

# 串行通信中的字节与字符

何香玲 张 跃 郑 钢 清华大学自动化系(100084)

## Abstract

The usage method of the InputMode properties for MSComm control is discussed in Visual Basic serial communication. Under the Binary Mode the data transmitted can resolve the existent problems while transmit over 128's character in the ASCII, and eliminate the limitation that String variable can only deal with 0~127's text character of the ASCII under the Text Mode. It implements in deed the communication between computer and singlechip in the Windows. At the same time, the application example that the byte is sent and received is given out in the serial communication. The practice proves that it has applied value in a certain extent.

**Keywords:** MSComm control, InputMode properties, byte and character, UniCode code

## 摘 要

本文讨论了 VB 串行通信中 MSComm 控件 InputMode 属性的使用方法。在 Binary 模式下传输数据, 可以解决 ASCII 码中传送超过 128 的字符时所存在的问题, 消除了 Text 模式下, String 变量只能处理 ASCII 码 0~127 的文本字符的缺陷, 真正实现 Windows 环境下计算机与单片机的通信。同时给出了串行通信中字节的发送与接收的应用实例, 具有一定的实用价值。

**关键字:** MSComm 控件, InputMode 属性, 字符与字节, UniCode 码

在用 VB6.0 的 MSComm 控件进行通信时, 串行传输都是使用字符的传输。而一般的串行控制也是以字符为主要的传输方式, 即对 InputMode 属性没有作特别的指定, 而此属性的默认值是 Text 模式, 也就是接收的方式将输入的数据当作文字予以解读。但在某些情况下, 传送的不一定是纯文字型态的数据, 而是数据或文件在传送, 在这种情况下, 位在传输线上的数据就是一个字节接着一个字节地被传送过来, 接收端收到这些数据后, 再予以组合形成数据资料或文件, 这种情形就是二进制数据(Binary)的传送。

## 1 问题的提出

在编程实践中, 发现采用 VB 的 MSComm 控件进行 PC 机与单片机的串行通信时, String 型变量处理通信数据存在着一个缺陷。因为字符型变量处理通信数据, 只适合处理 ASCII 码为 00H~7FH 的文本字符, 不能处理单片机发来的 80H~0FFH 字节。在这种情况下, 单片机如果要存储单元中的一个字节 0ABH 发送给计算机, 就只好拆成两个字符“A”和“B”, 作为 41H 和 42H 发送, 这显然是不对的。因此, 用 MSComm 控件的 Text 模式不能进行 PC 机与单

片机的串行通信。

## 2 文本与二进制数据的选择

使用 VB 的 MSComm 控件进行编程时, 需要注意 InputMode 属性, 因为它确定了 Input 属性取回数据的方式。当 InputMode=0 时, 数据以文本方式(Text Mode)取回, 即从数据接收缓冲区取回的是字符串。若数据只用 ANSI 字符集, 例如文本字符, 则可用这种方式。当 InputMode=1 时, 数据以二进制方式(Binary Mode)取回, 即取回的是二进制数据。当数据中包含非文本字符, 例如数据中嵌入控制字符、Null 等, 或者包含 ASCII 值大于 128(80H)的字节时, 则必须使用这种方式, 否则就无法正确处理通信数据。

在部分设备的计算机联机功能里, 比如单片机与 PC 机的通信中, 就可能会出现传送超过 ASCII 码 128 以上的情形, 若采用 VB 预设的文本模式将无法得到默认的结果, 必须采用二进制模式才可。

若系统建立之初不知道所传送的是字符还是字节的数据, 那么比较好的方式是采用字节的数据接收方式, 这样便不会失去任何一个数据, 而接收回来的

数据再作必要的处理与转换即可。

### 3 串行通信中字符与字节的概念

大家知道,西文字符一般是以单7位一个字节来表示的,最多有128种组合方式,最常用的表示方法是ASCII码。对应ASCII码表得到相对应的字符为可见字符。如果使用8个位形成一个字节,那么可能的组合就是256种,也就是ASCII码表上的256个字符,其中包括128~255的不可见字符。这些不可见字符与可见字符使得串行通信产生一些特别的现象。

