
14 Ethernet communication

last modified by Joey

on 2022/06/14 17:37

Table of Contents

Ethernet overview	3
IP address	3
Set PC network address	3
Test the network connection status	4
PLC Editor2 connect to PLC with Ethernet	6
PLC Editor2 Ethernet search function	8
Ethernet configuration	10
Hardware interface	10
Total numbers of links supported	10
IP address settings	11
TCP protocol	13
UDP protocol	13
Socket	14
Establish an Ethernet link by socket	14
LX5V-N socket configuration instructions	15
Ethernet instruction	16
SOCOPEN/Create a socket link	16
SOCCLOSE/Close socket link	18
SOCSEND/Ethernet free-form communication sending	19
SOCRECV/Ethernet free-form communication receiving	20
SOCMTCP/Ethernet ModbusTCP communication	20
Ethernet applications	21
Data exchange between two PLCs through ModbusTCP	21
Data exchange between two PLCs through Free TCP	23
Data exchange between two PLCs through Free UDP	27
List of special device related to Ethernet	30
Ethernet error codes table	38
Operational error	38

Ethernet overview

IP address

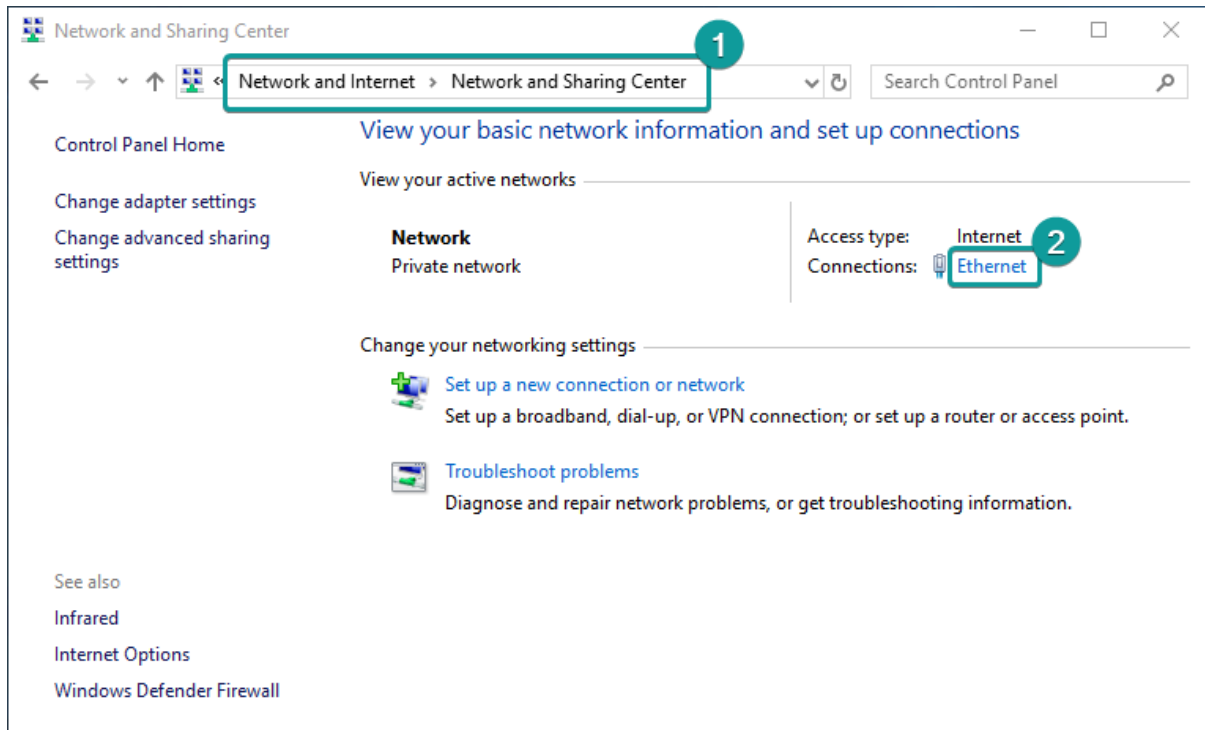
IP address consists of network address and host address, and distinguished by subnet mask. If programming device (such as PC) use network card to connect to LAN, the programming device and PLC must be in the same subnet. You can specify the subnet of a device by combining an IP address with a subnet mask.

