

Windows 环境的串口异步通信程序设计

李阜 陈小欧 (华中理工大学机械学院 430074)

摘要:本文讨论了串行异步通信有关的技术及 Windows 环境下利用 VB 进行串行异步通信程序的设计方法。

关键词:串行异步通信 流控制 Windows VB

在工业过程检测和控制系统中,通信是实现两端设备交换数据的主要手段,而 Windows 以其多任务、事件驱动和高级的图形用户界面(GUI)等特点受到广泛应用,所以 Windows 环境下通信程序的设计成为一个关键问题。但 Windows 不支持对系统 I/O 端口的直接访问,在 Windows 环境下实现通信的一个简便方法就是利用 Windows 中已提供驱动程序支持的串行口进行串行通信。虽然调用 Windows API 函数可以实现上述功能,但其使用起来复杂繁琐,更简单的方法是利用 VB 中的通信控制。本文就相关内容作些介绍。

1. 串行驱动程序的结构

Windows 的串行通信驱动程序是中断驱动的。每接收到一个字符,这个字符就会产生一个低级硬件中断,串行驱动程序会立即取得控制权并将字符放入用户定义的输入数据缓冲区,然后再将控制返回给正在运行的那个应用程序。如果输入缓冲区满了,驱动程序就用当前定义的流控制机制通知其他设备停止发送数据。

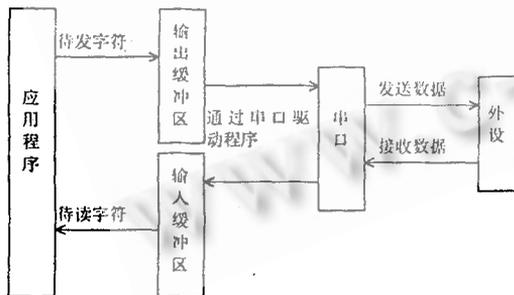


图 1 串行驱动程序的结构

发送数据的处理类似于接收数据的处理方法。应用程序把要发送的数据放入输出缓冲区,每当串口硬件发送一个字符后就会产生一些低级硬件中断,这时如果输出缓冲区中还有待输出的字符,串行驱动程序就会将此

字符送到串口以备输出。图 1 说明了这个处理过程。

2. 程序设计

VB 是 MicroSoft 公司在 1991 年推出的与 C 语言相当的 Windows 应用程序开发工具,VB 的“可视化”和“面向事件”的特性是简化 Windows 程序开发的关键。越来越多的人运用 VB 开发 Windows 应用程序。VB 的专业版包含了 MSCOMM.VBX 用户控制(Custom Control),它提供了在其中通过串口发送和接收数据的串口通信能力,使用户很方便地访问 Windows 串行通信驱动程序的大多数特性。

(1) 使用 MSCOMM.VBX

从 VB 工具箱(ToolBox)中可以看到通信控制,它象一个电话的话筒。把这个控制移到 VB 的表格(Form)上,即可用它来进行通信。若在 VB 工具箱中没有这个控制,则用 File 菜单的 Add File 命令把 MSCOMM.VBX 从 Windows 的 System 子目录下加到 VB 工具箱中。通信控制把许多复杂的操作留给 VB 和 Windows 处理,用户使用的关键是正确设置其中的一些属性(Property)。虽然通信控制有许多重要的属性,但需要首先熟悉的是如下几个属性:

CommPort 设置或返回通信口的编号,如 1 为 COM1, 2 为 COM2。格式为: [form.]Mscomm.CommPort [= portNumber%]。可以在设计时把 portNumber% 设为 199 的任何整数,但若这个口不存在,当用 PortOpen 属性来打开口时,就会产生错误。还要注意的是在用 PortOpen 属性来打开口前,必须先设置 CommPort 属性。缺省为 1(即用 COM1)。

Settings 设置或返回波特率、校验、数据位、停止位等参数。格式为: [form.]Mscomm.Settings [= ParamString\$], ParamString\$ 是一个包含四部分的字符串,如“9600,N,8,1”,其中第一部分为波特率,其有效值可以是 110、300、600、1200、2400、4800、9600、14400、19200

