XINJE

X-NET fieldbus

User manual

Wuxi XINJE Electric Co., Ltd.

Data No. PD06 20151009 3.2

Catalog

Preface X-NET introduction

X-NET protocol stack which is developed by XINJE Company committed to the automation production network. X-NET supports different physical media and networks. X_NET refers to OSI seven layers network protocol, it defines the service and function of each layer and make the network more efficient and simple.

The physical protocol supported by XD/XG/XL series PLC including RS485, RS232, OC (optical circle), USB_PHY. The network protocol includes TBN (Token Bus Net), OMMS (one master multi slave net). TBN is fit for embedded equipment network, the advantage includes real-time and multi-master. OMMS is fit for master-slave network, the advantage is wide applicability, but the response time is longer than TBN.

	TBN	OMMS
RS485	√	√
RS232		√
OC	√	√
USB_PHY		√

TBN belongs to industry fieldbus. It is token structure, any node in the network has access rights (token), any node who got the token can send message to other node. It doesn’t have master and slave. It makes the data access efficient and flexible.

OMMS focus on the connection between upper device and embedded equipment. The connection between equipments is permitted. The PLC connection with programming software and network configuration software uses OMMS protocol.

The MCP (motion control protocol) based on OMMS makes the multi-axis controlling simple. 8-axis 2ms control period can meet most of motion control needs. It needs one PLC and 8 servo drives to make the project, no need special communication module.

As the important part of X-NET protocol stack, Ethernet makes the fast data interaction between embedded devices and PC and remote data access across the Internet easier and faster. The Ethernet project used by XINJE company supports adaptive baud rate 10M/100Mbps, built-in complete TCP/IP protocol stack, DHCP service, DNS service.

Besides, the reliable UDP communication protocol developped by our company based on UDP protocol ensures the data integrity, effectiveness and fast interaction. Monitoring program in XDPPro software via Ethernet is faster than USB communication mode. Ethernet supports connecting directly or connecting via switchboard, which can meet different application requirements. Up to 30 PLCs based on Ethernet can connect together, the network range can extend to hundreds of meters when using switchboard.

X-NET includes X-NET fieldbus, X-NET motion fieldbus and Ethernet communication.

X-NET fieldbus is applied to XD series PLC and TG/TN series HMI.

X-NET motion fieldbus is applied to XD series PLC and DS3E series servo drive.

Ethernet communication is applied to PLC remote control, LAN monitoring, online programming, program upload and download, device communication.

1 X-NET fieldbus

1-1. Function summarize

1-1-1. Introduction

X-NET fieldbus is used between XD/XG/XL series PLC or XD/XG/XL series PLC and TG/TN series HMI.

The filebus has the advantages of intelligence and digitize, the max speed can up to 3M. It can work through RS485 and OC (optical fiber) for long distance and high speed communication. It has faster speed and better stability than Modbus protocol.

1-1-2. Wiring mode

Please use serial connection for TBN protocol running on RS485. As the star connection has low driving ability and short communication distance, strong signal reflection, higher communication error rate. It is not recommended to use star connection.

When TBN is running on OC, the send and receive of the two adjacent stations is connected by optical fiber. The send of the last station connects to the receive of the first station to form the ring network. It is widely used as the good anti-interference capability of optical signal. But the disadvantages are that it is complicated to make the tail fiber and the optical fiber is easy to break.

TBN_OC supports baud rate range 600bit~3Mbit.

Considering the driving capability of RS485 devices and the conversion speed of photoelectric module, the nodes cannot over 32 in one TBN network.

TBN_RS485 baud rate range is 600bit~3Mbit(up to 3M by using optical fiber, up to 1M by using shielded twisted pair). The communication speed and medium are related to the local environment.

As the TBN_RS485 using electrical signal to transfer data, there are some requirements for the communication distance. Please refer to the following table.

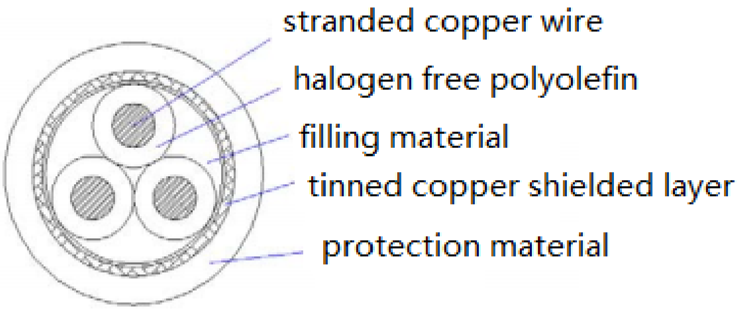
The relationship between TBN_RS485 filedbus cable length and baud rate:

Baud rate	Wire length
9.6-187.5Kbit/s	1000m
500Kbit/s	400m
1.5Mbit/s	200m
3Mbit/s	100m

TBN_RS485 cable is recommended to use shielded twisted pair cable (ASTP-120Ω for RS485&CAN, 21AWG, red blue twisted pair), according to the standard UL758-2008 &UL1581.

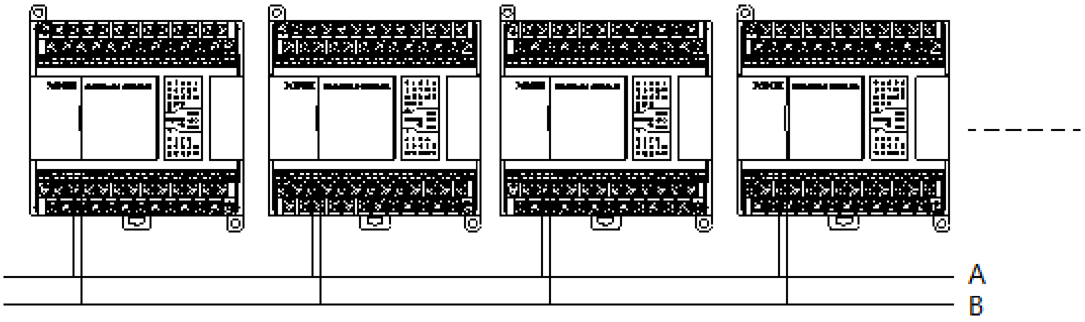
The shileded layers include Aluminium foil 100% and tinned copper net 70%.

The following diagram is the cable profile:



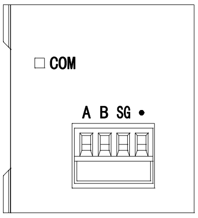
There are three kinds of connection methods for PLC.
1. PLC RS485 port on the main unit
2. PLC expansion BD board XD-NE-BD
3. PLC expansion ED board

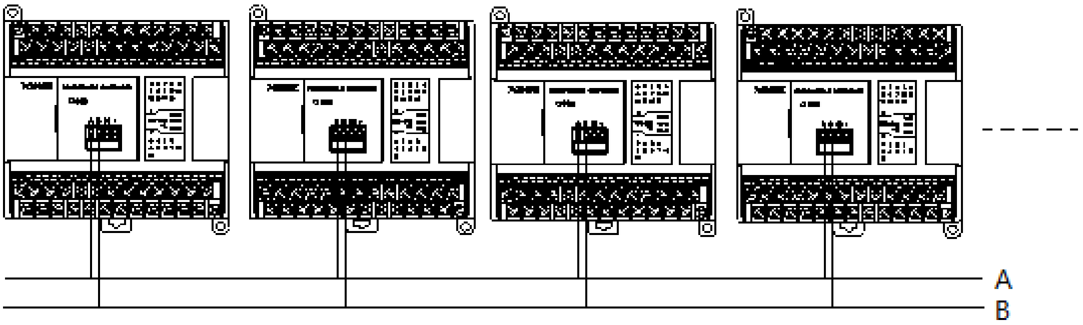
Connection method 1:
Terminal A is RS485+, terminal B is RS485-. Connnect A to A, B to B.



Note: RS485 port can communication through Modbus_RTU or X-NET. Please choose the mode in XINJEConfig software. As the communication effect of RS485 port on the PLC is not better than BD board, it is not recommended to use PLC RS485 port for X-NET communication.

Connection method 2:
There are four terminals including A, B, SG(signal ground), FG(shielded ground) on expansion BD board XD-RS485-BD. Normally, X-NET needs to connect terminal A and B, the shielded layer connects to FG. If there is requirements for anti-interference, such as motion control fieldbus, it needs to connect terminal A, B, SG.
XD-RS485-BD has switch to set terminal resistor. The default setting is switch OFF(left) without terminal resistor. If XD-RS485-BD is at the head or end of the filedbus, it needs to add 120ohm terminal resistor at both end, the switch must be ON (right).



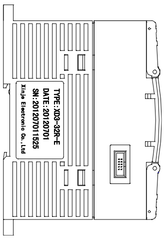


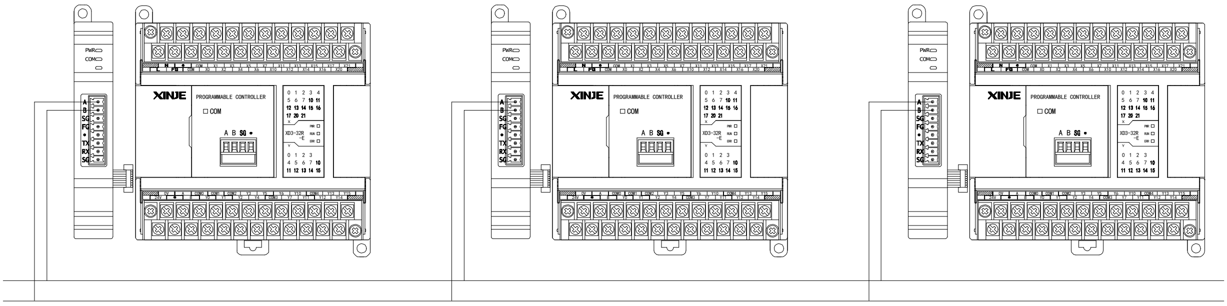
Note: RS485 port of XD-NE-BD can communication through Modbus_RTU or X-NET. Please choose the mode in XINJEConfig software.

Connection method 3:

Connect through the expansion ED board on the left side of PLC.

There are 7 terminals on the board including A, B, SG(signal ground), FG(shielded ground), TX, RX, SG. The last three terminals are for RS232 communication. Here we use fieldbus communication so terminal A and B are used. Usually, it needs to connect A, B and FG for fieldbus connection. If there is high requirements for anti-interference, please use three-wire 485 connection A, B and SG.





Note: the RS485 port of XD-NES-ED supports both Modbus-RTU and X-NET communication. The communication method can be set through XINJEConfig tool.

XG/XL series PLC only have two serial ports. XG series PLC from up to down are COM1, COM2. XL series PLC black round port is COM1, the top terminal is COM2, the serial port terminal from up to down are SG, B, A. The wiring method is serial connection A to A, B to B. If there is high requirements for anti-interference, please use three-wire 485 connection A, B and SG. The two serial ports all supports Modbus-RTU and X-NET communication. The communication method can be set through XINJEConfig tool.

1-1-3. XINJEConfig software

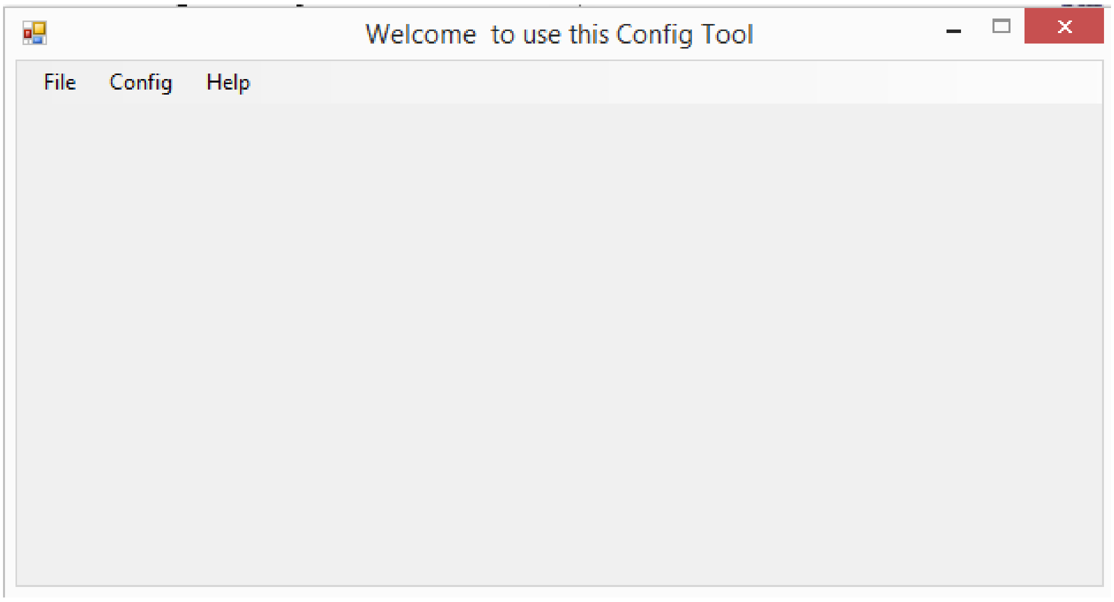
The using steps of XINJEConfig:

For example, two XD3-32T-E communicate with each other through RS485 port based on X-NET protocol.

Note: Please connect PLC with PC through USB cable before matching the PLC by software. Make sure to install the USB driver before using.

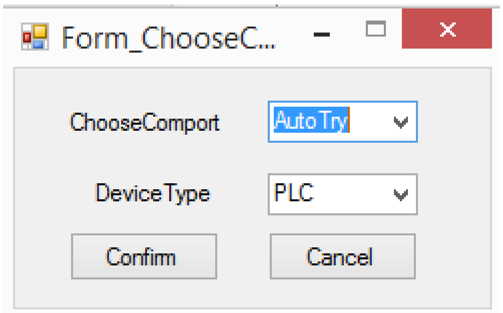
USB cable

Open the XINJEConfig software, click config/find device.



Welcome to use this Config Tool

Choose the com port connected PLC, device type is PLC.

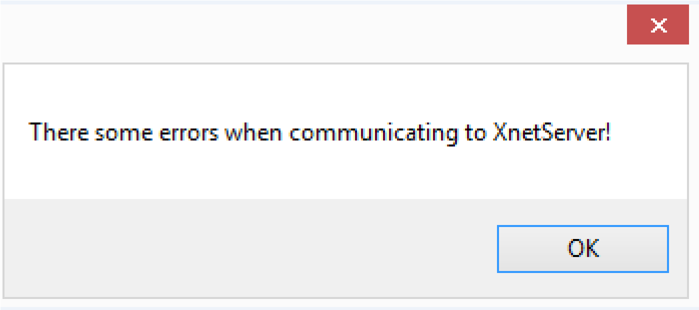


Form_ChooseC...

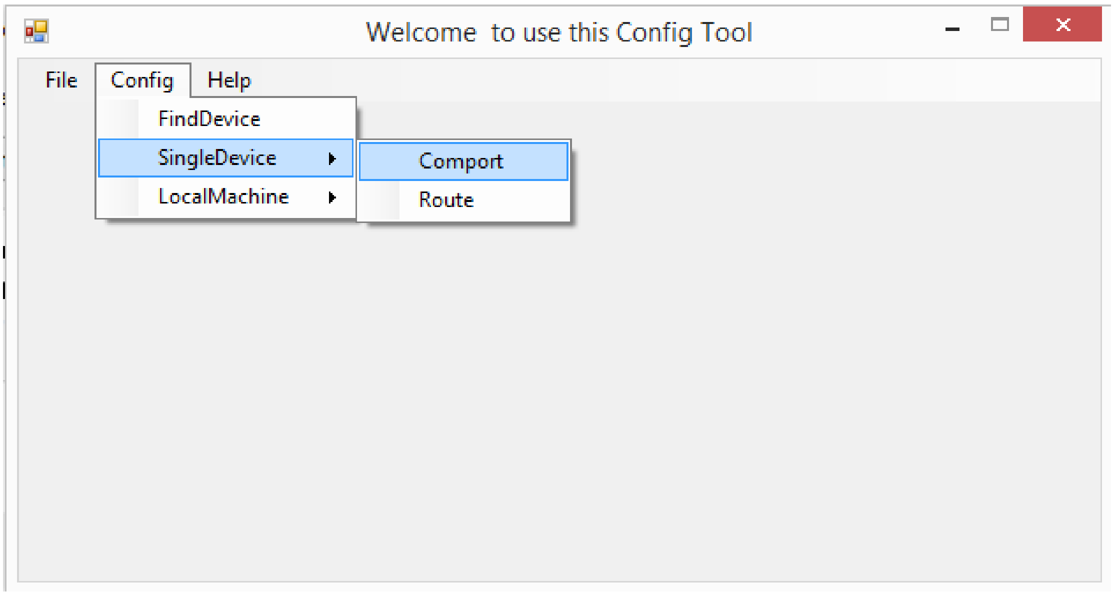
ChooseComport: Auto Try

DeviceType: PLC

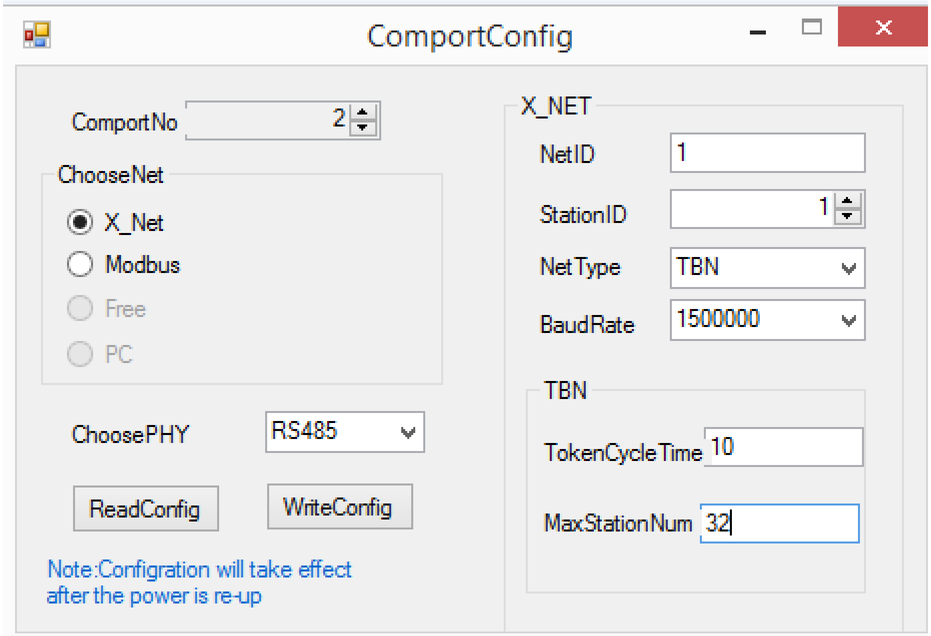
If the communication between PLC and PC is error, there will show below error. Please restart the software and configure again.


There some errors when communicating to XnetServer!

Click ok to back to main window. Then click config/single deivce/com port.



As the PLC RS485 port is serial port2, here we choose no.2 for comportNo. The protocol is X-NET. The physical layer is RS485.


ComportConfig

ComportNo 2

ChooseNet
● X_Net
○ Modbus
○ Free
○ PC

ChoosePHY RS485

X_NET
NetID 1
StationID 1
NetType TBN
BaudRate 1500000

TBN
TokenCycleTime 10
MaxStationNum 32

Note:Configration will take effect after the power is re-up

NetID: the network number of the two PLC. The device net ID in the same network must be the same. Here we set it to 1.

StationID: each PLC station number in the network. The two PLC station number is 1 and 2.

Net type:

TBN--- PLC communicate with PLC

TBN or HDN--- PLC communicate with HMI

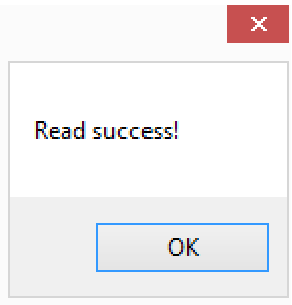
HDN--- PLC communicate with servo.

Baud rate: here we set it to 1.5M

TokenCycle time: the time of each station in the network cycle once, the unit is ms. Here we have two PLC, so the time set to 10ms.

Max station number: the max device numbers in the network.

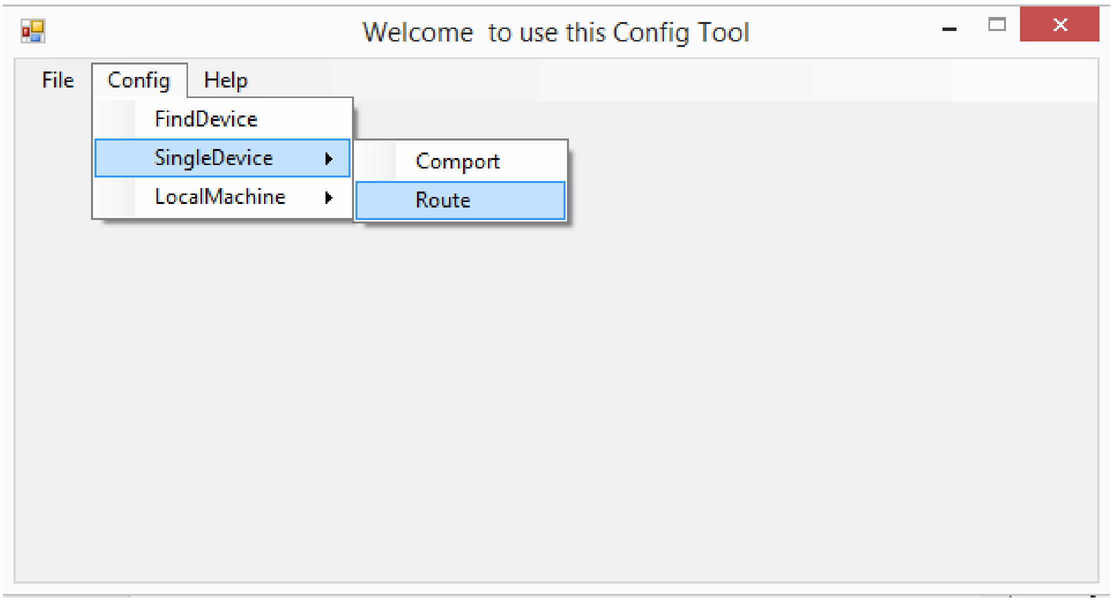
Click “write config”, it will show read success window.



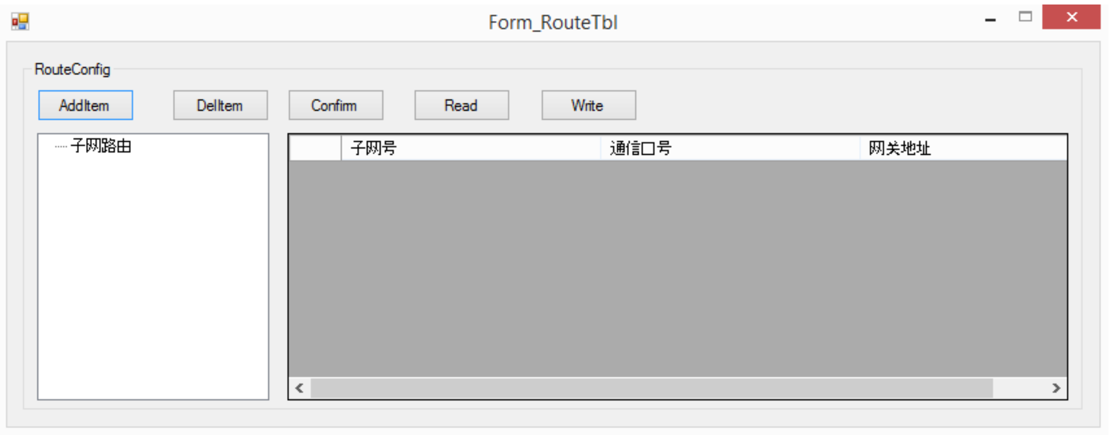
Read success!

Click ok to back to the main window. Cut off the PLC power and power on again to make the setting effective.

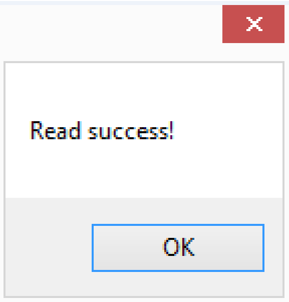
Then click config/single device/route.



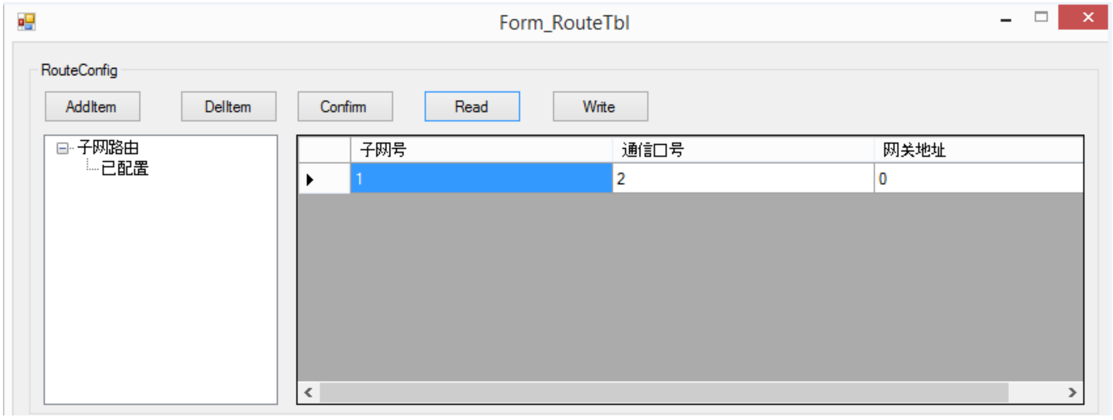
It shows the Form_RouteTb1 window.



Click “read”, it shows read success.


Read success!

Click ok, the Form_RouteTb1 window shows the subnet routing.



子网号	通信口号	网关地址
1	2	0

The subnet ID is the same to the netID. The communication port is the PLC physical terminal, RS485 port number is 2. Gateway address: the default setting is 0.

Click “write”, it will show write success. Then click ok. Please cut off the PLC power and power on again to make the setting effective.

Note:

1. After changing the device, it needs to configure again.

2. If it shows the error “there are some errors when communicating to XnetServer”, please restart the software and configure again.

1-2. Communication instruction

X-NET instructions include bit read and write, register read and write.

The communication instruction operand explanation:
1. Target net ID: all the devices construct the communication network. The ID of the network is the target net ID.
For example: 5 PLCs will communicate with each other in the network through X-NET protocol. User can define the net ID for this network.

2. Target station number: the station number of target device in the network.
For example: PLC connects to 3 frequency inverters, the frequency inverter station number is 1, 2, 3, and PLC station number is 4. The PLC will read the parameters of three inverters. Station number 1, 2, 3 are the target station number of PLC.

3. Target object type: the target device is coil or register.
For example:
The target device is coil X, the target object type is K1
The target device is coil Y, the target object type is K2
The target device is coil M, the target object type is K3
The target device is coil HM, the target object type is K8
The target device is register D, the target object type is K128
The target device is register HD, the target object type is K136

4. Target object number: the target device coil or register address in the network
For example: write register D0, write the D0 value to target address

5. Access object numbers: the first target register or coil address, always use together with object quantity.
For example: PLC needs to read the frequency inverter output frequency(H2103), output current(H2104) and bus voltage(H2105), the target object is H2103, the access object quantity is K3.

6. Local object address: the local coil or register address
For example: PLC register D0 value transfers to frequency inverter address H2103. So the local object address is D0, the target object address is H2103.

1-2-1. Read bit [BIT_READ]

1. Instruction explanation
Read the target coil to local coil.

Read bit [BIT_READ]

Execution conditoin	Edge trigger	Suitable model	XD, XG, XL
Hardware	V3.2 and up	Software	V3.2 and up

2. Operand

Operand	Function	Type
S1	Target net ID	16 bits constant or single word register
S2	Target station number	16 bits constant or single word register
S3	Target object type (refer to chapter 1-3)	16 bits constant or single word register
S4	Target object address (refer to chapter 1-3)	32 bits constant or double words register
S5	Access object numbers	16 bits constant or single word register
D1	Local object address	Local coil

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●				●		
S2	●	●		●	●				●		
S3	●	●		●	●				●		
S4	●	●		●	●				●		
S5	●	●		●	●				●		

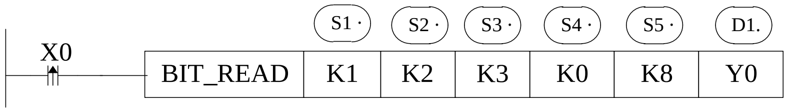
Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
D1	●	●	●	●	●	●	

*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM; S includes S, HS; T includes T, HT; C includes C, HC.

Intruction



Function: read the station no.2 coil M0~M7 to local address Y0~Y7 in ID1 network.

1-2-2. Write bit [BIT_WRITE]

1. Intruction explanation

Write the local coil value to target coil.

Write bit [BIT_WRITE]			
Execution conditoin	Edge trigger	Suitable model	XD, XG, XL
Hardware	V3.2 and up	Software	V3.2 and up

2. Operand

Operand	Function	Type
S1	Target net ID	16 bits constant or single word register
S2	Target station number	16 bits constant or single word register
S3	Target object type (refer to chapter 1-3)	16 bits constant or single word register
S4	Target object address (refer to chapter 1-3)	32 bits constant or double words register
S5	Access object numbers	16 bits constant or single word register
S6	Local object address	Local coil

3. Suitable soft component

Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●				●		
S2	●	●		●	●				●		
S3	●	●		●	●				●		
S4	●	●		●	●				●		
S5	●	●		●	●				●		

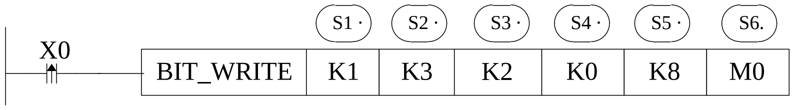
Bit

Operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S6	●	●	●	●	●	●	

*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM; S includes S, HS; T includes T, HT; C includes C, HC.

Intruction



Function: write the local coil M0~M7 to station no.3 Y0~Y7 in ID1 network.

1-2-3．Read register [REG_READ]

1. Instruction explanation

Read the target register to local register.

Read register [REG_READ]			
Execution conditoin	Edge trigger	Suitable model	XD, XG, XL
Hardware	V3.2 and up	Software	V3.2 and up

2. Operand

Operand	Function	Type
S1	Target net ID	16 bits constant or single word register
S2	Target station number	16 bits constant or single word register
S3	Target object type (refer to chapter 1-3)	16 bits constant or single word register
S4	Target object address (refer to chapter 1-3)	32 bits constant or double words register
S5	Access object numbers	16 bits constant or single word register
D1	Local object address	Local register

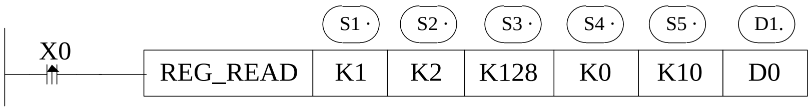
3. Suitable soft component

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
	S1	●	●		●	●				●		
	S2	●	●		●	●				●		
	S3	●	●		●	●				●		
	S4	●	●		●	●				●		
	S5	●	●		●	●				●		
	D1	●	●		●	●						

*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM; S includes S, HS; T includes T, HT; C includes C, HC.

Intruction



Function: read the station no.2 register D0~D9 to local address D0~D9 in ID1 network.

1-2-4. Write register [REG_WRITE]

1. Instruction explanation

Write the local register to target register.

Write register [REG_WRITE]			
Execution conditoin	Edge trigger	Suitable model	XD, XG, XL
Hardware	V3.2 and up	Software	V3.2 and up

2. Operand

Operand	Function	Type
S1	Target net ID	16 bits constant or single word register
S2	Target station number	16 bits constant or single word register
S3	Target object type (refer to chapter 1-3)	16 bits constant or single word register
S4	Target object address (refer to chapter 1-3)	32 bits constant or double words register
S5	Access object numbers	16 bits constant or single word register
S6	Local object address	Local register

3. Suitable soft component

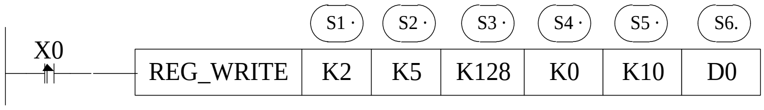
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ID	QD
S1	●	●		●	●				●		
S2	●	●		●	●				●		
S3	●	●		●	●				●		
S4	●	●		●	●				●		
S5	●	●		●	●				●		
S6	●	●		●	●						

*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM; S includes S, HS; T includes T, HT; C includes C, HC.

Intruction



Function: write the local register D0~D9 to station no.5 D0~D9 in ID2 network.

1-3. Communication address

The X-NET address of PLC soft components:
XD2, XD3, XL3 series PLC soft component address:

Type	Name	Soft component address	Type code	Quantity	X-NET address (decimal)	X-NET address (hex)
Coil, bit	X	X0~X77 (PLC)	1	64	0~63	0~3F
		X10000~X10077 (module 1)		64	4096~4159	1000~103F
		X10100~X10177 (module 2)		64	4160~4223	1040~107F
		X10200~X10277 (module 3)		64	4224~4287	1080~10BF
		X10300~X10377 (module 4)		64	4288~4351	10C0~10FF
		X10400~X10477 (module 5)		64	4352~4415	1100~113F
		X10500~X10577 (module 6)		64	4416~4479	1140~117F
		X10600~X10677 (module 7)		64	4480~4543	1180~11BF
		X10700~X10777 (module 8)		64	4544~4607	11C0~11FF
		X11000~X11077 (module 9)		64	4608~4671	1200~123F
		X11100~X11177 (module 10)		64	4672~4735	1240~127F
		X30000~X30077 (ED 1)		64	12288~12351	3000~303F
	Y	Y0~77 (PLC)	2	64	0~63	0~3F
		Y10000~Y10077 (module 1)		640	4096~4159	1000~103F
		Y10100~Y10177 (module 2)		64	4160~4223	1040~107F
		Y10200~Y10277 (module 3)		64	4224~4287	1080~10BF
		Y10300~Y10377 (module 4)		64	4288~4351	10C0~10FF
		Y10400~Y10477 (module 5)		64	4352~4415	1100~113F
		Y10500~Y10577 (module 6)		64	4416~4479	1140~117F

		Y10600~Y10677 (module 7)		64	4480~4543	1180~11BF
		Y10700~Y10777 (module 8)		64	4544~4607	11C0~11FF
		Y11000~Y11077 (module 9)		64	4608~4671	1200~123F
		Y11100~Y11177 (module 10)		64	4672~4735	1240~127F
		Y30000~Y30077 (ED 1)		64	12288~12351	3000~303F
	M	M0~M7999	3	8000	0~7999	0~1F3F
	S	S0~S1023	4	1024	0~1023	0~3FF
	T	T0~T575	5	576	0~575	0~23F
	C	C0~C575	6	576	0~575	0~23F
	ET	ET0~ET31	7	32	0~31	0~1F
	HM	HM0~HM959	8	960	0~959	0~3BF
	HS	HS0~HS127	9	128	0~127	0~7F
	HT	HT0~HT95	10	96	0~95	0~5F
	HC	HC0~HC95	11	96	0~95	0~5F
	HSC	HSC0~HSC31	12	32	0~31	0~1F
	SM	SM0~SM2047	13	2048	0~2047	0~7FF
	SEM	SEM0~SEM31	18	32	0~31	0~1F
Register, word	D	D0~D7999	128	8000	0~7999	0~1F3F
	TD	TD0~TD575	129	576	0~575	0~23F
	CD	CD0~CD575	130	576	0~575	0~23F
	SD	SD0~SD2047	131	2048	0~2047	0~7FF
	ETD	ETD0~ETD31	133	32	0~31	0~1F
	ID	ID0~ID99 (PLC)	134	100	0~99	0~63
		ID10000~ID10099 (module 1)		100	10000~10099	2710~2773
		ID10100~ID10199 (module 2)		100	10100~10199	2774~27D7
		ID10200~ID10299 (module 3)		100	10200~10299	27D8~283B
		ID10300~ID10399 (module 4)		100	10300~10399	283C~289F
		ID10400~ID10499 (module 5)		100	10400~10499	28A0~2903
		ID10500~ID10599 (module 6)		100	10500~10599	2904~2967
		ID10600~ID10699 (module 7)		100	10600~10699	2968~29CB
		ID10700~ID10799		100	10700~10799	29CC~2A2F

		(module 8)				
		ID10800~ID10899 (mdoule 9)		100	10800~10899	2A30~2A93
		ID10900~ID10999 (module 10)		100	10900~10999	2A94~2AF7
		ID30000~ID30099 (ED1)		100	30000~30099	7530~7593
	QD	QD0~QD99 (PLC)	135	100	0~99	0~63
		QD10000~QD10099 (module 1)		100	10000~10099	2710~2773
		QD10100~QD10199 (module 2)		100	10100~10199	2774~27D7
		QD10200~QD10299 (module 3)		100	10200~10299	27D8~283B
		QD10300~QD10399 (module 4)		100	10300~10399	283C~289F
		QD10400~QD10499 (module 5)		100	10400~10499	28A0~2903
		QD10500~QD1059 (module 6)		100	10500~10599	2904~2967
		QD10600~QD10699 (module 7)		100	10600~10699	2968~29CB
		QD10700~QD10799 (module 8)		100	10700~10799	29CC~2A2F
		QD10800~QD1089 (module 9)		100	10800~10899	2A30~2A93
		QD10900~QD10999 (module 10)		100	10900~10999	2A94~2AF7
		QD30000~QD30099 (ED 1)		100	30000~30099	7530~7593
	HD	HD0~HD999	136	1000	0~999	0~3E7
	HTD	HTD0~HTD95	137	96	0~95	0~5F
	HCD	HCD0~HCD95	138	96	0~95	0~5F
	HSCD	HSCD0~HSCD31	139	32	0~31	0~1F
	HSD	HSD0~HSD499	140	500	0~499	0~1F3
	FD	FD0~FD6143	141	6144	0~6143	0~17FF
	SFD	SFD0~SFD1999	142	2000	0~1999	0~7CF

XDM, XD5, XDC, XDE, XD5E, XG series PLC soft component address

Type	Name	Soft component address	Type code	Quantity	X-NET address (decimal)	X-NET address (hex)
Coil, bit	X	X0~X77 (PLC)	1	64	0~63	0~3F
		X10000~X10077		64	4096~4159	1000~103F

		(module 1)				
		X10100~X10177 (module 2)		64	4160~4223	1040~107F
		X10200~X10277 (module 3)		64	4224~4287	1080~10BF
		X10300~X10377 (module 4)		64	4288~4351	10C0~10FF
		X10400~X10477 (module 5)		64	4352~4415	1100~113F
		X10500~X10577 (module 6)		64	4416~4479	1140~117F
		X10600~X10677 (module 7)		64	4480~4543	1180~11BF
		X10700~X10777 (module 8)		64	4544~4607	11C0~11FF
		X11000~X11077 (module 9)		64	4608~4671	1200~123F
		X11100~X11177 (module 10)		64	4672~4735	1240~127F
		X11200~X11277 (module 11)		64	4736~4799	1280~12BF
		X11300~X11377 (module 12)		64	4800~4863	12C0~12FF
		X11400~X11477 (module 13)		64	4864~4927	1300~133F
		X11500~X11577 (module 14)		64	4928~4991	1340~137F
		X11600~X11677 (module 15)		64	4992~5055	1380~13BF
		X11700~X11777 (module 16)		64	5056~5119	13C0~13FF
		X30000~X30077 (ED 1)		64	12288~12351	3000~303F
	Y	Y0~77 (PLC)	2	64	0~63	0~3F
		Y10000~Y10077 (module 1)		640	4096~4159	1000~103F
		Y10100~Y10177 (module 2)		64	4160~4223	1040~107F
		Y10200~Y10277 (module 3)		64	4224~4287	1080~10BF
		Y10300~Y10377 (module 4)		64	4288~4351	10C0~10FF
		Y10400~Y10477		64	4352~4415	1100~113F

		(module 5)				
		Y10500~Y10577 (module 6)		64	4416~4479	1140~117F
		Y10600~Y10677 (module 7)		64	4480~4543	1180~11BF
		Y10700~Y10777 (module 8)		64	4544~4607	11C0~11FF
		Y11000~Y11077 (module 9)		64	4608~4671	1200~123F
		Y11100~Y11177 (module 10)		64	4672~4735	1240~127F
		Y11200~Y11277 (module 11)		64	4736~4799	1280~12BF
		Y11300~Y11377 (module 12)		64	4800~4863	12C0~12FF
		Y11400~Y11477 (module 13)		64	4864~4927	1300~133F
		Y11500~Y11577 (module 14)		64	4928~4991	1340~137F
		Y11600~Y11677 (module 15)		64	4992~5055	1380~13BF
		Y11700~Y11777 (module 16)		64	5056~5119	13C0~13FF
		Y30000~Y30077 (ED 1)		64	12288~12351	3000~303F
	M	M0~M74999	3	75000	0~74999	0~124F7
	S	S0~S7999	4	8000	0~7999	0~1F3F
	T	T0~T4999	5	5000	0~4999	0~1387
	C	C0~C4999	6	5000	0~4999	0~1387
	ET	ET0~ET39	7	40	0~39	0~27
	HM	HM0~HM11999	8	12000	0~11999	0~2EDF
	HS	HS0~HS999	9	1000	0~999	0~3E7
	HT	HT0~HT1999	10	2000	0~1999	0~7CF
	HC	HC0~HC1999	11	2000	0~1999	0~7CF
	HSC	HSC0~HSC39	12	40	0~39	0~27
	SM	SM0~SM4099	13	5000	0~4999	0~1387
	SEM	SEM0~SEM999	18	1000	0~999	0~3E7
Register, word	D	D0~D69999	128	70000	0~6999	0~1B57
	TD	TD0~TD4999	129	5000	0~4999	0~1387
	CD	CD0~CD4999	130	5000	0~4999	0~1387
	SD	SD0~SD4999	131	5000	0~4999	0~1387
	ETD	ETD0~ETD39	133	40	0~39	0~27
	ID	ID0~ID99 (PLC)	134	100	0~99	0~63

		ID10000~ID10099 (module 1)		100	10000~10099	2710~2773
		ID10100~ID10199 (module 2)		100	10100~10199	2774~27D7
		ID10200~ID10299 (module 3)		100	10200~10299	27D8~283B
		ID10300~ID10399 (module 4)		100	10300~10399	283C~289F
		ID10400~ID10499 (module 5)		100	10400~10499	28A0~2903
		ID10500~ID10599 (module 6)		100	10500~10599	2904~2967
		ID10600~ID10699 (module 7)		100	10600~10699	2968~29CB
		ID10700~ID10799 (module 8)		100	10700~10799	29CC~2A2F
		ID10800~ID10899 (module 9)		100	10800~10899	2A30~2A93
		ID10900~ID10999 (module 10)		100	10900~10999	2A94~2AF7
		ID11000~ID11099 (module 11)		100	11000~11099	2AF8~2B5B
		ID11100~ID11199 (module 12)		100	11100~11199	2B5C~2BBF
		ID11200~ID11299 (module 13)		100	11200~11299	2BC0~2C23
		ID11300~ID11399 (module 14)		100	11300~11399	2C24~2C87
		ID11400~ID11499 (module 15)		100	11400~11499	2C88~2CEB
		ID11500~ID11599 (module 16)		100	11500~11599	2CEC~2D4F
		ID30000~ID30099 (ED 1)		100	30000~30099	7530~7593
	QD	QD0~QD99 (PLC)	135	100	0~99	0~63
		QD10000~QD10099 (module 1)		100	10000~10099	2710~2773
		QD10100~QD10199 (module 2)		100	10100~10199	2774~27D7
		QD10200~QD10299 (module 3)		100	10200~10299	27D8~283B
		QD10300~QD10399 (module 4)		100	10300~10399	283C~289F

		QD10400~QD10499 (module 5)		100	10400~10499	28A0~2903
		QD10500~QD10599 (module 6)		100	10500~10599	2904~2967
		QD10600~QD10699 (module 7)		100	10600~10699	2968~29CB
		QD10700~QD10799 (module 8)		100	10700~10799	29CC~2A2F
		QD10800~QD10899 (module 9)		100	10800~10899	2A30~2A93
		QD10900~QD10999 (module 10)		100	10900~10999	2A94~2AF7
		QD11000~QD11099 (module 11)		100	11000~11099	2AF8~2B5B
		QD11100~QD11199 (module 12)		100	11100~11199	2B5C~2BBF
		QD11200~QD11299 (module 13)		100	11200~11299	2BC0~2C23
		QD11300~QD11399 (module 14)		100	11300~11399	2C24~2C87
		QD11400~QD11499 (module 15)		100	11400~11499	2C88~2CEB
		QD11500~QD11599 (module 16)		100	11500~11599	2CEC~2D4F
		QD30000~QD30099 (ED 1)		100	30000~30099	7530~7593
	HD	HD0~HD24999	136	25000	0~24999	0~61A7
	HTD	HTD0~HTD1999	137	2000	0~1999	0~7CF
	HCD	HCD0~HCD1999	138	2000	0~1999	0~7CF
	HSCD	HSCD0~HSCD39	139	40	0~39	0~27
	HSD	HSD0~HSD1999	140	2000	0~1999	0~7CF
	FD	FD0~FD8191	141	8192	0~8191	0~1FFF
	SFD	SFD0~SFD5999	142	6000	0~5999	0~176F

Note:

1. The input and output is octal, please calculate the input and output X-NET address as octal.
2. make sure there is no cross-border access for the target object.
3. XD5 series PLC register D range is D0~D59999.

