

# 利用 VB 开发无线通讯程序

刘建武 深圳凯斯泰尔通讯设备有限公司(518033)

## Abstract

In this paper, an application instance is given to introduce how to design radio communication program by using Visual Basic in a project

KeyWords: radio communication, VB, industrial automation

## 摘要

本文以工程上的应用为例,介绍了利用 VB 开发无线通讯程序的方法。

关键词:无线通讯,VB,工业自动化

无线通信广泛应用于我国国民经济的各个部门,在工业自动化控制领域中(特别是在炼油、化工、自来水生产)的应用越来越广泛,以下程序是作者在工厂的自动控制系统中,根据实际应用需要,开发出的无线通讯系统,作为整个控制系统的一个重要构成,与工厂的全厂自动化组成一个有机的整体,在已投入生产运行的半年中,一直运行稳定,用户反应良好。

## 1 串行通信

用 Visual Basic 开发串行通讯程序有两种方法:一是利用 WINDOWS SDK 的 API 函数,另一种是利用 VISUAL BASIC 自带的控件 MSCOMM。

### 1.1 与串行通讯系统有关的 WINDOWS API 函数

WINDOWS API 提供了三个核心动态链接库文件:Kernel32、User32 和 GDI32 供开发人员使用。Kernel32.dll 包括一些底层操作函数,User32.dll 包含了一些与 WINDOWS 有关的函数,GDI.dll 为图形设备接口库,和串口通讯有关的函数等均在 \WINDOWS\SYSTEM\子目录下的 User32.dll 动态链接库中。

利用 API 编写串口通信程序较为复杂,需要掌握的知识较多,但利用 API 可编写更为复杂的通讯程序,而且编程人员可以更透彻地了解通讯的底层功能。采用这种方法需要用到以下函数:

(1) Create File 函数打开指定的端口。

(2) CommConfigDialog 函数可以对话框的形式设置:波特率、数据位、奇偶校验和流控方式等。

(3) CommTimeOuts 结构定义超时设置,可用 GetCommTimeOuts 函数获得当前的设置,用 SetCommTimeOuts 函数完成设置。

(4) ReadFile、WriteFile 读写串行端口,与读写文件采用的方法相同。

(5) CloseHandle ( ) 可关闭串口,以释放独占资源。

关于 API 的资料,市场上有专门的书籍介绍 WINDOWS API 函数,这里就不再赘述。以上是 WINDOWS 95 中的串口通讯 API 函数,不同于 WINDOWS31 中的 OpenComm、CloseComm、ReadComm、WriteComm 等。

### 2.2 利用 VB 通讯控件 MSCOMM 开发无线通信程序

MSCOMM 控件提供了功能完善的串口数据发送及接收,利用 RS-232 进行串行通信所需的详细设置、规则都可以通过设置 MSCOMM 控件的各种属性而实现。利用 MSCOMM 的事件、函数可以完成数据的接收和发送。总之,通过 MSCOMM 控件进行通讯程序设计较之调用 WINDOWS SDK 的 API 函数进行设计,要简单、快捷,且只要较少的代码可实现相同的功能。

MSCOMM 控件对串行通讯有两种处理方式,一种是利用事件驱动方式:利用 MSCOMM 控件的 ONCOMM 事件捕获并处理通信错误及事件。另一种是使用查询方式,通过查询 Comm Event 属性的值来判断事件和错误。

如果在通讯过程中发生了一个通讯事件或错误将产生 ONCOMM 事件,通过查询 ONCOMM 事件的属性 CommEvent 可以确定错误的类型以及所发生的通讯事件,在程序设计中、我们可以根据不同的值来进行不同的操作。

## 2 无线通讯的传输协议及通讯接口

在我们的工程中,上位机采用研华公司的 PC610 型工控机,配置为 Intel P II350、128M 内存,操作系统为 WINDOWS95 OSR 2 中文简体版,通讯程序采用 VB 开发,下位机采用德国西门子公司生产的 PLC 作为数据采集,其数据采集精度为 12 位,即约为 1 / 4000,下位机实时采集数据之后,通过简单转换计算,在上位机进行采集时,向上位机发送数据。

