SYNTRON 森创

DS / RS 系列伺服驱动器

——— 通讯手册

版本 V8.3

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。

和利时电机具有本产品及其软件的专利权、版权和其它知识产权。未经授权，不得直接或者间接地复制、制造、加工、使用本产品及其相关部分。

和利时电机具有本使用说明书的著作权，未经许可，不得修改、复制使用说明书的全部或部分内容。

目录

第一章 产品概况 ..... - 1 -

1.1 概述 ..... - 1 -
1.2 通讯端口 ..... - 1 -
1.3 注意事项 ..... - 2 -
1.4 参数修改 ..... - 3 -
1.4.1 上位机修改 ..... - 3 -
1.4.2 通讯修改（单轴） ..... - 3 -
1.4.3 通讯修改（双轴） ..... - 4 -
1.4.4 手持调试板修改 ..... - 5 -

第二章 Modbus 协议 ..... - 6 -

2.1 接线说明 ..... - 6 -
2.2 关键参数（单轴） ..... - 6 -
2.3 关键参数（双轴） ..... - 7 -
2.4 Modbus_RTU 简介 ..... - 7 -
2.4.1 单轴驱动器示例 ..... - 7 -
2.4.2 双轴驱动器示例 ..... - 8 -
2.5 Modbus_ASCII 简介 ..... - 9 -
2.6 Modbus 运动控制配置 ..... - 10 -
2.6.1 速度模式指令（单轴） ..... - 10 -
2.6.2 速度模式指令（双轴） ..... - 11 -
2.7 双轴（二合一）特有协议 ..... - 11 -
2.7.1 双读双写指令 ..... - 11 -
2.7.2 差速模式指令（选配） ..... - 13 -
2.8 上位机软件 SyntronTS ..... - 13 -
2.8.1 接线与设置 ..... - 13 -
2.8.2 参数编辑器 ..... - 17 -
2.8.3 波形检测器 ..... - 21 -
2.8.4 D 参数监控 ..... - 31 -
2.8.5 注意事项 ..... - 31 -

第三章 CANopen ..... - 33 -

3.1 协议介绍 ..... - 33 -
3.2 关键参数（单轴） ..... - 34 -
3.3 关键参数（双轴） ..... - 35 -
3.4 CANopen 对象字典 ..... - 35 -

目录

3.4.1 控制字 6040H ..... - 36 -
3.4.2 状态字 6041H ..... - 36 -
3.4.3 对象字典列表 ..... - 37 -
3.5 CANopen 数据帧和标识符 ..... - 38 -
3.6 CANopen 报文简介 ..... - 39 -
3.6.1 NMT（Network Management 管理报文） ..... - 39 -
3.6.2 EMCY（Emergency Message 紧急报文） ..... - 41 -
3.6.3 SDO（Service Data Object 服务数据对象） ..... - 42 -
3.6.4 PDO（Process Data Object 过程数据对象） ..... - 42 -
3.7 CANopen 配置例程 ..... - 44 -
3.7.1 协议速度模式（Profile Velocity Mode） ..... - 44 -
3.7.2 协议位置模式（Profile Position Mode） ..... - 46 -
3.7.3 转矩模式（Torque Profile Mode） ..... - 48 -
3.7.4 回零点模式（Homing Mode） ..... - 49 -
3.8 CAN 自由协议 ..... - 50 -
3.8.1 速度模式（单轴） ..... - 50 -
3.8.2 速度模式（双轴） ..... - 51 -
3.8.3 差速模式（双轴） ..... - 52 -
3.8.4 双读双写（双轴） ..... - 52 -

第四章 报警信息 ..... - 54 -

4.1 单轴报警信息 ..... - 54 -
4.2 双轴报警信息 ..... - 55 -

第一章 产品概况

1.1 概述



图 1-1 DS/RS 系列单轴伺服驱动器



图 1-2 DS/RS 系列双轴（二合一）伺服驱动器

DS / RS 系列低压伺服驱动器支持 Modbus 和 CAN 的总线控制，驱动器通讯接口和支持协议如下：

类型	型号	通讯端口	通讯接口	通讯协议
单轴驱动	DS10240X DS10270X RS10240X RS10270X RS102120X	CN1 / CN2	CAN	CANopen
			RS485	Modbus
双轴驱动	DS20240X DS20270X RS20240X RS20270X	CN3 / CN4	CAN	CANopen / 差速协议
			RS232 / RS485	Modbus / 差速协议

本文为 DS / RS 系列低压伺服驱动器的通讯手册，包括单轴与双轴。

注意：X 为型号后缀，不同标号表示不同版本。

1.2 通讯端口

通讯端口为两个管脚定义相同 RJ45 接口，用于总线级联。用户可通过特制 CAT5 及以上数据网线组成总线网，端口定义见下页。

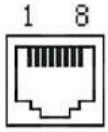


图 1-3 RJ45 的管脚分布

类型	Pin	定义	线色
RS232	1	RS232_RX	橙白
	2	RS232_TX	橙
GND	3	GND	绿白
RS485	4	RS485_A	蓝
	5	RS485_B	蓝白
CAN	7	CAN_L	棕白
	8	CAN_H	棕

注意：单轴驱动器没有 RS232 接口。

1.3 注意事项

- 在驱动器上电的情况下，**请勿热插拔总线接口！否则可能导致损坏！**
- 单轴驱动器组网时，网线请勿全压！ CN1 为总线进，CN2 为总线出，**请勿同时连接两台的 CN1 接口，否则可能导致损坏！**（双轴驱动可全压）
- **请勿将通讯端口内管脚短路！否则可能导致损坏！**
- CAN 和 Modbus 总线构成网络时，请用户自行加装 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，RS485 组网请接 RJ45 接口的 3、4、5 管脚，CAN 组网请接 3、7、8 管脚：

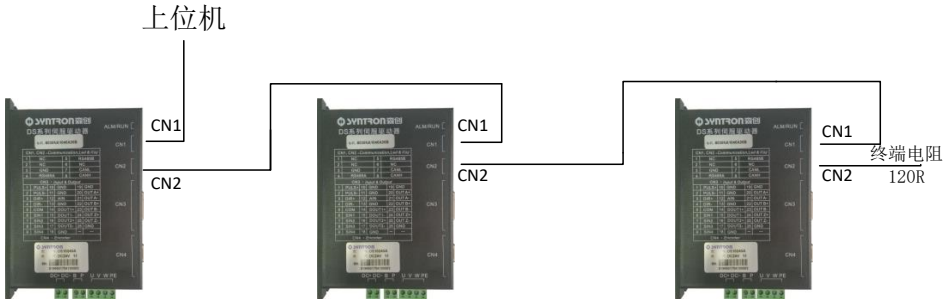


图 1-4 多台单轴驱动组网

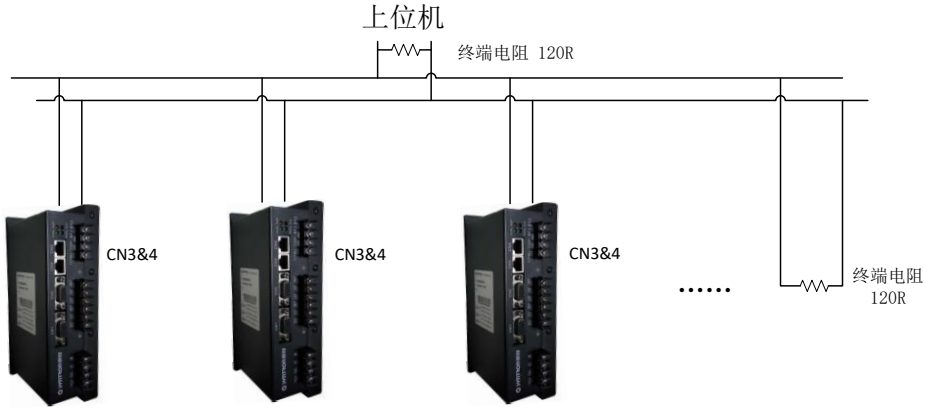


图 1-5 多台双轴驱动组网

1.4 参数修改

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。参数修改有三种方式，分别为上位机修改、通讯指令修改和手持调试板设置。

1.4.1 上位机修改

SyntronTS 上位机可支持和利时 TS / DS / RS 系列伺服驱动器的参数修改，参数修改在上位机参数编辑器完成，如需修改参数，请勾选相应的参数，并选择是否参数断电保存，然后点击写入参数即可。上位机介绍见 2.8 上位机软件 SyntronTS。

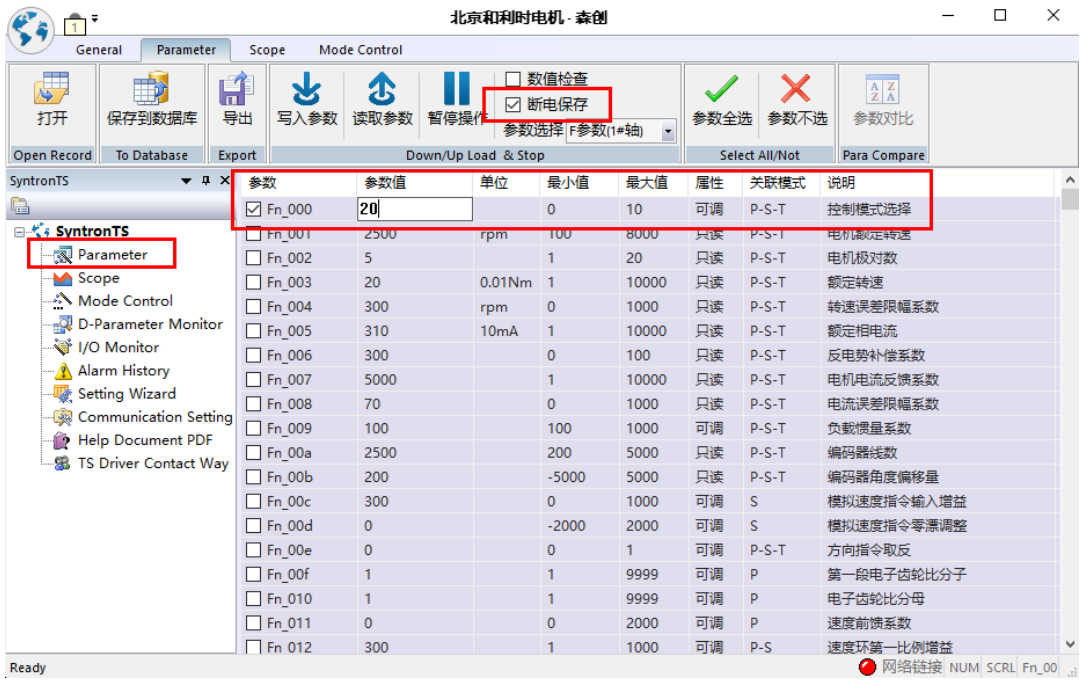


图 1-6 单轴驱动器修改 Fn00 示意

1.4.2 通讯修改（单轴）

如果用通讯修改参数且需要保存进驱动器 Flash，需先修改 Fn99 为 0，再使用通讯更改参数，否则该参数只写入 RAM，不保存驱动器内部 Flash。关键参数如下：

参数编号	CAN INDEX	Modbus Address	参数名称	参数内容	设定范围	出厂设定
Fn3F	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
Fn99	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	1

- Modbus 总线修改

协议详细请参考第二章 Modbus 协议，写单轴驱动 Fn00（控制模式）参数例程如下：

步骤	指令（HEX）	解析
1	01 06 00 99 00 00 59 E5	打开参数保存
2	01 06 00 00 00 14 89 C5	写 Fn00
3	01 06 00 99 00 01 98 25	关闭参数保存
4	重启生效	

伺服设置参数，例如控制模式、站号、波特率等修改完毕需重启生效。而速度、增益等参数则可以即时修改生效，例如写单轴驱动 Fn33（内部速度）：

步骤	指令（HEX）	解析
1	01 06 00 33 00 00 79 C5	写 Fn33 内部速度为 0
2	01 06 00 33 00 64 78 2E	写 Fn33 为 100rpm
3	01 06 00 33 FF 9C 38 5C	写 Fn33 为-100rpm

- CAN 总线修改

协议详细请参考第三章 CANopen，写单轴驱动 Fn33 参数例程如下：

发送：601 2B 33 20 00 C8 00 00 00　// SDO 写入 Fn33 速度 200RPM

回复：581 60 33 20 00 C8 00 00 00　// SDO 回复

如需参数保存，Fn99 写入顺序参考上一小节。

1.4.3 通讯修改（双轴）

如果用通讯修改参数且需要保存进驱动器 Flash，需先修改 Pn5.00A 为 1，再使用通讯更改参数，否则该参数只写入 RAM，不保存驱动器内部 Flash。关键参数见下页：

参数编号	通讯地址	参数名称	参数内容	设定范围	出厂设定
Pn5.00A	0x450A	通讯参数保存设置	0：参数在 RAM 中运行； 1：参数写入 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
Pn6.000	0x4600	密码	某些重要参数需要打开密码才能修改 默认 8847，如需修改重要参数，需设为 8848	0~9999	8847

参数修改例程可参考上一节。

1.4.4 手持调试板修改



图 1-7 手持调试板

手持调试板有修改参数的最高优先级（写入参数无需修改 Fn99）。

单轴 DS / RS 型号驱动器的 CN1 端口、双轴 DS / RS 型号驱动器的 CN5 端口支持外接手持调试板。

用手持调试板可以很方便的调整参数、观测伺服状态。具体使用方法见 DS/RS 使用手册中。

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
功能码选择	Dn 编号	状态观测参数 Dn	
	Fn 编号	功能参数 Fn	
	Pn 编号	共有参数（双轴二合一驱动）	
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 协议

2.1 接线说明

用户可通过 RS232 / 485 读写驱动器参数，也可以通过总线使用驱动器的速度模式进行运动控制，以下为常见接线，以单轴驱动器为例：

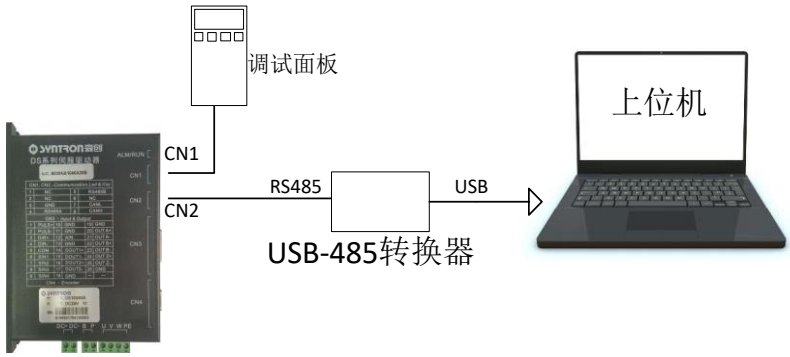


图 2-1 单轴驱动器连接 PC

2.2 关键参数（单轴）

用户用通讯的方式控制驱动器时，必须确认以下驱动器参数。驱动器默认 Modbus 地址为 1，波特率为 115200bps，ModBus_RTU 模式（协议采用 8 个数据位、1 个停止位、无奇偶校验），用户可根据需求设置。。