1-4. X-NET Communication application

Application 1:

There are two XD3 sereis PLC. They communicate with each other through XD-NE-BD based on X-NET protocol.

The purposes:

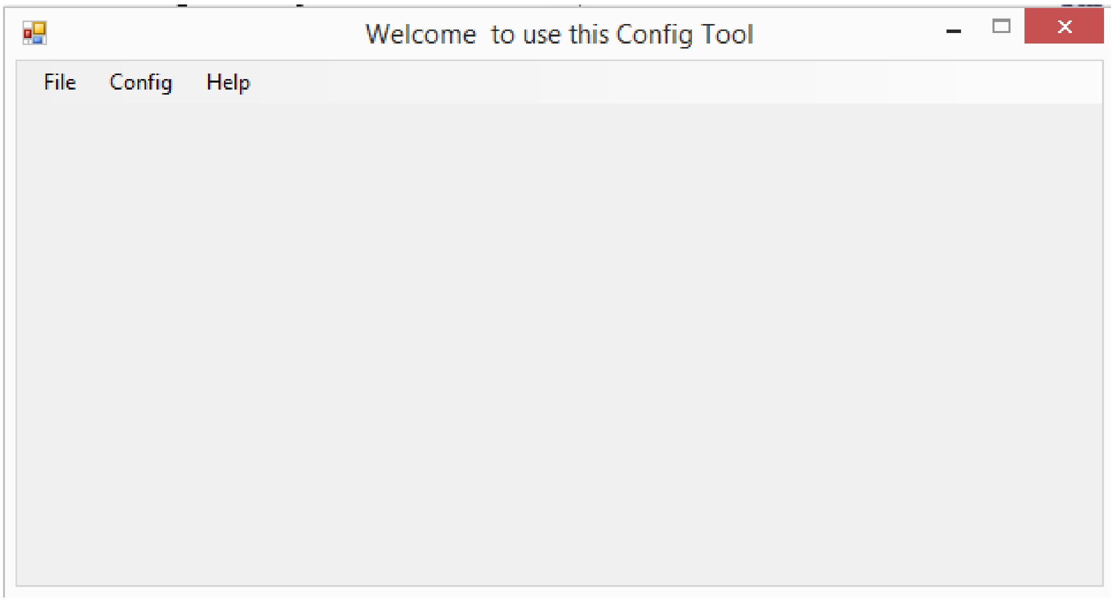
1. Write PLC A register D0~D9 to PLC B D0~D9
2. Write PLC A coil M0~M9 to PLC B M0~M9
3. Read PLC B register D0~D9 to PLC A D10~D19
4. Read PLC B coil M0~M9 to PLC A M10~M19

Step 1: wiring

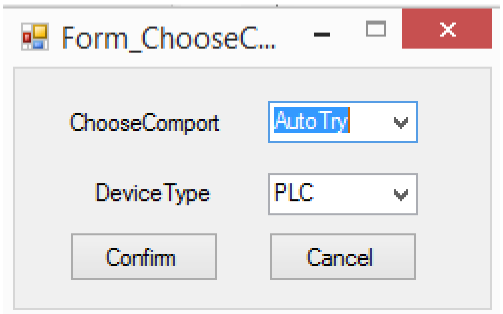
Connect terminal A to A, terminal B to B of the XD-NE-BD.

Step 2: RS485 port setting

1. Connect PLC A with PC through USB cable. Open the XINJEConfig software.



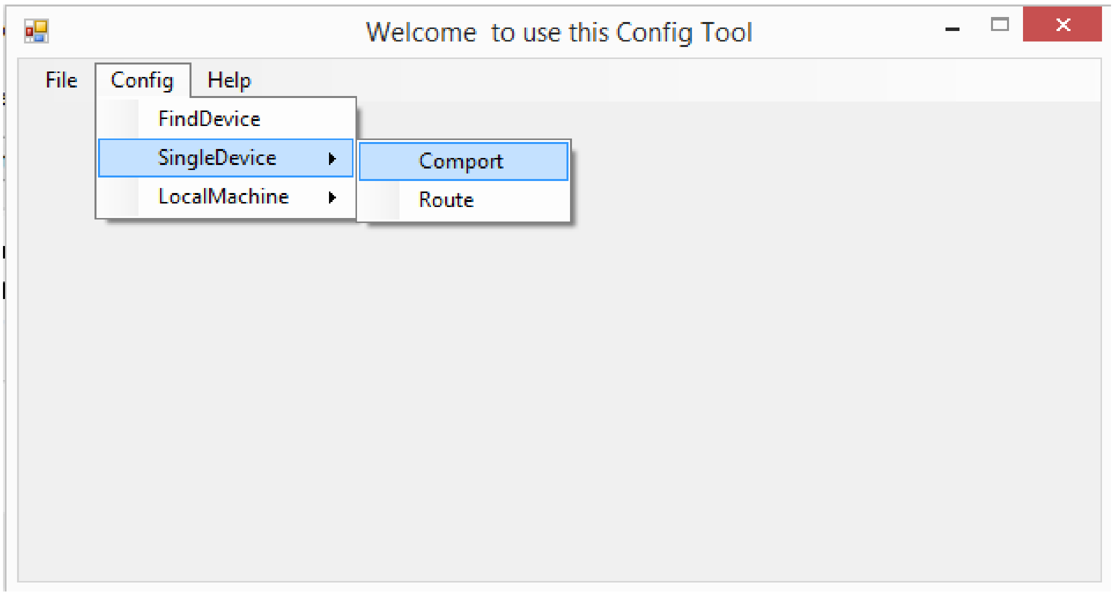
Choose the com port connected PLC, device type is PLC.



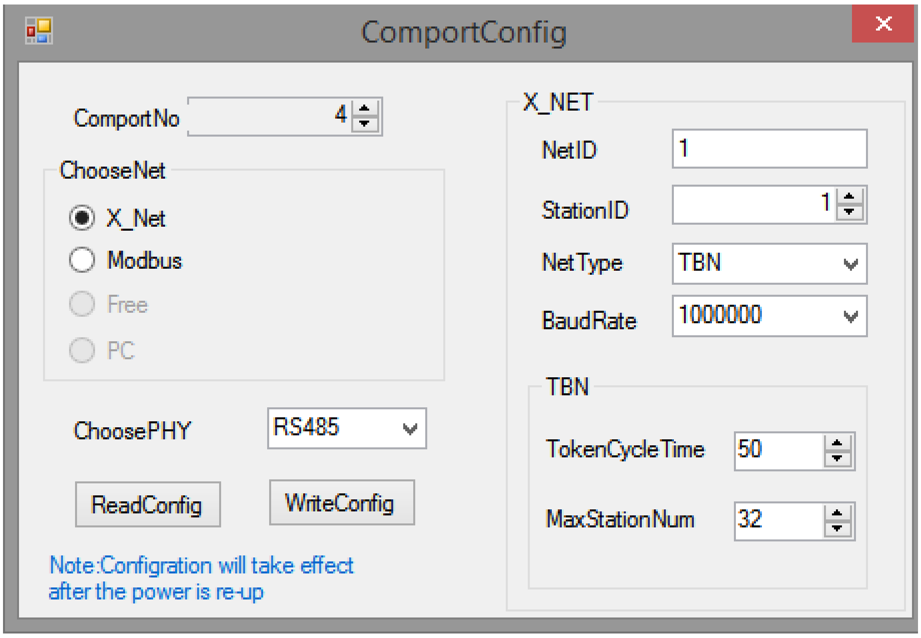
ChooseComport: Auto Try

DeviceType: PLC

Click ok to back to main window. Then click config/single deivce/com port.



As the PLC expansion XD-RS485-BD port is serial port 4, here we choose no.4 for comportNo. The protocol is X-NET. The physical layer is RS485.



ComportConfig

ComportNo 4

ChooseNet
● X_Net
○ Modbus
○ Free
○ PC

ChoosePHY RS485

X_NET
NetID 1
StationID 1
NetType TBN
BaudRate 1000000

TBN
TokenCycleTime 50
MaxStationNum 32

Note:Configration will take effect after the power is re-up

NetID: the network number of the two PLC. The device net ID in the same network must be the same. Here we set it to 1.

StationID: each PLC station number in the network. The two PLC station number is 1 and 2.

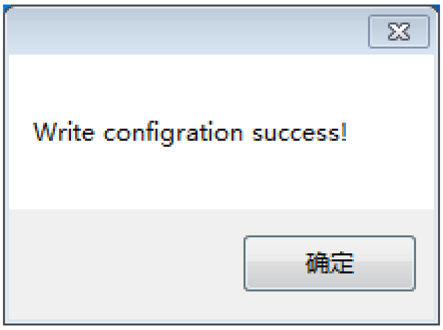
Net type: it is PLC communication, please choose TBN. If it is HMI and PLC communication, please choose OMMS or TBN. If it is PLC and servo communication, please choose OMMS. If it is PLC and analog ED board communication, please choose PPFD.

Baud rate: here we use shielded twisted pair, the max baud rate please choose 1M, if it is optical fiber, the max baud rate can be 3M.

TokenCycle time: the time of each station in the network cycle once, the unit is ms. If the communication data quantity is large, please set a large time to ensure the integrity of data. Here we only have two PLC, so the time is set to 50ms.

Max station number: the max device numbers in the network. As the X-NET communication max station no. is 32, so it is set to 32.

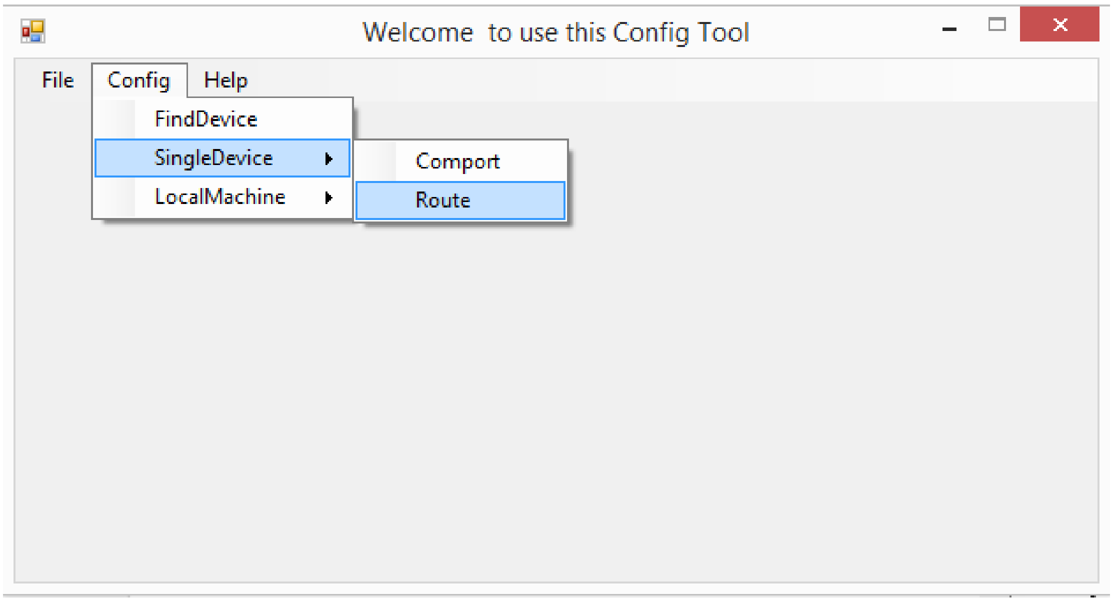
Click “write config”, it will show write success window.



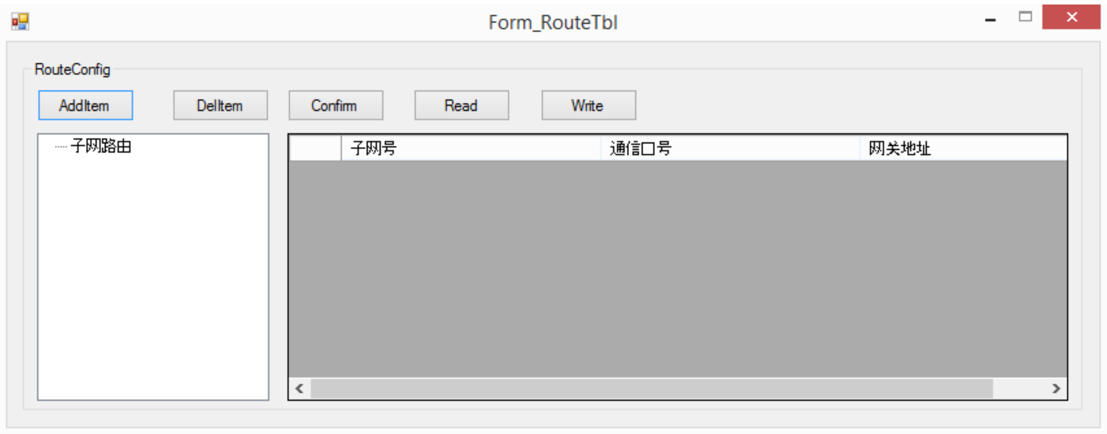
Write configration success!

Click ok to back to the main window. Cut off the PLC power and power on again to make the setting effective.

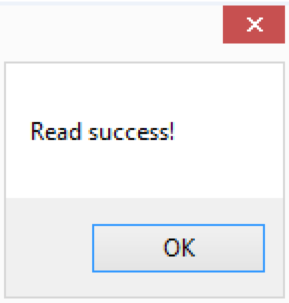
Then click config/single device/route.



It shows the Form_RouteTb1 window.

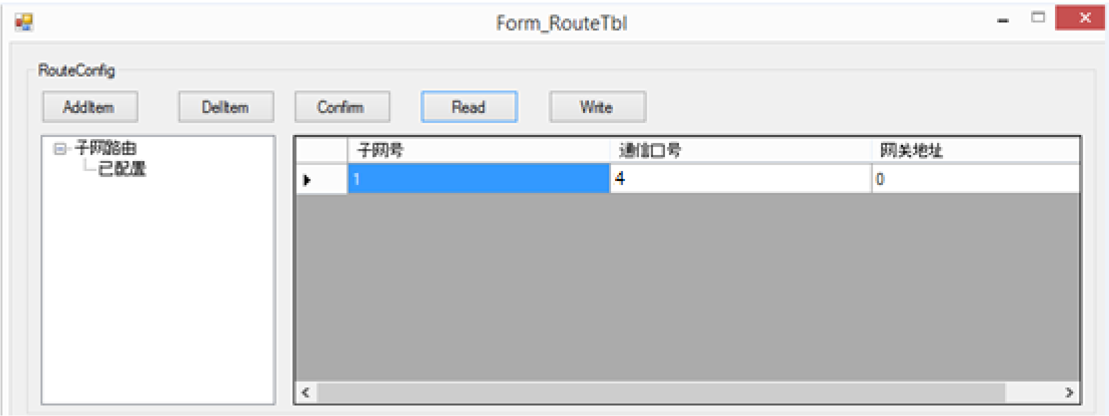


Click “read”, it shows read success.



Read success!

Click ok, the Form_RouteTb1 window shows the subnet routing.



子网号	通信口号	网关地址
1	4	0

The subnet ID is the same to the netID. The communication port is the PLC physical terminal, XD-NE-BD port number is 4. Gateway address: the default setting is 0.

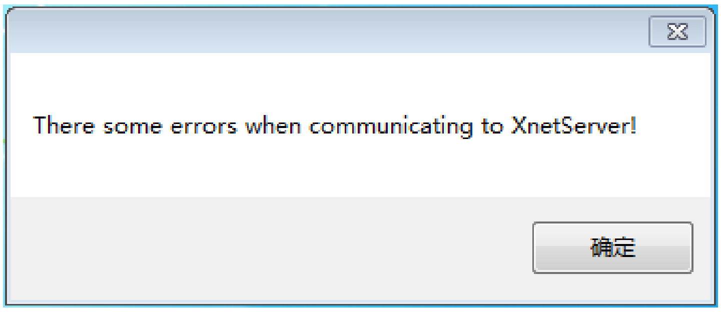
Click “write”, it will show write success. Then click ok. Please cut off the PLC power and power on again to make the setting effective.

PLC B RS485 configuration is the same to PLC A.

Note:

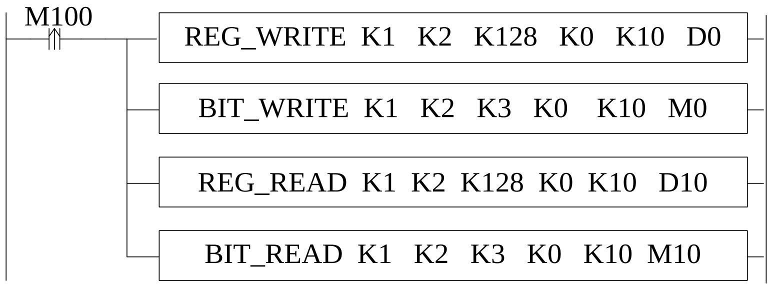
(1) After changing the device, it needs to configure again.

(2) If it shows below window, please restart the software and configure again.



There some errors when communicating to XnetServer!

Step 3: the program



```
REG_WRITE  K1   K2   K128   K0   K10   D0
BIT_WRITE  K1   K2   K3   K0    K10   M0
REG_READ  K1  K2  K128  K0  K10   D10
BIT_READ  K1   K2   K3   K0   K10  M10
```

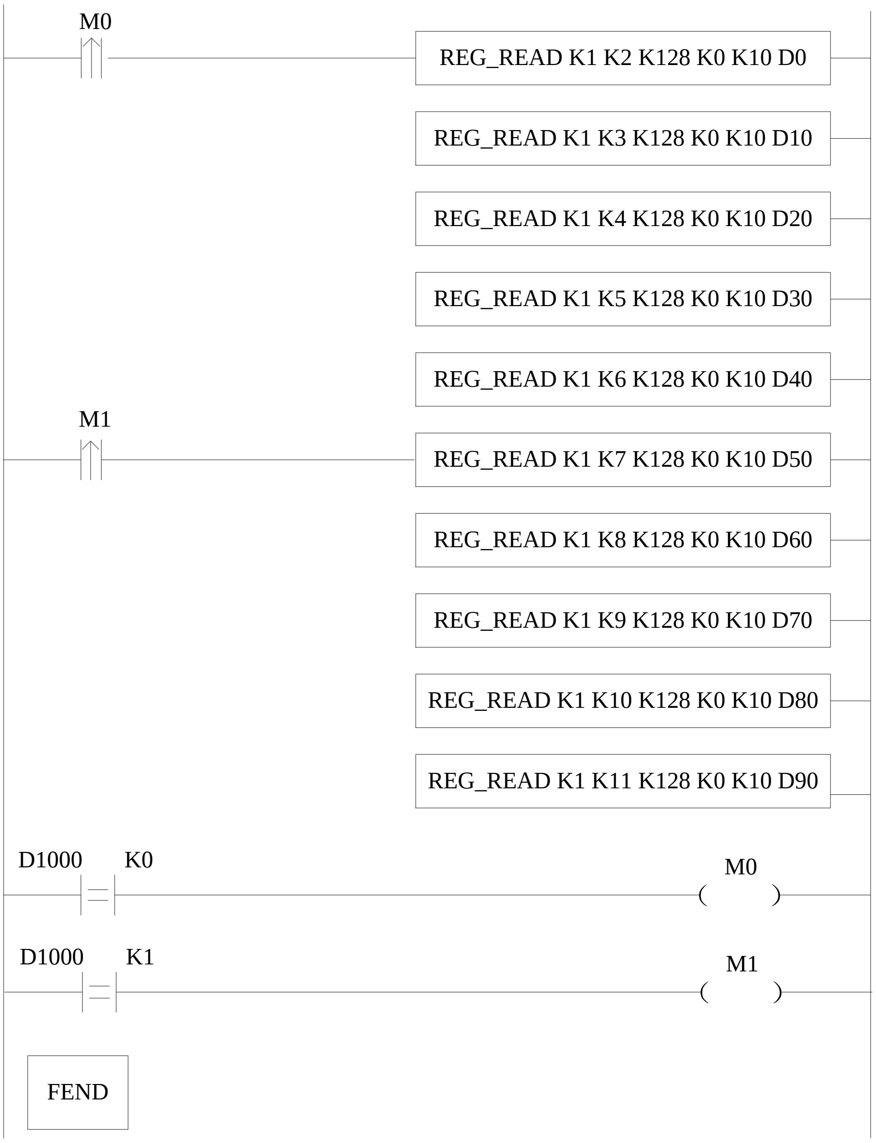
Application 2:

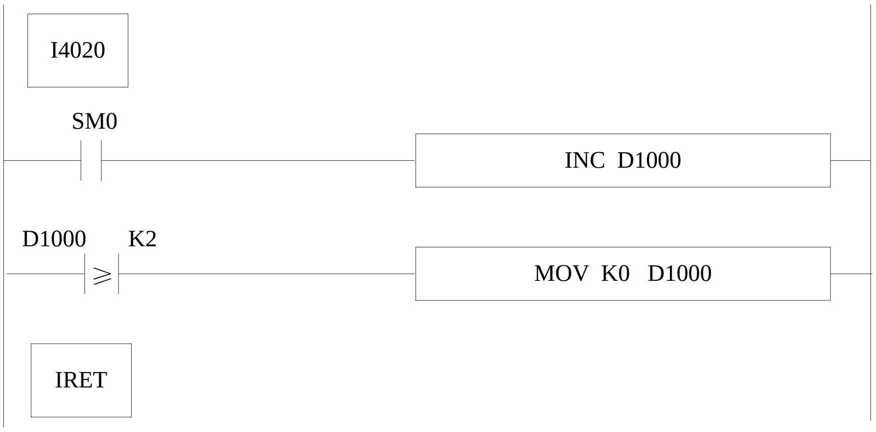
XD3-32T-E real-time reads D0~D9 of other 10 PLCs through the XD-NE-BD, then saves in D0~D100.

Step 1 wiring: connect all the terminal A to A, B to B of the XD-NE-BD. Then turn on the switch of the first and last XD-NE-BD in the fieldbus.

Step 2 port setting: the first PLC configuration is same to application 1. Other 10 PLCs must set the station no. to 2~11. Other settings are same to the first PLC.

Step 3 program:


M0
REG_READ K1 K2 K128 K0 K10 D0
REG_READ K1 K3 K128 K0 K10 D10
REG_READ K1 K4 K128 K0 K10 D20
REG_READ K1 K5 K128 K0 K10 D30
REG_READ K1 K6 K128 K0 K10 D40
M1
REG_READ K1 K7 K128 K0 K10 D50
REG_READ K1 K8 K128 K0 K10 D60
REG_READ K1 K9 K128 K0 K10 D70
REG_READ K1 K10 K128 K0 K10 D80
REG_READ K1 K11 K128 K0 K10 D90
D1000
K0
M0
D1000
K1
M1
FEND



Note:

(1) As different model of PLC has different memory which will affect the communication ability. We suggest that not put all the communication instructions in the same condition. Here we divide the 10 instructions into 2, each condition drives 5 instuctions. (XD3, XD2, XL3 cannot over 5 instructions, XD5, XDM, XDC cannot over 10, XDE, XD5E, XG1

cannot over 20)

(2) To ensure the communication integrity, here it triggers 5 instructions every 20ms. When the station or instruction quantity are large, please use pulse edge to trigger the instruction and extend trigger time interval. Normally close coil is not recommended to trigger the instruction.

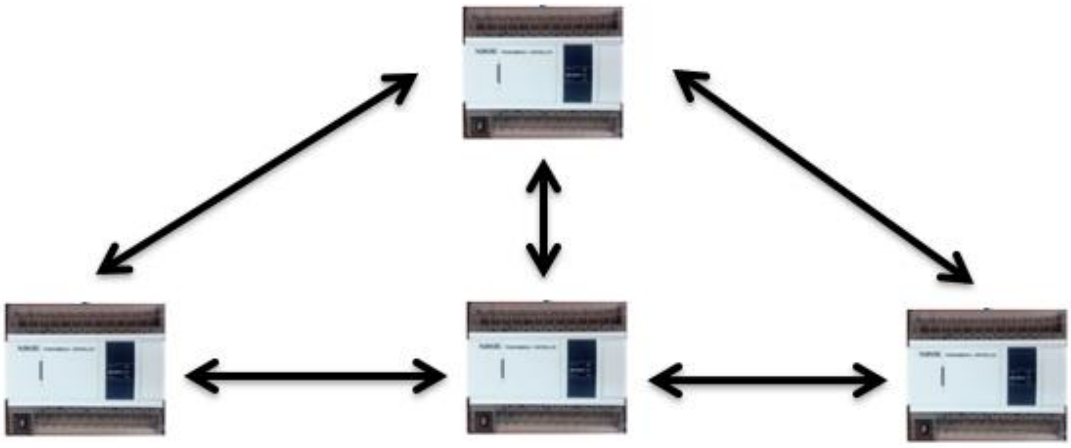
(3) Influenced by baud rate, it will get the best effect when the data quantity is 200word/10ms at baud rate 1M (1 word=1 register=16 coils). For example, trigger 5 instructions every 20ms, the register quantity of read instruction under 80 will get the best effect.

Application 3: multi-master and multi-slave mode

Modbus communication only can perform one master and multi-slave mode. X-NET fieldbus can support master-slave mode and multi-master multi-slave mode. In multi-master multi-slave mode, each PLC can be master to read and write other PLCs, it also can be accessed by other PLCs. This mode has high communication efficiency and reliability, flexible to control, is used in the application which has large quantity stations and complicated communication requirements.

The wiring method is same to master-slave mode, which is serial connection. Communication baud rate, configuration method, instructions are same to master-slave mode.

Below diagram is multi-master multi-slave networking mode. The black arrow means read and write the PLC.



Note: as the multi-master is in the same network, the communication quantity of each master station cannot over 200word/10ms. The data quantity of each instruction cannot over 125 words.

1-5． Communication register

1-5-1. Communication auxiliary register

	Register	Function	Explanation
Serial port 0	SD140	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station no. error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave busy 405: memory error (FLASH erasing)
	SD141	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD142	Free format communication send result	0: correct 410: free format send buffer overflow
	SD143	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start symbol 416: no end symbol
	SD144	Free format communication receive data quantity	Count by byte, not include start and end symbol
	……		
	SD149		
Serial port 1	SD150	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error

			182: station no. error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave busy 405: memory error (FLASH erasing)
	SD151	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD152	Free format communication send result	0: correct 410: free format send buffer overflow
	SD153	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start symbol 416: no end symbol
	SD154	Free format communication receive data quantity	Count by byte, not include start and end symbol
	……		
	SD159		
Serial port 2	SD160	Modbus read write instruction execution result	0: correct 100: receive error 101: receive overtime 180: CRC error 181: LRC error 182: station number error 183: send buffer overflow 400: function code error 401: address error 402: length error 403: data error 404: slave station busy 405: memory error　(FLASH easing)
	SD161	X-Net communication result	0: correct 1: communication overtime 2: memory error 3: receive CRC error
	SD162	Free format communication	0: correct

		send result	410: free format send buffer overflow
	SD163	Free format communication receive result	0: correct 410: send data length overflow 411: receive data short 412: receive data long 413: receive error 414: receive overtime 415: no start symbol 416: no end symbol
	SD164	Free format communication receive data quantity	Count as byte, not include start and end symbol
	……		
	SD169		
Serial port 3	SD170~SD179		
Serial port 4	SD180~SD189		
Serial port 5	SD190~SD199		

1-5-2. Communication auxiliary relay

	Coil	Function	Explanation
Serial port 1	SM1300	Station no. 1 in TBN network	Set ON means the station exsits
	SM1301	Station no. 2 in TBN network	
	SM1302	Station no. 3 in TBN network	
	SM1303	Station no. 4 in TBN network	
	……		
Serial port 2	SM1400	Station no. 1 in TBN network	Set ON means the station exsits
	SM1401	Station no. 2 in TBN network	
	SM1402	Station no. 3 in TBN network	
	SM1403	Station no. 4 in TBN network	
	……		
Serial port 3	SM1500~SM1599	Station in TBN network	Set ON means the station exsits
Serial port 4	SM1600~SM1699	Station in TBN network	Set ON means the station exsits
Serial port 5	SM1700~SM1799	Station in TBN network	Set ON means the station exsits

2 X-NET motion fieldbus

2-1. Function summary

X-NET motion bus is dedicated to providing a stable and reliable solution for multi-axis, high accuracy and responsive motion control systems.

XD series PLC can support X-NET motion bus model includes XDC, XDE, XG1 series.

XDC/XDE/XG1 series PLC has one channel motion fieldbus, can connect 20 axes at most, 2 axes pulse output. It can perform incremental position motion control, absolute position motion control, multi-segment motion control for single axis, synchronous motion control for multi-axis. Some instructions can real-time modify the target position and motion speed.

SM coil can perform jog run and back to origin for single axis.

2-1-1. Special vocabulary

The following table will explain the special vocabulary:

Vocabulary	Explanation
Origin	Electrical zero
machine datum point	mechanical zero
Zero point	The point whose pulse accumulated register is 0
Absolute position	The position of relative zero
Incremental position	Position offset before executing the relative instruction
Encoder feedback	Pulse feedback value of encoder counting
Axis state	Enable state, motion state

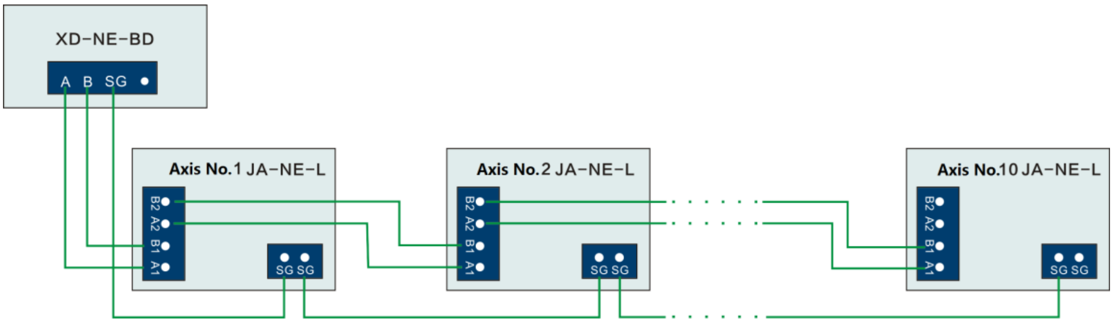
2-1-2. Fieldbus wiring

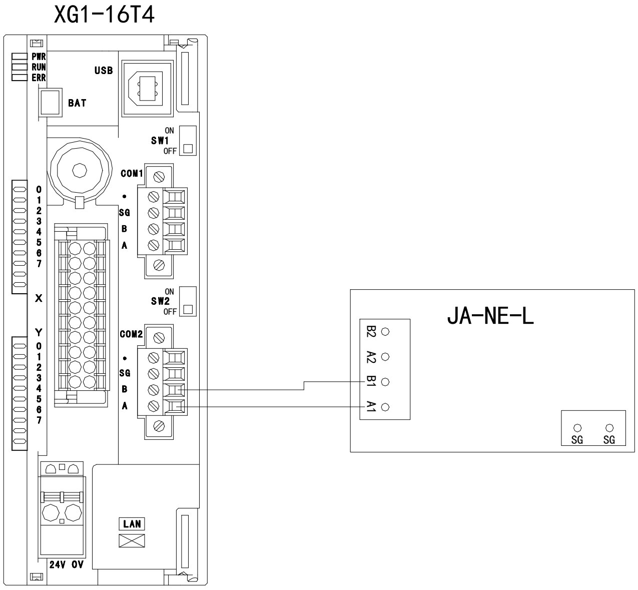
XD/XG series PLC is also can be called fieldbus multi-axis motion controller. The fieldbus motion controller communicates with servo drive through filedbus. The advantages are high performance, high reliability, easy to maintenance, save the wiring.

X-NET motion bus servo control system wiring: XDC/XDE

insert the RS485 expansion BD XD-NE-BD in the BD slot of PLC. There are 4 terminals on the BD: A, B, SG(signal), FG(ground).

The terminal A, B of BD will connect to the A1, B1 of JA-NE-L module. Insert the 9-pin female port of JA-NE-L into the CN1 port of servo drive. SG terminal of BD will connect to SG terminal of JA-NE-L. A1 and A2, B1 and B2 are shorted for module JA-NE-L. The following wiring diagram is one PLC control multi-servo.





XD-NE-BD **JA-NE-L**

X-NET motion bus servo control system wiring: XG1

XG1 must communicate with servo via serial port 2. The terminal of port 2 from up to down are SG, B, A. The terminal A and B will connect to DS3E servo drive module JA-NE-L terminal A1 and B1. A1 and A2, B1 and B2 are shorted for module JA-NE-L. SG terminal of BD will connect to SG terminal of JA-NE-L. Insert the 9-pin female port of JA-NE-L into the CN1 port of servo drive.

If one PLC controls multi-servo drives, the wiring method is as above diagram.

There is built-in terminal resistance in PLC BD and JA-NE-L. When the PLC connects many servo drives, in order to form a closed loop and reduce interference, please set ON the terminal resistance of BD board and the last JA-NE-L in the electric connection, other terminal resistance of JA-NE-L pelase set OFF.

Note:

1. The PLC can control 20 axes servo drives at the same time.
2. XDC/XDE PLC COM4 communication parameters keep default value. If it needs to change, please use XINJEConfig software to set the parameters. The configureation method please refer to chapter 2-6-1.
3. XG1 port 2 needs to configure the parameters manually, please refer to chapter 2-6-1.
4. Com1 and com2 of XDC series PLC are 8-pin round port (com1 default communication mode is X-NET, com2 default mode is Modbus).
5. The servo configuration method please refer to chapter 2-6-2.
6. It cannot only cut the power of servo drive when the PLC and servo power on.
7. Please cut off the power of XDC/XDE before install or uninstall the BD board.
8. To reduce the power supply interference, please connect the filter when PLC connects external AC power supply.
9. To reduce the power supply interference, please connect the filter when servo drive connects external AC power supply.
10. If the BD board communication parameters are correct in motion bus application, the XD-NE-BD light will flash, always ON or off are abnormal, please check the configuration or find other reasons.

2-1-3. PLC software

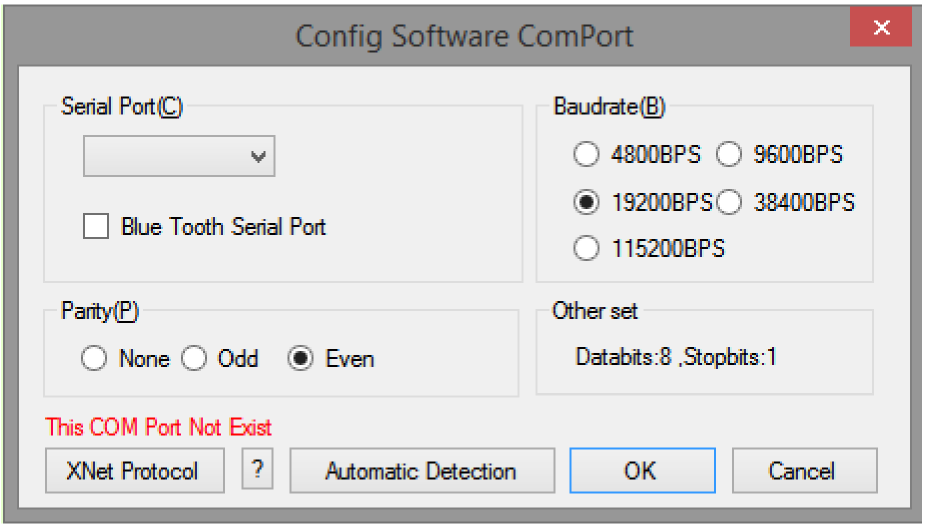
XDC series PLC needs to use XDPpro software (v3.2.1a and higher version).
It needs to install XNET service to support the XDPPro software running, please refer to chapter 2-6-1 for details.
Note: please run the software as administrator for the OS Win7 and up.

XD/XG series PLC software and PLC communication method:
After installing the XDPPro software and XNET service, open the XDPPro software.
XVP cable will connect the XDC PLC communication port and PC serial port, XDC series PLC panel has two 8-pin round communication ports, among them, the COM1 port default is X-Net communication, COM2 default is the ModBus communication, please connect them as you need.
XDE can choose x-net communication with XDPPro directly using USB cable (printer cable), or Modbus communication via 8-pin round port and Ethernet communication via Ethernet interface.
XG1 can communicate with XDPPro via USB (printer line) or Ethernet.

1. X-NET communication

For X-NET communication, please choose COM1 port. Open the XDPPro software, click software serial port config , it will show serial port config window(fig 2.1), click Xnet protocol, it will show Xnet config window(fig. 2.2), choose correct serial port and click ok, the system will connect automatically.

When the software status bar at the bottom right corner shows run or stop, that means the connection is successful. If it is failed, please restart the service in config window (fig. 2.3). click restart service to restart the service, or click stop service and start service to restart the service, then click ok to build the connection.



Config Software ComPort

Serial Port(C)

□ Blue Tooth Serial Port

Baudrate(B)

○ 4800BPS ○ 9600BPS

● 19200BPS ○ 38400BPS

○ 115200BPS

Parity(P)

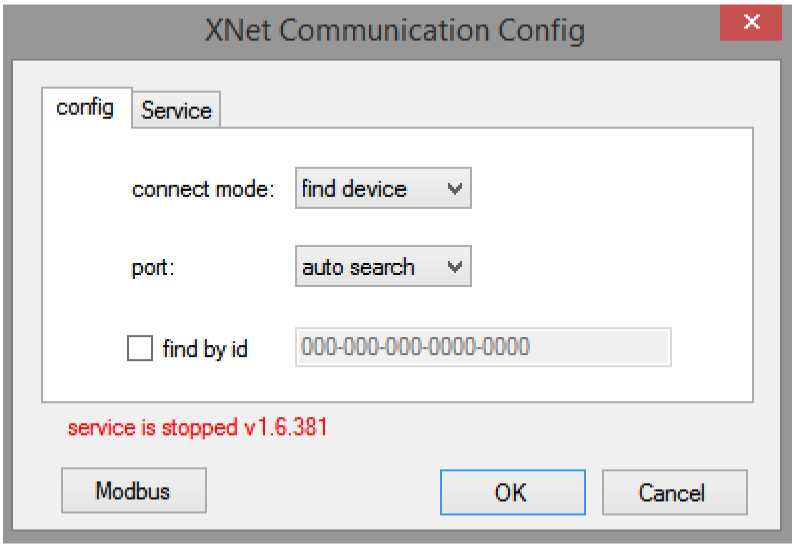
○ None ○ Odd ● Even

Other set

Databits:8 ,Stopbits:1

This COM Port Not Exist

Fig. 2.1



XNet Communication Config

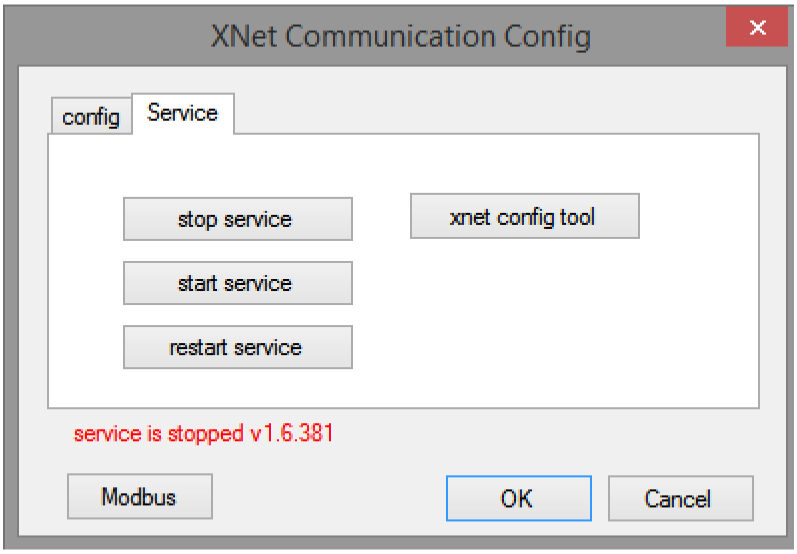
connect mode: find device

port: auto search

□ find by id 000-000-000-0000-0000

service is stopped v1.6.381

Fig. 2.2



service is stopped v1.6.381

Fig. 2.3

2. Modbus communication

For Modbus communication, please connect COM2. Open XDPPro software, click , it will show serial port config window (fig. 2.1), click automatic detection, when the status bar shows run or stop, that means the connection is successful.
After PLC connected PC, if the default model is not correct PLC model, please click file/change PLC model to choose the right one.

3. Ethernet communication

Refer to chapter 3-1.

2-2. Instructions

Motion fieldbus instructions:

Instruction	Function	Soft component	Chapter
MOTO	Incremental position motion	MOTO pos spd acc axNs	6-2-1
MOTOA	Absolute position motion	MOTOA pos spd acc axNs	6-2-2
MOTOS	Multi-segment speed run	MOTOS data para axNs	6-2-3
MOSTOP	Stop running	MOSTOP para axNs	6-2-4
MOGOON	Continue running	MOGOON axNs	6-2-5
MOSYN	Synchronous run	MOSYN para syn_axNs axNs	6-2-6
MOUSYN	Release synchronous run	MOUSYN axNs	6-2-7
MOWRITE	Write the present position	MOWRITE data axNs	6-2-8
MOREAD	Read the present position	MOREAD data axNs	6-2-9

2-2-1. Relative position motion [MOTO]

1. Instruction summarize

The instruction will run with relative position, it can real-time change the target absolute position, speed, acceleration deceleration time.

Incremental position motion [MOTO]			
16-bit instruction		32-bit instruction	MOTO
Execution condition	Rising edge or falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. operand

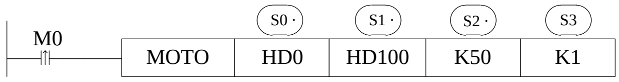
Operand	Function	Type
S0	Target position	32-bit integer
S1	Speed	32-bit integer
S2	Acceleration and deceleration time	32-bit integer
S3	Axis no.	16-bit constant

3. suitable soft component

word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
	S0	●								●		
	S1	●								●		
	S2	●								●		
	S3									●		

***Note: D means D, HD; TD means TD, HTD; CD menas CD, HCD, HSCD, HSD; DM means DM, DHM; DS means DS, DHS. M means M, HM, SM; S means S, HS; T means T, HT; C means C, HC.**

Function and action



● when M0 change from OFF to ON, axis S3 accelerates to speed S1 with acceleration speed S2, relative motion S0 stop.

S0: relative position, positive value means the motor running forward, negative value means the motor running reverse

S1: set to positive value, if set to negative value, it will move as absolute value.

S2: the time accelerating from 0 to setting speed, unit is ms.

S3: axis no. N, the range of N is 1~10

● relative position is the distance between present position and target position. For example: present position is 100, relative position is 300, it needs to send 300 pulses relative to the present position to reach the target position.

● when M0 from OFF to ON, absolute target position (SD2030+60*(N-1)) changes the relative position based on the original position, this value will be the target position.

● when it is running, to modify the value in register（SD2030+60*(N-1)）can real-time change the absolute target position. The instruction will move to the modified target position then stop. For example: the target position is 1000, present position is 0, it runs to position 600 when the condition is triggered.

(1) Change (SD2030+60*(N-1)) to 400 or -400, axis S3 forward decelerate until stop, then reverse accelerate to the position 400 or -400 and decelerate to stop. At this time change (SD2030+60*(N-1)) to 1200, axis S3 will forward run to position 1200 and decelerate to stop. (when the motor is enabled, write value in register SD2030+60*(N-1), the motor will run to related position. This operation can make the motor run forward or backward when the instruction is not executed.)

● servo enable, the speed register (SD2032+60*(N-1)) value is 1000, when M0 from OFF to ON, (SD2032+60*(N-1)) change to S1 value.

When the motor is running, modify (SD2032+60*(N-1)) can change the real-time speed, motor will change to new speed with setting acceleration/deceleration time.

If the speed is 0, the motor will stop with deceleration time. As the speed become 0 before reaching the target position, the motion complete flag (SM2001+20*(N-1)) will not reset. At this time, set new speed in (SD2032+60*(N-1)), the motor will run again.

