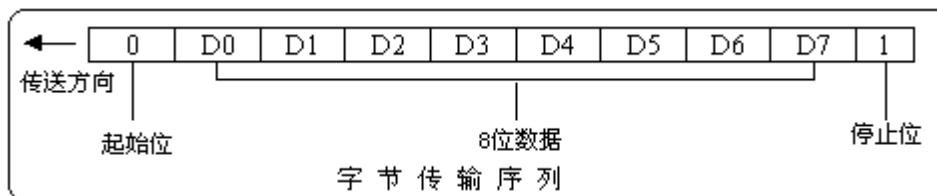


# OHR-C800系列电厂专用转速表 通信协议

本规约采用Modbus 规约RTU模式，可以方便地与多种组态软件相连接，其通讯驱动与Modicon Modbus\_RTU格式完全兼容。

## 1、字节格式：



每字节含8位二进制码，传输时加上一个起始位(0)，一个停止位(1)，共10位。其传输序列如上图所示，D0是字节的最低有效位，D7是字的最高有效位。先传低位，后传高位。

## 2、通讯数据格式

通讯时数据以字(WORD—2字节)的形式回送，回送的每个字中，高字节在前，低字节在后，如果2个字连续回送(如：浮点或长整形)，则高字在前，低字在后。

数据类型	寄存器数	字节数	说明
字节数据	1	1	
整形数据	1	2	一次送回，高字节在前，低字节在后
长整形数	2	4	分两个字回送，高字在前，低字在后
浮点数据			

## 3、帧格式

### 3.1 读取仪表继电器输出状态（功能码 01H）

3.1.1 上位机发送的帧格式：

顺序	代码	示例	说明
1	仪表地址	1	仪表的通讯地址（1-255之间）
2	01H	01H	功能码
3	起始寄存器地址高字节	01H	寄存器起始地址
4	起始寄存器地址低字节	00H	
5	寄存器个数高字节	00H	寄存器个数
6	寄存器个数低字节	02H	
7	CRC16 校验低字节	E9H	CRC 校验数据
	CRC16 校验高字节	7FH	

### 3.1.2 仪表回送的帧格式

顺序	代 码	说 明
1	仪表地址	仪表的通讯地址（1-255之间）
2	01H	功能码
3	回送数据域字节数(M)	
4	第一个数据	
.....	.....	
	第N个寄存器数据	
M+4	CRC 校验低字节	
M+5	CRC 校验高字节	

注：回送的数据包中每个继电器状态占用一位（1 = ON, 0 = OFF），第一个字节的最低位为寻址到的第一个继电器状态值，其余的依次向高位排列，无用位全部填零。

### 3.2 读取仪表保持寄存器内容（功能码 03H 或 04H）

#### 3.2.1 上位机发送的帧格式：

顺序	代 码	示例	说明
1	仪表地址	1	仪表的通讯地址（1-255之间）
2	03H 或 04H	03H	功能码
3	起始寄存器地址高字节	01H	寄存器起始地址
4	起始寄存器地址低字节	00H	
5	寄存器个数高字节	00H	寄存器个数
6	寄存器个数低字节	02H	
7	CRC16 校验低字节	C5H	CRC 校验数据
8	CRC16 校验高字节	F7H	

#### 3.2.2 仪表回送的帧格式（数据正常）

顺序	代 码	说 明
1	仪表地址	仪表的通讯地址（1-255之间）
2	03H 或 04H	功能码
3	回送数据域字节数(M)	
4	第一个寄存器数据	
.....	.....	
	第N个寄存器数据	
M+4	CRC 校验低字节	
M+5	CRC 校验高字节	

3.2.3如果起始寄存器地址或寄存器个数错误，仪表回送：

顺序	代 码	示 例	说 明
1	仪表地址	1	仪表的通讯地址（1-255之间）
2	83H 或 84H	83H	功能码——针对03H, 04H
3	02H	02H	错误代码
4	CRC 校验低字节	F1H	
5	CRC 校验高字节	C0H	

