

目 录

第一章 功能及构成.....	7
1.1 功能.....	7
1.2 控制方式.....	7
1.3 构成.....	8
第二章 安装与使用事项.....	9
2.1 使用环境.....	9
2.2 安装间距和方向.....	10
2.3 防止异物进入.....	10
2.4 电缆安装.....	10
2.5 使用事项.....	10
2.5.1 使用原则.....	10
2.5.2 使用环境.....	11
2.5.3 安装要求.....	11
2.5.4 电气连接前要求.....	11
2.5.5 操作.....	12
第三章 接口信号与接线.....	13
3.1 ED 系列驱动器型号.....	13
3.2 ED 系列驱动器接口分布.....	16
3.2.1 ED100 系列驱动器接口分布.....	16
3.2.2 ED430、ED620/ED630 系列驱动器接口分布.....	17
3.2.3 ED 系列驱动器接口功能介绍.....	18
3.3 驱动器接口电路.....	21
3.3.1 ED100 驱动器接口电路.....	21
3.3.2 ED430 ED620 ED630 驱动器接口电路.....	23
3.4 接线.....	25
3.4.1 ED430 系列驱动器电源接线.....	25
3.4.2 ED620/ED630 系列驱动器电源接线.....	26
3.4.3 EMC 安装.....	27

3.4.4 ED 驱动器和 PLC 接口接线.....	28
3.4.5 ED 驱动器和电机之间连线.....	29
第四章 ECO2WIN 软件使用说明.....	30
4.1 软件安装.....	30
4.2 快速入门.....	30
4.2.1 ED 运行所需的最低硬件要求.....	30
4.2.2 建立一个新的工程文件.....	31
4.2.3 打开一个工程文件.....	39
4.2.4 主窗体介绍.....	39
4.2.5 导入和导出系统文件.....	45
第五章 ECO2WIN 基本功能介绍.....	46
5.1 对象词典 (Object dictionary)	46
5.2 对象操作表 (Direct object entry)	47
5.3 设备控制 (Device control)	47
5.3.1 工作模式 (Operation mode)	48
5.3.2 控制命令和状态显示.....	52
5.3.3 设置运动参数.....	54
5.3.4 原点控制.....	55
5.3.5 自动正反转控制.....	55
5.3.6 错误诊断.....	56
5.4 参数设置 (Parameter setting)	57
5.4.1 位置环参数.....	58
5.4.2 速度环参数.....	59
5.4.3 电流环参数.....	60
5.4.4 励磁参数 (Commutation)	61
5.4.5 其它参数.....	63
5.5 内部编程.....	64
5.5.1 顺序编程(Sequences).....	64
5.5.2 数字输入(Digital input).....	70
5.5.3 比较器 (Comparator)	74
5.5.4 定时器、控制器内部事件 (Timer/Controller events)	76

5.5.5 计数器功能 (Counter)	79
5.5.6 计算器(Calculator).....	81
5.5.7 记录表格 (Recording table)	82
5.5.8 快速位置捕捉 (Position capture)	83
5.6 设备配置 (Device configuration)	84
5.6.1 数字输入 (Digital input)	84
5.6.2 数字输出 (Digital output)	86
5.6.3 电子齿轮比设置 (Electronic gear)	88
5.6.4 模拟量输出监控 (Analog monitor)	90
5.6.5 模拟量输入 (Analog input)	91
5.7 通讯参数设置(Communication).....	92
5.7.1 Tx-PDO 映射地址设置.....	94
5.7.2 Rx-PDO 映射地址设置.....	96
5.8 内部示波器 (Oscilloscope)	97
5.9 系统管理 (Administration)	98
第六章 电机试运转.....	100
6.1 电气元件的安装.....	100
6.1.1 连接编码器电缆到驱动器.....	100
6.1.2 连接电机电缆(带或不带抱闸)到电机电缆接驱动器相应接口.....	100
6.1.3 给驱动器提供逻辑电源 (控制器的电源 X4)	101
6.1.4 硬件复位.....	101
6.1.5 编码器信号检查.....	101
6.1.6 连接外部使能信号.....	102
6.1.7 给驱动器提供动力电源.....	102
6.1.8 连接编程电缆.....	102
6.2 电机试运转.....	102
6.2.1 电机参数配置.....	102
6.2.2 电流调整.....	104
6.2.3 检查速度环和位置环参数.....	105
6.2.4 状态控制.....	106
第七章 限位开关与原点控制.....	108

7.1 限位开关和原点开关.....	108
7.2 原点查找原理.....	109
7.3 原点模式.....	110
7.3.1 原点模式 1：参考负限位开关信号的原点模式.....	110
7.3.2 原点模式 2：参考正限位开关信号的原点模式.....	110
7.3.3 原点模式 3 和 4：参考正向原点开关和 index 信号的原点模式.....	111
7.3.4 原点模式 5 和 6：参考负向原点开关和 index 信号的原点模式.....	111
7.3.5 原点模式 7 和 10：参考原点开关、index 信号和起始正向运动的原点模式.....	113
7.3.6 原点模式 11 和 14：参考原点开关、index 信号和起始负向运动的原点模式.....	114
7.3.7 原点模式 15 和 16：保留.....	114
7.3.8 原点模式 17-31：没有 index 信号的原点模式.....	114
7.3.9 原点模式 32 和 33：由 index 信号确定的原点模式.....	116
7.3.10 原点模式 34：以当前位置做为原点的原点模式.....	116
7.3.11 原点模式-17 和-18：参考机械末端位置为原点的原点模式.....	116
7.3.12 原点模式-1 和-2：参考机械末端位置和 index 信号的原点模式.....	117
7.4 手动查找原点.....	118
7.5 Sequence 编程找原点范例.....	118
第八章 性能调节.....	120
8.1 正反转介绍.....	120
8.2 各种正反转控制介绍.....	120
8.2.1 带位置控制的定时正反转.....	120
8.2.2 带速度控制的定时正反转.....	122
8.2.3 带速度控制的位置正反转.....	123
8.2.4 带位置控制的速度正反转.....	123
8.3 伺服控制原理.....	124
8.4 ED 伺服参数设置（ED430/ED620/ED630）.....	125
8.4.1 速度环参数.....	125
8.4.2 位置环参数.....	125
8.4.3 参数调整步骤.....	125
第九章 通讯协议.....	132
9.1 硬件介绍.....	132

9.1.1 RS232 接口.....	133
9.1.2 RS485 接口.....	135
9.2 协议介绍.....	138
第十章 ED 伺服现场总线接口.....	141
10.1 现场总线简介.....	141
10.2 CAN 总线通讯.....	141
10.2.1 硬件介绍.....	142
10.2.2 软件介绍.....	144
10.2.3 CAN 通讯案例.....	150
10.3 Profibus 总线通讯.....	150
10.3.1 硬件介绍.....	150
10.3.2 软件介绍.....	153
10.3.3 Profibus 通讯案例.....	159
第十一章 常用对象列表.....	165
术语表.....	183
附录.....	186
附录 A: 总线通讯范例.....	187
一: ED 伺服与 S7-300 利用 Profibus DP 总线进行通讯.....	187
二: ED 伺服与施耐德 M258 利用 CANopen 总线进行通讯.....	198
三: ED 伺服与 Codesys 利用 CANopen 总线进行通讯.....	205
附录 B: 串口通讯范例.....	213
一、 Kinco 触摸屏与 ED 伺服驱动器的联接与应用.....	213
二、 西门子 S7200 PLC 和 ED 伺服利用 RS485 通讯.....	218
三、 松下 FP0 PLC 与 ED 伺服利用 RS232 通讯.....	219
四、 三菱 FX2N PLC 与 ED 伺服利用 RS422 通讯.....	220
五、 欧姆龙 CJ1W-CIF11 通讯单元与 ED 伺服利用 RS422 通讯.....	222
附录 C: 顺序编程 (Sequencer programming) 范例.....	228
一、 I/O 口定义及功能.....	228
二、 各数字口调用程序段定义.....	228
三、 程序定义.....	229
附录 D: ED 伺服应用范例.....	235

一、主从跟随控制.....	235
二、模拟量控制 ED 伺服电机速度和位置.....	241
附录 E: 全闭环控制.....	245
附录 F: 利用 ECO2LOAD 软件导入和导出数据.....	246
附录 G: 程序段密码保护功能.....	247
附录 H: 报警信息原因及排除.....	249
附录 I: 常见参数单位换算表.....	252

第一章 功能及构成

1.1 功能

ED 系列伺服系统，属于 Kinco®系列伺服中的一款总线型伺服系统，提供速度、位置和力矩全数字化控制。ED 系列驱动器提供丰富的通讯接口，目前支持的通讯方式有 RS232、RS485、CANopen、Profibus-DP, 上位控制器（PC、PLC、单片机等）通过上述通讯方式实现对驱动器内地址的读写（不同的地址具备不同的读/写属性），驱动器之间直接又可利用上述通讯方式方便的实现组网，为多轴之间数据的交换、传递提供快速、便捷的处理能力。

ED 系列伺服驱动器支持和 Kinco 系列触摸屏、文本显示器的 RS232、RS485 直接连接（kinco 的 HMI 系列产品均提供了 ED 系列伺服驱动器的驱动接口），针对 CANopen 和 Profibus-DP 总线应用，我们提供了标准的 EDS 和 GSD 文件，用于两种总线方式下组网配置。

除提供总线通讯功能外，ED 系列驱动器还可以存储 256 段程序（每段程序最多 8 条指令），利用内部程序可保存多达 256 段运动曲线，包含加速度、减速度、最大速度、目标位置等重要的运动参数，所有程序段均可以通过驱动器自带的数字输入口（Digital input）直接调用。

ED 系列伺服系统，目前有二相和三相共 2 个系列，二相系列有 ED100，三相系列有 ED430、ED620、ED630 共 3 个类别。二相系列伺服系统，由于电机采用的是多极对设计（50 个极对），低速扭矩大，驱动器可瞬时输出大电流，适合快速起停的应用场合。三相系列伺服系统，为市场主流 AC 伺服系统，已经在诸多行业得到了广泛应用。

1.2 控制方式

ED 驱动器的控制方式比较特别，不同于传统的日系脉冲型伺服控制方式，同时也不提供参数设定按键和数码管显示。

其控制方式主要通过以下三种

- 1: 通过串口通讯，CANopen 通讯，Profibus-DP 通讯直接规划运动曲线完成运动控制
- 2: 通过主编码器口信号进行主从跟随控制（ED100 不向外提供编码器输出接口）
- 3: 通过外部 I/O 触发调用由内部编程（或者直接设定对象字典等方式）规划好的运动曲线，完成伺服运动控制

其参数设定和显示方式主要通过以下二种

- 1: 通过内部编程或者直接设定对象字典等方式设定各种参数等，通过 PC 机连接 ECO2WIN 等软件查看参数

2: 通过串口通讯, CANopen 通讯, Profibus 通讯直接设定参数或者显示参数

1.3 构成

一套完整的 ED 系列伺服系统由 ED 驱动器、Kinco®伺服电机、电机电缆、编码器电缆、接线端子座、ENC15-9 Adapter (选配) 构成。

您可以登录我们的网站 www.kinco.cn 的“下载中心”去免费下载最新的软件和资料。

请检查电机和驱动器型号是否与您订购的相符, 仔细清点装箱货品是否齐全。若有疑问, 请尽快与我们联系。如果您在使用过程中有什么疑问, 请直接拨打电话: 0755-26585555 (深圳总机), 或者免费客户服务热线 4007005281 联系我们。

第二章 安装与使用事项

2.1 使用环境

工作环境温度： 0~40° C

保存温度： -10~70° C

工作环境湿度(不结露)： 5~95% (RH-2依照IEC 61131-2)

污染度： 2(依照 IEC 61131-2)

保护等级： IP20

安装场所： 无粉尘、干燥、可锁的（如电气柜）

安装位置： 垂直(参照“ED系列驱动器安装手册”)

安装高度： 海拔1000m(全系列)

ED430 系列功率损耗：

ED430-0020, AC220V, @1.8Arms 15W

ED430-0040, AC220V, @3.6Arms 24W

ED430-0075, AC220V, @4.0Arms 27W

ED430-0105, AC220V, @4.2Arms 30W

ED430-0126, AC220V, @5.0Arms 35W

ED430-0157, AC220V, @6.0Arms 42W

ED620 系列功率损耗：

ED620-0126, AC380V, @3.2Arms 40W

ED620-0157, AC380V, @4.5Arms 50W

ED620-0188, AC380V, @5.6Arms 60W

ED630 系列功率损耗：

ED630-0250, AC380V, @6.8Arms 85W

ED630-0300, AC380V, @10.0Arms 120W

ED630-0400, AC380V, @13.0Arms 150W

冷却方式： 空气对流，若连续电力损耗>40W，驱动器需加装散热片。Kinco ED100 驱动器没有外接散热片，ED430、ED620 和 ED630 系列驱动器，额定输出功率大于或等于 1.05kw 的所有型号都自带散热片。

2.2 安装间距和方向

请按照 3.4.3 “EMC 安装”一节中规定的间距安装，ED100 驱动器采用卧式安装，其它型号 ED 驱动器采用立式或卧式安装。

2.3 防止异物进入

1. 务必安装在控制柜内。
2. 在安装控制柜时，不要使金属屑、金属线头等异物进入伺服驱动器内。
3. 避免油、水、金属粉末等异物进入伺服驱动器内。
4. 存在有害气体及灰尘较多的应用场合，务必采用清洁空气对控制柜强制通风，避免这些物质进入驱动器内。

2.4 电缆安装

电机电缆和编码器电缆有一定的强度限制，为了保证电缆长时间使用，在连接电缆时请注意：

1. 仔细检查电缆的布线方法，不要在连接处加大电缆的弯曲度，也不要使电缆自身重量作用在接口部分，导致电缆在重力作用下（针对驱动器垂直安装、电缆较长、机器振动情况）从驱动器接口处松脱，接触不良，从而导致系统故障。
2. 在一些应用场合，电机电缆做往返运动，在这种应用，要注意电缆由于长时间往返折弯，会导致折弯处受损断裂，严重情况下，由于电缆内多根导线破损断裂导致短路，烧毁电机绕组和编码器电路。Kinco 伺服配套的电机电缆和编码器电缆具备良好的耐折弯性能，可以满足工业环境要求。**需要注意的是，用户自行延长的电缆，在选型时必须符合 Kinco 电缆性能参数，确保电缆的电流容量、耐折弯性能满足长时间、安全使用要求。**我们建议，在电缆频繁往返运动的场合，务必采用**拖链结构**来安装电机电缆和编码器电缆。
3. 为了避免编码器信号受干扰导致驱动器故障，务必确保编码器电缆和设备中强电电缆互相隔离，不要安装在同一个线槽内。

2.5 使用事项

2.5.1 使用原则

对本产品所有的运输、储藏、安装、连接、和维护工作，必须由有相应资格的熟练技术人员实行；必须遵守国家 and 地方的安全和意外事故预防相关规则。有资格的熟练技术人员是指有相关的职业资格，而且有执行上述工作的知识和经验。

在安装调试之前请仔细阅读本手册。不正确的安装、使用、操作可能导致人员或设备的伤害、损坏，在安装和调试驱动器时一定要遵守本手册和相关技术手册（参看选型手册技术规格）的技术要求。

本驱动器包含静电敏感部份，处置不当可能损害到驱动器。在你接触驱动器之前，请先释放身上带的静电。避免接触高度绝缘的材料（例如合成纤维、塑料等）。即使马达不动作时电机和主电源联线部分也可能带电，在带电的情况下驱动器的所有连线既不能插入也不能拔出，电压产生的电弧可能伤害人和联接器。不要在切断电源之后立刻接触连线，应该在切断电源之后至少等候6分钟，驱动器内电容残余电量要6分钟以后才能消失。

2.5.2 使用环境

该驱动器适应于在工业设备仪器中使用，符合低压产品标准73/23/EEC的要求。所有在本文中提到的技术规格和要求一定要严格遵守。除非采取特别措施，禁止在下列场合使用本驱动器：

1. 危险的场合
2. 靠近油、酸、瓦斯、水汽、灰尘、放射线等的场合。

2.5.3 安装要求

在安装和固定之前，请注意以下几点：

确保驱动器是安装在具有适当空间，且没有较大振动的底座上或电气柜里；

通风良好；

确保与毗连的元件之间有足够的空间距离；

当安装和操作时避免对人或设备造成伤害，不要直接接触、弯曲、损害带电的电气元件、接头，不要损坏原有绝缘层。

2.5.4 电气连接前要求

在文中被提到的所有安装要求一定要遵守，所有的相关工作一定要实行。

接线前请确认：

电气元件与电源断开并且不会出现意外；

再次检查设备已经断电；

确保附加监测和保护装置正常。

电源接通前，请确认：

是否采用了适用标准和规格的电源；

电源连线是否安全可靠；

电磁兼容装置是否可靠安装，例如接地线、滤波器、通讯电缆和动力电缆是否分开放置等。

2.5.5 操作

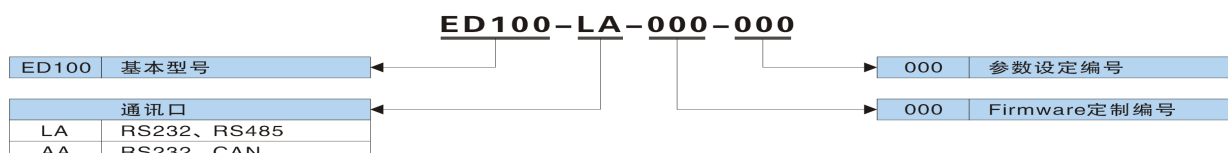
在试运行期间不要使监测和保护装置失效，电控柜的门及其他需要关闭的必须在操作期间被关闭，要改变操作状态，或有别的疑问，必要时请联系生产厂商或代理商。

第三章 接口信号与接线

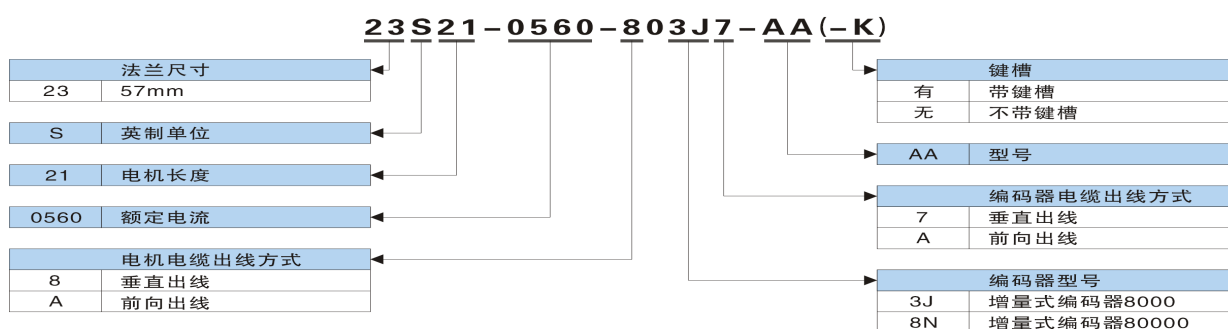
3.1 ED 系列驱动器型号

2 相 ED 伺服驱动器命名规则：

两相伺服驱动器命名规则



两相伺服电机命名规则



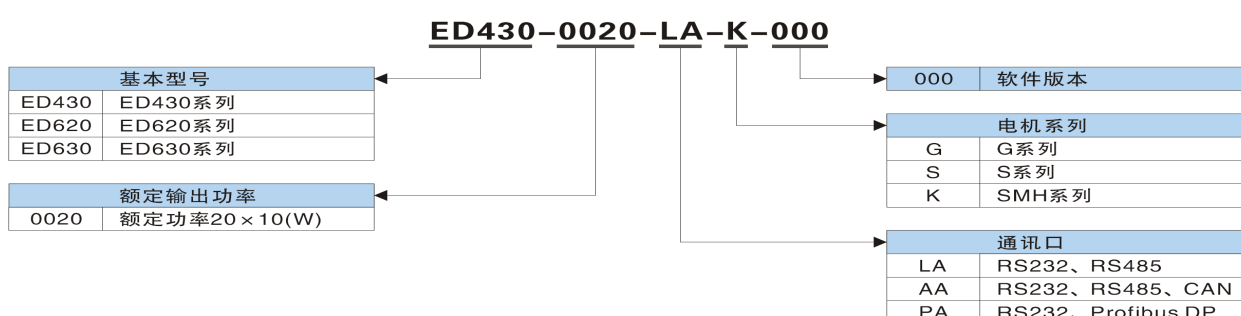
根据最大输出功率和提供的通讯接口，两相 ED100 系列驱动器可以分为下列型号：

型号	最大输出 功率	RS232	CAN	Profibus	RS485
ED100-AA-000-000	560W	有	有		
ED100-LA-000-000	560W	有			有

表 3.1.1 二相驱动器型号列表

3 相 ED 伺服驱动器命名规则：

三相伺服驱动器命名规则



根据额定功率和提供的通讯接口，三相 ED430 ED620 ED630 系列驱动器可以分为下列型号：

型号	额定功率	RS232	CAN	Profibus	RS485
ED430-0020-LA-K-000	200W	有			有
ED430-0020-AA-K-000	200W	有	有		有
ED430-0020-PA-K-000	200W	有		有	
ED430-0040-LA-K-000	400W	有			有
ED430-0040-AA-K-000	400W	有	有		有
ED430-0040-PA-K-000	400W	有		有	
ED430-0075-LA-K-000	750W	有			有
ED430-0075-AA-K-000	750W	有	有		有
ED430-0075-PA-K-000	750W	有		有	
ED430-0100-LA-K-000	1000W	有			有
ED430-0100-AA-K-000	1000W	有	有		有
ED430-0100-PA-K-000	1000W	有		有	
ED430-0105-LA-K-000	1050W	有			有
ED430-0105-AA-K-000	1050W	有	有		有
ED430-0105-PA-K-000	1050W	有		有	
ED430-0126-LA-K-000	1260W	有			有
ED430-0126-AA-K-000	1260W	有	有		有
ED430-0126-PA-K-000	1260W	有		有	
ED430-0125-LA-K-000	1250W	有			有
ED430-0125-AA-K-000	1250W	有	有		有
ED430-0125-PA-K-000	1250W	有		有	
ED430-0105-LA-S-000	1050W	有			有
ED430-0105-AA-S-000	1050W	有	有		有
ED430-0105-PA-S-000	1050W	有		有	
ED430-0157-LA-S-000	1570W	有			有
ED430-0157-AA-S-000	1570W	有	有		有
ED430-0157-PA-S-000	1570W	有		有	

表 3.1.2 三相驱动器型号列表

型号	额定功率	RS232	CAN	Profibus	RS485
ED620-0126-LA-K-000	1260W	有			有
ED620-0126-AA-K-000	1260W	有	有		有
ED620-0126-PA-K-000	1260W	有		有	
ED620-0157-LA-K-000	1570W	有			有
ED620-0157-AA-K-000	1570W	有	有		有
ED620-0157-PA-K-000	1570W	有		有	
ED620-0188-LA-K-000	1880W	有			有
ED620-0188-AA-K-000	1880W	有	有		有
ED620-0188-PA-K-000	1880W	有		有	
ED630-0180-LA-G-000	1800W	有			有
ED630-0180-AA-G-000	1800W	有	有		有
ED630-0180-PA-G-000	1800W	有		有	
ED630-0250-LA-G-000	2500W	有			有
ED630-0250-AA-G-000	2500W	有	有		有
ED630-0250-PA-G-000	2500W	有		有	
ED630-0300-LA-G-000	3000W	有			有
ED630-0300-AA-G-000	3000W	有	有		有
ED630-0300-PA-G-000	3000W	有		有	
ED630-0400-LA-G-000	4000W	有			有
ED630-0400-AA-G-000	4000W	有	有		有
ED630-0400-PA-G-000	4000W	有		有	

表 3.1.2 三相驱动器型号列表

注：各驱动器所配电机型号及电机电缆，编码器电缆型号详情请参考 ED 伺服选型手册！该手册您可以向我们的销售工程师或我们的代理商索取。您也可以登录我们的网站 www.kinco.cn 的“下载中心”去免费下载。

3.2 ED 系列驱动器接口分布

3.2.1 ED100 系列驱动器接口分布

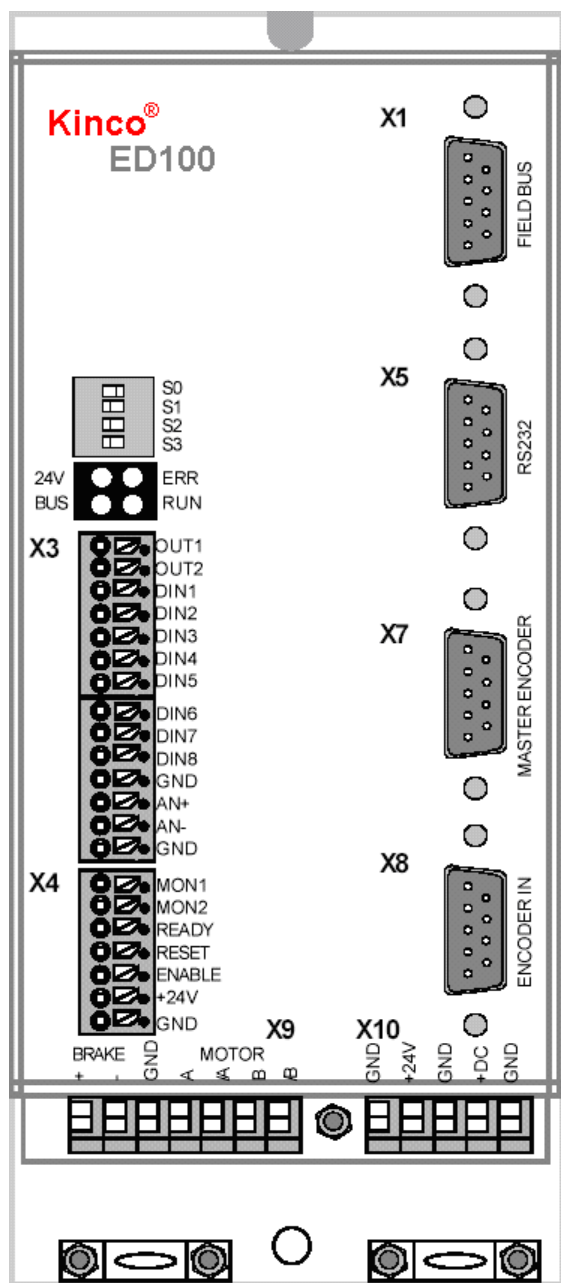


图 3.1 ED100 驱动器接口分布

X1: 现场总线口，提供 CANopen 总线或 RS485 接口，9 针公座。

X3: 数字输出/输入口、模拟量监控输入。

X4: 逻辑电源（24V）输入、模拟量输出、Enable 信号输入口、Reset 信号输入口、Ready 信号输出口。

X5: RS232 接口，一般用于 PC 下载参数，9 针母座。

X7: 外部主编码器信号输入口。

X8: 电机的编码器反馈信号输入口。

X9: 电机动力电缆接线口和刹车电源供应口。

X10: 动力电源输入口（输入范围 24VDC~70VDC）

在左图，X10 从左边起，GND/+24V, 和 X4 口的 GND/+24V 一样，都是 ED100 的控制逻辑电源输入口，只需要输入一路即可。

X10 右侧的 GND/+DC 为动力电源输入口。

24V 指示灯：逻辑电源输入指示。绿色常亮。

RUN 指示灯：逻辑电路正常运行指示。绿色闪烁。

BUS 指示灯：通讯（串口和总线）正常指示。红色闪烁。

ERROR 指示灯：1: 报警指示。红色常亮。

2: 驱动器重启指示。红色闪烁一次。

3: 程序底层刷入指示。红色常亮

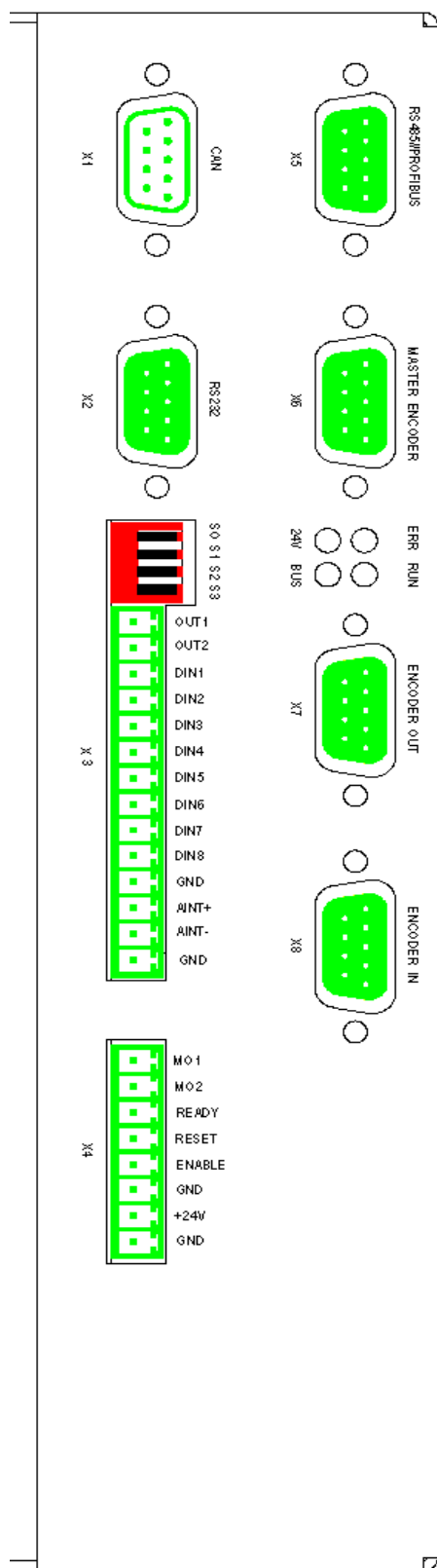
（同时其他三个指示灯全熄灭）

S0-S3: 驱动器站号拨码！二进制编制！范围 0-15！

超出范围，用软件内部偏移地址 2F80 设定。

（详见 P43 页）

3.2.2 ED430、ED620/ED630 系列驱动器接口分布



X1: CANopen 总线接口，9 针公座。

X2: RS232 接口，用于 PC 下载参数，9 针母座。

X3: 数字输出/输入口、模拟量输入口。

X4: 逻辑电源输入口、模拟量输出口、Enable 信号输入口、Reset 信号输入口、Ready 信号输出口。

X5: RS485 或 Profibus 通讯口。

X6: 电机编码器信号输出口。

X7: 主编码器信号（来自外部）输入口。

X8: 电机编码器信号输入口，接电机编码器电缆。

X9: 刹车电源输出口。

X10: 动力电源输入口（220VAC、380VAC、直流母线），

X11: 电机线接口（接电机动力电源）。

24V 指示灯：逻辑电源输入指示。绿色常亮。

RUN 指示灯：逻辑电路正常运行指示。绿色闪烁。

BUS 指示灯：通讯（串口和总线）正常指示。红色闪烁。

ERROR 指示灯：1：报警指示。红色常亮。

2：驱动器重启指示。红色闪烁一次。

3：程序底层刷入指示。红色常亮

（同时其他三个指示灯全熄灭）

S0-S3：驱动器站号拨码！二进制编制！范围 0-15！

超出范围，用软件内部偏移地址 2F80 设定。

（详见 P43 页）

图 3.2 ED430/ED620/ED630 驱动器接口分布

3.2.3 ED 系列驱动器接口功能介绍

ED100 系列驱动器接口功能：

接口	符号	功能
X1	FIELD BUS	CAN 总线接口或 RS485 接口，9 针公座。
X3	OUT1/OUT2	两个可自由定义的数字输出口，常用做“目标位置到”信号和报警信号输出
	Din1..Din8	数字输入口，上升或下降边沿触发或 BCD 编码触发内部程序段（可软件设置高电平或低电平输入有效），输入电平有效值+24VDC，电流>3mA，高电平持续时间>1ms。
	Din6	也可做为外部正限位信号输入口，具体使用方法参考 7.1 节。
	Din7	也可做为外部负限位信号输入口，具体使用方法参考 7.1 节。
	Din8	也可做为外部原点信号输入口，具体使用方法参考 7.1 节。
X4	AIN+/AIN-	模拟量输入口，常用于控制电机速度和力矩，输入范围±10V。
	MON1/MON2	两路模拟量输出口（0-5V），监控驱动器内部对象状态（可在软件里设置其监控的内容）。
	READY	数字输出口，输出为高电平（+24V）时表示驱动器处于“Ready to Switch On”状态。
	RESET	数字输入口，输入高电平（+24V）有效，上升沿触发时可复位驱动器故障
	ENABLE	数字输入口，使能信号输入口，驱动器工作时必须接到+24V，如果在驱动器工作时断开该输入，会导致驱动器报警“External enable low”。
	GND	逻辑地，为数字输入输出口公共端。
	+24V	逻辑电源“+24V”输入口，逻辑电源由驱动器外部电源提供。
X5	GND	逻辑电源“+24V 地”输入口。
	RS232	RS232 接口，9 针母座，常用于和 PC 连接编程用，也可以直接连接 kinco 触摸屏或文本显示器，也常用作其它控制器使用 RS232 和 ED 驱动器间通讯。
X7	MASTER ENCODER	主编码器信号输入口。
X8	ENCODER IN	电机的编码器信号输入口，直接连接伺服电机编码器电缆
X9	BRAKE/MOTOR	电机电缆接线口和抱闸电源输出

X10		ED100 动力电源和逻辑电源输入口
	GND	逻辑电源 GND 输入口，内部和 X4 的 GND 短接。
	+24V	逻辑电源+24V 输入口，内部和 X4 的+24V 短接。
	GND	动力电源“GND”接口。
	+DC	动力电源“+DC”接口，ED100 动力电源电压范围 24VDC~70VDC。
	GND	动力电源“GND”接口。

表 3.2 ED100 接口功能介绍

ED430、ED620 和 ED630 系列驱动器接口功能:

接口	符号	功能
X1	FIELD BUS	CAN 总线接口，9 针公座。
X5	FIELD BUS	RS485 或 Profibus 总线口。
X3	OUT1/OUT2	两个可自由定义的数字输出口，常用做“目标位置到”信号和报警信号输出。
	Din1..Din8	数字输入口，上升或下降边沿触发或 BCD 编码触发内部程序段（可软件设置高电平或低电平输入有效）。
	Din6	当不做为数字输入口使用时，可以做为外部的电机正向限位信号输入。
	Din7	当不做为数字输入口使用时，可以做为外部的电机负向限位信号输入。
	Din8	当不做为数字输入口使用时，可以做为外部原点信号输入。
	AIN+/AIN- GND	模拟量输入口，常用于控制电机速度和力矩，输入范围±10V。 逻辑地。
X4	MON1/MON2	两路模拟量输出口，监控驱动器内部对象状态。（可在软件里设置其监控的内容）。
	READY	数字输出口，输出为高电平（+24V）时表示驱动器无故障。
	RESET	数字输入口，上升沿信号输入时可复位驱动器故障。
	ENABLE	数字输入口，外部使能驱动器用，驱动器工作时必须接到+24V，如果在驱动器工作时断开该输入，会导致驱动器“External enable low”故障。
	GND	逻辑地。
	+24V GND	逻辑电源“24V”输入口，需要外部提供。 逻辑地。
X2	RS232	RS232 接口，9 针母座，常用于和 PC 连接编程用，也可以直接连接 kinco 触摸屏或文本显示器，也常用作其它控制器使用 RS232 和 ED 驱动器间通讯

X7	ENCODER OUT	电机编码器输出口, 在主从控制时可以输出电机编码器信号或者主编码器信号做为从站编码器信号输入。
X6	MASTER ENCODER	主编码器信号输入口。
X8	ENCODER IN	电机的编码器信号输入口, 直接连接伺服电机编码器电缆。
X9	BRAKE	抱闸电源输出。
X10	POWER/RB/DC -BUS	动力电源输入口、制动电阻接口、DC-BUS 输入。
X11	MOTOR	电机电缆接口。

表 3.3 ED430、ED620 和 ED630 接口功能介绍

3.3 驱动器接口电路

3.3.1 ED100 驱动器接口电路

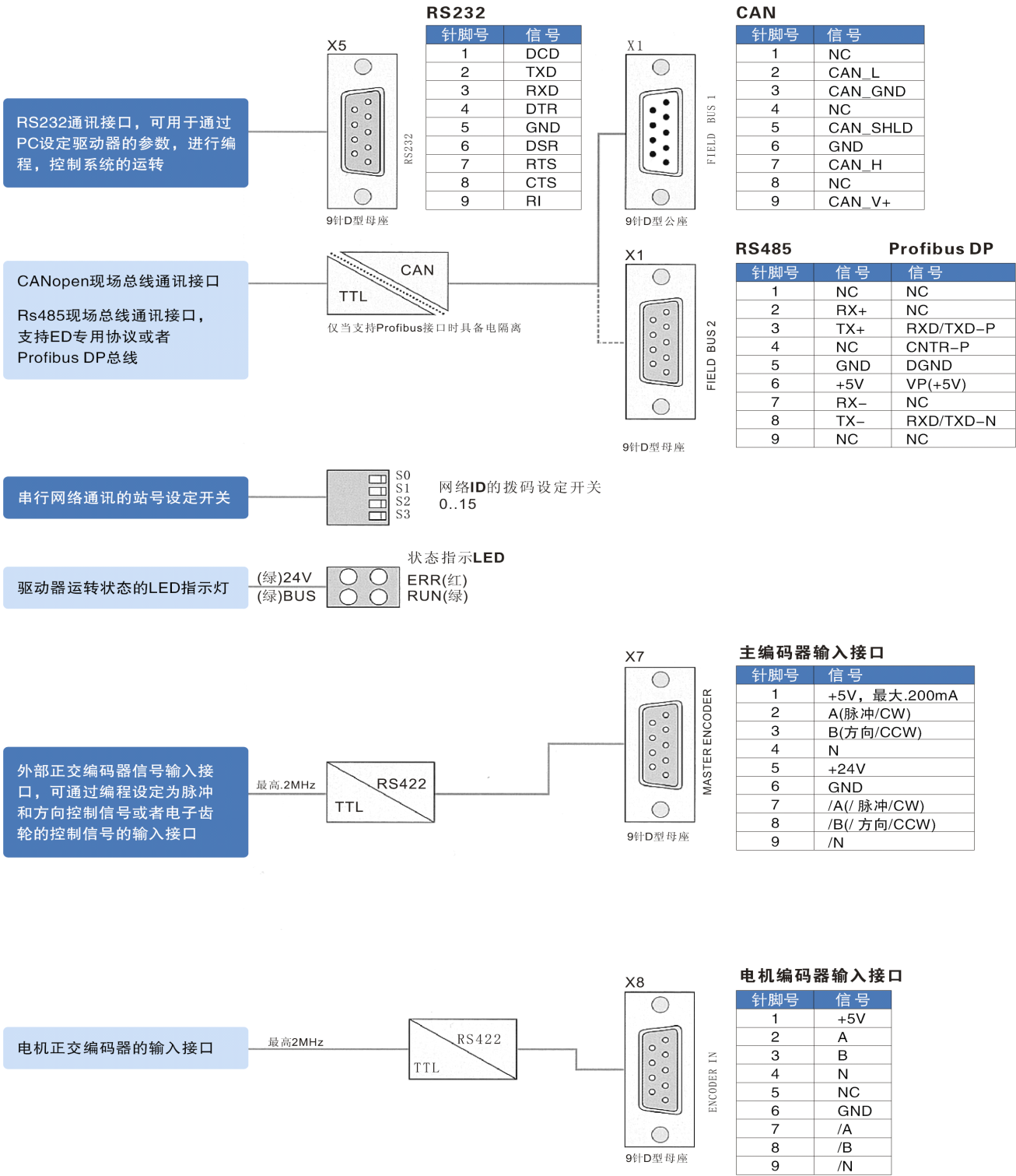
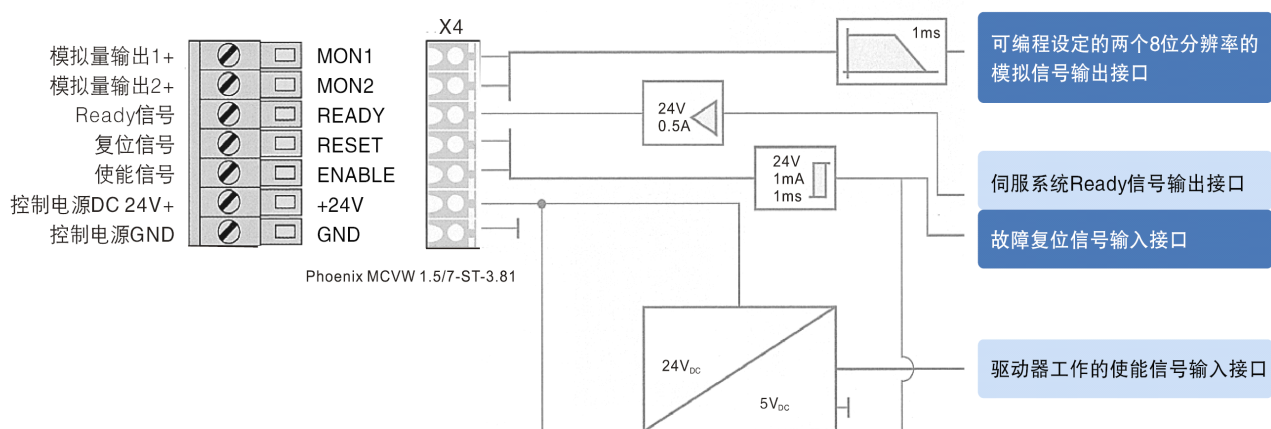
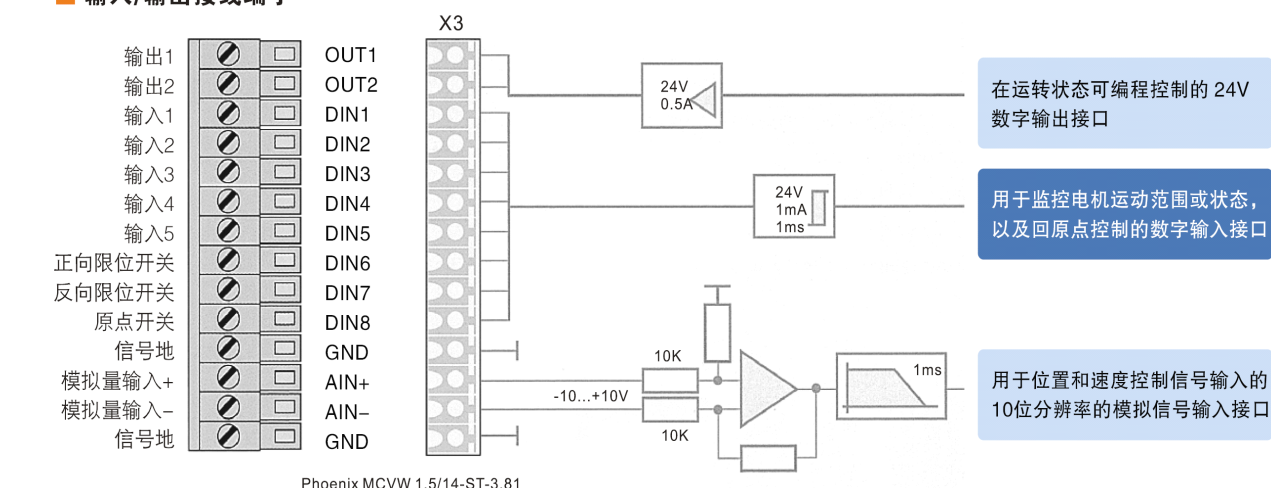
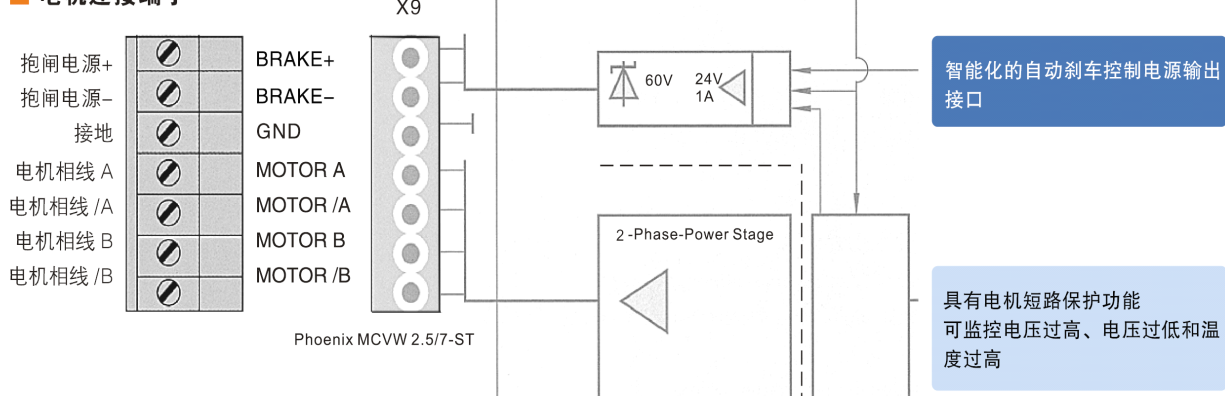


图 3.3 ED100 驱动器接口电路框架图-1

■ 输入/输出接线端子



■ 电机连接端子



■ 主电源连接端子

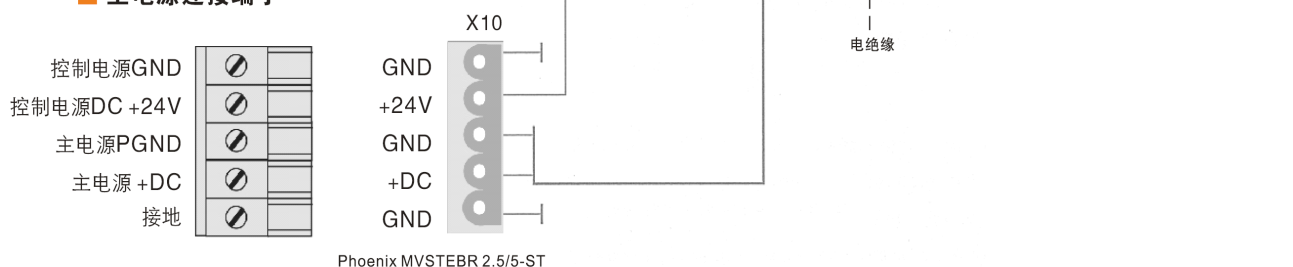


图 3.4 ED100 驱动器接口电路框架图-2

3.3.2 ED430 ED620 ED630 驱动器接口电路

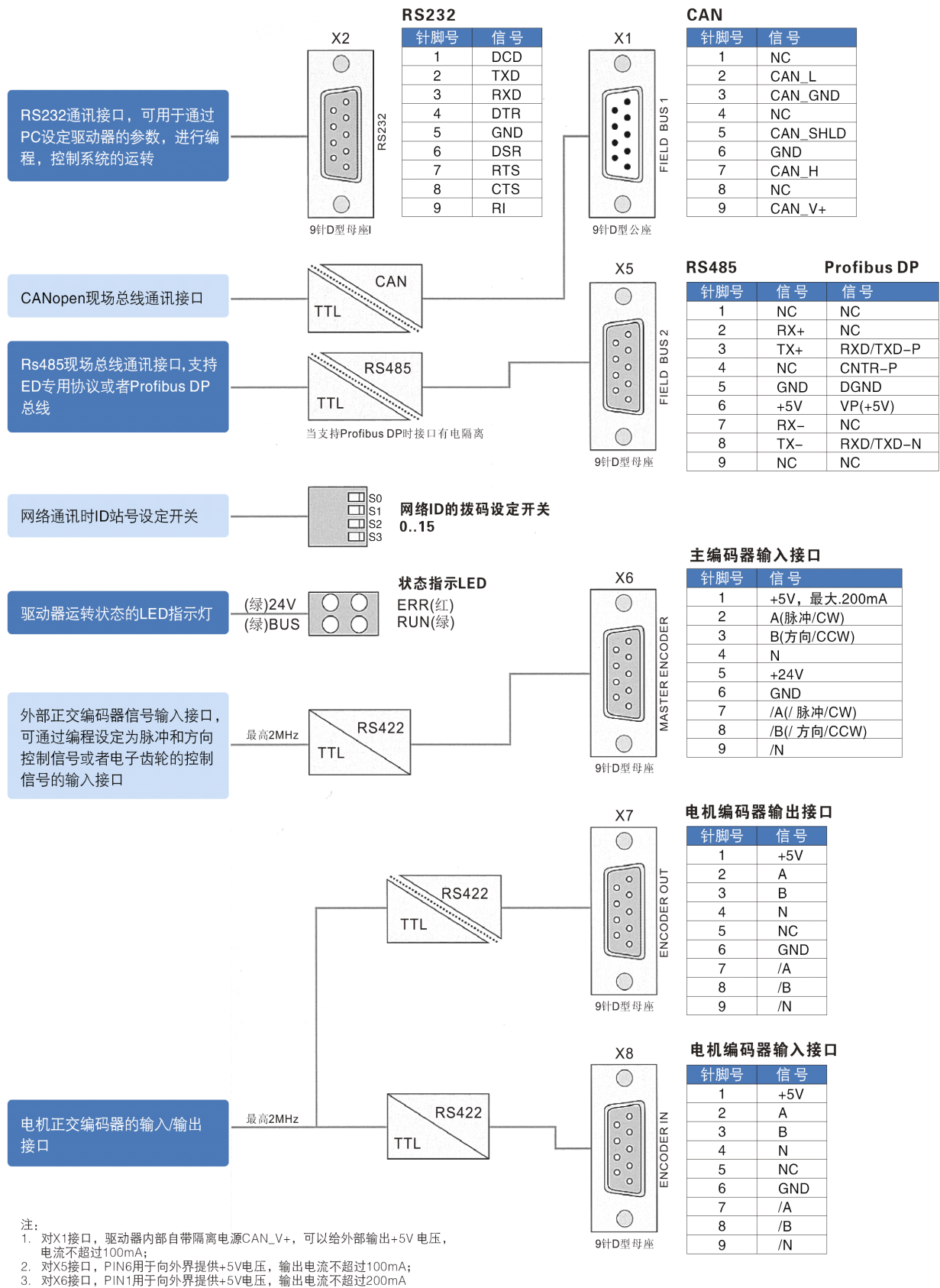


图 3.5 ED430、ED620、ED630 驱动器接口电路-1

■ 输入/输出接线端子

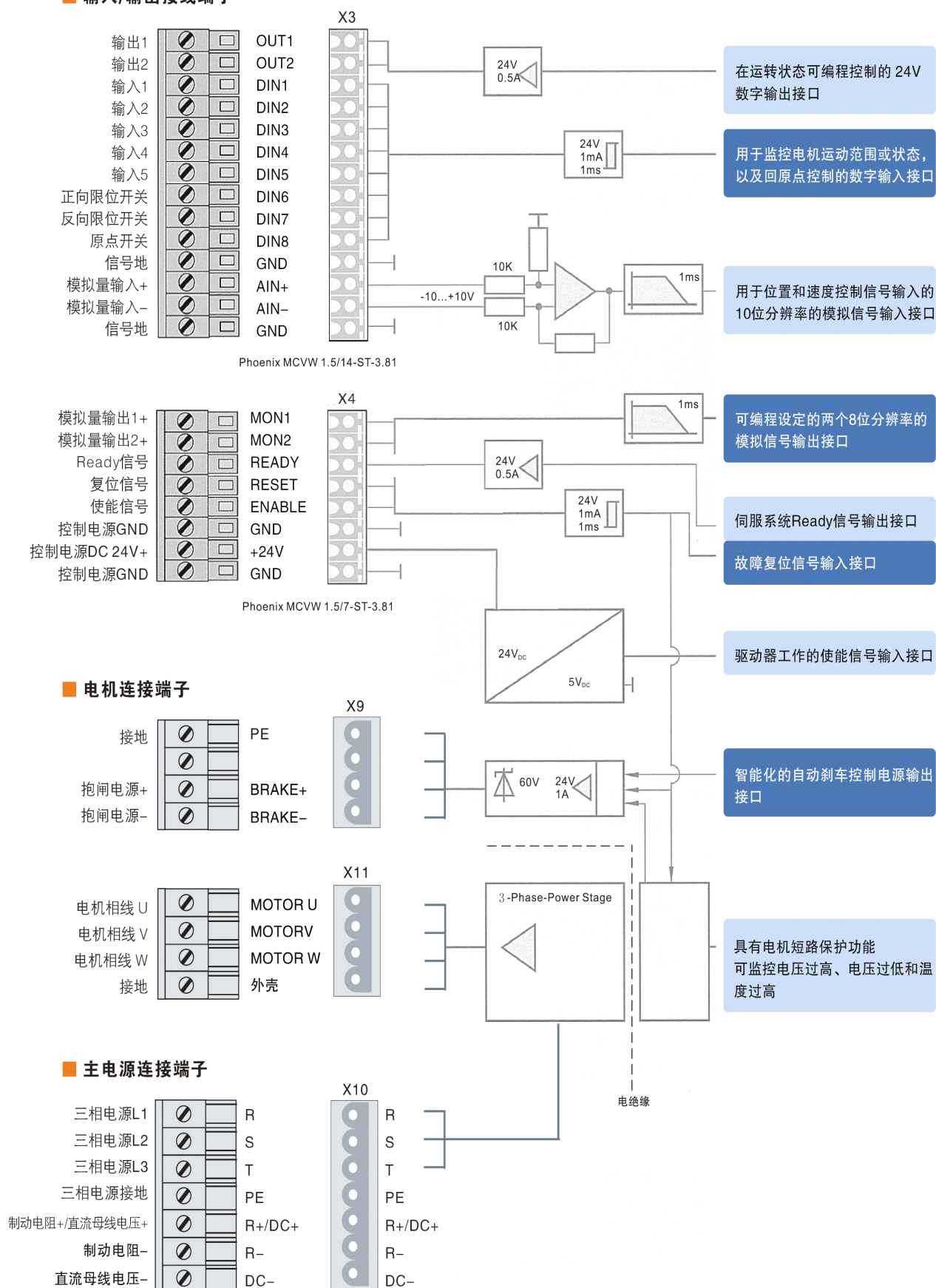


图 3.6 ED430、ED620、ED630 驱动器接口电路-2

3.4 接线

3.4.1 ED430 系列驱动器电源接线

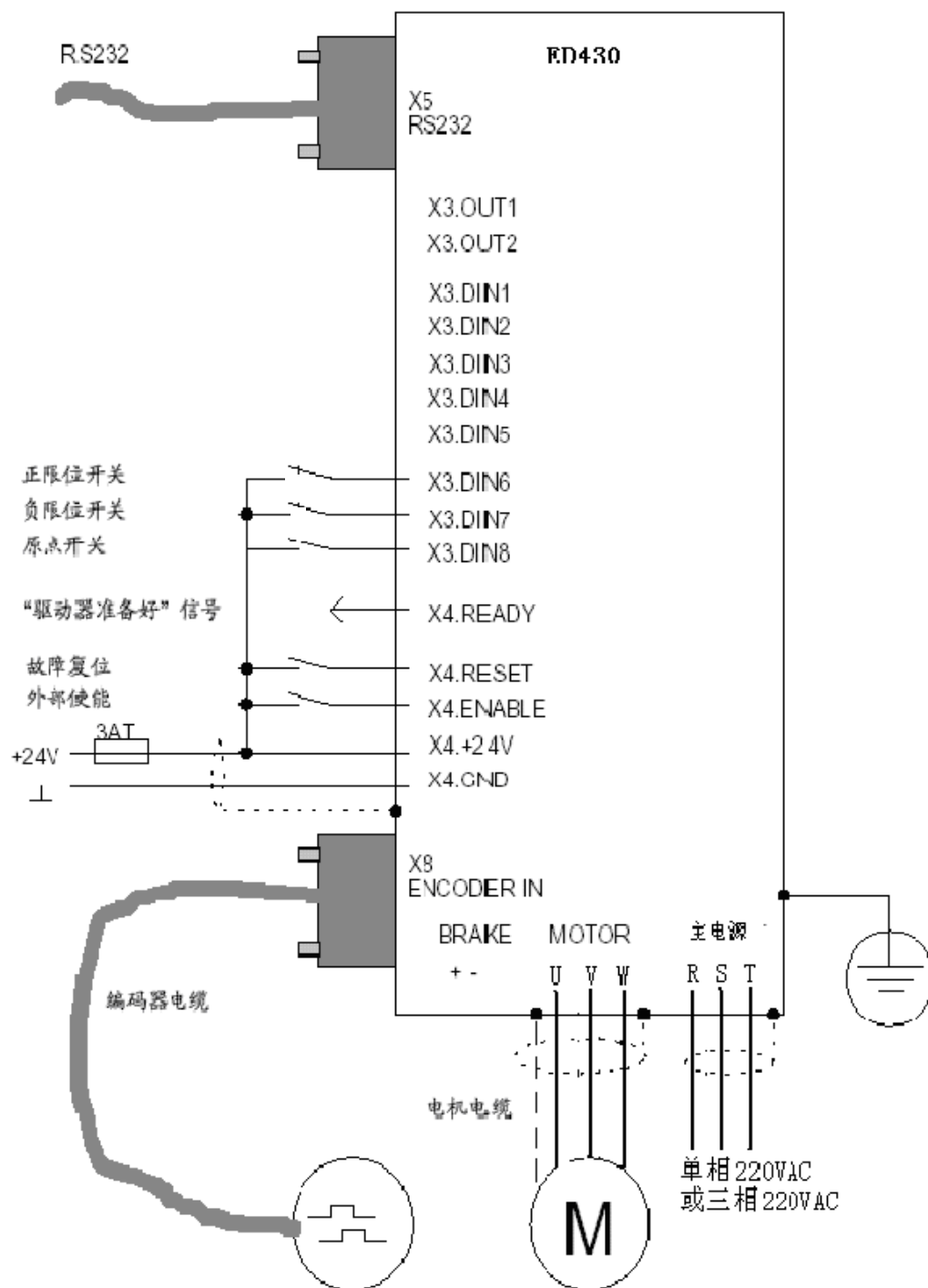


图 3.7 ED430 系列驱动器电源接线

3.4.2 ED620/ED630 系列驱动器电源接线

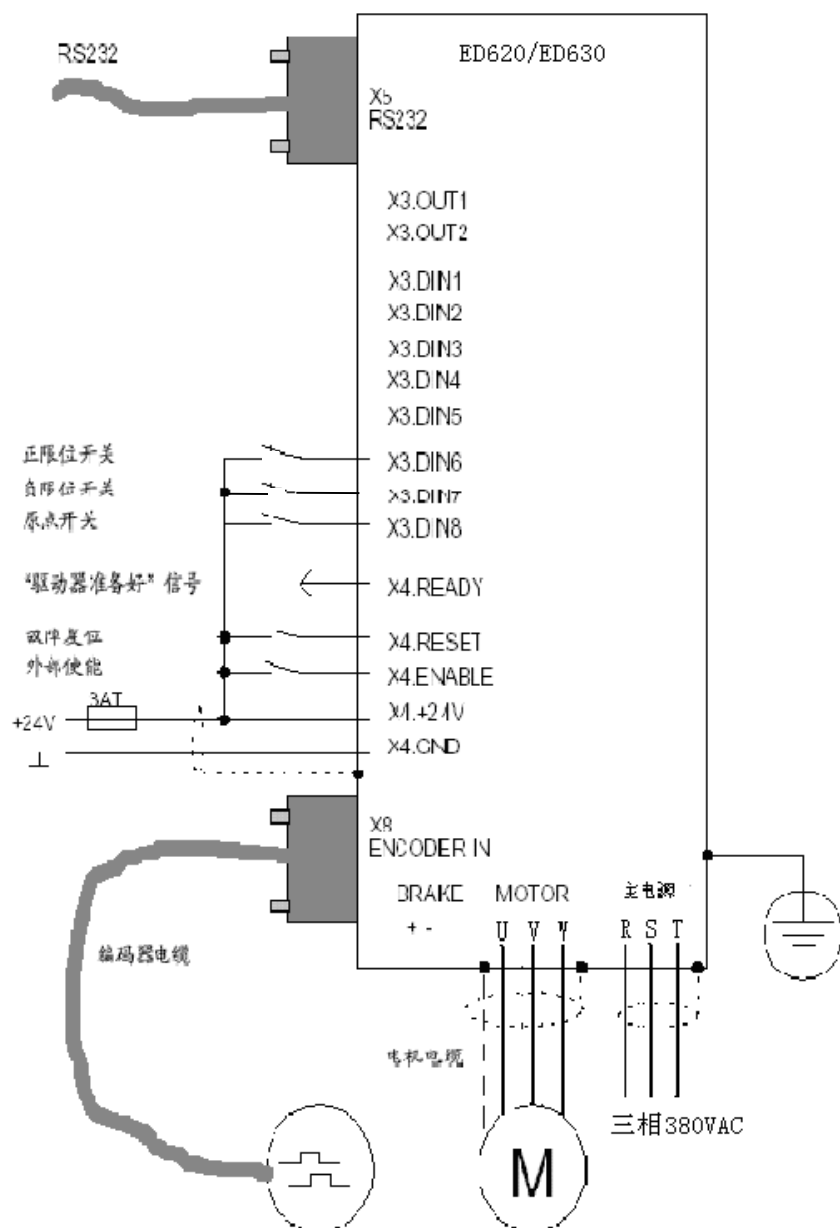


图 3.8 ED620、ED630 系列驱动器电源接线

3.4.3 EMC 安装

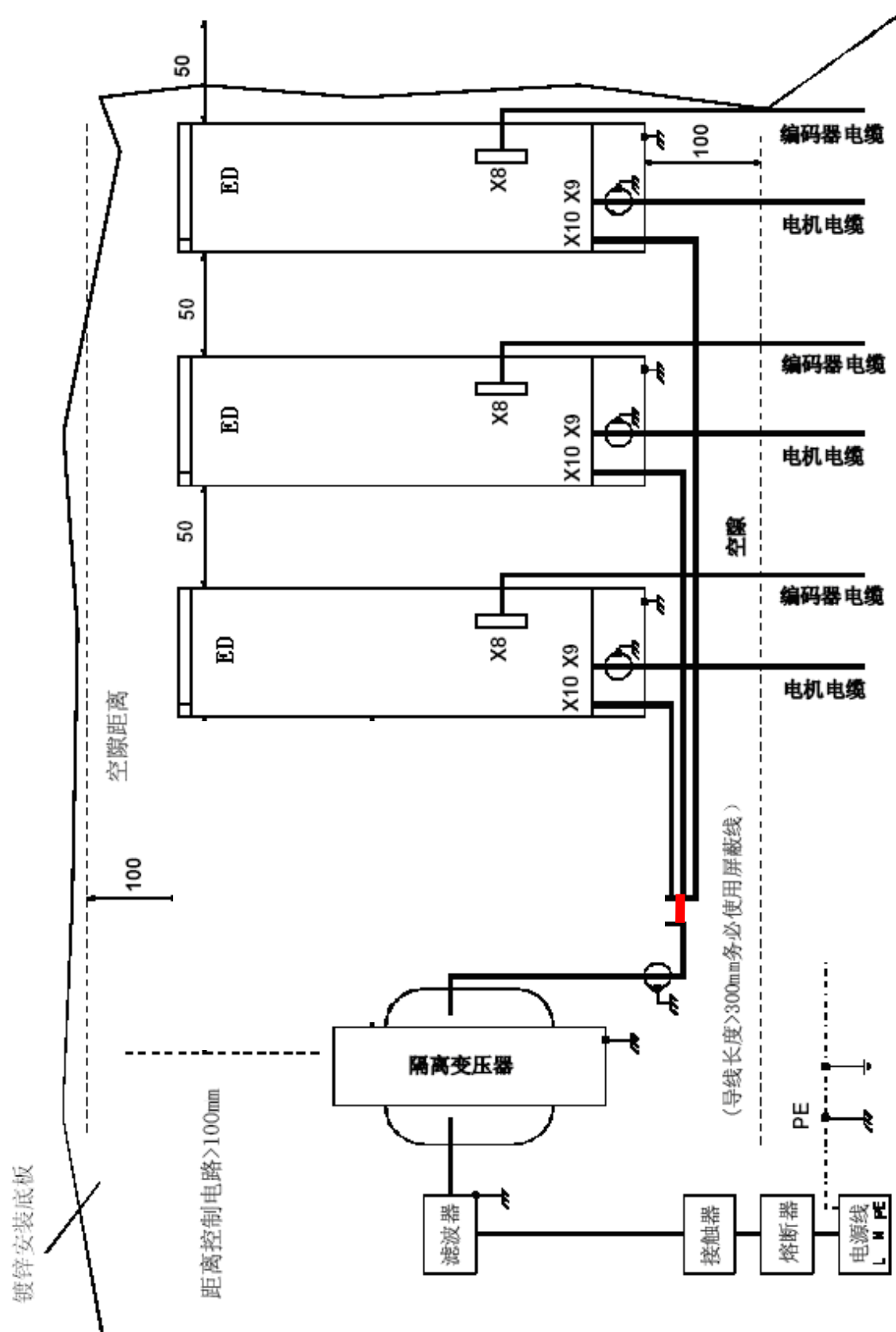


图 3.9 EMC 布线

3.4.4 ED 驱动器和 PLC 接口接线

说明：下图是 ED 100 和 PLC 典型接线图，同样适用 ED430、ED620 和 ED630。

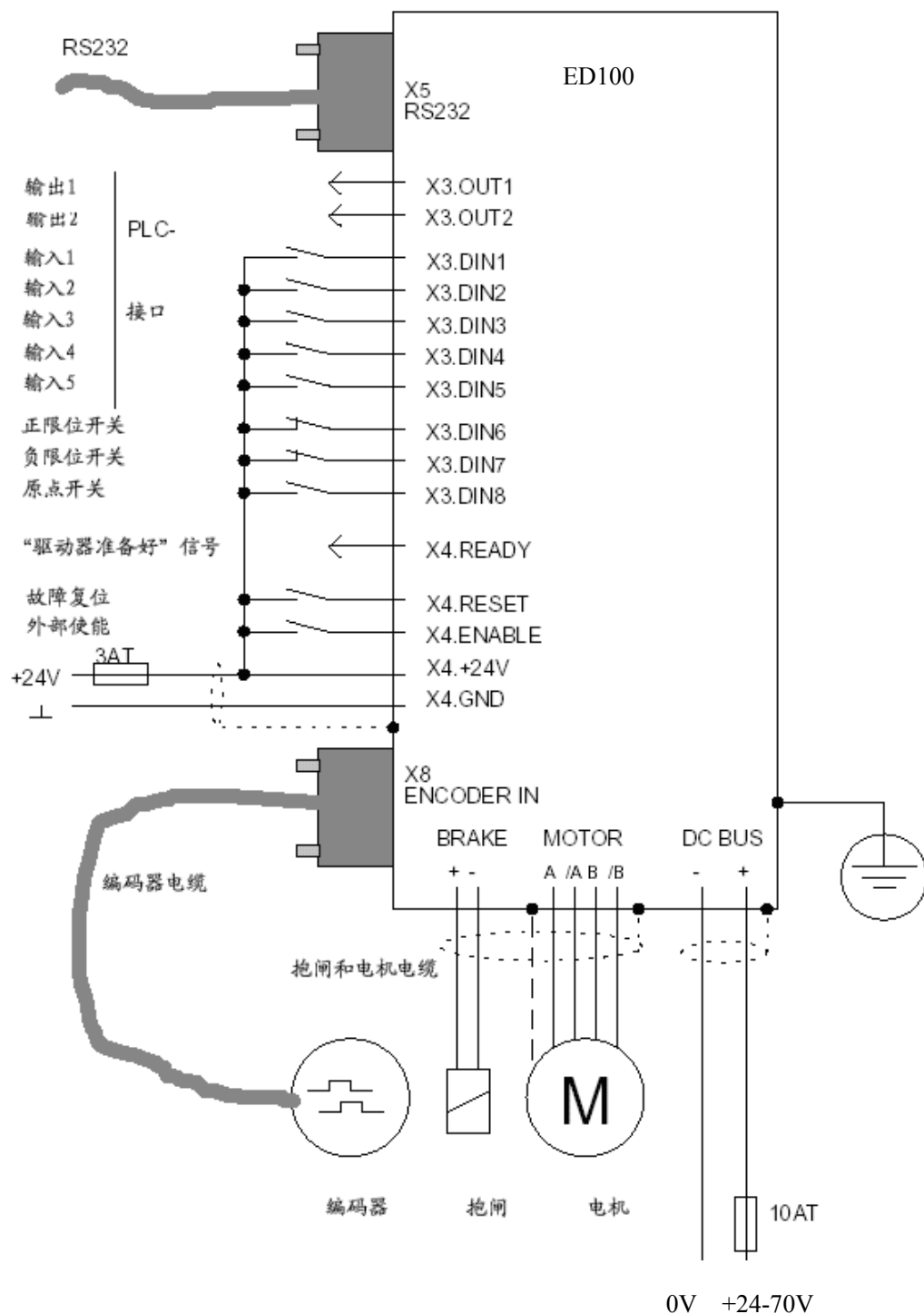


图 3.10 ED 驱动器和 PLC 接口图

3.4.5 ED 驱动器和电机之间连线

ED 伺服电机配套两根电缆，一根是编码器电缆，带 9 针插头，另一根是电机电缆，比编码器电缆粗。根据电机型号不同，各驱动器所配线缆型号及连接方式不同，具体连接详情请参考 ED 伺服选型手册！该手册您可以向我们的销售工程师或我们的代理商索取。您也可以登录我们的网站 www.kinco.cn 的“下载中心”去免费下载。

第四章 ECO2WIN 软件使用说明

本章将对 Kinco 伺服驱动器的编程软件 ECO2WIN 的使用进行详细的说明，本章主要描述 ECO2WIN V2.41 的版本功能。

4.1 软件安装

可以从 www.kinco.cn 下载 ECO2WIN 的安装软件。

请您下载以下文件：

- 1、 编程软件 ECO2WIN 0.0.0.241 英文版。
- 2、 参数导入和导出备份软件 ECO2LOAD 1.0.0.9 英文版。
- 3、 底层更新软件 ECO2FLASH 1.0.0.202 英文版。
- 4、 三相伺服软件补丁（dev）。

操作系统和硬件要求：

WIN98/2000/XP

奔腾 CPU（最少 200MHz）

内存大于 64MB

CD-ROM

带有 232 串口、并口或 USB 接口

软件介绍：

ECO2WIN：伺服驱动器的组态软件，用于调试驱动器、设定驱动器的各种运行参数，编辑 sequence 等。

ECO2LOAD：用于上传或下载驱动器的各种参数。

ECO2FLASH：用来更新的 Firmware（驱动器固件版本）。该软件请务必在我司技术人员指导下使用。

安装说明：

使用三相伺服 ED430/ED620/ED630 系列驱动器，安装完 ECO2WIN0.0.0.241 后，还需要安装三相伺服软件补丁（dev 文件）。

补丁安装方法：补丁软件包解压缩，将 Dev 文档覆盖至 ECO2WIN0.0.0.241 软件安装目录下的 Dev 文档。

4.2 快速入门

4.2.1 ED 运行所需的最低硬件要求

通过 ED 驱动器的 RS232 口或 CANopen 接口，使用 ECO2WIN 软件可以对 ED 系列所有的伺服驱动器进

行参数设置。

参照第 3 章或 ED 驱动器硬件手册正确连接伺服驱动器和电机。

- 利用 RS232 口进行编程的最低系统要求：

伺服驱动器 ED100、ED430、ED620、ED630；

提供给驱动器的控制逻辑电压 24VDC；

串行编程电缆，9 针 D 型头，详细接线见第 9.1 节 RS232C 通讯部分；

可选：ED 伺服电机；

可选：电机驱动电源，最小 24VDC(ED100)

- 利用 CANopen 接口进行编程的最低系统要求：

伺服驱动器 ED100、ED430、ED620、ED630(必须是带 CANopen 接口的驱动器)；

提供给驱动器的控制逻辑电压 24VDC ；

PEAK 公司 PEAK 系列 USB 或 LPT 适配器；

CANopen 通讯电缆，ED100 需要外部提供电源，具体见 PEAK 系列 USB 和 LPT 适配器使用说明.ED430、ED620、ED630 不需要外部提供电源；

可选：ED 伺服电机；

可选：电机驱动电源，最小 24VDC(ED100)

4.2.2 建立一个新的工程文件

1. 启动 ECO2WIN 软件

2. 点击 file->new 新建工程如图

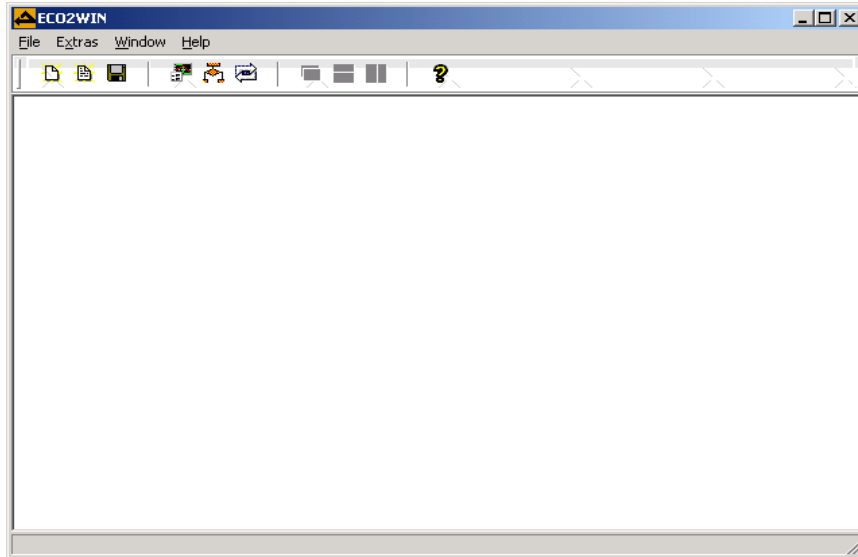


图 4.1 启动 ECO2WIN 软件

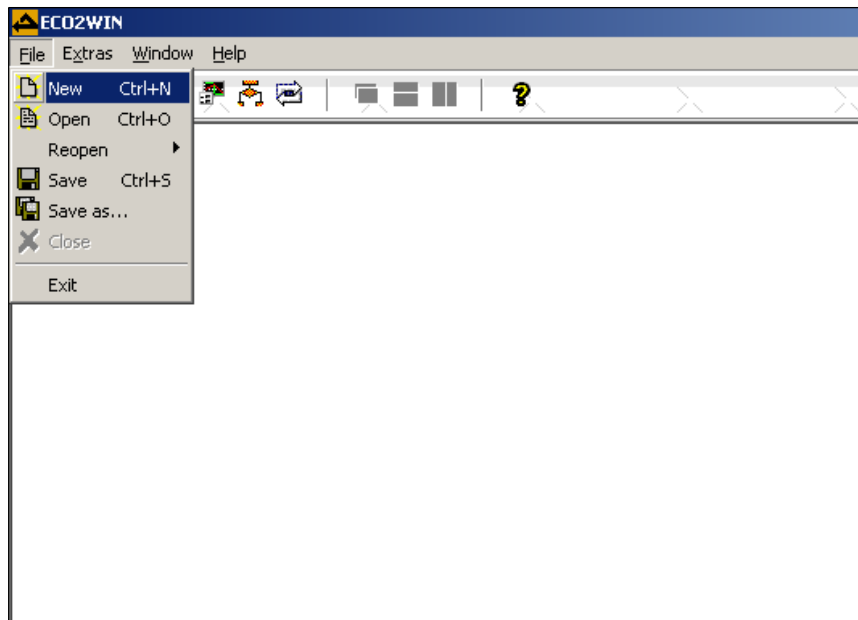


图 4.2 新建工程-1

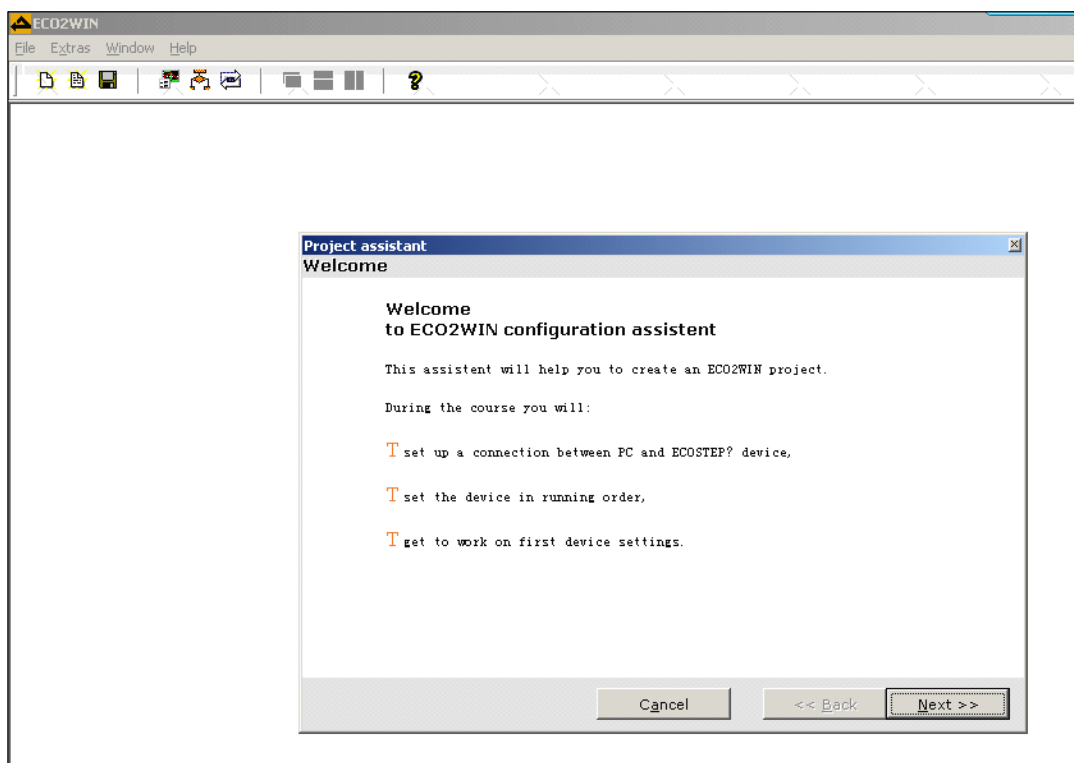


图 4.3 新建工程-2

3. 选择”automatic interface identification”自动搜寻驱动器，继续点击 Next

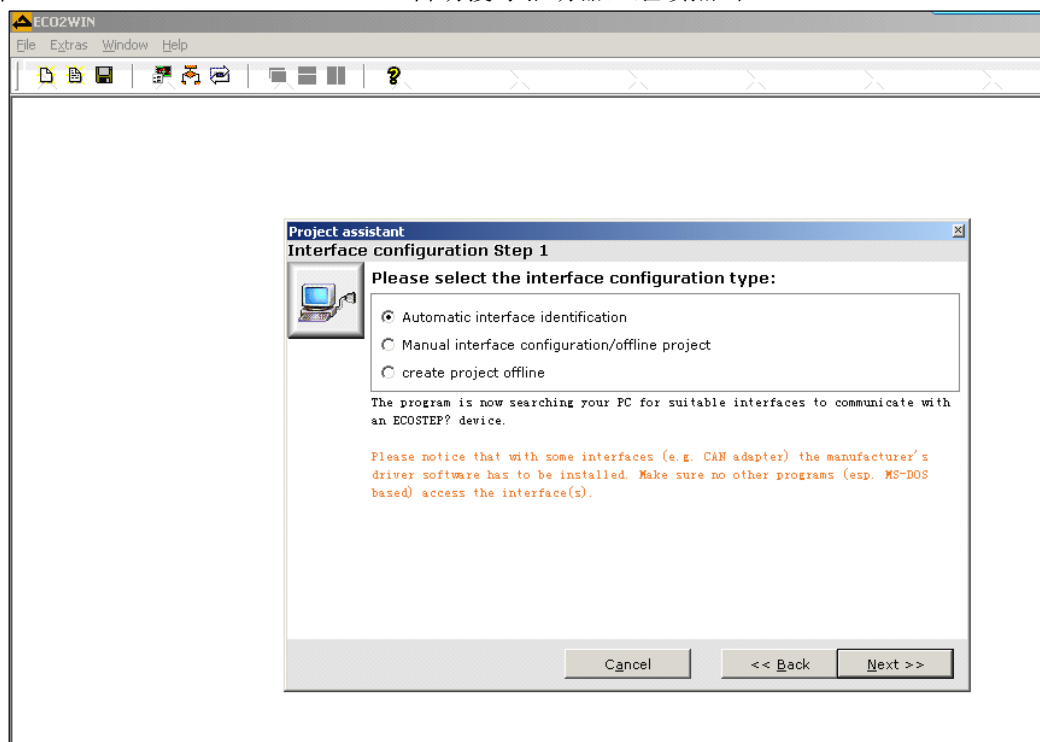


图 4.4 新建工程-3

4. 通讯端口配置

目前 ECO2WIN 软件支持串口和 PEAK 公司 CANopen 系列适配器，如果你使用的是 PEAK CANopen 适配器进行编程和对 ED 的参数设定，需要安装相应的 PEAK CANopen 适配器驱动程序。驱动程序可以到网站

上下载。

你可以手动选择通讯方式也可自动搜索，一般采用自动搜索方式，这样 ECO2WIN 会自动将当前 PC 机上所具备的与 ED 进行通讯的通讯方式列表显示，请选择其中一种，如果不能确认此通讯方式下的通讯参数设置正确与否，可按“Manual Configuration”进行参数配置，见下图：

