

Kinco®

Kinco CM 系列通讯型步进驱动器

Kinco CM Series Stepping Motor Drive



用户手册

USER'S MANUAL

Shenzhen Kinco Automation Co.,Ltd.

第一章 产品概述	1
1.1 产品确认.....	1
1.2 驱动器铭牌.....	1
1.3 产品型号说明.....	2
1.3.1 步进驱动器型号说明.....	2
1.3.2 产品特性.....	2
第二章 使用事项及安装要求	3
2.1 注意事项.....	3
2.2 环境条件.....	3
2.3 安装方向与机械尺寸.....	4
第三章 驱动器接口及连线	7
3.1 CM880A 驱动器.....	7
3.1.1 驱动器接口简介.....	7
3.1.2 I/O 接口.....	8
3.1.3 RS232 通讯接口.....	9
3.2 CM880B 和 CM880C 驱动器.....	10
3.2.1 驱动器接口简介.....	11
3.2.2 I/O 接口.....	11
3.2.3 拨码开关设置.....	12
3.2.4 RS232 通讯接口.....	12
3.3 CM422B 和 CM422C 驱动器.....	13
3.3.1 驱动器接口简介.....	14
3.3.2 I/O 接口.....	14
3.3.3 拨码开关设置.....	14
3.4 Console 线.....	15
3.4.1 转接线的针脚连线顺序如下:	15
第四章 驱动器外部接线图	16
第五章 步进调试软件使用说明	17
5.1 软件安装.....	17
5.2 快速入门.....	17
5.2.1 调试软件连接.....	17
5.2.2 调试软件联机.....	18
5.3 菜单介绍.....	21
5.4 驱动器控制.....	22
5.4.1 驱动器配置.....	22
5.4.2 电机配置.....	23
5.4.3 电流设置.....	24
5.4.4 细分设置.....	24
5.5 基本操作.....	25
5.6 I/O 操作.....	27
5.7 数据字典.....	29
5.8 驱动器属性.....	30
5.9 实时错误.....	31
5.10 历史错误.....	31
5.11 初始化/保存.....	32
第六章 输入输出口操作	33
6.1 数字信号输入.....	33
6.1.1 数字信号输入极性控制.....	33
6.1.2 数字信号输入功能.....	34
6.1.3 数字输入口接线.....	37

6.2 数字信号输出.....	38
6.2.1 数字信号极性控制.....	38
6.2.2 数字信号输出地址以及功能.....	38
6.2.3 数字输出口接线.....	39
第七章 模式操作.....	42
7.1 位置模式 (“1”模式)	42
7.2 速度模式 (“3”模式)	43
7.3 脉冲控制模式 (“-4”模式)	44
7.4 原点控制模式 (“6”模式)	44
7.5 多段位置控制模式 (“1”模式)	45
7.5.1 多段位置控制.....	45
7.6 多段速度控制模式 (“3”模式)	47
第八章 通讯功能.....	48
8.1 RS232 通讯.....	48
8.1.1 RS232 通讯硬件接口.....	48
8.1.2 RS232 通讯参数.....	48
8.1.3 自由传输协议.....	49
8.1.4 数据协议.....	49
8.1.5 RS232 通讯地址.....	52
8.2 RS485 通讯.....	52
8.2.1 RS485 通讯硬件接口.....	52
8.2.2 RS485 通讯参数.....	52
8.2.3 Modbus RTU 通讯协议.....	53
8.2.4 Modbus RTU 常用功能码简介.....	54
8.2.5 Modbus 常用通讯地址.....	56
8.3 CAN 通讯.....	56
8.3.1 硬件介绍.....	57
8.3.2 CAN 通讯参数.....	58
8.3.3 EDS 文件.....	58
8.3.4 CANopen 通讯协议.....	58
8.3.10 CANopen 常用通讯地址.....	69
第九章 报警排除.....	70
9.1 报警信息.....	70
9.2 历史错误窗口.....	71
9.3 自检错误状态字定义.....	72
第十章 附录.....	73
附录一 CANopen 总线通讯范例.....	73
驱动器与 F1 PLC 利用 CANopen 总线进行通讯.....	73
驱动器与 Peak CAN 利用 CANopen 总线进行通讯.....	81
附录二 RS485 串口通讯范例.....	85
驱动器与 KINCO 触摸屏 Modbus 协议通讯.....	85
驱动器与调试工具 Modbus 协议通讯.....	88
驱动器与西门子 S7-200Modbus 协议通讯.....	91
附录三 RS232 串口通讯范例.....	93
CM880A 与 KINCO 触摸屏自由协议通讯.....	93
CM880A 与串口调试工具自由协议通讯.....	94
Console 配置线.....	97
附录四 找原点方式.....	98
附录五 利用 Kincostep 调试软件导入和导出驱动器内参数.....	104
附录六 常用对象工程单位与内部单位换算关系表.....	107
附录七 常用对象列表.....	107

修改记录

版本	日期	修订描述
1.0	2015-2-26	1.0 版初稿
1.2	2015-6-5	1.修改 P26 例子 4 2.修改 P27 查找图 RS232 波特率
1.3	2016-3-11	添加 CM422B, CM422C, CM880B 和 CM880C 型号驱动器说明
1.4	2016-6-14	添加与 CM422B, CM422C 匹配的 Modbus 信息
1.5	2016-7-6	校对
1.6	2018-12-3	修正

安全事项

为了避免人身伤害和财产损失，请在调试及使用驱动器前仔细阅读以下安全信息。

以下安全措施必须严格遵守：

- 驱动器正常工作时内部最高将有 70VDC 左右的低电压，在切断驱动器电源 1 分钟后进行操作，此时指示灯彻底熄灭或确认端子上电压在 36VDC 内，再进行接线或检查，否则可能遭到电击。
- 请勿在驱动器及电机工作时进行接线，否则可能遭到电击。
- 请勿在通电后或驱动器运行时拆开驱动器外壳，否则可能遭到电击。
- 为了避免人身伤害和财产损失，非专业人员不可以对驱动器进行相关操作。
- 安装过程中请遵守相关技术规范和电气安装标准。驱动器必须有良好地接地，接地电缆的截面积不小于 1.25mm^2 。
- 请不要把任何物体放入驱动器内，否则可能造成驱动器损坏。
- 驱动器出现故障需要检修时，请将驱动器送回检修中心。私自打开驱动器或不正确的操作会损坏驱动器。未经允许，私自打开驱动器外壳，保修作废。
- 在废弃驱动器的时候，请按照工业废弃物的标准来处理，以免造成环境的污染。

⚠ 警告：

- 把此驱动器应用于直接涉及人身安全的机械设备（核动力控制、医疗设备、卡车、火车、飞机、如乐和安全防护设备等）时，必须安装防范的安装设备，避免出现可能发生的人身伤害。
- 电子设备均有相应的生命周期。机器设备必须有足够的安全措施，在驱动器失灵的情况下保证人员及设备本身安全。安装或使用驱动器的客户必须自己承担因为机器故障及错误操作驱动器造成的损失。

第一章 产品概述

1.1 产品确认

感谢您选择 Kinco 步进驱动器产品。收到产品后，请仔细核对以下项目。

- 查看包装箱及机身上的铭牌，确认驱动器型号与您订购的型号一致。
- 打开产品包装后，请确认产品有无损坏或缺少零件等异常情况。
- 请确认驱动器上所有固定螺丝有无松动。
- 请按产品清单核对收到的产品，如有缺少，请及时联系客服人员。

表 1-1 包装清单

产品清单	
名称	数量
驱动器	1 台
CM 系列步进驱动器使用指南	1 张
服务指南	1 张
合格证	1 张
长 10mm 直径 1.0mm 绝缘端子	6 个
长 15mm 直径 1.5mm 绝缘端子	6 个
Console 线 (RS232 串口转 RJ45 水晶头线)	1 条

注：Console 线不是标配品，如有需求请联系客服。

1.2 驱动器铭牌



图 1-1 驱动器铭牌

1.3 产品型号说明

1.3.1 步进驱动器型号说明

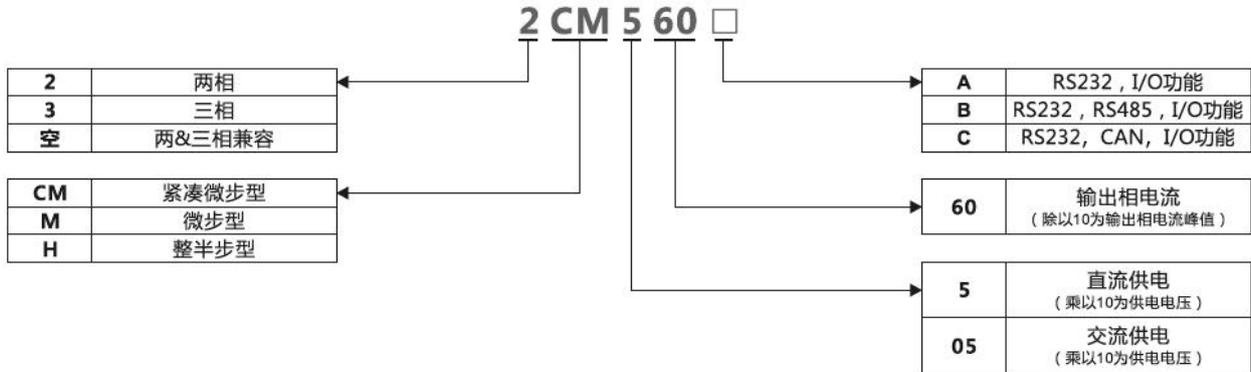


图 1-2 命名规则

1.3.2 产品特性

表 1-4

产品	供电电压 (Vdc)	输出相电流(Apeak)	IO 口	通讯接口	毛重(g)	净重(g)
CM422B	12~40	0.05~2.2	3 路输入	RS485	200	113
CM422C				CAN	200	
CM880A	24~70	0.1~8	6 路输入, 1 路输出	RS232	377	290
CM880B			3 路输入	RS232, RS485	333	253
CM880C				RS232, CAN		

- 支持驱动 2 相及 3 相步进电机。
- 多功能 IO 配置，适应不同控制方式，其中 DIN1 和 DIN2 支持双脉冲（CW/CCW）模式，脉冲方向（P+D）模式和增量式编码器（AB）模式控制。
- 支持 CANOpen 和 Modbus 总线协议控制，优化控制电路。
- 具有电机参数适应功能，发挥电机的最好性能。
- 保护功能：过压，欠压，过热和过流保护。

第二章 使用事项及安装要求

2.1 注意事项

- 电机固定螺丝必须锁紧；
- 固定驱动器时，必须确保每个固定处锁紧；
- 驱动器与电机电缆不能拉紧；
- 电机轴与设备轴安装必须保证对心良好，请使用连轴器或者胀紧套；
- 避免螺丝、金属屑等导电性异物或油等可燃性异物进入步进驱动器内；
- 步进驱动器与步进电机是精密设备，请不要使其坠落或者遭受强力冲击；
- 安全起见，请不要使用有损伤或零件损坏的步进驱动器。

2.2 环境条件

表 2-1 环境条件

环 境	条 件
温度	工作温度：0℃~40℃（不结冰） 储藏温度：-10℃~70℃（不结冰）
湿度	工作湿度：90%RH 以下（无凝露） 储藏湿度：90%RH 以下（无凝露）
空气	室内无日晒、无腐蚀性气体、无易燃性气体、无油气、无尘埃
高度	海拔 1000m 以下
振动	5.9 m/s^2

2.3 安装方向与机械尺寸

步进驱动器在墙壁上安装要垂直放置，上下须预留合适安装间距。使用制动电阻等发热性器件时，要充分考虑到散热情况，保证驱动器的散热空间，确保步进驱动器不受影响。（单位：mm）

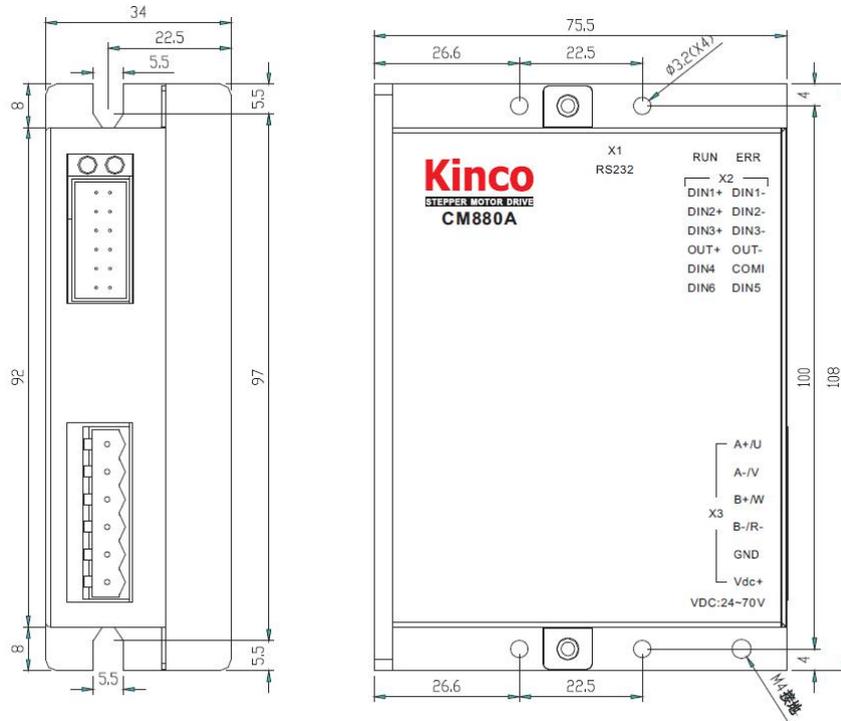


图 2-1 CM880A 尺寸

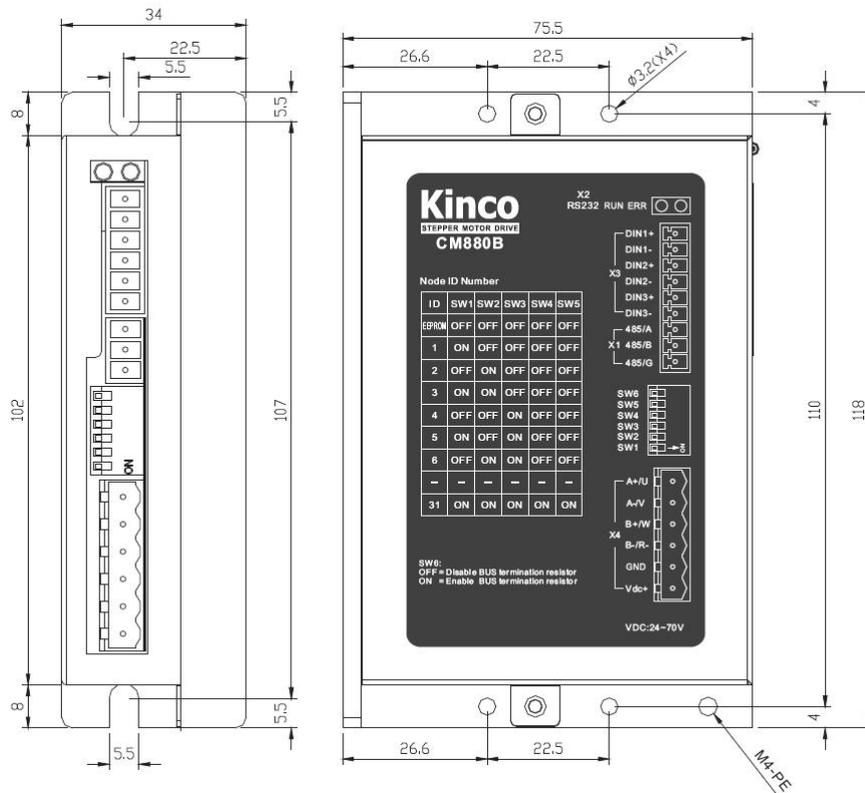


图 2-2 CM880B 尺寸

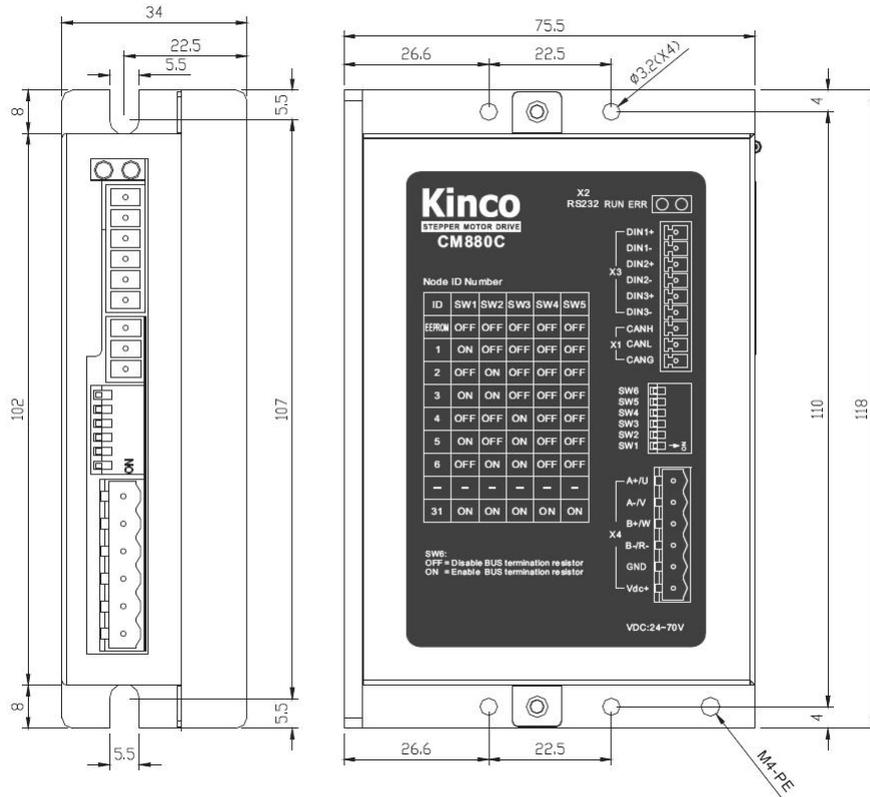


图 2-3 CM880C 尺寸

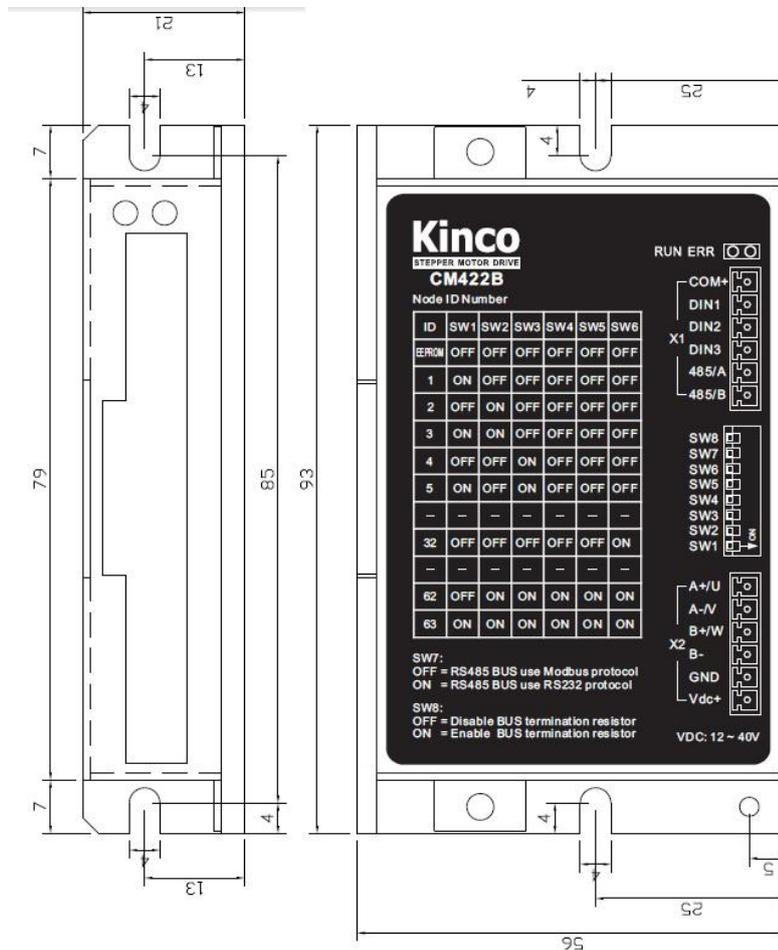


图 2-4 CM422B 尺寸

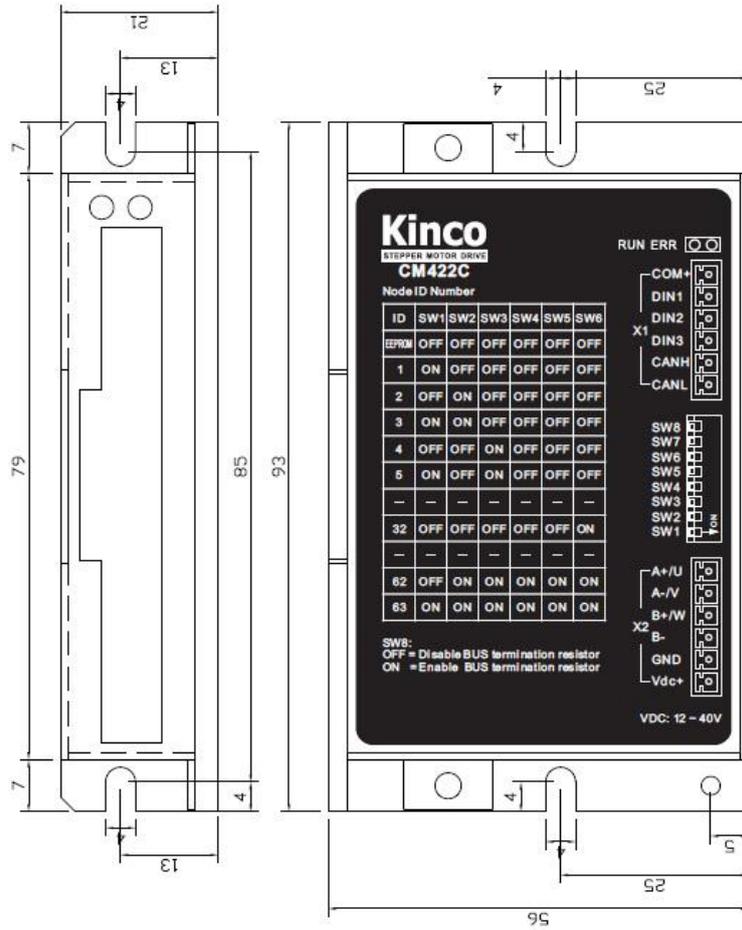


图 2-5 CM422C 尺寸

第三章 驱动器简介

3.1 CM880A 驱动器



图 3-1 CM880A 外观

3.1.1 驱动器接口简介

表 3-1 接口介绍

接口		位号	功能
RS232		X1	RS232 接口
IO 接口	DIN1+	X2	数字信号输入接口
	DIN1-		
	DIN2+		
	DIN2-		
	DIN3+		
	DIN3-		
	DIN4		
	DIN5		
	DIN6		
	COMI		
OUT+	数字信号输出接口		
OUT-			
电源和电机线接口	A+/U A-/V B+/W	X3	二相或三相步进电机接线口

	B-/R-		
	GND Vdc+		功率电源 DC24-70V 输入

3.1.2 I/O 接口

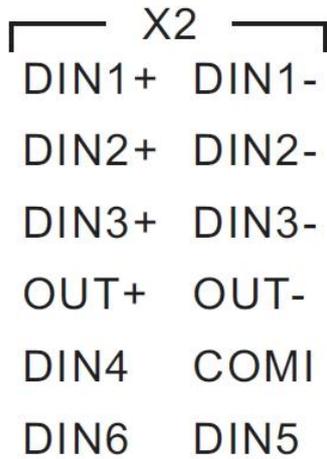


图 3-2 接线图

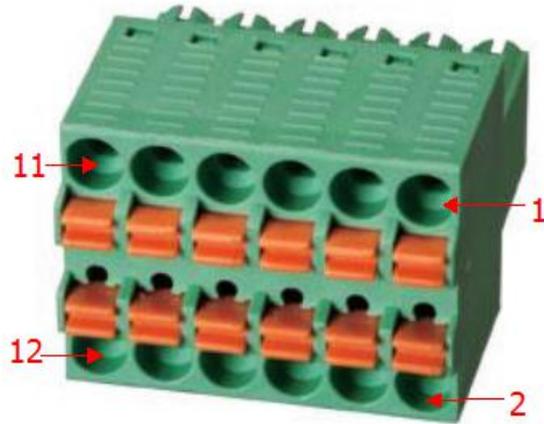


图 3-3 I/O 脚位

表 3-2 I/O 功能定义

名称	引脚	信号	描述	功能
X2 I/O	1	DIN1+	DIN1 正端输入	输入电压：5~24VDC 输入电流：8mA@5VDC, 12mA@24VDC 有效输入信号：大于 3VDC 无效输入信号：小于 1VDC
	2	DIN1-	DIN1 负端输入	
	3	DIN2+	DIN2 正端输入	
	4	DIN2-	DIN2 负端输入	
	5	DIN3+	DIN3 正端输入	
	6	DIN3-	DIN3 负端输入	
	9	DIN4	DIN4 端输入	输入电压：12~24VDC 输入电流：4mA@12VDC, 8mA@24VDC 有效输入信号：大于 8VDC 无效输入信号：小于 5VDC
	12	DIN5	DIN5 端输入	
	11	DIN6	DIN6 端输入	
	10	COMI	DIN4, DIN5, DIN6 输入公共端	最大输出电流：100mA 最大压降：0.8VDC@100mA 最大承受电压：30VDC
7	OUT+	OUT1 正端输出		
8	OUT-	OUT1 负端输出		

注：

- (1) DIN1 和 DIN2 输入使用高速光耦，最高输入频率为 400KHz。
- (2) DIN3~6 输入使用低速光耦，最高输入频率为 1KHz。
- (3) OUT 输出使用低速光耦，最高输出频率为 1KHz。

3.1.3 RS232 通讯接口

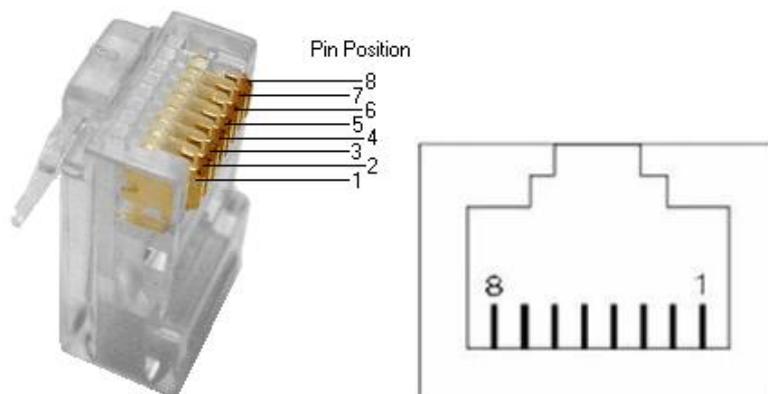


图 3-4 RJ45 插头和插座

表 3-3 RS232 针脚定义

名称	RJ45 针脚	信号	描述	功能
X1 RS232	1	NC	空	RS232 通讯接口
	2	NC	空	
	3	TX	数据发送端	
	4	GND	信号地	
	5	GND	信号地	
	6	RX	数据接收端	
	7	NC	空	
	8	NC	空	

注：可使用 console 线，转换 DB9 针头接口

3.2 CM880B 和 CM880C 驱动器



图 3-5 CM880B 外观



图 3-6 CM880C 外观

3.2.1 驱动器接口简介

表 3-4 接口介绍

接口		位号	功能
RS232		X2	RS232 接口
I/O 接口	DIN1+	X3	数字信号输入接口
	DIN1-		
	DIN2+		
	DIN2-		
	DIN3+		
	DIN3-		
总线通讯接口	485/A	CANH	总线通讯接口 CM880B 是 RS485 接口 CM880C 是 CAN 接口
	485/B	CANL	
	485/G	CANG	
电源和电机线接口	A+/U A-/V B+/W B-/R-	X4	二相或三相步进电机接线口
	GND VDC+		功率电源 DC24-70V 输入

3.2.2 I/O 接口

表 3-5 I/O 功能定义

名称	针脚	信号	描述	功能
X3 I/O	1	DIN1+	DIN1 正端输入	输入电压：5~24VDC 输入电流：8mA@5VDC, 12mA@24VDC 有效输入信号：大于 3VDC 无效输入信号：小于 1VDC
	2	DIN1-	DIN1 负端输入	
	3	DIN2+	DIN2 正端输入	
	4	DIN2-	DIN2 负端输入	
	5	DIN3+	DIN3 正端输入	
	6	DIN3-	DIN3 负端输入	

注：

- (1) DIN1 和 DIN2 输入使用高速光耦，最高输入频率为 50KHz。
- (2) DIN3 输入使用低速光耦，最高输入频率为 1KHz。

3.2.3 总线通讯接口

表 3-6 I/O 功能定义

CM880B 驱动器

名称	针脚	信号	描述	功能
X1	1	485/A	差分信号输入	RS485 为隔离电路，默认波特率为 19200 通讯数据位：8，停止位：1，无奇偶校验位
	2	485/B		
	3	485/G	RS485 的 GND	

CM880C 驱动器

名称	引脚	信号	描述	功能
X1	1	CANH	差分信号输入	CAN 为隔离电路，默认波特率为 500K
	2	CANL		
	3	CANG	CAN 的 GND	

3.2.3 拨码开关设置

表 3-6 驱动器节点 ID 号由拨码开关设置，

ID 号	SW1	SW2	SW3	SW4	SW5
EEPROM	OFF	OFF	OFF	OFF	OFF
1	ON	OFF	OFF	OFF	OFF
2	OFF	ON	OFF	OFF	OFF
3	ON	ON	OFF	OFF	OFF
4	OFF	OFF	ON	OFF	OFF
—	—	—	—	—	—
30	ON	ON	ON	ON	OFF
31	ON	ON	ON	ON	ON

注：当 SW1~SW5 都为“OFF”时，ID 号为 EEPROM 值，这值是指“设备站号（加偏移值）”（0x2FE400 对象）填入的值。

SW6 为总线终端电阻开关，为“ON”时将 120 欧电阻并联在总线上，为“OFF”时为断开。

3.2.4 RS232 通讯接口

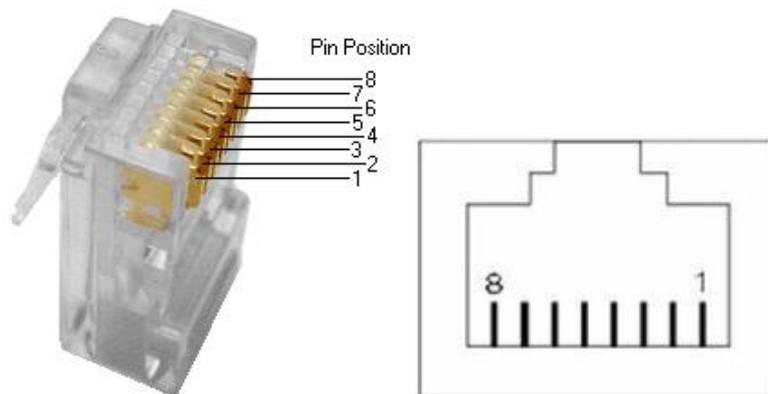


图 3-8 RJ45 插头和插座

表 3-7 RS232 引脚定义

名称	RJ45 引脚	信号	描述	功能
X1 RS232	1	NC	空	RS232 通讯接口
	2	NC	空	
	3	TX	数据发送端	
	4	GND	信号地	
	5	GND	信号地	

	6	RX	数据接收端
	7	NC	空
	8	NC	空

注：可使用 console 线，转换 DB9 针头接口

3.3 CM422B 和 CM422C 驱动器



图 3-9 CM422B 外观



图 3-10 CM422C 外观

3.3.1 驱动器接口简介

表 3-8 接口介绍

接口		位号	功能
I/O 接口	COM+	X1	数字信号输入接口
	DIN1		
	DIN2		
	DIN3		
	485/A		CANH
485/B	CANL		
电源和电机线接口	A+/U A-/V B+/W B-	X2	二相或三相步进电机接线口
	GND Vdc+		功率电源 DC12-40V 输入

3.3.2 I/O 接口

表 3-9 I/O 功能定义

名称	引脚	信号	描述	功能	
X1 I/O	1	COM+	DIN 公共正端输入	输入电压：5~24VDC 输入电流：8mA@5VDC, 12mA@24VDC 有效输入信号：大于 3VDC 无效输入信号：小于 1VDC	
	2	DIN1	DIN1 负端输入		
	3	DIN2	DIN2 负端输入		
	4	DIN3	DIN3 负端输入		
	5	485/A	CANH	差分信号输入	总线信号，内部使用不隔离电路，需驱动器间共 GND
	6	485/B	CANL		

注：

- (1) DIN1 和 DIN2 输入使用高速光耦，最高输入频率为 200KHz。
- (2) DIN3 输入使用低速光耦，最高输入频率为 1KHz。

3.3.3 拨码开关设置

表 3-10 驱动器节点 ID 号由拨码开关设置，

ID 号	SW1	SW2	SW3	SW4	SW5	SW6
EEPROM	OFF	OFF	OFF	OFF	OFF	OFF
1	ON	OFF	OFF	OFF	OFF	OFF
2	OFF	ON	OFF	OFF	OFF	OFF
3	ON	ON	OFF	OFF	OFF	OFF
—	—	—	—	—	—	—
63	ON	ON	ON	ON	ON	ON

注:

- (1) 当 SW1~SW7 都为“OFF”时, ID 号为 EEPROM 值, 这值是指“设备站号(加偏移值)”(0x2FE400 对象)填入的值。
- (2) SW7 为 CM422B, RS485 通讯协议选择, 为“OFF” 时是 ModBus RTU 协议, 为“ON” 时是 Kinco RS232 自由通讯协议, 用于连接上位机软件 KincoStep, 调试驱动器。
- (3) SW8 为总线终端电阻开关, 为“ON” 时将 120 欧电阻并联在总线上, 为“OFF” 时为断开。

3.4 Console 线

Console 线为驱动器与电脑的转接线, 一边连接电脑 RS232 (DB9 针头接口), 另一边连接驱动器的 RS232 (RJ45 接口), 图片如下。



图 3-11 RS232 转接线 (1)

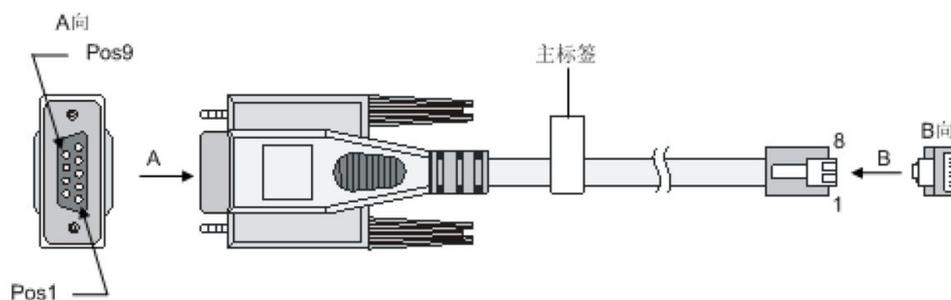


图 3-11 RS232 转接线 (2)

