

## 附录 3 RS485 通讯协议

GT300 系列变频器在 RS485 通讯控制方面采用了流行的 MODBUS 通讯协议，在使用 RS485 通讯前必须手工设定变频器的地址、通讯波特率、数据格式，并且在通讯过程中这些参数不可修改。

MODBUS 通讯协议采用了两种编码格式：ASCII (American Standard Code for Information Interchange) 或 RTU (Remote Terminal Unit)。ASCII 编码是将要传送的数据转换成对应的 ASCII 后再传送，而 RTU 则是资料直接传送，不再经过转换。

ASCII 编码格式下，每个 Byte 数据是由两个 ASCII 码组成，例如：0x1F，ASCII 的表示方式为 ‘1F’，分别由 ‘1’ (31Hex)、‘F’ (46Hex) 组合而成。以下是 0-9, A-F 的 ASCII 码

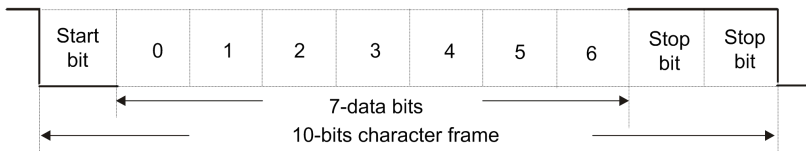
字符	‘0’	‘1’	‘2’	‘3’	‘4’	‘5’	‘6’	‘7’
ASCII code	30H	31H	32H	33H	34H	35H	36H	37H
字符	‘8’	‘9’	‘A’	‘B’	‘C’	‘D’	‘E’	‘F’
ASCII code	38H	39H	41H	42H	43H	44H	45H	46H

RTU 编码格式下，每个 Byte 数据是由两个 4-bit 十六进制字符组成，例如：0x1F RTU 表示方式为 ‘1FH’。

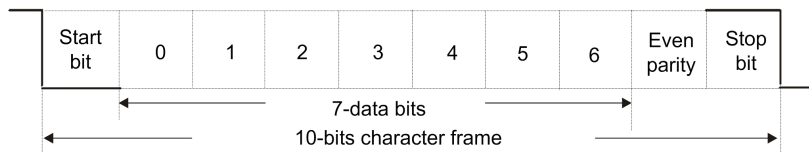
### 字符结构

10-bit 字符框（用于 7-bit 字符）：

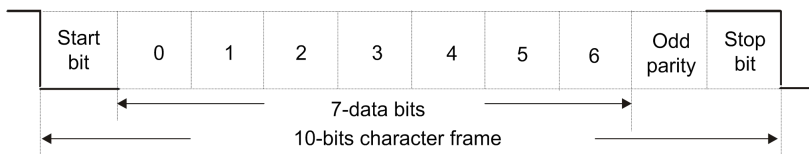
(7, N, 2)



(7, E, 1)

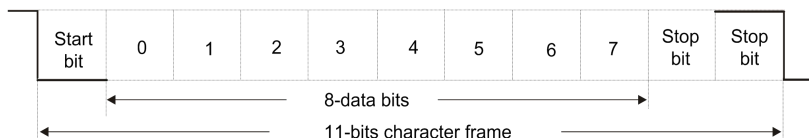


(7, O, 1)

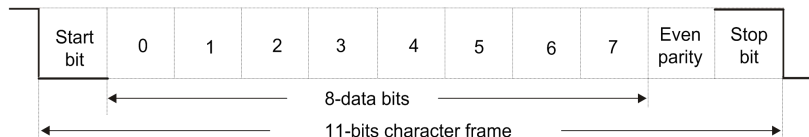


11-bit 字符框（用于8-bit 字符）：

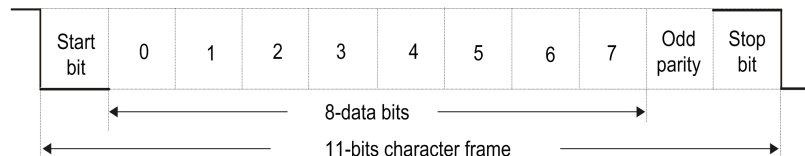
(8, N, 2)



(8, E, 1)



(8, O, 1)



## 通讯资料结构

通讯资料格式框:

ASCII 模式:

