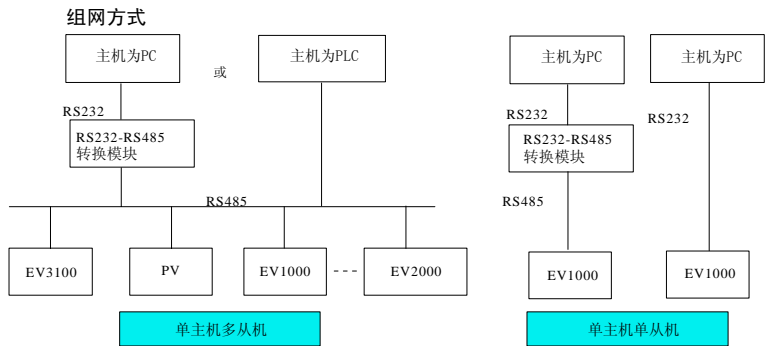


附录二 通讯协议



附图 1 变频器组网方式示意图

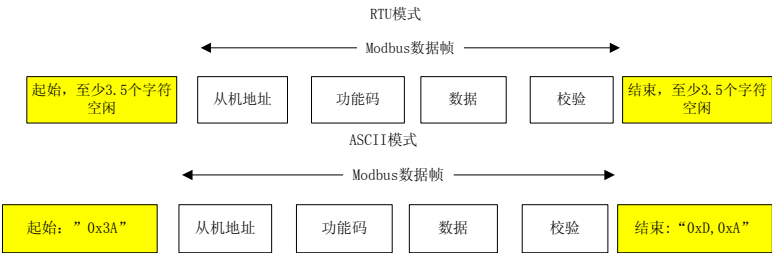
接口通讯方式

RS485 接口：异步，半双工。默认：8-N-2，19200bps。参数设置见 FF 组说明。

- 1. 变频器通讯协议为 Modbus 协议，除了支持常用的寄存器读写外，还扩充了部分命令对变频器功能码进行管理。
- 2. 变频器为从机，主从式点对点通信。主机使用广播地址发送命令时，从机不应答。
- 3. 在多机通讯或者长距离的情况下，建议将主站的信号地和变频器的信号地“GND”连接起来，以提高通讯的抗扰性。

协议格式

Modbus 协议同时支持 RTU 模式和 ASCII 模式，对应的帧格式如下：



附图 2：协议格式

Modbus 采用“Big Endian”编码方式，先发送高位字节，然后是低位字节。

RTU 方式：在 RTU 方式下，帧之间的空闲时间取功能码设定和 Modbus 内部约定值中的较大值。Modbus 内部约定的

最小帧间空闲如下：上个帧尾和下个帧头通过总线空闲时间不小于 3.5 个字节时间。数据校验采用 CRC-16，整个信息参与校验，校验和的高低字节需要交换后发送。具体的 CRC 校验请参考协议后面的示例。值得注意的是，帧间保持至少 3.5 个字符的总线空闲即可，帧之间的总线空闲不需要累加起始和结束空闲。

下面是读取 1 号机的 002 参数的请求帧：

地址	功能码	寄存器地址		读取字数		校验和	
0x01	0x03	0x00	0x02	0x00	0x01	0x25	0xCA

下面是为 1 号机的响应帧：

地址	功能码	应答字节数	寄存器内容		校验和	
0x01	0x03	0x02	0x13	0x88	0xB5	0x12

在 ASCII 方式下，帧头为“0x3A”，帧尾缺省为“0x0D”“0x0A”，帧尾还可由用户配置设定。在 ASCII 方式下，除了帧头和帧尾之外，其余的数据字节全部以 ASCII 码方式发送，先发送高 4 位元组，然后发送低 4 位元组。ASCII 方式下数据为 7 位长度。对于“A”~“F”，采用其大写字母的 ASCII 码。此时数据采用 LRC 校验，校验涵盖从从机地址到数据的信息部分。校验和等于所有参与校验数据的字符和（舍弃进位位）的补码。