原来的计算机里只有英文,英文只需用到ASCII码的前128个位置就足以表达全部的字符;在ASCII制订的时候,并没有考虑对多语种特别是对中国汉字这样的象形文字的支持。为此,后来又提出了不少解决方案,而Unicode是今后发展的方向。几家美国的厂商联手制定了Unicode,这个Unicode涵盖了世界上所有国家的字符码,每个文字用一个唯一的内码来表示,但是这种Unicode的特点是所有的字符是以两个字节表示,不仅中文使用两个字节,而且英文也使用两个字节表示。为了与ANSI编码兼容,西文字符的高字节为0,低字节就是以前ANSI编码,如ASCII码。以单字节表示西文字符,以双字节或多个字节来表示汉字等,汉字的处理方法与西文字符完全相同,需要以第一字符的高8位来判断是中文字符还是西文字符,这也是大家一般理解的汉字表示方法。

如:“A”这个字母会被表示成4100(低字节是41,高字节是00);“何”这个字符表示为BACE(低字节是BA,高字节是CE),可见汉字的第一个字节的高8位为1。

#### 3.1 ANSI、DBCS和Unicode的定义

VB使用Unicode存储和操作字符串。Unicode是一种用两个字节表示一个字符的字符集。另外一些程序,如Windows 95 API,使用ANSI或DBCS存储和操作字符串。当从VB移出字符串时,会遇到Unicode和ANSI/DBCS之间的差别。

1) ANSI (American National Standards Institute) 是个人计算机使用得最普遍的字符集。由于ANSI标准使用单一字节表示每个字符,因此最多只能有256个字符和标点符号代码。虽然对英语来说已经足够了,但不能完全支持其它语言。

2) DBCS (Double Byte Character Set) 是发行在亚洲大部分地区的Microsoft Windows系统使

用的字符集。它支持很多不同的东亚语言字母,如汉语、日语和朝鲜语。DBCS使用数字0~128表示ASCII字符集。其它大于128的数字作为前导字节字符,它并不是真正的字符,只是简单的表明下一个字符属于非拉丁字符集。在DBCS中,ASCII字符的长度是一个字节,而日语、朝鲜语和其它东亚字符的长度是2个字节。

3) Unicode是用两个字节表示每个字符的字符编码方案。国际标准组织(ISO)几乎为每种语言的每个字符和符号在0到65535( $2^{16}-1$ )范围内定义了一个数字(再加上为将来发展保留的一些空余空间)。在所有32位版本的Windows中,部件对象模型(COM)都使用Unicode,它是OLE和ActiveX技术的基础。Windows NT全部支持Unicode。虽然Unicode和DBCS都是双字节字符,但它们的编码方案完全不同。

#### 3.2 VB中字符的测试

如果打开VB的立即窗口,分别求A的ASCII码和65号ASCII码所对应的字符,结果会出现字符A的ASCII码编号(十进制位是65,十六进制位是41H)。当输入其它的英文字符或键盘上的可见符号,查看其ASCII码的编号,都会落于128以内。同样地,在立即窗口里也会出现65的ASCII码所对应的字符是字母“A”。

总之,不管是由字符求出ASCII的对应数,还是从数字中找出相对应的字符,在ASCII码128以内都不会有什么问题。如果超出ASCII码的128字符,就会出现错误。

在立即窗口键入命令字符串“? Chr(164)”,并按Enter键,结果发现什么也没有。如果想看看原来的编号值,键入命令字符串“? Asc(Chr(164))”,并按Enter键,这其实就是反相的作用,理论上应该看到的结果是164,但是在立即窗口里看到的却是“0”。这就是说,当传送超过128的字节时,VB就出现了问题,而且这个问题就出现在中文的组成字节上。上面的试验结果如图1所示。

如果在立即窗口上输入字符串“? Hex(Asc(“何”))”,并按下Enter键,结果出现“BACE”,同样地,双字节“BACE”所表示的字符是“何”字。在这里应该注意到,第一个字节的数字是超过128的。