### 3.3 设置仪表寄存器内容（功能码 06H 或10H 或16H）

3.3.1.1 功能码06H写单路，将一个字（2 字节）数据写入仪表寄存器中，上位机发送的帧格式：

顺序	代 码	示 例	说 明
1	仪表地址	1	仪表的通讯地址（1-255 之间）
2	06H	06H	功能码
3	寄存器地址高字节	09H	寄存器地址0905H
4	寄存器地址低字节	05H	
5	写入数据高字节	00H	写入数据43H
6	写入数据低字节	43H	
7	CRC 校验低字节	A6H	CRC 校验数据A6DBH
8	CRC 校验高字节	DBH	

3.3.1.2 仪表回送：如果写入正确，则仪表回送相同的数据。

3.2.2.1 功能码 16H 或10H 写多路寄存器，上位机发送的帧格式：

顺序	代 码	示 例	说 明
1	仪表地址	1	仪表的通讯地址（1-255之间）
2	16H 或10H	10H	功能码
3	寄存器起始地址高字节	09H	寄存器地址0923H
4	寄存器起始地址低字节	03H	
5	寄存器个数高字节	00H	00H
6	寄存器个数低字节	02H	字节数据、整形数据：01H 浮点数据、长整形数：02H
7	字节数（M）	4	字节数据、整形数据：02H 浮点数、长整形数：04H
8	数据高字节	00H	设置的数据为两个整形数10、50
	数据次高字节	0AH	
	数据次低字节	00H	
	数据低字节	32H	
M+8	CRC校验低字节	3DH	CRC校验数据

M+9	CRC校验高字节	78H	
-----	----------	-----	--

### 3.2.2仪表回送：(写入成功)

顺序	代码	示例	说明
1	仪表地址	1	仪表的通讯地址（1-255之间）
2	16H 或10H	10H	功能码
3	起始地址高字节	09H	寄存器起始地址0923H
4	起始地址低字节	03H	
5	寄存器个数高字节	00H	寄存器个数2
6	寄存器个数低字节	02H	
7	CRC校验低字节	54H	CRC校验数据
8	CRC校验高字节	B2H	

### 3.2.3仪表回送：(寄存器地址或数据错误)

顺序	代码	说明
1	仪表地址	仪表的通讯地址（1-255之间）
2	96H 或90H 或86H	功能码——针对16H, 10H, 06H
3	03H	错误代码
4	CRC校验低字节	
5	CRC校验高字节	

注：以上介绍中CRC校验为16位，高字节在前，低字节在后。

**4、通讯波特率：**通讯波特率可以在9600、19200、38400之间选择。出厂时，仪表已设置某一波特率。

**5、仪表地址：**仪表地址可以在1-253之间选择。仪表出厂时，已设置某一地址。

**6、通讯功能码：**01H(读取继电器状态) 03H或04H(召测数据) 06H (10H 或16H) (数据设置)

### 7、通讯数据CRC 校验：

**7.1 校验多项式：** $X^{16}+X^{12}+X^5+1$

**7.2 CRC 检验码的计算例程**见附录。

**7.3 CRC 检验**从第1 字节开始至CRC 校验高字节前面的字节数据结束。

## 8、仪表数据寄存器地址

表1 寄存器地址表

寄存器地址	数据名称	量程系数	单位	数据格式	备注
继电器状态，功能码01读取。一个继电器使用一个地址，					
0000H	继电器J1状态	-	-	bit	0表示继电器开
0001H	继电器J2状态	-	-	bit	1表示继电器合
0002H	继电器J3状态	-	-	bit	
0003H	继电器J4状态	-	-	bit	
0004H	继电器J5状态	-	-	bit	
0005H	继电器J6状态	-	-	bit	
0006H	继电器J7状态	-	-	bit	
0007H	继电器J8状态	-	-	bit	
0008H	PT断线信号状态	-	-	bit	

