WriteTotalTimeoutMultiplier;
    DWORD WriteTotalTimeoutConstant;
}COMMTIMEOUTS;
```

超时的计算方法有两种：第一种是 ReadIntervalTimeout 指定的在接收字符之间的最大时

间间隔。如果超过了这个时间，ReadFile 调用将立即返回。另一种方法是基于等待接收的字符的数量。ReadTotalTimeoutMultiplier 中的值乘以在调用 ReadFile 中请求的字符数，再加上 ReadTotalTimeoutConstant，就得到了调用 ReadFile 的总超时时间。

写超时可以被指定为只是在 WriteFile 调用中所花费的总时间。它的计算方法与计算总的读超时的方法相同。在这个结构中的时间都以毫秒为单位。

超时值是非常重要的。因为最坏的事情就是陷入一个循环中来等待来自串行端口的字符。当对 ReadFile 和 WriteFile 的调用正在串行端口等待时，调用线程将被有效地阻塞在驱动程序内部的事件对象中。这节省了在串行传送和接收操作中珍贵的 CPU 资源和电池能量。

3.7 保持活动

串行通信的问题之一是当串行链接是活动时防止系统掉电。Windows CE 系统由键盘的敲击次数和屏幕的点击次数来决定活动性。它不计入诸如串行端口传输数据之类的任务。为防止系统掉电，可以使用下面的两个函数中的任意一个来模拟键击：

```
VOID keybd_event(BYTE bVk,BYTE bScan,
DWORD dwFlags,DWORD dwExtraInfo);
UINT SendInput(UINT nInputs,LPINPUT
pInputs,int cbSize);
```

这些函数可以用来模拟键击。用键击能重置 Windows CE 所使用的活动计时器。Windows CE 使用这个计时器来确定系统何时自动断电。Windows CE 支持用于这两个函数的一个附加常量 ---KEYEVENTF_SILENT，它能防止在默认的键盘单击时发出声音。

下面是本程序的实例代码片段：

```
InitCommunication(HWND hWnd, LPTSTR
pszDevName)
//打开、初始化所选COM端口;pszDevName
是端口名称的地起址。
HANDLE hLocalDev;
HlocalDev = CreateFile(&pszDevName,
GENERIC_READ|GENERIC_WRITE,0, NULL,
OPEN_EXISTING, 0, NULL);
//配置端口
GetCommState(hLocal, &dcb);
dcb.BaudRate = CBR_19200;
dcb.fParity = FALSE;
dcb.fNull = FALSE;
dcb.StopBits = ONESTOPBIT;
dcb.Parity = NOPARITY;
dcb.ByteSize = 8;
SetCommState(hLocal, &dcb);
//设置延时
cto.ReadIntervalTimeout = 0;
cto.ReadTotalTimeoutMultiplier = 0;
cto.ReadTotalTimeoutConstant = 0;
cto.WriteTotalTimeoutMultiplier = 0;
cto.WriteTotalTimeoutConstant = 0;
SetCommTimeouts(hLocal, &cto);
//向串口中发送字符
SendThread(PVOID pArg);
//发送的线程
GetWindowText(hwndSText, szText, dim
(szText));
WriteFile(hComPort, szText, lstrlen(szText)
* sizeof(TCHAR), &cBytes, 0);
SetWindowText(hwndSText, TEXT("")); //
清屏
//当按 SEND 按钮时，调用这个线程。
```

//从串口中读字符

ReadThread(PVOID pArg);

//读线程

ReadFile(hComPort, pPtr, 1, &cBytes, 0);

把接收的字符经过适当的处理后,用
SendDlgItemMessage 函数发到接收文本框中。

这个线程在初始化端口时用 CreateThread 建立:

hReadThread = CreateThread(NULL, 0,

ReadThread, hWnd, 0, &dwTStat);

4 结论

通过上述介绍,我们可以了解到在Windows
CE 下进行串口通信的主要实现过程和方法。本
文虽然只是起到抛砖引玉的作用,但对于从事
PDA 开发,实现基于Windows CE的串口通信有
一定参考价值。 ■

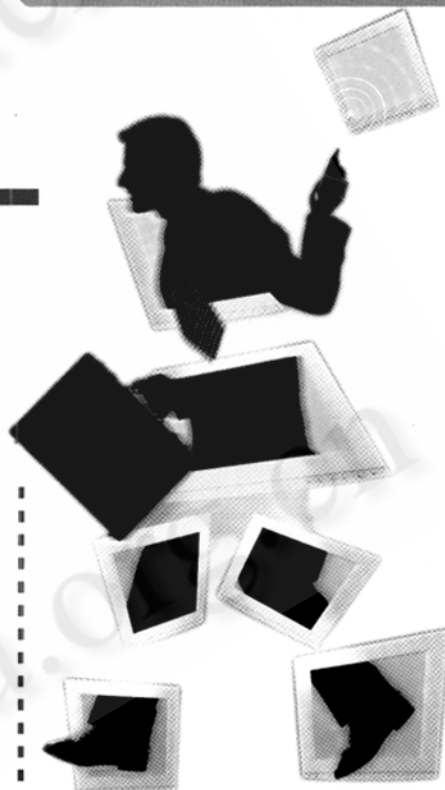
参考文献

- 1 Douglas Boling, Windows CE程序设计,北京
大学出版社,1999。
- 2 David J Kruglingski(美),潘爱民,王国印译,
Visual C++ 技术内幕,清华大学出版社,1999。
- 3 Nick Grattan, Marshall Brain, Windows CE
3.0 Application Programming,Microsoft
Corporation, 1997-1999。

引导企业需求, 创造客户价值

企业应用软件盛会将于今年6月在沪举行

由上海市人民政府信息化办公室,上海软件行业协会主办及多家IT媒体承办的“2003年上海国际软件行业年会”(CISE2003)以及相关专业展览和论坛将于今年6月下旬在上海举行。年会针对不同的行业应用特点举办不同主题的行业活动,其中包括2003年制造业信息技术应用展览及会议、2003年供应链及物流应用技术展览和会议、2003年电子政务应用技术展览和会议、2003年企业应用软件及开发技术展览和会议四大专题活动,做为今年亚太城市信息化高级论坛(APCITY2003)的系列活动之一,已连续五届成功举办的软件年会活动将继续以“引导企业需求,创造客户价值”为主旨,通过软件产品及方案的现场展示,主题演讲大会及多场开放式研讨会等多样的交流形式,为众多渴望了解,实施应用解决方案的企业及行业用户提供了一个全面了解业界领先的软件产品,技术及系统方案的良机。主办单位将在本次活动中为参展公司提供各类增值服务如定期软件需求安装企业信息服务,IT媒体和行业媒体立体宣传服务(包括免费广告报道与高级专访),行业信息与咨询服务,全部参加企业代表数据库和行业数据库服务及优质展览会及论坛整体宣传等,同时希望通过整合政府及各协办单位的资源从而更好的为参加公司和企业用户服务,有关活动的详细情况可浏览主题网站 www.softwarevent.org 或拨打咨询热线 **021-64477681/85/87**。



倪鸣

上海软件行业协会会务部,上海霖科

TEL: 021 - 64477680/81/82

FAX: 021 - 64477680

EMAIL: marknee@softwarevent.org

Web: www.softwarevent.org

2003年上海国际软件行业年会
引导企业需求,创造客户价值。
2003年6月

引导企业需求, 创造客户价值! 更多精彩内容, 尽在 www.softwarevent.org