SYNTRON 森创

TS 伺服驱动器

——— 通讯手册

版本 V 2

北京和利时电机技术有限公司
BEIJING HOLLYSYS ELECTRIC TECHNOLOGY CO.,LTD.

版权声明

SYNTRON 森创®是北京和利时电机公司（以下简称和利时电机）于 2005 年推出的产品品牌。这个品牌浓缩了公司的核心技术和影响力，是公司始终注重自主创新，保持技术优势的体现。

说明书的内容参照了相关法律基准和行业基准。如对本说明书提供的内容有疑问，请向销售人员咨询，致电客服热线，联系官网客服或致信本公司。

和利时电机保留在不事先通知的情况下，修改本手册中的产品和产品规格参数等权力。手册请联系销售人员，或在和利时电机的官方网站下载相关手册。

目录

第一章 产品概况

1.1 概述

TS 伺服驱动器标准版支持通过 RS232 / RS485 修改及观测内部参数。如需通讯控制，标准版仅支持 Modbus 内部速度模式；**Modbus 内部位置模式定制版**支持内部位置模式；**CANopen 定制版**支持 CANopen 标准协议。TS 的通讯接口和支持协议如下：

型号	通讯端口	通讯接口	通讯协议
TS 系列	CN1	CAN（选配）	CANopen
		RS232 / 485	Modbus

1.2 通讯端口

通讯端口 CN1 定义如下：

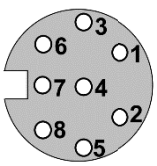


图 1-1 RJ45 的管脚分布

类型	Pin	定义
RS232	2	RS232_TXD
	8	RS232_RXD
	5	GND
RS485	7	RS485_A
	4	RS485_B
CAN	1	CAN_L
	6	CAN_H
屏蔽	3	屏蔽

1.3 注意事项

- 在驱动器上电的情况下，**请勿热插拔总线及其他接口！否则可能导致损坏！**
- 驱动**请勿将通讯端口内管脚短路！有可能造成损坏。**
- CAN 和 Modbus 总线构成网络时，一般需要在网络的两个终端各安装 120Ω 电阻进行阻抗匹配，以实现规范良好的通讯性能。推荐组网连接方式如下所示：

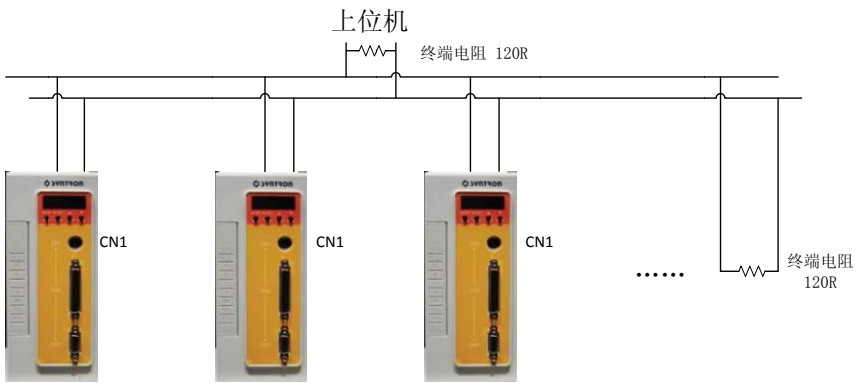


图 1-2 多台 TS 组网

1.4 参数修改

TS 伺服驱动器在使用时，用户需自行设定某些参数，例如运行模式、电机速度环比例积分等参数。参数修改有两种方式，分别为通讯修改和显示面板设置。

1.4.1 通讯修改

如果用通讯修改参数，需先确认 Fn99 为 0，这样通讯修改参数可以保存在驱动器内部 Flash，否则该参数只写入 RAM，不保存驱动器内部 Flash。

■ 关键参数

参数编号	CAN INDEX	Modbus Address	参数名称	参数内容	设定范围	出厂设定
Fn_3F	0x303F	0x003F	密码	某些重要参数需要打开密码才能修改 默认 9870，如需修改重要参数，需设为 9876	-	9870
Fn_99	0x2099	0x0099	通讯参数保存设置	1：参数在 RAM 中运行； 0：参数写入 flash 保存 注：Flash 写入寿命有限，如该参数经常更改且无需保存，请置 Fn99 为 1，否则会写坏驱动器 FLASH	0~1	0

■ Modbus 总线修改

协议详细请参考第二章 Modbus，例程如下：

发送	设备号	功能码	寄存器起始地址		数据		CRC 校验	
	01	06	00	00	00	14	89	C5
返回	设备号	功能码	寄存器起始地址		数据		CRC 校验	
	01	06	00	00	00	14	89	C5
解析	表示写 Fn00 为 20，即选择控制模式为 CANopen 模式							

■ CAN 总线修改

协议详细请参考第三章 CANopen，例程如下：

例 1：写内部速度 Fn_33

601 2B 33 20 00 C8 00 00 00 // Fn_33<--200RPM

581 60 33 20 00 C8 00 00 00 // 回复

1.4.2 显示面板修改

TS 驱动器显示面板可以很方便的调整参数、观测伺服状态，有修改参数的最高优先级。以下为数码管的显示：

数字	1	2	3	4	5	6	7	8	9	0	小数点	
字母	A	b	c	d	E	F	G	H	J	L	n	o
	P	q	r	S	T	U	V	y	Null	-		

数码管显示界面分为三层：当前状态界面、功能码选择界面以及参数界面，具体见下：

显示类别	显示定义	说明	
当前运行状态界面	默认状态	伺服 OFF，电压达到允许工作电压，显示 rEAdy	
		伺服 ON，默认显示当前电机转速值，例如 1000	故障警告时，闪烁显示当前的报警代码，如 E.oLod
功能码选择	Dn 编号	状态观测参数 Dn	
	Fn 编号	功能参数 Fn	
参数	参数值	显示伺服内部选定参数的值，如：1024，220	
按键	定义	操作说明	
MODE	界面切换	➢ 在“当前状态界面”、“功能码选择”之间切换 ➢ 在检查或编辑驱动器内部参数时，用于退回到功能码选择界面	
◀/SET	确认&移位	➢ 在功能码、参数界面，短按用来选择修改位，被选的修改位闪烁 ➢ 在功能码界面，长按 1 秒则进入该参数编辑界面 ➢ 在修改完参数后，长按 1 秒则保存修改后的参数值 ➢ JOG 模式的“运行/停止”切换功能按键	
▲	递增按键	➢ 选择功能码，修改参数时，被选择修改的闪烁“位”增 1 ➢ 在 JOG 模式按住该键控制电机正转	
▼	递减按键	➢ 选择功能码，修改参数时，被选择修改的闪烁“位”减 1 ➢ 在 JOG 模式按住该键控制电机反转	

第二章　Modbus

2.1　接线说明

用户可通过基于 RS232 / 485 的 Modbus 总线读写 TS 伺服驱动器参数，达到控制的目的，以下为常见接线：

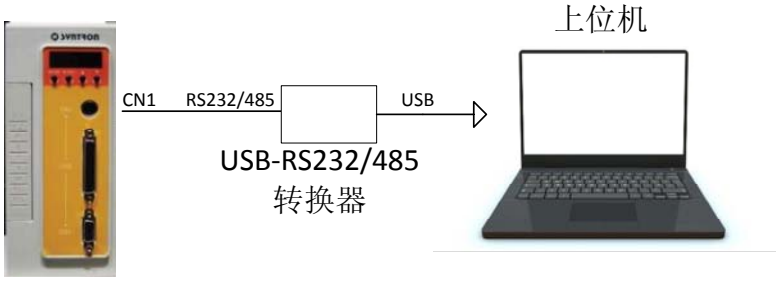


图 2-1　TS 连接 PC

2.2　关键参数

TS 伺服驱动器默认 Modbus 地址为 1，波特率为 115200bps，ModBus_RTU 模式，用户可根据需求设置。协议采用 8 个数据位、1 个停止位、无奇偶校验。

功能参数（Fn_00~Fn_2FF）对应的寄存器地址为：0x0000~0x02FF；

状态参数（Dn_00~Dn_2F）对应的寄存器地址为：0x1000~0x102F。

地址和数据均是高字节在前。

参数编号	Modbus 地址	参数名称	参数内容	设定范围	出厂设定
Fn_00	0x0000	控制模式	0：外部速度；1：内部速度；2：位置模式； 3：JOG 模式；4：转矩模式；5-10：混合控制模式； 14：内部位置模式；　20：CANopen 模式；	0-20	2
Fn_3A	0x003A	Modbus 地址	0 为广播地址，1~127 为站址编号	0~127	1
Fn_3C	0x003C	Modbus 通讯波特率	0：2400bps；　1：9600bps；　2：38400bps； 3：57600bps；　4：115200bps； 5：192000bps；　6：512000bps；	0~6	4
Fn_3D	0x003D	Modbus 模式选择	001：RS232 ModbusRTU； 002：RS232 ModbusASCII 101：RS485 ModbusRTU； 102：RS485 ModbusASCII	-	001
Fn_99	0x0099	通讯参数模式	1：指令在 RAM 中运行；　0：指令写入 Flash 保存 注：注：Flash 写入寿命有限，如该参数经常更改且无需保存，请置 Fn99 为 1，否则会写坏驱动器 FLASH	0~1	0

