Kinco 上海步科电气有限公司
Kinco Electric（Shanghai）Ltd.

KINCO® ED Series Servo Drives

User Manual

March 2006

Contents

Chapter 1 Functions and Composition of ED Series Servo Drives

1.1 Functions

As domestic models for ECOSTEP series, the ED series servo systems are a kind of fully-digital intelligent servo system among all Kinco® servo systems. This user manual is also applicable to the programming of ECOSTEP, ECOLIN, and ECOVARIO drives. For detailed drive parameter settings, refer to their respective hardware manuals. The ED series servo systems support fully-digital control over speed, position, and torque. The drive is integrated with such functions as drive, control, bus, and I/O. The drive can store up to 256 sequences of programs and each sequence can store one motion curve, including such important motion curve parameters as acceleration, deceleration, maximum speed, and target position. It can call a maximum of 128 motion curves directly through the eight I/O ports. With the communication function of ED, an external controller can call the 256 sequences of programs stored in the ED to modify internal system parameters of the drive. For a master/slave control system or a simple synchronization system, the implementation is simple and made possible when you input the external master encoder signals into the X7 port of the drive, without the need of any other complicated software or hardware settings.

The ED series servo systems support multiple communication modes, including RS232C, RS485, CANopen, and Profibus DP. With open protocols provided by us, a PLC, a PC, a single-ship microcomputer or other controller can complete the setting of internal parameters of the ED series servo drives and the control over motion curves freely. For simple application of direct connection between the touch screen and the servo system, the ED series servo drives can be directly connected with the eView touch screen. The user can then enter control parameters through the eView touch screen of text display. Meanwhile, the servo-related parameters and real-time data can be displayed. This application is suitable for the occasion requiring only simple I/O port control and logic control. In addition, this solution does not require a controller such as a PLC, a PC, or a single-ship microcomputer, which is necessary in a conventional solution, and thus saving the cost for the user.

The ED series servo drives support three types of Kinco® motors: multipolar, high-speed, and linear motors. Because multi-polar motors have a large number of poles (50), and the current switchover speed is high when the motor runs at a high speed (greater than 1000RPM) and the back electromotive force of the motor winding is greater, the torque of the motor will reduce when it runs at a high speed. However, the motor has good quick start/stop performance and is suitable for application environments of frequent start/stop and considerable load inertia to take the place of ordinary AC servo motors. High-speed motors are servo motors with less polar logarithm. The performance of these motors will not be compromised when they run at a high speed. They can deliver constant-torque output at a rated rotation speed around 3000 RPM. Since linear motors have simplified the mechanical structure, they deliver higher precision and quicker response speed.

Basic functions:
- Fully-digital speed, position, and torque control
- Monitoring and alarm function for such faults as short circuit, overvoltage/undervoltage, over-temperature, encoder fault, following error, and I^2*t
- Providing communication options RS-232, RS-485 or CANopen, Profibus DP
- Analog monitoring and control of internal objects like speed and torque

- Intelligent brake control
- Up to 27 homing modes
- Enabling drive and fault reset by external input
- Providing Ready state by digital output

Other functions:
- Supporting master/slave control (electronic gear ratio settable, direct connection of external encoder signal with X7: MASTER ENCODER interface)
- Storing up to 256 motion curves, with position, speed, acceleration, deceleration, and working mode definable for each curve
- 8 input ports to configure and call different programs
- Two programmable digital output interfaces used for signal feedback, like "Target reached" signal and fault alarm signal
- Compatible with traditional servo control modes and supporting Pulse/Dir and CW/CCW control modes
- Full closed loop control

1.2 Composition

A complete set of ED servo system is composed of an ED drive, a Kinco® servo motor, motor cable, encoder cable, four terminal blocks (configured with ED drive), PDC power supply (optional) hardware and the software ECO2WIN.

You may request Kinco servo drive CD from our sales engineers or our agents. The CD contains the electronic version of this user manual, the latest version of the Eco2win software, and related examples. You can also download the latest software and related documents on our website at www.kinco.cn.

Please check whether the motor and drive models conform to the order and check whether the goods are complete against the packing list. If you have any question, please contact us. If you have questions during operation, please contact Kinco® servo technical support engineers at 86-755-26585555 (telephone exchange in Shenzhen). At present, we have servo engineers ready for your service in Shenzhen, Shanghai, and Beijing. For detailed contact methods, please call 86-755-26585555 (telephone exchange in Shenzhen) or visit our website at www.kinco.cn.

Chapter 2 Installation and Use

2.1 Operating Environment

- Operating temperature: 0 ~ 40°C
- Storage temperature: -10°C ~ 70°C
- Operating humidity (non-condensing): 5 ~ 95% (RH-2, according to IEC 61131-2)
- Pollution degree: 2 (according to IEC 61131-2)
- Protection class: IP20
- Installation place: Dust-free, dry, and lockable (for example, an electrical cabinet)
- Installation position: Vertical (refer to ED Series Drive Installation Manual)
- Installation Height: Up to 1000m above sea level (full range)
- Power loss:
  @5.7Arms (UDC-bus<=170V)   40W, Note: the loss <40W when supply voltage < 170VDC
  and rated current = 5.7A
  @7.5Arms (UDC-bus<=170V)   40W, Note: the loss <40W when supply voltage < 170VDC
  and rated current = 5.7A
  @8.5Arms (UDC-bus<=170V)   70W, Note: the loss <70W when supply voltage < 170VDC
  and rated current = 5.7A
- Cooling mode: Convection; heat sink required if continuous power loss >40W

2.2 Installation Clearance and Direction

Install the device as per the clearance specified in Section 3.4.3 EMC Installation. The ED100 drive adopts horizontal installation. The ED200/ED216 drive adopts the vertical or horizontal installation.

2.3 Preventing Foreign Substances

1. The device must be installed in the control cabinet.
2. Prevent metal filings, metal wire ends or other foreign substances from entering the servo drive when installing the control cabinet.
3. Prevent oil, water, metal dust and other foreign substances from entering the servo drive.
4. Implement forced ventilation with the clean air for the control cabinet in places where hazardous gas or considerable dust exists to prevent such substances from entering the drive.

2.4 Installation of Encoder Cable

An encoder cable has a length limit. To ensure long-term reliable communication of cables, pay attention to the following points:
1. Check carefully the cabling method of cables. Do not excessively bend the cable at the connection part nor let the connection part bear the cable weight.

2. If the servo cable is to be moved during operation, pay attention to the load and bending of the cable. Especially, the extension cables must have good bend resistance. The servo motor cables and encoder cables provided by KINCO have undergone strict bending test and can be used reliably. If the servo motor is to be moved during operation, the encoder and motor cables shall be moved as well, ensuring that the strength and bending of cables are within permissible ranges. In this application occasion, a tow chain structure is recommended for the installation of motor cables and encoder cables.

2.5 Safety Instructions

2.5.1 General

All transportation, storage, installation, connection and service work must only be carried out by qualified service-trained professionals taking into account the national and regional safety and accident prevention regulations. The qualified skilled service-trained professionals are persons who have a relevant vocational qualification and proper knowledge of the execution of the above-mentioned work. Read the documentation carefully before installation and commissioning. Serious injury to persons and damage to device may result through improper use of the device or incorrect installation or operation. All technical specifications and conditions in this manual and related technical manuals (refer to technical specifications in the model selection manual) must be adhered to in all cases.
The device contains electrostatic sensitive components, which may be damaged by improper handling. Discharge yourself before you touch the device.   Avoid contact to high isolating materials (e.g. synthetic fibers, plastic films, etc.). The motor and the main power connectors may be energized even when the motor does not move. All electrical connectors of the device must neither be plugged in nor disconnected under voltage since electric arcs may damage persons and connectors. Do not touch components which are energized during operation immediately after disconnection from supply. Wait after disconnection of the device from the power supply at least 6 minutes before you touch components which are energized during operation.

2.5.2 Operating Environment

The drive is intended for use in industrial equipment and meets the requirements of the Low Voltage Directive 73/23/EEC. All technical specifications and permissible conditions mentioned in the documentation must be adhered to in all cases. The following are prohibited, unless expressly provided otherwise:
1. use in hazardous areas
2. use near oils, acids, gases, vapors, dust, radiation, etc.

2.5.3 Requirements for Mounting and Installation

Upon mounting and installation, note the following points:
● ensure that the equipment is installed on a suitable, low-vibration substructure or in an electric cabinet,
● allow good ventilation,

- make sure there is sufficient clearance between adjacent components,
- be careful when handling components to avoid injury to persons or equipment. Do not touch, bend or damage electronic components and contacts; do not change insulation layers.

2.5.4 Preconditions for Electric Connection

All installation instructions mentioned in the documentation must be considered and all work must be carried out.

Before electrical connection, ensure that
- the electric components are disconnected from the power supply and prevented from being switched on accidentally,
- you have double-checked that the equipment is de-energized;
- you have ensured that any additional monitoring and protective devices are properly installed.
- When connecting the equipment to the power supply, ensure that
- the applicable standards and regulations are complied with,
- the power connections are safe and secured,
- the EMC-compliant installation (e.g. separate arrangement of earth wires, filters, communication cables, and power cables) is realized.

2.5.5 Operation

Do not disable monitoring and protective devices during the trial run. Covers, electric cabinet doors etc. must always be closed during operation. Contact the manufacturer or agent if necessary.

Chapter 3 Interface Signal and Wiring

3.1 Models of ED Series Drives

Based on the maximum output power and communication ports provided, the ED series drives consist of the following models:

Model	Max. Output Power	RS232	CAN	Profibus	RS485	12-bit DAC
ED100-AA-000-000	560W	Yes	Yes			
ED100-LA-000-000	560W	Yes			Yes	
ED100-PA-000-000	560W	Yes		Yes		
ED200-AA-000-000	1800W	Yes	Yes		Yes	
ED200-ZA-000-000	1800W	Yes	Yes		Yes	Yes
ED200-PA-000-000	1800W	Yes		Yes		
ED200-QA-000-000	1800W	Yes		Yes		Yes
ED216-AA-000-000	3600W	Yes	Yes		Yes	
ED216-ZA-000-000	3600W	Yes	Yes		Yes	Yes
ED216-PA-000-000	3600W	Yes		Yes		
ED216-QA-000-000	3600W	Yes		Yes		Yes
ECOLIN200-ZA-000-000	1800W	Yes	Yes		Yes	Yes
ECOLIN200-QA-000-000	1800W	Yes		Yes		Yes
ECOLIN216-ZA-000-000	3600W	Yes	Yes		Yes	Yes
ECOLIN216-QA-000-000	3600W	Yes		Yes		Yes

Table 3.1   Model Table of Drives

Note: 12-bit DAC refers to the Digital-Analog Conversion with a resolution up to 2^{12} (=4096), which provides high-precision current control and delivers more stable motor control.

3.2 Interfaces of ED Series Drives

3.2.1 ED100 Series Drives
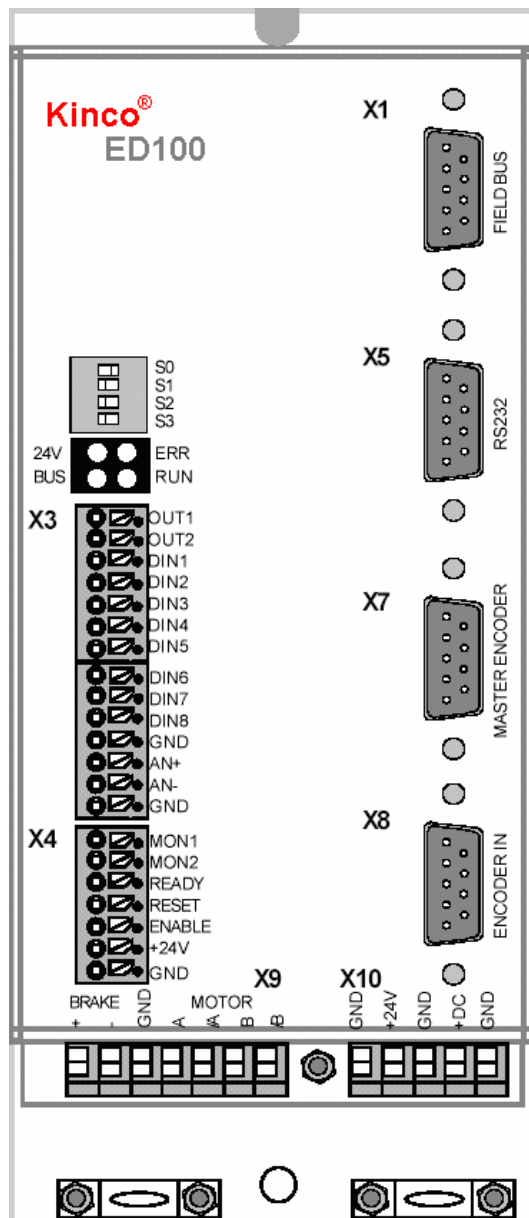
X1: Fieldbus interface, providing CANopen or RS-485 interface, 9-pin male connector

X3: Digital output/input interface

X4: Logic power (24V) input, analog monitoring output, and other functions

X5: RS-232 interface, usually used by the PC to download parameters, 9-pin female connector

X7: External master encoder signal input interface

X8: Encoder feedback signal input interface of the motor

X9: Power cable interface and brake power interface of the motor

X10: Power supply interface (input range: 24 VDC~70 VDC)

As shown in the figure on the left, the GND/+24V on the left of X10 is the same as the GND/+24V of X4; both are control logic power input interfaces of ED100; only one works at a time.

The GND/+DC on the right of X10 is the power supply interface.

Figure 3.1   Interfaces of the ED100 Drive

3.2.2 ED200/ED216/ECOLIN200/ECOLIN216 Series Drives
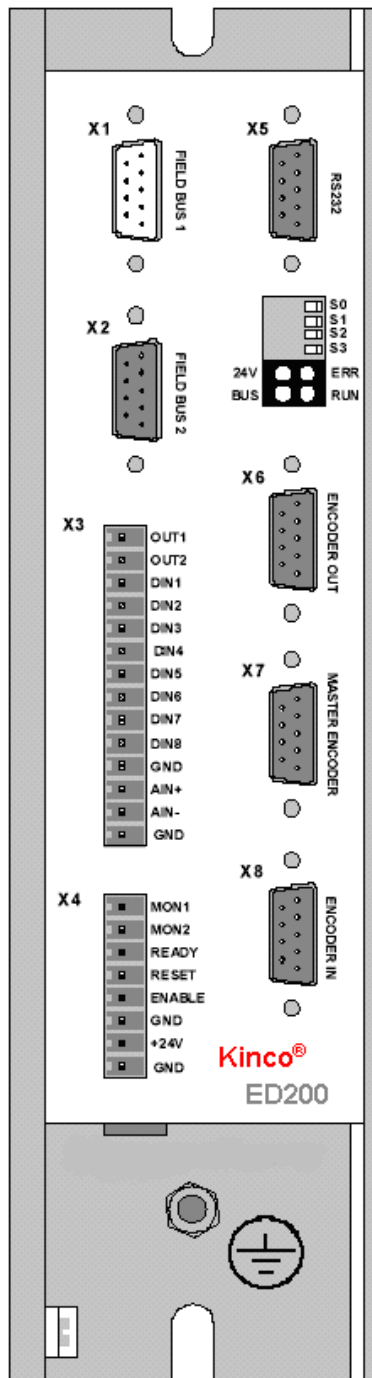
X1: CANopen bus interface, 9-pin male connector

X2: RS-485 or Profibus bus interface

X3: Digital output/input interface, analog input

X4: Logic power input interface, analog monitoring output interface, eternal enabling input interface, external fault reset input signal, Ready signal output interface

X5: RS-232 interface, used by the PC to download parameters, 9-pin female connector

X6: Synchronization signal output interface of the motor encoder

X7: External master encoder signal input interface

X8: Encoder feedback signal input interface of the motor, to the encoder cable

X9: Motor connection interface (to motor power supply cable) and brake power output interface

X10: Power supply input interface (24 VDC~170 VDC); the positive (+) and negative (-) poles of the power supply are connected to DC BUS+ and DC BUS- respectively.

Figure 3.2   Interfaces of the ED200/ED216/ECOLIN Drives

3.2.2 Functions of Drive Interfaces

Interface	Symbol	Function
X1	FIELDBUS	CANopen bus interface, 9-pin male connector. (**RS-485 interface of ED100 LA adopts this kind of interface**)
X2		RS-485 or Profibus bus interface (**ED100 does not have this interface**)
X3	OUT1/OUT2	Two customizable digital output interfaces, usually used for "Target reached" signal and alarm signal output
	Din1..Din8	Digital input interface, triggering internal program sequence by rising or falling edge or BCD code (setting high level or low level input to be valid by software)
	Din6	When it is not used as a digital input interface, it can serve as an external positive limit position signal input interface of the motor
	Din7	When it is not used as a digital input interface, it can serve as an external negative limit position signal input interface of the motor
	Din8	When it is not used as a digital input interface, it can serve as an external homing signal input interface
	AIN+/AIN-	Analog input interface, used to control motor speed and torque, with the input range of ±10V
X4	MON1/MON2	Two analog output interfaces, monitoring the status of internal objects of the drive.(Monitoring contents can be set in the software)
	READY	Digital output interface; when it outputs high level (+24V), it indicates the drive is free of fault
	RESET	Digital input interface; when the rising edge signal is inputted, it can reset drive fault
	ENABLE	Digital input interface, to be used by the external enabling drive. The drive must connect with +24V when it operates. If this input is disconnected when the drive is working, it will lead to the "External enable low" fault
	GND	Logic ground, serving as the common terminal of input/output interfaces
	+24V	Logic power "24V" input interface, to be provided externally
	GND	Logic power "24V GND" input interface, to be provided externally
X5	RS232	RS-232 interface, 9-pin female connector, usually connected with PC for programming purpose or directly connected with the eView touch screen or text display, or connected with other controllers for communication with the ED drive
X6	ENCODER OUT	Motor encoder output interface; output motor encoder signal as the slave encoder signal input in the master/slave control mode (**ED100 does not provide this interface**)
X7	MASTER ENCODER	Master encoder signal input interface, or programmed to set to pulse/direction signal control, CW/CCW control, electronic gear or electronic cam control
X8	ENCODER IN	Encoder signal input interface of the motor, directly connected to the servo motor encoder cable
X9	BRAKE/MOTOR	Motor cable connection interface and brake power output
X10	GND +24V	Power supply input interface (different input ranges for different types of drives)
	GND +DC	(ED100 provide this interface as logic power input interface, equal to the +24V and GND interface of X4)

	GND	ED100 24VDC~70VDC (specific to ED100 drive, provided between +DC and GND) ED200 24VDC~170VDC (specific to ED200 drive, provided between DC BUS+ and BUS-) ED216 24VDC~170VDC (specific to ED216 drive, provided between DC BUS+ and BUS-) ECOLIN200 24VDC~170VDC (specific to ECOLIN200 drive, provided between DC BUS+ and BUS-) ECOLIN216 24VDC~170VDC (specific to ECOLIN216 drive, provided between DC BUS+ and BUS-)

Table 3.2   Interface Function

3.3 Internal Wiring of Drive

3.3.1 Internal Wiring of ED100 Drive
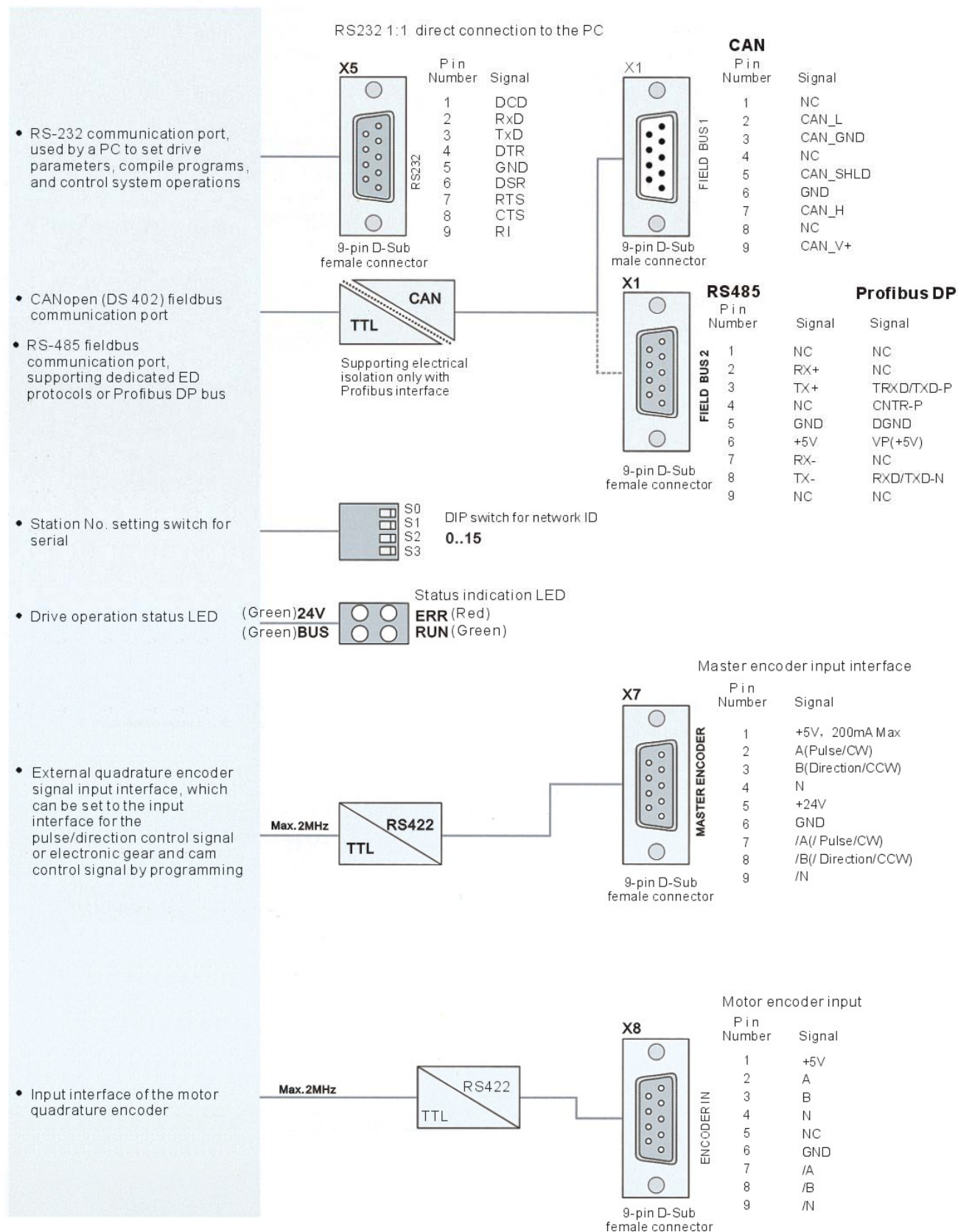
Figure 3.3 Internal Wiring of ED100 Drive (1)

▰ I/O Wiring Terminal

X3

Output 1	OUT1	
Output 2	OUT2	
Input 1	DIN1	
Input 2	DIN2	
Input 3	DIN3	
Input 4	DIN4	
Input 5	DIN5	
Positive limit switch	DIN6	
Negative limit switch	DIN7	
Homing switch	DIN8	
Signal ground	GND	
Analog input +	AIN+	
Analog input -	AIN-	
Signal ground	GND	

Phoenix MCVW 1.5/14-ST-3.81

- Programmable 24V digital output interface under operation mode
- Used to monitor motor operation range or status, as well as the digital input interface for homing control
- 10-bit resolution analog signal input interface for position and velocity control signal input

24V 0.5A

24V 1mA 1ms

-10...+10V   10K   10K   1ms

X4

Analog output 1+	MON1	
Analog output 2+	MON2	
Ready signal	READY	
Reset signal	RESET	
Enable signal	ENABLE	
Control power DC 24V＋	+24V	
Control power DC 0V	GND	

Phoenix MCVW 1.5/7-ST-3.81

- Two programmable 8-bit resolution analog signal output interfaces
- Ready signal output interface of the servo system
- Error reset signal input interface
- Enable signal input interface for drive operation

1ms

24V 0.5A

24V 1mA 1ms

24V$_{DC}$

5V$_{DO}$

▰ Motor Connection Terminal

X9

Brake power +	BRAKE+
Brake power -	BRAKE-
Ground	GND
Motor phase line A	MOTOR A
Motor phase line /A	MOTOR/A
Motor phase line B	MOTOR B
Motor phase line /B	MOTOR/B

Phoenix MCVW 2.5/7-ST

- Intelligent automatic brake control power output interface
- Provided with motor short-circuit protection function, monitoring over-voltage, under-voltage and over-temperature

60V   24V 1A

2-Phase-Power Stage

▰ Main Power Connection Terminal

X10

Control power DC 0V	GND
Control power DC +24V	+24 V
Main power DC 0V	GND
Main power +DC	+DC
Ground	GND

Phoenix MVSTEBR 2.5/5-ST

24..70V$_{DC}$

Figure 3.4 Internal Wiring of ED100 Drive (2)

3.3.2 Internal Wiring of ED200 Drive
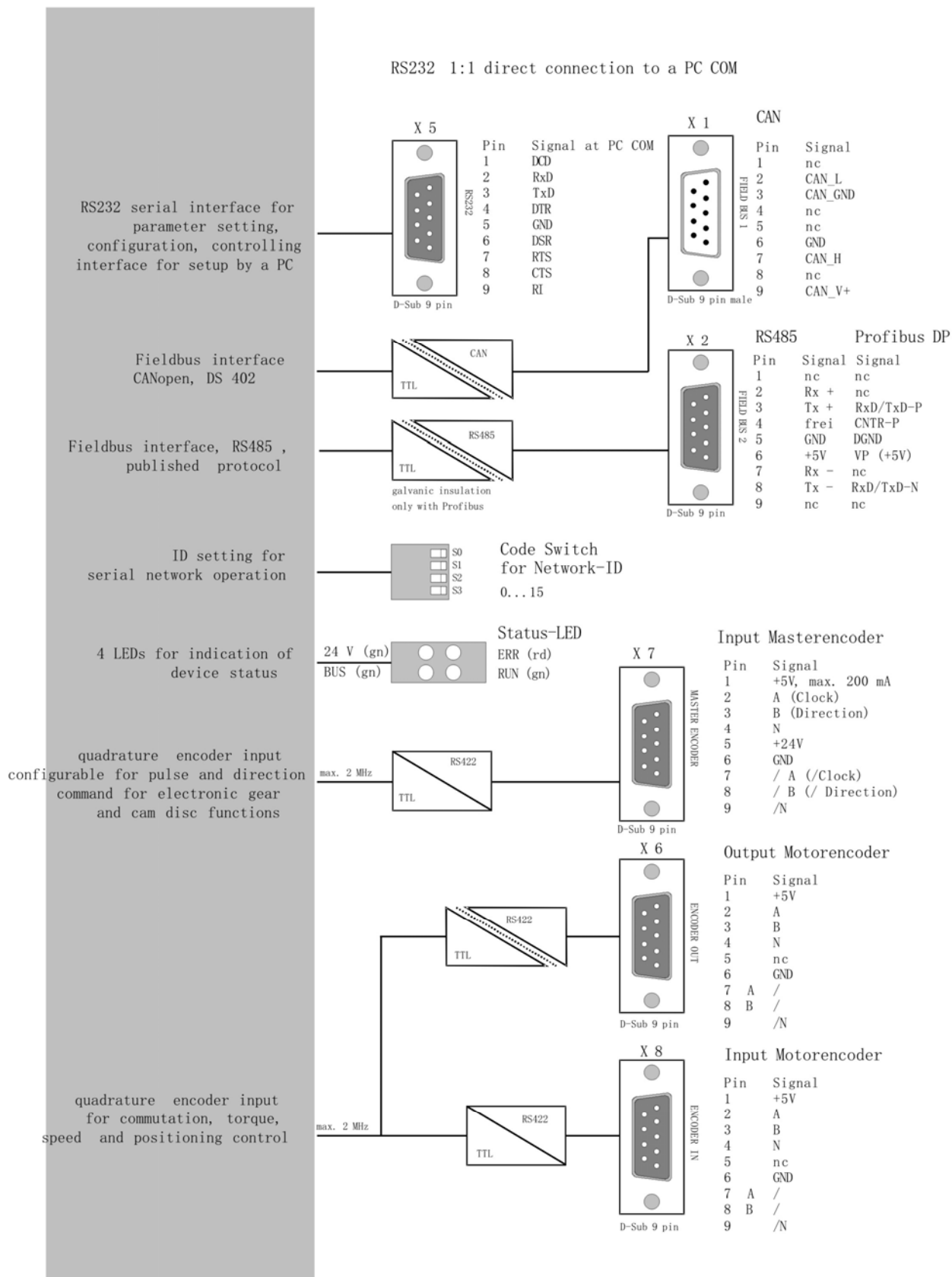
Figure 3.5 Internal Wiring of ED200 Drive (1)

Inputs / Outputs

Output 1	OUT 1	
Output 2	OUT 2	
Input 1	DIN 1	
Input 2	DIN 2	
Input 3	DIN 3	
Input 4	DIN 4	
Input 5	DIN 5	
Limit switch positive	DIN 6	
Limit switch negative	DIN 7	
Home switch	DIN 8	
Signal ground	GND	
Analog input +	AN +	
Analog input –	AN –	
Signal ground	GND	

X 3
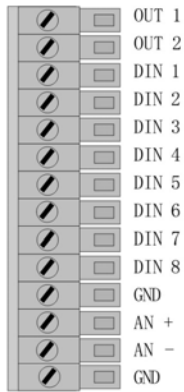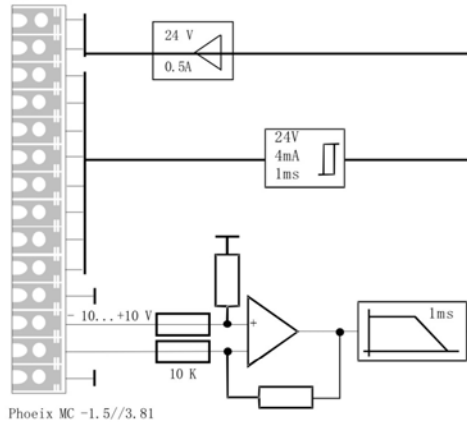
Phoeix MC -1.5//3.81

digital I/O PLC interface for running configurable motion profiles controlling/monitoring of motion ranges/ profiles, homing

analog command input for position and velocity 10 bit resolution

Monitor1	MON1	
Monitor2	MON2	
Ready	READY	
Error reset	RESET	
Enable Powerstage	ENABLE	
Signal ground	GND	
Supply+ 24V	+ 24V	
Supply ground	GND	
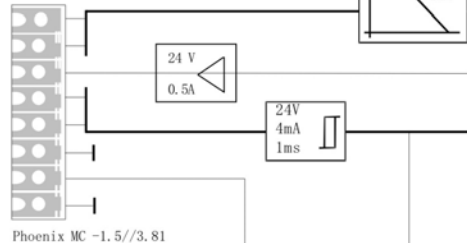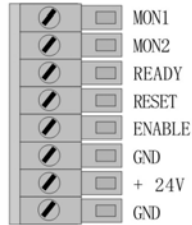
Phoenix MC -1.5//3.81

2 analog monitors scalable, configurable, 8 bit resolution

digital output ready

digital inputs: powerstage enable error reset

monitoring supply voltage, reset logic

Motor

Brake +	BRAKE +	
Brake –	BRAKE –	
	GND	
Motor Phase A	MOTOR A	
Motor Phase /A	MOTOR/A	
Motor Phase B	MOTOR B	
Motor Phase /B	MOTOR/B	

X9

Phoeix IC -2.5//5.08
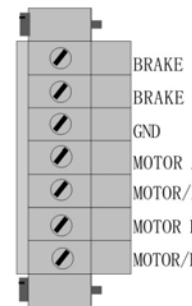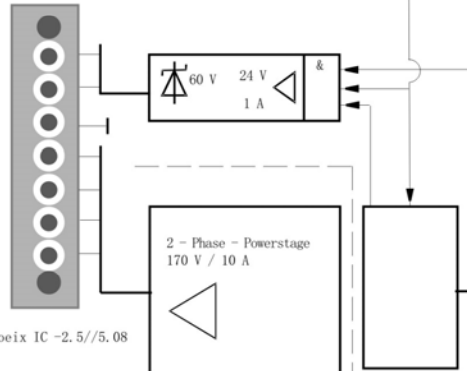
2 – Phase – Powerstage
170 V / 10 A

intelligent brake control providing automatic voltage reduction

control of the powerstage short circuit protection monitoring overvoltage, undervoltage and overtemperature

galvanic insulation

Power supply

DC Bus –	DC BUS –	
DC Bus +	DC BUS +	
	GND	

X10

24...170V_DC
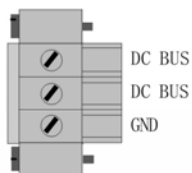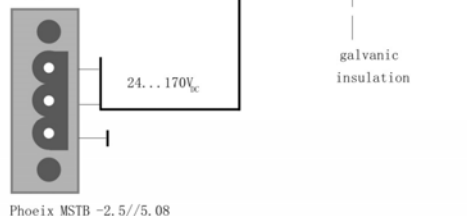
Phoeix MSTB -2.5//5.08

Figure 3.6 Internal Wiring of ED200 Drive (2)

3.4 Wiring

3.4.1 Wiring of Power Module P200 AA/BA
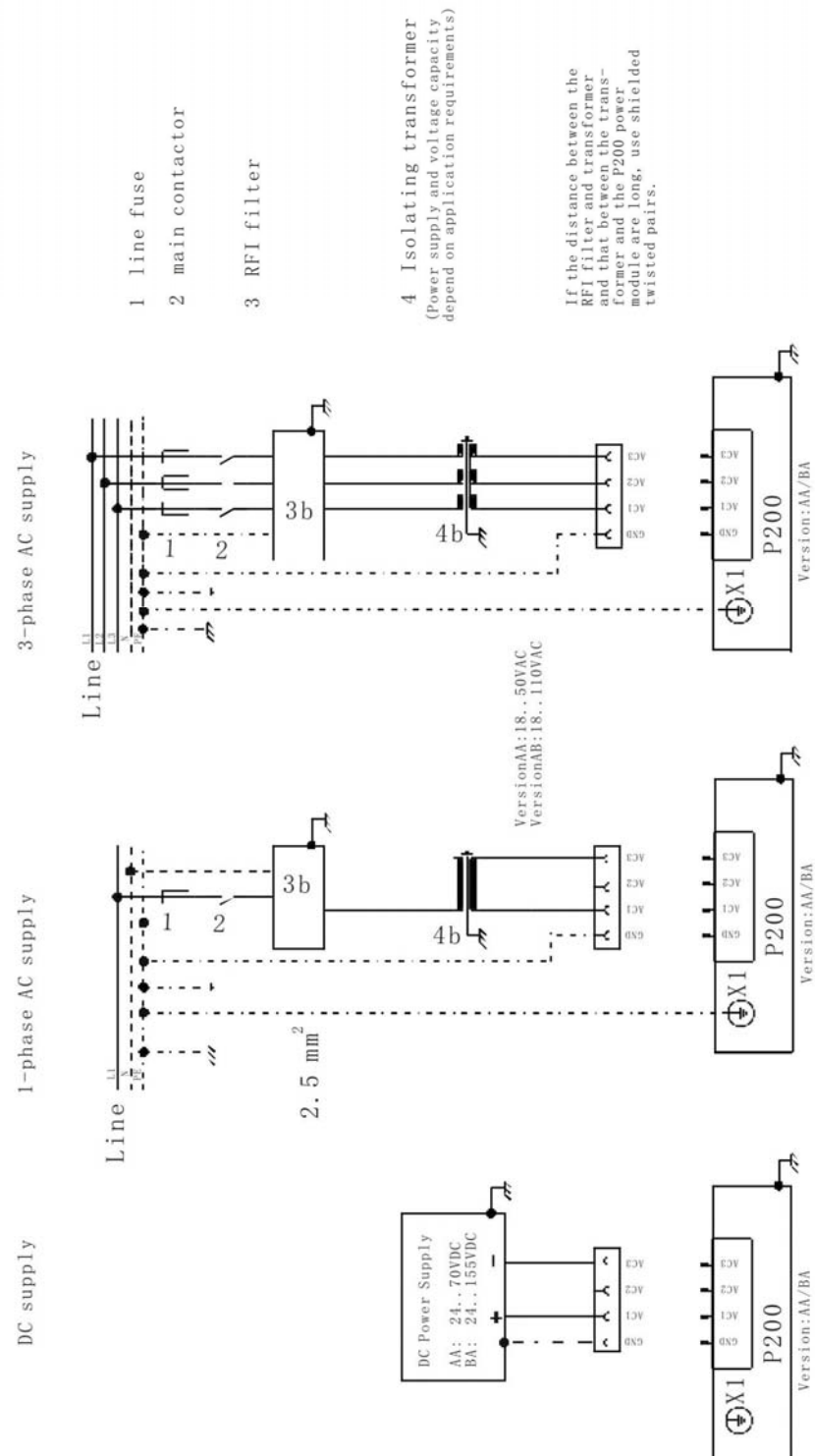
Figure 3.7   Wiring of Power Module P200 AA/BA

3.4.2 Wiring of ED Drive and Power Module P200 AA/BA
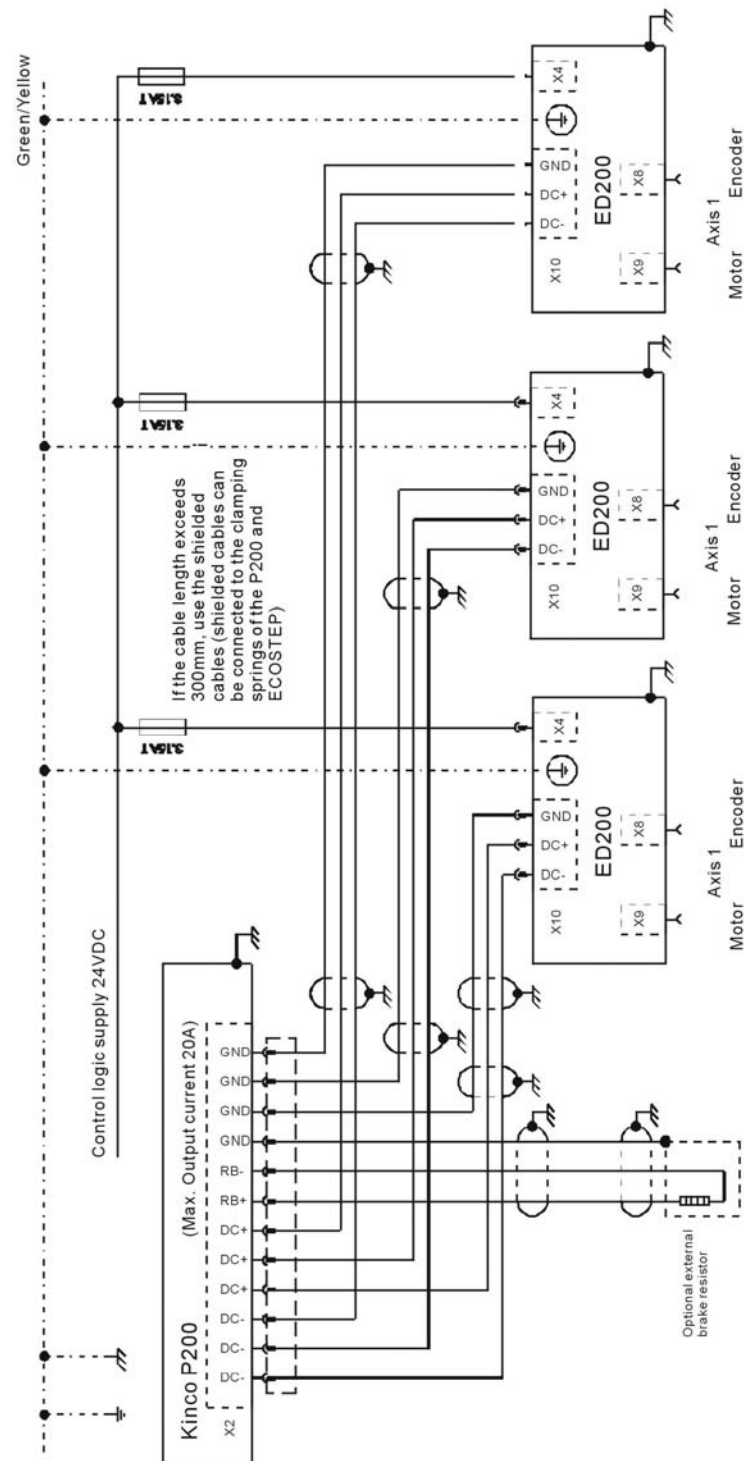
Figure 3.8 Wiring of ED200 Drive and P200

Note: The wiring diagram is also applicable to ED216, ECOLIN216 and ED100. Note that if the ED100 is used, the wiring diagram is different, and only the P200AA can be used to connect with the ED100. For details, refer to the ED100 hardware manual.