功能参数（Fn00~Fn17F）对应的寄存器地址为：0x0000~0x017F；

状态参数（Dn00~Dn2F）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
Fn3A	0x003A	Modbus 地址	0 为广播地址，1~127 为站址编号	0~127	1
Fn3C	0x003C	Modbus 通讯波特率	0：2400bps； 1：9600bps； 2：38400bps； 3：57600bps； 4：115200bps； 5：192000bps； 6：512000bps；	0~6	4
Fn3D	0x003D	Modbus 模式选择	101：ModbusRTU； 102：ModbusAS	-	101
Fn99	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 关键参数（双轴）

双轴伺服驱动器默认 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：

1#轴（F0.0.000~F0.6.0FF）对应的寄存器地址为 0x2000~0x26FF；

2#轴（F1.0.000~F1.6.0FF）对应的寄存器地址为 0x3000~0x36FF；

■ 公共参数：

Pn0.000~Pn6.00F 对应的寄存器地址为 0x4000~0x460F。

■ 状态参数：

1#轴（D0.0000~D0.003F）对应的寄存器地址为：0x5000~0x503F；

2#轴（D1.0000~D1.003F）对应的寄存器地址为：0x5100~0x513F。

以下为 Modbus 相关的关键参数：

参数编号	地址	参数名称	参数内容	设定范围	出厂设定
Pn5.000	0x4500	RS232 站号	0 为广播地址，1~255 为站址编号	0~255	1
Pn5.001	0x4501	RS232 通讯波特率	0：2400bps； 1：9600bps； 2：19200bps； 3：38400bps； 4：57600bps； 5：115200bps； 6：512000bps；	0~6	5
Pn5.002	0x4502	RS232 模式选择	0：ModbusRTU； 1：ModbusAS		0
Pn5.003	0x4503	RS485 站号	0 为广播地址，1~255 为站址编号	0~255	1
Pn5.004	0x4504	RS485 通讯波特率	0：2400bps； 1：9600bps； 2：19200bps； 3：38400bps； 4：57600bps； 5：115200bps； 6：512000bps；	0~6	5
Pn5.005	0x4505	RS485 模式选择	0：ModbusRTU； 1：ModbusAS		0
Pn5.00A	0x450A	通讯参数保存设置	0：参数在 RAM 中运行； 1：参数写入 flash 保存	0~1	0

2.4 Modbus_RTU 简介

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。支持的命令示例如下：

2.4.1 单轴驱动器示例

■ 例 1：读驱动器 IO 输出（0x01）

发送	站号	命令	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	站号	命令	字节数	数据	CRC 校验	
	01	01	01	01	90	48

解析	单轴驱动器有两路输出，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 表示两路均不导通；0x02 表示输出 2 导通；0x03 表示两路均导通；

■ 例 2：读驱动器 IO 输入（0x02）

发送	站号	命令	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	站号	命令	字节数	数据	CRC 校验	
	01	02	01	00	A1	88

解析	当前表示读取的 4 路输入均为 0； 4 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。

■ 例 3：读寄存器（0x03）

发送	站号	命令	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	站号	命令	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。

■ 例 4：写单个寄存器（0x06）

发送	站号	命令	寄存器起始地址		数据		CRC 校验	
	01	06	00	33	00	C8	78	53

返回	站号	命令	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

■ 例 5：写多个寄存器（0x10）

发送	站号	命令	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	站号	命令	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。

2.4.2 双轴驱动器示例

■ 例 1 读驱动器 IO 输出（0x01）

发送	站号	命令	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	站号	命令	字节数	数据	CRC 校验	
	01	01	01	01	90	48

解析	DS 双轴（二合一）驱动器有 4 路输出，通过以上指令读到 0x01 表示：输出 1 有效，其他未导通；0x00 表示均不导通；0x0F 表示 4 路均导通；

■ 例 2：读驱动器 IO 输入（0x02）

发送	站号	命令	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	站号	命令	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 0x83，二进制为 1000 0011，则表示输入 1、2 和 8 有效。

■ 例 3：读寄存器（0x03）

发送	站号	命令	寄存器起始地址		寄存器单元长度		CRC 校验	
	01	03	23	18	00	02	4F	88

返回	站号	命令	字节数	数据 1		数据 2		CRC 校验	
	01	03	04	00	64	01	F4	BB	FB

解析	表示读取 1#轴前两端速度 F0.3.18 和 F0.3.19，读出的速度分别为 100rpm 和 500rpm。

■ 例 4：写单个寄存器（0x06）

发送	站号	命令	寄存器起始地址		数据		CRC 校验	
	01	06	21	03	00	01	B2	36

返回	站号	命令	寄存器起始地址		数据		CRC 校验	
	01	06	21	03	00	01	B2	36

解析	表示写 F0.1.003 为 1，1#电机旋转方向取反

■ 例 5：写多个寄存器（0x10）

发送	站号	命令	寄存器起始地址		寄存器长度		字节数	数据				CRC 校验	
	01	10	23	18	00	02	04	00	32	00	64	DE	10

返回	站号	命令	寄存器起始地址		寄存器长度		CRC 校验	
	01	10	23	18	00	02	CA	4B

解析	表示连续写 1#轴前两段速度 F0.3.18 和 F0.3.19 为 50rpm 和 100rpm

2.5 Modbus_ASCII 简介

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 0DH 和 0AH。

常用数字的 ASCII 码如下页所示：

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

本节只解析数据结构，指令例程详见上一节。

2.6 Modbus 运动控制配置

驱动器支持基于 Modbus 协议的内部速度模式控制，配置如下：

2.6.1 速度模式指令（单轴）

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：

Fn00 = 1（内部速度模式）；
Fn99 = 1（通讯修改参数不保存）；
Fn33 = 0（初始上电速度为0）；
Fn38 = 0（初始状态不使能）。

STEP2. 操作：

1. 设置电机速度Fn33：
   发送：0x01 06 00 33 00 C8 78 53 //设置速度为200rpm
2. 使能电机：
   发送：0x01 06 00 38 00 01 C9 C7 //设置 Fn38=1，使电机使能；
3. 读 Dn00 电机转速：
   发送：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。

2.6.2 速度模式指令（双轴）

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：

F0.1.002 = 1，F1.1.002 = 1（1#轴、2#轴都为速度模式）；
F0.3.018 = 0，F1.3.018 = 0（初始上电速度为0）；
F0.1.000 = 0，F1.1.000 = 0（初始状态不使能）；
Pn5.00A=0 （通讯修改参数不保存）。

STEP2. 操作：

1. 设置电机速度F0.3.018（1#轴）和F1.3.018（2#轴）：
   发送：0x01 06 23 18 00 C8 03 DF //设置1#电机速度为200rpm
   发送：0x01 06 33 18 00 C8 07 1F //设置2#电机速度为200rpm
2. 使能电机：
   发送：0x01 06 21 00 00 01 42 36 //设置 F0.1.000=1，使 1#电机使能；
   发送：0x01 06 31 00 00 01 46 F6 //设置 F1.1.000=1，使 2#电机使能；
3. 读 D0/1.00 电机转速：
   发送：0x01 03 50 00 00 01 95 0A //读 1#电机速度，单位 rpm
   发送：0x01 03 51 00 00 01 94 F6 //读 2#电机速度，单位 rpm

■ **注意事项：**

驱动器每次上电后，只需通讯刷入速度值指令和使能指令，电机即可按要求运行，由于设置了通讯修改的参数不保存 Flash，断电后都会恢复初始设定状态。

2.7 双轴（二合一）特有协议

2.7.1 双读双写指令

为了更好的双轴同步控制，用户可通过双读双写指令控制双轴（二合一）驱动器进行参数读写，协议格式如下：

■ 读取两个不同的地址数据

发送	站号	命令	地址 1		地址 2		CRC 校验	
	01	43	High	Low	High	Low	Low	High

返回	站号	命令	地址 1		地址 2		数据 1		数据 2		CRC 校验	
	01	43	High	Low	High	Low	High	Low	High	Low	Low	High

以上指令读出地址 1 的值为数据 1，地址 2 的值为数据 2。

例如：同时读双轴驱动器编码器位置数据，1#、2#轴编码器位置地址分别为 0x5004 和 0x5104

发送：0x01 43 50 04 51 04 28 97 //读 0x5004 和 0x5104 的值

返回：0x01 43 50 04 51 04 A1L A1H A2L A2H CRCL CRCH

■ 写两个地址值

发送	站号	命令	地址 1		地址 2		数据 1		数据 2		CRC 校验	
	01	44	High	Low	High	Low	High	Low	High	Low	Low	High

返回	站号	命令	地址 1		地址 2		数据 1		数据 2		CRC 校验	
	01	44	High	Low	High	Low	High	Low	High	Low	Low	High

以上指令将数据 1 写入地址 1，数据 2 写入地址 2。

例如：同时写双轴驱动器两个轴的速度为 100rpm：

发送：0x01 44 23 18 33 18 00 64 00 64 9D 38

返回：0x01 44 23 18 33 18 00 64 00 64 9D 38

■ 通过双读双写指令进行电机控制时，步骤如下：

STEP1.确认驱动器参数：

F0.1.002 = 1，F1.1.002 = 1（1#轴、2#轴都为速度模式）；
F0.3.018 = 0，F1.3.018 = 0（初始上电速度为0）；
F0.1.000 = 0，F1.1.000 = 0（初始状态不使能）；
Pn5.00A=0 （通讯修改参数不保存）。

STEP2. 操作：

1. 使能电机：
   发送：0x01 44 21 00 31 00 00 01 00 01 75 34 //1#、2#轴使能
2. 设置 1#轴、2#轴电机速度（F0.3.018、F1.3.018，以下为不同速度指令参考）
   发送：0x01 44 23 18 33 18 00 64 00 32 1D 06 //1#轴100rpm，2#轴50rpm
   发送：0x01 44 23 18 33 18 00 64 00 64 9D 38 //1#轴 100rpm，2#轴 100rpm
   发送：0x01 44 23 18 33 18 00 64 FF 9C DD 4A //1#轴 100rpm，2#轴-100rpm
3. 读 D0/1.00 电机转速
   发送：0x01 43 50 00 51 00 68 95 //读 0x5000 和 0x5100 的值

4. 关闭电机使能，电机停止运转
   发送：0x01 44 21 00 31 00 00 00 00 00 E5 34 //1#2#轴失能

■ **注意事项：**

驱动器每次上电后，只需通讯刷入使能和速度值指令，电机即可按要求运行，由于设置了通讯修改的参数不保存 Flash，断电后都会恢复初始设定状态。

2.7.2 差速模式指令（选配）

为满足移动类 AGV 及机器人底盘等应用，双轴（二合一）驱动器支持自定义的差速指令，即通过一条串口指令同时控制两个电机，从而达到双轴完全同步的效果。

该模式请设置控制模式 F0/1.1.002 为 1（速度模式），设置双轴轮径（Pn6.00A / Pn6.00B），定义左右轮（Pn6.00C），使能电机 F0/1.1.000 为 1（使能）。

指令格式如下，其中，线速度单位为 mm/s，角速度单位为 mrad/s，两个字节组成的线速度、角速度、CRC 均低字节在前。

发送	站号	命令	线速度		角速度		CRC 校验	
	01	EA	ByteVL	ByteVH	ByteAL	ByteAH	CRCL	CRCH

2.8 上位机软件 SyntronTS

用户可以通过和利时电机提供的上位机软件 SyntronTS 实现驱动器的参数读写、状态观测等功能。以下为软件具体操作：

2.8.1 接线与设置

➢ 外部接线

上位机软件通过 RS232 或 RS485，以串口通讯的形式与伺服驱动器进行通讯，因此需要一条串口线，串口线推荐如下：

上位机	通讯方式	接口转换	线缆要求
台式机	RS232	DB9->RJ45	大部分的台式机都具有串行通讯端口，因此直接连接驱动器通讯端口 RS232 信号与台式机的串口即可。（RX、TX、GND）
台式机	RS485	USB->RJ45	上位机通过 USB 转 RS485 转换器连接伺服驱动器，RS485 端连接伺服 RJ45 通讯端口内的 RS485A、RS485B、GND
笔记本	RS232		上位机通过 USB 转 RS232 转换器连接伺服驱动器，RS232 端连接伺服 RJ45 通讯端口内的 TX、RX、GND

➢ 驱动器设置

请确认驱动器内部参数：站号、波特率，以备后续软件设置。一般情况下，驱动默认站号为 1，波特率为 115200bps。

➢ 上位机设置

1、查看 COM 端口号

电脑接好通讯线缆后，可以查看对应的通讯 COM 端口号，如下图。

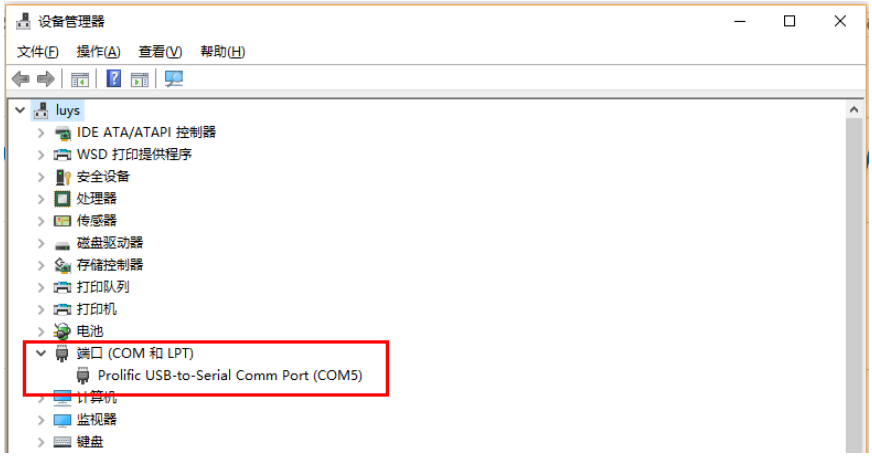


图 2-2 查看当前 COM 口

以 Windows 系统为例，右键单击计算机→管理，打开“计算机管理”对话框，双击“设备管理器”选项，打开“端口（COM 和 LPT）”，查看 COM 端口号，如上图“ (COM5)”。