ASCII 方式 Modbus 数据帧举例如下：

写入 4000（0xFA0）到从机 1 的内部寄存器 002 命令帧格式如下表：

LRC 校验=（01+06+00+02+0x0F+0xA0）的补码=0x48

	帧头	地址		功能码		寄存器地址				写入内容				LRC 校验		帧尾	
字符	:	0	1	0	6	0	0	0	2	0	F	A	0	4	8	CR	LF
ASCII	3A	30	31	30	36	30	30	30	32	30	46	41	30	34	38	0D	0A

变频器通过功能码可以设置不同的应答延时以适应各种主站的具体应用需要。对于 RTU 模式，实际的应答延时不小于 3.5 个字符间隔；对于 ASCII 模式，实际的应答延时不小于 1ms。

### 协议功能

Modbus 最主要的功能是读写参数，不同的功能码决定不同的操作请求。变频器 Modbus 协议支持以下功能码操作：

功能码	功能码意义
0x03	读取变频器功能码参数和运行状态参数
0x06	改写单个变频器功能码或者控制参数，掉电之后不保存
0x08	线路诊断
0x10	改写多个变频器功能码或者控制参数，掉电之后不保存
0x41	改写单个变频器功能码或者控制参数，掉电之后保存
0x42	功能码管理

变频器的功能码参数、控制参数和状态参数都映射为 Modbus 的读写寄存器。功能码参数的读写特性和范围遵循变频器用户手册的说明。变频器功能码的组号映射为寄存器地址的高字节，组内索引映射为寄存器地址的低字节。变频器的控制参数和状态参数均虚拟为变频器功能码组。功能码组号与其映射的寄存器地址高字节的对应关系如下：

F0 组：0x00；F1 组：0x01；F2 组：0x02；F3 组：0x03；F4 组：0x04；F5 组：0x05；F6 组：0x06；F7 组：0x07；F8 组：0x08；F9 组：0x09；FA 组：0x0A；Fb 组：0x0B；FC 组：0x0C；Fd 组：0x0D；FE 组：0x0E；FF 组：0x0F；FH 组：0x10；FL 组：0x11；Fn 组：0x12；FP 组：0x13；FU 组：0x14；变频器控制参数组：0x32；变频器状态参数组：0x33。

例如变频器功能码参数 F3.02 的寄存器地址为 0x302，变频器功能码参数 FF.01 的寄存器地址为 0xF01。

前面已经介绍了整个数据帧的格式，下面将集中介绍 Modbus 协议功能码和数据部分的格式和意义，也就是上述数据帧格式中的“功能码”和“数据”部分的内容，参见附图 2。这两部分组成了 Modbus 的应用层协议数据单元，下面提到的应用层协议数据单元就是指这两部分。以下对帧格式的说明以 RTU 模式为例，**ASCII 模式应用层协议数据单元的长度需加倍。**

读取变频器参数的应用层协议数据单元如下：

请求帧格式：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x03
起始寄存器地址	2	0x0000~0xFFFF
寄存器数目	2	0x0000~0x0004

应答帧格式：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x03
读取字节数	1	2*寄存器数目
读取内容	2*寄存器数目	

如果操作请求失败，应答为错误代码和异常代码。错误代码等于（功能码+0x80），异常代码标示错误原因。异常代码列举如下：

表 1 异常代码及其含义

异常代码	异常代码意义
0x1	非法功能码。
0x2	非法寄存器地址。
0x3	数据错误，即数据超过上限或者下限。
0x4	从机操作失败（包括数据在上下限范围之内，但是数据无效引起的错误）。
0x5	命令有效，正在处理中，主要应用在存储数据到非易失性存储器中。
0x6	从机忙，请稍后再试，主要应用在存储数据到非易失性存储器中。
0x18	信息帧错误：包括信息长度错误和校验错误。
0x20	参数不可修改。
0x22	参数受密码保护。