3.4.3 EMC Installation
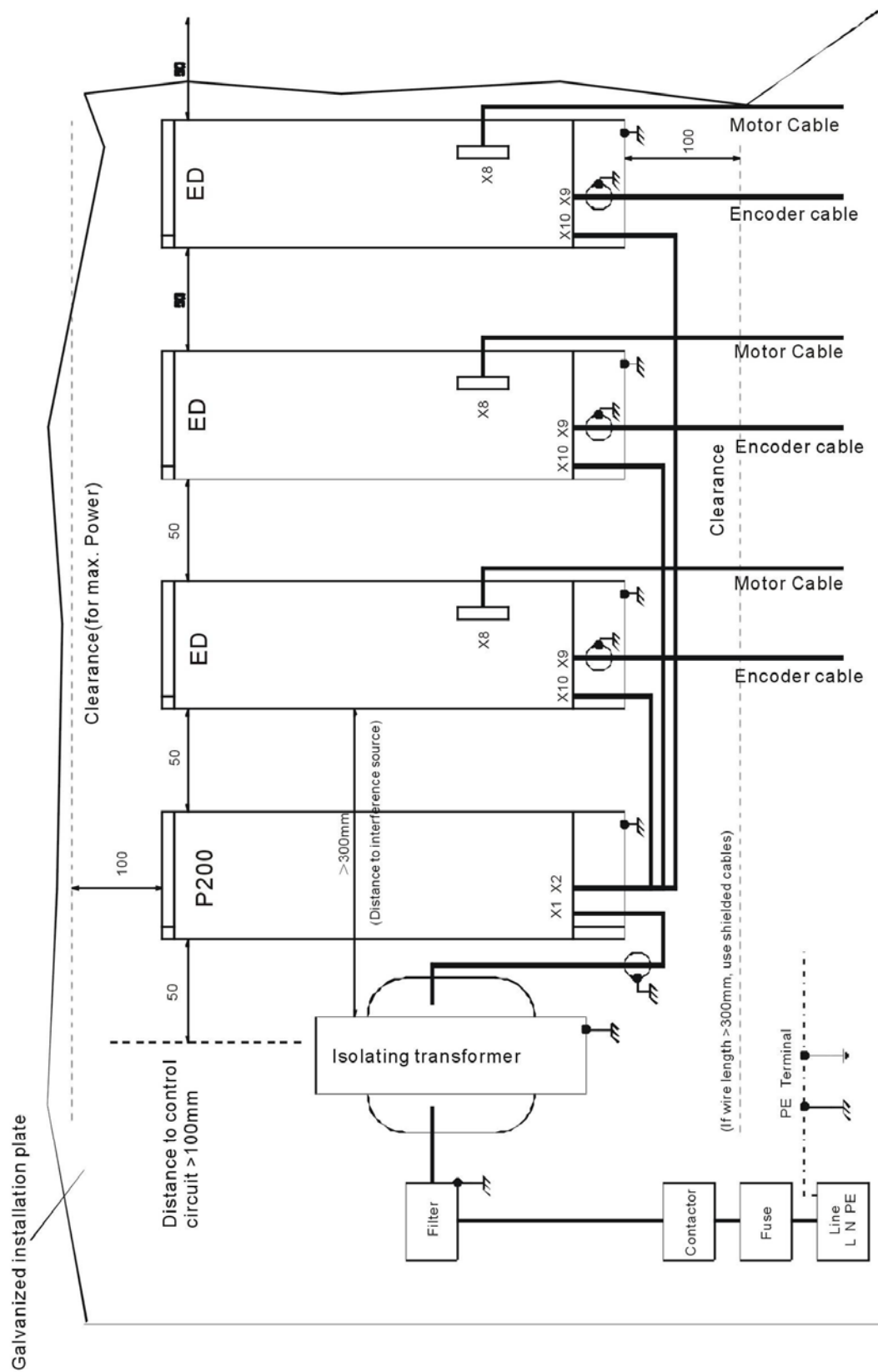
Figure 3.9 EMC Connection

3.4.4 Wiring of ED Drive and PLC Interface

Note: The following is the typical wiring diagram between the ED200 and the PLC. It is also applicable to ED216, ECOLIN216 and ED100. Note that the wiring of the ED100 power supply input is different. For details, refer to the hardware manual. The fuse at the power supply input end of the ED216 and ECOLIN216 (maximum output power 3600W) adopts 20A; the fuse at the power supply input end of the ED200 and ECOLIN200 (maximum output power 1800W) adopts 10A.
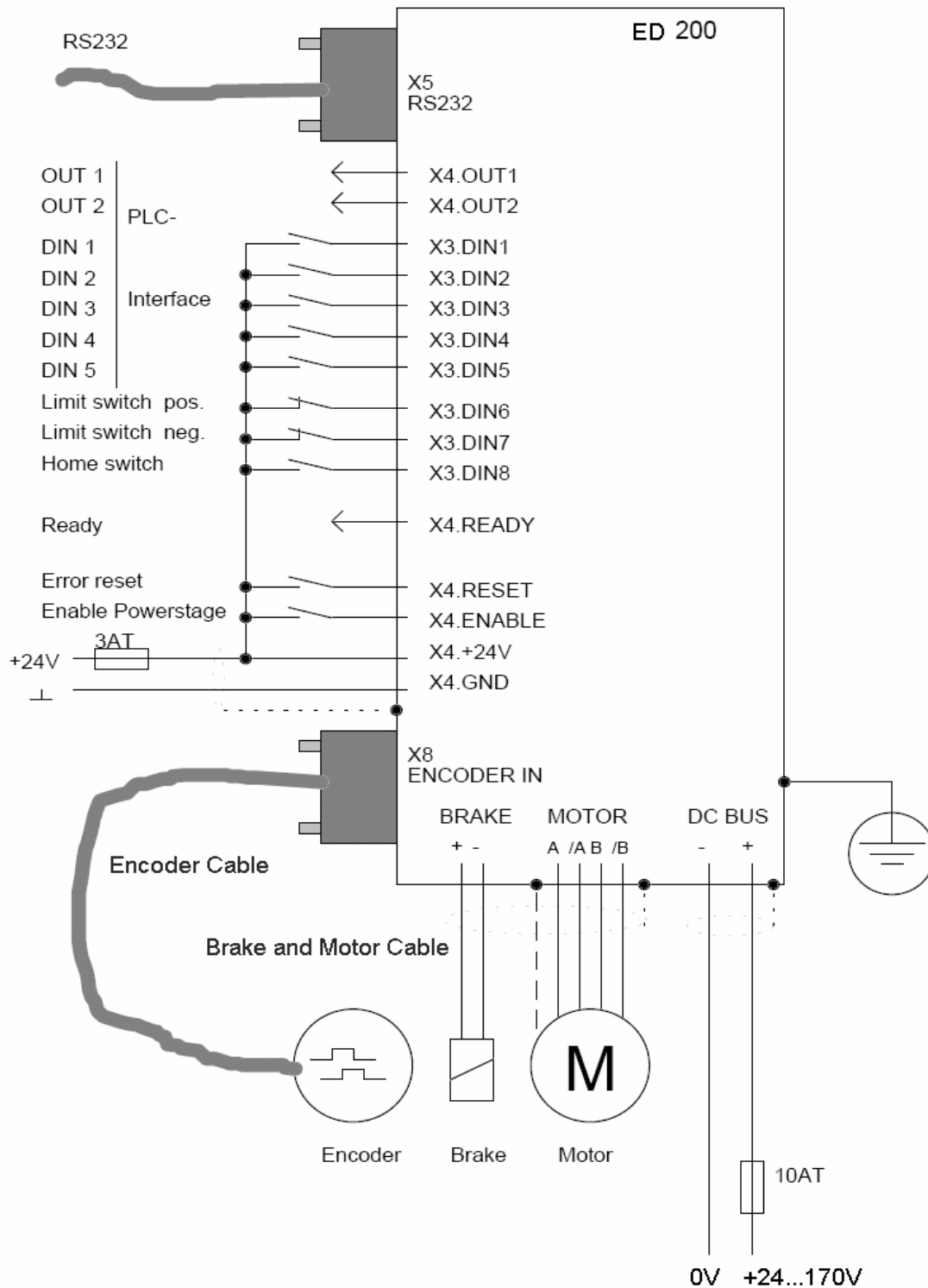
Figure 3.10   ED Drive and PLC Interface Diagram

3.4.5 Hardware Wiring of Minimum Configuration of ED Drive

Note: The following is the wiring diagram of the minimum configuration of the ED200. It is also applicable to the ED216 and ED100. Note that the ED100 power input voltage range is 24 VDC ~ 70 VDC and the fuse shall adopt an 8A fuse. For details, refer to ED100 hardware manual.   The fuse at the power supply input end of the ED216 and ECOLIN216 (maximum output power 3600W) adopts 20A; the fuse at the power supply input end of the ECOLIN216 (maximum output power 1800W) adopts 10A.
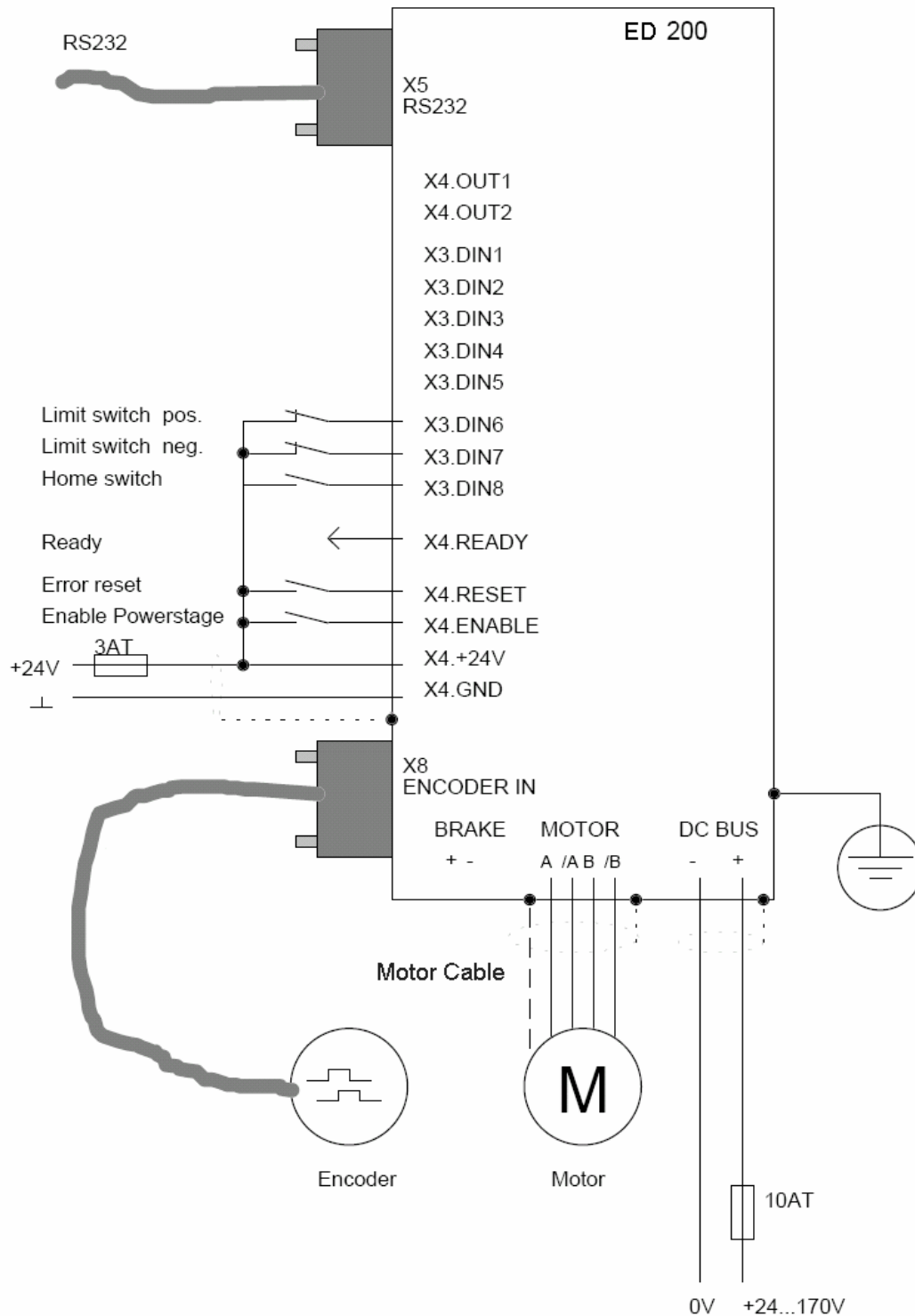
Figure 3.11   Hardware Wiring of Minimum Configuration of ED Drive

3.4.6 Wiring of ED Drive and Motor

The ED servo motor is configured with two pieces of cables: One is the encoder cable, which is configured with a 9-pin connector, and the other one is the motor cable, which is thicker than the encoder cable. The encoder cable can be directly connected to the X8 (Encode in) interface of the ED drive. The signal definition is as follows:

ED X8 (D-Sub Male Connector)	Signal	Pin No. of Round-head Connector (Absent in ED100)
1	5V	12
2	A	5
3	B	8
4	N	3
5	Free / (24V)	Free / (24V)
6	GND	10
7	/ A	6
8	/ B	1
9	/ N	4

Table 3.3   X8 Interface Signal Definition

If the front end of the motor is configured with a brake, a brake cable has to be configured in addition to the motor cable.   Four phase lines of the motor cable are connected to A, /A, B, and /B of the X9 interface of the drive respectively. Two wires of the brake cable are connected to the BRAKE+ and BRAKE- of the X9 interface respectively. For details, refer to the following table:

ED X9	23 Series Motor	34/42 Series Motor	SLM Series Linear Motor
A	Black wire	3	Black wire
/A	Orange wire	1	Orange wire
B	Red wire	4	Red wire
/B	Brown wire	2	Brown wire
GND	Yellow/green wire (Note 1)	Yellow/green wire	Yellow/green wire
Brake(-)	Darkish wire (configured with brake)	Brake wire 3 (configured with brake)	
Brake(+)	Light brown wire (configured with brake)	Brake wire 1 (configured with brake)	

Table 3.4   Motor Cable Connection

Note 1: The 23 series motors have two models: the motor cable of one model has only four wires, with shielded metal net inside, and the other model has five wires, among which four are motor phase lines and the rest one is the yellow/green ground wire. If it is a four-wire cable, the shielded metal net must be pressed under the spring and closely against the drive enclosure (as shown in the figure below). If it is a five-wire

cable, connect the yellow/green wire of the motor cable directly to the GND of the X9 interface. When the power source of the drive exceeds 60 VDC, the ground wire of the motor must be connected. The motor cable and the encoder cable of the 23 series of motors have already been connected to the motor body when they are delivered out of factory. However, the motor cable and the encoder cable for the 34 series and 42 series of motors are independent and pluggable.
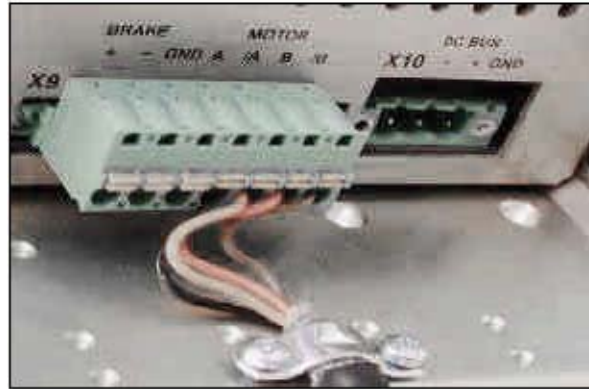
Figure 3.12   Fixing of Motor Cable

Chapter 4 Use of ECO2WIN Software

This chapter gives a detailed description of the programming software for Kinco servo drives—ECO2WIN. It mainly describes the functions of ECO2WIN version2.40.

4.1 Software Setup

The CD of the Kinco servo drive contains the setup software for ECO2WIN.

Operating system and hardware requirements:

      WIN98/2000/XP

      Pentium CPU min. 200 MHz

      Memory > 64 MB

      CD-ROM

      RS-232 interface, parallel interface or USB interface

Getting started:

When you insert the ECO2WIN setup CD into the CD-ROM, the system will automatically run the setup program, as shown in the figure below. Select the setup and debugging software:

Figure 4.1　Setup Interface

Note：The software to be installed consists of ECO2WIN, ECOFLASH and ECO2LOAD. Where, the ECO2WIN software is the compulsory control software for the ED drive, ECOFLASH is used to download special firmware, and ECO2LOAD is used by the PC to back up and copy drive parameters in the file form. In addition, the CD contains additional software, as shown in the following figure, which includes Adobe Acrobat Reader and the PEAK series CAN adapter driver. It is up to the user to decide whether to install such additional software. Please note that if the user has chosen the PEAK CAN adapter to control the ED drive, it is necessary to install the corresponding driver of the PEAK (Presently, PEAK company provides adapters with four optional types of interfaces: parallel interface, USB interface, PCI, and ISA).

Figure 4.2   Setup Interface

IMPORTANT: When the software is installed in a WIN98 system, it is necessary to use the ECO2WIN.INI file (in the directory of Programming software\ECO2WIN) in the CD to overwrite the file with the same name in the setup directory (c:\program file\jat\eco2win 0.0.0.241 by default), and only then the software can work normally. It is not necessary to perform such an overwriting operation in the WIN2000/XP operating system.

4.2 Quick Start

4.2.1 ED Hardware Minimum Configuration

Parameters of all servo drives the ED series can be set through the RS-232 or the CANopen interface by the ECO2WIN software. If only one servo drive is required to use the ECO2WIN software, it is better to have a suitable servo motor connected to the drive.

Connect the servo drive and the motor correctly according to Chapter 3 and the ED drive hardware manual.

- Minimum system requirements for programming through the RS-232 interface:

     Servo drive ED100, ED200 or ED216

     Control logic voltage 24 VDC provided for the drive

     Serial programming cable, 9-pin D-Sub connector; for detailed wiring, refer to RS-232C communication in Section 11.1.

     Optional: ED servo motor

     Optional: Power supply for the motor, min. 24 VDC

- Minimum system requirements for programming through the CANopen interface:

     Servo drive ED100, ED200 or ED216

     Control logic voltage 24 VDC provided for the drive

     PEAK series USB or LPT adapter

     CANopen communication cable with external power supply required; for details, refer to operation instructions for PEAK series USB and LPT adapters.

     Optional: ED servo motor

     Optional: Power supply for the motor, min. 24 VDC

4.2.2 Creating a New Project File

1. Start ECO2WIN.
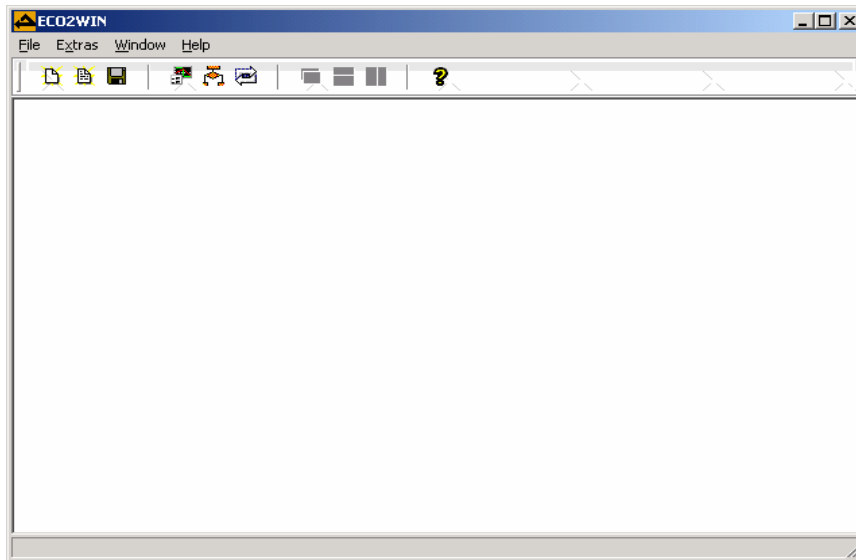
Figure 4.3   Starting ECO2WIN

2. Confirm the welcome window.
3. Set the COM port

Presently, ECO2WIN software supports serial port and PEAK CANopen series adapters. If you use the PEAK CANopen adapter for programming and setting of ED parameters, it is necessary to install the corresponding driver of PEAK CANopen adapter. The CD contains the driver.
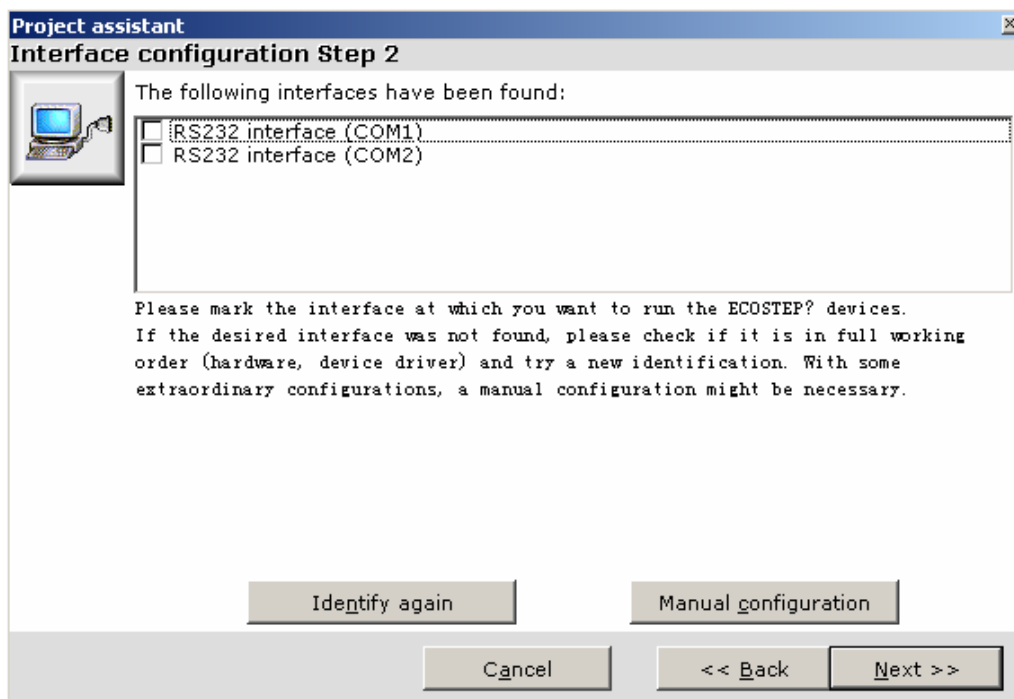
Figure 4.4   Setting COM Port

The communication mode can be selected either manually or by automatic search. In general, automatic search is recommended. In this way, the ECO2WIN will automatically display the list of the communication modes with the ED available on the current PC, and the user can select one mode. If you are not sure whether the settings of communication parameters are correct under this communication mode, click <**Manual Configuration**> to configure parameters manually, as shown in the following figure:
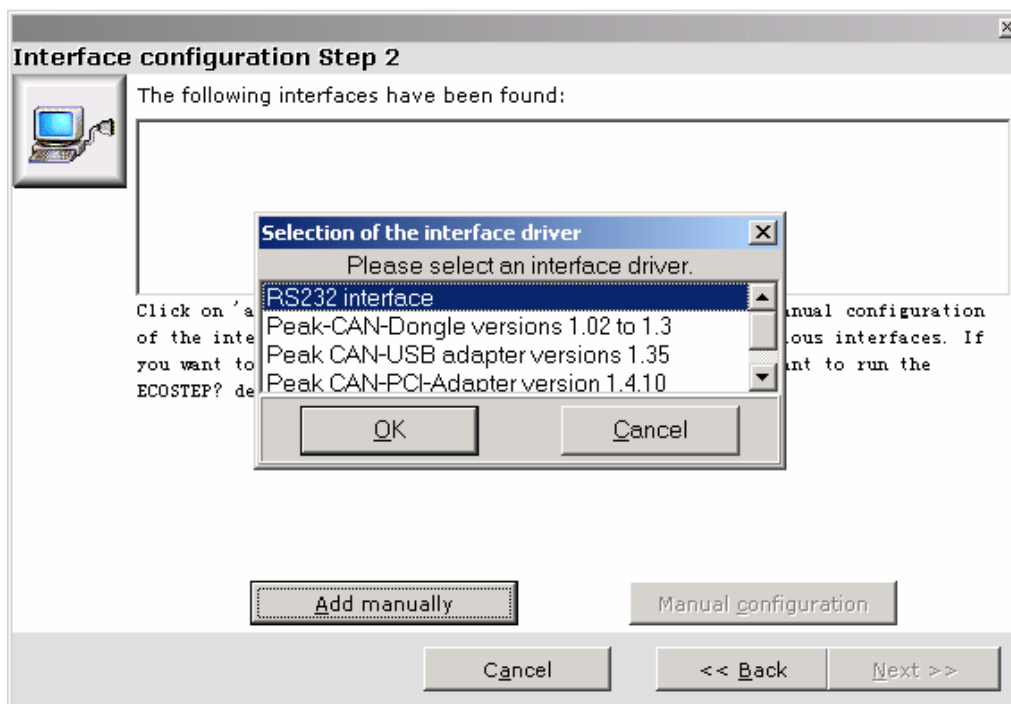
Figure 4.5   Manual Configuration

4. Configure the drive

Click <**Next**> to enter the next interface. The software starts to search the drive in the communication line according to the selected communication mode. Likewise, there are two search modes: default or automatic search. If you select **Manual Device Configuration** for manual search, you have to select the related drive model from the drive list, as shown in the figure below:
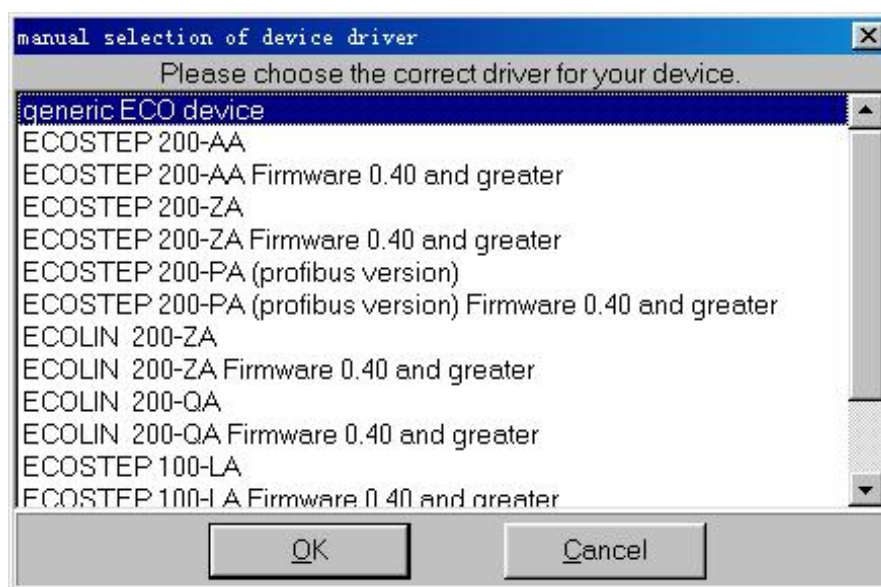
Figure 4.6   Drive Configuration

Automatic search mode is recommended. Once the system finds the drive, it will display the model number and the internal firmware version number of the connected drive, indicating the communication is normal. Then, you can proceed with next operation; otherwise, locate the fault.

5. Configure the servo motor.

Click <**Next**> to enter the motor model number selection interface, as shown in the figure below:
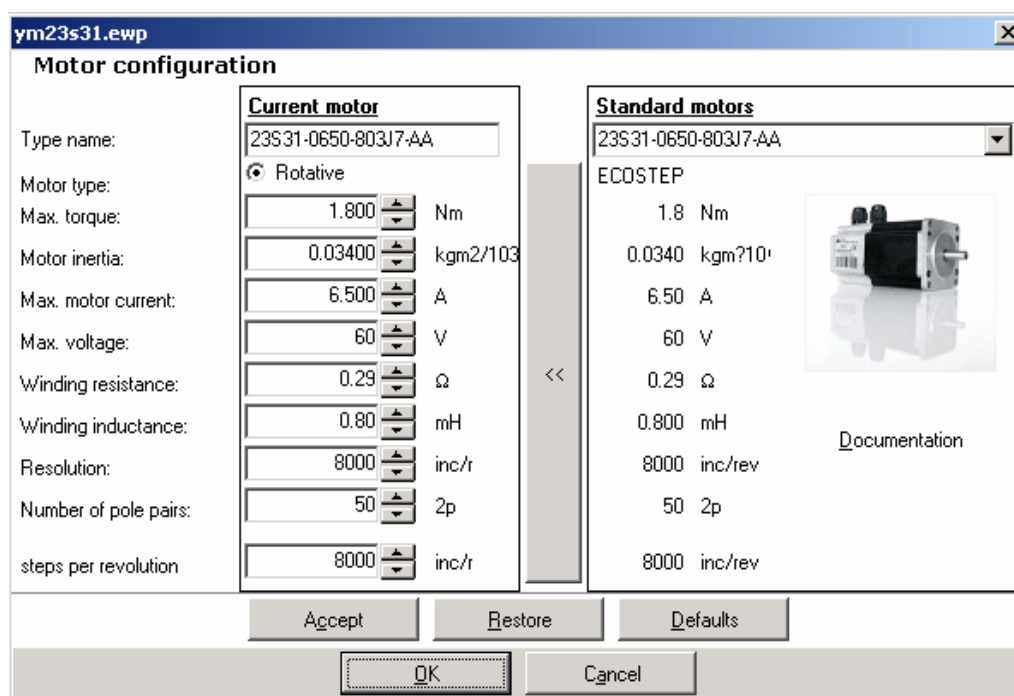
Figure 4.7   Motor Configuration

Select the model of the motor currently connected with the drive in the list on the right of the interface, and then click **<<<>** to confirm the selection.

6. Set mechanic parameters

When the previous step is completed, click **<Next>** to enter the Mechanics type selection interface. There are two options: **Rotative mechanic** (gear etc.) and **Linear mechanic** (belt, screw etc.), as shown below:
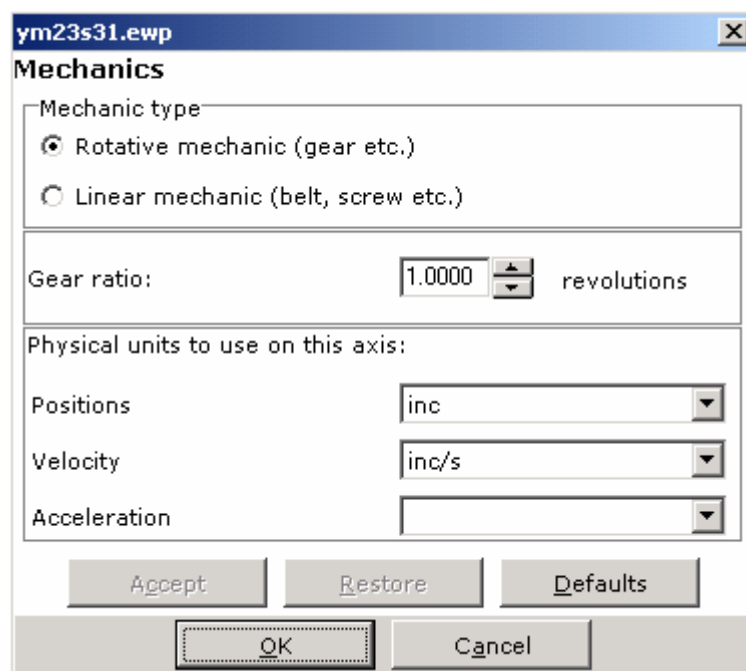
Figure 4.8   Setting Mechanic Parameters

7. Save

Click **<Next>** to enter the interface where you can enter the file name and path of the project to be created.

The software will automatically generate a default file name and path, but the user can modify the file name and path, as shown in the figure below:

Figure 4.9 Saving File and Path

Click <**Finished**> to create a new project. A new interface appears, inquiring whether to enter the ONLINE state (in the Online state, the user can set and monitor the servo parameters online), and click <**Yes**>. The following interface appears:
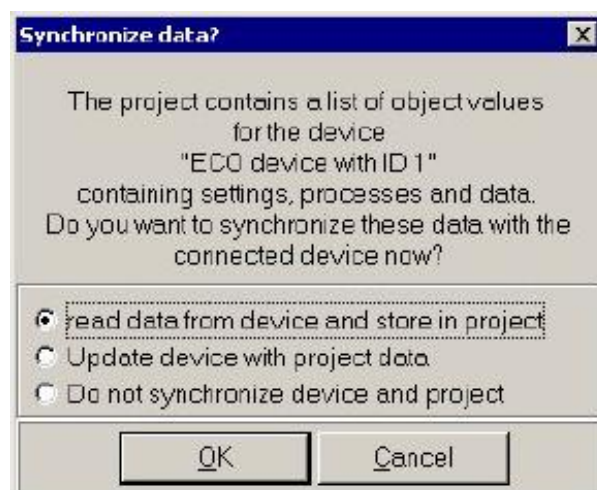
Figure 4.10

There are three options:

- **Read data from device and store in project**: Read all data from the currently connected ED drive and save them as the default settings for the new project;
- **Update device with project data**: Update the internal data of the drive with the data of the new project (this option is selected when the project is completed or an old project is called);
- **Do not synchronize device and project**: Do not synchronize data, and the system will call internal parameters of the drive and update display only when it is necessary. Select this option in general.

After a new project is created, the project main interface appears.

Note: If the RS-232 communication cable is used for programming, the above-mentioned two items will take 2 to 5 minutes (depending on PC configurations).

4.2.3 Opening a Project File

When the ECO2WIN software is running, if the user does not select **New** to create a new project, select **Open** to open a saved project. Likewise, the system will query whether to select **ONLINE** or **Synchronize data**. If the user determines that the settings on interface, device, motor, and mechanic in the current project are consistent with the currently connected system, the user can perform such operation; otherwise, be cautious on data updating.

4.2.4 Main Window

When a project is created or a project is open, the main window of the ECO2WIN software appears on the screen, as shown below:
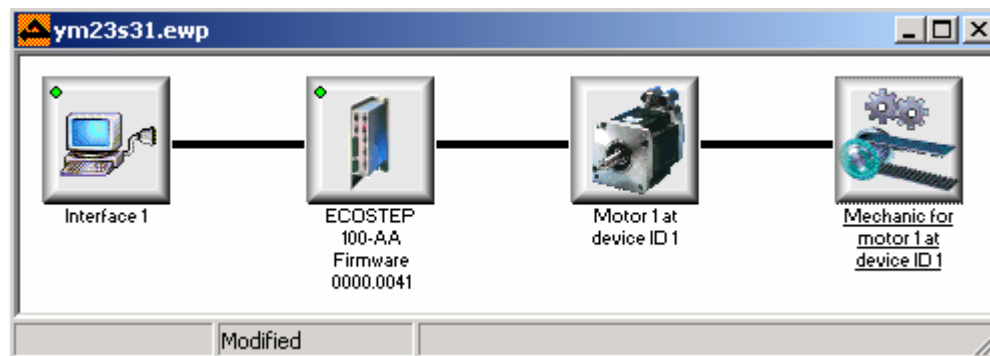
Figure 4.11

We can see from the above figure that the main window contains four icons of Interface1, device, motor, and mechanic. Right click each icon and a corresponding menu appears.

Note: Changes of properties in the menus of **Interface**, **ECO Device**, and **Mechanic for Motor** will only be effective after saving and reopening the project (or restarting the system by Administrator\restart, refer to Section 5.9).

4.2.4.1 "Interface" Icon

Especially, the properties of the interface icon should not be changed. The preset baud rate is limited to 9600 bps for the serial communication. On communication via CANopen, the baud rate has to be set according to the cable length. The standard setting is 1M Bauds.
Right click the Interface1, and the following dialog box appears.
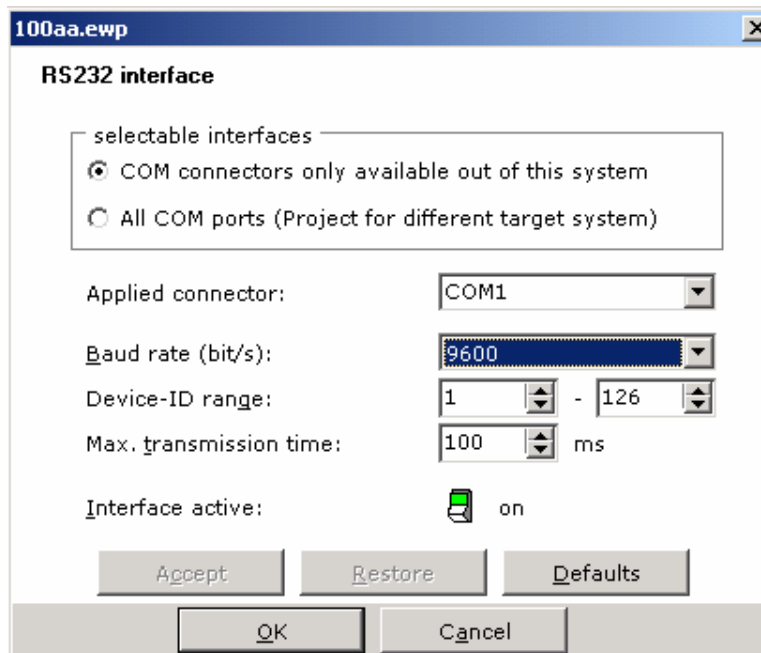
Figure 4.12

If the **Peak CAN** adapter is selected when creating a project, after right-clicking the Interface1, the CAN parameter setting window appears:
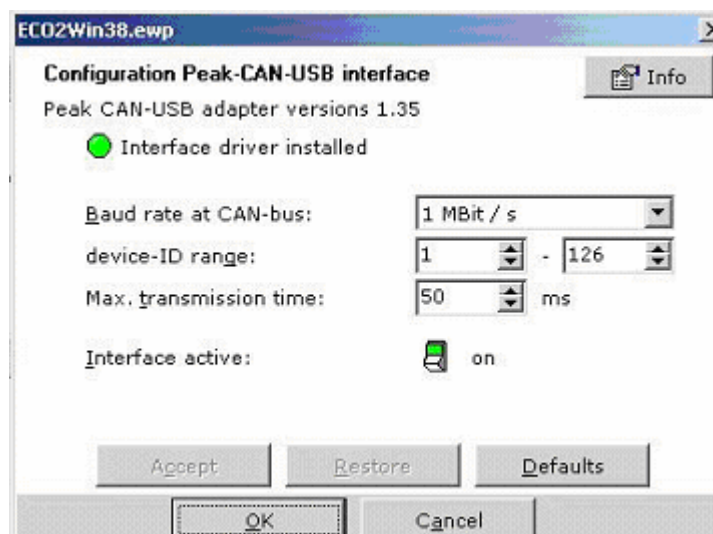
Figure 4.13

4.2.4.2 "ECO Device with ID" Icon

The "ECO Device with ID xx" icon contains main functions of the ED drive. Right click the icon and the following menu appears.
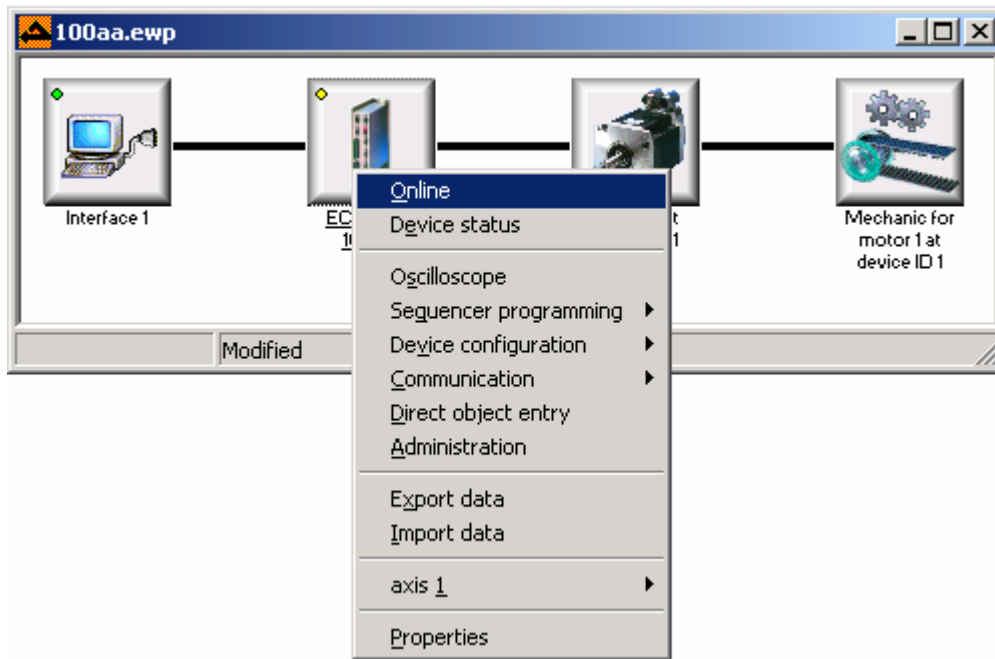
Figure 4.14   ECO Device with ID Menu

"ECO Device with ID" menu functions:

Name	Description
Online	When the **Online** option is selected, the communication to the ED drive is activated
Device status	Information window
Oscilloscope	A built-in oscilloscope software which can collect 4 channels of data with 1000 values for each channel and a minimal resolution of 1 ms (for details, refer to Section 5.8)
Sequence programming	Programming of the drive that can be called internally or externally (for details, refer to Section 5.5.1)
Device configuration	Definition of I/O functionality, definition of master/slave functions and analog inputs (for details, refer to Section 5.6)
Direct object entry	Direct reading and writing of internal data objects of the drive; this address table can be directly accessed by index and subindex
Administration	Administration window for such functions as saving, initializing or booting of the drive (for details, refer to Section 5.9)
Export data	Save internal data of the ED drive to a .ewv file
Import data	Downloading data from a .ewv file to the ED drive
Axis1	Control of the servo axis
Device Control	Control the motion of the motor (for details, refer to Section 5.3)
Parameter settings	Set PID parameters of the drive (for details, refer to Section 5.4)
Properties	Servo drive information window in the current project: changing the configured drive model is possible but not recommended (refer to Figure 4.15)

Table 4.1   Description of the "**ECO Device with ID**" Menu
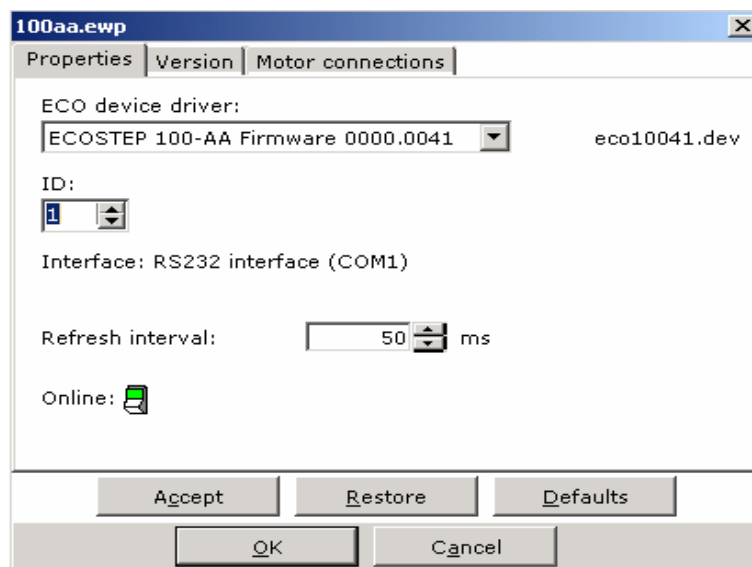
Figure 4.15   Properties Window

The ID in the above window is the DIP address on the drive automatically recognized by the system when a new project file is created. If the drive DIP switch address is manually modified, change the ID in the current window to be the same as the DIP switch address of the current drive. The drive has four ID codes, as shown in the figure below:

S0 is the lowest bit, S3 is the highest bit, On=1, Off=0

For example: 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

4.2.4.3 "Motor xx at Device with ID yy" Icon

Right click on the icon and the following window appears. The window is the same as the one for setting the **Motor** when creating a new project. It is recommended not to change the data here.
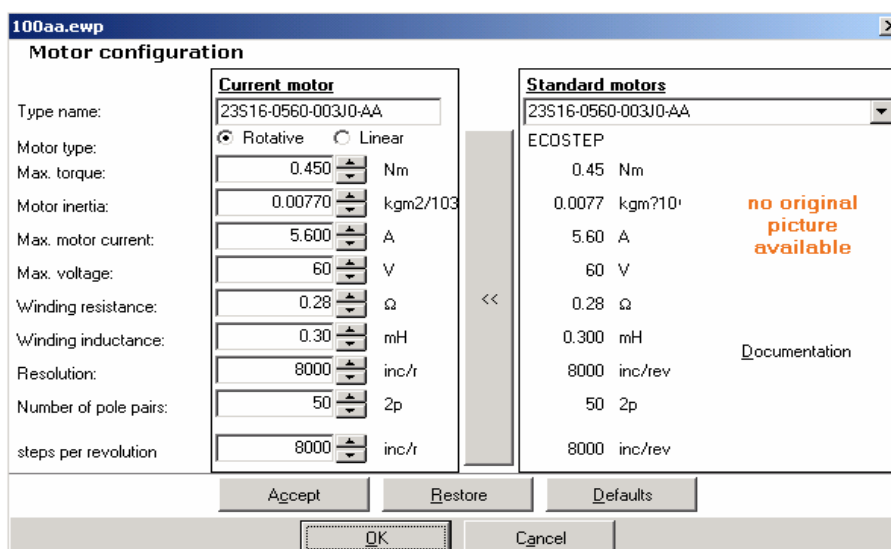
Figure 4.16   Motor Settings

4.2.4.4 "Mechanic for Motor" Icon

Right click on the **Mechanic** icon and the following window appears. The window is the same as the one for setting the **Mechanic** when creating a new project.

After entering corresponding data, the screw pilot and gear ratio are determined. The three parameters on the lowest part of the window determine the position, velocity, and the physical units of acceleration/deceleration.
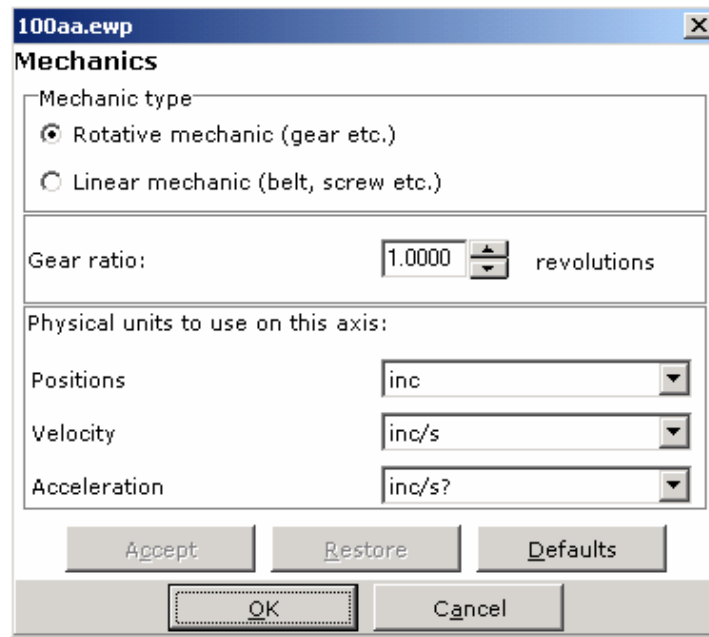
Figure 4.17   Setting Mechanic Structure

4.2.5 Importing/Exporting System Files

The file created with the **New** command or open with the **Open** command is only a project file, which contains the parameters of communication, drive, motor, and mechanics when creating the project file, as well as the most part of content data such as internal PID parameter and Sequence. If you are going to save drive parameters completely after finishing a servo project, it is necessary to use the Import and **Export** function of the Eco2win software. The **Export** function can save all data in the drive to a system file with the file extension of ewv. The default file name is values.ewv. The Import function can be used to re-import an exported system into the servo drive. These two functions are extremely useful for saving servo drive data. In addition, we provide the dedicated ECO2LOAD software to perform such operations as importing, exporting and saving system data or initializing and rebooting system hardware. It is recommended to use this software for importing and exporting drive data. For detailed operation, refer to Appendix F.
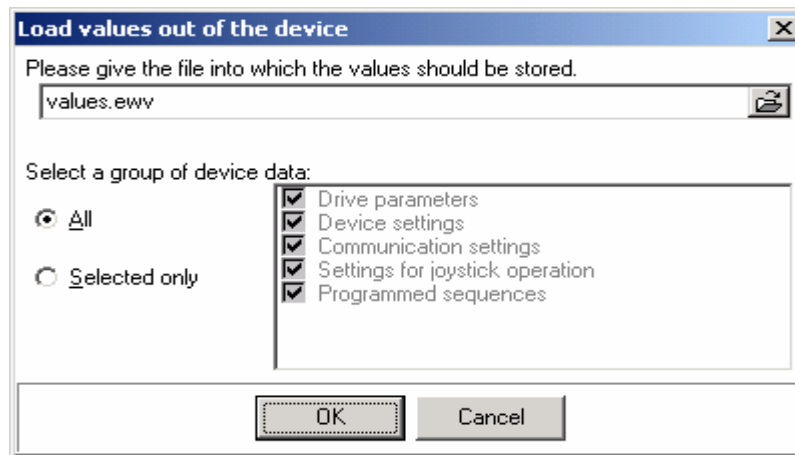
Figure 4.18   Export

When importing/exporting data, the user can select all data or only part of projects.

Chapter 5 Basic Functions of ECO2WIN

5.1 Object Dictionary

The ED drive features a special control mode and uses a kind of strictly defined object lists internally. We called such an object list as object dictionary. The object dictionary is designed on the basis of CANopen international standard, and all objects have clear functional definitions. The said objects are similar to the common memory addresses. Some objects including speed and position can be modified by an external controller, and some other objects like state and error information have to be modified by the drive itself. What we have to do is to assign values for objects or to read the state of internal objects of the drive. This control mode sounds a little difficult for beginners. However, once you have read through this manually and become familiar with the meanings of common objects, you will find this control mode extremely useful.

Objects have the following properties:
1. RW (Read-Write): Object can be read out or written in.
2. RO (read-only): Object can only be read out.
3. WO (write-only): Object can only be written in.
4. M (mappable): Object can be mapped, like indirect addressing.
5. S (Storable): Object can be stored to Flash-ROM and will not be lost even on power failure.

Each object consists of four bytes:
Address format:

Index	Sub	Bits	Property	Meaning

For example:

6040	00	16(=0x10)	RW	Device state control word
2509	00	8(=0x08)	R	Master/slave electronic gear ratio parameter settings
2509	02	32(=0x20)	W	Mapped slave axis speed
2509	03	16(=0x10)	MW	Numerator of electronic gear ratio
2509	04	16(=0x10)	MW	Denominator of electronic gear ratio
2509	05	8(=0x08)	W	Gear mode

Note: Index (index address), with the value range of 0x1000-0xXXXX (0x indicates hexadecimal system)
Subindex (Sub-address of index address), with the value range of 0x00-0xff
Bits, data length in the address, denoted in hexadecimal system:
8 –Data length is 1 byte = 8 bits
10 –Data length is 2 bytes = 16 bits
20 –Data length is 4 bytes = 32 bits
For details on objects provided by the ED drive, refer to Chapter 13, List of Common Objects.

5.2 Direct Object Entry

Right click the "**ECO Device with ID**" icon and then select "**Direct object entry**" in the pop-up menu. The following interface appears:
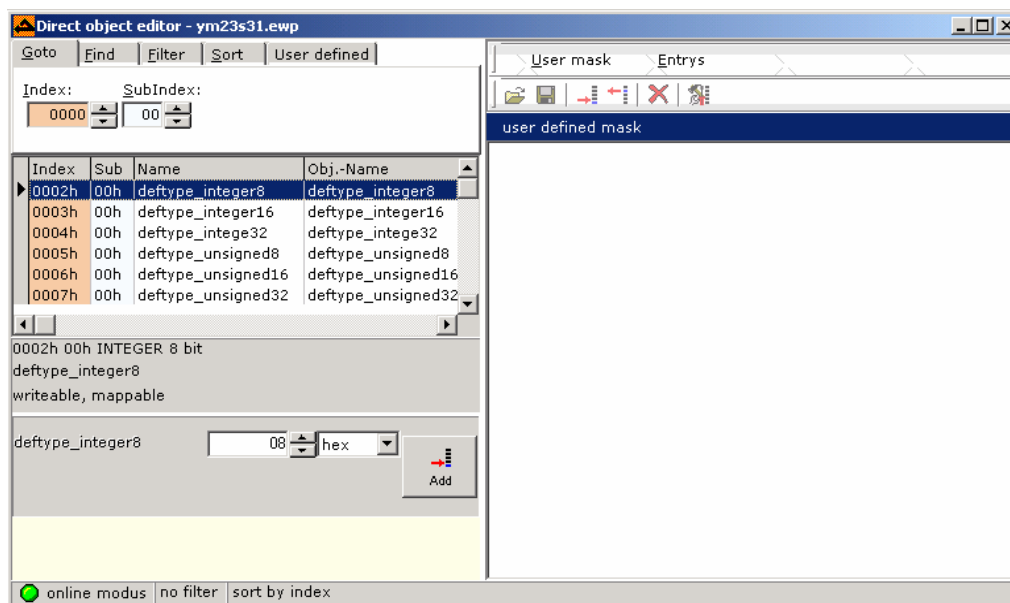
Figure 5.1   Direct Object (3) Editor

Enter index and subindex respectively in the **Index** and **SubIndex** boxes and press the **<Enter>** key. The highlight bar will directly jump to the object, the **Name** column will display the object name, and the left-lower corner of the window will display the data length, property, and value of the object.

If the property of an object is W, you can directly enter the value in the numeric column on the right (this operation is only allowed for a skilled ECO2WIN programmer who knows clearly the impact of the object to be modified on the drive).