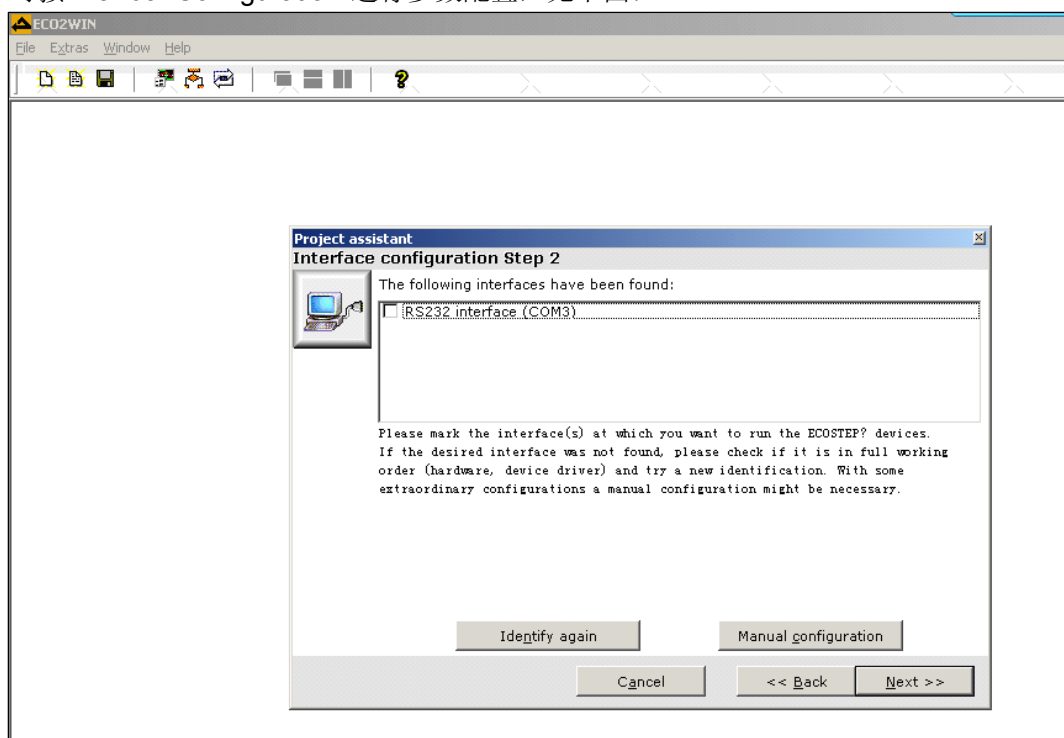


图 4.5 通讯口配置-1

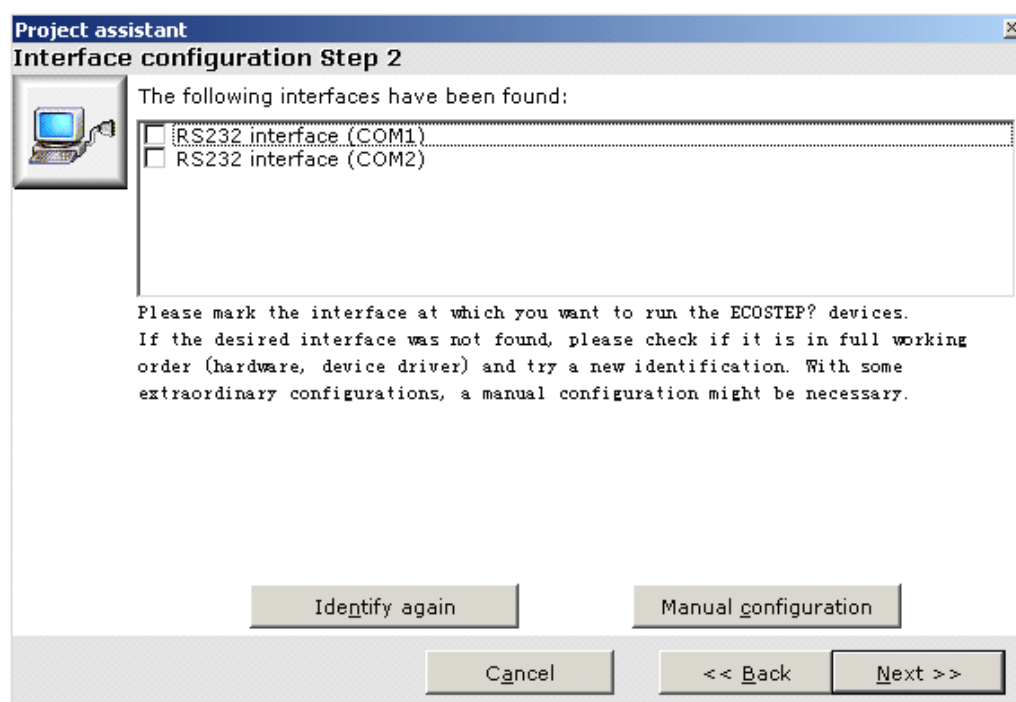


图 4.6 通讯口配置-2

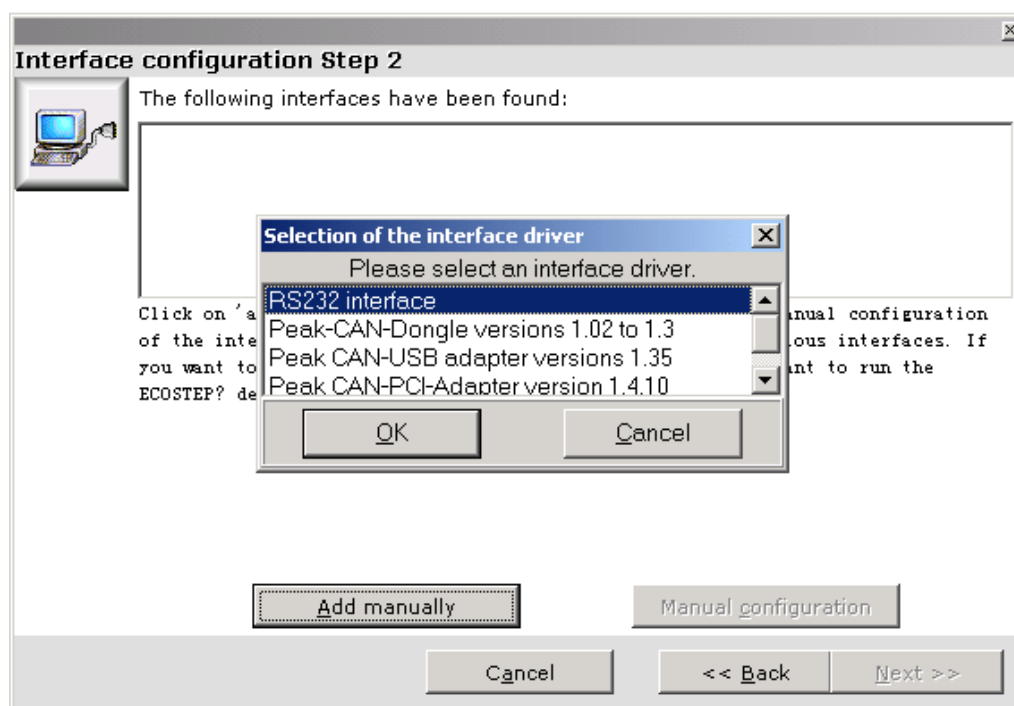


图 4.7 Manual Configuration

5. 驱动器配置

按“NEXT”进入下一个画面，软件开始使用所选择的通讯方式寻找通讯线路上的驱动器，同样你有两种方式寻找驱动器，缺省是自动搜索方式，如果选择“Manual Device Configuration”进行手动寻找，这时你就要从驱动器列表中准确找到相关驱动器型号，见下图：

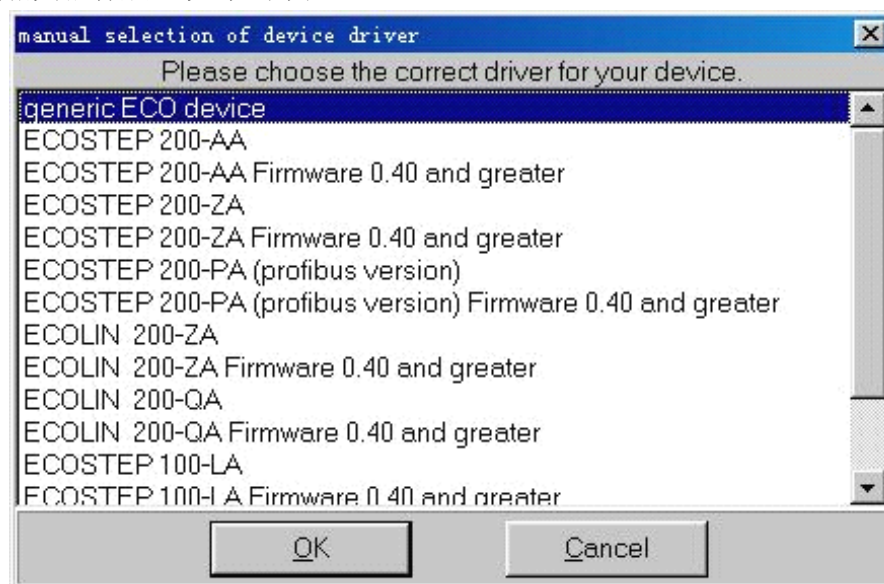
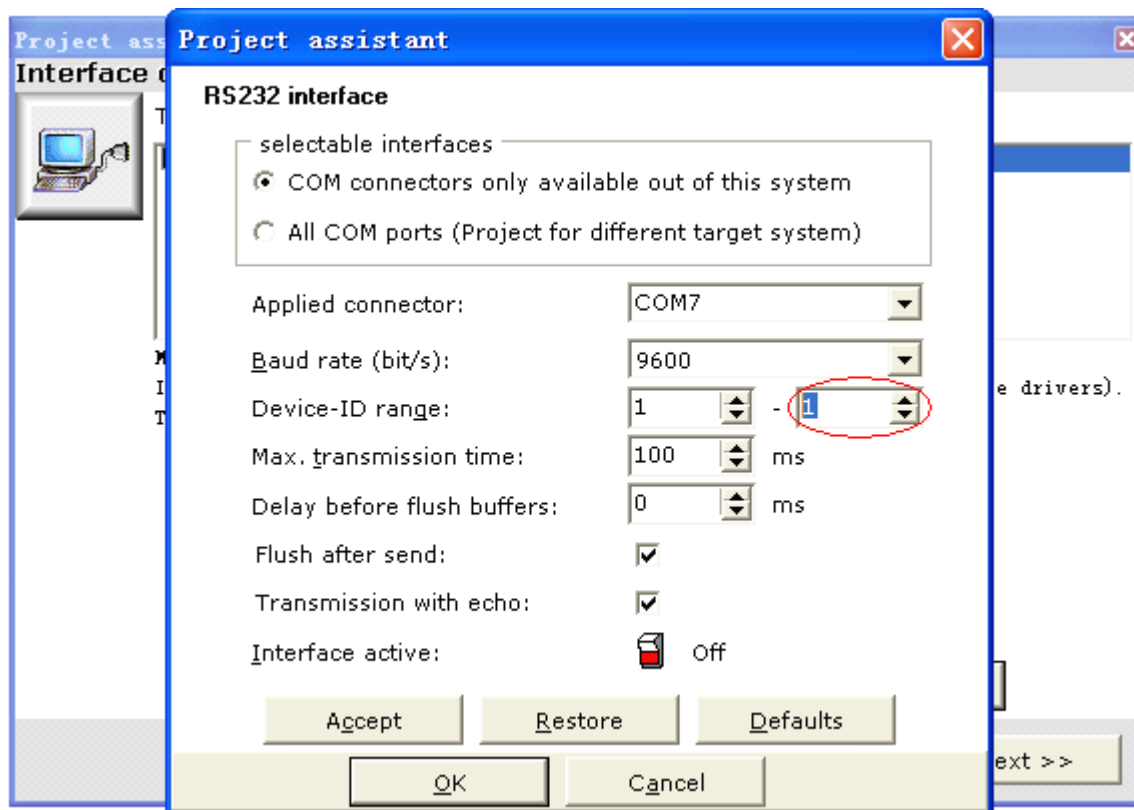


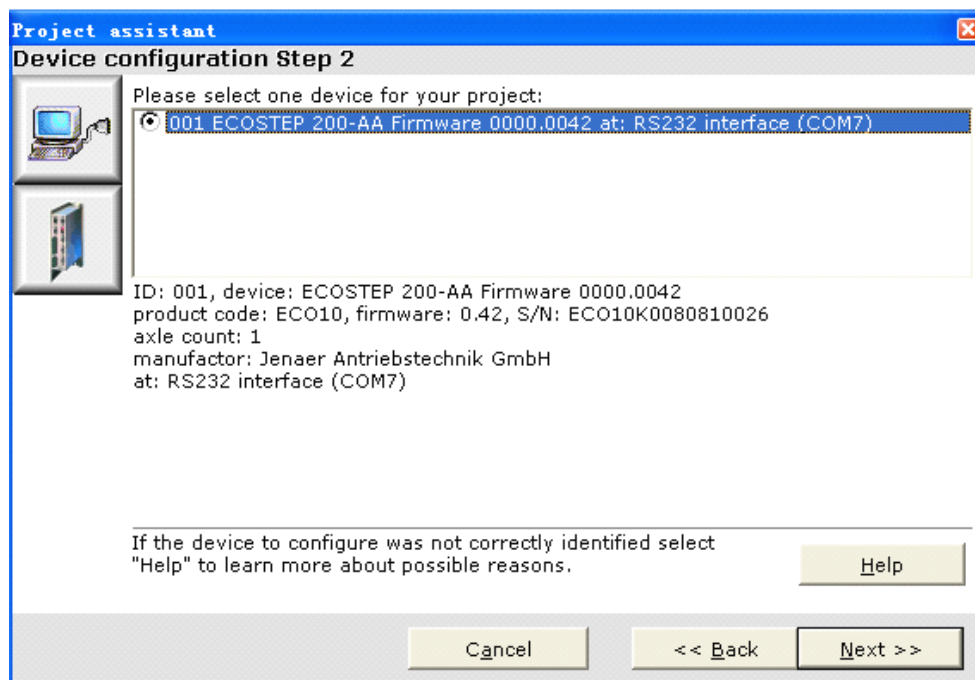
图 4.8 驱动器配置

建议采用自动搜索方式，您还可以选中所连接的通讯口后点击“MANUAL CONFIGURATION”会出现下面的画面”。在这里可以在红圈处输入连接在 **232、485、CAN** 网络上的伺服的最大地址即可，然后点击“OK”，然后再点击“NEXT”。如果不设置该项，那么软件会自动的从 1 到 126 去寻找伺服，以确保所有的伺服

都能连接上，如果您在这里输入伺服的最大站号，这样的话可以节省软件查找伺服的时间。



一旦系统找到了驱动器，就会显示出联线的驱动器型号以及驱动器内部的 **Firmware** 版本号，点击圆圈选中后会显示更详细的信息，例如驱动器 **SN** 序列号等，如下图。出现此画面证明通讯正常，你就可以进行下一步工作了，否则请查找通讯器件等原因。



如果是特殊的 **firmware**,系统是找不到驱动器的,系统会显示 **generic eco device**,点击选择即可,以下的连接可能会出现很多提示,此时不与理会,一直连下去即可。

6. 伺服电机配置

“NEXT”进入电机型号选择画面, 如下图:

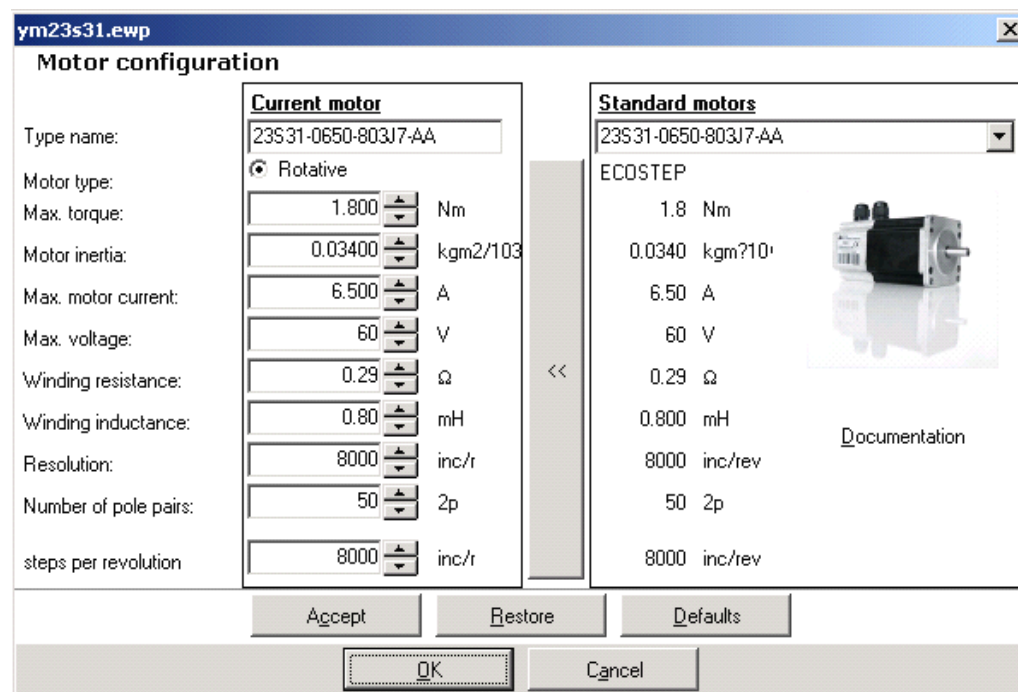


图 4.9 伺服电机参数配置

所配电机参数在出厂时已经设置好, 在这里直接按 **OK**, 无须选择右框中的伺服电机型号。

7. 确定机械参数

上一步完成后, 点击“NEXT”进入 “**Mechanics**”(机械连接类型)选择, 有两种类型, 一种为“**Rotative mechanic**”(旋转机械结构, 如齿轮等), 另一种是“**Linear mechanic**”(直线机械结构, 如同步带、丝杆等), 见下图:

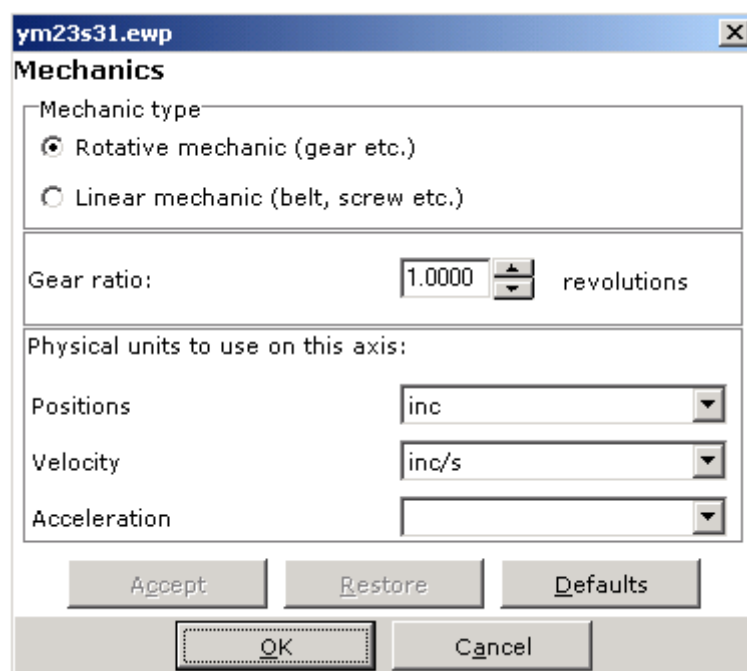


图 4.10 确定机械参数

8. 保存

上一步完成后，点击“NEXT”，进入的画面是请你输入你新创建的项目的文件名和路径，软件自动生成一个缺省的文件名和存储路径,你也可以修改为你喜欢的文件名和存储路径，如图：

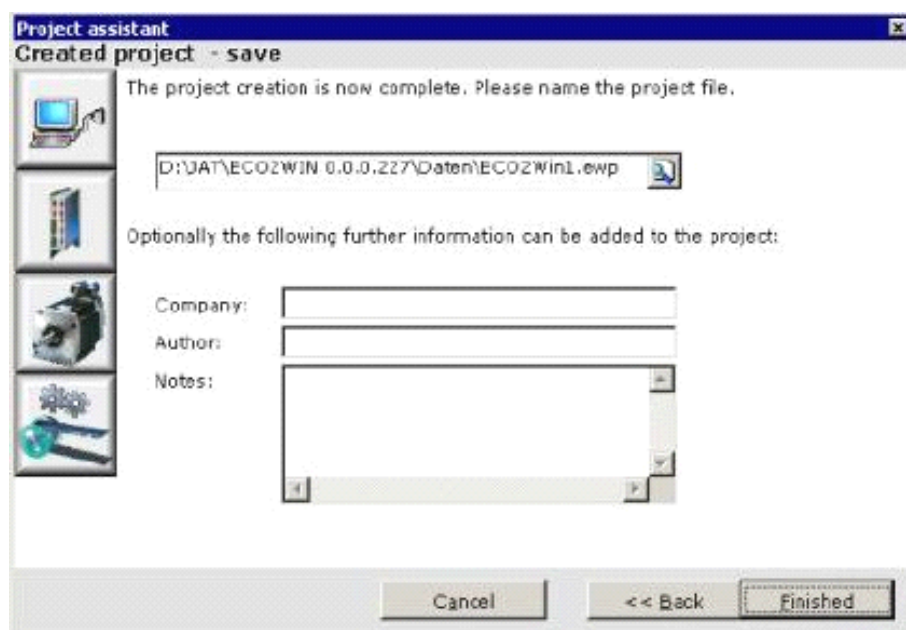


图 4.11 保存文件及存储路径

按“Finished”将完成新建工程项目，系统将弹出一个画面，询问是否进入“ONLINE”状态（在该状态下可以在线设置和监控伺服参数），选择“YES”系统将弹出如下画面：

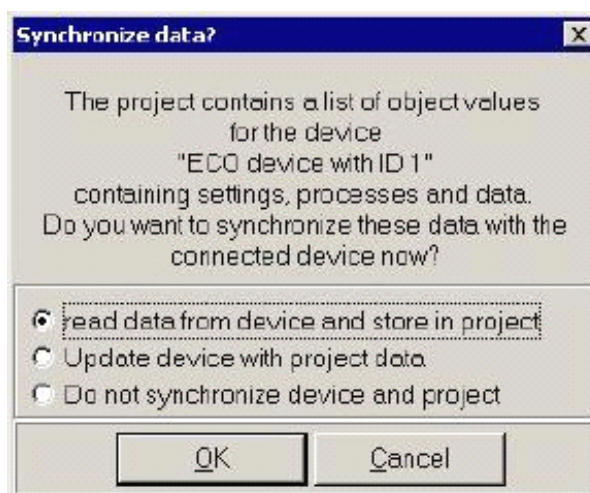


图 4.12

这里有三种选择：

- ③ “read data from device and store in project”，意思是从当前连接的 ED 驱动器内读取所有数据作为新建的项目的缺省设置(此选择一般用于从驱动器读取全部数据作为项目存档使用)；
- ③ “update device with project data”，用新建项目中的数据去更新驱动器内部的数据(此选择一般在项目完成后或调用旧项目之后使用)；
- ③ “do not synchronize device and project”，意思是不同步数据，系统只在需要时调出驱动器内参数并更新显示，一般选择此项。

完成上述创建新项目的过程后，画面进入工程主画面。

注意：前两项如果使用 RS232 通讯电缆编程，约需 2~5 分钟时间（依使用的 PC 机配置而定）

4.2.3 打开一个工程文件

在运行 ECO2WIN 之后，如果不选择“NEW”创建新项目，而是选择“OPEN”那么可以打开以前保存的工程文件，同样系统会被要求是否“ONLINE”和同步数据“Synchronize data”，如果用户判定此工程项目中有关“interface”、“device”、“motor”、“mechanic”四部分的设置与目前已经连接好的系统一致则可进行相关操作，否则进行数据更新时要十分小心。

4.2.4 主窗体介绍

当完成新项目设置或打开一个工程项目后，ECO2WIN 的主工作画面出现在屏幕上，见下图：

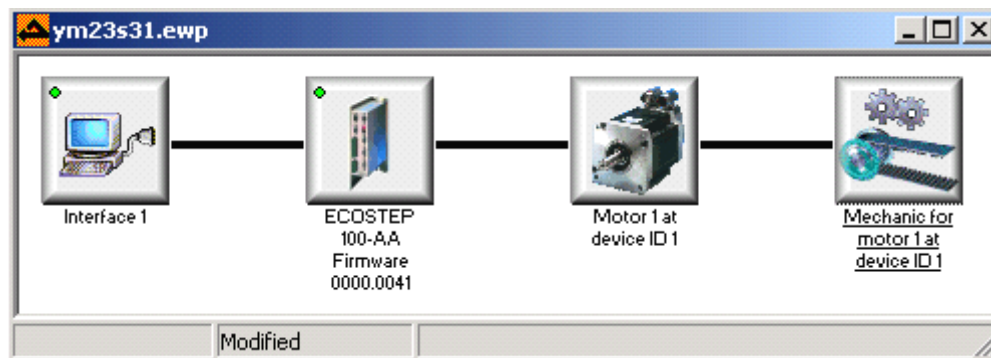


图 4.13

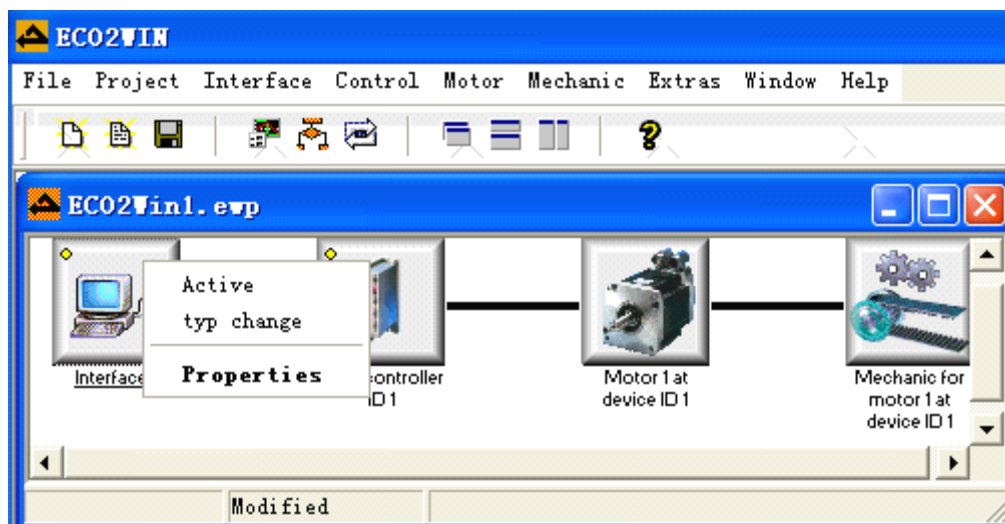
我们可以看到，主画面包含工程配置后的“interface1”、“device”、“motor”、“mechanic”四个图标，通过鼠标右键点击每个图标，就会弹出相应的菜单。

注意：对“Interface”，“ECO Device”和“Mechanic for Motor”的菜单里不同属性的修改，只有在保存和重新打开该工程文件后才有效(或者是使用 **Administrator\restart** 重新启动系统，请参考 5.9 节)。

4.2.4.1 “Interface”图标

尤其注意的是不要修改 interface 中的属性值，如果预先设置的是串口方式进行编程，那预先设置的波特率为 9600bps。如果是通过 CANopen 进行通讯，CANopen 接口的波特率的设置受通讯电缆的长度限制，标准设置为 1M Bauds。

鼠标右键点击“Interface1”，出现如下对话框：



点击“Properties”，进入属性菜单，如果是串口连接的，出现下图画面，这里你可以设置关于 RS232 通讯的相关参数，例如原来旧工程保存的串口号是 COM1,现在新连接的 PC 串口号非 COM1，则要在这一修改，设置好后点击“OK”确认。

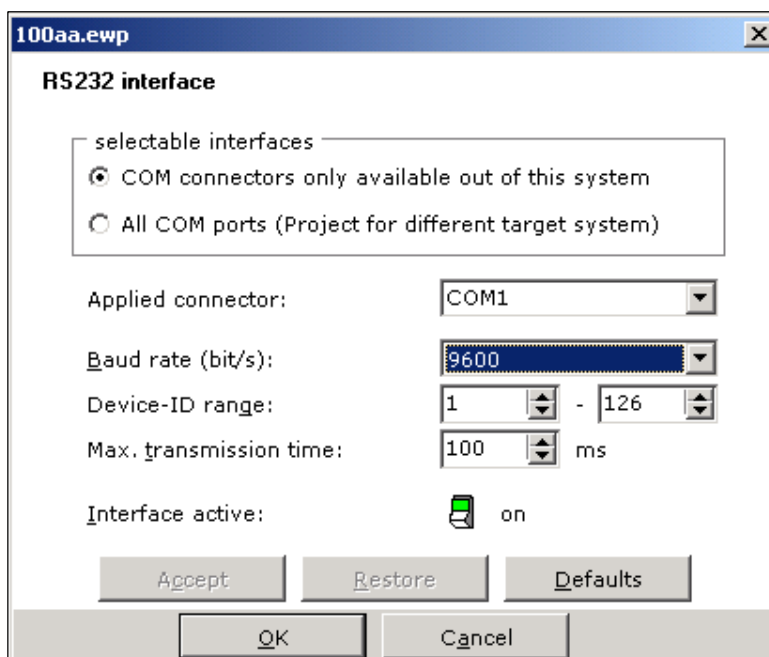


图 4.14

如果在建立该项目时选用编程所采用的通讯方式为 Peak CAN 适配器，那么点击“Properties”后，就会弹出一个 CAN 通讯参数设置窗体，用户可以修改波特率，站号搜寻范围等功能设置。

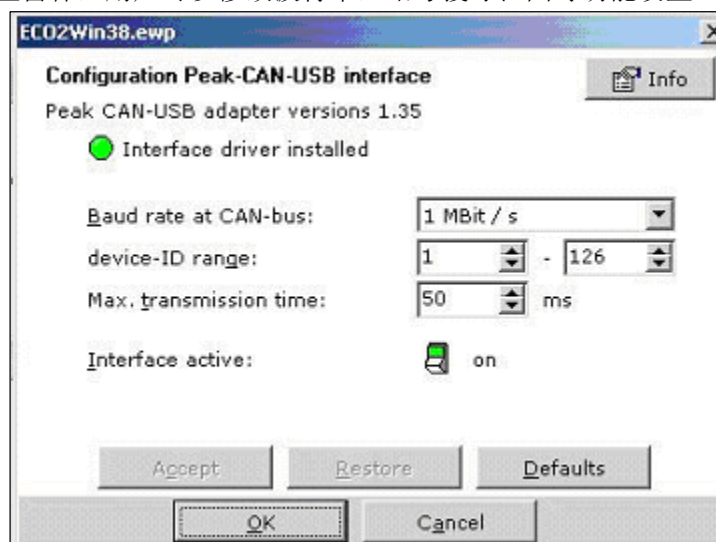
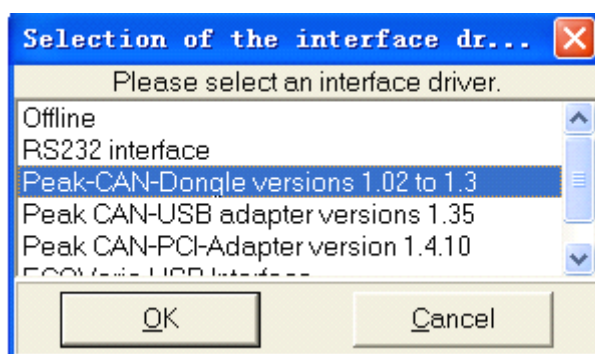


图 4.15

点击“Typ change”，进入通讯方式修改菜单。例如把串口通讯修改成 CAN 口通讯。如下图。



点击“Active”，可激活通讯口，此时图标左上角小原点变成绿色指示，反之黄色指示。

4.2.4.2 “ECO Device with ID”图标

“ECO Device with ID xx”图标内含有 ED 驱动器的主要功能，点击鼠标右键后弹出如下的菜单。

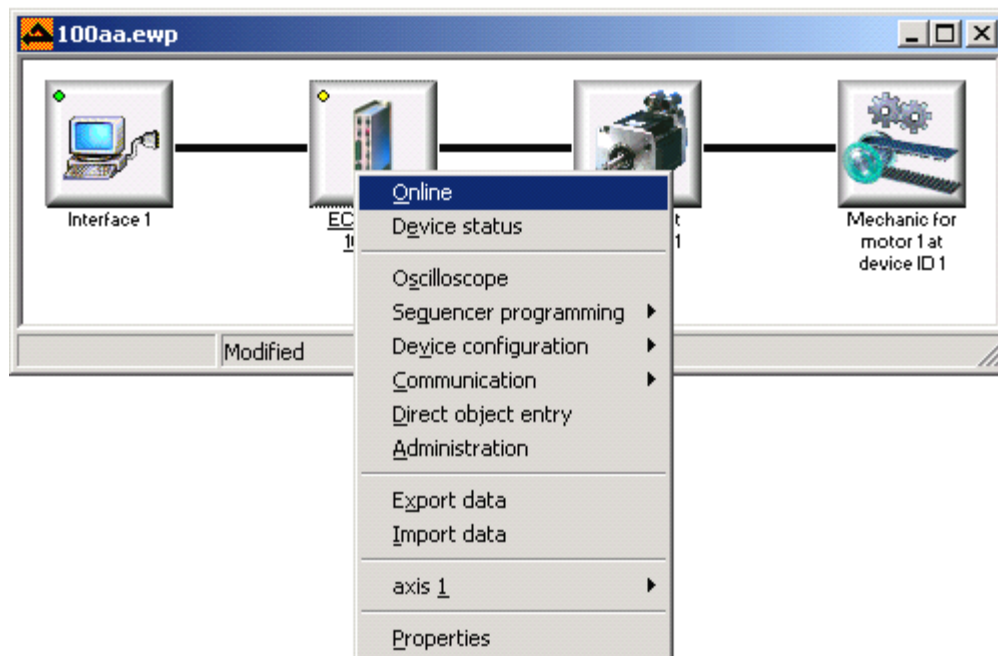


图 4.16 “ECO Device with ID”菜单

“ECO Device with ID”菜单功能描述：

名称	说明
Online	激活 Online 后，即可建立和 ED 驱动器的通讯。也可进行驱动器所有参数上传下载操作。详细见建立一个新的工程文件时最后一步操作询问是否“Online”一致
Device status	信息窗口，包含了软件，驱动器等信息，与用户操作无任何关系
Oscilloscope	内置的一个示波器软件，可以采集 4 路数据，每路 1000 个数据点，且采集最小分辨率达到 1ms（详细见 5.8）
Sequence programming	对驱动器进行编程，这些内部程序可以在驱动器内部或外部调用（详细见 5.5.1）
Device configuration	定义 I/O 口功能、定义主/从和模拟量输入功能(详细见 5.6)
Communication	Can 通讯参数设置(详细见 5.7)
Direct object entry	直接读写驱动器内部数据对象，可以通过 index 和 subindex 直接访问这个地址表格
Administration	Administration 窗体可以完成诸如保存，初始化或重新上电驱动器的功能。（详细见 5.9）

Export data	保存 ED 驱动器内部数据为一个 ewv 后缀名的文件。
Import data	从 ewv 文件中下载数据到 ED 驱动器内。
Axis1: Device Control	控制电机简单动作，正反转操作，原点操作，报警信息监控等(详见 5.3)
Axis1: Parameter settings	设置驱动器控制环参数，电机参数等（详见 5.4）
Properties	当前工程中伺服驱动器信息窗口，可以改变配置的驱动器型号（但一般不建议你修改）。一般用于修改站号，从而使同一个工程连接或者下载给不同站号的驱动器（但必须确保这些驱动器应用是一样的）。(见下图)

表 4.1 “ECO Device with ID”菜单功能描述

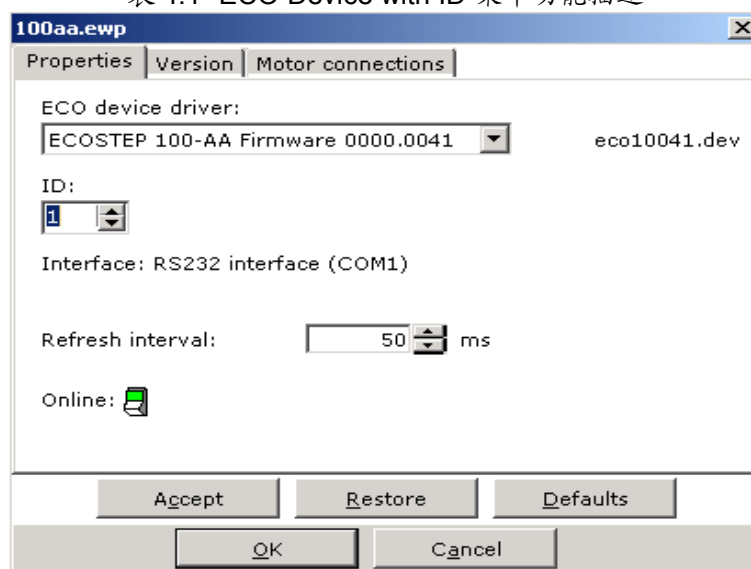


图 4.17 Properties 窗口

上述窗体中的 ID 为建立新的工程文件时系统自动识别到的驱动器上拨码设置地址，如果人为修改了驱动器拨码地址，也要在这个窗体中修改 ID 一栏的内容等于当前驱动器拨码地址值。驱动器上共有 4 位 ID 拨码，如图：



S0 为最低位，S3 最高位，On = 1，Off = 0

例如：S3=Off, S2=Off, S1=Off, S0=On, S3 S2 S1 S0 = 0 0 0 1 = 1

S3=Off, S2=Off, S1=On, S0=Off, S3 S2 S1 S0 = 0 0 1 0 = 2

除此之外在驱动器内部还提供一个内部对象地址设置 2F800008 设置 ID 偏移，此参数默认为 0，如果设置，则此时驱动器实际 ID 号=2F800008 设置值+驱动器上 4 位硬件 ID 拨码号。

例如：S3=Off, S2= On, S1=Off, S0=On, S3 S2 S1 S0 = 0 1 0 1 = 5

如果未设置 2F800008，默认=0。此时驱动器实际 ID 号=0+5=5。

如果设置 2F800008 =10。此时驱动器实际 ID 号=10+5=15。

4.2.4.3 “Motor xx at Device with ID yy” 图标

鼠标右键点击该图标后，弹出下面窗体，该窗体和建立一个新的工程文件时设置“Motor”时见到的一致。电机参数在出厂时候已经设置好，建议不要修改里面的数据。

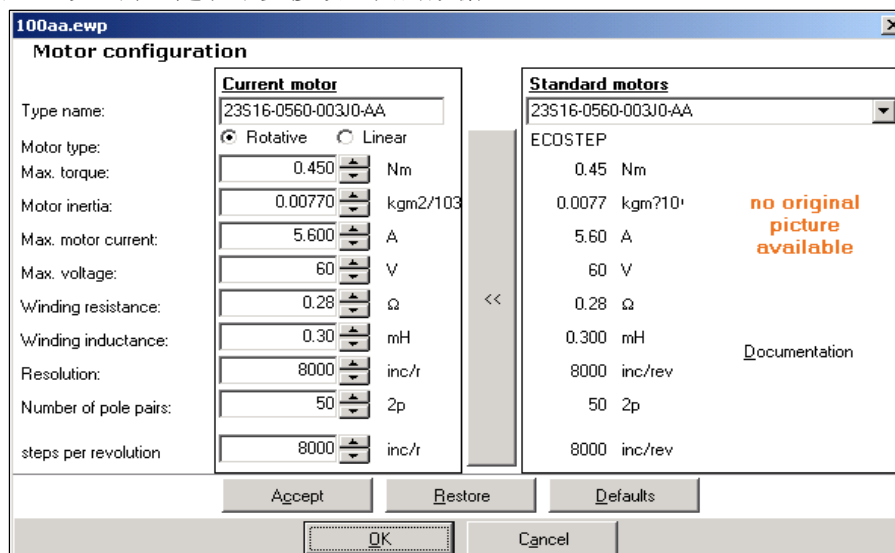


图 4.18 电机设置窗口

4.2.4.4 “Mechanic for Motor” 图标

鼠标右键点击“mechanic”后，弹出下面窗体，该窗体和建立一个新的工程文件时设置“Mechanic”时见到的一致，一旦输入了这些数据后，整个机构的丝杠导程、齿轮比等数据也就确定下来。最下面的 3 个输入栏确定位置、速度和加减速的物理单位。

建议初学用户也不要在这里修改任何单位等参数，以免程序中混淆不清。

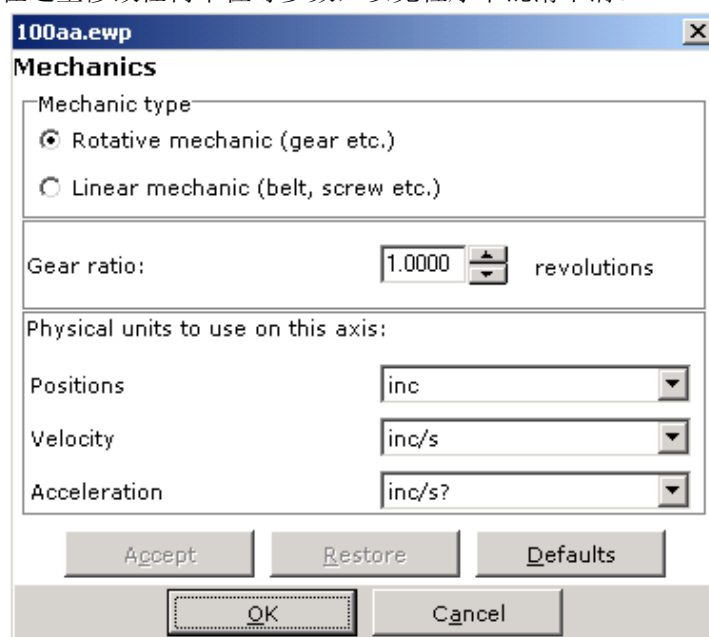


图 4.19 机械结构设置

4.2.5 导入和导出系统文件

利用 New 和 Open 建立和打开的文件只是一个所谓的工程文件，其内包含建立一个工程文件时设置的通讯、驱动器、电机和机械连接机构参数等内容，但并不包含内部 PID 参数以及 Sequence 等大部分内部数据，所以在您完成了一个伺服项目后，要想完整的保存驱动器内的参数，就需要使用 Eco2win 软件提供的 Import(导入)和 Export(导出)功能，Export(导出)的目的是将驱动器内的所有数据备份成一个以“ewv”为后缀名的系统文件，默认的文件名为 values.ewv。Import(导入)的目的是将用 Export 导出的系统文件重新导入到伺服驱动器内，这两个功能对保存伺服驱动器内部数据非常有用。

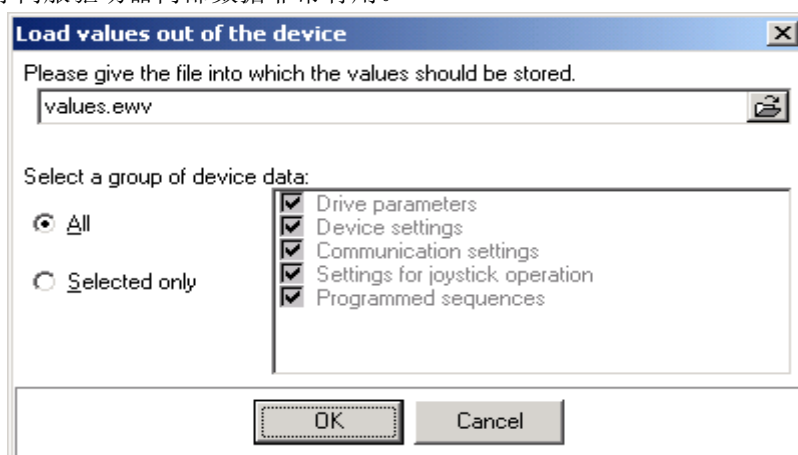


图 4.20 Export

在导入和导出数据时可以选择所有的数据，也可以只选择部分项目进行操作。

除外，我们还提供了一个专用的软件 ECO2LOAD 对系统数据进行导入和导出以及保存、初始化和重启动系统硬件等功能，建议使用该软件进行驱动器内数据的导入和导出。

详细说明请参照附录进行操作。

第五章 ECO2WIN 基本功能介绍

5.1 对象词典（Object dictionary）

如上文介绍，ED 伺服的主要一种控制方式就是由内部编程或者直接设定对象字典等方式规划好的运动曲线来完成的，在 ED 伺服内部使用了一种严格定义的对象列表，我们把它称作对象辞典，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）类似我们常说的内存地址，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。我们要做的是给需要的对象赋值或者是读取驱动器内部对象的状态，这种控制方式对初学者来说会有点麻烦，您可能要先通读本手册，在熟悉了常用对象的意义后，它又能给您带来最大的自由度。

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。

所有的对象都由 4 个字节组成。地址格式：

Index	Sub	Bits	属性	含义
例如：6040	00	16(=0x10)	RW	设备状态控制字
2509	00	8(=0x08)	R	主从电子齿轮比参数设置
2509	02	32(=0x20)	W	映射的从轴速度
2509	03	16(=0x10)	MW	电子齿轮比分子
2509	04	16(=0x10)	MW	电子齿轮比分母
2509	05	8(=0x08)	W	齿轮模式

说明：Index(索引地址)，范围 0x1000-0xFFFF（0x 表示 16 进制）

Subindex(Index 地址的子地址) 范围 0x00-0xff

Bits ,地址内数据长度，以 16 进制表示：

8 —数据长度为 1 个 byte=8 位

10—数据长度为 2 个 byte=16 位

20—数据长度为 4 个 byte=32 位

如果了解 ED 伺服提供的对象可以参考第 11 章“常用对象列表”。

5.2 对象操作表（Direct object entry）

鼠标右键点击“ECO Device with ID”图标弹出的菜单中选择“Direct object entry”，就可以出现：

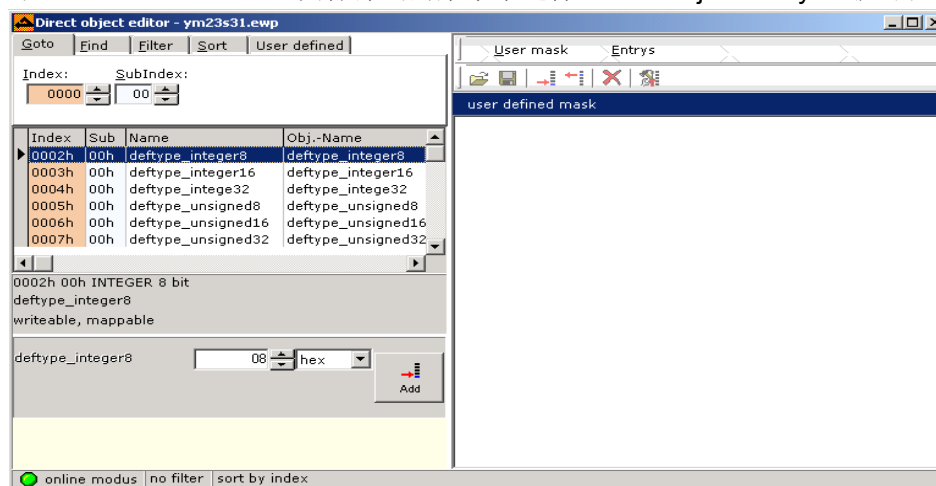


图 5.1 Direct object editor

在 Index 和 SubIndex 中分别输入 Index 和 SubIndex 地址,按下“Enter”键后,就可以直接跳到该对象处,并在 Name 一栏显示对象的名称,在窗体的左下脚显示对象数据长度、属性和数值。

如果这个对象属性为 W,那你可以直接在右边的数值框内直接输入数据(这种操作仅仅针对熟练的 ECO2WIN 编程人员,并且清楚要修改的对象对驱动器造成的影响)。

5.3 设备控制（Device control）

Device Control 是 Eco2win 软件内一个功能模块,可以完成伺服的在线设置和状态监控。在鼠标右键点击“ECO Device with ID”图标弹出的菜单中选择“Axis 1->Device control”,弹出该功能窗口。如果系统已经正常连接了电源,伺服驱动器上电后默认控制字(60400010)为 0x06,设备状态字(60410010)为 0x0031,工作模式(60600008)为 0。

注意：驱动器上电后,默认给控制字(60400010)写入 0x06,使伺服初始化。但是如有以下两种情况,驱动器控制字(60400010)默认初始化值被改写,伺服就无法完成初始化操作,从而也就无法动作。

- 1, 如果用户设置上电即调用程序,且程序段中把控制字(60400010)写入非 0x06;所以用户一定需要注意这一点.上电调用的程序一定要先对控制字写入 06!
- 2, 如果用户使用总线通讯时候,上位主站控制器等在通讯初始时候会对从站伺服所有配置的可写对象全部初始化为 0。所以使用总线通讯时候在主站初始化完成后,必须先对伺服从站控制字(60400010)设置为 0x06,让伺服正确完成初始化才能正常操作。

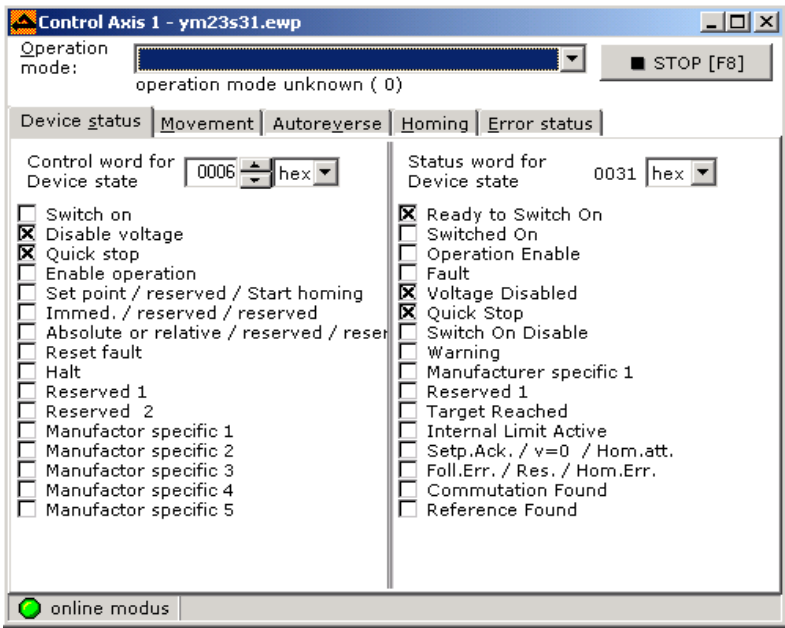


图 5.2 Control Axis 1

5.3.1 工作模式（Operation mode）

在 Device Control 窗口中的上方，有一个工作模式选择下拉框（Operation mode，见图），在该下拉框中，ECO2WIN 提供了下面多种模式选择：

模 式	含义	数值
1	绝对/相对位置模式	60600008=1
3	带位置控制的速度模式（也称作“带加减速的速度模式”）	60600008=3
-3	不带位置控制的速度模式（也称作“立即速度模式”）	60600008=-3
-4	主编码器控制模式	60600008=-4
4	力矩控制模式	60600008=4
6	原点控制模式	60600008=6
7	使用 CANopen 做插补运算的控制模式	60600008=7

表 5.1 工作模式定义

“工作模式”这个对象，在驱动器定义的“Object Dictionary”中对应的地址为：60600008，属性为 MW，修改 60600008=1，就相当于将驱动器置于“绝对/相对位置模式”。你可以通过“Device control”界面来修改这个值，也可以通过“Sequence programming”来修改，当然还可以利用 ED 提供的通讯接口，使用上位机（如 PC、PLC、单片机等）用通讯的方式来读写 60600008 这个地址对象。

个别的工作模式，如 4 和 7 是不能通过“Device Control”中的“Operation mode”下拉框来选择，只有

通过直接和间接（通讯）方式来修改（或者采用 Sequence）。

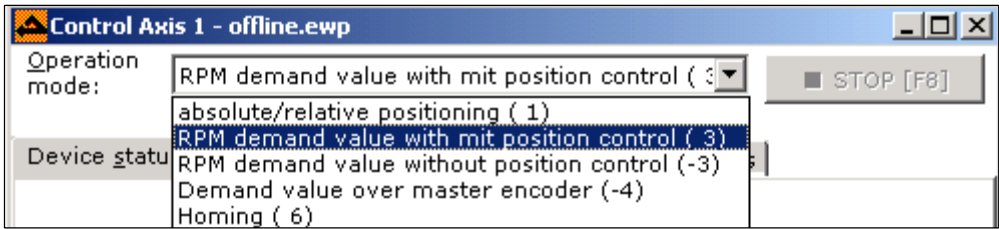


图 5.3 选择工作模式

下面简单介绍下各种工作模式：

模式 1：绝对/相对位置模式

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＝450，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＝1150 的位置。

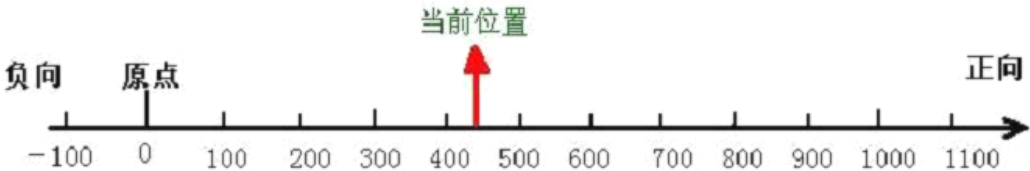


图 5.4 绝对/相对位置

在该模式下要定义的对象有：

对象	名称	值	含义
60600008	Chosen Mode of operation	1	设定工作模式为绝对/相对位置模式
60810020	Profile velocity	用户设定	最大速度
60830020	Profile acceleration	用户设定	加速度
60840020	Profile deceleration	用户设定	减速度
607A0020	Target position	用户设定	目标位置
60400010	Control word for Device	3F	绝对位置运动时的电机开始运动
	Status	先 F 后 5F	相对位置运动时的电机开始运动

表 5.2 绝对/相对位置模式有关对象

更详细的内容请参考第 11 章“常用对象列表”中“模式及控制”和“目标对象”小节部分。

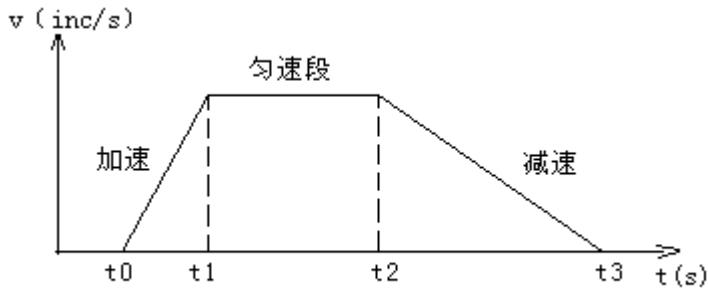
模式 3：带位置控制的速度模式（也称作“带加减速的速度模式”）

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：

$$V_t = V_o + at$$

$$V_t - \text{匀速部分速度}$$

Vo一起始速度
a - 加速度或减速度
t - 加速时间
 $S=Vot + (1/2) at^2$ S—加速段位移



模式3下的速度和时间曲线

在模式 3 下要定义的对象有：

对象	名称	值	含义
60600008	Chosen Mode of operation	3	设定工作模式模式 3
60FF0020	Target velocity	用户设定	目标速度
60830020	Profile acceleration	用户设定	加速度
60840020	Profile deceleration	用户设定	减速度
60400010	Control word for Device state	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

表 5.3 速度模式 3 有关对象

更详细的内容请参考第 11 章“常用对象列表”中“模式及控制”和“目标对象”部分。

模式-3: 不带位置控制的速度模式（也称作“立即速度模式”）

该模式下，当赋予目标速度一个新的值后，电机会以新的速度立即运动，不同于模式 3 有一个可以定义的加减速。

在模式-3 下要定义的对象有：

对象	名称	值	含义
60600008	Chosen Mode of operation	-3	设定工作模式模式-3
60FF0020	Target velocity	用户设定	目标速度
60400010	Control word for Device status	F	锁紧电机轴，如果“目标速度”值不等于零，电机将按照速度设定值运动

表 5.4 速度模式-3 有关对象

更详细的内容请参考第 11 章“常用对象列表”中“模式及控制”和“目标对象”部分

模式-4: 主编码器控制模式

该模式下，电机的运动直接受控于来自于驱动器 X7 口的外部编码器、如果接受外部的编码器（RS422 格式）就可以将驱动器设定为主从控制，驱动器做从站，电机轴做为从轴跟随 X7 口编码器主轴信号做跟随运动，跟随的速比可以通过设定电子齿轮比设定，见下表。。

在模式-4 下定义的对象主要有：

对象	名称	值	含义
60600008	Chosen Mode of operation	-4	设定工作模式模式-4
25090220	Mapping Slave velocity	60FF0020	映射 X7 口的数据值为目标速度
25090310	Gear factor	用户设定	电子齿轮比分子
25090410	Gear divider	用户设定	电子齿轮比分母
25090508	Gear mode	1	设置为主从模式，信号 4 倍频
60400010	Control word for Device Status	F	电机使能，电机轴锁紧。

表 5.5 控制模式-4 下相关对象

更详细的说明，请参考 5.6.3“电子齿轮比设置（Electronic gear）”一节。

模式 4: 力矩控制模式

该模式下电机将恒扭矩输出，输出扭矩决定于设定的“目标扭矩”值（Target torque）。实现该模式控制，你必须依据下面的操作顺序：

1. 利用 ECO2WIN 的“Direct object editor”或者“ Sequence”（参考 5.2）设定 60600008=4。
2. 设定 607100 为期待的电流值，对 ED100 可以最大设定为 8A，ED430 和 ED620 为 12A，ED630-0250 为 18A，ED630-0300 为 21A，ED630-0400 为 24A，
3. 在 Device control 中或者“Sequence”设定 60400010=6（或在“Device control”中直接设置），然后锁紧电机轴，也就是设定 60400010=F。

警告！在锁紧电机轴（第 3 步骤）前，务必注意驱动器由于是恒定力矩输出，电机的速度将只受“目标扭矩”值限制，所以在动作前请务必保证你的负载已经正确安装可以正常使用，并且要注意设定最大速度，切记扭矩控制模式下最大速度的限定必须通过“Parameters setting->Current controller”中的“Max. Current”来设置。

模式 6: 原点控制模式

模式 6 为原点模式，在该模式下驱动器工作在查找原点状态。ECO2WIN 软件提供 27 种原点方式，详细

的介绍见第 7 章“限位开关与原点控制”。

在模式 6 下要定义的对象有：

对象	名称	值	含义
60600008	Chosen Mode of operation	6	设定工作模式模式 6
60980008	Homing method	用户设定	原点方式，默认为 6
607C0020	Relocation zero point	用户设定	相对于原点的偏移量，默认为 0inc
60990120	Veloctiy during search for Reference swith	用户设定	寻找原点开关信号的速度,默认为 15625inc/s
60990220	Veloctiy during search for Reference set point	用户设定	寻找电机 index 信号的速度,默认为 1562.5inc/s
609A0020	Acceleration during homing	用户设定	在查找原点过程中的加速度 160000 inc/s ²
60400010	Control word for Device Status	先 F 后 1F	模式 6 下开始寻原点

表 5.6 原定模式下相关对象定义

定义这些对象既可以通过“Direct object entry”窗口直接输入，也可以通过“Device control”窗口中的“Homing”页输入，当然也可以在或者“Sequence”中直接设定

5. 3. 2 控制命令和状态显示

伺服电机与伺服驱动器组成的伺服系统有不同的工作状态。在机械工程里对空间坐标进行了描述：确定系统在不同时间的不同坐标(s(t), v(t), t)，这对我们的伺服系统也是一样的。伺服系统会提供给你电源是否开启、驱动器的故障或目标位置是否到等信息。ED 伺服系统采用 CANopen 国际标准定义内部对象，同样定义了伺服系统状态字和控制字，状态字代表驱动器当前状态，控制字用来改变驱动器状态。

在“Device Control”窗口的 Device status 一栏的左半部分为输入控制命令部分，在文本框中可以输入不同的控制命令，实现对伺服的运动控制目的。控制命令在对象地址表中定义地址为：60400010，这是一个具 16bit 的字，赋予该字不同的值，也就是等同于实现不同的控制动作，常见的控制字的值有：

值	命令
0x06	电机断电，设置驱动器为准备状态
0x0F	电机上电,电机轴处于锁紧状态
0x3F	绝对定位

0x5F	相对定位
0x0F	开始运动
0x1F	原点模式下开始寻找原点
0x80	复位故障

在“Device Control”窗口的 Device status 一栏的右半部分为状态显示区，ED 驱动器使用一个状态字 60410010（16bit）表示当前驱动器的状态，该状态字的每一位都代表了驱动器的一种状态，如 bit0 代表“准备上电”（“Ready to Switch on”），bit15 代表“原点找见”（“Reference Found”）。

位序号 (bit)	位(bit)	含义	对应状态字值
0	Ready to Switch on	准备上电	60410010=0x0001
1	Switched On	已上电	60410010=0x0002
2	Operation Enable	操作使能	60410010=0x0004
3	Fault	故障	60410010=0x0008
4	Voltage Disable	电压输出禁止	60410010=0x0010
5	Quick Stop	快速停止	60410010=0x0020
6	Switch On Disable	上电禁止	60410010=0x0040
7	Warning	警告	60410010=0x0080
8	Manufacturer specific 1	制造商指定 1，内部 保留	60410010=0x0100
9	Reserved 1	保留 1	60410010=0x0200
10	Target Reached	目标位置到	60410010=0x0400
11	Internal Limit Active	内部限位激活	60410010=0x0800
12	Setp.Ach./v=0/Hom. att.	脉冲响应	60410010=0x1000
13	Foll.Err./Res.Hom.Er r.	跟随误差/原点错误	60410010=0x2000
14	Commutation Found	找到电机励磁	60410010=0x4000
15	Reference Found	找到原点	60410010=0x8000

表 5.7 状态字每个位功能说明

通过状态字，我们可以立即判断伺服系统当前状态，该状态字不但可以利用 ECO2WIN 软件的“Device Control”窗口直接观察到，也可以利用 Sequence 编程以及 ED 提供的通讯接口来读取，从而在软件内部和外部

实现对 ED 的监控。

5.3.3 设置运动参数

在“Device Control”窗口的“Movement”页，见下图，可以设置位置和速度模式下的运动参数，这些参数可以在线修改。这里列出了 ED 伺服速度模式，位置模式下的相关运动参数，用户设置这些参数后，在工作模式选择下拉框（Operation mode）中设定工作模式，然后再在“Device Control”窗口的 Device status 一栏的左半部分输入不同的控制命令，就可以实现对伺服的运动控制。

在不同位置和速度模式下，具体要设置哪些运动参数请参考 5.3.1 工作模式介绍。需要说明的是该窗口中的“Min. Software position limit”（607D0120）和“Max. Software position limit”（607D0220）两个参数栏，分别设置的是电机正向(默认为逆时针方向)和反向极限位置（默认为顺时针方向），例如你设定了 Min. Software position limit=-160000inc, Max. Software position limit=320000inc，那在位置模式 1 下（这是内部软限位，只在位置模式下有效，其他控制模式请用硬件限位），当设定的目标位置（607A0020）超出了上述的两个参数设定的范围后，电机运行到这两个位置值时将不在动作。默认是零，表示没有限制。

如果要修改电机编码器的计数方向，可以通过修改 position polarity 和 velocity polarity 改变，具体请参考 5.4.5 一节。

这些运动参数也可以利用 Sequence 编程以及 ED 提供的通讯接口来读取，从而在软件内部和外部实现对 ED 的监控。

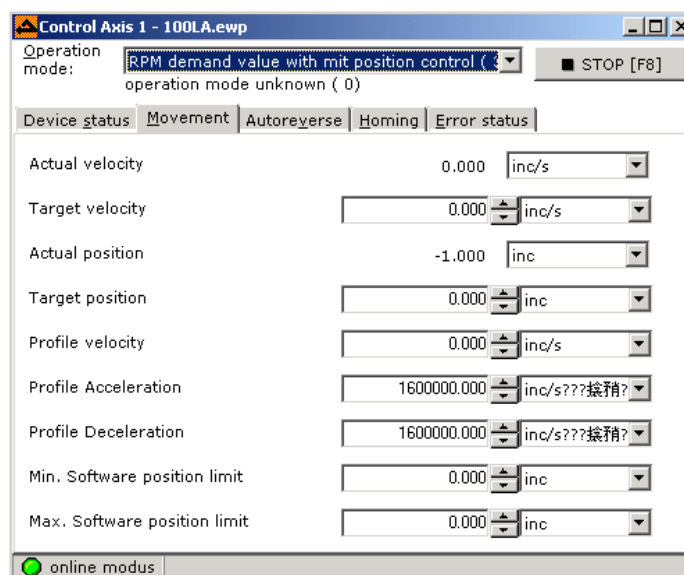


图 5.4 “Movement”页参数设置

5.3.4 原点控制

在“Device Control”窗口的“Homing”页，见下图 5.5，可以设置原点模式下的原点方式和其它有关参数，并设置电机查找原点动作的开始和停止。需要说明的是右边的“Time limit for homing”可以设置整个查找原点过程的最大时间，如果在这个设定的时间内，没有成功找见原点，系统将报“No reference”(原点无法找见)，右下脚的第 2 个指示灯变黄，如果在指定时间内成功找见原点，右下脚的第 2 个指示灯变绿,文字提示“Found reference”。这些运动参数也可以利用 Sequence 编程以及 ED 提供的通讯接口来读取，从而在软件内部和外部实现对 ED 的监控。

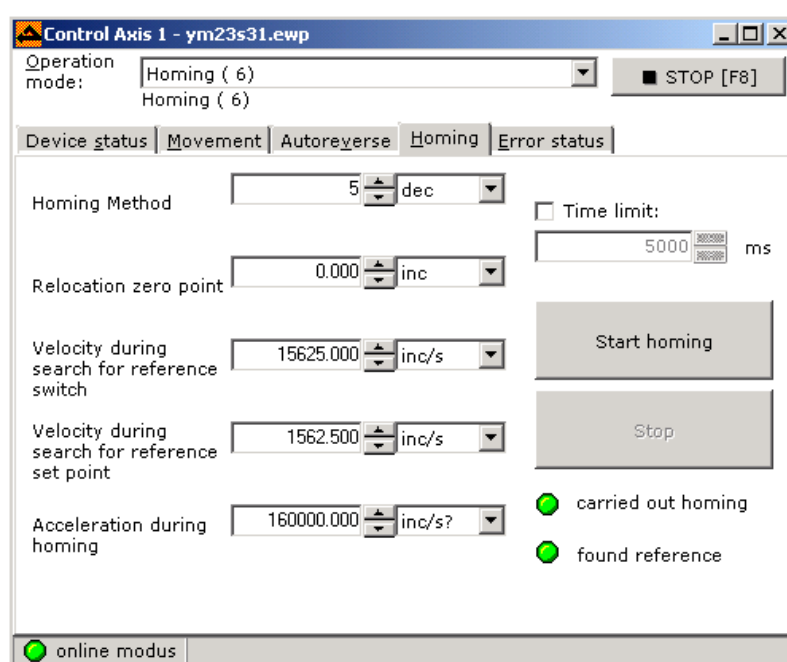


图 5.5 原点控制

5.3.5 自动正反转控制

在“Device Control”窗口的“Autoreverse”页，可以设置系统为不同的正反转模式，ECO2WIN 软件提供 5 种正反转控制方式，有：带位置控制的定时正反转、带速度控制的定时正反转、带速度控制的位置正反转、带位置控制的速度正反转和带速度控制的速度正反转，详细的介绍见第 8 章。这些运动参数也可以利用 Sequence 编程以及 ED 提供的通讯接口来读取，从而在软件内部和外部实现对 ED 的监控。

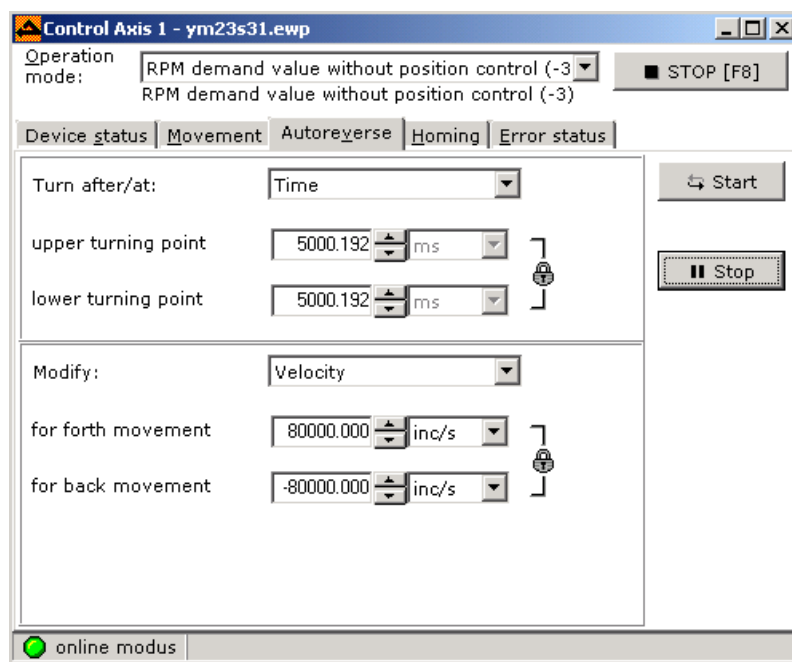


图 5.6 自动正反转设置

5.3.6 错误诊断

如果系统上电后，驱动器红色的“Error”LED 指示灯闪烁，表示系统有故障。

在“Device control”窗口的“Error status”页面用于监控驱动器发生故障时的故障状态，见下图。ED 提供一个状态字 26000220 表示驱动器的故障状态，该对象为 32bit 长的一个字，字的低 18 位，bit0~bit17,每位都代表一种故障，详细说明见附录“错误信息”一节。这些运动参数也可以利用 Sequence 编程以及 ED 提供的通讯接口来读取，从而在软件内部和外部实现对 ED 的监控。

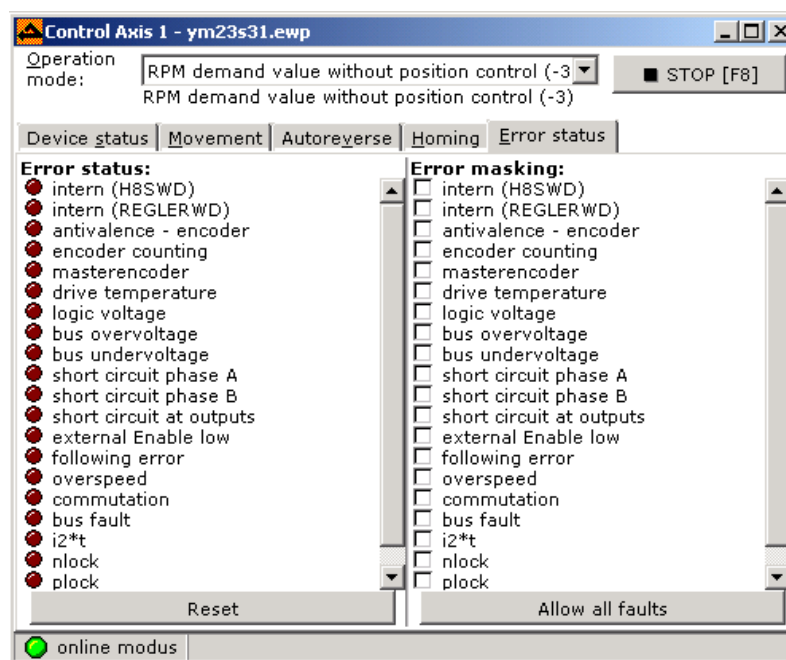


图 5.7 “Error status”故障诊断

窗口的左边显示的是当前故障状态，如果没有故障则全部显示为暗褐色指示灯，如果有故障，则相应的故障位指示灯会显示为红色，按下方的 Reset 按钮或在“Device status”页面的“Control word for Device control”一栏的控制字中输入 0x86，可以清除故障并将驱动器置于“准备状态”；窗口的右边为故障检测选择位，供测试时使用，出厂设置为全部检测，可以屏蔽相关故障让电机继续运转，如果你选择其中的一位（显示为“x”），则系统将不检测该单个故障。**切记，不要轻易选择取消，否则会导致由于系统不在检测这类故障而导致硬件烧毁。**

在此特别说明一下 $i2t$ 故障，该故障用来保护电机线圈过热，从而保护电机不会被烧掉。其包含两个参数，一个是最大电流值，一个是过热时间值，当实际电流超过设定值且持续时间三倍于过热时间设定值时，系统将产生 $i2t$ 故障。

如果电机故障复位成功，可以通过输入控制字 0x0F 重新锁紧电机轴。

详细报警信息请参考附录“报警信息原因及排除”

5.4 参数设置（Parameter setting）

伺服控制系统包含有位置环、速度环、电流环的控制，不同控制模式下电机相关运转参数的设置是不同的。位置环和速度环参数调节详情请参考 8.4 节伺服性能调整。电流环和励磁环参数出厂均已设置好，可参阅 6.2 章节核对。

5. 4. 1 位置环参数

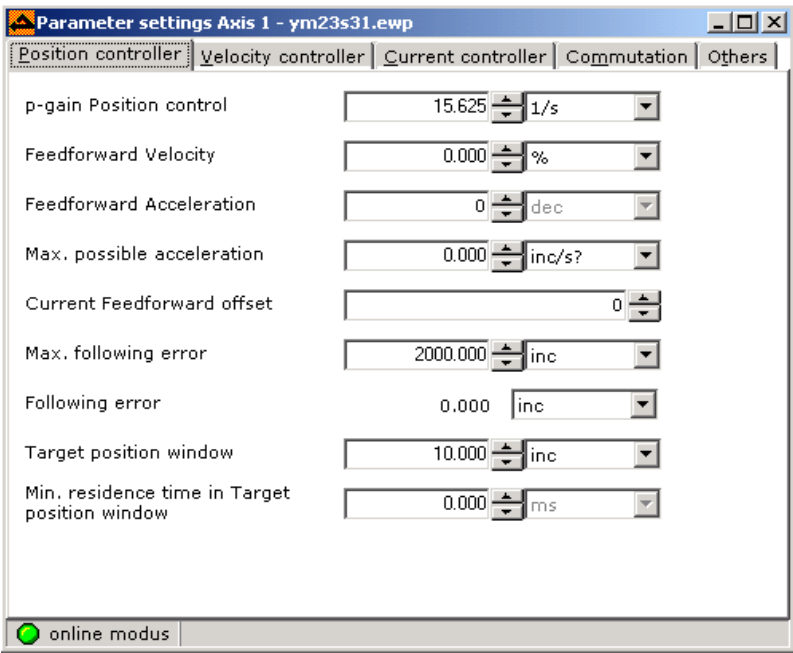


图 5.8 位置环参数设置

参数	参数说明	功能以及调试
p-gain position control	位置环增益	该值如果太大，会造成系统超调震动，尤其惯量比较大的情况下，但是如果太小，位置环响应慢，在跟随运动时尤其明显，因此要适当设置此值。
Feedforward velocity	前馈速度	此值主要在工作模式为 3 模式、1 模式和-4 模式时有用，主要用来减小 following error 。如果设置为 100%在急速停止的时候，过冲会很明显，如果设置为 0，在停止的时候类似 S 曲线。根据应用情况合适设置此值。
Feedforward Acceleration	前馈加速度	一般情况下不需要调整
Max.possible Acceleration	最大可能加速度	为了抑制大信号震荡，如果负载惯量超过电机转子惯量几十倍时候可以适当设置此值，可以参考 ECO2WIN 帮助里的 8.5.1.2
Current feedforward offset	电流前馈偏置	在特殊 firmware 内用到该值
Max.following error	最大跟随误差	根据工作情况设置此值，不要设置的太大，如果在不追求跟随精度的时候可以适当设置此值
Following error	实际跟随误差	实际的跟随误差
Target position window	目标位置窗口	当目标位置误差在此设置值内驱动器默认位置到

Min.residence time in target position window	目标位置到达 最小停留时间	目标位置到达后的停止时间
--	------------------	--------------

注：此画面设置位置环相关参数，其中位置环增益 Kpp（即 p-gain position control），前馈速度 Vff（Feedforward Velocity）和最大跟随误差(Max. Following Error)的设置的控制模式 1 和模式 3 时非常重要，跟随误差值的计算在电机正常运转时，可参照如下公式：

Following error=(V-V*Vff/100)/Kpp, V-Actual Velocity(计算时可以用 Profile Velocity 设定值)

所以当 Vff=0 时，Following error =V/Kpp, 必须确保 Following error 小于所设置的最大跟随误差，否则会因“Following error”而导致故障停机。

注意！ 系统对跟随误差的计算是通过设定的位置与当前位置比较得出。

当 Vff=100,即 100%前馈时，Following error =0，这时不考虑跟随误差了。

出于安全因素，驱动器出厂时设定的位置环参数比较小，如 Kpp=15.625 1/s, 用户可以根据采用的机械结构重新设定位置环的参数，一般情况，Kpp 都在 100-200 (1/s)之间，但这不是绝对的。

5.4.2 速度环参数

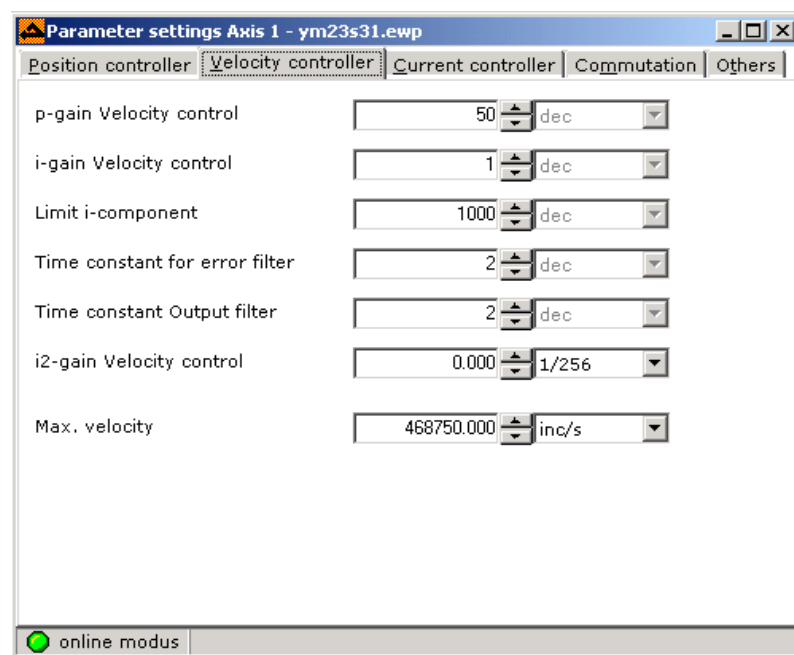


图 5.9 速度环参数设置

参数	参数说明	功能以及调试
p-gain velocity control	速度环比例增益	调节此值可以改善系统动态响应，尽可能的把此值设大。一般情况下，负载惯量越大设定值越大，但是设置过大电机会有噪音，可以通过设置 Time constant output filter 来改善，但是控制响应会

		慢。
i-gain velocity control	速度环积分增益	在低速的时候此值对系统作用效果比较明显，设置值不要超过 10
Limit i-component	积分增益限制	此值与速度环积分增益结合使用，原则上 $i\text{-gain velocity control} * \text{Limit i-component} > 1.3 * \text{actual max current}$ 。
Time constant for error filter	偏差滤波时间常数	1 对系统没有滤波，实际速度环增益= 速度环增益*偏差滤波时间常数。一般用在惯量比较大的场合，如果惯量太大需要提高动态响应必须把速度环增益设置大，但是设置大了电机会产生噪音，所以就运用此参数来达到同样的动态响应效果。
Time constant output filter	输出滤波时间常数	0 和 1 没有对系统滤波，如果设置此值可以减小一些噪音，但是会降低系统性能，必须适当设置否则系统不稳定
i2-gain velocity control	速度环积分增益（高分辨率用）	在高分辨率编码器上采用此参数,高分辨率设为 25dec 或 0.1（1/256）
Max.velocity	最大速度	速度的最大速度限制

5.4.3 电流环参数

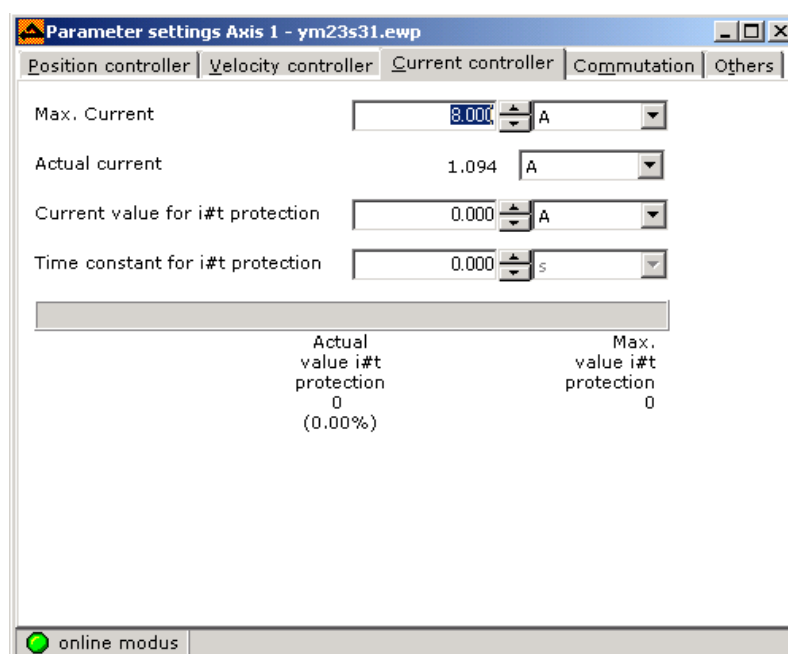


图 5.10 电流环参数

参数	参数说明	功能及调试
Max. current	最大电流	驱动器最大电流限制（根据电机额定电流设置，一般是三倍的电机额定电流）
Actual current	实际电流	实际电流
Current value for i## protection	i## 电流保护值	一般根据电机额定电流的 1.4 倍
Time constant for i## protection	i## 电流保护时间	电机过热保护时间一般 5-20S

本组参数出厂已经设置好，一般情况下用户无需修改。具体参数可参阅 6.2.2 电流参数

Acutal current 一栏显示当前电机相电流，通过其判断电机是否过流，往往还用于判断系统选择的电机是否过载。

5.4.4 励磁参数（Commutation）

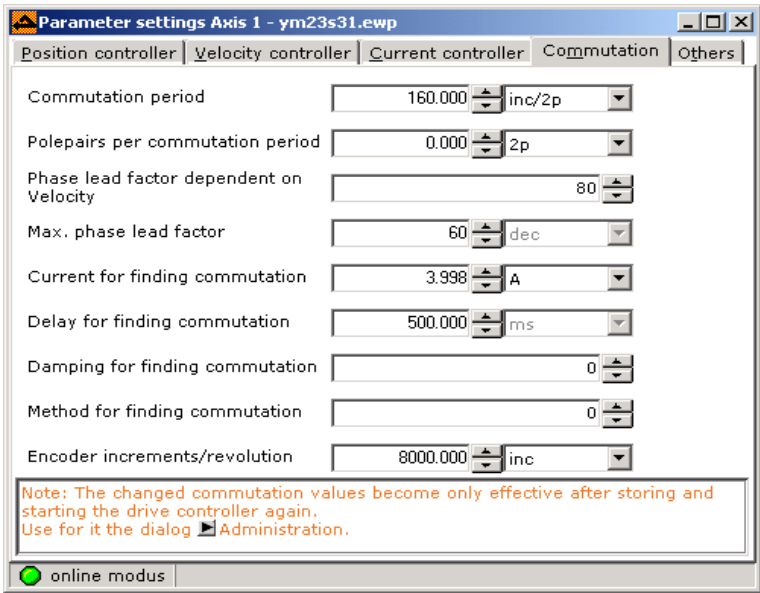


图 5.11 励磁参数设置

参数	参数说明	功能以及调试
Commutation period	励磁周期	一般设置为编码器 4 倍频后的分辨率
Polepairs per commutation period	电机极对数	电机极对数，参考电机选型手册
Phase lead factor dependent on velocity	基于速度的相位因数	该参数当二相电机在高速的时候有用，主要用来改变电流相位

Max.phase lead factor	最大相位因数	<p>如电机 23S21-0560-803j7-AA, 编码器分辨率=8000 inc/rev(inc/转), 极对数=50</p> <p>那么 commutation period = 8000/50 = 160</p> <p>polepairs per commutation period=0</p> <p>此时 Max.phase lead factor=60</p> <p>如果分辨率更高按比例计算, 如分辨率是 10000, 电机极对数是 2, 则 $160/5000 = 60/X$</p>
Current for finding commutation	找励磁的电流	找励磁时的电流, 如果带小惯量的电机, 如果电流过大容易找不到励磁。
Delay for finding commutation	找励磁的时间	寻找励磁时的时间
Damping for finding commutation	励磁阻尼	适当设置该参数可以减小找励磁时的跳动, 范围 (0-20)
Method for finding commutation	找励磁的方式	<p>3: 一般应用</p> <p>8: 上电不抖动应用 (硬件必须选配 ENC15-9)</p>
Encoder increments/revolution	编码器	两个 Z 信号之间的 AB 信号的个数

本组参数出厂已经设置好, 一般情况下用户无需修改。具体参数可参阅 6.2.1 电机参数

驱动器上电使能后 (设定驱动器控制字=F), 会提供一个电流让马达以一定速度转动一个很小角度从而确定马达位置状态, 如果正确则在 “Device control” 的 “Device status” 的 “Status word for Device state” (状态字) 中的 “Commutation found” 位选中显示。转动的角度大小决定于励磁参数中的 “Current for finding Commutation” 和 “Delay for finding commutation” 两个参数的大小。

用户工艺如果不允许上电转动小角度, 则可以按如下操作进行

1: 选配 KINCO 伺服选配件 ENC15-9 适配器!

