

用 VB 6.0 实现工控软件和下位机的串口通信

阚 江,孙苓生,张 明
(上海交通大学 电气工程系,上海 200030)

摘要: 针对自动化控制中出现的工控软件串口通信驱动程序不支持现有设备的问题,基于 DDE (Dynamic Data Exchange) 技术,用 VB 6.0 实现了一个能被绝大多数工控软件使用的串口通信接口程序。对下位机,利用 MSComm 控件收发串口数据;对工控软件,通过 DDE 方式与其交换数据,使得工控软件能透明地操作串口数据。实际工程运行表明,该方法实现简单,功能适用,运行可靠。另外,在工业现场的实际应用常常需要定时发送数据,而 VB 提供的定时器控件只能精确到 55 ms 左右,利用 Win API 实现了精度 1 ms 的高精度定时器。

关键词: 工控软件; 串口通信; 动态数据交换; 定时器

中图分类号: TP 319; TN 915.04 **文献标识码:** A **文章编号:** 1006-6047(2002)09-0034-04

目前常用的各类工控软件(InTouch, Fix, Cimplicity 等)都针对各大公司的知名产品或常用协议提供了很好的通信驱动程序,极大地方便了工程技术人员的应用和开发。但在自动控制的实际应用中,还经常出现工控软件提供的通信驱动程序不支持现有设备的问题。这一现象在采用非通用性设备时尤为普遍,而这些非通用性设备大多是通过串行口与上位机通信的。本文详细阐述了利用 VB 6.0 开发适用于多种工控软件的串口通信接口程序的原理及具体实现。

1 串口通信

在 Windows 下实现串口通信主要有 2 种方法^[1]:

- a. 直接调用 Windows API (Application Program Interface);
- b. 利用微软提供的 ActiveX 控件 MSComm (Microsoft Comm Control)。

第 1 种方法不仅要声明和调用大量 API 函数,还涉及到进程调度问题,十分烦琐,仅适用于 VC 开发环境。第 2 种方法实际上是通过 MSComm 控件间接调用了 API,而用控件使程序实现简便,并能满足多数情况下的工控要求。充分利用已有的 ActiveX 控件实现快速开发正是 VB 的优点之一。

2 和工控软件的通信

设计的串口通信程序作为工控软件和串口设备之间的一个接口,在收发串口数据的同时,还要和工控软件进行数据交换。和工控软件的数据交换涉及到进程间的通信(IPC)。Windows 下进程间通信主要方式^[2]为

- a. 动态数据交换 DDE (Dynamic Data Exchange);

- b. 使用剪贴板;

- c. 使用动态链接库 DLL (Dynamic Link Library);

- d. 使用对象链接和嵌入 OLE (Object Link Embedded)。

具体选择某种方式,需要综合考虑各种工控软件对数据交换方式的支持,以及工业控制的数据传输特点。剪贴板是一种静态交换数据的途径,DLL 不被大多数工控软件支持,而 OLE 不适合串行方式传输的数据;相比之下,DDE 则是一种简单、高效而又被工控软件广泛支持的数据交换方法。

DDE 用共享内存交换数据,用协议进行同步数据的传递。DDE 协议是一种开放的、与语言无关的、基于消息的协议,它允许多个应用程序实时地交换数据和命令。而对于 DDE 热链接,当数据更新时不需用户参与即可自行完成数据交换。更重要的是,通过 DDE 协议,应用程序之间无需接口,仅需与操作系统接口。这种灵活性使 DDE 成为 Windows 应用程序普遍支持的一种接口协议,也被几乎所有的工控软件支持,像 Intouch 更是以 DDE 通信为核心。

DDE 的基本概念是会话 (Conversation)。一次会话调用 2 个应用程序即服务器 (Server) 和客户 (Client)。串口通信程序向工控软件提供串口数据的服务,是服务器端,又称为发送端;而工控软件向串口通信程序发出请求,并接收其发送的数据,是客户端,又称为接收端。

DDE 协议用三层命名标识一个对话即服务器名 (Server Name),主题名 (Topic Name) 和项目名 (Item Name)。服务器名也称为应用程序名 (Application Name)。

3 程序实现

3.1 串口通信的程序实现

利用 MSComm 控件实现串口通信的具体操作已

收稿日期:2001-12-12

有很多文献论述,在此只强调一点,即无论发送还是接收,都要通过一个 Variant 类型变量做“中介”^[3]。

发送时,先将要发送的字符串(传送文本格式时)或单字节数组(传送二进制格式时)赋给一个 Variant 类型变量,再把该 Variant 变量赋值给 MSComm 的 Output 属性;接收时反之,先将 MSComm 的 Input 属性赋值给 Variant 变量,再根据不同情况赋值给字符串或单字节数组。

