

7.2 附录二： RS485 通讯协议

● 通讯协议简介

AC80C-Y 系列变频器标配 RS485 通讯接口，并采用国际标准的 ModBus 通讯协议进行的主从通讯。用户可通过 PC/PLC、上位机、主站变频器等实现集中控制（设定变频器控制命令、运行频率、相关功能码参数的修改，变频器工作状态及故障信息的监控等），以适应特定的应用要求。

● 应用方式

1、AC80C-Y 系列变频器具备接入 RS485 总线的“单主多从”控制网络。主机使用广播命令（从机地址为 0）时从机无应答。

2、AC80C-Y 只提供 RS485 接口，异步半双工。若外界设备的通讯口为 RS232 时，需要另加 RS232/RS485 转换器。

3、ModBus 协议定义了串行通讯中异步传输的信息内容及使用格式，可分为 RTU 方式和 ASCII 方式。AC80C-Y 为 RTU（远程终端单元）模式。

● 通讯帧结构

通讯数据格式如下：

字节的组成：包括起始位、8 个数据位、校验位和停止位。

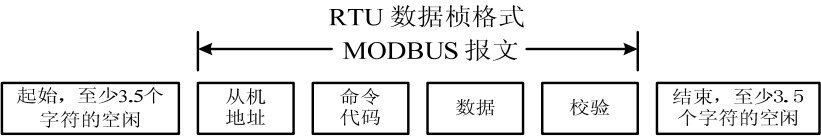
起 始 位	Bit1	Bit2	Bit3	Bit4	Bit5	Bit6	Bit7	Bit8	校验位	停 止 位
-------------	------	------	------	------	------	------	------	------	-----	-------------

一个帧的信息必须以一个连续的数据流进行传输，如果整个帧传输结束前超过 1.5 个字节以上的间隔时间，接收设备将清除这些不完整的信息，并错误认为随后一个字节是新一帧的地址域部分。同样的，如果一个新帧的开始与前一个帧的间隔时间小于 3.5 个字节时间，接收设备将认为它是前一帧的继续，由于帧的错乱，最终 CRC 校验值不正确，导致通讯错误。

RTU 帧的标准结构：

帧头	3.5 个字节的传输时间
从机地址	通讯地址： 0~247（十进制）（0 为广播地址）
命令代码	03H：读从机参数 06H：写从机参数 08H：回路自检测
数据区	参数地址，参数个数，参数值等
CRC CHK 低位	检测值：16 位 CRC 校验值
CRC CHK 高位	
帧尾	3.5 个字节的传输时间

在 RTU 模式中，新的一帧以至少 3.5 个字节的传输时间停顿间隔作为开始。紧接着传输的数据域依次为：从机地址、操作命令代码、数据和 CRC 校验字，每个域传输字节都是十六进制的 0...9，A...F。网络设备不断侦测网络总线，包括停顿间隔时间内。当接收到第一个域（地址信息），每个网络设备都对该字节进行解码以判断是否是发往自己的。在最后一个字节的传输完成，又以一个至少 3.5 个字节的传输时间间隔来表明本帧的结束，在此以后，一个新的消息可以开始。



● 命令代码及通讯数据描述

命令代码：03H，读取 N 个字(Word)，最多可以连续读取 5 个字。

例如：从机地址为 01H 的变频器，内存启始地址为 2100H([C-00])，读取连续 3 个字，则该帧的结构描述如下：

RTU 主机命令信息

START	3.5 个字节的传输时间
从机地址	01H
命令代码	03H
启始地址高位	21H
启始地址低位	00H
数据个数高位	00H
数据个数低位	03H
CRC CHK 低位	0FH
CRC CHK 高位	F7H
END	3.5 个字节的传输时间

RTU 从机回应信息（正常时）

START	3.5 个字节的传输时间
从机地址	01H
命令代码	03H
字节个数低位	06H
数据地址 2100H 高位	13H
数据地址 2100H 低位	88H
数据地址 2101H 高位	00H
数据地址 2101H 低位	00H
数据地址 2102H 高位	00H
数据地址 2102H 低位	00H
CRC CHK 低位	90H
CRC CHK 高位	A6H
END	3.5 个字节的传输时间

RTU 从机回应信息（异常时）

START	3.5 个字节的传输时间
从机地址	01H
命令代码	83H
错误代码	04H
CRC CHK 低位	40H
CRC CHK 高位	F3H
END	3.5 个字节的传输时间

命令代码：06H，写一个字(Word)