2: 伺服编码器电缆不再选用九针 ED 伺服编码器电缆, 而是选用十五针 CD 伺服编码器电缆! 详细具体型号请参考 ED 与 CD 选型手册。例如 ENCE-03-KL 改选为 ENCCA-03-KL。驱动器, 电机, 电机电缆型号照旧。

3: 软件设置 Method for finding commutation=8, 保存参数重启即可达到上电不抖动效果!

5.4.5 其它参数

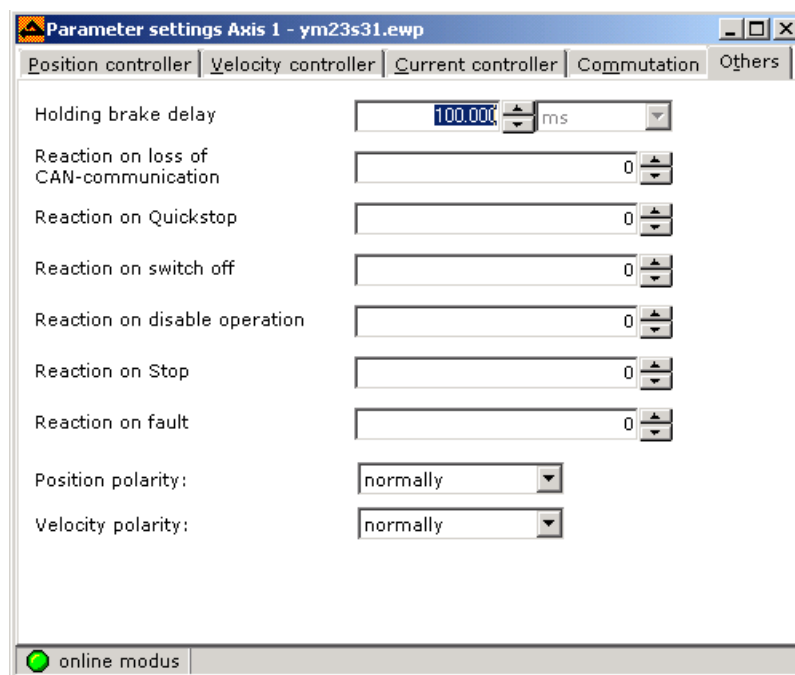


图 5.13

在这里，需要理解不同情况下系统反应设置：

Reaction on loss of CAN-communication(CANBUS 通讯中断并且 CAN 节点保护功能被启用时)：

设定值 0：没有动作

设定值 1：产生一个故障

设定值 2：控制字 BIT1 被复位

设定值 3：控制字 BIT2 被复位

其它值：没有动作

Reaction on quickstop(快速停止后反应)（驱动器由运行到快停，控制字的 Bit2 位被复位）

设定值 0：系统立即断开到电机的电流

设定值 1：按照系统轨迹运行减速度的减速至停止

设定值 2~4：按照快停减速度进行减速停止

Reaction on switch off

设定值 1：按照系统轨迹运行减速度减速停止

其它设定值：立即关断

Reaction on disable operation

设定值 1：按照系统轨迹运行减速度减速，并禁止运转

其它设定值：立即进入禁止运转状态

Reaction on stop 暂时保留

Reaction on fault

设定值 0:立即禁止驱动器

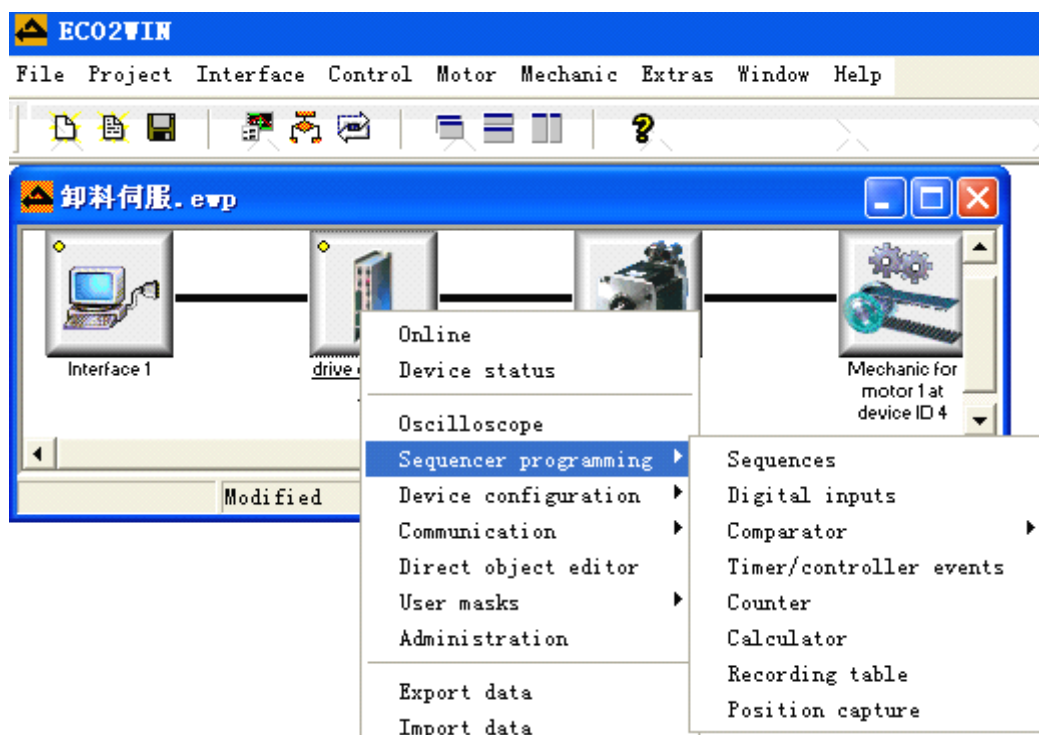
设定值 1: 如果可能先按轨迹运行减速, 再禁止

其它值: 如果可能先按快停减速, 再禁止

这里要强调的两个参数是: **Position polarity** 和 **Velocity polarity**, 驱动器出厂时设定的电机正向的计数方向为 CCW, 如果要修改这个出厂设定的方向, 可以通过这两个参数来修改。即这两个参数用于改变电机的转动方向。

5.5 内部编程

ED 内部编程包括七个部分, 如下图从上到下依次分别是程序编辑、数字量调用程序段设置、比较器、定时器/事件触发、计数器、计算器、记录表格、高速脉冲捕捉。内部编程是ED 驱动器强大控制功能集中的体现, ED 内部开辟出16.5KB 空间供256 段 程序段使用。编程过程简单快速、调用方式多样(输入触发、定时触发、事件触发、直接调用等、内部提供的 逻辑比较、数学运算、数据记录等功能都可在程序段中实现, 下面我们将分别加以介绍。



5.5.1 顺序编程 (Sequences)

打开sequences 编程窗口, 弹出如下画面:

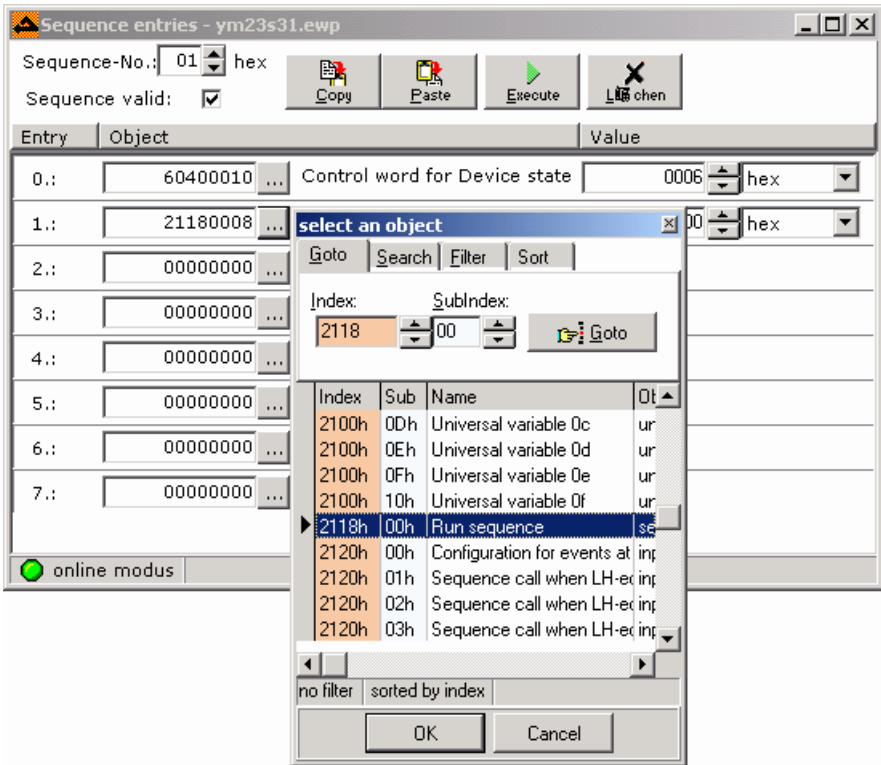


图 5.14 Sequence programming

从图中可以选择Sequences-No(从00~FF), 如果想让此段子程序有效, 必须将Sequence valid 打上勾, 编程过程非常简单, 左边选择准备赋值的对象的 Index 和 Sub index, 右边填入数值即可, 每段最多可以设置 8 个对象, 如果对 0x21180008 的寄存器赋值 xx (00~FF), 则可从该 Sequence 跳转到第 xx 段 Sequence 中。

顺序编程就是赋值给一些类似批处理程序一样可以顺序执行的对象。对象就像我们常说的内存地址, 其中一些对象可以由用户设定, 如速度值、位置值等, 而另外一些对象却只能由控制器设定, 如状态或错误状态。

驱动器提供多达 256 段程序空间供用户使用, 所有的程序都保存在可掉电保存的内存区间。每段程序都有一个相应的程序号, 最多包含 8 个具有 RW 属性的对象列表内的对象。调用一段程序后, 该程序段内的所有指令会按照从上到下的顺序执行一次。一个在线的命令或者是另外的一个程序段都不会中断该程序段内指令的执行。程序段可以被控制事件执行, 每个事件发生时可以被指定执行一段程序。如果一个事件发生, 那指定的这个程序段也会执行。因为事件有可能同时发生, 那么相应的程序将被事件发生的先后顺序执行。ED 伺服提供的典型控制事件有输入口的上下沿、内部编程确定的定时器时间到、两个值比较结果为真、定位末端(到位)等, 一些事件可以重复定义, 所以一段程序可以执行一次也可以执行多次。在一段程序内又可以调用另外一段程序。

Sequencer programming 窗口如下:

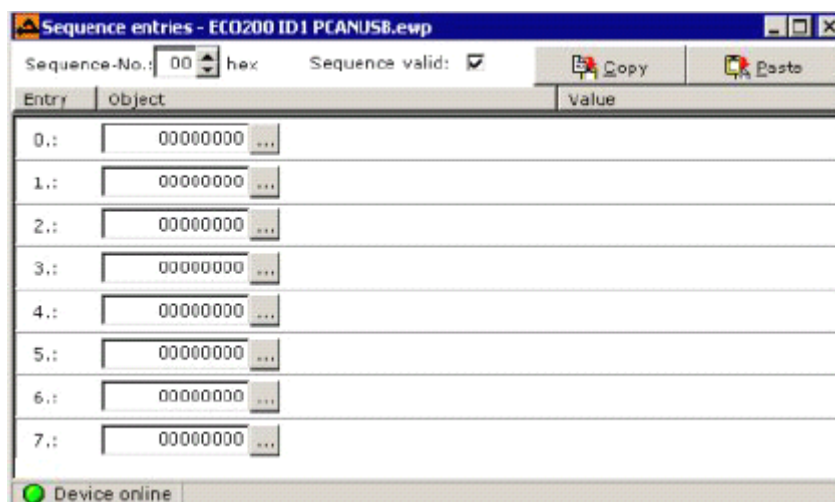


图 9.1 输入程序

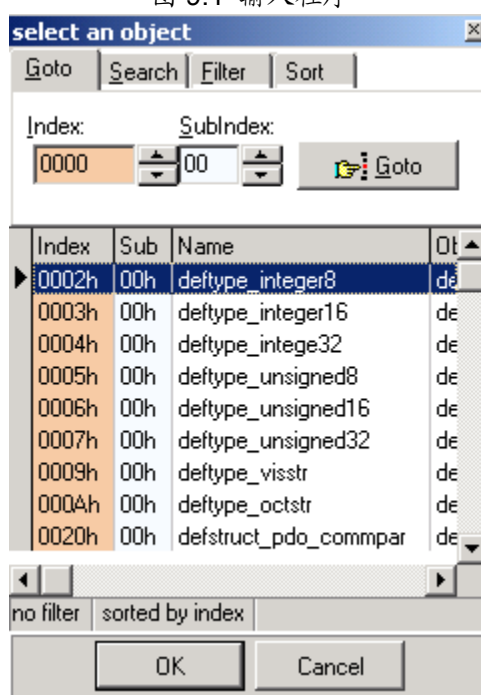


图 9.2 选择一个对象

每段程序内可以设定 8 个对象。

1. 从“Entry0”开始。
2. 点击 “...”,打开“Select an object”窗口。
3. 选择一个你需要的对象，输入到程序段中。
4. 点击确认。

选择的对象和对象提示显示在程序段相应行内，然后：

1. 赋值给对象。
2. 点击“Sequence valid”右边的选择框，出现“√”，表示该程序段有效。
3. 提供一个调用该程序段的条件。例如 5.5.2.1 节中边沿触发程序段，设置一个输入调用该段

程序。

范例：图 9.3 为两个连续的程序段。

第“00”段程序功能为：

- ③ Entry 0, 设置原点方式 1
- ③ Entry 1, 设定查找原点信号的速度为 10000inc/s
- ③ Entry 2, 设定查找原点过程中的加减速
- ③ Entry 3, 改变工作模式为“Homing”
- ③ Entry 4, 激活电机开始寻找原点
- ③ Entry 5, 如果原点找见, 跳转到 0x10 段程序

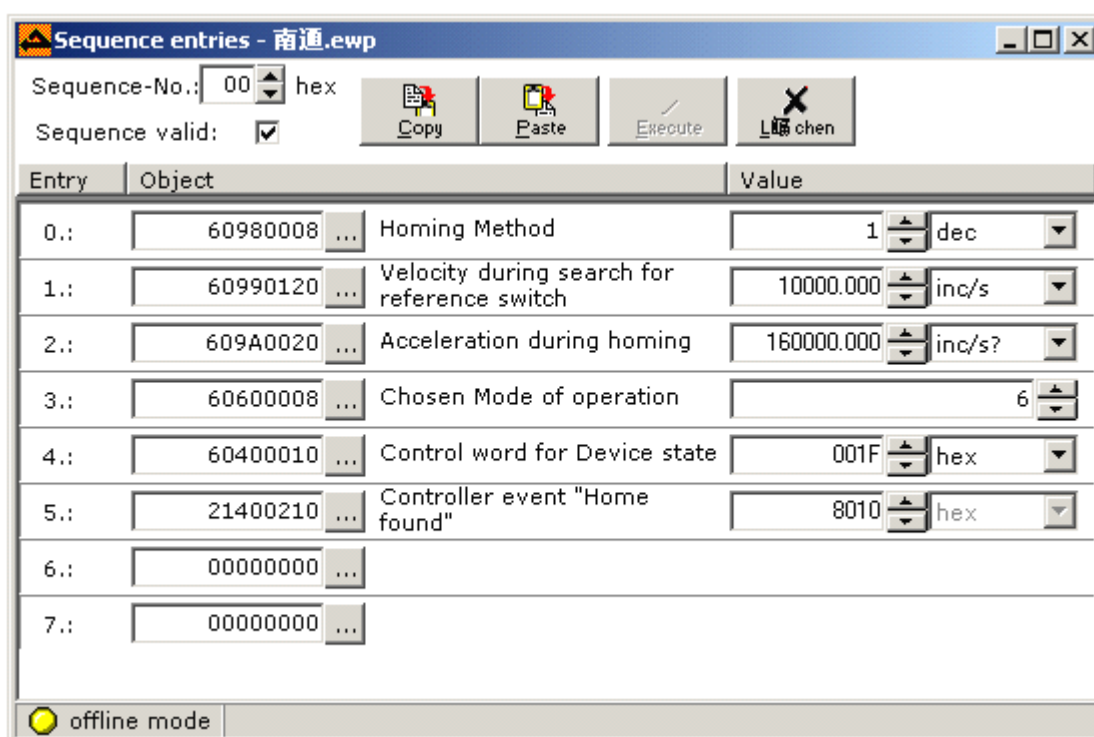


图 9.3 两个连续的 Sequences-1

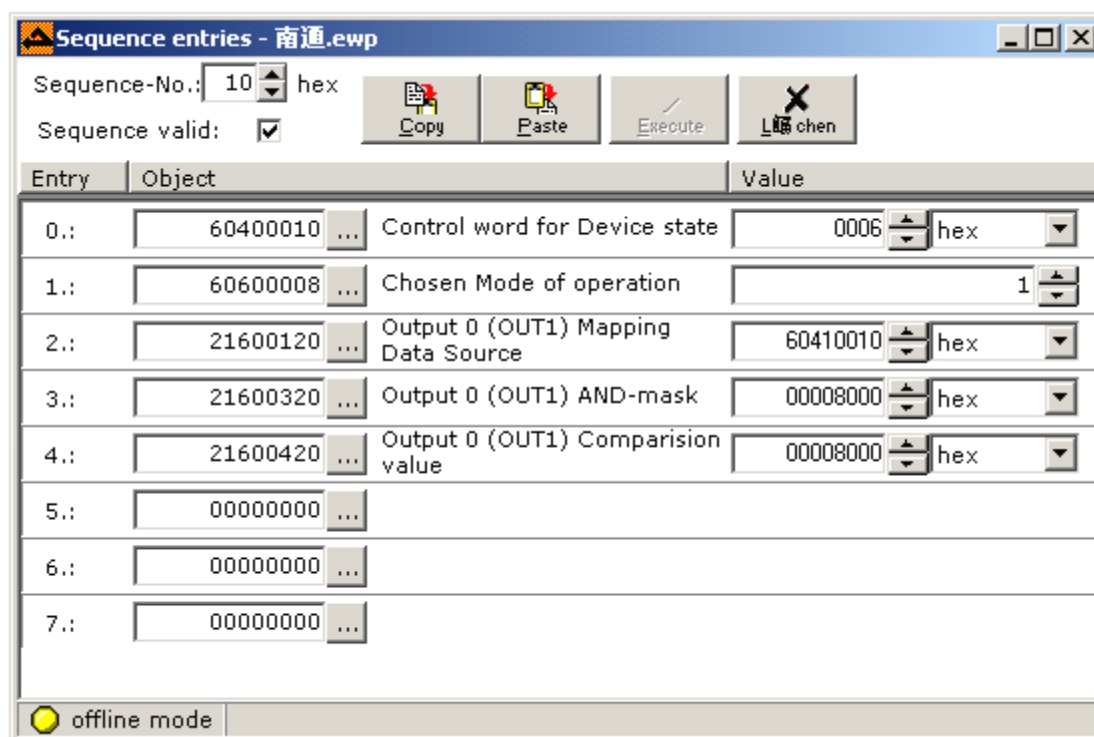


图 9.3 两个连续的 Sequences-2

程序段“0x10”功能为：

- ③ Entry 0, 松开电机轴，电机轴可自由转动
- ③ Entry 1, 设定工作模式为位置模式 1
- ③ Entry 2, 映射输出 1 的数据源为 604100 对象
- ③ Entry 3, 输出 1 的“AND-mask”设定为 8000
- ③ Entry 4, 输出 1 的比较值也为 8000
- ③ Entry 5, 如果原点找见，输出口 1 为高电平

在上面的例子中初始化原点模式中特定的参数，输出口 1 在“Reference found”时输出高电平（+24V），工作模式切换到位置模式。

由于 ED 伺服采用欧系伺服特有的对象字典编程模式，对于大部分中国内陆地区工程师不是很熟悉其编程原理，这里介绍一下顺序编程之经验步骤

1: 根据工艺规划好运动曲线，确定伺服各阶段所工作模式

2: 先在 sequence programming 中写入各工作模式下必须的对象字典及设定值。这些可参考根据 5.3.1 章节介绍的驱动器工作模式介绍。例如位置模式下必须要有选择模式（606000008），目标速度（60810020），目标位置（607A0020），控制字（60400010）这四个基本对象，还可以根据需要进行选择加减速等其他对象。

3: 模式写入后，确定各操作模式之间的转换条件，根据各种条件去第 5 章和常用对象列表里面找到各对象具体地址及其值该如何设置。例如无条件直接跳转程序段是 21180008，延时跳转是 213001 和 213002 结合应用，

比较跳转时 2180XXXX 这组参数，特殊事件触发是 2140XXXX 这组参数等等。

4: 以上就基本确定了整个运动曲线。接下来要做的就是确定各种运动曲线中的参数具体来源。例如 A 模式下的速度是固定不变的，可以在 sequence programming 中直接写入需要的速度，而其要定位的位置是根据某些外部通讯设定的参数通过运算得到的。这就需要到应用计算器 21A0XXXX 这组参数

5: 所有内部编程对象地址均可在第五章各对应章节找到。也可参考第十一章常用对象列表。详细顺序编程举例可参考附录

5.5.2 数字输入(Digital input)

5.5.2.1 边沿触发程序段

在Digital input 窗口设定ED 驱动器的8 个数字输入口可以用来调用内部程序段的功能。

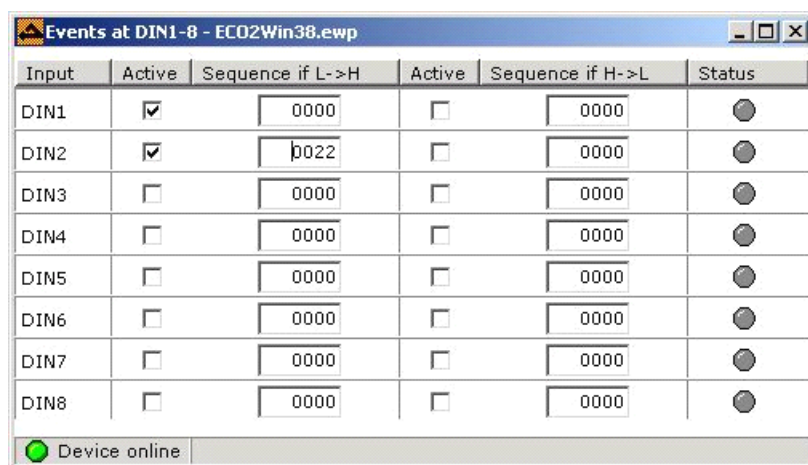


图5.15

此画面中“Active”由参数0x21210010 控制，16 个程序段号由0x21200110~0x21201010 共16 个对象指定。ED 系列驱动器共提供8 个Input 口(DIN1~DIN8)，其电平的变化可以触发不同的程序段，用户可以很方便地根据外部条件来触发电机执行不同的动作，比如图中DIN1 由L(低电平)跳到H（高电平）时将触发程序段 0x0000，DIN2 由L 跳到H 将触发0x0022 程序段。设置完毕后，对象地址表中0x21200110 中存有程序段号 0x0000，0x21200210 存有程序段号0x0022。注意：0xXXXX,0x 表示16 进制，前面两个XX 代表特殊含义，在其他功能介绍时会提到，默认为00，后面两个XX 代表Sequence No.

典型的伺服应用使用3 个输入口做为限位和原点输入（参考第8 章“限位开关与原点控制”）如果使用了这3 个输入口后，可以提供触发的数字输入口就只有5 个，如果你需要通过输入口调用更多的程序段，那就需要采用下一节介绍的编码方式来调用。

5.5.2.2 二进制编码触发程序段

1. 实现该方式的条件

- .有2~6 个输入信号供编码
- .提供一个高位输入供触发程序段
- .实际调用的程序段号=输入的BCD 值+偏移值（Offset）

2. 设定方法

打开Sequencer programming->Digital Input， 定义一个高位输入为触发信号。

BCD 输入	使能 BCD输入的触 发信号 (输 入)	逻辑状态	BCD 输入最 大值	偏移值 (Offset)	实 际 调 用 的 程 序 号	
					(Sequence No.) 没有偏移值	有偏移值
DIN1-2	DIN3-8	4	03	04	00-03	04-07
DIN1-3	DIN4-8	8	07	08	00-07	08-0F
DIN1-4	DIN5-8	16	0F	10	00-0F	10-1F
DIN1-5	DIN6-8	32	1F	20	00-1F	20-3F
DIN1-6	DIN7-8	64	3F	40	00-3F	40-7F

表5.10

例如:定义DIN1-DIN3 为BCD 输入,则可以在DIN4-DIN8 任一个输入做为高位的触发信号,下图定义DIN4 做为触发信号, 偏移值为08, 根据表5.10 定义, 设置DIN4 对应的“Sequence if L->H”一栏值为0708。

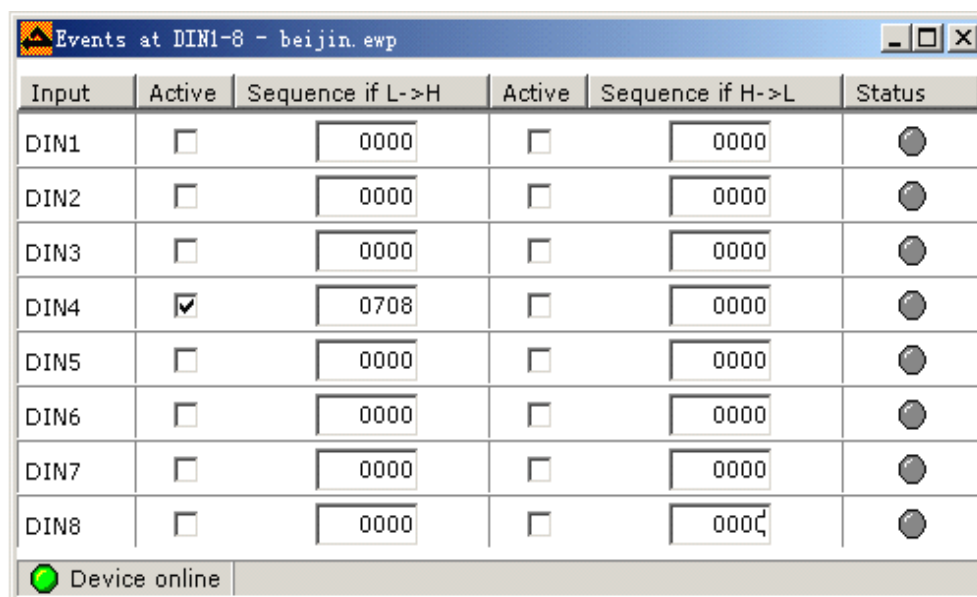


图5.16 2 进制编码方式调用程序段的设置

当DIN1~DIN3 输入对应的BCD=3 (DIN1=1, DIN2=1, DIN3=0, 1 代表高电平输入, 0 代表低电平输入) 时, 则DIN4 从L->H (低电平->高电平), 将调用第8+3=11 段Sequence, 对应的Sequence 号为0B(16 进制)。

当DIN1~DIN3 输入对应的BCD=6 (DIN1=0, DIN2=1, DIN3=1)时, 则DIN4 从L->H (低电平->高电平), 将调用第8+6=14 段Sequence, 对应的Sequence 号为0E(16 进制)。

设置的触发信号值, 如上表中的0708, 请参照表5.10 中设定。值: 0xXXXX, 前面2 位为高位, 对应的 值为表5.10 中第4 列内容, 后面2 位为低位, 代表偏移量, 对应的值为表5.10 中第5 列内容。

又如: 利用DIN1~DIN6 做为2 进制编码输入, 则可以定义DIN7~8 中任一个做为高位输入信号触发用。下图定义该信号为DIN7, 偏移值为40。当Din7 输入电平由L->H, 实际调用的程序段 = (Din1~Din6) 编码+ 0x40,当DIN1~DIN6 对应的BCD=3 时, 实际调用0x43 段程序。

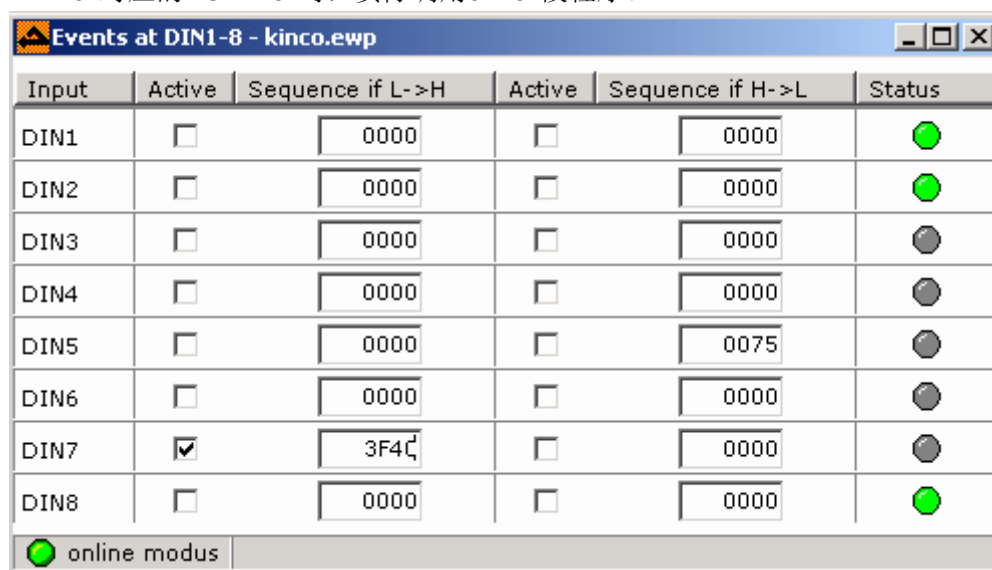


图5.17 Digital input

注意：用作BCD 编码用的输入最大到Din6。

内部程序段最大为256（0xff）段，所以（偏移值+BCD 值）务必不能超过该值。

使用规则：

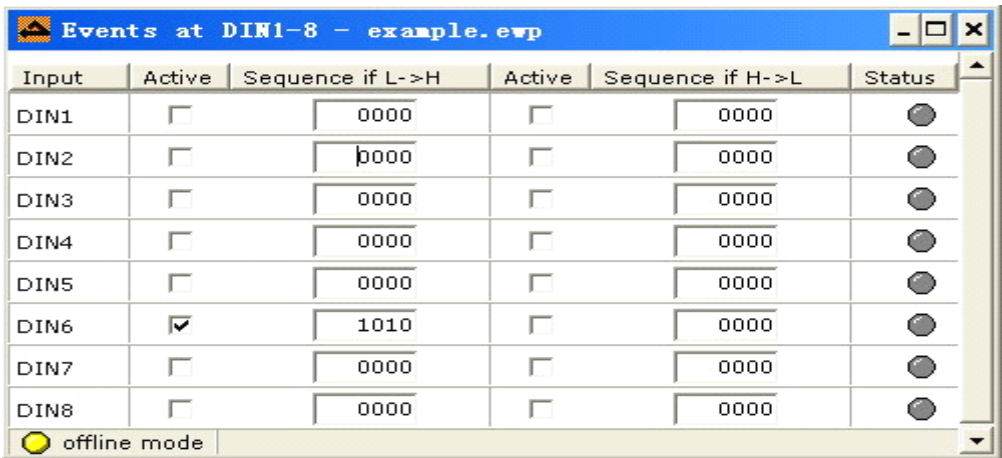
- 1、实际调用的程序段个数 $X \leq 2^{Y-1}-1$ （Y 代表高位输入的 DIN）；
例如：DIN5 可以调用的最大程序段个数为 $2^{5-1}-1=15=F$ （HEX）。可以调用 0~15，16 段程序
- 2、所需设置的 seq 调用个数=用连续调用的 seq 最高程序段—seq 最低程序段；
例如：如果要调用 seq10~seq20，17 个程序段，实际调用的程序段个数为 $20-10=10$ （HEX）
- 3、偏移量的设置：就是用调用程序段的起始值作为偏移量。

范例

要求用二进制编码触发程序段 seq10~seq20

- 1、调用个数 $20-10=10$ （HEX）
- 2、由于 $1F > 10 > F$ （根据上述使用规则 1，DIN5 可以调用 F 个程序段而 DIN7 可以调用 1F 个程序段），
所以要用 DIN6 输入为高位触发信号
- 3、偏移量用调用程序段的起始值（seq10），所以偏移量为 10

设置如下：



Input	Active	Sequence if L->H	Active	Sequence if H->L	Status
DIN1	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN2	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN3	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN4	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN5	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN6	<input checked="" type="checkbox"/>	1010	<input type="checkbox"/>	0000	
DIN7	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN8	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	

offline mode

如果要调用 seq15，也就是 $15=10$ （偏移量）+5，设置 DIN1，DIN3 为 ON（相当于， $2^{3-1}+2^{1-1}=5$ ）。用 DIN6 作为高位触发信号。

注意要点：

- 1、二进制调用 seq 程序段必须是连续的 seq 段号，但是程序一般不是连续触发的，所以一般在连续的段号内写入 21180008 对象的值来跳转到需要的程序段。
- 2、有时要求触发的时候不动作，要在 sequenceXX 中把 Active 的 ☒ 取消，然后在条件满足的时候再将 sequenceXX 中把 Active 的 ☒。这个对象地址是 20XX0120，XX 是 16 进制的程序段号。该组相关对象定义如下表：

Index[hex]	Subindex[hex]	名称	含义
2120	01-10	Sequence call	DIN1-8上升沿或者下降沿调用程序段号，实际应用意义不大
2121	00	Mask for events	DIN1-8上升沿或者下降沿调用程序段是否激活，常用于工艺互锁，例如手动/ 自动相互转换等
2122	00	Status input	8位只读变量。DIN1-8的高低电平状态。常与比较器结合应用在多个开关组合场合

5.5.3 比较器 (Comparator)

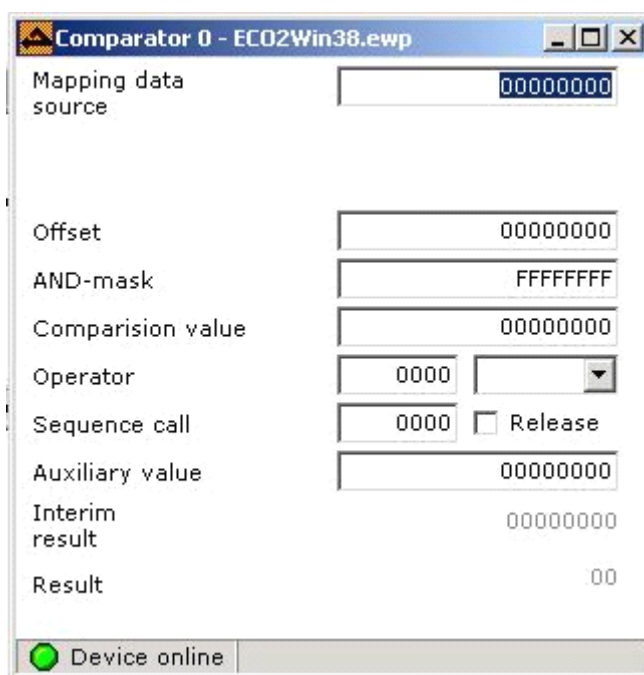


图5.18 Comparator

驱动器内部提供4 个比较器, 利用比较运算结果来触发程序段, 4 个比较器对应内部对象Index:

0x2180~0x2183, Sub index: 01~09, 详细的说明请参考第13 章“常用对象列表”中“比较器参数”部分。

计算过程如下:

$(\text{Mapping data source} + \text{Offset}) \text{ and } (\text{AND-mask}) = (\text{Interimresult})$

$\text{Result} = (\text{Interim result}) \text{ Operator } (\text{Comparison value})$

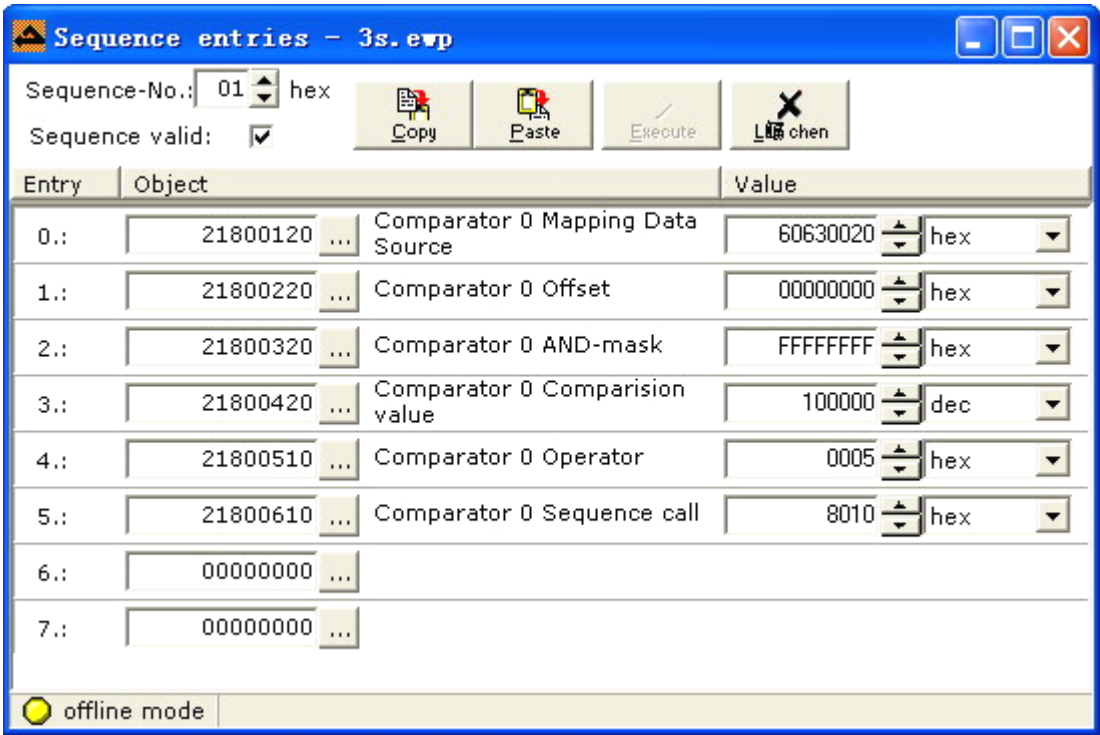
如果结果为真, Result 显示为 1, 点击“Release”, 事先输入的程序段将被触发, 要再次触发需再次点击 “Release”, 体现在Sequence 编程中也就是要再次比较。比较器对象定义如下:

Index[hex]	Subindex[hex]	名称	含义
2180— 2183	01	Mapping data source	映射的数据源
	02	Offset	偏移量
	03	And mask	逻辑与
	04	Comparision value	欲比较的值（注意是十六进制）
	05	Operator	比较符号
	06	Sequence call	比较结果为真时调用的程序段
	07	Auxiliary value	0x-80xx 内部辅助值，可读、写和映射到其它对象

表5.11 比较器有关对象

范例1

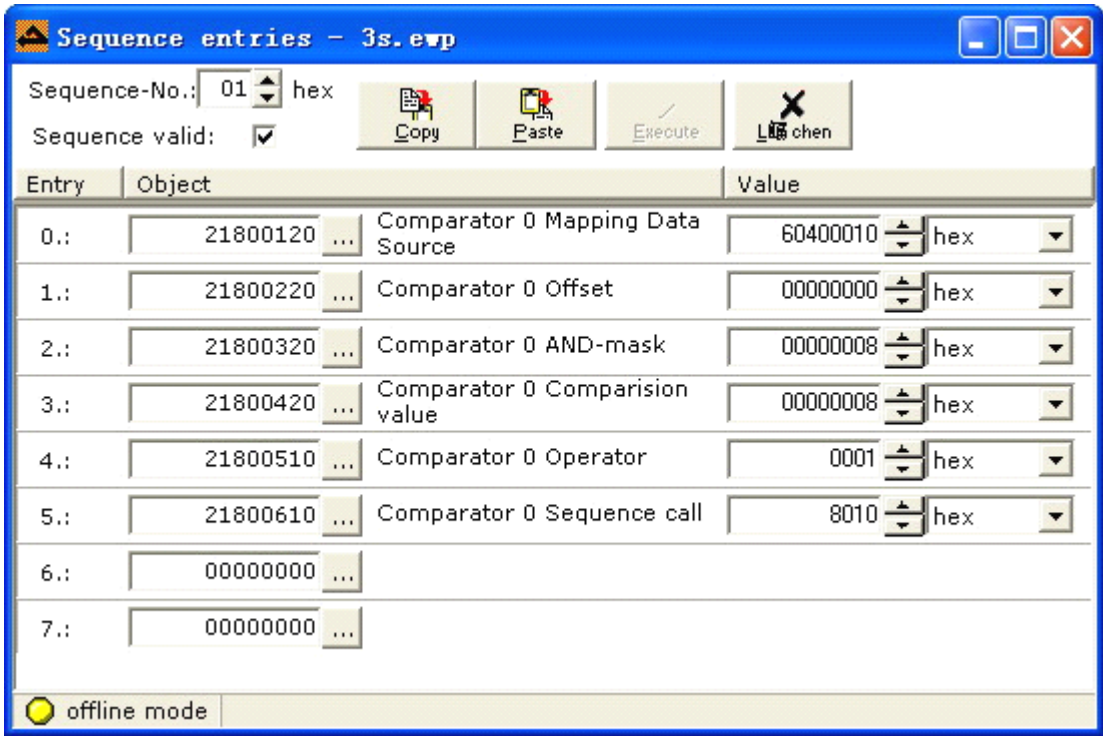
要求：触发seq1，当实际位置（60630020）超过100000（dec）时，调用seq10。其程序如下图所示。



说明：如果对一个寄存器中的数据大小进行比较的话，就将这个寄存器AND FFFFFFFF，所以21800320=FFFFFFF。因为比较器一旦触发，在没有取消比较前是1ms比较一次。所以当比较的寄存器内容在高速变化的时候，一般21800510采用≤（03）或≥（05），而不用=（1）。

范例2

要求：触发seq1，当驱动器报警时（60410010中的第4位为0N），调用seq10。其程序见下图。



说明：如果对一个寄存器中的“某一位”进行比较的话，就将这个寄存器AND ...00100... (BCD)，所以21800320=0008，目的是将其余不比较的位给屏蔽掉。一般21800510采用=（1）。

注意：比较器一旦被触发，就会一直在周期比较（周期为1ms），直到满足比较条件，跳转为止。如果中途要停止比较器可以将218X0610=0就可以停止比较器了。如果要启动比较器就将218X0610=80XX（HEX）。或者可以把映射数据源218X0120=0也可以停止比较器，如果要重新启动再把数据源赋过去！

5.5.4 定时器、控制器内部事件（Timer/Controller events）

5.5.4.1 定时器触发程序段

ED 提供了定时器和事件触发程序段功能，可以给编程带来很大的灵活性。如下图所示。

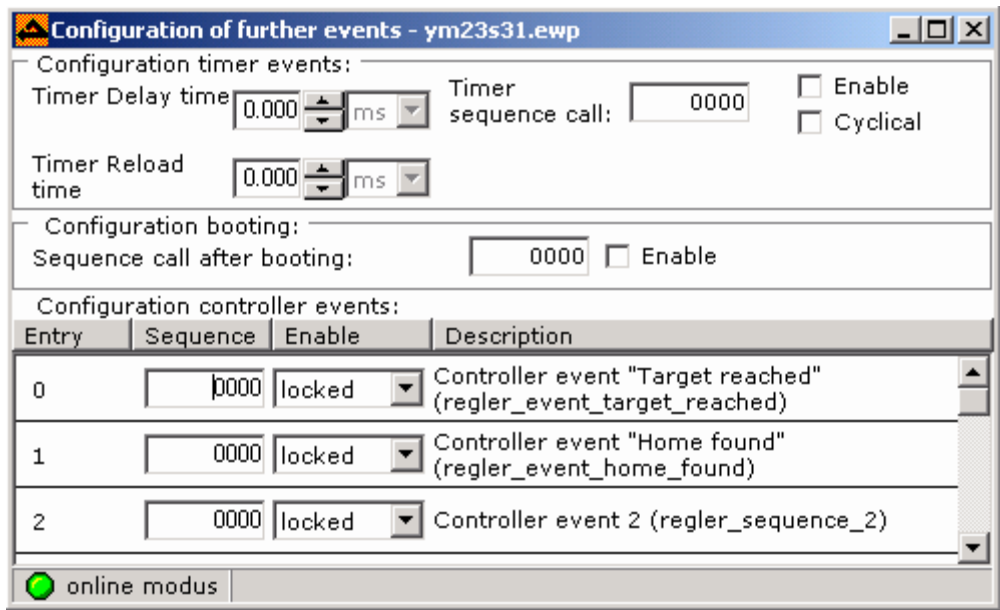


图5.19 定时器事件

图中的最上方为定时触发程序段，当我们在Timer sequence call 一栏输入SEQ 号（格式为80XX,XX 为程序段00~FF），Timer Delay time 输入延迟时间，再点击右上角的“Enable”，这时定时器启动，当定时时间到，相应的SEQ 就运行。

那么在SEQ 编程中要实现同样的功能，我们该如何写呢？这里要记住定时器所对应的两个对象（寄存器地址）为0x21300110 对应程序段号，0x21300220 对应定时时间，见下图：

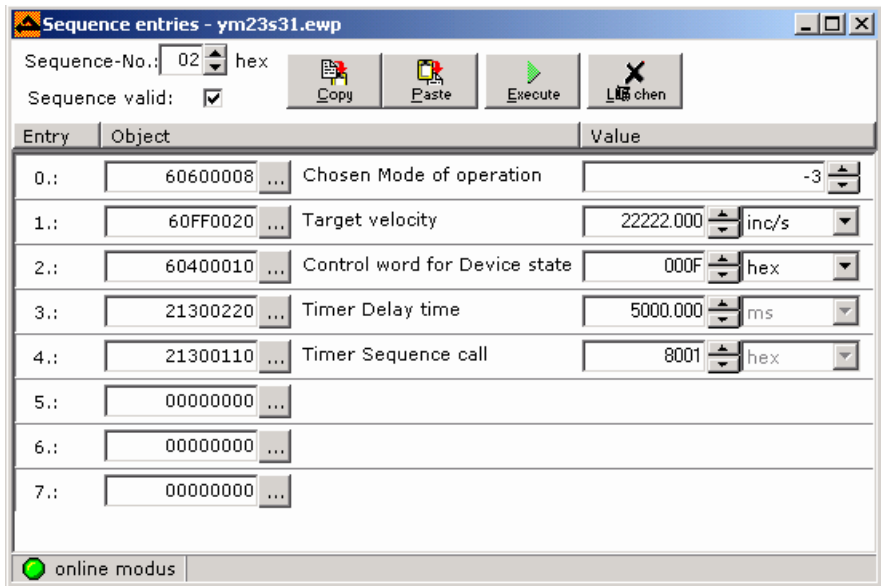
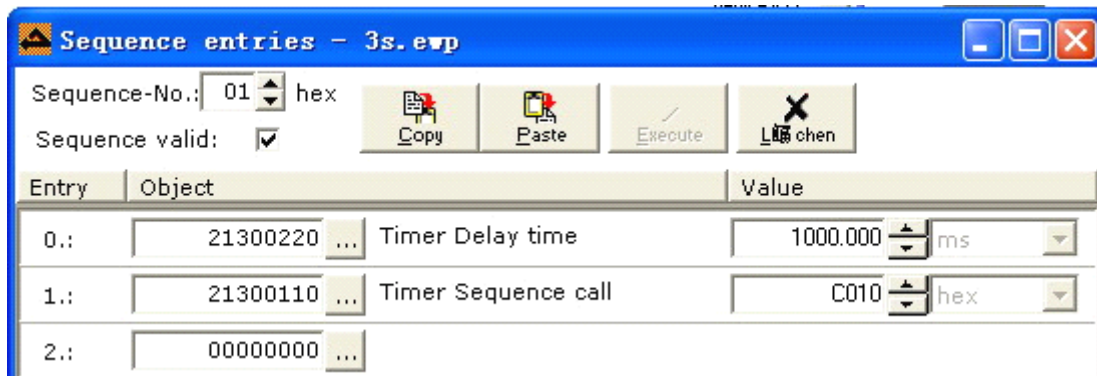


图5.20

上图中SEQ02 段程序的第3 和第4 两条指令表示：当SEQ02 被调用后，首先设定系统工作模式为-3，电机以22222inc/s 的速度运行，后设定定时器1 延迟时间=5S，定时时间一到，SEQ01 被调用。注意输入的SEQ 号高字节必须为80。

定时循环调用，如下图所示。



上图SEQ意思为：每1000ms调用SEQ10一次。注意输入的 SEQ 号高字节必须为C0，因为这是无条件的周期调用，如果使用不好很可能进入到死循环，所以建议在使用的时候要格外小心。

5.5.4.2 系统引导后自动调用某段程序(Sequence call after booting)

21500010 地址内容为 80xx, 那么伺服驱动器上电 (24V 控制电源) 之后, 就自动执行 SEQxx 段程序。

建议用户不要使用该功能, 如果非要使用, 请确保系统的机械机构提供了必要的安全保护, 并且确保在上电后由于立即执行的那段程序不会由于负载初始位置不同而造成碰撞。

5.5.4.3 控制器事件(Controller events)触发程序段

ED 驱动器提供的内部事件有 “Target reached” 和 “Home found” 等, 这些事件我们可以在 Sequence 编程中 加以利用。控制器事件对应的对象为 0x2140xx, 详细的事件说明见第 13 章 “常用对象列表” 中 “事件参数” 一节, 比如设定 21400110=8001, 21400110 对象功能为 Control event “Target reached”, 就是指在 “位置到位” 事件发生时调用第 1 段 sequence, 指定的 sequence 格式为 0x80xx, 16 进制表示。xx 范围: 00~FF。

5.5.5 计数器功能 (Counter)

ED 内部共有 4 组计数器, 对象地址分别为 0x2190~0x2193, 每个计数器有 Sub01 和 Sub02 两个子对象, Sub01 存放计数增量值 (见下图左 Increment 栏), Sub02 存放累加值, 当我们需要作累加运算时, 就将数值放在 Sub01, 此时 Sub02 内的数值会自动以 Sub01 中的值累加。

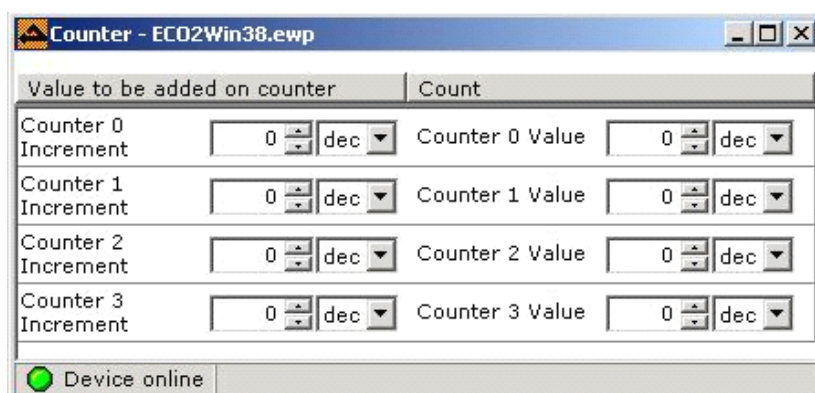


图5.21

范例: 利用外部 DIN 输入, 当每次输入口的电平由 L->H 变化时, 对计数器加 1, 见下图:

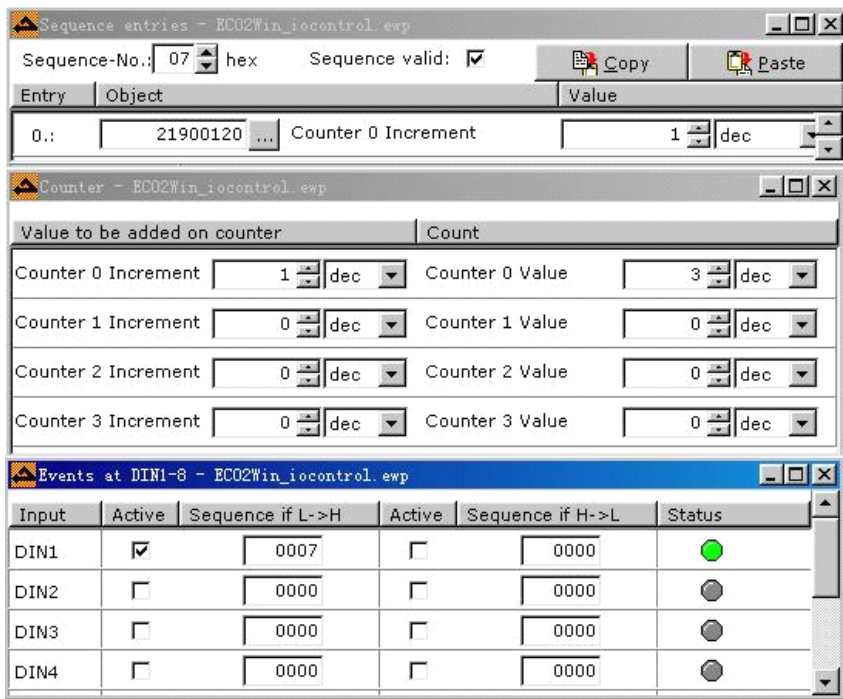
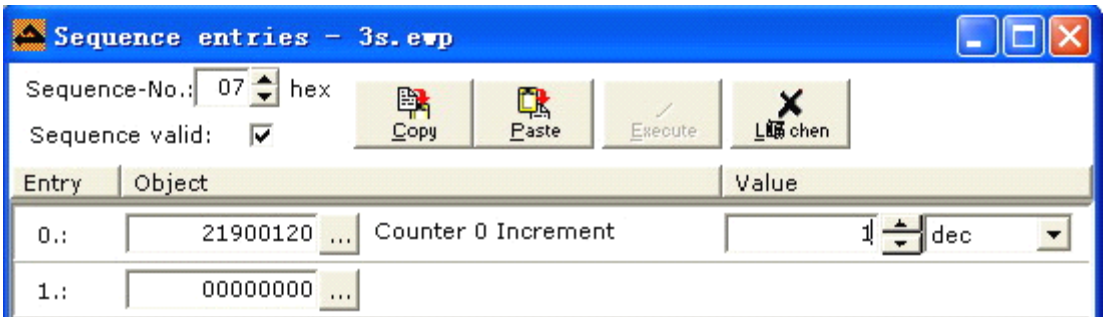


图5.22 计算器配置

上面的范例就是设定每次Din1 由L->H 时，将触发SEQ07，对计数器0（0x21900120）累加1，相当于对DIN1 的输入脉冲进行计数，这在许多场合是十分有用的。

记忆公式：21900220=21900220+21900120

Seq7编程为：



计数器对象定义如下：

Index[hex]	Subindex[hex]	含义
2190—2193	01	Increment，每次调用对象时自动增加的数据
	02	可写，计数器当前值

表5.12 计算器对象

5.5.6 计算器(Calculator)

驱动器内部提供一个计算器进行数据复制和加减乘除以及位逻辑运算处理，见下图：

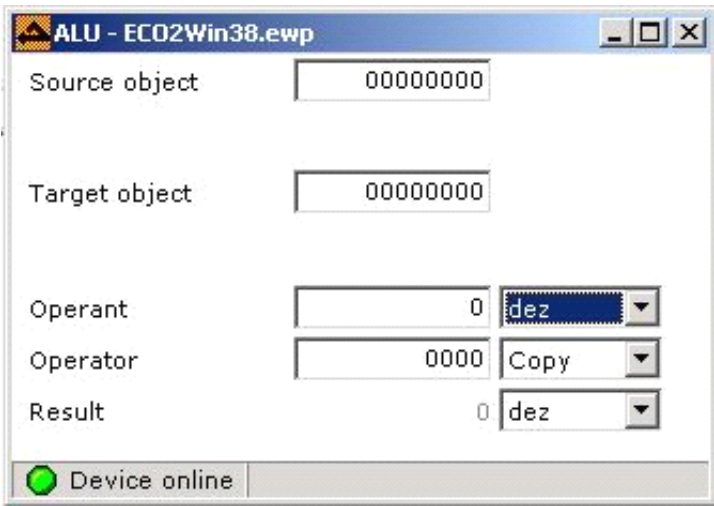
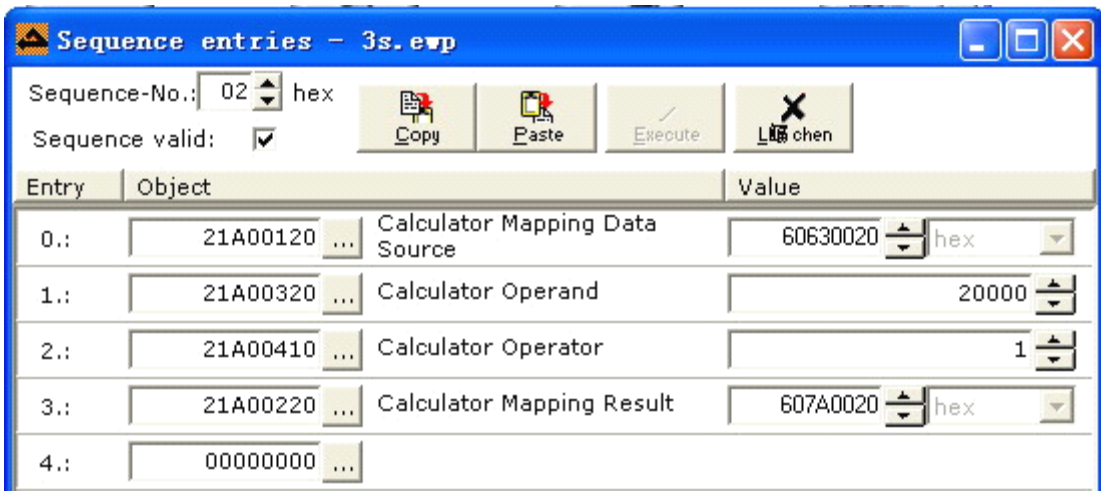


图5.23 计算器

所有运算在寄存器地址0x21A0 中进行，共有Sub01~05 五个寄存器，Sub01（Souce object）存放的是将要被运算的源参数，Sub02(Target object)存放运算的最后结果（必须是具备 Mapping 属性的参数寄存器地址，Sub03 是与源参数一起运算的数，Sub04（Operator）是运算符（共有COPY、+、-、*、/、AND、OR、XOR 八种，Sub05 存放运算结果，此处测试中Operant 请用直接参数寻址方式修改，注意填好Sub01、Sub03 和 Sub04，最后一步输入Target object 到Sub02 中，新的运算结果立即生成，在Sequence 运用计算功能时方法相同。

范例：将实际位置（60630020）+20000放入目标位置（607A0020）



记忆公式 A+C=M

A: 21A00120 任何地址都可以;

C: 21A00320 只可以放常数;

+： 21A00410 运算符（COPY、+、-、*、/、AND、OR、 XOR分别对应0、1、2、3、4、5、6）

M: 21A00220 结果，必须为可以MAPPING的地址，且放在运算最后

计算器对象定义：

Index[hex]	Subindex[hex]	含义
21A0	01	源对象
	02	目标对象（用于放置计算结果）
	03	被运算数
	04	运算符
	05	内部运算结果，最终被传递到目标对象内

表5.13 计算器对象定义

5.5.7 记录表格（Recording table）

驱动器内部提供256x32bit 的存储区（0x2D00~0x2DFF）供数据存放，用户可以灵活使用这个区域来保存一些数据供查询或二次处理，具体使用如下：

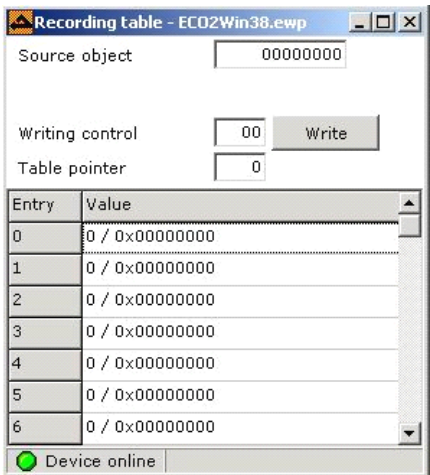


图5.24 记录用对象地址

实现这个功能由寄存器地址0x21B0 完成此寄存器包含Sub01~03 三个寄存器在source object ((Sub01)中输入目标寄存器的地址,在Table pointer (Sub03)中输入地址指针x1 (00~FF),在Writing controlSub02) 中输入1, 按“Write” 则将Source object 中的值写入到地址2Dxx 之中，同时地址指针加1，指向2Dxx+1 的寄存器。

记录表对象定义如下：

Index[hex]	Subindex[hex]	含义
	01	源对象，将要被写入表格内的值

	02	写命令
	03	写入表格内的位置，范围0x00～0xFF

表5.14 记录用对象定义

注：驱动器内部还开辟出**64KB** 的内存用来存储实时采样数据，共可存储**4** 组记录，采样时间间隔可以小到**1** 毫秒（缺省**10** 毫秒）。

5.5.8 快速位置捕捉（Position capture）

主编码器输入口X7 的Z 相（Pin4）快速输入时，伺服系统的当前位置（Actual position）值被捕获，并置于对象0x21C003 和0x21C004 中，后者为上一次捕获数据的备份。

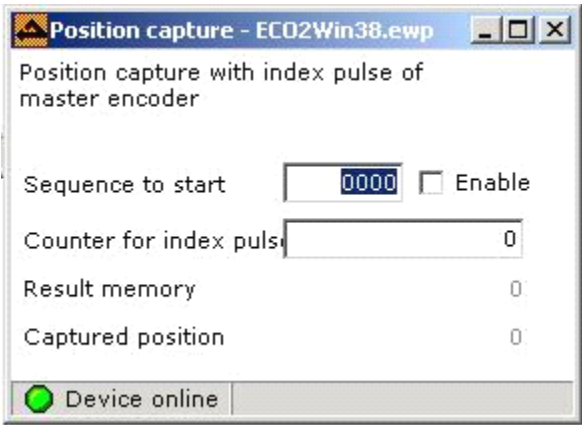


图5.25 Position capture

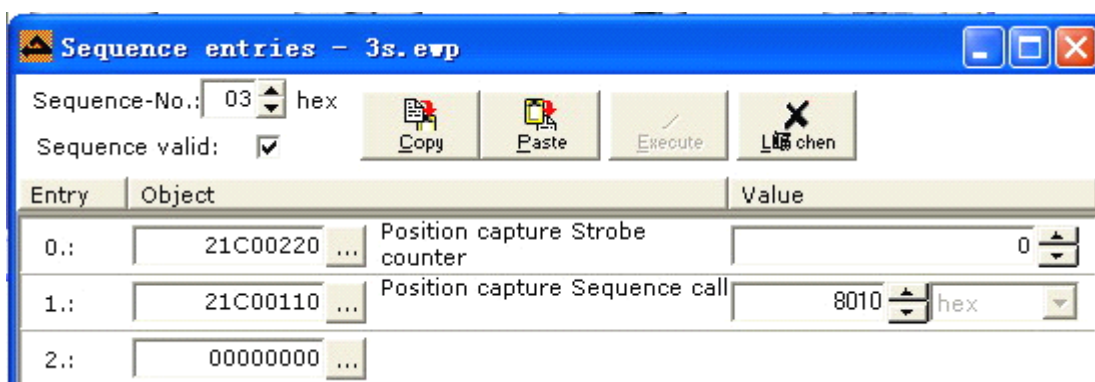
在Counter for index pulse 一栏由0->1 时，当前位置（Actual position）值将被拷贝到Result memory 和Capture position，点击“Enable”，则“Sequence to start”一栏内设定的程序段将被调用执行，如果要再次捕获 当前位置，需要将Counter for index pulse 内值清零，然后重新置为1。主编码器输入口的Z 相由L(低电 平)->H（高电平）时，Counter for index pulse 累加一次。

快速位置捕捉（Position capture）对象的定义如下：

Index[hex]	Subindex[hex]	含义
21C0	01	调用的Sequence 程序段号在21C002 计数从0->1 时，调用该程序段
	02	主编码器的Z 相脉冲计数
	03	Result memory, 被捕获的实际位置值
	04	Captured position, 前一次被捕获的实际位置值

表5.15 位置捕获对象定义

应用范例：



说明：每次触发快速捕捉口都要将快速捕捉口的位置计数器清零。所以要21C00220=0；

快速捕捉口信号到来时，调用seq10，在程序号前加80。

21C00110=80XX；

ED100，要将5V PNP信号接入外部主编码器的N端。

ED430, ED620, ED630，要在外部主编码器的 N和/N之间产生一个5V的电压差，所以PNP，NPN信号都可以。

5.6 设备配置（Device configuration）

ED 资源配置包括五个部分，如下图从上到下依次分别是数字量输入、数字量输出、电子齿轮、模拟量输入、模拟量输出监控，下面依次介绍：

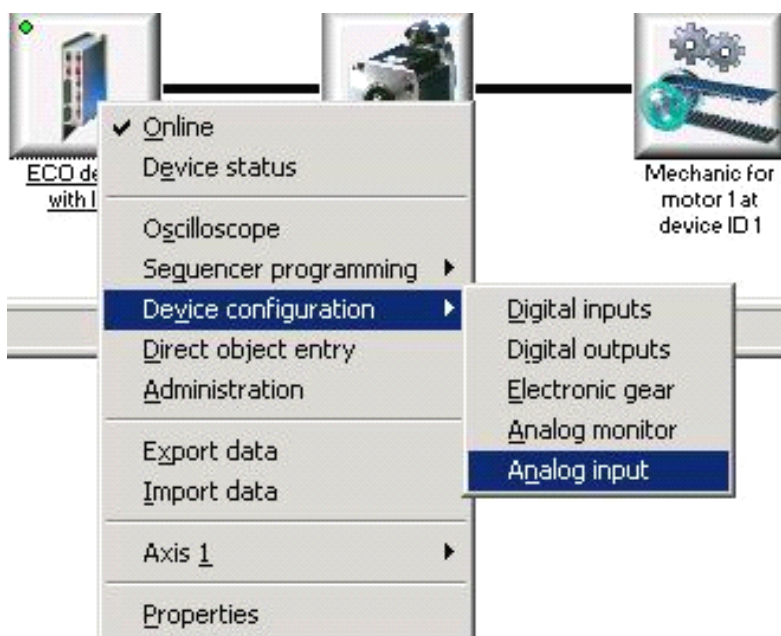


图 5.26 “Device configuration” 菜单

5.6.1 数字输入（Digital input）

ED 驱动器有 10 路数字开关量输入（Enable、Reset、8路可编程 DIN1~8、DC24V 输入），其状态由0x60FD0020 寄存器表达(高位到低位的顺序依次是Enable、Reset、DIN8...DIN1,注意此寄存器为

32BIT, 共4 个BYTE, 从高到低分别用Byte4~Byte1 表示, 其中Enable 功能是外部使能驱动器, 如果没接该信号, 驱动器将报故障, Reset 功能是复位驱动器故障状态, Din1~8 输入的许多功能前面已经说到, 下面介绍新的功能,

如图:

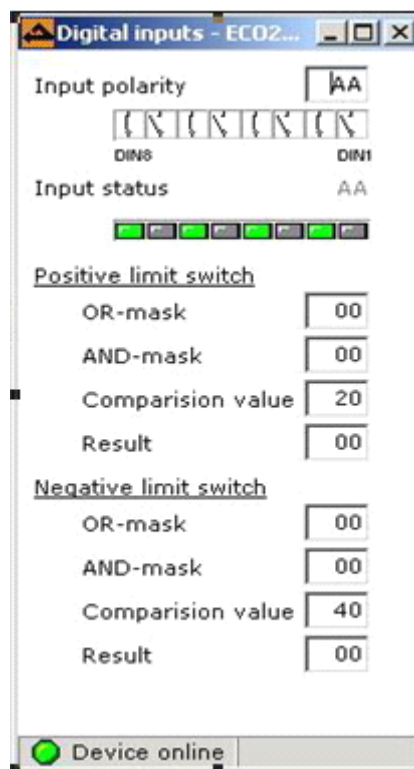


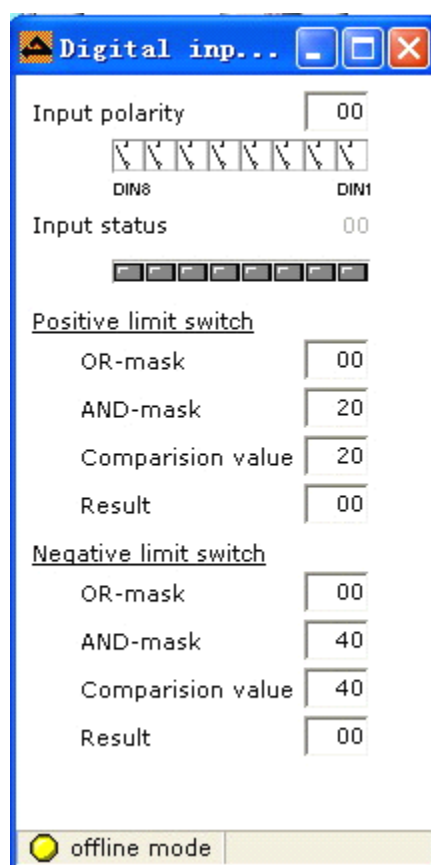
图5.27 Digital input

图中控制命令对应对象分别是 0x2170、0x2171、0x2172, 其中 “Input Polarity” 对应 0x21700008, 八个 Bit

对应DIN1~8 的输入状态转换命令, 当x 这一位为 “1” 时, 则相应输入DinX 极性被改变, 如果此寄存器的值为 00, 则输入的极性与实际输入相一致, 比如外部输入为 0x00(即输入全为低), 我们在 0x21700008 输入 AA, 则DIN8、6、4、2 将被认为是高电平输入; 若外部输入为 0x01 (即Din1 高电平输入有效, 我们在 0x21700008 中填入 01, 则DIN1 的状态在内部被切换到低电平。

对象0x2171 包含4 个Sub Index, 用来描述Din (LMT+), 判断DIN6 输入是否为高电平Sub01 (OR-mask) 目前功能保留, 我们在Sub02 (AND-mask) 中输入 20 (hex, 即 0010 0000), 用来屏蔽DIN6 之外其它输入状态, 在Sub03 (Comparision value) 中也输入 20 (hex), 则系统会自动将DIN1~8 的输入状态和 “AND-mask” 中的值进行 “逻辑与” 操作, 其结果与 “Comparision value” 中值比较, 如果相等则Sub04 (Result) 置为 “1”, 表明Din6 的输入为高电平, 否则结果为零, 表明Din6 为低电平输入。如此设置, 正向的限位开关信号才有效。

0x2172 与0x2171 相似, 只不过它用来描述Din7 (LMT-), 所以AND-mask 和Comparision value 设为 40。如此设置, 负向的限位开关信号才有效。如下图



5.6.2 数字输出 (Digital output)

驱动器包含两路可编程的数字量输出 (24V, 0.5A, 说明如下:

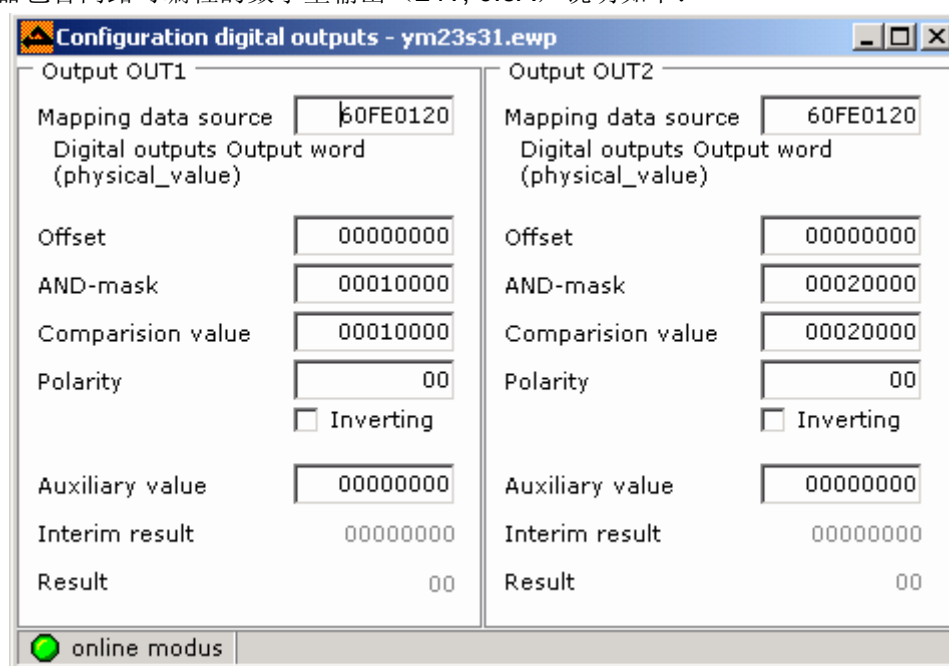


图5.28 Digital output

可以看到两路输出控制方式一样，都是通过内部逻辑处理后控制输出，具体说明如下： 两路输出对应的内部寄存器地址为0x2160 和0x2161

Index[hex]	Subindex[hex]	名称	含义
2160, 2161	01	Mapping data source	映射数据源
	02	Offset	偏移量
	03	AND-mask	与掩码
	04	Comparision value	比较值
	05	Polarity	极性
	06	Auxiliary	内部辅助值
	07	Interim result	比较结果
	08	Result	输出接口

表5.16 Dital output 有关对象

用户首先要Mapping 一个对象地址到Sub01，作为计算的源参数，再按下面计算步骤进行计算： 第一步，Sub01+Sub02 = Result1

第二步，Result1 and Sub03 = Sub07

第三步，Sub07 与Sub04 比较如果Sub05=0，则比较结果为真时，Sub08=1；比较结果为假时，Sub08=0

如果Sub05=1， 则比较结果为真时，Sub08=0；比较结果为假时，Sub08=1

第四步， Sub08 的结果为“1”，对应OUTPUT 输出高电平，为“0”则OUTPUT 输出低电平。

举例：如果要把OUT1 口设置为“目标位到输出”，OUT2 设置为报警输出，那么output 的设置如下：

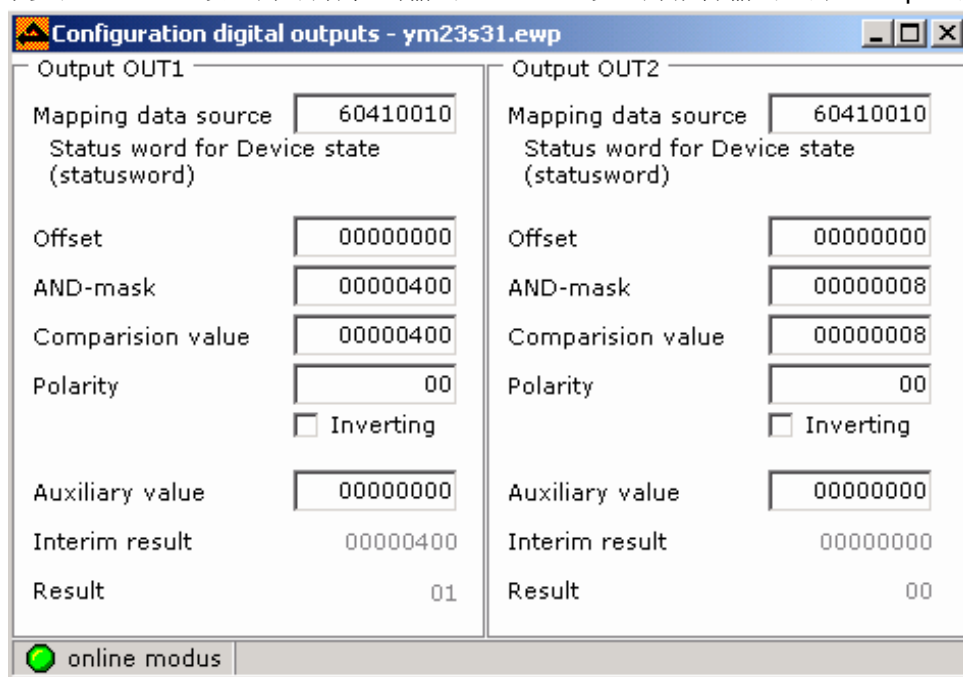


图5.29 Digital output 举例

从上面的设置可以看出，如果要输出“目标位到”信号，只需要“60410010”状态字中间的第11位“Target reached”位为“1”，也就是“Device Control”页“Status word for Device state”的“Target reached”位显示“选中”状态时，Out1 将输出高电平。上面的“AND-mask”输入的是“400”，就是为了屏蔽掉其余的位，单独留下第11位和“Comparision value”比较，而“Comparision value”设置为“400”也是为了只保留第11位为“真”，当系统在定位模式下处于“目标位到”状态时，out1 的比较结果result=1，Out1 输出高电平。同理，Out2 由于设置功能为报警输出，所以利用的也是60400010状态字，其中的第4位为“Fault”，可以做为判断驱动器是否有故障。

除了两路可编程的数字输出口OUT1 和OUT2 以外，ED 驱动器还有Ready、Brake+和Brake-共3路特定功能输出，Ready 输出是在驱动器处于“Ready to switch on”状态时输出高电平，在驱动器有故障或没有在“Ready to switch on”状态时，输出低电平。Brake+和Brake-用于提供抱闸需要的电源，电压24V，最大1A(持续100ms 后降低为0.5A)，持续时间可以通过“Parameters settings->Others”页中的“Holding brake delay”时间设置。

5.6.3 电子齿轮比设置 (Electronic gear)

参数0x2509 包含11个Sub Index，用来设置主从跟踪控制模式和脉冲/方向控制模式各种参数，首先要将Sub02 映射为0x60FF0020(Target velocity)，模式的选择由Sub05 决定，当Sub05=0 时，系统设定为主从跟踪模式，从轴速度 = (Sub03/Sub04)*Sub09，**Sub05**在更改后一定要保存重启才生效，见下图所示：

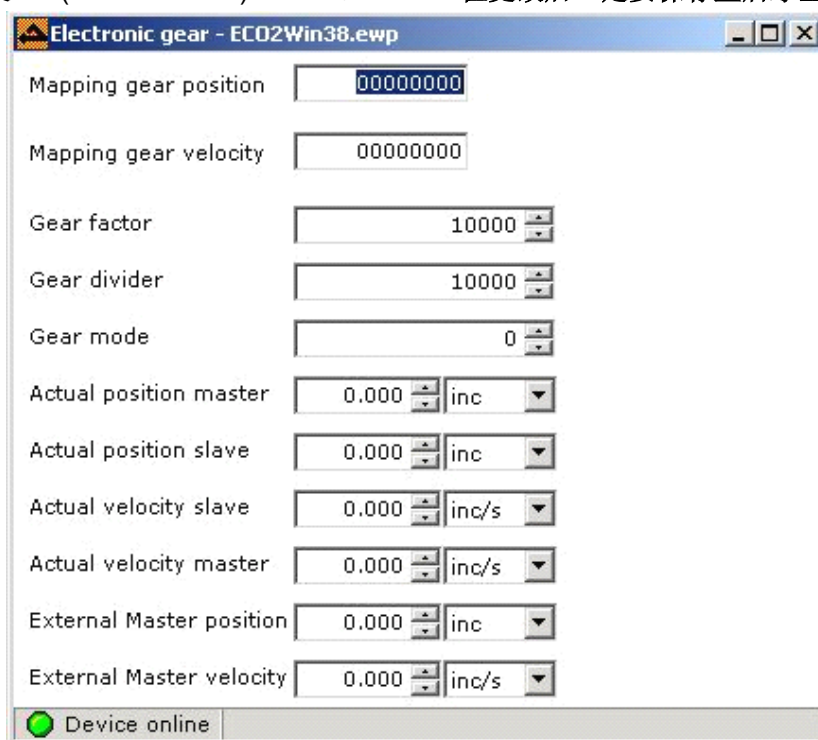


图5.30 Electronicgear

从上到下依次对应Sub01~Sub0B 共11 个寄存器。Actual position master 值为从主编码器口接收到的脉冲数，常用于判断主编码器口是否接收到外部的脉冲信号。

电子齿轮对象定义如下：

Index[hex]	Subindex[hex]	名称	含义
2509	01	Mapping Slave position	映射从轴位置数据源
	02	Mapping Slave velocity	映射从轴速度数据源，设置为60FF0020，即将来自 Master Encoder 口信号的频率值通过乘以电子齿轮比后直接转换为电机的目标速度值
	03	Gear factor	电子齿轮比的分子-32767至32767
	04	Gear divider	电子齿轮比的分母0至65535
	05	Gear mode	模式 0, 1: 主从模式, 4 倍频 2: 脉冲/方向信号控制模式 3: CW/CCW 信号控制模式
	06	Actual position master	Master encoder 口实际接收脉冲数
	07	Actual position slave	内部计算后赋值给从轴的脉冲数
	08	Actual velocity slave	实际从轴速度
	09	Actual velocity master	实际主轴速度
	0A	External Master position	外部主轴位置
	0B	External Master velocity	外部主轴速度

表5.17 “Electronic gear”（电子齿轮比）有关对象

注意！

1. 当不再使用电子齿轮比时，务必将**25090120**、**25090220** 中数据清零，否则速度模式下目标速度值无法写入。

2. 如果要使电子齿轮参数生效，也务必在设置了上述的参数后，点击“Administration->Save parameters to device now”后按钮保存后，并点击“Restart device now”按钮重新启动驱动器，方能生效。

3: ED100的电子齿轮分母不可以小于1000，因为分子（25090310）的赋值范围是 $\pm 2^{15}-1$ （ ± 32767 ）所以电子齿轮比只可以设置在1: 32.7之内。

ED430，ED630则没有电子齿轮分母不可以小于1000这个限制。

4: **Sub05**在更改后一定要保存重启才生效。

5.6.4 模拟量输出监控 (Analog monitor)

ED 驱动器带有两路独立模拟量输出，可监控内部对象的变化，常用于监控实际速度、实际电流等值变化，ECO2WIN 软件中Analog monitor 的设置窗体如下图所示：

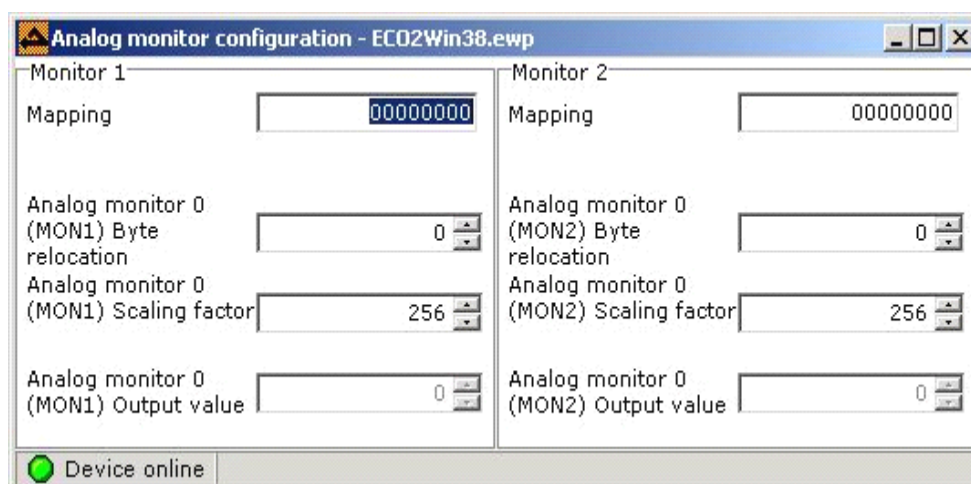


图5.31 Analog monitor

两路模拟量输出对应的内部对象分别为0x2400 和0x2401, 各包含4 个Sub 寄存器Sub01 为Mapping data (数据源), Sub02 为Preshift, Sub03 为Factor, Sub04 为Output value, ED 内部采用的D/A 为9 位bit, 数字 量范围为-256~255, 输出范围是0~5V, 0V 对应数字0, 计算公式如下:

$$\text{Output value} = (\text{Source value} * \text{factor}) / (256^{(\text{preshift}+1)})$$

$$\text{实际Vout} = 2.5\text{V} + \text{Output value} * 2.5\text{V}/300$$

此处的关键是选择合适的 Factor 使 Output value 的范围在-256~255 之间, Vout 的范围将在 0.367V~4.625V 之间, 具体范例见附录B“顺序编程”。

5.6.5 模拟量输入 (Analog input)

ED 带有一路模拟量差分输入, 采用10Bit 的ADC, 对应内部的对象地址为0x2508, 有6 个Sub, Sub01 为映射的目标对象的地址, 可以是目标速度、目标位置等内部对象, 从而实现用模拟量控制电机速度和位置的功能。Sub02 为 factor, Sub03 为 Shift, Sub04 为 Analog input in (数据来自 ADC), Sub05 为 Analog input value(result of multiplication), Sub06 为 Analog input highres state; ED 提供的ADC 分辨率位10 位, 范围从-512~512。

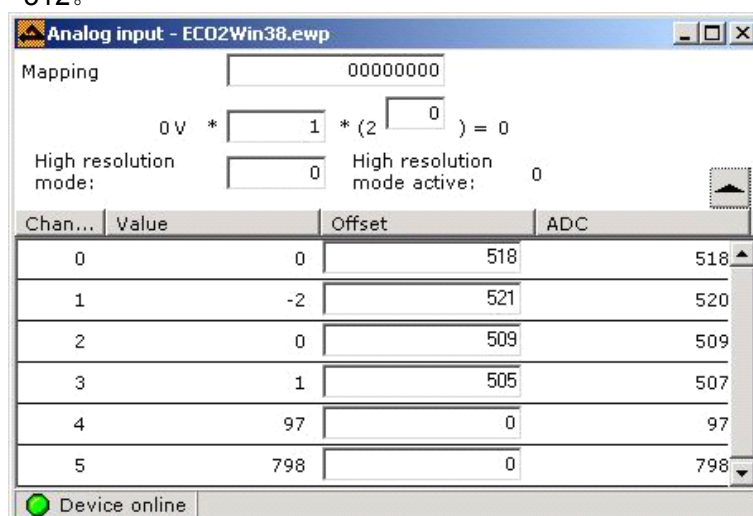


图5.32

计算公式: $\text{Sub01} = \text{Sub04} * \text{Factor} * (2^{\text{Shift}})$

下面举例说明如何利用模拟量输入进行电机调速我们假设输入 10V 对应最大正反转1500RPM经ADC后的对应输入数字量Vin 为-512~512。

第一步: 将 Sub01 映射到寄存器地址0x60FF0020,接着要确认Factor 的值,我们知道内部速度的单位是inc/64s,8000inc/转,1500RPM 转换成内部最大速度,即 $V_{\max} = (1500/60) * 8000 * 64$ (单位inc/64s)。第二步: 计算 $\text{Factor} = V_{\max} / 512 / (2^{\text{Shift}})$, 如果设定Shift=1,则Factor=12500。

这样,按照上面计算公式,我们可以得到:

目标速度

$\text{Sub01} = \text{Vin} * \text{Factor} * (2^{\text{Shift}}) = (\text{Vin} / 512) * V_{\max}$, 当输入

为10V 时, $\text{Vin} = 512, \text{Sub01} = V_{\max}$,

为0V 时, $\text{Vin} = 0, \text{Sub01} = 0$,

为-10V 时, $\text{Vin} = -512, \text{Sub01} = -V_{\max}$

由上面可以知道,计算Vmax 最关键,不同映射对象其内部的单位与输入要求会有不同,转换时要确定器内部的单位。

注意:1、当不在使用模拟量输入控制电机速度时,务必将 **25080120** 内值清零,否则在速度模式下目标速度值无法设定。

2、写入25080120的地址必须是可Mapping(可映射)。

3、处理所用单位为dec。

4、此功能无法设置模拟量死区,偏移等,如果要设置模拟量死区请使用KINCO CD系列伺服!

