MODBUS通讯协议

PLC WORLD

拟　　制：　谷鹏/陈林　　日期：2004-8-5

审　　核：　　日期：

审　　核：　　日期：

规格更改、修改记录

修改日期	版本	更改、修订内容	拟制人
2004-8-3	D01	初稿	谷鹏
2004-10-27	D02	（1）根据讨论删除了部分功能码 （2）修改了寻址方式	谷鹏
2004-11-19	D03	（1）更改软元件的寻址范围 （2）更改支持的功能码 （3）对具体的功能码的解释	陈林
2004-11-23	B01	基线化	谷鹏
2004-12-16	B01D01	更改其中的两处描述错误	陈林
2004-12-17	B01D02	（1） 增加通信举例 （2） 增加广播说明 （3） 细化元件寻址描述 （4） 更改微小的描述错误	陈林
2004-12-18	B01D03	增加详细描述	陈林
2004-12-19	B02	基线化	谷鹏

目录

1. 协议简介

Modbus 协议是应用于控制器上的一种通用语言。通过此协议，控制器相互之间、控制器经由网络和其它设备之间可以通信。它已经成为一通用工业标准。本规范主要描述了 modbus 协议在 GCM 中的实现。通讯采用应答方式，由主机发起请求，从机执行请求并且应答。GCM 作为从机通过地址设置加以区分， GCM 系列 PLC 自己组网时，最多允许 30 个从站，并且从机可设置地址范围为 1~31。GCM 系列 PLC 作为主站与其他设备（做从站）组网，没有地址范围限制，但也最多允许 30 个从站。GCM 系列 PLC 支持广播方式，广播地址为 00。

2. 接口方式

RS485 或 RS232 接口：异步，半双工。

默认数据格式： 8位数据位、偶校验、一位停止位，19200 bps,RTU。可设置为38，400 波特率、19，200 bps、9，600 bps、4，800 bps、2，400 bps、1，200 bps；最高可设置波特率为38，400bps。

数据域：支持2×252个字节（ASII模式）、252字节（RTU模式）

GCM系列PLC有两个通信口，其通信口0（也作为编程口）支持MODBUS从站，通信口1支持MODBUS主站和从站（可有后台软件设置）。

3. 协议格式

3.1. RTU模式

起始（至少 3.5 个字符空闲）	从机地址	功能代码	数据	CRC 高字节	CRC 低字节 PLC WORLD	结束（至少 3.5 个字符空闲）

消息发送至少要以 3.5 个字符时间的停顿间隔开始。在最后一个传输字符之后，一个至少 3.5 个字符时间的停顿标定了消息的结束。一个新的消息可在此停顿后开始。

整个消息帧必须作为一连续的流转输。如果在帧完成之前两个字符间有超过 1.5 个字符时间的停顿时间，认为帧错误，停止接收，清缓冲，直到通信主循环中，清错误标志（与 PLC 寄存器无关），重新启动接收。

也就是要保证两个帧间的间隔至少大于 3.5 个字符的时间，3.5 个字符的时间与具体的通信波特率有关，计算方法如下：

假如通信波特率为 19200，那么 1.5 个字符间隔 = 1/19200 *11*1.5*1000=0.86ms，

3.5 个字符间隔 = 1/19200 *11*3.5*1000=2ms。

下面是请求帧为读取 1 号机的 002 参数的数据帧：

地址	功能码	寄存器地址		读取字数		校验和	
0x01	0x03	0x00	0x02	0x00	0x01	0x25	0xCA

下面是为 1 号机的响应帧：

地址	功能码	应答字节数		寄存器内容		校验和	
0x01	0x03	0x00	0x02	0x13	0x88	0xE9	0x5C

3.2. ASCII模式：

起始字符 0x3A	从机地址	功能码	数据域	LRC 高字节	LRC 低字节	结束字符 0DH,0AH

ASCII 方式下：帧头为“ 0x3A ”,帧尾为 ”0x0D ” ”0x0A ”。消息中字符间发送的时间间隔最长不能超过 1 秒，否则接收的设备将认为传输错误。在 ASCII 方式下，数据字节全部以 ASCII 码方式发送，先发送高 4 位位元组，然后发送低 4 位位元组。例如：01，会传输 30，31 两个 ASCII 字符。此时数据采用 LRC 校验，校验涵盖从从机地址到数据的信息部分。校验和等于所有参与校验数据的字符和(舍弃进位位)的补码+1。

