TCP/IP communication based on Ethernet

User manual

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.

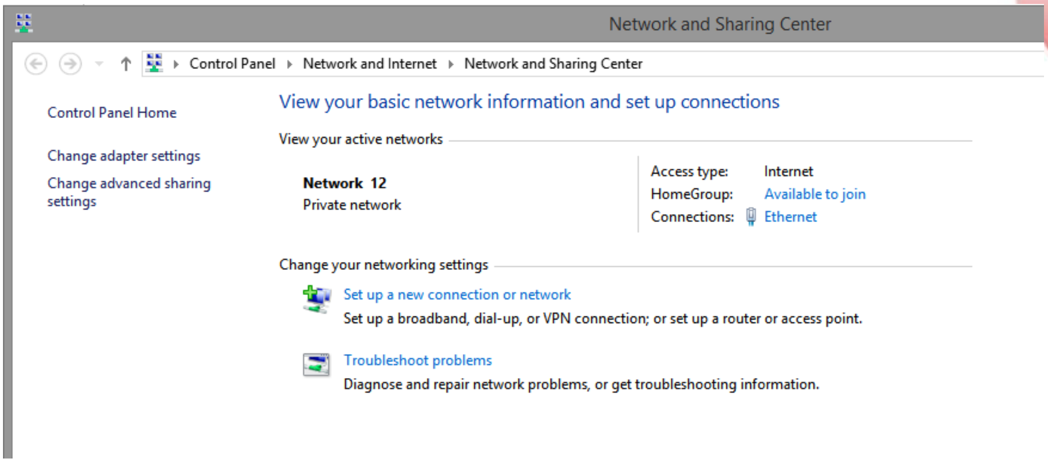
1-1-1. IP allocation

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:


Network and Sharing Center

Control Panel › Network and Internet › Network and Sharing Center

Control Panel Home

Change adapter settings

Change advanced sharing settings

View your basic network information and set up connections

View your active networks

Network 12
Private network

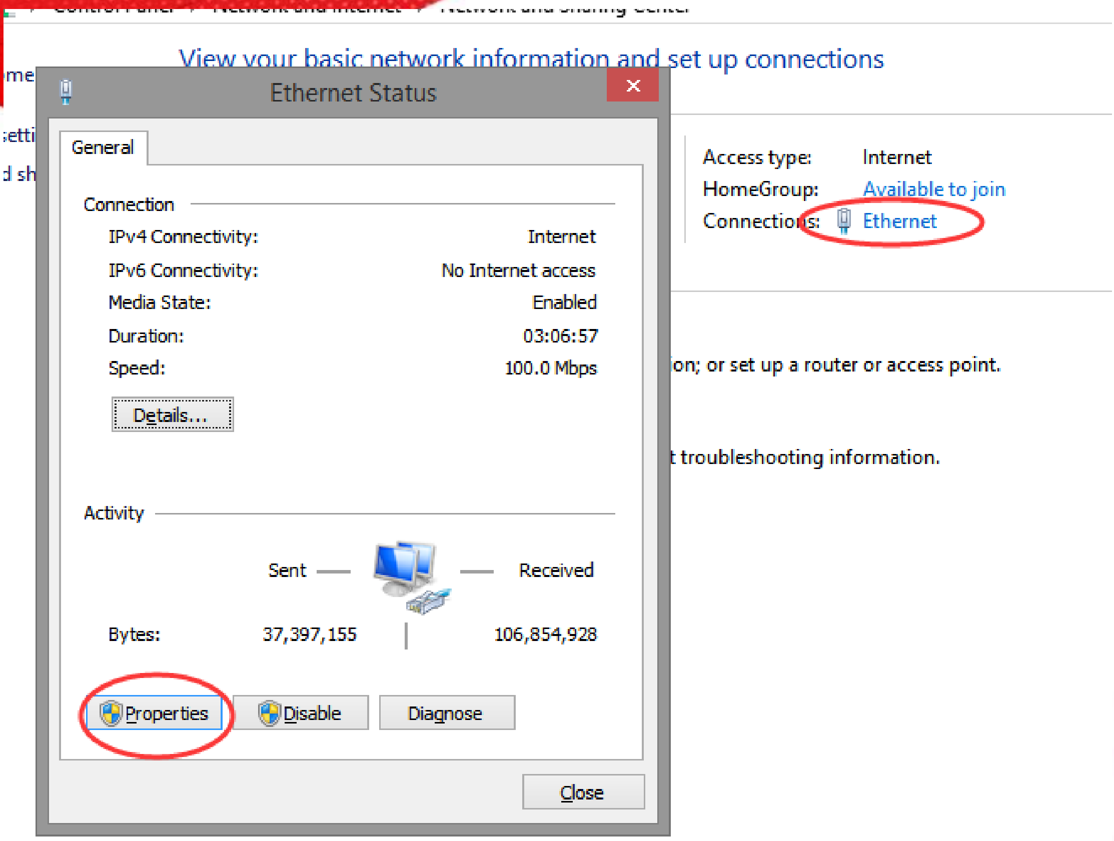
Access type: Internet
HomeGroup: Available to join
Connections: Ethernet

Change your networking settings

Set up a new connection or network
Set up a broadband, dial-up, or VPN connection; or set up a router or access point.

Troubleshoot problems
Diagnose and repair network problems, or get troubleshooting information.

2. Click the Ethernet connections, choose properties:



View your basic network information and set up connections

Ethernet Status

General

Connection

IPv4 Connectivity: Internet

IPv6 Connectivity: No Internet access

Media State: Enabled

Duration: 03:06:57

Speed: 100.0 Mbps

Activity

Sent — Received

Bytes: 37,397,155 | 106,854,928

Access type: Internet

HomeGroup: Available to join

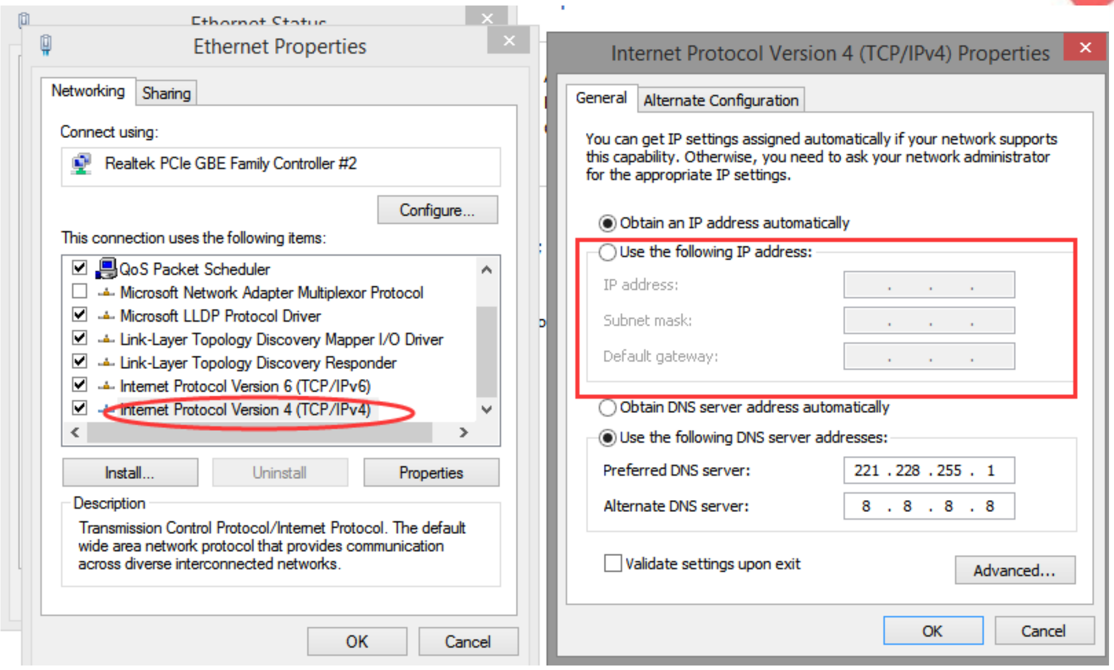
Connections: Ethernet

on; or set up a router or access point.

t troubleshooting information.

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.



Ethernet Properties

Networking | Sharing

Connect using:

Realtek PCIe GBE Family Controller #2

This connection uses the following items:

- ☑ QoS Packet Scheduler
- □ Microsoft Network Adapter Multiplexor Protocol
- ☑ Microsoft LLDP Protocol Driver
- ☑ Link-Layer Topology Discovery Mapper I/O Driver
- ☑ Link-Layer Topology Discovery Responder
- ☑ Internet Protocol Version 6 (TCP/IPv6)
- ☑ Internet Protocol Version 4 (TCP/IPv4)

Description

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

Internet Protocol Version 4 (TCP/IPv4) Properties

General | Alternate Configuration

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

● Obtain an IP address automatically

○ Use the following IP address:

IP address: . . .

Subnet mask: . . .

Default gateway: . . .