5.7 通讯参数设置(Communication)

ED 伺服是具备动态 PDO 配置能力的典型 CANopen SLAVE 设备。PDO 的配置存于设备中以便在没有 MASTER 设备(NMT MASTER)时启动后操作。Communication 可以用来观察NMT MASTER 对PDO 的配置或手工对PDO 的配置。

常见4组PDO,每组包括TxPDO和RxPDO,对应的ID号(注意,ID号的值越小,其优先级越高)为TxPDO1(181h ~ 1ffh)、RxPDO1(201h ~ 27Fh)、TxPDO2(281h ~ 2ffh)、RxPDO2(301h ~ 37Fh)、TxPDO3(381h~3ffh)、RxPDO3(401h~47Fh)、TxPDO4(481h~4ffh)、RxPDO4(501h~57Fh),在ED 伺服内部RxPDO1和TxPDO1、RxPDO2和TxPDO2已经被初始化预定义。

注意!目前ED 伺服不支持自发自收功能。

下面介绍PDO 的通讯配置每个PDO 包含三个通讯参数ID号(msg id)通讯类型(type)和禁止时间(inhibit time)。

通讯类型按CANopen 规定如下:

0(0x00): 同步报文(非周期性,当具有同步ID(080h)的报文出现在总线上时,PDO 被更新)

1-240(0x01-0XFO): 同步报文(周期性,要求在同步时间窗内周期性地发送数据,数值表示多少个(1~240)同步报文之后,PDO 将会更新,主要应用于运动控制)

252(0xFC): 由远程帧来的请求命令同步更新PDO 并响应请求

253(0xFD): PDO 连续更新,但只在远程帧请求后发送

255 (0xFF) 非同步 (RxPDO 每次接收时更新, TxPDO 在内容发生变化并不在禁止时间内时就发送) 禁止时间缺省为1000 微妙。

如下图所示:

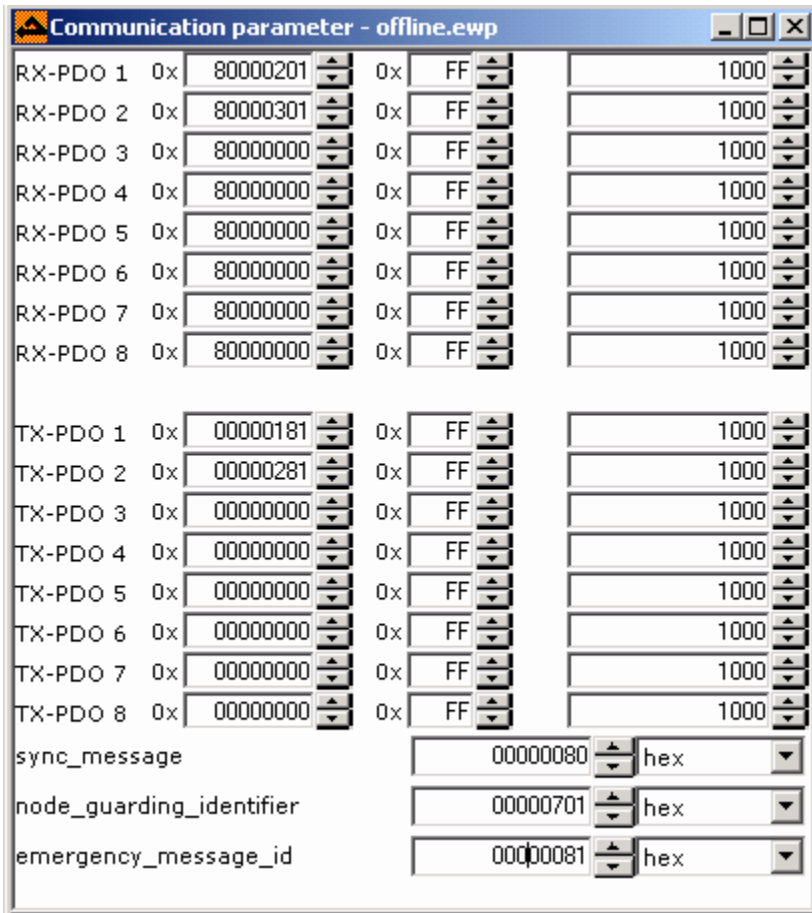


图5.33 Communication parameter

下面为PDO 配置的一个范例:

一个PDO 拥有ID 号为0x201 的CAN 报文,它一定被RxPDO1 所接收, 例如它可能是一个CANopen I/O 模块的输出 (Output),也就是它代表这个I/O 模块的数字输入,这个模块有64 路数字输入,它使用了一个PDO 所有可用的8 个 Bytes 数据,在ED 伺服中,这8 个 Bytes 数据被映射到通用的可映射的对象之中 (INDEX

0x2100,SUB01...16) 0x2100,01 和02 (每个数据字长4 bytes) .

另一个PDO 拥有ID 号为0x181 的CAN 报文应该被作为TxPDO1 而发送, 它将包含32 位的这个I/O 模块的输出值,这个输出值由 0x210003 映射,一旦数值内容有任何变化,0x210003 将被发送,但两次发送间隔不小于1000ms。

第三个PDO 拥有0x281,将在总线上出现Sync-Msg 的信号时被ED 伺服发送。我们要进行测试至少需要两个设备也就是需要一个参与者来给发送端以确认同时对于PDO 通讯需要“启动”CAN 网络,可由加入的“NMT MASTER”设备来启动,如果没有“NMT MASTER”,则必须启动 ED 伺服的 CAN-Node (一个内

部软件模块,这可以通过ED 伺服上电后设置自动将对象词典中0x1F8000 的值设为3(缺 省为0)来启动,此时 ED 伺服会向总线发出“NMT START”信息并且启动它自己,在设置该对象值之前你必须 注意总线上所有其它设备都已经上电并做好接收信息准备,因为多次启动网络对此没有任何影响,所以在总线 中有多个ED 伺服时,最好将每个ED 伺服都进行此项配置。

5. 7. 1 Tx-PDO 映射地址设置

TxPDO 1~8 的对象地址是0x1800~0x1807, 具体三个参数体现在3 个Sub index 中, 分别是:

Index (hex)	Sub index (hex)	名称	含义
1800~ 1807	01	tx_pdo_parameter_id	ID 号
	02	tx_pdo_parameter_type	通讯类型
	03	tx_pdo_parameter_inhibit	禁止时间

表5.18 Tx-PDO 定义

TxPDO 1~8 的Mapping 对应的地址为0x1A00~0x1A07,具体8 个对象地址体现在01~08 共8 个Sub index 中,我们在每个PDO 中可以映射8 个对(最多8 个字节,也就是你可以映射2x32bit 对象,或4x16bit, 或1x32+4x8bits 对象等等, 下图是一个TxPDO 的范例, [1]表示一个对象被定义, [0]表示没有对象被定义, [3] 表示三个对象被定义, 范例中Tx-PDO1 设置了一个对象21000320, TX-PDO2 设置了三个对象: 60FF0020(目 标速度), 60600008(工作模式)和60400010(控制命令)。

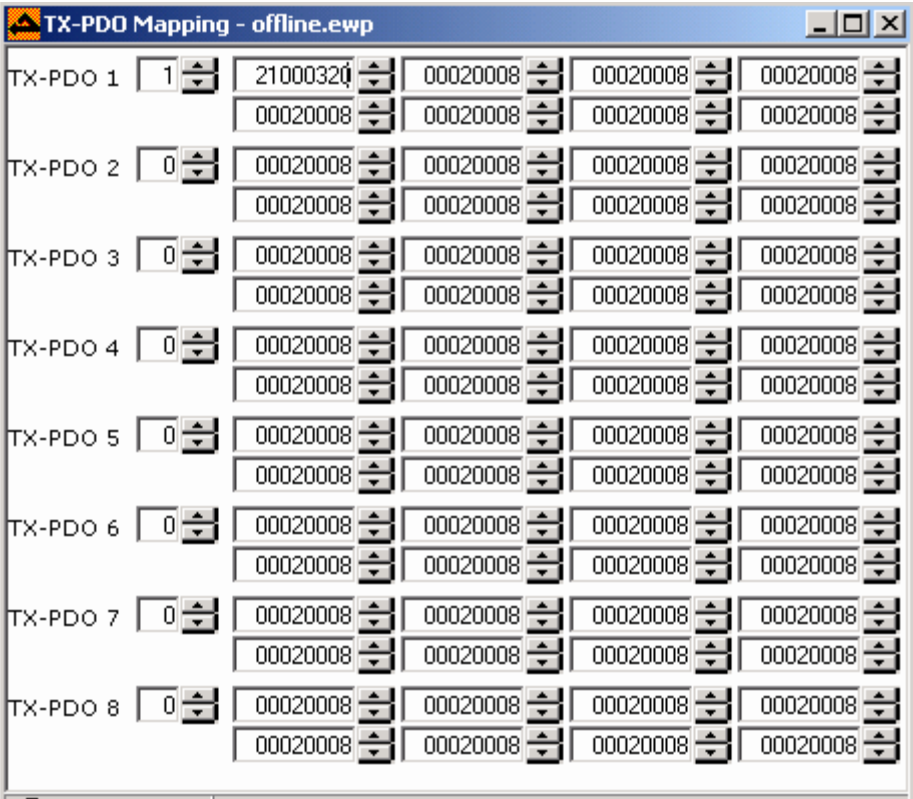


图5.34 TX-PDO 设置

5.7.2 Rx-PDO 映射地址设置

RxPDO 1~8 的对象地址是0x1400~0x1407，具体三个参数体现在3 个Sub index 中，分别是：

Index (hex)	Sub index (hex)	名称	含义
1400~ 1407	01	rx_pdo_parameter_id	ID 号
	02	rx_pdo_parameter_type	通讯类型
	03	rx_pdo_parameter_inhibit	禁止时间

表5.19 Rx-PDO 定义

Rx-PDO 1~8 的Mapping 对应的地址为0x1600~0x1607,具体8 个对象地址体现在01~08 共8 个 Sub index 中我们在每个RDO 中可以映射8 个对 (最多8 个字节)也就是你可以映射2x32bit 对象或4x16bit, 或 1x32+4x8bits 对象等等, 下图是一个 RxPDO 的范例, [2]表示两个对象被定义, [0]表示没有对象被定义, 范例中RX-PDO1 设置了两个对象21000120, 21000220。

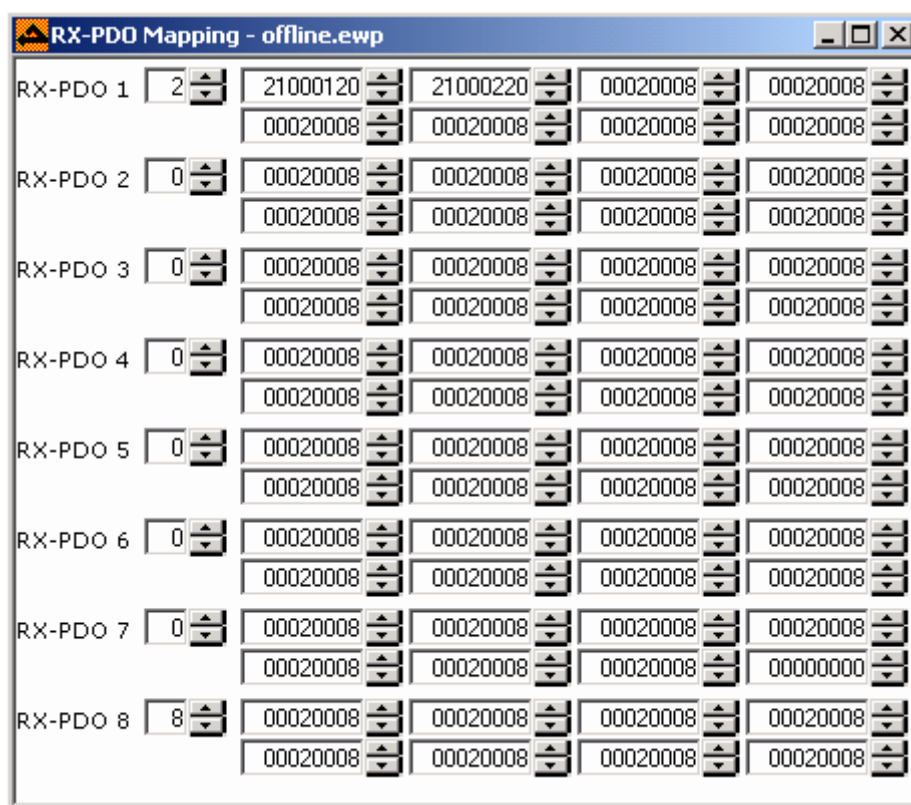


图5.35 RX-PDO 配置

相关地址

1: 0x1F8000 的值设为3 (缺省为0) 来启动, 用PDO一定要将这个地址设置为3;

地 址	子地址	位 数	命令类型	单位	详细解释
2F80	00	08	RW	无符号	对应于 DIP 开关值的地址号 0-127
2F81 /82	00	08	RW	无符号	CANBUS 的波特率 1MBIT/S (40M) 2F81=>0X00 2F82=>0X14 500KBIT/S (130M) 2F81=>0X00 2F82=>0X1C 250KBIT/S (270M) 2F81=>0X01 2F82=>0X1C 125KBIT/S (530M) 2F81=>0X03 2F82=>0X1C 50KBIT/S (1.3KM) 2F81=>0X47 2F82=>0X2F 20KBIT/S (3.3KM) 2F81=>0X53 2F82=>0X2F

2: 波特率一定要和上位机一样

3: 伺服的通讯站号=2F800008+拨码开关, 2F800008 默认为0。

5.8 内部示波器 (Oscilloscope)

如图所示, 本软件带有示波器功能, 用来以动态曲线方式监控运行参数, 例如速度、位置和电流等, 示波器 对我们判断电机的运动状态和调节性能有着非常重要的作用, 对于判断过冲、过载以及速度平稳性等有着非常 直观的效果。下面简单的介绍下它的使用。

1. 设置监控参数:

Oscilloscope 有 4 个监控通道可以同时监控驱动器的 4 个不同参数, 但如果你采用 RS232 和驱动器通讯, 那采集速度会比较慢, 一般我们建议你只选择你最关注的参数, 其他的通道 (“channel”) 取消。

2. 设置采样时间:

scanrate (ms) 参数用于设置采样周期 Number of value 用于设置采样点数, 例如下面的例子就是每 10ms 采集一次数据, 采集 400 次后刷新一下显示数据, 往往真正刷新一次显示的时间会大于上面两个 参数乘积, 这是因为还要考虑通讯的时间。

3. 采样数据条件:

要触发数据的采样, 必须采样条件真实发生, 首先要设置 “Trigger on signal” 条件, 例如下面例子设置 的条件是 “Actual velocity” =0 因为下面例子是电机以 0 位置为中点往返 40000inc 位置运动所以 actual velocity=0 情况是肯定可以发生的, 之所以这样设置, 是为了保证触发条件必然发生, 否则你将采集不 到任何数据。当采集数据时, 鼠标指针会显示为沙漏状态。

在示波器正常工作时, 你需要保证电机已经按照你需要的状态正在运行, 例如下面的例子就是在电机已 经设置为自动正反转, 往返位置 -40000 和 40000, 往返速度 200000inc/s, 工作模式为 -3。



图5.36 Oscilloscope

上图中的例子为一个自动正反转电机速度和位置的监控曲线，往返速度 200000inc/s，位置-40000到

40000inc，工作模式-3，通道1 设定为“Actual position”，通道2 设定为“Actual velocity”，红色线为速度曲线，绿色线为位置曲线。

采集完毕，你还可以将采集到的结果以数据或图片的方式保存下来，供比较。

用于将采集结果以数据的形式导出到一个文本文件中，用于将采集到的曲线保存为图形。

5.9 系统管理（Administration）

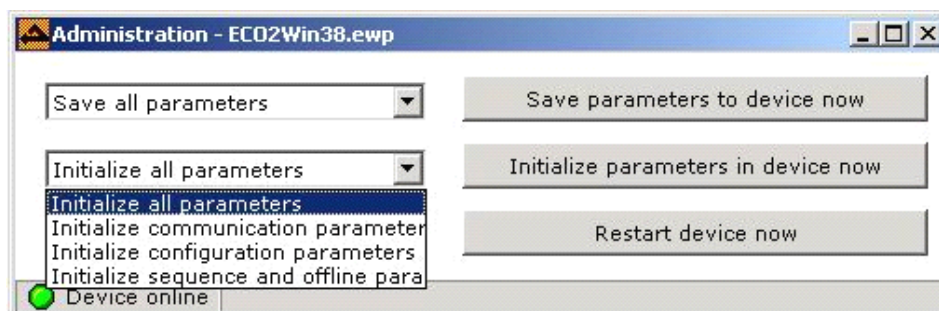


图 5.37

包括参数存储、参数初始化、设备重启动，最常用的是参数存储，当我们修改了 ED 内部参数后，我们如果想要驱动器下次上电时参数不要再次设置或下载，请记住一定要执行参数存储，这与工程项目的存储无关。

如果需要将驱动器内部参数设置为出厂时状态，你可以点击“**Initialize parameters in device now**”按钮，然后必须再点击“**Restart device now**”按钮，点击后，驱动器面板上的“**Error**”指示灯变为红色，几秒后，恢复为绿色，表示初始化操作成功。

第六章 电机试运转

本章我们先让伺服电机转起来，再对驱动器的功能作一些详细的说明。

只有专业人员或者是经过培训的人员才被允许运行 ED 系列伺服驱动器。如有需要步科电气将提供相关的培训。机器的生产厂家对机器的运行进行了必要的危险性分析，并且采取适当的措施确保一些无法预料的动作不会对人身以及财产造成危害。

检查配线是否完整、短路以及接地是否可靠。

所有活动的部分必须有安全防护避免人员触碰。

本系列伺服驱动器工作电压为 AC220V 或 AC380V，不要在带电状况下拔插连接端子。

在机器运转过程中，散热片和驱动器安装底板温度可能会升至 70℃ 以上，所以，在关闭电源，并且温度降到 40℃ 以下前，请避免碰触这些部件。

6.1 电气元件的安装

为了运行电机，需要配备下列各项元件：

- ③ 逻辑电压 24VDC(低压，与 230 VAC 安全隔离)—提供给驱动器控制电路的的逻辑电源。
- ③ 动力电源(24VDC~70VDC)—提供给 ED100 伺服驱动器系列；单相或者三相 220VAC 提供给 ED430 系列伺服驱动器；380VAC 提供给 ED620、ED630 系列伺服驱动器。
- ③ ED100 或者 ED430、ED620、ED630 驱动器 + 接线端子头
- ③ 电机及相关电缆(电机电缆、编码器电缆, RS232 编程通讯电缆)
- ③ 一个带原点和限位开关的机械结构（如导轨）

6.1.1 连接编码器电缆到驱动器

电机编码器信号按照 RS422 定义，编码器由驱动器 eencoder in 口提供 5V 电源，无需外接电源。

编码器电缆接驱动器口，编码器电缆信号定义请参考 ED 伺服选型手册。

6.1.2 连接电机电缆(带或不带抱闸)到电机电缆接驱动器相应接口

两相伺服电机的 4 根相线连接到 X9 的 (A、/A； B、/B)，地线(黄绿线)接 X9 口的 GND，X9 口的 BRAKE+和 BRAKE- 接抱闸线(24 V, 1A)，三相伺服电机 U、V、W、PE 分别接到伺服的驱动器的 U、V、W、PE 端。

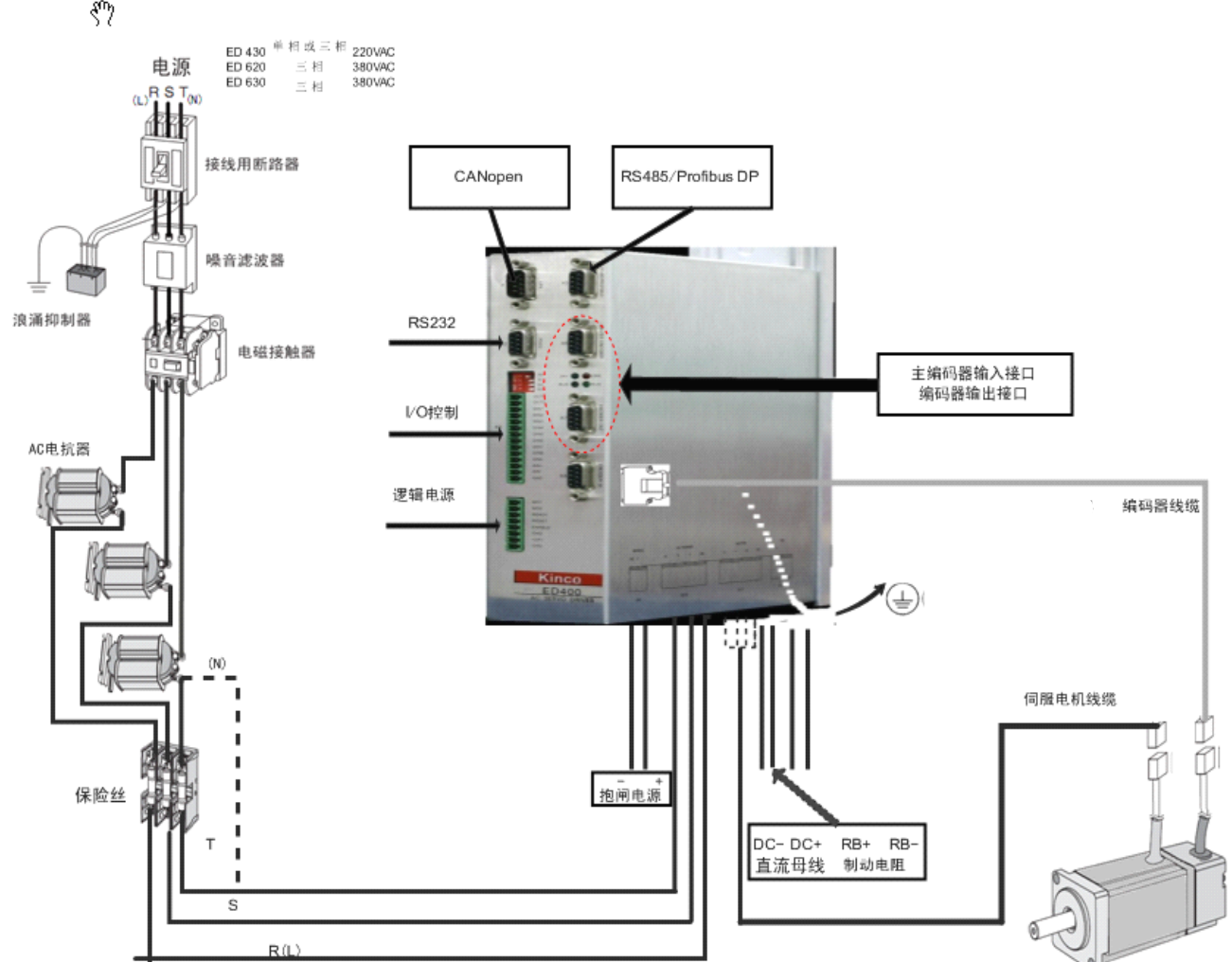
配线要固定在支架上夹紧。务必将电机的屏蔽线(两相 23S 系列电机)接在驱动器上（接 X9 的 GND, 屏蔽线压紧在固定螺丝下, 如图）。具体接线顺序请参考选型手册。

6.1.3 给驱动器提供逻辑电源（控制器的电源 X4）

逻辑电源电压有效范围 18V~30VDC，接 X4 的+24V 和 GND。

ED 伺服驱动器外部接线图请参考下图

Kinco ED伺服系统外部接线图



6.1.4 硬件复位

正确连接之后“RUN”指示灯会闪烁、“24 V”灯会是持续绿色。如果无误，可以通过重新上电或短接 X4 口的 RESET 和 24V 信号消除报警。

6.1.5 编码器信号检查

驱动器此时工作在它的基本状态。 如果将 RS232C 连接 PC，打开 Eco2win 软件， 在手动旋转电机轴的时候注意编码器数据 (Axis1->Device control->Movement->Actual position) 的变化，如果该处数据没有变化，请重新检测编码器电缆是否连接正确。如果 Master Encoder 口也接有一个主编码器，旋转主编码器就可以在功

能模块 (Device configuration->Electronic gear->Actual position master) 里监控到主编码器数据的变化。如果数据没有变化, 说明主编码信号不正确, 不符合 RS422 标准。

6.1.6 连接外部使能信号

X4 口的 ENABLE 为外部使能口, 可以直接接驱动器的+24V 到 ENABLE 端子。如果 ENABLE 没有接, 驱动器输入使能命令后将报 “external enable low” 故障。

6.1.7 给驱动器提供动力电源

接通驱动器的动力电源, 对 ED100, 动力电源范围: 24VDC~70VDC, ED430 动力电源为 220VAC, ED620 ED630 系列动力电源为 380VAC。接线方法见 3.2 章节相关内容。

6.1.8 连接编程电缆

编程电缆可以从我们公司购买, 也可以按照下面的定义自行制作。

编程电缆:

PC COM1,COM2		ED RS232
RxD 2	-----	2 TX
TxD 3	-----	3 RX
GND 5	-----	5 GND

这样, 电机运转的电气接线基本完成。

6.2 电机试运转

在电机试运转前, 请务必确认:

- ③ 所有连接到驱动器的电源必须与总线电源隔离。
- ③ 实际最大电压不能超过额定值。
- ③ 试机前务必使电机轴端悬空, 也就是不要带负载。

除了上述要求外, 在电机动作之前, 我们还要检查并确认参数设置是否恰当, 最大电流是否依照电机型号正确设置, 如果这些都已完成, 我们就可以开始运行电机了。

6.2.1 电机参数配置

首先利用 ECO2WIN 软件建立一个新的工程文件或打开一个已有的工程文件, 参考 4.2.2 和 4.2.3。ED 伺服驱动器在出厂就已经设置好正确的电机型号, 联线后在 Parameter setting->Commutation 窗口 (5.4.4 节) 中就会有下面的内容:

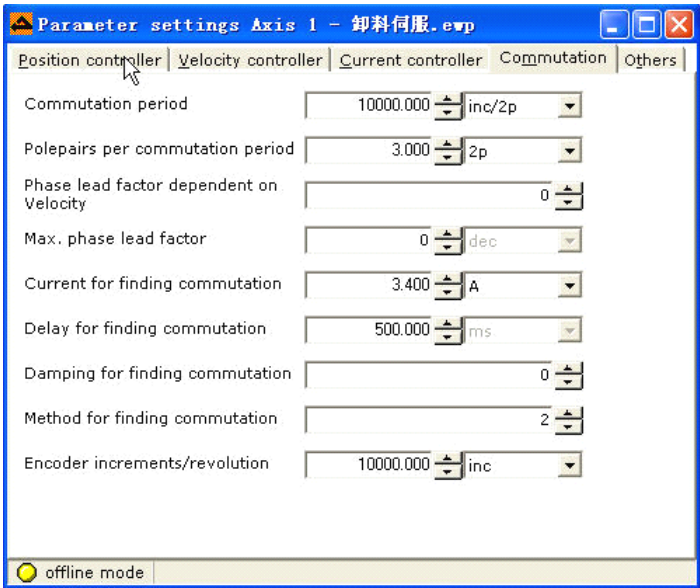


图 6.4

检查各个参数是否跟所用电机相对应。尤其是 commutation period 、 polepairs per commutation period、Encoder increments/revolution 三个参数。这些参数默认出厂都已经设置好，不建议做任何修改，只做核对。在如图 6.5 所示的画面里。按照下表所列的电机型号设置参数。

电机型号	Commutation period	Polepairs per commutation period	Current for finding commutation	Damping for finding commutation	Method for finding commutation	Encoder increments/resolution
SMH60S-0020-30AAK-3LKL	10000	3	686	0	3	10000
SMH60S-0040-30AAK-3LKL	10000	3	665	0	3	10000
SMH80S-0075-30AAK-3LKL	10000	3	836	2	3	10000
SMH80S-0100-30AAK-3LKL	10000	3	1108	2	3	10000
SMH110D-0105-20AAK-4LKC	10000	4	950	5	3	10000
SMH110D-0125-30AAK-4LKC	10000	4	1143	5	3	10000
SMH110D-0126-20AAK-4LKC	10000	4	1090	5	3	10000
SMH110D-0126-30AAK-4HKC	10000	4	692	5	3	10000
SMH110D-0157-30AAK-4HKC	10000	4	868	5	3	10000
SMH110D-0188-30AAK-4HKC	10000	4	997	5	3	10000
126D-0180-20AAK-3HG	10000	3	1201	5	3	10000
130D-0105-20AAK-2LS	10000	2	901	5	3	10000
130D-0157-20AAK-2LS	10000	2	1053	5	3	10000
155D-0250-20AAK-3HG	10000	3	1094	5	3	10000
155D-0300-20AAK-3HG	10000	3	1287	5	3	10000
155D-0400-20AAK-3HG	10000	3	1394	5	3	10000

表 6.1

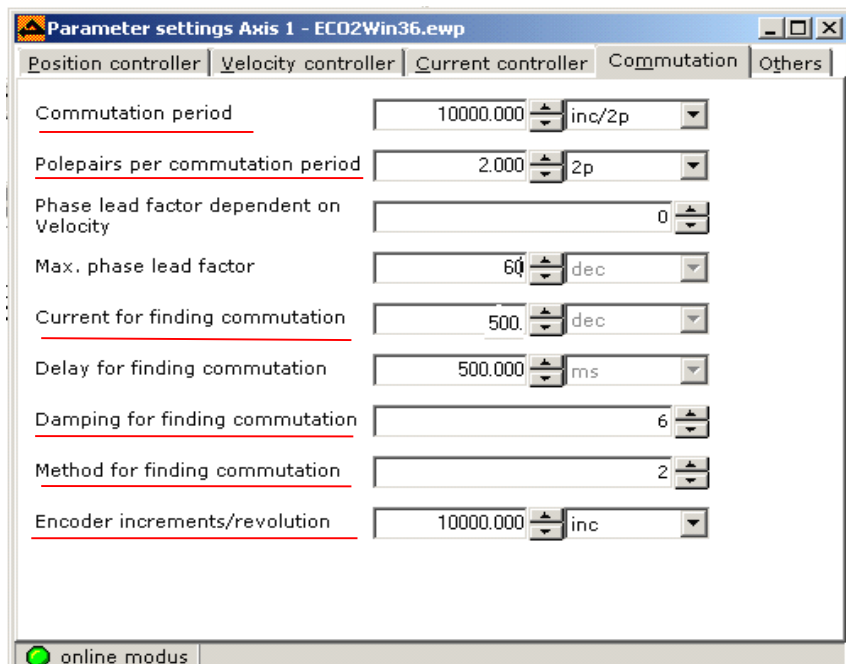


图 6.5

6.2.2 电流调整

调整最大电流。这些参数默认出厂都已经设置好，不建议做任何修改，只做核对。注意，一定要设定 **I²T** 保护参数。在如图 6.6 所示的画面里。按照下表所列的电机型号设置参数。

电机型号	Max. current	current value for iit protection	time constant for iit protection
SMH60S-0020-30AAK-3LKL	2047	755	100
SMH60S-0040-30AAK-3LKL	1994	731	100
SMH80S-0075-30AAK-3LKL	2047	920	100
SMH80S-0100-30AAK-3LKL	2047	1219	100
SMH110D-0105-20AAK-4LKC	2047	1045	100
SMH110D-0125-30AAK-4LKC	2047	1258	100
SMH110D-0126-20AAK-4LKC	2047	1199	100
SMH110D-0126-30AAK-4HKC	2047	761	100
SMH110D-0157-30AAK-4HKC	2047	955	100
SMH110D-0188-30AAK-4HKC	2047	1097	100
126D-0180-20AAK-3HG	2047	1321	100
130D-0105-20AAK-2LS	2047	991	100
130D-0157-20AAK-2LS	2047	1158	100
155D-0250-20AAK-3HG	2047	1203	100
155D-0300-20AAK-3HG	2047	1415	100
155D-0400-20AAK-3HG	2047	1533	100

表 6.2

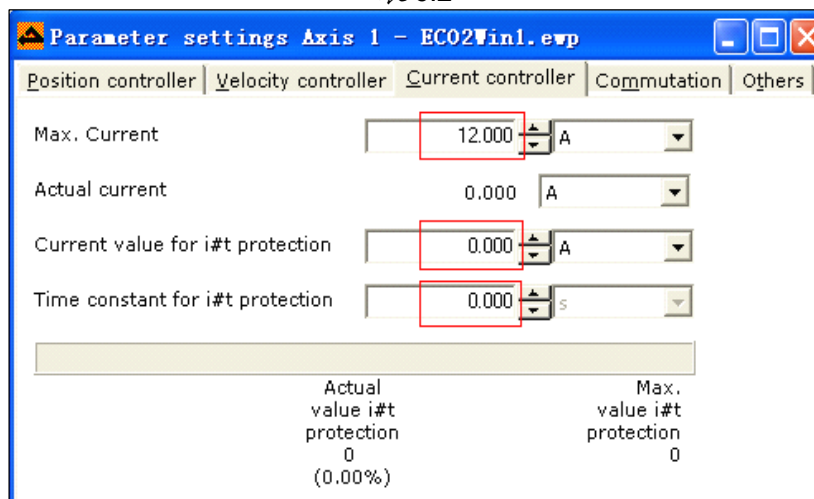


图 6.6

6.2.3 检查速度环和位置环参数

最后，我们检查一下速度环和位置环的参数默认值，下面是对应空载参数设置：

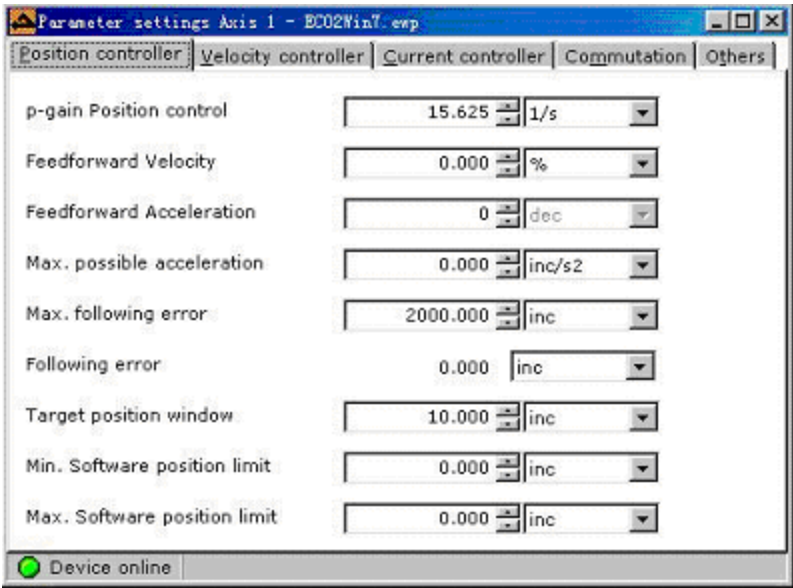


图 6.7 位置环默认参数

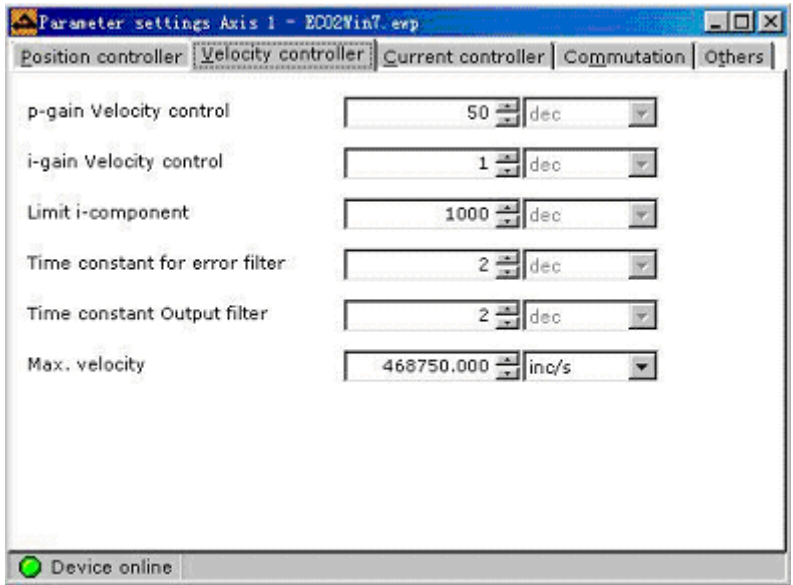


图 6.8 速度环默认参数

6.2.4 状态控制

现在我们已经完成了参数的设置，可以让电机开始动作了。在“Device control”中首先选择工作模式为速度模式 3, 然后设置目标速度 (Target velocity) =8000inc/s, 设置 Control word=0x06, 如果驱动器没有报警，则说明接线正常，再设置 Control word=0x0f, 此时电机就以设定的速度旋转。此时，你可以直接修改 Target velocity 的值，如果要停止电机旋转，可以将 Target velocity=0, 这时电机轴仍然是锁紧的。如果要松开电机轴，则只需设置 Control word=0x06。

参考第 5.3 节，尝试用不同的模式控制电机。

步骤：

接通驱动器逻辑和动力电源。

- 1. 轴是自由的，此时不要带负载。

2. “24V” LED 绿色、“RUN” LED 闪烁绿色
3. 在 Device Control 中设置工作模式为 3,控制命令=0x06
4. 控制命令=0x0F
5. 在 Device State Control 中 bit Commutation Found 是否为 1 或状态字值=0x4437 以及电机轴是否被锁住
6. 设置 Target velocity=10000inc/s (编码器分辨率为 10000)
7. 此时电机将以 60RPM 的速度旋转。
8. 如果驱动器报警,根据 5.3.6“错误诊断”一节分析故障原因。在默认参数下,驱动器很容易报“following error”故障,这时需要适当的调整位置环的 Kpp (“p_gain Position control”一栏)和 Vff (“Feedforward velocity”一栏)值,一般情况下,可以将 Kpp 适当调整(根据负载惯量大小适当调整,一般不超过 200),Vff 调整到 75%以上, following error 就会变的很小,驱动器也不再会报警“Following error”。具体对每个参数的调整,需要通读本手册并具备一定的现场经验。好在,需要我们调整的参数不多。

第七章 限位开关与原点控制

本章讲述了如何去配置末端限位和如何选择不同的原点寻找模式，这对一个有着起始和终止位置的系统是非常重要的。

7.1 限位开关和原点开关

设置一个直线或旋转轴的参考原点，至少需要一个电气“开关”作为“原点信号”，“开关”可以是机械或电气传感器。

输入 DIN6-DIN8，为限位和原点开关。

输入口	作用
DIN6	正限位开关
DIN7	负限位开关
DIN8	位于限位开关之间的原点开关

除了硬件接线外,要想设置 DIN6 和 DIN7 作为正限位和负限位输入,还需要在 Device configuration->Digital input 中设置相应的参数，如下图：

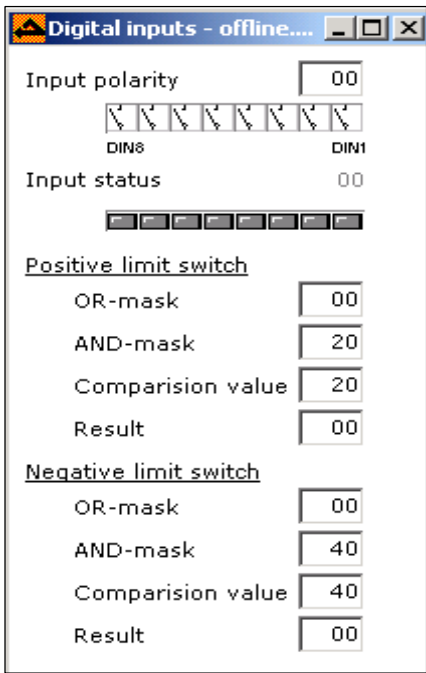


图 8.1 Digital inputs

如果要设置正限位有效，需在 **Positive limit switch** 的 **AND-mask** 内填入 **20**。

如果要设置负限位有效，需在 **Negative limit switch** 的 **AND-mask** 内填入 **40**。

原理：正限位 **DIN6** 填入的 **20** 为 **16** 进制，转化 **2** 进制 **0010 0000**，刚好是第 **6** 位。

负限位 **DIN7** 也是这个道理，**40HEX** 转化 **2** 进制 **0100 0000**，刚好是第 **7** 位。

7.2 原点查找原理

控制模式 6 为原点查找模式，在该模式下 EC02WIN 自动设定为查找原点。

在模式 6 下的对象主要有：

对象	名称	值	含义
60600008	Chosen Mode of operation	6	设定工作模式为模式 6
60980008	Homing method	用户设定	寻找原点方式
607C0020	Relocation zero point	用户设定	相对于原点的偏移量
60990120	Velocitiy during search for Reference swith	用户设定	寻找限位或原点开关的速度
60990220	Velocitiy during search for Reference set point	用户设定	寻找电机 index 信号的速度
609A0020	Acceleration during homing	用户设定	在查找原点过程中的加速度
60400010	Control word for Device State	1F	该模式下电机开始动作（即开始寻找原点）

表 7.1 原点对象定义

注：所有的开关需要一个+24V 的电平信号激活（激活为高时）。

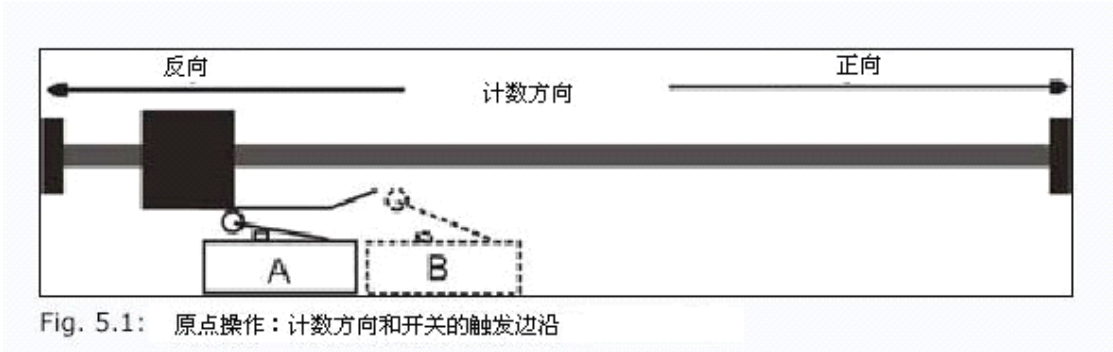


图 7.2 原点

图 7.2 中的计数和运动方向在下面叙述中相同，用户系统的计数方向可以通过 “Axis1->Device Control->Movement, Actual position” 判断，如果 Actual position 值增加则为正向。如果与用户期待的方向相反，用户可以设置地址值：607E0008，默认为 “0”，要修改计数方向，将该值修改为 “80”。也可以通过 “Parameters setting->Others” 中 “Position polarity” 和 “Velocity polarity” 改变，这点在前面已经介绍过。

7.3 原点模式

注：下面的 index 信号就是电机编码器的 N 相信号，或称之为电机编码器的 Z 相信号

7.3.1 原点模式 1：参考负限位开关信号的原点模式

DIN7 接负方向限位开关信号：

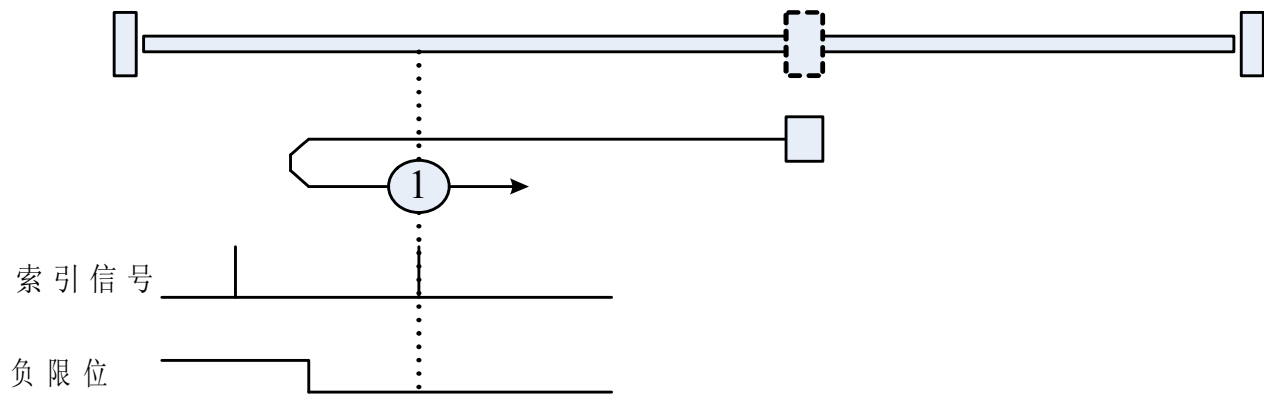


图 7.3 原点模式 1

限位开关处于负计数方向。当系统运行进入负方向直至限位开关触发+24V 时，立即进入正方向直至电机编码器发出第一个 Index 脉冲信号为止，在负方向距离停止位置最近的一个 Index 对应的位置就是该方式下的原点，此时，“reference found” 的状态位被置位，系统减速到停止。

优点：参考点完全与电机和机械关联。

缺点：如果更换电机或联轴器松弛，系统需重新定位原点。

7.3.2 原点模式 2：参考正限位开关信号的原点模式

DIN6 接正方向限位开关信号：

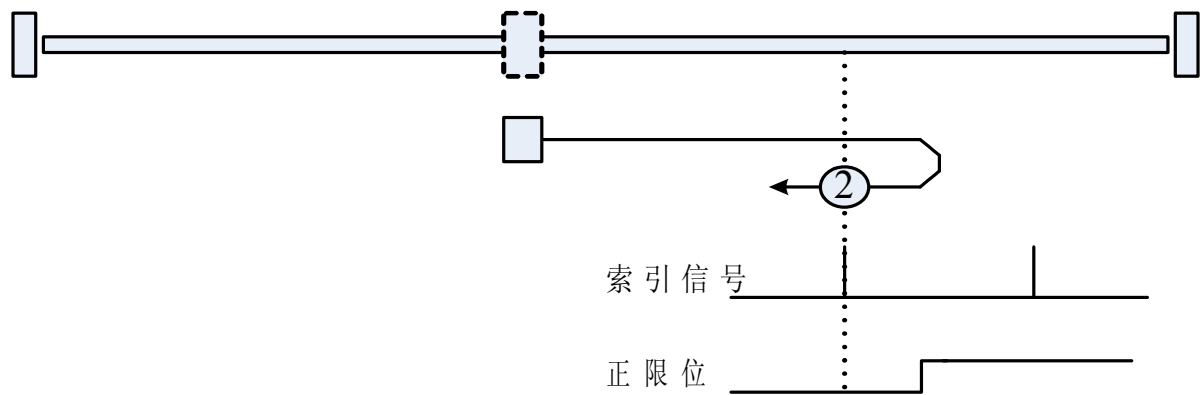


图 7.4 原点模式 2

限位开关处于正计数方向。当系统向正方向运动直至限位开关触发+24V, 然后立即向负方向运动直至电机编码器发出 INDEX 脉冲为止，在正向离停止位置最近的一个 Index 对应的位置就是该方式下的原点，此时“reference found” 的状态位被置位，系统减速到停止。

优点：参考点完全与电机和机械关联。
缺点：如果更换电机或联轴器松弛，系统需重新定位原点。

7.3.3 原点模式 3 和 4：参考正向原点开关和 index 信号的原点模式

DIN8 接位于正方向位置的原点开关信号：

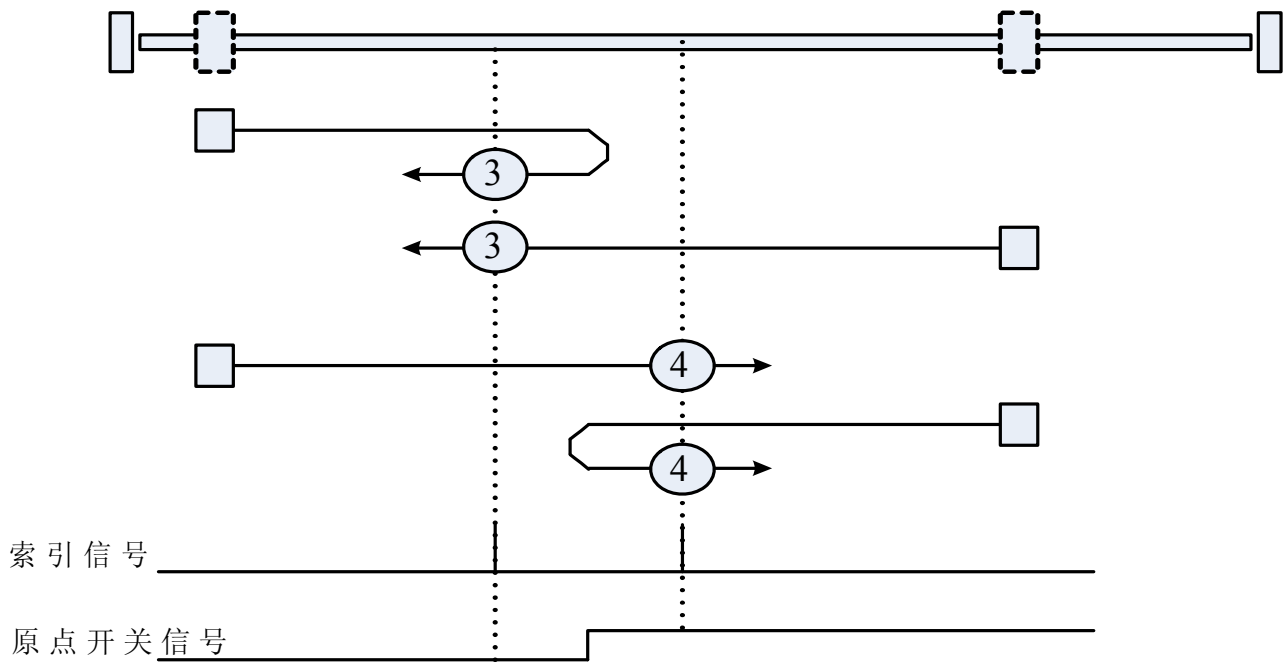


图 7.5 原点模式 3 和 4

往 Homing 开关方向运动，在检测到 Homing 开关的指示信号后停止，根据 Homing 开关的电平再正转或反转退开，一旦检测到 Homing 电平变化，电机的下一个 Index 信号将被记录为原点。“reference found”的状态位被置上，电机减速停止。

优点：原点完全与电机和机械关联。
缺点：如果更换了电机或联轴器松弛，系统需重新定位原点。

7.3.4 原点模式 5 和 6：参考负向原点开关和 index 信号的原点模式

DIN8 接位于负方向位置的原点开关信号。

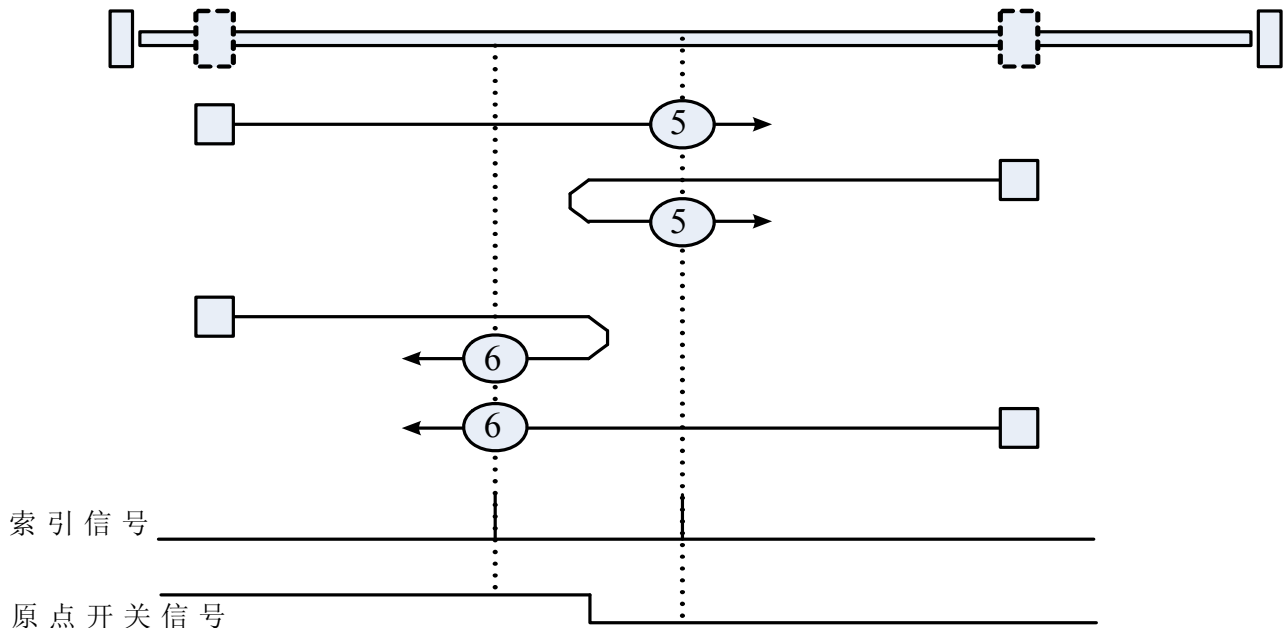


图 7.6 原点模式 5 和 6

往 Homing 开关方向运动，在检测到 Homing 开关的指示信号后停止，根据 Homing 开关的电平再正转或反转退开，一旦检测到 Homing 电平变化，下一个的马达 Index 信号将被记录作为参考点。“reference found” 的状态位被置上，电机减速停止。

- 优点：原点完全与电机和机械关联。
- 缺点：如果更换了电机或联轴器松弛，系统需重新定位原点。

7.3.5 原点模式 7 和 10：参考原点开关、index 信号和起始正向运动的原点模式

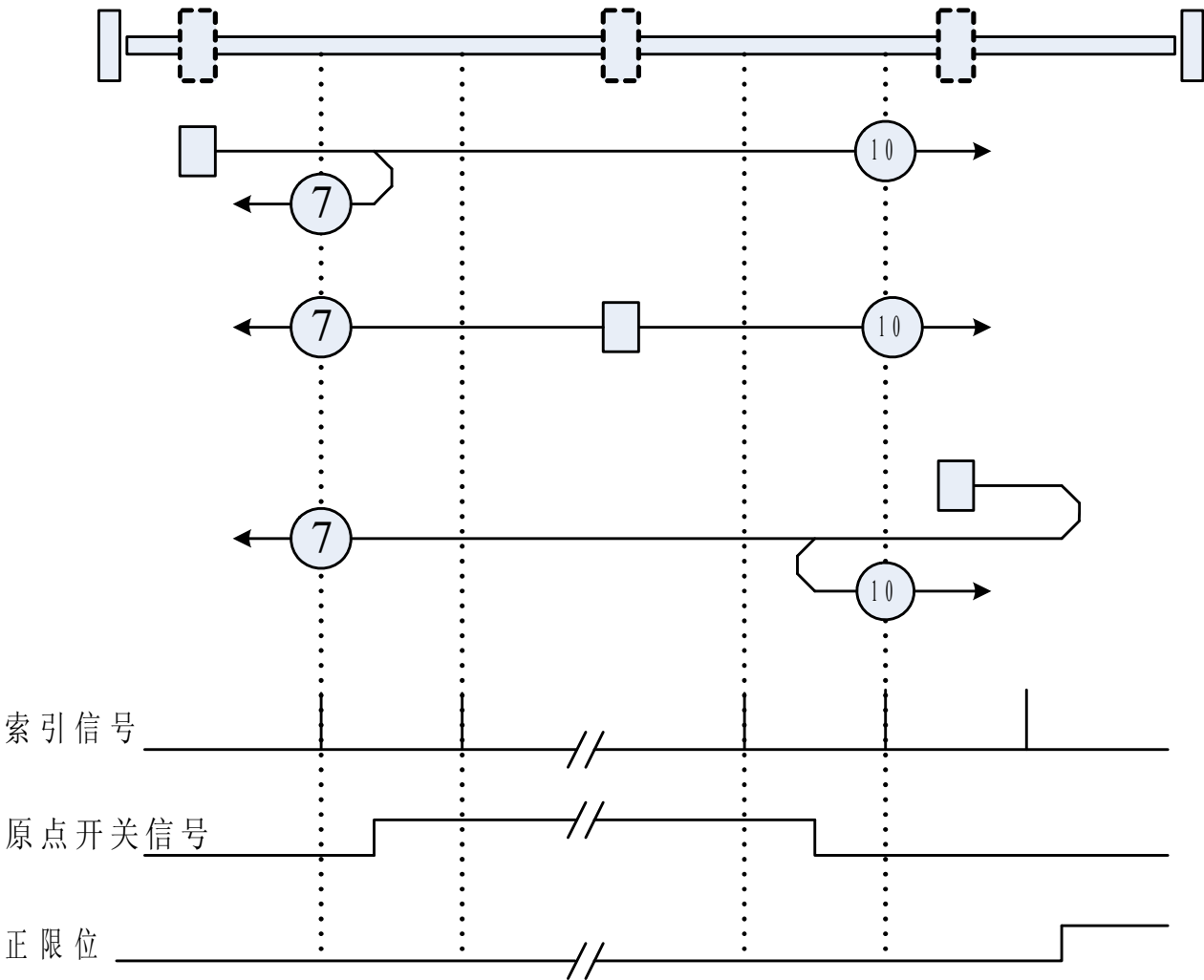


图 7.7 原点模式 7 和 10

图 7.7 以 DIN8 原点开关信号和 index 信号确定的原点方式，初始运动方向为正向。

7.3.6 原点模式 11 和 14：参考原点开关、index 信号和起始负向运动的原点模式

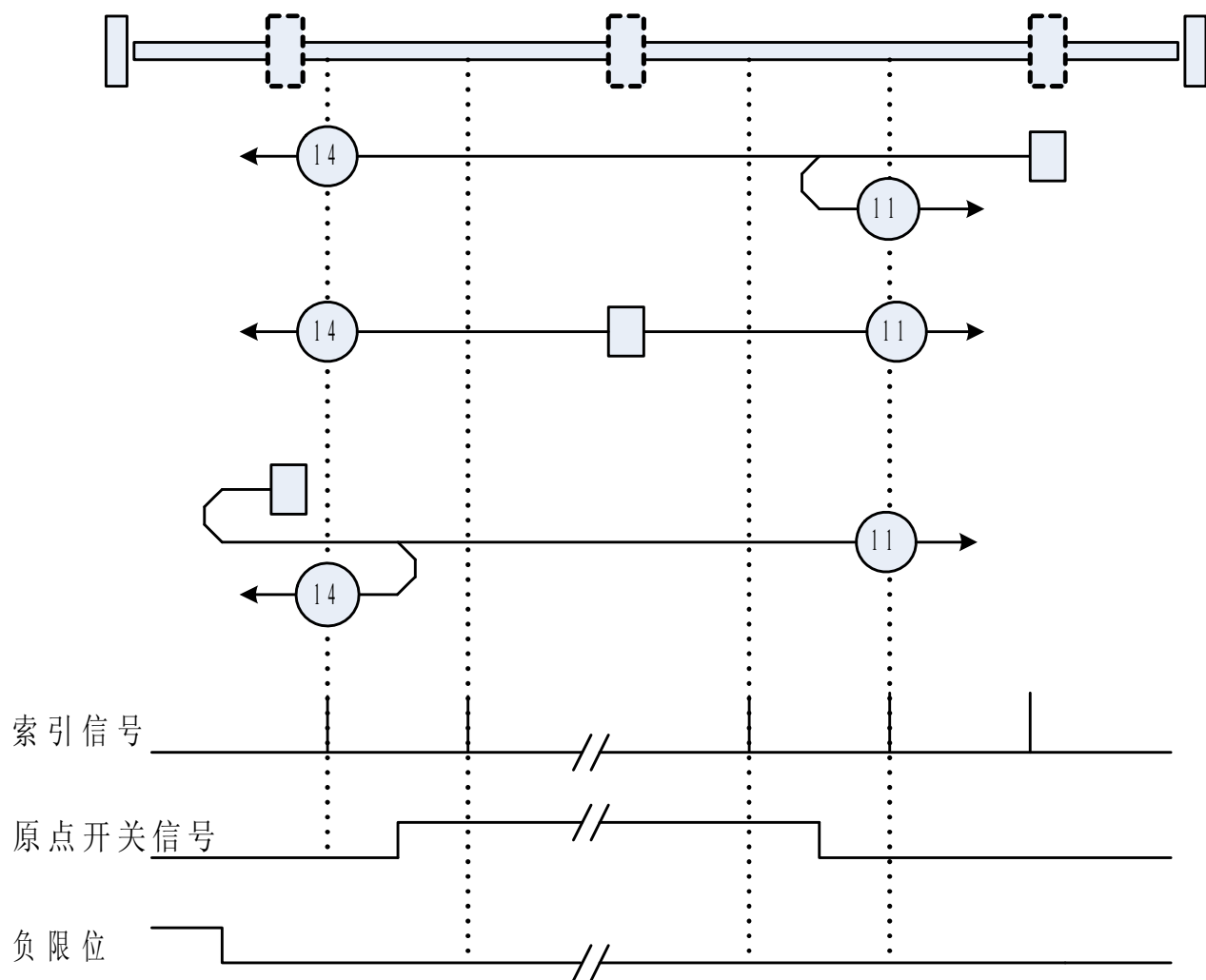


图 7.8 原点模式 11 和 14

图 7.8 以 DIN8 原点开关信号和 index 信号确定的原点方式，初始运动方向为负方向。

7.3.7 原点模式 15 和 16：保留

供将来扩展为其它的原点方式使用。

7.3.8 原点模式 17-31：没有 index 信号的原点模式

原点模式 17~31 与原点模式 1~14 相似，只是不需要 index pulse 信号。替代 index pulse 信号的是限位开关或原点开关的 H-L 下降沿和 L-H 上升沿。

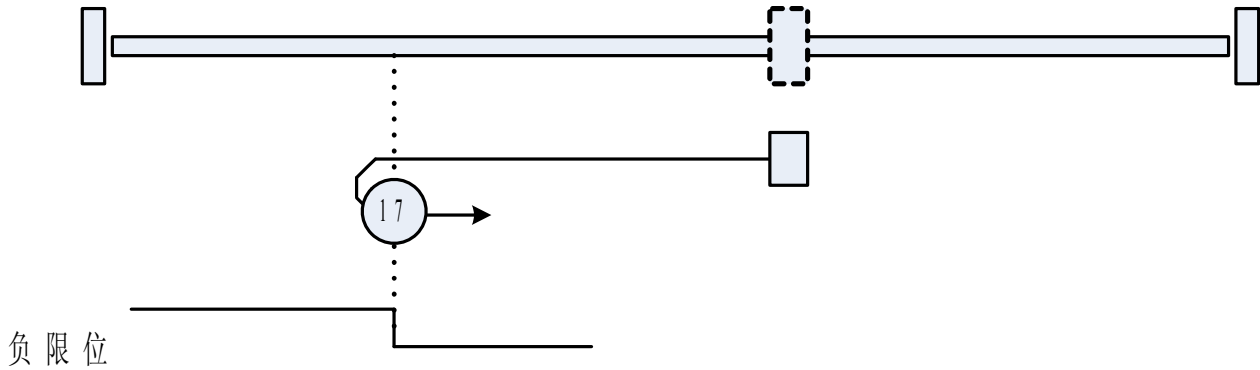


图 7.9 原点模式 17

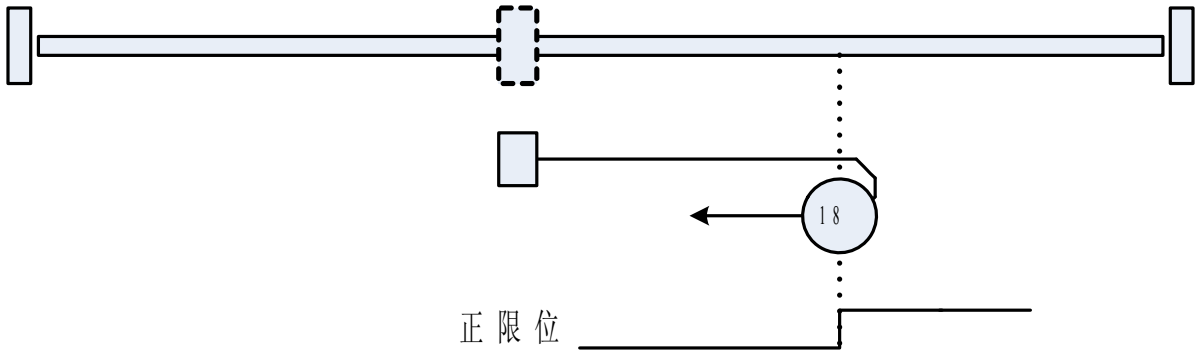


图 7.10 原点模式 18

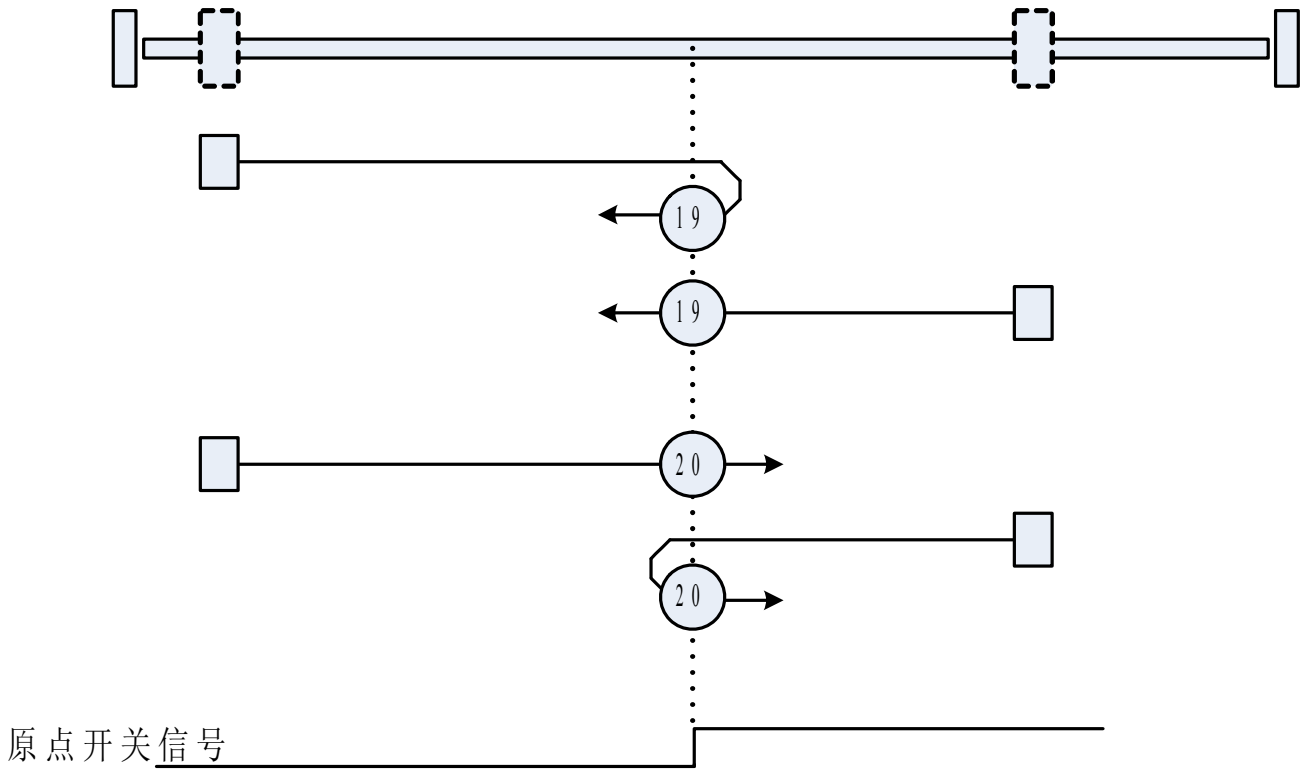


图 7.10 原点模式 19 和 20

带 Index Pulse 和不带 Index Pulse 信号的原点模式对比：

模式 1 和 2	模式 17 和 18
模式 3 和 4	模式 19 和 20
模式 5 和 6	模式 21 和 22
模式 7 和 10	模式 23 和 26
模式 11 和 14	模式 27 和 30

优点：原点依赖机械位置，不受电机影响。

缺点：机械开关容易变形，导致上升或下降沿位置变化，如果要使用这种方式，机械开关需可靠。

7.3.9 原点模式 32 和 33：由 index 信号确定的原点模式

从当前位置，电机继续反向或正向运动到相邻的 index pulse 信号，该 index pulse 信号就是参考原点。

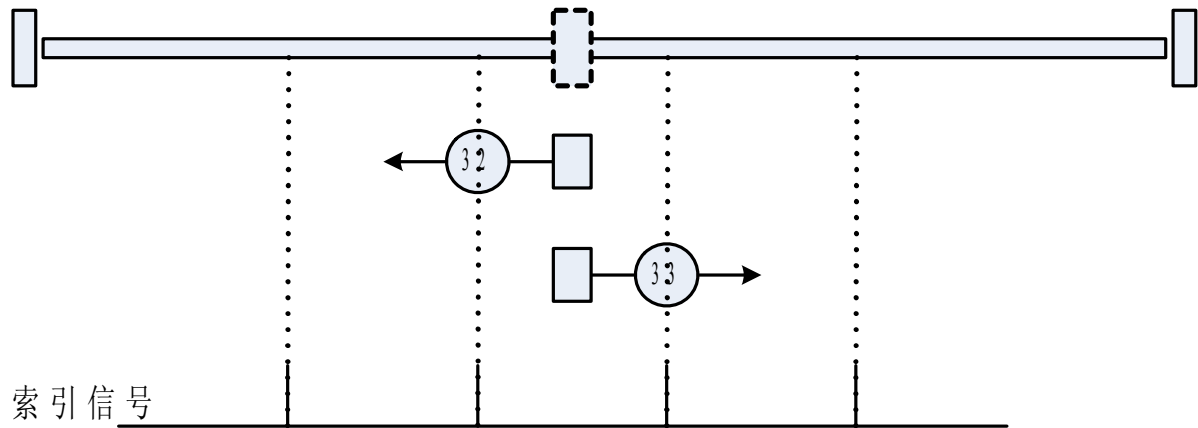


图 7.11 原点模式 32 和 33

7.3.10 原点模式 34：以当前位置做为原点的原点模式

定义当前位置为参考原点。

7.3.11 原点模式-17 和-18：参考机械末端位置为原点的原点模式

这两种模式依赖于机械末端位置，该位置为参考原点。

模式-17：电机带动负载向负方向运动直到遇到末端障碍停止，机械末端停止位作为参考原点。

模式-18：电机带动负载向正方向运动直到遇到末端障碍停止，机械末端停止位作为参考原点。

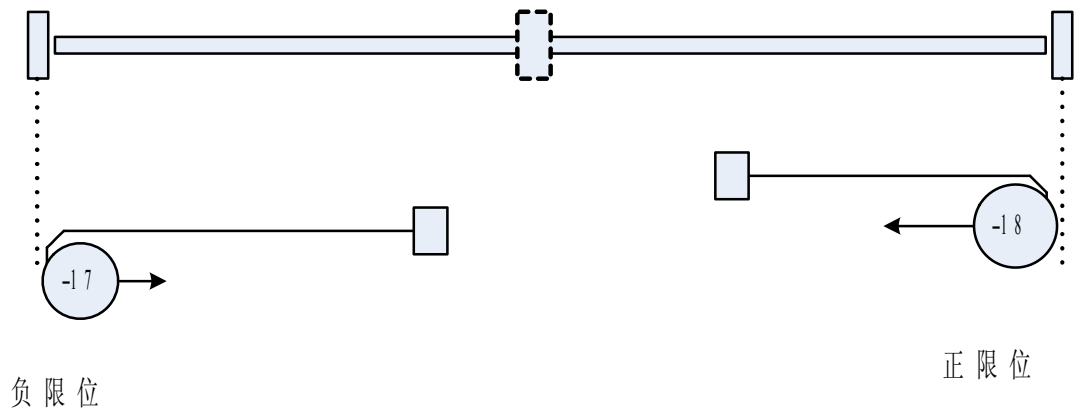


图 7.12 原点模式-17 和-18

优点：减少开关和传感器接线，机械末端位置可重复作为原点。

缺点：机械位置会变化，准确性不高。

7.3.12 原点模式-1 和-2：参考机械末端位置和 index 信号的原点模式

模式一1：电机带动负载向负方向运动直到碰到机械末端障碍，然后正向运动直到找到电机的 Index 信号，找到的 Index pulse 信号作为原点信号。

模式二2：电机向正方向运动直到碰到机械末端障碍，然后反向运动直到找到电机的 Index 信号，找到的 Index pulse 信号作为原点信号。

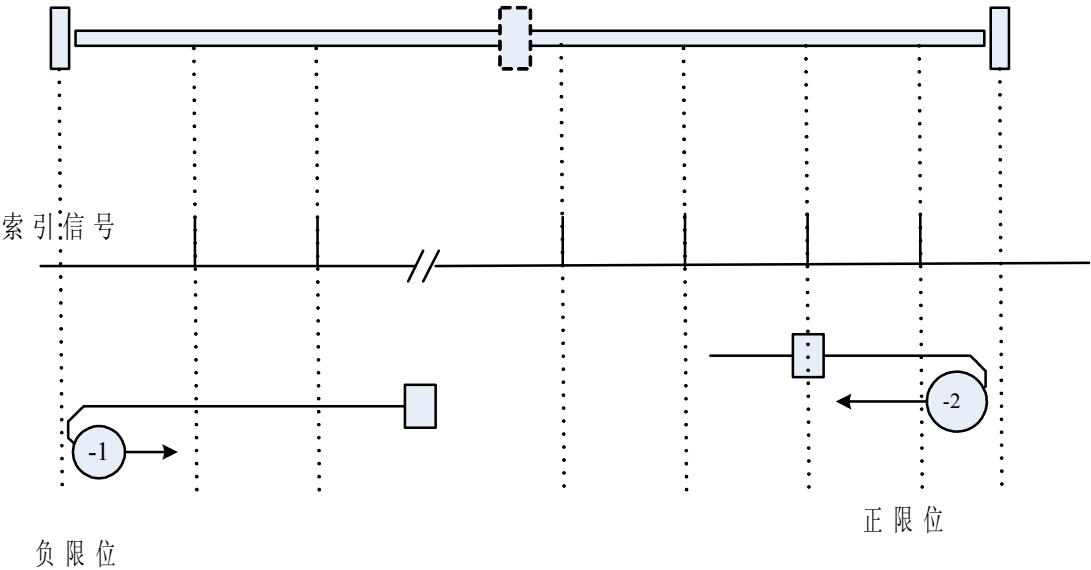


图 7.13 原点模式-1 和-2

优点：无需外接开关，原点与机械位置有关，机械的末端可以通过添加缓冲加以保护，利用 Index 信号，具有很高的重复精度。

缺点：在更换电机和联轴器松开后，需要重新校准原点。

7.4 手动查找原点

如果在“Control Axis”窗口的“Homing”页，在点击了“Reference Start”（“Start homing”）按钮后，该按钮下面的第2个指示灯变绿，显示文本“found reference”，则说明找到原点。如下图：

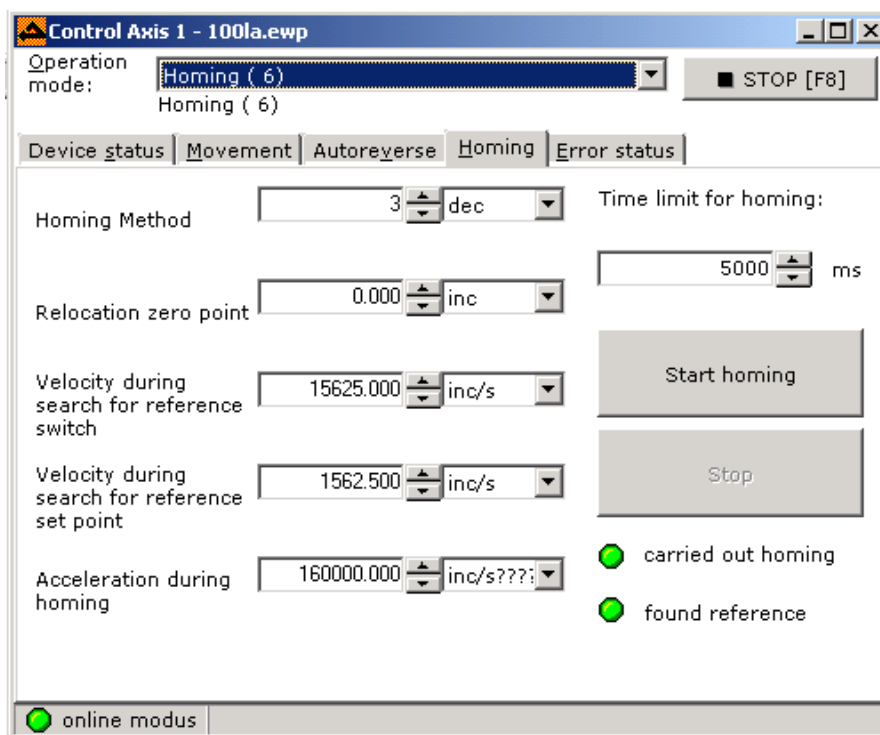


图 7.14 原点设置

如果没有找到原点，请检查下列值并修正。

- ③ 找原点的时间限制（Time limit for homing）是否过短
- ③ 设定寻找外部原点开关的速度
- ③ 找电气参考原点(Index 信号)的速度
- ③ 寻找原点的加速度

如果电机不动作，请检查：

- ③ 电机断电后是否可以连同机械自由运动

7.5 Sequence 编程找原点范例

注意：不要在同一个 **seq** 中出现两个相同地址。（除了计算器 **21A0XXXX** 之外）。

因为上电的时候伺服要有一个 **ready** 的过程，所以如果上电就要执行回原点程序的时候我们都会先将 **60400010=6**，然后延迟 **1000ms** 才调用回原点程序。如下

先将 **21500010=8001**（上电调用 **seq1**）

Seq1

60400010	6	先让伺服驱动器 ready
60980008	3	回原点模式，或称之为回原点方式
607C0020	0	原点的偏移量，默认为 0，所以如果将外部参考点设为 0 位置的时候这个参数可以不填。
60990120	10000	回外部参考点的速度
60990220	2000	回 N 相信号的速度此速度不宜太快，否则会产生比较大的过冲。
609A0020		回原点加速度，都可以不写，直接用默认值。
21300220	1000	延迟 1000ms 调用 seq2
21300110	8002	跳转到 seq2

Seq2

60600008	6	工作模式 6 为回原点
60400010	1F	找原点开始
21400210	8003	原点找到后跳转到 seq 3

Seq3

60600008	1	工作模式 1 为定位模式
607A0020	0	目标位置 0
60810020	10000	定位速度
60400010	3F	绝对定位

Seq3 中这样处理是因为找完原点后，伺服电机由于惯性会继续向前运转一点，从而实际位置 **60630020** 数值不是 **0**，好像伺服电机没有成功找到原点，这其实是正确的，伺服系统已经把之前的位置当做系统原点。出于编程便利考虑，用户可以在找完原点后利用定位模式绝对定位到 **0**，这样系统实际位置也就是 **0** 了！

第八章 性能调节

8.1 正反转介绍

在该模式下，电机将按照设定的模式不断进行正反转，用户可以在此模式下调整速度环和位置环参数，以期伺服发挥更好的控制性能。使用正反转模式调参数前请确认负载是否允许正反转，另外请确保驱动器电源在任何时刻都可以立即断开，以免发生故障。