2、软件启动

双击软件图标，启动软件，如图 2-3。

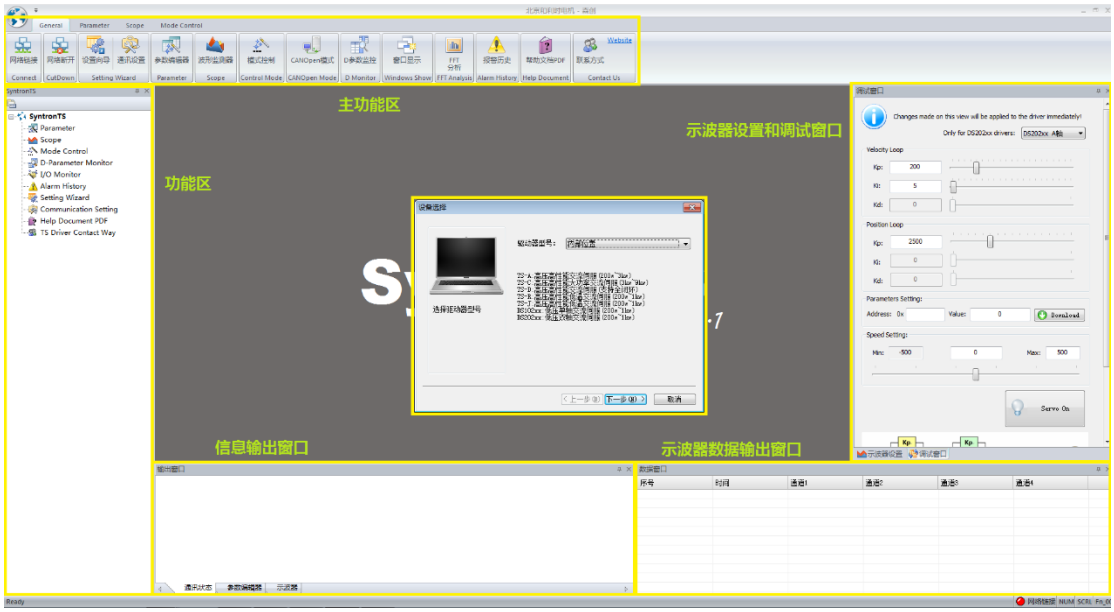


图 2-3 软件界面

程序启动后界面中间会自动弹出“设备选择”向导对话框（也可通过主功能区中的“设置向导”按钮打开该对话框）。通过该向导可以快速完成：设备型号选择、通讯设置、连接设备、指定工作路径、设置电机正/反转位置限幅值、读取额定转速和额定电流用于示波器显示等功能。

3、设备选择

设备选择栏中，可选择目标目标驱动器，其中单轴伺服驱动器选择型号为 DS102xx，双轴（二合一）伺服驱动器选择型号为 DS202xx。

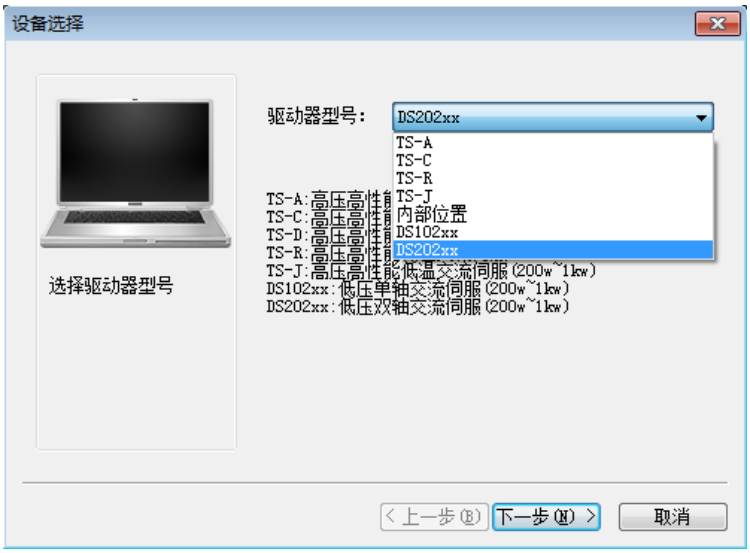


图 2-4 设备选择

4、通讯设置

单击向导对话框中的“通讯设置”按钮，或单击主功能区中的通讯设置按钮都可以打开“通讯设置”对话框，如下所示。

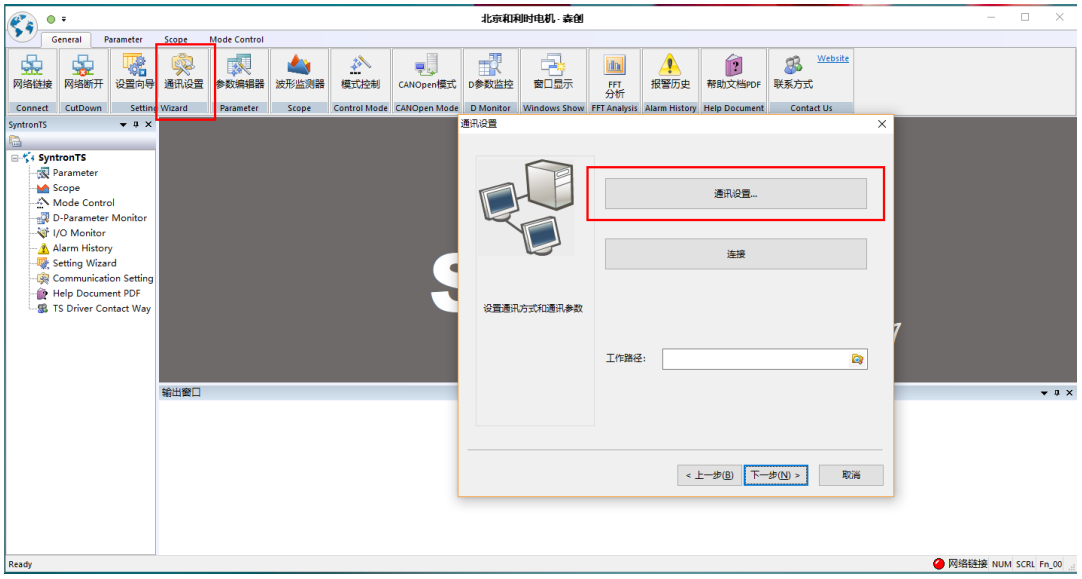


图 2-5 通讯设置

通讯方式选择 RS232/RS485，端口号选择目前串口号，波特率设为 115200bps，站号设为 1，同驱动器参数的设置一致，协议选择 ModBus RTU。

设置完毕后，单击“确定”按钮使设置生效，退出通讯设置。

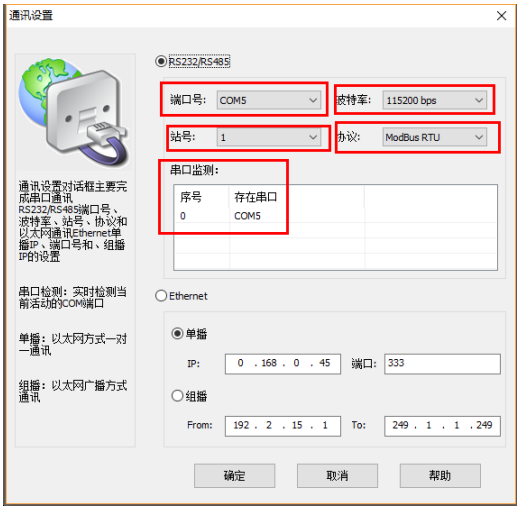


图 2-6 通讯设置详细

5、连接串口

通讯设置完成后，单击向导对话框中的“连接”按钮，或单击主工具栏中的“网络连接”按钮，即可完成通讯软件与驱动器的连接，如图所示：

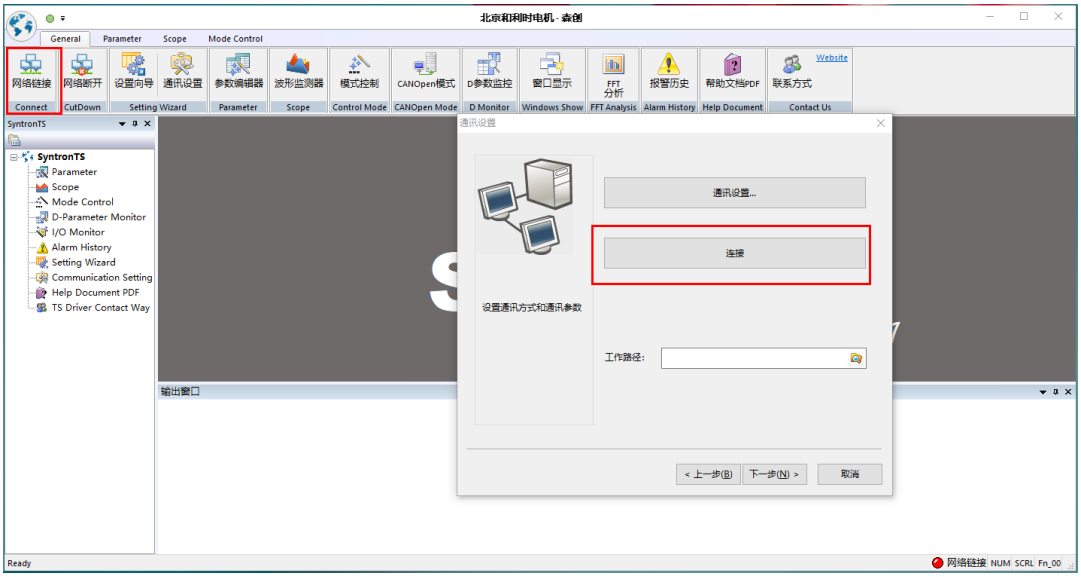


图 2-7 连接串口

连接成功后，输出窗口会提示相关信息，如下图：

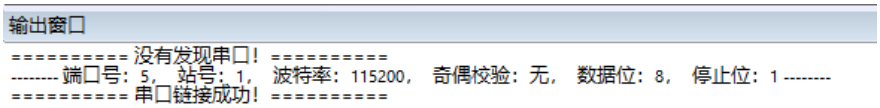


图 2-8 输出窗口

2.8.2 参数编辑器

用户可通过参数编辑器保存或修改驱动器参数。

1、打开参数表

双击主功能区中的**参数编辑器**或功能区的 **Parameter** 选项，即可打开参数编辑器参数表，如下图所示：

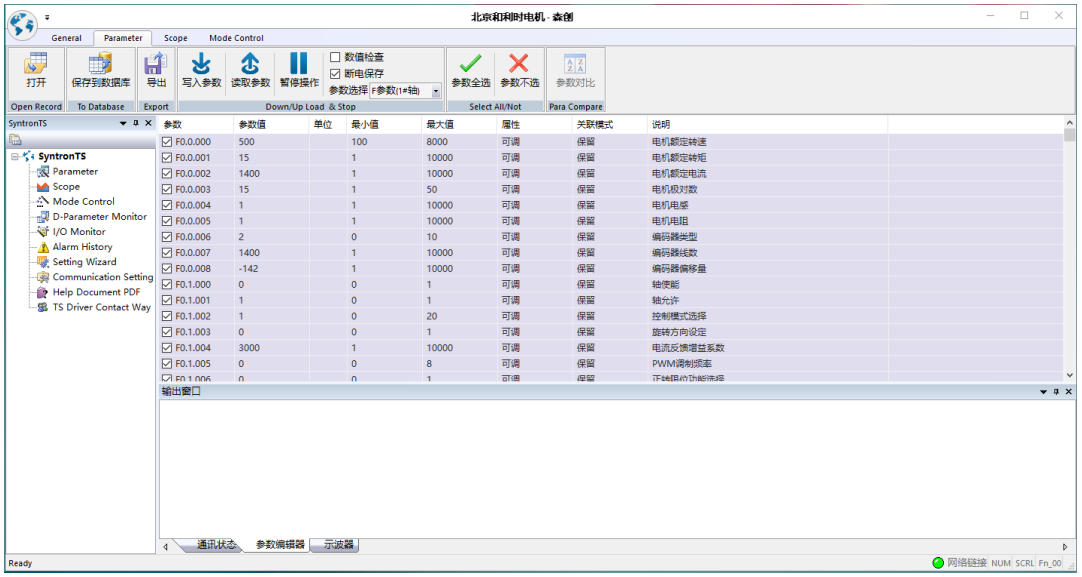


图 2-9 参数编辑器

2、编辑参数

双击参数值即可修改该参数，如下图所示：

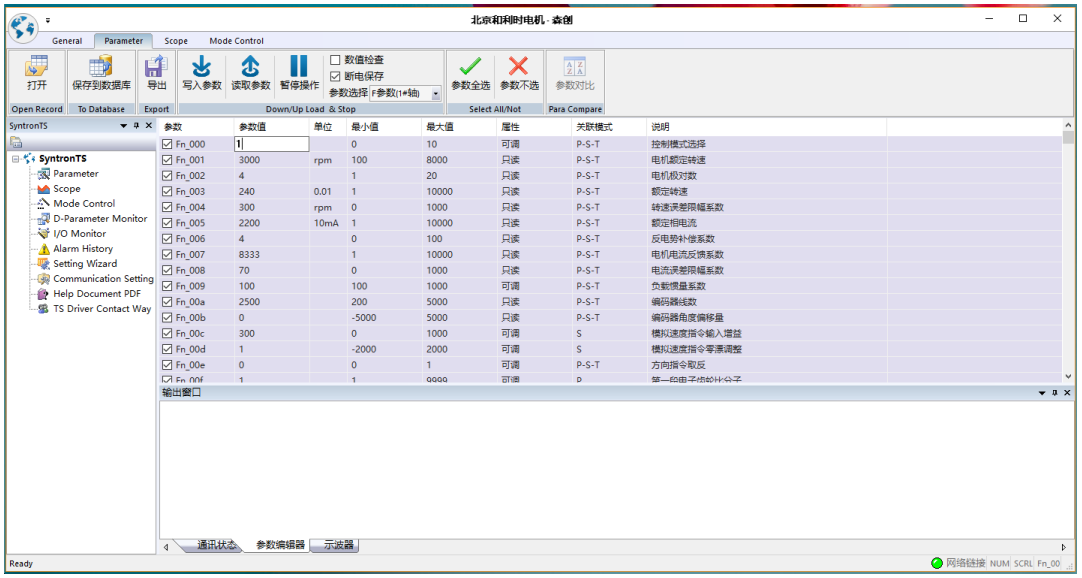


图 2-10 编辑参数

3、写入参数

在通讯连接正常情况下，单击“写入参数”按钮，会出现写入参数的进度条，进度条完毕后，即驱动器已写入勾选了的参数值。注意，写入参数请**尽量避免全选写入**，否则部分伺服默认参数可能会被修改，导致伺服工作异常。写入界面如下图所示：

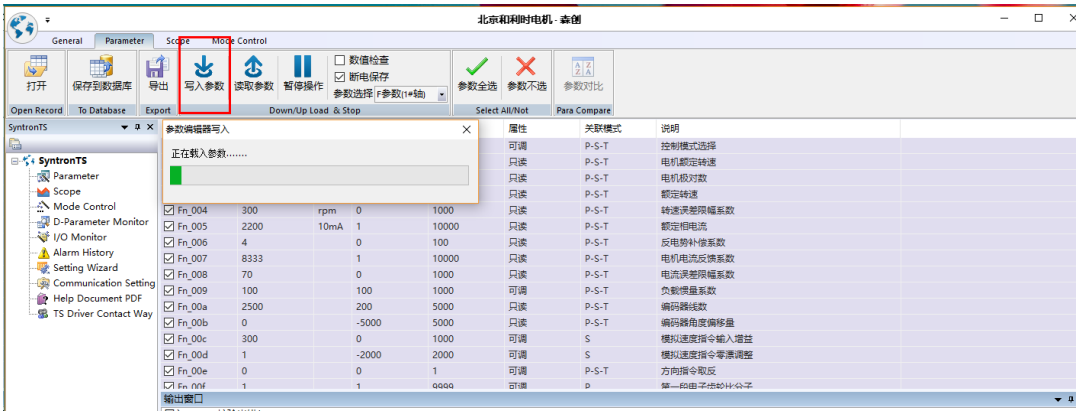


图 2-11 写入参数

4、读取参数

在通讯连接正常情况下，单击“读取参数”按钮，即可从驱动器中读取勾选的参数值，如图 17 所示。



图 2-12 读取参数

5、参数导入导出

参数保存有两种形式。一种是保存进数据库，另一种是导出。保存进数据库，则每次打开上位机，默认打开这个数据库参数；导出文件文件格式也是 mdb 数据库文件，可点击导入，导入至当前上位机参数表。

