Content

Chapter 1 Product Acceptance & Model Description

1.1 Product Acceptance

1.1.1 Items for Acceptance (Wires Included)

Table 1-1 Product acceptance

Item for Acceptance	Remark
Whether the model of a delivered FD2S series servo system is consistent with the specified model	Check the nameplate of a servo motor and that of a servo driver
Whether the accessories included in the packing list are complete	Check the packing list
Whether any breakage occurs	Check the external appearance completely for any losses that are caused by transportation
Whether any screws are loose	Check for loose screws with a screwdriver
Whether the motor wiring is correct	Purchase motor accessory packages if no wirings are purchased

1.1.2 Nameplate of Servo Driver

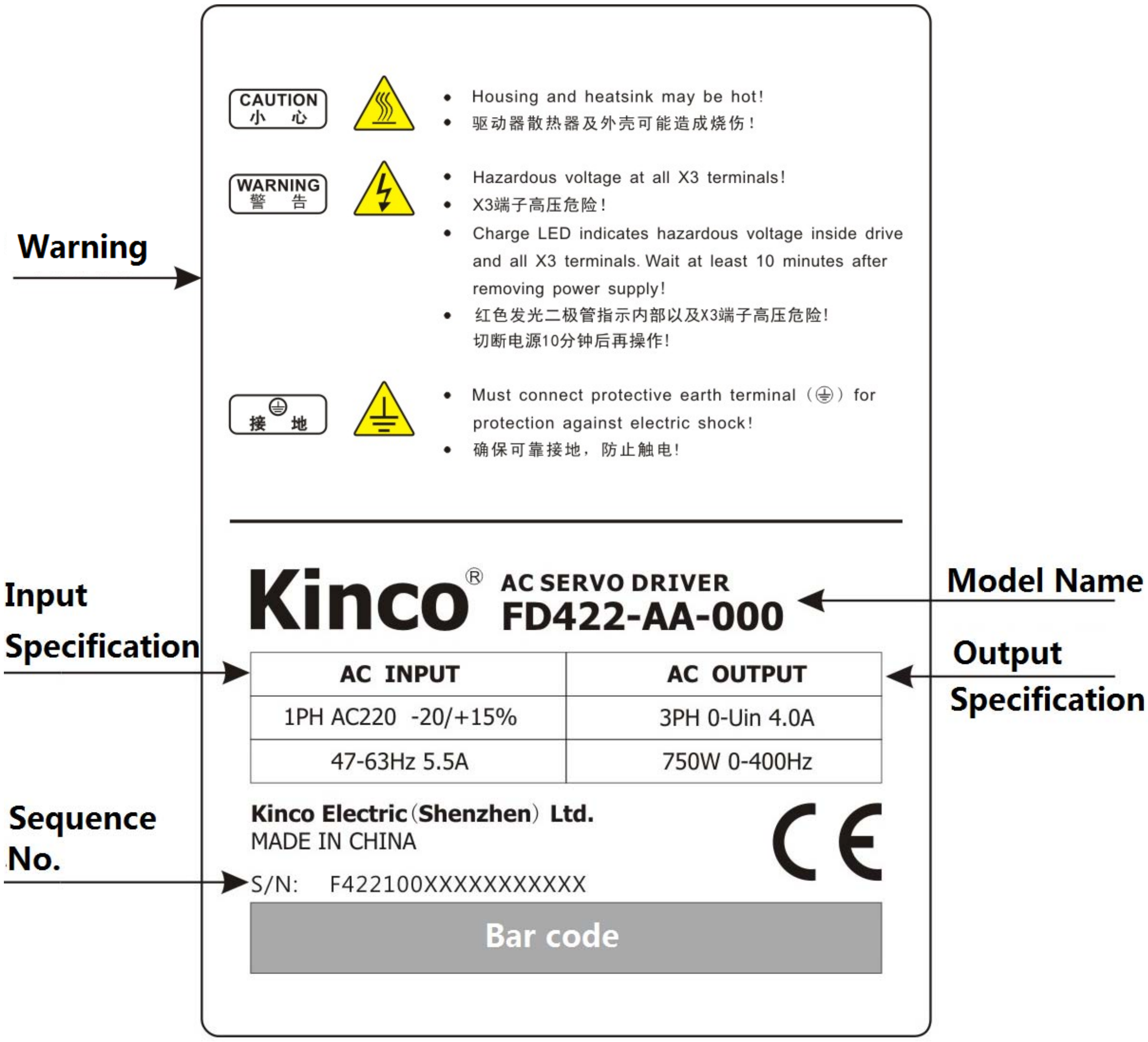


Fig. 1-1 Nameplate of a servo driver

1.1.3 Nameplate of Servo Motor

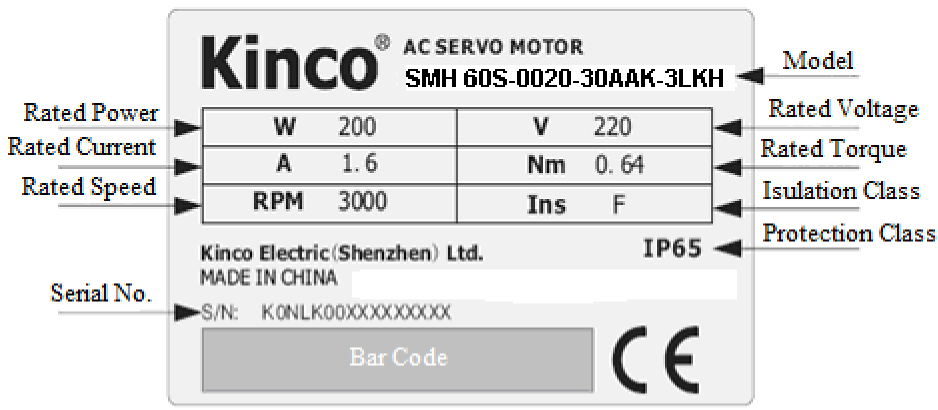


Fig. 1-2 Nameplate of a servo motor

1.2 Component Names

1.2.1 Component Names of FD2S Series Servo Driver

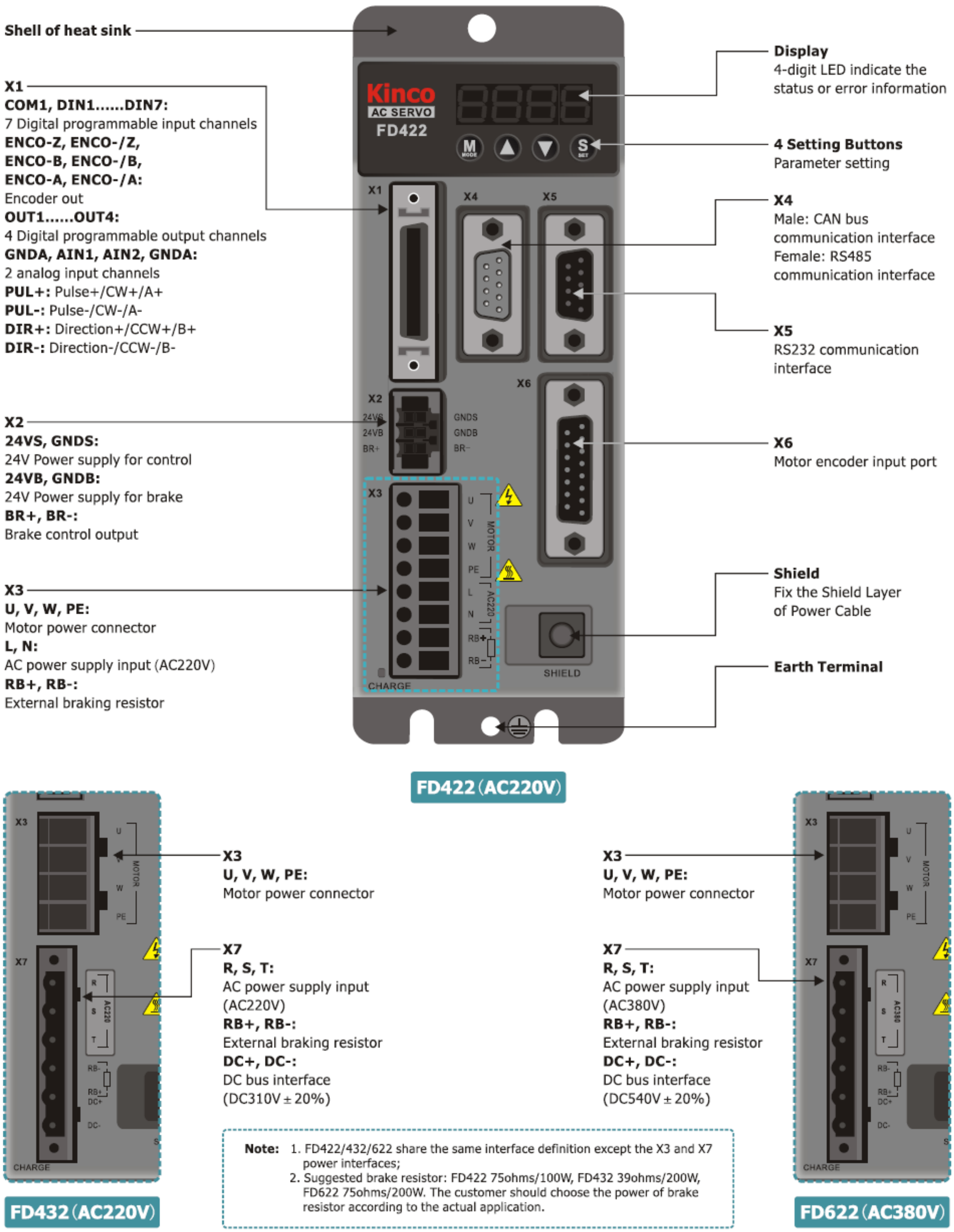


Fig. 1-3 Component Names of FD2S Series Servo Driver

1.2.2 Component Names of Servo Motor

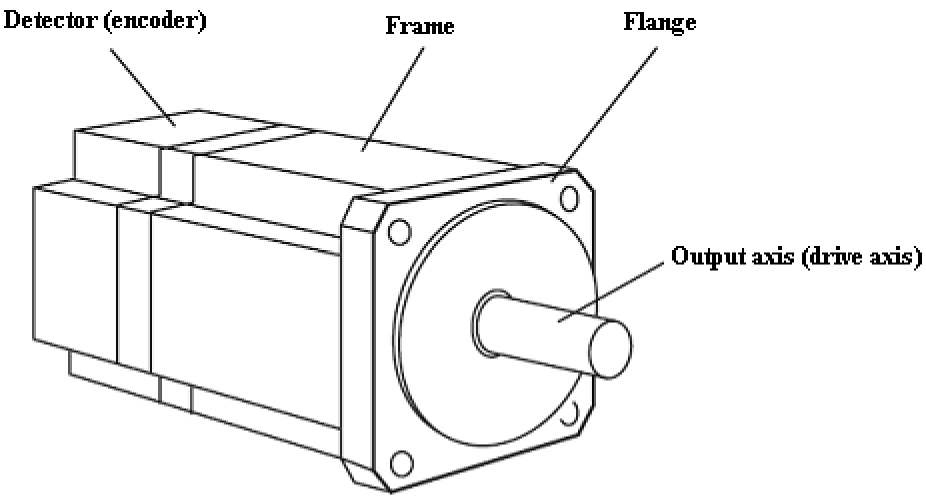


Fig. 1-4 Component names of a servo motor (brakes excluded)

1.3 Model Description of Servo Motors and Drivers

1.3.1 Servo Drivers

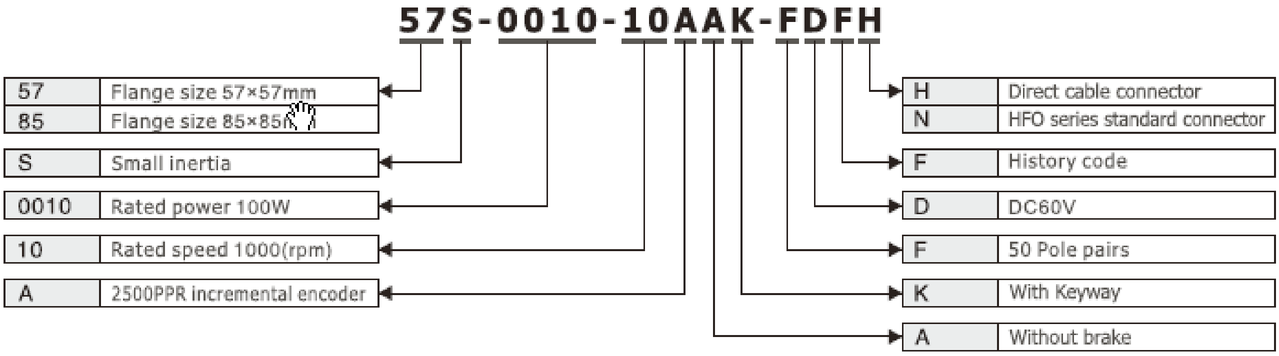
FD422-AA-000 Note: Users configure motor for this driver themselves

FD	FD series
1	Input voltage DC24V~70V
4	Input voltage AC220V
6	Input voltage AC380V
2	Driver size
2	FD generation 2
LA	RS232, RS485
AA/CA	RS232, CAN
000	Software version

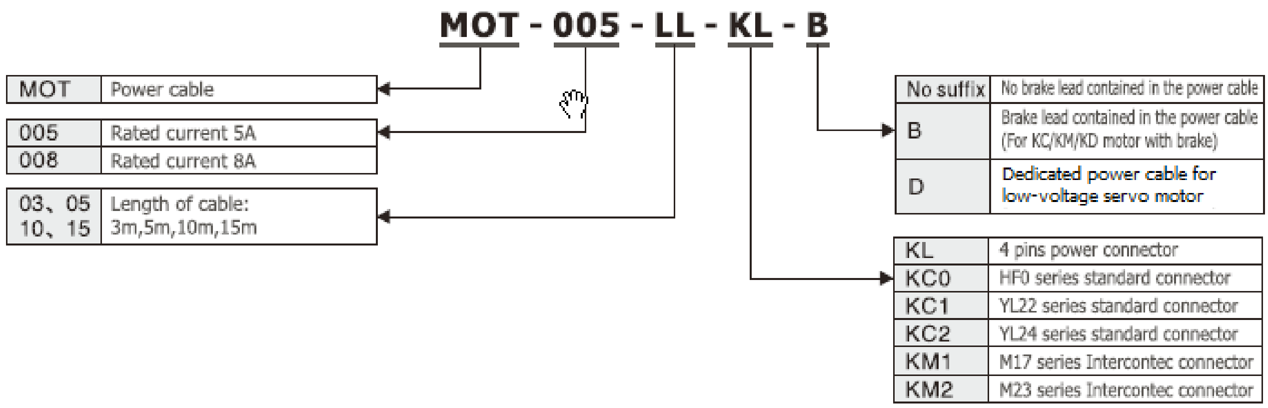
1.3.2 Servo Motors

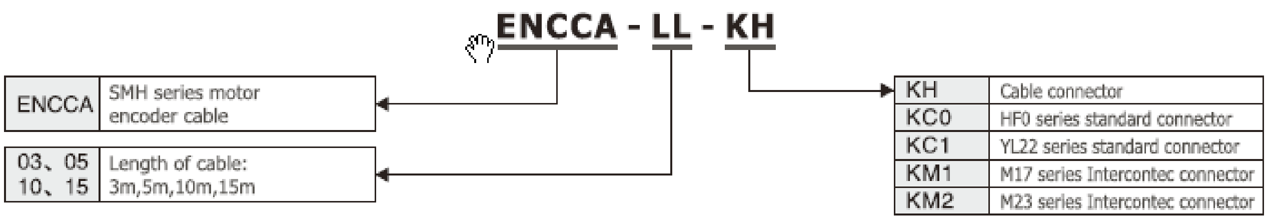
SMH 60S-0040-30AAK-3LKH

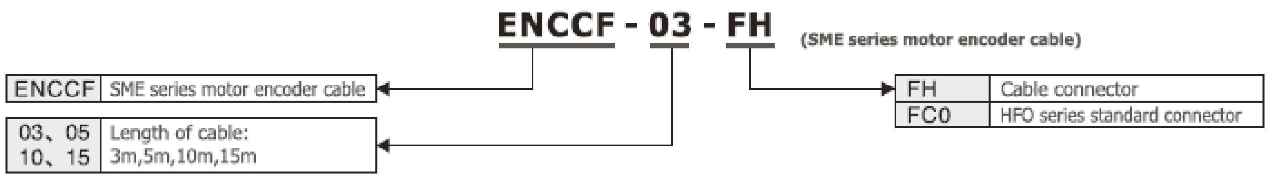
SMH	SMH series
60	Flange size 60X60mm
S	Small inertia
D	Medium inertia
H	High inertia
0040	Rated power 40 x 10(W)
30	Rated speed 30 x 100(rpm)
A	2500PPR incremental encoder
A	No brake
B	With brake
K	With keyway
A	No keyway
F	50 pole pairs
3	3 pole pairs
4	4 pole pairs
D	DC70V
L	AC220V
H	AC380V
K	Historical code
H	Direct cable connector
N	HFO series standard connector
C	YL series standard connector
M	2xM17 series Intercontec connector
E	2xM23 series Intercontec connector
D	M17+M23(Power M23 Intercontec connector, Encoder M17 Intercontec connector)

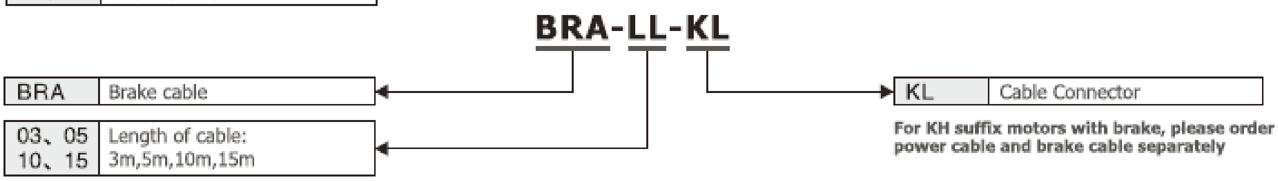


1.3.3 Power, Brake and Encoder cable of Motors









Chapter 2 Precautions and Installation Requirements

2.1 Precautions

- Tightly fasten the screws that fix the motor;
- Make sure to tightly fasten all fixed points when fixing the driver;
- Do not tighten the cables between the driver and the motor/encoder;
- Use a coupling shaft or expansion sleeve to ensure that both the motor shaft and equipment shaft are properly centered;
- Do not mix conductive materials (such as screws and metal filings) or combustible materials (such as oil) into the servo driver;
- Avoid the servo driver and servo motor from dropping or striking because they are precision equipment;
- For safety, do not use any damaged servo driver or any driver with damaged parts.

2.2 Environmental Conditions

Table 2-1 Environmental conditions

Environment	Condition
Temperature	Operating temperature: 0°C - 40°C (ice free) Storage temperature: - 10°C - 70°C (ice free)
Humidity	Operating humidity:5~ 90% RH (non-condensing) Storage humidity: 5~90% RH (non-condensing)
Air	Indoor (No direct sunlight), no corrosive gas or combustible gas No oil vapor or dust
Height	Below 2000 m above the sea level,it needs power derating after 1000m
Vibration	5.9 m/s2

2.3 Mounting Direction & Spacing

Please install the servo driver correctly according to following figure,or it will cause faults.
The servo driver should be vertically installed on wall.Take fully into account heat dissipation when using any heating components (such as braking resistors) so that the servo driver is not affected.

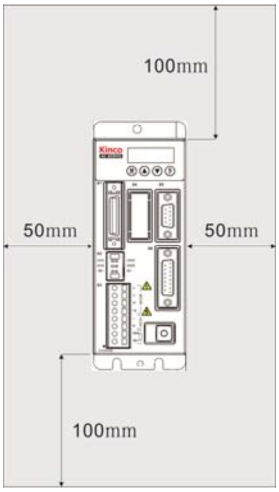


Fig. 2-1 Installing a servo driver

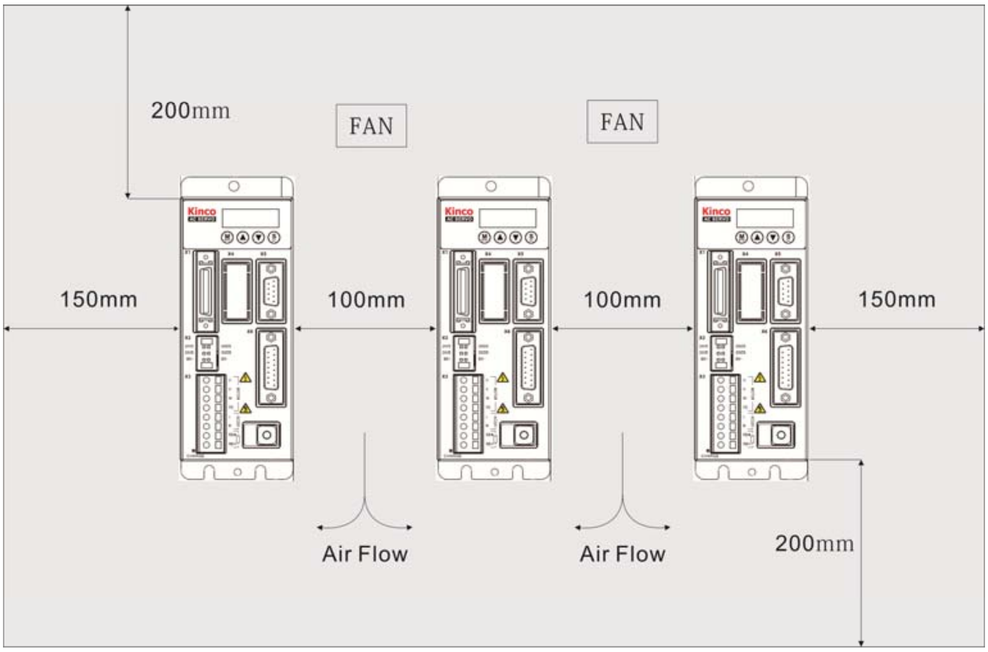


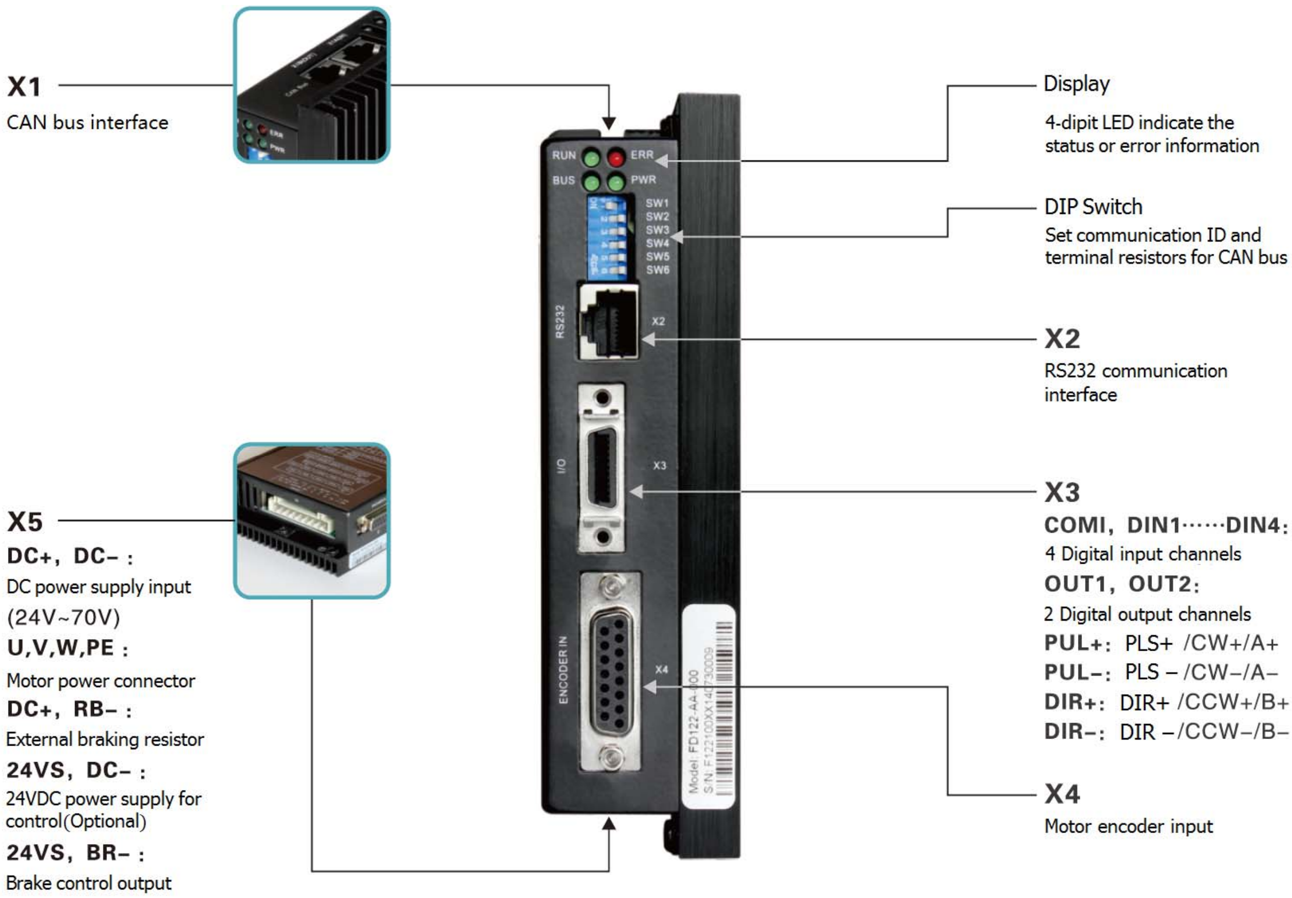
Fig. 2-2 Installing multiple servo drivers

Fig. 2-3 Mounting direction

Chapter 3 Interfaces and Wirings of FD2S Driver

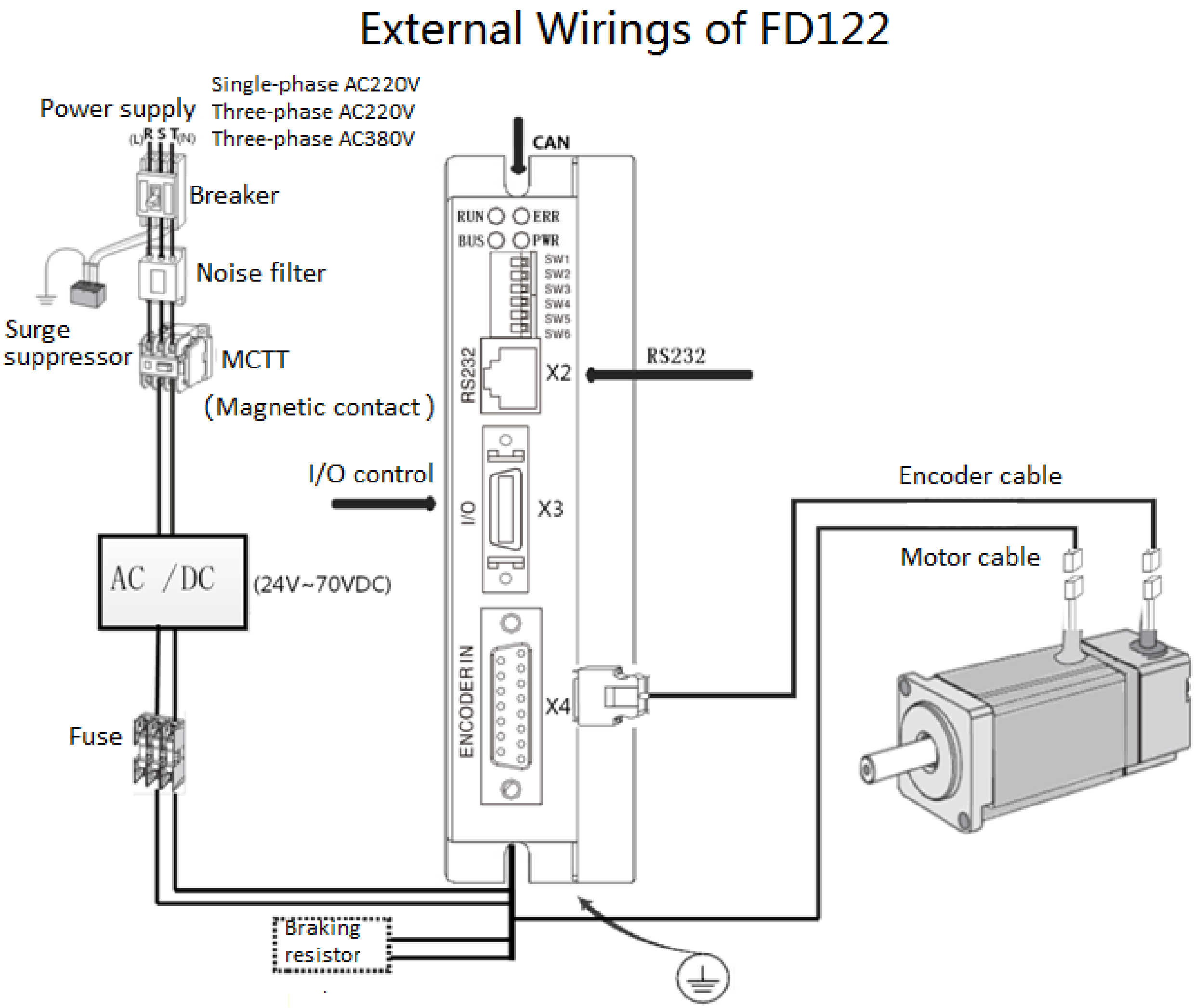
3.1 Interface and wiring of FD122

3.1.1 Panel and Interfaces Description of FD122

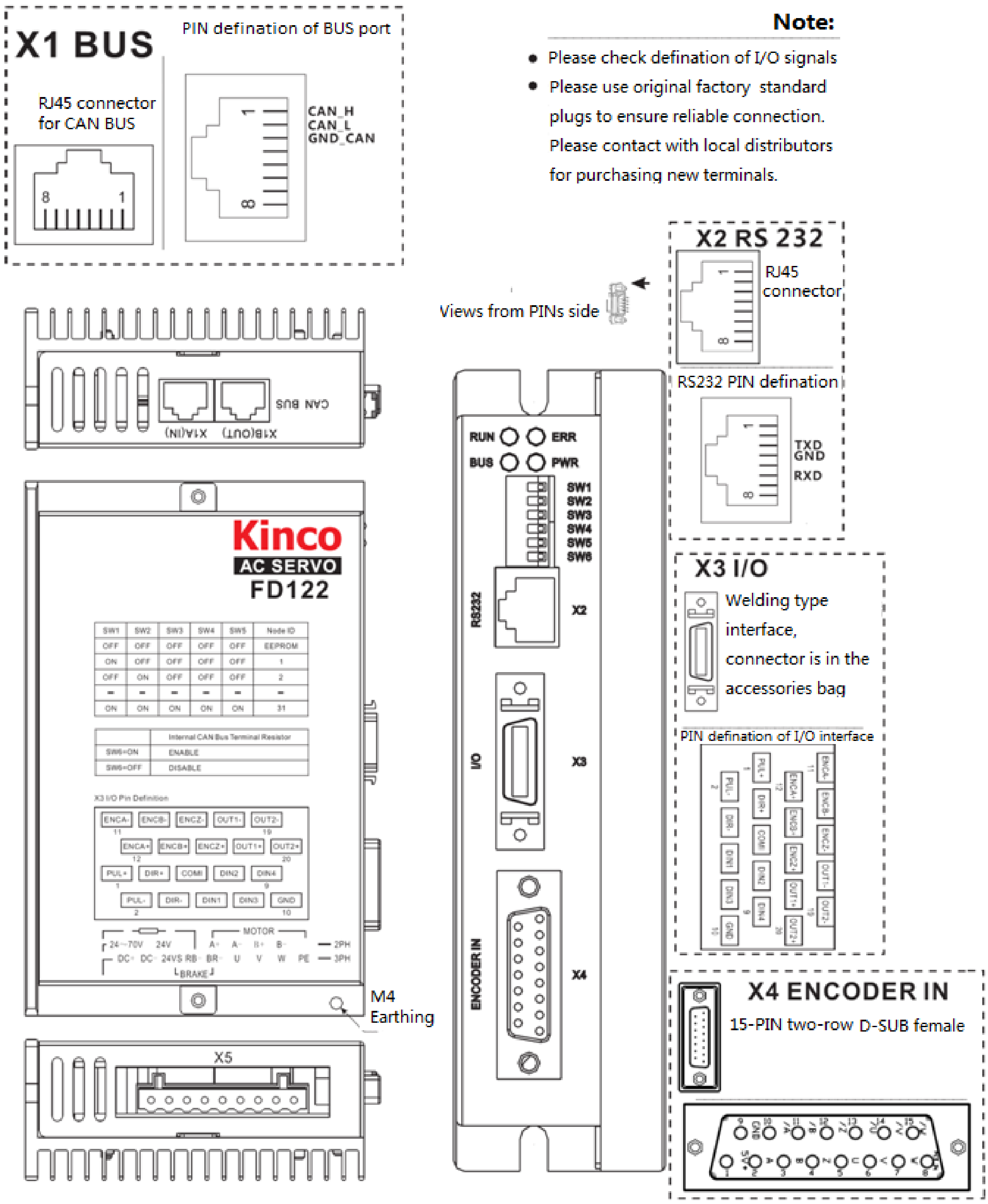


Interface	Driver	Function	Description
X1	FD122	CAN	CAN bus interface
X2		RS232	RS232 interface
X3		I/O	I/O port
X4		Encoder input	Motor encoder input interface
X5		Motor and power supply interface	24V~70VDC power supply, motor power, brake power supply, brake resistor interface

3.1.2 External Wiring of FD122



3.1.3 Interface Wiring Defination of FD122



3.1.3.1 CAN Bus Interface(X1)

Fig. 3-1 CAN Bus interface PINs defination

No.	Name	Function
1	CAN_H	CAN bus high
2	CAN_L	CAN bus low

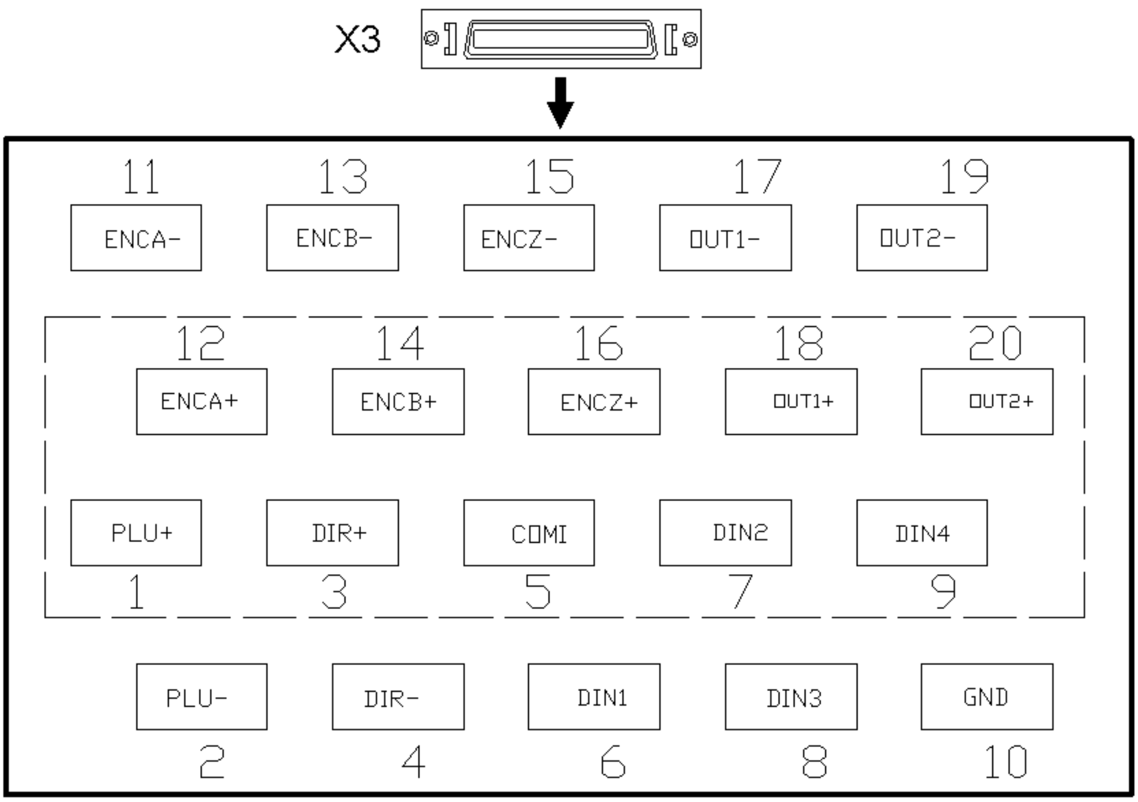
3	GND	Signal ground
Others	NC	Undefined

3.1.3.2 Communication Interface(X2)

Fig. 3-2 RS232 communication interface PINs defination

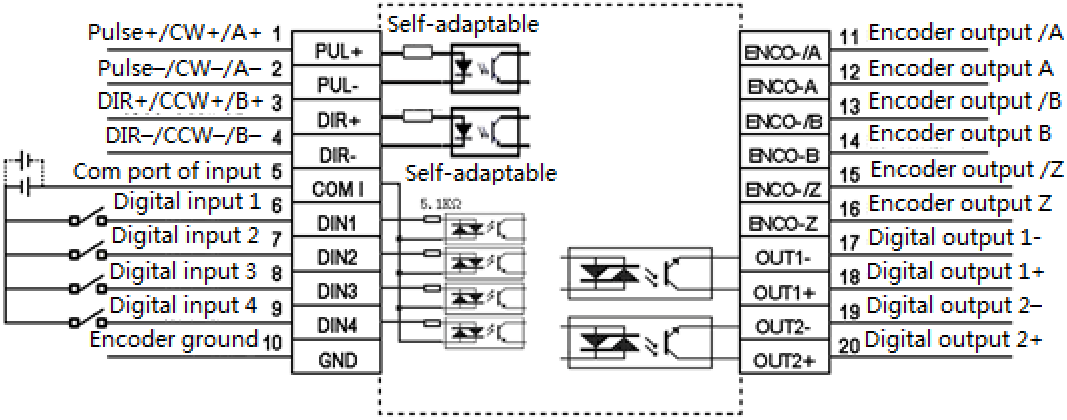
No.	Name	Function
3	TXD	Send data
4	GND	Signal ground
6	RXD	Receive data
Others	NC	Undefined

3.1.3.3 I/O Interface(X3)



Name	Function	Name	Function
COMI	Common port of digital input	PUL+/PUL-	Pulse input
DIN1~DIN4	Digtal input	DIR+/DIR-	Direction input
OUT1+/OUT1-	Digital output	ENCO-Z/ENCO-/Z	Encoder signal

OUT2+/OUT2-		ENCO-B/ENCO-/B ENCO-A/ENCO-/A	output
GND	Digital signal ground		



3.1.3.4 Encoder Input Interface（X4）

Fig. 3-3 Encoder input interface PINs defination

No.	Name	Function
1	5V+	5V output
2	A	A phase of encoder input
3	B	B phase of encoder input
4	Z	Z phase of encoder input
5	U	U phase of encoder input
6	V	V phase of encoder input
7	W	W phase of encoder input
8	PTC_IN	Undefined
9	GND	Ground of encoder signal
10	/A	A phase of encoder input
11	/B	B phase of encoder input
12	/Z	Z phase of encoder input
13	/U	U phase of encoder input
14	/V	V phase of encoder input
15	/W	W phase of encoder input

3.1.3.5 Motor/Power Supply Interface (X5)

Fig. 3-4 Motor power supply interface

PIN name	PIN function
DC+	Positive terminal of DC power supply and braking resistor
DC-	Negtive terminal of DC power supply and 24VDC power supply
24VS	Positive terminal of 24VDC power supply and braking
RB-	Negtive terminal of braking resistor
BR-	Negtive terminal of braking, A- phase of motor output
U	U phase of motor output, A- phase of motor output
V	V phase of motor output, B+ phase of motor output
W	W phase of motor output, B- phase of motor output
PE	Motor earthing

3.2 Interface and wiring of FD322S/FD332S/FD412S/FD422S/FD432S/FD622S

3.2.1 Interface Description

Table 3-1 Interfaces of FD322S/FD332S/FD412S/FD422S/FD432S/FD622S

<table>
<tr><th>Interface</th><th>Driver</th><th>Symbol</th><th colspan="2">Function</th></tr>
<tr><td rowspan="24">X1</td><td rowspan="24">FD322S
FD332S
FD412S
FD422S
FD432S
FD622S</td><td>COMI</td><td colspan="2">Common terminal of digital inputs</td></tr>
<tr><td>DIN1~DIN7</td><td colspan="2">Digital inputs. Valid signal:12.5V~24V.Invalid signal:<5V</td></tr>
<tr><td>OUT1+</td><td colspan="2">Digital output 1+</td></tr>
<tr><td>OUT1-</td><td colspan="2">Digital output 1-</td></tr>
<tr><td>OUT2+</td><td colspan="2">Digital output 2+</td></tr>
<tr><td>OUT2-</td><td colspan="2">Digital output 2-</td></tr>
<tr><td>OUT3</td><td colspan="2">Digital output 3</td></tr>
<tr><td>OUT4</td><td colspan="2">Digital output 4</td></tr>
<tr><td>COMO</td><td colspan="2">Common terminal of digital outputs</td></tr>
<tr><td>GND</td><td colspan="2">Ground signal</td></tr>
<tr><td>ENCO-Z</td><td colspan="2" rowspan="6">Motor encoder output interface</td></tr>
<tr><td>ENCO-/Z</td></tr>
<tr><td>ENCO-B</td></tr>
<tr><td>ENCO-/B</td></tr>
<tr><td>ENCO-A</td></tr>
<tr><td>ENCO-/A</td></tr>
<tr><td>AIN1</td><td colspan="2">Analog signal input 1. Input impedance: 200 K</td></tr>
<tr><td>GNDA</td><td colspan="2">Ground signal of analog</td></tr>
<tr><td>AIN2</td><td colspan="2">Analog signal input 2. Input impedance: 200 K</td></tr>
<tr><td>GNDA</td><td colspan="2">Ground signal of analog</td></tr>
<tr><td>PUL+</td><td>Pulse or positive pulse interface (+)</td><td rowspan="4">Input voltage range: 5V~24V</td></tr>
<tr><td>PUL-</td><td>Pulse or positive pulse interface (-)</td></tr>
<tr><td>DIR+</td><td>Direction or negative pulse interface (+)</td></tr>
<tr><td>DIR-</td><td>Direction or negative</td></tr>
</table>

			pulse interface (-)
X2		24VS/GNDS	Logic power supply:24 V ± 15%, >0.5A
		24VB/GNDB	Power supply for brake ,DC18~30V 2A
		BR+/BR-	Brake interface
X3	FD322S FD412S FD422S	U/V/W/PE	Motor cable interface
		L/N	Main power supply (Single-phase AC220V)
		RB+/RB-	Braking resistor interface
	FD332S FD432S FD622S	U/V/W/PE	Motor cable interface
X4	FD322S FD332S FD412S FD422S FD432S FD622S	BUS	RS485 or CAN interface
X5	FD322S FD332S FD412S FD422S FD432S FD622S	RS232	RS232 interface
X6		ENCODER IN	Encoder cable interface
X7	FD332S FD432S FD622S	R/S/T	Main power supply (CD432S: Single phase or 3-phase AC220V, CD622S: 3-phase AC380V)
		RB+/RB-	Braking resistor interface
		DC+/DC-	DC bus power supply(Cannot use together with R/S/T)

3.2.2 External Wirings

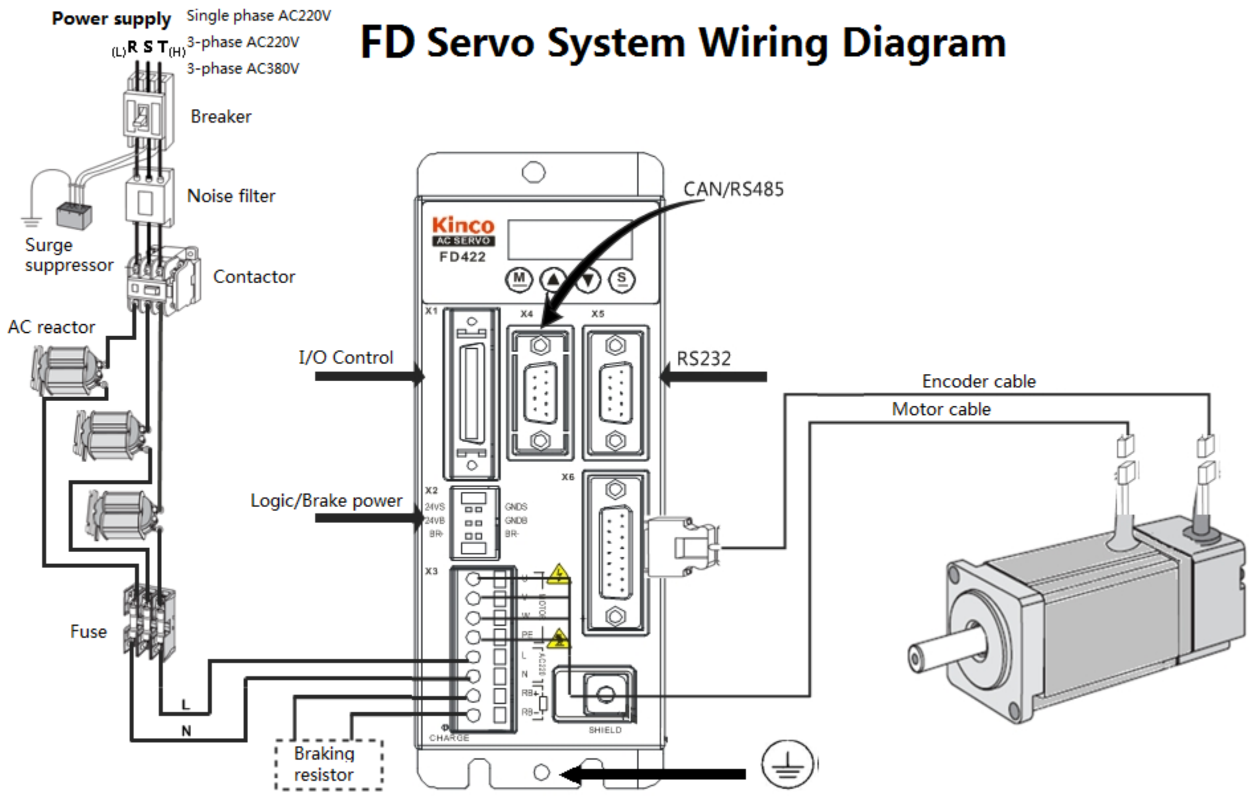
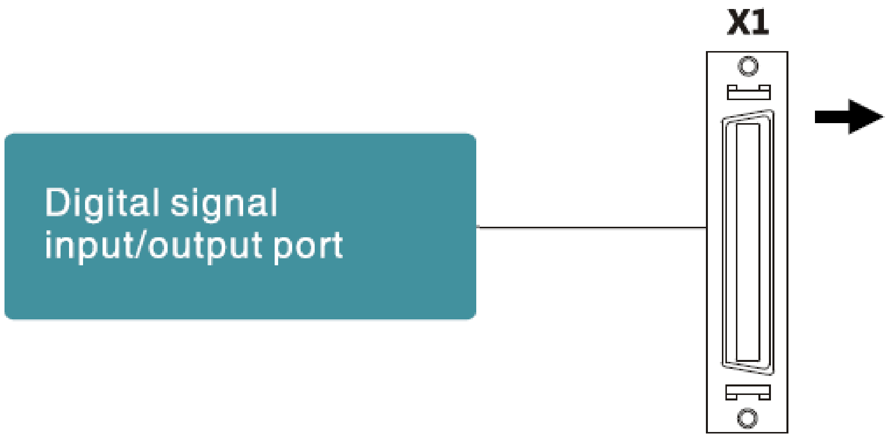
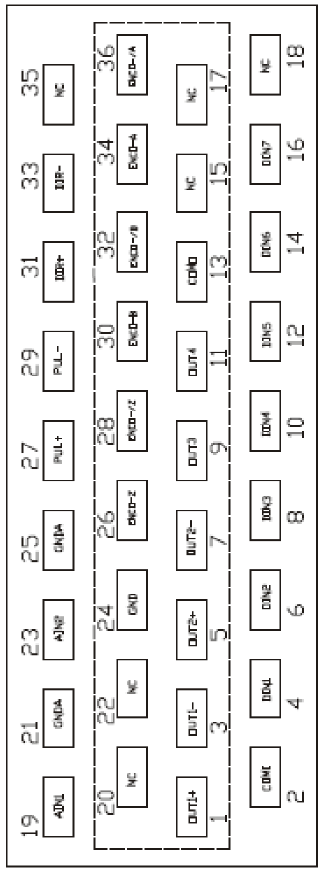


Fig. 3-1 External wirings diagram of FD2S drive

3.2.3 I/O Interface





X1

Pin No.	Signal	Pin No.	Signal	Pin No.	Signal	Pin No.	Signal
2	COM1	1	OUT1+	20	NC	19	AIN1
4	DIN1	3	OUT1–	22	NC	21	GNDA
6	DIN2	5	OUT2+	24	GND	23	AIN2
8	DIN3	7	OUT2–	26	ENCO–Z	25	GNDA
10	DIN4	9	OUT3	28	ENCO–/Z	27	PUL+
12	DIN5	11	OUT4	30	ENCO–B	29	PUL–
14	DIN6	13	COM0	32	ENCO–/B	31	DIR+
16	DIN7	15	NC	34	ENCO–A	33	DIR–
18	NC	17	NC	36	ENCO–/A	35	NC

Fig. 3-2 I/O interface of FD2S driver

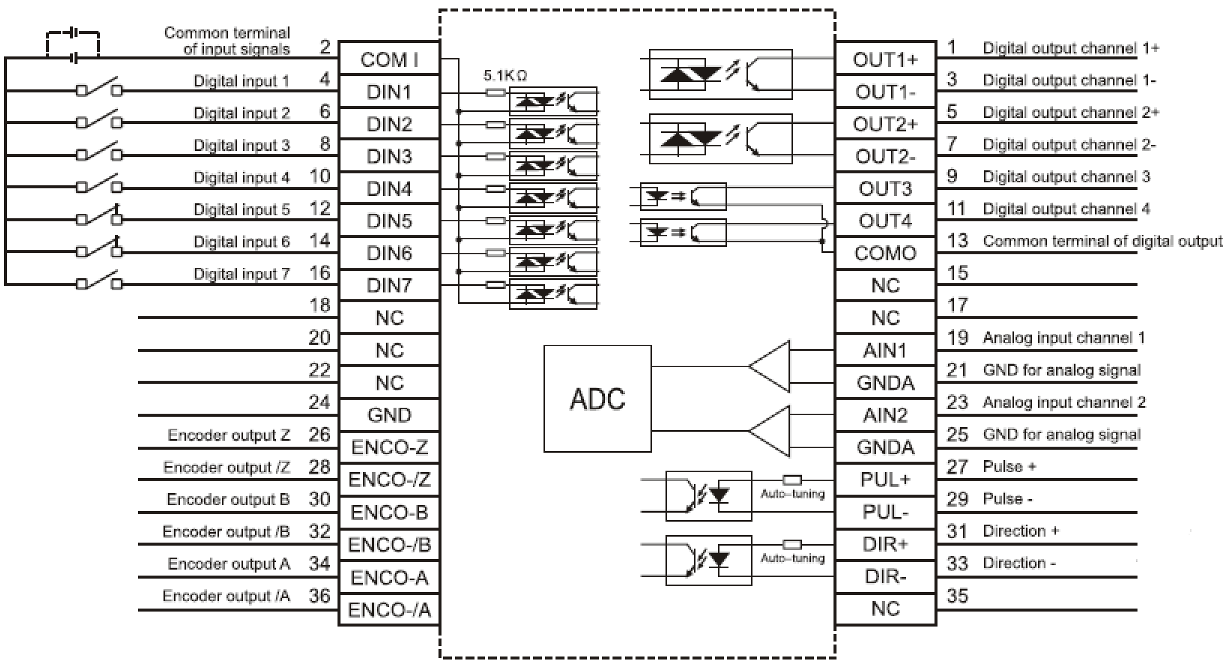
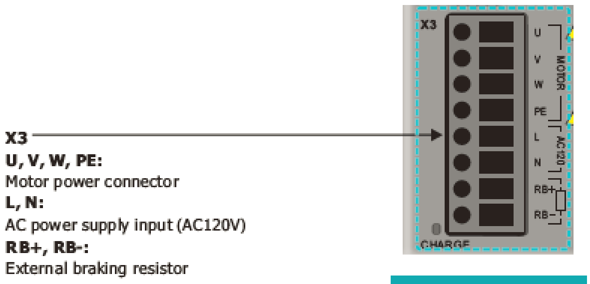
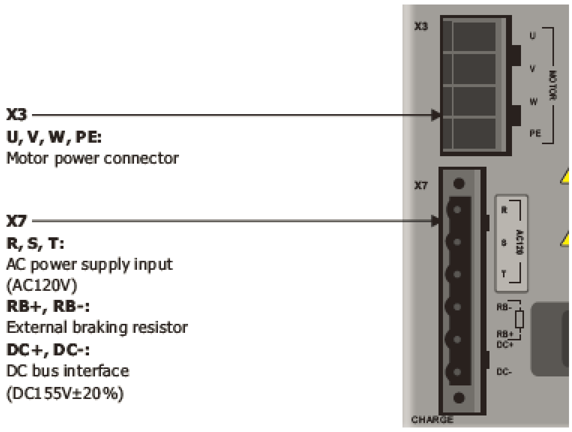


Fig. 3-3 Wirings of the I/O interface of FD2S driver

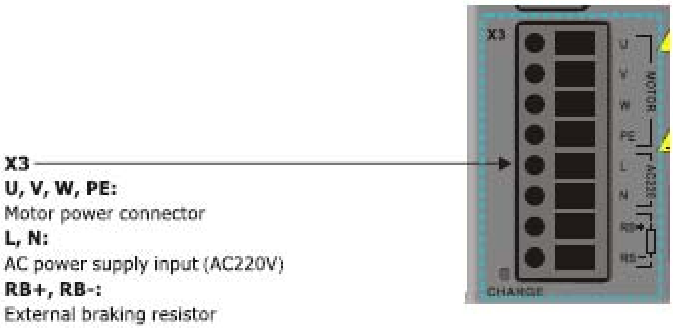
3.2.4 Power Interface of FD2S Driver (FD322SFD412S/FD422S/X3, FD332S/FD432S/FD622S/X3 and X7)



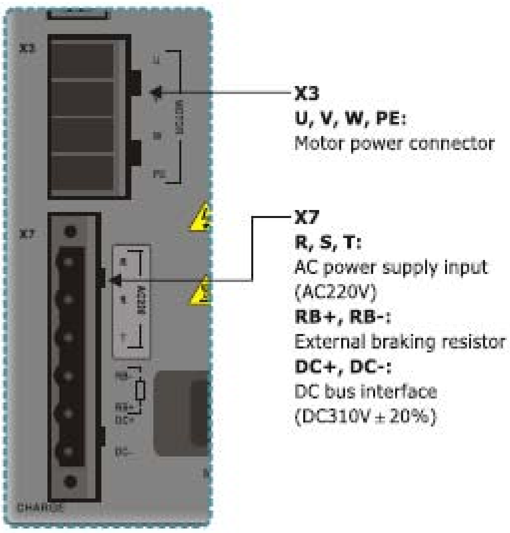
FD322S (AC120V)



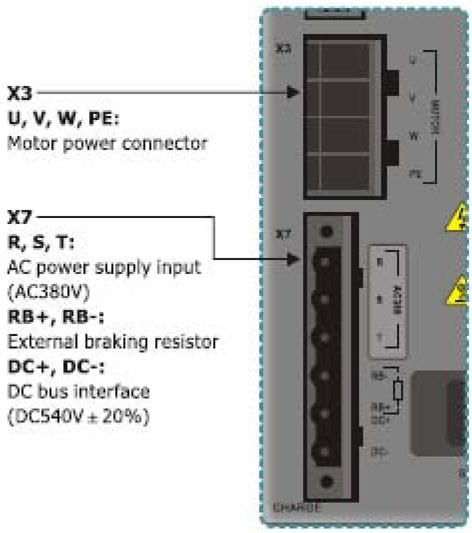
FD332S(AC120V)



FD422 (AC220V)



FD432 (AC220V)



FD622 (AC380V)

3.2.5 X4~X6 Interface

X4~X6 interface of FD2S driver use D-SUB connector.The styles of different D-SUB connectors are shown in following figure.

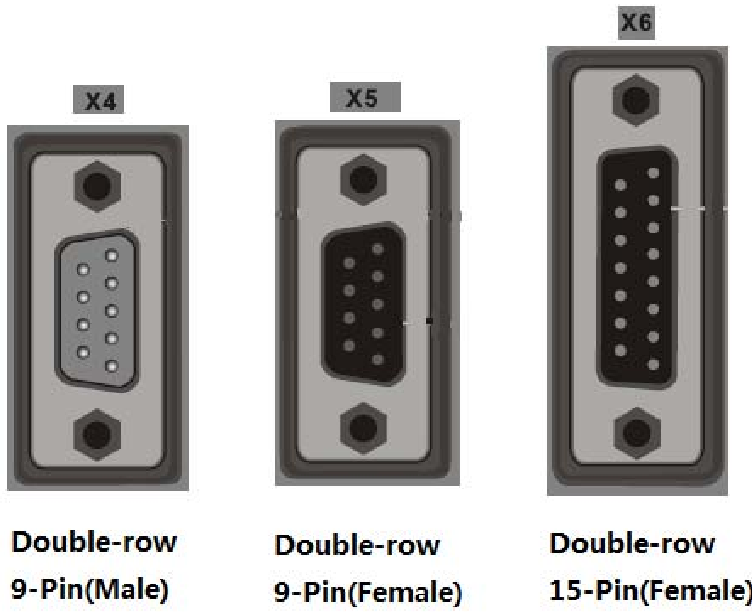


Fig.3-6 D-SUB connector diagram of driver

3.2.5.1 X4 Interface(RS485/CAN)

RS485:

Name	Pin	Signal	Descriptions	Function
RS485（9-Pin female）	1	NC	N/A	RS485 interface
	5	GND	Signal ground	
	6	+5V	Power	
	2	RX	Receive data	
	7	/RX		
	3	TX	Send data	
	8	/TX		
	4	NC	N/A	
	9	NC		

CAN:

Name	Pin	Signal	Descriptions	Function
CAN（9-Pin male）	1	NC		CAN bus interface
	5	NC		
	6	NC		
	2	CAN_L	CAN_L	
	7	CAN_H	CAN_H	
	3	GND	Signal ground	

	8	NC		
	4	NC		
	9	NC		

3.2.5.2 X5 Interface(RS232)

Name	Pin	Signal	Descriptions	Function
RS232（9-Pin female）	1	NC	N/A	RS232 interface
	2	TX	Send data	
	3	RX	Receive data	
	4	NC	N/A	
	5	GND	Signal ground	
	6	NC	N/A	
	7	NC		
	8	NC	N/A	
	9	NC		

3.2.5.3 X6 Interface (Encoder in)

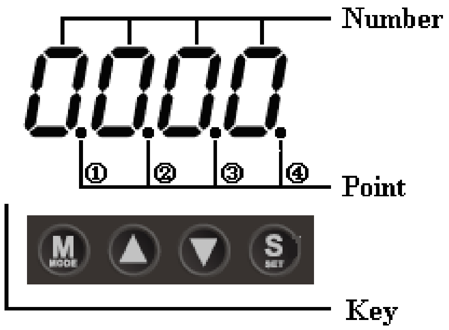
Name	Pin	Signal	Descriptions	Function
Encoder in（Double rows 15-Pin female）	1	+5V	5V output	Motor encoder input
	9	GND	0V	
	8	PTC_IN	PTC of motor input	
	2	A	A phase of encoder input	
	10	/A		
	3	B	B phase of encoder input	
	11	/B		
	4	Z	Z phase of encoder input	
	12	/Z		
	5	U	U phase of encoder input	
	13	/U		
	6	V	V phase of encoder input	
	14	/V		
	7	W	W phase of encoder input	
	15	/W		

Chapter 4 Digital Operation Panel

4.1 Introduction

A digital operation panel functions to set user parameters in a servo driver, execute instructions, or display parameters. Table 4-1 describes all display contents and functions of the digital operation panel.

Table 4-1 Display contents and functions of a digital operation panel



Number/ Point/Key	Function
①	Indicates whether data is positive or negative. If it is on, it indicates negative; otherwise it indicates positive.
②	Distinguishes the current object group and the address data in this object group during parameter settings. Indicates the higher 16 bits of the current 32-bit data when internal 32-bit data is displayed in real time. Indicates the earliest error when history records of errors (F007) are displayed.
③	Indicates a data display format when parameters are displayed and adjusted in real time. If it is on, it indicates the data is displayed in hexadecimal; otherwise it indicates the data is displayed in decimal. Indicates the latest error when the history records of errors (F007) are displayed.
④	If it is on, it indicates that internal data is currently displayed. If it flickers, it indicates that the power part of the driver is in the working status.
MODE	Switches basic menus. During the adjustment of parameters, short presses the key to move the bit to be adjusted, and long presses the key to return to the previous state.
▲	Presses ▲ to increase set values; long presses ▲ to increase numbers promptly.
▼	Presses ▼ to decrease set values; long presses ▼ to decrease numbers promptly.
SET	Enters the selected menu by pressing this key. Keeps current parameters in the enabled status. Confirms input parameters after parameters are set. Long presses this key to switch to higher/lower 16 bits when internal 32-bit data is displayed in real time.
P..L	Activates position positive limit signals.

n..L	Activates position negative limit signals.
Pn.L	Activates position positive/negative limit signals.
Overall Flicking	Indicates that an error occurs on the driver, and is in the alarm state.

If the parameter adjusting display mode is featured by the decimal system:
When the units place is flickering, press ▲ to add 1 to the current value; press ▼ to deduct 1 from the current value. When the tens place is flickering, press ▲ to add 10 to the current value; press ▼ to deduct 10 from the current value. When the hundreds place is flickering, press ▲ to add 100 to the current value; press ▼ to deduct 100 from the current value. When the thousands place is flickering, press ▲ to add 1000 to the current value; press ▼ to deduct 1000 from the current value.
If the parameter adjusting display mode is featured by the hexadecimal system:
When the units place is flickering, press ▲ to add 1 to the current value; press ▼ to deduct 1 from the current value. When the tens place is flickering, press ▲ to add 0X10 to the current value; press ▼ to deduct 0X10 from the current value. When the hundreds place is flickering, press ▲ to add 0X100 to the current value; press ▼ to deduct 0X100 from the current value. When the thousands place is flickering, press ▲ to add 0X1000 to the current value; press ▼ to deduct 0X1000 from the current value.
When adjusting decimal parameters, the display mode is automatically switched to the hexadecimal system if the data is greater than 9999 or less than -9999. In this case, the 3rd decimal point from left to right is highlighted.

4.2 Operation on Digital Operation Panel

Figure 4-1 Operation on a digital operation panel

Note: If a non real-time display interface is displayed for the control panel, and no key operation occurs, the real-time display interface is automatically skipped after 20 seconds to avoid misoperation.

Example 4-1: Set the denominator of electronic gear ratio to 10000 with number system switching

Press **MODE**. The main menu is displayed. Choose **F003**.
Press **SET**. The interface for selecting addresses is displayed.
Press ▲ to adjust data as **d3.35**.
Press **SET** to display the current value **d3.35**. Press **SET** again to modify the value d3.35. In this case, the 1st number at the right side is flickering. Short press **MODE** for three times to move to the first position on the left. Then press ▲. The value is increased to 9000. In this case, the current data is decimal.
Press ▲ again. The content of numeric display changes to “271.0”, and the 3rd decimal point (from left to right) flickers. In this case, the data is hexadecimal. Press **SET** to confirm the current value. The 1st decimal point on the right flickers. In this case, the denominator of the electronic gear ratio is modified to 10000.

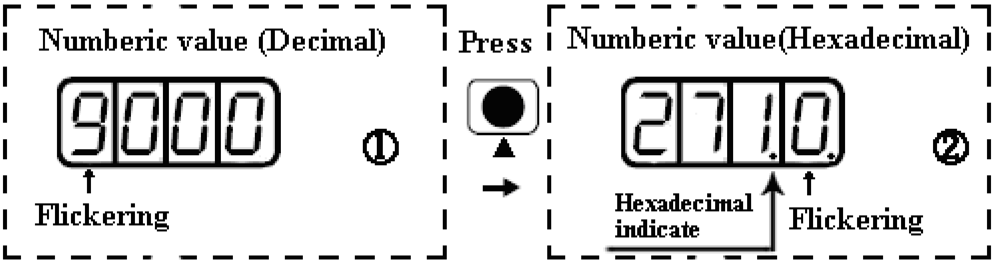


Figure 4-2 Number system conversion

Example 4-2: Set the speed to 1000 RPM/-1000 RPM with separate regulation of bits

Press **MODE**. The main menu is displayed. Choose **F000**.
Press **SET**. The interface for selecting addresses is displayed.
Press ▲ to adjust data as **d0.02**.
Press **SET** to display the current value d0.02. Press **SET** again to modify the value d0.02. In this case, the 1st number at the right side is flickering.
Short press **MODE** for three times to move to the 1st position on the left. Press ▲ to modify the value to 1. Press **SET** to confirm the current value. The 1st decimal point on the right flickers. In this case, the speed is 1000 RPM.
Press ▼ to modify the value to -1. In this case, the 1st decimal point on the left flickers, indicating that the current data is negative. Press **SET** to confirm the current value. The 1st decimal point on the right flickers. In this case, the speed is -10000 RPM.

Chapter 5 KincoServo Software Introductions

5.1 Software Installation

This software doesn't need to install.Users can download KincoServo software from our website: www.en.kinco.cn.

5.2 Quick Start

5.2.1 Hardware Configuration for Running KincoServo Software

KincoServo software can be used to configure all the parameters of FD2S Series servo driver via RS232 or CANopen port.Please refer to Chapter 3 to connect servo driver and motor before using it.

● System configuration for programming via RS232.
24VDC power supply for driver.
Serial programming cable,whose wiring diagram is as following figure.

PC FD2S Servo RS232 Interface(X5)

RxD 2 ---------------------------------- TXD 2

TxD 3 ---------------------------------- RXD 3

GND 5 ---------------------------------- GND 5

● System configuration for programming via CANopen.
24VDC power supply for driver.
PEAK series USB or LPT adapter from PEAK company.
CANopen communication cable,its wiring diagram is as following figure:

Pecan FD2S Servo CAN Interface(X4)

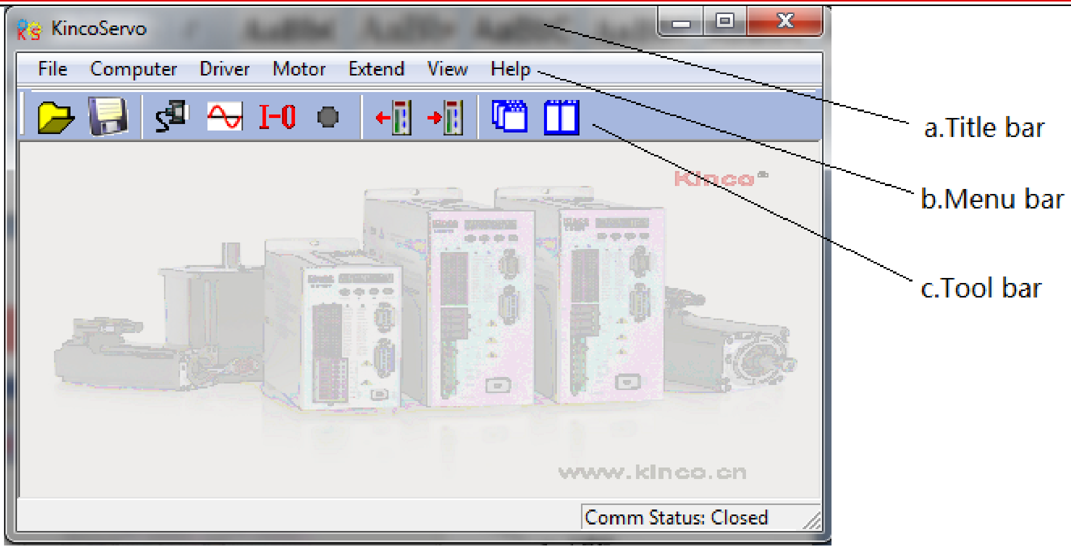
CAN_L 2 ---------------------------------- CAN_L 2

CAN_H 7 ---------------------------------- CAN_H 7

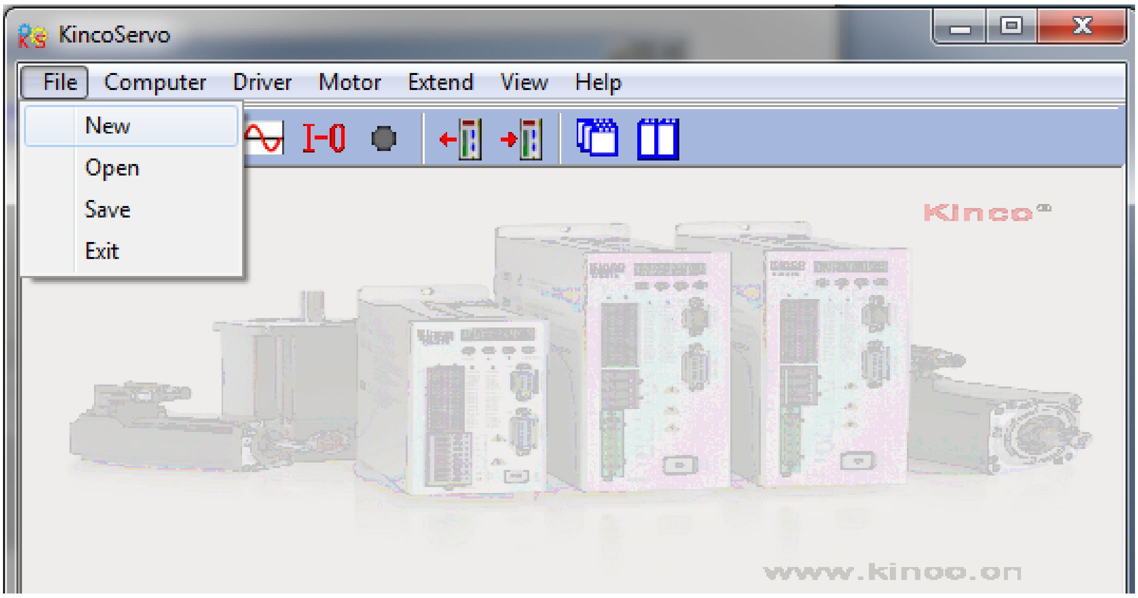
5.2.2 KincoServo Software Online

1.Open the folder of KincoServo and double click the icon 

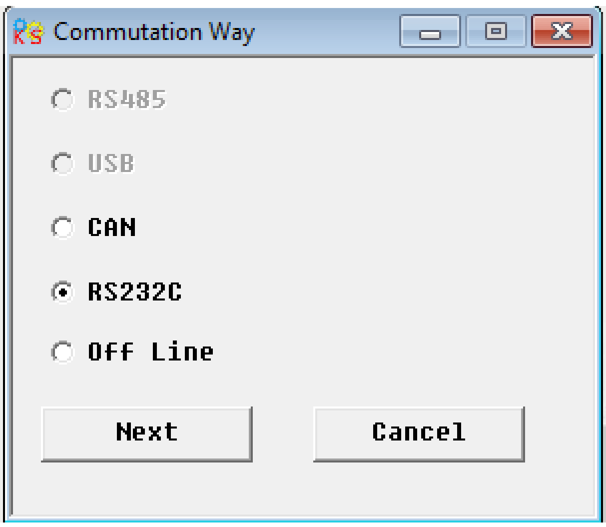
,then it will open the window as following figure:



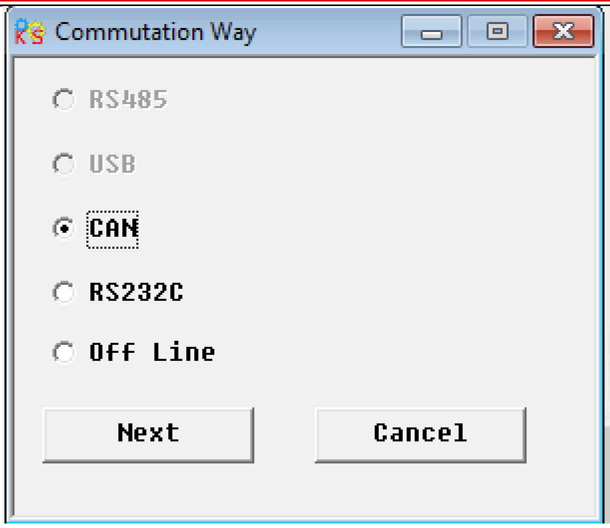
2.New Project.

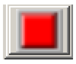


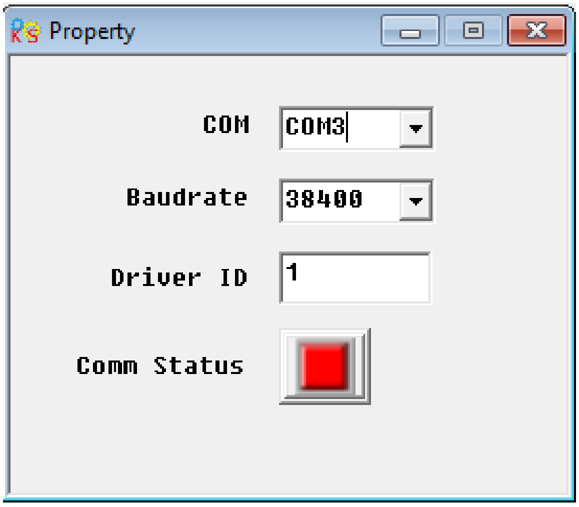
3.It will popup dialog box “**Commutation Way**”,if it uses serial port,then select “**RS232C**”and click “**Next**”.

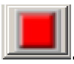


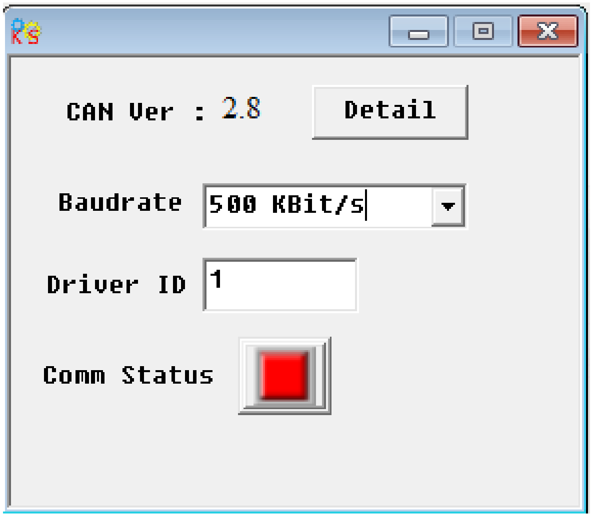
If it uses CAN tools such as PEAK-CAN,then select “**CAN**” and click “**Next**”.