○ Obtain DNS server address automatically

● Use the following DNS server addresses:

Preferred DNS server: 221 . 228 . 255 . 1

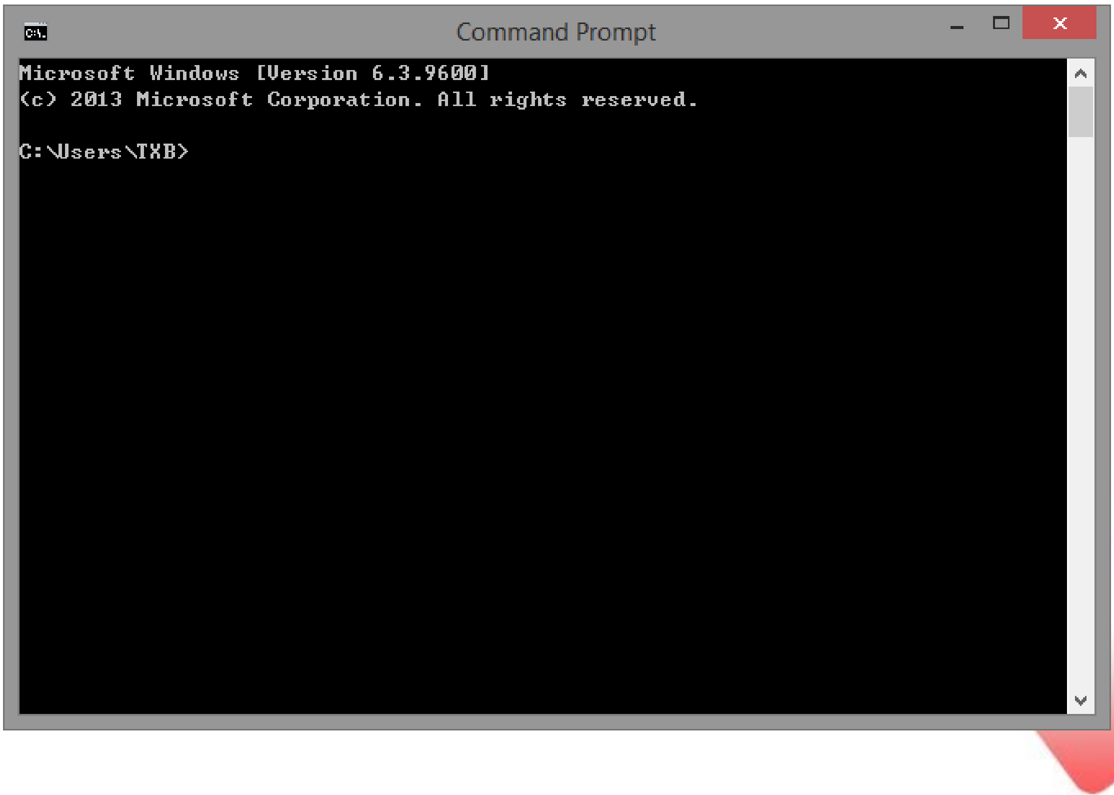
Alternate DNS server: 8 . 8 . 8 . 8

□ Validate settings upon exit

1-1-3. PING command

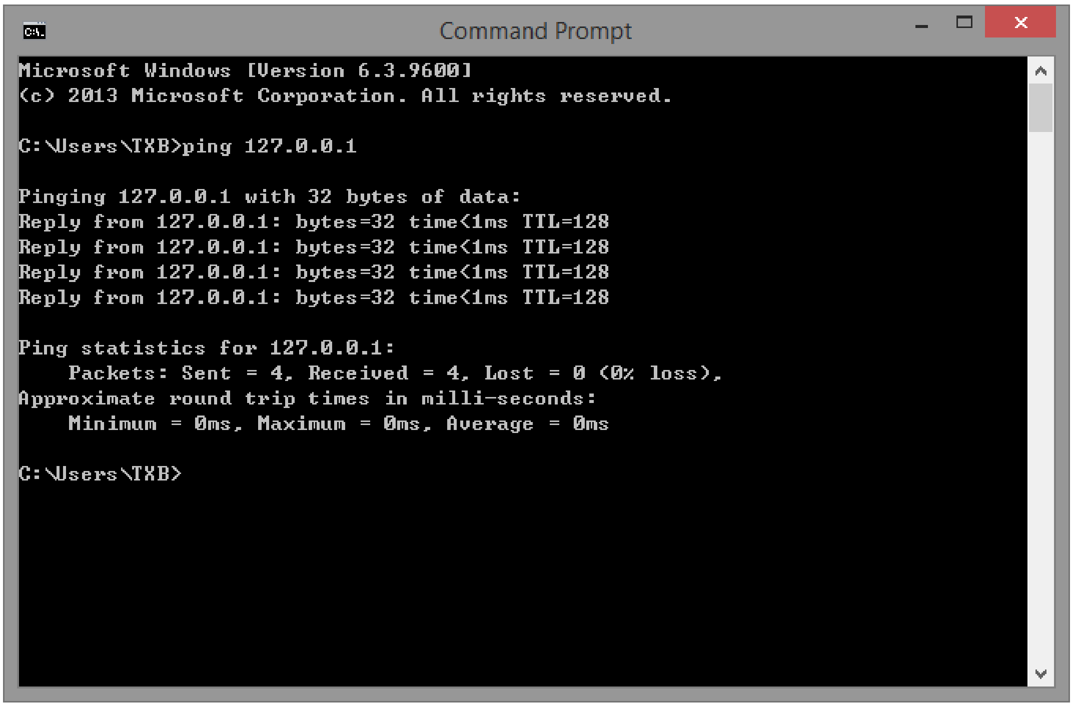
Through the PING command, you can check the local TCP/IP protocol, and whether it can be normal connection to other computer local area network (LAN).

1. open the command prompt



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\TXB>
```

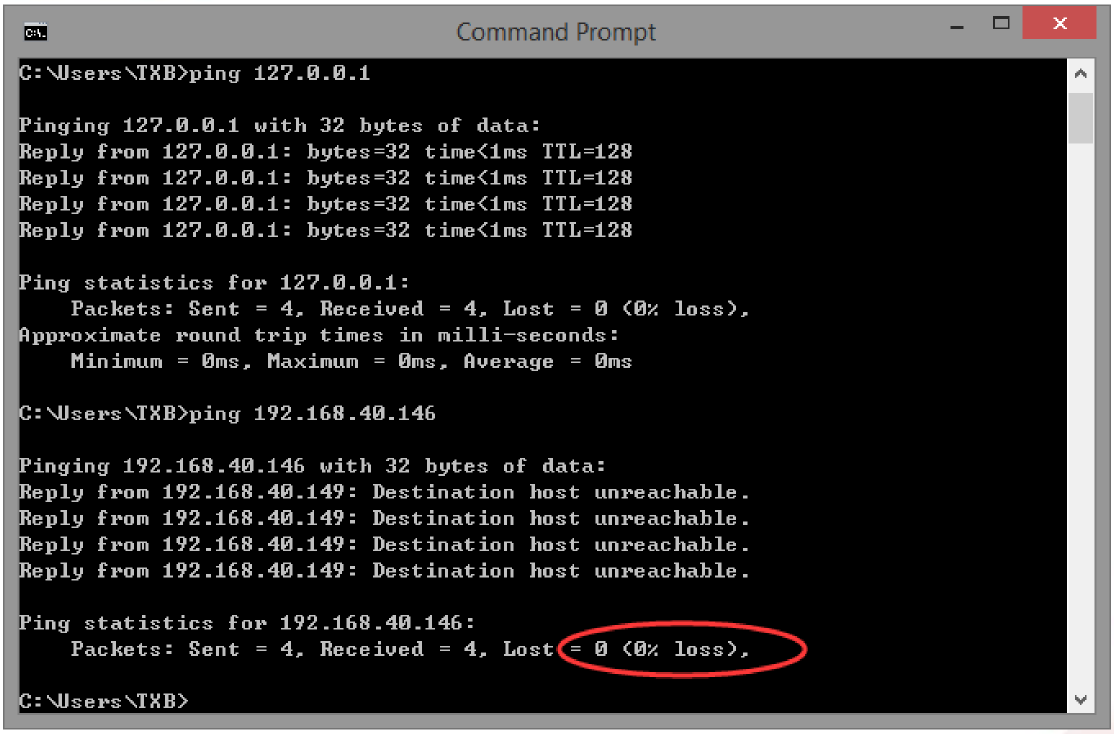
2. input “ping 127.0.0.1” to check the local TCP/IP protocol, it is normal when the receiving and sending data are same.



```
Microsoft Windows [Version 6.3.9600]
(c) 2013 Microsoft Corporation. All rights reserved.

C:\Users\TXB>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\TXB>
```

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.



```
C:\Users\TXB>ping 127.0.0.1

Pinging 127.0.0.1 with 32 bytes of data:
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128
Reply from 127.0.0.1: bytes=32 time<1ms TTL=128

Ping statistics for 127.0.0.1:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
    Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\TXB>ping 192.168.40.146

Pinging 192.168.40.146 with 32 bytes of data:
Reply from 192.168.40.149: Destination host unreachable.
Reply from 192.168.40.149: Destination host unreachable.
Reply from 192.168.40.149: Destination host unreachable.
Reply from 192.168.40.149: Destination host unreachable.

Ping statistics for 192.168.40.146:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),