3.4.1 转接线的针脚连线顺序如下:

DB9 母头	-----	RJ45 (网线口)
RXD (2)	-----	TXD (3)
TXD (3)	-----	RXD (6)
DTR (4)	-----	DSR (7)
GND (5)	-----	GND (4)&(5)
DSR (6)	-----	DTR (2)
RTS (7)	-----	CTS (8)
CTS (8)	-----	RTS (1)

第四章 驱动器外部接线图

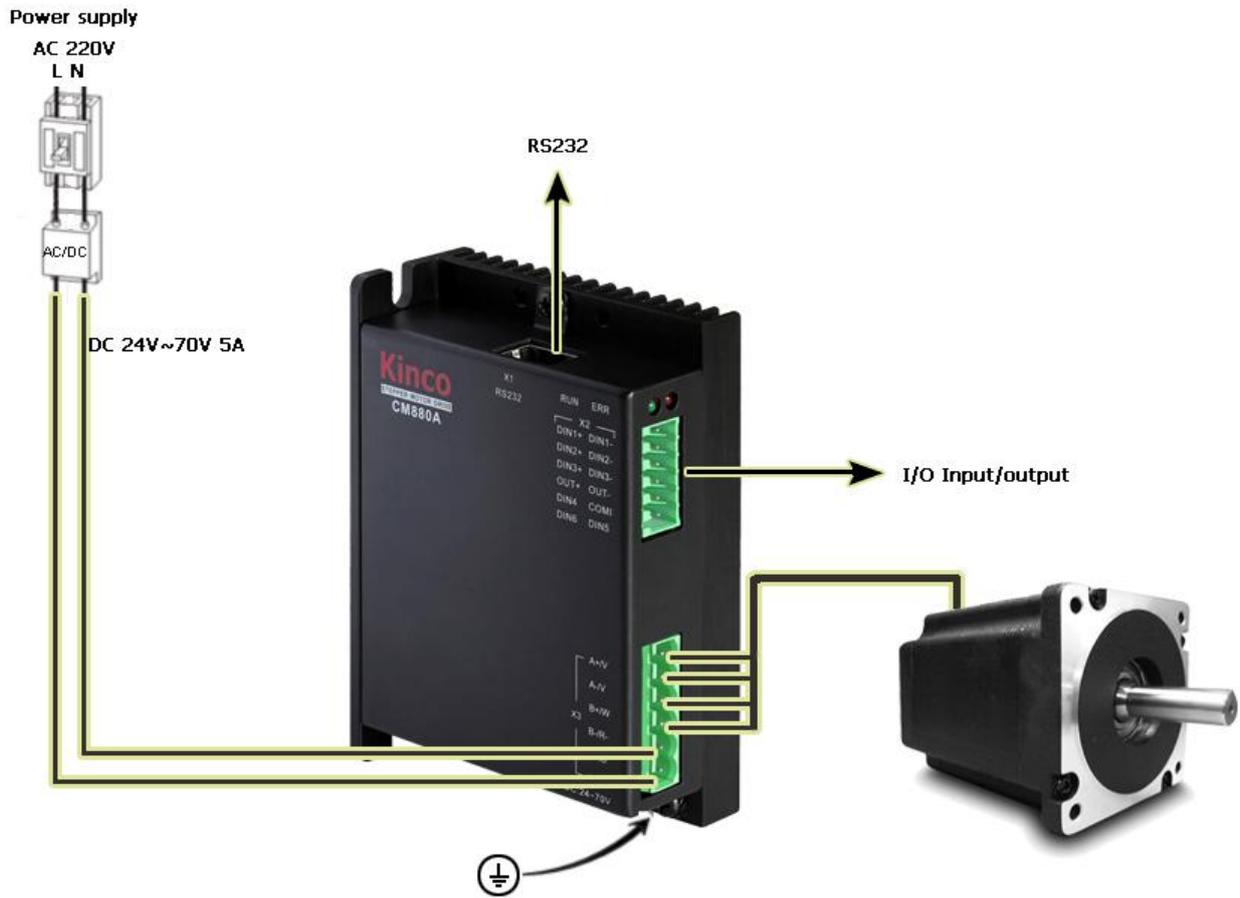


图 4-1 CM880A 驱动器外部接线图

第五章 步进调试软件使用说明

5.1 软件安装

本软件无需安装，用户可访问 www.kinco.cn 网站“服务与支持”-“下载中心”-“软件下载”，下载“Kinco 步进上位机调试软件（适用于 FM-CM 系列步进）”，解压缩即可使用。

5.2 快速入门

5.2.1 调试软件连接

使用前请参照第三章内容，正确连接步进驱动器和电机。首次使用时不要接电机线，需配置好驱动器后再接，以防配置较大相电流时，损坏电机。

利用 RS232 口进行编程的最低系统要求：FM 或 CM 系列步进驱动器，如 CM880A；提供给驱动器电压 24VDC；Console 线(详见附录三)

PC 9 针 D 型	RS232 (RJ45 接口)
RxD (2)	TXD (3)
TxD (3)	RXD (6)
GND (5)	GND (4)

如果用户手上拥有德国 PEAK-System 的 PCAN-USB 适配器，用户可以使用它连上 KincoStep 进行调试编程。



Pcan 9 针 D 型	CAN 接口
CANH (7)	CANH
CANL (2)	CANL
GND (3)	GND

5.2.2 调试软件联机

1. 打开调试软件文件夹，双击应该带有 KincoStep  图标，将打开软件使用平台如下：



2. 新建工程



3. 弹出“通讯方式”对话框，如果是串口连接选择“RS232”，点击下一步。



如果用 PECAN 等 CAN 工具连接选择“CAN”，点击下一步。



4. 进入通讯属性界面，设置 COM 口，波特率，驱动器 ID 号（下图是默认参数, 或是 127 调试 ID 号），

点击通讯状态按钮 。



如果用 PECAN 连接，则设置好 CAN 波特率，驱动器 ID 号（下图是默认参数），点击通讯状态按钮。



5. 观察右下角属性 Comm Status: Open COM1 38400, 通讯状态指示灯也变绿，证明已经联机成功。

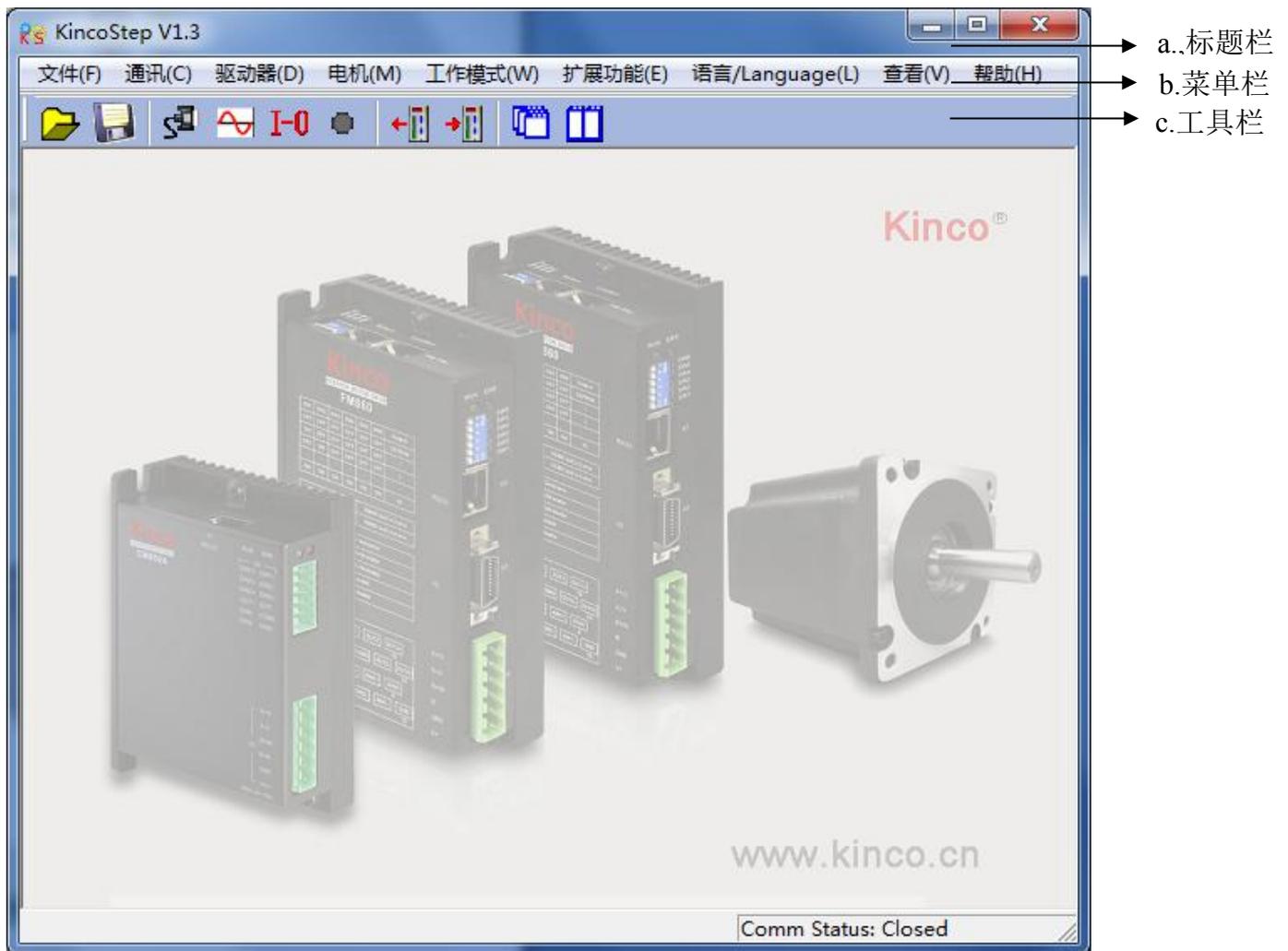


CAN 连接观察右下角属性 Comm Status: Open 500K Bit/S, 通讯状态指示灯也变绿，证明已经联机成功。



5.3 菜单介绍

打开 KincoStep 调试软件使用窗口如下：



如上图所示，最上面是菜单栏，鼠标点击各菜单有相应功能选择子菜单，下面工具栏图标是常用功能的快捷方式，各菜单功能描述。

表 5-1 菜单栏介绍

名称	功能说明
文件	可新建，保存，打开工程操作
通讯	用于设置驱动器与 PC 连接通讯方式，通讯参数等
驱动器	对驱动器进行控制，详细见 4.4 驱动器控制
电机	电机参数进行设置
工作模式	工作模式参数配置设置
扩展功能	读写驱动器参数配置，详细见附录例子
语言	设置语言切换，

5.4 驱动器控制

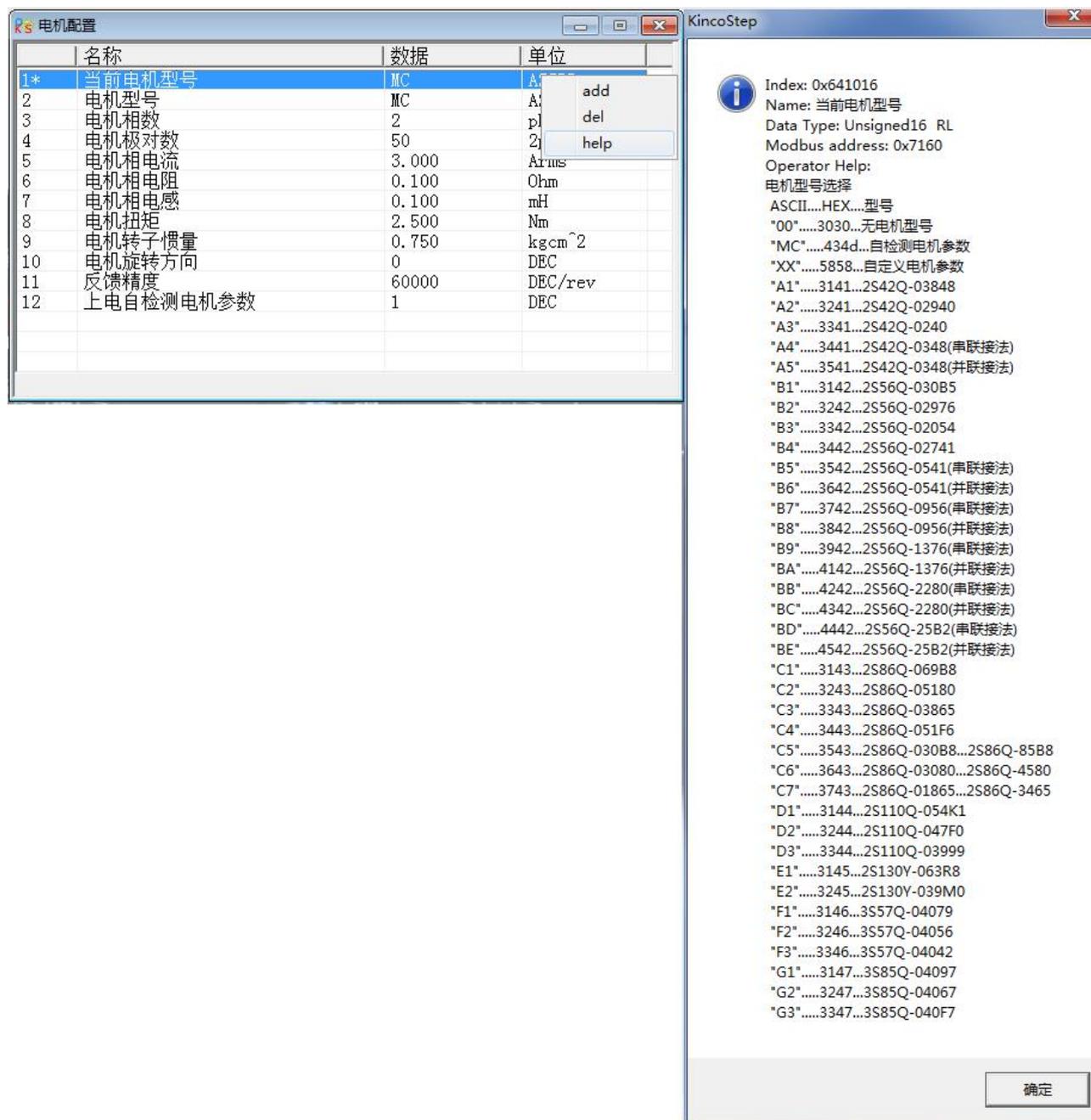
5.4.1 驱动器配置

	名称	数据	单位
1	用户密码	0	DEC
2	自动半流开关	1	DEC
3	自动半流百分比	50.000	%
4	自动半流时间	1.500	S
5	电机相电流限制	4.300	Arms
6	IO口5V输出开关	0	DEC
7	RS232波特率	38400.000	Bandrate
8	RS485波特率	19200.000	Bandrate
9	CAN波特率	50	DEC
10	设备站号	1	DEC
11	设备站号(加偏移值)	127	DEC

在本菜单可以进行驱动器的基本配置。例如用户密码，RS232 通讯等常用参数。可以右键鼠标，看帮助提示。



5.4.2 电机配置



用户可选择以下 3 种方式中的任意一种方式，配置电机参数。

1、自检测电机参数（驱动器出厂默认设置，电机型号为 MC）

驱动器出厂默认设置：

对于 CM880A, CM880B 和 CM880C：电机型号为 MC，电机相数为 2，电机相电流默认值 3A（有效值），如果相电流不是 3A，请设置为电机对应的相电流。

对于 CM422B 和 CM422C：电机型号为 MC，电机相数为 2，电机相电流默认值 0.7Arms（有效值），如果相电流不是 0.7A，请设置为电机对应的相电流。

驱动器默认自检测 2 相电机，若用户使用 3 相电机，驱动器的指示灯会报错：RUN 灯快闪，ERR 灯常亮，

(若用户使用调试软件, 实时错误菜单会显示驱动器内部错误和寻找电机错误), 用户需要将电机相数更改为 3, 存储电机参数, 驱动器重启后可正常使用。

2、选择电机型号

若用户不采用自检测电机参数, 可以直接选择对应电机型号, 驱动器会自动调用电机相关参数。

3) 自定义电机参数 (电机型号为 XX)

若采用第 3 方电机, 可以把电机型号设置为 XX, 用户自行输入电机相关参数。

5.4.3 电流设置

对于 CM880A, CM880B 和 CM880C: 出厂驱动器, 电机相电流默认值 3A (有效值) / 4.2A (峰值)。正常情况下, 设置范围为 0A (有效值) / 0A (峰值) ~ 5.7 (有效值) / 8A (峰值)。

对于 CM422B 和 CM422C: 出厂驱动器, 电机相电流默认值 0.7A (有效值) / 1A (峰值)。正常情况下, 设置范围为 0A (有效值) / 0A (峰值) ~ 1.6A (有效值) / 2.2A (峰值)。

用户可以根据应用需求, 设置电机相电流。用户更改电流设置后, 需要存储电机参数, 驱动器重启后可正常使用。若用户需要驱动器输出更大的电流, 请联系厂家。

5.4.4 细分设置

	名称	数据	单位
1*	脉冲模式	1	DEC
2	每转脉冲个数(细分)	1600	step/rev
3*	齿轮前脉冲频率	0	kHz
4*	齿轮后脉冲频率	0	kHz
5	齿轮前脉冲个数	0	DEC
6	齿轮后脉冲个数	0	DEC
7	脉冲滤波系数	3	DEC
8	脉冲频率限制	600	kHz

在脉冲模式 (-4 模式) 下, 细分设置: 细分=每转脉冲个数 / (360° / 步矩角)。

注: 两相电机设置每转脉冲个数必须 ≥ 200 ; 三相电机设置每转脉冲个数必须 ≥ 300 。

5.4.5 八线步进电机接线

对于 4, 8 线步进电机, 引线颜色见下图。其中 8 线电机有两种接法, 其性能差异如下: **a.** 并联接法使线圈电感变小, 适合高速运转, 但需驱动器提供更大的电流, 才能达到所需扭矩。**b.** 而串联接法使线圈电感变大, 适合低速运转, 驱动器提供小的电流, 就能达到所需扭矩, 见下图 3 和 4。

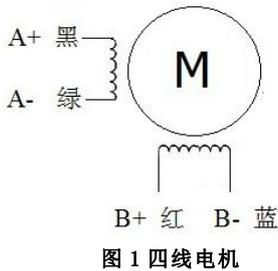


图 1 四线电机

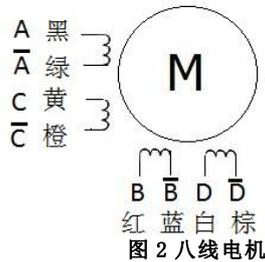


图 2 八线电机

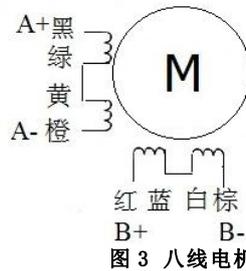


图 3 八线电机串行接法

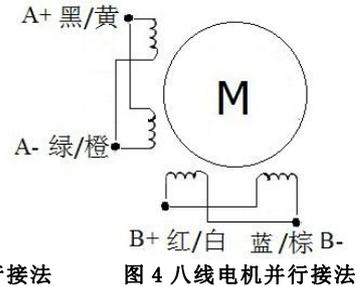


图 4 八线电机并行接法

5.5 基本操作

名称	数据	单位	
1*	有效工作模式	3	DEC
2*	状态字	437	HEX
3*	实际位置	11569.280	step
4*	实际速度 -rpm	100	rpm
5*	实际电流	3.029	Arms
6*	实际总线电压	70	V
7	工作模式	3	DEC
8	目标位置	0.000	step
9	目标速度	100.000	rpm
10	控制字	2f	HEX
11	目标电流	3.004	Arms
12	梯形速度	0.000	rpm
13	梯形加速度	10.000	rps/s
14	梯形减速度	10.000	rps/s

在本菜单可以进行驱动器的基本控制操作。各模式具体操作对象参考第 6 章模式操作。

另外, 在本菜单可以监控驱动器的主要运行数据。

例子 5-1: 利用 Kincostep 调试软件设置一个用户密码

第一步: 在上图“用户密码”设置密码为 1234 (密码范围 1-65535) 后, 回车确定。

第二步: 点击“驱动器”-“初始化/保存”菜单里面的“存储控制参数”进行保存; 如果修改了电机参数, 需要“存储电机参数”, 再点击“驱动器重启”。

第三步: 驱动器重启后密码生效, 用户不能设置常用参数, 必须先到“驱动器配置”菜单“用户密码”处设置正确密码, 才能设置参数。

第四步: 如果要取消密码, 必须先输入正确密码后再把密码设为 0, 保存重启即可。所以请务必记住您设置的密码。

	名称	数据	单位
1	用户密码	0	DEC
2	自动半流开关	1	DEC
3	自动半流百分比	50.000	%
4	自动半流时间	1.500	S
5	电机相电流限制	4.300	Arms
6	IO口5V输出开关	0	DEC
7	RS232波特率	38400.000	Bandrate
8	RS485波特率	19200.000	Bandrate
9	CAN波特率	50	DEC
10	设备站号	1	DEC
11	设备站号(加偏移值)	127	DEC

例子 5-2: 利用 KincoStep 调试软件对步进驱动进行速度模式操作 (手动运转)

第一步: 按照例子 5-3 设置, 选择 I/O 控制中的 DIN1 为“驱动器使能”, DIN2 为驱动器错误复位, DIN3 “驱动器工作模式控制”的定义。

第二步: 按照“模式控制”章节“速度模式”常用对象设置好各项基本参数, 如上图所示就是驱动器处于速度模式工作, 速度 100RPM. 如需反方向运转, 直接设置速度为负值即可。

5.6 I/O 操作

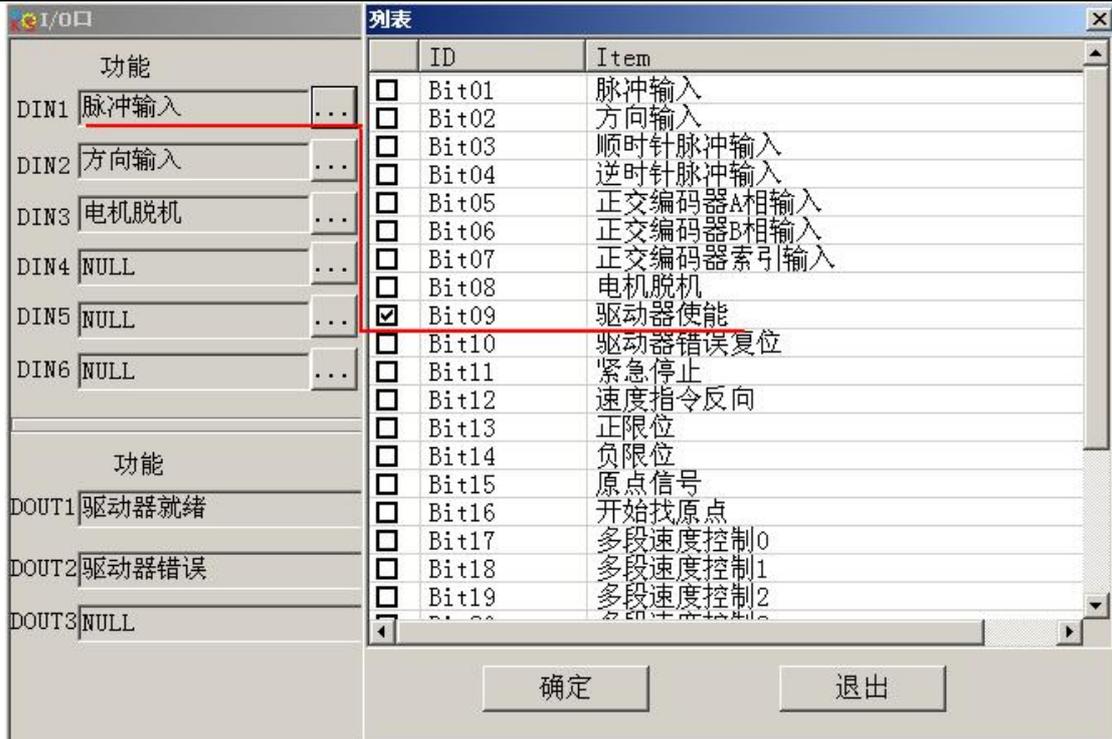


在本菜单可以对驱动器 I/O 进行功能定义，极性设置，具体操作步骤可参考例子说明。还可以监控 I/O 实际输入输出状态，模拟仿真等功能。上图所示是默认 I/O 设置。

例子 5-3：利用 KincoStep 调试软件对步进 I/O 进行设置

要求：选择 DIN1 为驱动器使能，DIN2 为驱动器错误复位，DIN3 为驱动器工作模式控制，DIN4 为正限位，DIN5 为负限位，DIN6 为找原点，其他按默认设置！

第一步：如下图所示，点击 DIN1 右侧红色标示的  按钮，在弹出的输入口功能列表里面把默认的“脉冲输入”取消，选择“驱动器使能”，点击确定退出。



第二步：仿照第一步操作，分别对各 I/O 进行定义，取消，更改等操作。最终设置好的画面如下图。



5.7 数据字典



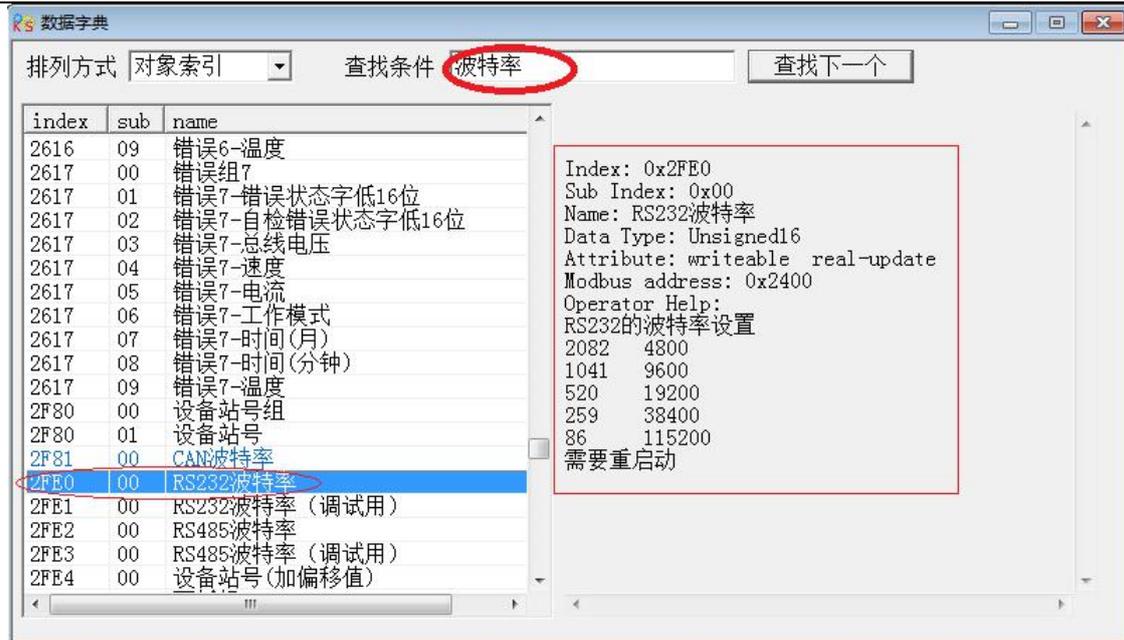
在本菜单可以查询到所有 CM 驱动器内部对象地址及详细含义。如上图所示，左边显示的对象符合 CANopen 定义的 INDEX, SUBINDEX 地址及其命名，右边有详细参数含义以及状态属性，数据长度等信息！

例子 5-4：利用 KincoStep 调试软件添加一个对象进行设置

要求：在任意菜单里面添加一个地址进行设置。本例以在“基本操作”菜单里面添加“RS232 波特率”对象为例说明

第一步：在“基本操作”菜单任意位置，点击右键，点击“add”，弹出“数据字典”对象列表。

第二步：在“查找条件”输入“波特率”模糊搜索关键词，点击“查找下一个”，系统搜索出 2FE0 这个“RS232 波特率”对象，右边显示其详细定义。



第三步：双击对象 RS232 波特率，其立即添加到基本操作菜单里，如下图所示。



第四步：如果要删除列表里的对象。选中对象→点击右键并选择“del”即删除。

如果要了解对象详细定义，选中对象点击右键后选择“help”即可显示详细定义。

5.8 驱动器属性

在本菜单可以看到驱动器的软件版本，序列号等信息。



	名称	数据	单位
1*	设备型号	40192	HEX
2*	设备名称	CM880A RS232	String
3*	硬件版本	V1.0	String
4*	软件版本	CM880A20150108	String
5*	制造商	Kinco Electric (Shenzhen) Ltd.	String
6*	硬件序列号	CM0880410150060004	String
7*	设备厂商站号	300	HEX
8*	设备时间	21619	S
9*	驱动器温度	24	degree

5.9 实时错误

在本菜单可以监控到当前报警信息！如下图所示，左边 16 进制数字代码是报警代码，小方框里勾选选择是否屏蔽报警（调试用途，不能保存），指示灯红色表示有报警，绿色则无，文字表示报警信息，具体报警原因请参考[第八章报警排除](#)。

注意：请慎重选择屏蔽报警，而且不是所有报警都可以屏蔽的！



5.10 历史错误

驱动器提供 8 组历史报警信息，用户可以查询到报警发生时的报警代码，电压，电流，温度，速度，工作模式，驱动器累计工作时间等信息，更好的方便用户设备维护。

“错误状态字低 16 位/高 16 位”是驱动器运行过程出现的错误，而“自检错误状态字低 16 位/高 16 位”是驱动器上电自检测硬件出现的错误。

“错误组当前位置”值指示最新发生的错误保存在那一组，数值 N 指示当前报警在“错误 N”，上一次记录是 N 减 1，下一次记录是 N 加 1，8 组错误记录循环覆盖。

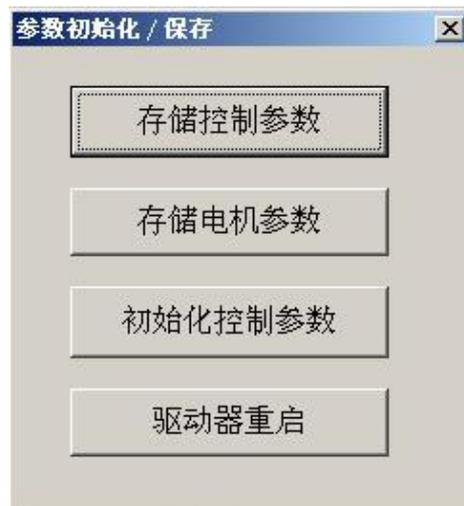
例如：错误组当前位置=4，表明当前的报警为“错误 4”。下一次的报警位置为“错误 5”。



	名称	数据	单位
1*	错误状态字高16位	0	HEX
2*	错误状态字低16位	0	HEX
3*	自检错误状态字高16位	0	HEX
4*	自检错误状态字低16位	0	HEX
5*	错误计数	14	DEC
6*	错误组当前位置	5	DEC
7*	错误0-错误状态字低16位	101	HEX
8*	错误0-自检错误状态字低16位	8000	HEX
9*	错误0-总线电压	28	V
10*	错误0-速度	0.000	rpm
11*	错误0-电流	0.000	Arms
12*	错误0-工作模式	0	DEC
13*	错误0-时间(月)	0	Mon
14*	错误0-时间(分钟)	342	Min
15*	错误0-温度	27	degree

5.11 初始化/保存

本菜单用来对驱动器参数保存，初始化，驱动器重启等。需要注意的是电机参数和其他参数是分别保存的！如果按钮处于灰色状态表明存储操作正在进行中。



第六章 输入输出操作

驱动器拥有 6 路数字输入口，1 路数字输出口，或者 3 路数字输入口，1 路数字输出口。数字输入输出口可以根据自己应用需求自由配置各种功能。对于没有物理输入输出的 IO 口，它们的默认值为无效低电平，可以用于调试驱动器。

6.1 数字信号输入

6.1.1 数字信号输入极性控制

利用 kicnostep 调试软件改变 IO 的极性，即切换 S1 为常开或常闭。S1 打开时为常开，闭合时为常闭。

例子 6-1：数字信号输入 DIN1 极性设置

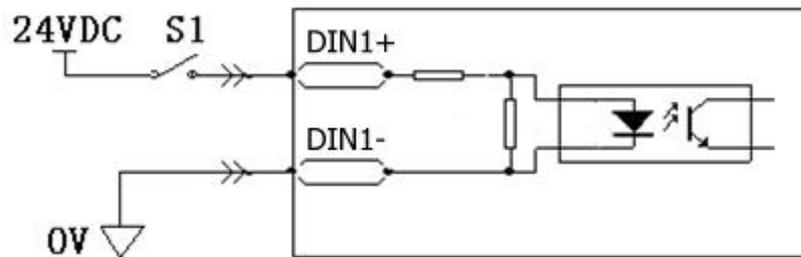


图 6-1 数字信号输入 DIN1 极性设置示意图

调试软件连接驱动器后，然后打开 IO 口页面，默认情况下极性指示灯显示为绿色，表示此时输入口为常开点，即 DIN1 状态无效。点击下图中 DIN4, DIN5 将极性指示灯改变成红色，则此时表示 DIN4, DIN5 变为常闭点，状态为有效。



图 6-2 kincostep 调试软件中数字 I/O 示意图

6.1.2 数字信号输入功能

表 6-1 数字信号输入功能及默认值

参数名称	功能	默认值
DIN1_Function 输入 1 功能	脉冲输入 方向输入 顺时针脉冲输入 逆时针脉冲输入	0x1 (脉冲)
DIN2_Function 输入 2 功能	正交编码器 A 相输入 正交编码器 B 相输入 正交编码器索引输入 电机脱机	0x2 (方向)
DIN3_Function 输入 3 功能	驱动器使能 驱动器错误复位 紧急停止 速度指令反向	0x80 (脱机)
DIN4_Function 输入 4 功能	正限位 负限位	0 (NULL)

	原点信号 开始找原点	
DIN5_Function 输入 5 功能	内部速度控制 0 内部速度控制 1 内部速度控制 2 内部速度控制 3	0 (NULL)
DIN6_Function 输入 6 功能	内部位置控制 0 内部位置控制 1 内部位置控制 2 内部位置控制 3 指令激活 驱动器工作模式控制 预使能	0 (NULL)