000AH	装置故障信号状态	-	-	bit	
常规数据 功能码03H、04H读取；一次可最多读取123个连续字节；（特别注意：长整形数据和浮点型数据占两个寄存器，必须一次读出，若读取一半将返回错误信息，数据读取组帧时务必注意数据格式，比如，报警参数数据从A00H开始排列，A00H处是报警功能控制字（整形数据），A01H是电压报警上限参数（长整形数据），如果要连续读取这两个参数，寄存器个数应设置为3个，若只读取报警功能控制字，寄存器个数应设置为1个；若读电压报警上限参数，寄存器个数应设置为2个）					
测量实时值					
0100H	频率	100	赫兹	长整形	
0102H	转速	1	r/min	长整形	
0104H	百分比转速	100	%	长整形	
0106H	电压	10	伏特	长整形	
测量峰值					
0108H	频率峰值	100	赫兹	长整形	
010AH	转速峰值	1	r/min	长整形	
010CH	百分比转速峰值	100	%	长整形	
010EH	电压峰值	10	伏特	长整形	
版本					
0800H	仪表型号	---	---	ASSIC码	见表2
0805H	仪表版本号	---	---	ASSIC码	
080AH	通讯规约版本	---	---	ASSIC码	
时钟参数设置					
0900H	当前日期和时间	---	---	整形	见表3
参数设置A组 报警动作值					
0A00H	J1动作值设置	100	%	长整形	
0A02H	J2动作值设置	100	%	长整形	
0A04H	J3动作值设置	100	%	长整形	
0A06H	J4动作值设置	100	%	长整形	
0A08H	J5动作值设置	100	%	长整形	
0A0AH	J6动作值设置	100	%	长整形	
0A0CH	J7动作值设置	100	%	长整形	
0A0EH	J8动作值设置	100	%	长整形	
0A10H	继电器动作返回值	100	%	长整形	
0A12H	前4个继电器动作条件	100	%	长整形	高于该设置值之后前4点才动作
参数设置B组 报警动作方式					
0A20H	J1动作方式	1	-	整形	=0频率下降过程中动作 =0x5A频率上升过程中动作
0A21H	J2动作方式	1	-	整形	
0A22H	J3动作方式	1	-	整形	
0A23H	J4动作方式	1	-	整形	
0A24H	J5动作方式	1	-	整形	
0A25H	J6动作方式	1	-	整形	
0A26H	J7动作方式	1	-	整形	
0A27H	J8动作方式	1	-	整形	
0A28H	J5动作后锁定	1	-	整形	=0继电器动作后不锁定 =0x5A继电器动作后锁定
0A29H	J6动作后锁定	1	-	整形	
0A2AH	J7动作后锁定	1	-	整形	
0A2BH	J8动作后锁定	1	-	整形	
参数设置C组 报警动作延时					

0A30H	继电器1动作延时	100	秒	长整形	
0A32H	继电器2动作延时	100	秒	长整形	
0A34H	继电器3动作延时	100	秒	长整形	
0A36H	继电器4动作延时	100	秒	长整形	
0A38H	继电器5动作延时	100	秒	长整形	
0A3AH	继电器6动作延时	100	秒	长整形	
0A3CH	继电器7动作延时	100	秒	长整形	
0A3EH	继电器8动作延时	100	秒	长整形	
0A40H	停机之后继电器复归延时时间	100	秒	长整形	
参数设置D组 系统参数设置					
0A50H	通信地址	1	-	整形	
0A51H	波特率	1	-	整形	见表4
0A52H	校验方式	1	-	整形	
0A53H	额定频率	1	-	整形	
0A54H	脉冲个数	1	-	整形	
0A55H	额定转速	1	-	整形	
0A56H	极对数	1	-	整形	
0A57H	PT防跳变频率	1	-	整形	
0A58H	PT抗干扰	1	-	整形	
0A59H	脉冲防跳变频率	1	-	整形	
0A5AH	脉冲抗干扰	1	-	整形	
0A5BH	显示平均次数	1	-	整形	
参数设置E组 变送设置					
0B00H	变送输出选择	1		整形	0-3(见表5)
0B01H	变送上限电流对应值	1		整形	小于20mA
0B02H	变送下限电流对应值	1		整形	小于上限电流值
0B03H	变送上限			长整形	大于下限值
0B05H	变送下限			长整形	小于上限值
命令寄存器					
0C00H	写入不同命令，有不同作用，具体见下面的命令列表				
	命令列表				
	2000H				清除电能
	4000H				校表参数恢复默认值
	4003H				校表使能
	4005H				校表结束
	4013H				变送上限校准使能
	4015H				变送下限校准使能
401AH				变送校准结束	
校表参数寄存器:功能码03H读取；功能码06H写入；					
C000H	额定电压	1		整形	50-500V（读/写）
C001H	额定电流	1		整形	1-15A（读/写）
C002H	脉冲常数	1		整形	100-12000（读/写）
C003H	脉冲输出选择	1		整形	0: P 1: Q 2: S（读/写）
C004H	电压有效值Ureg	1		长整形	只读
C006H	电流有效值Ireg	1		长整形	只读
C008H	有功功率Preg	1		长整形	只读
C00AH	无功功率Qreg	1		长整形	只读