C:\Users\TXB>
```

(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.

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.

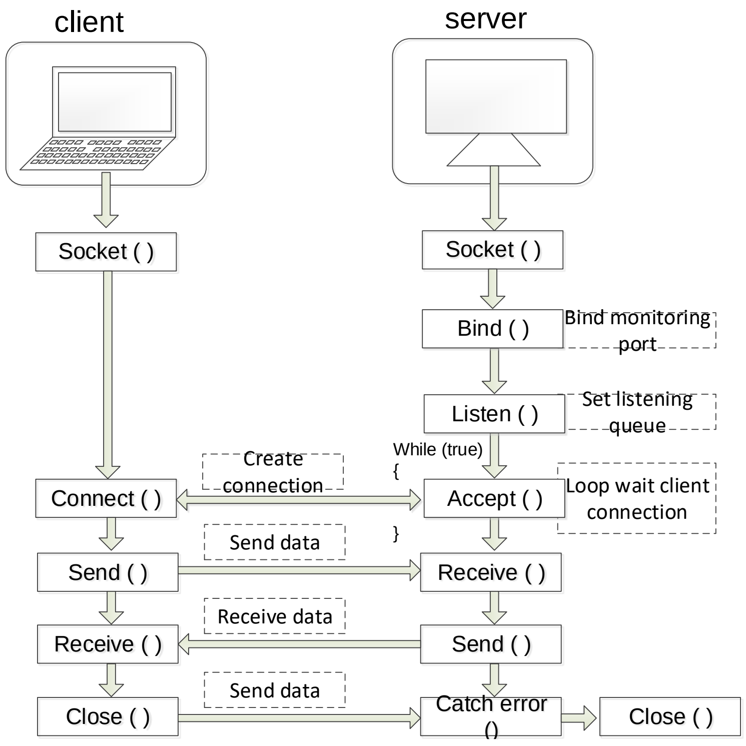
Server monitoring: the server socket does not locate specific client socket, but is in a state of waiting for the connection, and real-time monitors network state, waits for the client's connection request.

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".

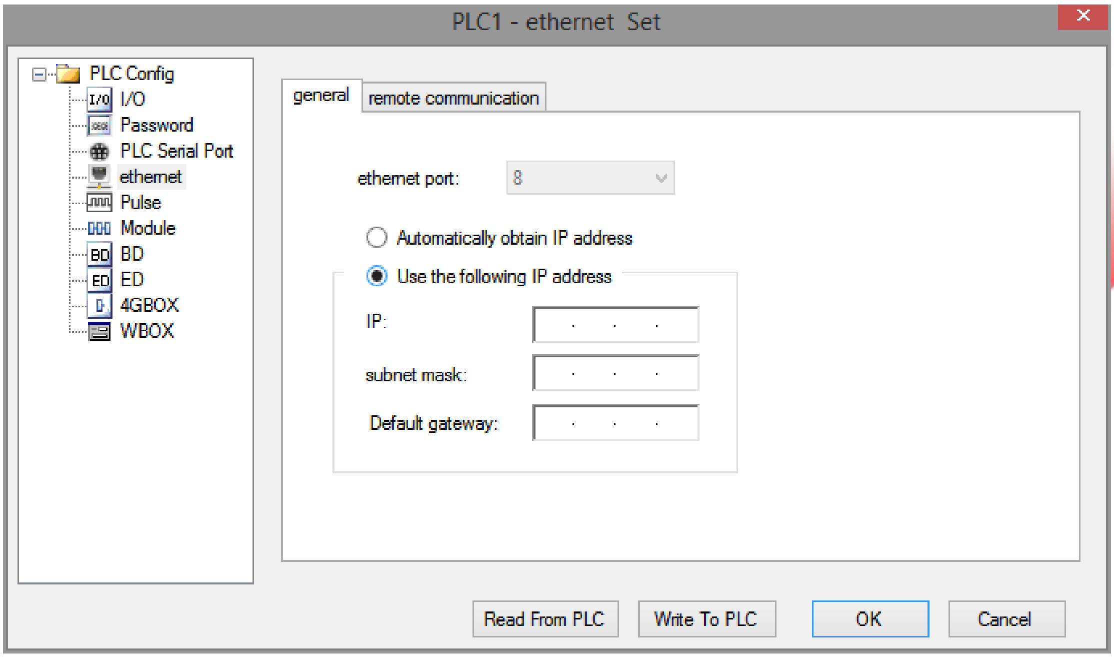
After the connection is established, the client and the server can send and receive data, after data transceiver is completed, the client or the server can request to close the connection, after the fourth "handshake", TCP connection is closed, all data transceiver interrupts.

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.



ethernet port: 8

○ Automatically obtain IP address

● Use the following IP address

IP: . . .

subnet mask: . . .

Default gateway: . . .

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

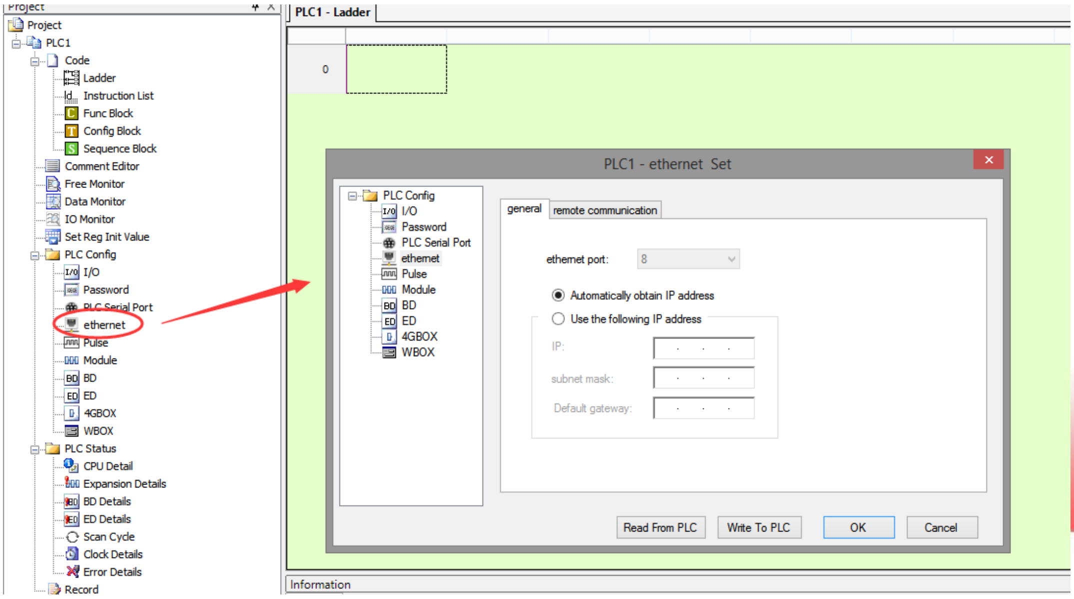
The default gateway: 192.168.6.1

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

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.



ethernet port: 8

● Automatically obtain IP address

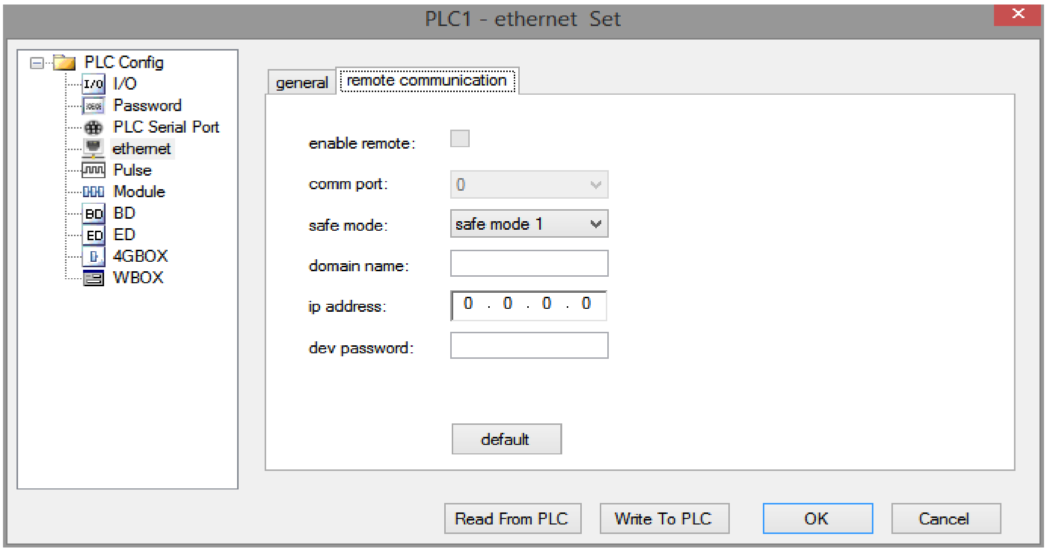
○ Use the following IP address

IP: . . .

subnet mask: . . .

Default gateway: . . .

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



enable remote: □

comm port: 0

safe mode: safe mode 1

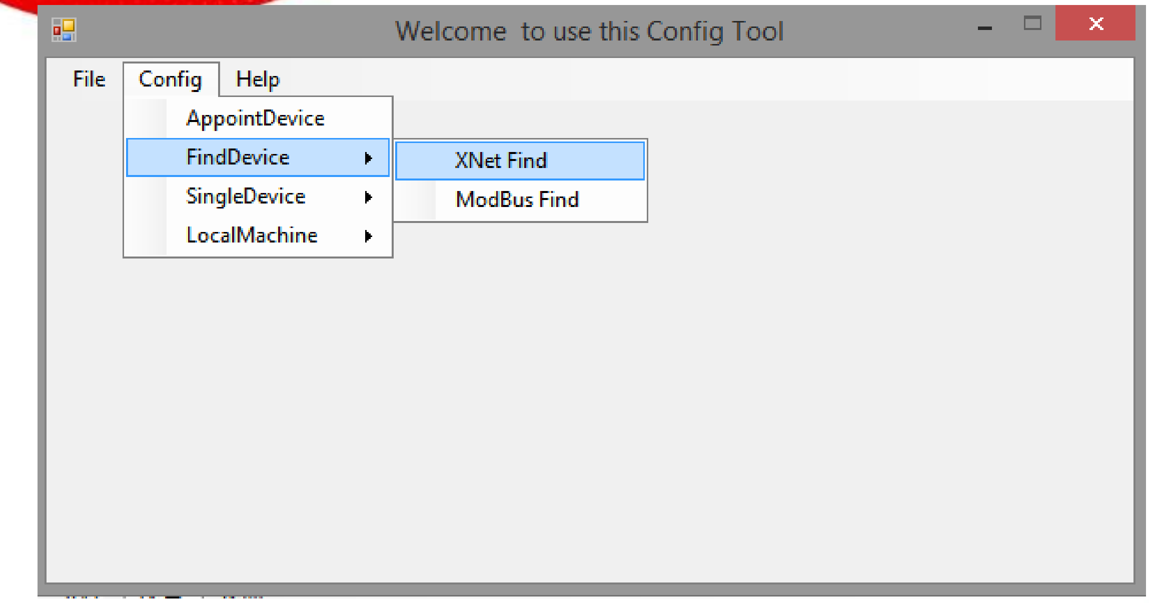
domain name:

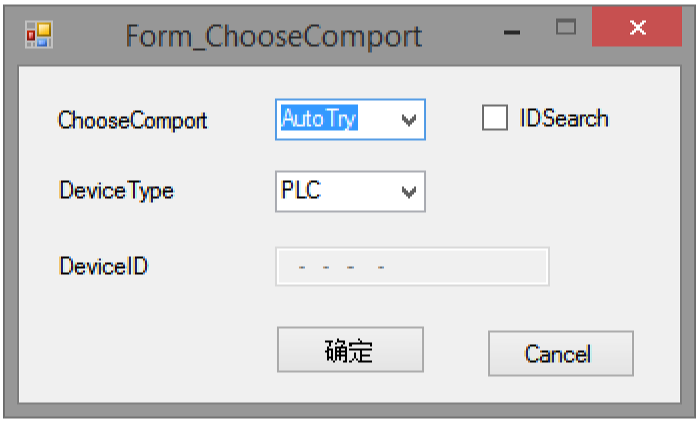
ip address: 0 . 0 . 0 . 0

dev password:

2-3. Configure ethernet parameters in XINJEConfig

Connect the PLC to PC, open the XINJE config tool, click config/find device/xnet find.





ChooseComport: AutoTry

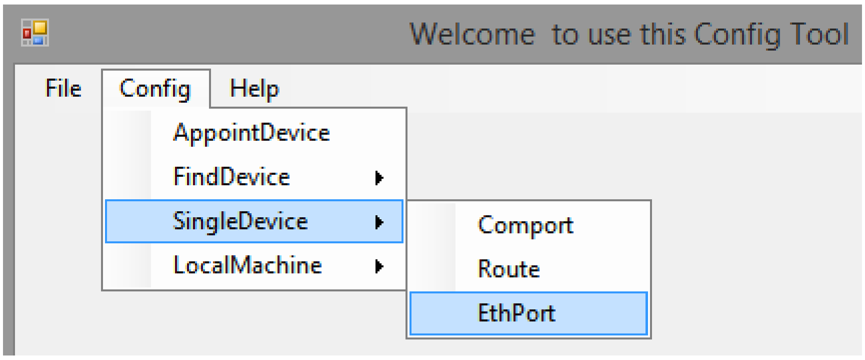
□ IDSearch

DeviceType: PLC

DeviceID: - - - -

If there is no error message, it means the connection is successful.

Click config/single device/etherport, set the Ethernet parameters. Please refer to chapter 2-1-1 for details.



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 no more than four TCP connections.

3-2-2. MODBUS address

When the programmable controller is seemed as the Modbus server, internal soft component number and its corresponding Modbus address number can refer to XINJE PLC programming manual "XD/XL series PLC instruction user manual" and "XG series PLC instruction user manual.

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 120
04H	Read input register	Read 3X address, max quantity is 120
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 2000
10H	Write multiple registers	Write 4X address, max quantity is 120

3-3. 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	XDE, XD5E, XG, XL5E
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	●											

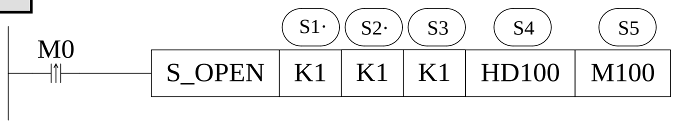
Bit

operand	System						
	X	Y	M*	S*	T*	C*	Dn.m