PC 连上驱动器后，右键点击驱动器图标，进入到“Axis1- Device control-Autoreverse”。窗口“Auto reverse”分两部分，上部分设定极限值，下部分设定目标值。在达到极限值后，电机反方向向新的对应目标值运动。

ECO2WIN 软件是 ED 驱动器和用户的一个沟通窗口，所有的控制命令都存储在驱动器内部，不会因为 ECO2WIN 软件退出而丢失。换句话说，即使 ECO2WIN 和 Windows 软件无法正常工作，驱动器也会照常工作。所以请确保安全。

8.2 各种正反转控制介绍

8.2.1 带位置控制的定时正反转

打开窗口“Control Axis1- Autoreverse”。

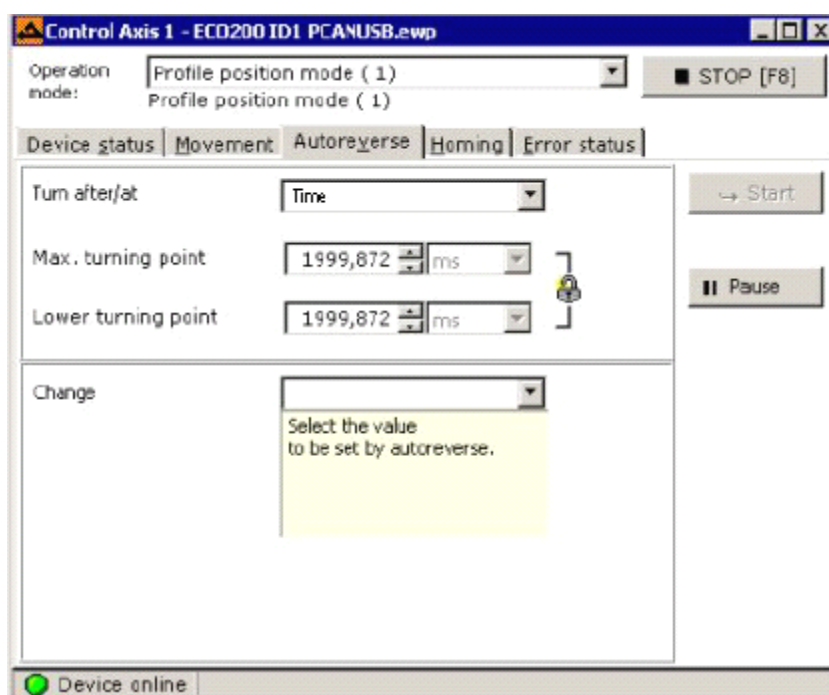


图 7.1 Autoreverse

1. 在 Max. turning point 输入值，例如 2000ms;

2. 点击锁标记, Lower turning point 一栏输入值自动设定等于 Max. turning point 中的值。

显示的时间值与输入的值略有不同, 这与内部转换有关 (即: 1999,872 替代 2000);

在下部分“Change..”下拉框选择“Position”:

1. 在“Positive movement”输入一个正向位置值, 切记输入的值不会引起机械碰撞。
2. 在“Negative movement”输入负向位置值, 同样记住输入的值不会引起机械碰撞。
3. 如果点击右边锁标记, 成锁住状态, 那么“Positive movement”和“Negative movement”内值是一对正负对称值。
4. 切换到“Movement”, 首先设定“Velocity on positioning”一个较小值。
5. 按下“Autoreverse”窗口的“Start”按钮, 启动电机。
6. 按下“Pause”按钮, 电机会停止正反转运动, 但电机轴仍然是锁紧的。

注: 按“F8”和“STOP”按钮, 电机轴将立即松开。

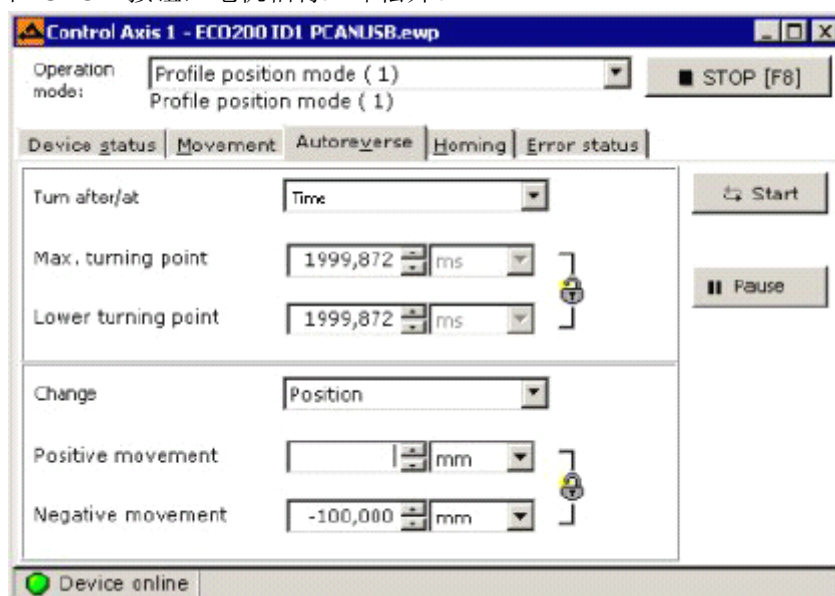


图 7.2 Autoreverse (自动正反转操作)

上面的数据设定电机以 2000ms 定时正反转, 在 2000ms 内电机以:

- ③ Movement 中设定的加速度加速到定义速度
- ③ Movement 中设定的减速度减速
- ③ 在指定的位置处停止

如果按照设定的加减速和目标速度, 到达正向或负向的位置小于 2000ms, 那电机会暂停, 直到 2000ms 时间到再开始反方向运动到设定位置。在电机运动期间, 可以设定上面的时间值和下面的位置值。

在“Movement”窗口中可以改变下面参数:

- ③ Velocity
- ③ Acceleration
- ③ Deceleration

如果驱动器“ERR”LED 指示灯闪烁，在这种情况下一般是“Following error”故障，可以在“Error status”查看具体故障。此时，可以减小“Movement”中速度、加速度和减速度值，必要时增大“Controller setting axis 1->Position controller, Max.Following Error”值。

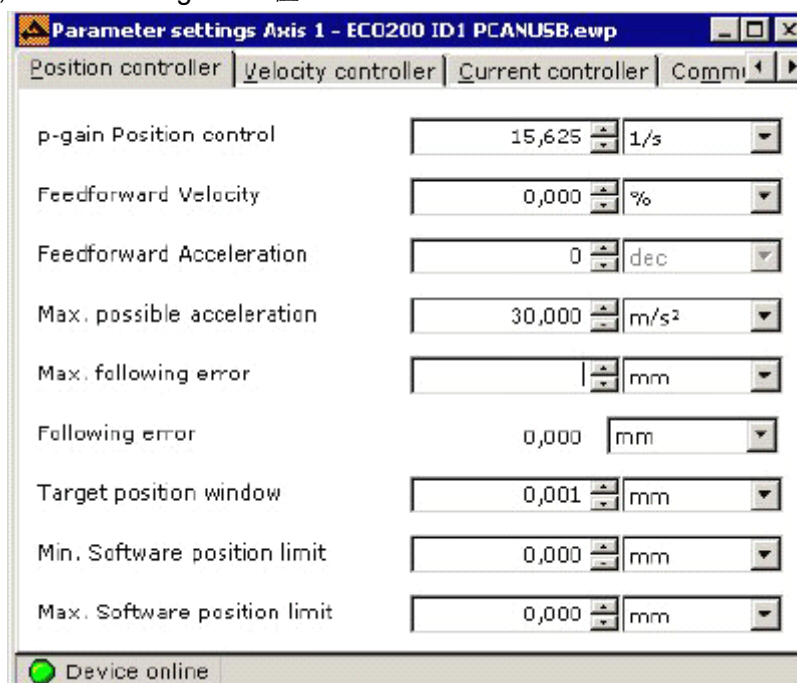


图 7.3 Position Controller

8.2.2 带速度控制的定时正反转

这种模式的定时正反转，需要将伺服设定为-3 的工作模式，如下图：

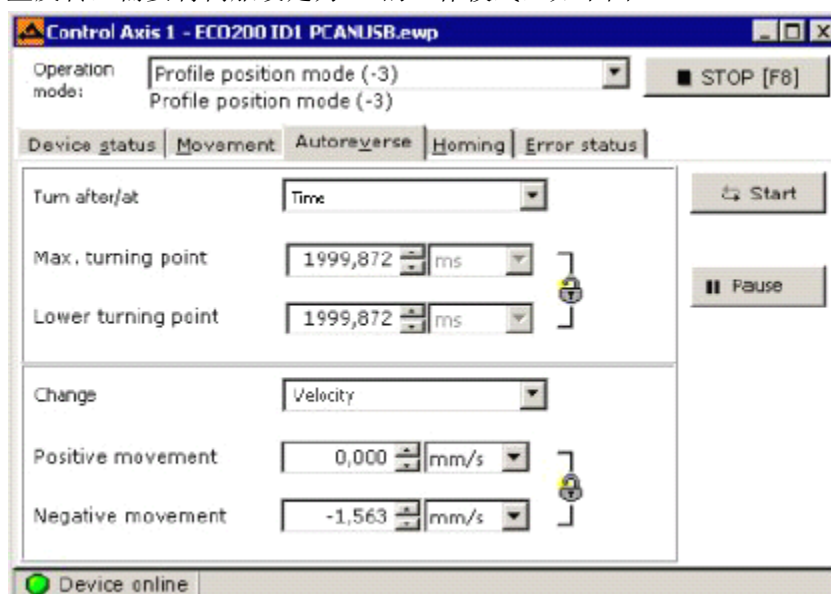


图 7.4 Autoreverse

工作模式-3 属于纯粹的速度控制，没有加速度和减速度。由于该模式没有位置控制所以也就不会有 Following error 故障发生。

电机的最大电流对该模式影响很大。速度曲线性能受“Velocity controller”中下面参数影响：

③ 速度环(“Velocity controller”)中的 I 增益。

③ 故障滤波时间常数

③ 输出滤波时间常数

调整上述参数可以避免电机出现过冲和得到更好的速度性能。

8.2.3 带速度控制的位置正反转

利用速度控制，在指定的位置正反转，工作模式设定为-3。

工作模式-3 属于纯粹的速度控制，没有加速度和减速度。由于该模式没有位置控制所以也就不会有 Following error 故障发生。

电机的最大电流对该模式影响很大。速度曲线性能受“Velocity controller”中下面参数影响：

③ 速度环(“Velocity controller”)中的 I 增益。

③ 故障滤波时间常数

③ 输出滤波时间常数

调整上述参数有利于避免电机过冲和得到更好的速度性能。

8.2.4 带位置控制的速度正反转

达到设定的速度后自动正反转，工作模式设定为 1。为了实现周期的正反转，定义在上面部分的极限速度应该在下面部分设定的位置到达之前达到，因此必须设定合适的加减速。

做为典型的位置控制，在窗口“Movement”中必须设定加速度和减速度值。

另外在窗口“Movement”中的速度值（“Velocity on positioning”）必须大于在窗口“Reversing”中上部分设定的速度值，否则，系统不会到达“Reversing”中设定的速度值，也就不会实现周期性的正反转。

如果“Reversing”中设定的正反向速度值大小不同，就会导致电机朝一个方向漂移，最终停留在一个稳定的末端位置，不再做正反转运动。

注意：如果电流环没有设置保护电流，当电机堵转时因为无过流保护，很容易烧毁电机。

8.3 伺服控制原理

下面我们对 ED 控制器的控制原理作个简介，将有助于用户对控制器的控制特性的了解，使系统达到理想的工作特性。

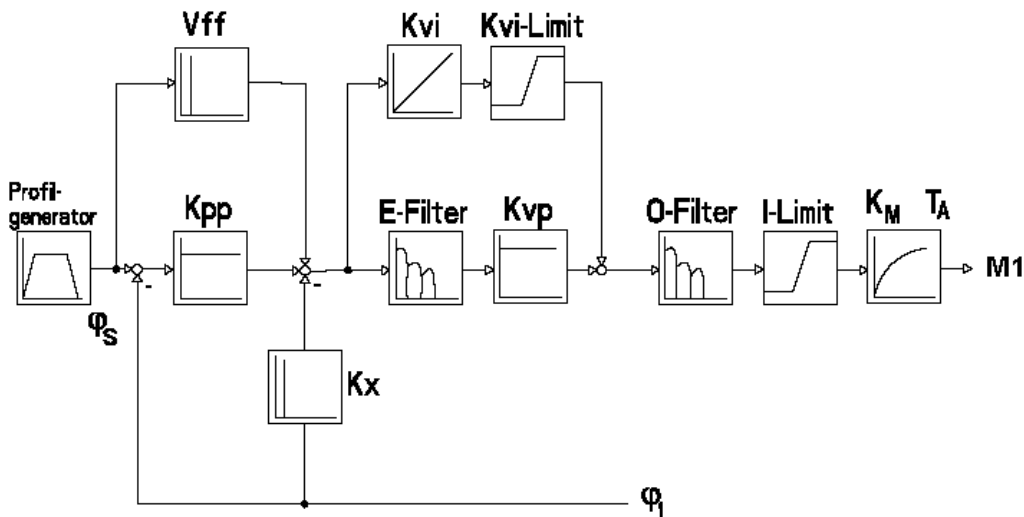


图 8.1 ED 伺服控制器原理

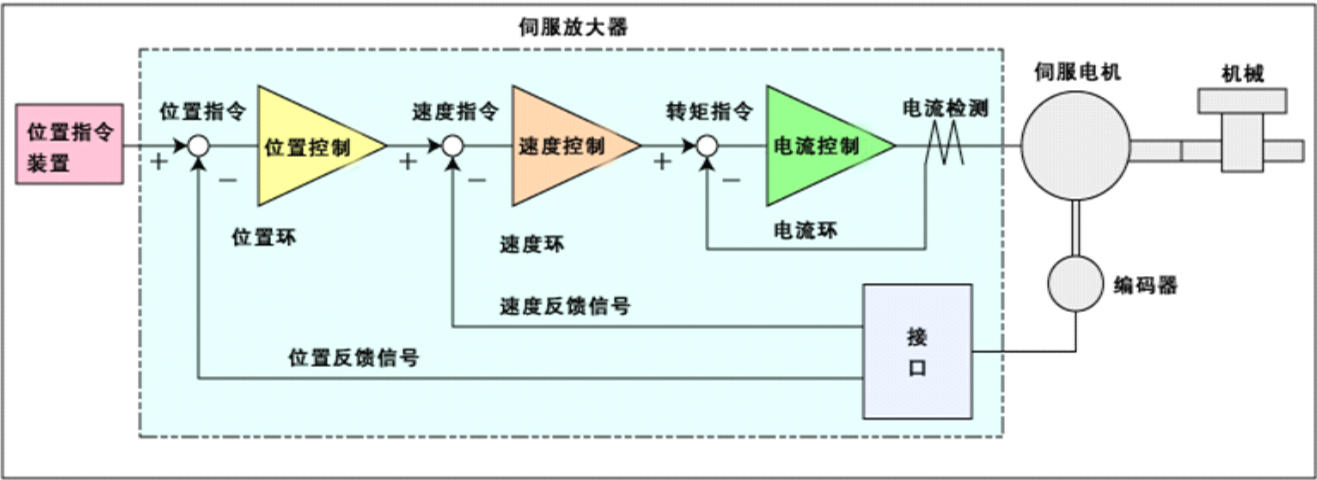


图 8.2 伺服控制器三环关系图

如图所示，和所有伺服系统类似，ED 的控制包括位置环、速度环和电流环三部分，电流环为最里层，速度环次之，位置环则为最外一层。三个控制环关系如下

电流环参数与电机有关(驱动器已经默认了所配电机的最佳参数，不需要调节)。

速度环参数和位置环参数需要根据负载情况进行适当调节。

在进行控制环调节时必须确保速度环带宽大于位置环带宽 2 倍以上，否则可能引起震荡。

8.4 ED 伺服参数设置（ED430/ED620/ED630）

8.4.1 速度环参数

参数	参数说明	功能以及调试
p-gain velocity control	速度环比例增益	调节此值可以改善系统动态响应，尽量把此值设大。一般情况下，负载惯量越大设定值越大，但是设置过大电机会有噪音，可以通过设置 Time constant output filter 来改善，但是控制响应会慢。
i-gain velocity control	速度环积分增益	在低速的时候此值对系统作用效果比较明显，设置值不要超过 10
Limit i-component	积分增益限制	此值与速度环积分增益结合使用，原则上 i-gain velocity control * Limit i-component $i > 1.3 * \text{actual max current}$ 。
Time constant for error filter	偏差滤波时间常数	1 对系统没有滤波，实际速度环增益 = 速度环增益 * 偏差滤波时间常数。一般用在惯量比较大的场合，如果惯量太大需要提高动态响应必须把速度环增益设置大，但是设置大了电机会有噪音，所以就运用此参数来达到同样的动态响应效果。
Time constant output filter	输出滤波时间常数	0 和 1 没有对系统滤波，如果设置此值可以减小一些噪音，但是会降低系统性能，必须适当设置否则系统不稳定
i2-gain velocity control	速度环积分增益（高分辨率用）	在高分辨率编码器上采用此参数，高分辨率设为 25dec 或 0.1（1/256）
Max.velocity	最大速度	速度的最大速度限制

8.4.2 位置环参数

参数	参数说明	功能以及调试
p-gain position control	位置环比例增益	：增大位置环比例增益可以提高位置环带宽，提高位置环带宽可减小定位时间，降低跟随误差，但设定太大会产生噪音甚至震荡，必须根据负载情况合理设置此参数。
Feedforward velocity	位置环速度前馈	增加位置环速度前馈可以减小位置跟随误差。减小位置环速度前馈可以是系统变慢，类似带加减速。

8.4.3 参数调整步骤

第一步：速度环参数调整

1) 根据公式: $B = K_{vp} * K_t * I_{max} * 10000 / (52654735 * J_{tot})$ 计算速度环带宽B

上式中各参数含义:

B: 速度环带宽 单位 (1/s)

K_t : 电机转矩常数 单位 (Nm/Arms) 具体根据电机型号查阅 ED 选型手册

I_{max} : 有效电流 单位 (A) 其等于驱动器峰值电流/1.414.

峰值电流表见下表:

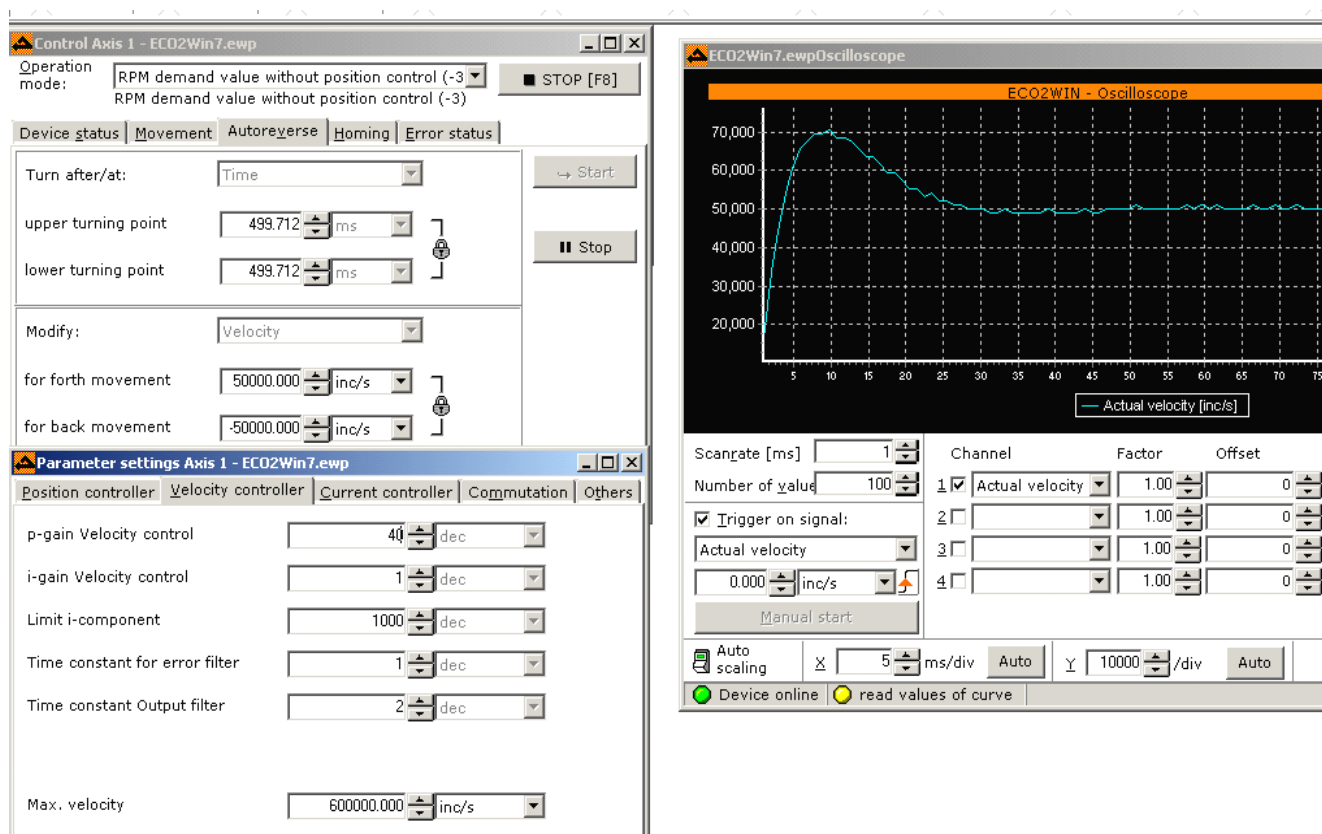
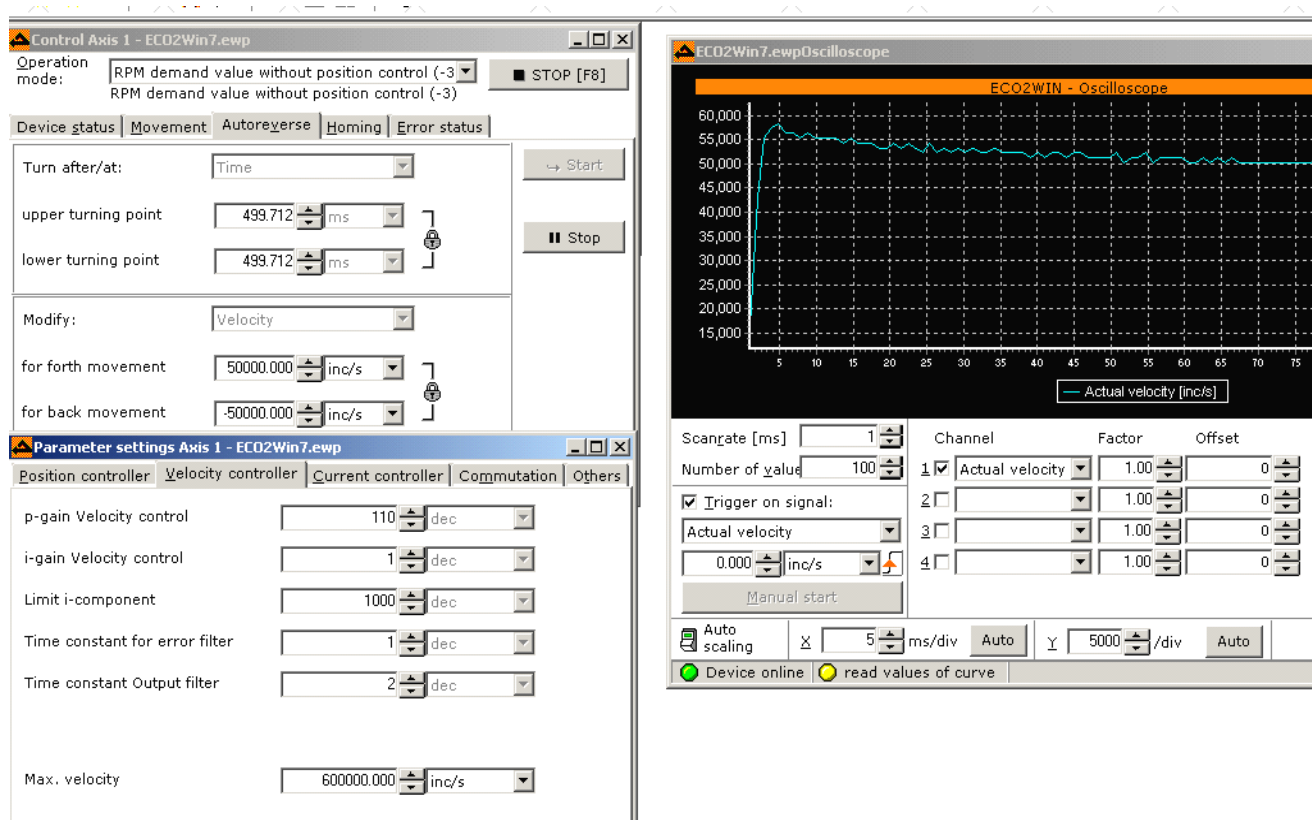
驱动器型号	所配电机型号	峰值电流 A
ED430-0020-XA-K-000	SMH60S-0020-30AAK-3LKX	9
ED430-0040-XA-K-000	SMH60S-0040-30AAK-3LKX	9
ED430-0075-XA-K-000	SMH80S-0075-30AAK-3LKX	13.5
ED430-0100-XA-K-000	SMH80S-0100-30AAK-3LKX	16.5
ED430-0105-XA-K-000	SMH110D-0105-20AAK-4LKC	16.5
ED430-0125-XA-K-000	SMH110D-0125-30AAK-4LKC	16.5
ED430-0126-XA-K-000	SMH110D-0126-20AAK-4LKC	16.5
ED620-0126-XA-K-000	SMH110D-0126-30AAK-4HKC	18
ED620-0157-XA-K-000	SMH110D-0157-30AAK-4HKC	18
ED620-0188-XA-K-000	SMH110D-0188-30AAK-4HKC	18
ED630-0250-XA-G-000	155D-0250-20AAK-3HG	18
ED630-0300-XA-G-000	155D-0300-20AAK-3HG	21
ED630-0400-XA-G-000	155D-0400-20AAK-3HG	27

J_{tot} : 系统惯量 单位 (kgm²) 将电机负载惯量折算到电机轴的惯量 J_l , 再加上电机自身的惯量 J_r , 得到 $J_{tot} = J_r + J_l$ 。

K_{vp} 速度环比例增益, 调整方法如下:

1) 使电机工作在带时间的自动正反转速度模式下 (-3 模式), 按照如图设置示波器参数。

2) 调整 K_{vp} 大小, 观察速度曲线。下图为在不同 K_{vp} 下得到速度曲线。由曲线可以看出 K_{vp} 越大速度响应越快。

图 (1) $K_{vp}=40$ 图 (2) $K_{vp}=110$

2) 根据负载情况调节合适的 K_{vi}

3) 调整滤波参数, 降低系统噪音.

Time constant for error filter: 偏差滤波时间常数, 一般设置为 1

Time constant output filter: 速度环输出滤波时间常数。当 p-gain velocity control 设置较大, 系统产生噪声的时候。可以通过增加该值, 适当消除噪声。但是此值设置越大, 系统响应越慢。

调整完速度环参数后, 根据应用需求对位置环参数进行调整。

第二步: 位置环参数调整

1) 调整位置环比例增益

调整完速度环带宽后, 根据应用需要调整 K_{pp} , 位置环带宽不能超过速度环带宽, 对于一般系统 Pc_Loop_BW (位置环带宽) $< Vc_Loop_BW$ (速度环带宽) / 2, 对 CNC 系统 $Pc_Loop_BW < Vc_Loop_BW / 4$ 为最佳。

2) 调整位置环速度前馈参数

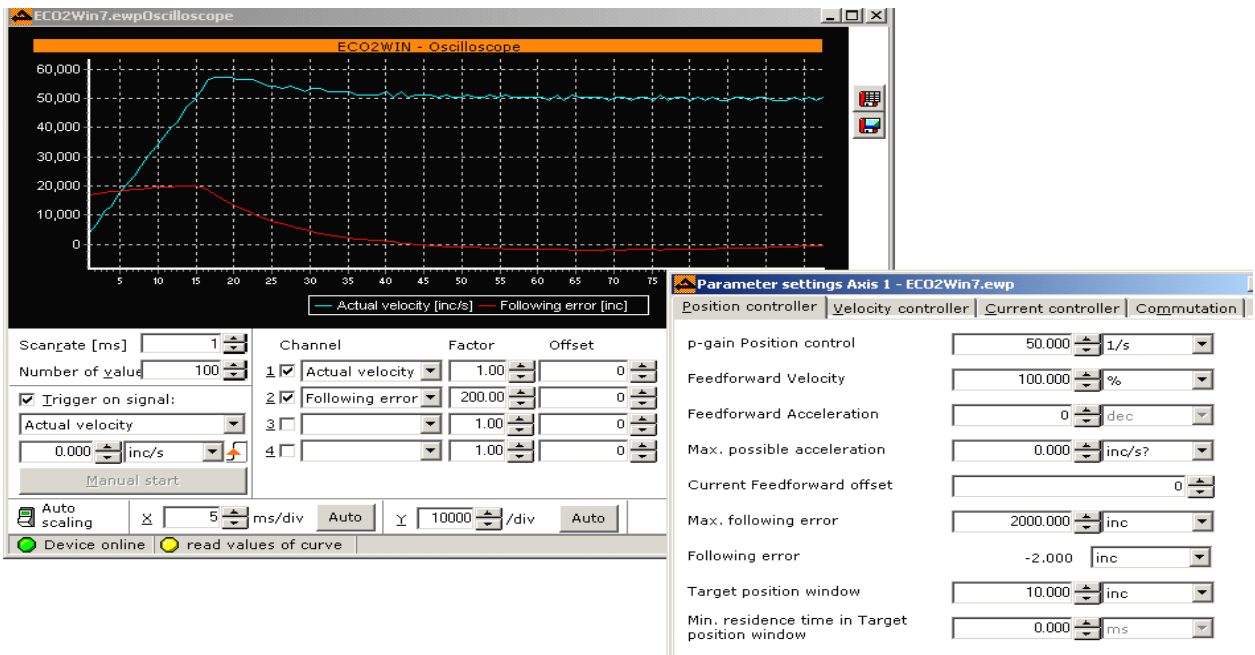
根据机器能够允许的位置误差以及耦合强弱来调节位置环速度前馈参数, 即

Feedforward velocity 一般设置为 100%, 如果系统不需要很高的位置响应, 可以减小此参数以减小过冲现象。

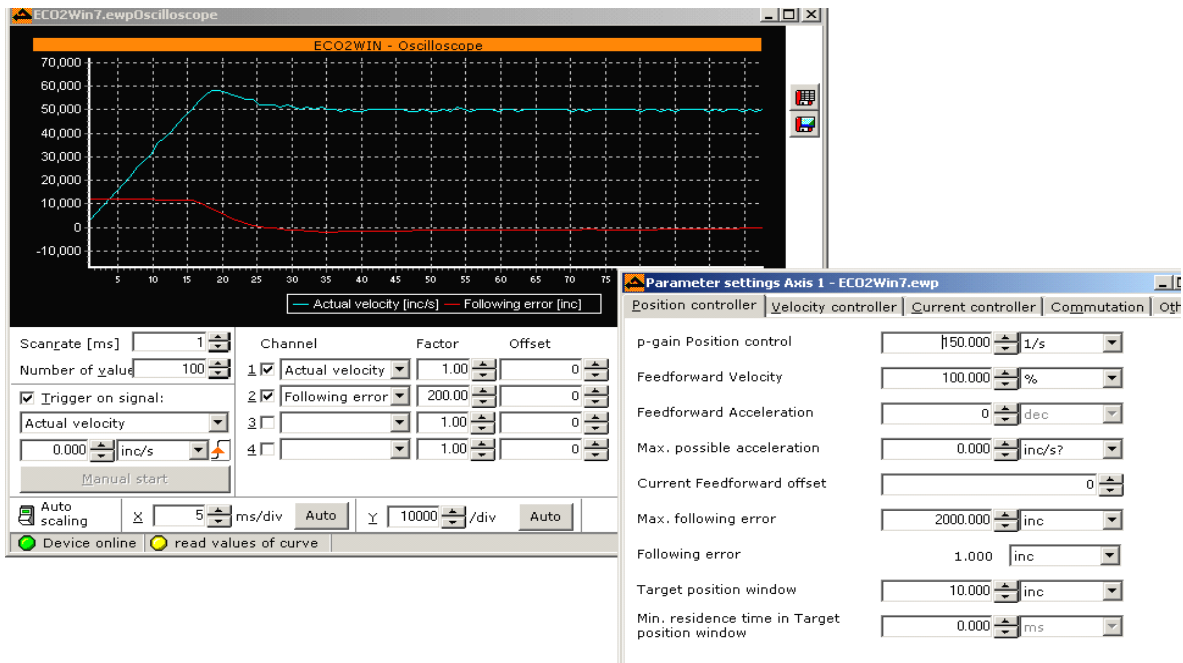
3) 采用示波器观察曲线, 方法如下:

使电机工作在带时间正反转的速度模式下 (工作模式 3), 设置示波器参数如下图所示:

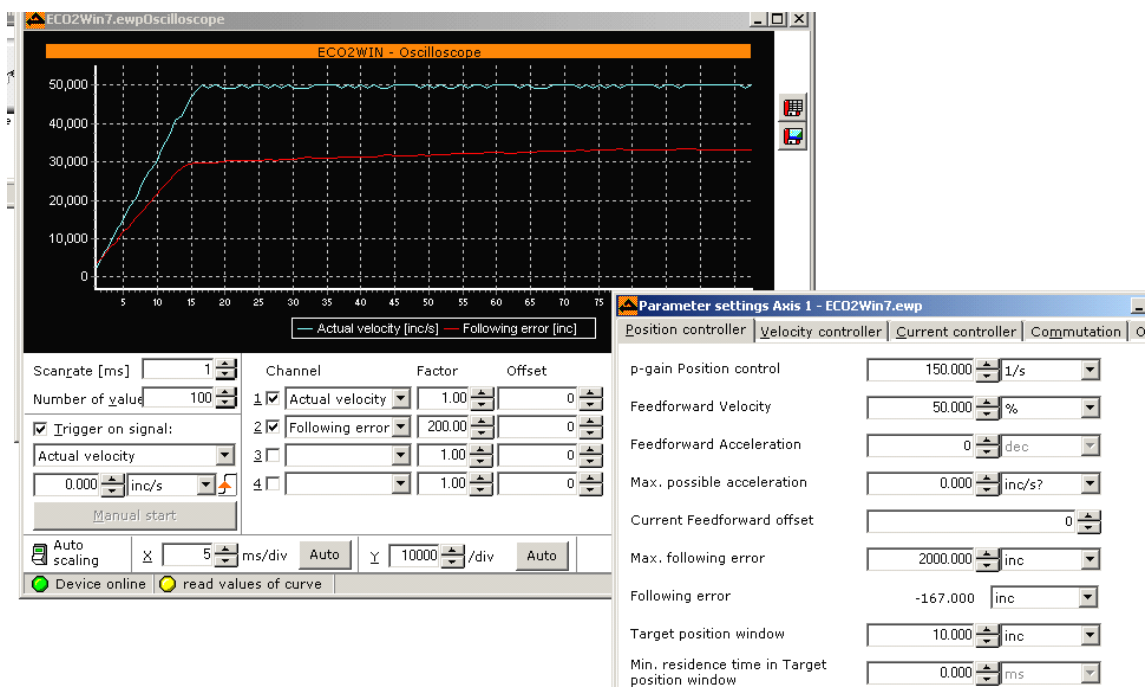
图 (一)、图 (二) 速度环速度前馈 100%, K_{vp} 为 150 时位置环速度响应要比 K_{vp} 为 50 快, 同时跟随误差也小, 但过冲要大。图 (三) 为 K_{vp} 为 150, 速度前馈为 50%, 与图 (一) 相比, 跟随误差要大很多, 同时响应变慢, 但是基本没有过冲。



图（一） $K_v=50, v_{ff}=100\%$



图（二） $K_v=150, v_{ff}=100\%$



图（三） $K_{vp}=150, v_{ff}=50\%$

第九章 通讯协议

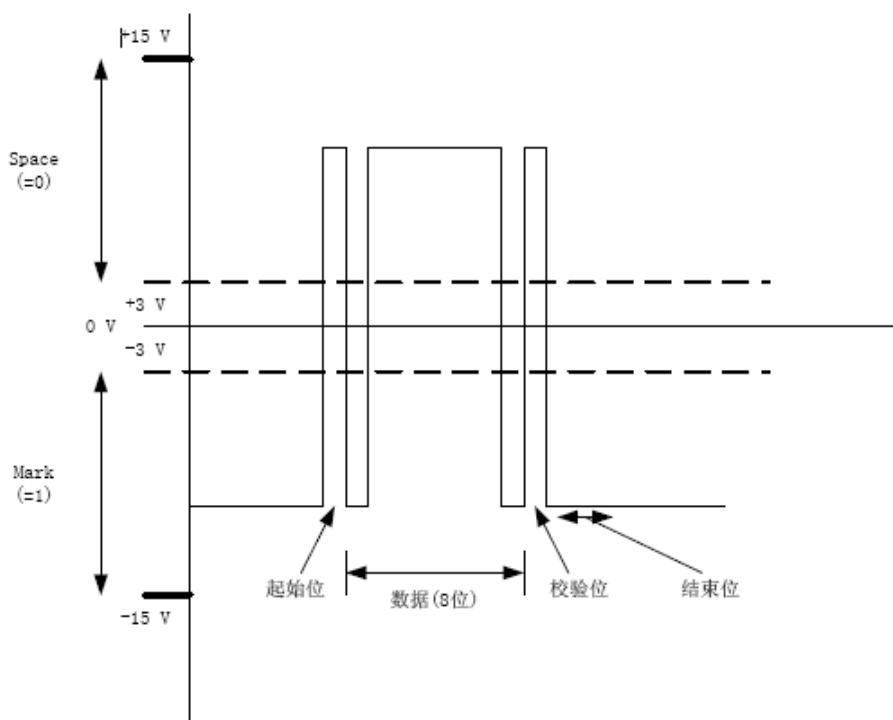
ED 系列总线伺服驱动器具备强大的通讯能力，采用基于对象词典（Object Dictionary）的控制方式，所有控制均归结为对内部对象的配置，而这种配置可以通过 RS232、RS485、CANopen、Profibus 等多种方式实现，从而完成对多站连接，并且可以多通讯口同时进行工作。

ED 系列总线伺服采用三种通讯协议：串口通讯协议、CAN 总线通讯协议和 Profibus 通讯协议。串口通讯协议主要适用于上位机通过 RS232、RS485 的方式与伺服进行通讯，CAN 总线通讯协议则适合于上位机通过 CAN 总线与伺服进行通讯，而 Profibus 通讯协议适合于西门子 S7-300、400 系列 PLC 及其它支持标准 Profibus 通讯协议的上位机与伺服通讯。

串口通讯是 ED 系列伺服驱动器应用比较广泛的一种通讯方式，广泛应用于一些单轴及多轴的系统中，它取代了原有的脉冲方向控制方式，使系统的抗干扰能力、透明性更好，同时为系统减少了运动控制模块，从而降低了系统成本。本章重点介绍串口通讯，总线协议将在下一章节予以介绍。

9.1 硬件介绍

“串行通信”是指设备间使用二根数据信号线(可能还需要控制线)一位一位地进行传输，每一位数据都占据一个固定的时间长度，这种通信方式使用的数据线少，成本低。如下图所示为异步传输模式（Asynchronous Communications Mode）一个字节的资料框架(Frame)由起始位，数据，校验位与结束位所组成。



一般来说，逻辑 1 (MARK) = -3V ~ -15V，逻辑 0 (SPACE) = +3 ~ +15V，每帧数据（7 位或 8 位）都包含一个高电压的起始位，一个低电压的结束位和一个校验位，而数据的传输波特率一般分为 9600Bit/S，19200Bit/S，

38400Bit/S 或 115200Bit/S。

串行数据在传输过程中,由于干扰可能引起信息的出错,例如,传输字符‘E’,其各位为:0100,0101=45H,由于干扰,可能使位变为1,这种情况,我们称为出现了“误码”。我们把如何发现传输中的错误,叫“检错”。发现错误后,如何消除错误,叫“纠错”。最简单的检错方法是“奇偶校验”,即在传送字符的各位之外,再传送1位奇/偶校验位。可采用奇校验或偶校验。

奇校验: 所有传送的数位(含字符的各数位和校验位)中,1的个数为奇数,如:

8位数据 01100101 1的个数为偶数,我们加一个1,变为奇数,所以校验位为1

8位数据 01100001 1的个数为奇数,我们加一个0,仍为奇数,所以校验位为0

偶校验: 所有传送的数位(含字符的各数位和校验位)中,1的个数为偶数,如:

8位数据 01100101 1的个数为偶数,我们加一个0,仍为偶数,所以校验位为0

8位数据 01100001 1的个数为奇数,我们加一个1,变为偶数,所以校验位为1

采用奇偶校验,1位误码能检出,2位及2位以上误码不能检出,同时,它不能纠错。在发现错误后,只能要求重发。但由于其实现简单,仍得到了广泛使用。

双方通讯的前提就是必须首先必须对通讯串口设置相同的数据位,起始位,结束位,波特率和奇偶校验。

ED 默认的通讯设置为:

波特率(默认值) = 9600bps

数据位 = 8

停止位 = 1

无奇偶校验

RS485 的波特率可以修改,波特率对应的对象地址为 2F9100,改变这个值后必须保存且重新上电。这个值的大小可由下面的公式计算出来:

$$N = (614400/\text{波特率}) - 1$$

9600bps N=63 位计时错误 0%;

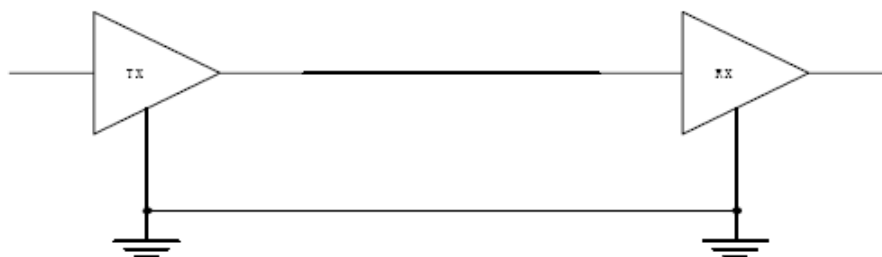
19200bps N=31 位计时错误 0%;

38400bps N=15 位计时错误 0%;

更高的波特率(>38400bps)会产生计时错误,57600bps N=11 位计时错误 3.125%,建议用户只采用 38400 及以下波特率进行通讯。

9.1.1 RS232 接口

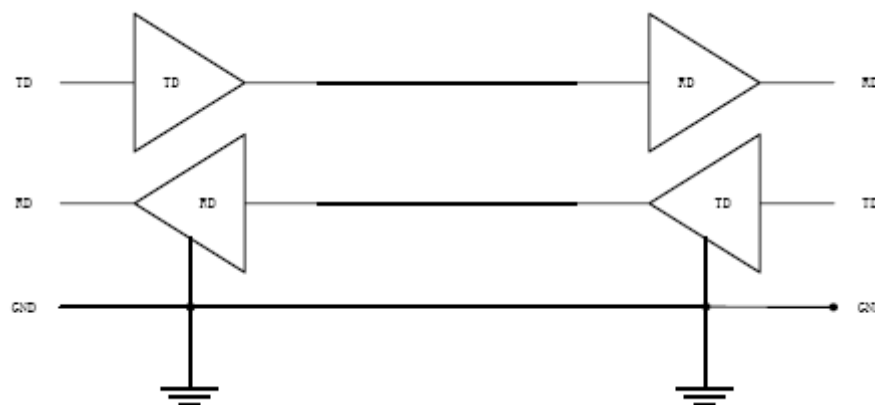
目前 RS-232 是 PC 机与通信工业中应用最广泛的一种串行接口。RS-232 被定义为一种在低速率串行通讯中增加通讯距离的单端标准。RS-232 以非平衡数据传输(Unbalanced Data Transmission)的接口方式,这种方式是以一根信号线相对于接地信号线的电压来表示一个逻辑状态 Mark 或 Space,下图为一个典型的连接方式:



典型的 RS-232 的接脚图(PC Compatible)如下

接脚号	接脚名	说明
1	Carrier Detect (CD)	
2	Received Data (RD)	数据接收信号线
3	TransmittedData (TD)	数据传送信号线
4	Data Terminal Ready (DTR)	
5	Signal Ground	接地信号线
6	Data Set Ready (DSR)	
7	Request To Send (RTS)	
8	Clear To Send(CTS)	
9	Ring Indicator (RI)	

如果不考虑CTS及RTS则ED系列伺服与PLC、HMI的典型连接方式如下：



由于 RS-232 是全双工传输模式，具有各自独立的传送 (TD) 及接收 (RD) 信号线与一根接地信号线，而 CTS 及 RTS 的目的是作为 ED 伺服与 PLC、HMI 的信号交换 (Handshaking) 控制用，只有少数 PLC 的 CTS 或 RTS 必需由 ED 伺服控制。

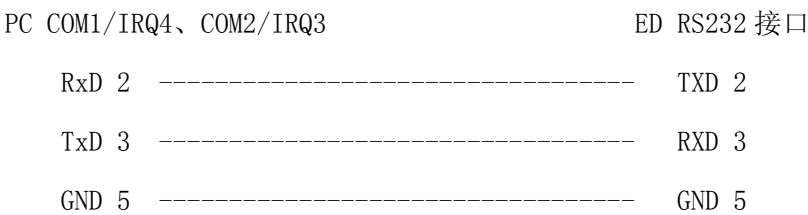
RS-232 连接线的长度是最被讨论的话题，标准规范是不可超过 50 英尺 (1 英尺=30 公分) 或者是电容值不可超过 2500pF。如果以电容值为标准，UTP CAT-5 连接线典型电容值的 17 pf/英尺，则允许的连接线长度为 147 英尺约 44 米。如果是有遮蔽的连接线，则它的容许长度会更长，在有干扰的环境连接线的允许长度会减少，在实际使用过程中，建议通讯线不要超过 10M。

由于 RS-232-C 接口标准出现较早，难免有不足之处，主要有以下四点：

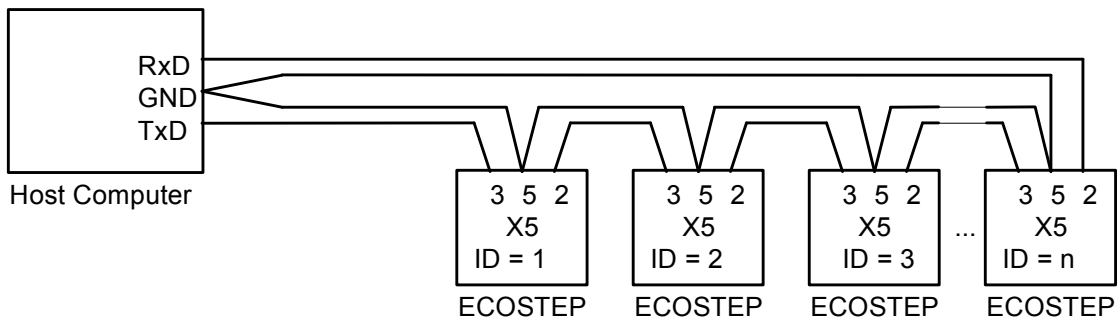
- (1) 接口的信号电平值较高，易损坏接口电路的芯片。
- (2) 传输速率较低，在异步传输时，波特率为 20Kbps。
- (3) 接口使用一根信号线和一根信号返回线而构成共地的传输形式， 这种共地传输容易产生共模干扰，所以抗噪声干扰性弱，当波特率越高其抗干扰的能力会呈倍数下降。
- (4) 所以其传输距离有限。

接线介绍

ED 伺服与上位机、HMI、PC 机通讯时，单台伺服的接线方式如下：



ED 伺服与上位机、HMI、PC 机通讯时，多台伺服的接线方式如下：

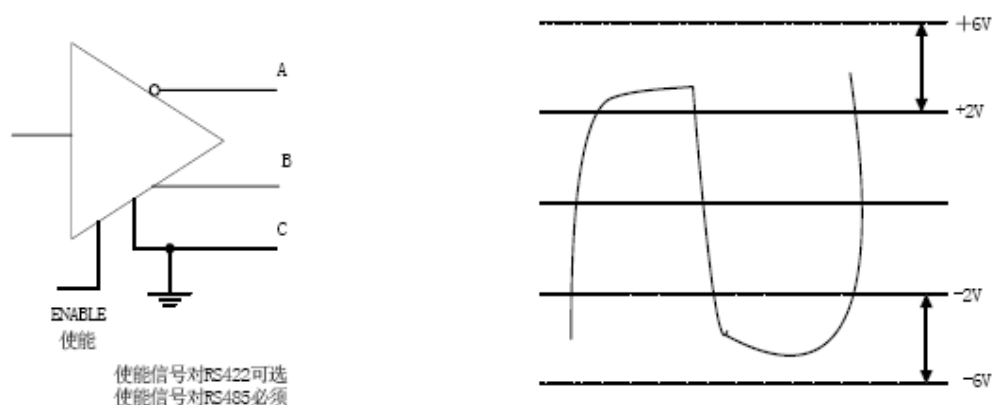


ID 号（地址号）通过 DIP 开关 S0/S1/S2/S3 的组合来选择。每一个 ED 的 ID 必须设置为唯一，地址号 0 为系统保留，不能设置。

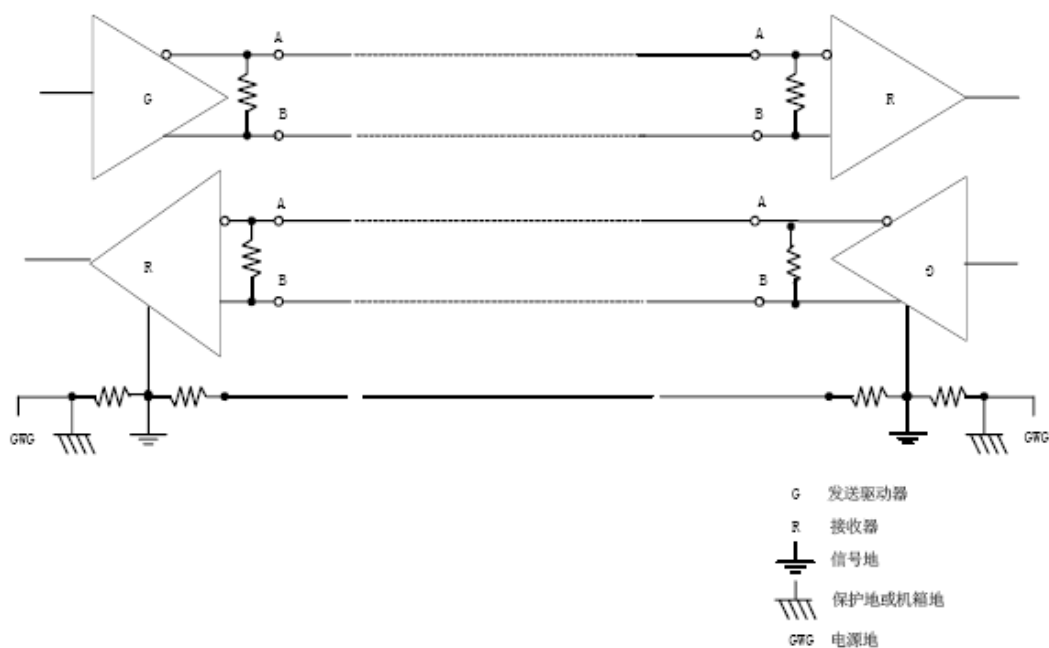
9.1.2 RS485 接口

RS422 接口

RS-422、RS-485 与 RS-232 不一样，数据信号采用差分传输方式，也称作平衡传输，它使用一对双绞线，将其中一线定义为 A，另一线定义为 B，通常情况下，发送驱动器 A、B 之间的正电平在+2~+6V，是一个逻辑状态，负电平在-2~6V，是另一个逻辑状态，另有一个信号地 C。在 RS-485 中还有一“使能”端，而在 RS-422 中这是可用可不用的。“使能”端是用于控制发送驱动器与传输线的切断与连接。当“使能”端起作用时，发送驱动器处于高阻状态，称作“第三态”，即它是有别于逻辑“1”与“0”的第三态。如下图所示：



RS-422 标准全称是“平衡电压数字接口电路的电气特性”，它定义了接口电路的特性。下图是典型的 RS-422 四线接口。实际上还有一根信号地线，共 5 根线。由于接收器采用高输入阻抗和发送驱动器比 RS232 更强的驱动能力，故允许在相同传输线上连接多个接收节点，最多可接 10 个节点。即一个主设备（Master），其余为从设备（Slave），从设备之间不能通信，所以 RS-422 支持点对多的双向通信。RS-422 四线接口由于采用单独的发送和接收通道，因此不必控制数据方向，各装置之间任何必须的信号交换均可以按软件方式（XON/XOFF 握手）或硬件方式（一对单独的双绞线）实现。



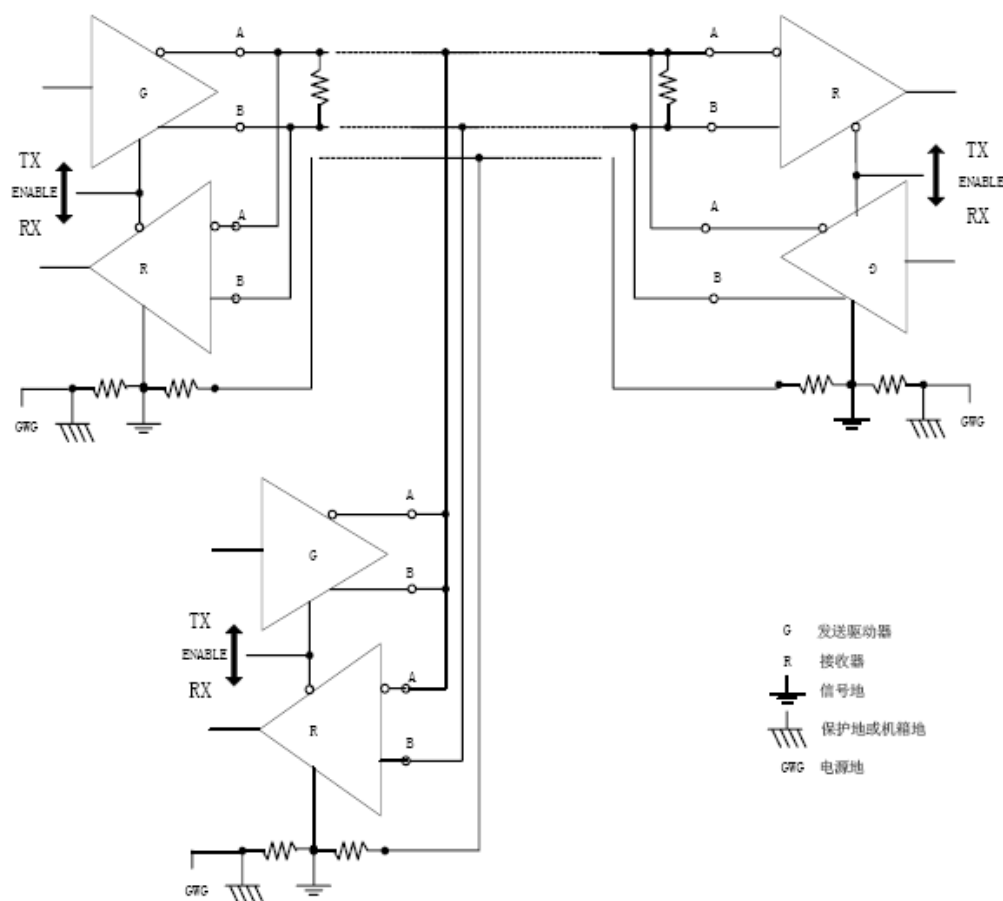
RS-422 的最大传输距离为 4000 英尺（约 1219 米），最大传输速率为 10Mb/s。其平衡双绞线的长度与传输速率成反比，在 100kb/s 速率以下，才可能达到最大传输距离。只有在很短的距离下才能获得最高速率传输。一般 100 米长的双绞线上所能获得的最大传输速率仅为 1Mb/s。RS-422 需要一终接电阻，要求其阻值约等于传输电缆的特性阻抗。在短距离传输时可不接终接电阻，即一般在 300 米以下不需接终接电阻。终接电阻接在传输电缆的最远端。

ED 系列伺服的接脚图如下：

接脚号	接脚名	说明
7	RX-	数据接收信号线A
2	RX+	数据接收信号线B
8	TX-	数据传输信号线A
3	TX+	数据传输信号线B
5	GND	接地信号线

RS485 接口

由于 RS-485 是从 RS-422 基础上发展而来的, 所以 RS-485 许多电气规定与 RS-422 相仿。如都采用平衡传输方式、都需要在传输线上接终接电阻等。RS-485 可以采用二线与四线方式, 二线制可实现真正的多点双向通信, 其中的使能信号控制数据的发送或接收, 如下图所示:



RS-485 的电气特性: 逻辑“1”以两线间的电压差为 $+(2\sim6)V$ 表示; 逻辑“0”以两线间的电压差为 $-(2\sim6)V$ 表示, RS-485 的数据最高传输速率为 10Mbps, RS-485 接口是采用平衡驱动器和差分接收器的组合, 抗共模干扰能力增强, 即抗噪声干扰性好。它的最大传输距离标准值为 4000 英尺, 实际上可达 3000 米, 另外 RS-232 接口在总线上只允许连接 1 个收发器, 即单站能力。而 RS-485 接口在总线上是允许连接多达 128 个收发器。即具有多站能力, 这样用户可以利用单一的 RS-485 接口方便地建立起设备网络。因 RS-485 接口具有良好的抗噪声干扰性, 长的传输距离和多站能力等上述优点就使其成为首选的串行接口。因为 RS485 接口组成的半双工

网络，一般只需二根连线，所以 RS485 接口均采用屏蔽双绞线传输。

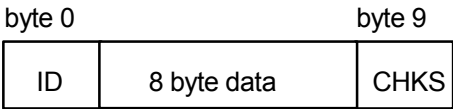
9.2 协议介绍

ED 系列伺服通讯协议为主从站协议，上位 PLC、PC、HMI 都能够与 ED 系列伺服进行通讯。ED 系列伺服是一款欧系风格的产品，支持 CAN、Profibus 现场总线通讯方式，所以 ED 系列伺服的串口通讯协议也是根据 CAN 现场总线的通讯协议演变过来的。由于 ED 系列伺服是标准的 CAN 从站设备，所有的参数、数值和功能都是通过 index 和 subindex 表示，那么上位机访问 ED 系列伺服也是访问伺服内部的这些 index 和 subindex。这些参数既可以通过 RS232C、RS485 来设定，也可以通过 CAN 来设定。

如果要与上位机通讯，要确保上位机和 ED 系列伺服的通讯参数一致：

9600 波特率（可更改）、8 个数据位、1 个停止位、无奇偶校验

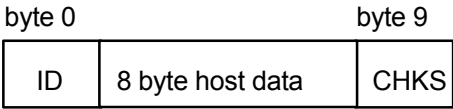
这个传输协议使用的是固定的十个字节的格式。



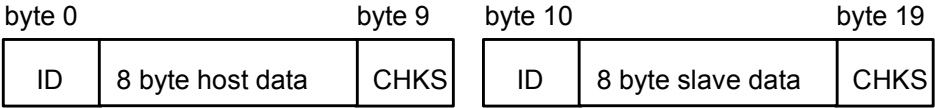
ID 号从站的地址号

CHKS = -SUM(byte0, ..., byte8), CHKS 为前面 9 个字节的和取负的计算结果的最后 2 位。

上位机传送：



上位机接收（ED 伺服返回）：



注意：每十个字节中都有一个属于自己的 CHKS。

如果上位机送一个包含了网络中没有的地址的数据给 ED, 那么就不会有从站做出响应，此时，上位机将只会收到 10 个字节的数据报文。主机正确地发送数据后，从站寻找符合自身地址的数据报文，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。

协议类型

数据协议不同于传输协议，其内容是上面的串口传输协议 10 个字节中间的 8 个字节的内容。访问不同的 Index 和 Subindex 是只需改变中间 8 个自己的数据即可，具体方式请参看下面的介绍。

下载

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。

主站传送：

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
CMD	INDEX	SUB INDEX			DATA		

CMD 指定数据传输的方向和数据的大小

23(0x16) 发送四个字节的数据(bytes 4...7 包含 32 位)

2b(0x16) 发送二个字节的数据(bytes 4, 5 包含 16 位)

2f(0x16)	发送一个字节的数据 (bytes 4 包含 8 位)
----------	----------------------------

INDEX	发送对象的地址
0	00000000
1	00000001
2	00000010
3	00000011
4	00000100
5	00000101
6	00000110
7	00000111
8	00001000
9	00001001
10	00001010
11	00001011
12	00001100
13	00001101
14	00001110
15	00001111
16	00010000
17	00010001
18	00010010
19	00010011
20	00010100
21	00010101
22	00010110
23	00010111
24	00011000
25	00011001
26	00011010
27	00011011
28	00011100
29	00011101
30	00011110
31	00011111
32	00100000
33	00100001
34	00100010
35	00100011
36	00100100
37	00100101
38	00100110
39	00100111
40	00101000
41	00101001
42	00101010
43	00101011
44	00101100
45	00101101
46	00101110
47	00101111
48	00110000
49	00110001
50	00110010
51	00110011
52	00110100
53	00110101
54	00110110
55	00110111
56	00111000
57	00111001
58	00111010
59	00111011
60	00111100
61	00111101
62	00111110
63	00111111
64	01000000
65	01000001
66	01000010
67	01000011
68	01000100
69	01000101
70	01000110
71	01000111
72	01001000
73	01001001
74	01001010
75	01001011
76	01001100
77	01001101
78	01001110
79	01001111
80	01010000
81	01010001
82	01010010
83	01010011
84	01010100
85	01010101
86	01010110
87	01010111
88	01011000
89	01011001
90	01011010
91	01011011
92	01011100
93	01011101
94	01011110
95	01011111
96	01100000
97	01100001
98	01100010
99	01100011
100	01100100
101	01100101
102	01100110
103	01100111
104	01101000
105	01101001
106	01101010
107	01101011
108	01101100
109	01101101
110	01101110
111	01101111
112	01110000
113	01110001
114	01110010
115	01110011
116	01110100
117	01110101
118	01110110
119	01110111
120	01111000
121	01111001
122	01111010
123	01111011
124	01111100
125	01111101
126	01111110
127	01111111
128	10000000
129	10000001
130	10000010
131	10000011
132	10000100
133	10000101
134	10000110
135	10000111
136	10001000
137	10001001
138	10001010

SUB INDEX 发送对象的子地址

DATA 内 4 个字节的顺序是低为在前，高位在后。例如要向从站内“目标速度”写入 600rpm，那么 $600\text{rpm} = 10\text{r/s} = 80000\text{inc/s} = 80000 * 64 = 5120000 \text{ inc}/64\text{s} = 4\text{E}2000$ ，60FF0020 内部单位 $\text{inc}/64\text{s}$ ，5120000 为 10 进制，4E2000 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 4E 20 00 只有 3 个字节，那么在高位补零，所以最终结果 = 00 4E 20 00

DATA: byte4=00

byte5=20

byte6=4E

byte7=00

从站响应:

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
RES	INDEX	SUB INDEX			RESERVED		

RES: 显示从站的响应:

60(0x16) 数据成功传送

80(0x16) 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

RES 备用

上传

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。

主站传送:

CMD	INDEX	SUB INDEX	RESERVED
-----	-------	-----------	----------

CMD	指定数据传输的方向
-----	-----------

40(0x16) 固定格式

INDEX 16位的地址

SUBINDEX 8位的子地址
RESERVED 字节4...7不用

从站响应:

byte 0	byte 1	byte 2	byte 3	byte 4	byte 5	byte 6	byte 7
RES	INDEX		SUB INDEX		DATA		

RES 显示从站的响应:

43 (0x16) 字节4...7包含32位数据

4B (0x16) 字节4, 5包含16位数据

4F (0x16) 字节4包含8位数据

80 (0x16) 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 ff 60 00 00 00 00 00 60 (该命令读取从站的目标速度60FF0020)

从站响应:

01 43 FF 60 00 00 20 4E 00 EF

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...byte7 保存。

byte4=00, byte5=20, byte6=4E, byte7=00

那么DATA= byte7 byte6 byte5 byte4 = 00 4E 20 00 = 5120000 = 80000 * 64

目标速度60FF0020的内部单位是inc/64s, 换算后DATA=80000 inc/s = 600rpm。

第十章 ED 伺服现场总线接口

10.1 现场总线简介

现场总线（Fieldbus）技术从提出到现在有二十多年了，作为工业数据总线，它主要解决工业现场智能化仪器仪表、控制器、执行机构等现场设备间的数字通信以及这些现场控制设备和高级控制系统之间的信息传递问题，通过模拟变数字实现了不同公司产品间的互操作性问题，使用户有了更大的选择权，尤其解决了流行几十年的传统系统过于封闭、难以维护的缺点。

采用现场总线控制技术，可大大简化系统集成的工作量、为控制系统的安装调试节省大量的费用，而系统的可靠性、稳定性却得到大幅提高，配合现场总线技术的各类总线诊断技术进一步提高了整个系统的性能。强大的通讯功能又使得系统更加开放透明。

现场总线是一种革命性的通讯控制技术，因其具有很多普通控制方式不具有的优点，所以才得到了迅速的推广应用，与老的控制方式比较起来它主要的优势如下：

- ⑤ 节约布线成本，减少布线时间，减小出错机率（对于大型设备尤为突出，如果当驱动器、变频器、传感器等放置到现场的话，可以节省大量的电缆费用）；
- ⑤ 减小施工难度，缩短施工周期；
- ⑤ 降低系统总成本（从安装、系统维护、升级方面大幅降低系统成本）；
- ⑤ 可靠性更高，抗干扰能力更强（比传统 485 通讯方式更为可靠，更不易受干扰）；
- ⑤ 走线少、全数字信息交互（模拟量通常易受干扰）；
- ⑤ 信息量更大（节点数据信息、状态信息、异常信息等均可方便提供）；
- ⑤ 实时性更高（比传递 485 通讯速度大大提高，是 485 通讯速度的 100 倍左右，且避免了 485 通讯方式的多控制器之间交换方式，直接由一个 PLC 来协调处理，实时性大为提高）；
- ⑤ 可维护性更强（可以很方便检测出系统故障所在，且几乎所有的 CAN 从站都具有故障诊断能力，便于排查及处理）；
- ⑤ 开发性更加（目前全球范围内生产总线产品设备的厂家达上千家，客户可以任意选择适合字节的设备）。

10.2 CAN 总线通讯

CAN 现场总线技术是集自动控制技术、通讯技术、传感技术、计算机技术、诊断技术、微电子技术、网络技术等于一体，是个革命性的技术，正被广泛应用于自动化各个领域。目前广泛使用的其它现场总线还有 Profibus、DeviceNet、ControlNet、HART、FF 等等，但是 CAN 总线是所有现场总线中最早出现的，也是最适合于机器自动化领域的现场总线，如今它已经广泛应用于汽车、飞机、轮船、印刷、纺织、电子等等加工领域，

是目前应用领域最为广泛的现场总线。

CAN 总线系统解决方案即是利用 CAN 总线的优点及其特长为机器自动化设备提供高效、可靠、性价比高的解决方案。作为机器自动化领域总线解决方案倡导者，CAN 总线系统解决方案更能满足您对性价比的要求。

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。

ED 系列伺服是标准的 CAN 从站设备，严格遵循 CANopen 2.0A/B 协议，任何支持该协议的上位机均可以与其进行通讯。ED 伺服内部使用了一种严格定义的对象列表，我们把它称作对象辞典，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）类似我们常说的内存地址，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
2509	00	8 (=0x08)	R		主从电子齿轮比参数设置
2509	02	32 (=0x20)	W		映射的从轴速度
2509	03	16 (=0x10)	MW		电子齿轮比分子
2509	04	16 (=0x10)	MW		电子齿轮比分母
2509	05	8 (=0x08)	W		齿轮模式

对象的属性有下面几种：

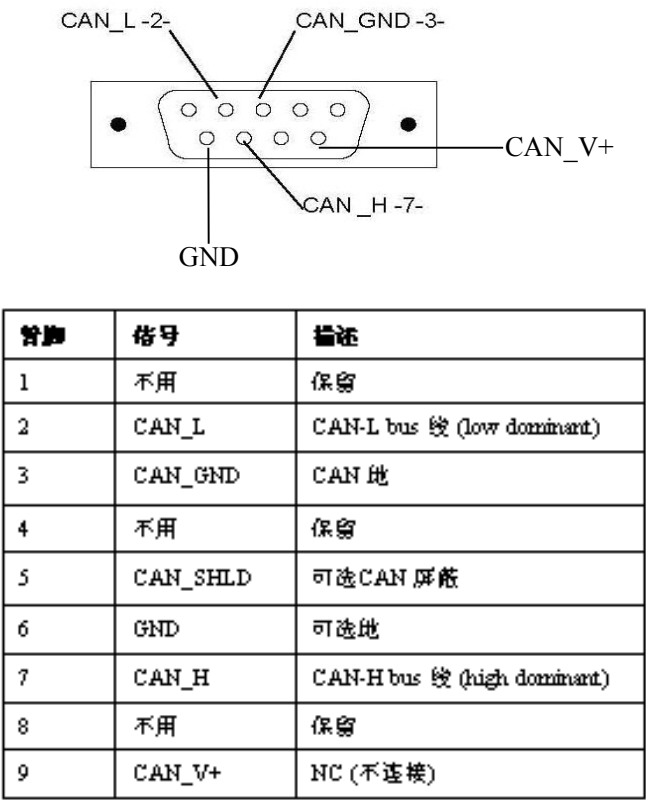
1. RW(读写)：对象可以被读也可以被写入；
2. RO(只读)：对象只能被读；
3. WO（只写）：只能写入；
4. M（可映射）：对象可映射，类似间接寻址；
5. S（可存储）：对象可存储在 Flash—ROM 区，掉电不丢失。

10.2.1 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模型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，最常用的就是双绞线信号，使用差分电压传送（常用总线收发器），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，用 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。

标准 CAN 从站 CAN 通讯接口图：



■注意：

- 1、所有从站的 2、7 脚直接相接即可，采用串连的方式接线, 不能采用星型连接方式；
- 2、PLC 端和最后一个负载端需要借 120 欧姆的终端电阻；
- 3、对于我们 ED430、ED620、ED630、CANopen I/O 设备是 6、9 脚不需要外部 24V 电源供电的，但 ED100 是需要外接 24V 电源的；
- 4、通讯电缆请采用屏蔽线，并做好接地处理（短距离通讯时 3 脚地线可以不接，但是长距离、高波特率通讯时建议把 3 脚接地）；
- 5、各种波特率所能够通讯的最长距离：

通讯速度	通讯距离
1Mbit/s	25M
800Kbit/s	50M
500Kbit/s	100M
250Kbit/s	250M
125Kbit/s	500M
50Kbit/s	600M
25Kbit/s	800M
10Kbit/s	1000M

10.2.2 软件介绍

10.2.2.1 EDS

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 的属性等等参数,类似于 Profibus 的 GSD 文件。因此在进行硬件配置前,我们首先需要把从站的 EDS 文件导入到上位组态软件中

10.2.2.2 SDO

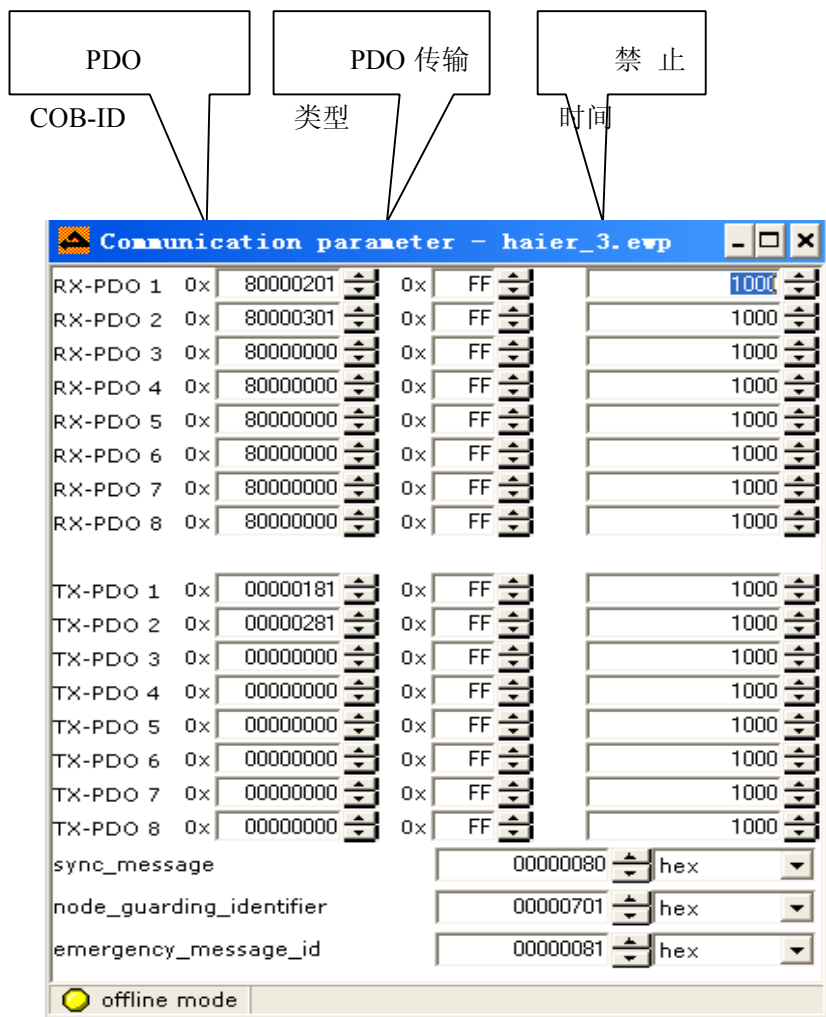
SDO 主要用来在设备之间传输低优先级的对象,典型是用来对从设备进行配置、管理,比如用来修改电流环、速度环、位置环的 PID 参数, PDO 配置参数等,这种数据传输跟 MODBUS 的方式一样,即主站发出后,需要从站返回数据响应。这种通讯方式只适合对参数的设置,不适合于对实时性要求较高的数据传输。通过 HMI、PLC、PC 与伺服通讯均采用的是 SDO 的方式。

SDO 的通讯方式分为上传和下载,上位机可以根据专用的 SDO 读写指令来读写伺服内部的 OD 即可。

10.2.2.3 PDO

PDO 一次性可传送 8 个字节的数据,没有其它协议预设(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 的传输方式打破了现有的数据问答式传输理念,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。

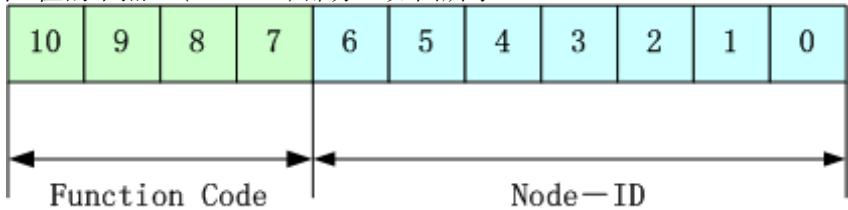
ED 伺服 PDO 设置画面:



PDO COB-ID

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 为 PDO 定义了相应的传输级别，有了这些传输级别后，控制器和伺服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，这样控制器和伺服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：



Node-ID由系统集成商定义，例如通过设备上的拨码开关设置(即伺服的站号)。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。

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
CANopen主/从连接集的对等对象			
对象	功能码 功能码 (ID-bits 10-7)	COB-ID	通讯参数在 OD中的索引
紧急	0001	081H-0FFH	1024H, 1015H
PD01(发送)	0011	181H-1FFH	1800H
PD01(接收)	0100	201H-27FH	1400H
PD02(发送)	0101	281H-2FFH	1801H
PD02(接收)	0110	301H-37FH	1401H
PD03(发送)	0111	381H-3FFH	1802H
PD03(接收)	1000	401H-47FH	1402H
PD04(发送)	1001	481H-4FFH	1803H
PD04(接收)	1010	501H-57FH	1403H
SD0(发送/服务 器)	1011	581H-5FFH	1200H
SD0(接收/客 户)	1100	601H-67FH	1200H
NMT Error Control	1110	701H-77FH	1016H-1017 H

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.