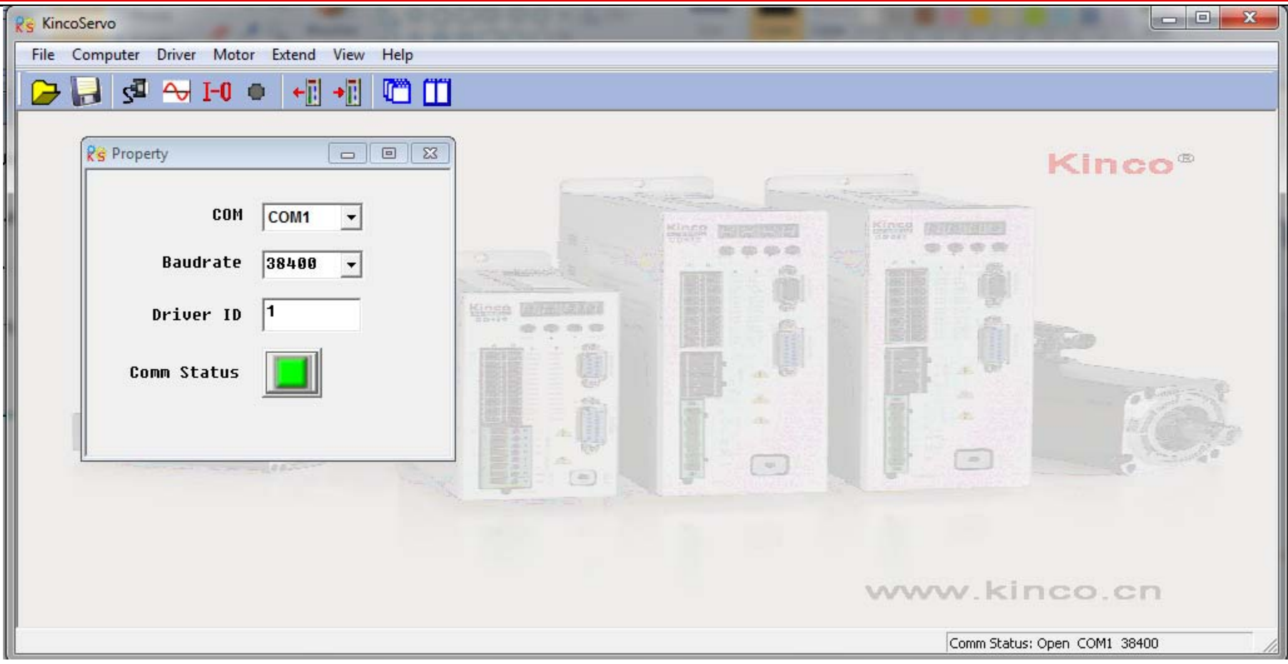
4．Enter communication property interface.Set the parameters like COM,Baudrate,Driver ID corresponding to the actual value in servo driver.Then click **Comm Status** button 。



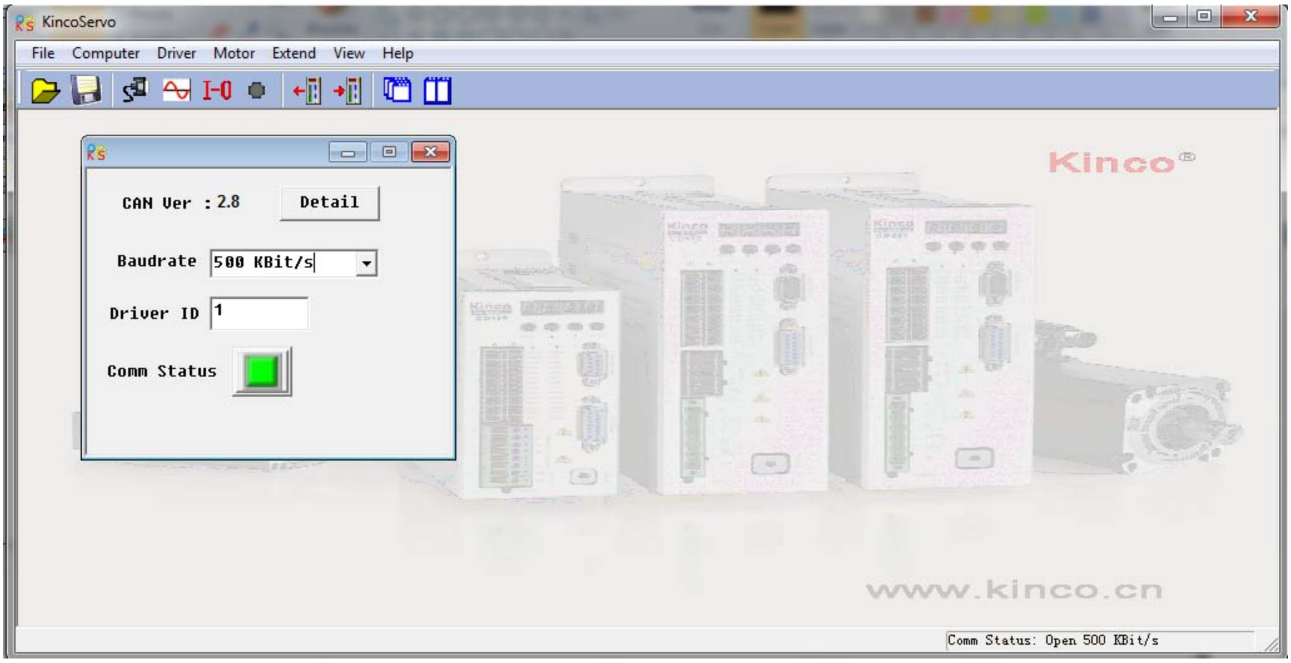
If it uses CAN connection,set the parameters like Baudrate,Driver ID.Then click **Comm Status** button .



5.Check the informations in the lower-right side.If the informations are like “Comm Status:Open COM1 38400” and the Comm Status turns green,it means KincoServo software is online successfully.

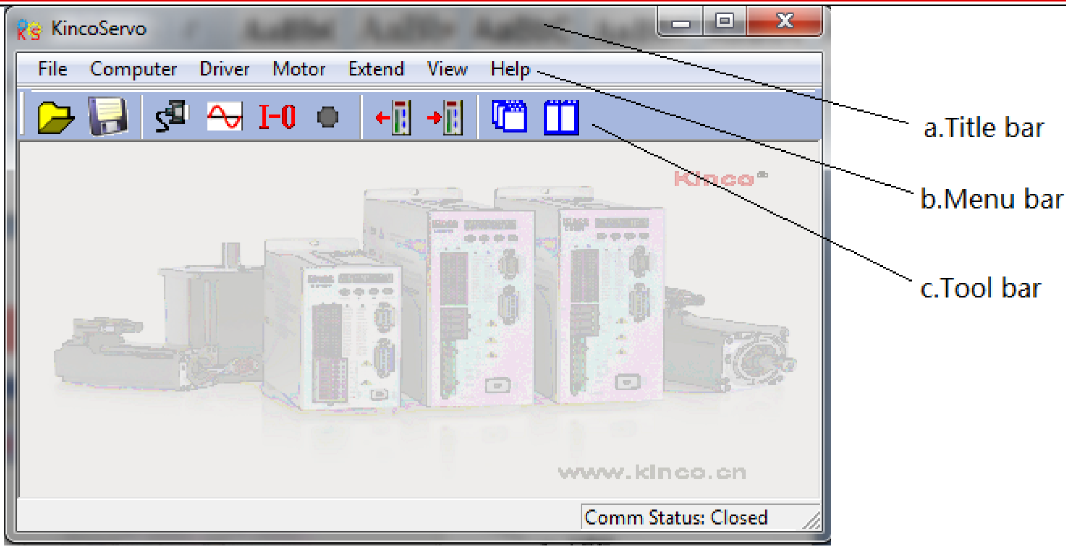


When it uses CAN connection,if the informations in the lower-right side are like “Comm Status:Open 500K Bit/S” and the Comm Status turns green,it means KincoServo software is online successfully.



5.3 Menu Introductions

Open KincoServo software as following figure:

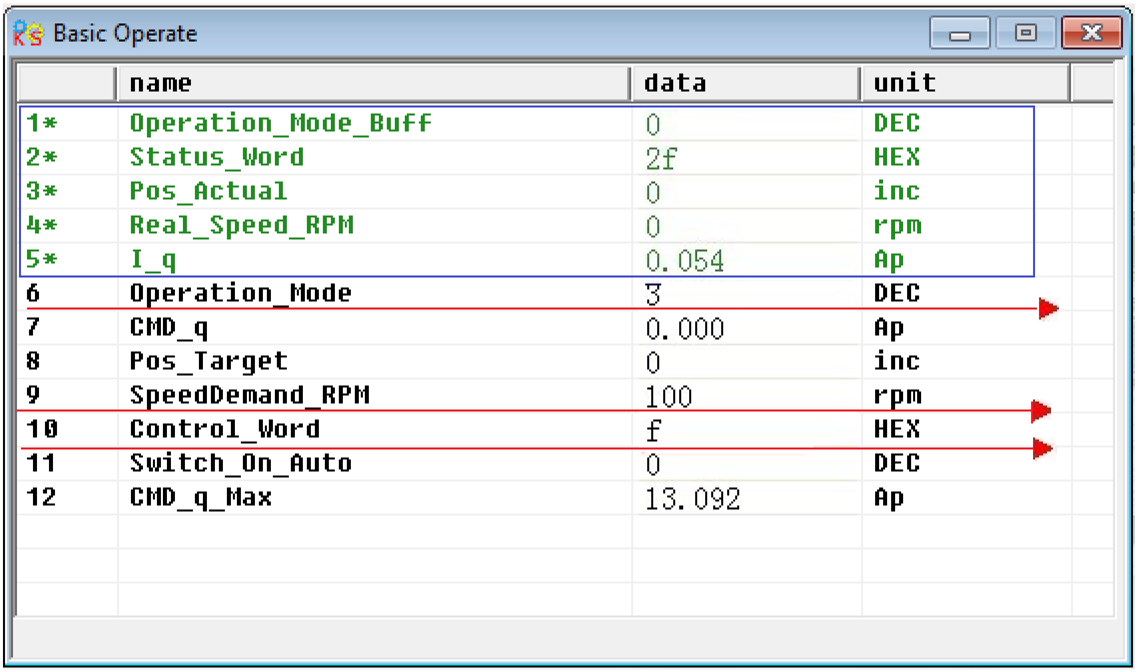


The descriptions of Menu bar are as following table.

Name	Descriptions
File	Used to New,Open,Save project.
Computer	Used to set communication property.
Driver	Used to control driver,more details please refer to 5.4
Motor	Used to configure motor parameters,more detail please refer to 6.1.3
Extend	Used to change language and read/write driver parameters.

5.4 Driver Control

5.4.1 Basic Operate

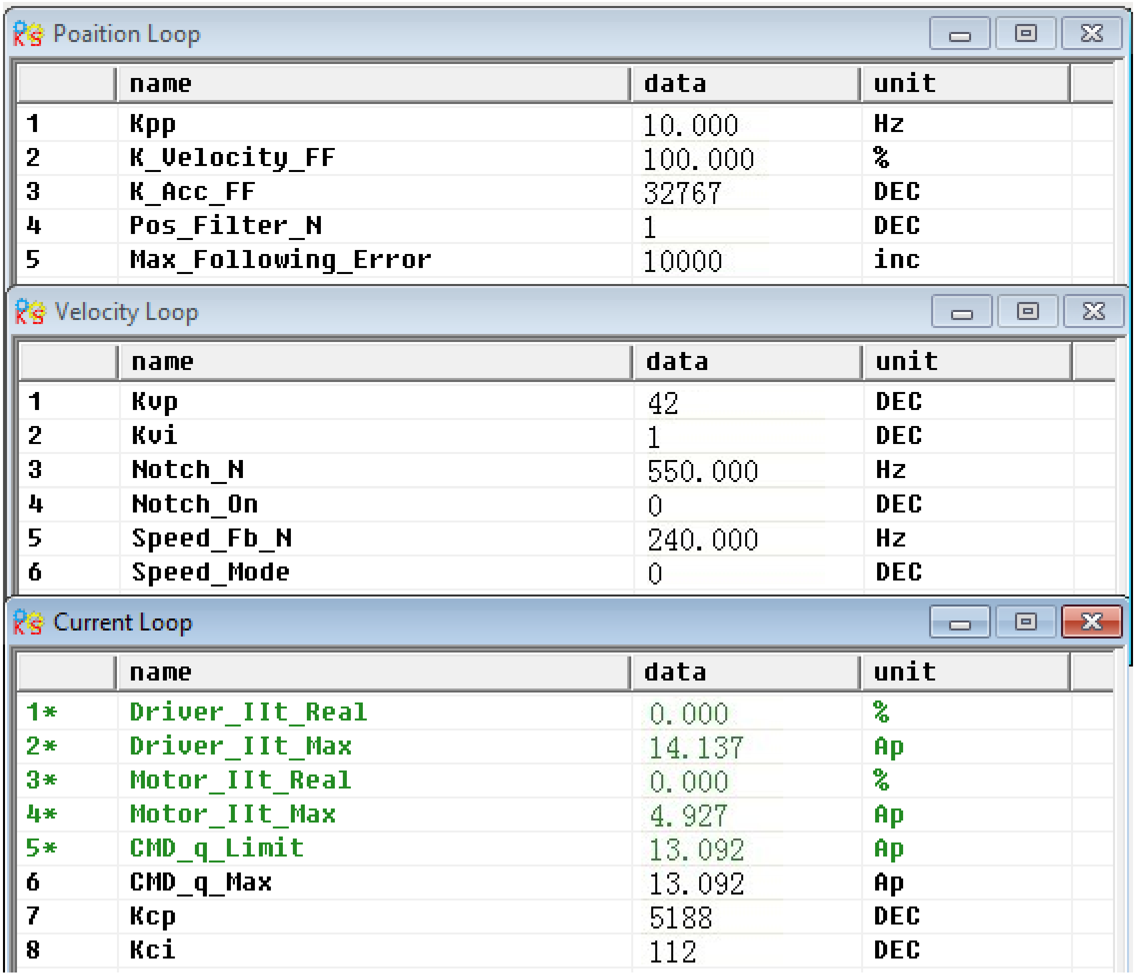


In this menu,it can do some basic control operation for driver.About more details of operation mode,please refer to Chapter8.

Example 5-1: Use KincoServo software to control servo running in speed mode by manual.

Step 1: Cancel the default setting of DIN1 and DIN3 according to Example 5-2.
Step 2: Set the basic parameters according to “Speed Mode” in Chapter 8.As shown on the red line in the figure,it means the driver is in speed mode.And the speed is 100RPM.Set the SpeedDemand_RPM as negative value when need to run reversed.

5.4.2 Control Loop



Poaition Loop

	name	data	unit
1	Kpp	10.000	Hz
2	K_Velocity_FF	100.000	%
3	K_Acc_FF	32767	DEC
4	Pos_Filter_N	1	DEC
5	Max_Following_Error	10000	inc

Velocity Loop

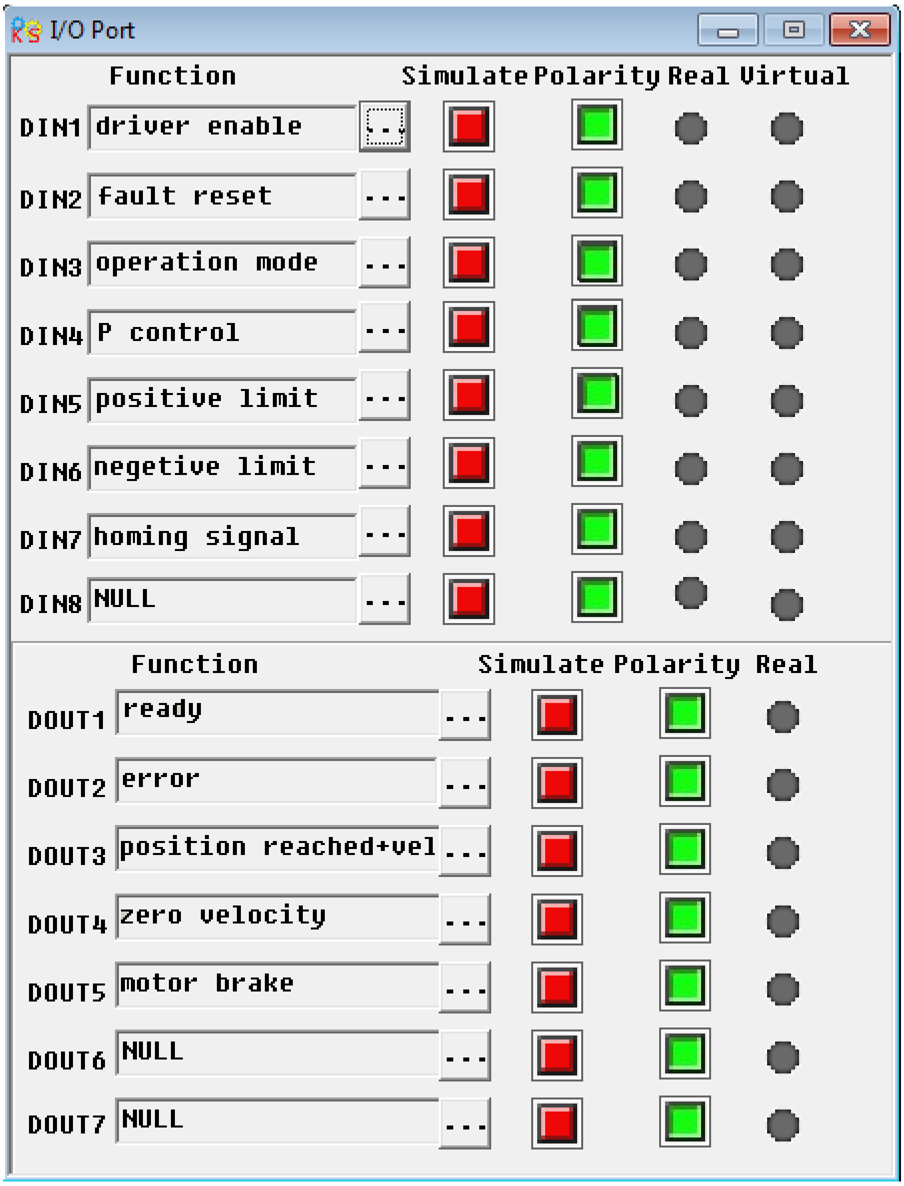
	name	data	unit
1	Kvp	42	DEC
2	Kvi	1	DEC
3	Notch_N	550.000	Hz
4	Notch_On	0	DEC
5	Speed_Fb_N	240.000	Hz
6	Speed_Mode	0	DEC

Current Loop

	name	data	unit
1*	Driver_IIt_Real	0.000	%
2*	Driver_IIt_Max	14.137	Ap
3*	Motor_IIt_Real	0.000	%
4*	Motor_IIt_Max	4.927	Ap
5*	CMD_q_Limit	13.092	Ap
6	CMD_q_Max	13.092	Ap
7	Kcp	5188	DEC
8	Kci	112	DEC

In this menu,it is used to adjust parameters for driver’s control performance.More details please refer to chapter 9.
Please be careful for parameters setting in Current Loop!If users use FD2S Servo driver together with the servo motors provided by Kinco Company,then it needn’t set the parameters in Current Loop.

5.4.3 I/O Port

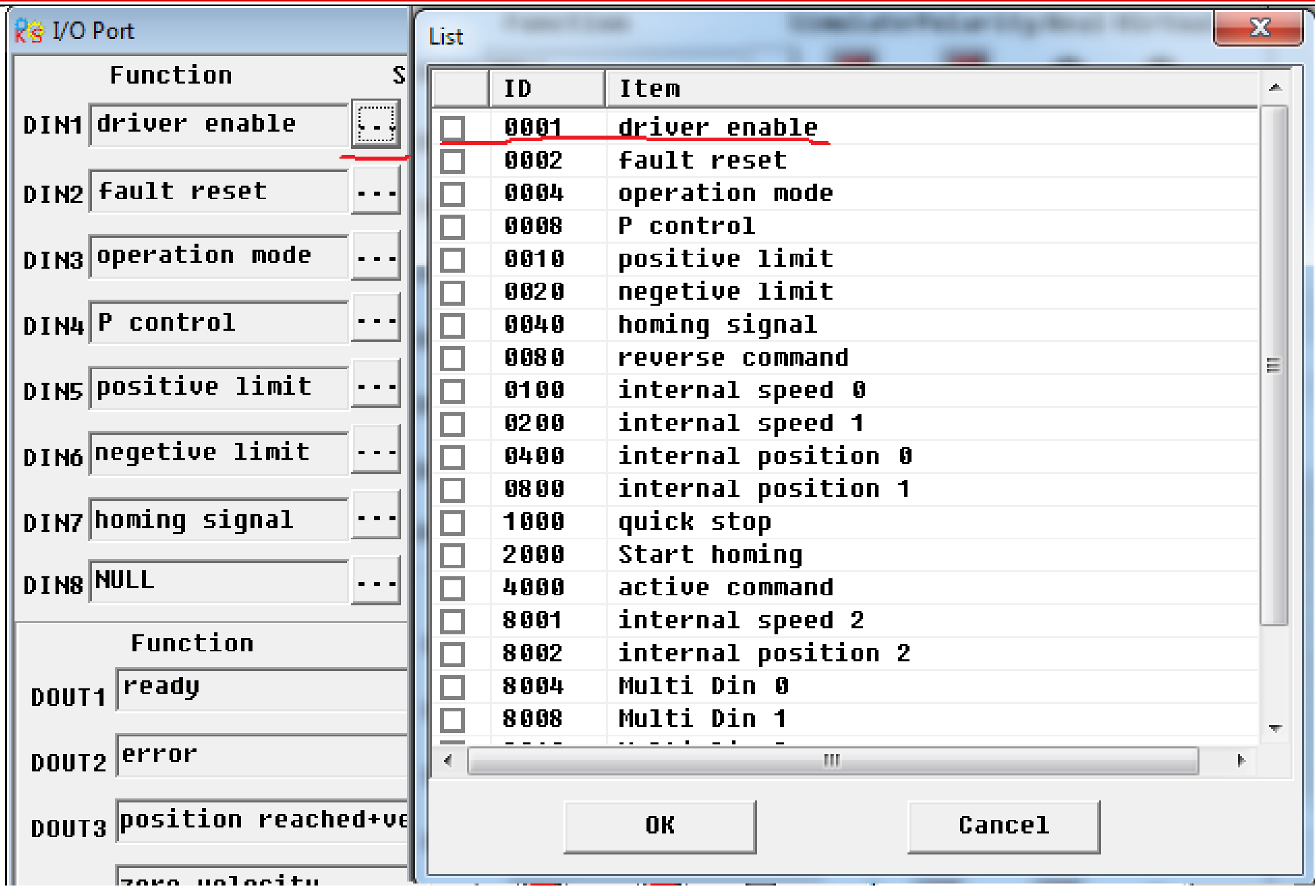


In this menu,it is used to set the functions and polarity of I/O ports,monitor the status of I/O ports and simulate the I/O ports.

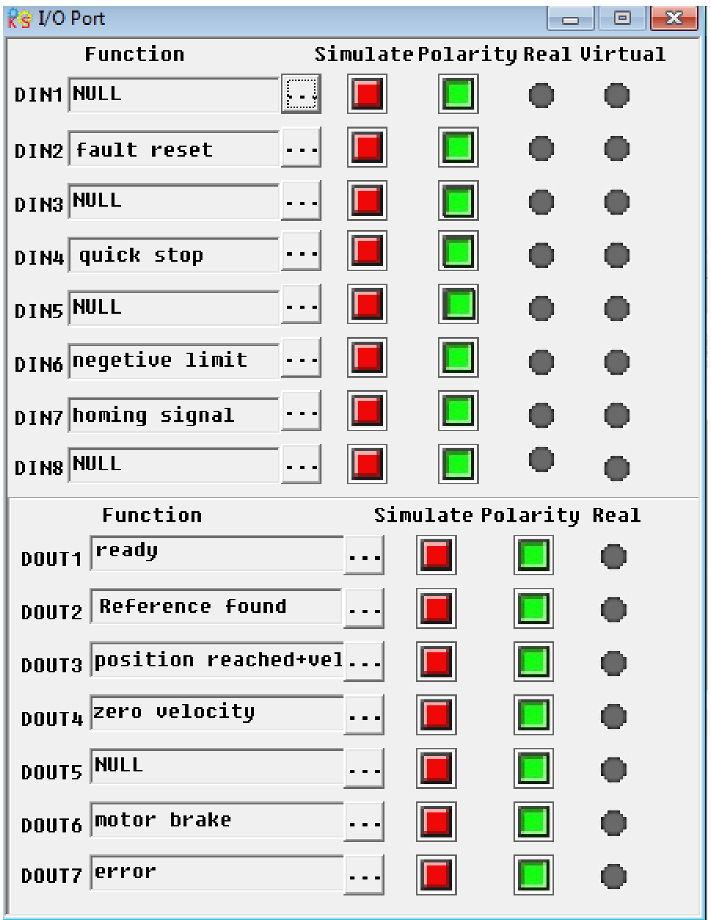
Example 5-2：Use KincoServo software to set the functions of I/O port

Requirement: Cancel the functions of DIN1, DIN3 and DIN5.Set DIN2 as default reset,DIN4 as emergency stop and OUT2 as Reference found.Others are set as default.

Step 1: Click the button  beside DIN1.Cancel the function “Driver enable” in the popup window as following figure, then click OK.

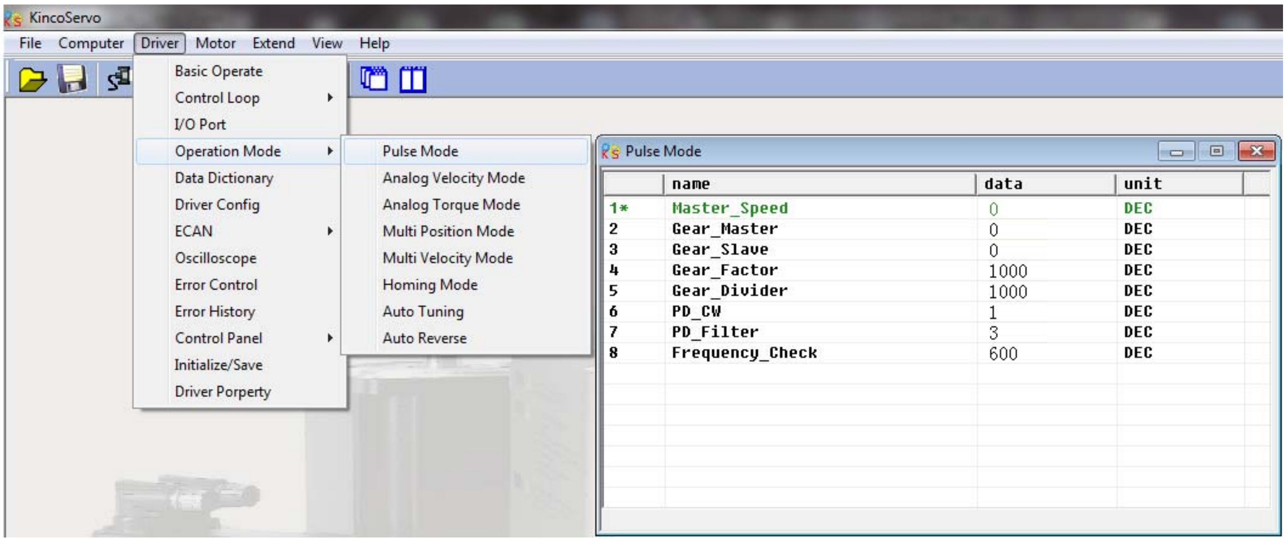


Step 2: Set all the functions of other I/O ports with the similar operations as step 1.Then select Driver -> Initialize/Save and click “Save control parameters”.The final settings of I/O ports are as following figure:

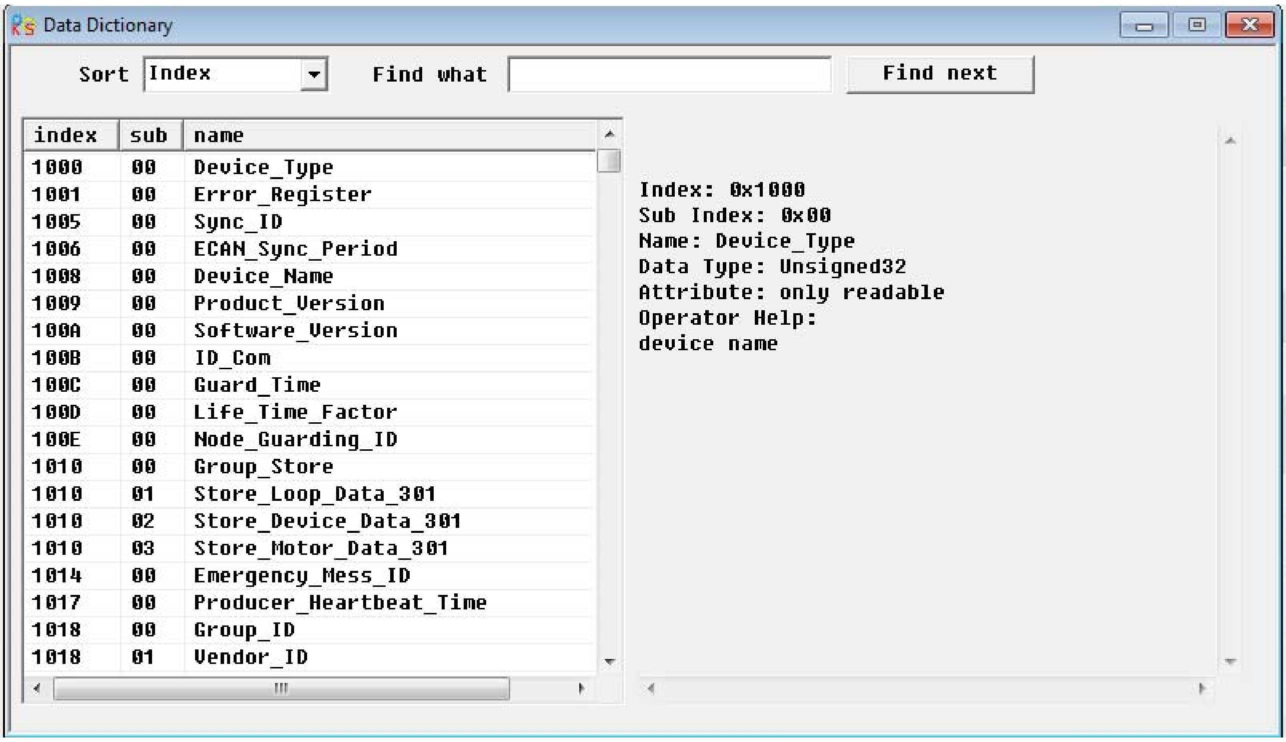


5.4.4 Operation Mode

In this menu,it is used to set and monitor the objects in each operation mode.More details please refer to chapter 9.Following figure is the menu for pulse mode.



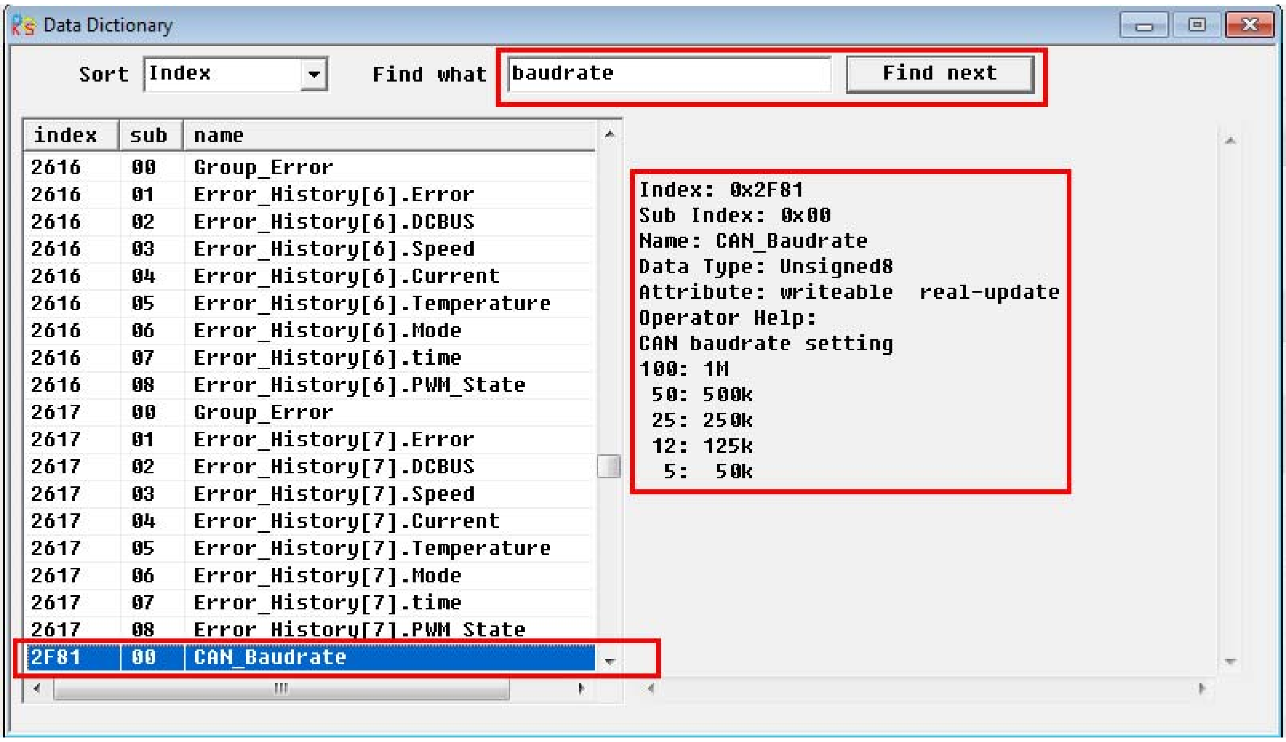
5.4.5 Data Object



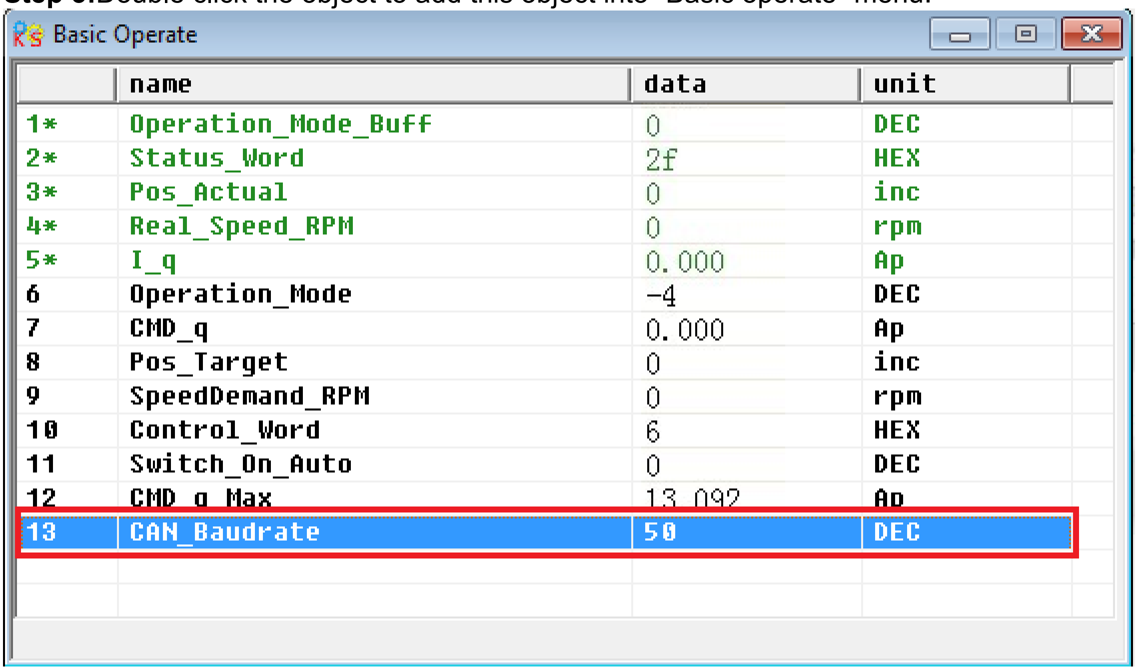
In this menu,it can be used to query the address and descriptions of all the objects in FD2S driver.As shown in above picture,there are Index,Subindex address and the name of the objects on the left side.On the right side,there are the descriptions of the object.

Example 5-3：Use KincoServo Software to Add an Object

Requirement:Add an address in any menu.Here   we will add “CANopen baudrate” in “Basic Operate”.
Step 1: Open “Basic Operate”,then righ click in the window of “Basic Operate”.Select “add”,then it will popup a window of “Data Object”.
Step 2: Enter “baudrate” in “Find what”,then click “Find next”.It will jump to the object “CAN_Baudrate” whose index address is 2F81.There are the descriptions of this object in the rightside. As shown in following figure.



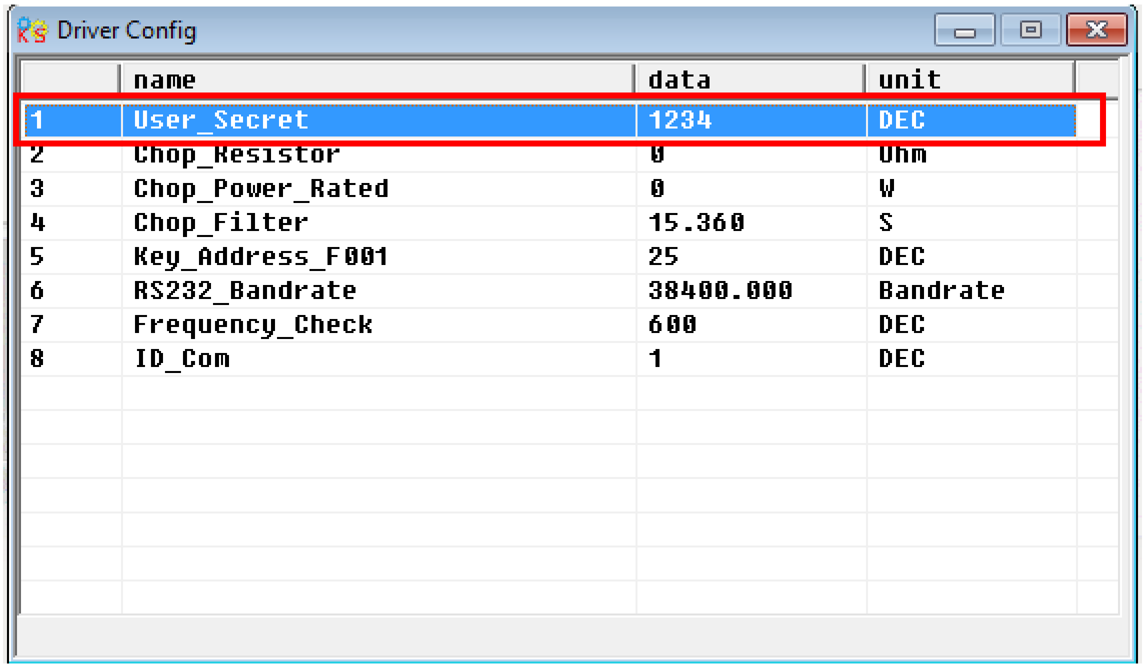
Step 3: Double click the object to add this object into “Basic operate” menu.



Step 4: If you need to delete the object in the menu.Right click the object and select “del”to delete the object.If you need to know more details of the object,then right click the object and select “help” to show the details.

5.4.6 Driver Config

In this menu,it is used to set the parameters such as User Password,Brake resistor,RS232 communication and so on.



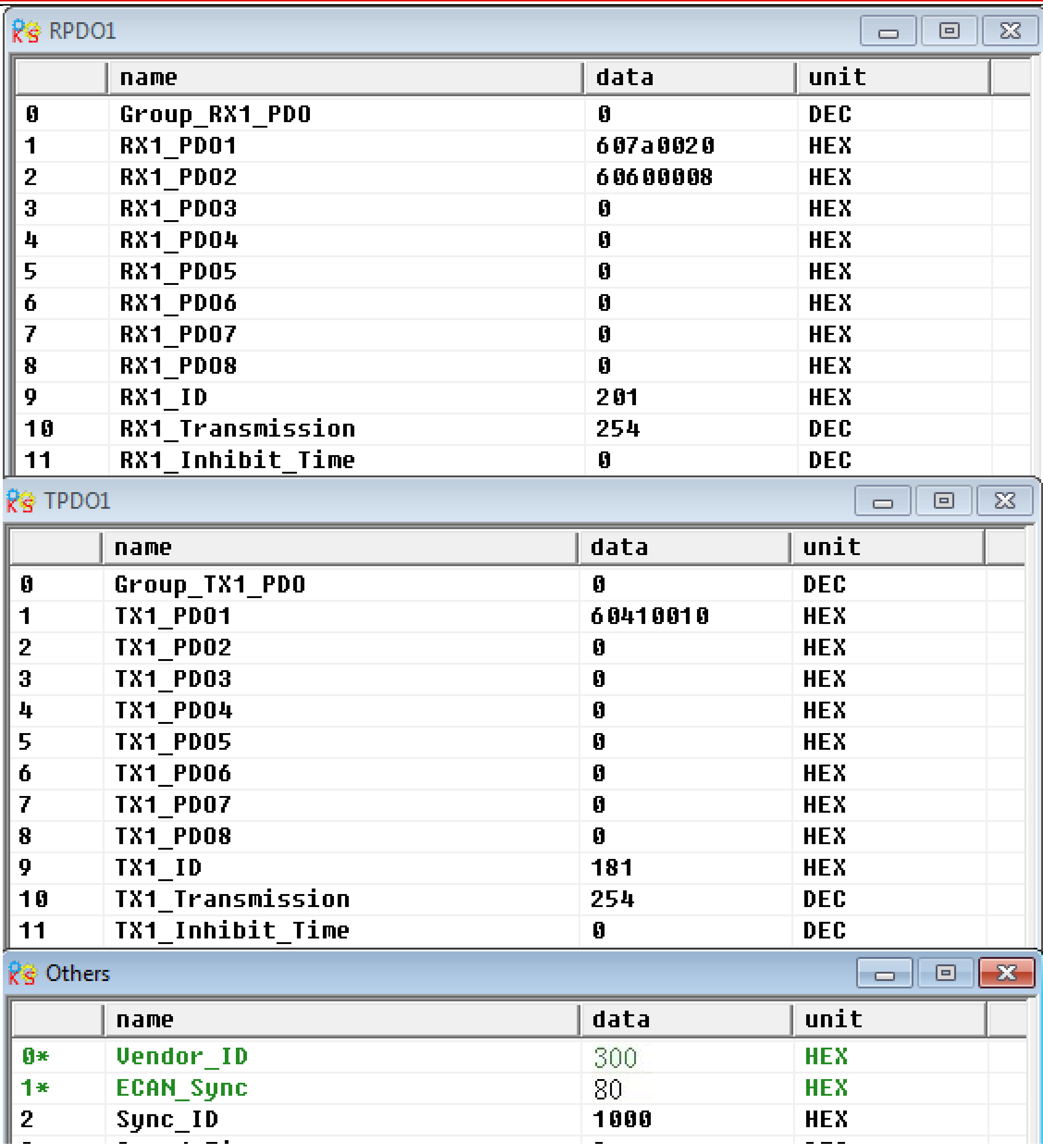
	name	data	unit
1	User_Secret	1234	DEC
2	Chop_Resistor	0	Ohm
3	Chop_Power_Rated	0	W
4	Chop_Filter	15.360	S
5	Key_Address_F001	25	DEC
6	RS232_Bandrate	38400.000	Bandrate
7	Frequency_Check	600	DEC
8	ID_Com	1	DEC

Example 5-4：Use KincoServo to set an User Password

Step 1: Set the number “1234”as password in the object “User_Secret” as shown in the red box in the figure above.
Step 2: Click “Save all control parameters” in Driver->Initialize/Save to save parameters,then Click “Reboot driver”.
Step 3: The password will be activated after rebooting driver.Then users can not set any parameters before entering the correct password in the object “User_Secret”in “Driver Config”.
Step 4: Enter 0 in the object “User_Secret” to cancel the password after entering correct password.

5.4.7 ECAN Setting（CANopen PDO Setting）

This menu is used to set CANopen communication parameters.About details please refer to chapter 10.



RPDO1

	name	data	unit
0	Group_RX1_PDO	0	DEC
1	RX1_PDO1	607a0020	HEX
2	RX1_PDO2	60600008	HEX
3	RX1_PDO3	0	HEX
4	RX1_PDO4	0	HEX
5	RX1_PDO5	0	HEX
6	RX1_PDO6	0	HEX
7	RX1_PDO7	0	HEX
8	RX1_PDO8	0	HEX
9	RX1_ID	201	HEX
10	RX1_Transmission	254	DEC
11	RX1_Inhibit_Time	0	DEC

TPDO1

	name	data	unit
0	Group_TX1_PDO	0	DEC
1	TX1_PDO1	60410010	HEX
2	TX1_PDO2	0	HEX
3	TX1_PDO3	0	HEX
4	TX1_PDO4	0	HEX
5	TX1_PDO5	0	HEX
6	TX1_PDO6	0	HEX
7	TX1_PDO7	0	HEX
8	TX1_PDO8	0	HEX
9	TX1_ID	181	HEX
10	TX1_Transmission	254	DEC
11	TX1_Inhibit_Time	0	DEC

Others

	name	data	unit
0*	Vendor_ID	300	HEX
1*	ECAN_Sync	80	HEX
2	Sync_ID	1000	HEX

5.4.8 Oscilloscope

Oscilloscope can help you adjust servo's parameters better by observing the curve of speed,position and so on.

There are two ways to open oscilloscope as following figures.

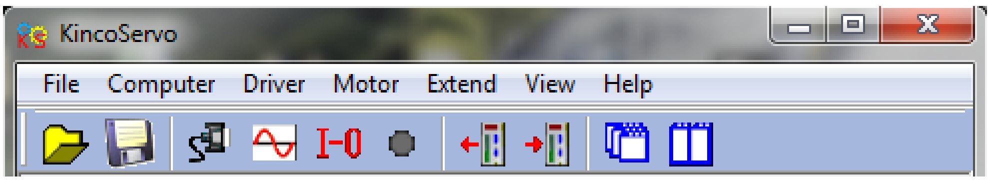


Fig.1.Oscilloscope shotcut in toolbar

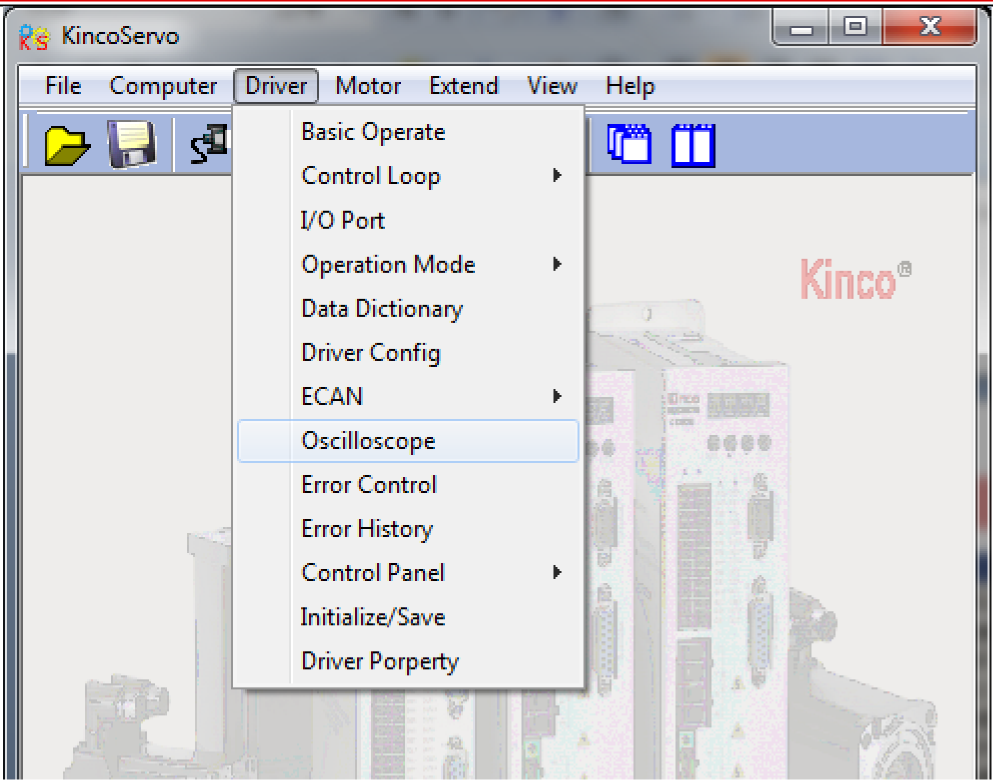
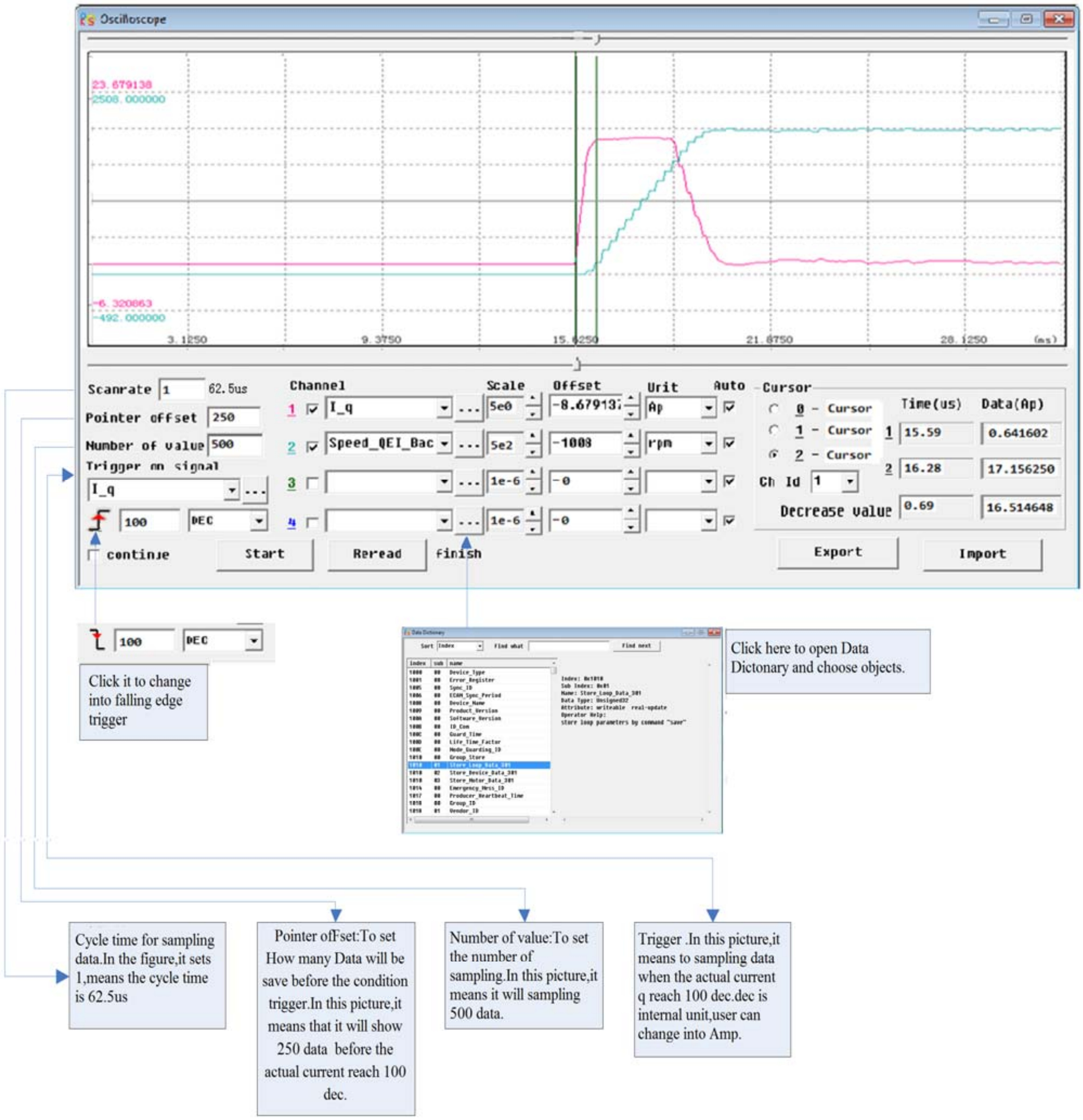
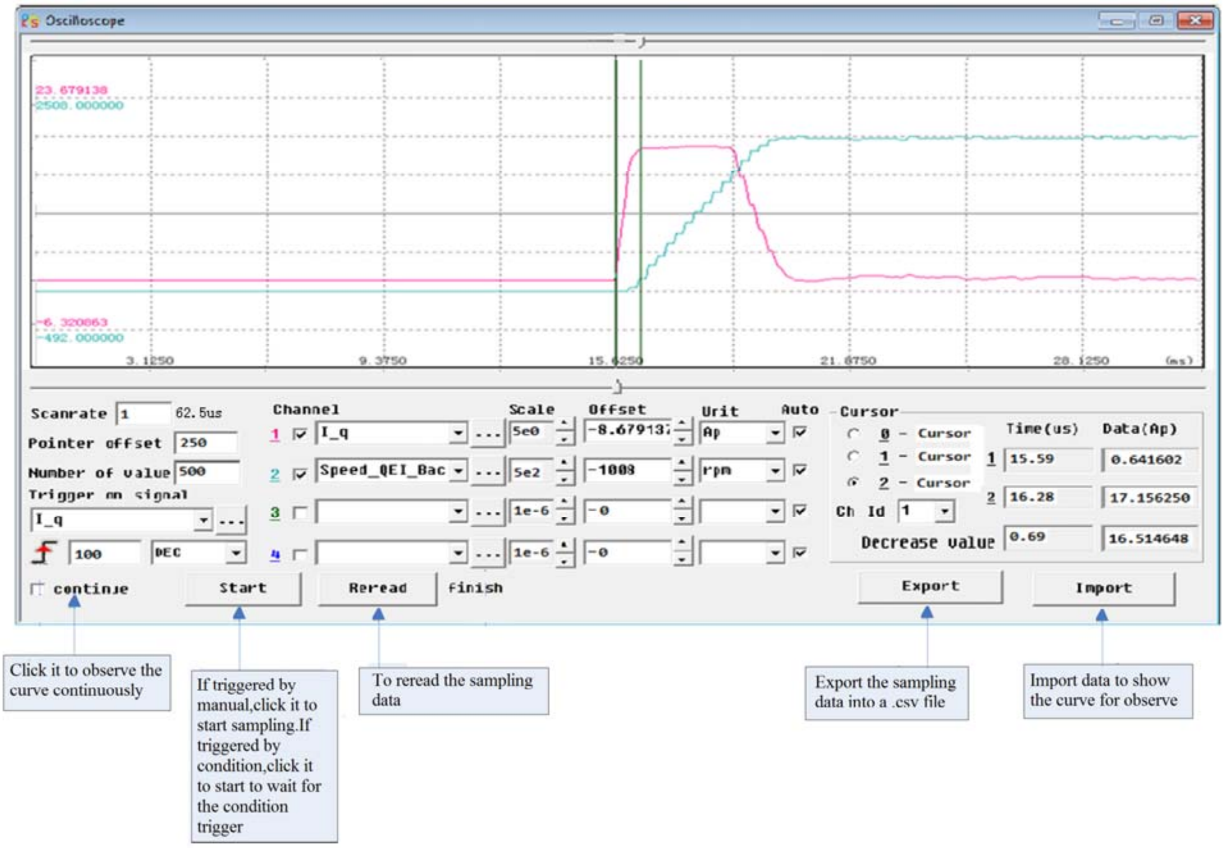
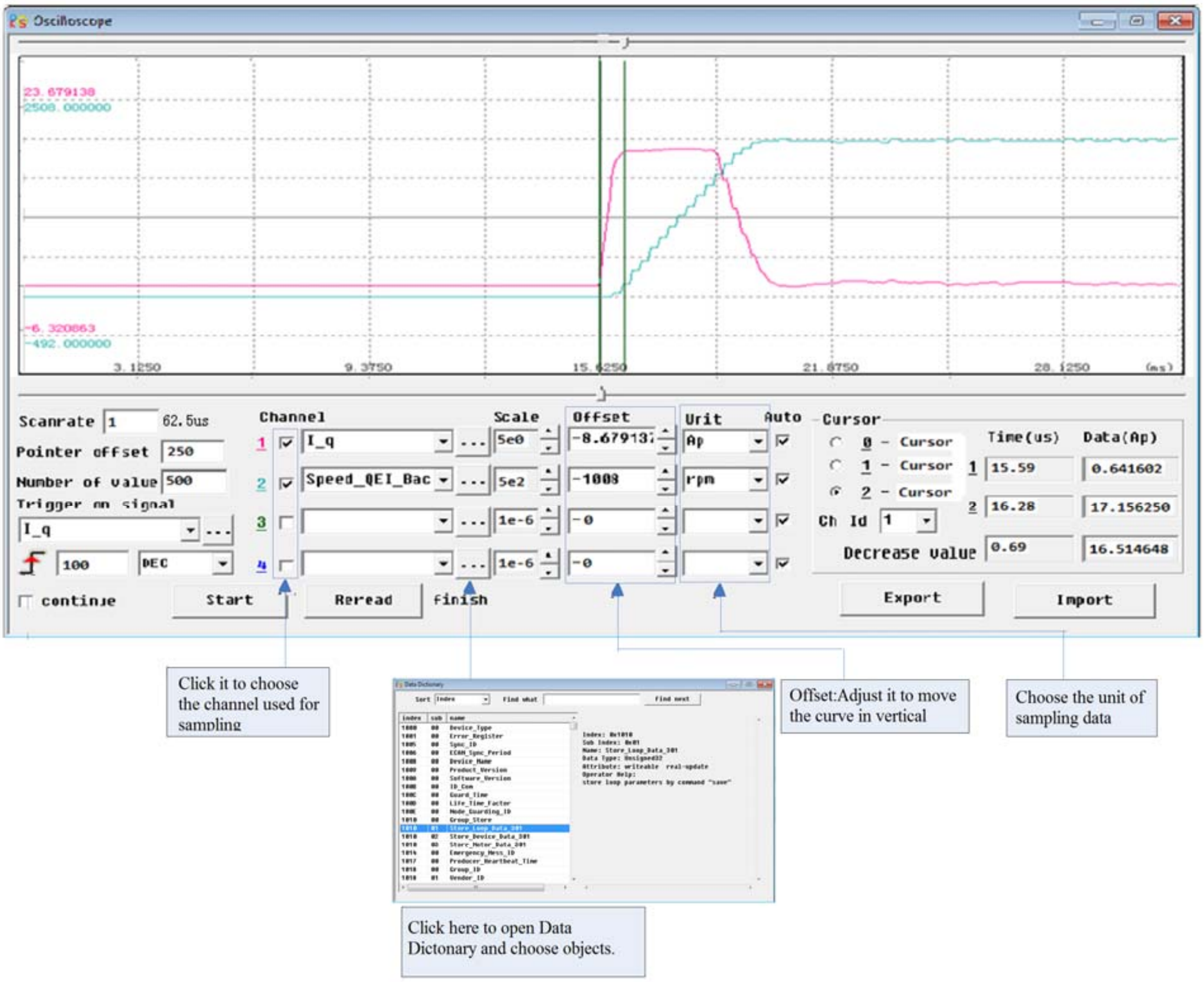


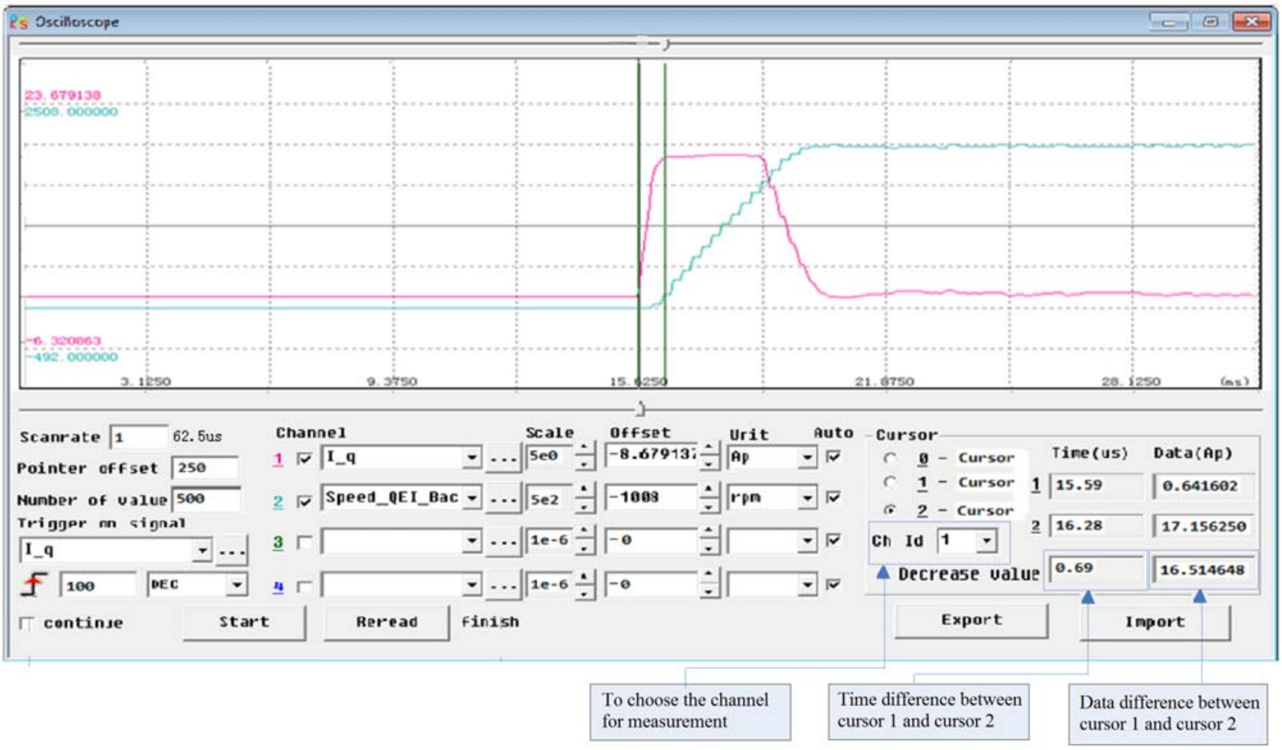
Fig.2.Menu bar---Driver--Oscilloscope

Follows are the parameters instructions in Oscilloscope.




Oscilloscope
Scanrate
Pointer offset
Number of value
Trigger on signal
Channel
Scale
Offset
Unit
Auto
Cursor
Time(us)
Data(Ap)
Decrease value
continue
Start
Reread
Finish
Export
Import
Click it to observe the curve continuously
If triggered by manual,click it to start sampling.If triggered by condition,click it to start to wait for the condition trigger
To reread the sampling data
Export the sampling data into a .csv file
Import data to show the curve for observe

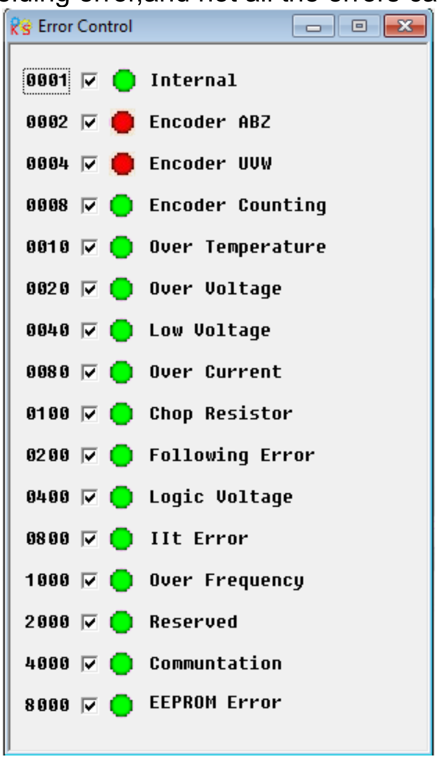

Oscilloscope
Scanrate
Pointer offset
Number of value
Trigger on signal
Channel
Scale
Offset
Unit
Auto
Cursor
Time(us)
Data(Ap)
Decrease value
continue
Start
Reread
Finish
Export
Import
Click it to choose the channel used for sampling
Click here to open Data Dictonary and choose objects.
Offset:Adjust it to move the curve in vertical
Choose the unit of sampling data



5.4.9 Error Control

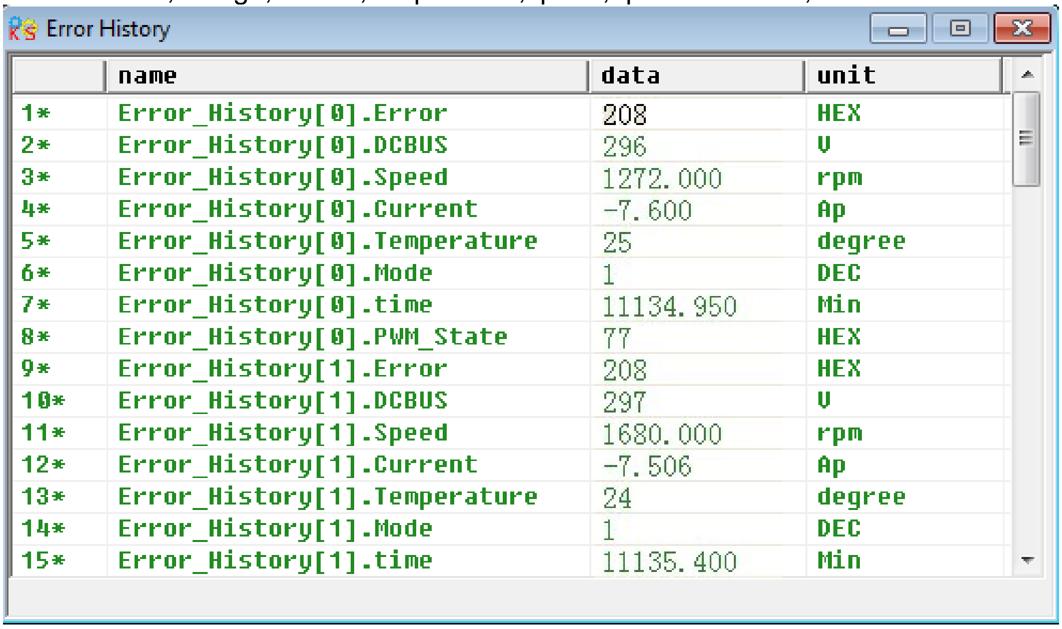
This menu is used to monitor the current error information.As shown in following figure,The Hex data is the same error code as shown in LED display on servo driver.The small box is used to choose whether to shield error or not.There is error when the lamp is red.The text is the descriptions of error.About more details please refer to chapter 11.

Note: Please be careful for shielding error,and not all the errors can be shielded.



5.4.10 Error History

FD2S Servo driver provides 7 groups of historical error informations.Users can query the informations such as error code,voltage,current,temperature,speed,operation mode,driver accumulated working time and so on.


Error History

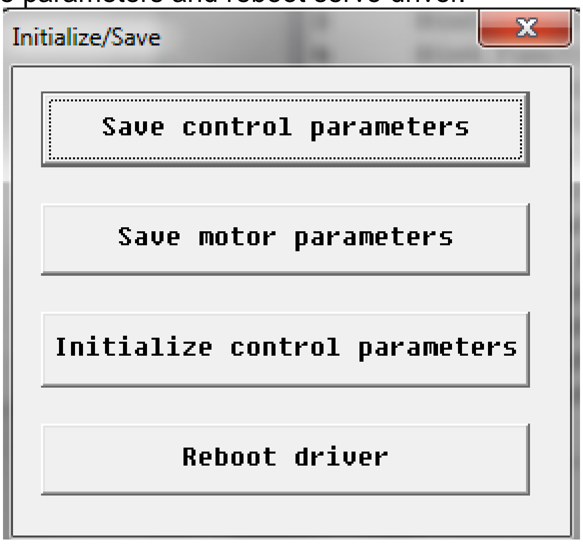
	name	data	unit
1*	Error_History[0].Error	208	HEX
2*	Error_History[0].DCBUS	296	V
3*	Error_History[0].Speed	1272.000	rpm
4*	Error_History[0].Current	-7.600	Ap
5*	Error_History[0].Temperature	25	degree
6*	Error_History[0].Mode	1	DEC
7*	Error_History[0].time	11134.950	Min
8*	Error_History[0].PWM_State	77	HEX
9*	Error_History[1].Error	208	HEX
10*	Error_History[1].DCBUS	297	V
11*	Error_History[1].Speed	1680.000	rpm
12*	Error_History[1].Current	-7.506	Ap
13*	Error_History[1].Temperature	24	degree
14*	Error_History[1].Mode	1	DEC
15*	Error_History[1].time	11135.400	Min

5.4.11 Control Panel

This menu is used to set and query all the parameters which are corresponding to the parameters from Group F000 to F007 in servo driver.

5.4.12 Initialize/Save

This menu is used to save and initialize parameters and reboot servo driver.



5.4.13 Driver Property

This menu is used to display the informations such as driver model,software version,serial number and so on.

Chapter 6 Motor Selection,Trial Operation and Parameter List

6.1 Driver and motor configuration

There is no default motor type set in driver,so users need to set the motor model before using the driver.Please refer to the selection table in 6.1.1 when setting the motor model.

6.1.1 Configuration Table for FD2S Servo Driver and Motor

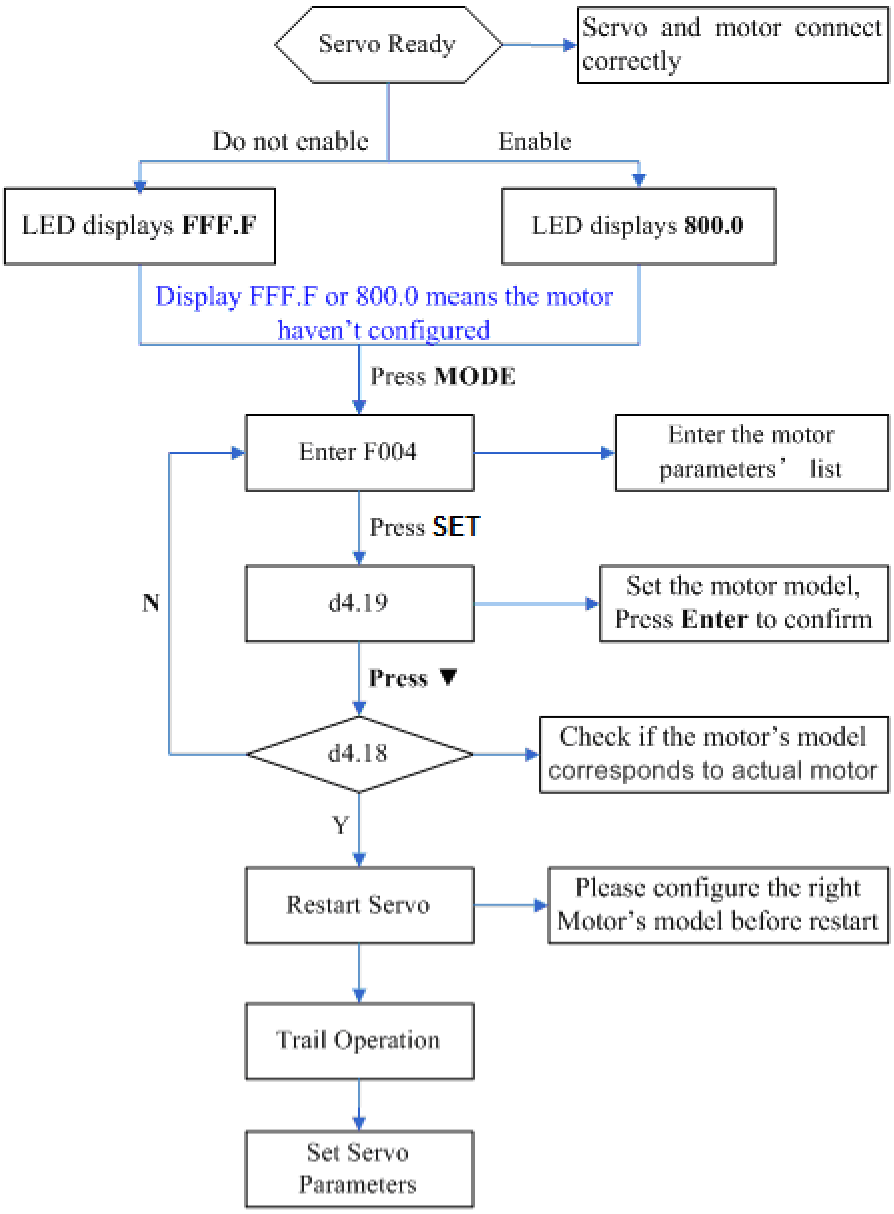
PC	LED	Motor Model	Suitable Servo					
LED CODE:EA01			CD412S FD412S	CD422S FD422S	With Fan CD422S-AF FD422S-AF(CF、LF)	CD432S FD432S	CD612S FD612S	CD622S FD622S
K@	404.b	Without motor configuration	LED displays FFF.F					
W0	305.7	SMC60S-0020-30E■K-3LKH		√				
W1	315.7	SMC60S-0040-30E■K-3LKH		√				
W2	325.7	SMC80S-0075-30E■K-3LKH		√				
WB	425.7	SMC130D-0100-20E■K-4LKP			√			
WC	435.7	SMC130D-0150-20E■K-4HKP					√	
WD	445.7	SMC130D-0200-20E■K-4HKP					√	
WO	4F5.7	SMC130D-0150-20E■K-4LKP				√		
WP	505.7	SMC130D-0200-20E■K-4LKP				√		
WQ	515.7	SMC130D-0300-30E■K-4HKP						√
WR	525.7	SMC130D-0300-20E■K-4HKP						√
Y0	305.9	SMS60S-0020-30J■K-3LKU		√				
Y1	315.9	SMS60S-0040-30J■K-3LKU		√				
Y2	325.9	SMS80S-0075-30J■K-3LKU		√				
Z0	305.A	SMS60S-0020-30K■K-3LKU		√				
Z1	315.A	SMS60S-0040-30K■K-3LKU		√				
Z2	325.A	SMS80S-0075-30K■K-3LKU		√				
KZ	5A4.b	SMH40S-0005-30A■K-4LKH	√					
KY	594.b	SMH40S-0010-30A■K-4LKH	√					
K0	304.b	SMH60S-0020-30A■K-3LK□		√				
K1	314.b	SMH60S-0040-30A■K-3LK□		√				
K2	324.b	SMH80S-0075-30A■K-3LK□		√				
K3	334.b	SMH80S-0100-30A■K-3LK□				√		
K4	344.b	SMH110D-0105-20A■K-4LK□				√		
K5	354.b	SMH110D-0125-30A■K-4LK□				√		
K6	364.b	SMH110D-0126-20A■K-4LK□				√		
K7	374.b	SMH110D-0126-30A■K-4HK□						√
K8	384.b	SMH110D-0157-30A■K-4HK□						√
K9	394.b	SMH110D-0188-30A■K-4HK□						√
KB	424.b	SMH130D-0105-20A■K-4HK□				√		√

KC	434.b	SMH130D-0157-20A■K-4HK□				√		√
KD	444.b	SMH130D-0210-20A■K-4HK□						√
KE	454.b	SMH150D-0230-20A■K-4HK□						√
F4	344.6	85S-0025-05AAK-FLFN-02		√				
F6	364.6	85S-0035-05AAK-FLFN-02		√				
F8	384.6	85S-0045-05AAK-FLFN-02		√				

6.1.2 Procedure for Motor configuration

If there is no motor type set in driver, then the driver will appear error FFF.F or 800.0.There are two ways to set the motor type in driver as follows:

1.Panel operation.