改写单个变频器参数的应用层协议数据单元如下：

请求格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x06
寄存器地址	2	0x0000~0xFFFF
寄存器内容	2	0x0000~0xFFFF

应答格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x06
寄存器地址	2	0x0000~0xFFFF
寄存器内容	2	0x0000~0xFFFF

如果操作请求失败，应答为错误代码和异常代码。错误代码等于（功能码+0x80），异常代码参见前面的描述。

线路诊断的应用层协议数据单元如下：

请求格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x08
子功能码	2	0x0000～0x0030
数据	2	0x0000～0xFFFF

应答格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x08
子功能码	2	0x0000～0x0030
数据	2	0x0000～0xFFFF

如果操作请求失败，应答为错误代码和意外代码。错误代码为 88H，异常代码参见前面的描述。

线路诊断支持的子功能列举如下：

子功能码	数据（请求）	数据（应答）	子功能意义
0x0001	0x0000	0x0000	重新初始化通讯：使无应答模式失效。
	0xFF00	0xFF00	重新初始化通讯：使无应答模式失效。
0x0003	“新帧尾”和“00”分别占据高低字节	“新帧尾”和“00”分别占据高低字节	设置 ASCII 模式的帧尾，这个“新帧尾”将代替老的换行符号，新帧尾掉电不保存。（注：新帧尾不能大于 0x7F，且不能等于 0x3A）
0x0004	0x0000	无应答	设置无应答模式，从机从此仅响应“重新初始化通讯请求”。主要用于隔离故障从机。
0x0030	0x0000	0x0000	设置从机不应答无效命令和错误命令。
	0x0001	0x0001	设置从机应答无效命令和错误命令。

改写多个变频器功能码和状态参数的应用层协议数据单元的请求格式：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x10
起始寄存器地址	2	0x0000～0xFFFF
操作寄存器数目	2	0x0001～0x0004
寄存器内容字节数	1	2*操作寄存器数目
寄存器内容	2*操作寄存器数目	

应答格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x10
起始寄存器地址	2	0x0000～0xFFFF
操作寄存器数目	2	0x0001～0x0004

该请求改写从起始寄存器地址开始的连续数据单元的内容。寄存器地址映射为变频器的功能码参数和控制参数等，具体的映射关系参见后面的寄存器地址映射关系定义。如果操作请求失败，异常响应如前所述。

连续存储多个寄存器参数时，变频器从最低地址的寄存器开始存储，一直到最高地址的寄存器，存储操作要么完全成功，要么从最先失败的存储地址返回。

功能码 0x41 用于改写单个变频器功能码或者控制参数，并且存储到非易失性存储单元中。其命令格式与 0x06 类似，唯一的区别是 0x06 命令操作的参数掉电后不保存，0x41 操作的参数掉电后保存。变频器中某些控制参数不能保存到非易失性存储单元中，因此对这些参数，功能码 0x41 和 0x06 具有相同的操作效果，这些参数将在后面介绍。

变频器功能码管理包括读取参数的上限和下限、读取参数特性、读取功能码菜单最大组内索引、读取下个功能码组号和上个功能码组号、读取当前显示状态参数索引以及显示下个状态参数等。参数特性包括参数的可读写特性、参数的单位以及定标关系等信息。这些命令用于远程修改变频器功能码参数。功能码管理的应用层协议数据单元如下：

请求格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x42
子功能码	2	0x0000～0x0007
数据	2	具体范围视变频器的类型而定