ED 伺服所支持的 COB-ID:

► 发送PDO (TXPDO)

发送PDO相对于伺服来说就是指伺服发送出去的数据, 这些数据由PLC来接收。发送PDO的功能码 (COB-ID)

为:

- 1、 0x180+伺服站号
- 2、 0x280+伺服站号
- 3、 0x380+伺服站号
- 4、 0x480+伺服站号

► 接收PDO (RXPDO)

接收PDO相对于伺服来说就是指伺服接收的数据, 这些数据由PLC来发送, 发送PDO的功能码 (COB-ID) 为:

- 1、 0x200+伺服站号
- 2、 0x300+伺服站号
- 3、 0x400+伺服站号
- 4、 0x500+伺服站号

由于ED系列伺服是按标准的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, 只要控制器和伺服都按照这个来定义即可。

PDO 传输类型

PDO 有两种传输方式:

■ 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), 伺服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。该方式下伺服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。该方式下伺服驱动器每接收到 n 个同步报文后, PDO 里的数据发送一次

■ 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

优先级越高)。

✦ 对于 ED 系列伺服驱动器，它支持所有 256 种传输方式，用户只需要根据控制器所支持的传输方式来选择 ED 伺服的传输方式即可。

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 100us

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，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！

✚ 报文格式：

■ (0x700+节点号) + 状态

✚ 状态：

■ 0：启动，4：停止，5：运行，127：预操作

■ 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！

✚ 主站请求报文格式：

■ (0x700+节点号) (该报文无数据)

✚ 从站响应报文格式：

■ (0x700+节点号) + 状态：

✚ 状态：

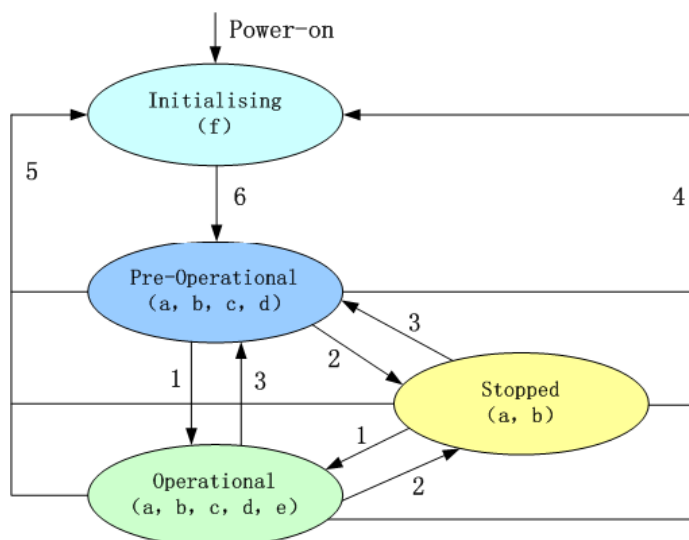
■ 数据部分包括一个触发位 (bit7)，触发位必须在每次节点保护应答中交替置“0”或者“1”。触发位在第一次节点保护请求时置为“0”。位 0 到位 6 (bits0~6) 表示节点状态

■ 状态：0：初始化，1：未连接，2：连接，3：操作，4：停止，5：运行，127：预操作

标准的 CAN 从站一般都只支持一种节点保护方式，ED 伺服支持的保护方式是：**节点保护**。

启动过程

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。可以用节点状态转换图表示这种初始化过程，如图所示。



注意:

►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:

a. 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), NMT命令字 (在括号中):

1: Start_Remote_node (0x01)

2: Stop_Remote_Node (0x02)

3: Enter_Pre-Operational_State (0x80)

4: Reset_Node (0x81)

5: Reset_Communication (0x82)

6: 设备初始化结束, 自动进入Pre_Operational状态, 发送Boot-up消息

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:

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:

命令字	NMT 服务
1	Start Remote Node
2	Stop Remote Node
128	Enter Pre-operational State
129	Reset Node
130	Reset Communication

例如, 如果要想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：

0x000:0x80 0x02

10.2.3 CAN 通讯案例

详细案例请看附录。

10.3 Profibus 总线通讯

PROFIBUS 是国际化的、开放的、不依赖于设备制造商的现场总线标准，广泛地应用于许许多多制造业自动化、流程自动化、楼宇、交通、电力等领域。满足欧洲开放式现场总线标准(EN50170)，由 Profibus-FMS(Fieldbus Message Specification)、Profibus-PA(Process Automation)和 Profibus-DP(Distributive Peripheral)等三部分组成。

Profibus-DP 最适合于高速数据传输(速率 9.6Kbit/s~12Mbit/s)、可实现非常经济的设备级网络，适用于现场控制器与分散智能 I/O 之间的通信，可满足交直流调速系统快速响应的要求。

Profibus-DP 以主从通讯(主站与从站间通讯)结合令牌通讯(主站之间通讯方式)方式进行工作，主从通讯严格按循环方式工作，所带看门狗定时器(Watchdog Timer)将在极短的时间内检查出主从站是否有故障，而且经过扩展的 PROFIBUS-DP 诊断功能可实现对故障的快速定位，每个站还可方便地改变其站号。

更多的信息可以到 WWW.PROFIBUS.COM 上获得。

10.3.1 硬件介绍

PROFIBUS-DP 主要采用 RS-485 方式传输，也可采用光纤技术传输。采用 RS-485 传输技术的基本特征如下：

- 网络拓扑：线性总线，两端有有源的总线终端电阻
- 传输速率：9.6K bit/s 至 12M bit/s
- 传输介质：屏蔽双绞线，也可取消屏蔽(取决于环境 EMC)
- 站点数：每个分段 32 个站，最多 126 站(含段间中继器)
- 连接插头：最好使用 9 针 D 型插头

安装要点：

1. 全部设备均与总线连接。
2. 每个分段上最多可接 32 个站(主站或从站)。
3. 每段的头和尾各有一个总线终端电阻，确保操作运行不发生误差。两个总线终端电阻必须永远有电源。

见图 1 所示。

4. 分段站超过 32 个时，必须使用中继器用以连接各总线段。串联的中继器一般不超过 3 个，见图 2 所示。
5. 电缆最大长度取决于传输速率。如使用 A 型电缆，则传输速率与长度如下表：

波特率 (K bit/s)	9.6	19.2	93.75	187.5	500	1500	12000
距离/段 (M)	1200	1200	1200	1000	400	200	100

波特率与传输距离

6. A 型电缆参数:

阻抗: 135-165W 电容: < 30pf/m 回路电阻: 110W

线规: 0.64mm² 导线面积 > 0.34mm²

7. RS-485 的传输技术的 PROFIBUS 网络最好使用 9 针 D 型插头, 插头针脚定义和接线请见下图 12.3。

8. 当连接各站时, 应确保数据线不要拧绞, 系统在高电磁发射环境 (如汽车制造业) 下运行应使用带屏蔽的电缆, 屏蔽可提高电磁兼容性 (EMC)。

9. 如用屏蔽金属编织线和屏蔽箔, 应在两端与保护接地连接, 并通过尽可能大的面积屏蔽接线来覆盖, 以保持良好的传导性。另外建议数据线必须与高压线隔离。

10. 超过 500Kbit/s 的数据传输速率时应避免使用短截线段, 应使用市场上现有的插头可使数据输入和输出电缆直接与插头连接, 而且总线插头可在任何时候接通或断开而并不中断其它站的数据通信。

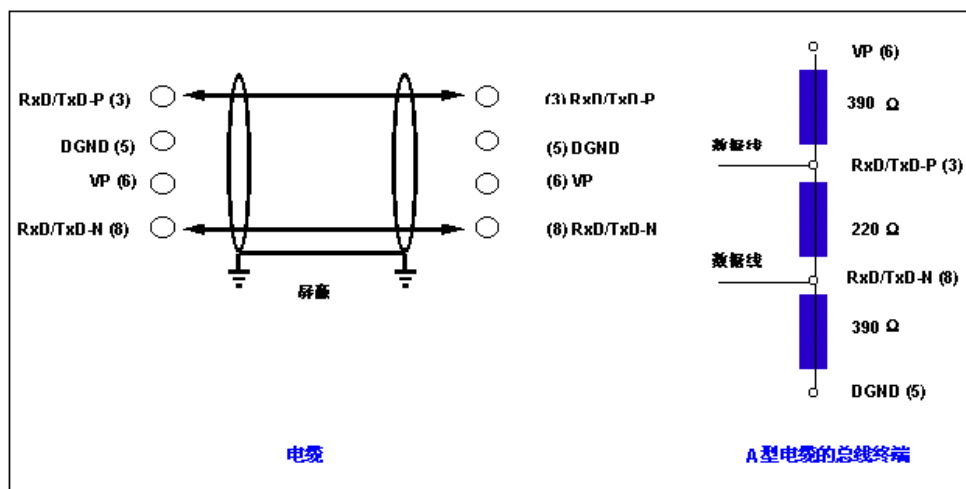


图 10.1 PROFIBUD-DP 和 PROFIBUS-FMS 的电缆接线和总线终端电阻

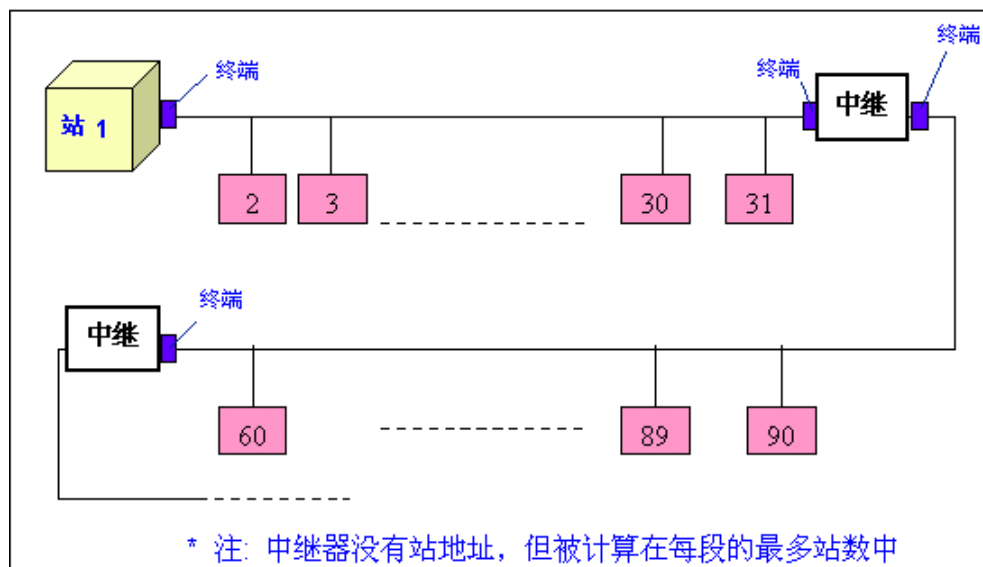


图 10.2 每个分段上最多可接 32 个站（主站或从站）

如下图所示，PROFIBUS 连接到 ED 伺服的 PROFIBUS 接口，接口引脚定义如下：

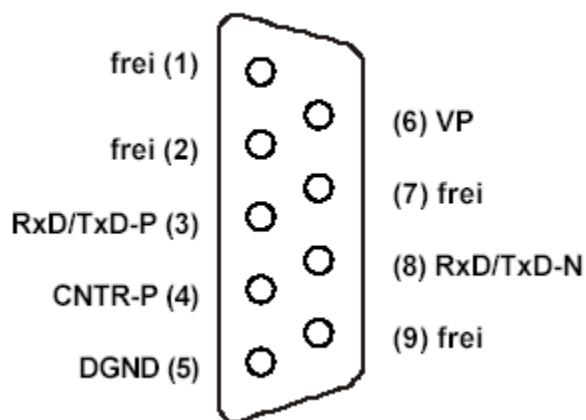


图 10.3 ED 伺服 PROFIBUS 接口引脚定义

依据 EN50170 的标准，包括支持 PIN4 的中继器辅助控制信号 CNTR-P。注意接口的电源由 ED 伺服本身提供，不要外接。

ED 伺服支持下列波特率：9.6Kbit/s、19.2Kbit/s、45.45Kbit/s、93.75Kbit/s、187.5Kbit/s、0.5Mbit/s、1.5Mbit/s、3Mbit/s、6Mbit/s、12Mbit/s，并且接口具备波特率自适应能力，无须设置。

站号通过 ED 伺服上的拨码开关进行设置（S0～S3, 01～15），站号偏置由内部参数寄存器设置（2F800008），注意站号 00 不允许使用。

状态机

ED 伺服作为标准 PROFIBUS-DP Slave, 其工作的状态机理满足 EN50170 中所制定的，如下图所示：

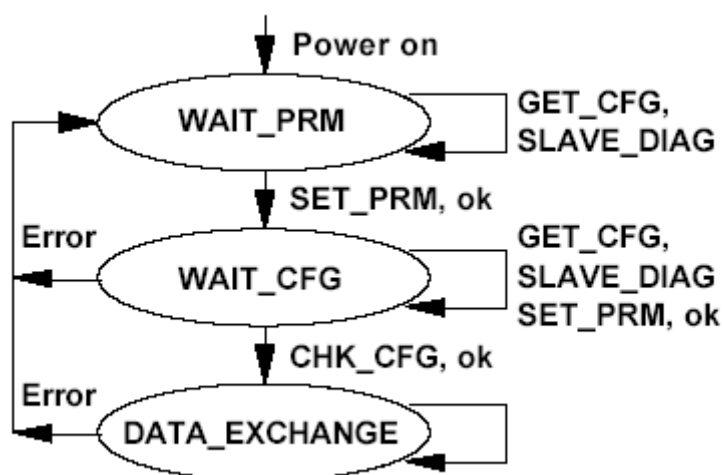


图 12.4 ED 伺服工作状态机理图

在 ED 伺服的逻辑电压上电后，ED 伺服进入“WAIT_PRM”等待参数初始化，此时 Slave 设备等待接收主站的参数化报文，不能进行数据交互；

如果 Slave 收到参数化报文，就会进入等待组态报文“WAIT_CFG”状态，这时主站需发送从站的组态报文（规定该从站的 I/O 字节数等），从站将此报文与已经存储的组态进行比较，如果正确则进入数据交互“DATA_EXCHANGE”，否则不进行数据交互；

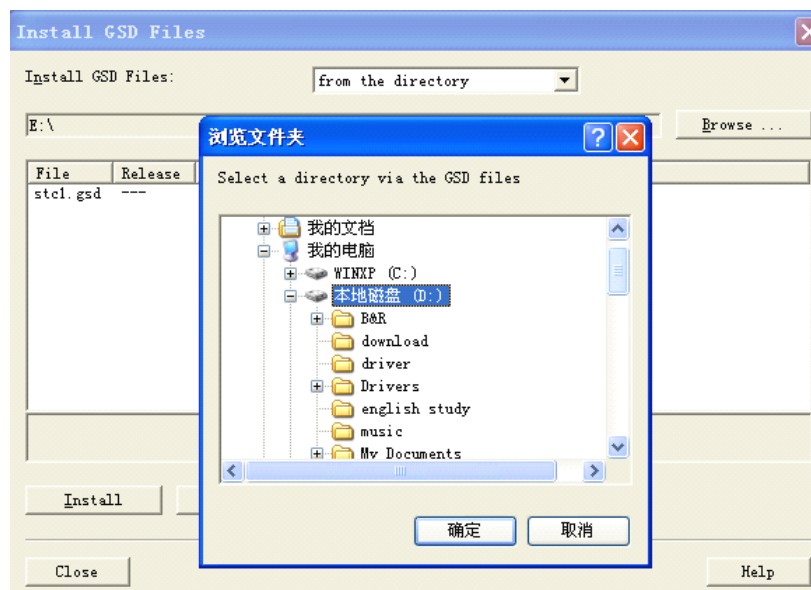
“DATA_EXCHANGE”期间，从站与主站将循环进行数据交互。

10.3.2 软件介绍

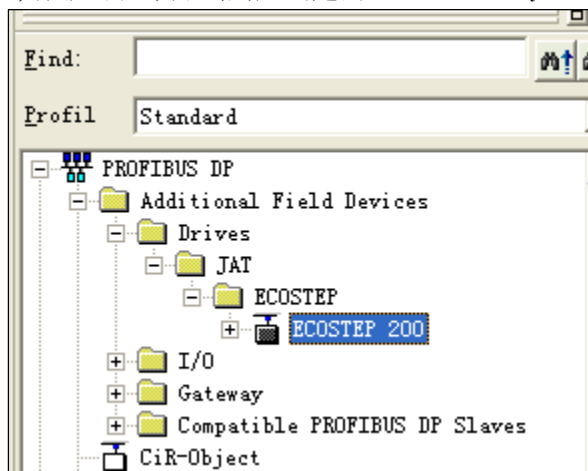
这部分将介绍如何通过 STEP-7 软件对伺服所要交换的数据及其总线通讯参数进行配置。

10.3.2.1 安装 GSD 文件

先在 STEP7 软件的 HW CONFIG 中进行安装 GSD 文件，在安装前首先把 HW config 软件里的工程关掉，然后在点击安装即可，如下图：



安装完毕后在“CATALOG”中“PROFIBUS DP\Additional Field Devices\Drivers”下可以发现“JAT\ECOSTEP\ECOSTEP200”，你就可以添加到用户创建的 DP master system 上。



在 ED 伺服的 GSD 文件中以模块方式提供最常见的运动控制状态的设置（回零、位置等），进一步还提供可供随机读写的寄存器模块。

如果你希望配置在 GSD 中未曾定义的对象，你可以非常方便地通过修改 GSD 文件中相关描述来实现，例：

```
Module="W controlword [6040,0]" 0x83,0xC0,0x60,0x40,0x00
```

```
EndModule
```

你只需拷贝一份，修改相关参数即可，比如添加控制模式操作：

```
Module="W TimerControl[2130,01]" 0x83,0xC0,0x21,0x30,0x01
```

```
EndModule
```

这样你重新安装 GSD 文件后，就会增加一个模块配置选项了。

10.3.2.2 GSD 文件模块配置

模块配置格式如下：

命令符	长度	INDEX (HIGH)	INDEX (LOW)	SUBINDEX
-----	----	--------------	-------------	----------

1)、循环读取命令

命令符 0x43

数据长度（紧跟在 SUBINDEX 后面的数据）描述分字节和字，由于 ED 所有参数的数据长度分为 1BYTE、2BYTE、4BYTE，所以描述数据长度的标识有 0x80 (1BYTE)、0x81 (2BYTE)、0x83 (4BYTE)、0xC0 (1WORD)、0xC1 (2WORD) 五种，发送时先送高字节后送低字节，发送缓冲区中数据将被连续刷新（通常在几个毫秒）。

例如：配置的数据 0x43、0xC1、0x60、0x63、0x00 表明参数地址

0x606300 (Position_actual_value, 实际位置值) 中的数值 (2WORD) 将被放入发送区（紧跟在字节 0x00 后面）。

2)、循环写命令

命令符 0x83

3)、随机进行数据读、写命令

命令符 0xC3

配置的格式固定如下：

0xC3	0x87	0x87	0x00	0x00	0x00
------	------	------	------	------	------

说明后面将紧跟两组各 8 个字节的输入和输出数据，这两组数据的格式完全一样，说明如下：

Index (H)	Index (L)	Sub	Command	DATA (H)	DATA	DATA	DATA (L)
-----------	-----------	-----	---------	----------	------	------	----------

其中 Command（控制命令）是关键，其格式如下：

SCS/CCS	Toggle	Len.	Exp.	Lsp.
---------	--------	------	------	------

CCS/BIT5..7：给服务端的控制命令

Client Command Specify(2=读请求，1=写请求)

CCS/BIT5..7：给客户端的控制命令

Client Command Specify(2=读应答，1=写应答，4=错误信息)

Tgl/Bit4：Toggle 位，被从请求报文中复制到响应报文中

Len/Bit2~3：数据 BUFFER 中未被使用的字节数（0~3）

Exp/BIT1：快速/片断传输

Lsp/Bit0：长度描述

其中 BIT0~4 在请求报文中将被忽略，但在应答报文中起作用。

如果一个请求不能得到正确回应，CCS 将等于 4（BIT7 将被置“1”），并且由于 ED 是作为 CANopen 的一个设备，所以依据 CANopen 的 DS301 标准，在数据区会被放置一个故障代码。

举例说明：

A) 主站向从站读取地址 0x100000(设备行规描述)，从站的回答将是：0x00000192(即十进制 402，CANopen 标准 DS402)

请求报文：0x10, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00

回应报文：0x10, 0x00, 0x00, 0x43, 0x00, 0x00, 0x01, 0x92

B) 主站向从站地址 0x607300 设定最大电流 1023 (HEX 03FF)

请求报文: 0x60, 0x73, 0x00, 0x20, 0x00, 0x00, 0x03, 0xFF

回应报文: 0x60, 0x73, 0x00, 0x60, 0x00, 0x00, 0x03, 0xFF

C) 主站向从站地址 0x607000 设定最大电流 3071 (0x0BFF), 这个值超过最大值 2047, 所以会出错, 于是类型 6、代码 9 错误代码 31 的报文回传

请求报文: 0x60, 0x73, 0x00, 0x20, 0x00, 0x00, 0x0B, 0xFF

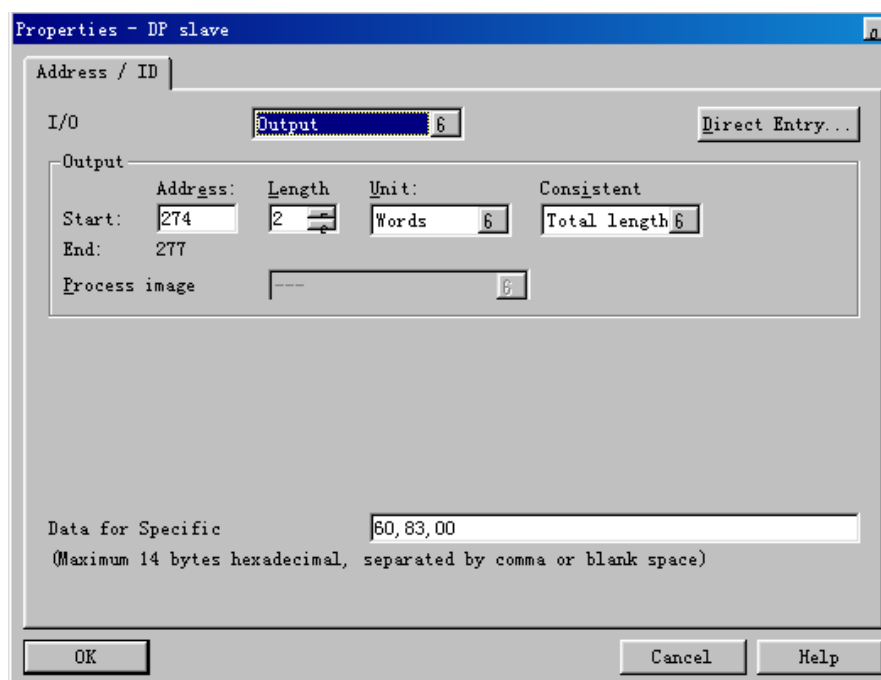
回应报文: 0x60, 0x73, 0x00, 0x80, 0x06, 0x09, 0x00, 0x31

D) 主站向从站请求调用子程序 SEQ02

请求报文: 0x21, 0x18, 0x00, 0x20, 0x00, 0x00, 0x00, 0x02

回应报文: 0x21, 0x18, 0x00, 0x6C, 0x00, 0x00, 0x00, 0x02

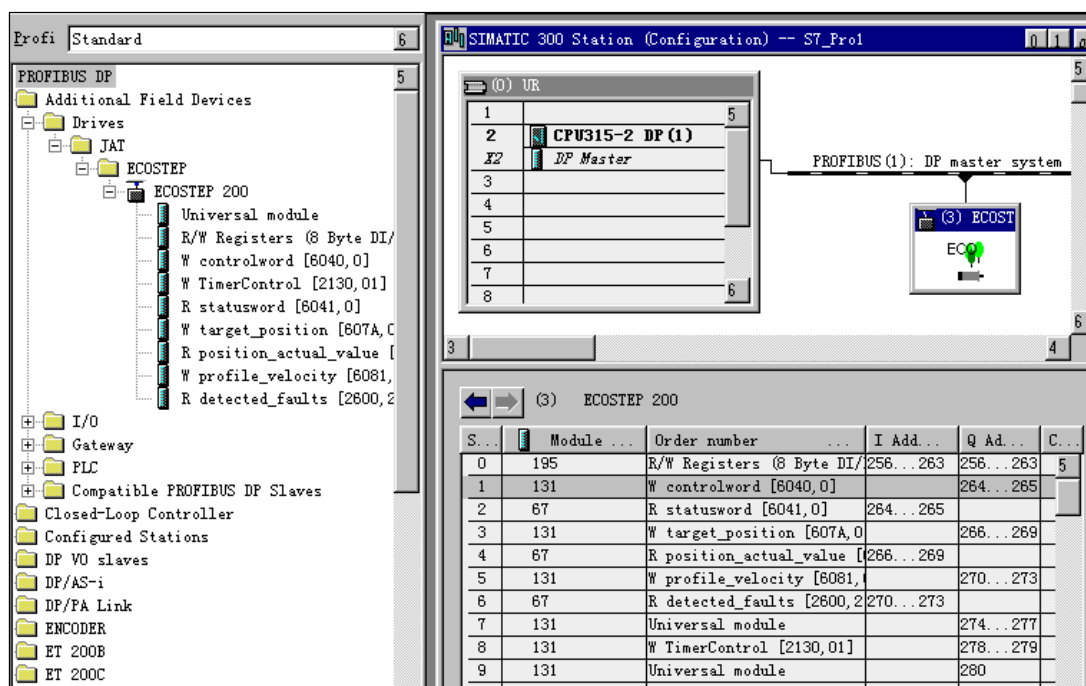
当然用户也可以通过在 HD 配置 Universal Module, 从其 DP-SLAVE 的 PROPERTY 菜单中设置输入和输出结构, 例如我们准备设置 Profile_acceleration, 对应 Index 0x6083, SUB00, 类型 UNSIGNED32, OUTPUT, 如下图:



我们可以看到, 最上方可以选择 I/O 类型, 我们选择 OUTPUT, 下面地址我们根据模块插入位置依次选择, 根据数据类型我们选择 2WORDS, 为了让数据一致传输, 最后一项请选择“TOTAL LENGTH”, 在下方我们输入参数的地址描述 INDEX 6083, SUB 00。建议用户均采用这样的方式来访问 GSD 文件里没有的 OD。

用户可在 PLC 的资源中选择一些地址寄存器对配置的参数进行存储, 可以通过符号表对对应配置参数进行描述, 注意每次要确保一次性将配置长度的数据进行处理, 对于 8 字节的随机数据读写, 尤其要注意, 因为 STEP7 中一般只能处理 4 字节, 所以请注意调用 SFC14(DPRD_DAT)和 SFC15(DPWR_DAT)来实现。

整个配置画面如下:



10.3.2.3 诊断

ED 伺服的头 6 个字节 (BYTE0~5) 诊断信息与 PROFIBUS 的标准相一致。此外字节 6 (描述外部诊断长度, 包含此字节) 会说明另外还有 4 个 BYTES 的进一步诊断数据提供对 ED 的深入诊断, 这 4 个字节的含义如下:

ERROR BIT (故障表示位)	ERROR EVENT (故障事件)
BYTE7, BIT0	FAULT_H8SWD_BIT
BYTE7, BIT1	FAULT_REGLERWD_BIT
BYTE7, BIT2	FAULT_ENC_ERROR_BIT
BYTE7, BIT3	FAULT_MOTENCCAP_BIT
BYTE7, BIT4	FAULT_MAENCCAP_BIT
BYTE7, BIT5	FAULT_OVERTEMP_BIT
BYTE7, BIT6	FAULT_UVMESS_BIT
BYTE7, BIT7	FAULT_OV_ERROR_BIT
BYTE8, BIT0	FAULT_UV_ERROR_BIT
BYTE8, BIT1	FAULT_A_ERROR_BIT
BYTE8, BIT2	FAULT_B_ERROR_BIT
BYTE8, BIT3	FAULT_OUT_DIAG_BIT

BYTE8, BIT4	FAULT_EX_ENABLE_BIT
BYTE8, BIT5	FAULT_FOLLOWINGERROR_BIT
BYTE8, BIT6	FAULT_OVERSPEEDERROR_BIT
BYTE8, BIT7	FAULT_COMMUFINDERROR_BIT
BYTE9, BIT0	FAULT_ABORT_CONNECT_BIT
BYTE7, BIT0	RESERVED (NOT USED)
BYTE7, BIT1	RESERVED (NOT USED)

上述故障表格在 GSD 文件中也有描述, 所以也可产生普通文本信息。

如果一个故障使能位在驱动器内部被置位将立即使外部诊断位 EXT_DIAG 对应的红灯闪烁，EXT_DIAG 所对应的诊断报文第一字节 BYTE0 的 BIT3 被置位。这样新的诊断信息只有在故障标志变化后才能被发送。

例：电机的编码器没有安装上，这将引起 FAULT_ENC_ERROR_BIT(BYTE7,BIT2) 置位和 EXT DIAG(BYTE0,BIT3)置位，诊断报文如下：

0000: 08 0C 00 02 05 4B 05 04 00 00 00

EXT DIAG 置位

FAULT ENC ERROR BIT 置位

在 CPU 的故障诊断中，也可看到信息：



10.3.2.4 操作服务

ED 伺服支持 SYNC 和 UNSYNC 全局控制命令用来同步数据传输和依据标准清除数据 (CLEAR DATA)。

如果 ED 伺服收到一个 SYNC 命令，将锁定输出数据在当前状态，并在随后的数据传输周期中，存储从 PROFIBUS 收到的所有输出数据，而其输出状态保持不变，直到 ED 伺服接收到下一个 SYNC 命令时，所存储的输出数据才发送到外部设备上。通过 UNSYNC 命令 ED 伺服退出 SYNC 模式。

10.3.3 Profibus 通讯案例

详细案例请看附录。

第十一章 常用对象列表

按照十章介绍的数据通讯协议，你用 JAVA, Visual C++ 或 Delphi 等软件，能编程实现各种控制方式。理解下面这些表格中数据将对程序员和使用者带来帮助。所有的参量都是用十六进制的数据格式传送的。在下面文档中我们采用 16 进制方式，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, 还有此寄存器的存储位数和读写属性，读或写标识(RW), 只读或只写标识(RO, WO)，映射标识(M)。

模式及控制: 0x6040

地 址	子地址	位数	命令 类型	单位	详细解释
6040	00	10	RW, M	bitcode	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 0x3F 立即进入绝对定位方式 0x5F 进入相对定位方式 0x1F 原点定位，(先0x0F) 0x80 清除内部故障
6041	00	10	RO, M	bitcode	状态字节显示驱动器的状态 0x0001 准备上电 0x0008 检测到错误 0x0400 目标到达 0x4000 控制器得电 0x8000 原点找到
6060	00	08	WO, M	number	工作模式: 1 带位置环的定位模式 3 带位置环的速度模式 -3 速度环(立即速度模式)

					-4 主从或脉冲/方向控制模式 6 回原点模式 7 基于CANopen的运动插补
--	--	--	--	--	--

测量数据: 0x6063

地 址	子地址	位 数	命令类型	单位	详细解释
6063	00	20	RO, M	inc	实际位置值
6069	00	20	RO, M	inc/64s	实际速度值
606B	00	20	RO, M	inc/64s	指令速度值
606C	00	20	RO, M	inc/64s	滤波后的实际速度值
6078	00	10	RO, M	整数	实际电流值
60FD	00	20	RO, M	位码	10个数字输入的状态字 (DIN1-8, RESET, External ENABLE): 0x0201 0000 DIN1为高电平 0x0280 0000 DIN8 (原点) 为 高电平 0x02C8 0000 DIN8, 7和4为高 电平

目标对象: 0x607A

地 址	子地址	位 数	命令类型	单位	详细解释
607A	00	20	RW, M	inc	工作模式1下的目标位置, 如果控制字设定为开始运动, 转变成指令位置。
60FC	00	20	RO, M	inc	工作模式1时的指令位置。
6081	00	20	RW, M	inc/64s	工作模式1时的梯形曲线的最大速度。
6083	00	20	RW, M	16inc/s ²	梯形曲线的加速度. 1000rad/s ² 约等于 80000inc/s ²
6084	00	20	RW, M	16inc/s ²	梯形曲线的减速度。

60FF	00	20	RW, M	inc/64s	在模式3、-3或-4时的目标速度。
6073	00	10	RW, M	整数	最大电流：见术语表
607F	00	20	RW, M	inc/64s	在模式1或3下的最大的极限速度 例如：分辨率为8000inc 转速1000RPM = 8533333 inc/64s

性能对象: 0x6065

地 址	子地址	位 数	命令类型	单位	详细解释
6065	00	20	RW, M	inc	设置为报警的最大位置跟随误差值： 2000 默认值 60 性能调整好的的驱动器
6067	00	20	RW, M	inc	“目标位置到达”的位置范围—默认值是10
607D	01	20	RW, M	inc	最小的软件设定极限位置
607D	02	20	RW, M	inc	最大的软件设定极限位置 (这两项如果都为零则没有限制)

原点控制: 0x6098

地 址	子地址	位 数	命令类型	单位	详细解释
6098	00	08	RW, M	整数	方式：重要的回原点的方式（从1到34） 34 将实际位置置零
6099	01	20	RW, M	inc/64s	寻找极限开关的速度
6099	02	20	RW, M	inc/64s	寻找N相信号的速度
609A	00	20	RW, M	16inc/s ²	加速度
607C	00	20	RW, M	inc	原点偏移

电机参数: 0x60F6

地 址	子地址	位 数	命令类型	单位	详细解释
-----	-----	-----	------	----	------

60F6	01	20	RW, M	整数	每极长度的脉冲增量 如50极、8000解析度的马达的 每极脉冲增量为160
60F6	02	10	RW, M	整数	极数
60F6	03	10	RW, M	整数	电流角度的相偏移, 该值正比 于速度值, 范围在80至400之间。
60F6	05	10	RW, M	整数	电流角度的最大相偏移, 极长 度的1/4[INC]
60F6	06	10	RW, M	整数	拖动负载的峰值电流
60F6	07	10	RW, M	整数	在传动方式1或3状态下的时 间延时 500 小负载(1-5倍的马达 惯量) 1000 大负载(20-50倍的马 达惯量)
60F6	09	10	RW, M	整数	传动方式 3 一般的应用 8 上电不抖动应用
60F6	11	10	RW, M	12A/2047	I ² T 功率限制 0-2047 为有效的范围
60F6	12	10	RW, M	[S]	测量有效电流的时段 0-12000 所有马达的时间常 数约为20分钟 10 起安全作用的最小 值

速度环参数: 0X60F9

地 址	子地址	位 数	命令类型	单位	详细解释
60F9	01	10	RW, M	inc/s	VC-KP 速度环的比例增益 50 软的增益 200 硬的增益
60F9	02	10	RW, M	整数	VC-KI 速度环的积分增益

					0 没有瞬间偏差修正 1 默认值 2 很强的修正，可导致振动
60F9	03	10	RW, M	整数	VC.KILIM, VC.KI的限定值, VC.KILIM应当高于静态电流的10%。
60F9	04	10	RW, M	整数	数字输入滤波值—速度环增益 $=EFILT * VC.KP$ 1 默认值，无超前补偿 2 很强的超前补偿 3-5 重负载-低滤波值
60F9	05	10	RW, M	整数	数字输出滤波值（速度环的） 2-5: 1-10倍的负载/马达惯量比 15-45: 10-100倍的负载/马达惯量比

位置环参数: 0X60FB

地 址	子地址	位 数	命令类型	单位	详细解释
60FB	01	10	RW, M	无符号	PC.KP 位置环的比例值， 例如 1000 默认值，软的修正值 3000 中等性能值 8000 好的性能值，低跟踪误差，高位置刚度
60FB	02	20	RW, M	无符号	PC.AMAX 产生最好效果，防止振动的值 该值= $MO / (\text{负载/马达惯量比} * 16 * 2 * PI) * \text{编码器的解析度}$
60FB	03	10	RW, M	整数	PC.VFFF 为了得到更高的动态性能的前馈速度增益 0-3000 平滑的起动

					12000-16384 迅速起动,低跟踪误差,这个参数对主从跟踪的应用很重要.
60FB	04	10	RW, M	整数	PC-AFFF 好的动态性能的前馈电流 该值= $12A/2^{23}$ *加速度

用于存储的参数:0x1010

说明,在上述寄存器中输入“65766173”实现存储参数的功能,系统随后会自动将此寄存器的值再置为“1”

地 址	子地址	位 数	命令类型	单位	详细解释
1010	01	20	RW	LOGIC	保存所有的参数—值为65766173
1010	02	20	RW	LOGIC	保存通讯的参数—值为65766173
1010	03	20	RW	LOGIC	保存应用的参数—值为65766173
1010	04	20	RW	LOGIC	保存离线程序 —值为65766173

CAN-PDO 参数: 0X1400-0X1A00

0X1400-7 (RX-特性参数/读)

0X1600-7 (RX-映射)

0X1800-7 (TX-特性参数/写)

0X1A00-7 (TX-映射)

程序参数:0X2000-FF,0X2120,0X21221

说明,在一个程序XX里最多可存8个参数(20XX0220=>20XX0320)到(20XX1020=>20XX1120)。

地 址	子地址	位 数	命令类型	单位	详细解释
2000	01	08	RW	LOGIC	激活—20XX指程序0XXX 0: 程序无效 1: 程序有效
2000	02	20	RW	无符号	参数0X20000220, 0号程序第一个参数的目标位置
2000	03	20	RW	无符号	0号程序第一个参数的值
.....				
2000	10	20	RW	无符号	0号程序第八个参数的目标位置
2000	11	20	RW	无符号	0号程序第八个参数的值

下面16个寄存器对应16种触发调用(由数字量DIN1~8状态变化触发),寄存器存的是被调用的程序号

2120	01	10	RW	无符号	程序被输入点DIN1L-H激活 寄存器值0X0020 表明程序0X20被调用
.....					
2120	08	10	RW	无符号	程序被输入点DIN8L-H激活 寄存器值0X0010 表明程序0X10被调用
2120	09	10	RW	无符号	程序被输入点DIN1H-L激活
.....					
2120	10	10	RW	无符号	程序被输入点DIN8H-L激活

DIN1~4 编码组合（用来触发程序）

0X21200608=>0X0F00 是另一个程序号编码被调用的例子，在这个例子里，当 21200510（DIN5 L-H）中输入 0F00 时，DIN1-4 组成的二进制编码所代表的程序号在 DIN5 由 L-H 变化时就会被调用。

要完成上面 16 种触发调用必须给予相应调用以使能，0X2121 的值就是这个功能，0X2118 可以实现对 2000~FF 任意一个程序的直接调用。

地 址	子地址	位 数	命令类型	单位	详细解释
2121	00	10	RW	无符号	高BYTE对H-L使能，低BYTE对L-H使能 0X1F1F DIN1-5 L-H 和H-L时均可被激活 0X010F DIN1 H-L 和DIN1-4 L-H 可被激活
2118	00	08	RW	无符号	用在编程当中，直接调用程序号 0X21180008=>0X20 表示0X20程序被调用。

定时器参数:0X2130

地 址	子地址	位 数	命令类型	单位	详细解释
2130	01	10	RW	80XX	在一段等待时间之后,开始执行指定的程序 0X8012 开始执行程序0X12

2130	02	20	RW	[ms]	等待的时间 0X21300110=>0X8012 0X21300220=>0X03E8 等待一秒之后开始执行程序0X12
------	----	----	----	------	---

事件参数:0X2140

地 址	子地址	位 数	命令类型	单位	详细解释
2140	01	10	RW	80XX	目标位置到达后, 执行XX段程序。
2140	02	10	RW	80XX	原点到达后, 执行XX段程序。
2140	09	10	RW	80XX	Servo off状态, ready 输出为低时执行xx段程序
2140	0A	10	RW	80XX	Servo on状态, ready 输出为高时执行xx段程序
2140	0B	10	RW	80XX	Servo on状态执行xx段程序
2140	0C	10	RW	80XX	工作模式使能时触发程序xx段程序
2140	10	10	RW	80XX	Enable输入触发程序xx段程序

输出参数: 0X2160 (OUT1), 0X2161 (OUT2)

地 址	子地址	位 数	命令类型	单位	详细解释
2160	01	20	RW	无符号	输出1的参数地址
2160	02	20	RW	无符号	偏置值,该值将被加到输出1的参数上
2160	03	20	RW	无符号	与值,该值与先前计算的结果相与.
2160	04	20	RW	无符号	比较值,用来与上面计算结果比较
2160	05	20	RW	无符号	极性控制, 对比较结果取反
2160	08	08	RO	无符号	比较结果 0X21600120=>0X60410010 0X21600220=>0X0037 0X21600320=>0XFFFF 0X21600420=>0XC037 0X21600520=>0X0000 如果驱动器发现交换、原点和在工作中, 比较结果为真, OUT1输出高电平。

限位开关参数: 0X2170, 0X2171 (LIMIT+), 0X2172 (LIMIT-)

地 址	子地址	位 数	命令类型	单位	详细解释
2170	00	08	RW	LOGIC	改变8个数字输入的极性 0X2170008=>0X60(DIN6+7是低电平)
2171	02	08	RW	LOGIC	对位置限位开关(DIN6)进行布尔与 0X21710208 =>0X20(如果输入极性为0X00,那么DIN6为高)
2171	03	08	RW	LOGIC	对位置限位开关(DIN6)进行布尔比较 0X21710208 =>0X20证实是否DIN6为高 0X21710308 =>0X20默认DIN6(0X40默认DIN7)
2171	04	08	RW	LOGIC	极限开关的状态 0X00: 不是高电平 0X01: 高电平,正方向运动不可能.

计数器参数:0X2190(计数器1) 到 0X2193(计数器4)

地 址	子地址	位 数	命令类型	单位	详细解释
2190	01	20	RW,M	无符号	要加到计数器上的值
2190	02	20	RW	无符号	计数器值

捕捉参数:0X21C0

用来捕捉实际位置值。

地 址	子地址	位 数	命令类型	单位	详细解释
21C0	01	10	RW	80XX	如果21C002从0到1, 跳到0xXX程序
21C0	02	20	RW, M	整数	在主编码器上的N-LIMPULS端有一个L-H的触发事件时开始计数, 计数方向为正。
21C0	03	20	RO, M	整数	结果, 如果事件21C002发生, 实际位置值被拷到这个地址。

21C0	04	20	RO, M	整数	如果参数21C002从0到1, 检测实际位置的内容。
------	----	----	-------	----	----------------------------

比较器参数:0X2180, 0X2181,0X2182,0X2183

如果比较值一旦为真,那么整个比较必须再次激活才能继续比较。

地 址	子地址	位 数	命令类型	单位	详细解释
2180	01	20	RW	无符号	用来和比较参数进行比较的参数
2180	02	20	RW	无符号	要增加到比较参数上的偏置值
2180	03	20	RW	无符号	和比较参数进行布尔运算的与值, 默认为0xFFFFFFFF
2180	04	20	RW	无符号	比较值, 和比较参数进行比较
2180	05	10	RW	无符号	操作选择 0x0001 = 等于 0x0002 < 小于 0x0003 <= 等于或小于 0x0004 > 大于 0x0005 >= 大于或等于 0x0006 <> 不等于
2180	06	10	RW	0X80X X	比较状态为真后 ,程序0xXX开始执行 0x21800120 ==>0x606C0020(速度) 0x21800420 ==>0x00823555(1000RPM) 0x21800510 ==>0x0005 0x21800610 ==>0x8012(只执行一次)
2180	08	20	RW, M	无符号	临时存储地址(存储SUB01中的值, 以16进制表示)
2180	09	20	RO	无符号	如果比较为真,值为1; 否则,值为0.

数学计算参数: 0X21A0

地 址	子地址	位 数	命令类型	单位	详细解释
21A0	01	20	RW	无符号	应当修改的源参数
21A0	02	20	RW	无符号	获取运算结果的目标参数
21A0	03	20	RW M	无符号	和源参数一起运算的数字
21A0	04	10	RW	无符号	四则运算: 0x0000 拷贝 0x0001 + 0x0002 - 0x0003 * 0x0004 /
21A0	05	20	RO	0X80XX	运算结果: 0x21A00120 =>0x21900120 0x21A00320 =>0x00000002 0x21A00410 =>0x0003(*) 0x21A00220 =>0x6070A220, 此为最后一步 (目标位置) 表1里的值乘以2,结果作为目标参数放在目标位置上.

表格参数:0x21B0

用来将值输进内部表格0x2D00-FF

地 址	子地址	位 数	命令类型	单位	详细解释
21B0	01	20	RW	无符号	源参数应当被放到这个表格
21B0	02	08	RW	无符号	写命令
21B0	03	08	RW	整数	表格里的位置 0x21B00120 =>0x60630020 实际位置 0x21B00308 =>0xFF 表格位置255 0x21B00208 =>0x01 写命令

					将实际位置写到表格位置 255 —用于示教.
--	--	--	--	--	-------------------------------

记录用的参数: 0X2201-FF

地 址	子地址	位 数	命令类型	单位	详细解释
2201	01	20	RW	无符号	第一个记录参数
2203	01	20	RW	无符号	排列上一个参数的记录值
2201	03	20	RW	无符号	第二个记录参数
2203	02	20	RW	无符号	排列上一个参数的记录值
2201	05	20	RW	无符号	第三个记录参数
2303	03	20	RW	无符号	排列上一个参数的记录值
2201	07	20	RW	无符号	第四个记录参数
2203	04	20	RW	无符号	排列上一个参数的记录值
2210	00	10	RW	无符号	计数器的大小, 指定队列有多大
2211	00	10	RW	无符号	记录队列里的位置
2214	00	10	RW	无符号	记录队列里的时间解析度

关于记录的例子: 实际速度

0x22010120 =>0x606C0020 实际速度

0x22140010 =>0x0005 时间解析度为5毫秒

0x22100010 =>0x01F4 开始记录500个数值

读出这个记录队列

0x22110010 =>0x1F4 第一个记录的实际速度值

0x22030120 读数[T=0.005]

0x22110010 =>0x0001 最后一个记录值

0x22030120 读数[T=2.500]

监控参数:0x2400,0x2401

ED有两个独立的模拟量监控输出点,每个点对应内部的数值.输出范围是0-5V,0用2.5V表示.

比例公式是:

$$[V / dimension] = 1V * internal dimension * factor / (256^{(1+preshift)})/120$$

地 址	子地址	位 数	命令类型	单位	详细解释
2400	01	20	RW	无符号	对象, 映射到模拟输出口1(Source value)
2400	02	08	RW	无符号	Preshift, 一般设定为0、1或2
2400	03	10	RW	无符号	Factor, 一般设定在0x0001-0x7FFF之间
<p>以上为监控口1的参数,监控口2要将地址改为2401</p> <p>例如: 设置模拟量输出口2输出电压对应实际电流值变化</p> <p>0x24010120 =>0x60780010</p> <p>0x240102008 =>0x0000 preshift=0</p> <p>0x24010310 =>0x001E factor=30</p> <p>0x24010410 =>(Source * factor) / (256^(preshift+1))</p>					

模拟量输入口:0x2508

模拟量口AIN+和AIN-可以映射给每一个内部对象(RW), 这个功能主要是用来设定速度值和电流的极限值。这个对象的内部范围是-512—512 DAC。

地 址	子地址	位 数	命令类型	单位	详细解释
2508	01	20	RW	无符号	映射目标地址, 如目标速度
2508	02	10	RW M	无符号	比例,通过公式来计算: 最大值 / 内部单位 / 2 ^{SHIFT} / 512
2508	03	08	RW	无符号	Shift 通常是0、1、2或3
2508	04	10	RO M	Int	来自模数转换器的值VALUE COMING FROM ADC
<p>例如: 对应最大转速为1500RPM的速度环的+/-10V的输入</p> <p>0X25080120 =>0x60FF0020 目标速度</p> <p>0X25080210 =>0x03 shift是3</p> <p>0X24010310 =>0x0C35 factor是3125</p>					

主从跟踪的参数:0x2509

该对象对应X7输入口。如果要用主从跟踪控制模式或脉冲/方向控制模式,那么就常用到如下参数

地 址	子地址	位 数	命令类型	单位	详细解释
2509	02	20	RW	无符号	映射速度到(60FF0020)
2509	03	10	RW M	无符号	电子齿轮比的分子
2509	04	10	RW M	无符号	电子齿轮比分母

2509	05	08	RW	无符号	模式: 0、1是4倍频解码; 2是脉冲方向控制 3CW/CCW
2509	06	20	RW M	无符号	主轴位置值, 可写可读, 往往做其它用途
2509	07	20	RW M	无符号	从轴位置值
2702	00	10	RW M	无符号	0, 编码器输出为电机编码器信号 1, 编码器输出为主编码器输入信号
2509	12	10	RW M	无符号	对主编码器输入增加滤波, 以前如果电子齿轮比太大会有噪音, 目前可以解决电子齿轮比太大引起噪音问题。

例如:电子齿轮比

0X250902200 =>0x60FF0020 目标速度

0X25090310 =>0x07D0 分子是2000

0X25090410 =>0x03E8 分母是1000

从轴以两倍的速度去跟踪主轴, 比例值可以通过内部比较器或外部模拟量输入改变.

滤波使用方法: 如果设置电子齿轮比为 20:1, 如果我们设置滤波为 12,

那么我们设置如下电子齿轮比 $\text{gear factor}/(\text{gear divider}*\text{Master Config.Filter})$

$\text{Gear factor}=20, \text{gear divider}=12.$

即: $\text{target velocity}=\text{master}*\text{gear factor}/(\text{gear divider}*\text{Master Config.Filter})*\text{Master Config.Filter}$

地 址	子地址	位 数	命令类型	单位	详细解释
2600	01	20	RW	LOGIC	该标志可用于使错误检测无效
2600	02	20	RW	LOGIC	错误代码
260002 详细报警信息请参考附录“报警信息原因及排除”					

错误代码:0x2600

布尔控制字操作:0x2840

地 址	子地址	位 数	命令类型	单位	详细解释
2840	01	10	RW	LOGIC	与实际控制字进行“和”操作
2840	02	10	RW	LOGIC	与实际控制字进行“或”操作

配置 RS485 和 CANopen 接口。

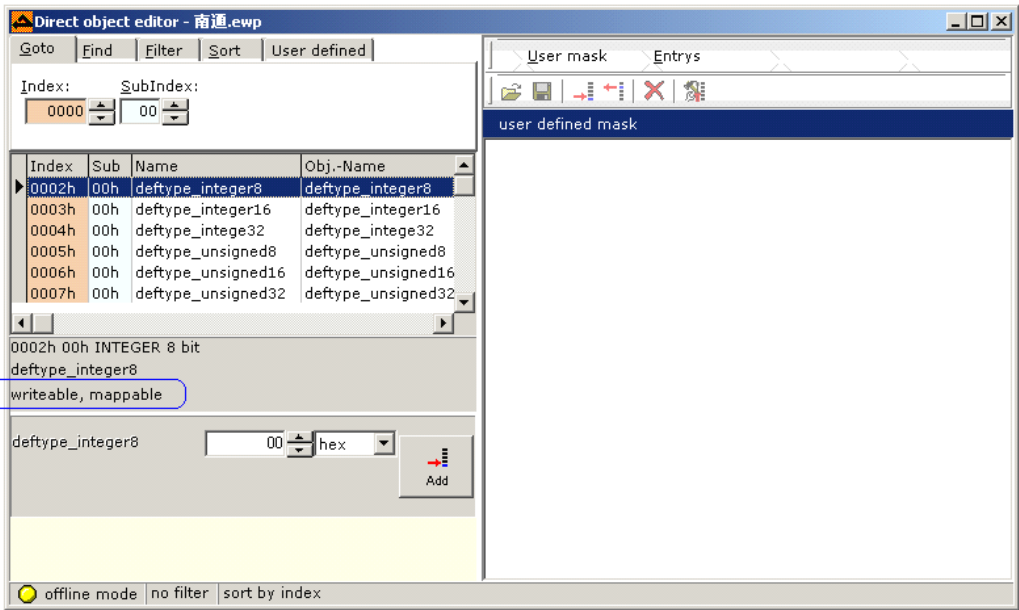
设备信息及软件版本 0X2FE0173

2FE0	03	20	RO	无符号	改型日期 0x0315490 意指20010128,意思是 1/28/2001出产.
------	----	----	----	-----	--

术语表

Mapping（映射）

地址里的值 Subindex-Index-Bitlength 是另外的目标的地址， 这种指针概念在多数高级语言里经常用到，例如实际的主位置地址是 2509-06-20， 如果你想要到一相对位置，你可“映射”它在相对目标地址 2180-01-20。 这种处理叫映射，是很有用的概念，如果目标能被映射则标记“M” 则在目标参数说明中。



Statemachine（机器状态）

电机加上驱动器的系统有不同的组合状态.，从机械工程我们知道空间描述：确定系统任何时候物体在什么地方(s(t), v(t), t)。这对我们的驱动器—电机系统也是一样的。但是我们要知道是否电源开启了，是否哪儿有错误或是否目标达到了和一些更多的东西。因此，在 CANopen 里，我们有状态字，它的代码确定了特殊的驱动器状态，使用控制字能改变驱动器状态。我们要说明下列各项重要的内容,那里我们能观察这些系统状态：

- ③ 接通电源
- ③ 开始回原点
- ③ 以绝对位置方式运行

对象	说明
状态字 60410010	重要标志组成: 错误,通讯,准备, 目标达到,原点发现,电机通、断电。
控制字 60400010	决定状态:电机通电, 使能, 复位,开始运动,绝对或相对运动。
操作方式 60600008	如：1 位置模式, 6: 原点模式。
原点 60980008	如 32 表示“在下一个 Z 相脉冲”。
速度 60810020	速度 v(t)曲线。

加速度 60830020	v(t) 曲线正的斜率。
减速度 60840020	v(t) 曲线负的斜率。
目标位置 607A0020	位置增量。

所以有下列各项过程:

动作/过程	控制字	状态字	状态
逻辑 on!	0x0006	0x0031	准备上电
电源 on!	0x000F	0x4437	通讯发现, 没错, 上电
电源 off!	0x0006	0x4031	没上电

Address (地址)

地址规定: 首先是地址(四个十六进制数组成), 子地址(二个十六进制数组成) 和在末端二个十六进制的代码表示数据的长度,我们拿状态字作例子:

(地址, 子地址, 位数) = (6041,00,10) 或 60410010

Bitcode (位编码)

位数 (位编码) 是指我们用一个代码如(1000 1110), 去表示几个事件的状态, 那可能是 8 位, 16 位或 32 位, 并转换成十六进制数。例如在 8 个数字量的输入口的高低状态, 我们在 DIN6 到 8 是高, (激活的) 和其他的全部为低, 用数 (11100000) 表示, 那是 0xE0。

Velocity internal scaling (速度内部单位)

RPM=编码器分辨率*64/60 [内部单位: inc/64 s]。

Analog dem value (模拟量目标值)

[-10V ..10V] ->比例值 [内部的] /变换因数[内在值/伏特] /2^SHIFT/ 512 ,

最大的速度是 1440U/min => 1440/ [60/ 8000/ 64] / 2^3 / 512 = 3000。

Acceleration (加速度内部单位)

单位是 rad/s² 时, 等于内部值[16 inc/s²]*16*2*pi /编码器分辨率, 如: 一皮带轮是 100 毫米/转, 编码器 8000 inc, 于是等于 9.81 /(0.1/ 2*pi)*8000 / (16*2*pi) = 4969 [16 inc/s²]。

Digital current 数字电流

电流值是从 0 – 2047, 这可以在驱动器里转换: 电流值=参数值* 最大驱动器电流[A] / 2047。

ED100 能到 8 A, 对于 ED430 和 ED620 峰值电流均为 12A, ED630-0250 峰值电流为 18A, ED630-0300 峰值电流为 21A, ED630-0400 峰值电流为 24A。

Digital friction current 数字的摩擦电流

数字的摩擦电流[Idac]: 低速时可间接在平均电流目标地址 6078,00 里算出来 , 这个值*1.2, 可近似作为

速度环积分参数 `vc_kilim`。

参数存储

将设置的参数保存到驱动器内部，这样掉电后重新启动驱动器，新的参数将发挥作用。

附录

附录 A：总线通讯范例

- 一：ED 伺服与 S7-300 利用 Profibus DP 总线进行通讯
- 二：ED 伺服与施耐德 M258 利用 CANopen 总线进行通讯
- 三：ED 伺服与 Codesys 利用 CANopen 总线进行通讯

附录 B：串口通讯范例

- 一．KINCO 触摸屏与 ED 伺服系统的联接与应用
- 二．西门子 S7200 和 ED 伺服通讯
- 三．松下 FP0 与 ED 伺服通讯
- 四．三菱 FX2N 与 ED 伺服通讯
- 五．欧姆龙 CJ1W 与 ED 伺服通讯

附录 C：顺序编程范例

附录 D：应用范例

- 一．主从跟随控制
- 二．模拟量控制 ED 伺服电机速度和位置

附录 E：全闭环控制

附录 F：利用 ECO2LOAD 导入和导出驱动器内参数

附录 G：程序段密码保护设置

附录 H：报警信息原因及排除

附录 I：常见参数单位换算表

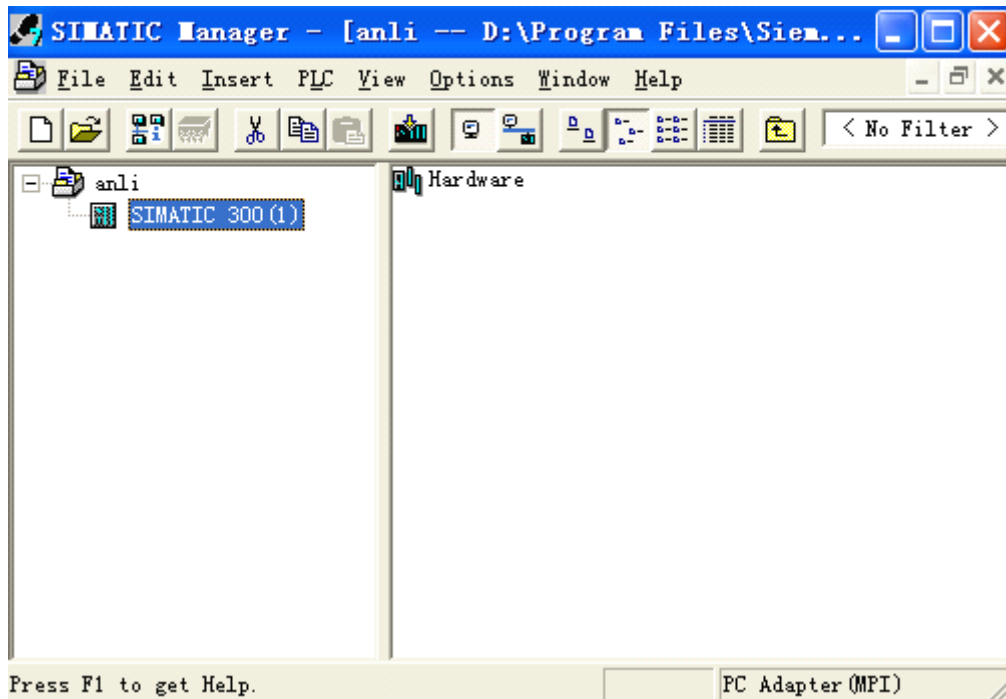
附录 A: 总线通讯范例

一：ED 伺服与 S7-300 利用 Profibus DP 总线进行通讯

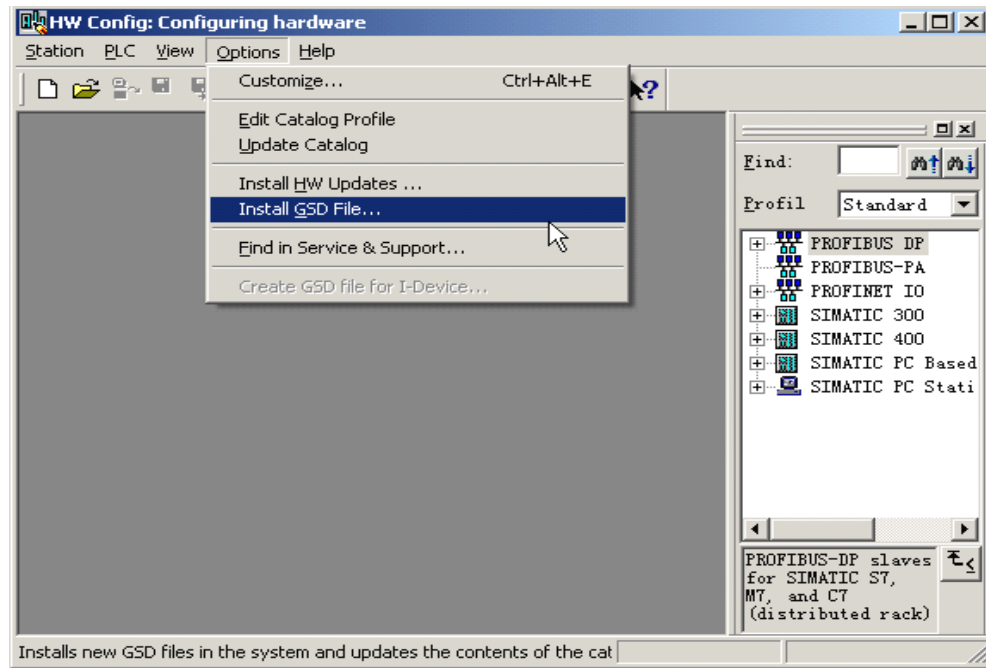
一、硬件连接。请参考第十章 Profibus 总线通讯硬件介绍，建议客户使用西门子原装紫色通讯电缆

二、STEP 7 软件设置

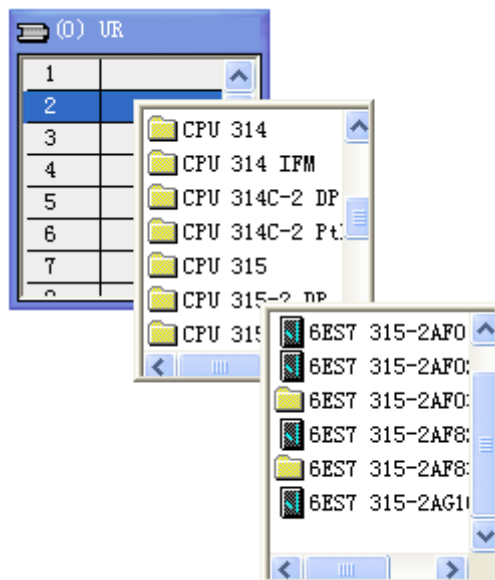
1. 首先新建一个 S7-300 工程：



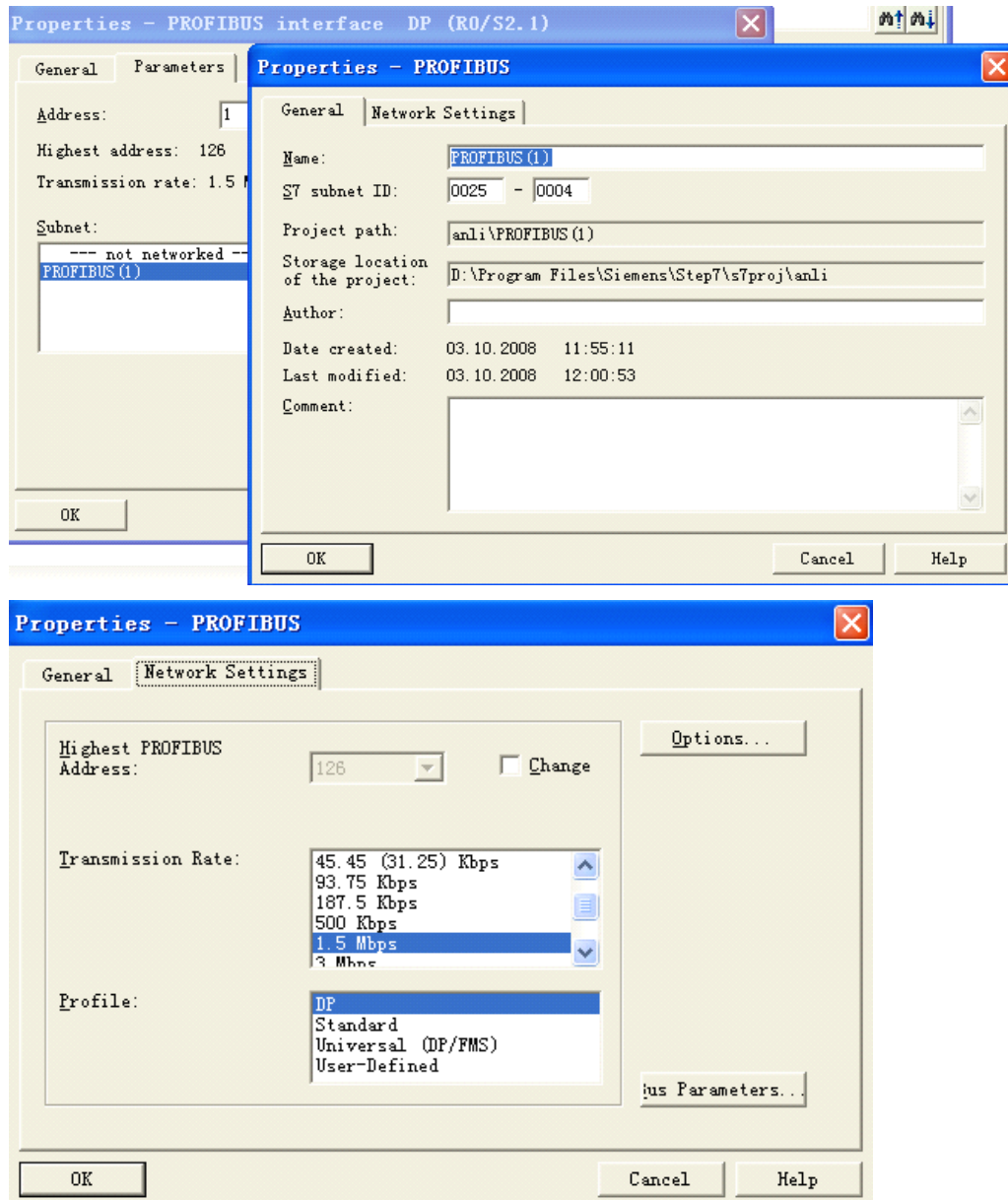
2. 进入“Hardware”进行硬件配置，添加 ED 的 GSD 文件，添加前先关闭当前工程的硬件组态窗口，以免发生冲突添加失败。



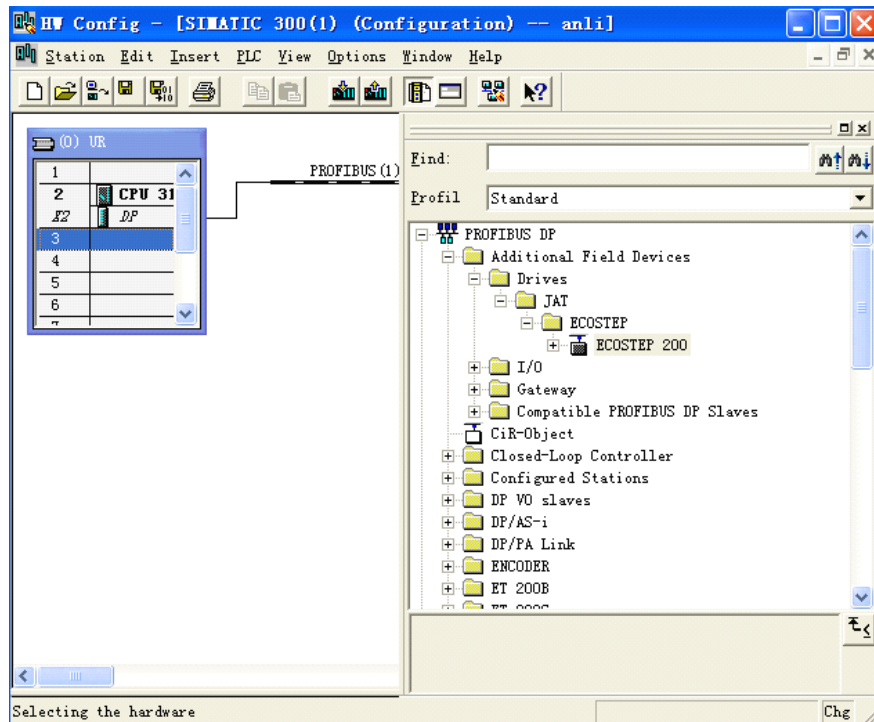
3. 然后再组态，选择带 PROFIBUS 总线接口的 PLC：



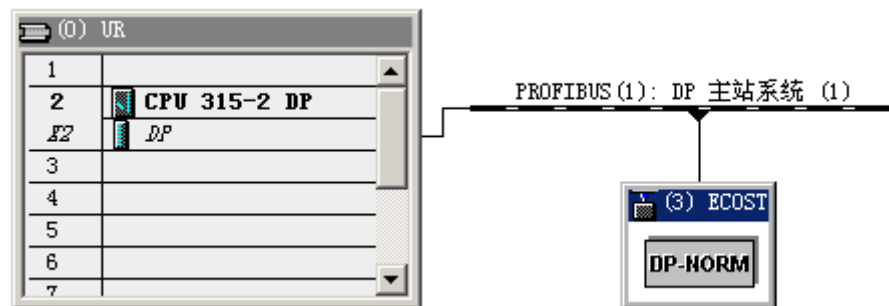
4. 设置 PROFIBUS 总线的通讯参数，ED 伺服支持最高 12M 的波特率，并且具有自适应功能，所以无需管伺服端波特率



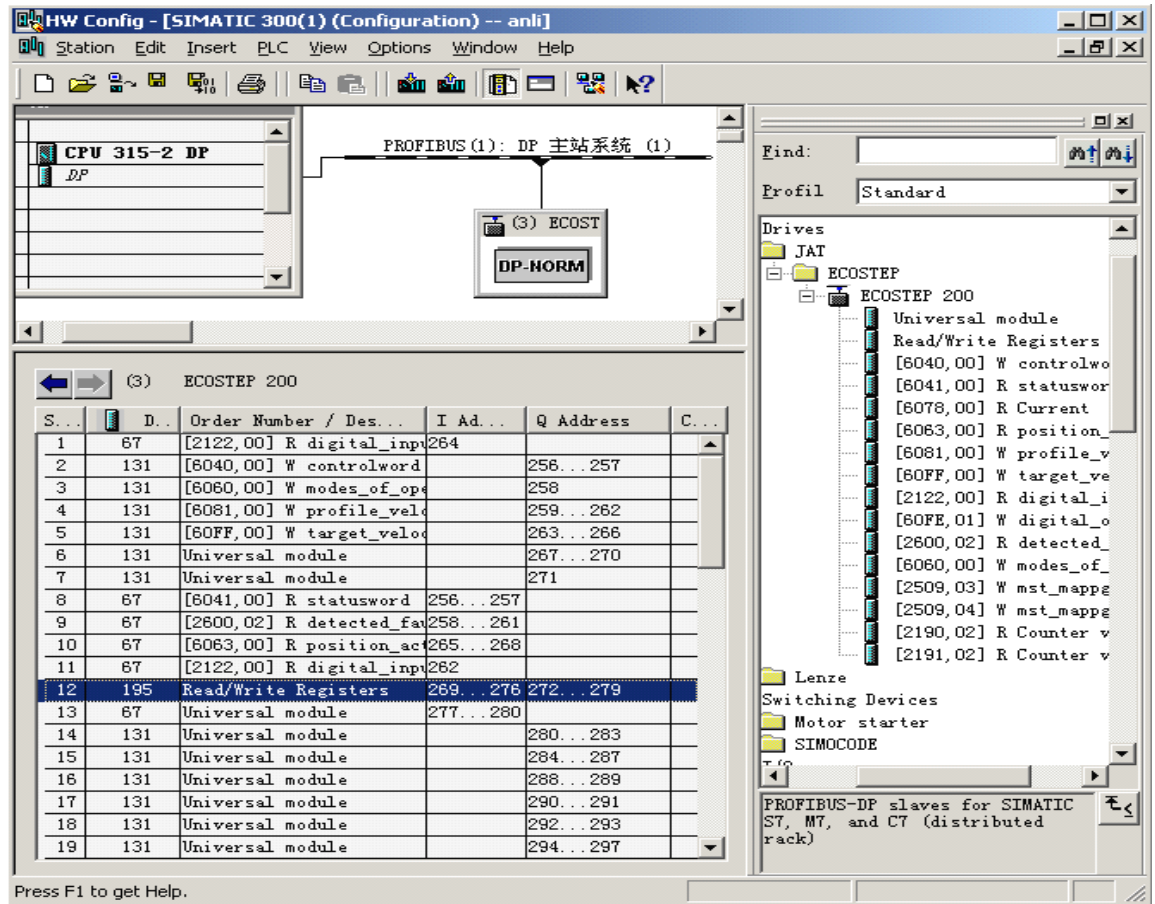
5. 找到 ED 伺服所在位置，然后进行组态：



6. 选中总线，拖入所要组态的 ECOSTEP 200 伺服，并设置站号，要与驱动器站号一致。



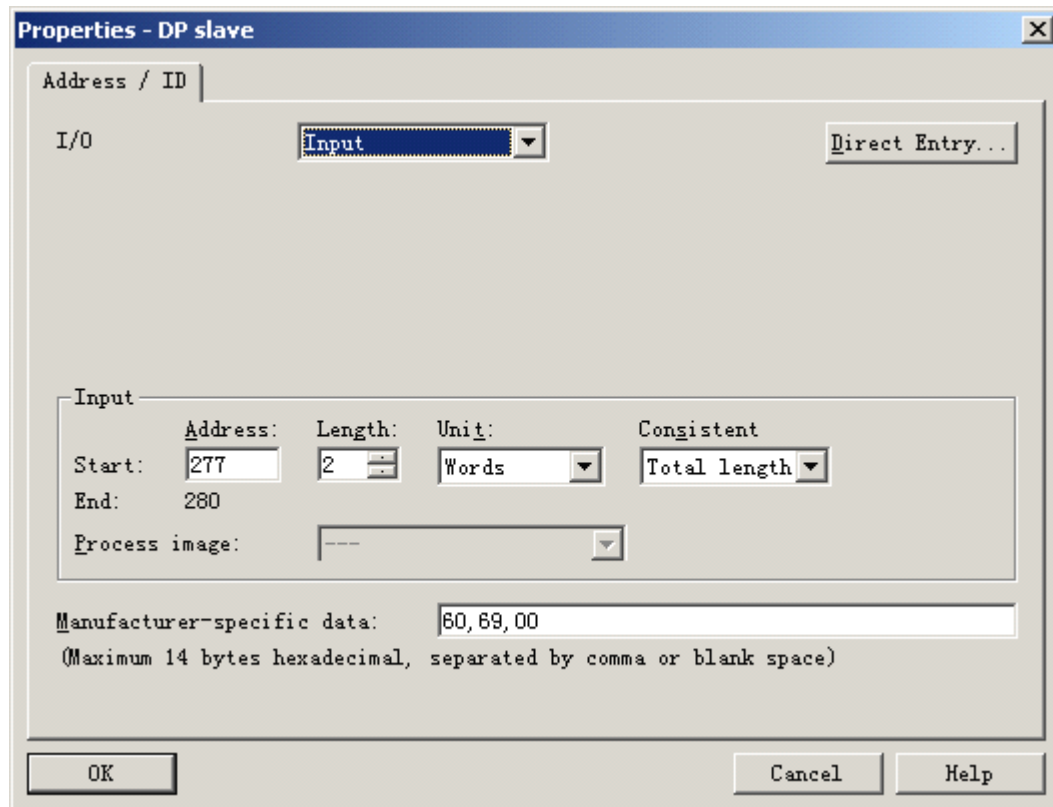
7. 然后对所要交换的数据进行组态, GSD 文件里所包含的对象如下图中红色处, 添加需要的对象。



8. 然后以类似的方式配置另外一个从站即可, 如果 GSD 文件里没有你所要想配置的 OD, 比如现在要配置 0x606900 (实际速度), 那么请添加 “Univesal module” 对象。

S...	D...	Order Number / Des...	I Ad...	Q Address	Comment
1	67	[2122,00] R digital_inpu	264		
2	131	[6040,00] W controlword		256...257	
3	131	[6060,00] W modes_of_ope		258	
4	131	[6081,00] W profile_velo		259...262	
5	131	[60FF,00] W target_veloc		263...266	
6	131	Universal module		267...270	
7	131	Universal module		271	
8	67	[6041,00] R statusword	256...257		
9	67	[2600,02] R detected_fax	258...261		
10	67	[6063,00] R position_act	265...268		
11	67	[2122,00] R digital_inpu	262		
12	195	Read/Write Registers	269...276	272...279	
13	67	Universal module	277...280		
14	131	Universal module		280...283	
15	131	Universal module		284...287	
16	131	Universal module		288...289	
17	131	Universal module		290...291	
18	131	Universal module		292...293	
19	131	Universal module		294...297	

9. 然后双击上图中蓝色字体部分, 弹出了 OD 配置对话框, 然后根据 OD 的属性来填写下面的各项即可:



I/O:

Input: OD 的属性为 readable

Output: OD 的属性为 writeable

out-input: OD 的属性为可读可写

empty-slot: 不配置任何对象

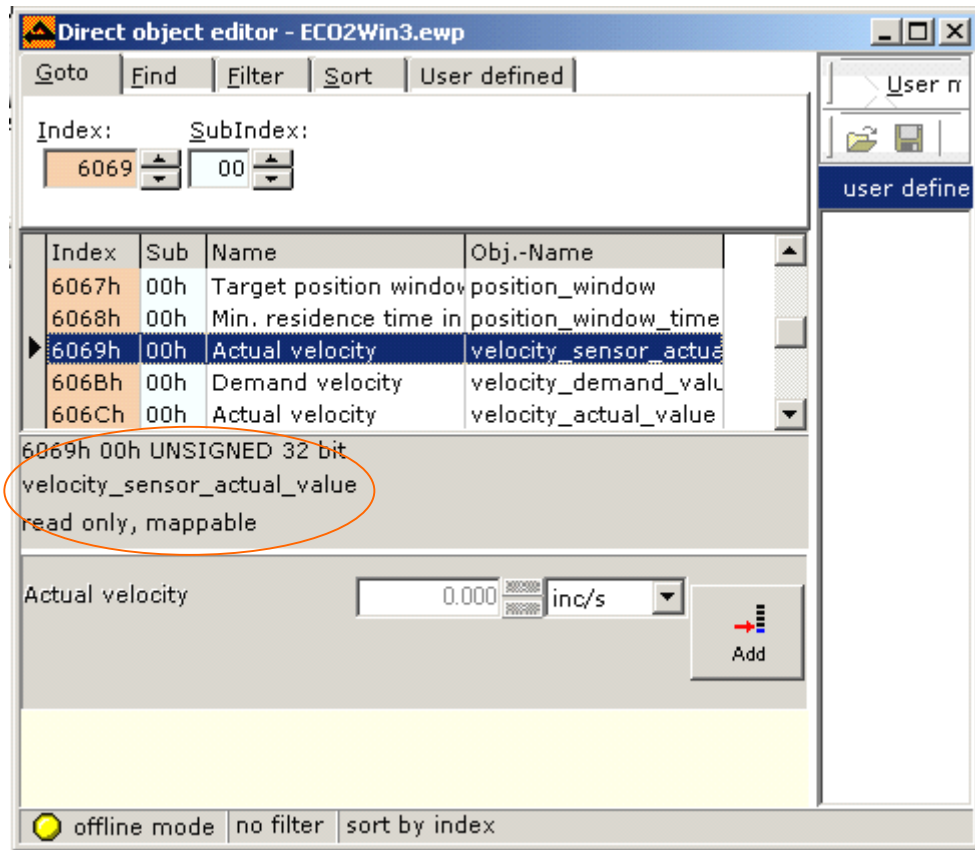
address: 针对不同的属性选择相应地址通道，对于 Output，请选择“Q Address”区域，对于 Input，请选择“I Address”区域。

Length: 对应 OD 的长度

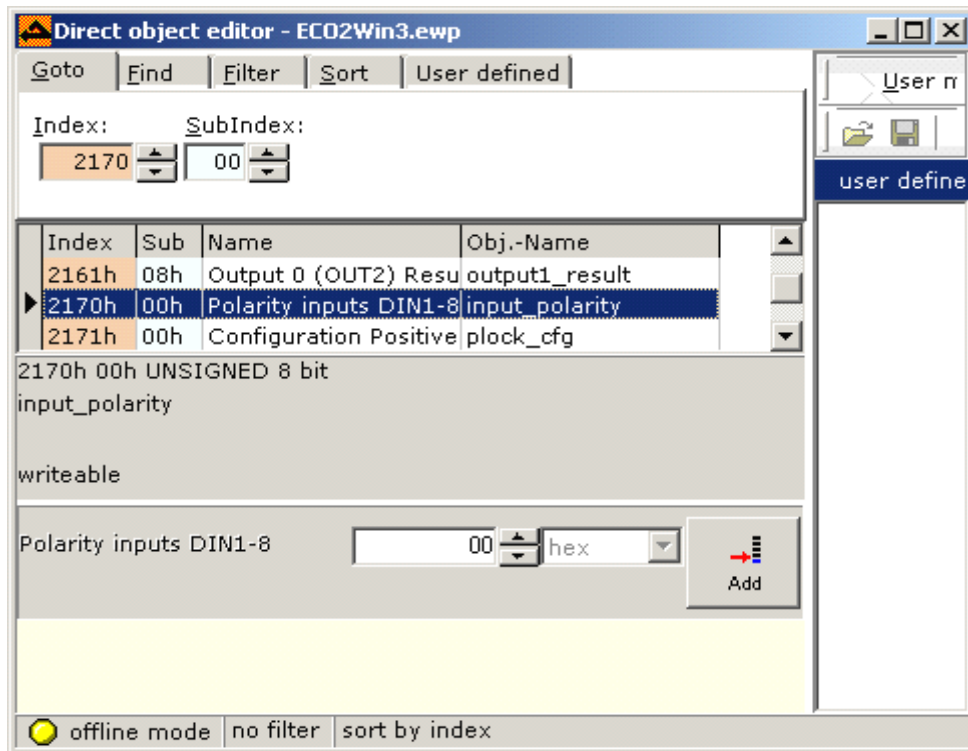
Unit: 对应前面 Length 所选择长度的单位

Consistent: 固定选择“Total Length”

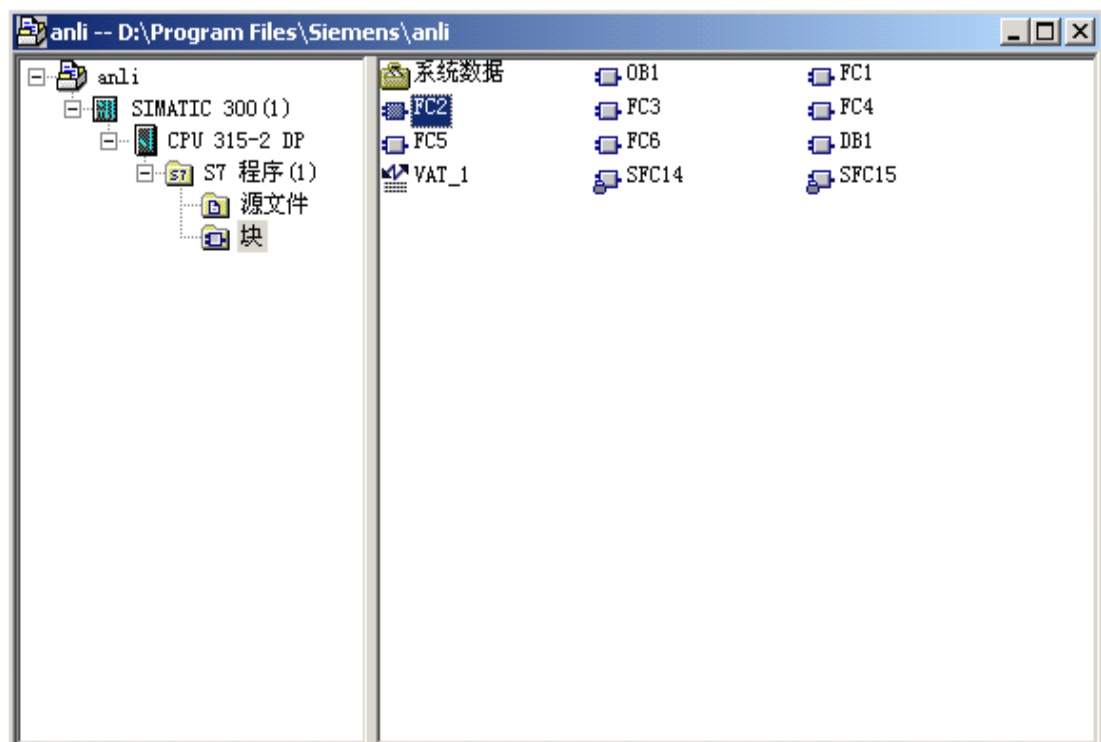
10. 配置完成后，点“OK”。我们可以在 EC02WIN/Direct object editor 中查询这些对象的属性，只要对象具有“mappable”属性，我们都可以采取添加“Univesal module”对象的方式来实现参数的读写。



11. 如步骤七图所示，配置完成后，我们就可以在 STEP 7 程序中引用这些寄存器了，例如，我们要使伺服上电，只需要往 PQW256 里写入 0xF 即可，如果要改变电机的模式为 1，那么只需要往 PQB258 里写入 0x1 即可。
12. ED 伺服中还有一些地址是不具有“mappable”属性的，如 0X217000 数字输入口极性，此时我们就不能采取添加“Univesal module”对象的方式来实现对该地址的写操作，添加后工程会报错。



13. 遇到上述情况我们可以在块里面添加系统功能块 SFC14、SFC15，通过调用 SFC14(DPRD_DAT)和 SFC15(DPWR_DAT)来实现该对象的读写。类似于 PLC 中自由口通讯，通讯报文由 8 个字节组成。



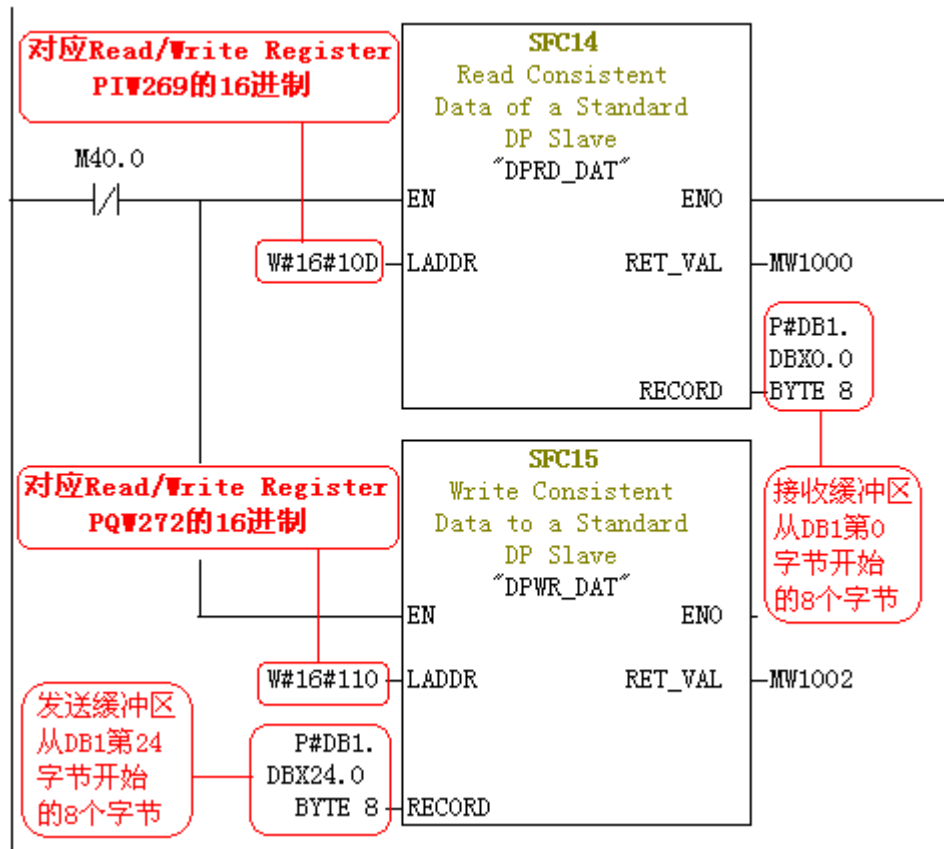
14. 要实现该功能我们首先要添加对象 Read/Write Register，如下图所示

S...	D...	Order Number / Des...	I Ad...	Q Address
1	67	[2122,00] R digital_inpu	264	
2	131	[6040,00] W controlword		256...257
3	131	[6060,00] W modes_of_ope		258
4	131	[6081,00] W profile_velo		259...262
5	131	[60FF,00] W target_veloc		263...266
6	131	Universal module		267...270
7	131	Universal module		271
8	67	[6041,00] R statusword	256...257	
9	67	[2600,02] R detected_fav	258...261	
10	67	[6063,00] R position_act	265...268	
11	67	[2122,00] R digital_inpu	262	
12	195	Read/Write Registers	269...276	272...279
13	67	Universal module	277...280	
14	131	Universal module		280...283
15	131	Universal module		284...287
16	131	Universal module		288...289
17	131	Universal module		290...291
18	131	Universal module		292...293
19	131	Universal module		294...297

15. 然后在主程序添加一个《=16byte 的 DB 数据块，用于发送或接受数据缓冲区。

Address	Name	Type	Initial value	Comment
0.0		STRUCT		
+0.0	RO_BYTE0	BYTE	B#16#0	
+1.0	RO_BYTE1	BYTE	B#16#0	
+2.0	RO_BYTE2	BYTE	B#16#0	
+3.0	RO_BYTE3	BYTE	B#16#0	
+4.0	RO_BYTE4	BYTE	B#16#0	
+5.0	RO_BYTE5	BYTE	B#16#0	
+6.0	RO_BYTE6	BYTE	B#16#0	
+7.0	RO_BYTE7	BYTE	B#16#0	

16. 在 OB 块中调用 SFC14（接收功能块）和 SFC15（发送功能块），SFC14 功能块中 LADDR 为 W#16#10D，实际就是对象 Read/Write Register PIW 的起地址，DB1.DBB0 开始的 8 个字节存储的是接收到得数据。SFC15 功能块中 LADDR 为 W#16#110，实际就是对象 Read/Write Register PQW 的起地址。DB1.DBB24 开始的 8 个字节存储的是当前要发送的，M40.0 为使能。程序如下图所示。



举例说明：

1) 主站向从站地址 INDEX6073, 00 设定最大电流 1023 (HEX 03FF)