串口发送的例程如下:

```
Private Sub TransmitData()  
    Dim bytSend() As Byte, varTemp as Variant  
    ...;将待发送数据存入 bytSend() 数组  
    MSComm1.CommPort = 1;使用 COM1  
    MSComm1.Settings = "9600,N,8,1";9600 波特,无奇偶校验,8 位数据,1 位停止位。  
    If Not MSComm1.PortOpen then;打开串口  
        MSComm1.PortOpen = True  
    End if  
    varTemp = bytSend;利用 varTemp 为中介  
    MSComm1.OutBufferCount = 0;清除发送缓冲区  
    MSComm1.Output = varTemp;发送打包后的数据  
    ...;                其他处理  
    MSComm1.PortOpen = False;  
End Sub
```

3.2 DDE 通信的程序实现

在 VB 中实现 DDE 通信很简洁。尽管和 DDE 相关的属性、事件和方法有很多,但实现 DDE 通信所必须设置的属性只有 LinkMode,LinkTopic 和 LinkItem。若作为 DDE Server,则只需设置前 2 个属性。表 1 详细说明了这 3 个属性的意义。

需要注意的是设计时,若窗体的 LinkMode 设置为缺省值 0,则运行时不能再改变 LinkMode。若让窗体作为源,设计时必须设置 LinkMode 属性为 1,在运行时可以改变 LinkMode 的值。

串口通信程序作为 DDE Server,下面是其属性设置的具体操作。

a. 在 VB 开发环境下,在“工程属性”的对话框中指定工程名,如 SerialComm。这就是串口通信程序作为 DDE Server 的 Application Name。

b. 将程序主窗口的 LinkMode 属性设为 1,LinkTopic 可任意给定,如 DataPool。此属性值即为 DDE Server 的 Topic Name。

c. 在主窗口上添加 2 个 Label 控件,不妨分别命名为 lblReceive 和 lblTrans。lblReceive 和 lblTrans 就分别是 2 个 DDE 对话的 Item Name。

完成设置后,串口通信程序就可通过 lblReceive 与 lblTrans 和工控软件交换数据了。从 VB 的角度看,lblTrans.Caption 就是工控软件要求发送的串口数据,而把从串口读入的数据赋值给 lblReceive.Caption,就相当于把串口输入的数据传递给了工控软件。为了不影响程序界面,可将这 2 个 Label 的 Visible 属性设为 False。DDE 属性说明如表 1 所示。

表 1 DDE 属性说明

Tab. 1 The main properties for DDE

属性	发送端(窗体)	接收端(控件)
	返回或设置用于 DDE 会话的链接类型并同时激活相应的链接,允许目标应用(客户端)启动与窗体的会话。 0:无 DDE 交互(缺省值)。 1:允许窗体上任何 Label, PictureBox 或 TextBox 控件为与该窗体建立 DDE 会话的客户端提供数据。如果存在这种链接,Visual Basic 在控件内容改变时会自动提醒目标应用程序。另外,目标应用程序能够向窗体上的 Label, PictureBox 以及 TextBox 控件存放数据。	返回或设置用于 DDE 会话的链接类型并同时激活相应的连接,允许该控件作为接收端启动会话。 0:无 DDE 交互(缺省值)。 1:自动——每次链接数据改变,目标控件都要更新。 2:手动——只有激活 LinkRequest 方法时,才更新目标控件。 3:通知——链接数据改变时,会产生 LinkNotify 事件,但只在 LinkRequest 方法激活时才会更新目标控件。
LinkTopic	返回或设置在 DDE 对话中发送端窗体需响应的 Topic Name	返回或设置发送端 Application Name 和 Topic Name
LinkItem	无此属性	返回或设置发送端的 Item Name

将串口读入数据发送给工控软件的例程如下:

```
Private Sub MSComm1_OnComm();MSComm 的事件  
  
    Dim vTemp as Variant  
    Select Case MSComm1.CommEvent  
        Case comEvReceive;串口接收到数据  
            MSComm1.InputLen = 1;一次读取一个字节  
            vTemp = MSComm1.Input;读取串口数据  
            lblReceive.Caption = CLng(vTemp);将串口数据发送给工控软件  
        Case ...;                其他事件处理  
        ...  
    End Select  
End Sub
```

3.3 DDE 通信的改进

在实际应用中,考虑到工控软件的数据处理能力有限,应将通信协议的转换工作交由通信接口程序完成。串口通信程序根据协议提取出每一条指令或数据,再将这些数据和指令同时上传给工控软件。工控软件发送串口数据时,也是向串口通信程序同时发送所有的数据或指令,由串口通信程序实现数据的打包发送。