ASCII 方式 Modbus 数据帧举例如下:

写入 4000(0xFA0)到从机 1 的内部寄存器 002 如下表：

LRC 校验=(01+06+00+02+0x0F+0xA0)的补码=0x48

	帧头	地址		功能码		寄存器地址				写入内容				LRC 校验		帧尾	
字符	:	0	1	0	6	0	0	0	2	0	F	A	0	4	8	CR	LF
ASCII	3A	30	31	30	36	30	30	30	31	30	46	41	30	34	38	0D	0A

地址域

消息帧的地址域包含两个字符（ASCII）或 8Bit（RTU）。主设备通过将要联络的从设备的地址放入消息中的地址域来选通从设备。当从设备发送回应消息时，它把自己的地址放入回应的地址域中，以便主设备知道是哪一个设备作出回应。

地址 0 是用作广播地址，以使所有的从设备都能认识。

3.3. 字符的连续传输

当消息在标准的 Modbus 系列网络传输时，每个字符或字节以如下方式发送（从左到右）：

最低有效位...最高有效位

ASCII 传输模式：采用 10 位传输，一定是 7 位的数据位，位序如下.

有奇偶校验 只能 1 个停止位

启始位	1	2	3	4	5	6	7	奇偶位	停止位

无奇偶校验 必须是 2 个停止位

启始位	1	2	3	4	5	6	7	停止位	停止位

图 4. 位顺序（ASCII）

RTU 传输模式，采用 11 位传输，一定是 8 位数据位，位的序列是：

有奇偶校验

启始位	1	2	3	4	5	6	7	8	奇偶位	停止位

无奇偶校验

启始位	1	2	3	4	5	6	7	8	停止位	停止位

3.4. 帧类型

MODBUS 的帧结构分为三种，请求帧、应答帧、错误帧三种。
主站的询问可能有以下几种情况：

（1） 从站收到了无通讯错误的请求，并进行正常处理，从站返回应答（正常帧）
（2） 从站收到的请求，但有通讯错误，则不进行返回，主站认为超时
（3） 从站收到了无通讯错误的请求，但不能处理这一通讯请求。则返回错误帧

3.4.1. 请求帧

功能代码	数据

3.4.2. 应答帧

功能代码	数据

功能码域复制请求帧的功能码

3.4.3. 错误帧三种

功能代码	错误代码

功能码是截获搜请求帧的功能码＋0x80

4. 功能码描述

4.1. ModBus功能码

我们支持的功能码如下：

功能码	名称	作用	协议支持
0x01	读取线圈状态	取得一组逻辑线圈的当前状态（ON/OFF）	○
0x 02	读取输入状态	取得一组开关输入的当前状态（ON/OFF）	○
0x 03	读取保持寄存器	在一个或多个保持寄存器中取得当前的二进制值	○
0x 05	强置（写）单线圈	写一个逻辑线圈的通断状态	○
0x 06	预置（写）单寄存器	把具体二进值装入一个数据寄存器	○
0x 08	回送诊断校验	把诊断校验报文送从机，以对通信处理进行评鉴	○
0x0f	强置（写）多线圈	强置（写）一串连续逻辑线圈的通断	○
0x10	预置（写）多寄存器	把具体的二进制值装入一串连续的保持寄存器	○

4.2. 元件的寻址方式描述

位元件的寻址

00001～00256 Y0～Y255 (Discrete Output)