应答格式如下：

应用层协议数据单元	数据长度（字节数）	取值或范围
功能码	1	0x42
子功能码	2	0x0000～0x0007
数据	2	0x0000～0xFFFF

如果操作请求失败，应答为错误代码和异常代码。操作失败则进行异常应答，异常应答码参见前面的描述。

功能码管理支持的子功能列举如下：

子功能码	数据（请求）	数据（应答）	子功能意义
0x0000	功能码组号和组内索引分别占据高低字节	功能码参数的上限	读取功能码参数的上限
0x0001	功能码组号和组内索引分别占据高低字节	功能码参数的下限	读取功能码参数的下限
0x0002	功能码组号和组内索引分别占据高低字节	功能码参数特性，具体参见下面说明	读取功能码参数的特性
0x0003	功能码组号占据高字节，低字节为“00”	组内索引的最大值	读取组内索引的最大值
0x0004	功能码组号占据高字节，低字节为“00”	下个功能码组号占据高字节，低字节为“00”	读取下个功能码组号
0x0005	功能码组号占据高字节，低字节为“00”	上个功能码组号占据高字节，低字节为“00”	读取上个功能码组号
0x0006	0x3300	当前显示的状态参数索引	读当前显示的状态参数索引
0x0007	0x3300	下个状态参数索引	显示下个状态参数

状态参数组不可修改，不支持读取上下限操作。

功能码参数特性为 2 个字节长度，位定义如下：

特性参数 (Bit)	值	含义
Bit2~Bit0	000B	无小数部分
	010B	1 位小数
	011B	2 位小数
	100	3 位小数
	其它	保留
Bit3	保留	
Bit5~Bit4	00B	修改步长为 1
	其它	保留
Bit7~Bit6	01B	可以修改
	10B	运行时不可修改
	11B	厂家设定，用户不可修改
	00B	实际参数，不可修改
Bit11~Bit8	0000B	其它单位或无单位
	0001B	单位为 HZ
	0010B	单位为 A
	0011B	单位为 V
	0100B	单位为 r/min
	0101B	单位为线速度 (m/s)
	0110B	单位为百分比 (%)
	其它	保留
Bit12	1	按 4 位元组约束修改上限
	0	按字节约束修改上限
Bit15~Bit13	保留	

变频器控制参数能够完成变频器启动、停止、设定运行频率等功能，通过检索变频器状态参数能够获取变频器的运行频率、输出电流、输出转矩等参数。具体的变频器控制参数和状态参数枚举如下：

表 2 变频器控制参数索引

寄存器地址	参数名称	能否掉电保存
0x3200	控制命令字	否
0x3201	主设定	能
0x3202	运行频率设定	能
0x3203	数字闭环给定	能
0x3204	脉冲闭环给定	能
0x3205	模拟输出 AO1 设定	否
0x3206	模拟输出 AO2 设定	否
0x3207	数字输出 DO 设定	否
0x3208	频率比例设定	否
0x3209	虚拟端子控制设定	否
0x320A	设定加速时间 1	能
0x320B	设定减速时间 1	能

表 3 变频器状态参数索引

寄存器地址	参数名称
0x3300	运行状态字 1
0x3301	当前主设定的实际运行值
0x3302	从机型号
0x3303	变频器机型
0x3304	软件版本
0x3305	当前运行频率
0x3306	输出电流
0x3307	输出电压
0x3308	输出功率
0x3309	运行转速
0x330A	运行线速度
0x330B	模拟闭环反馈
0x330C	母线电压
0x330D	外部计数器
0x330E	输出转矩
0x330F	开关量输入输出端子状态: BIT0~14=X1~X8, Y1, Y2, TC, FAN, BRAKE, FWD, REV
0x3310	实际长度
0x3311	补偿后运行频率
0x3312	第一次运行故障
0x3313	第二次运行故障
0x3314	第三次（最近一次）运行故障
0x3315	运行频率设定
0x3316	运行转速设定
0x3317	模拟闭环给定
0x3318	线速度设定
0x3319	VCI
0x331A	CCI
0x331B	设定长度
0x331C	设定加速时间 1
0x331D	设定减速时间 1
0x331E	命令给定通道: 0: 面板控制                      1: 端子控制                      2: 串口控制
0x331F	变频器状态字 2
0x3320	频率给定通道: 0: 数字给定 1, 键盘▲、▼调节 1: 数字给定 2; 端子 UP/DN 调节 2: 数字给定 3; 串行口 3: VCI 模拟给定 4: CCI 模拟给定 5: 端子 PULSE 给定 6: 面板模拟给定
0x3321	累计长度