5.3 Device Control

Device Control is a functional module in the Eco2win software, which can implement online settings and state monitoring of the servo drive. Right click the "**ECO Device with ID**" icon and then select **Axis 1->Device control** in the shortcut menu. The following functional window appears. If the system is connected to the power supply normally, the default control word (60400010) of the servo drive after power-on is 0x6, device state word (60410010) is 0x0031, and the operation mode (60600008) is 0.
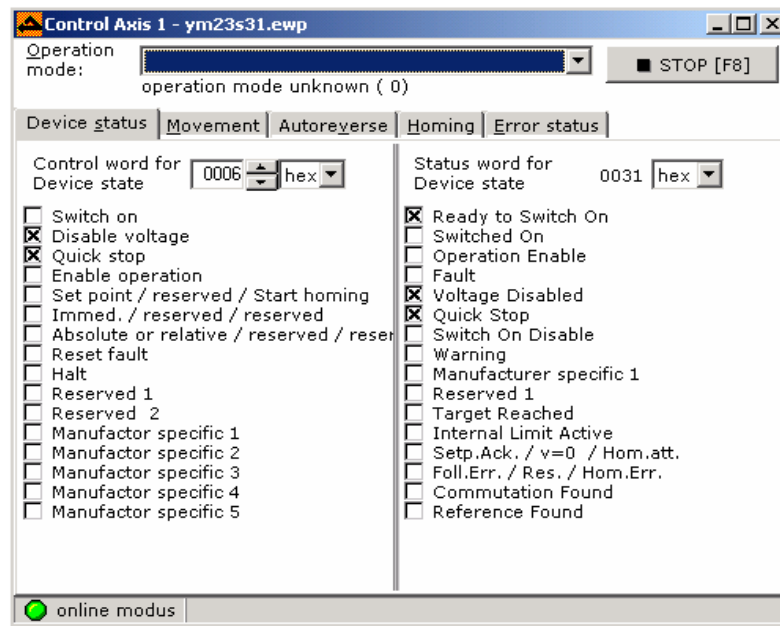
Figure 5.2   Control Axis 1

5.3.1 Operation Mode

An Operation mode drop-down list is provided in the upper part of the Device Control window. In the list box, the ECO2WIN provides the following optional modes:

Mode	Meaning	Value
1	Absolute/relative positioning	60600008=1
3	RPM demand value with mit position control (also called "velocity mode with acceleration/deceleration")	60600008=3
-3	RPM demand value without position control (also called "immediate velocity mode")	60600008=-3
-4	Demand value over master encoder (for example, master/slave control, pulse/direction control, and CW/CCW control are implemented by using this mode)	60600008=-4
4	Torque control mode	60600008=4
6	Homing	60600008=6
7	Control mode using CANopen for interpolation operation	60600008=7

Table 5.1 Operation Mode Definition

The object **Operation Mode** corresponds to the address 60600008 in the **Object Dictionary** defined by the drive. Its property is MW. Change 60600008 to 1, and the drive will be set to **Absolute/relative positioning** mode. The user can modify the value either in the Device Control window or by using the **Sequence programming** function. Alternatively, the user can use the communication interface provided by the ED to read and write the 60600008 object by a host (for example, a PC, PLC, or single-chip processor).

Some operation modes, for example, mode 4 and mode 7, cannot be selected in the **Operation mode** drop-down list box in the **Device Control** window. They can be modified only through direct and indirect (communication) mode.
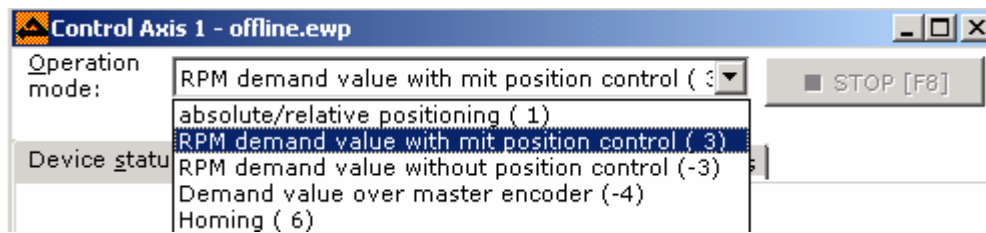
Figure 5.3   Selecting Operation Mode

A brief introduction to each operation mode is given below:

Mode 1: **Absolute/relative positioning**

Take this mode for example: In the coordinate system shown below, the red arrow indicates the actual position = 450. If it is defined as absolute motion, when the target position is set to 700, the motor will move to the position of coordinate = 700; if it is defined as relative motion, when the target position is set to 700, the motor will move to the position of coordinate = 1150.
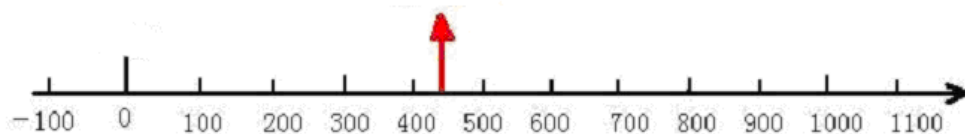
Figure 5.4   Absolute/Relative Position

In this mode, the following objects have to be defined:

Object	Name	Value	Meaning
60600008	Chosen Mode of operation	1	Set the operation mode to absolute/relative positioning
60810020	Profile velocity	User setting	Max. velocity
60830020	Profile acceleration	User setting	Acceleration
60840020	Profile deceleration	User setting	Deceleration
607A0020	Target position	User setting	Target position
60400010	Control word for Device status	0x3F 0x5F	Motor starts operation in absolute positioning Motor starts operation in relative positioning

Table 5.2   Objects of Absolute/Relative Positioning

For details, refer to the sections "Mode and Control" and "Target Object" in Chapter 13, "List of Common Objects".

Mode 3: **RPM demand value with mit position control (also called "velocity mode with acceleration/deceleration")**

This mode implements velocity control over the motor. The operation curve consists of three sequences: acceleration, uniform velocity, and deceleration, as shown below. The acceleration time can be calculated on

the basis of initial velocity, uniform velocity, and acceleration velocity.

$V_t = V_o + at$ V_t-Uniform velocity

                              V_o-Initial velocity

                              a – Acceleration or deceleration

                              t – Acceleration time

$S = V_o t + (1/2) at^2$ S- Acceleration displacement
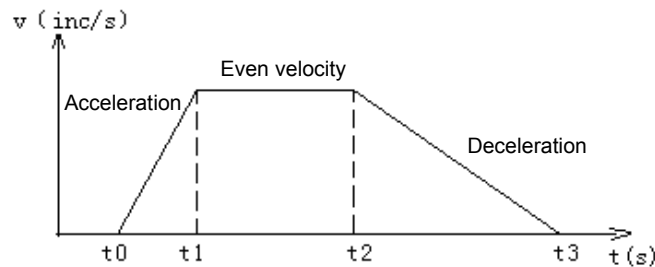
Velocity and time curves in mode 3

In mode 3, the following objects have to be defined:

Object	Name	Value	Meaning
60600008	Chosen Mode of operation	3	Set the operation mode to mode 3
60FF0020	Target velocity	User setting	Target velocity
60830020	Profile acceleration	User setting	Acceleration
60840020	Profile deceleration	User setting	Deceleration
60400010	Control word for Device state	0x0F	Lock the motor shaft; if the above parameters are correctly set, the motor will run according to the requirements of these parameters

Table 5.3   Objects of Mode 3

For details, refer to the sections "Mode and Control" and "Target Object" in Chapter 13, "List of Common Objects".

Mode -3: **RPM demand value without position control** (**also called** "**immediate velocity mode**")

In this mode, when a new value is assigned to the target velocity, the motor will run at the new velocity immediately, without a definable acceleration/deceleration as described in mode 3.

In mode -3, the following objects have to be defined:

Object	Name	Value	Meaning
60600008	Chosen Mode of operation	-3	Set the operation mode to mode -3
60FF0020	Target velocity	User setting	Target velocity

60400010	Control word for Device status	0x0F	Lock the motor shaft; if the Target velocity is not zero, the motor will run at the set velocity

<center>Table 5.4   Objects of Mode -3</center>

For details, refer to the sections "Mode and Control" and "Target Object" in Chapter 13, "List of Common Objects".

Mode -4:   **Demand value over master encoder** (**for example, master/slave control, pulse/direction control**, **and CW/CCW control are implemented by using this mode**)

In this mode, the movement of the motor is directly controlled by the external encoder, pulse/direction, CW/CCW pulse signal from the X7 interface of the drive. If the system receives signal from the external encoder (RS-422 format), set the drive to master/slave mode. The drive will serve as the slave and the motor shaft will be the slave shaft to follow the encoder master shaft signal of the X7 interface to perform the following movement. The velocity rate of the following movement can be set by the electronic gear ratio, as shown in the table below. If the system receives the pulse/direction or CW/CCW signal, it will be controlled by the external pulse command, like the traditional servo drives (like Panasonic and Mitsubishi).
In mode -4, the following objects have to be defined:

Object	Name	Value	Meaning
60600008	Chosen Mode of operation	-4	Set the operation mode to mode -4
25090220	Mapping Slave velocity	60FF0020	The data value of the mapping X7 interface is the target velocity
25090310	Gear factor	User setting	Numerator of electronic gear ratio
25090410	Gear divider	User setting	Denominator of electronic gear ratio
25090508	Gear mode	1 2 3	Set to pulse/direction signal control Set to master/slave control Set to CW/CCW pulse signal control
60400010	Control word for Device status	0x0F	Motor enabling

<center>Table 5.5   Objects of Control Mode -4</center>

For details, refer to Section 5.6.3, Electronic Gear.

Mode 4: **Torque control mode**

In this mode, the motor will output at constant torque. The output torque depends on the value of target torque. To implement the torque control mode, follow the operation procedures below:
1. Set 606000=4 with the Direct object editor of the ECO2WIN software (refer to Section 5.2).
2. Set the 607100 to a desirable current value in the Direct object editor; set it to 8A at the maximum for ED100, to 12A for ED200, and 14A for ED216.
3. Set 60400010=6 in the Device control (or directly set it in the Device control), and then lock the motor shaft, that is, set 60400010=F.

Warning: Before locking the motor shaft, pay attention to the drive. Because it has constant torque output, the motor velocity is only restricted by the value of target torque. Make sure the load is correctly installed and in normal operation before any operation. Remember to set the maximum velocity. In the torque control mode, the maximum velocity can be set only through the **Max. Current** in the **Parameters setting->Current controller** menu.

Mode 6: **Homing**

Mode 6 is the homing mode. In this mode, the drive works in the state of searching for reference. The ECO2WIN software provides 27 homing modes. For details, refer to Chapter 8, Limit Switch and Homing.
In mode 6, the following objects have to be defined:

Object	Name	Value	Meaning
60600008	Chosen Mode of operation	6	Set the operation mode to mode 6
60980008	Homing method	User setting	Homing method, 6 by default
607C0020	Relocation zero point	User setting	Offset relative to the zero point, 0 inc by default
60990120	Velocity during search for Reference switch	User setting	Velocity during search for reference switch, 15625 inc/s by default
60990220	Velocity during search for Reference set point	User setting	Velocity during search for motor index signal, 1562.5 inc/s by default
609A0020	Acceleration during homing	User setting	Acceleration during homing, 160000 inc/s^2
60400010	Control word for Device status	0x1F	Start homing in mode 6

Table 5.6   Definition of Homing Objects

To define these objectives, values can be entered either in the **Direct object entry** window or in the **Device control** window.

5.3.2 Control Command and Status Display

The servo motor has a different operation status from the servo system composed of the servo drive.   We learn the description of space coordinate from the mechanic engineering: determining different coordinates of the system at different times (s(t), v(t), t).   This is also the case in our servo systems. The servo system will provide you with such information as whether the power is switched on, whether the drive has a fault, or whether the target position is reached. The ED servo system adopts the CANopen international standard to define internal objects as well as its status word and control word. The status word represents the current status of the drive, and the control word is used to change the drive status.
In the left part of the **Device status** column in the **Device Control** window, the user can enter different control commands in the text box to implement control over the motion of the servo system. The control command defines the address in the object address table as 60400010, which is a 16-bit word. Assigning different values to the word will implement different control actions. Common values for the control word include:

    Value   Command

0x06   Power off, set drive to ready state

0x0F   Power on, motor shaft in locking state

0x3F   Absolute positioning

0x5F   Relative positioning

0x0F   Start motion

0x1F   Start searching reference in homing mode

0x80   Fault reset

The right part of the Device status column in the Device Control window is a status display area. The ED drive uses a status word 60410010 (16-bit) to indicate the status of the current drive.

Each bit of the status word indicates a status. For example, bit0 represents "**Ready to Switch on**" and bit15 represents "**Reference Found**".

Bit number (bit)	Bit	Meaning	Value of Corresponding Status Word
0	Ready to Switch on	Ready to switch on	60410010=0x0001
1	Switched On	Powered on	60410010=0x0002
2	Operation Enable	Operation enabled	60410010=0x0004
3	Fault	Fault	60410010=0x0008
4	Voltage Disable	Voltage output disabled	60410010=0x0010
5	Quick Stop	Quick stop	60410010=0x0020
6	Switch On Disable	Power-on disabled	60410010=0x0040
7	Warning	Warning	60410010=0x0080
8	Manufacturer specific 1	Manufacturer specific 1, reserved	60410010=0x0100
9	Reserved 1	Reserved 1	60410010=0x0200
10	Target Reached	Target reached	60410010=0x0400
11	Internal Limit Active	Internal limit activated	60410010=0x0800
12	Setp.Ach./v=0/Hom.att.	Pulse response	60410010=0x1000
13	Foll.Err./Res.Hom.Err.	Following error/homing error	60410010=0x2000
14	Commutation Found	Commutation found	60410010=0x4000
15	Reference Found	Reference found	60410010=0x8000

Table 5.7   Functional Description of Each Bit of Status Word

We can judge the current status of the servo system according to the status word. The status word can be directly observed in the **Device Control** window of the ECO2WIN software or read out through the communication interface provided by the Sequence and ED. In this way, the user can monitor the ED either inside or outside of the software.

5.3.3 Setting Movement Parameters

In the **Movement** tab of the **Device Control** window, as shown below, the user can set movement parameters in position and velocity mode. These parameters can be modified online.
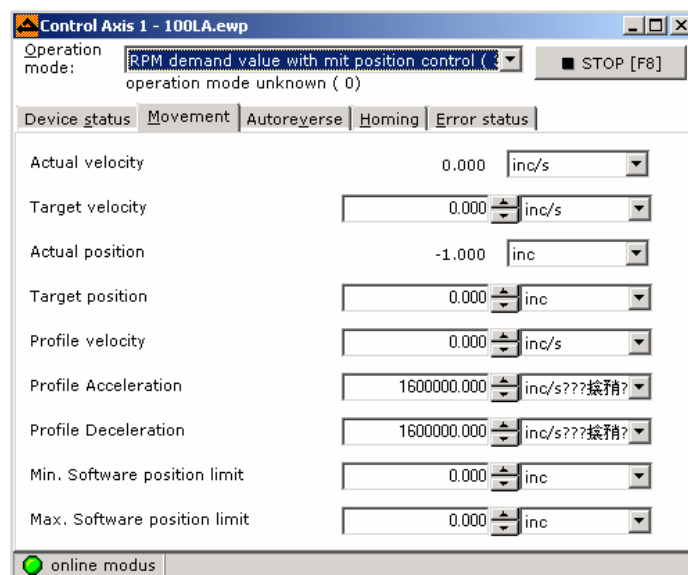
Figure 5.4   Parameter Settings in Movement Tab

For detailed movement parameters to be set in different position and velocity modes, refer to the description of operation modes in Section 5.3.1. It should be noted that the **Min. Software position limit** and **Max. Software position limit** parameters are used to set the forward (counterclockwise direction by default) and reverse limit position (clockwise direction by default) of the motor. If you set **Min. Software position limit** = -160000inc and **Max. Software position limit** = 320000 inc, in position mode 1, when the set target position (607A00) goes beyond the range defined by these two parameters, the motor will no longer operate when it arrives at these two position values. The default value is 0, indicating no limit exists.

To modify the counting direction of the motor encoder, modify **position polarity** and **velocity polarity**. For details, refer to Section 5.4.5.

5.3.4 Homing Control

Click the **Homing** tab in the **Device Control** window, as shown in the following figure. Where, you can set the homing methods and other related parameters in the homing mode and set the start/stop of homing search. It should be noted that **Time limit for homing** can set the maximum duration for searching. If no reference is found within the time limit, the system will report "No reference", and the second LED in the lower-right corner will turn yellow. If a reference is found within the time limit, the second Led in the lower-right corner will turn green, and the system will give the prompt of "Found reference".
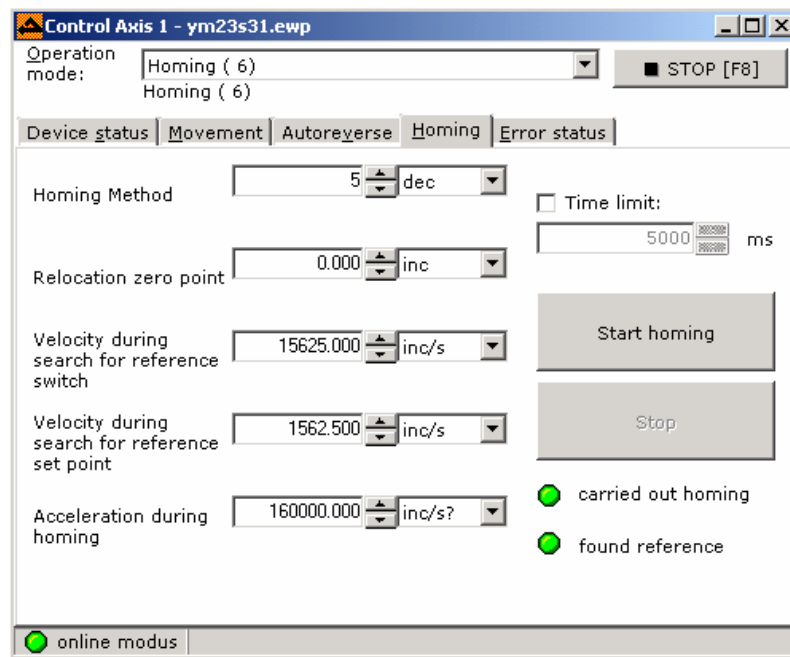
Figure 5.5   Homing Control

5.3.5 Auto Forward/Reverse Rotation Control

In the **Autoreverse** tab of the **Device Control** window, the user can set different forward or reverse rotation modes for the system. The ECO2WIN provides five forward/reverse rotation control modes: Timed forward/reverse rotation with position control, timed forward/reverse rotation with speed control, position forward/reverse rotation with speed control, speed forward/reverse rotation with position control, and speed forward/reverse rotation with speed control. For details, refer to Chapter 7.
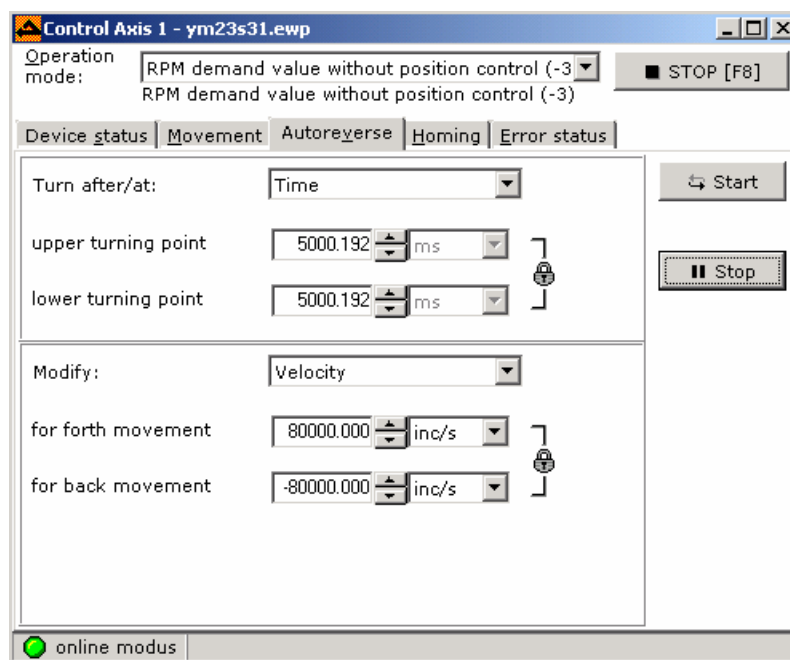
Figure 5.6 Autoreverse Setting

5.3.6 Error Diagnosis

If the red **Error** LED of the drive blinks after the system is powered on, it indicates a system fault.
The **Error status** tab in the **Device Control** window is designed to monitor fault status of the drive, as shown in the figure below: The ED drive assigns a status word 26000220 to indicate the fault status of the drive. The object is a 32-bit word. Each bit of the lower 18 bits and bit0 to bit17 of the word represents a fault. For details, refer to Chapter 13, Error Codes.
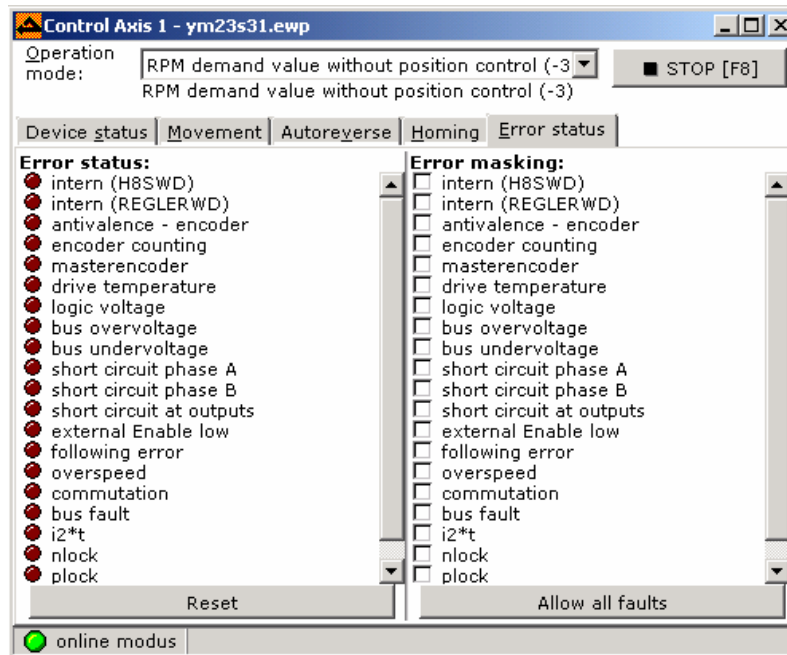
Figure 5.7 Error Status Fault Diagnosis

The left pane of the window displays current fault status. If no fault is found, all LEDs are dark brown. If a fault is detected, the corresponding LED turns red. Click the <**Reset**> button in the lower part of the pane, or enter the control word 0x86 in the **Control word for Device control** box in the **Device status** tab to remove the fault or to set the drive to the "Ready" status. The right pane of the window is the error detection selection bits intended for test. The factory default setting is to detect all faults. The user can choose to mask all faults to let the motor continue to run or to select a bit (marked with "x") to ignore the single fault by the system. Remember to ignore a fault only when necessary; otherwise, the hardware may be burned when the system no longer detects this kind of fault.

Take the i^2t fault for example. This fault is designed to protect the motor coil from overheating. It consists of two parameters: one is the maximum current and the other is overheat time. When actual current is greater than the set value and the duration is three times over the overheat time, the system will generate a i^2t fault. Once the motor is reset upon a fault, the user can enter the control word 0x0f to re-lock the motor shaft.

The following errors can be corrected easily:
Temperature: The temperature of the heat sink of the drive shall not exceed 80°C. Check the ambient temperature and reduce the temperature if necessary.
Logic supply: 24 VDC logic control power (X4 interface), normally too low
Overvoltage: An overvoltage fault occurs if the ED100 power voltage > 75 VDC or the ED200/ED216 power voltage >180 VDC
Undervoltage: Too low a motor voltage; a undervoltage fault occurs if the voltage < 24VDC

External Enable: No 24V input at the Enable end of the X4 interface

Following error: Re-set the max. following error value, optimize PID parameters of the controller, reduce acceleration/deceleration

Overspeed: The velocity value is higher than limit, reduce velocity value

Bus error: Check power supply of the bus

I^2 t: Motor overloading, cool down

The following errors indicate faults with the drive:

Intern (H8SWD): Internal fault of the drive; send the drive to the manufacturer for checking

Intern (REGLERWD): Internal fault of the drive; send the drive to the manufacturer for checking

The following errors indicate faults with the motor:

antivalence-encoder: Encoder cable fault, check whether the cable is connected to the X8 interface correctly.

Encoder counting error: Disturbance on encoder cable, incorrect connection of shielded and ground cables, or broken encoder disc due to the force of motor shaft.

Phase A: Wrong connection, short circuit, or disconnection of phase A line of the motor

Phase B: Wrong connection, short circuit, or disconnection of phase B line of the motor

Several errors may occur at the same time under some circumstances. In this case, do not mask errors.

Note:

1. Do not mask errors!

2. Fault detection ensures safe operation of the system. Masking errors prevents the system from switching off and affects system safety.

5.4 Parameter Setting

The servo control system contains position controller, velocity controller, and current controller. Operation parameters of the motor have different settings under different control modes.

5.4.1 Position Controller
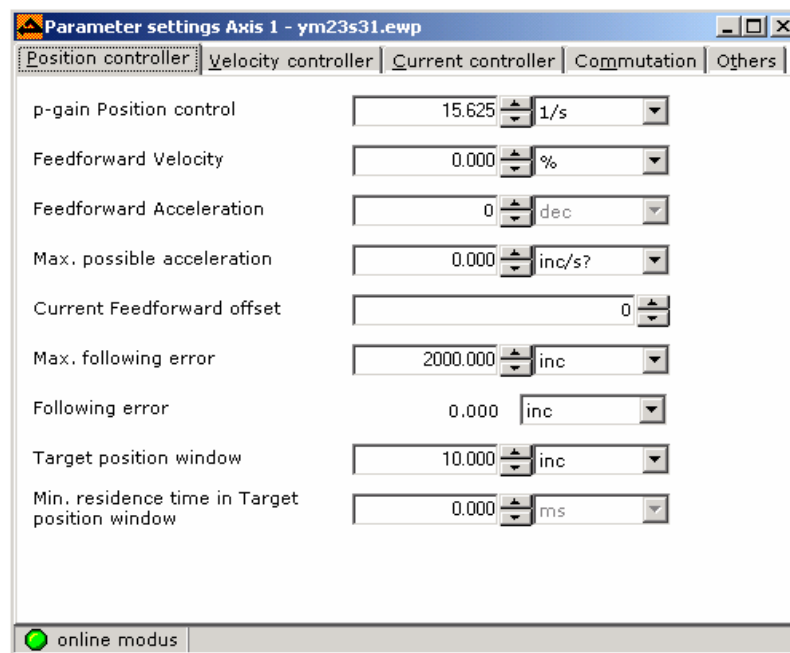
Figure 5.8   Position Controller

In this window, the user can set position controller parameters. Where, Kpp (that is, p-gain position control), Vff (Feedforward Velocity), and Max. Following Error are very important parameters in control mode 1 and control mode 3. When the motor runs normally, the following error value can be calculated according to the following formula:

 Following error=(V-V*Vff/100)/Kpp,   V-Actual Velocity (Using the **Profile Velocity** set value in calculation)
When Vff=0, **Following error** =V/Kpp, and ensure the **Following error** shall be less than the **Max. Following Error**; otherwise, the **Following error** may result in system shutdown.

Note: The calculation of the following error is based on the comparison between the set position and the actual position.

When Vff=100, that is, 100% feedforward, and **Following error** = 0, the following error can be ignored.
For safety purpose, the value of the position controller is small when the drive is delivered out of the factory, for example, Kpp =15.625 1/s. The user can re-set the position controller parameter according to the mechanic structure in use. In general, Kpp is in the value range of 100 to 200 (1/s), with some exceptions.

5.4.2 Velocity Controller
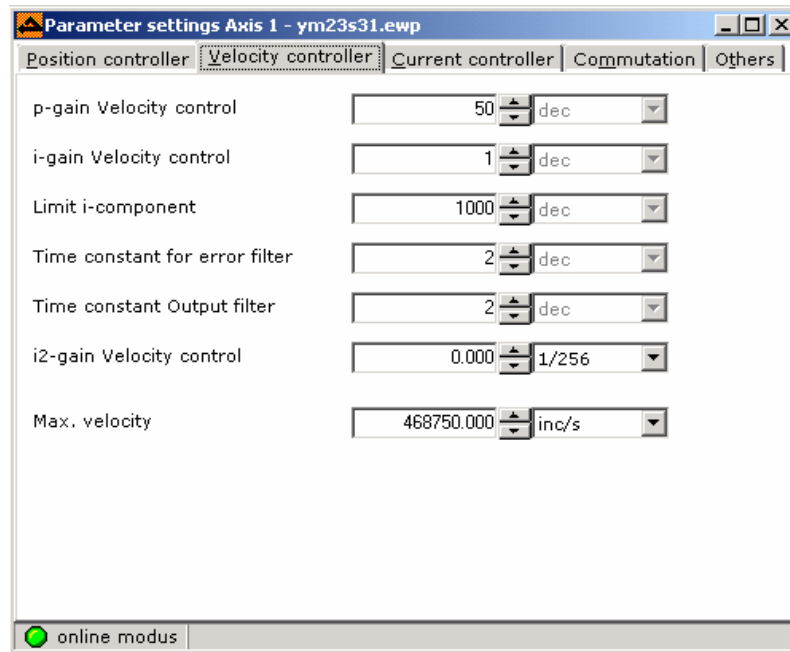
Figure 5.9   Velocity Controller

In this window, the user can set important parameters for the velocity controller, including:

      Kvp (p-gain velocity control);

      Kvi (I-gain velocity control);

      Limit I-component;

      e_filter: Time constant for error filter;

      o_filter: Velocity output filter time constant;

      Max. velocity: Maximum velocity

For details, refer to related configuration tables in Chapter 10, Controller Performance Adjustment. When motor vibration or noise becomes obvious, adjust these parameters according to Table 10.2 in chapter 10, Controller Performance Adjustment.

5.4.3 Current Controller
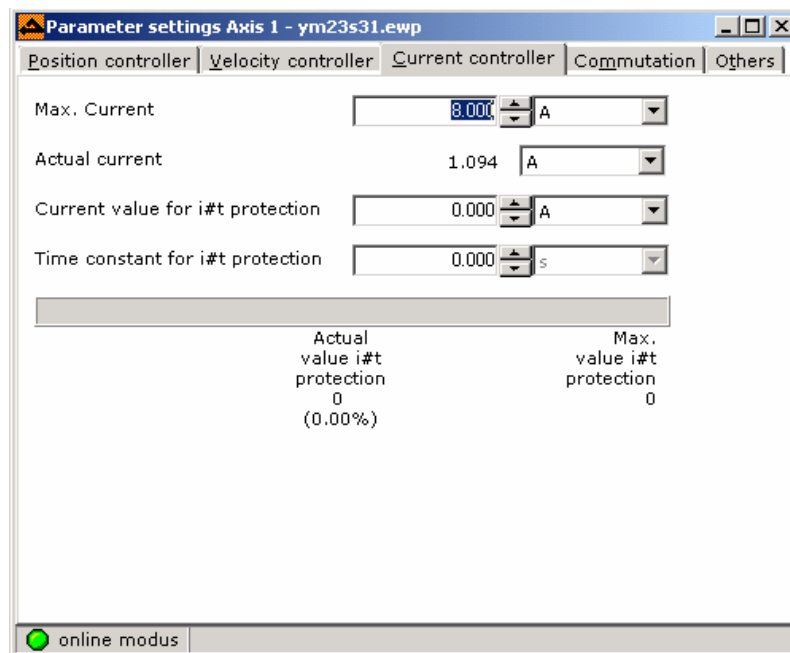
Figure 5.10　Current Controller

In this window, the user can set the maximum rated current (**Max. Current**) for the motor according to different drives. The value is set to 8A for ED100, 12A for ED200, and 14A for ED300 (Note that almost all parameters have a unit of "**dec**", which is an internal unit of the system and whose corresponding value is the value actually saved and processed in the drive). The **Actual current** frame displays the current phase current of the motor, based on which the user can judge whether the current exceeds the rated value and virtually whether the motor selected by the system is overloaded. The above figure shows the current controller of a motor when it carries no load.

Current value for i#t protection:　Set the current value for i^2t protection
Time constant for i#t protection:　Set the time value for i^2t protection

5.4.4 Commutation
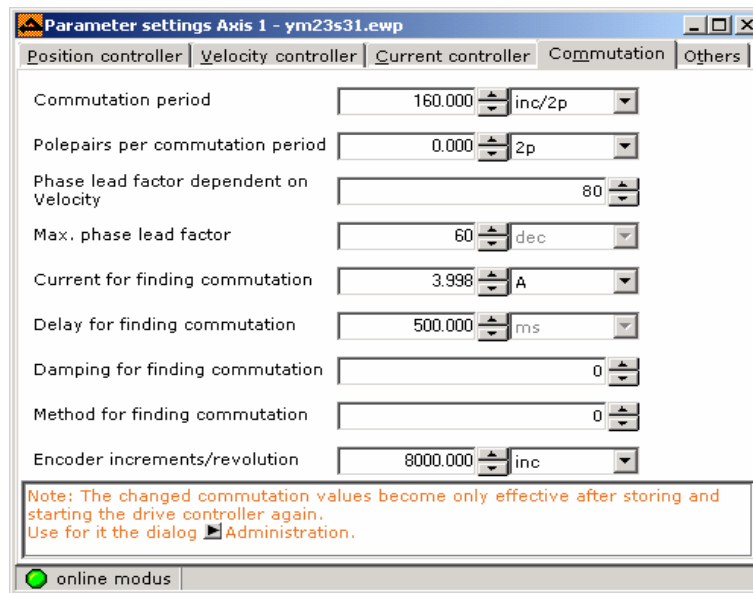
Figure 5.11   Commutation

After the drive is powered on and enabled (set drive control word to "F"), it will provide a current for the motor to rotate to a certain angle so as to determine the position status of the motor. If it is correct, it will be displayed in **Device Control→ Device Status→ Status word for Device state→ Commutation found**. The angle of rotation depends on the values of two parameters **Current for finding Commutation** and **Delay for finding commutation**.

Settings of **Commutation** parameters depend on the motor and operation environment. In most cases, these parameters are set when the project file is created. Proper setting of **Commutation** is of vital importance for servo application. Particular attention shall be paid to this point when using the ED servo drive.
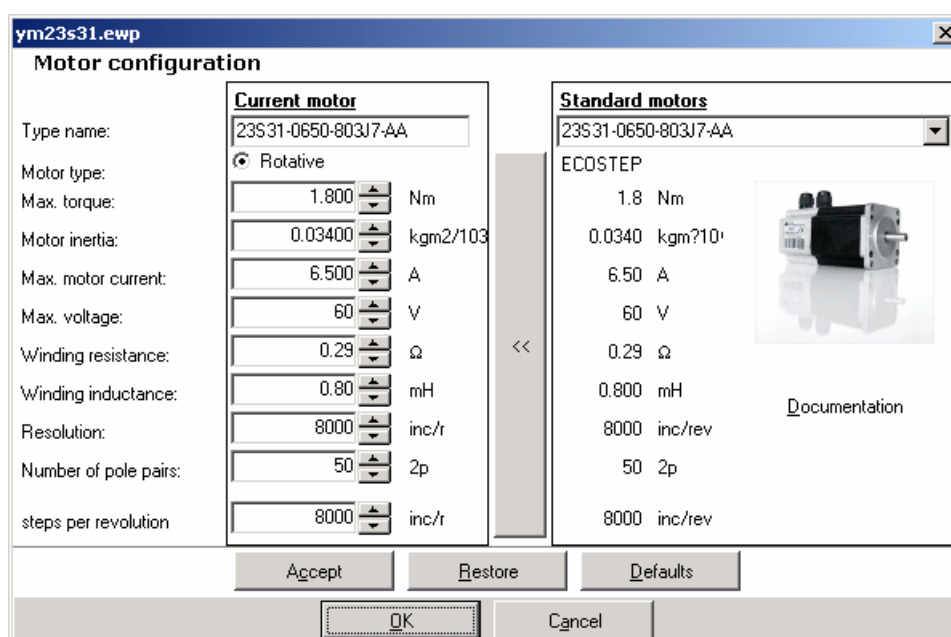
Figure 5.12   Motor Configuration

The computation formula for Commutation period (corresponding to 0x60F60120) and Polepairs per commutation period (corresponding to 0x60F60210) shown in Figure 5.11 is as follows:

For Kinco rotative servo motors:

commutation period = Number of pulses per revolution/ pole pairs

For example, for the motor 23S21-0560-803j7-AA, if encoder resolution = 8000 inc/rev, and pole pairs = 50

Then, commutation period = 8000/50 = 160

polepairs per commutation period=0

Note: We can also enter 8000 and 50 respectively.

For **Kinco linear servo motors**:

commutation period = pole period/resolution

For example, for linear motor SLM-040-192-200, if resolution with grating scale = 1 um and magnet period = 32 mm, then,

commutation period = 32/0.001 = 32000

polepairs per commutation period=0

A group of commutation objects are defined as follows:

Index [hex]	Subindex [hex]	Name	Current Value	Meaning
60F6	01	Commutation period	160	Commutation period, with its computation formula described earlier in this section
	02	Polepairs per commutation period	0(1)	Pole pairs, set to 0 for a linear motor
	03	Phase lead factor dependent on velocity	80	Depending on motor model, operating voltage and application, referring to Table 5.9
	05	Max. phase lead factor	60	Refers to 8000inc/rec, and is proportionally increased with higher resolution
	06	Current for finding commutation	1024	Induction current, normally less than the rated current of the motor
	07	Delay for finding commutation	500ms	Standard value
	08	Damping for finding commutation	0	Maximum (0-20) Reducing the offset during homing
	09	Commutation method	0	3: For horizontal axes 2: For motors or linear motors with low pole pairs 1. For vertical axes

Table 5.8   Meanings of Commutation Objects

The value of the object 60F60310 (Phase lead factor dependent on velocity) depends on the motor model

and operating voltage. For detailed settings, refer to the table below:

Motor Type	Voltage			
	24VDC	60VDC	90VDC	150VDC
23S16-0560-803J7-AA	230	130	-	-
23S21-0560-803J7-AA	260	180	-	-
23S31-0560-803J7-AA	300	230	180	-
34S42-0560-803J7-AA	-	100	90	80
34S80-0560-803J7-AA	-	210	200	180
42S90-1400-803J7-AA	-			300
80B31-0330-A03JA-AA	0	0	0	0
80B32-0340-A03JA-AA	0	0	0	0
80B33-0460-A03JA-AA	0	0	0	0
110B31-0640-A03JA-AA	0	0	0	0
110B32-0700-A03JA-AA	0	0	0	0
110B33-0640-A03JA-AA	0	0	0	0

Table 5.9   Settings of Commutation Parameters

5.4.5 Others
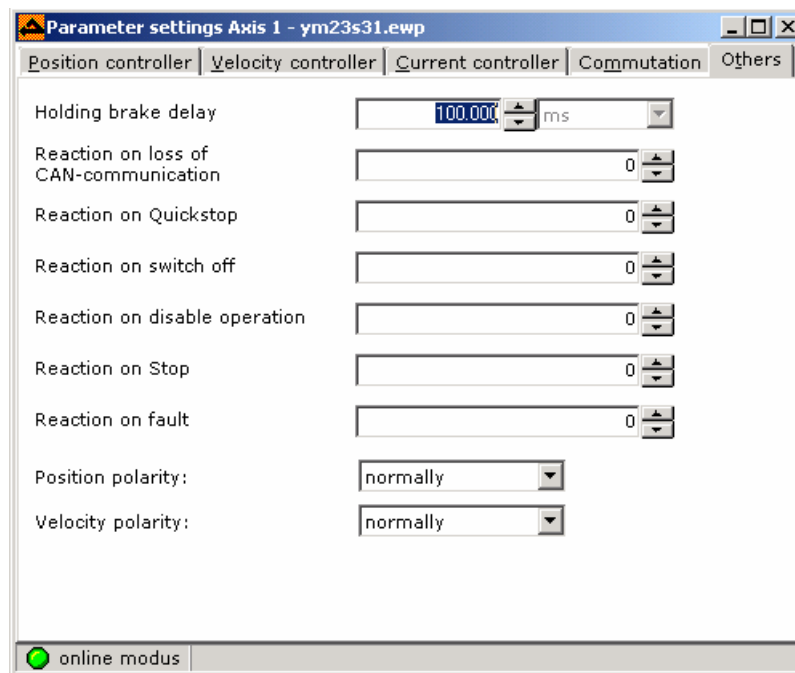
Figure 5.13

Here, the user needs to understand different system settings under different circumstances:

Reaction on loss of CAN-communication (CANBUS communication interrupted and CAN node protection function enabled):

Set value 0: No action

Set value 1: Generate an error

Set value 2: Control word BIT1 is reset

Set value 3: Control word BIT2 is reset

Other value: No action

Reaction on quickstop (The drive changes from motion to a quick stop, and the control word Bit2 is reset)

0: Disconnect the power supply of the motor immediately

1:    Decelerate according to the system deceleration speed until standstill

2~4: Decelerate and stop according to the quick stop deceleration

Reaction on switch off

1: 1. Decelerate according to the system deceleration speed until standstill

Other values: Switch off immediately

Reaction on disable operation

1: Decelerate according to system deceleration speed and forbid operation

Other values: operation disabled immediately

Reaction on stop: reserved

Reaction on fault

0: Disable drive immediately

1: Decelerate according to running track and then disable operation if possible

Other values: Press Quickstop to decelerate and the disable operation if possible

Pay attention to two parameters **Position polarity** and **Velocity polarity**. The positive counting direction of the motor is set to CCW when a drive is delivered out of factory. To change the default direction, modify these two parameters.

5.5 Internal Programming

The ED drive provides powerful control functions, which find good expression in its internal programming control function. The ED provides 16.5 KB space for the use of 256 sequences. The programming process is simple and fast and supports multiple calling modes (input triggered, timed triggered, event triggered, direct calling, etc.). The internal functions such as logic comparison, arithmetic operation, and data recording can be implemented in the sequences, as described later in this section.

5.5.1 Sequences

Open the Sequences programming window, and the following interface appears:
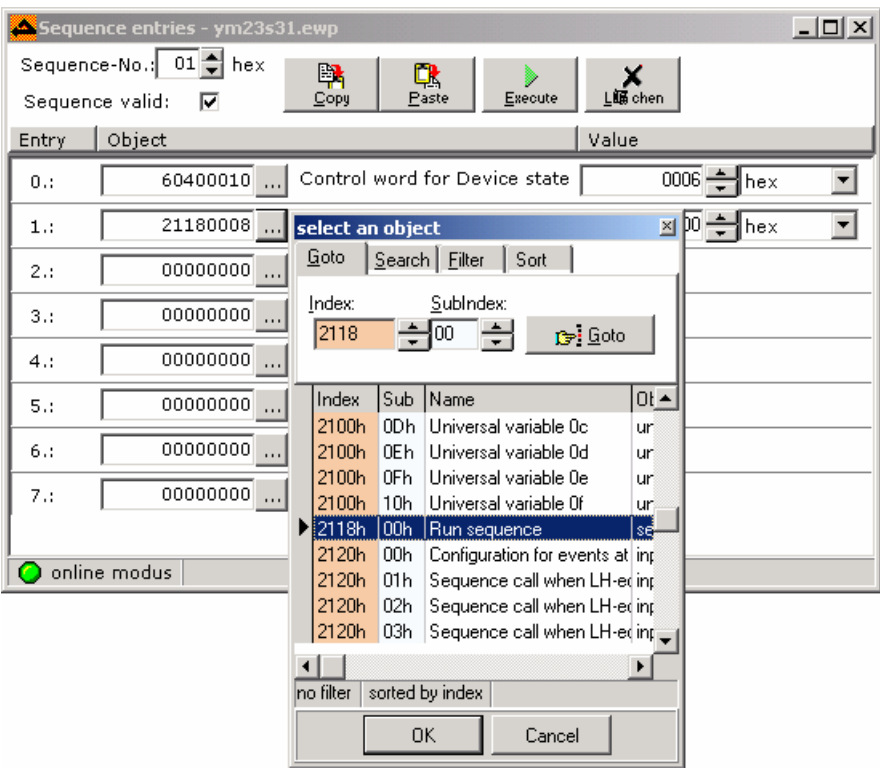
Figure 5.14   Sequence Programming

In the figure, we can select Sequences-No (00~FF). To validate this field, select the Sequence valid check box. The programming is simple. Select the Index and Subindex of an object, and then enter values in the right pane. Up to 8 objects can be set in a sequence. If you assign the value xx (00~FF) to the register of 0x21180008, the system will jump from the current sequence to sequence xx.

For details, refer to Chapter 9, Sequencer Programming.

5.5.2 Digital Input

5.5.2.1 Edge Controlled Call

In the **Digital Input** window, the user can set the sequences to be called by 8 digital inputs of the ED drive so that the internal sequences can be called by the digital inputs.
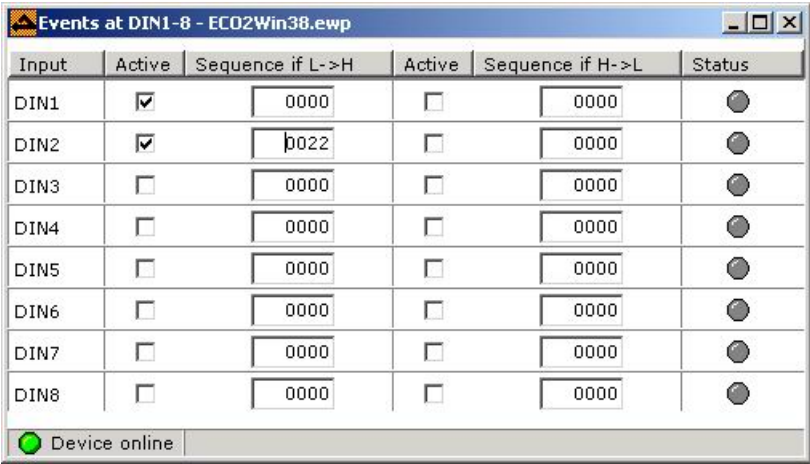
Figure 5.15

In the above figure, the parameter **Active** is controlled by the parameter 0x21210010, and 16 sequence numbers are assigned by 16 objects 0x21200110~0x21201010. The ED series drives provide 8 inputs (DIN1 ~ DIN8), and the change of their levels can call different sequences. Therefore, the user can conveniently call the motor to execute different actions according to external conditions. For example, when DIN1 in the figure changes from L (low level) to H (high level), the sequence 0x0000 will be called; when DIN2 changes from L to H, the sequence 0x0022 will be called.   Upon completion of setting, 0x21200110 in the object index list stores the sequence number 0x0000, and 0x21200210 stores the sequence number 0x0022. Note that in 0xXXXX, 0x represents the hexadecimal system; the first two XX has special meanings, which will be described respectively in different functions and the default value is 00; the last two XX represents Sequence No.

A typical servo application uses three inputs as limit and homing inputs (refer to Chapter 8, Limit Switch and Homing). If three inputs are used, only five digital inputs will be available. To call more sequences through the inputs, you have to call sequences by the coding scheme described in the next section.

5.5.2.2 Binary Code Call

1. Conditions for binary code calling:
    2 ~ 6 input signals for coding
    A high-bit input for calling the sequence
    The sequence number actually called = BCD value entered + Offset

2. Setting method

Open the window **Sequencer programming->Digital Input** and define a high-bit input as the trigger signal.