S5			●				

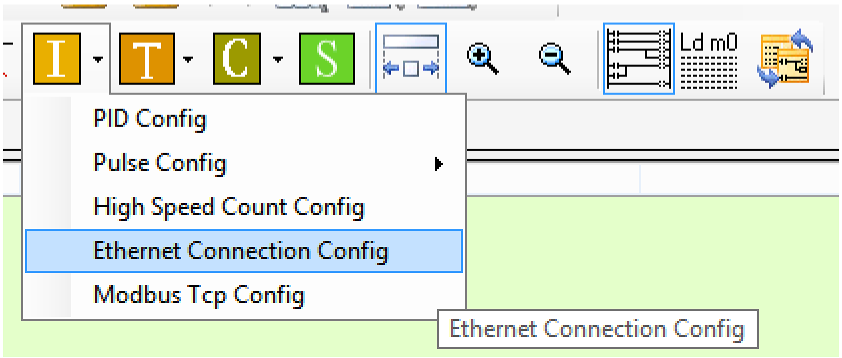
***Note: D means D　HD ；TD means TD　HTD ；CD means CD　HCD　HSCD　HSD;　DM means DM　DHM;**

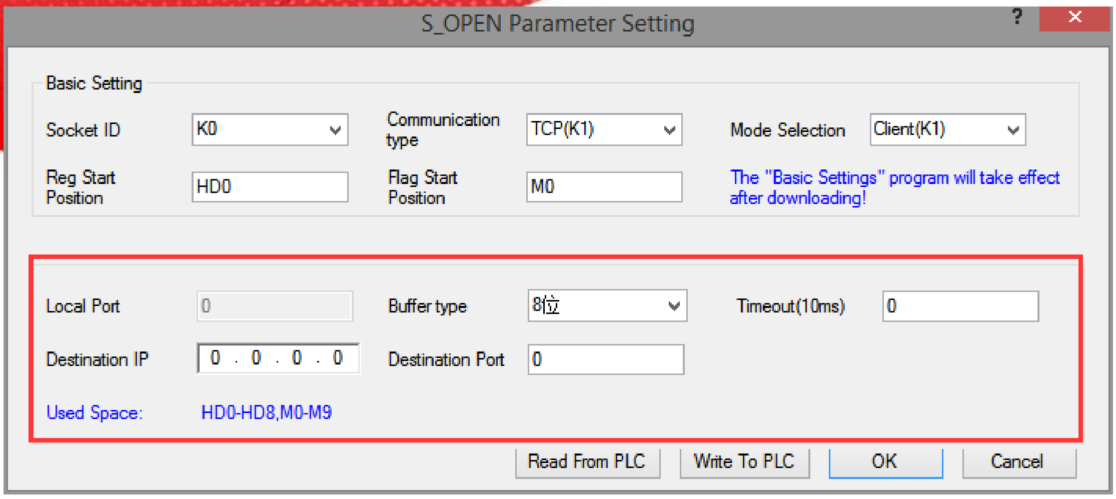
DS means DS DHS. M means M　HM　SM；S means S　HS；　T means T HT；C means C　HC.

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





S_OPEN Parameter Setting

Basic Setting

Socket ID: K0

Communication type: TCP(K1)

Mode Selection: Client(K1)

Reg Start Position: HD0

Flag Start Position: M0

The "Basic Settings" program will take effect after downloading!

Local Port: 0

Buffer type: 8位

Timeout(10ms): 0

Destination IP: 0 . 0 . 0 . 0

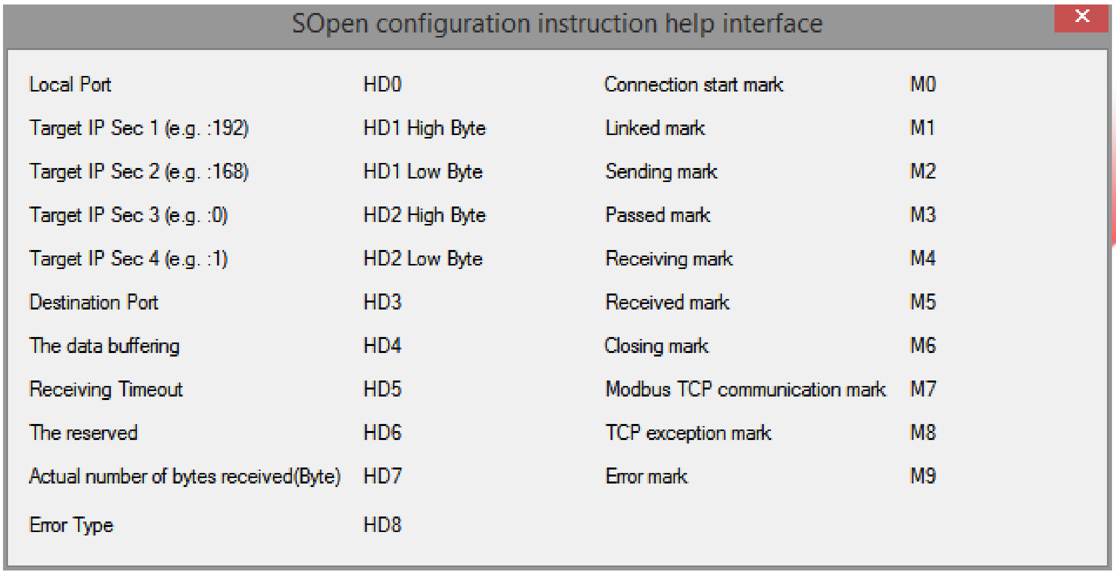
Destination Port: 0

Used Space: HD0-HD8,M0-M9

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:



SOpen configuration instruction help interface

Local Port	HD0	Connection start mark	M0
Target IP Sec 1 (e.g. :192)	HD1 High Byte	Linked mark	M1
Target IP Sec 2 (e.g. :168)	HD1 Low Byte	Sending mark	M2
Target IP Sec 3 (e.g. :0)	HD2 High Byte	Passed mark	M3
Target IP Sec 4 (e.g. :1)	HD2 Low Byte	Receiving mark	M4
Destination Port	HD3	Received mark	M5
The data buffering	HD4	Closing mark	M6
Receiving Timeout	HD5	Modbus TCP communication mark	M7
The reserved	HD6	TCP exception mark	M8
Actual number of bytes received(Byte)	HD7	Error mark	M9
Error Type	HD8		

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 device IP which is in the same subnet of local device.

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. Buffer type

When the value is 0, it is 8-bit mode. When the vlaue is non-zero, it is 16-bit mode. If the actual receiving package is larger than setting length, it will automatical change to 16-bit mode.

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.

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.

7. Data receiving length

Execute S_RCV, the actual received data length, the unit is byte.

8. Error code

The error message of Ethernet free format communication and Modbus TCP communication please refer to chapter 4-4.

9. Flag bit

The communication flag bits are shown as below (take head address Mn as an example)

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	XDE, XD5E, XG, XL5E
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

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

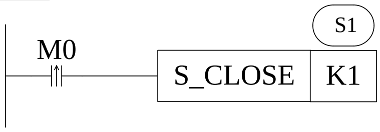
3. Suitable soft component

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	XDE, XD5E, XG, XL5E
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

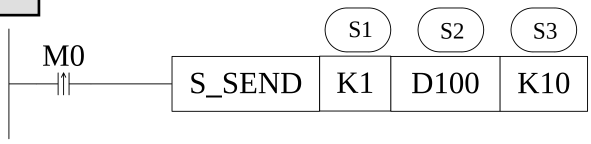
3. Suitable soft component

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
- Please pay attention to the S_OPEN data buffer type in the socket ID.

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	XDE, XD5E, XG, XL5E
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

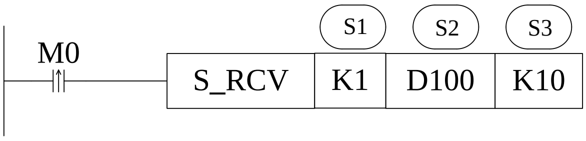
3. Suitable soft component

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