2.3 协议说明（Modbus_RTU）

在 Modbus_RTU 模式下，每个 8 位数据由两个 4 位的十六进制字符组成，例如 1 个字节的数据 0x64；RTU 模式采用 CRC 校验勘误（低字节在前），以超过 10ms 的静止时段代表通讯结束。支持的功能码示例如下：

读驱动器 IO 输出（0x01）

发送	设备号	功能码	寄存器起始地址		寄存器单元长度		CRC 校验	
	01	01	00	00	00	01	FD	CA
返回	设备号	功能码	字节数		数据		CRC 校验	
	01	01	01		01		90	48
解析	TS 驱动器有 4 路输出，通过以上指令读到 0x01 表示：输出 1 有效，输出 2 为 0 未导通；0x00 表示 4 路均不导通；0x0F 表示 4 路均导通；							

读驱动器 IO 输入（0x02）

发送	设备号	功能码	寄存器起始地址		寄存器单元长度		CRC 校验	
	01	02	00	00	00	01	B9	CA
返回	设备号	功能码	字节数		数据		CRC 校验	
	01	02	01		00		A1	88
解析	当前表示读取的 8 路输入均为 0； 8 路输入分别对应二进制低位到高位，例如返回 0x03，二进制为 0011，则表示输入 1 和 2 有效。							

读寄存器（0x03）

发送	设备号	功能码	寄存器起始地址		寄存器单元长度		CRC 校验		
	01	03	10	2E	00	02	A0	C2	
返回	设备号	功能码	字节数	数据 1		数据 2		CRC 校验	
	01	03	04	F3	2D	00	0B	18	B9
解析	表示读取 Dn2E = 0xF3 2D 和 Dn2F = 0x000B 两个寄存器（编码器绝对位置），则返回的编码器位置值为 0x00 0B F3 2D。								

写单个寄存器（0x06）

发送	设备号	功能码	寄存器起始地址		数据		CRC 校验	
	01	06	00	33	00	C8	78	53
返回	设备号	功能码	寄存器起始地址		数据		CRC 校验	
	01	06	00	33	00	C8	78	53
解析	表示写 Fn33 为 200rpm							

写多个寄存器（0x10）

发送	设备号	功能码	寄存器起始地址		寄存器长度		字节数	数据				CRC 校验	
	01	10	00	30	00	02	04	00	05	00	05	20	B9
返回	设备号	功能码	寄存器起始地址		寄存器长度		CRC 校验						
	01	10	00	30	00	02	41	C7					
解析	表示连续写 Fn30 和 Fn31 为 5。												

2.4 协议说明（Modbus_ASCII）

在 Modbus_ASCII 模式下，每 8 位数据由两个 ASCII 字符组成，例如 1 个字节数据 0x64 以 ASCII 字符的‘6’（36H）和‘4’（34H）组成。ASCII 采用 LRC 校验模式，相比 RTU 模式增加起始字符为‘:’（3AH），校验后加结束码 End1 和 End0 分别为 0D H 和 0AH。

常用数字的 ASCII 码如下图所示：

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

Modbus_ASCII 数据结构如下：

STX	起始字符 ’:’（3AH）
ADR	通讯地址：1byte 包含两个 ASCII 码，例如地址 1 为：30H 31H
CMD	命令码：1byte 包含两个 ASCII 码
DATA(n-1)	数据内容：n-word=2n-byte 包含了 4n 个 ASCII 码，n<=12
……	
DATA(0)	
LRC	校验码：1byte 包含两个 ASCII 码
END1	结束码 1：0DH CR
END2	结束码 2：0AH LF

例程详见上一节。

2.5 内部速度模式

在使用 Modbus 总线控制电机运转时，一般步骤如下：

STEP1. 确认驱动器参数：

Fn_00 = 1（速度模式）；
Fn_99 = 1（通讯修改参数不保存）；
Fn_33 = 0（初始上电速度为0）；
Fn_38 = 0（初始状态不使能）。

STEP2. 设置：

1. 设置电机速度Fn33：
   0x01 06 00 33 00 C8 78 53 //设置速度为200rpm

2. 使能电机：

0x01 06 00 38 00 01 C9 C7 //设置 Fn38=1，使电机使能；

3.读 Dn00 电机转速：

0x01 03 10 00 00 01 80 CA //读速度，单位 rpm

■ **注意事项：**

驱动器每次上电后，只需刷入 Fn33 速度值和 Fn38 使能，电机即可按要求运行，由于写入了 Fn99，断电后写入的值不保存，都会恢复初始设定状态。

2.6 内部位置模式

TS 系列驱动器可定制支持内部位置模式。内部位置以触发 32 组位置信息的形式实现。相关参数如下：

2.6.1 关键参数

参数编码	定 义
Fn_100	PATH#1 定义寄存器 1
Fn_101	PATH#1 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_102	PATH#1 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_103	PATH#1 电机运转速度（rpm）
Fn_104	PATH#1 加速时间(ms)
Fn_105	PATH#1 减速时间(ms)
Fn_106	PATH#1 延时时间(ms)
Fn_107	PATH#1 定义寄存器 2
Fn_108	PATH#2 定义寄存器 1
Fn_109	PATH#2 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_10A	PATH#2 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_10B	PATH#2 电机运转速度（rpm）
Fn_10C	PATH#2 加速时间(ms)
Fn_10D	PATH#2 减速时间(ms)
Fn_10E	PATH#2 延时时间(ms)
Fn_10F	PATH#2 定义寄存器 2
Fn_110	PATH#3 定义寄存器 1
Fn_111	PATH#3 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_112	PATH#3 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_113	PATH#3 电机运转速度（rpm）
Fn_114	PATH#3 加速时间(ms)
Fn_115	PATH#3 减速时间(ms)

Fn_116	PATH#3 延时时间(ms)
Fn_117	PATH#3 定义寄存器 2
Fn_118	PATH#4 定义寄存器 1
Fn_119	PATH#4 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_11A	PATH#4 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_11B	PATH#4 电机运转速度（rpm）
Fn_11C	PATH#4 加速时间(ms)
Fn_11D	PATH#4 减速时间(ms)
Fn_11E	PATH#4 延时时间(ms)
Fn_11F	PATH#4 定义寄存器 2
Fn_120	PATH#5 定义寄存器 1
Fn_121	PATH#5 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_122	PATH#5 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_123	PATH#5 电机运转速度（rpm）
Fn_124	PATH#5 加速时间(ms)
Fn_125	PATH#5 减速时间(ms)
Fn_126	PATH#5 延时时间(ms)
Fn_127	PATH#5 定义寄存器 2
Fn_128	PATH#6 定义寄存器 1
Fn_129	PATH#6 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_12A	PATH#6 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_12B	PATH#6 电机运转速度（rpm）
Fn_12C	PATH#6 加速时间(ms)
Fn_12D	PATH# 6 减速时间(ms)
Fn_12E	PATH# 6 延时时间(ms)
Fn_12F	PATH#6 定义寄存器 2
Fn_130	PATH#7 定义寄存器 1
Fn_131	PATH#7 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_132	PATH#7 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_133	PATH#7 电机运转速度（rpm）
Fn_134	PATH#7 加速时间(ms)
Fn_135	PATH#7 减速时间(ms)
Fn_136	PATH#7 延时时间(ms)
Fn_137	PATH#7 定义寄存器 2
Fn_138	PATH#8 定义寄存器 1

Fn_139	PATH#8 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_13A	PATH#8 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_13B	PATH#8 电机运转速度（rpm）
Fn_13C	PATH#8 加速时间(ms)
Fn_13D	PATH#8 减速时间(ms)
Fn_13E	PATH#8 延时时间(ms)
Fn_13F	PATH#8 定义寄存器 2
Fn_140	PATH#9 定义寄存器 1
Fn_141	PATH#9 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_142	PATH#9 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_143	PATH#9 电机运转速度（rpm）
Fn_144	PATH#9 加速时间(ms)
Fn_145	PATH#9 减速时间(ms)
Fn_146	PATH#9 延时时间(ms)
Fn_147	PATH#9 定义寄存器 2
Fn_148	PATH#10 定义寄存器 1
Fn_149	PATH#10 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_14A	PATH#10 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_14B	PATH#10 电机运转速度（rpm）
Fn_14C	PATH#10 加速时间(ms)
Fn_14D	PATH#10 减速时间(ms)
Fn_14E	PATH#10 延时时间(ms)
Fn_14F	PATH#10 定义寄存器 2
Fn_150	PATH#11 定义寄存器 1
Fn_151	PATH#11 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_152	PATH#11 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_153	PATH#11 电机运转速度（rpm）
Fn_154	PATH#11 加速时间(ms)
Fn_155	PATH#11 减速时间(ms)
Fn_156	PATH#11 延时时间(ms)
Fn_157	PATH#11 定义寄存器 2
Fn_158	PATH#12 定义寄存器 1
Fn_159	PATH#12 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_15A	PATH#12 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_15B	PATH#12 电机运转速度（rpm）
Fn_15C	PATH#12 加速时间(ms)