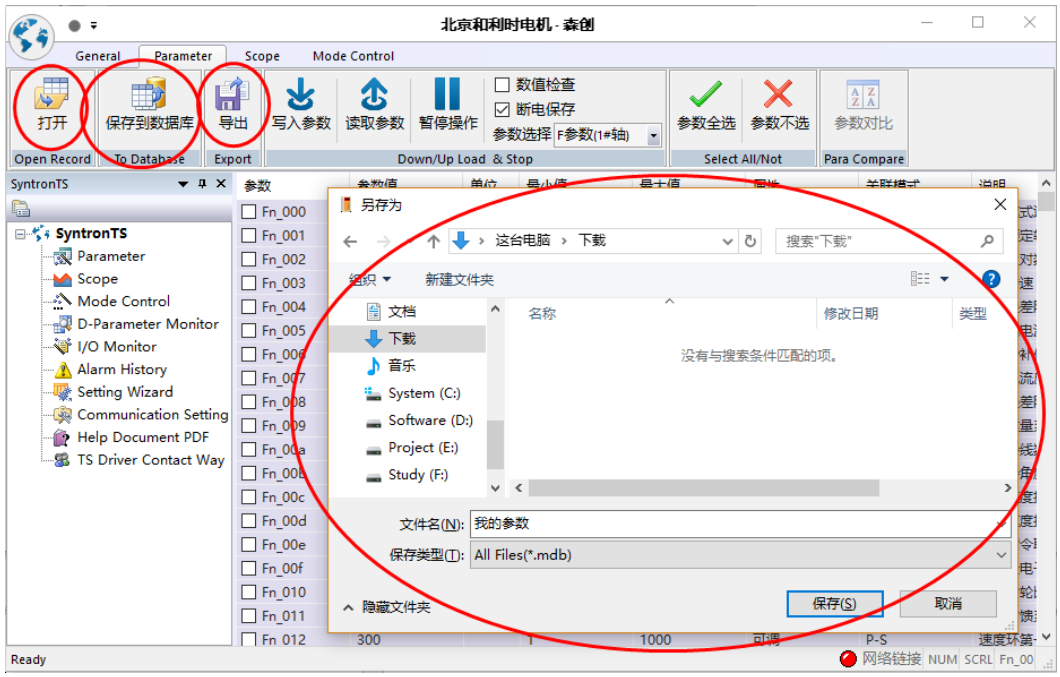


图 2-13 参数导出

6、全选与取消全选

单击参数编辑器工具条中的“参数全选”按钮可全部勾选参数，单击“参数不选”按钮可全部勾选参数。

若选择部分参数，则全选按钮需要中转。参数表右边空白处为临时中转区，单击该区域后，再点击参数全选，即可选定所有参数。如下图所示。



图 2-14 参数全选与不选

2.8.3 波形检测器

1、打开波形检测器

双击主功能区中的“波形检测器”按钮，或功能区中的“Scope”按钮，即可打开示波器：

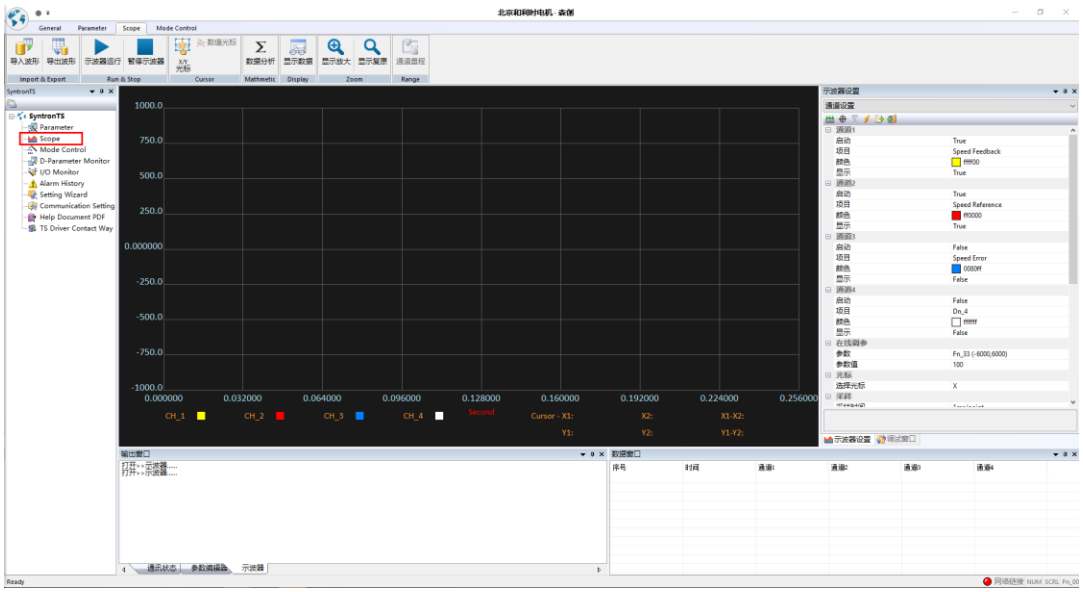


图 2-15 示波器界面

2、配置示波器

示波器的配置主要在“示波器设置”区中完成，配置步骤如下：

- 启动：True 表示启动该通道，False 表示关闭该通道，如下图。

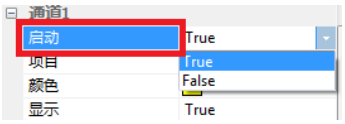


图 2-16 通道开关

- 项目：选择该通道观察的项目，例如 Position Reference(位置给定)、Position Error(位置误差)、Speed Reference(速度给定)、Speed Feedback(速度反馈)等，如下所示。

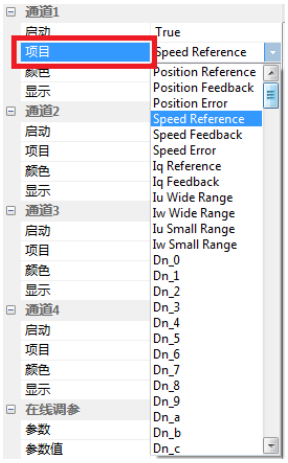


图 2-17 观察项目

- 颜色：选择该通道曲线的显示颜色，每次改动颜色后，下次再运行示波器时曲线颜色改动生效：

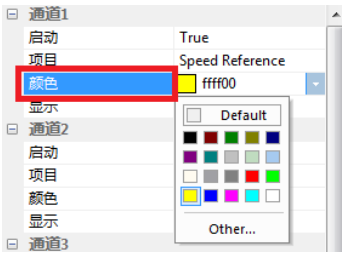


图 2-18 曲线颜色

- 显示：True 表示显示该通道曲线，False 表示不显示该通道曲线，如图 23。

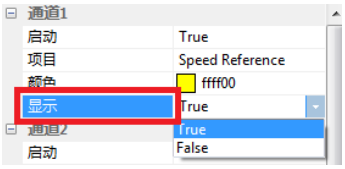


图 2-19 通道显示

- 在线调试：该功能可以在示波器正在运行或者未运行时实时修改驱动器的参数，一次只能修改一个值。在“参数”选项中选择要修改的参数，“参数值”选项中填入参数值。然后单击“在线调试”按钮参数即可实时写入驱动器内部，如下所示。

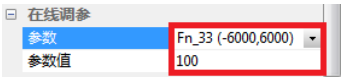


图 2-20 在线调参

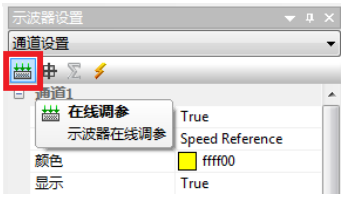


图 2-21 调参写入

- 光标：选择示波器光标，提供三种选择：X 光标（时间轴光标）、Y 光标（幅值光标）、X+Y 光标（时间轴光标和幅值光标）：

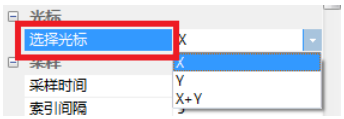


图 2-22 光标选择

然后单击示波器设置区或示波器工具条的光标按钮，或在示波器区域单击右键选择“光标”，即可打开光标功能，取消光标时只需再次单击光标按钮即可：

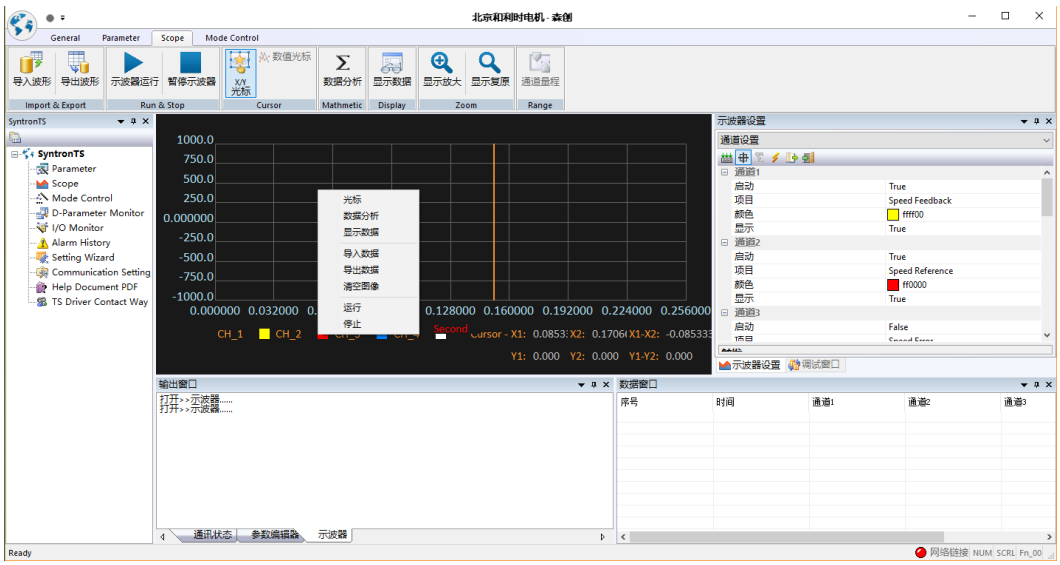


图 2-23 光标选择

用鼠标拖动光标，右下角可以实时显示光标的刻度值，如下所示。

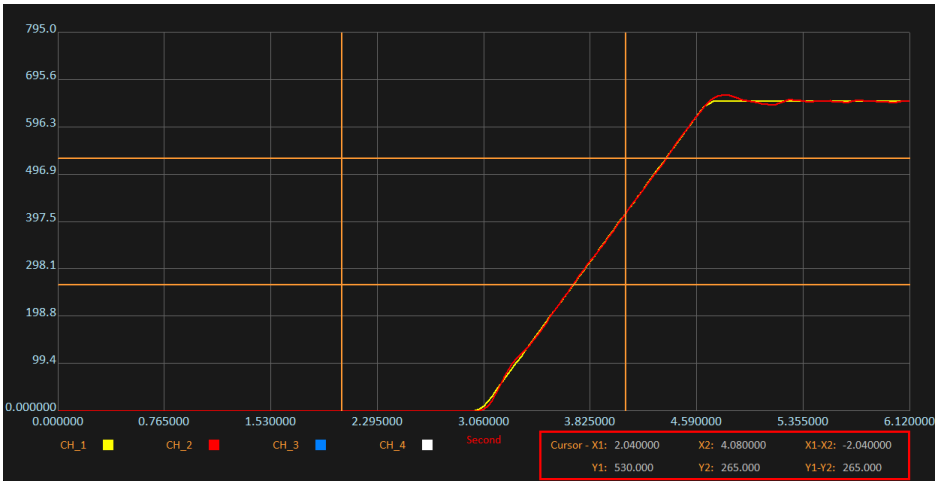


图 2-24 示波器光标显示

- 采样时间：示波器波形曲线每两个点之间的时间间隔，如下图：

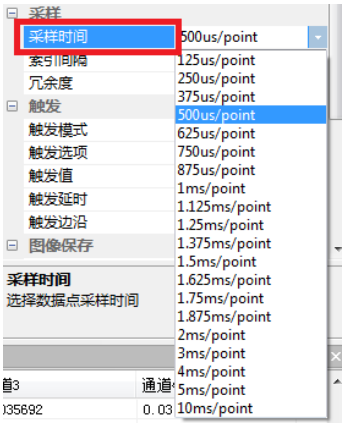


图 2-25 采样时间

- 采样间隔：当采样间隔设为 1 时，每个通道曲线的点数为 2048 个。当采样间隔设为 2 时，每个通道曲线的点数为 1024 个。为了提高示波器运行时曲线的显示速度，可以通过设置采样间隔来调整曲线的总数据点个数，如下图：

