TCP/IP communication based on Ethernet

User manual

WUXI XINJE ELECTRIC CO., LTD.

Data No. PD07 20240223EN 1.4.3

- Basic explanation

Thank you for purchasing Xinje Ethernet PLC.

This manual mainly introduces Ethernet function of PLC.

Please read this manual carefully before using and wire after understanding the content.

About software and programming instructions, please refer to related manuals.

Please hand this manual over to operation users.

- Notices for users

Only experienced operator can wire the plc. If any problem, please contact our technical department.

The listed examples are used to help users to understand, so it may not act.

Please confirm that PLC specifications and principles are suitable when connect PLC to other products. Please conform safety of PLC and machines by yourself when use the PLC. Machines may be damaged by PLC errors.

- Responsibility declaration

The manual content has been checked carefully, however, mistakes may happen.

We often check the manual and will correct the problems in subsequent version. Welcome to offer advices to us.

Excuse us that we will not inform you if manual is changed.

- Contact information

If you have any problem about products, please contact the agent or Xinje company.
Tel: 0086 510-85134136 85123803
Fax: 0086 510-85111290
Address: No. 816 Jianzhu West Road, Binhu District, Wuxi City, Jiangsu Province
Code : 214072

2018, 11, 12

CATALOG

1 Ethernet communication overview

1-1. The basic concept of Ethernet

Before the Ethernet communication, let's understand some Ethernet concepts such as IP address allocation, PC network address and settings.

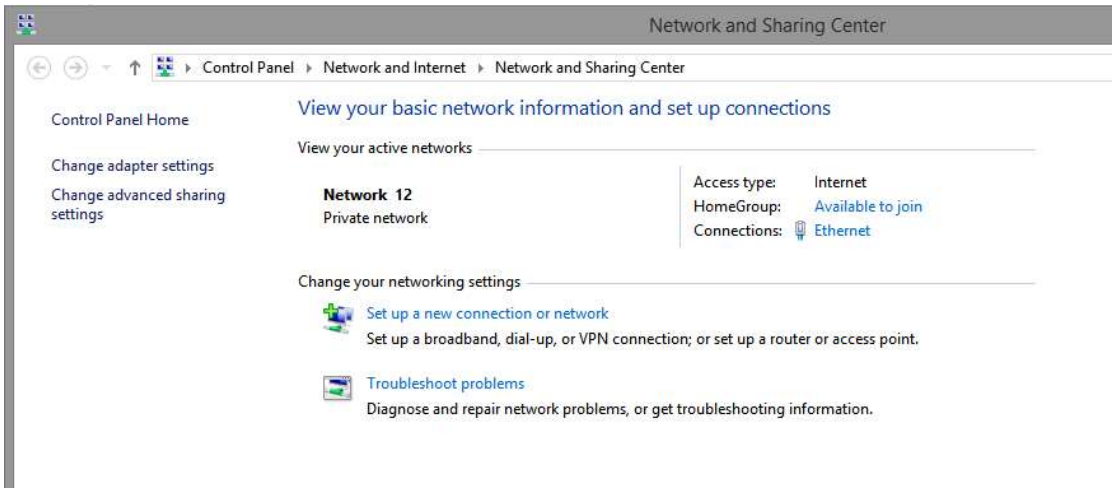
1-1-1. IP allocation

If programmable devices (such as PC) using LAN network card to connect to the factory (or the Internet), the PLC and programming device must be in the same subnet. Combination of IP address and subnet mask can be specified subnet of the equipment.
Network ID is the IP address of the first part, the top three 8-bit groups (such as IP addresses for 211.154.184.16, 211.154.184 represents network ID) decided the user's IP network. The value of the subnet mask is usually 255.255.255.0. However, because of your computer is in the local area network (LAN), subnet mask (for example, 255.255.254.0) may have different values to set the unique subnet. Subnet mask and the equipment IP address will do logic AND operation to define the boundary of the IP subnet.

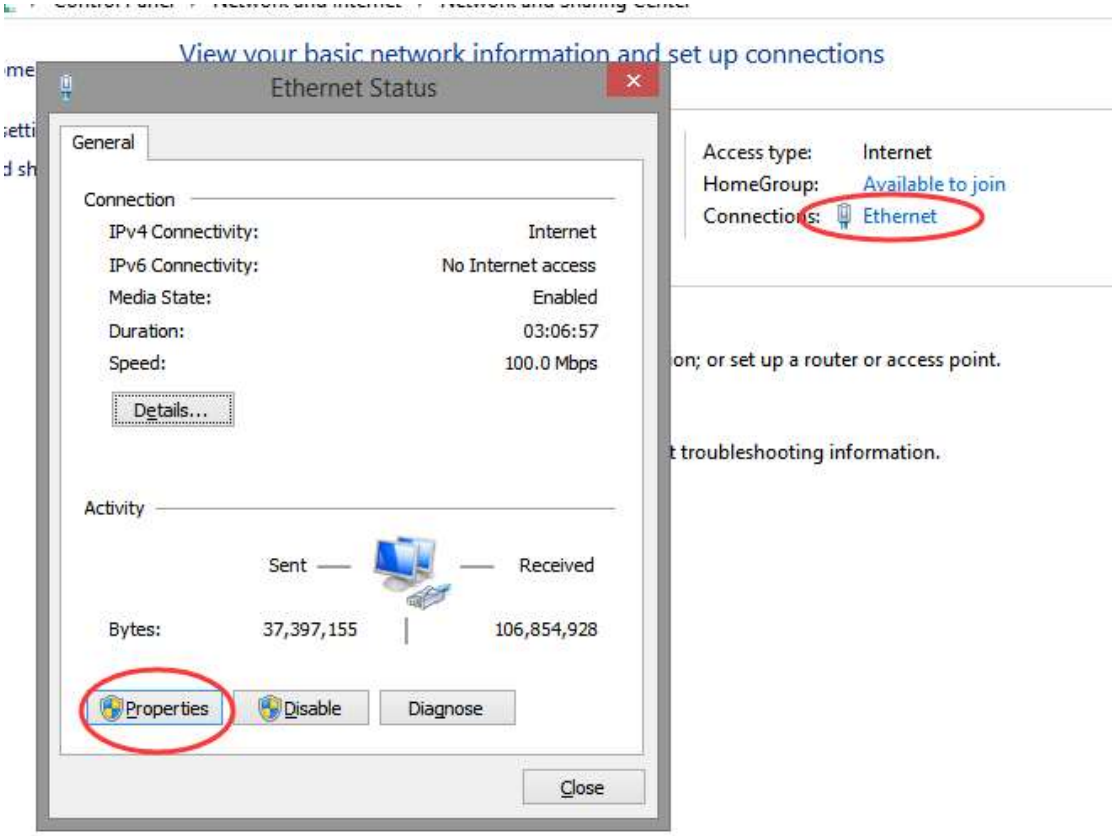
1-1-2. PC network address

Please check your programming device IP address as the following steps.

1. Open the network and sharing center:

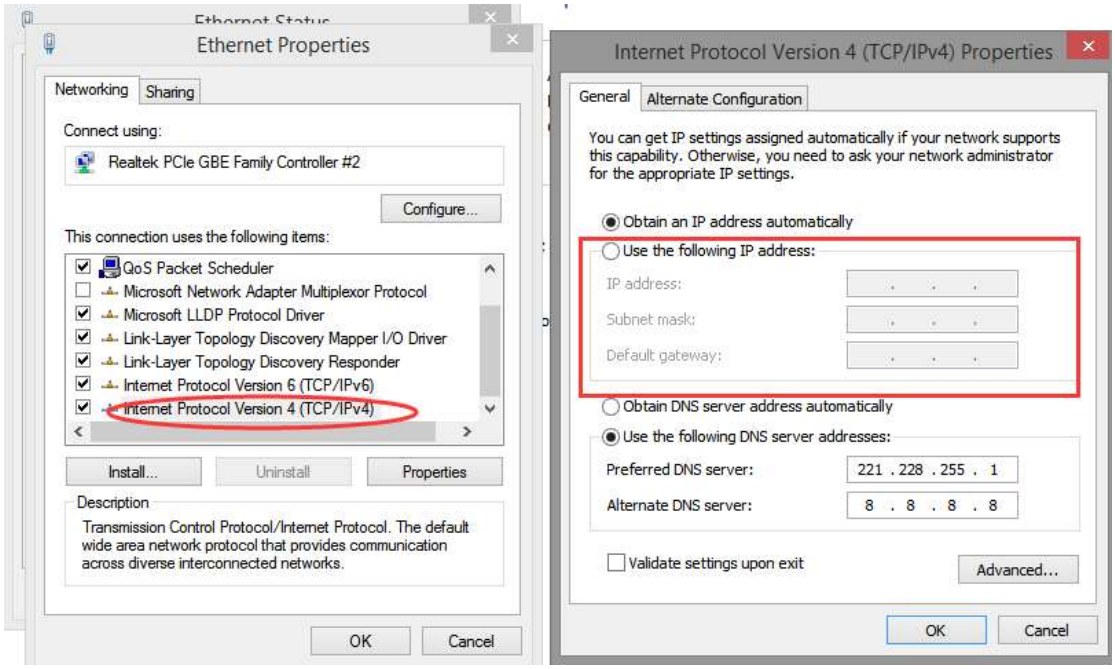


2. Click the Ethernet connections, choose properties:



3. Set the PC IP address, make it in the same subnet.

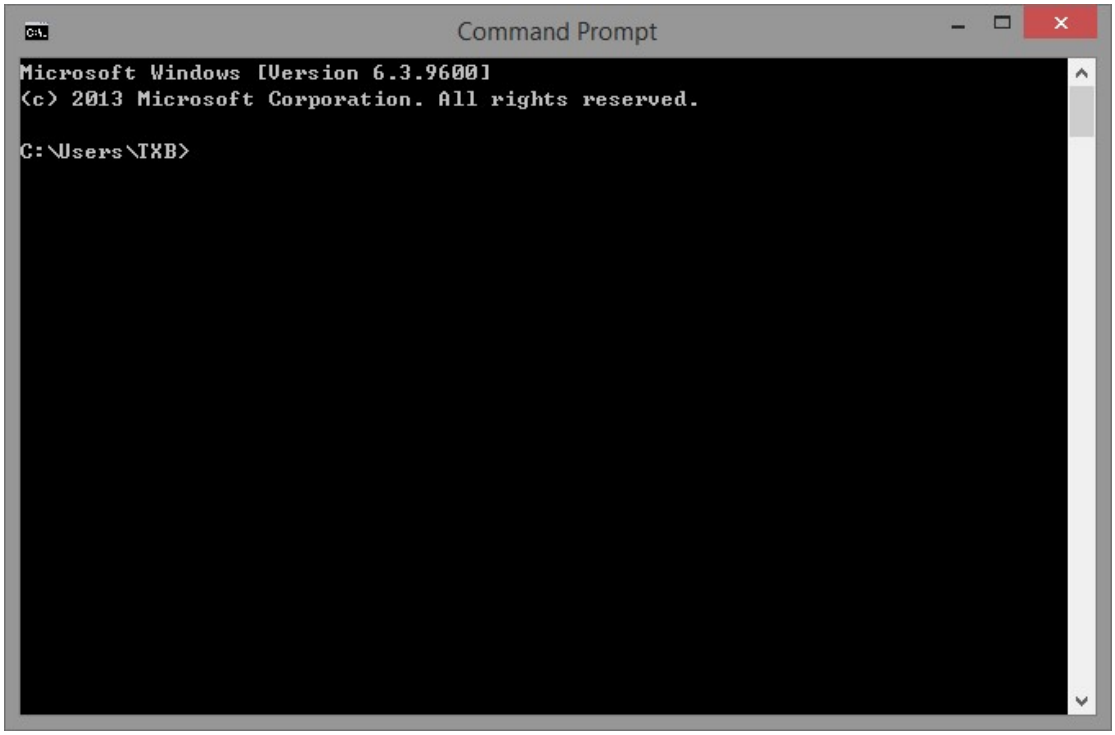
For example, the PLC IP is 192.168.2.1, the PC IP is set to 192.168.2.200, the subnet mask is 255.255.255.0. default gateway can be vacant. Then the PC can connect to the CPU.



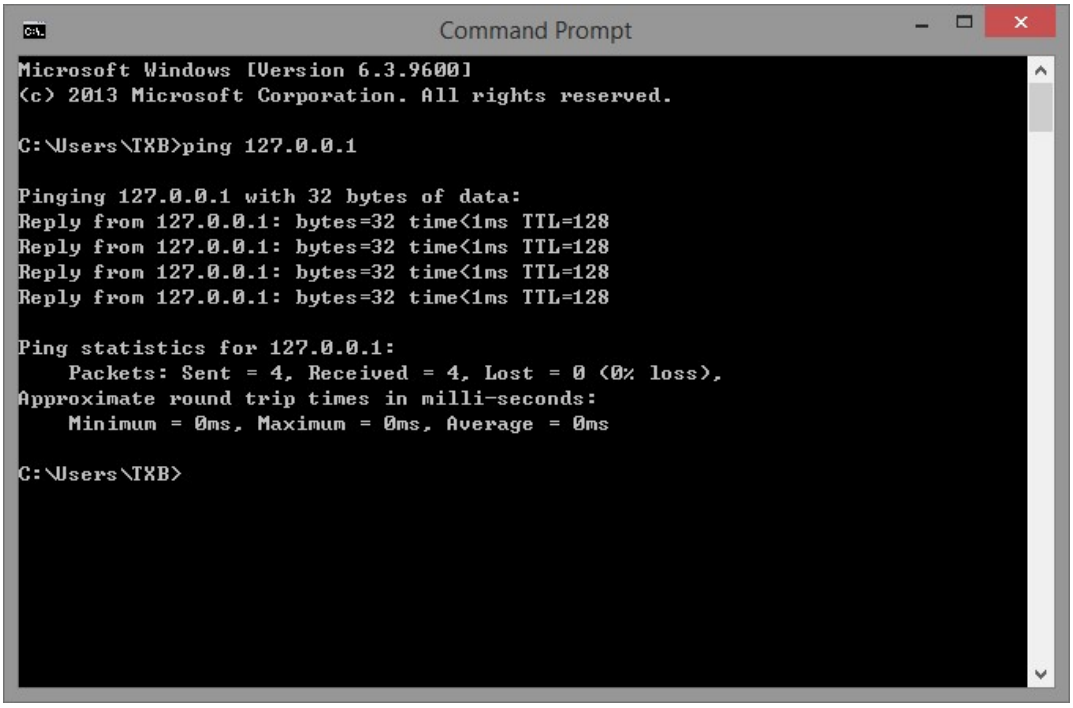
1-1-3. PING command

Through the PING command, you can check the local TCP/IP protocol, and whether it can be normal connection to other computer local area network (LAN).

1. open the command prompt

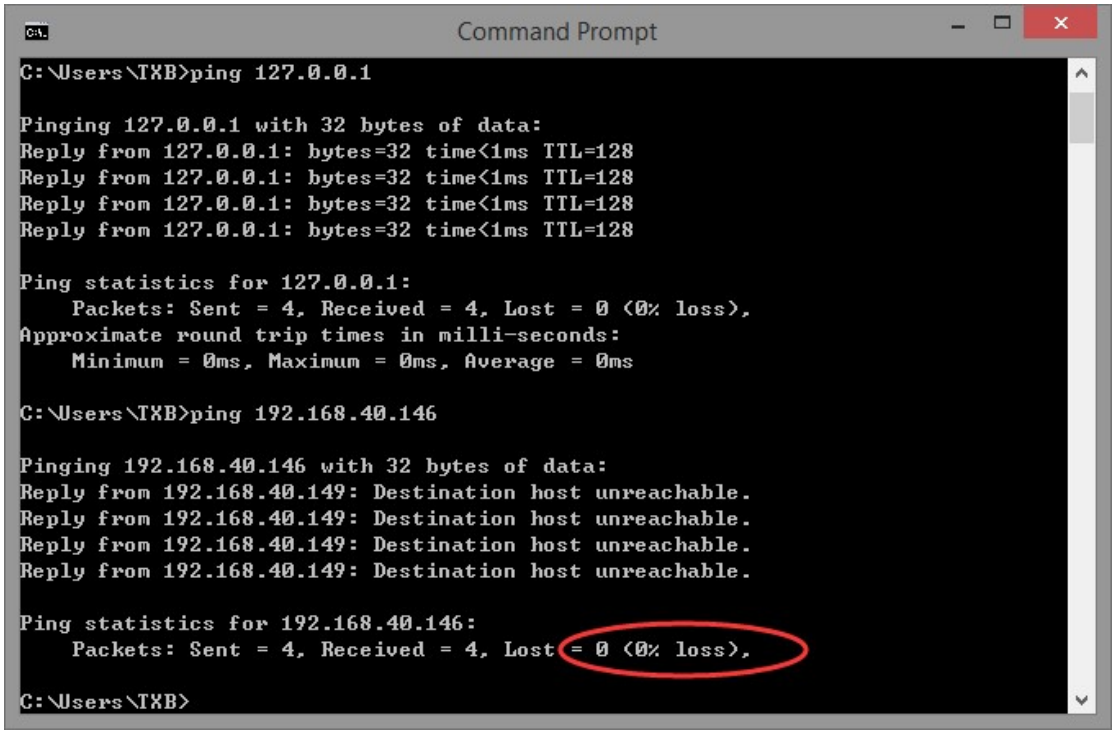


2. input “ping 127.0.0.1” to check the local TCP/IP protocol, it is normal when the receiving and sending data are same.



4. input ‘ping network device ip” command to check whether the PC can connect to other PC in the LAN.

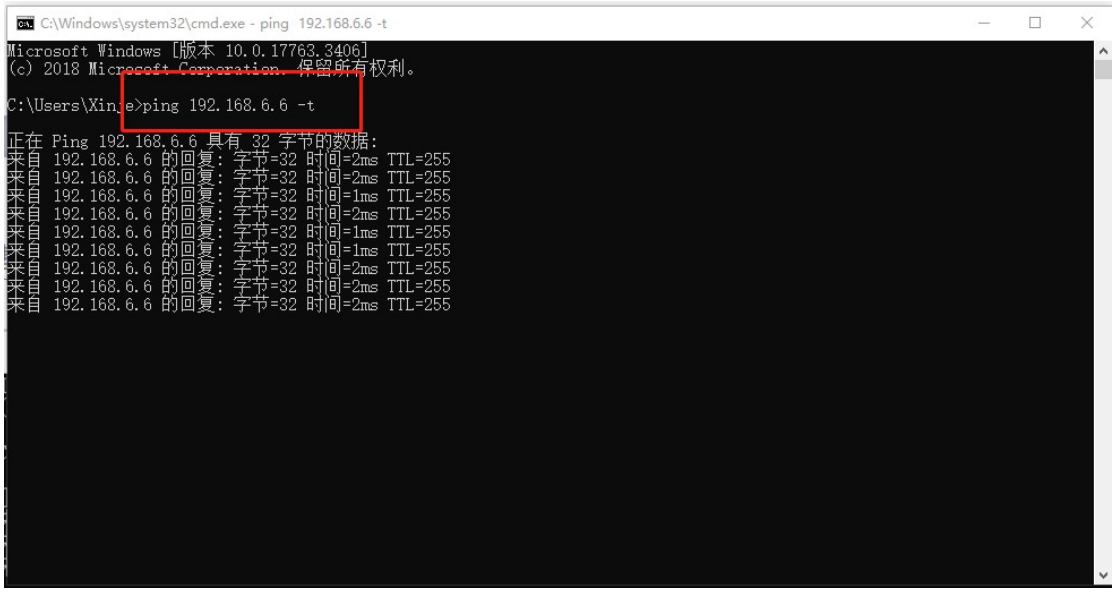
   (1) input the command “ping 192.168.40.146”, if the result shows “0% loss”, this PC can connect the PC with IP 192.168.40.146.



(2) input the command “ping 192.168.40.127”, it shows “100% loss”, which means cannot connect to the PC with IP 192.168.40.127.

Note: in the ping statistics information, only 0% loss means communication normal.

The "ping network device IP" command can only ping four times. If you want to ping continuously, you can use the "ping network device IP -t" command, as shown in the following figure:



1-2．TCP IP protocol

TCP/IP protocol is a popular Ethernet communication protocol, compared with ISO open interconnection model, adopts a more open way, it has been recognized by the U.S. department of defense, and is widely used in practical engineering. TCP/IP protocol can be used in a variety of channels and the underlying protocol (such as T1, X.25 and RS232 serial interface). Specifically, TCP/IP protocol is including TCP protocol, IP protocol, the UDP protocol, ICMP protocol and some other groups.

1-2-1．Port number

In Ethernet, the communication based on TCP or UDP must use the port number to communicate with the upper application, port range is from 0 to 65535, some port numbers have default functions, such as port 80 for browsing the web service, port 21 for FTP service, port 502 for MODBUS TCP communications, and so on.

1-2-2．UDP protocol

UDP is the user data protocol, which is a simple connectionless transmission model with min protocol . UDP protocol doesn’t have handshake mechanism, so the reliability of protocol is only equal to the underlying network. It cannot provide protection for receiving and sending message. UDP also provides checksum to ensure the integrity of data, and addresses different functions via different port numbers.

1-2-3．TCP protocol

1. The basic principle of TCP

TCP is transport control protocol, a connection-oriented, reliable transport layer protocol. Connection-oriented means a normal TCP transporting needs to build special virtual circuit between the TCP client and TCP server. To transmit data via TCP, a connection between the ends of the host must be established.

TCP provides reliable, orderly and error checking message function for application program running in the host machine which communicates through Ethernet. TCP can guarantee all the receiving and sending bytes have the same content and sequence. TCP protocol creates connections between active devices (i.e., a building connection device) and passive devices (i.e., receiving connection device). Once the connection is established, either party may initiate data transmission.

TCP protocol is a kind of "flow", which means that the message does not exist end flag, all received message is considered to be part of the data stream. For example, the client device sends three pieces of message to the server, each one is 20 bytes. Server only received a 60-byte "flow"

(assuming the server performs a receive operation after received three pieces of message).

2. The basic principle of socket

Socket (Socket) is the foundation of communication and basic operation unit to support the TCP/IP network communication. It is the abstract representations of the endpoint in the network communication process, contains five kinds of information for network communication: connection protocol, the IP address of the local host, port of the local process, the IP address of the remote host, the port of remote process.

When the application layer communicates through the transport layer, TCP will meet the problem of providing concurrent service for multiple application processes. Multiple TCP connections or more application processes may need pass through the same TCP port to transmit data. To distinguish different application processes and connection, many computer operating system provides a socket interface for the application and the TCP/IP protocol interaction. Application layer and transport layer can distinguish communication from different application processes or network connections through the socket interface, realize the data transmission of concurrent service.

3. Establish a socket connection

To establish a socket connection needs a pair of sockets at least, one runs on the client (also called the TCP client), called ClientSocket, another run on the server (also called the TCP server), called ServerSocket.

Socket connection process is divided into three steps: the server monitoring, the client request, connection confirmation.

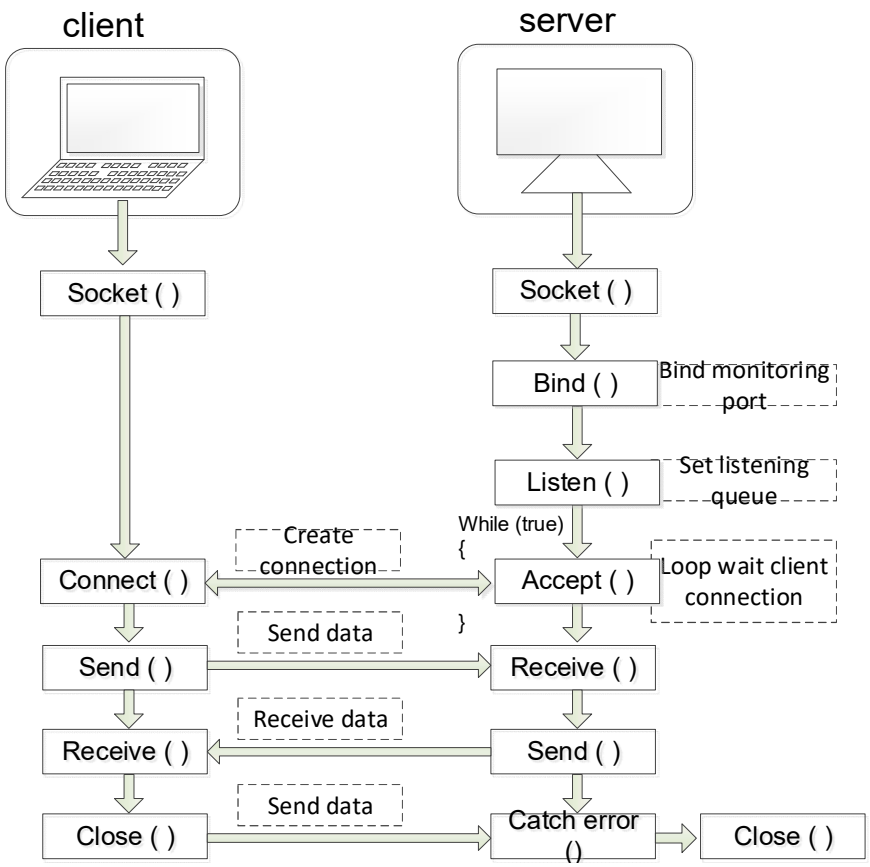
Server monitoring: the server socket does not locate specific client socket, but is in a state of waiting for the connection, and real-time monitors network state, waits for the client's connection request.

Client requests: the client socket connection requests are put forward, the target is a server socket. For this reason, the client socket must first describe the server socket, and point out the server socket address and port number, and then the server socket connection requests are put forward.

Connection confirm: when the server socket receives the client socket connection request, it will response to the request of the client socket, set up a new thread, send a description of the server socket to the client, once the client confirms the description, the two sides have established connection. The server socket is in the listening state, continues to receive other client socket connection requests.

When creating a socket connection, you can specify the transport layer protocol, the socket can support different transport layer protocol (TCP or UDP), when using TCP protocol to connect the socket, the connection is a TCP connection.

TCP communication diagram:



In above diagram, the server socket is in the listening state, client connection requests to the server, the server receives a connection request and sends the reply to confirm the information to the client, after the client received message, it sends confirmation information to the server. After completion of the allocation of resources, a TCP connection is established successfully, this process is called "three-way handshake".

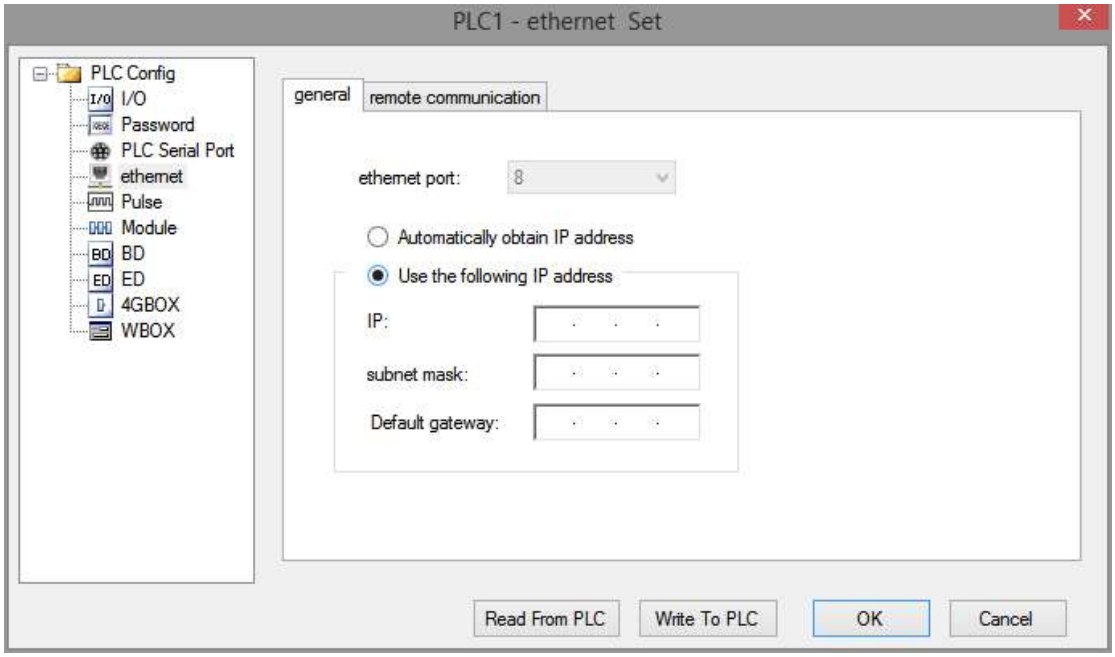
After the connection is established, the client and the server can send and receive data, after data transceiver is completed, the client or the server can request to close the connection, after the fourth "handshake", TCP connection is closed, all data transceiver interrupts.

2 Ethernet parameters

2-1. Ethernet parameters

2-1-1. IP address parameters

It needs to set the IP address in the Ethernet communication as the unique identification of each device. There are four parameters, the following charts are the IP setting interface of programming software.



Obtain the IP

Support obtain the IP address automatically, static setting function, PLC initial setting is automatical obtain.

Automatic obtain mode: when there is a DHCP server in the subnet, IP, subnet mask, default gateway are assigned by the DHCP server. Without a DHCP server, network parameters use the default values:

IP address: 192.168.6.6

Subnet mask: 255.255.255.0

The default gateway: 192.168.6.1

Static specified mode: users assigned IP, subnet mask, default gateway information. Only supports private IP address information.

IP address type	IP address range	IP device quantity
Class A private address	10.0.0.0-10.255.255.255	16777216
Class B private address	172.16.0.0-172.31.255.255	1048576
Class C private address	192.168.0.0-192.168.255.255	65535

UDP multicast address

IP address type	IP address range	IP address
Type D address	224.0.0.0~224.0.0.255	Reserved multicast address (permanent group address)
	224.0.1.0~224.0.1.255	Public multicast address
	224.0.2.0~238.255.255.255	Available multicast addresses for users (temporary group addresses)
	239.0.0.0~239.255.255.255	administratively scoped addresses

Note: It is recommended that users use IP addresses between 224.0.2.0 ~ 238.255.255.255.

2-1-2. Function specification

Item	Parameter
Number of communication channels	Ethernet series: 2 channels (same IP) XDH/XLH/XG2/XL5H series: 1 channel
Communication speed	100Mbps
Maximum space between stations	100m
Network topology	Linear, star shape

Communication type	Maximum number of network nodes
Free format TCP	32
UDP unicast	32
UDP multicast	32
Modbus TCP Client	XDH/XLH, Ethernet model: 32 XL5H: 4 XD3E: 8
Modbus TCP Server	Number of supported clients: XL5H: 4 XDH/XLH: 16 Ethernet model: 8

Note:

There are a maximum of 32 TCP protocols, including free format TCP and Modbus TCP;

Up to 32 UDP protocols, including UDP unicast and UDP multicast;

XDH and XLH series firmware versions 3.7.3 and above support UDP multicast functionality;

Ethernet PLC 3.7.2 and above firmware versions support UDP multicast functionality.

The UDP multicast function is only supported for Ethernet based PLCs with firmware versions

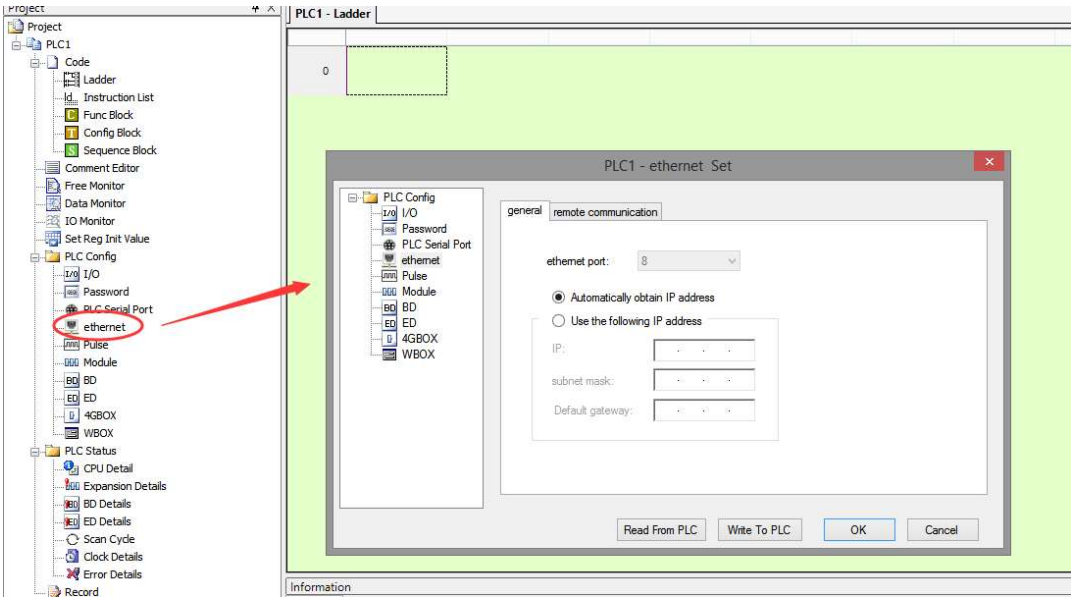
3.7.2 and above.

When using PLC as a server, Ethernet models of firmware version 3.7.2 and above support 8 clients; Ethernet models below version 3.7.2 support four clients. XDH/XLH models with firmware versions 3.7.2 and above support 16 clients, while XDH/XLH models with versions below 3.7.2 support 4 clients.

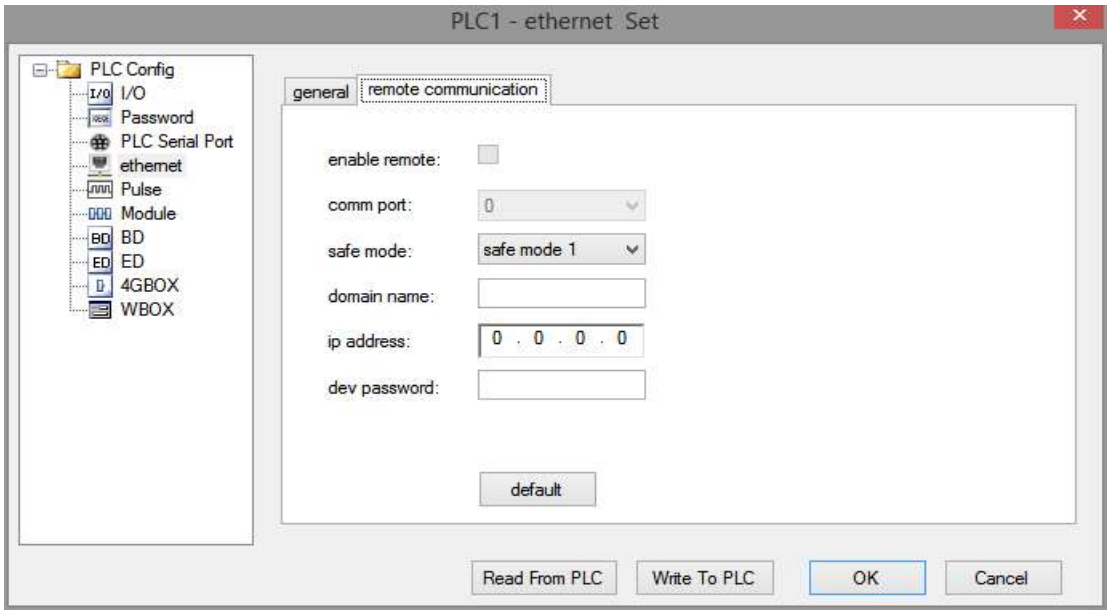
Ethernet models include: XD3E, XD5E, XDME, XL5E, XL5N, XLME.

2-2. Configure the Ethernet parameters in the software

Open the XINJE PLC programming software, click the ethernet in the left side, refer to below figure. This function is only avaliable for Ethernet model.

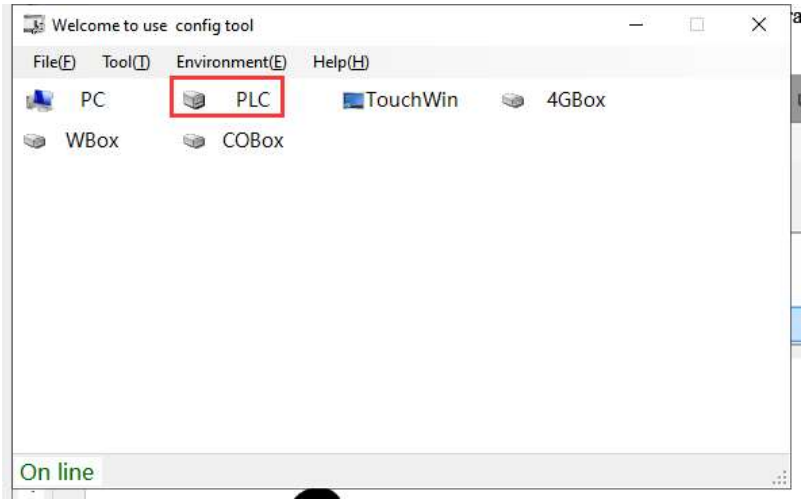


Select remote communication in the above figure, you can configure the remote parameter, it no needs to set these parameters when communicating in the local area network (LAN), after completion of all the parameters, please restart the PLC to make the settings effective.



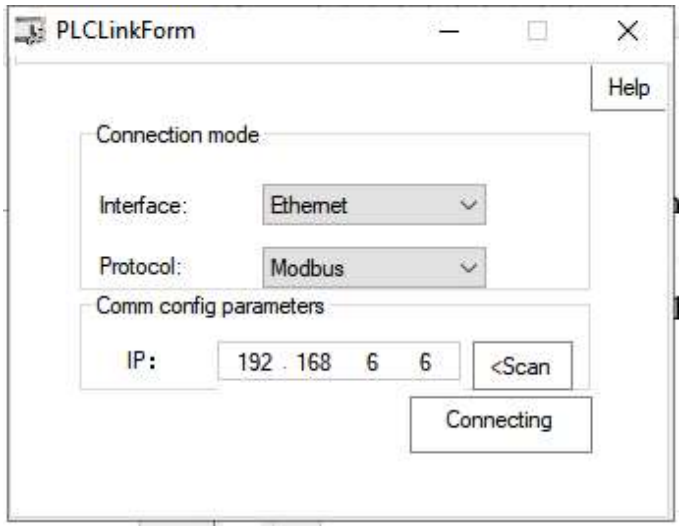
2-3. Configure ethernet parameters in XINJEConfig

When configuring in XINJEConfig for Ethernet models, use a programming cable to connect the PLC and computer. Open the XINJEConfig configuration tool (using v2.3.0.9 version as an example), and select PLC from the configuration tool.



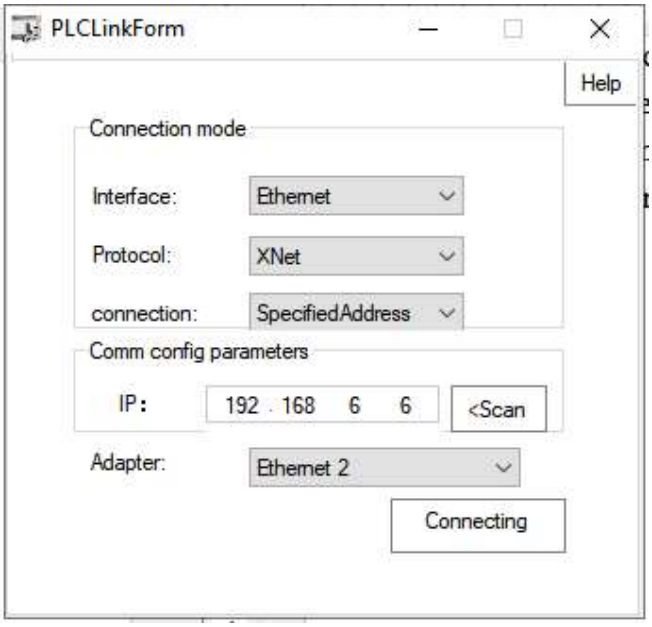
Method 1: Connect in Modbus TCP mode

In the pop-up dialog box, select Ethernet for the communication interface and Modbus for the communication protocol. At this time, connect the PLC using the Modbus TCP protocol, as shown in the following figure.

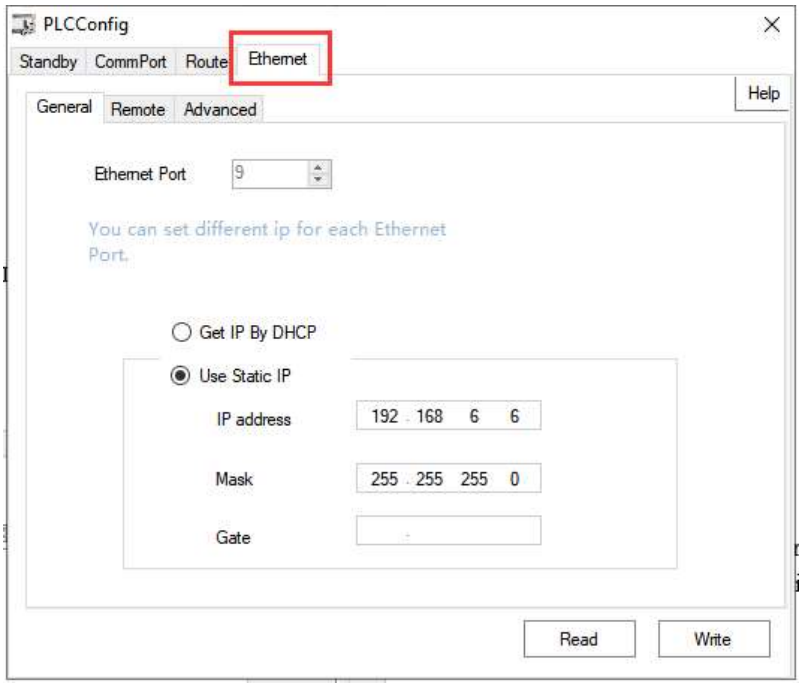


Method 2: Connect in Xnet mode

Select Ethernet for the communication interface, XNet for the communication protocol, and the specified address for the connection method. At this time, connect the PLC using the XNet protocol, set the device IP and corresponding network card, and click to connect the device, as shown in the following figure.



When configuring Ethernet parameters, select Ethernet configuration. Please refer to Section 2-1-1 for the description of configuration items, and the functions are the same as those of XDPPro configuration.


PLCConfig
Standby
CommPort
Route
Ethernet
Help
General
Remote
Advanced
Ethernet Port
9
You can set different ip for each Ethernet Port.
Get IP By DHCP
Use Static IP
IP address
192 . 168 . 6 . 6
Mask
255 . 255 . 255 . 0
Gate
Read
Write

3 Wiring and communication protocol

3-1. Wiring mode

The physical interface of Ethernet model is RJ45, the wiring cable is recommended to use UTP and STP cable, single length cannot be more than 100 meters. Switch type is recommended to use MB/GB adaptive switch.

3-2. MODBUS TCP protocol

3-2-1. MODBUS TCP overview

MODBUS TCP combined standard TCP/IP, Ethernet physical network and MODBUS as the data representation method of data application protocol. MODBUS TCP communication message is encapsulated in Ethernet TCP/IP packets, MODBUS protocol one frame maximum length is 256 bytes.
MODBUS TCP/IP has two type of devices: Modbus TCP/IP clinet and server.

MODBUS client:
Client (TCP Client) launched a connection request to the Server (TCP Server), the connection is established successfully, it only allows the Client to initiate communication request.
When the Ethernet model is the MODBUS TCP client, it establishes a TCP connection through S_OPEN instruction, initiates MODBUS request by M_TCP instruction.

MODBUS server:
The server listened to port 502, waited for the client connection request, after the connection was established successfully, it responsed to the data communication request in accordance with the Modbus TCP protocol specification.
Ethernet devices defaulted open this service when power on, the maximum response is shown in the table below.
Note: The number of clients supported by PLC is as follows:

Firmware version	PLC model	Number of supported clients
Below 3.7.2	XD5E/XL5E/XDME/XLME/XDH/XLH	4
3.7.2 and up	XL5H	4
	XD3E/XD5E/XL5E/XL5N/XDME/XLME	8

Firmware version	PLC model	Number of supported clients
	XDH/XLH	16

3-2-2. MODBUS address

When the programmable controller serves as a Modbus server, the internal software component numbers and corresponding Modbus address numbers are as follows.

(1) Modbus address and internal software component comparison table for XD3E series PLC

Note: For the calculation of Modbus addresses for X and Y, please refer to the bottom of the table.

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
Coil, bit object	M	M0~M7999	8000	0~1F3F	0~7999
	X	X0~X77 (main body)	64	5000~503F	20480~20543
		X10000~X10077 (#1 module)	64	5100~513F	20736~20799
		X10100~X10177 (#2 module)	64	5140~517F	20800~20863
		X10200~X10277 (#3 module)	64	5180~51BF	20864~20927
		X10300~X10377 (#4 module)	64	51C0~51FF	20928~20991
		X10400~X10477 (#5 module)	64	5200~523F	20992~21055
		X10500~X10577 (#6 module)	64	5240~527F	21056~21119
		X10600~X10677 (#7 module)	64	5280~52BF	21120~21183
		X10700~X10777 (#8 module)	64	52C0~52FF	21184~21247
		X11000~X11077 (#9 module)	64	5300~533F	21248~21311
		X11100~X11177 (#10 module)	64	5340~537F	21312~21375
		X20000~X20077 (#1 BD)	64	58D0~590F	22736~22799
		X20100~X20177 (#2 BD)	64	5910~594F	22800~22863
		X30000~X30077 (#1 ED)	64	5BF0~5C2F	23536~23599
	Y	Y0~Y77 (main body)	64	6000~603F	24576~24639
		Y10000~Y10077 (#1 module)	64	6100~613F	24832~24895

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		Y10100~Y10177 (#2 module)	64	6140~617F	24896~24959
		Y10200~Y10277 (#3 module)	64	6180~61BF	24960~25023
		Y10300~Y10377 (#4 module)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (#5 module)	64	6200~623F	25088~25151
		Y10500~Y10577 (#6 module)	64	6240~627F	25152~25215
		Y10600~Y10677 (#7 module)	64	6280~62BF	25216~25279
		Y10700~Y10777 (#8 module)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (#9 module)	64	6300~633F	25344~25407
		Y11100~Y11177 (#10 module)	64	6340~637F	25408~25471
		Y20000~Y20077 (#1 BD)	64	68D0~690F	26832~26895
		Y20100~Y20177 (#2 BD)	64	6910~694F	26896~26956
		Y30000~Y30077 (#1 ED)	64	6BF0~6C2F	27632~27695
	S	S0~S1023	1024	7000~73FF	28672~29695
	SM	SM0~SM2047	2048	9000~97FF	36864~38911
	T	T0~T575	576	A000~A23F	40960~41535
	C	C0~C575	576	B000~B23F	45056~45631
	ET	ET0~ET31	32	C000~C01F	49152~49183
	SEM	SEM0~SEM31	32	C080~C09F	49280~49311
	HM※1	HM0~HM959	960	C100~C4BF	49408~50367
	HS※1	HS0~HS127	128	D900~D97F	55552~55679
	HT※1	HT0~HT95	96	E100~E15F	57600~57695
	HC※1	HC0~HC95	96	E500~E55F	58624~58719
	HSC※1	HSC0~HSC31	32	E900~E91F	59648~59679
Register, word object	D	D0~D7999	8000	0~1F3F	0~7999
	ID	ID0~ID99 (main body)	100	5000~5063	20480~20579
		ID10000~ID10099 (#1 module)	100	5100~5163	20736~20835
		ID10100~ID10199 (#2 module)	100	5164~51C7	20836~20935
		ID10200~ID10299 (#3 module)	100	51C8~522B	20936~21035

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		ID10300~ID10399 (#4 module)	100	522C~528F	21036~21135
		ID10400~ID10499 (#5 module)	100	5290~52F3	21136~21235
		ID10500~ID10599 (#6 module)	100	52F4~5357	21236~21335
		ID10600~ID10699 (#7 module)	100	5358~53BB	21336~21435
		ID10700~ID10799 (#8 module)	100	53BC~541F	21436~21535
		ID10800~ID10899 (#9 module)	100	5420~5483	21536~21635
		ID10900~ID10999 (#10 module)	100	5484~54E7	21636~21735
		ID20000~ID20099 (#1 BD)	100	58D0~5933	22736~22835
		ID20100~ID20199 (#2 BD)	100	5934~5997	22836~22935
		ID30000~ID30099 (#1 ED)	100	5BF0~5C53	23536~23635
Register, word object	QD	QD0~QD99 (main body)	100	6000~6063	24576~24675
		QD10000~QD10099 (#1 module)	100	6100~6163	24832~24931
		QD10100~QD10199 (#2 module)	100	6164~61C7	24932~25031
		QD10200~QD10299 (#3 module)	100	61C8~622B	25032~25131
		QD10300~QD10399 (#4 module)	100	622C~628F	25132~25231
		QD10400~QD10499 (#5 module)	100	6290~62F3	25232~25331
		QD10500~QD10599 (#6 module)	100	62F4~6357	25332~25431
		QD10600~QD10699 (#7 module)	100	6358~63BB	25432~25531
		QD10700~QD10799 (#8 module)	100	63BC~641F	25532~25631
		QD10800~QD10899 (#9 module)	100	6420~6483	25632~25731
		QD10900~QD10999 (#10 module)	100	6484~64E7	25732~25831
		QD20000~QD20099 (#1 BD)	100	68D0~6933	26832~26931

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		QD20100~QD20199 (#2 BD)	100	6934~6997	26932~27031
		QD30000~QD30099 (#1 ED)	100	6BF0~6C53	27632~27731
	SD	SD0~SD2047	2048	7000~77FF	28672~30719
	TD	TD0~TD575	576	8000~823F	32768~33343
	CD	CD0~CD575	576	9000~923F	36864~37439
	ETD	ETD0~ETD31	32	A000~A01F	40960~40991
	HD※1	HD0~HD999	1000	A080~A467	41088~42087
	HSD※1	HSD0~HSD499	500	B880~BA73	47232~47731
	HTD※1	HTD0~HTD95	96	BC80~BCDF	48256~48351
	HCD※1	HCD0~HCD95	96	C080~C0DF	49280~49375
	HSCD※1	HSCD0~HSCD31	32	C480~C49F	50304~50335
	FD※2	FD0~FD5119	5120	C4C0~D8BF	50368~55487
	SFD※2	SFD0~SFD1999	2000	E4C0~EC8F	58560~60559
	FS※2	FS0~FS47	48	F4C0~F4EF	62656~62703

(2) XD5E, XDME, XL5E, XL5N, XL5H, XLME series Modbus address and internal software component comparison table.

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
Coil, bit object	M	M0~M20479	20480	0~4FFF	0~20479
	X	X0~X77 (main body)	64	5000~503F	20480~20543
		X10000~X10077 (#1 module)	64	5100~513F	20736~20799
		X10100~X10177 (#2 module)	64	5140~517F	20800~20863
		X10200~X10277 (#3 module)	64	5180~51BF	20864~20927
		X10300~X10377 (#4 module)	64	51C0~51FF	20928~20991
		X10400~X10477 (#5 module)	64	5200~523F	20992~21055
		X10500~X10577 (#6 module)	64	5240~527F	21056~21119
		X10600~X10677 (#7 module)	64	5280~52BF	21120~21183
		X10700~X10777 (#8	64	52C0~52FF	21184~21247

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		module)			
		X11000~X11077 (#9 module)	64	5300~533F	21248~21311
		X11100~X11177 (#10 module)	64	5340~537F	21312~21375
		X11200~X11277 (#11 module)	64	5380~53BF	21376~21439
		X11300~X11377 (#12 module)	64	53C0~53FF	21440~21503
		X11400~X11477 (#13 module)	64	5400~543F	21504~21567
		X11500~X11577 (#14 module)	64	5440~547F	21568~21631
		X11600~X11677 (#15 module)	64	5480~54BF	21632~21695
		X11700~X11777 (#16 module)	64	54C0~54FF	21696~21759
		X20000~X20077 (#1 BD)	64	58D0~590F	22736~22799
		X20100~X20177 (#2 BD)	64	5910~594F	22800~22863
		X30000~X30077 (#1 ED)	64	5BF0~5C2F	23536~23599
	Y	Y0~Y77 (main body)	64	6000~603F	24576~24639
		Y10000~Y10077 (#1 module)	64	6100~613F	24832~24895
		Y10100~Y10177 (#2 module)	64	6140~617F	24896~24959
		Y10200~Y10277 (#3 module)	64	6180~61BF	24960~25023
		Y10300~Y10377 (#4 module)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (#5 module)	64	6200~623F	25088~25151
		Y10500~Y10577 (#6 module)	64	6240~627F	25152~25215
		Y10600~Y10677 (#7 module)	64	6280~62BF	25216~25279
		Y10700~Y10777 (#8 module)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (#9 module)	64	6300~633F	25344~25407
		Y11100~Y11177 (#10	64	6340~637F	25408~25471

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		module)			
		Y11200~Y11277 (#11 module)	64	6380~63BF	25472~25535
		Y11300~Y11377 (#12 module)	64	63C0~63FF	25536~25599
		Y11400~Y11477 (#13 module)	64	6400~643F	25600~25663
		Y11500~Y11577 (#14 module)	64	6440~647F	25664~25727
		Y11600~Y11677 (#15 module)	64	6480~64BF	25728~25791
		Y11700~Y11777 (#16 module)	64	64C0~64FF	25792~25855
		Y20000~Y20077 (#1 BD)	64	68D0~690F	26832~26895
		Y20100~Y20177 (#2 BD)	64	6910~694F	26896~26956
		Y30000~Y30077 (#1 ED)	64	6BF0~6C2F	27632~27695
	S	S0~S7999	8000	7000~8F3F	28672~36671
	SM	SM0~SM4095	4096	9000~9FFF	36864~40959
	T	T0~T4095	4096	A000~AFFF	40960~45055
	C	C0~C4095	4096	B000~BFFF	45056~49151
	ET	ET0~ET39	40	C000~C027	49152~49191
	SEM	SEM0~SEM127	128	C080~C0FF	49280~49407
	HM※1	HM0~HM6143	6144	C100~D8FF	49408~55551
	HS※1	HS0~HS999	1000	D900~DCEF	55552~56551
	HT※1	HT0~HT1023	1024	E100~E4FF	57600~58623
	HC※1	HC0~HC1023	1024	E500~E8FF	58624~59647
	HSC※1	HSC0~HSC39	40	E900~E927	59648~59687
Register, word object	D	D0~D20479	20480	0~4FFF	0~20479
	ID	ID0~ID99 (main body)	100	5000~5063	20480~20579
		ID10000~ID10099 (#1 module)	100	5100~5163	20736~20835
		ID10100~ID10199 (#2 module)	100	5164~51C7	20836~20935
		ID10200~ID10299 (#3 module)	100	51C8~522B	20936~21035
		ID10300~ID10399 (#4 module)	100	522C~528F	21036~21135
		ID10400~ID10499 (#5 module)	100	5290~52F3	21136~21235
		ID10500~ID10599 (#6	100	52F4~5357	21236~21335

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		module)			
		ID10600~ID10699 (#7 module)	100	5358~53BB	21336~21435
		ID10700~ID10799 (#8 module)	100	53BC~541F	21436~21535
		ID10800~ID10899 (#9 module)	100	5420~5483	21536~21635
		ID10900~ID10999 (#10 module)	100	5484~54E7	21636~21735
		ID11000~ID11099 (#11 module)	100	54E8~554B	21736~21835
		ID11100~ID11199 (#12 module)	100	554C~55AF	21836~21935
		ID11200~ID11299 (#13 module)	100	55B0~5613	21936~22035
		ID11300~ID11399 (#14 module)	100	5614~5677	22036~22135
		ID11400~ID11499 (#15 module)	100	5678~56DB	22136~22235
		ID11500~ID11599 (#16 module)	100	56DC~573F	22236~22335
		ID20000~ID20099 (#1 BD)	100	58D0~5933	22736~22835
		ID20100~ID20199 (#2 BD)	100	5934~5997	22836~22935
		ID30000~ID30099 (#1 ED)	100	5BF0~5C53	23536~23635
	QD	QD0~QD99 (main body)	100	6000~6063	24576~24675
		QD10000~QD10099 (#1 module)	100	6100~6163	24832~24931
		QD10100~QD10199 (#2 module)	100	6164~61C7	24932~25031
		QD10200~QD10299 (#3 module)	100	61C8~622B	25032~25131
		QD10300~QD10399 (#4 module)	100	622C~628F	25132~25231
		QD10400~QD10499 (#5 module)	100	6290~62F3	25232~25331
		QD10500~QD10599 (#6 module)	100	62F4~6357	25332~25431
		QD10600~QD10699 (#7 module)	100	6358~63BB	25432~25531
		QD10700~QD10799 (#8	100	63BC~641F	25532~25631

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		module)			
		QD10800~QD10899 (#9 module)	100	6420~6483	25632~25731
		QD10900~QD10999 (#10 module)	100	6484~64E7	25732~25831
		QD11000~QD11099 (#11 module)	100	64E8~654B	25832~25931
		QD11100~QD11199 (#12 module)	100	654C~65AF	25932~26031
		QD11200~QD11299 (#13 module)	100	65B0~6613	26032~26131
		QD11300~QD11399 (#14 module)	100	6614~6677	26132~26231
		QD11400~QD11499 (#15 module)	100	6678~66DB	26232~26331
		QD11500~QD11599 (#16 module)	100	66DC~673F	26332~26431
		QD20000~QD20099 (#1 BD)	100	68D0~6933	26832~26931
	QD	QD20100~QD20199 (#2 BD)	100	6934~6997	26932~27031
		QD30000~QD30099 (#1 ED)	100	6BF0~6C53	27632~27731
	SD	SD0~SD4095	4096	7000~7FFF	28672~32767
	TD	TD0~TD4095	4096	8000~8FFF	32768~36863
	CD	CD0~CD4095	4096	9000~9FFF	36864~40959
	ETD	ETD0~ETD39	40	A000~A027	40960~40999
	HD※1	HD0~HD6143	6144	A080~B87F	41088~47231
	HSD※1	HSD0~HSD1023	1024	B880~BC7F	47232~48255
	HTD※1	HTD0~HTD1023	1024	BC80~C07F	48256~49279
	HCD※1	HCD0~HCD1023	1024	C080~C47F	49280~50303
	HSCD※1	HSCD0~HSCD39	40	C480~C4A7	50304~50343
	FD※2	FD0~FD8191	8192	C4C0~E4BF	50368~58559
	SFD※2	SFD0~SFD4095	4096	E4C0~F4BF	58560~62655
	FS※2	FS0~FS47	48	F4C0~F4EF	62656~62703

(3) XDH and XLH Series Modbus Addresses and Internal Software Components.