为此,对 VB 需开辟多个 Label 控件,对每个控件,始终存放通信帧数据段中的固定位置的一个或多个字节,如图 1 所示。

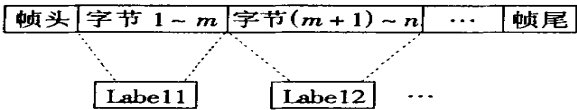


图 1 Label 和帧数据对应示意图
Fig. 1 The relationship between label and data frame

为使通信程序能适应更多的应用环境,可采用动态加载控件的办法,即在通信程序启动时,根据设定的帧长加载相应的 Label 控件数。但在采用变长帧的通信协议时,给通信程序设定的帧长应不小于可能出现的最大帧长。

为了实现动态加载 Label 控件,需建立控件数组。先在通信程序主窗口添加 2 个 Visible 属性为 False 的 Label 控件,分别命名为 lblReceive 和 lblTrans,再将 2 个控件的 Index 属性都设为 0,就能在程序中用 Load 方法动态添加 Label 控件。

实现动态加载 Label 控件的程序如下:

```
Private Sub Form_Load()  
    Dim i As Integer  
    Me.LinkMode = 0;不允许 DDE 通信  
    ...; 其他设置  
    If m_lFrameDataLen > 2 Then ;m_lFrameDataLen 存储指定的帧长  
        For i = 1 To m_lFrameDataLen \ 2 - 1;每条指令占 2 个字节  
            Load lblReceive(i)  
            Load lblTrans(i)  
        Next i  
    End If  
    Me.LinkMode = 1;允许 DDE 通信  
Exit Sub
```

4 定时器的改进

在工业控制的实际应用中,常有定时发送数据的要求。VB 为开发人员提供了一个 Timer 控件,可方便地实现定时功能。但由于其实现方法的原因(通过 WM_TIMER 消息实现定时响应),Timer 控件只能精确到 55 ms。要想得到更高精度的定时功能,只有依赖于 Windows API 了。

在 Windows API 中,有一组多媒体定时器函数(Multimedia Timer Functions),该组函数属于 WINMM.DLL,可以提供最高 1 ms 的定时精度。表 2 列出其中一些函数的名称和主要功能。

多媒体定时器是通过回调函数(Callback Function)精确定时的。所谓回调函数,就是指应用程序中被 Windows 系统所调用的函数。在大多数调

表 2 主要多媒体定时器函数及功能概述

Tab. 2 The main functions for multimedia timer

函数名	函数功能
timeGetDevCaps	得到当前设备所能支持的分辨率
timeBeginPeriod	设定定时器的最高分辨率,必须在即将使用定时器服务前调用,和 timeEndPeriod 配对使用。
timeEndPeriod	清除先前设定的最高分辨率。
timeSetEvent	启动定时器,该定时器在自己的线程中运行,当到达指定时间后,系统调用指定的回调函数(即 TimerProc)执行操作
timeKillEvent	中止指定的定时器
TimeProc	被系统调用的回调函数

用 WinAPI 的 VB 应用程序中,应用程序中的函数通过主动调用位于 DLL 中的 API 函数而实现某种功能;而要调用回调函数的 API 函数,应用程序中应该有一个调用该 API 的函数和一个被该 API 调用的回调函数,程序运行时,普通函数先调用 API 函数,用 API 函数再调用应用程序提供的回调函数,最终实现精确定时功能。总之,回调函数不是由应用程序直接调用,而是由 API 函数调用的。图 2 显示了该调用过程。

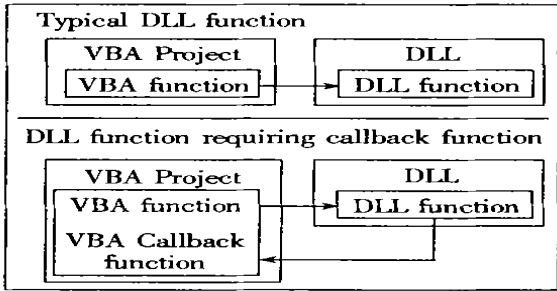


图 2 回调函数的调用示意图
Fig. 2 The call of a callback function

在使用多媒体定时器时,timeSetEvent 函数就是调用回调函数的 API 函数,程序中用如下方法调用该 API 函数,并启动多媒体定时器。

```
g_TimerID=timeSetEvent(g_lDelay,g_lResolution,AddressOf TimeProc,0,0)
```

根据语句,该定时器以 g_lResolution 毫秒为精度,在 g_lDelay 毫秒后调用 TimeProc 函数。AddressOf 是 VB 中取函数地址的关键字;参数中的第 1 个 0 指传递给回调函数的参数,实际可以是任意有用的值;第 2 个表示仅触发一次,如果改为 1,则表示反复触发。

