

# 如何在串口通讯程序中处理数据包

## 文档信息

### 公司名称

北京春笛网络信息技术服务有限公司

### 电子邮件

shenzy@mailier.com.cn

### 电话

010-82355864,82358387,82356956,82356576,82356577

## 常规信息

### 说明：

在串口通讯程序中，经常要收到数据包，常有网友问及如何从这些数据包中提取需要的数据，如何处理校验等，在这篇文章里我举两个例子予以说明，程序说明为 VC++6.0。关于串口编程建立程序的细节，请参阅我主页上的其它文章。同时，此文也适于其它通讯程序中艰数据报文的处理。

### 正文：

首先，应该指出的是，所有这些处理均在串口事件处理函数 oncommunication() 中进行。每当串口缓冲区中有一个或一个以上字符时触发串口通讯事件，该事件就驱动（调用）串口事件通讯处理函数 oncommunication()，在这里就可以对接收到的数据进行处理，提取需要的数据。举两个例子，一个是较为简单的位数据格式的处理，另一个是 NMEA 无线通讯格式的处理，最后回答一位网友提出的问题，大家也可以探讨一下。

### 1、问题：

一个数据包，其串头为一个字符，字符值为 7EH（16 进制）'~'，其后紧跟一字符 'E'，然后是数据串，串尾也为字符值为 7EH 的一个字符：即 ~Exxxxxx...~ 如何处理这些数据？

我们仍以串口调试助手源程序及其详细编程过程之一 中的 OnComm()处理为例：

```
void CCommTestDlg::OnComm()
{
// TODO: Add your control notification handler code here
VARIANT variant_inp;
COleSafeArray safearray_inp;
LONG len,k;
BYTE rxdata[2048]; //设置 BYTE 数组 An 8-bit integer that is not signed.
CString strtemp;
if(m_ctrlComm.GetCommEvent()==2) //事件值为 2 表示接收缓冲区内有字符
{
//////////以下你可以根据自己的通信协议加入处理代码
variant_inp=m_ctrlComm.GetInput(); //读缓冲区
safearray_inp=variant_inp; //VARIANT 型变量转换为 ColeSafeArray 型变量
len=safearray_inp.GetOneDimSize(); //得到有效数据长度
for(k=0;k<len;k++)
safearray_inp.GetElement(&k,rxdata+k); //转换为 BYTE 型数组
for(k=0;k<len;k++) //将数组转换为 CString 型变量
{
BYTE bt=*(char*)(rxdata+k); //字符型
strtemp.Format("%c",bt); //将字符送入临时变量 strtemp 存放
m_strRXData+=strtemp; //加入接收编辑框对应字符串,在这儿,编辑框不是必须的,可做相应处理
char ch=(char)bt;
if(ch=='E')
{
//在此处设置一个可以接收数据的全局标志,说明接收到数据前的 'E' 标志了,下一步可以读数据了,同时将 m_strRXData 清空
flag=2;
m_strRXData.Empty(); //下一次接收的便为有用的数据
}
}
}
```

```

if(ch==0x7e)
{
flag=1; //下面可以提取数据了
}
if(flag==1) //标志为 1 ,
{
...//提取数据
flag=0; //提取完后，置标志为 0
}
}
}
//UpdateData(FALSE); //更新编辑框内容
}

```

## 2、NMEA 无线通讯格式的处理

### 2.1 NMEA-0183 报文格式

字符串 ( ASCII 字符 ) 格式如下 :

\$XXXX,XX,XX,XX,..... \* hh<CR><LF>

\$ : 串头

XXXX: 串头

XX : 数据字段 , 字母或数字

XX : 数据字段 , 字母或数字

XX : 数据字段 , 字母或数字

, : 逗号

.....

\* : 星号 , 串尾

hh : \$与 \* 之间所有字符代码的校验和 , ( 注意 : 校验和 h 为半 Byte 校验 , \*后第 1 个 h 表示高 4 位校验和 , 第 2 个 h 表示低 4 位校验和。得到校验值后 , 再转换成 ASCII 字符。 )

<CR> : 0DH,回车控制符

<LF> : 0AH,换行控制符

## 2.2 校验处理

由于数据是动态接收，所以数据的处理也是动态进行，尽管有时会收到几个字符才触发一个串口事件，但字符的接收是一个一个接收的，因此就可以在程序中先判断串头\$是否到达，若串头到达，就可以开始计算校验，直至串尾\*到达，这时\*号后面的两个字符就是校验码，收到这两个校验字符，就可以与自己计算的校验值比较，若不正确，就报错，并继续处理下面的数据，若正确，则处理接收的字符，提取需要的数据。

## 2.3 程序

```
CString m_strReceived;
```

```
CString m_strChecksum;
```

```
int flag;
```

```
char ch 为每次收到的字符
```

```
m_strReceived += (char)ch;
```

```
switch(ch)
```

```
{
```

```
case '$':
```

```
checksum=0; //开始计算 CheckSum
```

```
flag=0;
```

```
break;
```

```
case '*':
```

```
flag=2;
```

```
c2=checksum & 0x0f; c1=((checksum >> 4) & 0x0f);
```

```
if (c1 < 10) c1+= '0'; else c1 += 'A' - 10;
```

```
if (c2 < 10) c2+= '0'; else c2 += 'A' - 10;
```

```
break;
```

```
case CR:
break;
case LF:
m_strReceived[port-1].Empty();
break;
default:
if(flag>0)
{
m_strChecksum += ch;
if(flag==1)
{
strCheck=strCheck+c1+c2;
if(strCheck!=m_strChecksum)
{
m_strReceived.Empty();
}
else
{
strInstruction=m_strReceived[port-1].Left(6);
if(strInstruction=="$QGOKU") //如果串头正确
{
char *temp=(char*)((LPCTSTR)m_strReceived);//转换

int speed=(atoi(temp+7));// 提取 int 型数据
char splevel=*(temp+25); //提取 char 型数据

}

}
m_strChecksum.Empty();
}
```

```
flag--;  
}  
else  
checksum=checksum^ch;  
break;  
}
```

### 3、网友的问题

另外，我回答了一位网友的问题，大家也可以探讨一下：

**问题如下：**

我用你的串口程序收来的十六进制数据是这个样的：

00 10 10 C0 00 F0 F0 AB AC AD

我现在要将高四位取出来,也就是

011C0FFAAA(这点我不会,但我用 Left 实现了,可得到的是字符,不是我要的数值)

我只要 011C0FF.

我要把 011C0FF 进行如下的处理

011 转化成十进制

C 不变

0FF 也变成十进制

后显示,成 17 C 255

**答：**右移得到 011C0FF 后，可将其放在一个字符型变量 CString m\_strReceive 中：

然后将其转换：

```
char *temp=(char*)((LPCTSTR)m_strReceive;
```

```
char tbuf[6]; //temporary viable
```

```
tbuf[0]=temp[1]; tbuf[1]=temp[2]; tbuf[2]=temp[3]; tbuf[3]=0; //011 最后为 0 表示结束
```

```
int data1=atoi(tbuf);
```

```
char chdata2=temp[4]; //C
```

```
tbuf[0]=temp[5]; tbuf[1]=temp[6]; tbuf[2]=temp[7]; tbuf[3]=0;  
int data3=atoi(tbuf); //0FF
```

以上 data1,chdata2,data3 即为你要的数据