01201～01456 X0～X255 (Discrete Input)

02001～04000 M0～M1999 (Discrete M Relay)

04401～04656 SM0～SM255 (Discrete M Relay)

06001～06992 S0～S991 (Discrete S Relay)

08001～08256 T0～T255 (Status of T0～T255)

09201～09456 C0～C255 (Status of C0～C255)

离散量输入的寻址（针对X位元件）
00001-00256

字元件的寻址
00001～08000 D0～D7999

08001～08256 SD0～SD255

08501～08516 Z0～Z15
09001～09256 T0～T255 (Current Value of T0～T255)

09501～09700 C0～C199 (Current Value of C0～C199，16-bit)

09701～09812 C200～C255 (Current Value of C200～C255，32-bit)

以上的地址是面向用户地址，例如触摸屏，输入逻辑地址1，对地址0寻址（发送按照协议地址0发送）；但是当用户使用GCM系列的PLC作主站，需要自己组织发送的数据报文，那么用户需要输入实际的协议地址，例如用户需要读取Y0，Y0的协议地址是0，用户需要组的帧01 01 00 00 00 01，第一个01是地址（可设的），第二个01功能码，接下来的00 00表示起始地址（Y0的地址），00 01是读取的个数1个。

（1）读写元件功能码和与内存映射：

功能码	功能码名称	Modicon 数据地址	可操作元件类型	注释
01	读线圈	0:xxxx	Y、X、M、SM、S、T、C	读位
02	读离散量输入	1:xxxx	X	读位
03	读保存寄存器	4:xxxx	D、SD、Z、T、C	读字
05	写单个线圈	0:xxxx	Y、M、SM、S、T、C	写位
06	写单个寄存器	4:xxxx	D、SD、Z、T、C	写字
15	写多个线圈	0:xxxx	Y、M、SM、S、T、C	写位
16	写多个寄存器	4:xxxx	D、SD、Z、T、C	写字

注：0 表示线圈，1 表示离散量输入，4 表示寄存器，xxxx 表示范围 1-9999，每一种类型有独立的逻辑地址范围就是 1-9999（协议地址是从 0 开始的），0、1、4 并不具备物理上的意义，不参加实际的寻址。

（2）PLC 元件与 Modbus 传输中的协议地址的对应关系：

元件类型	物理元件	协议地址	支持的功能码	注释
Y	Y0-Y255	0000-0255	01、05、15	输出的状态

X	X0-X255	1200-01455 0000-0255	01、05、15 02	输入的状态，支持两种地址
M	M0-M2000	2000-3999	01、05、15	
SM	SM0-SM256	4400-4655	01、05、15	
S	S0-S991	6000-6991	01、05、15	
T	T0-T255	8000-8255	01、05、15	T 元件的状态
C	C0-C255	9200-9455	01、05、15	C 元件的状态
D	D0-D7999	0000-7999	03、06、16	
SD	SD0-SD255	8000-8255	03、06、16	
Z	Z0-Z15	8500-8515	03、06、16	
T	T0-T255	9000-9255	03、06、16	T 元件的当前值
C	C0-C199	9500-9699	03、06、16	C 元件（WORD）的当前值
C	C200-C255	9700-9811	03、16	C 元件（DWORD）的当前值

注：协议地址与 Modicon 的数据的逻辑地址有对应关系，协议地址是从 0 开始，Modicon 的数据的逻辑地址是从 1 开始的，也就是说协议地址+1=Modicon 的数据的逻辑地址，例如：M0 协议地址是 2000，它对应的 Modicon 的数据的逻辑地址是 0:2001，在实际对 M0 的读写是通过协议地址完成，例如对 M2000 元件的读取帧（主站发出）：