C00CH	视在功率Sreg	1		长整形	只读
C00EH	高频脉冲常数HFConst	1		整形	只写
C00FH	功率增益寄存器GP1	1		整形	只写
C010H	电压系数KUrms	1		长整形	只写
C012H	电流系数KIrms	1		长整形	只写
C014H	功率系数Kpqs	1		长整形	只写
C016H	相位补偿Gphs	1		整形	只写
C017H	有功偏置Poffset	1		整形	只写
C018H	无功偏置Qoffset	1		整形	只写
C019H	电压有效值偏置 URMSOFFSET	1		整形	只写
C01AH	电流有效值偏置 IRMSOFFSET	1		整形	只写
C01BH	变送上限	1000		整形	500-2000 (读/写)
C01CH	变送下限	1000		整形	500-2000 (读/写)
C01DH					

表2 版本格式

寄存器地址	高字节	低字节
寄存器800H-81DH	0	ASCII

表3 日期和时间的格式

寄存器地址	高字节	低字节
寄存器0	年 (BCD码)	月 (BCD码)
寄存器1	日 (BCD码)	时 (BCD码)
寄存器2	分 (BCD码)	秒 (BCD码)

表4 波特率表

数值	波特率	备注
0	1200bps	
1	2400bps	
2	4800bps	
3	9600bps	
4	19200bps	

表5 变送输出序号对应表

序号	输出参数	备注
0	无	变送输出关闭
1	频率	
2	转速	
3	百分比转速	
4	电压	
5	频率峰值	
6	转速峰值	
7	百分比转速峰值	
8	电压峰值	

```
//读取8个继电器  
01 01 00 00 00 08 3D CC  
//读取测量值  
01 03 01 00 00 0C 44 33  
//读取设置参数  
01 03 0A 00 00 20 47 CA
```