Please configure the right motor's model before restart. If customers want to reset the motor model, they should set D4.19 to 303.0 (Press SET to confirm) and then d4.00 to 1(Save motor parameters), after restart the servo they can reset motor model and servo parameters according to the above chart

2.KincoServo software operation

Connect the servo to PC, open the KincoServo, then Menu—Driver—Control Panel—F004, in the F004, in the F004, set the 19th operation: **Motor Num** (Please refer to the servo and motor configuration table), after that press Enter to confirm, then restart servo.

Please configure the right Motor's model before restart. If the customers want to reset the motor model, they should set d4.19 (Motor Num in F004) to 00(Press SET to confirm), then enter the **Initialize/Save** page, click the **Save motor parameters.** After restart the servo, they can reset the motor model and set servo parameters.

6.2 Trial Operation

6.2.1 Objective

The trial operation allows you to test whether the driver works properly, and whether the motor runs stably.

6.2.2 Precautions

Ensure the motor type is set correctly.
Ensure that the motor is running without load. If the motor flange is fixed on the machine, ensure that the motor shaft is disconnected from the machine.
Ensure that motor cables, motor encoder cables, and power circuits (power lines and control power lines) are properly connected. For details, see Chapter 3.
During the trial operation, if you long press ▲ or ▼ when the motor is running, pulse signals, digital input signals, and analog signals of the external controller are temporarily unavailable, so safety must be ensured.
During the trial operation, the system automatically adopts the instantaneous speed mode, that is, the "-3" mode.
After the trial operation, Group F006 exits automatically. To enter Group F006 again, you must re-activate the trial operation.
If motor/encoder cables are wrongly connected, the actual rotation speed of the motor may be the possible maximum rotation speed, or the rotation speed is 0 and the actual current value is the maximum value. In this case, make sure to release the button; then check cable connection and test it again.
If there is problem in the keys,then trial operation can not be used.

6.2.3 Operating Procedure

Please make sure the correct wiring of STO(refer to chanpter 3.4.3) before using trial operation,or the driver will display error 200.0.
Operate by panel:
Press **MODE** to enter Group F004. Select the object address "d4.18", and check the motor type.
Press **MODE** to enter Group F000. Select the object address "d0.02", and set the target speed to "SpeedDemand_RPM".
Press **MODE** to enter Group F006. Arrange a test for keys, with the default value of d6.40. Firstly, press ▼ to adjust the data to d6.31. Then, press ▼, the data automatically changes to "d6.15". Finally, press ▲ to adjust the data to d6.25.
Press **SET** to activate trial operation. In this case, the numeric display is "adc.d", and the motor shaft releases. When long pressing ▲ or ▼, the motor automatically locks, and runs according to "+SpeedDemand_RPM" or "-SpeedDemand_RPM" separately. During the trial operation, the numeric displays the motor speed in real time.
The motor set counter clockwise as positive direction.If the direction is not fit for the requirement ,users can change the direction through the parameter d2.16 in Group F002.
Operate by CD-PC software:

1：Set motor mode in "Motor" in the software.
2：Refer to Fig.5-1 to operate by manual.

6.2.4 Diagram of Trial Operation

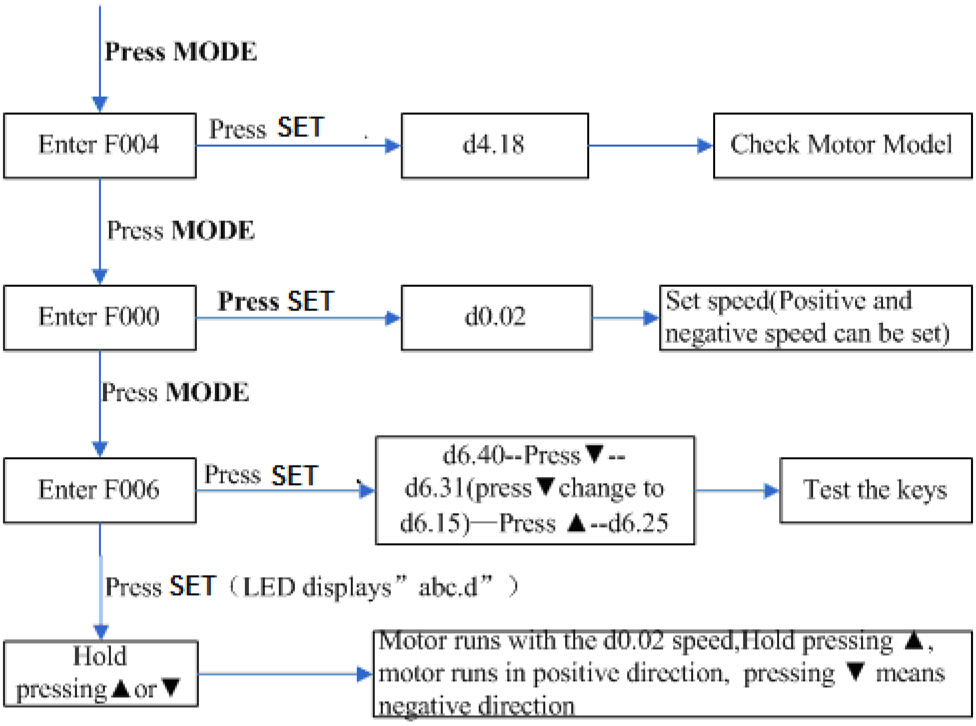


Fig.6-1 Trial operation

6.3 Descriptions of Parameters

Group F000 represents an instruction group, and the parameters in this group cannot be saved.
The address d4.00 is used to save the motor parameters set for Group F004. Note that this group of parameters must be set when customers choose third-party motors, but these parameters need not to be set for the motors delivered and configured by our company.
d2.00, d3.00 and d.5.00 represent the same address, and are used to save all setup parameters except those of motors (Group F001/F002/F003/F004/F005). Three numeric objects (d2.00/d3.00/d5.00) are developed to facilitate customers.

Parameter List: Group F000 (To Set Driver Instructions)

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
d0.00	60600008	Operation_Mode	0.004 (-4): Pulse control mode, including pulse direction (P/D) and double pulse (CW/CCW) modes. 0.003 (-3): instantaneous speed mode 0001 (1): Internal position control mode 0003 (3): Speed mode with	-4	/

			acceleration/deceleration 0004 (4): Torque mode Note: Only applied in the working mode where no external signals control the driver.		
d0.01	2FF00508	Control_Word_Easy	000.0: Releases the motor 000.1: Locks the motor 001.0: Clears errors Note: Only applied in the situation where enabling a driver or wrong resetting is not controlled by external signals. After the wrong reset of the driver, the motor must be enabled again.	0	/
d0.02	2FF00910	SpeedDemand_RPM	Sets the motor's target rotation speed when the driver works in the "-3" or "3" mode and the address d3.28 is set to 0 (without external analog control).	0	/
d0.03	60710010	Target_Torque %	Sets input torque instructions (current instructions) when the driver works in the "4" mode and the address d3.30 is set to 0 (without external analog control).	0	-2047～2047
d0.04	2FF00A10	Vc_Loop_BW	Sets the velocity loop bandwidth. The unit is Hz. This variable can only be set after auto tuning is performed properly; otherwise the actual bandwidth goes wrong, which causes abnormal working of the driver. If the auto tuning result is abnormal, setting this parameter may also cause abnormal working of the driver. Note: This parameter cannot be applied when auto tuning is unavailable. After setting this parameter, apply d2.00 to save the settings as required.	/	0～600
d0.05	2FF00B10	Pc_Loop_BW	Sets the position loop bandwidth. The unit is Hz. Note: After setting this parameter, apply d2.00 to save the settings as required.	/	/
d0.06	2FF00C10	Tuning_Start	If the variable is set to 11, auto tuning starts. All input signals are neglected during auto tuning. The variable is automatically changed to 0 after auto tuning is completed. Sets the variable to other values to end auto tuning.	0	/

Parameter List: Group F001 (To Set Real-Time Display Data)

Numeric Display	Internal Address	Variable Name	Displayed Content
d1.00	2FF00F20	Soft_Version_LED	Software version of numeric display

Numeric Display	Internal Address	Variable Name	Displayed Content
d1.01	2FF70020	Time_Driver	Accumulated working time of the driver (S)
d1.02	2FF01008	Motor_IIt_Rate	Ratio of real iit to the maximum iit of a motor
d1.03	60F61210	Motor_IIt_Real	Actual data of motor overheat protection The formula of conversion between display value and actual current(Average value): $I_{rms} = \frac{\sqrt{Motor\_IIt\_Real*512}}{2047} * \frac{I_{peak}}{\sqrt{2}}$ I_{peak} is the max. peak value of the output current of driver.
d1.04	2FF01108	Driver_IIt_Rate	Ratio of real iit to the maximum iit of a driver
d1.05	60F61010	Driver_IIt_Real	Actual data of driver overheat protection
d1.06	2FF01208	Chop_Power_Rate	Ratio of actual power to rated power of a braking resistor
d1.07	60F70D10	Chop_Power_Real	Actual power of a braking resistor
d1.08	60F70B10	Temp_Device	Temperature of a driver (°C)
d1.09	60790010	Real_DCBUS	Actual DC bus voltage
d1.10	60F70C10	Ripple_DCBUS	Fluctuating value of the bus voltage (Vpp)
d1.11	60FD0010	Din_Status	Status of an input port
d1.12	20101410	Dout_Status	Status of an output port
d1.13	25020F10	Analog1_out	Filter output of external analog signal 1
d1.14	25021010	Analog2_out	Filter output of external analog signal 2
d1.15	26010010	Error_State	Error state
d1.16	26020010	Error_State2	Error state word 2
d1.17	60410010	Status_Word	Driver status word bit0：Ready to switch on bit1：Switch on bit2：Operation enable bit3：Falt bit4：Voltage Disable bit5：Quick Stop bit6：Switch on disable bit7：Warning bit8：Reserved bit9：Reserved bit10：Target reach bit11：Internal limit active bit12：Step.Ach./V=0/Hom.att. bit13：Foll.Err/Res.Hom.Err. bit14：Commutation Found bit15：Referene Found
d1.18	60610008	Operation_Mode_Buff	Efficient working mode of a driver
d1.19	60630020	Pos_Actual	Actual position of a motor
d1.20	60FB0820	Pos_Error	Position following error
d1.21	25080420	Gear_Master	Count of input pulses before electronic gear
d1.22	25080520	Gear_Slave	Count of executed pulses after electronic gear
d1.23	25080C10	Master_Speed	Pulse speed entered by the master axis (pulse/mS)
d1.24	25080D10	Slave_Speed	Pulse speed of the slave axis (pulse/mS)
d1.25	606C0010	Real_Speed_RPM	Real speed (rpm) Internal sampling time: 200 mS

Numeric Display	Internal Address	Variable Name	Displayed Content
d1.26	60F91910	Real_Speed_RPM2	Real speed (0.01 rpm) Internal sampling time: 200 mS
d1.27	60F91A10	Speed_1mS	Speed data (inc/1 mS) Internal sampling time: 1 mS
d1.28	60F60C10	CMD_q_Buff	Internal effective current instruction
d1.29	60F61710	I_q	Actual current The formula of conversion between display value andactual current: $I_{rms} = \frac{I\_q}{2047} * \frac{I_{peak}}{\sqrt{2}}$ I_{peak} is the max. peak value of the output current of driver.
d1.30	60F90E10	K_Load	Load parameter
d1.31	30100420	Z_Capture_Pos	Position data captured by encoder index signals

Parameter List: Group F002 (To Set Control Loop Parameters)

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
d2.00	2FF00108	Store_Loop_Data	1: Stores all setup parameters except those of a motor 10: Initializes all setup parameters except those of a motor	0	/
d2.01	60F90110	Kvp	Sets the response speed of velocity loop		0～32767
d2.02	60F90210	Kvi	Time used to adjust speed control to compensate minor errors		0～16384
d2.03	60F90308	Notch_N	Notch/filtering frequency setting for a velocity loop, used to set the frequency of the internal notch filter, so as to eliminate the mechanical resonance produced when the motor drives the machine. The formula is F=Notch_N*10+100. For example, if the mechanical resonance frequency is F = 500 Hz, the parameter should be set to 40.	45	0～90
d2.04	60F90408	Notch_On	Enable or disable the notch filter 0: Disable the trap filter 1: Enable the trap filter	0	/
d2.05	60F90508	Speed_Fb_N	You can reduce the noise during motor operation by reducing the feedback bandwidth of velocity loop. When the set bandwidth becomes less, the motor responds slower. The formula is F=Speed_Fb_N*20+100. For example, to set the filter bandwidth to "F = 500 Hz", you need to set the parameter to 20.		0～45
d2.06	60F90608	Speed_Mode	0: Speed response after traveling through a low-pass filter 1: Direct speed response without filtering 2: Feedback on output feedback	0	/

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
d2.07	60FB0110	Kpp	Proportional gains on position loop Kpp	1000	0～16384
d2.08	60FB0210	K_Speed_FF	0 indicates no feedforward, and 256 indicates 100% feedforward	256	0～256
d2.09	60FB0310	K_Acc_FF	The data is inversely proportional to the feedforward	7FF.F	32767～10
d2.10	2FF00610	Profile_Acce_16	To set trapezoidal acceleration (rps/s) in the “3” and “1” modes	610	0～2000
d2.11	2FF00710	Profile_Dece_16	To set trapezoidal deceleration (rps/s) in the “3” and “1” modes	610	0～2000
d2.12	60F60110	Kcp	To set the response speed of the current loop and this parameters does not require adjusting	/	/
d2.13	60F60210	Kci	Time used to adjust current control to compensate minor errors	/	/
d2.14	60730010	CMD_q_Max	Indicates the maximum value of current instructions	/	/
d2.15	60F60310	Speed_Limit_Factor	The factor that limits the maximum speed in the torque mode $\begin{cases} F_{Actual\ torque} = F_{Set\ torque} \cdots\cdots V_{Actual\ speed} < V_{Maximum\ speed} \\ F_{Actual\ torque} = F_{Set\ torque} - N*(V_{Actual\ speed} - V_{Maximum\ speed}) \cdots V_{Actual\ speed} > V_{Maximum\ speed} \end{cases}$ V the maximum speed complies with d2.24 Max_Speed_RPM parameter settings	10	0～1000
d2.16	607E0008	Invert_Dir	Runs polarity reverse 0: Counterclockwise indicates the forward direction 1: Clockwise indicates the forward direction	0	/
d2.17	60F90E10	K_Load	Indicates load parameters	/	20～15000
d2.18	60F90B10	Kd_Virtual	Indicates the kd of observers	1000	0～32767
d2.19	60F90C10	Kp_Virtual	Indicates the kp of observers	1000	0～32767
d2.20	60F90D10	Ki_Virtual	Indicates the ki of observers	0	0～16384
d2.21	60F91010	Sine_Amplitude	Proper increase in this data will reduce the tuning error, but machine vibration will become severer. This data can be adjusted properly according to actual conditions of machines. If the data is too small, the auto tuning error becomes greater, or even causes a mistake.	64	0～1000
d2.22	60F91110	Tuning_Scale	It is helpful to reduce the auto tuning time by reducing the data, but the result may be unstable.	128	0～16384
d2.23	60F91210	Tuning_Filter	Indicates filter parameters during auto-tuning	64	1～1000
d2.24	60800010	Max_Speed_RPM	Limits the maximum rotation speed of motors	5000	0～6000

Parameter List: Group F003 (To Set Input/Output & Pattern Operation Parameters)

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
d3.00	2FF00108	Store_Loop_Data	1: Stores all setup parameters except motors 10: Initializes all setup parameters except motors	0	/
d3.01	20100310	Din1_Function	000.1: Driver enable 000.2: Driver fault reset 000.4: Operation mode control 000.8: P control for velocity loop 001.0: Position positive limit 002.0: Position negative limit 004.0: Homing signal 008.0: Reverse speed demand 010.0: Internal speed control 0 020.0: Internal speed control 1 800.1: Internal speed control 2 040.0: Internal position control 0 080.0: Internal position control 1 800.2: Internal position control 2 800.4 Multi Din 0 800.8 Multi Din 1 801.0 Multi Din 2 802.0 Gain switch 0 804.0 Gain switch 1 100.0: Quick stop 200.0: Start homing 400.0: Activate command Note:DinX_Function(X is 1-7) is used to define the function of digital inputs.	000.1	/
d3.02	20100410	Din2_Function		000.2	/
d3.03	20100510	Din3_Function		000.4	/
d3.04	20100610	Din4_Function		000.8	/
d3.05	20100710	Din5_Function		001.0	/
d3.06	20100810	Din6_Function		002.0	/
d3.07	20100910	Din7_Function		004.0	/
d3.08	2FF00D10	Dio_Polarity	Sets IO polarity	0	/
d3.09	2FF00810	Dio_Simulate	Simulates input signals, and enforce output signals for outputting	0	/
d3.10	20000008	Switch_On_Auto	Automatically locks motors when drivers are powered on 0: No control 1: Automatically locks motors when drivers are powered on	0	/
d3.11	20100F10	Dout1_Function	000.1: Ready 000.2: Error 000.4: Position reached 000.8: Zero velocity 001.0: Motor brake 002.0:Velocity reached 004.0: Index 008.0: The maximum speed obtained in the torque mode 010.0: PWM ON 020.0: Position limiting	000.1	/
d3.12	20101010	Dout2_Function		000.0	/
d3.13	20101110	Dout3_Function		00a.4	/

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
d3.14	20101210	Dout4_Function	040.0: Reference found 080.0: Reserved 100.0: Multi Dout 0 200.0: Multi Dout 1 400.0: Multi Dout 2 Note:DoutX_Function(X is 1-5) is used to define functions of the digital outputs.	000.8	/
d3.15	20101310	Dout5_Function		000.0	/
d3.16	20200D08	Din_Mode0	If a digital input is defined as Operation mode control,then this operation mode is selected when the input signal is invalid	-4	/
d3.17	20200E08	Din_Mode1	If a digital input is defined as Operation mode control,then this operation mode is selected when the input signal is valid	-3	/
d3.18	20200910	Din_Speed0_RPM	Multi-speed control: 0 [rpm]	0	/
d3.19	20200A10	Din_Speed1_RPM	Multi-speed control: 1 [rpm]	0	/
d3.20	20200B10	Din_Speed2_RPM	Multi-speed control: 2 [rpm]	0	/
d3.21	20200C10	Din_Speed3_RPM	Multi-speed control: 3 [rpm]	0	/
d3.22	25020110	Analog1_Filter	Used to smooth the input analog signals F (Filter Frequency) = 4000/ (2π* Analog1_Filter) T (Time Constant) = Analog1_Filter/4000 (S)	5	1～127
d3.23	25020210	Analog1_Dead	Sets dead zone data for external analog signal 1	0	0～8192
d3.24	25020310	Analog1_Offset	Sets offset data for external analog signal 1	0	-8192～8192
d3.25	25020410	Analog2_Filter	Used to smooth the input analog signals Filter frequency: f=4000/(2π* Analog1_Filter) Time Constant: T = Analog1_Filter/4000 (S)	5	1～127
d3.26	25020510	Analog2_Dead	Sets dead zone data for external analog signal 2	0	0～8192
d3.27	25020610	Analog2_Offset	Sets offset data for external analog signal 2	0	-8192～8192
d3.28	25020708	Analog_Speed_Con	Chooses analog-speed channels 0: Invalid analog channel 1: Valid analog channel 1 (AIN1) 2: Valid analog channel 2 (AIN2) Valid mode -3 and 3	0	/
d3.29	25020A10	Analog_Speed_Factor	Sets the proportion between analog signals and output speed	1000	/
d3.30	25020808	Analog_Torque_Con	Chooses analog-torque channels 0: Invalid analog channel 1: Valid analog channel 1 (AIN1)	0	/

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
			2: Valid analog channel 2 (AIN2) Valid mode 4		
d3.31	25020B10	Analog_Torque_Factor	Sets the proportion between analog signals and output speed (current)	1000	/
d3.32	25020908	Analog_MaxT_Con	0: No control 1: Max. torque controlled by AIN 1 2: Max. torque controlled by AIN 2	0	/
d3.33	25020C10	Analog_MaxT_Factor	Indicates the max torque factor on analog signal control	8192	/
d3.34	25080110	Gear_Factor	Indicates the numerator to set electronic gears when the operation mode is -4	1000	-32767～32767
d3.35	25080210	Gear_Divider	Indicates the denominator to set electronic gears when the operation mode is -4	1000	1～32767
d3.36	25080308	PD_CW	Pulse mode control 0...CW/CCW 1...Pulse/Direction 2...Incremental encoder Note:After changing this parameter,it needs to save by d2.00/d3.00/d5.00 and then reboot driver.	1	/
d3.37	25080610	PD_Filter	To flat the input pulse. Filter frequency: f=1000/(2π* PD_Filter) Time constant: T = PD_Filter/1000 Unit: S Note: If you adjust this filter parameter during the operation, some pulses may be lost.	3	1～32767
d3.38	25080810	Frequency_Check	Indicates the limitation on pulse input frequency (k Hz)	600	0～600
d3.39	25080910	PD_ReachT	Indicates the position reached time window in the pulse mode Unit: mS	10	0～32767
d3.40	2FF10108	Din_Position_Select_L	Select which internal position will be set.(The range of L is 0-7) Din_Pos0 Din_Pos1 Din_Pos2 Din_Pos3 Din_Pos4 Din_Pos5 Din_Pos6 Din_Pos7	0	
d3.41	2FF10210	Din_Position_M	Refer to d3.42	0	
d3.42	2FF10310	Din_Position_N	The position of internal position set in Din_Position_Select_L Din_Pos = Din_Position_M*10000+Din_Position_N	0	
d3.43	20200F10	Din_Control_Word	Absolute positioning/Relative positionin gsetting 2F:Absolute positioning	2F	

Numeric Display	Internal Address	Variable Name	Meaning	Default Value	Range
			4F:Relative positioning Note:This parameter needs to save and reboot driver after change.		
d3.44	20201810	Din_Speed4_RPM	Multi-speed control: 4 [rpm]	0	
d3.45	20201910	Din_Speed5_RPM	Multi-speed control: 5 [rpm]	0	
d3.46	20201A10	Din_Speed6_RPM	Multi-speed control: 6 [rpm]	0	
d3.47	20201B10	Din_Speed7_RPM	Multi-speed control: 7 [rpm]	0	

Parameter List: Group F004 (To Set Motor Parameters)

Numeric display	Internal Address	Variable Name	Meaning
d4.00	2FF00308	Store_Motor_Data	1: Stores the set motor parameters
d4.01	64100110	Motor_Num	Host computer (ASCII code) numerical display (hexadecimal) "00"..... ....303.0 About the motor number please refer to chapter 6.1.1. Note: 1.Set the motor parameters refer to chapter 6 before operating. 2.It must use capital letter when set this parameter by PC. 3.It needs to save by d4.00 and reboot driver after changing this parameter.
d4.02	64100208	Feedback_Type	Type of encoders 001.1: Differential ABZ and differential UVW signals 001.0: Differential ABZ and UVW signals of TTL 000.1: ABZ of TTL and differential UVW signals 000.0: ABZ of TTL and UVW signals of TTl
d4.03	64100508	Motor_Poles	Number of motor poles pairs [2p]
d4.04	64100608	Commu_Mode	Searching excitation mode
d4.05	64100710	Commu_Curr	Searching excitation current [dec]
d4.06	64100810	Commu_Delay	Delay in searching excitation [mS]
d4.07	64100910	Motor_IIt_I	Indicates current settings on overheat protection of motors Ir[Arms]*1.414*10
d4.08	64100A10	Motor_IIt_Filter	Indicates time settings on overheat protection of motors Time: N*256/1000 Unit: S
d4.09	64100B10	Imax_Motor	Indicates max peak current of motors I[Apeak]*10
d4.10	64100C10	L_Motor	Indicates phase inductance of motors L[mH]*10
d4.11	64100D08	R_Motor	Indicates phase resistance of motors

Numeric display	Internal Address	Variable Name	Meaning
			R[Ω]*10
d4.12	64100E10	Ke_Motor	Indicates the reverse electromotive force of motors Ke[Vp/krpm]*10
d4.13	64100F10	Kt_Motor	Indicates the torque coefficient of motors Kt[Nm/Arms]*100
d4.14	64101010	Jr_Motor	Indicates the rotor inertia of motors Jr[kgm^2]*1 000 000
d4.15	64101110	Brake_Duty_Cycle	Indicates the duty cycle of contracting brakes 0~2500[0…100%]
d4.16	64101210	Brake_Delay	Indicates the delay time of contracting brakes Default value: 150 ms
d4.17	64101308	Invert_Dir_Motor	Indicates the rotation direction of motors
d4.18	64101610	Motor_Using	Current using motor type. **PC Software Numeric Display Model** "K0"...................304.B.......SMH60S-0020-30 "K1"..................314.B…….SMH60S-0040-30 "K2"..................324.B…….SMH80S-0075-30 "K3"..................334.B…….SMH80S-0100-30 "K4"..................344.B……SMH110D-0105-20 "K5"..................354.B……SMH110D-0125-30 "K6".................364.B…......SMH110D-0126-20 "K7"……….......374.B…….SMH110D-0126-30 "K8"…………...384.B.........SMH110D-0157-30 "K9".................394.B…....SMH110D-0188-30 KB"……..……...424.B…....SMH130D-0105-20 “KC"…………...434.B…….SMH130D-0157-20 “KD"…………...444.B…….SMH130D-0210-20 “KE"…………...454.B…....SMH150D-0230-20 "S0"………...305.3…..130D-0105-20AAK-2LS "S1".............315.3…..130D-0157-20AAK-2LS "S2"………...325.3….130D-0157-15AAK-2LS "S3".............335.3….130D-0200-20AAK-2HS "S4".............345.3….130D-0235-15AAK-2HS "F8"………...384.6…..85S-0045-05AAK-FLFN "E0".............304.5……….SME60S-0020-30 "E1".............314.5…….......SME60S-0040-30 "E2"................324.5………….SME80S-0075-30

Parameter List: Group F005 (To Set Driver Parameters)

Numeric Display	Internal Address	Variable Name	Meaning	Default Value
d5.00	2FF00108	Store_Loop_Data	1: Stores all control parameters except motor parameters 10: Initializes all control parameters except motor parameters	0
d5.01	100B0008	ID_Com	Station No. of Drivers Note: To change this parameter, you need to save it with the address	1

			"d5.00", and restart it later.	
d5.02	2FE00010	RS232_Bandrate	Set the baud rate of RS232 port 540 19200 270 38400 90 115200 Note: To change this parameter, you need to save it with the address "d5.00", and restarts it later.	270
d5.03	2FE10010	U2BRG	Sets the baud rate of RS232 port 540 19200 270 38400 90 115200 You need not restart it,but it can't be saved.	270
d5.04	60F70110	Chop_Resistor	Indicates the values of braking resistors	0
d5.05	60F70210	Chop_Power_Rated	Indicates the nominal power of a braking resistor	0
d5.06	60F70310	Chop_Filter	Indicates the time constant of a braking resistor Time: N*256/1000 Unit: S	60
d5.07	25010110	ADC_Shift_U	Indicates data configuration of U phase shift. Note:Factory parameters	/
d5.08	25010210	ADC_Shift_V	Indicates data configuration of V phase shift Note:Factory parameters	/
d5.09	30000110	Voltage_200	ADC original data when DC bus voltage is 200 V Note:Factory parameters	/
d5.10	30000210	Voltage_360	ADC original data when DC bus voltage is 360 V Note:Factory parameters	/
d5.11	60F60610	Comm_Shift_UVW	Indicates the excitation pointer of a motor Note:Factory parameters	/
d5.12	26000010	Error_Mask	Indicates error masks Note:Factory parameters	FFF.F
d5.13	60F70510	RELAY_Time	Indicates the relay operating time of capacitor short-circuits Unit: mS Note:Factory parameters	150
d5.14	2FF00408	Key_Address_F001	Sets numeric display data	/
d5.15	65100B08	RS232_Loop_Enable	0：1 to 1 1：1 to N	0
d5.16	2FFD0010	User_Secret	User password.16bits.	0～65535

Chapter 7 Operation on Input/Output Ports

KINCO FD2S servo driver has 7 digital input ports (a digital input port can receive high-level or low-level signals, depending on whether high-level or low-level signals are chosen at the COM terminal) and 5 digital output ports,OUT1-OUT4 ports can drive 100 mA load, and BR port can drive 500 mA load, and can directly drive the internal contracting brake device. You can freely configure all functions on digital input/output ports according to application requirements.

7.1 Digital Input

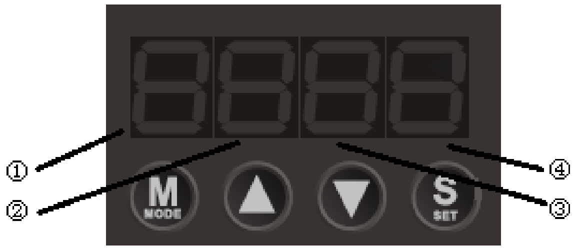
7.1.1 Polarity Control on Digital Input Signals

Note:all the digital inputs are normally open by default.

Table 7-1 Simplified IO polarity setting variables

Numeric Display	Variable Name	Meaning
d3.08	Dio_Polarity	Sets IO polarity

Table 7-2 Polarity setting methods for digital input signals



④	②	③	④
Input/output port selection 0: Output port 1: Input port	Channel selection Input: 1-8 Output: 1-7	Reserved	0：The inputs are normally close 1：The inputs are normally open Others:Check the current status

Example 7-1: Polarity Setting for Digital Input Signal DIN1

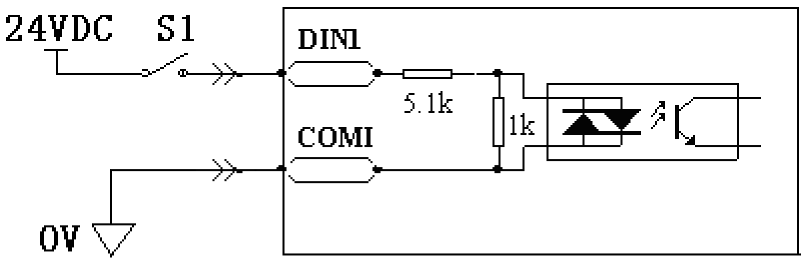


Fig.7-1 Polarity setting for digital input signal DIN1

7.1.1.1 Use panel to change the polarity

Table 7-3 Polarity setting for digital input signal DIN1

④	②	③	④
Input/output port selection Set to 1 (input port selected)	Channel selection Set to 1 (DIN 1 selected)	Reserved	0: DIN1 is enabled when S1 opens 1: DIN1 is enabled when S1 closes

Namely, if d3.08 is set to “110.0”, it indicates that DIN1 is normally close.If d3.08 is set to “110.1”, it indicates that DIN1 is normally open.

7.1.1.2：Use PC software to change polarity

Use the PC software to connect to FD2S Servo and then open I/O port.The LED under polarity are green,it indicates that the inputs are normally open.As following figure,if you change the LED of DIN5 and DIN6 into red,it indicates that DIN5 and DIN6 are normally close.

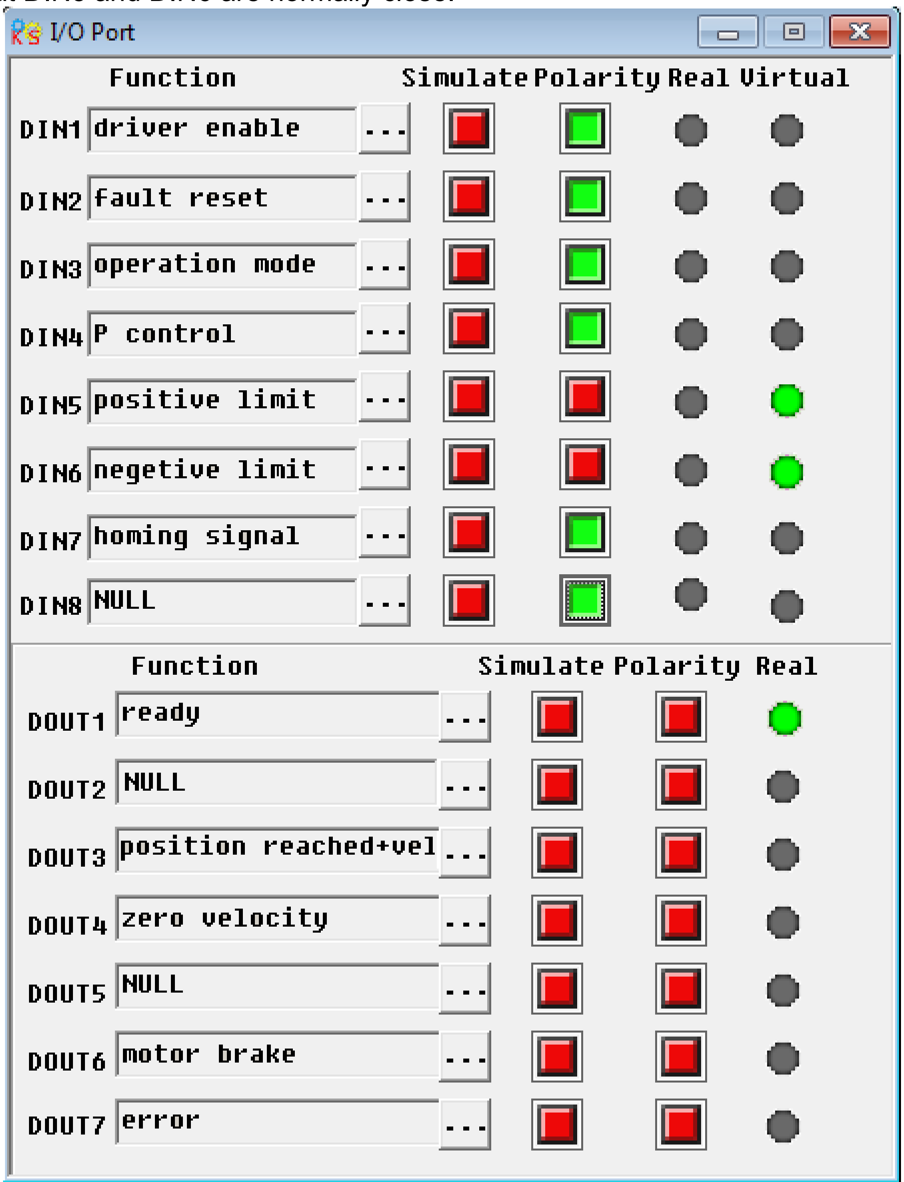


Fig.7-2 Digital I/O in PC software

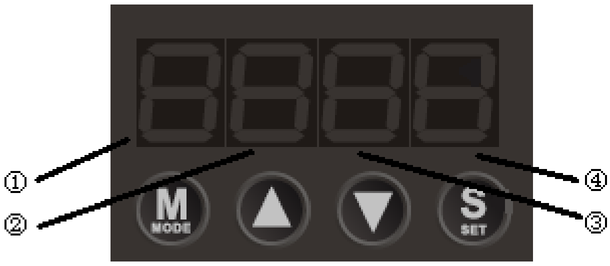
7.1.2 Simulation of Digital Input Signals

Table 7-4 IO simulation variable

Numeric Display	Variable Name	Meaning
d3.09	Dio_Simulate	Simulates input signals, and enforces output signals for outputting

Dio_Simulate (IO simulation) is for the software to simulate inputting of a valid signal. “1” indicates that the input signal is valid, and “0” indicates that the input signal is invalid.

Table 7-5 Settings on simulation of digital input signals



①	②	③	④
Input/output port selection 0: output port 1: input port	Channel selection Input: 1-8 Output: 1-7	Reserved	0: No input signal is simulated, and no output signal is compulsorily outputted 1: Input signal is simulated, and output signal is outputted compulsorily Other: Check the current status

Example 7-2: Simulate digital input DIN1

Table 7-6: Simulate digital input DIN1

①	②	③	④
Input/output port selection Set to 1 (input port selected)	Channel selection Set to 1 (DIN 1 selected)	Reserved	0: Invalid DIN1 simulation 1: Valid DIN1 simulation

Namely, if d3.09 is set to “110.0”, it indicates that no DIN1 input signals are simulated; if d3.09 is set to “110.1”, it indicates that DIN1 input signals are simulated.

7.1.3 Status Display of Digital Input Signals

Table 7-7 Variables for status display of digital input signals

Numeric Display	Variable Name	Meaning
d1.11	Din_Status	Status of input ports

Din_Status (hexadecimal) is used to display the status of the actually input external signals in real time.

7.1.4 Addresses & Functions of Digital Input Signals

Table 7-8 Addresses & default functions of digital input signals

Numeric Display	Variable Name	Meaning	Default Value

<table>
<tr><td>d3.01</td><td>Din1_Function</td><td rowspan="7">000.1: Driver enable
000.2: Driver fault reset
000.4: Operation mode control
000.8: P control for velocity loop
001.0: Position positive limit
002.0: Position negative limit
004.0: Homing signal
008.0: Reverse speed demand
010.0: Internal speed control 0
020.0: Internal speed control 1
800.1: Internal speed control 2
040.0: Internal position control 0
080.0: Internal position control 1
800.2: Internal position control 2
800.4 Multi Din 0
800.8 Multi Din 1
801.0 Multi Din 2
802.0 Gain switch 0
804.0 Gain switch 1
100.0: Quick stop
200.0: Start homing
400.0: Activate command

Note:DinX_Function(X is 1-7) is used to define the function of digital inputs.</td><td>000.1 (Driver enable)</td></tr>
<tr><td>d3.02</td><td>Din2_Function</td><td>000.2 (Driver fault reset)</td></tr>
<tr><td>d3.03</td><td>Din3_Function</td><td>000.4 (Operation mode control)</td></tr>
<tr><td>d3.04</td><td>Din4_Function</td><td>000.8 (P control for velocity loop)</td></tr>
<tr><td>d3.05</td><td>Din5_Function</td><td>001.0 (Position positive limit)</td></tr>
<tr><td>d3.06</td><td>Din6_Function</td><td>002.0 (Position negative limit)</td></tr>
<tr><td>d3.07</td><td>Din7_Function</td><td>004.0 (Homing signal)</td></tr>
</table>

Table 7-9 Meaning of defined functions of digital input signals

Function	Meaning
Disable	Used to cancel the function of this digital input.
Driver enable	By default, the driver enable signal is valid, and the motor shaft is locked.
Driver fault reset	Signals on the rising edge are valid, and alarms are cleared.
Operation mode control	To switch between two operation modes. You can freely determine the operation modes corresponding to valid signals and invalid signals by performing settings through d3.16 Din_Mode0 (choose 0 for operation mode) of Group F003 and Din_Mode1 (choose 1 for operation mode) of Group F003.
P control for velocity loop	Indicates the control on stopping integration in velocity loop. The control is applied in the occasion where high-speed system stop occurs, but overshooting is not expected. Note: In the “-3” mode, if the signal is valid, fixed errors occur between the actual speed and target speed.
Position positive limit	Indicates the limit of forward running of motors (normally closed contact by default). By default, the driver regards position positive limits as valid, and polarity can be modified to adjust to normally open switches.
Position negative limit	Indicates the limit of inverted running of motors (normally closed contact by default). By default, the driver regards position negative limits as valid, and polarity can be modified to adjust to normally open switches.
Homing signal	To find origins of motors.
Reverse speed demand	To reverse the target speed in the speed mode ("-3" or “3”).
Internal speed control 0	To control internal multiple speeds. Note: For details, see Section 7.5 Internal Multi-Speed Control.
Internal speed control 1	
Internal speed control 2	

Internal position control 0 Internal position control 1 Internal position control 2	To control internal multiple positions. Note: For details, see Section 7.4 Internal Multi-Position Control.
Multi Din 0 Multi Din 1 Multi Din 2	To switch multiple electronic gear
Gain switch 0 Gain switch 1	To switch multiple gain parameters(P-gain of velocity loop,i-gain of velocity loop,p-gain of position loop)
Quick stop	When the signal is valid, the motor shaft releases. After the signal is removed, the driver requires re-enabling.
Start homing	When the rising edge of the signal is detected,it will start homing command.
Activate command	When the rising edge of the signal is detected,it will activate the internal position control

Example 7-3: Driver Enable Setting

Requirement: The “driver enable” function is controlled through an external digital output port. In this example, the digital input port DIN1 is defined as the “driver enable” function. Table 7-10 shows the setup method.

Table 7-10 Digital Input Port DIN1 Defined as the “Driver Enable” Function

Numeric Display	Variable Name	Parameter Settings
d3.01	Din1_Function	Set to 000.1
d3.00	Store_Loop_Data	Set to 1

Note: Any digital output of DIN1-7 can be defined as “driver enable”, and is set to 000.1, that is, bit 0 is valid.

Requirement: Enable the function of automatically powering on the driver by setting internal parameters in drivers instead of external digital input ports. Table 7-11 describes the setup method.

Table 7-11 Enabling the function of automatically powering on the driver by setting internal parameters in drivers

Numeric Display	Variable Name	Parameter Settings
d3.01- d3.07	DinX_ Function (1~7)	None of the digital input port can be set to 000.1, that is, the Enable function is not controlled by any digital input port.
d3.10	Switch_On_Auto	Set to 1
d3.00	Store_Loop_Data	Set to 1

Users can also use PC software to define I/O functions.Open the I/O port menu,click the button in red box as shown in following figure,then select the required function.

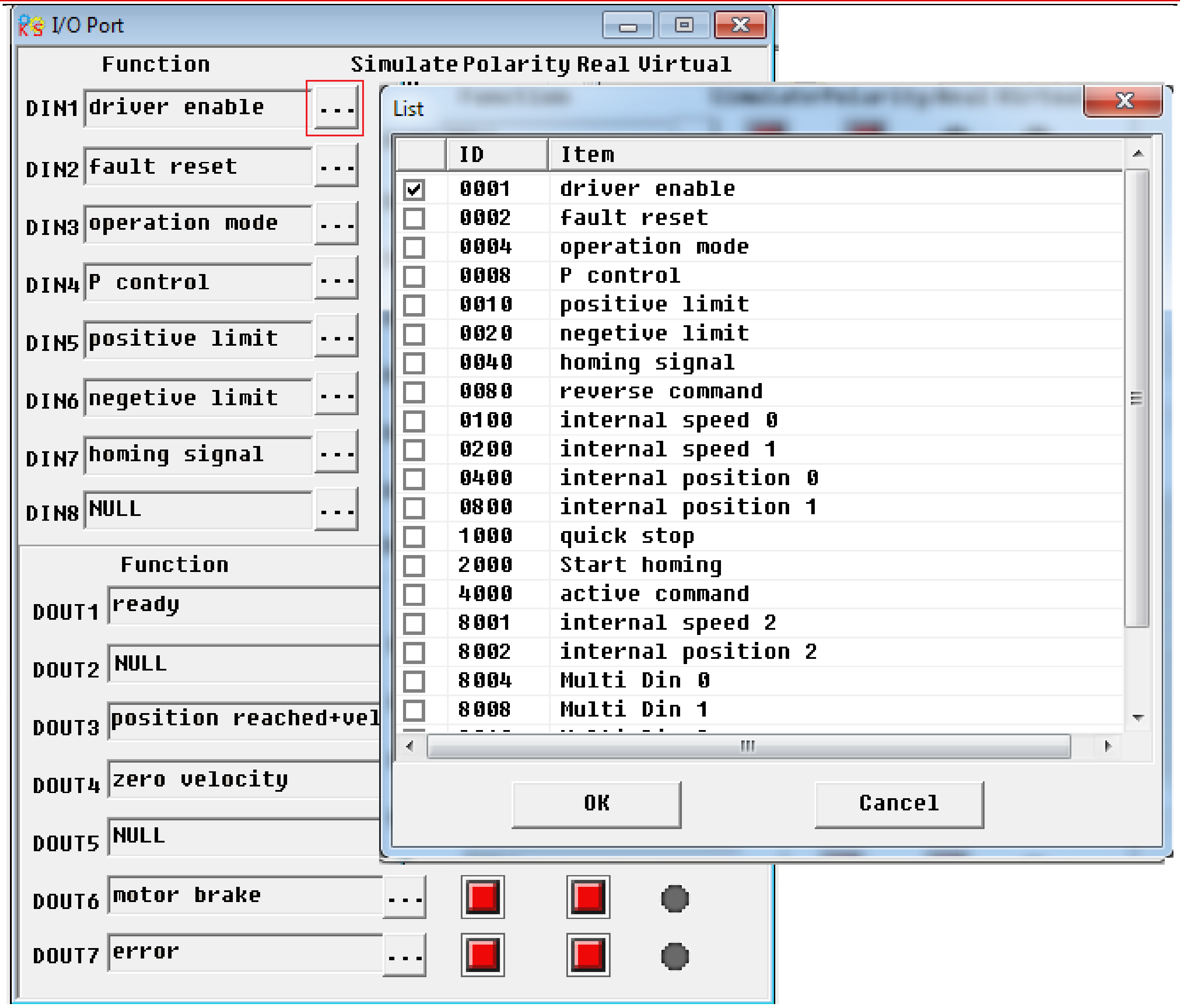


Fig.7-2 Set digital I/O function in PC software

Example 7-4: Disabling Position Positive/Negative Limit Settings

When the driver is delivered, the DIN5 of the motor is the position positive limit and DIN6 is the position negative limit by default. If there are no external position positive/negative limit switches, this function must be disabled so that the servo driver can work properly. Table 7-12 describes the setup method.

Table 7-12: Disabling position positive/negative limit settings

Numeric Display	Variable Name	Parameter Settings
d3.05	Din5_Function	Change the default value 001.0 (position positive limit) to 000.0
d3.06	Din6_Function	Change the default value 002.0 (position negative limit) to 000.0
d3.00	Store_Loop_Data	Set to 1

Example 7-5: Operation Mode Control on Drivers

Requirements: Defines the input port DIN3 as the operation mode control on drivers, and the operation mode is "-4" (pulse control mode) when DIN3 fails, and is "-3" (instantaneous speed mode) when DIN3 is valid. Table 7-13 describes the setup method.

Table 7-13 Settings on operation mode control on drivers

Numeric Display	Variable Name	Parameter Settings
d3.03	Din3_Function	Set to 000.4
d3.16	Din_Mode0	Set to 0.004 (-4)
d3.17	Din_Mode1	Set to 0.003 (-3)
d3.00	Store_Loop_Data	Set to 1

Note: If the driver is required to operate in some mode with power on, one of the digital input must be set as function "Operation Mode Control". Then you can set the operation modes that require in the parameters d3.16 or d3.37 in Group F003.

7.1.5 Wirings of Digital Input Port

1. NPN wiring diagram (to the controller that supports low level output)

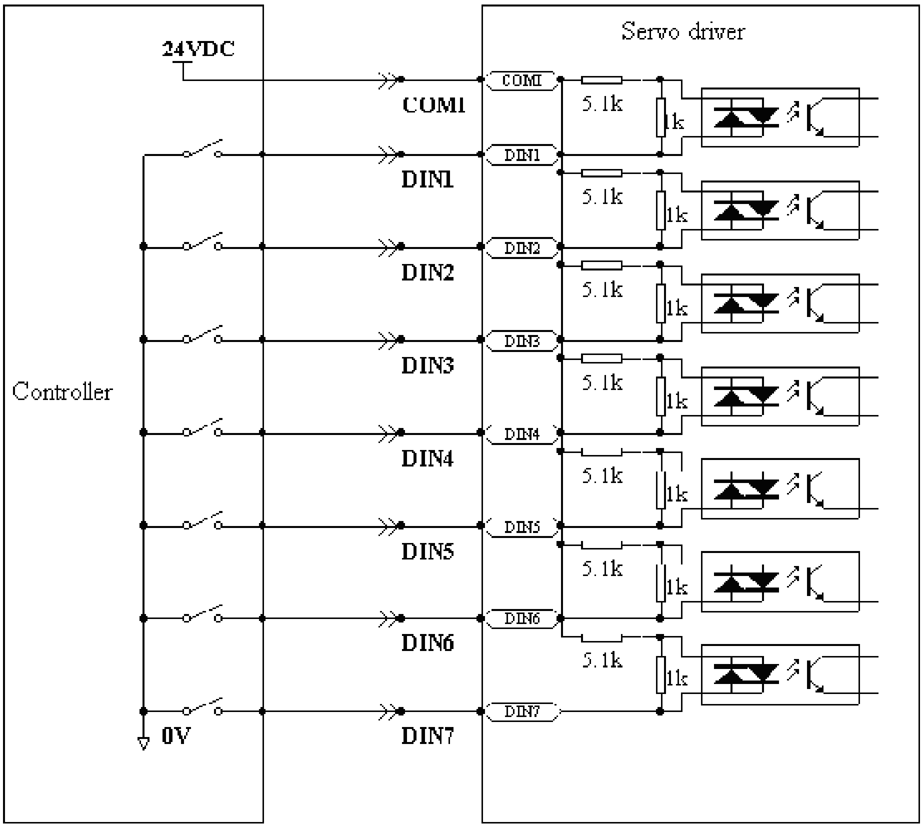


Fig.7-4 NPN wiring diagram (to the controller that supports low level output)

2. PNP wiring diagram (to the controller that supports high level output)

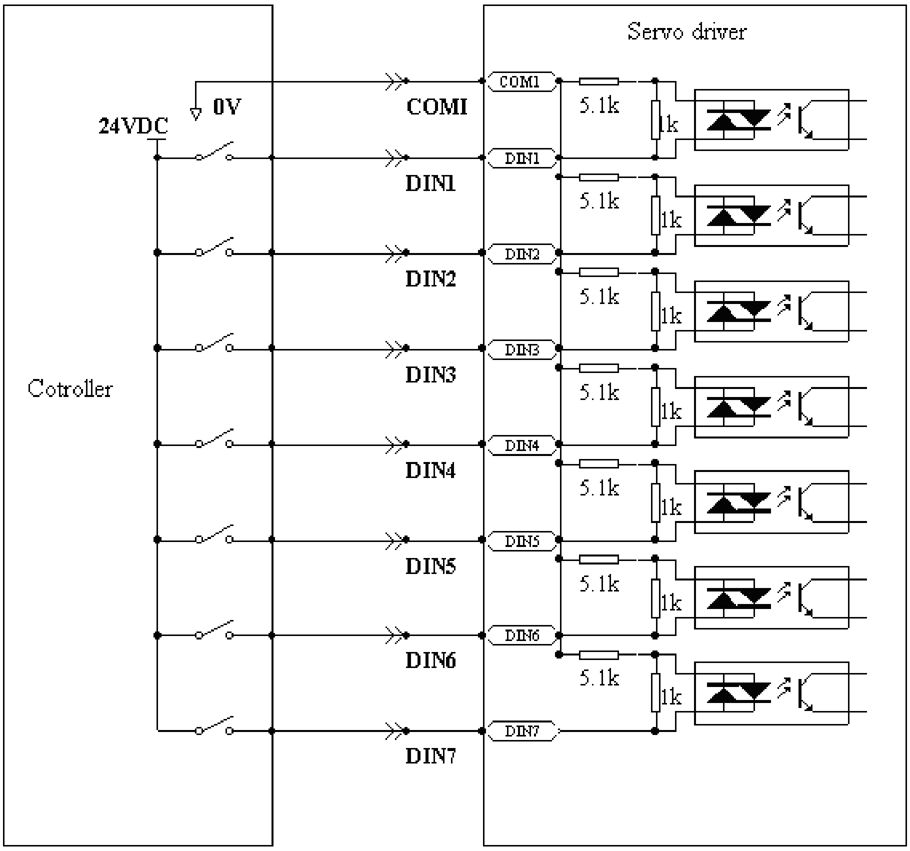


Fig.7-5 PNP wiring diagram (to the controller that supports high level output)

7.2 Digital Output

7.2.1 Polarity Control on Digital Output Signals

Note: All the digital output are normally open by default.

Table 7-14 Variables for setting simplified IO polarity

Numeric Display	Variable Name	Meaning
d3.08	Dio_Polarity	Sets IO polarity

Dio_Polarity (simplified IO polarity settings) is used to set the polarity of valid digital output signals. The number "1" indicates normally open, and "0" indicates normally close.Default is 1.

Example 7-6: Polarity setting for digital output OUT1

7.2.1.1：Use panel to change polarity

Table 7-15 Polarity setting for digital output OUT1(Default is ready function)

①	②	③	④
Input/output port selection Set to 0 (Output port selected)	Channel selection Set to 1 (OUT1 selected)	Reserved	0: OUT1 is normally close 1: OUT1 is normally open.

Namely, if d3.08 is set to "010.0", it indicates that OUT1 is normally close.If d3.08 is set to "010.1", it indicates that OUT1 is normally open.

7.2.1.2：Use PC software to change polarity,please refer to 7.1.1.2.

7.2.2 Simulation of Digital Output Signals（More details please refer to 7.1.2）

Table 7-16 IO simulation variables

Numeric Display	Variable Name	Meaning
d3.09	Dio_Simulate	Simulates input signals, and force the output signal

Dio_Simulate (IO simulation) is to simulate the output of a valid signal. The number “1” indicates that the output signal is valid, and “0” indicates that the output signal is invalid.

7.2.3 Status Display of Digital Output Signals

Table 7-17 Variables for status display of digital output signals

Numeric Display	Variable Name	Meaning
d1.12	Dout_Status	Status of an output port

Din_Status (hexadecimal) displays the status of actual external output signals in real time.

7.2.4 Addresses and Functions of Digital Output Signals

Table 7-18 Addresses and default functions of digital output signals

<table>
<tr><th>Numeric Display</th><th>Variable Name</th><th>Meaning</th><th>Default Value</th></tr>
<tr><td>d3.11</td><td>Dout1_Function</td><td rowspan="5">000.1: Ready
000.2: Error
000.4: Position reached
000.8: Zero velocity
001.0: Motor brake
002.0:Velocity reached
004.0: Index
008.0: The maximum speed obtained in the torque mode
010.0: PWM ON
020.0: Position limiting
040.0: Reference found
080.0: Reserved
100.0: Multi Dout 0
200.0: Multi Dout 1
400.0: Multi Dout 2</td><td>000.1 (Ready)</td></tr>
<tr><td>d3.12</td><td>Dout2_Function</td><td>000.2 (Error)</td></tr>
<tr><td>d3.13</td><td>Dout3_Function</td><td>00a.4 (Position reached/Velocity reached/Max. velocity limit)</td></tr>
<tr><td>d3.14</td><td>Dout4_Function</td><td>000.8 (Zero velocity)</td></tr>
<tr><td>d3.15</td><td>Dout5_Function</td><td>001.0 (Motor brake)</td></tr>
</table>

Table 7-19 Meanings of the functions defined by digital output signals

Function	Meaning
Disable	Cancel the function of this digital output
Ready	The driver is ready for operation.
Error	Alarm signals are output, indicating that the driver is faulty.
Position reached	In the “-4” mode of pulse control, the target position data keeps

	unchanged in the window (d3.39) of the time of reaching the target position, and position errors are within the window of reaching the target position.
Zero velocity	After the motor is enabled, it is outputted when the motor speed is 0.
Motor brake	The driver enables the motor, and contracting brake output is valid.
Velocity reached	In the “-3” or "3” internal speed control mode, signals are output after they reach the target speed.
Index	Z phase signal output (the speed should not be too high).
Max. velocity limit	In the “4” analog – torque mode, signals are output after the max restricted speed is reached.
PWM ON	The driver enables the motor.
Motor limiting	Motor is in the status of position limiting.
Reference found	Homing is finished.

Example 7-7: “Ready” settings

Requirement: The OUT1 is defined as the “Ready” function. For details on settings, see Table 7-19。

Table 7-20 “Ready” settings

Numeric Display	Variable Name	Parameter Settings
d3.11	Dout1_Function	Set to 000.1
d3.00	Store_Loop_Data	Set to 1

7.2.5 Wiring of Digital Output Port

1. Internal circuit diagram of digital output ports

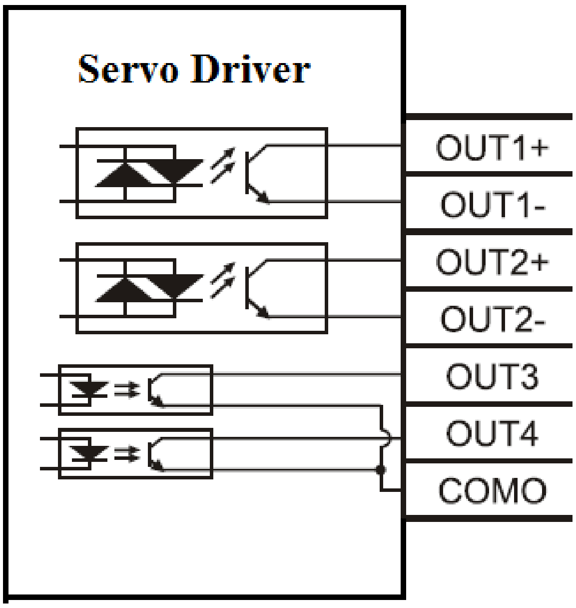


Fig.7-6 Internal circuit diagram of digital output

Note:

1.OUT3 and OUT4 use the same common terminal(COMO).

2.NPN Wiring Diagram（OUT1-OUT7 all support this）

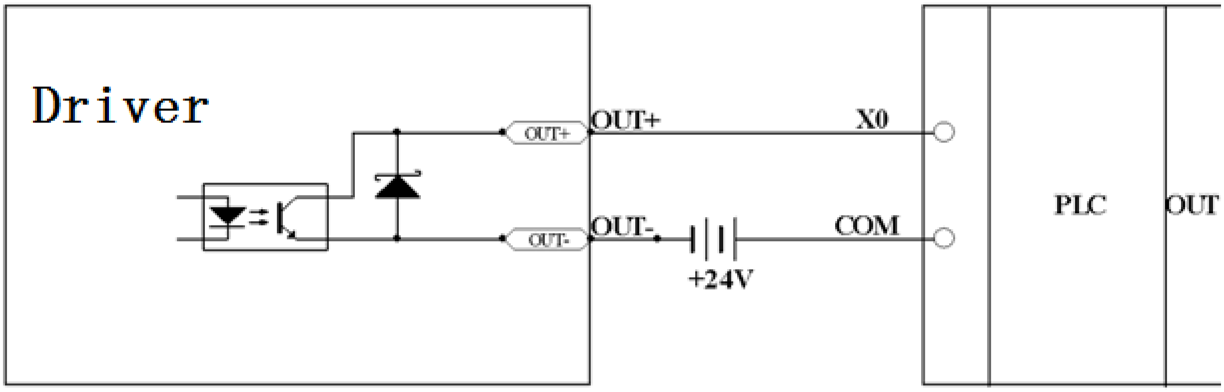


Fig.7-7 NPN wiring diagram (to controllers that support valid low level input)

3. PNP wiring diagram (Only OUT1,OUT2 and OUT7 support this wiring)

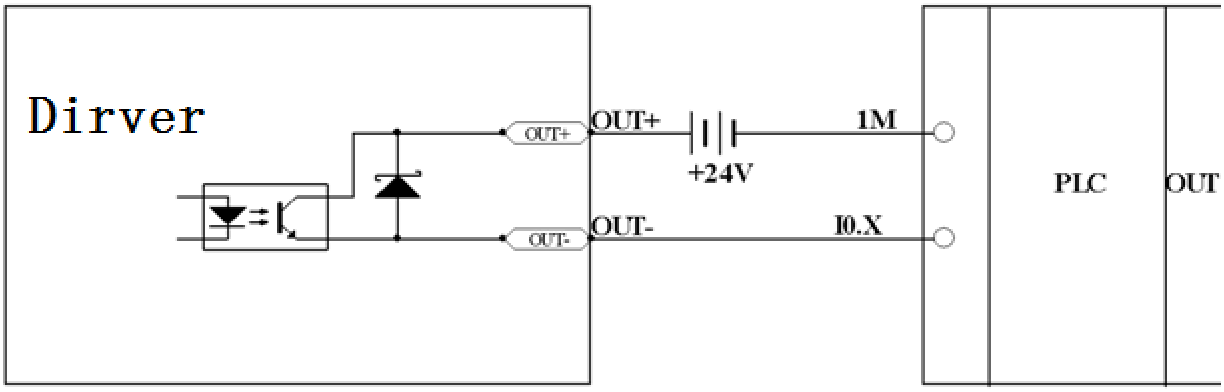


Fig.7-8 PNP wiring diagram (to controllers that support valid low level input)）

4. To connect a relay to the digital output port, do remember to connect a diode in inverse parallel, as shown in Fig.7-9.

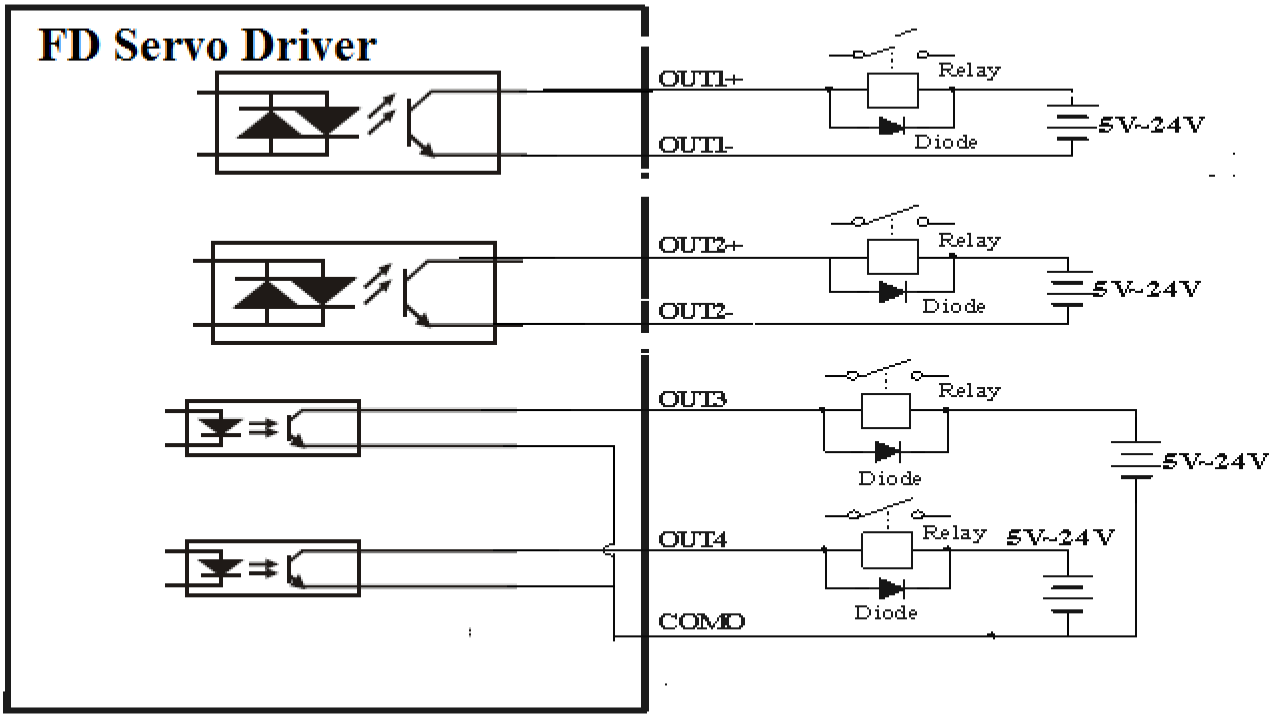


Fig.7-9 Connect a relay to the digital output port

Chapter 8 Operation Mode

8.1 Pulse Control Mode ("-4" Mode)

8.1.1 Wiring in Pulse Control Mode

1. Wiring diagram of FD2S driver in pulse control mode

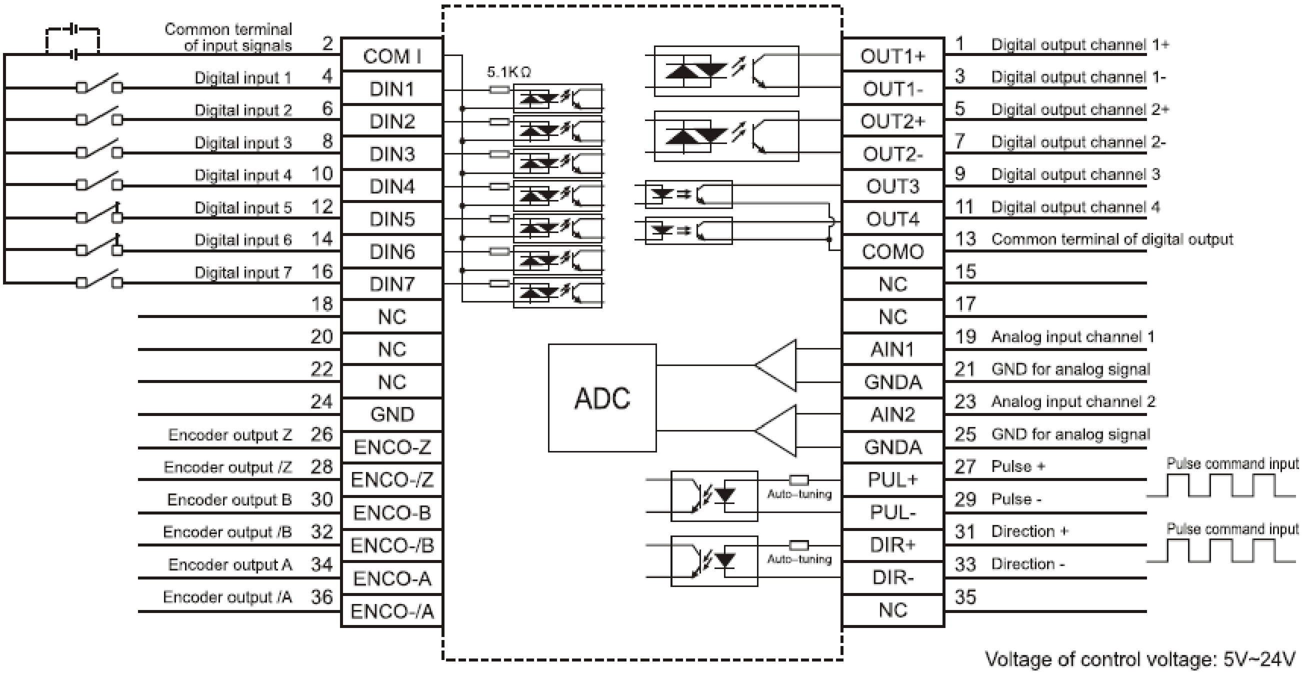


Fig. 8-1 Wiring diagram of FD2S driver in pulse control mode

2.Common anode connection (to controllers that support valid low level output)

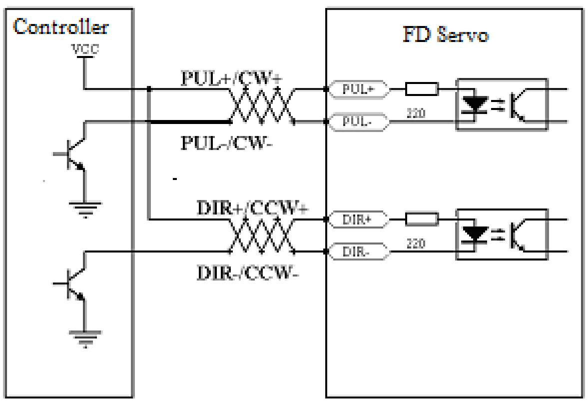


Fig. 8-2 Common anode connection (to controllers that support valid low level output)

3. Common cathode connection (to controllers that support valid high level output)

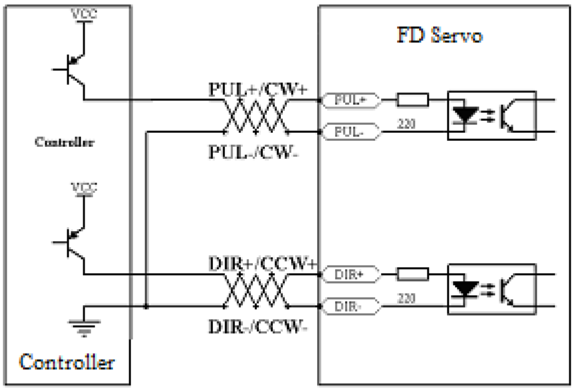


Fig. 8-3 Common cathode connection (to controllers that support valid high level output)

8.1.2 Parameters for Pulse Control Mode

1. Parameters for electronic gear ratio

Table 8-1 Parameters for electronic gear ratio

Numeric Display	Variable Name	Meaning	Default Value	Range
d3.34	Gear_Factor	Numerator of electronic gear 0 in mode -4	1000	-32767~32767
d3.35	Gear_Divider	Denominator of electronic gear 0 in mode -4	1000	1~32767

Parameters for electronic gear ratio are used to set the numerator and denominator of electronic gears when the driver operates in mode -4.

Command pulse input → F1 → $\frac{Gear\_Factor}{Gear\_Divider}$ → Command pulse output → F2

Namely: F2= $\frac{Gear\_Factor}{Gear\_Divider}$ * F1

If the electronic gear ratio is 1:1, 10000 pulses are inputted externally (the resolution of encoders is 2500 PPR, quadruple), and the motor turns a circle. If the electronic gear ratio is 2:1, 10000 pulses are inputted externally, and the motor turns two circles.

Multi electronic gears can be defined by DIN with function "Multi DinX" as shown in following table.

Multi Din 2	Multi Din 1	Multi Din 0	Descriptions	Parameter	
				Name	Address
0	0	0	Electronic gear 0	Gear_Factor 0	25080110
				Gear_Divider 0	25080210
0	0	1	Electronic gear 1	Gear_Factor 1	25090110
				Gear_Divider 1	25090210
0	1	0	Electronic gear 2	Gear_Factor 2	25090310
				Gear_Divider 2	25090410
0	1	1	Electronic gear 3	Gear_Factor 3	25090510
				Gear_Divider 3	25090610
1	0	0	Electronic gear 4	Gear_Factor 4	25090710
				Gear_Divider 4	25090810
1	0	1	Electronic gear 5	Gear_Factor 5	25090910
				Gear_Divider 5	25090A10

1	1	0	Electronic gear 6	Gear_Factor 6	25090B10
				Gear_Divider 6	25090C10
1	1	1	Electronic gear 7	Gear_Factor 7	25090D10
				Gear_Divider 7	25090E10

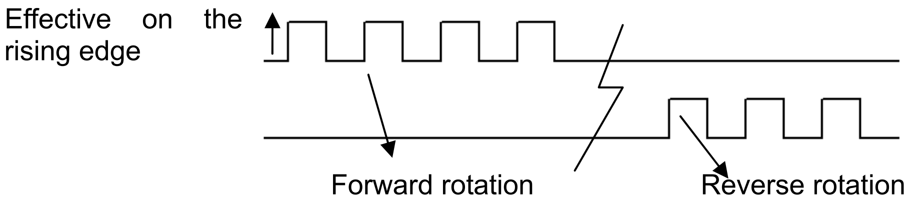
The default value of Gear_Factor and Gear_Divider are 1000.

2. Parameters for pulse mode selection

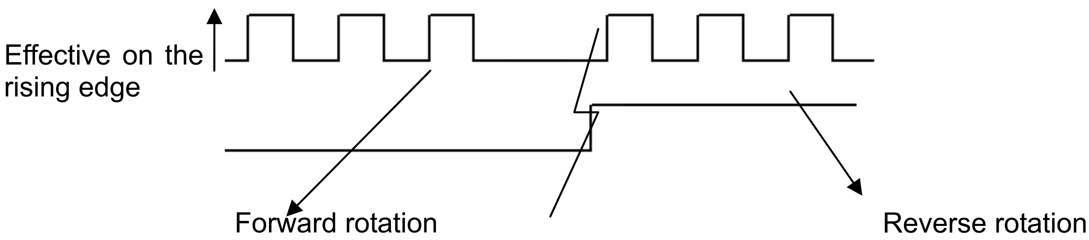
Table 8-2 Parameters for pulse mode selection

Numeric Display	Variable Name	Meaning	Default Value	Range
d3.36	PD_CW	0: Double pulse (CW/CCW) mode 1. Pulse direction (P/D) mode 2. Incremental encoder mode Note: To change this parameter, you need to save it with d3.00, and restarts it later.	1	N/A

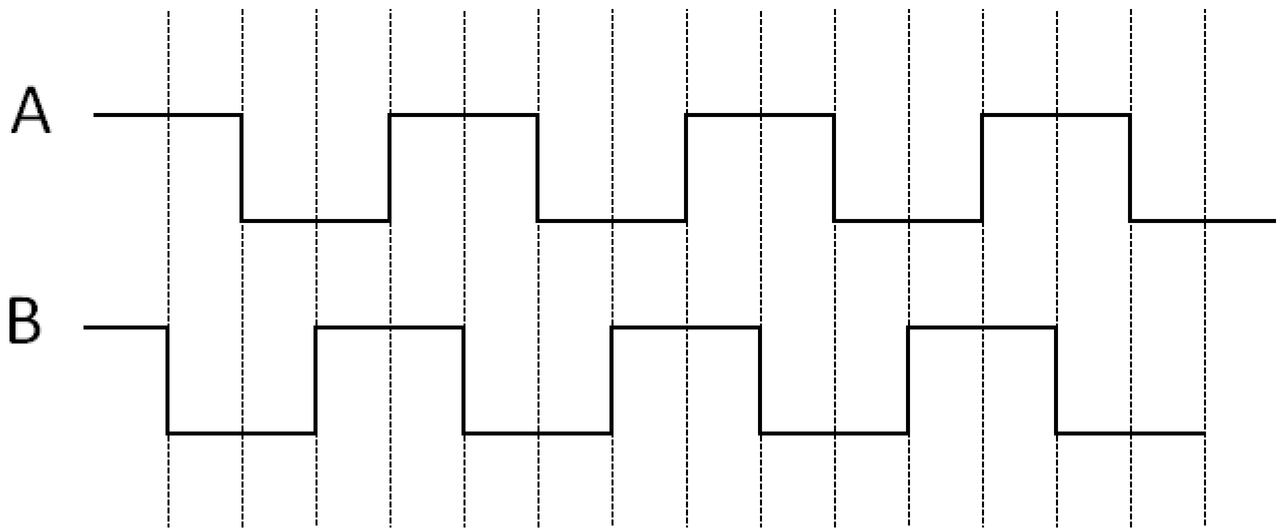
Double pulse (CW/CCW) mode (d3.36 = 0)



Pulse direction (P/D) mode (d3.36 = 1)



Incremental encoder mode (d3.36=2)



Parameters for pulse filtering coefficient

Table 8-3 Parameters for pulse filtering coefficient

Numeric Display	**Variable Name**	**Meaning**	**Default Value**	**Range**
d3.37	PD_Filter	Used to smooth the input pulses. Filter frequency: f = 1000/(2π* PD_Filter) Time constant: T = PD_Filter/1000 Unit: S Note: If you adjust this parameter during the operation, some pulses may be lost.	3	1~32767

When a driver operates in the pulse control mode, if the electronic gear ratio is set too high, it is required to adjust this parameter to reduce motor oscillation; however, if the parameter adjustment is too great, motor running instructions will become slower.