上位机和下位机采用问答式(Polling)数据传输协议,每一台下位机可接受三路模拟量输入,八路脉冲量输入,四路脉冲量输出。在我们的项目中,由于要测量三个不同位置的远端压力,每一台下位机只需要测量一个压力信号。因此我们总共使用了三台下位机。根据以后需要可多扩充至 256 台下位机,而一台下位机也可接收更多的输入信号。在此通讯系统中,采用的是一点对多点的广播式通信,而非点对点通信,所以在我们的通讯协议中增加了站点识别码,每个下位机都有自己的站号,可以很方便的在下位机设置站号,整个系统中,站号必须唯一才能保证各个通讯点准确无误地运行。

上位机向下位机发送命令为 ASCII 码,下位机向上位机回送的有 ASCII 和 BCD 码,数据测量部分采用 BCD 码,是为了节省数据位资源,以及更灵活地进行数据处理。上位机与下位机之间的通讯协议为:55 7E 81 0 N -N AA 其中 N 表示站号,-N 表示站反。

我们使用的无线 MODEM 是日本的日立 HITACHI、M338VC,工作频率可在 220 - 235MHZ 中设定,保证工作在干扰源较少的频带。为了可靠进行传输,在工作中设为 600BPS,无奇偶校验位,8 位数据位,1 位停止位可使工作处于很稳定的状态。

## 3 应用实例

以下为所开发的无线通讯程序,并应用在实践中。

该程序中,本人是利用 MSCOMM 控件而非 WINDOWS SDK 的 API 函数进行程序设计;在信号处理中,则采用事件驱动方式,即通过捕获 MSCOMM 控件中的 ONCOMM 事件来处理通信错

误及事件。要采用 MSCOMM 控件实行串口通信,首先需添加一 MSCOMM 控件到窗体中,该控件一般不在通用工具窗口中,可右键单击工具窗,选中 COMPONENTS...,在 Control 表中,选中 Microsoft Comm Control 6.0,此时,在工具窗中会出现 MSCOMM 图标,便可使用。

由于该项目的软件源代码较长,在此拿出和串口通讯有关的程序片断供大家参考。

```
Sub Form_Load ( ) '初始化串口
    .....
    Const NONE = 0, SERVER = 1
    IF LinkMode = NONE then
        LinkMode = SEVER
    END IF
    LinkTopic = "Radio" '设制此程序为服务器模式,为其他
    用户程序提供数据,便于系统通讯及系统集成,连接的主题为
    Radio.
    comm1.CommPort = 2 '使用通讯端口 2
    comm1.Settings = " 600, n, 8, 1" '设定通讯波特率为
    600bps,无校验,8 位数据位,1 位停止位。
    comm1.SThreshold = 0 '发送数据时禁止产生 On-
    Comm 事件
    comm1.InBufferSize = 1024 '定义输入缓冲区为 1024
    字节
    comm1.InBufferCount = 0 '清零输入缓冲区
    comm1.OutBufferCount = 0 '清空输出缓冲区
    comm1.InputLen = 0 '读整个缓冲区
    comm1.PortOpen = True '打开端口
    zh = zf = 0 '初始化站号、站反为 0
    timer1.Interval = 3000 '设定定时器时间间隔为 3S
    .....
End Sub
```