BCD Input	Trigger Signal (Input) to Enable BCD Input	Logic Status	Max. Value of BCD Input	Offset	Sequence No.	
					without Offset	with Offset
DIN1-2	DIN3-8	4	03	04	00-03	04-07
DIN1-3	DIN4-8	8	07	08	00-07	07-0F
DIN1-4	DIN5-8	16	0F	10	00-0F	0F-1F
DIN1-5	DIN6-8	32	1F	20	00-1F	1F-3F
DIN1-6	DIN7-8	64	3F	40	00-3F	3F-7F

Table 5.10

For example, define DIN1-DIN3 as BCD inputs, and select any input from DIN4-DIN8 as a high-bit signal. The following figure defines DIN4 as the trigger signal and offset as 08. According to definitions in Table 5.10, set the "Sequence if L->H" of DIN4 to 0708.
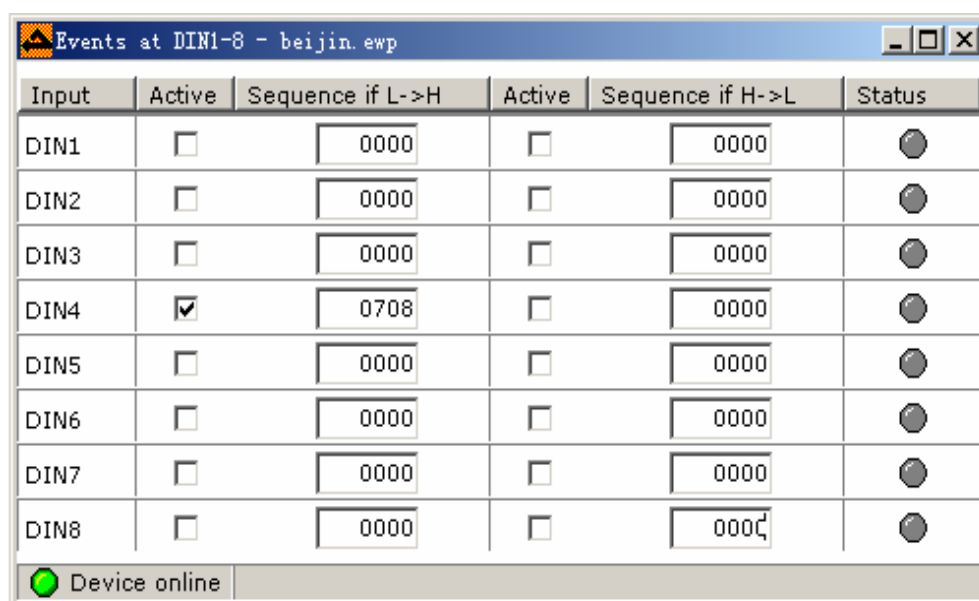
Figure 5.16 2   Calling Sequences by BCD Inputs

When entering BCD=3 (DIN1=1,DIN2=1,DIN3=0, 1 indicating high level input, 0 indicating low level input) for DIN1 to Din3, and if DIN4 changes from L to H (low level → high level), the sequence 8+3=11 will be called, and the corresponding Sequence No. is 0B (hexadecimal).

When entering BCD=6 (DIN1=0, DIN2=1, DIN3=1) for DIN1 to Din3, and if DIN4 changes from L to H (low level → high level), the sequence 8+6=14 will be called, and the corresponding Sequence No. is 0E (hexadecimal).

Set the value of the trigger signal with a reference to Table 5.10, for example, 0708 in the above table. In the value 0xXXXX, the first two bits are high bits and correspond to the fourth column in Table 5.10. The last two bits are low bits and represent the offset, corresponding to the fifth column in Table 5.10.

Another example is that when DIN1 ~ DIN6 serve as BCD inputs, any one from DIN7~8 can be defined as the high-bit trigger signal. The following figure defines the trigger signal as DIN7 and offset as 40.   When input level of DIN7 changes from L to H, actually called sequence = (DIN1 ~ DIN5) code + 0x40. When the BCD corresponding to DIN1 ~ DIN6 is 3, the sequence 0x43 will be called.
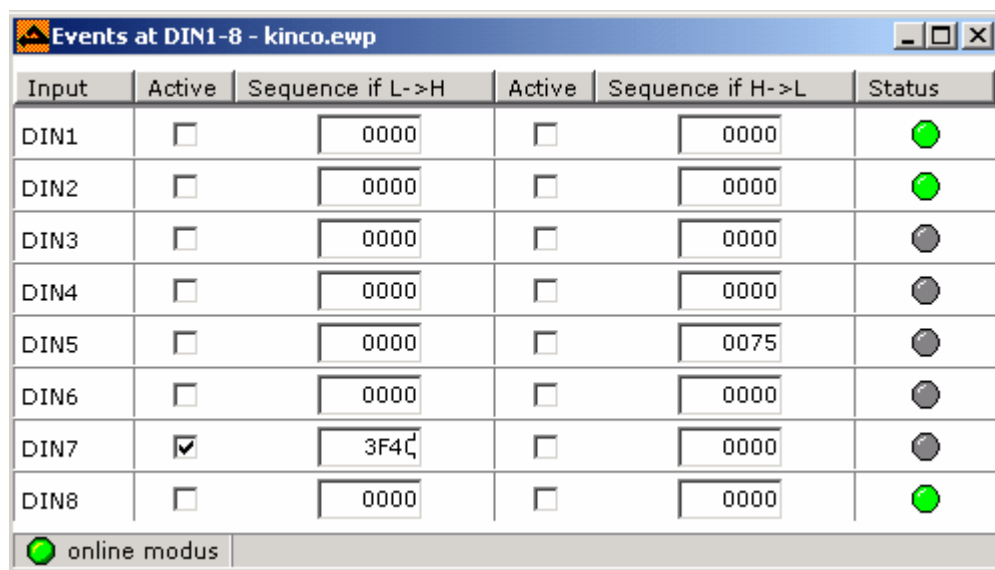
Figure 5.17 Digital Input

Note: The maximum BCD input can be DIN6.

There can be 256 (0xff) sequences at the maximum, so the offset + BCD value cannot exceed the value.

5.5.3 Comparator

Figure 5.18 Comparator

The drive provides four comparators and calls sequences by comparing operation results. Four comparators correspond to internal objects Index, 0x2180~0x2183, Sub index 01~09. For details, refer to the section Comparator Objects in Chapter 13, List of Common Objects.

The computation process is as follows:

(Mapping data source + Offset) and (AND-mask)= (Interim result)

Result = (Interim result) Operator (Comparison value)

If the result is true, the Result = 1. Click **Release** and the pre-entered sequence will be called. To re-call the sequence, it is necessary to click **Release** again. In the sequence, the comparison has to be made again.

Comparator objects are defined as follows:

Index [hex]	Subindex [hex]	Name	Meaning
2180-2183	01	Mapping data source	Mapping data source
	02	Offset	Offset
	03	And mask	AND
	04	Comparison value	Comparison value
	05	Operator	Operator
	06	Sequence call	Sequence to be called when comparison result is True 0x-80xx

	07	Auxiliary value	Internal auxiliary value, can be read, written, or mapped to other objects

Table 5.11   Comparator Objects

5.5.4 Timer/Controller Events

5.5.4.1 Timer Sequence Call

The sequences in the ED can be called by timer and controller events, which adds great flexibility to programming, as shown in the figure below:
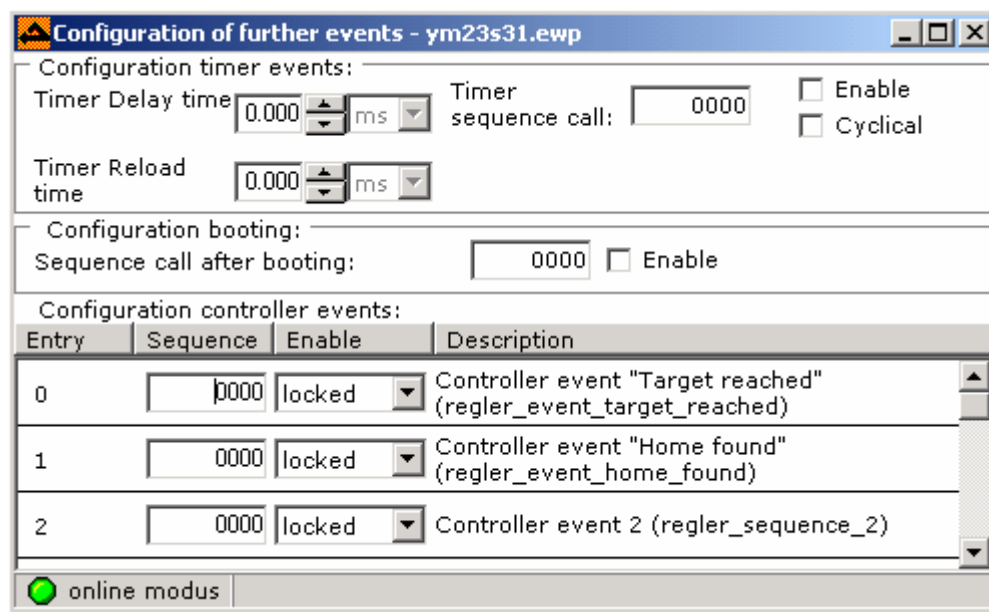
Figure 5.19   Timer Events

In the upper part of the window, the user can configure timer events for calling sequences. When we enter SEQ No. (in the format of 80XX, XX indicating sequence 00~FF), and enter delay time in the **Timer Delay** time box, and select the **Enable** check box in the upper-right corner, the timer is enabled. When the timed time arrives, the corresponding sequence will be called. How to implement this function in sequence programming? Remember that the timer corresponds to two objects (register addresses): 0x21300110 corresponding to sequence No., and 0x21300210 corresponding to timed time, as shown in the figure below:
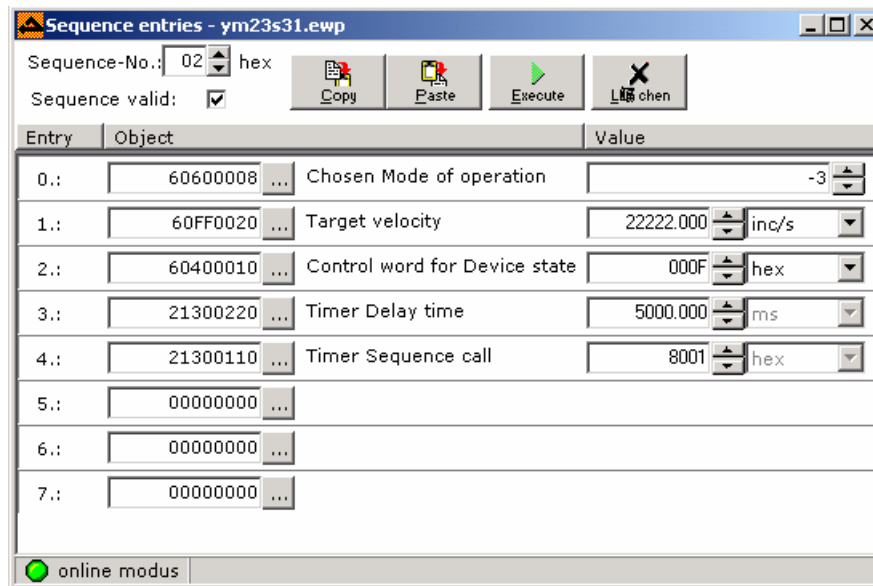
Figure 5.20

We see from the above figure that the SEQ02 has two commands: 3 and 4. When SEQ00 is called, it firstly sets the system operation mode to -3 and the motor runs at the velocity of 22222 inc/s; and then it sets the delay of the timer 1 to 5s. When the timed time arrives, SEQ01 will be called. Note that the high bits for SEQ No. must be 80.

5.5.4.2 Sequence Call after Booting

We recommend not to use this function. Make sure the mechanical structure of the system provides necessary protection and the sequence to be executed immediately after power-on will not cause collision due to different initial load positions in case of using this function.

5.5.4.3 Sequence Calling by Controller Events

The ED drive provides such internal events as "Target reached" and 'Home found", which can be used in sequence programming. The controller event corresponds to the object 0x2140xx. For detailed description, refer to Chapter 13, List of Common Objects. For example, set 21400110=8001, and define the function of the object 21400110 as control event "Target reached", that is, calling the sequence 1 when the "Target reached" event happens. The designated sequence format is 0x80xx, indicated in hexadecimal system, and xx is in the range of 00 ~ FF.

5.5.5 Counter

The ED drive provides four counters and they correspond to objects indexes of 0x2190 ~ 0x2193. Each counter has two sub-objects: Sub01 and Sub02. Sub01 is used to store increments (see Increment column in the figure below), and Sub02 is used to store accumulated value. To perform accumulative addition, place the numbers in Sub01, and the value in Sub02 will add the numbers in Sub01 automatically.
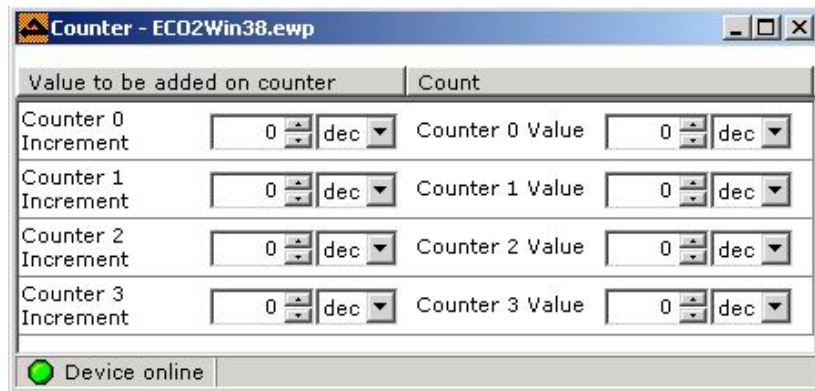
Figure 5.21

For example, add 1 to the counter whenever the level of the external DIN input changes from L to H, as shown in the figure below:
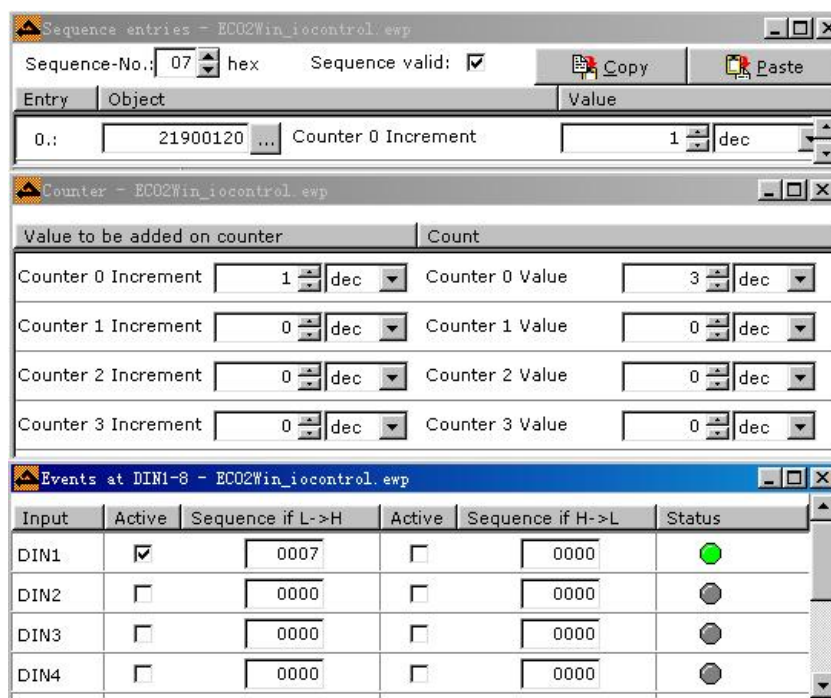
Figure 5.22 Counter Configuration

In the above example, whenever DIN1 changes from L to H, the SEQ07 will be called, and 1 is added to the counter 0 (0x21900120). In other words, it counts the input pulses of DIN1. It is very useful in many applications.

Counter objects are defined as follows:

Index [hex]	Subindex [hex]	Meaning
2190-2193	01	Increment, which is automatically added when the object is called
	02	Counter value

Table 5.12   Counter Objects

5.5.6 Calculator

The ED provides a calculator for data copying, arithmetic operation and bit logic operation and processing, as shown in the figure below:
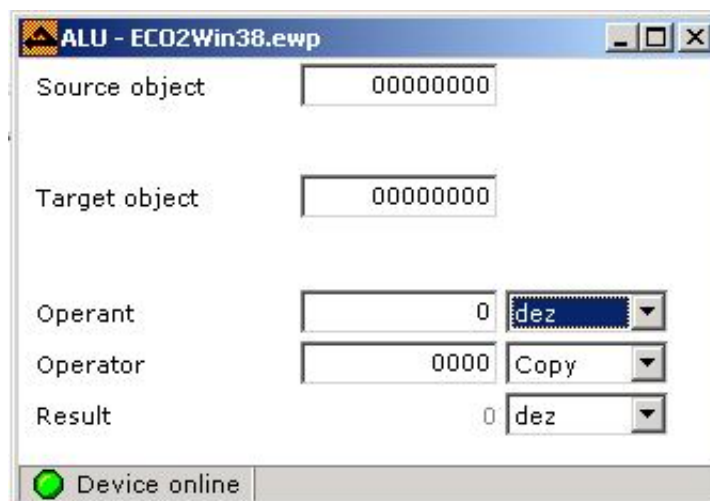
Figure 5.23   Calculator

All operations are executed via object 0x21A0. There are five registers Sub01 to 05. Sub01 (Source object) stores the source objects; Sub02 (Target object) stores the final results of operation (register index with mapping property); Sub03 (Operant) contains the numbers to be computed with source objects; Sub04 (Operator) stores operators (including eight operators, namely, COPY, +, -, *, /, AND, OR, XOR); Sub05 stores operation result. In the test, the operant can be modified by direct addressing. Enter the values for Sub01, Sub03 and Sub04, and finally enter target object in Sub02, and a new operation result is generated immediately. The computation in the sequence is the same.

Calculator objects are defined as follows:

Index [hex]	Subindex [hex]	Meaning
21A0	01	Source object
	02	Target object (contains computation result)
	03	Operand
	04	Operator
	05	Result of the operation; is transferred to target object

Table 5.13   Calculator Objects

5.5.7 Recording Table

The drive provides a 256 x 32 bit storage area (0x2D00~0x2DFF) for data storage. The user can use this area flexibly to store some data for query or secondary processing, as detailed below:
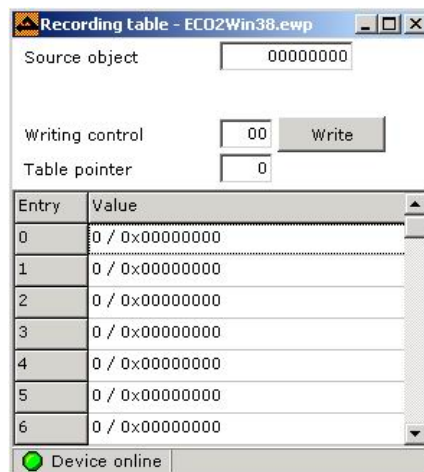
Figure 5.24   Recording Object Indexes

This function is implemented by the register 0x21B0. It contains three registers of Sub01 to 03. Enter the index of target register in the **Source object** (Sub01) box, enter address pointer Xx (00~FF) in the **Table pointer** (Sub03) box, and enter **1** in the **Writing control** (Sub02) box. Click <**Write**> to add the value entered in the **Source object** box to the address 2Dxx, and at the same time, 1 is added to the address pointer and it points to the 2Dxx+1 register.

Recording table objects are defined as follows:

Index [hex]	Subindex [hex]	Meaning
21B0	01	Source object, the value of which will be written in the table
	02	Write command
	03	Position in the table, value range: 0x00 to 0xFF

Table 5.14   Recording Table Objects

Note: The drive allocates a 64KB memory to store real-time sampling data, which can store up to four groups of records, with a minimum sampling interval of 1 ms (10 ms by default).

5.5.8 Position Capture

Via the very fast input "N" of the master encoder interface (X7, PIN4), the actual position of the servo system is captured and put into objects 0x21C003 and 0x21C004. The latter is the backup for the previous captured data.
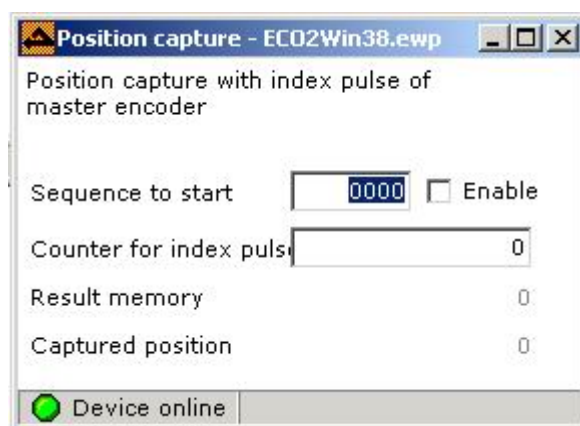
Figure 5.25   Position Capture

When the value in the Counter for index pulse box changes from 0 to 1, the value of actual position will be copied to the Result memory and Capture position. Select the Enable check box and the sequence set in the Sequence to start box will be started. To capture the actual position again, clear the value in the Counter for index pulse and set it to 1 again. When "N" of the master encoder input X7 changes from L (low level) to H (high level), 1 will be added to the Counter for index pulse field.

Position capture objects are defined as follows:

Index [hex]	Subindex [hex]	Meaning
21C0	01	Sequence number, which is called when the counter of the **21**C002 changes from 0 to 1
	02	Count for "N" pulses of X7
	03	Result memory: contains the actual position value
	04	Captured position: contains the actual position captured in the previous time

Table 5.15   Definition of Position Capture Objects

5.6 Device Configuration

The ED resource configuration consists of five parts, namely, digital inputs, digital outputs, electronic gear, analog input, and analog monitor, as described one by one below:
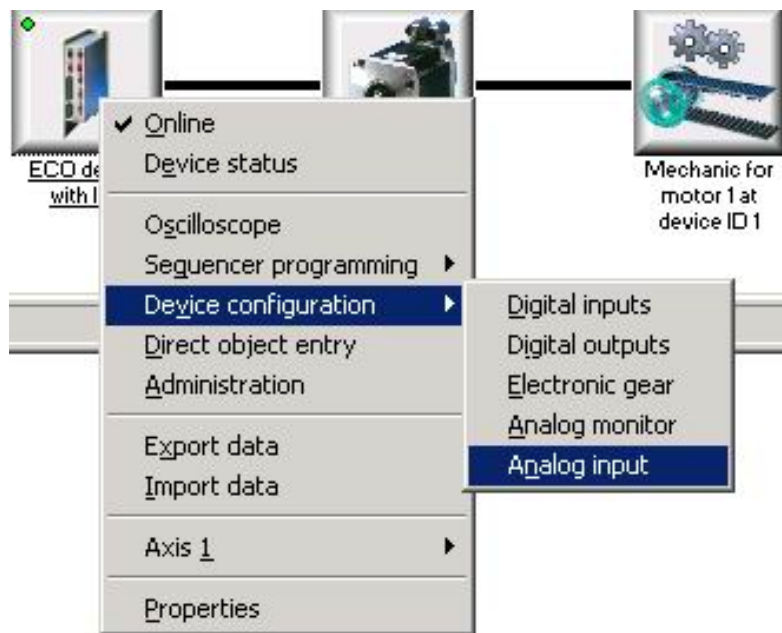
Figure 5.26   Device Configuration Menu

5.6.1 Digital Input

The ED drive has 10 digital inputs (Enable, Reset, 8 programmable DIN1-8, DC24V input), and their statuses are expressed by the register 0x60FD0020 (sequence from higher bits to lower bits is Enable, Reset, DIN8...DIN1). Note that the register has 32Bits, that is, 4 bytes (byte 4 – byte 1 from higher bits to lower bits). Where, **Enable** is used to enable the drive externally. If the drive does not receive this signal, it will generate an error. Reset is to reset the error status of the drive. Some functions of DIN1~8 have been described before. Some new functions are described as follows, as shown in the figure below:
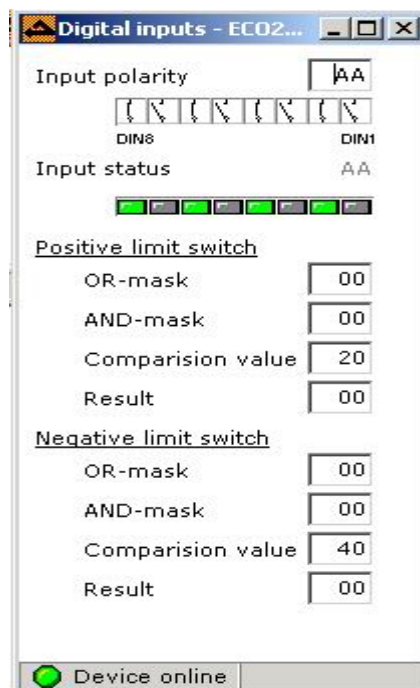
Figure 5.27 Digital Input

In the figure, the control commands correspond to the objects of 0x2170, 0x2171, and 0x2172 respectively. Where, Input Polarity corresponds to 0x21700008, and eight bits correspond to the input status conversion commands for DIN1 to 8. When x is 1, the polarity of the corresponding DINx changes. If the register value is 00, the polarity of the input is consistent with actual inputs. If the external input is 0x00 (inputs are all low), we enter AA for 0x21700008, and DIN8, 6, 4, 2 will be regarded as high level inputs. If the external input is 0x01 (DIN1 high level input valid), we enter 01 for 0x21700008, and the status of DIN1 will be switched to low level.

The object 0x2171 contains four sub-indexes used to describe DIN6 (LMT+) and to judge whether DIN6 is a high level input. Sub01 (OR-mask) is reserved for future use. We enter 20 (hex., that is, 0010 0000) in Sub02 (AND-mask) to screen input status other than DIN6. Enter 20 (hex.) in Sub03 (Comparison value), and the system will automatically perform "logic AND" operation between input status of DIN1 to 8 and the values in "AND-mask", and compare the result with the value in "Comparison value". If they are the same, Sub04 (Result) is set to 1, indicating DIN6 is a high level input; otherwise, Sub04 (Result) is set to 0, indicating DIN6 is a low level input. In this way, only the positive limit switch signal is valid.

0x2172 is similar to 0x2171, but it is used to describe DIN7 (LMT-). Therefore, AND-mask and Comparison value must be set to 60. In this way, the negative limit switch signal is valid.

5.6.2 Digital Output

The drive has two programmable digital outputs (24V, 0.5A), as described below:

Figure 5.28   Digital Output

We can see from the above figure that two outputs have the same control mode, that is, output after internal logic processing, as detailed below:
Two outputs correspond to register addresses of 0x2160 and 0x2161 respectively.

Index [hex]	Subindex [hex]	Name	Meaning
2160, 2161	01	Mapping data source	Mapping data source
	02	Offset	Offset
	03	AND-mask	AND-mask
	04	Comparison value	Comparison value
	05	Polarity	Polarity
	06	Auxiliary	Internal auxiliary value
	07	Interim result	Comparison output
	08	Result	Output interface

Table 5.16   Digital Output Objects

First, the user has to map an object address to Sub01 as the source data for computation. Then, perform the computation according to the steps below:

Step 1, Sub01+Sub02 = Result1

Step 2, Result1 and Sub03 = Sub07

Step 3, Compare Sub07 with Sub04.

If Sub05=0, the comparison result is True, and Sub08=1; if the comparison result is False, Sub08=0

If Sub05=1, the comparison result is True, and Sub08=0; if the comparison result is False, Sub08=1

Step 4, if the result of Sub08 is 1, the corresponding OUTPUT is of high level; if the result is 0, the corresponding OUTPUT is of low level.

For example, if OUT1 is to be set to **Target reached** and OUT2 is to be set to alarm output, the settings of the output are as follows:

Table 5.29   Example of Digital Output

We can see from the above settings that, to output the **Target reached** signal, set the 11th bit **Target**

reached in the status word 60410010 to 1. In other words, when the **Target reached** check box in the **Device Control→ Status word for Device state** is selected, OUT1 will be of high level. Enter 400 in the **AND-mask** box to screen other bits, with only bit 11 left to compare with the Comparison value field. Set the **Comparison value** field to 400 to ensure that the bit 11 is **True**. When the system is in the **Target Reached** state in the positioning mode, the comparison result of OUT1 is result=1 and the OUT1 is of high level. Likewise, the OUT2 is for alarm output by using the 60400010 status word. Where, bit4 is **Fault** and can be used to judge whether the drive has a fault.

In addition to two programmable digital outputs OUT1 and OUT2, the ED drive has three special functional outputs of Ready, Brake+ and Brake-. The **Ready** output is to output a high level value when the drive is in the "**Ready to switch on**" status, and to output a low level value when the drive has a fault or is not in the "**Ready to switch on**" status. Brake+ and Brake- are used to provide brake power supplied, with a voltage of 24V and a maximum of 1A (continue for 100ms and then reduced to 0.5A). The duration can be set in the **Parameters settings → Others → Holding brake delay** interface.

5.6.3 Electronic Gear

The parameter 0x2509 contains 11 sub indexes, used to set various parameters for master/slave mode and pulse/direction control mode. Firstly, map Sub02 to 0x60FF0020 (Target velocity). The control mode depends on Sub05. When Sub05=2, the system receives the pulse/direction signal from the X7 interface and implements the pulse/direction signal control mode. When Sub05=0, the system selects the master/slave mode, and the slave velocity = (Sub03/Sub04)*Sub02. When Sub05=3, the system selects the CW/CCW control mode, as shown below:

Figure 5.30 Electronic Gear

In the above figure, the parameters correspond to 11 registers of Sub01 to Sub0B from the top down. The value in the **Actual position master** field is the number of pulses received from the X7 interface and is used

to judge whether the X7 interface can receive external pulse signals.

Electronic gear objects are defined as follows:

Index [hex]	Subindex [hex]	Name	Meaning
2509	01	Mapping Slave position	Mapping Slave position data source
	02	Mapping Slave velocity	Mapping slave velocity, set to 60FF0020, that is, frequency value from the X7 signal is multiplied by the electronic gear ratio and then converted into target velocity of the motor
	03	Gear factor	Numerator of electronic gear ratio
	04	Gear divider	Denominator of electronic gear ratio
	05	Gear mode	Mode 0, 1: Master/slave, quadruple frequency multiplication 2: Pulse/direction signal control mode 3: CW/CCW signal control mode
	06	Actual position master	Number of pulses actually received from the master encoder interface
	07	Actual position slave	Number of pulses assigned to the slave after internal computation
	08	Actual velocity slave	Slave velocity
	09	Actual velocity master	Master velocity
	0A	External Master position	
	0B	External Master velocity	

Table 5.17　Electronic Gear Objectives

Note:

1. If the electronic gear ratio is no longer used, clear the data in 25090120 and 25090220; otherwise, the target velocity cannot be written in the velocity mode.

2. To validate electronic gear parameters, select <**Administration**->**Save parameters to device now**> to save the parameters after configuration, and then click <**Restart device now**> to reboot the drive.

5.6.4 Analog Monitor

The ED drive has two independent analog outputs to monitor the change of internal objects. They are usually used to monitor actual velocity and actual current. The **Analog Monitor** window in the ECO2WIN software is shown in the following figure:
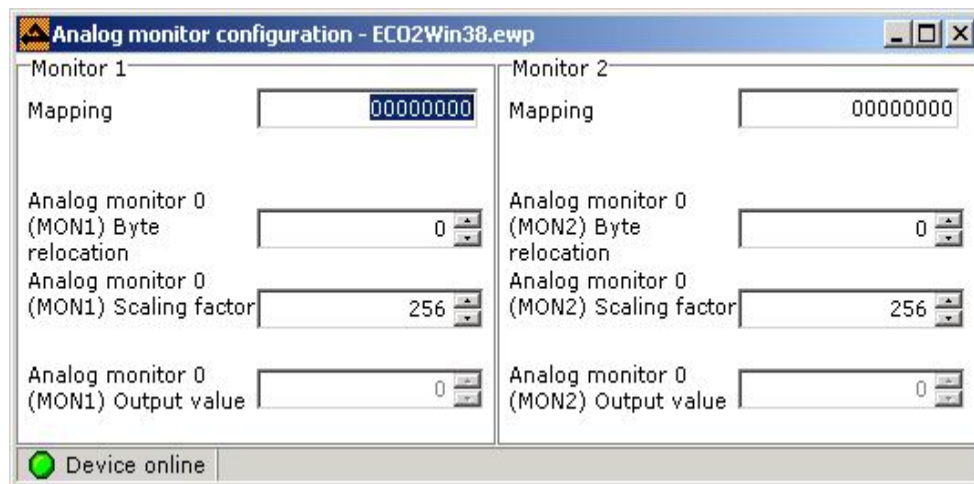
Figure 5.31   Analog Monitor

Two analog outputs correspond to the objects 0x2400 and 0x2401 respectively, and each contains four sub indexes. Sub01 is mapping data; Sub02 is Preshift; Sub03 is Factor; Sub04 is Output value. The ED adopts 9-bit D/A, with the value range of -256 ~ 255 and the output range of 0 ~ 5V. 0V corresponds to the number 0. The computation formula is as follows:

Output value= (Source value * factor)/($256^{(preshift+1)}$)
Actual Vout = 2.5V + Output value*2.5V/300
Here, it is important to select an appropriate factor to keep the Output value in the range of -256 ~ 255 and keep the Vout in the value range of 0.367V~4.625V. For details, refer to Appendix B, Sequencer Programming.

5.6.5 Analog Input

The ED has an analog differential input, adopts the 10-bit ADC, and corresponds to the internal object of 0x2508. It has six sub indexes. Sub01 is the mapping object index and can be target velocity, target position, or any other internal object, so that it can implement control over motor velocity and position by analog inputs. Sub02 is **factor**; Sub03 is **Shift**; Sub04 is **Analog input in** (data from ADC); Sub05 is **Analog input value** (result of multiplication); Sub06 is **Analog input highres state**; the ED provides the ADC resolution of 10 bits and is in the value range of -512 ~ 512.
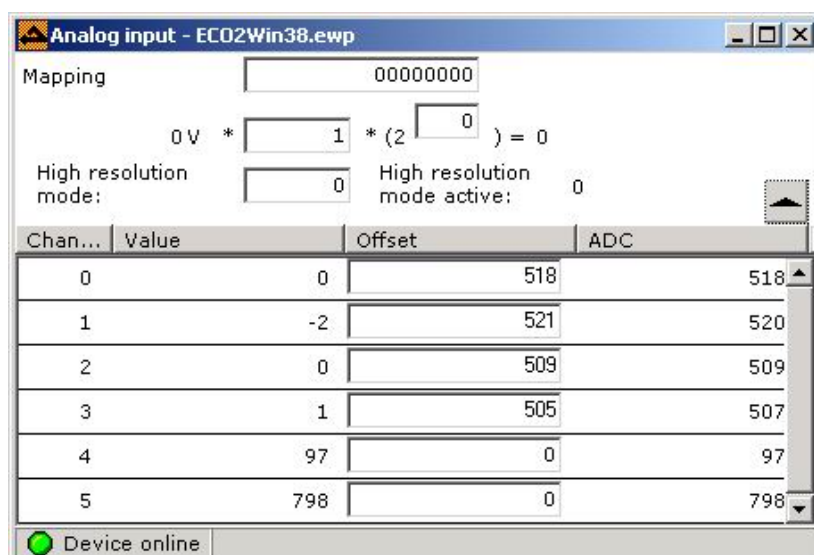
Figure 5.32

Computation formula: Sub01 = Sub04*Factor*(2^Shift)

The following example shows how to adjust the motor velocity by analog inputs. Suppose the input 10V corresponds to the maximum forward/reverse rotation of 1500 RPM, and the corresponding digital inputs Vin after ADC is -512 ~ 512.

Step 1: Map Sub01 to the register address 0x60FF0020 and then confirm the value of Factor. The internal velocity unit is inc/64s, and 8000inc/revolution. The value 1500RPM can be converted into the maximum velocity, that is, Vmax = (1500/60)*8000*64 (unit: inc/64s).

Step 2: Computation

Factor=Vmax/512/(2^Shift), suppose Shift=1, and then Factor=12500.

According to the above computation formula, we can obtain:

Target velocity Sub01=Vin*Factor*(2^Shift)=(Vin/512)*Vmax,

When input is 10V, Vin=512, Sub01=Vmax,

When input is 0V, Vin=0,Sub01=0,

When input is -10V, Vin=-512, Sub01=-Vmax

We can see from the above that the computation of Vmax is of key importance. Different mapping objects have different internal units and different input requirements. Ascertain the internal unit during conversion.

Note: When the control over the motor velocity by analog inputs is no longer required, clear the value in 25080120; otherwise, target velocity cannot be set in the positioning mode.

5.7 Communication

The ED servo is typical CANopen SLAVE equipment provided with dynamic PDO configuration capability. PDO configurations are saved to the equipment so that it may be started for operation when the Master (NMT Master) is absent. Communication can be used to observe the configuration of PDO by NMT Master or to configure the PDO manually.

There are four groups of common PDOs and each group consists of TxPDO and RxPDO. Corresponding IDs (the smaller the ID is, the higher priority it has) include TxPDO1 (181h~1ffh), RxPDO1 (201h~27Fh), TxPDO2

(281h~2ffh), RxPDO2 (301h~37Fh), TxPDO3 (381h~3ffh), RxPDO3 (401h~47Fh), TxPDO4 (481h~4ffh), and RxPDO4 (501h~57Fh). In the ED drive, RxPDO1 and TxPDO1, RxPDO2 and TxPDO2 have been pre-determined during initialization.

Note: The ED servo drive does not support self transmission and receiving function at present.

The PDO communication configuration is described later in this section. Each PDO consists of three communication parameters, ID (msg id), communication type (type), and inhibition time (inhibit time).

According to the CANopen standard, communication types cover:

0(0x00): Synchronous packet (non-periodic, PDO will be updated when the packet with synchronous ID (080h) occurs on the bus)

 1-240 (0x01-0XFO): Synchronous packet (periodic, sending data periodically in the synchronization time window, and the number indicates the number (1-240) of synchronous packets after which the PDO will be updated; mainly applicable to motion control)

252(0xFC): The request from remote frames updates the PDO synchronously and responds to the request.

253(0xFD): PDO continuous update, sending only after being requested by remote frames.

255 (0xFF): Non-synchronous (RxPDO will update upon each reception, and TxPDO will send packets in non-inhibition time whenever the content changes)

Default inhibition time is 1000 ms.

As shown in the figure below:

Figure 5.33 Communication Parameter

The following is a PDO configuration example:

A PDO has the CAN packet with the ID of 0x201 and it will be received by RxPDO1. For example, it may be the output of a CANopen I/O module, that is, it represents the digital input of this I/O module. This module has

64 digital inputs and it uses all 8 bytes available with a PDO. In the ED servo drive, these 8 bytes are mapped to general and mappable objects (INDEX 0x2100, SUB01…16) 0x2100, 01 and 02 (each data has a length of 4 bytes).

Another PDO has the CAN packet with the ID of 0x181 and it shall be transmitted as TxPDO1. It contains 32-bit output of the I/O module. The output value is mapped by 0x210003. Once the output content changes, 0x210003 will be transmitted at an interval of no less than 100 ms.

The third POD has 0x281 and will be transmitted by the ED servo drive when a Sync-Msg signal occurs on the bus.

At least two devices are required if we carry out a test. In other words, we need a participant to confirm the transmitting end, and meanwhile, we have to enable the CAN network for PDO communication, which may be done by a participating "NMT MASTER" device. If the NMT MASTER is absent, it is necessary to start the CAN-Node (an internal software module) of the ED drive. This can be done by setting the 0x1F8000 in the object dictionary to 3 (default value is 0) after the ED servo drive is powered on. At this time, the ED servo drive will send the NMT START information to the bus and start itself. Before setting the object value, make sure all other devices over the bus have been powered on and ready for receiving information. Starting the network for multiple times has no impact on the setting. Therefore, if the bus carries more than one ED servo drives, it is preferable to perform this configuration for each ED servo drive.

5.7.2 Tx -PDO Mapping Address Settings

Object addresses for TxPDO 1~8 are 0x1800~0x1807. Three specific parameters are represented by three sub indexes, as shown below:

Index (hex)	Sub index (hex)	Name	Meaning
1800~1807	01	tx_pdo_parameter_id	ID
	02	tx_pdo_parameter_type	Communication type
	03	tx_pdo_parameter_inhibit	Inhibition time

Table 5.18   Tx-PDO Definition

Mapping addresses of TxPDO 1~8 are 0x1A00~0x1A07. Specifically, 8 object addresses are represented in 8 sub indexes ranging from 01 to 08.   Each PDO can map up to 8 objects (8 bytes at the maximum). Therefore, you can map 2 x 32 bit objects, or 4 x 16 bit objects, or 1 x 32 + 4 x 8 bits objects, and so on. The following figure shows an example of TxPDO. Where, [**1**] indicates one object is defined; [**0**] indicates no object is defined, and [**3**] indicates three objects are defined. In the example, Tx-PDO1 sets one object: 21000320;TX-PDO2 sets three objects: 60FF0020 (Target velocity), 60600008 (Operation mode), and 60400010 (control command).
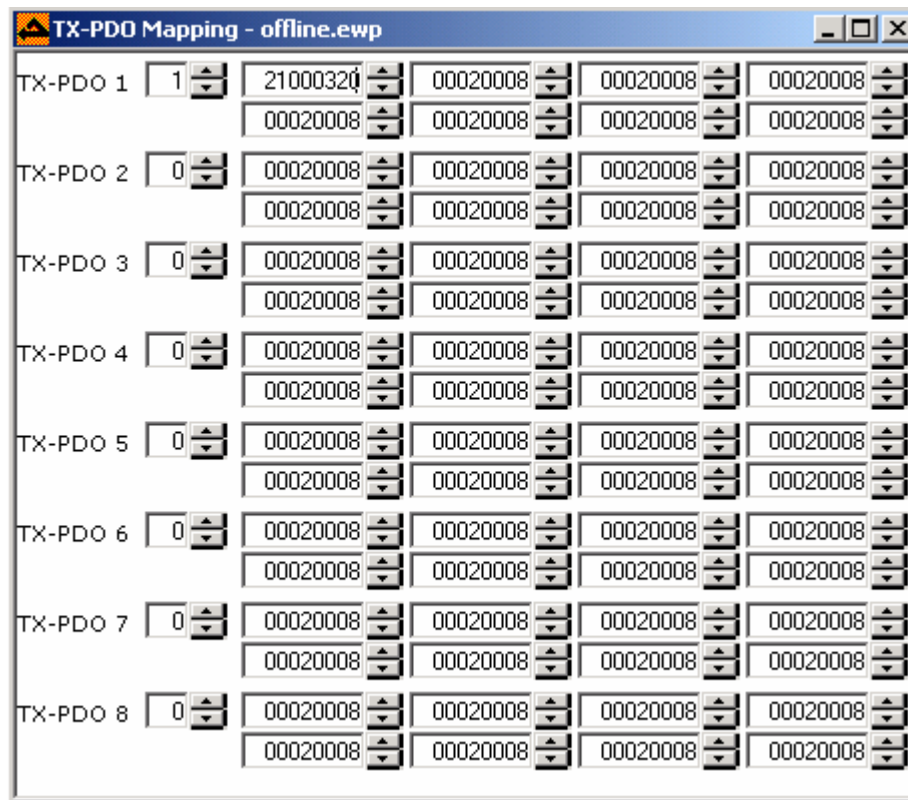
Figure 5.34   TX-PDO Settings

5.7.2 Rx-PDO Mapping Address Settings

Object addresses for RxPDO 1~8 are 0x1400~0x1407. Three specific parameters are represented by three sub indexes, as shown below:

Index (hex)	Sub Index (hex)	Name	Meaning
1400~1407	01	rx_pdo_parameter_id	ID
	02	rx_pdo_parameter_type	Communication type
	03	rx_pdo_parameter_inhibit	Inhibition time

Table 5.19   Rx-PDO Definition

Mapping addresses of Rx-PDO 1~8 are 0x1600~0x1607. Specifically, 8 object addresses are represented in 8 sub indexes ranging from 01 to 08. Each PDO can map up to 8 objects (8 bytes at the maximum). Therefore, you can map 2 x 32 bit objects, or 4 x 16 bit objects, or 1 x 32 + 4 x 8 bit objects, and so on. The following figure shows an example of RxPDO. Where, [**2**] indicates two objects are defined; [**0**] indicates no object is defined. In the example, RX-PDO1 sets two object: 21000120 and 21000220.
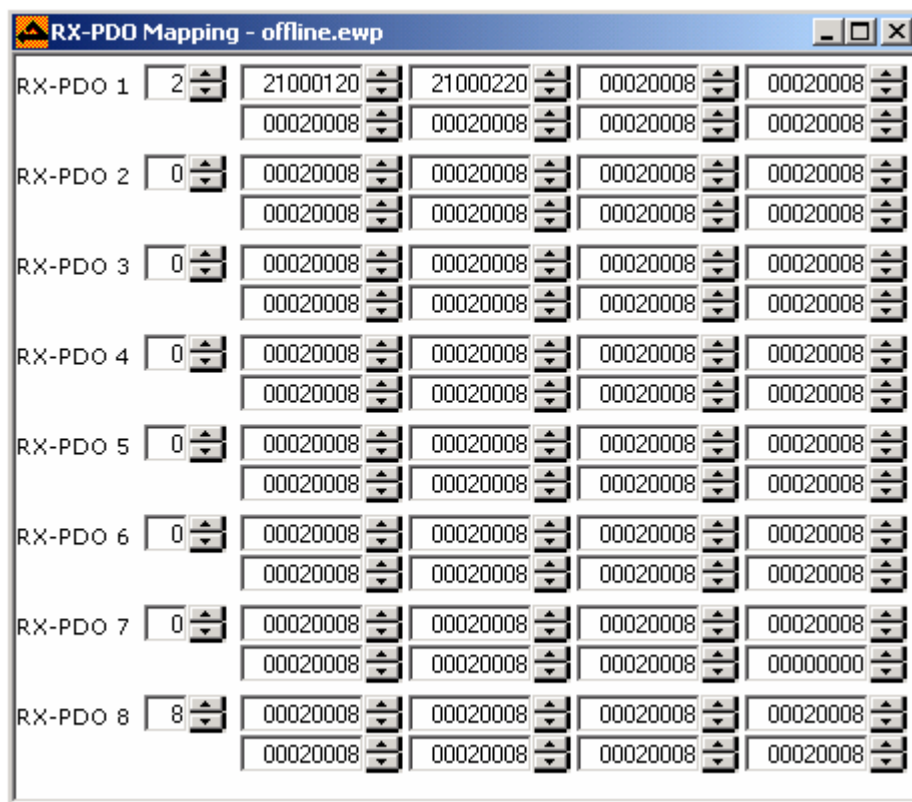
Figure 5.36　RX-PDO Configuration

5.8 Oscilloscope

As shown in the figure below, the software supports the oscilloscope function, which enables the system to monitor such operation parameters as velocity, position and current in dynamic curves. The oscilloscope is very important in judging the operation status and adjusting performance of the motor. It provides good visual effect when judging overshoot, overvoltage, and velocity stability. A brief introduction to its application is as follows:

1. **Setting monitor parameters**:

The oscilloscope has four monitor channels to monitor four different parameters of the drive. If you adopt a RS-232 interface for communication with the drive, the collection speed will be low. In general, it is recommended that only the most important parameter be selected and other channels be cancelled.

2. **Setting sampling time**:

The parameter scanrate (ms) is used to set sampling period, and the parameter **Number of value** is used to set the number of sampling points. In the example below, the sampling period is 10 ms and the display data will be refreshed after each 400 collections. Usually, the refresh rate is greater than the result of the above two parameters due to the communication time.