回调函数 TimeProc 的参数列表有严格要求,下面给出了 TimeProc 的写法和多媒体定时器调用的方法。需注意的是,尽管此处 2 段程序写在了一起,但由于 VB 的特殊性,AddressOf 关键字只能应用于通用模块中的函数,所以实际的程序中,TimeProc 必须在通用模块中使用。

```
Public Sub TimeProc(ByVal lIDTimer As Long,ByVal ReservedMsg As Long,ByVal lParamUser As long,ByVal Rese-
```

```

rvedParam1 As Long , _
    ByVal ReservedParam2 As Long)
...
frmMain.MSComm 1.Output = g _ bytTrans
g _ TimerID = timeSetEvent (g _ lngDelay ,
g _ lngTimePeriod ,AddressOf TimeProc ,0 ,0) ;再次触发
...
End Sub

Private Function TimerStart() As Boolean
Dim TimeResolution As TIMECAPS
If timeGetDevCaps ( TimeResolution , 8 ) < >
TIMERR _ NOERROR Then
    MsgBox ;定时器错误 ,数据将无法发送
    Timer Start = False
    Exit Function
End If
If TimeResolution.wPeriodMin < TARGET _ RESO-
LUTION Then
    g _ lngTimePeriod = TARGET _ RESOLUTION
Else
    g _ lngTimePeriod = TimeResolution.wPeriodMin
End If
timeBeginPeriod (g _ lngTimePeriod)
g _ TimerID = timeSetEvent (g _ lngDelay ,g _ lng-
TimePeriod ,AddressOf TimeProc ,0 ,0)
If g _ TimerID = 0 or Err.Number Then
    MsgBox ;定时器启动错误 ,数据将无法发
送 ,vbOKOnly
    timeEndPeriod (g _ lngTimePeriod)
    Timer Start = False
    Exit Function
Else
    TimerStart = True
End If
End Function

```

5 程序应用

以 InTouch 为例说明该通信接口程序的应用方法。

在 InTouch 里先建立“DDE 访问名”,如 Serial。将“DDE 应用程序/服务程序名称”填入 SerialComm,“DDE 主题名”填入 DataPool。然后,对需要读取串口数据的标记名的“DDE 访问名”设为 Serial,“项”设为 lblReceive(i);对需要发送串口数据的标记名,仍然设“DDE 访问名”为 Serial,但把“项”设为 lblTrans(i)。其中,i 是相应控件数组元素的标号。

完成上述设置后,对 InTouch 而言,这 2 个标记名就相当于串口的输入输出缓冲,收发串口数据完全转化为对 2 个标记名的读写操作。

该通信接口程序已经在 InTouch 和一称量装置的通信中得到了应用,完全符合现场工作的要求。

参考文献:

- [1] 刘瑜. Windows 环境下串行通信程序设计[J]. 计算机应用研究,1999,(11):78-80.
LIU Yu. Design serial communication program in Windows[J]. **Computer Application Research**,1999,(11):78-80.
- [2] 程铁皋,程文斌. Windows 动态数据交换程序设计[M]. 北京:北京航空航天大学出版社,1995.
CHENG Tie-gao, CHENG Wen-bin. Design dynamic data exchange program in Windows[M]. Beijing:Beijing University of Aeronautics and Astronautics Press,1995.

(责任编辑:汪仪珍)

作者简介:

阚江(1976-),男,江苏张家港人,硕士研究生,主要研究领域为自动控制在电力系统中的应用(E-mail:kanjiang@21cn.com);

孙苓生(1948-),男,上海人,高级工程师,主要研究领域为自动控制系统应用;

张明(1966-),男,上海人,高级工程师,主要研究领域为自动控制系统应用。

Serial communication between industrial control software and slave system with VB 6.0

KAN Jiang ,SUN Ling-sheng ,ZHANG Ming

(Shanghai Jiaotong University, Shanghai 200030, China)

Abstract : Regarding to that the serial communication driver provided with industrial control software does not support current devices in automation control ,a DDE-based serial communication software with VB 6.0 is developed ,which can be applied by most industrial control software. For the slave system ,the MSComm Controller is used to receive and transmit serial data ,for the industrial control software ,the data is exchanged transparently via DDE(Dynamic Data Exchange) . The engineering practice proves that it is easy in implementation ,applicable in functions and reliable in operation. In addition ,the data are normally transmitted by time schedule in industrial control and the timer control provided by VB can not be less than 55 ms ,WinAPI is hence adopted to set a timer with 1 ms resolution.

Key words : industrial control software ; serial communication ; DDE ; timer