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
Coil, bit object	M	M0~M20479	20480	0~4FFF	0~20479
	X	X0~X77 (main body)	64	5000~503F	20480~20543
		X10000~X10077 (#1 module)	64	5100~513F	20736~20799
		X10100~X10177 (#2 module)	64	5140~517F	20800~20863
		X10200~X10277 (#3 module)	64	5180~51BF	20864~20927
		X10300~X10377 (#4 module)	64	51C0~51FF	20928~20991
		X10400~X10477 (#5 module)	64	5200~523F	20992~21055
		X10500~X10577 (#6 module)	64	5240~527F	21056~21119
		X10600~X10677 (#7 module)	64	5280~52BF	21120~21183
		X10700~X10777 (#8 module)	64	52C0~52FF	21184~21247
		X11000~X11077 (#9 module)	64	5300~533F	21248~21311
		X11100~X11177 (#10 module)	64	5340~537F	21312~21375
		X11200~X11277 (#11 module)	64	5380~53BF	21376~21439
		X11300~X11377 (#12 module)	64	53C0~53FF	21440~21503
		X11400~X11477 (#13 module)	64	5400~543F	21504~21567
		X11500~X11577 (#14 module)	64	5440~547F	21568~21631
		X11600~X11677 (#15 module)	64	5480~54BF	21632~21695
		X11700~X11777 (#16 module)	64	54C0~54FF	21696~21759
		X20000~X20077 (#1 BD)	64	58D0~590F	22736~22799
		X20100~X20177 (#2 BD)	64	5910~594F	22800~22863
		X30000~X30077 (#1 ED)	64	5BF0~5C2F	23536~23599
	Y	Y0~Y77 (main body)	64	6000~603F	24576~24639
		Y10000~Y10077 (#1	64	6100~613F	24832~24895

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		module)			
		Y10100~Y10177 (#2 module)	64	6140~617F	24896~24959
		Y10200~Y10277 (#3 module)	64	6180~61BF	24960~25023
		Y10300~Y10377 (#4 module)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (#5 module)	64	6200~623F	25088~25151
		Y10500~Y10577 (#6 module)	64	6240~627F	25152~25215
		Y10600~Y10677 (#7 module)	64	6280~62BF	25216~25279
		Y10700~Y10777 (#8 module)	64	62C0~62FF	25280~25343
	Y	Y11000~Y11077 (#9 module)	64	6300~633F	25344~25407
		Y11100~Y11177 (#10 module)	64	6340~637F	25408~25471
		Y11200~Y11277 (#11 module)	64	6380~63BF	25472~25535
		Y11300~Y11377 (#12 module)	64	63C0~63FF	25536~25599
		Y11400~Y11477 (#13 module)	64	6400~643F	25600~25663
		Y11500~Y11577 (#14 module)	64	6440~647F	25664~25727
		Y11600~Y11677 (#15 module)	64	6480~64BF	25728~25791
		Y11700~Y11777 (#16 module)	64	64C0~64FF	25792~25855
		Y20000~Y20077 (#1 BD)	64	68D0~690F	26832~26895
		Y20100~Y20177 (#2 BD)	64	6910~694F	26896~26956
		Y30000~Y30077 (#1 ED)	64	6BF0~6C2F	27632~27695
	S	S0~S7999	8000	7000~8F3F	28672~36671
	SM	SM0~SM4095	4096	9000~9FFF	36864~40959
	T	T0~T4095	4096	A000~AFFF	40960~45055
	C	C0~C4095	4096	B000~BFFF	45056~49151
	ET	ET0~ET39	40	C000~C027	49152~49191
	SEM	SEM0~SEM127	128	C080~C0FF	49280~49407

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
	HM※1	HM0~HM6143	6144	C100~D8FF	49408~55551
	HS※1	HS0~HS999	1000	D900~DCEF	55552~56551
	HT※1	HT0~HT1023	1024	E100~E4FF	57600~58623
	HC※1	HC0~HC1023	1024	E500~E8FF	58624~59647
	HSC※1	HSC0~HSC39	40	E900~E927	59648~59687
Register, word object	D	D0~D20479	20480	0~4FFF	0~20479
	ID	ID0~ID99 (main body)	100	5000~5063	20480~20579
		ID10000~ID10099 (#1 module)	100	5100~5163	20736~20835
		ID10100~ID10199 (#2 module)	100	5164~51C7	20836~20935
		ID10200~ID10299 (#3 module)	100	51C8~522B	20936~21035
		ID10300~ID10399 (#4 module)	100	522C~528F	21036~21135
		ID10400~ID10499 (#5 module)	100	5290~52F3	21136~21235
		ID10500~ID10599 (#6 module)	100	52F4~5357	21236~21335
		ID10600~ID10699 (#7 module)	100	5358~53BB	21336~21435
		ID10700~ID10799 (#8 module)	100	53BC~541F	21436~21535
		ID10800~ID10899 (#9 module)	100	5420~5483	21536~21635
		ID10900~ID10999 (#10 module)	100	5484~54E7	21636~21735
		ID11000~ID11099 (#11 module)	100	54E8~554B	21736~21835
		ID11100~ID11199 (#12 module)	100	554C~55AF	21836~21935
		ID11200~ID11299 (#13 module)	100	55B0~5613	21936~22035
		ID11300~ID11399 (#14 module)	100	5614~5677	22036~22135
		ID11400~ID11499 (#15 module)	100	5678~56DB	22136~22235
		ID11500~ID11599 (#16 module)	100	56DC~573F	22236~22335
		ID20000~ID20099 (#1 BD)	100	58D0~5933	22736~22835

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		ID20100~ID20199 (#2 BD)	100	5934~5997	22836~22935
		ID30000~ID30099 (#1 ED)	100	5BF0~5C53	23536~23635
	QD	QD0~QD99 (main body)	100	6000~6063	24576~24675
		QD10000~QD10099 (#1 module)	100	6100~6163	24832~24931
		QD10100~QD10199 (#2 module)	100	6164~61C7	24932~25031
		QD10200~QD10299 (#3 module)	100	61C8~622B	25032~25131
		QD10300~QD10399 (#4 module)	100	622C~628F	25132~25231
		QD10400~QD10499 (#5 module)	100	6290~62F3	25232~25331
		QD10500~QD10599 (#6 module)	100	62F4~6357	25332~25431
		QD10600~QD10699 (#7 module)	100	6358~63BB	25432~25531
		QD10700~QD10799 (#8 module)	100	63BC~641F	25532~25631
	QD	QD10800~QD10899 (#9 module)	100	6420~6483	25632~25731
		QD10900~QD10999 (#10 module)	100	6484~64E7	25732~25831
		QD11000~QD11099 (#11 module)	100	64E8~654B	25832~25931
		QD11100~QD11199 (#12 module)	100	654C~65AF	25932~26031
		QD11200~QD11299 (#13 module)	100	65B0~6613	26032~26131
		QD11300~QD11399 (#14 module)	100	6614~6677	26132~26231
		QD11400~QD11499 (#15 module)	100	6678~66DB	26232~26331
		QD11500~QD11599 (#16 module)	100	66DC~673F	26332~26431
		QD20000~QD20099 (#1 BD)	100	68D0~6933	26832~26931
		QD20100~QD20199 (#2 BD)	100	6934~6997	26932~27031
		QD30000~QD30099 (#1	100	6BF0~6C53	27632~27731

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		ED)			
	SD	SD0~SD4095	4096	7000~7FFF	28672~32767
	TD	TD0~TD4095	4096	8000~8FFF	32768~36863
	CD	CD0~CD4095	4096	9000~9FFF	36864~40959
	ETD	ETD0~ETD39	40	A000~A027	40960~40999
	HD※1	HD0~HD6143	6144	A080~B87F	41088~47231
	HSD※1	HSD0~HSD1023	1024	B880~BC7F	47232~48255
	HTD※1	HTD0~HTD1023	1024	BC80~C07F	48256~49279
	HCD※1	HCD0~HCD1023	1024	C080~C47F	49280~50303
	HSCD※1	HSCD0~HSCD39	40	C480~C4A7	50304~50343
	FD※2	FD0~FD8191	8192	C4C0~E4BF	50368~58559
	SFD※2	SFD0~SFD4095	4096	E4C0~FC2F	58560~64559
	FS※2	FS0~FS47	256	F4C0~F4EF	62656~62911

(4) XG Series Modbus Address and Internal Software Components:

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
Coil, bit object	M	M0~M20479	20480	0~4FFF	0~20479
	X	X0~X77 (main body)	64	5000~503F	20480~20543
		X10000~X10077 (#1 module)	64	5100~513F	20736~20799
		X10100~X10177 (#2 module)	64	5140~517F	20800~20863
		X10200~X10277 (#3 module)	64	5180~51BF	20864~20927
		X10300~X10377 (#4 module)	64	51C0~51FF	20928~20991
		X10400~X10477 (#5 module)	64	5200~523F	20992~21055
		X10500~X10577 (#6 module)	64	5240~527F	21056~21119
		X10600~X10677 (#7 module)	64	5280~52BF	21120~21183
		X10700~X10777 (#8 module)	64	52C0~52FF	21184~21247
		X11000~X11077 (#9 module)	64	5300~533F	21248~21311
		X11100~X11177 (#10 module)	64	5340~537F	21312~21375

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		X11200~X11277 (#11 module)	64	5380~53BF	21376~21439
		X11300~X11377 (#12 module)	64	53C0~53FF	21440~21503
		X11400~X11477 (#13 module)	64	5400~543F	21504~21567
		X11500~X11577 (#14 module)	64	5440~547F	21568~21631
		X11600~X11677 (#15 module)	64	5480~54BF	21632~21695
		X11700~X11777 (#16 module)	64	54C0~54FF	21696~21759
	Y	Y0~Y77 (main body)	64	6000~603F	24576~24639
		Y10000~Y10077 (#1 module)	64	6100~613F	24832~24895
		Y10100~Y10177 (#2 module)	64	6140~617F	24896~24959
		Y10200~Y10277 (#3 module)	64	6180~61BF	24960~25023
		Y10300~Y10377 (#4 module)	64	61C0~61FF	25024~25087
		Y10400~Y10477 (#5 module)	64	6200~623F	25088~25151
		Y10500~Y10577 (#6 module)	64	6240~627F	25152~25215
		Y10600~Y10677 (#7 module)	64	6280~62BF	25216~25279
		Y10700~Y10777 (#8 module)	64	62C0~62FF	25280~25343
		Y11000~Y11077 (#9 module)	64	6300~633F	25344~25407
		Y11100~Y11177 (#10 module)	64	6340~637F	25408~25471
		Y11200~Y11277 (#11 module)	64	6380~63BF	25472~25535
		Y11300~Y11377 (#12 module)	64	63C0~63FF	25536~25599
		Y11400~Y11477 (#13 module)	64	6400~643F	25600~25663
		Y11500~Y11577 (#14 module)	64	6440~647F	25664~25727

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		Y11600~Y11677 (#15 module)	64	6480~64BF	25728~25791
		Y11700~Y11777 (#16 module)	64	64C0~64FF	25792~25855
	S	S0~S7999	8000	7000~8F3F	28672~36671
	SM	SM0~SM4095	4096	9000~9FFF	36864~40959
	T	T0~T4095	4096	A000~AFFF	40960~45055
	C	C0~C4095	4096	B000~BFFF	45056~49151
	ET	ET0~ET39	40	C000~C027	49152~49191
	SEM	SEM0~SEM127	128	C080~C0FF	49280~49407
	HM※1	HM0~HM6143	6144	C100~D8FF	49408~55551
	HS※1	HS0~HS999	1000	D900~DCEF	55552~56551
	HT※1	HT0~HT1023	1024	E100~E4FF	57600~58623
	HC※1	HC0~HC1023	1024	E500~E8FF	58624~59647
	HSC※1	HSC0~HSC39	40	E900~E927	59648~59687
Register, word object	D	D0~D20479	20480	0~4FFF	0~20479
	ID	ID0~ID99 (main body)	100	5000~5063	20480~20579
		ID10000~ID10099 (#1 module)	100	5100~5163	20736~20835
		ID10100~ID10199 (#2 module)	100	5164~51C7	20836~20935
		ID10200~ID10299 (#3 module)	100	51C8~522B	20936~21035
		ID10300~ID10399 (#4 module)	100	522C~528F	21036~21135
		ID10400~ID10499 (#5 module)	100	5290~52F3	21136~21235
		ID10500~ID10599 (#6 module)	100	52F4~5357	21236~21335
		ID10600~ID10699 (#7 module)	100	5358~53BB	21336~21435
		ID10700~ID10799 (#8 module)	100	53BC~541F	21436~21535
		ID10800~ID10899 (#9 module)	100	5420~5483	21536~21635
		ID10900~ID10999 (#10 module)	100	5484~54E7	21636~21735
		ID11000~ID11099 (#11 module)	100	54E8~554B	21736~21835
		ID11100~ID11199 (#12	100	554C~55AF	21836~21935

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		module)			
		ID11200~ID11299 (#13 module)	100	55B0~5613	21936~22035
		ID11300~ID11399 (#14 module)	100	5614~5677	22036~22135
		ID11400~ID11499 (#15 module)	100	5678~56DB	22136~22235
		ID11500~ID11599 (#16 module)	100	56DC~573F	22236~22335
	QD	QD0~QD99 (main body)	100	6000~6063	24576~24675
		QD10000~QD10099 (#1 module)	100	6100~6163	24832~24931
		QD10100~QD10199 (#2 module)	100	6164~61C7	24932~25031
		QD10200~QD10299 (#3 module)	100	61C8~622B	25032~25131
		QD10300~QD10399 (#4 module)	100	622C~628F	25132~25231
		QD10400~QD10499 (#5 module)	100	6290~62F3	25232~25331
		QD10500~QD10599 (#6 module)	100	62F4~6357	25332~25431
		QD10600~QD10699 (#7 module)	100	6358~63BB	25432~25531
		QD10700~QD10799 (#8 module)	100	63BC~641F	25532~25631
		QD10800~QD10899 (#9 module)	100	6420~6483	25632~25731
		QD10900~QD10999 (#10 module)	100	6484~64E7	25732~25831
		QD11000~QD11099 (#11 module)	100	64E8~654B	25832~25931
		QD11100~QD11199 (#12 module)	100	654C~65AF	25932~26031
		QD11200~QD11299 (#13 module)	100	65B0~6613	26032~26131
		QD11300~QD11399 (#14 module)	100	6614~6677	26132~26231
		QD11400~QD11499 (#15 module)	100	6678~66DB	26232~26331

Type	Component	Range	Quantity	Modbus address (hex)	Modbus address (decimal)
		QD11500~QD11599 (#16 module)	100	66DC~673F	26332~26431
	SD	SD0~SD4095	4096	7000~7FFF	28672~32767
	TD	TD0~TD4095	4096	8000~8FFF	32768~36863
	CD	CD0~CD4095	4096	9000~9FFF	36864~40959
	ETD	ETD0~ETD39	40	A000~A027	40960~40999
	HD※1	HD0~HD6143	6144	A080~B87F	41088~47231
	HSD※1	HSD0~HSD1023	1024	B880~BC7F	47232~48255
	HTD※1	HTD0~HTD1023	1024	BC80~C07F	48256~49279
	HCD※1	HCD0~HCD1023	1024	C080~C47F	49280~50303
	HSCD※1	HSCD0~HSCD39	40	C480~C4A7	50304~50343
	FD※2	FD0~FD8191	8192	C4C0~E4BF	50368~58559
	SFD※2	SFD0~SFD4095	4096	E4C0~FC2F	58560~64559
	FS※2	FS0~FS47	48	F4C0~F4EF	62656~62911

【Note】:

※1: The area marked with ※1 is the power failure retention area; Flash area marked with ※2.

※2: The addresses in the above table are used when the PLC is used as the lower computer and Modbus RTU or Modbus ASCII protocol communication is used. Generally, the upper computer is: configuration/touch screen/PLC.

※3: If the upper computer is a PLC, write the program according to the Modbus-RTU or Modbus-ASCII protocol.

※4: If the upper computer is SCADA or HMI, there are two situations: the first one has a Xinje driver, such as Xinje HMI, which can be directly written using the PLC internal software components (Y0/M0); The second type does not have a Xinje driver, so choose Modbus-RTU or Modbus-ASCII protocol, and then use the addresses in the above table to define data variables.

※5: The input and output points are in octal. Please calculate the corresponding input and output point Modbus address according to octal. For example, the Modbus address corresponding to Y0 is H6000, the Modbus address corresponding to Y10 is H6008 (not H6010), and the Modbus address corresponding to Y20 is H6010 (not H6020).

※6: When the Modbus address exceeds K32767, it needs to be represented in hexadecimal and the address needs to be preceded by "0". For example, the Modbus address of HD0 is 41088 in decimal (beyond K32767), and K41088 cannot be written to the software, so it needs to be represented as H0A080 in hexadecimal.

※7: Modbus address calculation for X and Y, taking X as an example, the Modbus address calculation for Y is the same as for X.

X0: 20480 X10: 20480+8 X20: 20480+16 X30: 16384+24….

X10000: 20736 X10010: 20736+8 X10020: 20736+16….

X1020: 20800 X10210: 20800+8 X10220: 20800+16….

3-2-3. MODBUS function code

Ethernet model PLC supports the following Modbus communication function codes:

Function code	Function	Descriptions
01H	Read coil	Read 0X address, max quantity is 2000
02H	Read input coil	Read 1X address, max quantity is 2000
03H	Read holding register	Read 4X address, max quantity is 125
04H	Read input register	Read 3X address, max quantity is 125
05H	Write single coil	Write single 0X address
06H	Write single register	Write single 4X address
0FH	Write multiple coils	Write 0X address, max quantity is 1976
10H	Write multiple registers	Write 4X address, max quantity is 123

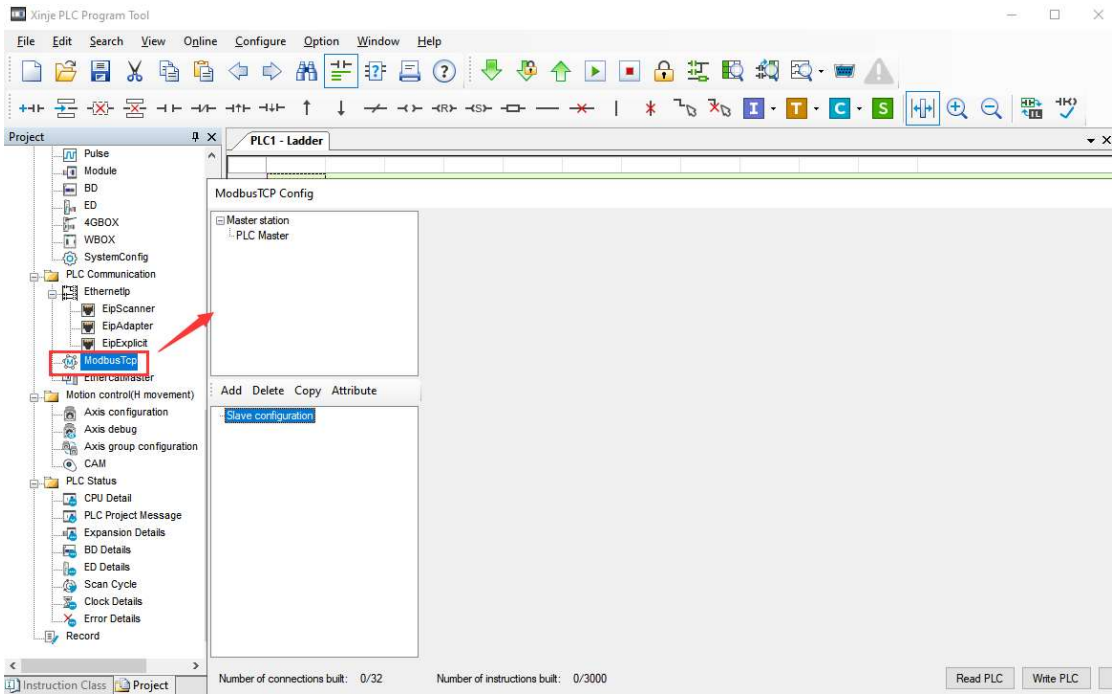
3-3. Modbus TCP graphics configuration

3-3-1. Overview

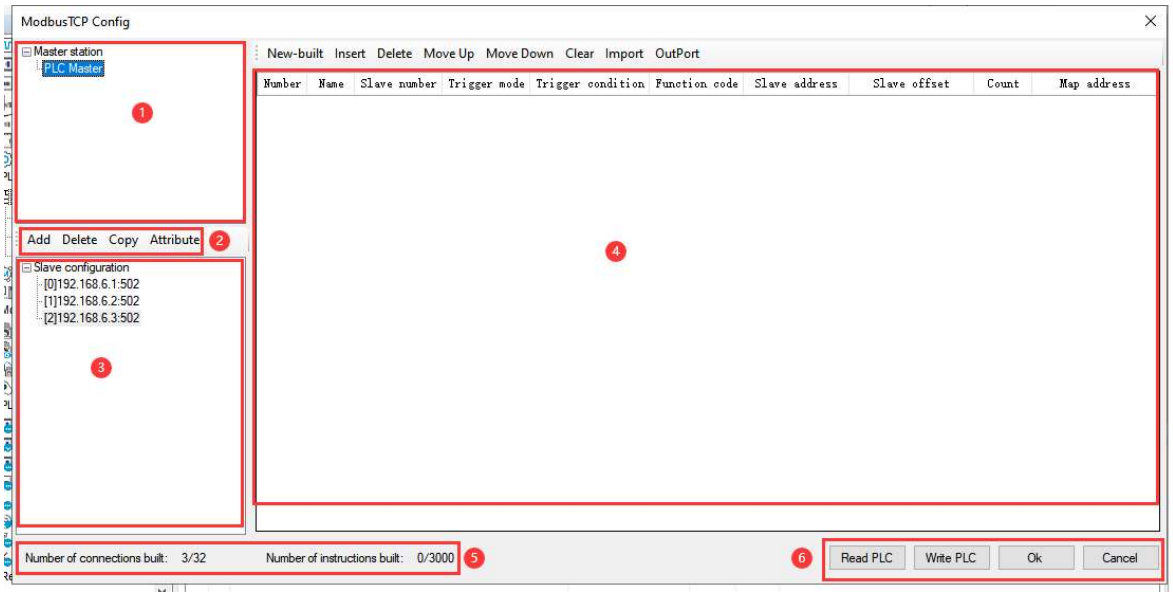
ModbusTCP, as a standard protocol for industrial communication, is widely used on site. Ethernet models integrate the MODBUS-TCP protocol, including servers and clients. In order to easily achieve communication data interaction with MODBUS-TCP devices, for XDH/XLH models, V3.7.3 and above firmware can support MODBUS-TCP graphical configuration. If users need to flexibly use MODBUS-TCP to achieve specific requirements, or if the device to be connected does not support MODBUS-TCP and only supports free format TCP/IP, communication data interaction can be achieved by establishing sockets. The establishment of sockets does not conflict with the establishment of connection resources through graphical configuration.

3-3-2. Modbus TCP master station configuration

Modbus TCP master (client) configuration for XDH/XLH models, supports establishing connections with 32 Modbus TCP slaves (server) simultaneously, and establishes a maximum of 3000 connection instructions for sharing with 32 Modbus TCP slaves (server). The process of establishing connection configuration is as follows:
Open the XDPpro software, click Modbus TCP in the left project bar to enter the configuration interface.



1. Modbus TCP Graphical Configuration Table

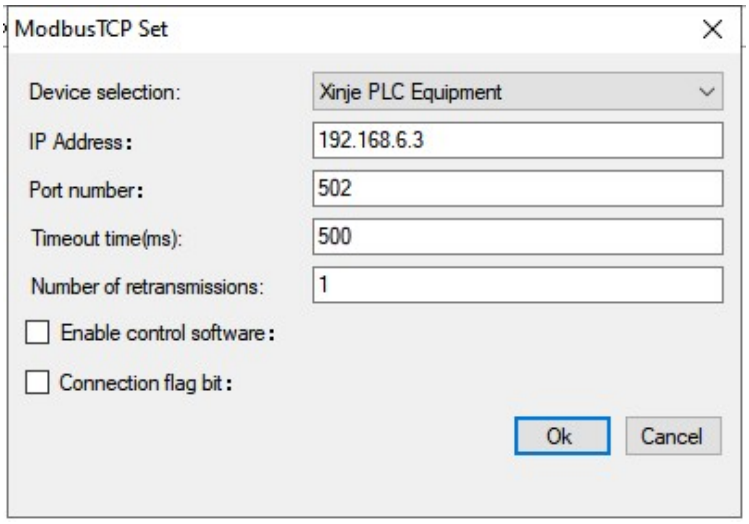


【Area 1】: Display the configuration information of the master station;

【Area 2】:

- Support the addition, deletion, replication, and attribute functions of a slave node

Add	Add a default slave node at the bottom and position the cursor to the added slave node.
Delete	Delete the selected slave node by the user. Clicking this function is invalid when the current tree node is empty.
Copy	The user clicks the copy button to copy a selected slave information (attribute+instruction configuration information) and automatically paste it to the bottom of the tree node. At the same time, the IP address is changed to the default IP

	address, and the cursor is positioned to the pasted slave node.
Attribute	Open the Modbus TCP settings interface of the selected slave node 

- The following contents can be set in the Modbus TCP configuration interface

Device selection	Xinje PLC and other Modbus equipment; Default Xinje PLC device
IP address	The IP address of the target PLC; Default 192.168.6.1, starting from 1, the next item defaults to the previous address +1
Port number	Fill in 502 by default
Time out	Default setting 500ms, range: 10-65535
Enable control software component	By default, it is not enabled. Enabling can set the coil control of the PLC. When not enabled: PLC automatically establishes a TCP connection to the target IP after running; When enabled: Only bit registers are supported, and TCP connections are only established to the target IP when the coil is set ON. When the conditions are not met, close the TCP connection
Connection flag bit	Store the result of the successful connection of this device in the corresponding connection flag bit register

【Area 3】: Display slave configuration information;

【Area 4】:

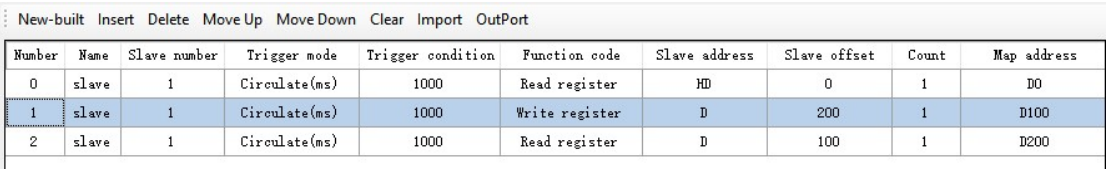
- Support users to select relevant instruction configuration functions for slave nodes, including create, insert, delete, move up, down, clear, import, and export.
- Display the command information of the selected slave node.

Number	Name	Slave number	Trigger mode	Trigger condition	Function code	Slave address	Slave offset	Count	Map address
0	slave	1	Circulate(ms)	1000	Read register	D	0	1	D0
1	slave	1	Circulate(ms)	1000	Read register	D	0	1	D0
2	slave	1	Circulate(ms)	1000	Read register	D	0	1	D0
3	slave	1	Circulate(ms)	1000	Read register	D	0	1	D0

【Area 5】: Monitor the number of connections currently established and the number of instructions established

【Area 6】: Supports functions such as read PLC, writ PLC, and save data (confirm, cancel).

2. Add instructions



Number	Name	Slave number	Trigger mode	Trigger condition	Function code	Slave address	Slave offset	Count	Map address
0	slave	1	Circulate(ms)	1000	Read register	HD	0	1	D0
1	slave	1	Circulate(ms)	1000	Write register	D	200	1	D100
2	slave	1	Circulate(ms)	1000	Read register	D	100	1	D200

- Name: The name of the current mapping instruction, which can be modified by the user;
- Slave station number: default 1, range 0-247;
- Triggering method: cycle (ms) and conditional triggering
  - Cycle (ms): When the triggering method is cycle, the value in the triggering condition is the cycle period, in ms; Range: 0-2 ^ 32-1;
  - Conditional trigger: When the trigger method is conditional trigger, the trigger condition is SM/M/HM coil or bit of word. Default to edge triggering, implemented by the PLC.
- Triggering conditions: Depending on the triggering method, check the validity of this item when "confirmed";
- Function code
  - When the user selects a Xinje PLC as the device

Read coil	The maximum number of reading coils supports 2000
Write coil	The maximum number of write coils is 1960
Read register	The maximum number of read registers supported 125
Write register	The maximum number of write registers supports 122

  - When the user selects other MODBUS devices

Read coil (01H)	Read 0X type addresses, maximum quantity 2000
Read input coil (02H)	Read 1X type addresses, maximum quantity 2000
Read register (03H)	Read 4X type addresses, maximum quantity 125
Read input register (04H)	Read 3X type addresses, maximum quantity 125
Write single coil (05H)	Write a single 0X type address
Write single register (06H)	Write a single 4X type address
Write multiple coils (0FH)	Write 0X type addresses, with a maximum number of 1960
Write multiple registers (10H)	Write 4X type addresses, with a maximum number of 122

- Slave station address space

  If the current slave station is a Xinje PLC, this is the register type corresponding to the function code. The reference settings are as follows:
  - Read and write coils, pull-down options: M, X, Y, HM, S, SM, T, C, ET, SEM, HS, HT, HC, HSC;
  - Read and write registers, pull-down options: D, HD, ID, QD, SD, TD, CD, ETD, HSD, HTD, HCD, HSCD, FD, SFD, FS.
- Quantity: The length of data that can be read or written, with a default of 1. The

maximum length of data that can be read or written depends on the above function code.

- Mapping address: coil status, cache address in the master station. Default is D0.

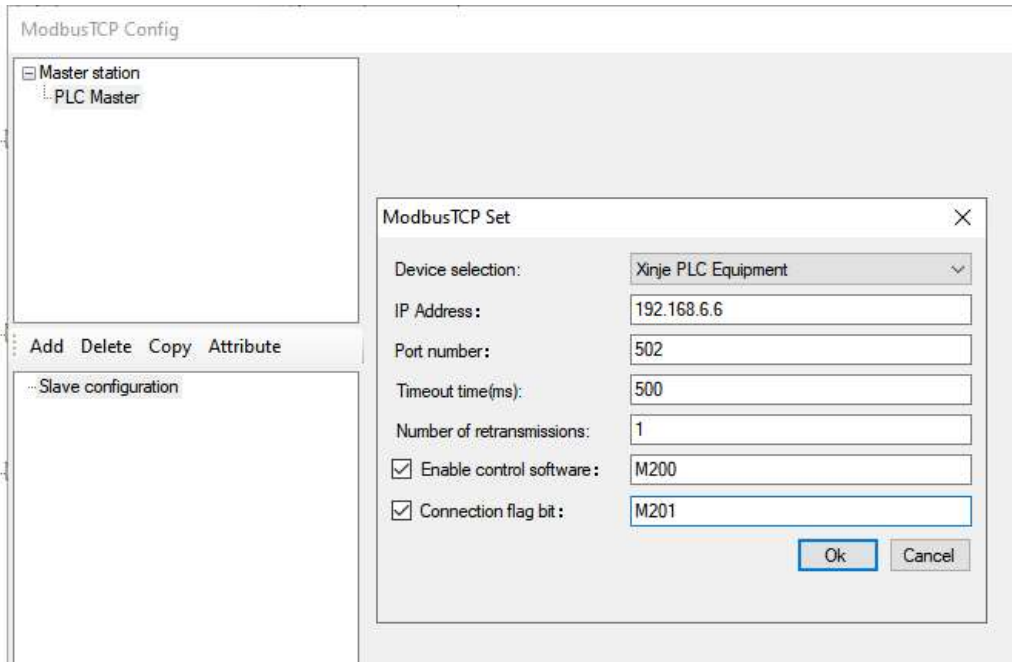
3-3-3. Modbus TCP graphical application

By using Modbus TCP graphical configuration function, automatic connection and data exchange between two PLCs can be established upon power on. Taking the communication between two XDH-60T4 as an example, the IP address of PLC 1 (client) is 192.168.6.10, and that of PLC 2 (server) is 192.168.6.6.

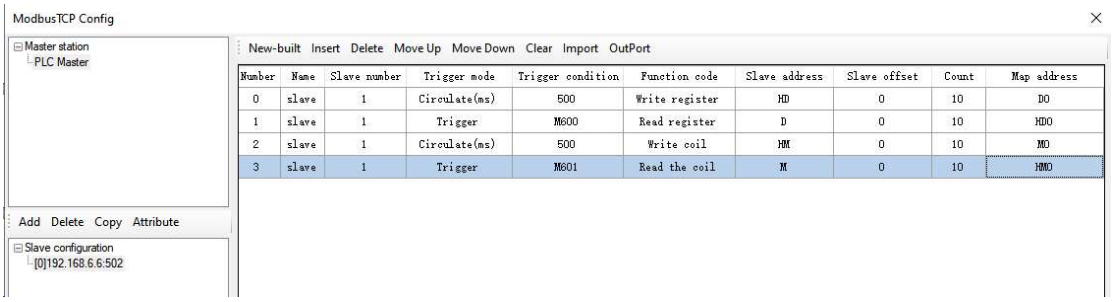
The operation of this case is as follows:

1. The client performs a register write operation and writes the data from the 10 registers of the client D0-D9 to the 10 registers of the server HD0-HD9 in a loop of 500ms as a trigger;
2. The client performs a register reading operation and reads the data from the 10 registers of D100-D109 on the server into the 10 registers of HD100-HD109 on the client using the trigger method (M600);
3. The client performs a write coil operation and writes the status of the 10 coils of the client M0-M9 to the ten coils of the server HM0-HM9 in a loop of 500ms as the trigger method;
4. The client performs a coil reading operation by triggering (M601) to read the status of the 10 coils of the server's M0-M9 into the 10 coils of the client's HM0-HM9.

(1) Configure the IP address and related configuration information of the slave station as follows:



(2) Create instructions for the four data interaction operations mentioned above, as follows:



(3) Check the status of the connection between the client and server, as well as information on data interaction, as shown in the following figure:

3-4. Free format protocol

Freedom communication based on Ethernet is divided into two categories: TCP and UDP, Ethernet model using TCP communication can be used as a TCP client (TCP client), can also be used as a TCP server (TCP server).

1. as a TCP client, take the initiative to establish a TCP connection with the TCP server, and bind socket ID.
2. as the TCP server, waiting for the TCP client and establish a TCP connection, and bind socket ID.
3. using UDP, listening to the specified local port, and bind socket ID.

Based on the above three forms, which can realize the freedom of Ethernet communication. Freeform communication in the form of a block of data to transmit data, restricted by PLC cache, a single to send and receive data volume of 1000 bytes.

Based on the above three forms, it can realize the free communication of Ethernet. Free format communication transfers the data in the form of data block, be restricted by PLC cache, single-time sending and receiving data volume is 1000 bytes.

Free format communication parameters:

Data buffer mode: 8-bit, 16-bit

1. 8-bit buffer communication: the high byte of the register is invalid, PLC only uses the low byte of the register to send and receive data.
2. 16-bit buffer communication: for the received data, PLC saves the low byte first, then saves the high byte; for the sending data, PLC sends the low byte first, then sends the high byte.
3. When the received data package length is larger than setting length, data will be stored as 16-bit buffer mode.

4 Ethernet communication instruction

4-1. Ethernet communication instruction overview

Ethernet communication instructions include: communication task opening and closing, send/receive data, MODBUS TCP. When using Ethernet instruction, please follow the following steps:

(1) open communications task: confirm the communication protocols and communication type, configure communication parameters, to create a TCP connection/UDP port listening, and bind socket ID.

(2) to realize the data communication: open successful communications task, achieve free Ethernet communication or MODBUS TCP data communications.

(3) close communications task: after communicating with target device, or TCP connection is abnormal, it needs to close communication tasks.

4-1-1. Create TCP connection/UDP port listening [S_OPEN]

1. Overview

Communication task creates the instruction, use together with abort communication task instruction S_CLOSE.

Create TCP connection /UDP port listening [S_OPEN]			
16-bit instruction	S_OPEN	32-bit instruction	-
Execution condition	Edge triggered	Suitable model	XD3E, XD5E, XDME, XDH, XG, XL5E, XL5N, XLME, XL5H, XLH
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

Operand	Function	Type
S1	Socket ID	16-bit, BIN
S2	Communication type	16-bit, BIN
S3	Local device communication mode	16-bit, BIN
S4	Parameter block start address	16-bit, BIN
S5	Flag start position	Bit

3. Suitable soft component

word

operand	System									constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
S1	●									●		
S2	●									●		
S3	●									●		
S4	●											

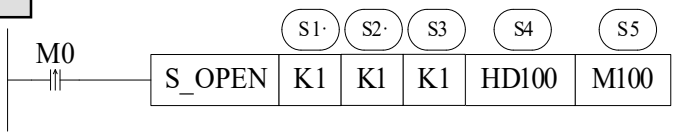
Bit

operand	System						
	X	Y	M*	S*	T*	C*	Dnm
S5			●				

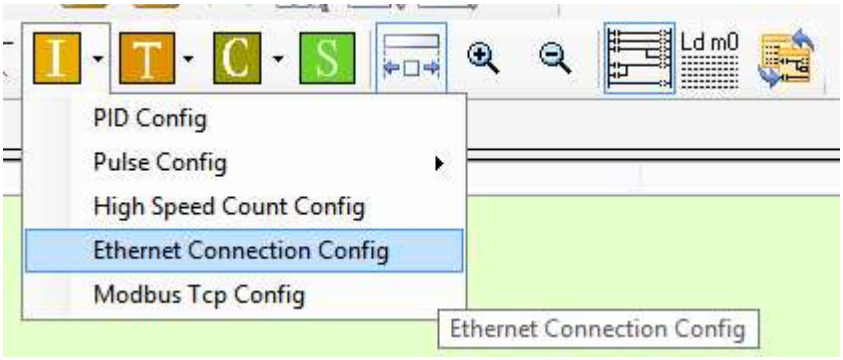
***Note: D means D HD ; TD means TD HTD ; CD means CD HCD HSCD HSD; DM means DM DHM;**

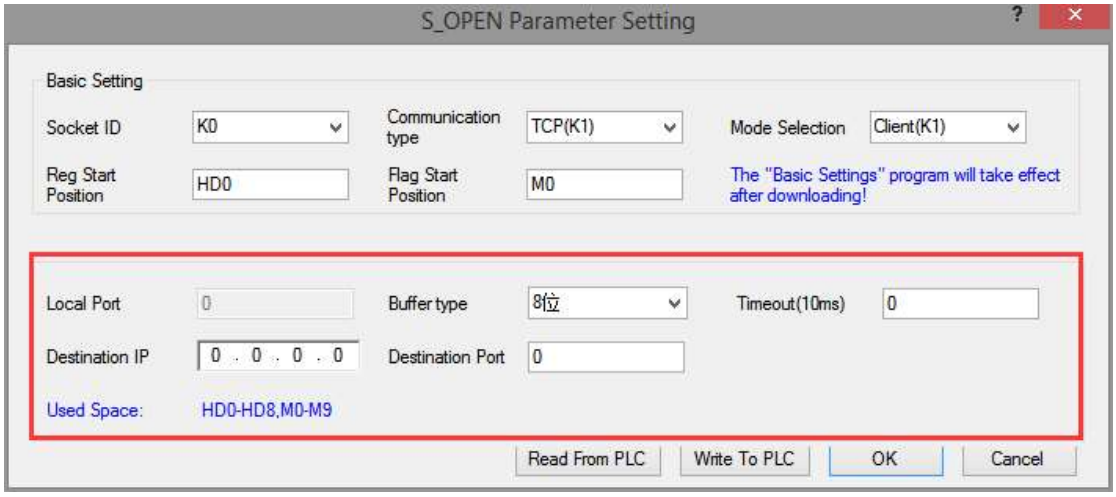
DS means DS DHS. M means M HM SM; S means S HS; T means T HT; C means C HC.

Function and action



- Create the communication task, when M0 rising edge is coming, the instruction will create one TCP connection or open UDP port listening once.
- S1: socket ID, range: K0~K63. Note: the socket quantity cannot be more than 64, TCP quantity cannot over 32, UDP quantity cannot be more than 32.
- S2: communication type, range: K0, K1. K0 is UDP, K1 is TCP.
- S3: communication mode. Range: K0, K1. K0 is server, K1 is client.
- S4: parameter block start address, occupy 9 registers from S4 to S4+8.
- S5: flag start position, occupy 10 coils from S5 to S5+9.
- This instruction can be set through the following window
  Note: 1. The server needs to first open the socket and wait for the client to connect, otherwise the socket may not be established successfully.
  2. The UDP multicast function is only supported for Ethernet PLC firmware version 3.7.2 and above, and XDH/XLH series firmware 3.7.3 and above versions.

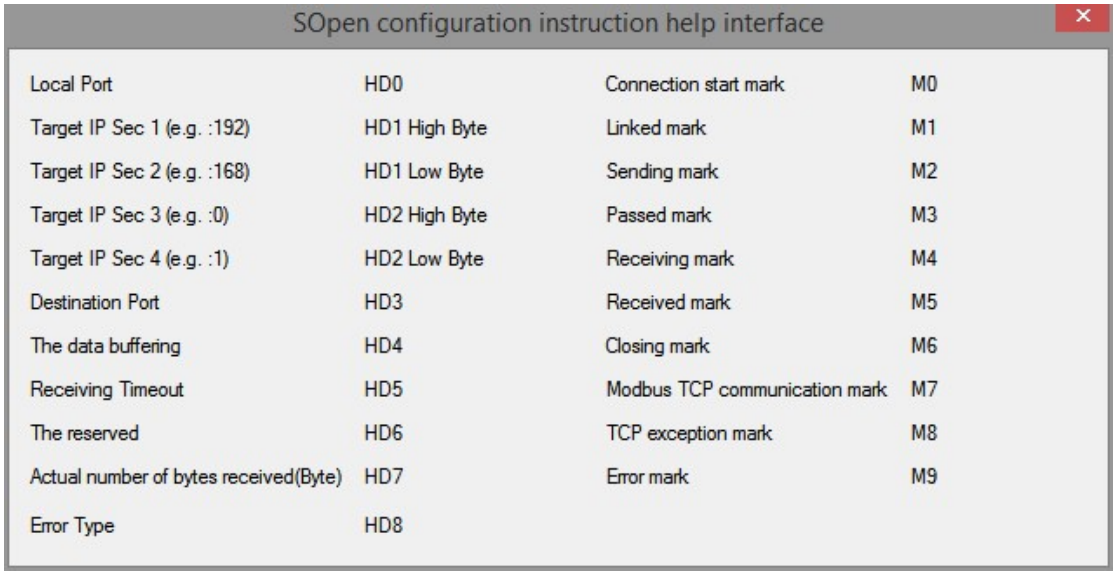




Note: the parameters in the red frame will be effective after power on the PLC again.

- Ethernet error flag SM1921 is ON when communication is abnormal, the error information will be stored in SD1920 and SD1921, please refer to chapter 4-3.

Take above image as an example, the address starting from HD0 and flag address starting from M0 are shown as below:



Parameter explanation:

The communication task created by S_OPEN is divided into three categories: TCP client, TCP server, UDP. The parameters used by the three types are different, please refer to below table.

Communication type	Local port	destination IP	Destination port	Buffer type	Timeout	Received bytes	Error code
TCP client	-	√	√	√	√	√	√
TCP server	√	-	-	√	√	√	√
UDP	√	√	√	√	√	√	√

1. Local port

The range is 1 to 60000, port 502 and 531 is special port which can not be used. Local port only can be used by one communication task.

2. Destination IP

The target IP refers to the IP address of the target communication device, with a value range of 0-254, which is in the same subnet as the local machine.

3. Destination port

The net port no. of target device. The range is 1 to 65535. The port must be 502 for modbus tcp communication.

4. Data Buffer mode

When the value is 0, it is 8-bit mode. When the vlaue is non-zero, it is 16-bit mode. The actual received data packet length is received based on the corresponding set buffer length.

5. Timeout

The time from PLC requests data receiving to the receiving data ends. The range is 0 to 65536. The unit is 10ms. 0 means the timeout is disabled, it will continue receiving data. Non-zero means the timeout function is enabled. The receiving timeout is effective for S_RCV and M_TCP.

If the timeout is set to 300ms, it will wait for 300ms when the request begins, and terminate at once when the data is received successfully. If it hasn't received data over 300ms, the present instruction will end and report the receiving timeout error.

Note: When the receive timeout time is set to 0 in versions V3.7.3 and above, M_TCP will default to a receive timeout time of 10 seconds, and S_RCV will default to no receive timeout time.

6. TCP keep alive

(1) the value is 0, TCP keep alive function is not enabled.

(2) the value is non-zero, TCP keep alive function is enabled.

Connection is in the inactive state over a period of time, when the keep alive function is enabled, it will send keep alive detection to the object, if the sender did not receive the response message, then the other host will be confirmed as unreachable. Triggering time is 1 ~ 5 min, when it is abnormal, TCP abnormal flag is set on.

Note: The TCP keep alive function is only supported for Ethernet based PLCs with firmware versions 3.7.2 and above.

7. Data receiving mode

Automatic reception: If the other party sends too quickly during reception, the data that is not received will be automatically discarded; Not receiving or receiving timeout will also discard the data sent by the other party.

8. Receiving data length

Execute S_ RCV instruction, the actual length of received data, in bytes.

9. Error code

The error message when Ethernet free format communication and Modbus TCP communication are abnormal, please refer to chapter 4-4.

10. Flag bit

The functional description of communication related flag bits is shown in the table below:
(Address description starts with Mn)

Bit address	Flag bit	Function
Mn	Connecting	Creating the connection, M (n) is ON
M (n+1)	Connected	Creating connection completed, M (n+1) is ON
M (n+1)	Sending	Data is sending, M (n+2) is ON
M (n+3)	Sent	Sending data completed, M (n+3) is ON
M (n+4)	Receiving	Data is receiving, M (n+4)is ON
M (n+5)	Received	Data receiving completed, M (n+5) is ON
M (n+6)	Closing	The present connection is closing, M (n+6) is ON
M (n+7)	MODBUS TCP communicating	When executing M_TCP instruction, M (n+7) is ON
M (n+8)	TCP abnormal	TCP connection is abnormal, M (n+8) is ON
M (n+9)	Error flag	Communication is error, M (n+9) is ON

4-1-2. Communication termination [S_CLOSE]

1. Instruction overview

Communication termination instruction, please use together with S_OPEN.

Communication termination [S_CLOSE]			
16-bit	S_CLOSE	32-bit	-
Execution condition	Edge triggering	Suitable model	XD3E, XD5E, XDME, XDH, XG, XL5E, XL5N, XLME, XL5H, XLH
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

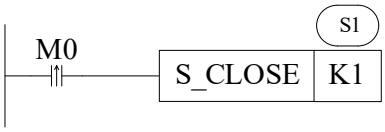
Operand	Function	Type
S1	Close socket ID	16-bit, BIN

3. Suitable soft component

word	operand	System									Constant	Module	
		D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
	S1	●									●		

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

Function and action



- Terminate the communication task when the rising edge of M0 is coming.
  Note: this instruction must be used together with S_OPEN.
- S1: the socket ID which needs to close, the operand can be register or constant, the range is K0~K63.
- After this instruction is executed, the instruction M_TCP, S_SEND, S_RCV based on this socket ID cannot run anymore.

4-1-3. Free format communication send [S_SEND]

1. Instruction overview

Free format communication send instruction needs to use together with S_OPEN and S_CLOSE.

Free format communication send [S_SEND]			
16-bit	S_SEND	32-bit	-
Execution condition	Edge triggering	Suitable model	XD3E, XD5E, XDME, XDH, XG, XL5E, XL5N, XLME, XL5H, XLH
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

Operand	Function	Type
S1	Socket ID	16-bit, BIN
S2	Send data local register head address	16-bit, BIN
S3	Send data quantity	16-bit, BIN

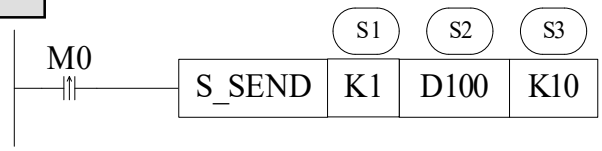
3. Suitable soft component

word

operand	System									Constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
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.**

Function and action



- Free format communication send instruction, it will send data when the M0 rising edge is coming.
  Note: this instruction must be used together with S_OPEN and S_CLOSE.
- S1: socket ID, the operand can be register or constant, the range is K0~K63
- S2: local register sending head address
- S3: send data quantity, the operand can be register or constant
- Please input this instruction in the ladder chart
- When using, pay attention to the data buffer type in the S_OPEN instruction in the socket ID (16 bits/8 bits).
- When the buffer bit is 8 bits, only the low byte data of the register should be sent. For example, to send the low byte data in registers D100 to D107, S3 should be set to 8.
- When the buffer bit is 16 bits, both high and low byte data of the register will be sent. For example, to send high and low byte data from D100 to D107, S3 should be set to 16, and when sending, the low byte should be in front of the high byte.

4-1-4. Free format communication receive [S_RCV]

1. Instruction overview

Free format communication receive instruction needs to use together with S_OPEN and S_CLOSE.

Free format communication receive [S_RCV]			
16-bit	S_RCV	32-bit	-
Execution condition	Normally ON/OFF, edge triggering	Suitable model	XD3E, XD5E, XDME, XDH, XG, XL5E, XL5N, XLME, XL5H, XLH
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

Operand	Function	Type
S1	Socket ID	16-bit, BIN
S2	Receive data local register head address	16-bit, BIN
S3	Receive data quantity	16-bit, BIN

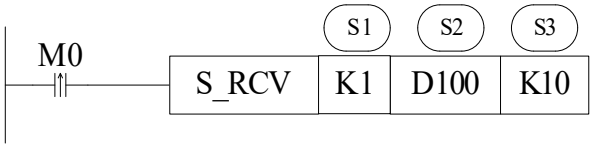
3. Suitable soft component

word

operand	System									Constant	Module	
	D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
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.**

Function and action



- Free format communication receive instruction, it will receive data when the M0 rising edge is coming.
  Note: this instruction must be used together with S_OPEN and S_CLOSE.
- S1: socket ID, the operand can be register or constant, the range is K0~K63

- S2: local register receiving head address
- S3: receive data quantity, the operand can be register or constant
- Please input this instruction in the ladder chart
- When using, pay attention to the data buffer type in the S_OPEN instruction in the socket ID (16 bits/8 bits).
- When the buffer bit is 8 bits, only the low byte data of the register should be sent. For example, to send the low byte data in registers D100 to D107, S3 should be set to 8.
- When the buffer bit is 16 bits, both high and low byte data of the register will be sent. For example, to send high and low byte data from D100 to D107, S3 should be set to 16, and when sending, the low byte should be in front of the high byte.

4-1-5. MODBUS communication [M_TCP]

1. Instruction overview

When PLC is client, receive and send data in modbus tcp protocol. It can be used together with S_OPEN and S_CLOSE.

MODBUS TCP communication [M_TCP]			
16-bit	M_TCP	32-bit	-
Execution condition	Edge triggering	Suitable model	XD3E, XD5E, XDME, XDH, XG, XL5E, XL5N, XLME, XL5H, XLH
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

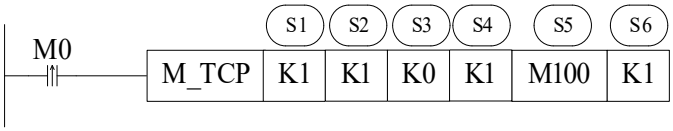
Operand	Function	Model
S1	Remote station no.	16-bit, BIN
S2	Modbus communication function code	16-bit, BIN
S3	Target head address	16-bit, BIN
S4	Register or coil quantity	16-bit, BIN
S5	Local head address	16-bit, BIN
S6	Socket ID	16-bit, BIN

3. Suitable soft component

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

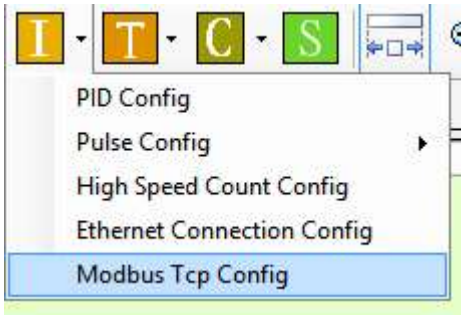
***Note: D means D HD ; TD means TD HTD ; CD means CD HCD HSCD HSD; DM means DM DHM; DS means DS DHS.**

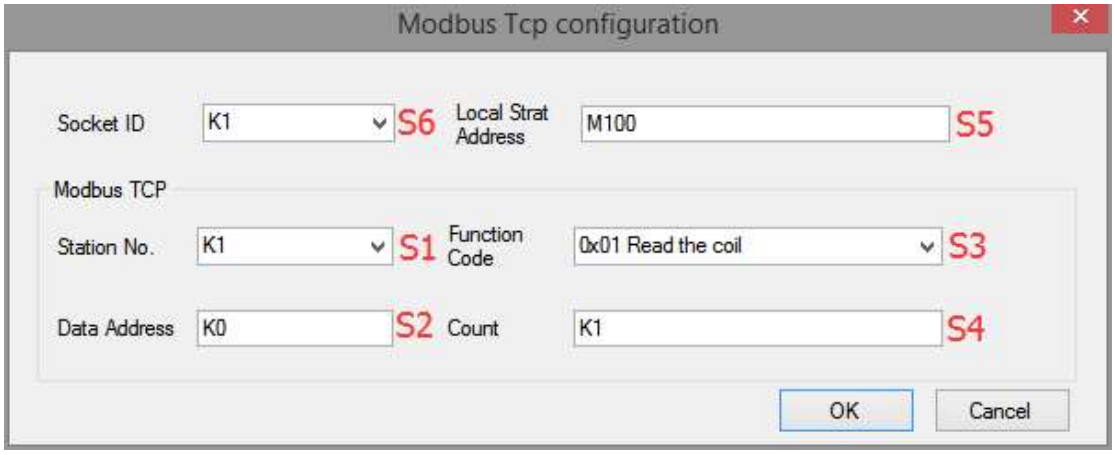
Function and action



- MODBUS TCP communication instruction, it will Modbus TCP communicate once when M0 rising edge is coming.
- S1: remote communication station no., the range is K0~K247
- S2: MODBUS communication function code
- S3: target head address, it is Modbus communication address.
- S4: communication data quantity
- S5: local head address
- S6: socket ID, specify the TCP connection, the target port must be 502.
- This instruction must be used together with S_OPEN and S_CLOSE.
- M_TCP is only effective when PLC is client, and receives and sends the data of Modbus TCP protocol.

  Note: As a server, ModbusTCP has a port number of 502 and does not require writing communication instructions. The client can establish a socket and write communication instructions.
- This instruction needs to set through the following window





Function code:

Value	Function code	Value	Function code
K1	Read the coil	K3	Read the register
K2	Read the input discrete magnitude	K4	Read input register
K5	Write single coil	K6	Write single register
K15	Write multiple coil	K16	Write multiple register

4-1-6. Ethernet communication example

Example 1:

By using the following program, PLC can automatically create three forms of communication tasks: TCP client, TCP server, and UDP after power on, and achieve data transmission and reception based on each communication task. The IP address of PLC 1 is 192.168.1.12, and the IP address of PLC 2 is 192.168.1.6.

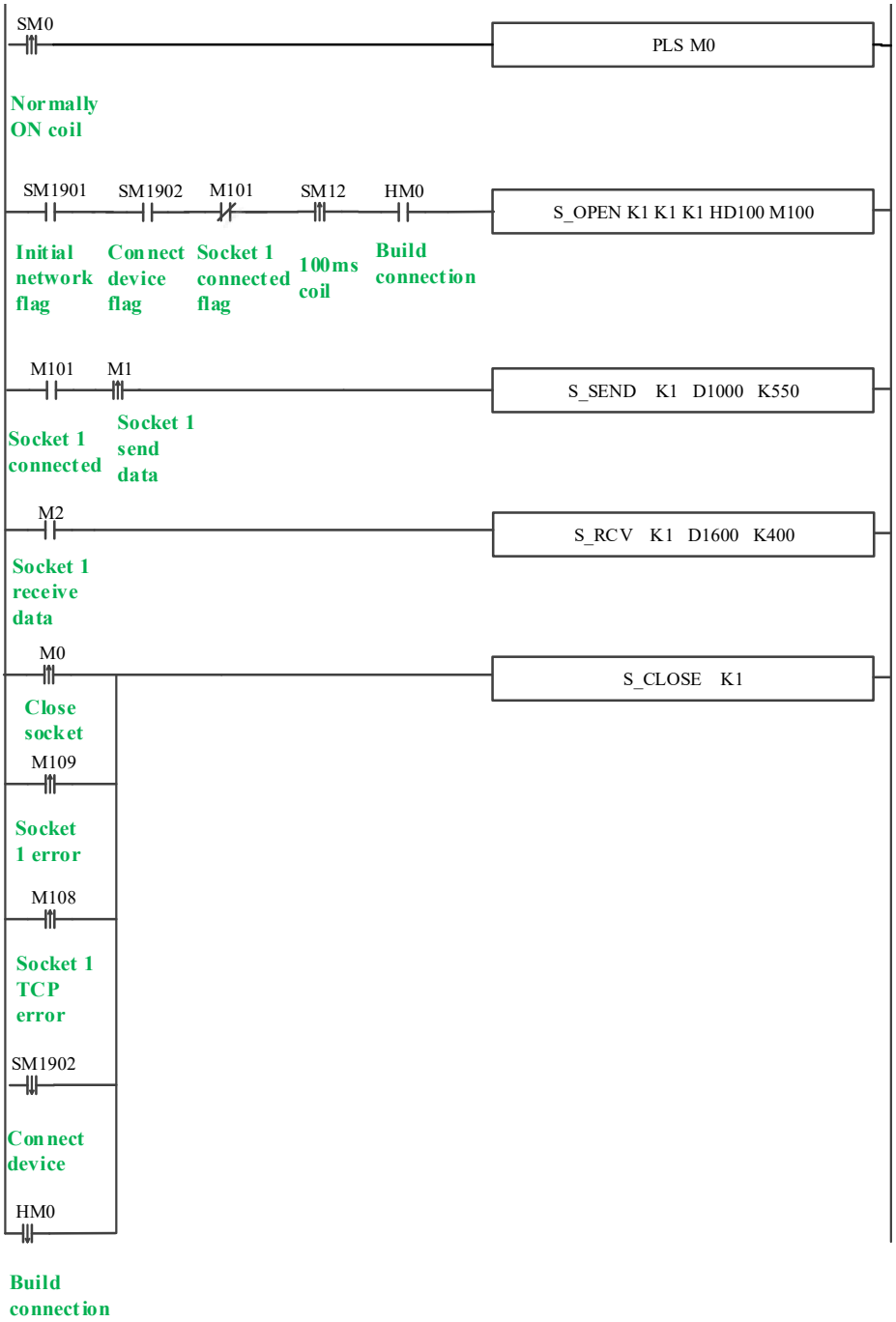
Note: The server needs to first open the socket and wait for the client's connection, otherwise the socket may not be established successfully.

Program operation:

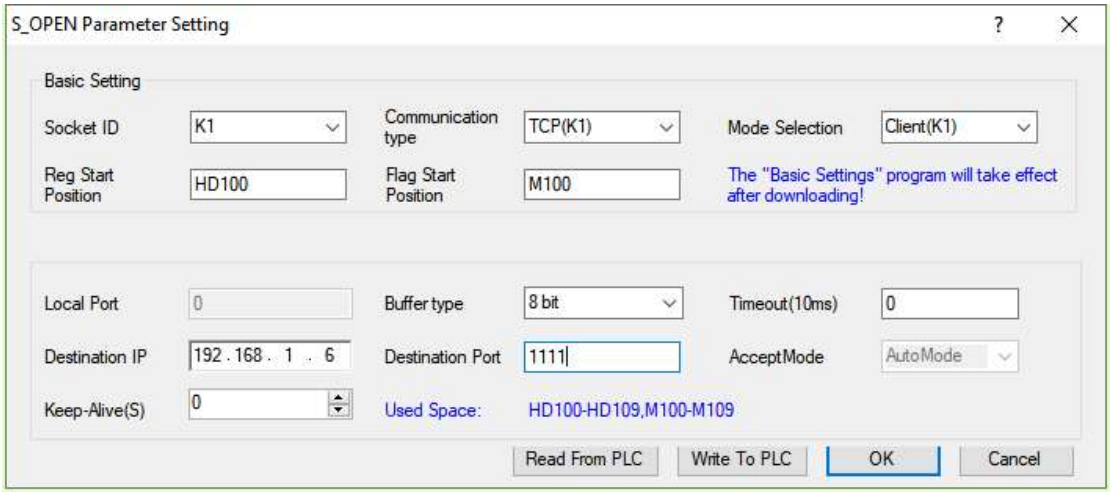
(1) After PLC 1 is powered on, it actively establishes a TCP connection to the TCP server service port 1111 of PLC 2 as a TCP client and binds a socket ID of 1. After the connection is successfully established, it sends the low 8-bit of D1000~D1549 to PLC 2 D2600~D3149, while continuously receiving data from PLC 2 D2000~D2399 and storing it in the low 8-bit of registers D1600~D1999. When a TCP connection encounters an exception or the sender does not receive a response message within the set live time (where the live time is set to 2 seconds), the TCP connection is actively closed and rebuilt.

Due to the varying number of Ethernet ports in different series of PLCs, please distinguish which Ethernet port the network cable is connected to when using communication related coils SM1902 or SM1903. (SM1902 is the symbol for connecting network devices, used in the first network port of a dual port model or in a single port model to connect to a switch/router/other network device. SM1903 is the symbol for connecting network devices, used in the second network port of a dual port model to connect to a switch/router/other network device).