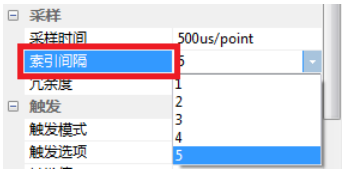


图 2-26 采样间隔

- 冗余度：通过设置冗余度的数值，来确定数据传输过程出现错误帧的时候通讯软件的重复请求次数，如下图：

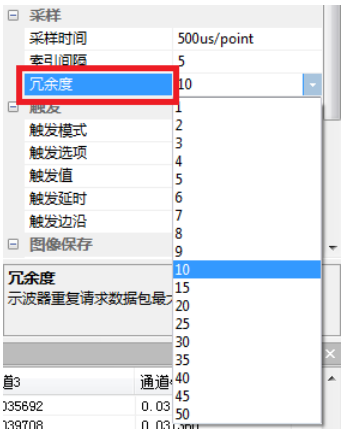


图 2-27 冗余度

- 触发模式：自动模式指示波器运行启动时马上开始采集数据；标准模式指驱动器满足标准触发模式设定的触发条件时才开始采集数据，如下图：



图 2-28 触发模式

- 触发选项：在标准触发模式下，由触发选项来指定以四个通道中哪个通道作为触发标准，如下图：



图 2-29 触发选项

- 触发值：在标准触发模式下，单击“Click here”右边的按钮，可以弹出“触发值”对话框，如下图所示：



图 2-30 触发值



图 2-31 触发值设定

- 触发边沿：在标准触发模式下，该选项用来设定以“上升沿”还是“下降沿”触发，如下图：

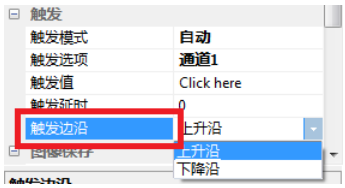


图 2-32 触发边沿

3、运行示波器

通讯连接正常下，单击“运行”按钮，即可启动示波器的运行：

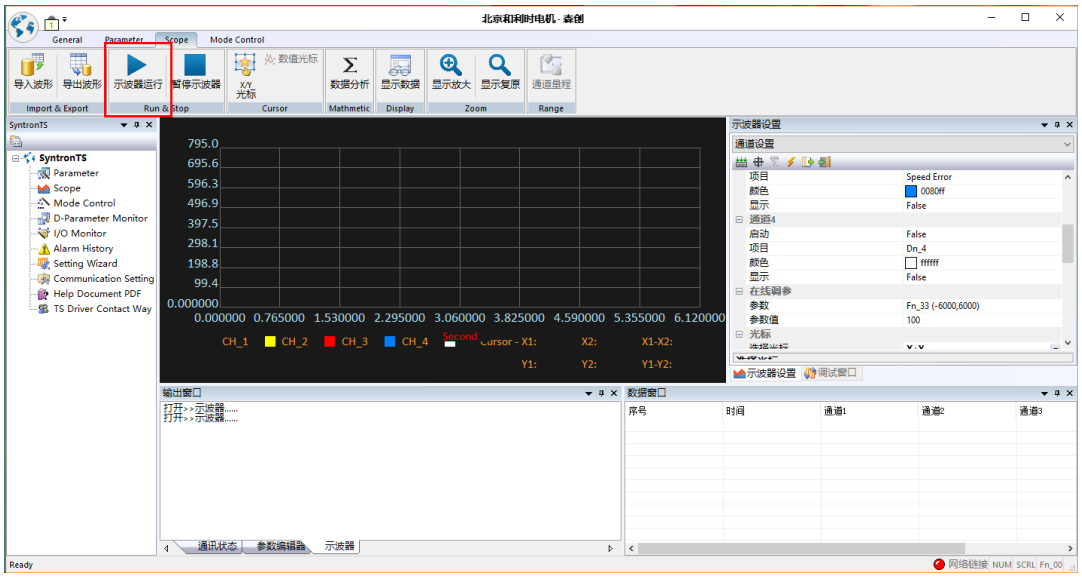


图 2-33 启动示波器

4、暂停示波器

单击“暂停”按钮，即可暂停示波器的运行，终止通讯软件与驱动器之间的数据传输。

5、放大和取消放大

- 放大：如果想观察波形曲线的局部特征，可以采用示波器的放大功能。单击示波器工具条中的“显示放大”按钮，该按钮按下表示选中该功能→按下鼠标左键→拖动鼠标从左上向右下拖动→可以拖出红色矩形框→抬起鼠标左键，即可放大该矩形框内部的图形曲线，如下图。