表 6-2 数字信号输入定义功能的含义

功能	含义	
脉冲输入	脉冲加方向模式下，脉冲信号输入。	注：这些功能只能配置在 DIN1 和 DIN2 高速 I/O 口上，其它 I/O 不支持。
方向输入	脉冲加方向模式下，方向信号输入。	
顺时针脉冲输入	双脉冲模式下，顺时针脉冲输入。	
逆时针脉冲输入	双脉冲模式下，逆时针脉冲输入。	
正交编码器 A 相输入	增量式编码器模式，A 相输入。	
正交编码器 B 相输入	增量式编码器模式，B 相输入。	
正交编码器索引输入	增量式编码器模式，Z 相输入。	
电机脱机	电机脱机信号，有效状态下电机为松轴状态。	
驱动器使能	驱动器使能信号，有效状态下的控制字可由 202033 对象“输入控制字”进行设置。	
驱动器错误复位	清除错误报警，信号上升沿有效，	
紧急停止	信号有效时，按急停模式停止后，电机松轴。	
速度指令反向	在速度模式（模式）将目标速度反向。	
正限位	电机正向运行极限限位（默认为常闭点，可以通过修改极性来适应常开开关）	
负限位	电机反向运行极限限位（默认为常闭点，可以通过修改极性来适应常开开关）	
原点信号	原点开关信号	
开始找原点	开始执行找原点指令，信号上升沿有效。	
多段速度控制 0	用于控制多段速度切换，信号输入 3~0 这 4 个位组合成 1 个 16 进制数。例如：控制 2 为 1，其它为 0，则当前 16 进制数为 0x4，对应的是第 4 段速度。	
多段速度控制 1		
多段速度控制 2		

多段速度控制 3	
多段位置控制 0	
多段位置控制 1	用于控制多段位置切换。信号输入 3~0 这 4 个位组合成 1 个 16 进制数。例如：控制 2 和 3 为 1，其它为 0，则当前 16 进制数为 0xC，对应的是第 12 段位置。
多段位置控制 2	
多段位置控制 3	
指令激活	激活目标位置或者目标位置段，信号上升沿有效。
驱动器工作模式控制	用于实现两种工作模式之间的切换。 有效信号、无效信号对应的工作模式可以自定义，需要通过 202031 对像“输入工作模式选择 0”和 202032 对像“输入工作模式选择 1”进行设置。
预使能	外部信号使能驱动器，如果输入信号变为无效状态，运行中的驱动器会报警。

例子 6-1：驱动器使能设置

本例选用 kicnstep 调试软件来设置 I/O 功能，软件连上驱动器后，打开 I/O 页面，点击 DIN1 右侧红色标示的  按钮，在列表中选“驱动器使能”功能，然后确定。不需要的功能则去掉前面的勾。



图 6-3 调试软件设置 I/O 功能示意图

例子 6-2 脉冲输入模式设置

驱动器脉冲输入功能只允许接 DIN1 和 DIN2 两个高速 I/O 口，其它 I/O 不支持。脉冲输入模式需要

两个 IO 口，当 DIN1 选择“脉冲输入”，DIN2 会自动配置为“方向输入”。如果配置 DIN1 信号控制（如：“脉冲输入”），相应的 DIN2 要配置，不能为“NULL”，否则-4 工作模式不会激活。

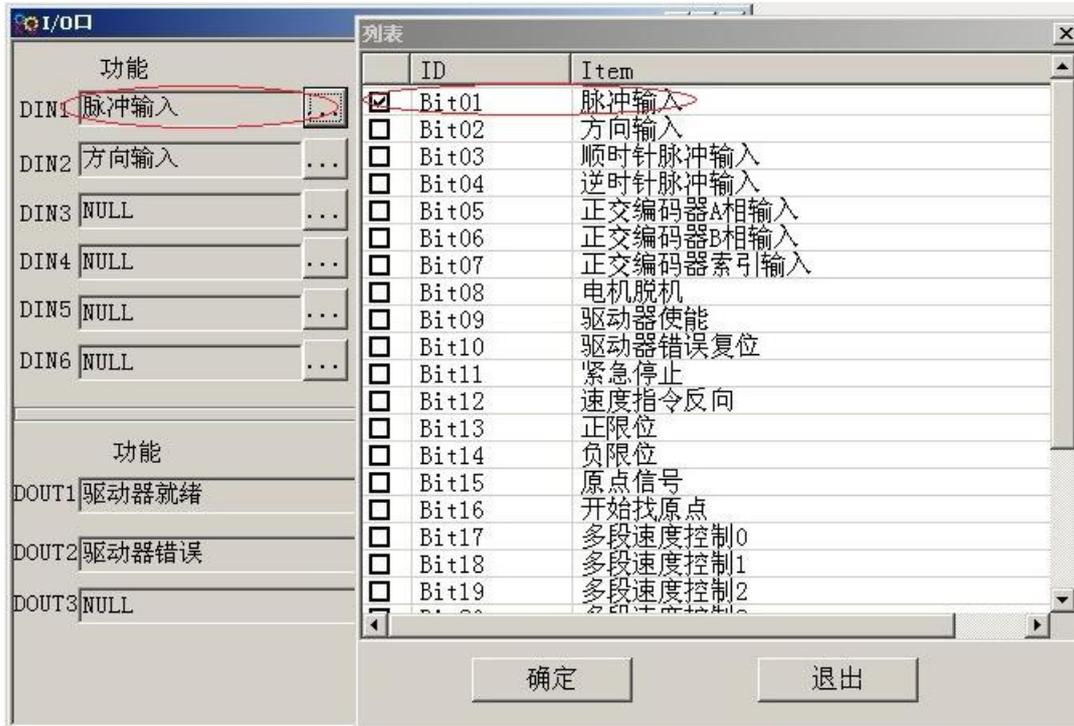


图 6-4 调试软件设置 I/O 功能示意图

6.1.3 数字输入口接线

1. NPN 方式接线图（支持低电平输出有效的控制器）

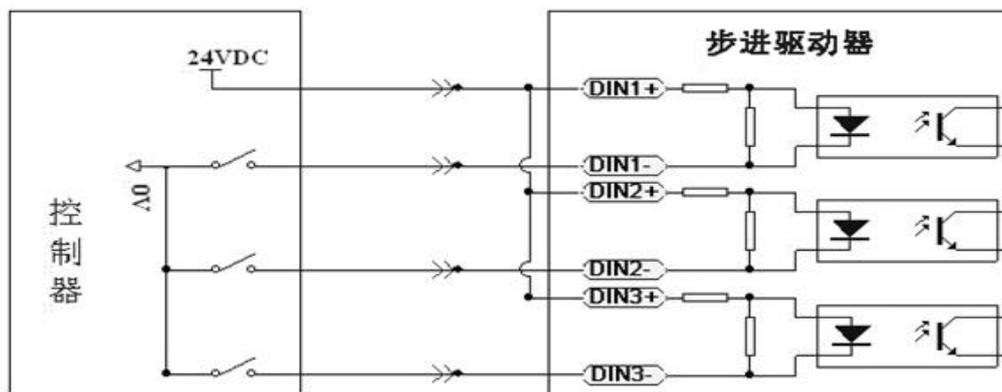


图 6-5 NPN 方式接线图（共阳接线方式）

2. PNP 方式接线图（支持高电平输出有效的控制器）

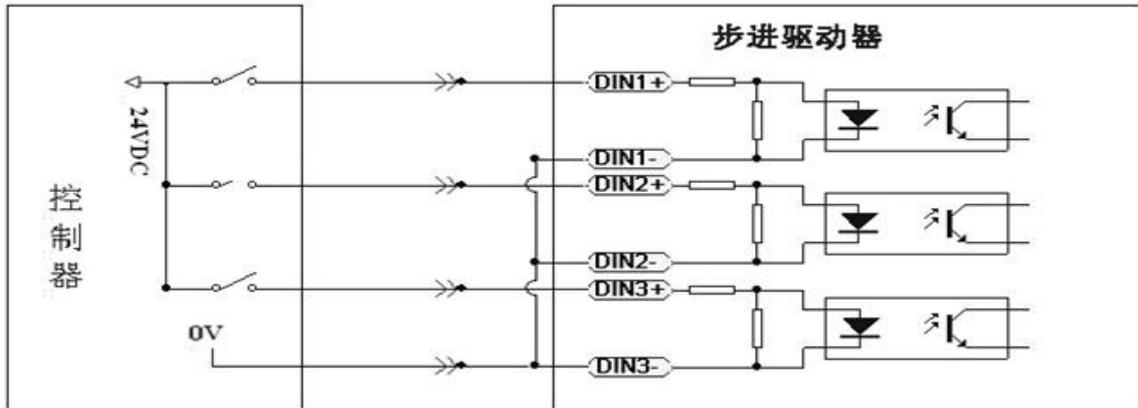


图 6-6 PNP 方式接线图（共阴接线方式）

6.2 数字信号输出

6.2.1 数字信号极性控制

数字输出口默认情况下全部为常开点。若用 KincoStep 调试软件改变输出口极性，请参考例子 5-1。

6.2.2 数字信号输出地址以及功能

表 6-3 数字信号输出地址及默认功能

参数名称	含义	默认值
DOUT1_Function 输出 1 功能	驱动器就绪 驱动器错误 电机位置到 电机零速 保留 电机速度到 保留 保留 电机锁轴 电机限位中 原点找到 索引信号倍频 吸收母线电压	0x1（驱动器就绪）

表 6-4 数字输出信号定义功能的含义

功能	含义
驱动器就绪	驱动器处于可操作状态。
驱动器错误	驱动器处于错误报警状态。
电机位置到	在位置模式下，目标位置数据在“位置到时间窗口”内没有变化，同时位置误差在“位置到窗口”内。
电机零速	电机使能后，电机速度为零状态。
保留	保留（功能待定）

电机速度到	在速度模式控制下，电机速度达到目标速度。
保留	保留（功能待定）
保留	保留（功能待定）
电机锁轴	驱动器使能电机。
电机限位中	电机处于限位状态
原点找到	寻找原点结束。
索引信号倍频	以索引信号为原点，输出设定周期脉冲
吸收母线电压	吸收母线电压，只限于驱动 3 相电机时使用，需把电阻接 Vdc+和 R-

例子 6-3：驱动器就绪设置

本例选用 kicnostep 调试软件来设置 I/O 功能，软件连上步进驱动器后，打开 I/O 页面，点击 DOUT1 右侧红色标示的  按钮，在列表中选“驱动器就绪”功能，然后确定。不需要的功能则去掉前面的勾。设置方法如下图。

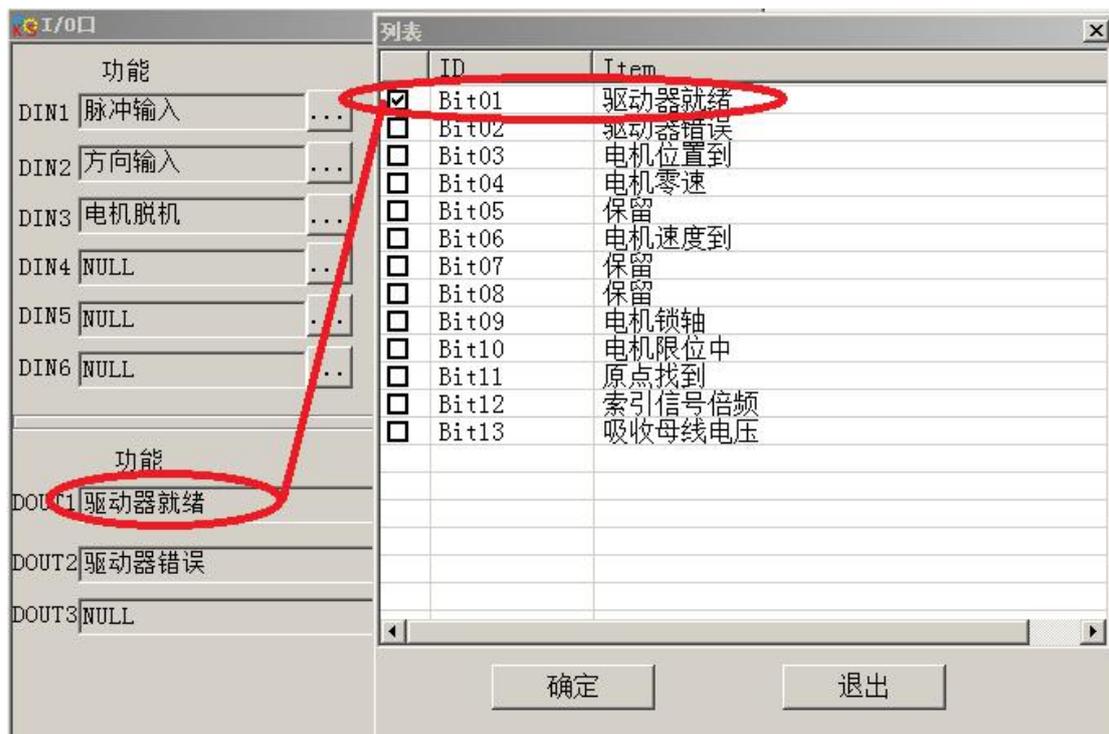


图 6-7 驱动器就绪设置

6.2.3 数字输出口接线

1. 数字输出内部电路图

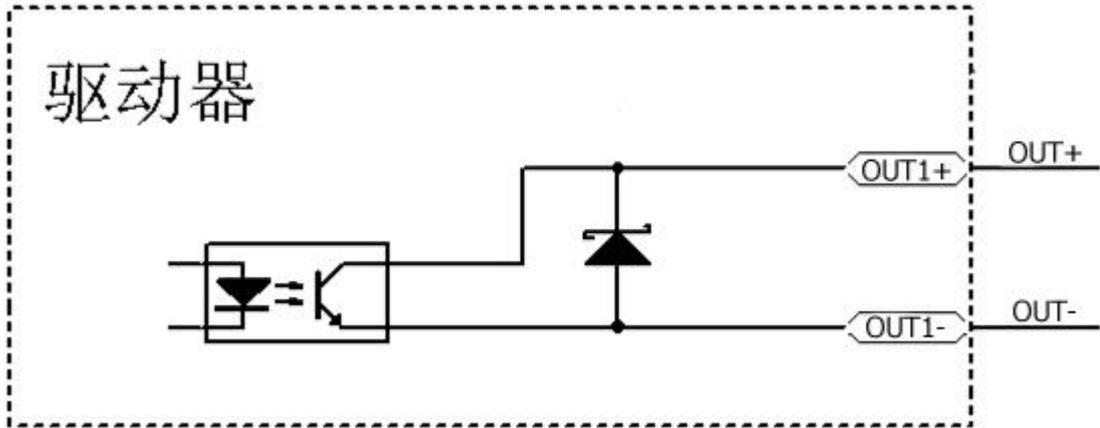


图 6-8 数字输出内部电路图

2. NPN 方式接线图（OUT1-OUT3 都支持此种接法，支持低电平输入有效的控制器）

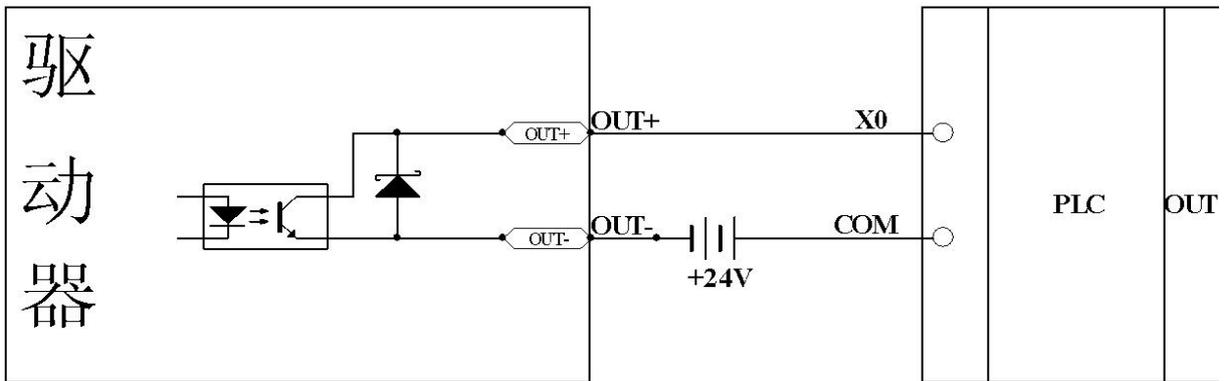


图 6-9 NPN 方式接线图（共阴接线方式）

3. PNP 方式接线图（仅 OUT3 支持此种接法，支持高电平输入有效的控制器）

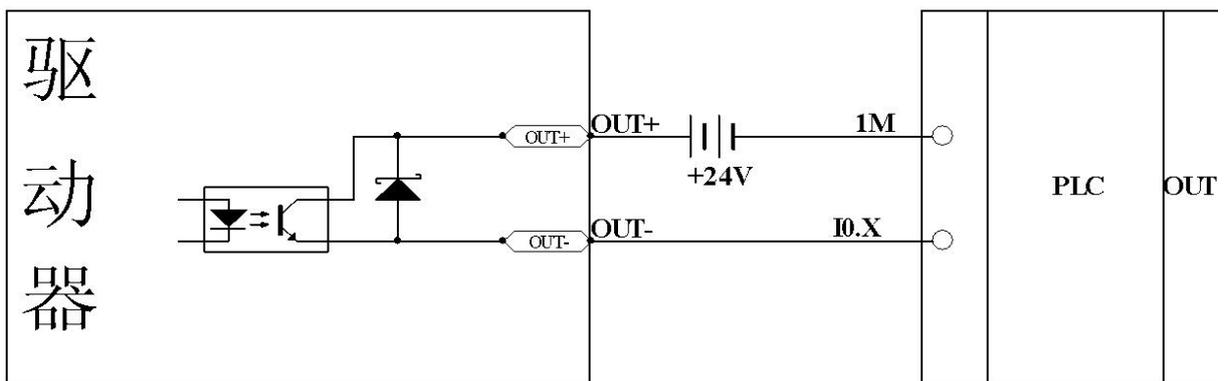


图 6-10 PNP 方式接线图（共阳接线方式）

4. 数字输出口接续继电器，注意必须按图 6-11 操作。

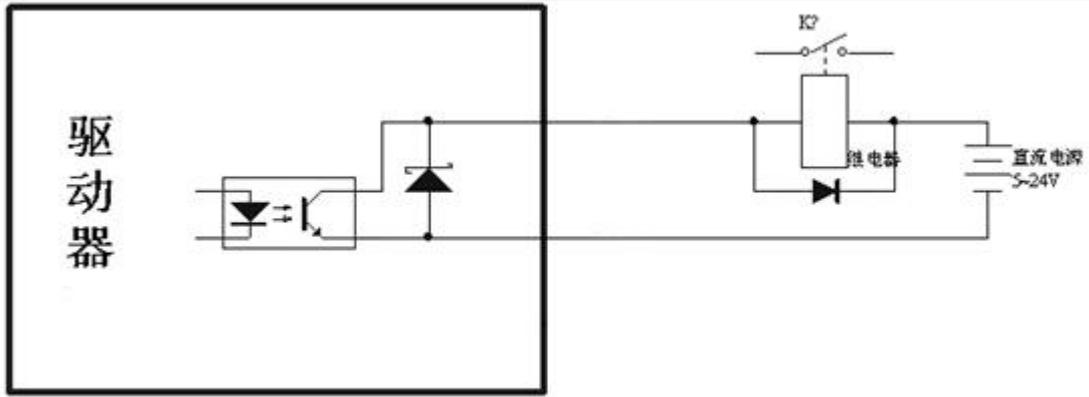


图 6-11 数字输出口接继电器图（注意反并联二极管）

第七章 模式操作

7.1 位置模式（“1”模式）

现在对该模式举例说明，如下图的一个坐标系，红色箭头标识为当前位置=440，如果定义为绝对位置运动，当将目标位置设定为 700 后电机将运动到坐标=700 的位置；如果定义为相对位置运动，当将目标位置设定为 700 后电机将运动到坐标=1140 的位置。

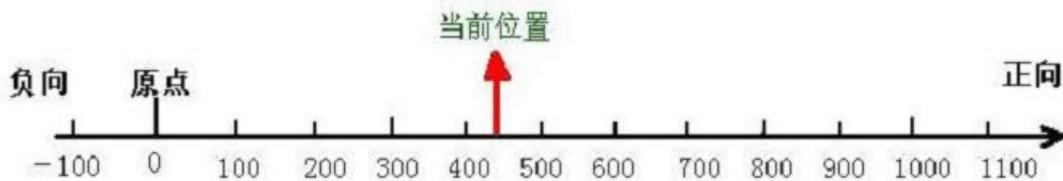


图 7-1 绝对/相对位置

表 7-1 位置模式 1 下的对象定义

名称	CANopen 地址	Modbus 地址	值	含 义
工作模式	60600008	0x3500	1	设定工作模式为绝对/相对位置模式
梯形速度	60810020	0x4A00	用户设定	梯形曲线速度
梯形加速度	60830020	0x4B00	用户设定	加速度
梯形减速度	60840020	0x4C00	用户设定	减速度
目标位置	607A0020	0x4000	用户设定	位置模式下的定位目标位置
控制字	60400010	0x3100	先 F 后 1F	绝对位置运动时，上次目标位置到后，电机开始运动
			先 2F 后 3F	绝对位置运动时，电机立即开始运动
			先 4F 后 5F	相对位置运动时，上次目标位置到后，电机开始运动
			先 6F 后 7F	相对位置运动时，电机立即开始运动
			103F	根据目标位置的变化立即开始绝对运动

更详细的内容请参考[附录七“常用对象列表”](#)中“模式及控制”和“目标对象”部分

通过通讯进行位置模式控制，可参考附录各通讯案例

也可通过外部 I/O 设定进行位置模式控制

举例：

1) 在绝对位置模式下，从位置 440，运行到位置 700。操作过程：A) 设好参数，“工作模式”设为 1，“梯形速度”设为 100rpm，“梯形加速度”和“梯形减速度”都设为 10rps/s，“目标位置”设为 700，“控制字”由 6 变为 2F 使电机有电流锁轴（如果已经锁轴跳过），再写入 3F，电机立即向目标位置运动，停止在 700

位置。如果还需运行到位置 1140，需设“目标位置”为 1140，“控制字”先写入 2F，再写入 3F，电机立即向目标位置运动，停止在 1150 位置。

2) 在相对位置模式下，从位置 440，运行到位置 700。操作过程：A) 设好参数，“工作模式”设为 1，“梯形速度”设为 100rpm，“梯形加速度”和“梯形减速度”都设为 10rps/s，“目标位置”设为 260，“控制字”由 6 变为 6F 使电机有电流锁轴（如果已经锁轴跳过），再写入 7F，电机立即向目标位置运动，停止在 700 位置。如果还需运行到位置 1140，需设“目标位置”为 440，“控制字”先写入 6F，再写入 7F，电机立即向目标位置运动，停止在 1140 位置。

7.2 速度模式（“3”模式）

“3”这种模式可以实现对电机的速度控制，运行曲线包括加速、匀速和减速三段，如下图，加速的时间可以通过初始速度和匀速段速度以及加速度计算出来：

$$V_t = V_o + at \quad V_t - \text{匀速部分速度}$$

$$V_o - \text{起始速度}$$

a - 加速度或减速度

t - 加速时间

$$S = V_o t + (1/2) at^2 \quad S - \text{加速段位移}$$

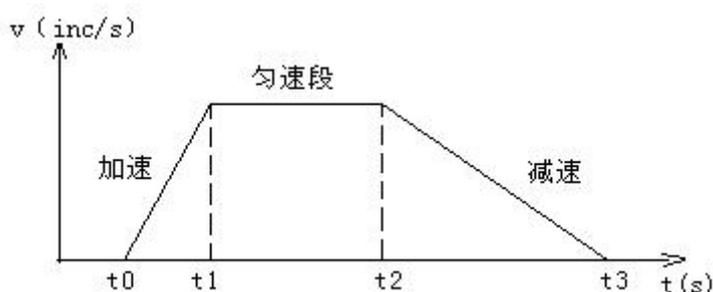


图 7-2 模式 3 下的速度和时间曲线

表 7-2 模式 3 下的对象定义

名称	CANopen 地址	Modbus 地址	值	含义
工作模式	60600008	0x3500	3	设定速度模式
目标速度	60FF0020	0x6F00	用户设定	目标速度
梯形加速度	60830020	0x4B00	用户设定	加速度
梯形减速度	60840020	0x4C00	用户设定	减速度
控制字	60400010	0x3100	先 6 后 F	锁紧电机轴，上面参数设置正确，将按上面参数要求开始运动。当赋予目标速度新值后，电机会以新的速度立即运动。

更详细的内容请参考[附录七“常用对象列表”](#)中“模式及控制”和“目标对象”部分

通过通讯进行速度模式控制，可参考附录各通讯案例

也可通过外部 I/O 设定进行速度模式控制

7.3 脉冲控制模式（“-4”模式）

该模式下，电机的运动直接受控于来自于驱动器 X3 口的外部脉冲，在该模式下要定义的对象有：

表 7-3 模式-4 下的对象定义

名称	CANopen 地址	Modbus 地址	值	含 义
脉冲模式	25080308	0x1930	1	脉冲模式，由 I0 口 DIN1 和 DIN2 配置
工作模式	60600008	0x3500	-4	设定工作模式为脉冲控制模式
每转脉冲个数	64101810	0x7180	用户设定	电机每转脉冲个数
控制字	60400010	0x3100	F	锁紧电机轴，电机按外部脉冲控制运动

注：更改脉冲模式需要重新定义 I0 口 DIN1 和 DIN2，具体操作见例子 6-3 脉冲输入模式设置。

更详细的内容请参考附录“常用对象列表”中“模式及控制”和“目标对象”部分

7.4 原点控制模式（“6”模式）

执行绝对位置定位时，必须要定义原点。例如：在下图所示的 X-Y 平面上，定位 $(X, Y) = (100 \text{ mm}, 200 \text{ mm})$ 之前，必须建立机器的原点 $(0, 0)$ 。

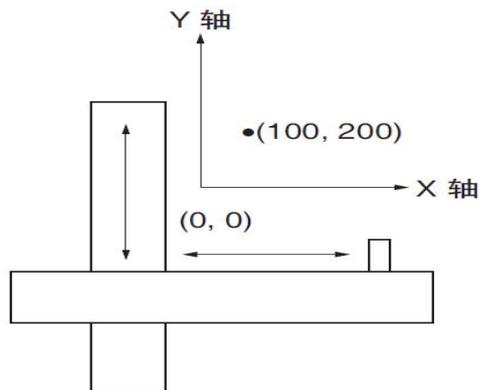


图 7-3 原点控制模式

表 7-4 模式 6 下的对象定义

名称	CANopen 地址	Modbus 地址	值	含 义
工作模式	60600008	0x3500	6	设定工作模式为原点模式
原点模式	60980008	0x4D00	用户设定	寻找原点模式
原点偏移	607C0020	0x4100	用户设定	寻找原点后的位置偏移
原点转折信号速度	60990120	0x5010	用户设定	寻找转折信号时的速度
原点信号速度	60990220	0x5020	用户设定	寻找原点时的速度
原点加速度	609A0020	0x5200	用户设定	原点加速度
控制字	60400010	0x3100	先 F 后 1F	锁紧电机轴，电机开始运动

找原点方式参考附录四找原点方式。

通过通讯进行原点模式控制案例参考附录各通讯章节。

通过外部 I/O 设定进行原点模式控制。

7.5 多段位置控制模式（“1”模式）

多段位置模式是通过外部信号激活内部设定好的目标位置进行电机控制。激活有三个前提条件：

1. 多段位置控制只可以在“1”模式下进行，其他模式无效。
2. 模拟-速度控制(25020708)对象为 0，此时模拟-速度通道无效。
3. 至少有一个外部输入信号定义为“多段位置控制 0/1/2/3”。

多段位置输入信号 0, 1, 2, 3 这四个信号将组合成二进制码用于选择位置段 0~15 中的任意一个作为目标位置。它们对应关系如下表：

表 7-5 二进制对应码

多段位置控制 3	多段位置控制 2	多段位置控制 1	多段位置控制 0	输入位置控制	输入速度控制
0	0	0	0	0	0
0	0	0	1	1	1
0	0	1	0	2	2
0	0	1	1	3	3
0	1	0	0	4	4
0	1	0	1	5	5
0	1	1	0	6	6
0	1	1	1	7	7
1	0	0	0	8	8
1	0	0	1	9	9
1	0	1	0	10	10
1	0	1	1	11	11
1	1	0	0	12	12
1	1	0	1	13	13
1	1	1	0	14	14
1	1	1	1	15	15

注：如果多段位置输入信号 0, 1, 2, 3 有不选择的，则信号默认为 0 值。

7.5.1 多段位置控制

电机需要走七段位置，要求位置第 0 段以 100 RPM 的速度走到 5000 个脉冲的位置，第 1 段以 150 RPM 的速度走到 15000 个脉冲的位置，第 2 段以 175 RPM 的速度走到 28500 个脉冲的位置，第 3 段以 200 RPM 的速度走到-10500 个脉冲的位置，第 4 段以 300 RPM 的速度走到-20680 个脉冲的位置，第 5 段以 325 RPM 的速度走到-30550 个脉冲的位置，第 6 段以 275 RPM 的速度走到 850 个脉冲的位置，第 7 段以 460 RPM 的速度走到 15000 个脉冲的位置。

设置 I/O 如下表：

表 7-6 I/O 设置

参数名称	值
DIN1	驱动器使能
DIN2	驱动器工作模式控制
DIN3	指令激活
DIN4	多段位置控制 0
DIN5	多段位置控制 1
DIN6	多段位置控制 2

表 7-7 位置和速度设置

名称	CANopen 地址	Modbus 地址	值	含 义
输入位置控制0	20200120	0x0C10	5000	多段位置控制0
输入速度控制0	20201120	0x0D10	100	多段速度控制0
输入位置控制1	20200220	0x0C20	15000	多段位置控制1
输入速度控制1	20201220	0x0D20	150	多段速度控制1
输入位置控制2	20200320	0x0C30	28500	多段位置控制2
输入速度控制2	20201320	0x0D30	175	多段速度控制2
输入位置控制3	20200420	0x0C40	-10500	多段位置控制3
输入速度控制3	20201420	0x0D40	200	多段速度控制3
输入位置控制4	20200520	0x0C50	-20680	多段位置控制4
输入速度控制4	20201520	0x0D50	300	多段速度控制4
输入位置控制5	20200620	0x0C60	-30550	多段位置控制5
输入速度控制5	20201620	0x0D60	325	多段速度控制5
输入位置控制6	20200720	0x0C70	850	多段位置控制6
输入速度控制6	20201720	0x0D70	275	多段速度控制6
输入位置控制7	20200820	0x0C80	15000	多段位置控制7
输入速度控制7	20201820	0x0D80	460	多段速度控制7
多段速度/位置切换延时	20203810	0x0F80	10	多段速度/位置切换后, 有效输入的延时时间
输入模式控制0	20203108	0x0F10	1	“驱动器工作模式控制”无效输入时工作模式
输入模式控制1	20203208	0x0F20	3	“驱动器工作模式控制”有效输入时工作模式
输入控制字	20203310	0x0F30	2F	“驱动器使能”或“电机脱机”, 有效输入时“控制字”设置
梯形加速度	60830020	0x4B00	50	加速度
梯形减速度	60840020	0x4C00	50	减速度

存储配置参数	2FF00108	0x2910	1	1: 存储除电机外的所有设定参数 10: 初始化除电机外的所有可保存参数
--------	----------	--------	---	---

注：多段位置单位为“step”，多段速度单位为“rpm”，加速度单位为“rps/s”，通讯时需转换为内部 DEC 单位。

设置好参数后，操作如下：

1. 给驱动器使能信号；
2. 选择要走的位置段。更改DIN4，DIN5，DIN6电平，选择位置段。
3. 指令激活，执行所选位置段程序。

7.6 多段速度控制模式（“3”模式）

控制方式为通过外部输入信号激活内部设定好的目标速度进行电机控制。激活有三个前提条件：

1. 多段位置控制只可以在“1”模式下进行，其他模式无效。
2. 模拟-速度控制(25020708)对象为0，此时模拟-速度通道无效。
3. 至少有一个外部输入信号定义为“多段速度控制 0/1/2/3”。

多段速度输入信号 0, 1, 2, 3 这四个信号将组合成二进制码，用于选择速度段 0~15 中的任意一个作为目标位置。它们对应关系如下表

表 7-8 多段速度二进制码列表

多段速度控制 3	多段速度控制 2	多段速度控制 1	多段速度控制 0	输入速度控制
0	0	0	0	0
0	0	0	1	1
0	0	1	0	2
0	0	1	1	3
0	1	0	0	4
0	1	0	1	5
0	1	1	0	6
0	1	1	1	7
1	0	0	0	8
1	0	0	1	9
1	0	1	0	10
1	0	1	1	11
1	1	0	0	12
1	1	0	1	13
1	1	1	0	14
1	1	1	1	15

注：如果多段速度输入信号 0, 1, 2, 3 有不选择的，则信号默认为 0 值。

第八章 通讯功能

使用通讯控制驱动器需要注意以下几点：

1: 驱动器默认 I/O 口 DIN1 为脉冲输入，DIN2 为方向输入，DIN3 为脱机输入，使用通讯控制时需先行取消此三项输入口功能。（通过调试软件取消）

2: 驱动器内部对象有工程量单位和内部单位区分。使用通讯写入和读取时需要换算为内部单位 DEC。使用通讯写入和读取时均为内部单位，需要注意换算。详情参考附录“常用对象换算关系表”。

3: 使用 CANopen SDO 读写功能，RS232 串口通讯协议和 RS485 Modbus 通讯协议进行控制时，务必处理好多条读写数据指令间隔时序，确保通讯网络上任意时刻只有一个驱动器发送数据请求，并做好通讯错误处理等，以免通讯进入死循环。

4: 驱动器有些内部对象尽管数据长度为多个 BIT，但实际工程使用时无需使用最大值，所以部分对象默认有最大值限制，例如目标速度虽然为 32 位数据长度，理论上可以写一个很大的数据，但是实际上电机不允许这样高速，所以驱动器内部限制其最大值 24576000（换算为 1500RPM）以确保系统安全，超过最大值的数据驱动器不接受，自动默认为此最大值。

8.1 RS232 通讯

8.1.1 RS232 通讯硬件接口

与上位机、HMI、PC 通讯时，单台步进机的接线方式如下，也可以直接使用 Console 配置线。

PC 9 针 D 型	RJ45 (X2)
RxD (2)	TXD (3)
TxD (3)	RXD (6)
GND (5)	GND (4)

8.1.2 RS232 通讯参数

表 8-1 RS232 通讯参数

名称	CANopen 地址	Modbus 地址	值	含义
设备站号	0x100B0008	0x0600	1	设备驱动器当前站号。 注：1) ID 站号默认为 1，可由 0x2FE400 设置 1~127。 2) 更改该参数需要存储，再重新启动。
设备站号(加偏移)	0x2FE40008	0x2800	127	设备驱动器站号 1~127；当更改该参数需要存储，再重新启动。
RS232 波特率	0x2FE00010	0x2400	259	用于设置串口的波特率 设定值 波特率，默认值：259

