

通信规约

电源分配控制单元 MODBUS 通信协议 INP 型

文件编号:	编写: 杨国军
版本: V1.03	审核:
日期: 2017-06-09	批准:

1. 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。默认波特率 115200,数据位 8,停止位 1,无校验;

485 线序定义: 1/2: TXD; 3/6: RXD

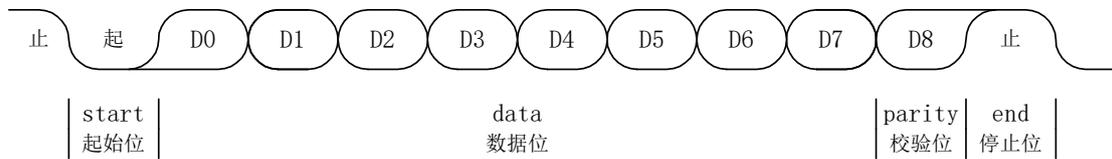
5. 485 座子 con2.con3

RJ45: 1/2: TXD 3/6: RXD 4/5: 电源 + 7/8: 电源 -

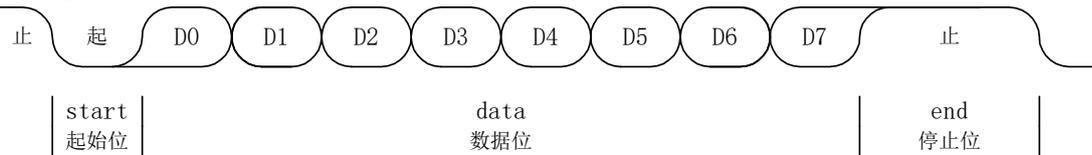
数据帧格式:

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:



2. 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。

2.1. 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)

2.1.1. 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。

2.1.2. 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。ModBus 通讯规约定义功能码为 1~127(01H~7FH)。DCM3366 型电表利用其中的一部分功能码。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
06H	写单个寄存器	写入单个寄存器

10H

写多个寄存器

写入多个寄存器

1、 03, 读寄存器

DCM3366D-4-W1 型电表采用 ModBus 通讯规约, 利用通讯命令, 可以进行读取点 (保持寄存器或返回值输入寄存器)。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)。这样读取的寄存器值都是 2 字节。一次最多可读取寄存器数是 40。由于一些可编程控制器不用功能码 03, 所以功能码 03 被用作读取点和返回值。

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。

2、 06, 写单个寄存器:

主机利用这条命令把单点数据保存到电表的存储器。子机也用这个功能码向主机返送信息。

2.1.3. 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、10H) 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

2、 与功能码 06 对应的数据区格式:

数据顺序	1	2
数据含义	起始地址	写入寄存器的数据
字节数	2	2

3、 与功能码 10 对应的数据区格式:

数据顺序	1	2	...	N
数据含义	起始地址	写入数据 1	...	写入数据 N
字节数	2	2	...	2

2.1.4. 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。

2.2. 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 1。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。

2.3. CRC-16 码的计算步骤

- 1、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。
- 2、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。
- 3、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)。
- 4、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。

- 5、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。
- 6、重复步骤 2 到步骤 5，进行下一个 8 位的处理。
- 7、最后得到的 CRC 寄存器即为 CRC 码，低字节在前，高字节在后。

2.4. 出错处理

不做任何返回。

级联地址设置：

第一步：通过 net，输入设备 IP 地址，账户/密码均为 admin，登录界面。

第二步：设备设置下的设备设置，工作模式选择从设备 1-9，分别对应 485 的地址 1-9