3. **Conditions for data sampling**:

To trigger data sampling, the sampling condition must actually happen. Firstly, set the sampling condition in **Trigger on signal** frame. In the example below, the trigger condition is that the actual velocity is 0 because, in this example, the motor takes position 0 as the midpoint and will conduct the reciprocating movement of 40000inc. Therefore, the condition of actual speed = 0 can happen. Such setting is to ensure the occurrence of trigger condition; otherwise, you will not collect any data. During data collection, the mouse pointer changes to the shape of a hourglass.

When the oscilloscope is working normally, ensure that the motor works in a desirable mode. In the following example, the motor is set to **Autoreverse** mode, with the reciprocating position range of -40000 to 40000 and the reciprocating speed of 200000inc/s; the operation mode is -3.
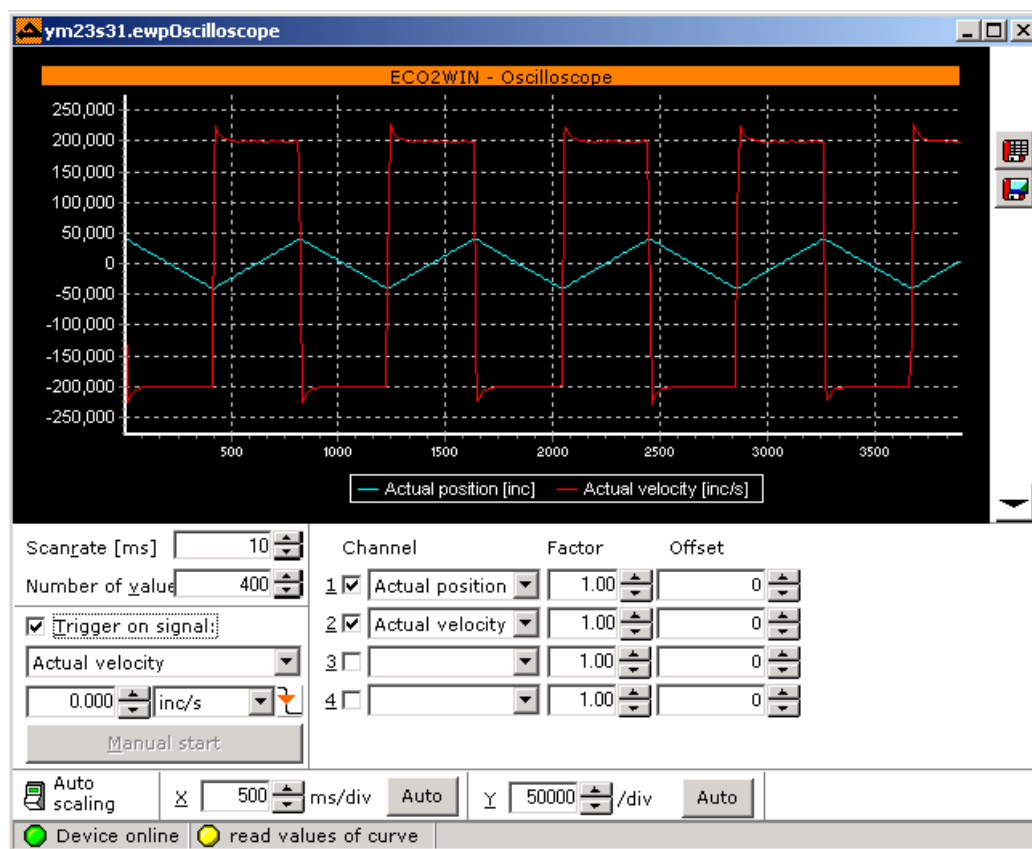
Figure 5.36   Oscilloscope

The above figure shows the velocity and position monitoring curves of a motor in auto reverse mode. The reciprocating speed is 200000inc/s, the reciprocating position range is -40000 to 40000, and the operation mode is -3. Channel 1 is set to Actual position and channel 2 is set to Actual velocity. Red line indicates velocity curve and green line indicates position curve.

Upon the completion of data collection, you can save the collected data in the format of data or graph for comparison:

is used to export the collected data into a text file; is to save the collected curves as graphs.

5.9 Administration
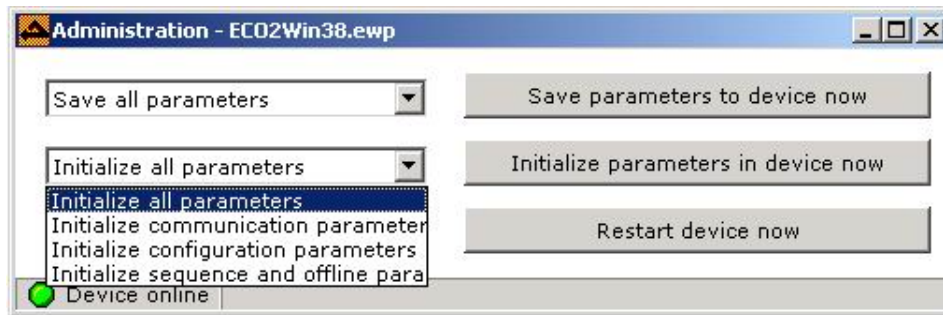
Figure 5.37

The administration function covers saving parameters, initializing parameters, and restarting the device. The most frequently used function is saving parameters. Whenever we modify the parameters in the ED drive, remember to save the modifications so that you don't have to configure or download parameters again when the drive is powered on in the next time. This is irrelevant to the saving of project.

To restore drive parameters to factory default settings, click <**Initialize parameters in device now**> and then click <**Restart device now**>. Then, the **Error** LED on the front panel of the drive turns red and will then turn green in several seconds, indicating the initialization succeeds.

Chapter 6 Trial Run of Motor

In this chapter, we will first describe the trial run of the motor and then detail on he functions of the drive.

6.1 Installation of Electrical Components

To run the motor, the following components have to be configured:

● Logic voltage 24VDC (low voltage, isolated from 230 VAC), the logic power supply provided for the control circuit of the drive.

● Power supply (24VDC~150VDC)—Power supply provided for the drive.

● ED200 or ED100 drive + wiring terminal (power terminal X10, motor terminal X9, I/O terminal X3, logic power terminal X4)

● Motor and related cables (motor cable, encoder cable, RS-232 programming and communication cable)

● A mechanical structure with homing and limit switches (for example, guide rail)

Figure 6.2

Figure 6.1

6.1.1 Connecting Encoder Cable to Drive

The motor encoder signal is defined according to RS-422. The X8 interface of the drive provides the encoder with 5V power supply, so that no external power supply is required.
The encoder cable is connected to the X8 interface of the drive. For definition of encoder cable signals, refer to Section 3.4.6.

6.1.2 Connecting Motor Cable (with/without Brake) to X9 Interface of the Drive

Connect four phase lines of the servo motor to X9 (A, /A; B, /B), connect the ground wire (yellow/green wire)

to GND of the X9 interface, and connect BRAKE+ and BRAKE- of the X9 interface to the brake cables (24V, 1A). All cables must be fixed securely to racks. Make sure to connect the ground wire of the motor (34S/42S series motors) or shielded cable (23S series motors) to the drive (for 34S/42S motors, connected to the ground screw of the drive; for 23S series, to the GND of the X9 interface; shielded cables must be pressed securely by the clamping screw, as shown in the figure below). For wiring sequence, refer to Section 3.4.6.
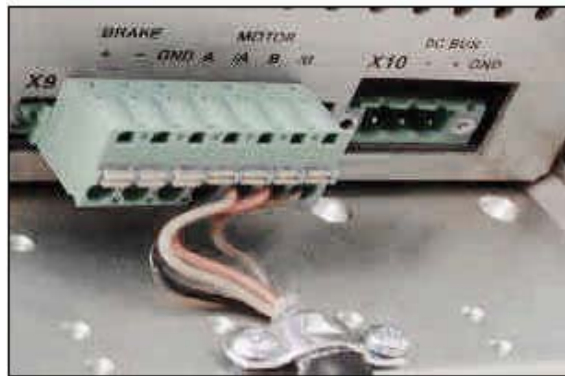
Figure 6.3 Motor Connection

6.1.3 Providing Logic Power Supply for Drive (Controller Power X4)

Valid value range for logic power voltage: 18V – 30V DC, to +24V and GND of the X4 interface.

6.1.4 Hardware Reset

If the wiring is correct, the RUN indicator blinks and the 24V indicate will turn green. If the BUS indicator goes on, it indicates the power supply for the drive is not connected. If no error is found, power on the drive again or short-circuit the RESET and 24V signals of the X4 interface to remove the alarm.

6.1.5 Encoder Signal Check

The drive is in the basic state. If you connect the drive to a PC via the RS-232C interface, open the ECO2WIN software, and you can view the encoder data (**Axis1->Device control->Movement->Actual position**) while rotating the motor shaft manually. If the data does not change, please check whether the encoder cable is connected correctly. If the **Master Encoder** interface (X7) is also connected to a master encoder, you can monitor the change of master encoder data in the functional module (**Device configuration->Electronic gear->Actual position master**) when rotating the master encoder. If the data does not change, it indicates that the master encoder data is incorrect and does not comply with RS-422 standard.

6.1.6 Connection with External Enable Signal

ENABLE of the X4 interface is an external enabling interface. The +24V terminal of the drive can be directly connected to the ENABLE terminal. If the ENABLE terminal is not connected, the drive will generate an "external enable low' fault when entering the enable command.

6.1.7 Providing BUS Power (X10) for Drive

Connect the power supply of the drive. For ED100, the power supply range is 24VDC ~ 70VDC. For ED200 and ED216, the power supply range is 24VDC ~ 170VDC. For wiring methods, refer to Section 3.2.1, ED100

Series Drives, and Section 3.2.2, ED200/ED216/ECOLIN200/ECOLIN216 Series Drives.

6.1.8 Connection of Programming Cable

You can purchase the programming cable from us or make it by yourself according to the definition below:
Programming cable:
PC COM1, COM2   ED   X5
        RxD 2   --------------------- 2 TX
        TxD 3   --------------------- 3 RX
        GND 5   --------------------- 5 GND
Up till now, the electric wiring of the motor is completed.

6.2 Trial Run of the Motor

Before trial run, make sure:
- All power supplies connected with the drive must be isolated from the bus power supply.
- Actual maximum voltage cannot exceed the rating.
- Motor shafts must be suspended in the air before trial run, that is, no load on the motor.

In addition, check and confirm all parameters are correctly set and make sure the maximum current is set according to the motor model before motor operation. After all necessary checks are made, run the motor.

6.2.1 Setting Motor Parameters

First, create a new project file or open an existing one with the ECO2WIN software. For details, refer to Section 4.2.2 and Section 4.2.3. If the user selects the motor model when creating a project file, the **Parameter setting->Commutation** window (See Section 5.4.4) will display the following information:
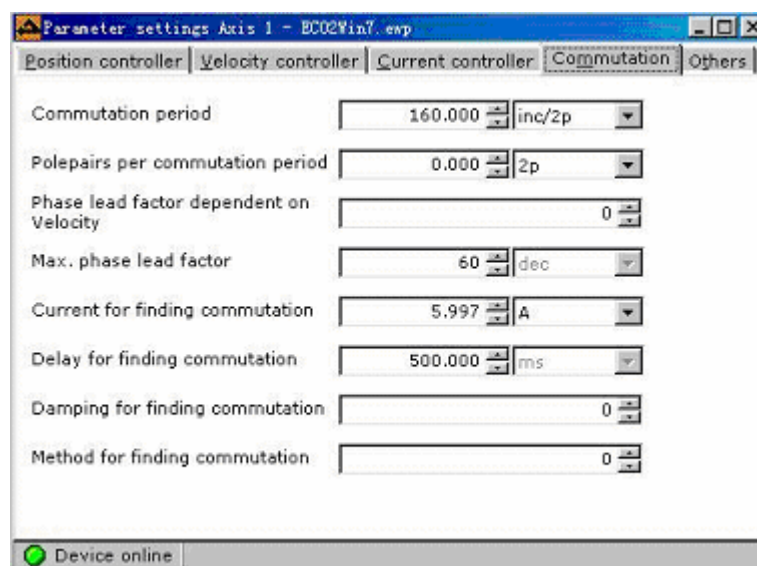
Figure 6.4

For a Kinco rotative motor, the parameters are preset as shown in the above figure. The parameter "Phase lead factor dependent on Velocity" is different and it is to be set according to the requirements set out in Section 5.4.4. For a Kinco linear motor, set the motor according to the requirements described in Section

5.4.4.

6.2.2 Current Adjustment

Adjust the maximum current. If a correct motor model is selected when creating a project file, the maximum current of the motor is already set. For details, refer to Section 5.4.3, Current Controller.

6.2.3 Checking Velocity Controller and Position Controller Parameters

Finally, check default settings of the velocity controller and position controller. The load-free parameters are set as follows:
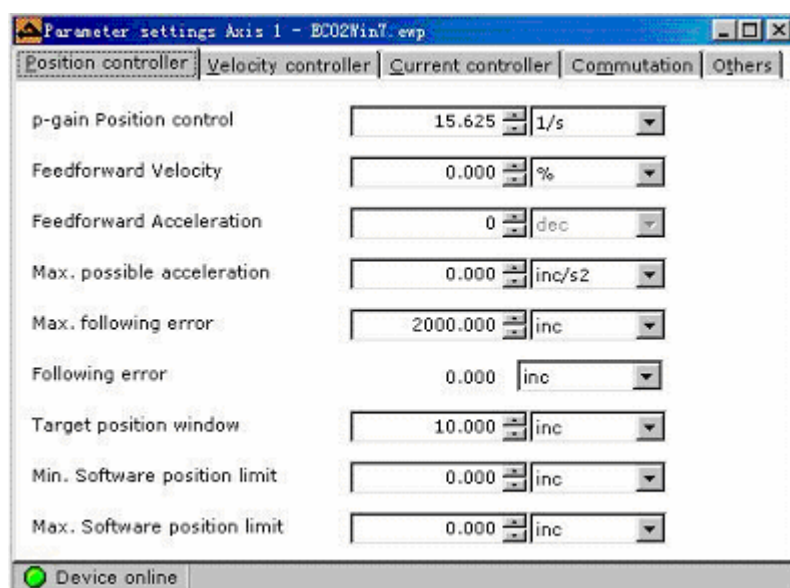
Figure 6.5 Default Settings of Position Controller
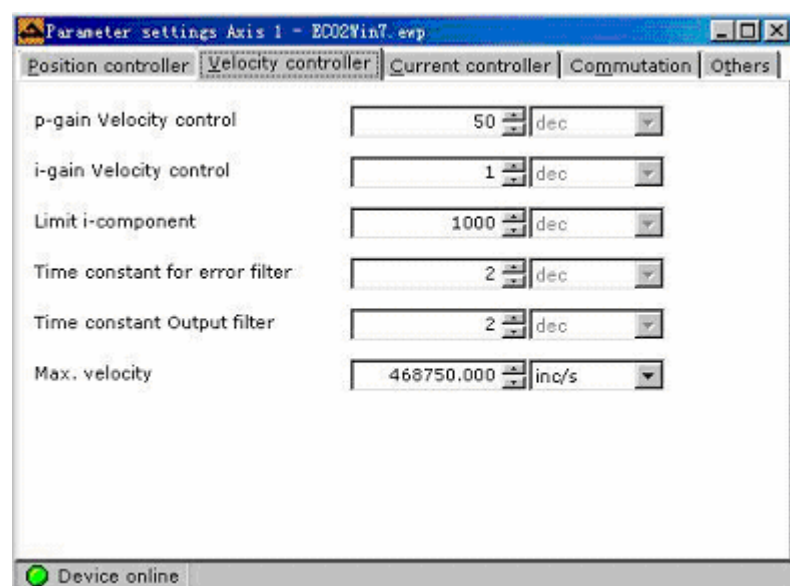
Figure 6.6 Default Settings of Velocity Controller

6.2.4 Status Control

Up to now, we have set all parameters and can start the motor. In the **Device Control** window, set the operation mode to velocity mode -3, and set Target velocity = 8000inc/s, and set Control word = 0x06. If the drive does not generate an alarm, it indicates the wiring is correct. Then, set Control word=0x0f, and the motor will run at a velocity of 60 RPM. At this time, you can change the **Target velocity** value directly. To stop the motor, set **Target velocity**=0, but the motor shaft is still locked. To loosen the motor shaft, set Control word=0x06.

Run the motor in different operation modes with a reference to Section 5.3.

Steps:

1. Connect the logic and power supply of the drive.
2. The shaft is freely moveable, without any load.
3. The 24V LED turns green and the RUN LED blinks in green.
4. In the **Device Control** window, set operation mode to -3 and control word to 0x06.
5. Set control command to 0x0F
6. Check whether bit Commutation Found in the Device State Control is 1, or whether the status word = 0x4437, or whether the motor shaft is locked.
7. Set Target velocity=8000inc/s
8. The motor will run at a velocity of 60 RPM.
9. If the drive generates an alarm, troubleshoot according to section 5.3.6, Error Diagnosis. Under default settings, the drive is likely to report a "following error" fault. At this time, it is necessary to adjust the Kpp (p_gain Position control) value and Vff (Feedforward velocity) value of the position controller. In general, set the Kpp to 100-200 (1/s) and Vff to a value above 75%. The "following error" will be reduced and the drive will no longer report the "following error". To adjust parameters, it is necessary to read through this manual and have certain field experience. Fortunately, only a few parameters have to be adjusted.

Chapter 7 Parameter Setting in Forward/Reverse Rotation Mode

7.1 Introduction to Forward/Reverse Rotation

In this mode, the drive works in the forward/reverse rotation state, and the parameters of the velocity controller and of the position controller will be optimized.

The **Auto reverse** window consists of two parts: in the upper part, the user can set limit values; in the lower part, the user can set target values. Once reaching the limit values, the motor will move in an opposite direction towards new target values.

ECO2WIN is a user interface for the ED drive. All control commands are stored in the drive and will not get lost even if the ECO2WIN software exits. In other words, even if the ECO2WIN and Window software cannot work, the drive can work normally.

Note: Make sure the power supply can be switched off immediately at any time.

7.2 Introduction to Different Forward/Reverse Rotation Modes

7.2.1 Timed Forward/Reverse Rotation with Position Control

Open the window **Control Axis1- Autoreverse**.
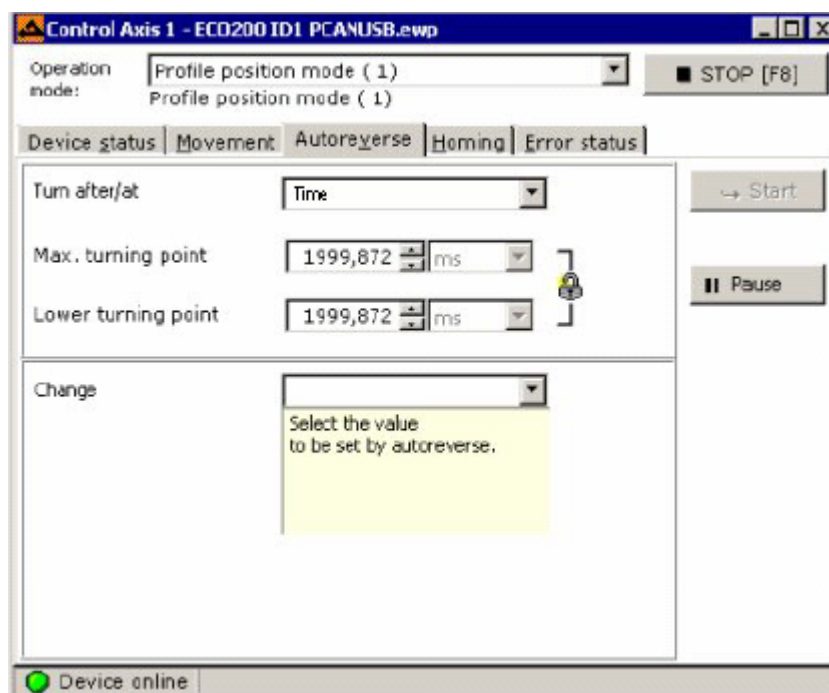
Figure 7.1   Autoreverse

1. Enter a value in the **Max. turning point** box, for example, 2000 ms.
2. Click the lock symbol. The value in the **Lower turning point** box is adjusted to the same value in the

Max. turning point box.

The time values shown may vary slightly from the entered values because of internal conversion (for example, 1999,872 instead of 2000 ms).

Select **Position** in the **Change** frame:

1. Enter a value for the **Positive movement** that does not lead to a mechanical collision.
2. Enter a value for the **Negative movement** that does not lead to a mechanical collision.
3. Click the lock symbol on the right. If it is locked, the values for **Positive movement** and **Negative movement** are set symmetrically to the zero position.
4. Change to the **Movement** tab and set **Velocity on positioning** to a small value at first.
5. Click <**Start**> in the **Autoreverse** window to start the motor.
6. Click the <**Pause**> button, and the motor stops running, but the motor shaft is still locked.

Note: Press <**F8**> or <**Stop**>, and the motor shaft will be loosened immediately.
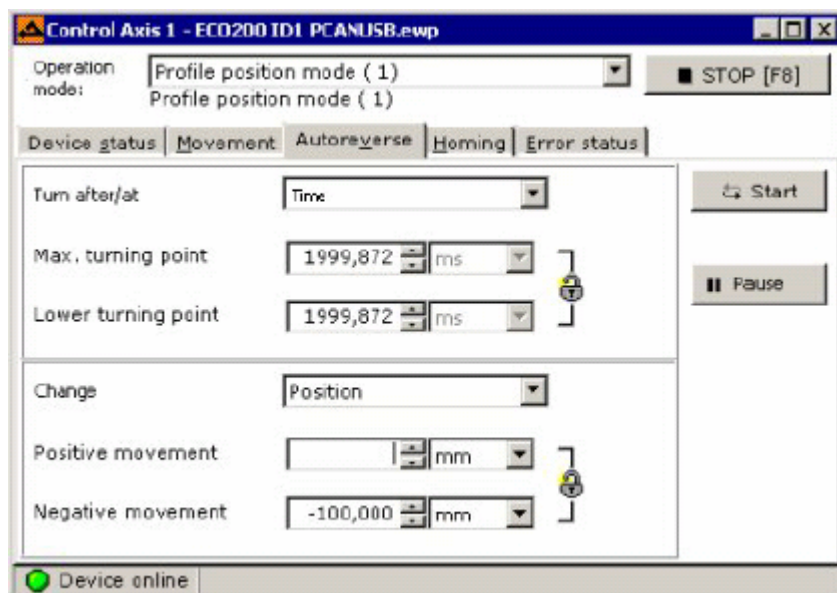
Figure 7.2 Autoreverse

According to the data shown in the above figure, the motor will run in the timed forward/reverse rotation mode in 2000ms. During the period, the motor will:

- Accelerate to the defined velocity at the acceleration defined in **Movement**.
- Decelerate according to the deceleration defined in **Movement**.
- Stop in a designated position.

If the motor reaches the positive or negative position in less than 2000 ms when it runs at the set acceleration/deceleration and target velocity, the motor will pause until 2000 ms is reached. Then, it will move in the opposite direction to the set position. Time values and position values can be set even when a motor is running.

In the Movement tab, the following parameters can be changed:

- Velocity
- Acceleration
- Deceleration

If the ERR LED of the drive blinks, normally a following error occurs. Check it in the **Error status** window.

Reduce the values for velocity, acceleration and deceleration in the **Movement** tab. Increase the values for **Max. Following Error** in the **Controller setting axis 1->Position controller** tab if necessary.
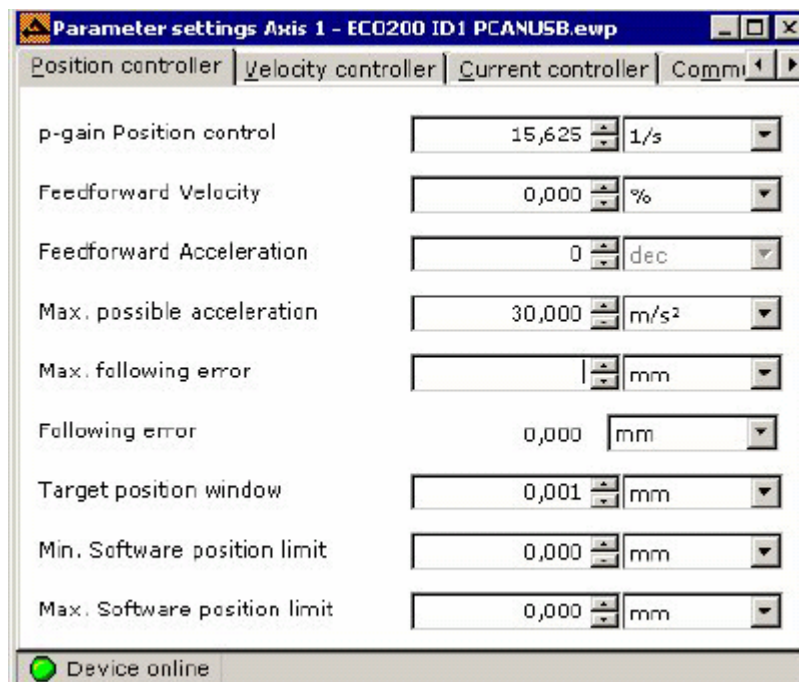
Figure 7.3 Position Controller

7.2.2 Timed Forward/Reverse Rotation with Speed Control

For timed forward/reverse rotation with speed control, it is necessary to set the operation mode of the servo drive to -3, as shown below:
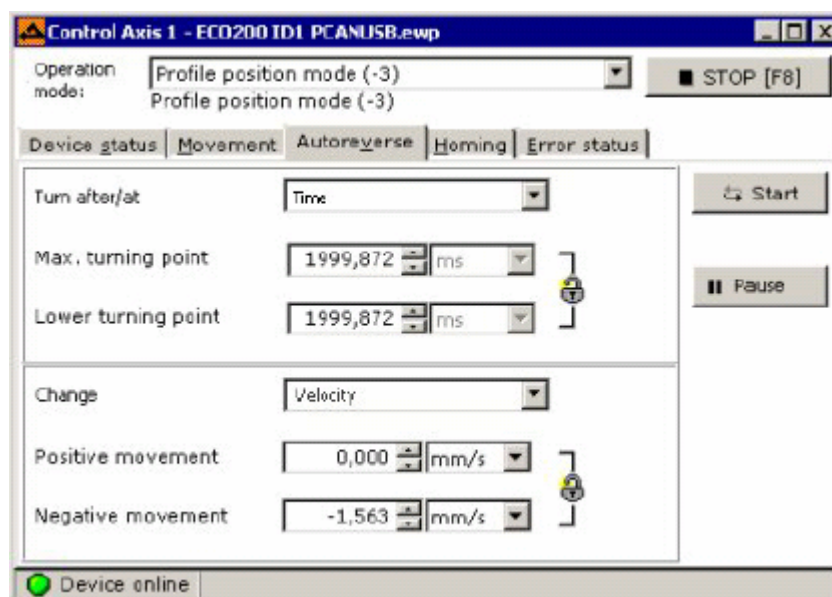
Figure 7.4   Autoreverse

Operation mode -3 is pure velocity control without regarding acceleration and deceleration. No following error occurs because this operation mode has no position control.

The maximum current of the motor has considerable impact on this mode. The velocity curve can be influenced by the following parameters in the **Velocity controller** tab:

- p-gain position control
- i-gain Velocity control
- Time constant for error filter
- Time constant output filter

Adjust these parameters in a way that provokes little overshoot and produces better velocity performance.

7.2.3 Position Forward/Reverse Rotation with Speed Control

For forward/reverse rotation on certain positions with speed control, set the operation mode to -3.
Operation mode -3 is pure velocity control without regarding acceleration and deceleration. No following error occurs because this operation mode has no position control.

The maximum current of the motor has considerable impact on this mode. The velocity curve can be influenced by the following parameters in the Velocity controller tab:

- p-gain position control
- i-gain Velocity control
- Time constant for error filter
- Time constant output filter

Adjust these parameters in a way that provokes little overshoot and produces better velocity performance.

7.2.4 Speed Forward/Reverse Rotation with Position Control

For auto forward/reverse rotation at a defined velocity, set the operation mode to 1. To get a periodic forward/reverse rotation, the limit velocities defined in the upper part of the window have to be reached within the positions defined in the lower part of the window. Therefore, suitable acceleration and deceleration have to be set.

As a typical position control, set acceleration and deceleration values in the **Movement** tab.

Additionally, the value for **Velocity on positioning** in the **Movement** tab must be greater than the value for velocity defined in the upper part of the **Reversing** tab. Otherwise, the system cannot reach the velocity defined in the **Reversing** tab and cannot perform periodic forward/reverse rotation.

Asymmetric values for forward/reverse rotation speed in the **Reversing** tab lead to drifting in one direction and finally to a stable end position without forward/reverse rotation.

Chapter 8 Limit Switch and Homing

This chapter describes how to configure position limits and how to select different homing modes, which is of vital importance to a system with starting and ending positions.

8.1 Limit Switch and Homing Switch

To set a reference point of a linear or rotative axis, at least one electric switch is necessary. "Switch" means mechanic switches as well as electronic sensors.

Inputs DIN6 – DIN8 are limit position switches and homing switches.

Input	Use
DIN6	Positive limit switch
DIN7	Negative limit switch
DIN8	Homing switch, positioned between the limit switches

In addition to hardware wiring, to set the DIN6 and DIN7 as positive limit and negative limit inputs, it is necessary to set corresponding parameters in the **Device configuration->Digital input** menu, as shown in the figure below:
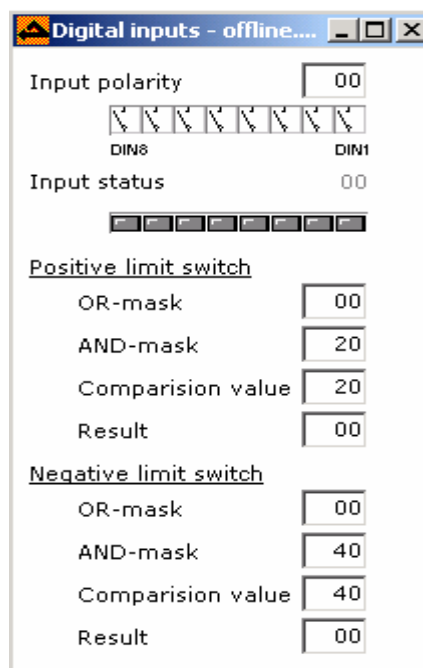
Figure 8.1 Digital Inputs

To validate the positive limit, enter 20 in the AND-mask of the positive limit switch.
To validate the negative limit, enter 40 in the AND-mask of the negative limit switch.

8.2 Search Principle

The control mode 6 is Homing mode. In this mode, the ECO2WIN searches the reference point automatically. In mode 6, the following objects have to be defined:

Object	Name	Value	Meaning
60600008	Chosen Mode of operation	6	Set the operation mode to mode 6
60980008	Homing method	User setting	Homing method
607C0020	Relocation zero point	User setting	Offset relative to the zero point
60990120	Velocity during search for Reference switch	User setting	Velocity during search for limit or reference switch
60990220	Velocity during search for Reference set point	User setting	Velocity during search for Reference set point
609A0020	Acceleration during homing	User setting	Acceleration during homing
60400010	Control word for Device state	0x1F	Motor starts operation in this mode (start searching)

Table 8.1   Definition of Homing Objects

Note: A +24V level signal is required for activation of all switches (active HIGH).
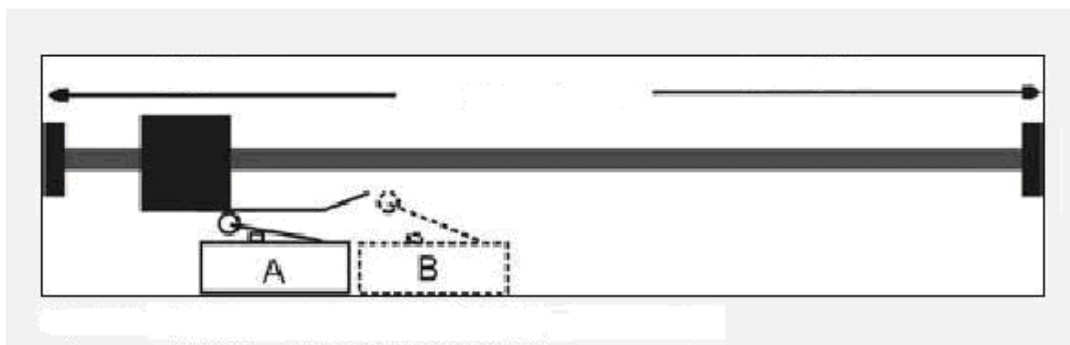
Figure 8.2   Homing

The counting and moving directions in Figure 8.2 are the same as those described below. The counting direction of the user system can be judged by "**Axis1->Device Control->Movement, Actual position**". If the value of Actual position is increased, it indicates the positive direction. If the direction is opposite to the direction expected by the user, the user can set the address value 607E0008. The default value is 0. To change the counting direction, change the value to 80. The user can also change the value by **Position polarity** and **Velocity polarity** in the **Parameters setting->Others** window, as described earlier in the document.

8.3 Homing Modes

8.3.1 Homing Mode 1: Homing with Negative Limit Switch

DIN7 is connected with the negative limit switch signal.

Negative

Counting d
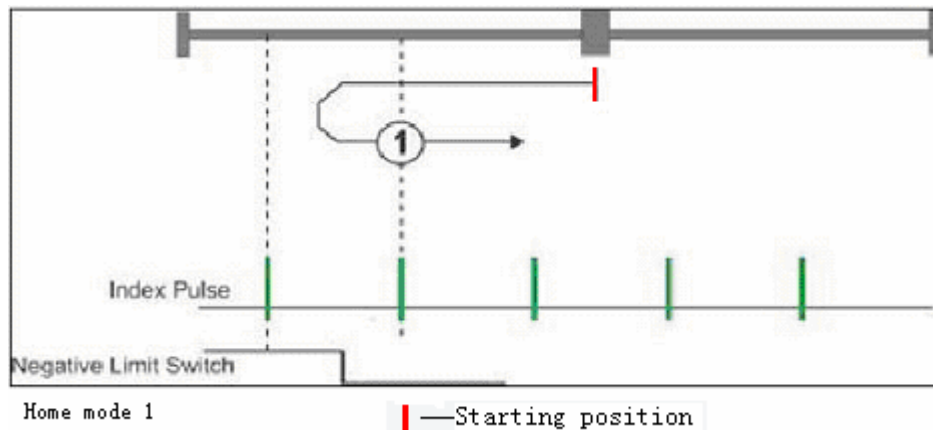
Figure 8.3   Homing Mode 1

The limit position switch is in the negative counting direction. The system enters the negative direction until the limit switch switches to +24V. Then, the axis drives into the positive direction up to the first index impulse signal of the motor encoder. The position corresponding to the index nearest to the stop position in the negative direction is the reference in the mode. Then, the "reference found" is set in the status word and the system is decelerated till standstill.

Advantage: The reference point is connected absolutely to the motor and mechanical part.
Disadvantage: If the motor is changed or the coupling is loosened, the system must be re-calibrated.

8.3.2 Homing Mode 2: Homing with Positive Limit Switch

DIN6 is connected with positive limit switch signal.
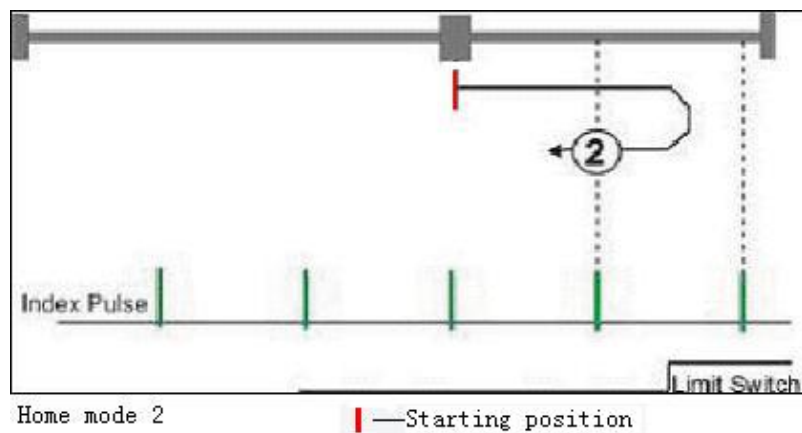
Figure 8.4   Homing Mode 2

The limit switch is in the positive counting direction. The system enters the positive direction until the limit switch switches to +24V. Then, the axis drives into the negative direction up to the first index impulse signal of the motor encoder. The position corresponding to the index nearest to the stop position in the positive direction is the reference in the mode. Then, the "reference found" is set in the status word and the system is decelerated till standstill.

Advantage: The reference point is connected absolutely to the motor and mechanical part.
Disadvantage: If the motor is changed or the coupling is loosened, the system must be re-calibrated.

8.3.3 Homing Modes 3 and 4: Homing with Positive Homing Switch and Index

DIN8 is connected with the positive homing switch signal:
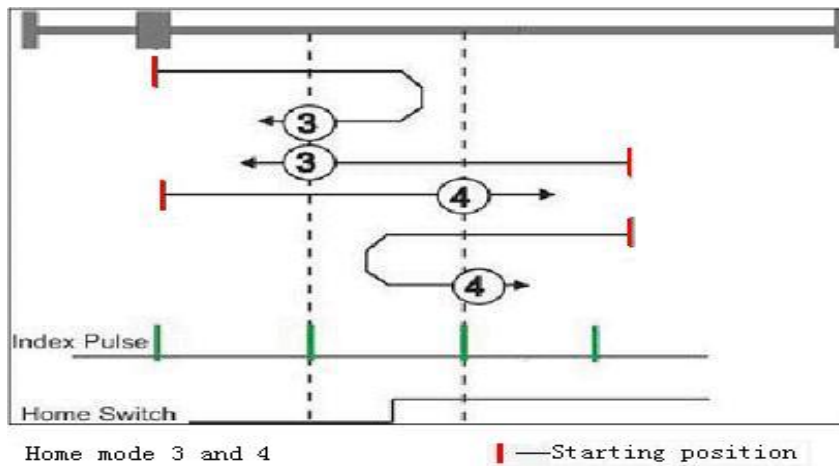
Figure 8.5   Homing Mode 3 and 4

The drive moves along the homing switch direction and stops when detecting the instruction signal from the homing switch. According to the level of the homing switch the drive turns either in positive or negative direction. As soon as the change of homing level is recognized, the next index pulse signal of the motor encoder is recorded as the reference point. The bit "reference found" is set in the status word and the axis is decelerated till standstill.

    Advantage: The reference point is connected absolutely to the motor and mechanical part.
    Disadvantage: If the motor is changed or the coupling is loosened, the system must be re-calibrated.

8.3.4 Homing Modes 5 and 6: Homing with Negative Homing Switch and Index

DIN8 is connected with the negative homing switch signal:
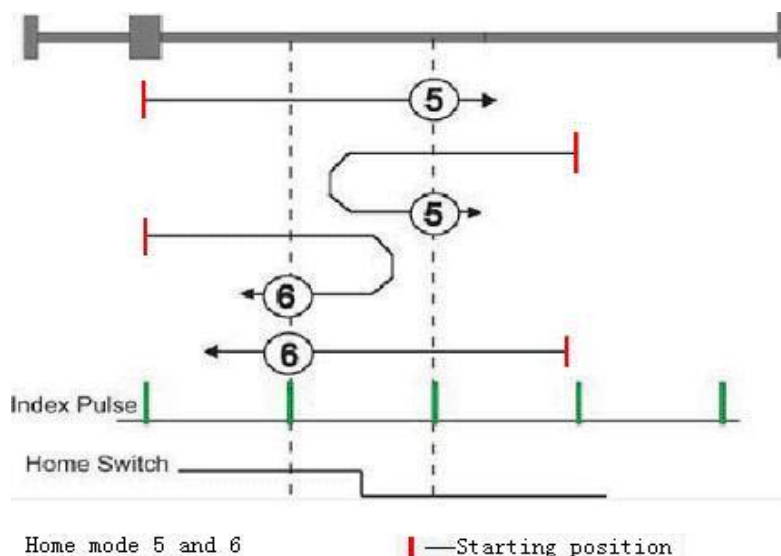
Figure 8.6   Homing Mode 5 and 6

The drive moves along the homing switch direction and stops when detecting the instruction signal from

the homing switch. According to the level of the homing switch the drive turns either in positive or negative direction. As soon as the change of homing level is recognized, the next index pulse signal of the motor encoder is recorded as the reference point. The bit "reference found" is set in the status word and the axis is decelerated till standstill.

Advantage: The reference point is connected absolutely to the motor and mechanical part.

Disadvantage: If the motor is changed or the coupling is loosened, the system must be re-calibrated.

8.3.5 Homing Modes 7 and 10: Homing with Homing Switch, Index and Positive Initial Movement
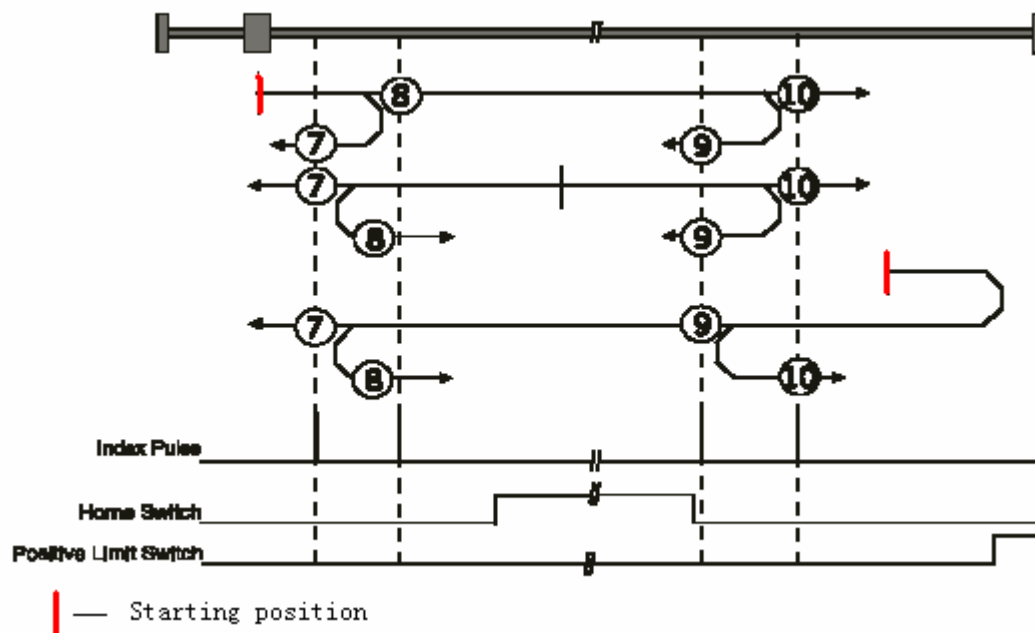
Figure 8.7   Homing Modes 7 to 10

Figure 8.7 shows the homing mode defined by the DIN8 homing switch signal and index signal, with the initial movement direction as positive direction.

8.3.6 Homing Modes 11 and 14: Homing with Homing Switch, Index and Negative Initial Movement
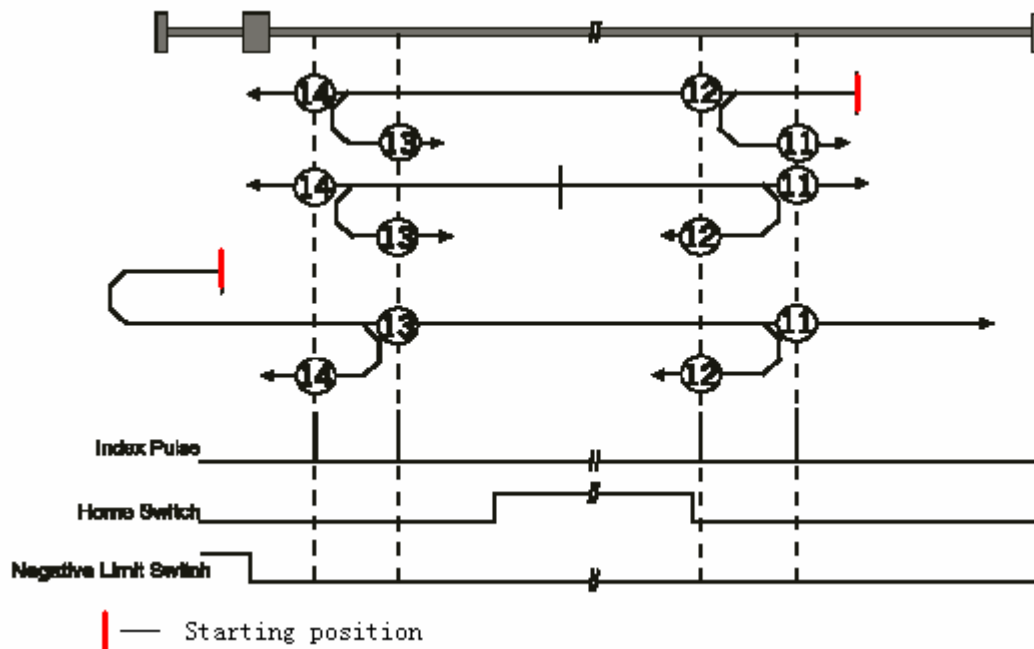
Figure 8.8　Homing Modes 11 to 14

Figure 8.8 shows the homing mode defined by the DIN8 homing switch signal and index signal, with the initial movement direction as negative direction.

8.3.7 Homing Modes 15 and 16: Reserved

They are reserved for future use.

8.3.8 Homing Modes 17-31: Homing without Index Pulse

Homing modes 17-31 correspond to homing modes 1-14, without the need of the index pulse signal. Instead of the index pulse signal, only the H-L-edge and L-H-edge of the limit switch or the homing switch is evaluated.
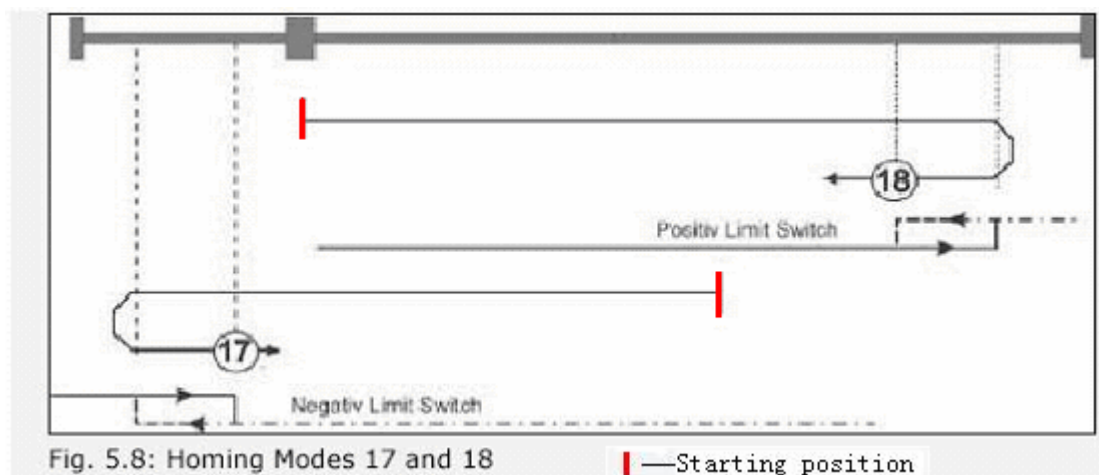
Fig. 5.8: Homing Modes 17 and 18

Figure 8.9   Homing Modes 17 and 18
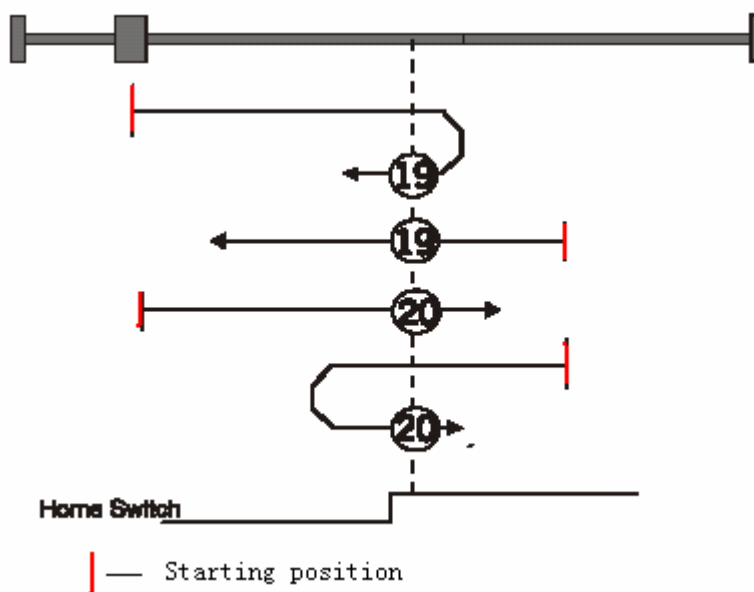
Figure 8.10   Homing Modes 19 and 20

Modes with Index Pulse	Modes without Index Pulse
Modes 1 and 2	Modes 17 and 18
Modes 3 and 4	Modes 19 and 20
Modes 5 and 6	Modes 21 and 22
Modes 7 and 10	Modes 23 and 26
Modes 11 and 14	Modes 27 and 30

Advantage: The reference point refers to the mechanical position and therefore independent on the motor.