下面是由上位计算机往下位机发送数据,传送命令的子过程,该子过程由一个定时器 Timer1 控制,每隔 3s 执行一次,其时间间隔可在 Form\_Load ()中通过设定 Timer1.Interval 的值进行控制。

```
Sub Timer1_Timer ()
    Dim duration, tm As Variant
    Dim out As String
    zh = zh + 1
    If (zh > 3)Then '设定站号,因为系统共有 3 个下
        zh = 1 '机,其站号分别为 1、2、3,因
    End If '此当站号大于 3 时又循环成 1 号站
    zf = (55 - zh) '站号与站反互为补
    ex = 0
    in = ""
```

```

comm1.RTSEnable = True      '将 MODEM 的 PTT 置为高
电平
If comm1.OutBufferCount <> 0 Then '清零发送缓冲区
    comm1.OutBufferCount = 0
End If
out = ""
out = Chr (5) & Chr (26) & Chr (29) & Chr (0) & Chr
(4) & Chr (f) & Chr (70)
comm1.Output = out          '上位机向下位机发送协议
码
Do                          '该循环使协议码全部发送完毕
Loop Until comm1.OutBufferCount = 0
duration = Timer + .15
Do
Loop Until Timer > duration '适当延时
comm1.RTSEnable = False     '将 MODEM 的 PTT 置
为低,使其为接受状态。
comm1.InBufferCount = 0     '清零接收缓冲区
comm1.RThreshold = 1        '当接收缓冲区中有 1 个字
符,触发 ONCOMM 事件
Select Case zh              '置下发标志
    Case 1: f1.Text = -1
    Case 2: f2.Text = -1
    Case 3: f3.Text = -1
End Select
End Sub

    以下为 MSCOMM 的 ONCOMM 事件处理子程
    序,在此子程序中是接收和处理代码的语句,其接收
    的代码为:0 FF FF FF FF FF FF 0 7E 81 0 N -N X X
    X X X X X X X X X X X X X X X X,其中前面的均为十
    六进制数,X 是用 BCD 码表示的采集过来的数据,可
    以根据需要,在此子程序中添加其它内容:
Sub Comm1_OnComm ()
    .....
    Select Case comm1.CommEvent
    .....
    Case comEvSend ... '发送缓冲区中的字符数
    少于
    'SThreshold 时,插入相应代码
    Case comEvEOF ... '收到文件结束符(ASCII 字
    符 26)
    '时,插入相应代码
    Case comEventFrame ... '硬件检测到帧错误,插
    入相
    '应代码
    Case comEventRxover '接收缓冲区满,清零
    comm1.InBufferCount = 0 '接收缓冲区
    Case comEventTxFull '发送缓冲区满,清零
    comm1.OutBufferCount = 0 '送缓冲区
    Case comEventReceive

```

```

comm1.RThreshold = 0        '设置其值为 0,
则接收到数据以后,也不产生 OnComm 事件
If Len (n) <40 Then        '取回整个代码
    cominput = comm1.Input
    in = in & cominput
Else
    For i = 1 To 16        '将前 16 个数取
    to
        sinn ( ) = Asc (Mid (n, i, 1)) '数组 sinn
        ()中
    Next i
    For i = 1 To 16        '打印源代码
        picture3.Print Format$(sinn ( ), "000" );
        " ";
        cx = cx + 1
        If cx > 8 Then
            cx = 0
            picture3.Print
        End If
    Next i
    '以下为包头校验
    If (sinn (1) = 0 And sinn (2) = 255 And sinn (3)
    = 255 And sinn (4) =
    255 And sinn (5) = 255 And sinn (6) = 255
    And sinn (7) = 255
        And sinn (8) = 0 And sinn (9) = 126 And
    sinn (10) = 129 And
        sinn (11) = 0 And sinn (12) = zh And sinn
    (13) = zf)Then
        combin = ((255 * sinn (4) + sinn (5) -
    200) / (255 * 4 - 200))
        If combin <0 Then
            combin = 0
        End If
        '针对不同站号,其数据取到不同的文本框中,
        并置对
        '应点接收到数据信号标志
        Select Case zh
            Case 1
                f1.Text = 1
                text1.Text = Format$(combin, "
    00.000" )
            Case 2
                f2.Text = 1
                text2.Text = Format$(combin, "
    00.000" )
            Case 3
                f3.Text = 1
                text3.Text = Format$(combin, "
    00.000" )
        End Select
    Else                    '若校验码不对,对该站重发三次
        zh = zh - 1
        cs = cs + 1
        If cs > 3 Then
            cs = 0
            (下转第 20 页)