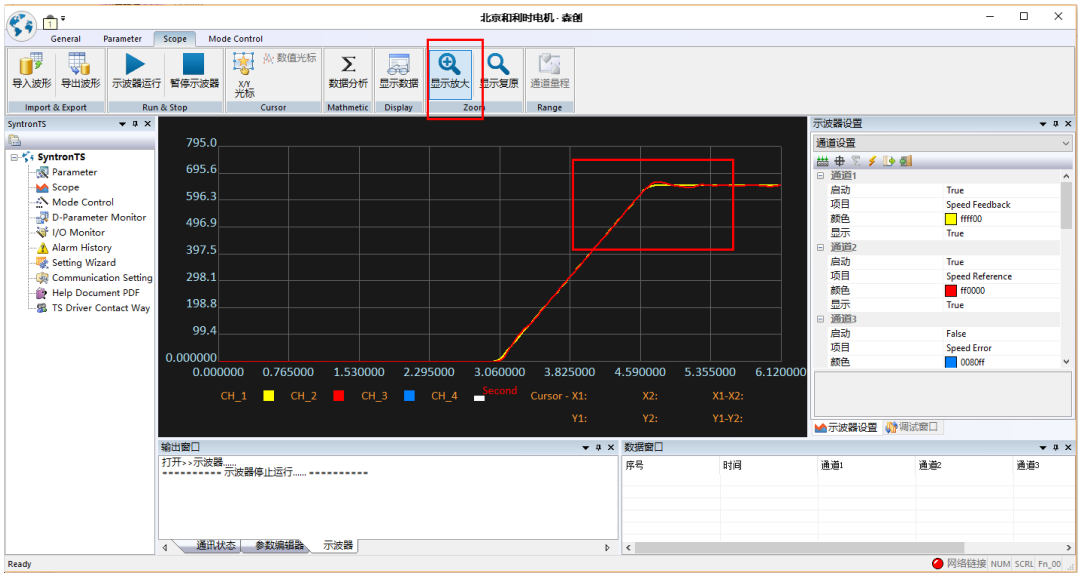


图 2-34 放大

- 取消放大：在“显示放大”按钮按下状态下，按下鼠标左键→从右下向左上方向拖动鼠标→抬起鼠标左键，即可取消放大状态。

6、数据分析

该功能可计算光标 X1 和 X2 范围之间各通道数据点的最大值、最小值和平均值。

- 打开 X 坐标：根据前述方法打开光标功能，拖动光标 X1 和 X2，从而确定希望进行数据分析的区域范围，如下图：

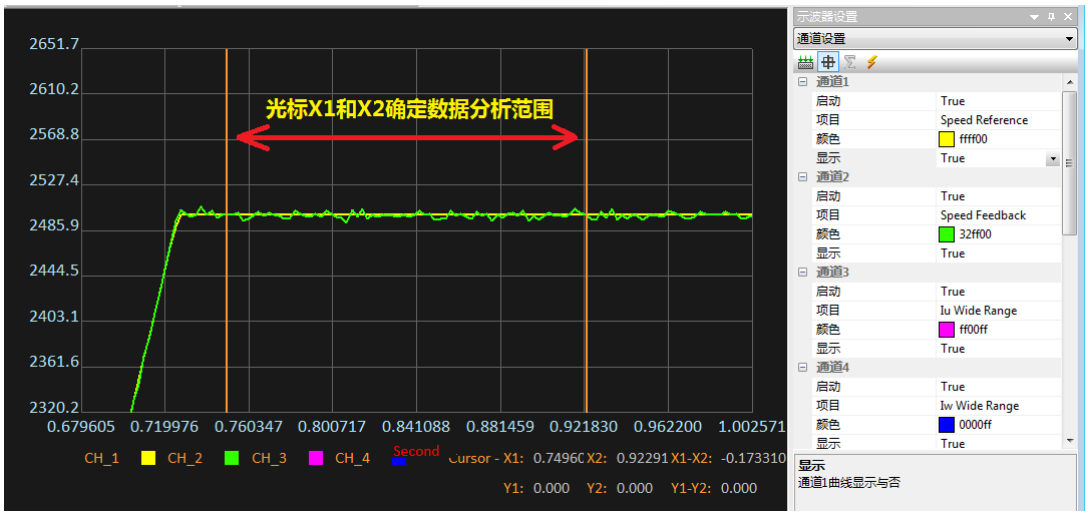


图 2-35 分析区域

- 打开数据分析对话框：确定好数据分析范围后，打开“示波器数据分析”对话框，

可以右键→数据分析，或单击示波器设置（或示波器工具条）中的分析按钮，即可打开“示波器数据分析”对话框。

- 进行数据分析：单击“示波器数据分析”对话框中的“数据分析”按钮，即可开始计算，单击“确定”或“取消”按钮可以关闭该对话框，如下图所示：

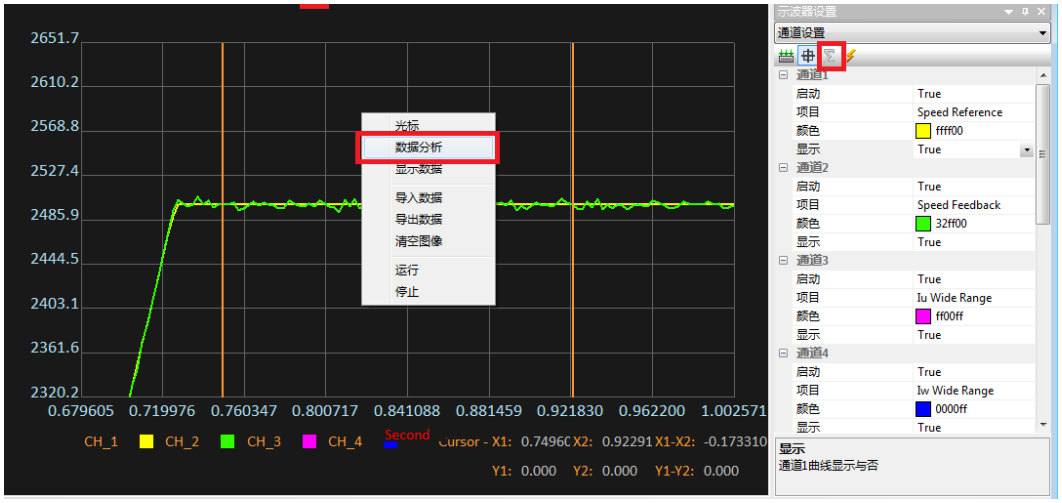


图 2-36 数据分析



图 2-37 数据开始分析

7、显示数据

右键→显示数据或单击示波器设置区（或示波器工具条）中的“显示数据”按钮，即可在“数据窗口”中显示各个通道的数据值，如下所示：

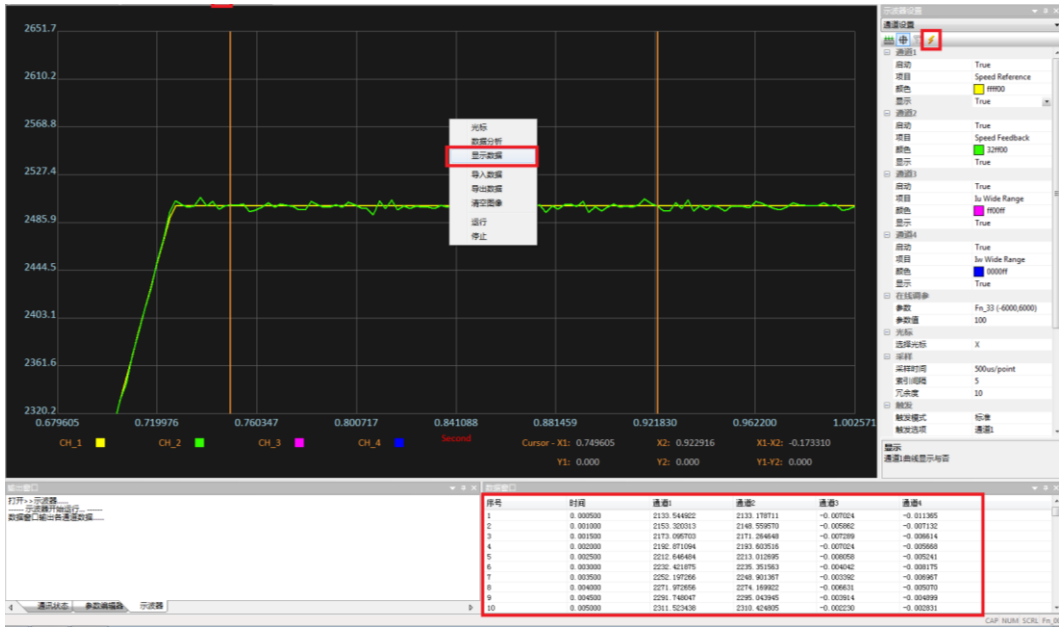


图 2-38 显示数据

8、导出、导入和清空图像

- 导出波形：右键→导出数据或单击示波器工具条中的“导出波形”按钮，弹出“另存为”对话框，选择保存路径，文件名称，单击“保存”按钮即可保存，如下所示：

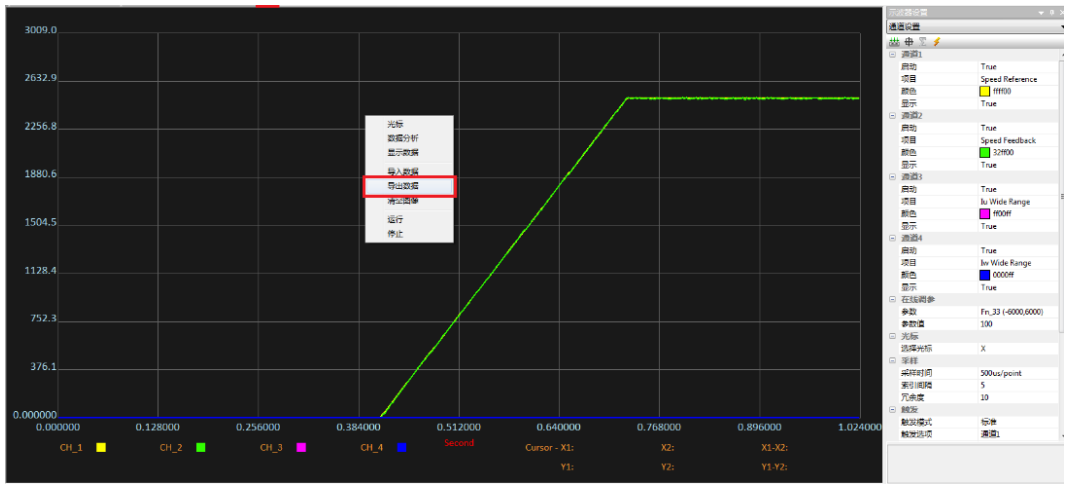


图 2-39 波形导出

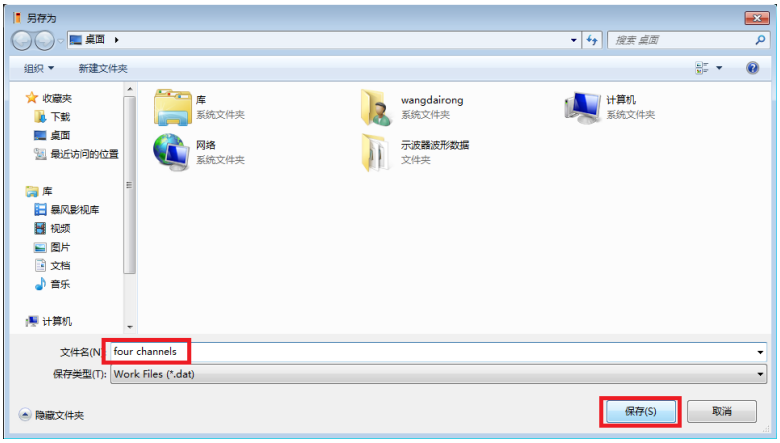


图 2-40 保存波形

- 导入波形：如图想查看之前保存的波形图像，可以右键→导入数据，弹出“打开”对话框，选择之前的.dat 数据文件，单击“打开”按钮即可，如下所示：

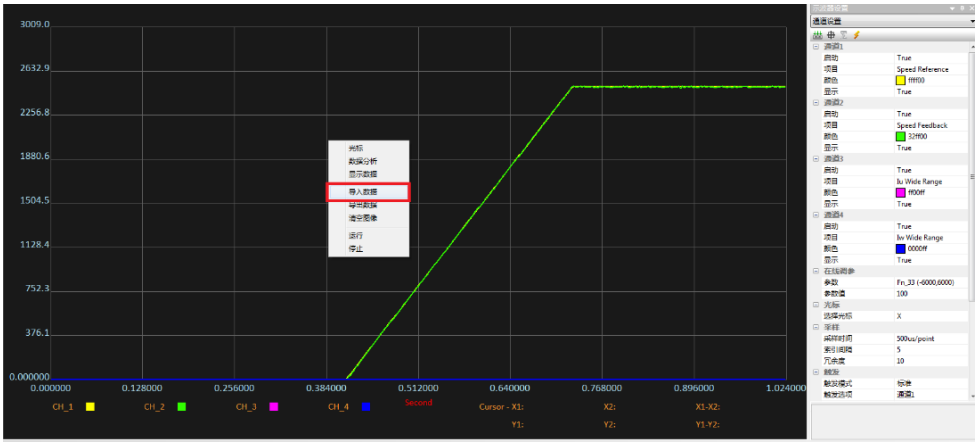


图 2-41 导入波形

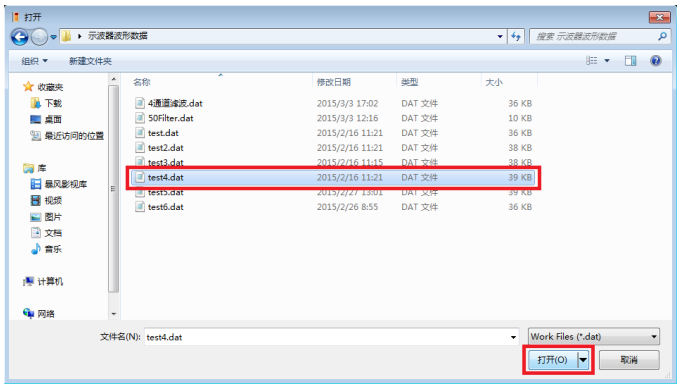


图 2-42 打开波形文件

- 清空图像：如果想清空示波器现有图像可以右键→清空图像。

2.8.4 D 参数监控

在通讯连接正常的情况下，单击主功能区中的“D 参数监控”，即可打开相应对话框。单击“Start Refresh”按钮，即可实时监测四个选定的 D 参数值以及驱动器上 I/O 管脚的信号状态，如果当前管脚有信号，则 I/O 监测器中该管脚对应的图标显示填充色，否则显示未填充的灰色，如下所示：

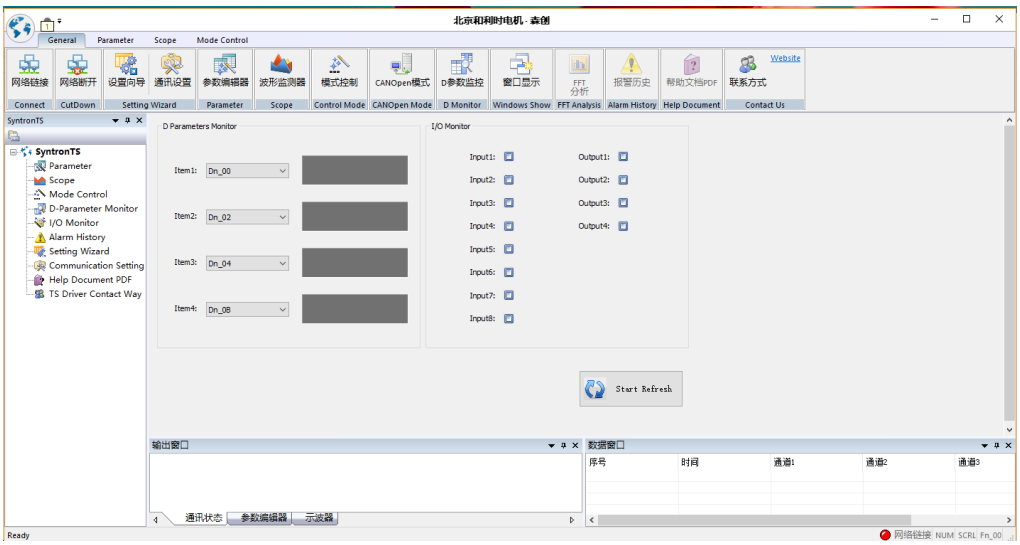


图 2-43　I/0 监测

2.8.5 注意事项

- 严禁在驱动器带电情况下插拔通讯线缆，否则会造成损坏。
- 请尽量选择可调属性的参数进行修改，若全部参数写入，可能会报错误，如出现错误，请查看该参数是否为只读，并取消当前参数勾选。
- 在通过通讯软件向驱动器内烧写参数的时候，请注意通讯设定参数，在通讯软件的烧写过程中如果修改了驱动器的通讯设定参数，再次连接需要按驱动设定来设置上位机站号波特率信息。
- 如果已经开启实时监测，那么关闭监测器对话框前一定要暂停监测，否则可能出现死机现象。
- 不要在通讯正在进行的时候断开连接，例如：正在读写参数或者示波器正在进行数据传输的时候断开连接，否则程序将可能出错。

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。

DS / RS 伺服驱动器在 CANopen 总线网络（支持“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

用户可通过 CAN 总线控制 DS / RS 型号伺服驱动器，以下为常见测试接线：

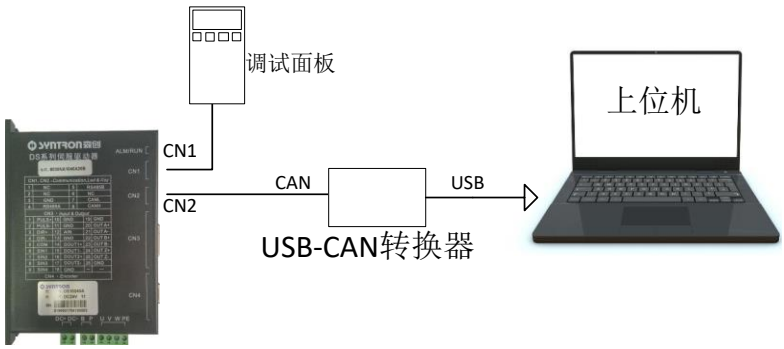


图 3-1 上位机 CAN 测试连接

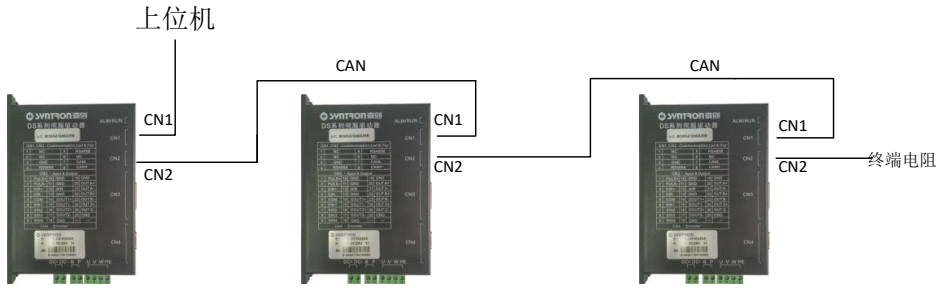


图 3-2 CAN 总线组网示例

用户可通过 USB 转 CAN 工具来进行测试和熟悉。

3.2 关键参数（单轴）

单轴驱动器 CAN 总线 ID 默认为 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
Fn00	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
Fn8D	0x208D	协同模式目标 ID	如目标 CAN ID 驱动报警，则协同模式生效，本驱动器报警方式由 FnF8 决定，报警码为 Err_BE。	0-127	0
Fn8E	0x208E	故障上报时间	0：故障后上报一次报警码； >1：故障后定时上报时间（单位：ms）	0-4000	50
Fn8F	0x208F	6063H 取反	0：正转增计数；1：正转减计数；	0-1	0
Fn99	0x2099	通讯参数模式	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
Fn9A	0x209A	CAN 通讯 ID	0 为广播地址，1~127 为站址编号	0~127	1
Fn9C	0x209C	CAN 通讯波特率	CAN 总线波特率，单位为 Kbps	100-1000	500
Fn9E	0x209E	TPDO 上传机制	0:上电自动上传; 1:需配置启动上传;	0~1	0
FnEA	0x20EA	6063 清零	置 1 则对象字典 6063 清 0	0~1	0
FnF7	0x20F7	CAN 总线断线检测时间	CAN 总线断线检测时间，单位为 ms，报警方式由 FnF8 决定，报警码为 Err_Lo。	0-30000	0
FnF8	0x20F8	自主停车、总线报警方式	0：报警；1：关使能；2：给零速；	0-2	0
FnFF	0x20FF	CAN 限位主动上传开关	0：限位等信息不主动上传； 1：限位信息、位置到达信息通过紧急帧上传。	0-1	0
Dn12	0x3012	报警信息码	见附录报警信息		
Dn1D	0x301D	接收帧总数	*开发参数，用于简易检测接收 can 指令的总数		
Dn1E	0x301E	发送帧总数	*开发参数，用于简易检测发送 can 报文的总数		

3.3 关键参数（双轴）

双轴驱动器 1#、2#轴 CAN 总线 ID 分别默认为 1、2，波特率为 1000Kbps。

参数编号	CAN INDEX	参数名称	参数内容	设定范围	出厂设定
F0.1.002	0x2102	1#轴控制模式	1：内部速度模式； 2：位置模式； 20：CANopen 模式	0-20	1
F1.1.002	0x3102	2#轴控制模式			
Pn5.006	0x4506	CAN 通讯 ID	1#轴 ID 为个位十位，2#轴 ID 为百位千位， 例如 0201 表示 1#轴 ID 为 1，2#轴 ID 为 2	0-9999	0201
Pn5.007	0x4507	CAN 波特率	CAN 总线波特率，单位为 Kbps	0-1000	1000
Pn5.009	0x4509	通讯选择	001：RS232；010：RS4855；100：CAN	0-111	111
Pn5.00A	0x450A	通讯写保护设置	0：参数写入 RAM；1：参数写入 flash	0~1	0
Pn5.00B	0x450B	CAN 总线断线检测时间	CAN 总线断线检测时间（单位：ms），超时则触发保护动作	0-9999	0
Pn5.00C	0x450C	CAN 总线断线保护动作	0：报警； 1：关使能； 2：给零速	0-9999	0
Pn5.00D	0x450D	差速信息上报开关	0：不上报； 1：上报； CAN 差速指令用	0-1	0
Pn6.002	0x4602	通讯报警清除	通讯报警清除	0-1	0
Pn6.00A	0x460A	轮径	单位 0.1mm，CAN 差速指令用	0-9999	0
Pn6.00B	0x460B	轮距	单位 0.1mm，CAN 差速指令用	0-20000	0
Pn6.00C	0x460C	左右轮定义	差速指令用，适用于两轮差速系统，表示 1 轴或者 2 轴是左轮电机，由此定义行进方向	0-1	0
Dn0.012	0x5012	报警码	见 4.2 双轴报警信息	0-20	0

3.4 CANopen 对象字典

对象字典（Object Dictionary）是 CANopen 最重要的特性，每一个对象字典由 16 位的索引（Index）和 8 位的子索引（Sub Index）来寻址。设备状态可通过读对应的对象字典来查找，也可以通过写操作控制类的对象字典进行控制。一个节点的对象字典 Index 范围在 0x1000 到 0x9FFF 之间，标准结构如下表所示：

Index（hex）	内容
0000	保留
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）
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

驱动器的 Fn、Dn、Pn 参数地址位于制造商制定数据区域内（0x6000~0x9FFF），Fn、Pn 参数数据长度为 16bit，Dn 参数数据长度为 32bit，参考地址如下：

类型	Fn 起始地址	Dn 起始地址	Pn 起始地址
单轴驱动	0x2000	0x3000	-
双轴驱动	1#轴：0x2000 2#轴：0x3000	1#轴：0x5000 2#轴：0x5100	0x4000

以下为常用对象字典的简介：

3.4.1 控制字 6040H

对象字典 6040H，在 DS402 中定义如下：

Bit15-9	**Bit8**	**Bit7**	**Bit6-4**	**Bit3**	**Bit2**	**Bit1**	**Bit0**
Reserved	Halt	Fault Reset	Operation Mode specific	Enable Operation	Quick Stop	Enable Voltage	Switch on

DS / RS 系列伺服驱动器仅支持以下控制操作：

- 基本控制：首次使能伺服驱动器，必须依次写入 0x06、0x07、0x0F。之后写 0x07 则伺服断使能，再写 0x0F 则伺服使能；
- 报警复位：驱动器发生报警需要复位时，先写入 0x00，再写入 0x80，则当前报警清除，驱动器进入初始上电状态，如需操作，需再置该节点为可操作状态，进行模式配置。
- 紧急停止：写入 0x02，伺服会以对象字典 605AH 的规划曲线进入急停状态。

注意：部分版本可能不支持该指令，急停功能另外在 Fn 参数中实现。双轴驱动器暂未支持 603F、6040 复位、6040 急停等对象字典，主要功能均在 Fn 和 Dn 参数实现

3.4.2 状态字 6041H

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 数据低 8 位状态如下（0x 31）：

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

在控制字 6040H 写完 7 后，6041H 数据低 8 位状态如下（0x 33）：

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

在控制字 6040H 写完 F 后，6041H 数据低 8 位状态如下（0x 37）：

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

3.4.3 对象字典列表

DS / RS 驱动器支持的 DS402 对象字典如下表所示：

Mode	Index	Name	Type	Attr.	unit
ALL	603Fh	Error Code	UNSIGNED16	RO	-
ALL	6040h	Controlword	UNSIGNED16	RW	-
ALL	6041h	Statusword	UNSIGNED16	RO	-
ALL	6060h	Modes of operation	INTEGER8	RW	-
ALL	6061h	Modes of operation display	INTEGER8	RO	-
PP	6062h	Position demand value	INTEGER32	RO	pulse
PP	6063h	Position actual value*	INTEGER32	RO	increments
PP	6064h	Position actual value	INTEGER32	RO	cmd position
PP	6065h	Following error window	UNSIGNED32	RW	pulse
PP	6066h	Following error time out	UNSIGNED16	RW	ms
PP	6067h	Position window	UNSIGNED32	RW	pulse
PP	6068h	Position window time	UNSIGNED16	RW	ms
PV	6069h	velocity_sensor_actual_value	INT32	RO	0.1rpm
PV	606Bh	velocity_demand_value	INT32	RO	0.1rpm
PV	606Ch	velocity_actual_value	INT32	RO	
PV	606Dh	velocity_window	UINT16	RW	0.1rpm
PV	606Eh	velocity_window_time	UINT16	RW	ms
PV	606Fh	velocity_threshold	UINT16	RW	0.1rpm
PV	6070h	velocity_threshold_time	UINT16	RW	ms
PT	6071h	Target torque	INTEGER16	RW	0.1%Torque
PP	607Ah	Target position	INTEGER32	RW	pulse
HM	607Ch	Home offset	INTEGER32	RW	-