Related register

After PLC is running, servo is enable, modify SD register can change the target absolute position, speed and other pararmeters, the parameters will be effective after 6~16ms. But the related regsiters in instruction will not affect the target position, speed and acc/dec time. One motion filedbus can connect 20 axes, the axis no. is 1 to 20. Please see the parameters of each axis in the following table.

Table 1: set parameters (N=1~20)

Address	Defintion	Type	Unit	Remark
SD2030+60*(N-1)	Absolute position	32-bit integer	Pulse numbers	Coordinate position is calculated from target position setting pulse numbers. Modify the position when stop or running, it will move toward the setting position with setting speed. The setting position is

				absolute position.
SD2032+ 60*(N-1)	Speed	32-bit integer	Pulse/second	
SD2034+ 60*(N-1)	Acceleration time	32-bit integer	ms	The time Accelerating from 0 max speed
SD2036+ 60*(N-1)	Deceleration time	32-bit integer	ms	The time decelerating from max speed to 0

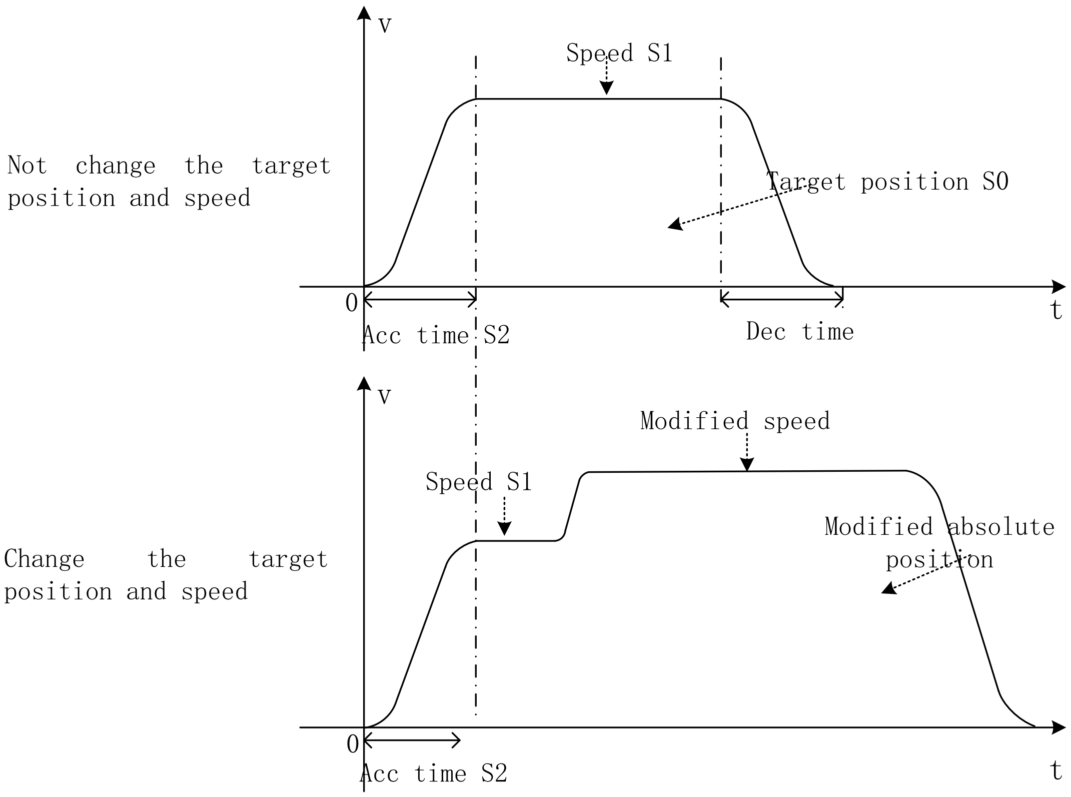
Table 2: state bit (N=1~20)

Address	Definition	Remark
SM2000+20*(N-1)	Servo enable flag	ON: servo enable state
SM2001+20*(N-1)	Running flag	ON: the pulse is outputting
SM2004+20*(N-1)	Axis error flag	ON: error

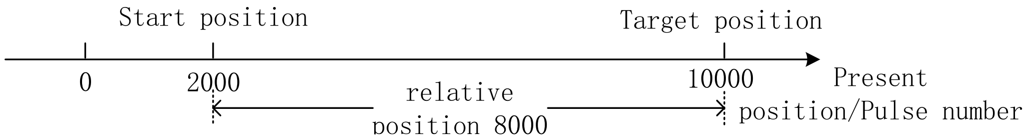
Application

The motor present position is 2000, it needs to use MOTO to run 10000 pulses with the speed 5000Hz. It will change the speed to 6000Hz, and target position 20000 pulses when the motor is running. The acceleration and deceleration time is 50ms.

◆relative position mode

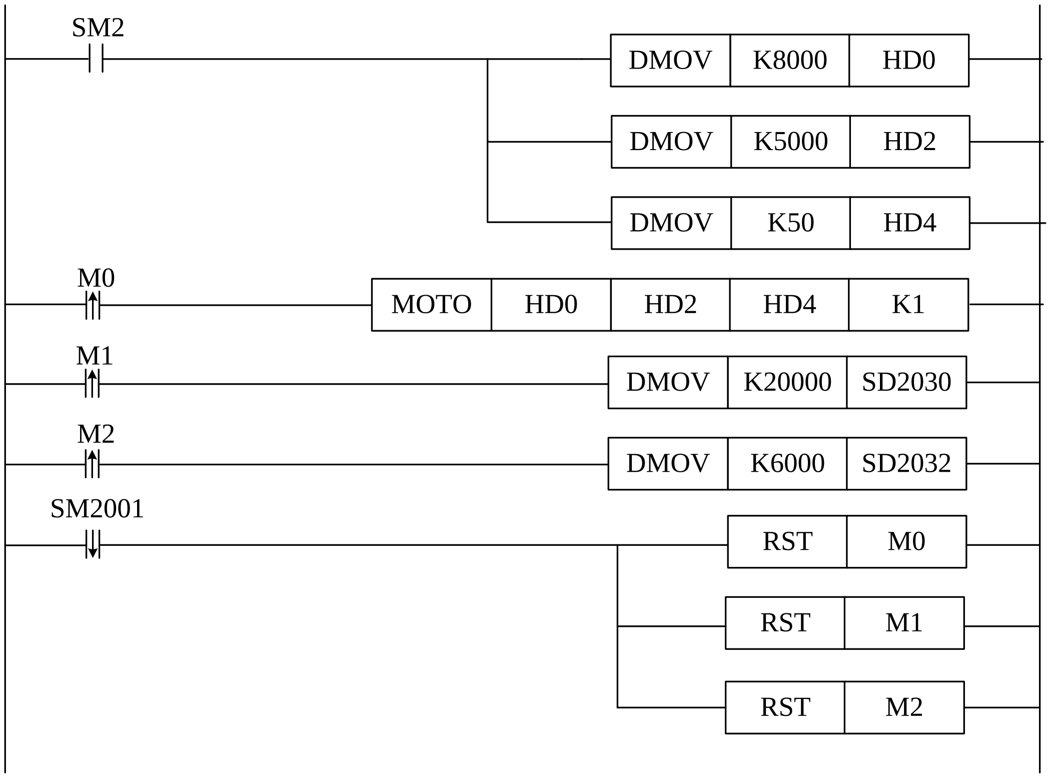

V
Speed S1
Not change the target position and speed
Target position S0
0
t
Acc time S2
Dec time
V
Modified speed
Speed S1
Modified absolute position
Change the target position and speed
0
Acc time S2
t

◆relative position mode, motor running distance



The present position is 2000, it needs 8000 pulse numbers to get the target position of 10000 pulse numbers in relative mode.

◆the program in relative mode



When SM2 is ON, send the pulse numbers, speed, acceleration deceleration time to the related registers.

Servo is enable, M0 is ffrom OFF to ON, it runs incremental motion instruction MOTO.

M1 is from OFF to ON, send absolute position to the related register.

M2 is from OFF to ON, send the new speed to the related register.

Running flag SM2001 is reset after the pulse sending is completed, reset M0, M1, M2.

2-2-2．Absolute position motion [MOTOA]

1. Instruction summarize

This instruction runs with absolute position, it can real-time modify the target position and speed when it is running.

Absolute position motion [MOTOA]			
16-bit		32-bit	MOTOA
Execution condition	Rising edge/falling edge of the coil	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

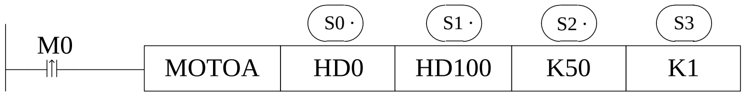
Operand	Function	Model
S0	Target position	32-bit integer
S1	Speed	32-bit integer
S2	The time accelerating from 0 to S1	32-bit integer
S3	Axis no.	16-bit constant

3. Suitable soft component

word	operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
	S0	●								●		
	S1	●								●		
	S2	●								●		
	S3									●		

*Note: D means D, HD; TD means TD, HTD; CD means CD, HCD, HSCD, HSD; DM means DM, DHM; DS means DS, DHS. M means M, HM, SM; S means S, HS, T means T, HT; C means C, HC.

Function and action



● when M0 is from OFF→ON, axis S3 accelerates to speed S1 with acceleration speed S2, absolute moves to position S0 and stop.

S0: absolute position , the value can be positive or negative , if the value is equal to the present position, the motor will not run. If the value is less than present position, the motor will reverse run.

S1: set to positive value, if set to negative value, it will run as abosulte value.

S2：the time accelerating from 0 to target speed, unit is ms.

S3: axis no. N, N is from 1 to 10.

● absolute position, the distance from zero point to target position. For example, the present position is 100, the absolute position is 300, the motor needs 200 pulses to the target position.

● when M0 is from OFF to ON, absolute target position (SD2030+60*(N-1)) changes to S0. If (SD2030+60*(N-1)) increases, the motor will run forward, if (SD2030+60*(N-1)) decreases, the motor will run reverse.

● when it is running, modify (SD2030+60*(N-1)) to change the absolute target position. The instruction will move as the modified target position until stop.

For example: the target position is 1000, when the condition is activated, it gets to position 600. (1) At this time, modify (SD2030+60*(N-1)) to 400 or -400, axis S3 will forward decelerate to stop, then reverse accelerates to position 400 or -400 and decelerate to stop. (2) at this time, (SD2030+60*(N-1)) is 1200, axis S3 forward runs to position 1200 and decelerates to stop.

● servo is enable, (SD2032+60*(N-1)) changed to 1000, when M0 is from OFF to ON, (SD2032+60*(N-1)) changed to S1.

When the motor is running, modify (SD2032+60*(N-1)) can change the real-time speed, the motor will change to new speed with acceleration and deceleration time.

If the speed is 0, the motor will stop with acceleration and deceleration time. As the speed become 0 before reaching the target position, the running flag (SM2001+20*(N-1)) will not reset. At this time, if set (SD2032+60*(N-1)) new speed, the motor will run again.

Related register

The related register about absolute position is same to relative motion instruction, please refer to chapter 2-2-1 tables.

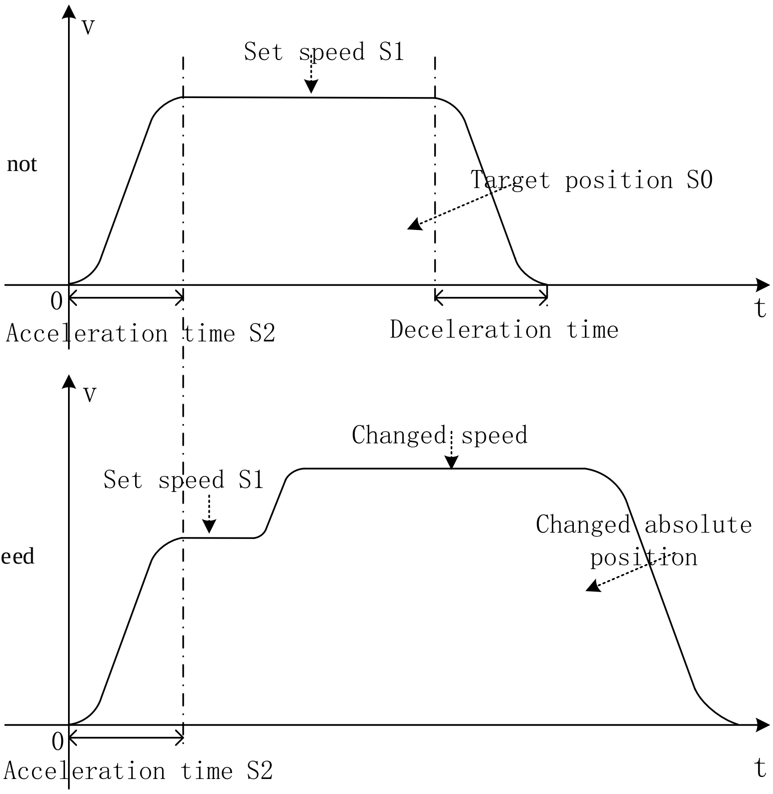
Application

The present position of the motor is 2000, it will moves 10000 pulse numbers with speed 5000Hz. When it is running, the speed will change to 6000Hz, the target absolute position changes to 20000 pulse numbers. The acceleration and deceleration time is 50ms.

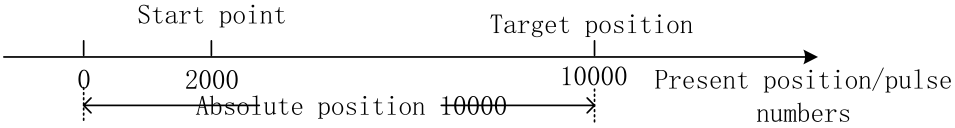
- the diagram in absolute position mode:

①the speed and position did not change

② target position and speed changed:

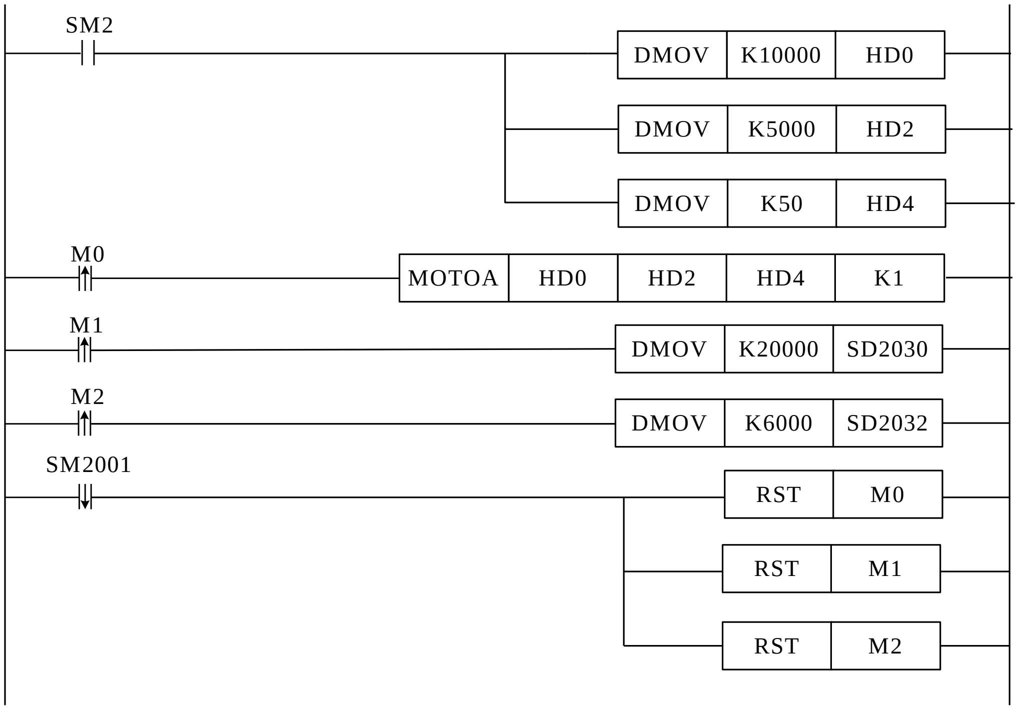


- motor running distance in absolute position mode



The present position is 2000, it needs 10000 pulse numbers to reach target position 10000 in absolute position mode.

- The ladder chart in absolute position mode:



SM2 is initial ON coil, it sends the pulse numbers, speed and acceleration, deceleration time in the related registers.
Servo drive is enable, M0 is from OFF to ON, it run the absolute position instruction MOTOA.
M1 is from OFF to ON, it sends the absolute target position in the related register,
M2 is from OFF to ON, it sends the new speed in the related register.
The running flag SM2001 reset after the pulse sending completed, related coil reset.

2-2-3. Multi-speed running [MOTOS]

1. Instruction summrize

This instruction cannot change the target position when running, but it can change the present speed.

Multi-speed running [MOTOS]			
16-bit		32-bit	MOTOS
Execution condition	Rising/falling edge	Suitable model	XDC, XDE, XG
Hardware	-	software	-

2. Operand

Operand	Function	Type
S0	Data starting address	32-bit integer
S1	Parameter starting address	32-bit integer
S2	Axis no.	16-bit constant

3. suitable soft component

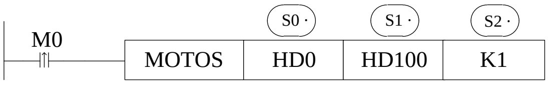
word

operand	System								constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
S0	●										
S1	●										
S2									●		

*Note: D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM.

DS includes DS, DHS. M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C HC.

Function and action



● when M0 is from OFF to ON, axis S2 will run in multi-speed incremental position mode or absolute position mode with S1 parameters, S2 target position and speed.

S0: data starting address. Set the pulse position and speed.

S1: parameter starting address. Set the running mode, running segment and acceleration/deceleration time.

S2: axis no. N, N range is from 1 to 10.

● servo is enable, speed register (SD2032+60*(N-1)) is 1000, when M0 is from OFF to ON, (SD2032+60*(N-1)) changed to S0 segment 1 speed.

When the motor is running, modify (SD2032+60*(N-1)) can cahnge the speed, the motor will change to new speed with acceleration/deceleration time. The modified speed is effective for the present segment.

If the speed is 0, the motor will stop with acceleration/deceleration time. As the speed becoming 0 before reaching the target position, the running flag (SM2001+20*(N-1)) will not reset. At this time, set new speed in (SD2032+60*(N-1)), the motor will run again.

● there is acceleration and deceleration time when the speed changed in each segment, the slope is same to the first segment.

● the present segment register (SD2016+60*(N -1)) can show the pulse is running in which segment

● when it is running, it can not modify the SD register to change the target position, acceleration/deceleration time, running mode and pulse segment numbers.

Related register

- **Data starting address:**

Address	Contents	Notes
S0+0 (double words)	Position	Segment 1
S0+2 (double words)	Speed	
S0+4	Reserved	
S0+6	Reserved	
S0+8	Reserved	
…	…	…
S0+(N-1)*10+0 (double words)	Position	Segment N
S0+(N-1)*10+2 (double words)	Speed	
S0+(N-1)*10+4	Reserved	
S0+(N-1)*10+6	Reserved	
S0+(N-1)*10+8	Reserved	

- **Parameter starting address:**

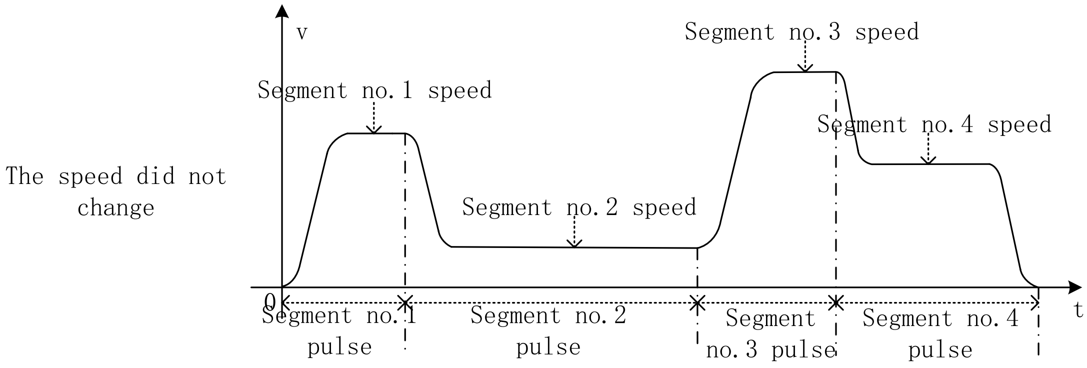
Address	**Contents**
S1+0 (double words)	32-bit integer, motion mode (0: relative, 1: absolute)
S1+2 (double words)	32-bit integer, motion segment (1~100)
S1+4 (double words)	32-bit integer, acceleration time (the time accelerate from 0 to the first segment speed, the following speed changing is same to this acceleration speed), unit: ms
S1+6 (double words)	32-bit integer, deceleration time (the time decelerate from the last segment speed to 0), unit: ms

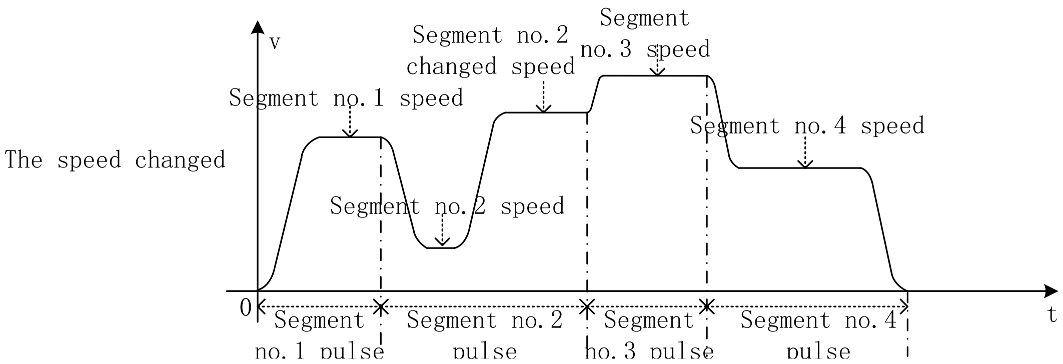
Application

It outputs four segments of pulse by MOTOS instruction. The speed is changed to 6000Hz at the second segment. Each segment parameters are shown as below:

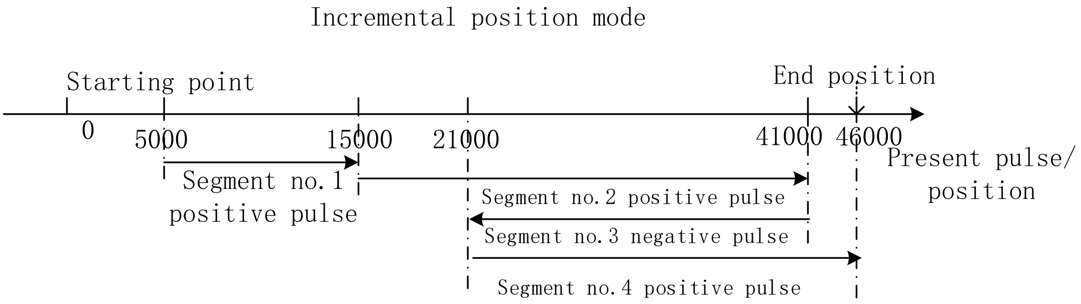
Name	Frequency (Hz)	Pulse numbers
Segment no.1	5000	10000
Segment no.2	1000	26000
Segment no.3	7500	-20000
Segment no.4	4000	25000
Acceleration/deceleration time	50ms	

- The running diagram:



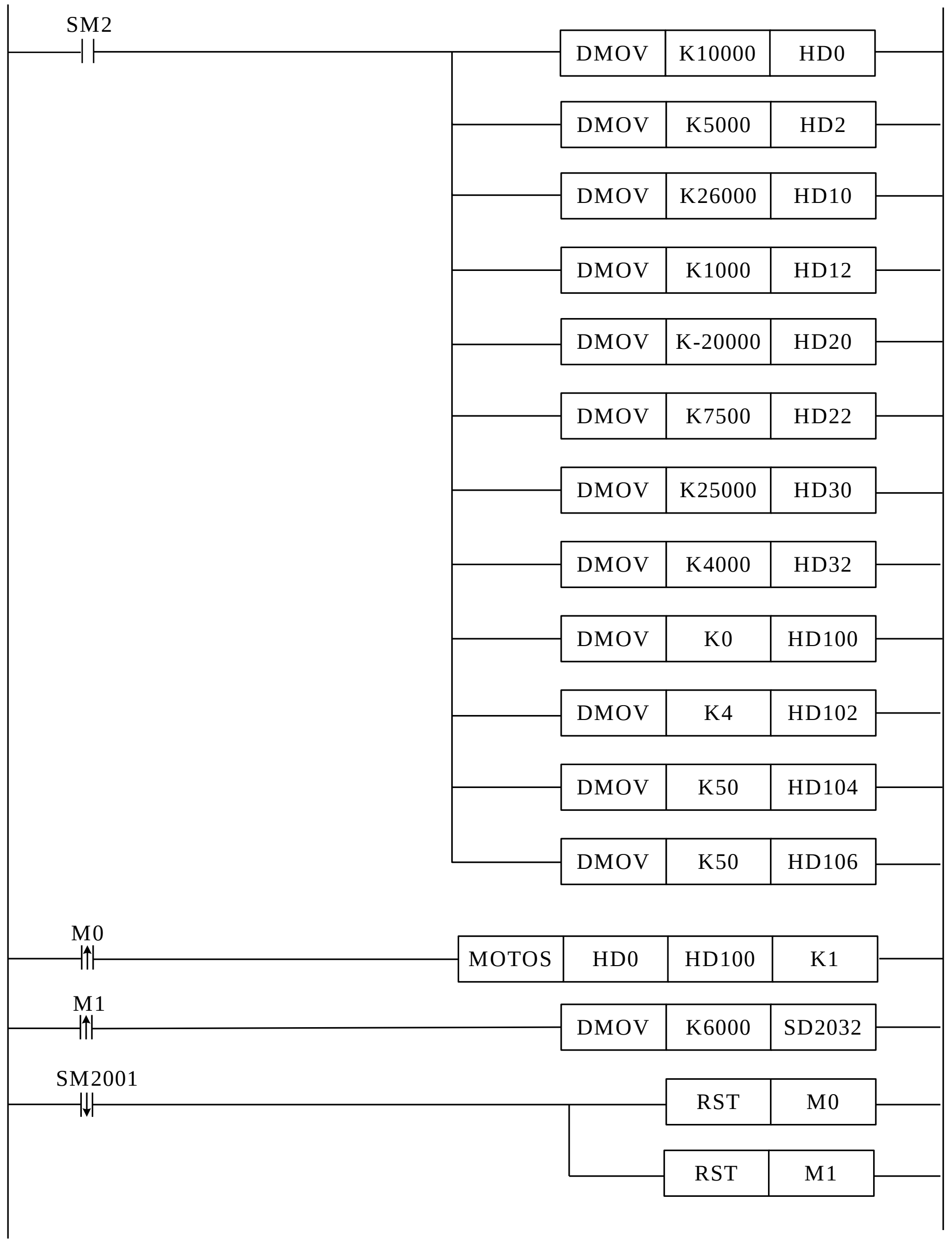


◆the motor running diagram in relative position mode


Incremental position mode
Starting point
End position
0
5000
15000
21000
41000
46000
Present pulse/ position
Segment no.1 positive pulse
Segment no.2 positive pulse
Segment no.3 negative pulse
Segment no.4 positive pulse

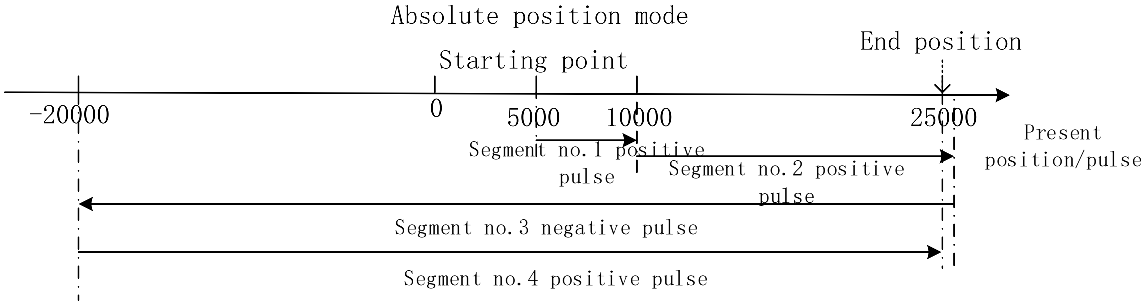
the present position is 5000, in the incremental position mode, the first segment sends 10000 pulses, the motor will go to the position of 15000 pulses; the second segment sends 26000 pulses, the motor will go to the position of 41000 pulses; the third segment sends -20000 pulses, the motor will reverse go to the position of 21000 pulses; the fourth segment sends 25000 pulses, the motor will go to the position of 46000 pulses.

◆the program in relative position mode:



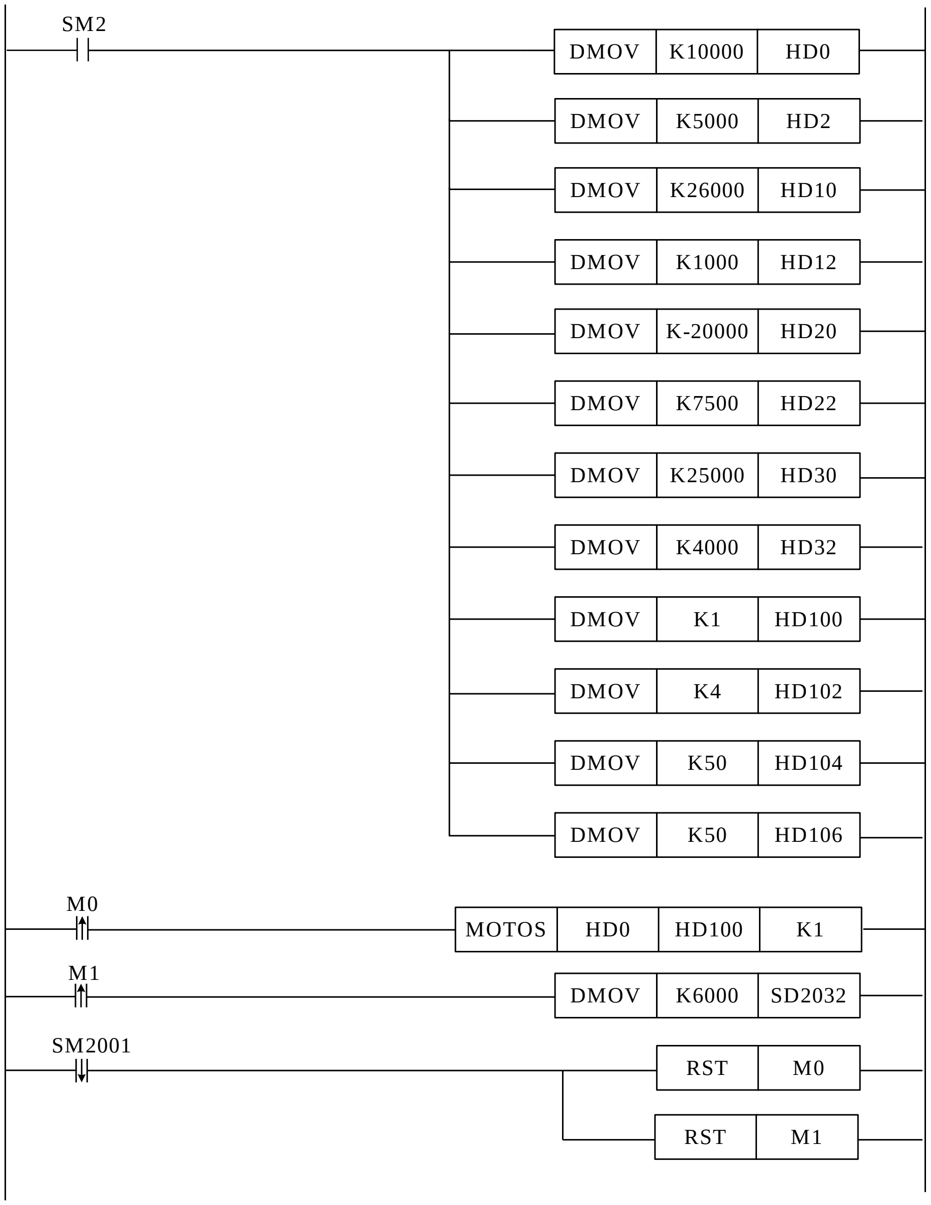
PLC start running, when the coil SM2 is ON, the pulse numbers, speed, motion mode, segment number, acceleration/deceleration time will be stored in related registers.
The servo enabled, M0 is from OFF to ON, it executes the instruction MOTOS.
M1 is from OFF to ON, the new speed is stored in the related register.
When the pulse sending end, the running flag SM2001 reset, the related coils will be reset.

◆the motor running diagram in absolute position mode:


Absolute position mode
Starting point
End position
-20000
0
5000
10000
25000
Present position/pulse
Segment no.1 positive pulse
Segment no.2 positive pulse
Segment no.3 negative pulse
Segment no.4 positive pulse

the present position is 5000, in absolute position mode, the first segment sends 5000 pulses, the motor goes to the position of 10000 pulses; the second segment sends 16000 pulses, the motor goes to the position of 26000 pulses; the third segment sends -46000 pulses, the motor reverse runs to the position of -20000; the fourth segment sends 45000 pulses, the motor goes to the position of 25000.

◆the program in absolute position mode:



PLC start running, when the coil SM2 is ON, the pulse numbers, speed, motion mode, segment number, acceleration/deceleration time will be stored in related registers.

The servo enabled, M0 is from OFF to ON, it executes the instruction MOTOS.

M1 is from OFF to ON, the new speed is stored in the related register.

When the pulse sending end, the running flag SM2001 reset, the related coils will be reset.

2-2-4．Stop running [MOSTOP]

1. Instruction summary

This instruction can stop multi-mode motion.

Stop running [MOSTOP]			
16-bit		32-bit	MOSTOP
Execution condition	Rising/falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S0	Stop mode or deceleration distance	32-bit integer
S1	Axis no.	16-bit constant

3. Suitable component

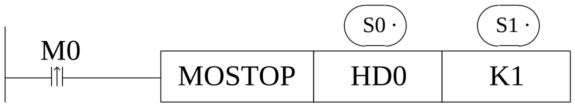
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
S0	●								●		
S1									●		

***Note:** D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and action



● when M0 is from OFF to ON, axis S1 stop running in different mode as the S0 parameters.

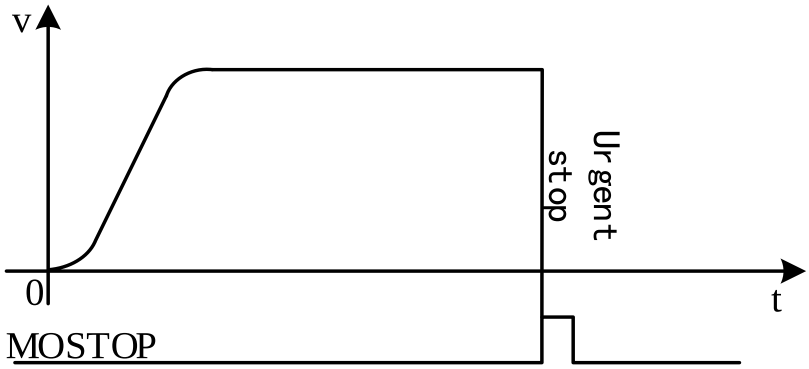
S0: set the pulse stop mode or deceleration distance.

S1: axis number N, N range is from 1 to 20.

● stop mode: urgent stop, slow stop

◆ urgent stop (K-1):

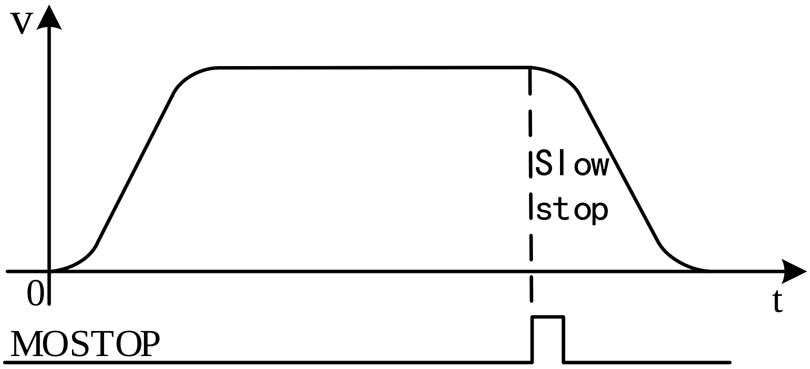
When S0 is K-1, the motor urgent stop.



Note: stop running immediately, it may has mechanical damage.

◆slow stop (K0):

When S0 is K0: decelerate as the time (SD2036+60*(N-1)), decelerate to stop.



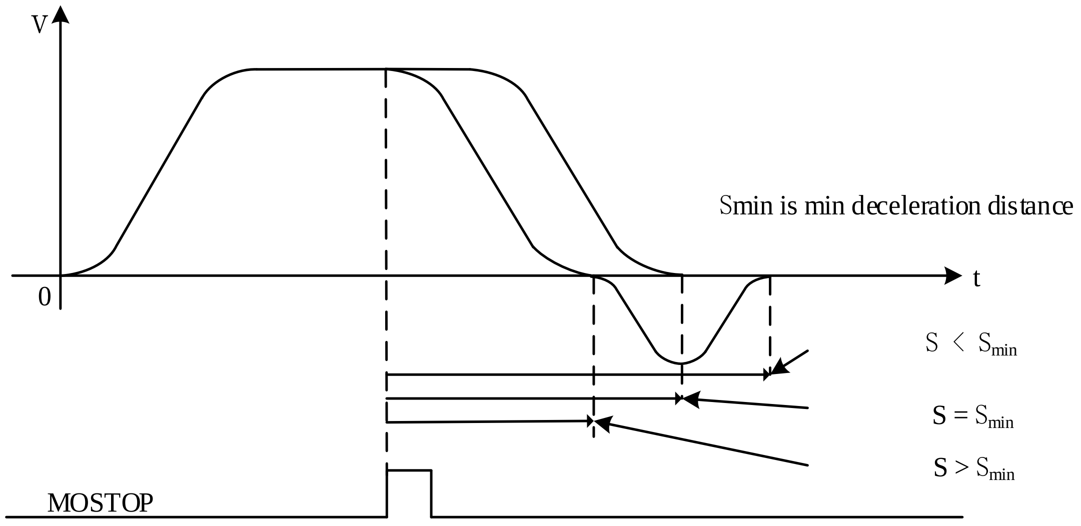
◆fixed-length stop (positive value):

S0 is positive value: slow stop, deceleration distance S is set to positive value.

(1) If S is less than min deceleration distance Smin (calculate from deceleration time SD2036+60*(N-1)), forward decelerate to stop, then reverse run the distance S.

(2) If S is larger than Smin, decelerate to stop. If the distance is too long, the motor will run with constant speed then decelerate to stop.

(3) If S is larger than Smin and over the limit, the motor will take the limit position as the target position.



2-2-5. Continue running [MOGOON]

1. Instruction summary

If the motor stop running, it can make it continue running to the target position.

Continue running [MOGOON]			
16-bit		32-bit	MOGOON
Execution condition	Rising/falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S	Axis number	16-bit constant

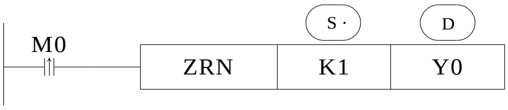
3. Suitable soft component

word	operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
	S									●		

***Note:** D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and action



- When M0 is from OFF to ON, axis S will continue running. The instruction complete flag（SM2003+20*(N-1)）will be ON after motion completed.

  S: axis no. N, the range of N is from 1 to 20.
- Work with MOSTOP, to perform pause function.
- if another instruction is executed for the same axis after executing MOSTOP, it will not work to execute MOGOON.

2-2-6. Synchronous run [MOSYN]

1. Instruction summary

The master axis and slave axis (or high speed count) will synchronous run.

Synchronous run [MOSYN]			
16-bit	-	32-bit	MOSYN
Execution condition	Rising/falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S0	Synchronous speed times	32-bit floating number
S1	Master axis no.	16-bit constant
S2	Slave axis no.	16-bit constant

3. Suitable soft component

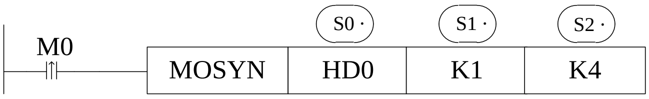
word

Operand	System								Constant		Module
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
S0	●								●		
S1									●		
S2									●		

***Note:** D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and action



● when M0 is from OFF to ON, the master and slave axis position will be locked, and keep synchronous.

S0: synchronous motion speed times (floating number), which is master and slave axis speed.

S1: master axis no.N, the range of N is from 1 to 20 or -1 to -4.

S2: slave axis no.N, the range of N is from 1 to 20.

● the synchronous speed mode depends on S0:

(1) S0 is negative value: slave axis keeps synchronous motion with master axis as reverse speed times |S0|.

(2) S0 is 0, the slave axis bundles with master axis, but the slave axis speed is 0.

(3) S0 is positive value, slave axis keeps synchronous motion with master axis as speed times S0.

● the master axis depends on S1:

(1) S1 is 1~20, the master axis is for pulse output.

(2) S1 is -1~-4, the master axis is high speed counter. -1 is corresponding to high speed counter HSC0, -2 is corresponding to high speed counter HSC2, -3 is corresponding to high speed counter HSC4, -4 is corresponding to high speed counter HSC6, the high speed counter input please refer to PLC high speed counter input terminal.

● it must bundle when master and slave axis stop.

If the master axis stop, the slave axis is running, the slave axis cannot be bundled, the slave axis will stop after the instruction end.

● Modify the synchronous speed times through register (SD2038+60*(N-1)), the times must be floating number.

● when S1 is set to 1~20, it can use with MOTO, MOTOA, MOTOS, MOSTOP to do synchronous motion.

● when S1 is set to -1~-4, it can use with manual pulse generator to do synchronous motion.

● when using manual pulse generator, too large following times will cause motor vibration, please adjust through register SD2059+60*(N-1).

Table 1: parameters (N=1~20)

Address	Definition	Type	Unit	Mark
SD2038+ 60*(N-1)	Synchronous motion speed times	32-bit floating number		Motion axis speed/ target axis speed
SD2044+ 60*(N-1)	Positioning completion width	32-bit integer	Pulse number	Positioning complete threshold, if the difference between setting value and encoder feedback value is less than this value, the running flag is OFF
SD2059+ 60*(N-1)	Auto-tuning filter coefficient	32-bit integer		The range is 0~9999. when using manual pulse generator, too large following times will cause motor

				vibration, please adjust through this parameter.

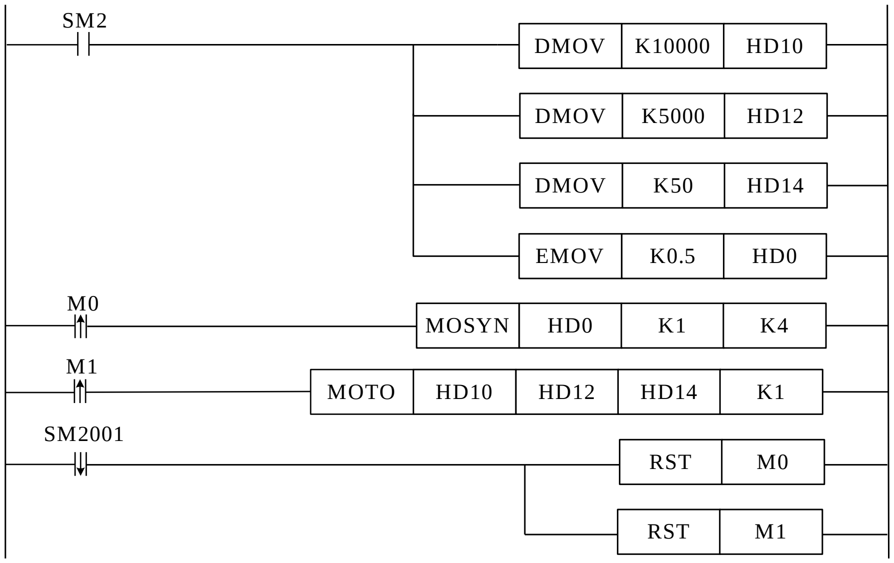
Table 2: state bit (N=1~20)

Address	Definition	Mark
SM2000+20*(N-1)	Servo enable flag	ON: servo enable state
SM2001+20*(N-1)	Running flag	ON: pulse is outputting
SM2004+20*(N-1)	Axis error flag	ON: error

Application 1

Bundle the master axis no.1 motor and slave axis no.4 motor with MOSYN instruction, the slave axis will follow the master axis with the speed 5000Hz and run 10000 pulses. The acceleration and deceleration tiem is 50ms. The slave axis speed is 0.5 times of master axis.