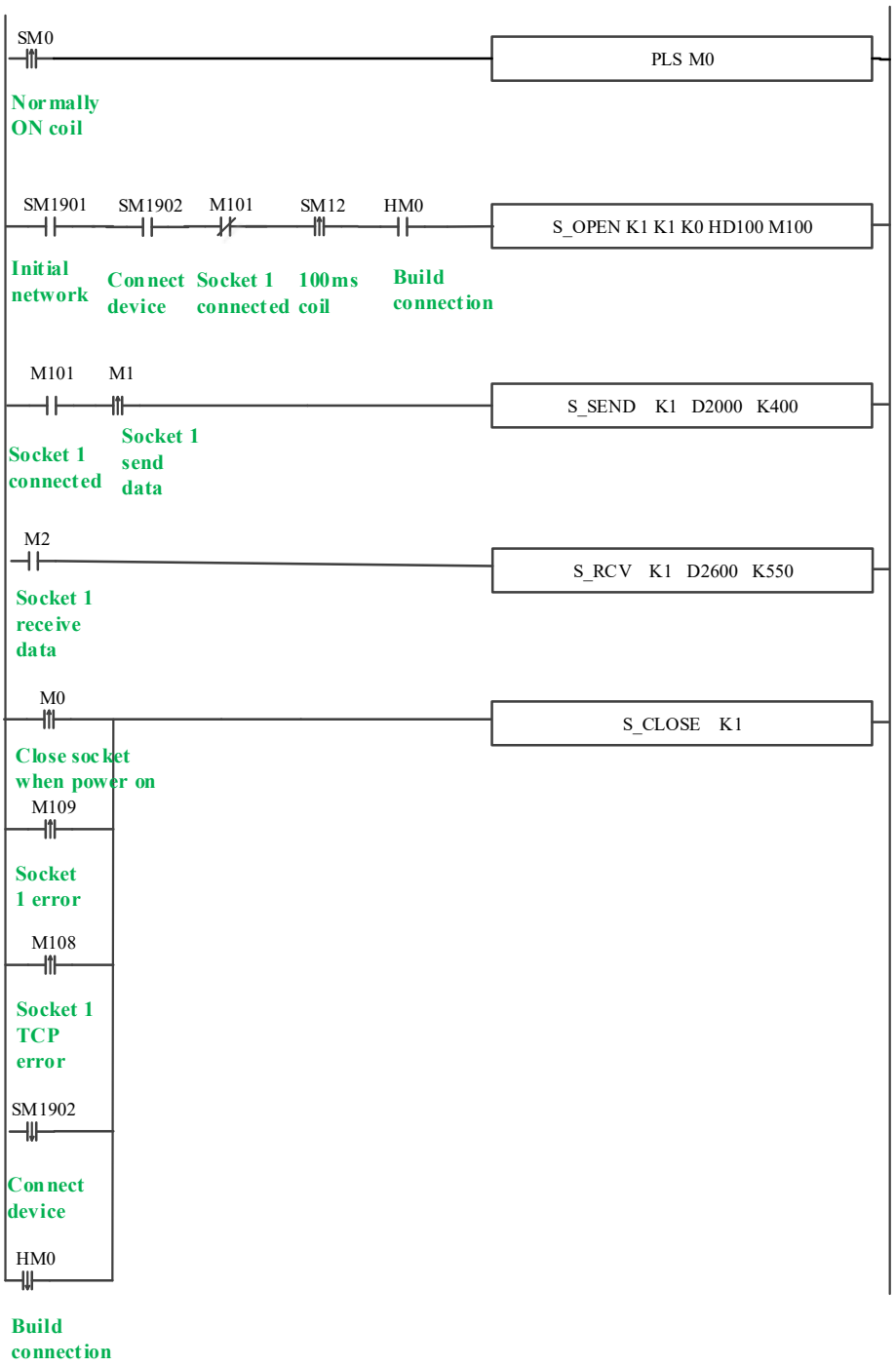
PLC1 program:


SM0
PLS M0
Normally ON coil
SM1901
SM1902
M101
SM12
HM0
S_OPEN K1 K1 K1 HD100 M100
Initial network flag
Connect device flag
Socket 1 connected flag
100ms coil
Build connection
M101
M1
S_SEND K1 D1000 K550
Socket 1 connected
Socket 1 send data
M2
S_RCV K1 D1600 K400
Socket 1 receive data
M0
S_CLOSE K1
Close socket
M109
Socket 1 error
M108
Socket 1 TCP error
SM1902
Connect device
HM0
Build connection

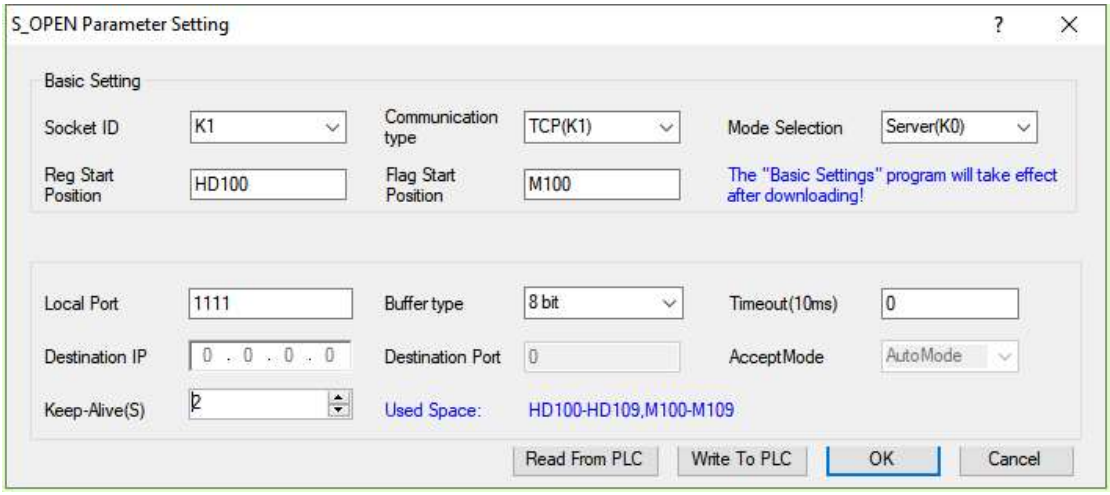
The configuration information for the client socket S_OPEN is as follows:



PLC 2 program:

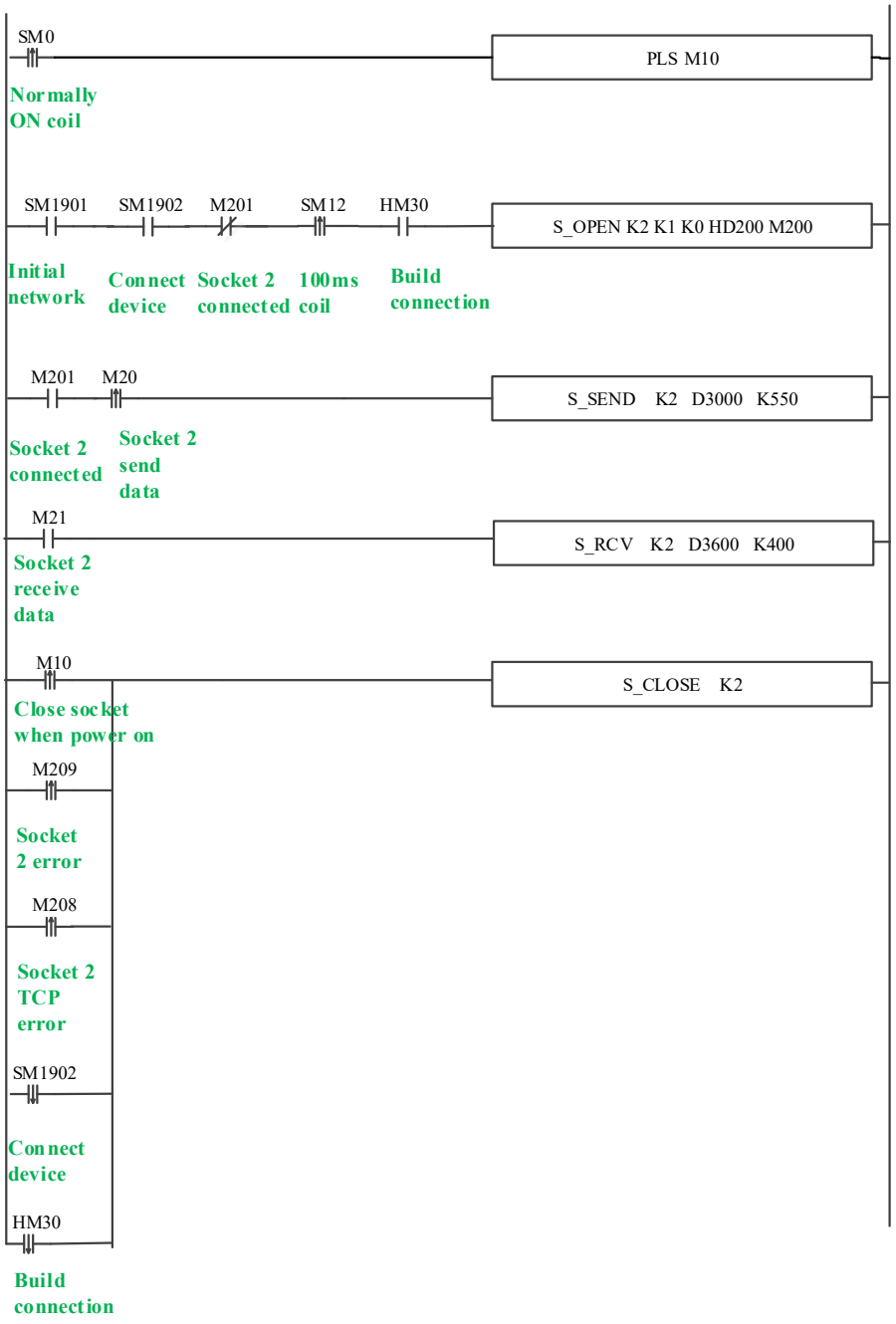

SM0
PLS M0
Normally ON coil
SM1901
SM1902
M101
SM12
HM0
S_OPEN K1 K1 K0 HD100 M100
Initial network
Connect device
Socket 1 connected
100ms coil
Build connection
M101
M1
S_SEND K1 D2000 K400
Socket 1 connected
Socket 1 send data
M2
S_RCV K1 D2600 K550
Socket 1 receive data
M0
S_CLOSE K1
Close socket when power on
M109
Socket 1 error
M108
Socket 1 TCP error
SM1902
Connect device
HM0
Build connection

The configuration information for server socket S_OPEN is as follows

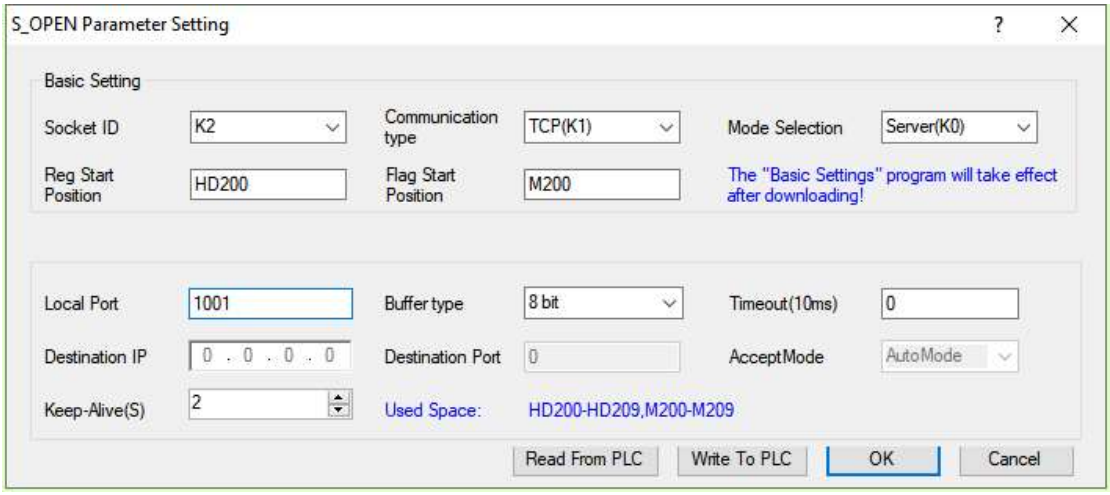


(2) After PLC 1 is powered on, it actively listens to port 1001 as a TCP server and waits for the TCP client device of PLC 2 to establish a TCP connection and bind a socket ID of 2. After the connection is successfully established, it sends the low 8-bit of D3000-D3549 to the connected device PLC 2, while continuously receiving data from the connected device PLC 2. The data is stored in the low 8-bit of registers D3600-D3999. When a TCP connection encounters an exception or the sender does not receive a response message within the set live time (where the live time is set to 2 seconds), the TCP connection is actively closed and rebuilt.

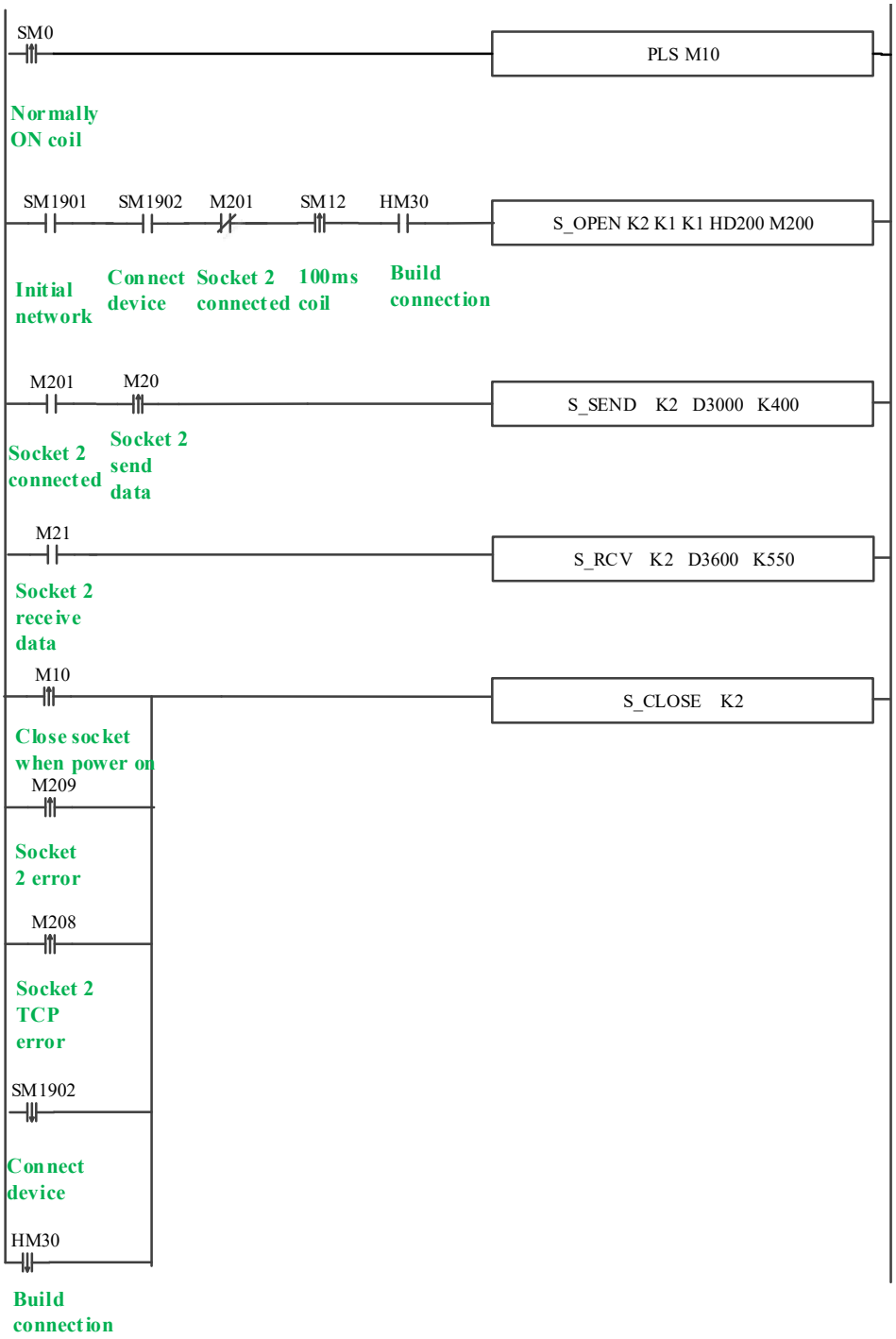
PLC 1 program:


SM0
PLS M10
Normally ON coil
SM1901
SM1902
M201
SM12
HM30
S_OPEN K2 K1 K0 HD200 M200
Initial network
Connect device
Socket 2 connected
100ms coil
Build connection
M201
M20
S_SEND K2 D3000 K550
Socket 2 connected
Socket 2 send data
M21
S_RCV K2 D3600 K400
Socket 2 receive data
M10
S_CLOSE K2
Close socket when power on
M209
Socket 2 error
M208
Socket 2 TCP error
SM1902
Connect device
HM30
Build connection

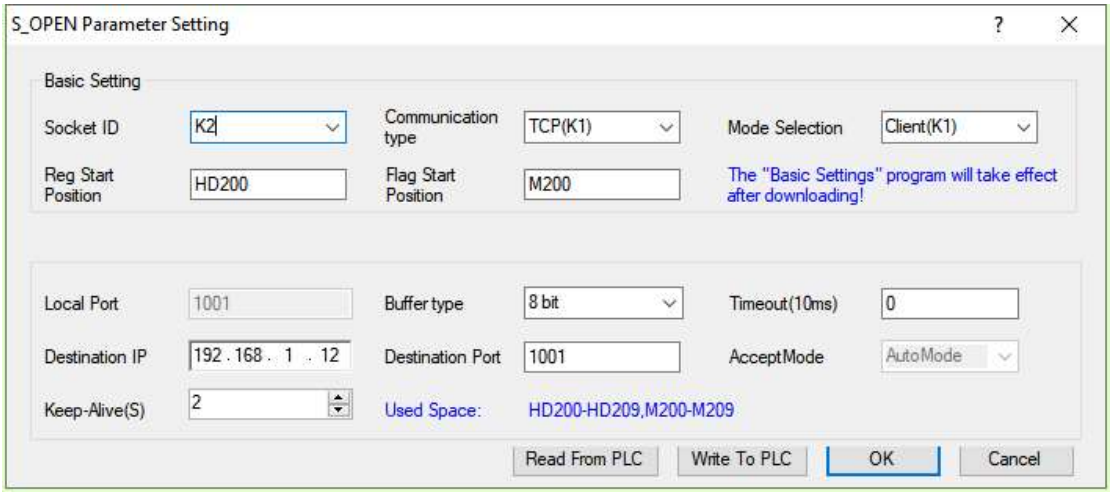
The configuration information for server socket S_OPEN is as follows



PLC 2 program:

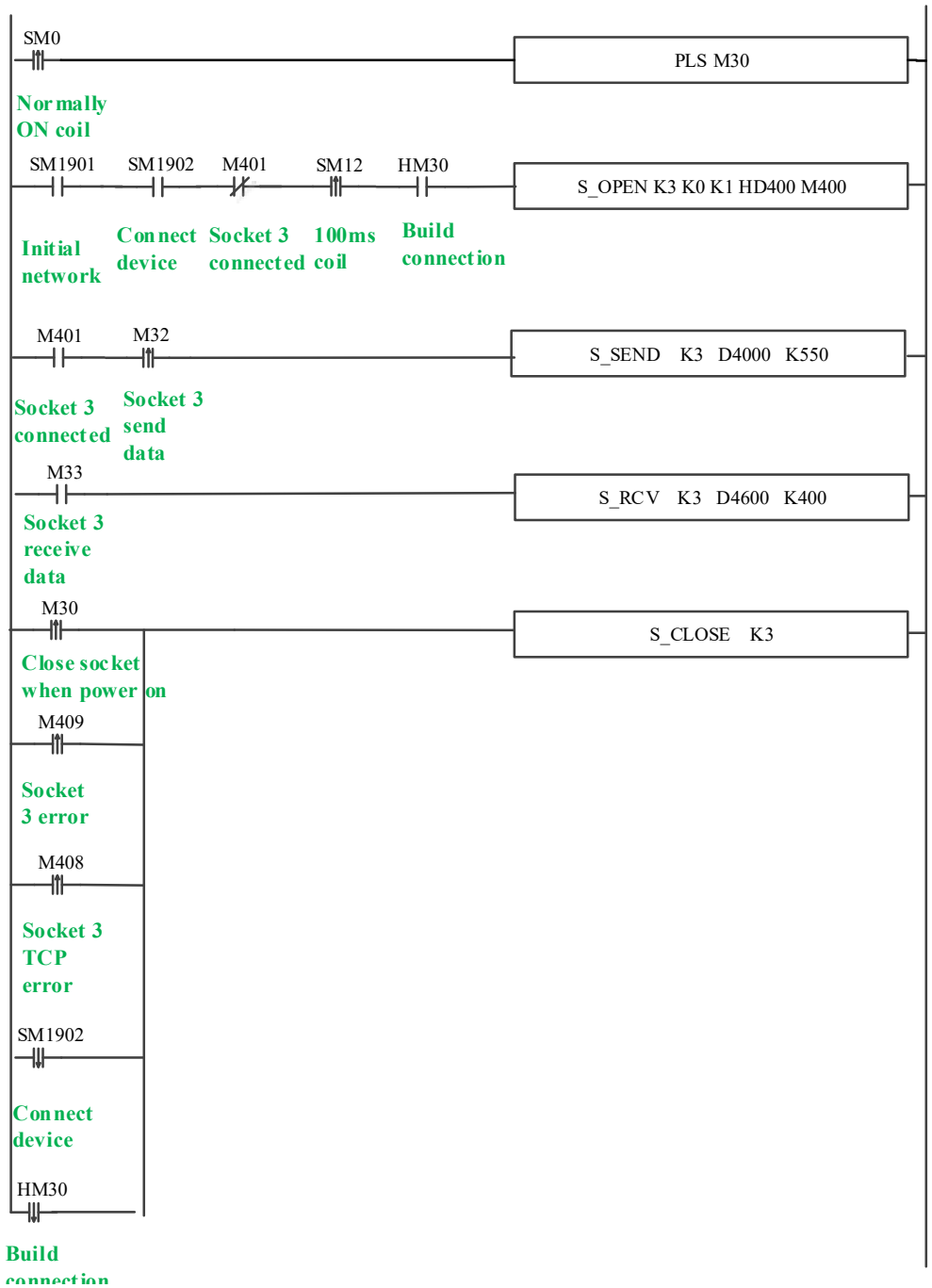

SM0
PLS M10
Normally ON coil
SM1901
SM1902
M201
SM12
HM30
S_OPEN K2 K1 K1 HD200 M200
Initial network
Connect device
Socket 2 connected
100ms coil
Build connection
M201
M20
S_SEND K2 D3000 K400
Socket 2 connected
Socket 2 send data
M21
S_RCV K2 D3600 K550
Socket 2 receive data
M10
S_CLOSE K2
Close socket when power on
M209
Socket 2 error
M208
Socket 2 TCP error
SM1902
Connect device
HM30
Build connection

The configuration information for client socket S_OPEN is as follows

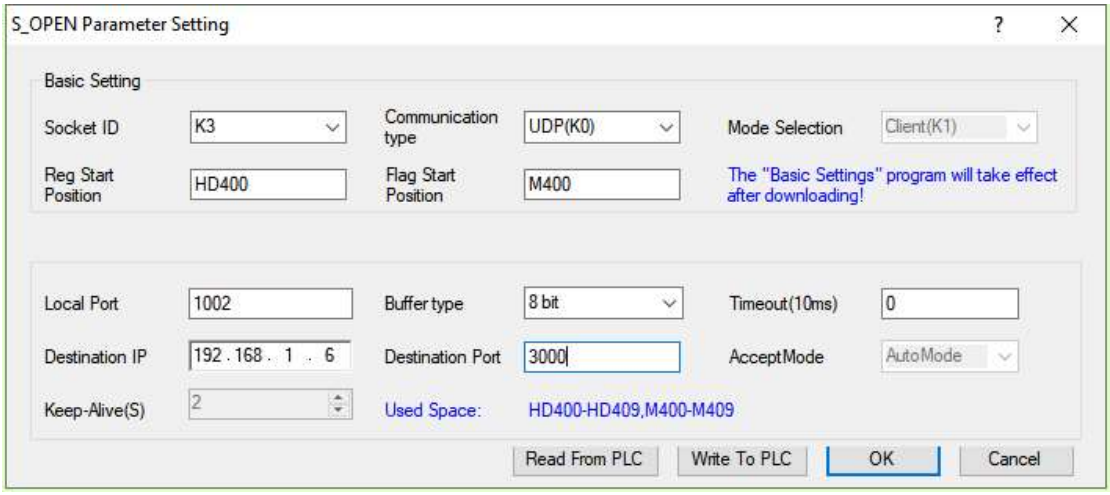


(3) After powering on PLC 1, UDP communication is used with an IP address of 192.168.1.12. The local port is set to 1002, the target IP is 192.168.1.6, and the target port is 3000. The socket ID is bound to 3. After the connection is successfully established, the low 8-bit of D4000-D4549 are sent to PLC 2, and the data from PLC2 is continuously received and stored in registers D4600~D4999. When UDP unicast encounters abnormal connections, it actively closes the UDP unicast connection and reconstructs the connection.

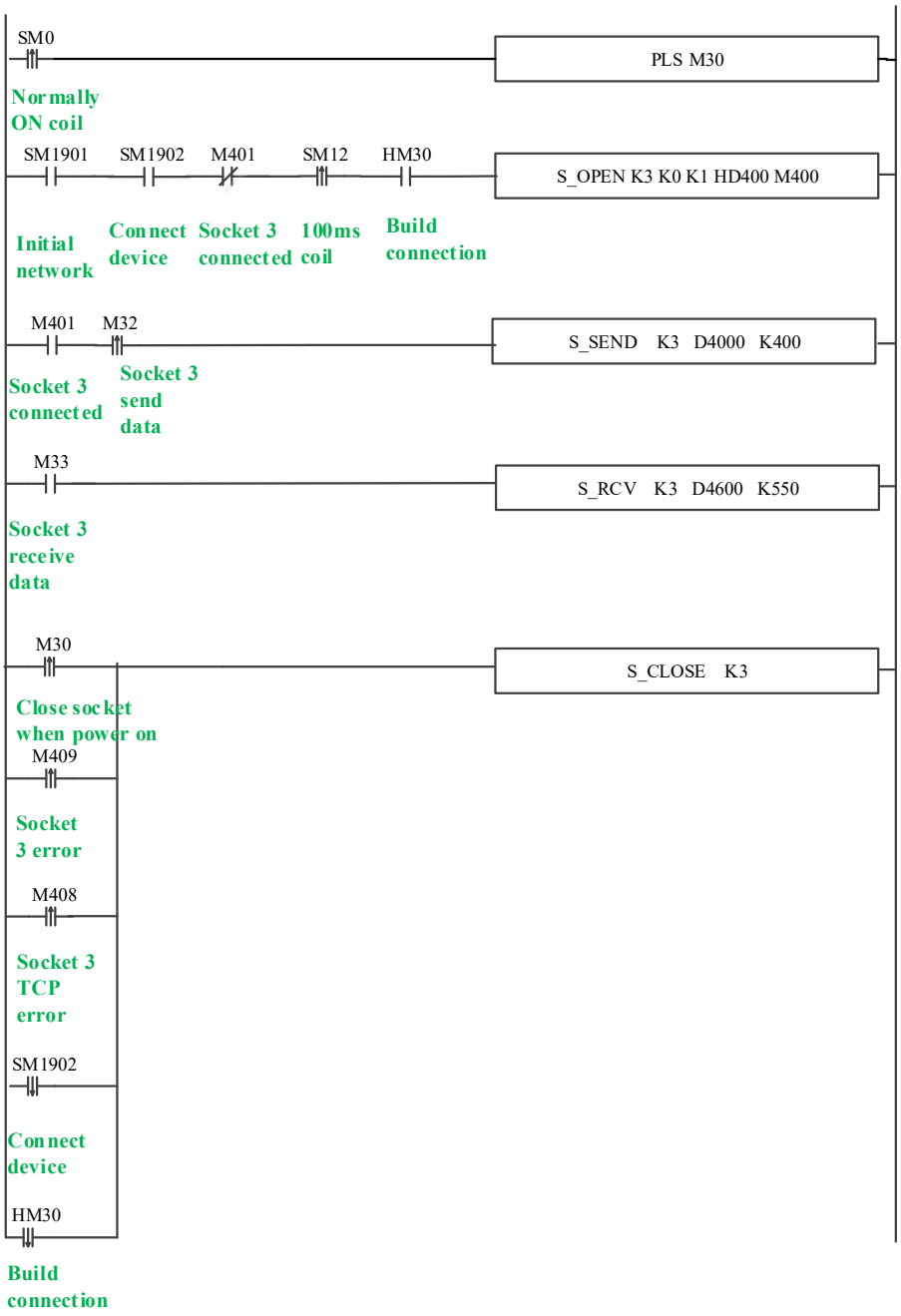
PLC 1 program:



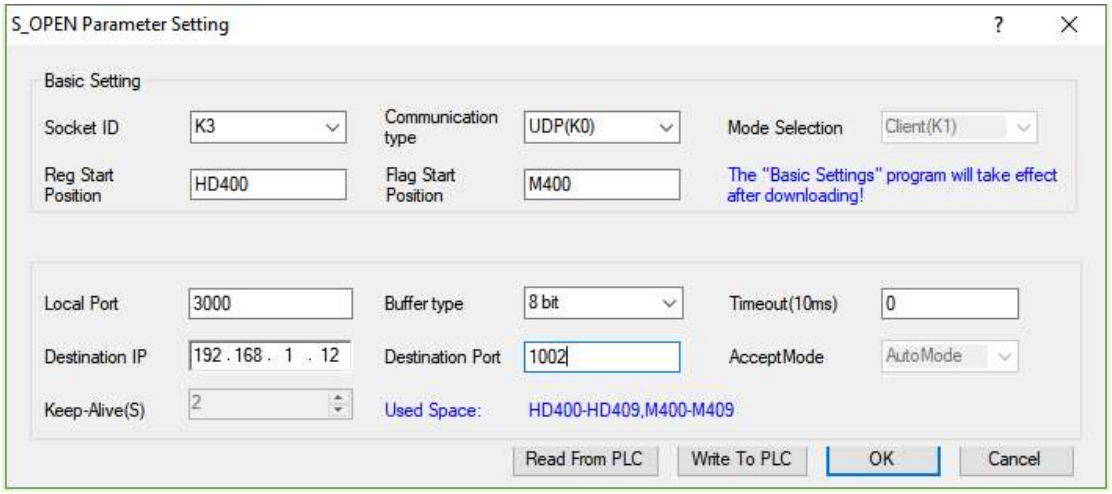
The configuration information for UDP socket S_OPEN is as follows



PLC 2 program:


SM0
PLS M30
Normally ON coil
SM1901
SM1902
M401
SM12
HM30
S_OPEN K3 K0 K1 HD400 M400
Initial network
Connect device
Socket 3 connected
100ms coil
Build connection
M401
M32
S_SEND K3 D4000 K400
Socket 3 connected
Socket 3 send data
M33
S_RCV K3 D4600 K550
Socket 3 receive data
M30
S_CLOSE K3
Close socket when power on
M409
Socket 3 error
M408
Socket 3 TCP error
SM1902
Connect device
HM30
Build connection

The configuration information for UDP socket S_OPEN is as follows



Example 2:

Through the following program, the PLC can automatically communicate with MODBUS-TCP server devices A and B after power on. The IP address of the PLC is 192.168.1.12, the IP address of device A is 192.168.1.6, the Modbus station number is 1, the IP address of device B is 192.168.1.14, and the Modbus station number is 1.

Note: As a server, ModbusTCP does not require writing communication instructions.

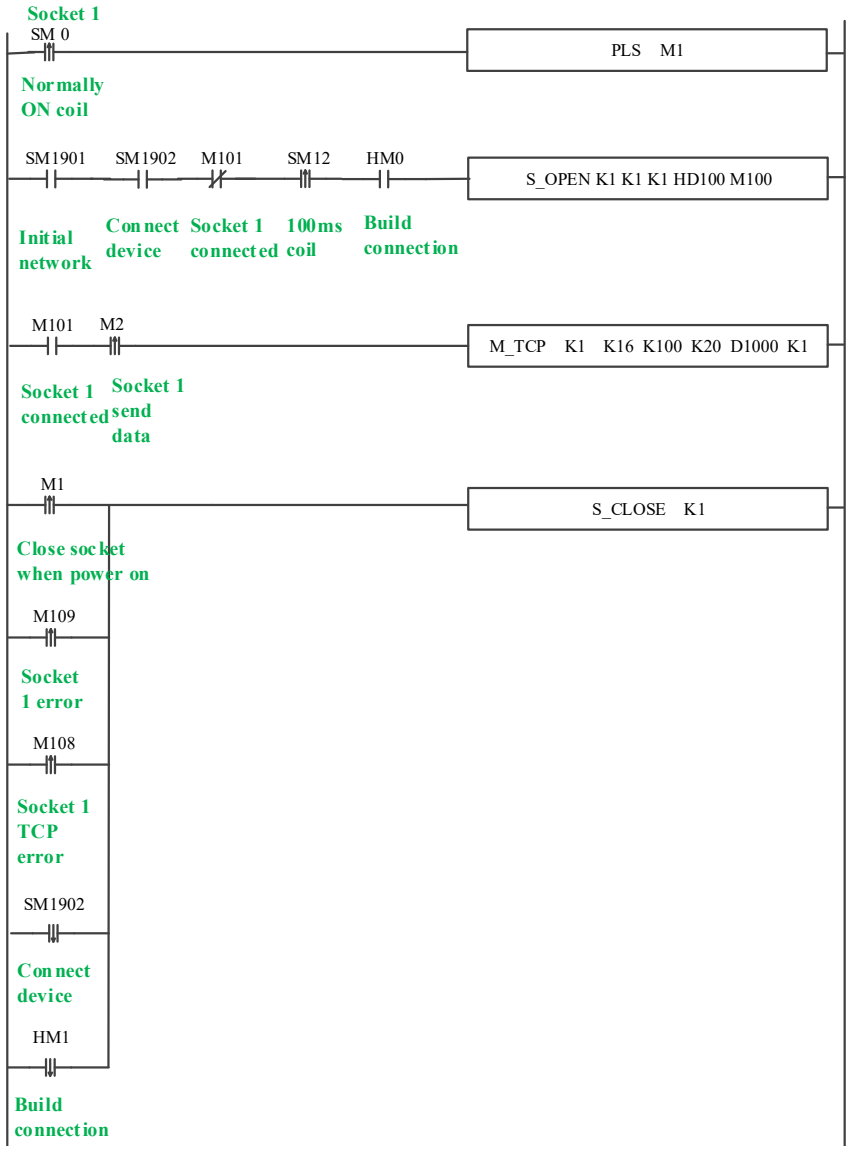
Due to the varying number of Ethernet ports in different series of PLCs, when using communication related coils SM1902 or SM1903, please pay attention to distinguishing which Ethernet port is connected to the PLC by the network cable (SM1902 is the symbol for connecting network devices, used in the first Ethernet port of a dual port model or in a single port model to connect to switches/routers/other network devices). SM1903 is a symbol for connecting network devices, used in dual port models where the second network port is connected to a switch/router/other network devices.

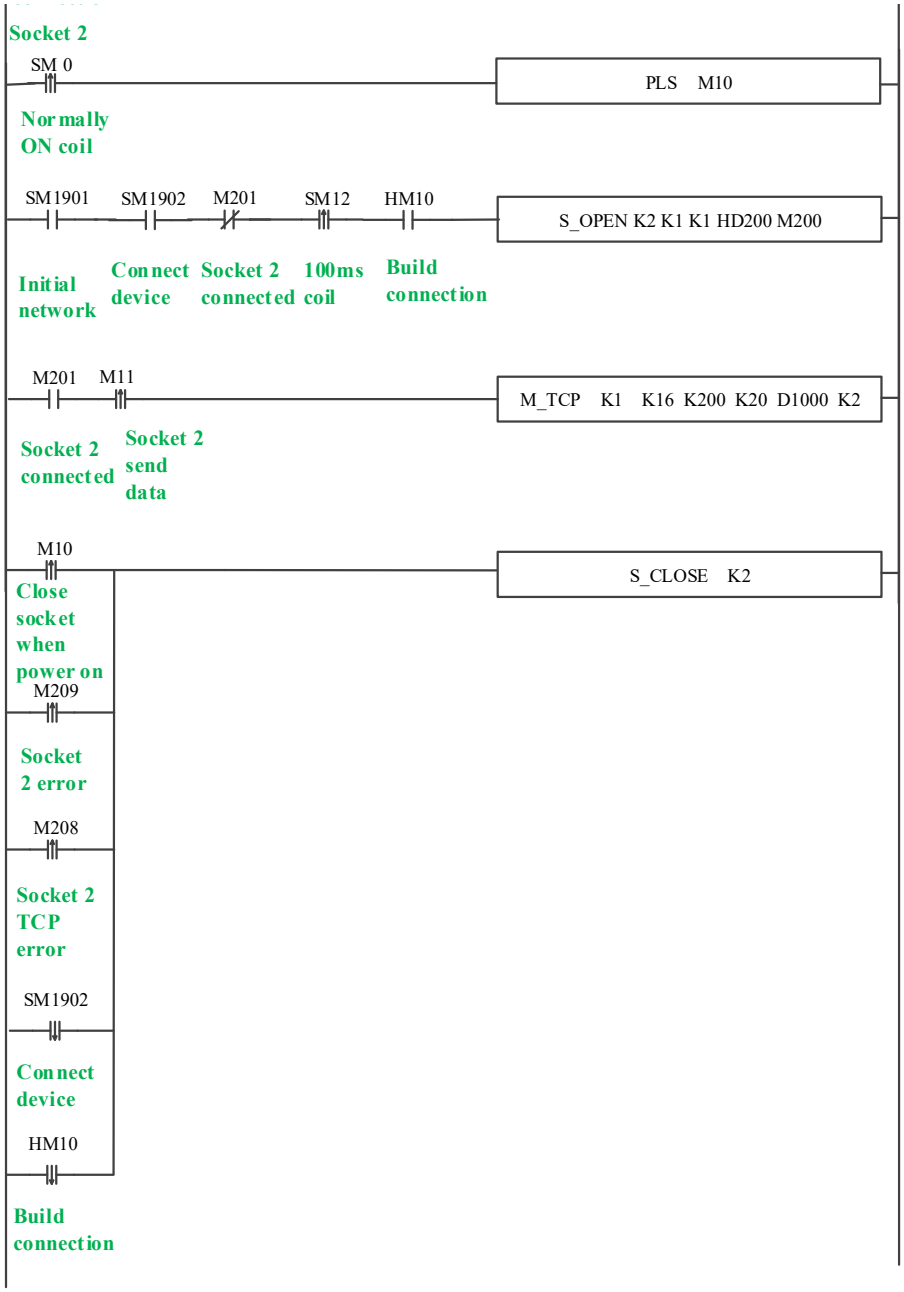
Program operation:

(1) After the PLC is powered on, it actively establishes a TCP connection to the TCP server service port 502 of device A as a TCP client and binds the socket ID to 1. After the connection is successfully established, the value of D1000-D1019 is written to device A's 4x100-4x119 every 1 second. When a TCP connection encounters an exception or the sender does not receive a response message within the set live time (where the live time is set to 2 seconds), the TCP connection is actively closed and rebuilt.

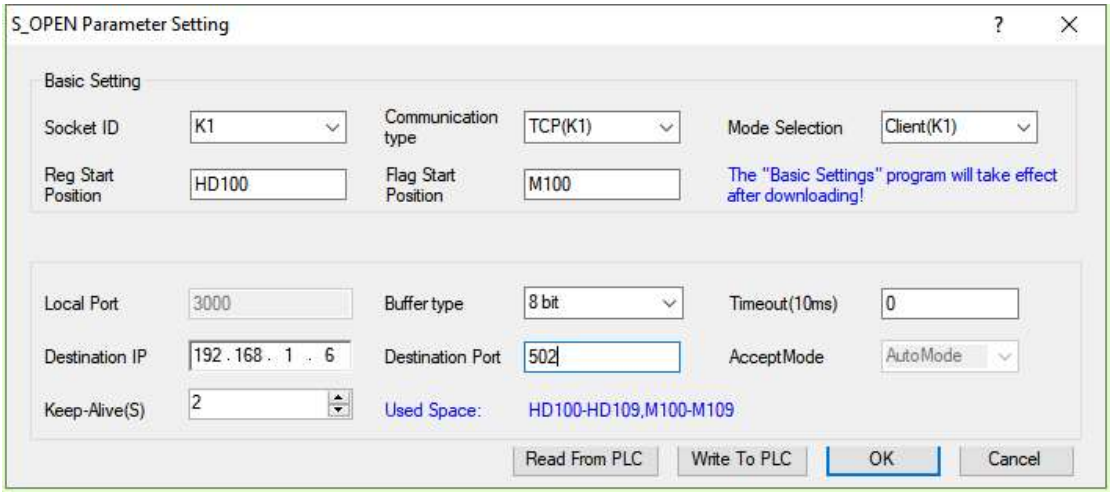
(2) After the PLC is powered on, it actively establishes a TCP connection to the TCP server service port 502 of device B as a TCP client and binds a socket ID of 2. After the connection is successfully established, the value of D1000-D1019 is written to device B's 4x200-4x219 every 1 second. When a TCP connection encounters an exception or the sender does not receive a response message within the set live time (where the live time is set to 2 seconds), the TCP connection is actively closed and rebuilt.

Program:

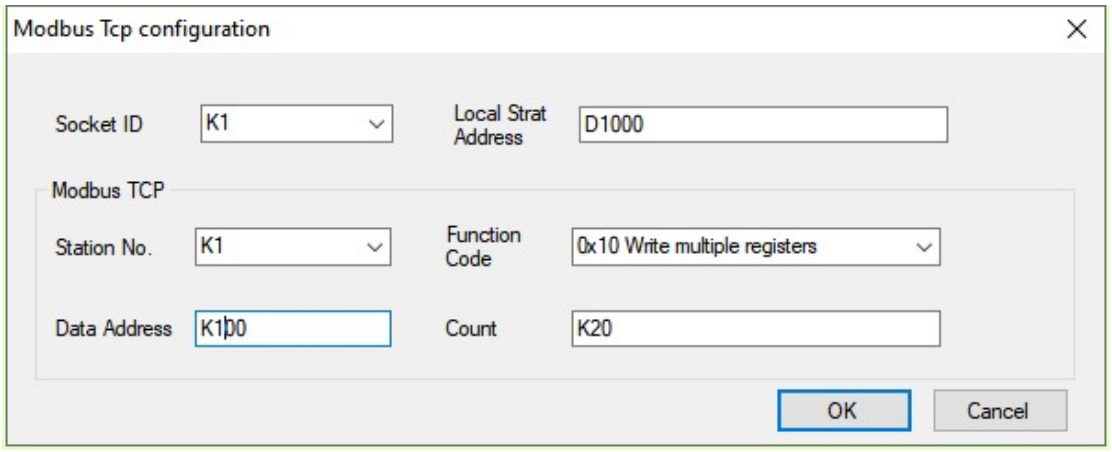

Socket 1
SM 0
PLS M1
Normally ON coil
SM1901
SM1902
M101
SM12
HM0
S_OPEN K1 K1 K1 HD100 M100
Initial network
Connect device
Socket 1 connected
100ms coil
Build connection
M101
M2
M_TCP K1 K16 K100 K20 D1000 K1
Socket 1 connected
Socket 1 send data
M1
S_CLOSE K1
Close socket when power on
M109
Socket 1 error
M108
Socket 1 TCP error
SM1902
Connect device
HM1
Build connection


Socket 2
SM 0
PLS M10
Normally ON coil
SM1901
SM1902
M201
SM12
HM10
S_OPEN K2 K1 K1 HD200 M200
Initial network
Connect device
Socket 2 connected
100ms coil
Build connection
M201
M11
M_TCP K1 K16 K200 K20 D1000 K2
Socket 2 connected
Socket 2 send data
M10
S_CLOSE K2
Close socket when power on
M209
Socket 2 error
M208
Socket 2 TCP error
SM1902
Connect device
HM10
Build connection

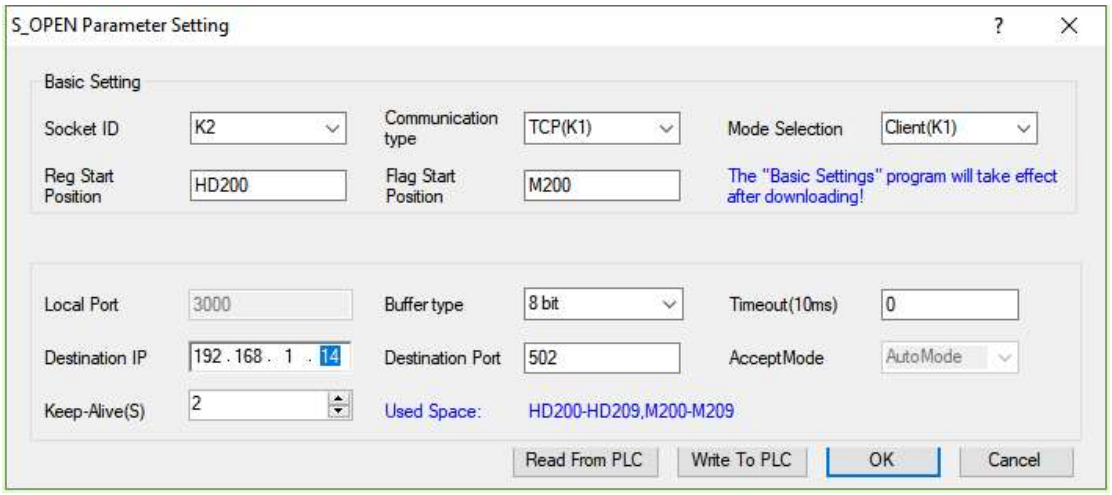
The configuration information for socket 1 S_OPEN is as follows:



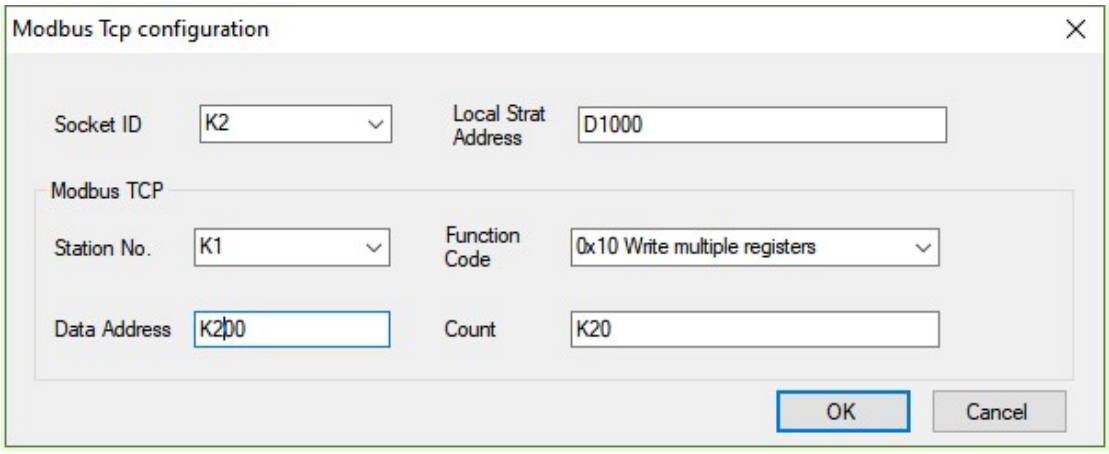
The configuration information for socket 1 M_TCP is as follows



The configuration information for socket 2 S_OPEN is as follows



The configuration information for socket 2 M_TCP is as follows



Example 3: By using the following program, the PLC can automatically create a UDP multicast communication task after being powered on. When there is an abnormality in the connection, it can actively close the UDP multicast connection and rebuild it. Implement one send multiple receive. The IP address of PLC 1 is 192.168.1.6, PLC 2 is 192.168.1.12, and PLC 3 is 192.168.1.14.

Due to the varying number of Ethernet ports in different series of PLCs, please distinguish which Ethernet port the network cable is connected to when using communication related coils SM1902 or SM1903. (SM1902 is the symbol for connecting network devices, used in the first network port of a dual port model or in a single port model to connect to a switch/router/other network device. SM1903 is the symbol for connecting network devices, used in the second network port of a dual port model to connect to a switch/router/other network device)

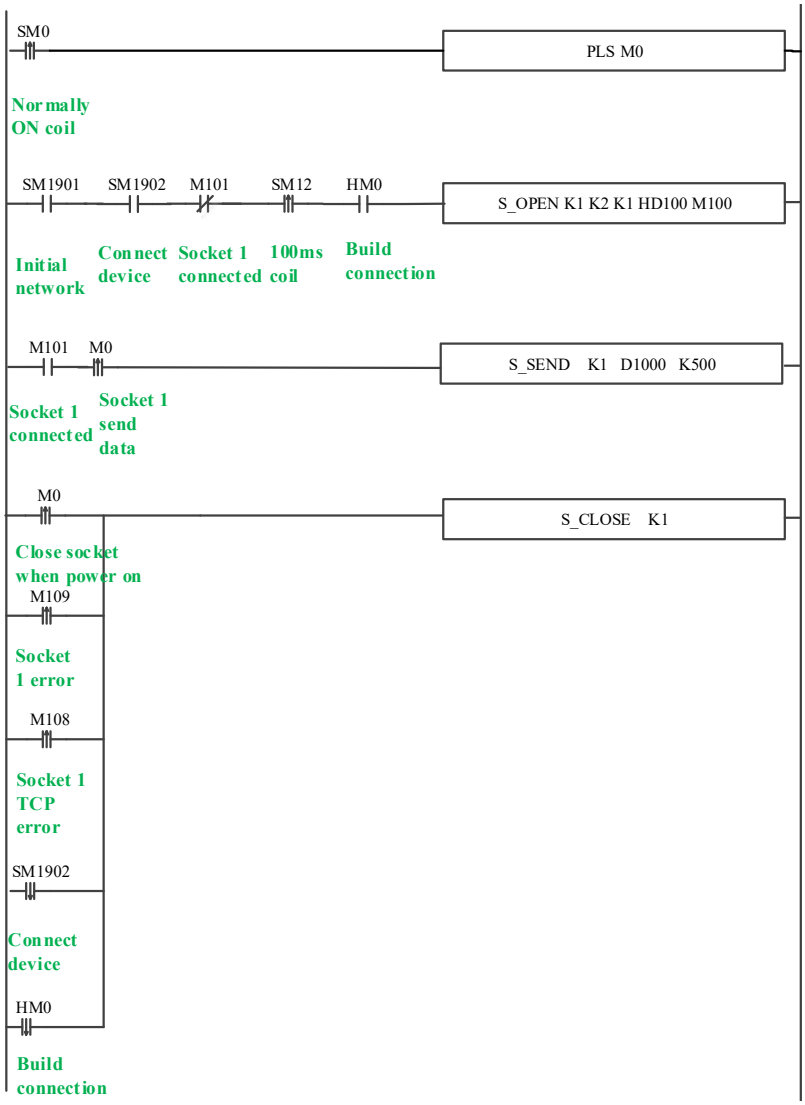
Program operation:

(1) After PLC1 is powered on, UDP multicast communication is used. The target IP is set to 230.0.0.0, the target port is 7000, and the socket ID is bound to 1. After establishing a successful connection, PLC 1 sends the low 8-bit of D1000-D1499 at a frequency of 1 second. PLCs 2 and 3 continuously receive data from PLC1 and store it in registers D1000-D1499.

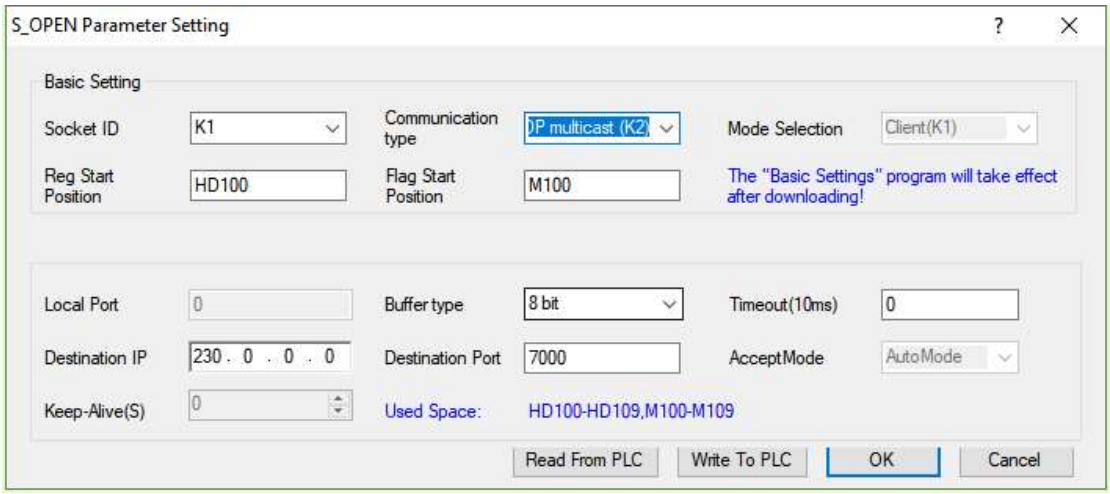
(2) After PLC2 is powered on, it uses UDP multicast communication, sets the target IP to 230.0.0.0, the target port to 7000, and binds the socket ID to 1. After establishing a successful connection, PLC 2 continuously receives data from PLC1 and stores it in the lower eight bits of registers D1000~D1499.

(3) After PLC3 is powered on, it uses UDP multicast communication, sets the target IP to 230.0.0.0, the target port to 7000, and binds the socket ID to 1. After establishing a successful connection, PLC 3 continuously receives data from PLC1 and stores it in the lower eight bits of registers D1000~D1499.

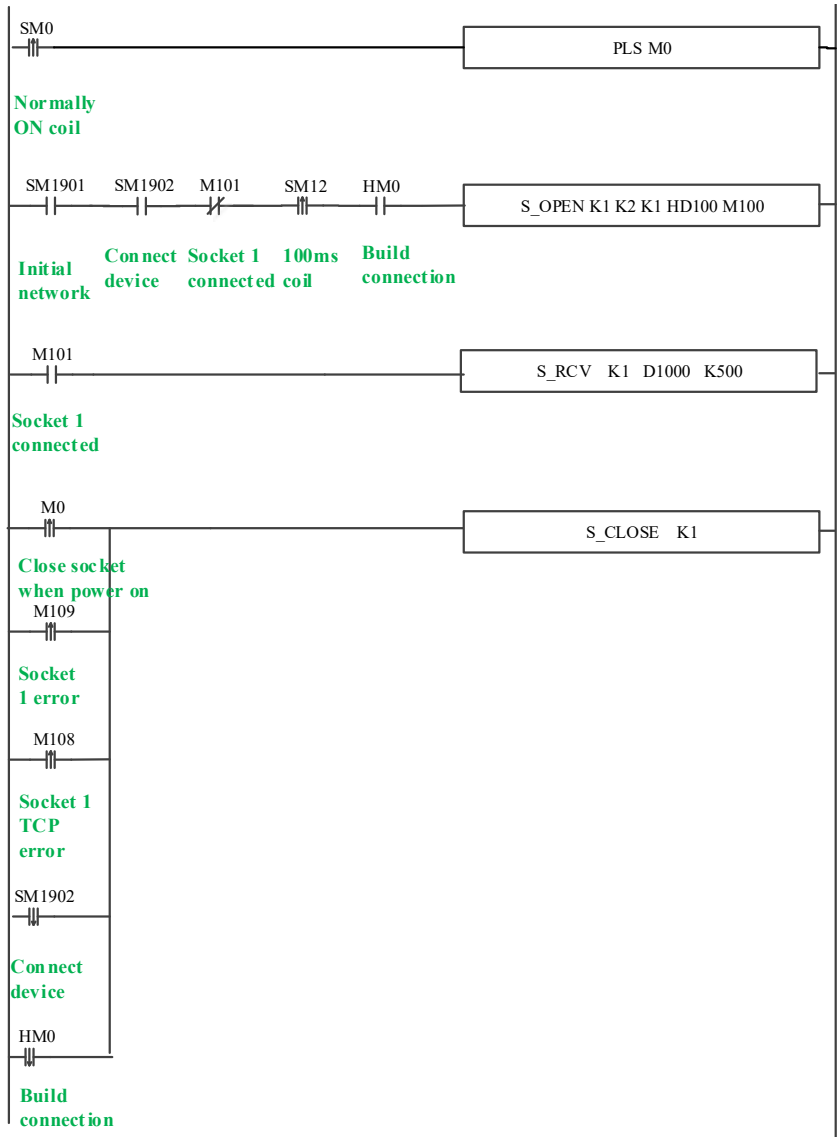
PLC1 program:



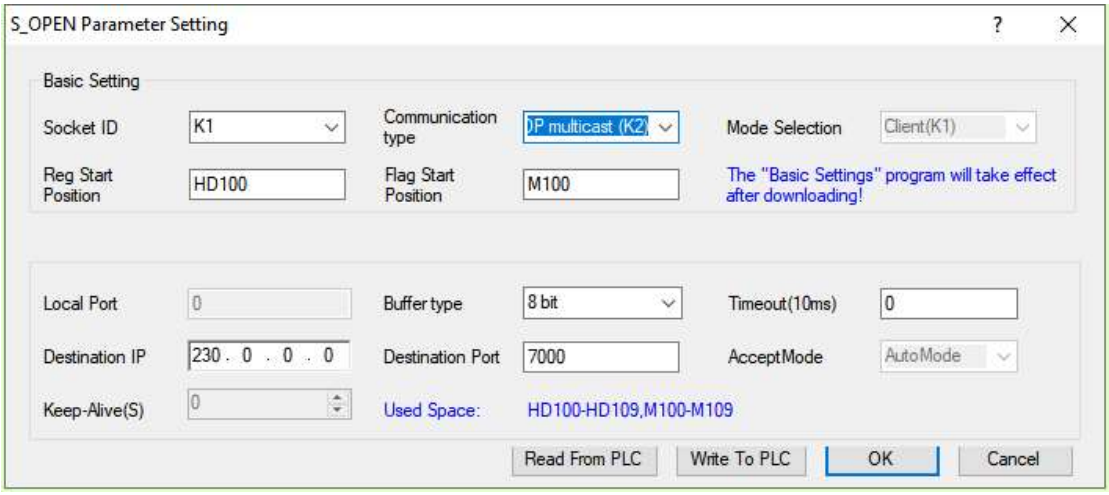
The UDP multicast S_OPEN parameter configuration is as follows:



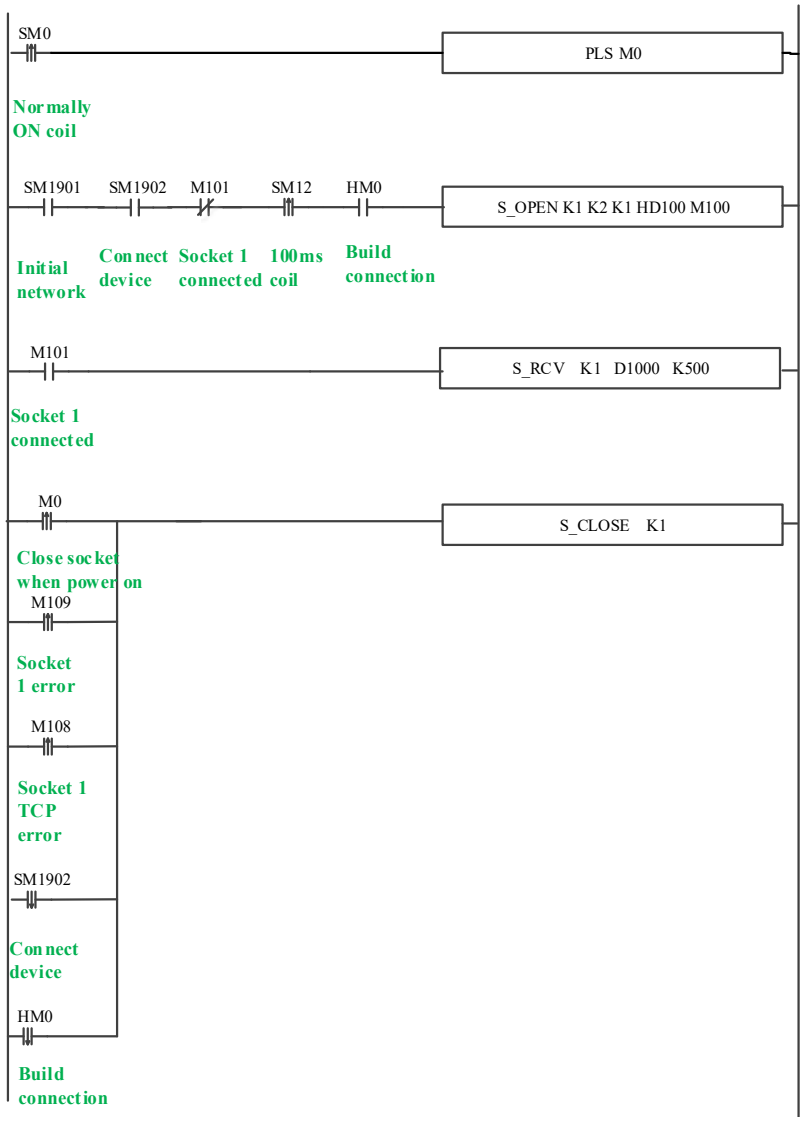
PLC2 program:



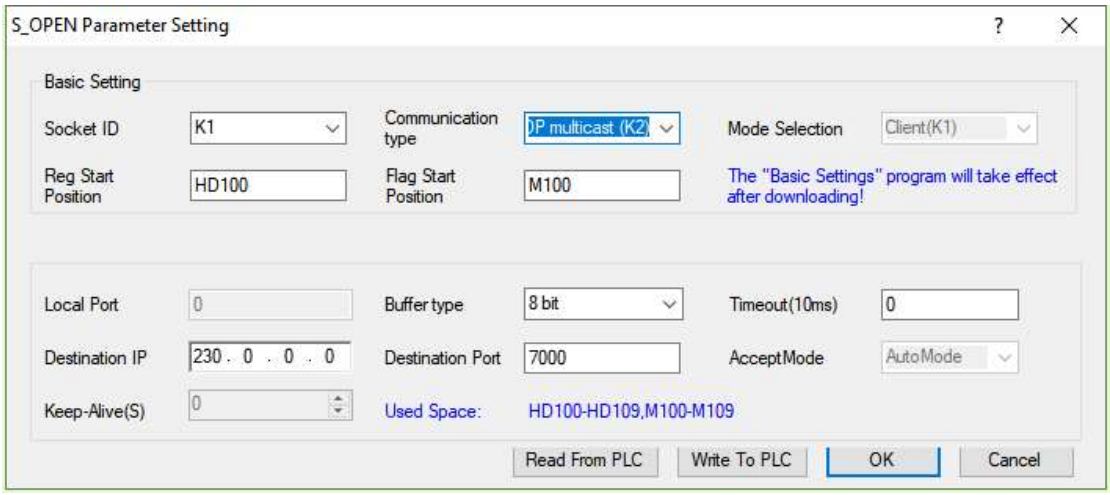
The UDP multicast S_OPEN parameter configuration is as follows:



PLC3 program:



The UDP multicast S_OPEN parameter configuration is as follows:



4-2．Read write communication port parameters

To ensure the normal implementation of Ethernet communication, it is recommended to use communication port parameter read/write instructions when writing communication programs. Firstly, by calling the communication parameter read instruction, the corresponding parameters on the communication port are read into the specified register group. The user then modifies the corresponding values in the register group as needed, and then writes the modified values of the register group to the corresponding communication port configuration through the communication parameter write instruction.

4-2-1．Read serial port parameters [CFGCR]

(1) Overview

Read the serial port parameters into the specified registers in the local machine.