STX	开始字符 ‘:’ (3AH)
ADR 1	通讯地址: 8-bit 地址包含了 2 个 ASCII 码
ADR 0	
CMD 1	指令码: 8-bit 指令包含了 2 个 ASCII 码
CMD 0	
DATA (n-1)	数据内容: n×8-bit 资料包含了 2n 个 ASCII 码 n≤16, 最多 32个 ASCII 码
.....	
DATA 0	
LRC CHK 1	校验值: 8-bit 校验值包含了 2 个 ASCII 码
LRC CHK 0	
END 1	结束字符: END1= CR (0DH), END0= LF(0AH)
END 0	

RTU 模式:

START	超过 10 ms 静止时间或3.5个字节传输时间
ADR	通讯地址: 8-bit 地址
CMD	指令码: 8-bit 指令
DATA (n-1)	数据内容: N×8-bit 资料, n≤32
.....	
DATA 0	
CRC CHK Low	CRC校验值: 16-bit 校验值由 2 个 8-bit 字符组成
CRC CHK High	
END	超过 10 ms 静止时间或3.5个字节传输时间

**ADR (通讯地址)**

合法的通讯地址范围在 1 到 247 之间。通讯地址为 0 表示对所有变频器进行广播，在此情况下变频器将不会响应任何信息给主机。

例如：对通讯地址为16(十进制)之变频器进行通讯：

ASCII 模式：(ADR 1, ADR 0) = '1','0' => '1'=31H, '0'=30H

RTU 模式：(ADR) = 10H

**功能码 (Function) 及 数据内容 (Data Characters)**

03: 读出变频器寄存器内容

06: 写入一个WORD 至变频器寄存器

08: 回路侦测

10: 写入多个WORD 至变频器寄存器

指令码：03H，读取变频器寄存器内容。

例如：从地址01H 变频器的起始地址2102H 连续读取2个字。

RTU模式

询问讯息字符串格式：

Address	01H
Function	03H
Starting address	21H
	02H
Number of data (count by word)	00H
	02H
CRC CHK Low	6FH
CRC CHK High	F7H

回应讯息字符串格式：

Address	01H
Function	03H
Number of data (count by byte)	04H
Content of data address 2102H	17H
	70H
Content of data address 2103H	00H
	00H
CRC CHK Low	FEH
CRC CHK High	5CH

## ASCII模式

询问讯息字符串格式:

STX	‘.’
Address	‘0’
	‘1’
Function	‘0’
	‘3’
Starting address	‘2’
	‘1’
	‘0’
	‘2’
Number of data (count by word)	‘0’
	‘0’
	‘0’
	‘2’
LRC Check	‘D’
	‘7’
END	CR
	LF

回应讯息字符串格式:

STX	‘.’
Address	‘0’
	‘1’
Function	‘0’
	‘3’
Number of data (count by byte)	‘0’
	‘4’
Content of starting address 2102H	‘1’
	‘7’
	‘7’
	‘0’
Content of address 2103H	‘0’
	‘0’
	‘0’
	‘0’
LRC Check	‘7’
	‘1’
END	CR
	LF

指令码: 06H, 写一个字至变频器寄存器

例如: 将6000(1770H)写到地址为01H 变频器的0100H 地址

## RTU模式

询问讯息字符串格式:

Address	01H
Function	06H
Data address	01H
	00H
Data content	17H
	70H
CRC CHK Low	86H
CRC CHK High	22H

回应讯息字符串格式:

Address	01H
Function	06H
Data address	01H
	00H
Data content	17H
	70H
CRC CHK Low	86H
CRC CHK High	22H

## ASCII模式

询问讯息字符串格式:

STX	‘.’
Address	‘0’
	‘1’
Function	‘0’
	‘6’
Data address	‘0’
	‘1’
	‘0’
	‘0’
Data content	‘1’
	‘7’
	‘7’
	‘0’
LRC Check	‘7’
	‘1’
END	CR
	LF

回应讯息字符串格式:

STX	‘.’
Address	‘0’
	‘1’
Function	‘0’
	‘6’
Data address	‘0’
	‘1’
	‘0’
	‘0’
Data content	‘1’
	‘7’
	‘7’
	‘0’
LRC Check	‘7’
	‘1’
END	CR
	LF

指令码: 08H, 通讯回路测试

此命令用来测试主机（通常为PC或PLC）与变频器间通讯是否正常，变频器将收到的资料内容原封不动的回送给主机。

## RTU模式

询问讯息字符串格式:

Address	01H
Function	08H
Data address	00H
	00H
Data content	17H
	70H
CRC CHK Low	EEH
CRC CHK High	1FH

回应讯息字符串格式:

Address	01H
Function	08H
Data address	00H
	00H
Data content	17H
	70H
CRC CHK Low	EEH
CRC CHK High	1FH