The ladder chart:



The PLC starts running. When SM2 is ON, the pulse numbers, speed, acceleration and deceleration time, synchronous speed times are stored in related registers.

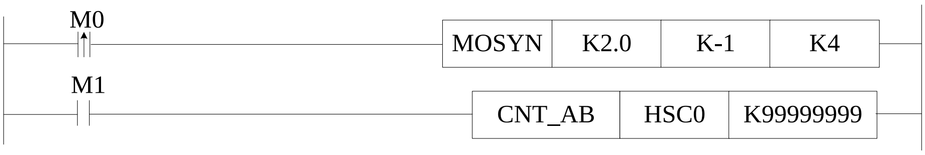
Servo enabled, M0 is from OFF to ON, bundle the master axis and slave axis.

When M1 is from OFF to ON, PLC executes the instruction MOTO.

When the pulse sending completed, the running flag SM2001 reset, the related coils reset.

Application 2

Bundle the high speed counter HSC0 with slave axis No.4. the slave axis can follow the manual pulse generator motion. The slave axis speed is 2 times of main axis.
The ladder chart:



Servo is enabled, M0 is from OFF to ON, bundle the high speed counter HSC0 with slave axis No.4. M1 is from OFF to ON, HSC0 starts to count. At this time, rotate the manual pulse generator, axis 4 will follow its motion.

2-2-7. Release synchronous run [MOUSYN]

1. Instruction summarize

Release the synchronous motion between master axis and slave axis (or high speed count).

Release synchronous run[MOUSYN]			
16-bit		32-bit	MOUSYN
Execution condition	Rising/falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

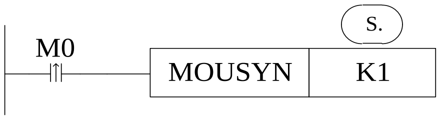
Operand	Function	Type
S	Slave axis number	16-bit constant

3. Suitable soft component

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
	S									●		

***Note:** D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and action



- when M0 is from OFF to ON, release the two axes synchronous running.
  S: slave axis number N, the range of N is from 1 to 20;
- please release the synchronous running when master axis and slave axis stop.
- MOSTOP urgent stop mode can stop the slave axis running and release the synchronous bundling. At this time, the slave axis will urgent stop and the master slave will keep running. This mode has speed sudden change. This method is not recommended to use regularly.

2-2-8. Write in present position [MOWRITE]

1. Instruction summarize

This instruction can modify the motion axis present absolute position. It is used to correct the position.

Write in present position [MOWRITE]			
16-bit		32-bit	MOWRITE
Execution condition	Rising / falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

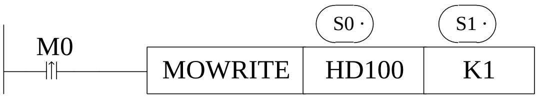
Operand	Function	Type
S0	Present position	32-bit integer
S1	Axis number	16-bit constant

3. Suitable soft component

Word	Operand	System								Constant	Module	
		D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
	S0	●										
	S1									●		

***Note:** D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.
M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and action



● when M0 is from OFF to ON, modify the axis present absolute position (SD2008+60*(N-1)) to S0.

S0: the motion axis present absolute position

S1: slave axis number N, the range of N is from 1 to 20

● this instruction is invalid for multi-speed motion and synchronous motion.

● when modifying the present position (SD2008+60*(N-1)), the present displacement quantity (SD2006+60*(N-1)), target position setting pulse numbers (HSD100+20*(N-1)), target position feedback pulse numbers (HSD104+20*(N-1)), present displacement pulse numbers (HSD108+20*(N-1)) will change.

● when the motor enabled, it can clear the parameters in below table 1 and 2.

● when the axis present position （SD2008+60*(N-1)）is larger than 2^{24}（16777216）, bad precision and motion vibration will appear, at this time, please reset the present position by MOWRITE instruction, present displacement （SD2006+60*(N-1)）will not be affected.

● there are four methods to change the present position:

① return to origin through (SM2014+20*(N-1)), (SM2015+20*(N-1)), the present position will change.

② when PLC stop, modify the target position feedback pulse numbers (+20*(N-1)), the present position will change

③ when PLC is running, execute MOWRITE can modify (HSD104+20*(N-1)), the present position will change.

④ when PLC is running, servo does not enable, the servo position is chagned by external force, register (HSD104+20*(N-1)) will change. When the servo enabled, this value is valid.

Related register

Table 1: state quantity parameters (N=1~20)

Address	Definition	Type	Unit	Mark
SD2006+ 60*(N-1)	Present displacement quantity	32-bit integer	Pulse number	The displacement relative to the last stop position
SD2008+ 60*(N-1)	Present position	32-bit integer	Pulse number	The coordinate position, calculated from target position feedback pulse numbers

Table 2: self preserving state (N =1~20)

Address	Definition	Type	Unit	Mark
HSD100+ 20*(N-1)	Target position setting pulse numbers	64-bit integer	Encoder count	Relative zero position
HSD104+	Target position	64-bit	Encoder	Relative zero position

20*(N-1)	feedback pulse numbers	integer	count	
HSD108+ 20*(N-1)	Present displacement pulse numbers	64-bit integer	Encoder count	Single time motion instruction displacement quantity

2-2-9. Read present position [MOREAD]

1. Intruction summarize

This instruction can read the present absolute position.

Read present position [MOREAD]			
16-bit		32-bit	MOREAD
Execution condition	Rising/falling edge	Suitable model	XDC, XDE, XG
Hardware	-	Software	-

2. Operand

Operand	Function	Type
S0	Read present position	32-bit integer
S1	Axis number	16-bit constant

3. Suitable soft component

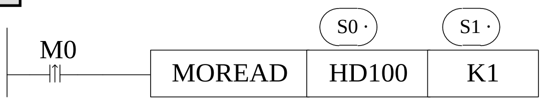
Word

Operand	System								Constant	Module	
	D*	FD	TD*	CD*	DX	DY	DM*	DS*	K/H	ND	QD
S0	●										
S1									●		

***Note:** D includes D, HD; TD includes TD, HTD; CD includes CD, HCD, HSCD, HSD; DM includes DM, DHM; DS includes DS, DHS.

M includes M, HM, SM; S includes S, HS; T includes T, HT; C includes C, HC.

Function and action



● when M0 is from OFF to ON, MOREAD will refresh the state parameter in SD register and read axis S1 present absolute position (SD2008+60*(N-1)) in S0 register.

S0: the register address to store the present absolute position
S1: slave axis number N, the range of N is from 1 to 20

- EMOV or DMOV instruction can transfer the data of SD register (dword).

2-3. Function teaching

2-3-1. Speed mode

X-NET fieldbus speed mode takes speed as control target, and meets the motion requirements by setting the speed constantly.

Speed mode related coil and registers

Address	Definition	Type	Initial value	Preset value	Notes
SFD3000 +60*(N-1)	Running mode	16-bit integer	0	3	0: position control with motion planning 3: user-defined motion planning
SFD3029 +60*(N-1)	Position feedback deviation upper limit	16-bit integer	2500	-1	Positive integer: deviation upper limit -1: ignore the deviation value

Address	Definition	Type	Unit	Notes
SD2008+ 60*(N-1)	Present position	32-bit integer	Pulse numbers	Absolute position, converted by target position feedback pulse numbers
SD2010+ 60*(N-1)	Present speed	32-bit integer	Pulse numbers/ second	Calculated by feedback value
SD2012+ 60*(N-1)	Instantaneous speed setting		Pulse numbers/ second	Speed setting value of single control period
SD2032+ 60*(N-1)	Speed setting	32-bit integer	Pulse numbers/ second	

Servo parameters

Servo parameters	Function description	Default value	10-axis setting value	20-axis setting value

P0-01	Control mode 8: fieldbus torque mode 9: fieldbus speed mode 10: fieldbus position mode	6	9	9

Application notes:

- Set register SFD3029+60*(N-1) to -1, otherwise, SD2002+60*(N-1) will report the position offset 20006
- Please set register SD2032+60*（N-1）to control the motor speed in speed mode, the speed setting is not related to SD2034+60*（N-1）and SD2036+60*（N-1）

2-3-2．Torque mode

X-NET fieldbus torque mode takes torque as control target. User must set the torque value for actual application. The torque value is related to the speed and load.

Torque mode related coil and register

Address	Definition	Type	Initial value	Preset value	Notes
SFD3000 +60*(N-1)	Running mode	16-bit integer	0	3	0: position control with motion planning 3: user-defined motion planning
SFD3029 +60*(N-1)	Position feedback deviation upper limit	16-bit integer	2500	-1	Positive integer: deviation upper limit -1: ignore the deviation value

Address	Definition	Type	Unit	Notes
SD2006+ 60*(N-1)	Present displacement	32-bit integer	Pulse numbers	Displacement relative to the last time stop position, which is the displacement in this instruction
SD2008+ 60*(N-1)	Present position	32-bit integer	Pulse numbers	Absolute position, calculated by target position feedback pulse numbers
SD2010+ 60*(N-1)	Present speed	32-bit integer	Pulse numbers/ second	Calculated by feedback value
SD2012+ 60*(N-1)	Instantaneous speed setting		Pulse numbers/ second	Speed setting value of single control period
SD2020+ 60*(N-1)	Present torque	Floating number	N.m	Servo P7-02 set to 3, when SD2029+60*（N-1）set to 1，

				SD2020+60*（N-1） shows present torque
SD2024+ 60*(N-1)	Torque setting	32-bit integer	1/1000 rated	Effective mode: Servo P0-01=8（torque mode）always be effective, servo P0-01=9 or 10（speed or position mode）: be effective when SD2028+60*（N-1）=1
SD2026+ 60*(N-1)	Reverse torque setting	32-bit integer	1/1000 rated	Be effective when SD2028=1 in position or speed mode. At this time, servo value P3-28, P3-29 are invalid, Uneffective in torque mode.
SD2028+ 60*(N-1)	Torque control mode setting	16-bit integer		Servo P0-01=9 or 10（speed or position mode）：when SD2028+60*（N-1）=1, SD2024+60*（N-1）, SD2026+60*（N-1）are effective, at this time, servo value P3-28, P3-29 are uneffective.
SD2029+ 60*(N-1)	Torque feedback enable bit	16-bit integer		0: uneffective 1：SD2020+60*（N-1） is present torque（servo P7-02 must set to 3）
SD2032+ 60*(N-1)	Speed limit	32-bit integer	Pulse numbers/ second	

Servo parameters

Servo parameters	Function description	Default value	10-axis setting value	20-axis setting value
P0-01	Control mode 8: fieldbus torque mode 9: fieldbus speed mode 10: fieldbus position mode	6	8	8

1. Application of fieldbus torque mode

(1) Servo parameters setting please refer to chapter 2-6-2. P0-01 set to 8(torque mode), P7-02 set to 3.

(2) Set SFD3000+60*（N-1）=3，SFD3029+60*（N-1）= -1 ignore the deviation，enable SM2010+20*（N-1），SM2000+20*（N-1）show ON and servo drive shows“RUN” means enable is successful. Set SD2029+60*（N-1）=1，at this time, SD2020+60*（N-1） is effective.

(3) Set present speed limit SD2032+60*（N-1）. SD2024+60*（N-1）initial value is 0，motor has no force, please adjust this value according to the load. SD2020+60*（N-1）can shows the present torque value.

SD2020	0.02337563	Floating number	Present torque	Show present torque
SD2029	1	Single word	Torque feedbak enable bit	Set ON the enable bit
SD2032	5000	Double words	Axis 1 speed setting	Set the speed limit
SD2024	1000	Double words	Torque setting	Set present torque
SD2026	0	Double words	Reverse torque setting	Torque mode is uneffective
SD2010	5000	Double words	Axis 1 present speed	Real-time speed feedback
SD2012	5000	Double words	Axis 1 instantaneous speed setting	Real-time speed setting
SD2028	0	Single word	Torque control mode setting	Torque mode uneffective

2. The meaning of reverse torque and torque setting

Torque setting SD2024+60*（N-1）:

Torque mode: the setting value is present torque (thousandth rated), the positive and negative value means the forward and reverse torque limit.

Speed mode or positon mode: such as torque control mode SD2028+60*（N-1）=1, the setting value is just forward torque limit (thousandth rated).

Reverse torque setting SD2026+60*（N-1）:

Torque mode: uneffective

Speed mode or position mode: such as torque control mode SD2028+60*（N-1）=1, the setting value is just reverse torque limit (thousandth rated).

2-3-3. Back to origin

It no needs to make program for X-NET fieldbus returning to the origin, please just set the close signal (SFD3036+60*(N-1)), origin signal (SFD3037+60*(N-1)), high speed of returning to the origin VH(SFD3040+60*(N-1)), low speed of returning to the origin VL(SFD3042+60*(N-1)), creeping speed (SFD3044+60*(N-1)). When the servo enable signal is ON, the returning to the origin operation can be done through positive returning to origin coil (SM2014+20*(N-1)) and reverse returning to origin coil (SM2015+20*(N-1)).

The parameters table:

Address	Definition	Type	Unit	Initial value	Notes
SFD3034 +60*(N-1)	Min position limit terminal	16-bit integer		0xFF	Appoint the X terminal no., 0xFF is no terminal, negative number represents inverse logic, the inverse logic of X0 is -30000

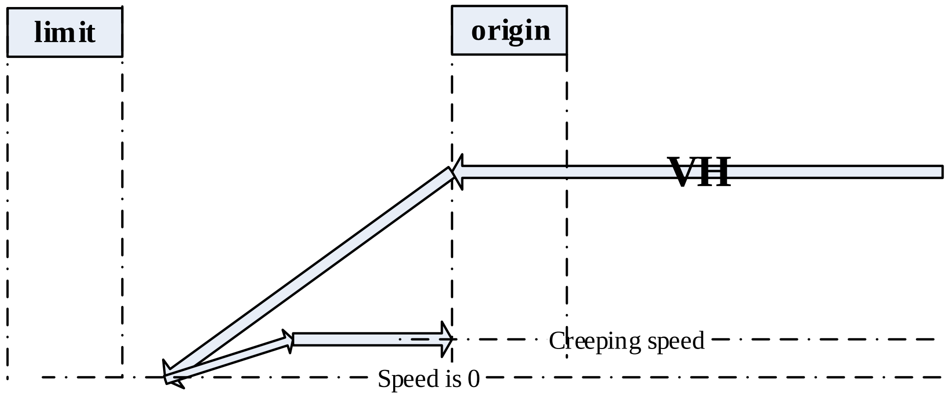
SFD3035 +60*(N-1)	Max position limit terminal	16-bit integer		0xFF	Appoint the X terminal no., 0xFF is no terminal, negative number represents inverse logic, the inverse logic of X0 is -30000
SFD3036 +60*(N-1)	Close signal terminal	16-bit integer		0xFF	Appoint the X terminal no., 0xFF is no terminal, negative number represents inverse logic, the inverse logic of X0 is -30000
SFD3037 +60*(N-1)	Origin terminal	16-bit integer		0xFF	Appoint the X terminal no., 0xFF is no terminal, negative number represents inverse logic, the inverse logic of X0 is -30000
SFD3038 +60*(N-1)	Return to origin mode	16-bit integer		0	0: no Z phase mode. search the close point with VH speed, then search the origin point with VL speed, decelerate after finding the origin point, then search the origin point in reverse direction with creeping speed 2: Z phase mode. search the close point with VH speed, then search the origin point with VL speed, decelerate after finding the origin point, then search the origin point in reverse direction with creeping speed, search the servo encoder Z phase in forward direction 10: hard limit return mode. Search the origin in reverse direction with –VH speed after touching the ±hard limit, the speed becomes VH after touching the origin, the following action is same to mode 0 12: hard limit return to Z phase mode. Search the origin in reverse direction with –VH speed after touching the ±hard

					limit, the speed becomes VH after touching the origin, the following action is same to mode 2.
SFD3040 +60*(N-1)	Return speed VH	32-bit integer	Pulse/ second	0	
SFD3042 +60*(N-1)	Return speed VL	32-bit integer	Pulse/ second	0	
SFD3044 +60*(N-1)	Creeping speed	32-bit integer	Pulse/ second	0	

There are three modes of return to origin: no Z phase signal mode, Z phase signal mode, hard limit return mode.

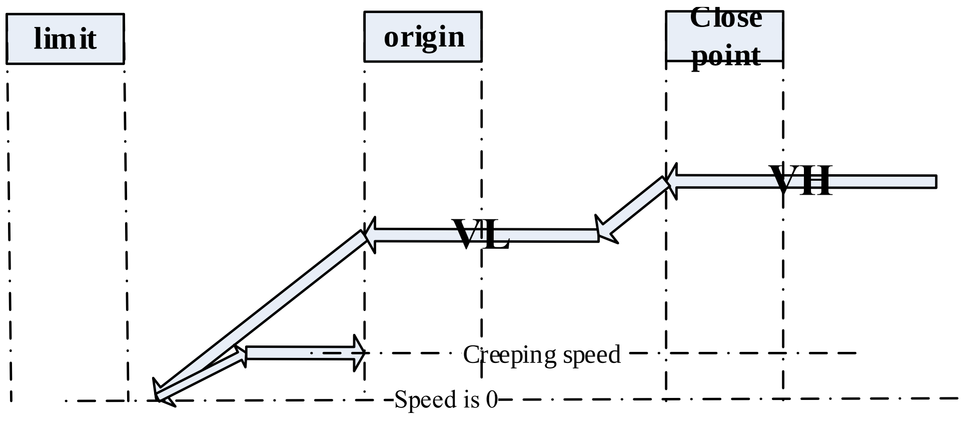
1. No Z phase signal mode

- Return to the origin point in reverse direction, no close signal:



Motion description: the motor returned to the origin with speed VH, when it touched the falling edge of origin signal, the speed become zero, it returned to the origin in reverse direction with creeping speed, and stop moving when touched rising edge of origin signal.

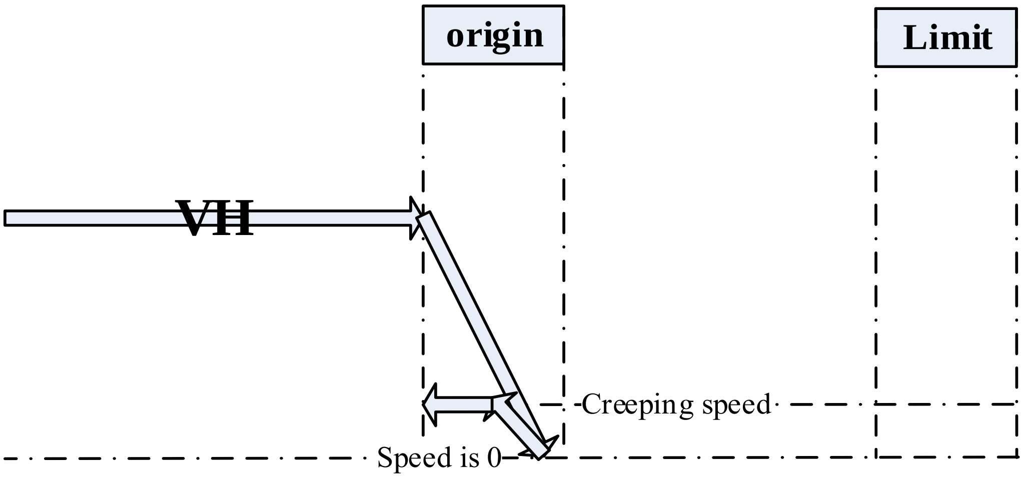
- Return to origin in reverse direction, there is close signal:



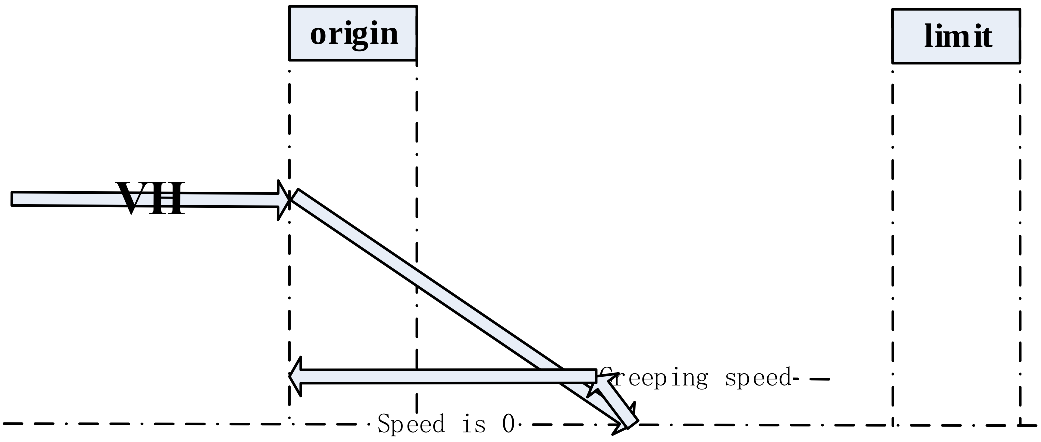
Motion description: the motor return to origin with speed VH, it touched close signal and the speed changed to VL and continued returning to origin, the speed become 0 when it touched falling edge of origin signal, it returned to origin in reverse direction with creeping speed, and stop moving when touched rising edge of origin signal.

- Return to the origin in forward direction, no close signal:

If the origin signal is long:



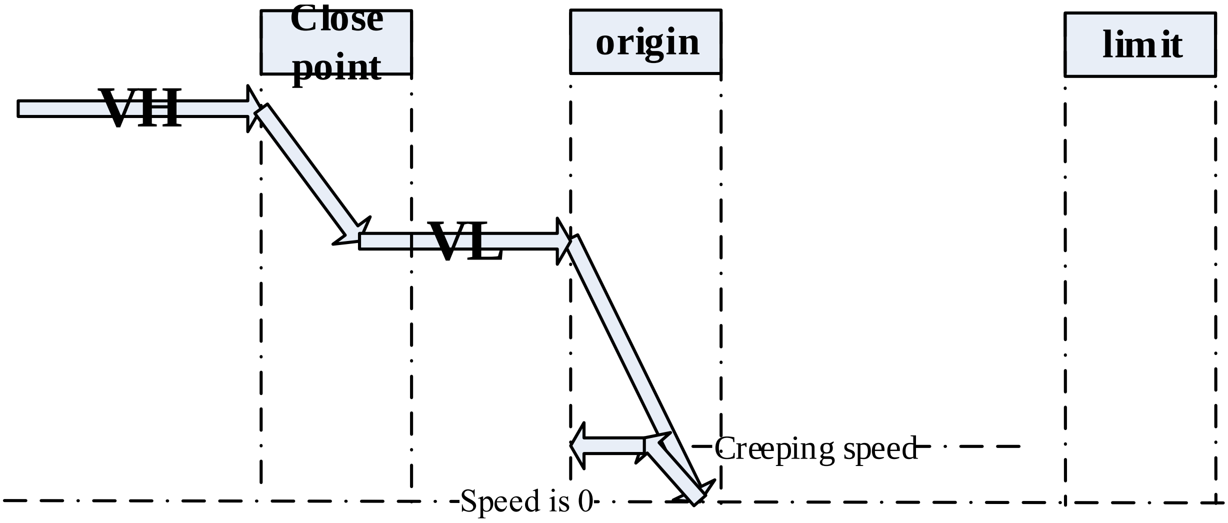
If the origin signal is short:



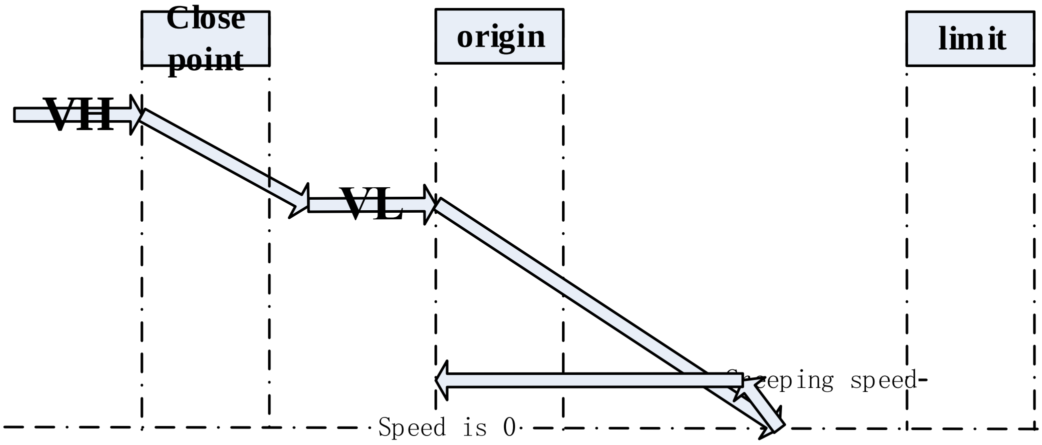
Motion description: the motor returned to origin with VH speed, it touched the rising edge of origin signal and the speed become 0 and returned to origin in reverse direction with creeping speed, it touched the falling edge of origin signal and stop returning.

- Return to origin point in forward direction, there is close signal:

If the origin signal is long:



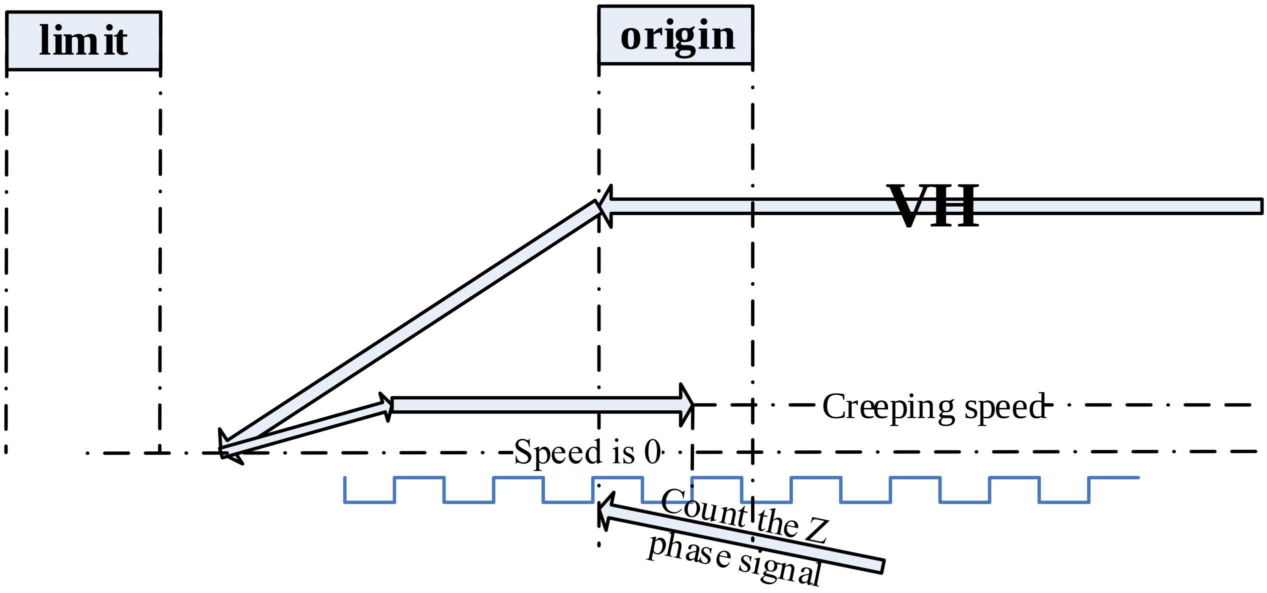
If the origin signal is short:



Motion description: the motor returned to origin with speed VH, it touched close signal and the speed become VL and continued returning to origin, it touched rising edge of origin signal and the speed become 0 and returned to origin in reverse direction with creeping speed, it stop returning when touching the falling edge of origin signal.

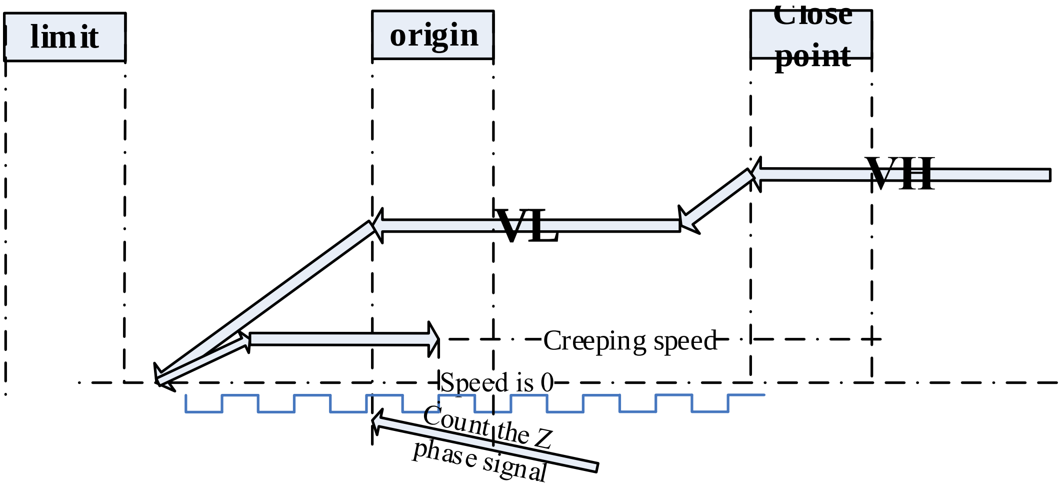
2. Z phase signal mode:

- Return to origin in reverse direction, no close signal:



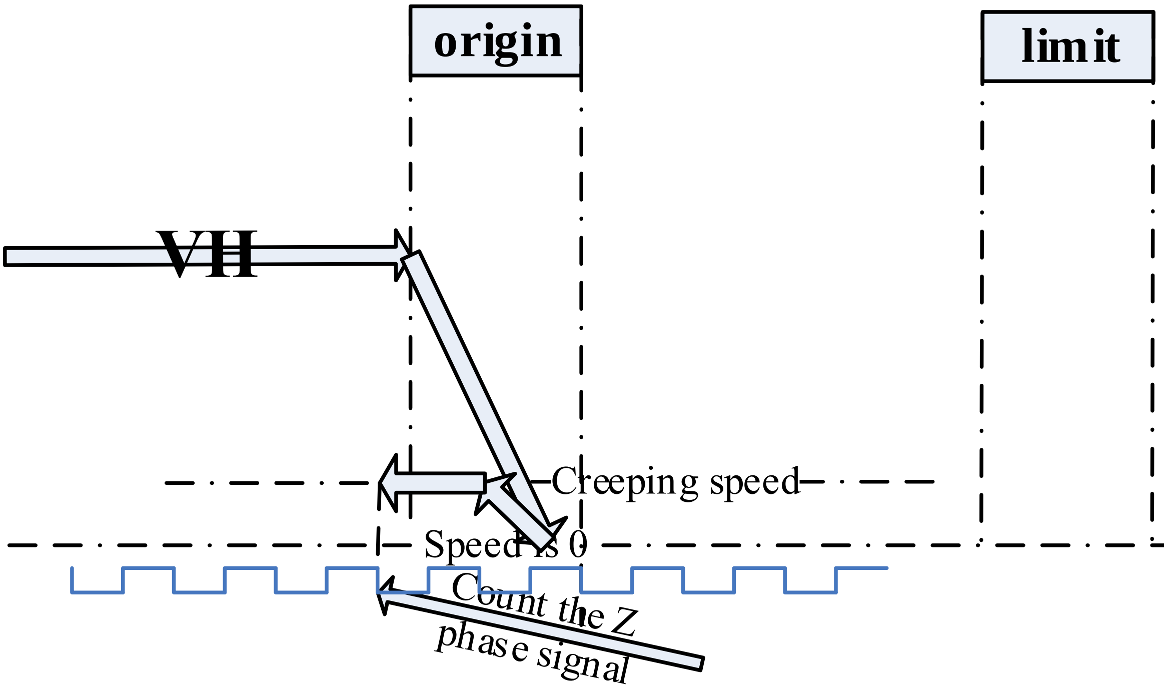
Motion description: the motor returned to origin with VH speed, it touched falling edge of origin signal and the speed become 0, it returned to origin in reverse direction with creeping speed, it searched Z phase signal when touching the rising edge of origin signal, it stop returning when found the Z phase signal.

- Return to origin in reverse direction, there is close signal



Motion description: the motor returned to origin with VH speed, it touched close signal and the speed become VL and continued returning to origin, the speed become 0 when it touched falling edge of origin signal, then it returned to origin in reverse direction with creeping speed, it searched servo Z phase signal when it touched rising edge of origin signal, it stop moving after found the Z phase signal.

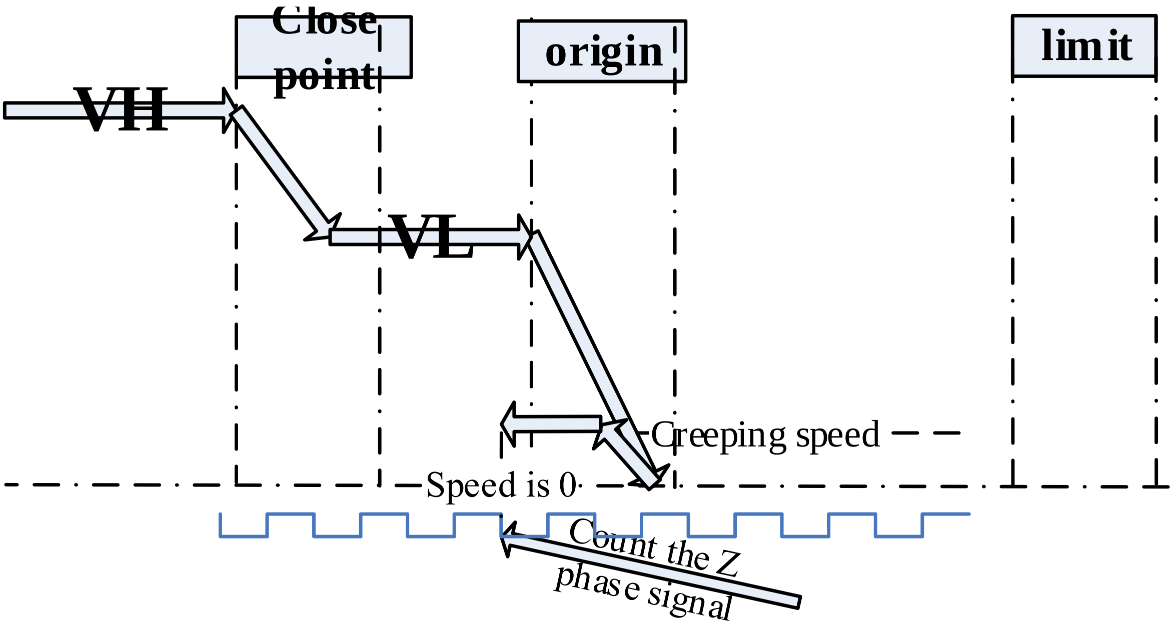
- Return to origin in forward direction, no close signal:



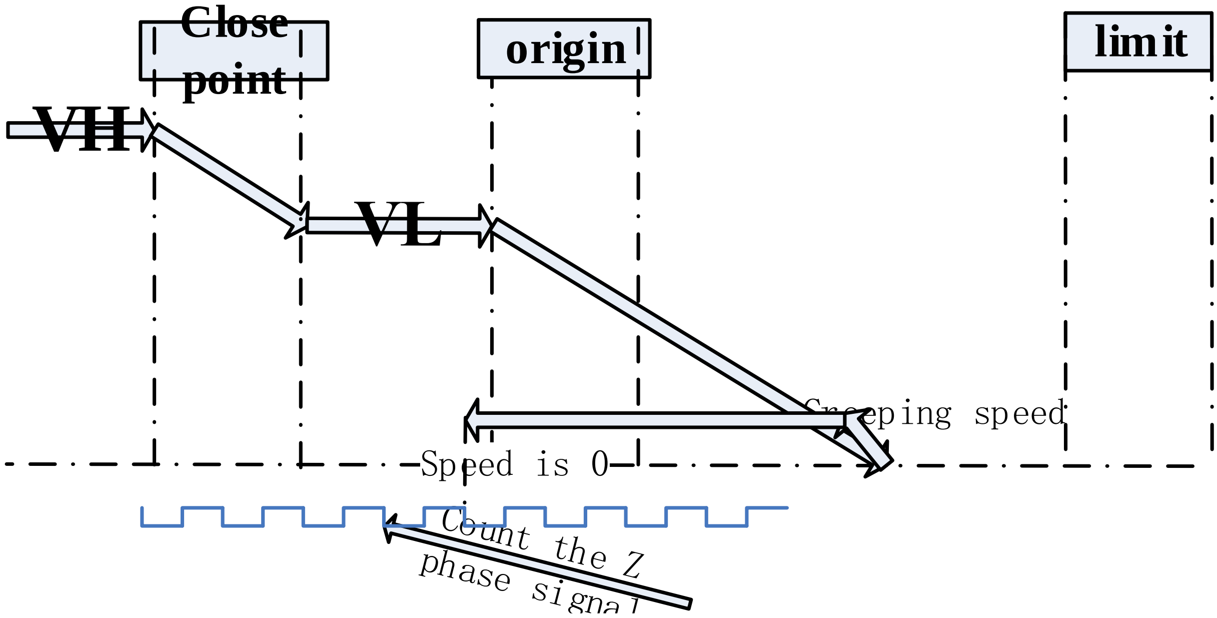
motion description: the motor returned to origin with VH speed, it touched rising edge of origin signal and the speed become 0, then it returned to origin in reverse direction with creeping speed, it searched servo Z phase signal when it touched falling edge of origin signal, it stop moving after found the Z phase signal.

- Return to origin in forward direction, there is close signal:

If the origin signal is long:



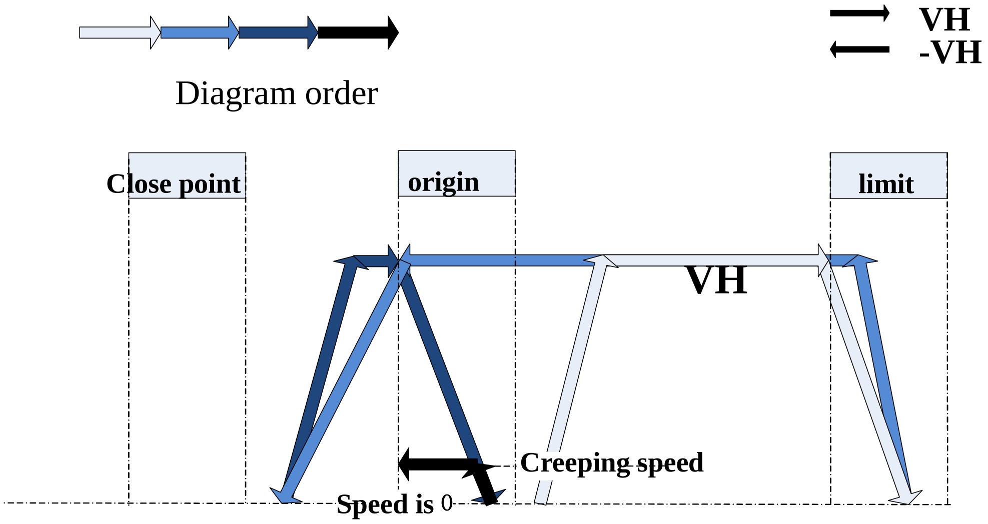
If the origin signal is short:



motion description: the motor returned to origin with VH speed, it touched close signal and the speed become VL and continued returning, the speed become 0 when it touched rising edge of origin signal and returned to origin in reverse direction with creeping speed, it searched servo Z phase signal when it touched falling edge of origin signal, it stop moving when it found the Z phase signal.