Fn_15D	PATH#12 减速时间(ms)
Fn_15E	PATH#12 延时时间(ms)
Fn_15F	PATH#12 定义寄存器 2
Fn_160	PATH#13 定义寄存器 1
Fn_161	PATH#13 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_162	PATH#13 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_163	PATH#13 电机运转速度（rpm）
Fn_164	PATH#13 加速时间(ms)
Fn_165	PATH#13 减速时间(ms)
Fn_166	PATH#13 延时时间(ms)
Fn_167	PATH#13 定义寄存器 2
Fn_168	PATH#14 定义寄存器 1
Fn_169	PATH#14 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_16A	PATH#14 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_16B	PATH#14 电机运转速度（rpm）
Fn_16C	PATH#14 加速时间(ms)
Fn_16D	PATH#14 减速时间(ms)
Fn_16E	PATH#14 延时时间(ms)
Fn_16F	PATH#14 定义寄存器 2
Fn_170	PATH#15 定义寄存器 1
Fn_171	PATH#15 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_172	PATH#15 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_173	PATH#15 电机运转速度（rpm）
Fn_174	PATH#15 加速时间(ms)
Fn_175	PATH#15 减速时间(ms)
Fn_176	PATH#15 延时时间(ms)
Fn_177	PATH#15 定义寄存器 2
Fn_178	PATH#16 定义寄存器 1
Fn_179	PATH#16 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_17A	PATH#16 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_17B	PATH#16 电机运转速度（rpm）
Fn_17C	PATH#16 加速时间(ms)
Fn_17D	PATH#16 减速时间(ms)
Fn_17E	PATH#16 延时时间(ms)
Fn_17F	PATH#16 定义寄存器 2

Fn_180	PATH#17 定义寄存器 1
Fn_181	PATH#17 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_182	PATH#17 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_183	PATH#17 电机运转速度（rpm）
Fn_184	PATH#17 加速时间(ms)
Fn_185	PATH#17 减速时间(ms)
Fn_186	PATH#17 延时时间(ms)
Fn_187	PATH#17 定义寄存器 2
Fn_188	PATH#18 定义寄存器 1
Fn_189	PATH#18 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_18A	PATH#18 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_18B	PATH#18 电机运转速度（rpm）
Fn_18C	PATH#18 加速时间(ms)
Fn_18D	PATH#18 减速时间(ms)
Fn_18E	PATH#18 延时时间(ms)
Fn_18F	PATH#18 定义寄存器 2
Fn_190	PATH#19 定义寄存器 1
Fn_191	PATH#19 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_192	PATH#19 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_193	PATH#19 电机运转速度（rpm）
Fn_194	PATH#19 加速时间(ms)
Fn_195	PATH#19 减速时间(ms)
Fn_196	PATH#19 延时时间(ms)
Fn_197	PATH#19 定义寄存器 2
Fn_198	PATH#20 定义寄存器 1
Fn_199	PATH#20 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_19A	PATH#20 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_19B	PATH#20 电机运转速度（rpm）
Fn_19C	PATH#20 加速时间(ms)
Fn_19D	PATH#20 减速时间(ms)
Fn_19E	PATH#20 延时时间(ms)
Fn_19F	PATH#20 定义寄存器 2
Fn_1A0	PATH#21 定义寄存器 1
Fn_1A1	PATH#21 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1A2	PATH#21 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1A3	PATH#21 电机运转速度（rpm）

Fn_1A4	PATH#21 加速时间(ms)
Fn_1A5	PATH#21 减速时间(ms)
Fn_1A6	PATH#21 延时时间(ms)
Fn_1A7	PATH#21 定义寄存器 2
Fn_1A8	PATH#22 定义寄存器 1
Fn_1A9	PATH#22 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1AA	PATH#22 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1AB	PATH#22 电机运转速度（rpm）
Fn_1AC	PATH#22 加速时间(ms)
Fn_1AD	PATH#22 减速时间(ms)
Fn_1AE	PATH#22 延时时间(ms)
Fn_1AF	PATH#22 定义寄存器 2
Fn_1B0	PATH#23 定义寄存器 1
Fn_1B1	PATH#23 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1B2	PATH#23 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1B3	PATH#23 电机运转速度（rpm）
Fn_1B4	PATH#23 加速时间(ms)
Fn_1B5	PATH#23 减速时间(ms)
Fn_1B6	PATH#23 延时时间(ms)
Fn_1B7	PATH#23 定义寄存器 2
Fn_1B8	PATH#24 定义寄存器 1
Fn_1B9	PATH#24 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1BA	PATH#24 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1BB	PATH#24 电机运转速度（rpm）
Fn_1BC	PATH#24 加速时间(ms)
Fn_1BD	PATH#24 减速时间(ms)
Fn_1BE	PATH#24 延时时间(ms)
Fn_1BF	PATH#24 定义寄存器 2
Fn_1C0	PATH#25 定义寄存器 1
Fn_1C1	PATH#25 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1C2	PATH#25 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1C3	PATH#25 电机运转速度（rpm）
Fn_1C4	PATH#25 加速时间(ms)
Fn_1C5	PATH#25 减速时间(ms)
Fn_1C6	PATH#25 延时时间(ms)

Fn_1C7	PATH#25 定义寄存器 2
Fn_1C8	PATH#26 定义寄存器 1
Fn_1C9	PATH#26 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1CA	PATH#26 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1CB	PATH#26 电机运转速度（rpm）
Fn_1CC	PATH#26 加速时间(ms)
Fn_1CD	PATH#26 减速时间(ms)
Fn_1CE	PATH#26 延时时间(ms)
Fn_1CF	PATH#26 定义寄存器 2
Fn_1D0	PATH#27 定义寄存器 1
Fn_1D1	PATH#27 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1D2	PATH#27 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1D3	PATH#27 电机运转速度（rpm）
Fn_1D4	PATH#27 加速时间(ms)
Fn_1D5	PATH#27 减速时间(ms)
Fn_1D6	PATH#27 延时时间(ms)
Fn_1D7	PATH#27 定义寄存器 2
Fn_1D8	PATH#28 定义寄存器 1
Fn_1D9	PATH#28 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1DA	PATH#28 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1DB	PATH#28 电机运转速度（rpm）
Fn_1DC	PATH#28 加速时间(ms)
Fn_1DD	PATH#28 减速时间(ms)
Fn_1DE	PATH#28 延时时间(ms)
Fn_1DF	PATH#28 定义寄存器 2
Fn_1E0	PATH#29 定义寄存器 1
Fn_1E1	PATH#29 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1E2	PATH#29 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1E3	PATH#29 电机运转速度（rpm）
Fn_1E4	PATH#29 加速时间(ms)
Fn_1E5	PATH#29 减速时间(ms)
Fn_1E6	PATH#29 延时时间(ms)
Fn_1E7	PATH#29 定义寄存器 2
Fn_1E8	PATH#30 定义寄存器 1
Fn_1E9	PATH#30 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1EA	PATH#30 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）

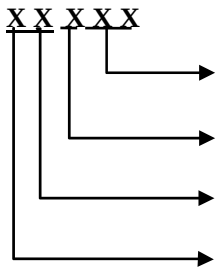
Fn_1EB	PATH#30 电机运转速度（rpm）
Fn_1EC	PATH#30 加速时间(ms)
Fn_1ED	PATH#30 减速时间(ms)
Fn_1EE	PATH#30 延时时间(ms)
Fn_1EF	PATH#30 定义寄存器 2
Fn_1F0	PATH#31 定义寄存器 1
Fn_1F1	PATH#31 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1F2	PATH#31 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1F3	PATH#31 电机运转速度（rpm）
Fn_1F4	PATH#31 加速时间(ms)
Fn_1F5	PATH#31 减速时间(ms)
Fn_1F6	PATH#31 延时时间(ms)
Fn_1F7	PATH#31 定义寄存器 2
Fn_1F8	PATH#32 定义寄存器 1
Fn_1F9	PATH#32 位置坐标的十位到十万位（机械坐标单位，mm 或°）
Fn_1FA	PATH#32 位置坐标的个位和三个小数位（机械坐标单位，mm 或°）
Fn_1FB	PATH#32 电机运转速度（rpm）
Fn_1FC	PATH#32 加速时间(ms)
Fn_1FD	PATH#32 减速时间(ms)
Fn_1FE	PATH#32 延时时间(ms)
Fn_1FF	PATH#32 定义寄存器 2
Fn_200	定义回归机械零点的方式
Fn_201	定义回归机械零点的第一速度值
Fn_202	定义回归机械零点的第二速度值
Fn_203	定义机械零点的用户位置坐标值高位
Fn_204	定义机械零点的用户位置坐标值低位
Fn_205	机械传动结构定义参数 MOTO_REVS
Fn_206	机械传动结构定义参数 MECH_MOV
Fn_207	定义回归机械零点后延时时间（ms）
Fn_208	回零后，执行的路径编码
Fn_209	停止设定 0：按当前执行路径定义的减速度停止； 1：立即停止；
Fn_20A	路径（PATH# n）触发信号源选择输入 0：IO； 1：RS485，RS232；
Fn_20B	需要执行的路径号（PATH# n）
Fn_20C	内部位置路径（PATH# n）触发信号输入
Fn_20D	内部位置回零路径触发信号输入