01 01 07 D0 00 01 FD 47

01 代表站号；

01 代表功能码；

07 D0 代表起始地址，07D0 的十进制为 2000；

00 01 代表读取的元件个数；

FD 47 是 CRC 校验码；

特别注意：对 SD、SM 的写入是要看 SD、SM 的读写属性（参看编译规约），如果该特殊元件不是可写入的，从站对该报文不作处理，但会返回正确的应答。

4.3. 错误代码描述

异常代码	异常代码意义	实现

0x01	非法功能码。	○
0x02	非法寄存器地址。	○
0x03	非法数据。	○

5. 数据和控制码的具体描述

5.1. 读取线圈状态(0x01 Read Coil Status)

读取从站的位元件的状态，不支持广播。最多支持 2000 个位元件。

参考软元件的地址划分，每次读取的元件类型为 1 类，例如不能在一帧中将 X 和 Y 元件（两种类型）一起读回来；读取该类软元件的地址和数据范围不能超过协议中规定的范围，例如：Y 元件的协议地址范围 0000～0255（ Y0～Y255），如果读取的起始地址是 1，读取的元件个数是 256，则会返回地址错误（错误码 02），因为从 1 起始的 Y 元件只有 255 个；读取的起始地址是 0，读取的元件个数是 257，则会返回数据错误（错误吗 03），因为读取元件的个数超过了 256，实际只定义了 256 个 Y 元件；读取的起始地址为 0，读取元件的个数是 256 则会返回 256 个元件的状态；而在读取 M 元件时，M 元件的协议地址范围是 2000～3999（M0～M1999），如果读取的起始地址是 2000，那么读取的元件个数是 2000 个就会返回 2000 个 M 元件的状态。也就是必须保证读取的元件是实际定义的（在范围内）。

（1） 请求帧

Address	Function code (01H)	起始地址		元件个数		校验码 (CRC或LRC)
		H	L	H	L	

（2）响应帧

如果读取的地址不是 8 的倍数，剩下的位由 0 填充（由高位开始填充）。

Address	Function code (01H)	Number of byte read读取的元件个数（字节数）(n)	Read data 读取的数据 No.1	……	Read data 读取的数据 No.n	校验码 (CRC或LRC)

B7	B6	B5	B4	B3	B2	B1	B0

5.2.读取离散量输入状态(0x02 Read Input Status)

读取从站的位元件的状态，不支持广播。最多支持 256 个位元件（目前的型号只定义

256 个 X 元件)。

（2） 请求帧

Address	Function code (02H)	起始地址		元件个数		校验码 (CRC或LRC)
		H	L	H	L	

（2）响应帧

如果读取的地址不是 8 的倍数，剩下的位由 0 填充（由高位开始填充)。

Address	Function code (02H)	Number of byte read读取的元件个数（字节数）(n)	Read data 读取的数据 No.1	……	Read data 读取的数据 No.n	校验码 (CRC或LRC)

B7	B6	B5	B4	B3	B2	B1	B0

5.3.读取保持寄存器（0x03 Read Holding Registers)

读取保持寄存器是读取从站的数据（字）寄存器值，(最多可以有 125 个数据寄存器)。不支持广播。

（1） 请求帧

Address	Function code (03H)	起始地址		元件个数		校验码 (CRC或LRC)
		H	L	H	L	

（2）响应帧

Address	Function code (03H)	Number of byte read读取的元件个数（字节数）(n)	Read data 读取的数据 No.1		……	Read data 读取的数据 No.n		校验码 (CRC或LRC)
			H	L		H	L	

5.4. 强置（写）单线圈（0x05 Force Single Coil)

强置(写)单线圈是向从站写入位元件值，与编程协议的强制不同。允许广播(broadcast)，即写入所有从站的相同元件。最多支持 1 个位元件。

请求帧

Address	Function code (05H)	起始地址		写入的元件值		校验码 (CRC或LRC)
		H	L	H	L	

注：写入元件的值为 0xFF00(ON,1)或者 0x0000(OFF,0)