功能：将一个字数据写入被指定的数据地址中，可用于修改变频器参数值。

例如：将 5000（1388H）写到从机地址 1 变频器的 3000H 地址处。则该帧的结构描述如下：

RTU 主机命令信息

START	3.5 个字节的传输时间
从机地址	01H
命令代码	06H
写数据地址高位	30H
写数据地址低位	00H
数据内容高位	13H
数据内容低位	88H
CRC CHK 低位	8BH
CRC CHK 高位	9CH
END	3.5 个字节的传输时间

RTU 从机回应信息（正常时）

START	3.5 个字节的传输时间
从机地址	01H
命令代码	06H
写数据地址高位	30H
写数据地址低位	00H
数据内容高位	13H
数据内容低位	88H
CRC CHK 低位	8BH
CRC CHK 高位	9CH
END	3.5 个字节的传输时间

RTU 从机回应信息（异常时）

START	3.5 个字节的传输时间
从机地址	01H
命令代码	86H
错误代码	01H
CRC CHK 低位	83H
CRC CHK 高位	A0H
END	3.5 个字节的传输时间

命令代码：08H，回路自检测

功能：送回与主机指令信息相同的从机响应信息，用于检测主机与从机之间的信号传输是否正常。
其中检测代码及数据可任意设置。

RTU 主机命令信息

START	3.5 个字节的传输时间
从机地址	01H
命令代码	08H
检测代码高位	00H
检测代码低位	00H
数据高位	13H
数据低位	88H
CRC CHK 低位	EDH
CRC CHK 高位	5DH
END	3.5 个字节的传输时间

RTU 从机回应信息（正常时）

START	3.5 个字节的传输时间
从机地址	01H
命令代码	08H
检测代码高位	00H
检测代码低位	00H
数据高位	13H
数据低位	88H
CRC CHK 低位	EDH
CRC CHK 高位	5DH
END	3.5 个字节的传输时间

RTU 从机回应信息（异常时）

START	3.5 个字节的传输时间
从机地址	01H
命令代码	88H
错误代码	03H
CRC CHK 低位	06H
CRC CHK 高位	01H
END	3.5 个字节的传输时间

通讯帧错误校验方式

标准的 Modbus 串行网络采用两种错误检测方法。奇偶校验用于对每个字符的校验，CRC 检测用于对一帧数据的校验。

1、奇偶校验

用户可以配置控制器是奇或偶校验，或无校验。这将决定了每个字符中的奇偶校验位是如何设置的。

如果指定了奇或偶校验，“1”的位数将算到每个字符的位数中（ASCII 模式 7 个数据位，RTU

中 8 个数据位)。例如 RTU 字符帧中包含以下 8 个数据位： 1 1 0 0 0 1 0 1

整个“1”的数目是 4 个。如果便用了偶校验，帧的奇偶校验位将是 0，使得整个“1”的个数仍是 4 个。如果便用了奇校验，帧的奇偶校验位将是 1，使得整个“1”的个数是 5 个。

如果没有指定奇偶校验位，传输时就没有校验位，也不进行校验检测。代替一附加的停止位填充至要传输的字符帧中。

2、CRC-16（循环冗余校验）

使用 RTU 帧格式，帧包括了基于 CRC 方法计算的帧错误检测域。CRC 域检测了整个帧的内容。CRC 域是两个字节，包含 16 位的二进制值。它由传输设备计算后加入到帧中。接收设备重新计算收到帧的 CRC，并与接收到的 CRC 域中的值比较，如果两个 CRC 值不相等，则说明传输有错误。

CRC 是先存入 0xFFFF，然后调用一个过程将帧中连续的 6 个以上字节与当前寄存器中的值进行处理。仅每个字符中的 8Bit 数据对 CRC 有效，起始位和停止位以及奇偶校验位均无效。

CRC 产生过程中，每个 8 位字符都单独和寄存器内容相异或 (XOR)，结果向最低有效位方向移动，最高有效位以 0 填充。LSB 被提取出来检测，如果 LSB 为 1，寄存器单独和预置的值相异或，如果 LSB 为 0，则不进行。整个过程要重复 8 次。在最后一位（第 8 位）完成后，下一个 8 位字节又单独和寄存器的当前值相异或。最终寄存器中的值，是帧中所有的字节都执行之后的 CRC 值。

CRC 的这种计算方法，采用的是国际标准的 CRC 校验法则，用户在编辑 CRC 算法时，可以参考相关标准的 CRC 算法，编写出真正符合要求的 CRC 计算程序。

现在提供一个 CRC 计算的简单函数给用户参考（用 C 语言编程）：

```
unsigned int crc_chk_value(unsigned char *data_value,unsigned char length)
{
    unsigned int crc_value=0xFFFF;
    int i;
    while(length--)
    {
        crc_value^=*data_value++;
        for(i=0;i<8;i++)
        {
            if(crc_value&0x0001)
            {
                crc_value=(crc_value>>1)^0xa001;
            }
            else
            {
                crc_value=crc_value>>1;
            }
        }
    }
    return(crc_value);
}
```