3. hard limit return mode

(1) return to the origin in forward direction, the initial position exsits on the right of the origin



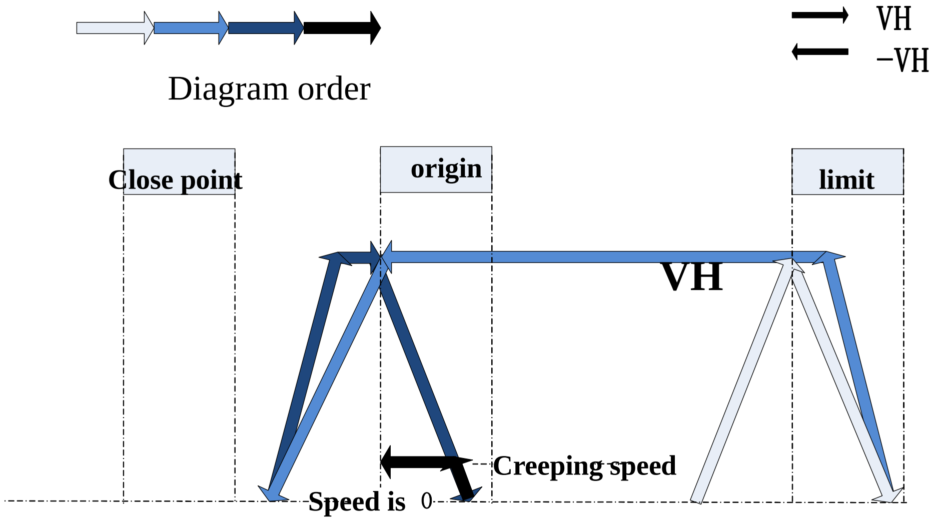
⇨ after VH high speed touching the max limit rising edge, decelerate then accelerate in reverse direction

➡ accelerate to –VH in reverse direction, decelerate then accelerate in reverse direction after touching the origin falling edge

➡ accelerate to VH in reverse direction, decelerate then accelerate in reverse direction after touching the origin rising edge

➡ acelerate to reverse creeping speed, then touch the origin falling edge and decelerate to 0 to finish the returning to origin action

(2) return to the origin in forward direction, the initial position is on the right of the origin, and just accelerate to the max limit



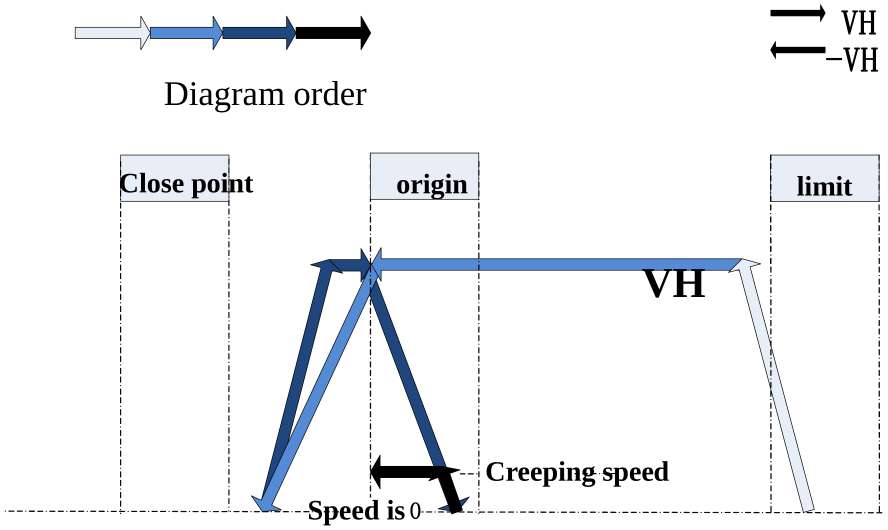
⇨ accelerate to high speed VH, just touch the the rising edge of max limit, then decelerate and accelerate in reverse direction

➡accelerate to –VH in reverse direction, touch the rising edge of origin, then decelerate and accelerate in reverse direction

➡ accelerate to VH in reverse direction, touch the rising edge of origin, then decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the falling edge of origin, decelerate to 0 to finish the back to origin action

(3) return to origin in forward direction, the initial position is just on the max limit



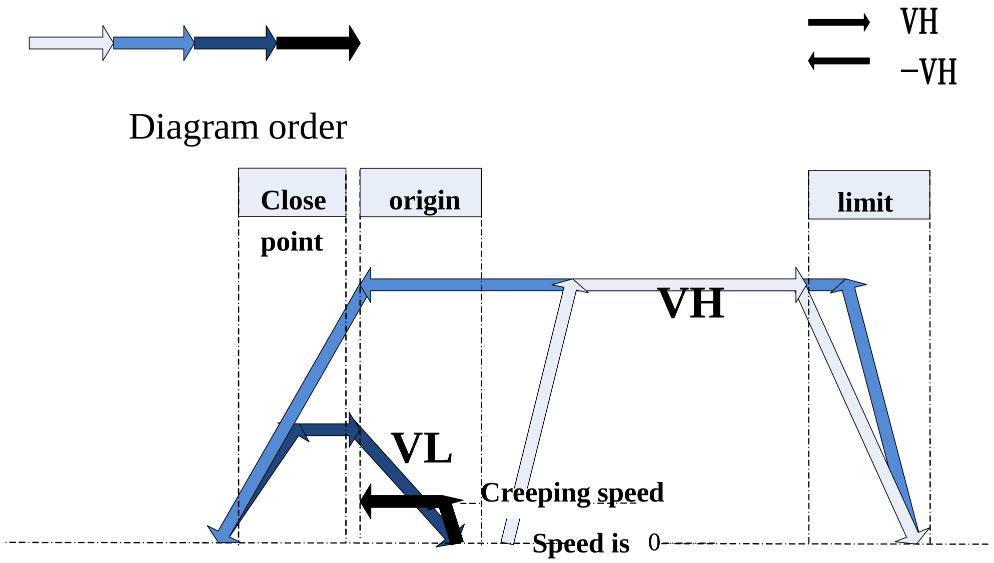
⇨ accelerate to –VH and move to high speed in reverse direction

⇨ after touching the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to VH in reverse direction, touch the rising edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the falling edge of origin, decelerate to 0 to finish the return to origin action

(4) return to origin in forward direction, the origin signal is too short and near the close signal, touch the close signal



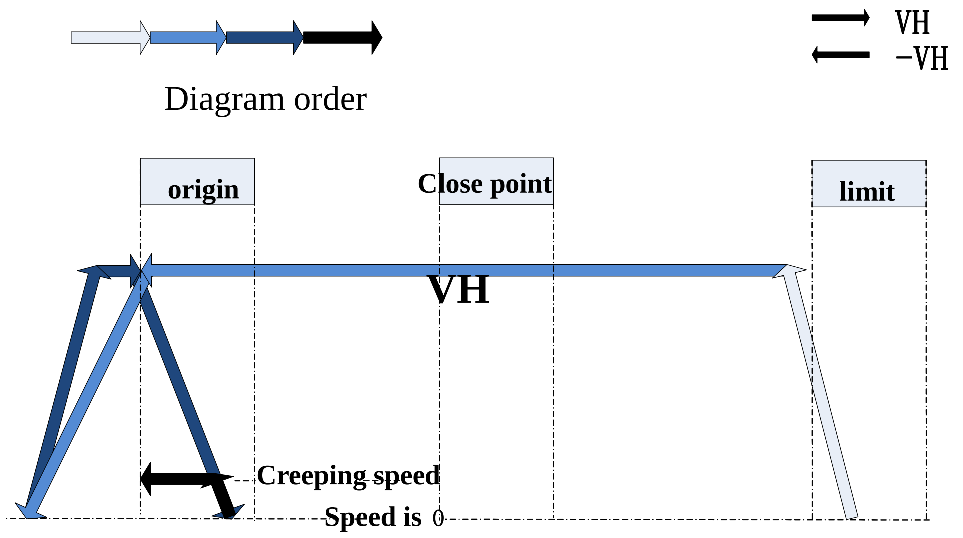
⇨ high speed VH touch the rising edge of max limit, decelerate and accelerate in reverse direction

⇨ accelerate to –VH in reverse direction, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerating segment touches the rising edge of close point, the speed becomes VL, keep moving with speed VL, decelerate and accelerate in reverse direction after touching the rising edge of origin

➡ accelerate to reverse creeping speed, touch the falling edge of origin, decelerate to 0 to finish the back to origin action

(5) return to origin in forward direction, the close signal is between origin and max limit, initial position is on the max limit



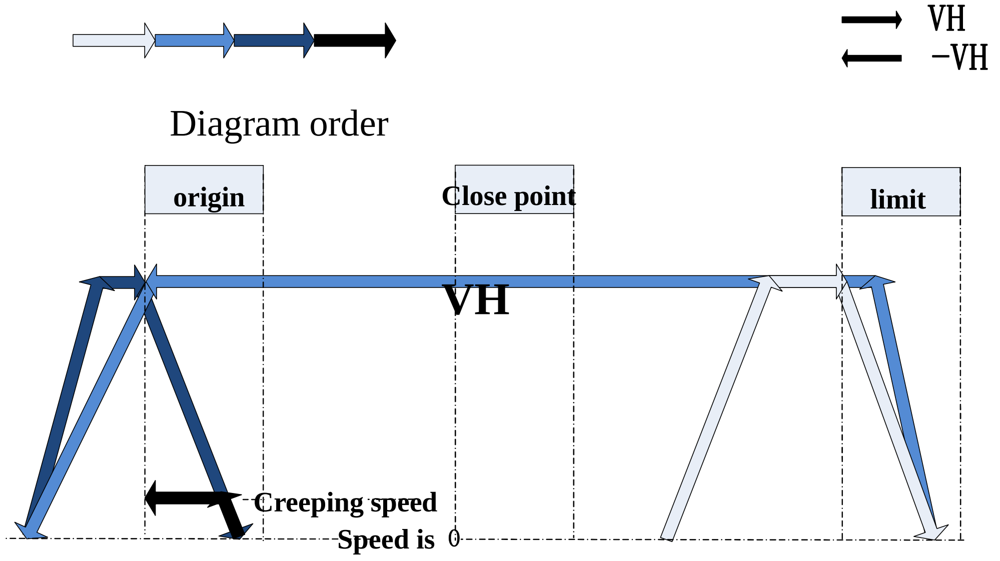
⇨ accelerate to –VH and high speed move in reverse direction

➡ decelerate and accelerate in reverse direction after touching the falling edge of origin

➡ reverse accelerate to VH, decelerate and accelerate in reverse direction after touching the rising edge of origin

➡ accelerate to reverse creeping speed, touch the falling edge of origin, decelerate to 0 to finish the back to origin action

(6) return to origin in forward direction, close point signal is between origin and max limit, inintial position is between close point and origin.



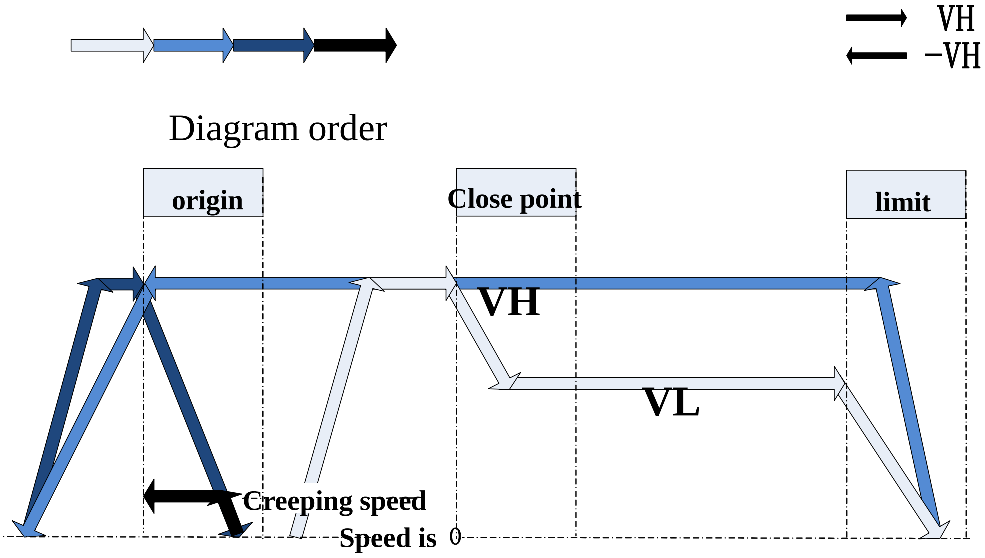
⇨ VH high speed touches rising edge of max limit, decelerate and accelerate in reverse direction

➡ reverse accelerate to –VH, touch falling edge of origin, accelerate in reverse direction

➡ reverse accelerate to VH, touch the rising edge of origin, decelerate and accelearte in reverse direction

➡ accelerate to reverse creeping speed, touch the falling edge of origin, decelerate to 0 to finish the back to origin action

(7) return to origin in forward direction, close point signal is between origin and max limit, initial position is between origin and close point



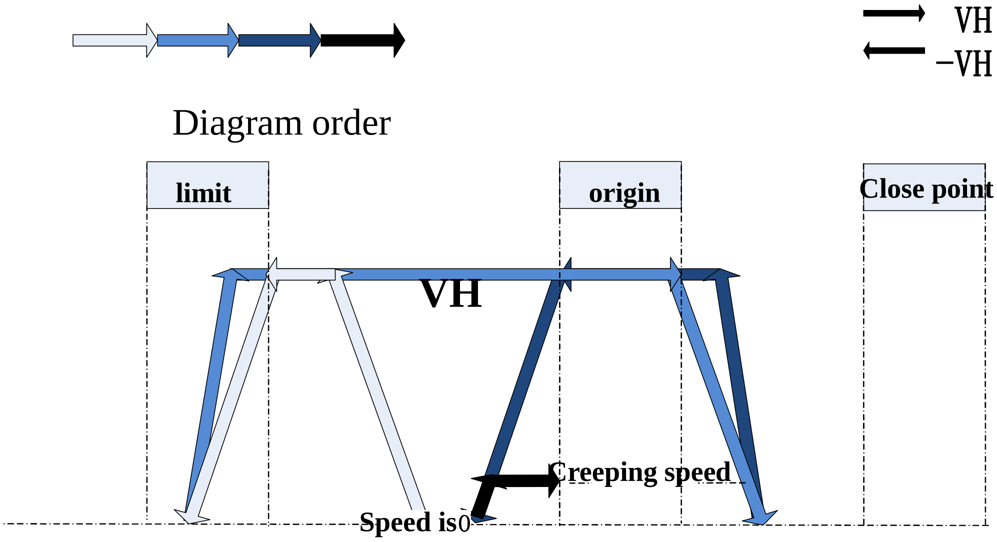
➡ VH high speed touch the rising edge of close point signal, decelerate to VL, then touch the rising edge of max limit, decelerate and accelerate in reverse direction

➡ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerate to VH, touch the rising edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the falling edge of origin, decelerate to 0 to finish the back to origin action

(8) reverse return to origin, initial position is on the left of origin



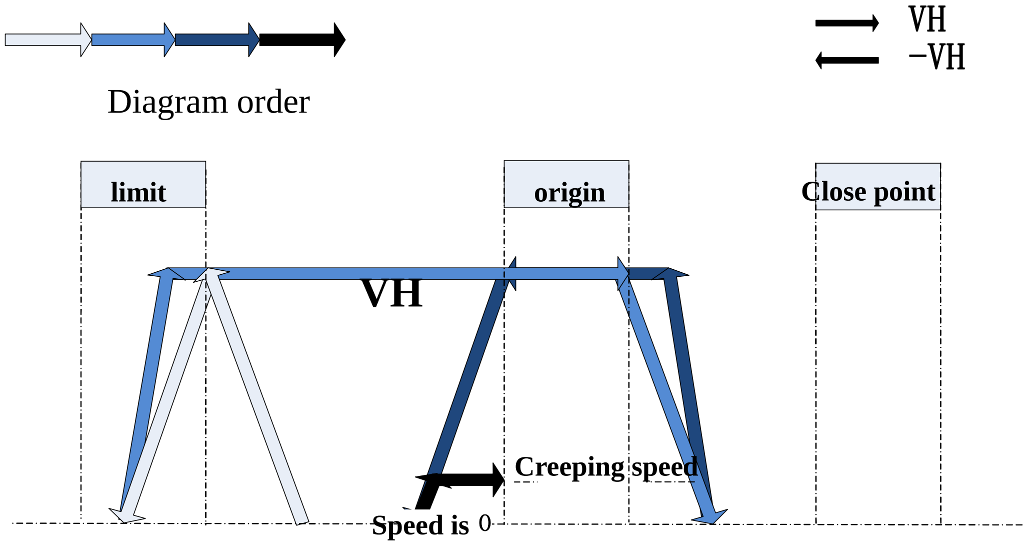
⇨ -VH high speed touch the rising edge of min limit, decelerate and accelerate in reverse direction

⇨ reverse accelerate to VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

⇨ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

⇨ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

(9) reverse return to origin, initial position is on the left of origin, and just accelerate to the min limit



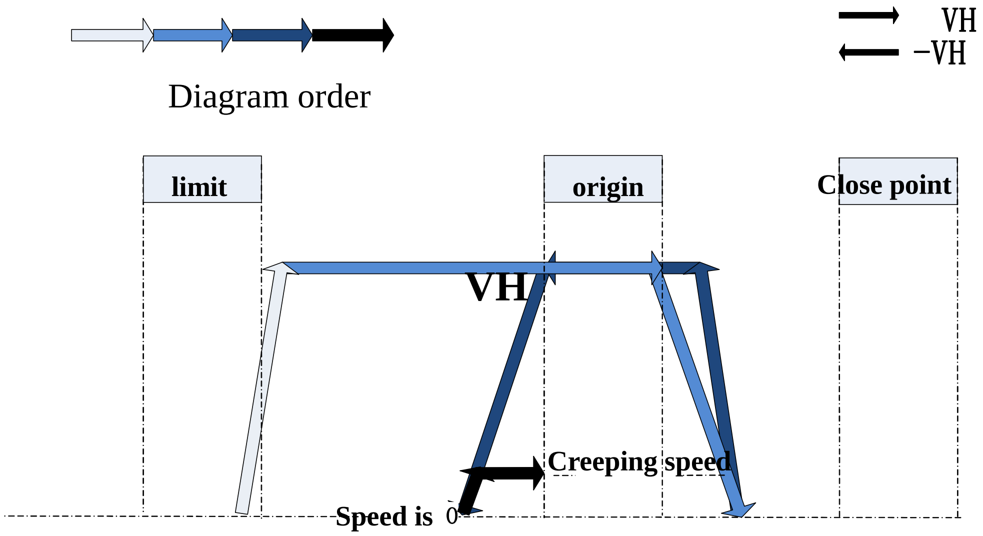
⇨ accelerate to high speed –VH, just touch the rising edge of min limit, decelerate and accelerate in reverse direction

⇨ reverse accelerate to VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

⇨ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

(10) reverse return to origin, initial position is just on the min limit



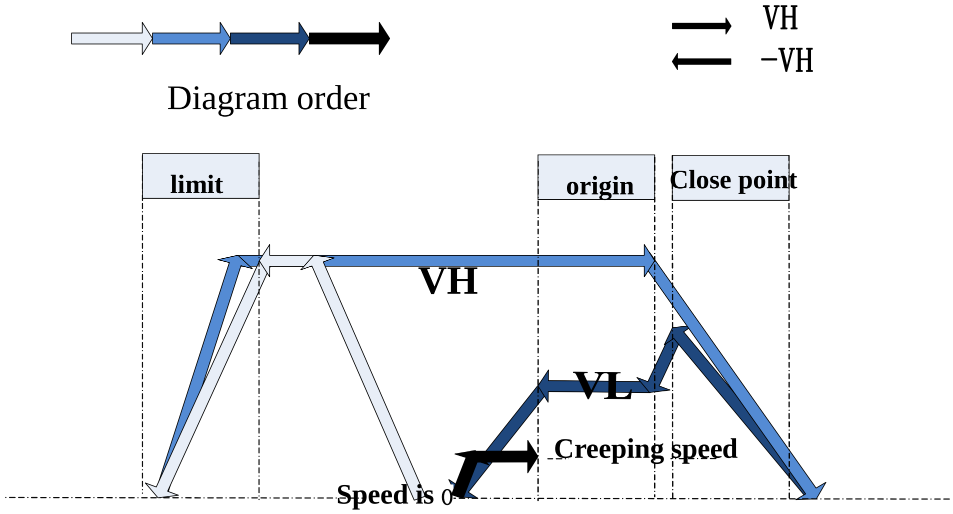
⇨ accelerate to VH and move in reverse direction

➡ touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

(11) reverse return to origin, initial position is on the left of origin, touch the close point signal



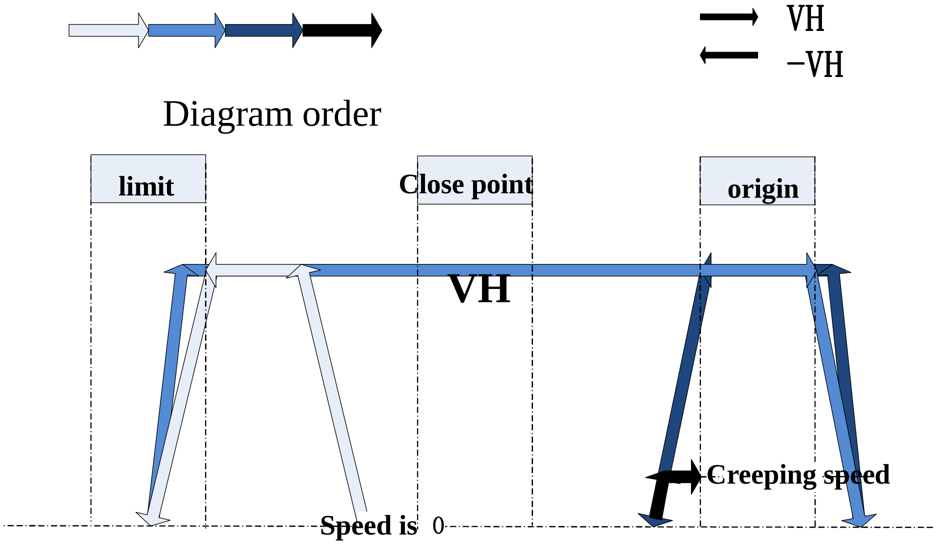
⇨ -VH high speed touch rising edge of min limit, decelerate and accelerate in reverse direction

➡ reverse accelerate to VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerating segment touches the falling edge of close point, the speed becomes –VL, keep the speed –VL and moving in reverse direction, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

(12) reverse return to origin, the initial position is between min limit and close point



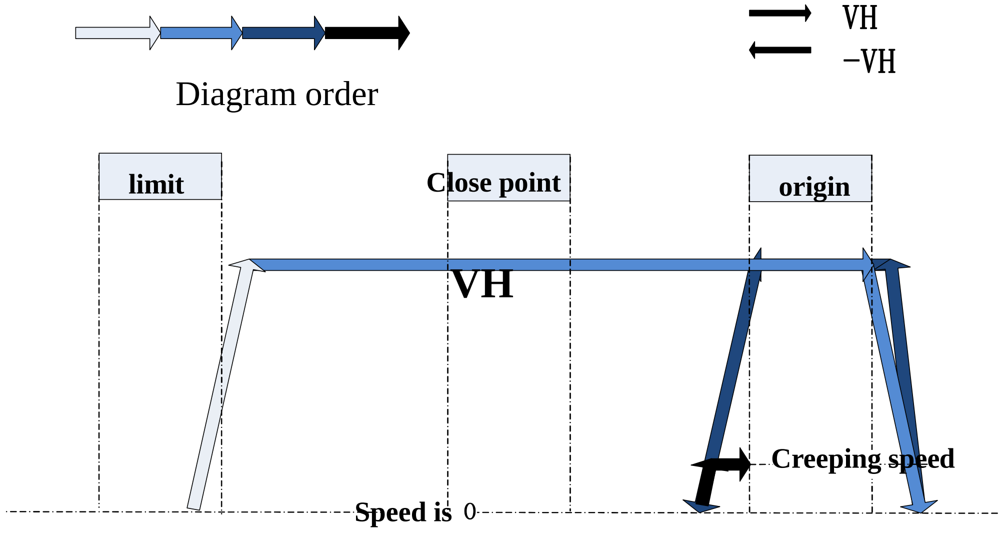
➡ -VH high speed touches rising edge of min limit, decelerate and accelerate in reverse direction

➡ reverse accelerate to VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

(13) reverse return to origin, initial position is on the min limit



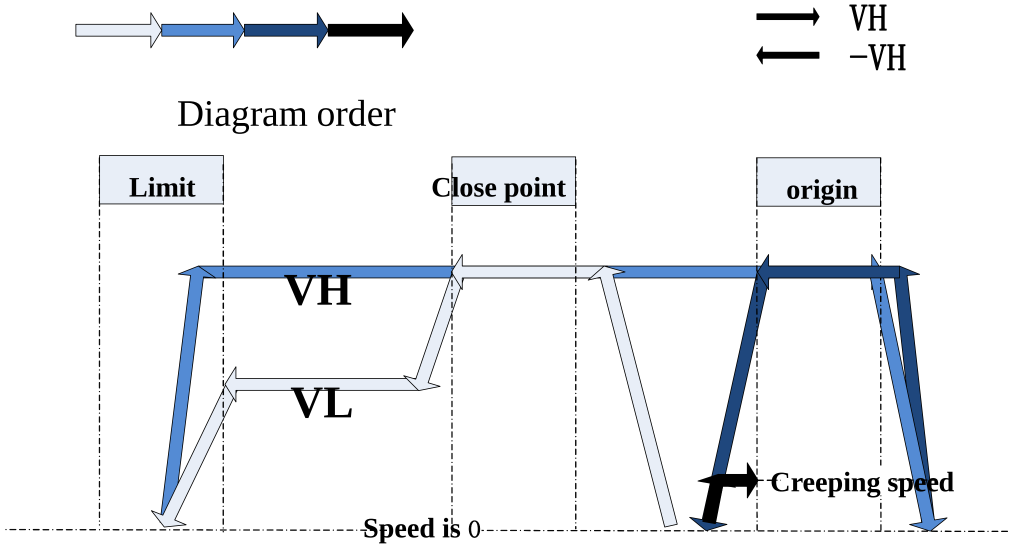
⇨ accelerate to VH, high speed moving in reverse direction

➡ touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

(14) reverse return to origin, initial position is between close point and origin



⇨ -VH high speed touches the falling edge of close point signal, decelerate to –VL, touch the rising edge of min limit, decelerate and accelerate in reverse direction

➡ reverse accelerate to VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ reverse accelerate to –VH, touch the falling edge of origin, decelerate and accelerate in reverse direction

➡ accelerate to reverse creeping speed, touch the rising edge of origin, decelerate to 0 to finish the back to origin action

4. hard limit return to Z phase mode

the return to origin logic is the same to hard limit returning mode, searching Z phase number is the same to Z phase mode.

Note:

(1) For no Z phase mode and Z phase mode: the motor will stop immediately if it touches the min limit when it is returning to origin in reverse direction. At this time, it cannot execute reverse returning to origin, but can execute forward returning to origin. The motor will stop immediately if it touches the max limit when it is returning to origin in forward direction. At this time, it cannot execute forward returning to origin, but can execute reverse returning to origin. Hard limit returning mode is for the occasion that may touch the hard limit.

(2) Executing forward returning to origin SM2014+20*(N-1), the min limit cannot work. Executing reverse returning to origin SM2015+20*(N-1), the max limit cannot work.

2-3-4. Jog run

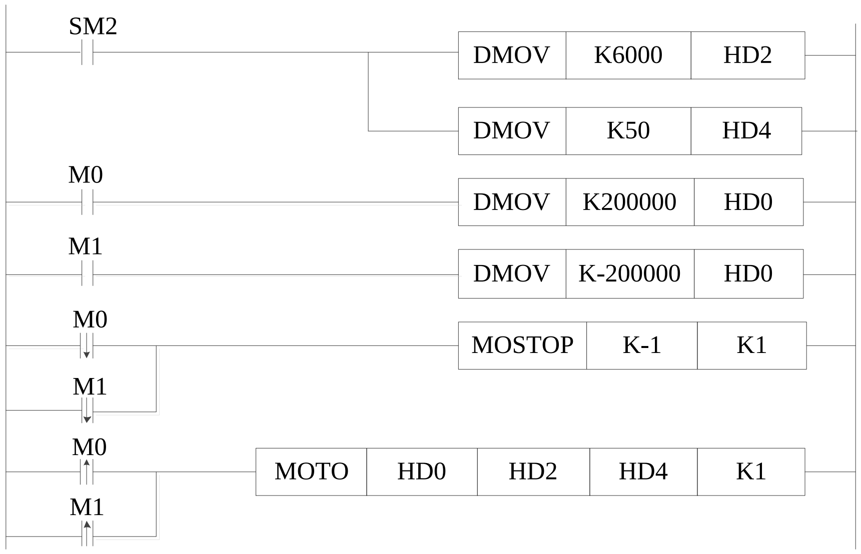
X-NET filedbus has jogging function.

Set the pulses of jog for one time in register (SD2040+60*(N-1)), set the jogging speed in register (SD2042+60*(N-1)). The forward jogging is triggered by the coil (SM2011+20*(N-1)), the reverse jogging is triggered by the coil (SM2012+20*(N-1)).

The jogging signal SM2011+60*(N-1), SM2012+60*(N-1) will be reset immediately after set on. If it needs to keep on jogging, user must make program in HMI or PLC to set on the signal in 100ms, the motion axis will accelerate to jog speed and constant speed run.

Method 1

Continuous jogging: Set the target position to the larger number in the range of target limit value through MOTO instruction.



Note:

SM2: store the value in related register when the PLC starts to run

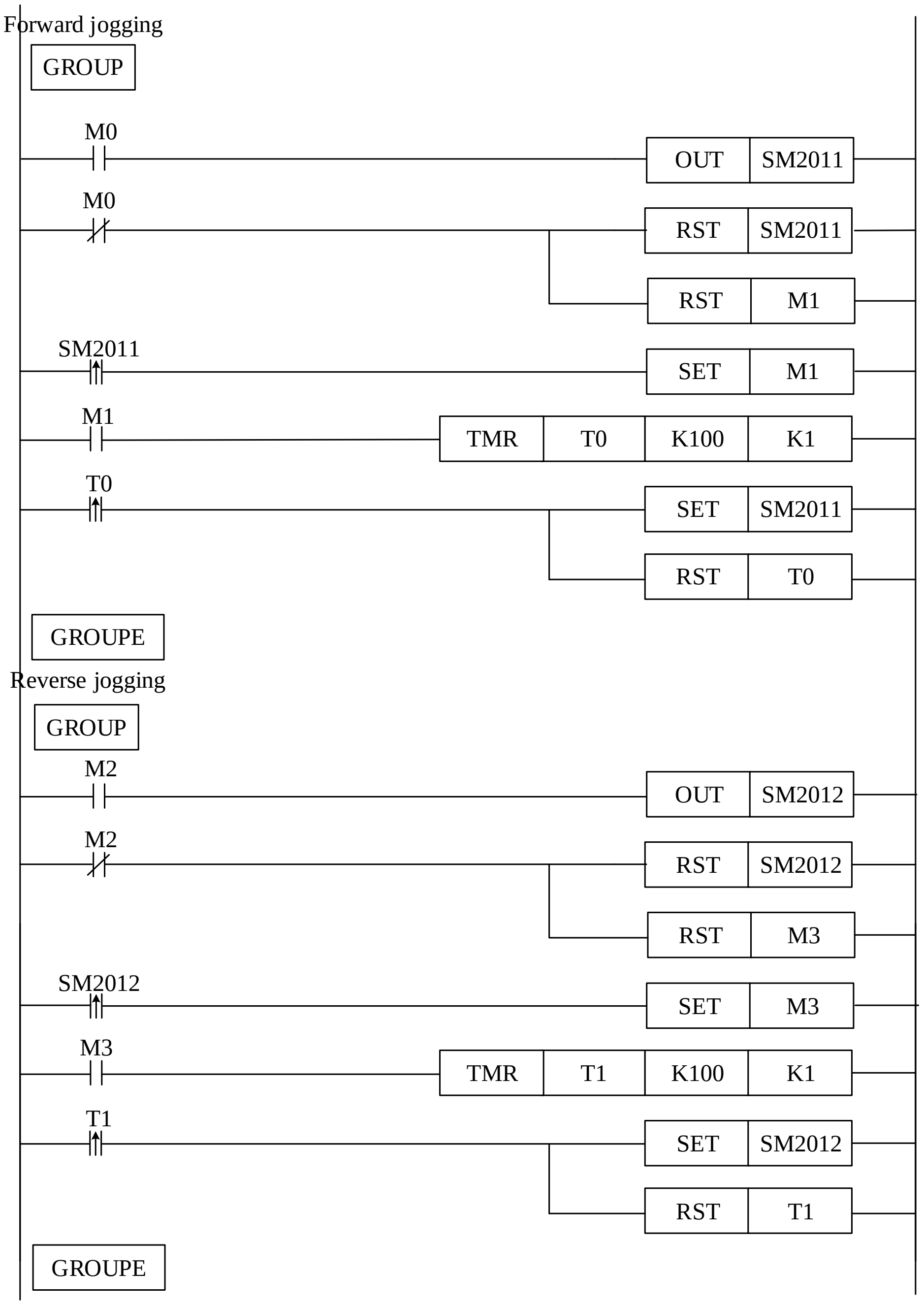
M0: forward jogging button. Press M0 to make the motor run forward. When M0 is reset, the motor stop at once.

M1: reverse jogging button. Press M0 to make the motor run reverse. When M0 is reset, the motor stop at once.

Method 2

Set ON the jogging coil once by timer 100ms. No.1 motor forward or reverse jogging run at 1500Hz.

The program:



Note:

Set the motor step length SD2040 to 1500, jogging frequency SD2042 to 1500Hz in motor enable state.

M0: forward jogging. Press M0 on the HMI screen, the motor will forward jogging run at 1500Hz. Release the M0 button, the motor will stop running.

M2: reverse jogging. Press M2 on the HMI screen, the motor will reverse jogging run at 1500Hz. Release the M2 button, the motor will stop running.

The related register and coil for jog running:

Address	Definition	Type	Unit	Initial value	Note
SFD3024+	Jogging step	32-bit	Pulse	1000	--

60*(N-1)	length initial value	integer	number		
SFD3026+ 60*(N-1)	Jogging speed initial value	32-bit integer	Pulse number/ second	1000	--
SD2040+ 60*(N-1)	Jogging step length	32-bit integer	Pulse number	0	--
SD2042+ 60*(N-1)	Jogging speed	32-bit integer	Pulse number/ second	0	--
SM2011+ 60*(N-1)	Forward jogging	--	--	--	The system will reset automatically after enable
SM2012+ 60*(N-1)	Reverse jogging	--	--	--	The system will reset automatically after enable

Note:

1. The value in SFD register can be changed when the servo enable is OFF.
2. The value in SD register can be changed when the servo enable is ON.

2-3-5. Full closed-loop

In some applications, the equipment makes high precision position control through the grating ruler or encoder, X-NET movement bus full closed-loop motion is achieved by high speed counting and servo feedback to form closed-loop control.

Related coil and register:

Control bit

Address	Definition	Note
SM2016+20*(N-1)	Full closed-loop enable	Set ON: switch the system to the full closed-loop motion state, and the execution of motion instructions shall be based on the position of high speed counting. This operation takes effect for up to 50ms. Set OFF: switch the system from the full closed loop motion state to the normal motion state, that is, the movement instruction execution shall be subject to the position of the servo encoder. This operation takes effect for up to 50ms.

Preset parameters

Address	Definition	Type	Unit	Initial value	Note
SFD3006 +60*(N-1)	Numerator of full closed-loop pulse	16-bit integer		0	The numerator of high speed counter and motion equivalent pulse ratio, the denominator is servo motor pulses per circle, default is 10000 (consistent with servo pulses per circle), at this time, SD2014 is consistent with HSCD [2*i]
SFD3028 +60*(N-1)	High speed counter setting related to full closed-loop	16-bit integer		0	0：HSC0 1：HSC2 ......
SFD3052 +60*(N-1)	Full closed-loop position gain initial value	Floating number		0	When the servo enabled, this register value is assigned to SD2052+60*N
SFD3058 +60*(N-1)	Full closed-loop position deviation limit	Floating number		0	Full closed-loop instruction and feedback deviation limit

Address	Definition	Type	Unit	Note
SD2052+ 60*(N-1)	Full closed-loop position gain	Floating number		This parameter can adjust the speed of acceleration process, that is, the speed of response instruction. It is recommended to take the ratio of servo pulse and full closed-loop high speed counting as the initial value, and then increase slowly to debug. The smaller the gain, the slower the full closed-loop response, the greater the deviation between instruction and feedback, and even the position deviation (the position deviation limit is SFD2058+60*N floating number) alarm. If this gain is too large, vibration and overshoot will appear. Set a suitable gain as needed.

State parameters

Address	Definition	Type	Unit	Note
SD2014+60*(N-1)	Present full closed-loop position	32-bit integer	Pulse number	Coordinate position, converted from full closed-loop position feedback pulse number, when SFD3006+60*N is 10000, this value is consistent with high speed counter value

Address	Definition	Type	Unit	Note
HSD112+20*(N-1)	Full closed-loop position feedback pulse number	64-bit integer	Full closed-loop encoder counting	The unit is full closed-loop (end terminal) encoder (grating scale) counting, relative to the zero position

Example:

1. There is a lead screw device, servo motor drives sliding block on one side of the lead screw, screw position feedbacks through the binding encoder, the slider needs to achieve high precision positioning through the encoder counting, and adjust the full closed-loop to the best state.

Steps:

(1) Choose axis no.1 as the motion axis, set the full closed-loop parameters:

SFD3006: 10000. SD2014/HSD112 is consistent with the preset high speed counter value HSCD[2*i].

SFD3028: 0. The first channel of high speed counter HSC0.

SFD3052: 200. The initial value of the gain can be given a moderate value, too small will report the position deviation, too large will cause large vibration;

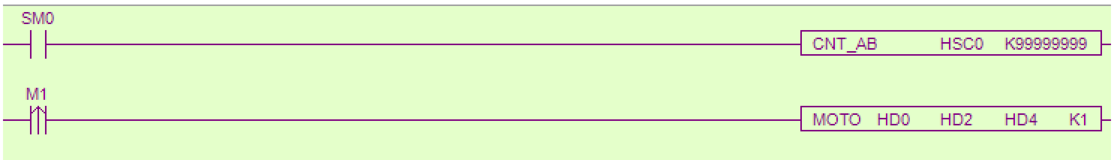
SFD3058: 1000. Preset deviation limit value 1000.

(2) Whether the present servo state meets the working condition:

SD2000=2；SM2000=ON；SM2001=OFF.

If all the conditions are met, set ON the coil SM2016, confirm SFD3052 is written in SD2052, and the value of HSD112/SD2014 is same to HSCD0.

(3) Full closed-loop status supports MOTO and MOTOA instruction, here we take MOTO as an example, and write the related CNT_AB instruction in the program, the target position unit of instruction is based on the position of position sensor (SD2014+60*（N-1）or HSCD[2*i] counter value), servo motor position is just a reference.



```
SM0
CNT_AB  HSC0  K99999999
M1
MOTO  HD0  HD2  HD4  K1
```

(4) Execute the MOTO instruction, if the motor is not running in the ideal state, please adjust the value of SFD3052, then enable SM2016 again, and observe whether the motor is in optimal running state.

Note: please write in the SFD value in inactive state.

Diagnosis and treatment of phenomena:
When using the full closed-loop function, check the error type SD2002+60*(N-1) to solve the problem. If SD2002+60(N-1) reports position deviation, the following conditions are possible:

- SFD3052+60*(N-1) position gain preset value too small
- SFD3058+60*(N-1) deviation limit preset value too small
- MOTO or MOTOA instruction preset pulse frequency too large
- High speed counting positive direction is not consistent with motor running direction. Solution: modify the value SFD3047+60*(N-1)
- The current equipment mechanical principle doesn’t meet the full closed-loop running condition (whether the grating ruler or encoder correctly synchronizes the current axis)

2-3-6. Online simulation

The X-NET motion bus is based on the actual running state and cannot meet the running state without the feedback from the servo. Online simulation can simulate the current running state without actual servo system, the parameter and state feedback is given by the PLC internal calculations, in accordance with the calculation process of connected servo system.

Related coils:

Address	Definition	Note
SM2010+20*(N-1)	Servo enable	ON: servo enable OFF: servo not enable
SM2018+20*(N-1)	Online simulation	After servo enabled, simulation run without connecting servo system, cannot switch when running.

The using method of simulation is same to actual application, it needs to set PLC com port parameters, and SM2010+20*(N-1) enables present axis, and set ON the coil SM2018+20*(N-1) (N is axis number), then PLC will be in online simulation state.

Note:

(1) online simulation one kind of debugging mode, there is no actual data feedback.
(2) Online simulation is invalid when communicating with servo system.
(3) XDC/XDE needs to install XD-NE-BD to use the online simulation.

2-3-7. Protection

PLC will enter online download mode when communication is error or servo enable is off, the servo system will urgent stop. If the PLC stop, the servo will urgent stop.

Note:

1. The power of servo system must cut off after the PLC power cut off, otherwise the servo will revolve rapidly when the PLC startup.
2. The power on order: the servo and PLC must power on at the same time, or the servo power on first.

2-4. System coil and register

When XD/XG series PLC connect to fieldbus servo system, the default can connect 10-axis, the max can connect up to 20-axis. The parameters of 10-axis and 20-axis are different. PLC parameters please refer to the following table, the servo parameters should be changed at the same time. Refer to chapter 2-6-2 for details.

Address	Definition	Connected 10-axis	Connected 20-axis
SFD2990	Instruction refresh period (unit: us)	3000 (default)	6000
SFD2991	Slave station number	10 (default)	20
SFD2992	Error retry time	20 (default)	20

XD/XG series PLC has 1-channel motion fieldbus, can connect 20-axis. The parameters of each axis are shown as the following.

Table 2-4-1: basic parameters (N=1~20)

Address	definition	Type	Unit	Initial value	Notes
SFD3000 +60*(N-1)	Running mode	16-bit integer		0	0: position control with trajectory planning 3: usef-defined motion planning
SFD3001 +60*(N-1)	Motor enocder type	16-bit integer			1: incremental encoder 2: single circle absolute value encoder 3: multi-circle absolute value encoder
SFD3002 +60*(N-1)	Encoder ppr	32-bit integer		10000	The feedback counting value of encoder rotating one circle. It can modify the encoder ppr through this parameter (for example, 17-bit encoder, please set 2^{17} which is 131072)
SFD3004 +60*(N-1)	Moving distance per rotation	32-bit integer	Pulse number	10000	The reference equivalent of motion (screw lead), if the unit is pulse number, the unit is reference moving quantity. The pulse numbers for motor running one circle.
SFD3006 +60*(N-1)	Numerator of full closed-loop ratio	16-bit integer		0	Numerator of ratio between high speed counter and motion equivalent pulse, denominator is servo motor pulse number

					per rotate, default value is SFD3004+60*(N-1)(consistent with servo pulse number per rotate), at this time, SD2014 is consistent with high speed counter value HSCD[2*i]
SFD3010 +60*(N-1)	Origin position	64-bit integer		0	After returning to origin, system will store the value in SD2008+60*(N-1) to calculate the present position
SFD3014 +60*(N-1)	Min position limit	32-bit integer	Pulse number	-1000000000	Min soft limit. Present position SD2008+60*(N-1) less than this value, SD2002+60*(N-1) will produce the alarm 20002, which means min soft limit over the travel
SFD3016 +60*(N-1)	Max position limit	32-bit integer	Pulse number	1000000000	Max soft limit. Present position SD2008+60*(N-1) larger than this value, SD2002+60*(N-1) will produce the alarm 20001, which means max soft limit over travel
SFD3018 +60*(N-1)	Max speed limit	32-bit integer	Pulse number /second	500000	Set this value according to the motor max speed and rated speed. It will run at max limit speed if the speed is over the max limit speed.
SFD3020 +60*(N-1)	The fastest acceleration time	32-bit integer	ms	10	PLC can calculate the acceleration slope and the time accelerating from 0 to max speed according to the acceleration/deceleration time. If the time accelerating from 0 to max speed is less than fastest acceleration time, it will run as the fastest acceleration time. It cannot set to 0 or less than 0.
SFD3022 +60*(N-1)	The fastest deceleration time	32-bit integer	ms	10	PLC can calculate the deceleration slope and the time decelerating from max speed to 0 according to the acceleration/deceleration time. If the time decelerating from

					max speed to 0 is less than fastest deceleration time, it will run as the fastest deceleration time.
SFD3024 +60*(N-1)	Jogging step length initial value	32-bit integer	Pulse number	100	After enabled, PLC will move as the setting jogging step length.
SFD3026 +60*(N-1)	Jogging speed initial value	32-bit integer	Pulse number /second	1000	After enabled, PLC will moveas the setting jogging speed
SFD3028 +60*(N-1)	Full closed-loop related high speed count setting	16-bit integer		0	
SFD3029 +60*(N-1）	Position feedback deviation upper limit	16-bit integer		2500	Positive integer: deviation upper limit -1: ignore the deviation value
SFD3034 +60*(N-1)	Min position limit terminal	16-bit integer		0xFF	Set the X terminal no., 0xFF is no terminal, negative number means negative logic. Note: X0 positive logic is 0, negative logic is -30000.
SFD3035 +60*(N-1)	Max position limit terminal	16-bit integer		0xFF	Set the X terminal no., 0xFF is no terminal, negative number means negative logic. Note: X0 positive logic is 0, negative logic is -30000.
SFD3036 +60*(N-1)	Close signal terminal	16-bit integer		0xFF	Set the X terminal no., 0xFF is no terminal, negative number means negative logic. Note: X0 positive logic is 0, negative logic is -30000.
SFD3037 +60*(N-1)	Origin terminal	16-bit integer		0xFF	Set the X terminal no., 0xFF is no terminal, negative number means negative logic. Note: X0 positive logic is 0, negative logic is -30000.
SFD3038 +60*(N-1)	Return to the origin mode	16-bit integer		0	0: no Z phase mode. search the close point with VH speed, then search the origin point with VL speed, decelerate after finding the origin point, then

					search the origin point in reverse direction with creeping speed 2: Z phase mode. search the close point with VH speed, then search the origin point with VL speed, decelerate after finding the origin point, then search the origin point in reverse direction with creeping speed, search the servo encoder Z phase in forward direction 10: hard limit return mode. Search the origin in reverse direction with –VH speed after touching the ±hard limit, the speed becomes VH after touching the origin, the following action is same to mode 0 12: hard limit return to Z phase mode. Search the origin in reverse direction with –VH speed after touching the ±hard limit, the speed becomes VH after touching the origin, the following action is same to mode 2.
SFD3040 +60*(N-1)	Returning speed VH	32-bit integer	Pulse number / second	0	High speed of returning to origin speed
SFD3042 +60*(N-1)	Returning speed VL	32-bit integer	Pulse number /second	0	Low speed of returning to origin speed
SFD3044 +60*(N-1)	Creeping speed	32-bit integer	Pulse number /second	0	Creeping speed of origin return
SFD3047 +60*(N-1)	Motion direction logic	16-bit integer		0	0: positive logic, command speed is positive, the motor run forward, command speed is negative, the motor run reverse. 1: negative logic, command speed is positive, the

					motor run reverse, the command speed is negative, the motor run forward.
SFD3048 +60*(N-1)	Positioning finished width initial value	32-bit integer	Pulse number	20	The PLC will have positioning finished signal when target instruction and actual encoder feedback is in this width range, it no needs to get the signal after the positioning finished.
SFD3052 +60*(N-1)	Full closed-loop position gain initial value	32-bit floating number			When servo enabled, this register value will be sent to SD2052+60*(N-1)
SFD3058 +60*(N-1)	Full closed-loop position deviation limit	32-bit floating number			Full closed-loop instruction and feedback deviation limit

Table 2-4-2: state parameters (N=1~20)

Address	Definition	Model	Unit	Note
SD2006+ 60*(N-1)	present offset	32-bit integer	Pulse number	The offset relative to the last time stop position, which is the offset in this instruction
SD2008+ 60*(N-1)	Present position	32-bit integer	Pulse number	absolute position, calculate from the target position feedback pulse number
SD2010+ 60*(N-1)	Present speed	32-bit integer	Pulse number/ second	Calculate from motor actual feedback speed
SD2012+ 60*(N-1)	Instantaneous speed measurement		Pulse number /second	The setting speed of single control period
SD2014+ 60*(N-1)	Present full closed-loop position	32-bit integer		Full closed-loop absolute position, convert from full closed-loop position feedback pulse, when SFD3006+60*(N-1) and SFD3004+60*(N-1) is same, this value is consistent with high speed count value
SD2016+ 60*(N-1)	Present segment (represent segment no.	32-bit integer		

	n)			
SD2020+ 60*(N-1)	Present torque	32-bit floating number	N.m	When servo P7-02 set to 3, SD2029+60*(N-1) set to 1, SD2020+60*(N-1) will show present torque. Unit is N.m
SD2024+ 60*(N-1)	Torque setting	32-bit integer	1/10000 rated	Effective mode: Servo P0-01=8(torque mode): always effective Servo P0-01=9 or 10(speed moe or position mode): be effective when SD2028+60*(N-1)=1
SD2026+ 60*(N-1)	Reverse torque setting	32-bit integer	1/10000 rated	In speed mode or position mode, be effective when SD2028=1. At this time, servo value P3-28, P3-29 is ineffective. Ineffective in torque mode.
SD2028+ 60*(N-1)	Torque control mode	16-bit integer		Servo P0-01=9 or 10(speed mode or position mode): when SD2028+60*(N-1)=1, SD2024+60*(N-1) and SD2026+60*(N-1) are effective, at this time, servo value P3-28, P3-29 is ineffective.
SD2029+ 60*(N-1)	Torque feedback enable	16-bit integer		0: ineffective 1: SD2020+60*(N-1) shows present torque value(servo P7-02 must be 3)

Table 2-4-3: setting parameters (N=1~20)

Address	Definition	Model	Unit	Note
SD2030+ 60*(N-1)	Position setting	32-bit integer	Pulse number	Coordinate position, conversion of target position setting pulse number. Change the position setting value, the motor will move to the setting position with the setting speed (SD2032+60*(N-1)).
SD2032+ 60*(N-1)	Speed setting	32-bit integer	Pulse number /second	
SD2034+ 60*(N-1)	Acceleration time setting	32-bit integer	ms	The time accelerate from 0 to the max speed
SD2036+ 60*(N-1)	deceleration time setting	32-bit integer	ms	The time decelerate from max speed to 0
SD2038+ 60*(N-1)	Synchronized motion speed	32-bit floating		Tracking axis speed/ be tracked axis speed

	ratio	number		
SD2040+ 60*(N-1)	Jogging step length	32-bit integer	Pulse number	When servo enabled, system will send SFD3024+60*(N-1) to this register as initial value. After servo enabled, this register can be modified online.
SD2042+ 60*(N-1)	Jogging speed	32-bit integer	Pulse number /second	When servo enabled, system will send SFD3026+60*(N-1) to this register as initial value. After servo enabled, this register can be modified online.
SD2044+ 60*(N-1)	Positioning finished width	32-bit integer	Pulse number	The threshold to judge the positioning finished, the moving flag is OFF when the difference between setting value and encoder feedback is less than this parameter. When the servo is enabled, the system will send the SFD3048+60*(N-1) to this parameter, this parameter value can be changed online.
SD2052+ 60*(N-1)	Full closed-loop position gain	32-bit floating number		This parameter can adjust the speed of aceleration process, which is response instruction speed. We suggest to use the ratio of servo pulse and full closed-loop high speed counter as initial value, then debug by increasing the value slowly. The smaller the gain, the slower the full closed-loop response, the larger of the deviation between instruction and feedback, and will cause position deviation alarm(position deviation limit is SFD2058+60*N). The larger the gain, it will cause vibration and overshoot. Please set a suitable gain as needs. When full closed-loop enabled, system will send the SFD3052+60*(N-1) to this register as initial value. After servo enabled, this register can be modified online.
SD2059+ 60*(N-1)	Synchronized motion filter factor	32-bit integer		The range is 0~9999.