Parameters for pulse frequency control

Table 8-4 Parameters for pulse frequency control

Numeric Display	Variable Name	Meaning	Default Value	Range
d3.38	Frequency_Check	Indicates the limitation on pulse input frequency (kHz)	600	0~600

5. Parameters for gain control on position loops and velocity loops

Current loops are related to motor parameters (optimal parameters of the selected motor are default for the driver and no adjusting is required).

Parameters for velocity loops and position loops should be adjusted properly according to loading conditions. During adjustment of the control loop, ensure that the bandwidth of the velocity loop is at least twice of that of the position loop; otherwise oscillation may occur.

Table 7-5 Parameters for gain control on position loops

Numeric Display	Variable Name	Meaning	Default Value	Range
d2.07	Kpp	Indicates the proportional gain Kpp 0 of the position loop	1000	0~16384
d2.08	K_Velocity_FF	0 indicates no feedforward, and 256 indicates 100% feedforward	256	0~256
d2.09	K_Acc_FF	The value is inversely proportional to the feedforward	32767	32767~10
d0.05	Pc_Loop_BW	Sets the bandwidth of the position loop in Hz.	0	/
d2.26	Pos_Filter_N	Average filter parameter	1	/

Proportional gains of the position loop Kpp: If the proportional gain of the position loops increases, the bandwidth of the position loop is improved, thus reducing both the positioning time and following errors. However, too great bandwidth may cause noise or even oscillation. Therefore, this parameter must be set properly according to loading conditions. In the formula Kpp=103* Pc_Loop_BW,Pc_Loop_BW indicates the bandwidth of the position loop. The bandwidth of a position loop is less than or equal to that of a velocity loop. It is recommended that Pc_Loop_BW be less than Vc_Loop_BW /4 (Vc_Loop_BW indicates the bandwidth of a velocity loop).

Velocity feedforward of the position loop K_Velocity_FF : the velocity feedforward of a position loop can be increased to reduce position following errors. When position signals are not smooth, if the velocity feedforward of a position loop is reduced, motor oscillation during running can be reduced. Acceleration feedback of the position loop K_Acc_FF (adjustment is not recommended for this parameter): If great gains of position loops are required, the acceleration feedback K_Acc_FF can be properly adjusted to improve performance.

$$\text{K\_Acc\_FF} = \frac{I_p * K_t * Encoder\_R}{250000 * \sqrt{2} * J_t * \pi}$$

Note: K_Acc_FF is inversely proportional to the acceleration feedforward.

Table 8-6 Parameters for gain control on position loops

Numeric Display	Variable Name	Meaning	Default Value	Range
d2.01	Kvp	Sets the response speed of a velocity loop	100	0~32767
d2.02	Kvi	Adjusts speed control so that the time of minor errors is compensated	2	0~16384
d2.05	Speed_Fb_N	You can reduce the noise during motor operation by reducing the feedback bandwidth of velocity loops (smoothing feedback signals of encoders). When the set bandwidth becomes smaller, the motor responds slower. The formula is F=Speed_Fb_N*20+100. For example, to set the filter bandwidth to "F = 500 Hz", the parameter should be set to 20.	45	0~45

Proportional gain of velocity loop Kvp: If the proportional gain of the velocity loop increases, the responsive bandwidth of the velocity loop also increases. The bandwidth of the velocity loop is directly proportional to the speed of response. Motor noise also increases when the velocity loop gain increases. If the gain is too great, system oscillation may occur.
Integral gain of velocity loop Kvi: If the integral gain of the velocity loop increases, the low-frequency intensity is improved, and the time for steady state adjustment is reduced; however, if the integral gain is too great, system oscillation may occur.
Multiple gains can be defined by DIN with the function “Gain Switch 0” and “Gain Switch 1” as shown in following table.

Gain Switch 1	Gain Switch 0	Descriptions	Parameters	
			Name	Address
0	0	Gain 0	Kvp of Gain 0	60F90110
			Kvi of Gain 0	60F90210
			Kpp of Gain 0	60FB0110
0	1	Gain 1	Kvp of Gain 1	23400410
			Kvi of Gain 1	23400510
			Kpp of Gain 1	23400610
1	0	Gain 2	Kvp of Gain 2	23400710
			Kvi of Gain 2	23400810
			Kpp of Gain 2	23400910
1	1	Gain 3	Kvp of Gain 3	23400A10
			Kvi of Gain 3	23400B10
			Kpp of Gain 3	23400C10

If DIN is defined as “Gain Switch” function,then the parameter “PI_Switch” will disable.
Parameter “PI_Point”(60F92808) is used to display the current gain.
Auto-tuning can only be used to set Gain 0.
Vc_Loop_BW and Pc_Loop_BW are only corresponding to Gain 0.Other Gain needs to set by manual.
“PI_Switch” is used to switch Gain 0 and Gain 1.In mode -4,1 and 3,it will use Gain 1 when “Position reached” signal is valid,and use Gain 0 when “Position reached” signal is invalid.

8.1.3 Examples of Pulse Control Mode

In the pulse control mode, follow the steps below to configure a driver:
Step 1: Confirm whether the functions of the driver require enabling through external digital input ports. To enable the driver through external digital input ports, see Table 6-12 in Example 6-3 for settings. If it is not necessary to enable the driver through external digital input ports, you can disable the enabling control function of external digital input ports by referring to Table 6-13 of Example 6-3, and enable the driver by setting its internal parameters.

Step 2: Confirm whether limit switches are required. By default, the driver operates in the limit status after being powered on. In this case, the numeric display has limit status display. If there is no limit switches, please disable the function of limit switches by referring to Example 6-4.
Step 3: Confirm mode switching bits and operation modes by referring to the settings in Example 6-5. The factory default settings of the driver are as follows: When no signal is inputted on DIN3, the driver operates in the “-4” mode (pulse control mode).
Step 4: After function configuration on digital input ports, it is required to set parameters such as pulse modes and electronic gear ratio.
Step 5: Save parameters.

Example 8-1: Pulse control mode “-4” – enable the driver through external digital input

Requirement: DIN1 is used for enabling the driver, DIN2 is used for error resetting, and DIN3 controls the operation modes of the driver (the mode is “-4” when no signal is inputted, and the mode is “-3” when signal is inputted). Limit switches are unavailable. The pulse form is pulse/direction, and the electronic rear ratio is 2:1. Table 8-7 describes the setup method.

Table 8-7: Pulse control mode “-4” – enable the driver through external digital input

Numeric Display	Variable Name	Meaning	Parameter Settings
d3.01	Din1_Function	Defines the functions of digital input port 1	000.1 (Driver enable)
d3.02	Din2_Function	Defines the functions of digital input port 2	000.2 (Fault reset)
d3.03	Din3_Function	Defines the functions of digital input port 3	000.4 (Operation mode control)
d3.05	Din5_Function	Defines the functions of digital input port 5	The default value 001.0 changes to 000.0 (position positive limits are disabled)
d3.06	Din6_Function	Defines the functions of digital input port 6	The default value 002.0 changes to 000.0 (position negative limits are disabled)
d3.16	Din_Mode0	Select this operation mode when input signals are invalid	Set to 0.004 (-4) mode (pulse control mode)
d3.17	Din_Mode1	Select this operation mode when input signals are valid	Set to 0.003 (-3) mode (instantaneous speed mode)
d3.34	Gear_Factor	Indicates the numerator to set electronic gears in the “-4” operation mode (pulse control mode)	Set to 2000
d3.35	Gear_Divider	Indicates the denominator to set electronic gears in the “-4” operation mode (pulse control mode)	Set to 1000
d3.36	PD_CW	0: Double pulse (CW/CCW) mode 1. Pulse direction (P/D) mode Note: To change this parameter, you need to save it with the address “d3.00”, and restarts it later.	Default value is 1 (pulse direction)
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

Example 8-2 Pulse control mode "-4" – enable the driver automatically after driver power on

Requirement: The auto power-on function of the driver is enabled, DIN2 is used for error resetting, and DIN3 controls the operation modes of a driver (the mode is "-4" when no signal is inputted, and the mode is "3" when signal is inputted). Limit switches are unavailable. The pulse form is pulse/direction, and the electronic rear ratio is 1:2. Table 8-8 describes the setup method.

Table 8-8 Pulse control mode "-4" – enable driver automatically after driver power on

Numeric Display	**Variable Name**	**Meaning**	**Parameter Settings**
d3.01-d3.07	DinX_ Function (1~7)	Defines the functions of digital input ports 1-7	None of the digital input port can be set to 000.1, that is, the Enable function is not controlled by any digital input port.
d3.02	Din2_Function	Defines the functions of digital input port 2	000.2 (Error resetting)
d3.03	Din3_Function	Defines the functions of digital input port 3	000.4 (Control on operation modes for the driver)
d3.05	Din5_Function	Defines the functions of digital input port 5	The default value 001.0 changes to 000.0 (position positive limits are disabled)
d3.06	Din6_Function	Defines the functions of digital input port 6	The default value 002.0 changes to 000.0 (position negative limits are disabled)
d3.10	Switch_On_Auto	0: No control 1:Automatically locks the motor when the driver is powered on	Set to 1
d3.16	Din_Mode0	Select this operation mode when input signals are invalid	Set to 0.004 (-4) mode (pulse control mode)
d3.17	Din_Mode1	Select this operation mode when input signals are valid	Set to 0.003 (-3) mode (instantaneous speed mode)
d3.34	Gear_Factor	Indicates the numerator to set electronic gears in the "-4" operation mode (pulse control mode)	Set to 1000
d3.35	Gear_Divider	Indicates the denominator to set electronic gears in the "-4" operation mode (pulse control mode)	Set to 2000
d3.36	PD_CW	0: Double pulse (CW/CCW) mode 1. Pulse direction (P/D) mode Note: To change this parameter, you need to save it with the address "d3.00", and restarts it later.	Default value is 1 (pulse direction)
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

8.2 Speed Mode ("-3" or "3" Mode)

In the instantaneous speed mode ("-3" mode), the actual speed reaches the target speed instantly. As a contrast, in the speed mode with acceleration/deceleration ("3" mode), the actual speed gradually increases until it reaches the target speed. Both the acceleration and deceleration (trapeziform shape) are configured respectively by d2.10 and d2.11. In the "3" mode, you can set Kpp to enable/disable position loops. If a position loop is enabled, speed oscillation is less than that when the loop is disabled. If Kpp is 0, it indicates that the position loop is closed.

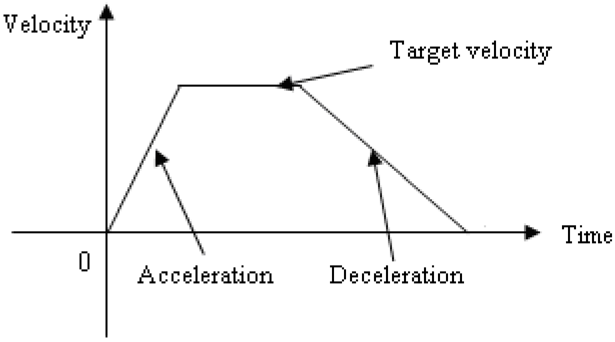


Fig. 8-4 The speed mode "3" with acceleration/deceleration

8.2.1 Wiring in Analog – Speed Mode

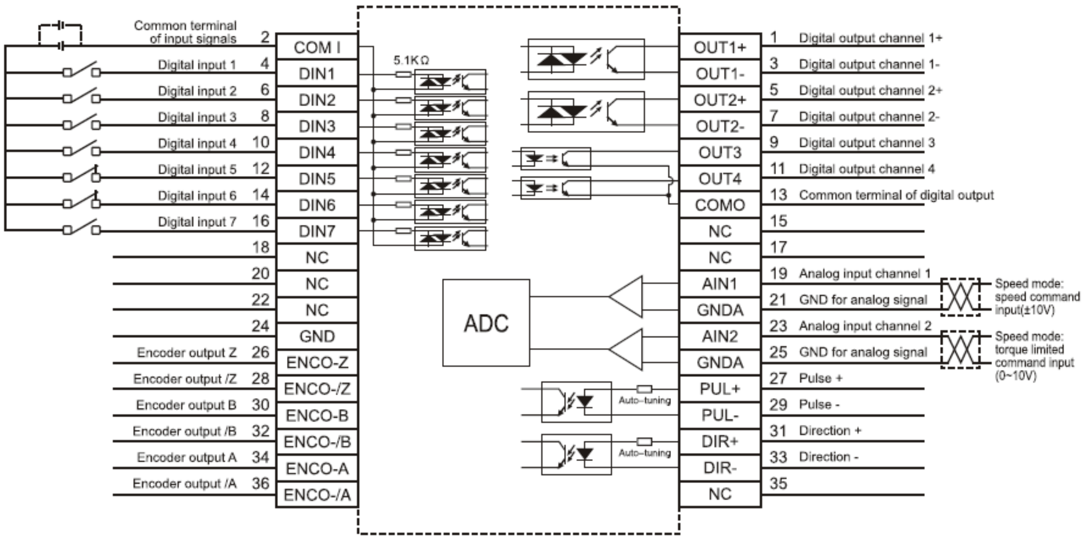


Fig. 8-5 Wiring diagram of FD2S Servo in analog–speed mode

8.2.2 Parameters for Analog – Speed Mode

Table 8-9 Parameters for analog – speed mode

Numeric Display	Variable Name	Meaning	Default Value	Range
d3.22	Analog1_Filter	Used to smooth the input analog signals. Filter frequency: f=4000/(2π* Analog1_Filter) Time Constant (T) = Analog1_Filter/4000 (S)	5	1~127
d3.23	Analog1_Dead	Sets dead zone data for external analog signal 1	0	0~8192
d3.24	Analog1_Offset	Sets offset data for external analog signal 1	0	-8192~8192
d3.25	Analog2_Filter	Used to smooth the input analog signals. Filter frequency: f=4000/(2π* Analog1_Filter) Time Constant (T) = Analog2_Filter/4000 (S)	5	1~127
d3.26	Analog2_Dead	Sets dead zone data for external analog signal 2	0	0~8192

d3.27	Analog2_Offset	Sets offset data for external analog signal 2	0	-8192~8192
d3.28	Analog_Speed_Con	Chooses analog-speed channels 0: Invalid analog channel 1: Valid analog channel 1 (AIN1) 2: Valid analog channel 2 (AIN2) 10～17：AIN1 for "Din_Speed (X-10)" 20～27：AIN2 for "Din_Speed (X-20)" Valid in mode -3, 3 and 1.	0	N/A
d3.29	Analog_Speed_Factor	Sets the proportion between analog signals and output speed	1000	N/A
d3.32	Analog_MaxT_Con	0: No control 1: Max torque that Ain1 can control 2: Max torque that Ain2 can control	0	N/A
d3.33	Analog_MaxT_Factor	Indicates the max torque factor for analog signal control	8192	N/A

When d3.28 is 1 or 2,mode 1 is invalid,mode 3 and -3 are valid.
When d3.28 is 10~17 or 20~27,mode 1,3 and -3 are valid.
When d3.28 is 10~17(AIN1 for "Din_Speed (X-10)"),the corresponding speed is as following table.

10	11	12	13	14	15	16	17
Din_Speed 0	Din_Speed 1	Din_Speed 2	Din_Speed 3	Din_Speed 4	Din_Speed 5	Din_Speed 6	Din_Speed 7

When d3.28 is 20~27(AIN1 for "Din_Speed (X-10)"),the corresponding speed is as following table.

20	21	22	23	24	25	26	27
Din_Speed 0	Din_Speed 1	Din_Speed 2	Din_Speed 3	Din_Speed 4	Din_Speed 5	Din_Speed 6	Din_Speed 7

8.2.3 Analog Signal Processing

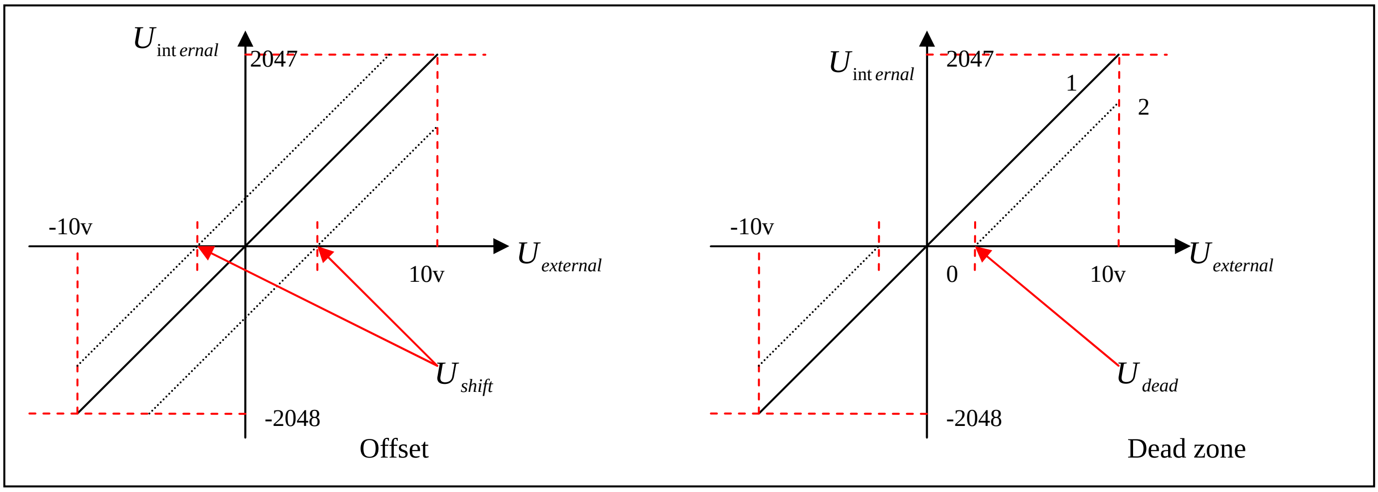


Fig. 8-6 Analog signal processing

Electrical control on internal variables is available only after ADC conversion and offset of external analog signals, and judgment of dead zone signals.

For offset processing, see the left part in Fig. 8-6; for dead zone processing, see the right part in Fig. 8-6.

Mathematical equation for offset processing: $U_{internal} = U_{external} - U_{shift}$

Mathematical equation for dead zone processing:

$$\begin{cases} U_{internal} = 0 \cdots\cdots\cdots\cdots\cdots\cdots -U_{dead} \le U_{external} \le U_{dead} \\ U_{internal} = U_{external} - U_{dead} \cdots\cdots\cdots \begin{cases} -U_{dead} > U_{external} \\ U_{dead} < U_{external} \end{cases} \end{cases}$$

Mathematical equation for integrated processing (offset and dead zone)

$$\begin{cases} U_{internal} = 0 \cdots\cdots\cdots\cdots\cdots\cdots\cdots\cdots -U_{dead} \le U_{external} - U_{shift} \le U_{dead} \\ U_{internal} = U_{external} - U_{shift} - U_{dead} \cdots\cdots\cdots \begin{cases} -U_{dead} > U_{external} - U_{shift} \\ U_{dead} < U_{external} - U_{shift} \end{cases} \end{cases}$$

Table 8-10 Analog signal variables

Variable	Meaning	Range
$U_{internal}$	Internal data corresponding to the external voltage	-10 V – 10 V corresponds to -2048 – 2047 when no offset or dead zone voltage exists
$U_{external}$	External input voltage	-10V – 10V
U_{shift}	Offset voltage	0 – 10 V corresponds to $Analog\_Offset$ 0~8191
U_{dead}	Dead zone voltage	0 – 10 V corresponds to $Analog\_Dead$ 0~8191

The obtained analog signal $U_{internal}$ obtains U_{filter} after passing through a first-order low-pass filter, and is applied by the internal programs again.

In the analog – speed mode, if the analog signal U_{filter} that passes through the filter is multiplied by a factor, this signal will be regarded as the internal target speed V_{demand}.

Mathematical formula: $V_{demand} = Factor * U_{filter} \cdots\cdots -2048 \le U_{filter} \le 2047$

V_{demand} Formula for V_{rpm} conversion: $V_{rpm} = \frac{1875 * V_{demand}}{512 * Encoder\_R}$

Note: The resolution unit of an encoder is inc/r.

8.2.4 Calculation Procedure for Analog – speed Mode

Table 8-11 Calculation procedure for analog – speed mode

Procedure	**Method**	**Formula**
Step 1	Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings	$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$
Step 2	Calculate V_{demand} according to the required speed V_{rpm}	$V_{rpm} = \frac{1875 * V_{demand}}{512 * Encoder\_R}$
Step 3	Calculate $Factor$ according to U_{filter} and V_{demand}	$V_{demand} = Factor * U_{filter}$
Step 5	Calculate $Analog\_Dead$ according to the required dead zone voltage	$8191/10v = Analog\_Dead / U_{dead}$
Step 5	Calculate $Analog\_Offset$ according to the required offset voltage	$8191/10v = Analog\_Offset / U_{shift}$

8.2.5 Examples of Analog – Speed Mode

In the analog – speed mode, follow the steps below to set a driver:
Step 1: Confirm whether it is necessary to enable the driver through external digital input ports. To enable the driver through external digital input ports, see Table 6-12 in Example 6-3 for settings. If the driver does not require enabling through external digital input ports, you can disable the enabling function of external digital input ports by referring to Table 6-13 of Example 6-3, and enable the auto power-on function of the driver by setting its internal parameters.
Step 2: Confirm whether limit switches are required. By default, the driver operates in the limit status after being powered on. In this case, the numeric display has limit status display. If limit switches are unavailable, please disable the function of limit switches by referring to Example 6-4.
Step 3: Confirm the mode switching positions and operation modes by referring to the settings in Example 6-5. The factory default settings are as follows: When no signal is inputted to DIN3, the driver operates in the "-4" mode (d3.16 = -4); when signal is inputted to DIN3, the driver operates in the "-3" mode (d3.17 = -3). If the driver is required to operate in the speed mode after being powered on, set d3.16 to -3 or 3.
Step 4: After configuring functions on digital input ports, select the analog – speed channel, and set parameters such as analog – speed factors, dead zone, offset and filtering.
Step 5: Save parameters.

Example 8-3: Analog – speed mode (without setting the dead zone voltage and offset voltage)

Requirement: DIN1 is used for enabling the driver, DIN2 is used for error resetting, and DIN3 controls the operation modes of the driver (the mode is "-3" when no signal is inputted, and is "3" when signal is inputted). Limit switches are unavailable. The voltage 10V corresponds to the rated rotation speed of 3000 rpm, and -10V corresponds to the rated rotation speed of -3000 rpm. Select analog channel 1 (AIN1) to control the speed.

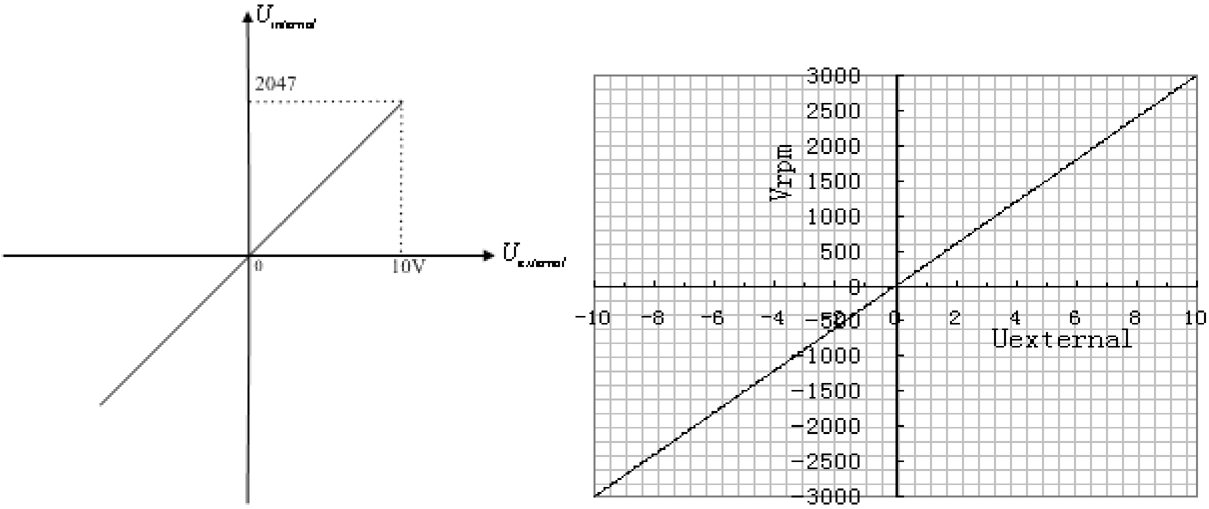


Fig. 8-7 Schematic diagram of Example 8-3

Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings:

$$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$$ (In this example, $U_{dead} = 0$, and $U_{shift} = 0$)

Result: U_{filter} =2047

Calculate V_{demand} according to the required speed V_{rpm}:

$$V_{rpm} = \frac{1875 * V_{demand}}{512 * \text{Encoder\_R}} = 3000RPM$$ (Encoder_R is 10000 inc/r)

Result: $V_{demand} = 8192000$

Calculate $Factor$ according to U_{filter} and V_{demand}:

$V_{demand} = Factor * U_{filter}$

Result: $Factor = 4000$

Table 8-12 Parameter settings in Example 8-3

Numeric Display	Variable Name	Meaning	Parameter Settings
d3.01	Din1_Function	Define the functions of digital input port 1	000.1 (Driver enable)
d3.02	Din2_Function	Define the functions of digital input port 2	000.2 (Error resetting)
d3.03	Din3_Function	Define the functions of digital input port 3	000.4 (Control over operation modes of drivers)
d3.05	Din5_Function	Define the functions of digital input port 5	The default value 001.0 changes to 000.0 (position positive limits are disabled)
d3.06	Din6_Function	Define the functions of digital input port 6	The default value 002.0 changes to 000.0 (position negative limits are disabled)
d3.16	Din _Mode0	Select this operation mode when input signals are invalid	Set to 0.003 (-3) mode (instantaneous speed mode)
d3.17	Din _Mode1	Select this operation mode when input signals are valid	Set to 0.003 (3) mode (speed mode with acceleration/deceleration)
d3.22	Analog1_Filter	Used to smooth the input analog signals. Filter frequency: f=4000/(2π* Analog1_Filter) Time Constant (T) = Analog1_Filter/4000 (S)	
d3.23	Analog1_Dead	Set dead zone data for external analog signal 1	Set to 0
d3.24	Analog1_Offset	Set offset data for external analog signal 1	Set to 0
d3.28	Analog_Speed_Con	Chooses analog-speed channels 0: Invalid analog channel 1: Valid analog channel 1 (AIN1) 2: Valid analog channel 2 (AIN2) 10～17： AIN1 for "Din_Speed (X-10)" 20～27： AIN2 for "Din_Speed (X-20)" Valid in mode -3, 3 and 1.	Set to 1
d3.29	Analog_Speed_Factor	Set the proportion between analog signals and output speed	Set to 4000
d2.10	Profile_Acce_16	Set the acceleration in operation mode 3 and 1.(rps/s)	610 by defaut
d2.11	Profile_Dece_16	Set the deceleration in operation mode 3 and 1.(rps/s)	610 by defaut
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

Example 8-4 Analog – speed mode (setting the dead zone voltage)

Requirement: The dead zone voltage ranges from - 0.5 V to 0.5 V, that is, the speed is 0 when the voltage ranges from - 0.5 V to 0.5 V. The voltage 10 V corresponds to 3000 rpm, and -10 V corresponds to -3000 rpm. Select analog channel 1 (AIN1) to control the speed.

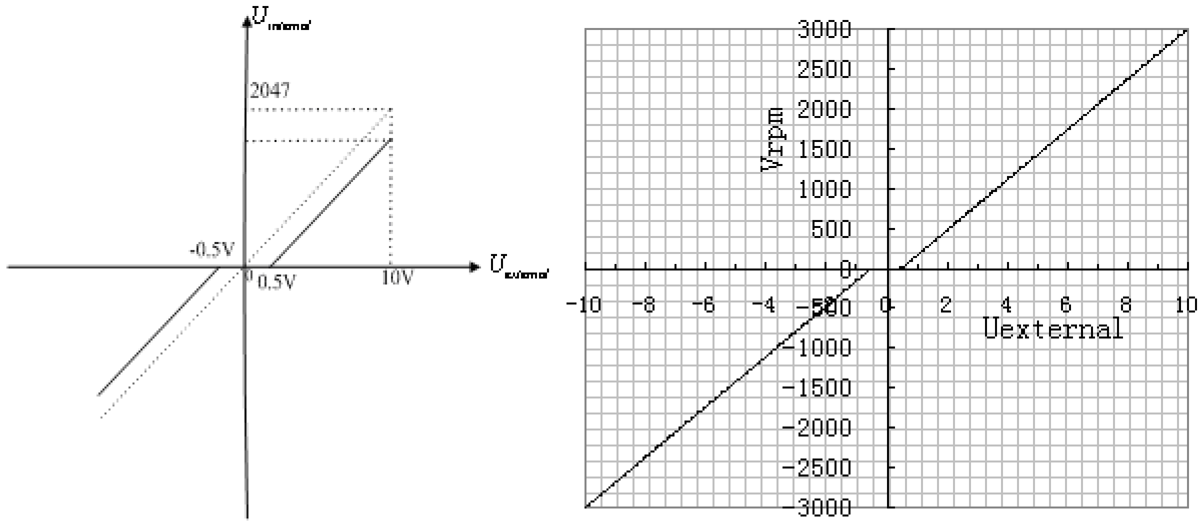


Fig. 8-8 Schematic diagram of Example 8-4

Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings:

$$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$$ (In this example, $U_{dead} = 0.5$, and $U_{shift} = 0$)

Result: U_{filter} =1944

Calculate V_{demand} according to the required speed : V_{rpm}

$$V_{rpm} = \frac{1875 * V_{demand}}{512 * \text{Encoder\_R}} = 3000 RPM$$, (Encoder_R:10000 inc/r)

Result: $V_{demand} = 8192000$

Calculate U_{filter} according to V_{demand} and $Factor$:

$V_{demand} = Factor * U_{filter}$

Result: $Factor$ =4213

Calculate $Analog1\_Dead$ according to the required dead zone voltage:

$8191/10v = Analog1\_Dead / U_{dead}$

Result: $Analog1\_Dead$ =410

The following changes are required on the basis of Example 8-3.

Table 8-13 Parameter settings in Example 8-4

d3.23	Analog1_Dead	Sets dead zone data for external analog signal 1	Set to 410
d3.29	Analog_Speed_Factor	Sets the proportion between analog signals and output speed	Set to 4213
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

Example 8-5 Analog – speed mode (setting the offset voltage)

Requirement: The offset voltage is 1 V, that is, the speed is positive when the voltage is greater than 1 V, and is negative when the voltage is less than 1 V. In this case, the voltage 10 V corresponds to 3000 rpm, and -9 V corresponds to -3000 rpm (in case of -10 V, the corresponding speed is less than -3000 rpm). Select analog channel 1 (AIN1) to control the speed.

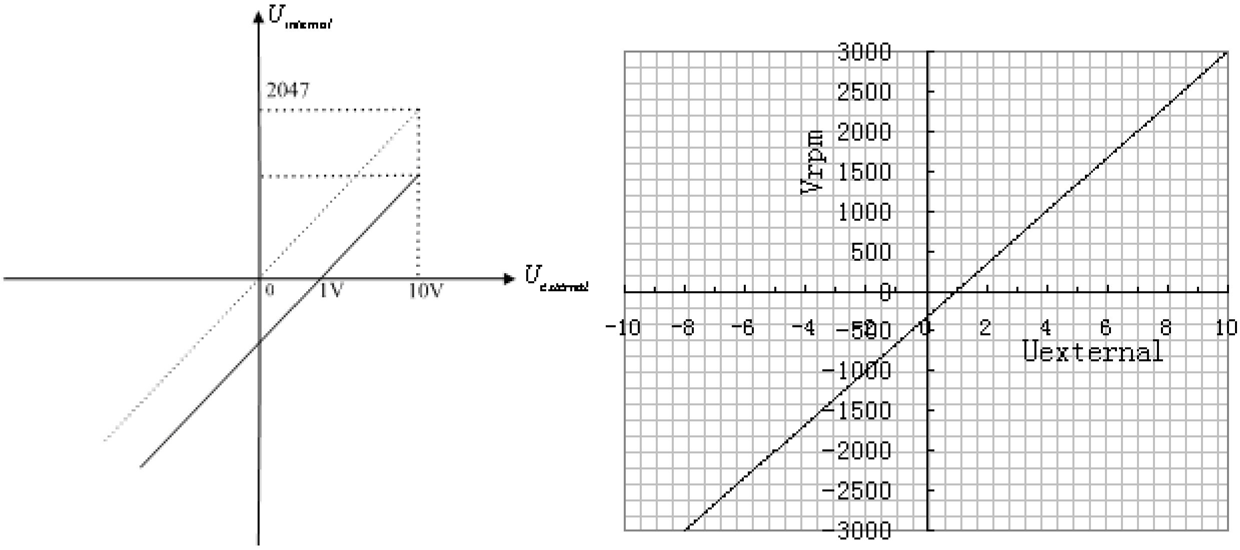


Fig. 8-9 Schematic diagram of Example 8-5

Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings:

$$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$$ (In this example, $U_{dead} = 0$, and $U_{shift} = 1$)

Result: $U_{filter} = 1842$

Calculate V_{demand} according to the required speed : V_{rpm}

$$V_{rpm} = \frac{1875 * V_{demand}}{512 * \text{Encoder\_R}} = 3000RPM$$, (Encoder_R:10000 inc/r)

Result: $V_{demand} = 8192000$

Calculate U_{filter} according to V_{demand} and $Factor$:

$$V_{demand} = Factor * U_{filter}$$

Result: $Factor$ =4447

Calculate $Ana\log 1\_Offset$ according to the required offset voltage:

$$8191/10v = Ana\log 1\_Offset / U_{shift}$$

Result: $Ana\log 1\_Offset$ =819

The following changes are required on the basis of Example 8-3.

Table 8-14 Parameter settings in Example 8-5

d3.24	Analog1_Offset	Sets offset data for external analog signal 1	Set to 819
d3.29	Analog_Speed_Factor	Sets the proportion between analog signals and output speed	Set to 4447
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

Example 8-6: Analog – speed mode (setting the dead zone voltage and offset voltage)

Requirement: Set the offset voltage to 1V, the dead zone voltage to 0.5V to 1.5V, and the max speed corresponding to 10V to 3000 rpm. Select analog channel 1 (AIN1) to control the speed.

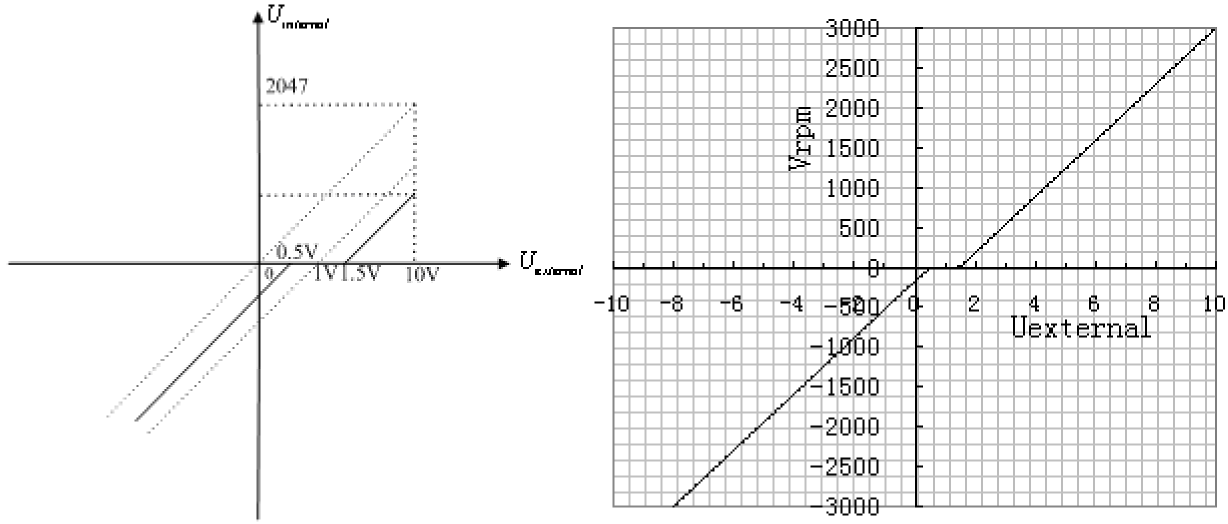


Fig. 8-10 Schematic diagram of Example 8-6

Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings:

$$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$$ (In this example, $U_{dead} = 0.5$, and $U_{shift} = 1$)

Result: U_{filter} =1740

Calculate V_{demand} according to the required speed : V_{rpm}

$$V_{rpm} = \frac{1875 * V_{demand}}{512 * \text{Encoder\_R}} = 3000RPM$$, (Encoder_R:10000 inc/r)

Result: $V_{demand} = 8192000$

Calculate $Factor$ according to U_{filter} and V_{demand} :

$$V_{demand} = Factor * U_{filter}$$

Result: $Factor$ =4708

Calculate $Ana\log1\_Dead$ according to the required dead zone voltage:

$$8191/10v = Ana\log1\_Dead / U_{dead}$$

Result: $Ana\log1\_Dead$ =409

Calculate $Ana\log1\_Offset$ according to the required offset voltage:

$$8191/10v = Ana\log1\_Offset / U_{shift}$$

Result: $Ana\log1\_Offset$ =819

The following changes are required on the basis of Example 8-3.

Table 8-15 Parameter settings in Example 8-6

d3.23	Analog1_Dead	Sets dead zone data for external analog signal 1	Set to 409
d3.24	Analog1_Offset	Sets offset data for external analog signal 1	Set to 819
d3.29	Analog_Speed_Factor	Sets the proportion between analog signals and output speed	Set to 4708
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

8.3 Torque Mode ("4" Mode)

8.3.1 Wiring in Analog – Torque Mode

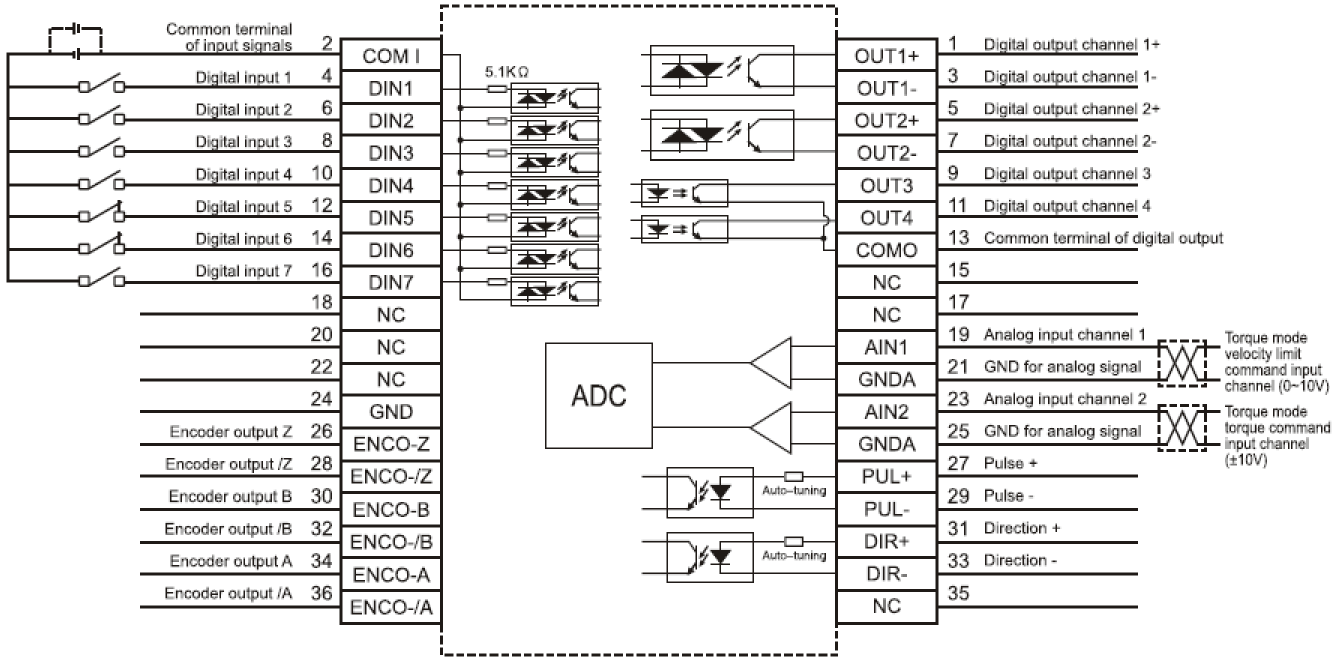


Fig. 8-11 Wiring diagram of FD2S Servo in analog – torque mode

8.3.2 Parameters for Analog – Torque Mode

Table 8-16 Parameters for analog – torque mode

Numeric Display	Variable Name	Meaning	Default Value	Range
d3.22	Analog1_Filter	Used to smooth the input analog signals. Filter frequency: f=4000/(2π* Analog1_Filter) Time Constant: τ = Analog1_Filter/4000 (S)	5	1~127
d3.23	Analog1_Dead	Sets dead zone data for external analog signal 1	0	0~8192
d3.24	Analog1_Offset	Sets offset data for external analog signal 1	0	-8192~8192
d3.25	Analog2_Filter	Used to smooth the input analog signals. Filter frequency: f=4000/(2π* Analog1_Filter) Time Constant (T) = Analog2_Filter/4000 (S)	5	1~127
d3.26	Analog2_Dead	Sets dead zone data for external analog signal 2	0	0~8192
d3.27	Analog2_Offset	Sets offset data for external analog signal 2	0	-8192~8192

d3.30	Analog_Torque_Con	Selects analog - torque channels 0: Invalid analog channel 1: Valid analog channel 1 (AIN1) 2: Valid analog channel 2 (AIN2) Valid mode 4	0	N/A
d3.31	Analog_Torque_Factor	Sets the proportion between analog signals and output torque (current)	1000	N/A
d2.15	Speed_Limit_Factor	The factor that limits the maximum speed in the torque mode $\begin{cases} F_{Actual\_torque} = F_{Demand\_torque} \cdots\cdots \\ F_{Actual\_torque} = F_{Demand\_torque} - N*(V_{Actual\_speed} - V_{Max\_speed}) \cdots\cdots \end{cases}$ V_{max_speed} complies with d2.24 Max_Speed_RPM parameter settings.	10	0~1000
d2.24	Max_Speed_RPM	Limits the max rotation speed of the motor	5000	0~6000

8.3.3 Analog Signal Processing

In the analog – torque mode, external analog command signals are directly inputted to the current loops in the driver, thus directly controlling target current through the internal current loop. Analog signal is processed in the same way as that in the analog – speed mode.

In the analog – torque mode, I_{demand} is calculated according to the specified T_{demand} with the formula of $T_{demand} = K_t * \frac{I_{demand}}{\sqrt{2}}$ (K_t is a torque constant).

$Factor$ is calculated according to I_{demand} and U_{filter} with the formula of $I_{demand} = \frac{Factor * U_{filter}}{2048 * 2048} * Ipeak$ ($Ipeak$ indicates the peak current of a driver).

Table 8-17 K_t and $Ipeak$ parameters

Motor Model	K_t **(Nm/A)**	**Driver Model**	$Ipeak$ **(A)**
SMH60S-0020-30AXK-3LKX	0.48	FD422S	15
SMH60S-0040-30AXK-3LKX	0.48		
SMH80S-0075-30AXK-3LKX	0.662		
SMH80S-0100-30AXK-3LKX	0.562	FD432S	27.5
SMH110D-0105-20AXK-4LKX	0.992		
SMH110D-0126-20AXK-4LKX	1.058		
SMH130D-0105-20AXK-4HKX	1.1578		
SMH130D-0157-20AXK-4HKX	1.191		
SMH110D-0126-30AXK-4HKX	1.058	FD622S	25
SMH110D-0157-30AXK-4HKX	0.992		
SMH110D-0188-30AXK-4HKX	1.058		
SMH130D-0105-20AXK-4HKX	1.1578		
SMH130D-0157-20AXK-4HKX	1.191		
SMH130D-0210-20AXK-4HKX	1.3232		
SMH150D-0230-20AXK-4HKX	1.65		

8.3.4 Calculation Procedure for Analog – Torque Mode

Table 8-17 Calculation procedure for analog – torque mode

Procedure	Method	Formula
Step 1	Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings	$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$
Step 2	Calculate I_{demand} according to the required torque T_{demand}	$T_{demand} = K_t * \frac{I_{demand}}{\sqrt{2}}$
Step 3	Calculate $Factor$ according to U_{filter} and I_{demand}	$I_{demand} = \frac{Factor * U_{filter}}{2048 * 2048} * Ipeak$
Step 4	Calculate $Analog\_Dead$ according to the required dead zone voltage	$8191/10v = Analog\_Dead / U_{dead}$
Step 5	Calculate $Analog\_Offset$ according to the required offset voltage	$8191/10v = Analog\_Offset / U_{shift}$

8.3.5 Examples of Analog – Torque Mode

In the analog – torque mode, follow the steps below to configure a driver:
Step 1: Confirm whether it is necessary to enable the driver through external digital input ports. To enable the driver through external digital input ports, see Table 6-12 in Example 6-3 for settings. If the driver does not require enabling through external digital input ports, you can disable the enabling function of external digital input ports by referring to Table 6-13 of Example 7-3, and enable the auto power-on function of the driver by setting its internal parameters.
Step 3: Confirm mode switching positions and operation modes by referring to the settings in Example 6-5. The factory default settings for the driver are as follows: When no signal is inputted to DIN3, the driver operates in the “-4” mode (d3.16 = -4); when signal is inputted to DIN3, the driver operates in the “-3” mode (d3.17 = -3). If the driver is required to operate in the torque mode (“4” mode), please set d3.16 or d3.17 to 4. In case d3.16 = 4, if DIN3 has no input signals when the driver is powered on, the driver operates in the “4” mode. In case d3.17 = 4, if DIN3 has input signals, the driver operates in the “4” mode.
Step 3: After configuring functions on digital input ports, select the analog – torque channel, and set parameters such as analog – torque factors, dead zone, offset, filtering, speed limit factors, and max speed limits.
Step 4: Save parameters.

Example 8-7: Analog – torque mode (without setting the dead zone voltage and offset voltage)

Requirement: DIN1 is used for enabling the driver, DIN2 is used for error resetting, and DIN3 controls the operation modes of the driver (the mode is “4” when no signal is inputted, and is “3” when signal is inputted). The motor Kt is 0.48 Nm/A, and the peak current of drivers is 15 A. The analog input voltage -10 V corresponds to -0.64 Nm, and 10 V corresponds to 0.64 Nm. Select analog channel 2 (AIN1) to control the torque.

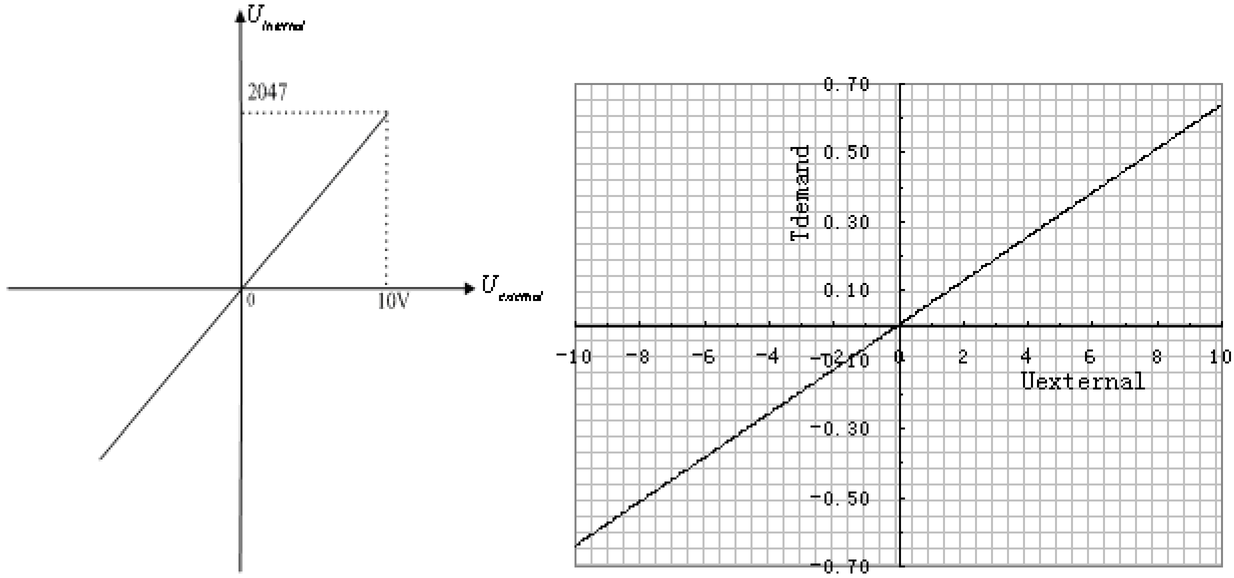


Fig. 8-13 Schematic diagram of Example 8-7

Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings:

$$\frac{2047}{10v} = \frac{U_{filter}}{10v - U_{shift} - U_{dead}}$$ (In this example, $U_{dead} = 0$, and $U_{shift} = 0$)

Result: U_{filter} =2047

Calculate I_{demand} according to the required torque T_{demand}:

$$I_{demand} = \frac{T_{demand}}{K_t} * \sqrt{2}$$

Result: I_{demand} =1.89

Calculate $Factor$ according to U_{filter} and I_{demand}:

$$Factor = \frac{I_{demand}}{U_{filter} * Ipeak} * 2048 * 4096$$

Result: $Factor = \frac{1.89}{2047 * 15} * 2048 * 4096 = 515$

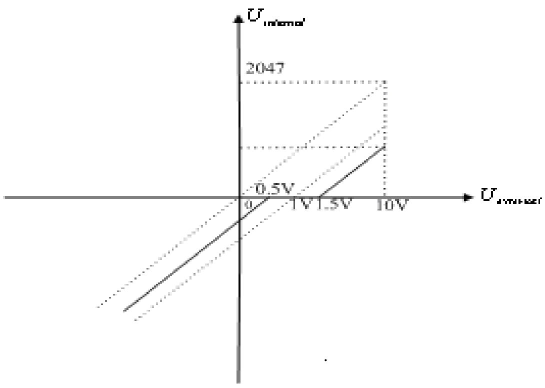
Table 8-18 Parameter settings in Example 8-7

Numeric Display	**Variable Name**	**Meaning**	**Parameter Settings**
d3.01	Din1_Function	Defines the functions of digital input port 1	000.1 (Driver enable)
d3.02	Din2_Function	Defines the functions of digital input port 2	000.2 (Error resetting)
d3.03	Din3_Function	Defines the functions of digital input port 3	000.4 (Control over operation modes of drivers)
d3.16	Din _Mode0	Select this operation mode when input signals are invalid	Set to 0004 (4) mode (torque mode)
d3.17	Din _Mode 1	Select this operation mode when input signals are valid	Set to 0.003 (3) mode (speed mode with acceleration/deceleration)
d3.25	Analog2_Filter	Used to smooth the input analog signals. Filter frequency: f=4000/(2π*	

		Analog1_Filter) Time Constant: T = Analog2_Filter/4000 (S)	
d3.26	Analog2_Dead	Sets dead zone data for external analog signal 2	Set to 0
d3.27	Analog2_Offset	Sets offset data for external analog signal 2	Set to 0
d3.31	Analog_Torque_Factor	Sets the proportion between analog signals and output torque (current)	Set to 515
d3.30	Analog_Torque_Con	Selects analog - torque channels 0: Invalid analog channel 1: Valid analog channel 1 (AIN1) 2: Valid analog channel 2 (AIN2) Valid mode 4	Set to 2
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

Example 8-8: Analog – torque mode (setting the dead zone voltage and offset voltage)

Requirement: The offset voltage is 1V, and the dead zone voltage is 0.5V. The motor Kt is 0.48 Nm/A, and the peak current of the driver is 15A. The analog input voltage 10V corresponds to 0.64Nm. Select analog channel 2 (AIN2) to control the torque.



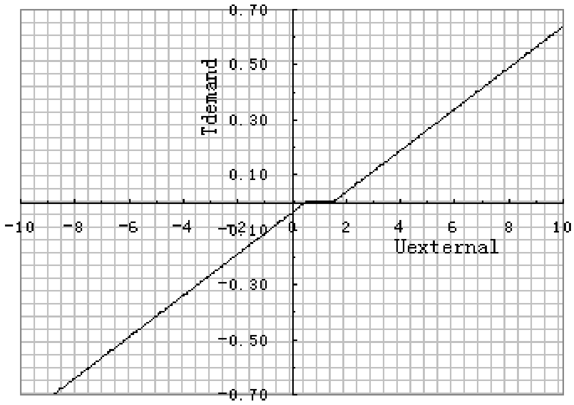


Fig. 8-14 Schematic diagram of Example 8-8

Calculate U_{filter} according to the offset voltage and dead zone voltage that require settings:

$$\frac{2047}{10v}=\frac{U_{filter}}{10v-U_{shift}-U_{dead}}$$ (In this example, $U_{dead}=0.5$, and $U_{shift}=1$)

Result: U_{filter} =1740

Calculate I_{demand} according to the required torque T_{demand} :

$$I_{demand} = \frac{T_{demand}}{K_t} * \sqrt{2}$$

Result: I_{demand} = 1.89

Calculate $Factor$ according to U_{filter} and I_{demand} :

$$Factor = \frac{I_{demand}}{U_{filter} * Ipeak} * 2048 * 4096$$

Result: $Factor = \frac{1.89}{1740 * 15} * 2048 * 4096 = 606$

Calculate $Ana\log 2\_Dead$ according to the required dead zone voltage:

$$Analog2\_Dead = \frac{8191}{10v} * U_{dead}$$

Result: $Ana\log 2\_Dead$ = 410

Calculate $Ana\log 2\_Offset$ according to the required offset voltage:

$$Analog2\_Offset = \frac{8191}{10v} * U_{shift}$$

Result: $Ana\log 2\_Offset$ =819

The following changes are required on the basis of Example 8-7.

Table 8-19 Parameter settings in Example 8-8

d3.26	Analog2_Dead	Sets dead zone data for external analog signal 2	Set to 410
d3.27	Analog2_Offset	Sets offset data for external analog signal 2	Set to 819
d3.31	Analog_Torque_Factor	Sets the proportion between analog signals and output torque (current)	Set to 2362
d3.00	Store_Loop_Data	1: Storing all configured parameters for the control loop 10: Initializing all parameters for the control loop	Set to 1

8.4Internal Multi-position Control Modes (“1” Mode)

In Internal multi-position control mode, we can activate internal set target position though an external signal to control motors. The activation has two preconditions:

1, multi-position control mode can only be activated in Mode 1, it can’t be activated in other modes.

2, At least one of the external input signal is defined as “Internal position control 0”, “Internal position control 1 “ or “Internal position control 2 “, which means at least one address of digital tubes-d3.01 ~ d3.07 is set to “040.0”’, “080.0” or “800.2.

“Internal position control 0” , “Internal position control 1” and “Internal position control 2 “, these three signals will be combined into binary codes used to select a target position between “Position 0~7”.

Internal position 0	Internal position 1	Internal position 2	Corresponding position	Position section numberic display	Corresponding speed	Speed section numberic display

0	0	0	Din_Pos0	d3.40select position section sequence number d3.41select position section high bit d3.42select position section low bit	Din_Speed0_RPM	d3.18
0	0	1	Din_Pos1		Din_Speed1_RPM	d3.19
0	1	0	Din_Pos2		Din_Speed2_RPM	d3.20
0	1	1	Din_Pos3		Din_Speed3_RPM	d3.21
1	0	0	Din_Pos4		Din_Speed4_RPM	d3.44
1	0	1	Din_Pos5		Din_Speed5_RPM	d3.45
1	1	0	Din_Pos6		Din_Speed6_RPM	d3.46
1	1	1	Din_Pos7		Din_Speed7_RPM	d3.47

Table 8-20 Internal Multi-position Control Mode Parameter Table

Note: In this control mode, "position section X" can be positive or negative, it can be flexibly set; while the corresponding speed must be positive. Other parameters such as acceleration, deceleration, etc, can use the default value; also can be changed through digital tube.

Example 8-9: Internal multi-position control mode

A motor needs to go eight position sections. In position section 0, it should reach the 5000 pulse location at the speed of 100RPM.In position section 1, it should reach the 15000 pulse location at the speed of 150RPM.In position section 2, it should reach the 28500 pulse location at the speed of 175RPM.In position section 3, it should reach the -105000 pulse location at the speed of 200RPM. In position section 4, it should reach the -20680 pulse location at the speed of 300RPM. In position section 5, it should reach the -30550 pulse location at the speed of 325RPM. In position section 6, it should reach the 850 pulse location at the speed of 275RPM. In position section 7, it should reach the 15000 pulse location at the speed of 460RPM.

Table 8-21 Internal Multi-position Control Mode Demand

DIN1	The driver is enabled, the motor shaft is locked
DIN3	Driver working mode（invalid 1，valid-3）
DIN4	Internal position 0
DIN5	Internal position 1
DIN6	Internal position 2
DIN6:DIN5:DIN4=0:0:0	Select position and speed in section 0
DIN6:DIN5:DIN4=0:0:1	Select position and speed in section 1
DIN6:DIN5:DIN4=0:1:0	Select position and speed in section 2
DIN6:DIN5:DIN4=0:1:1	Select position and speed in section 3
DIN6:DIN5:DIN4=1:0:0	Select position and speed in section 4
DIN6:DIN5:DIN4=1:0:1	Select position and speed in section 5
DIN6:DIN5:DIN4=1:1:0	Select position and speed in section 6
DIN6:DIN5:DIN4=1:1:1	Select position and speed in section 7

DIN6	Activate command（execute the selected position section）

Define the meanings of the input points:
Table 8-22 Internal Multi-position Control Mode Configuration

Numberic display	Variable name	Configuration way
d3.01	Din1_Function	000.1（Driver enabled）
d3.03	Din3_Function	000.4（Set driver mode）
d3.04	Din4_Function	040.0（Internal position control 0）
d3.05	Din5_Function	080.0（Internal position control 1）
d3.06	Din6_Function	800.2 (Internal position control 2)
d3.07	Din7_Function	400.0（Activate command）
d3.16	Din_mode 0	Set 0001 （1） Mode Internal multi-position control mode
d3.17	Din_mode 1	Set 0.004 (-4) Mode Pulse-control mode
d3.00	Storage parameters	1(Storage configuration parameters)

Set position and speed:
Table 8-23 Internal Multi-position and Speed Configuration

Numberic display	**Variable Name**	**Parameters Settings**
d3.43	Relative / Absolute position selection	Set to 2F(absolute location)
d3.40	Set the position section number to 0	Set to 0（select position section 0）
d3.41	Set the high bit of position section（N*10000）	Set to 0
d3.42	Set the low bit of position section	Set to 5000（set the position of section 0 t0 5000）
d3.18	Set the speed of section 0	Set to 100（set the speed of section 0 to 100）
d3.40	Set the position section number to 1	Set to 1（select position section 1）
d3.41	Set the high bit of position section（N*10000）	Set to 1
d3.42	Set the low bit of position section	Set to 15000（set the position of section 1 t0 15000））
d3.19	Set the speed of position section 1	Set to 150（set the speed of section 1 to 150）
d3.40	Set the position section number to2	Set to 2（select position section 2）
d3.41	Set the high bit of position section（N*10000）	Set to 2
d3.42	Set the low bit of position section	Set to 28500（set the position of section 2 t0 28500）
d3.20	Set the speed of position section 1	Set to 175（set the speed of

		section 2 to 175）
d3.40	Set the position section number to 3	Set to 3（select position section 3）
d3.41	Set the high bit of position section（N*10000）	Set to 3
d3.42	Set the low bit of position section	Set to 10500（set the position of section 3 t0 10500）
d3.20	Set the speed of position section 3	Set to 200（set the speed of section 3 to 200）
d2.10	Acceleration	Default 610 rps/s
d2.11	Deceleration	Default 610 rps/s
d3.00	Storage parameter	1 （ storage configuration parameters）

Set all these parameters, then：
1. Enable the driver, which means to make the digital input DIN1 high-level.
2. Select the position section, which means to change the electrical level of DIN4,DIN5 and DIN6.
3. Activate instructions and execute the program, which means to make the digital input DIN7 high-level.

Notice:
In multi-position control mode, select location method by setting the different value of the digital tube d3.43.If you choose absolute positioning mode, set it to "F"; if the instructions require immediate updating, set it to "2F"; if you choose relative positioning method, set it to "4F".To change these parameters successfully, you have to save the value of d3.00,and then restart.

8.5 Internal Multi-speed Control Modes ("-3" or "3" Mode)

In this control mode, external input signals are used to activate the internally configured target speed to control the motor. There are two prerequisites for activation:
1. Multi-speed control is available in the "-3" or "3" mode, and is unavailable in other modes.
2. Set d3.28 to 0. In this case, the analog – speed channel is invalid.
3. At least one external input signal DinX_Function defines Bit8 or Bit9.

For example, define Din2_Function corresponding to Din2 as 010.0, and Din3_Function corresponding to Din3 as 020.0. In this way, the combination of the two above signals is used to choose any one of Din_Speed0_RPM, Din_Speed1_RPM, Din_Speed2_RPM or Din_Speed3_RPM as the target speed.

Table 8-24 Parameters for internal multi-speed control modes

Internal Speed Control 0 (Din_Sys.Bit8)	Internal Speed Control 1 (Din_Sys.Bit9)	Meaning	Numeric Display	Valid Object (numeric display operation)
0	0	Multi-speed control: 0 [rpm]	d3.18	Din_Speed0_RPM
1	0	Multi-speed control 1 [rpm]	d3.19	Din_Speed1_RPM
0	1	Multi-speed control 2 [rpm]	d3.20	Din_Speed2_RPM
1	1	Multi-speed control 3 [rpm]	d3.21	Din_Speed3_RPM

Note: If you need to set the target speed precisely, it is required to set Din_Speed0, Din_Speed1, Din_Speed2 and Din_Speed3 with a host computer. The four data units are internal units and are suitable for users who are familiar with drivers. Din_SpeedX_RPM indicates the data after converting Din_SpeedX into the unit of rpm to facilitate users. Conversion involves both the reading and writing processes, and does not require calculation by users.

Example 8-10: Internal multi-speed control

Requirement: You need to define the digital input ports DIN6 and DIN7 as internal speed control, DIN1 as driver enabling and DIN2 as operation mode control of the driver (the mode is "3" when the driver is valid, and is "-3" when the driver is invalid). For detailed requirements, see Table 8-25. For the setting method, see

Table 7-26.

Table 8-25 Requirements on internal multi-speed control

DIN6:DIN7=0:0	To execute the multi-step 1 speed (100 rpm)
DIN6:DIN7=1:0	To execute the multi-step 2 speed (200 rpm)
DIN6:DIN7=0:1	To execute the multi-step 3 speed (300 rpm)
DIN6:DIN7=1:1	To execute the multi-step 3 speed (400 rpm)
DIN1	To enable the driver, and lock the motor shaft
DIN2	To control operation modes of the driver (the mode is “3” when the driver is valid, and is “-3” when the driver is invalid)

Table 8-26 Setting methods for internal multi-speed control

Numeric Display	Variable Name	Setting Method
d3.01	Din1_Function	Set to 000.1 (Driver enable)
d3.02	Din2_Function	Set to 000.4 (control over operation modes of drivers)
d3.06	Din6_Function	Set to 010.0 (internal speed control 0)
d3.07	Din7_Function	Set to 020.0 (internal speed control 1)
d3.16	Din_Mode0	Set to 0.003 (3) mode (speed mode with acceleration/deceleration)
d3.17	Din_Mode1	Set to 0.003 (-3) mode (instantaneous speed mode)
d3.18	Din_Speed0_RPM	Set to 100 [rpm]
d3.19	Din_Speed1_RPM	Set to 200 [rpm]
d3.20	Din_Speed2_RPM	Set to 300 [rpm]
d3.21	Din_Speed3_RPM	Set to 400 [rpm]
d3.00	Store_Loop_Data	Set to 1

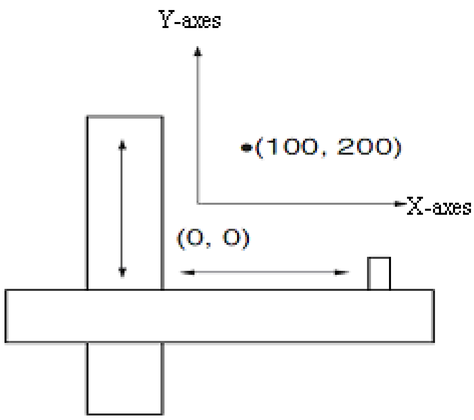
8.6 Internal Torque Control Mode (“4” Mode)

In the internal torque mode, only the current loop of the driver operates. Set d0.03 (CMD_q target current) parameter directly to obtain the desired target torque. The prerequisite is that d3.30 must be set to 0. In this case, the analog–torque channel is invalid.

8.7 Homing Mode (“6” Mode)

1, Summary

To make a system execute positioning in accordance with its absolute positioning, the first step is to define the origin. For instance, as shown in the following XY plane, to navigate to (X, Y) = (100mm, 200mm), you must define the origin of the machine firstly. It’s necessary to define the origin.



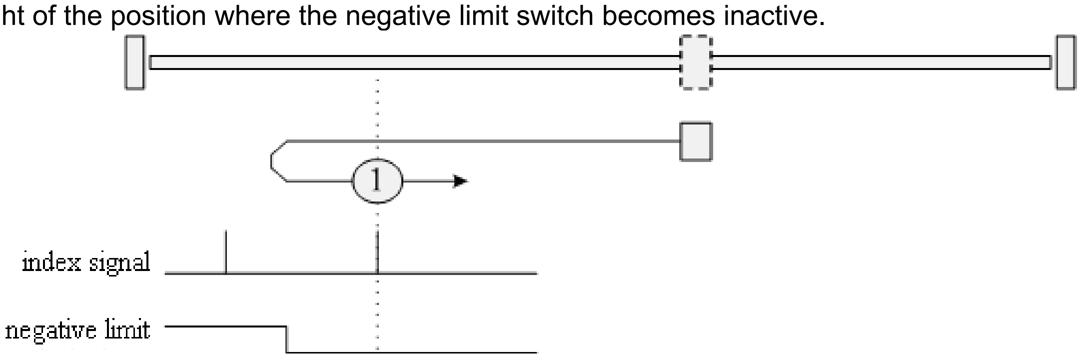
2, Procedure of homing
Use the following steps to homing:
1. Set the external I / O parameters, and then save.
2. Set the data for homing, and then save.
3. Execute homing.

3, Configuration of the data for homing
Here are simple descriptions of the data for executing homing.

0x607C0020	Home_Offset	Home offset	In Homing mode, set the offset relative to the zero point.
0x60980008	Homing_Method	Homing method	Select the homing method
0x60990120	Homing_Speed_Switch	Speed for searching the limit switch	Set the speed for searching the limit switch which defined as homing signal.
0x60990220	Homing_Speed_Zero	Speed for searching the Zero point.	Only valid when find Index signal.
0x60990308	Homing_Power_On	Homing when power on	Every time after power on,it will start homing once.
0x609A0020	Homing_Accelaration	Homing acceleration	Control the acceleration of homing

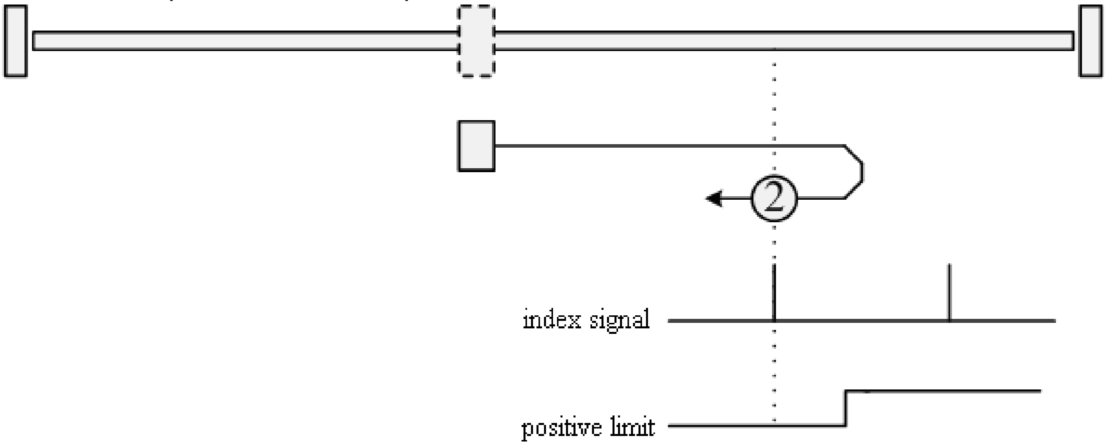
CD has 27 methods for homing, referring the CANopen's definition of DSP402.
1st-14th methods use Z signal as homing signal.
17th-30th methods use external signal as homing signal.

Method 1: Homing on the negative limit switch and index pulse
Using this method, the initial direction of movement is leftward if the negative limit switch is inactive (here shown as low). The home position is at the first index pulse to the right of the position where the negative limit switch becomes inactive.



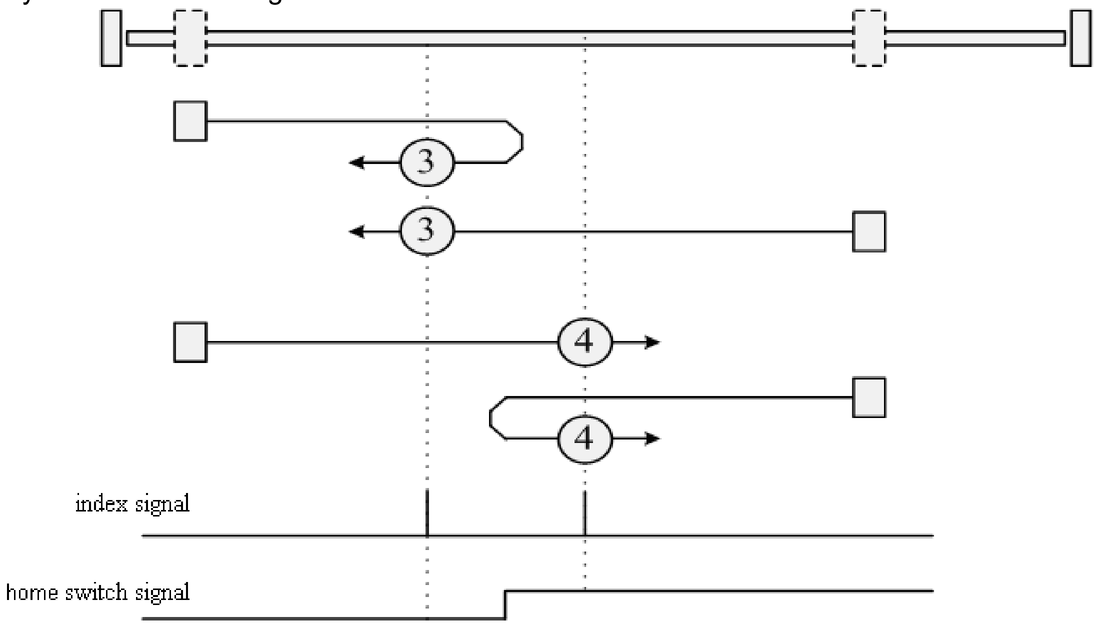
Method 2: Homing on the positive limit switch and index pulse

Using this method, the initial direction of movement is rightward if the positive limit switch is inactive (here shown as low). The position of home is at the first index pulse to the left of the position where the positive limit switch becomes inactive.



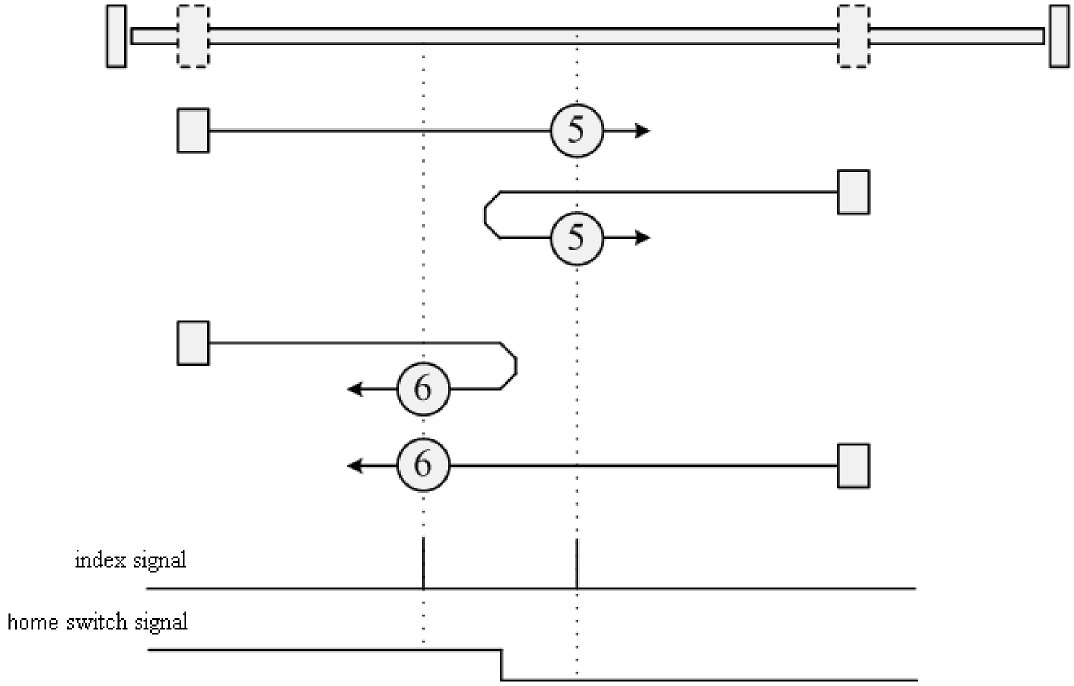
Methods 3 and 4: Homing on the positive home switch and index pulse

Using methods 3 or 4, the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either the left or right of the pint where the home switch changes state. If the initial position is sited so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.