Fn_20E	停止当前路径（PATH# n）执行
Fn_20F	
Fn_210	PATH#1 S 加减速时间(ms)
Fn_211	PATH#2 S 加减速时间(ms)
Fn_212	PATH#3 S 加减速时间(ms)
Fn_213	PATH#4 S 加减速时间(ms)
Fn_214	PATH#5 S 加减速时间(ms)
Fn_215	PATH#6 S 加减速时间(ms)
Fn_216	PATH#7 S 加减速时间(ms)
Fn_217	PATH#8 S 加减速时间(ms)
Fn_218	PATH#9 S 加减速时间(ms)
Fn_219	PATH#10 S 加减速时间(ms)
Fn_21A	PATH#11 S 加减速时间(ms)
Fn_21B	PATH#12 S 加减速时间(ms)
Fn_21C	PATH#13 S 加减速时间(ms)
Fn_21D	PATH#14 S 加减速时间(ms)
Fn_21E	PATH#15 S 加减速时间(ms)
Fn_21F	PATH#16 S 加减速时间(ms)
Fn_220	PATH#17 S 加减速时间(ms)
Fn_221	PATH#18 S 加减速时间(ms)
Fn_222	PATH#19 S 加减速时间(ms)
Fn_223	PATH#20 S 加减速时间(ms)
Fn_224	PATH#21 S 加减速时间(ms)
Fn_225	PATH#22 S 加减速时间(ms)
Fn_226	PATH#23 S 加减速时间(ms)
Fn_227	PATH#24 S 加减速时间(ms)
Fn_228	PATH#25 S 加减速时间(ms)
Fn_229	PATH#26 S 加减速时间(ms)
Fn_22A	PATH#27 S 加减速时间(ms)
Fn_22B	PATH#28 S 加减速时间(ms)
Fn_22C	PATH#29 S 加减速时间(ms)
Fn_22D	PATH#30 S 加减速时间(ms)
Fn_22E	PATH#31 S 加减速时间(ms)
Fn_22F	PATH#32 S 加减速时间(ms)

2.6.1 参数说明

- 回归机械零点方式选择（**Fn_200**）

Fn_200 结构：



A、回零后，执行的路径编码

B、回零找 Z 脉冲设置

C、回零方向和零位信号选择

D、零点回归运动启动方式

序号	取值	含义
A	00	找到零点并冲过零点后反向对齐零点
	01~32	找到零点后走 Path1~ Path32 设定的定长
	33	找到零点并冲过零点后立即停止，不反向对齐零点
B	0	回零不找 Z 脉冲（当 C 项设置为 0，1，2，3 时）
	1	回零找 Z 脉冲（当 C 项设置为 0，1，2，3 时）
C	0	专用机械零点开关信号，CW（顺时针）反转方向回归机械零点
	1	专用机械零点开关信号，CCW（逆时针）正转方向回归机械零点
	2	反转限位开关信号 CWL，电机按照 CW（顺时针）反转方向回归机械零点
	3	正转限位开关信号 CCWL，电机按照 CCW（逆时针）正转方向回归机械零点
	4	CW（顺时针）反转直接寻找 Z 脉冲作为回归原点
	5	CCW（逆时针）正转直接寻找 Z 脉冲作为回归原点
	6	CW（顺时针）反转，转矩到达信号作为回归原点
	7	CCW（逆时针）正转，转矩到达信号作为回归原点
D	0	无需回原点
	1	伺服 on，马上执行回归机械零点运动
	2	伺服 on，由有效回零信号触发执行回归机械零点运动

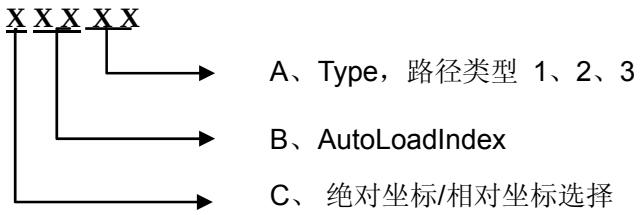
- 零点回归速度（**Fn_201**，**Fn_202**）单位 **rpm**
  Fn_201：第一机械零点回归速度，高速回零的速度
  Fn_202：第二机械零点回归速度，收到零位信号后低速找 Z 信号的速度
- 机械零点坐标值（**Fn_203**，**Fn_204**）单位 **mm**
  Fn_203：多圈圈数
  Fn_204：单圈的脉冲数
- 机械传动结构定义（**Fn_205**，**Fn_206**）
  Fn_205：伺服电机旋转圈数；

Fn_206：伺服电机旋转 Fn_205 圈，机械部件移动位移（mm）或旋转的角度（°）。

参数 Fn_205 和 Fn_206 定义了电机圆周分辨单位与用户机械坐标单位之间的换算关系，即伺服单元所有位置坐标参数都可以按照用户机械最终控制目标的机械坐标单位进行设置。

- **PATH# xx** 定义寄存器 1（以 Fn_100 为例）

例：Fn_100 结构



序号	取值	含义
A	1	SPEED 定速控制： 1、当 Fn_106（即：PATH# xx 延时时间（ms）为 0）时，速度到达“电机运转速度” 参数设定的速度值之后，以该速度值一直运行下去； 2、当 Fn_106（即：PATH# xx 延时时间（ms）不为 0）时，延时时间从 TRIG 信号开始计时，则经过 Fn_106 设定的延时时间后，自动载入 AutoLoadIndex 指定的路径。
	2	SINGLE 定位控制：伺服电机按照 Fn_205、Fn_206 两个参数设定的位置运行，运行完指定的圈数后则停止。
	3	AUTO 定位控制：伺服电机按照 Fn_205、Fn_206 两个参数设定的位置运行，运行完指定的圈数后自动载入 AutoLoadIndex 指定的路径编号。
B	当 Type= 3 时，当前路径执行完成，自动跳跃到 AutoLoadIndex 指定的路径。	
C	0	相对坐标
	1	绝对坐标

- **PATH# xx** 位置坐标

  Fn_101：位置坐标的十位到十万位（机械坐标单位，mm 或 °）。

  Fn_102：位置坐标的个位和三个小数位（机械坐标单位，mm 或 °）。

  注：若“PATH# xx 位置坐标”为正，则 TRIG 后电机正转（逆时针）；

  若“PATH# xx 位置坐标”为负，则 TRIG 后电机反转（顺时针）。

- **PATH# xx** 电机运转速度（**rpm**）

  路径执行时的电机运转速度。

- **PATH# xx** 加速时间（**ms**）**:Ta**

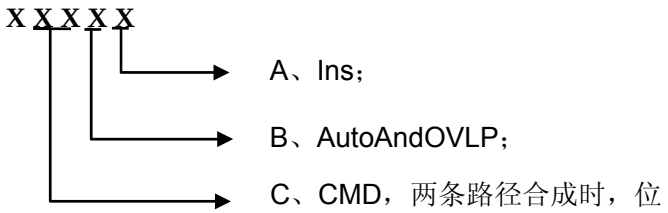
  路径执行时的电机的加速时间。

- **PATH# xx** 减速时间（**ms**）**:Td**

  路径执行时的电机的减速时间。

- **PATH# xx** 延时时间（**ms**）
  路径执行完成延时时间。
- **PATH# xx S** 加减速时间（**ms**）**: Ts**
  当 Ts = 0 时,采用 T 型加减速
  当 Ts > 0 时,采用 S 加减速,但必须保证 Ta/2 >Ts。
- **PATH# xx** 定义寄存器 2（以 Fn_107 为例）

例：Fn_107 结构：



序号	取值	含义
A	0	不允许打断上一条正在执行的路径
	1	允许打断上一条正在执行的路径
B	当 Type=3 时，当前路径执行完成，自动跳跃到 AutoLoadIndex 指定的路径	
C	0	位置命令终点，直接指定为“**PATH# xx** 位置坐标寄存器”的指定值
	1	位置命令终点由上一次命令终点，加上指定的增加量
	2	位置命令终点由目前位置反馈，加上指定的增加量

2.6.2 配置流程

- 相关参数：

参数编号	Modbus 地址	参数名称	参数内容	设定范围	出厂设定
Fn_20A	0x020A	路径触发信号源选择输入	0：IO； 1：RS485，RS232；	0-1	1
Fn_20B	0x020B	路径序号	当 Fn_20A = 1, Fn_20B 激活	1~32	0
Fn_20C	0x020C	内部位置路径触发信号输入	置 1 表示电机将按着定义的路径执行	0-1	0
Fn_20D	0x020D	内部位置回零路径触发信号输入	置 1 表示电机将按着定义的速度，方式执行回零点	0-1	0
Fn_20E	0x020E	停止当前路径（PATH# n）执行	0：正常运行； 1：停止路径执行或回零动作；	0-1	0
Dn_1C	0x101C	路径执行状态显示	Dn_1C =1 路径执行中； Dn_1C =0 路径执行完成或没有路径执行。	-	-