				2082 4800 1041 9600 520 19200 259 38400 86 115200 需要重新启动
存储控制环参数	0x2FF00108	0x2910	1	1: 存储设定的所有配置参数; 10: 初始化所有的配置参数;

8.1.3 自由传输协议

驱动器的 RS232 通讯遵循严格的主从站协议。上位机能将任何数据传给步进驱动器，设定了地址的驱动器在计算这些数据后，并且返回一个应答。

RS232 使用的传输协议采用固定的十个字节的格式。

byte 0		byte 9	
ID	8 byte data	CHKS	

ID 号从站的地址号

$CHKS = -SUM(\text{byte}0, \dots, \text{byte}8)$, CHKS 为上述计算结果的最后 2 位。

上位机传送:

byte 0		byte 9	
ID	8 byte host data	CHKS	

上位机接收（驱动器返回）:

byte 0		byte 9	
ID	8 byte slave data	CHKS	

注意：每十个字节就有一个自己的 CHKS。

如果上位机送一个网络中不存在的地址数据给驱动器，那么就不会驱动器响应。主机正确地发送数据后，从站会寻找相对应地址号的数据，并检查校验值，如果该值和从站计算的值不符合，则从站不响应。

8.1.4 数据协议

数据协议不同于传输协议，其内容是上面的 RS232 传输协议 10 个字节中间的 8 个字节的内容。所有的参数、数值和功能都是通过 index 和 sub-index 表示。它有两种格式：下载和上传。

A: 下载，就是主站发送命令往从站内的对象写入值，下载到不存在的目标地址将产生错误。

表 8-2 主站传送

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
CMD	INDEX		SUB INDEX	DATA			

CMD 指定数据传输的方向和数据的大小。

0x23 发送四个字节的数据 (bytes 4...7 包含 32 位)

0x2b 发送二个字节的数据 (bytes 4, 5 包含 16 位)

0x2f 发送一个字节的数据 (bytes 4 包含 8 位)

INDEX 发送对象的地址

SUB INDEX 发送对象的子地址

DATA 内 4 个字节的顺序是低位在前，高位在后。

例如：

要向从站内“目标位置”写入 7650 DEC，7650 为 10 进制，1DE2 为 16 进制。由于要写入的对象长度为 4 个字节，目前计算结果 1D E2 只有 2 个字节，那么在高位补零，所以最终结果=00 00 1D E2。

DATA: byte4=E2

byte5=1D

byte6=00

byte7=00

表 8-3 从站响应

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
RES	INDEX		SUB INDEX	RESERVED			

RES: 显示从站的响应:

0x60 数据成功传送

0x80 错误，由字节 4...7 产生

INDEX 16 位的地址 和主站传送的一样

SUBINDEX 8 位的子地址 和主站传送的一样

RESERVED 备用

例如：

主站向从站发送“下载”命令：

01 23 7A 60 00 E2 1D 00 00 03 （该命令写入从站的目标位置 607A0020）

从站响应：

01 60 7A 60 00 E2 1D 00 00 C6

表示：

01—从站地址为 1

60—传送到的数据为 2 个字节，由响应的 10 个字节中的 byte4...byte5 保存。

byte4=E2, byte5=1D, byte6=00, byte7=00

那么 DATA= byte7 byte6 byte5 byte4 = 1DE2 (hex) =7650 DEC

B:上传，就是主站发送命令读取从站内的对象地址，上传不存在的目标地址将产生错误。

表 8-4 主站传送：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
CMD	INDEX		SUB INDEX	RESERVED			

CMD 指定数据传输的方向

0x40

INDEX 16 位的地址

SUBINDEX 8 位的子地址

RESERVED 字节 4...7 不用

表 8-5 从站接收：

Byte0	Byte1	Byte2	Byte3	Byte4	Byte5	Byte6	Byte7
RES	INDEX		SUB INDEX	DATA			

RES 显示从站的响应：

0x43 字节 4...7 包含 32 位数据

0x4B 字节 4, 5 包含 16 位数据

0x4F 字节 4 包含 8 位数据

0x80 错误，字节 4...7 产生了错误

INDEX 16 位的地址 和主站传送的一样

SUBINDEX 8 位的子地址 和主站传送的一样

DATA 如果没有错误，byte4...byte7 共 4 个字节保存的是读取的从站对象内数值，低位在前，高位在后，正确的值=byte7, byte6, byte5, byte4；如果有错误，这 4 个字节内数据就不再等于读取的从站内的对象数值。例如：

主站向从站发送“上传”命令：

01 40 7A 60 00 00 00 00 00 E5 （该命令读取从站的目标位置 607A0020）

从站响应：

01 43 7A 60 00 E2 1D 00 00 E3

表示：

01—从站地址为 1

43—接收到的数据为 4 个字节，由响应的 10 个字节中的 byte4...byte5 保存。

byte4=E2, byte5=1D, byte6=00, byte7=00

那么 DATA= byte7 byte6 byte5 byte4 = 1DE2 (hex) =7650 DEC

8.1.5 RS232 通讯地址

各模式操作主要对象操作地址，详见模式控制章节。

常用对象操作地址见[附录七常用对象列表](#)。

全部通讯地址见参数列表章节。

RS232 通讯案例见附录。

8.2 RS485 通讯

8.2.1 RS485 通讯硬件接口

进驱动器 I0 端子的 RS485 通讯功能，可以用来修改步进驱动器内部参数以及监控步进驱动器状态等。接线图如下所示。

RS485 转接线引脚说明，由 9 针 D 型接口转 RJ45 接口

9 针 D 型头	I0 口
RX+ (2) & TX+ (3)	485_A
RX- (7) & TX- (8)	485_B
GND (5)	GND

- 注意：1、所有从站的 TX、RX 脚直接相接即可，采用并连的方式接线，不能采用星型连接方式；
2、主站端和最后一个从站端，需要接 120 欧姆的终端电阻；
3、通讯电缆请采用屏蔽双绞线，并做好接地处理；

8.2.2 RS485 通讯参数

表 8-6 通讯参数

名称	CANopen 地址	Modbus 地址	值	含义
设备站号	0x100B0008	0x0600	1	设备驱动器当前站号。 注：1) ID 站号可以由 SW6-SW1 设置 1~63 或由 0x2FE400 设置 1~127。 2) 更改该参数需要存储再重新启动。
设备站号(加偏移)	0x2FE40008	0x2800	127	设备驱动器站号 1~127；注：仅 SW6-SW1 都为“OFF”时，设置才有效。当更改该参数需要存储再重新启动。
RS485 波特率	0x2FE00010	0x2400	520	用于设置串口的波特率 设定值 波特率

				1041 520 259 86 需要重新启动	9600 19200 38400 115200
存储控制环参数	0x2FF00108	0x2910	1	1: 存储设定的所有配置参数; 10: 初始化所有的配置参数;	
其他参数			固定值	数据位 = 8 停止位 = 1 无奇偶校验	

8.2.3 Modbus RTU 通讯协议

驱动器 RS485 支持 MODBUS RTU 通讯协议，每个 8-bit 数据由两个 4-bit 的十六进制数据组成，即一般十六进制组成的数，数据结构为 11 Bit 字符格式，数据位为 8 Bit，校验方式为 CRC 校验。其内部对象为不连续的数据寄存器，（被上位机读写时映射为 4X）。

Modbus RTU 通讯协议基本格式

由发送设备将 Modbu 报文构造为带有已知起始和结束标记的帧，这使设备可以在报文的开始收新帧，并且知道何时报文结束。在 RTU 模式，报文帧由时长至少为 3.5 个字符时间表的空闲间隔区分。

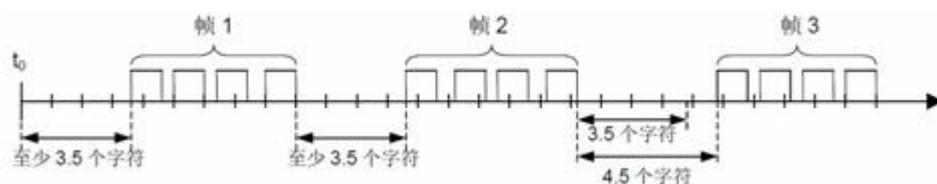


图 8-1 帧报文 1

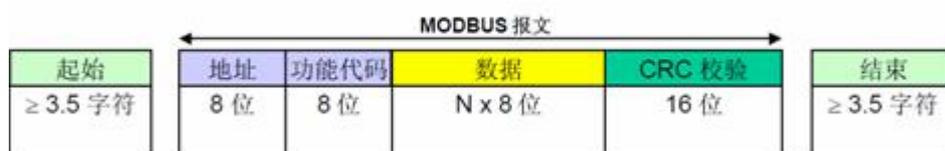


图 8-4 MODBUS 通讯格式

整个报文帧必须以连续的字符发送。如果两个字符之间的空闲间隔大于 1.5 个符时间，则报文帧被认为不完整应该被接收节点丢弃。

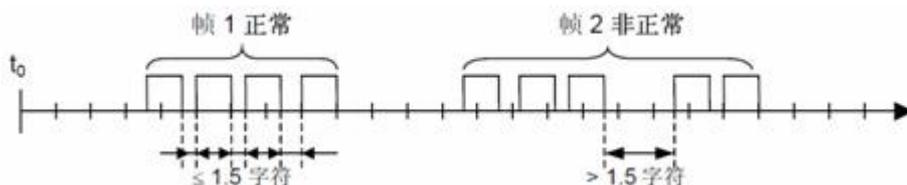


图 8-2 帧报文 2

8.2.4 Modbus RTU 常用功能码简介

功能码 0x03: 读数据寄存器

表 8-7 请求格式

目标站号	功能码	起始地址 高字节	起始地址 低字节	读取个数 高字节	读取个数 低字节	CRC
1 字节	03	1 字节	1 字节	1 字节	1 字节	2 字节

表 8-8 正确应答格式

站号	功能码	返回数据字 节数	寄存器 1 高字节	寄存器 1 低字节	...	CRC
1 字节	03	1 字节	1 字节	1 字节	...	2 字节

范例说明:

发送报文 01 03 32 00 00 01 8A B2 含义

01: ID 号

03: 功能码 读数据寄存器

32 00: 读对象“状态字”60410010 之 modbus 地址

00 01: 读取 word 数据个数

8A B2: 校验码

响应报文 01 03 02 00 31 79 90 含义

01: ID 号

03: 功能码 读数据寄存器

02: 返回 byte 数据个数

00 31: 返回对象“状态字”数据

79 90: 校验码

功能码 0x06: 写单数据寄存器

表 8-9 请求格式

目标站 号	功能 码	寄存器地址 高字节	寄存器地址 低字节	寄存器值 高字节	寄存器值 低字节	CRC
1 字节	06	1 字节	1 字节	1 字节	1 字节	2 字节

应答格式: 若设置成功, 原文返回。

范例说明:

发送报文 01 06 31 00 00 0F C7 32 含义

01: ID 号

06: 功能码 写单个 WORD

31 00: 写对象“控制字” 60400010 之 modbus 地址，数据长度为 1 个 WORD

00 0F: 写入数据 16 进制 000F

C7 32: 校验码

功能码 0x10: 写多保持寄存器

表 8-10 请求格式

目标站号	功能码	起始地址高字节	起始地址低字节	数量高字节	数量低字节	字节数	寄存器值 1 高字节	寄存器值 1 低字节	...	CRC
1 字节	10	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	1 字节	...	2 字节

表 8-11 正确应答格式

目标站号	功能码	起始地址高字节	起始地址低字节	数量高字节	数量低字节	CRC 校验码
1 字节	10	1 字节	1 字节	1 字节	1 字节	2 字节

范例说明:

发送报文 01 10 6F 00 00 02 04 00 00 00 32 9B 88 含义

01: ID 号

10: 功能码 写多个 WORD

6F 00: 写“目标速度” 60FF0020 之 modbus 地址，数据长度为 2 个 WORD

00 02: 写入 2 个 WORD

04: 数据长度为 4 个 BYTE (2 个 WORD)

00 00 00 32: 写入数据 16 进制 00320000，十进制 3276800，换算为 200RPM

9B 88: 校验码

响应报文 01 10 6F 00 00 02 5C DC 含义

01: ID 号

10: 功能码 读数据寄存器

6F 00: 对象地址

00 02: 写入 WORD 数量

5C DC: 校验码

若访问出错数，则作响应错误，则返回的异常功能码为 0x80 + 功能码

表 8-12 异常应答格式

站号	异常功能码	异常码	CRC
1 字节	1 字节	1 字节	2 字节

表 8-13 异常码值含义

异常码值	含义
0x01	功能码不支持
0x02	寄存器地址不存在
0x03	数据错误, 或寄存器数量不对
0x04	写操作失败, 包括数据超范围, 或对象只读。

8.2.5 Modbus 常用通讯地址

各模式操作主要对象操作地址见模式控制章节。

常用对象操作地址见[附录七常用对象列表](#)（不是所有内部对象都支持 RS485）。

RS485 通讯案例见[附录二](#)。

8.3 CAN 通讯

开放的现场总线标准中 CANopen 是最著名和成功的一种, 已经在欧洲和美国获得广泛的认可和大量应用。1992 年在德国成立了“自动化 CAN 用户和制造商协会” (CiA, CANinAutomation), 开始着手制定自动化 CAN 的应用层协议 CANopen。此后, 协会成员开发出一系列 CANopen 产品, 在机械制造、铁路、车辆、船舶、制药、食品加工等领域获得大量应用。目前 CANopen 协议已经欧洲最重要的工业现场总线标准 EN-50325-4。

CM 系列步进驱动器是标准的 CAN 从站设备, 遵循 CANopen2.0A/B 协议, 任何支持该协议的上位机均可以与其进行通讯。驱动器内部使用了一种严格定义的对象列表, 它称作对象辞典, 这种对象辞典的设计方式基于 CANopen 国际标准, 所有的对象有明确的功能定义。这里说的对象 (Objects) 类似常说的内存地址, 有些对象如速度和位置等可以由外部控制器修改, 有些对象却只能由驱动器本身修改, 如状态、错误信息。这些对象如下:

	Index	Sub	Bits	属性	含义
例如:	6040	00	16 (=0x10)	RW	设备状态控制字
	6060	00	8 (=0x08)	RW	工作模式
	607A	00	32 (=0x20)	W	目标位置
	6041	00	16 (=0x10)	MW	设备状态字

对象的属性有下面几种:

1. RW(读写): 对象可以被读也可以被写入。
2. RO(只读): 对象只能被读。
3. WO(只写): 只能写入。

4. M (可映射): 对象可映射, 类似间接寻址。
5. S (可存储): 对象可存储在 Flash-ROM 区, 掉电不丢失。
6. “Bits” 表示对象的数据类型, 0x08 为字节, 0x10 为整型, 0x20 为长整型。

8.3.1 硬件介绍

CAN 通讯协议主要描述设备之间的信息传递方式, CAN 层的定义与开放系统互连模型 OSI 一致, 每一层与另一设备上相同的那一层通讯, 实际的通讯发生在每一设备上相邻的两层, 而设备只通过物理层的物理介质互连, CAN 的规范定义了模型的最下面两层数据链路层和物理层。CAN 总线物理层没有严格规定, 能够使用多种物理介质例如双绞线光纤等, 最常用的是双绞线信号, 使用差分电压传送 (常用总线收发器产生), 两条信号线被称为 CAN_H 和 CAN_L, 静态时均是 2.5V 左右, 此时状态表示为逻辑 1, 也可以叫做隐位, 若 CAN_H 比 CAN_L 高表示逻辑 0, 称为显位, 此时通常电压值为 CAN_H = 3.5V 和 CAN_L = 1.5V, 竞争时显位优先。

CAN 转接线引脚说明, 由 9 针 D 型接口转 RJ45 接口

9 针 D 型头	IO 口
CAN_H (7) -----	CAN_H
CAN_L (2) -----	CAN_L
GND (3) -----	GND

■注意:

- 1、所有从站的 CAN_L、CAN_H 直接相接即可, 采用串连的方式接线, 不能采用星型连接方式;
- 2、主站端和最后一个从站需要接 120 欧姆的终端电阻。
- 3、CM 系列步进驱动不需要外部 24V 电源给 CAN 供电。
- 4、通讯电缆请采用屏蔽双绞线, 并做好接地处理
- 5、各种波特率所理论上能够通讯的最长距离:

表 8-14 波特率最大通讯距离

通讯速度	通讯距离
1Mbit/S	25M
500Kbit/s	100M
250Kbit/s	250M
125Kbit/s	500M
50Kbit/s	600M
25Kbit/s	800M

8.3.2 CAN 通讯参数

表 8-15 通讯参数

名称	CANopen 地址	Modbus 地址	值	含义
设备站号	0x100B0008	0x0600	1	设备驱动器当前站号。 注：1) ID 站号可以由 SW6-SW1 设置 1~63 或由 0x2FE400 设置 1~127。 2) 更改该参数需要存储再重新启动。
设备站号(加偏移)	0x2FE40008	0x2800	127	设备驱动器站号 1~127；注：仅 SW6-SW1 都为“OFF”时，设置才有效。当更改该参数需要存储再重新启动。
CAN 波特率	2F810008	0x2300	50	CAN 波特率设置 设定值 波特率 100: 1M 50: 500k 25: 250k 12: 125k 5: 50k 需要保存重新启动
存储控制环参数	0x2FF00108	0x2910	1	1: 存储设定的所有配置参数； 10: 初始化所有的配置参数；

8.3.3 EDS 文件

EDS（电子数据表格）文件是 PLC 所连接从站的标识文件或者类似码，通过该文件来辨认从站所属的类型（是 401、402、403 中的何种类别，或者属于 402 中的哪一种设备）。该文件包含了从站的所有信息，比如生产厂家、序列号、软件版本、支持波特率种类、可以映射的 OD 及各个 OD 的属性等参数，类似于 Profibus 的 GSD 文件。因此在进行硬件配置前，首先需要把从站的 EDS 文件导入到上位组态软件中

8.3.4 CANopen 通讯协议

8.3.4.1 SDO 协议

COB-ID 是 CANopen 通讯协议的特有方式，它的全称是 Communication Object Identifier-通讯对象-ID，这些 COB-ID 定义了相应的传输级别，有了这些传输级别后，控制器和驱动器就能够在各自的软件里配置里定义相同的传输级别和其里面的传输内容，这样控制器和驱动器都采用的同一个传输

级别和传输内容后，数据的传输即透明化了，也就是双方都知道所要传输的数据内容了，也就不需要在传输数据时还需要对方回复数据是否传输成功。

缺省ID分配表是基于CANopen 2.0A定义的11位CAN-ID（CANopen 2.0B协议COB-ID是27位），包含一个4位的功能码部分和一个7位的节点ID(Node-ID)部分。如图所示：

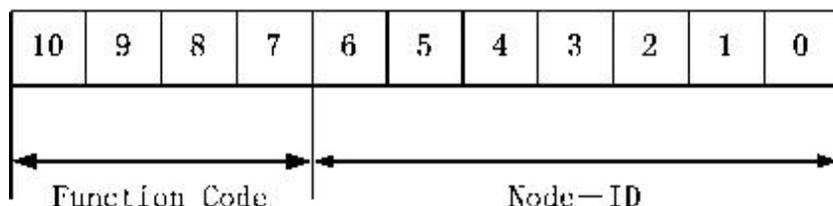


图 8-3 CAN-ID 分布格式

Node-ID由系统集成商定义，例如通过设备上的拨码开关设置(即步进驱动器的站号)。Node-ID范围是1~127（0不允许被使用）。

Function Code: 数据传输的功能码，定义各种PDO、SDO、管理报文的传输级别，功能码越小，优先级越高。

表 8-16 CANopen 预定义主/从连接集 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	0000	000H	-
SYNC	0001	080H	1005H, 1006H, 1007H
TIME SSTAMP	0010	100H	1012H, 1013H

表 8-17 CANopen 主/从连接集 CAN 标识符分配表:

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	0001	081H-0FFH	1024H, 1015H
TPD01(发送)	0011	181H-1FFH	1800H
RPD01(接收)	0100	201H-27FH	1400H
TPD02(发送)	0101	281H-2FFH	1801H
RPD02(接收)	0110	301H-37FH	1401H
TPD03(发送)	0111	381H-3FFH	1802H
RPD03(接收)	1000	401H-47FH	1402H
TPD04(发送)	1001	481H-4FFH	1803H
RPD04(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
NMT Error Control	1110	701H-77FH	1016H-1017H

注:

1. COB-ID越小, 优先级越高;
2. 每一个级别的COB-ID前面的功能码是固定格式;
3. COB-ID为00H、80H、100H、701H-77FH、081H-0FFH均为系统管理格式.

8.3.4.2 SDO 协议

SDO (Service data object) 主要用来在设备之间传输的低优先级的对象, 典型是用来对从设备进行配置、管理, 比如用来修改电流环、速度环、位置环的 PID 参数, PDO 配置参数等, 这种数据传输跟 MODBUS 的方式一样, 即主站发出后, 需要从站返回数据响应。这种通讯方式只适合对参数的设置, 不适合于对实时性要求较高的数据传输。

在 CANopen 协议中, SDO 用来访问一个设备的对象字典。访问者被称作客户 (client), 对象字典被访问且提供所请求服务的 CANopen 设备别称作服务器 (server)。客户的 CAN 报文和服务器的应答 CAN 报文总是包含 8 字节数据 (尽管不是所有的数据字节都一定有意义)。一个客户的请求一定有来自服务器的应答。

SDO 有 2 种传送机制:

- 加速传送 (Expedited transfer) : 最多传输 4 字节数据
- 分段传送 (Segmented transfer) : 传输数据长度大于 4 字节

表 8-16 SDO 的基本结构如 Client→Server/Server→Client

Byte0	Byte1-2	Byte3	Byte4-7
SDO Command specifier	对象索引	对象子索引	数据

SDO 命令字包含如下信息:

- 下载/上传 (Download / upload)
- 请求/应答 (Request /response)
- 分段/加速传送 (Segmented / expedited transfer)
- CAN 帧数据字节长度
- 用于后续每个分段的交替清零和置位的触发位 (toggle bit)

SDO 中实现了 5 个请求/应答协议:

- 启动域下载 (Initiate Domain Download);
- 域分段下载 (Download Domain Segment);
- 启动域上传 (Initiate Domain Upload);

- 域分段上传 (Upload Domain Segment)
- 域传送中止 (Abort Domain Transfer)。

下载 (Download) 是指对对象字典进行写操作, 上传 (Upload) 指对对象字典进行读操作。协议的 SDO 命令字 (SDO CAN 报文的第一个字节) 语法和细节在下面部分说明: (‘-’ 表示不相关, 应为 0)。

表 8-17 启动域下载

启动域下载 (Initiate Domain Download) Byte0							Byte1-3	Byte4-7	
Bit	7-5	4	3	2	1	0	对象索引及子索引		
Client →	ccs=1	-	n			e		s	数据
← Server	scs=3	-	-	-	-	-		-	

说明:

ccs: client command specifier, 1=启动下载请求。

scs: server command specifier, 3=启动下载响应。

n: 表示报文数据中无意义数据的字节数【从 (8-n) 字节到第 7 字节数据无意义】(当 e=1 且 s=1 时 n 有效, 否则 n 为 0)。

e: e=0 时正常传送, e=1 时加速传送。

s: 表示是否指明数据长度, 0 为数据长度未指明, 1 为数据长度指明。

e=0, s=0: 由 CiA 保留。

e=0, s=1: 数据字节为字节计数器, byte 4 是数据低位部分 (LSB), byte 7 是数据高位部分 (MSB)。

e=1, s=1: 数据字节数 4-n 为将要下载 (download) 的数据。

e=1, s=0: 下载字节数目不详的数据。

举例:

0x2f 为下载一个字节的的数据 (bytes 4 包含 8 位)

0x2b 为下载二个字节的数据 (bytes 4, 5 包含 16 位)

0x23 为下载四个字节的数据 (bytes 4, 5, 6, 7 包含 32 位)

0x21 为下载超过四个字节数据的起始帧, 分段下载数据

0x60 为数据成功下载

表 8-18 域分段下载

域分段下载 (Download Domain Segment) Byte0							Byte1-7
Bit	7-5	4	3	2	1	0	
Client →	ccs=0	t	n			c	数据
← Server	scs=1	t	-	-	-	-	-

说明:

n: 无意义的数据字节数。如果没有指明段长度, 则为 0。

c : 0 = 有后续分段需要 download, 1 = 最后一个段。

t : 触发位, 后续每个分段交替清零和置位 (第一次传送为 0, 等效于 request/response)。

举例:

0x00/0x10 为下载分段数据, 包含八个字节数据 (bytes 1-7)

0x0b 为最后一段数据, 包含二个字节数据 (bytes 1, 2)

0x20/0x30 为分段数据成功下载

表 8-19 启动域上传

启动域上传(Initiate Domain Upload) Byte0							Byte1-3	Byte4-7
Bit	7-5	4	3	2	1	0	对象索引及子索引	
Client→	ccs=2	-	-	-	-	-		-
←Server	scs=2	-	n		e	s		数据

说明: n, e, s: 与启动域下载相同

举例:

0x40 为请求上传对象的数据

0x4f 为上传一个字节的的数据 (bytes 4 包含 8 位)

0x4b 为上传二个字节的数据 (bytes 4, 5 包含 16 位)

0x43 为上传四个字节的的数据 (bytes 4, 5, 6, 7 包含 32 位)

0x41 为上传超过四个字节的数据的起始帧, 分段上传数据

表 8-20 域分段上传

域分段上传(Upload Domain Segment) Byte0							Byte1-7
Bit	7-5	4	3	2	1	0	
Client→	ccs=3	t	-	-	-	-	-
←Server	scs=0	t	n			c	数据

说明: n, c, t : 与域分段下载相同

举例: 0x60/0x70 为请求上传分段数据

0x00/0x10 为上传分段数据, 包含八个字节数据 (bytes 1-7)

0x19 为最后一段数据, 包含三个字节数据 (bytes 1, 2, 3)

表 8-21 域传送中止

域传送中止(Abort Domain Transfer) Byte0							Byte1-3	Byte4-7
Bit	7-5	4	3	2	1	0	对象索引及子索引	错误代码
Client→/←Server	cs=1	-	-	-	-	-		

举例: 0x80 为中断传送

表 8-22 错误代码

错误代码	含义
0x0504 0001	客户机/服务器命令符无效或未知
0x0601 0002	尝试写一个只读对象
0x0602 0000	对象不在对象字典内
0x0604 0041	对象不能被映射到 PDO
0x0607 0010	数据类型不匹配
0x0609 0011	子索引不存在
0x0609 0030	参数值无效,不在范围内
0x0800 0000	一般错误

例子 1: 读取控制字参数

表 8-23 发送 SDO 报文

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	40	40	60	00	00			

0x600+Node_ID -- COB-ID

40 -- 请求上传对象的数据

60 40 00 -- 控制字对象地址

00 -- 无用数据

表 8-24 返回 SDO 报文

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	4b	40	60	00	06	00	00	00

0x580+Node_ID -- COB-ID

4b -- 上传二个字节的数据 (bytes 4,5 包含 16 位)

60 40 00 -- 控制字对象地址

0006 -- 数据

例子 2: 修改控制字参数时

表 8-25 发送 SDO 报文

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x600+Node_ID	8	2b	40	60	00	2f	00	00	00

0x600+Node_ID -- COB-ID

2b -- 下载二个字节的数据 (bytes 4,5 包含 16 位)

60 40 00 -- 控制字对象地址

002f -- 数据

表 8-26 返回 SDO 报文

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x580+Node_ID	8	60	40	60	00	2f	00	00	00

0x580+Node_ID -- COB-ID

60 -- 数据成功下载

60 40 00 -- 控制字对象地址

002f -- 数据

8.3.4.3 PDO 协议

PDO (Process data object) 一次性可传送 8 个字节的数据, 没有其它协议预设定 (意味着数据内容已预先定义), 主要用来传输需要高频率交换的数据。PDO 的传输方式采用全新的数据交换模式, 设备双方在传输前先在各个设备定义好数据接收和发送区域, 在数据交换时直接发送相关的数据到对方的数据接收区即可, 减少了问答式的询问时间, 从而极大的提高了总线通讯的效率, 从而得到了极高的总线利用率。

1) 发送PDO (TPDO)

发送PDO相对于步进来说就是指步进驱动器发送出去的数据, 这些数据由PLC来接收。发送PDO的功能码 (COB-ID) 为:

- 1、 0x180+步进站号
- 2、 0x280+步进站号
- 3、 0x380+步进站号
- 4、 0x480+步进站号

2) 接收PDO (RPDO)

接收PDO相对于步进来说就是指步进接收的数据, 这些数据由PLC来发送, 发送PDO的功能码 (COB-ID) 为:

- 1、 0x200+步进站号
- 2、 0x300+步进站号
- 3、 0x400+步进站号
- 4、 0x500+步进站号

由于FM系列步进是按标准的CANopen 2.0A协议来设计的, 但同时也支持CANopen 2.0B协议, 也就

是说，如果上面的8个PDO不够用的情况下，您还可以定义新的PDO，比如用0x43FH来作为1号站的通讯PDO，只要控制器和步进驱动器都按照这个来定义即可。

3) PDO 传输类型

PDO 有两种传输方式：

1、同步 (SYNC)：由同步报文触发传输（传输类型：0-240）在该传输模式下，控制器必须具有发送同步报文的能力（频率最高为 1KHZ 的周期发送的报文），步驱动器在接收到该同步报文后在发送。

- 非周期：由远程帧预触发传送，或者由设备子协议中规定的对象特定事件预触发传送。该方式下步进驱动器每接收到一个同步报文 PDO 里的数据即发送一次。
- 周期：传送在每 1 到 240 个 SYNC 消息后触发。该方式下步进驱动器每接收到 n 个同步报文后，PDO 里的数据发送一次

2、异步(传输类型：254/255)。从站报文数据改变后即发送，不管主站是否询问，而且可以定义同一个报文两次发送之间的时间间隔，避免高优先级报文一直占据总线（PDO 的数值越低优先级越高）。

对于 CM 系列步进驱动器，目前支持同步周期传输和异步传输方式。

4) PDO 禁止时间

一个 PDO 可以指定一个禁止时间，即定义两个连续 PDO 传输的最小间隔时间，避免由于高优先级信息的数据量太大，始终占据总线，而使其它优先级较低的数据无力竞争总线的问题。禁止时间由 16 位无符号整数定义，单位 1ms。

5) 例子：

用 TPDO 更新“目标速度”，使用对象 0x1A00 来更新对象，使用对象 0x1800 配置 0x1A00 映射对象的属性。

表 8-28 对象 0x1800 属性

对象 0x1800 属性		
子索引	值	含义
0	3	对象有 3 个子索引
1	0x250+Node_ID	节点 ID
2	254	异步传输
3	50	禁止传输时间间隔

表 8-29 对象 0x1A00TPDO 映射

对象 0x1A00 TPDO 映射		
子索引	值	含义
0	1	1 个对象映射到 PDO 中

1	0x60ff0020	对象 0x60ff, 子索引 0x00, 由 32 位组成
---	------------	-------------------------------

表 8-30 发送报文格式

Identifier	DLC	Daten							
		0	1	2	3	4	5	6	7
0x250+Node_ID	8	目标速度							

8.3.4.4 Boot-up 过程

在网络初始化过程中, CANopen 支持扩展的 boot-up, 也支持最小化 boot-up 过程。扩展 boot-up 是可选的, 最小 boot-up 则必须被每个节点支持。两类节点可以在同一个网络中同时存在。如果使用 CAL 的 DBT 服务进行 ID 分配, 则节点必须支持扩展 boot-up 过程。

可以用节点状态转换图表示这两种初始化过程, 如下图所示。扩展 boot-up 的状态图在预操作和操作状态之间比最小化 boot-up 多了一些状态。

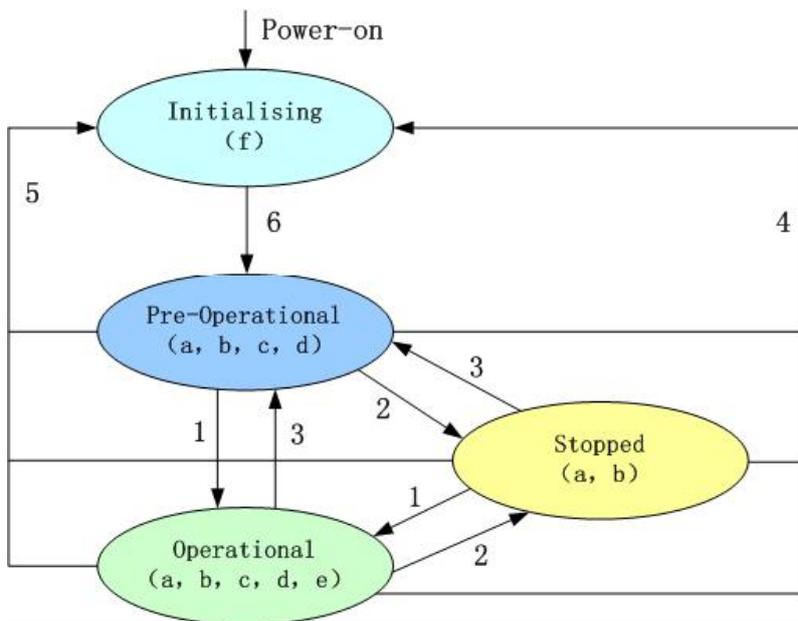


图 8-4 节点状态转换图

注:

- 图中括号内的字母表示处于不同状态那些通讯对象可以使用:
 - NMT , b. Node Guard , c. SDO , d. Emergency , e. PDO , f. Boot-up
- 状态转移 (1-5 由 NMT 服务发起),
 - 1: 启动远程节点 (Start_Remote_node)
 - 2: 停止远程节点 (Stop_Remote_Node)
 - 3: 进入预操作状态 (Enter_Pre-Operational_State)
 - 4: 复位节点 (Reset_Node)

5: 复位通讯(Reset_Communication)

6: 设备初始化结束, 自动进入“预操作”状态, 发送 Boot-up 消息

在任何时候 NMT 服务都可使所有或者部分节点进入不同的工作状态。NMT 服务的 CAN 报文由 CAN 头(COB-ID=0)和两字节数据组成; 第一个字节表示请求的服务类型(‘NMT command specifier’), 第二个字节是节点 ID, 或者 0 (此时寻址所有节点)。

8.3.4.5 NMT 模块控制 (NMT Module Control)

可以通过 NMT 管理报文来实现在各种模式之间切换, 只有 NMT-Master 节点能够传送 NMT Module Control 报文, 所有从设备都必须支持 NMT 模块控制服务, 同时 NMT Module Control 消息不需要应答。NMT 消息格式如下:

表 8-29 NMT-Master → NMT-Slave(s)

COB-ID	Byte 0	Byte 1
0x000	CS	Node-ID

当 Node-ID=0, 则所有的 NMT 从设备被寻址。CS 是命令字, 可以取如下值

表 8-30 CS 命令值

命令字	NMT 服务
0x01	启动远程节点
0x02	停止远程节点
0x80	进入预操作状态
0x81	复位节点
0x82	复位通讯

例如, 如果要让一个处于操作状态的 2 号节点返回到预操作状态, 那么控制器发送如下的报文即可:

表 8-31 控制器报文

COB-ID	Byte 0	Byte 1
0x000	0x80	0x02

8.3.4.6 保护方式 (监督类型)

监督类型是指在运行过程中主站选择何种检查方式检查从站, 方式有两种: 心跳报文(Heartbeat)和节点保护(Node Guarding)。通过这两种方式来判断从站是否出现故障, 并根据这些故障做出相应的处理!