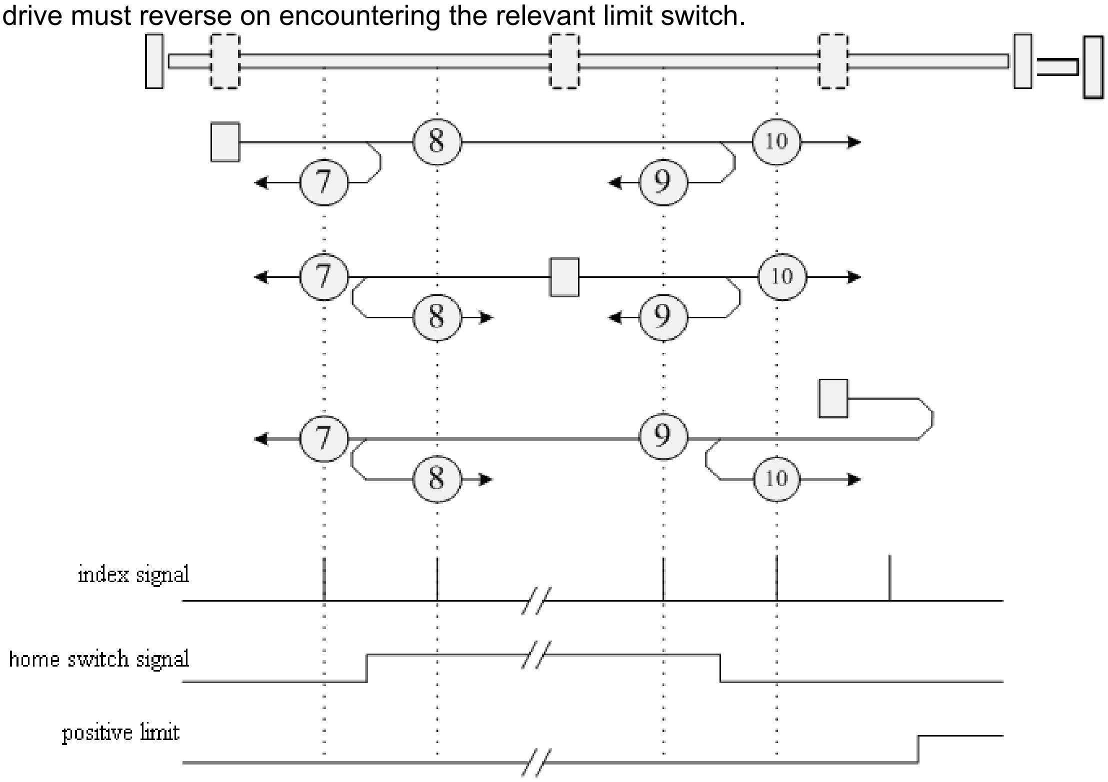
Methods 5 and 6: Homing on the negative home switch and index pulse

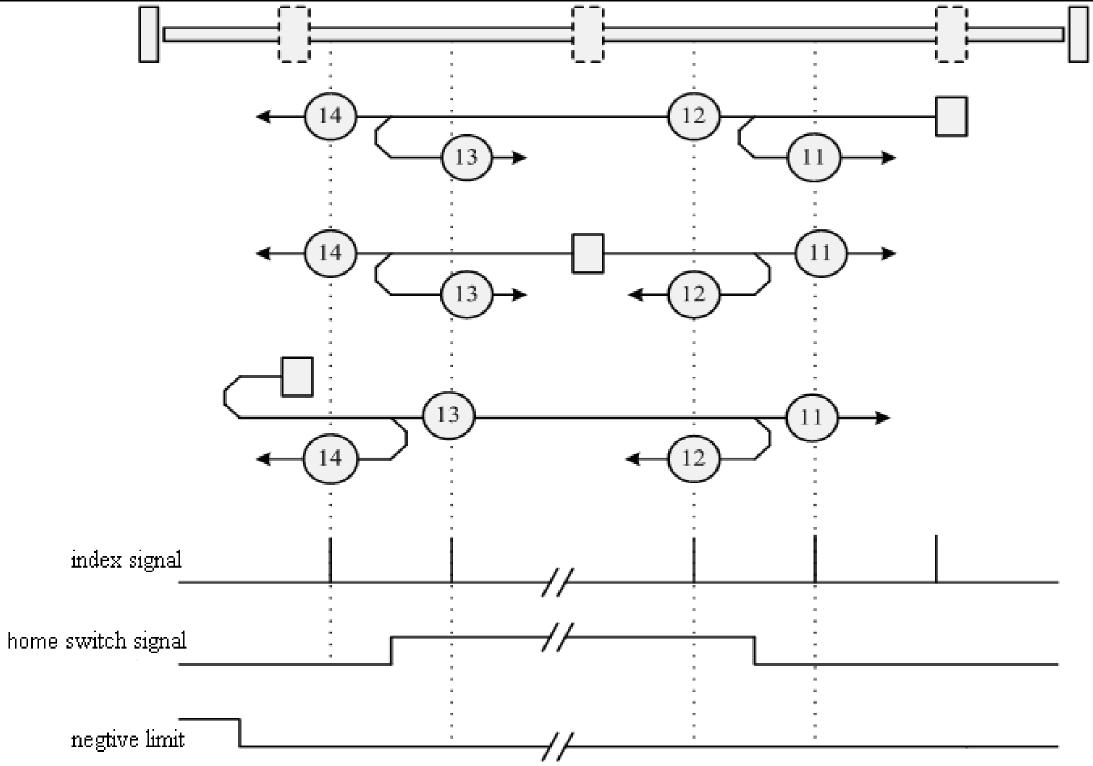
Using methods 5 or 6, the initial direction of movement is dependent on the state of the home switch. The home position is at the index pulse to either the left or the right of the point where the home switch changes state. If the initial position is sited so that the direction of movement must reverse during homing, the point at which the reversal takes place is anywhere after a change of state of the home switch.



Methods 7 to 14: Homing on the home switch and index pulse

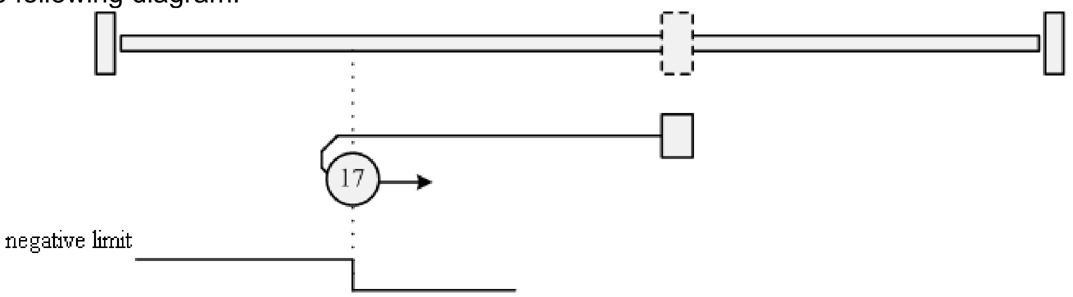
These methods use a home switch that is active over only a portion of the travel; in effect the switch has a "momentary" action as the axle position sweeps past the switch.
Using methods 7 to 10, the initial direction of movement is to the right, and using methods 11 to 14, the initial direction of movement is to the left, except if the home switch is active at the start of motion. In this case, the initial direction of motion is dependent on the edge being sought. The home position is at the index pulse on either side of the rising or falling edges of the home switch, as shown in the following two diagrams. If the initial direction of movement leads away from the home switch, the drive must reverse on encountering the relevant limit switch.

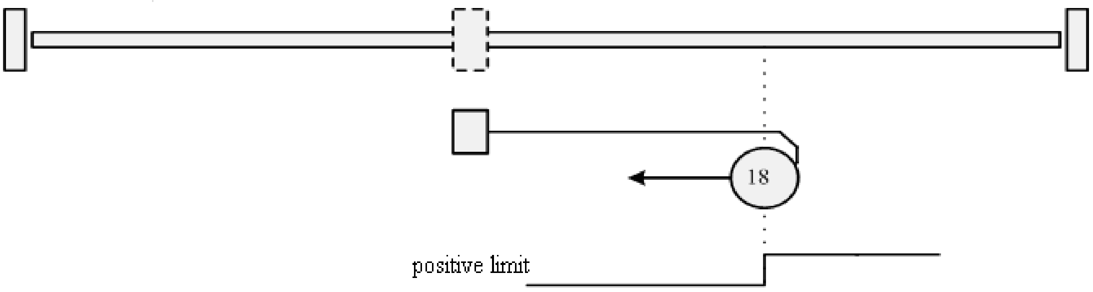


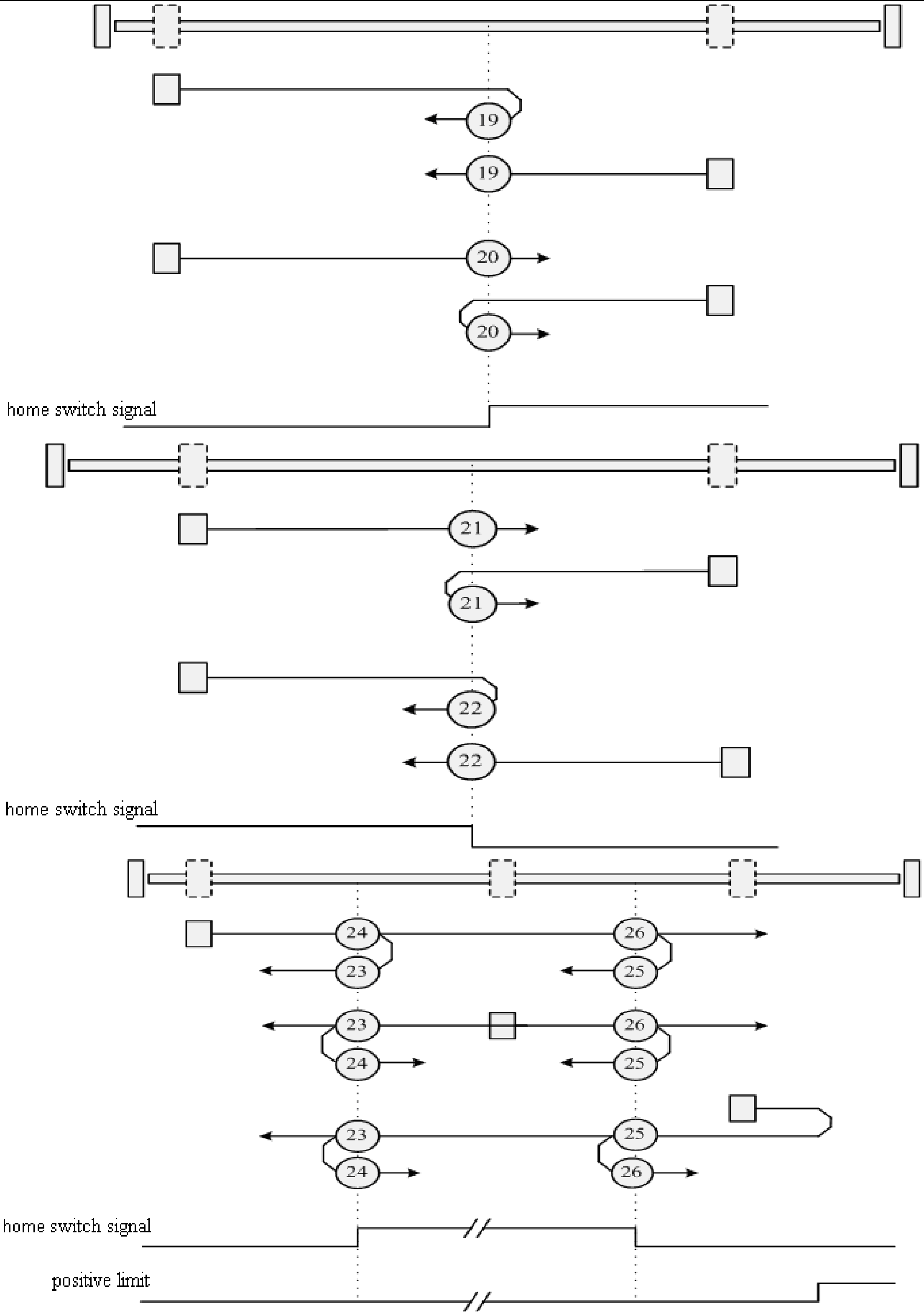


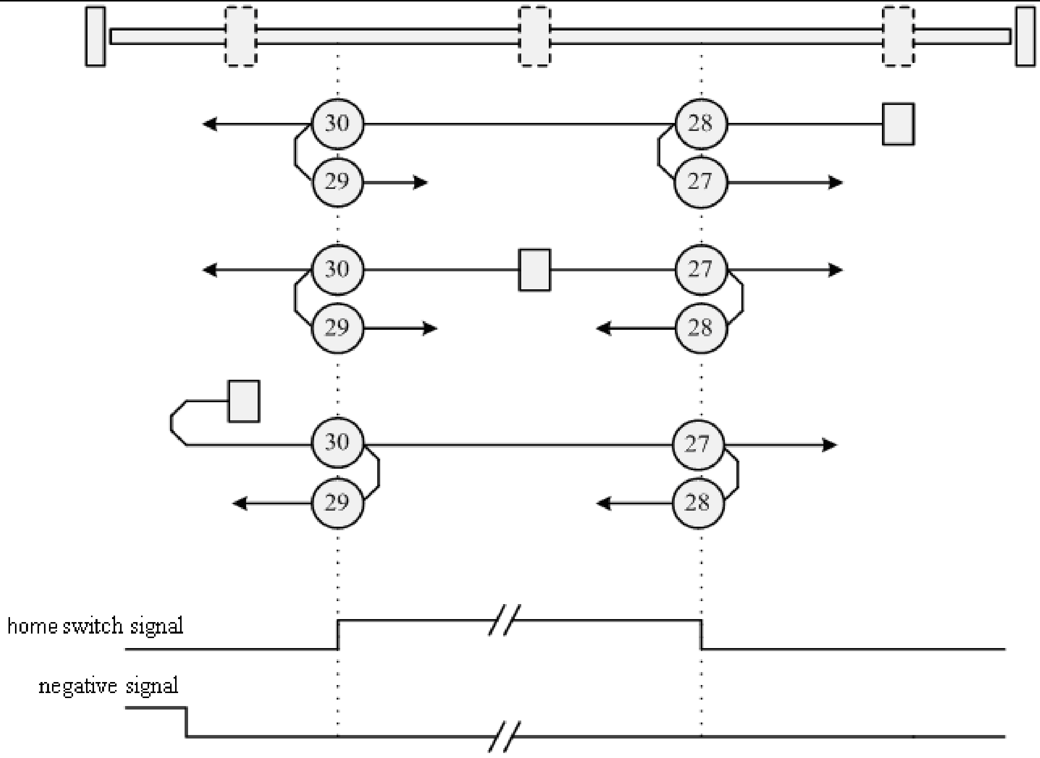
Methods 15 and 16: Reserved
These methods are reserved for future expansion of the homing mode.

Methods 17 to 30: Homing without an index pulse
These methods are similar to methods 1 to 14, except that the home position is not dependent on the index pulse; it is dependent only on the relevant home or limit switch transitions. For example, methods 19 and 20 are similar to methods 3 and 4, as shown in the following diagram:



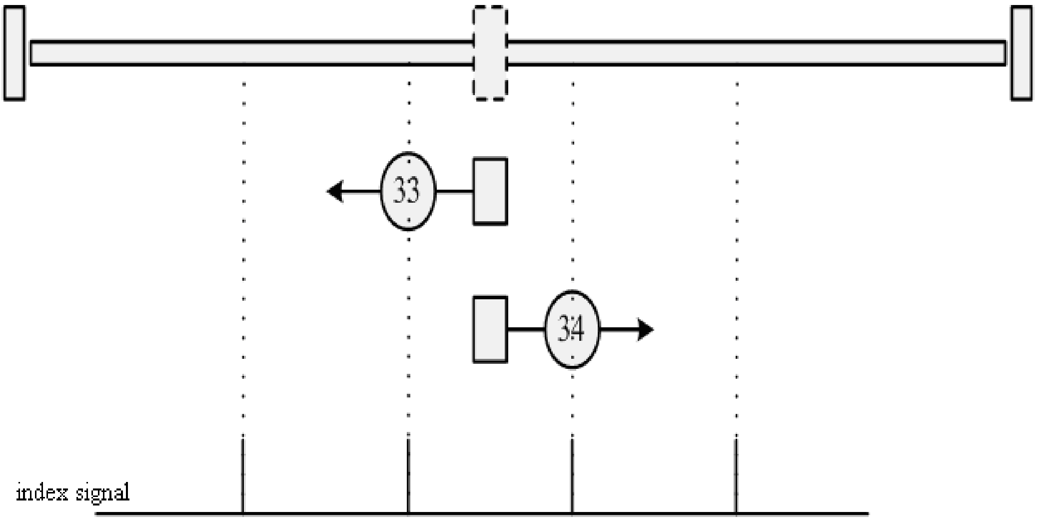



19
19
20
20
home switch signal
21
21
22
22
home switch signal
24
26
23
25
23
26
24
25
25
23
24
26
home switch signal
positive limit



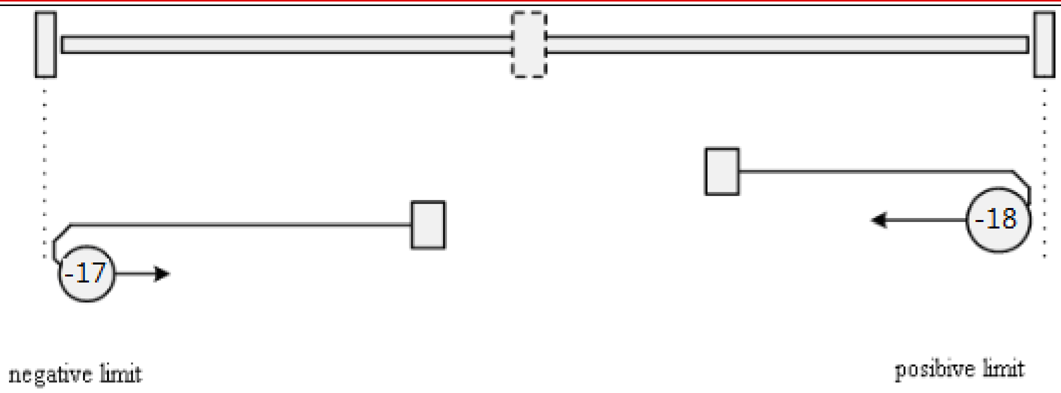
Methods 31 and 32: Reserved
These methods are reserved for future expansion of the homing mode.

Methods 33 and 34: Homing on the index



Method 35: Homing on the current position
In this method, the current position is taken to be the home position.

Methods -17 and -18: Use the mechanical terminal as reference point

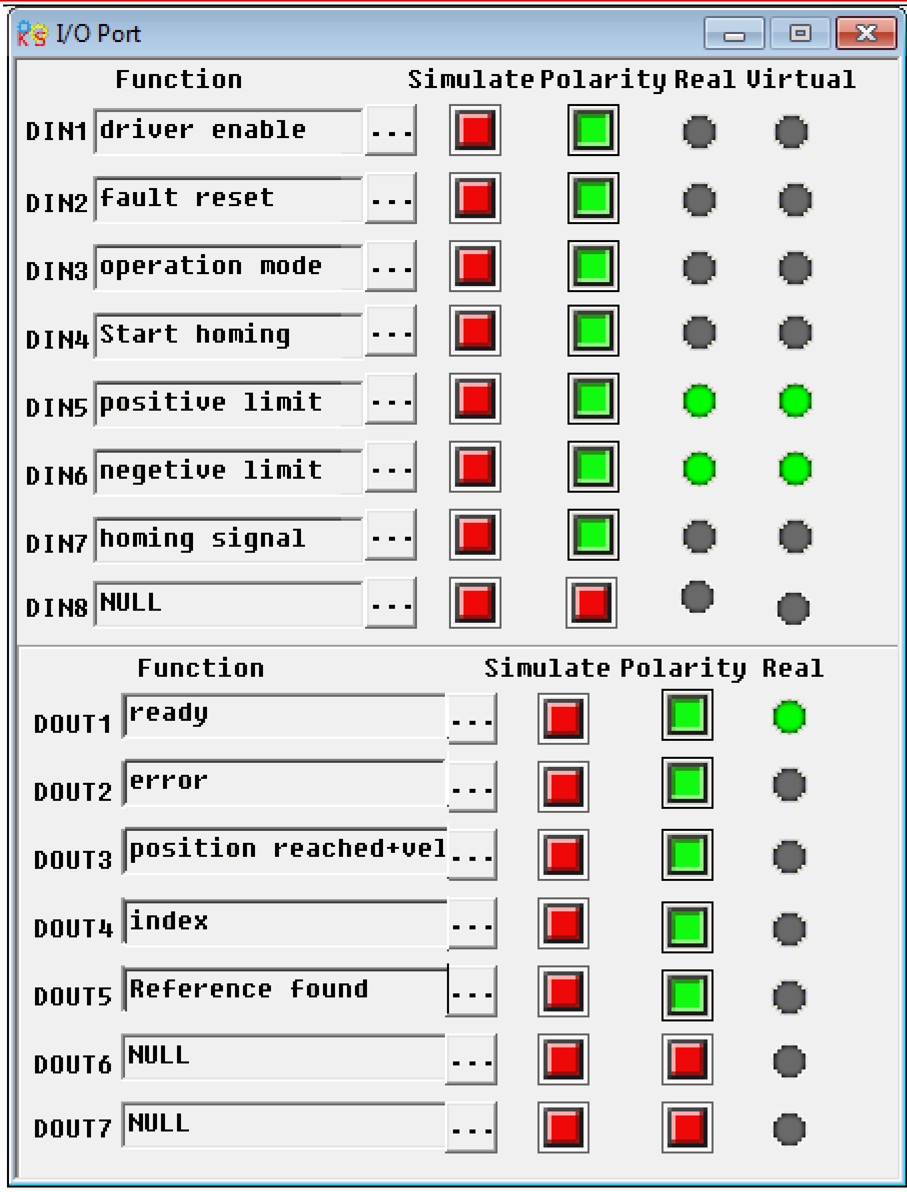


Example 8-11：Using method 7 for homing.

Set parameters.

Numberic display	Parameter Name	meaning	Setting Value
d3.01	Din1_Function	000.1: Driver enabled 000.2: Driver error reset 000.4: Operation mode 001.0:Positive limit 002.0:Negative limit 004.0:Origin signal 200.0:Start homing	000.1 （Driver enabled）
d3.02	Din2_Function		000.2 （Driver error reset）
d3.03	Din3_Function		000.4 （Driver model control）
d3.04	Din4_Function		200.0 （Start homing）
d3.05	Din5_Function		001.0 （Positive limit）
d3.06	Din6_Function		002.0 （Negative limit）
d3.07	Din7_Function		004.0 （Home signal）
d3.14	Dout4_Function	004.0:Index signal appears	004.0 (Index signal appears)
d3.15	Dout4_Function	040.0:Origin found	040.4 (origin found)
d3.16	Din_Mode0	Select this mode when the input signal is invalid	0.004 (-4)
d3.17	Din_Mode1	Select this mode when the input signal is valid	0.003 (-3)
d3.00	Store_Loop_Data	1：Storage all the setting parameters except those of motor 10：Initialize all the setting parameters except those of motor	0001 (1)

At this time, computer software shows:



Notice: The positive and negative limits are default to normally closed point. Otherwise, the Panel will alarm and display P.L (positive limit) and N.L (No limit). Only when the alarm is eliminated, the origin control mode can be normally used.

Computer monitoring status is：

Basic Operate

	name	data	unit
1*	Operation_Mode_Buff	-4	DEC
2*	Status_Word	4437	HEX
3*	Pos_Actual	0	inc
4*	Real_Speed_RPM	0	rpm
5*	I_q	0.000	Ap
6	Operation_Mode	-4	DEC
7	CMD_q	0.000	Ap
8	Pos_Target	0	inc
9	SpeedDemand_RPM	0	rpm
10	Control_Word	2f	HEX
11	Switch_On_Auto	0	DEC
12	CMD_q_Max	6.797	Ap

Set parameters for homing.

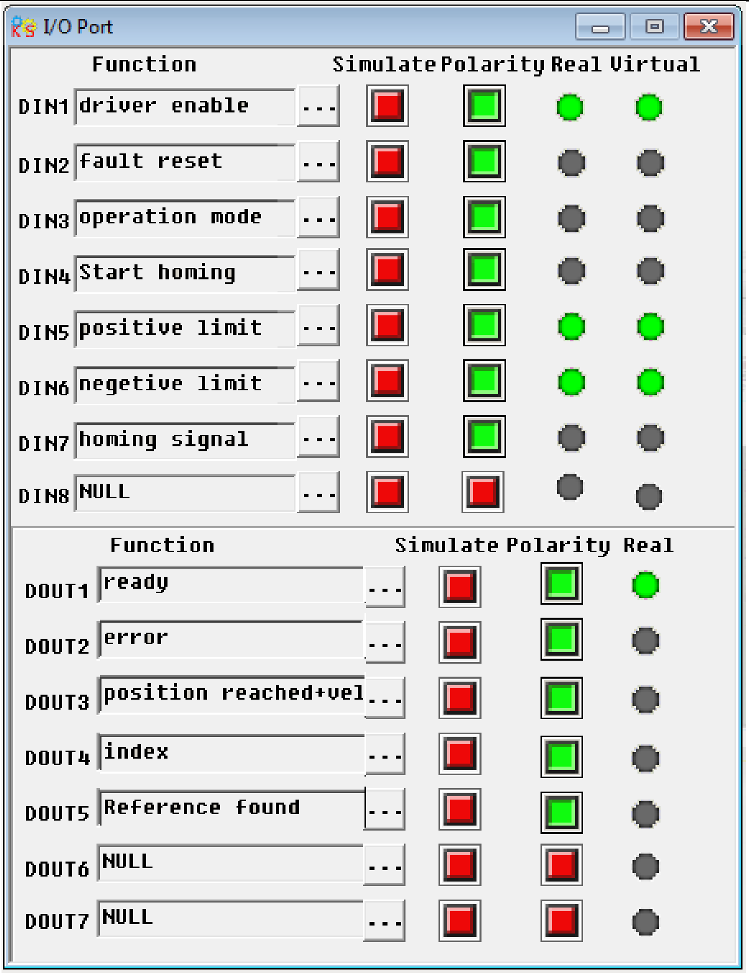
Homing Mode

	name	data	unit
1	Home_Offset	0	inc
2	Homing_Method	7	DEC
3	Homing_Speed_Switch	150.000	rpm
4	Homing_Speed_Zero	100.000	rpm
5	Homing_Power_On	0	DEC
6	Homing_Accelaration	50.000	rps/s
7	Homing_Current	1.780	Ap

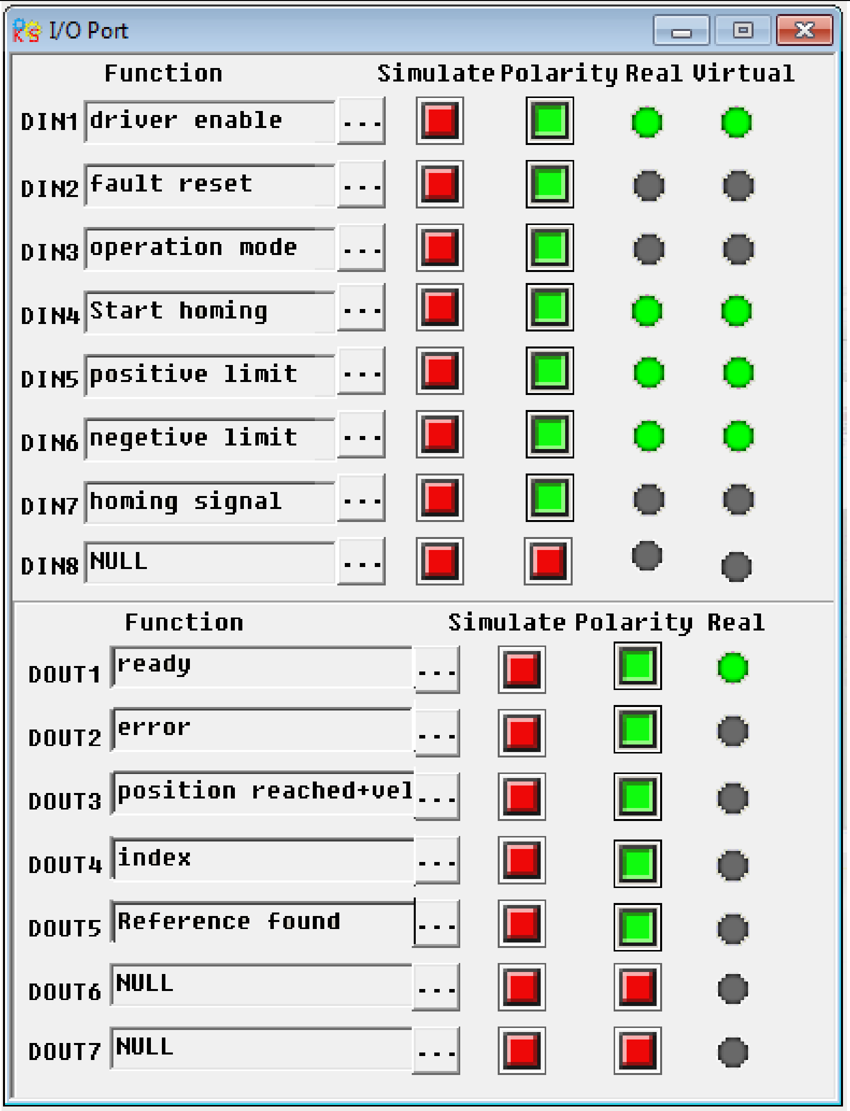
In common circumstance, only need to set up the model of origin and the rest of the parameters are default. In some case, “Electrify and then find the origin” is set to 1, at the same time the definition-- “Start finding the origin” is eliminated.

Start homing.

(1). Enable motor, which means the digital input point 1 is set to high-level. The computer motoring picture is shown below:

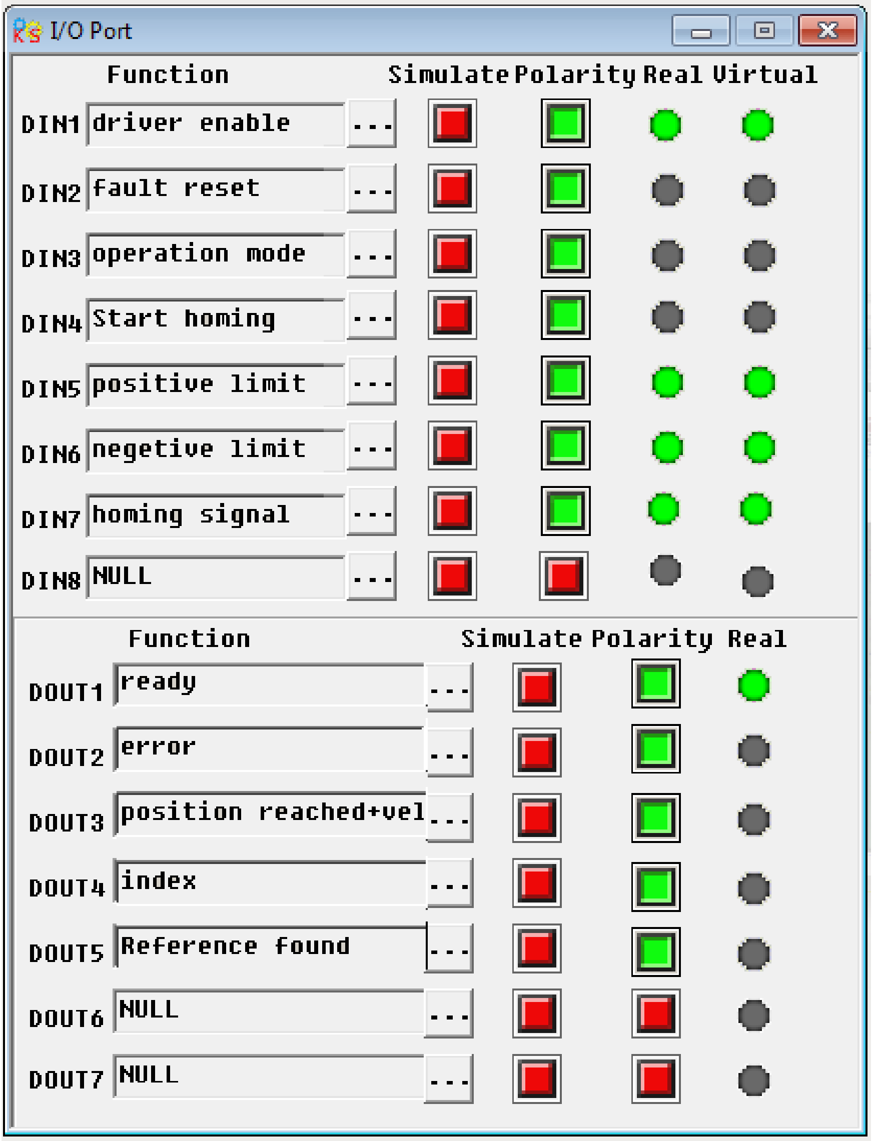


(2). Send "Start finding the origin" signal to motor, which means the digital input point 4 is set to high-level. The computer motoring picture is shown below:

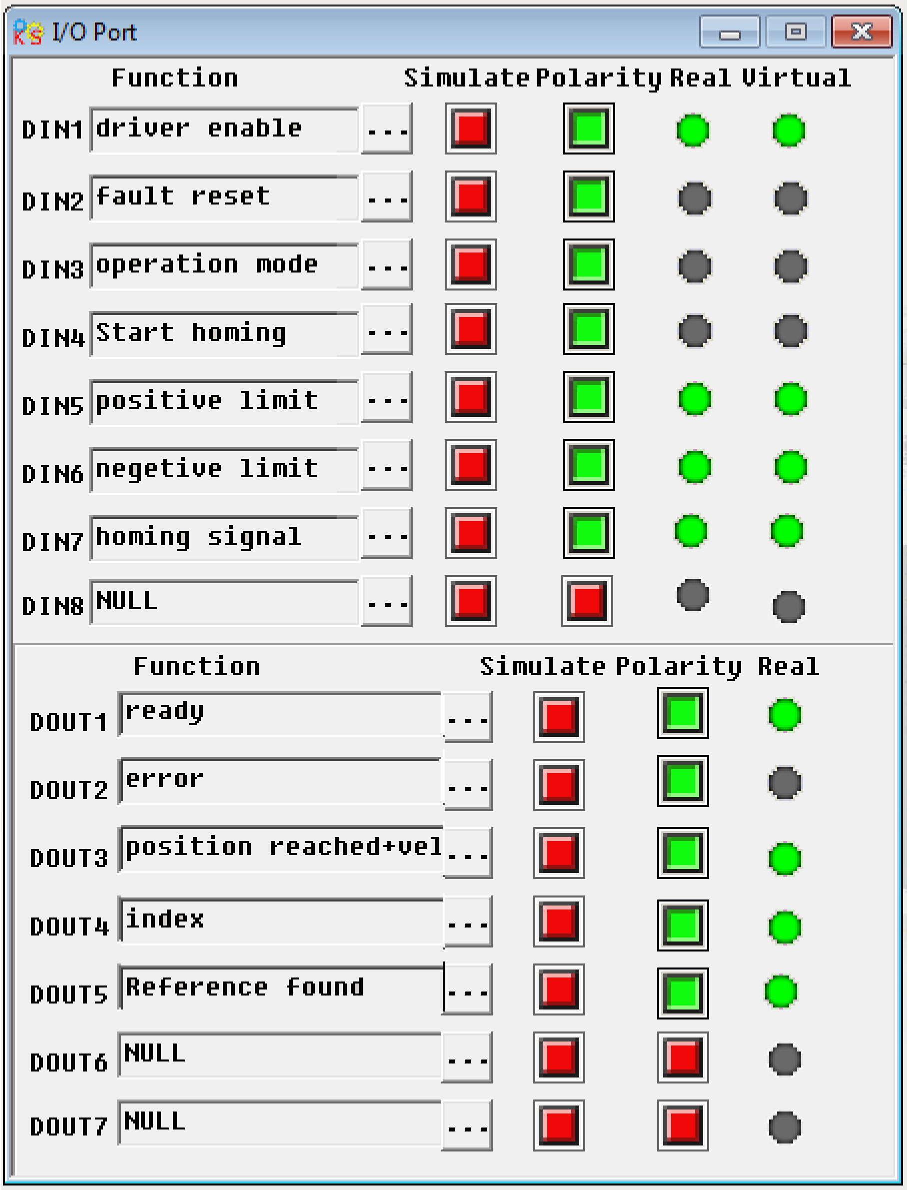


Note: "Start finding the origin" signal is a pulse signal, requires only a rise, not need to always be on. If you want to start next time, a rise pulse is enough.

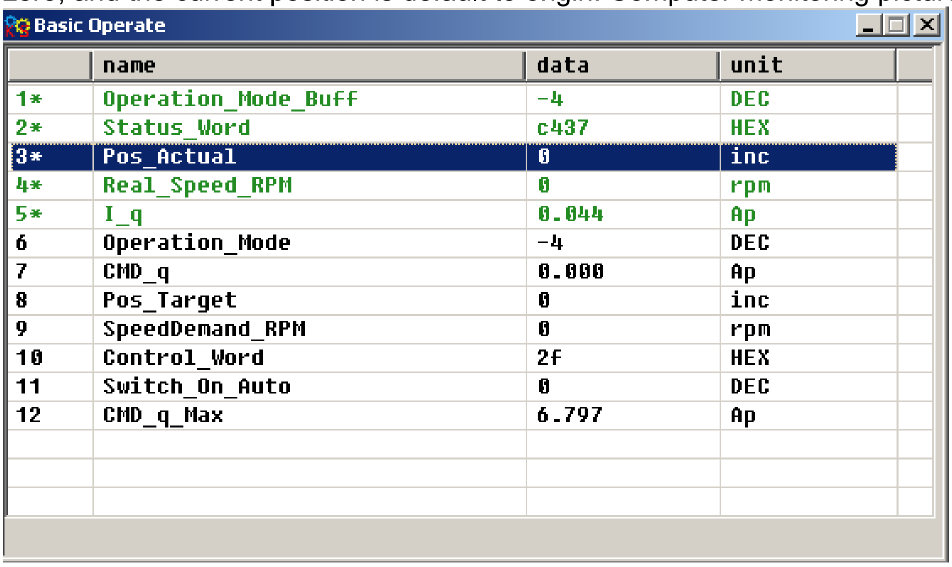
(4). After the external find the origin, computer monitoring picture is as follows:



(5). Driver searches the Z phase signal in mode 7, and ultimately find the origin. Computer monitoring picture is shown as follows:



At this point, you have completed the origin search function, then the drive position is automatically set to zero, and the current position is default to origin. Computer monitoring picture is as shown:


Basic Operate

	name	data	unit
1*	Operation_Mode_Buff	-4	DEC
2*	Status_Word	c437	HEX
3*	Pos_Actual	0	inc
4*	Real_Speed_RPM	0	rpm
5*	I_q	0.044	Ap
6	Operation_Mode	-4	DEC
7	CMD_q	0.000	Ap
8	Pos_Target	0	inc
9	SpeedDemand_RPM	0	rpm
10	Control_Word	2f	HEX
11	Switch_On_Auto	0	DEC
12	CMD_q_Max	6.797	Ap

Chapter 9 Control Performance

9.1 Auto Reverse

In this mode,motor will run forward and reverse continuously according to the setting mode.User can set parameters in velocity loop and position loop in this mode.Please make sure auto forward/reverse is allowed in the machine before using this mode and make sure the power of driver can be cut off anytime to advoid accident.

Operation procedure for auto reverse:

1：Use KincoServo software to online according to chapter 5.

2：Set speed mode control according to 5.4.1.

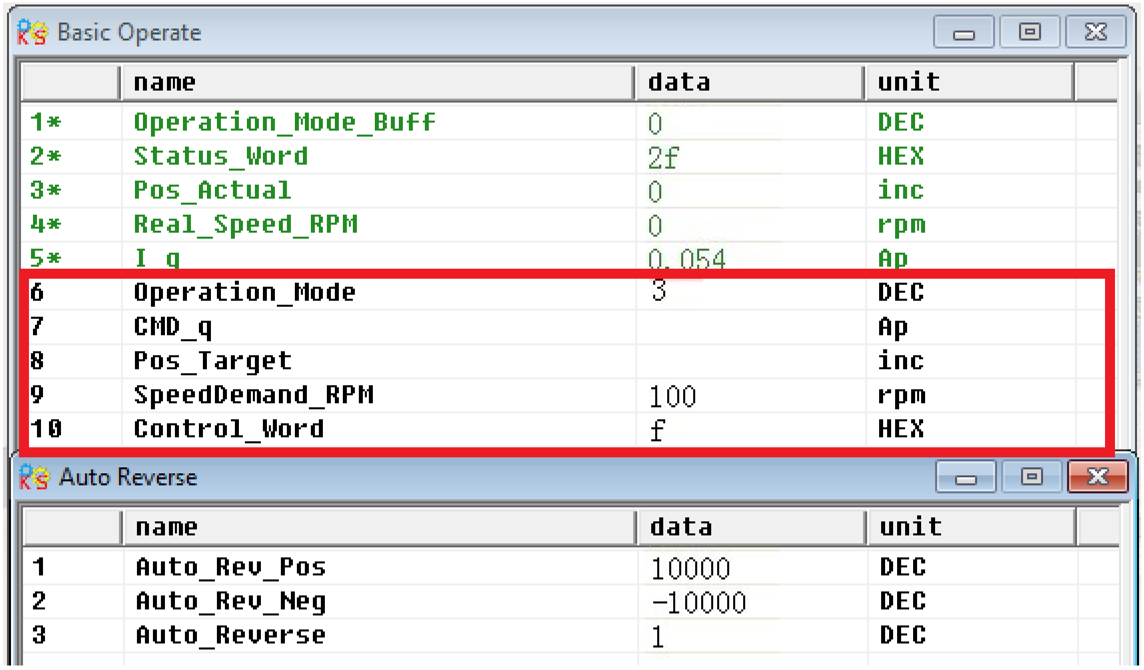
3：Click the menu “Driver-Operation mode-Auto Reverse” and set the parameter for auto reverse.

Set “Auto_Reverse” as 0 for no control.

Set “Auto_Reverse” as 1 for position control.The motor will run between the position “Auto_Rev_Pos” and”Auto_Rev_Neg”.The unit is inc.The speed depends on target velocity.

Set “Auto_Reverse” as 3 for time control.The motor will run between time “Auto_Rev_Pos” and”Auto_Rev_Neg”.The unit is ms.The speed depends on target velocity.

Following figure shows the parameters need to set.In this figure,the servo will run between -10000 inc and 10000 at speed 100RPM.



Basic Operate

	name	data	unit
1*	Operation_Mode_Buff	0	DEC
2*	Status_Word	2f	HEX
3*	Pos_Actual	0	inc
4*	Real_Speed_RPM	0	rpm
5*	I_q	0.054	Ap
6	Operation_Mode	3	DEC
7	CMD_q		Ap
8	Pos_Target		inc
9	SpeedDemand_RPM	100	rpm
10	Control_Word	f	HEX

Auto Reverse

	name	data	unit
1	Auto_Rev_Pos	10000	DEC
2	Auto_Rev_Neg	-10000	DEC
3	Auto_Reverse	1	DEC

9.2 Driver Performance Tuning

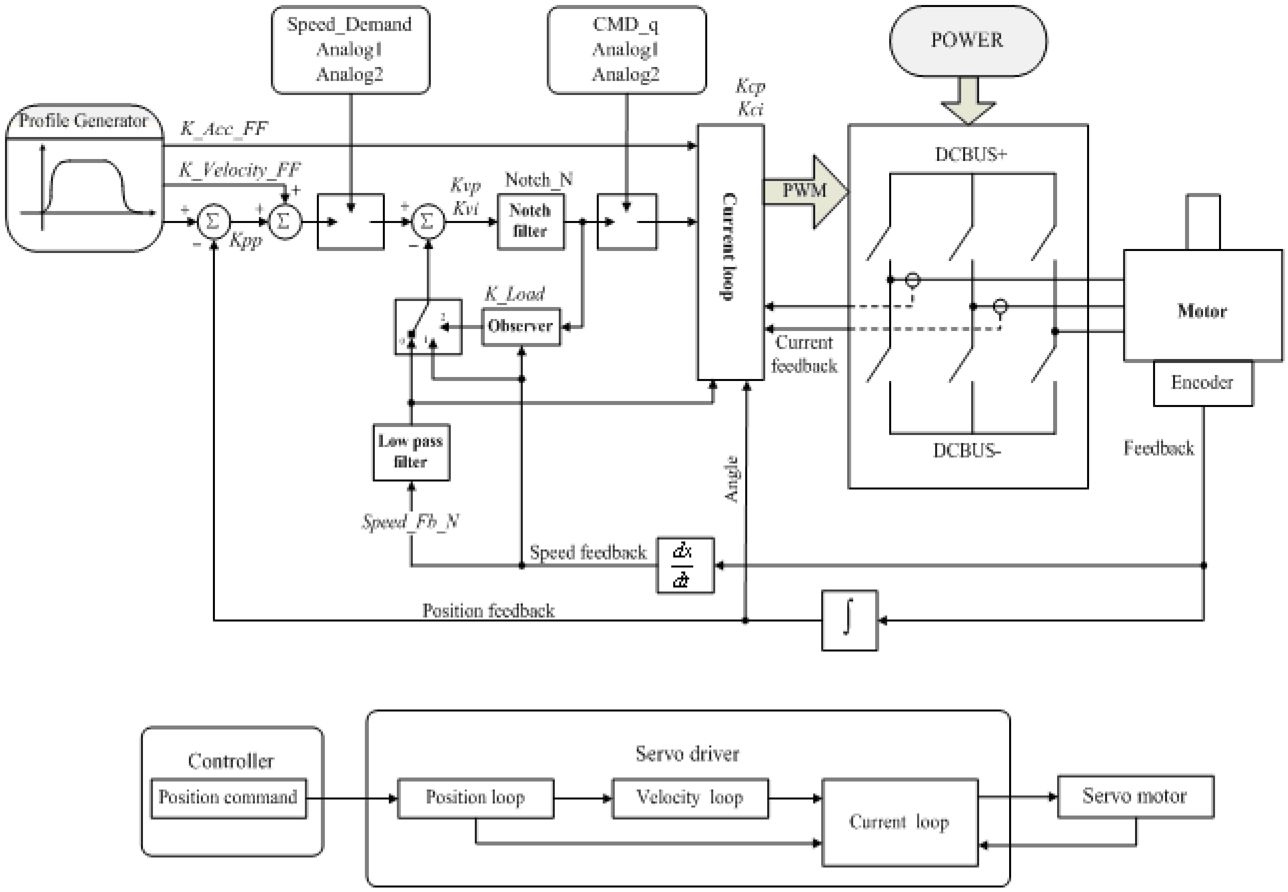


Fig. 9-1 Schematic diagram for control loop adjustment

As shown in Fig. 9-1, a typical servo system contains three control loops, namely, position loop, velocity loop, and current loop.

Current loop are related to motor parameters (optimal parameters of the selected motor are default for the driver and no adjusting is required).

Parameters for velocity loop and position loop should be adjusted properly according to load conditions.

During adjustment of the control loop, ensure that the bandwidth of the velocity loop is at least twice of that of the position loop; otherwise oscillation may occur.

9.2.1 Manual Adjustment

1. Parameters for velocity loop

Table 9-1 Parameters for velocity loop

Numeric Display	Variable Name	Meaning	Default Value	Range
d2.01	Kvp	Sets the response speed of a velocity loop	100	0~32767
d2.02	Kvi	Adjusts speed control so that the time of minor errors is compensated	2	0~16384
d2.05	Speed_Fb_N	Reduces the noise during motor operation by reducing the feedback bandwidth of velocity loops (smoothing feedback signals of encoders). When the set bandwidth becomes smaller, the motor responds slower. The formula is F=Speed_Fb_N*20+100.	45	0~45

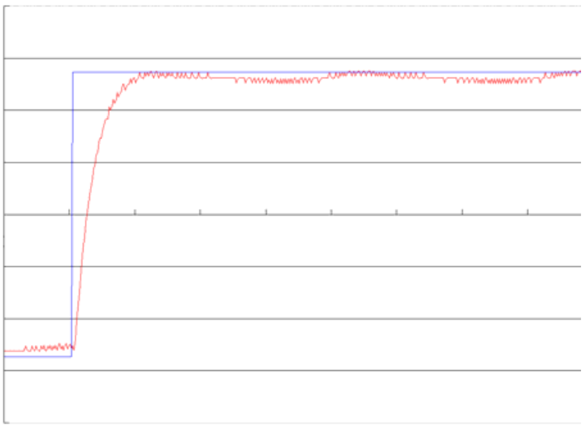
		For example, to set the filter bandwidth to "F = 500 Hz”, you need to set the parameter to 20.		

Proportional gain of velocity loop Kvp: If the proportional gain of the velocity loop increases, the responsive bandwidth of the velocity loop also increases. The bandwidth of the velocity loop is directly proportional to the speed of response. Motor noise also increases when the velocity loop gain increases. If the gain is too great, system oscillation may occur.
Integral gain of velocity loop Kvi: If the integral gain of the velocity loop increases, the low-frequency intensity is improved, and the time for steady state adjustment is reduced; however, if the integral gain is too great, system oscillation may occur.

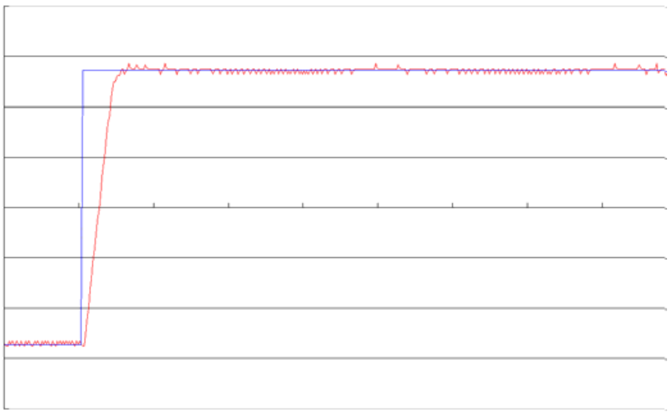
Adjustment steps:
Step 1: Adjust the gain of velocity loop to calculate the bandwidth of velocity loop
Convert the load inertia of the motor into the inertia Jl of the motor shaft, and then add the inertia Jr of the motor itself to obtain Jt = Jr + Jl. Put the result into the formula:

$$\text{Vc\_Loop\_BW} = \text{Kvp} * \frac{I_p * K_t * Encoder\_R}{J_t * 204800000 * \sqrt{2} * 2\pi}$$

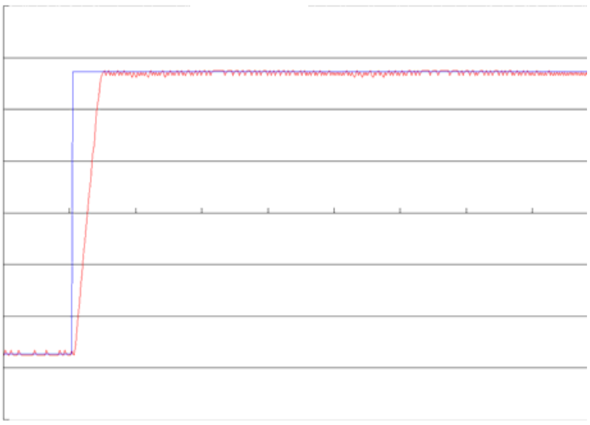
To calculate the bandwidth of the velocity loop Vc_Loop_BW according to the adjusted the gain of velocity loop Kvp, only adjust Kvi according to actual requirements.
Adjust the impact of Kvp and Kvi, as shown in Fig.9-2.
For the effect of Kvp adjustment, see the first to the fourth from left of Fig. 9-2. Kvp gradually increases from the first to the fourth from left. The value of Kvi is 0.
For the effect of Kvi adjustment, see the first to the fourth from right of Fig. 9-2. Kvi gradually increases from the first to the fourth from right. The value of Kvp remains unchanged.



Left 1

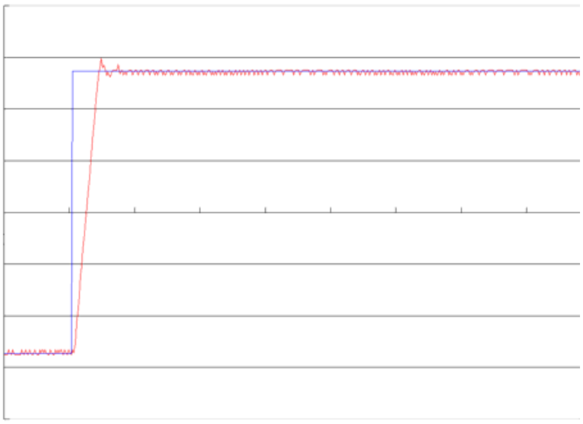


Right 1

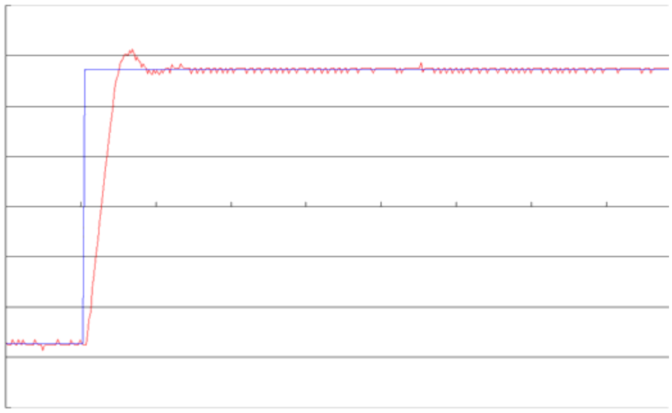


Left 2

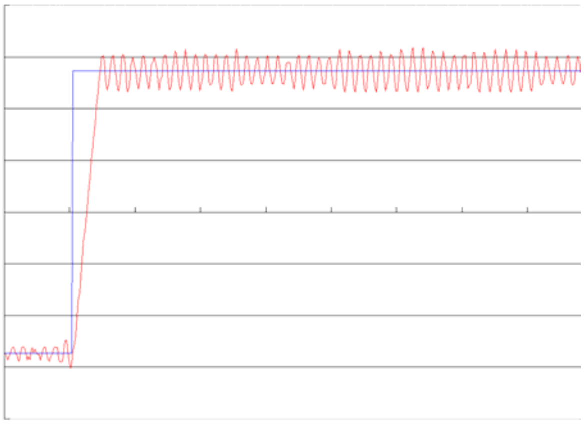
Right 2



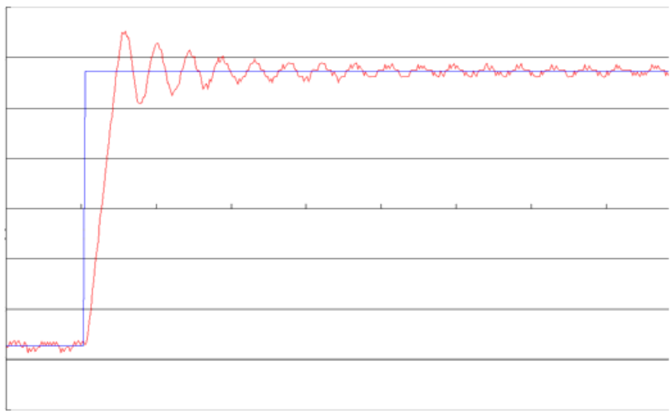
Left 3



Right 3



Left 4



Right 4

Fig.9-2 Schematic diagram of gain adjustment of velocity loop

Step 2: Adjust parameters for feedback filter of velocity loop

During gain adjustment of a velocity loop, if the motor noise is too great, you can properly reduce the parameter Speed_Fb_N for feedback filter of the velocity loop; however, the bandwidth F of the feedback filter of velocity loop must be at least three times of the bandwidth of velocity loop; otherwise oscillation may occur. The formula for calculating the bandwidth of feedback filter of velocity loop is F = Speed_Fb_N*20+100 (Hz).

2. Parameters for position loop

Table 9-2 Parameters for position loop

Numeric Display	Variable Name	Meaning	Default Value	Range
d2.07	Kpp	Indicates the proportional gain of the position loop Kpp	1000	0~16384
d2.08	K_Velocity_FF	0 indicates no feedforward, and 256 indicates 100% feedforward	256	0~256
d2.09	K_Acc_FF	The value is inversely proportional to the feedforward	7FF.F	32767~10
d0.05	Pc_Loop_BW	Sets the bandwidth of the position loops in Hz	0	N/A
/	Pos_Filter_N	Set the average filter	1	1~255

Proportional gain of the position loop Kpp: If the proportional gain of the position loop increases, the bandwidth of the position loop is improved, thus reducing both the positioning time and following errors. However, too great bandwidth may cause noise or even oscillation. Therefore, this parameter must be set properly according to loading conditions. In the formula Kpp=103* Pc_Loop_BW, Pc_Loop_BW indicates the bandwidth of the position loop. The bandwidth of a position loop is less than or equal to that of a velocity loop. It is recommended that Pc_Loop_BW be less than Vc_Loop_BW /4 (Vc_Loop_BW indicates the bandwidth of a velocity loop).

Velocity feedforward of the position loop K_Velocity_FF: the velocity feedforward of a position loop can be increased to reduce position following errors. When position signals are not smooth, if the velocity feedforward of a position loop is reduced, motor oscillation during running can be reduced.
Acceleration feedback of the position loop K_Acc_FF (adjustment is not recommended for this parameter): If great gains of position rings are required, the acceleration feedback K_Acc_FF can be properly adjusted to improve performance. $\mathrm{K\_Acc\_FF} = \frac{I_p * K_t * Encoder\_R}{250000 * \sqrt{2} * J_t * \pi}$ Note: K_Acc_FF is inversely proportional to the acceleration feedforward.
Pos_Filter_N is used for average filter of the speed produced by target position.Setting this parameter as N means to average N data.

Adjustment procedure:
Step 1: Adjust the proportional gain of a position loop.
After adjusting the bandwidth of the velocity loop, it is recommended to adjust Kpp according to actual requirements (or directly fill in the required bandwidth in Pc_Loop_BW, and the driver will automatically calculate the corresponding Kpp). In the formula Kpp = 103*Pc_Loop_BW, the bandwidth of the position loop is less than or equal to that of the velocity loop. For a common system, Pc_Loop_BW is less than Vc_Loop_BW /2; for the CNC system, it is recommended that Pc_Loop_BW is less than Vc_Loop_BW /4.
Step 2: Adjust velocity feedforward parameters of the position loop.
Velocity feedforward parameters (such as K_Velocity_FF) of the position loop are adjusted according to position errors and coupling intensities accepted by the machine. The number 0 represents 0% feedforward, and 256 represents 100% feedforward.

3. Parameters for pulse filtering coefficient

Table 9-3 Parameters for pulse filtering coefficient

Numeric Display	Variable Name	Meaning	Default Value	Range
d3.37	PD_Filter	Used to smooth the input pulses. Filter frequency: f = 1000/(2π* PD_Filter) Time constant: T = PD_Filter/1000 Unit: S Note: If you adjust this filter parameter during the operation, some pulses may be lost.	3	1~32767

When a driver operates in the pulse control mode, if the electronic gear ratio is set too high, this parameter must be adjusted to reduce motor oscillation; however, if the parameter adjustment is too great, motor running instructions will become slower.

9.2.2 Auto Adjustment (Only for Velocity Loops)

Auto adjustment is only available for velocity loops (see Section 8.11 for manual adjustment of position loops) when both forward rotation and reverse rotation of a motor are allowable, and the loadings do not change much during the operation. You can determine the total inertia of motor loadings through gain auto tuning, and then manually enter the desired bandwidth. The driver will automatically calculate appropriate Kvp and Kvi values. The motion curve is in the shape of a sine curve, as shown in Fig. 9-3.

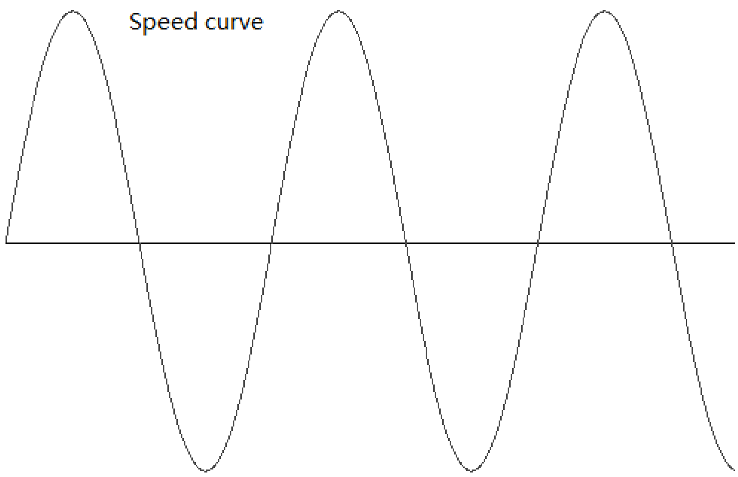


Fig.9-3 Speed curve

K_Load represents the internal data that displays the actual inertia of the system.

$$K\_Load = \frac{I_p * K_t * Encoder\_R}{62500 * \sqrt{2\pi} * J_t}$$

In the above formula:
Ip represents the maximum peak output current in units of “A”;
Kt represents the torque constant of the motor in units of “Nm/Arms”;
Encoder_R represents the resolution of a motor encoder in units of “inc/r”;
Jt represents the total inertia of the motor and loadings in units of “kg*m^2”.

Table 9-4 Parameters for controlling gain auto tuning

Numeric Display	Variable Name	Meaning	Default Value	Range
d0.06	Tuning_Start	Auto tuning starts after the variable is set to 11. All input signals are ignored during auto tuning. The variable is automatically changed to 0 after auto tuning is completed. Sets the variable to other values to end auto tuning.	0	/
d0.04	Vc_Loop_BW	Sets the bandwidth of the velocity loop in Hz. The variable can only be set after auto tuning is performed properly; otherwise the actual bandwidth goes wrong, which causes abnormal working of the driver. If the auto tuning result is abnormal, setting this parameter may also cause abnormal working of the driver. Note: This parameter cannot be applied when auto tuning is unavailable.	0	0~600
d2.17	K_Load	Indicates loading parameters	/	20~15000
d2.21	Sine_Amplitude	Proper increase in this data will reduce the tuning error, but machine vibration will become severer. This data can be adjusted	64	0~1000

		properly according to actual conditions of machines. If the data is too small, the auto tuning error becomes greater, or even causes a mistake		
d2.22	Tuning_Scale	It is helpful to reduce the auto tuning time by reducing the data, but the result may be unstable.	128	0~16384
d2.23	Tuning_Filter	Indicates filter parameters during auto-tuning	64	1~1000

Auto tuning is a process where the suitable and stable K_Load value is automatically calculated. In the auto tuning mode, the data of numeric display is automatically switched to the real-time display mode of K_Load data. When K_Load data gradually becomes stable, the driver automatically adjusts Kvp and Kvi data of a velocity loop, so that the actual bandwidth of the velocity loop is 50Hz. When K_Load data becomes stable, the driver automatically stops auto tuning operation; then you need to customize Vc_Loop_BW, representing the desired bandwidth of the velocity ring. Finally, run the test system in the actual environment, and save the parameters.
Precautions:
Auto tuning applies when both forward rotation and reverse rotation of a motor are allowable, and the loadings do not change much during the operation. When forward rotation or reverse rotation of the motor is not allowable on a device, it is recommended to adjust the parameters manually.
During auto tuning operation, pulse signals, digital input signals, and analog signals of the external controller are temporarily unavailable, so safety must be ensured.
Before auto tuning operation, it is recommended to properly adjust the Kvp, Kvi and Speed_Fb_N (a feedback filter parameter) values of the velocity loop to prevent visible oscillations when the system works in the speed mode. If necessary, adjust the data of d2.03 notch filter to inhibit resonance.
The time for different load tuning varies, and generally a few seconds is required. The auto tuning time can be reduced by presetting the K_Load value to a predicted value that is close to the actual value.
Vc_Loop_BW can be written only after successful auto tuning, otherwise the driver may work improperly. After you write the desired bandwidth of the velocity loop in Vc_Loop_BW, the driver automatically calculates the corresponding values of Kvp, Kvi and Speed_Fb_N. If you are dissatisfied with low-speed smoothness, you can manually adjust Kvi. Note that auto tuning does not automatically adjust the data of a notch filter.
In the following circumstances, auto tuning parameters should be adjusted:
When the friction in a rotation circle of the motor is uneven, it is required to increase the amplitude of d2.21 sine wave to reduce the impacts caused by uneven friction. Note that d2.21 increases when the oscillation amplitude of the loadings increase.
If auto tuning lasts for a long time, initial evaluation of the total inertia is available. It is recommended to set K_Load to an evaluation value before auto tuning.
If auto tuning is unstable, the stability of auto tuning increases when d2.22 increases properly, but the time for auto tuning slightly increases.
In the following conditions, auto adjustment goes wrong. In this case, you can only set parameters manually:
The load inertia is featured by great fluctuation.
Mechanical connection rigidity is low.
Clearances exist in the connection between mechanical elements.
The load inertia is too great, while Kvp values are set too low.
If the load inertia is too great, K_Load data will be less than 20; if the load inertia is too little, K_Load data will be greater than 15000.

9.3 Oscillation Inhibition

If resonance occurs during machine operation, you can adjust a notch filter to inhibit resonance. If resonance frequency is known, you can directly set Notch_N to (BW-100)/10. Note that you need to set Notch_On to 1 to enable the notch filter. If you do not know exactly the resonance frequency, you can firstly set the max value of d2.14 current instruction to a low one, so that the oscillation amplitude is within the acceptable range; then try to adjust Notch_N to check whether resonance disappears.
If machine resonance occurs, you can calculate the resonance frequency by observing the waveform of the

target current with the oscilloscope function of the driver.

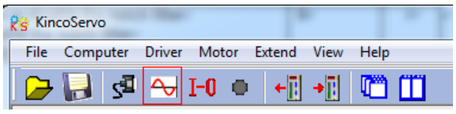
Table 9-5 Parameters for oscillation inhibition

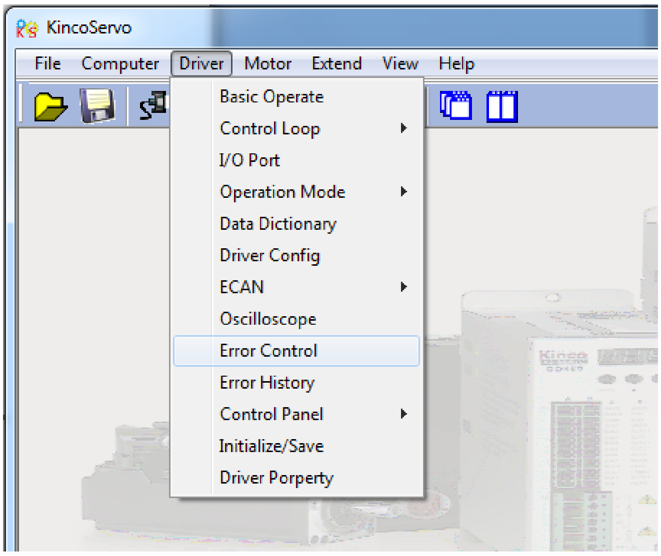
Numeric Display	Variable Name	Meaning	Default Value	Range
d2.03	Notch_N	Notch/filtering frequency setting for a velocity loop, used to set the frequency of the internal notch filter, so as to eliminate the mechanical resonance produced when the motor drives the machine. The formula is F = Notch_N*10 + 100. For example, if the mechanical resonance frequency is F = 500 Hz, the parameter should be set to 40.	45	0~90
d2.04	Notch_On	Enable or disable the notch filter 0: Disable the notch filter 1: Enable the notch filter	0	/

9.4 Debugging Example

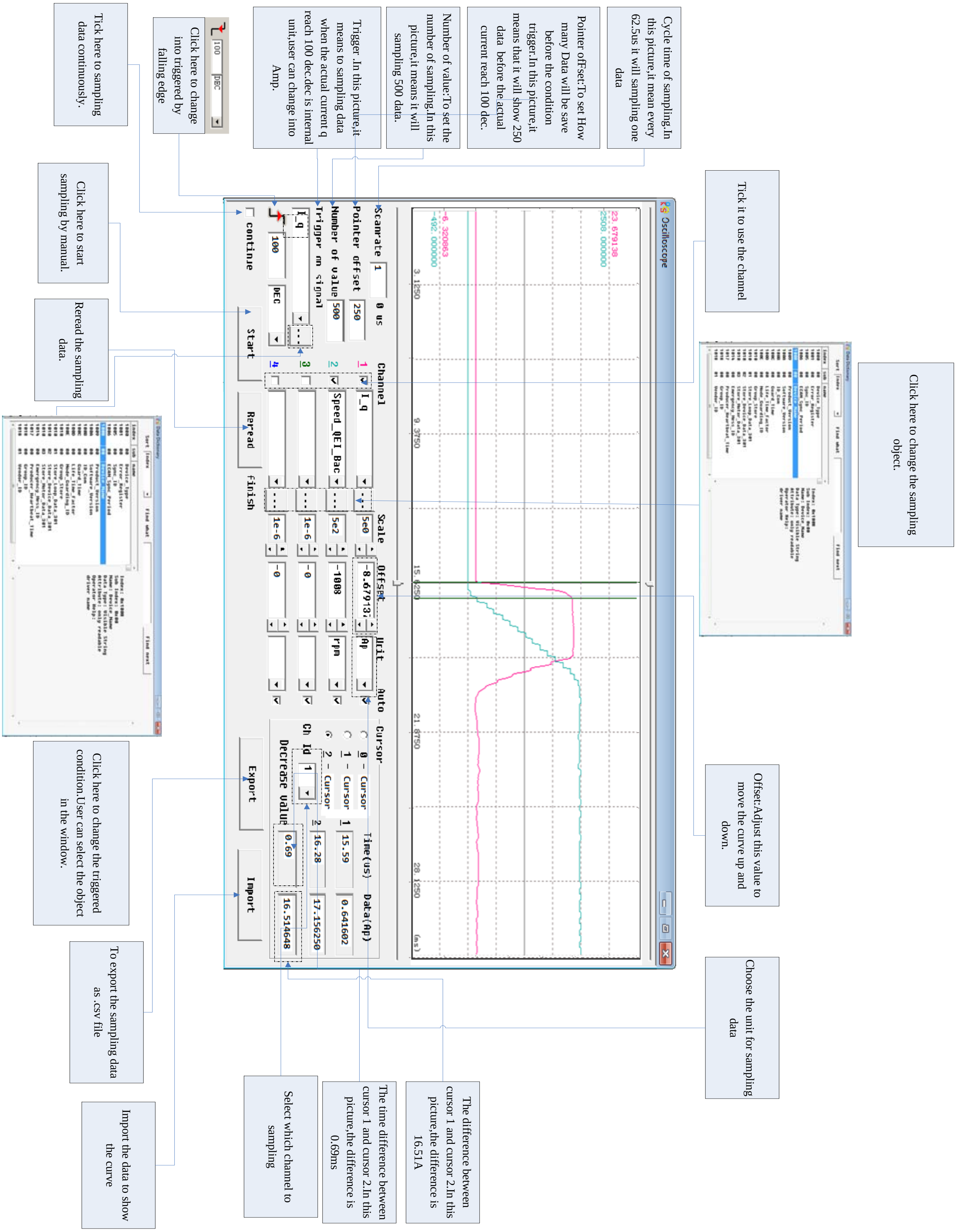
9.4.1 Oscilloscope

1.Enter oscilloscope





2.Parameters for Oscilloscope



89.4.2 Procedure for Parameter Adjustment

1、Velocity Loop Adjustment

(1) Adjust Kvp according to the load.

① Set motor running at Auto Reverse mode by position(Operation mode -3),then open oscilloscope and set the parameters to observe the curve.As shown in following figures.

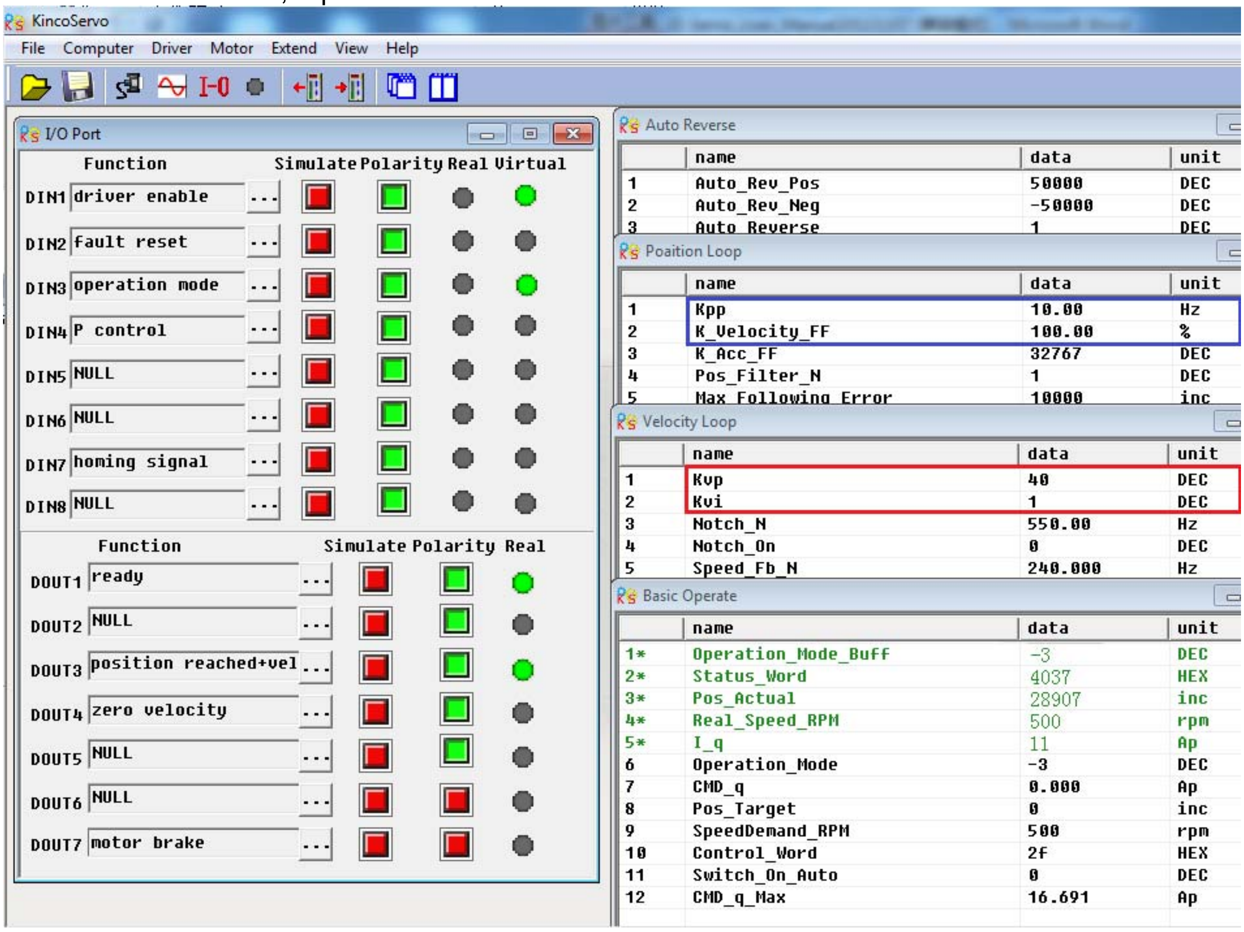
② Adjust Kvp and observe the speed curve.Following figures show the different curve in different Kvp.According to the curve,it shows that the bigger value of Kvp,the faster response of speed.