Read the serial port parameters [CFGCR]			
16-bit instruction	CFGCR	32-bit instruction	-
Execution condition	Normally ON/OFF coil, edge triggering	Suitable model	XD, XL, XG
Firmware	-	Software	V3.4 and up

(2) Operand

Operand	Function	Type
D	Specify the first address of the local register	16 bits, BIN
S1	Specify the number of serial port parameters to be read	16 bits, BIN
S2	Specify the serial port number to be read	16 bits, BIN

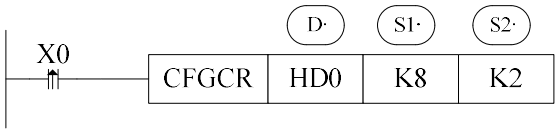
(3) Suitable soft component

operand	Word											Bit						
	System								Constant	Module		System						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
D	●																	
S1	●	●							●									
S2	●								K									

Note: D represents D and HD; TD represents TD and HTD; CD represents CD, HCD, HSCD, HSD; DM stands for DM and DHM; DS stands for DS and DHS.

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

(4) Function and action



- Operand S1: The number of registers occupied by reading serial port parameters, usually 8

(Ethernet port parameters are 9).

- Operand S2: Serial port number range: K0~K5. K0: COM0, K1: COM1, K2: COM2, or COM2-RS232 or COM2-RS485, K3: COM3, K4: COM4, K5: COM5, K9: Ethernet port.
- Read the 8 parameters of serial port 2 into HD0~HD7. The specific parameter names and definitions can be found in sections 4-2-4.

4-2-2. Write serial port parameters [CFGCW]

(1) Instruction overview

Write the values from the specified registers in the local machine to the specified serial port.

Write serial port parameters [CFGCW]			
16-bit instruction	CFGCW	32-bit instruction	-
Execution condition	Normally ON/OFF coil, edge triggering	Suitable model	XD, XL, XG
Firmware	-	Software	V3.4 and up

(2) Operand

Operand	Function	Type
S1	Specify the first address of the local register	16 bits, BIN
S2	Specify the number of serial port parameters to be written	16 bits, BIN
S3	Specify the serial port number for writing	16 bits, BIN

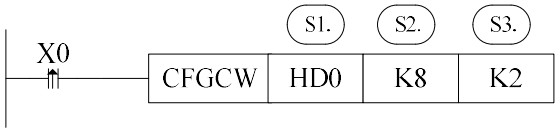
(3) Suitable soft component

Operand	Word											Bit						
	System								constant	module		system						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
S1	●																	
S2	●	●							●									
S3	●								K									

Note: D represents D and HD; TD represents TD and HTD; CD represents CD, HCD, HSCD, HSD; DM stands for DM and DHM; DS stands for DS and DHS.

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

(4) Function and action



- Operand S2: The number of registers occupied by writing serial port parameters, usually 8 (Ethernet port parameters are 9).

- Operand S3: Serial port number range: K0~K5. K0: COM0, K1: COM1, K2: COM2, or COM2-RS232 or COM2-RS485, K3: COM3, K4: COM4, K5: COM5, K9: Ethernet port.
- Write the values from HD0 to HD7 into the parameters of serial port 2. The specific parameter names and definitions can be found in sections 4-2-4.
- After writing, the PLC needs to power on again to make the parameters take effect.

4-2-3. Set the IP address [IPSET]

(1) Instruction overview

Set the IP address of the local device.

Set IP address [IPSET]			
16-bit instruction	IPSET	32-bit instruction	-
Execution condition	Edge triggering	Suitable model	XD3E, XD5E, XDME, XDH, XG, XL5E, XL5N, XLME, XL5H, XLH
Firmware	V3.5.3b and up	Software	V3.5.3 and up

(2) Operand

Operand	Function	Type
S0	Specify local register address	16-bit integer
S1	Specify the register numbers (K4, K12)	16-bit integer
S2	Specify the local serial port no. (K9)	16-bit integer

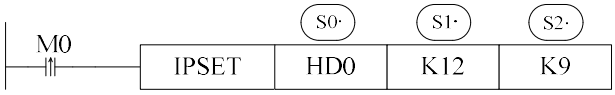
(3) Suitable soft component

Operand	Word soft component											Bit soft component						
	System								Constant	Module		System						
	D	FD	TD	CD	DX	DY	DM	DS	K/H	ID	QD	X	Y	M	S	T	C	Dn.m
D0	●								●									
D1	●								●									
D2	●								●									

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

(4) Function and action

Instruction format



- Write the network parameters in HD0-HD11 to the Ethernet port of PLC

Address	Function	Example	Data format
HD0	IP	192	Decimal
HD1		168	Decimal
HD2		51	Decimal
HD3		103	Decimal
HD4	Subnet mask	255	Decimal
HD5		255	Decimal
HD6		255	Decimal
HD7		0	Decimal
HD8	Default gateway	192	Decimal
HD9		168	Decimal
HD10		51	Decimal
HD11		1	Decimal

- S0: Specifies the first address of the local register.
- S1: The value is K4 or K12.

  K4: Write only the IP address, for example, IP address: 192.168.51.103.

  K12: Write the IP address, subnet mask, default gateway to the Ethernet port of the PLC;

  For example, IP address: 192.168.51.103

  Subnet mask: 255.255.255.0

  Default gateway: 192.168.51.1
- S2: the Ethernet port parameters of the PLC are fixed to K9.

Note:

(1) After the parameters are written, the PLC needs to be powered on again to take effect;

(2) When the current IP address is automatically obtained, executing the IPSET command will change the IP address to a fixed IP address;

(3) Set the IP to 0, you can change the fixed IP to automatically obtain IP.

- Register address of Ethernet port parameters

Address	Function	Type	Data format
SD1930	IP	Read only	Decimal
SD1931		Read only	Decimal
SD1932		Read only	Decimal
SD1933		Read only	Decimal
SD1934	subnet mask	Read only	Decimal
SD1935		Read only	Decimal
SD1936		Read only	Decimal
SD1937		Read only	Decimal
SD1938	default gateway	Read only	Decimal
SD1939		Read only	Decimal
SD1940		Read only	Decimal
SD1941		Read only	Decimal

Note: The Ethernet parameter registers are read-only. To change the IP address, you must use the IPSET instruction.

4-2-4. Serial port parameter name and setting

Assuming that HD0~HD14 correspond to serial port parameters, the parameter names and settings represented by each register are shown in the table below:

Parameter address	Parameter name and setting				
	MODBUS communication (HD0=1)	Free format communication (HD0=2)	X-NET communication		Ethernet communication (HD0=3)
			OMMS (HD0=3)	TBN (HD0=3)	
HD0	Network type: 1: MODBUS 2: free format 3: X-NET 4: MODBU-TCP				
HD1	MODBUS station no. 1~254	Baud rate Refer to table 1	Network number 0~32767	Network number 0~32767	Network number IP high two bytes
HD2	Transmission mode 0: RTU 128: ASCII	Frame format Refer to table 2	Station no. 0~100	Station no. 0~100	Station no. IP low two bytes
HD3	Baud rate Refer to table 1	Free property bit7: 1: Has a starting character 0: no starting character bit6： 1: There is a terminator 0: No terminator	Physical layer type 0: PHY_RS485 1: PHY_SOF (Unidirectional fiber ring network) 2: PHY_OFPP (Fiber optic dot network) 3: PHY_RS232 4: PHY_RS422 5: PHY_TTL (TTL level network)		
HD4	frame format refer to table 2	Starting character	Link layer type 0: TBN 1: HDN 2: CCN 3: PPFD 4: PPU 5: Ethernet		
HD5	retry count 0~5	terminator	OMMS Properties 128: Supports periodic communication, otherwise not supported	Baud rate Refer to table 1	Subnet Mask High Two Bytes
HD6	Response overtime 0~65535	Frame timeout time 0~255	OMMS baud rate Refer to table 1	token rotation time 1~60000 (ms)	Subnet mask low two bytes

HD7	Delay before sending 0~255	Response timeout 0~65535 (0 is infinite waiting)	OMMS Slave List Each bit of each byte in the array represents whether the slave station can be accessed (valid for the master station, i.e. the station number is 1)	Maximum number of stations 1~100	Gateway address two bytes higher
HD8	-	-	-	-	Gateway address two bytes lower

[Note]: The table does not include "buffer bits" in free format communication mode, so "buffer bits" cannot be read and written using CFGCR and CFGCW instructions, but can be read and written using MOV instructions. The address of "buffer bits" is shown in Appendix 3.

Table 1: baud rate

Value	Baud rate	Value	Baud rate	Value	Baud rate	Value	Baud rate
1	300 bps	7	19200 bps	13	256000 bps	19	1000000 bps
2	600 bps	8	28800 bps	14	288000 bps	20	1200000 bps
3	1200 bps	9	38400 bps	15	384000 bps	21	1500000 bps
4	2400 bps	10	57600 bps	16	512000 bps	22	2400000 bps
5	4800 bps	11	115200 bps	17	576000 bps	23	3000000 bps
6	9600 bps	12	192000 bps	18	768000 bps		

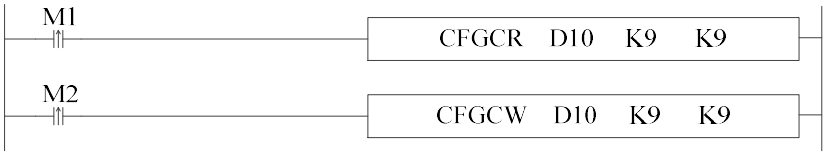
Table 2: frame format

Stop bit		Parity bit			Data bit length		
Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
00: 1 bit 01: 1.5 bit 10: 2 bits		000: none 001: odd 010: even 011: vacant 100: Mask			000: 5 bits 001: 6 bits 010: 7 bits 011: 8 bits 100: 9 bits		

4-2-5. Read write port parameter application

Example 1: By using the parameter read instruction [CFGCR] and write instruction [CFGCW], the network parameters of the PLC are read into 9 consecutive registers D10~D18. After modification, the network parameters of the 9 consecutive registers D10~D18 are written into the serial port settings of the PLC.

PLC program:



PLC1-自由监控

监控 | 添加 修改 删除 删除全部 | 上移 下移 置顶 置底

寄存器	监控值	字长	进制	注释
D10	0003	单字	16进制	
D11	C0A8	单字	16进制	IP地址前两位，C0对应K192，A8对应K168
D12	003C	单字	16进制	IP地址后两位，00对应K0，3C对应K60
D13	0000	单字	16进制	
D14	0005	单字	16进制	
D15	FFFF	单字	16进制	子网掩码前两位，分别对应255.255
D16	FF00	单字	16进制	子网掩码后两位，分别对应255.0
D17	C0A8	单字	16进制	默认网关前两位，分别对应192.168
D18	0001	单字	16进制	默认网关后两位，分别对应0.1

D11: IP address first two bits, C0 corresponds to K192, A8 corresponds to K168

D12: IP address last two bits, 00 corresponds to K0, 3C corresponds to K60

D15: subnet mask first two bits, correspond to 255.255

D16: subnet mask last two bits, correspond to 255.0

D17: default gateway first two bits, correspond to 192.168

D18: default gateway last two bits, correspond to 0.1

When M1 is set, it triggers the network parameter reading of the PLC. After modifying the network parameters, set ON M2 to write the modified network parameters into the PLC. After writing, the PLC will power off and then power on to make the serial port parameters effective.

4-3. Ethernet communication flag and register

Communication registers:

Address	Format	Function	Explanation
SD1905	Hex	IP net number	The first two bytes of IP address
SD1906	Hex	IP station no.	The last two bytes of IP address
SD1907	Hex	Subnet mask	The first two bytes of subnet mask
SD1908	Hex		The last two bytes of subnet mask
SD1909	Hex	Defaulted gateway	The first two bytes of defaulted gateway
SD1910	Hex		The last two bytes of defaulted gateway
SD1920	Decimal	Abnormal socket ID	Abnormal socket ID, only be effective when the connection is not created
SD1921	Decimal	Error code	1: the socket ID is over the range 2: not registered socket ID sends a communication request 3: communication type error, out of the range 0---TCP 1---UDP 4: TCP connection quantity out of the range, max is 32 5: UDP connection quantity out of the range, max is 32 6: communication mode error, out of the range, 0---Server 1---Client 7: Abnormal flag bit (usually abnormal flag bit in XDPPRO software) 8: Target port error (check if the target port setting is 0) 9: Local port error (check if the local port setting is 0) 10: Communication busy
SD1930	Decimal	IP address	IP address first byte
SD1931	Decimal		IP address second byte
SD1932	Decimal		IP address third byte
SD1933	Decimal		IP address fourth byte
SD1934	Decimal	Subnet mask	Subnet mask first byte
SD1935	Decimal		Subnet mask second byte
SD1936	Decimal		Subnet mask third byte
SD1937	Decimal		Subnet mask fourth byte
SD1938	Decimal	Default gateway	Default gateway first byte
SD1939	Decimal		Default gateway second byte
SD1940	Decimal		Default gateway third byte
SD1941	Decimal		Default gateway fourth byte

Communication coils:

Address	Function	Explanation
SM1900	Log in remote server successfully flag	Set on when the remote connection succeeded
SM1901	Ethernet initialization completed flag	MODBUS TCP Server/TCP IP/ XNET
SM1902	Connect net device flag	First network port of dual network port models or single network port model connect to swither/router/ other net devices
SM1903	Connect net device flag	Second network port of dual network port models connect to swither/router/ other net devices
SM1921	Ethernet error flag	Set on when the error in any of the SD1921 generated

4-4. Ethernet communication error list

Error code	Explaination
0	Communication normal
1	The socket which is needed to OPEN already created connection
2	Return error when creating the socket
3	Bind appointed port failed
4	TCPServerAccept failed
5	TCPClientConnect failed
6	When calling Send, Recv, Clos, the specified socket hasn't created connection
7	Call Send return failed
8	Call Recv return failed
10	The specified sending data length is out of the range
11	The specified receiving data length is out of the range
20	When UDP communicating, received data is not from specified IP
21	When UDP communicating, received data is not from specified port
30	Actual received data length is larger than specified length
31	Actual received data length is less than specified length
32	Received data length error (non specified length)
33	Sending data length error
40	Receive timeout
50	Specified target port error, MODBUS TCP is not port 502, The using port is out of range (1~60000)
51	Port reuse (indicating that the port is used for both TCP and Modbus TCP)
60	Socket communication busy
61	No receiving task when receiving data (usually when the PLC receives data without calling S_RCV)
62	Parameter setting error

63	Remote shutdown
64	Socket type error
65	Insufficient memory for task request (task request too fast)
66	Incorrect use of IP address
67	Port usage error
68	Sending blocking error
70	Socket index error
71	Socket connection status error
100	Receive error
101	Receive timeout
182	Station no. 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 (Erase Flash)

5. EtherNet/IP communication

EtherNet/IP using requirements			
Suitable model	XDH, XLH, XG2		
Firmware	V3.7.4 and higher	Software	V3.7.17a and up

5-1. Ethernet/IP overview

EtherNet/IP (Ethernet/Industrial Protocol) uses standard IEEE 802.3 technology and is an industrial automation communication protocol based on Ethernet. EtherNet/IP uses standard Ethernet and TCP/IP technology to transmit CIP communication packets.
The EtherNet/IP protocol mainly has the following characteristics: based on Ethernet technology, supporting TCP/IP and UDP/IP protocols, providing explicit and implicit messages, supporting multiple data formats, and supporting device description files.

Based on Ethernet technology

The EtherNet/IP protocol uses Ethernet technology for communication, which has the advantages of high speed, wide area, and low cost of Ethernet, and can achieve real-time control and data communication in the field of industrial automation.

Support TCP/IP and UDP/IP protocols

The EtherNet/IP protocol supports TCP/IP and UDP/IP protocols, and different protocols can be selected for communication according to the needs of the application. The TCP/IP protocol ensures the reliability and integrity of data transmission, and is suitable for control and communication situations that require high reliability; The UDP/IP protocol is suitable for broadcasting and multicast communication scenarios, with the advantages of low latency and high efficiency.

Provide explicit and implicit messages

The EtherNet/IP protocol provides two communication methods: explicit and implicit messaging. Explicit messaging uses TCP/IP protocol for communication, with flexible message formats and expandable functionality; Implicit messaging uses UDP/IP protocol for communication, suitable for broadcasting and multicast scenarios, with the advantages of simple message format and low latency.

Supports multiple data formats

The EtherNet/IP protocol supports multiple data formats, including bits, bytes, integers, floating-point numbers, etc., and can meet the data transmission needs of different application scenarios.

Supporting device description files

The EtherNet/IP protocol supports device description files, which can identify and configure devices connected to the network, improving device interoperability and manageability.
In summary, EtherNet/IP protocol is a modern industrial automation communication protocol with flexible, efficient, and reliable characteristics, widely used in industrial control, intelligent manufacturing, robotics and other fields.

5-2．Ethernet/IP nouns overview

Abbreviation	Explanation
IEEE 802.3	A standard specification in the field of communication technology, also known as Ethernet protocol; This standard specification defines the transmission method and format of data in Ethernet networks
EIP	Ethernet/IP, Industrial Ethernet
CIP	Common Industrial Protocol. Used to describe various industrial automation protocols
EipScanner	EIP master station, referred to as scanner in Etehrnet/IP
EipAdapter	EIP slave station, referred to as adapter in Ethernet/IP
EDS	Electronic Data Sheets, used to describe Ethernet/IP device
RPI	Request/Response Interval, also known as communication cycle
PPS	Packet Per Second, the number of data packets transmitted per second

5-3．Ethernet/IP communication specification

5-3-1．Implicit function specification

Scanner (Main Station)	Communication specification parameters
Suitable model	XDH, XLH, XG2
Slave station connection numbers	≤128
Number of shared connections	Adapter+Scanner≤256 pieces
Ethernet Maximum communication volume	4000pps
Data length	1~724 words (Note: 1 word=2 bytes)
RPI	1ms~65535ms
Adapter (Slave Station)	Communication specification parameters
Suitable model	XDH, XLH, XG2
Label name	≤64 bytes
Instance ID	100~199
Mapping first address	Support D/HD registers
Data length	1~724 word (Note: 1 word=2 bytes)
Number of shared connections	Adapter+Scanner ≤256 pieces
Allow configuration items	Input (O-->T) + Output (T-->O) = 256 pieces
RPI	1ms~65535ms

5-3-2. Explicit function specification

Client parameter	Communication specification parameters
Suitable model	XDH, XLH, XG2
Name	≤64 bytes
Allow configuration items	32 slave stations share 3000 instructions
Maximum byte length of communication data packet	504 bytes (CIP packet head+CIP packet data)
Timeout time	10~65535ms
Number of retransmissions	1~15
Enable control	Only support M0~M199999, HM0~HM19999
Connection flag bit	Only support M0~M199999, HM0~HM19999

Server parameter	Communication specification parameters
Suitable model	XDH, XLH, XG2
Number of connected clients	≤16
Number of configurable labels	≤5000
Label name	≤64 bytes
Maximum byte length of communication data packet	504 bytes (CIP packet head+CIP packet data)

5-3-3．Client and server support variable types

Client variable type	Server variable types	Data length
-	BIT	Bit (8-bit)
BOOL	BOOL	Bool (8-bit)
SINT	SINT	Short integer (8-bit)
USINT	USINT	Unsigned short integer (8-bit)
INT	INT	Integer (16-bit)
UINT	UINT	Unsigned double integer (16-bit)
DINT	DINT	Double integer (32-bit)
UDINT	UDINT	Unsigned long integer (32-bit)
LINT	LINT	Long integer (64-bit)
ULINT	ULINT	Unsigned long integer (64-bit)
REAL	REAL	Real (32-bit)
LREAL	LREAL	Long real (64-bit)
BYTE	BYTE	A bit string with a length of 8 (8-bit)
WORD	WORD	A bit string with a length of 16 (16-bit)
DWORD	DWORD	A bit string with a length of 32 (32-bit)
LWORD	LWORD	A bit string with a length of 64 (64-bit)

5-4．Ethernet/IP explicit/implicit communication

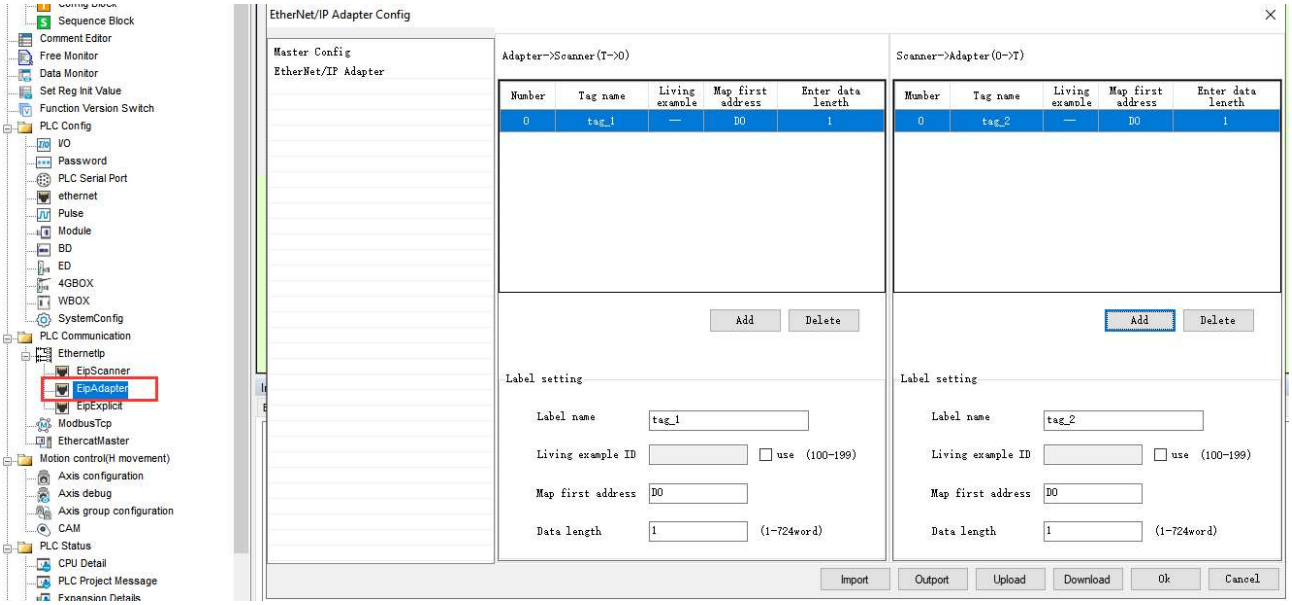
In the Ethernet/IP protocol, there are two different data transfer methods between devices or between devices and multiple devices, namely implicit and explicit functions. Their functions and usage methods are not exactly the same. Below, we will provide corresponding functional introductions for these two communication methods

5-4-1．Implicit function

The implicit function in the Ethernet/IP protocol refers to the method of data transmission through I/O data tables, which is usually used in real-time control and monitoring applications. In implicit functionality, two important components need to be used: a scanner and an adapter.

5-4-1-1. Adapter configuration

Before conducting network data exchange, the device needs to configure the address and length of the implicit message to be transmitted in the corresponding adapter configuration interface. The label setting and signature must be configured, and the instance ID is an optional configuration. The operation method is as follows:



① Double click to enter Ethernet/IP adapter configuration interface.

② Adapter—>Scanner(T->O) data configuration area:

The data configured in this configuration area is in the T ->O direction, and the data transmission direction is from the adapter to the scanner.

Add	Add one piece of Adapter—>Scanner(T->O) data configuration.
Delete	Delete the selected Adapter—>Scanner(T->O) data configuration.
Label name	As the target end, use the corresponding configured label name as the connection and respond to the connection establishment request. The label name can support a maximum of 64 bytes
Instance ID*1	As the target end, respond to the connection request by using the corresponding configured instance ID as the connection path. The optional range of this instance ID is 100-199;
Map first address*2	The starting address for data transmission in this connection;
Data length*3	The number of registers connected for data transmission.

Note:

※ 1: The label name must be set, and the instance ID is an optional configuration;

※ 2: The starting address currently supports two register types, D and HD;

※ 3: When establishing a connection with the corresponding label or instance ID, it is important to note that the data length configured by the adapter should be consistent with the data length configured by the scanner to avoid communication anomalies when establishing the corresponding connection.

③ Scanner—> Adapter(O->T) data configuration area.

The data configured in this configuration area is in the O ->T direction, and the data transmission direction is from the scanner to the adapter for data transmission.

Add	Add one piece of Scanner—> Adapter(O->T) data configuration
Delete	Delete the selected Scanner—> Adapter(O->T) data configuration
Label name	As the target end, use the corresponding configured label name as the connection and respond to the connection establishment request. The label name can support a maximum of 64 bytes
Instance ID*1	As the target end, respond to the connection request by using the corresponding configured instance ID as the connection path. The optional range of this instance ID is 100-199;
Map first address*2	The starting address for data transmission in this connection

Data length*3	The number of registers connected for data transmission
Import	Import the configured information into the current configuration interface in the form of an XML file
Export	Export the configured information in the form of an XML file
Upload	Upload the configuration information downloaded to the PLC to the current configuration interface, and the uploaded configuration information will overwrite the existing configuration information on the current interface
Download	Download the configuration information of the current configuration interface to the PLC. The downloaded configuration information will overwrite the original configuration information in the PLC and take effect in real time with the new configuration information
Ok	Click OK to save the configuration information for the current page
Cancel	Click to cancel the configuration information for the current page and discard it

Note:

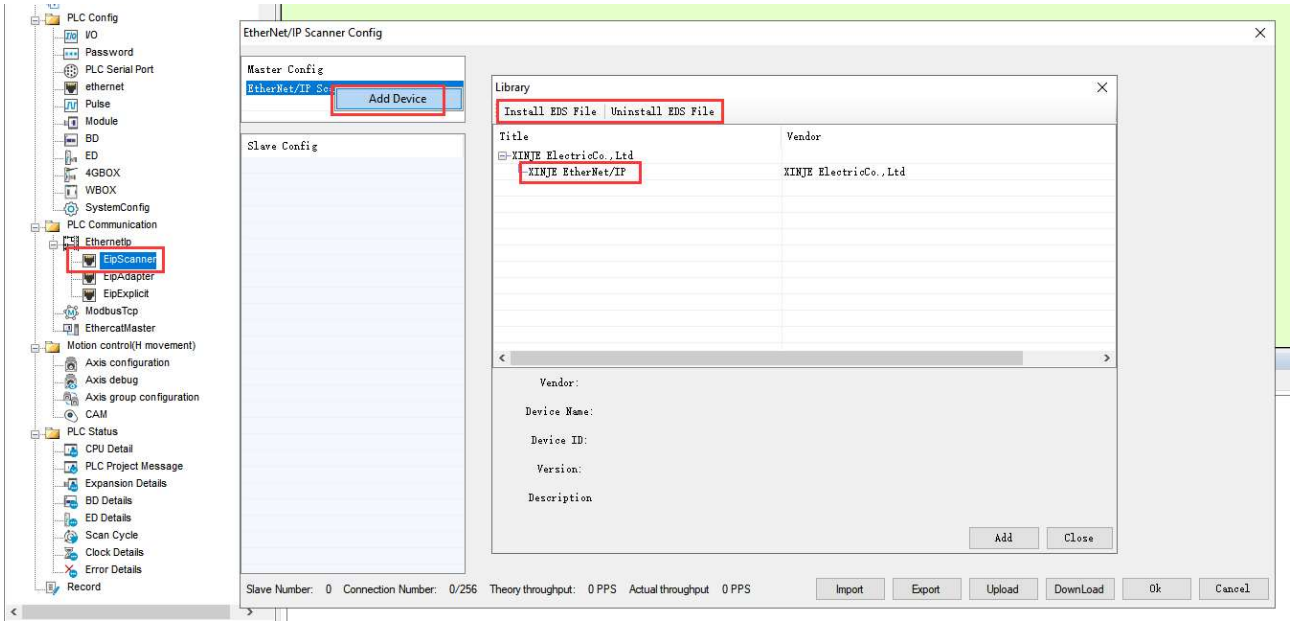
※ 1: The label name must be configured, and the instance ID is an optional configuration;

※ 2: The starting address currently supports two register types, D and HD;

※ 3: When establishing a connection with the corresponding tag or instance ID, it is important to note that the data length configured by the adapter should be consistent with the data length configured by the scanner to avoid communication anomalies when establishing the corresponding connection.

5-4-1-2. Scanner configuration

1. Load EDS file, add slave device.



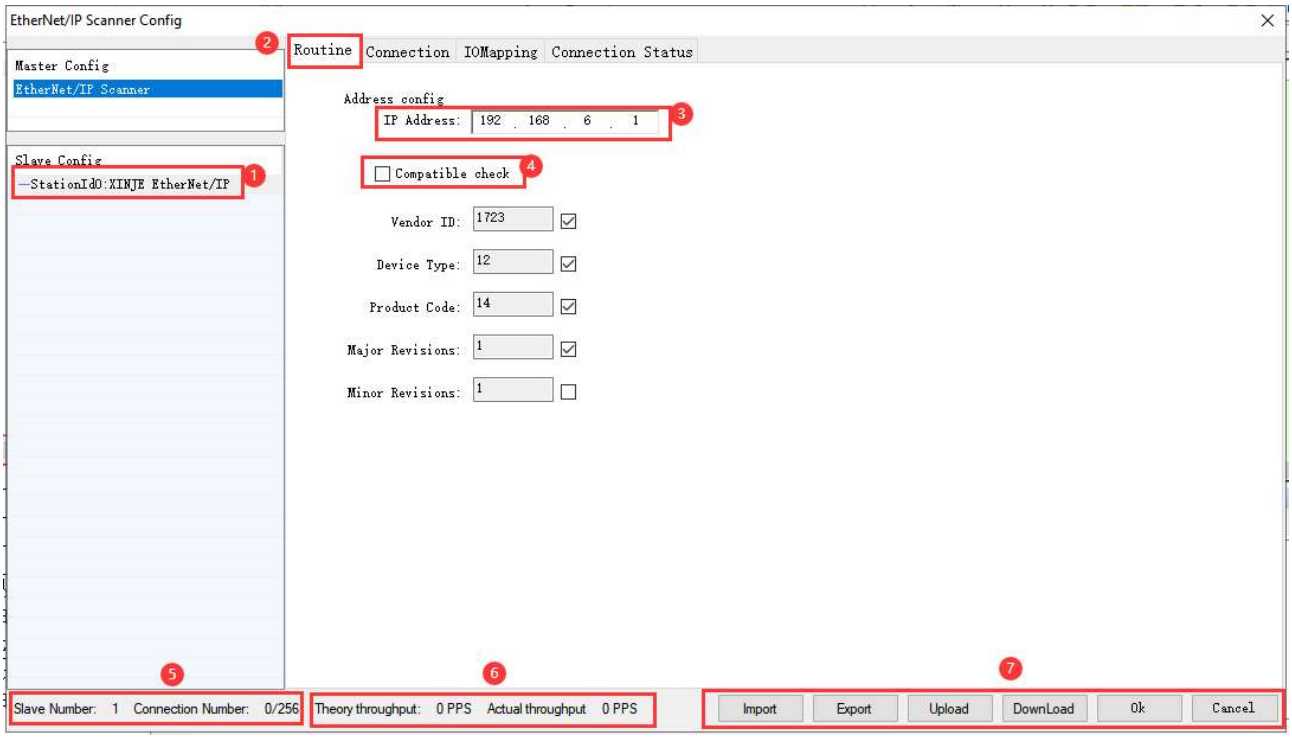
(1) Double click on EipScanner to enter the EtherNet/IP Scanner parameter configuration interface;

(2) Right click on EtherNet/IP Scanner to add devices;

(3) Load or unload third-party EDS files in the device library to prepare for the next communication configuration step;

(4) Double click or click to add the EDS file to be communicated, and add the specified slave to the slave configuration information bar.

2. Configure General Settings for Slave Stations



1		Double click on the node corresponding to the slave station configuration bar to configure relevant communication information
2		Click on Routine to configure the IP address and compatibility check accordingly
3		The IP address is the IP address of the slave device under the corresponding node
4		If compatibility check is conditional, check the checked conditions to determine whether the EDS file matches the slave device. If compatibility check is checked, all relevant information will be matched with the information in the EDS file by default. It should be noted that compatibility checks are conducted during the connection period. If the check fails, the connection cannot be made
5		The slave number is used to count the number of slave stations connected under the current master station, and the connection number is used to count the number of connections established between the master station and all connected slave stations. The specific specifications for the slave number or supported connections supported by Ethernet/IP communication can be found in 5-3. Ethernet/IP communication specifications
6		Theoretical throughput *1 is used to display the network throughput of the current connection, while actual throughput is used to display the throughput of the entire Ethernet network of the current device;
7	Import	Import the configured information into the current configuration interface in the form of an XML file
	Export	Export configured information in the form of an XML file
	Upload	Upload the configuration information downloaded to the PLC to the current configuration interface, and the uploaded configuration information will overwrite the existing configuration information on the current interface
	Download	Download the configuration information of the current configuration interface to the PLC. The downloaded configuration information will overwrite the original configuration information in the PLC and take effect in real time with the new configuration information
	OK	Click OK to save the configuration information for the current page
	Cancel	Click to discard the configuration information for the current page.

Note:

※ 1: pps is the unit of network throughput, also known as Packet Per Second, represents the total number of packet data packets that can be sent and received within 1 second.

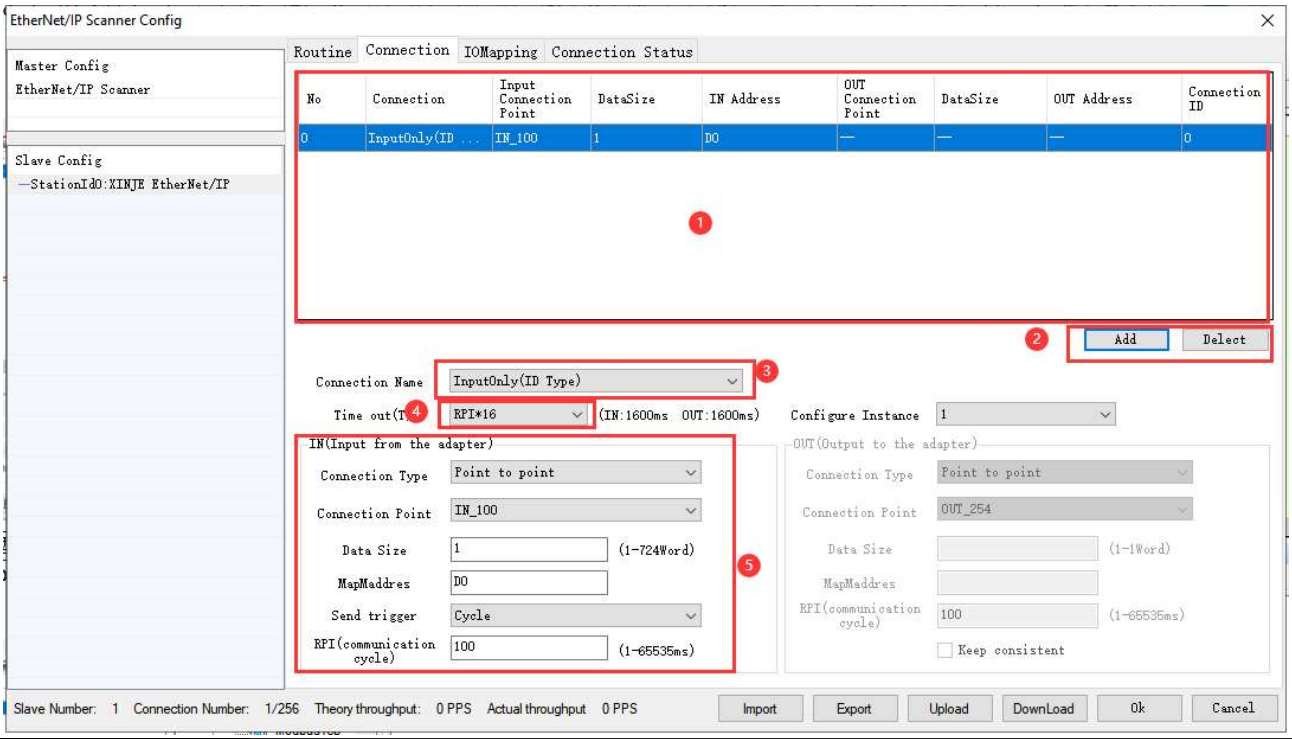
- When used as an InputOniy connection, the calculate formula for each connection:

When RPI<100ms, the theoretical throughput pps=1000ms/RPI+10;
When RPI>100ms, the theoretical throughput pps=1000ms/RPI * 2.
Example: Two PLCs establish implicit communication, and two InputOniy type connections are established in the Scanner connection configuration interface. One connection has an RPI communication cycle of 110ms, and the other connection has an RPI communication cycle of 10ms. So the total theoretical throughput pps=1000/110*2+(1000/10+10)=128pps.

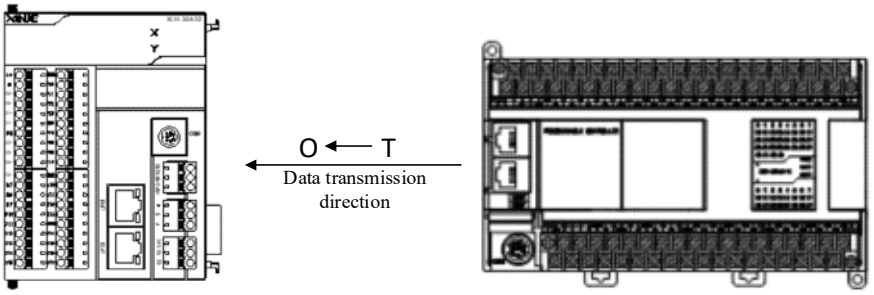
- When used as an ExclusiverOwner connection, the calculate formula for each connection:

RPI_1: Communication cycle from adapter input (T ->O) direction;
RPI_2: Communication cycle in the direction of output to adapter (O ->T);
Theoretical throughput pps=1000ms/RPI_1+1000ms/RPI_2.
Example: Two PLCs establish implicit communication and establish an ExclusiverOwner connection on the Scanner connection configuration interface. The communication period from the input (T ->O) direction of the adapter RPI_1 is 100ms, and the communication period from the output to the adapter (O ->T) direction RPI_2 is 10ms. The total theoretical throughput pps is 1000/100+1000/10=110pps.

3. Add the connection

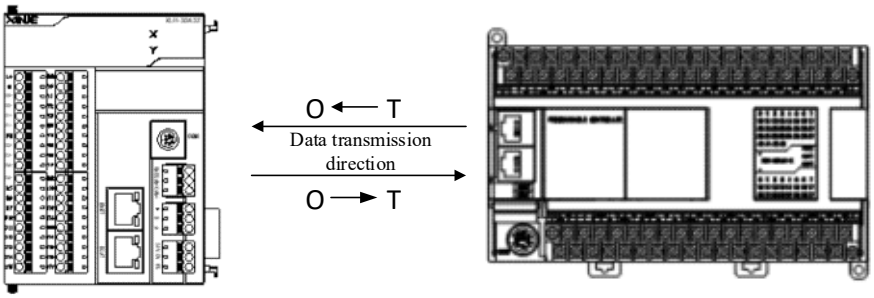


1	Connection display area	The connection display box can show connection types and corresponding configuration information
	Connection ID	Assign a unique connection ID to the added connection, which will not change with the addition or removal of the connection
2	Add	Clicking on add will create a new connection.
	Delete	Select the corresponding established connection, click delete to delete the selected connection
3	Connection name	● Only input, supports two types: ID Type and Tag Type, used by the scanner to request data from the adapter. The adapter can only send data to the scanner, that is, data transmission is in the T ->O direction.

<table>
<tr><td></td><td></td><td>


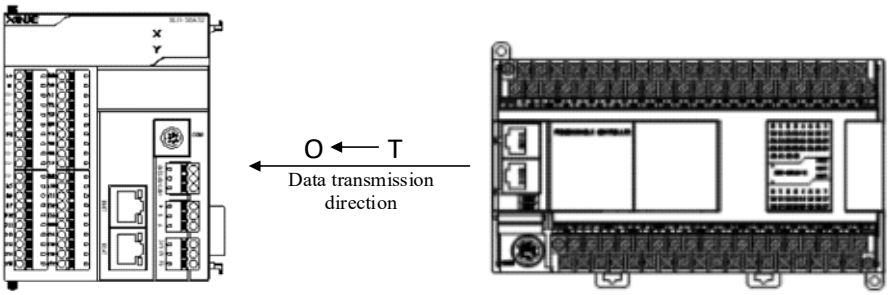
Scanner (O)
EtherNet/IP master
Adapter (T)
EtherNet/IP slave

ExclusiveOwner: Supports ID Type/Tag Type, which allows for bi-directional data transmission between the scanner and adapter, allowing for simultaneous data transmission in both T ->O direction and O ->T direction.



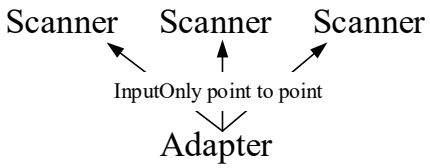
Scanner (O)
EtherNet/IP master
Adapter (T)
EtherNet/IP slave

ListenOnly: The ID Type connection type that only supports multicast types is used by the scanner to listen to the adapter and only receive data. Its data direction is from the adapter to the scanner, that is, data transmission is in the T ->O direction.

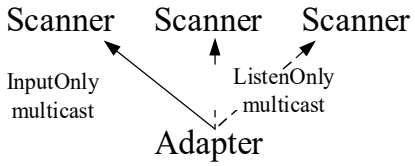


Scanner (O)
EtherNet/IP master
Adapter (T)
EtherNet/IP slave
</td></tr>
<tr><td>4</td><td>Timeout (T)</td><td>Timeout refers to the time it takes to wait for a response after a request is sent. If no response is received within the specified timeout period after the request is sent, it will be considered a request failure. The timeout time is set according to the network situation, and RPI refers to the communication cycle.</td></tr>
<tr><td>5</td><td>Connection type</td><td>There are two methods available for users to use: point-to-point and multicast:

Point to point: Implement data exchange between any two devices in the network, in which corresponding data frames need to be sent between any two scanners and adapters. (Note: The solid arrow represents the data frame that needs to be sent)



</td></tr>
</table>

		● Multicast: data exchange between multiple devices in the network. In this way, multiple scanners can simultaneously obtain data in the T ->O direction sent from the same adapter, and only one corresponding data frame needs to be sent during data exchange, which can save adapter network resources to a certain extent. (Note: The implementation arrow represents the data frames that need to be sent, and the dashed arrow represents the fewer data frames sent compared to point-to-point transmission in multicast.)  When using ListenOnly (ID Type) multicast, it must be attached to an InputOnly (ID Type) or ExclusiveOwner (ID Type) connection, and the corresponding connection type attached to InputOnly or ExclusiveOwner must also be multicast. The configured data size and RPI communication cycle must be consistent with the attached connection type, otherwise the establishment will not be successful.
	Connection point[*1]	The tag name or instance ID required to establish communication.
	Data size	The number of registers connected for data transmission
	Map first address [*2]	The starting address for data transmission in this connection
	Send trigger	● Loop: Trigger the scanner periodically based on the set RPI; ● State change: When the status of the adapter changes, the scanner is triggered. If the status of the adapter changes periodically and is less than 1/4 of the RPI, the scanner is triggered periodically at 1/4 of the RPI; ● Application trigger: Trigger rules are consistent with state changes.
	RPI	Used to set the communication time for the corresponding connection cycle, with a default of 100ms and a setting range of 1-65535ms. RPI (communication cycle) can be set according to the priority of data transmission and reception, so as to adjust the overall communication volume for data transmission and reception.

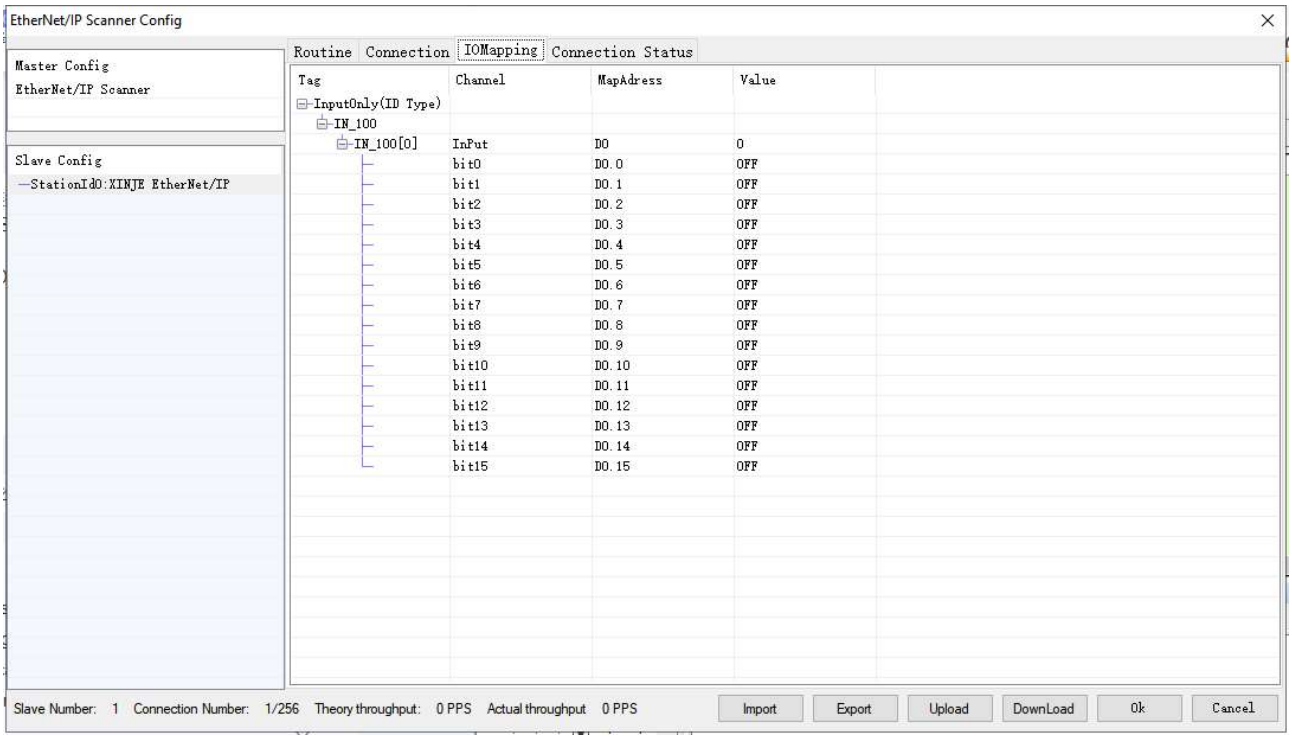
Note:

※ 1: When establishing a connection with the corresponding tag or instance ID, it is important to note that the data length configured by the adapter should be consistent with the data length configured by the scanner to avoid communication abnormalities when establishing the corresponding connection;

※ 2: The starting address currently supports two register types, D and HD.

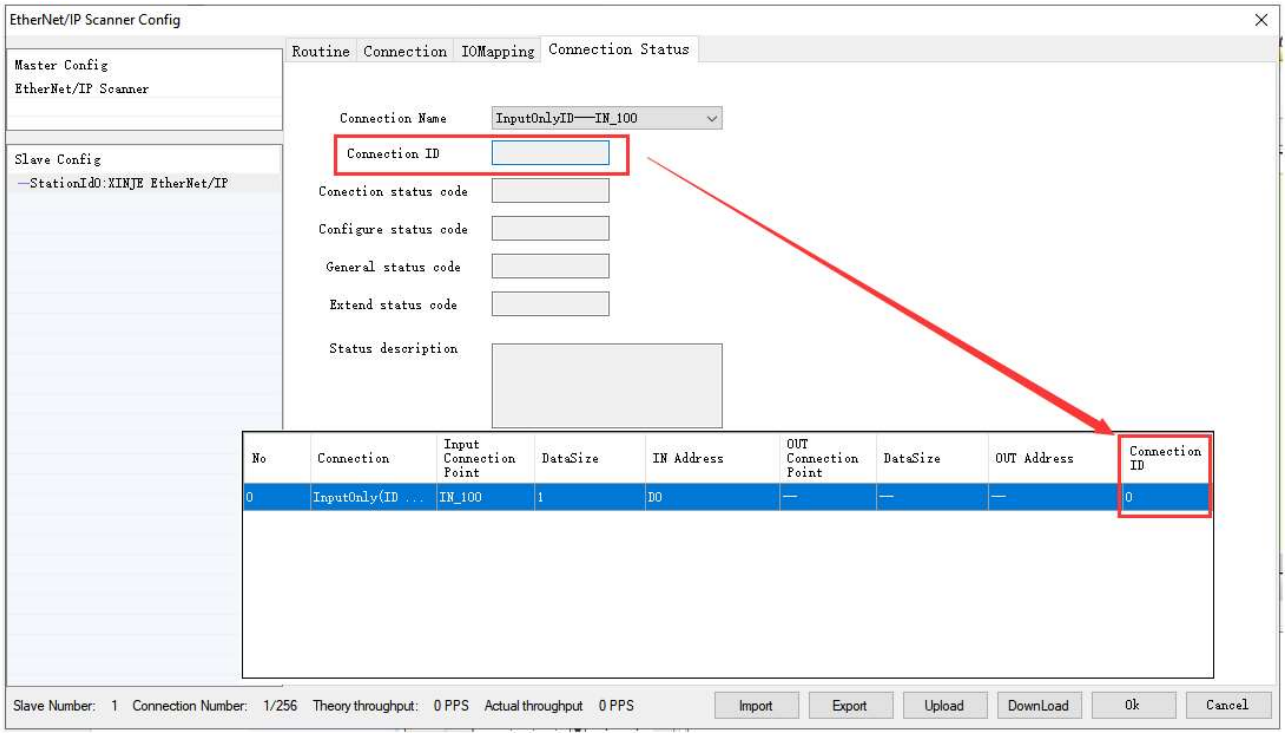
4. IO mapping

The IO mapping interface can display mapping addresses for different connection configurations and view the status of data in real-time.



5. Connection status

You can view the status information of each connection in real-time, where the connection ID on the "Connection" configuration interface is consistent with that on the "Connection Status" configuration interface.



Connection name	Select the various connections that have been added to the current slave station's "connections".
Connection ID	The connection ID corresponding to the connection.
Connection	Display the current connection status in hexadecimal.

status code	
Configure status code	Display the current configuration status in hexadecimal.
General status code	Display the current general status in hexadecimal.
Extend status code	Display the current extended status code in hexadecimal.
Status description	Used to display the status information of the current connection.

Note: For detailed extension status codes, please refer to Appendix Ethernet/IP communication extension codes.

6. Rules for judging descriptive information

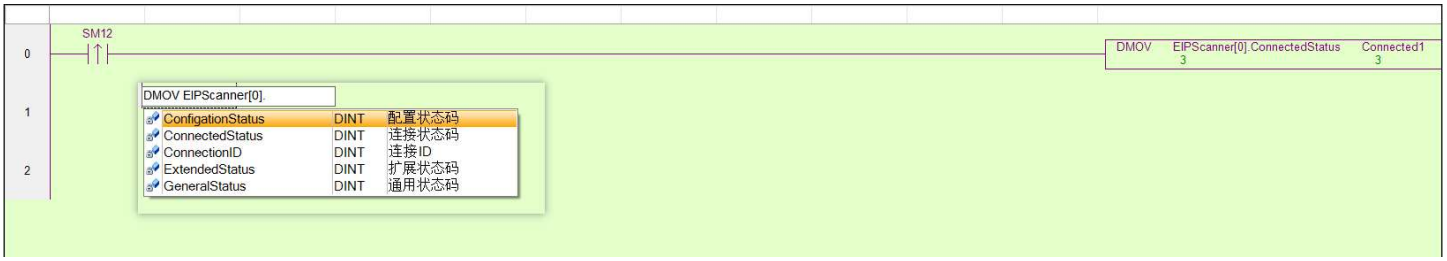
The combination of information that is not in the following three states is prompted in the "state description": Undefined Error!

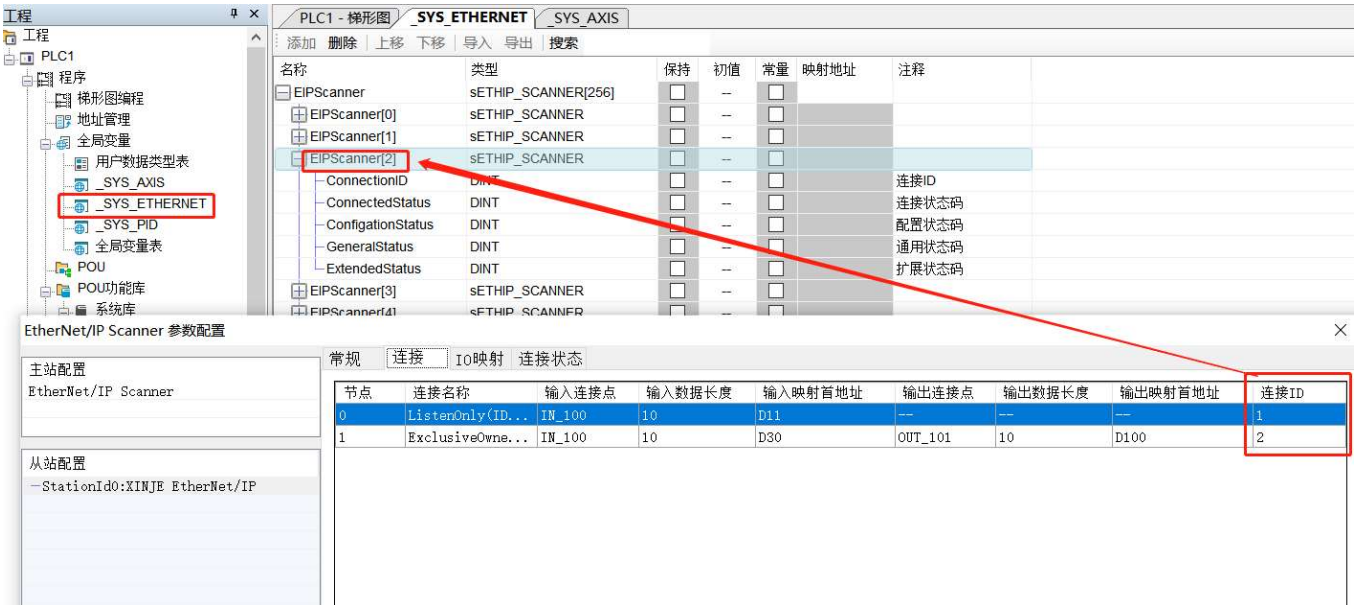
- When the "Connection Status Code"=3, "General Status Code"=0, and "Extended Status Code"=0, the configuration status code does not need to be determined, and the status description prompts "Connection successful, communication normal";
- When "Connection Status Code"=1 and "Configure Status Code"=4, the status description prompts "Unable to find IP or the IP does not support EIP";
- When the "Connection Status Code"=1 and the "General Status Code"=1, there is no need to determine the configure status code. The specific information in the status description is prompted based on the "Extended Status Code".

7. Viewing Connection Status of Structural System Variables

In the ladder diagram, the system structure variables can be directly called to view the current connection status. When calling the ladder diagram or freely monitoring the connection status, the corresponding structure number is the "connection ID" of the corresponding connection.

Example: View the communication status with connection ID 0 through a ladder diagram to determine if the corresponding connection status code is 3. If the connection status code is 3, it indicates successful communication.

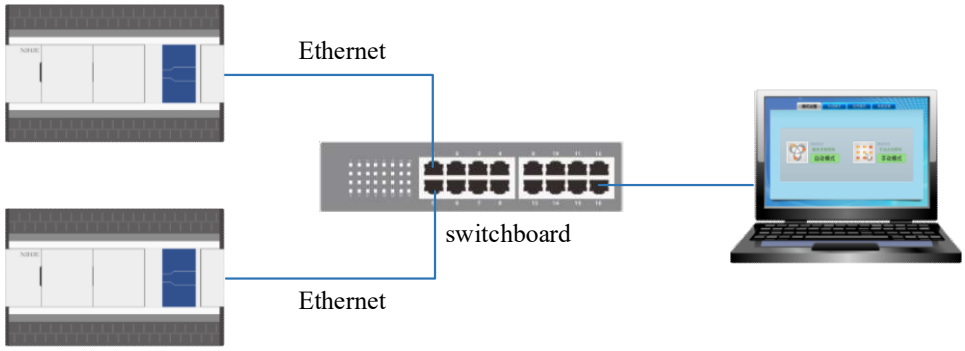




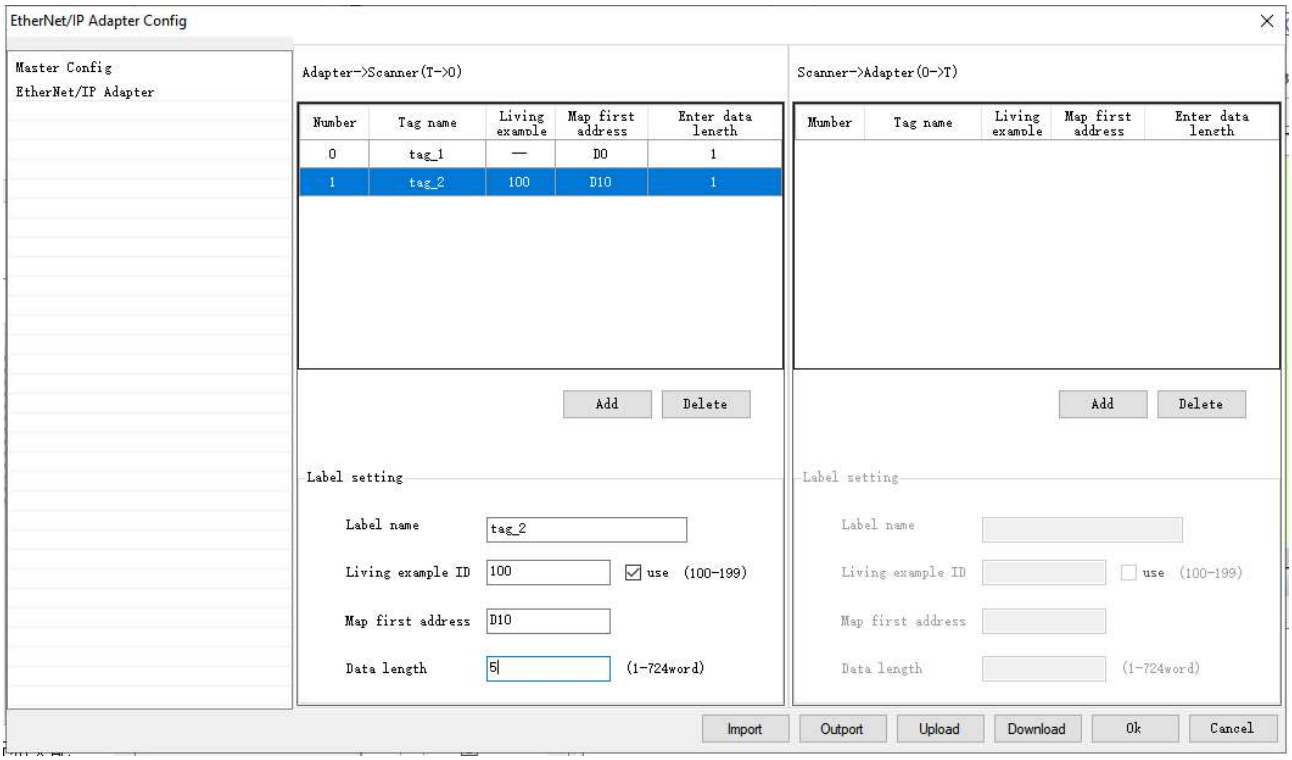
5-4-1-3. Implicit communication application

Application 1: Implicit communication by using two XDH-60T-E PLCs with InputOnly connection type.

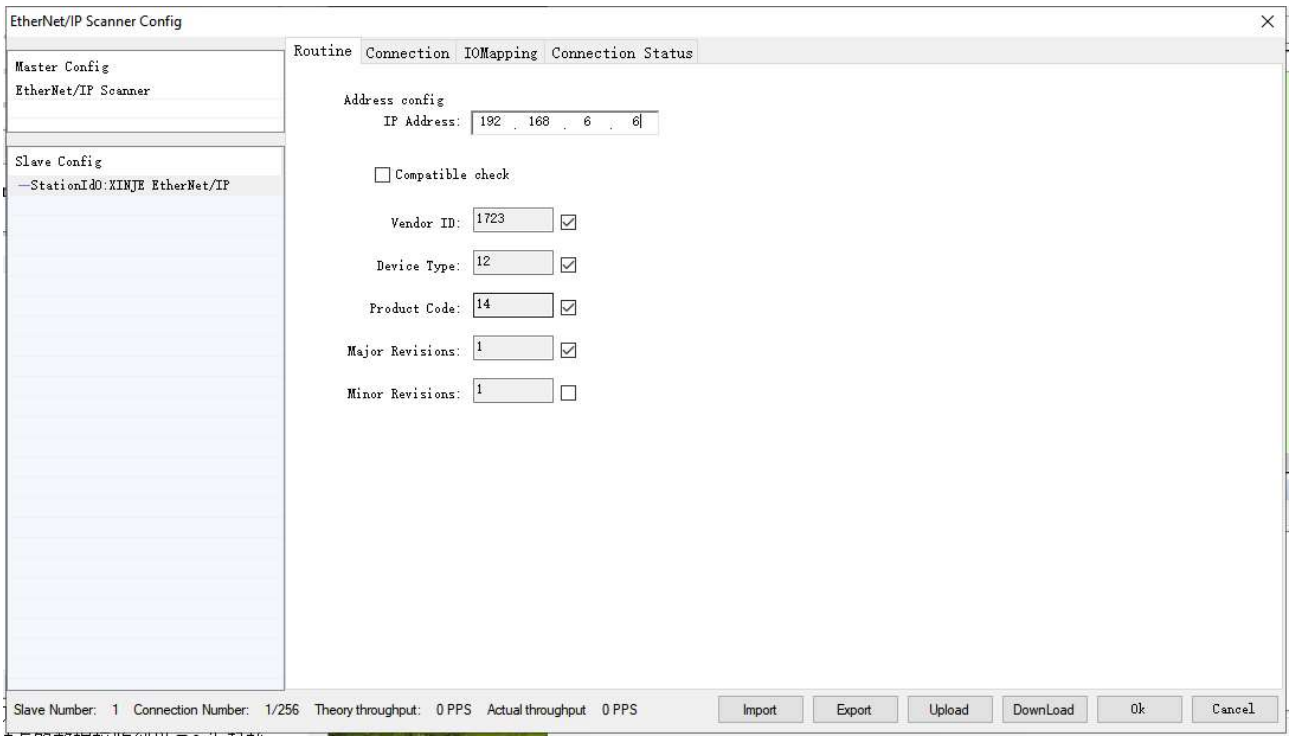
Use PLC1: XDH-60T-E (IP 192.168.6.6) as the adapter and PLC2: XDH-60T-E (IP 192.168.6.7) as the scanner to achieve implicit communication between two PLCs. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with that of the adapter.