- 心跳报文：从站“监督时间”周期性的发送报文到主站，如果超过“心跳消费者时间”后主站还没有收到从站的下一个心跳报文，那么主站判断从站出错！Heartbeat Producer → Consumer(s)

表 8-32 心跳报文

COB-ID	Byte 0
0x700 + Node_ID	状态

表 8-33 状态值

状态值	含义
0x00	启动(Boot-up)
0x04	停止(Stopped)
0x05	运行(Operational)
0c7f	预操作(Pre-operational)

当一个 Heartbeat 节点启动后它的 Boot-up 报文是其第一个 Heartbeat 报文。

- 节点保护：主站以“监督时间”周期性的发送报文到从站，从站接收到后即回应，如果超过“监督时间*寿命因子”时间后，主站还没有收到从站回应的报文，那么主站判断从站出错！

NMT-Master 节点发送远程帧（无数据）如下：

NMT-Master → NMT-Slave

COB-ID
0x700+Node_ID

NMT-Slave 节点发送如下报文应答：

表 8-34 NMT-Master → NMT-Slave

COB-ID	Byte0
0x700 + Node_ID	Bit 7 : 触发位 Bit6-0 : 状态

Byte0 数据部分包括一个触发位 (bit7)，触发位必须在每次节点保护应答中交替置“0”或者“1”。

触发位在第一次节点保护请求时置为“0”。位 0 到 6 (bits0~6) 表示节点状态。

表 8-35 状态值含义

状态值	含义
0	初始化(Initializing)
1	未连接(Disconnected)*
2	连接(Connecting)*
3	准备(Preparing)*
4	停止(Stopped)
5	运行(Operational)
127	预操作(Pre-operational)

注意：带 * 号的状态只有支持扩展 boot-up 的节点才提供。注意状态 0 从不在节点保护应答中出现，

因为一个节点在这个状态时并不应答节点保护报文。

一个节点不能够同时支持 Node Guarding 和 Heartbeat 协议，只能选其中一作保护。

8.3.10 CANopen 常用通讯地址

各模式操作主要对象操作地址见模式控制章节。

常用对象操作地址见 [附录七常用对象列表](#)。

全部通讯地址见参数列表章节。

CANopen 通讯案例见附录。

第九章 报警排除

9.1 报警信息

EPR 红灯开启或闪烁时，表明驱动器出现报警故障，具体故障参照“错误状态字”或“自检错误状态字”。报警信息代码为十六进制数据。当驱动器出现故障的时候，报警代码里面对应的位就会亮红灯。



表 9-1 故障报警及处理措施

报警信息	指示灯		报警原因	处理措施
	RUN	ERR		
驱动器内部错误	慢闪	快闪	1、电机配置错误 2、驱动器内部问题	1、详见“历史错误”窗口里的“自检错误状态” 2、联系厂家
驱动器输出短路	熄灭	快闪	1、电机相线短路 2、驱动器内部问题	1、检查电机接线 2、联系厂家
驱动器总线电压过高	快闪	快闪	1、动力电电源电压过高 2、高速停止场合反馈能量过高	1、检查动力电源 2、加制动电阻
驱动器总线电压过低	熄灭	开启	1、动力电源电压过低 2、急速启动	1、检查动力电源 2、减小加速度

驱动器温度过高	熄灭	慢闪	驱动器功率模块超过 80 度	检查使用环境温度是否大于 40 度
EEPROM 内部错误	快闪/慢闪	开启	1、更新驱动器底层程序造成 2、驱动器内部问题	初始化参数后保存再重新启动
寻找电机错误	快闪	开启	1、电机未接线或接线错误 2、电机配置错误	1、检查电机接线 2、详见“历史错误”窗口里的“自检错误状态”
内部逻辑电压异常	慢闪	开启	内部逻辑电压 15V 或 5V 不在正常值范围	联系厂家
5V 输出电流过载			5V 输出电流过大	请检查 5V 负载接线
跟踪误差			负载过大或者卡死。	检查负载或减小加速度。
总线错误			总线通讯关闭	检查总线通讯参数
输入脉冲频率过高			输入脉冲频率超过频率允许最大值。	检查输入脉冲频率是否超过最大值
外部预使能信号			I0 口配置外部预使能信号，而外部没有输入有效信号	检查外部接线和确认输入信号
正限位报警			I0 口配置了正限位，驱动器检测到有效信号输入。	
负限位报警			I0 口配置了负限位，驱动器检测到有效信号输入。	

注：慢闪频率为 0.5Hz，快闪频率为 5Hz

9.2 历史错误窗口



	名称	数据	单位
1*	错误状态字高16位	0	HEX
2*	错误状态字低16位	0	HEX
3*	自检错误状态字高16位	0	HEX
4*	自检错误状态字低16位	0	HEX
5*	错误计数	14	DEC
6*	错误组当前位置	5	DEC
7*	错误0-错误状态字低16位	101	HEX
8*	错误0-自检错误状态字低16位	8000	HEX
9*	错误0-总线电压	28	V
10*	错误0-速度	0.000	rpm
11*	错误0-电流	0.000	Arms
12*	错误0-工作模式	0	DEC
13*	错误0-时间(月)	0	Mon
14*	错误0-时间(分钟)	342	Min
15*	错误0-温度	27	degree

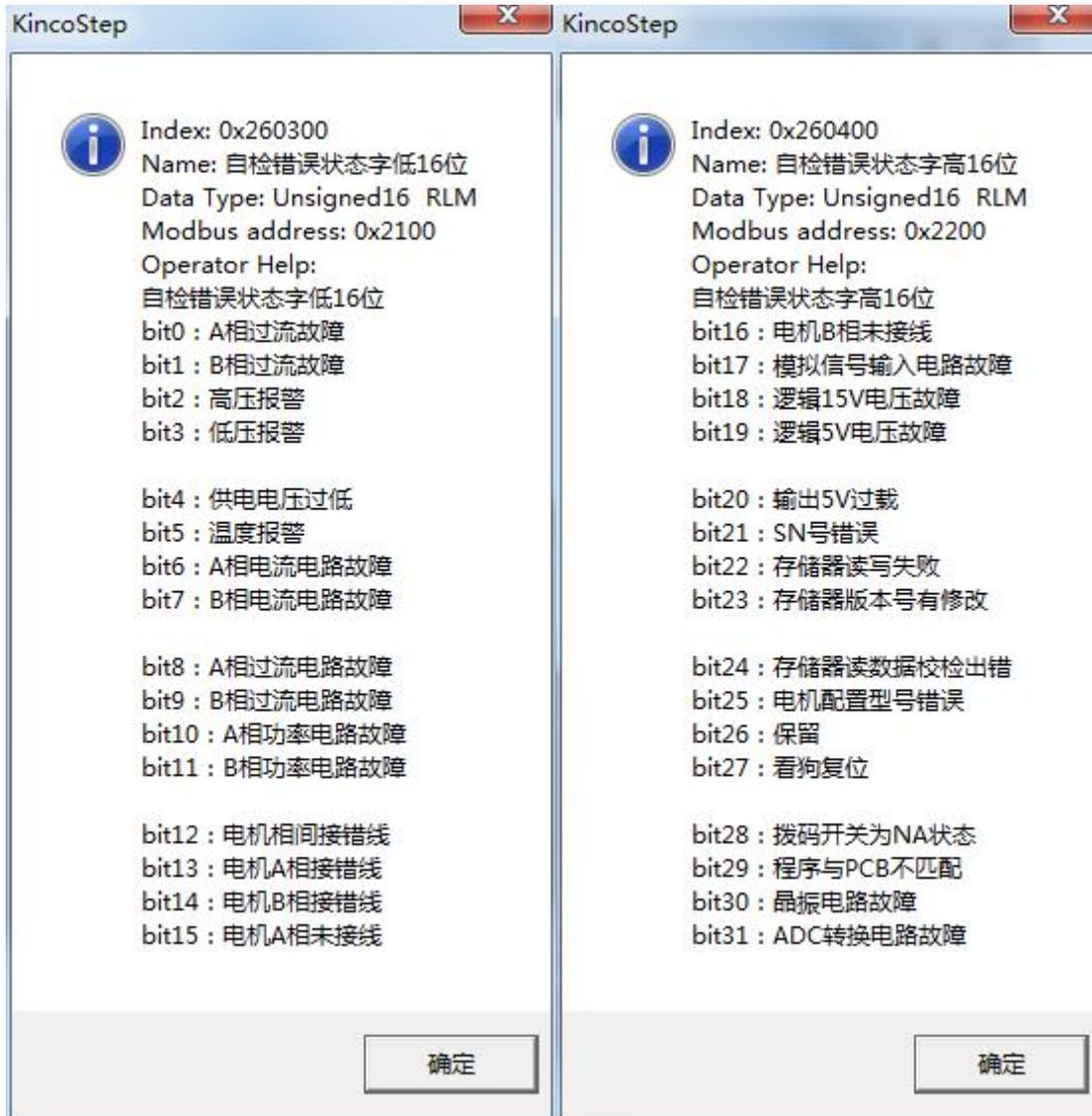
驱动器提供 8 组历史报警信息，用户可以查询到报警发生时的报警代码，电压，电流，温度，速度，工作模式，驱动器累计工作时间等信息，更好的方便用户设备维护。

“错误状态字低 16 位/高 16 位”是驱动器运行过程出现的错误，而“自检错误状态字低 16 位/高 16 位”是驱动器上电自检测硬件出现的错误。

“错误组当前位置”值指示最新发生的错误保存在那一组，数值 N 指示当前报警在“错误 N”，上一次记录是 N 减 1，下一次记录是 N 加 1，8 组错误记录循环覆盖。

例如：错误组当前位置=4，表明当前的报警为“错误 4”。下一次的报警位置为“错误 5”。

9.3 自检错误状态字定义



第十章 附录

附录一 CANopen 总线通讯范例

驱动器与 F1 PLC 利用 CANopen 总线进行通讯

1. 硬件接线

F1 PLC CAN 接口	CAN 接口
CAN_H (7)	CAN_H
CAN_L (2)	CAN_L
GND (3)	GND

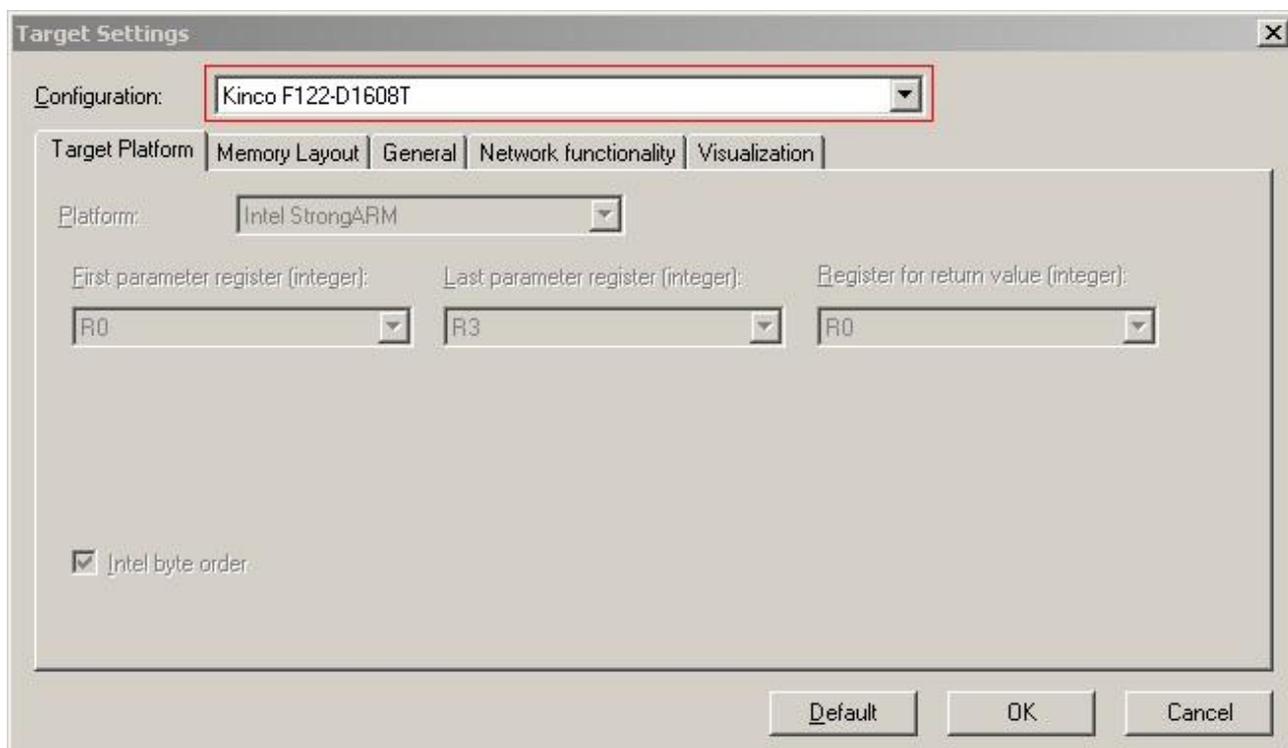
■注意:

- (1) 多个从站的采用串连的方式接线, 不能采用星型连接方式;
- (2) PLC 端 CAN1 和 CAN2 是完全独立的, 可以任意使用
- (3) PLC 有终端电阻, 通过拨码开关选择。

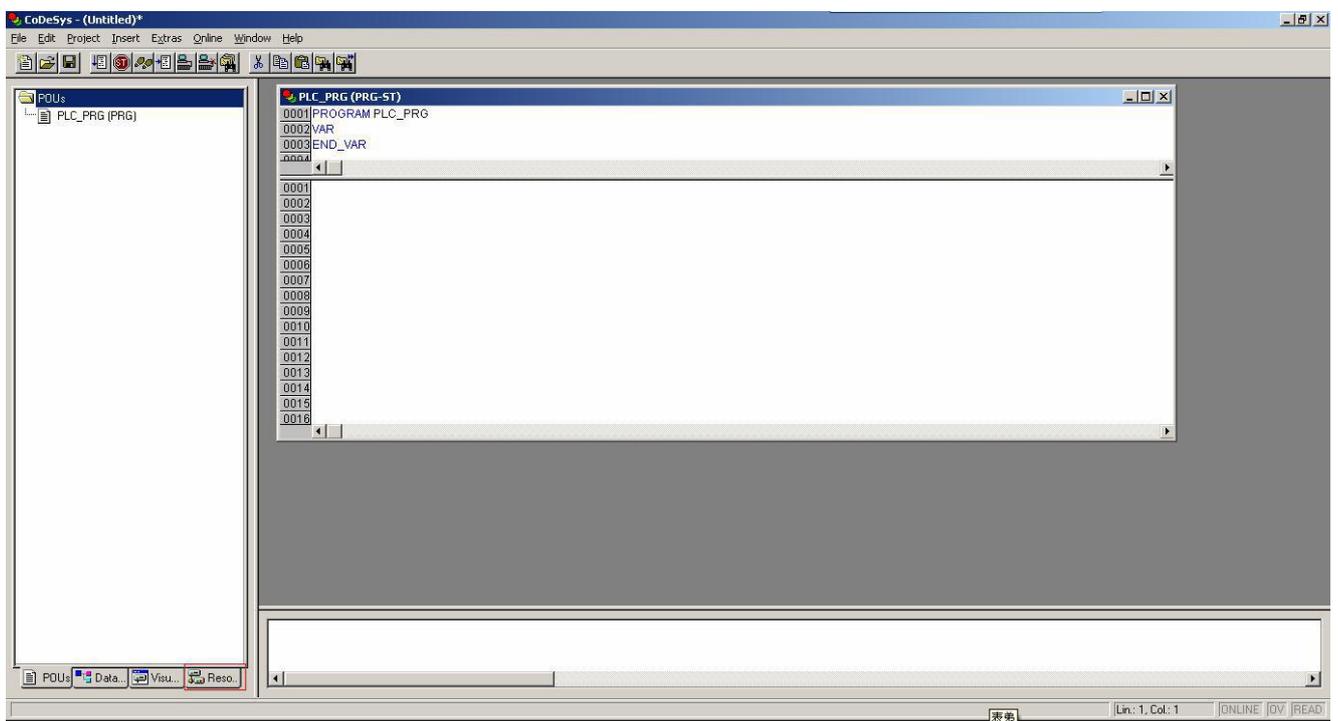
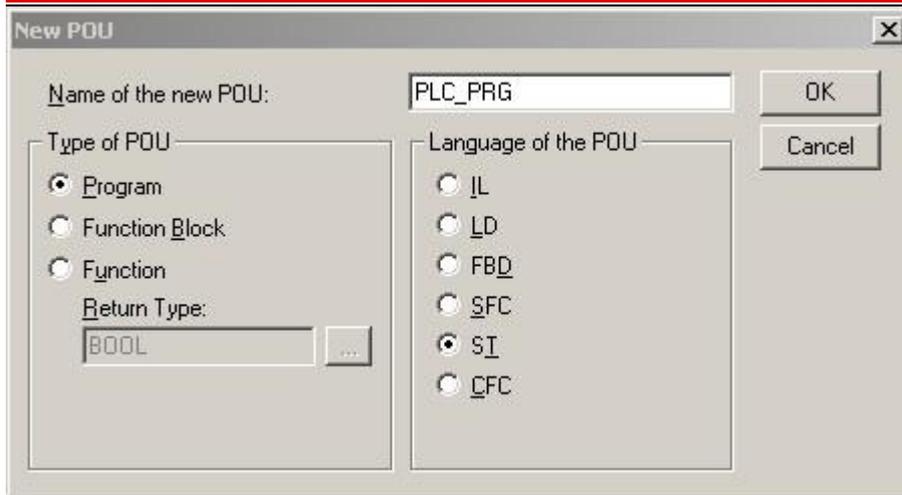
2、参数设置。波特率, 站号等参数设置参考 8.3 CANopen 总线通讯

3、软件编程。它与 FM860 驱动器的编程是一样的。

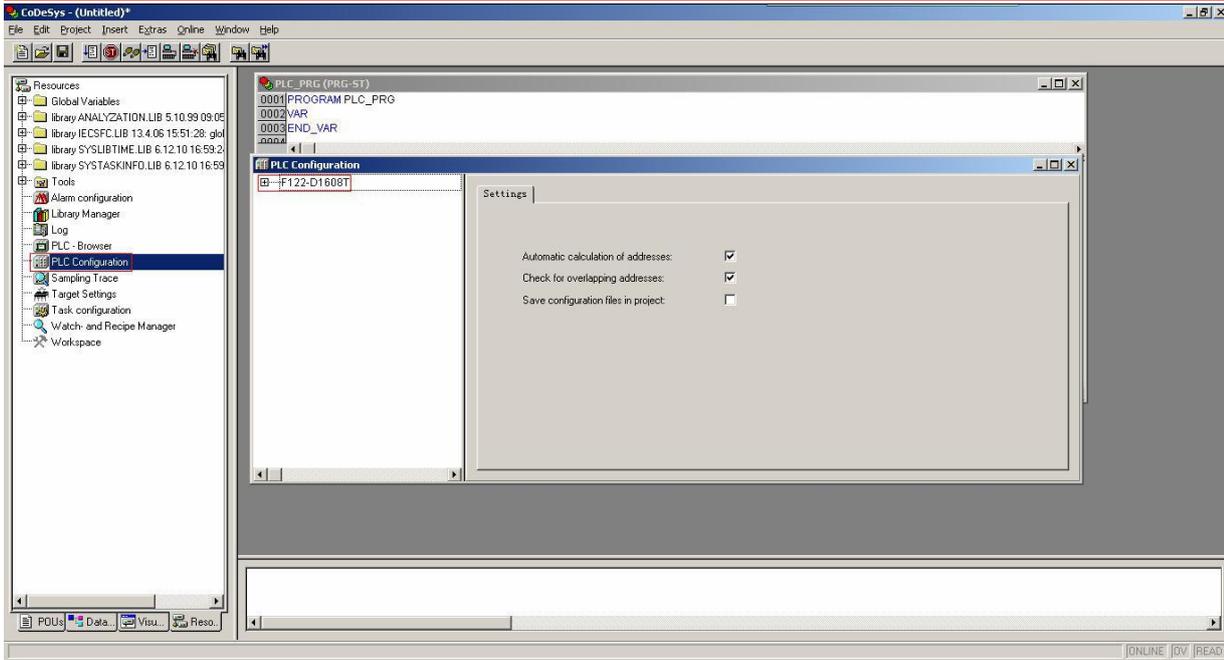
- (1) 新建工程, 选择 Kinco F122-D1608T 运行平台, 点击 OK。



- (2) 根据自己的喜好选择合适的编程语言, 然后点击 OK。



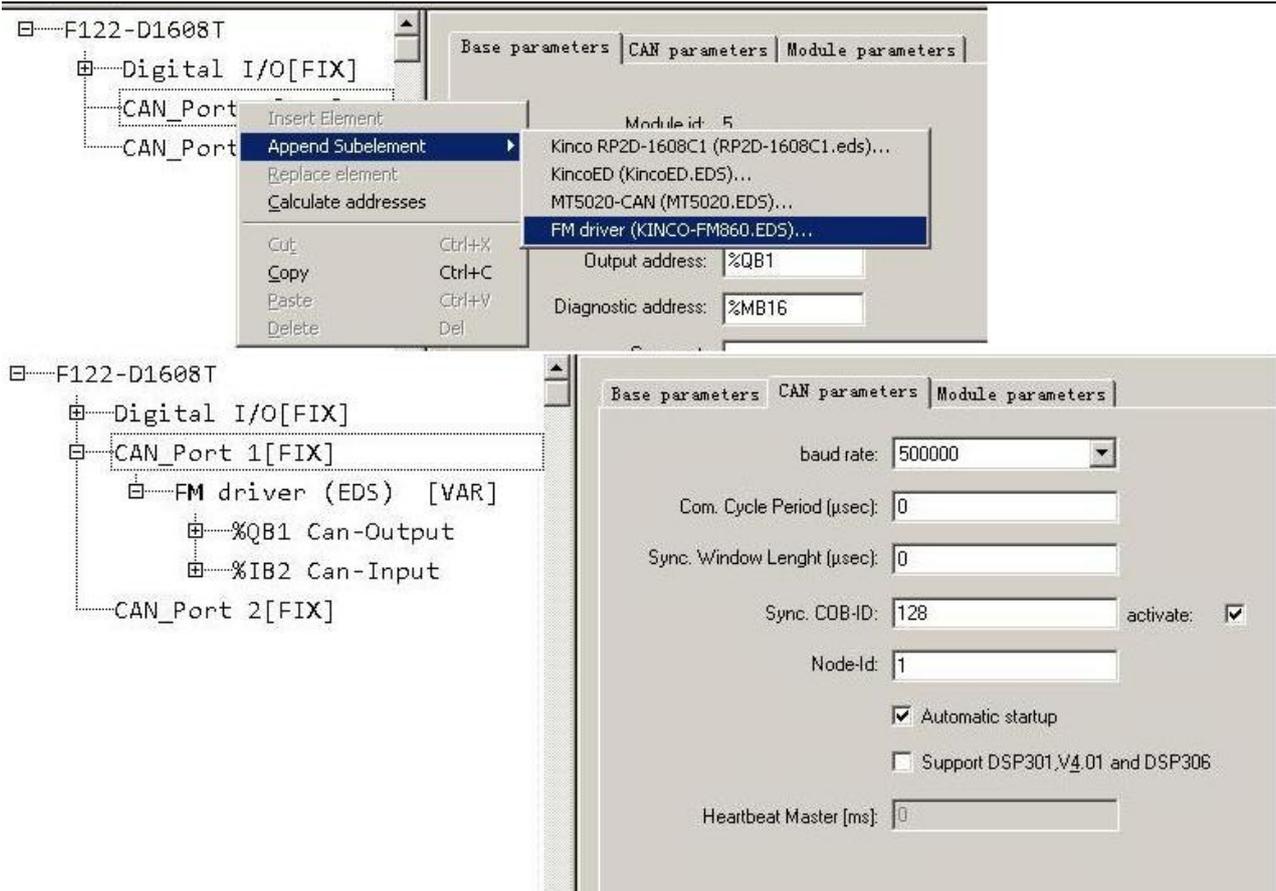
(3) 新建工程完成后的界面如下图所示，然后点击红色方框内“Resources”进入 PLC Configuration 页面。



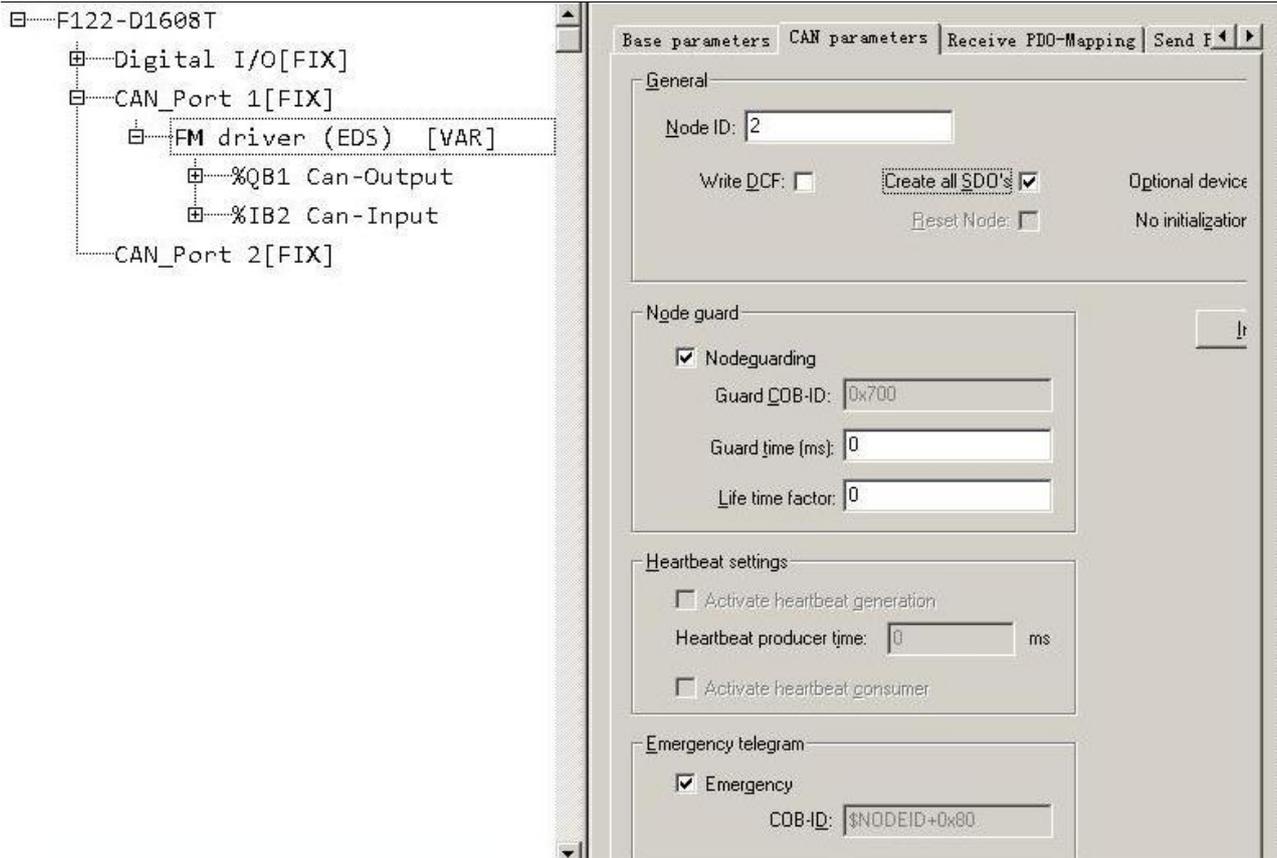
(4) 然后在 F1 下面添加 CANopen 从站，首先需要添加从站的 EDS 文件到 Codesys 里面来，点击主菜单栏里的“Extras/add configuration file”选项，弹出一个对话框，找步进的 EDS，点击打开即可，系统会提示您这个 EDS 文件放置的位置，点击 OK 即可；



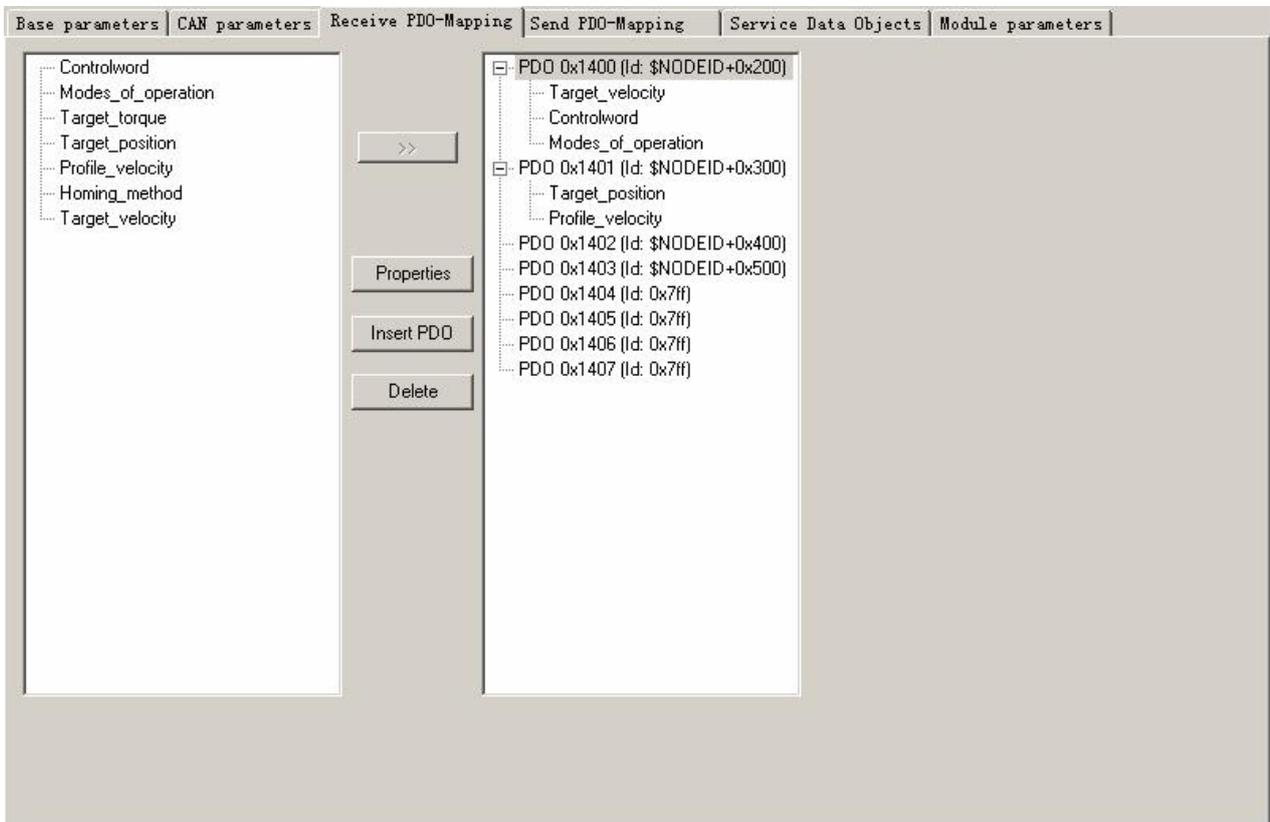
(5) F1 有两个 CAN 口，都可以做主站，设置主站的站号、波特率。如果需要同步报文，请先激活“active”，然后在“Com. Cycle period”里设置同步报文的传输周期和 COB-ID；

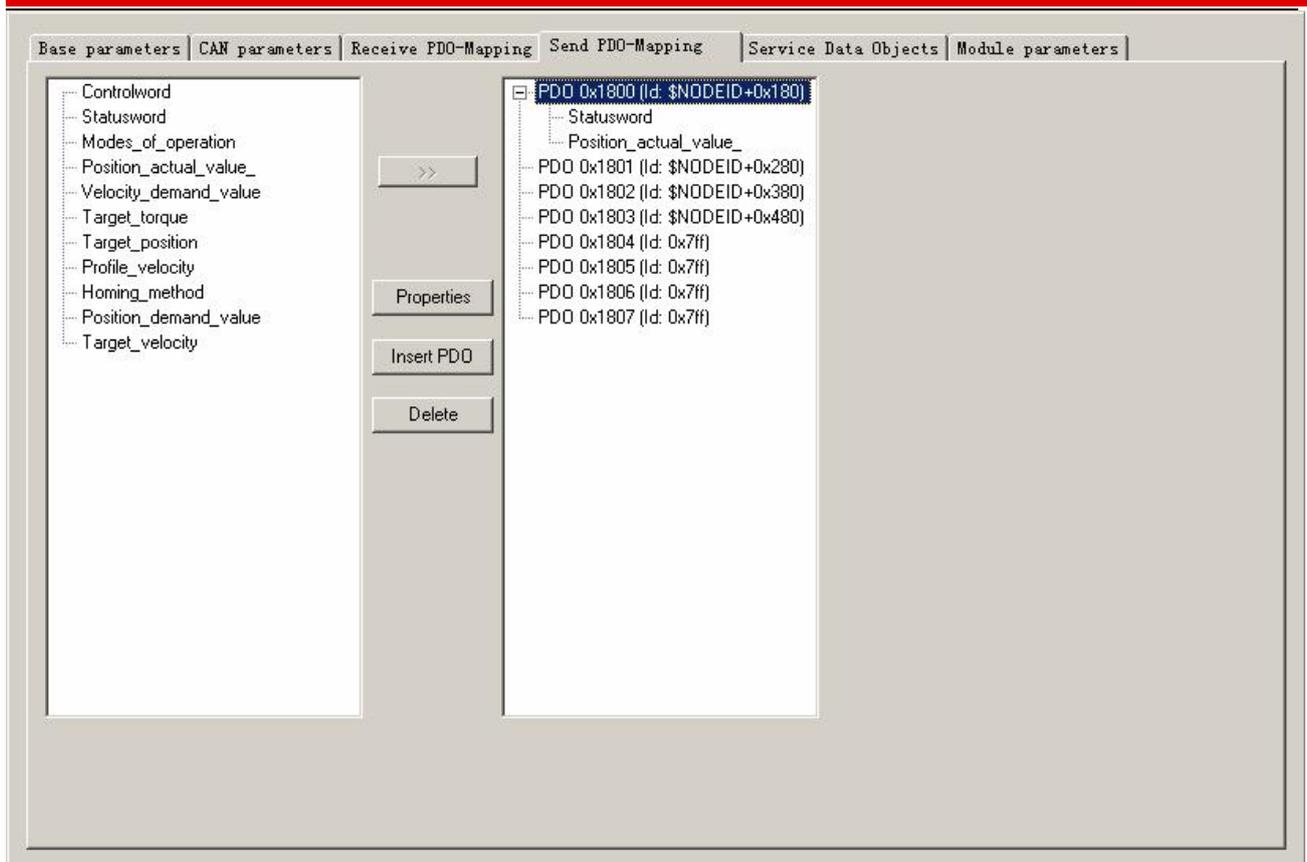


(6) 然后在选中“CanMaster”并点击右键，选择“Append FM Driver”，系统就会把 FM 步进驱动器添加到网络中了；按照这种方式添加您所想要添加的从站个数即可；添加完从站后，需要一一设置每个从站的通讯起始地址、站号、节点保护类型、RX-PDO、TX-PDO 等参数。