Table 2-4-4: self-maintaining state parameters (N=1~20)

Address	Definition	Model	Unit	Note
HSD100+ 20*(N-1)	Target position setting pulse numbers	64-bit integer	Encoder count	Encoder counting value relative to the absolute zero position
HSD104+ 20*(N-1)	Target position feedback pulse numbers	64-bit integer	Encoder count	Encoder counting value relative to the absolute zero position

HSD108+20*(N-1)	Present offset pulse numbers	64-bit integer	Encoder count	The count value relative to the present motion instruction starting encoder position
HSD112+20*(N-1)	Full closed-loop position feedback pulse number	64-bit integer	Full closed-loop enocder counting	The unit is Full closed-loop(end terminal) encoder(grating ruler) counting, relative to the zero position

Table 2-4-5: state bit parameters (N=1~20)

Address	Definition	Note
SM2000+20*(N-1)	Servo enable flag	ON: servo enable state
SM2001+20*(N-1)	Moving flag	ON: pulse is outputting. Set it OFF when the motion stop
SM2003+20*(N-1)	Instruction completed flag	ON: instruction execution completed. Set it OFF when the instruction start executing.
SM2004+20*(N-1)	Axis error flag	ON: error
SM2005+20*(N-1)	Min position limit status	ON: the present position is less than min position limit or min position limit signal is effective
SM2006+20*(N-1)	Max position limit status	ON: the present position is larger than max position limit or max position limit is effective
SM2009+20*(N-1)	Slave aixs binding flag	ON: present axis is bound

Table 2-4-6: control bit parameters (N=1~20)

Address	Definition	Note
SM2010+20*(N-1)	Servo enable	ON: servo enable; OFF: servo disable
SM2011+20*(N-1)	Forward jogging	System will automatical reset after enable
SM2012+20*(N-1)	Reverse jogging	System will automatical reset after enable
SM2013+20*(N-1)	Clear the servo alarm	System will automatical reset after enable
SM2014+20*(N-1)	Forward return to origin	System will automatical reset after enable
SM2015+20*(N-1)	Reverse return to origin	System will automatical reset after enable
SM2016+20*(N-1)	Full closed-loop enable	Set ON: switch to full closed-loop state, motion instruction is based on high speed counting position. Be effective in 50ms after executing this operation. Set OFF: switch from the full closed-loop state to normal motion state, motion instruction is based on servo encoder position. Be effective in 50ms after executing this operation.
SM2017+20*(N-1)	Decelerate and stop	System will automatical reset after enable

SM2018+20*(N-1)	Online simulation	After enable, when the servo is not connected to the system, simulation run, cannot switch when running.

2-5. Error and state message

The following table shows the error details and solutions.

Control bit parameter (N=1~20)

Address	Definition	Value	Notes	Operation	Solution
SD2000+60*(N-1) (double words)	Servo status	0	disconnect		Check the wiring between servo and PLC
		1	READY		
		2	ON		
SD2002+60*(N-1) (double words)	Error message	20001	Max soft position limit overrange	Urgent stop	Reverse jogging to go out the overrange area, clear the alarm flag and code by manual
		20002	Min soft position limit overrange	Urgent stop	Reverse jogging to go out the overrange area, clear the alarm flag and code by manual
		20003	Max electrical position limit overrange	Urgent stop	Reverse jogging to go out the overrange area, clear the alarm flag and code by manual
		20004	Min electrical position limit overrange	Urgent stop	Reverse jogging to go out the overrange area, clear the alarm flag and code by manual
		20005	Over speed alarm	Decelerate stop	Decrease the instruction target speed
		20006	Position offset alarm	Urgent stop, shut off the enable	the servo parameter P0-05 must be 0. Check if the machine stalled and offset of position instruction and feedback are too large. Please solve the problem and enable the servo again.
		20010	Servo alarm	Urgent stop, shut off the enable	The servo alarm which can be cleared by manual can be cleared through F0-00 or SM2013+20*(N-1). the servo alarm which cannot be cleared by manual should be cleared according to the servo manual. Please clear the alarm flag bit and code by manual after alarm

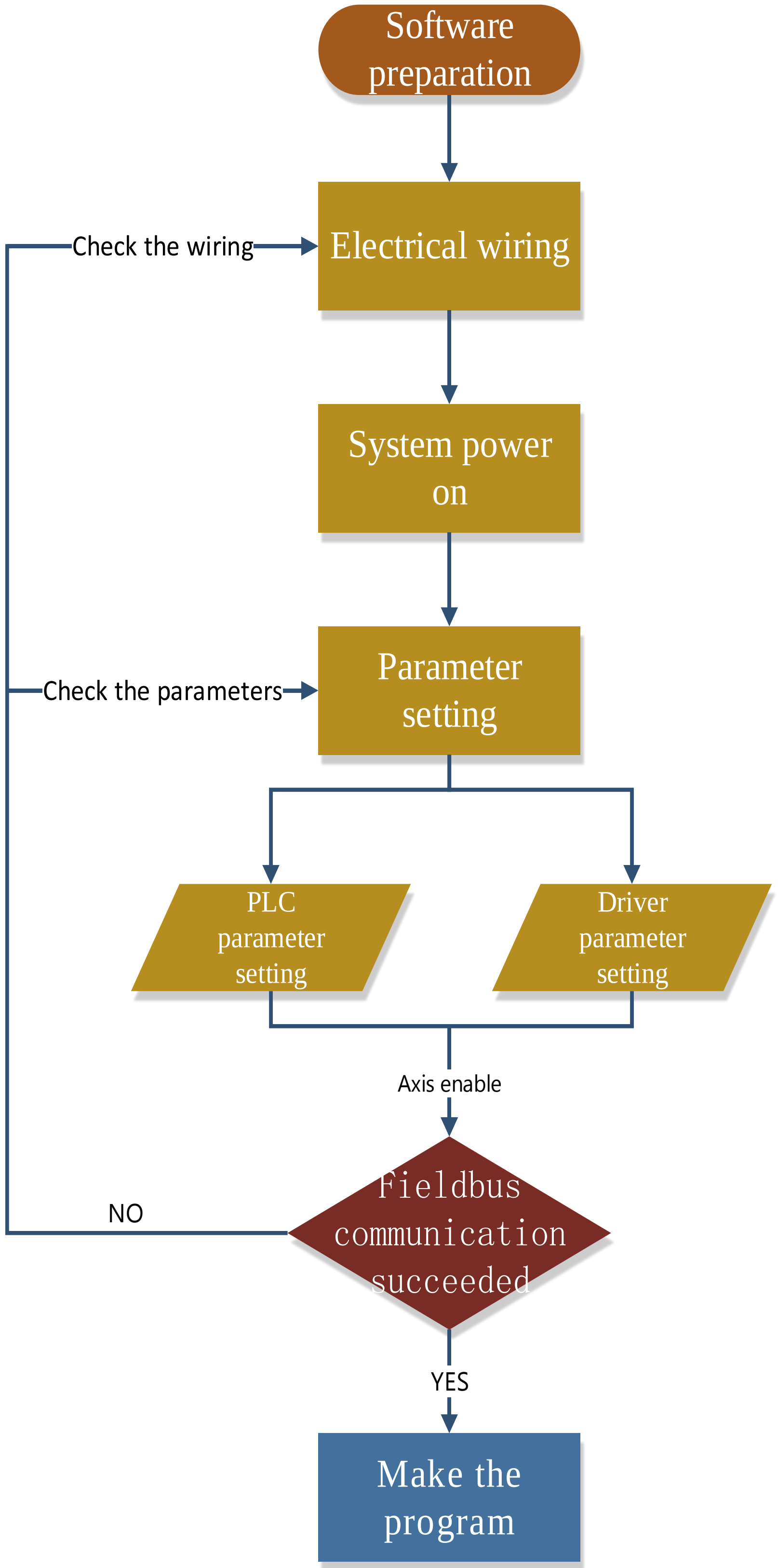
					is disarmed.
		20011	Servo communication error	Urgent stop and shut off the enable, switch the PLC status to online downloading	Check the communication parameters and cables
		20020	Motion instruction target point overrange alarm	Motion instruction execution invalid	Change to suitable target position
		20021	Motion instruction target speed overrange	Instruction execution invalid	Change to suitable target speed
		20022	The segment number of multi-segment motion instruction overrange	Motion instruction execution invalid	Change to suitable segment number of multi-segment insturction
		20023	Motion instruction acceleration/deceleration time overrange	Motion instruction execution invalid	Change to suitable instruction motion acceleration/deceleration time
		20024	Reserved		
		20025	Bound axis no. overrange	Motion instruction execution invalid	Check the bound axis no. of the instruction
		20026	The input point of returning to zero terminal overrange	Cannnot go back to zero	Check the input terminal settings for returning to zero, including close terminal and origin terminal
		20030	Present motion status cannot meet the instruction execution condition	Motion instruction execution invalid	Execute the instruction again when the bound axis motion flag SM2001+20*(N-1) is OFF and servo enable flag SM2000+20*(N-1) is ON
		20031	The motion status of bound	Bound instruction	Execute the MOSYN instruction when the bound axis motion flag

			axis cannot meet the bound instruction execution condition	execution invalid	SM2001+20*(N-1) is OFF and servo enable flag SM2000+20*(N-1) is ON
		20032	Present axis motion mode setting error (SFD3000+60*N)	SM2010+20*(N-1) set ON is invalid, cannot enable the servo	Check whether the single word of SFD3000+60*(N-1) is 0, 1, 3, enable the servo again after correcting
		20033	Reserved		
		20034	The motor has been in enable status when user enabled the servo	The enable operation is invalid	Check the servo enable mode which caused the servo enabled, then enable the servo again
		20035	Motor type is not set	The motor cannot be enabled	Check the register SFD3001+60*(N-1), correct it and run the PLC again. SFD3001+60*(N-1) motor type code: 1: incremental encoder motor 2: single rotation absolute encoder motor 3: absolute encoder motor 4: stepper motor 5: xinje encoder
		20036	Present return to origin failure alarm	Present motion stop	1: check whether touch the soft limit 2: whether the returning to origin process is complete
SD2004+60*(N-1) (double words)	Fieldbus communication error times				(1) Check the servo drive parameters P7 (2) Check the communication cable between PLC and servo

2-6. Appendix

Fieldbus configuration process:

Motion fieldbus configuration process



2-6-1. XINJEConfig software

- **XINJEConfig installation**

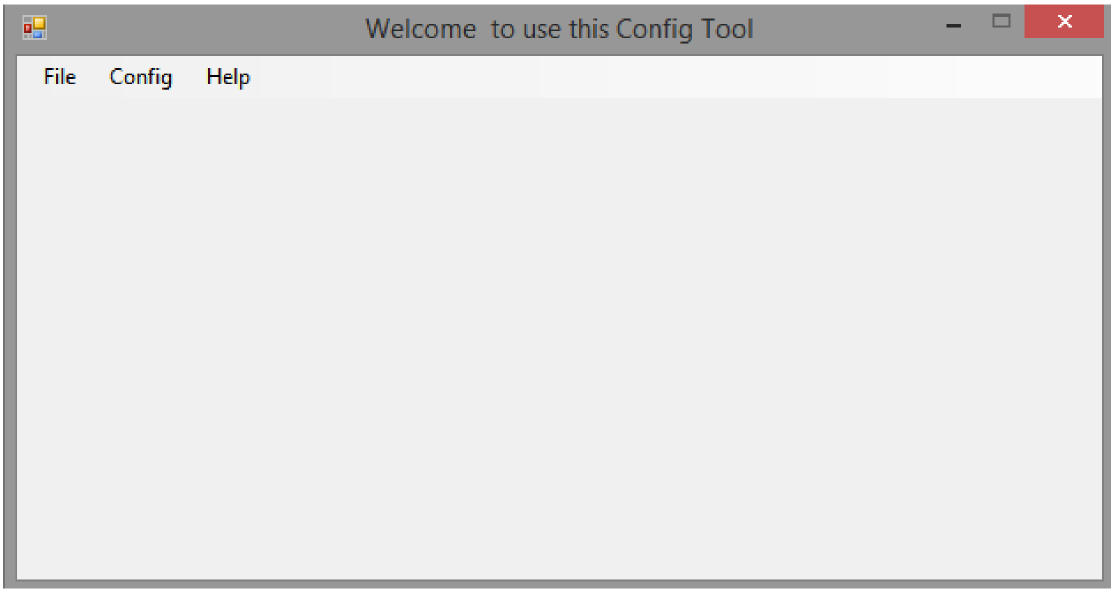
1. Please contact us to get this software. Generally, user can find it in the XD software installation package.
2. install the software according to the guide.

- **Use XINJEConfig**

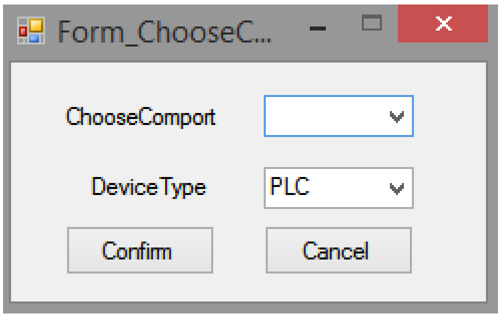
Please connect PLC with PC through USB cable before using XINJE config.

USB cable can work after the driver is installed which will be installed together with the XINJE config.

1. Open the XINJE config, it will show the following window:



2. Click “config”-Finddevice, it will show Form_ChooseCompot window.

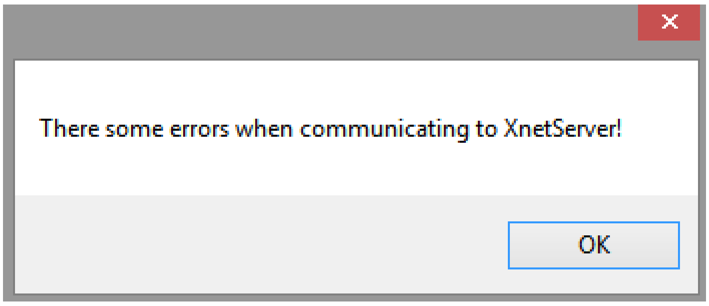


ChooseComport:

DeviceType: PLC

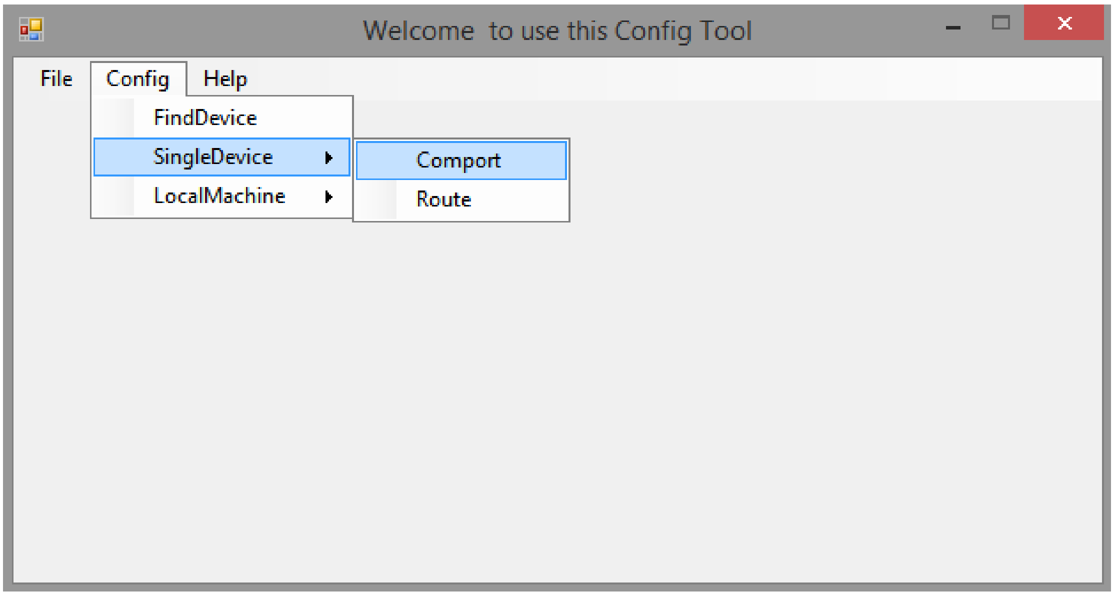
3. Select the com port on your PC, the device type is PLC, then click confirm. If it shows the following error, please restart the software and do it again until it connected the device

successfully.

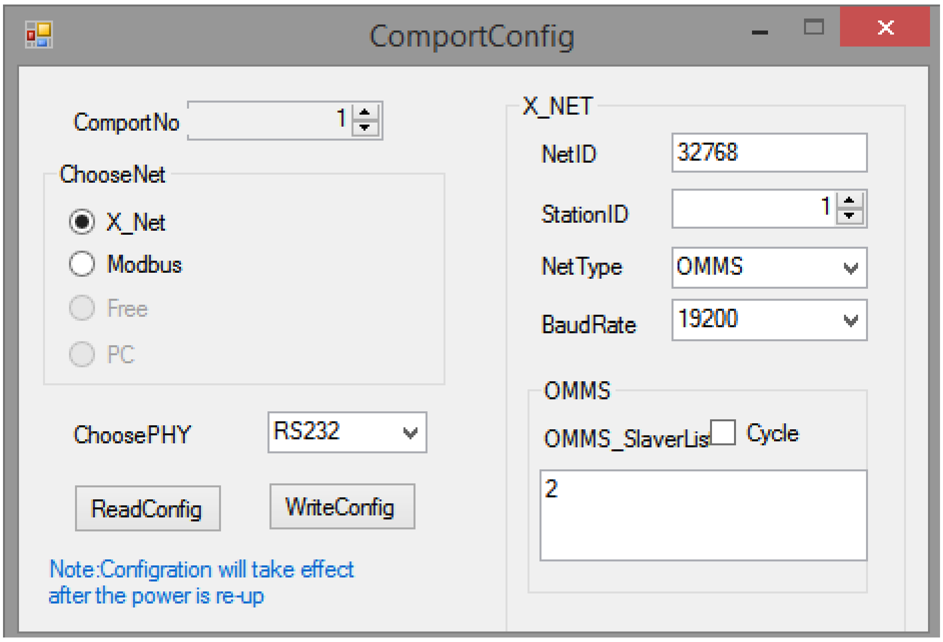


There some errors when communicating to XnetServer!

4. Click ok to return to the main window, click config-Singledevice-Comport.



5. It will show the serial port configuration window.



ComportNo 1

ChooseNet

● X_Net

○ Modbus

○ Free

○ PC

ChoosePHY RS232

Note:Configration will take effect after the power is re-up

X_NET

NetID 32768

StationID 1

NetType OMMS

BaudRate 19200

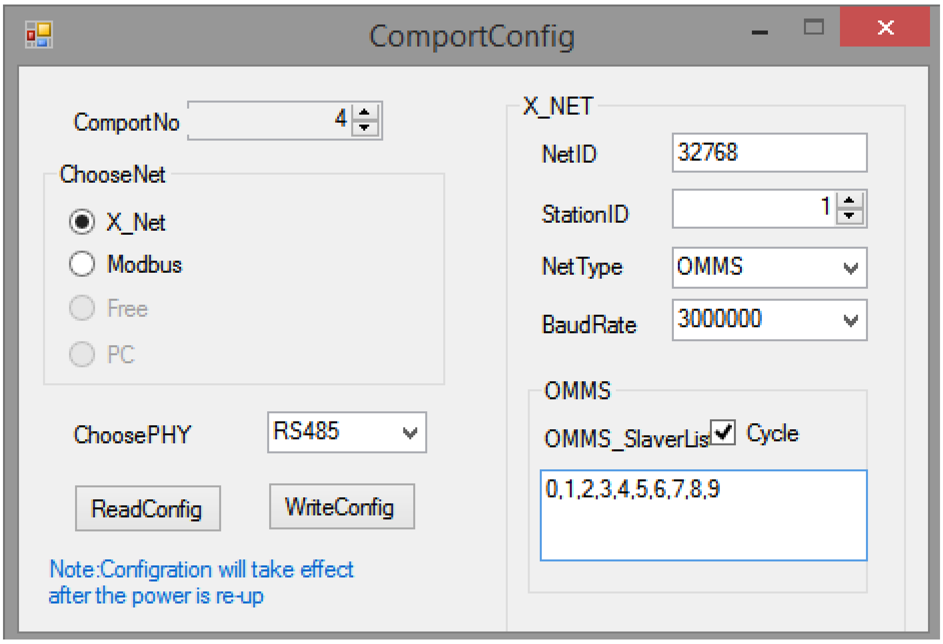
OMMS

OMMS_SlaverLis □ Cycle

2

6. As the XDC/XDE series PLC RS485 BD board is serial port 4, here please set the

comport to 4. The net type is X-NET. The PHY please set to RS485. If the model is XG1 series PLC, please choose serial port 2.



ComportNo 4

● X_Net
○ Modbus
○ Free
○ PC

ChoosePHY RS485

Note:Configration will take effect after the power is re-up

NetID 32768

StationID 1

NetType OMMS

BaudRate 3000000

OMMS_SlaverLis ☑ Cycle

0,1,2,3,4,5,6,7,8,9

NetID: the network number of all the devices must be consistant in the same network. The network number can be set to any value.

Station ID: the PLC station no.

Net type: Please select OMMS when the PLC communicates with servo drive.

Baud rate: depend on the servo baud rate.

Cycle: PLC communicate with servo in cycle.

OMMS slaver list: keep default or set the slave station no.

7. Click “write config”. It will show the write in successful window.
8. Click confirm, restart the PLC again to make the configuration effective.

Note:

(1) When the device is changed, please configure again.
(2) Please run the XINJE config as administrator for the PC system higher than win7.
(3) Please close the antivirus software when using the XINJEconfig.

2-6-2. Servo drive parameters

It needs to set the servo drive parameters before debugging the program. The fieldbus parameters are shown as the following.

1. Parameters

10-axis and down(include 10-axis): change the default value to suitable value for parameter P0-01, P0-03, P7-00, P7-01, P7-02.

10-axis to 20-axis: change the default value to suitable value for parameter P0-01, P0-03, P7-00, P7-01, P7-02, P7-05, P7-07.

Servo parameter	Functions	Default value	10-axis value	20-axis value
P0-01	Control mode 1 8: fieldbus torque mode 9: fieldbus speed mode 10: fieldbus position mode	6	10	10
P0-03	Enable mode 1: IO enable 2: software enable 3: fieldbus enable	1	3	3
P7-00	RS485 station no.	1	1~10	1~20
P7-01	RS485 serial port parameters	0x2206	0x2213*	0x2213*
P7-02	RS485 communication protocol 1: Modbus 2: simple XNet	1	2	2
P7-03	Sampling time (unit: character)	9	--	--
P7-04	Slave station space time (unit: character)	15	--	--
P7-05	Slave station quantity 1~10	10	10	20
P7-06	Retry times	10	--	--
P7-07	Instruction refresh period (unit: us)	3000	3000	6000
……				
P7-20	The Z phase numbers in returning to origin mode. If it is 0, search the first Z phase; if it is negative value, search the reverse direction Z phase	1	--	--

Note:

(1) *: 1 stop bit, even parity, baud rate is 3Mbps.

(2) Sampling time and slave station space time are set to character. The actual needed time is transferring setting characters(start bit+data bit+parity bit+stop bit) with setting baud rate.

(3) The time distributed to slave station: slave station transferring time + slave station space time, the slave station data will send in the middle period of the time.

2. Monitor

It can monitor the communication status between servo motor and PLC through the following servo parameters.

Parameter	Contents
U0-61	Communication error times
U0-62	Synchronizing frame receiving error times (overtime or data error)

U0-64	Data frame receiving error times (overtime or data error)
U0-66	CRC error times
U0-67	UART error times (UART module error: RS485 noise is too large, CPU has not read the shifting register in time and caused the data damaged.)
U0-68	Communication overtime times. If the servo communication error period is larger than or equal to P7-06, U0-68 +1, servo Xnet switches to “initial status”, UART priority decreases, waits the synchronizing frame, servo will not alarm now.

3 Ethernet communication

3-1. Function summary

3-1-1. Introduction

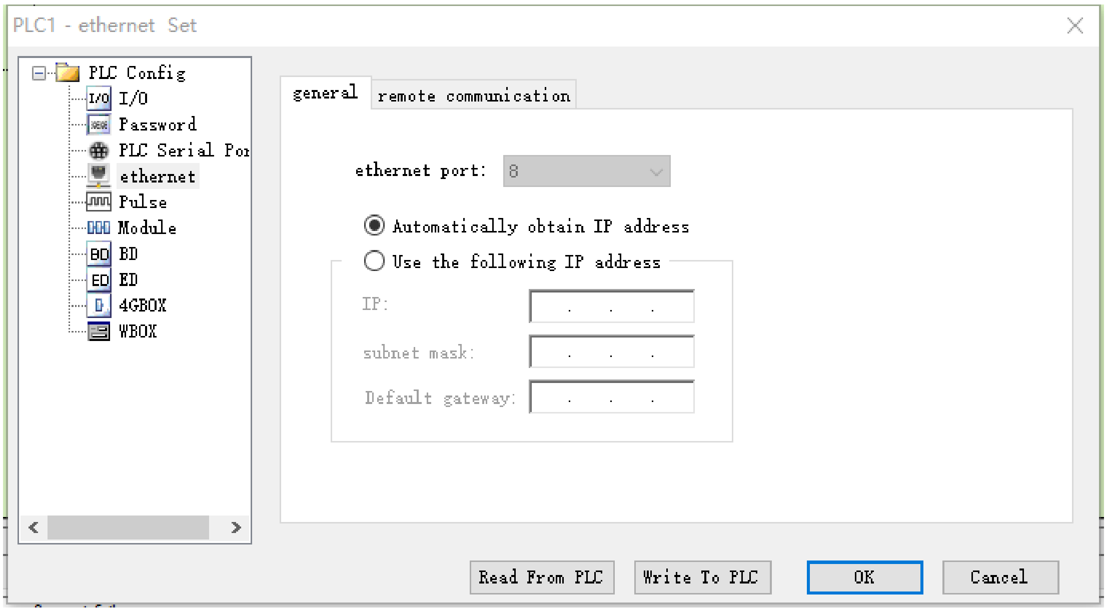
Ethernet communication is mainly used in XDE/XD5E/XG1 series PLC with RJ45 physical interface, which is more rapid and stable than USB communication mode. When data monitoring of PLC is carried out in XD/XG/XL series PLC programming software, the user can obtain better feeling of real-time data and faster program download through Ethernet communication. And the connection mode of Ethernet communication has obvious advantages than RS485 and USB, reliable and easy connection mode is especially suitable for many sets of PLC communication, users need only a switch to control any PLC on the spot, at the same time, high speed, stable Ethernet network provides strong support for large capacity data interaction.

In addition to LAN applications, Ethernet also supports remote searching, data monitoring, and program download of PLC over the Internet.

3-1-2. Ethernet parameters

■ **IP address parameters**

XDE/XD5E/XG1 series Ethernet was developed based on the standard TCP/IP protocol. The IP address should be set as each device number. There are four parameters for IP address setting, as shown in the figure below. The image is the IP address setting interface of XINJE XDPPro.



ethernet port: 8

● Automatically obtain IP address

○ Use the following IP address

IP: . . .

subnet mask: . . .

Default gateway: . . .

(1) IP address acquisition method

XDE/XD5E/XG1 series Ethernet port PLC supports IP address automatic acquisition and static IP address setting. The default setting of PLC is automatic acquisition.

When the IP address acquisition mode of PLC is set as static IP address, the IP address becomes static IP after PLC power off and restart. After the IP address of the PLC is set to automatically obtain, when there is equipment providing DHCP service in the PLC LAN, PLC IP address, subnet mask and default gateway will be automatically assigned by the DHCP Server; When there is no equipment providing DHCP service in PLC LAN, the IP address and other parameters of PLC will be the following initial values:

IP address: 192.168.6.6
Subnet mask: 255.255.255.0
Default gateway: 192.168.6.1

General applications are recommended as follows:

Communication protocol	Typical applications	Configuration mode	others
Modbus TCP	HMI, OPC, SCADA (WINCC, Kingview)	Static setting	--
X-NET	XDPPro programming online, XINJE cloud monitoring	Automatical obtain	ID automatic searching communication
X-NET	PLC Ethernet communication function	Static setting	Appointed IP communication

(2) IP address

The defaulted IP address of XDE/XD5E/XG1 series Ethernet port PLC is class C, and the user setting of class C address is set as 192.168.A.B (the value range of A is 0~255, and the value range of B is 1~254).

(3) Subnet mask

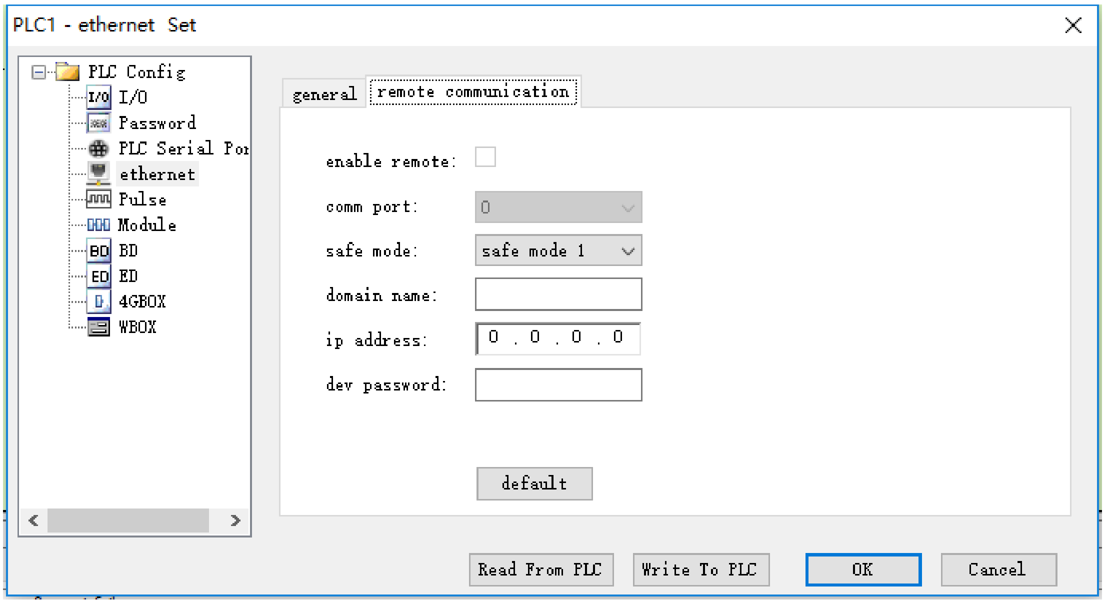
The default subnet mask of Ethernet port PLC is 255.255.0. When the devices in the PLC LAN is greater than 254, it can be reduced appropriately.

(4) Default gateway

In LAN connections, the default gateway may not be set when communicating only with devices in the local network. If the PLC turns on the remote communication function, the gateway needs to be set to router IP, which can access the Internet.

■ Remote communication parameters

The parameters need to be set when it requires PLC to realize remote monitoring functions, such as remote debugging, uploading and downloading PLC programs, and XINJE cloud equipment monitoring, etc. As shown in the figure below, the picture is remote parameter setting interface of XINJE XDPPro.



enable remote: □

comm port: 0

safe mode: safe mode 1

domain name:

ip address: 0 . 0 . 0 . 0

dev password:

(1) Enable remote

After remote communication is enabled, PLC will log in target server as user’s settings (needs to install XINJE server) and perform remote communication function.

(2) Comm port

Ethernet remote communication port, cannot be changed.

(3) Safe mode

Ethernet remote communication supports safe mode 1.

(4) Domain name

Ethernet remote communication domain name, the default name is www.x-net.info.

(5) IP address

Ethernet remote communication server IP address, it no needs to set when the server domain name is set. If the server domain name is not set, please input correct server IP. XINJE remote server defaulted IP is 61.160.67.86.

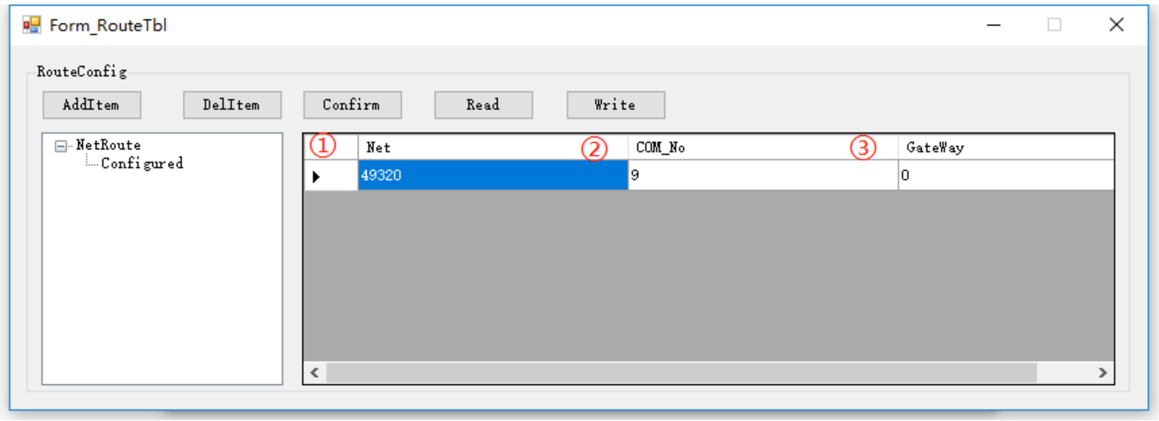
(6) Device password

Ethernet remote communication password, it can log in successfully after inputting correct password.

■ Routing table parameters

XDE/XD5E/XG1 series Ehternet port PLC supports x-net communication protocol, x-net protocol supports multi-network communication, and the target equipment of communication is uniquely specified by network number and site number. When the device adds a communication network, it needs to add a corresponding routing table, and the relevant configuration of the routing table is completed through XINJE Config.

The network number corresponding to PLC in LAN communication through Ethernet is determined by the first two segments of IP address. When PLC uses class C IP address 192.168.x.x, the corresponding network number is 49320. The Ethernet port PLC will have a defaulted routing information when it leaves the factory, as shown in the figure below.



Net	COM_No	GateWay
49320	9	0

(1) Target network number

This is the network number of communication target. 49320 is corresponding to TCP/IP network. The first two segments of PLC IP address 192.168 converting to hex format is HCOA8, and then converting to decimal format is 49320.

(2) Communication port

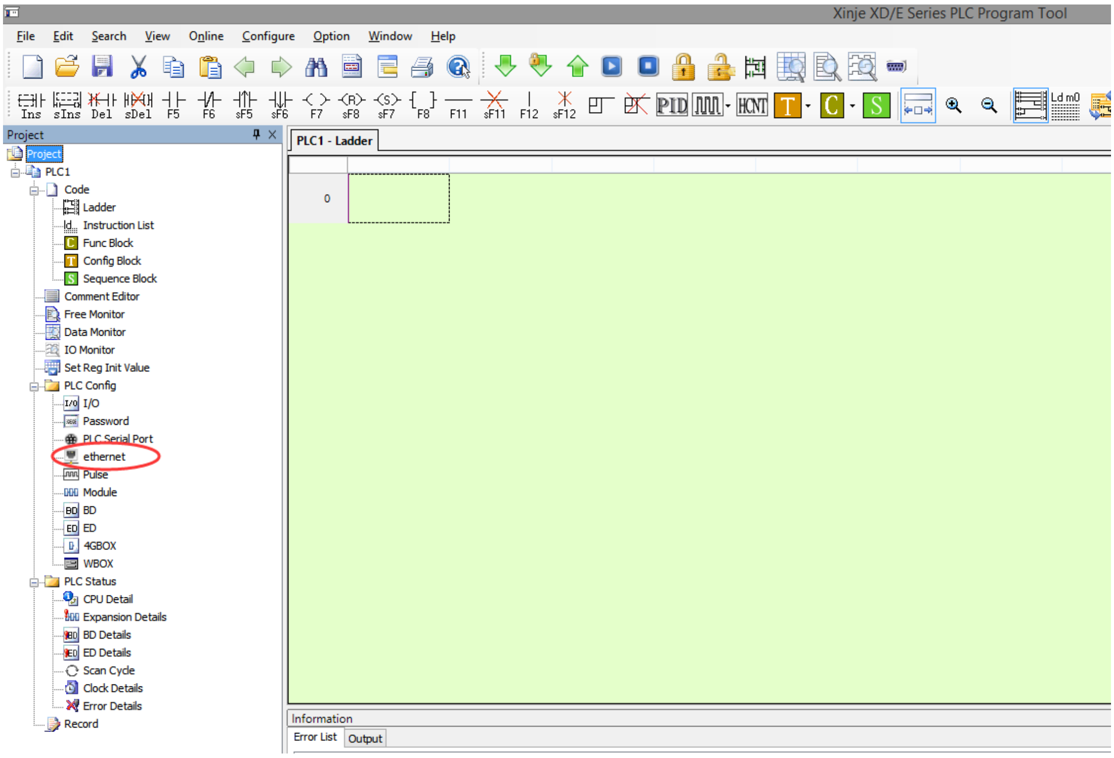
COM_No is the equipment communication port, Ethernet communication uses PLC RJ45 port, the port no. is 9.

(3) Gateway

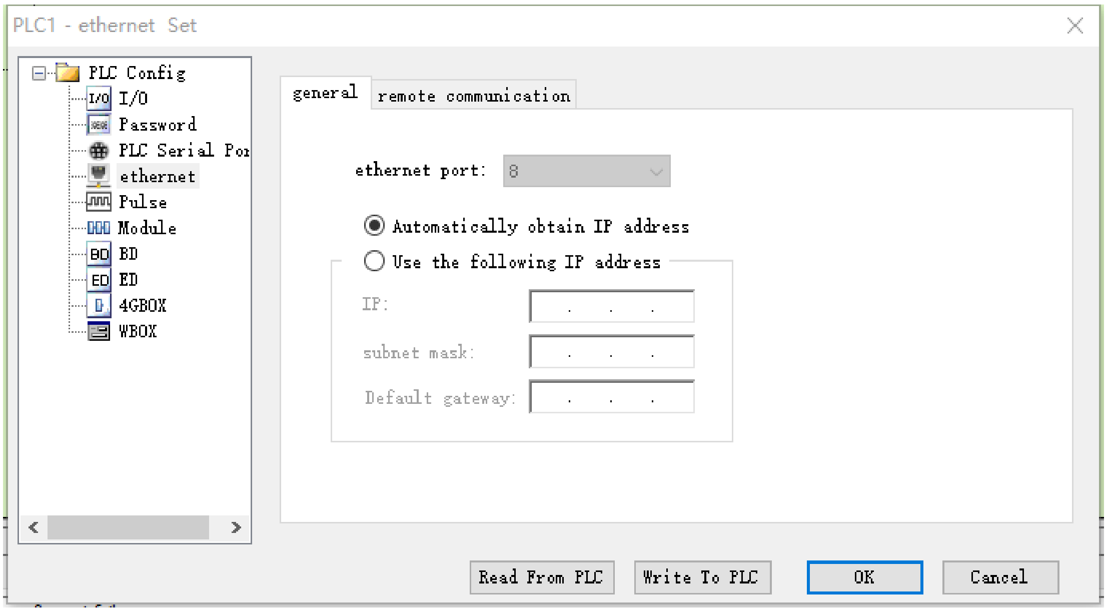
When the device and the target communication device are in the same network, no device forwarding information is required. The communication GateWay is set as the default value 0. When the device and the target communication device are not in the same network, the device needs to forward the information, and the communication gateway is set as the station number of the forwarding device. Here, PLC and other communication devices are in the TCP/IP network with the network number 49320, so the GateWay is set to 0.

3-1-3. Ethernet configuration of XINJE XDPPro

Open the XDPPro software, click Ethernet in the software, refer to below figure. Make sure the PLC model is XD5E or XG series, otherwise it will show the error message current model not support.



When the PLC model is set to XDE/XD5E/XG, please open Ethernet window, refer to below figure. Each part explanation please refer to chapter 3-1-2.



ethernet port: 8

● Automatically obtain IP address

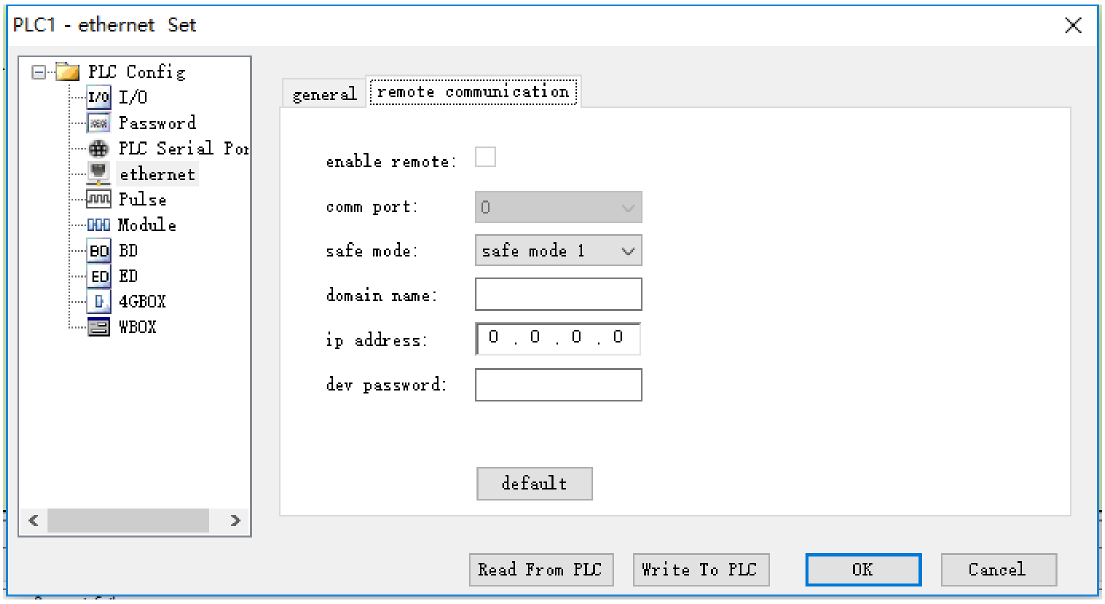
○ Use the following IP address

IP: . . .

subnet mask: . . .

Default gateway: . . .

Click remote communication to enter remote parameters configuration window. After setting all the parameters, please cut off the power and power on again to make the setting effective.