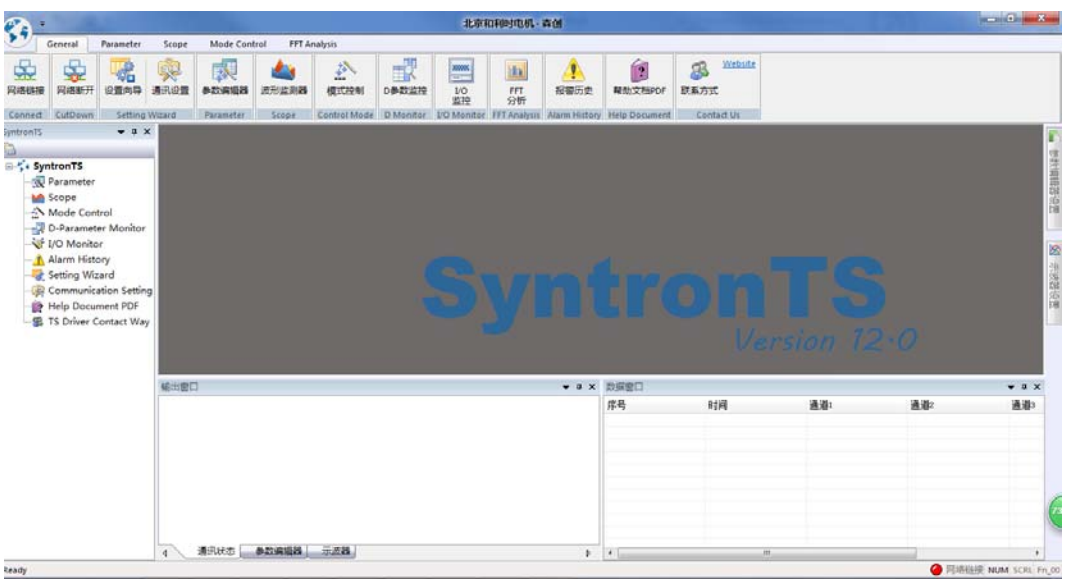
- 路径执行步骤：
  1、设置 Fn_00=14

2、设置 Fn_20A=1
3、设置 Fn_20B 为需要执行的路径号，这里以第一段路径为例，则 Fn_20B=1
4、设置第一段路径的相关参数：Fn_100~Fn_107
5、设置 Fn_20C = 0
6、使能（可通过外部 IO 或 RS232，RS485 通讯修改 Fn38=1 实现）
7、设置 Fn_20C = 1，电机将按着定义的路径执行
8、设置 Fn_20C = 0，为下次作准备，
9、如果要重复执行此路径，则重复第 7 步，第 8 步即可

- 回零例程：
  1、设置 Fn_00=14
  2、设置 Fn_20A=1
  3、设置 Fn_20D=0
  4、使能（可通过外部 IO 或 RS232，RS485 通讯修改 Fn38=1 实现）
  5、设置 Fn_20D=1，电机将按着定义的速度，方式执行回零点动作
  6、设置 Fn_20D=0。
- 停止寄存器：
  Fn_20E=0 正常运行
  Fn_20E=1 停止路径执行或回零动作

2.7 上位机软件

用户可以通过和利时电机提供的上位机软件 SyntronTS.exe 实现参数读写、状态观测等功能，软件界面如下：



配套有《SyntronTS 通讯软件简单使用说明.pdf》，用户可通过业务员或技术人员索取。

第三章 CANopen

3.1 协议介绍

CAN 是控制器局域网络(Controller Area Network, CAN)的简称，是由德国 BOSCH 公司开发，并最终成为应用广泛的国际标准（ISO 11898）。而 CANopen 是一种架构在 CAN 上的高层通讯协定，它由非营利组织 CiA（CAN in Automaion）进行标准的起草及审核工作。基本的 CANopen 设备及通讯子协定定义在 CiA draft standard 301 中。以 CiA 301 为基础再扩充了针对运动控制的子协定 CiAdraft standard 402。

CAN 总线由 2 根线（CAN-H 和 CAN-L）组成，结构简单，并且内部集成了错误探测和管理模块；采用了多主竞争式总线结构，可在各节点之间实现自由通信。

TS 伺服驱动器在 CANopen 总线网络（参考“CIA 301”和“CIA 402”）中作为从站使用，参数功能通过使用对象字典“制造商指定数据”区实现，所有的参数、参数值和功能都是通过 Index 和 Sub-Index 组成的地址来访问和存取。

理论上可以有 CAN 节点可以达到 127 个，而实际中最大节点数取决于 CAN 收发器的性能。CAN 总线的长度取决于通讯速度具体如下：

通讯波特率	CAN 总线最大长度
1Mbps	25m
500Kbps	100m
250Kbps	250m
125Kbps	500m
100Kbps	600m
50Kbps	1000m

用户通过 CAN 总线控制 TS 伺服驱动器，以下为常见测试及控制模式：

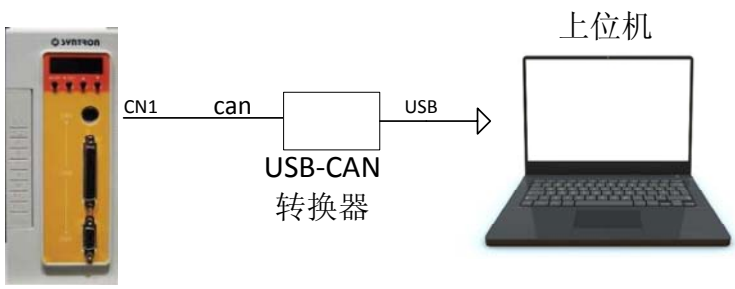


图 3-1 上位机 CAN 测试连接

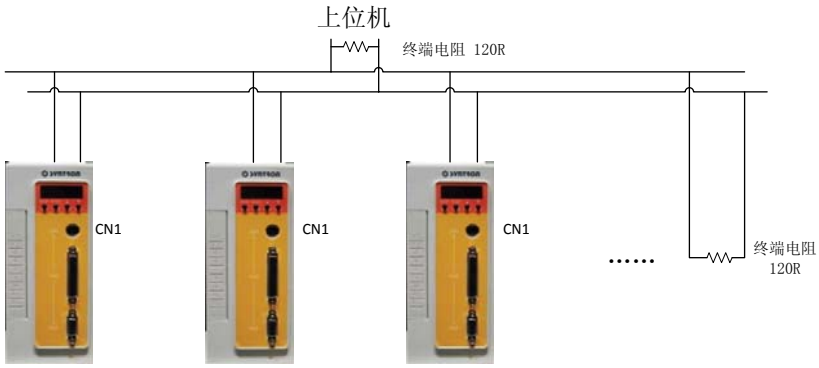


图 3-2 CAN 总线组网示例

3.2 关键参数

TS 驱动器 CAN 总线默认 ID 为 1，波特率为 500Kbps，用户可按需求配置。

参数编号	CAN INDEX	参数名称	参数内容	设定范围	出厂设定
Fn_00	0x2000	控制模式	0：外部速度；1：内部速度；2：位置模式； 3：JOG 模式；4：转矩模式；5-8：混合控制模式 20：CANopen 模式	0-20	2
Fn_9A	0x209A	CAN 通讯 ID	0 为广播地址，1~127 为站址编号	0~127	1
Fn_9C	0x209C	CAN 通讯波特率	CAN 总线波特率，单位为 Kbps	100-1000	500
Dn_12	0x3012	报警信息码	见附录报警信息		

3.3 CANopen 对象字典

对象字典（OD：Object Dictionary）是 CANopen 最重要的特性，它将设备的描述标准化。对象字典中的每一个对象都由 16 位的 Index 和 8 位的 Sub Index 来寻址。一个节点的对象字典 Index 范围在 0x1000 到 0x9FFF 之间。

TS 驱动器和 Fn 和 Dn 参数分别在制造商制定数据区域内，其中 Fn 参数起始地址为 0x2000，对应 Fn00；Dn 状态观测参数起始地址为 0x3000，对应 Dn00。

标准设备子协议区包含了一类设备的所有数据对象。它们都可以通过 CAN 网络读写以达到控制的目的。例如对象字典 6041H，在 DS402 中定义如下：

Bit15-14	Bit13-12	Bit11	Bit10	Bit9	Bit8	Bit7
Manufacturer Specific	Operation Mode Specific	Internal Limit Active	Target Reached	Remote	Manufacturer Specific	Warning
Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Switch On Disabled	Quick Stop	Voltage Disabled	Fault	Operation Enabled	Switch On	Ready to Switch On

在上电初始状态下，6041H 状态如下（0x00 50）：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	0	0	0	0	0	0	0	0	1	0	1	0	0	0	0

在控制字 6040H 写完 6 后，6041H 状态如下（0x00 31）：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0	1

在控制字 6040H 写完 7 后，6041H 状态如下（0x00 33）：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	0	0	0	0	0	0	0	0	0	1	1	0	0	1	1

在控制字 6040H 写完 F 后，6041H 状态如下（0x00 37）：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	0	0	0	0	0	0	0	0	0	1	1	0	1	1	1

驱动报警后，6041H 状态如下（0x00 88）：

Bit	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
M	0	0	0	0	0	0	0	0	1	0	0	0	1	0	0	0

对象字典的结构如下表所示：

Index（hex）	内容
0000	Notused
0001-001F	静态数据类型（标准数据类型，如 Boolean，Integer 16）
0020-003F	复杂数据类型（预定义由简单类型组合成的结构如 PDOCommPar，SDOParameter）
0040-005F	制造商定义的复杂数据类型
0060-007F	设备子协议定义的静态数据类型
0080-009F	设备子协议定义的复杂数据类型
00A0-0FFF	保留
1000-1FFF	通讯子协议区域（如设备类型，错误寄存器，支持的 PDO 数量）
2000-5FFF	制造商指定数据区域
6000-9FFF	标准设备子协议区域(例如“DSP-402 驱动和运动控制子协议”等)
A000-BFFF	标准接口子协议区域
C000-FFFF	保留