请求报文: 0x60,0x73,0x00,0x20,0x00,0x00,0x03,0xFF

回应报文: 0x60,0x73,0x00,0x60,0x00,0x00,0x03,0xFF

发送的请求报文

DB1.DBB24=0X60

DB1.DBB25=0x73

DB1.DBB26=0x00

DB1.DBB27=0x20

DB1.DBB28=0x00

DB1.DBB29=0x00

DB1.DBB30=0x03

DB1.DBB31=0xFF

接收的应答报文

DB1.DBB0=0x60

DB1.DBB1=0x73

DB1.DBB2=0x00

DB1.DBB3=0x60

DB1.DBB4=0x00

DB1.DBB5=0x00

DB1.DBB6=0x03

DB1.DBB7=0xFF

2) 主站向从站请求调用子程序 SEQ02

请求报文: 0x21,0x18,0x00,0x20,0x00,0x00,0x00,0x02

回应报文: 0x21,0x18,0x00,0x6C,0x00,0x00,0x00,0x02

发送的请求报文

DB1.DBB24=0X21

DB1.DBB25=0x18

接收的应答报文

DB1.DBB0=0x21

DB1.DBB1=0x18

DB1.DBB26=0x00	DB1.DBB2=0x00
DB1.DBB27=0x20	DB1.DBB3=0x6c
DB1.DBB28=0x00	DB1.DBB4=0x00
DB1.DBB29=0x00	DB1.DBB5=0x00
DB1.DBB30=0x00	DB1.DBB6=0x00
DB1.DBB31=0x02	DB1.DBB7=0x02

3) 主站向从站请求将 DIN6, DIN7 的极性由常开改为常闭

请求报文: 0x21,0x70,0x00,0x20,0x00,0x00,0x00,0x60

回应报文: 0x21,0x70,0x00,0x60,0x00,0x00,0x00,0x60

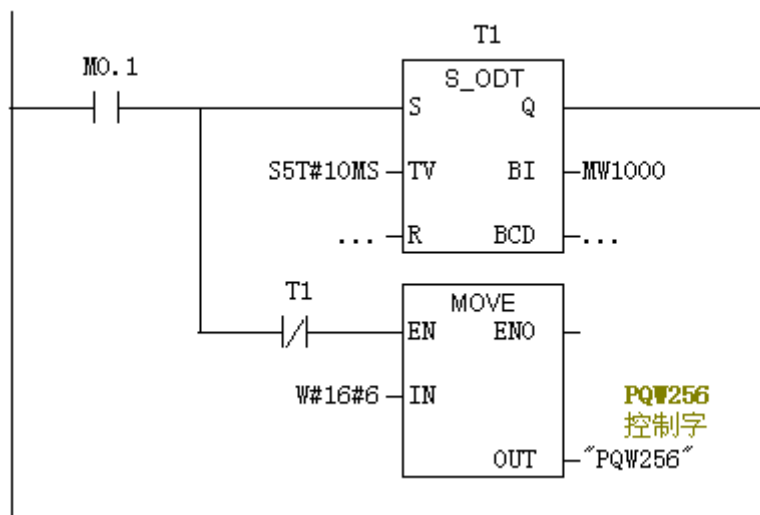
发送的请求报文

接收的应答报文

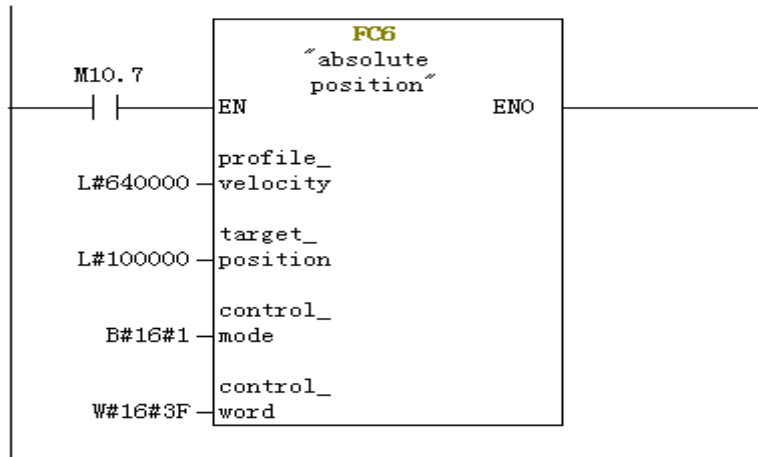
DB1.DBB24=0x21	DB1.DBB0=0x21
DB1.DBB25=0x70	DB1.DBB1=0x70
DB1.DBB26=0x00	DB1.DBB2=0x00
DB1.DBB27=0x20	DB1.DBB3=0x60
DB1.DBB28=0x00	DB1.DBB4=0x00
DB1.DBB29=0x00	DB1.DBB5=0x00
DB1.DBB30=0x00	DB1.DBB6=0x00
DB1.DBB31=0x60	DB1.DBB7=0x60

17. 通过定时器周期性改变发送缓冲区 DB 数据块中的数据, 同时调用 SFC14, SFC15 就可以读写数据了。这种方式适合读写不经常改动的数据。

18. 配置好所有需要的对象后, 就可以开始编程了, 注意在刚上电总线初始化完成后, PLC 需要先向 ED 伺服驱动器控制字 PQW256 赋 W#16#6, 然后再进行其他操作, 否则驱动器将对其他指令无响应



19. 可以把伺服的各个功能做成功能块, 如绝对定位、原点回归、点动运行等, 然后在主程序中调用赋值, 下图是 ED 绝对定位程序。范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服-范例程序” 下载。

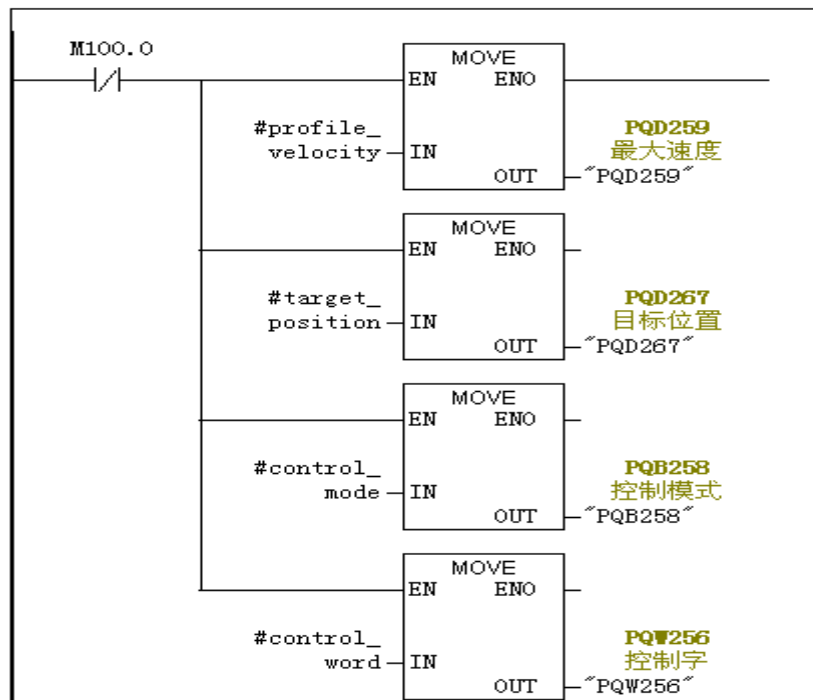


FC6 : Title:

Comment:

Network 1: Title:

Comment:



三、通讯报警故障处理

1: 软件 BLOCK 功能块中必须要有 OB82,OB86,OB100,OB121,OB122.否则通讯故障总线断开后,重启不会自动恢复。详情请参考 STEP 7 手册

2: 程序中使用 FB125 程序来判断系统中的错误,FB125 可以诊断总线上站点的各种故障信息。FB125 是一个中断驱动功能块,它可以检测到从站的组态,掉站等故障,并且给出详细的报错信息。使用 FB125 来诊断故障必须在 OB1, OB82 和 OB86 中调用。详情请参考 STEP 7 手册

二：ED 伺服与施耐德 M258 利用 CANopen 总线进行通讯

一、硬件连线

主站	从站 1	从站 2	-----从站 N
M258 CANopen 口	ED X1 口	ED X1 口	
2 CAN_L	2	2	
7 CAN_H	7	7	

注意！ 总线首位两端在 2 针和 7 针间连接 120—150 欧姆的电阻。注意 ED100 需要外接电源。

二、设定 Kinco 伺服的 ID 地址和波特率

设置 Kinco 伺服的 ID 地址，伺服驱动器上拨码设置为 1（0—15）。如果超过 15 个节点，在 Eco2win 软件中设置 2F800010，指定伺服的 ID 偏移地址（0—127），如 1，则该 Kinco 伺服的实际地址为 ID+1。

设置 Kinco 伺服的 CAN 通讯波特率参数，地址分别为 2F810010 和 2F820010，根据 Kinco 伺服的操作手册，设定一个相应的波特率，如 2F810010 设为 0x00，2F820010 设为 0x14，既波特率为 1Mbps/s；如果 2F810010=0x00，2F820010=0x1c，波特率=500kbit/s。

在设置了上述参数后，在 system manager 中设置 CX1000 的 CANopen 模块的通讯参数相同。

三、编写 PLC 程序（范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服- 范例程序” 下载）

一、 硬件配置

- （1）添加 EDS 文件。
- （2）配置好站号，波特率等参数。

具体操作如下图：

1.创建新项目



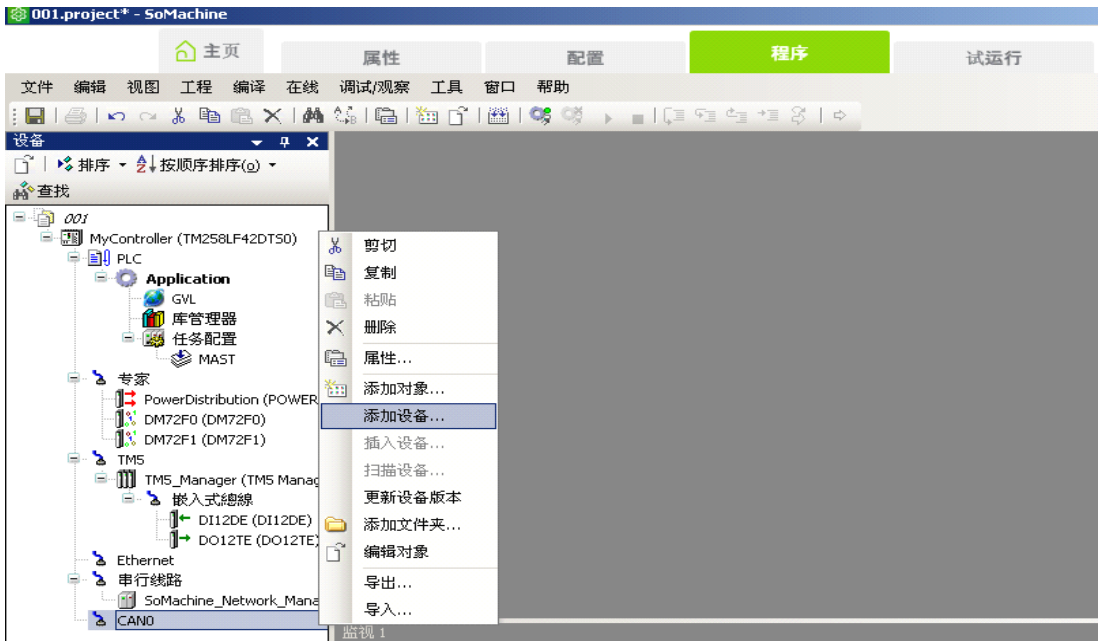
2.保存文件为 001



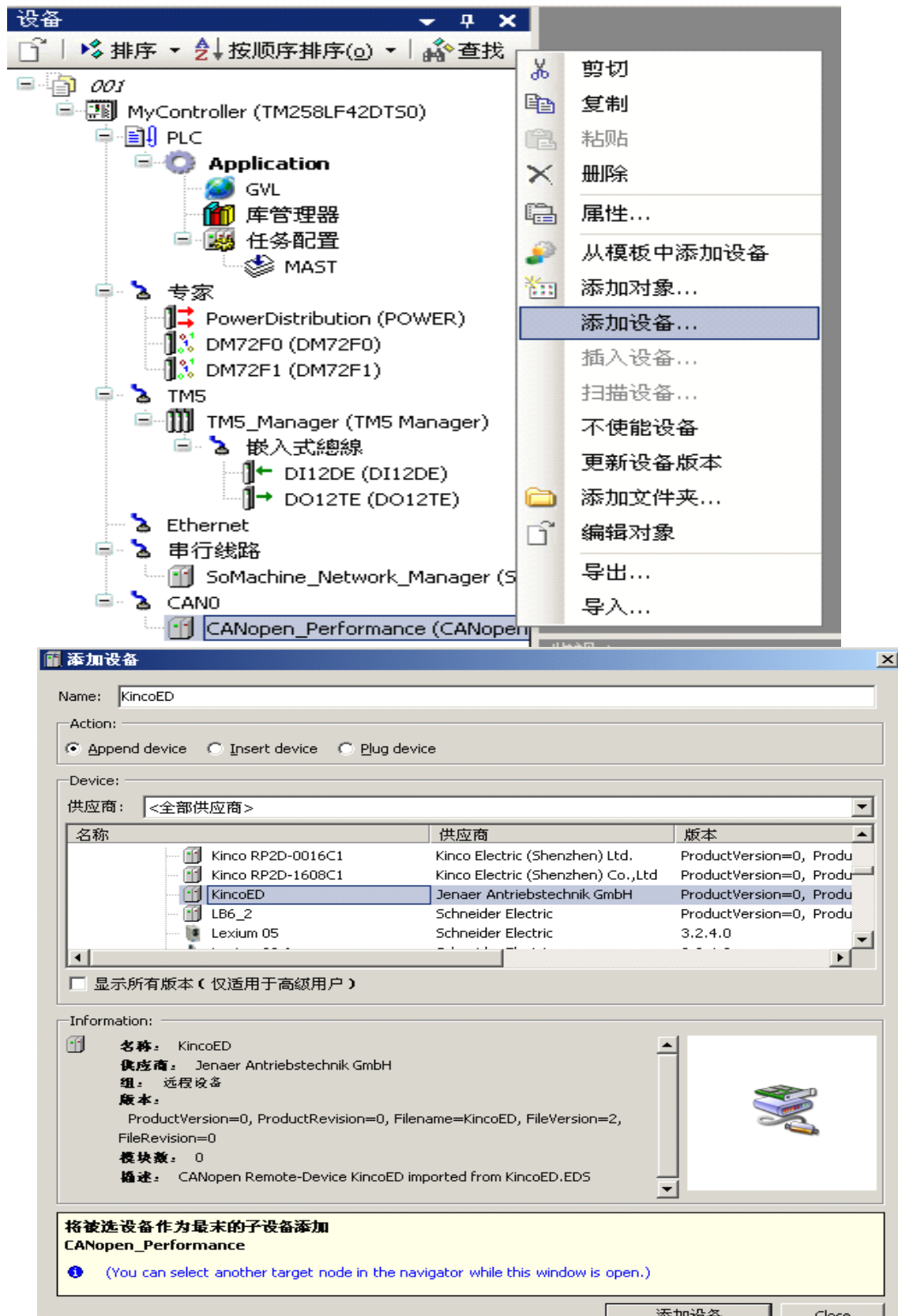
3.配置 PLC



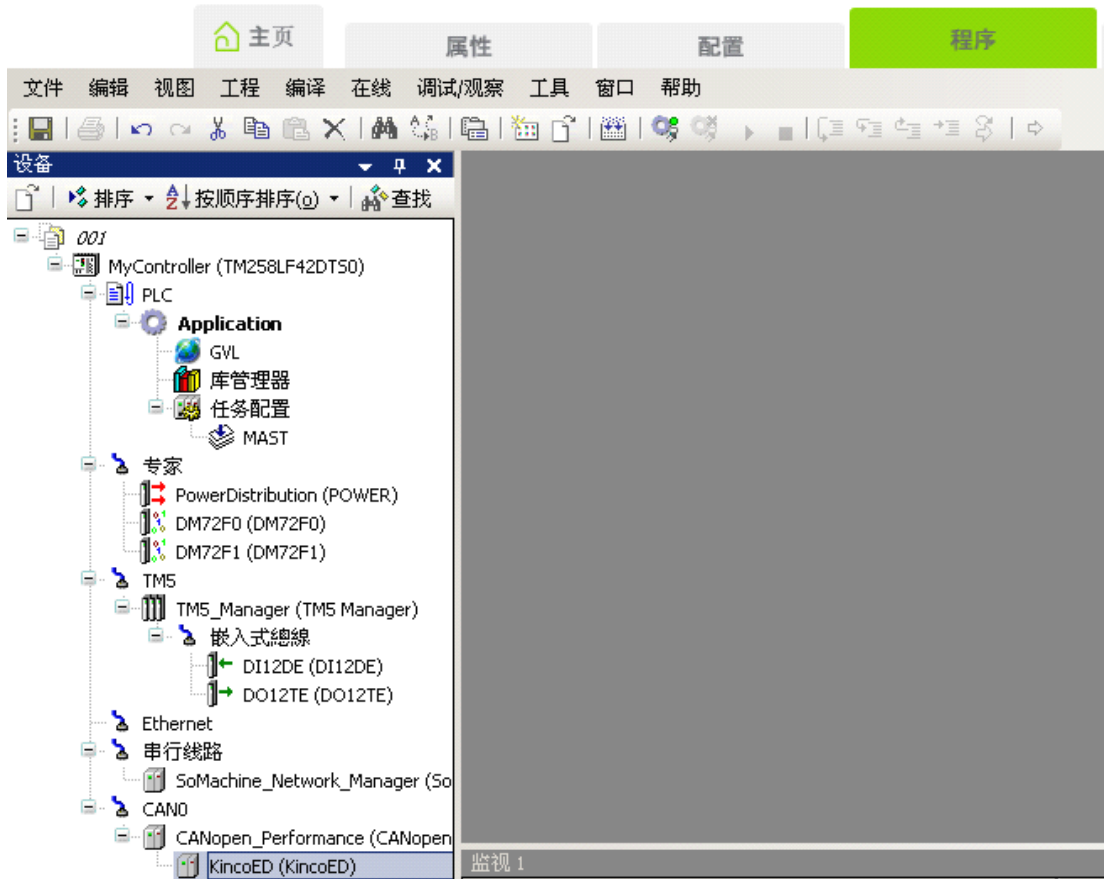
4.添加 PLC 的 CAN 驱动



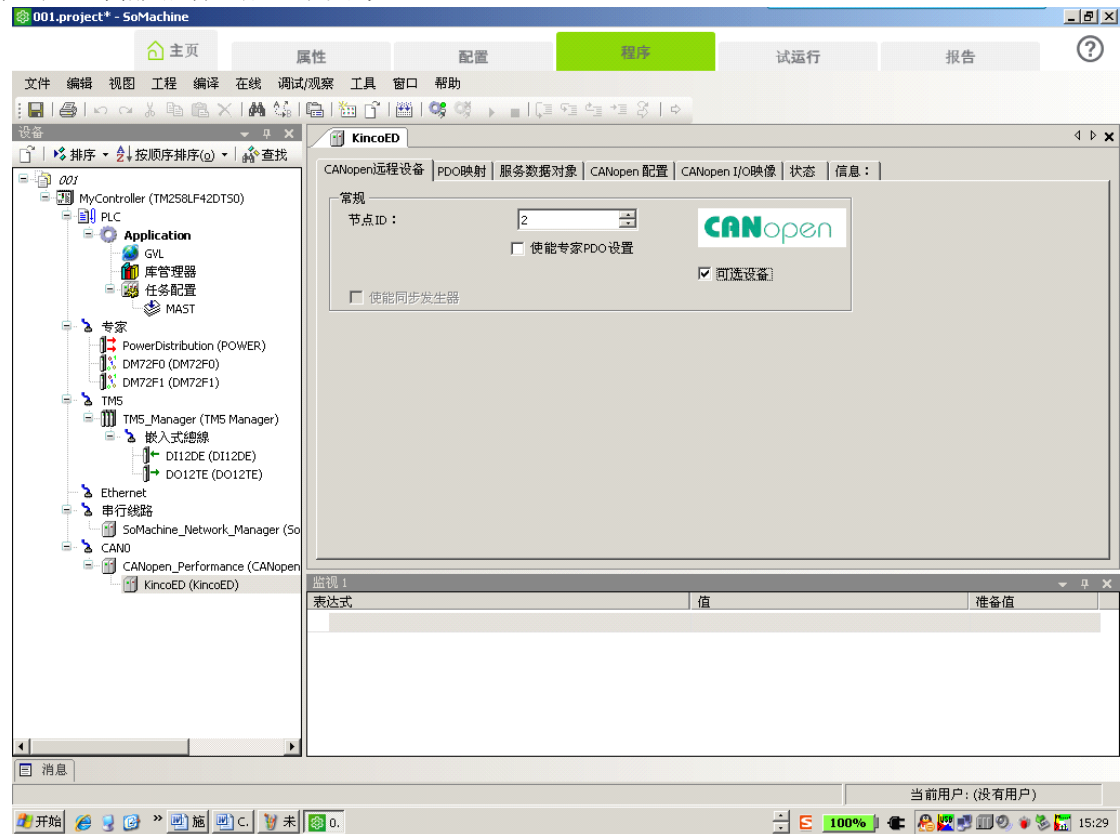
5. 添加 EDS 文件



6. 添加完成后，ED 伺服驱动会显示在左边栏。



7. 双击 ED 伺服驱动，配置节点号。



8. 双击配置里的 PLC，配置主站波特率。与 ED 伺服端要一致

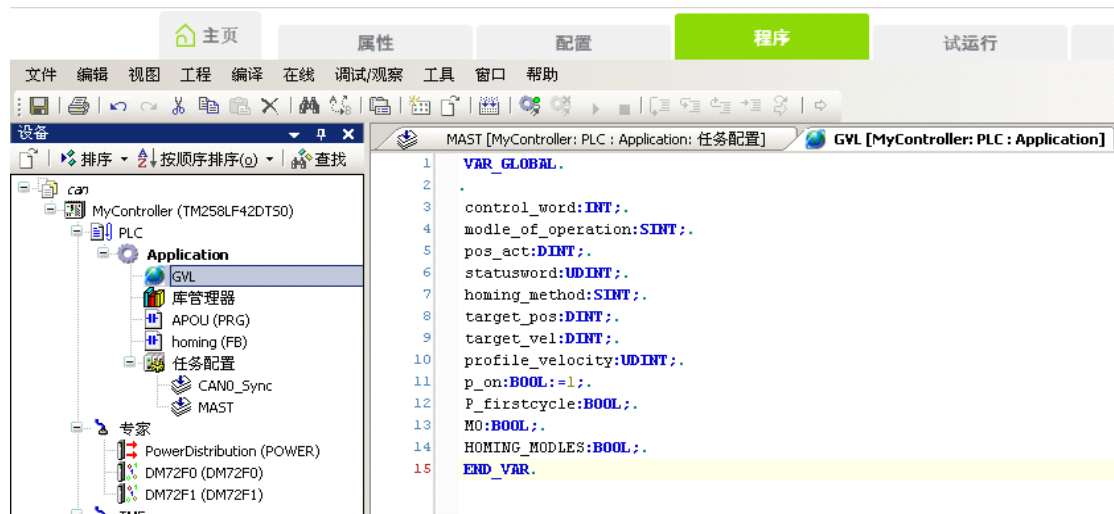


9.配置想要的目标对象，例如下图配置控制字。

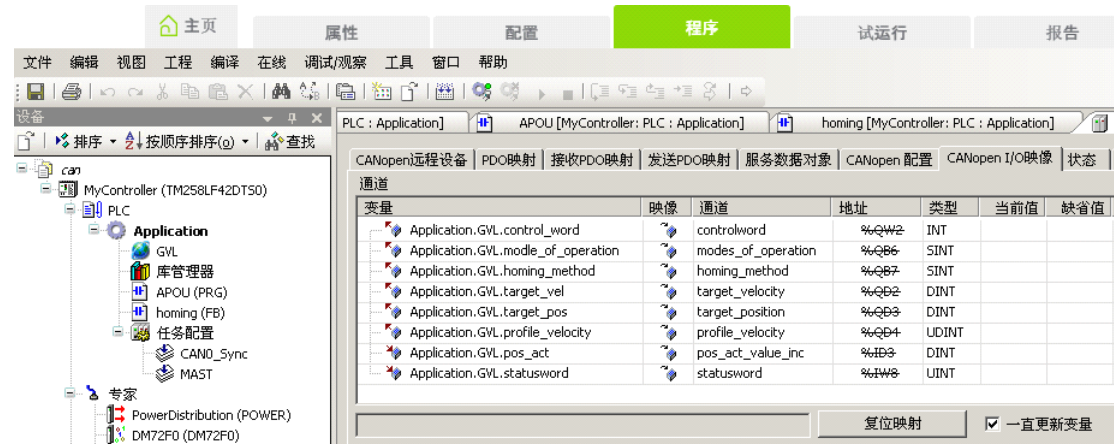


二. 编写程序。

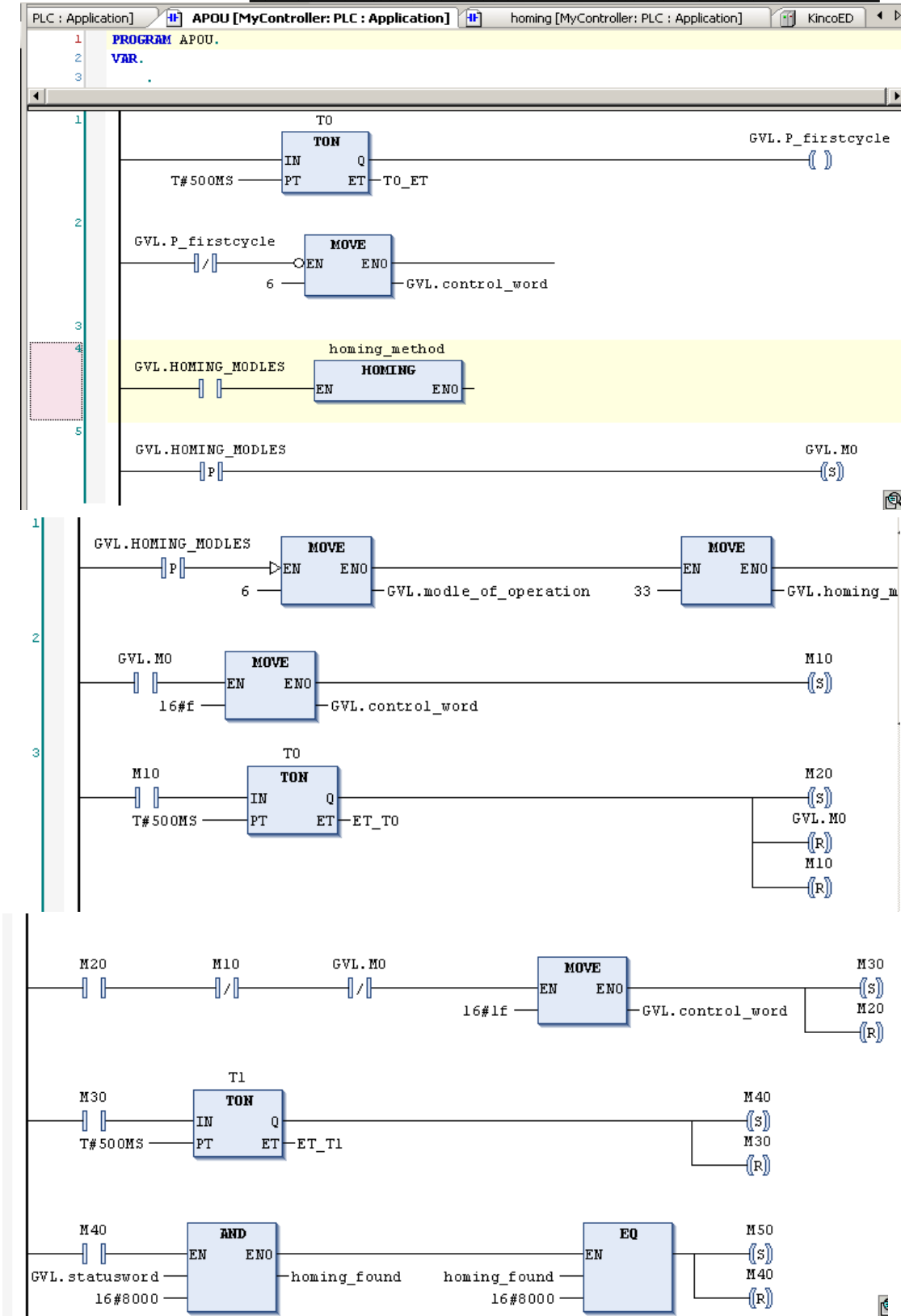
1.设置全局变量。



2.将全局变量，对应 ED 里面地址。



3.编写程序。注意上电时首先需要初始化伺服控制字为 6，否则伺服对其他指令无响应。



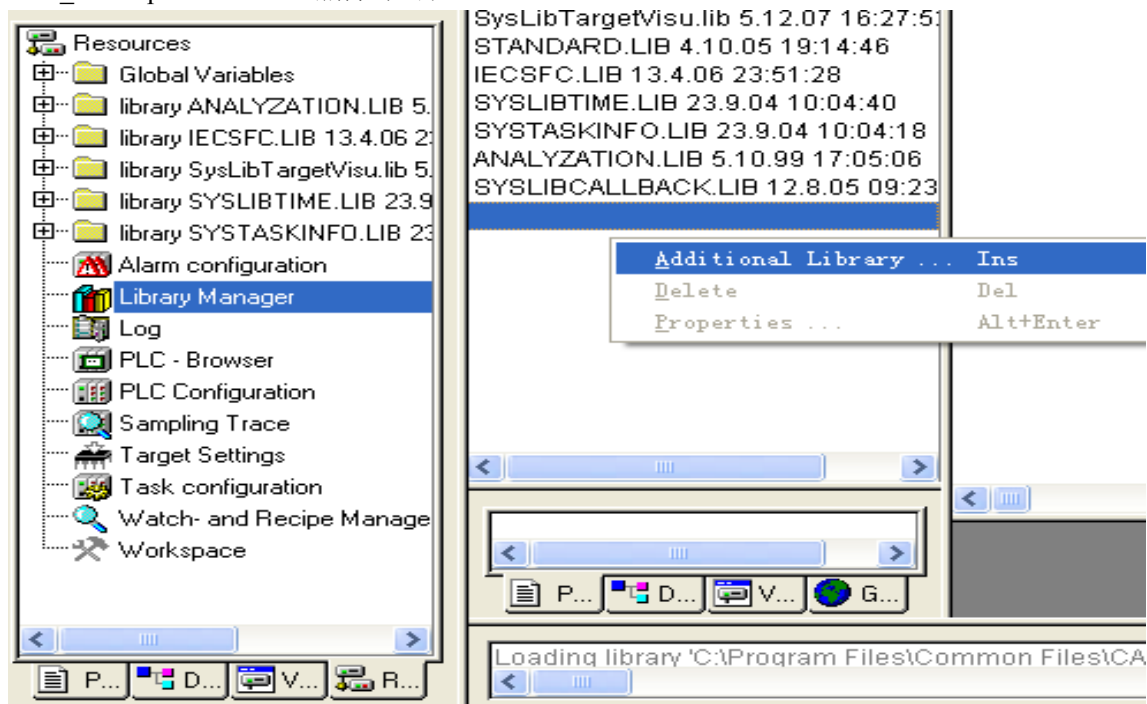
图示为自动找原点程序，速度和位置模式程序同上，可试着自行编写。

三：ED 伺服与 Codesys 利用 CANopen 总线进行通讯

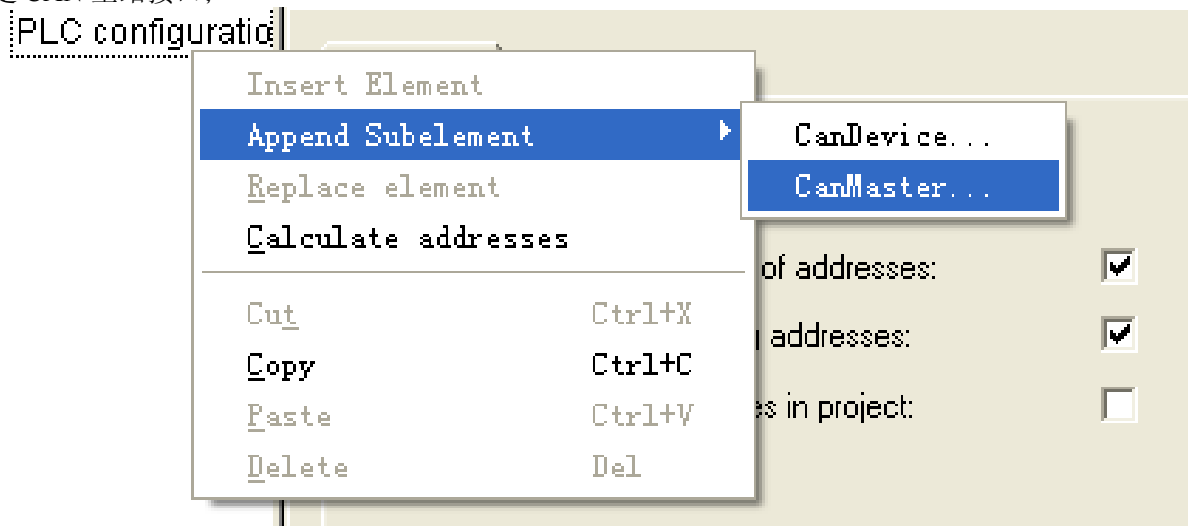
一：硬件接线。由于可以运行 Codesys 硬件平台有很多，请参考各硬件平台管脚定义接线

二：软件编程（范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服-范例程序” 下载）

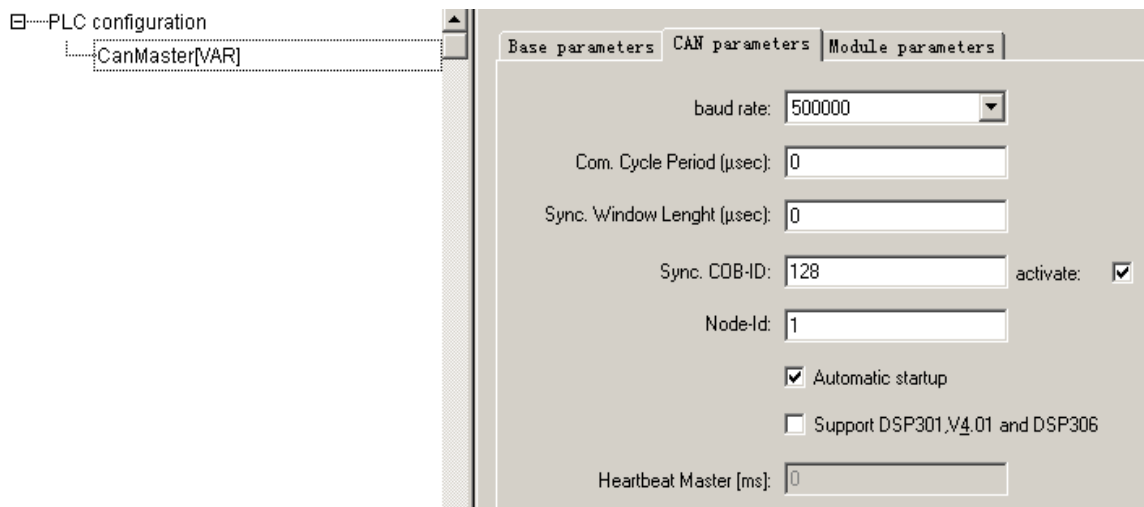
1：首先添加 CANopen 相关库函数，点击“Library manager”，然后在库函数对话框空白处点击右键选择“Additional library”，然后在弹出的对话框中选择“3S_CANopenDevice.lib”，然后再添加“3S_CANopenMaster.lib”，最后再把“3S_CANopenDevice.lib”删除即可；



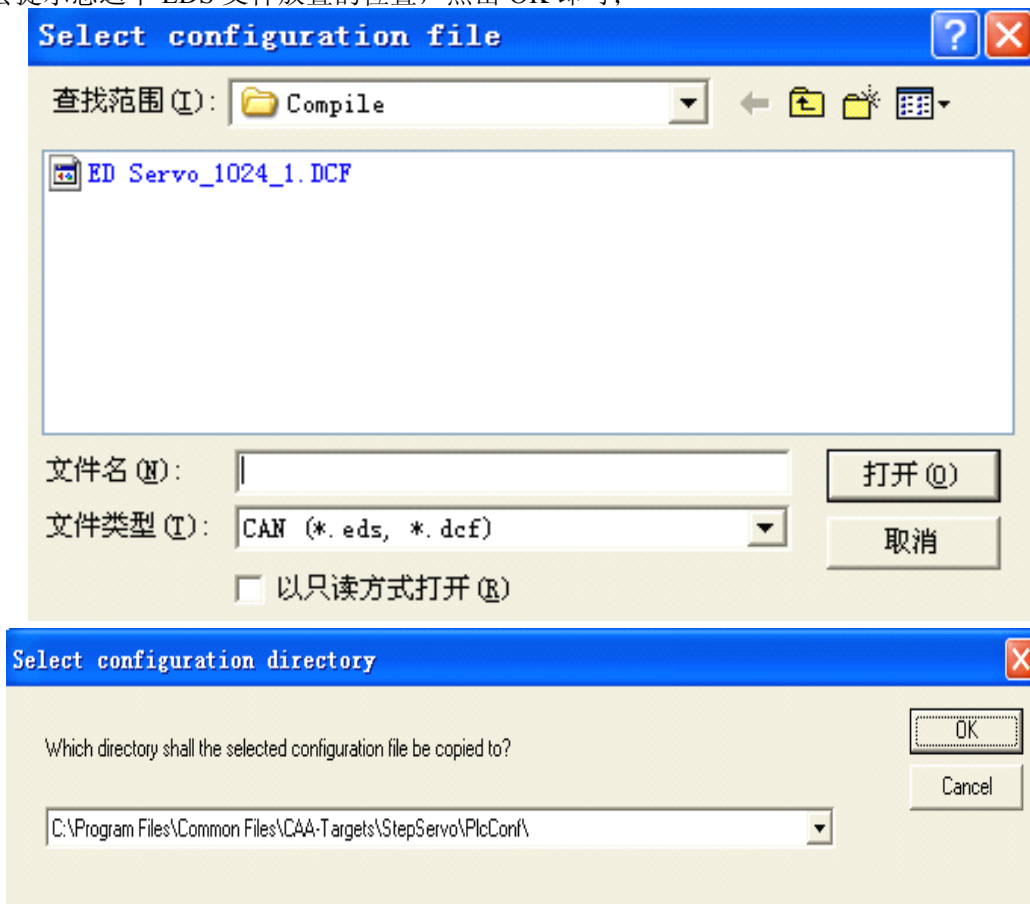
2. 选中“PLC Configuration”，然后点击鼠标右键，选择“Append subelement”，然后在选择“CanMaster”：这里添加的是 CAN 主站接口；



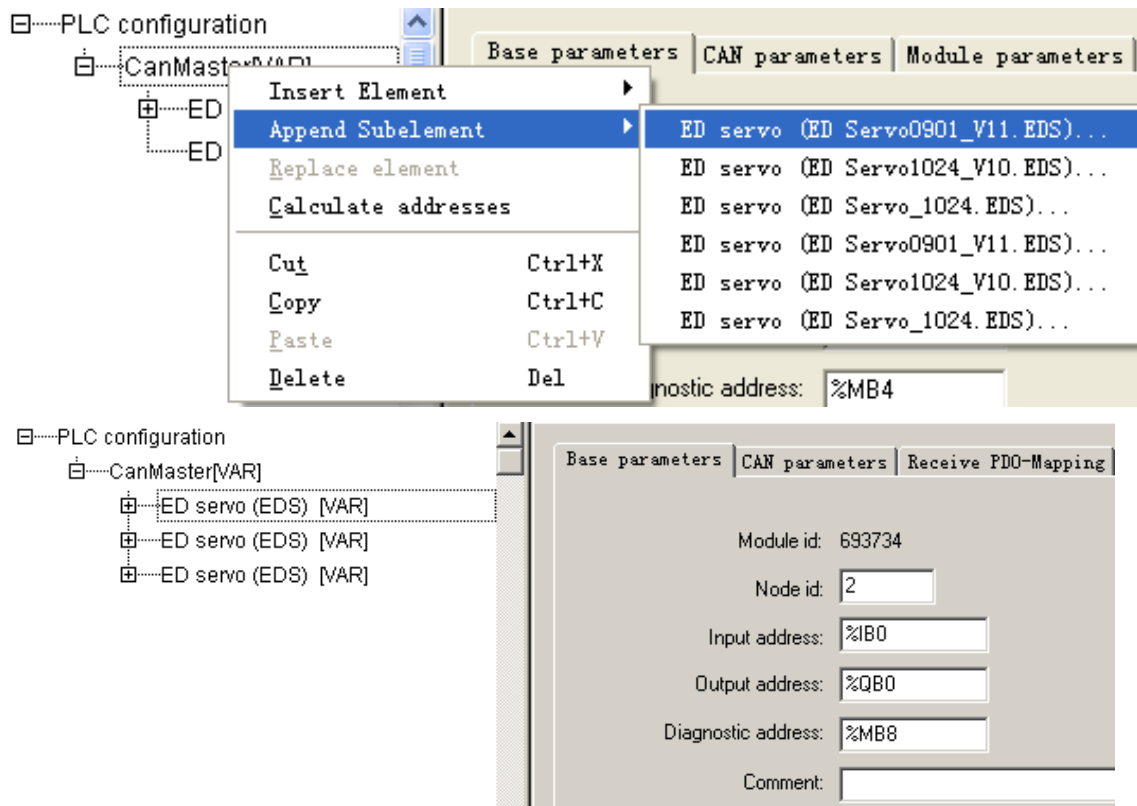
3. 然后进入“CAN parameter”项里设置总线通讯的波特率，如果需要同步报文，请先在激活“active”，然后在“Com. Cycle period”里设置同步报文的传输周期和 COB-ID；



4. 然后我们需要添加 CANopen 从站，首先我们需要添加从站的 EDS 文件到 Codesys 里面来，点击主菜单里的“Extras”里的“add configuration file”选项，然后弹出一个对话框，然后找 ED 伺服的 EDS，点击打开即可，然后系统会提示您这个 EDS 文件放置的位置，点击 OK 即可；

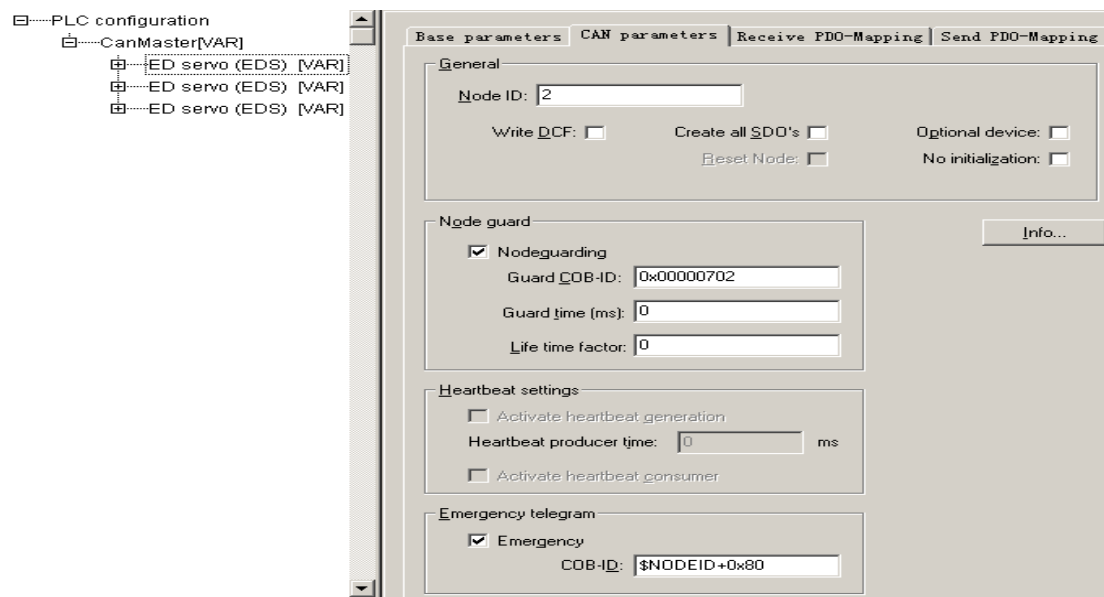


5. 然后在选中“CanMaster”并点击右键，选择“Append ED servo”，系统就会把 ED 伺服添加到网络中了；按照这种方式添加您所想要添加的从站个数即可；



6. 添加完从站后，我们要一一设置每个从站的通讯起始地址、站号、节电保护类型、RX-PDO、TX-PDO 等参数。首先设置“Base parameter”里的参数；根据实际伺服的站号设置“Node id”，其它几个参数不需修改；

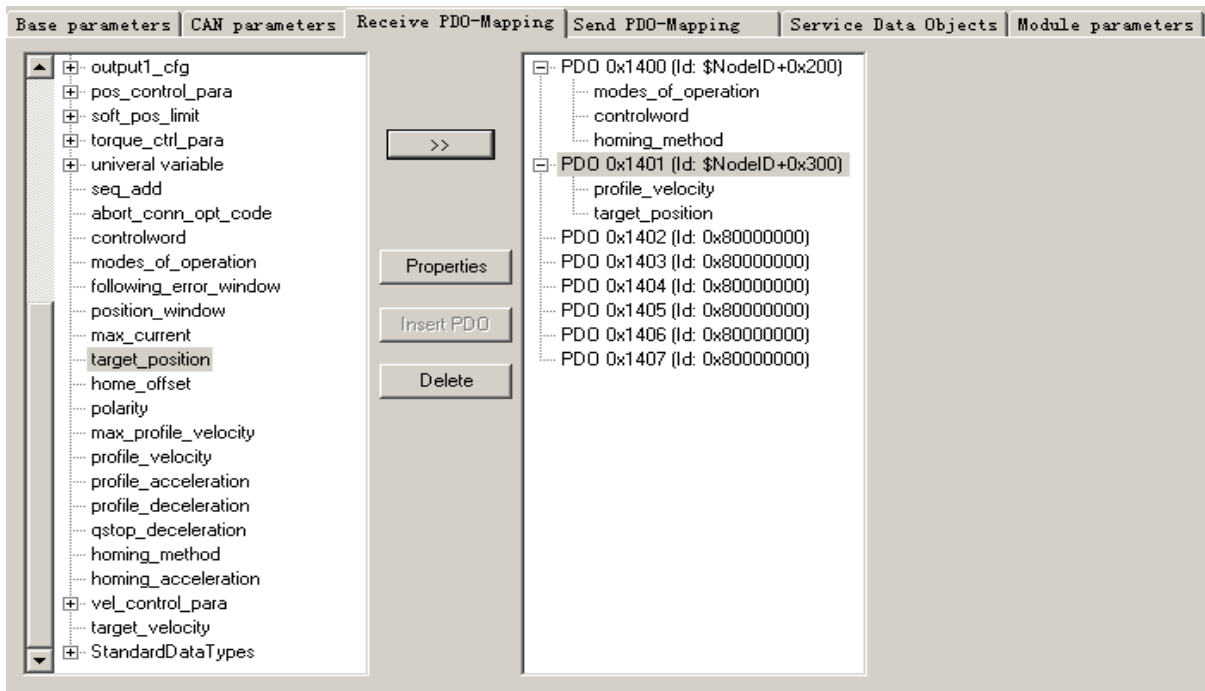
7. 进入“CAN parameters”，设置“Node ID”为伺服的站号，进入“Node guard”里设置节点保护的时间，然后在“Emergency telegram”里设置紧急报文为“\$NODEID+0x80”，其它不需修改



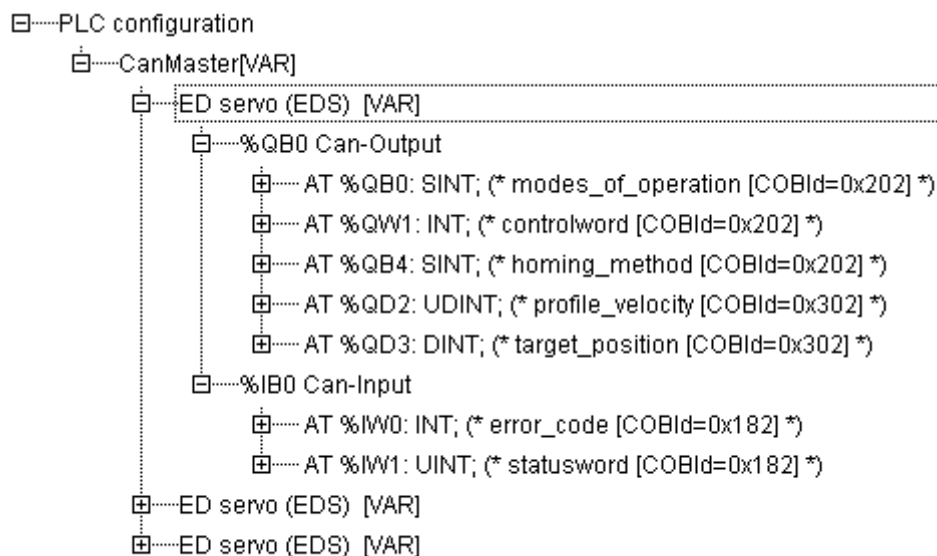
8. 进入“Receive PDO-Mapping”对话框，然后设置每个 PDO 里所映射的内容；设置方法如下：

- ▶ 选中所要映射的 PDO，这里我们选择 PDO1
- ▶ 选中所要映射到 PDO1 里的对象，然后点击“>>”；
- ▶ 按照同样的方式映射其它 PDO 即可；

按照同样的方法设置 TX-PDO 即可；

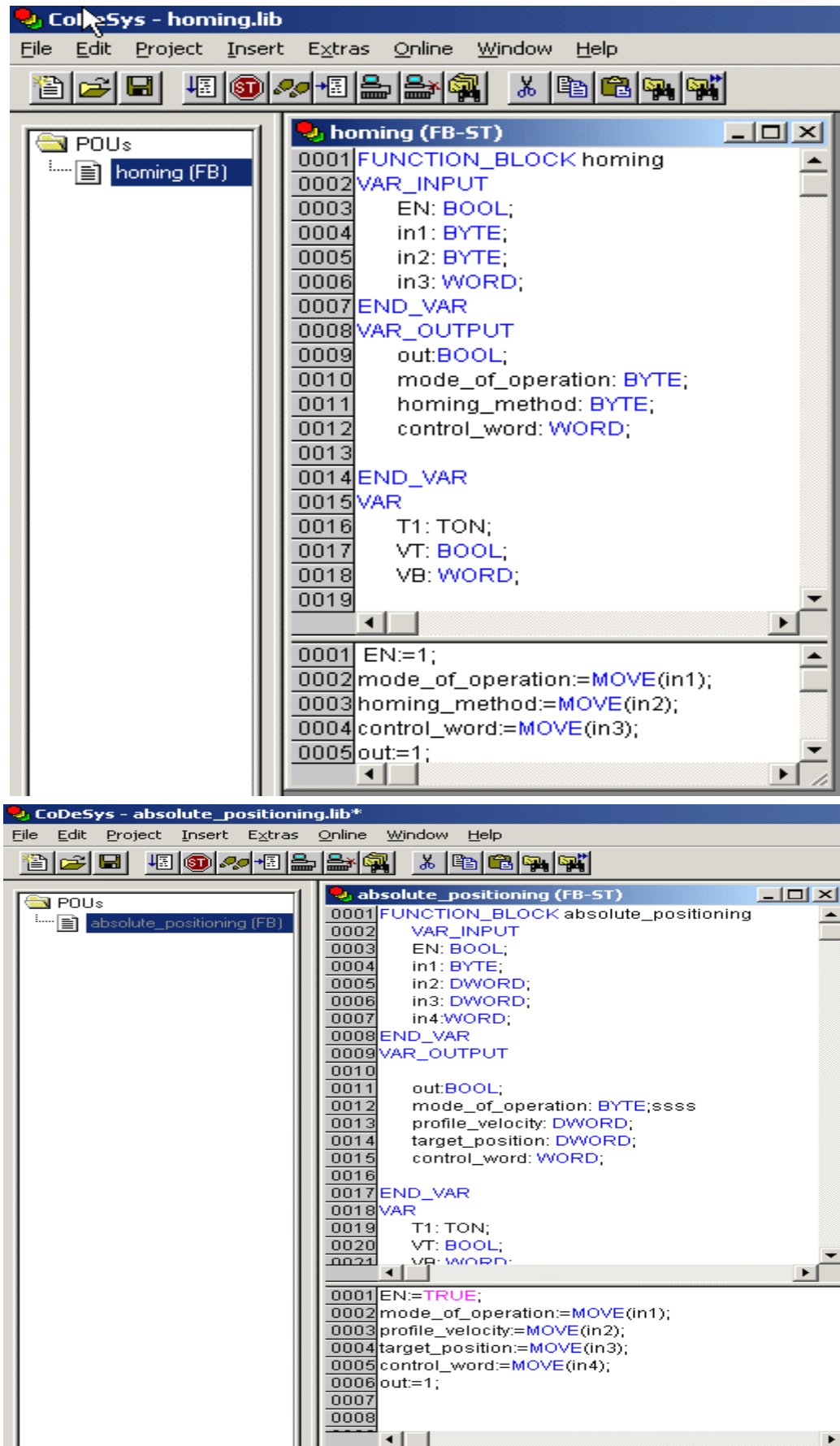


9. 设置完成所有的伺服后，我们点击硬件配置里的树型结构，就可以看到映射到每个 PDO 里的 OD 了，同时还能看到每个 OD 对应的输入和输出寄存器，从右图可以看到，ID1 伺服的 Controlword 对应的寄存器为 QW1，而 error_code 对应的寄存器为 IW0，通过在软件里控制这些寄存器就可以控制伺服了；

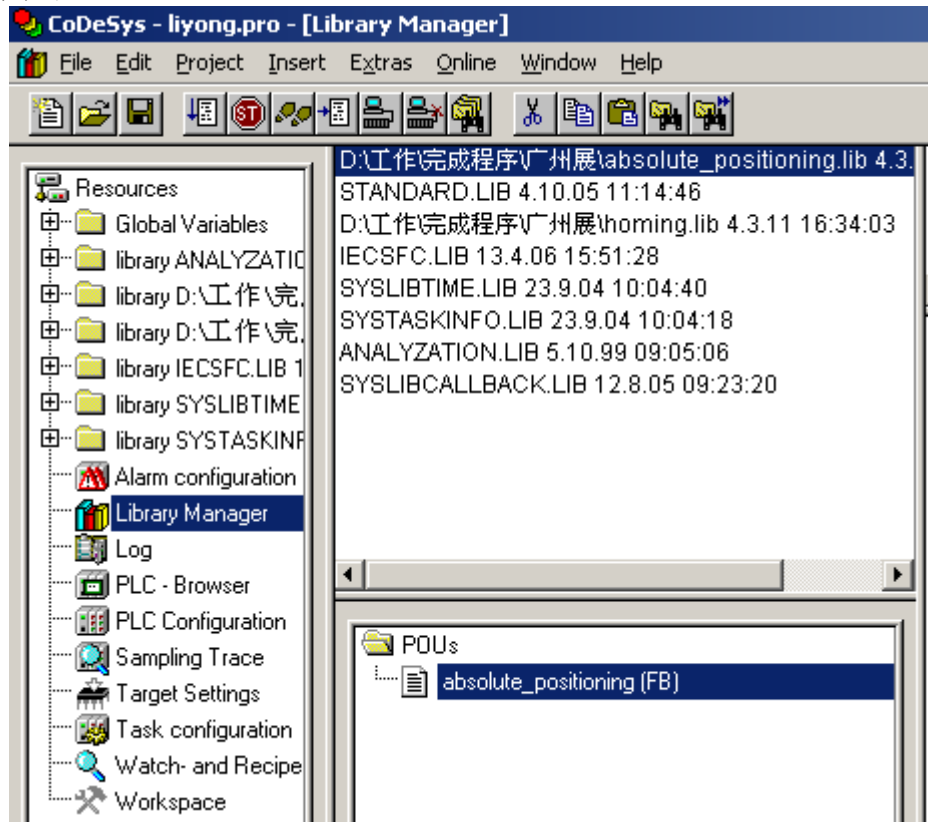


10. 配置好所有参数后就可以开始编程了。我们可以将 ED 伺服的各个功能做成库文件，如下图所示原点

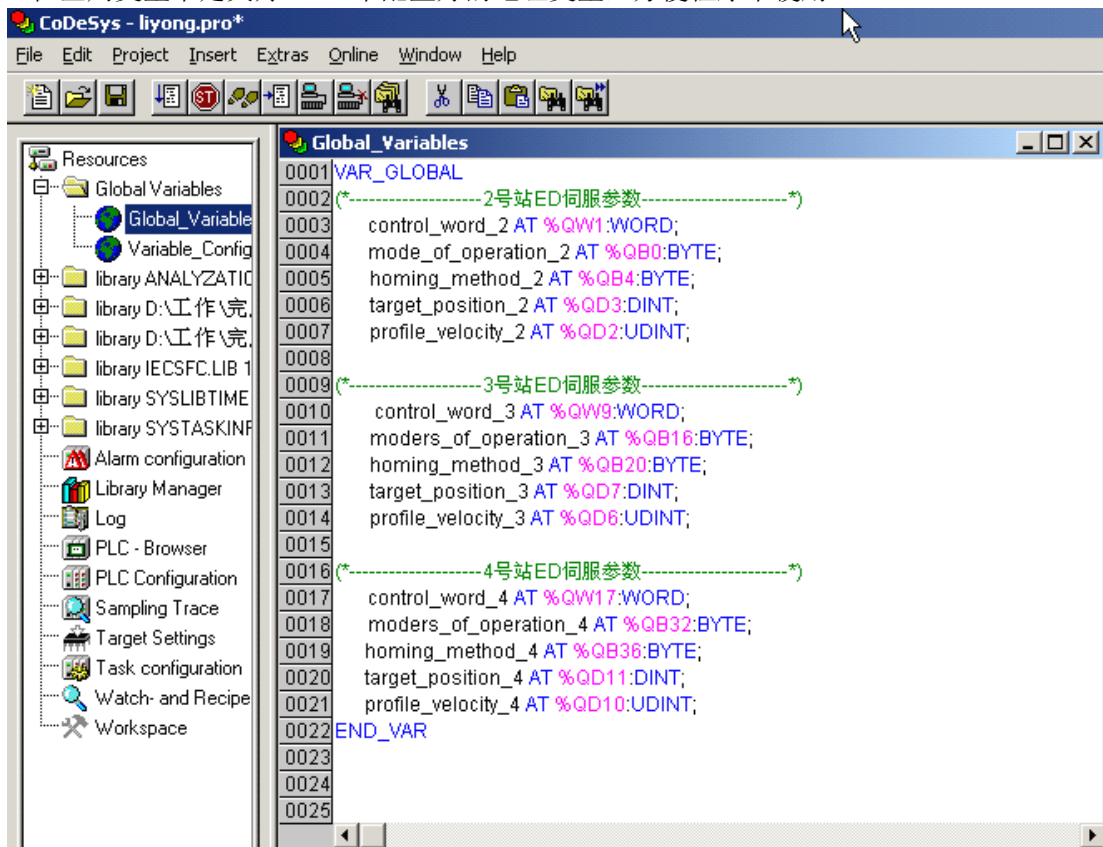
库、绝对定位库。



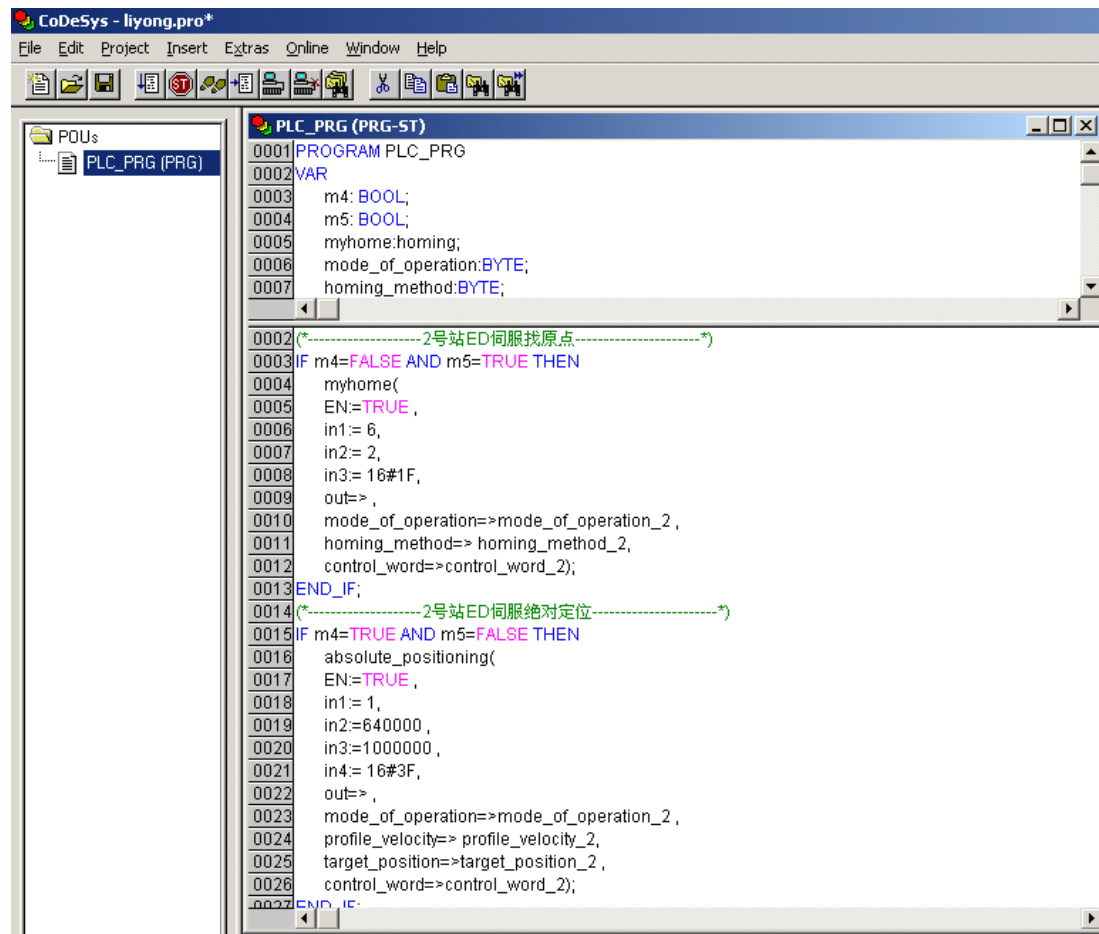
11. 在编程使用时先将原点库文件及绝对定位库文件添加到工程中，需要使用时直接调用即可。



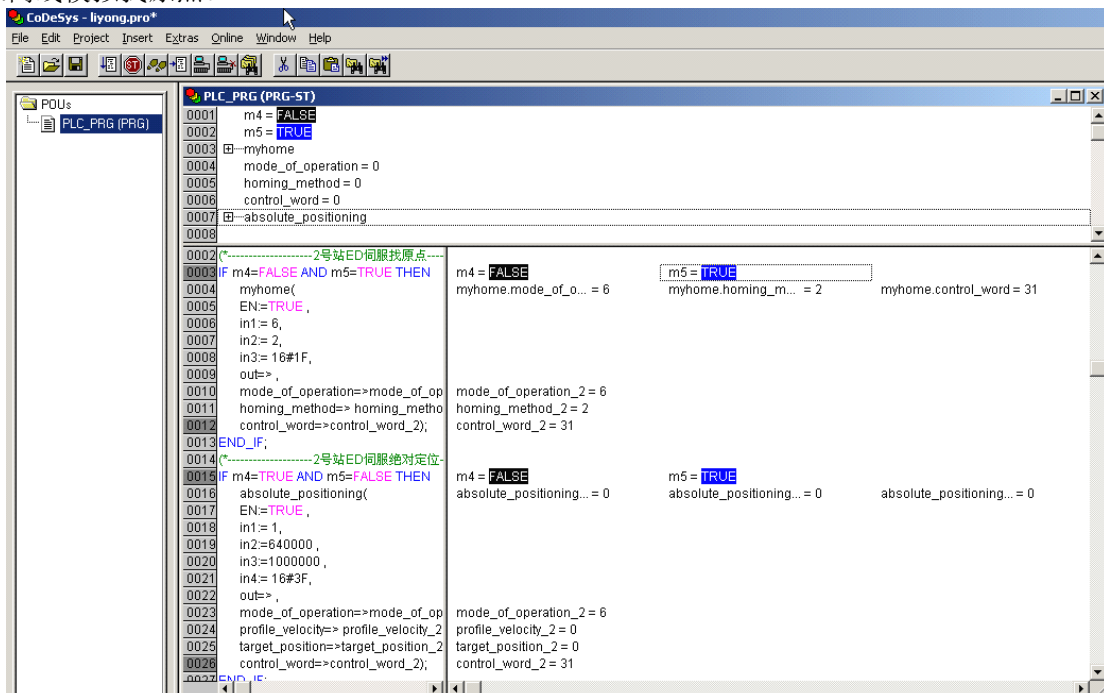
12. 在全局变量中定义好 PDO 中配置好的地址变量，方便程序中使用。



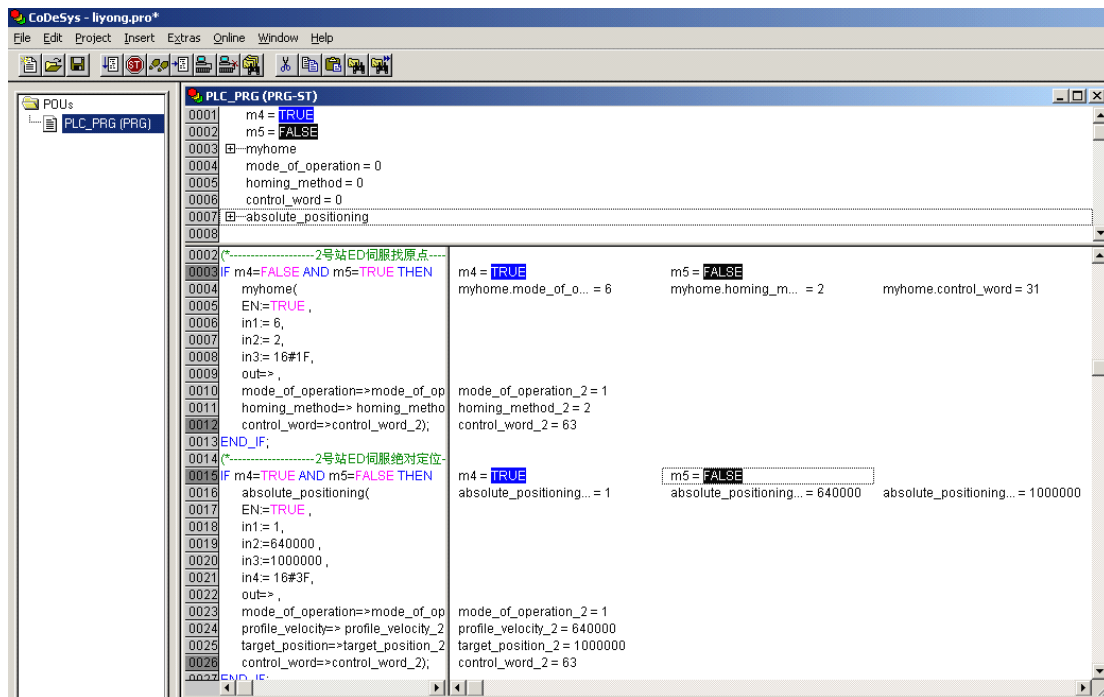
13. ST 语言编写程序。注意在 CODESYS 与伺服建立 CAN 通讯后，首先需要初始化伺服控制字为 6，否则伺服对其他指令无响应。



14. 离线模拟找原点。



15.离线模拟绝对定位。



附录 B: 串口通讯范例

一、 Kinco 触摸屏与 ED 伺服驱动器的联接与应用

注意：

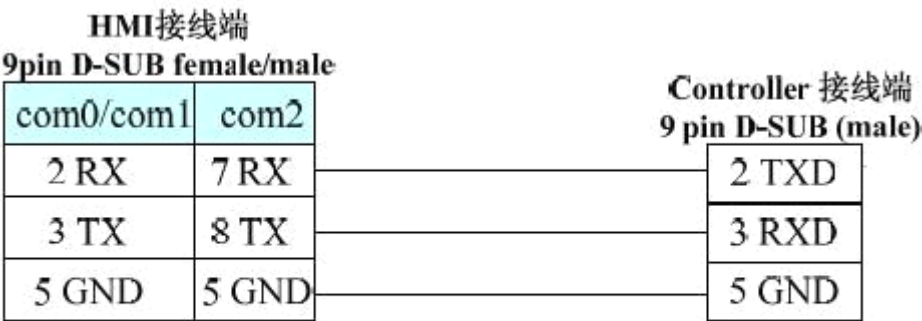
1: Kinco 全系列文本和触摸屏等都可以直接与 ED 驱动器串口连接，本范例只介绍 Kinco 触摸屏中的主打产品 MT4000,MT5000 系列与 ED 伺服连接。文本系列， MT500 系列触摸屏， MT6000 系列触摸屏等请参考各产品使用手册或咨询 Kinco 技术人员。

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 ED 伺服通讯范例”了解详细应用

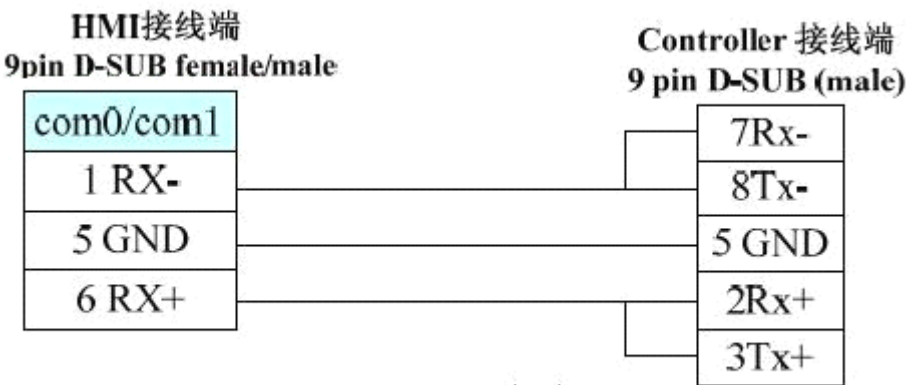
Kinco MT4000, MT5000 系列触摸屏可以直接与 ED 驱动器串口连接，用户可以用触摸屏设置 ED 的内部参数和运行状态。触摸屏既可以与单个 ED 驱动器相联，也可以与多个 ED 驱动器相联，联接方式为 RS232C 或 RS485。

1. 触摸屏控制单台 ED 伺服

a. 硬件接线



232通讯方式



485通讯方式

b. 通讯参数设置

触摸屏的通讯参数设置见图“D1.1”,注意的是 PLC 的站号就是 ED 驱动器的 ID 号相同, ED 驱动器出厂时 ID 默认为 1,所以触摸屏控制单台 ED 伺服时 PLC 站号设置为 1。如果 ED 的 ID 号设置为 N,那么下面的 PLC 站号也要设置为 N。

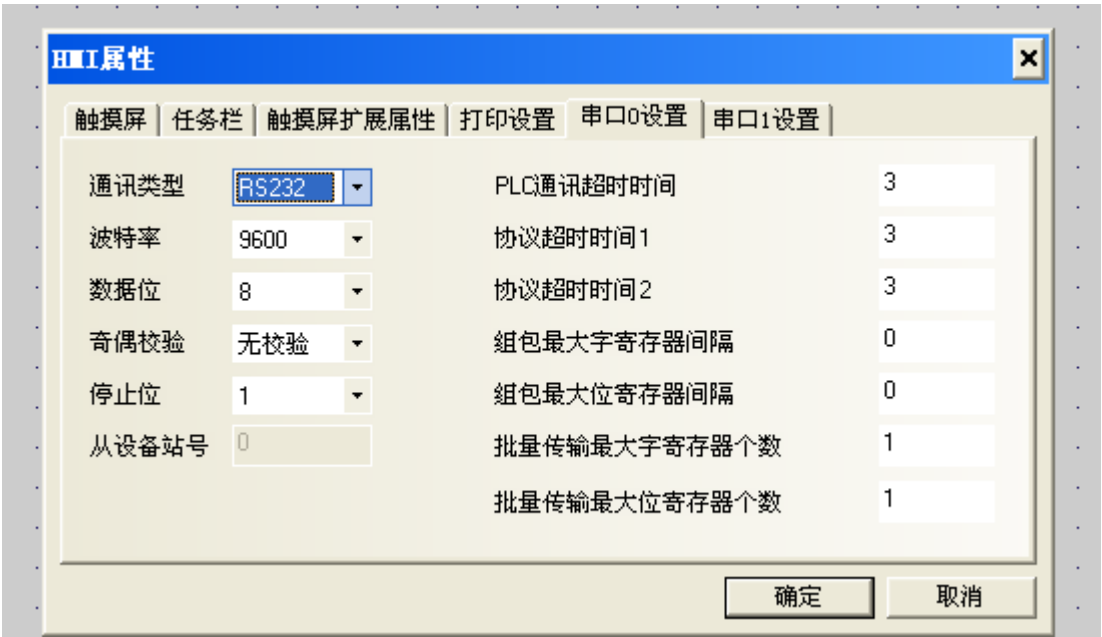


图 D1.1 屏通讯参数

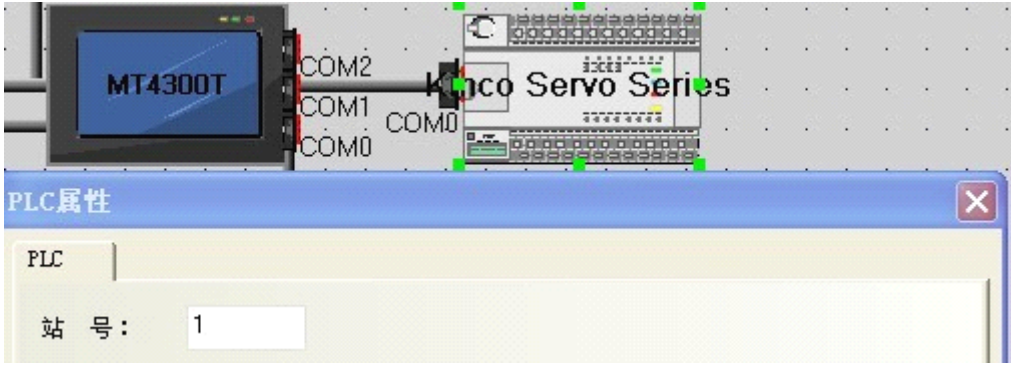


图 D1.1 屏站号参数

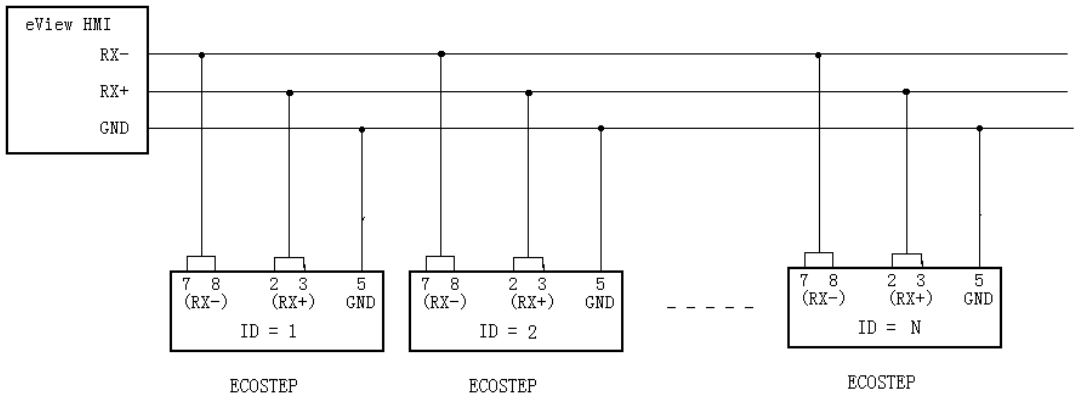
c. 地址参数设置

编写 HMI 程序时,首先设置 ED 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 ED 内部对象 60FF0020(目标速度)。



2. 触摸屏控制多台 ED 伺服

a、RS485 接线



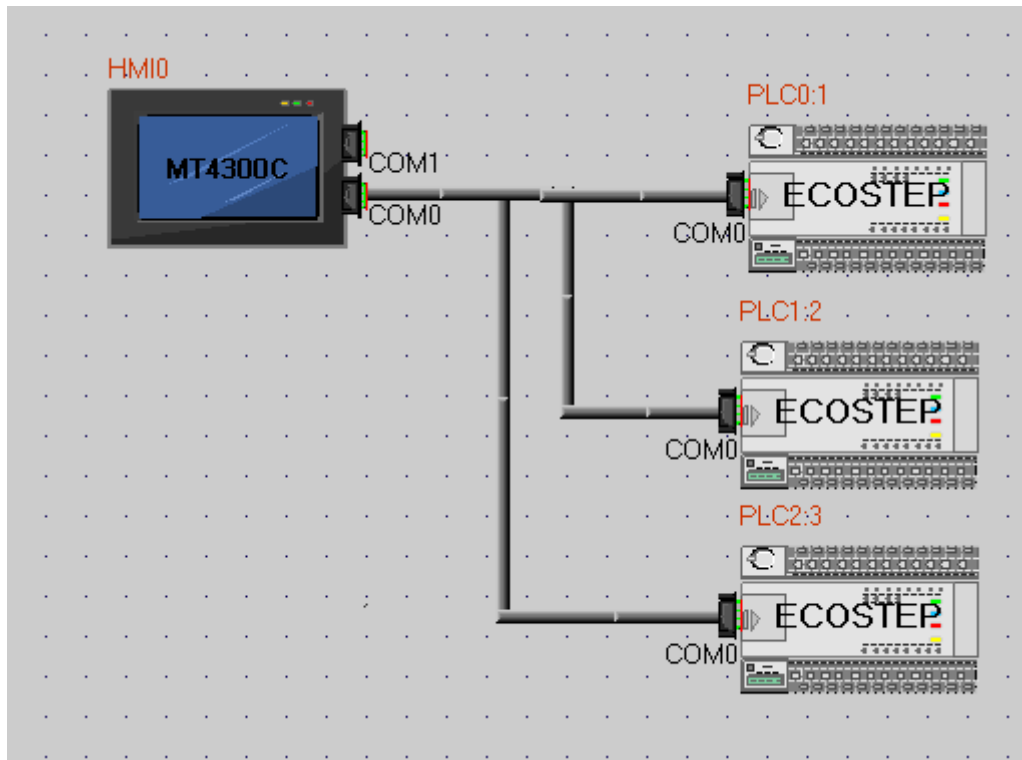


图 D1.3

c. 参数设置

触摸屏参数设置同“图 D1.1”，各台伺服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种的伺服（注意这里的 PLC 编号并不是设置的伺服站号，仅仅表示用户连接多台伺服的顺序，如上图，PLC0:1 表示的就是 PLC 编号是 0，站号是 1）。



d. 地址参数设置

与连接单台伺服时候设置完全一样。只是注意不同伺服选择不同的 PLC 编号。

二、西门子 S7200 PLC 和 ED 伺服利用 RS485 通讯

1. 通讯参数设置

S7200 端:

S7200 Plc 必须使用自由口通讯协议, SMB30=09(自由口方式);

S7200 Plc发送和接收数据长度最长设定为256Byte, 本范例设定VB1280~VB1290为发送缓冲区, 设置VB536~VB546为接收缓冲区。

ED 驱动器端:

ED 驱动器端 RS485 通讯参数: baud =9600bps, Data length=8, Stop bit = 1, Parity=none

ED驱动器必须使用带RS485接口的型号(带Profibus总线的驱动器无RS485口)。

2. 接线

二者间采用 485 方式通讯:

ED RS485 口		S7200
7、8(RX-)	—————	8(D-)
2、3(RX+)	—————	3(D+)
5(GND)	—————	5(GND)

3. 范例功能说明

S7200 plc 输入口	功能	说明
I0.0	60600008=-3	设定为立即速度模式
I0.1	60FF0020=1000000inc/s	设定速度模式下目标速度
I0.2	60600008=0x01	设定为位置模式
I0.3	607A0020=40000000	设定位置模式下的目标位置值
I0.4	60810020=60rpm	设定位置模式下的最大速度
I0.5	60400010=0x5F	控制命令=相对运动
I0.6	60400010=0x06	电机断电
I0.7	60400010=0x0F	电机上电

4. 系统运行

首先输入 I0.6, 然后输入 I0.7, 电机使能;

I0.0 设定控制器为速度模式,I0.1 设定目标速度值,电机即处于速度模式下运行

I0.2 设定控制器为位置模式, I0.4 设定最大速度和 I0.3 设定目标位置值, I0.5 设定电机为相对定位。

5. 其它

S7200 通过 RS485 与 KINCO 伺服 X2(485)口相连时，用户还可用 KINCO 触摸屏通过 RS232 与 KINCO 伺服连接完成监控以及参数的设置，实现一个带完整的输入、输出、显示和设定功能的系统。

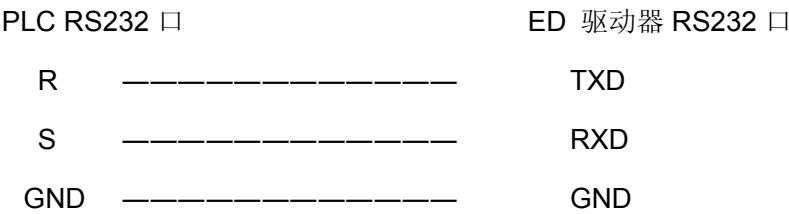
范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服- 范例程序” 下载。

三、松下 FP0 PLC 与 ED 伺服利用 RS232 通讯

1. 通讯参数设置

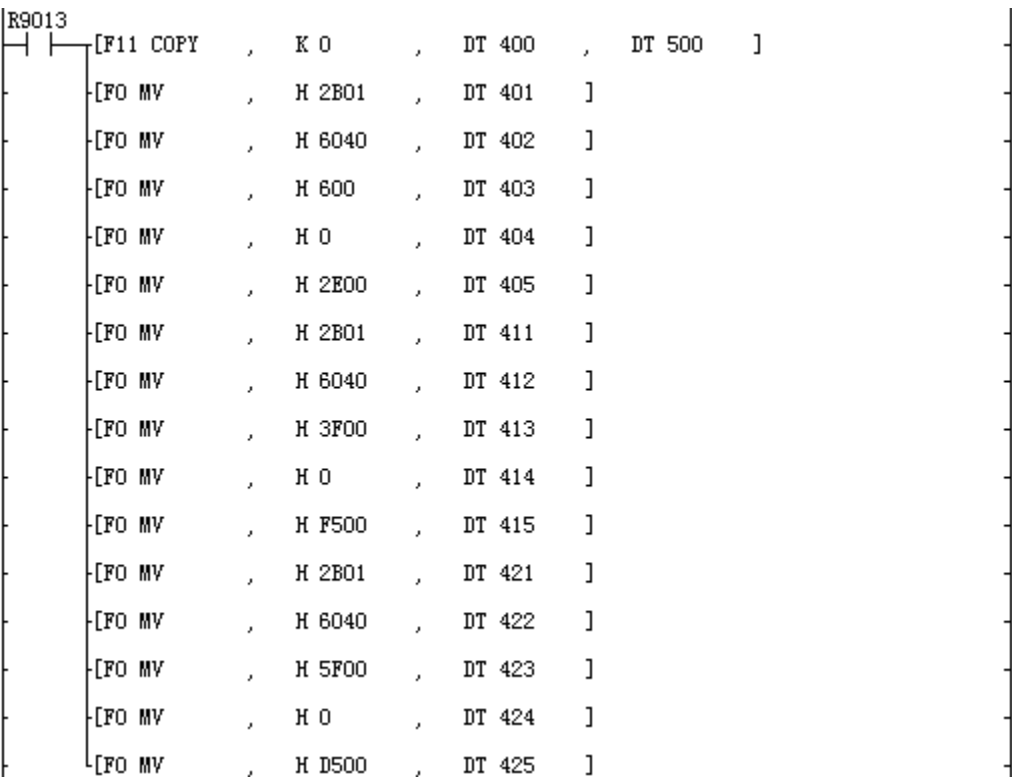
在松下 PLC 编程软件的“PLC 系统寄存器设置”中，设置 No. 412=通用通讯，NO. 414=9600，N0413=数据长 8 位，停止位 1 位，无效验，接收缓冲区起始地址 DT2000，容量 20。