对于VB的串行传输来说,当碰到超过128的字符时,它认为是非英文字符,必须与下一个到达的位组合成一个文字才是正常的字符,若无下一个字节的

出现与其合并,便将此超过 128 的字符以文字模式传送,而且传送动作将会失败,因为 VB 无法处理此种情形。试验结果如图 2 所示。

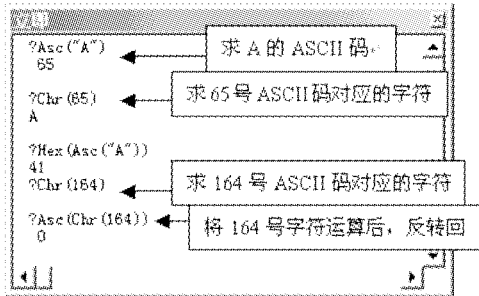


图 1 字符的测试

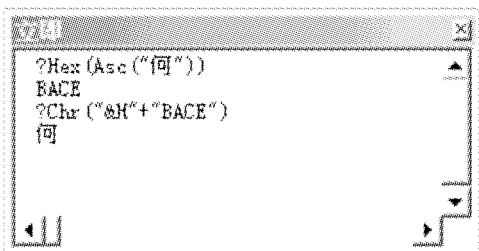


图 2 字符的测试

### 3.3 VB 中字符数的计算

VB 是否以两个字节来处理字符呢? 在 VB 的立即窗口里输入字符串“?Len(“XYZ 何香玲”)”,结果会出现“6”,代表的是共有 6 个字符,不仅英文算一个字符,而且中文也算一个字符。当在 VB 的立即窗口里输入字符串“LenB(“XYZ 何香玲”)”时,结果会出现“12”,LenB 函数的意义是传回该字符串的字节数,因此传回 12 表示字符串中含有 12 个字节。

由此可见,VB 在处理字符串时,的确是以双字节的方式在处理,而且计算字符总数时也是以两个字节作一个单位。

如果我们要求得的字符总数是英文以一个字节代表,中文以两个字节代表,那么必须使用 StrConv 函数。在 VB 的立即窗口里输入以下的表达式: ? LenB(StrConv(“XYZ 何香玲”,vbFromUnicode))所得的答案是 9,也就是上面的字符串中是 9 个字节,参数 vbFromUnicode 表示欲转换的字符串是来自 Unicode。试验结果如图 3 所示。

### 4 数据在硬件线路上的传输

由于 VB 在字符串的处理上使用的是以两个字节代表一个字符,但是由 RS-232 传送出去的时候是否也是传送两个字节呢? 其实不尽然,有的时候是一个字节,有的时候是两个字节。如果传送的是英文,VB 会自动将原来补在英文中的空白字节那掉,而只

传送真正的字节,也就是只传送一个字节;而中文字则还是传送两个字节。所以,程序中的字符与被传送的字符不一定均使用两个字节作为传递单位。如果是英文,被收到的字符串会原样地呈现;而若是中文字,则当系统接收时,会等到完整的字节到达时才会组合成双位字符,也就是说,VB 会判断第一个字节的值是否大于 128。如果 RTHreshold=1,当一个中文汉字或一个英文字母从串口送出时,均会引发接收事件;如果 RTHreshold=2,那么英文字母必须传送超过两个字符,才会引发接收事件,但中文字只要一个字就可以引发接收事件。试验结果如图 4 所示。

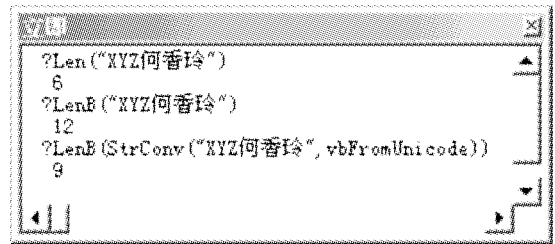
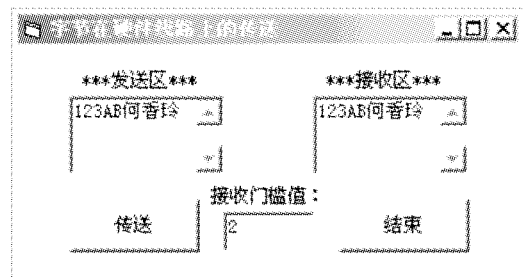
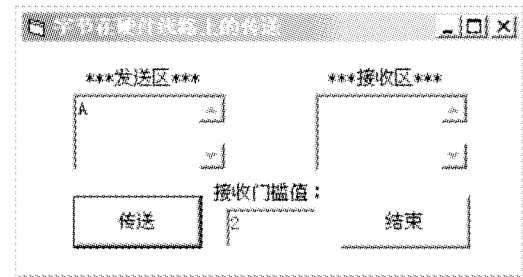


图 3 字符数的计算



(a)



(b)

