

Mscomm 概述

Microsoft Communications Control（以下简称 MSComm）是 Microsoft 公司提供的简化 Windows 下串行通信编程的 ActiveX 控件，它为应用程序提供了通过串行接口收发数据的简便方法。MSComm 控件在串口编程时非常方便，程序员不必去花时间去了解较为复杂的 API 函数，而且在 VC、VB、Delphi 等语言中均可使用。具体的来说，它提供了两种处理通信问题的方法：事件驱动(Event-driven)方法和查询法。

事件驱动通讯是处理串行端口交互作用的一种非常有效的方法。在许多情况下，在事件发生时需要得到通知，例如，在串口接收缓冲区中有字符，或者 Carrier Detect (CD) 或 Request To Send (RTS) 线上一个字符到达或一个变化发生时。在这些情况下，可以利用 MSComm 控件的 OnComm 事件捕获并处理这些通讯事件。OnComm 事件还可以检查和处理通讯错误。所有通讯事件和通讯错误的列表，参阅 CommEvent 属性。在编程过程中，就可以在 OnComm 事件处理函数中加入自己的处理代码。这种方法的优点是程序响应及时，可靠性高。每个 MSComm 控件对应着一个串行端口。如果应用程序需要访问多个串行端口，必须使用多个 MSComm 控件。

常用属性

MSComm 编程序必须了解的属性:

属性	功能
CommPort	设置并返回通讯端口号，缺省为 COM1
Settings	以字符串的形式设置并返回波特率、奇偶校验、数据位、停止位
PortOpen	设置并返回通讯端口的状态。也可以打开和关闭端口
Input	从接收缓冲区返回和删除字符
Output	向传输缓冲区写一个字符串
InputLen	设置每次 Input 读入的字符个数，缺省值为 0，表明读取接收缓冲区中的全部内容
InBufferCount	返回接收缓冲区中已接收到的字符数，将其置 0 可以清除接收缓冲区
InputMode	定义 Input 属性获取数据的方式（为 0：文本方式；为 1：二进制方式）
RThreshold	和 SThreshold 属性，表示在 OnComm 事件发生之前，接收缓冲区或发送缓冲区中可以接收的字符数

CommPort 属性设置或返回通讯端口号。在设计时，value 可以设置成从 1 到 16 的任何数（缺省值为 1）。但是如果用 PortOpen 属性打开一个并不存在的端口时，MSComm 控件会产生错误 68（设备无效）。必须在打开端口之前设置 CommPort 属性。

Settings 属性设置或返回串口波特率、奇偶校验、数据位、停止位参数。参数格式为 "BBBB,P,D,S" ;其中,BBBB 为波特率, P 为奇偶校验, D 为数据位数, S 为停止位数。value 的缺省值是: "9600,N,8,1"。其它校验字符为: 奇效验----'O';偶效验----'E';

InputMode 属性设置或返回通信方式。comInputModeText(0, 缺省值) 通过 Input 属性

以文本方式取回数据。comInputModeBinary (1) 通过 Input 属性以二进制方式检取回数据。

RThreshold 属性在 MSComm 控件发送数据前设置为要接收的字符数。当接收字符后，若 Rthreshold 属性设置为 0 (缺省值) 则不产生 OnComm 事件。若设置 Rthreshold 为 n，接收缓冲区收到 n 个字符都会使 MSComm 控件都将产生 OnComm 事件。

InputLen 属性设置或返回 Input 属性从接收缓冲区读取的字符数。Input 属性从接收缓冲区中读取的字符数。InputLen 属性的缺省值是 0。设置 InputLen 为 0 时，使用 Input 将使 MSComm 控件读取接收缓冲区中全部的内容。若接收缓冲区中 InputLen 字符无效，Input 属性返回一个零长度字符串 ("")。

InBufferCount 属性为缓冲区中已接收的字符数。该属性在从输出格式为定长数据的机器读取数据时非常有用。

使用方法

使用 MSComm 控件的一般使用方法为：

初始化串口：通讯端口号 (CommPort)、串口配置(Settings)。打开串口 (PortOpen)。设置通信方式 (InputMode = 1)。

发送请求数据：清空缓冲区 (InBufferCount = 0)、设置读取的长度 (InputLen = 0)、设置触发事件的接收长度 (Rthreshold = n)、设置发送内容 (Output)。