## ASCII模式

询问讯息字符串格式:

STX	‘.’
Address	‘0’
	‘1’
Function	‘0’
	‘8’
Data address	‘0’
	‘0’
	‘0’
	‘0’
Data content	‘1’
	‘7’
	‘7’
	‘0’
LRC Check	‘7’
	‘0’
END	CR
	LF

回应讯息字符串格式:

STX	‘.’
Address	‘0’
	‘1’
Function	‘0’
	‘8’
Data address	‘0’
	‘0’
	‘0’
	‘0’
Data content	‘1’
	‘7’
	‘7’
	‘0’
LRC Check	‘7’
	‘0’
END	CR
	LF

指令码：10H，写多个字至变频器寄存器

例如：将5000(1338H)、4000(0FA0H)写到地址为01H 变频器的0500H 和0501H 地址中

#### RTU模式

询问讯息字符串格式：

Address	01H
Function	10H
Data address	05H
	00H
Number of data (count by word)	00H
	02H
Number of data (count by byte)	04H
The first data content	13H
	88H
The second data content	0FH
	A0H
CRC CHK Low	4DH
CRC CHK High	D9H

回应讯息字符串格式：

Address	01H
Function	10H
Starting data address	05H
	00H
Number of data (count by word)	00H
	02H
CRC CHK Low	41H
CRC CHK High	04H



## ASCII模式

询问讯息字符串格式:

STX	'0'
Address	'0'
	'1'
Function	'1'
	'0'
Starting data address	'0'
	'5'
	'0'
	'0'
Number of data (count by word)	'0'
	'0'
	'0'
	'2'
Number of data (count by byte)	'0'
	'4'
The first data content	'1'
	'3'
	'8'
	'8'
The second data content	'0'
	'F'
	'A'
LRC Check	'0'
	'A'
END	CR
	LF

回应讯息字符串格式:

STX	'0'
Address	'0'
	'1'
Function	'1'
	'0'
Data address	'0'
	'5'
	'0'
	'0'
Number of data (count by word)	'0'
	'0'
	'0'
	'2'
LRC Check	'E'
	'8'
END	CR
	LF

**CHK (check sum: 校验值)****ASCII 模式:**

ASCII 模式采用LRC (Longitudinal Redundancy Check) 校验值。LRC 校验值是将ADR1 至最后一个数据内容加总, 得到结果以256 为单位, 超出部分去除(例如得到结果为十六进制128H 则只取28H), 然后计算二次补码后得到结果即为LRC 校验值。

例如第一范例询问信息的校验码：01H+03H+21H+02H+00H+02H=29H，然后取2的补数=D7H。

#### RTU 模式：

RTU 模式采用CRC (Cyclical Redundancy Check) 校验值，CRC 校验值以下列步骤计算：

步骤 1：加载一个内容为FFFFH 的16-bit 缓存器 (称为CRC缓存器)。

步骤 2：将指令信息第一个字节与16-bit CRC 缓存器的低字节进行 XOR 运算，并将结果存回CRC 缓存器。

步骤 3：将CRC缓存器之内容向右移1bit，最左位bit填入0。

步骤 4：检查右移的值，若为0将步骤3的新值存回CRC 缓存器；否则将CRC缓存器与A001H 进行 XOR 运算，并将结果存入CRC 缓存器。

步骤 5：重复步骤 3 及步骤 4，直到CRC缓存器内容已被右移了8 bit。此时，该字节已完成处理。

步骤 6：对指令信息下一个字节重复步骤2 至步骤5，直到所有字节皆完成处理，CRC缓存器的最后内容即是CRC 值。当在指令信息中传递CRC值时，低字节须与高位元组交换顺序，即低字节将先被传送。

#### 范例

下例以C 语言产生CRC 值。此函数(function)需要两个参数：

Unsigned char\* data ← 指向消息缓冲区的指针

Unsigned char length ← 消息缓冲区中字节数目

此函数将传回unsigned int 型CRC校验值。

```
unsigned int crc_chk(unsigned char* data, unsigned char length)
{
    int j;
    unsigned int reg_crc=0xFFFF;
    while(length--)
    {
```

```
reg_crc ^= *data++;  
for(j=0;j<8;j++)  
{  
    if(reg_crc & 0x01) /* LSB(b0)=1 */  
    {  
        reg_crc=(reg_crc>>1) ^ 0xA001;  
    }  
    else  
    {  
        reg_crc=reg_crc >>1;  
    }  
}  
}  
return reg_crc;  
}
```