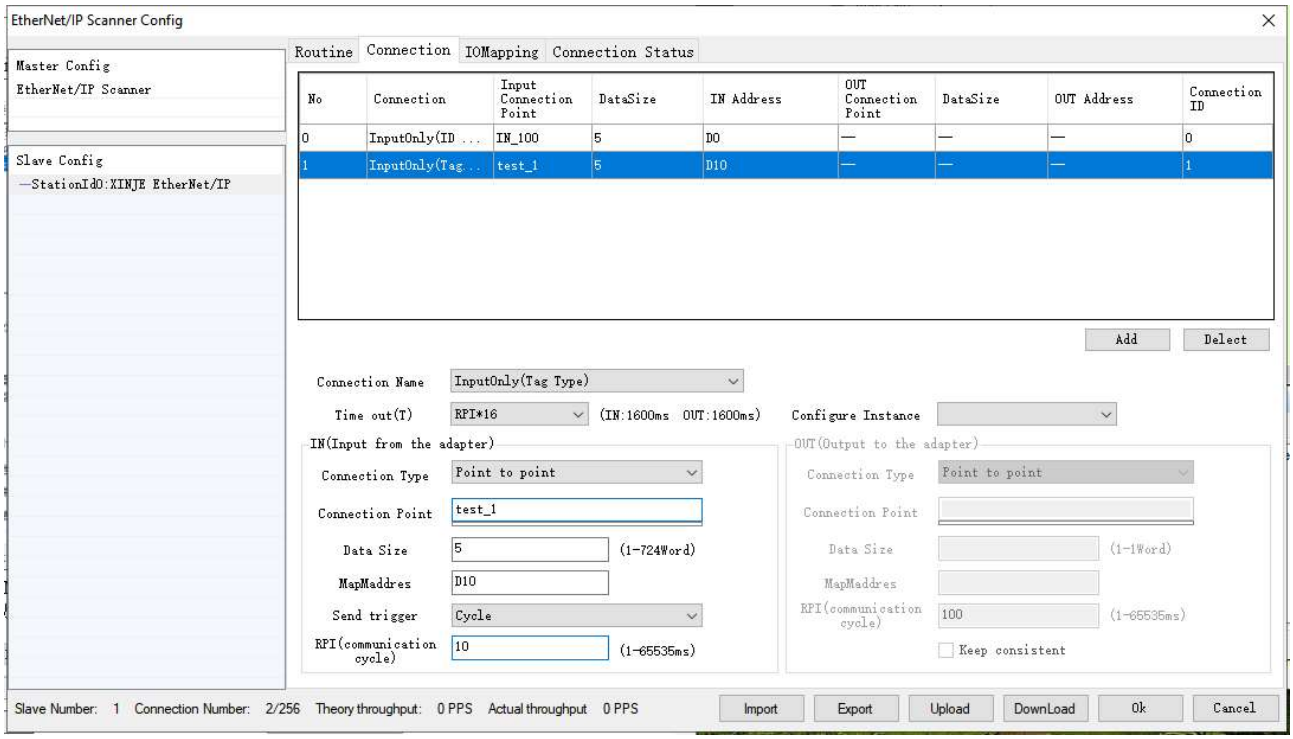
Step 1: Create and add a communication type with test_1 as the label in the adapter, mapping the first address to D0, and inputting a data length of 5. Use the communication type with test_2 as the label for instance ID100 (check the use of instance ID), the mapping first address is D10 and the data length is 5.



Step 2: Add slave devices to the scanner and perform relevant configuration operations on the adapter's IP address and compatibility check:



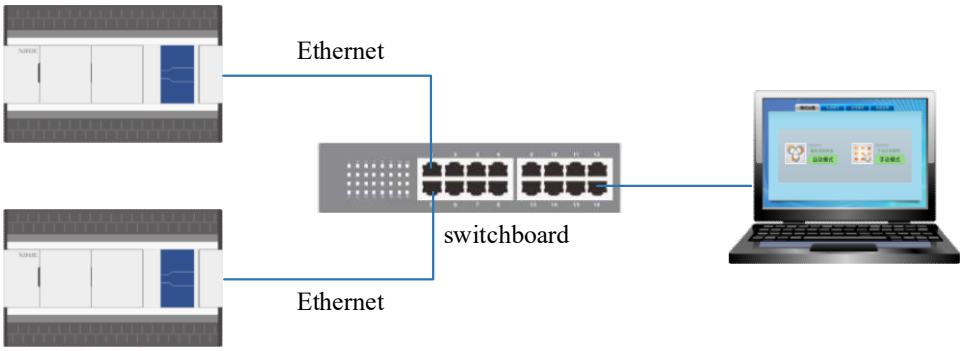
Step 3: Add two types of connections, InputonIy (Tag Type) and InputonIy (ID Type), and establish the first connection of InputonIy (ID Type). Use the Adapter ->Scanner input connection point as IN_100 and the connection type as point-to-point to receive data into five registers with D0 as the starting address. Establish the second connection of InputonIy (Tag Type), and use the Adapter ->Scanner input connection point as test_1, The data with point-to-point connection type is received in 5 registers starting from D10.



Step 4, enter the connection status to check the connection status of IN_100 and test_1. When the connection status shows successful connection and communication is normal, it indicates that the communication has been successfully established. You can also check whether the data is correct through the corresponding mapped register.

Application 2: Implicit communication by using two XDH-60T-E PLCs with ExclusiveOwner connection type

Use PLC1: XDH-60T-E (IP 192.168.6.6) as the adapter and PLC2: XDH-60T-E (IP 192.168.6.7) as the scanner to achieve implicit communication between two PLCs. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with that of the adapter.



Step 1:

---Create two connections in Adapter ->Scanner (T ->O) direction.

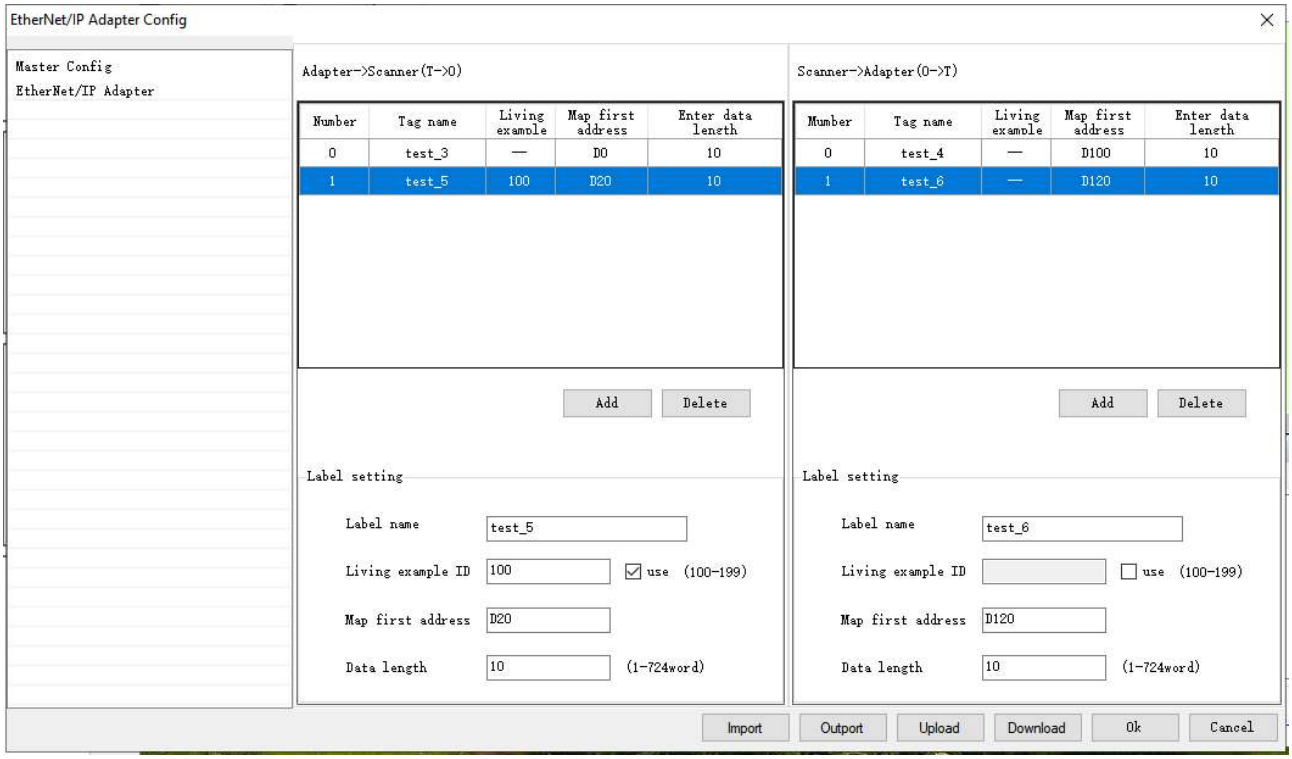
Connect1: communication type with the label as test_3 on the adapter. Map the first address as D0 and input data length as 10.

Connection 2: communication type with instance ID100, label name test_5 on the adapter. Map the first address as D20 and input data length as 10.

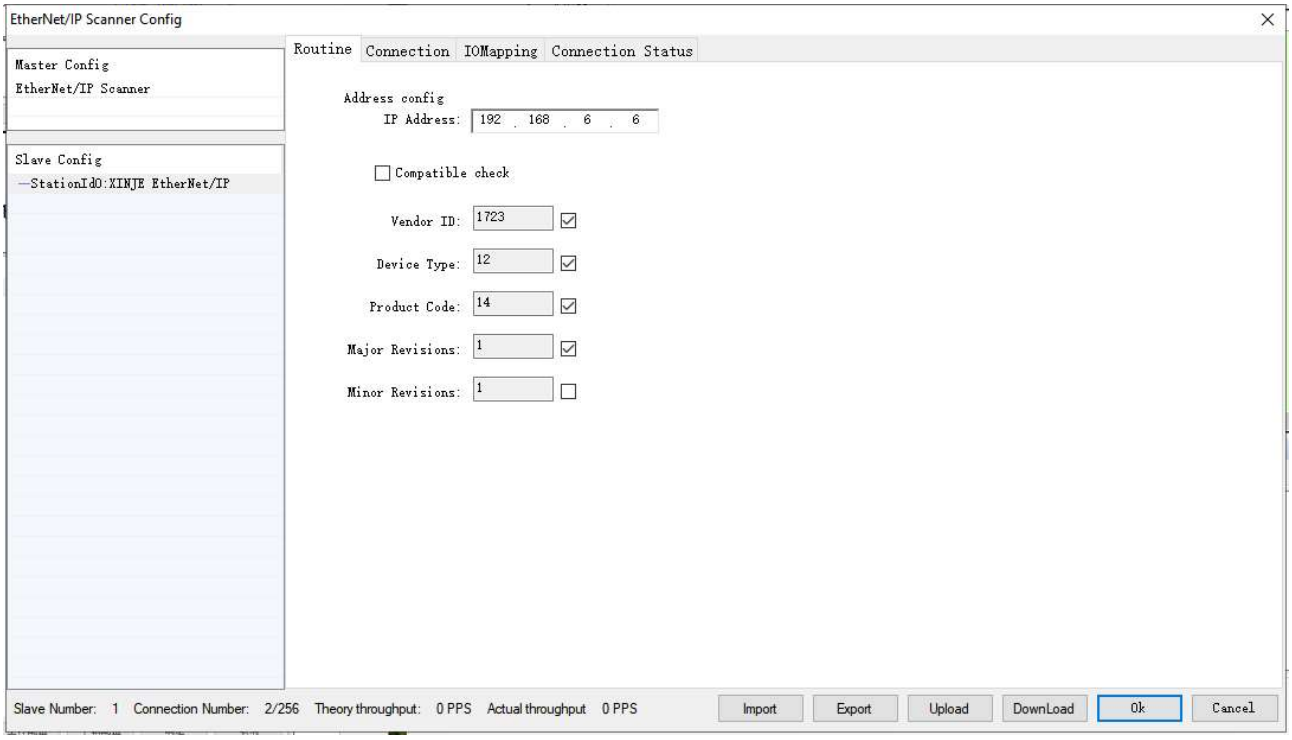
---Create two connections in Scanner->Adapter(O->T) direction.

Connection 1: communication type with the label as test_4 on the adapter. Map the first address as D100 and input data length as 10.

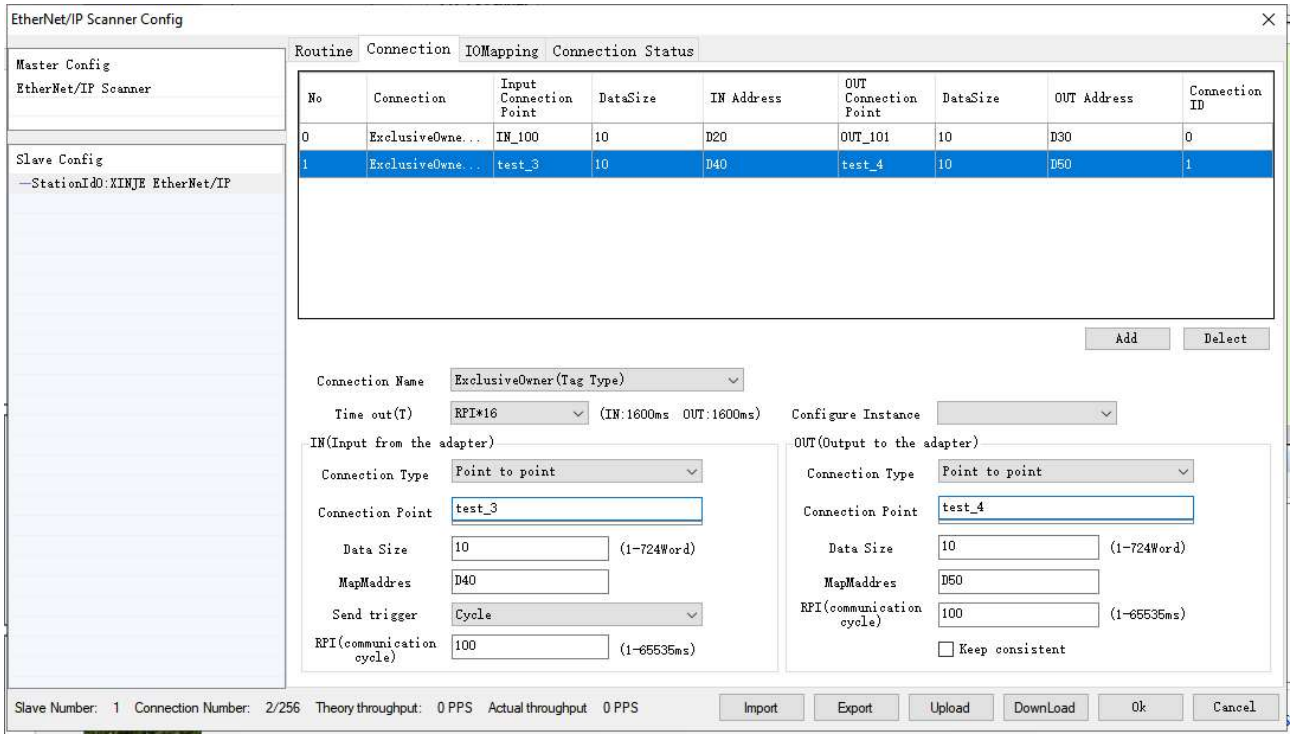
Connection 2: communication type with the instance ID101, label name test_6 on the adapter. Map the first address as D120 and input data length as 10.



Step 2: Add slave devices to the scanner and configure the IP address and compatibility check of the adapter accordingly.



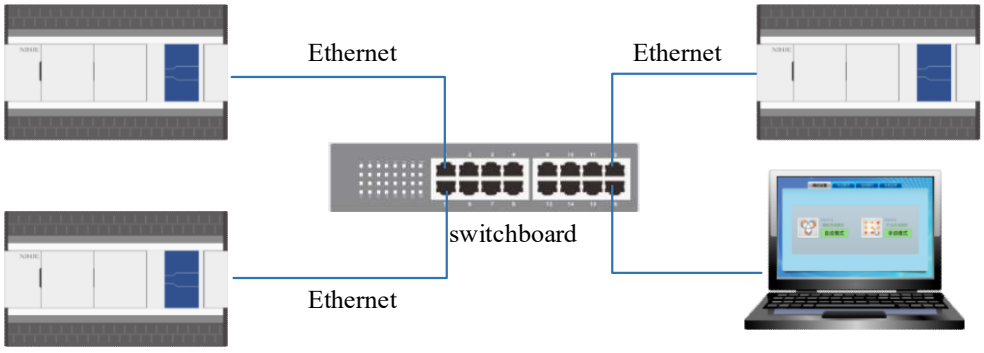
Step 3: Add two types of connections on the scanner: ExclusiveOwner (Tag Type) and ExclusiveOwner (ID Type); Establish the first connection for ExclusiveOwner (ID Type), and receive data from Adapter ->Scanner with input connection point IN_100 and connection type point-to-point into 10 registers starting from D20. Send out 10 data from Scanner ->Adapter with output connection point OUT_101 and connection type point-to-point, D30 as the starting address. Establish a second connection for ExclusiveOwner (Tag Type), and receive data from Adapter ->Scanner with input connection point test_3 and connection type point-to-point into 10 registers with D40 as the starting address. Send out 10 data from Scanner ->Adapter with output connection type point-to-point, connection point test_4 and D50 as the starting address.



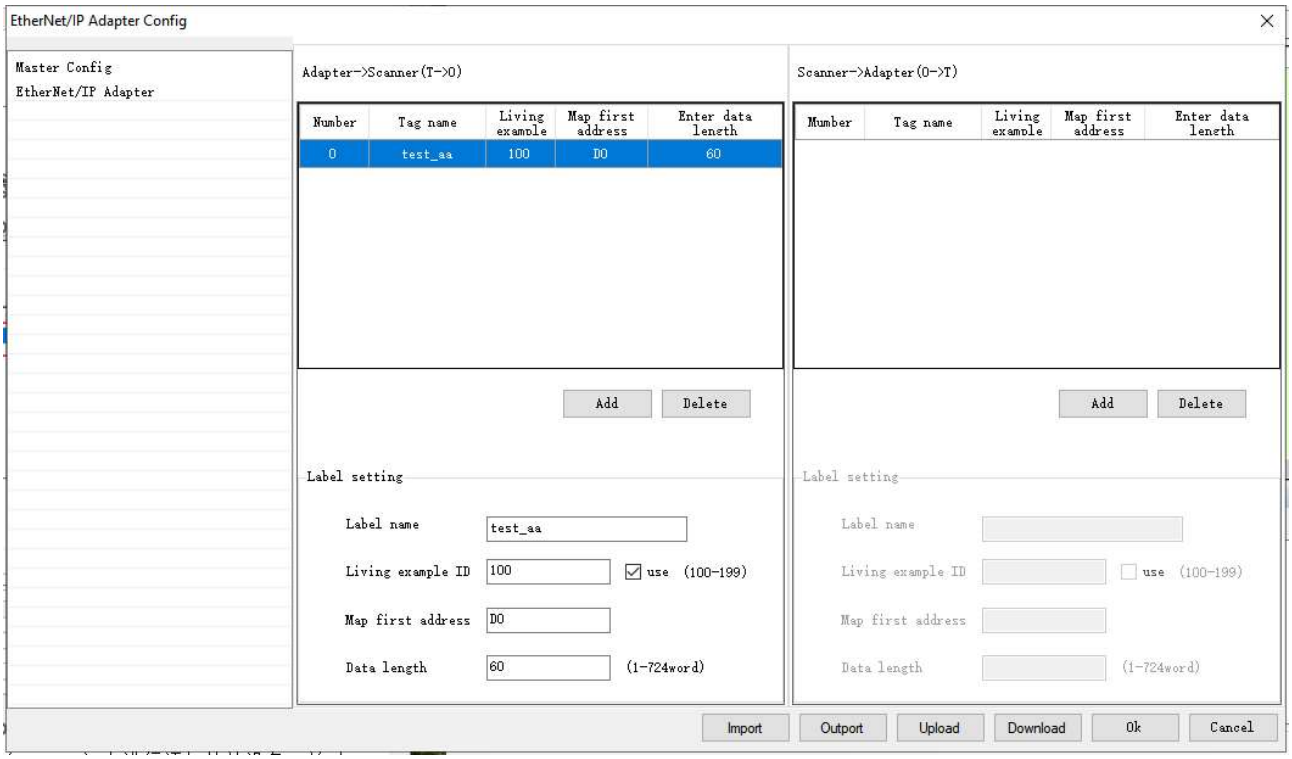
Step 4: Enter the connection status to check the connection status of IN_101 and test_3. When the connection status shows successful connection and communication is normal, it indicates that the communication has been successfully established. The corresponding mapped register can also be used to check if the data is correct.

Application 3: Implicit communication by using three XDH-60T-E PLCs with ListenOnly connection type.

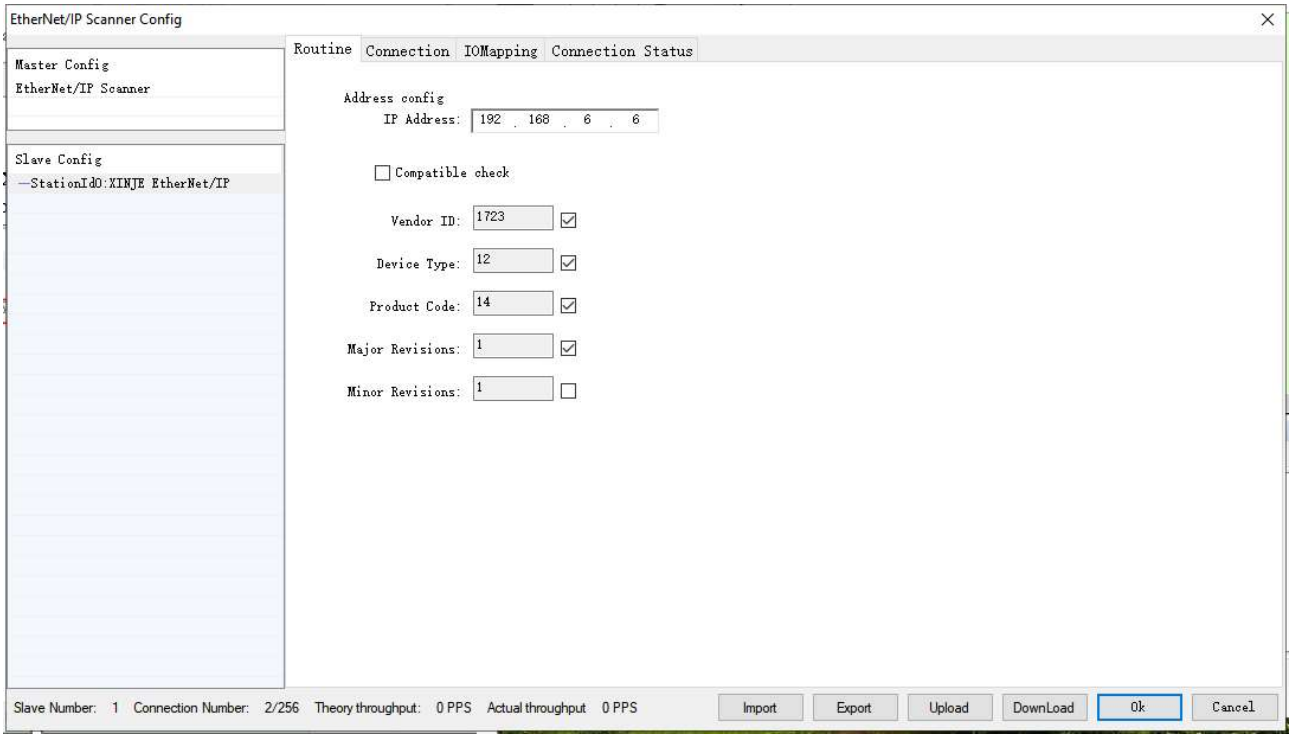
Using PLC1: XDH-60T-E (IP 192.168.6.6) as the adapter, PLC2: XDH-60T-E (IP 192.168.6.7) as the scanner, and PLC2: XDH-60T-E (IP 192.168.6.20) as the scanner, to achieve multicast transmission of the 60 register data of adapter D0-D59 to the other two scanners' HD0-HD59 registers. During the connection creation process, attention should be paid to the connection type used, connection point, data size set, and RPI (communication cycle). The configuration needs to be consistent with the adapter.



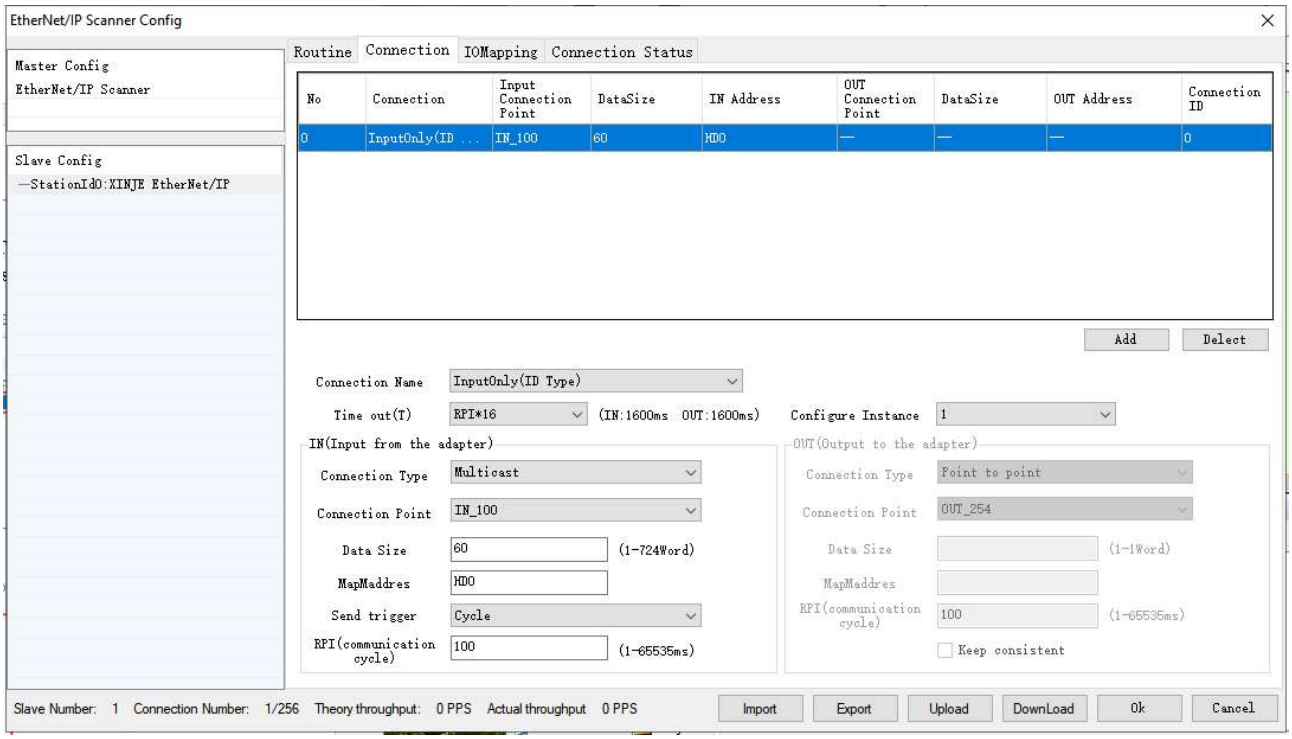
Step 1: Create a communication type with an instance ID100 labeled as test_aa in Adapter ->Scanner (T ->O) direction on the adapter, mapping the initial address to D0, and entering a data length of 60 (check the use of instance ID).



Step 2: Add slave devices to the scanner of PLC2: XDH-60T-E (IP 192.168.6.7) and perform relevant configuration operations on the adapter's IP address and compatibility check:

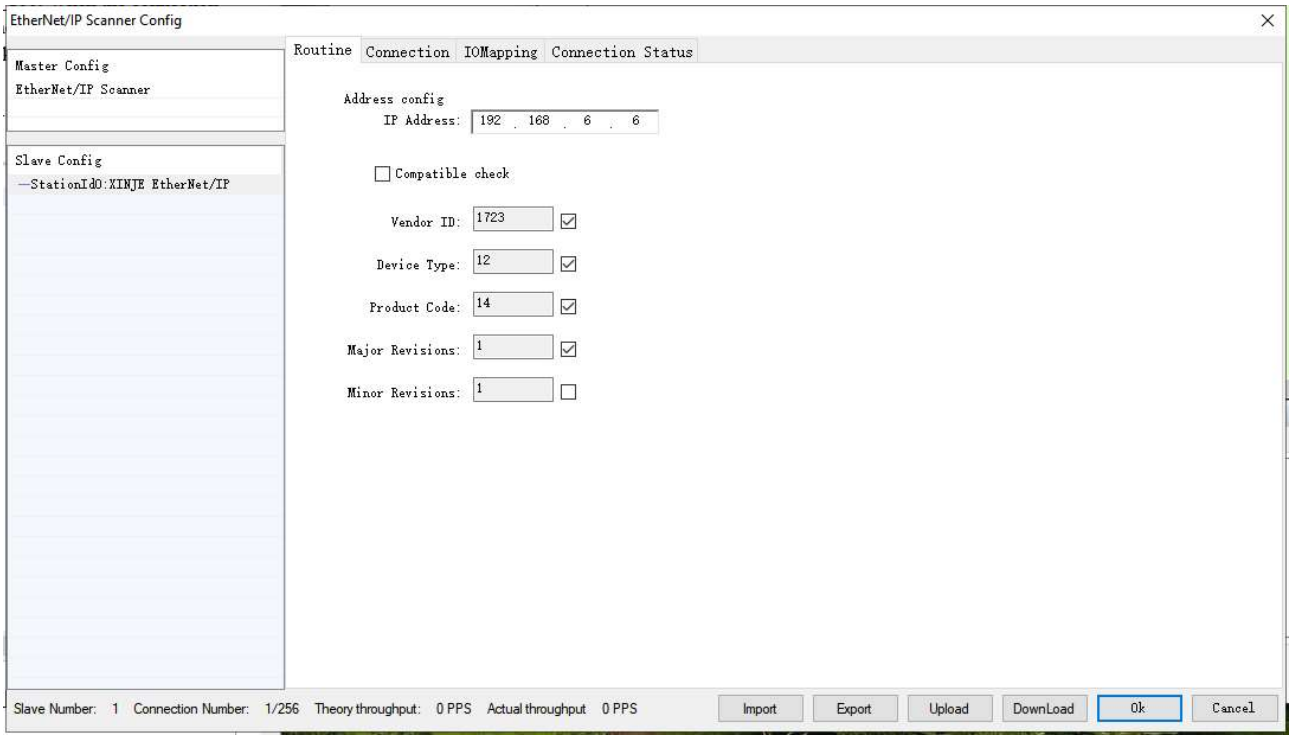


Step 3: Add an InputOnly (ID) connection type on the scanner of PLC2: XDH-60T-E (IP 192.168.6.7), with multicast as the connection type, IN_100 as the connection point, 60 words in data size, HD0 as the mapping address, and 100ms as the RPI (communication cycle).

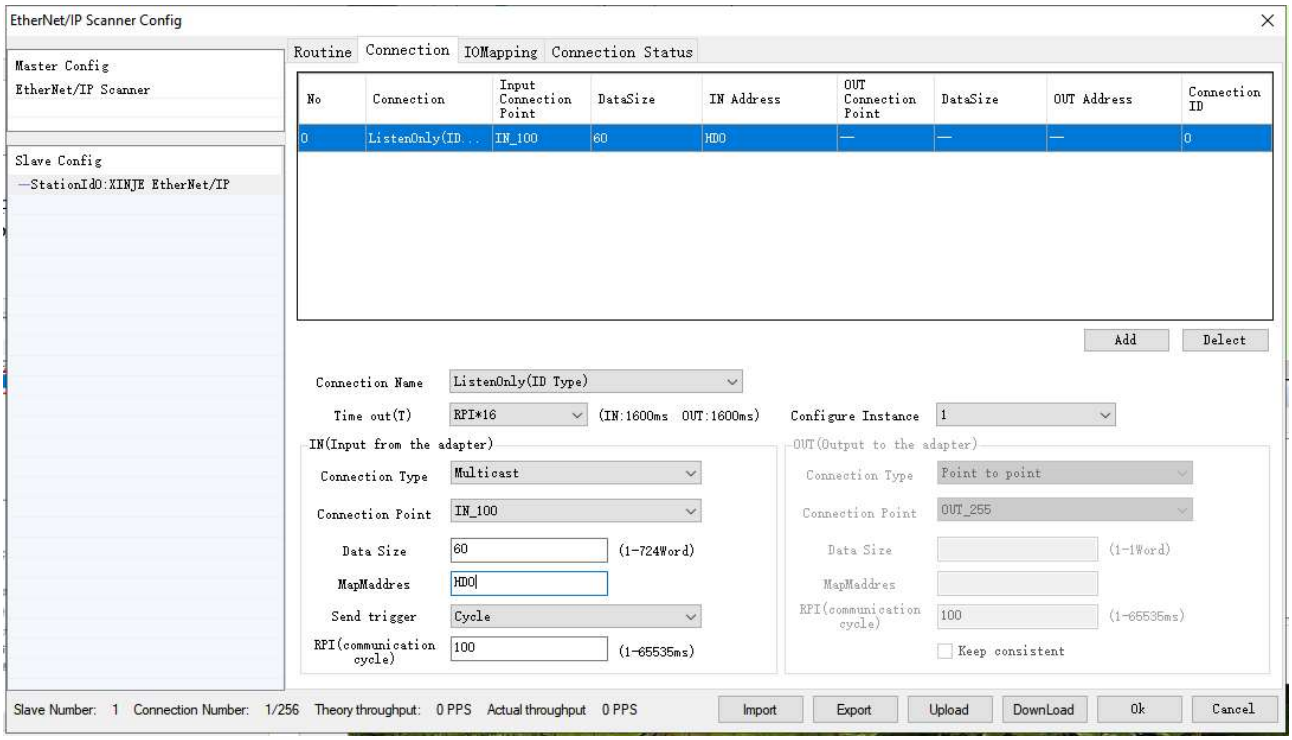


Step 4: Enter the connection status to check the connection status of InputOnly---IN_100. When the connection status shows successful connection and communication is normal, it indicates that the communication has been successfully established. The corresponding mapped register can also be used to check if the data is correct.

Step 5: Add slave devices to the scanner of PLC3: XDH-60T-E (IP 192.168.6.20) and perform relevant configuration operations on the adapter's IP address and compatibility check:



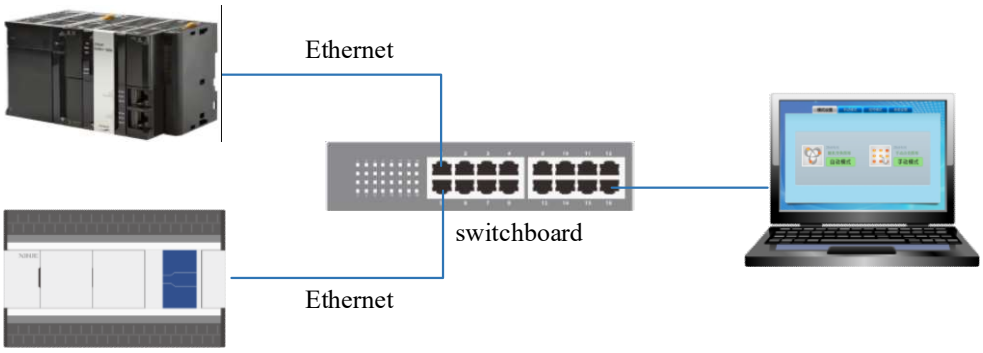
Step 6: On the scanner of PLC3: XDH-60T-E (IP 192.168.6.20), add a ListenOnly (ID Type) connection type to listen for the T ->O direction data sent by PLC1 to PLC2. Select multicast as the connection type, IN_100 as the connection point, 60 words in data size, HD0 as the mapping address, and 100ms as the RPI (communication period).



Step 7: Enter the connection status to check the connection status of ListenOnly---IN_100. When the connection status shows successful connection and communication is normal, it indicates that the communication has been successfully established. The corresponding mapped register can also be used to check if the data is correct.

Application 4: Using Xinje PLC XDH-60T-E as an adapter and Omron PLC NJ501-1500 as a scanner for implicit communication.

PLC1: XDH-60T-E (IP 192.168.250.20) as the adapter and PLC2: NJ501-1500 (IP 192.168.250.1) as the scanner to achieve implicit communication between two PLCs. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with the data size of the adapter.



Step 1: Add four connections on the adapter XDH-60T4-E in the direction of Adapter->Scanner(T->O).

Connection 1: Label name test_a, mapping first address D0, data length 10.

Connection 2: Label name test_c, instance ID100, mapping first address is D30, data length 10.

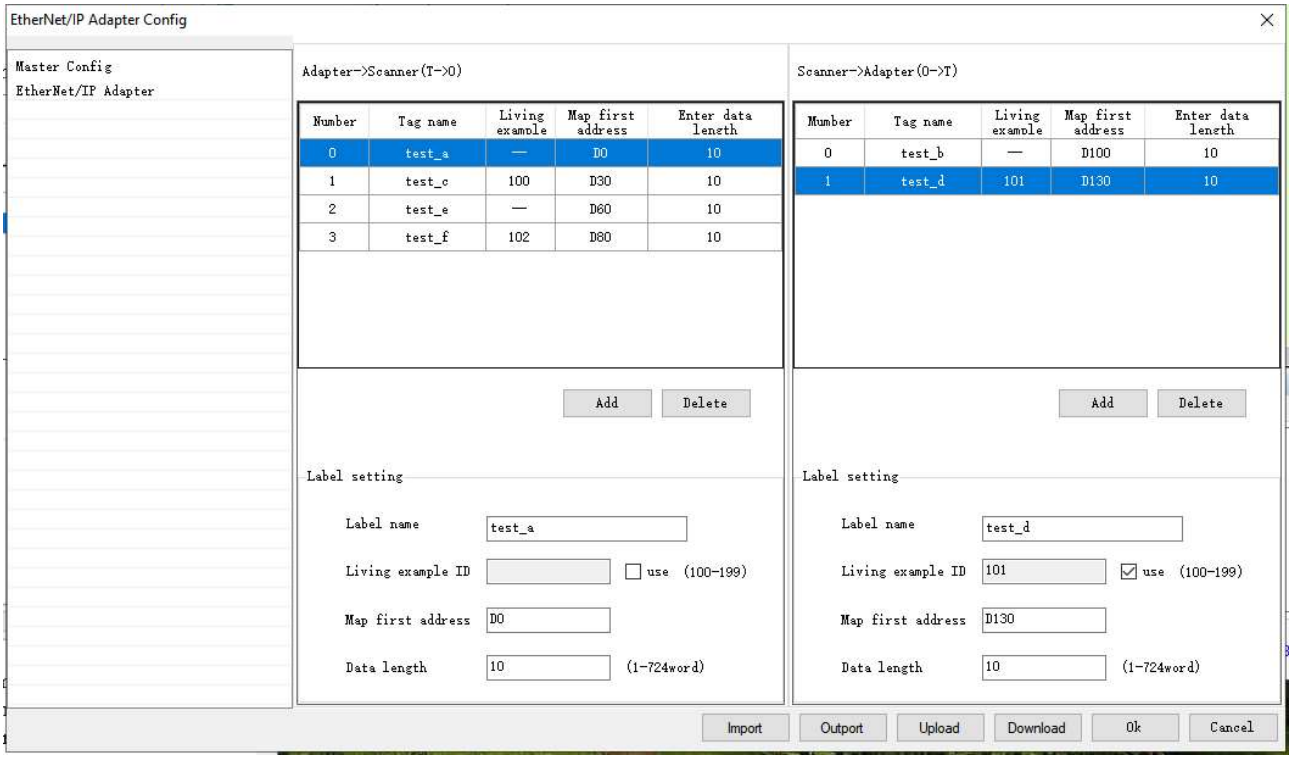
Connection 3: Label name test_e, mapping first address D60, data length 10.

Connection 4: Label name test_f, instance ID 102, mapping first address D80, data length 10.

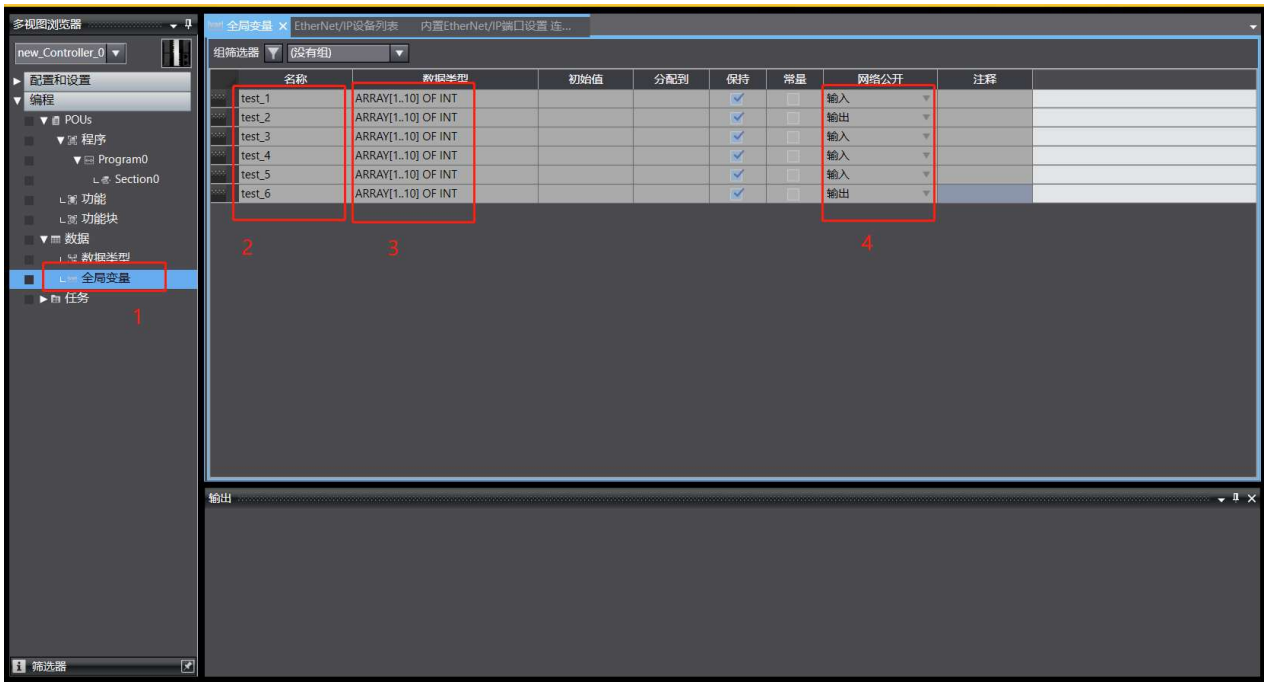
Add two connections in the direction of Scanner->Adapter(O->T).

Connection 1: Label name test_b, mapping first address D100, data length 10.

Connection 2: Label name test_d, instance ID101, mapping first address D130, data length 10.

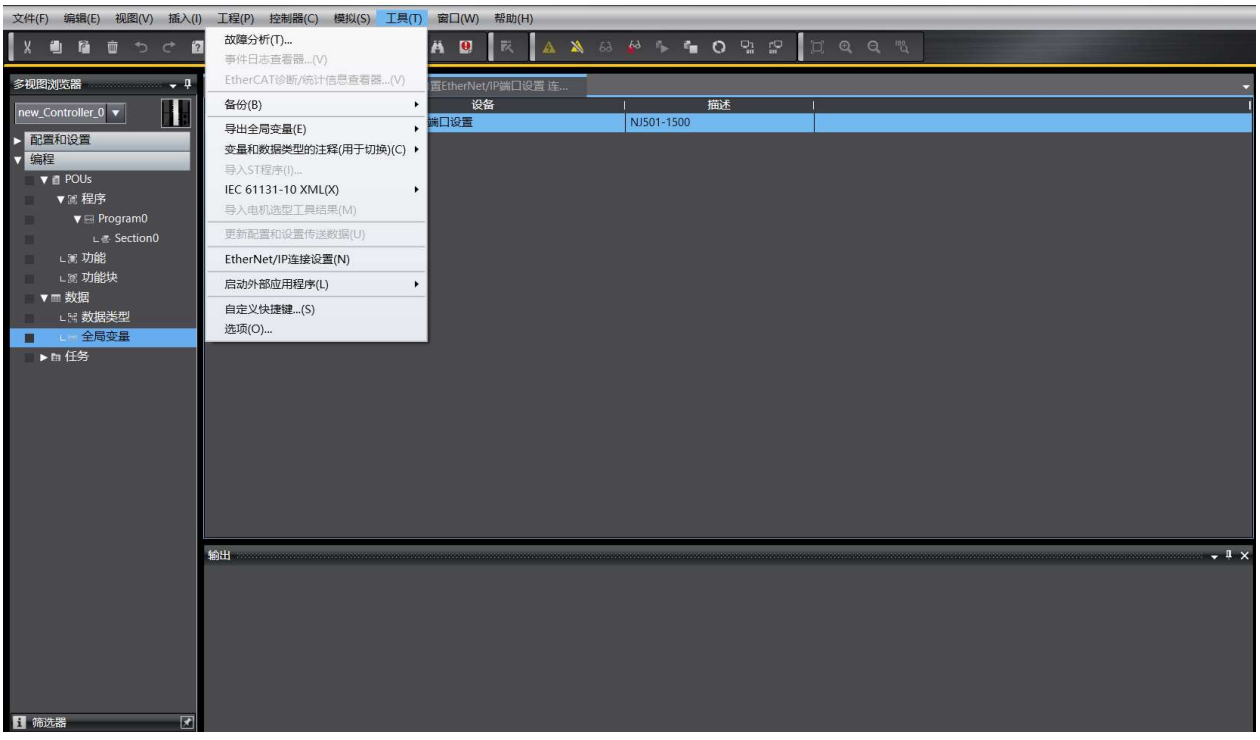


Step 2: Define the variables that require EIP communication in the Omron Sysmac Studio programming software. Global variables that require data reception and processing with the communication object during EIP communication are selected as inputs in the network public, while global variables that require data transmission and processing with the communication object are selected as outputs in the network public.

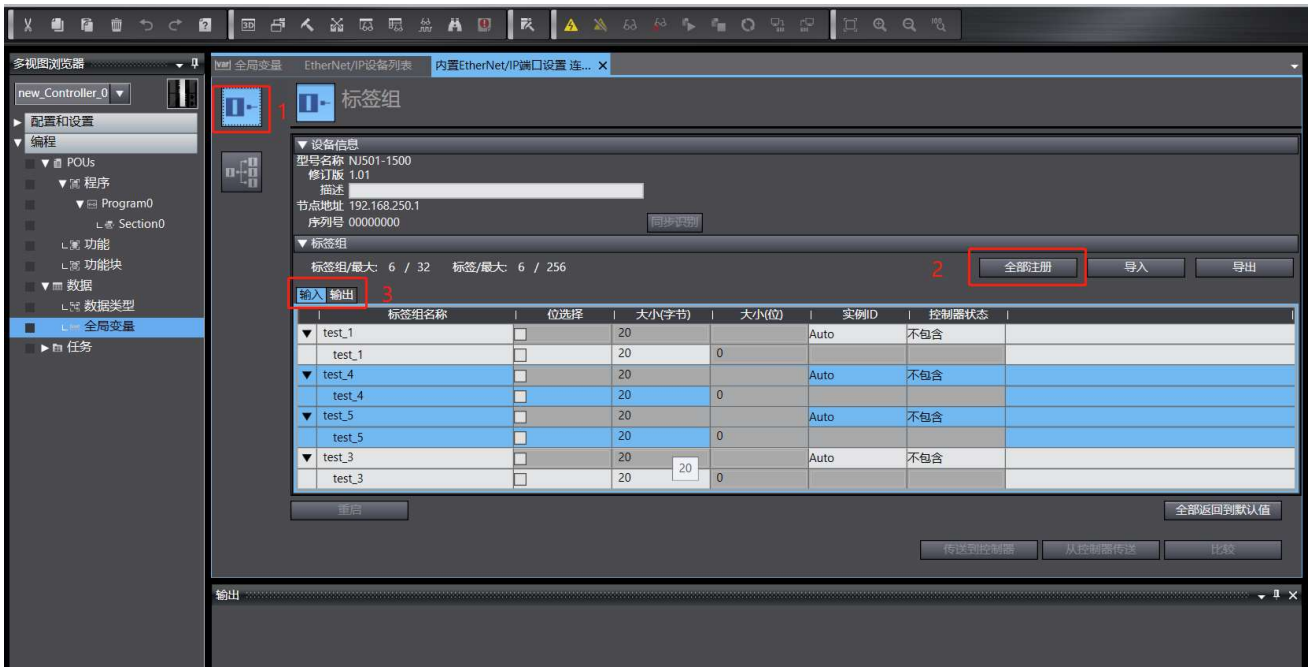


- Double click on the global variable to create a new one and add the variable type and data length to be transferred;
- Customize the name of the created variable;
- Define data types and lengths based on the length of input/output;
- Select the corresponding input and output types for the defined variable in the network public as needed.

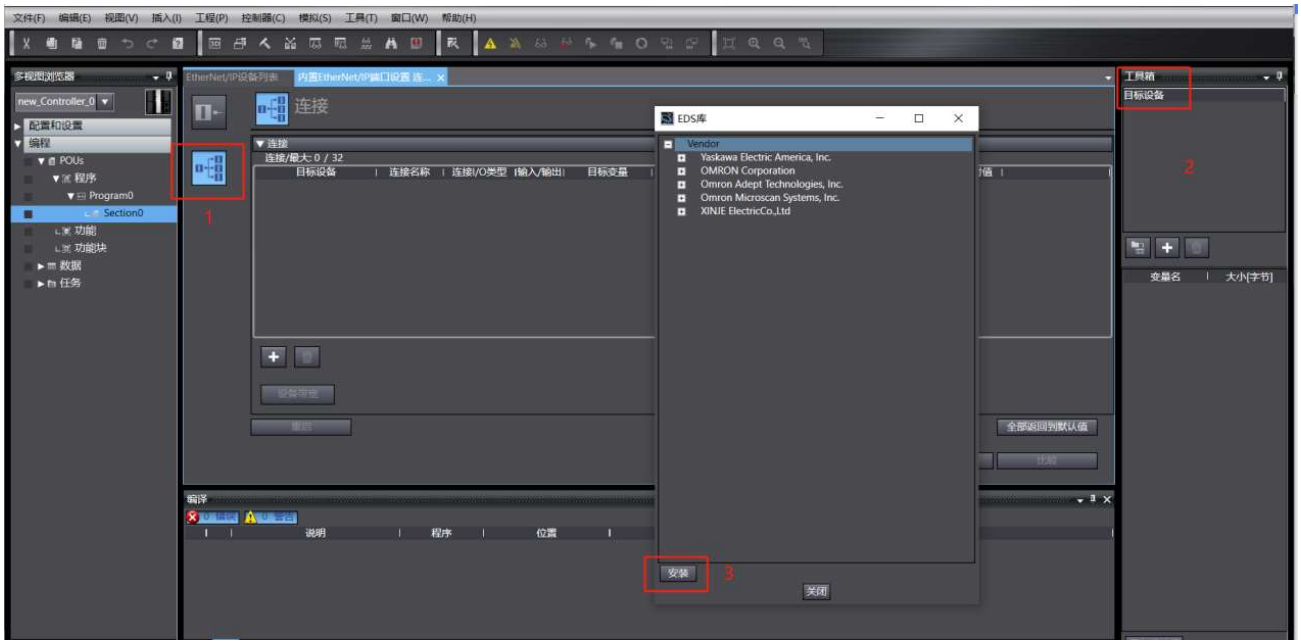
Step 3: Enter the Ethernet/IP connection settings operation page, click on Tools in the function bar, select and click on Ethernet/IP connection settings, and finally double-click on the built-in Ethernet/IP port settings to enter the Ethernet/IP configuration interface.



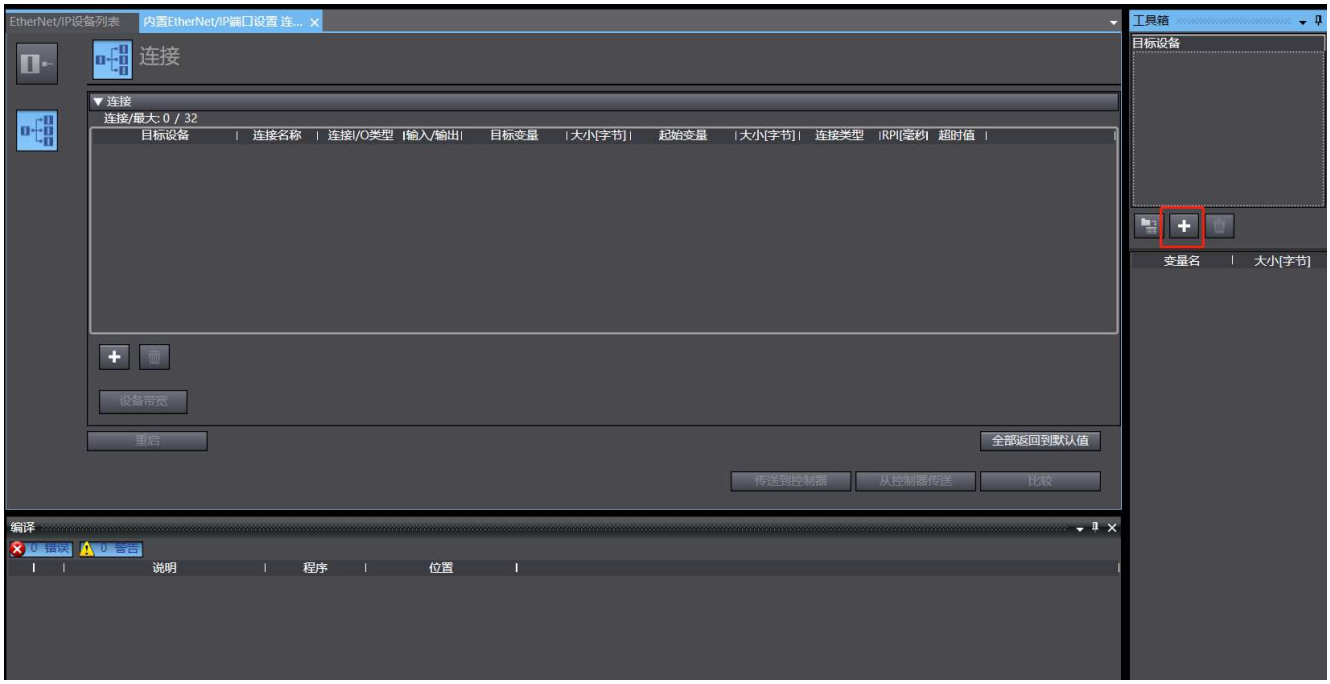
Step 4: Enter the built-in Ethernet/IP port setting operation page, select the label group operation page, and register the global variables for input and output in the relevant network public. You can click on input/output to view the registered variable information.

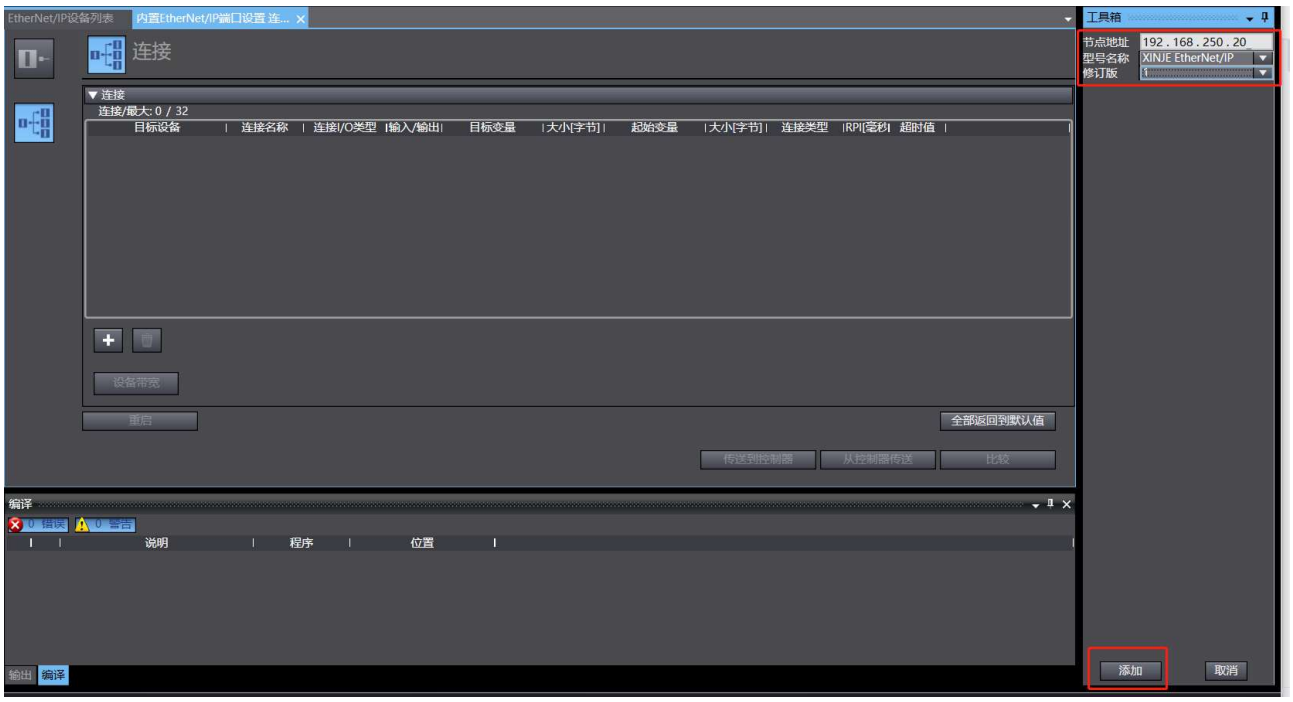


Step 5: Go to the built-in Ethernet/IP port setting operation page and select the connection operation page. Right click on the blank space in the toolbox on the right side of the connection operation page to enter the displayed EDS library. Add the EDS file corresponding to Xinje Ethernet/IP to it.