## 附录1 CRC 校验码的计算——算法

```
unsigned short CRC16(puchMsg, usDataLen)
unsigned char *puchMsg ; /* 要进行CRC校验的消息 */
unsigned short usDataLen ; /* 消息中字节数 */
{
    unsigned char uchCRChi = 0xFF ; /* 高CRC字节初始化 */
    unsigned char uchCRCLo = 0xFF ; /* 低CRC 字节初始化 */
    unsigned uIndex ; /* CRC循环中的索引 */
    while (usDataLen-- /* 传输消息缓冲区 */)
    {
        uIndex = uchCRChi ^ *puchMsgg++ ; /* 计算CRC */
        uchCRChi = uchCRCLo ^ auchCRChi[uIndex] ;
        uchCRCLo = auchCRCLo[uIndex] ;
    }
    return (uchCRChi << 8 | uchCRCLo) ;
}

/* CRC 高位字节值表 */
static unsigned char auchCRChi[] = {
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0,
    0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1, 0x81, 0x40,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40, 0x00, 0xC1, 0x81, 0x40,
    0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0, 0x80, 0x41, 0x00, 0xC1,
    0x81, 0x40, 0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41,
    0x00, 0xC1, 0x81, 0x40, 0x01, 0xC0, 0x80, 0x41, 0x01, 0xC0,
```

```

    0x80, 0x41, 0x00, 0xC1, 0x81, 0x40
} ;

/* CRC低位字节值表*/
static char auchCRCLo[] = {
    0x00, 0xC0, 0xC1, 0x01, 0xC3, 0x03, 0x02, 0xC2, 0xC6, 0x06,
    0x07, 0xC7, 0x05, 0xC5, 0xC4, 0x04, 0xCC, 0x0C, 0x0D, 0xCD,
    0x0F, 0xCF, 0xCE, 0x0E, 0x0A, 0xCA, 0xCB, 0x0B, 0xC9, 0x09,
    0x08, 0xC8, 0xD8, 0x18, 0x19, 0xD9, 0x1B, 0xDB, 0xDA, 0x1A,
    0x1E, 0xDE, 0xDF, 0x1F, 0xDD, 0x1D, 0x1C, 0xDC, 0x14, 0xD4,
    0xD5, 0x15, 0xD7, 0x17, 0x16, 0xD6, 0xD2, 0x12, 0x13, 0xD3,
    0x11, 0xD1, 0xD0, 0x10, 0xF0, 0x30, 0x31, 0xF1, 0x33, 0xF3,
    0xF2, 0x32, 0x36, 0xF6, 0xF7, 0x37, 0xF5, 0x35, 0x34, 0xF4,
    0x3C, 0xFC, 0xFD, 0x3D, 0xFF, 0x3F, 0x3E, 0xFE, 0xFA, 0x3A,
    0x3B, 0xFB, 0x39, 0xF9, 0xF8, 0x38, 0x28, 0xE8, 0xE9, 0x29,
    0xEB, 0x2B, 0x2A, 0xEA, 0xEE, 0x2E, 0x2F, 0xEF, 0x2D, 0xED,
    0xEC, 0x2C, 0xE4, 0x24, 0x25, 0xE5, 0x27, 0xE7, 0xE6, 0x26,
    0x22, 0xE2, 0xE3, 0x23, 0xE1, 0x21, 0x20, 0xE0, 0xA0, 0x60,
    0x61, 0xA1, 0x63, 0xA3, 0xA2, 0x62, 0x66, 0xA6, 0xA7, 0x67,
    0xA5, 0x65, 0x64, 0xA4, 0x6C, 0xAC, 0xAD, 0x6D, 0xAF, 0x6F,
    0x6E, 0xAE, 0xAA, 0x6A, 0x6B, 0xAB, 0x69, 0xA9, 0xA8, 0x68,
    0x78, 0xB8, 0xB9, 0x79, 0xBB, 0x7B, 0x7A, 0xBA, 0xBE, 0x7E,
    0x7F, 0xBF, 0x7D, 0xBD, 0xBC, 0x7C, 0xB4, 0x74, 0x75, 0xB5,
    0x77, 0xB7, 0xB6, 0x76, 0x72, 0xB2, 0xB3, 0x73, 0xB1, 0x71,
    0x70, 0xB0, 0x50, 0x90, 0x91, 0x51, 0x93, 0x53, 0x52, 0x92,
    0x96, 0x56, 0x57, 0x97, 0x55, 0x95, 0x94, 0x54, 0x9C, 0x5C,
    0x5D, 0x9D, 0x5F, 0x9F, 0x9E, 0x5E, 0x5A, 0x9A, 0x9B, 0x5B,
    0x99, 0x59, 0x58, 0x98, 0x88, 0x48, 0x49, 0x89, 0x4B, 0x8B,
    0x8A, 0x4A, 0x4E, 0x8E, 0x8F, 0x4F, 0x8D, 0x4D, 0x4C, 0x8C,
    0x44, 0x84, 0x85, 0x45, 0x87, 0x47, 0x46, 0x86, 0x82, 0x42,
    0x43, 0x83, 0x41, 0x81, 0x80, 0x40
};

```

附录 2、仪表通讯数据样例：( 以下所有数据皆为十六进制)