3.4 CANopen 数据帧和标识符

CANOpen 协议的通讯对象主要利用了 CAN 协议中的数据帧和远程帧。为了区分不同的通讯对象，CANOpen 协议利用数据帧/远程帧中的 ID。其中第 7 位到第 10 位为功能代码。第 0 位到第 6 位为节点 ID，用以区分不同节点的相同功能。这样就允许最多 127 个从节点与主节点通讯。CANopen 协议数据帧结构如下：

帧头	仲裁域		控制域	数据域	校验域	应答域	帧尾
	COB-ID（通讯对象标识符）	RTR（远程请求）					
1 位	11 或 29 位	1 位	6 位	0～8 字节	16 位	2 位	7 位

RTR 为远程帧标识，置为 0 表示该报文为数据帧，置为 1 表示该报文为远程帧。

CANopen 通信对象 ID(COB-ID)结构如下：

COB-ID										
功能码				节点 ID(NODE ID)						
10	9	8	7	6	5	4	3	2	1	0

每个 CANopen 数据帧都以 COB-ID 开头，COB-ID 用作 CAN 帧的标识符。在配置阶段，接受帧的 COB-ID 的节点是帧的提供者或使用者。COB_ID 越小报文优先级别越高。下面是预定义 CAN 标识符分配表：

CANopen 预定义主/从连接集的广播对象			
对象	功能码（ID-bits 10-7）	COB-ID	通讯参数在 OD 中的索引
NMT Module Control	0	000H	-
SYNC	1	080H	1005H，1006H，1007H
TIME SSTAMP	10	100H	1012H，1013H
CANopen 主/从连接集的对等对象			
对象	功能码功能码（ID-bits 10-7）	COB-ID	通讯参数在 OD 中的索引
EMCY	1	081H-0FFH	1014H，1015H
PDO1(发送)	11	181H-1FFH	1800H
PDO1(接收)	100	201H-27FH	1400H
PDO2(发送)	101	281H-2FFH	1801H
PDO2(接收)	110	301H-37FH	1401H
PDO3(发送)	111	381H-3FFH	1802H
PDO3(接收)	1000	401H-47FH	1402H
PDO4(发送)	1001	481H-4FFH	1803H
PDO4(接收)	1010	501H-57FH	1403H
SDO(发送/服务器)	1011	581H-5FFH	1200H
SDO(接收/客户)	1100	601H-67FH	1200H
Heartbeat(心跳报文)	1110	701H-77FH	1016H，1017H

3.5 CANopen 报文

CANopen 通讯模型定义了如下几种报文（通讯对象）：

SDO	Service Data Object	用于非时间关键数据，如驱动器参数的设置
PDO	Porcess Data Objec	快速过程数据交互（如：实际位置）
EMCY（紧急对象）	Emergency Message	故障信息传输
SYNC（同步对象）	Synchronization Message	多个 CAN 节点的同步
NMT（网络管理）	Network Management	网络服务：例如，可以同时激活所有的 CAN 节点。
Heartbeat（心跳报文）	Node Guarding	通过信息的规范监控通讯参与者。

3.5.1 NMT(Network Management 管理报文)

NMT 管理报文用来实现 CANopen 网络中主节点对从节点监控和管理，如初始化、启动和停止节点，侦测失效节点。这种服务主要采用主从通讯模式来实现，此消息不需要应答。只有主节点能够发送 NMT Modual Control 报文。

- NMT 消息格式

COB-ID	Byte0	Byte1
0x000	命令字	Node-ID

COB-ID 为 0x000，Byte0 为命令字，占一个字节，具体格式如下：

命令字	NMT 服务	
1(0x01)	Start Remote Node	节点置为可操作状态
2(0x02)	Stop Remote Node	节点置为停止状态
128(0x80)	Enter Pre-operational State	节点置为预操作状态
129(0x81)	Reset Node	节点置为应用重置状态
130(0x82)	Reset Communication	节点置为通讯重置状态

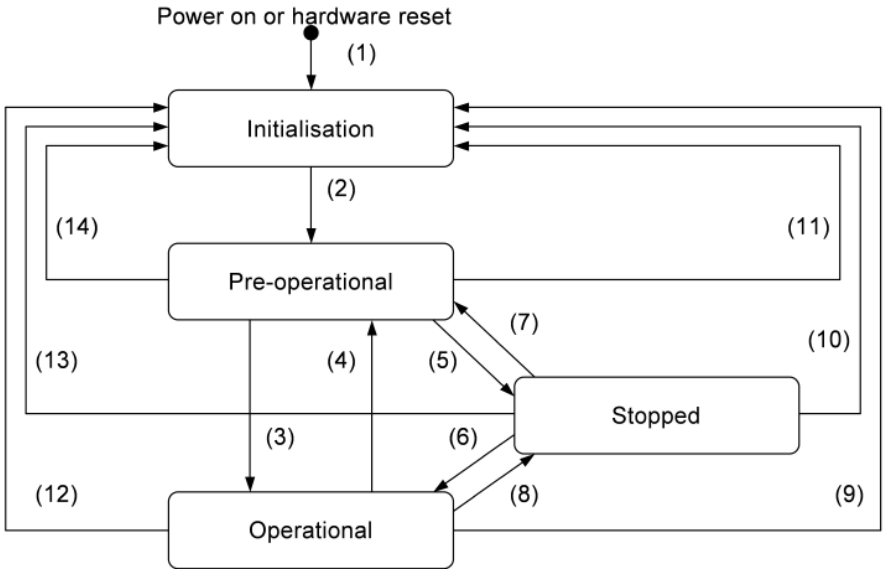
Byte 1 为 CANopen 网络设备节点地址，0 表示广播。

例：000h 01h 06h——将节点 6 置为可操作状态；

000h 80h 00h——将所有节点置为预操作状态；

- NMT 节点保护（NMT Node Guarding）

通过节点保护服务，MNT 主节点可以检查每个节点的当前状态，当这些节点没有数据传送时这种服务尤其有意义。NMT 状态机如下。



Transition	Description
(1)	At Power on, the initialisation state is entered autonomously
(2)	Once initialisation is finished, the Pre-Operational state is automatically entered
(3), (6)	Start_Remote_Node
(4), (7)	Enter_Pre-Operational_State
(5), (8)	Stop_Remote_Node
(9), (10), (11)	Reset_Node
(12), (13), (14)	Reset_Communication

3.5.2 EMCY（Emergency Message 紧急报文）

驱动器在故障报警时，会主动上传错误代码，格式如下：

COB-ID	Byte0	Byte1
0x080+Node_ID	错误码低位	错误码高位

注：错误码见**第四章 报警信息**。

3.5.3 SDO（Service Data Object 服务数据对象）

SDO 主要用来访问节点的对象字典（OD）。其中被访问的对象字典的所在设备作为 Server，访问对象字典的设备作为 Client。SDO 用来在设备之间传输低优先级的对象，典型是对从设备进行配置、管理，比如修改速度环的 PID 参数、PDO 配置参数等。

SDO 报文结构如下：

	COB_ID	BYTE0	BYTE1	BYTE2	BYTE3	BYTE4	BYTE5	BYTE6	BYTE7
		CMD	INDEX		SUB INDEX	DATA			
问	0x600+ Node_ID	0x23 写 4byte	LEB+MSB（0xFn 0x20 Fn 参数 0xDn 0x30 Dn 参数 CANopen 对象字典 Index）		0x00	DATA1	DATA2	DATA3	DATA4
		0x2B 写 2byte				DATA1	DATA2	-	-
		0x2F 写 1byte				DATA1	-	-	-
		0x40 读数据				0x00	0x00	0x00	0x00
答	0x580+ Node_ID	0x43 回 4byte				DATA1	DATA2	DATA3	DATA4
		0x4B 回 2byte				DATA1	DATA2	-	-
		0x4F 回 1byte				DATA1	-	-	-
		0x60 写寄存器回复				DATA1	DATA2	DATA3	DATA4
		0x80 错误码				DATA1	DATA2	DATA3	DATA4

例 1：读速度 Dn_00

601 40 00 30 00 00 00 00 00 // 读当前速度

581 4B 00 30 00 64 00 00 00 // 转速为 100rpm

例 2：写内部速度 Fn_33

601 2B 33 20 00 C8 00 00 00 // Fn_33<--200RPM

581 60 33 20 00 C8 00 00 00 // 回复

3.5.4 PDO（Process Data Object 过程数据对象）

PDO 是用来发送（TPDO）或者接收（RPDO）数据的，从而实现实时数据的传输。数据从一个生产者传到一个或多个消费者，1 个 PDO1 次最多传输 8 个字节的数据。以下为 PDO 的对象字典参数和例程：

映射参数 Index		通信参数 Index	
RPDO	RPDO	TPDO	TPDO
1600h	1400h	1800h	1A00h

- RPDO配置

例如，以下RPDO1和RPDO2的配置，若CANopen节点号设为2：

	序号	报文内容	解析
RPDO1 0x202	1	602 2F 00 16 00 00 00 00 00	RPDO1 stop
	2	602 23 00 16 01 10 00 40 60	6040h映射至RPDO1
	3	602 2F 00 16 00 01 00 00 00	RPDO1 enable
RPDO2 0x302	1	602 2F 01 16 00 00 00 00 00	RPDO2 stop
	2	602 23 01 16 01 20 00 83 60	6083h映射至RPDO2
	3	602 23 01 16 02 20 00 84 60	6084h映射至RPDO2
	4	602 2F 01 16 00 02 00 00 00	RPDO2 enable