- Please pay attention to the S_OPEN data buffer type in the socket ID.

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	XDE, XD5E, XG, XL5E
Firmware	V3.5.3 and up	Software	V3.5.3 and up

2. Operand

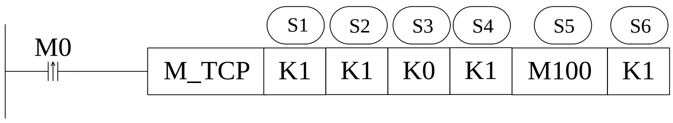
Operand	Function	Model
S1	Remote station no.	16-bit, BIN
S2	Modbus communication function code	16-bit, BIN
S3	Target head address	16-bit, BIN
S4	Register or coil quantity	16-bit, BIN
S5	Local head address	16-bit, BIN
S6	Socket ID	16-bit, BIN

3. Suitable soft component

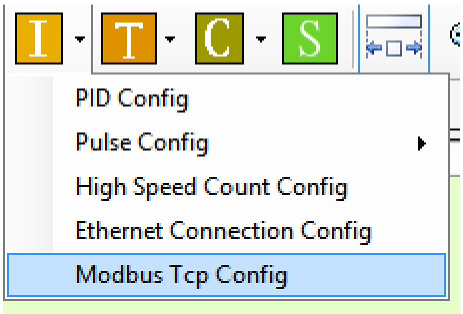
Word	operand	System									Constant	Module	
		D	FD	ED	TD	CD	DX	DY	DM	DS	K/H	ID	QD
	S1	●									●		
	S2	●									●		
	S3	●									●		
	S4	●									●		
	S5	●											
	S6	●									●		

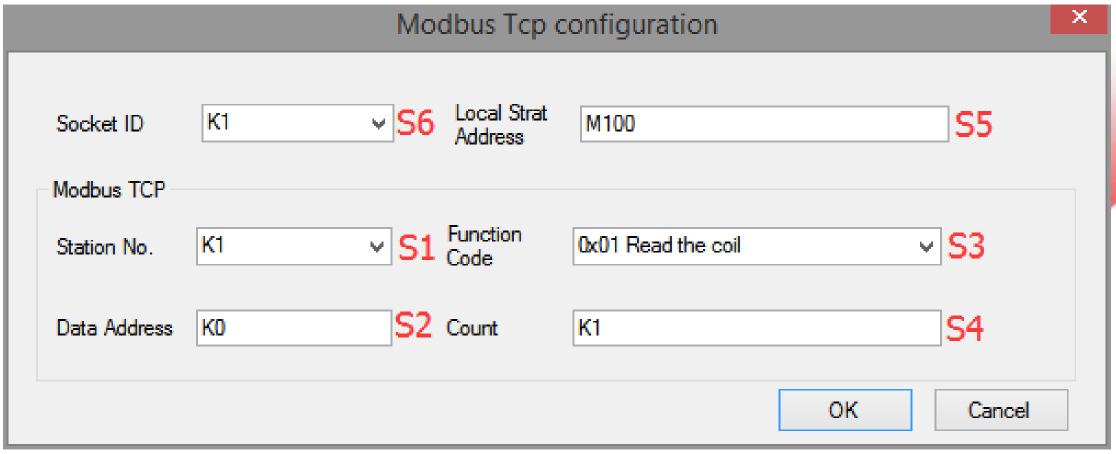
***Note: D means D HD ; TD means TD HTD ; CD means CD HCD HSCD HSD; DM means DM DHM; DS means DS DHS.**

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.
- This instruction needs to set through the following window




Modbus Tcp configuration

Socket ID: K1 S6
Local Strat Address: M100 S5

Modbus TCP

Station No.: K1 S1
Function Code: 0x01 Read the coil S3
Data Address: K0 S2
Count: K1 S4

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

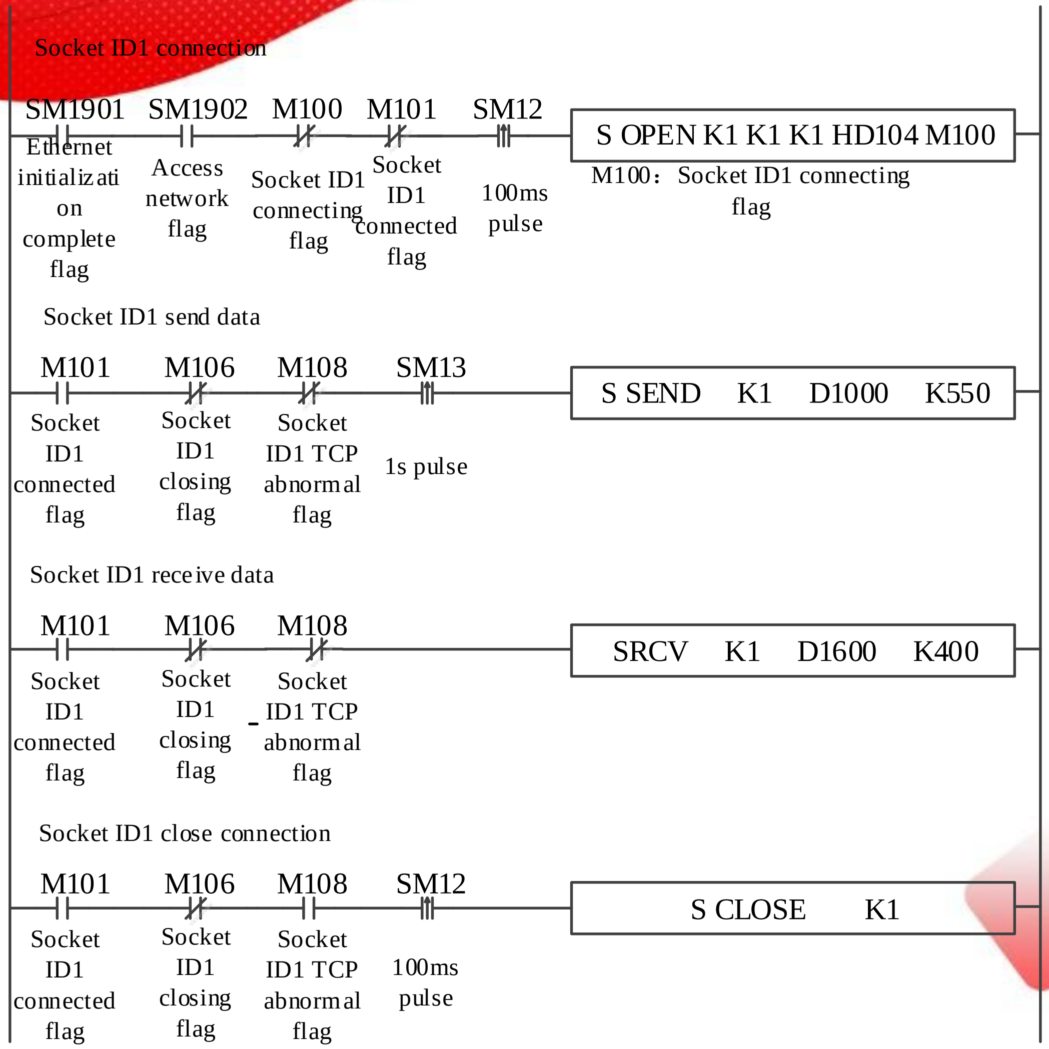
Example 1:
After PLC power on, automatically create TCP client, TCP server and UDP communication task, and send and receive data on the basis of each communication task. The IP address of the PLC is 192.168.0.60, the IP address of target device B is 192.168.0.100.

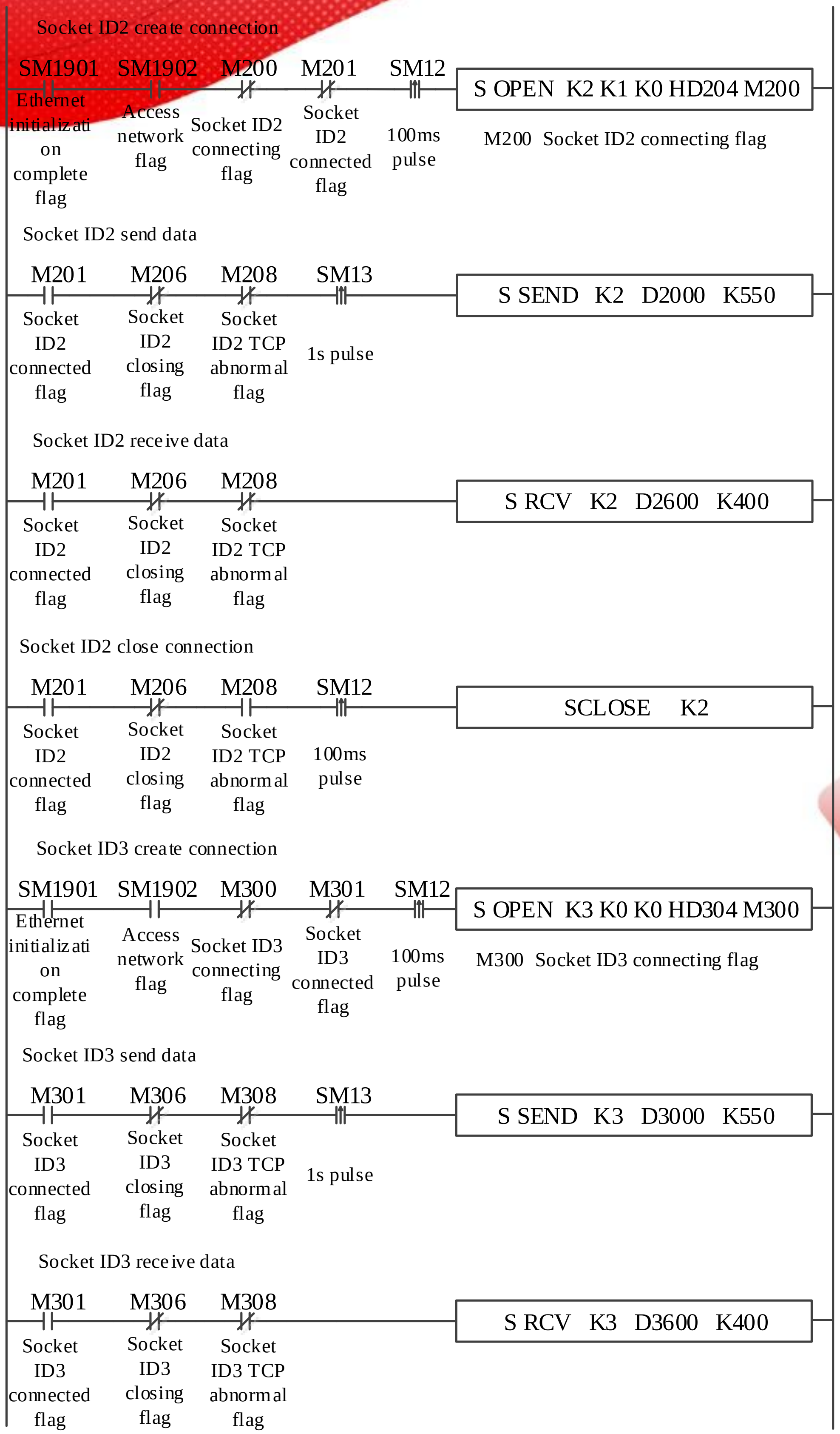
Program operation:
(1) after PLC powers on, the PLC which is TCP client creates TCP connection with TCP server port 1000 of device B and binds the socket ID1. After the connection is created, it sends low 8-bit data of D1000 to D1549 to device B every 1 second, and receives the data and saves in low 8-bit of D1600 to D1999. When the TCP connection is abnormal, it will close the TCP connection and create again.

(2) after PLC powers on, the PLC which is TCP server listens port 1001, waits TCP client creating TCP connection and binds socket ID2. After the connection is created, it sends low 8-bit data of D2000 to D2549 to connected device every 1 second, and receives the data and saves in low 8-bit of D2600 to D2999. When the TCP connection is abnormal, it will close the TCP connection and create again.

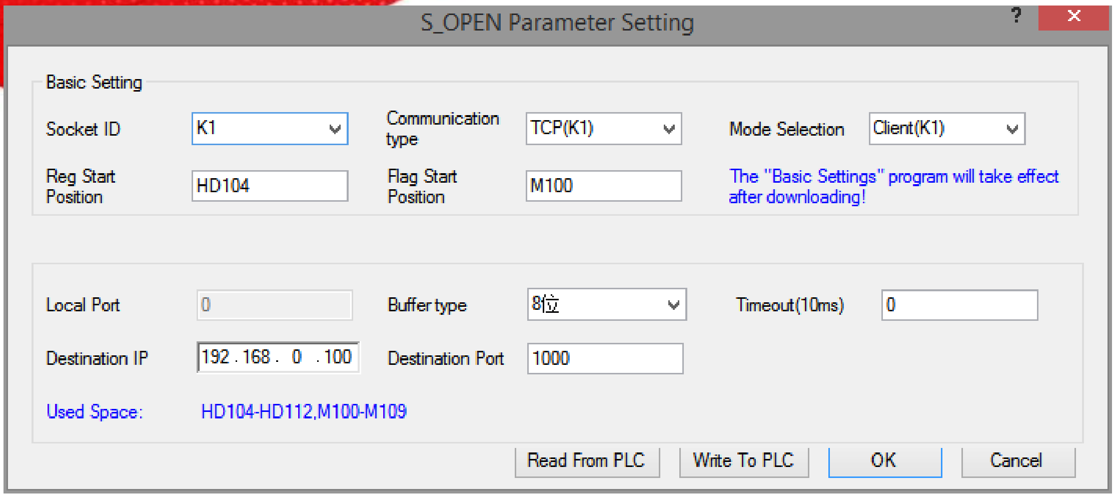
(3) after PLC powers on, it communicates in UDP mode, the local device port is 1002, target IP is 192.168.0.100, target port is 3000, binds the socket ID3. After the connection is created, it sends low 8-bit data of D3000 to D3549 to connected device every 1 second, and receives the data and saves in low 8-bit of D3600 to D3999.

PLC program:


Socket ID1 connection
SM1901
Ethernet initialization complete flag
SM1902
Access network flag
M100
Socket ID1 connecting flag