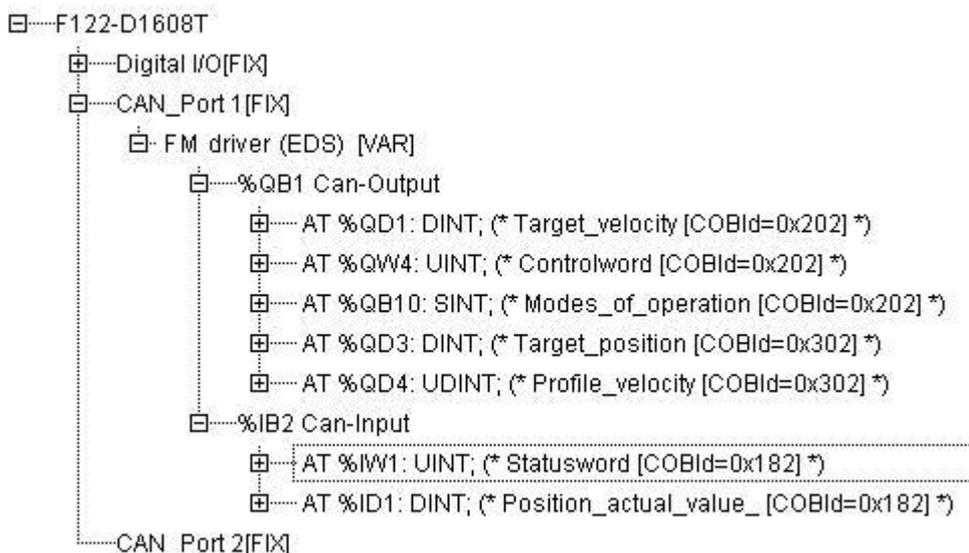


(7) 根据需要配置相应从站的 PDO 对象

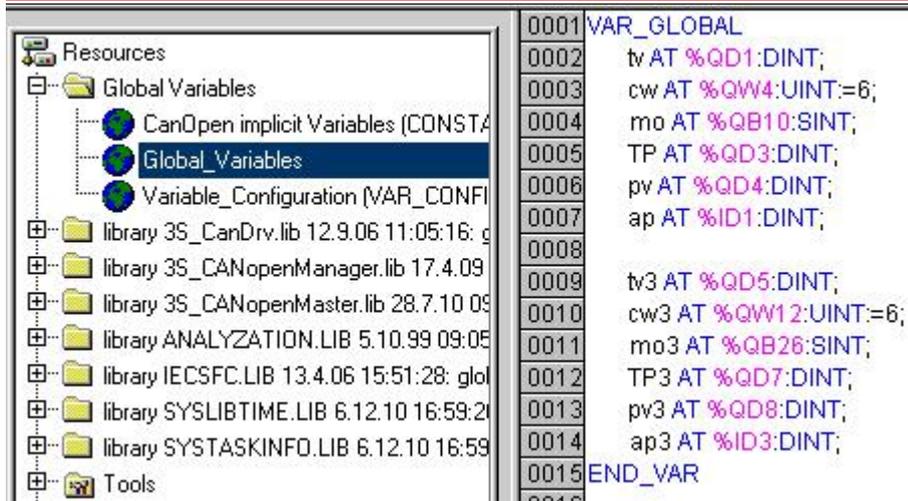




(8) 设置完成所有的步进驱动器后，点击硬件配置里的树型结构，就可以看到映射到每个 PDO 里的 OD 了，同时还能看到每个 OD 对应的输入和输出寄存器，从右图可以看到，ID2 步进的 Controlword 对应的寄存器为 QW4，而 Statusword 对应的寄存器为 IW1，通过在软件里控制这些寄存器就可以控制步进了；



(9) 配置多个从站时参照以上的方法，配置完成后就可以编程控制 FM860 步进驱动器了。可以先在全局变量中定义好对应的变量名便于编程引用，也可以直接引用相应的地址。



(10) 程序如下图所示。详细各模式控制要点参考[第 6 章模式操作](#)。（进行控制之前请参考[第 7 章通讯功能](#)注意要点）。注意在 F1 与步进建立 CAN 通讯后，首先需要初始化步进控制字为 f，否则步进对其他指令无响应。

```
0001 PROGRAM JD2
0002 VAR
0003     m1: BOOL;
0004     m2: BOOL;
0005     spv: DINT;
0006     m3: BOOL;
0007     m4: BOOL;
0008     m5: BOOL;
0009     actual_pos: DINT;
0010     m6: BOOL;
0011 END_VAR
0001 actual_pos:=ap;
0002 (* power off*)
0003 IF m1=1 THEN
0004     tv:=27300;
0005     cw:=6;
0006     mo:=3;
0007     m1:=0;
0008 END_IF
0009 (* velocity*)
0010
0011 IF m2=1 THEN
0012     tv:=273000;
0013     cw:=47;
0014     mo:=3;
0015     m2:=0;
0016 END_IF
0017
0018 (* absolute position*)
0019 IF m3=1 THEN
0020     tp:=0;
0021     pv:=2730000;
0022     cw:=63;
0023     mo:=1;
0024     m3:=0;
0025 END_IF
```

(11) 如果遇到一些 EDS 文件中没有的、不常用的对象，可以采用 SDO 的方式来读写相应的对象。格式如下图所示，对于 CAN1 和 CAN2 接口，只是 wDrvNr 总线接口号码不同，CAN1 是 0，CAN2 是 1

```

0001 PROGRAM sdo
0002 VAR
0003   sdo1:CanOpenSendSDO;
0004   m9: BOOL;
0005   m10: BOOL;
0006   sdo2: CanOpenSendSDO;
0007   ww: ARRAY [0..7] OF BYTE;
0008   val: DWORD;
0009 END_VAR
0010
0011 (*SDO写速度*)
0012 sdo1( Enable:= m9,(*使能*)
0013       wDrvNr:= 0, (*总线接口号, CAN1固定为0*)
0014       ucNodeId:=2, (*从站站号*)
0015       wIndex:=16#60FF, (*OD INDEX*)
0016       bySubIndex:=16#00, (*subINDEX*)
0017       ucModus:= 16#23, (*use 16#23 FOR 4-BYTE-write-request
0018                           use 16#27 FOR 3-BYTE
0019                           use 16#2B for 2-byte
0020                           use 16#2F for 1-byte
0021                           use 16#21 FOR downloading more than 4 bytes using the segmented transfer*)
0022       ucByte0 :=16#10,
0023       ucByte1 :=16#A8,
0024       ucByte2 :=16#29,
0025       ucByte3 :=00) ;
0026 (*SDO读实际速度*)
0027 sdo2(Enable:= m10, (*使能*)wDrvNr:= 0, (*总线接口号, CAN1固定为0*)ucNodeId:=2, (*从站站号*)
0028      wIndex:=16#606C, (*OD INDEX*)
0029      bySubIndex:=16#00, (*subINDEX*)
0030      ucModus:=16#40);(*SDO-mode, use 16#40 for read-request*)
0031 IF sdo2.bAnswerRec THEN
0032   val := SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[7]),24);
0033   val := val + SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[6]),16);
0034   val := val + SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[5]),8);
0035   val := val + BYTE_TO_DWORD(sdo2.ucAnswerBytes[4]);
0036 END_IF

```

驱动器与 Peak CAN 利用 CANopen 总线进行通讯

Peak 公司的 CAN 适配器有 ISA、PCI、USB-CAN 等多种产品, 提供了 Windows 98/ME 和 Windows 2000/XP 的设备驱动 (*. vxd 和 *. sys) 和动态连接库 (*. dll), 支持的软件有 VB、VC、Delphi 和 BCB 等。下面以 Peak 公司的 PCAN-USB 总线卡连接驱动器, 使用 PCAN-View 软件通过 CANopen 总线控制驱动器的说明。

其它软件编程可参考操作此范例!

- (1) 参照 PCAN-USB 的硬件手册安装驱动
- (2) 硬件连接

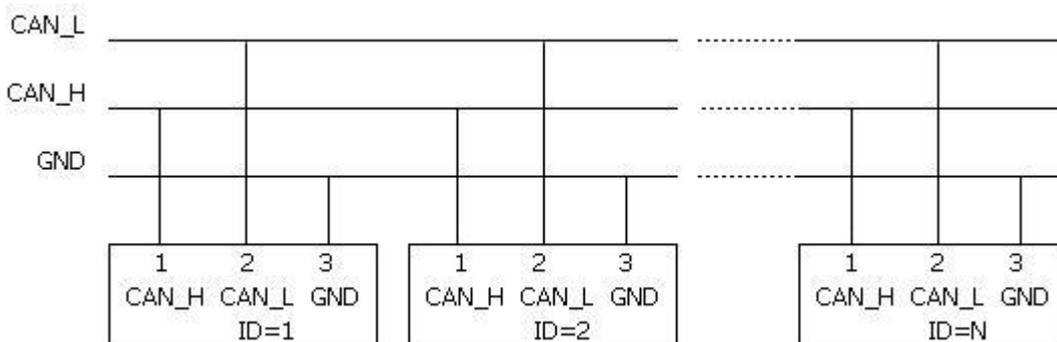


图 10-1 多台驱动器串联

总线终端在 CAN_L 和 CAN_H 间连接 120—150 欧姆的电阻。

3) 参考总线通讯章节设定 FM 的 ID 地址和波特率，默认波特率 500K，站号 1。更改必须保存重启有效。

4) 完成上述步骤后，就可以参照 CANopen 通讯协议来对 CM 步进驱动器进行控制。由于 PCAN-View 软件无法导入 EDS 文件不方便对 PDO 操作，本例按照通讯协议规定的数据格式对 CM 步进各种模式进行控制。（进行控制之前请参考第 7 章通讯控制注意要点）

下图示例发送 6040 为 3F，图中下方是发送数据，上面是返回数据

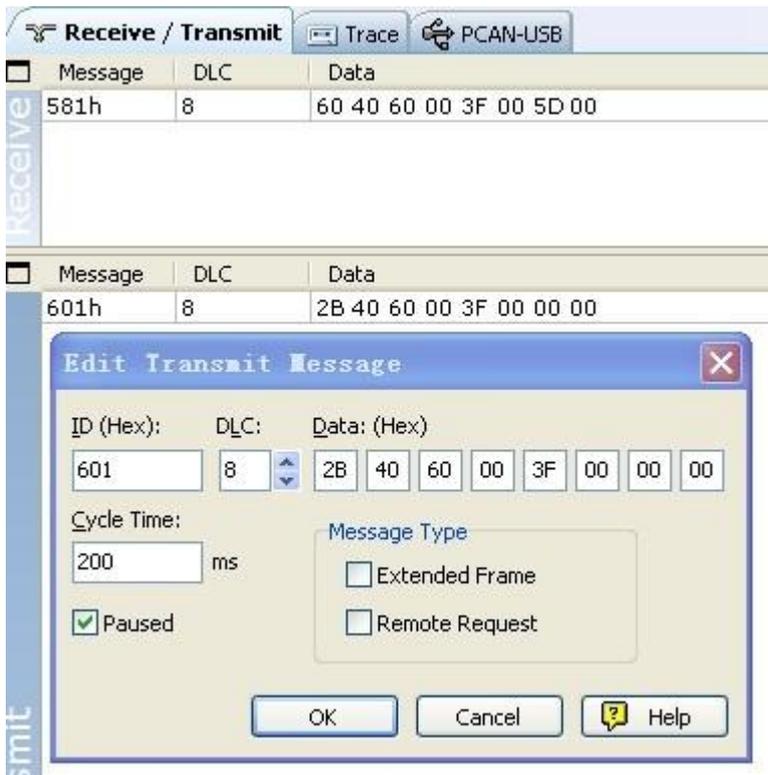


表 10-1 原点模式发送和接收数据报文

原点控制模式（控制字先 F 后 1F）				
CANopen 地址	变量名称	设置值	发送及回复报文（ID=1）	备注
6060008	工作模式	0x6	601 2F 60 60 00 06 00 581 60 60 60 00 06 00	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875]
6098008	原点模式	0x14	601 2F 98 60 00 14 00 581 60 98 60 00 14 00	
60990120	原点转折 信号速度	200RPM	601 23 99 60 01 00 00 32 00 581 60 99 60 01 00 00 32 00	
60990220	原点信号	150RPM	601 23 99 60 02 00 80 25 00	

	速度		<u>581 60 99 60 02 00 80 25 00</u>	
60400010	控制字	0xF	<u>601 2B 40 60 00 0F 00</u> <u>581 60 40 60 00 0F 00</u>	
60400010	控制字	0x1F	<u>601 2B 40 60 00 1F 00</u> <u>581 60 40 60 00 1F 00</u>	
	<u>601 40 41 60 00 00 00 00 00</u>		读取状态字，9437 表示原点找到	

表 10-2 位置控制模式发送和接收数据报文

位置控制模式（控制字绝对定位先 2F 后 3F 相对定位先 4F 后 5F，103F 立即更新）				
CANopen 地址	变量名称	设置值	报文（ID=1）	备注
60400010	控制字	0xF	<u>601 2B 40 60 00 0F 00</u> <u>581 60 40 60 00 0F 00</u>	
60600008	工作模式	0x1	<u>601 2F 60 60 00 01 00</u> <u>581 60 60 60 00 01 00</u>	
607A0020	目标位置	50000 DEC	<u>601 23 7A 60 00 50 C3 00 00</u> <u>581 60 7A 60 00 50 C3 00 00</u>	
60810020	梯形速度	200RPM	<u>601 23 81 60 00 00 00 32 00</u> <u>581 60 81 60 00 00 00 32 00</u>	速度 RPM 需要转换为内部单位
60830020	梯形加速度	10rps/s	使用默认值	DEC
60840020	梯形减速度	10rps/s	使用默认值	DEC=[(RPM*512*60000)/1875]
				梯形加减速默认单位 DEC，
				DEC=[(RPS/S*65536*60000)/100
				0/4000]
60400010	控制字	0x2F	<u>601 2B 40 60 00 2F 00</u> <u>581 60 40 60 00 2F 00</u>	
		0x3F(绝对定位)	<u>601 2B 40 60 00 3F 00</u> <u>581 60 40 60 00 3F 00</u>	
		0x4F	<u>601 2B 40 60 00 4F 00</u> <u>581 60 40 60 00 4F 00</u>	
		0x5F(相对定位)	<u>601 2B 40 60 00 5F 00</u> <u>581 60 40 60 00 5F 00</u>	
			<u>601 40 41 60 00 00 00 00 00</u>	读取状态字，D437 表示位置到

表 10-3 速度控制模式发送和接收数据报文

速度控制模式				
CANopen 地址	变量名称	设置值	报文 (ID=1)	备注
60600008	工作模式	0x3	<u>601 2F 60 60 00 03 00</u> <u>581 60 60 60 00 03 00</u>	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875] 梯形加减速默认单位 DEC, DEC=[(RPS/S*65536*60000)/1000/4000]
60FF0020	目标速度	150RPM	<u>601 23 FF 60 02 00 80 25 00</u> <u>581 60 FF 60 02 00 80 25 00</u>	
60400010	控制字	0xF	<u>601 2B 40 60 00 0F 00</u> <u>581 60 40 60 00 0F 00</u>	
60830020	梯形加速 度	10rps/ s	使用默认值	
60840020	梯形减速 度	10rps/ s	使用默认值	

注意：通讯模式下，数据以十六进制格式传输。

附录二 RS485 串口通讯范例

驱动器与 KINCO 触摸屏 Modbus 协议通讯

注意:

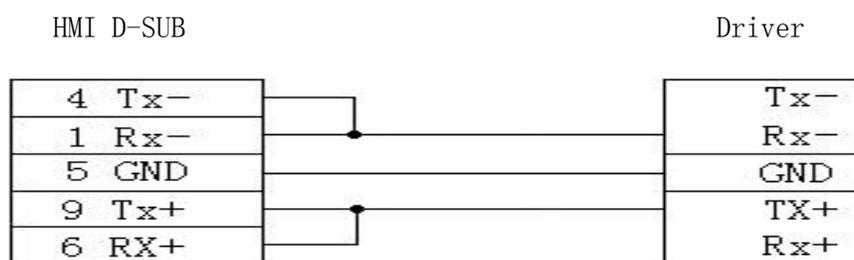
1: Kinco 全系列文本和触摸屏等都可以与 FM 驱动器 RS485 串口连接, 本范例只介绍 Kinco 触摸屏的主打产品 MT4000, MT5000 系列与步进驱动器连接。其它系列产品等请参考各产品使用手册或咨询 Kinco 技术人员。

2: 本范例只介绍简单连接, 用户可到 Kinco 网站上“下载中心-资料下载-范例程序”下载“Kinco 与 FM 步进驱动器与 485 通讯范例”了解详细应用。

Kinco MT4000, MT5000 系列触摸屏可以与 JD 驱动器 RS485 串口连接, 用户可以用触摸屏设置驱动器的内部参数和运行状态。触摸屏既可以与单个驱动器相联, 也可以与多个驱动器相联。

(一) 触摸屏控制单台 FM 步进

a. 硬件接线



b. 通讯参数设置

触摸屏选择 Modbus RTU 驱动, 触摸屏的通讯参数设置见图“D1.1”, 注意的是 PLC 的站号就是 CM 驱动器的 ID 号, CM 驱动器出厂时 ID 默认为 1, 所以触摸屏控制单台 CM 步进时 PLC 站号设置为 1。如果 FM 的 ID 号设置为 N, 那么下面的 PLC 站号也要设置为 N。波特率, 站号等参数设置参照 485 通讯协议章节。



c. 地址参数设置

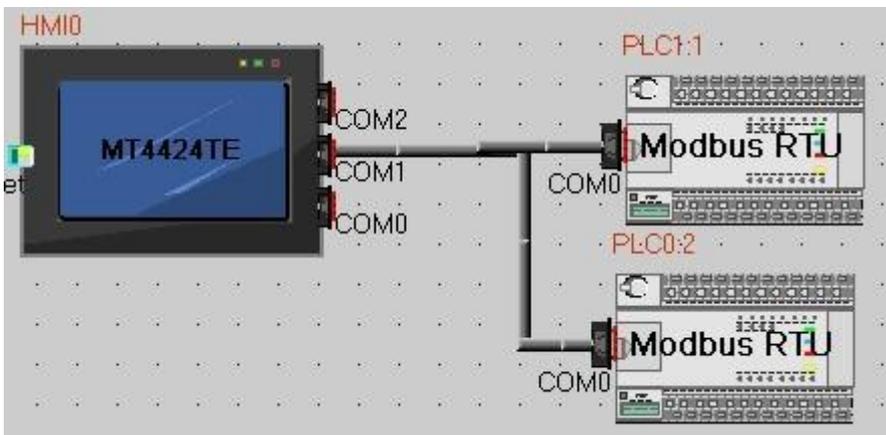
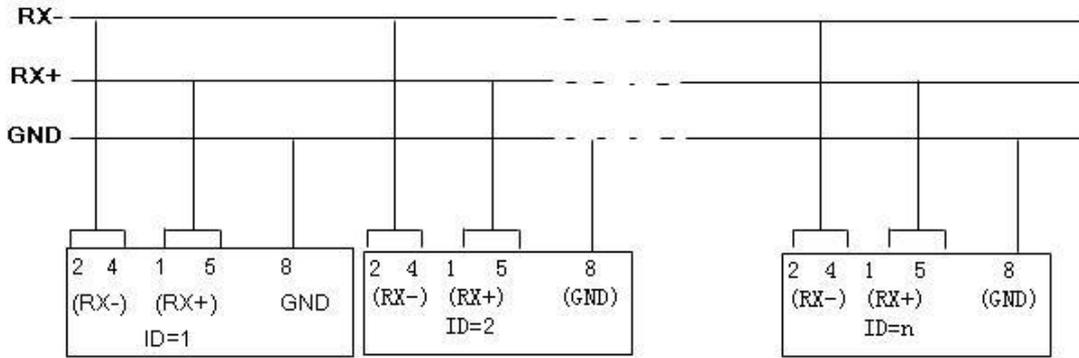
编写 HMI 程序时, 选择地址类型为 4X, (CM 所有对象为 4X, 而且全部不连续) 参照附录常用对象列表。如下图设置数值输入元件为 CM 内部对象 60FF0020 (目标速度)。其 Modbus 地址为 0x6F00, 换算成十进制+1=28417



(二) 触摸屏控制多台 CM 步进驱动器

a. 硬件接线

HMI



b. 参数设置

触摸屏参数设置同各台步进驱动器分别设置不同站号即可。在元件属性里面屏只要选择 PLC 编号，设置参数就会自动对应选中的步进驱动器（注意这里的 PLC 编号并不是设置的步进站号，仅仅表示用户连接多台步进驱动器的顺序，如上图，PLC0:2 表示的就是 PLC 编号是 0，站号是 2）。



驱动器与调试工具 Modbus 协议通讯

CM 步进 RS485 通讯口支持 Modbus RTU 通讯协议，用户可以通过 VB、VC 等软件按照 modbus 协议编写程序对 FM 步进驱动器进行控制。下面以常见串口 Modbus 调试工具软件通过 Modbus 协议控制 CM 步进驱动器进行说明。其它软件编程可参考操作此范例！

1. 接线，PC 需通过 RS232-RS485 模块，转换模块 485 端通过如下线缆连接 FM 驱动器

转换模块 RS485 端	驱动器 RS485 口 (X1)
D+	----- (RX+, TX+)
D-	----- (RX-, TX-)
GND	----- GND

2. 参考串口通讯章节设定的 ID 地址和波特率，默认波特率 19200，站号 1。更改必须保存重启有效。
3. 完成上述步骤后，就可以参照 Modbus RTU 通讯协议协议来对步进驱动器进行控制。（进行控制之前请参考第八章通讯控制注意要点）

图示例读对象 0x60400010 “控制字”（Modbus 地址 0x3100）数据。

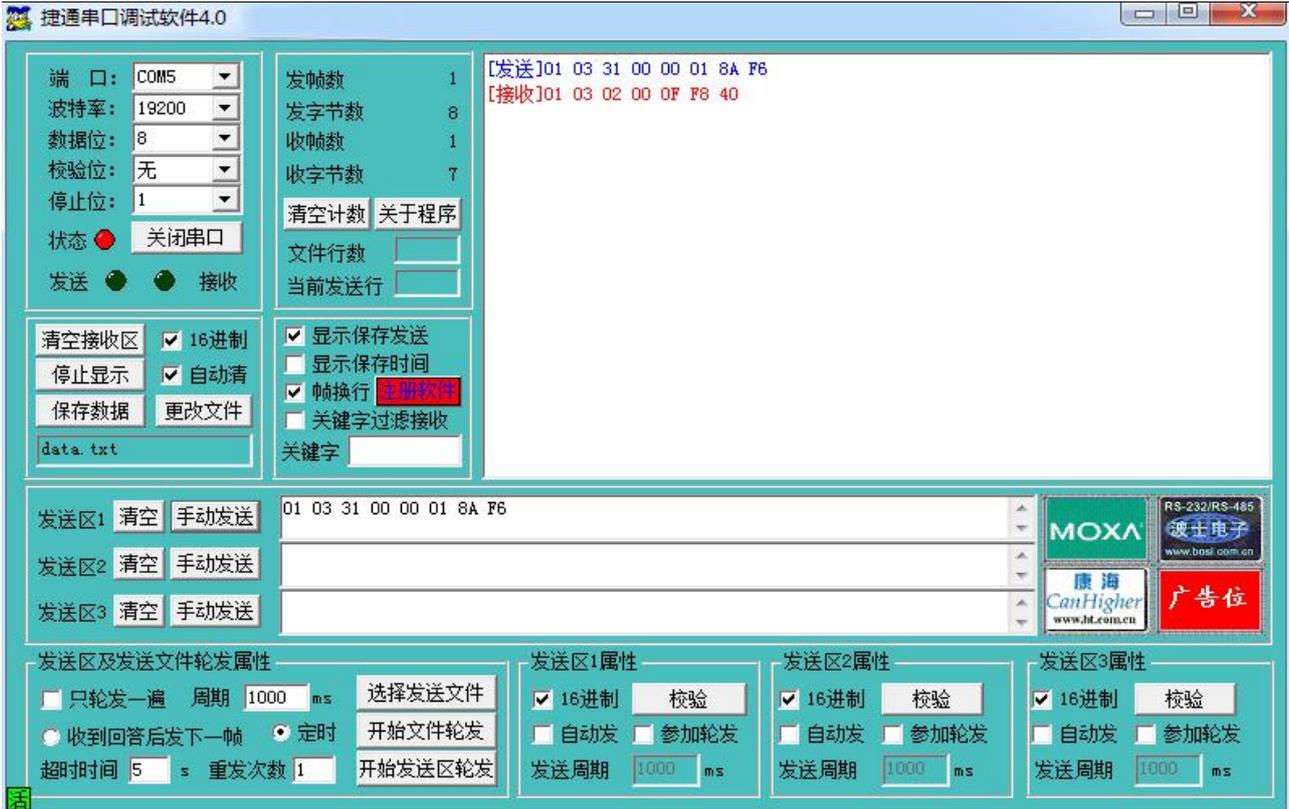


表 10-4 原点和位置控制模式发送报文

原点控制模式（控制字先 F 后 1F）				
Modbus 地址	名称	设置值	发送及回复报文（ID=1）	备注
0x3500	工作模式	0x6	01 06 35 00 00 06 06 04 01 06 35 00 00 06 06 04	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875]
0x4D00	原点模式	0x14	01 06 4D 00 00 14 9E A9 01 06 4D 00 00 14 9E A9	
0x5010	原点转折信号速度	200RPM	01 10 50 10 00 02 04 00 00 00 32 8F 75 01 10 50 10 00 02 51 0D	
0x5020	原点信号速度	150RPM	01 10 50 20 00 02 04 80 00 00 25 E5 AF 01 10 50 20 00 02 51 02	
0x3100	控制字	0xF	01 06 31 00 00 0F C7 32 01 06 31 00 00 0F C7 32	
0x3100	控制字	0x1F	01 06 31 00 00 1F C6 FE 01 06 31 00 00 1F C6 FE	
01 03 32 00 00 01 8A B2			读取状态字，9437 表示原点找到	
位置控制模式（控制字绝对定位先 2F 后 3F 相对定位先 4F 后 5F，103F 立即更新）				

Modbus 地址	变量名称	设置值	发送及回复报文 (ID=1)	备注
0x3100	控制字	6	<u>01 06 31 00 00 06 07 34</u> <u>01 06 31 00 00 06 07 34</u>	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875] 梯形加减速默认单位 DEC, DEC=[(RPS/S*65536*60000)/1000/4000]
0x3500	工作模式	1	<u>01 06 35 00 00 01 47 C6</u> <u>01 06 35 00 00 01 47 C6</u>	
0x4000	目标位置	50000DEC	<u>01 10 40 00 00 02 04 C3 50 00 00 FE</u> <u>39</u> <u>01 10 40 00 00 02 54 08</u>	
0x4A00	梯形速度	200RPM	<u>01 10 4A 00 00 02 04 00 00 00 32 3D</u> <u>19</u> <u>01 10 4A 00 00 02 57 D0</u>	
0x4B00	梯形加速度	10rps/s	使用默认值	
0x4C00	梯形减速度	10rps/s	使用默认值	
0x3100	控制字	2F	<u>01 06 31 00 00 2F C6 EA</u> <u>01 06 31 00 00 2F C6 EA</u>	
		3F(绝对定位)	<u>01 06 31 00 00 3F C7 26</u> <u>01 06 31 00 00 3F C7 26</u>	
		4F	<u>01 06 31 00 00 4F C6 C2</u> <u>01 06 31 00 00 4F C6 C2</u>	
		5F(相对定位)	<u>01 06 31 00 00 5F C7 0E</u> <u>01 06 31 00 00 5F C7 0E</u>	

表 9-5 速度控制模式发送报文

速度控制模式				
Modbus 地址	变量名称	设置值	报文 (ID=1)	备注
0x3500	工作模式	3	<u>01 06 35 00 00 03 C6 07</u> <u>01 06 35 00 00 03 C6 07</u>	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875] 梯形加减速默认单位 DEC, DEC=[(RPS/S*65536*60000)/1000/4000]
0x6F00	目标速度	150RPM	<u>01 10 6F 00 00 02 04 80 00 00</u> <u>25 F2 46</u> <u>01 10 6F 00 00 02 5C DC</u>	

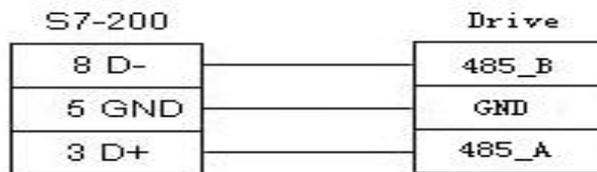
0x4B00	梯形加速度	10rps/s	使用默认值
0x4C00	梯形减速度	10rps/s	使用默认值
0x3100	控制字	F	01 06 31 00 00 0F C7 32 01 06 31 00 00 0F C7 32

注意：通讯模式下数据以十六进制格式传输。

驱动器与西门子 S7-200Modbus 协议通讯

1、硬件接线

单台硬件接线如下图，多台硬件接线参考 485 通讯章节多台接线说明



2、通讯参数设置

CM 步进驱动器参数设置参考 485 通讯章节，默认参数 Modbus RTU, 波特率 19200, 无校验

S7-200 PLC 通讯参数设置使用软件自带库函数通讯，利用库函数初始化指令设置对应通讯参数。

如下图！

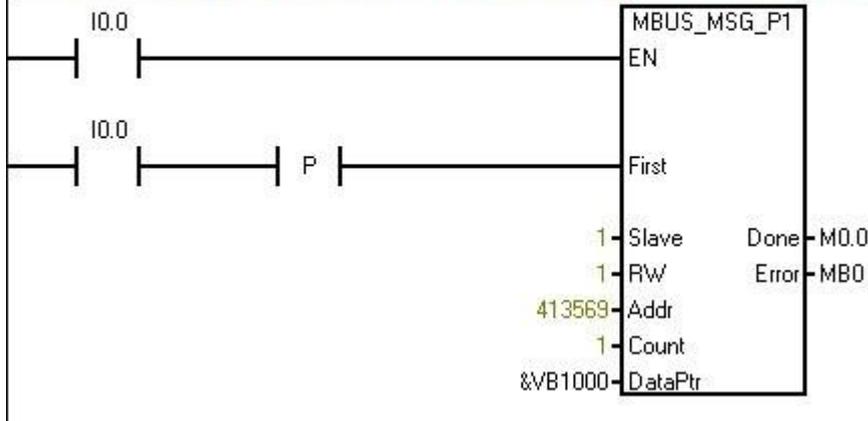


图 9-11 S7-200 PLC 通讯参数

3、程序编制（进行控制之前请参考第七章通讯控制注意要点）

PLC 软件中利用库函数自带 MODBUS_MSG 指令发送或者接收数据，函数各部分定义说明如下图

I0.0写MODBUS地址0x3500对应十进制13569 映射为4X 设定为位置控制模式
 发送数据VB1000=0 VB1001=1 设定工作模式为1
 其中 slave表示JD站号 RW为0表示读 为1表示写 Addr为413569
 count是读写字数 Dataptr是发送或者接受的数据缓存区 必须使用指针 不能直接寻址



4、范例程序说明

表 10-6 案例

S7200 PLC 输入口	功 能	说 明
I0.0	写 60600008=1	设定为位置模式
I0.1	写 607A0020=10000	设定位置模式下的目标位置值
I0.2	写 60810020=1000rpm	设定位置模式下的目标速度
I0.3	写 60400010=0x4F 后 0x5F	控制命令=相对运动
I0.4	读 60630020	读电机实际位置
I0.5	读 60410010	读驱动器状态字

附录三 RS232 串口通讯范例

CM880A 与 KINCO 触摸屏自由协议通讯

注意：Kinco 全系列文本和触摸屏等都可以与 CM 步进驱动器 RS232 串口连接，本范例只介绍 Kinco 触摸屏的主打产品 MT4000, MT5000 系列与 CM 步进连接。其它系列产品等请参考各产品使用手册或咨询 Kinco 技术人员。

Kinco MT4000, MT5000 系列触摸屏可以与 CM 步进驱动器 RS232 串口连接，用户可以用触摸屏设置 CM 的内部参数和运行状态。触摸屏既可以与单个 CM 驱动器相联，也可以与多个 CM 驱动器相联。

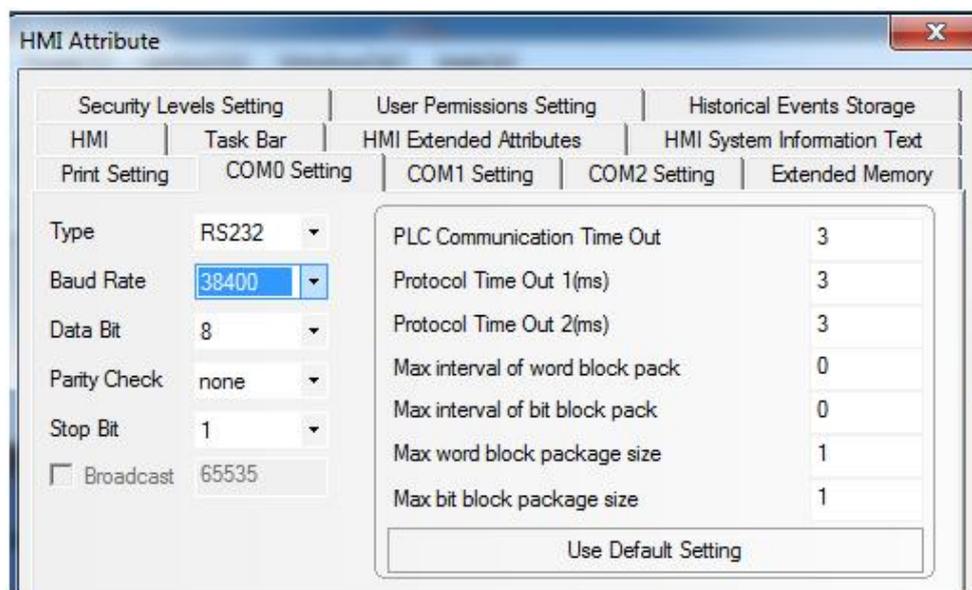
（一）触摸屏控制单台 CM 步进驱动器

a. 硬件接线



b. 通讯参数设置

触摸屏选择 Kinco stepping 驱动，触摸屏的通讯参数设置见上图，注意的是 PLC 的站号就是 CM 驱动器的 ID 号，CM 驱动器出厂时 ID 默认为 1，所以触摸屏控制单台 CM 步进驱动器时 PLC 站号设置为 1。如果 CM 的 ID 号设置为 N，那么下面的 PLC 站号也要设置为 N。CM880A 波特率，站号等参数设置参照 232 通讯协议章节。





c. 地址参数设置

编写 HMI 程序时，首先设置 CM 对象的数据长度，也就是下图中所谓地址类型，分别分为 08（8 个 Bit），10（16 个 bit），20（32 个 bit），然后再地址栏里依次设置 Index 地址，和 Subindex 地址，中间用“.” 隔开。如下图设置数值输入元件为 CM 内部对象 60FF0020（目标速度）。