### 注意

从机型号的编码规则如下:从机型号范围为 0~9999, 百位和千位用来区别 TD、EV 等不同的变频器系列。十位和个位标识变频器是 1000 系列还是 2000 系列或者 3000 系列或者 3100 系列。例如: TDXXXX 系列变频器的从机型号为:  $0*1000+0*100+XXXX/100$ ; EVXXXX 系列变频器的从机型号为:  $1*1000+0*100+XXXX/100$ ; PVXXXX 系列变频器的从机型号为:  $1*1000+0*100+XXXX/100+1$ 。

变频器控制字位定义如下:

控制字 (位)	值	含义	功能描述
Bit2、1、0	111B	运行命令	启动变频器
	110B	方式 0 停车	按设定的减速时间停车
	101B	方式 1 停车	自由停车
	011B	方式 2 停车	保留
	100B	外部故障停车	自由停车, 变频器显示外部故障
	其余	无命令	
Bit3	1	反转	设置运行命令有效时的运转方向(对点动命令无效)
	0	正转	
Bit4	1	点动正转	
	0	点动正转停止	
Bit5	1	点动反转	
	0	点动反转停止	
Bit6	1	允许加减速	保留
	0	禁止加减速	
Bit7	1	上位机控制有效	当前上位机下发的控制字有效
	0	上位机控制无效	当前上位机下发的控制字无效
Bit8	1	主设定有效	使能主设定
	0	主设定无效	禁止主设定
Bit9	1	故障复位有效	
	0	故障复位无效	
Bit15~Bit10	000000B	保留	

注意: 点动运行给定 (Bit4, Bit5) 不得与控制字 Bit0~Bit2 同时有效!

变频器状态字 1 的位定义如下:

状态字 (位)	值	含义	备注
Bit0	1	变频器运行	
	0	变频器停机	
Bit1	1	变频器反转	
	0	变频器正转	
Bit2	1	达到主设定	
	0	未达到主设定	
Bit3	1	允许通讯控制	
	0	禁止通讯控制	
Bit7~4	0000B	保留	



状态字（位）	值	含义	备注
Bit15～8	00～0xFF	故障代码	0：表示变频器正常； 非 0：表示有故障，具体故障代码的含义参见相关类型变频器的用户手册。例如电机过载 E014 的故障代码为 0x0E，欠压为 0x1F。

变频器状态字 2 的位定义如下：

状态字（位）	值	含义
Bit0	1	点动运行
	0	非点动运行
Bit1	1	闭环运行
	0	非闭环运行
Bit2	1	PLC 运行
	0	非 PLC 运行
Bit3	1	多段频率运行
	0	非多段频率运行
Bit4	1	普通运行
	0	非普通运行
Bit5	1	摆频
	0	非摆频
Bit6	1	欠压
	0	非欠压
其余		保留

### 几点说明

1. 对于 ASCII 码格式的数据帧，如果帧长为偶数，该帧被丢弃。
2. 变频器在恢复缺省参数和参数辨识阶段不能通讯，完毕通讯恢复正常。
3. 变频器内部参数 FH.09、FP.03 不可通过通讯设置修改。
4. 可以通过写 FP.00 验证用户密码，当验证密码成功后，上位机获得访问变频器参数的权限，访问完成后，可以通过向 FP.00 写无效的密码来关闭本次访问的权限。
5. 多个多功能输入端子功能设置相同会导致功能紊乱，用户在通过 MODBUS 协议修改多功能端子功能时要避免这种情况发生。