M101
Socket ID1 connected flag
SM12
100ms pulse
S OPEN K1 K1 K1 HD104 M100
M100：Socket ID1 connecting flag
Socket ID1 send data
M101
Socket ID1 connected flag
M106
Socket ID1 closing flag
M108
Socket ID1 TCP abnormal flag
SM13
1s pulse
S SEND K1 D1000 K550
Socket ID1 receive data
M101
Socket ID1 connected flag
M106
Socket ID1 closing flag
M108
Socket ID1 TCP abnormal flag
SRCV K1 D1600 K400
Socket ID1 close connection
M101
Socket ID1 connected flag
M106
Socket ID1 closing flag
M108
Socket ID1 TCP abnormal flag
SM12
100ms pulse
S CLOSE K1


Socket ID2 create connection
SM1901
Ethernet initialization complete flag
SM1902
Access network flag
M200
Socket ID2 connecting flag
M201
Socket ID2 connected flag
SM12
100ms pulse
S OPEN K2 K1 K0 HD204 M200
M200 Socket ID2 connecting flag
Socket ID2 send data
M201
Socket ID2 connected flag
M206
Socket ID2 closing flag
M208
Socket ID2 TCP abnormal flag
SM13
1s pulse
S SEND K2 D2000 K550
Socket ID2 receive data
M201
Socket ID2 connected flag
M206
Socket ID2 closing flag
M208
Socket ID2 TCP abnormal flag
S RCV K2 D2600 K400
Socket ID2 close connection
M201
Socket ID2 connected flag
M206
Socket ID2 closing flag
M208
Socket ID2 TCP abnormal flag
SM12
100ms pulse
SCLOSE K2
Socket ID3 create connection
SM1901
Ethernet initialization complete flag
SM1902
Access network flag
M300
Socket ID3 connecting flag
M301
Socket ID3 connected flag
SM12
100ms pulse
S OPEN K3 K0 K0 HD304 M300
M300 Socket ID3 connecting flag
Socket ID3 send data
M301
Socket ID3 connected flag
M306
Socket ID3 closing flag
M308
Socket ID3 TCP abnormal flag
SM13
1s pulse
S SEND K3 D3000 K550
Socket ID3 receive data
M301
Socket ID3 connected flag
M306
Socket ID3 closing flag
M308
Socket ID3 TCP abnormal flag
S RCV K3 D3600 K400

Socket ID1 S_OPEN setting:



S_OPEN Parameter Setting

Basic Setting

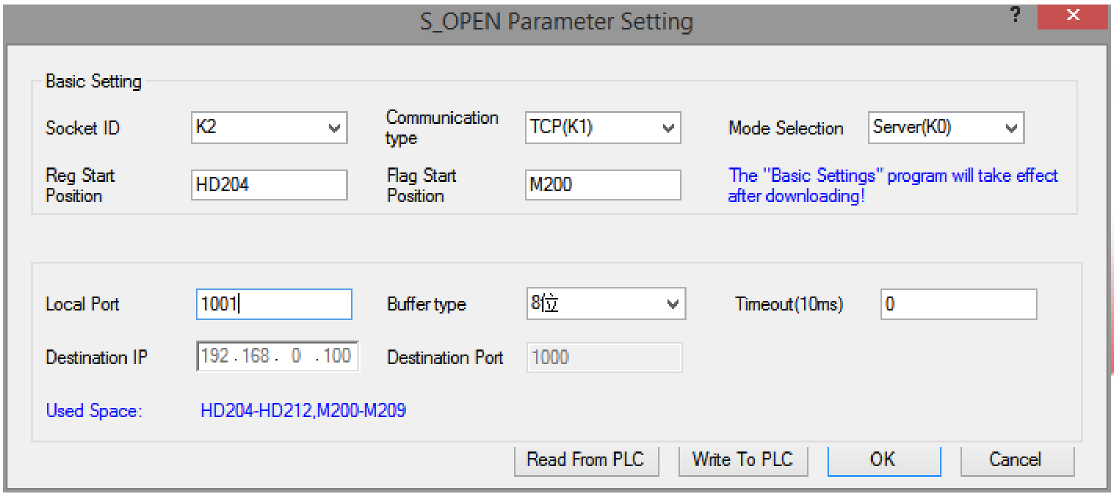
Socket ID: K1
Communication type: TCP(K1)
Mode Selection: Client(K1)
Reg Start Position: HD104
Flag Start Position: M100

The "Basic Settings" program will take effect after downloading!

Local Port: 0
Buffer type: 8位
Timeout(10ms): 0
Destination IP: 192 . 168 . 0 . 100
Destination Port: 1000

Used Space: HD104-HD112,M100-M109

Socket ID2 S_OPEN setting:



S_OPEN Parameter Setting

Basic Setting

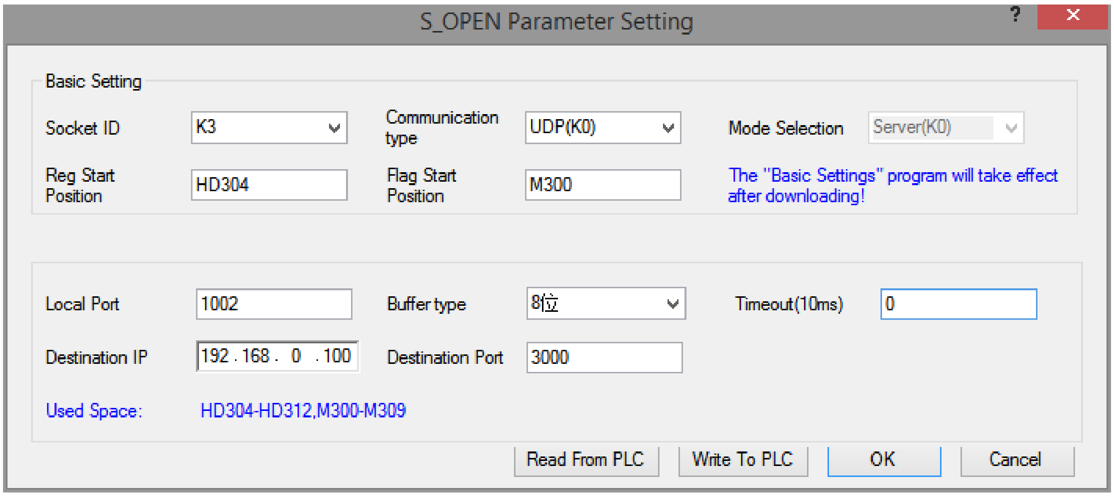
Socket ID: K2
Communication type: TCP(K1)
Mode Selection: Server(K0)
Reg Start Position: HD204
Flag Start Position: M200

The "Basic Settings" program will take effect after downloading!

Local Port: 1001
Buffer type: 8位
Timeout(10ms): 0
Destination IP: 192 . 168 . 0 . 100
Destination Port: 1000

Used Space: HD204-HD212,M200-M209

Socket ID3 S_OPEN setting:



S_OPEN Parameter Setting

Basic Setting

Socket ID: K3
Communication type: UDP(K0)
Mode Selection: Server(K0)
Reg Start Position: HD304
Flag Start Position: M300

The "Basic Settings" program will take effect after downloading!

Local Port: 1002
Buffer type: 8位