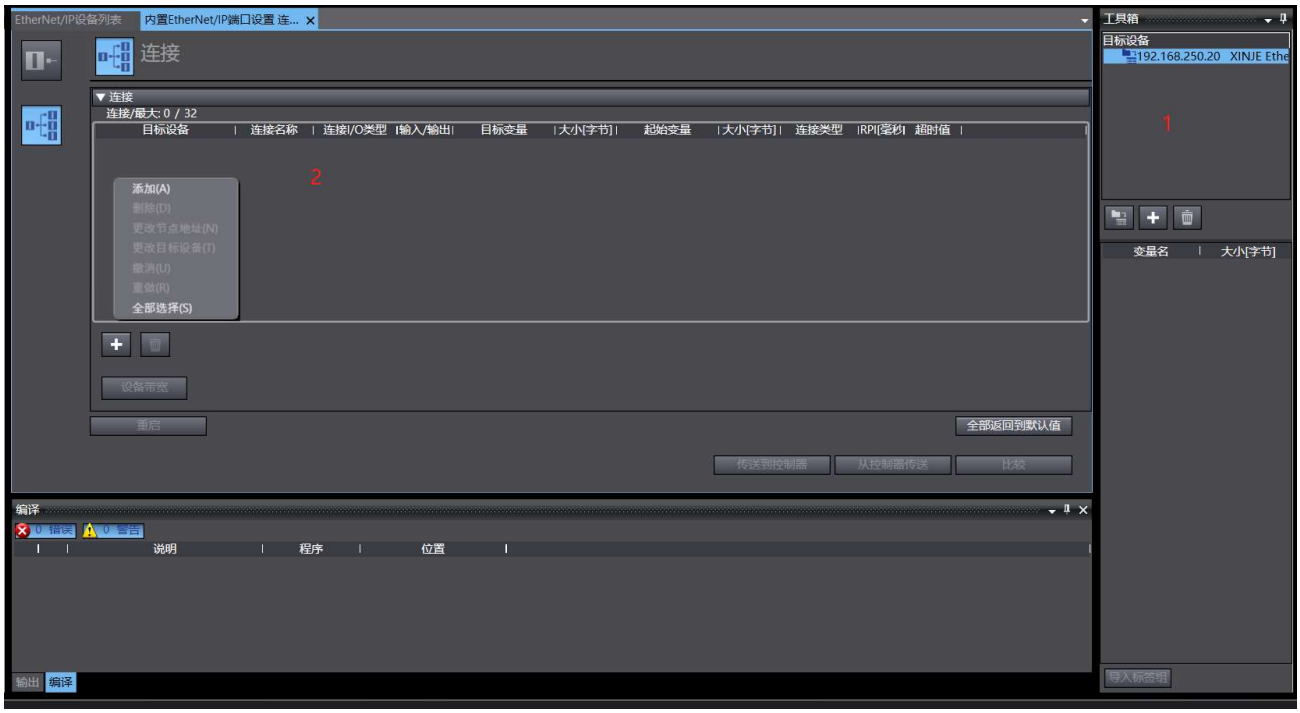


Step 6: Click the add button + in the toolbox on the right side of the connection operation page, and three operation requirements for adding objects will appear: node address (IP address of the object to be connected), model name (matching EDS file of the object to be connected), revised version (select the version of the EDS file of the connected object), and the operation is shown in the figure. After the establishment is completed, click the add button, and the addition and configuration information will be completed as shown in the figure below:

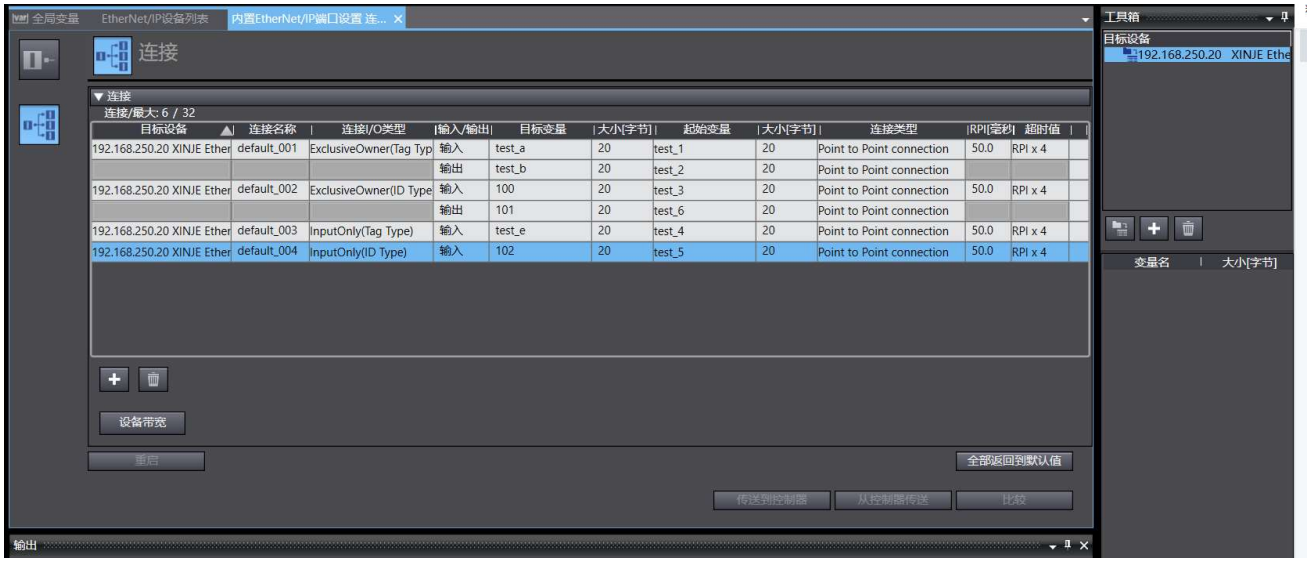




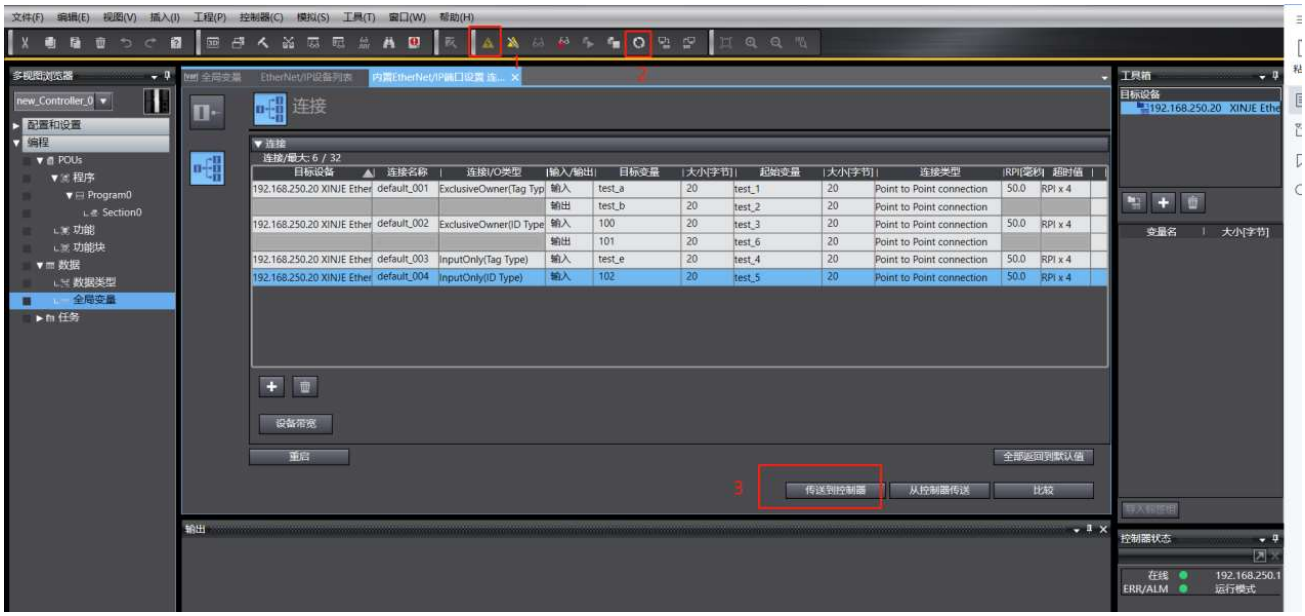
Step 7: Right click on the blank space in the toolbox or the blank space in the connection area to add a connection.



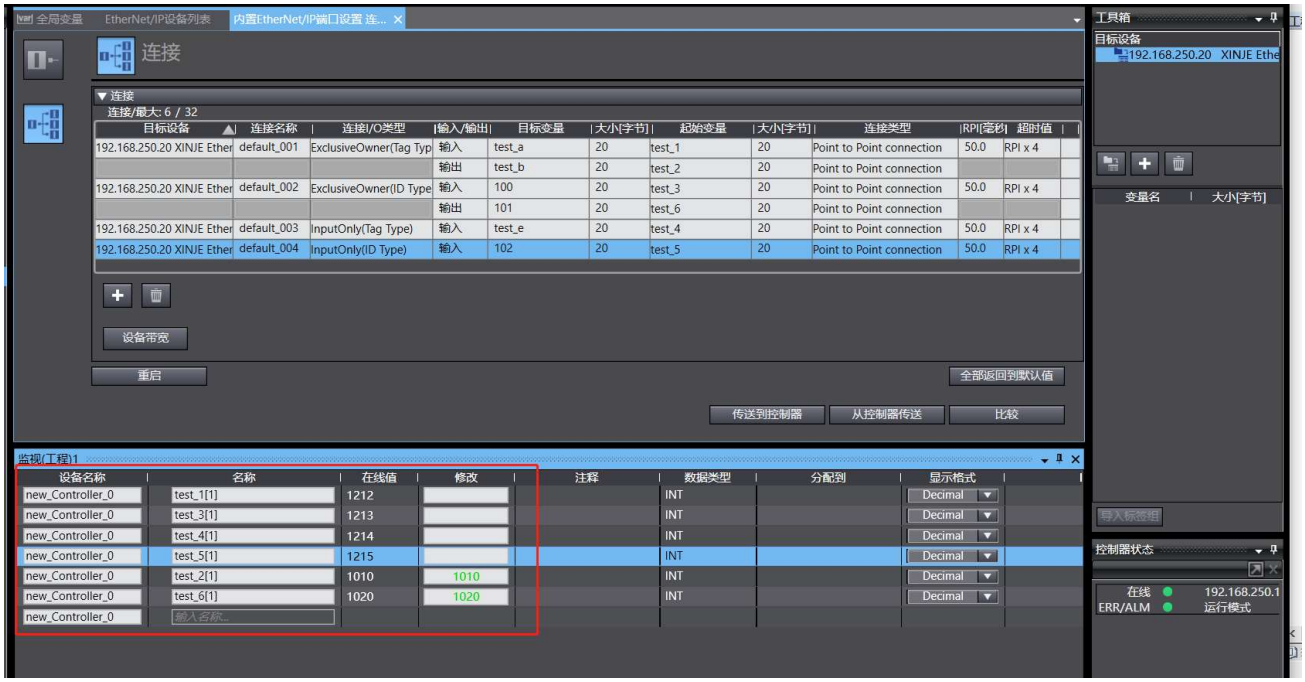
Step 8: Add ExclusiveOwner (Tag Type), ExclusiveOwner (ID Type), and IputOnIy (Tag Type) and IputOnIy (ID Type) connections, and communicate with tag variables or instance IDs respectively. The configured variable types are shown in the following figure:

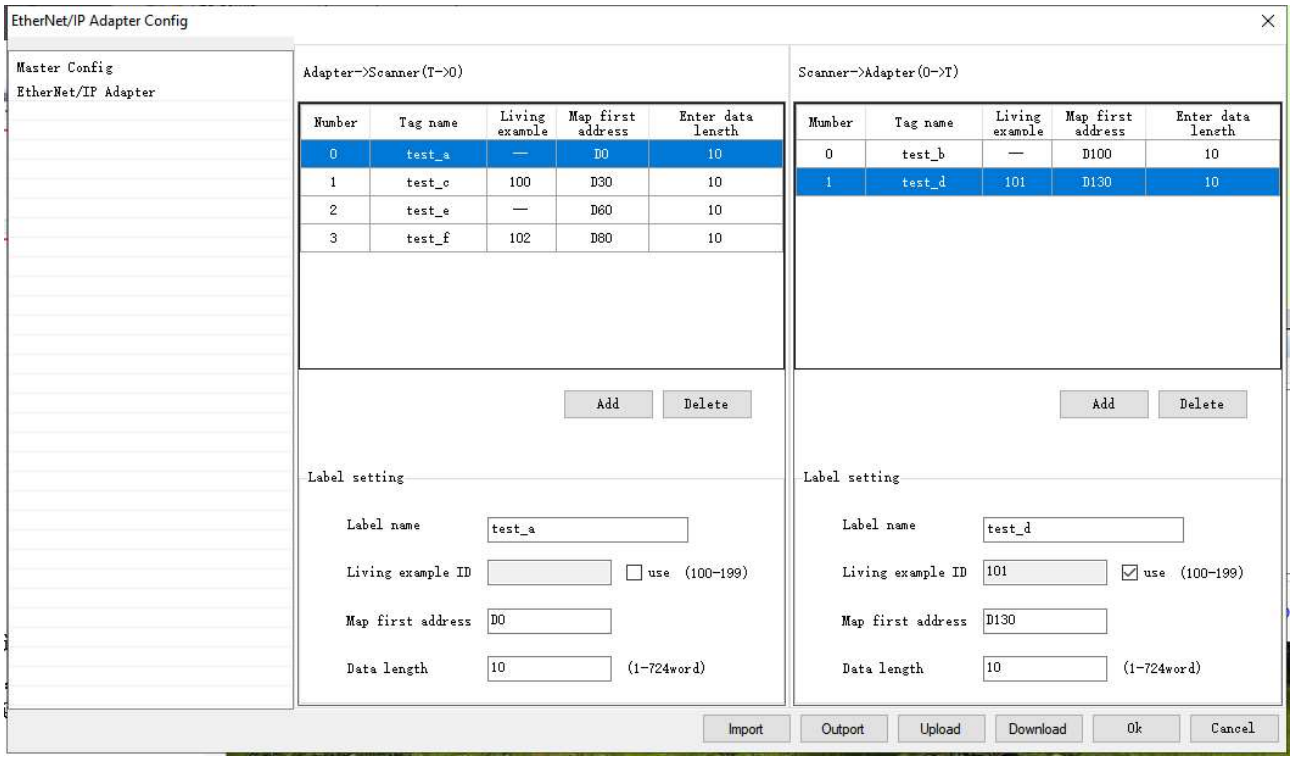


Step 9: After completing the information configuration, click on "online", then click on "synchronize" to download the project information to the controller, and finally click on "transfer to controller" to transfer the connection configuration information to the controller.



Step 10: Control the input and output data, and check whether the sending and receiving data is normal through the monitoring window.

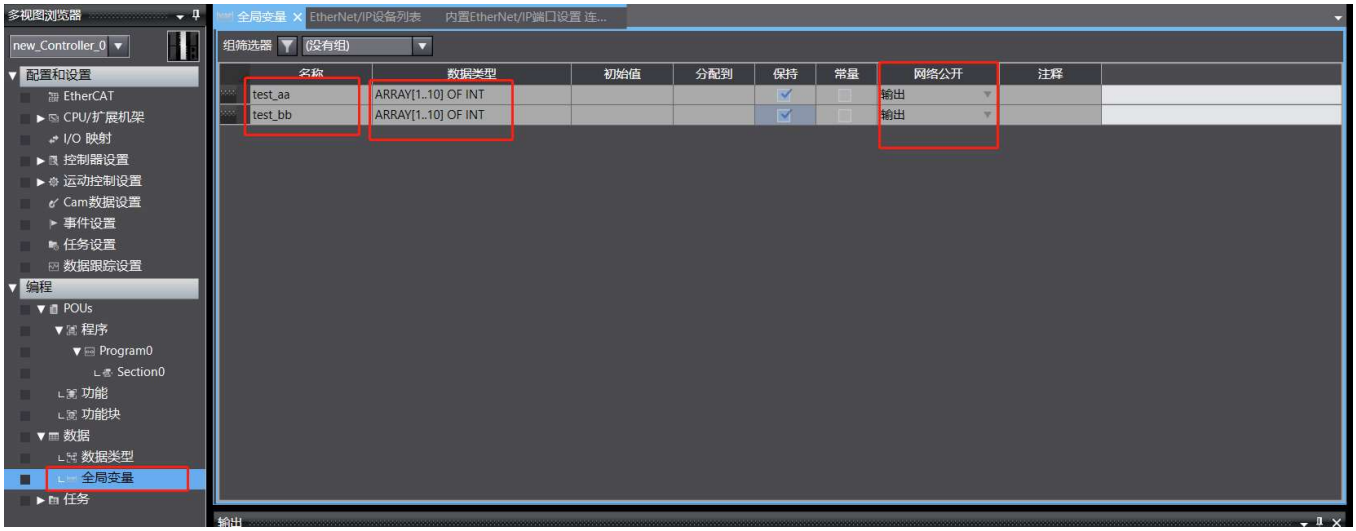




Application 5: Use Xinje PLC XDH-60T-E as the scanner and Omron PLC NJ501-1500 as the adapter for implicit communication.

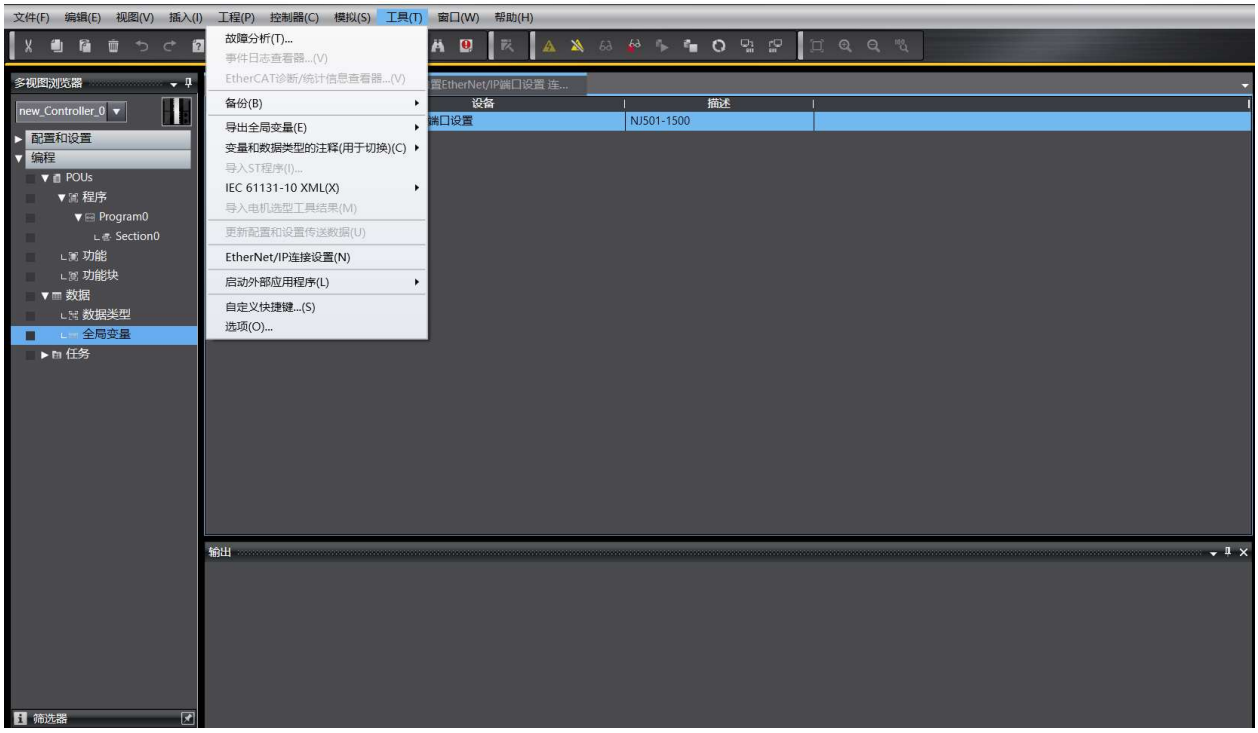
Use PLC1: XDH-60T-E (IP 192.168.250.20) as the scanner and PLC2: NJ501-1500 (IP 192.168.250.1) as the adapter to achieve implicit communication between two PLCs. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with the data size of the adapter.

Step 1: Define the variables that need to be used for EIP communication in the Omron Sysmac Studio programming software. During the EIP communication process, the global variables that need to be sent and processed with the communication object are selected as output in the network public.

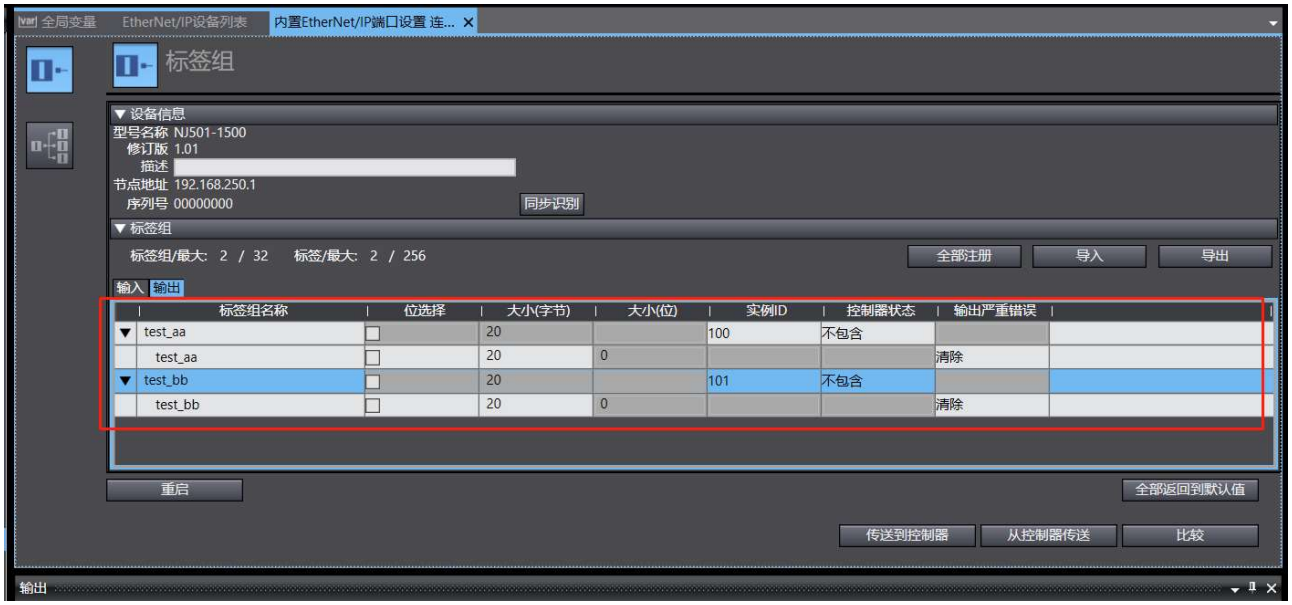


- Double click on the global variable to create a new one and add the variable type and data length to be transferred;
- Customize the name of the created variable;
- Define the data type and length based on the length of the output;
- Select the corresponding output type for the defined variable in the network public as needed.

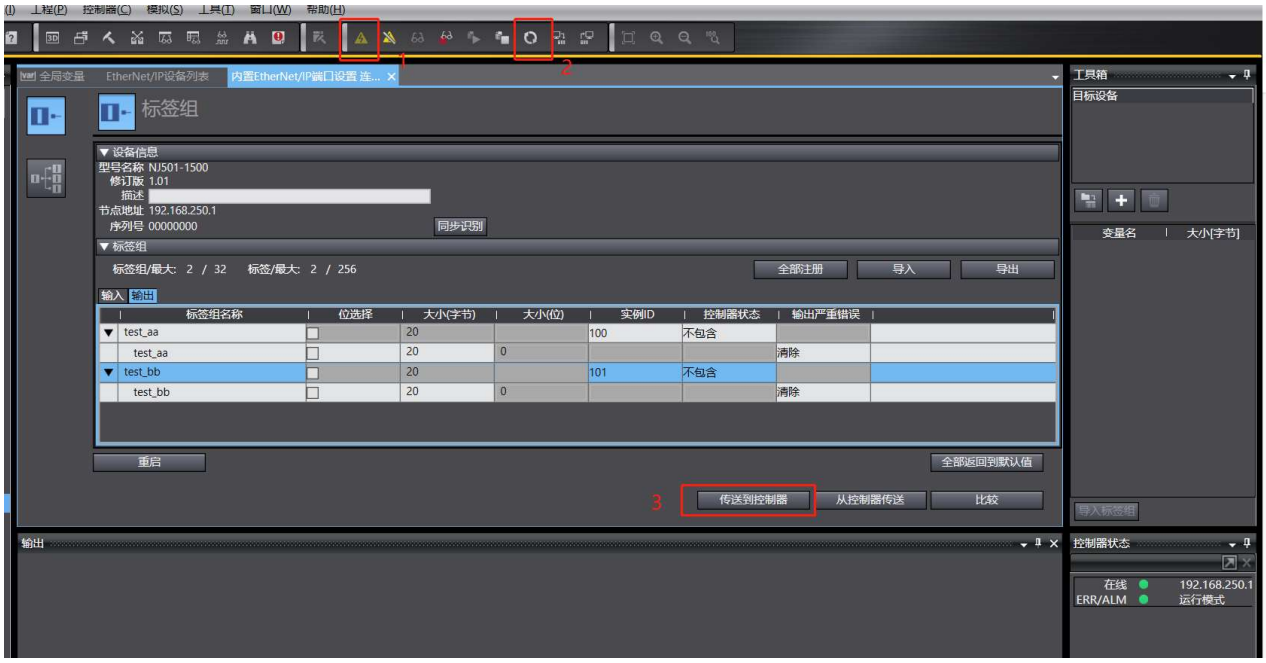
Step 2: Enter the Ethernet/IP connection settings operation page, click on Tools in the function bar, select and click on Ethernet/IP connection settings, and finally double-click on the built-in Ethernet/IP port settings to enter the Ethernet/IP configuration interface.



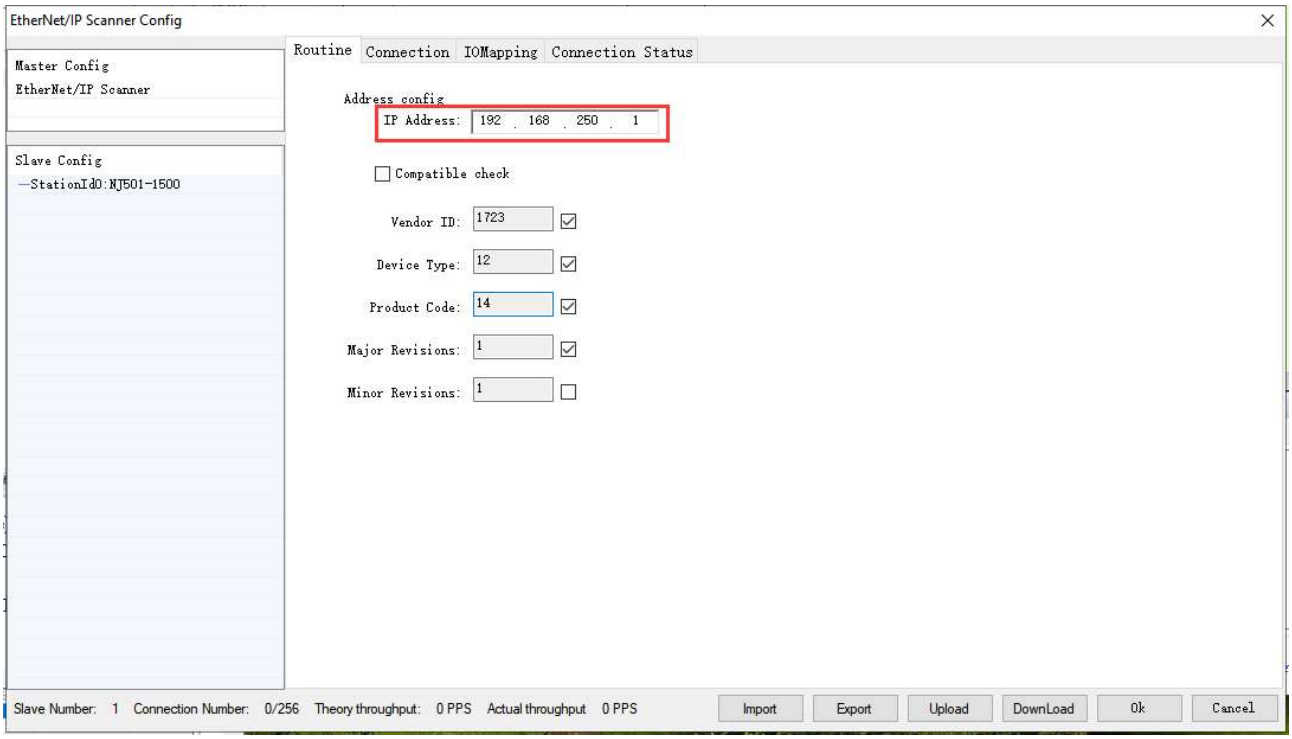
Step 3: Go to the built-in Ethernet/IP port setting operation page, select the label group operation page, and register the output global variables in the relevant network public. You can click on the output to view the registered variable information.



Step 4: After completing the information configuration, click on "online", then click on "synchronize" to download the project information to the controller, and finally click on "transfer to controller" to transfer the connection configuration information to the controller.

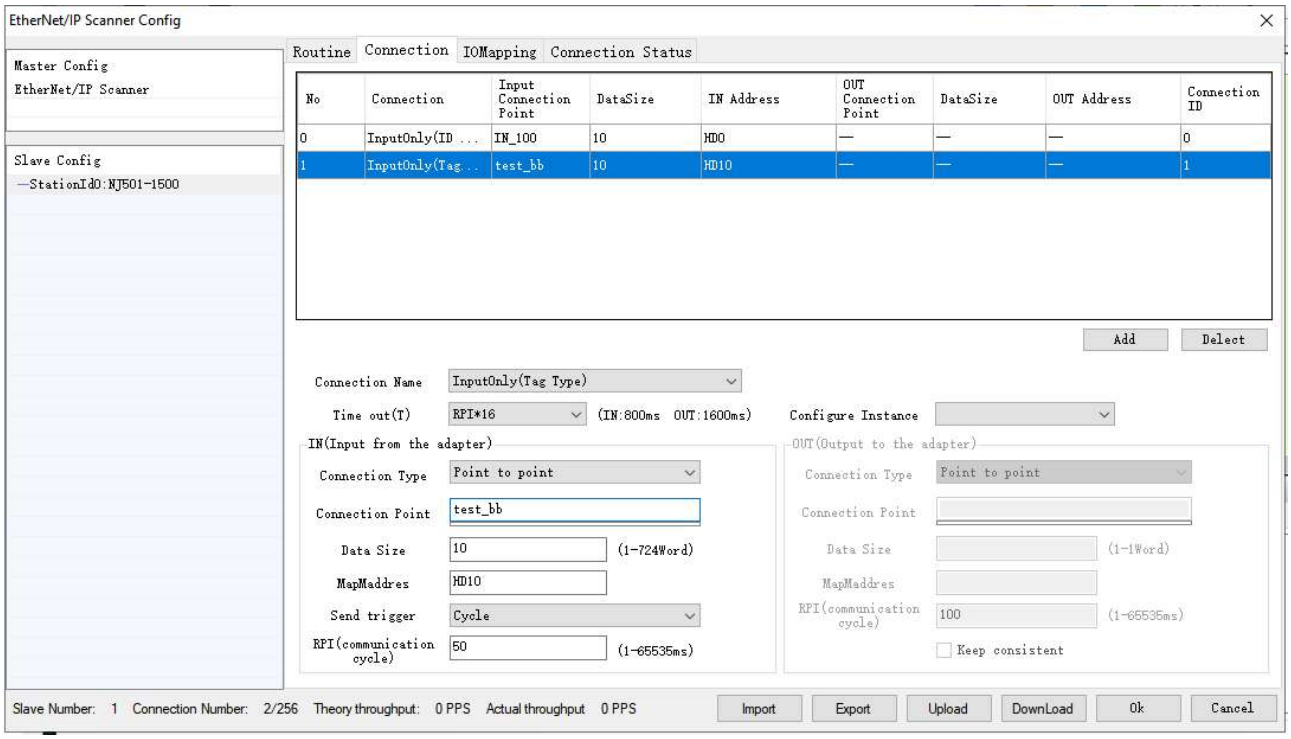


Step 5: In the Xinje XDPPRO programming software scanner, add the Omron NJ501-1500 slave device and perform relevant configuration operations on the adapter's IP address and compatibility check:

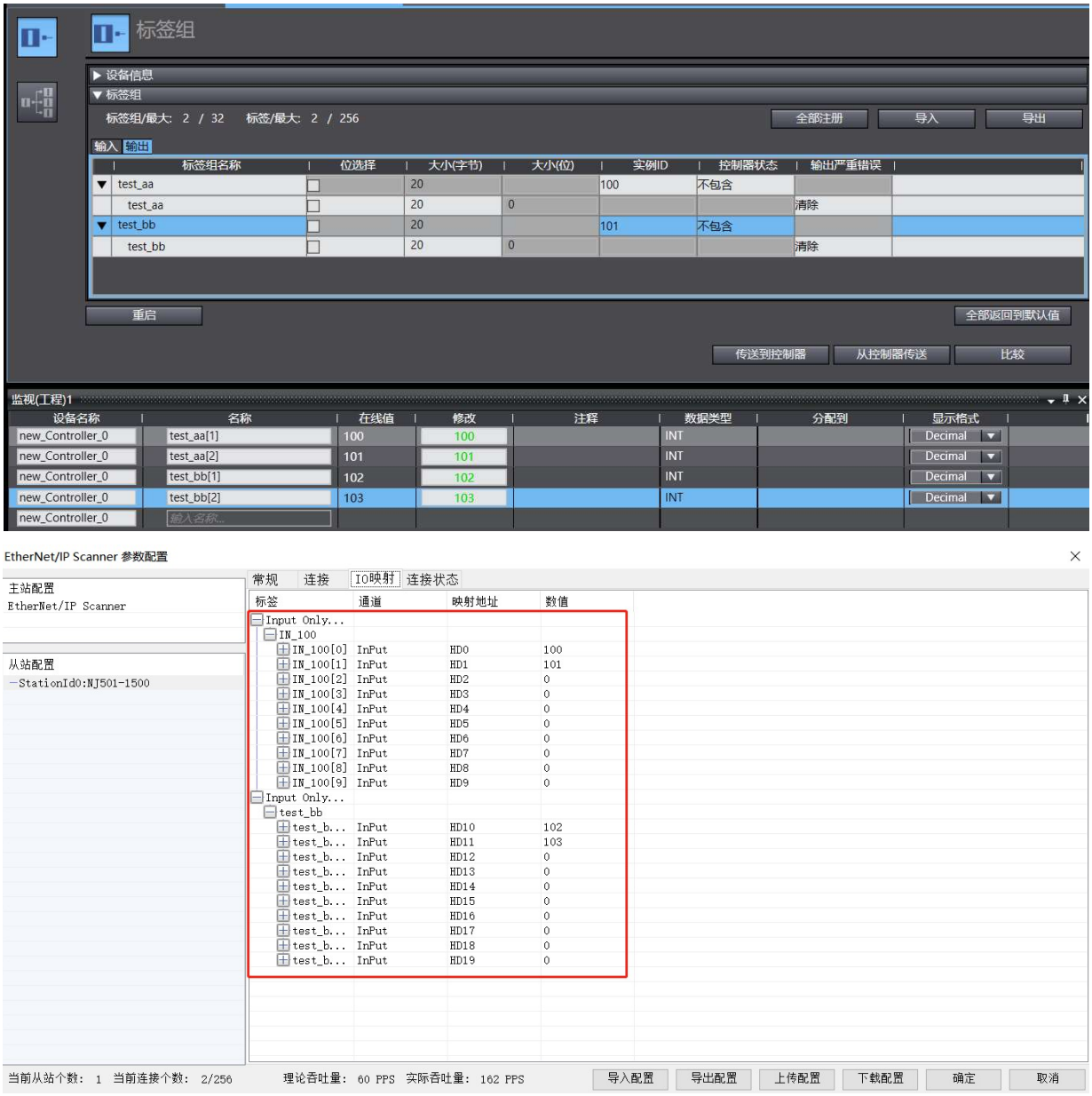


Step 6: Click on the connection to add two types of connections, Input Only (ID Type) and Input Only (tag Type). Connection 1: label name test_bb, data length 10, mapping first address HD10. Connection 2: instance ID (IN_100), input data length 10, mapping first address HD10.

After configuration is completed, click download to download configuration to the PLC. The configured variable types are shown in the following figure:



Step 7: Click on IO mapping or connection status to view the current communication data or connection status.



Application 6: Use Xinje PLC XDH-60T-E as the adapter and Keyence PLC KV-5500 as the scanner for implicit communication.

PLC1: XDH-60T-E (IP 192.168.6.6) as the adapter and PLC2: KV-5500 (IP 192.168.6.10) as the scanner, implicit communication between two PLCs can be achieved. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with that of the adapter.

Step 1: Add three connections in the adapter XDH-60T4-E in the direction of Adapter->Scanner(T->O).

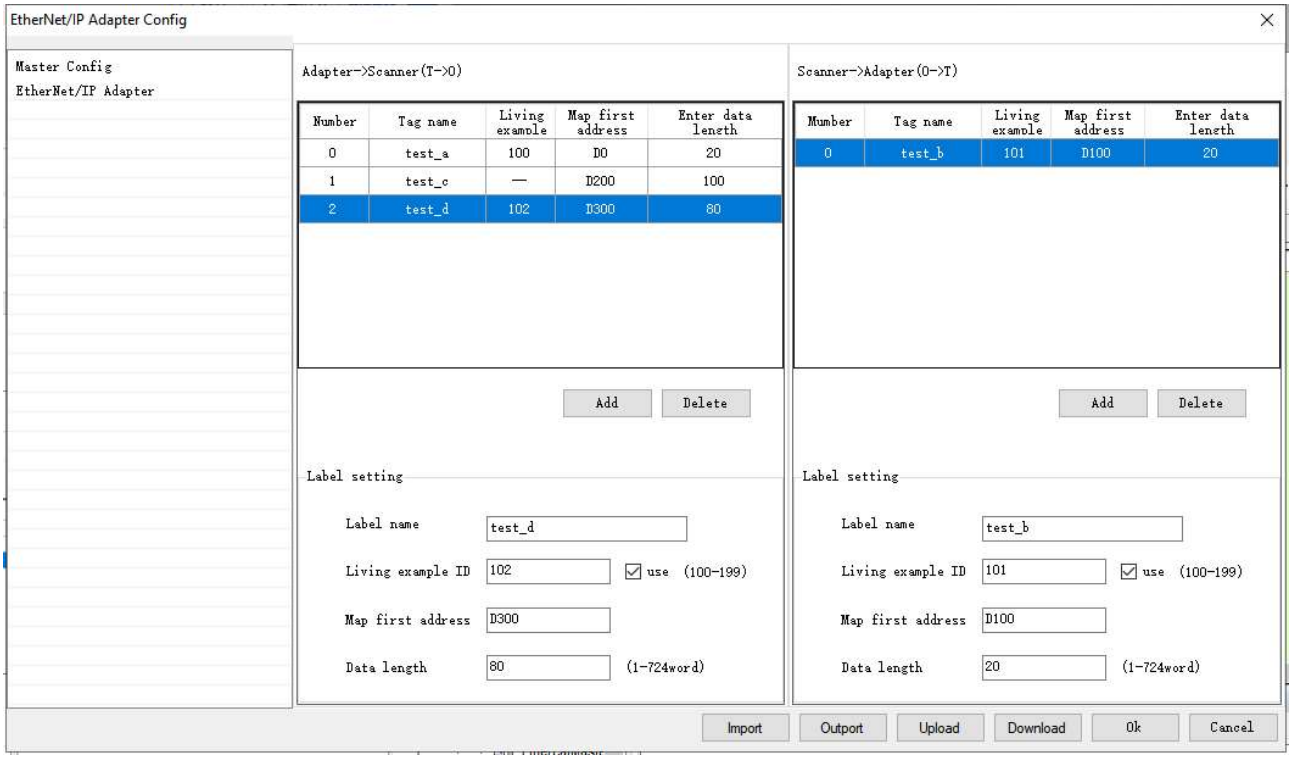
Connection 1: instance ID100, label name test_a, mapping first address D0, data length 20.

Connection 2: label name test_c, mapping first address D200, data length 100.

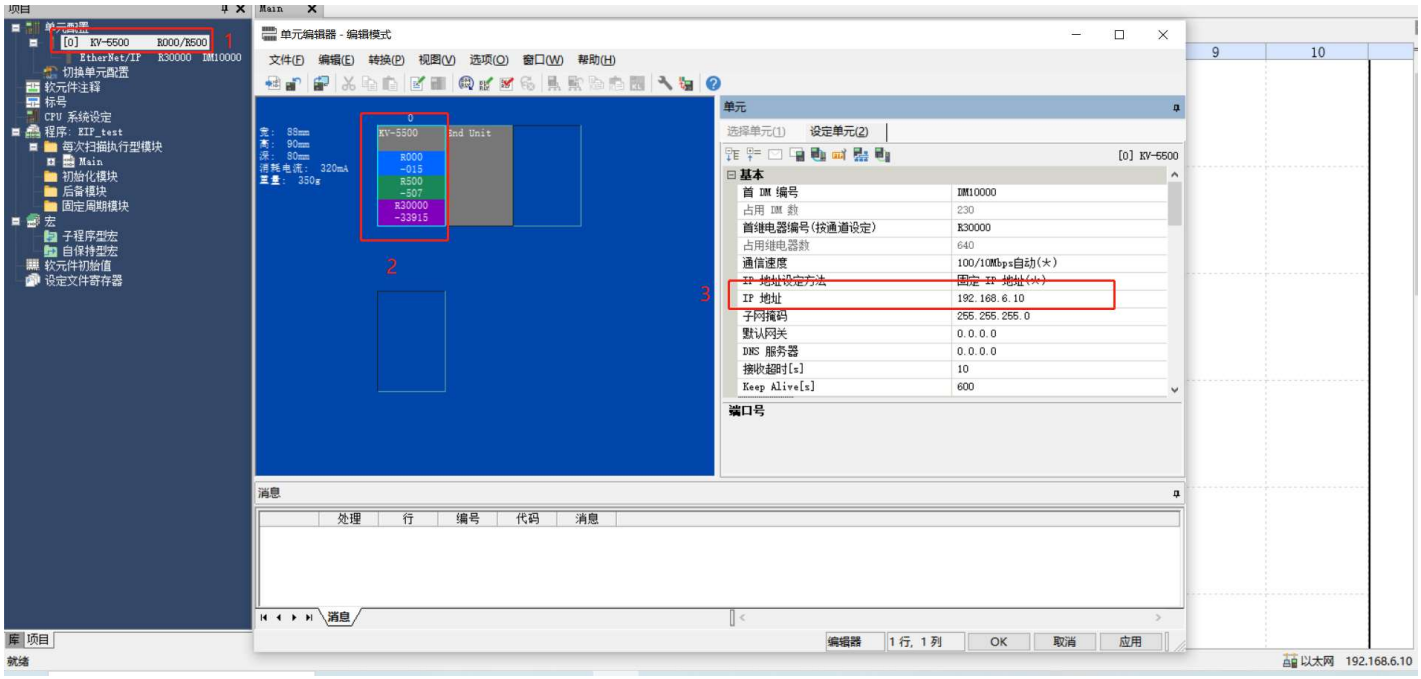
Connection 3: instance ID102, label name test_d, label name test_d, mapping first address D300, data length 80.

Add a connection in the direction of Scanner->Adapter(O->T), instance ID101, label name test_b, mapping first

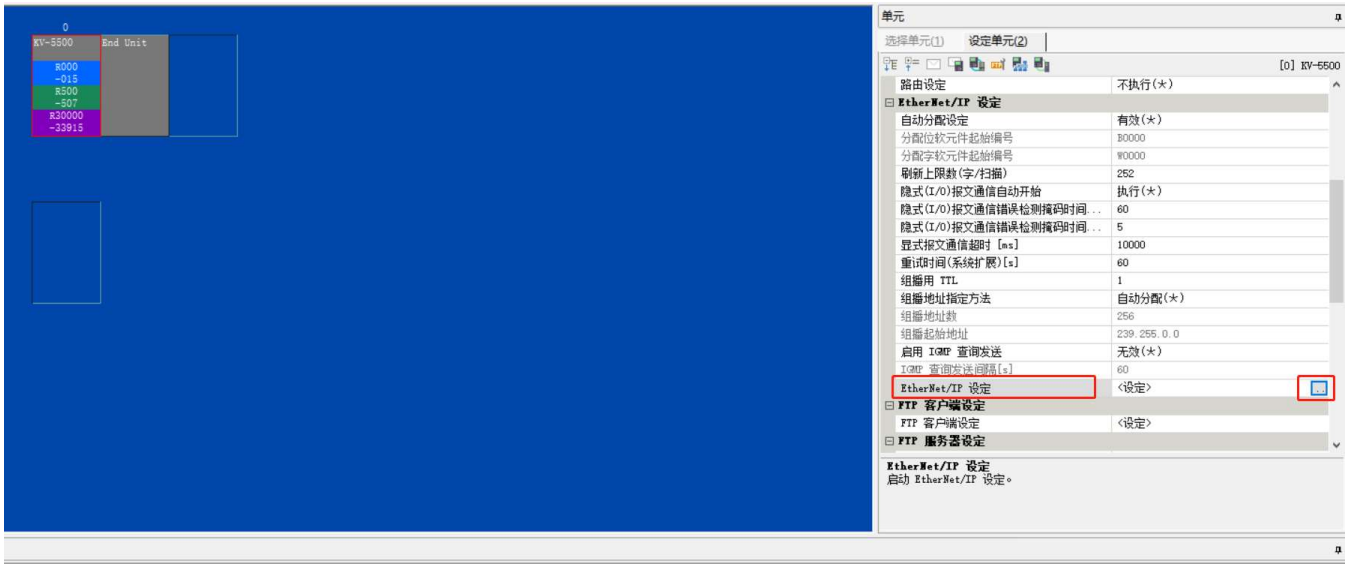
address D100, data length 20.

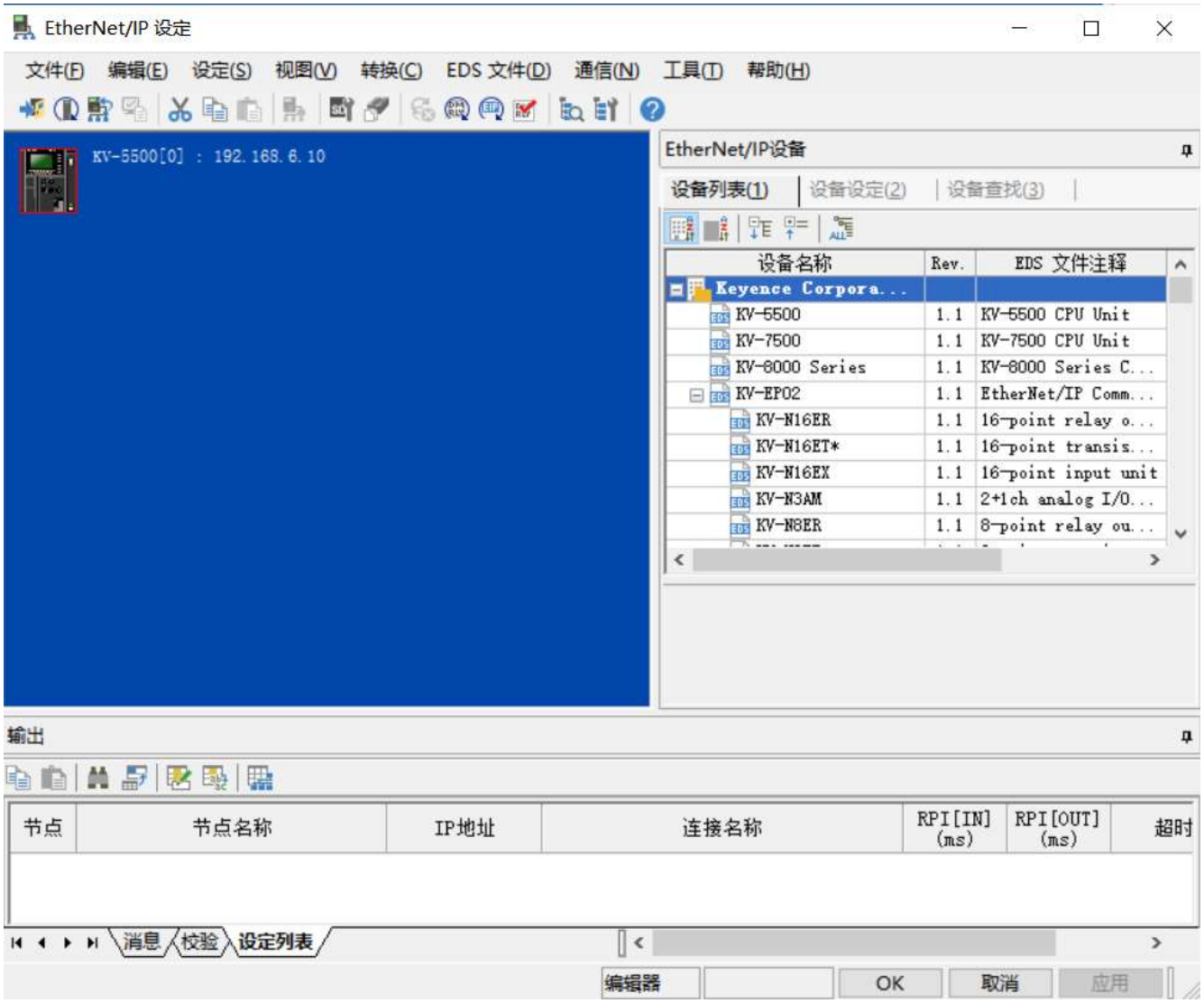


Step 2: In the Keyence KV STUDIO programming software, after connecting to the PLC to be communicated, double-click KV-5500 under the unit configuration to enter the unit editor - edit mode. Double click the CUP unit to configure its IP address, ensuring that it is in the same network segment as the adapter.

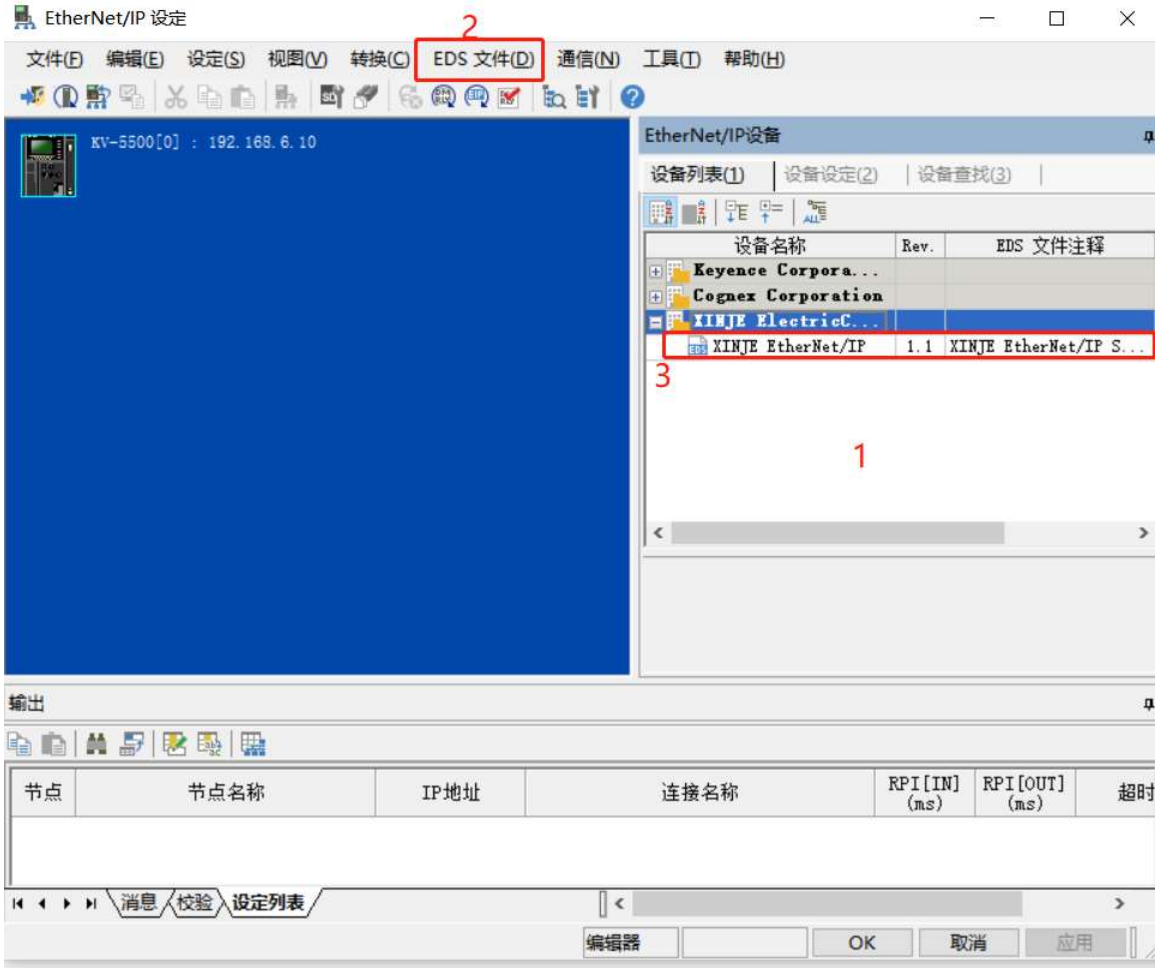


Step 3: In the programming mode of the unit editor, find the Ethernet/IP settings, click the function key on the right side of the Ethernet/IP settings, and enter the Ethernet/IP settings configuration interface.

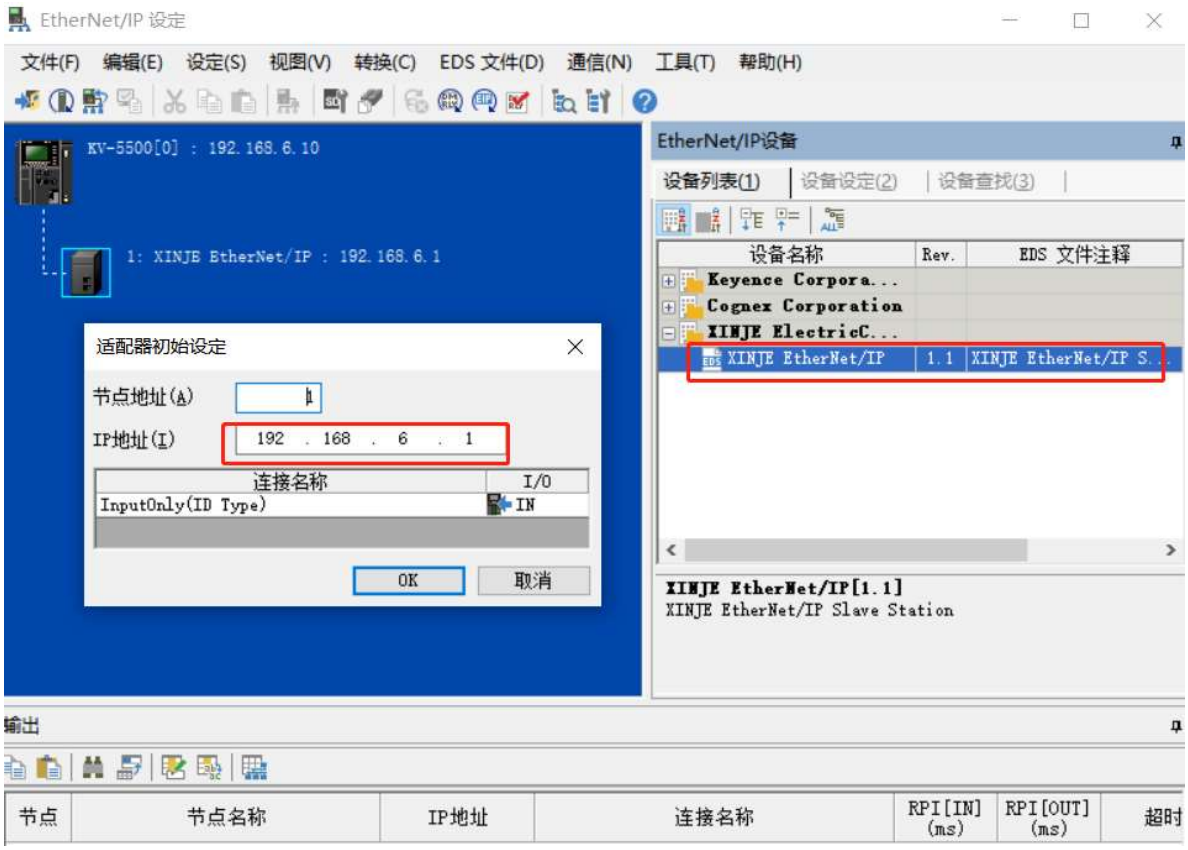




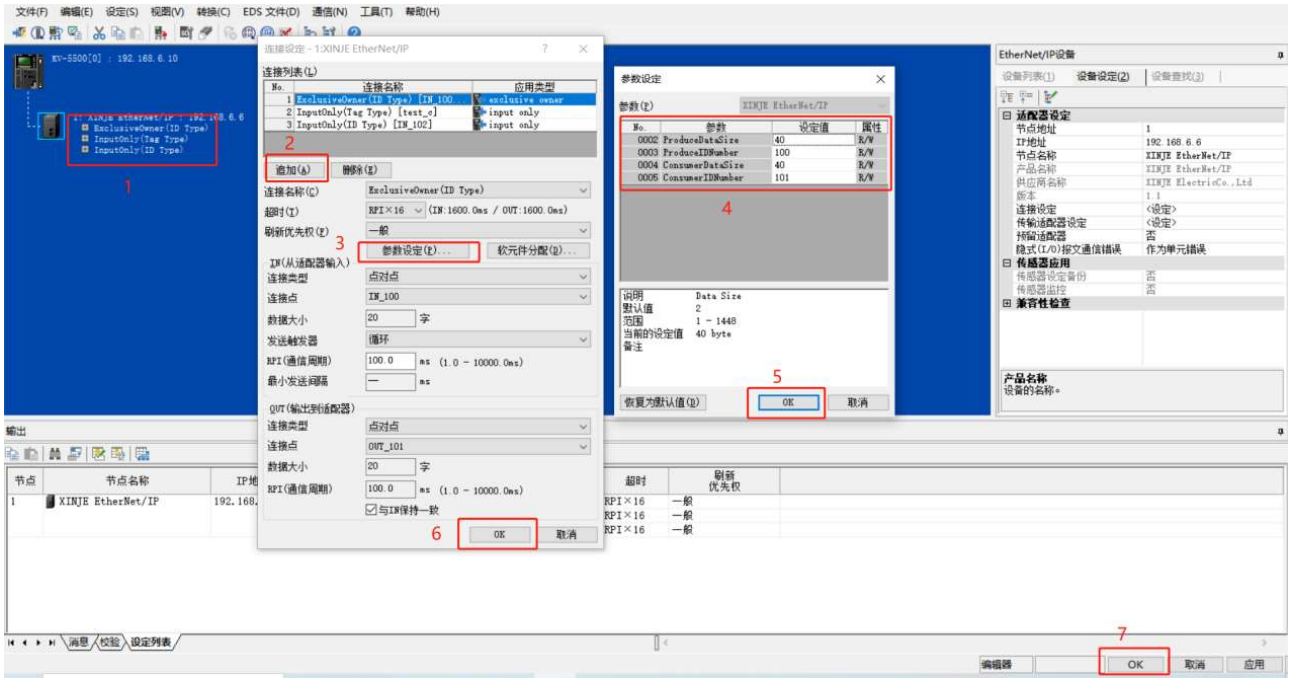
Step 4: Right click on the blank space of "EtherNet/IP Device" or "EDS File (D)" on the function bar to add XDH-60T4-E as the EDS file for the adapter. After adding, you can view the corresponding XINJE EtherNet/IP EDS file in "EtherNet/IP Device".



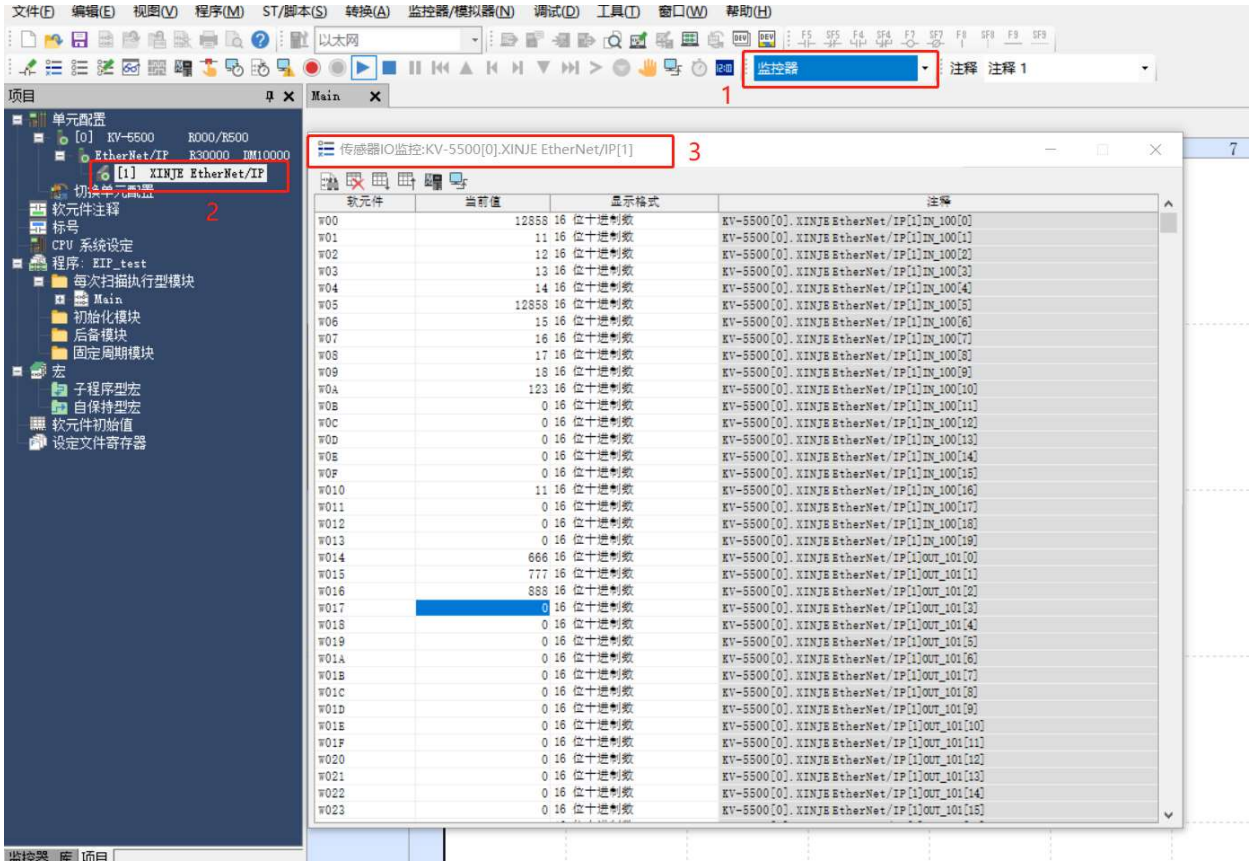
Step 5: Double click on the XINJE EtherNet/IP EDS file to use it as an adapter, and configure its adapter IP address in the adapter initial settings dialog box.



