

## 数显表通讯协议(适用于GD80/90系列)

通讯采用ModBus规约N81格式(8个数据位、一个停止位、没有校验位)。

数据帧格式为:地址码(一字节)、功能码(一字节)、数据区、CRC校验(两字节)。

读取功能码为03。(下位机回送所有数据均为两字节)

读取测量数据发送:地址、03、 00、00、 00、03、 CRC (高字节,低字节)  
数据地址 读取个数(三相需一次读取)

下位机回送:地址、03、06(回送字节数)、测量数据(六字节A相B相C相,按“高字节、低字节”排)、CRC。

例:地址为1,测量数据为500.0(数据全部为16进制)

发送: 01 03 00 00 00 03 05 CB

接收: 01 03 06 13 88 13 88 13 88 4A 31

其他数据寄存器地址:00 00 (测量数据) 00 06 (小数点位置) 03 08 (波特率,  
0—1200, 1—2400, 2—4800, 3—9600, 4—19200) 03 02 (上  
限值,不带小数点) 03 00 (地址,只用于修改地址) 03 12  
(变比)

写入功能码为06。(下位机回送与发送的完全相同)

数据格式:地址、06、 XX、XX、 XX、XX、 CRC (两字节)  
数据地址 写入数据

例:地址为1,修改上限阈值为5000

发送: 01 06 03 02 13 88 25 18

接收: 01 06 03 02 13 88 25 18 (13 88对应的十进制5000写入上限阈值寄存  
器)

注意:00 00 寄存器地址对应的测量数据寄存器无法写入,只能读出。

03 00 寄存器地址对应的本机地址只能写入修改值,无法读出。

以上是就三相数显表而言的,对于单相数显表,只需稍作修改即可。

## 附录：-----通讯规约

### 1. ModBus基本规则:

- 1.1 所有RS485通讯回路都应遵照主/从方式。依照这种方式，数据可以在一个主站（如：PC）和32个子站（如：GD2000）之间传递。
- 1.2 主站将初始化和控制在RS485通讯回路上传递的所有信息。
- 1.3 任何一次通讯都不能从子站开始。
- 1.4 在RS485回路上的所有通讯都以“信息帧”方式传递。
- 1.5 如果主站或子站接收到含有未知命令的信息帧，则不予以响应。

“信息帧”就是一个由数据帧（每一个字节为一个数据帧）构成的字符串（最多255个字节），是由信息头和发送的编码数据构成标准的异步串行数据，该通讯方式也与RTU通讯规约相兼容。

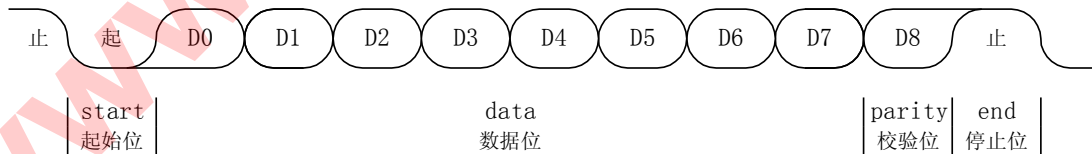
### 2. 数据帧格式:

通讯传输为异步方式，并以字节（数据帧）为单位。在主站和子站之间传递的每一个数据帧都是11位的串行数据流。

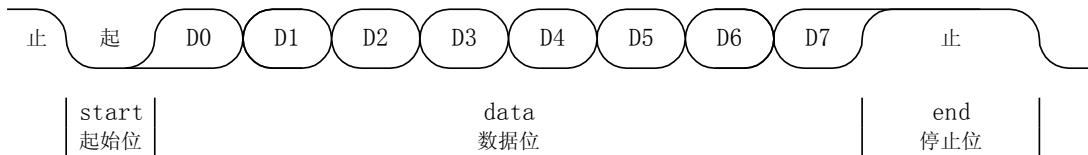
数据帧格式:

起始位	1位
数据位	8位（低位在前、高位在后）
奇偶校验位	1位：有奇偶校验位；无：无奇偶校验位
停止位	1位：有奇偶校验位；2位：无奇偶校验位

有校验位的时序图:



无校验位的时序图:



### 3. 通讯规约:

当通讯命令发送至仪器时，符合相应的地址码的设备接收通讯命令，并除去地址码，读取信息，如果没有出错，则执行相应的任务；然后把执行结果返送给发送者。返送的信息中包括地址码、执行动作的功能码、执行动作后的数据以及错误校验码（CRC）。如果出错就不发送任何信息。

#### 3.1. 信息帧格式:

START	ADD	CS	DATA	CRC	END
初始结构	地址码	功能码	数据区	错误校验	结束结构
延时（相当于4个字节的 时间）	1字节 8位	1字节 8位	N字节 N×8位	2字节 16位	延时（相当于4个字节的 时间）

##### 3.1.1. 地址码（ADD）:

地址码为每次通讯传送的信息帧中的第一个数据帧（8位），从0到255。这个字节表明由用户设定地址码的子机将接收由主机发送来的信息。并且每个子机都有唯一的地址码，并且响应回送均以各自的地址码开始。主机发送的地址码表明将发送到的子机地址，而子机发送的地址码表明回送的子机地址。