(2) Adjust Kvi according to load.

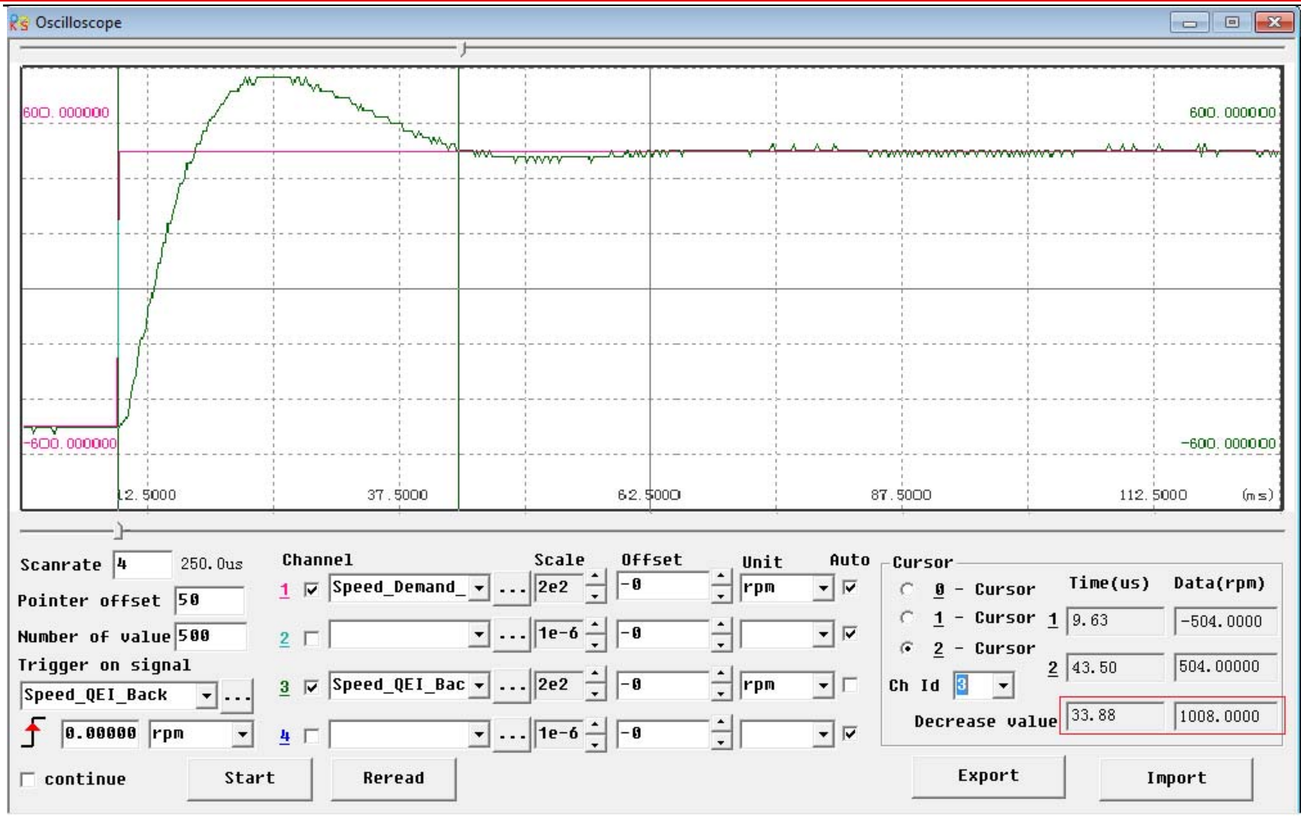
(3) Adjust Speed_Fb_N to reduce system noise.

Speed_Fb_N:This parameter is used to reduce system noise.But the bigger value of this parameter,the slower response of system.

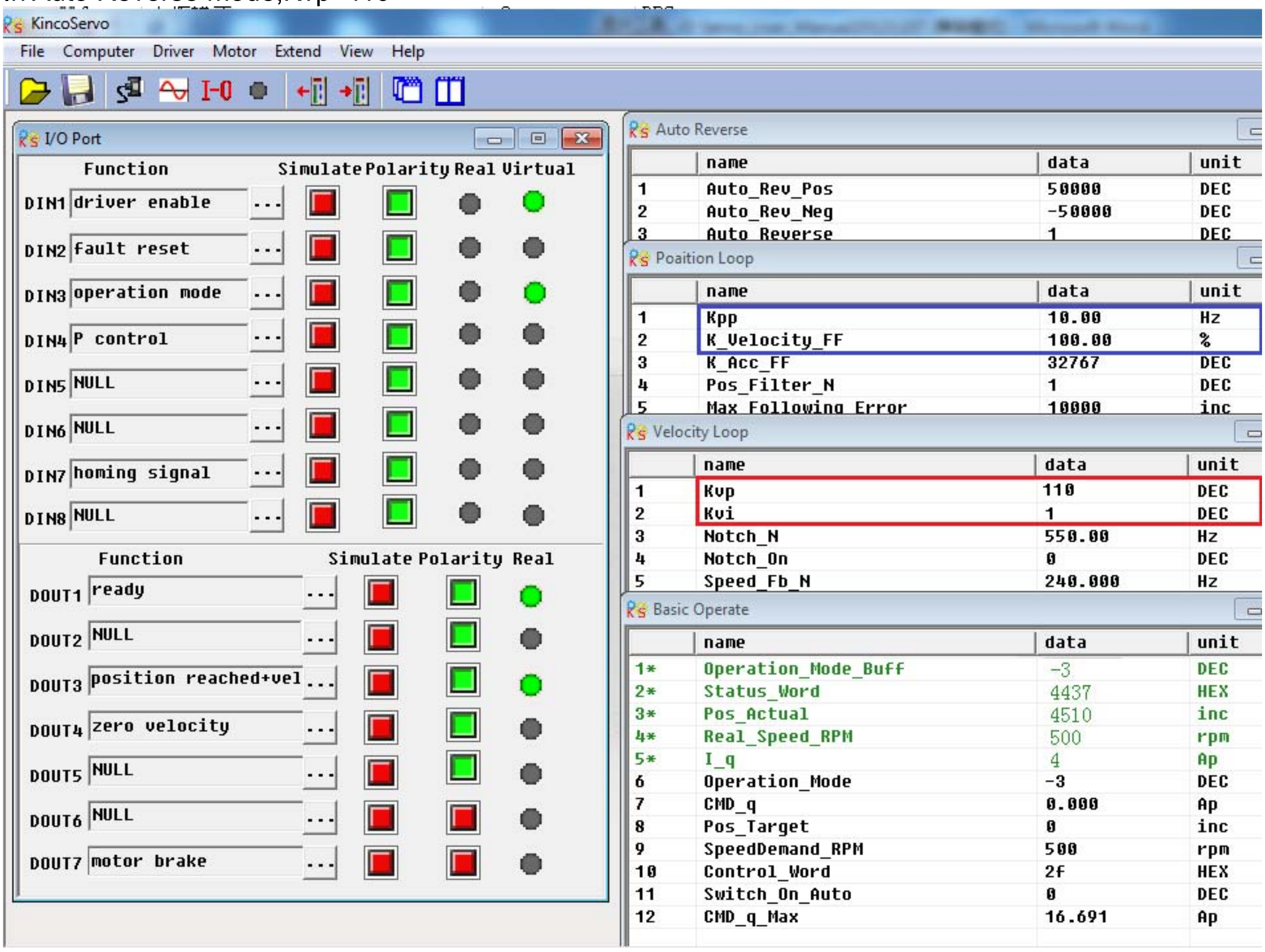
In Auto Reverse mode,Kvp=40



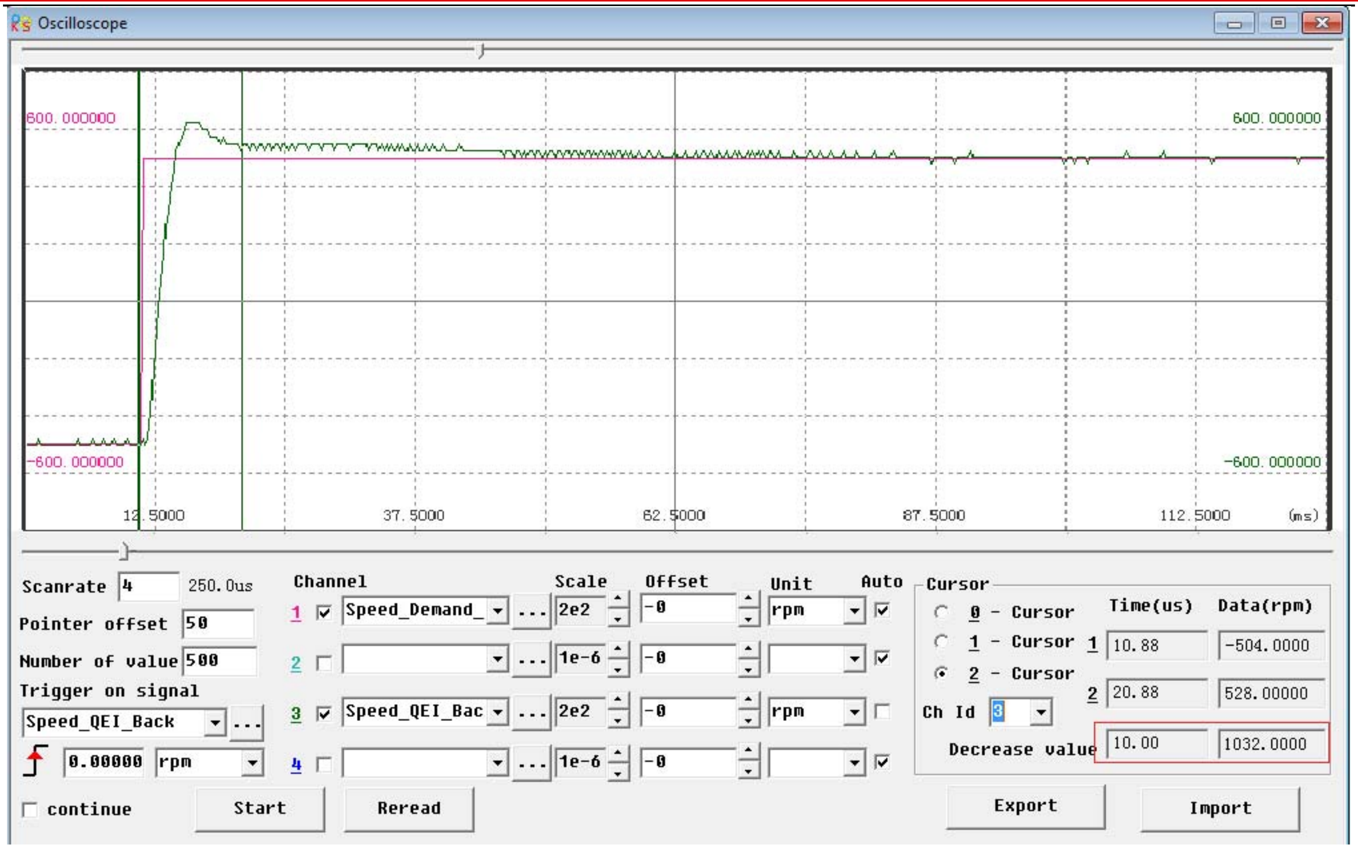
The oscilloscope is shown as follows:actual speed response is 33.88ms



In Auto Reverse mode,Kvp=110



The oscilloscope is shown as follows:actual speed response is 10.00ms



2.Position Loop Adjustment

(1) Adjust Kpp.

(2)Adjust Vff（K_Velocity_FF）

Adjust Vff parameter according to the allowable position error and coupling performance of machine. Normally Vff is 100%.If system doesn’t need high response for position,then this parameter can be decreased to reduce overshoot.

(3)Use oscilloscope to observe curve.

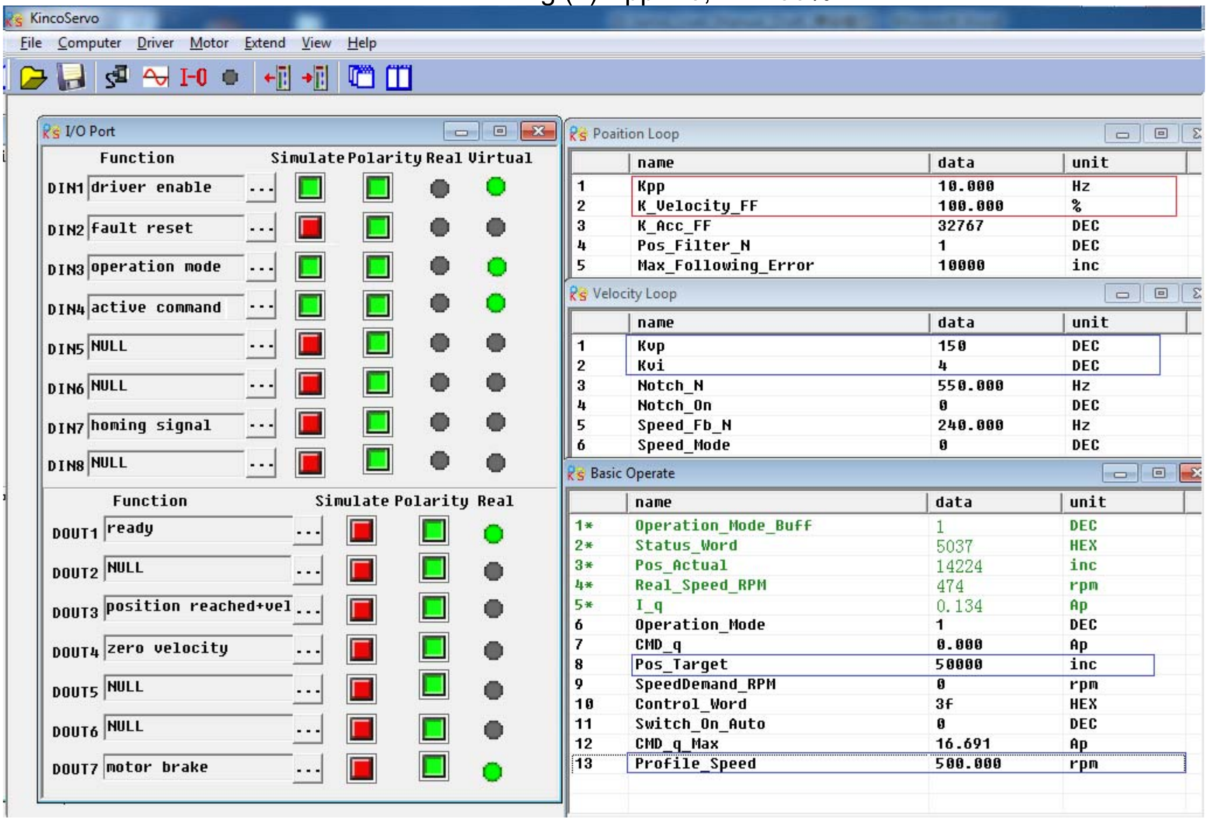
Set motor running at Auto Reverse mode by time (Operation mode 3),set parameters of oscilloscope as following figure.

In Fig.(1) and Fig.(2),Vff is 100%,When Kpp is 30,the response of position loop is faster than the one when Kpp is 10.Meanwhile the following error is also less,but overshoot is bigger.

Fig.(3),Kpp is 30,Vff is 50%.Compare with Fig.(2),the following error is bigger,but response becomes slower and there is almost no overshoot.

Internal position mode,target position is 50000 inc.

Fig.(1) Kpp=10,Vff=100%



The oscilloscope is as following: max. following error is 69 inc.

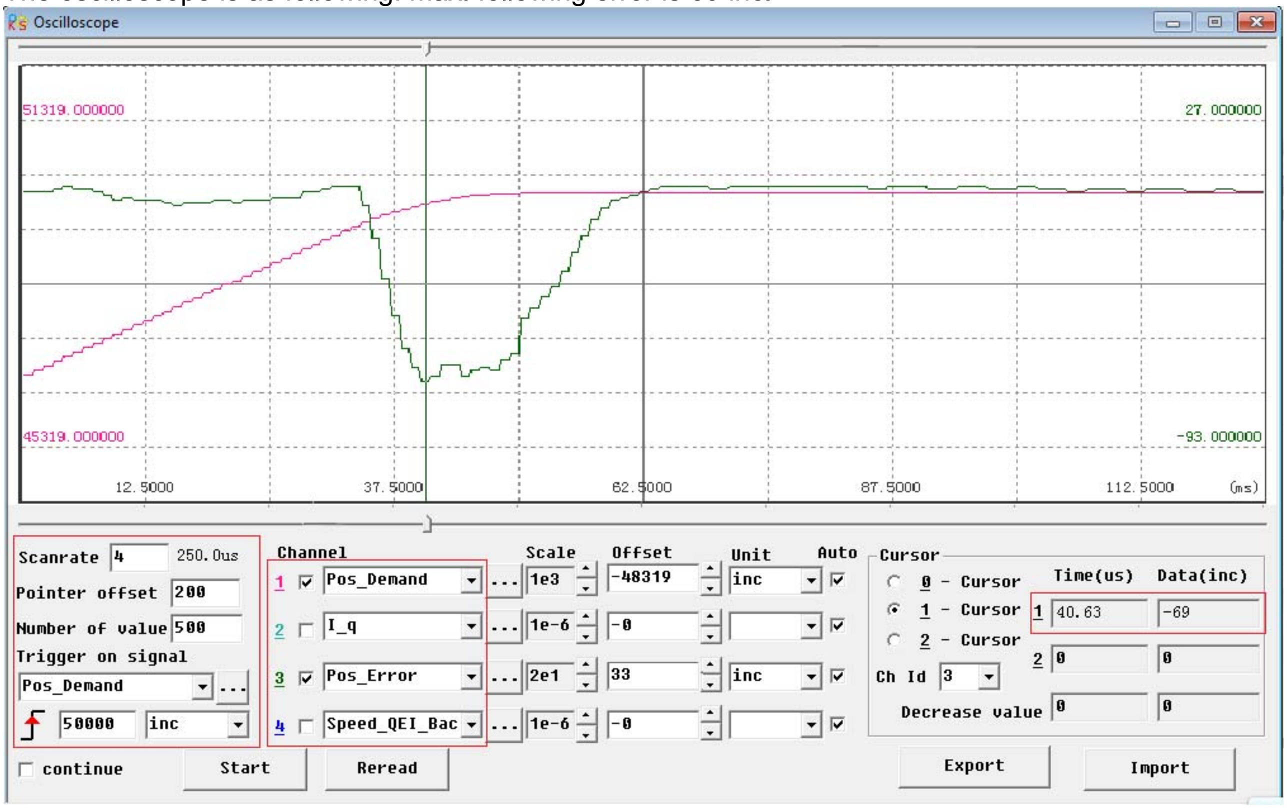
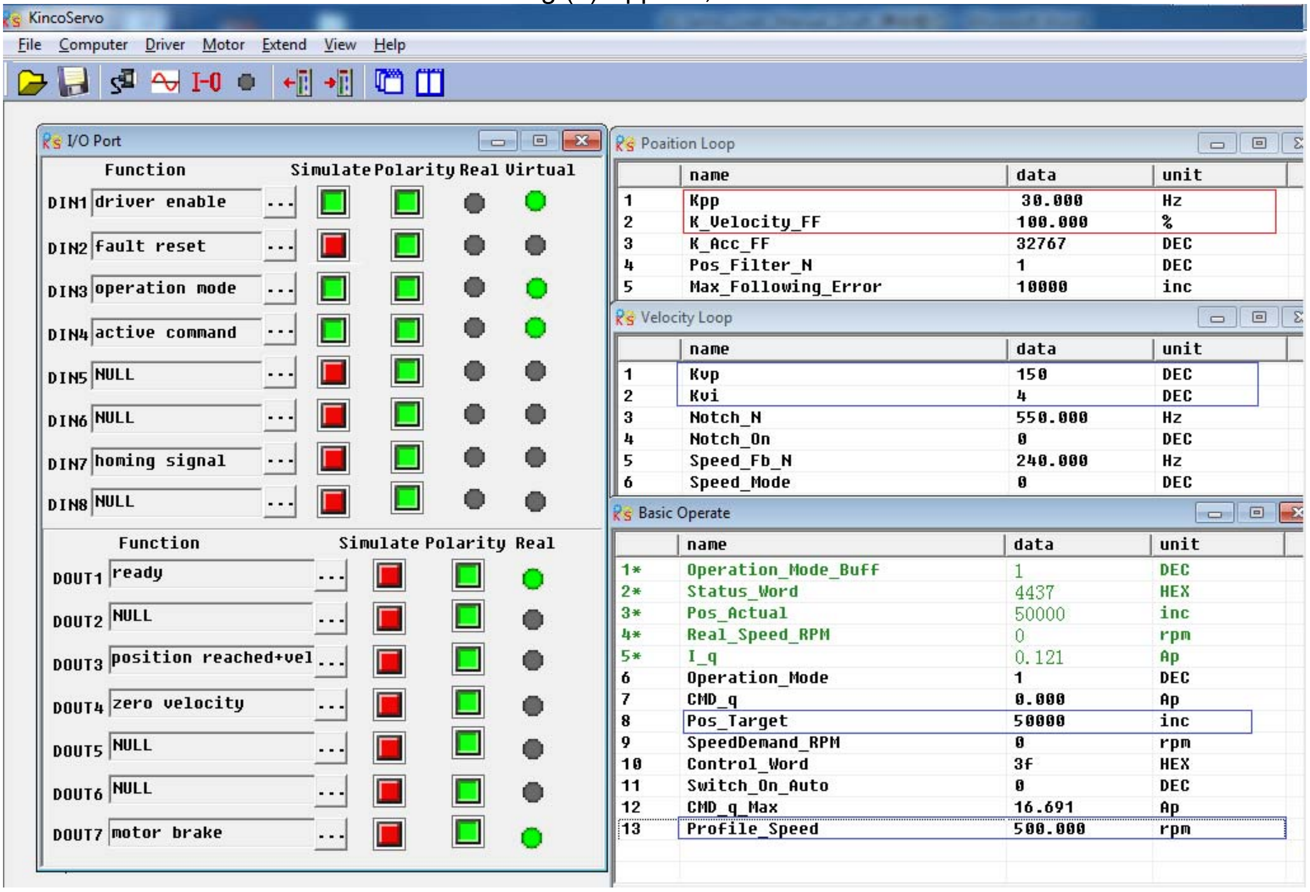


Fig.(2) Kpp=30,Vff=100%



The oscilloscope is as following:max. following error is 53 inc.

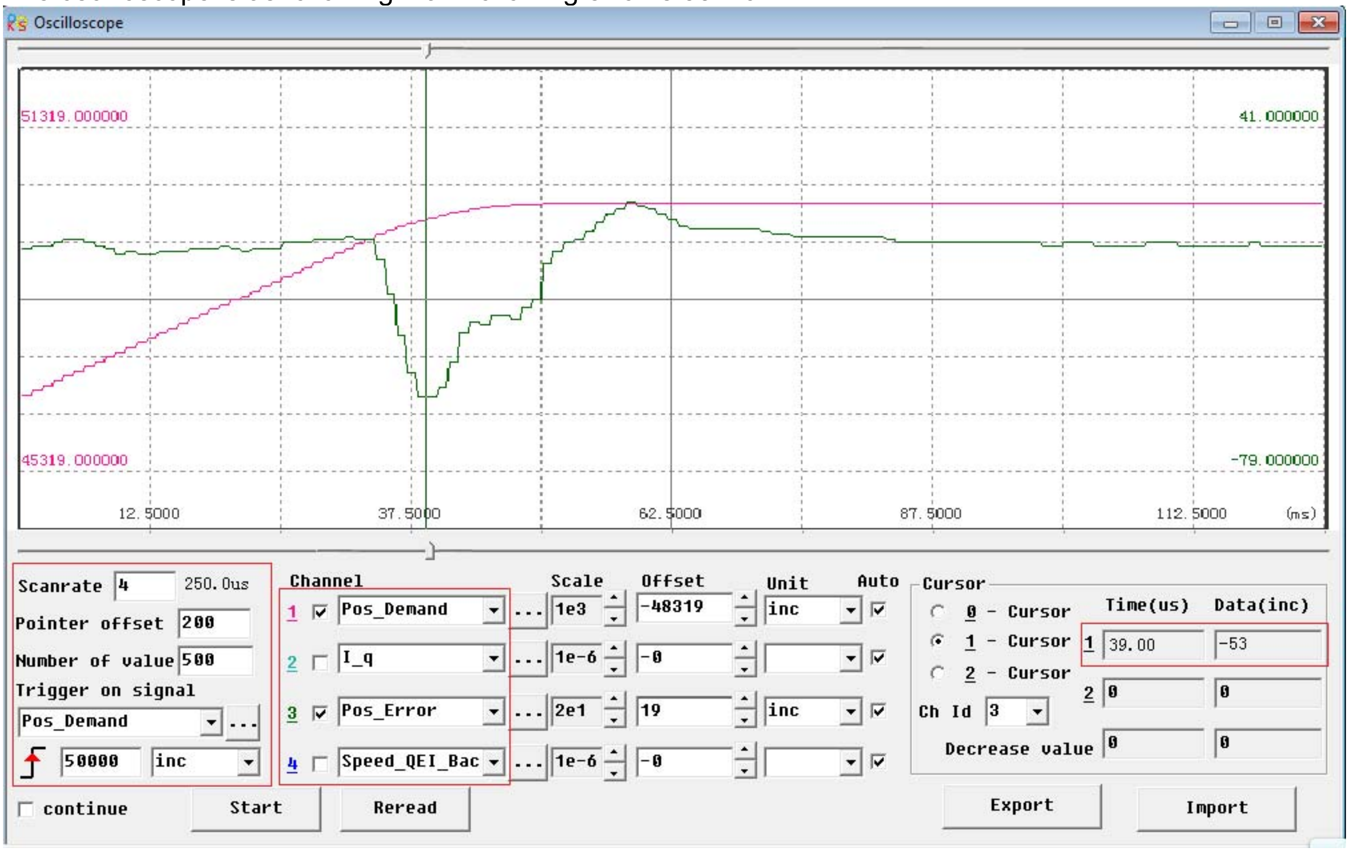
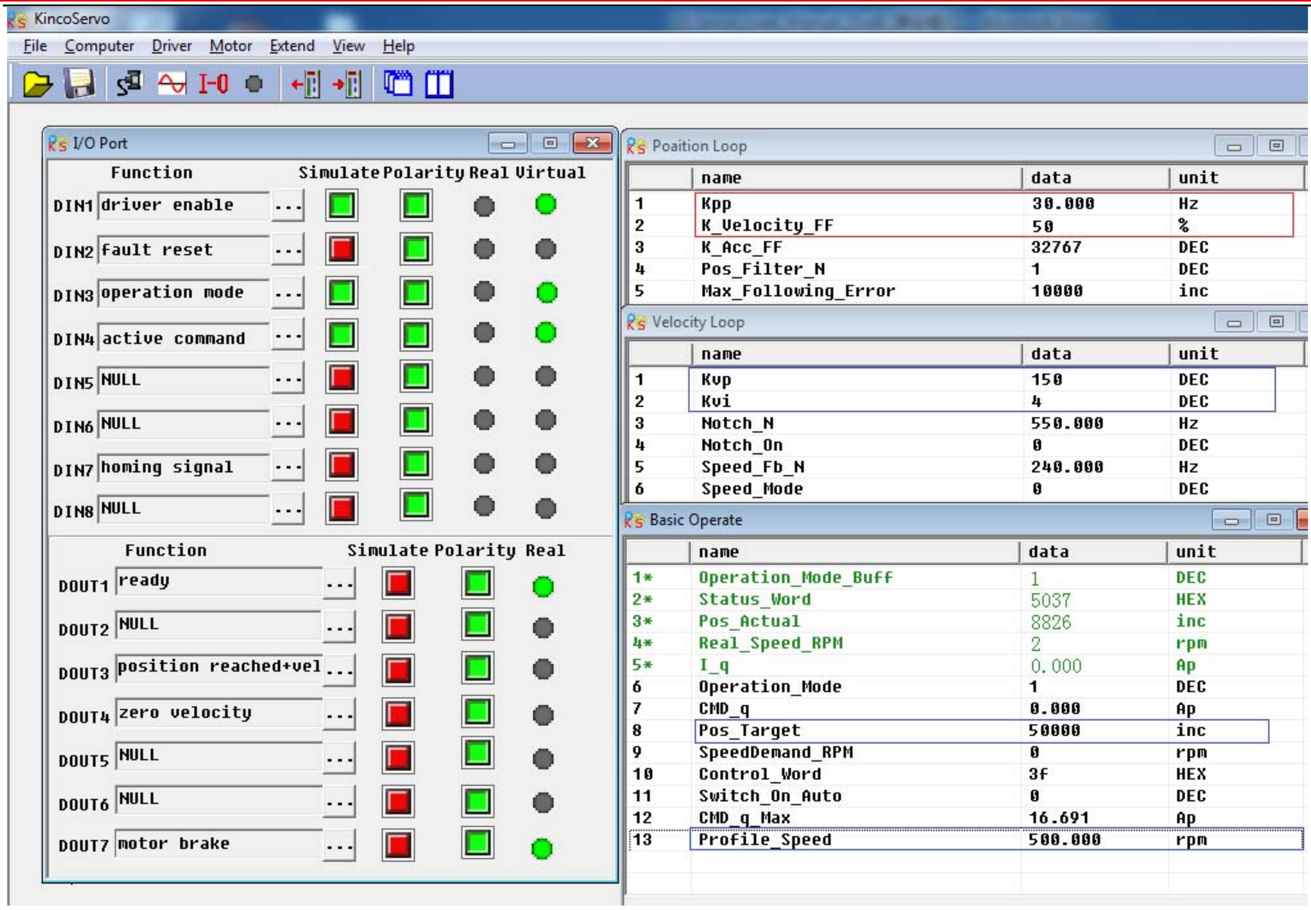
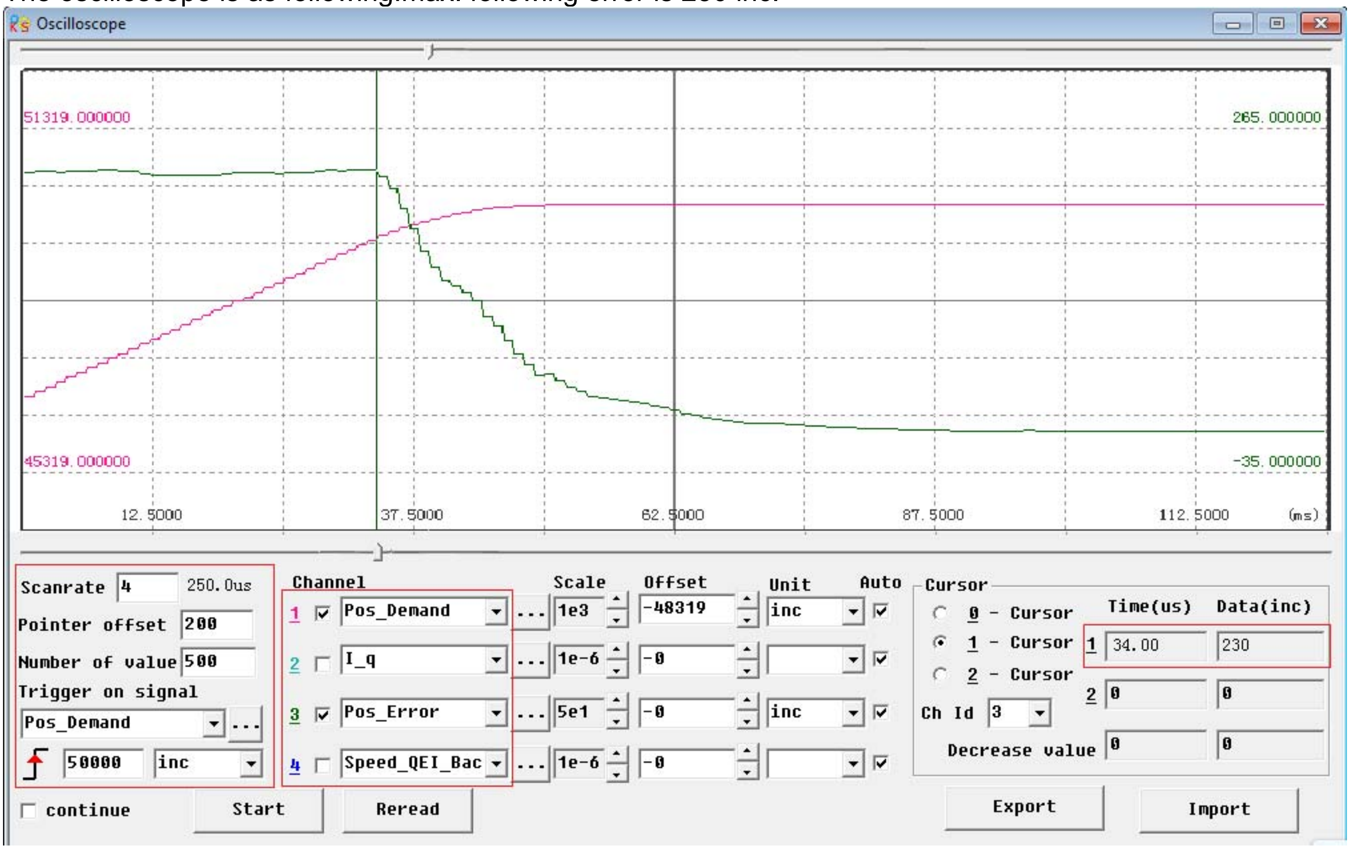


Fig.(3) Kpp=30,Vff=50%



The oscilloscope is as following:max. following error is 230 inc.



9.4.3 Easy Use Function

Easy Use aim to help users set parameters of control loop quickly, and the adjusted performance can satisfy the need of most of applications. There is also a new area for users to set the important and frequently-used parameters.

Steps of Easy Use

1. There are some frequently-used parameters in the menu of EASY, please set and confirm one by one.
   1.1 If motor type (EA01) hasn't been changed, please change EA00 to 1 to save all parameters;
   1.2 If motor type (EA01) has been changed, please change EA00 to 2 to save all parameters **and reboot servo.**
   1.3 After completing process of EASY, please run the servo. If the performance is satisfying, it is unnecessary to execute the process of TunE. Otherwise, please execute the process of TunE.
2. Parameters will become effective immediately after inputted in TunE but parameters can only be saved by Tn00.
   2.1 Please write 1 into Tn03 to start inertia measuring and then servo will adjust parameters of control loop automatically according to measured result.
   2.2 Please run the servo. If the performance is still unsatisfying, please change the stiffness by Tn01.

Notice

1. Inertia measurement might cause shaking of machine, please shut off the power or driver immediately.
2. It is strongly recommended that execute the flow of TunE after the flow of EASY and adjust the stiffness.
3. EASY and TunE menu are designed to solve the setting of servo by button originally. If users initialize parameters by PC software, EASY and TunE will only show EA00, EA01 and Tn00 for safety. Users have to confirm motor type by EA01.After that, the parameters become default and the LED will display in a complete way.

Reason for the failure of tuning

1. Wrong wire connection;
2. Wrong motor type configuration;
3. Much too low of Stiffness;
4. Mechanical gap existing;
5. Accelerated or decelerated torque is smaller than fiction torque.

Talbe-1 Motor and Servo configuration

PC	LED	Motor Model	Suitable Servo					
LED CODE:EA01			CD412S FD412S	CD422S FD422S	With Fan CD422S-AF FD422S-AF(CF、LF)	CD432S FD432S	CD612S FD612S	CD622S FD622S
K@	404.b	Without motor configuration	LED displays FFF.F					
W0	305.7	SMC60S-0020-30E■K-3LKH		√				
W1	315.7	SMC60S-0040-30E■K-3LKH		√				
W2	325.7	SMC80S-0075-30E■K-3LKH		√				
WB	425.7	SMC130D-0100-20E■K-4LKP			√			
WC	435.7	SMC130D-0150-20E■K-4HKP					√	
WD	445.7	SMC130D-0200-20E■K-4HKP					√	
WO	4F5.7	SMC130D-0150-20E■K-4LKP				√		
WP	505.7	SMC130D-0200-20E■K-4LKP				√		
WQ	515.7	SMC130D-0300-30E■K-4HKP						√
WR	525.7	SMC130D-0300-20E■K-4HKP						√
Y0	305.9	SMS60S-0020-30J■K-3LKU		√				
Y1	315.9	SMS60S-0040-30J■K-3LKU		√				
Y2	325.9	SMS80S-0075-30J■K-3LKU		√				
Z0	305.A	SMS60S-0020-30K■K-3LKU		√				
Z1	315.A	SMS60S-0040-30K■K-3LKU		√				
Z2	325.A	SMS80S-0075-30K■K-3LKU		√				
KZ	5A4.b	SMH40S-0005-30A■K-4LKH	√					
KY	594.b	SMH40S-0010-30A■K-4LKH	√					
K0	304.b	SMH60S-0020-30A■K-3LK□		√				
K1	314.b	SMH60S-0040-30A■K-3LK□		√				
K2	324.b	SMH80S-0075-30A■K-3LK□		√				
K3	334.b	SMH80S-0100-30A■K-3LK□				√		
K4	344.b	SMH110D-0105-20A■K-4LK□				√		
K5	354.b	SMH110D-0125-30A■K-4LK□				√		
K6	364.b	SMH110D-0126-20A■K-4LK□				√		
K7	374.b	SMH110D-0126-30A■K-4HK□						√
K8	384.b	SMH110D-0157-30A■K-4HK□						√
K9	394.b	SMH110D-0188-30A■K-4HK□						√
KB	424.b	SMH130D-0105-20A■K-4HK□				√		√
KC	434.b	SMH130D-0157-20A■K-4HK□				√		√
KD	444.b	SMH130D-0210-20A■K-4HK□						√
KE	454.b	SMH150D-0230-20A■K-4HK□						√
F4	344.6	85S-0025-05AAK-FLFN-02		√				
F6	364.6	85S-0035-05AAK-FLFN-02		√				
F8	384.6	85S-0045-05AAK-FLFN-02		√				

Table-2 EASY Parameters instruction

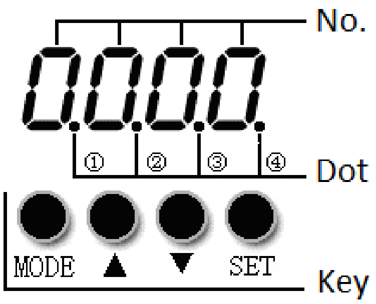
LED Display	Parameters	Description	Default
EA01	Motor Model	**Refer as Talbe-1, users should save and reboot servo after changing**	**404b**
EA02	Command Type	Modify the first LED on the right to change the command type; meanwhile the operation mode and definition of IO will be changed. 0：CW/CCW 1：P/D 2：A/B phase control 3：CW/CCW by RS422 4：P/D by RS422 5：A/B phase control by RS422 6：Analog Speed by AN1 7：Analog Speed by AN2 8：Communication Notice: It is invalid when users set 3,4,or 5 into EA01 in FD2S and CD2S When command type is 0-5, the control mode is -4. When command type is 6-7, the control mode is -3. When command type is 8, it means the servo is FD2S and DIN1, DIN2, DIN3 will be shielded	1
EA03	Gear Factor numerator	Valid when EA02 is set to 0-5. Default display is in decimal. If the number is bigger than 10000, the display is in hexadecimal. Notice: please see the different way of LED display between decimal and hexadecimal in Table-4.	1000
EA04	Gear Factor denominator		1000
EA05	Analog Speed Factor	Valid when EA02 is set to 6 or 7. The relationship between Analog input voltage and speed of motor is rpm/V Perhaps to be invalid if the factor is too big when the motor is equipped with a high resolution encoder.	300
EA06	1. Polar of Alarm Output 2.Application 3.Limited Switch 4. Load Type	The meaning of each LED from left to right: (1) Polar of Alarm Output. 0 represent normally closed contacts, 1 represent normally open contacts. (2) Limited Switch. 0 represent keeping the default,1 represent shielding all limited switch. (3) Application. It influences the control loop. 0 represent P2P,1 represent CNC,2 represent Master/Slave mode (4) Load Type. It influences the control loop. 0 represent nothing, 1 represent belt, 2 represent ball screw.	1001
EA00	Saving Parameters	Write "1" to save all the parameters. Write "2" to save all the parameters and reboot the servo, users **MUST** reboot the driver if changed the motor type) Write "3" to reboot the servo Write "10" to initialize the parameters Notice: After saving the parameters, the servo will set the control loop according to the load type and application	-
Tn01	Stiffness Level	Level 0-31, determine the BW of velocity loop and the position loop. The bigger the level is, the bigger the stiffness is. If this parameter is too big suddenly, the gain will change remarkably and the machine will be unstable. Notice: For safety, when setting Tn01, the data will be valid immediately, so the parameters should be set level by level.	belt:10 screw:13
Tn02	Inertia Ratio	Ratio of load inertia and motor inertia (* 0.1). Servo will calculate K_Load automatically according to inertia ratio and influence the proportion gain of velocity loop. Formula: Kvp=VC_LOOP_BW×K_Load/4096. VC_LOOP_BW represent the BW of position loop. Notice: For safety, when setting Tn02, the data will be valid immediately, so the parameters should be set level by level.	belt:3 screw:5

LED Display	Parameters	Description	Default
Tn03	Inertia measuring	1) Set 1 to enable motor and start inertia measuring. It contains the following operation: 1. shield all the control from external I/O 2. switch operation Mode to 10 3. enable the driver 4. set 0x2FF00C to 11 5. start shaking the shaft of motor and get the result 6. restore all the control of external I/O 2) After confirming, the LED will stop flashing and show the Tuning result. While 1 means success; -1,-2,-3,-4 means failure due to some reasons. If it is successful, the control loop parameters will be set automatically and the stiffness will be set to 4-13 according to inertia ratio and Tn03 will show 1. If it is failed, the stiffness will be set to 10 and the inertia ratio will be set to 30(*0.1) and Tn03 will show error code.	-
Tn04	Measuring Distance	Distance of inertia measuring(*0.01), maximum is 0.4 round	0.22
Tn00	parameters Saving	Write“1”to save all the parameters. Write“2”to save all the parameters and reboot the servo , Write “3” to reboot the servo Write “10” to initialize the parameters Notice: Users **MUST** reboot the driver if changed the motor type.	
4. Notice: EASY and TunE menu are designed to solve the setting of servo by button originally. If users initialize parameters by PC software, EASY and TunE will only show EA00, EA01 and Tn00 for safety. Users have to confirm motor type by EA01.After that, the parameters become default and the LED will display in a complete way.			

Table-3 Stiffness level

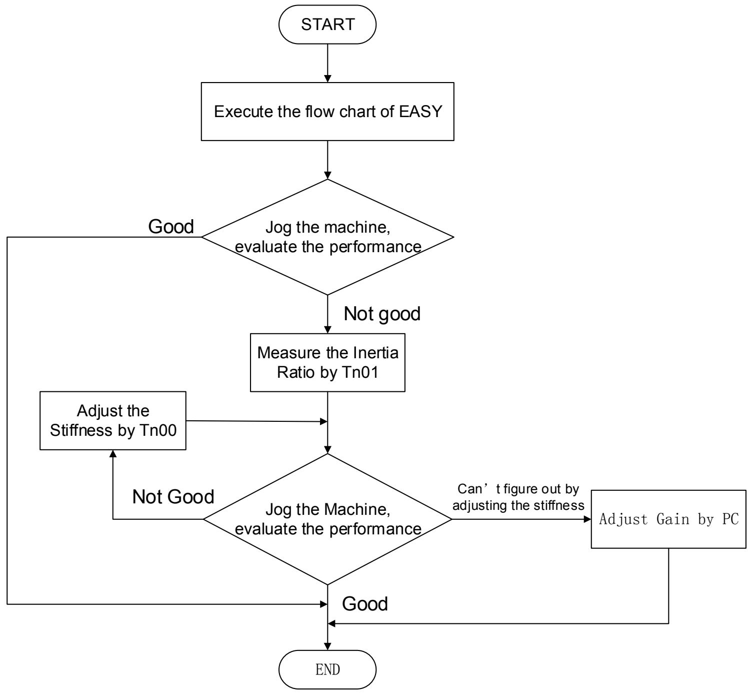
Stiffness level	Kpp/0.01Hz]	Kvp/[0.1Hz]	Speed feedback filter[Hz]	Stiffness level	Kpp/[0.01Hz]	Kvp/0[0.1Hz]	Speed feedback filter [Hz]
0	70	25	100	16	1945	700	480
1	98	35	100	17	2223	800	560
2	139	50	100	18	2500	900	620
3	195	70	100	19	2778	1000	700
4	264	95	100	20	3334	1200	800
5	334	120	100	21	3889	1400	900
6	389	140	120	22	4723	1700	1000
7	473	170	120	23	5556	2000	1000
8	556	200	140	24	6389	2300	1000
9	639	230	160	25	7500	2700	1000
10	750	270	180	26	8612	3100	1000
11	889	320	200	27	9445	3400	1000
12	1056	380	240	28	10278	3700	1000
13	1250	450	300	29	11112	4000	1000
14	1500	540	360	30	12500	4500	1000
15	1667	600	420	31	13889	5000	1000
Notice: When setting stiffness or inertia ratio, it is useless to raise stiffness any more if Kvp is more than 4000. And it will decrease band width if going on increasing the inertia ratio. If the resolution of encoder is less than 80000 PPR, the range of stiffness is from 0 to 22.							

Table-4 Operation of Panel

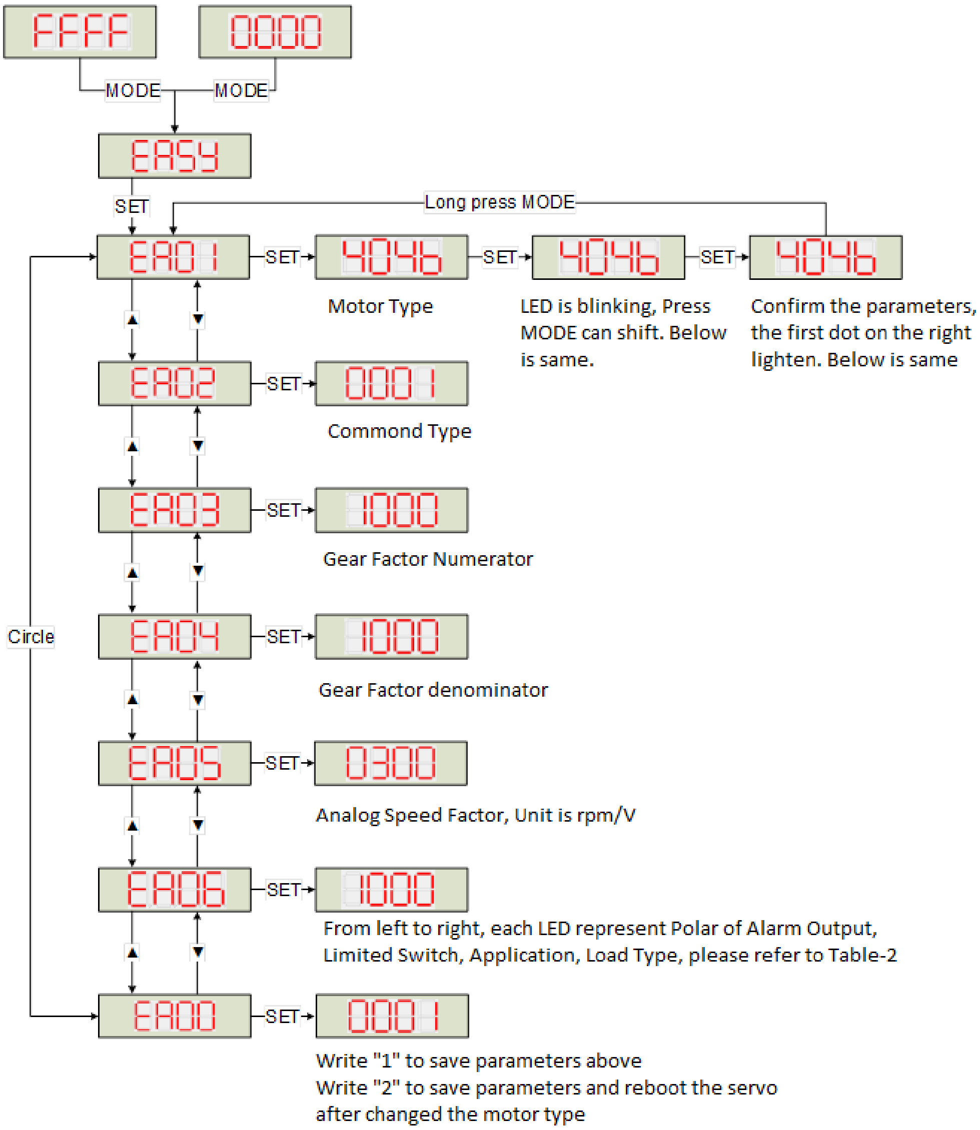


Description	
MODE	Switch menus； When setting parameters, press can shift, long press can return to the previous menus.
▲	Press▲ can increase the number, long press can increase quickly
▼	Press▼ can decrease the number, long press can decrease quickly
③	Shining represent displaying in hexadecimal, otherwise in decimal.
SET	Enter the selected menu； Enter the status of parameters setting； affirm the parameters；
Display FFF.F	Without motor configuration, please operate according to the flow chart of “Easy” and make it sure to save the parameters and reboot the servo.

Flow Chart of Adjusting Gain

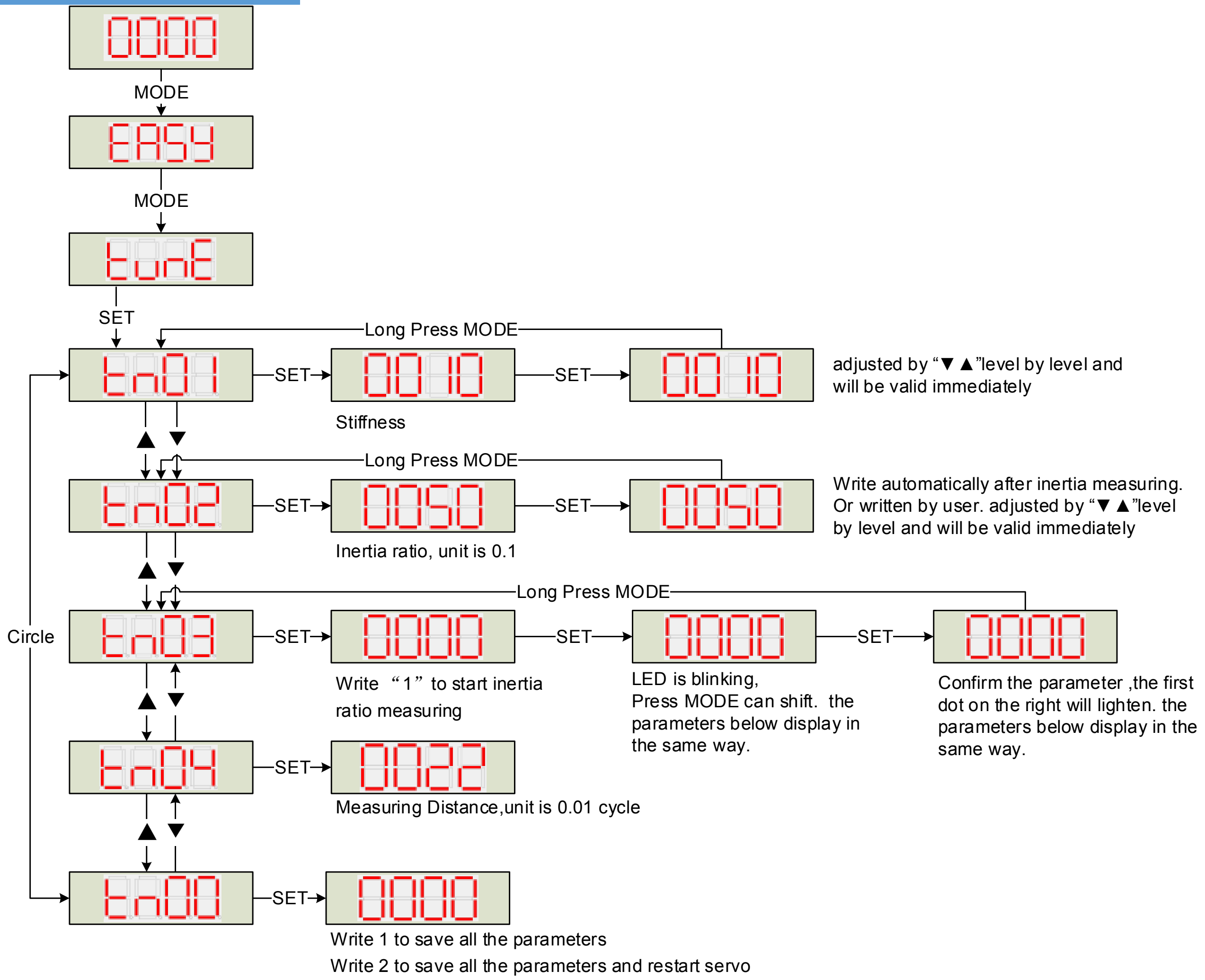


Flow Chart of EASY



Notice: Must execute in order. Exit automatically if there is no operation in 60s and users have to start again. The data input will be valid immediately, but need to be saved by EA00

Flow Chart of TunE



Notice: The data will be valid immediately, but need to be saved by Tn00.

For safety, when setting Tn01 or Tn02, the data will be valid immediately and these two parameters should be set level by level.

Chapter 10 Communication

FD2S Servo 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 methods including RS232, RS485 and CANopen. It supports the connection of multiple sites and simultaneous operation of multiple communication ports.

Notice:

1.DIN1 is set as driver enable function and DIN3 is set as operation mode control function by default.Before using communication control,it must cancel the functions of these two DIN.

2.There are internal unit and engineering unit.All the parameters use internal unit when using communication control,so it need to convert the unit.About more details about the relationship of the units please refer to Appendix.

3.When using read/write function of SDO of CANopen,RS232 and RS485 communication,make sure there is only one command in the network at the same time,and good communication error handling, etc., in order to avoid communication into an infinite loop.

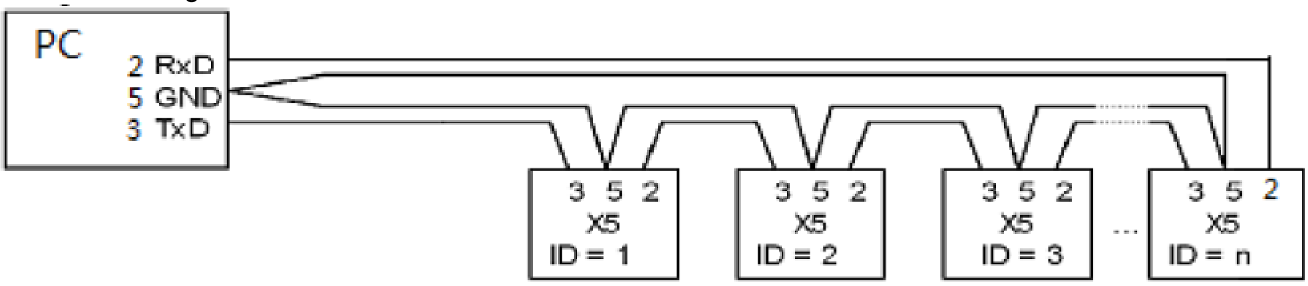
10.1 RS232 Communication

10.1.1 RS232 Communication Interface

The wiring diagram between PC and single FD2S Servo is as following:

PC		FD2S Servo RS232(X3)
2 RxD	----------------------------------	TXD 2
3 TxD	----------------------------------	RXD 3
5 GND	---------------------------------	GND 5

The wiring diagram between PC and multiple FD2S Servo is as following：（D05.15 must be set as 1,and restart
driver after setting）



FD2S SERVO

Note:1.It is the same way to connect FD2S Servo to HMI or other controllers.(The PIN definition of HMI or other controllers may be different from PC's).

2.When using the wiring of multiple FD2S Servo,all the FD2S Servo will receive the command at the same time.

10.1.2 RS232 Communication Parameters

LED Display	Internal Address	Name	Meaning	Default value
d5.00	2FF00108	Store_Loop_Data	1：Store all control parameters except motor parameters 10：Initialzie all control parameters except motor parameters	0
d5.01	100B0008	ID_Com	Station No. of Drivers Note: To change this parameter, you need to save it with the address “d5.00”, and restart it later.	1
d5.02	2FE00010	RS232_Bandrate	Set the baud rate of RS232 port 540 19200 270 38400 90 115200 Note: To change this parameter, you need to save it with the address “d5.00”, and restarts it later.	270
d5.15	65100B08	RS232_Loop_Enable	0：1:1 1：1:N Note:It needs to restart driver after changing this parameter.	0
Other parameters			Data bit = 8 Stop bit = 1 Parity = None	Constant

10.1.3 Transport Protocol

The RS-232C communication of the FD2S Servo driver strictly follows a master/slave protocol. The host computer can send any data to FD2S driver. The driver configured with ID No. will calculate such data and return a reply.

This transport protocol of RS232 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 lowest byte of the calculation result.

The host sends:

byte 0 byte 9

ID	8 byte host data	CHKS

When D5.15 is 0,FD Servo sends:

byte 0 byte 9

ID	8 byte slave data	CHKS

When D5.15 is 1,FD Servo sends:

byte 0		byte 9	byte 0		byte 9
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 an ID not existed in the network to the FD2S Servo driver, no FD2S Servo driver 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.

10.1.3.1 Data Protocol

A data protocol is different from a transport protocol. It contains 8 bytes of all 10 bytes of the above RS-232. Definition of CD servo driver internal data complies with the CANopen international standard. All parameters, values and functions are expressed by index and subindex.
A:Download. 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:

byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7
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 7650 inc into "Target Position" in the slave, the unit of 607A0029 is inc, 7650 is in decimal system, and 1DE2 is in hexadecimal system.Since the length of the object to be written is 4 bytes and the calculation result 1D E2 has only 2 bytes,zero shall be filled to the higher-order bits. Therefore, the final result = 00 00 1D E2.
DATA： byte4=E2
byte5=1D
byte6=00
byte7=00
Slave responds：

byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7
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 example:
Host sends:
01 23 7A 60 00 E2 1D 00 00 03 （This command is to write data into target position 607A0020）
Slave responds:

01 60 7A 60 00 E2 1D 00 00 C6
Means:
01—Station No. of slave is 1
60—Data successfully sent.And data are saved in byte4…byte5.
byte4=E2，byte5=1D，byte6=00，byte7=00
Then,DATA= byte7 byte6 byte5 byte4 = 1DE2（hex）=7650 inc

B:Upload. 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 host sends:

byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7
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 responds:

byte0	byte1	byte2	byte3	byte4	byte5	byte6	byte7
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 maste
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:
Host sends:
01 40 7A 60 00 00 00 00 00 E5 （This command is to read data of target position 607A0020）
Slave responds
01 43 7A 60 00 E2 1D 00 00 E3
Means:
01—Station No. of slave is 1
43—Receive 4 bytes of data and save into byte4…byte5.
byte4=E2，byte5=1D，byte6=00，byte7=00
Then DATA= byte7 byte6 byte5 byte4 = 1DE2（hex）=7650 inc

10.1.4 RS232 Communication Address of Servo Parameters

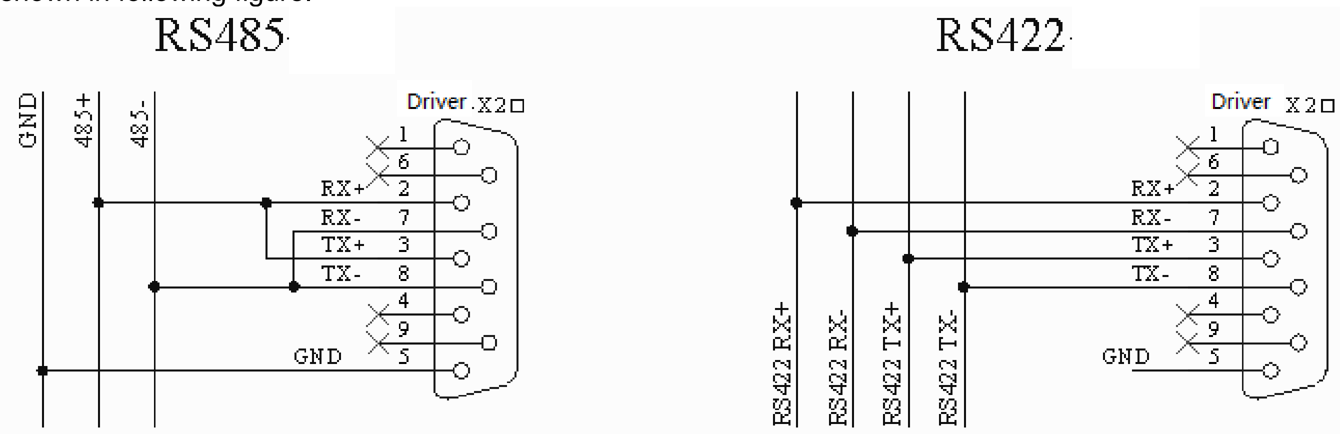
About the objects of each operation mode please refer to chapter8.

About common object address please refer to object list in Appendix.
About all the communication address please refer to parameters list.
About RS232 communication example please refer to Appendix.

10.2 RS485 Communication

10.2.1 RS485 Communication Interface

The X2 interface of FD2S Servo driver supports RS485 and RS422 communication.The wiring diagram is shown in following figure.



10.2.2 RS485 Communication Parameters

LED Display	Name	Meaning	Default Value
d5.01	ID_Com	Station No. of Drivers Note: To change this parameter, you need to save it with the address "d5.00", and restart it later.	1
	RS485_Bandrate	Set the baud rate of RS485 port Note: This parameter must be changed in KincoServo software.	540
Other parameters		Data bit = 8 Stop bit = 1 Parity = None	Constant

10.2.3 MODBUS RTU

The RS485 interface of FD2S Servo driver supports Modbus RTU protocol.
Modbus RTU protocol format

Start(No less than 3.5 characters of messages interval)	Station No.	Function code	Data	CRC
	1 Byte	1 Byte	N Bytes	2 Bytes

Function code of Modbus

0x03：Read data registers

Request format：

Station No.	Function Code	High Byte of Start Address	Low Byte of Start Address	High byte of Address Length (Word)	Low byte of Address Length (Word)	CRC check
1 Byte	03	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes

Normal response format:

Station No.	Function Code	Return data length(Bytes)	High byte of Register 1	Low byte of Register 1	…	CRC check
1 Byte	03	1 Byte	1 Byte	1 Byte	…	2 Bytes

If there is error such as non-exist address,then it will return function code 0x81.

For example:Send message 01 03 32 00 00 02 CA B3

Meaning:

01： Station NO.

03： Function code:read data registers

32 00：Read address starting from 4x3200(Hex).This is the modbus address corresponding to parameter“Status word”(60410010)

00 02：Read 2 words of data

CA B3：CRC check.

0x06：Write single data register

Request format:

Station No.	Function Code	High Byte of Register	Low Byte of Register	High byte of writing value	Low byte of writing value	CRC check
1 Byte	06	1 Byte	1 Byte	1 Byte	1 Byte	1 Bytes

Response format:If writing successful,then return the same message.

If there is error such as address over range,non-exist address and the address is read only,then it will return function code 0x86.

For example:Send message 01 06 31 00 00 0F C7 32

Meaning:

01： Station No.

06： Function code,write single WORD

31 00： Modbus address for writing data.This is the address corresponding to parameter “control word”(60400010)

00 0F： Write data 000F(Hex)

C7 32： CRC check.

0x10：Write multiple registers

Request format:

Station No.	Function Code	High Byte of Start Address	Low Byte of Start Address	High byte of Address Length (Word)	Low byte of Address Length (Word)	Data length (Bytes)	High byte of Data 1	Low byte of Data 1	…	CRC check
1 Byte	10	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	1 Byte	…	2 Bytes

Normal respons format:

Station No.	Function Code	High Byte of Start Address	Low Byte of Start Address	High byte of Address Length (Word)	Low byte of Address Length (Word)	CRC check
1 Byte	10	1 Byte	1 Byte	1 Byte	1 Byte	2 Bytes

If there is error such as address over range,non-exist address and the address is read only,then it will return function code 0x90

For example:Send message 01 10 6F 00 00 02 04 55 55 00 08 1A 47
Meaning:
01：   Station No.
10：   Function code,write multiple WORDs
6F 00：   Modbus address for writing data. This is the address corresponding to parameter “Target Velocity”(60FF0020)
00 02:   Address length is 2 WORD.
04：   Data length is 4 Bytes(2 words)
55 55 00 08：Write data 00085555(Hex) into address.
1A 47：   CRC check

10.2.4 RS485 Communication Address of Servo Parameters

About the objects of each operation mode please refer to Appendix.
About common object address please refer to object list in Appendix.(Not all the objects support RS485)
About RS485 communication example please refer to Appendix.

10.3 CANopen Communication

CANopen is one of the most famous and successful open fieldbus standards.It has been widely recognized and applied a lot in Europe and USA. In 1992,CiA (CANinAutomation) was set up in Germany,and began to develop application layer protocol CANopen for CAN in automation. Since then, members of CiA developed a series of CANopen products,and applied in a large number of applications in the field of machinery manufacturing such as railway, vehicles, ships, pharmaceutical, food processing etc..Nowadays CANopen protocol has been the most important industrial fieldbus standard EN-50325-4 in Europe
The FD2S series servo supports standard CAN (slave device), strictly follow CANopen2.0A / B protocol, any host computer which support this protocol can communicate with it. FD2S Servo uses of a strictly defined object list, we call it the object dictionary, this object dictionary design is based on the CANopen international standards, all objects have a clear definition of the function. Objects said here similar to the memory address, we often say that some objects, such as speed and position,can be modified by an external controller, some object were modified only by the drive itself, such as status and error messages.
These objects are as following:

For example:

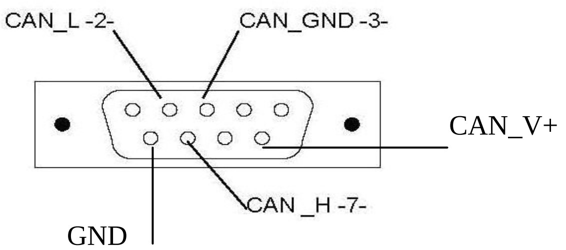
Index	Sub	Bits	Attribute	Meaning
6040	00	16(=0x10)	RW	Control word
6060	00	8(=0x08)	RW	Operation mode
607A	00	32(=0x20)	W	Target position
6041	00	16(=0x10)	MW	Status word

The attributes of objects are as follows:
1. RW:The object can be both read and written.
2. RO:The object can be read only
3. WO:The object can be written only.
4. M:The object can be mapping,similar to indirect addressing.
5. S:The object can be stored in Flash-ROM without lost after power failure.

10.3.1 Hardware Introduction

CAN communication protocol describes a way of transmitting information between devices, The definition of CAN layer is the same as the open systems interconnection model OSI, each layer communicates with the same layer in another device, the actual communication takes place adjacent layers in each device,but the devices only interconnect by the physical media of thephysical layer in the model.CAN standard defines data link layer and physical layer in the mode. The physical layer of CAN bus is not strictly required, it can use a variety of physical media such as twisted pair Fibre. The most commonly used is twisted pair signal, sent by differential voltage transmission (commonly used bus transceiver). The two signal lines are called CAN_H and CAN_L. The static voltage is approximately 2.5V, then the state is expressed as a logical 1, also called hidden bit. It represents a logic 0 when CAN_H is higher than the CAN_L, we called it apparent bit,then the voltage is that CAN_H = 3.5V and CAN_L= 1.5V,apparent bit is in high priority.

The standard CAN interface is as following figure:



Pin	Name	Description
1	NC	Reserved
2	CAN_L	CAN_L bus (low dominant)
3	CAN_GND	CAN ground
4	NC	Reserved
5	CAN_SHLD	Optional shield for CAN
6	GND	Optional ground
7	CAN_H	CAN_H bus（high dominant)
8	NC	Reserved
9	CAN_V+	NC

■Note:
1、All CAN_L and CAN_H of slaves connect directly by using series connection, not star connection.
2、There must be connected a 120 ohm resistance in start terminal(master) and end terminal(slave).
3、All FD2S Servo driver don't need external 24VDC supply for CAN interface.
4、Please use the shield wires for communication cable,and make good grounding(Pin.3 is advised to grounding when
communication is in long distance and high baudrate）.
5、The max. distance at different baudrate are shown in following table:

Baudrate	Distance
1Mbit/s	25M
800Kbit/s	50M
500Kbit/s	100M
250Kbit/s	250M
125Kbit/s	500M
50Kbit/s	600M
25Kbit/s	800M
10Kbit/s	1000M

10.3.2 Software Introduction

10.3.1.1 EDS

EDS（Electronic Data Sheet）file is an identification documents or similar code of slave device,to identify what kind of slave device is(Like 401,402 and 403,or which device type of 402).This file includes all information of slaves,such as manufacturer,sequence No.,software version,supportable baudrate,mappable OD and attributes of each OD and so on,similar to the GSD file for Profibus.Therefore,we need to import the EDS file of slave into the software of master before we configure the hardware.

10.3.1.2 SDO

SDO is mainly used in the transmit the low priority object between the devices, typically used to configure and mange the device,such as modifying PID parameters in current loop,velocity loop and position loop,and PDO configuration parameters and so on.This data transmission mode is the same as Modbus,that is it needs reponse from slave when master sends data to slave.This communication mode is suitable for parameters setting,but not for data transmission frequently.

SDO includes upload and download.The host can use special SDO instructions to read and write the OD of servo.

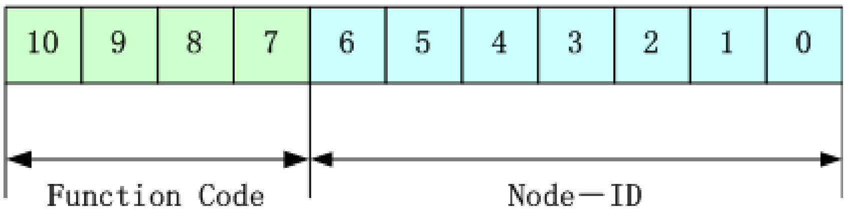
10.3.1.3 PDO

PDO can transport 8 bytes of data at one time,and no other protocol preset(Mean the content of the data are preset),it is mainly used to transmit data in high frequency.PDO uses brand new mode for data exchange,it needs to define the data receiving and sending area before the transmission between two devices,then the data will transmit to the receiving area of devices directly when exchanging data.It greatly increase the efficiency and ultilization of the bus communication.

PDO COB-ID

COB-ID is a unique way of CANopen communication protocol,it is the short name of Communication Object Identifier. These COB-ID defines the respective transmission levels for PDO, These transport level, the controller and servo will be able to be configured the same transmission level and the transmission content in the respective software.Then both sides know the contents of data to be transferred, there is no need to wait for the reply to check whether the data transmission is successful or not when transfering data.

The default ID allocation table is based on the CAN-ID(11 bits) defined in CANopen 2.0A（The COB-ID of CANopen 2.0B protocol is 27 bits）,include function code(4 bits) and Node-ID(7 bits) as shown in following figure:



Node-ID is defined by system integrators,such setting by the DIP switch on the devices(Like servo’s station No.).The range of Node-ID is 1~127(0 is forbidden).

Function Code:The function code for data transmission define the transmission level of PDO,SDO and management message.The smaller the function code,the higher the priority.

The allocation table for CAN identifiers in master/slave connection set predefined by CANopen is as follows:

Broadcast objects			
Object	Function code（ID-bits 10-7）	COB-ID	Index of communication parameter in OD
NMT Module Control	0000	000H	-
SYNC	0001	080H	1005H，1006H，1007H
TIME SSTAMP	0010	100H	1012H，1013H
Reciprocity objects.			
Object	Function code（ID-bits 10-7）	COB-ID	Index of communication parameter in OD
Emergency	0001	081H-0FFH	1024H，1015H
PDO1(Send)	0011	181H-1FFH	1800H
PDO1(Receive)	0100	201H-27FH	1400H
PDO2(Send)	0101	281H-2FFH	1801H
PDO2(Receive)	0110	301H-37FH	1401H
PDO3(Send)	0111	381H-3FFH	1802H
PDO3(Receive)	1000	401H-47FH	1402H
PDO4(Send)	1001	481H-4FFH	1803H
PDO4(Receive)	1010	501H-57FH	1403H
SDO(Send/Server)	1011	581H-5FFH	1200H
SDO(Receive/Client)	1100	601H-67FH	1200H
NMT Error Control	1110	701H-77FH	1016H-1017H

Note:

1. The smaller the COB-ID,the higher the priority.
2. The function codes of COB-ID in every level are fixed.
3. COB-ID of 00H, 80H, 100H, 701H-77FH, 081H-0FFH are system management format.

The COB-ID supported by FD2S Servo:

Send PDO（TXPDO）

Send PDO of servo means servo sends out data,and these data are received by PLC.The function codes of send PDO (COB-ID) are as follows:

1、 0x180+Station No. of Servo

2、 0x280+ Station No. of Servo

3、 0x380+ Station No. of Servo
4、 0x480+ Station No. of Servo

Receive PDO（RXPDO）

Receive PDO of servo means servo receive data,and these data are sent by PLC.The function codes of receive PDO(COB-ID) are as follows:

1、 0x200+ Station No. of Servo
2、 0x300+ Station No. of Servo
3、 0x400+ Station No. of Servo
4、 0x500+ Station No. of Servo

FD2S Servo is designed according to the standard of CANopen 2.0A protocol,and it also supports CANopen 2.0B protocol.Therefore,if 8 PDOs are not enough,users can define new PDO,for example,set 0x43FH as the communication PDO of Station No.1,but it needs the controllers and servo define PDO by the same rule.

PDO transmission types:

PDO supports two transmission mode:

SYNC: Transmission is triggered by the synchronization message（Transmission type:0-240）

In this transmission mode, controller must have the ability to send synchronous messages（The message is sent periodically at a maximum frequency of 1KHz）,and servo will send after receiving the synchronous message.

Acyclic:Pre-triggered by remote frame,or by specific event of objects speicficed by the equipment sub-protocol.In this mode,servo will send out data as soon as receiving the data of synchronous message PDO.

Cyclic:Triggered after sending 1 to 240 SYNC messages.In this mode,servo will send out data in PDO after receiving n SYNC messages.

ASYNC(Transmission Type:254/255):

Slave sends out message automatically as soon as the data change,and it can define an interval time between two messages which can advoid the one in high priority always sending message.(The smaller number of PDO,the higher its priority)

PDO Inhibit Time:

Each PDO can define an inhibit time,that is the minimum interval time between two continuous PDO transmission.It is used to advoid the PDO in higher priority always occupying the communication.The inhibit time is 16bit unsigned integer,its unit is 100us.