- TPDO配置

TPDO有异步触发、同步触发两种触发方式，可以通过SDO访问对象字典的方式来修改。1-240方式为同步触发（响应同步帧）；254表示定时触发（可设时间）；255方式表示事件触发（状态变化上传）。

例如，以下TPDO1和TPDO2的配置，若CANopen节点号设为2：

	序号	报文内容	解析	
TPDO1 0x182	1	602 2F 00 1A 00 00 00 00 00	TPDO1 stop	本例是将对象字典6041H设为TPDO1，让其以50ms的频率向总线上报；上报方式为254；
	2	602 23 00 1A 01 10 00 41 60	6041h	
	3	602 2F 00 1A 00 01 00 00 00	TPDO1 enable	
	4	602 2F 00 18 05 90 01 00 00	设置触发时间50ms	
	5	602 2F 00 18 03 90 01 00 00		
	6	602 2F 00 18 02 FE 00 00 00	254上报方式	
TPDO2 0x282	1	602 2F 01 1A 00 00 00 00 00	TPDO2 stop	本例是将对象字典6064H和606CH设为TPDO2，让其以50ms的频率同时向总线上报；上报方式为254；
	2	602 23 01 1A 01 20 00 64 60	6064h	
	3	602 23 01 1A 02 20 00 6C 60	606Ch	
	4	602 2F 01 1A 00 02 00 00 00	TPDO2 enable	
	5	602 2F 01 18 05 90 01 00 00	设置触发时间50ms	
	6	602 2F 01 18 03 90 01 00 00		
	7	602 2F 01 18 02 FE 00 00 00	254上报方式	

3.5.5 NMT 错误控制

➢ 心跳报文（Heartbeat）

CANopen 节点会以一个固定的频率发送心跳报文，用于告诉主机自己的状态。心跳报文的格式：COB-ID 为 0x700+ ID，数据为一字节的状态数据。例如 ID 为 1 的驱动器在接入 CAN 总线后会主动发送报文：701 00。

心跳间隔为字典 0x1017 设置值(单位 ms)，设置 0x1017 为 0，则禁止心跳报文，设置 0x1017 为非 0，则使能心跳报文。

例 1：601 2B 17 10 00 00 00 00 00 //禁止心跳报文

例 2：601 2B 17 10 00 64 00 00 00 //使能心跳报文，心跳间隔 100ms

COB-ID	Byte0	状态含义
0x700+Node_ID	0x00	BOOTUP 启动状态
	0x04	STOPPED 停止
	0x05	OPERATIONAL 可操作
	0x7F	PRE-OPERATIONAL 预操作

3.6 CANopen 配置步骤

3.6.1 速度模式

驱动器可以接收主站速度指令运行，以下为相关对象字典及配置步骤：

■ **相关对象字典：**

Index	Name	Type	Attr.	unit
6040h	Controlword	UNSIGNED16	RW	
6041h	Statusword	UNSIGNED16	RO	
6060h	Modes of operation	INTEGER8	RW	
6061h	Modes of operation display	INTEGER8	RO	
6069 h	velocity_sensor_actual_value	INT32	RO	
606B h	velocity_demand_value	INT32	RO	0.1rpm
606C h	velocity_actual_value	INT32	RO	1rpm
6083 h	profile_acceleration	UINT32	RW	millisecond from 0rpm to 3000rpm
6084 h	profile_deceleration	UINT32	RW	millisecond from 0rpm to 3000rpm
606D h	velocity_window	UINT16	RW	0.1rpm
606E h	velocity_window_time	UINT16	RW	ms
606F h	velocity_threshold	UINT16	RW	0.1rpm
6070 h	velocity_threshold_time	UINT16	RW	ms
60FF h	target_velocity	INT32	RW	0.1rpm

■ **配置步骤（以CAN节点2为例）：**

STEP1. 确认驱动器参数：Fn_00 = 20（CANopen模式）

STEP2. 节点置为可操作状态:

000 01 02——将节点2置为可操作状态；

STEP3. RPDO 映射:

以下示例配置3个RPDO，0x6040映射至RPDO1，0x6083和0x6084映射至RPDO2，0x60FF映射至RPDO3：

RPDO1：RPDO1，对应COB-ID为202

602 2F 00 16 00 00 00 00 00 //RPDO1 stop

602 23 00 16 01 10 00 40 60 //6040h

602 2F 00 16 00 01 00 00 00 // RPDO1 enable

RPDO2：RPDO2，对应COB-ID为302

602 2F 01 16 00 00 00 00 00 //RPDO2 stop

602 23 01 16 01 20 00 83 60 //6083h and 6084h

602 23 01 16 02 20 00 84 60 //6083h and 6084h

602 2F 01 16 00 02 00 00 00 // RPDO2 enable

RPDO3：RPDO3，对应COB-ID为402

602 2F 02 16 00 00 00 00 00 //RPDO3 stop

602 23 02 16 01 20 00 FF 60 //60FFh

602 2F 02 16 00 01 00 00 00 // RPDO3 enable

STEP4. 设置

1. 设置运行模式【Mode of operations:6060h】为速度模式（Profile Velocity Mode）：

602 2F 60 60 00 03 00 00 00//设置6060h为3（速度控制为PV）

2. 设置加速度【Profile acceleration:6083h】(单位时间：ms从 0rpm 到 3000rpm)；设置减速度【Profile deceleration:6084h】(单位时间：ms 从0rpm到3000rpm)：

302 E8 03 00 00 E8 03 00 00 //设置加减速为1s

3. 设置目标速度【Target velocity:60FFh】（单位：0.1rpm）：

402 64 00 00 00

4. 设置控制字【Controlword:6040h】来使能电机：

(如果伺服已经使能，收到速度指令后会立刻动作)

602 2B 40 60 00 06 00 00 00//设置6040h为6

602 2B 40 60 00 07 00 00 00 //设置6040h为7

602 2B 40 60 00 0F 00 00 00 //设置6040h为F，使电机使能；

5. 确认状态字【Statusword:6041h】得到伺服状态。.

■ **注意事项：**

1. 速度模式下，主站可以得到以下信息：
   - 确认【Velocity demand value:606Bh】得到内部速度指令值。(单位: 0.1rpm)
   - 确认【Velocity_sensor_actual_value:6069h】得到实际速度。(单位: 0.1rpm)
2. 主站可以设置速度监控阈值：
   - 设置【Velocity window:606Dh】分配速度到达区。(单位: 0.1rpm)
   - 设置【Velocity widnow time:606Eh】速度到达的时间。 (单位:ms)
   - 设置【Velocity threshold:606Fh】分配零速阈值。 (单位: 0.1rpm)
   - 设置【Velocity widnow time:6070h】零速时间。 (单位:ms)

3.6.2 位置模式

驱动器接收主站位置指令可控制电机到达目标位置。

■ **相关对象字典：**

Index	Name	Type	Attr.	unit
6040h	Controlword	UNSIGNED16	RW	
6041h	Statusword	UNSIGNED16	RO	
6060h	Modes of operation	INTEGER8	RW	
6061h	Modes of operation display	INTEGER8	RO	
6062h	Position demand value	INTEGER32	RO	pulse
6063h	Position actual value*	INTEGER32	RO	increments
6064h	Position actual value	INTEGER32	RO	
6065h	Following error window	UNSIGNED32	RW	pulse
6066h	Following error time out	UNSIGNED16	RW	ms
6067h	Position window	UNSIGNED32	RW	pulse
6068h	Position window time	UNSIGNED16	RW	ms
607Ah	Target position	INTEGER32	RW	pulse
6081h	Profile velocity	UNSIGNED32	RW	0.1rpm
6083h	Profile acceleration	UNSIGNED32	RW	millisecond from 0rpm to 3000rpm
6084h	Profile deceleration	UNSIGNED32	RW	millisecond from 0rpm to 3000rpm
6093h	Position factor	UNSIGNED32	RW	
60F4h	Following error actual value	INTEGER32	RO	pulse
60FCh	Position demand value	INTEGER32	RO	

■ **配置步骤（以CAN节点2为例）：**

STEP1. 确认驱动器参数：Fn_00 = 20（CANopen模式）

STEP2. 节点置为可操作状态:

000 01 02——将节点2置为可操作状态；

STEP3. RPDO 映射:

配置4个RPDO:0x6040映射至RPDO1，0x6083和0x6084映射至RPDO2，0x607A映射至RPDO3，0x6081映射至RPDO4。

RPDO1：RPDO1，对应COB-ID为202

602 2F 00 16 00 00 00 00 00 //RPDO1 stop
602 23 00 16 01 10 00 40 60 //6040h
602 2F 00 16 00 01 00 00 00 // RPDO1 enable

RPDO2：RPDO2，对应COB-ID为302

602 2F 01 16 00 00 00 00 00 //RPDO2 stop
602 23 01 16 01 20 00 83 60 //6083h and 6084h
602 23 01 16 02 20 00 84 60 //6083h and 6084h
602 2F 01 16 00 02 00 00 00 // RPDO2 enable