CM880A 与串口调试工具自由协议通讯

RS232 通讯口通讯协议，用户可以通过 VB、VC 等软件编写程序对驱动器进行控制。下面以常见串口调试工具软件通过自由协议控制驱动器进行说明。其它软件编程可参考操作此范例！

1. 接线

PC RS232 口

驱动器 RS232（RJ45 接口）

RXD (2)	-----	TXD (3)
TXD (3)	-----	RXD (6)
GND (5)	-----	GND (4)

2. 参考串口通讯章节设定驱动器的 ID 地址和波特率，默认波特率 38400，站号 1。更改必须保存重

启有效。

3. 完成上述步骤后，就可以参照 RS232 自由通讯协议来对 CM 步进驱动器进行控制。（进行控制之前请参考第七章通讯控制注意要点）

图示例发送 6040 为 3F, 图中蓝色方框是发送数据，绿色方框是发送正确返回数据。红色部分是通讯参数

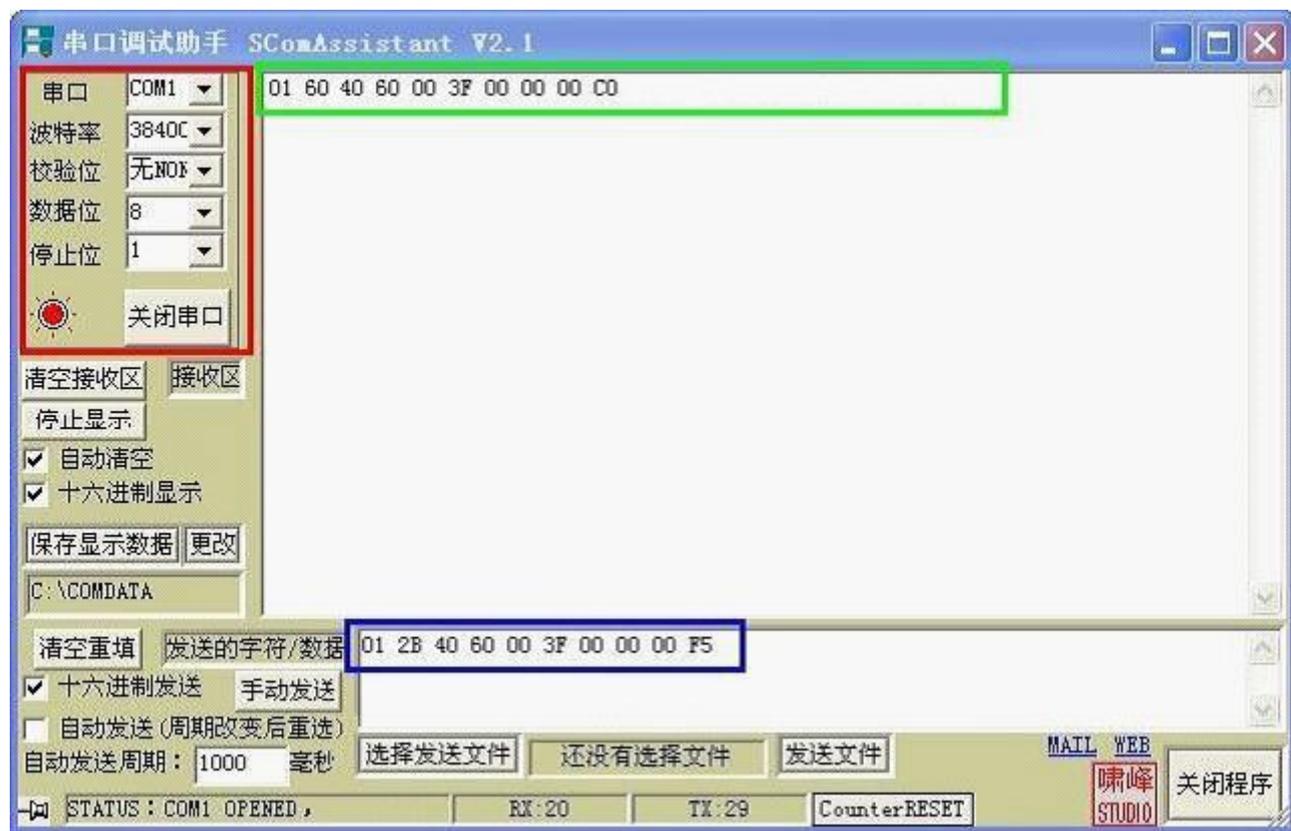


表 10-7 原点和位置控制模式发送报文

原点控制模式（控制字先 F 后 1F）				
canopen 地址	变量名称	设置值	发送及回复报文（ID=1）	备注
60400010	控制字	0xF	01 2B 40 60 00 0F 00 00 00 25 01 60 40 60 00 0F 00 00 00 F0	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875]
60600008	工作模式	0x6	01 2F 60 60 00 06 00 00 00 0A 01 60 60 60 00 06 00 00 00 D9	
60980008	原点模式	0x23	01 2F 98 60 00 23 00 00 00 B5 01 60 98 60 00 23 00 00 00 84	
60990120	原点转折信号速度	200RPM	01 23 99 60 01 00 00 32 00 30 01 60 99 60 01 00 00 32 00 73	
60990220	原点信号	150RPM	01 23 99 60 02 00 80 25 00 3C	

	速度		01 60 99 60 02 00 80 25 00 FF	
60400010	控制字	0x1F	01 2B 40 60 00 1F 00 00 00 15 01 60 40 60 00 1F 00 00 00 E0	
01 40 41 60 00 00 00 00 00 1E 读取状态字，9437 表示原点找到				
位置控制模式（控制字绝对定位先 2F 后 3F 相对定位先 4F 后 5F，103F 立即更新）				
canopen 地址	变量名称	设置值	发送及回复报文（ID=1）	备注
60400010	控制字	0xF	01 2B 40 60 00 0F 00 00 00 25 01 60 40 60 00 0F 00 00 00 F0	速度 RPM 需要转换为内部单位 DEC DEC=[(RPM*512*60000)/1875] 梯形加减速默认单位 DEC， DEC=[(RPS/S*65536*60000)/1000/4000]
60600008	工作模式	0x1	01 2F 60 60 00 01 00 00 00 0F 01 60 60 60 00 01 00 00 00 DE	
607A0020	目标位置	50000 DEC	01 23 7A 60 00 50 C3 00 00 EF 01 60 7A 60 00 50 C3 00 00 B2	
60810020	梯形速度	200RPM	01 23 81 60 00 00 00 32 00 C9 01 60 81 60 00 00 00 32 00 8C	
60830020	梯形加速度	10rps/s	使用默认值	
60840020	梯形减速度	10rps/s	使用默认值	
60400010	控制字	0x2F	01 2B 40 60 00 2F 00 00 00 05 01 60 40 60 00 2F 00 00 00 D0	
		0x3F(绝对定位)	01 2B 40 60 00 3F 00 00 00 F5 01 60 40 60 00 3F 00 00 00 C0	
		0x4F	01 2B 40 60 00 4F 00 00 00 E5 01 60 40 60 00 4F 00 00 00 B0	
		0x5F(相对定位)	01 2B 40 60 00 5F 00 00 00 D5 01 60 40 60 00 5F 00 00 00 A0	

表 10-8 速度控制模式发送报文

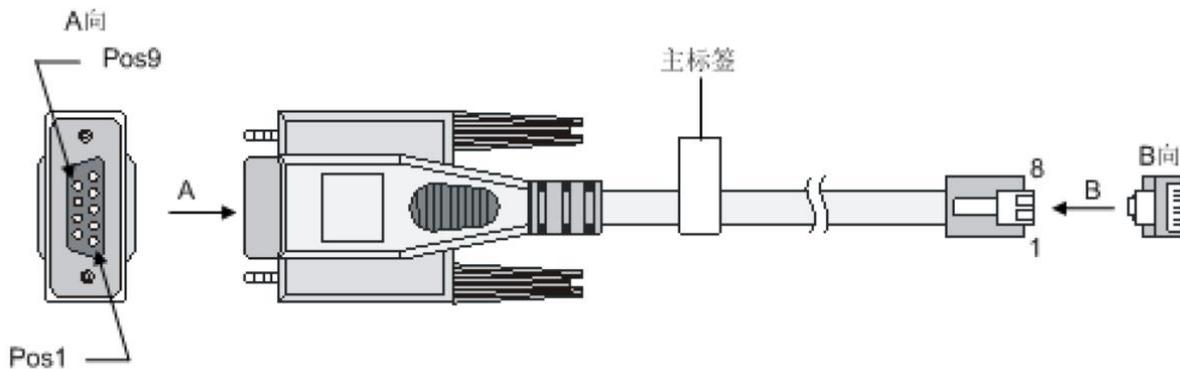
速度控制模式				
canopen 地址	变量名称	设置值	报文（ID=1）	备注

60600008	工作模式	0x3	01 2F 60 60 00 03 00 00 00 0D 01 60 60 60 00 03 00 00 00 DC	速度 RPM 需要转换为内部 单位 DEC DEC=[(RPM*512*60000)/1875] 梯形加减速默认单位 DEC, DEC=[(RPS/S*65536*60000) /1000/4000]
60FF0020	目标速度	150RPM	01 23 FF 60 00 00 80 25 00 D8 01 60 FF 60 00 00 80 25 00 9B	
60830020	梯形加速度	10rps/s	使用默认值	
60840020	梯形减速度	10rps/s	使用默认值	
60400010	控制字	0xF	01 2B 40 60 00 0F 00 00 00 25 01 60 40 60 00 0F 00 00 00 F0	

注意：通讯模式下，数据以十六进制格式传输。

Console 配置线

Console 配置线为驱动器与电脑的转接线, 一头连接电脑的 RS232 (DB9 针头接口), 另一边接驱动器的 RS232 (RJ45 接口)。实物图片如下图:



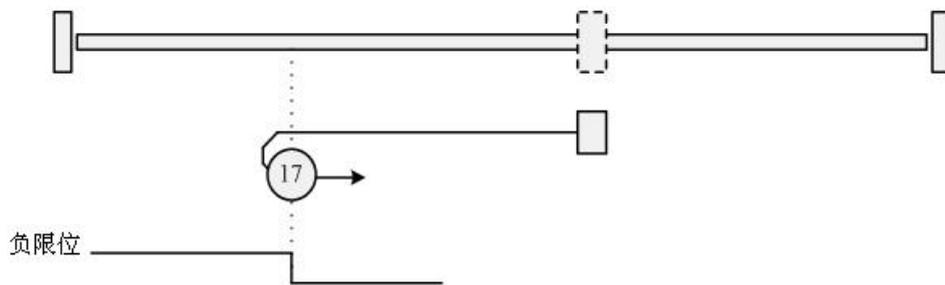
转接线的针脚连线顺序如下：

DB9 母头	RJ45 (网线口)
RXD (2)	TXD (3)
TXD (3)	RXD (6)
DTR (4)	DSR (7)
GND (5)	GND (4)&(5)
DSR (6)	DTR (2)
RTS (7)	CTS (8)
CTS (8)	RTS (1)

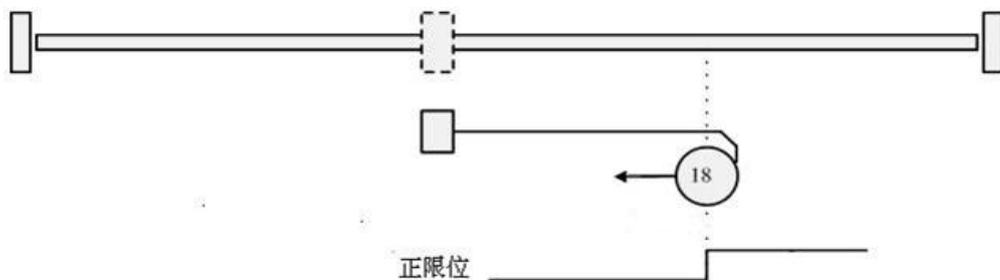
附录四 找原点方式

CM 驱动器找原点方式完全参考 CANopen 关于 DSP402 的定义。支持 17-30，和 35 原点模式

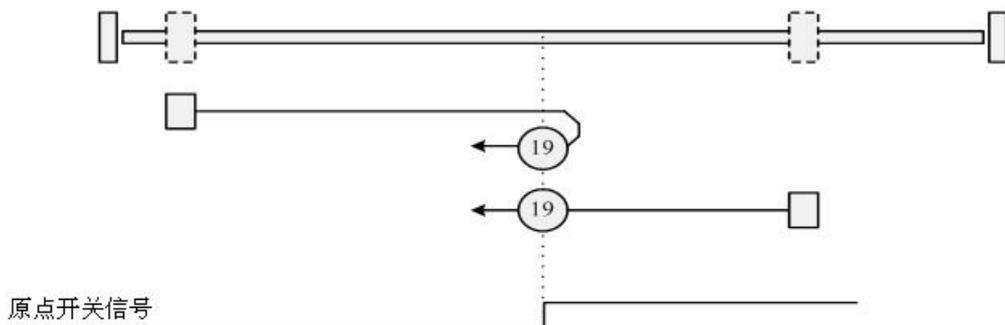
原点模式 17：以负限位为原点信号，运动轨迹如下图，遇到负限位有效信号时停止。



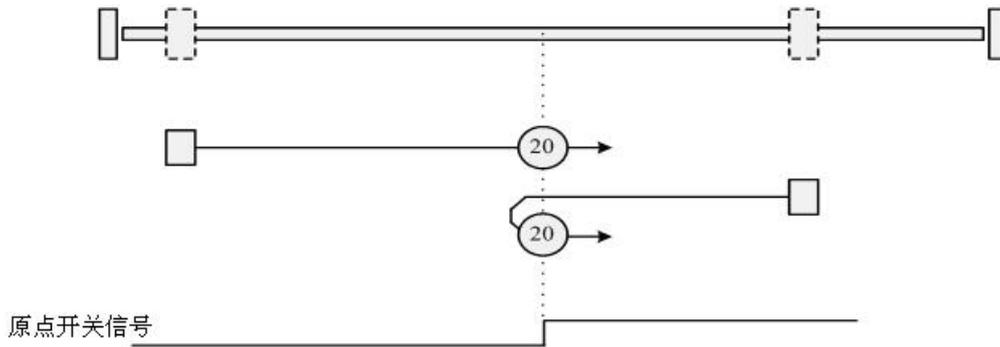
原点模式 18：以正限位为原点信号，运动轨迹如下图，遇到正限位有效信号时停止。



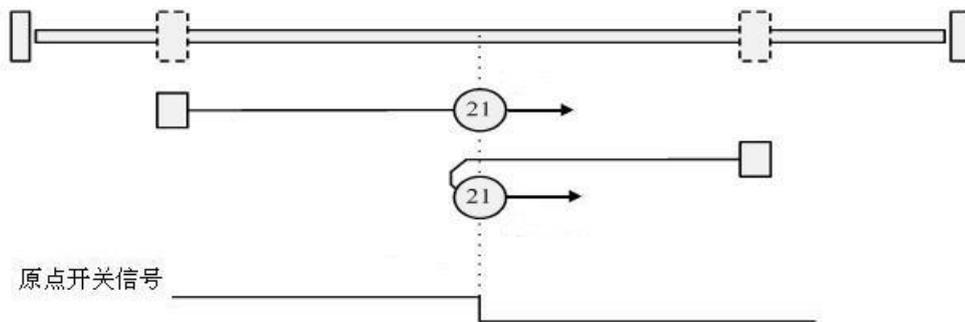
原点模式 19：以外接原点开关为原点触发信号，初始运动方向为正方向，运动轨迹如下图，遇到原点开关信号下降沿时停止。



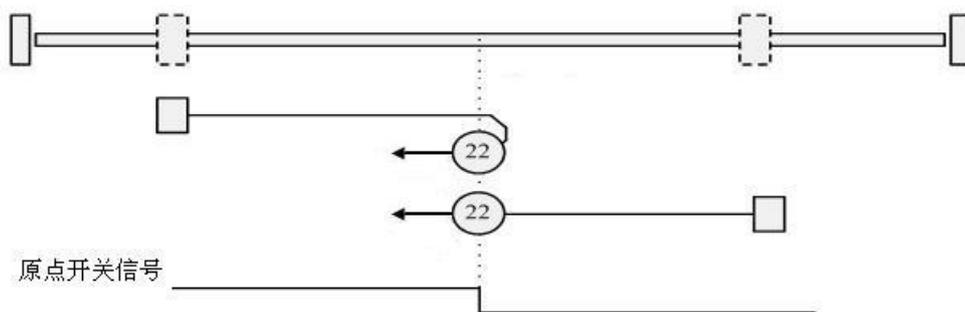
原点模式 20: 以外接原点开关为原点触发信号，初始运动方向为正方向，运动轨迹如下图，遇到原点开关信号上升沿时停止。



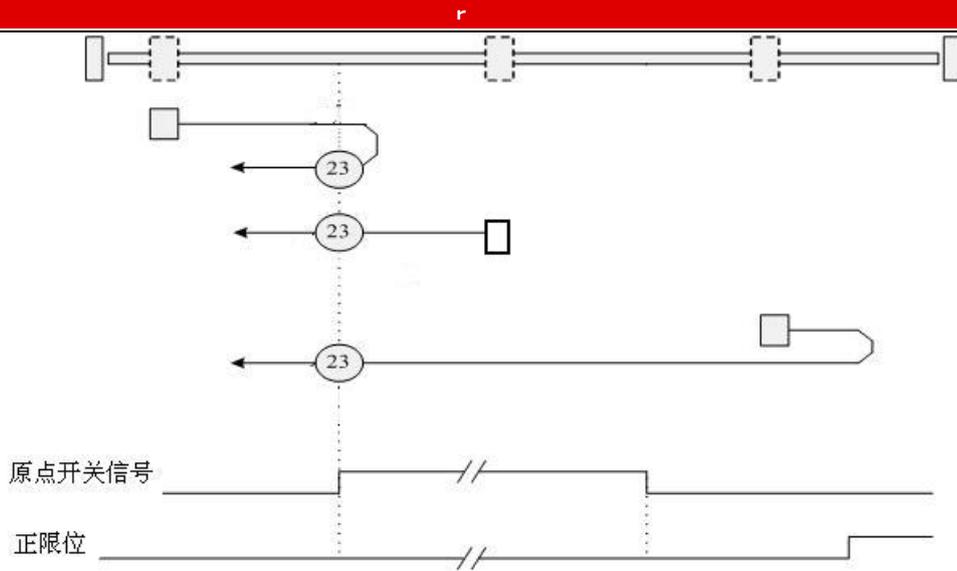
原点模式 21: 以外接原点开关为原点触发信号，初始运动方向为负方向，运动轨迹如下图，遇到原点开关信号下降沿时停止。



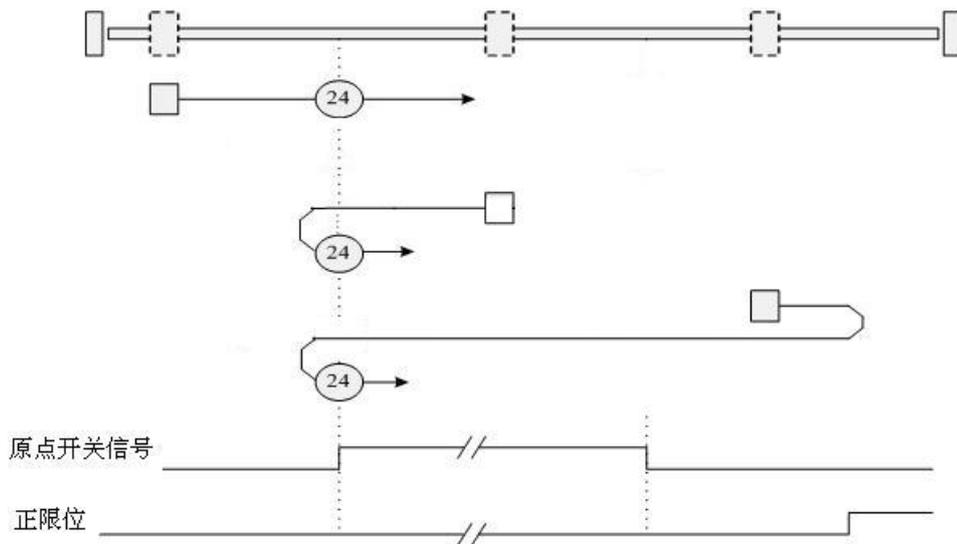
原点模式 22: 以外接原点开关为原点触发信号，初始运动方向为负方向，运动轨迹如下图，遇到原点开关信号上升沿时停止。



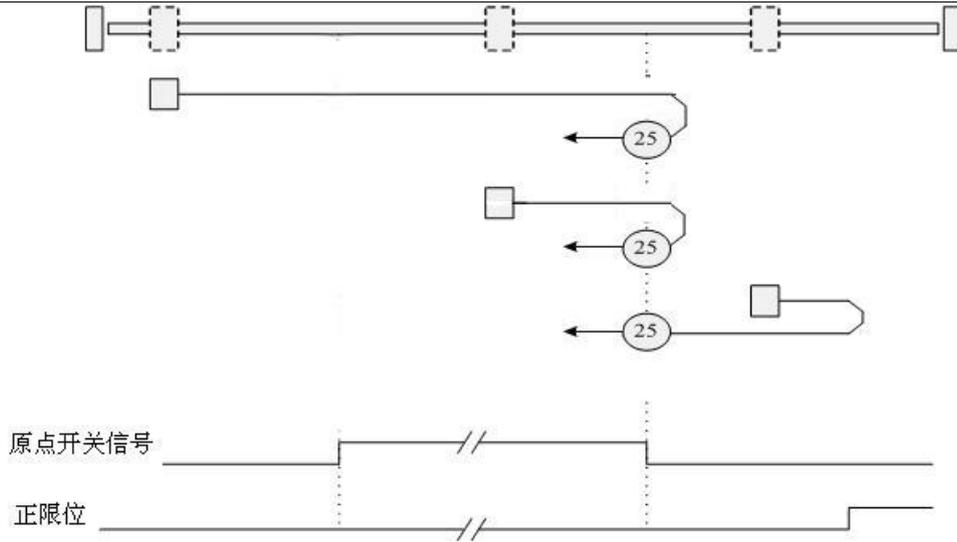
原点模式 23: 带正限位，以外接原点开关为原点触发信号，初始运动方向为正方向，运动轨迹如下图，遇到正限位有效信号时反转，遇到原点开关信号负方向下降沿时停止。



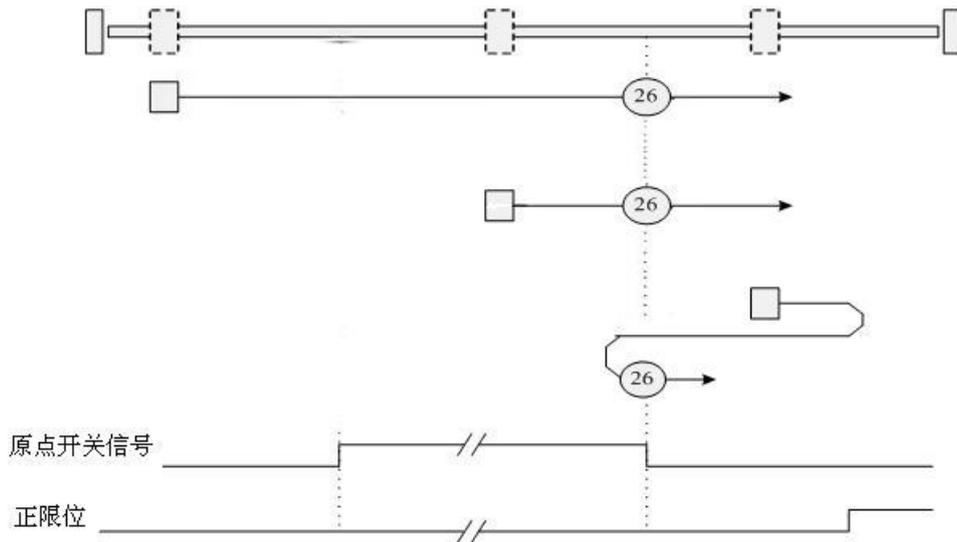
原点模式 24: 带正限位，以外接原点开关为原点触发信号，初始运动方向为正方向，运动轨迹如下图，遇到正限位有效信号时反转，遇到原点开关信号正方向上升沿时停止。



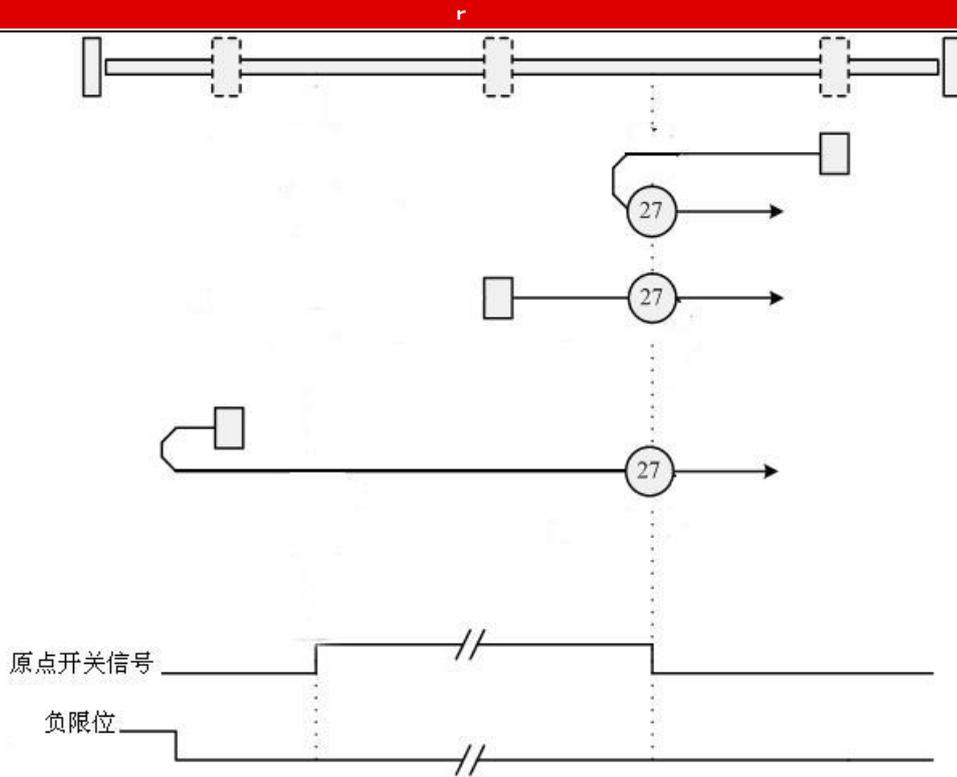
原点模式 25: 带正限位，以外接原点开关为原点触发信号，初始运动方向为正方向，运动轨迹如下图，遇到正限位有效信号时反转，遇到原点开关信号负方向上升沿时停止。



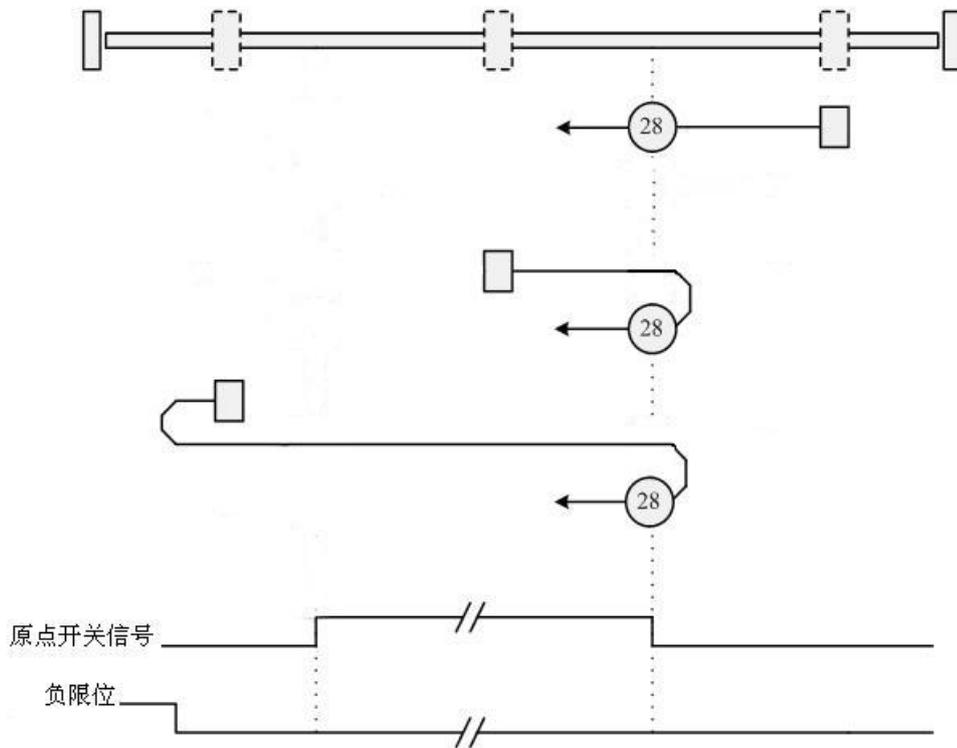
原点模式 26: 带正限位，以外接原点开关为原点触发信号，初始运动方向为正方向，运动轨迹如下图，遇到正限位有效信号时反转，遇到原点开关信号正方向下降沿时停止。



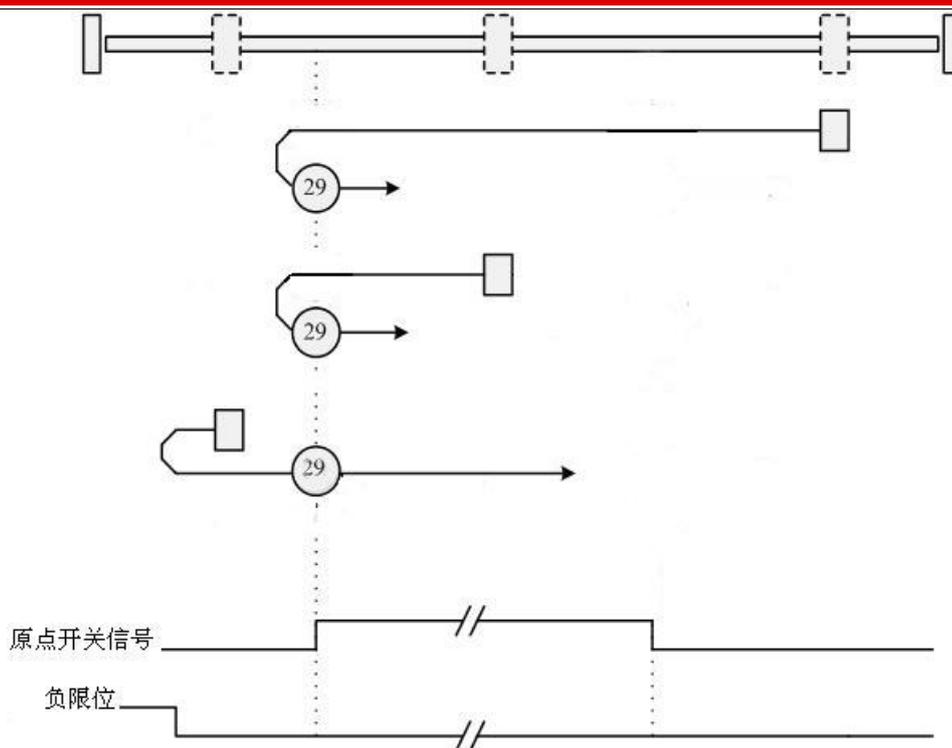
原点模式 27 带负限位，以外接原点开关为原点触发信号，初始运动方向为负方向，运动轨迹如下图，遇到负限位有效信号时反转，遇到原点开关信号正方向下降沿时停止。



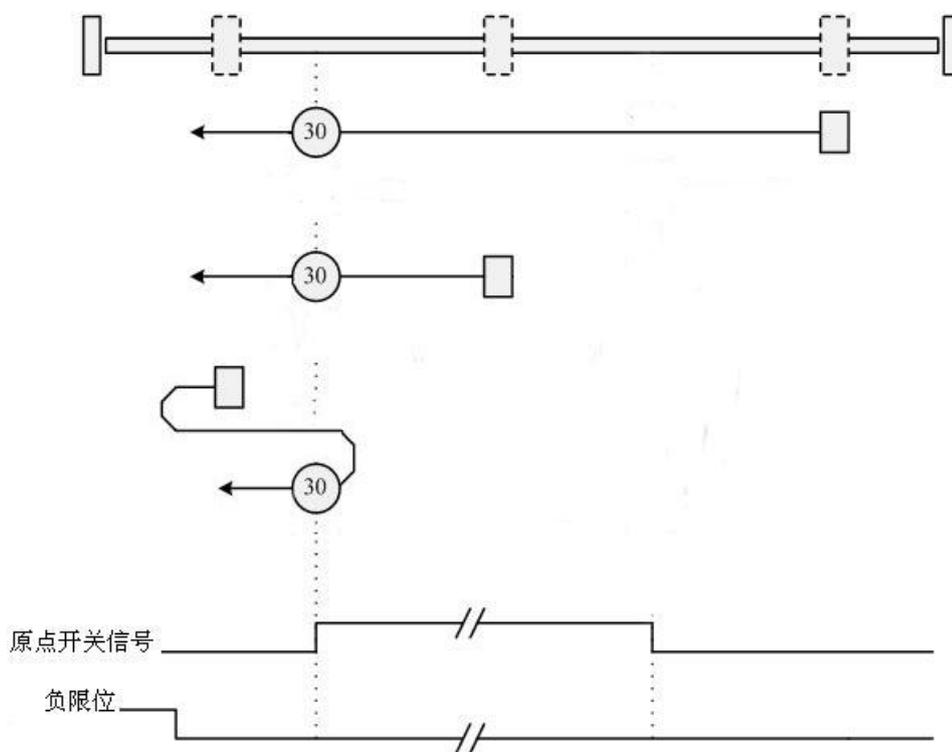
原点模式 28 带负限位，以外接原点开关为原点触发信号，初始运动方向为负方向，运动轨迹如下图，遇到负限位有效信号时反转，遇到原点开关信号负方向上升沿时停止。