### CRC 校验

考虑到提高速度的需要，CRC-16 通常采用表格方式实现，下面为实现 CRC-16 的 C 语言源代码，注意最后的结果已经交换了高低字节，也就是结果就是要发送的 CRC 校验和。

```

unsigned short CRC16 ( unsigned char *msg,          /* The function returns the CRC as a
unsigned char length)                               unsigned short type */
{
    unsigned char uchCRCHi = 0xFF;                  /* high byte of CRC initialized */
    unsigned char uchCRCLo = 0xFF;                  /* low byte of CRC initialized */
    unsigned uIndex;                                /* index into CRC lookup table */
    while (length--)                                /* pass through message buffer */
    {
        uIndex = uchCRCLo ^ *msg++;                 /* calculate the CRC */
        uchCRCLo = uchCRCHi ^

```

```

        (crcvalue[uIndex] >>8) );
        uchCRCHi =crcvalue[uIndex]&0xff;
    }
    return (uchCRCHi | uchCRCLo<<8) );
}

/* Table of CRC values */
const unsigned int  crcvalue[ ] = {
0x0000,0xC1C0,0x81C1,0x4001,0x01C3,0xC003,0x8002,0x41C2,0x01C6,0xC006,0x8007,
0x41C7,
0x0005,0xC1C5,0x81C4,0x4004,0x01CC,0xC00C,0x800D,0x41CD,0x000F,0xC1CF,0x81
CE,0x400E,
0x000A,0xC1CA,0x81CB,0x400B,0x01C9,0xC009,0x8008,0x41C8,0x01D8,0xC018,0x801
9,0x41D9,
0x001B,0xC1DB,0x81DA,0x401A,0x01E,0xC1DE,0x81DF,0x401F,0x01DD,0xC01D,0x8
01C,0x41DC,
0x0014,0xC1D4,0x81D5,0x4015,0x01D7,0xC017,0x8016,0x41D6,0x01D2,0xC012,0x801
3,0x41D3,
0x0011,0xC1D1,0x81D0,0x4010,0x01F0,0xC030,0x8031,0x41F1,0x0033,0xC1F3,0x81F2
,0x4032,
0x0036,0xC1F6,0x81F7,0x4037,0x01F5,0xC035,0x8034,0x41F4,0x003C,0xC1FC,0x81F
D,0x403D,
0x01FF,0xC03F,0x803E,0x41FE,0x01FA,0xC03A,0x803B,0x41FB,0x0039,0xC1F9,0x81F
8,0x4038,
0x0028,0xC1E8,0x81E9,0x4029,0x01EB,0xC02B,0x802A,0x41EA,0x01EE,0xC02E,0x802
F,0x41EF,
0x002D,0xC1ED,0x81EC,0x402C,0x01E4,0xC024,0x8025,0x41E5,0x0027,0xC1E7,0x81E
6,0x4026,
0x0022,0xC1E2,0x81E3,0x4023,0x01E1,0xC021,0x8020,0x41E0,0x01A0,0xC060,0x8061
,0x41A1,
0x0063,0xC1A3,0x81A2,0x4062,0x0066,0xC1A6,0x81A7,0x4067,0x01A5,0xC065,0x8064
,0x41A4,
0x006C,0xC1AC,0x81AD,0x406D,0x01AF,0xC06F,0x806E,0x41AE,0x01AA,0xC06A,0x80
6B,0x41AB,
0x0069,0xC1A9,0x81A8,0x4068,0x0078,0xC1B8,0x81B9,0x4079,0x01BB,0xC07B,0x807
A,0x41BA,
0x01BE,0xC07E,0x807F,0x41BF,0x007D,0xC1BD,0x81BC,0x407C,0x01B4,0xC074,0x80
75,0x41B5,
0x0077,0xC1B7,0x81B6,0x4076,0x0072,0xC1B2,0x81B3,0x4073,0x01B1,0xC071,0x8070
,0x41B0,
0x0050,0xC190,0x8191,0x4051,0x0193,0xC053,0x8052,0x4192,0x0196,0xC056,0x8057,
0x4197,
0x0055,0xC195,0x8194,0x4054,0x019C,0xC05C,0x805D,0x419D,0x005F,0xC19F,0x819
E,0x405E,
0x005A,0xC19A,0x819B,0x405B,0x0199,0xC059,0x8058,0x4198,0x0188,0xC048,0x8049,
0x4189,
0x004B,0xC18B,0x818A,0x404A,0x004E,0xC18E,0x818F,0x404F,0x018D,0xC04D,0x804
C,0x418C,
0x0044,0xC184,0x8185,0x4045,0x0187,0xC047,0x8046,0x4186,0x0182,0xC042,0x8043,
0x4183,
0x0041,0xC181,0x8180,0x4040}