```

## 2) 提高软件鲁棒性

软件鲁棒性是指软件在遇到违反规范的情况下,程序仍能继续执行的性能。采用以下方法可提高软件质量。

### ① 防止浮点错误

在发生浮点错误时程序将出错而不能继续执行,这一般是在除法运算中分母为零引起的。作者建议程序中遇有除法运算的地方,若分母为变量则先判断其是否为零,若为零则根据具体情况赋给变量一个合理值。比如在数据处理过程中都会用到的标度变换程序将采集的模拟量转化为物理量时,需除以 K 系数,当存储 K 系数的文件变化使其为零时,程序自动赋理论值给 K。

### ② 防止死循环

有时由于各种原因会导致死循环并最终导致死机。在遇有循环时加上判断语句,一旦循环次数超过一定数目则自动跳出循环。这样就可避免死循环的发生。

### ③ 输出限位

在计算机控制系统中软件的一些输出直接作用于系统,其与设备的接口就显得尤其重要。因此应在对设备进行了解的基础上,对各输出口进行了软件限幅以免超出设备的输出极限。另外,在程序执行结束后应自动将各输出口置成零以避免非运行状态系统的输出口带电。

### ④ 意外情况数据挽救

若每次的采集数据结果都非常重要,则为防止在数据未及正常保存的情况下出现意外情况,程序中应增加数据挽救功能,使得在采集过程结束后仍能找到有用数据。这可以通过在内存中开一区域,边采集边将数据存放于一固定的内存区域,若未正常保存数据,则从该段内存中读出数据。

## 3.2 提高软件抗干扰能力

在计算机控制系统中,采集的数据是否可靠非常重要。不可靠的数据轻则使系统控制精度降低,重则导致系统控制失灵。除了硬件抗干扰方法,数字滤波是提高采集数据可靠性的最有效的措施,在作者的其它文章中已对此有详细论述,这里不再赘述。另外,在比较特殊的场所可定时监测一些特征点,若超过其极限值,则自动停止工作。

## 3.3 提高故障诊断能力

通常,一个故障从存在到排除,大体经历以下过程:1)发现故障,识别故障;2)确诊故障的原因和部位;3)采取措施,消除故障。首先是故障检测即如何发现故障的问题。一般情况下,操作失误及程序可能遇到的执行错误都应给出错误提示,这样,这一类的故障根据提示内容很快就可以得到解决。若程序执行过程中未出现错误提示,但结果不对,则首先判断是软件原因还是硬件原因,若属软件原因则启用备份软件或其它版本软件;若属硬件故障则另行处理。所以,提高故障诊断能力和软件的容错性是息息相关的。

## 4 结束语

本文主要介绍了计算机控制系统中一些提高软件可靠性的方法,作者在某大型技术改造项目中使用这些方法,该项目可靠运行一年多的实践表明这些方法是行之有效的。当然,可靠性是一门年轻的科学,它的发展是永无止境的。不同的系统可根据具体情况进行分析、比较、试验,找到最适合的方法。

## 参考文献

1. 于春梅 航空机轮实验台采集机设计与鼓轮速度控制研究 西北工业大学硕士论文 2000.4
2. 潘新民,王燕芳,微型计算机控制技术,人民邮电出版社,1999.6

(上接第9页)

```

zh = zh + 1
End If
End If
Exit Sub
End If
comm1.RThreshold = 1
End Select

```

End Sub

## 4 结论

本文所开发的无线通讯程序,经过长时间的运行,非常稳定与其他的通信程序一起构成完整的通信

子系统。通讯子系统与其他各种检测、控制子系统组成一个完善的工厂自动化控制系统。

在我们此项目的实践工作中,觉得编写无线通讯程序与有线通讯程序的最大的区别是用 MSCOMM 控件的 RTSEnable 属性对无线 MODEM 的 PTT 电平的操作。当然,在发送和接收代码中,适当地延时是非常重要的,这是所有初学编写通讯程序的朋友经常忽略的一点,在这方面,本人曾经走过不少弯路,所以在此特别提醒注意。