（2）响应帧

响应帧是请求帧的重复。

Address	Function code (05H)	起始地址		写入的元件值		校验码 (CRC或LRC)
		H	L	H	L	

5.5. 预置（写）单寄存器（0x06 Preset Single Register）

强置（写）单寄存器是向从站写入字元件值，与编程协议的强制不同。允许广播(broadcast)，即写入所有从站的相同元件。最多支持 1 个字元件。

请求帧

Address	Function code (06H)	起始地址		写入的元件值		校验码 (CRC或LRC)
		H	L	H	L	

（2）响应帧

响应帧是请求帧的重复。

Address	Function code (06H)	起始地址		写入的元件值		校验码 (CRC或LRC)
		H	L	H	L	

5.6. 回送诊断校验

回送诊断校验，可以得到诊断寄存器并得到通讯错误信息。

诊断码		
0x00	Return Query Data	请求帧返回
0x 01	Restart Comm Option	重启通信选项
0x 04	Force Listen Only Mode	从机进入 LISTEN ONLY 模式
0x0a	Clear Ctrs and Diagnostic Reg	清除计数器和诊断寄存器
0x0b	Return Bus Message Count	返回总线报文计数
0x0c	Return Bus Comm. Error Count	返回总线 CRC 错误计数
0x0d	Return Bus Exception Error Cnt	返回从站异常差错计数
0x0e	Return Slave Message Count	返回从站报文计数

0x0f	Return Slave No Response Cnt	返回从站无响应计数
0x12	Return Bus Char. Overrun Cnt	返回总线字符超限计数

5.6.1. 请求帧返回

（1） 请求帧

Address	Function code (0x08H)	功能字码		任意字符		校验码 (CRC或LRC)
		(0x00H)	(0x00H)	H	L	

（2）响应帧

将请求帧原样返回

Address	Function code (0x08H)	功能字码		任意字符		校验码 (CRC或LRC)
		(0x00H)	(0x00H)	H	L	

注：作为主站能发送的任意字符长度为 2，作为从站接收任意字符只受帧长度的限制。

5.6.2. 重新启动通信选项

当收到该帧时，可以将 PLC 带出只听模式。（支持广播帧）

（1） 请求帧

当数据域正常为 0x00 00 或者 0xff 00。

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或 LRC)
		0x00H	0x01H	H	L	

（2）响应帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或 LRC)
		0x00H	0x01H	H	L	

5.6.3. 从机进入LISTEN ONLY 模式

从站进入 LISTEN ONLY 模式，命令都不执行，也不作回应，从站只认重新启动通信选

项命令，并且当收到该命令后，从站进入在线模式。（支持广播帧）

（1） 请求帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x04H）	0x00H	0x00H	

（2）响应帧

无返回

5.6.4. 清计数器和诊断寄存器

清除所有的计数器（支持广播帧）。

（1） 请求帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0AH）	0x00H	0x00H	

（2）响应帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0AH）	0x00H	0x00H	

5.6.5. 返回总线报文计数

记录从上一次启动、清除计数器或加电之后，从站在到的所有主站发出的报文总数，不包括 CRC 错误的报文。

（1） 请求帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0BH）	0x00H	0x00H	

（2）响应帧

Address	Function code	功能字码	数据域	校验码 (CRC或LRC)

	（0x08H）	（0x00H）	（0x0BH）	H	L	

5.6.6. CRC错误计数值

记录从上一次启动、清除计数器或加电之后，从站在到的 CRC 错误的数量。

（1） 请求帧

Address	Function code （0x08H）	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0CH）	0x00H	0x00H	

（2）响应帧

Address	Function code （0x08H）	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0CH）	H	L	

5.6.7. 返回从站异常差错计数

记录从上一次启动、清除计数器或加电之后，从站检测到的异常差错数量，也包括广播报文中的检测到的差错。

（1） 请求帧

Address	Function code （0x08H）	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0DH）	0x00H	0x00H	