```

如果在线计算各个发送字节的 CRC 校验和，则需要耗费较多时间，但是能够节省表格占用的程序空间。在线计算 CRC 的代码如下：

```

unsigned int crc_check (unsigned char *data,unsigned char length)
{
    int i;
    unsigned crc_result=0xffff;
    while (length-->0)
    {
        crc_result^=*data++;
        for (i=0;i<8;i++)
        {
            if (crc_result&0x01)
                crc_result=(crc_result>>1)^0xa001;
            else
                crc_result=crc_result>>1;
        }
    }
    return (crc_result==((crc_result&0xff)<<8)|(crc_result>>8));
}

```

### 应用举例

启动 5#变频器正转，转速设定为 50.00HZ（内部表示为 5000）的命令如下：

	地址	功能码	寄存器地址	寄存器数目	寄存器内容字节数	寄存器内容	校验和
请求	0x05	0x10	0x3200	0x0002	0x04	0x01C7,0x1388	0x16A9
响应	0x05	0x10	0x3200	0x0002	无	无	0x4EF4

5#变频器点动正转：

	地址	功能码	寄存器地址	寄存器内容	校验和
请求	0x05	0x06	0x3200	0x00D0	0x876A
响应	0x05	0x06	0x3200	0x00D0	0x876A

5#变频器点动停止：

	地址	功能码	寄存器地址	寄存器内容	校验和
请求	0x05	0x06	0x3200	0x00C0	0x86A6
响应	0x05	0x06	0x3200	0x00C0	0x86A6

5#变频器故障复位：

	地址	功能码	寄存器地址	寄存器内容	校验和
请求	0x05	0x06	0x3200	0x0180	0x86C6
响应	0x05	0x06	0x3200	0x0180	0x86C6

读取 5#变频器的运行频率，变频器应答运行频率为 50.00HZ：

	地址	功能码	寄存器地址	寄存器数目或者读取字节数	寄存器内容	校验和
请求	0x05	0x03	0x3301	0x0001	无	0xDB0A
响应	0x05	0x03	无	0x02	0x1388	0x44D2

改写 5#变频器的加速时间 1（即功能码 F0.10）为 10.0s，掉电不保存。

	地址	功能码	寄存器地址	寄存器内容	校验和
请求	0x05	0x06	0x000A	0x0064	0xA9A7
响应	0x05	0x06	0x000A	0x0064	0xA9A7

读取 5#变频器的输出电流，变频器应答输出电流为 30.0A。

	地址	功能码	寄存器地址	寄存器数目或者读取字节数	寄存器内容	校验和
请求	0x05	0x03	0x3306	0x0001	无	0x6ACB
响应	0x05	0x03	无	0x02	0x12C	0x49C9

读取 5#变频器的减速时间（即 F0.11），变频器应答减速时间为 6.0S。

	地址	功能码	寄存器地址	寄存器数目或者读取字节数	寄存器内容	校验和
请求	0x05	0x03	0x000B	0x0001	无	0xF4C4
响应	0x05	0x03	无	0x02	0x003C	0x4995