等;第二部分为校验, N 表示不校验, E 表示偶校验, O 表示奇校验, S 表示空格校验, M 表示符号校验(Mark);第三部分为数据位位数, 可以是 4、5、6、7、8;最后一部分是停止位位数, 可以是 1、1.5、2, 缺省为“9600、N、8、1”。

PortOpen 设置或返回通信口的状态(TRUE:打开或 FALSE:关闭)。设计时不可见。格式为: [Form.] Mscmm. PortOpen [= TRUE/FALSE]。要注意的是在用 PortOpen 属性来打开口前, 必须先设置正确的 CommPort 属性值。

Input 从接收缓冲区取出并返回字符串。格式为: [Form.] Mscmm. Input

Output 向发送缓冲区写入字符。格式为 [Form.] Mscmm. Output = Outstring \$, Outstring \$ 为要发送的字符串。

在一个表格(Form)中加入通信控制, 假定其控制名为 Mscmm1, 再在 Form-Load() 事件过程中加入如下几行程序后, 即可用 Input 和 Output 性质通过串口 1 来接收和发送字符了:

```
Sub Form-Load( )
Mscmm1. CommPort = 1
Mscmm1. Settings = "9600, N, 8, 1"
Mscmm1. PortOpen = True
...
End Sub
```

这时要考虑的一个问题是: 发送时要知道输出缓冲区是否已满, 读字符时要知道输入缓冲区是否有字符, 否则会丢失字符或有错误发生。这种情况就牵涉到下边几个性质的使用:

InBufferSize 设置或返回接收缓冲区字节值, 格式为: [form.] Mscmm. InBufferSize [= number%]。number% 为缓冲区大小的字节值。在没有流控制的情况下如果接收缓冲区太小, 就有可能发生缓冲区溢出的危险; 太大又会占用较多的内存空间, 要设为合理的值。缺省为 1024 字节。

InBufferCount 返回接收缓冲区中已接收字符的个数, 设计时不可见。其格式为: [form.] Mscmm. InbufferCount [= 0]。设置 InBufferCount 为 0, 则清除接收缓冲区。

OutBufferSize 设置或返回发送缓冲区字节值。格式为: [form.] Mscmm. OutBufferSize [= number%]。在没有流控制的情况下如果发送缓冲区太小, 就有可能发生

缓冲区溢出的危险; 太大又会占用较多的内存空间, 要合理设置。缺省为 512 字节。

OutBufferCount 返回发送缓冲区中等待送出的字符个数, 设计时不可见。格式为: [form.] Mscmm. OutbufferCount [= 0]。设置 InBufferCount 为 0, 则清除接收缓冲区。

InputLen 设置或返回的是用 Input 性质从缓冲区读字符串时每次读出的字符个数。这个性质对于读出数据块中定长数据串十分有用。格式为: [form.] Mscmm. InputLen [= numChars%]。设 InputLen 为 0, 则用 Input 读缓冲区时, 通信控制一次读出整个缓冲区内容, 把缓冲区读空。InputLen 属性的缺省设置是 0。需要注意的是如果输入缓冲区中的字符数没有 InputLen 所设置的那么多, 用 Input 属性就会返回一个零长度的字符串(""), 所以用户在用 Input 读数据前应该检查 InBufferCount 属性, 来确定是否有所要求的字符数。

输出缓冲区是否已满, 要用 OutBufferSize 减 OutBufferCount 的值来判断, 而输入缓冲区是否有字符要用 InBufferCount 来判断。

(2) 事件驱动

通信控制提供了两种方法来处理通信。上边所说的用 InbufferCount 或 OutBufferCount 性质来判断处理属于查询的方法。而事件驱动的通信是处理串口间通信的一个强有力的方法。在很多情况下只想在某一事件发生时得到通知, 如一个字符接收到或 CD(Carrier Detect) 线上发生了变化或 RTS(Request To Send) 线上发生了变化。在这种情况下, 用户可以用通信控制的 OnComm 事件来捕获和处理这些事件。OnComm 事件也可以检测到通信中发生的错误。具体发生了什么事或错误, 由通信控制的 CommEvent 属性来判断。