##### 3.1.2. 功能码（CS）:

功能码是每次通讯传送的信息帧中的第二个数据帧。作为主机请求发送，通过功能码告诉子机执行什么动作。作为子机响应，子机发送的功能码与主机发送来的功能码一样，并表明子机已响应主机进行操作。如果子机发送的功能码的最高位是1（功能码>127），则表明子机没有响应或出错。

下表列出的功能码都具有具体的含义及操作。

MODBUS部分功能码

功能码	定义	操作
03H	读寄存器	读取一个或多个寄存器的数据
06H	写寄存器	把16位二进制数写入寄存器

### 03, 读寄存器:

功能码03H映射的数据区的保持和输入寄存器值都是16位（2字节）。

子机响应的命令格式是子机地址、功能码、数据区及CRC码。数据区的数据都是每2个字节为一组的双字节数，且高字节在前。

### 06，写寄存器：

Modbus通讯规约中的寄存器值是16位(即2字节)，并且高位在前。下位机回送与发送的完全一样。

#### 3.1.3. 数据区 (DATA)：

数据区随功能码不同而不同。由主机发送的读命令（03H）信息帧的数据区与子机应答信息帧的数据区是不同的，由主机发送的写命令（06H）信息帧的数据区与子机应答信息帧的数据区是完全相同。数据区包含需要子机执行什么动作或由子机采集的需要回送的信息。这些信息可以是数值、参考地址等等。例如，功能码告诉子机读取寄存器的数值，则数据区必须包含要读取寄存器的起始地址及读取长度（寄存器个数）。

##### 1、与功能码03对应的数据区格式：

主机发送

数据顺序	1	2
数据含义	起始地址	读寄存器个数
字节数	2	2

子机应答

数据顺序	1	2
数据含义	回送字节数	N个寄存器的数据
字节数	1	2×N

##### 与功能码06对应的数据区格式：

数据顺序	1	2
数据含义	寄存器地址	写入数据1
字节数	2	2

### 3.1.4. 错误校验码 (CRC):

主机或子机可用校验码进行判别接收信息是否出错。有时, 由于电子噪声或其他一些干扰, 信息在传输过程中会发生细微的变化, 错误校验码保证了主机或子机对在传送过程中出错的信息不起作用。这样增加了系统的安全和效率。错误校验码采用CRC-16校验方法。

二字节的错误校验码, 低字节在前, 高字节在后。

注意: 信息帧的格式都是相同的: 地址码、功能码、数据区和错误校验码。

### 3.2. 错误校验

冗余循环码 (CRC) 包含2个字节, 即16位二进制。CRC码由发送端计算, 放置于发送信息的尾部。接收端的设备再重新计算接收到信息的CRC码, 比较计算得到的CRC码是否与接收到的相符, 如果二者不相符, 则表明出错。

CRC码的计算方法是, 先预置16位寄存器全为0。再逐渐把每8位数据信息进行处理。在进行CRC码计算时只用8位数据位, 起始位及停止位, 如有奇偶校验位的话也包括奇偶校验位, 都不参与CRC码计算。

在计算CRC码时, 8位数据与寄存器的数据相异或, 得到的结果向低位移一位, 用0填补最高位。再检查最低位, 如果最低位为1, 把寄存器的内容与预置数相异或, 如果最低位为0, 不进行异或运算。

这个过程一直重复8次。第8次移位后, 下一个8位再与现在寄存器的内容相异或, 这个过程与上以上一样重复8次。当所有的数据信息处理完后, 最后寄存器的内容即为CRC码值。

### 3.3. CRC-16码的计算步骤为:

置16位寄存器为十六进制FFFF (即全为1)。称此寄存器为CRC寄存器。

把一个8位数据与16位CRC寄存器的低位相异或, 把结果放于CRC寄存器。

把寄存器的内容右移一位 (朝低位), 用0填补最高位, 检查最低位 (移出位)。

如果最低位为0: 复第3步 (再次移位)。

如果最低位为1: CRC寄存器与多项式A001 (1010 0000 0000 0001) 进行异或。

重复步骤3和4, 直到右移8次, 这样整个8位数据全部进行了处理。

重复步骤2到步骤5，进行下一个8位的处理。

最后得到的CRC寄存器即为CRC码，低字节在前，高字节在后。

### 3.4. 信息帧格式举例

#### 3.4.1. 功能码03:

00 03 00 00 00 03 04 1A

CRC码低字节  
CRC码高字节  
数据低字节  
数据高字节  
地址低字节  
地址高字节  
功能码  
子机地址

#### 3.4.2. 功能码06

子机地址为00，把0064保存到地址0004。

00 06 00 04 00 64 C8 31

CRC码低字节  
CRC码高字节  
数据低字节  
数据高字节  
地址低字节  
地址高字节  
功能码  
子机地址

附录1：数据和地址。

表1：功能码03H所映射的数据区-基本数据：

基本数据 (Basic)		
地址(Address)	项目(Item)	说明
0000H	A相测量数据	
0002H	B相测量数据	单相时不用
0004H	C相测量数据	单相时不用
0006H	小数点位置	乘变比之前

表2：功能码06H所映射的数据区：

项目Item	起始地址 Start address	取值范围Value	单位Unit
地址	03 00H	1-32	
继电器阈值	03 02H	0-9999	
波特率	03 08H	0-4	
CT	03 12H	1~64000	