2. 接线



3. 程序

将用到的指令事先存入数据区



DT401-DT405 的指令： 01 2B 40 60 00 06 00 00 00 2E 电机断电

DT411-DT415 的指令： 01 2B 40 60 00 3F 00 00 00 F5 控制命令为 3F，电机做相对定位

DT422-DT425 的指令： 01 2B 40 60 00 5F 00 00 00 D5 控制命令为 5F，电机做相对定位

注释：停止指令 01 2B 40 60 00 06 00 00 00 2E 共 10 个 Byte。

01—ED 的 ID

2B—写入

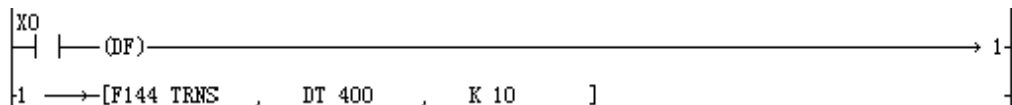
6040—控制字

06—停止控制字

2E—为效验码，效验码计算请参考第 11.1.2 节，用户可设计一段子程序自动计算效验码。

上述计算全部以 16 进制计算，具体的控制和状态指令请参考 ED 的对象词典表。

最后利用松下 PLC 的发送串口指令 F144 将指令发送到 ED 驱动器。



上述指令将 DT401 开始的 10 个 byte 的数据发送到串口，而 DT400—DT405 在上面已经设为停止，该命令发出后，电机（ID=1）停止动作。

范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服- 范例程序”下载。

四、三菱 FX2N PLC 与 ED 伺服利用 RS422 通讯

1. FX2N 通过 FX2N-485BD 和 ED 间 RS485 接线

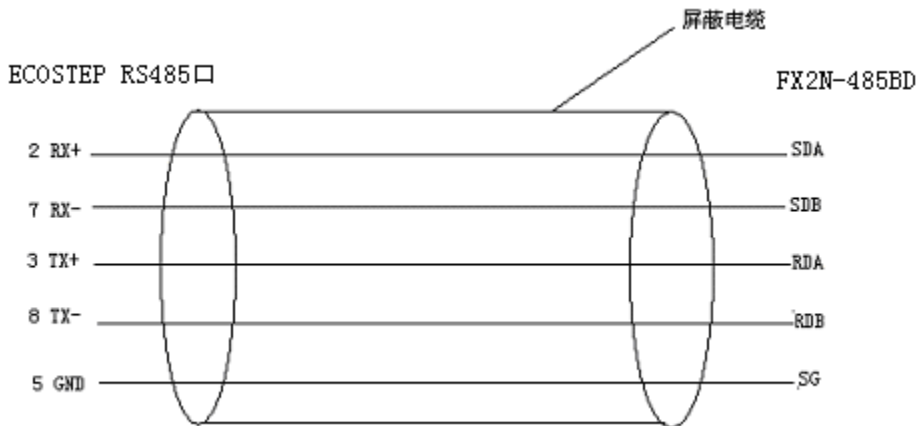


图 D4.1

2. 通讯参数配置

三菱 FX2N 端：

设置 D8120=H0C81，无协议，无起始符，无结束符，数据位长 8 位，停止位 1 位，无效验，波特率 9600，D8120 = H0C81 (b15..b12= 0000, b11..b8=1100, b7..b4=1000, b3..b0= 0001)。

Bit位	定义	0	1
B0	Data length	0—7bits	1—8bits

1	B2,B	Parity	00-no parity, 01-Odd parity, 11-Even parity		
	B3	stop bits	0- 1 bits	1- 2 bits	
	B4	Baud rate-bps	1000- 9600bps		
	B5		1001 –19200bps		
	B6				
	B7				
	B8	Header character	0- none	1 –D8124(指定, Default=STX , 02H)	
	B9	Terminator character	0- none	1 –D8125(指定, Default=ETX , 03H)	
	B10	Communication Control (see timing diagrams page 10-20 onwards)	无协议		
	B11		(b12, b11, b10)		
	B12		(0, 0, 0) : RS Instruction is not being used (RS232C interface)		
			(0, 0, 1) : Terminal mode -RS232C interface		
			(0, 1, 0) : Interlink mode - RS232C interface (FX2N V2.00 or above)		
			(0, 1, 1) : Normal mode 1- RS232C, RS485(422) interfaces (RS485 FX2N(C) only)		
			(1, 0, 1) : Normal Mode 2 - RS232C interface (FX only)		
			Computer Link		
			(b12, b11, b10)		
			(0, 0, 0) : RS485(422) interface		
			(0, 1, 0) : RS232C interface		
	B13	F X-485	Sum check	No check	Added automatically
	B14	N etwor	protoc ol	No protocol	Dedicated Protocol
	B15	k	protoc ol	Format 1	Format 4

ED 端: 2F910008, 设 N=63, 既波特率 9600

3. 通讯协议

采用自由通讯协议，FX2N 端依据 ED 的指令传输，

写：10 个字节（固定）的传输协议，ID 为 ED 的地址，由拨码设置，0 为保留字

CHKS 的计算：(0— (byte0++byte 9)) 结果值的最后两位

byte0		byte 9
ID	8 Byte Date	CHKS

例如：将控制模式设为速度模式 3

01 2B 60 60 00 03 00 00 00 11

01—ID 地址， 2B- 写命令， 606000— 改变控制方式的地址， 11—CHKS

读：例如读地址为 1 的 ED 的 485 通讯参数，通过查询 ED 的对象词典，RS485 通讯参数波特率设置地址为 2F9100,则读命令为：

01 40 91 2F 00 00 00 00 00 FF

FF- 为效验码 = (0— (byte0++byte 9))

40—为读指令

4. 范例说明

范例 485RW.PMW 为三菱程序，

X2- Power on

X3- Power off

X4- 设控制模式为 3

X5—改变速度为 10 转/分

X6- Reset

X7- 读 ED 的 485 通讯的波特率

范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服- 范例程序”下载。

五、欧姆龙 CJ1W-CIF11 通讯单元与 ED 伺服利用 RS422 通讯

一. 硬件接线：

与 ED 以 RS422 连接方式：

ED 系列伺服的 RS485 接口管脚图如下：

管脚号	接脚名	说明
7	RX-	数据接收信号线A
2	RX+	数据接收信号线B
8	TX-	数据传输信号线A
3	TX+	数据传输信号线B
5	GND	接地信号线

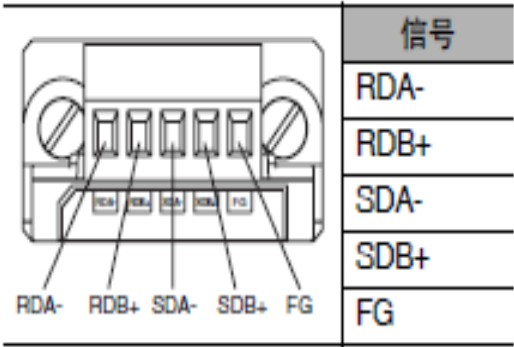
CJ1W-CIF11 RS-422A 转换单元最大距离为 50M

外观：

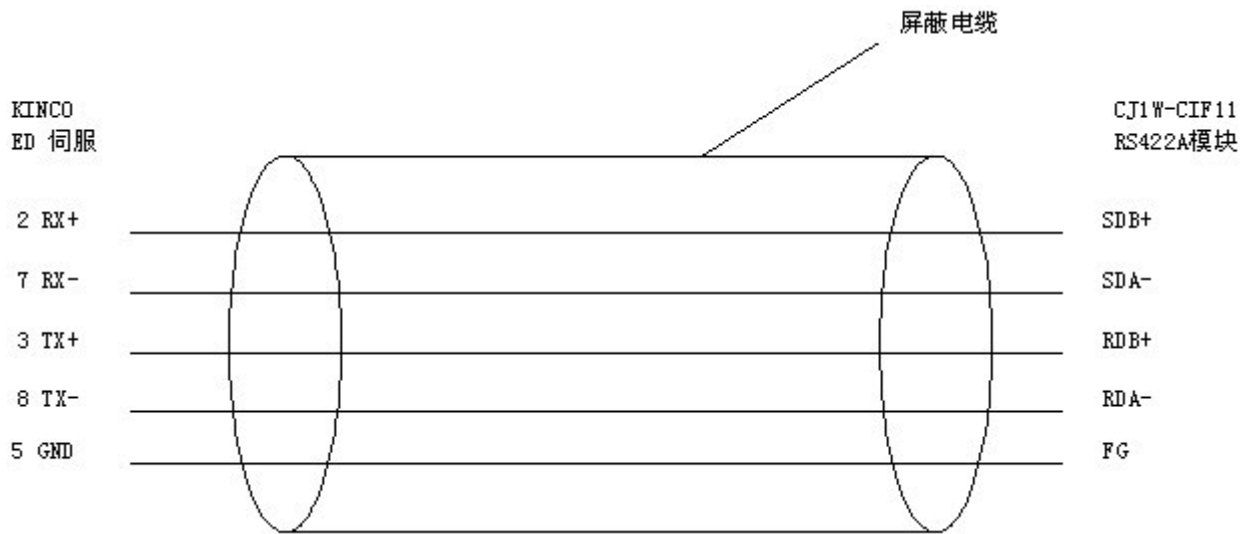


电气性能：

• RS-422A/485 端子台



如下图接线：



二. 软件设置：

1. 通讯参数配置：

OMRON 端：设置无协议 RS232C，无起始符，无结束符，数据位 8 位，停止位 1 位，无校验，波特率 9600Kbps。

ED 伺服端默认通讯参数为：数据位 8 位，停止位 1 位，无校验，波特率 9600Kbps。

在欧姆龙软件 PLC 设置对话框中，点击“串行选项端口”（Serial Option Port）标签。



配置好参数，下载 PLC 设置。

然后断电重启 PLC。

ED 伺服端在 ECO2WIN 软件中设置：2F910008,设 N=63,既波特率 9600

三．范例说明

范例为 OMRON 程序

0.00-Power on

0.01-Power off

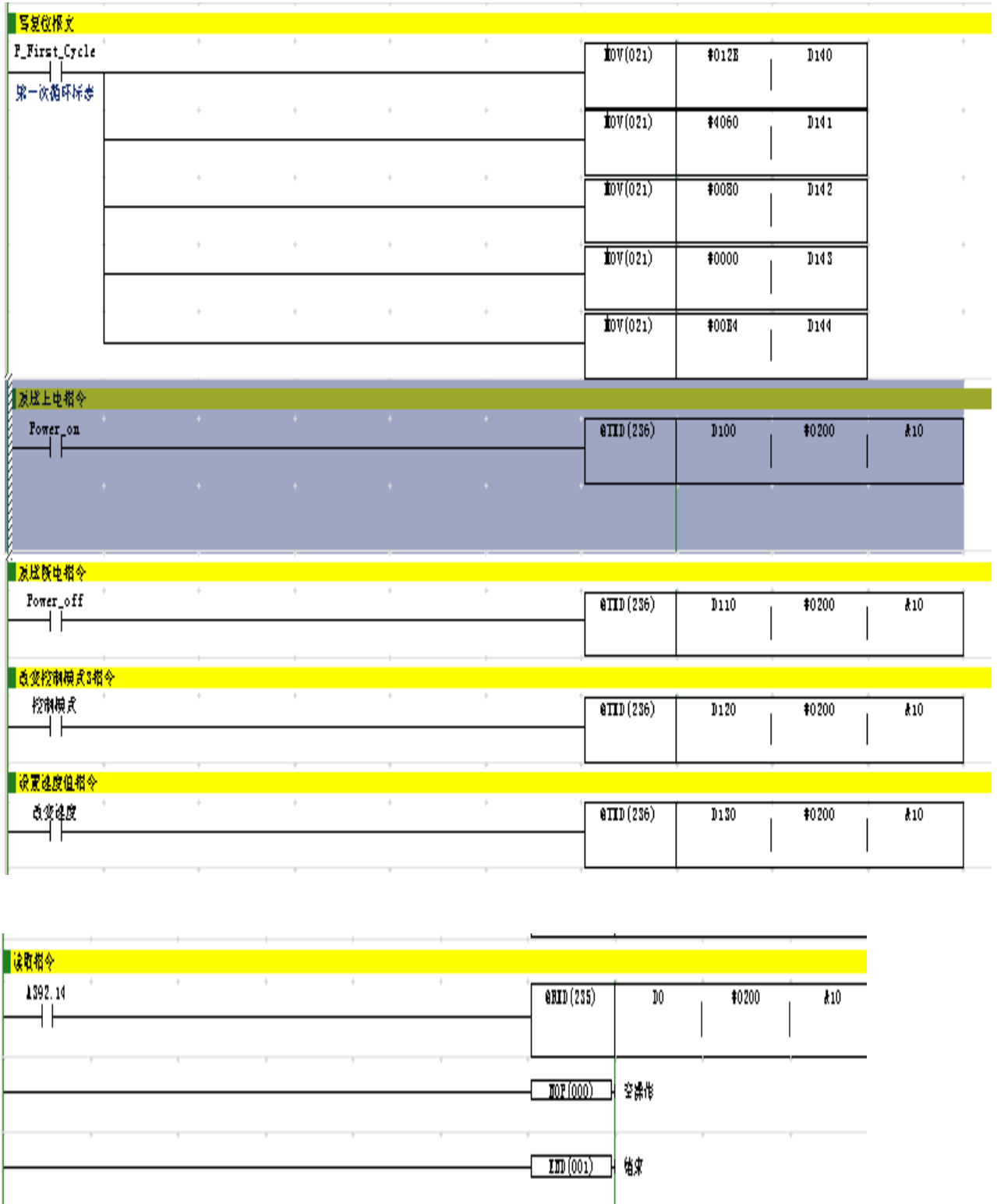
0.02-设置控制模式为 3

0.03-改变速度为 60 转/分

0.04-Reset

【程序名：新建程序1】									
【原名称：原1】									
写上位机报文									
P_First_Cycle 第一次循环标志						MOV(021)	#012B		D100
						MOV(021)	#4060		D101
						MOV(021)	#000F		D102
						MOV(021)	#0000		D103
						MOV(021)	#0025		D104
写断机报文									
P_First_Cycle 第一次循环标志						MOV(021)	#012B		D110
						MOV(021)	#4060		D111
						MOV(021)	#0006		D112
						MOV(021)	#0000		D113
						MOV(021)	#002E		D114

写控制模式3报文									
P_First_Cycle 第一次循环标志						MOV(021)	#012F		D120
						MOV(021)	#6060		D121
						MOV(021)	#0003		D122
						MOV(021)	#0000		D123
						MOV(021)	#000D		D124
写速度60 RPM 报文									
P_First_Cycle 第一次循环标志						MOV(021)	#0123		D130
						MOV(021)	#FF60		D131
						MOV(021)	#0000		D132
						MOV(021)	#C409		D133
						MOV(021)	#00B0		D134



范例程序 www.kinco.cn “下载中心-资料下载-伺服系统-ED 伺服- 范例程序” 下载。

附录 C： 顺序编程（**Sequencer programming**）范例

注意：

- 1: 本范例介绍 I/O 口控制伺服在回原点模式，位置模式，速度模式下的运转。
- 2: 为了更好的展示 ED 伺服强大内部编程功能，本范例还结合介绍了计数器，定时器，比较器，高速脉冲捕捉，模拟量输出监控，高低速设置不同 PID 参数等功能，还介绍程序跳转，程序互锁等逻辑控制功能。
- 3: 本范例仅作为功能展示应用，不针对任何具体应用场合，客户具体应用需看具体情况加以消化吸收。

一、I/O 口定义及功能

DIN1: 每触发一次，计数器加 1

DIN2: 自动模式（此时点动模式无效），伺服先回原点，原点找到后把当前位置加上 100 作为下面绝对定位的位置，然后依次执行相对定位 20000inc，定位到达后绝对定位，定位到达后经 OUT1 输出一个维持 1 秒的信号

DIN3: 伺服以速度模式 500RPM 运转，计数器 1 的值等于 10 后，停止运转，计数器 1 清零

DIN4: 点动正转（此时自动模式无效）

DIN5: 点动反转（此时自动模式无效）

DIN6: 硬件正限位

DIN7: 硬件负限位

DIN8: 原点开关

Position Capture 口: 要求有信号时,对主编码器值清零.

RESET: 故障复位

ENABLE: 外部使能

OUT1: 数字输出（PLC 标准、最大电流=0.5A）

根据程序需要在目标位置到达后维持 1 秒的高电平

OUT2: 数字输出（PLC 标准、最大电流=0.5A）

高电平，在驱动器有故障下

MON1: 模拟量监控电机电流

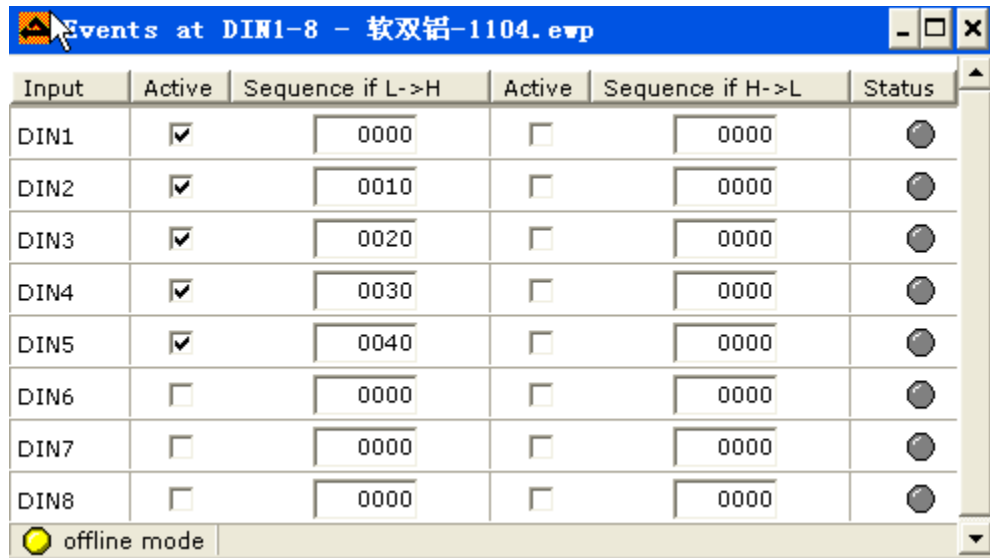
MON2: 模拟量监控电机速度

二、各数字口调用程序段定义

I/O 口	调用程序段号	I/O 口	调用程序段号
DIN1	Seq 00	DIN4	Seq30-31

DIN2	Seq 10-18	DIN5	Seq40-41
DIN3	Seq20-22	Position Capture <input type="checkbox"/>	Seq50-51

定义输入口:



三、程序定义

Seq.00 (计数器计数)

21900120 = 1	Counter 0 Increment	计数器 1 在 Seq 00 每执行一次计数 1 次, 总计数值保存在 21900220 里面
--------------	---------------------	---

Seq.10 (点动/自动互锁)

20110108 = 1	Sequence 11 valid	程序段 11 有效, 允许自动模式
20310108=0	Sequence 31 valid	程序段 31 无效, 禁止点动正转
20410108=0	Sequence 41 valid	程序段 41 无效, 禁止点动反转
21180008=11	Run sequence	跳转程序段到 11

Seq.11 (驱动器 Drive off)

60400010 = 06	Control word	电机断电
21300220=500	Timer Delay time	延时 500 毫秒
21300110=8012	Timer Sequence call	延时跳转到程序段 12

Seq.12 (准备回原点)

60600008 = 6	Chosen mode of operation	设置为原点模式
--------------	--------------------------	---------

607C0020=0	Relocation zero point	原点偏移
60980008=5	Homing Method	找原点方式，自行根据需要设置
60990120=10000	Velocity during search for reference switch	找寻原点 INDEX 速度，自行根据需要设置
60990220=50000	Velocity during search for reference set point	找寻原点速度，自行根据需要设置
60400010 = F	Controlword	电机上电锁轴，准备开始回原点
21180008 = 13	Run sequence	跳转程序段到 13

Seq.13 (开始找原点)

60400010 = 1F	Controlword	开始回原点
21400210=8014	Controller event "Home found"	原点找到后，跳转程序段到 14

Seq.14(绝对定位到 0)

60600008 = 1	Chosen mode of operation	设置为位置模式 1
607A0020 = 0	Target position	目标位置值
60810020 = 100000	Profile velocity	最大速度 100000 inc/s，自行设置
60830020 = 500000	Profile acceleration	加速度 500000 * 16 inc/s，自行设置
60840020 = 500000	Profile deceleration	减速度 500000 * 16 inc/s，自行设置
60400010 = 3F	Controlword	执行绝对定位
21400110 = 8015	Controller event "Target reached"	目标位置到后，跳转程序段到 15

Seq.15(把当前位置作为下面绝对定位位置)

21A00120 =60630020	Calculator Mapping Data Source	计数器目标数据源（当前位置）
21A00320=100	Calculator Operand	计算器被执行数
21A00410=1	Calculator Operator	执行加运算
21A00220=20170520	Calculator Mapping Result	把计算结果放到程序段 17 第二行右边的值
21180008=16	Run sequence	跳转程序段到 16

Seq.16(相对定位到 20000)

60400010 = F	Controlword	准备执行相对定位（执行相对定位这条必须要有）
60600008 = 1	Chosen mode of operation	设置为位置模式 1
607A0020 = 20000	Target position	目标位置 20000inc（2 转）
60810020 = 100000	Profile velocity	最大速度 100000 inc/s，自行设置
60830020 = 500000	Profile acceleration	加速度 500000 * 16 inc/s，自行设置
60840020 = 500000	Profile deceleration	减速度 500000 * 16 inc/s，自行设置
60400010 = 5F	Controlword	执行相对对定位
21400110 = 8017	Controller event "Target reached"	目标位置到后，跳转程序段到 17

Seq.17 绝对定位)

60600008 = 1	Chosen mode of operation	设置为位置模式 1
607A0020 =	Target position	目标位置值来自 14 段程序计算结果，这里 设置不设置均可
60810020 = 100000	Profile velocity	最大速度 100000 inc/s，自行设置
60830020 = 500000	Profile acceleration	加速度 500000 * 16 inc/s，自行设置
60840020 = 500000	Profile deceleration	减速度 500000 * 16 inc/s，自行设置
60400010 = 3F	Controlword	执行绝对定位
21400110 = 8018	Controller event "Target reached"	目标位置到后，跳转程序段到 18

Seq.18(OUT1 输出高电平)

21600120=60FE0120	(OUT1) Mapping Data Source	配置 OUT1 数据源为 60FE0120（输出控制字）
21600220=0	(OUT1) Offset	偏移量 0
21600320=10000	(OUT1) AND-mask	逻辑与值 10000
21600420=10000	(OUT1) Comparision value	比较值 10000
21600520=0	(OUT1) Polarity	极性 0 等于输出高电平
60FE0120=10000	Digital outputs Output word	输出控制字为 10000，OUT1 输

		出高电平
21300220=1000	Timer Delay time	延时 1000 毫秒
21300110=8019	Timer Sequence call	延时跳转程序段到 19

Seq.19(OUT1 输出断开)

60FE0120=0	Digital outputs Output word	输出控制字为 10000, OUT1 输出高电平
------------	-----------------------------	--------------------------

Seq.20 (速度模式)

60600008 = 3	Chosen mode of operation	设置为速度模式 3
60FF0020 = 100000	Target velocity	目标速度=100000 inc/s (600RPM)
60830020 = 500000	Profile acceleration	加速度 500000 * 16 inc/s, 自行设置
60840020 = 500000	Profile deceleration	减速度 500000 * 16 inc/s, 自行设置
60400010 = F	Controlword	伺服锁轴, 执行速度模式
21180008=21	Run sequence	跳转程序段到 21

Seq. 21(比较计数器 1 的值)

21800120=21900220	Comparator 0 Mapping Data Source	把计数器 1 的值 21900220 作为欲比较值
21800220=0	Comparator 0 Offset	偏移量 0
21800320=FFFFFFFF	Comparator 0 AND-mask	逻辑与
21800420=10	Comparator 0 Comparison value	比较值 10
21800510=1	Comparator 0 Operator	比较操作 等于
21800610=8022	Comparator 0 Sequence call	比较成功, 程序跳转到 22

Seq.22 (停止伺服, 计数器清零)

60FF0020 = 0	Target velocity	目标速度=0 inc/s 停止伺服
21900220= 0	Counter 0 Value	计数器 1 的值清零

Seq.30 (点动/自动互锁)

20110108=0	Sequence 11 valid	程序段 11 有效，禁止自动模式
20310108=1	Sequence 31 valid	程序段 31 有效，允许点动正转
20410108=0	Sequence 41 valid	程序段 41 无效，禁止点动反转
21180008=31	Run sequence	跳转程序段到 31

Seq.31 (速度模式)

60600008 = 3	Chosen mode of operation	设置为速度模式 3
60FF0020 = 100000	Target velcity	目标速度 = 100000 inc/s (600RPM)
60830020 = 500000	Profile acceleration	加速度 500000 * 16 inc/s, 自行设置
60840020 = 500000	Profile deceleration	减速度 500000 * 16 inc/s, 自行设置
60400010 = F	Controlword	伺服锁轴，执行速度模式

Seq40,41 点动反转与 Seq30,31 点动正转类似，只是速度变为负值。60FF0020 = -100000

Seq.50 (高速脉冲捕捉)

21C00220=0	Position capture Strobe counter	高速脉冲计数清零，允许高速脉冲捕捉
21C00110=8051	Position capture Sequence call	高速脉冲捕捉后，跳转程序段到 51

Seq.51 (主编码器脉冲清零)

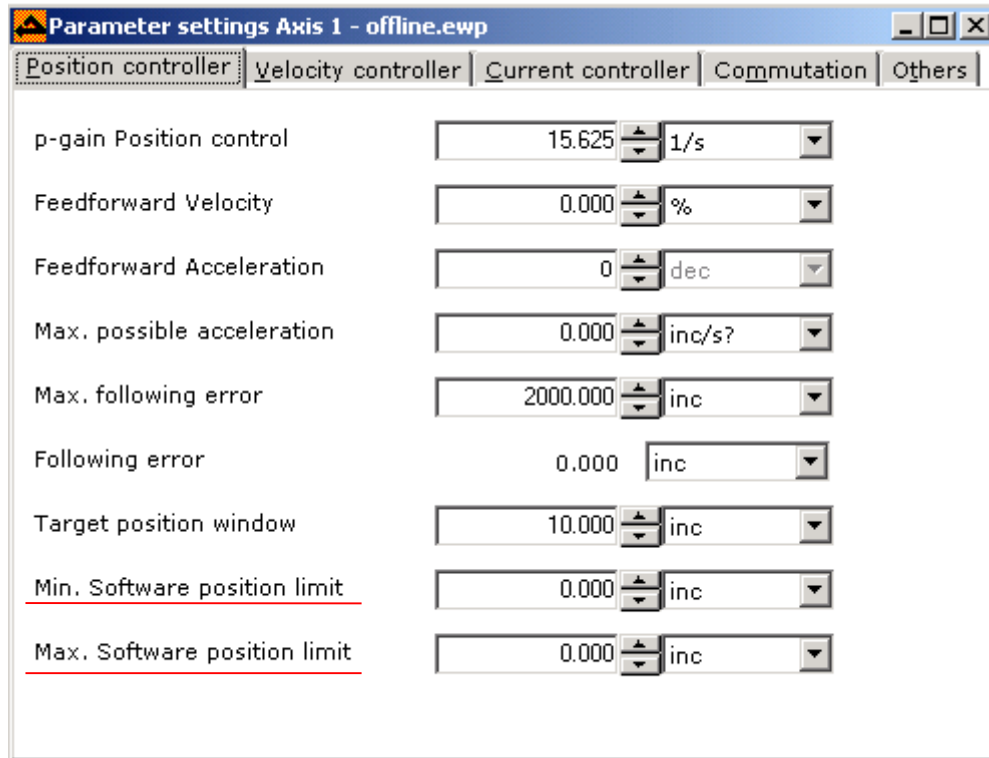
25090620=0	Actual position master	主编码器计数清零
21180008=8050	Run sequence	跳转程序段到 50，以便循环捕捉

定义硬件限位 参见第七章！

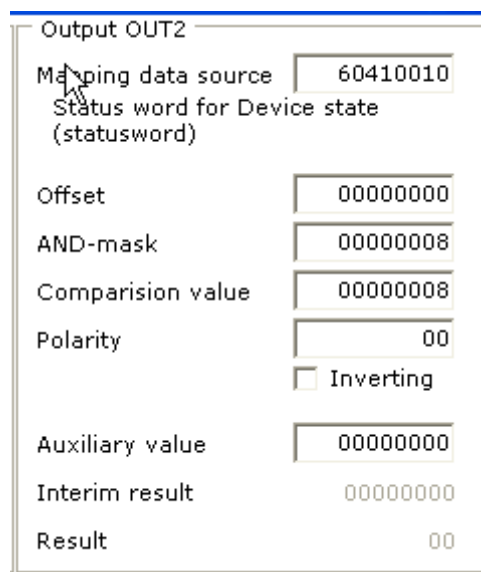
根据需要定义软件限位：（仅在位置模式下有效）

设置 min.software position limit=0

max.software position limit=0.



定义 OUT2 输出口：该输出在： **Fault=1**（报警） 条件满足时将输出高电平。



定义模拟量监控口：

每个模拟量口（mon1/2）有一个输出范围 0~5V (2.5V 等于映射对象值=0 时输出)。驱动器内每个对象都可以映射到模拟量输出口，公式为：

$$U_{mon} = 1V * \text{internal notation of dimension} * \text{factor} / 256^{(\text{preshift}+1)/120}$$

下面例子将模拟量输出口 1 映射到电机实际电流，模拟量输出口 2 映射到电机实际速度：

Monitor 1: Mapping = 60780010 (curr_act_val)

Pre-shift= 0

Factor = 30

$$U_{mon1} = 1V \times \frac{\frac{2047}{256^{(1+0)}} \times 30}{120} \times \frac{1}{120} = 0.166V/A \quad (I_{mon1}: -12A \dots +12A = 0.5V \dots 4.5V)$$

$V_{out} = U_{mon1} \times I + 2.5$ 上式中 2047（单位 DEC）为驱动器内部单位，其对应电流 12A 这里需要做工程量和物理量的转换

注： V_{out} =驱动器输出实际电压值。

I = 驱动器内部电流值。

Monitor 2: Mapping = 606C0020 (vel_actual_val)

Pre-shift=2

Factor=94

$$U_{mon2} = 1V \times \frac{\frac{20000000}{256^{(1+2)}} \times 118}{2344rpm \times 120} = 0.5mv/rpm \quad (U_{mon2}: -5000 \dots 5000rpm = 0 \dots 5V)$$

$V_{out2} = 2.5V + U_{mon2} \times \text{电机转速 rpm}$ 。上式中 640000（单位 DEC）为驱动器内部单位，其对应速度 60RPM（电机编码器分辨率 10000）， 这里需要做工程量和物理量的转换

最后保存所有参数：

“Administration—>Save parameters to device now”。

附录 D： ED 伺服应用范例

一、主从跟随控制

当使用主从跟随模式控制 ED 伺服驱动器时，主编码器信号可以由 PLC、运动控制卡、编码器、ED 伺服编码器输出口、变频器输出口等提供的 5V 差分信号，

1. 接线示意图：

ED 伺服驱动器和 ED 伺服驱动器接线。

主伺服端			从伺服端		
编码器输出口（九针公头）			编码器输入口（九针公头）		
+5V	1	_____	1	+5V	
A	2	_____	2	A	
B	3	_____	3	B	
N	4	_____	4	N	
	5		5		
GND	6	_____	6	GND	
/A	7	_____	7	/A	
/B	8	_____	8	/B	
/N	9	_____	9	/N	

注意：ED100 无编码器输出口

ED 伺服驱动器和外部编码器接线.



2. 参数设置

1) 在程序编辑里(Sequencer programming—>Sequences), 选择第一和第二段程序有效, 程序分别设置如下:

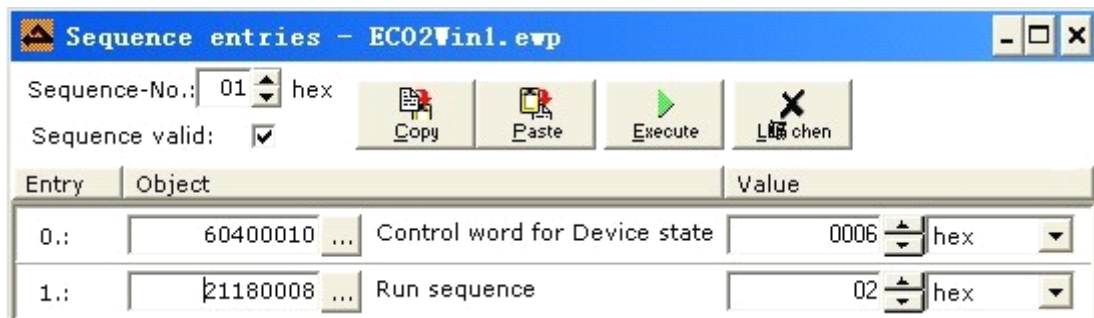
Seq01.

60400010=6

电机断电

21180008=2

跳转到第 2 段程序



Seq02.

60400010=F

60400010=F

60600008=-4

主从模式

25090220=60ff0020

速度映射

25090508=0

接受主从跟随控制

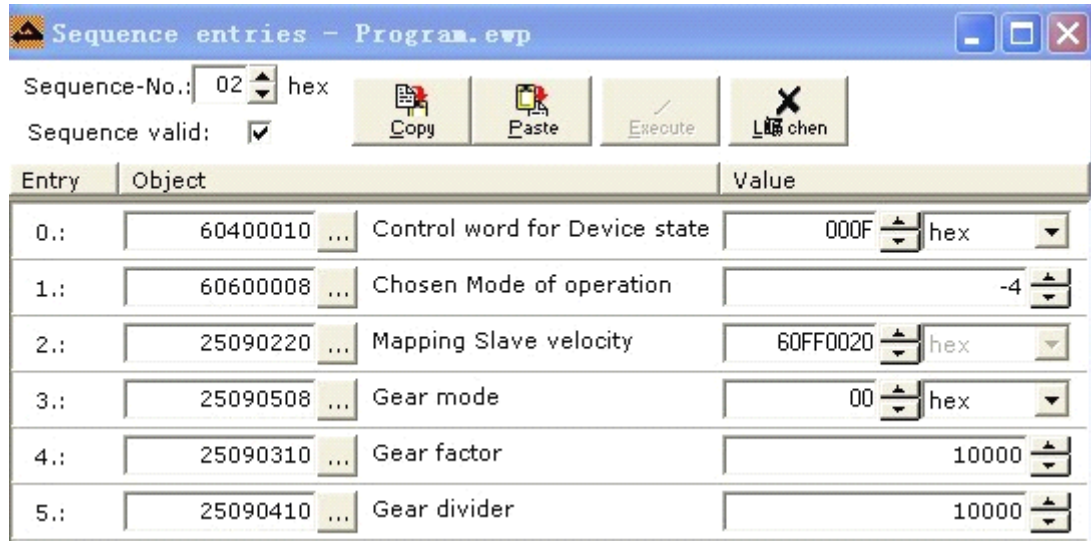
25090310=10000

齿轮比的分子 Factor, 可修改

25090410=10000

齿轮比的分母 Divider, 可修改

(Gear factor / Gear divider) × 伺服电机编码器分辨率 = 伺服电机旋转一圈所需的脉冲数。



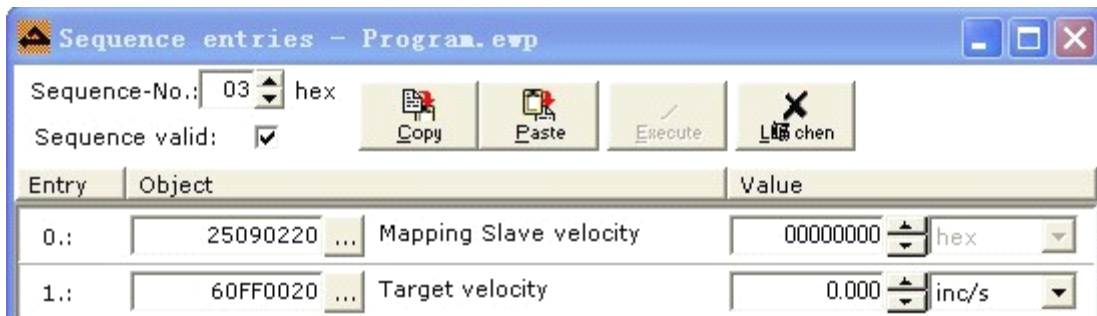
Seq03.

25090220=0

取消接受脉冲和方向信号控制

60FF0020=0

映射后的速度为 0，防止误动作



Seq04.

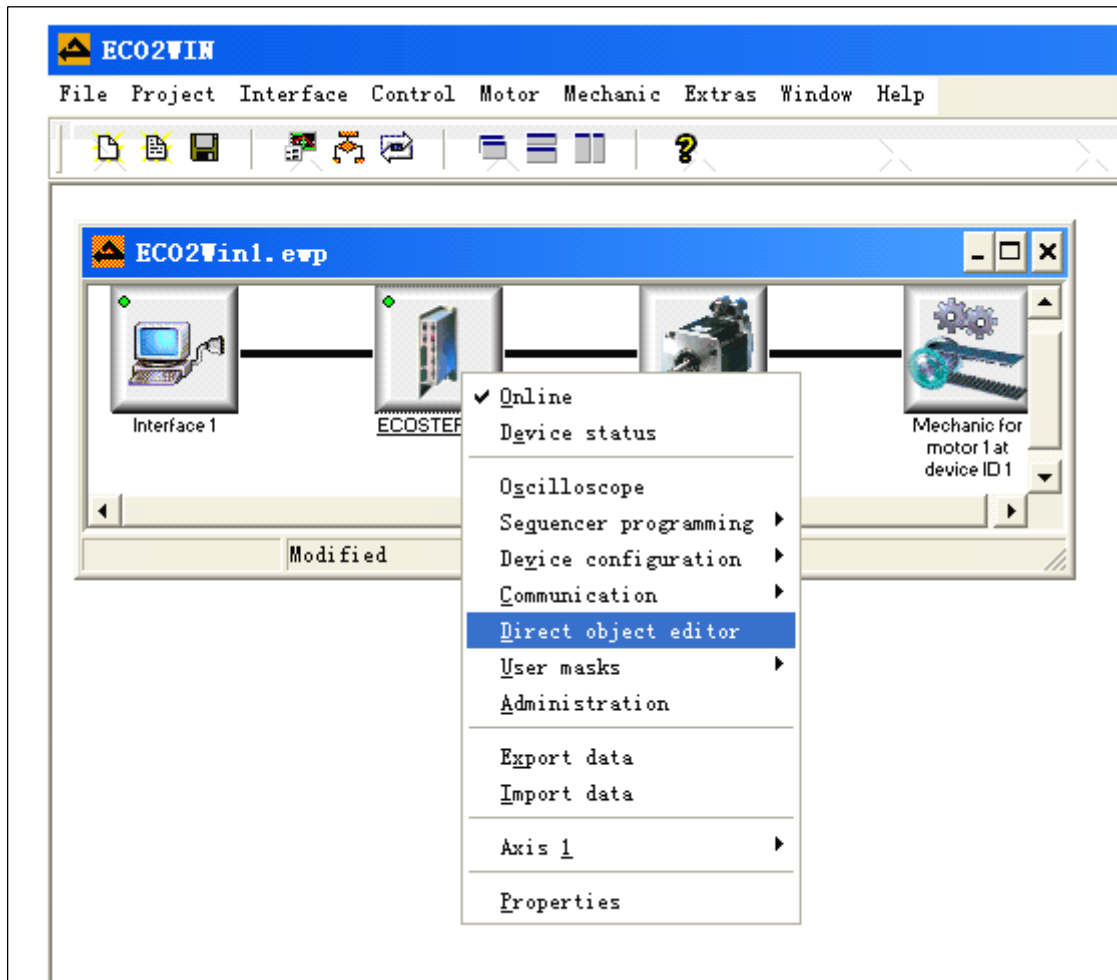
60400010=6

电机断电

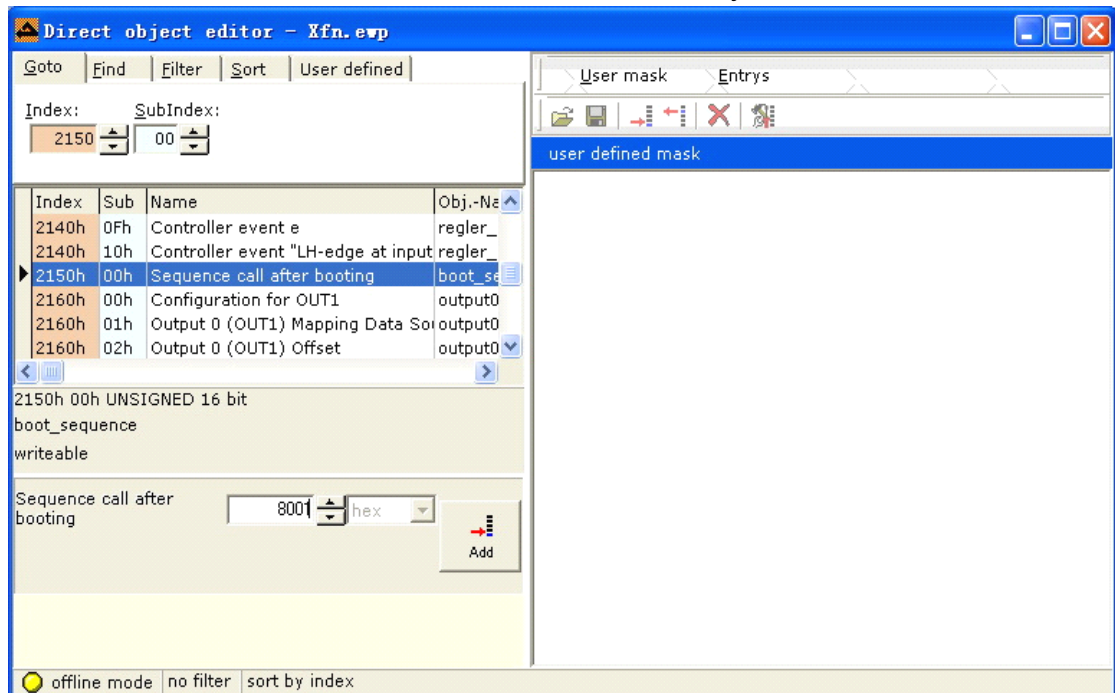


2) 如果要上电后就进入主从跟随模式，需如下设置：

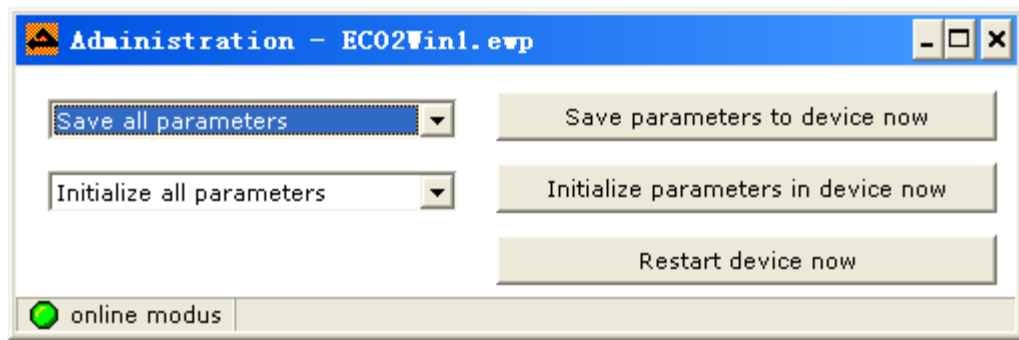
在 Direct Object entry 中设置 21500010=8001，驱动器上电后即可直接调用第一段程序，



ECOSTEP ID1 驱动器图标点击右键，弹出菜单中点击 Direct object editor 进入画面。



在 Index: 中输入 2150 ; SubIndex: 中输入 00 ; Sequence call after booting 中输入 8001; 最后点击 Add。



点击 **Save parameters to device now** 保存所有参数，下一步 **Restart device now**。重新上电，检查电机轴是否锁紧，如果是说明设备成功。

3) 如果是通过 I/O 控制驱动器进入主从跟随模式，则：

将驱动器上的 DIN1 接到外部控制器的输出口，通过外部控制器(如 PLC)来触发 DIN1,通过调用第 1 段程序来实现用“外部编码器信号控制 ED 伺服驱动器”的目的。DIN2“取消编码器信号控制 ED 伺服驱动器”功能，DIN3 电机轴松开。

ED 共有 8 个输入口(DIN1-DIN8)，供外部调用，每个输入口可以设定对应两段内部程序(上升沿和下降沿)，输入要求@24V 4mA，高电平 >12V 低电平 <5V 输入延迟要求 1ms。

详细见硬件手册。

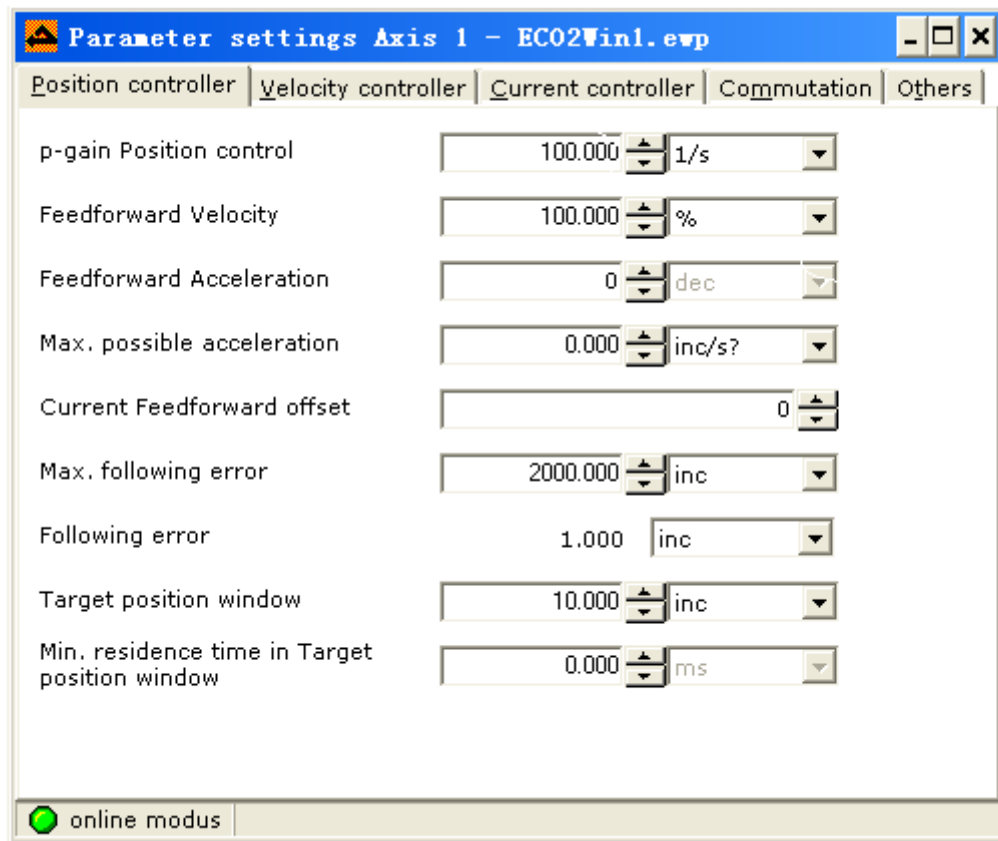
具体设置如下：

Sequencer programming->Digital input (以下设置为上升沿有效):

Events at DIN1-8 - 100aa.ewp					
Input	Active	Sequence if L->H	Active	Sequence if H->L	Status
DIN1	<input checked="" type="checkbox"/>	0001	<input type="checkbox"/>	0000	
DIN2	<input checked="" type="checkbox"/>	0003	<input type="checkbox"/>	0000	
DIN3	<input checked="" type="checkbox"/>	0004	<input type="checkbox"/>	0000	
DIN4	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN5	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN6	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN7	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN8	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
online modus					

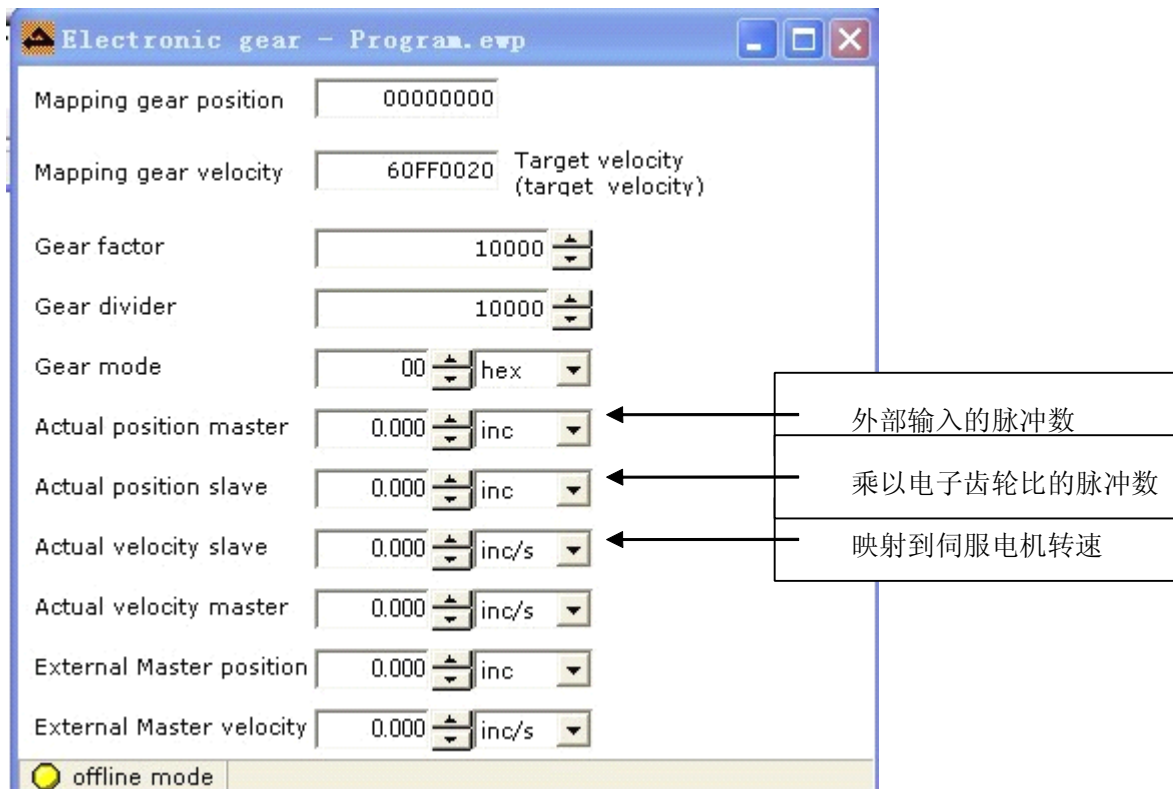
4) 如果在使用时驱动器报警：Following Error，可以设置 Axis1->Parameter

settings->Position controller->P-Gain position control =100-200, Feedforward Velocity = 100 设置好后请保存（Administration->Save parameters to device now）。如下图：



5) 监控脉冲接收

Device—>Configuration->Electronic gear-> Actual position master 一栏为驱动器实际接收到脉冲数。



6) 最后务必保存所有的参数 (Administration—>Save parameters to device now)

小提示:

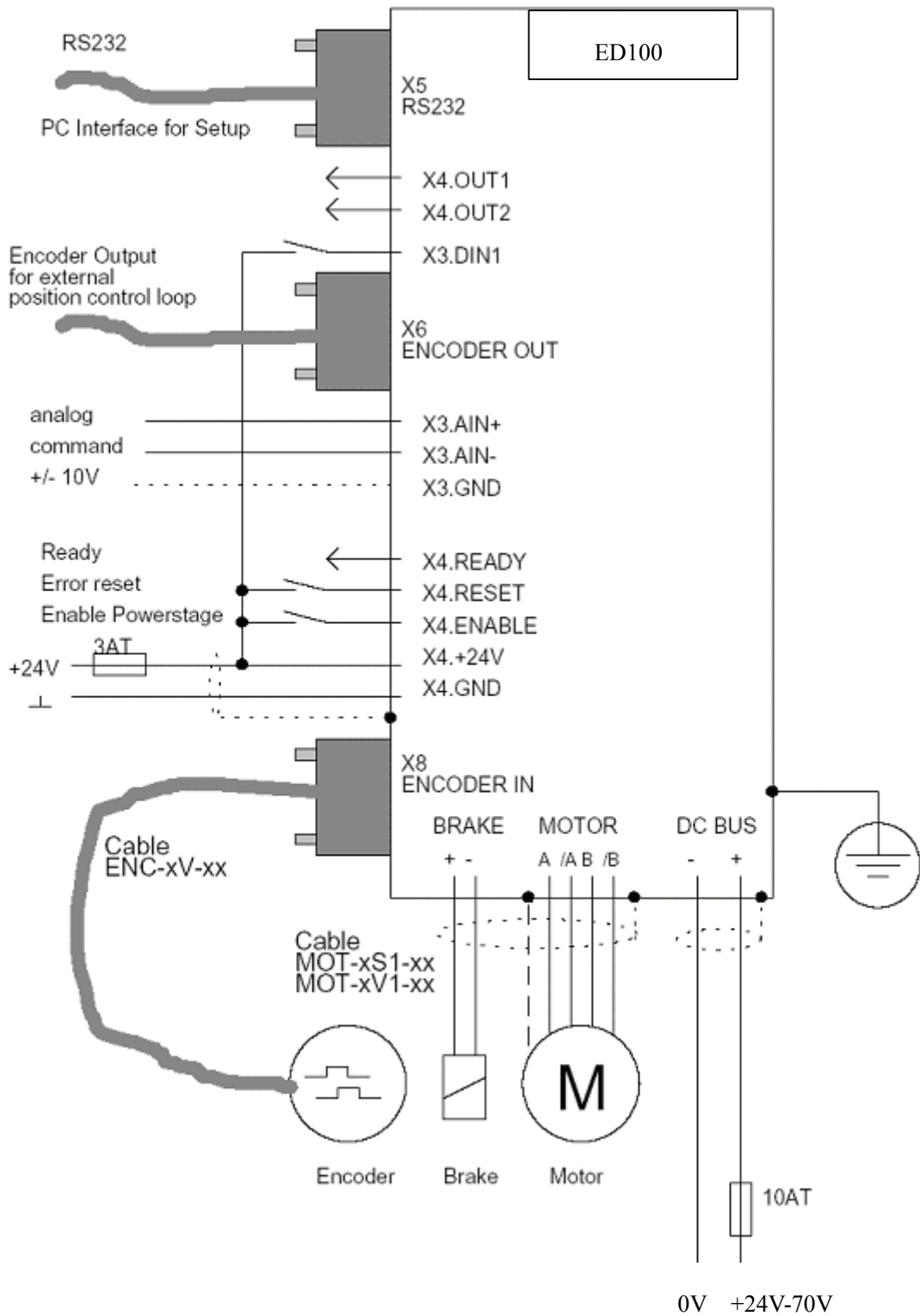
1: 如果发现外部编码器的方向和客户定义方向相反的, 即: 当编码器正方向走, 输入到伺服驱动器是负数, 就将A, /A或 B, /B其中的一对的两个线调换一下即可将输入的正负调过来。

2: ED伺服电子齿轮比是修改立刻生效的, 和普通的伺服修改后要重启才可以生效相比, 更适合多齿轮比的场合。

3: 电子齿轮比分子 (25090310) 的赋值范围是 ± 32767 , 分母 (25090410) 的赋值范围是**0~65535**, 所以电子齿轮比只可以设置在1: 32.7之内。另外ED100的电子齿轮分母不可以小于1000, ED430, ED620, ED630等没有这个限制。

二、模拟量控制 **ED** 伺服电机速度和位置

1. 接线



2. 配置 Digital input

Din1 的上升沿触发 Seq01,让驱动器 ON, 并进入模拟量控制模式。Din1 的下降沿触发 Seq02,让驱动器 Off, 电机轴松开。

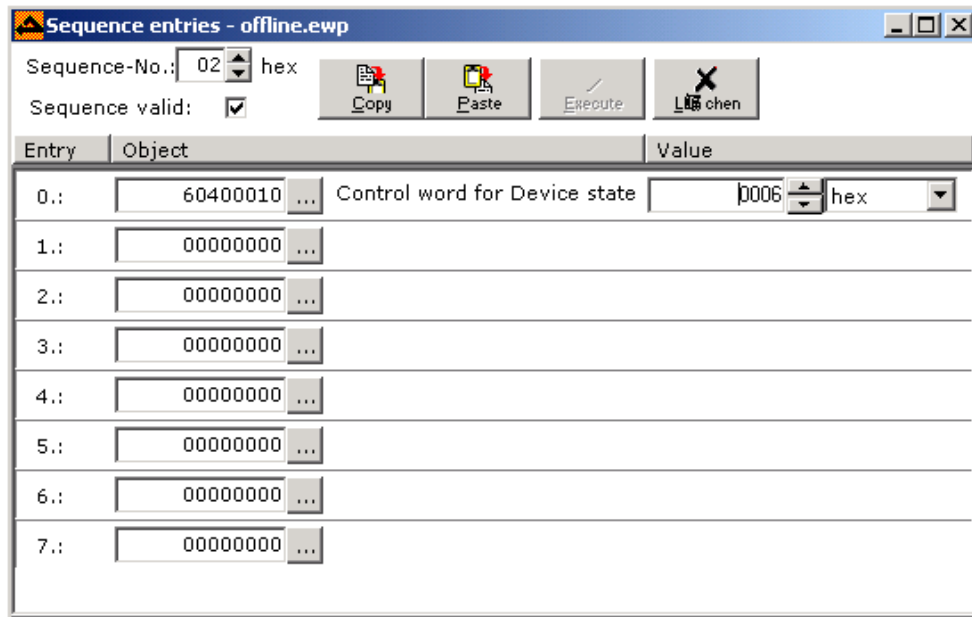
Events at DIN1-8 - offline.ewp					
Input	Active	Sequence if L->H	Active	Sequence if H->L	Status
DIN1	<input checked="" type="checkbox"/>	0001	<input checked="" type="checkbox"/>	0002	
DIN2	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN3	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN4	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN5	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN6	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN7	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	
DIN8	<input type="checkbox"/>	0000	<input type="checkbox"/>	0000	

3. 编写程序

Seq01:

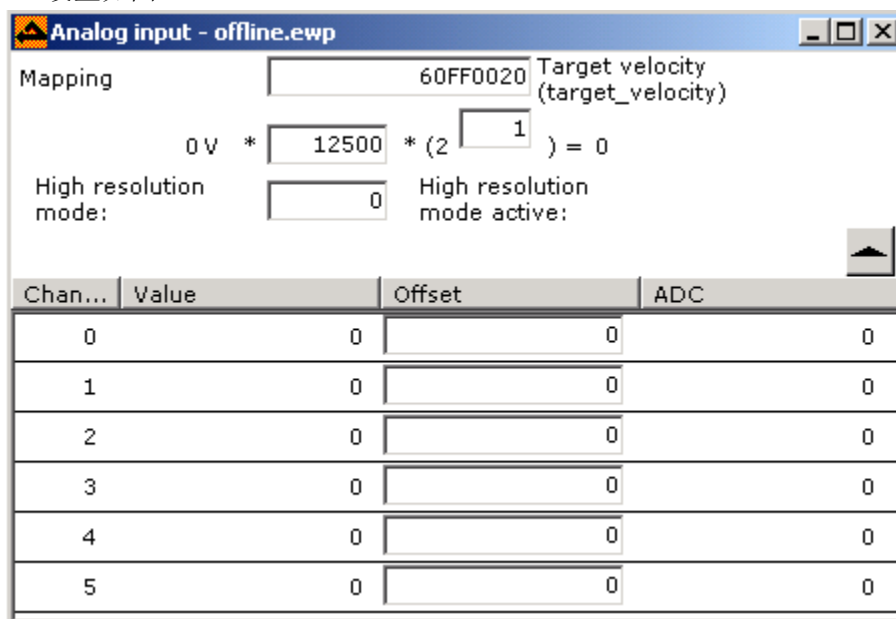
Sequence entries - offline.ewp			
Sequence-No.: 01		hex	
Sequence valid: <input checked="" type="checkbox"/>		Copy	Paste
		Execute	Cancel
Entry	Object	Value	
0.	60400010 ... Control word for Device state	000F hex	
1.	60600008 ... Chosen Mode of operation	-3	
2.	00000000 ...		
3.	00000000 ...		
4.	00000000 ...		
5.	00000000 ...		
6.	00000000 ...		
7.	00000000 ...		

Seq02:



4. 设置模拟量端口

举例：假设输入 $\pm 10V$ 对应最大正反转 1500RPM。根据 5.6.5 一节中的计算公式，设定 $shift=1$ ，则 $factor=12500$ 。设置如图：



如果将映射对象改为目标位置 607A0020,将控制模式改为 1，就可以实现用模拟量控制位置。此时相当于位置模式下的目标位置接受模拟量控制，其具体值就是由上面速度换算公式同样换算。

5. 保存参数

在完成上面的所有操作后，务必保存（Administration->Save）。

附录 E： 全闭环控制

全闭环控制常用于高精度的定位系统，由于从伺服电机轴到负载往往都有一套连接结构，例如直线导轨，所以从伺服电机轴反馈回来的编码器信号就不能代表负载的位移和速度信号，如果系统对定位的要求精度不高，这种应用方法通过伺服的半闭环控制就可以完成，但是，在高精度的定位要求下，如类似于 μm 级别精度要求时，就只有采用全闭环的控制方式，当然，对整个系统的机械结构的设计和装配要求也就有了相应的提高。传统的全闭环控制，都采用将负载端的反馈信号接入到控制器端，如高端的运动控制卡，而 kinco 伺服驱动器可以采用一种特殊的 Firmware 来实现全闭环的控制功能。

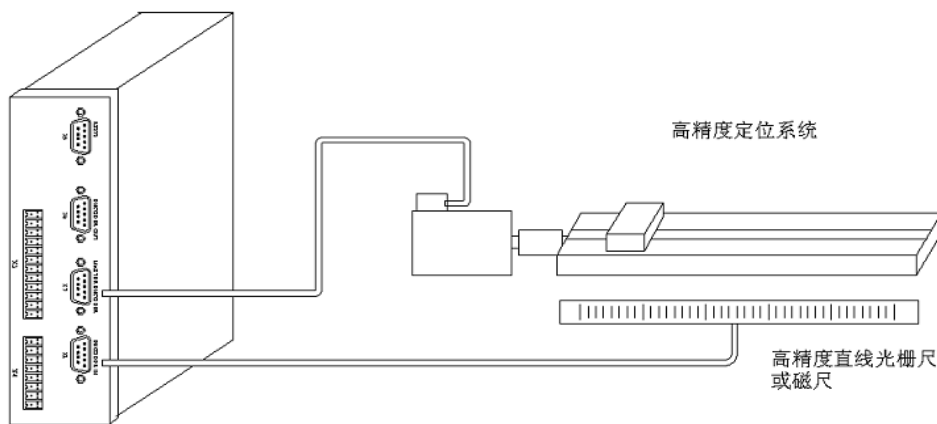


图 E. 1 全闭环控制

接线：可以采用全闭环控制的伺服驱动器包括 Kinco 全系列的伺服驱动器，不同于非全闭环控制，伺服电机的编码器电缆不是连接到驱动器的 X8 (Encoder in)口，而是连接到 X7 (Master encoder)口，如图，负载端的反馈信号直接连接到 X8 (Encoder in)口，这点一定要注意！

Firmware 区别：00010040.E10，直线编码器的反馈只作用于位置环，适合于丝杠传到的刚性较好的系统。而 00020040.E10,直线编码器的反馈作用于速度环和位置环，适合于向同步带之类的弹性系统。

Firmware 导入：利用 Eco2flash 软件直接导入全闭环的 Firmware，导入后务必保存并重新启动系统。

极性设置：电气连线和 Firmware 正确导入后，首先要在 ECO2WIN 软件中调整直线编码器的极性，否则控制环无法正常工作。利用“Direct object entry”直接设置“607E00”对象，Bit6 (0x40) 用于调整直线编码器和电机编码器的关系。首先，设置 607E00=0，手动逆时针旋转电机轴，观察 position_actual_value (在“Device control-> Movement”页 Actual position)值必须朝正计数方向变化，否则将 607E00=0x40。在确定了 bit6 的值后，就可以通过改变 bit7 的值（此时 bit6 的值不允许再改变）来改变“position_actual_value”的计数方向。

PID 参数设置：

00010040.E10：在进行其他参数设置时，首先必须将 max_current (“parameter settings->current controller”) 和 vc_kp (“parameter settings->velocity controller->p-gain velocity control”)的值调小，尤其要注意

的是速度环的比例增益(vc_kp)，该参数决定于直线编码器的分辨率，例如直线光栅尺的分辨率为 5um，那 vc_kp=5。

00020040.E10：在进行其他参数设置时，首先必须将 max_current 调小。同样也要注意 vc_kp 的设置，该参数值依赖于电机编码器的分辨率。而位置环的比例增益受 [直线编码器分辨率]/[电机编码器]值影响。例如：一个 10mm/转的丝杆系统，电机的编码器分辨率=10000um/8000=1.25um，使用 1um 分辨率的直线光栅尺时，上面的比值 = 1.25um/1um=1.25，位置环的比例增益就要参考这个值设置。

附录 F：利用 ECO2LOAD 软件导入和导出数据

我们有一个专用软件 ECO2LOAD 可以对 Kinco 伺服驱动器进行导入和导出数据操作，运行后界面如下：

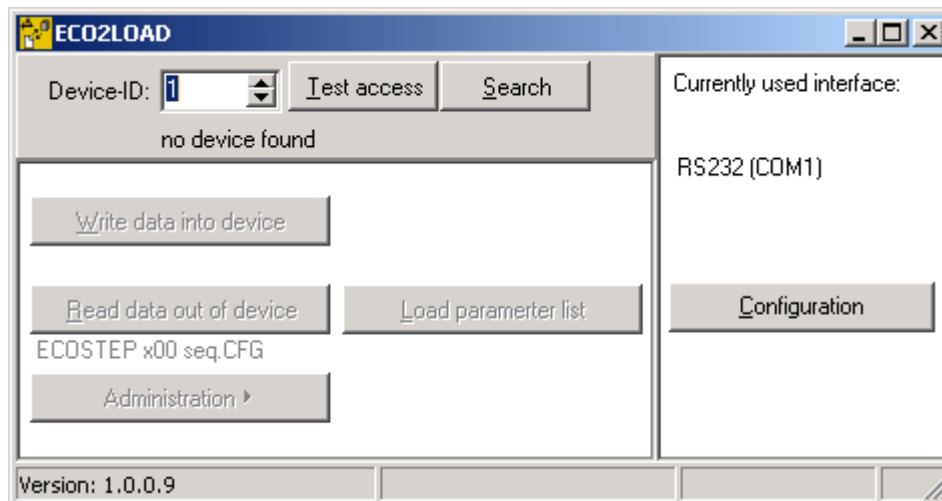


图 F.1

Device-ID 窗口内的数据为默认的驱动器的 ID 码，下面显示 “no device found”，首先要寻找到设备，通过右边的 “Test access” 按钮判断目前 PC 是否正常连接到驱动器上，如果寻找到驱动器，将在下面显示 “device found”。如果没有该提示，可以通过 “Search” 按钮自动搜寻，直到找到为止。该软件支持 RS232、CANopen 和 USB，每种通讯参数又可以单独设置，可以点击 Configuration 按钮，在弹出的窗口内进行设置，如图 F. 2。当正常找到驱动器后，界面显示如图 F. 3：

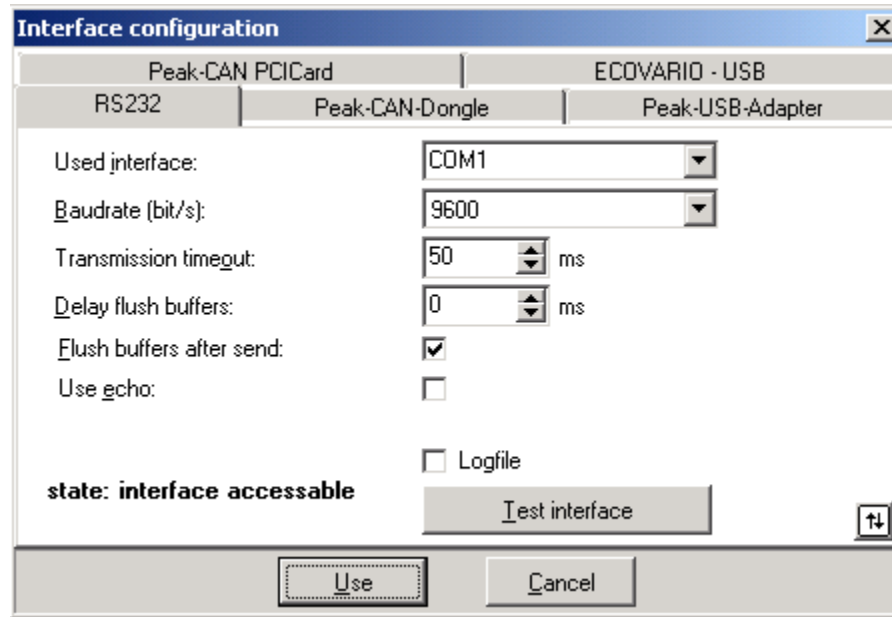


图 F.2

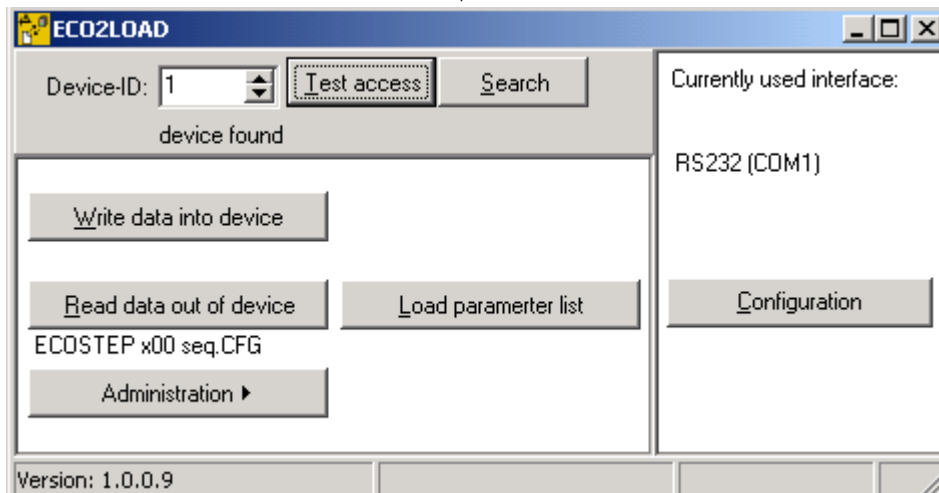


图 F.3

通讯正常建立后，就可以进行数据的导入和导出的操作了。首先需要选择导入和导出数据的范围，也就是要选择需要对哪个部分的数据进行导入和导出。该软件提供对“comm”、“seq”、“dev”、“joy”和“all”5 种不同部分的参数进行读写操作，这 5 种不同参数依次为：通讯参数、sequence 程序、设备配置、手柄设置参数和所有参数，点击“Load parameter list”按钮，分别选择不同的配置文件，例如要对“seq”进行操作，那可以选择“ECOSTEP x00 seq.CFG”文件。在完成了这些操作后，就可以利用“Read data out of device”读取驱动器内部数据。而“Write data into device”主要是将前期导出的数据文件再次导入到驱动器内部。

该软件除具有导入和导出功能外，还具有重新引导（启动），保存，初始化参数的功能，通过“Administration”按钮即可分别完成这些操作。

附录 G：程序段密码保护功能

ED 伺服驱动器提供一个 32 位数据地址供用户设置密码，从而保护伺服内部 Sequence 程序段。密码保护后

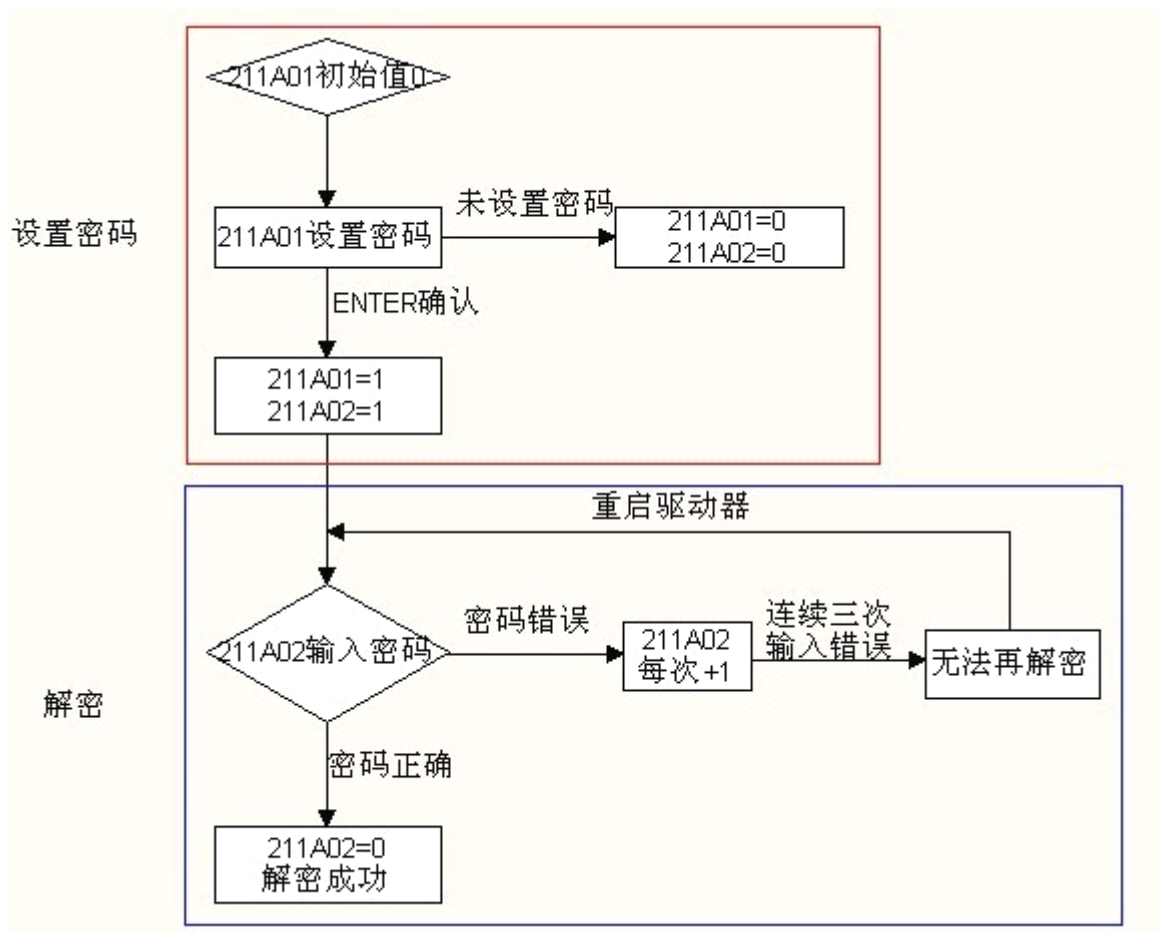
Sequence 程序段中所有内容无法显示，也不能再进行编程！即使完全上载该驱动器所有参数后再下载给另外驱动器，程序也失去作用！详细操作如下：

1:ECO2WIN 软件连接驱动器上线后，然后选中驱动器图标，点右键，选择 Direct object editor，然后选择参数 211A 01 这个参数！

2: 在未设置密码保护功能时候，211A01 默认值为 0，输入您想要的密码（最大设置 4294967295），例如 1234,按回车确定后，改值自动恢复为 1。此时到 sequence programming-sequence 程序段中会看到所有内容无法显示，也不能再进行编程！即受到密码保护！（设定密码和解密过程请参考示意图）

3: 设置密码后，如果需要解开密码查看程序段内容，则选择 Direct object editor，然后选择参数 211A 02 这个参数！如果您已经设置了密码保护，该值默认为 1（没有设置密码保护则为 0），此时如果您输入正确设置的密码，该值自动恢复为 0！如果输入的是错误的密码，该值每次自动增加 1，输入 3 次错误密码后，即使输入正确密码也不能再解密！需重启驱动器重新设置！（设定密码和解密过程请参考示意图）

4: 如果忘记密码则只能通过 Administration 中的 Initalize 初始化驱动器，然后点击 Reboot 重启驱动器！然后重新下载未受密码保护程序才能重新看到程序段内容。所以请谨慎保存您未受密码保护的源程序（**不要把受密码保护后的程序作为程序备份存档**），否则忘记密码后无法恢复到源程序。



附录 H：报警信息原因及排除

故障代码 (260002 的值)	报警信息	可能原因	处理措施
0x00001	Intern (H8SWD)	驱动器内部错误	返厂维修
0x00002	Intern (REGLERWD)	驱动器内部错误	返厂维修
0x00004	Antivalence-encoder 编码器故障	1: 编码器未正确连接到驱动器 2: 编码器插头接触不良 3: 编码器电缆断线 4: 电机编码器损坏 5: 驱动器编码器口损坏	1: 正确连接编码器电缆到“Encoder in”口 2: 检查插头 3: 按选型手册测量线缆通断, 重新焊接维修 4: 电机返厂维修 5: 返厂维修
0x00008	encoder counting 编码器计数错误	1: 编码器受到干扰 2: 电机编码器损坏	1: 排除干扰(接地, 动力线的屏蔽网接到驱动器外壳, 加滤波器, 使用屏蔽双绞线, 使用磁环等) 2: 电机返厂维修
0x00010	Masterencoder 主编码器计数错误	1: 外部编码器信号不被驱动器支持 2: 外部编码器口接线错误 3: 驱动器接口损坏	1: 更换外部编码器为 5V 差分信号 2: 按选型手册正确接线 3: 返厂维修
0x00020	drive temperature 驱动器温度过高	1: 负载过大或驱动器功率过小 2: 外部环境温度过高或通风散热不好 3: 驱动器风扇或内部问题	1: 减小负载或选择大功率驱动器 2: 处理好通风散热 3: 返厂维修
0x00040	logic voltage 驱动器逻辑电压低	1: 24V 电源电压过低 2: 逻辑电源线未正确连接 3: 驱动器内部问题	1: 检查电源 2: 正确连接 3: 返厂维修
0x00080	bus overvoltage 驱动器总线电压过高	1: 动力电电源电压过高 2: 高速停止场合反馈能量过	1: 更换电源 2: 加制动电阻

		高 3: 驱动器内部问题	3: 返厂维修
0x00100	bus undervoltage 驱动器总线电压过低	1: 动力电源电压过低 2: 先上控制电, 后上动力电 3: 急速启动 4: 驱动器内部问题	1: 检查动力电源 2: 先上动力电, 后上控制电 3: 减小加速度 4: 返厂维修
0x00200	short circuit phase A A相短路	1: 电机动力线短路 2: 驱动器内部问题	1: 检查电缆线和电机 2: 返厂维修
0x00400	short circuit phase B B相短路		
0x00800	short circuit at outputs READY, OUT1, OUT2 or BRAKE 输出短路	1: 未正确接线输出短路 2: 输出口受到干扰误报警 3: 驱动器内部问题	1: 正确接线 2: 排除干扰(加反向二极管, 加中间继电器转换, 检查 24V 电源等) 3: 返厂维修
0x01000	External Enable low 外部使能低	1: ENABLE 端子未正确连接 或使能输入断开 2: 驱动器内部问题	1: 正确连接, 测量使能输入电平信号 2: 返厂维修
0x02000	following error 跟随误差过大	1: 驱动器控制环参数设置不当 2: 负载过大或者卡死 3: 编码器反馈信号问题 4: 驱动器内部问题	1: 设置合适的控制参数。位置反馈设为 100%, 适当增大 MAX.following error 及 KPP 等 2: 调整负载或者空载运行监测, 选择更大功率电机 3: 电机返厂维修 4: 返厂维修
0x04000	overspeed 速度过高	1: 默认的 MAX.velocity 被修改 2: 驱动器内部问题	1: 修改为正确的值 2: 返厂维修

0x08000	commutation 找励磁错误	1: 默认的找励磁参数设置不当 2: 动力电缆线序未正确连接 3: 电机编码器信号反馈错误 4: 驱动器内部问题	1: 修改为正确的参数 2: 按选型手册正确连接 3: 检查电缆等, 电机返厂 4: 返厂维修
0x10000	bus fault 总线通讯错误	1: CAN 总线通讯错误 2: 驱动器内部问题	1: 正确连接通讯电缆, 正确设置好通讯参数, 排除通讯干扰 2: 返厂维修
0x20000	i ² × t 电流过载	1: 驱动器控制环参数设置不当。系统震荡 2: 负载过大或者卡死 3: 驱动器内部问题	1: 设置合适的控制参数。如 KVP 设置, 电流环参数等 2: 调整负载或者空载运行监测, 选择更大功率电机 3: 返厂维修
0x40000	Nlock 负限位故障	1: 插补模式下负限位故障 2: 驱动器内部问题	1: 检查限位 2: 返厂维修
0x80000	plock 正限位故障	1: 插补模式下正限位故障 2: 驱动器内部问题	1: 检查限位 2: 返厂维修

注意:

1: 有些时候, 有多个报警同时发生。例如 **A** 相短路和 **B** 相短路经常一起发生, 此时 **260002** 报警代码为两者 (**0x00200** 和 **0x004000**) 的总和 **0x00600**

2: 故障检测可以保证系统安全工作, 如果屏蔽某位故障检测, 当发生这种故障时, 系统将不会自动切断, 从而无法保证系统的安全。

附录 I：常见参数单位换算表

ED 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 inc/s，内部单位是 dec，两者关系是 $1\text{inc/s}=64\text{dec}$ ！假设需要速度为 60rpm，也就是 10000inc/s（编码器分辨率 10000），那么用通讯控制时需写入速度为 640000dec！下表列出常用需要换算单位的参数表！

参数名称	工程单位	内部单位	换算关系
速度	inc/s	DEC	1 inc/s=64dec
加速度	inc/s*s	DEC	16 inc/s*s=1dec
KPP(p -gain Position control)	1/S	DEC	1 1/S=64dec
位置前馈 Feedforward Velocity	%	DEC	1 1%=164dec
电流	A	DEC	驱动器峰值电流=2047dec ED100 峰值电流为 8 A ED430 峰值电流为 12A ED620 峰值电流为 12A ED630-0250 峰值电流为 18A ED630-0300 峰值电流为 21A ED630-0400 峰值电流为 24A