Disadvantage: The mechanic switch is easily to deform and results in rising or falling edge position change. The mechanic switches must be reliable in this mode.

8.3.9 Homing Modes 32 and 33: Homing on Index Pulse

From the actual position, the motor continues to move to the adjacent index pulse signal along the negative or positive direction and defines the index pulse signal as the reference point.

Abb. 5.9: Homing Modes 32 and 33   ▌—Starting position
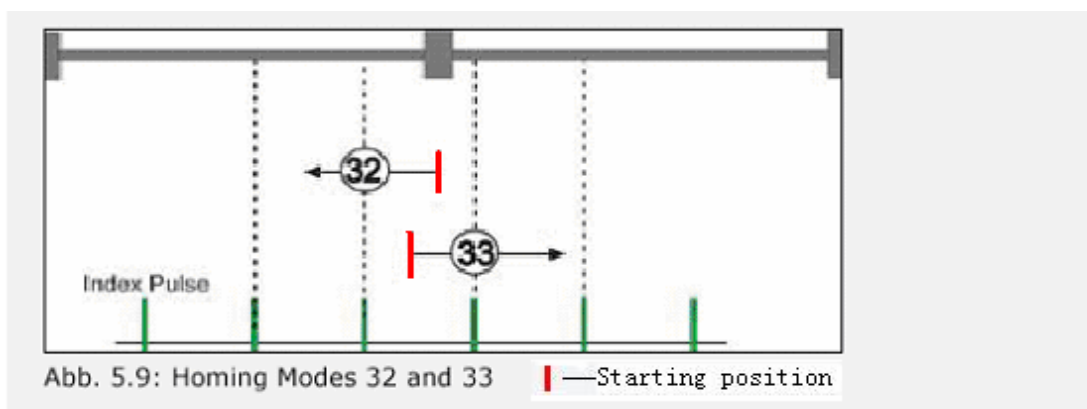
Figure 8.11   Homing Modes 32 and 33

8.3.10 Homing Mode 34: Homing on Actual Position

This homing mode defines the actual position as the reference point.

8.3.11 Homing Modes -17 and -18: Homing on End Stop

These two modes depend on mechanic end stop and define such position as the reference point.
Mode -17: The motor moves in the negative direction until end stop. The mechanic end stop is defined as the reference point.
Mode -18: The motor moves in the positive direction until end stop. The mechanic end stop is defined as the reference point.

Fig. 5.10: Homing Modes -17 and -18   ▌—Starting position
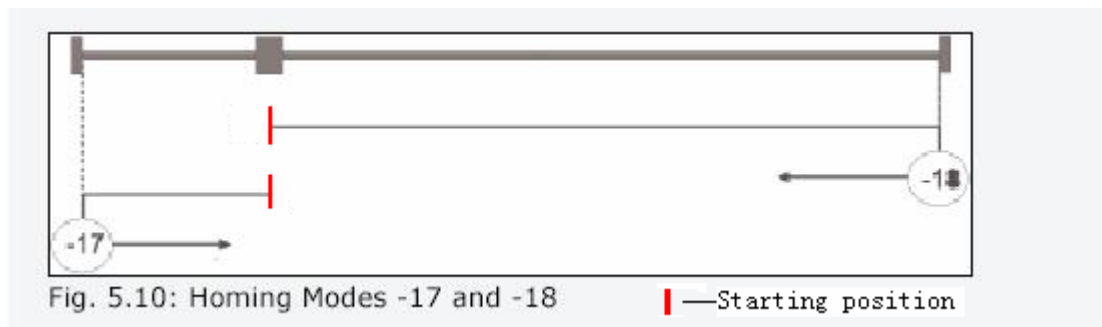
Figure 8.12   Homing Modes -17 and -18

Advantage: Reduce switches and sensors wiring; the mechanic end position is a reproducible reference point.
Disadvantage: The mechanic position may change, with low accuracy.

8.3.12 Homing Modes -1 and -2: Homing on Limit Stop with Index Pulse

Mode -1: The motor moves along the negative direction until the limit stop. Then, it turns into the positive direction until the index pulse is searched, and its position is defined as the reference point.
Mode -2: The motor moves along the positive direction until the limit stop. Then, it turns into the negative direction until the index pulse is searched, and its position is defined as the reference point.

Fig. 5.11: Homing Modes -1 and -2   ▌—Starting position
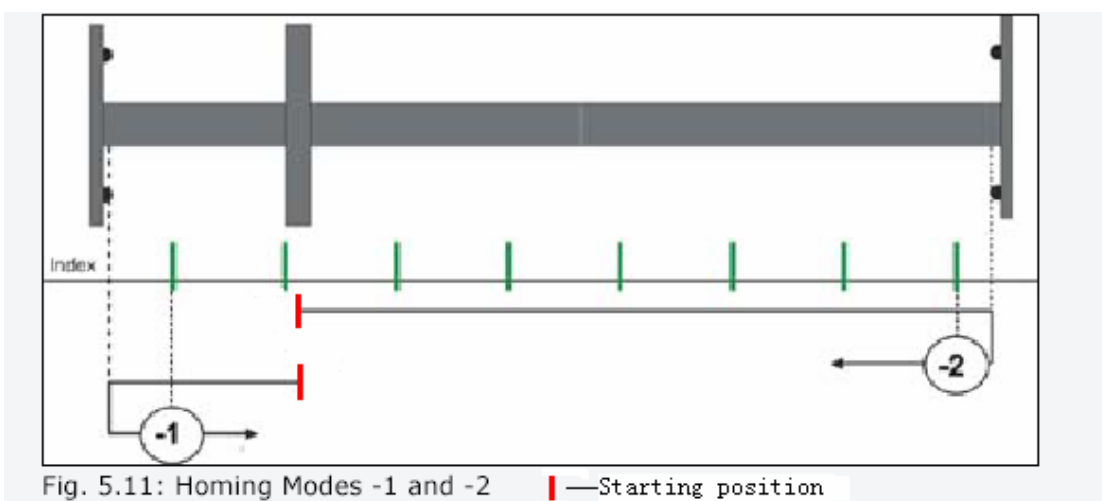
Figure 8.13   Homing Modes -1 and -2

Advantage: No external switches, the reference point is coupled to the mechanic position; the end positions can be protected by buffers; high reproducibility can be achieved by the index pulse.
Disadvantage: If the motor is changed or the coupling is loosened, the reference point must be re-calibrated.

8.3.13 Found Reference

Click the **Homing** tab in the **Control Axis** window, and then click the **<Start homing>** button. The second LED below the button turns green and the text "found reference" is displayed, indicating the reference point is found, as shown in the figure below:
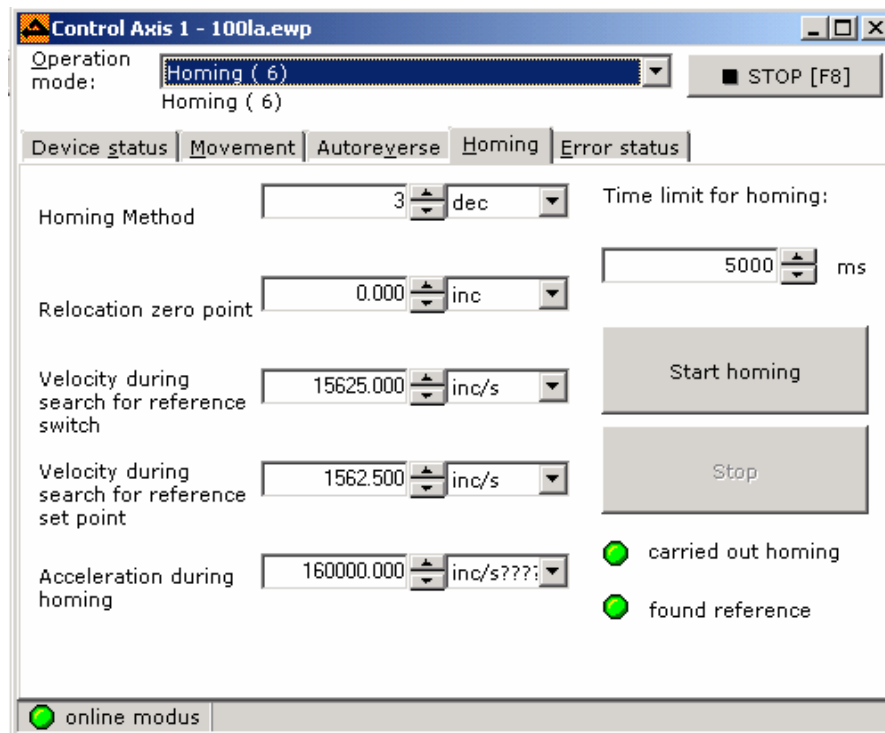
Figure 8.14   Reference Settings

If homing is not successful, check and correct the following values:
- Time limit for homing
- Velocity for search for homing switch
- Velocity for search for reference point (index pulse)
- Acceleration for homing

If the motor does not move, check:
- whether the motor can run freely with the mechanical part after power-off.

Chapter 9 Sequencer Programming

Sequencer programming means to assign values to objects that are executed similarly to a batch program. Objects can be similar to memory addresses; some can be set by the user, for example, velocity or position values; others can only be set by the servo controller, for example, status word or error messages.

The drive provides up to 256 sequences for the user. All programs are saved to the memory section which is secure against power failure. Each program sequence has a corresponding program number, which contains objects in the eight object lists with RW property. After a program sequence is called, all instructions in the program sequence will be executed once in an descending order. An online command or another program sequence will not interrupt the execution of the instructions in the program sequence. A program sequence can be executed by a control event. Upon occurrence of each event, a sequence of designated program can be executed. If an event occurs, the designated program sequence will be executed. Some events may occur at the same time, corresponding programs will be executed according to the sequence of triggered events. The ED servo drive provides such typical control events as upper/lower falling edge of input interface, timer time determined by internal program arrived, comparison result of two values as true, positioning end (reached), etc. Some events may be defined repeatedly. Therefore, a program sequence may be executed once or repeatedly. Another sequence of program may be called in one program sequence.

The sequencer programming window is shown as follows:
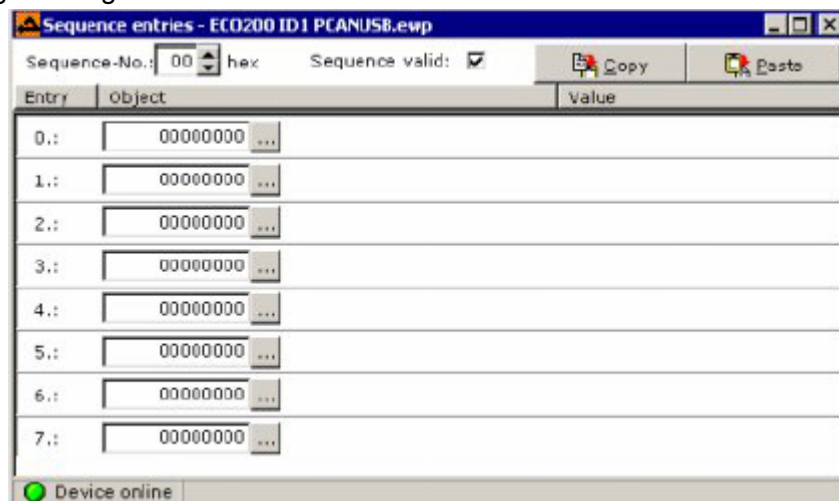
Figure 9.1 Input Program
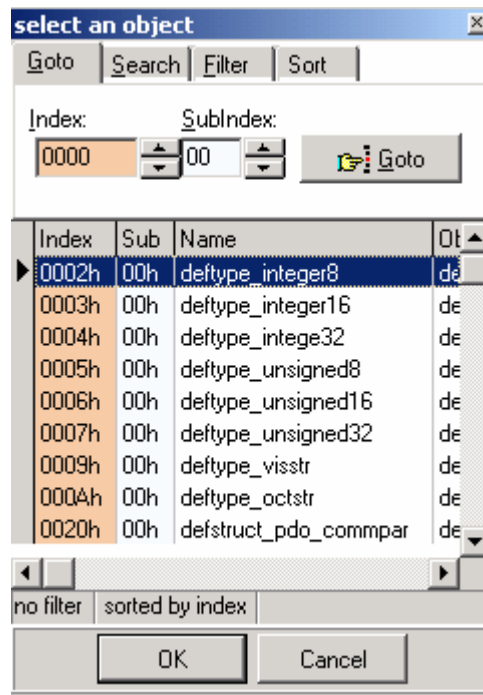
Figure 9.2 Selecting an Object

Eight objects may be assigned to each sequence.
1. Start with row "Entry0".
2. Click "…" to open the **Select an object** window.
3. Select an object and enter it to the sequence.
4. Click <**OK**>.

The selected object and the object comment are entered in the sequence, and then:
1. Assign a value to the object.
2. Click the box on the right of Sequence valid, and a "√" mark appears, indicating the program sequence is valid.
3. Enter conditions on which the sequence will be executed. For example, edge-triggered program sequence described in Section 5.5.2.1. Set an input to call the program sequence.

Example: Figure 9.3 shows two successive sequences.

The function of the "00" sequence is as follows:
● Entry 0: Set homing mode -1
● Entry 1: Set the search velocity for the homing signal as 100 mm/s.
● Entry 2: Sets acceleration and deceleration of homing.
● Entry 3: Change the operation mode to Homing.
● Entry 4: Activates the motor and starts homing.
● Entry 5: Switch to sequence 0x10 if reference point is found.
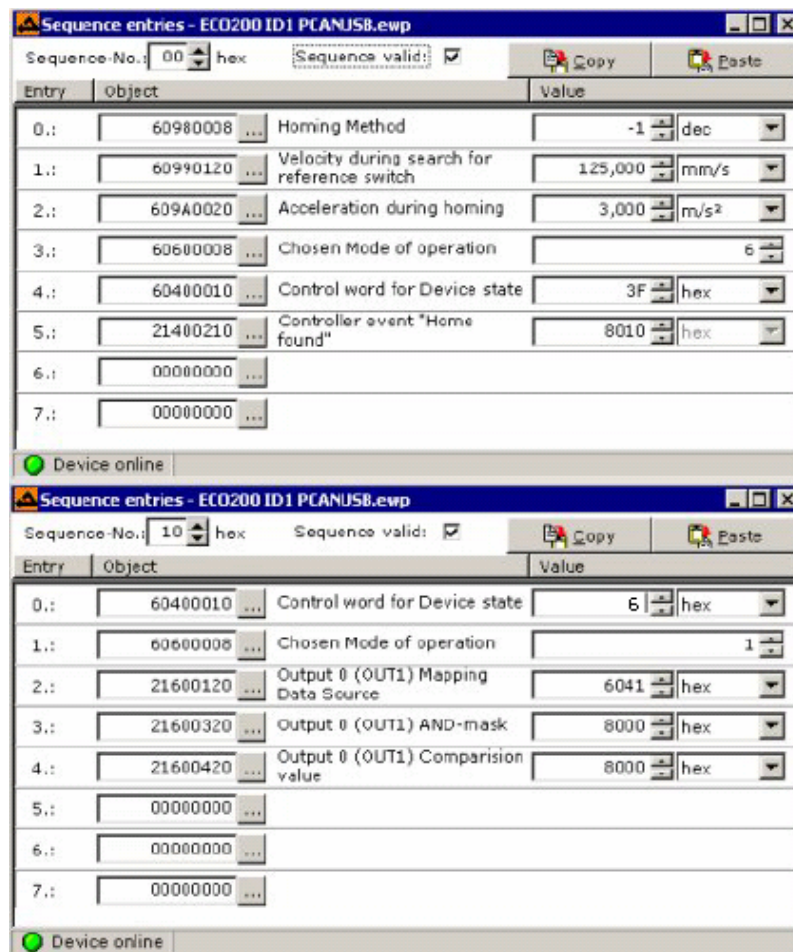
Figure 9.3   Two Successive Sequences

The function of sequence 0x10 is as follows:

- Entry 0: Loosen the motor shaft, and the motor axis is freely moveable
- Entry 1: Switch the operation mode to positioning mode 1
- Entry 2: Map the data source of the output 1 as the object 604100
- Entry 3: Set the **AND-mask** of output 1 to 8000
- Entry 4: The comparison value of output 1 is 8000
- Entry 5: The output interface 1 is of high level if the reference point is found

In the above example, initialize specific parameters in the homing mode. The output interface 1 outputs a high level value (+24V) upon "Reference found", and the operation mode switches to the positioning mode.

Chapter 10 Controller Performance Adjustment

The following section gives a brief introduction to the control principle of the ED drive, which helps users gain a in-depth understanding of controller features so that the system can achieve the maximal performance.
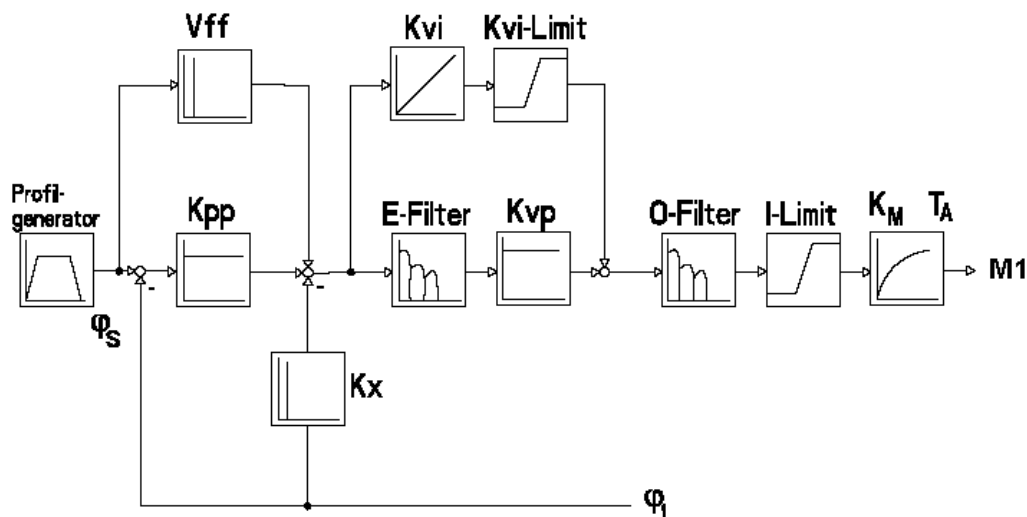
Figure 10.1   Controller Principle

Like all other servo systems, the ED control consists of position controller, velocity controller, and current controller. The current controller is in the innermost layer, the position controller is in the outermost layer, and the velocity controller is in the middle layer. The following section gives a brief introduction to some important parameters shown in the above figure.

> Vff—Velocity feedforward
> Kpp—Position Proportional Gain
> Kx—Velocity Feedback Coefficient
> E_Filter—Velocity error (Error, actual setting and feedback) filter
> Kvp—Velocity Proportional Gain
> Kvi—Velocity Integral Gain
> Kvi_Limit—Velocity Integral Limit
> O_Filter—Velocity Output Filter
> I_Limit—Current Limit
> KM—Torque Constant
> TA—Coil Time Constant (=L/R)

The following "Recipe" table is an important reference. When the motor underperforms, adjust these important parameters.

Encoder resolution 8000 inc/U:

C [Nm/rad]	JL / JM	Kvp	E-Filter	O-Filter	Kpp
>= 250	1	60	1	1	4000
	5	100	1	1	4000
	10	250	1	2	4000
	15	250	1	4	4000
	30	600	2	10	3500
	100	600	4	20	2000
80	5	60	1	3	3000
	10	150	1	10	3000
	15	200	1	10	3000
	30	200	1	20	2500
	100	350	3	40	2000

Table 10.1

Encoder resolution 16000 inc/U:

C [Nm/rad]	JL / JM	Kvp	E-Filter	O-Filter	Kpp
>= 250	1	5	1	1	5000
	5	5	1	1	5000
	10	6	1	1	5000
	15	8	1	2	5000
	30	20	1	5	4500
	100	35	1	7	4500
80	5	5	1	1	4000
	10	6	1	1	4000
	30	10	1	2	4000
	100	20	1	7	4000

Table 10.2

Chapter 11 Communication Protocols

The ED supports powerful communication capabilities and adopts the control mode based on an object dictionary. All controls come down to the configuration of internal objects. The configuration can be implemented by multiple means including RS-232, RS-485, CANopen, and Profibus. It supports the connection of multiple sites and simultaneous operation of multiple communication ports. This chapter presents a description of communication ports and communication protocols.

11.1 RS-232C Interface

11.1.1 Physical Interface

11.1.1.1 Single-Axis Application

The ED requires a three wire RS-232C to connect to a host computer. Connect the wiring terminal X5 directly to the COM port of a PC as follows:

```
PC COM1/IRQ4, COM2/IRQ3          ED X5
              RxD 2   ----------------------   2 TX
              TxD 3   ----------------------   3 RX
              GND 5   ----------------------    5 GND
```

11.1.1.1 Multi-Axis Application

The host (for example, a PC, PLC, etc.) serves as a master and the ED serves as a slave. The wiring is shown as follows:
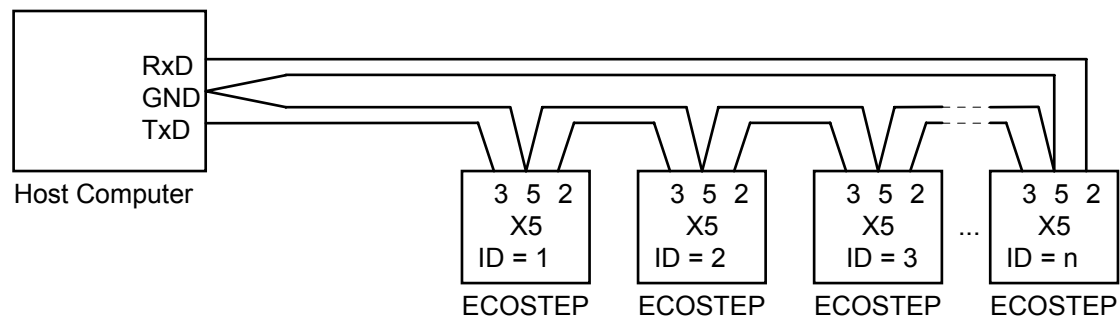
Figure 11.1   RS-232 Multi-Axis Application

ID numbers (address number) are selectable by DIP switches S0/S1/S2/S3. Every ED needs an unique ID No. ID 0 is reserved by the system.

11.1.2 Transport Protocol

The RS-232C communication of the ED strictly follows a master/slave protocol. The host computer can send data to any ED device connected to the TXD line of the host. Upon receiving such data, the device will immediately transfer the data to the next device in the transmission loop. The last device in the loop will send the data to the host, ensuring that each ED drive in the loop can receive data packets. Each ED can process

data and judge whether such data is deliver to it (whether the ID byte in the instruction is the same as its own ID No.). If yes, it will process the data; otherwise, it will not process such data.

The ED requires the following settings:

9600 baud rate, 8 bits, 1 stop bit, no parity check

This transport protocol uses a data packet with fixed length of10 bytes.
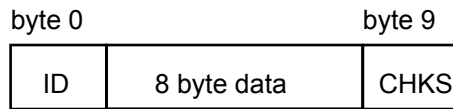
byte 0   byte 9

ID	8 byte data	CHKS

ID is the ID No. of the slave

CHKS =-SUM(byte0,…,byte8), CHKS is the last two digits of the calculation result.

The host sends:

byte 0   byte 9

ID	8 byte host data	CHKS

The host receives:

byte 0   byte 9   byte 10   byte 19

ID	8 byte host data	CHKS		ID	8 byte slave data	CHKS
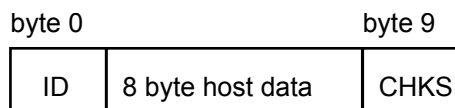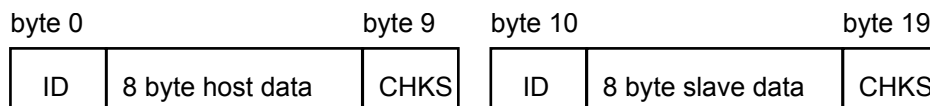
Note: Each 10-byte packet has its own CHKS.

If the host sends a data packet with an unused ID to the ED, no slave will respond to the packet. Then, the host will only receive the 10-byte data packet. After the host sends the data correctly, the slave will find the data packets in compliance with its own ID and check the CHKS value. If the checksum does not match, the slave will not make a response.

At present, the hardware supports only 15 slaves. A system with up to 127 slaves is under development.

11.2 RS-485 Interface

11.2.1 Physical Interface

For multi-axis application, the ED supports 2-wire (half duplex) and 4-wire (full duplex) RS-485 communication. The ED supports a baud rate of 250 Kbps at the maximum and provides IEC1000-4-4 compliant ESD protection. The wiring diagram is shown as follows:
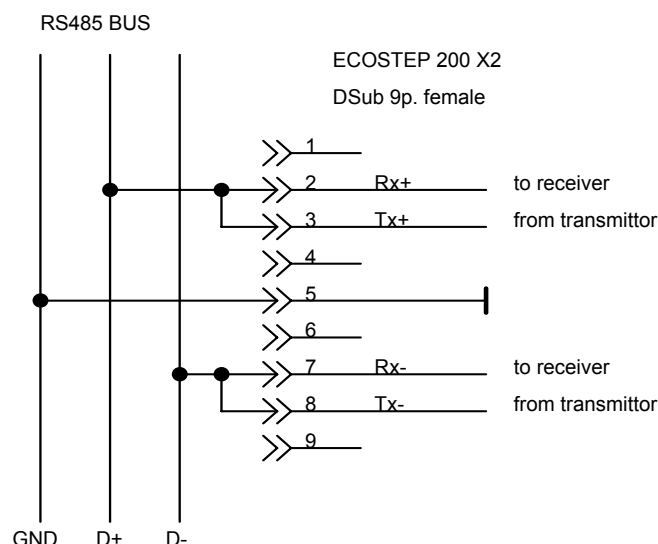
Figure 11.2   RS-485 Connection

As shown in the above figure, multiple drives can be connected to the RS-485 bus. The RS-485 communication protocol supports multi-axis applications.

ID numbers (address number) are selectable by DIP switches S0/S1/S2/S3. Each ED has an unique ID No. ID 0 is reserved by the system and cannot be set by the drive.

11.2.2 Transport Protocol

The RS-485 communication of the ED strictly follows a master/slave protocol. The host computer can send any data to each ED device connected to the bus. The drive configured with ID No. will calculate such data and return a reply. Only one device (host or slave) can send data over the network at the same time.

Default communication settings for the ED are as follows:

Baud rate (default) = 9600 Bps

Data bits = 8

Stop bits = 1

No parity check.

The baud rate of the RS485 communication protocol is adjustable. The object for RS-485 baud rate adjustment is index 2F9100. After changing the value, it is necessary to save it and reboot the system so that it will take effect.

The value is calculated according to the following formula:

N = (614400/baud rate) - 1

9600 bps   N=63 bit timed error 0%;

19200 bps N=31 bit timed error 0%;

38400 bps N=15 bit timed error 0%;

Higher baud rate (>38400 bps) will lead to timed errors. 57600 bps N=11 bit timed error 3.125%.

The transport protocol used by the RS-485 uses a data packet with fixed length of10 bytes.

byte 0   byte 9

ID	8 byte data	CHKS

ID is the ID No. of the slave

CHKS =-SUM(byte0,…,byte8), CHKS is the last two digits of the calculation result.

The host sends:

byte 0   byte 9

ID	8 byte host data	CHKS

The slave sends/The host receives:

byte 0   byte 9

ID	8 byte slave data	CHKS

Note: Each 10-byte packet has its own CHKS.

If the host sends an ID not existed in the network to the ED, no ED will make a reply. After the host sends the data correctly, the slave will find the data packets in compliance with its own ID and check the CHKS value. If the checksum does not match, the slave will not make a response.

At present, the hardware supports only 15 slaves. A system with up to 127 slaves is under development.

11.3 Data Protocol

A data protocol is different from a transport protocol. It contains 8 bytes of all 10 bytes of the above RS-232 and RS-485 protocols. Definition of ED internal data complies with the CANopen international standard. All parameters, values and functions are expressed by index and subindex. These parameters can be set either by RS-232C, RS-485 or by CAN.

11.3.1 Download (From Host to Slave)

Download refers to that the host sends a command to write values into the objects in the slave, and the host generates an error message when the value is downloaded to a non-existent object.

The host sends:

byte 0   byte 1   byte 2   byte 3   byte 4   byte 5   byte 6   byte 7

CMD	INDEX	SUB INDEX			DATA		

CMD	Specifies the direction of data transfer and the volume of data.
23(0x16)	Sends 4-byte data (bytes 4...7 contain 32 bits)
2b(0x16)	Sends 2-byte data (bytes 4, 5 contain 16 bits)
2f(0x16)	Sends 1-byte data (bytes 4 contains 8 bits)
INDEX	index in the object dictionary where data should be sent
SUB INDEX	Subindex in object dictionary where data should be sent

In all four bytes in data, the lower-order bits are arranged before the higher-order bits. To write 600 RPM into Target Velocity in the slave, 600rpm = 10r/s = 80000inc/s = 80000 * 64 = 5120000 inc/64s = 4E2000. Where, the internal unit of 60FF0020 is inc/64s, 5120000 is in decimal system, and 4E2000 is in hexadecimal system. Since the length of the object to be written is 4 bytes and the calculation result 4E 20 00 has only 3 bytes.

Zero shall be filled to the higher-order bits. Therefore, the final result = 00 4E 20 00.

DATA: byte4=00

          byte5=20

          byte6=4E

          byte7=00

The slave answers:

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

RES: Displays slave response:

60(0x16)   Data successfully sent

80(0x16)   Error, bytes 4…7 contain error cause

INDEX   16-bit value, same as that sent by the master

SUBINDEX   8-bit value, same as that sent by the master

RES   Reserved for future use

11.3.2 Upload (From Slave to Master)

Upload refers to that the master sends a command to read object address in the slave and the master will generate an error if a non-existent target address is uploaded.

The master sends:

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

CMD   Specifies the direction of data transfer

40(0x16)

INDEX   16-bit value

SUBINDEX   8-bit subindex

RESERVED   Bytes 4…7 not used

The slave receives:

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

RES   Displays slave response:

43(0x16)   bytes 4...7 contain 32-bit data

4B(0x16)   bytes 4, 5 contain 16-bit data

4F(0x16)   byte 4 contains 8-bit data

80(0x16)   error, bytes 4…7 contain error cause

INDEX   16-bit value, same as that sent by the master

SUBINDEX   8-bit value, same as that sent by the master

If the data contains no error, byte 4…byte 7 save the object value read from the slave, with the lower-order bits arranged before the higher-order bits. Correct value = byte7, byte6, byte5, byte4. If there is an error, data

contained in these four types is no longer object values read from the slave. For example:

The master sends a "upload" command to the slave:
01 40 ff 60 00 00 00 00 00 60   (This command reads target velocity 60FF0020 from the slave)

The slave answers:
01 43 FF 60 00 00 20 4E 00 EF

Indicates: 01—Slave index is 1
43 – Received data contains 4 bytes, saved to byte 4…byte 7 in the 10 bytes of the response
byte4=00, byte5=20, byte6=4E, byte7=00

Then, DATA= byte7 byte6 byte5 byte4 = 00 4E 20 00 = 5120000 = 80000 * 64
The internal unit of the target velocity 60FF0020 is inc/64s. After conversion, DATA = 80000 inc/s = 600 rpm.

Chapter 12 Fieldbus Interface of ED Drive

12.1 CANopen Bus Interface

The ED servo is typical CANopen Slave equipment with dynamic PDO configuration capability. PDO configurations are saved to the equipment so that it may be started for operation when the Master (NMT Master) is not available. Communication can be used to observe the configuration of the PDO by NMT Master or to configure the PDO manually.

For details, refer to Section 5.7, Communication Parameter Settings.

12.2 PROFIBUS Interface

12.2.1 Introduction to PROFIBUS

PROFIBUS is an international, open fieldbus standard independent of equipment manufacturers and is widely applied in such fields as manufacturing automation, process automation, buildings, transport, and electric power industry. It satisfies the European open fieldbus standard (EN50170) and consists of three parts: Profibus-FMS (Fieldbus Message Specification), Profibus- PA (Process Automation), and Profibus-DP (Distributive Peripheral).

Profibus- DP is most suitable for high-speed data transfer (at a rate of 9.6Kbit/s~12Mbit/s), implements economic equipment-level network, applicable to the communication between online controller and distributed intelligent I/O, and satisfies the quick response requirements of the AC/DC velocity regulation system.

Profibus- DP works in the mode of master/slave communication (communication between master and slave) and token communication (communication between masters). The master/slave communication works in strict accordance with the cycle mode. All watchdog timers will check whether the master and slave have any fault in a very short period of time, and the extended PROFIBUS-DP diagnosis function can locate the fault quickly, and each station can modify its station number conveniently.

For more information, please visit WWW.PROFIBUS.COM.

12.2.2 Installation Description

PROFIBUS-DP transfers data mainly in the RS-485 mode or by optical fiber technology. Basic features for RS-485 transmission technology are as follows:

    ——Network topology: linear bus, with active bus terminal resistors at both ends

    ——Transmission rate: 9.6K bit/s to 12M bit/s

    ——Transmission media: Shielded twisted pairs, or unshielded cables (depending on environment EMC)

    ——Number of nodes: Each segment supports 32 nodes, with 126 nodes at the maximum (including inter-segment repeater)

    ——Connector: Preferably use 9-pin D-Sub connector

Notes on installation:

    1. All devices must be connected to the bus.

    2. Each segment can contain a maximum of 32 nodes (master or slave).

3. A bus terminal resistor is located in the beginning and end of each segment to prevent misoperation. Two bus terminal resistors must always be powered. See figure 13.1.

4. If a segment contains more than 32 nodes, use a repeater to connect different bus segments. Number of repeaters in serial connection shall not exceed 3, as shown in Figure 13.2.

5. Maximum cable length depends on transmission rate. If a Type-A cable is used, the relationship between the transmission rate and the cable length is shown in the table below:

Baud rate (K bit/s)	9. 6	19. 2	93. 75	187. 5	500	1500	12000
Distance/Segment (M)	1200	1200	1200	1000	400	200	100

Table 12.1 Baud Rate and Transmission Distance

6. Parameters for Type-A cable:
Impedance: 135-165W　　Capacitance: < 30pf/m　Loop resistance: 110W
Wire gauge: 0.64mm^2　　Wire area: > 0.34mm^2

7. The 9-pin D-Sub connector is recommended for the PROFIBUS network using the RS-485 transmission technology. The definition and wiring of pins are shown in Figure 12.1.

8. When connecting different nodes, make sure the data cables shall be in good order. If the system is applied in an environment with high electromagnetic radiation (for example, automobile making industry), use shielded cables to enhance the EMC performance.

9. If braid shielded cable and shielding foil are used, the two ends should be connected to the protection ground (PGND). Use shielded cables with as much area as possible for coverage to ensure sound conductivity. In addition, data cables shall be isolated from high-voltage cables.

10. If the data rate exceeds 500 Kbit/s, avoid using stub sections. Use connectors currently available on the market to connect the data I/O cables directly with the connector. The bus connector can be connected or disconnected at any time, without interrupting data communication of any other nodes.
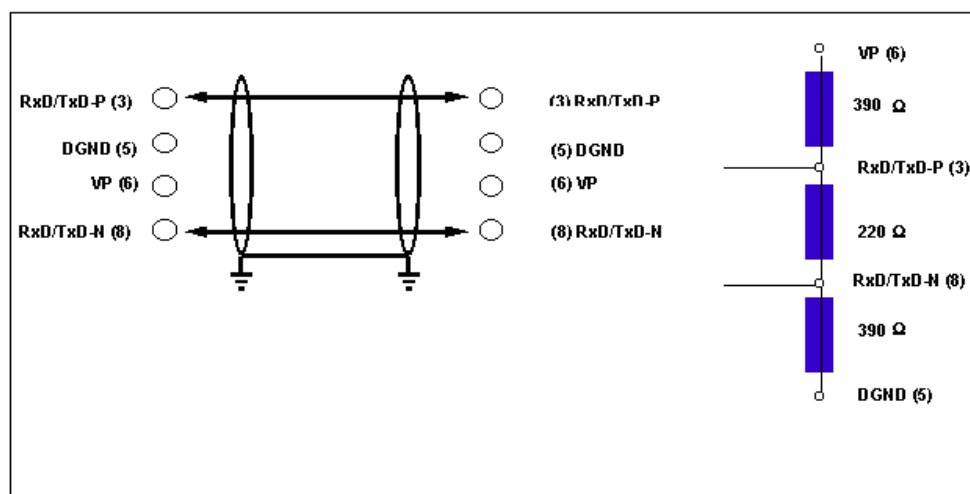
Figure 12.1 Wiring and Bus Terminal Resistors of PROFIBUD-DP and PROFIBUS-FMS
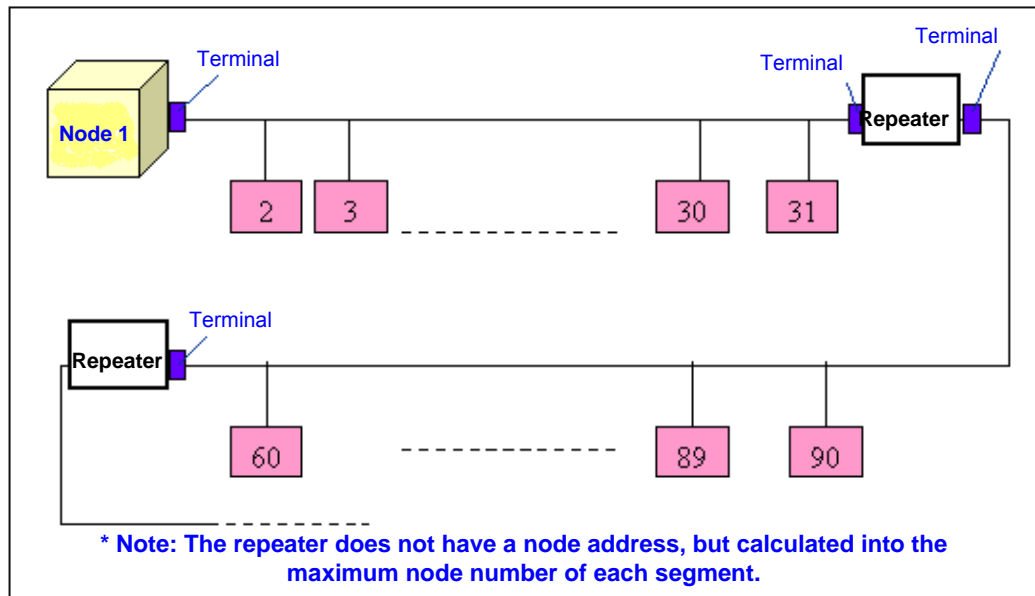
12.2. A Maximum of 32 Nodes (Master or Slave) in a Segment

12.2.3 Connection with ED

As shown in the figure below, connect the PROFIBUS to the PROFIBUS interface of the ED drive. Pins are defined as follows:
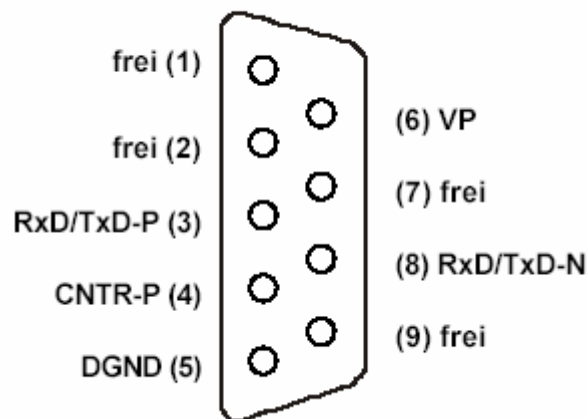
Figure 12.3 Pin Definition of PROFIBUS Interface of ED Drive

According to the EN50170 standard, it includes the repeater auxiliary control signal CNTR-P which supports PIN4.Note that the interface is powered by the ED drive, without the need of external power supply.

The ED drive supports the following baud rates: 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, and 12Mbit/s. The interface is adaptive to baud rate, without the need of configuration.

Node IDs of the ED drive are settable by DIP switches S0-S3, 01-15. Offset of the node ID is set by the internal parameter register (2F800008). Note that the node ID 00 is not allowed to use.

12.2.4 ED Drive State Machine

As a standard PROFIBUS-DP Slave, the ED drive has a state machine in compliance with EN50170, as shown below:
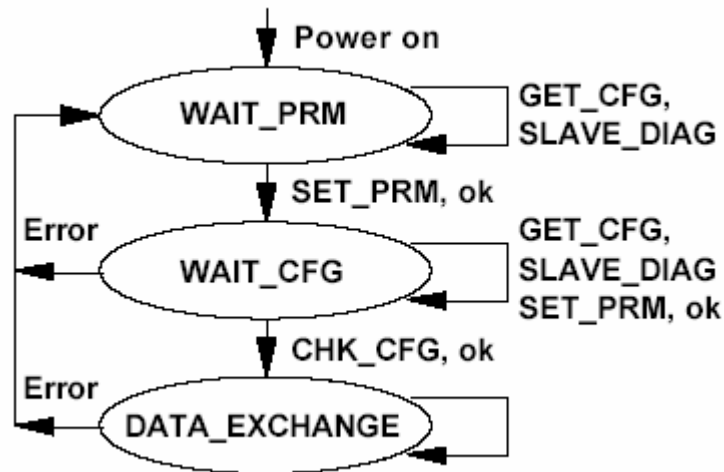
Figure 12.4 State Machine Mechanism of the ED Drive

After the logic voltage of the ED drive is turned on, the ED drive enters the "WAIT_PRM" to wait for parameter initialization. At this time, the slave waits to receive parameter packets from the master and will not conduct data exchange;

If the slave receives parameter packets, it will enter the state "WAIT_CFG" of waiting for configuration. Then, the master needs to send the configuration packets of the slave (specifying the I/O bytes of the slave). The slave will compare the received packets with stored configuration. If they are consistent, it will enter data exchange state "DATA_EXCHANGE"; otherwise, it will not perform data exchange.

During the "DATA_EXCHANGE" period, cyclic data exchange is carried out between the master and the slave.

12.2.5 Setting PROFIBUS Communication Parameters of the ED Drive with Siemens STEP7

12.2.5.1 Installing GSD File

Install the file in the HW CONFIG interface of the STEP7 software, as shown in the figure below. Upon completion of installation, the JAT\ECOSTEP\ECOSTEP200 appears in the PROFIBUS DP\Additional Field Devices\Drivers directory in the CATALOG. Then, you can add it to the DP master system created by the user.
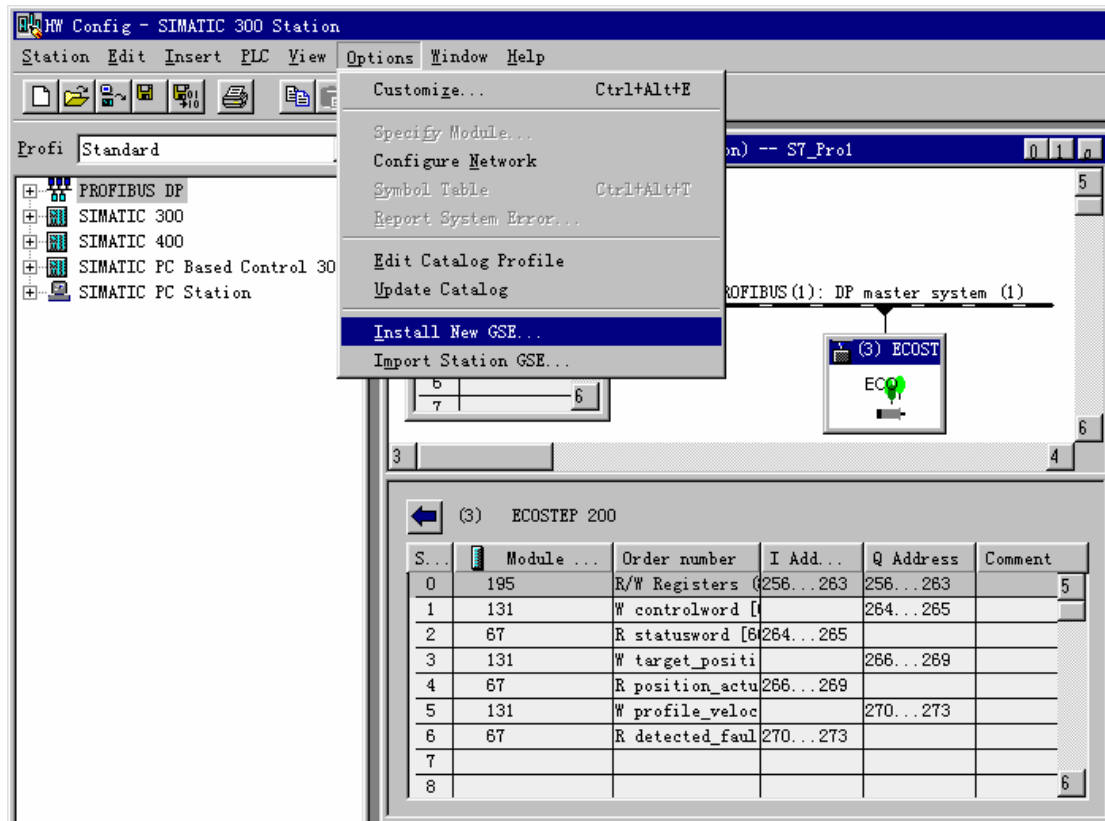
12.2.5.1 Installing GSD File

The GSD files of the ED drive provide settings for most common operation control states (homing, positioning, etc.) in the form of modules. It further provides register modules for random reading and writing.