（2）响应帧

Address	Function code （0x08H）	功能字码		数据域		校验码 (CRC或LRC)
		（0x00H）	（0x0DH）	H	L	

5.6.8. 返回从站报文计数

记录从上一次启动、清除计数器或加电之后，从站收到的对从站寻址的报文数目。

（1） 请求帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		(0x00H)	(0x0EH)	0x00H	0x00H	

(2) 响应帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		(0x00H)	(0x0EH)	H	L	

5.6.9. 返回从站无响应计数

记录从上一次启动、清除计数器或加电之后，从站没有返回的报文数量。

(1) 请求帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		(0x00H)	(0x0FH)	0x00H	0x00H	

(2) 响应帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		(0x00H)	(0x0FH)	H	L	

5.6.10. 从站收到字符超限计数值

记录从上一次启动、清除计数器或加电之后，由于收到的字符超限导致无法寻址的的报文数量。

(1) 请求帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		(0x00H)	(0x12H)	0x00H	0x00H	

(2) 响应帧

Address	Function code (0x08H)	功能字码		数据域		校验码 (CRC或LRC)
		(0x00H)	(0x12H)	H	L	

5.7. 强置（写）多线圈(0x0F Hex)Force Multiple Coils

最多能写 1968 个（0x07b0）,根据元件的定义范围有变化。支持广播。

（1） 请求帧

Address	Function code (0FH)	起始地址		元件个数		字节数 (n)	写入元件值 No.1	••• •••	写入元件值 No.N	校验码(CRC或LRC)
		H	L	H	L					

B7	B6	B5	B4	B3	B2	B1	B0

（2）响应帧

Address	Function code (0FH)	起始地址		元件个数		校验码 (CRC或LRC)
		H	L	H	L	

5.8. 预置（写）多寄存器(10 Hex) Preset Multiple Registers

最多写 120 个寄存器（0x78）支持广播。

（1） 请求帧

Address	Function code (0x10H)	起始地址		元件个数		字节数 (n)	写入元件值 No.1		••• •••	写入元件值 No.N		校验码(CRC或LRC)
		H	L	H	L		H	L		H	L	

（2）响应帧

Address	Function code (0x10H)	起始地址		元件个数		校验码 (CRC或LRC)
		H	L	H	L	

5.9. 故障响应帧（0x80+功能码）

响应帧：

Address	Function code （功能码）	错误代码（见前）	校验码 (CRC或LRC)

功能码是截获搜请求帧的功能码＋0x80

指令的错误代码：

异常代码	异常代码意义	
0x01	非法功能码。	
0x02	非法寄存器地址。	
0x03	数据个数错误。	

5.10. MODBUS通信控制举例

Modbus 从站不主动发送任何报文，只有接收到对本地的信息后才根据具体情况看是否响应主站的报文。从站仅支持 Modbus 功能码 01，02，03，05，06，08，15，16，其余的均以不合法的功能码响应（除广播帧）。

5.10.1. 读取双字元件的处理

C 元件的当前计数值为字元件或双字元件，C200-C255 为双字元件，对 C200-C255 的读写也是通过读写寄存器的功能码（03、16）来完成，每两个寄存器的地址对应一个 C 双字元件，读写时只能成对的读取寄存器。例如读取 C200-C202 三个 C 双字元件 RTU 帧：