图 4 数据在硬件线路上的传输

### 5 字节传送与接收的应用实例

字符与字节的关系,前面已经作了讨论。在 MSComm 控件的接收方式里面,除了标准的字符接收方式外,还有一个字节的接收方式。而 MSComm 控件的默认声明是 Text 而不是 Binary,使用 Text 模式将无法接收与发送 ASCII 码 128 以上的单一字符,因为 VB 会把 128 以上的字节认为是双位字符的一部分,且等待下一个字节的到达,VB 继而将此两

个位组合成一个文字。在文字的传送过程中，会有128以上的ASCII码被传送。所以，使用二进制Binary方式就可以解决ASCII码128以上的问题。

功能设计：当按下“传送”按钮后，送出在传送区键入的文字；接着按下“接收”按钮，就把对方送过来的数据以字节的方式显示在接收区文本框内。

### 5.1 数据的发送

在PC机与单片机或两台PC机之间通信时，先声明两个byte型的动态数组，分别用于存储传送和接收的数据，在送出数据之前，先对数据作些处理，由于要传送的是字节数据，因此，以文字作为输入方式在传送之前，先使用StrConv函数，把双字节字符和单字节字符作一个转换，转换完后再将数据送出去。

发送程序代码如下：

```

Dim OutByte()As Byte,InByte()As Byte
Private sub CmdSend_Click()
Dim Buf$
Buf=Trim(txtSend.Text)
OutByte=StrConv(Buf,vbFromUnicode)
MSComm1.Output= OutByte
End sub

```

### 5.2 数据的接收

将MSComm1控件的InputLen属性设置为0，只要在输入缓冲区内有数据，MSComm1控件的Input属性便将数据读入InByte变量中。然后从字节变量InByte中撷取真正的二进位数据，显示在接收区文本框内。

接收程序代码为：

```

Private sub CmdReceive_Click()
Dim Buf$,i%
InByte=MSComm1.Input
For i=LBound(InByte())To UBound(InByte())
Buf=Buf+Hex(InByte(i))+Chr(32)
Next i
TxtReceive.Text=Buf
End sub

```

试验结果如图5所示

### 5.3 区位码复原汉字

如果用区位码复原汉字，程序代码如下：

```

Private Sub CmdSend_click()
Dim MyString as String
Dim PutCode As String
.....
PutCode = txtSend.Text
MyString = Chr("&H" + PutCode)

```

```

MSComm1.Output = MyString
.....
End Sub

```

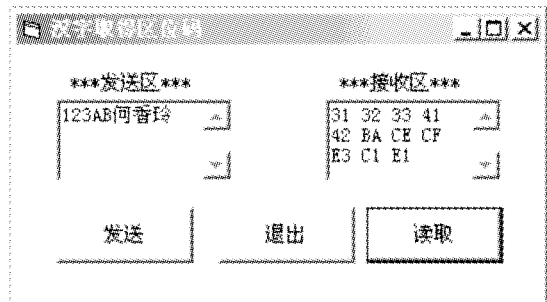


图5 字符取得区位码

试验结果如图6所示：

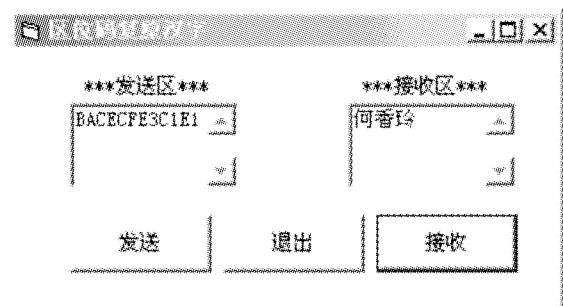


图6 区位码复原成汉字

此程序的试验执行在一台计算机的环境中进行，使用同一个通信串口，将RS232的第2脚与第3脚短路，便可以自发自收。

## 6 结束语

在日常使用中，我们实际上熟悉的是字码（内码）。在中文WIN9X下，我们输入一个双八位字节，就得到一个汉字，如果认为这双八位字节就是对应这样的字形。这是错误的。其实对于字库来说，内码是真正的内存中对中文的表示方法，是查找字形的索引。如果换另一个编码标准的字体，同一个字符串就会呈现不同的字形，也就是乱码。

### 参考文献

- 1 范逸之编著. Visual Basic与RS232串行通讯控制[M].北京:中国青年出版社,2000
- 2 陈海云.巧用VB的Byte型变量实现与单片机通讯.微计算机信息,2000(1)
- 3 王文峰等.利用VB Binary方式开发通用串行通讯软件.计算机应用,1999(12)

[收稿日期:2001.11.29]