Protection mode（Supervision type）

Supervision type is to choose which way master uses to check slave during operation,and check whether slave is error or not and handle the error.

Heartbeat message:Slave send message to master cyclically during supervision time.If master hasn't received the message from slave after heartbeat time,then master will consider slave as error.

Message format

(0x700+NodeID)+Status

Status：

0：Start 4:Stop 5:Run 127:Pre-operational

Node Guarding: Slave send message to master cyclically during supervision time.If master hasn't received the message from slave after supervision time,then master will consider slave as error.

The format of master request message:

（0x700+NodeID）（No data in this message）

Format of slave response message:

（0x700+NodeID）+Status:

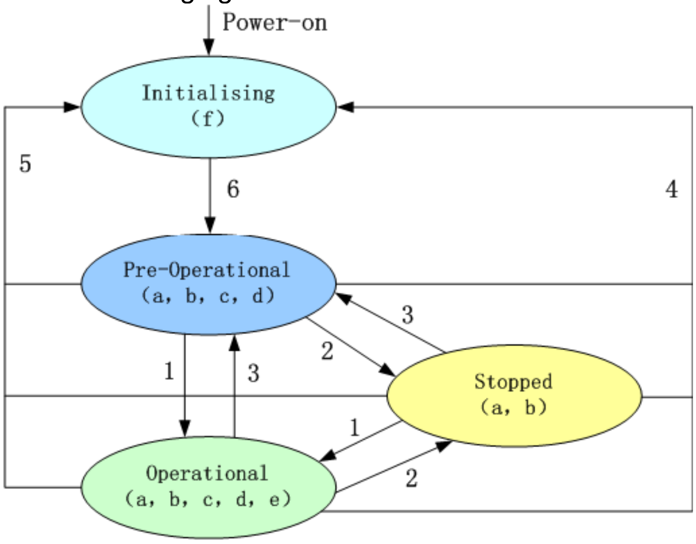
Status:

The bit7 of the data is triggered bit.This bit will alternately set to 0 or 1 in the response message.It will be set to 0 at the first request of node guarding.The bit0 ~ bit6 indicate the status of node.

Status: 0:Initialization 1:No connection 2.Connection 3:Operational 4:Stop 5:Run 127:Pre-operational

Normally standard CAN slave only one protection mode,but FD2S Servo can support both.

Boot-up process
The boot-up process is shown in following figure.



Note:

►The letters in the parenthesis means the objects which can used in this status:
a. NMT ， b. Node Guard ， c. SDO ， d. Emergency ， e. PDO ， f. Boot-up
► State transition（1-5 are sent by NMT service）,NMT command as shown in the parenthesis:
1：Start_Remote_node (0x01)
2：Stop_Remote_Node (0x02)
3：Enter_Pre-Operational_State (0x80)
4：Reset_Node (0x81)
5：Reset_Communication (0x82)
6：Initialization finish,enter pre-operational status and send boot-up message.

NMT management message can be used to change the modes.Only NMT-Master node can send NMT Module Control message,and all slave must support NMT Module Control service,meanwhile NMT Module Control message needn't response.The format of NMT message is as follows:

NMT-Master ➔ NMT-Slave(s)

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

When Node-ID is 0,then all the NMT slave device are addressing.CS is command,its value is as follows:

Command	NMT Service
1	Start Remote Node
2	Stop Remote Node
128	Enter Pre-operational State
129	Reset Node
130	Reset Communication

For example, If you want a node in the operational status to return to the pre-operational status,then the controller needs to send following message:

0x000:0x80 0x02

10.3.3 CANopen Communication Parameters

LED	Internal	Name	Meaning	Default

Display	Address			Value
d5.00	2FF00108	Store_Loop_Data	1：Save all control parameters except motor parameters 10：Initialize all control parameters except motor parameters	0
d5.01	100B0008	ID_Com	Driver station No. Note:It needs to save and restart driver after changing this parameter.	1
	2F810008	CAN_Bandrate	Baudrate of CAN port: Note: It needs to save and restart driver after changing this parameter.This parameter can only set in KincoServo software.	50

10.3.4 CANopen Communication Address of Servo Parameters

About the objects of each operation mode please refer to Appendix.
About common object address please refer to object list in Appendix.
About all the communication address please refer to parameters list.
About CANopen communication example please refer to Appendix.

Chapter 11 Alarm and Troubleshooting

11.1 Alarm Messages

Digital flickering on the display indicates that an alarm occurs indicating that the driver is faulty. For details about faults, see Table 11-1 "Fault codes". A code of the alarm message is represented by a hexadecimal data, and four numeric displays appear. If the driver is faulty, the corresponding bits in the alarm codes are set to "1". For example, if an encoder is not connected, the 1st and 2nd bits of the faulty code are set to "1". As a result, "0006" is displayed.

Table 11-1 Fault codes

1st bit in numeric display (left)				2nd bit in numeric display				3rd bit in numeric display				4th bit in numeric display (right)			
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EEPROM Error	Commutation	STO Error	Over Frequency	Ilt Error	Logic Voltage	Following Error	Chop Resistor	Over Current	Low Voltage	Over Voltage	Over Temperature	Encoder Counting	Encoder UVW	Encoder ABZ	Internal

A maximum of 7 generated alarms can be stored in the driver. For details, enter the menu of Group F007. Press **Enter**. The interface of faulty codes is displayed. The errors that you first discovered are those that have occurred most recently. Press ▲ or ▼ to browse the messages of historical alarms. If the decimal point at the lower right corner in the second bit of the numeric display is on, it indicates that the earliest alarm message is just browsed; if the decimal point at the lower right corner in the third bit of the numeric display is on, it indicates that the latest alarm message is just browsed.

For details on error messages, you need to access PC software via a communication port to check the working status of the driver when an error occurs. Here are some messages of the driver for your reference:

1. Error codes;
2. Bus voltage when an error occurs;
3. Motor speed when an error occurs;
4. Motor current when an error occurs;
5. Driver temperature when an error occurs;
6. Working mode of the driver when an error occurs;
7. Accumulated working time of the driver when an error occurs;

11.2 Alarm Causes & Troubleshooting

Alarm code	Alarm Information	Alarm Cause	Troubleshooting
FFF.F /800.0	No motor configured	There is no motor type set in servo driver	Set the motor type in d4.01.
000.1	Internal	Internal problem	Please contact manufacturer
000.2	Encoder ABZ	The ABZ signal cable is disconnected.	Check the cable.
000.4	Encoder UVW	The UVW signal cable is disconnected.	Check the cable.
000.8	Encoder Counting	Interferences are suppressed. Encoder cable problem	Check encoder cable. Remove interference(Such as connect the motor cable to SHIELD terminal etc.)
000.6	Encoder Error	ABZ and UVW signals of the encoders incur error simultaneously.	Check the cable.
001.0	Over Temperature	The driver temperature exceeds 83°C.	Check whether the selected driver has enough power.
002.0	Over Voltage	The bus voltage of the driver exceeds the allowable range.	Check the input voltage,or determine whether a braking resistor is connected.
004.0	Low Voltage	The voltage of the driver bus is below the allowable range.	Check the input power. Power on AC first,then power DC. Reduce deceleration.
008.0	Over Current	The power tube in the driver is faulty, or short circuit occurs on the phase line of the motor.	Check motor wires. If the motor works properly, it can be judged that faults occur on the power tube in the driver.
010.0	Chop Resistor	The actual power of brake resistor is larger than rated power	Change brake resistor.
020.0	Following Error	Control loop parameters setting problem. Overload or block. Encoder signal problem.	Set VFF (d2.08) as 100%,increase kpp(d2.07) and kvp(d2.01). Choose bigger power motor or check whether the load is blocked. Check the encoder cable.
040.0	Logic Voltage	The logic voltage is lower than 18V.	Check the logic power supply 24V.
080.0	IIt Error	Control loop parameters setting problem. Overload or block.	Increase kvp(d2.01). Choose bigger power motor or check whether the load is blocked.
100.0	Over Frequency	The input pulse frequency exceeds the allowable maximum value.	Check the input pulse frequency and the maximum permissible value of the frequency.（d3.38）。
200.0	STO Error	STO Error	Check the wiring according to Chapter 3.4.
400.0	Commutation	UVW signal of encoder cable problem	Check encoder cable.
800.0	EEPROM Error	Because of updating firmware. Driver internal problem.	Initialize all control parameters and save,then restart driver. Contact manufacturer.
888.8	Driver abnormal working states	Logic power supply problem. Driver internal problem.	Check 24VDC power supply. Contact manufacturer.

Chapter 12 Appendix

Appendix 1 Instructions of operation mode via Communication

1. Position mode(Mode 1)

Take this mode for example: In the coordinate system shown below, the red arrow indicates the current 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.

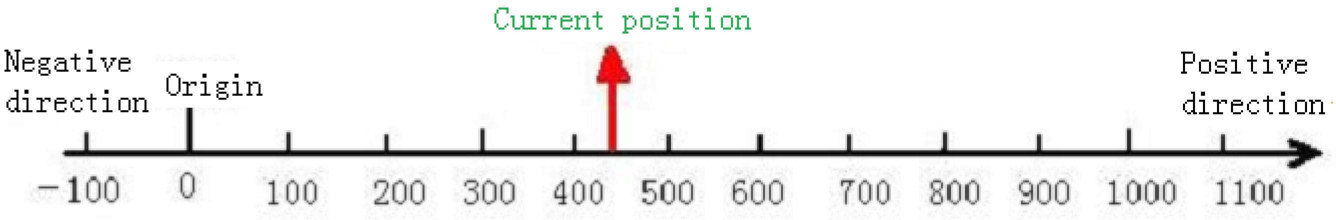


Fig.1 Absolute/Relative positioning

In mode 1, the following objects have to be defined :

CANopen Address	Modbus Address	Value	Meaning
60600008	0x3500	1	Set as position mode
60810020	0x4A00	User setting	Profile velocity
60830020	0x4B00	User setting	Acceleration
60840020	0x4C00	User setting	Deceleration
607A0020	0x4000	User setting	Target position
60400010	0x3100	2F -> 3F 4F -> 5F 103F 105F	Start absolute positioning Start relative positioning Start absolute positioning while target position change Start relative positioning while target position change

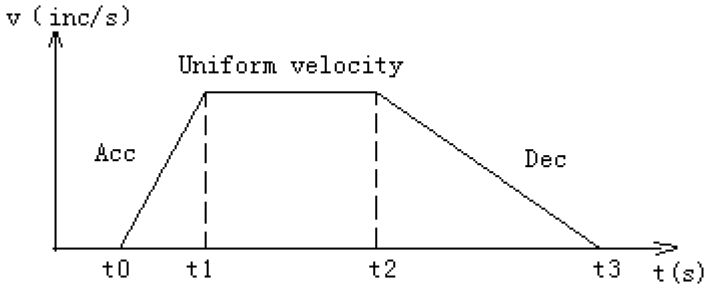
More details please refer to “Mode and Control” and “Target Object” in Appendix.
About position mode controlled by communication,please refer to communication example in Appendix.

2. Speed Mode(Mode -3 or 3)

Mode 3 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.

Vt＝Vo＋at　　　　　Vt－Uniform velocity
Vo－Initial velocity
a - Acceleration or deceleration
t - Acceleration time

$S = Vot + (1/2) at^2$ S—Acceleration displacement



Velocity and time curves in mode 3

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

CANopen 地址	Modbus Address	Value	Meaning
60600008	0x3500	3 or -3	Set as speed mode
60FF0020	0x6F00	User setting	Target velocity
60830020	0x4B00	User setting	Acceleration
60840020	0x4C00	User setting	Deceleration
60400010	0x3100	F	Start running

More details please refer to “Mode and Control” and “Target Object” in Appendix.
About position mode controlled by communication,please refer to communication example in Appendix.

3. Master-slave mode(Mode -4)

In this mode, the movement of the motor is directly controlled by the external encoder, pulse/direction, CW/CCW pulse signal from the X1 interface of the drive. If the system receives signal from the external encoder, 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 X1 interface to perform the following movement. The velocity rate of the following movement can be set by the electronic gear ratio.
In mode -4, the following objects have to be defined:

CANopen Address	Modbus Address	Value	Meaning
60600008	0x3500	-4	Set as master-slave mode
25080110	0x1910	User setting	Factor of electronic gear
25080210	0x1920	User setting	Divider of electronic gear
25080310	0x1930	User setting	Pulse mode 0...CW/CCW mode 1... Pulse/Direction mode 2...Incremental encoder mode Note:This parameter must save after change.
60400010	0x3100	F	Start running

More details please refer to “Mode and Control” , “Target Object” and “Master-slave mode” in Appendix.

4.Torque Mode(Mode 4)

In this mode, the motor will output at constant torque. The output torque depends on the value of target torque.The conversion formula is $T_{demand} = K_t * \frac{I_{demand}}{\sqrt{2}}$, K_t is torque constant,users can find it in the catalog. I_{demand} is peak current.

In mode 4, the following objects have to be defined:

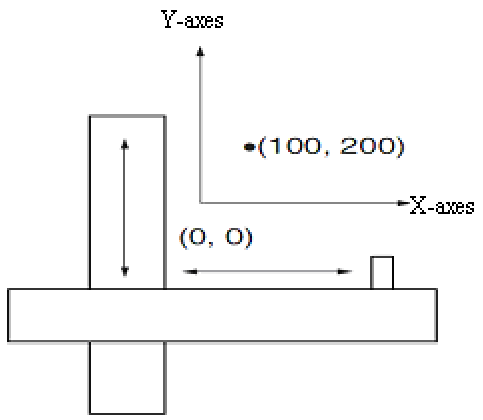
CANopen Address	Modbus Address	Value	Meaning
60600008	0x3500	-4	Set as torque mode
60710010	0x3C00	User setting	Target torque
60730010	0x3D00	User setting	Max. current
60800010	0x4900	User setting	Max. speed
60400010	0x3100	F	Start running

More details please refer to “Mode and Control” and “Target Object” in Appendix.

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.

5. Homing mode(Mode 6)

To make a system execute positioning in accordance with its absolute positioning, the first step is to define the origin. For instance, as shown in the following XY plane, to navigate to (X, Y) = (100mm, 200mm), you must define the origin of the machine firstly. It’s necessary to define the origin.



In mode 6, the following objects have to be defined:

CANopen Address	Modbus Address	Value	Meaning
60600008	0x3500	6	Set as homing mode
607C0020	0x4100	User setting	Home offset
60980008	0x4D00	User setting	Homing method
60990120	0x5010	User setting	Homing speed for searching home signal
60990220	0x5020	User setting	Homing speed for searching index signal
609A0020	0x5200	User setting	Homing acceleration
60400010	0x3100	F->1F	Start running

More details about homing method please refer to homing methods in Appedix.

6. Driver Status Display

FD2S Servo driver uses object 60410010(Modbus address is 0x3200) to indicate the current status of driver.The definitions of every bit are as following:

bit	Definition	Meaning	Value
0	Ready to Switch on	Ready to switch on	60410010＝0x0001
1	Switched On	Already switched on	60410010＝0x0002
2	Operation Enable	Operation enable	60410010＝0x0004
3	Fault	Driver fault	60410010＝0x0008
4	Voltage Disable	Voltage output disable	60410010＝0x0010
5	Quick Stop	Emergency stop	60410010＝0x0020
6	Switch On Disable	Switch on disable	60410010＝0x0040
7	Warning	Warning	60410010＝0x0080
8	Manufacturer specific 1	Reserved	60410010＝0x0100
9	Reserved 1	Reserved 1	60410010＝0x0200
10	Target Reached	Target position reach	60410010＝0x0400
11	Internal Limit Active	Internal limit active	60410010＝0x0800
12	Setp.Ach./v=0/Hom.att.	Pulse response	60410010＝0x1000
13	Foll.Err./Res.Hom.Err.	Following error/Reference error	60410010＝0x2000
14	Commutation Found	Commutation found	60410010＝0x4000
15	Reference Found	Reference found	60410010＝0x8000

Appendix 2:Example for CANopen Communication

1.Canopen communication between Kinco F1 PLC and FD2S Servo

1.1 Wiring diagram

F1 PLC CAN port FD2S CAN port (X4)

CAN_L 2 ---------------------------------- CAN_L 2

CAN_H 7 ---------------------------------- CAN_H 7

■**Note:**

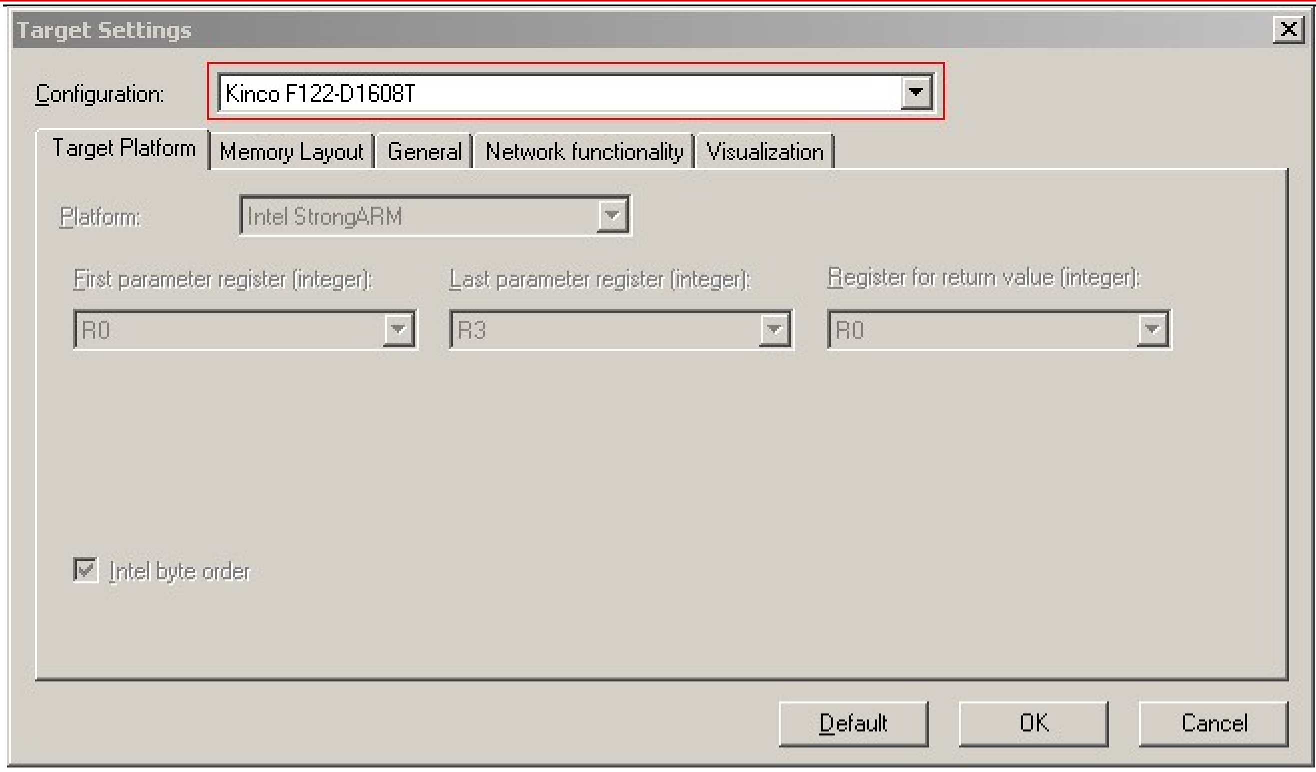
1.It must use series connection for multiple slaves.
2.CAN1 and CAN2 of F1 PLC are separately,can be used at the same time.
3.There are terminal resistors in PLC which set by DIP switch.Therefore,it needs a 120ohm terminal resistor in the end of the communication cable(In the last slave).

1.2 Parameter setting.

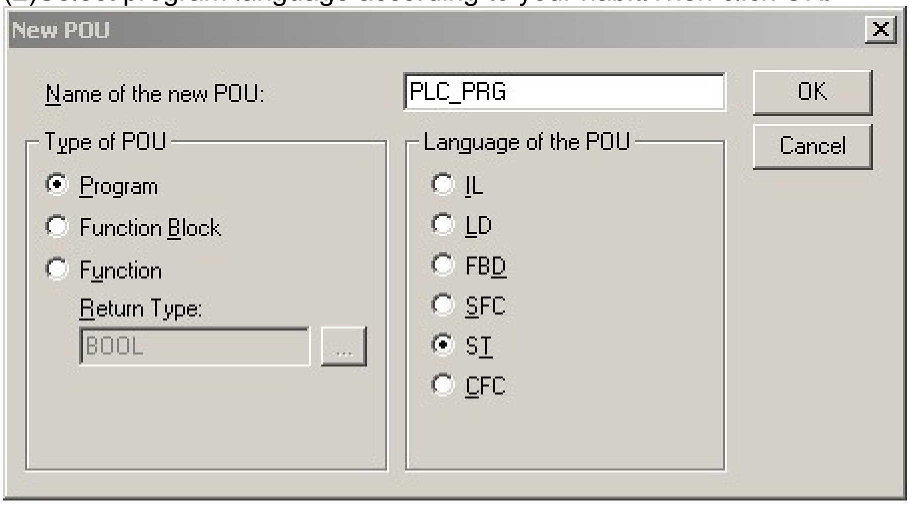
About the settings of FD2S parameters such as baudrate and station No.,please refer to the chapter of CANopen.

1.3 Software program

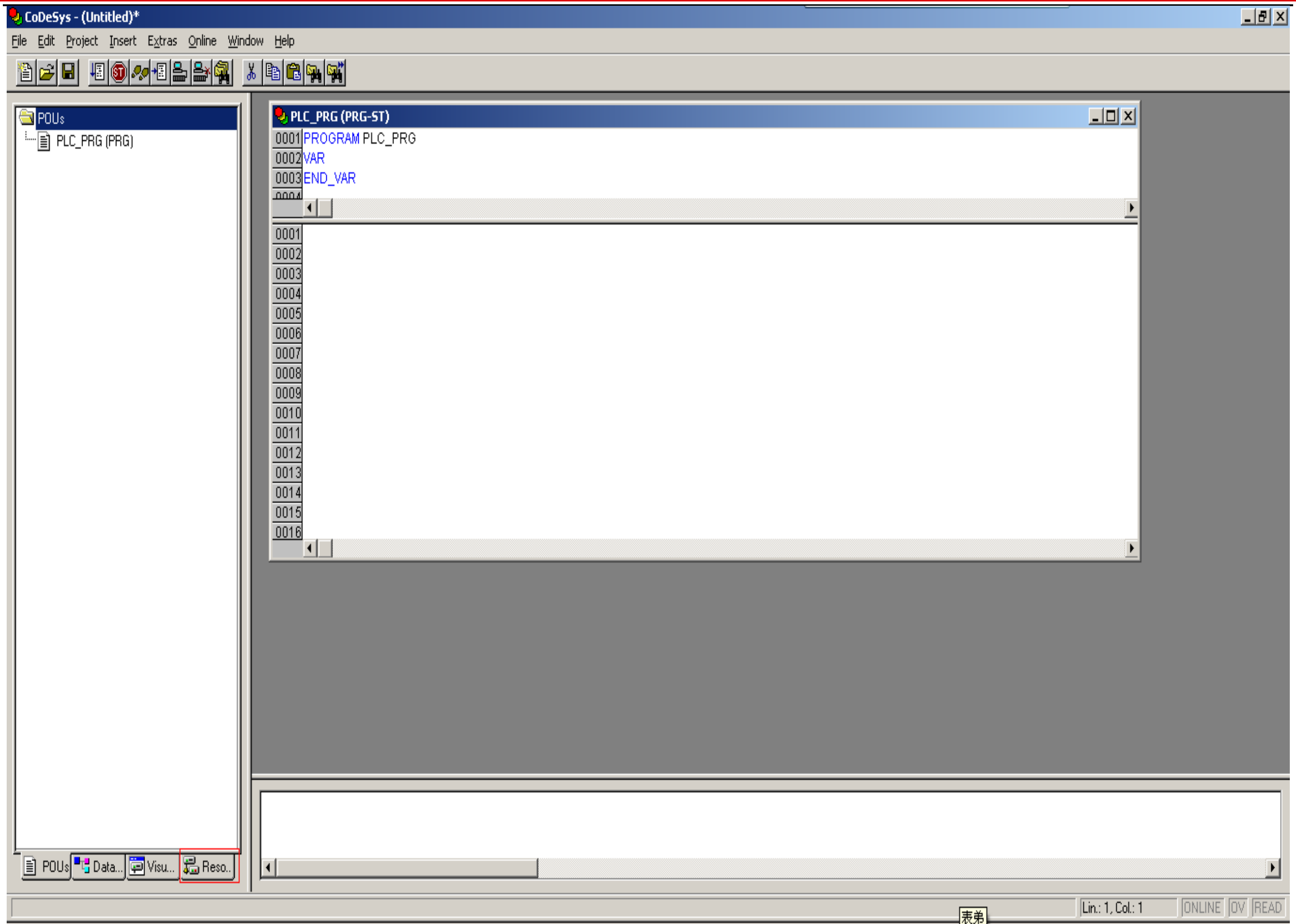
(1)Create new project,select Kinco F122-D1608T and click OK.

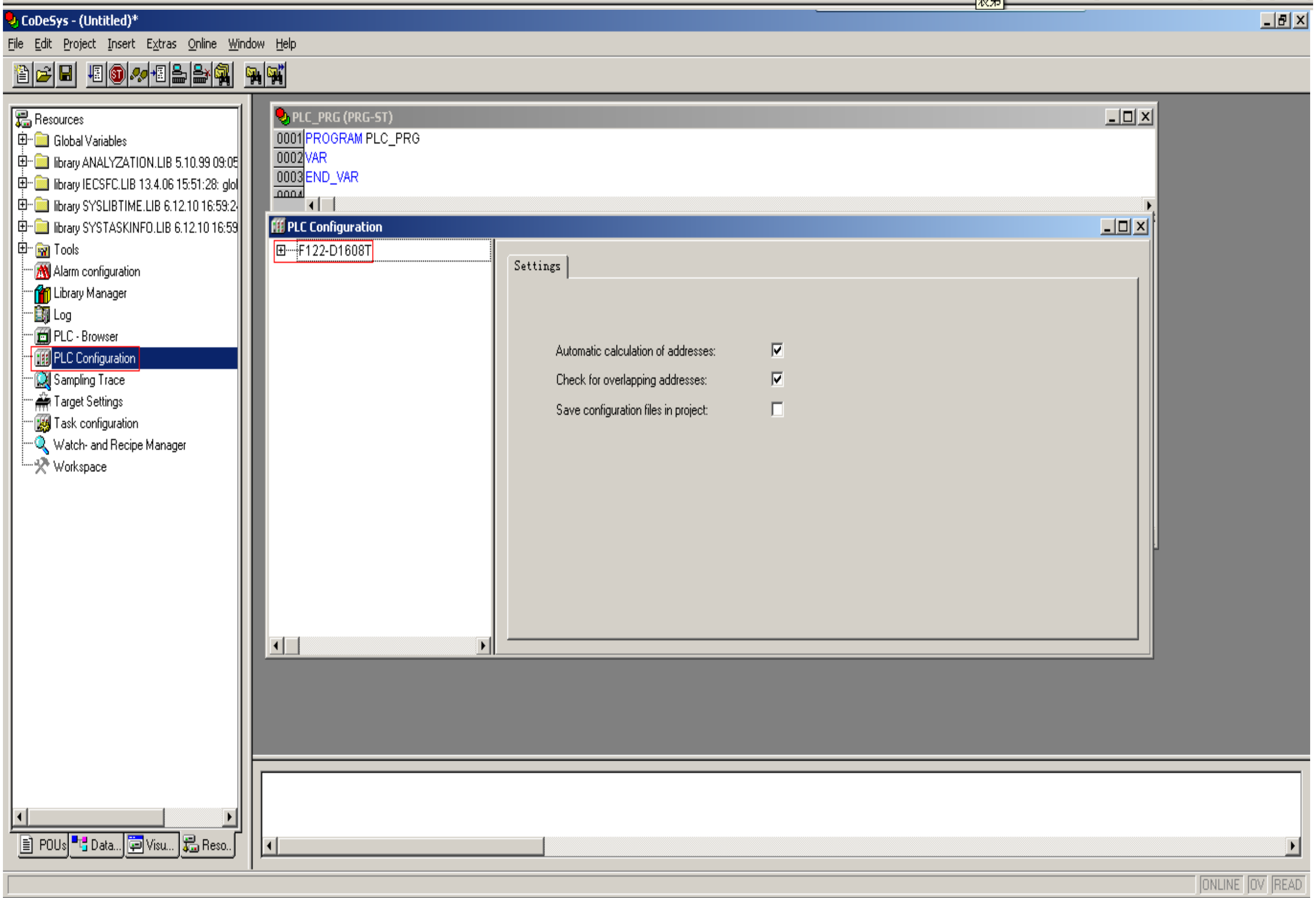


(2)Select program language according to your habit.Then click OK.

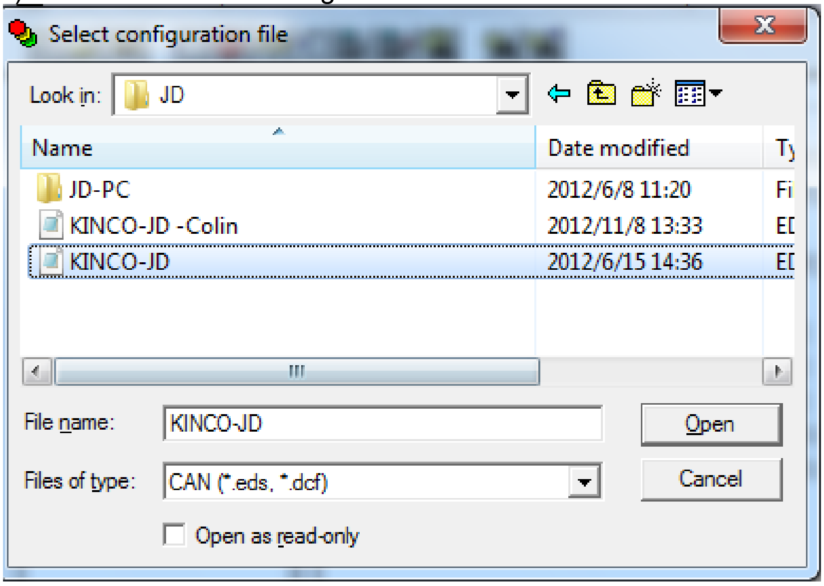


(3)Select “Resources” option and click “PLC Configuration”.

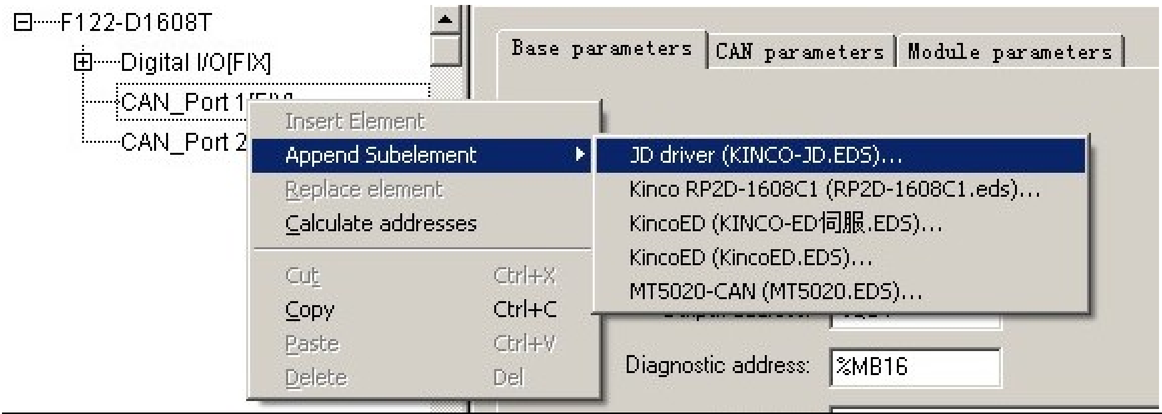

CoDeSys - (Untitled)*
File Edit Project Insert Extras Online Window Help
POUs
PLC_PRG (PRG)
PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
POUs Data... Visu... Reso...
Lin.: 1, Col.: 1 ONLINE OV READ

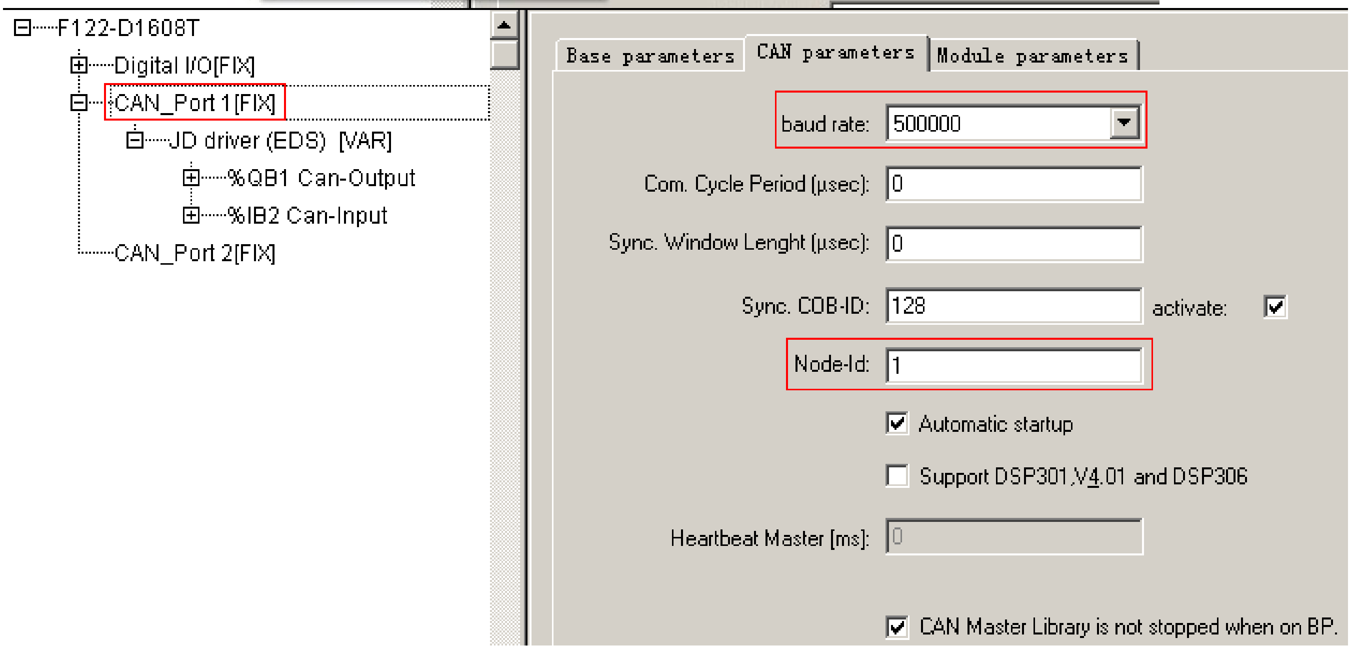

CoDeSys - (Untitled)*
File Edit Project Insert Extras Online Window Help
Resources
Global Variables
library ANALYZATION.LIB 5.10.99 09:05
library IECSFC.LIB 13.4.06 15:51:28: glol
library SYSLIBTIME.LIB 6.12.10 16:59:2
library SYSTASKINFO.LIB 6.12.10 16:59
Tools
Alarm configuration
Library Manager
Log
PLC - Browser
PLC Configuration
Sampling Trace
Target Settings
Task configuration
Watch- and Recipe Manager
Workspace
PLC_PRG (PRG-ST)
0001 PROGRAM PLC_PRG
0002 VAR
0003 END_VAR
PLC Configuration
F122-D1608T
Settings
Automatic calculation of addresses:
Check for overlapping addresses:
Save configuration files in project:
POUs Data... Visu... Reso...
ONLINE OV READ

(4)Click “Extras->add configuration file” to add EDS file of FD2S Servo.

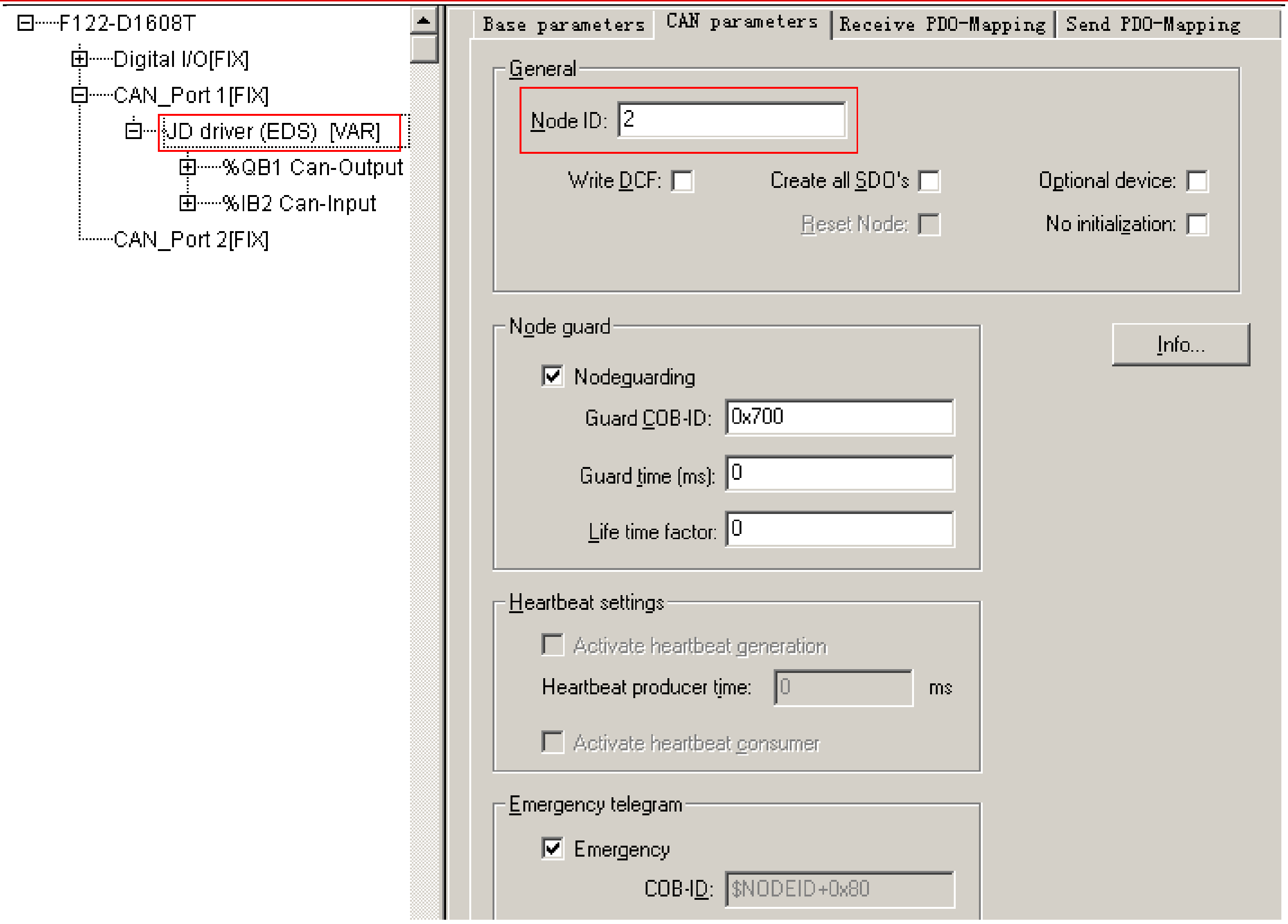


(5)There are two CAN ports in F1 PLC.Both of them can be used as master. Set baudrate and Node-ID for CAN port.If you need synchronous message,please click “activate” ,then set “Com.Cycle period” and “Sync.COB-ID”.

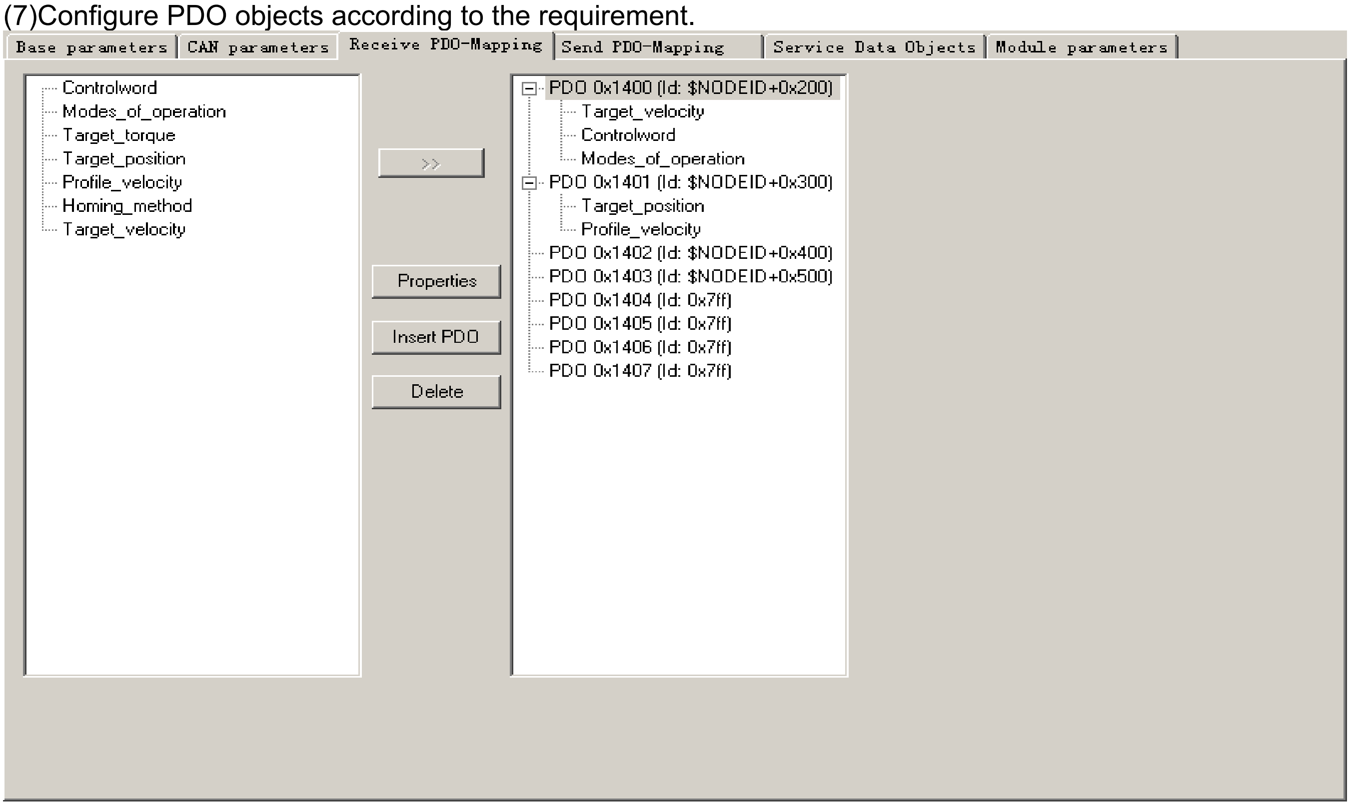


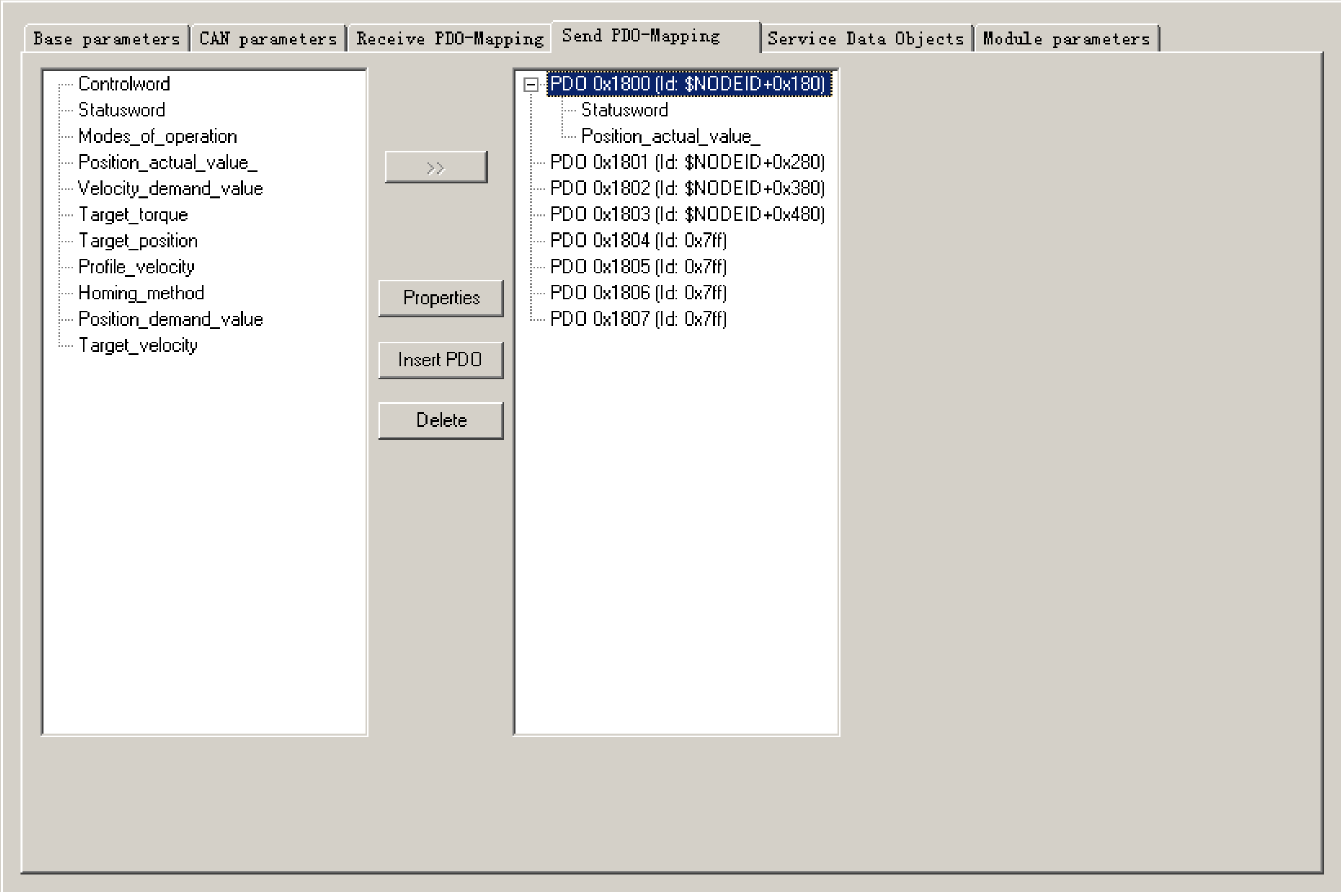


(6) Right click CAN port and select “Append Subelement->FD2S driver” to add slaves.Then set parameters such as Node ID,Nodeguarding,RX-PDO and TX-PDO.

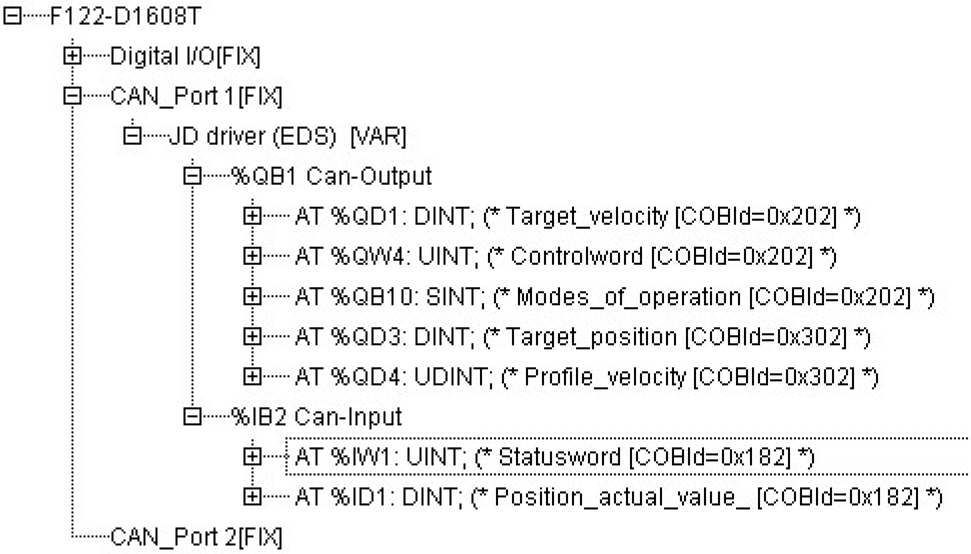


(7)Configure PDO objects according to the requirement.

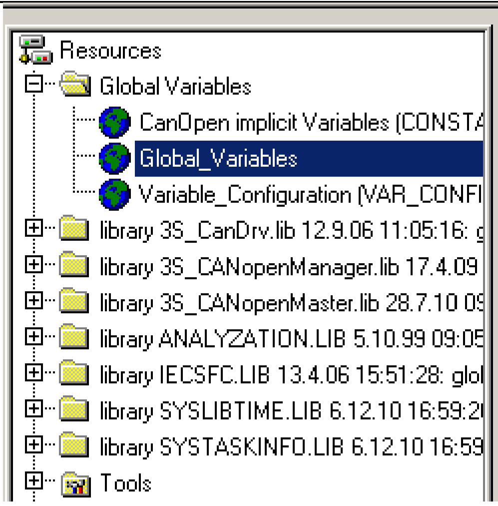




(8)After configure all the parameters,there will be all the registers corresponding to all the OD as shown in following figure.For example,the register for Controlword is QW4,and the register for Statusword is IW1.8.



(9)Configure other slaves according to procedure above.Then we can start to program.In the program,we can use the register directly or define gloable variables.




```
VAR_GLOBAL
    tv AT %QD1:DINT;
    cw AT %QW4:UINT:=6;
    mo AT %QB10:SINT;
    TP AT %QD3:DINT;
    pv AT %QD4:DINT;
    ap AT %ID1:DINT;

    tv3 AT %QD5:DINT;
    cw3 AT %QW12:UINT:=6;
    mo3 AT %QB26:SINT;
    TP3 AT %QD7:DINT;
    pv3 AT %QD8:DINT;
    ap3 AT %ID3:DINT;
END_VAR
```

(10)The program is as following figure.More details please refer to the chapter of operation mode.**<u>After creating communication between F1 PLC and servo,it needs to set a initial value 6 to the object “Controlwrod”,or other command can’t be effective in servo.</u>**

```
PROGRAM JD2
VAR
    m1: BOOL;
    m2: BOOL;
    spv: DINT;
    m3: BOOL;
    m4: BOOL;
    m5: BOOL;
    actual_pos: DINT;
    m6: BOOL;
END_VAR
```

```
actual_pos:=ap;
(* power off*)
IF m1=1 THEN
tv:=27300;
cw:=6;
mo:=3;
m1:=0;
END_IF
(* velocity*)

IF m2=1 THEN
tv:=273000;
cw:=47;
mo:=3;
m2:=0;
END_IF

(* absolute postion*)
IF m3=1 THEN
tp:=0;
pv:=2730000;
cw:=63;
mo:=1;
m3:=0;
END_IF
```

If the objects are not in the EDS file or not commonly use,then we can use SDO to read and write these ojectes,as shown in following figure.

```
END_VAR

(*Write velocity by SDO*)
sdo1(     Enable:= m9,(*Enable*)
    wDrvNr:= 0, (*Port number,0:CAN1, 1: CAN2*)
    ucNodeId:=2 ,(*Staion No. of slave*)
    wIndex:=16#60FF ,(*OD INDEX*)
    bySubIndex:=16#00 ,(*subINDEX*)
    ucModus:= 16#23, (*use 16#23 FOR 4-BYTE-write-request
                      use 16#27 FOR 3-BYTE
                      use 16#2B for 2-byte
                      use 16#2F for 1-byte
                      use 16#21 FOR downloading more than 4 bytes using the segmented transfer*)
    ucByte0 :=16#10,
    ucByte1 :=16#A8,
    ucByte2 :=16#29,
    ucByte3 :=00) ;
(*Read actual speed by SDO*)
sdo2(Enable:= m10, (*Enable*) wDrvNr:= 0, (*Port number,0:CAN1, 1: CAN2*)  ucNodeId:=2 ,(*Staion No. of slave*)
    wIndex:=16#606C ,(*OD INDEX*)
    bySubIndex:=16#00 ,(*subINDEX*)
    ucModus:=16#40);(*SDO-mode, use 16#40 for read-request.*)
IF sdo2.bAnswerRec THEN
   val := SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[7]),24);
   val := val + SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[6]),16);
   val := val + SHL(BYTE_TO_DWORD(sdo2.ucAnswerBytes[5]),8);
   val := val + BYTE_TO_DWORD(sdo2.ucAnswerBytes[4]);
END_IF
```

2.CANopen Communication between FD2S Servo and Peak CAN.

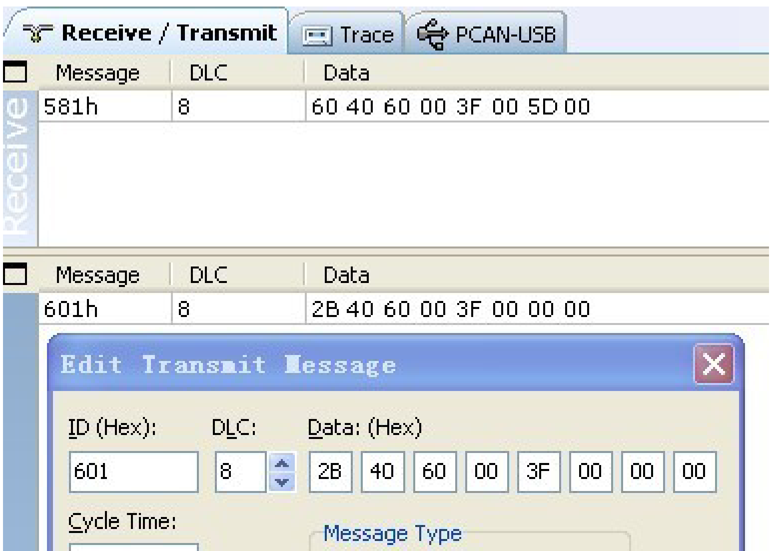
Peak company has many kinds of CAN adapter such as ISA,PCI,USB-CAN and so on.This example is to use PCAN-USB connected to FD2S Servo.

2.1 Wiring

Master	Slave 1	Slave 2	-----	Slave N
PCAN_USB CAN 口	FD2S X4 port	FD2S X4 port		FD2S X4 port
2 CAN_L	2CAN_L	2CAN_L		2CAN_L
7 CAN_H	7CAN_H	7CAN_H		7CAN_H

It needs to add a 120-150 ohm resistor between PIN2 and PIN7 in the terminal(Slave N).

2.2 Set the communication parameters such as baudrate,ID according to FD2S Servo.Then open PCAN-VIEW(Software for PCAN-USB) to send and receive data.
Following figure is the example to send command to set 6040 as 3F.The lower part of the figure is to send data,the upper part of the figure is to receive data.



Following is the example about sending and receiving messages for different operation mode.(The sataion No. is 1)

Homing mode（The controlword should change from F to 1F)				
Internal Address	Name	Setting value	Message（ID=1）	Note
60400010	Control word	F	601 2B 40 60 00 0F 00 581 60 40 60 00 0F 00	DEC=[(RPM*512*Encoder_resolution)/1875]
60600008	Operation mode	6	601 2F 60 60 00 06 00 581 60 60 60 00 06 00	
60980008	Homing method	33	601 2F 98 60 00 21 00 581 60 98 60 00 21 00	
60990120	Velocity for searching limit switch	200RPM	601 23 99 60 01 55 55 08 00 581 60 99 60 01 55 55 08 00	
60990220	Velocity for searching phase-N signal	150RPM	601 23 99 60 02 00 40 06 00 581 60 99 60 02 00 40 06 00	
60400010	Control word	1F	601 2B 40 60 00 1F 00 581 60 40 60 00 1F 00	
601 40 41 60 00 00 00 00 Read status word,C037 means reference found.				

Position mode（Control word should change from 2F to 3F for absolute positioning,and change from 4Fto5F for relative positioning.103F or 105F means activate immediately when position change.)				
Internal Address	Name	Setting value	Message（ID=1）	Note
60400010	Control word	F	601 2B 40 60 00 0F 00 581 60 40 60 00 0F 00	DEC=[(RPM*512*Encoder_resolution)/1875]
60600008	Operation mode	1	601 2F 60 60 00 01 00 581 60 60 60 00 01 00	
607A0020	Target velocity	50000inc	601 23 7A 60 00 50 C3 00 00 581 60 7A 60 00 50 C3 00 00	
60810020	Profile velocity	200RPM	601 23 81 60 00 55 55 08 00 581 60 81 60 00 55 55 08 00	
60830020	Acceleration	Default value 610.352rps/s	NULL	DEC=[(RPS/S*65536*Encoder_resolution)/1000/4000]
60840020	Deceleration	Default value 610.352rps/s	NULL	
60400010	Control word	2F(Absolute positioning)	601 2B 40 60 00 2F 00 581 60 40 60 00 2F 00	
		3F(Absolute positioning)	601 2B 40 60 00 3F 00 581 60 40 60 00 3F 00	
		4F(Relative positioning)	601 2B 40 60 00 4F 00 581 60 40 60 00 4F 00	
		5F(Relative positionin	601 2B 40 60 00 5F 00 581 60 40 60 00 5F 00	

		g)		
601 40 41 60 00 00 00 00 00 Read status word.D437 means target position reach.				
Speed mode				
Internal Address	Name	Setting value	Message（ID=1）	Note
60600008	Operation mode	3	601 2F 60 60 00 03 00 581 60 60 60 00 03 00	DEC=[(RPM*512*Encoder_resolution)/1875] DEC=[(RPS/S*65536*Encoder_resolution)/1000/4000]
60FF0020	Target velocity	150RPM	601 23 FF 60 00 00 40 06 00 581 60 FF 60 00 00 40 06 00	
60400010	Control word	F	601 2B 40 60 00 0F 00 581 60 40 60 00 0F 00	
60830020	Acceleration	Default value 610.352rps/s	NULL	
60840020	Deceleration	Default value 610.352rps/s	NULL	

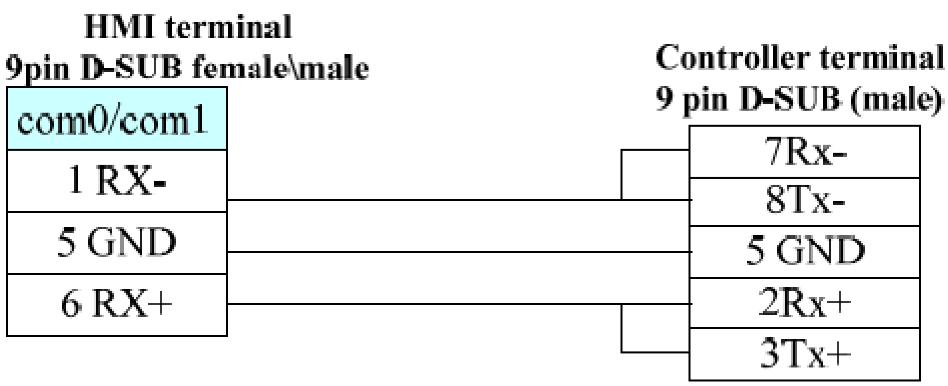
Note:All the data are Hexadecimal format when using communication.

Appendix 3:Example for RS485 Communication

1.Modbus Communication Between FD2S Servo and Kinco HMI

(1) HMI control single FD2S Servo.

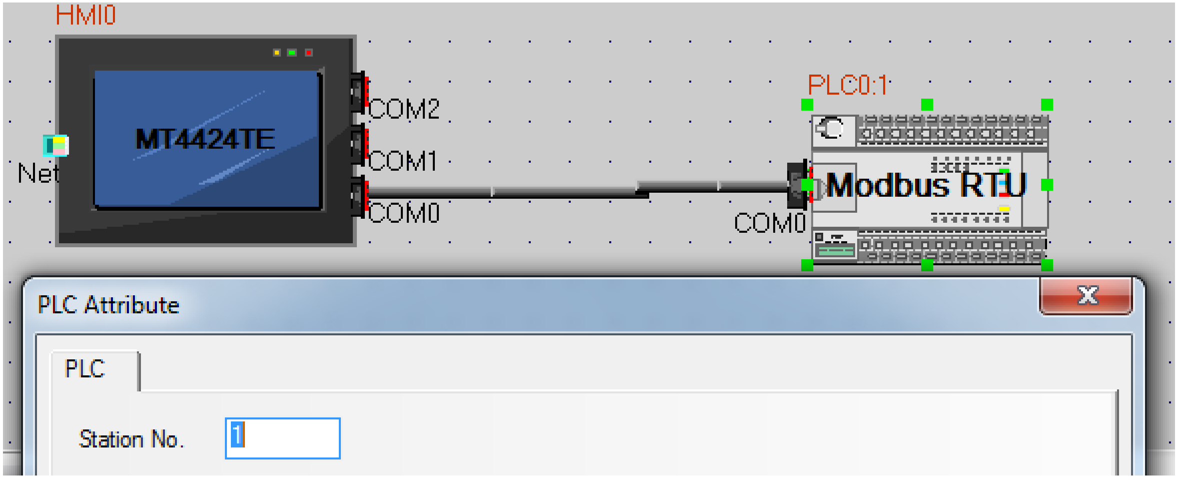
a. Wiring diagram

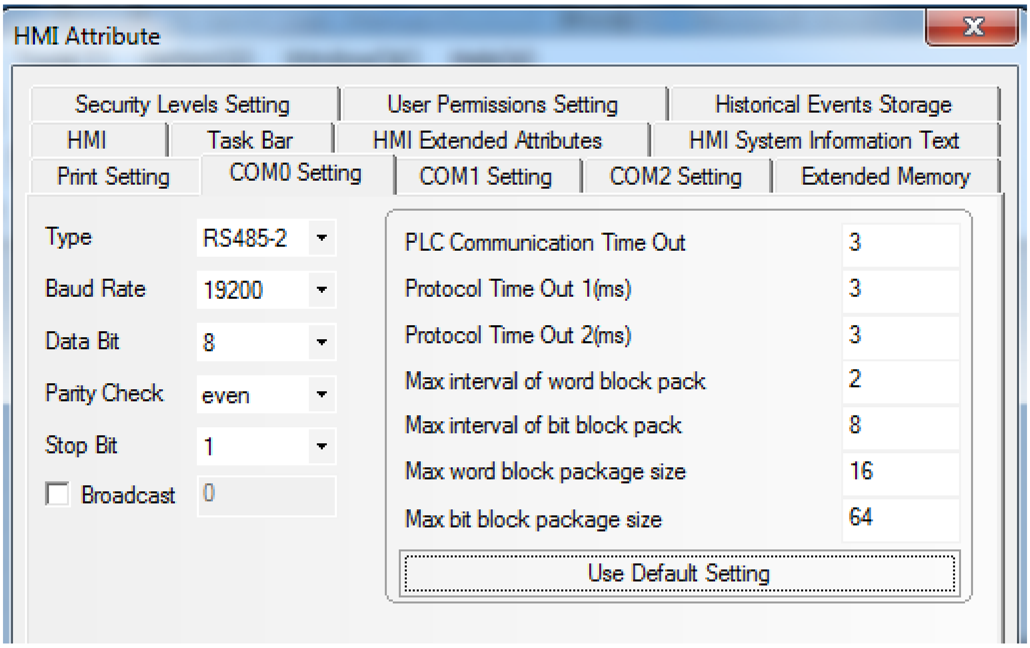


RS485 Communication

b. Parameters setting

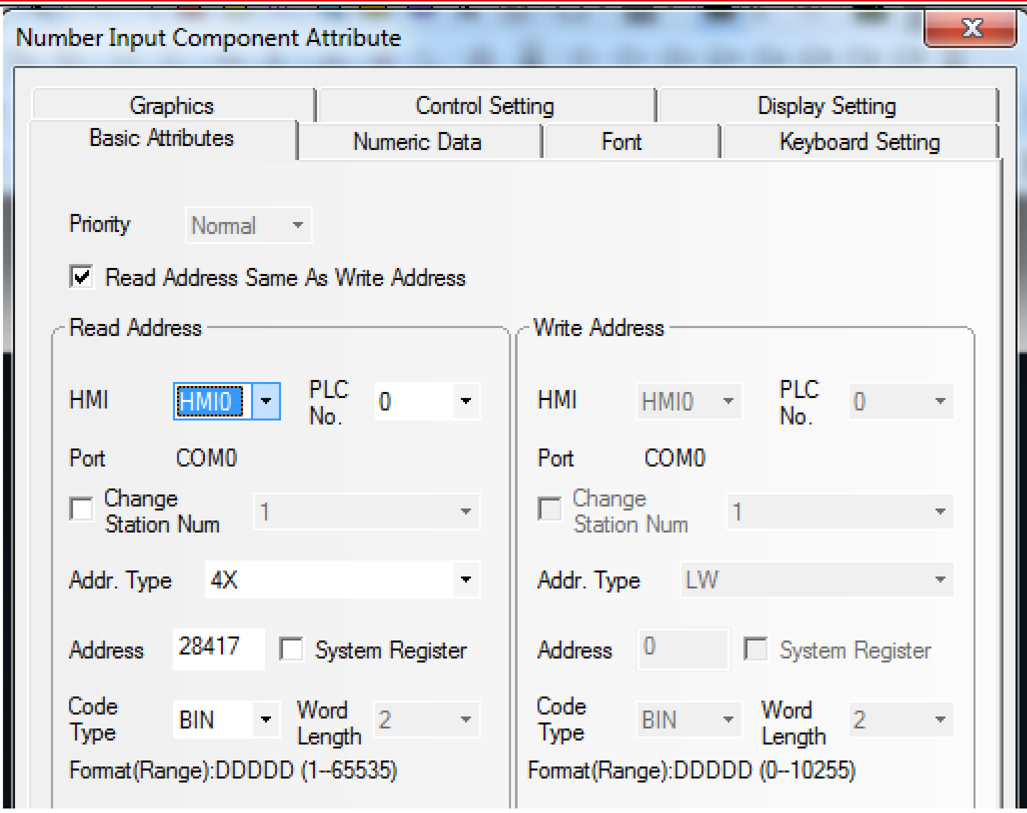
It needs to choose Modbus RTU in HMI software,the communication parameters are as following figure.The PLC station No. must be set the same as the ID of FD2S Servo.





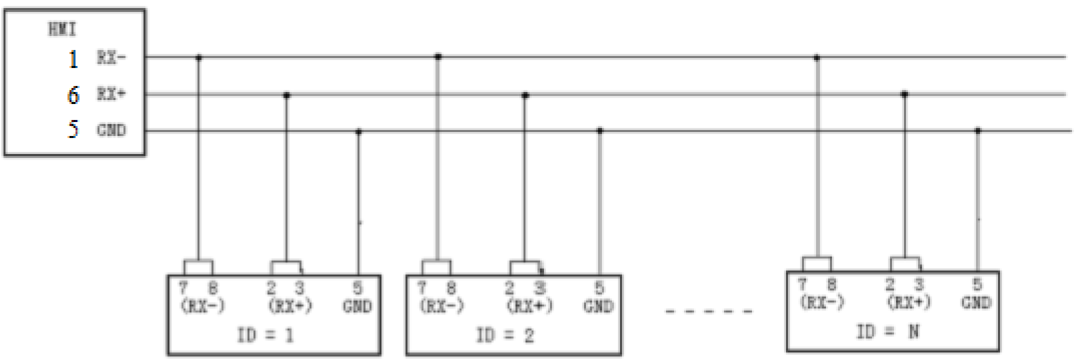
c. Address setting

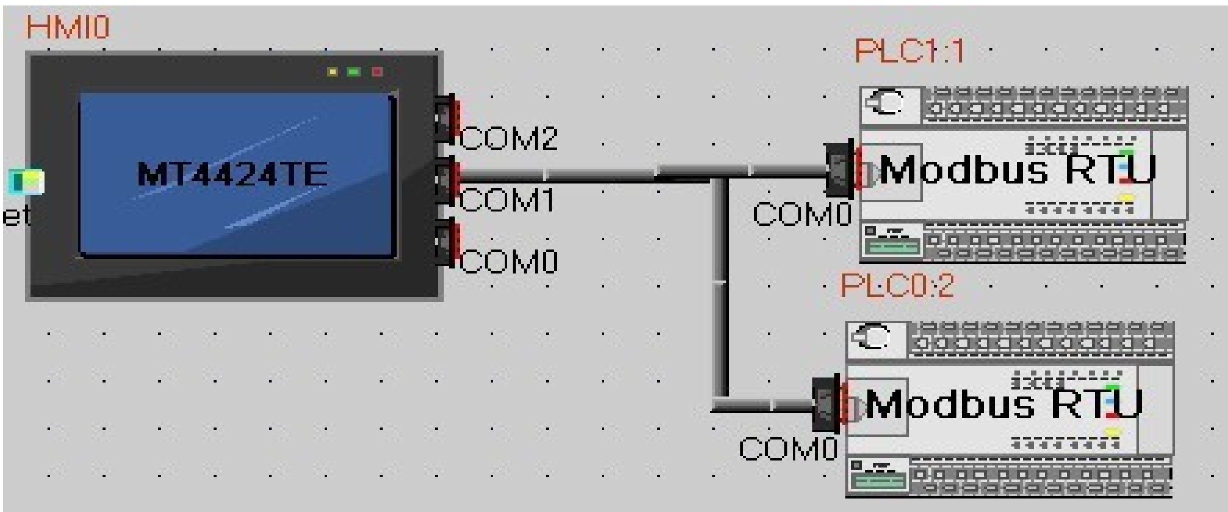
It needs to use address type 4X in HMI program(All the objects of FD2S Servo are corresponding to 4X).According to Modbus address of objects in the Common Object List,the Modbus address of the object “Target velocity”(60FF0020) is 0x6F00,its decimal value is 28416.When we use this address in HMI,we need to add 1,so in HMI the address for “Target velocity” is 28417 as shown in following figure.



(2)HMI control multiple FD2S Servo

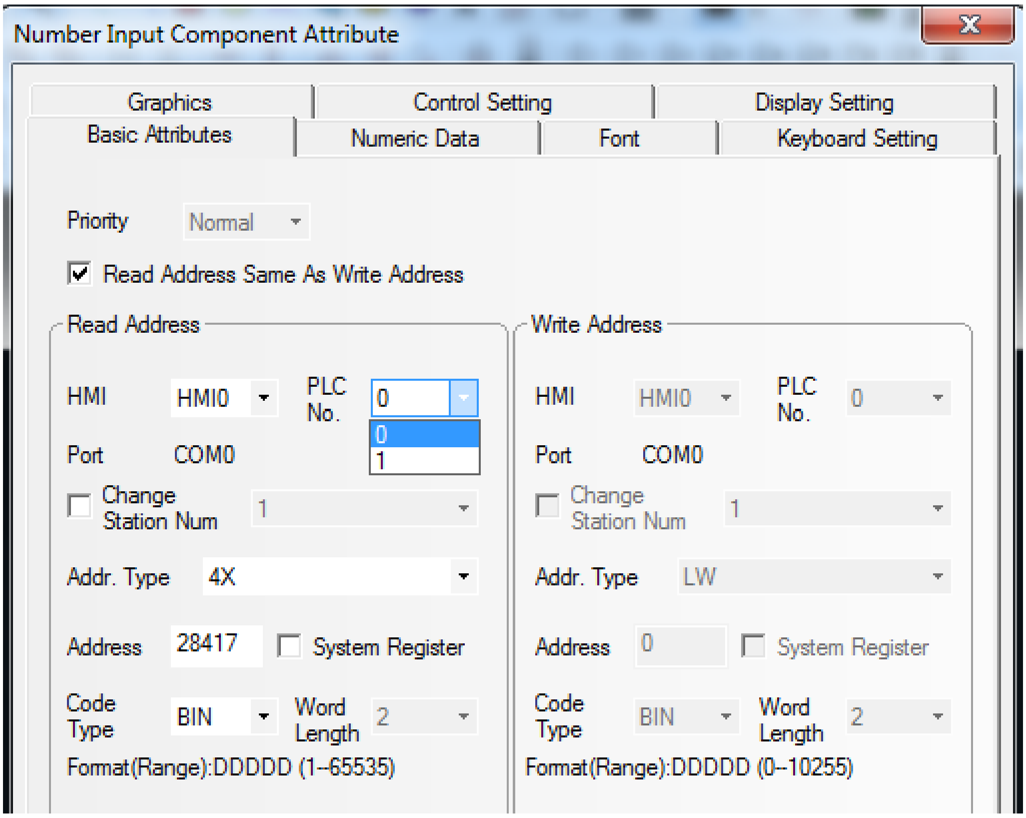
a、Wiring diagram





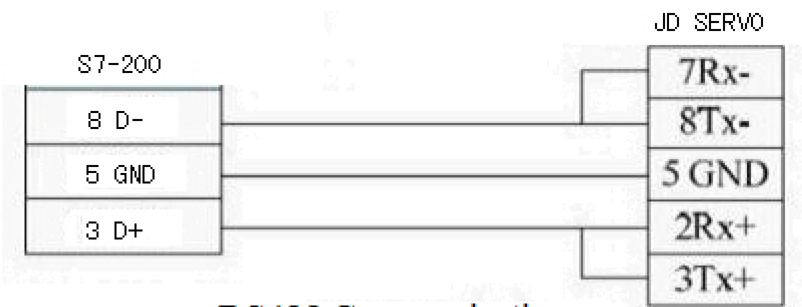
b. Parameter setting

The parameters setting in HMI is the same as above example,the difference is to set different station no. for different servo.In the attribute of components in HMI,it needs to select the PLC No. for different servo.(The PLC No. is not the servo station No.,as shown in the figure above,PLC0:2 means the PLC No. is 0,and station No. is 2)

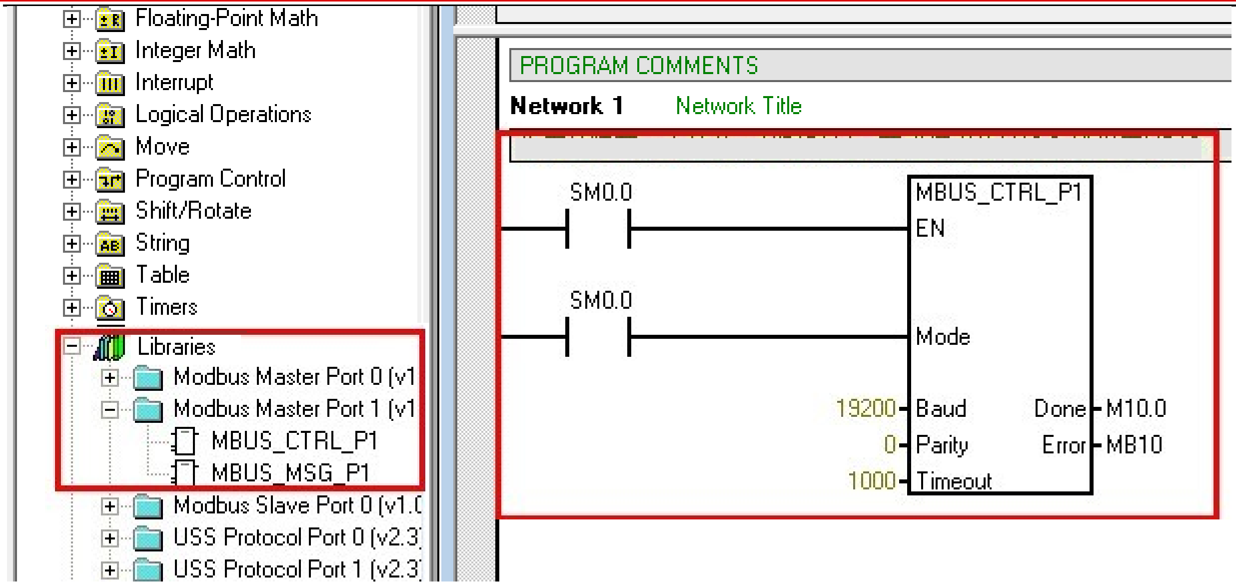


2. Modbus Communication Between FD2S Servo and Siemens S7-200

(1)Wiring diagram

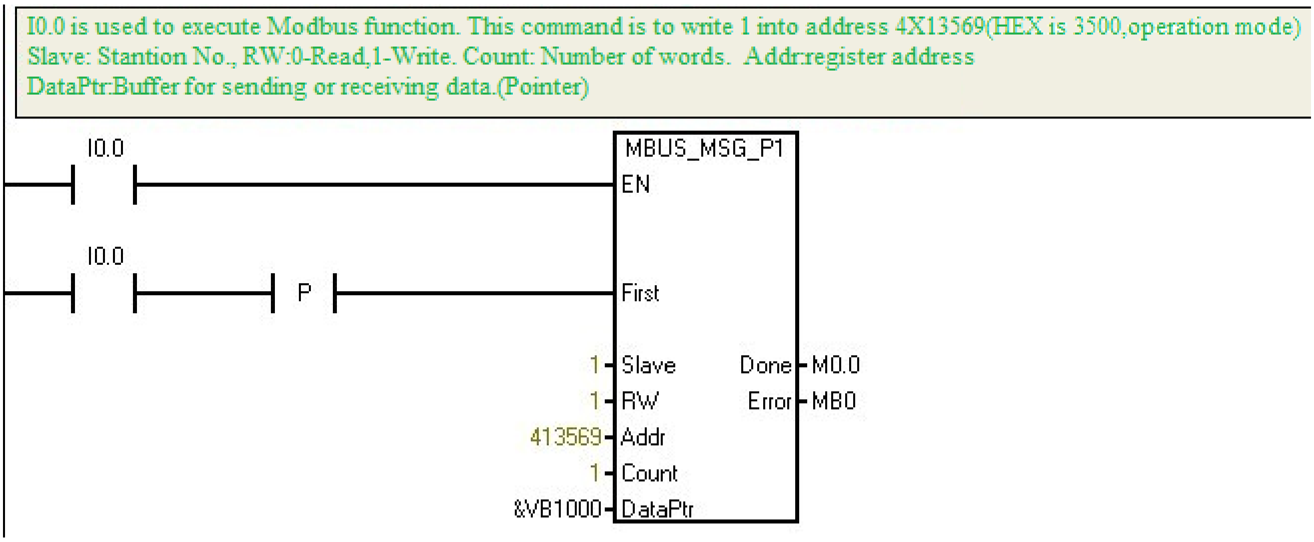


(2)Parameter setting.
About the parameter setting of FD2S Servo please refer to Chapter 10.2.The default parameters are Modbus RTU,19200,8,None,1.
In the software of S7-200 PLC,there is a library function used to set communication parameters as shown in following figure.