Timeout(10ms): 0
Destination IP: 192 . 168 . 0 . 100
Destination Port: 3000

Used Space: HD304-HD312,M300-M309

Example 2:

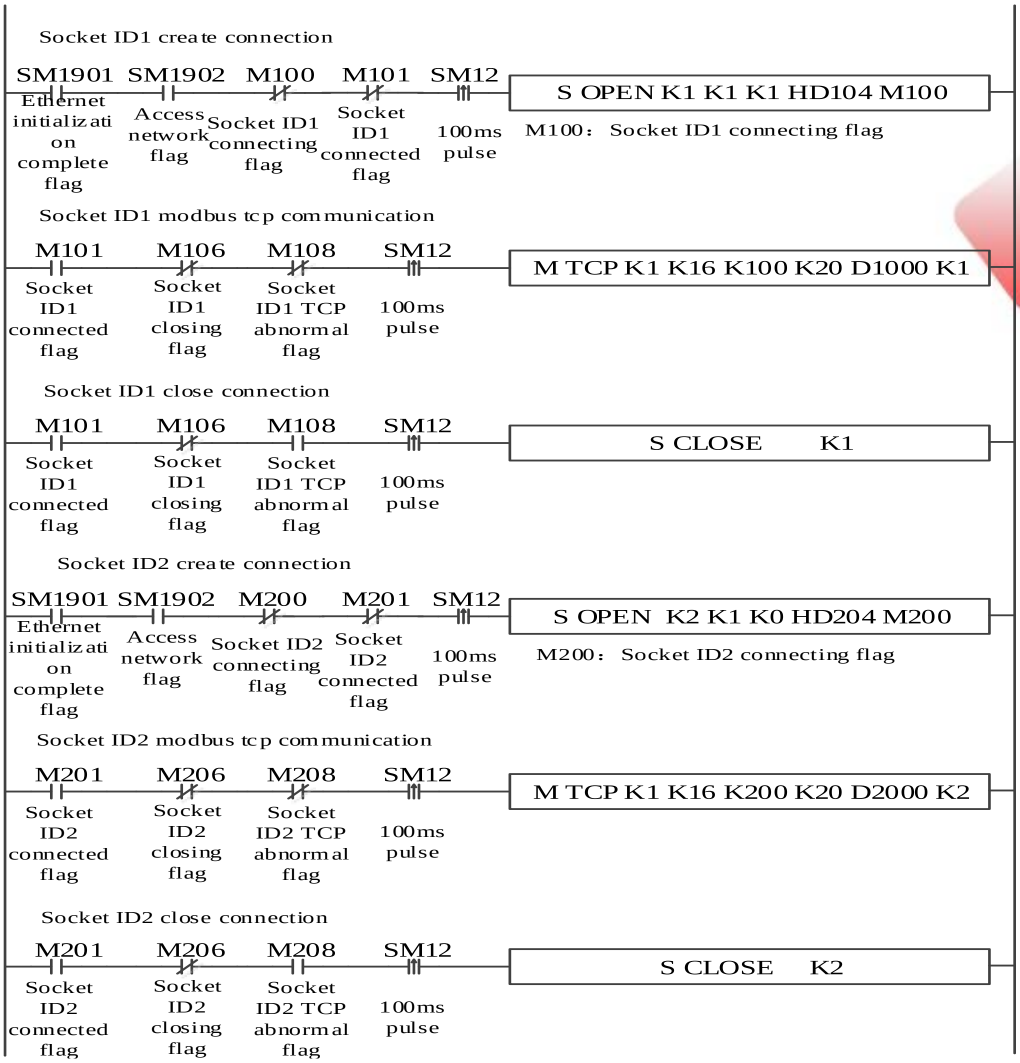
After PLC power on, it will communicate with Modbus TCp server device A and B, the PLC IP is 192.168.0.60, device A IP is 192.168.0.40, Modbus station no. is 1, device B IP is 192.168.0.70, Modbus station no. is 1.

Program operation:

(1) After PLC power on, as the TCP client, it will create TCP connection with TCP server port 502 of device A, and bind the socket ID1, after the connection is created successfully, it will write the value of D1000 to D1019 to device A address 4x100 to 4x119 every 100ms. When the TCP connection is abnormal, it will close the TCP connection and create again.

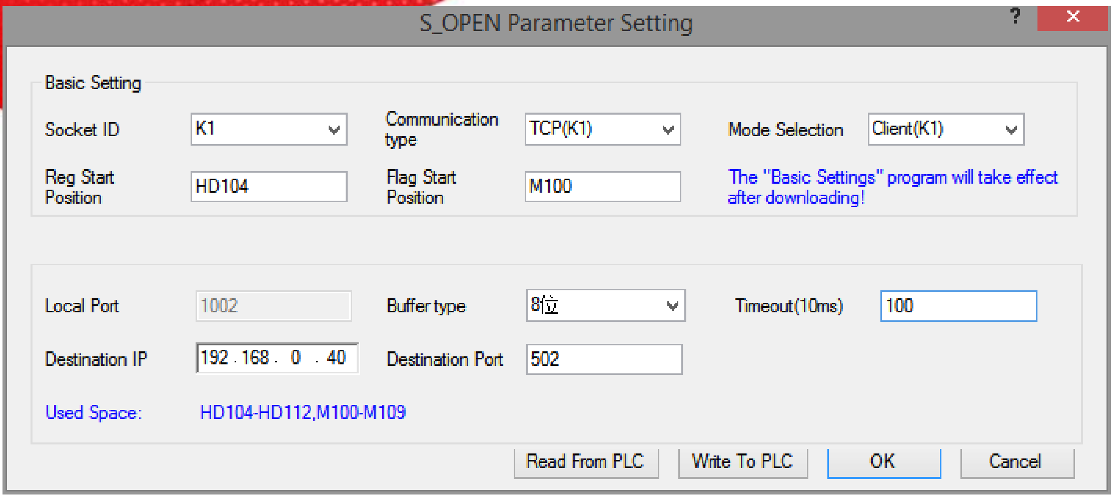
(2) After PLC power on, as the TCP client, it will create TCP connection with TCP server port 502 of device B, and bind the socket ID2, after the connection is created successfully, it will write the value of D2000 to D2019 to device B address 4x200 to 4x219 every 100ms. When the TCP connection is abnormal, it will close the TCP connection and create again.

Program:



Socket ID1 create connection
SM1901
Ethernet initialization complete flag
SM1902
Access network flag
M100
Socket ID1 connecting flag
M101
Socket ID1 connected flag
SM12
100ms pulse
S OPEN K1 K1 K1 HD104 M100
M100： Socket ID1 connecting flag
Socket ID1 modbus tcp communication
M101
Socket ID1 connected flag
M106
Socket ID1 closing flag
M108
Socket ID1 TCP abnormal flag
SM12
100ms pulse
M TCP K1 K16 K100 K20 D1000 K1
Socket ID1 close connection
M101
Socket ID1 connected flag
M106
Socket ID1 closing flag
M108
Socket ID1 TCP abnormal flag
SM12
100ms pulse
S CLOSE K1
Socket ID2 create connection
SM1901
Ethernet initialization complete flag
SM1902
Access network flag
M200
Socket ID2 connecting flag
M201
Socket ID2 connected flag
SM12
100ms pulse
S OPEN K2 K1 K0 HD204 M200
M200： Socket ID2 connecting flag
Socket ID2 modbus tcp communication
M201
Socket ID2 connected flag
M206
Socket ID2 closing flag
M208
Socket ID2 TCP abnormal flag
SM12
100ms pulse
M TCP K1 K16 K200 K20 D2000 K2
Socket ID2 close connection
M201
Socket ID2 connected flag
M206
Socket ID2 closing flag
M208
Socket ID2 TCP abnormal flag
SM12
100ms pulse
S CLOSE K2

Socket ID1 S_OPEN setting:



S_OPEN Parameter Setting

Basic Setting

Socket ID: K1

Communication type: TCP(K1)

Mode Selection: Client(K1)

Reg Start Position: HD104

Flag Start Position: M100

The "Basic Settings" program will take effect after downloading!

Local Port: 1002

Buffer type: 8位

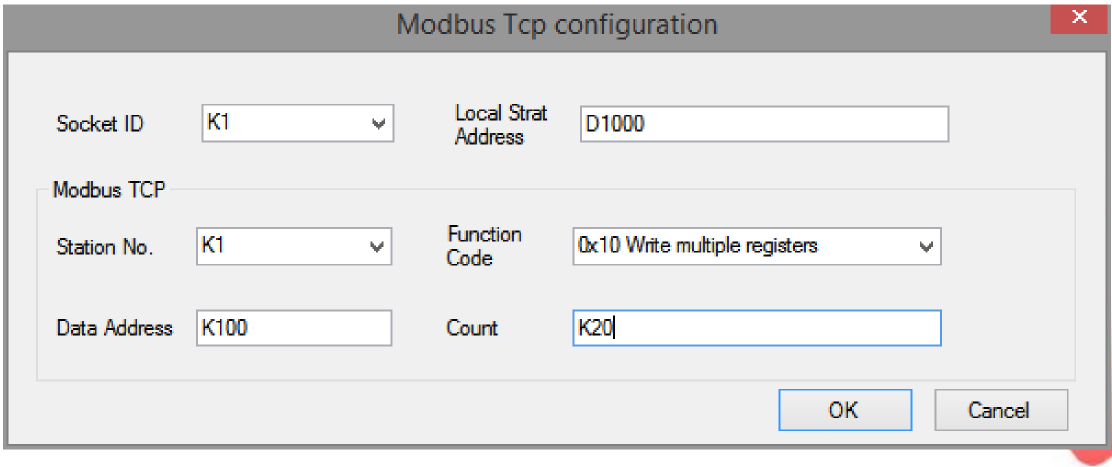
Timeout(10ms): 100

Destination IP: 192 . 168 . 0 . 40

Destination Port: 502

Used Space: HD104-HD112,M100-M109

Socket ID1 M_TCP setting:



Modbus Tcp configuration

Socket ID: K1

Local Strat Address: D1000

Modbus TCP

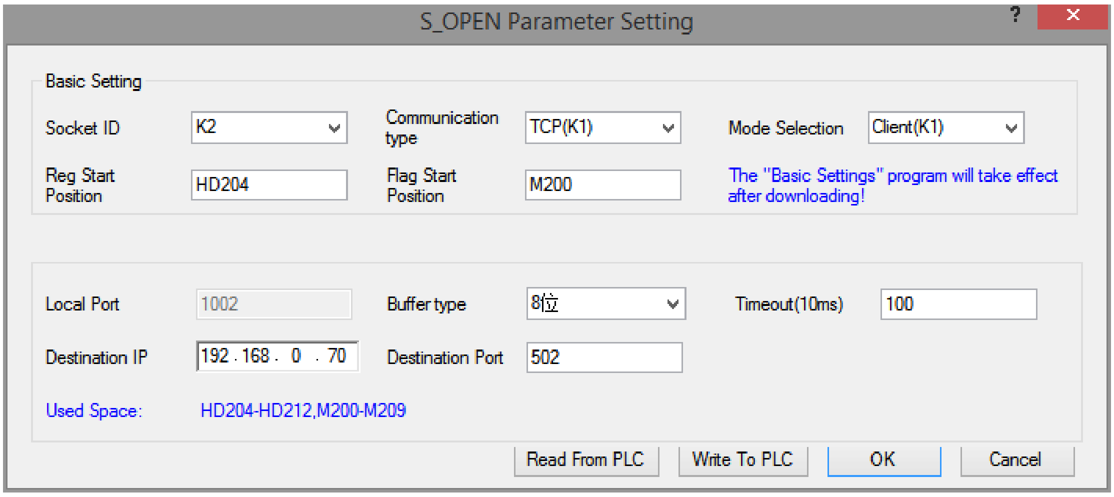
Station No.: K1