### 通信数据地址的定义

该部分是通信数据的地址定义，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。

功能码参数地址表示规则：

以功能码序号为参数对应寄存器地址，但要转换成十六进制（组别参数已经是十六进制数无需转换），如 FA.12 则用十六进制表示该功能码地址为 0A0CH。

另外，由于 EEPROM 频繁被存储，会减少 EEPROM 的使用寿命，对于用户而言，有些功能码在通讯的模式下，无需存储，只需更改片内 RAM 中的值就可以满足使用要求。要实现该功能，只要把对应的功能码地址最高位由 0 变成 1 就可以实现。如：功能码 F0.07 不存储到 EEPROM 中，只修改 RAM 中的值，可将地址设置为 8007H；该地址只能用作写片内 RAM 时使用，不能用做读的功能，如做读为无效地址。

## 通信协议的参数字址定义:

定 义	参数字址	功 能 说 明		属性
变频器设定参数	Fx.xxH	x.xx 表示参数号码。例如: F5.05由 0505H 来表示。		
控制指令	1000H	0001H	正转运行	W/R
		0002H	反转运行	
		0003H	正转点动	
		0004H	反转点动	
		0005H	停机	
		0006H	自由停机	
		0007H	故障复位	
		0008H	点动停止	
监视状态	1001H	0001H	正转运行中	R
		0002H	反转运行中	
		0003H	待机中	
		0004H	故障中	
通讯设定值	2000H	通信设定值范围(-10000~10000) <b>注意:</b> 通信设定值是相对值的百分数(-100.00%~100.00%), 可做通信写操作。当作为频率源设定时, 相对的是最大频率(F0.04)的百分数; 当作为转矩给定时, 相对的是转矩上限(F3.07)的百分数。当作为PID给定或者反馈时, 相对的是PID的百分数。		W/R
监视参数	3000H	运行频率		R
	3001H	设定频率		R
	3002H	输出电流		R
	3003H	输出电压		R
	3004H	输出转速		R
	3005H	输出功率		R

定 义	参数字址	功 能 说 明	属性
监视参数	3006H	输出转矩	R
	3007H	直流母线电压	R
	3008H	PID设定值	R
	3009H	PID反馈值	R
	300AH	输入端子状态	R
	300BH	输出端子状态	R
	300CH	模拟量AVI值	R
	300DH	模拟量ACI值	R
变频器故障地址	5000H	如附表3-1	R
通讯故障地址	5001H	如附表3-2	R

5000H 中的数据与实际故障对照表 附表 3-1

数据	故障类型	数据	故障类型
00H	无故障	0CH	电机过载 (OL1)
01H	加速过电流 (ocA)	0DH	变频器过载 (OL2)
02H	减速过电流 (ocd)	0EH	外部故障 (EF)
03H	恒速过电流 (ocn)	0FH	RS485 通讯故障 (CE-1)
04H	加速过电压 (ovA)	10H	保留
05H	减速过电压 (ovd)	11H	电流检测故障 (itE)
06H	恒速过电压 (ovn)	12H	面板通讯故障 (CE-4)
07H	停机过电压 (ovS)	13H	电机自学习故障 (tE)
08H	母线欠压故障 (Lv)	14H	EEPROM 操作故障 (EEP)
09H	输入侧缺相 (LP)	15H	PID 断线故障 (PIDE)
0AH	输出短路 (SC)	16H~1BH	保留
0BH	过热故障 (OH1)	1CH	输出侧缺相 (SP0)

错误通讯时的额外回应

当变频器做通讯连接时，如果产生错误，变频器会回应错误码且将命令码的最高位（bit7）设为1（即Function code and 80H）回应给主机，让主机知道有错误产生。

ASCII模式

STX	'.'
Address	'0'
	'1'
Function	'8'
	'6'
Fault address	'5'
	'0'
	'0'
	'1'
Fault code	'0'
	'1'
LRC Check	'2'
	'7'
END	CR
	LF

RTU模式

Address	01H
Function	86H
Fault address	50H
	01H
Fault code	01H
CRC CHK Low	F0H
CRC CHK High	C9H

5001H 中的数据与实际故障对照表 附表 3-2

通讯故障地址	5001H	00H	无故障
		01H	命令码错误
		02H	非法地址
		03H	非法数据
		04H~05H	保留
		06H	变频器忙
		07H~09H	保留
		10H	密码错误
		11H	校验错误
		12H	参数更改无效
		13H	系统被锁定
		14H	数据个数非法