If you plan to configure an object not defined in the GSD, you can implement this function conveniently by modifying related description in the GSD file. For example:

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

EndModule

Make a copy and modify related parameters, for example, adding control mode:

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

EndModule

After you reinstall the GSD file, a new module configuration option is added.

12.2.5.2 Configuration of GSD File Module

Configuration format of the module is as follows:

Command String	Length	INDEX(HIGH)	INDEX(LOW)	SUBINDEX

1) Cyclic reading command

Command string: 0x43

Data length (data following the SUBINDEX) description is classified into byte and word. Since the data length of all ED parameters include 1Byte, 2Byte, and 4Byte, IDs used to describe data length include 0x80 (1BYTE), 0x81 (2BYTE), 0x83 (4BYTE), 0xC0 (1WORD), and 0xC1 (2WORD). During transmission, the higher-order

bytes will be sent first and then the lower-order bytes. Data in the sending buffer will be refreshed continuously (usually in an interval of several ms).

For example, the configuration date 0x43, 0xC1, 0x60, 0x63, and 0x00 indicate that the value (2WORD) in the parameter index 0x606300 (Position_actual_value) will be placed in the sending area (immediately following the byte 0x00).

2) Cyclic writing command

Command string: 0x83

IDs for data length include 0x80 (1BYTE), 0x81 (2BYTE), 0x83 (4BYTE), 0xC0 (1WORD), and 0xC1 (2WORD).

3) Random data reading/writing command

Command string: 0xC3

The configuration format is as follows:

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

Immediately after the note, there will be two groups of 8-byte input/output data. The format of these two data groups is exactly the same, as shown below:

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

Where, Command is of vital importance. Its format is shown below:

SCS/CCS	Toggle	Len.	Exp.	Lsp.

SCS/BIT5..7: Control command to the Server

Server Command Specify (2=reading request, 1=writing request)

CCS/BIT5..7: control command to the Client

Client Command Specify (2=reading reply, 1=writing reply, 4=error information)

Tgl/Bit4: Toggle bit, copied to the response packet from the request packet

Len/Bit2~3: Bytes not used in the data buffer (0~3)

Exp/BIT1: Express/fragment transmission

Lsp/Bit0: Length description

Where, BIT0~4 will be ignored in the request packet, but will work in the response packet.

If a request is not properly responded, CCS equals to 4 (BIT7 set to 1). Since the ED serves as a CANOPEN device, a fault code will be placed in the data area according to the CANOPEN DS301 standard.

For example:

A) The master reads the address 0x100000 (equipment specification description) from the slave, and the slave answers: 0x00000192 (402 in decimal system, CANOPEN DS402 standard).

   Request packet: 0x10, 0x00, 0x00, 0x40, 0x00, 0x00, 0x00, 0x00

   Reply packet: 0x10,0x00,0x00,0x43,0x00,0x00,0x01,0x92

B) The master sets the maximum current 1023 (HEX 03FF) in the slave address 0x607300

   Request packet: 0x60, 0x73, 0x00, 0x20, 0x00, 0x00, 0x03, 0xFF

   Reply packet: 0x60, 0x73, 0x00, 0x60, 0x00, 0x00, 0x03, 0xFF

C) The master sets the maximum current 3071 (0x0BFF) in the slave address 0x607000. The value is greater than 2047, so an error occurs, and the packet with type 6, code 9 and error code 31 is returned.
    Request packet: 0x60, 0x73, 0x00, 0x20, 0x00, 0x00, 0x0B, 0xFF
    Reply packet: 0x60, 0x73, 0x00, 0x80, 0x06, 0x09, 0x00, 0x31

D) The master calls the sequence SEQ02 from the slave
    Request packet: 0x21, 0x18, 0x00, 0x20, 0x00, 0x00, 0x00, 0x02
    Reply packet: 0x21, 0x18, 0x00, 0x6C, 0x00, 0x00, 0x00, 0x02

The user can also configure Universal Module in the HD, and then set input/output structure in the **PROPERTY** menu in DP-SLAVE. For example, to set Profile_acceleration, corresponding to Index 0x6083, SUB00, type UNSIGNED32, OUTPUT, as shown in the figure below:
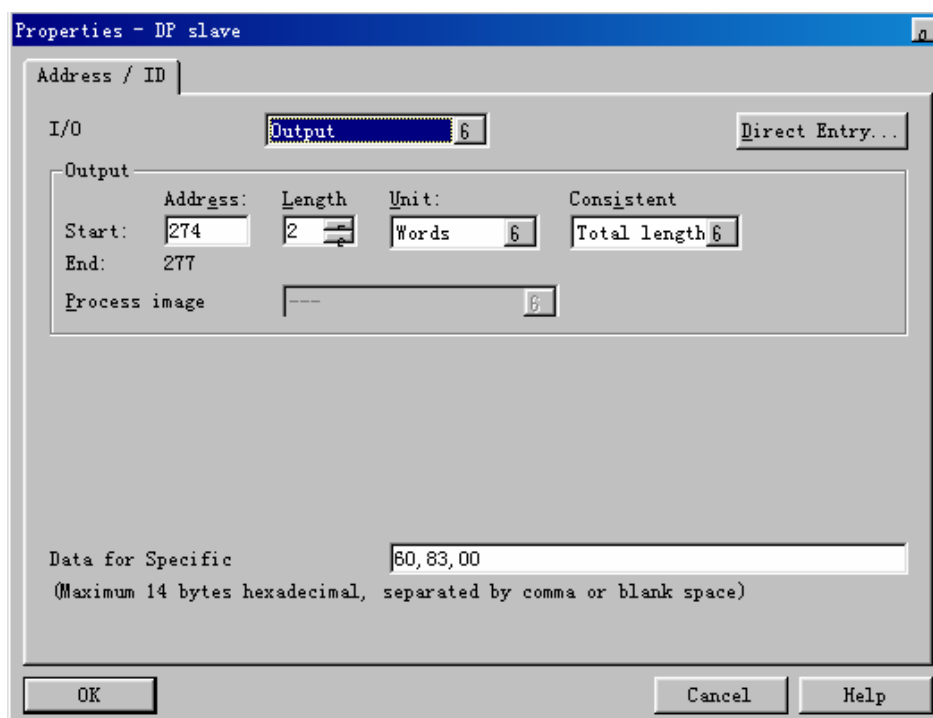
Figure 12.6

In the above figure, we can select OUTPUT in the I/O type box. Then, we can select addresses according to the insertion positions of modules. Select 2 WORDs for data type. To ensure consistent data transmission, select TOTAL LENGTH. In the lower of the window, enter the index description of the parameter: INDEX 6083, SUB 00.

The user can select some address registers from PLC resources to save configured parameters and describe corresponding configuration parameters by the symbol table. Make sure to process the configured length of data at a time. Pay special attention to reading and writing of random 8-byte data, because the STEP7 can process only 4 bytes in general. Call the SFC14 (DPRD_DAT) and SFC15 (DPWR_DAT) to implement this function.

The configuration window is shown below:
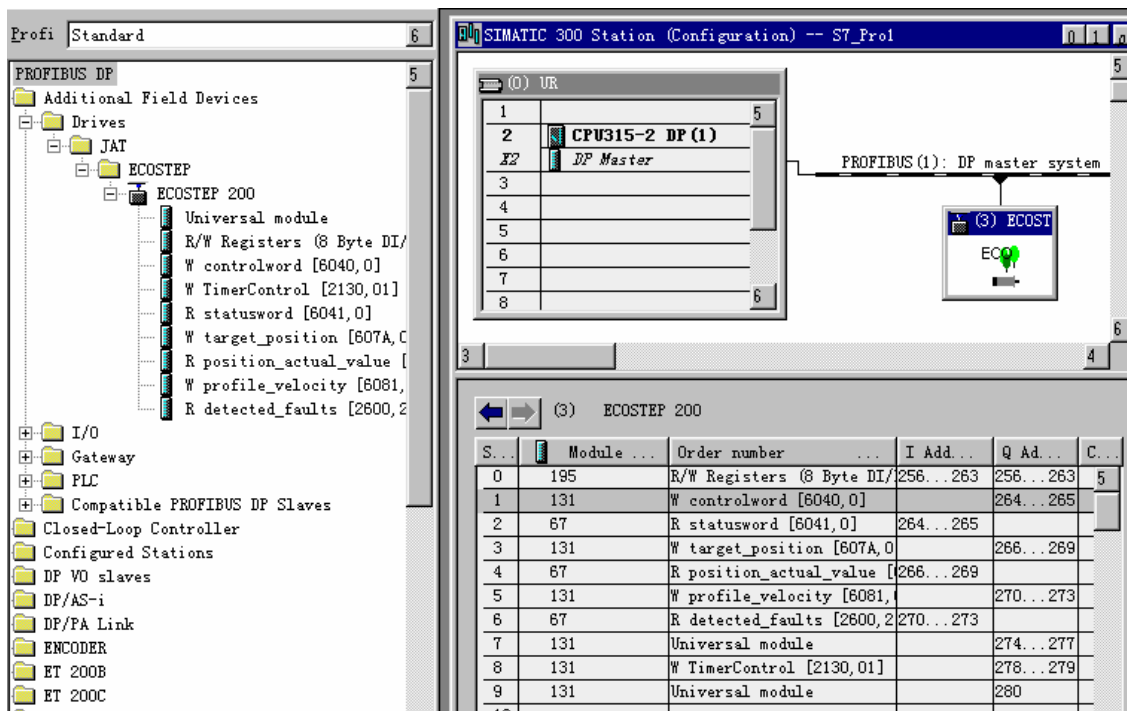
Figure 12.7 STEP7 GSD Configuration Window

12.2.5.3 Diagnosis and Auxiliary Information

The first 6-byte (BYTE0~5) diagnosis information of the ED drive is consistent to the PROFIBUS standard. Where, Byte6 (describing external diagnosis length, including this byte) will describe whether there are four additional bytes to provide further diagnosis data for the ED. Meanings of these four additional bytes are as follows:

ERROR BIT (fault bit)	ERROR EVENT (fault event)
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,BIT0	RESERVED(NOT USED)

Table 12.2   Fault Information

The above fault information table is also described in the GSD file, so it can generate common text information.

If a fault enable bit in the drive is set, the red LED corresponding to the external diagnosis bit EXT_DIAG will blink immediately. BIT3 of the first byte BYTE0 in the diagnosis packet corresponding to the EXT_DIAG is set. New diagnosis information will be sent only after the fault flag changes.

For example, the motor encoder is not installed, which will lead to the setting of FAULT_ENC_ERROR_BIT(BYTE7,BIT2) and EXT_DIAG(BYTE0,BIT3). The diagnosis packet is as follows:

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

EXT_DIAG Set   FAULT_ENC_ERROR_BIT Set

The following information is shown in CPU fault diagnosis:
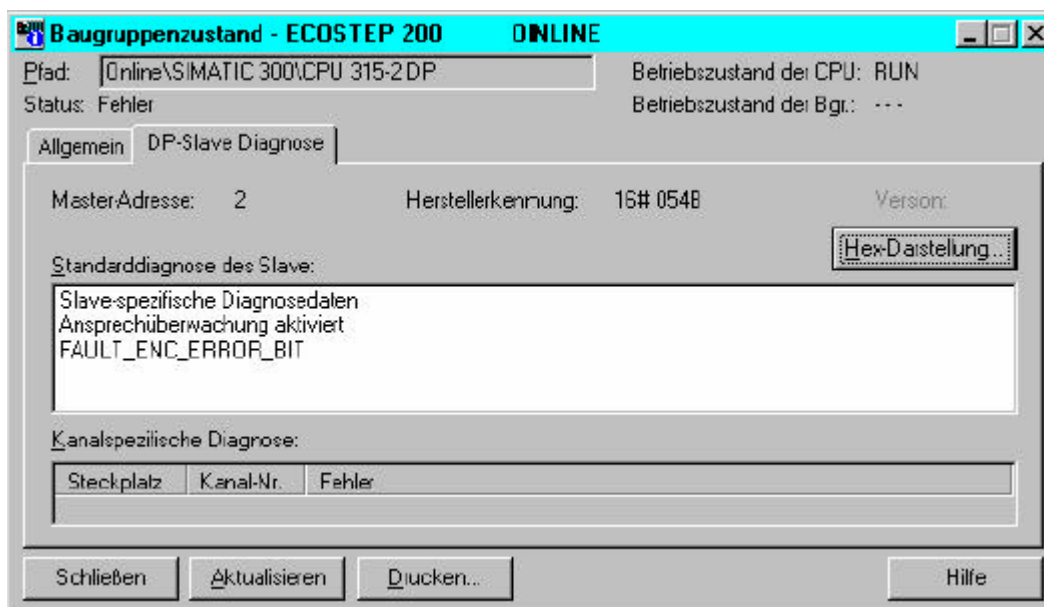
Figure 12.8

12.2.5.4 Operation Service

The ED servo supports SYNC and UNSYNC global commands, which are used to synchronize data transmission and clear data according to the standard.

If the ED servo receives a SYNC command, it will lock the output data in the current status and will store all output data from the PROFIBUS in the subsequent data transmission cycles, with the output status remained unchanged until the ED servo receives the next SYNC command. Only until then will the stored output data be transferred to an external device. The UNSYNC command is used by the ED servo to exit the SYNC mode.

Chapter 13 List of Common Objects

Based on the data communication protocols described in Chapter 11, the user can design programs with JAVA, Visual C++ or Delphi software to implement different control modes.   Understanding of the data in the following tables will be helpful for programmers and users. All parameter values are transferred in hexadecimal data. In the later sections of this document, we adopt the hexadecimal system and use Index (16-bit index) and Subindex (8-bit subindex) to represent the register addressing. The digit 08 indicates the register will store data up to 1 byte, and the digit 10 indicates that the register will store data up to 2 bytes, and the digit 20 indicates the register will store data up to 4 bytes. It also covers the storage digits and read/write property of the register, read or write flag (RW), read-only or write-only flag (RO, WO), and mapping flag (M).

Modes and Control: 0x6040 ff

Index	Sub-index	Bits	Command Type	Unit	Description
6040	00	10	RW, M	bitcode	Use control word to change status of drive => machine state 0x06 power off 0x0F power on 0x0B Quick stop, load stops-voltage switched off 0x3F immediate absolute positioning 0x5F relative positioning 0x1F Homing (0x0F first) 0X80 clear internal faults
6041	00	10	RO, M	bitcode	Status byte shows the status of drive 0X0001 ready to be powered on 0X0008 error detected 0X0400 target reached 0X4000 controller powered on 0X8000 reference found
6060	00	08	WO, M	number	Operation modes: 1 Positioning with position loop 3 Velocity with position loop -3 Velocity loop (immediate velocity mode) -4 Master/slave or pulse/direction control mode 6 Homing 7. CANOPEN based motion interpolation

Measurement data: 0x6063 ff

Index	Sub-index	Bits	Command Type	Unit	Description
6063	00	20	RO, M	inc	Actual position value
6069	00	20	RO, M	inc/64s	Actual velocity value
606B	00	20	RO, M	inc/64s	Velocity demand value
606C	00	20	RO, M	inc/64s	Actual Velocity value after filtering
6078	00	10	RO	integer	Actual current value
60FD	00	20	RO	bitcode	Status words for 10 digital inputs (DIN1-8, RESET, External ENABLE): 0x0201 0000 DIN1 is "High level" 0x0280 0000 DIN8 (reference point) is "High level" 0x02C8 0000 DIN8, 7 and 4 are "High level"

Target object: 0x607A ff

Index	Sub-index	Bits	Command Type	Unit	Description
607A	00	20	RW, M	inc	Target position in operation mode 1, shift to demand position if control word starts motion
60FC	00	20	RO, M	inc	Demand position in operation mode 1
6081	00	20	RW, M	inc/64s	Maximum velocity of trapezium profile in mode 1
6083	00	20	RW, M	$16inc/s^2$	Acceleration of the trapezium profile $1000rad/s^2$ is roughly 80,000 inc/s^2
6084	00	20	RW, M	$16inc/s^2$	Deceleration of trapezium profile
60FF	00	20	RW, M	inc/64s	Target velocity in mode 3, -3, or 4
6073	00	10	RW	integer	Maximum current: See glossary-->IDAC
607F	00	20	RW, M	inc/64s	Maximum possible velocity in mode 1 or 3 For example: resolution 8000 inc 1000 RPM = 8,533,333 inc/64s

Performance object: 0x6065 ff

Index	Sub-index	Bits	Command Type	Unit	Description
6065	00	20	RW, M	inc	Maximum following error at which the drive generates an alarm 2000 default value 60 well tuned drive
6067	00	20	RW, M	inc	position window for "target reached flag"—default value is 10
607D	01	20	RW, M	inc	Minimum software end position
607D	02	20	RW, M	inc	Maximum software end position (if both are zero, there is no limit)

Homing: 0x6098 ff

Index	Sub-index	Bits	Command Type	Unit	Description
6098	00	08	RW, N	integer	Methods: Important homing methods (from 1 to 34) 34 Set the actual position to zero
6099	01	20	RW, M	inc/64s	Velocity for searching limit switch
6099	02	20	RW, M	inc/64s	Velocity for searching phase-N signal
609A	00	20	RW, M	$16inc/s^2$	Acceleration
607C	00	20	RW, M	inc	Home offset

Motor object: 0x60F6 ff

Index	Sub-index	Bits	Command Type	Unit	Description
60F6	01	20	RW, M	integer	Number of pulse increments per pole length for example, the value for 50-pole, 8000-resolution motor is 160
60F6	02	10	RW, M	integer	Number of poles
60F6	03	10	RW, M	integer	Phase offset of current angle, in proportional to the velocity, in the value range of 80 to 400
60F6	05	10	RW, M	integer	Maximum phase offset of current angle, a quarter of pole length [**INC**]
60F6	06	10	RW, M	integer	Current peak value to drive the load
60F6	07	10	RW, M	integer	Time delay at commutation method 1 or 3 500 small load (1-5 times the motor inertia) 1000 high load (20-50 times the motor inertia)
60F6	09	10	RW, M	integer	Drive method 3 horizontal applications 1 weight compensation, vertical applications
60F6	11	10	RW, M	12A/2047	I^2T current limit 0-2047 effective range
60F6	12	10	RW, M	[S]	Time period to measure the effective current 0-12000 time constant of all motors are about 20 minutes 10 the minimum safe value

121

Speed loop object: 0X60F9 ff

Index	Sub-index	Bits	Command Type	Unit	Description
60F9	01	10	RW, M	inc/s	VC_KP proportional gain of velocity loop 50 soft gain 200 hard gain
60F9	02	10	RW, M	integer	VC_KI integral gain of velocity loop 0 no correction of transient deviations 1 default value 2 strong correction, can cause oscillation
60F9	03	10	RW, M	integer	VC.KILIM, limit value for VC.KI; VC.KILIM shall be higher than 10% of the static current.
60F9	04	10	RW, M	integer	Digital input filter value—velocity loop gain = EFILT*VC.KP 1 Default value, no lead compensation 2 Strong lead compensation 3-5 heave load—low filter
60F9	05	10	RW, M	integer	Digital output filter value (for velocity loop) 2-5: 1-10 times load/motor inertia 15-45: 10-100 times load/motor inertia

Position loop object: 0X60FB ff

Index	Sub-index	Bits	Command Type	Unit	Description
60FB	01	10	RW, M	unsigned	PC_KP proportional value of position loop, for example: **1000** default value, soft correction 3000 value for middle performance 8000 good performance value, with low following error, high position stiffness
60FB	02	20	RW, M	unsigned	PC_AMAX value for the optimal performance and least vibration Value = MO/(load/motor inertia ratio*16*2*PI)*encoder resolution
60FB	03	10	RW, M	integer	PC_.VFFF feedforward velocity gain for high dynamic performance 0-3000 smooth starting 12000-16384 quick starting, low following error, important for master/slave following applications
60FB	04	10	RW, M	integer	PC_AFFF feedforward current for strong dynamic performance value = 2A/2^23*acceleration

Save object: 0x1010 ff

Note that the user can enter **65766173** in the above register to store parameters. The system will then automatically reset the register value to **1**.

Index	Sub-index	Bits	Command Type	Unit	Description
1010	01	20	RW	LOGIC	Store all parameters—value 65766173
1010	02	20	RW	LOGIC	Store communication parameter—value 65766173
1010	03	20	RW	LOGIC	Store application parameter—value 65766173
1010	04	20	RW	LOGIC	Store offline program—value 65766173

CAN-PDO object: 0X1400-0X1A00

0X1400-7 (RX_Parameter/read)

0X1600-7 (RX_mapping)

0X1800-7 (TX_parameter/write)

0X1A00-7 (TX_mapping)

Program object: 0X2000-FF, 0X2120, 0X21221

Note that up to 8 parameters can be saved to a program XX (from (20XX0220=>20XX0320) to (20XX16=>20XX1120))

Index	Sub-index	Bits	Command Type	Unit	Description
2000	01	08	RW	LOGIC	Activation--20XX indicates sequence 0XXX 0: sequence not active 1: sequence active
2000	02	20	RW	unsigned	Parameter 0X20000220, target position of the first parameter in sequence 0
2000	03	20	RW	unsigned	Value of the first parameter in sequence 0
……					……
2000	10	20	RW	unsigned	Target position of the eighth parameter in sequence 0
2000	11	20	RW	unsigned	Value of the eighth parameter in sequence 0

The following 16 registers correspond to 16 triggering operations (triggered by status change of digital inputs DIN1-8). The registers save the number of programs to be called.

2120	01	10	RW	unsigned	Sequence, activated by event DIN1L-H Register value is 0X0020, indicating sequence 0X20 is called
……					
2120	08	10	RW	unsigned	Sequence, activated by event DIN8L-H Register value is 0X0010, indicating sequence 0X10 is called
2120	09	10	RW	unsigned	Sequence, activated by event DIN1H-L
……					
2120	10	10	RW	unsigned	Sequence, activated by event DIN8H-L

DIN1~4 coded combination (used to triggering sequences)

0X21200608=>0X0F00 is another example of coding the sequence number to be called. In the example,

when 0F00 is entered in 21200510 (DIN5 L-H), the sequence number represented by the DIN1-4 binary code will be called when DIN5 changes from L to H.

To complete the above 16 triggering operations, the corresponding calls must be enabled. The value of 0X2121 provides this function. The 0X2118 can directly call any one of 2000~FF sequences.

Index	Sub-index	Bits	Command Type	Unit	Description
2121	00	10	RW	unsigned	High byte to enable H-L, low byte to enable L-H 0X1F1F　　DIN1-5 L-H and H-L are active 0X010F　　DIN1 H-L and DIN1-4 L-H are active
2118	00	08	RW	unsigned	Direct call of sequence number, used in programming 0X21180008=>0X20 Sequence 0X20 is called

Timber object: 0X2130

Index	Sub-index	Bits	Command Type	Unit	Description
2130	01	10	RW	80XX	The number of sequence starts after a waiting time 0X8012　　　starts sequence 0X12
2130	02	20	RW	[ms]	waiting period 0X21300110=>0X8012 0X21300220=>0X03E8 starts sequence 0X12 after 1s

Event object: 0X2140

Index	Sub-index	Bits	Command Type	Unit	Description
2140	01	10	RW	80XX	Sequence XX starts after target reached
2140	02	10	RW	80XX	Sequence XX starts after reference found
2140	09	10	RW	80XX	Servo off state, sequence xx starts when ready output is low
2140	0A	10	RW	80XX	Servo on state, sequence xx starts when ready output is high
2140	0B	10	RW	80XX	Sequence xx starts in Servo on state
2140	0C	10	RW	80XX	Sequence xx triggered when operation mode enabled
2140	10	10	RW	80XX	Sequence xx triggered after input of Enable

Output object: 0X2160 (OUT1), 0X2161 (OUT2)

Index	Sub-index	Bits	Command Type	Unit	Description
2160	01	20	RW	unsigned	Object address of output 1
2160	02	20	RW	unsigned	Offset value, which will be added to the value of output 1 object
2160	03	20	RW	unsigned	AND_value, AND operation is performed between this value and the previous result

2160	04	20	RW	unsigned	Compare_value, to be compared with the previous calculation result
2160	05	20	RW	unsigned	Polarity control, forward/reverse rotation the compared result
2160	08	08	RO	unsigned	Comparison output 0X21600120=>0X60410010 0X21600220=>0X0037 0X21600320=>0XFFFF 0X21600420=>0XC037 0X21600520=>0X0000 OUT1 outputs high level if the drive found commutation and reference and is in operation, and the comparison result is true.

Limit switch object: 0X2170, 0X2171 (LIMIT+), 0X2172 (LIMIT-)

Index	Sub-index	Bits	Command Type	Unit	Description
2170	00	08	RW	LOGIC	Change polarity of the 8 digital inputs 0X2170008=>0X60 (DIN6+7 are low level)
2171	02	08	RW	LOGIC	Bollean and_value for the limit switch (DIN6) 0X21710208 =>0X20 (DIN6 is high if the input polarity is 0X00)
2171	03	08	RW	LOGIC	Bollean compare_value for the limit switch (DIN6) 0X21710208 =>0X20 confirms whether DIN6 is high 0X21710308 =>0X20 default DIN6 (0X40 default DIN7)
2171	04	08	RW	LOGIC	Status of the limit switch 0X00: not high level 0X01: high level, no further motion in positive direction possible

Counter object: 0X2190 (counter 1) to 0X2193 (counter 4)

Index	Sub-index	Bits	Command Type	Unit	Description
2190	01	20	RW,M	unsigned	Value added to counter
2190	02	20	RO	unsigned	Counter value

Capture object: 0X21C0

It is used to capture actual position value.

Index	Sub-index	Bits	Command Type	Unit	Description
21C0	01	10	RW	80XX	Jumps to sequence 0xXX if 21C002 goes from 0 to 1
21C0	02	20	RW	integer	Start counting if there is a L-H event on the N-LIMPULS of the master encoder; the counting direction is positive
21C0	03	20	RW	integer	Result, if event 21C002 takes place, the actual position value is copied into this address

21C0	04	20	RW	integer	Detects the content of actual position if 21C002 goes from 0 to 1

Comparator object: 0X2180, 0X2181, 0X2182, 0X2183

Once the comparison result is true, the whole comparator has to be activated again.

Index	Sub-index	Bits	Command Type	Unit	Description
2180	01	20	RW	unsigned	Object used to compare with the comparator object
2180	02	20	RW	unsigned	Offset to be added to the comparator object
2180	03	20	RW	unsigned	AND_value for Boolean operation with the comparator object, 0XFFFFFFFF by default
2180	04	20	RW	unsigned	Comparison value to be compared with the comparator object
2180	05	10	RW	unsigned	Choice of operation: 0x0001 = equal 0x0002 < less than 0x0003 <= equal to or less than 0x0004 > greater than 0x0005 >= greater than or equal to 0x0006 <> not equal to
2180	06	10	RW	0X80XX	Sequence 0xXX starts if the comparison status is True 0x21800120 =>0x606C0020 (velocity) 0x21800420 =>0x00823555 (1000RPM) 0x21800510 =>0x0005 0x21800610 =>0x8012 (called once)
2180	08	20	RW, M	unsigned	Temporary storage address (storing values in USB01, expressed in hexadecimal system)
2180	09	20	RO	unsigned	Value is 1 if comparison is True; otherwise 0

Arithmetic object: 0X21A0

Index	Sub-index	Bits	Command Type	Unit	Description
21A0	01	20	RW	unsigned	Source object that should be modified
21A0	02	20	RW	unsigned	Destination object that gets the result of the operation
21A0	03	20	RW	unsigned	Number that is operated with the source object
21A0	04	10	RW	unsigned	Arithmetic operation: 0x0000 copy 0x0001 + 0x0002 - 0x0003 * 0x0004 /
21A0	05	20	R	0X80XX	Result of operation:

					0x21A00120 =>0x21900120
					0x21A00320 =>0x00000002
					0x21A00410 =>0x0003(*)
					0x21A00220 =>0x6070A220, last step
					(target position)
					Value from table 1 is multiplied by 2 and is copied
					into target position as destination.

Table object: 0x21B0

Used to write values into the internal table 0x2D00-FF.

Index	Sub-index	Bits	Command Type	Unit	Description
21B0	01	20	RW	unsigned	Source objects should be put into the table
21B0	02	08	RW	unsigned	Write command
21B0	03	08	RW	integer	Position in the table:
					0x21B00120 =>0x60630020
					Actual position
					0x21B00308 =>0xFF
					table position 255
					0x21B00208 =>0x01
					Write command
					The actual position is written into table position
					255—used for teach-in.

Recording object: 0X2201-FF

Index	Sub-index	Bits	Command Type	Unit	Description
2201	01	20	RW	unsigned	First recording object
2203	01	20	RO	unsigned	Array with recording values of the previous object
2201	03	20	RW	unsigned	Second recording object
2203	02	20	RO	unsigned	Array with recording values of the previous object
2201	05	20	RW	unsigned	Third recording object
2303	03	20	RO	unsigned	Array with recording values of the previous object
2201	07	20	RW	unsigned	Fourth recording object
2203	04	20	RO	unsigned	Array with recording values of the previous object
2210	00	10	RW	unsigned	Counter size, specifying how big the array is
2211	00	10	RW	unsigned	Position in the recorded array
2214	00	10	RW	unsigned	Time resolution of the recorded arrays
Recording example: actual velocity					
0x22010120 =>0x606C0020 actual velocity					
0x22140010 =>0x0005 time resolution 5 ms					
0x22100010 =>0x01F4 starts recording of 500 values					
Reading out of the recorded array:					
0x22110010 =>0x1F4 actual velocity value of the first record					
0x22030120 reading [T=0.005]					

0x22110010 =>0x0001   last recorded value

0x22030120 reading [T=2.500]

Monitor object: 0x2400, 0x2401

The ED has two independent analog monitors, and each monitor is corresponding to any internal value. The output range is 0-5V, and 0 is represented by 2.5V.

The scaling formula is:

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

Index	Sub-index	Bits	Command Type	Unit	Description
2400	01	20	RW	unsigned	Object, mapped to analog output 1 (Source value)
2400	02	08	RW	unsigned	Preshift, set to 0, 1, or 2
2400	03	10	RW	unsigned	Factor, normally between 0x0001 and 0x7FFF

The above are parameters for monitoring interface 1. For monitoring interface 2, change the address to 2401. For example,   set the output voltage of the analog output interface 2 to correspond to actual current change

0x24010120   =>0x60780010

0x240102008 =>0x0000   preshift=0

0x24010310   =>0x001E   factor=30

0x24010410 =>(Source * factor) / (256^(preshift+1))

Analog input interface: 0x2508

The analog interface AIN+ and AIN- can be mapped to every internal object (RW). It is mainly used to set the velocity value or the limit of the current value. The internal range of the object is -512 – 512 DAC.

Index	Sub-index	Bits	Command Type	Unit	Description
2508	01	20	RW	unsigned	Mapped target address, for example, target velocity
2508	02	10	RW	unsigned	Factor, calculated according to the formula: maximum value/internal unit/2^SHIFT / 512
2508	03	08	RW	unsigned	Shift, often 0, 1, 2, or 3
2508	04	10	RO	Int	VALUE COMING FROM ADC

For example: +/-10V input for velocity loop with maximum velocity of 1500 RPM

0X25080120   =>0x60FF0020 target velocity

0X25080210   =>0x03   shift is 3

0X24010310   =>0x0C35 factor is 3125

Master/slave object: 0x2509

This object corresponds to the X7 input interface. In case of master/slave or pulse/direction control mode, the following parameters are often used:

Index	Sub-index	Bits	Command Type	Unit	Description
2509	02	20	RW	unsigned	Velocity mapped to (60FF0020)
2509	03	10	RW	unsigned	Numerator of electronic gear ratio
2509	04	10	RW	unsigned	Denominator of electronic gear ratio
2509	05	08	RW	unsigned	Mode:

Index	Sub-index	Bits	Command Type	Unit	Description
					0, 1 is quadruple decoding 2 is pulse/direction control
2509	06	20	RW	unsigned	Master position value, writable and readable, often for other uses
2509	07	20	RW	unsigned	Slave position value

For example: Electronic gear ratio

0X250902200   =>0x60FF0020 target velocity

0X25090310   =>0x07D0   numerator is 2000

0X25090410   =>0x03E8   denominator is 1000

The slave runs twice as fast as the master without having worse performance. It is possible to change gear ratio by changing the comparator or external analog input.

Error code: 0x2600

Index	Sub-index	Bits	Command Type	Unit	Description
2600	01	20	RW	LOGIC	This mask could be used to disable error types
2600	02	20	RW	LOGIC	Error code

00000004	inverse voltage, encoder fault
00000008	encoder counting error
00000010	drive temperature >80°C
00000020	logic voltage <18V
00000040	main circuit voltage >180V
00000080	main circuit voltage <24V
00000100	short circuit of phase A
00000200	short circuit of phase B
00000400	short circuit current at "Ready", "OUT1, 2" or "brake"
00001000	external Enable low, although the drive has been activated
00002000	exceeding error range during operation
00004000	overspeed, encoder frequency > 4MHZ
00008000	motor not found
00020000	I^2*T fault

Boolean control word operation: 0x2840

Index	Sub-index	Bits	Command Type	Unit	Description
2840	01	10	RW	LOGIC	"AND" operation with the actual control word
2840	02	10	RW	LOGIC	"OR" operation with the actual control word

Bus Characteristic object: 0x2F80, 0x2F90-2

Configuration of interfaces RS-485 and CANopen.

Index	Sub-index	Bits	Command Type	Unit	Description
2F80	00	08	RW	unsigned	Address number 0-127 corresponding to DIP switch value
2F81/82	00	08	RW	unsigned	Baud rate of CANBUS

						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
2F90	00	08	RW		unsigned	Interface type: 0 Standard RS-485 1 JETTER RS485
2F91	00	08	RW		unsigned	Baud rate 0x3F 9.6K 0x1F 19.2K 0x0F 38.4K
2F92	00	08	RW		unsigned	Index high 0x60 for JETTER, to indicate the address of high byte 0x60XX

Device information and software version object: 0X2FE0

Index	Sub-index	Bits	Command Type	Unit	Description
2FE0	01	20	RO	unsigned	Software version
2FE0	03	20	RO	unsigned	Date of modification 0x0315490 means 20010128, which says the date of 1/28/2001.

Glossary

Mapping

The value of an object at address Subindex-Index-Bitlength is another object address. This pointer concept is known at many high level programming languages. For example the actual master position address is 2509-06-20. To get a relative position, map it at the address of the comparator target 2180-01-20.   This procedure is called mapping and generates a very powerful concept. Objects capable of being mapped are signed with "M" in the object catalog.

Statemachine

The motor plus drive is defined as a system with different discrete states. From the mechanical engineering the space description where (s(t), v(t), t) determines the whole system at any time is known. This is also true for this drive-motor system, but it is also important to know whether the power is turned on, whether there is an error or whether the nominal target is reached and some more things.   In CANopen this is described by a status word. Its bitcode determines a special drive status. By use of the control word it is possible to change the states of the drive. Now the following important tasks are described, by which we can observe this statemachine:

- Switch on the drive
- Start homing
- Positioning in absolute mode

Object	Description
Status word 60410010	Consisting of important flags: error, communication, Ready, target reached, reference found, motor switched on/off
Control word 60400010	Determines states: power on motor, Enable, Reset, Start motion, absolute or relative motion
Operation mode 60600008	For example: 1. Positioning, 6. Homing
Homing 60980008	For example, 32 means "at next n-phase pulse"
Velocity 60810020	Velocity v(t) curve
Acceleration 60830020	Positive slope in v(t)
Deceleration 60840020	Negative slope in v(t)
Target position 607A0020	Position increments

Therefore we have the following course:

Action/course	Control Word	Status Word	Status
Logic on!	0x0006	0x0031	Ready to switch on
Power on!	0x000f	0x4437	Communication found, no error, power on
Power off!	0x0006	0x4031	Power off

Choose the homing method: Number 32 (60980008 => 0x20) and start homing:

Action/Course	Control Word	Status Word	Operation Mode	Status
	0x000f	0x4437	0x01	Motor and controller powered on

homing32		0x06	
Start	0x001f		
		0xD437	Reference found
Ready to make an absolute/relative positioning	0x000f 0x004f	0x01	
		0xC437	Set Point/Start" Fifth bit in control word is set

To start motion, firstly initialize the value for the acceleration, deceleration and operation velocity. By these minor operations, we can handle 80% of communication. To reset the system, enter 0X80H in the control word and check whether the status word is 0x0031.

Action/Course	Control Word	Status Word	Velocity	Object Position	Status
	0x000F	0xC437	800inc/s	8000	Initialize
Start	0x001F				
		0xD037			Start motion
		0xD437			Target reached

Address

An address is specified in the following way: First the index (consisting of four hexadecimal numbers), then the subindex (consisting of two hexadecimal numbers), and at the end a two hexadecimal code for the bit length of the data. For example:

(Index, Subindex, Bitlength) = (6041,00,10) or (60410010)

Bitcode

Bitcode means to represent several event states by a code (1000 1110), which could be 8 bit, 16 bit or 32 bit and convert them into hexadecimal numbers.   An example is the distribution of the states high/low at the 8 digital inputs. Code DIN6 to 8 means high (active) and all the others mean low by the number (1110 0000) which is 0xE0.

Velocity internal scaling

RPM = encoder resolution * 64/60 [**internal unit: inc/64 s**]

Analog dem value

[**-10V ..10V**] - > factor [**internal unit**] /conversion factor [**internal unit/ volt**] /2^SHIFT/ 512,

Maximum velocity is 1440U/min => 1440/ [**60/ 8000/ 64**] / 2^3 / 512 = 3000.

Acceleration

When the unit is rad/s^2, it equals to the internal value [**16 inc/s2**]*16*2*pi /encoder resolution. For example, when a pulley runs at 100 mm/revolution and the encoder is 8000 inc, it equals to 9.81 /(0.1/ 2*pi)*8000 / (16*2*pi) = 4969 [**16 inc/s2**].

Digital current

The current value ranges from 0 to 2047, which can be converted in the drive: current value = value *

maximum drive current [**A**]/2047.

The ED100 can reach 8A, the ED200 can reach 12A, and the ED216 can reach 14A in rated phase current.

Digital friction current

Digital friction current [**Idac**]:   can be measured indirectly by averaging the current object address 6078,00 at slow speed. This value is multiplied by 1.2 and can be used as first approximation of the limit value of the integral parameter in the velocity loop vc_kilim.

Reference storage

Save the set parameters to the drive. In this way, new parameters will take effect when the drive is powered off and then rebooted.

Appendixes

Appendix A: Technical Specifications of the ED Drive

Operating Modes

Positioning drive by fieldbus	CANopen, Profibus-DP, RS-485 or RS-232
Positioning drive by PLC interface	Digital I/O
Positioning drive by pulse/direction signal	RS-422, 24V or 5V signals
Velocity controlled by analog command	10Bit resolution, ±10V analog input
Master/Slave positioning	RS-422, 24V or 5V encoder signals

Controller Loop

Sampling time for velocity controller	0.25ms (4KHz)
Sampling time for position controller	1.0ms (1KHz)
Sampling time for current controller	0.0625ms (16KHz)

Power Consumption Specifications

Max. RMS phase current	ED100: 5.6Arms; ED200: 8.6Arms
Max. phase current	ED100:8A (DC); ED200: 12A (DC); ED216: 14A (DC)
Max. output voltage	UDC-BUS
Max. output power	ED100: 0.5Kw; ED200: 1.8Kw; ED216: 3.6Kw
Short-circuit protection of motor output	Motor phases to DC BUS and between motor cables
Min. inductance of motor winding	>=0.5mH (UDC-BUS<=70V) >=1.0mH (UDC-BUS<=170V)
Length of motor cable	Max. 10m (otherwise consult supplier)
Peak frequency of output current	16.4KHz

Electrical Specifications

Control power supply	18~30VDC ripple < 10%
Fuse of control of power supply (recommended)	3A
DC bus power supply	ED100: 24~70V DC; ED200: 24~170V DC; ED216: 24~170V DC
Fuse of bus power supply (recommended)	10A (ED200), 20A(ED216)
Short-circuit protection of motor output	Motor phases to DC BUS and between motor cables

Operating Environment

Operating temperature	0~40°C
Storage temperature	−10~70°C
Humidity (non-condensing)	5~95% (RH-2 according to IEC 61131-2)
Pollution degree	2 (according to IEC 61131-2)
Protection class	IP20
Place of installation	Dust-free, dry, and lockable (for example, an electrical cabinet)
Installation position	Vertical (refer to "ED Installation")

Installation altitude	Up to 1000m above sea level (full range)
Power loss	
ED100	
@ 5.6Arms (UDC-BUS<=70V)	<=20W
ED200	
@ 5,7Arms (UDC-BUS<=170V)	<=40W
@ 7,5 Arms (UDC-BUS<=100V)	<=40W
@ 8,5 Arms (UDC-BUS<=170V)	<=70W
Cooling	Convection
	Heat sink required if power loss > 40W

Mechanical Specifications

Housing	Aluminum, chromium plated surface
Dimension without heat sink (HxBxT mm)	ED100: 200x87x50 (without mating connectors)
	ED200: 240x62x170 (without mating connectors)
Dimension with heat sink (HxBxT mm)	ED200: 240x102x170 (without mating connectors)
Weight (without heat sink)	ED100: 0.5kg; ED200: 1.8kg
Weight (with heat sink)	ED200: 3.4kg
Fixing cable and strain relief	Metal clips, max. cable diameter 15 mm
Connector for CAN	9-pin male connector
Interface: RS485(X2), RS232(X5) Encoder output (X6) Master encoder input (X7) Encoder (Motor) Input (X8)	9-pin female connector
I/O connectors (X3, X4)	Pluggable terminal with screw: Phonix MC-1, 5/3, 8 1
Motor connector	Pluggable terminal with screw: Phonix IC-2,5/3, 08
Bus power supply connectors (X10)	Hot-pluggable terminal with screw: Phonix MSTB-2,5/5, 08

Communication Interfaces

CAN	CANopen, conforming to DS301, DSP402 Max. 1M baud rate, max. 127 nodes Max. cable length 40m @ 1M baud rate Max. cable length 1Km @ 50k baud rate
RS-232	RS-485 (2/4-wire) 9.6K baud rate, max. 15 nodes Max. cable length 10m
RS-485 (2/4-wire)	CANopen, conforming to DS301, DSP402 38.4K baud rate, max. 15 nodes Max. cable length 400m
Profibus DP	Max. 12M baud rate, max. 127 nodes

Motor Encoder Input

Motor encoder supply	DC 5V, max. 0.2A (provided by drive)

Signal specification	Differential TTL line driver (RS-422: A, B, N, /A, /B, /N)
Input frequency	Max. 2 MHz
Pulse width	Min. 0.125 s
Edge clearance	Min. 0.1 s
Electrical isolation	None

Motor Encoder Output (Unavailable with ED100)

External supply	4.8~5.2 VDC (max. 0.15A)
Signal specification	Differential TTL line driver (RS-422: A, B, N, /A, /B, /N)
Output frequency	Max. 2 MHz
Electrical isolation	None

Master Encoder Input

Master encoder supply (provided internally by the drive)	5V DC (max. 0.2A) or 24V DC (max. 0.1A)
Signal specification	Differential TTL line driver (RS-422: A, B, N, /A, /B, /N) or 24V signal (A, B, N)
Output frequency	Max. 2 MHz
Edge clearance	Min. 0.1 s
Pulse width	Min. 0.125 s
Electrical isolation	None

Digital Inputs

Number of inputs	10, among which 8 programmable
Rated input voltage	DC 24V
Input voltage range	DC20 ~ 30V
Input current	About 4 mA
Input impedance	5kΩ
Input delay	About 1 ms
Input voltage for state "ON"	>13V
Input voltage for state "OFF"	<4V
Electrical isolation	None

Digital Outputs

Number of outputs	3, among which 2 programmable
Type of outputs	High level +24V
Rated output voltage	24V DC
Output voltage range	20~30V
Output current	Max. 0.5A
Brake output	24V, max. 1A (reduced to 0.5A after 100 ms)
Regeneration energy of inductive loads	Max. 0.2J (internal Zener diodes provide fast

	demagnetization of inductive loads)
Electrical isolation	None

Analog Input

Input voltage range	±10V (differential input)
Input voltage	Max. 15V
Input impedance	About 100 kΩ
Input delay	About 0.1 ms
Resolution	10Bit
Electrical isolation	None

CE conformance

EMC Directive (Test conditions conform to "EMC Installation Procedures")	Conforms to EC Directive 89/336/EEC EN 50082.2 (interference immunity) EN 55011, Class B (RFI Emission)
Low voltage directive	Conforms to EC Directive 73/23/EEC EN60204.1 / VDE113, EN50178 /VDE160

Appendix B: Example of Sequencer Programming

1. Application

Motor resolution: 1 revolution = 8000 inc

Control of ED drive by PLC using I/O

Predefined functions:

Drive On/Off (=Enable/Disable motor current)

Homing

11 positioning curves

Jog mode forward/reverse

Monitoring actual velocity and actual current of the motor by analog inputs

Output interfaces output a signal indicating the target position is reached

2. Definition of I/O

DIN1..4   : Coding and control of sequences

DIN5   : Triggering and calling sequences

DIN6   : Jog switch forward

DIN7   : Jog switch reverse

DIN8   : Homing switch

RESET   : Fault reset

ENABLE   : External Enable

OUT1   : Digital output (PLC standard, maximum current = 0.5A)

High level, if drive = ON & no error & reference = valid & target position (positioning mode)

reached & target velocity (jog mode) reached

OUT2   : Digital output (PLC standard, maximum current = 0.5A)

High level, if ENABLE=1 & no error

3. Operation Sequence

After power on, the "Driver on" sequence must be activated. Homing should be performed before positioning. Subsequently, to perform positioning, it is necessary to execute the 0C sequence in the positioning mode.