PP	6081h	Profile velocity	UNSIGNED32	RW	0.1rpm
Mode	Index	Name	Type	Attr.	unit
PV/PP	6083h	profile_acceleration	UINT32	RW	millisecond from 0rpm to 3000rpm
PV/PP	6084h	profile_deceleration	UINT32	RW	millisecond from 0rpm to 3000rpm
PT	6087h	Torque slope	INTEGER32	RW	-
PP/HM	6093h	Position factor	UNSIGNED32	RW	-
HM	6098h	Homing method	INTEGER8	RW	-
HM	6099h	Homing speeds	ARRAY	RW	0.1rpm
HM	609Ah	Homing acceleration	UNSIGNED32	RW	millisecond from 0rpm to 3000rpm
PP	60F4h	Following error actual value	INTEGER32	RO	pulse
PP	60FCh	Position demand value	INTEGER32	RO	pulse
ALL	60FDh	Digital input	UNSIGNED32	RO	-
ALL	60FEh	Digital output	UNSIGNED32	RO	-
PV	60FFh	target_velocity	INTEGER32	RW	0.1rpm

注意：PP 为协议位置模式，PV 为协议速度模式，PT 为转矩模式，HM 为回原点模式，ALL 为全部模式。

3.5 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 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。而本文描述的 CANopen 指令语句，皆以 COB-ID + DATA 的格式呈现，例如指令“601 2B 17 10 00 00 00 00 00”中，“601”为 COB-ID，后续的 8 个 byte 为数据域。

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。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

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 （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

3.6 CANopen 报文简介

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	Emergency Message	故障信息传输
SYNC	Synchronization Message	同步信息：多个 CAN 节点的同步
NMT	Network Management	网络服务：例如，可以同时激活所有的 CAN 节点

3.6.1 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	DLC	Byte0	Byte1
0x000	2	命令字	Node-ID

COB-ID 为 0x000，Byte0 为命令字，占一个字节，Byte 1 为 CANopen 网络设备节点地址，0 表示广播。

例：发送：000h 01h 06h　// 将节点 6 置为可操作状态；
　　发送：000h 01h 00h　// 将所有节点置为可操作状态；

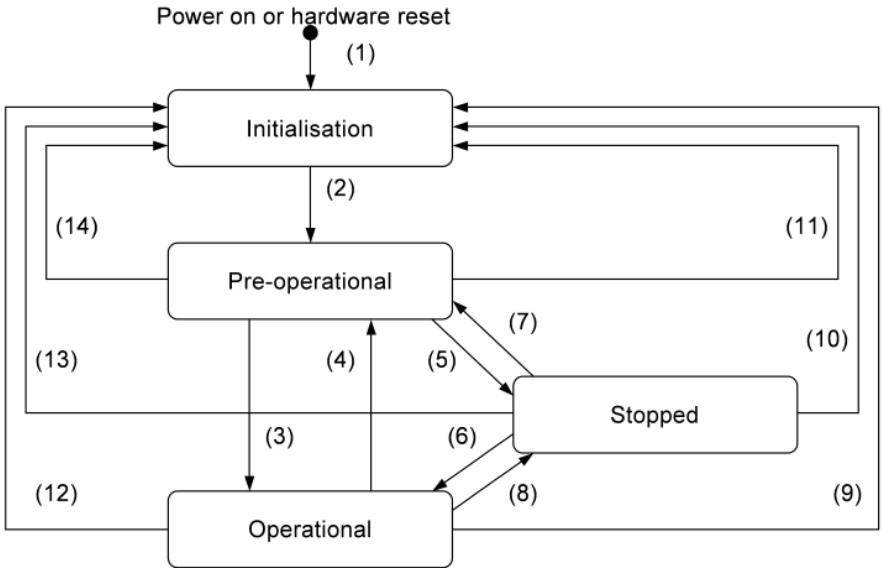
命令字具体格式如下：

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

- 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。节点保护分为心跳报文（Heartbeat）和心跳监护（Life Guard）。

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

- 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 2 的驱动器在接入 CAN 总线后会主动发送报文：0x702 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：发送：602 2B 17 10 00 00 00 00 00　//禁止心跳报文

例 2：发送：602 2B 17 10 00 64 00 00 00　//使能心跳报文，心跳间隔 100ms

COB-ID	DLC	Byte0	状态含义
0x700+Node_ID	1	0x00	BOOTUP 启动状态
		0x04	STOPPED 停止
		0x05	OPERATIONAL 可操作
		0x7F	PRE-OPERATIONAL 预操作

- 心跳监测（Life Guard）

主站可以通过 Life Guard 来监测从站状态：

	COB-ID	DLC	Byte0
主站问	0x700+Slave_ID	0	-
从站答	0x700+Slave_ID	1	State

State 状态字节含义见上一节，低 7 位为状态含义，最高位为翻转位，这个位的值必须在两个连续的响应中交替出现，例程如下：

例 1：上电监控从站状态：从站处于 PRE-OPERATIONAL 状态（以 2 号 ID 为例）

发送：702　　// Life guard 状态询问
回复：702　FF　　//状态回复,1 个字节长度
发送：702　　// Life guard 状态持续询问
回复：702　7F　　//状态回复,最高位翻转

例 2：主站发送激活后：从站处于 OPERATIONAL 状态（以 2 号站为例）

发送：000　01 02　//激活节点 2
发送：702　　// Life guard 状态询问
回复：702　85　　//状态回复,1 个字节长度
发送：702　　// Life guard 状态持续询问
回复：702　05　　//状态回复,最高位翻转

3.6.2　EMCY（Emergency Message 紧急报文）

驱动器在正常情况下，错误码 603Fh 值为 0。驱动器在故障报警时，会通过紧急报文主动上传一次错误码，同时 603Fh 显示相关错误内容。错误码上传模式单轴驱动器可

配置为定时上传模式，上传时间由驱动器内部参数控制（Fn8E）。紧急报文格式如下：

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

报警发生后，详细信息请参见**第四章 报警信息**。建议用户通过报警码排查报警原因，软报警可以通过控制字 6040h、Fn 参数或者 IO 清除。

3.6.3 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），主站问，从站答。主要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，比如模式配置、PDO 配置等。

指令结构如下：

<table>
<tr><td colspan="2" rowspan="2">COB_ID</td><td>Byte0</td><td>Byte 1</td><td>Byte 2</td><td>Byte 3</td><td>Byte 4</td><td>Byte 5</td><td>Byte 6</td><td>Byte 7</td></tr>
<tr><td>CMD</td><td colspan="2">INDEX</td><td rowspan="9">SUB INDEX</td><td colspan="4">DATA</td></tr>
<tr><td rowspan="4">问</td><td rowspan="4">0x600+ Node_ID</td><td>0x23 写 4byte</td><td rowspan="8" colspan="2">Fn 参数地址
Dn 参数地址
Pn 参数地址
CANopen 对象字典 Index</td><td rowspan="2">DATA1</td><td>DATA2</td><td>DATA3</td><td>DATA4</td></tr>
<tr><td>0x2B 写 2byte</td><td rowspan="2">0x00</td><td rowspan="2">0x00</td><td rowspan="2">0x00</td></tr>
<tr><td>0x2F 写 1byte</td><td rowspan="2">0x00</td></tr>
<tr><td>0x40 读数据</td><td>0x00</td></tr>
<tr><td rowspan="4">答</td><td rowspan="4">0x580+ Node_ID</td><td>0x43 回 4byte</td><td rowspan="4">DATA1</td><td>DATA2</td><td>DATA3</td><td>DATA4</td></tr>
<tr><td>0x4B 回 2byte</td><td rowspan="2">0x00</td><td rowspan="2">0x00</td><td rowspan="2">0x00</td></tr>
<tr><td>0x4F 回 1byte</td></tr>
<tr><td>0x60 写回复</td><td>DATA2</td><td>DATA3</td><td>DATA4</td></tr>
<tr><td colspan="2">0x80 错误码</td></tr>
</table>

例 1：读单轴驱动器速度 Dn00（若 CAN-ID 为 2）

发送：602 40 00 30 00 00 00 00 00 // 读当前速度

回复：582 4B 00 30 00 64 00 00 00 // 转速为 100rpm

例 2：写单轴驱动器内部速度 Fn33（若 CAN-ID 为 2）

发送：602 2B 33 20 00 C8 00 00 00 // Fn33 写入 200RPM

回复：582 60 33 20 00 C8 00 00 00 // 回复

注意：在 CANopen 指令中，INDEX 和 DATA 都为低字节在前。

3.6.4 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，实现实时数据的传输，只传不回。1 个 PDO 单次最多传输 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	
RPDO1~4	TPDO1~4	RPDO1~4	TPDO1~4
1600h~1603h	1A00h~1A03H	1400h~1403h	1800h~1803h

单轴驱动器支持设置 4 个 TPDO 或 RPDO，支持断电保存 PDO 的设置，信息保存在 Fn100~Fn17C 中。每个 PDO 配置信息占用 13 个 Fn 寄存器。4 个 RPDO 的配置信

息首寄存器分别是 Fn100、Fn110、Fn120、Fn130；4 个 TPDO 的配置信息首寄存器分别是 Fn140、Fn150、Fn160、Fn170。

PDO 每组寄存器结构如下（以 TPDO1 为例，数值为十进制）：

Fn140	Fn141	Fn142	Fn143-144	Fn145	Fn146
上传对象数量	TPDO1_ID	传输类型	上报时间	SUB1 索引	SUB1 子索引
Fn147	Fn148	Fn149	Fn14A	Fn14B	Fn14C
SUB2 索引	SUB2 子索引	SUB3 索引	SUB3 子索引	SUB4 索引	SUB4 子索引

双轴驱动器支持每个轴设置4个TPDO或RPDO，后续会支持断电保存PDO的设置。信息将保存在Fx.7.000~Fx.7.07F中。每个PDO配置信息占用13个Fn寄存器。

1#轴4个RPDO的配置信息首寄存器分别是F0.7.000、F0.7.010、F0.7.020、F0.7.030；4个TPDO的配置信息首寄存器分别是F0.7.040、F0.7.050、F0.7.060、F0.7.070。

2#轴4个RPDO的配置信息首寄存器分别是F1.7.000、F1.7.010、F1.7.020、F1.7.030；4个TPDO的配置信息首寄存器分别是F1.7.040、F1.7.050、F1.7.060、F1.7.070。

PDO 每组寄存器结构如下（以 1#轴 RPDO1 为例，数值为十进制）：

F0.7.000	F0.7.001	F0.7.002	F0.7.003/ F0.7.004	F0.7.005	F0.7.006
上传对象数量	TPDO1_ID	传输类型	上报时间	SUB1 索引	SUB1 子索引
F0.7.007	F0.7.008	F0.7.009	F0.7.00A	F0.7.00B	F0.7.00C
SUB2 索引	SUB2 子索引	SUB3 索引	SUB3 子索引	SUB4 索引	SUB4 子索引

- RPDO配置

RPDO主要用于快速变化数据的传输，例如速度、控制字等对象字典。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 20 00 FF 60	60FFh映射至RPDO1 sub1
	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 10 00 40 60	6040h映射至RPDO2 sub1
	3	602 23 01 16 02 20 00 FF 60	60FFh映射至RPDO2 sub2
	4	602 2F 01 16 00 02 00 00 00	RPDO2 enable

由上例，发送“202 64 00 00 00”表示通过RPDO1发送给60FFh的速度值为100。发送“302 06 00 64 00 00 00”表示通过PRDO2发送给6040h的值为6，发送给60FFh的值为100，分别对应。数据域长度为所配置对象字典的总长度。

1600h~1603h的子索引（=0）表示RPDO的启用或停止，当DATA1为0时为停止，DATA1不为0时，表示当前RPDO所配的对象字典数量，并且启动当前RPDO。

1600h~1603h的子索引（>0）表示RPDO的对象字典序列，其数据域DATA1表示该对对象字典的长度，一般为8位、16位或32位，数据域DATA2表示对象字典子索引，数据域DATA3~DATA4为对象字典。

- TPDO配置

TPDO用于快速传输用户需求的信息，例如位置、状态等对象字典。TPDO有异步触发、同步触发两类触发方式，可通过SDO配置相关对象字典来修改。

例如，以下TPDO1和TPDO2的一种配置，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 32 00 00 00	设置触发时间50ms	
	5	602 2F 00 18 03 32 00 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6063H和6069H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 63 60	6063h	
	3	602 23 01 1A 02 20 00 69 60	6069h	
	4	602 2F 01 1A 00 02 00 00 00	TPDO2 enable	
	5	602 2F 01 18 05 32 00 00 00	设置触发时间50ms	
	6	602 2F 01 18 03 32 00 00 00		
	7	602 2F 01 18 02 FE 00 00 00	254上报方式	

1A00h~1A03h的数据结构参考1600h~1603h。

1800h~1803h的子索引（=5，=3）表示TPDO的触发时间，触发时间在DATA1-DATA4里，时间单位为ms；子索引（=2）表示TPDO的触发方式，触发方式在DATA1中，其中1-240方式为同步触发（响应同步帧SYNC）；254表示以触发时间定时触发；255方式表示事件触发（状态变化上传）。

3.7 CANopen 配置例程

3.7.1 协议速度模式（Profile Velocity Mode）

驱动器可以接收主站速度指令运行，该模式下，必须设置的对象字典有：6060H、60FFH、6083H、6084H（6083/6084默认1000ms）。

以CAN节点2为例，以下为配置步骤。该例程中配置了1个RPDO：0x60FF映射至RPDO3，配置了2个TPDO，用户可根据需求自行设置PDO映射。

STEP1. 确认驱动器参数：1、CANopen控制模式；2、通讯修改参数不保存：

单轴：Fn00=20；Fn99=1；

双轴：F0.1.002=20；F1.1.002=20；Pn5.00A=0；

STEP2. 节点置为可操作状态:

发送：000 01 02——将节点2置为可操作状态；

STEP3. RPDO 映射：

以下示例配置1个RPDO：

RPDO3：对应COB-ID为0x402

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

发送：602 23 02 16 01 20 00 FF 60 // 60FFh映射

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

STEP4. TPDO映射：

以下示例配置2个TPDO：将【velocity_sensor_actual_value: 6069h】和【Position actual value:6063h】配置为TPDO1，将【Statusword:6041h】和【双轴驱动器1#轴负载率：5002h】配置为TPDO2，TPDO上传方式为同步上传。

TPDO1：对应COB-ID为0x182

发送：602 2F 00 1A 00 00 00 00 00

发送：602 23 00 1A 01 20 00 69 60 //实际转速对应对象字典6069h

发送：602 23 00 1A 02 20 00 63 60 //码盘位置对应对象字典6063h

发送：602 2F 00 1A 00 02 00 00 00

发送：602 2F 00 18 02 01 00 00 00 //同步上传模式

发送：602 2F 00 18 03 02 00 00 00

发送：602 2F 00 18 05 02 00 00 00

TPDO2：对应COB-ID为0x282

发送：602 2F 01 1A 00 00 00 00 00

发送：602 23 01 1A 01 10 00 41 60 //状态字对应对象字典0x6041

发送：602 23 01 1A 02 10 00 02 50 //双轴驱动器1#轴负载率0x5002

发送：602 2F 01 1A 00 02 00 00 00

发送：602 2F 01 18 02 01 00 00 00 //同步上传模式

发送：602 2F 01 18 03 02 00 00 00

发送：602 2F 01 18 05 02 00 00 00

STEP5. 设置

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

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

2. 设置加速度和减速度【Profile acceleration:6083h；Profile deceleration:6084h】(单位时间：ms，从0rpm到3000rpm的时间，最大可设200K)：