01 03 25 E4 00 06 8E F3

25 E4 为起始地址 9700

00 06 表示读取 6 个元件

8E F3 为校验码

返回的数据中 9700 9791 两个地址表示 C200 的内容，9700 为高 16bit，9701 为低 16bit。

在读取双字元件时如果读取的开始地址不是偶数，会返回错误代码非法地址，如果读取的个数不是偶数，会返回错误代码非法的数据。

例如：主站发送：01 03 25 E5 00 04 5E F2

主站发送读取开始地址是 25 E5（十进制 9701）的四个字元件，

从站响应：01 83 02 C0 F1

从站应答：非法的数据地址

从站发送：01 03 25 E4 00 05 CE F2

主站读取开始地址是 25E4 的 5 个字元件

从站应答：01　83　03　01　31

从站返回非法的数据

5.10.2. 读取LONG INT类型数据的处理

PLC 元件的存储方式，一个 LONG INT 类型，可能存在两个 D 元件内，例如：D3，D4 存一个 LONG INT 型的数，EMERSON PLC 认为 D3 存储的是高 16 位，D4 存储的是低 16 位，当主站通过 MODBUS 读取 LONG INT 数据时，读回数据后也应该按照 EMERSON PLC 对 LONG INT 的存储原则重组 32 位的数据。FLOAT 的存储原则等同于 LONG INT 的存储原则。

5.10.3. 对元件读取的处理

除了 08 功能码，其他支持的功能码都是对元件读写的，原则上讲最多允许一帧读 2000 个位元件，写 1968 个位元件 ，读取 125 个字元件，写 120 个字元件。但由于实际的协议地址对不同的元件是分开的，不连续（例如 Y255 的协议地址是 255，X0 的协议地址是 1200），因此在对元件的读写操作时，一次读取的元件只能是一种类型的元件，而读取元件的最多数目也与实际定义的元件个数有关系，例如读取 Y 元件，Y0-Y255，协议地址范围 0-255，对应的 Modicon 的数据的逻辑地址是 1-256，在读取 Y 元件时最多允许读取 256 个元件，举例如下：

注：从站的地址都是 01，后两个字节都是 CRC 校验码，第二字节是功能码。

（1） 主站发送：01　01　00　00　01　00　3D　9A

01 地址，功能码 01，00 00 起始地址，01 00 读取元件个数 3D 9A 校验

从站应答：会返回正确的应答

（2） 主站发送：01　01　00　00　01　01　FC　5A

主站读取 01 01 个元件也就是 257 个，超出了定义的 Y 元件的范围

从站应答：01　81　03　00　51

从站应答是非法的数据值，原因是 257>256，256 是最大允许的 Y 元件数

（3） 主站发送：01　01　00　64　00　A0　7D　AD

主站读取起始地址 00 64（十进制 100），元件个数 00 A0（十进制 160）

从站应答：01　81　02　C1　91

从站应答非法数据地址，从 100 开始的 Y 元件只有 156 个读 160 个非法。

（4） 主站发送：01　01　01　2C　00　0A　7C　38

主站读取 01 2C（十进制 300）的 10 个位元件

从站应答：01　81　02　C1　91

从站应答非法数据地址，由于协议地址 300 没有位元件的定义。

（5）　主站发送：01　04　00　02　00　0A　D1　CD

主站发送功能码 04 的帧

从站应答：01　84　01　82　C0

从站应答非法的功能码

（6）　主站发送：01　02　00　00　00　0A　F8　0D

主站读取输入元件（X 元件）从起始地址 00 00 读 10 个（X0-X9）

从站返回：01　02　02　00　00　B9　B8

从站返回了正确信息 02 个字节内容是 00 00

（7）　主站发送：01　01　04　B0　00　0A　BC　DA

主站读取 04 B0（十进制 1200）开始的 10 个位元件，也是 X0-X9

从站应答：01 01 02 00 00 B9 FC

从站应答 02 个字节内容是 00 00

注：X 元件不支持写入，SM，SD 元件的可写属性请参考元件列表。

5.11.　对广播的描述

从站支持广播，但不是每一个功能码都支持，从站支持的功能码 01，02，03，05，06，08，15，16（十进制），其中为 01，02，03 读取元件不支持广播，发送了广播会没有返回；05，06，15，16 是写元件支持广播的功能码，发送了广播没有返回，但从站会处理接收的数据；08 是诊断功能码，它的子功能码 0x01，0x04，0x0A（16 进制）支持广播，其他的不支持广播。