The network address could be calculated by performing logic and operation between IP address and subnet mask. If the addresses are in the same network, it means that communication is possible.

Number	Network device 1			Network device 2			Network connection
	IP	Subnet mask	Network address	IP	Subnet mask	Network address	
1	192.168.0.1	255.255.255.0	192.168.0.0	192.168.0.10	255.255.255.0	192.168.0.0	Yes
2	192.168.0.1	255.255.255.0	192.168.0.0	192.168.1.10	255.255.255.0	192.168.1.0	No
3	192.168.0.1	255.255.255.1	192.168.0.1	192.168.0.10	255.255.255.1	192.168.0.0	No

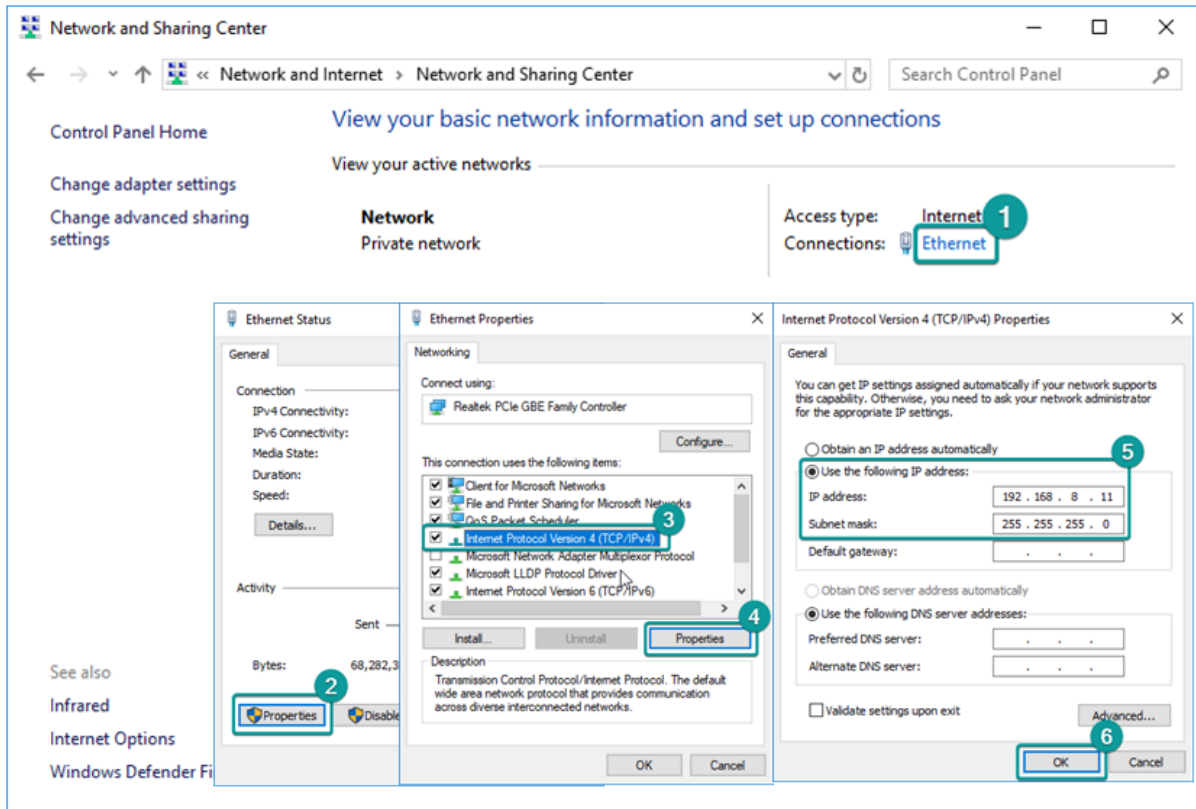
Set PC network address

(1) Click “Control panel”→”Network and Internet”→”Network and sharing center”.



(2) Click “Ethernet”→”Properties”→”Internet protocol version 4”.