PLC1 - ethernet Set

enable remote: □

comm port: 0

safe mode: safe mode 1

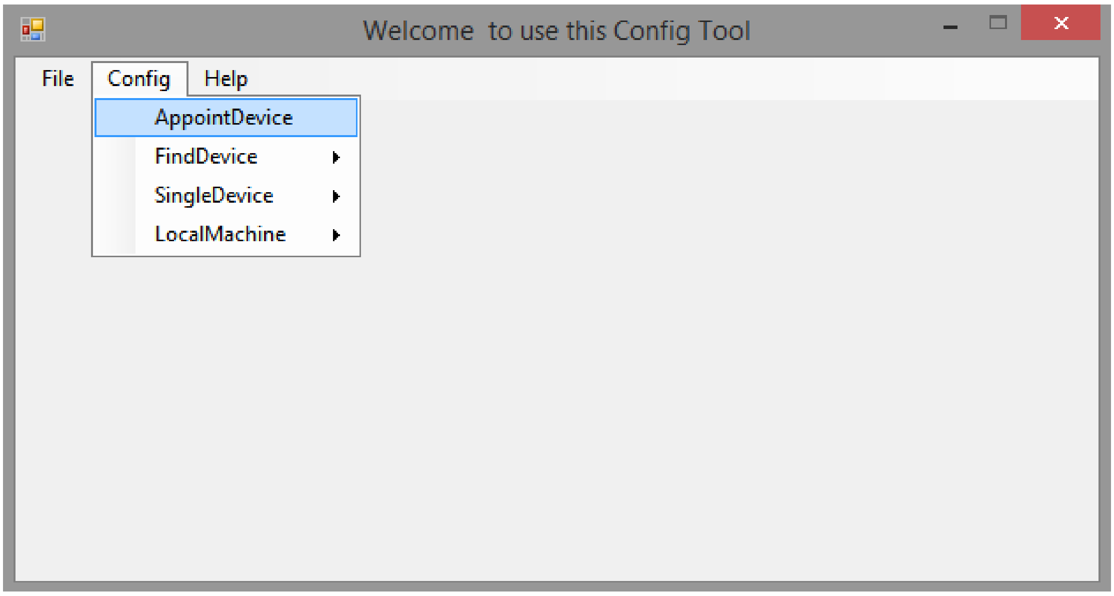
domain name:

ip address: 0 . 0 . 0 . 0

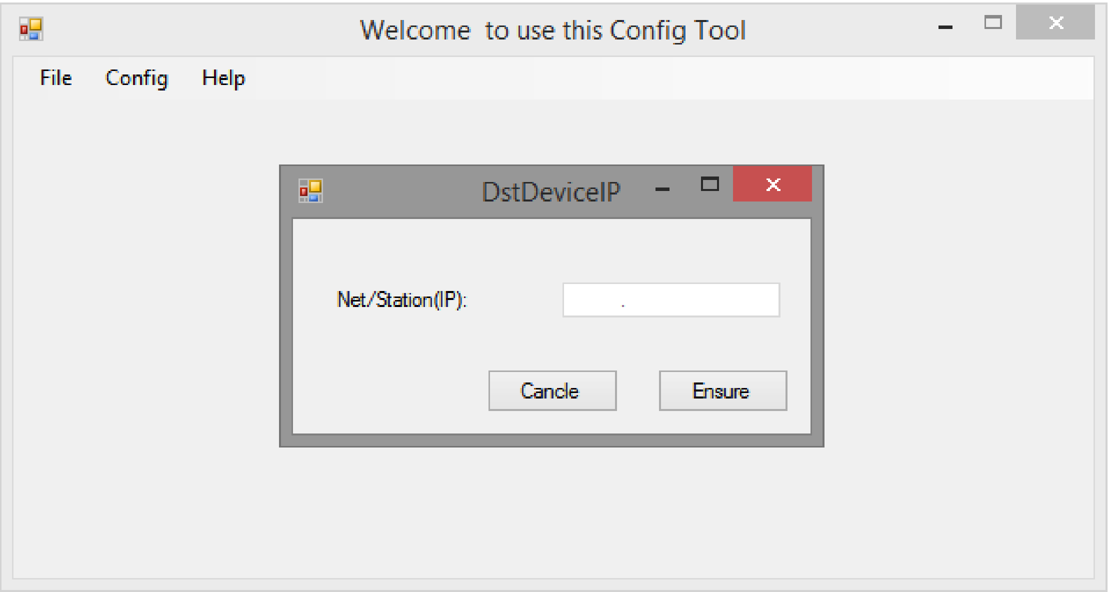
dev password:

3-1-4. Ethernet configuration of XINJE Config

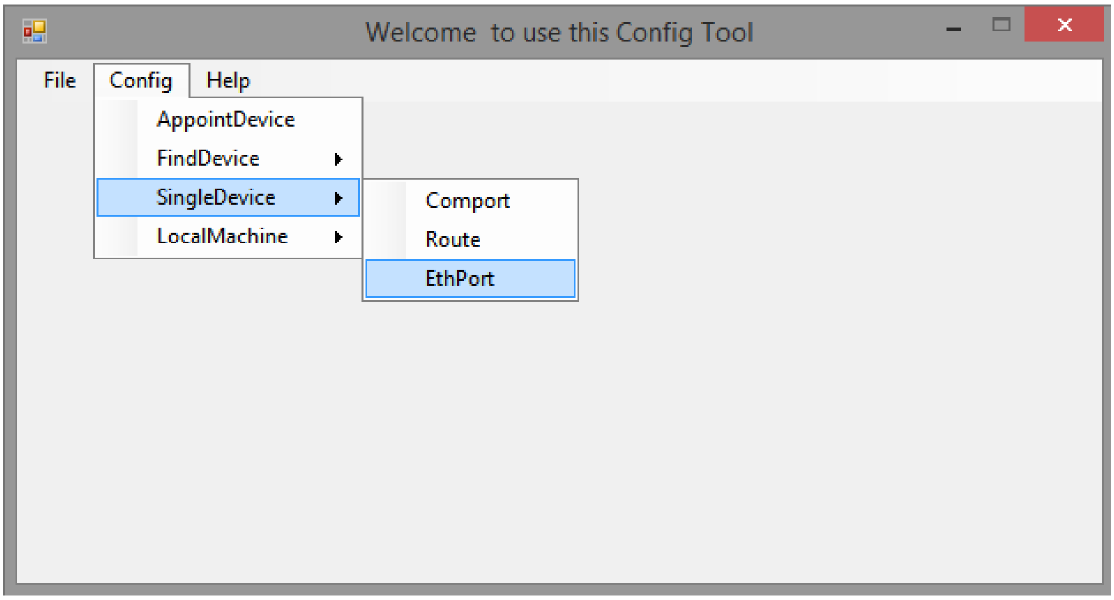
Please click config/appoint device in the XINJE config tool to search the device IP.

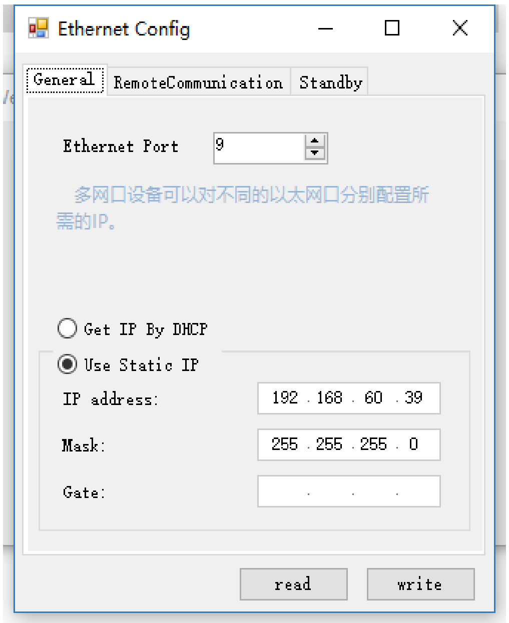


It will show below window, please input correct IP address of the device. (note: if there are more than one PLC has the same IP in the same network, it will connect any of them)



Click config/single device/ethport. The configuration explanation please refer to chapter 3-1-2. Both this config tool and the XDPPro software can configure the Ethernet parameters.





Ethernet Port 9

多网口设备可以对不同的以太网口分别配置所需的IP。

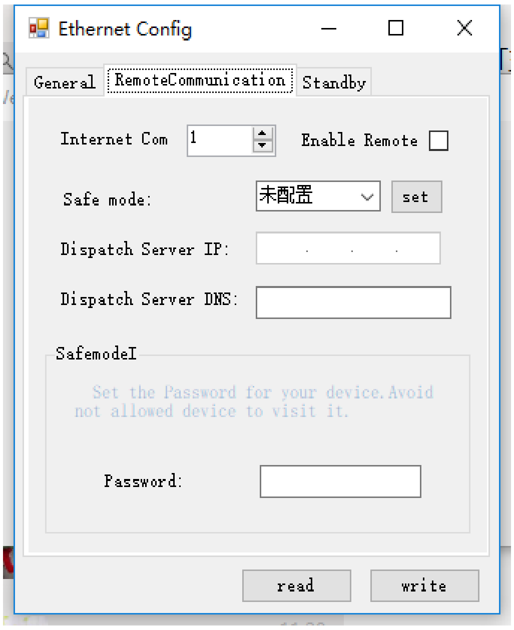
○ Get IP By DHCP

● Use Static IP

IP address: 192 . 168 . 60 . 39

Mask: 255 . 255 . 255 . 0

Gate: . . .



Internet Com 1 Enable Remote □

Safe mode: 未配置 set

Dispatch Server IP: . . .

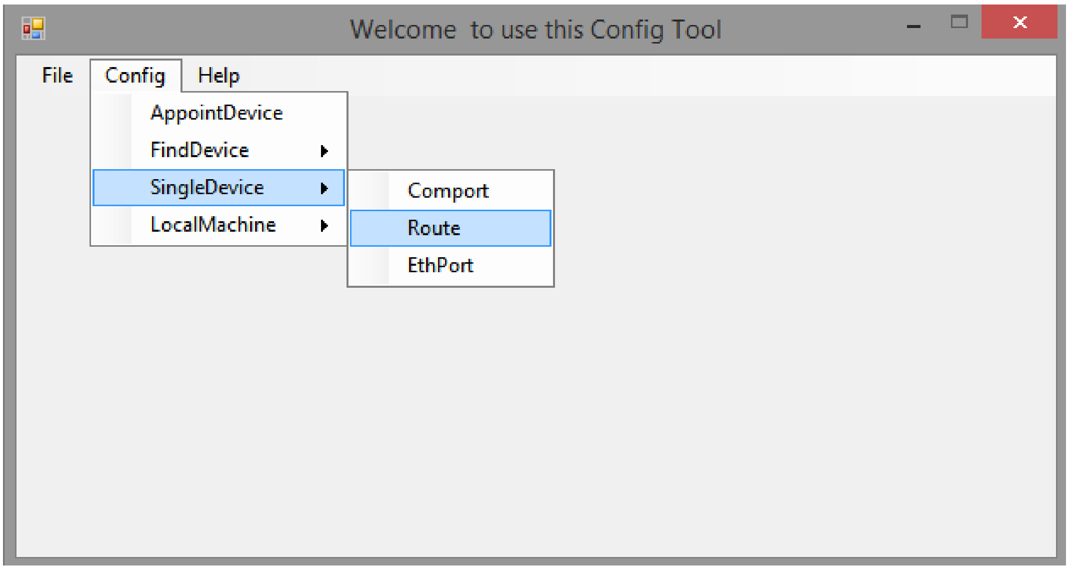
Dispatch Server DNS:

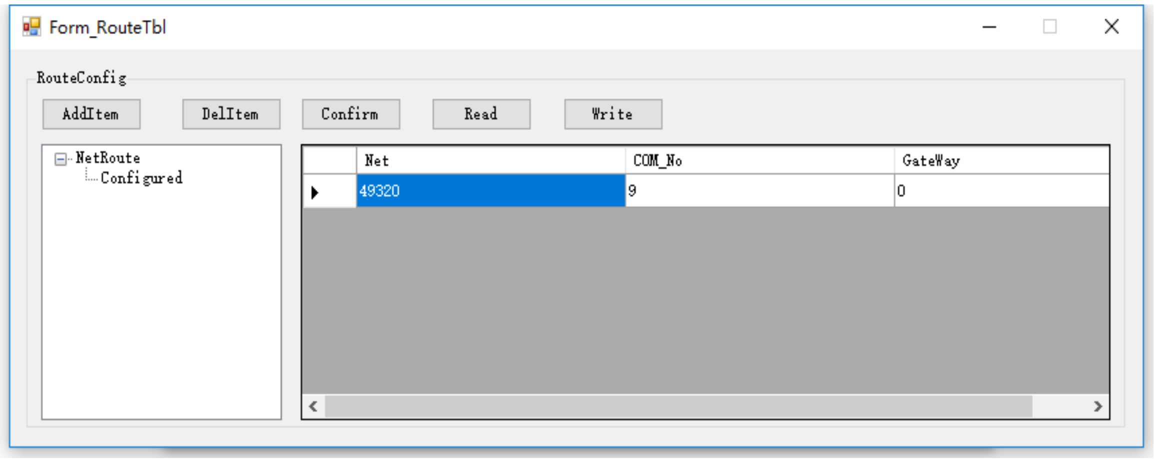
SafemodeI

Set the Password for your device.Avoid not allowed device to visit it.

Password:

Click config/single device/route to set the PLC router information. The details please refer to chapter 3-1-2. PLC has one defaulted routing information which can be checked when reading the routing table.





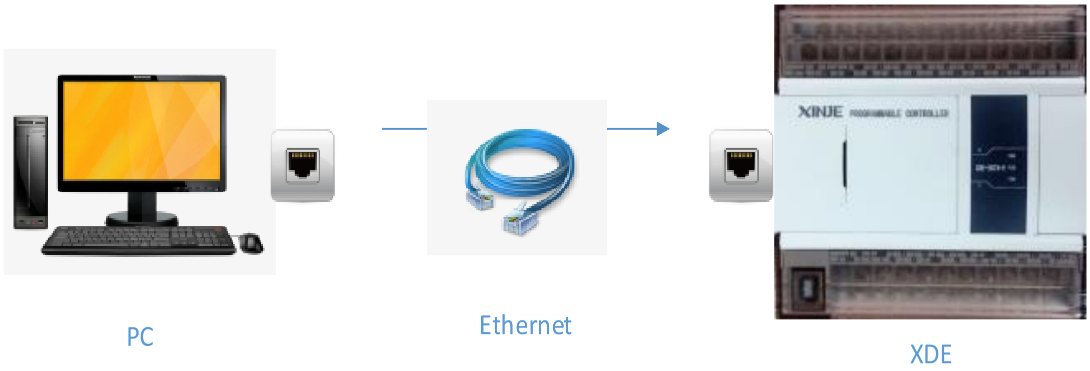
Net	COM_No	GateWay
49320	9	0

3-2. Wiring method

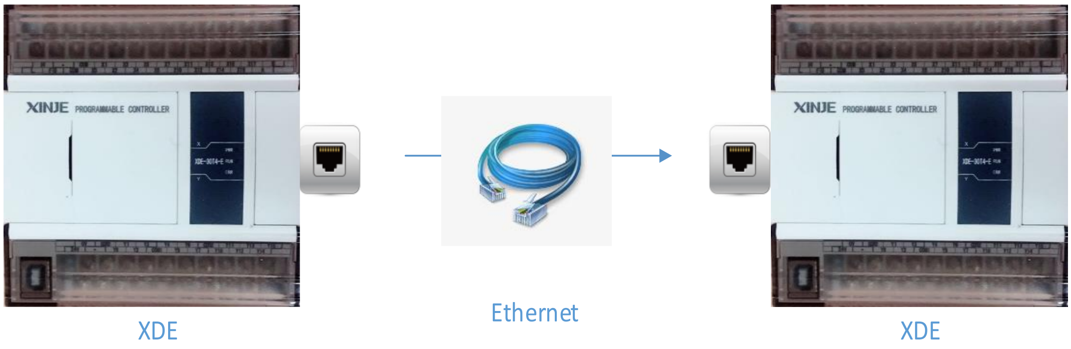
XDE/XD5E/XG1 series Ethernet port PLC uses Ethernet communication when the physical interface is RJ45, connection mode includes direct connection mode and transfer connection mode. When wiring, CAT5e UTP and STP network cables are selected. The length of a single network cable is not more than 100 meters. The switch type is recommended for a gigabit switch, such as a TP-link gigabit switch. After the completion of the connection, the physical connection can be determined by observing the status of indicator light at RJ45 port on the PLC. If the orange indicator light is always on and the green one is flashing, the connection of Ethernet port will be effective.

3-2-1. Direct connection mode

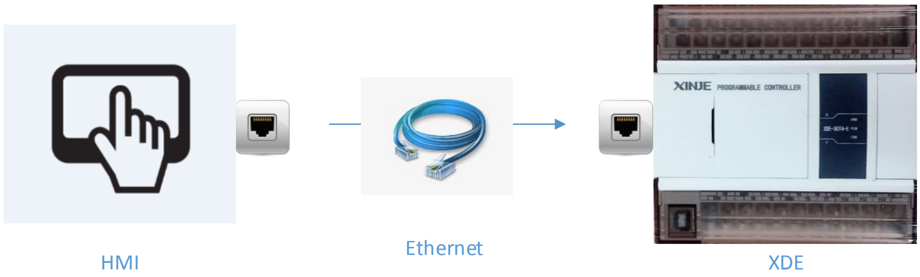
Direct connection is to connect the RJ-45 port of two devices directly with one network cable. This connection method is only applicable for one-to-one communication between two devices, such as PLC and XDPPro communication or two PLC communication. In this connection mode, the equipment which doesn't provide DHCP service needs to be set as fixed IP. The specific wiring diagram is as follows:



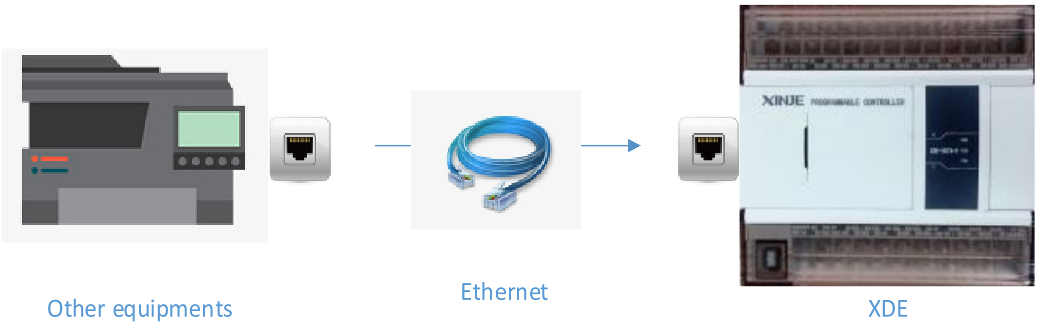
PLC and PC



PLC and PLC



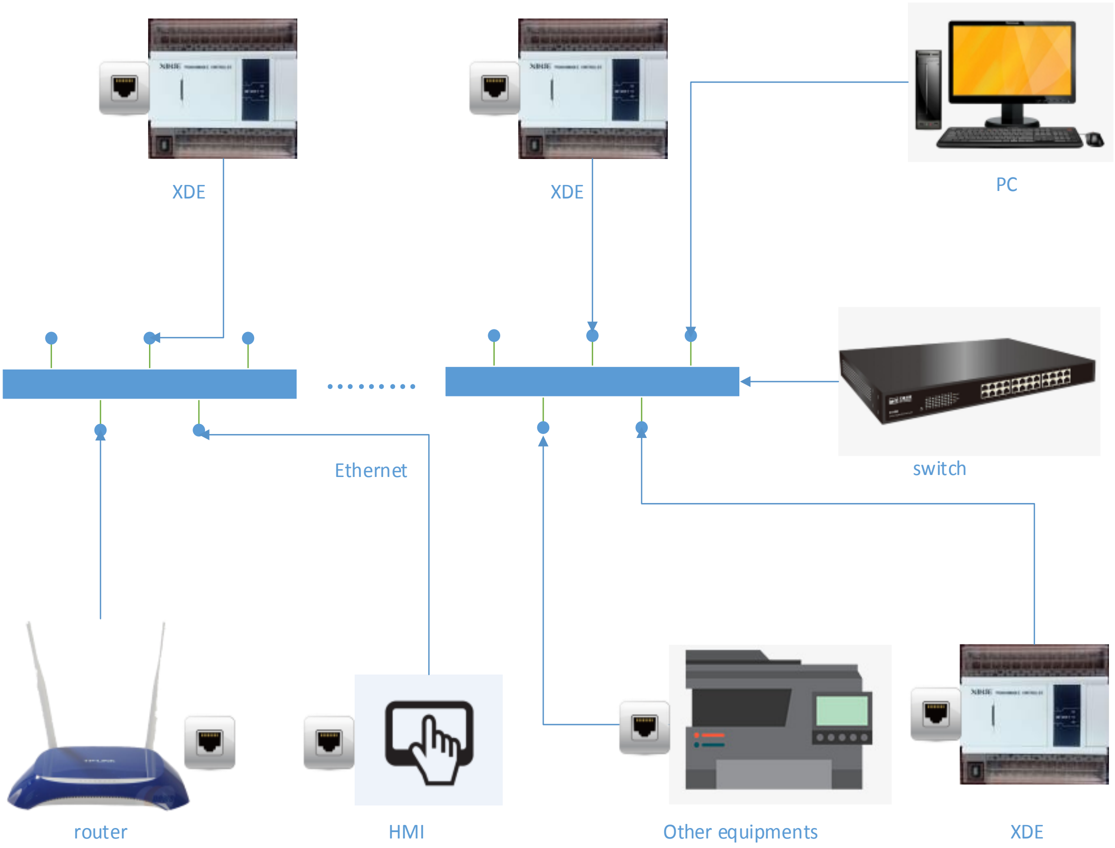
PLC and HMI



PLC and other equipments

3-2-2. Tranfer connection mode

The transfer connection mode is applicable to the multi-device network communication. All communication devices are connected through the switch. The connection mode is each PLC (including other devices involved, such as the PC, HMI and router) RJ-45 interface is connected to the RJ-45 interface on the switch through the Ethernet cable. The specific wiring diagram is as follows:



Trasfer connection mode

When the router is included in the networking equipment and PLC wants to realize remote communication in the external network through the router, it needs to choose the "remote communication" option. If the IP parameters of PLC are set to manual filling, it needs to fill in the gateway as the IP address of the router.

3-3. Communication instruction

Ethernet communication also uses X-NET instruction, which are bit read/write, register read/write, the instruction object is XDE/XD5E/XG series PLC. The Ethernet communication object is confirmed by IP address, there is a little different for using target network no. and target station no. The following will explain in details.
Note: X-NET instruction please refer to chapter 1-2.

3-3-1. Operand explanation in communication instruction

1. Target network number

The IP address is a four-byte address, and the target network number is the first two bytes of the IP address.

Example: the IP address of the object PLC is 192.168.6.60, and the first two bytes of this IP address are converted into corresponding hexadecimal system respectively. Byte 1 is stored in the high 8-bit address, while byte 2 is stored in the low 8-bit address (note: if byte 1 first bit is letter, please add 0 before the letter), as shown in the following table:

	Byte 1	Byte 2	Byte 3	Byte 4
IP address	192	168	X	X
Hex format	0xC0	0xA8		
Target network number	H0C0A8			

2. Target station number

The IP address is a four-byte address, and the target station number is the last two bytes of the IP address.

Example: the IP address of the object PLC is 192.168.6.60, and the last two bytes of this IP address are converted into corresponding hexadecimal system respectively. Byte 3 is stored in the high 8-bit address, while byte 4 is stored in the low 8-bit address (note: if byte 3 first bit is letter, please add 0 before the letter), as shown in the following table:

	Byte 1	Byte 2	Byte 3	Byte 4
IP address	X	X	6	60
Hex format			0x06	0x3C
Target station number			H63C	

3. Target object type

The type of coil or register in the target device in a communications network.

For example, if the target device communicates with coil X, the target object type is K1

The target device communicates with coil Y, and the target object type is K2

The target device communicates with coil M, and the target object type is K3

The target device communicates with coil HM, and the target object type is K8

The target device communicates with register D, and the target object type is K128

The target device communicates with register HD, and the target object type is K136

4. Target object address

The coil or register address in the target device in a communication network.

Example: Write bit M0: writes the native M0 state to the specified address in the target station number

Write register D0: writes the native D0 value to the specified address in the target station number

Read bit M1: read the specified address state in the target station number to the native M1

Read register D1: read the contents of the specified register in the target station number to the native D1

5. Access object number

The first coil/register address when the machine reads and writes to the target station number. It is generally used in combination with the "number of objects".

For example, if PLC1 wants to read the values of D0~D5 in PLC2, the target object address is the communication address of PLC2 register D0~D5, and the number of access objects is K6.

6. Local object

Coil/register in the machine that needs to be transmitted to the target station.

For example, if PLC1 wants to transfer the value of D0 to specified address of PLC2, the local object is D0.

7. Communication condition

The condition for Ethernet communication can be normally on/off coils and up/down edges. When the normally on/off coil is triggered, the Ethernet instruction is always executed, and each instruction takes more than or equal to 2ms to execute. The maximum throughput of 10ms of PLC Ethernet port data is 5000 words. For multiple slave station communication or large amount of communication data, the phenomenon of slow communication may occur. The oscillation coil can be used as triggering condition.

When the up/down edge is triggered, the Ethernet instruction is executed only once, and only when the next up/down edge is reached, will the Ethernet instruction be executed again.

3-3-2. Communication example

Example: two XDE PLC Ethernet communicate with each other through the RJ45 port.

(1) write the values of the 10 registers of D0-D9 of no. A PLC to D0-D9 of no. B PLC;

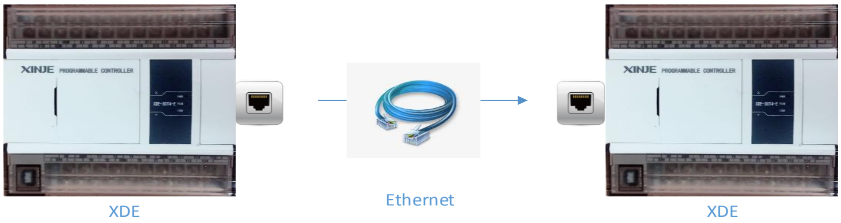
(2) write the values of the 10 registers of the M0-M9 of no. A PLC to the M0-M9 of no. B PLC;

(3) read the values of the 10 registers of the D0-D9 of PLC no. B into the D10-D19 of PLC no. A;

(4) read the values of the 10 registers M0-M9 of PLC no. B into PLC no. A's M10-M19.

Step 1: wiring

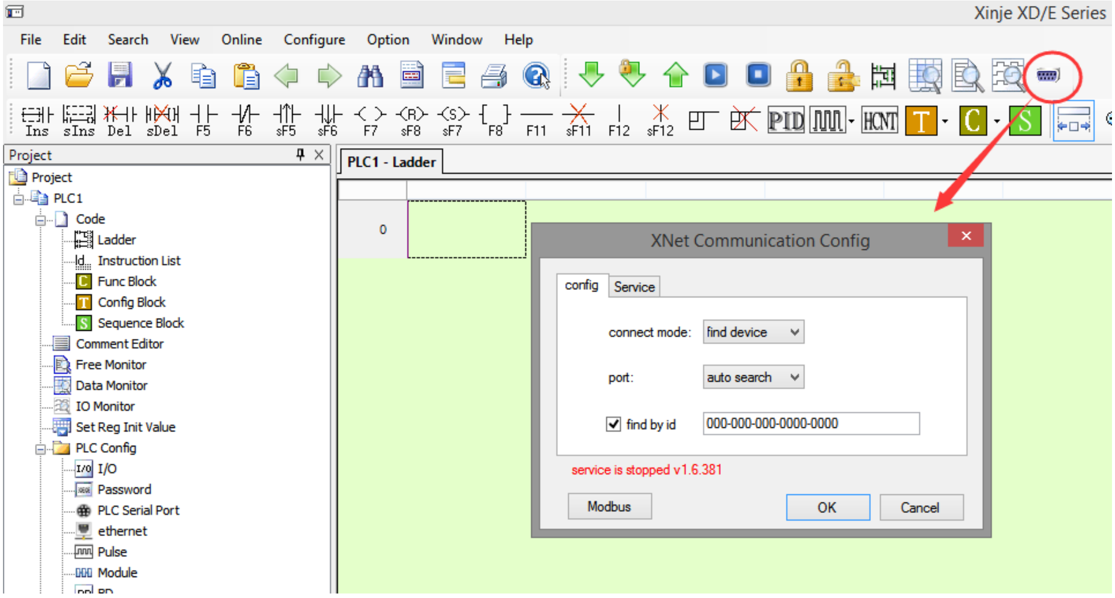
Connect the two PLC RJ45 port with CAT5e Ehternet cable.



Step 2: Ehternet parameter setting

1. Use the USB download cable to connect the PLC of no. A to the computer.

2. Open the PLC programming software to connect to PLC through Xnet, and set the IP of PLC as 192.168.6.10 as shown in the figure below. Please cut the PLC power and power on again after setting the parameters.



XNet Communication Config

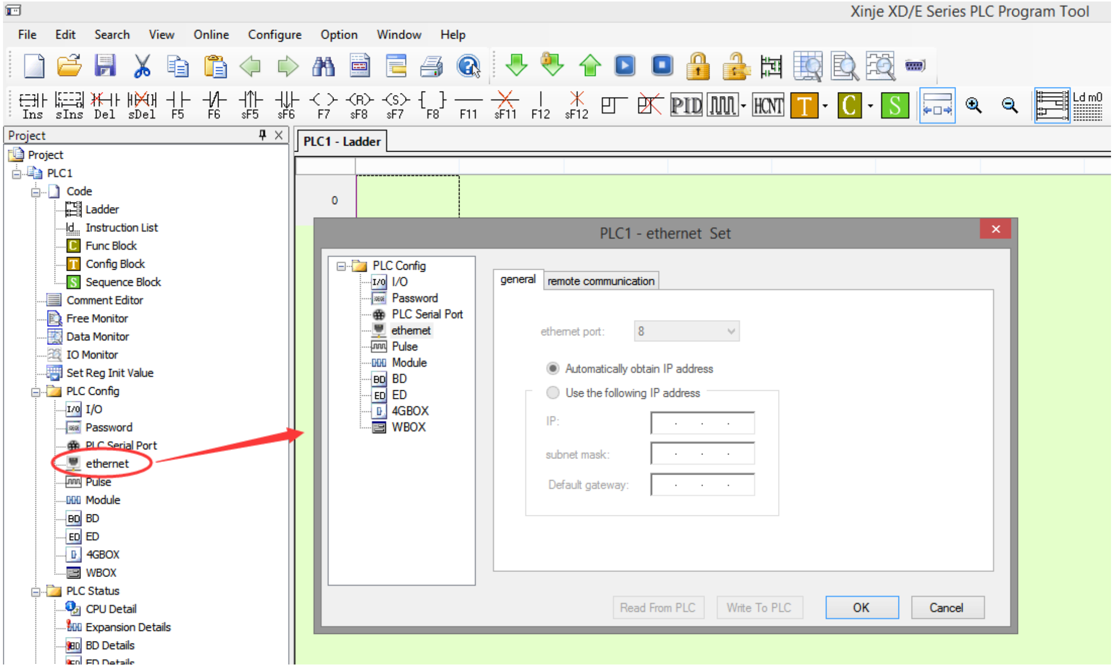
config | Service

connect mode: find device

port: auto search

☑ find by id 000-000-000-0000-0000

service is stopped v1.6.381



PLC1 - ethernet Set

general | remote communication

ethernet port: 8

● Automatically obtain IP address

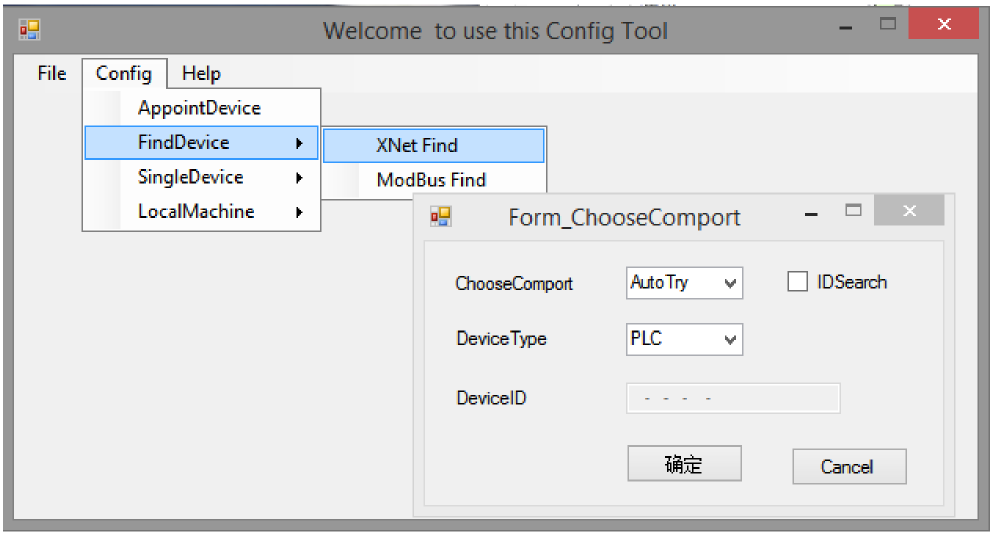
○ Use the following IP address

IP: . . .

subnet mask: . . .

Default gateway: . . .

3. Open the XINJEConfig software, connect PLC to it through X-NET mode.



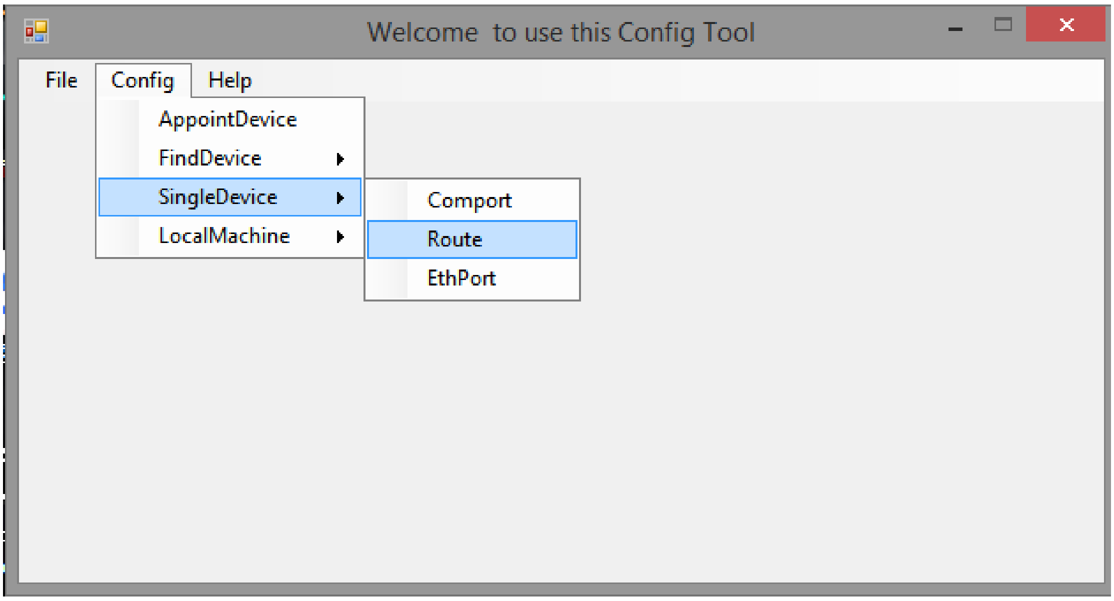
ChooseComport: AutoTry

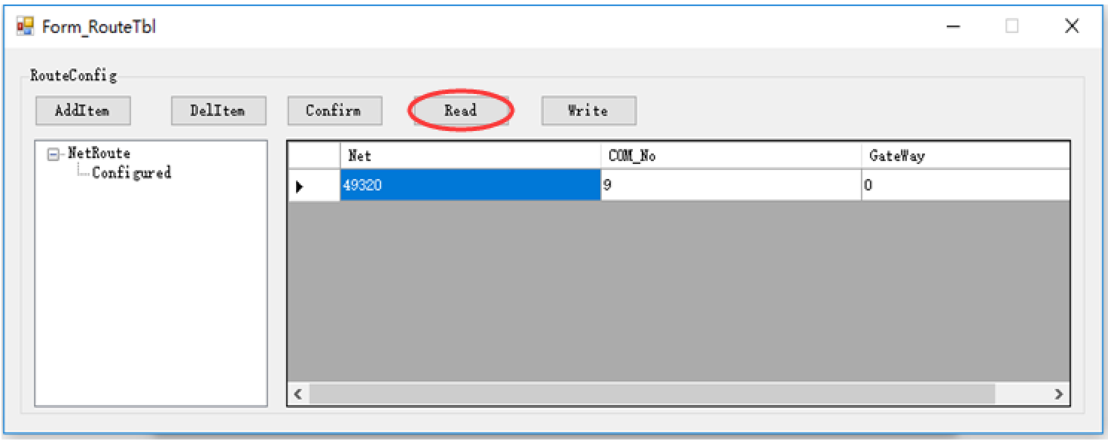
□ IDSearch

DeviceType: PLC

DeviceID: - - - -

4. Click config/single device/route, then click read, it will show default router information, if there is no information, please add as below. Please cut the power and power on again after setting. PLC no.A Ethernet configuration is finished.

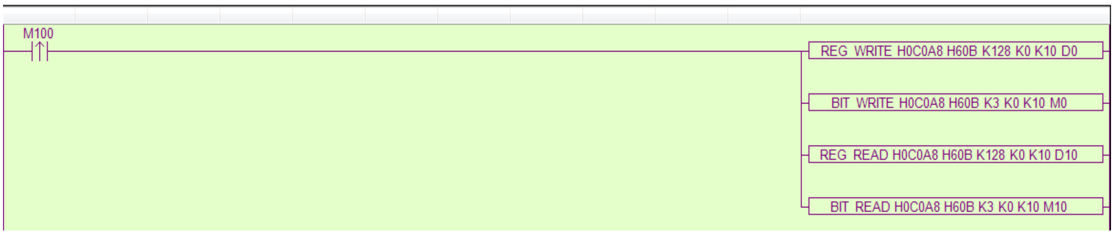




Net	COM_No	GateWay
49320	9	0

5. Set PLC no.B IP address as 192.168.6.11. the Ehternet configuration is same to PLC no.A.

Step 3: make the program



```
M100
REG WRITE H0C0A8 H60B K128 K0 K10 D0
BIT WRITE H0C0A8 H60B K3 K0 K10 M0
REG READ H0C0A8 H60B K128 K0 K10 D10
BIT READ H0C0A8 H60B K3 K0 K10 M10
```

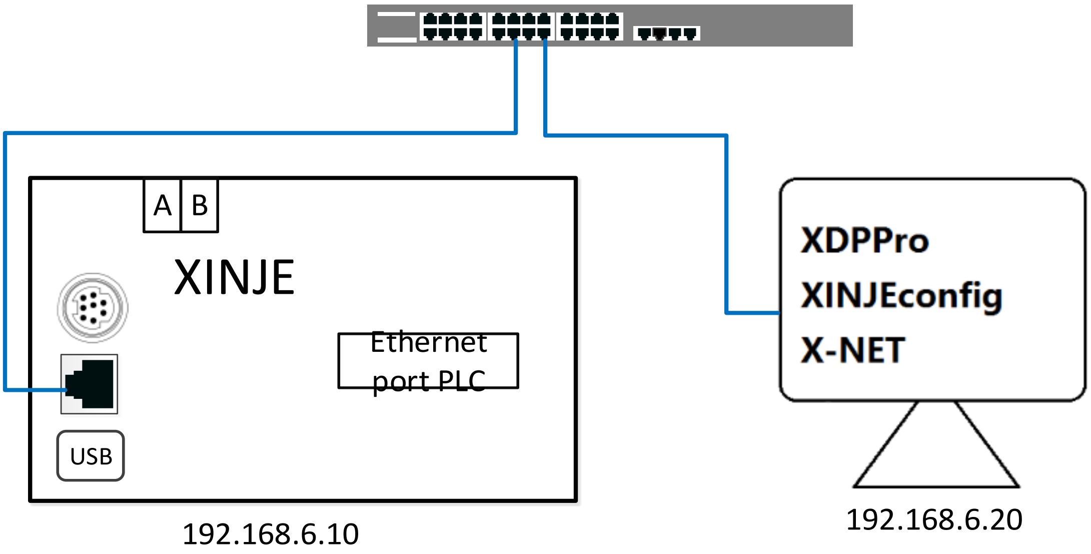
In the program, when the coil M100 from OFF to ON once, PLC no.A will read and write the register and coil of PLC no.B once.

3-4. LAN connection

In LAN environment, XDE/XD5E/XG and other Ethernet port PLC can communicate with three types of network equipment through PLC RJ45 port. The first is to communicate with the PLC programming software XDPPro through the x-net protocol, which is convenient for users to edit and debug PLC program. The second is communication with Ethernet HMI and industrial SCADA through Modbus_TCP protocol. PLC serves as Modbus_TCP slave device to facilitate data collection in industrial field. The third is other PLC communication through the x-net protocol and x-net communication instruction.

3-4-1. Communicate with XDPPro

In a LAN environment, there are two ways for programming software to connect Ethernet: "specify the address" and "search the device ID." Connect the computer installed XDPPro and the Ethernet port PLC with the same switch, the IP address of the Ethernet port PLC is set to 192.168.6.10, and the IP address of the computer is set to 192.168.6.20. Taking this case as an example, the two connection modes are described in details.

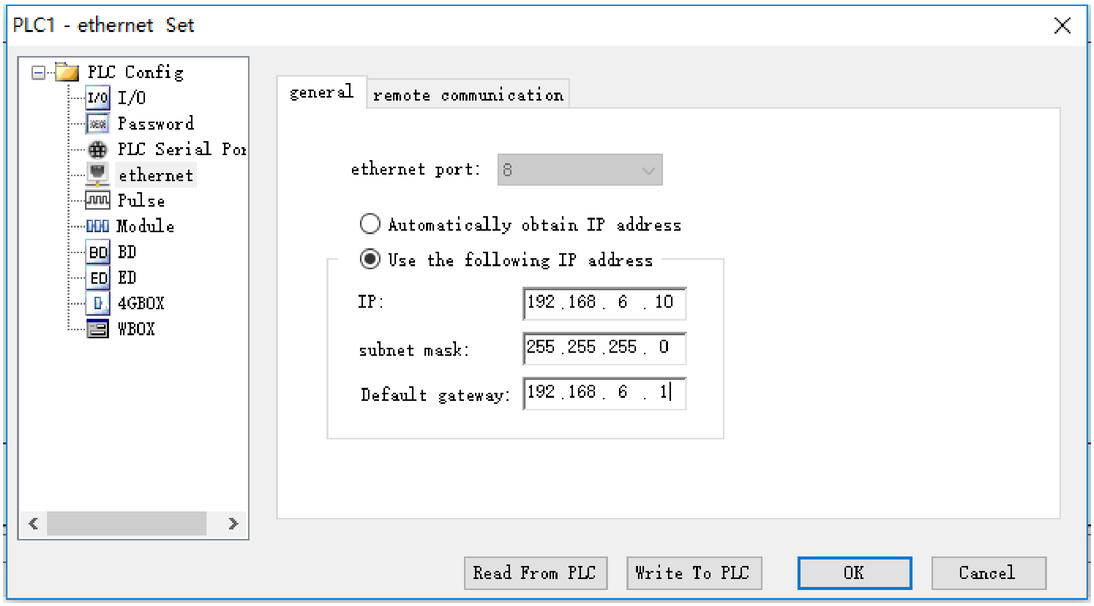


Step 1: wiring

Connect Ehternet PLC and switch, connect PC and switch through CAT5e Ehternet cable.