(3)Program

It needs to use the Modbus function (MODBUS_MSG) to send and receive data.The descriptions of Modbus function are shown in following figure.



(4)Example descriptions

S7200 plc Inputs	Function	Description
I0.0	Write 60600008=1	Set as position mode
I0.1	Wirte 607A0020=10000	Set the target position
I0.2	Write 60810020=1000rpm	Set the profile velocity
I0.3	Write 60400010=0x4F first,then 0x5F	Start relative positioning
I0.4	Read 60630020	Read the actual position
I0.5	Read 60410010	Read the status word

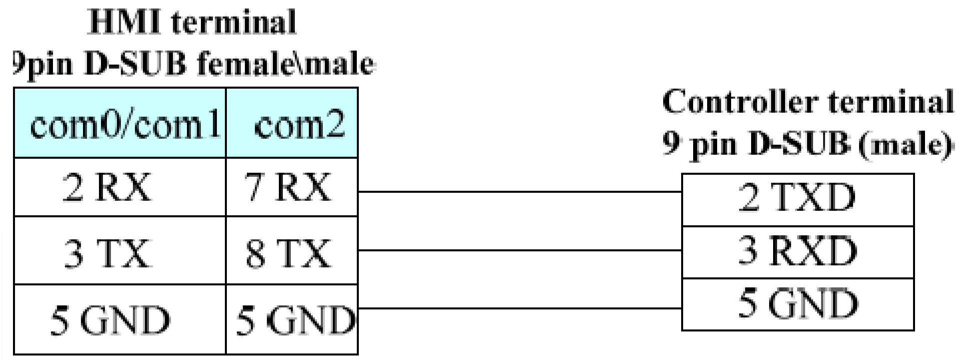
Appendix 4:Example for RS232 Communication

1.Communication between FD2S Servo and Kinco HMI.

Kinco MT4000 and MT5000 series HMI can communicate with RS232 port of FD2S Servo.Users can set internal parameters of FD2S Servo and display the status of FD2S Servo.Kinco HMI can communicate with single FD2S Servo,and also can communicate with multiple FD2S Servo via RS232.

(1) HMI control single FD2S Servo

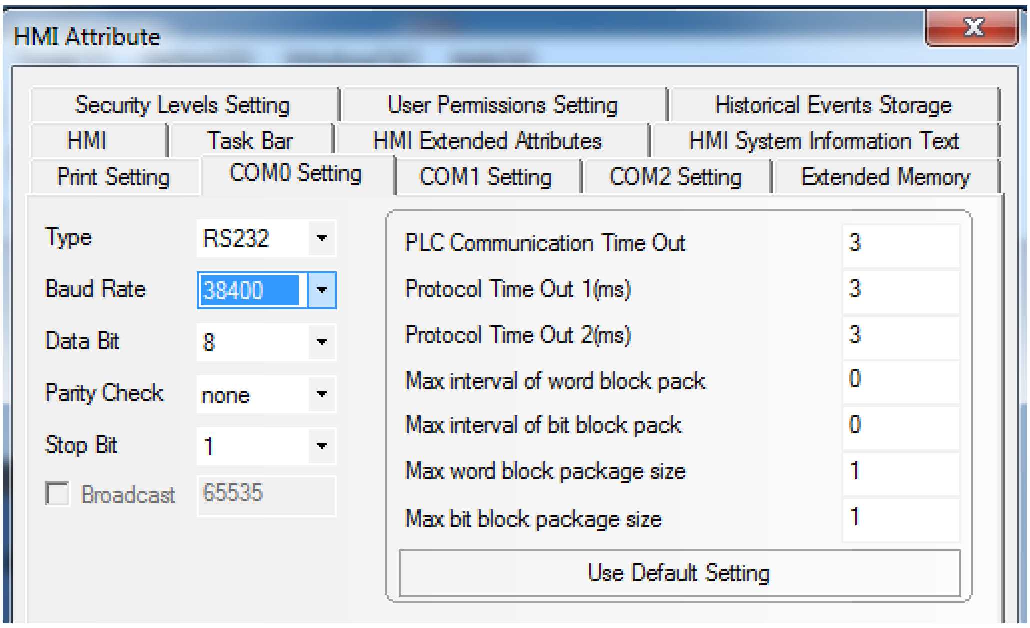
a． Wiring diagram

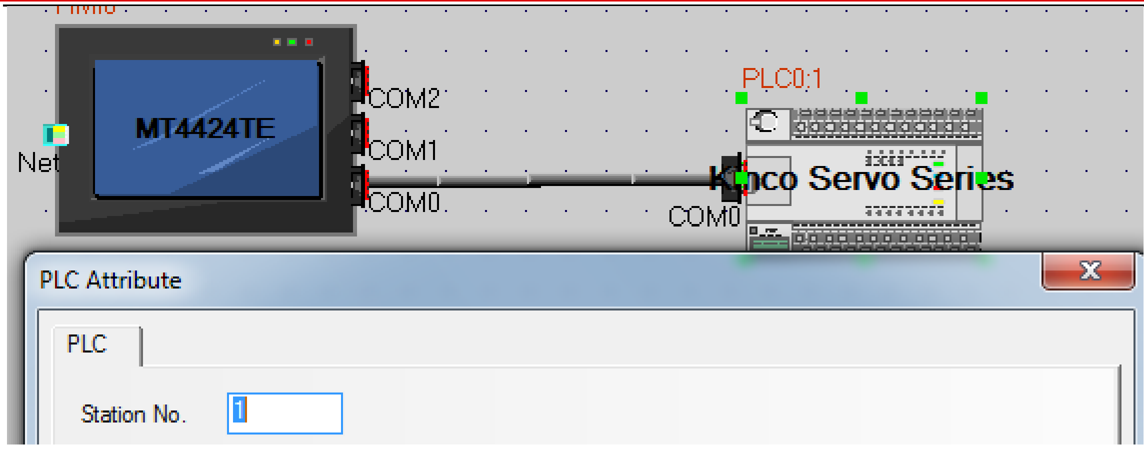


RS232 connection

b. Communication parameters setting

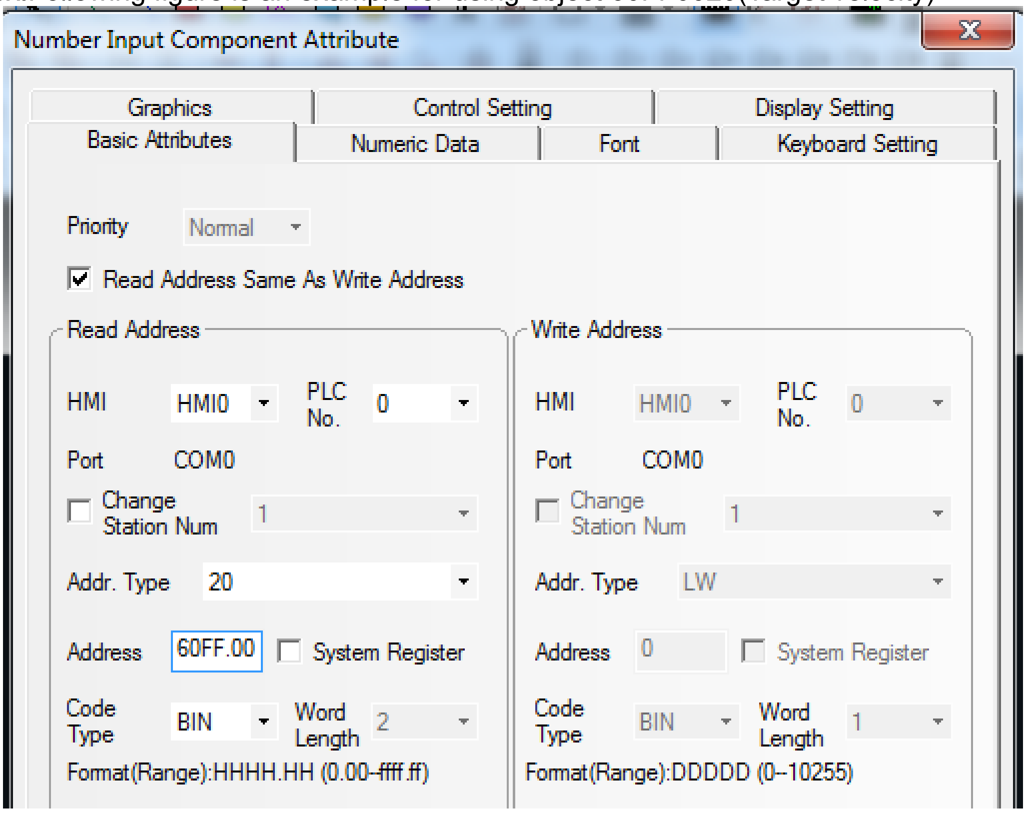
It needs to choose Kinco Servo Series driver in HMI.The parameters setting are shown in following figure.





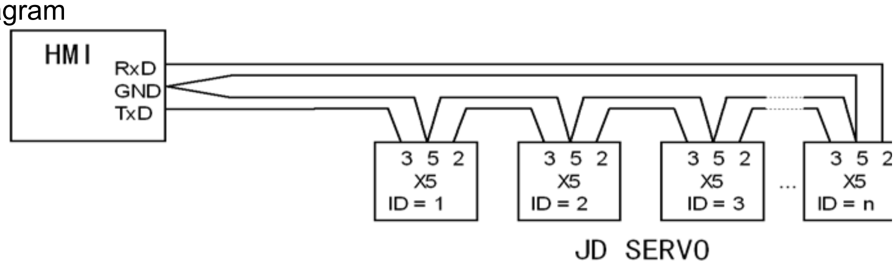
c. Address setting

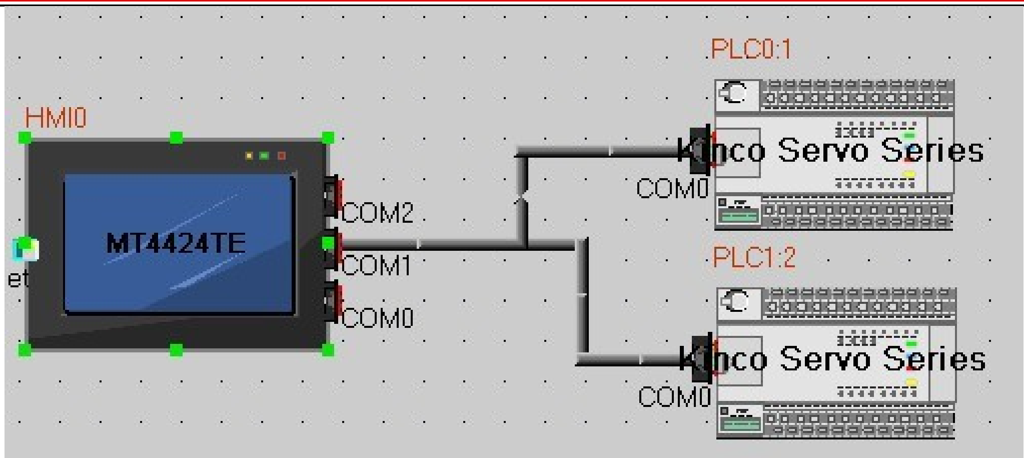
There are three address types in HMI software which are corresponding to the data length of the objects in FD2S Servo.These address types are 08(8 bits),10(16 bits) and 20 (32 bits).The format of the address is Index.Subindex.Following figure is an example for using object 60FF0020(Target velocity)



(2)HMI controls multiple FD2S Servo (D05.15 must set as 1)

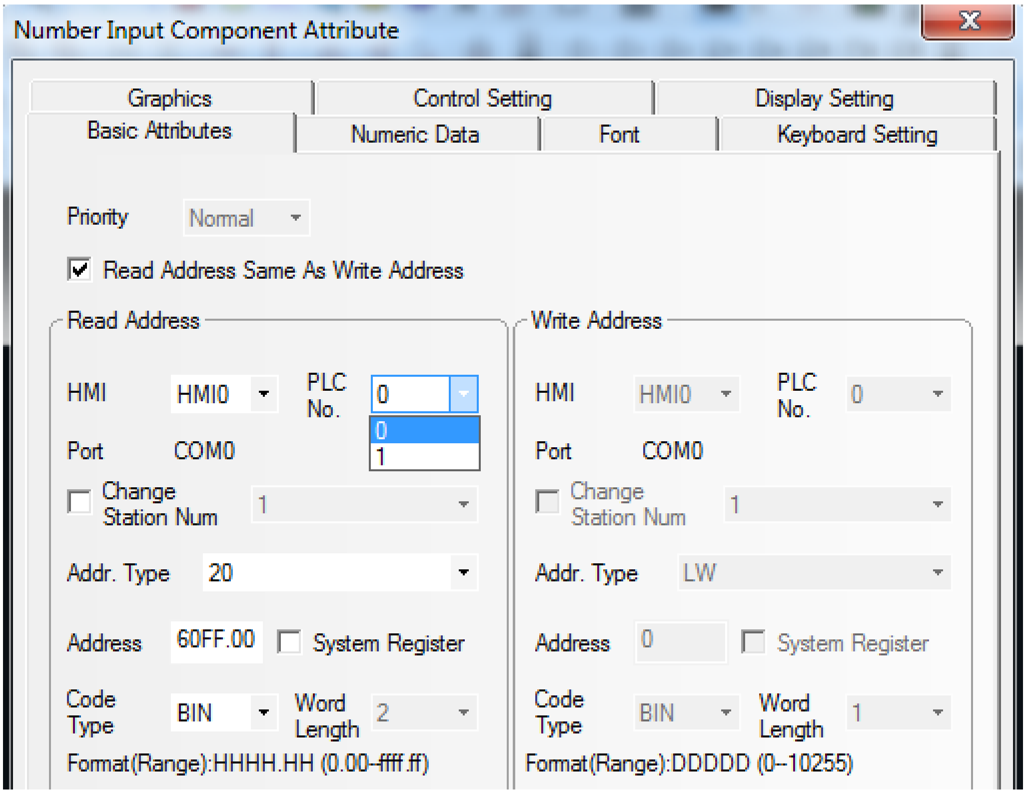
a、Wiring diagram





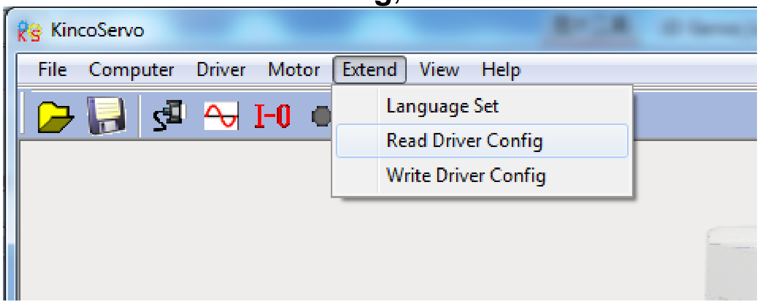
b. Parameters setting

The parameters setting in HMI is the same as above example,the difference is to set different station no. for different servo.In the attribute of components in HMI,it needs to select the PLC No. for different servo.(The PLC No. is not the servo station No.,as shown in the figure above,PLC0:1 means the PLC No. is 0,and station No. is 1)

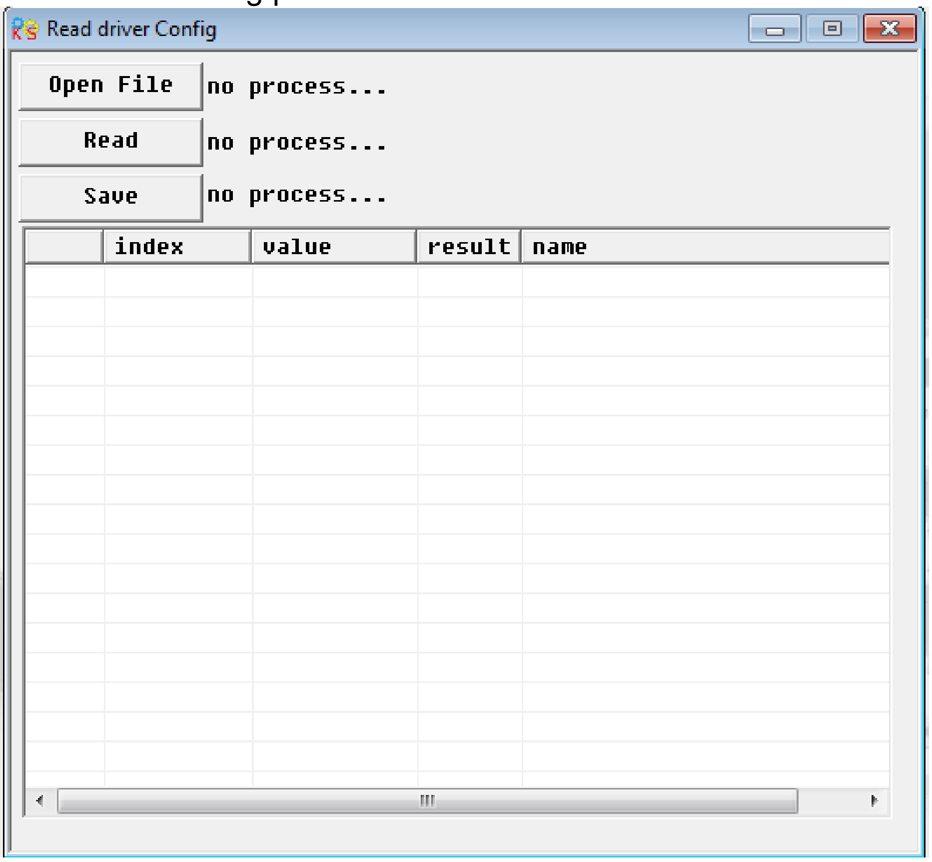


Appendix 5: Use KincoServo software to import and export driver parameters.

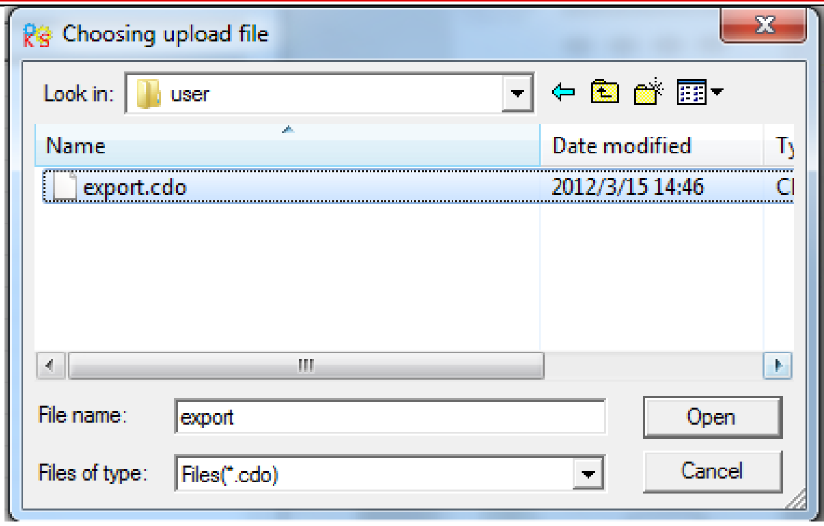
Export: It means to upload the parameters from driver and save in PC.
1.Select the **Menu->Extend->Read Driver Config**;



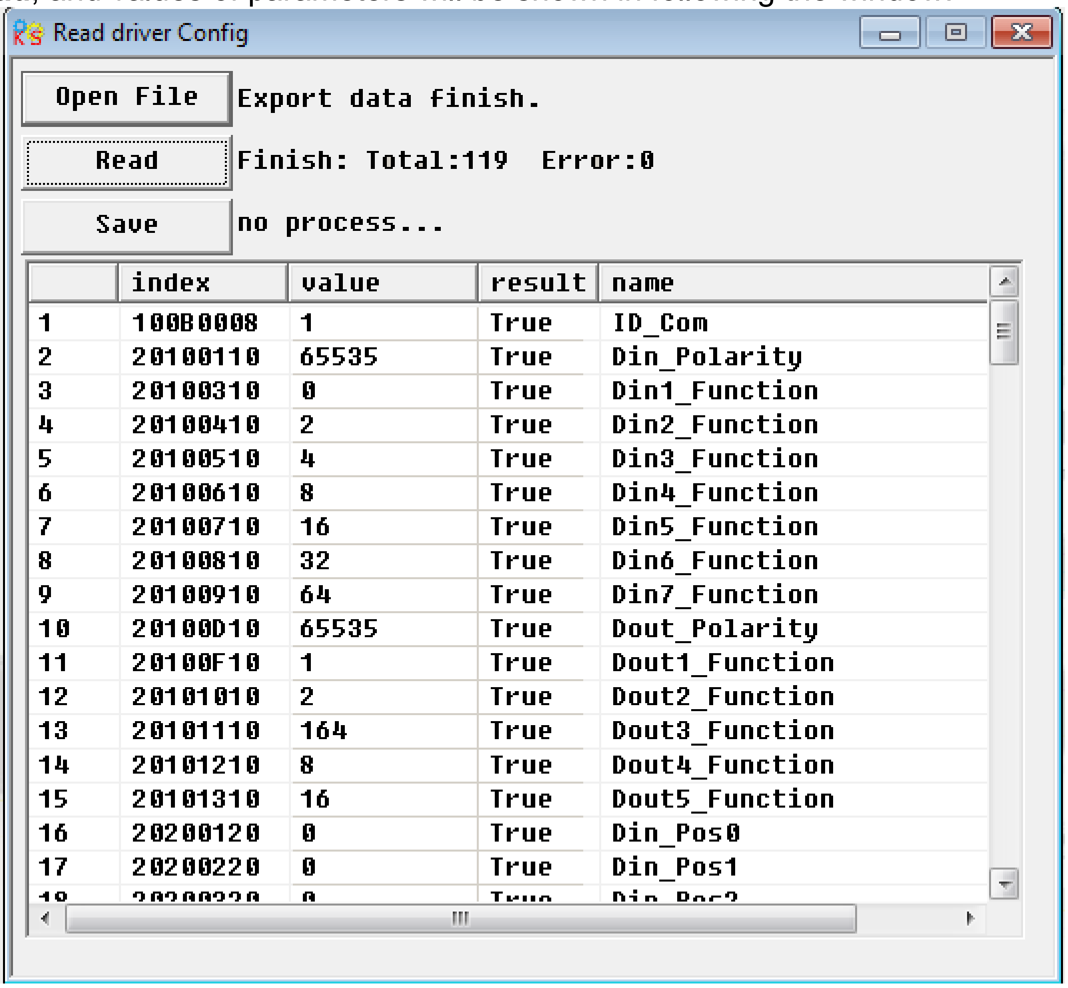
2.Open the window as the following picture:



3.Click the **Open File, pop up a dialog box like that:**

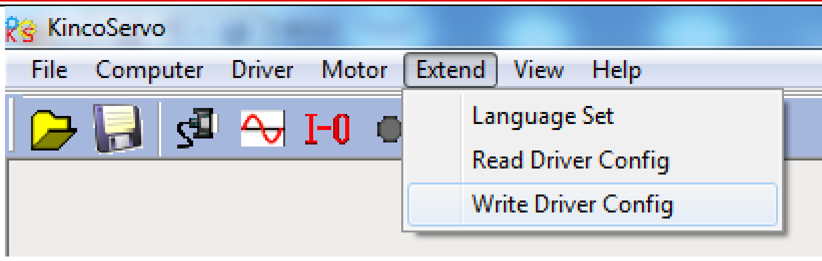


4.Select the **export.cdo,** click the **Open,** the parameters will be listed in the window, and then click the **Read**, and values of parameters will be shown in following the window:

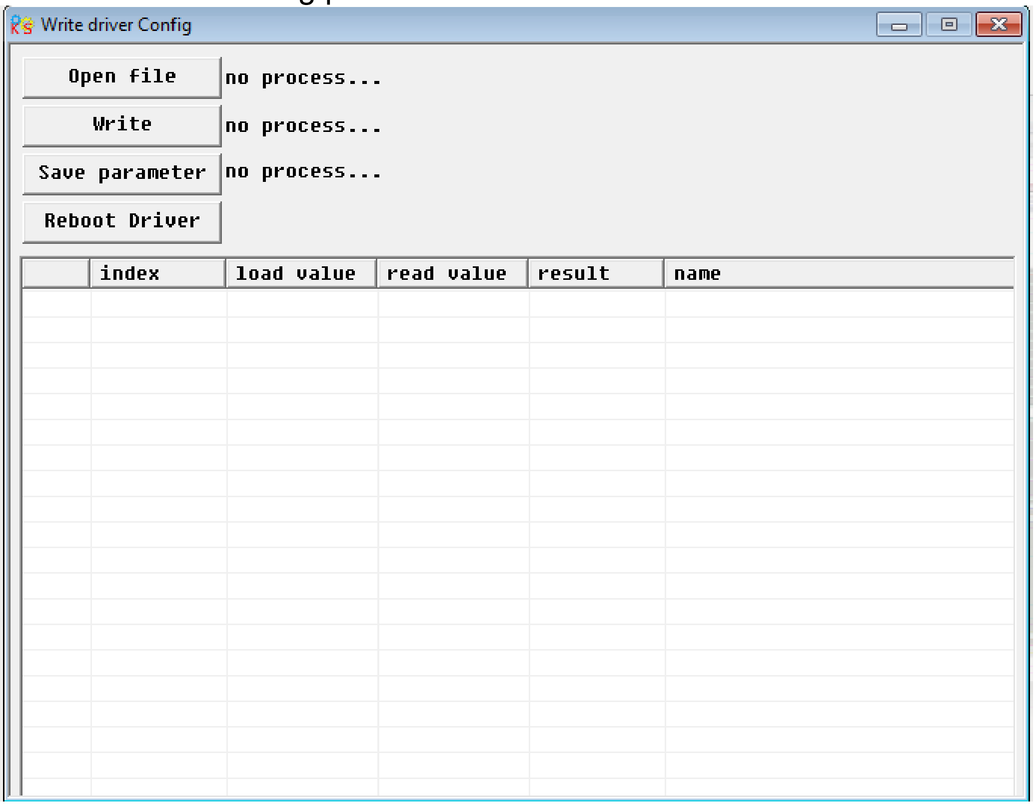


5.At last, choose the **Save,** and input the file name, so the data in driver is uploaded.

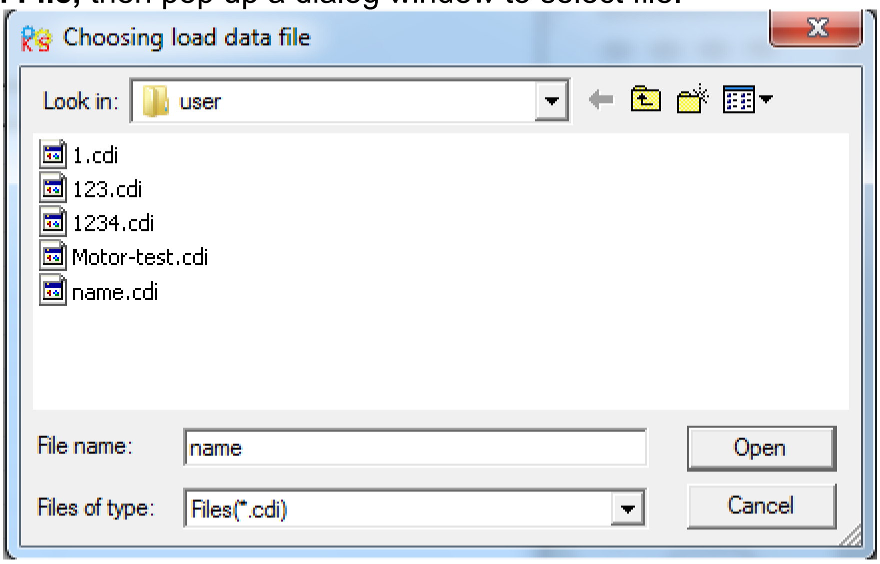
Import: It means to download the parameters into servo driver.
1.Select the **Menu->Extend->Write Driver Config**:



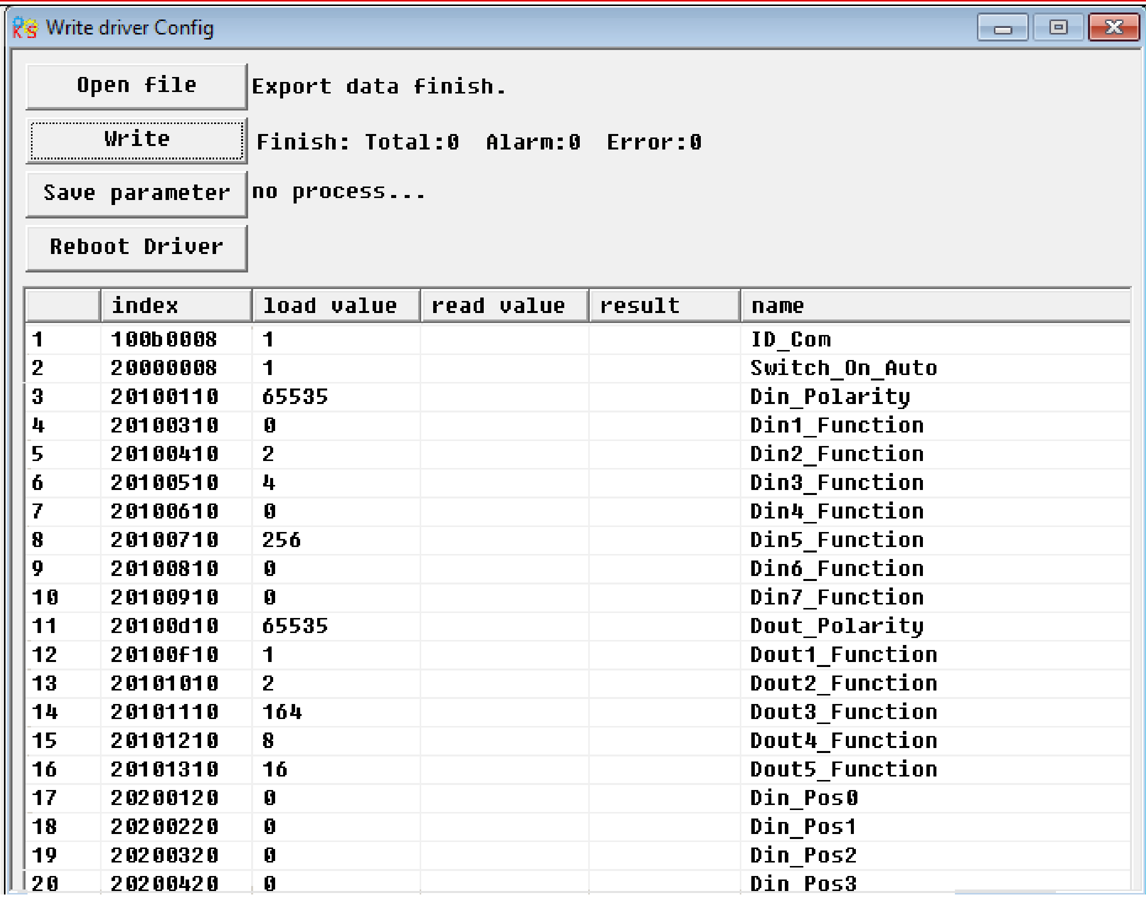
2.Open the window as the following picture.



3.Click the **Open File,** then pop up a dialog window to select file.



4.Select one of the file that needed to be download to driver. For example we choose Motor-test.cdi, Click **Open**. The parameters and their values in this file will be shown in the window:



5.Then click the **Write**, so the parameters are downloaded to driver. After that do not forget to click **Save Parameter,** then the parameters are saved in driver.

Appendix 6: Conversion between engineering unit and internal unit of common objects.

There are engineering unit and internal unit for some internal objects in FD2S Servo.When driver is controlled by communication,some objects use internal unit,therefore it needs to convert the unit.For example,the engineering unit for speed is RPM,and the internal unit is dec.Their conversion formular is 1RPM=2730dec(Resolution of encoder is 10000).Suppose to set speed as 10 RPM,then you need to send data 27300dec to the driver when using communication control.
Following table is the list of common conversion unit.

Parameter Name	Engineering Unit	Internal Unit	Conversion Fomular
Velocity	RPM	dec	dec=[(RPM*512*Encoder_resolution)/1875]
Acceleration	r/s*s	dec	dec=[(RPS/S*65536* Encoder_resolution)/4000000]
Kpp	hz	dec	1 hz= 100dec
K_Velocity_FF	%	dec	100%=256dec
Notch_N	hz	dec	Hz=dec*45+100
Speed_Fb_N	hz	dec	Hz=dec*20+100
Current	A	dec	1 Arms=1.414 Ap=105dec

Appendix 7: Common Objects List

Based on the data communication protocols described in Chapter 10,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:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
6040	00	10	0x3100	RW	bitcode	Use control word to change status of drive =>machine state 0x06 Motor power off 0x0F Motor power on 0x0B Quick stop, load tops-voltage switched off 0x2F-3F Start absolute positioning immediately 0x4F-5F Start relative positioning immediately 0x103F Start absolute positioning while target position changes. 0x105F Start relative positioning while target position changes 0x0F-1F Start homing 0X80 Clear internal error.
6041	00	10	0x3200	RO	bitcode	Status byte shows the status of drive bit0：ready to switch on bit1：switch on bit2：operation enable bit3：falt bit4：Voltage Disable bit5：Quick Stop bit6：switch on disable bit7：warning bit8：internal reserved bit9：reserved bit10： target reach bit11： internal limit active bit12： Step.Ach./V=0/Hom.att. bit13： Foll.Err/Res.Hom.Err. bit14： Commutation Found bit15： Referene Found
6060	00	08	0x3500	WO	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:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
6063	00	20	0x3700	RO	inc	Actual position value
606C	00	20	0x3b00	RO	DEC=[(RPM*512*Encoder_resolution)/1875]	Actual velocity value
6078	00	10	0x3E00	RO	number	Actual current value
60FD	00	20	0x6D00	RO	bitcode	Status words for digital inputs bit0: Negative limit signal status bit1: Positive limit signal status bit2: Home signal status bit3: Hardware lock signal status

Target object:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
607A	00	20	0x4000	RW	inc	Target position in operation mode 1, shift to demand position if control word starts motion
6081	00	20	0x4A00	RW	DEC=[(RPM*512*Encoder_resolution)/1875]	Maximum velocity of trapezium profile in mode 1
6083	00	20	0x4B00	RW	DEC=[(RPS/S*65536*Encoder_resolution)/4000000]	Acceleration of the trapezium profile Default value：610.352rps/s
6084	00	20	0x4C00	RW		Deceleration of trapezium profile Default value：610.352rps/s
60FF	00	20	0x6F00	RW	DEC=[(RPM*512*Encoder_resolution)/1875]	Target velocity in mode 3, -3, or 4
6071	00	10	0x3C00	RW	1Arms=1.414Ap=105dec	Target current
6073	00	10	0x3D00	RW		Maximum current
6080	00	20	0x4900	RW，M	RPM	Maximum velocity. Actual velocity in mode 4. Maximum velocity in other mode.

Multiple position,multiple speed.

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
2020	01	20	0x0C10	RW	DEC	Multiple position control 0
2020	02	20	0x0C20	RW	DEC	Multiple position control 1
2020	03	20	0x0C30	RW	DEC	Multiple position control 2
2020	04	20	0x0C40	RW	DEC	Multiple position control 3
2020	10	20	0x0D00	RW	DEC	Multiple position control 4

2020	11	20	0x0D10	RW	DEC	Multiple position control 5
2020	12	20	0x0D20	RW	DEC	Multiple position control 6
2020	13	20	0x0D30	RW	DEC	Multiple position control 7
2020	05	20	0x0C50	RW	RPM	Multiple speed control 0
2020	06	20	0x0C60	RW	RPM	Multiple speed control 1
2020	07	20	0x0C70	RW	RPM	Multiple speed control 2
2020	08	20	0x0C80	RW	RPM	Multiple speed control 3
2020	14	20	0x0D40	RW	RPM	Multiple speed control 4
2020	15	20	0x0D50	RW	RPM	Multiple speed control 5
2020	16	20	0x0D60	RW	RPM	Multiple speed control 6
2020	17	20	0x0D70	RW	RPM	Multiple speed control 7

Performance object

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
6065	00	20	0x3800	RW，M	inc	Maximum following error at which the drive generates an alarm Default value 10000inc
6067	00	20	0x3900	RW，M	inc	Position reach window position range for “target reached” flag Default value 10inc
607D	01	20	0x4410	RW，M	inc	Soft positive limit
607D	02	20	0x4420	RW，M	inc	Soft negative limit. (if both are zero, there is no limit)

Homing

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
6098	00	08	0x4D00	RW	integer	Homing methods
6099	01	20	0x5010	RW	DEC=[(RPM*512*Encoder_resolution)/1875]	Velocity for searching limit switch
6099	02	20	0x5020	RW		Velocity for searching phase-N signal
609A	00	20	0x5200	RW	DEC=[(RPS/S*65536*Encoder_resolution)/4000000]	Acceleration
607C	00	20	0x4100	RW	inc	Home offset

Velocity loop object:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
60F9	01	10	0x6310	RW	inc/s	VC_KP proportional gain of velocity loop 50 soft gain 200 hard gain
60F9	02	10	0x6320	RW	integer	VC_KI integral gain of velocity loop 0 no correction of transient deviations 1 default value 2 strong correction, can cause oscillation
60F9	05	10	0x6350	RW	integer	Speed feedback filter

Position loop object:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
60FB	01	10	0x6810	RW	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	10	0x6820	RW	integer	Velocity feedforward
60FB	03	10	0x6830	RW	integer	Acceleration feedforward
60FB	05	10	0x6850	RW	integer	Smooth filter

Pulse input parameters:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
2508	01	10	0x1910	RW	integer	Numerator of electronic gear ratio
2508	02	10	0x1920	RW	unsigned	Denominator of electronic gear ratio
2508	03	08	0x1930	RW	integer	Pulse mode control 0...CW/CCW 1...Pulse/Direction 2...Incremental encoder 10..CW/CCW(RS422 type) 11..Pulse/Direction(RS422 type) 12.. Incremental encoder (RS422 type) Note:0,1,2 are used for PIN4,5,9,10,14,15 of Master_Encoder interface,they are TTL signal. 10,11,12 are used for PIN6,7,8,11,12,13,they are differential signal.
2508	04	20	0x1940	RW	inc	Input pulse amount before electronic gear.
2508	05	20	0x1950	RW	inc	Execute pulse amount after electronic gear

2508	06	10	0x1960	RW	DEC	Filter for pulse input
2508	0C	10	0x19C0	RW	pulse/mS	Pulse speed of master
2508	0D	10	0x19D0	RW	pulse/mS	Pulse speed of slave

Storage parameters:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
2FF0	01	08	0x2910	RW	unsigned	1：Save all control parameters 10：Initialize all control parameters. Note ： Only for control parameters,exclude motor parameters.
2FF0	03	08	0x2930	RW	unsigned	1：Save motor parameters

Input and output parameters:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
2010	03	10	0x0830	RW	unsigned	Function definition of digital input 1
2010	04	10	0x0840	RW	unsigned	Function definition of digital input 2
2010	05	10	0x0850	RW	unsigned	Function definition of digital input 3
2010	06	10	0x0860	RW	unsigned	Function definition of digital input 4
2010	07	10	0x0870	RW	unsigned	Function definition of digital input 5
2010	08	10	0x0880	RW	unsigned	Function definition of digital input 6
2010	09	10	0x0890	RW	unsigned	Function definition of digital input 7
2010	1D	10	0x09D0	RW	unsigned	Function definition of digital input 8
2010	0F	10	0x08F0	RW	unsigned	Function definition of digital output 1
2010	10	10	0x0900	RW	unsigned	Function definition of digital output 2
2010	11	10	0x0910	RW	unsigned	Function definition of digital output 3
2010	12	10	0x0920	RW	unsigned	Function definition of digital output 4
2010	13	10	0x0930	RW	unsigned	Function definition of digital output 5
2010	1E	10	0x09E0	RW	unsigned	Function definition of digital output 6
2010	1F	10	0x09F0	RW	unsigned	Function definition of digital output 7
2010	0A	10	0x08A0	RO	bitcode	Status of digital input bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6 bit6：Din7 bit7：Din8
2010	14	10	0x0940	RO	bit code	Status of digital output bit0：Dout1 bit1：Dout2 bit2：Dout3 bit3：Dout4 bit4：Dout5 bit5：Dout6 bit6：Dout7

2010	01	10	0x0810	RW	bitcode	Polarity of digital input 0：Normally-open；1：Normally-close bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6 bit6：Din7 bit7：Din8 Default value is FF
2010	0D	10	0x08D0	RW	bitcode	Polarity of digital output 0：Normally-open；1：Normally-close bit0：Dout1 bit1：Dout2 bit2：Dout3 bit3：Dout4 bit4：Dout5 bit5：Dout6 bit6：Dout7 Default value is FF
2010	02	10	0x0820	RW	bitcode	Simulation of digital input bit0：Din1 bit1：Din2 bit2：Din3 bit3：Din4 bit4：Din5 bit5：Din6 bit6：Din7
2010	0E	10	0x08E0	RW	bitcode	Simulation of digital output bit0：Dout1 bit1：Dout2 bit2：Dout3 bit3：Dout4 bit4：Dout5 bit5：Dout6 bit6：Dout7

Error code:

Index	Subindex	Bits	Modbus Address	Command Type	Unit	Descriptions
2601	00	10	0x1F00	RO	unsigned	Current error code: bit0：Internal bit 1：Encoder ABZ bit 2：Encoder UVW bit 3：Encoder counting bit 4：Over temperature bit 5：Over voltage bit 6：Low voltage bit 7：Over current bit 8：Chop resistor bit 9：Following error

						bit 10：Logic voltage bit 11：IIt error bit 12：Over frequency bit 13：Reserved bit 14：Commutation bit 15：EEPROM
2610	00	10	/	RO	unsigne d	Error code of historical alarm 0
2611	00	10	/	RO	unsigne d	Error code of historical alarm 1
2612	00	10	/	RO	unsigne d	Error code of historical alarm 2
2613	00	10	/	RO	unsigne d	Error code of historical alarm 3
2614	00	10	/	RO	unsigne d	Error code of historical alarm 4
2615	00	10	/	RO	unsigne d	Error code of historical alarm 5
2616	00	10	/	RO	unsigne d	Error code of historical alarm 6
2617	00	10	/	RO	unsigne d	Error code of historical alarm 7

Bus specification parameters:

Index	Subindex	Bits	Command Type	Command Type	Unit
100B	00	08	RW	unsigned	Station No. of driver Default value:1 Note:it needs to save and restart driver after change.
2F81	00	08	RW	unsigned	Baudrate for CAN Setting value Baudrate 100： 1M 50： 500k 25： 250k 12： 125k 5： 50k 1： 10k Default value: 50 Note:it needs to save and restart driver after change.
2FE0	00	10	RW	unsigned	Baudrate for RS232 Setting value Baudrate 540 19200 270 38400 90 115200 Default value: 270 Note:it needs to save and restart driver after change.

2FE2	00	10	RW	unsigned	Baudrate for RS485 Setting value Baudrate 1080 9600 540 19200 270 38400 90 115200 Default value: 540 Note:it needs to save and restart driver after change.

CAN-PDO parameters：0X1400-0X1A00
0X1400-7（RX Parameter/Read）
0X1600-7（RX Mapping）
0X1800-7（TX Parameter/Write）
0X1A00-7（TX Mapping）

Below are all the internal parameters address in servo.

Index	Subindex	Bits	Descriptions	Modbus Address
0x1000	0x00	20	Device_Type	0x0400
0x100B	0x00	08	EL.ID_Com	0x0600
0x2000	0x00	08	EL.Switch_On_Auto	0x0700
0x2010	0x27	08	Group_OD_RW	0x0800
0x2010	0x01	10	EL.Din_Polarity	0x0810
0x2010	0x02	10	Din_Simulate	0x0820
0x2010	0x03	10	EL.Dinx_Function[0]	0x0830
0x2010	0x04	10	EL.Dinx_Function[1]	0x0840
0x2010	0x05	10	EL.Dinx_Function[2]	0x0850
0x2010	0x06	10	EL.Dinx_Function[3]	0x0860
0x2010	0x07	10	EL.Dinx_Function[4]	0x0870
0x2010	0x08	10	EL.Dinx_Function[5]	0x0880
0x2010	0x09	10	EL.Dinx_Function[6]	0x0890
0x2010	0x0A	10	Din_Status.All	0x08A0
0x2010	0x0B	10	Din_Virtual.All	0x08B0
0x2010	0x0C	10	Din_Sys	0x08C0
0x2010	0x0D	10	EL.Dout_Polarity	0x08D0
0x2010	0x0E	10	Dout_Simulate	0x08E0
0x2010	0x0F	10	EL.Doutx_Function[0]	0x08F0
0x2010	0x10	10	EL.Doutx_Function[1]	0x0900
0x2010	0x11	10	EL.Doutx_Function[2]	0x0910
0x2010	0x12	10	EL.Doutx_Function[3]	0x0920
0x2010	0x13	10	EL.Doutx_Function[4]	0x0930
0x2010	0x14	10	Dout_Status.All	0x0940
0x2010	0x15	10	Dout_Virtual.All	0x0950
0x2010	0x16	10	Dout_Sys.All	0x0960
0x2010	0x17	10	EL.CMD_Active_Filter	0x0970
0x2010	0x18	10	EL.Zero_Speed_Window	0x0980
0x2010	0x19	08	EL.Limit_Function	0x0990
0x2010	0x1A	10	Reserve	0x09A0
0x2010	0x1B	20	Pos_L_Pos	0x09B0
0x2010	0x1C	20	Pos_L_Neg	0x09C0
0x2010	0x1D	10	EL.Dinx_Function[7]	0x09D0

0x2010	0x1E	10	EL.Doutx_Function[5]	0x09E0
0x2010	0x1F	10	EL.Doutx_Function[6]	0x09F0
0x2010	0x20	08	Rising_Captured1	0x0A00
0x2010	0x21	08	Falling_Captured1	0x0A10
0x2010	0x22	08	Rising_Captured2	0x0A20
0x2010	0x23	08	Falling_Captured2	0x0A30
0x2010	0x24	20	Rising_Capture_Pos1	0x0A40
0x2010	0x25	20	Falling_Capture_Pos1	0x0A50
0x2010	0x26	20	Rising_Capture_Pos2	0x0A60
0x2010	0x27	20	Falling_Capture_Pos2	0x0A70
0x2020	0x1B	08	Group_OD_RW	0x0C00
0x2020	0x01	20	EL.Din_Pos[0]	0x0C10
0x2020	0x02	20	EL.Din_Pos[1]	0x0C20
0x2020	0x03	20	EL.Din_Pos[2]	0x0C30
0x2020	0x04	20	EL.Din_Pos[3]	0x0C40
0x2020	0x05	20	EL.Din_Speed[0]	0x0C50
0x2020	0x06	20	EL.Din_Speed[1]	0x0C60
0x2020	0x07	20	EL.Din_Speed[2]	0x0C70
0x2020	0x08	20	EL.Din_Speed[3]	0x0C80
0x2020	0x09	10	Group_OD_RW	0x0C90
0x2020	0x0A	10	Group_OD_RW	0x0CA0
0x2020	0x0B	10	Group_OD_RW	0x0CB0
0x2020	0x0C	10	Group_OD_RW	0x0CC0
0x2020	0x0D	08	EL.Din_Mode0	0x0CD0
0x2020	0x0E	08	EL.Din_Mode1	0x0CE0
0x2020	0x0F	10	EL.Din_Control_Word	0x0CF0
0x2020	0x10	20	EL.Din_Pos[4]	0x0D00
0x2020	0x11	20	EL.Din_Pos[5]	0x0D10
0x2020	0x12	20	EL.Din_Pos[6]	0x0D20
0x2020	0x13	20	EL.Din_Pos[7]	0x0D30
0x2020	0x14	20	EL.Din_Speed[4]	0x0D40
0x2020	0x15	20	EL.Din_Speed[5]	0x0D50
0x2020	0x16	20	EL.Din_Speed[6]	0x0D60
0x2020	0x17	20	EL.Din_Speed[7]	0x0D70
0x2020	0x18	10	Group_OD_RW	0x0D80
0x2020	0x19	10	Group_OD_RW	0x0D90
0x2020	0x1A	10	Group_OD_RW	0x0DA0
0x2020	0x1B	10	Group_OD_RW	0x0DB0
0x2030	0x00	10	EL.Index_Window	0x1000
0x2310	0x04	08	Group_OD_RW	0x1200
0x2310	0x01	20	Auto_Rev_Pos.All	0x1210
0x2310	0x02	20	Auto_Rev_Neg.All	0x1220
0x2310	0x03	08	Auto_Reverse	0x1230
0x2310	0x04	10	Stop_Time	0x1240
0x2340	0x0D	08	Group_OD_RW	0x1400
0x2340	0x01	08	EL.Step_Stop_Mode	0x1410
0x2340	0x02	10	EL.Step_Stop_Amp	0x1420
0x2340	0x03	08	EL.Encoder_Out_Select	0x1430
0x2340	0x04	10	EL.Kvp[1]	0x1440
0x2340	0x05	10	EL.Kvi[1]	0x1450
0x2340	0x06	10	EL.Kpp[1]	0x1460
0x2340	0x07	10	EL.Kvp[2]	0x1470
0x2340	0x08	10	EL.Kvi[2]	0x1480
0x2340	0x09	10	EL.Kpp[2]	0x1490
0x2340	0x0A	10	EL.Kvp[3]	0x14A0
0x2340	0x0B	10	EL.Kvi[3]	0x14B0

0x2340	0x0C	10	EL.Kpp[3]	0x14C0
0x2340	0x0D	08	EL.Keba	0x14D0
0x2502	0x10	08	Group_OD_RW	0x1600
0x2502	0x01	10	EL.Analog1_Filter	0x1610
0x2502	0x02	10	EL.Analog1_Dead	0x1620
0x2502	0x03	10	EL.Analog1_Offset	0x1630
0x2502	0x04	10	EL.Analog2_Filter	0x1640
0x2502	0x05	10	EL.Analog2_Dead	0x1650
0x2502	0x06	10	EL.Analog2_Offset	0x1660
0x2502	0x07	08	EL.Analog_Speed_Con	0x1670
0x2502	0x08	08	EL.Analog_Torque_Con	0x1680
0x2502	0x09	08	EL.Analog_MaxT_Con	0x1690
0x2502	0x0A	10	EL.Analog_Speed_Factor	0x16A0
0x2502	0x0B	10	EL.Analog_Torque_Factor	0x16B0
0x2502	0x0C	10	EL.Analog_MaxT_Factor	0x16C0
0x2502	0x0D	10	EL.Analog_Dead_High	0x16D0
0x2502	0x0E	10	EL.Analog_Dead_Low	0x16E0
0x2502	0x0F	10	Analog1_out	0x16F0
0x2502	0x10	10	Analog2_out	0x1700
0x2507	0x02	08	Group_OD_RW	0x1800
0x2507	0x01	20	Position_Offset	0x1810
0x2507	0x02	10	Velocity_Offset	0x1820
0x2508	0x0F	08	Group_OD_RW	0x1900
0x2508	0x01	10	EL.Gear_Factor[0]	0x1910
0x2508	0x02	10	EL.Gear_Divider[0]	0x1920
0x2508	0x03	08	EL.PD_CW	0x1930
0x2508	0x04	20	Gear_Master	0x1940
0x2508	0x05	20	Gear_Slave	0x1950
0x2508	0x06	10	EL.PD_Filter	0x1960
0x2508	0x07	10	Gear_Div_Error	0x1970
0x2508	0x08	10	EL.Frequency_Check	0x1980
0x2508	0x09	10	EL.PD_ReachT	0x1990
0x2508	0x0A	08	EL.Master_Capture_Enable	0x19A0
0x2508	0x0B	10	Reserve	0x19B0
0x2508	0x0C	10	Master_Speed	0x19C0
0x2508	0x0D	10	Slave_Speed	0x19D0
0x2508	0x0E	08	CPLD_Shift.All	0x19E0
0x2508	0x0F	20	Master_Capture	0x19F0
0x2509	0x0E	10	Group_OD_RW	0x1A00
0x2509	0x01	10	Gear_Factor[1]	0x1A10
0x2509	0x02	10	Gear_Divider[1]	0x1A20
0x2509	0x03	10	Gear_Factor[2]	0x1A30
0x2509	0x04	10	Gear_Divider[2]	0x1A40
0x2509	0x05	10	Gear_Factor[3]	0x1A50
0x2509	0x06	10	Gear_Divider[3]	0x1A60
0x2509	0x07	10	Gear_Factor[4]	0x1A70
0x2509	0x08	10	Gear_Divider[4]	0x1A80
0x2509	0x09	10	Gear_Factor[5]	0x1A90
0x2509	0x0A	10	Gear_Divider[5]	0x1AA0
0x2509	0x0B	10	Gear_Factor[6]	0x1AB0
0x2509	0x0C	10	Gear_Divider[6]	0x1AC0
0x2509	0x0D	10	Gear_Factor[7]	0x1AD0
0x2509	0x0E	10	Gear_Divider[7]	0x1AE0
0x250A	0x08	08	Group_OD_RW	0x1B00
0x250A	0x01	20	EL.Master_Period	0x1B10
0x250A	0x02	08	EL.Closed_Loop	0x1B20

0x250A	0x03	08	EL.Master_Direction	0x1B30
0x250A	0x04	10	EL.Closed_Error	0x1B40
0x250A	0x05	20	Reserve	0x1B50
0x250A	0x06	10	Reserve	0x1B60
0x250A	0x07	20	Pos_Abs_Master	0x1B70
0x250A	0x08	10	Master_Speed_VL	0x1B80
0x2600	0x00	10	Error_Mask	0x1C00
0x2601	0x00	10	Error_State.All	0x1F00
0x2602	0x00	10	Error_State2	0x2000
0x2605	0x06	08	Group_OD_RW	0x2200
0x2605	0x01	10	Error_Mask	0x2210
0x2605	0x02	10	EL.Store_Mask_ON	0x2220
0x2605	0x03	10	EL.Store_Mask_OFF	0x2230
0x2605	0x04	10	Error_Mask2	0x2240
0x2605	0x05	10	EL.Store_Mask_ON2	0x2250
0x2605	0x06	10	EL.Store_Mask_OFF2	0x2260
0x2F81	0x00	08	EL.CAN_Baudrate	0x2300
0x2FE0	0x00	10	EL.RS232_Bandrate	0x2400
0x2FE1	0x01	10	(word *) U2BRG	0x2500
0x2FE1	0x01	20	ED_Sim	0x2510
0x2FE2	0x00	10	EL.RS485_Bandrate	0x2600
0x2FE3	0x00	10	(word *) U1BRG	0x2700
0x2FF0	0x15	08	Group_OD_RW	0x2900
0x2FF0	0x01	08	Store_Loop_Data	0x2910
0x2FF0	0x02	08	Store_Device_Data	0x2920
0x2FF0	0x03	08	Store_Motor_Data	0x2930
0x2FF0	0x04	08	EL.Key_Address_F001	0x2940
0x2FF0	0x05	08	Group_OD_RW	0x2950
0x2FF0	0x06	10	Group_OD_RW	0x2960
0x2FF0	0x07	10	Group_OD_RW	0x2970
0x2FF0	0x08	10	Group_OD_RW	0x2980
0x2FF0	0x09	10	Group_OD_RW	0x2990
0x2FF0	0x0A	10	Group_OD_RW	0x29A0
0x2FF0	0x0B	10	Group_OD_RW	0x29B0
0x2FF0	0x0C	10	Tuning_Start	0x29C0
0x2FF0	0x0D	10	Group_OD_RW	0x29D0
0x2FF0	0x0E	10	Group_OD_RW	0x29E0
0x2FF0	0x0F	20	Soft_Version_LED	0x29F0
0x2FF0	0x10	08	Group_OD_RW	0x2A00
0x2FF0	0x11	08	Group_OD_RW	0x2A10
0x2FF0	0x12	08	Group_OD_RW	0x2A20
0x2FF0	0x13	10	No_Motor	0x2A30
0x2FF0	0x14	10	Real_Speed_RPM	0x2A40
0x2FF0	0x15	08	Store_Resolver	0x2A40
0x2FF0	0x15	08	Reserve	0x2A40
0x2FF7	0x00	20	Time_Driver	0x2D00
0x2FFD	0x00	10	User_Secret	0x2E00
0x2FFF	0x00	10	Bootloader	0x2F00
0x6004	0x00	20	Pos_Abs	0x3000
0x6040	0x00	10	Control_Word	0x3100
0x6041	0x00	10	Status_Word.All	0x3200
0x605A	0x00	10	EL.Quick_Stop_Mode	0x3400
0x605B	0x00	10	EL.Shutdown_Stop_Mode	0x3410
0x605C	0x00	10	EL.Disable_Stop_Mode	0x3420
0x605D	0x00	10	EL.Halt_Mode	0x3430
0x605E	0x00	10	EL.Fault_Stop_Mode	0x3440

0x6060	0x00	08	Operation_Mode	0x3500
0x6061	0x00	08	Operation_Mode_Buff2	0x3600
0x6063	0x00	20	Pos_Actual	0x3700
0x6064	0x00	20	Pos_Actual	0x3710
0x6065	0x00	20	EL.Max_Following_Error	0x3800
0x6067	0x00	20	EL.Target_Pos_Window	0x3900
0x606B	0x00	20	Speed_Demand_Buff	0x3A00
0x606C	0x00	20	Speed_Real_Filter	0x3B00
0x6071	0x00	10	Group_OD_RW	0x3C00
0x6071	0x00	10	CMD_q	0x3C00
0x6073	0x00	10	Max_Current	0x3D00
0x6073	0x00	10	EL.CMD_q_Max	0x3D00
0x6078	0x00	10	Actual_Current	0x3E00
0x6078	0x00	10	I_q_b	0x3E00
0x607A	0x00	20	Pos_Target	0x4000
0x607C	0x00	20	EL.Home_Offset	0x4100
0x607D	0x02	08	Group_OD_RW	0x4400
0x607D	0x01	20	EL.Soft_Positive_Limit	0x4410
0x607D	0x02	20	EL.Soft_Negative_Limit	0x4420
0x607E	0x00	08	EL.Invert_Dir	0x4700
0x607F	0x00	20	EL.Max_Speed	0x4800
0x6080	0x00	10	Group_OD_RW	0x4900
0x6081	0x00	20	Profile_Speed	0x4A00
0x6083	0x00	20	EL.Profile_Acce	0x4B00
0x6084	0x00	20	EL.Profile_Dece	0x4C00
0x6085	0x00	20	EL.Quick_Stop_Dece	0x3300
0x6098	0x00	08	EL.Homing_Method	0x4D00
0x6099	0x05	08	Group_OD_RW	0x5000
0x6099	0x01	20	EL.Homing_Speed_Switch	0x5010
0x6099	0x02	20	EL.Homing_Speed_Zero	0x5020
0x6099	0x03	08	EL.Homing_Power_On	0x5030
0x6099	0x04	10	EL.Homing_Current	0x5040
0x6099	0x05	08	EL.Home_Offset_Mode	0x5050
0x6099	0x06	20	Speed_Pos_Average	0x5060
0x6099	0x07	20	Speed_Demand_Diff	0x5070
0x6099	0x08	10	Pos_Filter_Err1	0x5080
0x6099	0x09	20	Pos_Filter_Out_Err	0x5090
0x6099	0x0A	20	Profile_Dece_Buff	0x50A0
0x609A	0x00	20	EL.Homing_Accelaration	0x5200
0x60F4	0x00	20	Pos_Error	0x5500
0x60F5	0x07	10	Group_OD_RW	0x5600
0x60F5	0x01	10	Kci_d	0x5610
0x60F5	0x02	20	PID_Limit_q	0x5620
0x60F5	0x03	20	PID_Limit_d	0x5630
0x60F5	0x04	10	EL.Kap	0x5640
0x60F5	0x05	10	EL.Kad	0x5650
0x60F5	0x06	10	EL.User_IIt_I	0x5660
0x60F5	0x07	10	EL.User_IIt_Filter	0x5670
0x60F6	0x27	08	Group_OD_RW	0x5800
0x60F6	0x01	10	EM.Kcp	0x5810
0x60F6	0x02	10	EM.Kci	0x5820
0x60F6	0x03	10	EL.Speed_Limit_Factor	0x5830
0x60F6	0x04	10	EM.N_Compensation	0x5840
0x60F6	0x05	10	EM.N_bEMF	0x5850
0x60F6	0x06	10	Comm_Shift_UVW	0x5860
0x60F6	0x07	10	Voltage_Angle_Adjust	0x5870

0x60F6	0x08	10	CMD_q	0x5880
0x60F6	0x09	10	CMD_d	0x5890
0x60F6	0x0A	08	SVPWM	0x58A0
0x60F6	0x0B	10	K_DC	0x58B0
0x60F6	0x0C	10	CMD_q_Buff_Filter	0x58C0
0x60F6	0x0D	10	CMD_d_Buff	0x58D0
0x60F6	0x0E	10	CMD_q_Max_Buff	0x58E0
0x60F6	0x0F	10	CMD_q_Limit	0x58F0
0x60F6	0x10	10	Driver_IIt_Real	0x5900
0x60F6	0x11	10	Driver_IIt_Max	0x5910
0x60F6	0x12	10	Motor_IIt_Real	0x5920
0x60F6	0x13	10	Motor_IIt_Max	0x5930
0x60F6	0x14	10	I_a	0x5940
0x60F6	0x15	10	I_b	0x5950
0x60F6	0x16	10	Angle	0x5960
0x60F6	0x17	10	I_q	0x5970
0x60F6	0x18	10	I_d_b	0x5980
0x60F6	0x19	20	PID_q_Sum	0x5990
0x60F6	0x1A	20	PID_d_Sum	0x59A0
0x60F6	0x1B	20	PID_q_Out	0x59B0
0x60F6	0x1C	20	PID_d_Out	0x59C0
0x60F6	0x1D	10	PID_q_Int	0x59D0
0x60F6	0x1E	10	PID_d_Int	0x59E0
0x60F6	0x1F	10	U_a	0x59F0
0x60F6	0x20	10	U_b	0x5A00
0x60F6	0x21	10	U_q	0x5A10
0x60F6	0x22	10	(word *) PDC1	0x5A20
0x60F6	0x23	10	(word *) PDC2	0x5A30
0x60F6	0x24	10	(word *) PDC3	0x5A40
0x60F6	0x25	10	Angle_B	0x5A50
0x60F6	0x26	10	User_IIt_Real	0x5A60
0x60F6	0x27	10	Z_Capture_Angle	0x5A70
0x60F7	0x12	08	Group_OD_RW	0x6000
0x60F7	0x01	10	EL.Chop_Resistor	0x6010
0x60F7	0x02	10	EL.Chop_Power_Rated	0x6020
0x60F7	0x03	10	EL.Chop_Filter	0x6030
0x60F7	0x04	10	EL2.Ripple_DCBUS_Filter	0x6040
0x60F7	0x05	10	EL2.RELAY_Time	0x6050
0x60F7	0x06	08	Reserve	0x6060
0x60F7	0x07	10	Reserve	0x6070
0x60F7	0x08	10	EL2.Temp_Device_Offset	0x6080
0x60F7	0x09	10	(word *) DTCON1	0x6090
0x60F7	0x0A	08	EL.Frequency_Switch_Enable	0x60A0
0x60F7	0x0B	10	Temp_Device	0x60B0
0x60F7	0x0C	10	Ripple_DCBUS	0x60C0
0x60F7	0x0D	10	Chop_Power_Real	0x60D0
0x60F7	0x0E	10	Reserve	0x60E0
0x60F7	0x0F	20	PWM_Time_Current	0x60F0
0x60F7	0x10	20	PWM_Time_Last	0x6100
0x60F7	0x11	10	STO_Status	0x6110
0x60F7	0x12	10	Real_DCBUS	0x6120
0x60F9	0x2B	08	Group_OD_RW	0x6300
0x60F9	0x01	10	EL.Kvp[0]	0x6310
0x60F9	0x02	10	EL.Kvi[0]	0x6320
0x60F9	0x03	08	EL.Notch_N	0x6330
0x60F9	0x04	08	EL.Notch_On	0x6340

0x60F9	0x05	08	EL.Speed_Fb_N	0x6350
0x60F9	0x06	08	EL.Speed_Mode	0x6360
0x60F9	0x07	10	EL.Kvi_T32	0x6370
0x60F9	0x08	20	EL.Kvi_Sum_Limit	0x6380
0x60F9	0x09	08	EL.PI_Switch	0x6390
0x60F9	0x0A	20	EL.Target_Speed_Window	0x63A0
0x60F9	0x0B	10	EL.Kd_Virtual	0x63B0
0x60F9	0x0C	10	EL.Kp_Virtual	0x63C0
0x60F9	0x0D	10	EL.Ki_Virtual	0x63D0
0x60F9	0x0E	10	EL.K_Load	0x63E0
0x60F9	0x0F	10	Sine_Frenquency_Adj	0x63F0
0x60F9	0x10	10	EL.Sine_Amplitude	0x6400
0x60F9	0x11	10	EL.Tuning_Scale	0x6410
0x60F9	0x12	10	EL.Tuning_Filter	0x6420
0x60F9	0x13	10	Tuning_Time	0x6430
0x60F9	0x14	10	EL.Zero_Speed_Time	0x6440
0x60F9	0x15	08	EL.Output_Filter_N	0x6450
0x60F9	0x16	10	Speed_QEI_Back	0x6460
0x60F9	0x17	20	Speed_Fb_Out1	0x6470
0x60F9	0x18	10	Real_Speed_RPM	0x6480
0x60F9	0x19	10	Real_Speed_RPM2	0x6490
0x60F9	0x1A	10	Speed_1mS	0x64A0
0x60F9	0x1B	20	Speed_Real_Filter	0x64B0
0x60F9	0x1C	20	Speed_Error	0x64C0
0x60F9	0x1D	10	Speed_Err_Err	0x64D0
0x60F9	0x1E	20	Speed_Curr_Out	0x64E0
0x60F9	0x1F	20	Speed_Curr_Sum	0x64F0
0x60F9	0x20	10	CMD_q_PID	0x6500
0x60F9	0x21	20	PID_Virtual	0x6510
0x60F9	0x22	20	Speed_Virtual	0x6520
0x60F9	0x23	10	Error1_Virtual	0x6530
0x60F9	0x24	10	Tuning_Input	0x6540
0x60F9	0x25	10	Tuning_Sine	0x6550
0x60F9	0x26	20	Tuning_Sum	0x6560
0x60F9	0x27	10	Tuning_Time_Count	0x6570
0x60F9	0x28	08	PI_Point	0x6580
0x60F9	0x29	20	Speed_Demand_Filter	0x6590
0x60F9	0x2A	10	EL.K_Load_N	0x65A0
0x60F9	0x2B	20	EL.Quick_Stop_Dece2	0x65B0
0x60FB	0x0E	08	Group_OD_RW	0x6800
0x60FB	0x01	10	EL.Kpp[0]	0x6810
0x60FB	0x02	10	EL.K_Velocity_FF	0x6820
0x60FB	0x03	10	EL.K_Acc_FF	0x6830
0x60FB	0x04	08	EL.Pos_Speed_Filter	0x6840
0x60FB	0x05	10	EL.Pos_Filter_N	0x6850
0x60FB	0x06	08	EL.Store_Position	0x6860
0x60FB	0x07	20	EL.Pos_Shift	0x6870
0x60FB	0x07	10	Reserve	0x6870
0x60FB	0x08	20	Pos_Error	0x6880
0x60FB	0x09	20	Speed_Calculat_Buff	0x6890
0x60FB	0x0A	20	Speed_Demand_Pos	0x68A0
0x60FB	0x0B	20	Profile_Speed_Buff	0x68B0
0x60FB	0x0C	10	Acc_Feedforward	0x68C0
0x60FB	0x0D	20	Pos_Filter_Out	0x68D0
0x60FB	0x0E	20	Pos_Target_Profile	0x68E0
0x60FC	0x00	20	Pos_Demand	0x6C00

0x60FD	0x00	10	Din_Status.All	0x6D00
0x60FD	0x00	20	Digital_Inputs	0x6D00
0x60FE	0x01	08	Group_OD_RW	0x6E00
0x60FE	0x01	20	Digital_Outputs	0x6E10
0x60FF	0x00	20	Speed_Demand	0x6F00
0x6410	0x1A	08	Group_OD_RW	0x7000
0x6410	0x01	10	EM.Motor_Num	0x7010
0x6410	0x02	08	EM.Feedback_Type	0x7020
0x6410	0x03	20	EM.Feedback_Resolution	0x7030
0x6410	0x04	20	EM.Feedback_Period	0x7040
0x6410	0x05	08	EM.Motor_Poles	0x7050
0x6410	0x06	08	EM.Commu_Mode	0x7060
0x6410	0x07	10	EM.Commu_Curr	0x7070
0x6410	0x08	10	EM.Commu_Delay	0x7080
0x6410	0x09	10	EM.Motor_IIt_I	0x7090
0x6410	0x0A	10	EM.Motor_IIt_Filter	0x70A0
0x6410	0x0B	10	EM.Imax_Motor	0x70B0
0x6410	0x0C	10	EM.L_Motor	0x70C0
0x6410	0x0D	08	EM.R_Motor	0x70D0
0x6410	0x0E	10	EM.Ke_Motor	0x70E0
0x6410	0x0F	10	EM.Kt_Motor	0x70F0
0x6410	0x10	10	EM.Jr_Motor	0x7100
0x6410	0x11	10	EM.Brake_Duty_Cycle	0x7110
0x6410	0x12	10	EM.Brake_Delay	0x7120
0x6410	0x13	08	EM.Invert_Dir_Motor	0x7130
0x6410	0x14	10	EM.Motor_Num	0x7140
0x6410	0x15	10	EM.Motor_BW	0x7150
0x6410	0x16	10	Motor_Using	0x7160
0x6410	0x17	08	EM.Motor_With_Brake	0x7170
0x6410	0x18	10	EM.Temp_Motor_Ref	0x7180
0x6410	0x19	10	Temp_Motor	0x7190
0x6410	0x1A	10	EM.Gain_Factor	0x71A0

Appendix 8: Selection for Brake Resistor

Driver Model	Driver Power[W]	Brake Resistor[Ω]			Brake Resistor Model(Ref.)	Brake Resistor Power[W]（Ref.）	Brake Resistor Withstand Voltage[VDC]（Min.）
		Min.	Max.	Ref.			
FD422S-AA-000	200W	39	100	75	T-75R-100	100	500
	400W						
	750W						
FD432S-AA-000	1.0KW	27	51	39	T-39R-200	200	
	1.05KW						
	1.26KW						
FD622S-AA-000	1.26KW	47	150	75	T-75R-200		800
	1.57KW						
	1.88KW						
	2.1kw						
	2.3kw						

Note:Please set brake resistor value and power in d5.04 and d5.05 when using brake resistor. Please select brake resistor power according to real application.

Appendix 9: Selection for Fuse

Driver Model	Driver Power[W]	Specification
FD422S-AA-000	200W	3.5A/250VAC
	400W	7A/250VAC
	750W	15A/250VAC
FD432S-AA-000	1000W	20A/250VAC
	1.05KW	20A/250VAC
	1.26KW	25A/250VAC
FD622S-AA-000	1.26KW	15A/500VAC
	1.57KW	
	1.88KW	20A/500VAC
	2.1KW	25A/250VAC
	2.3KW	