Function Code: 0x10 Write multiple registers

Data Address: K100

Count: K20

Socket ID2 S_OPEN setting:



S_OPEN Parameter Setting

Basic Setting

Socket ID: K2

Communication type: TCP(K1)

Mode Selection: Client(K1)

Reg Start Position: HD204

Flag Start Position: M200

The "Basic Settings" program will take effect after downloading!

Local Port: 1002

Buffer type: 8位

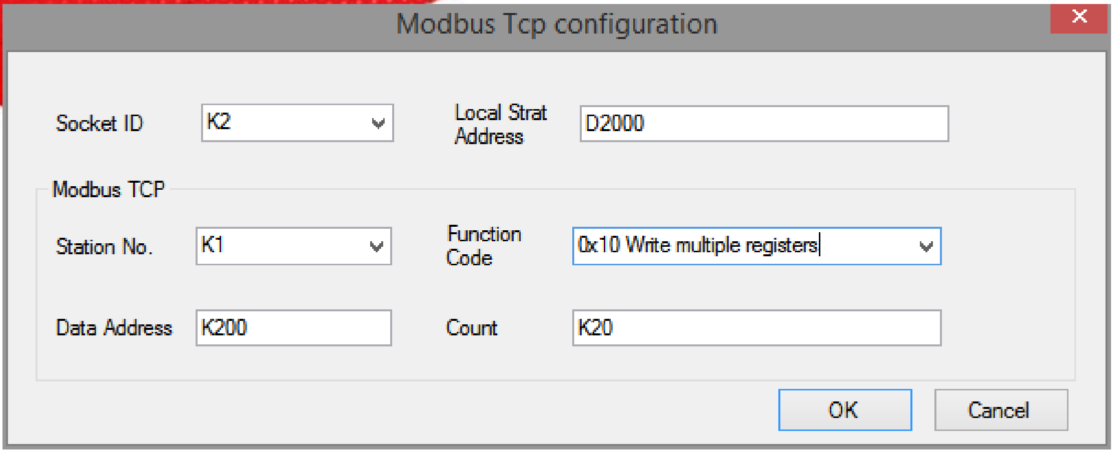
Timeout(10ms): 100

Destination IP: 192 . 168 . 0 . 70

Destination Port: 502

Used Space: HD204-HD212,M200-M209

Socket 2 M_TCP setting:



Modbus Tcp configuration

Socket ID: K2

Local Strat Address: D2000

Modbus TCP

Station No.: K1

Function Code: 0x10 Write multiple registers

Data Address: K200

Count: K20

4-2. Read write instruction for com port

When Ethernet communication is carried out, in order to ensure the normal realization of communication, it is recommended to use communication port parameter reading instruction [CFGCR] and writing instruction [CFGCW] when making communication program.

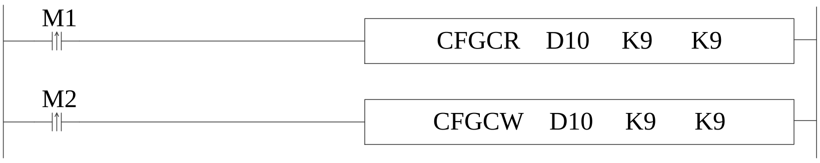
First, by calling the CFGCR instruction, the corresponding parameters of the communication port are read into the specified register group, and then the user modifies the corresponding values in the register group as required, and then writes the modified values of the register group into the corresponding communication port configuration through the CFGCW instruction. Refer to section 6-5 of XD/XL series programmable controller user manual [basic instructions] .

4-2-1. Com port parameters communication example

Example 1:

Through parameter reading instruction [CFGCR] and writing instruction [CFGCW], the network parameters of PLC are read into register D10~D18, and the network parameters of register D10~D18 are written into the serial port setting of PLC after modification.

PLC program:



When M1 is set on, the network parameters of PLC are triggered to read. After modifying the network parameters, set on M2, and the modified network parameters can be written into the PLC. After the writing, the PLC power off and power on again, the serial port parameters take effect.

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

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	Such as 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
40	Receive timeout
50	Specified target port error, MODBUS TCP is not port 502, The using port is out of range(1~60000)
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 (Flash ROM)

Que esse conteúdo tenha agregado valor e conhecimento pra você!

Seu contato é importante para nós!

- www.kalatec.com.br
- Instagram - @kalateceautomação
- Facebook – kalatecautomação

NOSSAS FILIAIS

Matriz Campinas – SP
Rua Salto, 99
Jd. do Trevo
(19) 3045-4900

Filial São Paulo – SP
Av. das Nações Unidas,
18.801 – 11o Andar
(11) 5514-7680

Filial Joinville – SC
R. Almirante Jaceguay, 3659
Bairro Costa e Silva
(47) 3425-0042