接收事件处理：取回串口中的字符长度(InBufferCount)、取回返回的数据 (Input)。

Modbus 协议介绍

通讯命令

Modbus 协议是一种通信标准,包含 RTU 和 ASCII。我们只需要了解 RTU 的读写命令及其打包过程。与我们编程相关的 Modbus 协议大致格式是：站地址 + 命令号 + 起始地址 + 操作数量 + 数据段 + 效验码。其回复格式为：站地址 + 命令号 +

站地址一个 0~255 的设备标号,占用一个字节。命令号决定设备如何操作,如读位,写位,读寄存器,写寄存器等。起始地址是指寄存器在设备的地址编码,如 VW0 的绝对地址是 2336。操数量,是指操作(读或写)的单元格式。数据段只有写操作有,包含数据的字节长度和数据内容。效验码用于检验传输数据的正确性。他们各自的命令格式及其回复见表 1 和表 2。

表 1 Modbus 读写命令格式

操作(命令号)	命令格式	字节数	举例（十六进制表示）
位 读 取	站地址(1) 功能号(1) 起始地址(2) 数量	8	01 01 00 01 00 02
(01/02)	(2) CRC(2)		CRC

字 读 取	站地址(1) 功能号(1) 起始地址(2) 数量	8	01 03 00 01 00 03
(03/04)	(2) CRC(2)		CRC
位写入(15)	站地址(1) 功能号(1) 起始地址(2) 操作位	9+n	01 0F 00 13 00 0A 02
	数(2) 数据字节数(1) 数据(n) CRC(2)		CD 01 CRC
字写入(16)	站地址(1) 功能号(1) 起始地址(2) 操作字	9+n	01 10 00 01 00 02 04
	节数(2) 数据字节数(1) 数据(n) CRC(2)		00 0A 01 02 CRC

表 2 Modbus 协议读写返回格式

操作(命令号)	返回数据格式	字节数	举例（十六进制表示）
位 读 取	站地址(1)-功能号(1)-字节数(1)-数	5+n	01 01 03 01 00 02 CRC
(01/02)	据(n)-CRC(2)		
字 读 取	站地址(1)-功能号(1)-字节数(1)-数	5+n	01 03 04 01 00 03 01
(03/04)	据(n)-CRC(2)		CRC
位写入(15)	站地址(1)-功能号(1)-起始地址(2)-	8	01 0F 00 13 00 0A CRC
	操作位数(2)-CRC(2)		
字写入(16)	站地址(1)-功能号(1)-起始地址(2)-	8	01 10 00 01 00 02 CRC
	操作字节数(2)-CRC(2)		

数据格式

在读写过程中并不能直接传输数据。位数据需要将各个位从低到高的顺序排列。如读取v0.0-v0.4,返回的数据顺序是: (字节高端)-> v0.4 v0.3 ... v0.1 ->(字节低端)。即是说返回的字节为6 ,二进制为"0000 0101",那么V0.0和V0.2为1。V0.1、V0.3和V0.4为0。写入数据顺序与此相同。

字数据需要将数据进行交换。如图1所示。读取VW0返回的数据为: byte1 byte2,那么VW0的实际数据为 byte2 byte1。如果VW0表示的为一个word类型数据,那么它的真实值为: byte2+byte1*256,即交换为"byte2 byte1"后,强转为word型的值。VD0类似,需要将VW0和VW1分别交换强转为响应的数据类型。可以编写函数改变这种字节顺序。这些都是由Modbus协议规定的。