OnComm 通信控制的事件。每当通信有错误或某事件发生时, 通信控制就会产生此事件。但是事件或错误的数字代码放在 CommEvent 属性中。

CommEvent 返回最近的通信事件或错误的数字代码。设计时不可见, 运行时只读。格式为: [form.] Mscmm. CommEvent。其事件或错误的代码放在 VB 目录下的 CONSTANT.TXT 文件中。

具体通信控制中什么事或错误会引发 OnComm 事件, 要由不同的设置来控制。与此密切相关的两个属性是 RThreshold 和 SThreshold。

RThreshold 设置或返回通信控制把 CommEvent 属

性值设为 MSCOMM-EV-SEND 并引发 OnComm 事件之前,在发送缓冲区中允许有的最小字符数。例如,把 RThreshold 性质的值设为 1,则缓冲区中每接收到一个字符就把 CommEvent 的值置为常量 MSCOMM-EV-RECEIVE,且引发 OnComm 事件,在这个事件过程中就可以捕获 MSCOMM-EV-RECEIVE 事件,进行必要的处理。设为其他值则每当输入缓冲区中有相应数量的字符时便把 CommEvent 的值置为 MSCOMM-EV-RECEIVE,且引发一 OnComm 事件,这对于成块传送有利,不必每接收到一个字符就产生一次 OnComm 事件。设为 0,则不引发 OnComm 事件。缺省为 0。

SThreshold 作用与 RThreshold 类似,但又不同。设置 SThreshold 的值为 0(缺省值),通信控制在发送数据时不会引发发送数据 OnComm 事件。设为其他值,则在发送缓冲区中的字符数从多于降到少于 SThreshold 时,CommEvent 属性就会被设置为 MSCOMM-EV-SEND,并且引发 OnComm 事件。即缓冲区中的字符数的值经过 SThreshold 时, MSCOMM-EV-SEND 事件会引发一次。例如,若 SThreshold 等于五,那么 MSCOMM-EV-SEND 事件只有当发送缓冲区中的字符数从大于五降到四时才会发生。如果发送缓冲区中的字符数从来没有超过 SThreshold,则事件就永远不会发生。

(3)其他属性及相关内容

另一些有用的性质有 Handshaking, ParityReplace, NullDiscard。

Handshaking 设置或返回握手协议。格式为:
[form.]MSComQD

m.Handshaking [= portocol%]。握手协议指的是数据从硬件口传送到接收缓冲区中的内部通信协议,它确保当数据快速达到串行口,而缓冲区将溢出,通信驱动程序来不及把数据送入缓冲区时不会丢失数据。流控制(也称作握手协议)有两种,即硬件流控制和软件流控制。

ParityReplace 设置或返回当数据流中出现检验错时代替出错字符的字符。设 ParityReplace 属性值为空字符串(""),则不进行校验。格式为:[form.]MSComm.ParityReplace [= char\$]。缺省时,通信控制用问号(?)代替出现校验错的字符。

NullDiscard 设定串行口接到空字符(ASCII 为 0 的字符)时是否把它送入接收缓冲区。设为 True 则口接收到空字符不送入接收缓冲区;设为 False 口接收到空字符会送入接收缓冲区。缺省为 False。格式:[form.]MSComm.NullDiscard[= True / False]。

要注意的是 Windows 事实上是一个多任务的操作环境,这就必须有一种机制来保证在任意时刻只有一个应用程序拥有串行口的访问权。所以在不用串行口时要把口关闭,使其他应用程序也可以用这个口。如果在程序中需要同时访问多于一个的串行口,就必须用多于一个的通信控制。串口的口地址和中断地址可以通过 Windows 的控制面板来改变。