原点模式 29 带负限位，以外接原点开关为原点触发信号，初始运动方向为负方向，运动轨迹如下图，遇到负限位有效信号时反转，遇到原点开关信号正方向上升沿时停止。



原点模式 30 带负限位，以外接原点开关为原点触发信号，初始运动方向为负方向，运动轨迹如下图，遇到负限位有效信号时反转，遇到原点开关信号负方向下降沿时停止。

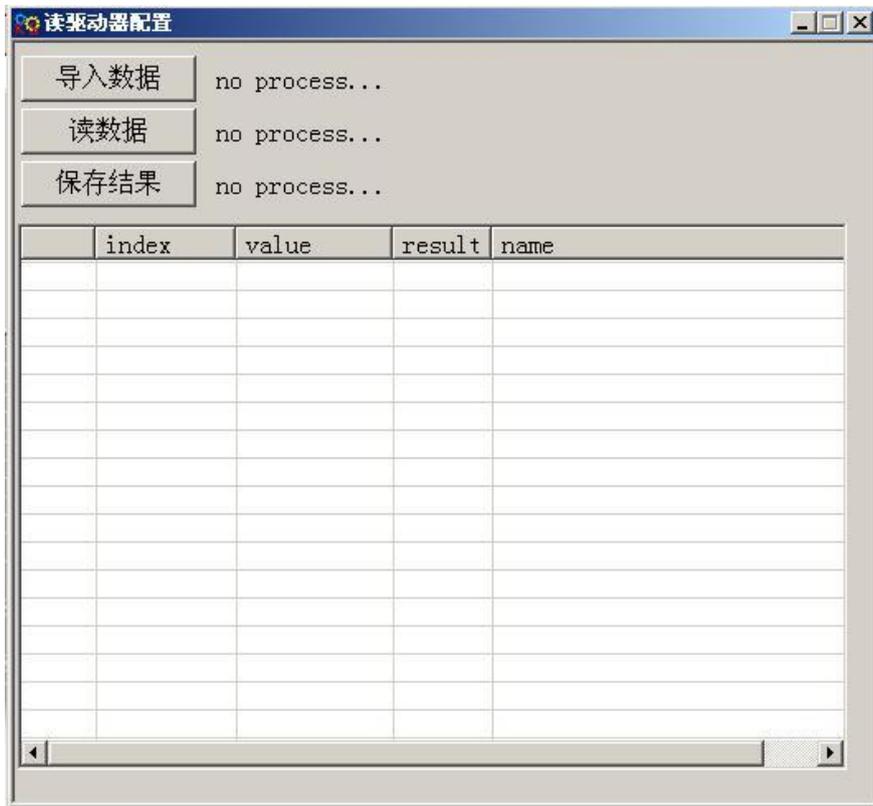


原点模式 35: 以电机当前位置为参考原点。

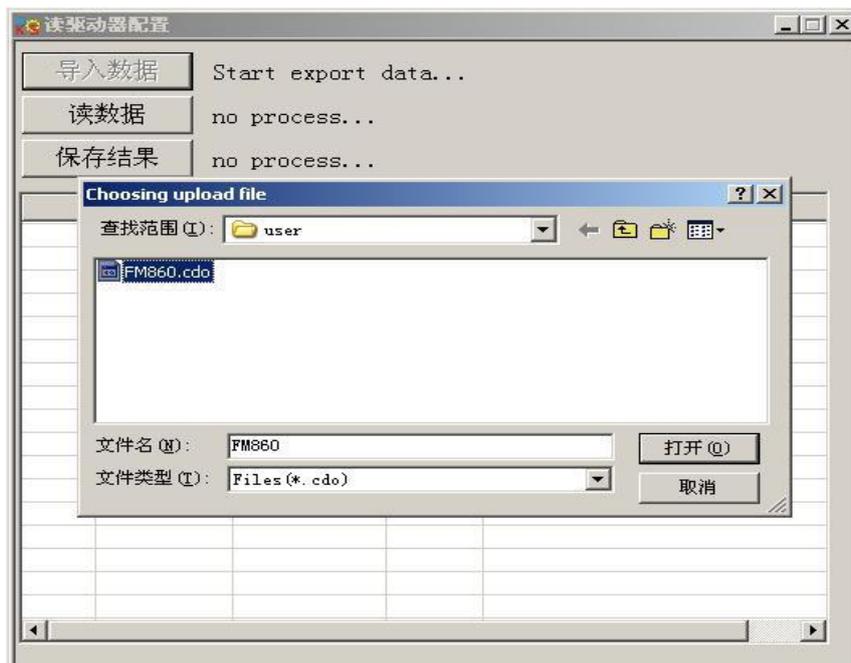
附录五 利用 Kincostep 调试软件导入和导出驱动器内参数

一台驱动器参数调试好后，可以利用 Kinco 步进上位机调试软件来导出驱动器内部参数保存在 PC 上，方便下载给其他驱动器！详细操作步骤如下

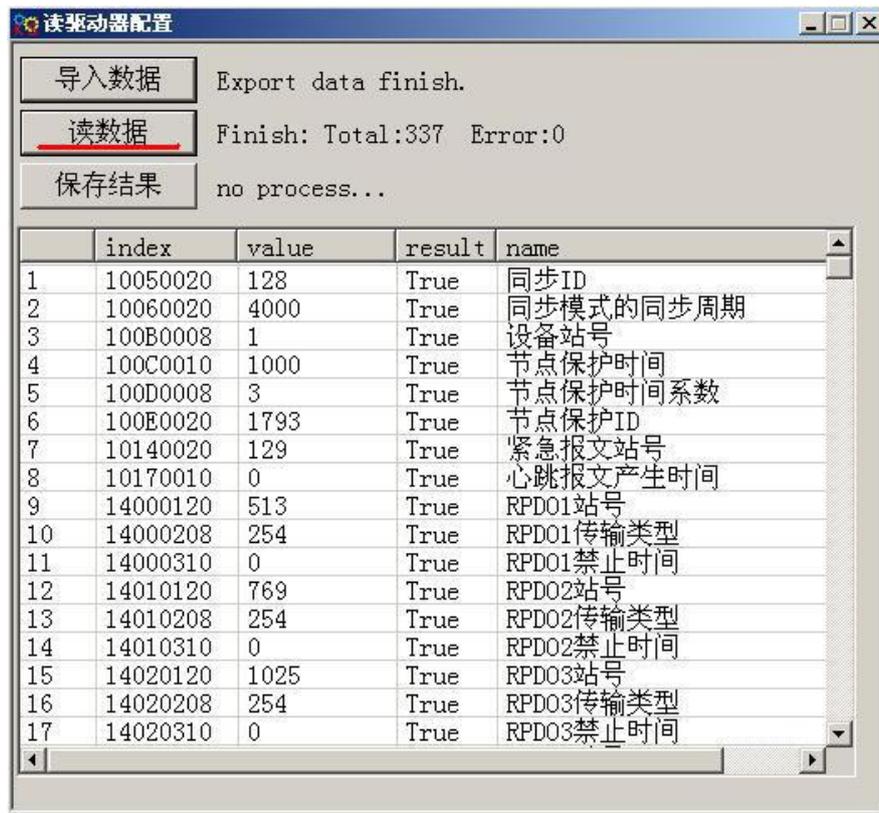
1、参照第 5 章快速入门，软件连上驱动器，点击扩展功能，点击读驱动器配置



2、点击导入数据，选择默认软件目录下的 user 文件夹下的 cdo 文件，点击打开

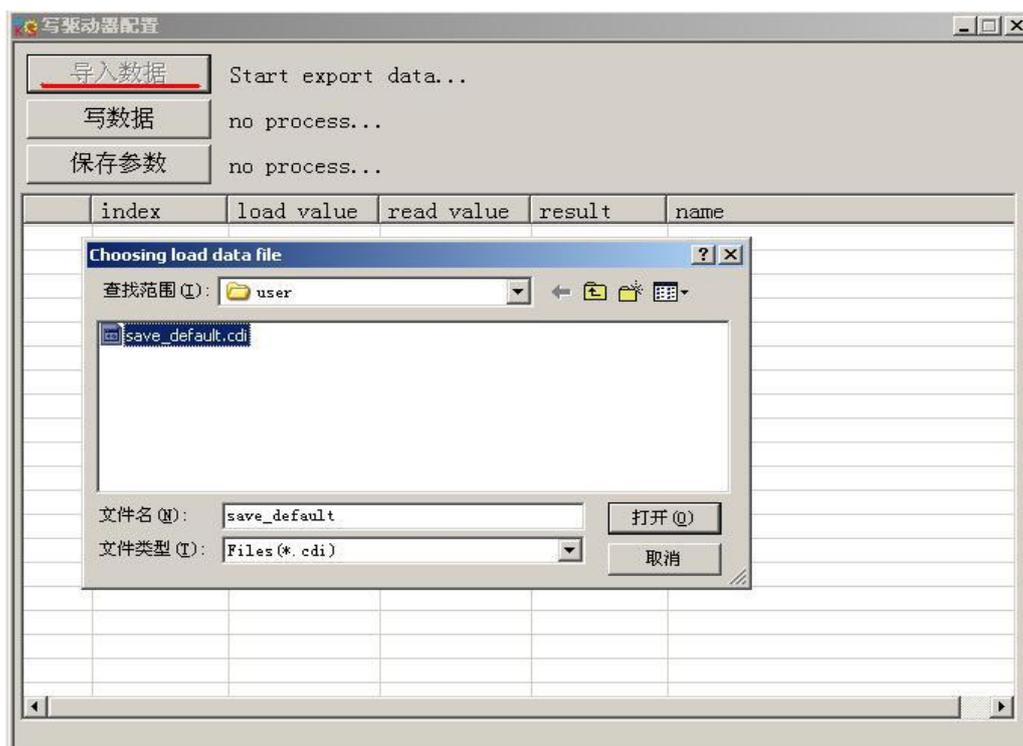


3、点击读数据，如果软件和驱动器正确连接，value 栏会读出当前驱动器所有数据

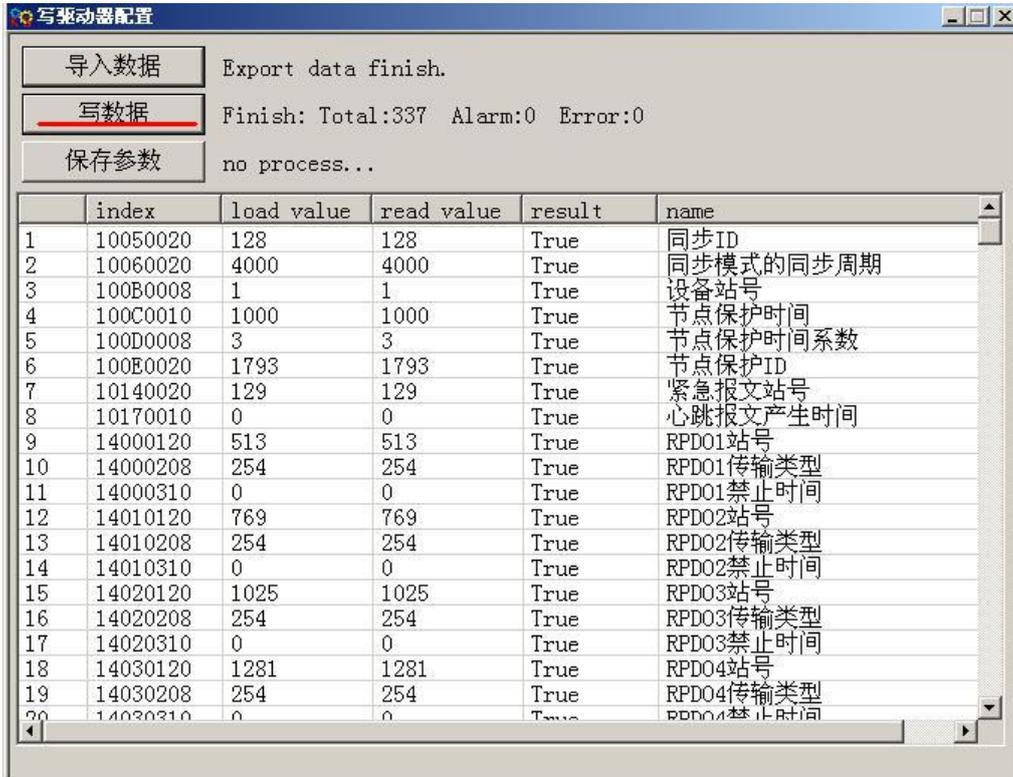


4、点击保存结果，给数据定义个名字保存到要保存的位置，这样完成读驱动器数据操作

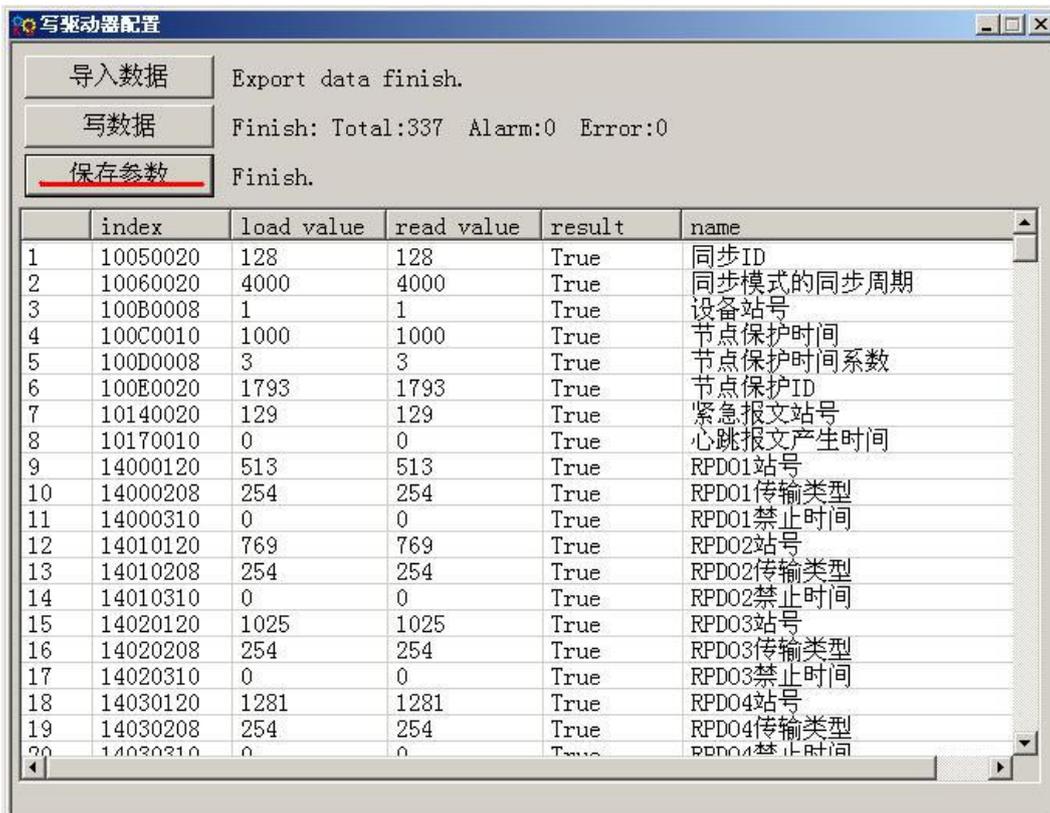
5、若要把保存的数据下载给新驱动器，软件连上驱动器，点击扩展功能，点击写驱动器配置，选择导入数据，选择保存的 cdi 文件



6、点击写数据，程序会用保存的数据（load value 栏）数据更新驱动器目前的数据（read value 栏），result 栏显示 true 表示更新成功。



7、点击保存参数，参数自动保存给驱动器，保存过程中显示 driver save，保存完毕显示 finish



附录六 常用对象工程单位与内部单位换算关系表

CM 内部对象有些参数具有工程单位和内部单位，在通讯控制时候需要注意换算，例如速度工程单位是 RPM，内部单位是 DEC，两者关系是 1RPM 约等于 16384dec（内部转一圈 60000DEC）！假设需要速度为 10rpm，那么用通讯控制时需写入速度为 163840dec！下表列出常用需要换算单位的参数表！

表 10-9 内部单位换算表

参数名称	工程单位	内部单位	换算关系
速度	RPM	DEC	$DEC = [(RPM * 512 * 60000) / 1875]$
加速度	r/s*s	DEC	$DEC = [(RPS/S * 65536 * 60000) / 4000000]$
每转脉冲个数 (细分)	step/rev	DEC	1rev=60000DEC, 如果 400step=1rev, 则 1step=60000/400=150DEC

附录七 常用对象列表

按照第 7 章介绍的数据通讯协议，所有的对象都是基于 CANopen 数据格式建立。在下面文档中数据采用了 16 进制方式表达，CANopen 地址由 Index+ Subindex 组成。用 Index(16 位地址)、Subindex (8 位子地址) 形式表示寄存器寻址，位数 08 表示此寄存器将存放的数据长度为 1 个 Byte, 位数 10 表示存放的数据长度为 2 个 Byte, 位数 20 表示存放的数据长度为 4 个 Byte, 访问此寄存器时需注意它的读写属性，读或写标识(RW)，只读或只写标识(RO, WO)，映射标识(M)。

表 10-10 模式及控制

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
6040+00	10	0x3100	RW	Bit	bit0: 上电 (Switch on) bit1: 禁止输出电压 (Disable voltage) bit2: 快速停止 (Quick stop) bit3: 操作 (Enable operation) bit4: 新设定值/开始找原点 (New setpoint/Start homing) bit5: 立即更改设置 (Change set immediately) bit6: 绝对或相对 (Absolute or relative) bit7: 故障复位 (Fault reset) bit8: 保留 bit9: 保留 bit10: 保留 bit11: 制造商特定1 (manufacturer specific 1) bit12: 制造商特定2 (manufacturer specific 2) bit13: 制造商特定3 (manufacturer specific 3) bit14: 制造商特定4 (manufacturer specific 4) bit15: 制造商特定5 (manufacturer specific 5) 用控制字节改变驱动器的状态 => 机器状态 0x06 电机断电

					0x0F 电机上电 0x0B 快速停止, 负载停止-电压断开 0x2F-3F 进入绝对定位方式 0x4F-5F 进入相对定位方式 0x103F 根据目标位置变化立即进入绝对定位 0x0F-1F 原点定位 0x80 清除内部故障
6041+00	10	0x3200	RO	Bit	状态字节显示驱动器的状态 bit0: 准备上电 (ready to switch on) bit1: 已上电 (switch on) bit2: 使能 (operation enable) bit3: 故障 (fault) bit4: 禁止输出电压 (Voltage Disable) bit5: 快速停止 (Quick Stop) bit6: 上电禁止 (switch on disable) bit7: 警告 (warning) bit8: 内部保留 bit9: 保留 bit10: 目标位置到 (target reach) bit11: 内部限位激活 (internal limit active) bit12: 脉冲响 (Step. Ach. /V=0/Hom. att.) bit13: 跟随误差/原点错 (Foll. Err/Res. Hom. Err.) bit14: 找到电机励磁 (Commutation Found) bit15: 原点找到 (Referene Found)
6060+00	08	0x3500	RW	DEC	工作模式: 1---位置模式 3---速度模式 -4—脉冲模式 6---回原点模式
6061+00	08	0x3600	RO	DEC	有效工作模式

表 10-11 测量数据

canopen 地址	位数	modbus 地址	命令类型	单 位	详细解释
6063+00	20	0x3700	RO	DEC	实际位置值
606C+00	10	0x3B00	RO	RPM	实际速度 (rpm) 内部采样时间为10mS
6078+00	10	0x3E00	RO	1Ap=1.414*Arms	实际电流值 CM880A, CM880B, CM880C: 1 Arms =79dec CM422B, CM422C: 1 Arms =238dec
60FD+00	20	0x6D00	RO	Bit	输入口状态 bit0: 负限位信号状态 bit1: 正限位信号状态 bit2: 原点信号状态

6079+00	10	0x3F00	RO	V	实际母线电压
---------	----	--------	----	---	--------

表 10-12 目标对象

Canopen地址	位数	modbus地址	命令类型	单 位	详细解释
607A+00	20	0x4000	RW	DEC	工作模式1下的目标位置,如果控制字设定为开始运动,转变成为有效指令位置。
6081+00	20	0x4A00	RW	$DEC=[(RPM*512*60000)/1875]$	工作模式1时的梯形曲线的最大速度。
6083+00	20	0x4B00	RW	$DEC=[(RPS/S*65536*60000)/400000]$	梯形曲线的加速度,默认值: 10rps/s
6084+00	20	0x4C00	RW		梯形曲线的减速度,默认值: 10rps/s
6085+00	20	0x3300	RW		紧急停止的减速度,默认值: 100rps/s
60FF+00	20	0x6F00	RW	$DEC=[(RPM*512*60000)/1875]$	在模式3时的目标速度。
6073+00	10	0x3D00	RW	$1 A_p=1.414*Arms$	目标电流 CM880A, CM880B, CM880C: 1 Arms =79dec CM422B, CM422C:1 Arms =238dec
6080+00	10	0x4900	RW	RPM	最大速度限制
605A+00	08	0x3400	RW	DEC	紧急停止模式

表 10-13 多段位置, 多段速控制: (速度单位换算 $DEC=[(RPM*512*60000)/1875]$)

Canopen地址	位数	modbus地址	命令类型	单 位	详细解释
2020+01	20	0x0C10	RW	DEC	多段位置控制0
2020+02	20	0x0C20	RW	DEC	多段位置控制1
2020+03	20	0x0C30	RW	DEC	多段位置控制2
2020+04	20	0x0C40	RW	DEC	多段位置控制3
2020+05	20	0x0C50	RW	DEC	多段位置控制 4
2020+06	20	0x0C60	RW	DEC	多段位置控制 5
2020+07	20	0x0C70	RW	DEC	多段位置控制 6
2020+08	20	0x0C80	RW	DEC	多段位置控制 7
2020+09	20	0x0C90	RW	DEC	多段位置控制8
2020+0A	20	0x0CA0	RW	DEC	多段位置控制9
2020+0B	20	0x0CB0	RW	DEC	多段位置控制10
2020+0C	20	0x0CC0	RW	DEC	多段位置控制11

2020+0D	20	0x0CD0	RW	DEC	多段位置控制 12
2020+0E	20	0x0CE0	RW	DEC	多段位置控制 13
2020+0F	20	0x0CF0	RW	DEC	多段位置控制 14
2020+10	20	0x0D00	RW	DEC	多段位置控制 15
2020+11	20	0x0D10	RW	DEC	多段速度控制 0
2020+12	20	0x0D20	RW	DEC	多段速度控制 1
2020+13	20	0x0D30	RW	DEC	多段速度控制 2
2020+14	20	0x0D40	RW	DEC	多段速度控制 3
2020+15	20	0x0D50	RW	DEC	多段速度控制 4
2020+16	20	0x0D60	RW	DEC	多段速度控制 5
2020+17	20	0x0D70	RW	DEC	多段速度控制 6
2020+18	20	0x0D80	RW	DEC	多段速度控制 7
2020+19	20	0x0D90	RW	DEC	多段速度控制 8
2020+1A	20	0x0DA0	RW	DEC	多段速度控制 9
2020+1B	20	0x0DB0	RW	DEC	多段速度控制 10
2020+1C	20	0x0DC0	RW	DEC	多段速度控制 11
2020+1D	20	0x0DD0	RW	DEC	多段速度控制 12
2020+1E	20	0x0DE0	RW	DEC	多段速度控制 13
2020+1F	20	0x0DF0	RW	DEC	多段速度控制 14
2020+20	20	0x0E00	RW	DEC	多段速度控制 15
2020+36	08	0x0F60	RW	DEC	多段位置控制选择显示
2020+37	08	0x0F70	RW	DEC	多段速度控制选择显示
2020+38	10	0x0F80	RW	ms	多段速度/位置切换延时

表 10-14 性能对象

canopen地址	位数	modbus地址	命令类型	单位	详细解释
6065+00	20	0x3800	RW	DEC	设置为报警的最大位置跟随误差值 默认值 200DEC
6067+00	20	0x3900	RW	DEC	位置到窗口 “目标位置到达”的位置范围 默认值 10DEC
607D+01	20	0x4410	RW	DEC	软限位正设置

607D+02	20	0x4420	RW	DEC	软限位负设置
---------	----	--------	----	-----	--------

表 10-15 原点控制

canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
6098+00	08	0x4D00	RW	DEC	寻找原点模式 详见原点控制模式章节
6099+01	20	0x5010	RW	DEC=[(RPM*512 *60000)/1875]	寻找极限开关的速度
6099+02	20	0x5020	RW		寻找原点信号的速度
609A+00	20	0x5200	RW	DEC=[(RPS/S*65536 *60000)/4000000]	寻找原点时加速度
607C+00	20	0x4100	RW	DEC	原点偏移
6099+04	10	0x5040	RW	1 Arms=1.414*Arms	寻找原点时电流 CM880A, CM880B, CM880C: 1 Arms =79dec CM422B, CM422C: 1 Arms =238dec

表 10-16 输入输出参数

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
2010+03	20	0x0830	RW	Bit	数字输入 1 功能定义
2010+04	20	0x0840	RW	Bit	数字输入 2 功能定义
2010+05	20	0x0850	RW	Bit	数字输入 3 功能定义
2010+06	20	0x0860	RW	Bit	数字输入 4 功能定义
2010+07	20	0x0870	RW	Bit	数字输入 5 功能定义
2010+08	20	0x0880	RW	Bit	数字输入 6 功能定义
2010+10	20	0x0900	RW	Bit	数字输出 1 功能定义
2010+11	20	0x0910	RW	Bit	数字输出 2 功能定义
2010+12	20	0x0920	RW	Bit	数字输出 3 功能定义
2010+0B	10	0x08B0	RO	Bit	输入口状态 bit0: Din1 bit1: Din2 bit2: Din3 bit3: Din4 bit4: Din5 bit5: Din6
2010+18	10	0x0980	RO	Bit	输出口状态 bit0: Dout1 bit1: Dout2 bit2: Dout3
2010+01	10	0x0810	RW	Bit	输入信号极性定义

					0: 常闭; 1: 常开 bit0: Din1 bit1: Din2 bit2: Din3 bit3: Din4 bit4: Din5 bit5: Din6 默认值 FF
2010+0E	10	0x08E0	RW	Bit	输出口极性定义 0: 常闭; 1: 常开 bit0: Dout1 bit1: Dout2 bit2: Dout3 默认值 FF
2010+02	10	0x0820	RW	Bit	输入口信号模拟 bit0: Din1 bit1: Din2 bit2: Din3 bit3: Din4 bit4: Din5 bit5: Din6
2010+0F	10	0x08F0	RW	Bit	输出口信号模拟 bit0: Dout1 bit1: Dout2 bit2: Dout3
2020+31	08	0x0F10	RW	DEC	输入工作模式控制0 无输入时模式
2020+32	08	0x0F20	RW	DEC	输入工作模式控制1 有输入时模式
2020+33	10	0x0F30	RW	DEC	输入控制字

表 10-17 脉冲输入参数

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
6410+18	10	0x7180	RW	DEC	每转脉冲个数
2508+03	08	0x1930	RO	DEC	脉冲模式控制 0... 双脉冲模式 1... 脉冲方向模式 2... 增量式编码器模式 注: 通过配置I0来选择脉冲模式。
2508+04	20	0x1940	RW	DEC	电子齿轮前输入脉冲数
2508+05	20	0x1950	RW	DEC	电子齿轮后输入脉冲数
2508+06	10	0x1960	RW	DEC	脉冲平滑滤波系数滤 波频率 $f=1000/(2\pi * Pulse_Filter)$

					时间常数 $\tau = \text{Pulse_Filter}/1000$ (S)
2508+0C	10	0x19C0	RW	kHz	电子齿轮前脉冲频率
2508+0D	10	0x19D0	RW	kHz	电子齿轮后脉冲频率
6410+18	10	0x7180	RW	Step/rev	电机每转脉冲数

表 10-18 电机参数:

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
6410+01	10	0x7010	RW	HEX	电机型号选择 ASCII....HEX....型号 "00".....3030...无电机型号 "MC".....434d...自检测电机参数 "XX".....5858...自定义电机参数 "A1".....3141...2S42Q-03848 "A2".....3241...2S42Q-02940 "A3".....3341...2S42Q-0240 "A4".....3441...2S42Q-0348(串联接法) "A5".....3541...2S42Q-0348(并联接法) "B1".....3142...2S56Q-030B5 "B2".....3242...2S56Q-02976 "B3".....3342...2S56Q-02054 "B4".....3442...2S56Q-02741 "B5".....3542...2S56Q-0541(串联接法) "B6".....3642...2S56Q-0541(并联接法) "B7".....3742...2S56Q-0956(串联接法) "B8".....3842...2S56Q-0956(并联接法) "B9".....3942...2S56Q-1376(串联接法) "BA".....4142...2S56Q-1376(并联接法) "BB".....4242...2S56Q-2280(串联接法) "BC".....4342...2S56Q-2280(并联接法) "BD".....4442...2S56Q-25B2(串联接法) "BE".....4542...2S56Q-25B2(并联接法) "C1".....3143...2S86Q-069B8 "C2".....3243...2S86Q-05180 "C3".....3343...2S86Q-03865 "C4".....3443...2S86Q-051F6 "C5".....3543...2S86Q-030B8...2S86Q-85B8 "C6".....3643...2S86Q-03080...2S86Q-4580 "C7".....3743...2S86Q-01865...2S86Q-3465 "D1".....3144...2S110Q-054K1 "D2".....3244...2S110Q-047F0 "D3".....3344...2S110Q-03999 "E1".....3145...2S130Y-063R8

					"E2".....3245...2S130Y-039M0 "F1".....3146...3S57Q-04079 "F2".....3246...3S57Q-04056 "F3".....3346...3S57Q-04042 "G1".....3147...3S85Q-04097 "G2".....3247...3S85Q-04067 "G3".....3347...3S85Q-040F7
6410+05	08	0x7050	RW	2p/r	电机极对数,默认是50
6410+0B	10	0x70B0	RW	1Arms=10dec	电机相电流
6510+04	10	0x8040	RW	1Arms=10dec	电机相电流限制
6410+0C	10	0x70C0	RW	1mH=10dec	电机相电感
6410+0D	10	0x70D0	RW	1Ω=100dec	电机相电阻
6410+13	08	0x7130	RW	DEC	电机旋转方向 0: 顺时针 1: 逆时针
6410+16	10	0x7160	RO	HEX	当前使用电机型号
6410+19	08	0x7190	RW	DEC	电机参数上电自检测开关 0: 禁止 1: 使能
6410+1A	08	0x71A0	RW	DEC	电机相数 2: 二相步进电机 3: 三相步进电机
6410+1B	08	0x71B0	RW	DEC	IO口5V电压输出开关 0: 禁止 1: 使能

表 10-19 用于存储的参数:

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
2FF0+01	08	0x2910	RW	DEC	存储控制环参数 1: 存储设定的所有配置参数 10: 初始化所有的配置参数 注: 存储控制环参数, 不包括电机参数。
2FF0+03	08	0x2930	RW	DEC	存储电机的参数 1: 存储设定的所有电机参数

表10-20 错误代码:

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
2601+00	10	0x1F00	RO	Bit	实时报警错误状态低16位 错误状态字低16位 bit0: 内部错误报警 bit1: 过流报警

					bit2: 高压报警 bit3: 低压报警 bit4: 温度报警 bit5: 内部逻辑电压异常 bit6: 5V输出电流过载 bit7: EEPROM内部错误 bit8: 寻找电机错误 bit9: 跟随误差报警 bit10: 总线错误 bit11: 输入脉冲频率过高 bit12: 外部预使能信号 bit13: 正限位报警 bit14: 负限位报警 bit15: 拨码开关为NA状态
2602+00	10	0x2000	R0	Bit	实时报警错误状态高16位 位0: -- 位15:
2603+00	10	0x2100	R0	Bit	上电自检检测错误状态低16位 bit0: A相过流故障 bit1: B相过流故障 bit2: 高压报警 bit3: 低压报警 bit4: 供电电压过低 bit5: 温度报警 bit6: A相电流电路故障 bit7: B相电流电路故障 bit8: A相过流电路故障 bit9: B相过流电路故障 bit10: A相功率电路故障 bit11: B相功率电路故障 bit12: 电机相间接错线 bit13: 电机A相接错线 bit14: 电机B相接错线 bit15: 电机A相未接线
2604+00	10	0x2200	R0	Bit	上电自检检测错误状态高16位 bit16: 电机B相未接线 bit17: 模拟信号输入电路故障 bit18: 逻辑15V电压故障 bit19: 逻辑5V电压故障 bit20: 输出5V过载 bit21: SN号错误 bit22: 存储器读写失败 bit23: 存储器读写超时 bit24: 存储器读数据校检出错 bit25: 电机配置型号错误 bit26: 保留 bit27: 看狗复位 bit28: 拨码开关为NA状态

					bit29: 程序与PCB不匹配 bit30: 晶振电路故障 bit31: ADC转换电路故障
2610+00	08	无	R0	Bit	驱动器错误组0
2611+00	08	无	R0	Bit	驱动器错误组 1
2612+00	08	无	R0	Bit	驱动器错误组 2
2613+00	08	无	R0	Bit	驱动器错误组 3
2614+00	08	无	R0	Bit	驱动器错误组 4
2615+00	08	无	R0	Bit	驱动器错误组 5
2616+00	08	无	R0	Bit	驱动器错误组 6
2617+00	08	无	R0	Bit	驱动器错误组 7

表 10-21 总线特性参数:

Canopen 地址	位数	modbus 地址	命令类型	单位	详细解释
100B+00	08	0x0600	RW	DEC	设备驱动器当前站号。 注: 1) ID 站号可以由 SW6-SW1 设置 1~63 或由 0x2FE400 设置 1~127。 2) 更改该参数需要存储再重新启动。
2FE4+00	08	0x2800	RW	DEC	设备驱动器站号1~127; 注: 仅SW6-SW1都为“OFF”时, 设置才有效。当更改该参数需要存储再重新启动。
2F81+00	08	0x2300	RW	DEC	CAN波特率设置 默认值 50 设定值 波特率 100: 1M 50: 500k 25: 250k 12: 125k 5: 50k 需要保存重新启动
2FE0+00	10	0x2400	RW	DEC	RS232 波特率设置 默认值 259 设定值 波特率 2082 4800 1041 9600 520 19200 259 38400 86 115200 需要重新启动
2FE2+00	10	0x2600	RW	DEC	RS485 波特率设置 默认值 520 设定值 波特率 1041 9600 520 19200

					259 38400 86 115200 需要重新启动
--	--	--	--	--	----------------------------------



上海步科自动化股份有限公司

Shanghai Kinco Automation Co., Ltd.

地 址：深圳市南山区高新科技园北区朗山一路6号意中利工业园1号厂房1-3层（518057）

电 话：86-755-26585555

传 真：86-755-26616372

<http://www.kinco.cn>

Email: sales@kinco.cn