   发送：602 23 83 60 00 E8 03 00 00 //设置6083h加速时间为1s

   发送：602 23 84 60 00 E8 03 00 00 //设置6084h减速时间为1s

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

发送：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，首次使能电机，必须先依次给控制字6040h写入0x6、0x7、0xF；后续断使能写入0x7；使能则写入0xF；

发送：602 2B 40 60 00 06 00 00 00 //设置6040h为6，使电机准备

发送：602 2B 40 60 00 07 00 00 00 //设置6040h为7，使电机失能

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

5. 发送同步帧得到TPDO包含的对象字典状态。

发送： 080 //发送同步帧，配置同步上传的TPDO将触发

返回1：182 byte0-7 //TPDO1上报信息

返回2：282 byte0-3 //TPDO2上报信息

■ **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. 目标速度60FFh为32位有符号数，正负对应电机正反转。

3.7.2 协议位置模式（Profile Position Mode）

驱动器接收主站位置指令可控制电机到达目标位置，该模式下，必须设置的对象字典有：6060H、607AH、6081H、6083H、6084H、6093H（6083/6084 默认为 1000ms，6093 默认 1 比 1）。

以 CAN 节点 2 为例，以下为配置步骤。该例程中配置了 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。用户可根据需求自行设置 PDO 映射。

STEP1. 确认驱动器参数：1、CANopen控制模式；2、通讯修改参数不保存：

单轴：Fn00=20；Fn99=1；

双轴：F0.1.002=20；F1.1.002=20；Pn5.00A=0。

STEP2. 节点置为可操作状态:

发送：000 01 02——将节点2置为可操作状态；

STEP3. RPDO 映射:

配置4个RPDO:。

RPDO1：对应COB-ID为0x202

发送：602 2F 00 16 00 00 00 00 00 //RPDO1 stop
发送：602 23 00 16 01 10 00 40 60 //6040h
发送：602 2F 00 16 00 01 00 00 00 // RPDO1 enable
RPDO2：对应COB-ID为0x302
发送：602 2F 01 16 00 00 00 00 00 //RPDO2 stop
发送：602 23 01 16 01 20 00 83 60 //6083h and 6084h
发送：602 23 01 16 02 20 00 84 60 //6083h and 6084h
发送：602 2F 01 16 00 02 00 00 00 // RPDO2 enable
RPDO3：对应COB-ID为0x402
发送：602 2F 02 16 00 00 00 00 00 //RPDO3 stop
发送：602 23 02 16 01 20 00 7A 60 //607Ah
发送：602 2F 02 16 00 01 00 00 00 // RPDO3 enable
RPDO4：对应COB-ID为0x502
发送：602 2F 03 16 00 00 00 00 00 //RPDO4 stop
发送：602 23 03 16 01 20 00 81 60 //6081h
发送：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
   发送：602 2F 60 60 00 01 00 00 00 //设置6060h为1（Profile Position Mode）
2. 设置目标位置【Target position:607Ah】. (单位: pulse)
   发送：402 60 EA 00 00 //60000脉冲
3. 设置速度【Profile velocity:6081h】. (单位: 0.1rpm)
   发送：502 B8 0B 00 00 //300rpm
4. 设置加速度和减速度【Profile acceleration:6083h; Profile deceleration:6084h】(单位时间：ms，从0rpm到3000rpm的时间，最大可设8K)：
   发送：302 E8 03 00 00 E8 03 00 00
5. 设置电子齿轮比【Position factor：6093h；Sub1：Numerator；Sub2：Divisor】
   发送：602 2B 93 60 01 01 00 00 00 //设置电子齿轮比分子
   发送：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：.
   发送：602 2B 40 60 00 06 00 00 00 // 设置6040h为6
   发送：602 2B 40 60 00 07 00 00 00 // 设置6040h为7
   发送：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：
   增量位置触发：
   发送：202 0F 00 //6040的bit4先清零
   发送：202 5F 00 //增量位置触发

发送：202 4F 00 //bit4清零，为下次启动做准备

绝对位置触发：

发送：202 1F 00 //绝对位置触发

发送：202 0F 00 //清零

8. 确认位置实际值【Position actual value: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6040h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。

2、上位机可根据驱动器的状态字6041h来判断是否要给伺服新数据。当驱动器的状态字6041的bit10为1时，表示电机已到达位置，驱动器可以接收新的位置数据和命令了；如果为0，表示电机正在执行中。主站可以设置速度监控阈值：

- 设置【Position window: 6067h】位置到达区。
- 设置【Position window time: 6068h】位置到达时间。

3、在增量位置模式下，电机在执行命令中接收到新的位置指令，则新命令暂不被采纳，而会保存在缓存区。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。

4、在增量位置模式下，驱动器支持位置触发后的速度修改。

5、在绝对位置模式下，电机在执行命令中接收到新的位置指令，则执行最新的位置指令，原位置不再执行。

3.7.3 转矩模式（Torque Profile Mode）

驱动器支持转矩模式运行，在该模式下，必须设置的对象字典有：6060H、6087H、6071H。以 CAN 节点 2 为例，以下为配置：

STEP1. 确认驱动器参数：1、CANopen控制模式；2、通讯修改参数不保存：

单轴：Fn00=20；Fn99=1；

双轴：F0.1.002=20；F1.1.002=20；Pn5.00A=0.

STEP2.节点置为可操作状态:

发送：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】为转矩模式：

   发送：602 2F 60 60 00 04 00 00 00 //设置6060h为4（转矩模式）

2. 设置加减速【Torque slope:6087h】：

   发送：602 2F 87 60 00 E8 03 00 00 //设置6087H为1000；

3. 设置目标转矩【Target torque:6071h】：

   发送：602 2F 71 60 00 64 00 00 00 // 100对应电机转矩的10%；

4. 设置状态字【Controlword:6040h】来使能电机：

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

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

发送：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，**单轴**驱动器最大限制速度为 Fn95，地址为 0x2095。双轴驱动器最大限制速度为 F0/1.4.026，地址分别为 0x2426 和 0x3426。

3.7.4 回零点模式（Homing Mode）

用户可设置速度、加减速等配置回零模式。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

■ 以CAN节点2为例，配置步骤如下：

STEP1. **确认驱动器参数：1、CANopen控制模式；2、通讯修改参数不保存：**

单轴：Fn00=20；Fn99=1；

双轴：F0.1.002=20；F1.1.002=20；Pn5.00A=0；

STEP2. 节点置为可操作状态:

发送：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：

发送：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】：

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

3. 设置回零速度【Homing speeds:6099h】：

发送：602 23 99 60 01 D0 07 00 00
发送：602 23 99 60 02 C8 00 00 00

4. 设置回零加减速【Homing acceleration:609Ah】：
   发送：602 23 9A 60 00 E8 03 00 00
5. 设置电子齿轮比【Position factor:6093h】：
   发送：602 23 93 60 01 01 00 00 00 //设置6093H，齿轮比分子1；
   发送：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：.
   发送：602 2B 40 60 00 06 00 00 00 //设置6040h为6
   发送：602 2B 40 60 00 07 00 00 00 //设置6040h为7
   发送：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：
   发送：602 2B 40 60 00 0F 00 00 00
   发送：602 2B 40 60 00 5F 00 00 00
   发送：602 2B 40 60 00 4F 00 00 00

注意：零点限位开关在输入口定义中，应设为18。

3.8 CAN 自由协议

3.8.1 速度模式（单轴）

用户可在内部速度模式（Fn00=1）下，直接通过 SDO 读写驱动器 Fn 功能参数控制电机。读取 Dn 状态观测参数，以及设置 TPDO 上报信息可辅助用户的控制。示例如下：

■ **相关参数：**

Index	Name	Type	Attr.	unit
2000h	Fn00 控制模式	UNSIGNED16	RW	
2033h	Fn33 内部速度	INTEGER16	RW	1rpm
2038h	Fn38 使能	UNSIGNED16	RW	
2099h	Fn99 通讯指令保存开关	UNSIGNED16	RW	
20AAh	FnAA 速度模式减速倍率	UNSIGNED16	RW	

■ **配置步骤（以CAN节点2为例）：**

用户可通过手持面板修改，也可通过通讯修改参数，具体步骤如下：

STEP1. 确认驱动器参数：

Fn00 = 1（速度模式）；

Fn33 = 0（初始上电速度为0）；

Fn38 = 0（初始状态不使能）；
Fn99 = 1（通讯修改参数不保存）.

STEP2. 设置：

1. 设置电机速度Fn33：
   发送：602 2B 33 20 00 64 00 00 00 //设置速度为100rpm
2. 使能电机：
   发送：602 2B 38 20 00 01 00 00 00 //设置 Fn38=1，使电机使能；

3.读 Dn00 电机转速：
   发送：602 40 00 30 00 00 00 00 00 //读速度，单位 rpm

■ **注意事项：**

1、部分 CANopen 对象字典也可在非 CANopen 模式下读取，例如读编码器绝对位置，在 Modbus 下可以读 Dn2E 和 Dn2F，而在 CAN 模式下，可以读对象字典 6063H：
   发送：602 40 63 60 00 00 00 00 00
   回复：582 43 63 60 00 D1 D2 D3 D4
   以上，D1-D4 分别为低位-高位；

2、DS、RS 型伺服驱动器支持本手册中对象字典，如用户需求也可定制增加相应功能，具体请联系和利时电机业务员及技术员。

3、DS、RS 型伺服驱动器的 EDS 文件请联系和利时业务员及技术员获取。

3.8.2 速度模式（双轴）

用户可在内部速度模式（F0.1.002 / F1.1.002=1）下，直接通过 SDO 读写驱动器 Fn 功能参数控制电机。读取 Dn 状态观测参数，以及设置 TPDO 上报信息可辅助用户的控制。示例如下：

■ **相关参数：**

Axis No.	Index	Name	Type	Attr.	unit
1	2102h	F0.1.002 1#轴控制模式	UNSIGNED16	RW	
	2318h	F0.3.018 1#轴内部速度	INTEGER16	RW	1rpm
	2100h	F0.1.000 1#轴使能	UNSIGNED16	RW	
2	3102h	F1.1.002 2#轴控制模式	UNSIGNED16	RW	
	3318h	F1.3.018 2#轴内部速度	INTEGER16	RW	1rpm
	3100h	F1.1.000 2#轴使能	UNSIGNED16	RW	
ALL	450Ah	Pn5.00A 通讯指令保存开关	UNSIGNED16	RW	

■ **配置步骤（以1#轴CAN节点2为例）：**

用户可通过手持面板修改，也可通过通讯修改参数，具体步骤如下：

STEP1. 确认驱动器参数：

F0.1.002 = 1（速度模式）；
F0.3.018 = 0（初始上电速度为0）；
F0.1.000 = 0（初始状态不使能）；
Pn5.00A = 0（通讯修改参数不保存）.

STEP2. 设置：

1. 设置电机速度F0.3.018：
发送：602 2B 18 23 00 64 00 00 00 //设置1#电机速度为100rpm
2. 使能电机：
发送：602 2B 00 21 00 01 00 00 00 //置 1，使 1#电机使能；
3.读 D0.000 电机转速：
发送：602 40 00 50 00 00 00 00 00 //读 1#轴速度，单位 rpm

3.8.3 差速模式（双轴）

双轴（二合一）驱动器支持自定义的差速指令，即通过一条 CAN 报文同时控制双轴动作，从而达到双轴完全同步的控制。

该模式请设置控制模式 F0/1.1.002 为 20（CANopen 模式），根据需求设置差速上报开关（Pn5.00D）、双轴轮径（Pn6.00A / Pn6.00B）、定义左右轮（Pn6.00C）。

设置完毕后，第一步，先使能驱动器：

发送：602 2B 00 21 00 01 00 00 00 //置 1，使 1#电机使能；
发送：602 2B 00 31 00 01 00 00 00 //置 1，使 2#电机使能；

第二步，写差速指令，格式如下：

ID	指令				线速度		角速度	
0x600+Node_ID	EA	00	00	00	ByteVL	ByteVH	ByteAL	ByteAH

上报信息格式如下：

ID	指令				线速度		角速度	
0x180+Node_ID	EA	00	00	00	ByteVL	ByteVH	ByteAL	ByteAH

其中，线速度单位为 mm/s，角速度单位为 mrad/s，两个字节组成的线速度、角速度均低字节在前。

3.8.4 双读双写（双轴）

双轴（二合一）驱动器支持 1 条指令同时读写两个轴的相同 Fn、Dn 参数，指令地址以 1#轴为基准。指令格式如下：

COB_ID		Byte0	Byte 1	Byte 2	Byte 3	Byte 4	Byte 5	Byte 6	Byte 7
		CMD	INDEX		SUB	DATA			
问	0x600+	0xE8 读	Fn 参数地址		INDEX	0x00	0x00	0x00	0x00

	Node_ID	0xE9 写	Dn 参数地址		DATA1	DATA2	DATA3	DATA4
答	0x580+	0xE8 读回复	Pn 参数地址		DATA1	DATA2	DATA3	DATA4
	Node_ID	0x60 写回复			DATA1	DATA2	DATA3	DATA4

例 1：读取两轴编码器单圈位置：

发送：601 E8 04 50 00 00 00 00 00 //读 0x5004 和 0x5104，以 1#轴地址为基准

返回：581 E8 04 50 00 B7 13 AB 12 //返回 D0.04=5047, D1.04=4779

例 2：写两轴速度值：

发送：601 E9 18 23 00 36 00 CA FF //写 0x2318 和 0x3318 分别为 54 和-54

返回：581 60 18 23 00 64 00 CA FF //写返回

由于双读双写协议可以修改驱动器的 Fn 参数，因此也可以使用该协议进行速度模式的控制。

第四章 报警信息

4.1 单轴报警信息

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、或者对象字典0x603F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
ERR_HE	0x110E	第一段速找零信号报警
ERR_oE	0x160E	第二段速找零信号报警
ERR_IE	0x120E	回原点找 Z 报警

另外，如果驱动器使能正反限位功能（FnFF=1），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：

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

4.2 双轴报警信息

驱动器产生报警时，自动通过EMCY向主站报告，用户还可读取相关Dn0/1.12参数或者对象字典0x603F来获知报警信息。报警信息定义如下（X=0为1#轴，X=1为2#轴）：

驱动器显示	传输码	含义
ERX_01	0x0001	编码器故障 ABZ 报警
ERX_02	0x0002	编码器故障 UVW 报警
ERX_03	0x0003	位置超差
ERX_04	0x0004	失速
ERX_05	0x0005	电流采样（中点）故障
ERX_06	0x0006	过载
ERX_07	0x0007	欠压
ERX_08	0x0008	过压
ERX_09	0x0009	过流
ERX_0A	0x000A	放电报警瞬时功率大
ERX_0B	0x000B	放电回路频繁动作平均功率大
ERX_0C	0x000C	参数读写异常
ERX_0D	0x000D	输入口功能定义重复