**变频器的定标关系**

1) 频率的定标为 1: 100

欲使变频器按 50Hz 运转，则主设定应为 0x1388（5000）。

2) 时间的定标为 1: 10

欲使变频器加速时间为 30S，则功能码设定应为 0x012c（300）。

3) 电流的定标为 1: 10

若变频器反馈电流为 0x012c，则该变频器当前电流为 30A。

4) 输出功率为其绝对值。

5) 其它（如端子输入、输出等）请参考变频器用户手册。

## 附录三 推荐的配件参数

说明：电抗器、EMI 滤波器等配件用户自配。以下所列的型号在我司变频器上经过试验，如有需要，请与本公司联系。

### 输入、输出电抗器

型号说明

交流进线电抗器：TDL-4AI01-0015，其中 0015 表示功率等级，同变频器功率等级说明。

交流输出电抗器：TDL-4AO01-0015，其中 0015 表示功率等级，同变频器功率等级说明。

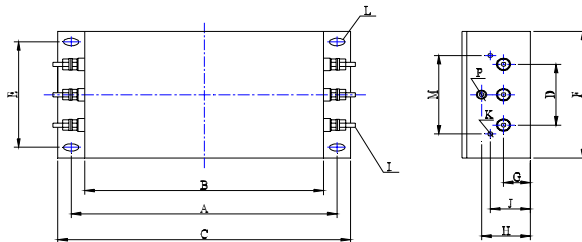
附表 8 EV1000 系列交流进线电抗器机械参数表

适用变频器	电抗器型号
EV1000-2S0004G	TDL-2AI01-0004
EV1000-2S0007G	TDL-2AI01-0007
EV1000-2S0015G	TDL-2AI01-0015
EV1000-2S0022G	TDL-2AI01-0022
EV1000-4T0007G	TDL-4AI01-0007
EV1000-4T0015G	TDL-4AI01-0015
EV1000-4T0022G	TDL-4AI01-0022
EV1000-4T0037G/P	TDL-4AI01-0037
EV1000-4T0055G/P	TDL-4AI01-0055

附表 9 EV1000 系列交流输出电抗器机械参数表

适用变频器	电抗器型号
EV1000-2S0004G	TDL-2AO01-0004
EV1000-2S0007G	TDL-2AO01-0007
EV1000-2S0015G	TDL-2AO01-0015
EV1000-2S0022G	TDL-2AO01-0022
EV1000-4T0007G	TDL-4AO01-0007
EV1000-4T0015G	TDL-4AO01-0015
EV1000-4T0022G	TDL-4AO01-0022
EV1000-4T0037G/P	TDL-4AO01-0037
EV1000-4T0055G/P	TDL-4AO01-0055

### EMI 滤波器



附图 3 EMI 滤波器外形图

附表 10 EMI 滤波器机械参数表

适用变频器	滤波器型号	尺寸[mm]															概重[kg]
		A	B	C	D	E	F	G	H	I	J	K	M	N	P	L	
EV1000-2S0004G	DL-5EBT1	184	160	202	42	60	86	18	58	M4	38	-	-	-	M4	6.4×9.4	1.7
EV1000-4T0007G																	
EV1000-4T0015G																	
EV1000-4T0022G	DL-10EBT1	184	160	202	42	60	86	18	58	M4	38	-	-	-	M4	6.4×9.4	1.7
EV1000-2S0007G																	
EV1000-4T0037G/P	DL-20EBT1	243	220	261	58	70	100	25	90	M6	58	M4	74	49	M6	6.4×9.4	3.6
EV1000-4T0055G/P																	
EV1000-2S0015G	DL-25EBT1	243	220	261	58	70	100	25	90	M6	58	M4	74	49	M6	6.4×9.4	3.6
EV1000-2S0022G																	