Step 6: Click the "+" corresponding to the added adapter to enter the connection setting configuration interface. Click "Add" in the configuration interface to add the specified connection name according to the application type. Select the corresponding connection name and click on parameter settings. According to the size of the adapter's configured data, configure the data size in the scanner accordingly. After completing the configuration, click OK to complete the parameter configuration. Finally, click Download to download the configuration information to the PLC controller.



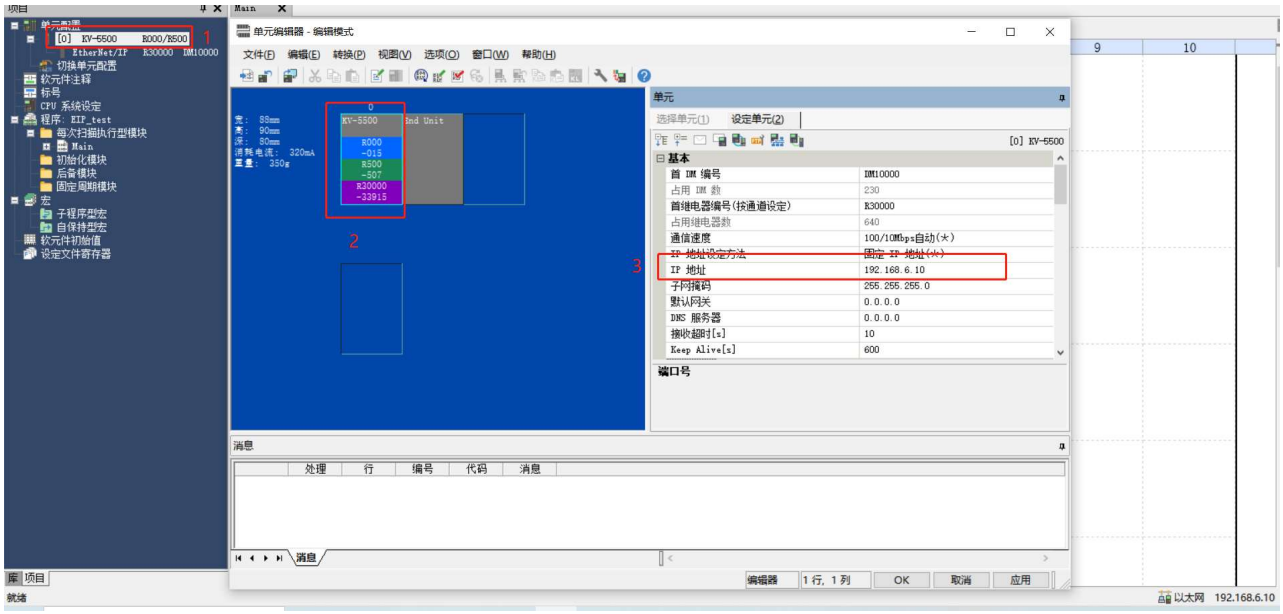
Step 7: Double click XINJE Ethernet/IP in monitor mode, operate and monitor the corresponding data in the IO monitoring table, and verify whether communication is normal.



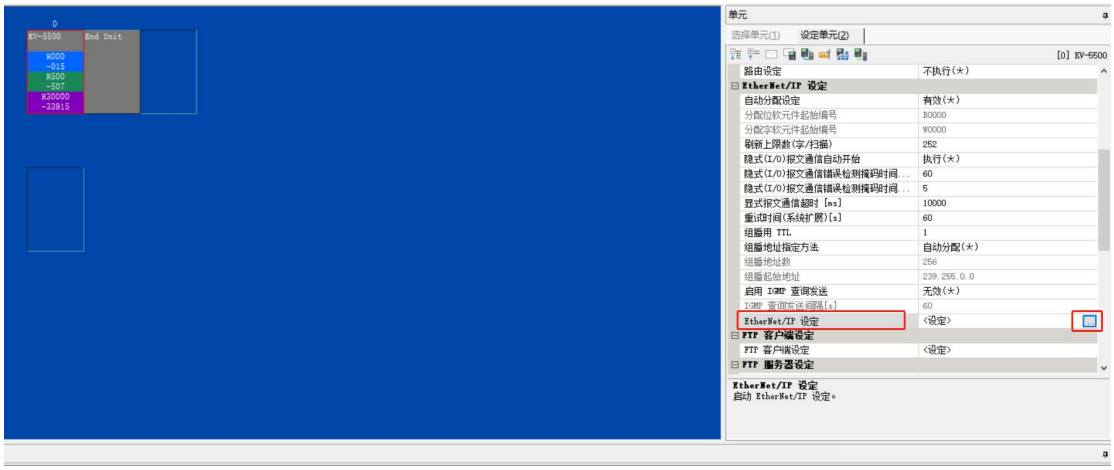
Application 7: Use Xinje PLC XDH-60T-E as the scanner and Keyence PLC KV-5500 as the adapter for implicit communication.

PLC1: XDH-60T-E (IP 192.168.6.6) as the scanner and PLC2: KV-5500 (IP 192.168.6.10) as the adapter, implicit communication between two PLCs can be achieved. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with that of the adapter.

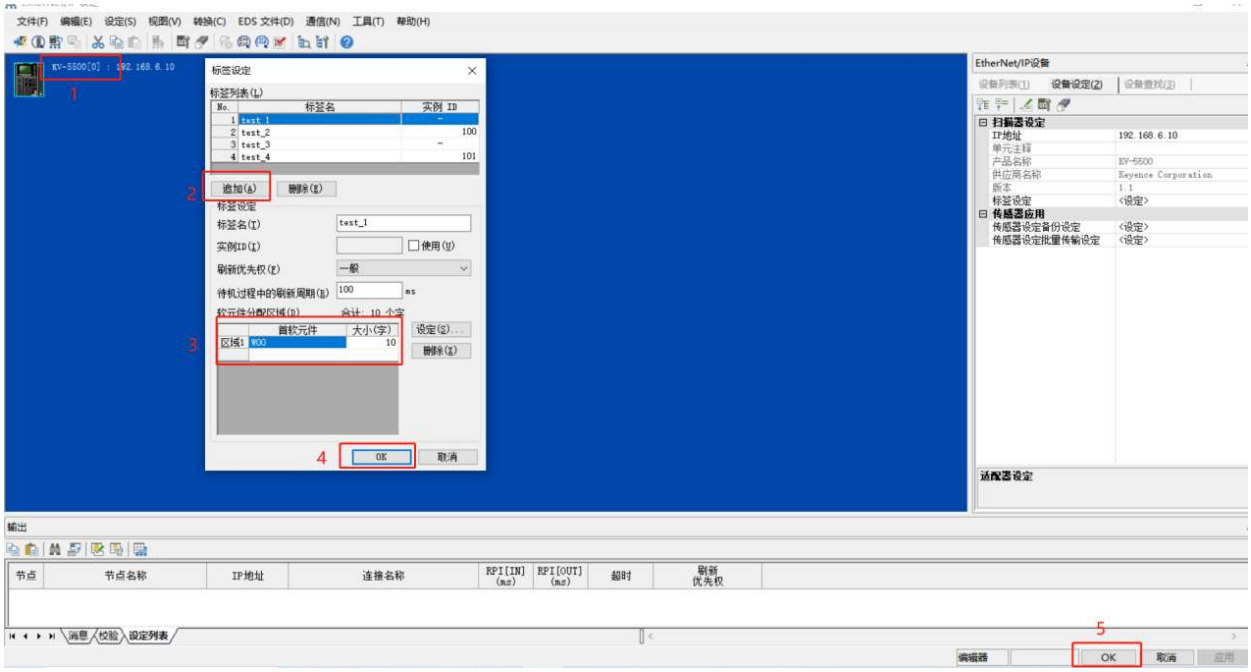
Step 1: In the Keyence KV STUDIO programming software, after connecting to the PLC to be communicated, double-click KV-5500 under the unit configuration to enter the unit editor - edit mode. Double click the CUP unit to configure its IP address, ensuring that it is in the same network segment as the scanner.



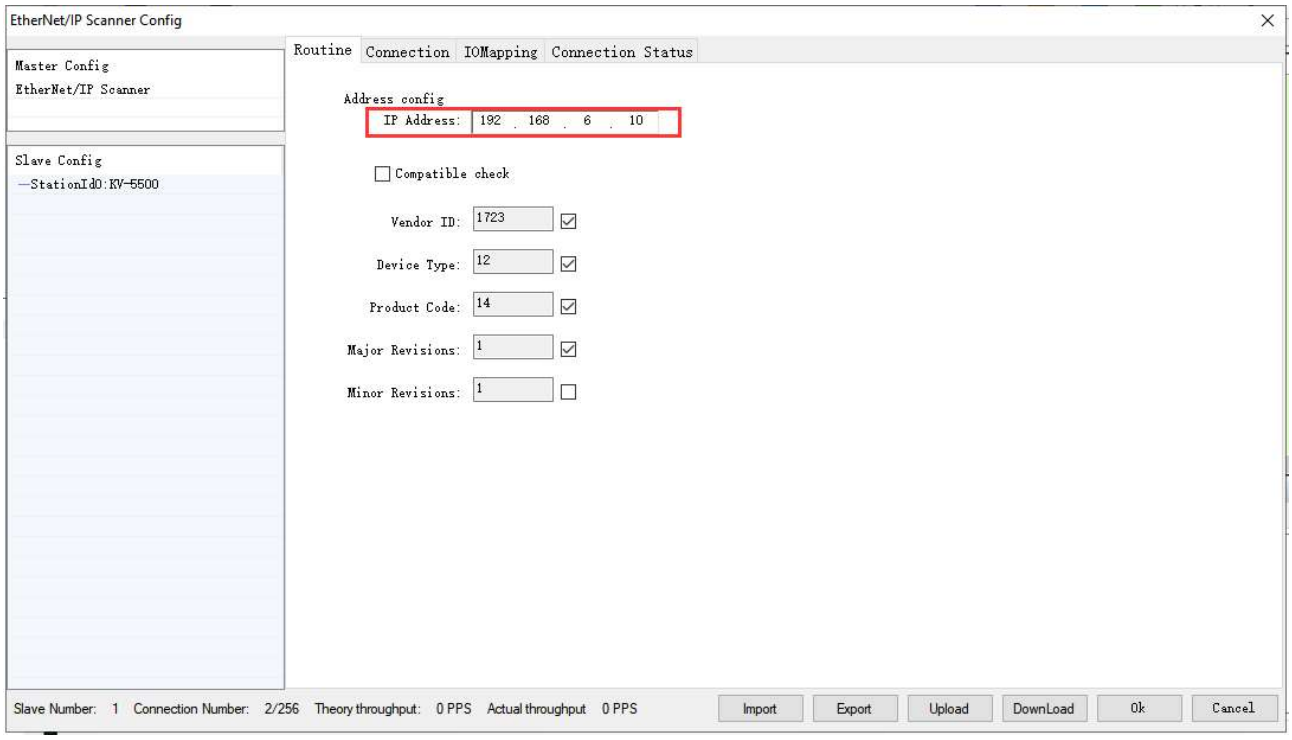
Step 2: In the programming mode of the unit editor, find the Ethernet/IP settings, click the function key on the right side of the Ethernet/IP settings, and enter the Ethernet/IP settings configuration interface.



Step 3: Double click KV-5500 to enter the label setting interface, click add to add label connection, configure the data size corresponding to the added label, click OK to complete the corresponding information configuration, and finally click download to download the configuration information to the PLC controller.

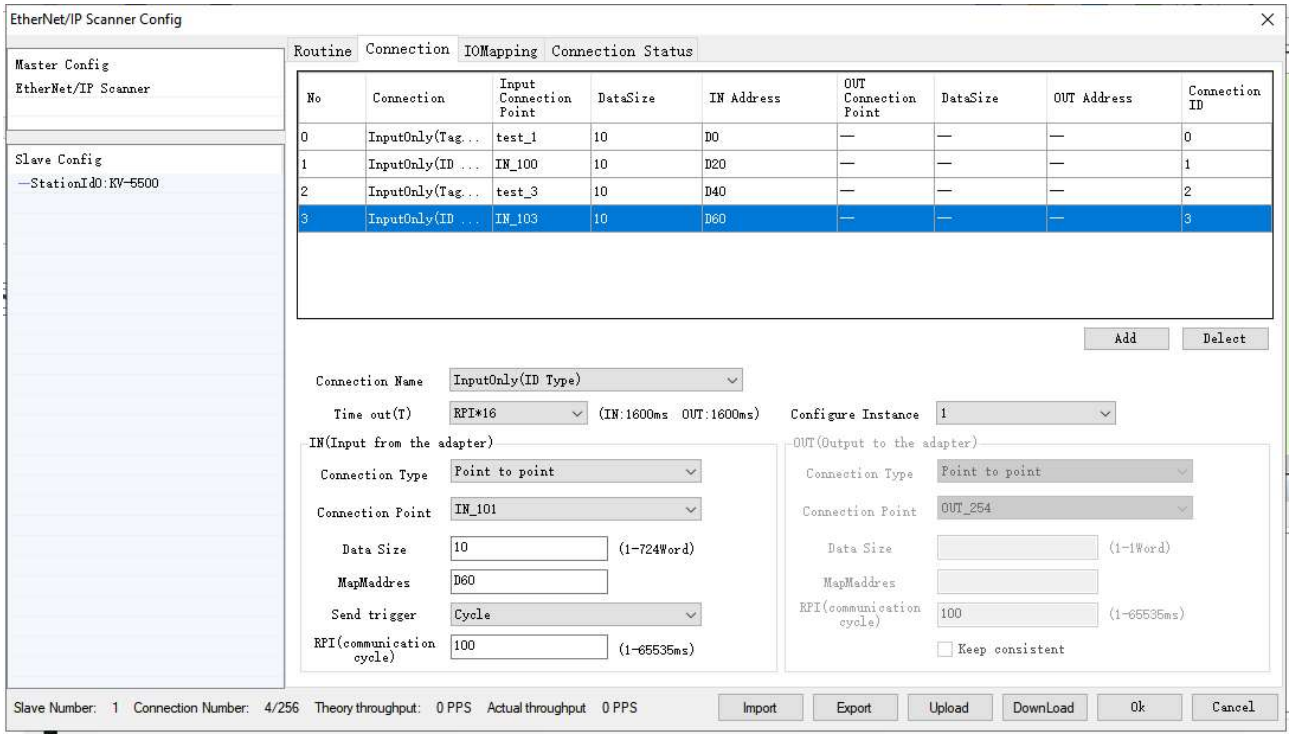


Step 4: In the Xinje XDPPRO programming software scanner, add the Omron KV-5500 slave device and perform relevant configuration operations on the adapter's IP address and compatibility check:

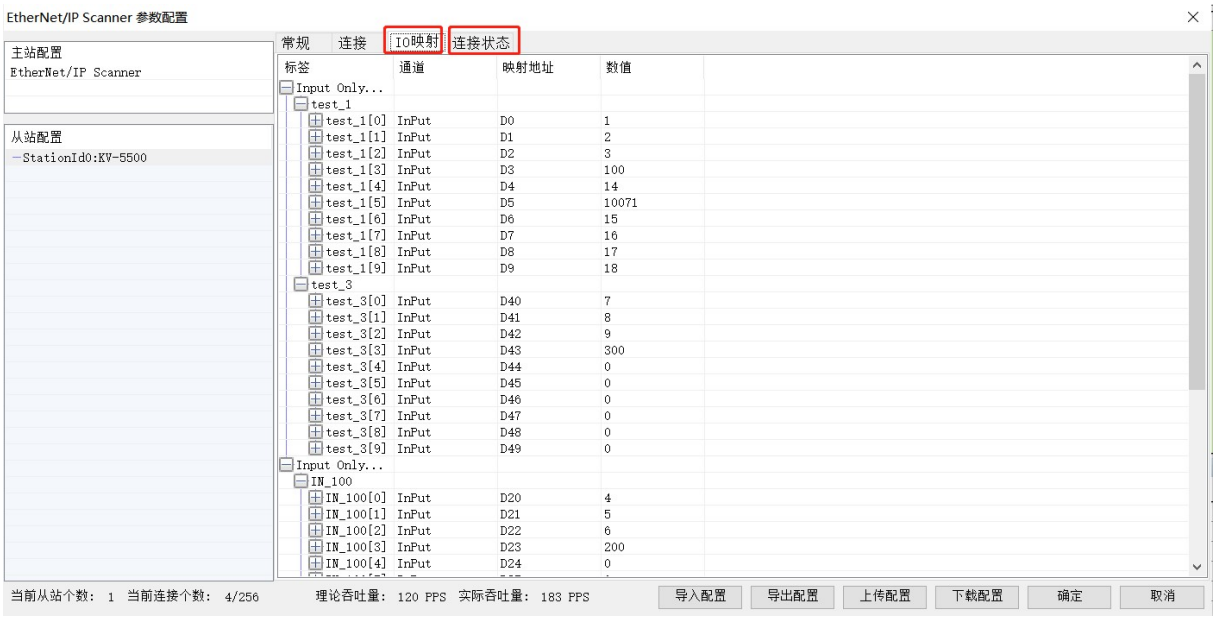


Step 5: Add two types of connections: Input Only(ID Type) and Input Only(tag Type).
Connection 1: label name test_1, data length 10, mapping first address D0.
Connection 2: instance ID (IN_100), data length 10, mapping first address D20.
Connection 3: label name test_3, data length 10, mapping first address D40.
Connection 4: instance ID (IN_101), data length 10, mapping first address D60.
After completing the configuration, click download to download the configuration to the PLC. The variable

types configured are shown in the following figure:



Step 6: Click on IO mapping or connection status to operate and monitor corresponding data, and verify whether communication is normal.



Application 8: Implicit communication between Xinje PLC XDH-60T4-E as adapter and Xinje PLC XSDH-60A32-E as scanner.

PLC1: XDH-60T-E (IP 192.168.6.6) as the adapter and PLC2: XSDH-60A32-E (IP 192.168.6.200) as the scanner to achieve implicit communication between two PLCs. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with the data size of the adapter.

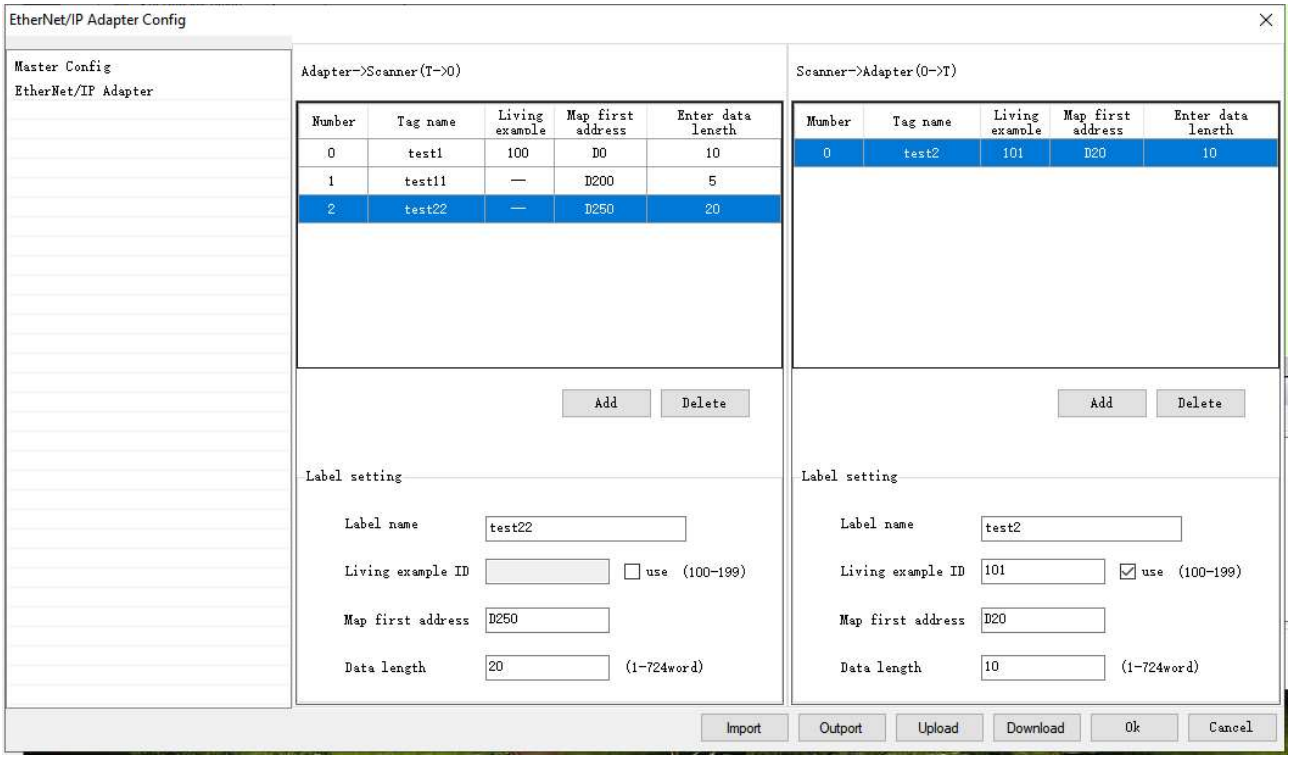
Step 1: Add three connections in adapter XDH-60T4-E in the direction of Adapter->Scanner(T->O).

Connection 1: instance ID100, label name test1, mapping first address D0, data length D10.

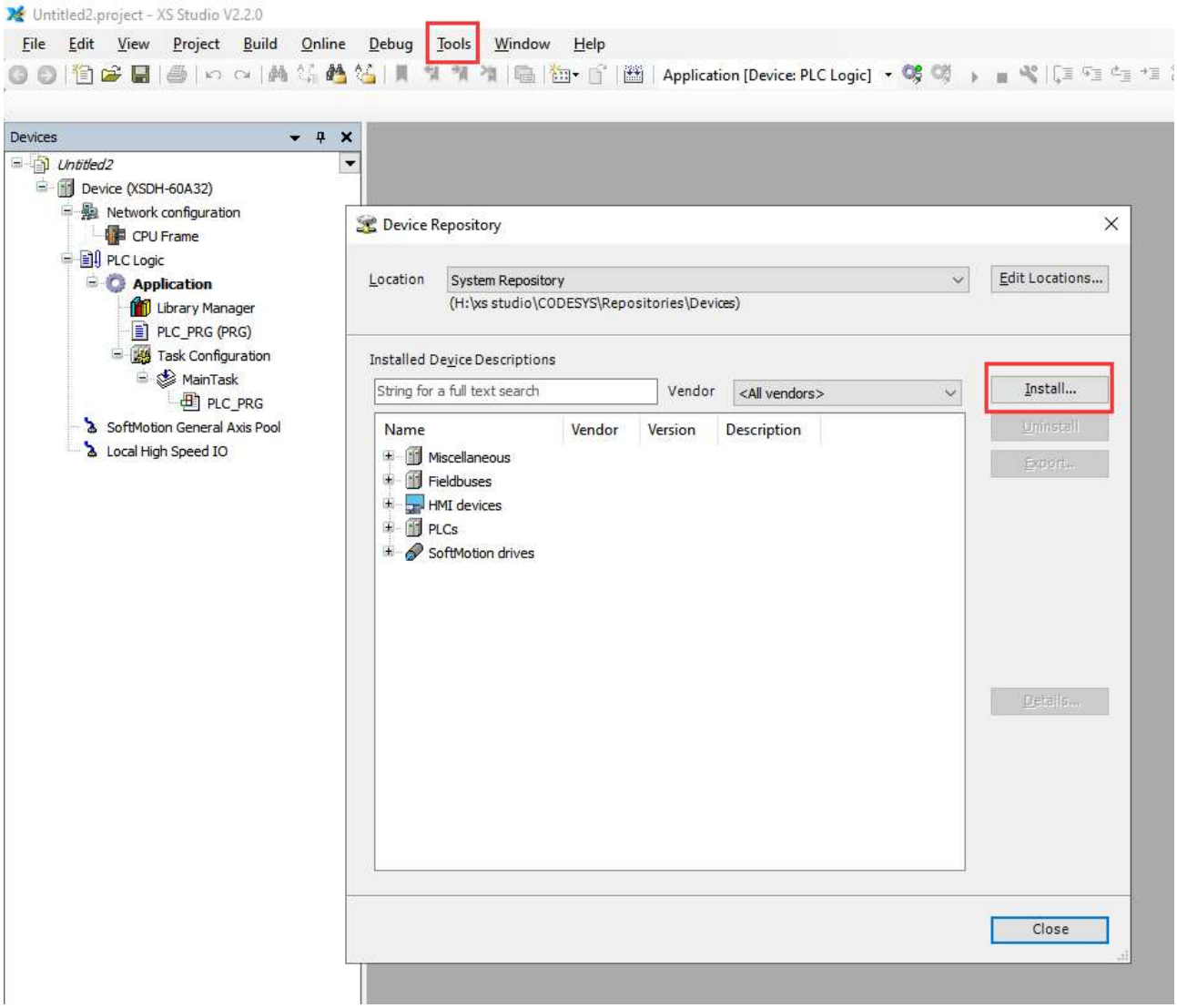
Connection 2: label name test11, mapping first address D200, data length 5.

Connection 3: label name test22, mapping first address D250, data length 20.

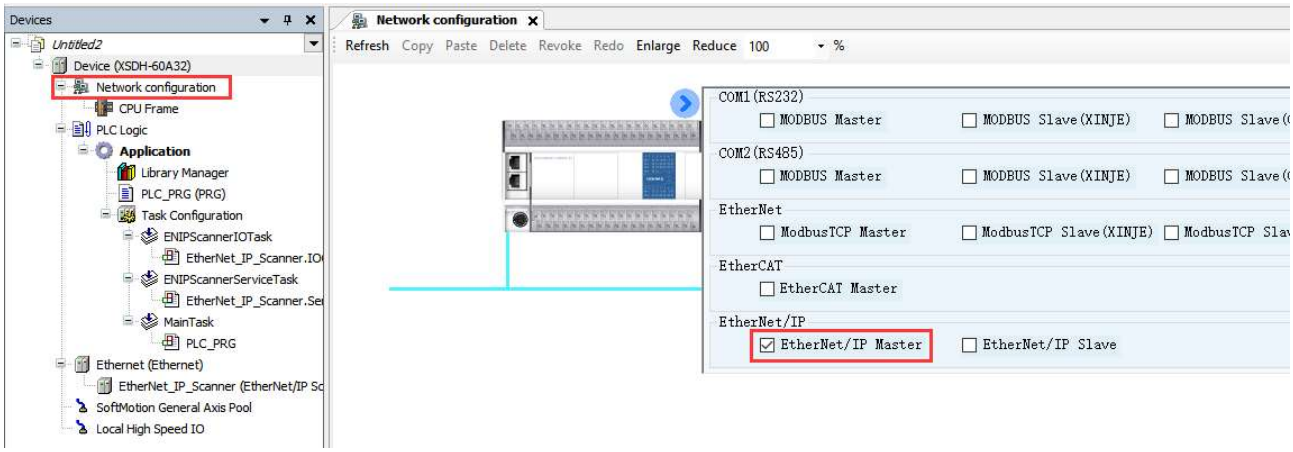
Add one connection in the direction of Scanner->Adapter(O->T). Instance ID101, label name test2, mapping first address D20, data length 10.

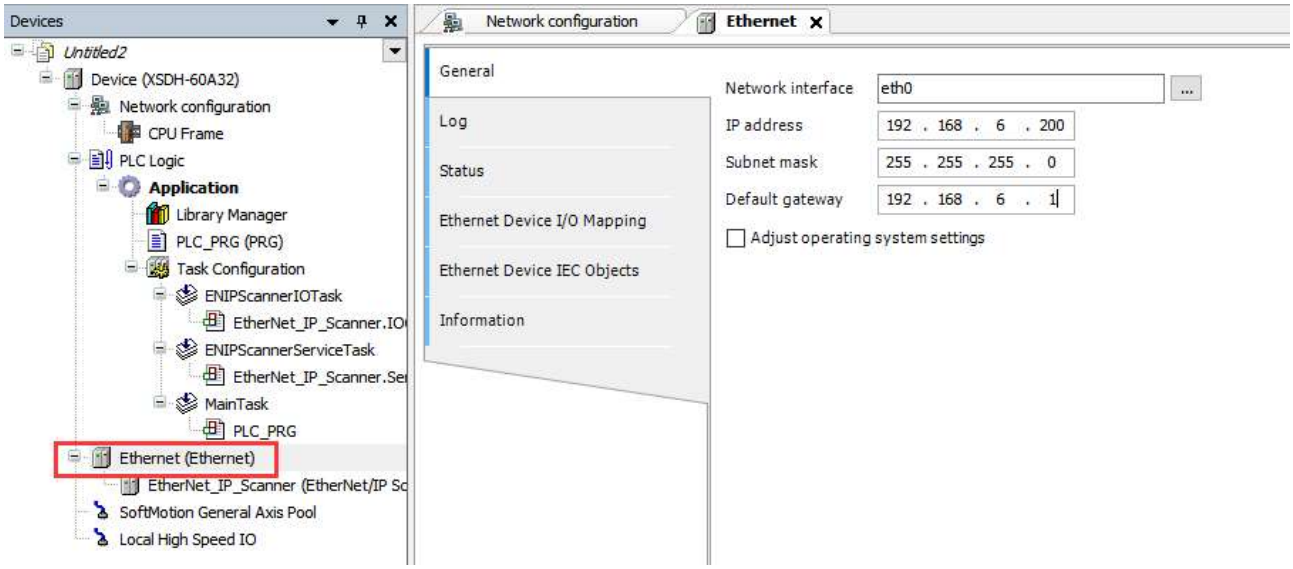


Step 2: In the XS Studio programming software, click on the tool to import the EDS file as an adapter.

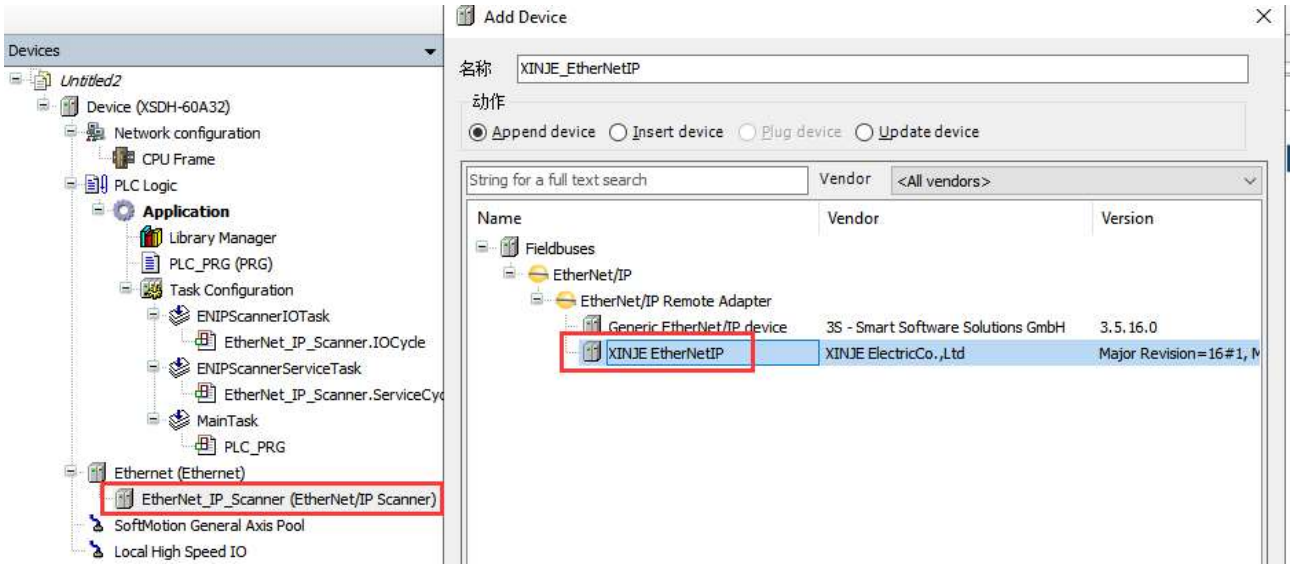


Step 3: Click on Network Configuration, add an EthernetIP master station in the network configuration, create an Ethernet-IP_Scanner, double-click Ethernrt to select the network card to use.

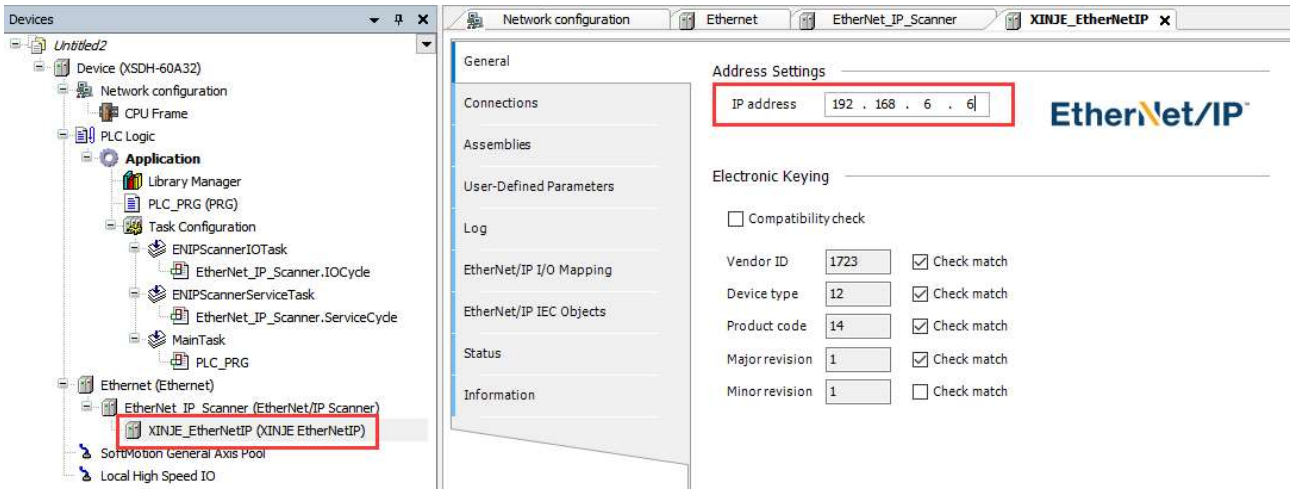




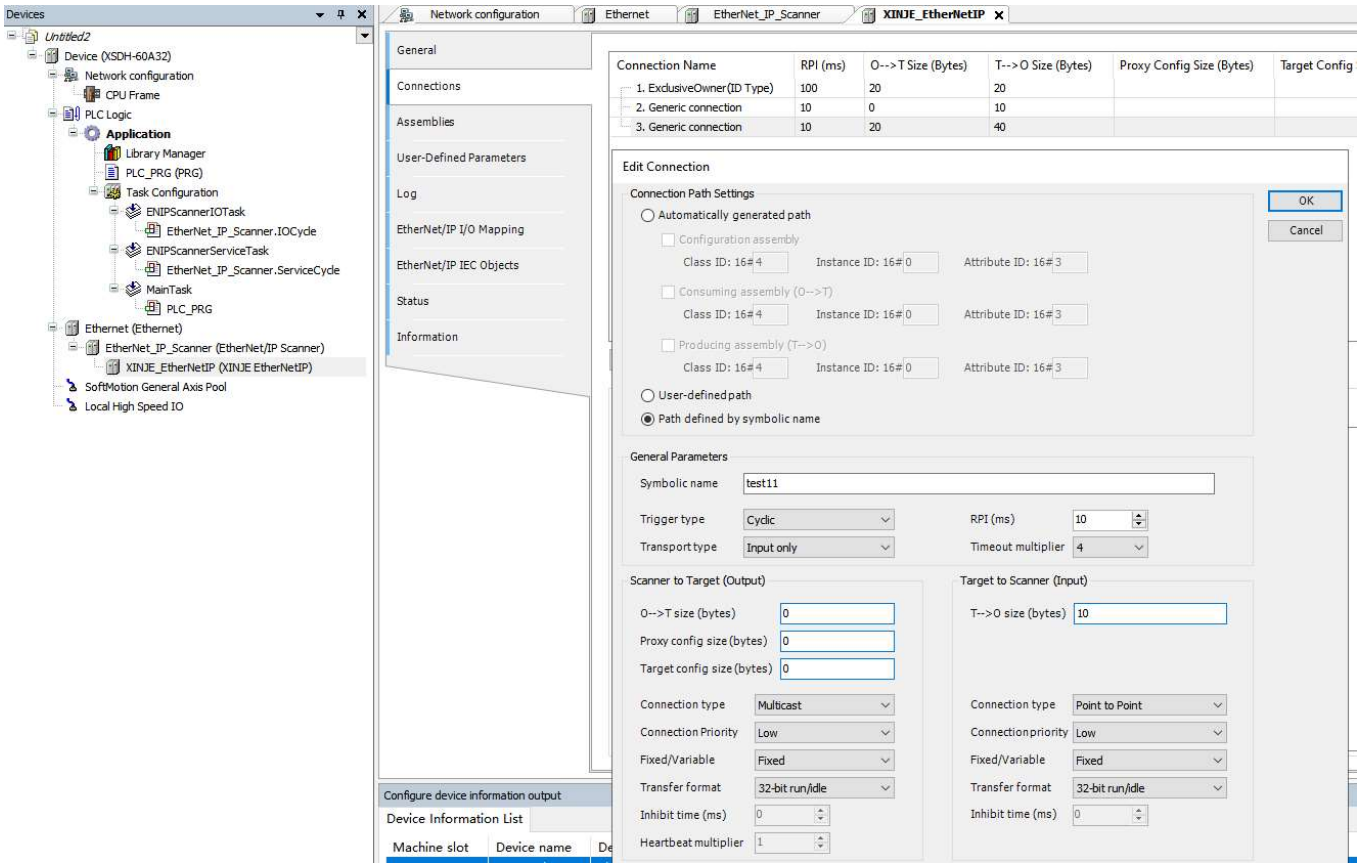
Step 4: Right click on Ethernet_IP_Scanner to add the corresponding XINJE EtherNetIP slave device.



Step 5: Double click on XINJE EtherNetIP to enter the corresponding configuration interface, click on General to configure the IP address of the adapter.

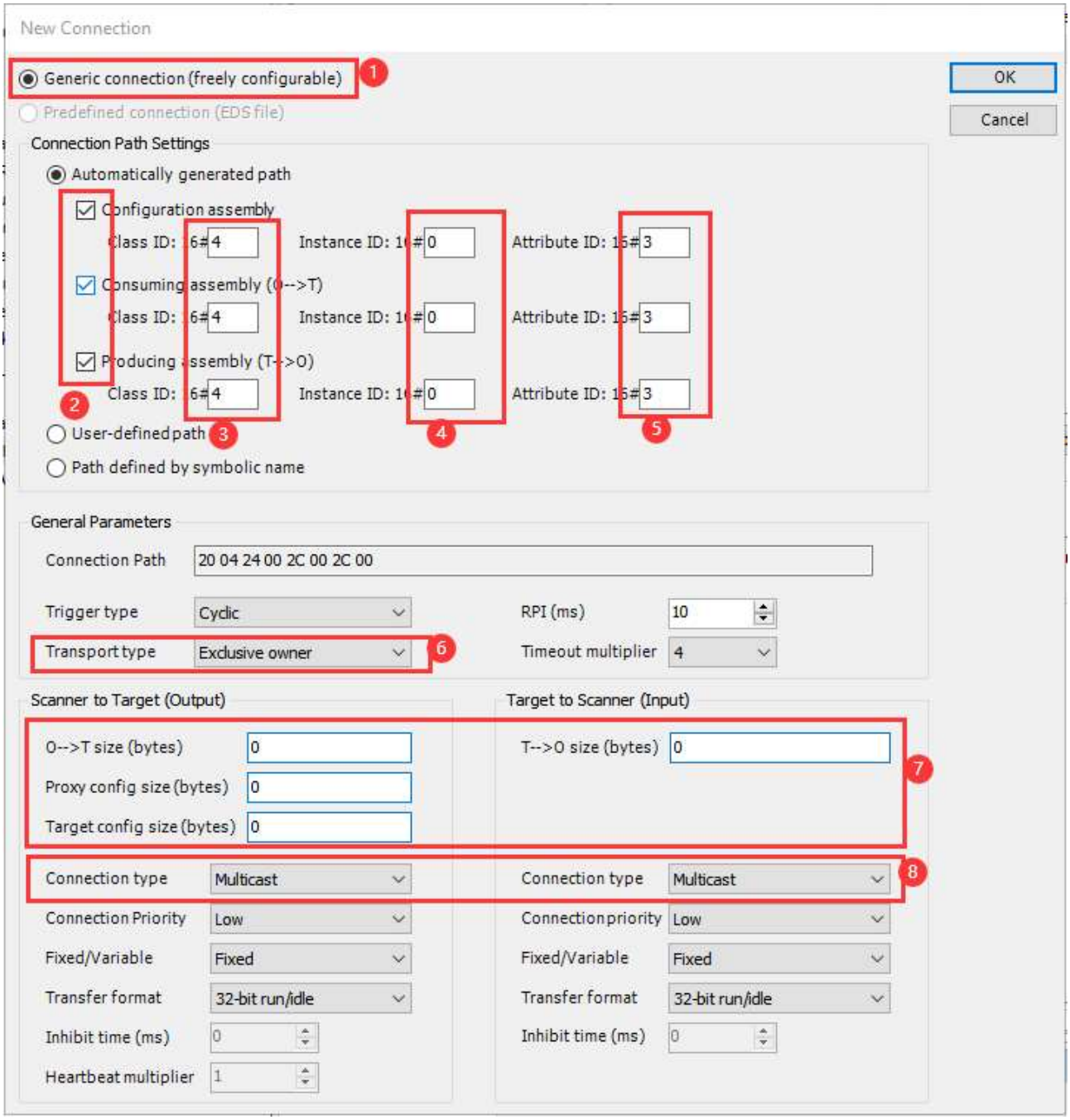


Step 6: Click on the connection to add a label connection that matches the adapter data size. The first connection is to establish an exclusive owner transmission type, with a point-to-point connection type and a data size of 20 bytes. The second connection is to create a label name test11, with a transmission type of input only and a connection type of point-to-point and a data size of 10 bytes. The third connection is to create a label name test22, with a transmission type of input only and a connection type of point-to-point and a data size of 40 bytes.



Note:

(1) When creating a new connection using the "instance ID" and "exclusive owner" connection type, the configuration information is roughly as follows:

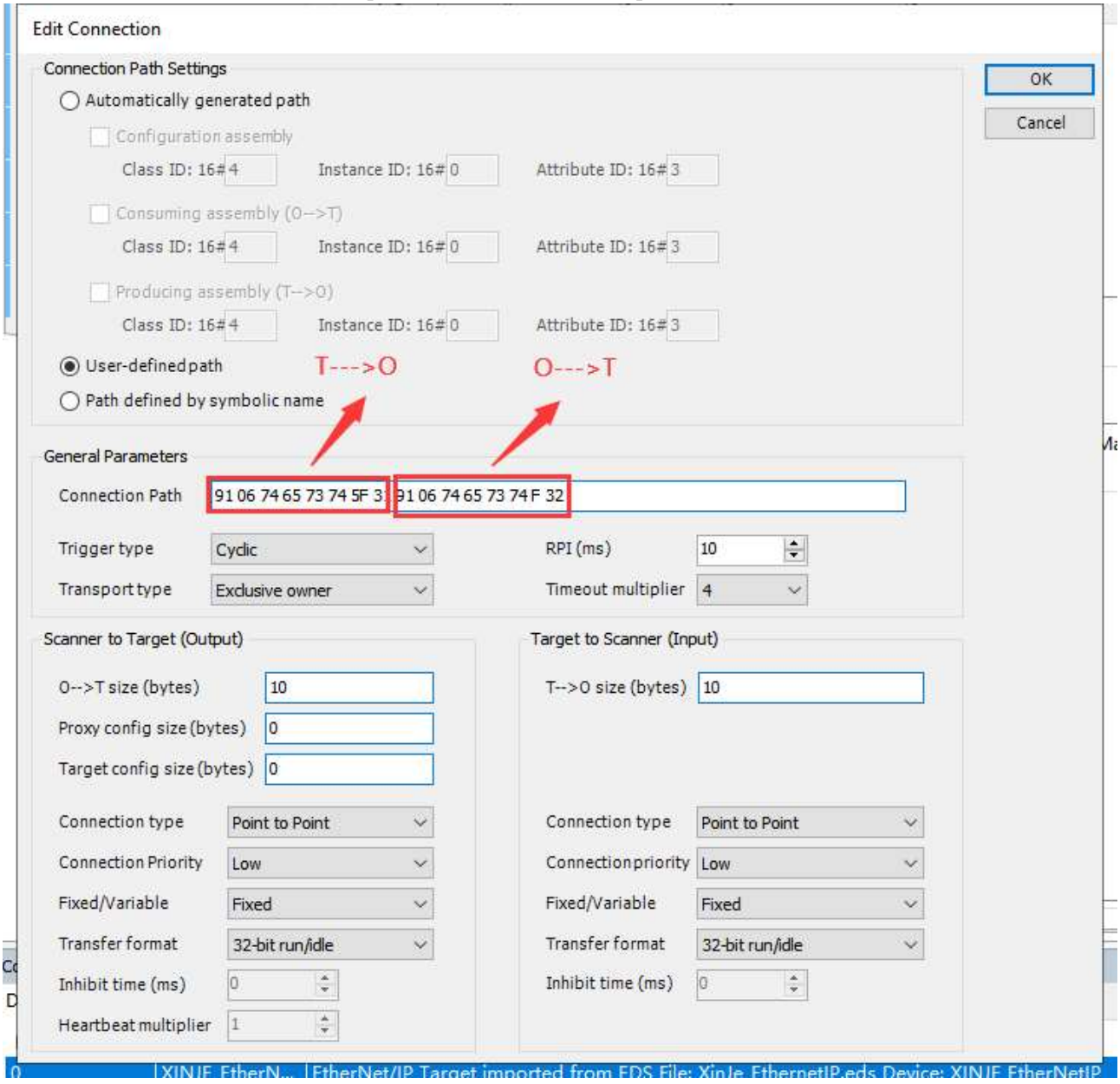


1	Select automatic path generation to enable instance ID configuration
2	Check the corresponding boxes for configuration assembly, consuming assembly, and producing assembly
3	Class ID is default value 4
4	Instance ID: The instance ID for configuration assembly is set to 1 by default. When creating a "exclusive owner" connection, the instance ID for consuming assembly (O ->T) should be consistent with the instance ID configured by the adapter. If "input only" connection is created, data in the direction of configuration (O ->T) will not be configured. The instance ID for consuming assembly (O ->T) must be filled in as FE, and the instance ID for producing assembly (T ->O) should be consistent with the instance ID configured in adapter.
5	Attribute ID is default value 3
6	Transport type select as actual using condition
7	The size of data to be transmitted for corresponding configuration
8	Configure the corresponding connection types as needed

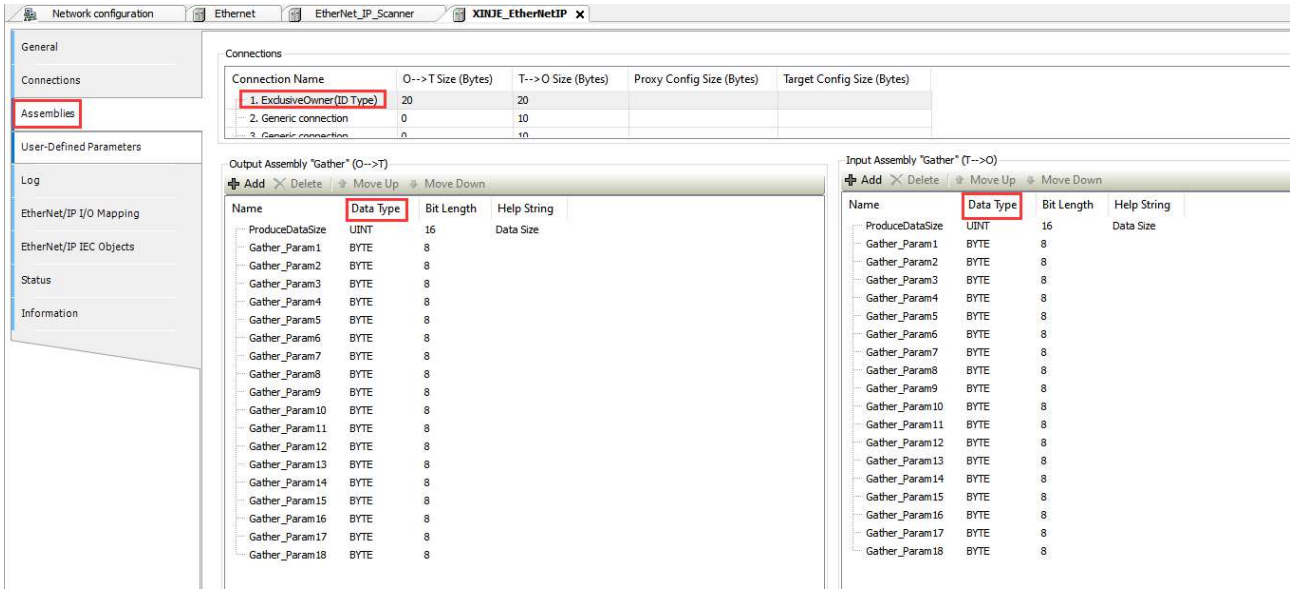
(2) When creating a new connection using the "tag" and "exclusive owner" connection type, the configuration information is roughly as follows:

The connection path needs to be generated based on the tag name configured by the adapter, and the connection path in the T ->O direction needs to be placed before the connection path in the O ->T direction;

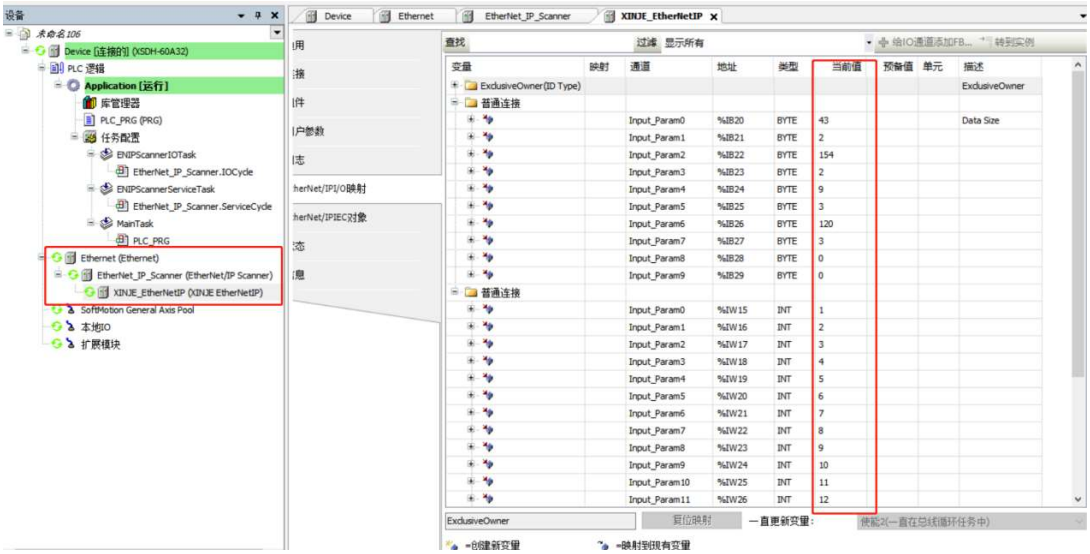
Quick generation of connection path: Click on the path defined by symbolic name, fill in the required label name, and then click on user-defined path to obtain a connection path code.



Step 7: Click on the assemblies to configure the data types in the specified connection input/output components as needed.



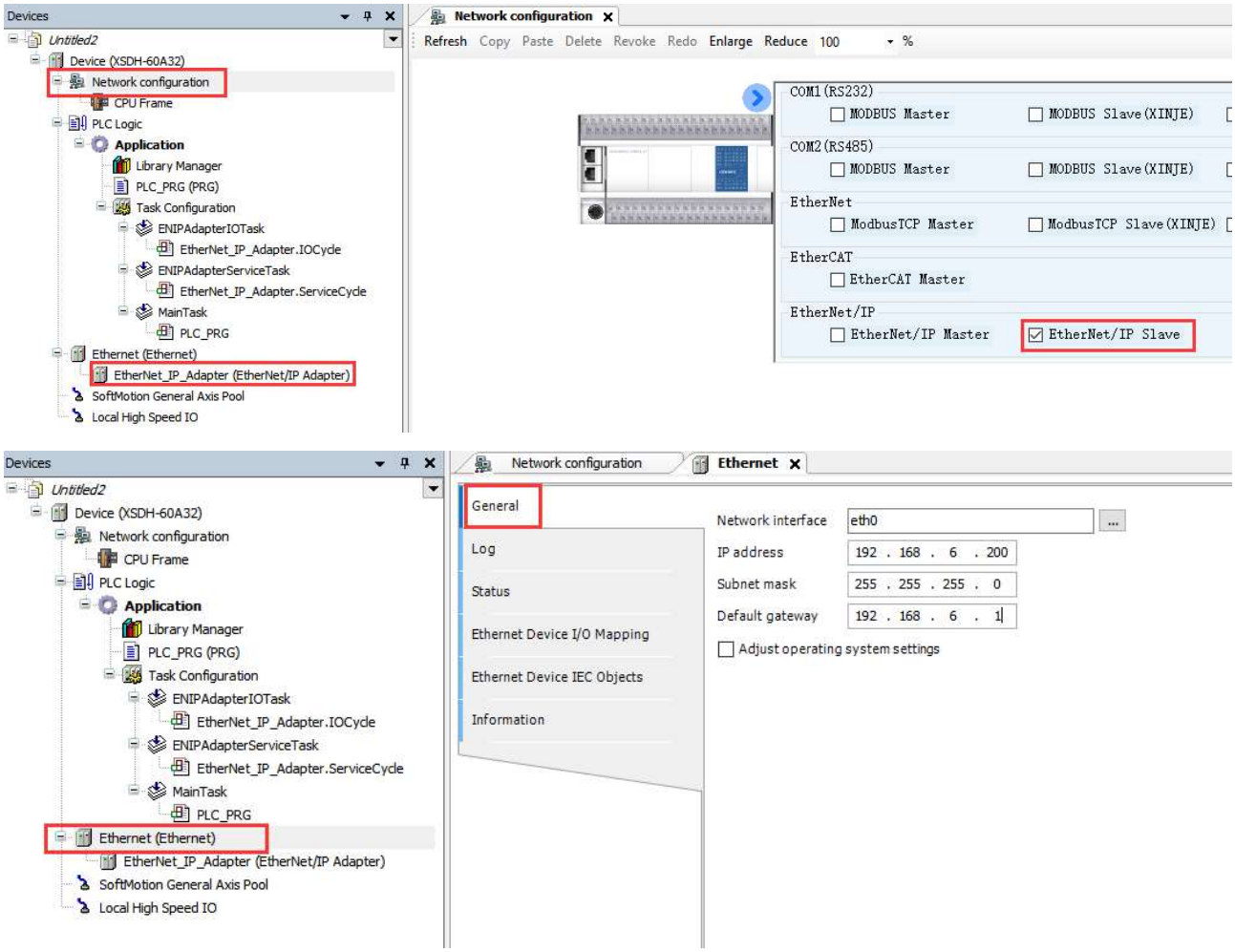
Step 8: Check the current communication status of the corresponding left tree, click on IO mapping to monitor whether data transmission is normal.



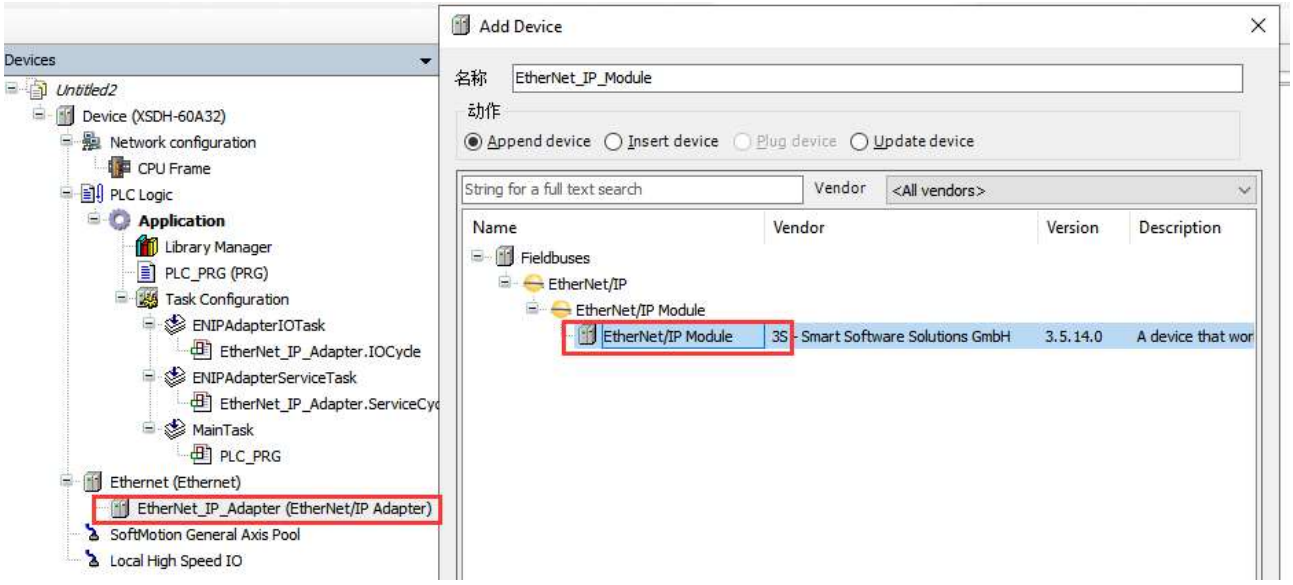
Application 9: Using Xinje PLC XDH-60T4-E as a scanner and Xinje PLC XSDH-60A32-E as an adapter for implicit communication.

PLC1: XDH-60T-E (IP 192.168.6.6) as the scanner and PLC2: XSDH-60A32-E (IP 192.168.6.200) as the adapter to achieve implicit communication between two PLCs. During the connection creation process, it is important to ensure that the data size of the connection point used is consistent with the data size of the adapter.

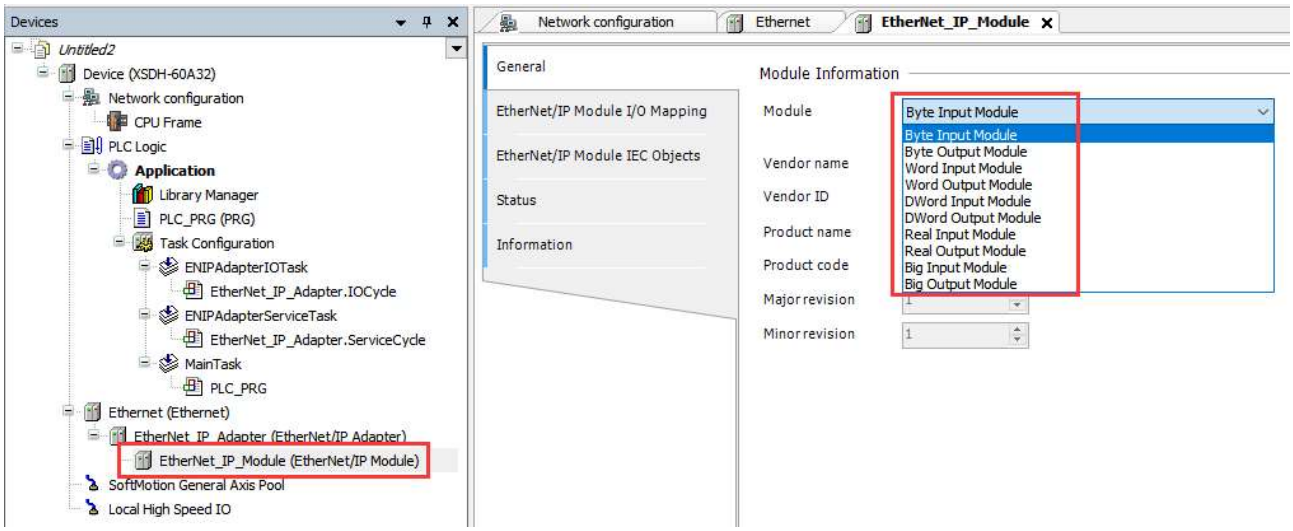
Step 1: Click on Network Configuration. Add an EthernetIP slave to the network configuration and create an Ethernet_IP_Adapter. Double click on Ethernrt to select the network card to use.