4. Digital Definition

DIN1	DIN2	DIN3	DIN4	Seq.	Function	Target Position (inc)
1	1	1	1	0F	Driver On	
0	1	1	1	0E	Driver Off	
1	0	1	1	0D	Homing	
0	0	1	1	0C	Set to positioning mode	

1	1	0	1	0B	Positioning curve #1	+4000inc (relative)
0	1	0	1	0A	Positioning curve #2	+2000inc (relative)
1	0	0	1	09	Positioning curve #3	-2000inc (relative)
0	0	0	1	08	Positioning curve #4	-4000inc (relative)
1	1	1	0	07	Positioning curve #5	+8000inc (absolute)
0	1	1	0	06	Positioning curve #6	+4000inc (absolute)
1	0	1	0	05	Positioning curve #7	+2000inc (absolute)
0	0	1	0	04	Positioning curve #8	0inc (absolute)
1	1	0	0	03	Positioning curve #9	-8000inc (absolute)
0	1	0	0	02	Positioning curve #10	-4000inc (absolute)
1	0	0	0	01	Positioning curve #11	-2000inc (absolute)
0	0	0	0	00	Set to jog mode	

5. Definition of Sequences

Seq.0F (Driver on)

60400010 = 6	Control word	Motor off
60810020 = 0	Profile velocity	Max. velocity = 0.1 inc/s in positioning mode
607A0020 = 0	Target position	Target position = 0 inc
21600320 = 0	Output0.and_mask	Output0.and_mask=0
21600420 = 1	Output0.cmp_mask	Output0.cmp_mask=1
21180008 = 1F	Run sequence	Call Seq. 1F

Seq.1F

60400010 = 15	Control word	Motor on

Seq.0E (Drive off)

60400010 = 06	Control word	Motor off

Seq.0D (Homing)

21600320 = F0FF H	Output0.and_mask	Output0_cfg.and_mask
21600420=D0C7	Output0.cmp_mask	Output0_cfg.cmp_mask:OUT1,if commutation=found & home=found & operation = enabled
60600008 = 6	Chosen mode of operation	Set to homing mode
28400110 = FFCF	Controlword_bits.and	For control word AND, prepared to start homing
21180008 = 1D	Run sequence	Call Seq.1D

Seq.1D (Homing)

28400210 = 16	Controlword_bits.or	Bit4=1, start homing

Seq.0C (set to positioning mode)

21600320 = C4FF H	Output0.and_mask	
21600420=C437	Output0.cmp_mask	OUT1=1, if status word satisfies: commutation=found & home=found & operation = enabled
60600008 = 1	Chosen mode of operation	Set to positioning mode 1
28400110 = FFCF	Controlword_bits.and	For control word "AND", bit4, bit5=0, prepared to start positioning

Seq.0B (Positioning curve #11)

607A0020 = 4000	Target position	Target position: 4000 inc
28400110 = FFBF	Controlword_bits.and	Output0_cfg.cmp_mask:OUT1,if commutation=found & home=found & operation = enabled
60600008 = 6	Chosen mode of operation	Set to homing mode 6
28400110 = FFBF	Controlword_bits.and	For control word AND, bit6=0, absolute positioning mode
60810020 = 80000	Profile velocity	maximum velocity: 80000 inc/s
60830020 = 500000	Profile acceleration	500000 * 16 inc/s
60840020 = 500000	Profile deceleration	500000 * 16 inc/s
21180008 = 0C	Run sequence	Call Seq.0C and be prepared for next positioning

Sequences 0A to 08 similar to sequence 0B, with different target positions (relative)

Seq 07 (Positioning #7)

607A0020 = 8000	Target position	Target position: 8000 inc
28400110 = FFBF H	Controlword_bits.and	Output0_cfg.cmp_mask:OUT1,if commutation=found & home=found & operation = enabled
28400210 = 30 H	Controlword_bits.or	Bit4,bit5=1, start absolute positioning
60810020 = 80000	Profile velocity	Maximum velocity: 80000 inc/s
60830020 = 500000	Profile acceleration	500000 * 16 inc/s
60840020 = 500000	Profile deceleration	500000 * 16 inc/s
21180008 = 0C H	Run sequence	Call Seq.0C and be prepared for next positioning

Sequences 06 to 01 similar to sequence 07 to define different target positions (absolute)

Seq.12 (Jog stop)

60FF0020 = 0	Target velocity	Target velocity = 0 inc/s

Seq.11 (Jog reverse)

60FF0020 = -8000	Target velocity	Target velocity = -8000 inc/s
60830020 = 16000	Profile acceleration	1000 * 16 inc/s
60840020 = 160000	Profile deceleration	10000 * 16 inc/s

Seq.11 (Jog forward)

60FF0020 = 8000	Target velocity	Target velocity = 8000 inc/s
60830020 = 16000	Profile acceleration	1000 * 16 inc/s
60840020 = 160000	Profile deceleration	10000 * 16 inc/s

Seq. 00 (Jog mode)

21600320 = C4FF H	Output0.and_mask	
21600420=C437 H	Output0.cmp_mask	Output0_cfg.cmp_mask:OUT1,if commutation=found & home=found & operation = enabled
60600008 = 3	Chosen mode of operation	Set to profile velocity mode 3
60810020 = 0	Profile velocity	Maximum velocity: 0 inc/s
60FF0020 = 0	Target velocity	Target velocity = 0 inc/s
28400210 = 30 H	Control_bits.or:	bit4, bit5=1, be prepared to jog

Definition of Inputs:

Definition of Homing Parameters

Definition of software position limits:
Set min. software position limit=0
max. software position limit=0.

Definition of Outputs:

When commutation=found & home=found & operation=enabled & target=reached, high level will be output.

Definition of Analog Monitors:

Every analog monitor output (mon1/2) has an output range of 0 to 5V (2.5V represents 0 of the mapped value). Every object in the drive can be mapped to the analog monitor output by the following formula:

Umon=1V * internal notation of dimension * factor / 256^(preshift+1)/120

According to the following example, analog output 1 is mapped to the actual motor current and monitor output 2 is mapped to the actual motor velocity:

Monitor 1:   Mapping   =   60780010   (curr_act_val)

              Pre-shift=   0

              Factor   =   30

Umon1 = 1V*2047/12A*30/256^(1+0)/120 = 0.166V/A (lmon1: -12A..12A = 0..5V)

Monitor 2:   Mapping   =   606C0020   (vel_actual_val)

              Pre-shift=   2

              Factor   =   118

Umon1 = 1V * 20000000/2344rpm * 118/256^(1+2)/120= 0.5mv/rpm (Nmon1: -5000..5000rpm = 0..5V)

Saving all parameters:

"Administration->Save parameters to device now".

Appendix C: Application of ED Drive

1. Position and Velocity Control by Pulse/Direction Signal

When the Pulse/Dir signal is used to control the ED servo drive, the Pulse/Dir signal can be the TTL or differential signal provided by such pulse sources as PLC and motion control card. Different modes have different wiring methods. To prevent the impact of surrounding environment on the pulse signal, we design a PDC servo connector to connect the servo drive with the external pulse signal source.

1). Wiring diagram:
Connect the ED servo drive to the Siemens S7200 as follows:
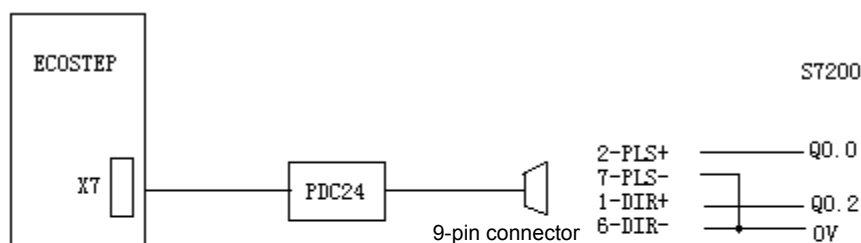
S7200 uses P/D mode to control ECOSTEP
Single-end wiring, PLS- and DIR- short circuited to the 0V of the S7200, PLS+
connected to impulse output interface, DIR+ connected to direction output interface.
Note that the direction signal shall be set up before the pulse signal.

Connect the ED servo drive to the Mitsubishi FX2N-16MT
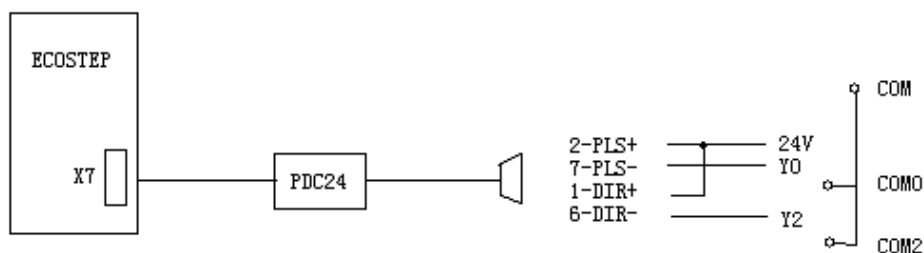
In the event of differential P/D input (signal level 24V/5V), it can be connected with a PDC24 or PDC5 adapter.
For details of the PDC24/5 adapter, refer to the document PDC adapter usage and performance description.pdf.

2). Parameter setting

a) In Sequencer programming->Sequences, select sequence 01 and sequence 02 and set them as follows:

Seq01.
    60400010=6   Motor off
    21180008=2   Go to sequence 2

Seq02.
    60400010=f   60400010=f
    60600008=-4   Master/slave mode
    25090220=60ff0020   Velocity mapping
    25090508=2   Receiving pulse/direction signal control
    25090310=10000   Factor of gear ratio, modifiable
    25090410=10000   Divider of gear ratio, modifiable

Seq03.
    25090220=0   Canceling the control of pulse/direction
                                          signal

Seq04.
    60400010=6   Motor off

b) To enter the P/D mode after power-on, set it as follows:
    Set 21400a10=8001 in Direct Object entry. The drive will directly call the first sequence after power-on.
    Remember to save the program after entering the word. Exit the system, turn off the system, and then
    reboot the system to validate the setting. If other servo parameters are set later, the content in 21400A10
    will be lost. It is necessary to repeat the setting procedure.

c) To make the drive enter the P/D mode through the I/O,
    Connect the DIN1 in the drive to the output of the external controller. Trigger the DIN1 through an
    external controller (for example, a PLC), and implement control over the ED drive through external
    pulse/direction signal by calling the sequence 1. DIN2 is to cancel the control over the ED drive through
    external pulse/direction signal, and Din3 is to loosen the motor shaft.
    The ED has eight inputs (DIN1-DIN8) for external calling. Each input can be set to correspond to two
    internal sequences (rising and falling edges). The input requirements include 24V 4mA, high level >12V,
    low level <5V, and input delay 1 ms.
    For details, refer to the hardware manual.
    The settings are as follows:
    Sequencer programming->Digital input (the following setting represents a valid rising edge):

d) If the drive generates an alarm during operation: Following Error, set Axis1->Parameter settings->Position controller->P-Gain position control =100-200. After setting, save the configuration (Administration->Save parameters to device now), as shown in the figure below:

e) Receiving monitor pulse

The **Device->Configuration->Electronic gear-> Actual position master** indicates the actual pulses received by the drive.
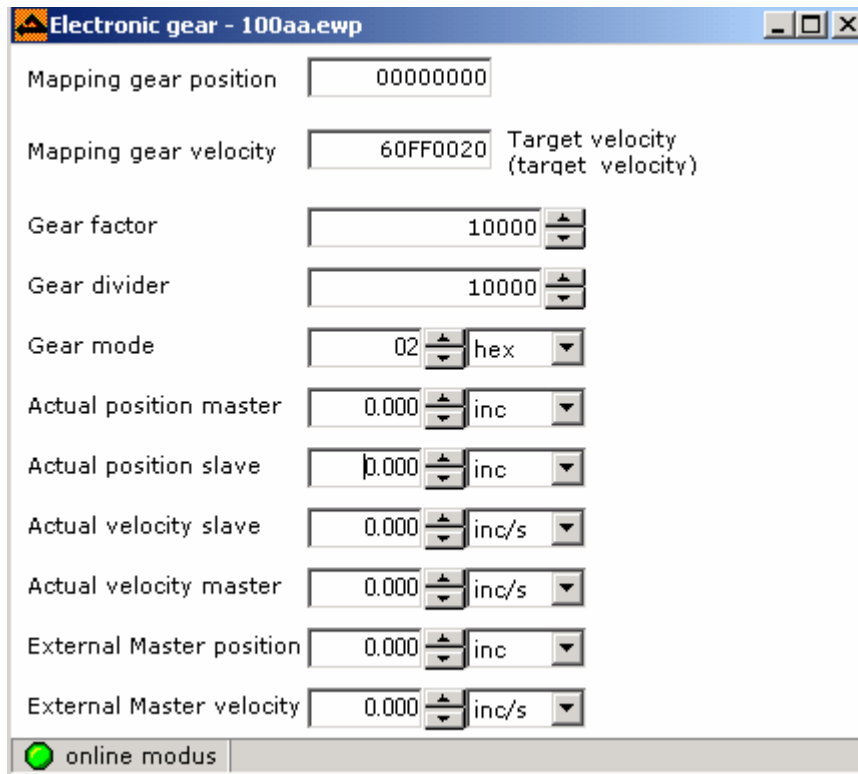
f)   Save all configurations (Administration->Save parameters to device now).

Note: The direction signal shall be established before the pulse signal; otherwise, the pulse may get lost.
          PDC adapter is a kind of adapter developed by us to convert TTL signal into differential signal.

2.   Master/Slave Control

In the previous section Position and Velocity Control by Pulse/Direction Signal, if the user sets 25090508 (gear mode) to 0, the control mode will change to master/slave control mode. The position and velocity signals of the servo drive are encoder signals from the master encoder interface of the ED drive.

3. CW/CCW Pulse Signal Control

Settings of the CW/CCW pulse signal control are basically the same as those in the pulse/direction signal control mode, with the exception of Gear mode = 3. Refer to the following table in wiring. The wiring method is the same as that described in the Position and Velocity Control by Pulse/Direction Signal section.
X7 (MASTER ENCODER)

Pin No.	Signal
2	PLS+/CW+
3	DIR+/CCW+
7	PLS-/CW-
8	DIR-/CCW-

4. Control of the Velocity and Position of the ED Servo Motor by Analog Inputs

1). Wiring
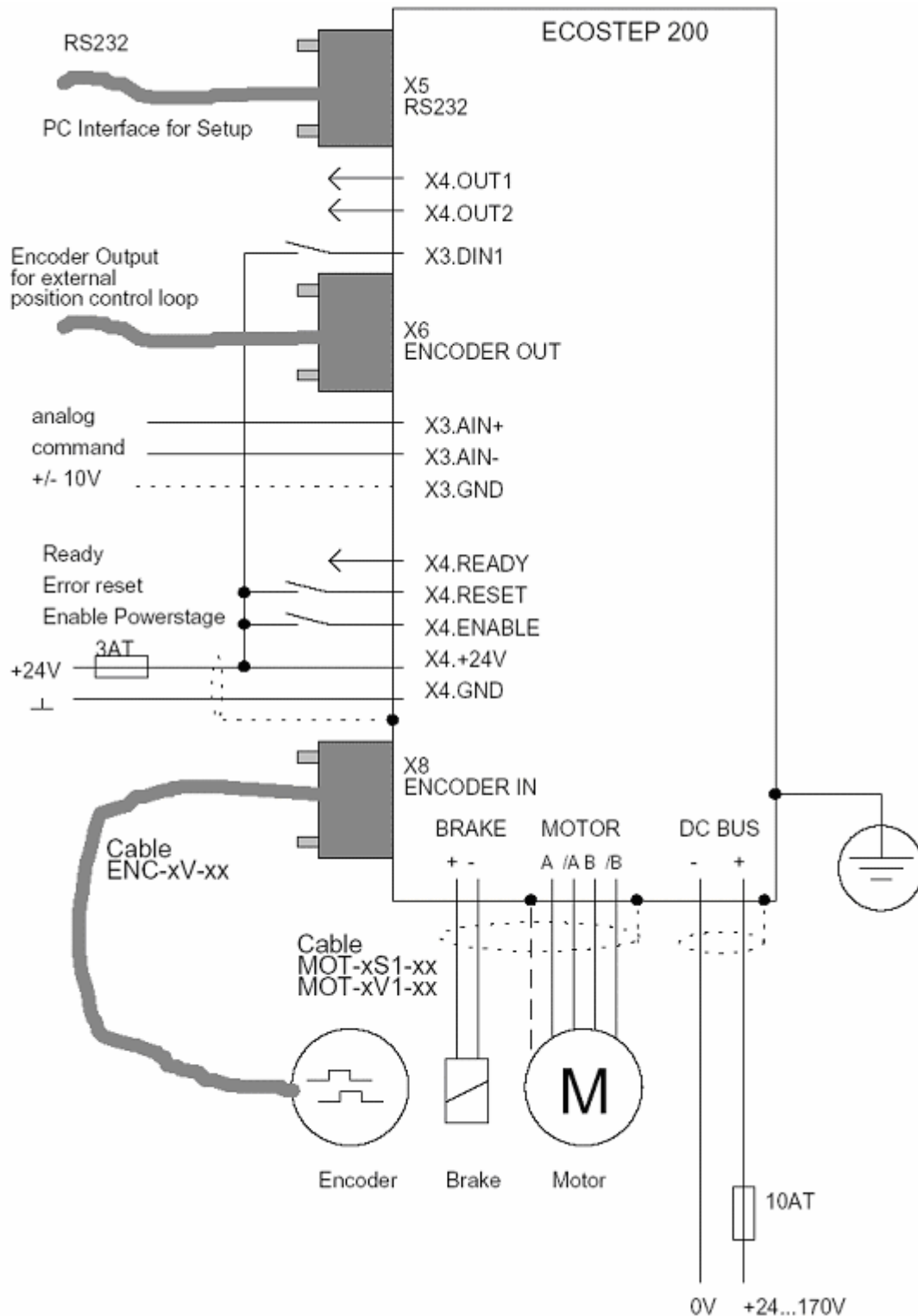
2). Configuring digital inputs

The rising edge of DIN1 triggers Seq01, turns on the drive, and makes the drive enter the analog input control mode. The falling edge of DIN1 triggers Seq02, turns off the drive, and loosens the motor shaft.

3). Designing the program

Seq01:

Seq02:

4). Setting analog ports

For example, suppose the input +/-10V corresponds to the maximum forward/reverse rotation of 1500RPM. According to the formula in Section 5.6.5, set shift=1, then factor=12500. The settings are shown in the figure below:

If the user changes the mapping object to object position 607A0020 and changes the control mode to 1, the user can control the position by analog inputs.

5). Saving configurations

Upon completion of above operations, save the configurations (Administration->Save).

Appendix D: Communication Examples

1. Connection of Touch Screen with ED Servo Drive and Application

The eView touch screen can be directly connected with an ED drive through the RS-232 interface. The user can set the parameters and operation status of the ED drive through the touch screen. An eView touch screen can be connected to either a single ED drive or several ED drives through RS-232C or Rs-485.

1). Install driver file

To enable the eView touch screen software EasyBuilder500 to recognize the ED drive, copy the driver of the ED drive ecostep.PDS to the directory of C: \EB500\Drivers.

2). A touch screen controls a single ED drive

a. Wiring

```
        Touch screen (PLC RS-232) interface                    ED X5 (RS-232)
                        RxD 3     --------------------------   2 TX
                        TxD 2     --------------------------   3 RX
                        GND 5     --------------------------   5 GND
```

b. Setting communication parameters

For the settings of communication parameters of the touch screen, refer to Figure D1.1. The PLC station No. must be the same as the ID of the ED drive. The default ID of the ED drive is 1, so the PLC station No. should be set to 1 when the touch screen controls a single ED servo drive.

Figure D1.1

If the ID of the ED drive is set to N (1-15), the PLC station No. of the touch screen should be set to N also.

c. Setting address parameters
It is necessary to set the address of the ED object when designing eView program. Firstly, set the Index and then the Subindex. As shown in the figure below, set the data input component to the value of the ED object 607F0020 (maximum limit).
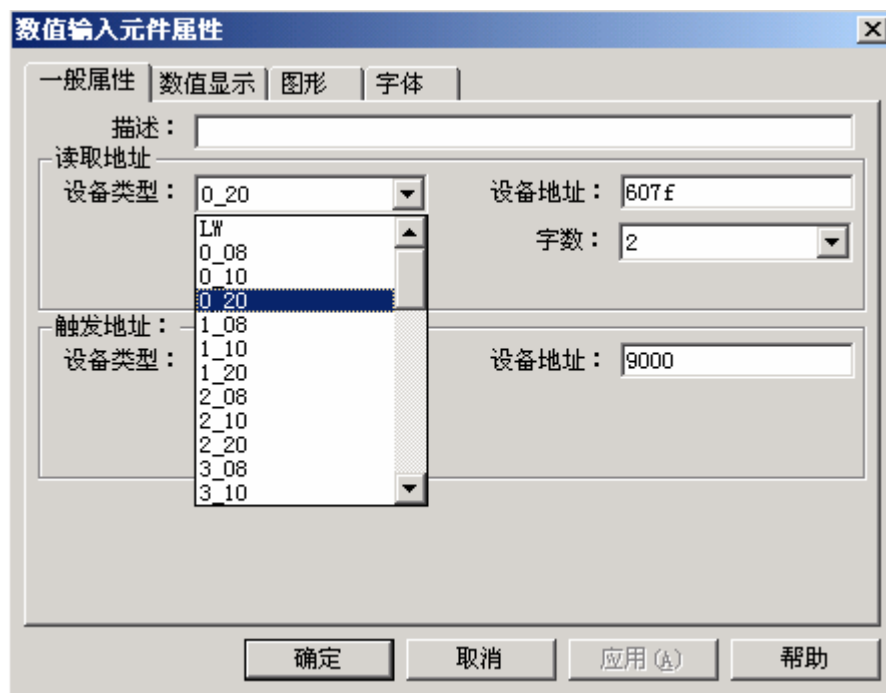
Figure D1.2

3). A touch screen controls several ED servo drives

a. Wiring
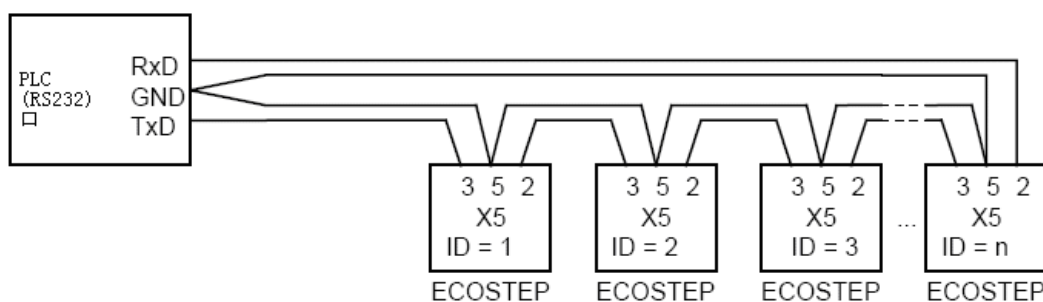
Figure D1.3

2. Parameter setting

Parameter settings for the touch screen are the same as those in Figure D1.1. Note that the PLC No. must be different from the ID of the ED drive. For example, if the touch screen controls two ED servo drives and the IDs for them are 1 and 2 respectively, the PLC station No. in the touch screen must be set to a number different from 1 or 2, for example, set to 3 or any other value.

When the eView touch screen controls several ED servo drives, it uses extended address mode instead of standard mode. This is different from the touch screen controlling a single ED servo drive. Set the parameters in the **Edit→System Parameters→Editor** menu in the EasyBuilder500 software, as shown in the figure below:
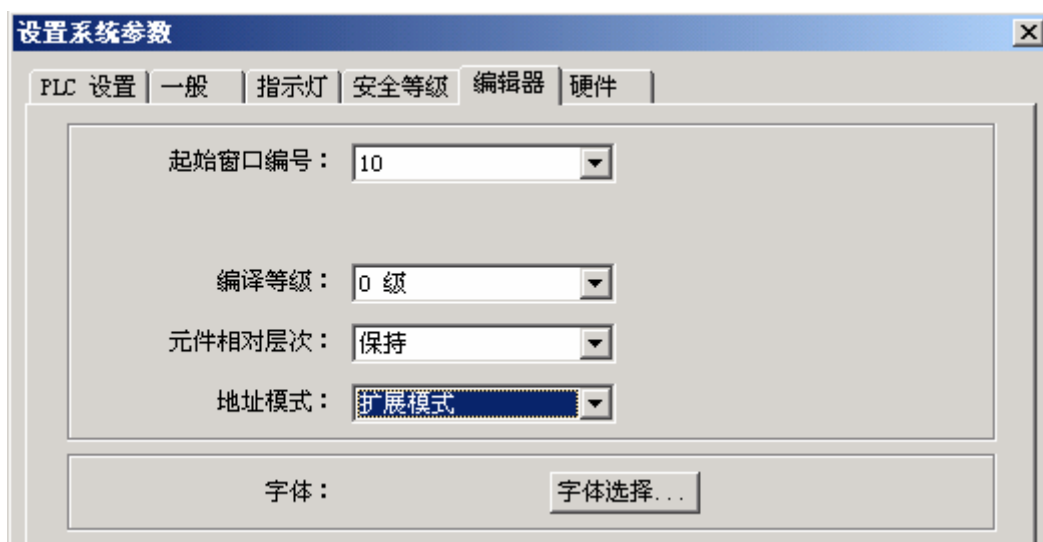
Figure D1.4

c. Setting address parameters
It is necessary to set the address of the ED object when designing eView program. Firstly, set Index and then Subindex. Different from address setting of the touch screen controlling a single ED servo drive, address has to be added before the Index. For example, to control the object with the ID of 1, add **1#** before the Index. As shown in the figure below, set the numeric input component to the ED object **607F0020** (maximum velocity), and enter **1#607f** in equipment address box, and enter **0_20** in equipment type box.

Figure D1.5

    4). Example
For example of the EasyBuilder500 driver file and the example of programs for the eView touch screen to control the ED drive, refer to the CD of the Kinco servo drive.
The following is an interface in the example:

Figure D1.6

The user can also use a PC to control the ED drive through the COM port and by the online simulation mode of the Easybuilder, without the need of compiling any serial communication program.

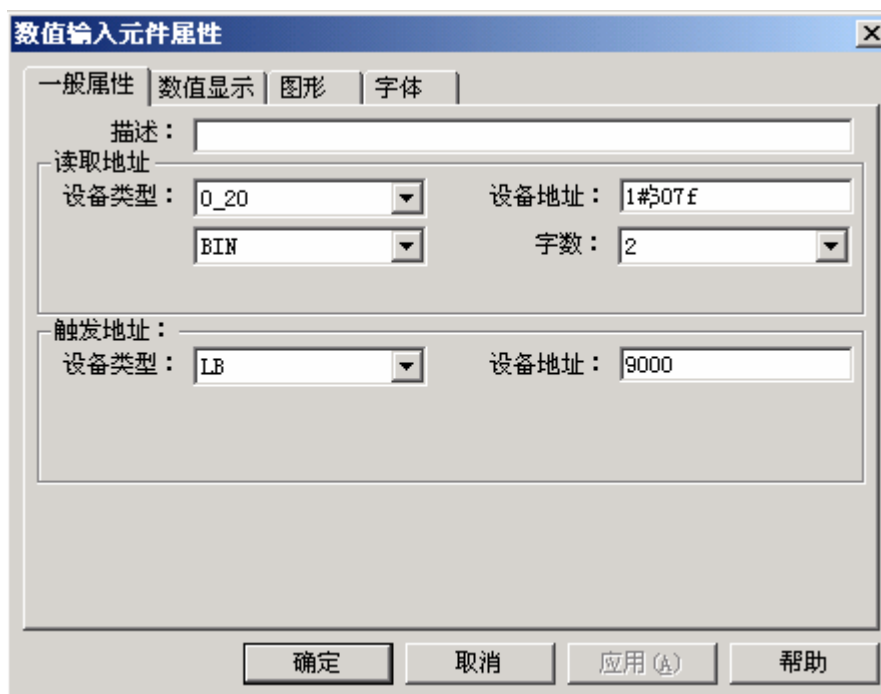## 2. Communication between Siemens S7200 PLC and the ED through RS-485 Interface

1). Setting communication parameters

S7200 end:

S7200 PLC must use the free port communication protocol, SMB30=09 (free port mode);

The maximum length of buffer to be sent and received by the S7200 PLC is set to 256 bytes. In this example, set VB500H (VB1280) ~ VB50AH (VB1290) as the sending buffer and set VB536H~VB546H as the receiving buffer.

ED drive end:

RS-485 communication parameters on the ED drive end: baud =9600bps, Data length=9, Stop bit = 1, Parity=none

The ED drive must use the RS-485 interface.

2). Wiring

They communicate with each other through the RS-485 interface:

ED200	RS-485 interface		S7200		eView HMI
7, 8(RX-)	———————		8(D-)	------------	1(RX-)
2, 3(RX+)	———————		3(D+)	------------	2(RX+)
5(GND)	———————		5(GND)	------------	5(GND)

3). Functional description

S7200 PLC Input	Function	Description
I0.0	60600008=-3	Set to immediate velocity mode
I0.1	60FF0020=1000000inc/s (117rpm)	Set target velocity in velocity mode
I0.2	60600008=0x01	Set to positioning mode
I0.3	607A0020=	Set target position in the positioning mode
I0.4	60810020=60rpm	Set the maximum velocity in the

		positioning mode
I0.5	60400010=0x5F	Control command = relative motion
I0.6	60400010=0x06	Motor off
I0.7	60400010=0x0F	Motor on
I0.8		

4). System operation

Firstly enter 10.6, then enter 10.7, and enable the motor;

Set the controller to the velocity mode and set target velocity;

Set the controller to positioning mode, set maximum velocity and target position, and set the motor to relative or absolute positioning.

5). Others

S7200 can be connected to the X2 (RS-485) interface of the KINCO servo through the RS-485 interface. The eView touch screen of the user can be connected to the KINCO servo to implement monitoring and parameter setting functions and to deliver a system with complete input, output, display and setting functions. For sample programs, refer to the user CD of the Kinco servo.

3. Communication between Panasonic FP0 (with RS-232 Interface) PLC and ED

1). Setting communication parameters

In the PLC system register settings in the Panasonic PLC programming software, set No.412 = universal communication, NO413 = data bits 8, stop bit=1, no parity check, receiving buffer DT2000, and capacity 1000.

2). Wiring

           PLC RS-232 Interface   ED Drive RS-232 Interface
               R   ————————————————   2TXD
               S   ————————————————   3RXD
             GND   ————————————————   5GND

3). Program

Store the commands to be used in the data area in advance.

```
R9013
 ├──┤ ├──[F11 COPY   ,   K 0      ,   DT 400   ,   DT 500   ]
        ├─[F0 MV    ,   H 2B01    ,   DT 401   ]
        ├─[F0 MV    ,   H 6040    ,   DT 402   ]
        ├─[F0 MV    ,   H 600     ,   DT 403   ]
        ├─[F0 MV    ,   H 0       ,   DT 404   ]
        ├─[F0 MV    ,   H 2E00    ,   DT 405   ]
        ├─[F0 MV    ,   H 2B01    ,   DT 411   ]
        ├─[F0 MV    ,   H 6040    ,   DT 412   ]
        ├─[F0 MV    ,   H 3F00    ,   DT 413   ]
        ├─[F0 MV    ,   H 0       ,   DT 414   ]
        ├─[F0 MV    ,   H F500    ,   DT 415   ]
        ├─[F0 MV    ,   H 2B01    ,   DT 421   ]
        ├─[F0 MV    ,   H 6040    ,   DT 422   ]
        ├─[F0 MV    ,   H 5F00    ,   DT 423   ]
        ├─[F0 MV    ,   H 0       ,   DT 424   ]
        └─[F0 MV    ,   H D500    ,   DT 425   ]
```
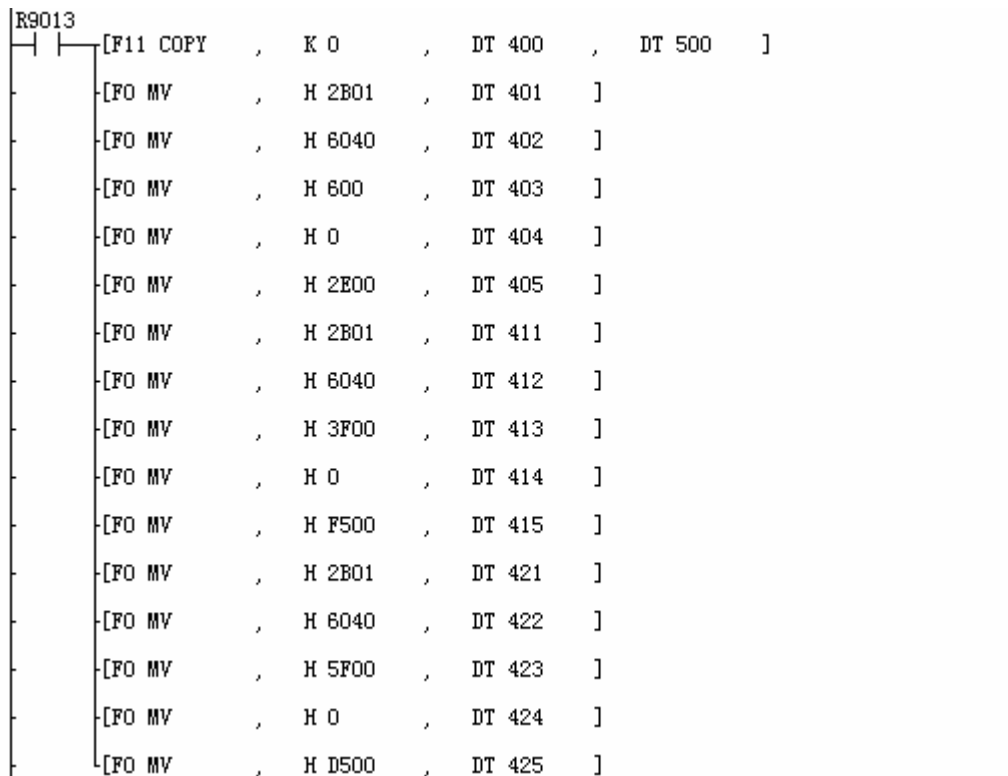
DT401-DT405 command: 01 2B 40 60 00 06 00 00 00 2E   motor off

DT411-DT415 command: 01 2B 40 60 00 3F 00 00 00 F5   control command 3F, relative positioning for motor

DT422-DT425 command: 01 2B 40 60 00 5F 00 00 00 D5   control command 5F, relative positioning for motor

Note: The stop command 01 2B 40 60 00 06 00 00 00 2E contains 10 bytes.

      01—ID of the ED drive

     2B-Write in

    6040-Control word

    06-Stop control word

    2E—Check code; for calculations of check code, refer to Section 11.1.2. The user can design a subprogram to automatically calculate the check code.

All above calculations are based on the hexadecimal system. For detailed control and status commands, refer to the object dictionary of the ED.

At last, the Panasonic PLC sends the serial command F144 to the ED drive.

```
X0
 ├──┤ ├── (DF)───────────────────────────────────→ 1
 ├1 ──→[F144 TRNS   ,   DT 400   ,   K 10   ]
```
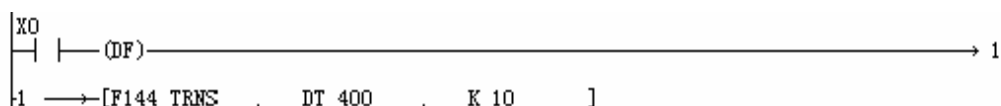
The above command will send the first 10 bytes of the DT401 to the serial port, and the DT400-DT405 is set to stop. When the command is sent out, the motor (ID=1) stops operation.

4. Communication between Mitsubishi FX2N PLC and ED through RS-485

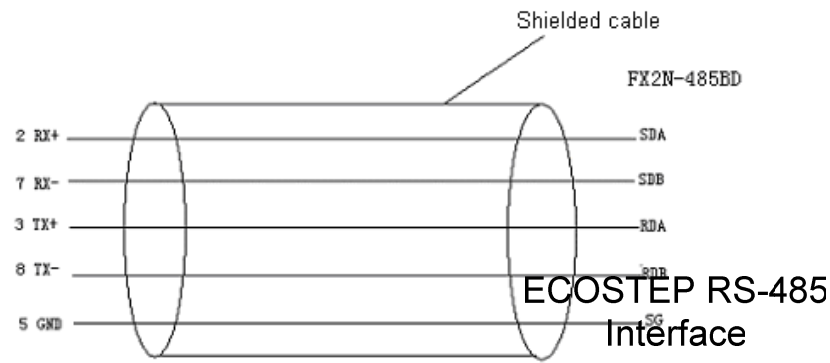   a). FX2N is connected to the RS-485 interface of the ED through the FX2N-485BD

ECOSTEP RS-485 Interface

Figure D4.1

b). Configuring communication parameters

Mitsubishi FX2N end

Set D8120=H0C81, no protocol, no start character, no end character, digital bits = 8, stop bits = 1, no parity check, baud rate = 9600, D8120 = H0C81 (b15..b12= 0000, b11..b8=1100, b7., b4=1000, b3..b0= 0001).

Bits	Definition		0	1
B0	Data length		0-7bits	1-8bits
B2,B1	Parity		00-no parity, 01-Odd parity, 11-Even parity	
B3	Stop bits		0- 1 bits	1- 2 bits
B4 B5 B6 B7	Baud rate-bps		1000- 9600bps 1001 –19200bps	
B8	Header character		0- none	1 - D8124 (designated, Default = STX , 02H)
B9	Terminator character		0- none	1 - D8125 (designated, Default = ETX , 03H)
B10 B11 B12	Communication Control (see timed diagrams page 10-20 onwards)		No protocol (b12, b11, b10) (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	FX-485 Network	Sum check	No check	Added automatically
B14		protocol	No protocol	Dedicated Protocol

B15		protocol	Format 1		Format 4

ED end: 2F910008, set N=63, that is, baud rate of 9600

c). Communication protocol

Adopt free communication protocol, and FX2N transfers data according to the ED commands.

Write: 10-byte (fixed) transmission protocol, ID as the ED address, to be set by DIP switch, 0 reserved

CHKS calculation: last two bits of the result (0-(byte0+ …..+byte 9))

byte0   byte 9

ID	8 Byte Date	CHKS

For example,   set the control mode to velocity mode 3

01 2B 60 60 00 03 00 00 00 11

01 – ID address, 2B – Write command, 606000 – change index of control mode, 11 - CHKS

Read: For example, read the 485 parameters of the ED with the ID of 1. By querying the ED object dictionary, the set address of the RS-485 communication parameter baud rate is 2F9100, and the read command is:

01 40 91 2F 00 00 00 00 00 FF

FF- check code = (0 – (byte0 + …..+byte 9))

40-Read command

d). Description of the example

The example 485RW.PMW is a Mitsubishi program.

X2- Power on

X3- Power off

X4- Set control mode to 3

X5 – Change velocity to 10 rev./m

X6- Reset

X7- Read the baud rate of RS-485 communication of the ED drive

For a detailed example, **refer to the user CD of the Kinco servo.**

Appendix E: Full Closed Loop Control

The full closed loop control is usually applicable to high-precision positioning system. Usually, a set of connection mechanism exists between the servo motor and the load, for example, linear guide, so the encoder signal fed back from the servo motor shaft cannot represent the displacement and velocity signal of the load. If the requirement of the system for positioning precision is not high, this application can be implemented by semi-closed loop control of the servo drive. However, if the requirement for positioning precision is high, for example, the precision to um level, the full closed loop control mode has to be adopted. As a result, it raises higher requirements for the mechanical structure design and assembly of the whole system. Under the conventional full closed loop control mode, the feedback signal on the load end is connected to the controller end, for example, high-end motion control card, and the Kinco servo drive adopts a kind of special firmware to implement the full closed loop control function.
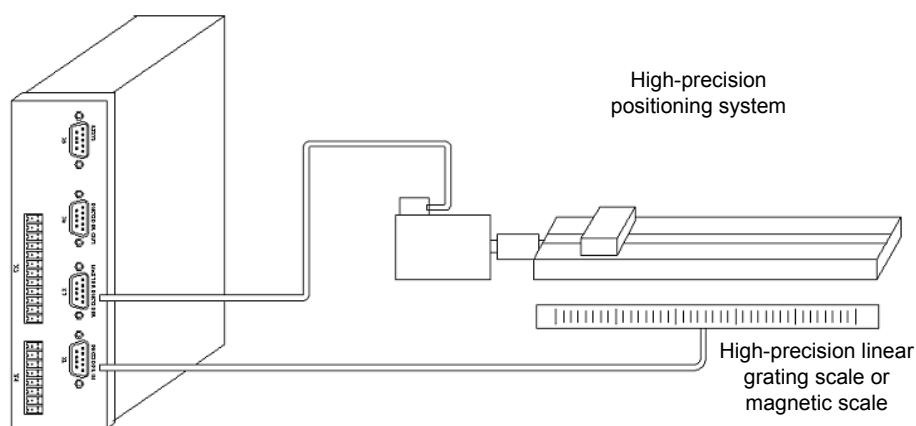

Figure E.1 Full Closed Loop Control

Wiring: The full range of Kinco servo drives support the full closed loop control. Different from non full closed loop control, the encoder cable of a servo motor is not directly connected to the X8 (Encoder in) interface, but to the X7 (Master encoder) interface. As shown in the above figure, the feedback signal on the load end is directly connected to the X8 (Encoder in) interface.

Firmware difference: 00010040.E10. The feedback from the linear encoder only works on the position loop and is applicable to the system conducted by screw rod and with high stiffness. For 00020040.E10, the feedback from the linear encoder works on the velocity loop and position loop and is applicable to the flexible system like synchronous belt.

Firmware importing: Use the Eco2flash software to import the full closed loop firmware directly. Save the configuration and reboot the system after importing.

Polarity setting: After electrical connection and firmware importing, adjust the polarity of the linear encoder in the ECO2WIN software first; otherwise, the control loop cannot work. Set the **607E00** object directly by **Direct object entry**. Bit6 (0x40) is used to adjust the relationship between the linear encoder and the motor encoder. First, set **607E00=0**, manually rotate the motor shaft counterclockwise, and observe whether the value of **position_actual_value** (**Actual position** in the **Device control-> Movement** interface) changes towards the positive counting direction; otherwise, set 607E00=0x40. After determining the value of bit6,

change the counting direction of **position_actual_value** by changing the value of bit7 (at this time, the value of bit6 cannot be changed).

PID parameter setting:

00010040.E10: Before setting other parameters, lower the values of ax_current ("parameter settings->current controller") and vc_kp ("parameter settings->velocity controller->p-gain velocity control"). Pay special attention to the proportional gain (vc_kp) of the velocity controller, which determines the resolution of the linear encoder. For example, if the resolution of the linear grating scale is 5 um, vc_kp=5.

00020040.E10:   Before setting other parameters, lower the value of max_current. Meanwhile, pay attention to the setting of vc_kp, which depends on the resolution of the motor encoder. The proportional gain of the position controller is affected by the value of [linear encoder resolution] / [motor encoder].   For example,   In a 100mm/revolution screw rod system, if the motor encoder resolution =10000um/8000=1.25um, when using a linear grating scale with a resolution of 1um, the proportion = 1.25um/1um=1.25. The proportional gain of the velocity controller shall be set according to this value.

Appendix F: Importing and Exporting Data by ECO2LOAD

We provide special software ECO2LOAD for data import and export in a Kinco servo drive. The interface is shown as follows:
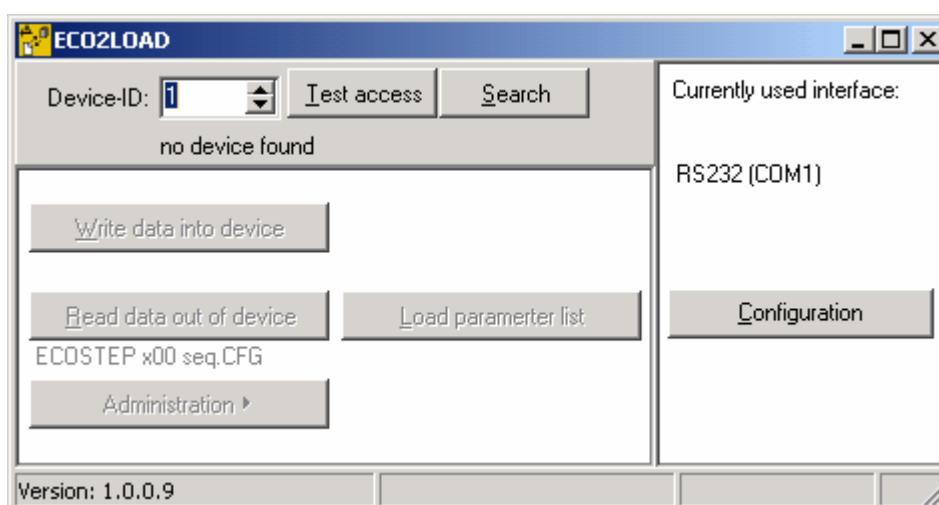

Figure F.1

The data in the **Device-ID** box is the ID of the default drive. The message "no device found" is displayed below. To search a device, click the <**Test access**> to judge whether the PC is connected normally to a drive. If a drive is found, the message "device found" will be displayed below the box. If no such a message appears, click <**Search**> for automatic search until a drive is found. The software supports RS-232, CANOPEN and USB interface. Each kind of communication parameters can be set separately. Click <**Configuration**> to set them in the window appeared, as shown in Figure F.2. When a drive is found normally, the interface is shown in Figure F.3.
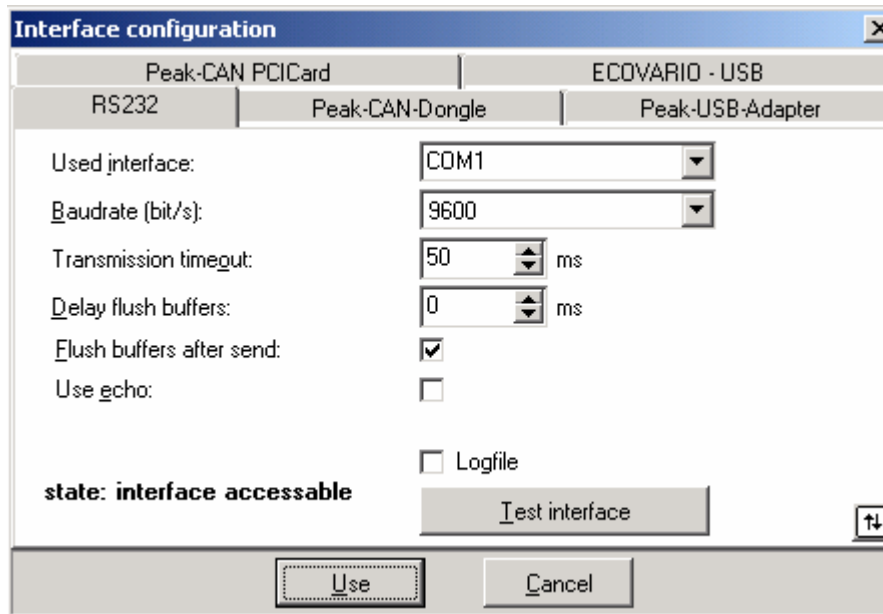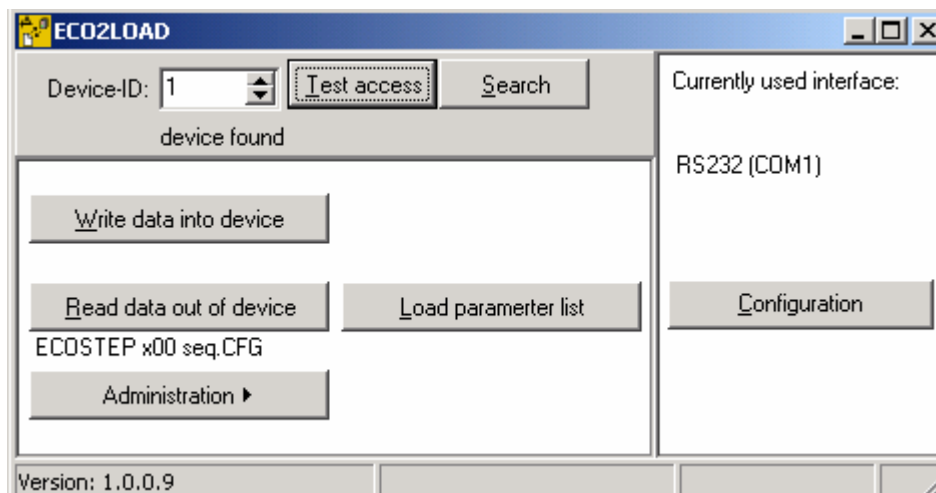
Figure F.2

Figure F.3

After the communication is normally established, the user can perform data import and export operations. Firstly, select the data range for import or export operation, that is, the segment of data for import or export. This software supports reading and writing operation on five different parts of parameters, namely, "comm", "seq", "dev", "joy" and "all". They stand for communication parameters, sequences, device configuration, joystick configuration, and all parameters respectively. Click <**Load parameter list**>, and select different configuration files. For example, to perform "seq" operation, select the ECOSTEP x00 seq.CFG file. Upon completion of these operations, use the **Read data out of device** option to read drive data. The **Write data into device** option is to write the previously exported data into the drive again.

In addition to data import and export function, the software also supports the functions of rebooting (boot), saving, and initializing parameters. To perform these operations, click <**Administration**>.