Step 2: set the IP address of PLC and PC

1. Refer to chapter 3-1-3 to set the PLC IP.



PLC1 - ethernet Set

ethernet port: 8

○ Automatically obtain IP address

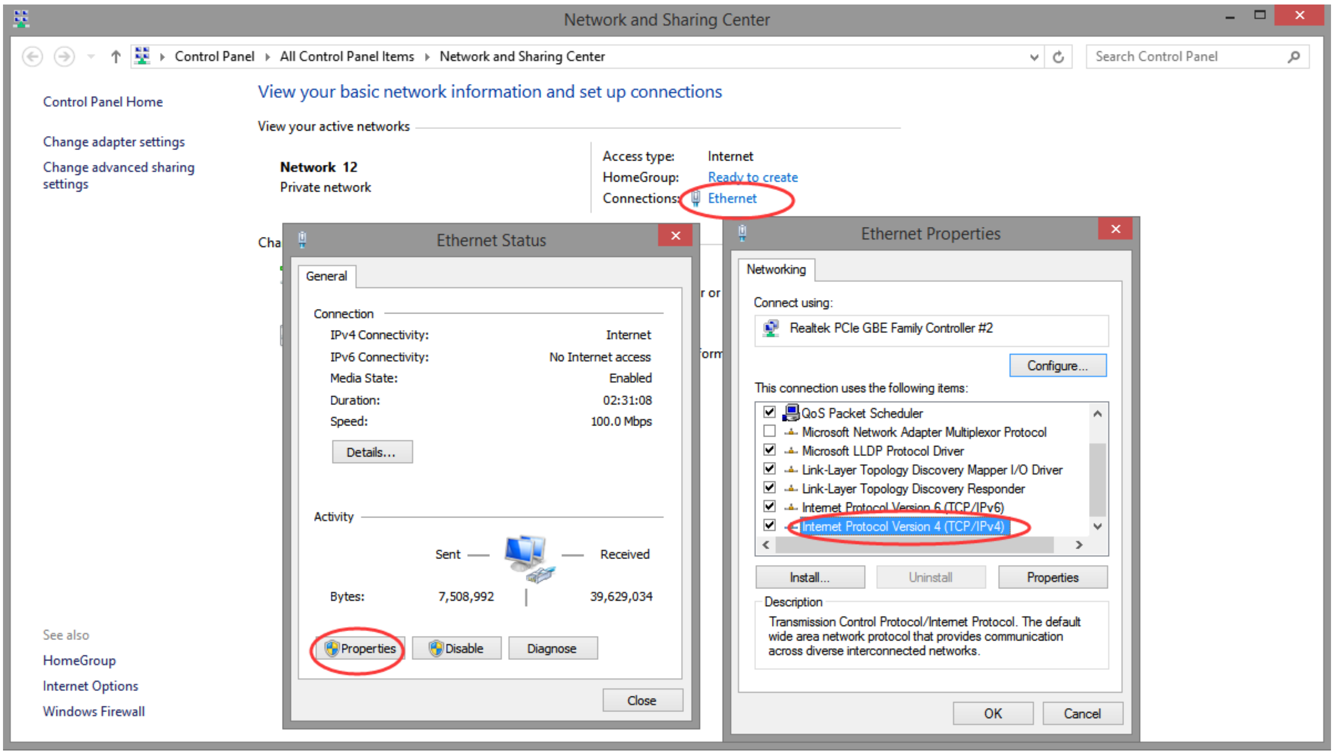
● Use the following IP address

IP: 192 . 168 . 6 . 10

subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 6 . 1

2. Click the network icon in the PC, right click it and open the network and sharing center.
3. Double click the local Ethernet connection, and double click properties, then click Internet protocol version 4. Double click it.



Network and Sharing Center

View your basic network information and set up connections

View your active networks

Network 12
Private network

Access type: Internet
HomeGroup: Ready to create
Connections: Ethernet

Ethernet Status

General

Connection
IPv4 Connectivity: Internet
IPv6 Connectivity: No Internet access
Media State: Enabled
Duration: 02:31:08
Speed: 100.0 Mbps

Activity
Sent — Received
Bytes: 7,508,992 | 39,629,034

Ethernet Properties

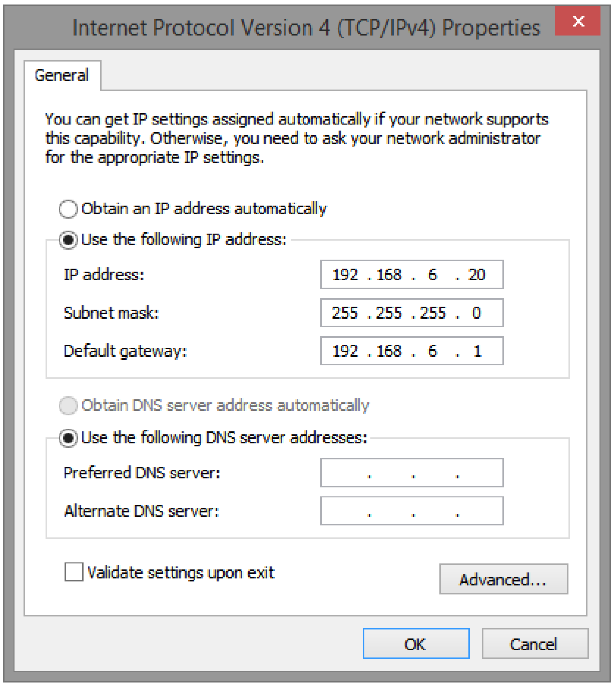
Networking

Connect using:
Realtek PCIe GBE Family Controller #2

This connection uses the following items:
☑ QoS Packet Scheduler
☐ Microsoft Network Adapter Multiplexor Protocol
☑ Microsoft LLDP Protocol Driver
☑ Link-Layer Topology Discovery Mapper I/O Driver
☑ Link-Layer Topology Discovery Responder
☑ Internet Protocol Version 6 (TCP/IPv6)
☑ Internet Protocol Version 4 (TCP/IPv4)

Description
Transmission Control Protocol/Internet Protocol. The default wide area network protocol that provides communication across diverse interconnected networks.

4. Set the IP address as below figure.



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 6 . 20

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 6 . 1

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

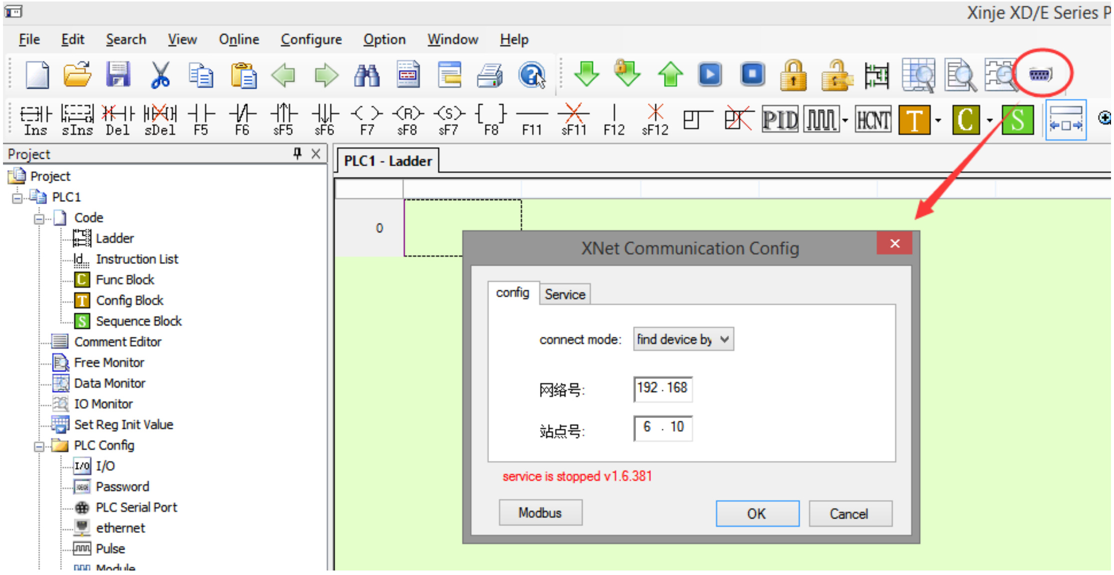
□ Validate settings upon exit

Step 3: monitor the PLC in XDPPro software through the Ehternet port.

There are two ways to mointor the PLC through Ehternet port.

1. Specified address

Open the XDPPro software, click serial port config/x-net protocol, choose find device by id, then fill in the network no. and station no. with PLC IP address. Click OK to connect.



XNet Communication Config

config | Service

connect mode: find device by

网络号: 192 . 168

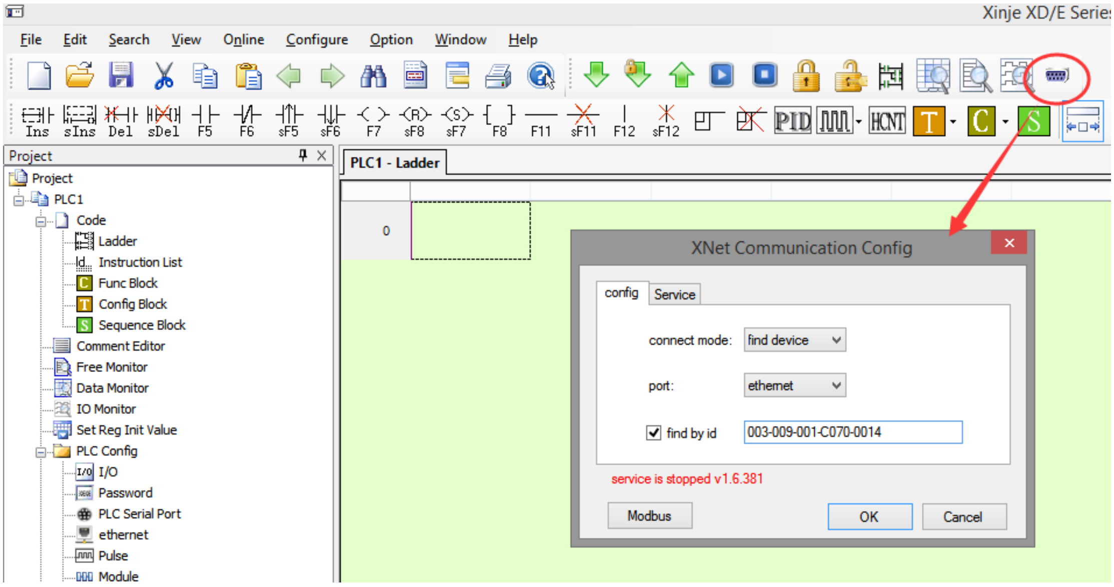
站点号: 6 . 10

service is stopped v1.6.381

2. Search the device ID

Open the XDPPro software, click serial port config/xnet protocol, choose find device/ethernet, then fill in the device ID of PLC(please check the ID on the PLC product label or click the PLC CPU detail in the software). Click OK to connect.

Note: A computer may have more than one network card. Please use only one network card when communicating with the PLC through Ethernet. The network card only sets one IP address.



XNet Communication Config

config | Service

connect mode: find device

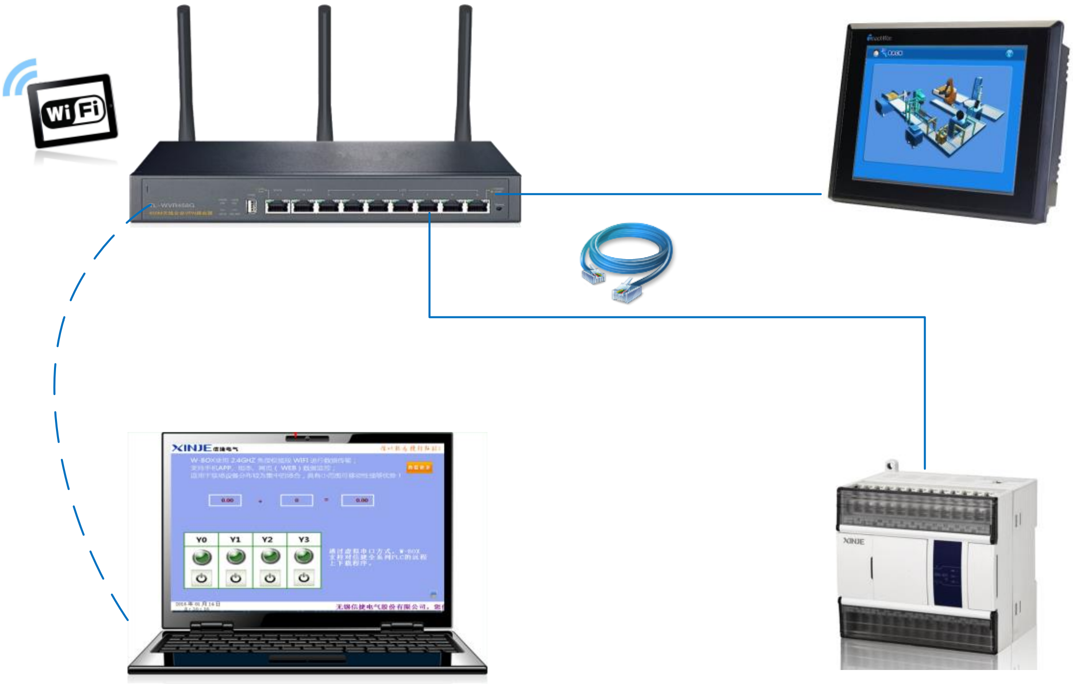
port: ethernet

☑ find by id 003-009-001-C070-0014

service is stopped v1.6.381

3-4-2. Communicate with Modbus_TCP device

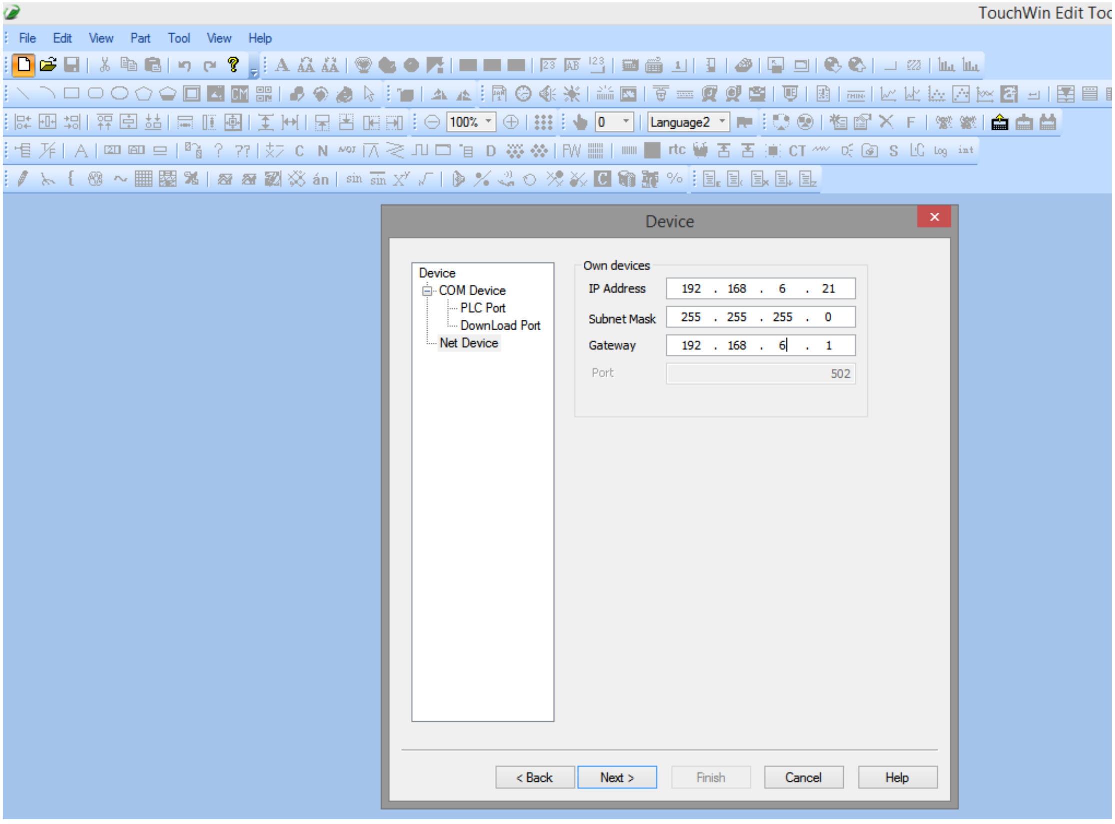
XDE/XD5E/XG series PLC supports the standard ModBus_TCP protocol and can communicate with the master station as ModBus_TCP slave station (Ethernet port PLC cannot be the master ModBus station). In this example, XDE series PLC is used as the controller, and TG765-ET series HMI is used as the experimental model and kingview as the SCADA software. The network connection structure is as follows.



Note: one Ehternet port PLC can support up to 4 TCP/IP connections, and communicating with 4 Modbus_TCP master station at the same time.

- Ethernet HMI

1. Build a new project, set the HMI Ethernet parameters. Please set the HMI, PC and PLC in the same network segment.



Device

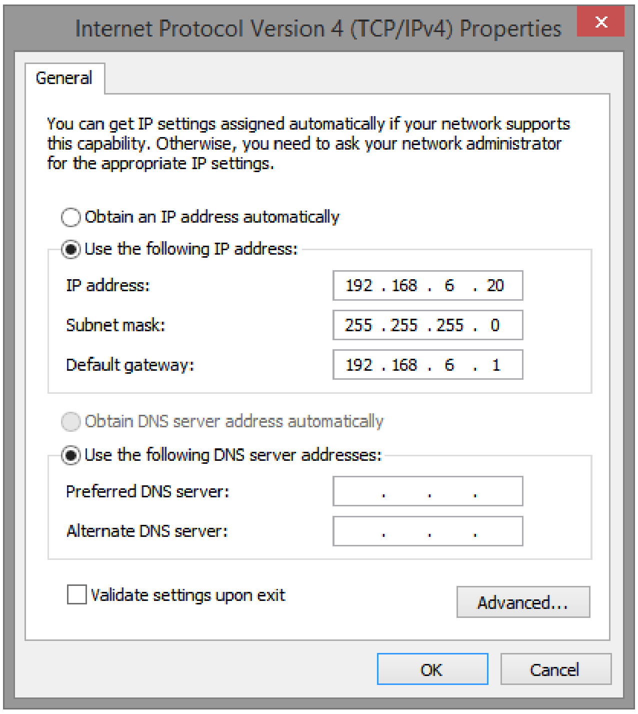
Own devices

IP Address 192 . 168 . 6 . 21

Subnet Mask 255 . 255 . 255 . 0

Gateway 192 . 168 . 6 . 1

Port 502



Internet Protocol Version 4 (TCP/IPv4) Properties

General

You can get IP settings assigned automatically if your network supports this capability. Otherwise, you need to ask your network administrator for the appropriate IP settings.

○ Obtain an IP address automatically

● Use the following IP address:

IP address: 192 . 168 . 6 . 20

Subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 6 . 1

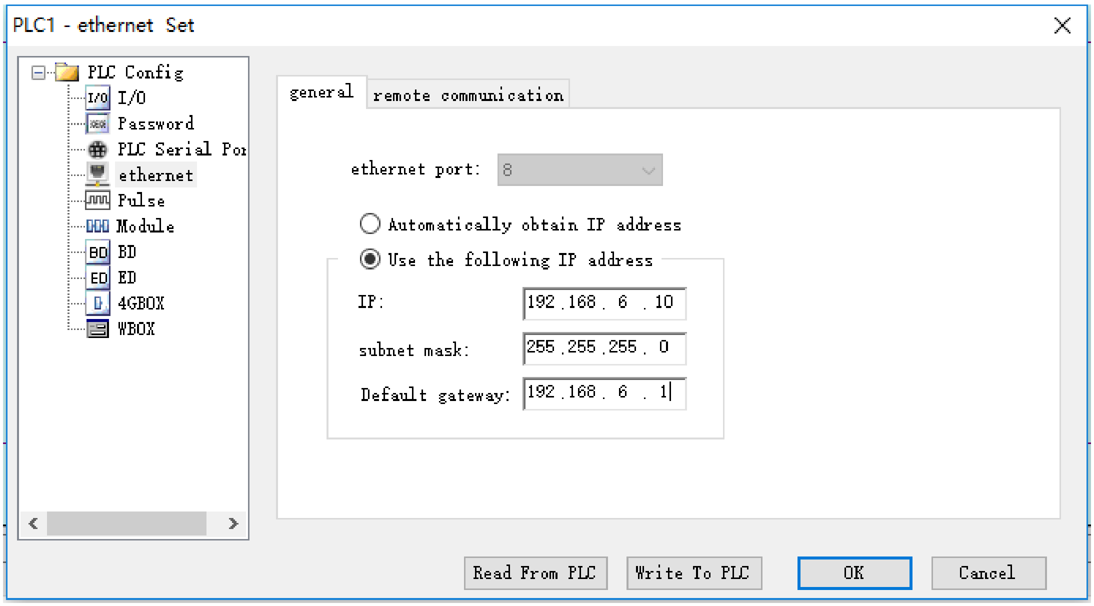
○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: . . .

Alternate DNS server: . . .

□ Validate settings upon exit



PLC1 - ethernet Set

ethernet port: 8

○ Automatically obtain IP address

● Use the following IP address

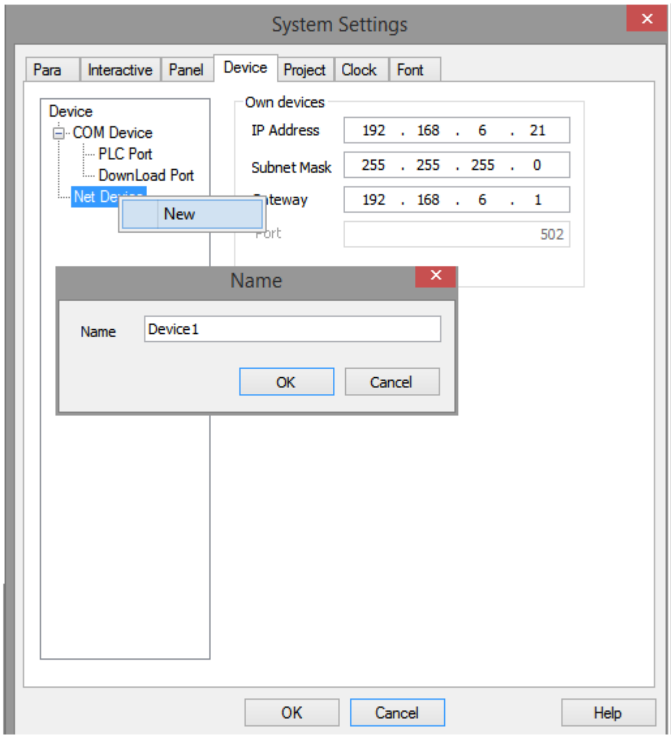
IP: 192 . 168 . 6 . 10

subnet mask: 255 . 255 . 255 . 0

Default gateway: 192 . 168 . 6 . 1

2. Right click the net device to build a new device, click Modbus_TCP protocol, fill in the IP address of communication object, other parameters please keep as default value. Click next until finish the setting.

   Note: one HMI can add up to 8 net devices.



System Settings

Own devices

IP Address 192 . 168 . 6 . 21

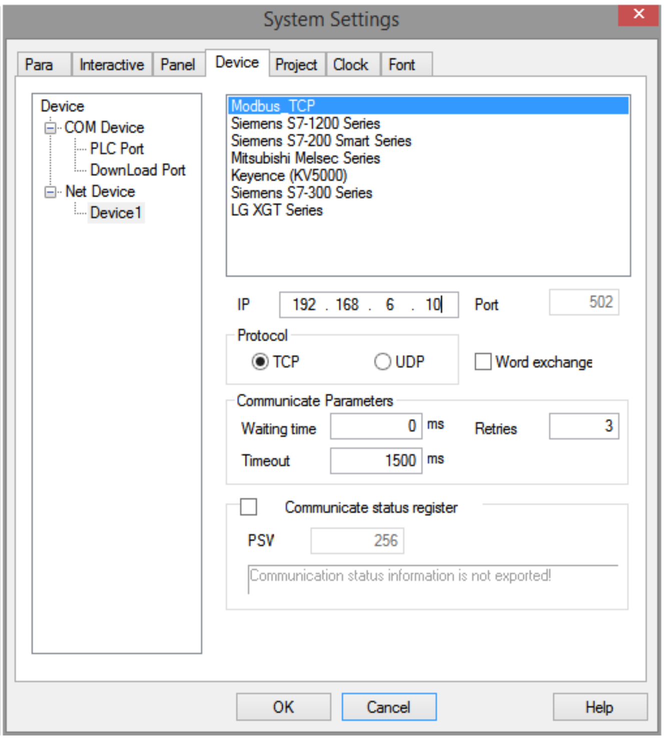
Subnet Mask 255 . 255 . 255 . 0

Gateway 192 . 168 . 6 . 1

Port 502

Name

Name Device1



System Settings

IP 192 . 168 . 6 . 10 Port 502

Protocol ● TCP ○ UDP □ Word exchange

Communicate Parameters

Waiting time 0 ms Retries 3

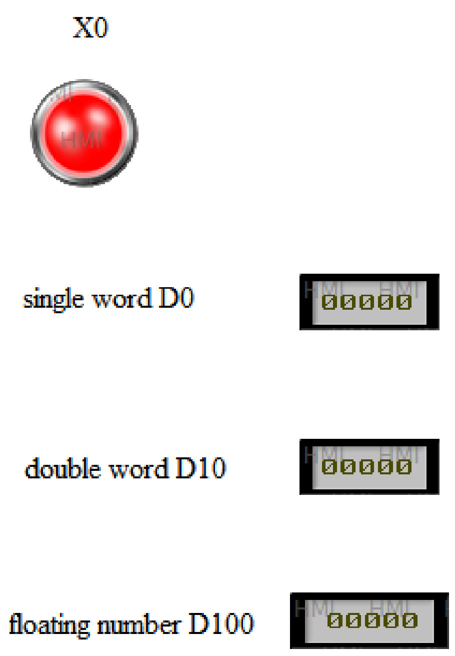
Timeout 1500 ms

□ Communicate status register

PSV 256

Communication status information is not exported!

3. In this example, we choose lamp button and data monitor as demonstrate object. HMI needs the communication device modbus address, please refer to XD/XL series PLC manual chapter 6-2-3.



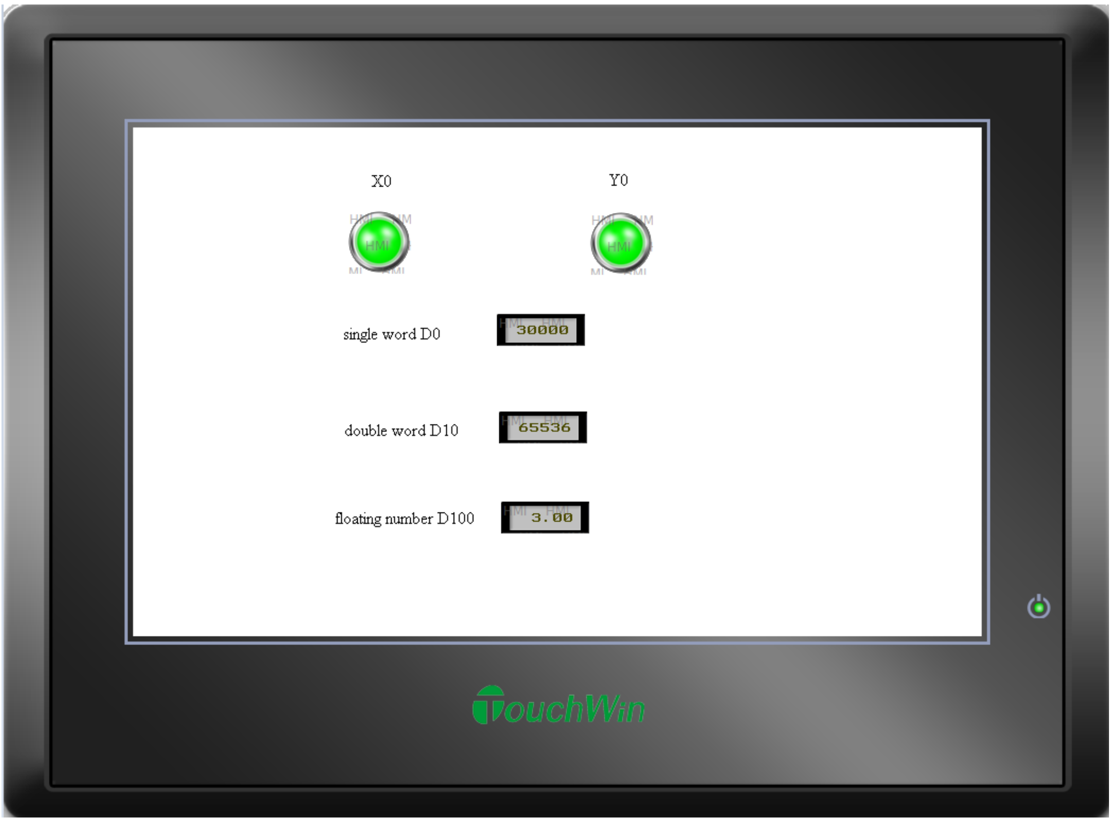
X0

Y0

single word D0 00000

double word D10 00000

floating number D100 00000



X0

Y0

single word D0 30000

double word D10 65536

floating number D100 3.00

3-4-3. Communicate with X-NET device

XDE/XD5E/XG series PLC can communicate with each other through RJ45 port, or with other x-net-enabled devices. The specific implementation process can be referred to chapter 3-3-2.

Note: when XDE/XD5E/XG series PLC communicate with other devices that support the x-net protocol, a PLC can communicate with up to 30 devices at the same time (the programming software will occupy the resources of one device when monitoring PLC through the Ethernet port).

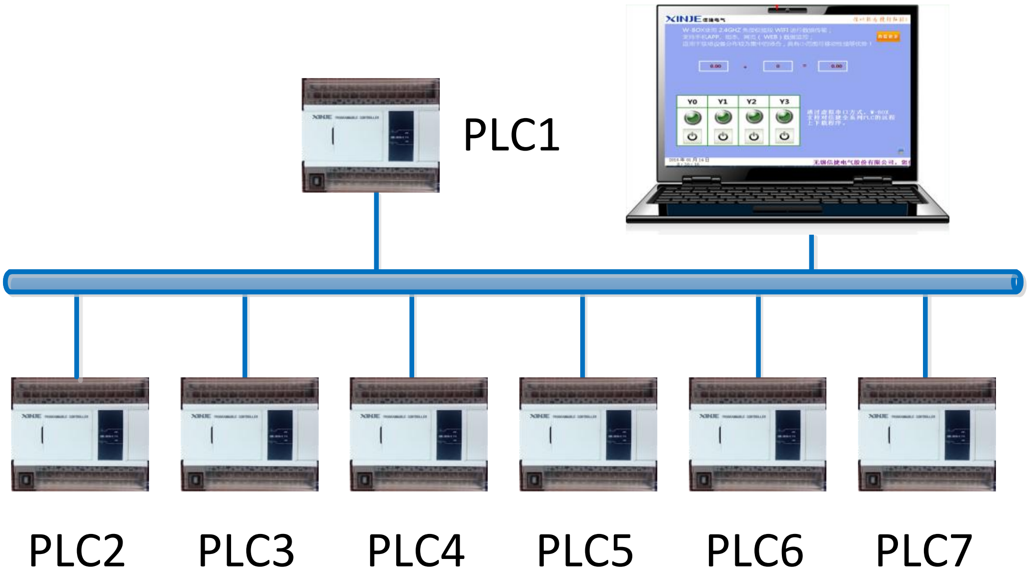
3-4-4. Communication networking mode

One master multi-slave network

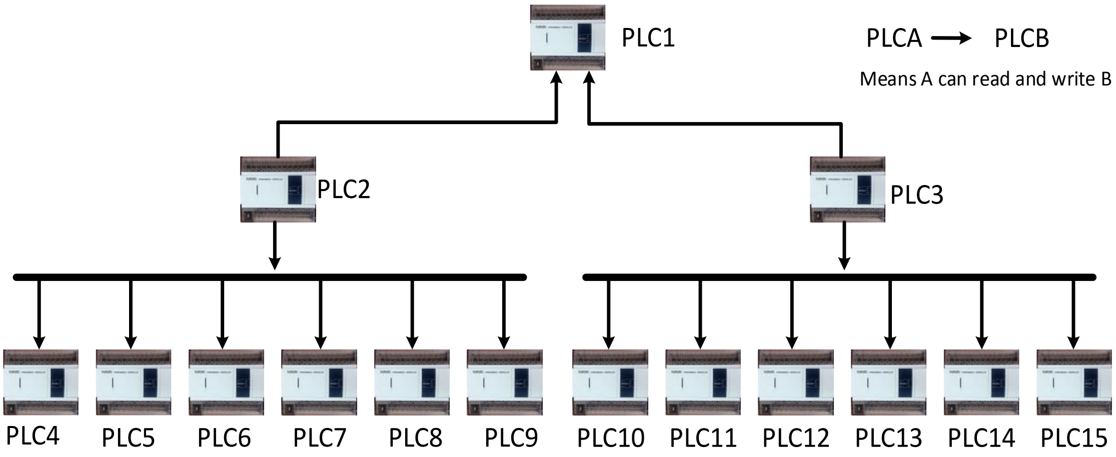
With the rapid development of industrial automation, the single machine capacity of industrial automation equipment and the process control scale of production system are increasing, the operating parameters are getting higher and higher. This requires that our automated production system must change from labor-intensive, equipment-intensive to information-intensive and knowledge-intensive. Distributed Control System (Distributed Control System) is the product of this transformation process. It is the result of the monitoring of production process, the development of Control technology and the comprehensive application of computer network technology.

In the LAN environment, XDE/XD5E/XG series PLC adopts the form of master and multi-slave network, and then carries out real-time data collection, historical storage and background analysis in the SCADA software, so as to achieve the purpose of function decentralization, location decentralization, centralized operation and management of the distributed control system.

The following figure shows a simple master and multi-slave network. PLC1 is master PLC, it can read and write data of PLC2 to PLC7. In addition, the data of all PLC can be monitored by the SCADA software.

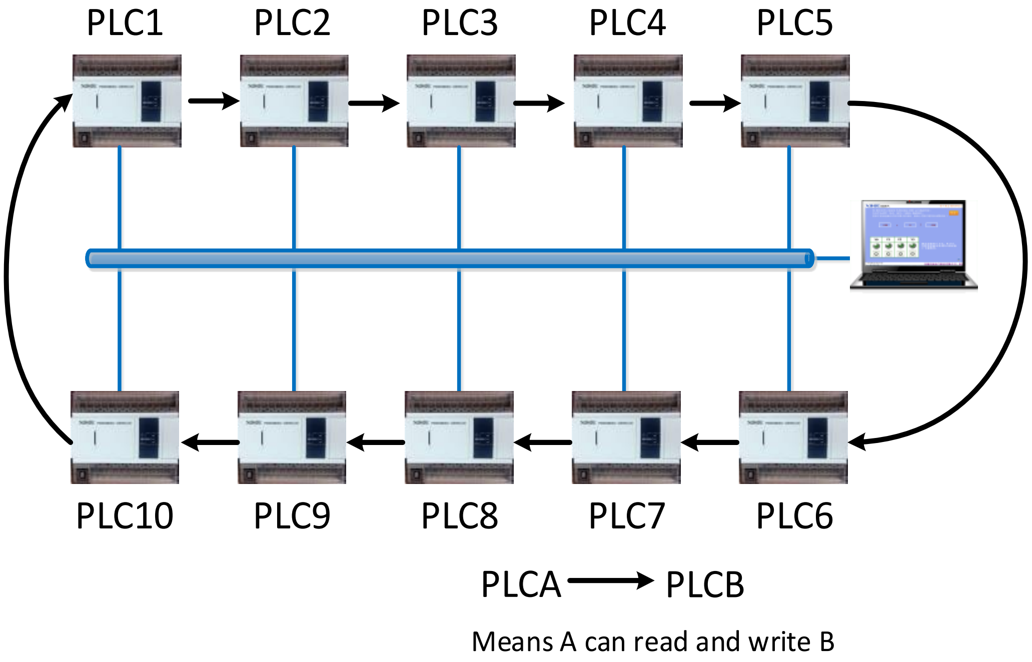


In a large industrial control site, there may be a situation where the number of PLC required is greater than 31, which exceeds the maximum number of equipment that a single PLC can communicate with. The problem can be solved by a reasonable combination of master and multi-slave network. As shown in the figure below, PLC2 and PLC3 are the master stations of actual communication, respectively collecting data from subordinate slave stations and transmitting their own data to PLC1. PLC1 acts as the logical master station of the whole system, which can control all PLC of the whole system.



Annular sequential control network mode

In the industrial automation field, most of them adopt pipeline production mode. Each production process is closely connected with the before and after production process, and there are more data interactions. Therefore, the annular sequential control network method can be adopted to improve the coordination of the whole production process. As shown in the figure below, it is a typical annular sequential control network. On this basis, it can also add SCADA software to realize real-time monitoring, historical storage and background analysis of data of each PLC.



3-5. Internet connection

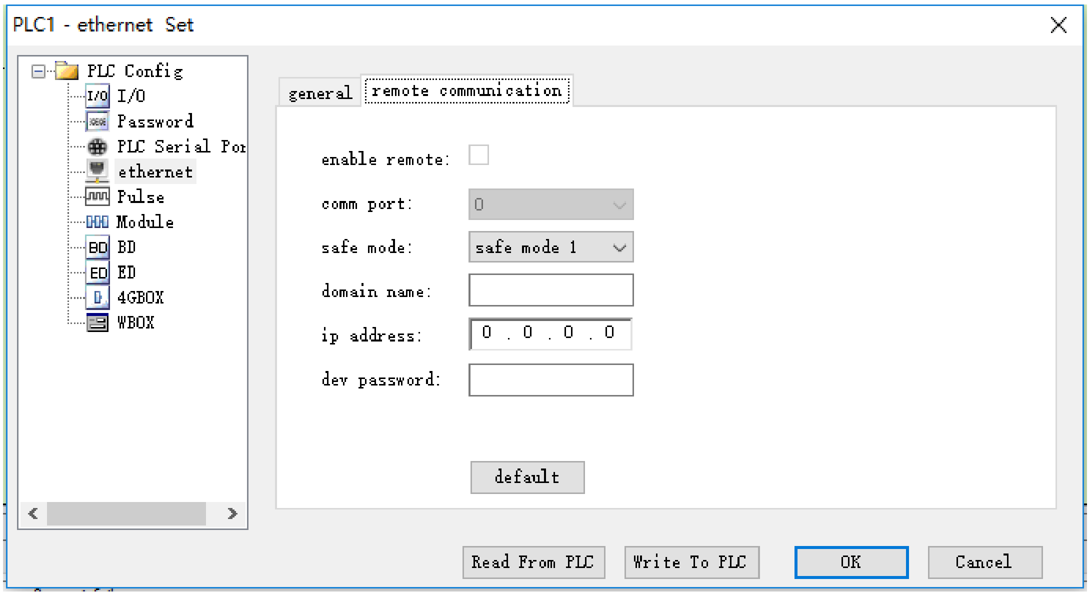
3-5-1. Remote communicate with XDPpro

Step 1 preparation

- XDPpro software (v3.5 and up)
- XDE/XD5E/XG series PLC (firmware v3.5 and up)
- PC (connected Internet)
- Router (connected Internet)
- Gigabit switch (optional)
- CAT5e Ethernet cable

Step 2 remote connection configuration

Open XDPpro software, click ethernet/remote communication, refer to chapter 3-1-2 for details. Here we fill in default value, the domain name is www.x-net.info. Then power on the PLC again.



PLC1 - ethernet Set

enable remote: □

comm port: 0

safe mode: safe mode 1

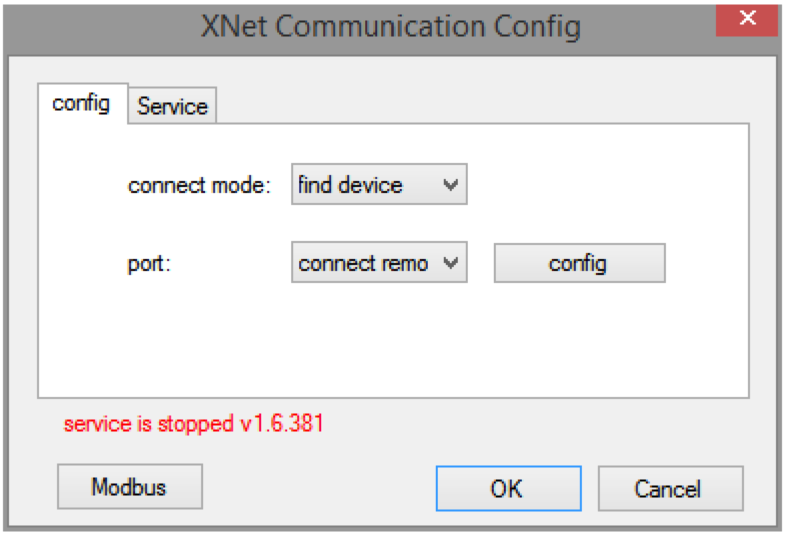
domain name:

ip address: 0 . 0 . 0 . 0

dev password:

Step 3 remote connection

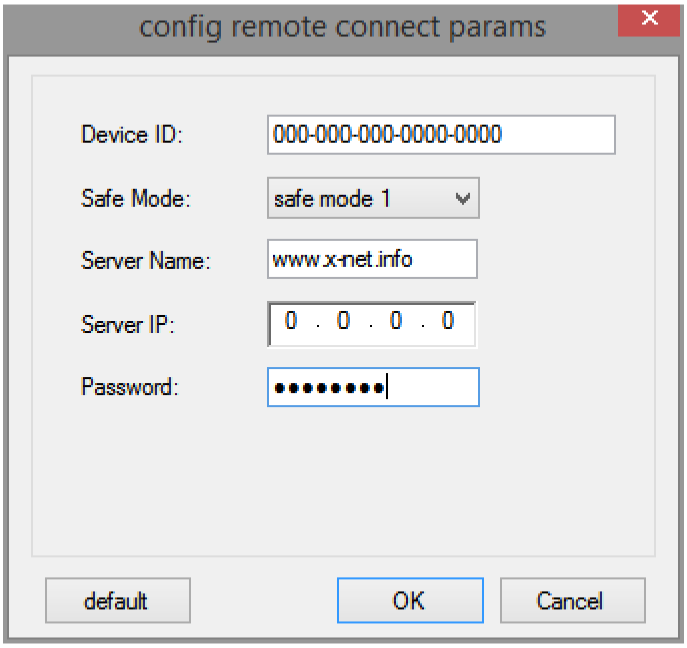
1. Click software serial port config, choose x-net protocol
2. Choose find device/connect remote
3. Click the config to show the remote connection parameters configuration window
4. Fill in the PLC parameters, click ok to finish the setting



connect mode: find device

port: connect remo

service is stopped v1.6.381



Device ID: 000-000-000-0000-0000

Safe Mode: safe mode 1

Server Name: www.x-net.info

Server IP: 0 . 0 . 0 . 0

Password: ••••••••

Step 4 solve the communication error

If it shows “the receiver has not logged in” or “connect remote device failure”, please confirm below items:

- the device ID is correct
- Server domain name and IP address are effective
- Device password is correct
- Flag bit SM1900 is ON
- Present PC network is good
- The Ethernet cable connection is well

After checking all the items, please power on again.

3-5-1. XINJE cloud

After the remote communication function of XDE/XD5E/XG and other Ethernet PLC is enabled, the device can be added to XINJE cloud web SCADA to realize the functions of remote data monitoring, data storage and data analysis. Below is the relevant information of the XINJE cloud web SCADA.
Web site: www.xinje.net:910
User name: xinje
Initial password: 85134136

3-6. Communication coil and register

Communication registers

Address	Format	Function	Explanation
SD194 (double word)	Decimal	X-Net communication result	X-Net instruction communication times
SD196 (double word)	Decimal		X-Net instruction communication failure times
SD198 (double word)	Decimal		X-Net instruction communication overtime times
SD1905	Hex	IP network no.	IP address first two bytes
SD1906	Hex	IP station no.	IP address last two bytes
SD1907	Hex	Subnet mask	Subnet mask first two bytes
SD1908	Hex		Subnet mask last two bytes
SD1909	Hex	Default gateway	Default gateway first two bytes
SD1910	Hex		Default gateway last two bytes

Communication coil

Address	Function	Note
SM1900	Remote connection flag	Set ON when the remote connection succeeded

3-7. Error and solution

When communicating with x-net, the programming software first establishes the connection and then finds the device. There are several types of error reporting as the following table.

Error code	Error type	Solution
Report error message when building the remote connection		
1	connect remote device failure	unstable network environment may lead to this, changing the environment to restart the software.
16	the sponsor did not log in	unstable network environment may lead to this, changing the environment to restart the software.
17	recipient not logged in	ID is error, BOX or XDE not logged in(check remote parameters and network)
18	use the safe mode inconsistent	Remote parameters of device or PC are not correct.
19	password error	Password is not correct. It is recommended to use 8-bit password.
255	login failed	Check the remote parameter settings (default is recommended), check the computer network environment, firewall. If it still cannot connect, try uninstalling the native antivirus software or security guard, then reinstalling the XINJE software.
Report error message when find the device		
1	config failure(port is not in the config table)	The port is occupied, restart the service and check if other software is using the port.
2	config port failure	Restart the service and software.
3	can't find port	Restart the service and find the device
4	find device timeout	To see if the software and firmware version matches, check the connection parameters and connection between the computer and the device, and between the device and the device. If still cannot, please try stop PLC when reboot and then initialize the device.
5	bind device failure	Restart the service and software.

WUXI XINJE ELECTRIC CO., LTD.
4th Floor Building 7,Originality Industry park, Liyuan Development Zone, Wuxi City, Jiangsu Province 214072
Tel: (510) 85134136
Fax: (510) 85111290