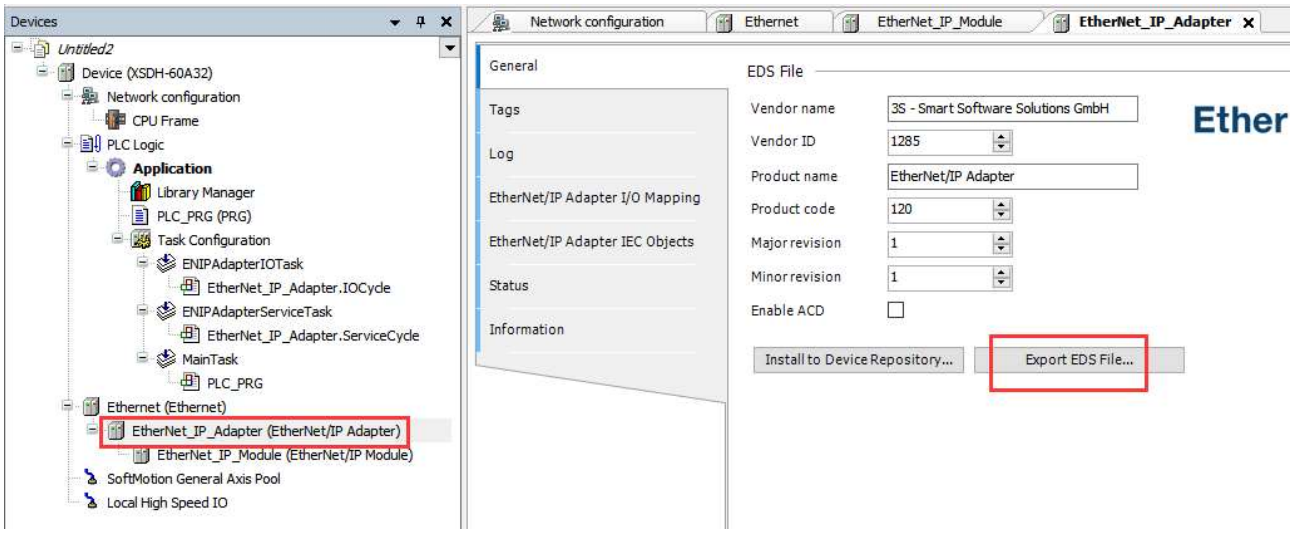
Step 3: Right click on Ethernet_IP_Adapter to add the required Ethernet/IP module.



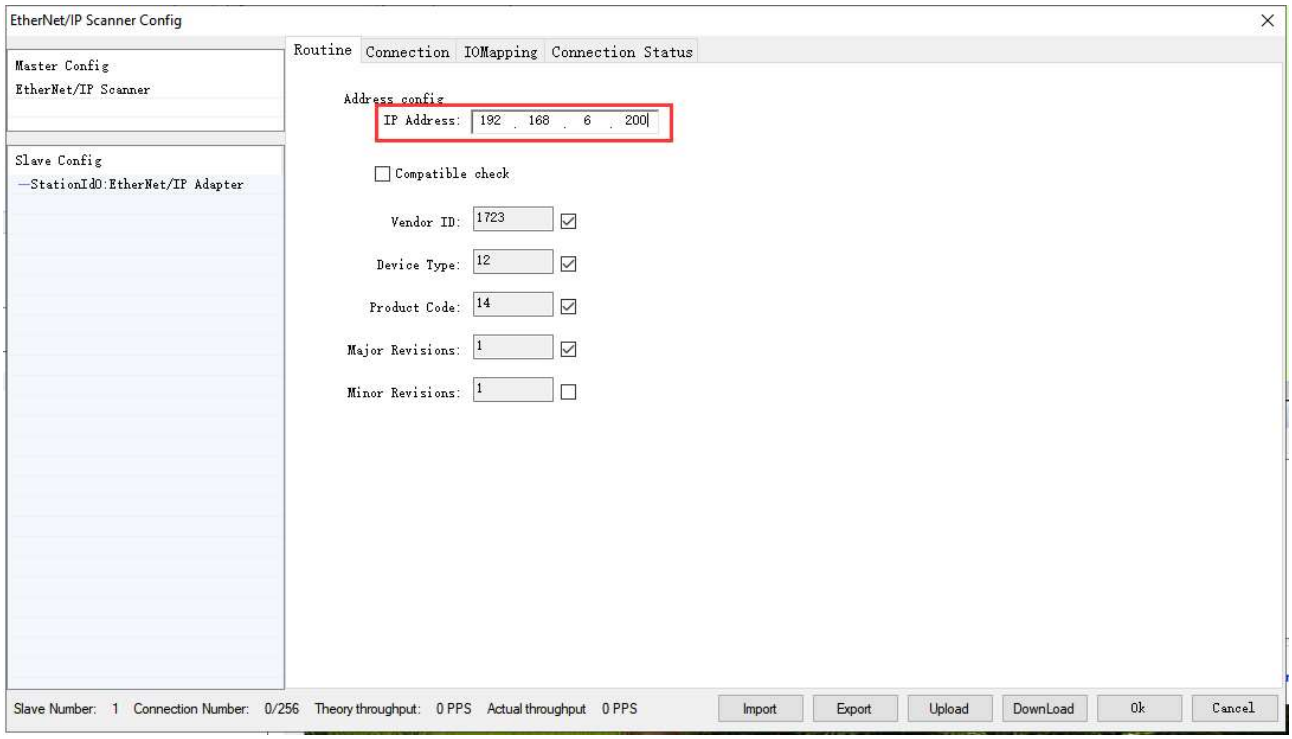
Step 4: Double click on the corresponding Ethernet/IP module to be added, and select Word Output Module and Word Input Module on the usual interface.



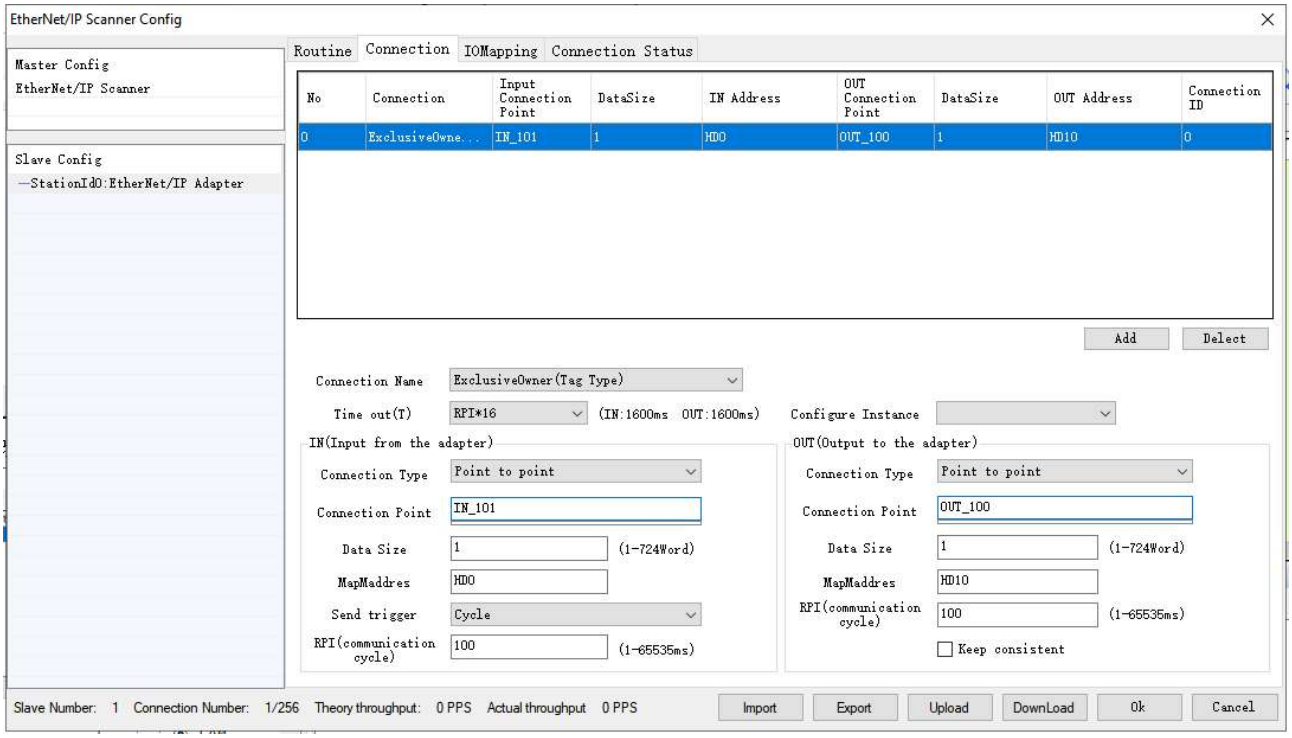
Step 5: Double click Ethernet_IP_Adapter to export the configured information in the form of an EDS file.



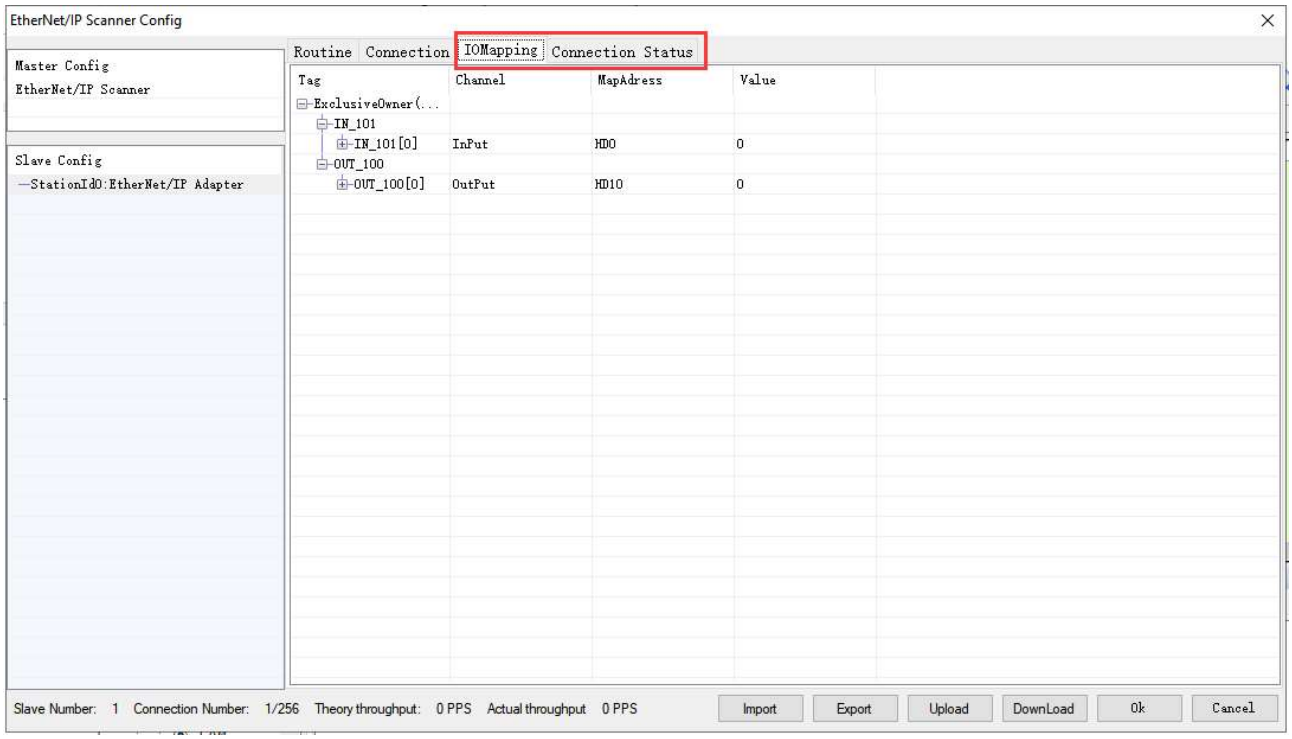
Step 6: On the Xinje XDPPro programming software scanner, load the exported EDS file, add the corresponding slave device after loading, and perform relevant configuration operations on the adapter's IP address and compatibility check:



Step 7: Click on the connection to view the connection type, and the imported connection type can also be modified according to actual needs.



Step 8: Click on IO mapping or connection status to operate and monitor corresponding data, and verify whether communication is normal.

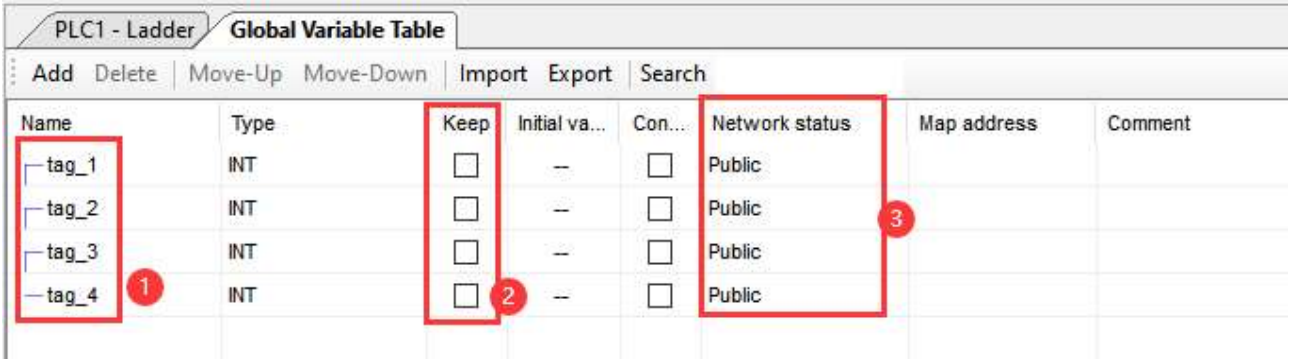


5-4-2. Explicit communication

Explicit message is a point to point communication method in which the client sends a request to the server and waits for the server to respond; Label communication is a communication method based on label address that reads or writes data by accessing the label address in the device. Explicit messages consist of two parts: the client and the server.

5-4-2-1. Explicit server

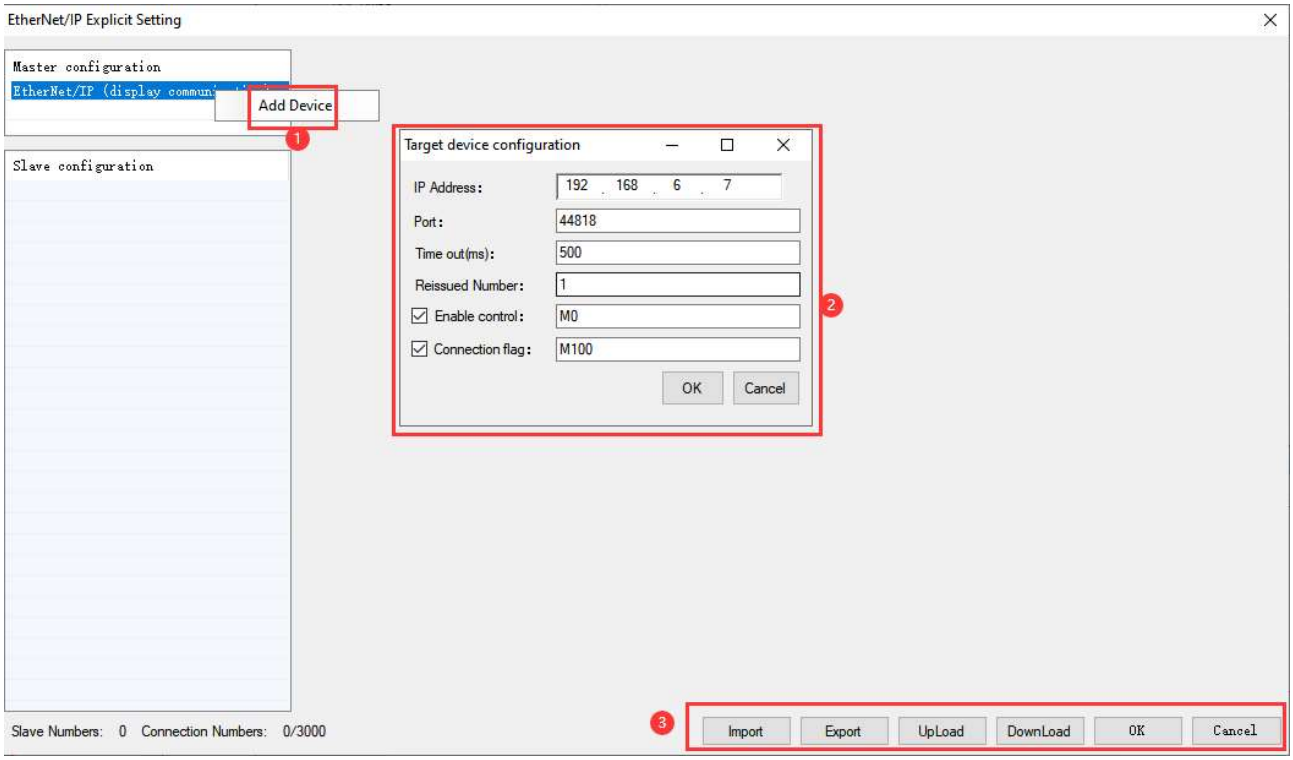
The name (case insensitive), label type, data type, length, and mapping address of explicit messages need to be defined in the global variable table of the server in advance. After definition, click download to download the configuration to the PLC and wait for the client to establish a connection with it.



(1) Names are not case sensitive, meaning tag_1 is equivalent to TAG_1.
(2) The mapping address is power off holding registers, please select this to maintain the value.
(3) The network status please set to public.

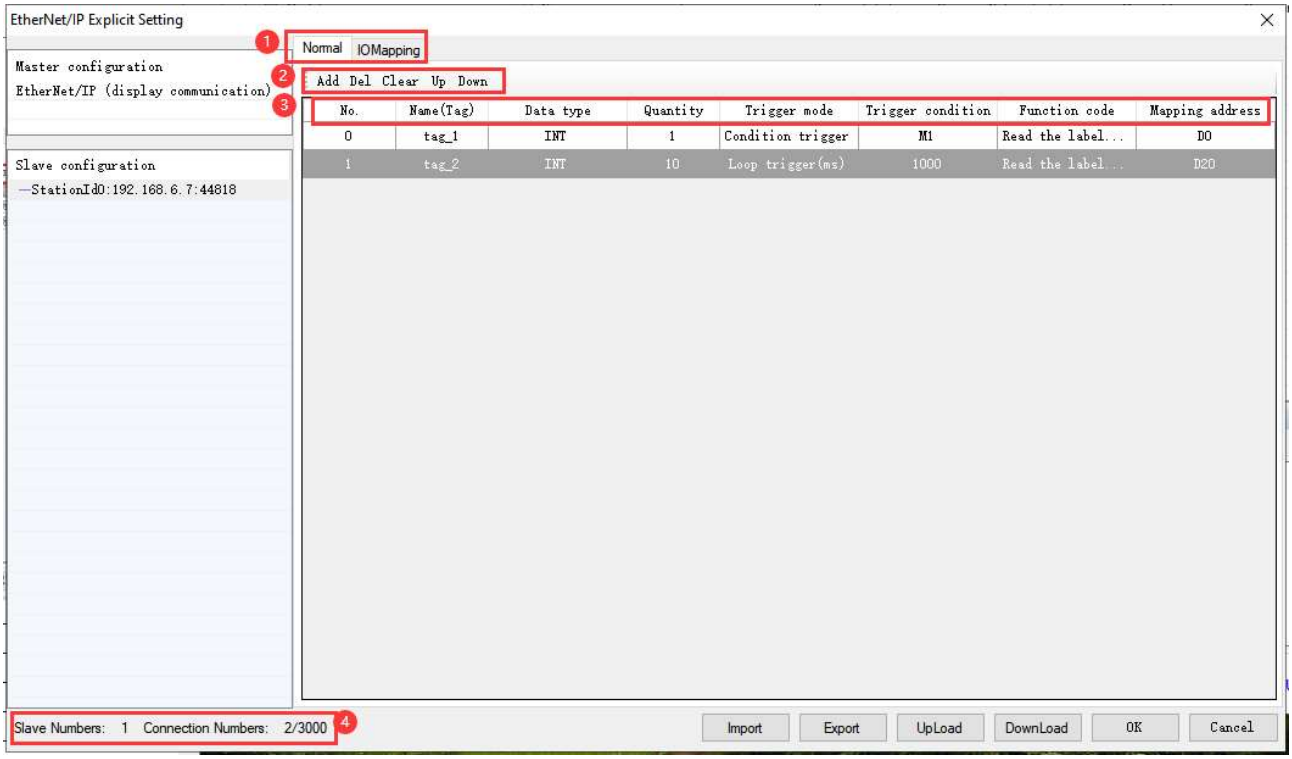
5-4-2-2. Explicit client

1. Add device



1	Right click on EthetNet/IP (Display Communication) in the main station configuration to add devices.	
2	Configure the target devices for adding slave stations accordingly.	
	IP address	As the IP address of the server PLC; Default 192.168.6.1, starting from 1, the next one defaults to the previous address +1
	Port	Default 44818, fixed and cannot be modified.
	Time out	The default setting is 500ms, with a range of 1-65535.
	Reissue number	When the triggering method is conditional triggering, if the communication timeout occurs, it will be resent with a default number of times of 1, and the allowed input range is 1-15.
	Enable control	By default, it is not enabled. Enabled to set local coil control. When not enabled: PLC automatically establishes a connection to the target IP after running; When enabled: Only bit registers are supported, and a connection to the target IP is only established when the coil is normally ON. Close the connection when the conditions are not met.
	Connection flag	Store the result of the successful connection of this device in the corresponding connection flag register.
3	Import	Import the configured information into the current configuration interface in the form of an XML file.
	Export	Export the configured information in the form of an XML file.
	Upload	Upload the configuration information downloaded to the PLC to the current configuration interface, and the uploaded configuration information will overwrite the existing configuration information on the current interface.
	Download	Download the configuration information of the current configuration interface to the PLC. The downloaded configuration information will overwrite the original configuration information in the PLC and take effect in real time with the new configuration information.
	OK	Click OK to save the configuration information for the current page.
	Cancel	Click to discard the configuration information for the current page.

2. Add connection



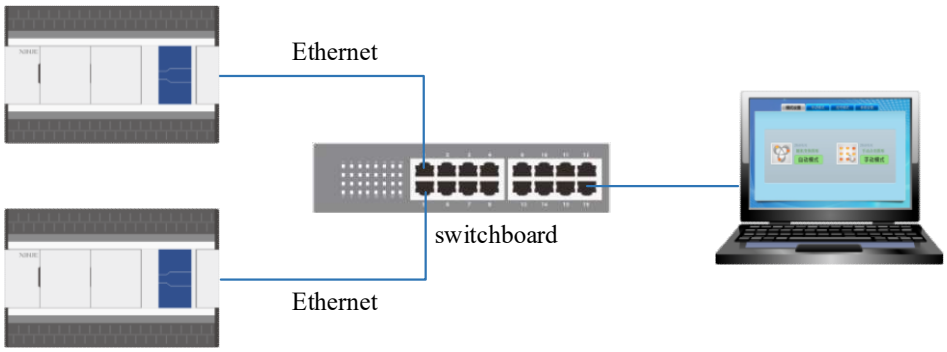
1	Normal	Add the specified node to the slave station in this interface to establish a connection.
	IO mapping	View or monitor detailed address information for adding connection mappings.
2	Add	Clicking on add will create a new connection.
	Delete	Select the corresponding established connection, click delete to delete the selected connection.
	Clear	Delete all configuration information on this interface.
	Up	For the selected established connection, click Move Up to move it up by one unit.
	Down	For the selected established connection, click move down to move down one unit.
3	No.	Click on add to create a connection. This number will automatically increase by 1
	Name	The name of the connection should be consistent with the label of the explicit server, ensuring the correct label name and data type. Note: The number of data corresponding to the name must not exceed the data defined by the server, otherwise communication will fail
	Data type	For specific supported data types, please refer to 5-3-3. client and server support variable types
	Quantity	The number of label variables corresponding to the read or write operation of this connection
	Trigger mode	Cyclic triggering: Triggering in cycles according to the set triggering conditions; Conditional triggering: When the set triggering condition changes state, it triggers the explicit client
	Function code	Read tag (0x4c): Read tag service, where the client reads the specified tag data from the server; Write tag (0x4d): Write tag service, where the client writes the specified tag data to the server; Mapping address: Maps read label data or cached label data to be written into PLC registers.
4	Slave number	Count the number of slave stations connected under the current master station.
	Connection number	Count the number of connections established between the master station and all connected slave stations. The specific specifications for the number of slave stations or supported

		connections supported by Ethernet/IP communication can be found in section 5-3. Ethernet/IP communication specifications.

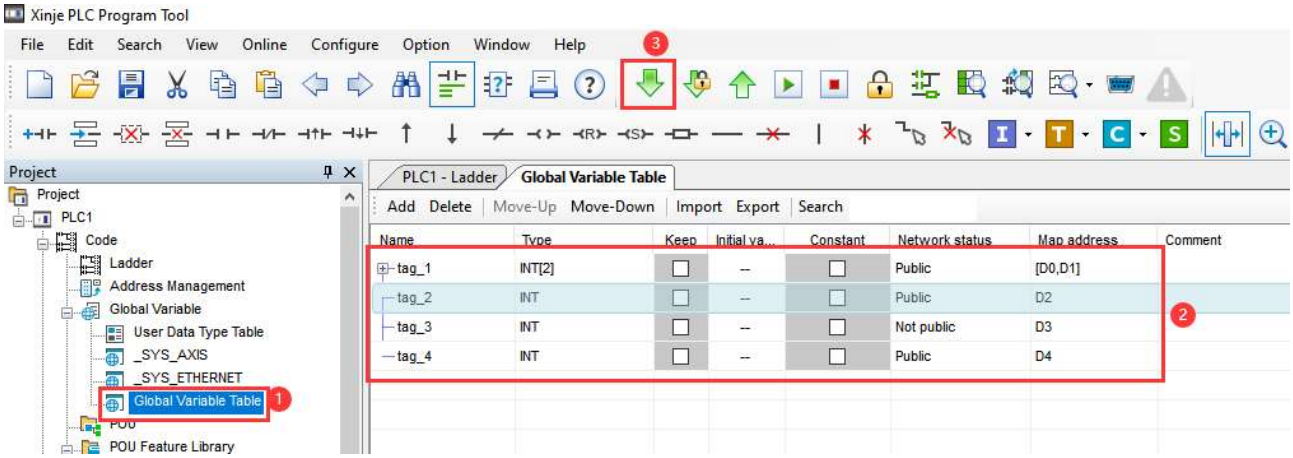
5-4-2-3. Application

Application 1: Two Xinje XDH-60T-E for explicit label communication.

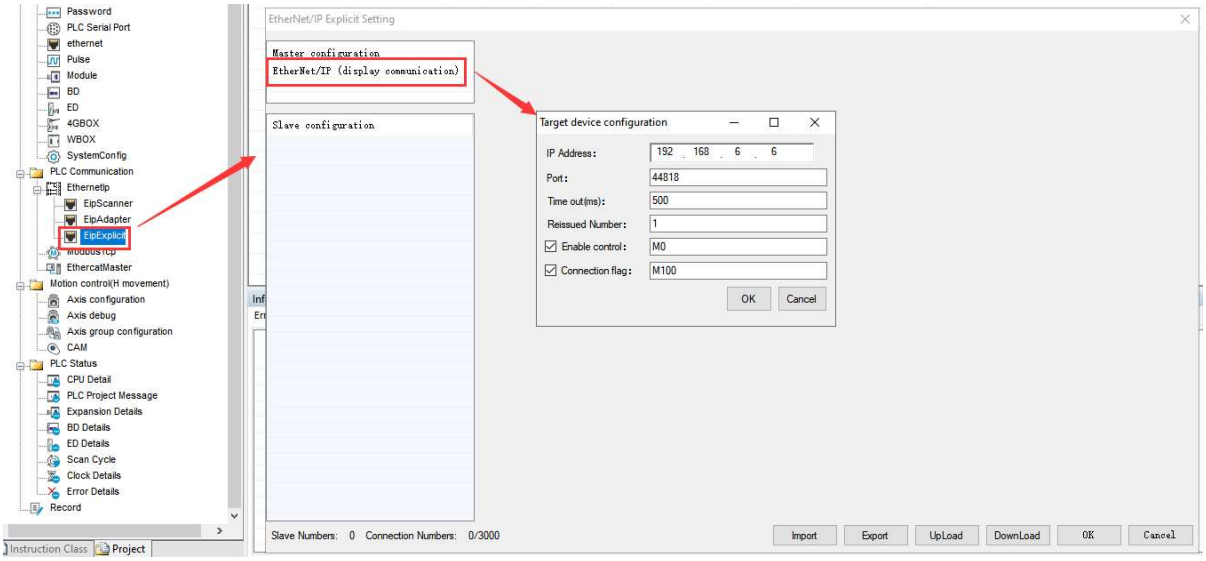
PLC1: XDH-60T-E (IP 192.168.6.6) as the explicit server and PLC2: XDH-60T-E (IP 192.168.6.7) as the explicit client to achieve explicit label communication between two PLCs.



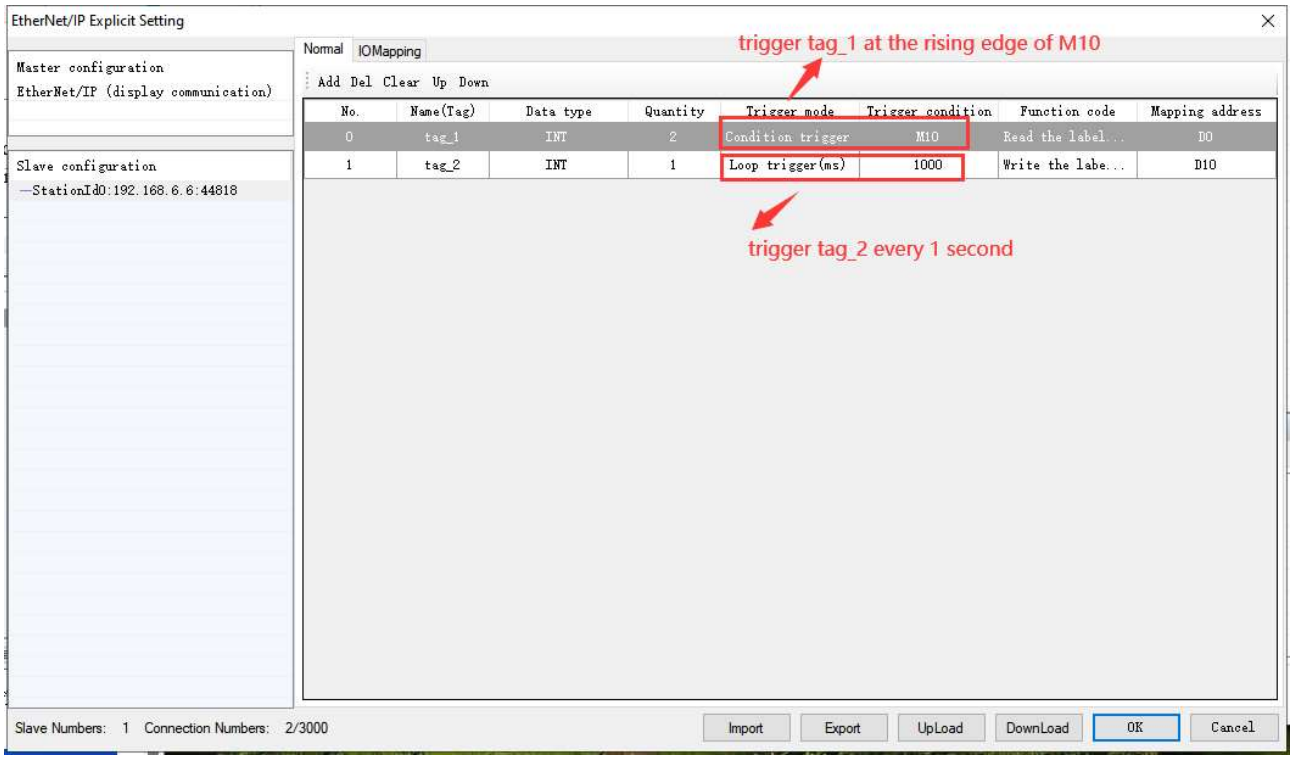
Step 1: Create corresponding variables in the global variable table on the server, and select the network state corresponding to the label as public state. The specific operation configuration is as follows:



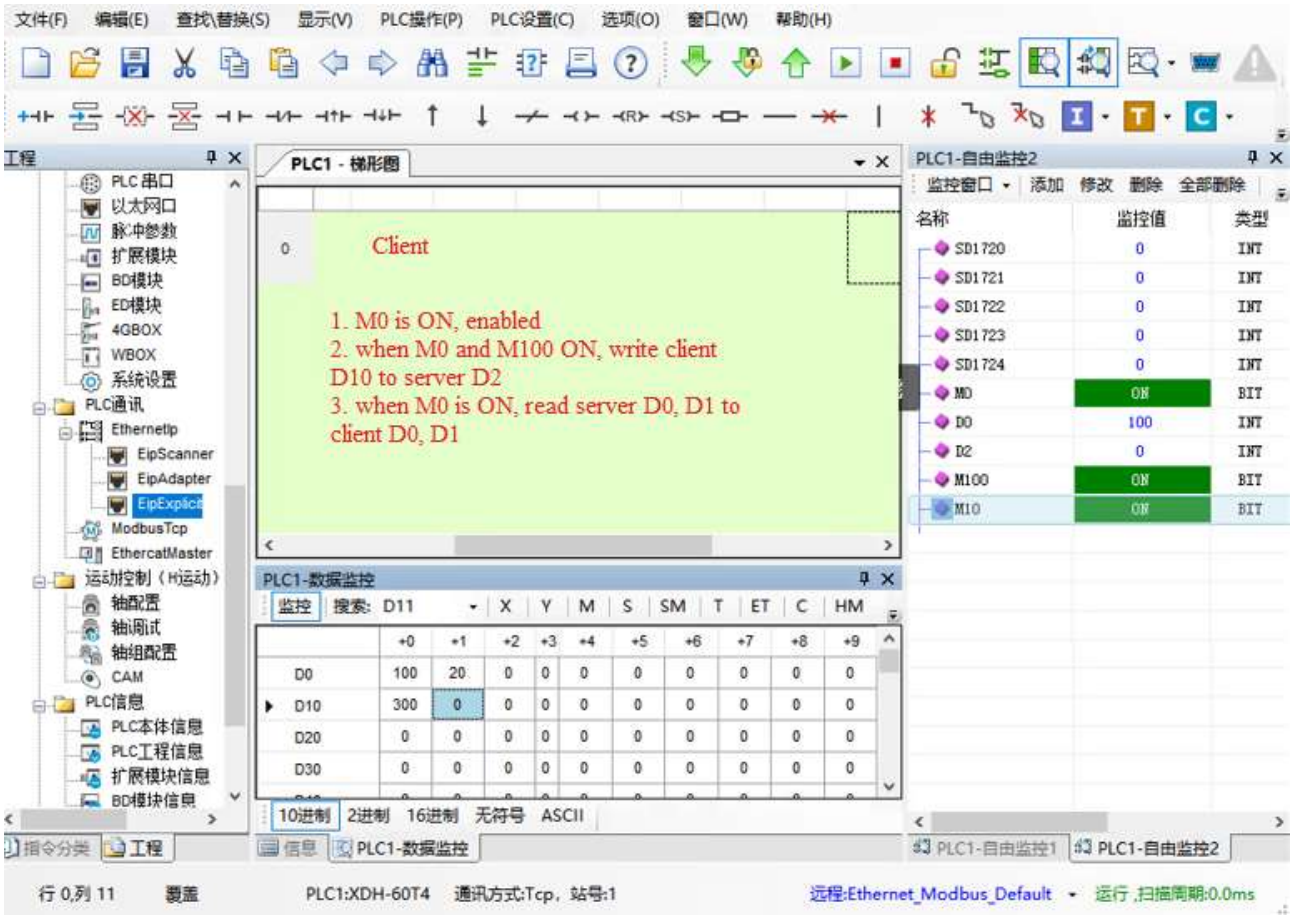
Step 2: Add a server device to the client and configure the corresponding IP address and related parameters for the specified communication server:

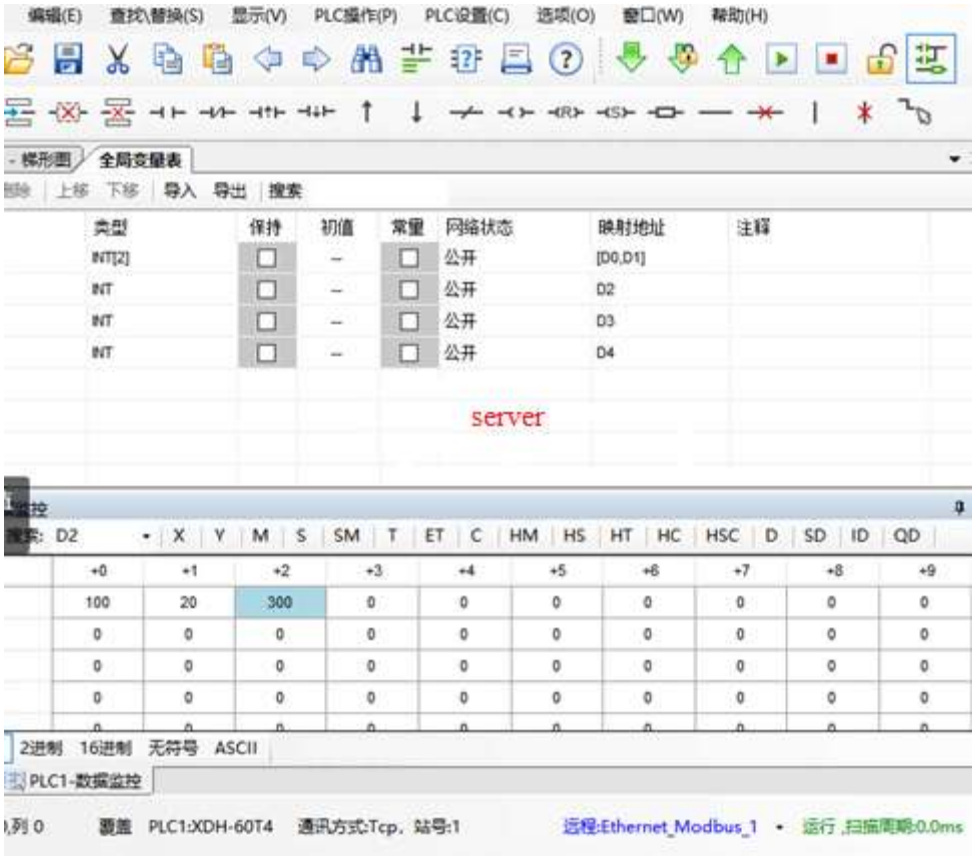


Step 3: Add a connection to the client. Users can choose the corresponding triggering method and triggering conditions based on the actual application situation, and perform tag reading and writing operations on the server to the mapping address specified by the client.



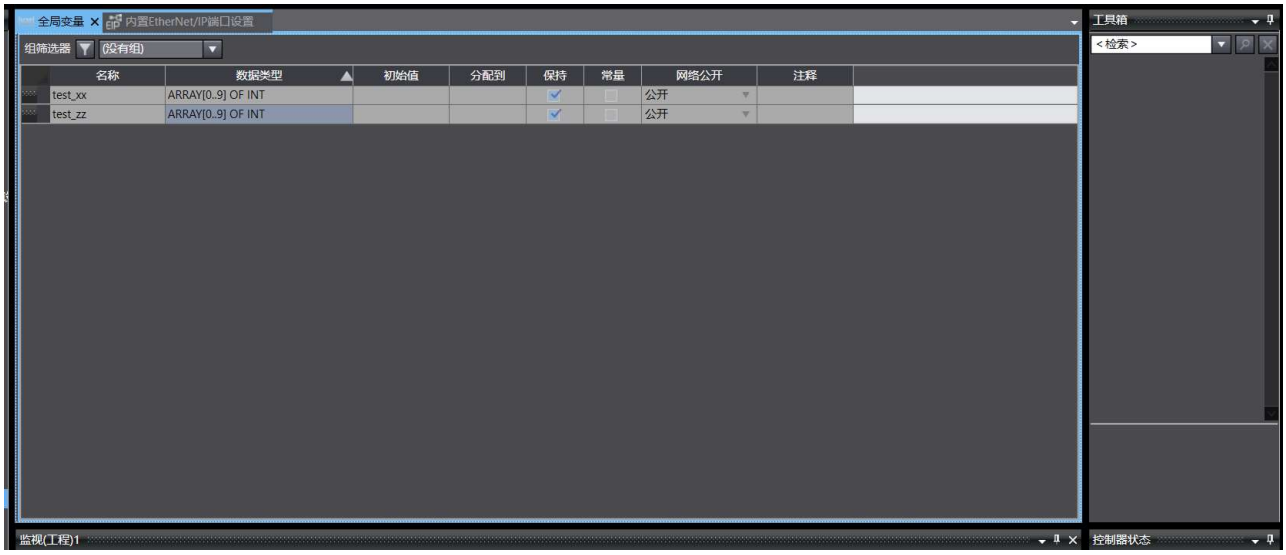
Step 4: After adding the configuration, click "Download" to download the configuration information to the PLC. After downloading, monitor the corresponding mapping address and check its communication status.



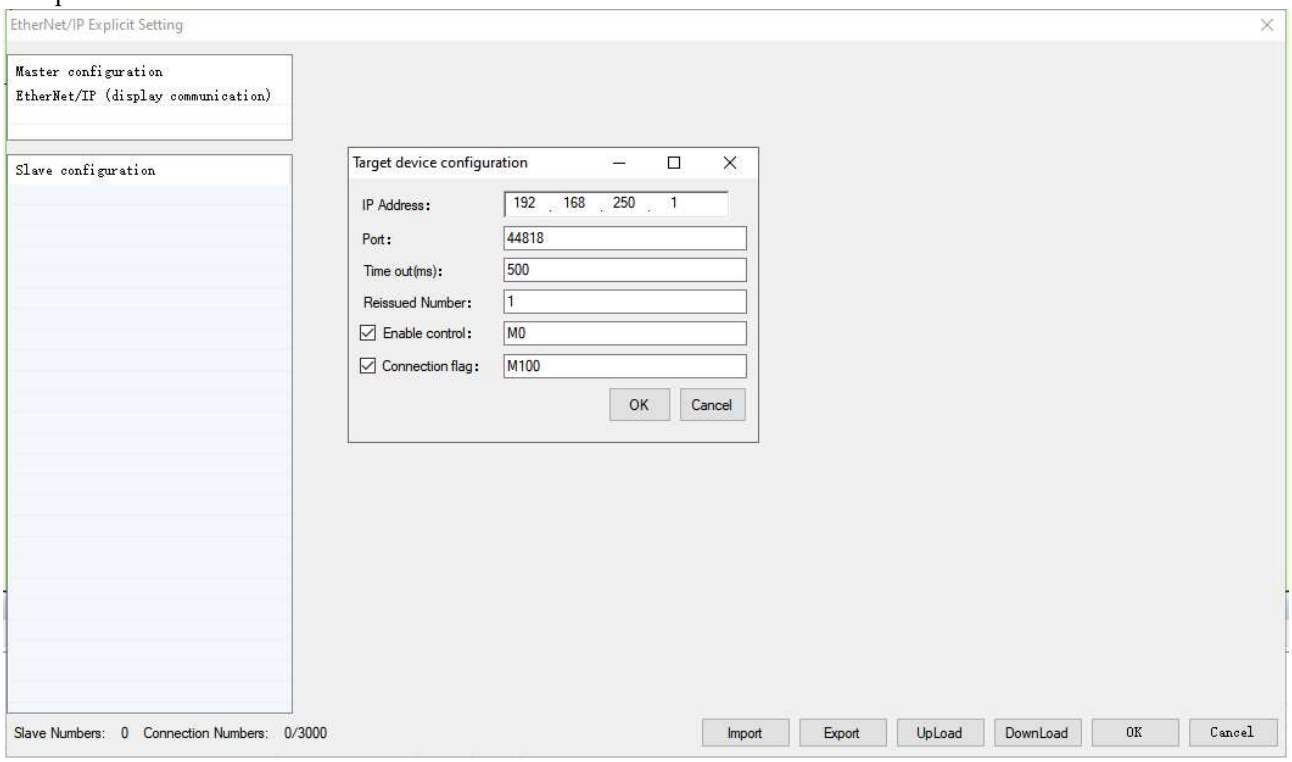

编辑(E) 查找\替换(S) 显示(V) PLC操作(P) PLC设置(C) 选项(O) 窗口(W) 帮助(H)
梯形图 全局变量表
上移 下移 导入 导出 搜索
类型 保持 初值 常量 网络状态 映射地址 注释
INT[2] 公开 [D0,D1]
INT 公开 D2
INT 公开 D3
INT 公开 D4
server
D2 X Y M S SM T ET C HM HS HT HC HSC D SD ID QD
+0 +1 +2 +3 +4 +5 +6 +7 +8 +9
100 20 300 0 0 0 0 0 0 0
2进制 16进制 无符号 ASCII
PLC1-数据监控
覆盖 PLC1:XDH-60T4 通讯方式:Tcp，站号:1 远程:Ethernet_Modbus_1 运行 ,扫描周期0.0ms

Application 2: Use Xinje XDH-60T-E as the client and Omron NJ501-1500 as the server for explicit tag communication.

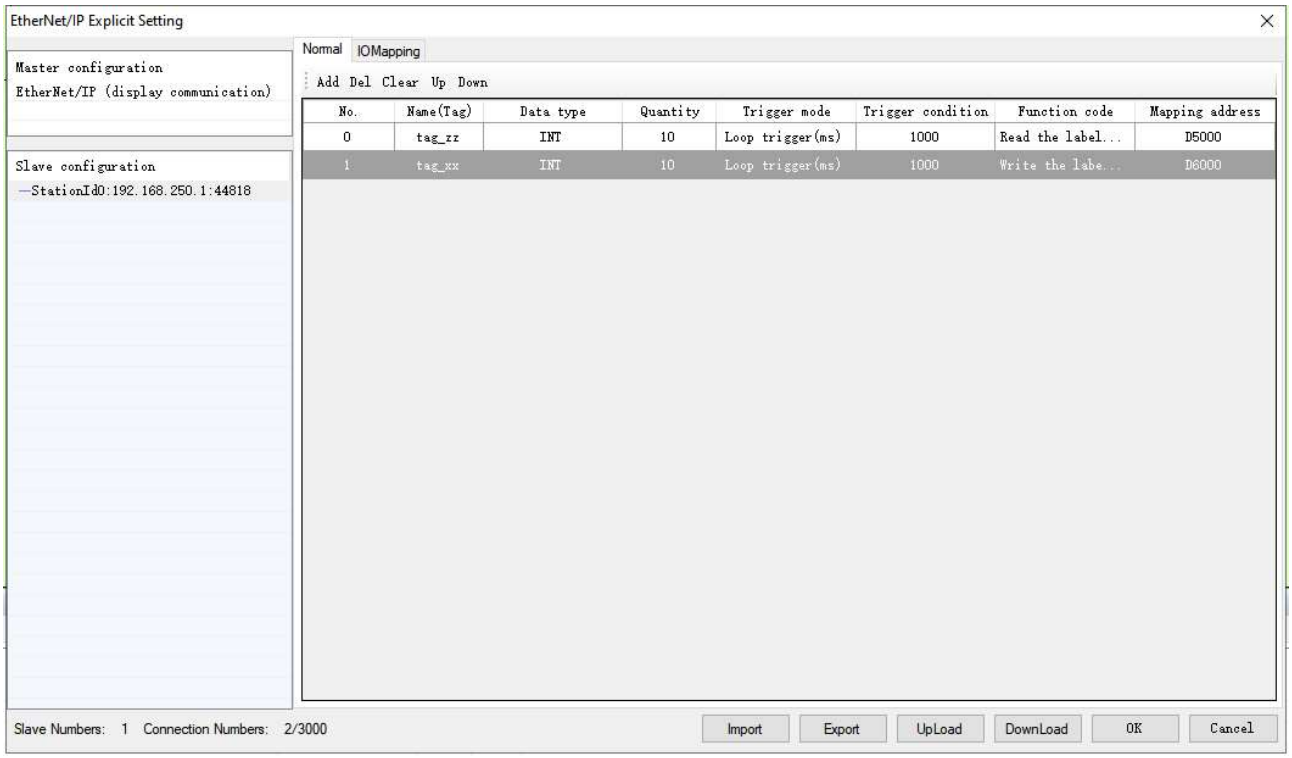
Step 1: Define the variables that need to be communicated in the Omron Sysmac Studio programming software, and select the public state of the established label network as public.



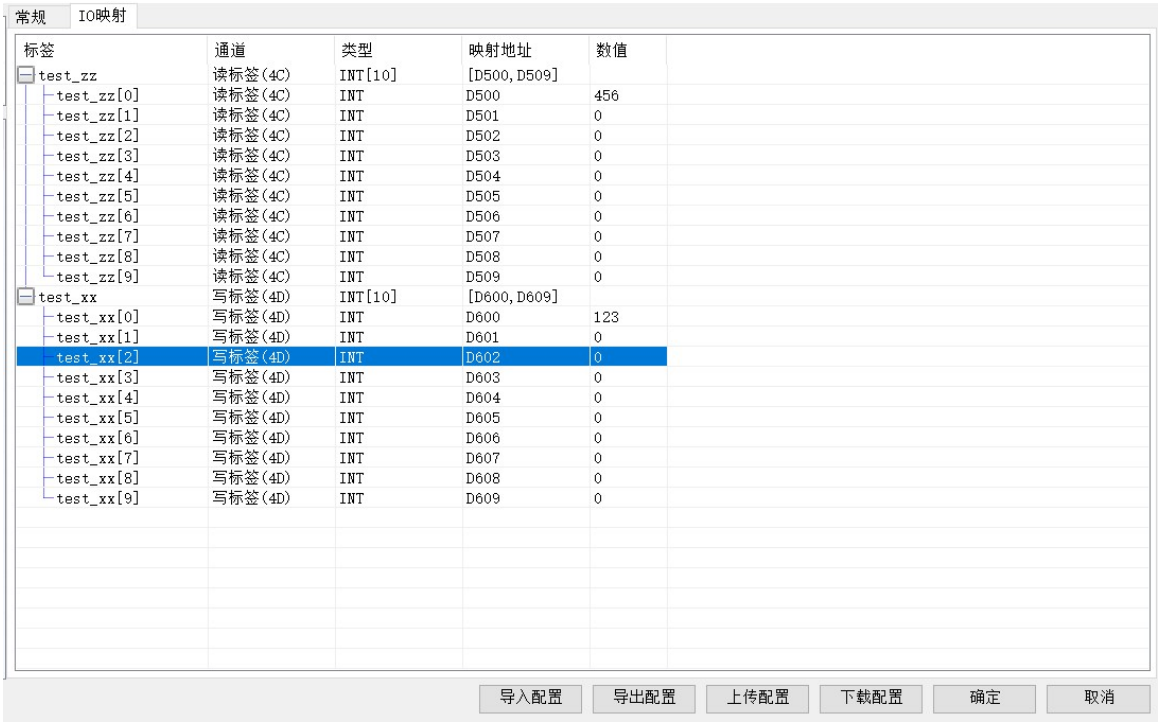
Step 2: Add a server device to the client and configure the corresponding IP address and related parameters for the specified communication server:

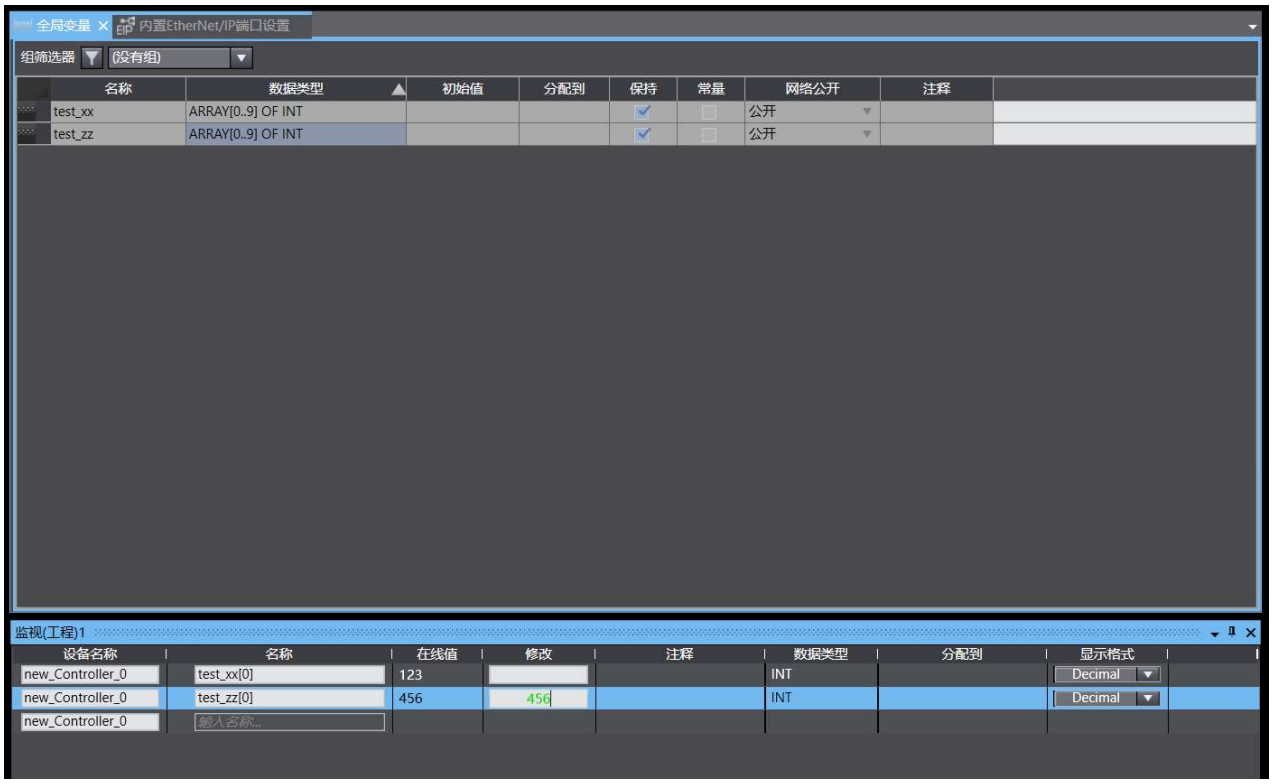


Step 3: Add a connection in the client, with the first connection established as a read label method and the second connection established as a write label method.



Step 4: After setting ON M0, when the M100 enable connection flag is set successfully, it indicates that the connection has been established. Click on the IO mapping to check if the communication between read and write data is normal.




全局变量
内置EtherNet/IP端口设置
组筛选器
(没有组)
名称
数据类型
初始值
分配到
保持
常量
网络公开
注释
test_xx
ARRAY[0..9] OF INT
公开
test_zz
ARRAY[0..9] OF INT
公开
监视(工程)1
设备名称
名称
在线值
修改
注释
数据类型
分配到
显示格式
new_Controller_0
test_xx[0]
123
INT
Decimal
new_Controller_0
test_zz[0]
456
456
INT
Decimal
new_Controller_0
输入名称...

-

Appendix

Code	Explanation	Reason and solution
0xFF	No extension	-
0x100	FWD repeated opening	-
0x103	Class triggering invalid	-
0x106	Conflict of ownership	Error reason: The connection point in the O ->T direction of the slave station configuration has already been used. Problem point: The connection point in the O>T direction of the slave station configuration has already been used. Solution: Replace the connection points in the O ->T direction of the main station.
0x107	Connection not found	Error reason: Connection not found. Problem point: It is highly likely that EDS does not match or is missing configuration items. Solution: Determine if the EDS of the slave station is correct.
0x108	Invalid connection type	-
0x109	Invalid connection size	Reason for error: T ->O or O ->T data size setting error, or configuration data length setting error. Problem point: T ->O or O ->T data size setting error, or configuration data length setting error. Solution: Modify the length of the master station data and modify the length of the slave station data
0x110	Device not configured	-
0x111	RPI not support	Error reason: The RPI slave (adapter) setting is not supported. Problem point: It is highly likely that EDS does not match. Solution: Choose the correct EDS.
0x112	RPI value is unacceptable	Error reason: ListenOnly (ID type) RPI configuration error. Problem point: ListenOnly (ID type) RPI requires the same configuration as RPI that depends on InputOnly and ExclusiveOwner. Solution: Learn the correct usage of ListenOnly.
0x113	Connection limit reached	-
0x114	Supplier product code mismatch	Reason for error: Supplier ID or product code check error in compatibility check. Problem point: The EDS selected for the configuration of the slave station equipment does not match that of the master station. Solution: Choose the correct EDS.
0x115	Product type mismatch	Error reason: Device type check error in compatibility check. Problem point: The EDS selected for the configuration of the

-

Code	Explanation	Reason and solution
		slave station equipment does not match that of the master station. Solution: Choose the correct EDS.
0x116	Revision mismatch	Reason for error: The main revision check error in compatibility check. Problem point: The EDS selected for the configuration of the slave station equipment does not match that of the master station. Solution: Choose the correct EDS.
0x117	Invalid connection point	Reason for error: The connection point selection for T ->O or O ->T is incorrect. Problem point: The connection points selected by the master station (scanner) for T ->O or O ->T do not match the configuration of the slave station (adapter). Solution: Choose the correct connection point.
0x118	Invalid configuration format	-
0x119	No control connection	Error reason: ListenOnly (ID type) connection type configuration error. Problem point: The ListenOnly (ID type) connection type cannot be configured as point-to-point, or when configured as multicast, it requires an InputOnly and ExclusiveOwner connection point to also be configured as multicast. Solution: Learn the correct usage of ListenOnly.
0x11A	Reaching target connection limit	-
0x11B	RPI is less than the limit	-
0x11C	Transfer class not supported	Problem point: These issues basically belong to EDS mismatch. Solution: Choose the correct EDS.
0x11D	Production trigger not supported	
0x11E	Direction not supported	
0x11F	O-T fixed variable invalid	
0x120	T-O fixed variable invalid	
0x121	O-T priority invalid	
0x122	T-O priority invalid	
0x123	O-T connection type invalid	
0x124	T-O connection type invalid	
0x125	O-T redundant owner invalid	-
0x126	T-O redundant owner	-

-

Code	Explanation	Reason and solution
	invalid	
0x127	O-T size invalid	Reason for error: (1) The data size setting for O ->T connection is incorrect; (2) Connection point option configuration error. Problem points: (1) The data size setting for O ->T connection is incorrect; (2) The connection point selected by the master station does not match the connection point configured by the slave station. Solution: (1) Modify the O ->T data length of the main station for connection or modify the data length of the consumer connection point configured by the slave station; (2) Modify the connection points of the master station for connection or modify the connection points configured by the slave station.
0x128	T-O size invalid	Error reason: The data size setting for T ->O connection is incorrect. Problem point: The data size setting for T ->O connection is incorrect. Solution: Modify the T ->O data length of the main station's connection or modify the data length of the slave station's configuration producer connection point.
0x129	Invalid configuration path	-
0x12A	Invalid consumption path	Error reason: Connection point option configuration error. Problem point: The connection point selected by the master station does not match the connection point configured by the slave station. Solution: Modify the connection points of the master station for connection or modify the connection points configured by the slave station.
0x12B	Production path is invalid	Error reason: Connection point option configuration error. Problem point: The connection point selected by the master station does not match the connection point configured by the slave station. Solution: Modify the connection points of the master station for connection or modify the connection points configured by the slave station.
0x12C	No configuration symbol	-
0x12D	No consumption symbol	Error reason: Connection label name configuration error. Problem point: The connection label name selected by the master site does not match the connection label name configured by the

-

Code	Explanation	Reason and solution
		slave station. Solution: Modify the connection label name of the master station for connection or modify the connection label name configured by the slave station.
0x12E	No production symbol	Error reason: Connection label name option configuration error. Problem point: The connection label name selected by the master site does not match the connection label name configured by the slave station. Solution: Modify the connection label name of the master station for connection or modify the connection label name configured by the slave station.
0x12F	Invalid application path combination	-
0x130	Inconsistent consumption data format	-
0x131	Inconsistent production data format	-
0x132	Not support empty FORWARD OPEN	-
0x133	Wrong connection timeout multiplier	-
0x134	T-O connector size mismatch	-
0x135	T-O fixed variable mismatch	-
0x136	T-O connection priority mismatch	-
0x137	Transport category mismatch	-
0x138	T-O production trigger mismatch	Error reason: ListenOnly (ID type) trigger condition configuration error. Problem point: The trigger conditions for ListenOnly (ID type) need to be configured the same as those for InputOnly and ExclusiveOwner. Solution: Learn the correct usage of ListenOnly.
0x139	T-O production inhibition mismatch	-
0x203	Connection timeout	Error reason: Communication timeout. Problem point: There is no data packet within the timeout multiple time in the O ->T or T ->O direction. Solution: Usually, set PRI higher.
0x204	Unconnected send	-

-

Code	Explanation	Reason and solution
	timeout	
0x205	parameter error	-
0x206	Message too large	-
0x207	Unconnected packet without reply	-
0x208	Service demand connection	-
0x301	No available buffer memory	-
0x302	Bandwidth unavailable	-
0x303	Label filter not available	-
0x304	Real time data not configured	-
0x311	Port unavailable	-
0x312	Link address not available	-
0x315	Invalid segment type value	Reason for error: The default connection point for O ->T is incorrect. Problem point: EDS mismatch. Solution: Replace with the correct EDS.
0x316	Path connection mismatch	-
0x317	Invalid network segment	-
0x318	Invalid link address	-
0x319	The second resource is not available	-
0x31A	Connection established	-
0x31B	Established direct connection	-
0x31C	Others	-
0x31D	Redundant connection mismatch	-
0x31E	No more consumer resources available	-
0x31F	No target path resources	-
0x320	Supplier specific	-
0x813	Unconfigured outside subnet mask	-

WUXI XINJE ELECTRIC CO., LTD.

816 Jianshe West Road, Binhu District,

Wuxi City, Jiangsu Province 214072, China

Tel: 0510-85134136

Fax: (510) 85111290

Email: Fiona.xinje@vip.163.com

sales@xinje.com