● 通讯数据地址的定义

该部分是通讯数据的地址定义，用于控制变频器的运行、获取变频器状态信息及变频器相关功能参数设定等。

(1) AC80C-Y 系列功能参数地址表示规则

以变频器功能参数序号为寄存器地址，分为高字节与低字节两部分。高字节表示功能参数所在组序号，低字节表示功能参数的组内序号，需转换成十六进制。

地址域高位字节定义：

参数组码号	本组参数首地址
F0 基本参数组	0x0000（不存入 EEPROM） 0x1000（存入 EEPROM）
F1 运行控制参数组	0x0100（不存入 EEPROM） 0x1100（存入 EEPROM）
F2 开关量端子参数组	0x0200（不存入 EEPROM） 0x1200（存入 EEPROM）
F3 模拟量端子参数组	0x0300（不存入 EEPROM） 0x1300（存入 EEPROM）
F4 键盘及显示参数组	0x0400（不存入 EEPROM） 0x1400（存入 EEPROM）
F5 电机参数组	0x0500（不存入 EEPROM） 0x1500（存入 EEPROM）
F6 矢量控制参数组	0x0600（不存入 EEPROM） 0x1600（存入 EEPROM）
F7 保留	0x0700（不存入 EEPROM） 0x1700（存入 EEPROM）
F8 V/F 控制参数组	0x0800（不存入 EEPROM） 0x1800（存入 EEPROM）
F9 保留	0x0900（不存入 EEPROM） 0x1900（存入 EEPROM）
FA 故障及保护参数组	0x0A00（不存入 EEPROM） 0x1A00（存入 EEPROM）
Fb 过程 PID 控制参数组	0x0B00（不存入 EEPROM） 0x1B00（存入 EEPROM）
FC 多段速、PLC 功能与摆频参数组	0x0C00（不存入 EEPROM） 0x1C00（存入 EEPROM）
Fd 通讯控制功能参数组	0x0D00（不存入 EEPROM） 0x1D00（存入 EEPROM）
FE 保留	0x0E00（不存入 EEPROM） 0x1E00（存入 EEPROM）
FF 保留	0x0F00（不存入 EEPROM） 0x1F00（存入 EEPROM）
C 监控参数组	0x2100
通讯控制参数组	0x3000 或 0x2000

注意：由于通讯存在频繁改写参数值的可能，如果 EEPROM 频繁被存储会减少使用寿命。对于用户而言，有些功能码参数在通讯的模式下，无须存储，只需更改片内 RAM 中的值就可以满足使用要求。AC80C-Y 通讯协议规定当使用写命令（06H）时，若功能码参数地址域最高位为 0，只写入变频器 RAM 中，掉电不存储，若功能码参数地址域高半字节为 1，写入 EEPROM 中，即掉电存储。

例如改写功能参数 **[F0.14]**，不存入 EEPROM 中，地址表示为 000EH，存入 EEPROM 中，地址表示为 100EH。

（2）通讯控制参数组地址说明：

功能说明	地址定义	数据意义说明			R/W 特性
通讯给定频率	0x3000 或 0x2000	0～32000 对应 0. 00Hz～320. 00Hz			W/R
通讯命令设定	0x3001 或 0x2001	0000H: 无命令			W
		0001H: 正转运行			
		0002H: 反转运行			
		0003H: 正转点动			
		0004H: 反转点动			
		0005H: 减速停机			
		0006H: 自由停机			
		0007H: 故障复位			
变频器状态	0x3002 或 0x2002	Bit0	0: 停机状态	1: 运行状态	R
		Bit1	0: 非加速状态	1: 加速状态	
		Bit2	0: 非减速状态	1: 减速状态	
		Bit3	0: 正向	1: 反向	
		Bit4	0: 变频器正常	1: 变频器出现故障	
		Bit5	0: GPRS 正常	1: GPRS 锁机	
变频器故障码	0x3003 或 0x2003	变频器当前故障代码（见故障代码表）			R/W
通讯给定上限频率	0x3004 或 0x2004	0～32000 对应 0. 00Hz～320. 00Hz			W
通讯给定 PID 设定值	0x3008 或 0x2008	0～1000 对应 0.0～100. 0%			W
通讯给定 PID 反馈值	0x3009 或 0x2009	0～1000 对应 0.0～100. 0%			W

从机回应异常信息的错误代码含义：

错误代码	说明
1	命令代码错误
2	保留
3	CRC 校验错误
4	非法地址
5	非法数据
6	运行中参数不能更改
7	保留
8	变频器忙（EEPROM 正在存储中）
9	参数值超限
10	保留参数无法更改
11	读取参数字节数有误