MSCOMM.VBX 通信控制还有一些与 RTS、CTS、DTR、DSR 和 CD 信号线等有关的属性,但对于一些简单的任务来说,只要两根数据线和一根背景信号线三根信号线就行了,则许多与这些信号线有关的属性是没有用的,并且 VB 对与这些信号线有关的性质的缺省设置为不用,所以可以不管这些性质。

```

主窗口文件清单 Form1.frm
Sub Comm1-OnComm ( )
    Select Case Comm1.CommEvent '根据不同的 CommEvent 属性值来确定引起 OnComm()事件的具体原因,然后进行不同的处理。
        Case 2 'MSCOMM-EV-RECEIVE = 2
            If Len(text2.Text) > 2048 Then
                text2.Text = ""
            End If
            '如果 text2 里边的文本太长,则把它清空.在这里设定的最大长度为 2k 字节
            text2.SelText = Comm1.Input
            '把从缓冲区中读出的字符串放在 text2 文本之后
        End Select
    End Sub
Sub port-Click ( )
    '选择 Setting 菜单的 port 菜单项,显示设置窗口
    comm.Show
End Sub
Sub Text1-KeyPress (keyascii As Integer)
    '在文本框 text1 中输入文本时,按回车键,则把文本框 text1 中的文本输出,然后清除里边的文本
    If keyascii = 13 Then '回车键被按下
        keyascii = 0
        Form1.Comm1.Output = text1.Text + Chr$(13) + Chr$(10)
        text1.Text = ""
    End If
End Sub
设置窗口文件清单 COMM.FRM
Dim sbaudrate As String
Dim sdata As String
Dim sstop As String
Dim sparity As String
Dim sport As String

```

```

Sub Command1-Click ( ) 'OK button click
On Error GoTo error0
For i = 0 To 3
If option1(i).Value = True Then
'option1( )为框架(Fram)Baud Rate 中的选择按钮
(Option Button)组,根据其索引值(Index)来选择波特率。
Select Case i
Case 0 sbaudrate = "1200"
Case 1 sbaudrate = "2400"
Case 2 sbaudrate = "4800"
Case 3 sbaudrate = "9600"
End Select
End If
Next i
For i = 0 To 4
If option3(i).Value = True Then
'Option3( )为框架 Parity 中的选择按钮组,根据其
索引值(即 i)来选择校验方法。
Select Case i
Case 0 sparity = "N"
Case 1 sparity = "O"
Case 2 sparity = "E"
Case 3 sparity = "M"
Case 4 sparity = "S"
End Select
End If
Next i
For i = 0 To 3
If option6(i) = True Then 'option6( )为框架 Data
Bits 中的选择按钮组,用来选择数据位。
Select Case i
Case 0 sdata = "5"
Case 1 sdata = "6"
Case 2 sdata = "7"
Case 3 sdata = "8"
End Select
End If
Next i
For i = 0 To 2
If option5(i).Value = True Then
'option5( )为框架 Stop Bits 中的选择按钮组,用来
选择停止位。
Select Case i
Case 0 sstop = "1"
Case 1 sstop = "1.5"
Case 2 sstop = "2"
End Select

```

```

End If
Next i
form1.Comm1.Settings = sbaudrate + "," + sparity + "," + sdata + "," + sstop '设置通信控制的 Settings 属性值
For i = 0 To 1
If option2(i).Value = True Then
'option2( )为框架 Port 中的选择按钮组,用来选择
通信口
Select Case i
Case 0 sport = "1"
Case 1 sport = "2"
End Select
End If
Next I
If form1.Comm1.PortOpen = True Then
form1.Comm1.PortOpen = False '在口是打开的
情况下再打开口会出现错误,所以若口是打开的就先把
口关闭
End If
form1.Comm1.CommPort = sport
form1.Comm1.PortOpen = True
unload comm '关闭对话框
Exit Sub
error0: '错误处理过程。这里是把错误信息显示出来,若想对错误进行进一步的处理,要在此根据错误号
Err 进行相应的处理
msg = "The error message for error number "
msg = msg & Err & " is:" & CHR$(13) &
CHR$(10)
msg = msg & """" & Error(Err) & """"
MsgBox msg ' Display message.
Exit Sub
End Sub
Sub Command2-Click ( ) 'Cancel click
unload comm '单击(Click)Cancel 时什么也不作,
只把窗口关闭
End Sub
Sub Form-Load ( )
'调入缺省的设置值
option1(3).Value = True '波特率为 9600
option2(0).Value = True '用 COM1
option3(0).Value = True '不校验
option5(0).Value = True '1 个停止位
option6(3).Value = True '8 个数据位
End Sub

```

(来稿时间:1996年9月)