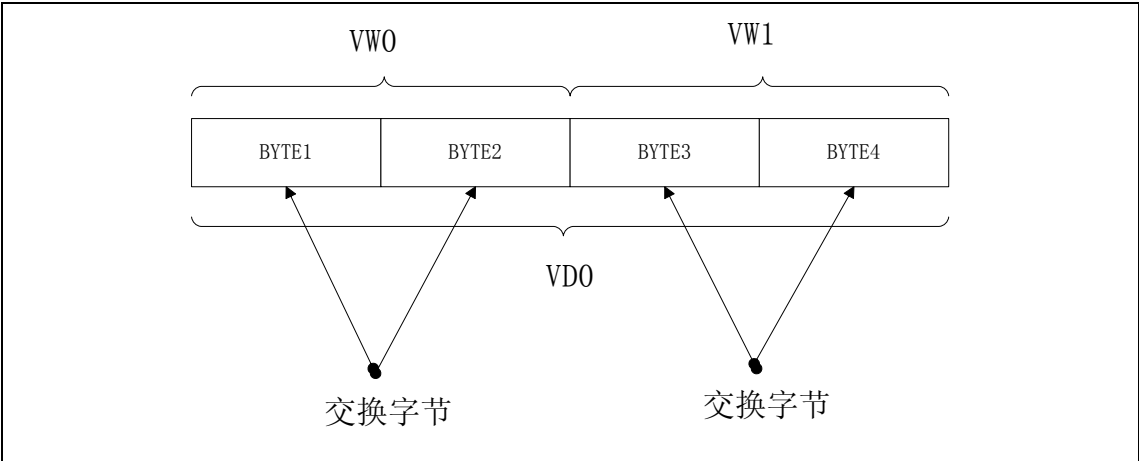


图 1 字节顺序

地址对照表

字寻址

ModbusRTU: 03, 04, 06, 16 号功能, 地址按寄存器寻址, 每个寄存器分配 2 个字节

寄存器名称	内容	字数量	寄存器地址		操作
			十进制	十六进制	
AI	本地模拟量输入	16	0~15	0000~000F	只读
XAI ^[1]	从设备模拟量输入转换值	512	16~527	0010~020F	只读
PAI	从设备模拟量输入原始值	256	528~783	0210~030F	只读
T	定时器区	128	784~911	0310~038F	只读
C	计数器区	128	912~1039	0390~040F	只读
SM ^[2]	特殊寄存器区	512	1040~1551	0410~060F	读写
XAQ ^[3]	从设备模拟量输出转换值	512	1552~2063	0610~080F	读写
AQ	本地模拟量输出值	16	2064~2079	0810~081F	读写
PAQ	从设备模拟量输出原始值	256	2080~2335	0820~091F	读写
V	用户变量区	512	2336~2847	0920~0B1F	读写
M	位变量区	512	2848~3359	0B20~0D1F	读写
S	顺序控制区	16	3360~3375	0D20~0D2F	读写
L	暂存参数区	16	3376~3391	0D30~0D3F	读写
V ^[4]	用户扩展变量区	10240	3392~13631	0D40~353F	读写
RAM	存储指令 RAM 区	10000	20000~29999	4E20~752F	只读

[1] XAI 区每个通道的数据类型为浮点型, 占 2 个寄存器。
[2] 对于 SM 区的前 256 个字为只读, 后 256 个字作为永久存储区, 可读可写, 即从地址

0x0510~0x060F 的区域可以支持寄存器写，部分作为系统功能参数，详细情况见系统参
数说明。

[3] XAQ 区每个通道的数据类型为浮点型，占 2 个寄存器。

[4] 用户扩展变量区 V 延续用户变量区的寻址，即对应于 VW[512]~VW[10751]。

位寻址

ModbusRTU：01，02，05，15 号功能

寄存器名称	内容	字数量	寄存器地址		操作
			十进制	十六进制	
I	本地数字量输入	4	0~63	0000~003F	只读
XI	从设备数字量输入	256	64~4159	0040~103F	只读
T	定时器区	8	4160~4287	1040~10BF	只读
C	计数器区	8	4288~4415	10C0~113F	只读
SM ^[1]	特殊标志位区	512	4416~12607	1140~313F	读写
XQ	从设备数字量输出	256	12608~16703	3140~413F	读写
Q	本地数字量输出	4	16704~16767	4140~417F	读写
V	用户变量区	512	16768~24959	4180~617F	读写
M ^[2]	位变量区	256	24960~29055	6180~717F	读写
S	顺序控制区	16	29056~29311	7180~727F	读写
L	暂存参数区	16	29312~29567	7280~737F	读写

[1] 对于 SM 区的前 256 个字为只读，后 256 个字作为永久存储区，可读可写，即从地址 0x1940~0x213F 的区域可以支持寄存器的写。

[2] 对于 M 区的位变量，由于寄存器地址的限制，只支持前 256 个字的位寻址。