RPDO3：RPDO3，对应COB-ID为402

602 2F 02 16 00 00 00 00 00 //RPDO3 stop
602 23 02 16 01 20 00 7A 60 //607Ah
602 2F 02 16 00 01 00 00 00 // RPDO3 enable

RPDO4：RPDO4，对应COB-ID为502

602 2F 03 16 00 00 00 00 00 //RPDO4 stop
602 23 03 16 01 20 00 81 60 //6081h
602 2F 03 16 00 01 00 00 00 // RPDO4 enable

STEP4. 设置

1. 设置运行模式【Mode of operations:6060h】为位置模式.
   602 2F 60 60 00 01 00 00 00 //设置6060h为1（profile position :PP）
2. 设置目标位置【Target position:607Ah】. (单位: pulse)
   402 60 EA 00 00 //60000脉冲
3. 设置速度【Profile velocity:6081h】. (单位: 0.1rpm)
   502 B8 0B 00 00 //300rpm
4. 设置加速度【Profile acceleration:6083h】和减速度【Profile deceleration:6084h】(单位时间：ms 从 0rpm 到 3000rpm)：
   302 E8 03 00 00 E8 03 00 00
5. 设置电子齿轮比【Position factor -Numerator:6093h-sub1】and【Position factor -Divisor:6093h-sub2】
   602 2B 93 60 01 01 00 00 00 //设置电子齿轮比分子
   602 2B 93 60 02 01 00 00 00 //设置电子齿轮比分母
6. 设置控制字【Controlword:6040h】来使能电机：.

602 2B 40 60 00 06 00 00 00 //设置6040h为6
602 2B 40 60 00 07 00 00 00 //设置6040h为7
602 2B 40 60 00 0F 00 00 00 //设置6040h为F，使电机使能；

7. 触发电机运行：
   202 0F 00 //6040的bit4先清零；
   202 5F 00 //6040的bit6设置为1电机运行于增量位置模式下，bit4置为1电机运动
   202 4F 00 //6040的bit4清零，为下次启动做准备；
8. 确认状态字【Statusword:6063h】得到电机位置反馈.
9. 确认状态字【Statusword:6041h】得到驱动器状态，例如错误位、目标到达等。

■ **注意事项：**

1、驱动器是以6040的bit4的↑（上升沿）接收新的位置命令，所以每次执行完一次运行后需要把此位清零。上位机应根据驱动器的状态字6041的bit12来判断是否要给伺服新数据。当驱动器的状态字6041的bit12为0时，表示驱动器可以接收新的位置数据和命令了，如果为1，即使驱动器有接收数据，但是命令是不被采纳的，而会保存在缓存区。

2、绝对方式下，需要不断更新位置数据。

3、驱动器最大可缓存 32 条位置触发信息（每条触发指令含加减速、目标脉冲和目标速度四个对象字典内容），超过缓存数的位置指令将丢失，所以请勿高速更新位置指令。

3.6.3 转矩模式

驱动器支持转矩运行，以下为相关对象字典和配置：

■ **相关对象字典：**

Index	Name	Type	Attr.	unit
6040h	Controlword	UNSIGNED16	RW	
6041h	Statusword	UNSIGNED16	RO	
6060h	Modes of operation	INTEGER8	RW	
6061h	Modes of operation display	INTEGER8	RO	
6071h	Target torque	INTEGER16	RW	0.1%Torque
6087h	Torque slope	INTEGER32	RW	

■ **配置步骤（以CAN节点2为例）：**

STEP1. 确认驱动器参数：Fn_00 = 20（CANopen模式）

STEP2. 节点置为可操作状态:

000 01 02——将节点2置为可操作状态；

STEP3. RPDO 映射此例暂不配置

STEP4. 设置

1. 设置运行模式【Mode of operations:6060h】为转矩模式：
   602 2F 60 60 00 03 00 00 00//设置6060h为4（PT转矩模式）
2. 设置加减速【Torque slope:6087h】：
   602 2F 87 60 00 E8 03 00 00//设置6087H为1000；
3. 设置目标转矩【Target torque:6071h】：
   602 2F 71 60 00 64 00 00 00//设置6071H为100，则为电机转矩的10%；
4. 设置状态字【Controlword:6040h】来使能电机：
   602 2B 40 60 00 06 00 00 00//设置6040h为6
   602 2B 40 60 00 07 00 00 00 //设置6040h为7
   602 2B 40 60 00 0F 00 00 00 //设置6040h为F，使电机使能；
5. 确认状态字【Statusword:6041h】得到驱动器状态。

■ **注意事项：**

转矩模式下的最大限制速度为 Fn95，地址为 0x2095。

3.6.4 回零点模式

用户可设置速度、加减速等配置回零模式，关键对象字典及设置如下：

■ **相关对象字典：**

Index	Name	Type	Attr.	unit
6040h	Controlword	UNSIGNED16	RW	
6041h	Statusword	UNSIGNED16	RO	
6060h	Modes of operation	INTEGER8	RW	
6061h	Modes of operation display	INTEGER8	RO	
607Ch	Home offset	INTEGER32	RW	
6093h	Position factor	UNSIGNED32	RW	
6098h	Homing method	INTEGER8	RW	
6099h	Homing speeds	ARRAY	RW	0.1rpm
609Ah	Homing acceleration	UNSIGNED32	RW	millisecond from 0rpm to 3000rpm

■ **配置步骤（以CAN节点2为例）：**

STEP1. 确认驱动器参数：Fn_00 = 20（CANopen模式），Fn99=1（通讯修改参数不保存）.

STEP2. 节点置为可操作状态:

000 01 02——将节点2置为可操作状态；

STEP3、设置

1. 设置运行模式【Mode of operations:6060h】为回原点模式：
   602 2F 60 60 00 06 00 00 00//设置6060h为6（回原点模式）
2. 设置回零方式【Homing method:6098h】：

602 2F 98 60 00 14 00 00 00//设置6098H为20

3. 设置回零速度【Homing speeds:6099h】：

602 23 99 60 01 D0 07 00 00

602 23 99 60 02 C8 00 00 00

4. 设置回零加减速【Homing acceleration:609Ah】：

602 23 9A 60 00 E8 03 00 00

5. 设置电子齿轮比【Position factor:6093h】：

602 23 93 60 01 01 00 00 00 //设置6093H，齿轮比分子1；

602 23 93 60 02 01 00 00 00 //设置6093H，齿轮比分母1；

6. 设置控制字【Controlword:6040h】来使能电机：.

602 2B 40 60 00 06 00 00 00 //设置6040h为6

602 2B 40 60 00 07 00 00 00 //设置6040h为7

602 2B 40 60 00 0F 00 00 00 //设置6040h为F，使电机使能；

7. 触发电机运行：

602 2B 40 60 00 0F 00 00 00

602 2B 40 60 00 5F 00 00 00

602 2B 40 60 00 4F 00 00 00

■ **注意事项：**

驱动器支持13种回零方式,对象字典0x6098的不同回零方式定义如下：

Index	Sub-Index	设定值	回零方式
0x6098	0x00	-1	反向回零,找Z,转矩到达信号为零位信号
		-2	正向回零,找Z,转矩到达信号为零位信号
		-17	反向回零,不找Z,转矩到达信号为零位信号
		-18	正向回零,不找Z,转矩到达信号为零位信号
		1	反向回零,找Z,负限位信号为零位信号
		2	正向回零,找Z,正限位信号为零位信号
		3	反向回零,找Z,机械零点信号为零位信号
		4	正向回零,找Z,机械零点信号为零位信号
		17	反向回零,不找Z,负限位信号为零位信号
		18	正向回零,不找Z,正限位信号为零位信号
		19	反向回零,不找Z,机械零点信号为零位信号
		20	正向回零,不找Z,机械零点信号为零位信号
		35	不回零

第四章 报警信息

驱动器产生报警时，自动通过EMCY向主站报告，用户还可读取Dn12来获知报警信息。报警信息定义如下：

驱动器显示	传输码	含义
ERR_no	0x1516	不可识别故障
ERR_oc	0x160C	过流
ERR_oL	0x1614	过载
ERR_oH	0x1611	放电回路频繁动作平均功率大
ERR_LH	0x1411	放电报警瞬时功率大
ERR_oU	0x161C	过压
ERR_PA	0x170A	参数故障
ERR_IC	0x120C	电流采样（中点）故障
ERR_EC	0x0E0C	编码器故障 ABZ 报警
ERR_PE	0x170E	位置超差
ERR_St	0x1A1B	失速
ERR_LU	0x141C	欠压
ERR_EH	0x0E11	编码器故障 UVW 报警
ERR_Ao	0x0A16	缺相保护
ERR_FE	0x0F0E	FPGA 配置失败
ERR_Id	0x120D	输入口功能定义重复
ERR_dE	0x0D0E	功率母线未准备好
ERR_SL	0x1A14	速度指令太小
ERR_Fd	0x0F0D	FPGA 死机
ERR_cE	0x0C0E	通讯错误
ERR_bE	0x0B0E	协同模式报警
ERR_Lo	0x1416	CAN 总线断线报警

另外，如果驱动器使能正反限位功能，当到达正反限位时，也会主动通过 EMCY 向主站报告。状态信息定义如下：

状态信息		
CCW_Disable	0x8681	正限位
CW_Disable	0x8682	负限位
POS_Arrived	0x8683	位置到达