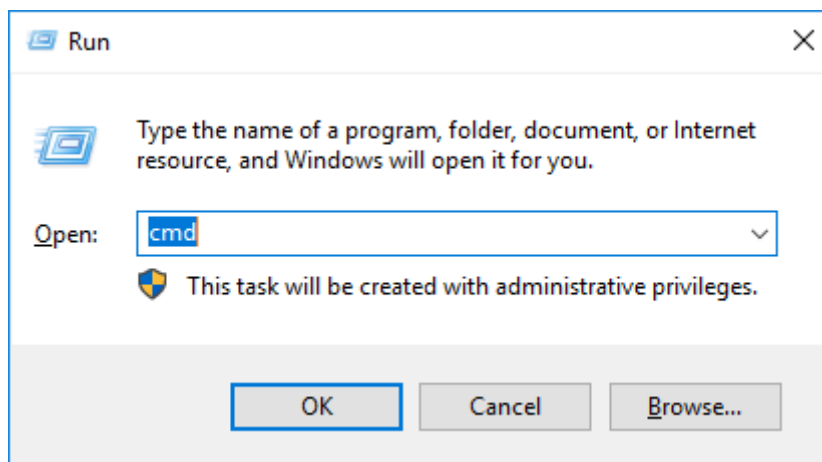
(3) Set the IP address and subnet mask on the same network address as the PLC. The IP address that has been used in LAN could not be set. If the IP of PLC is 192.168.8.8, and the subnet mask is 255.255.255.0. The IP address as shown below could be set to connect PC to PLC.



Test the network connection status

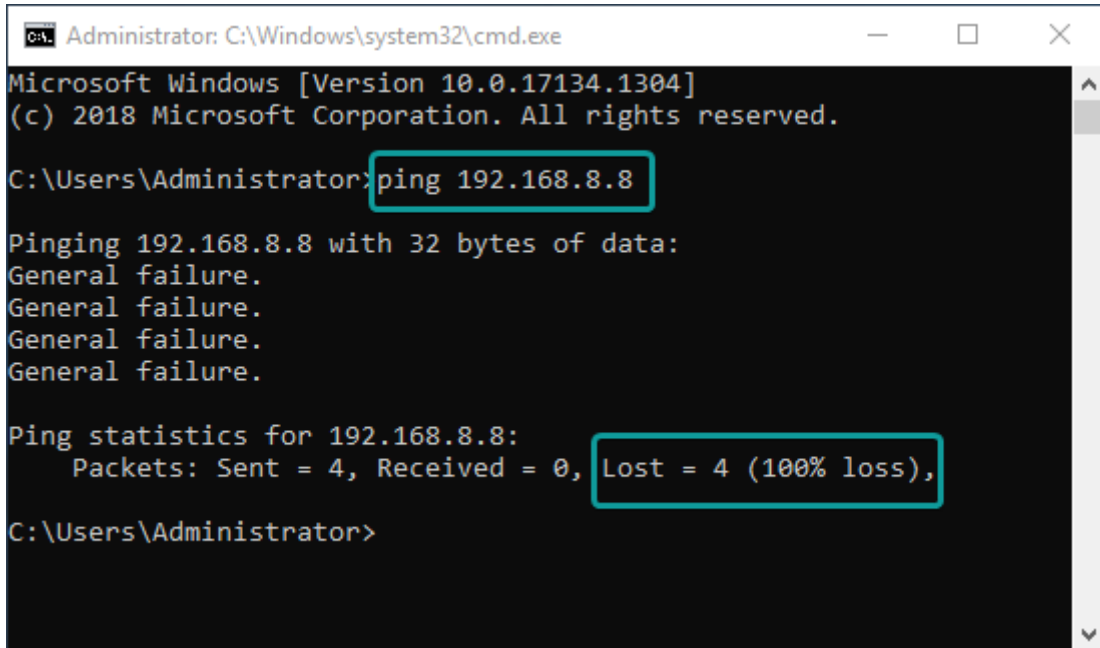
Test the connection status between PC and PLC by ping command.

- (1) Press "WIN" and "R" keys, and input "cmd".



- (2) If the IP address of PLC is 192.168.8.8.

1) Input "ping 192.168.8.8", and enter. If it display "100% loss", it means that PLC could be connected.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

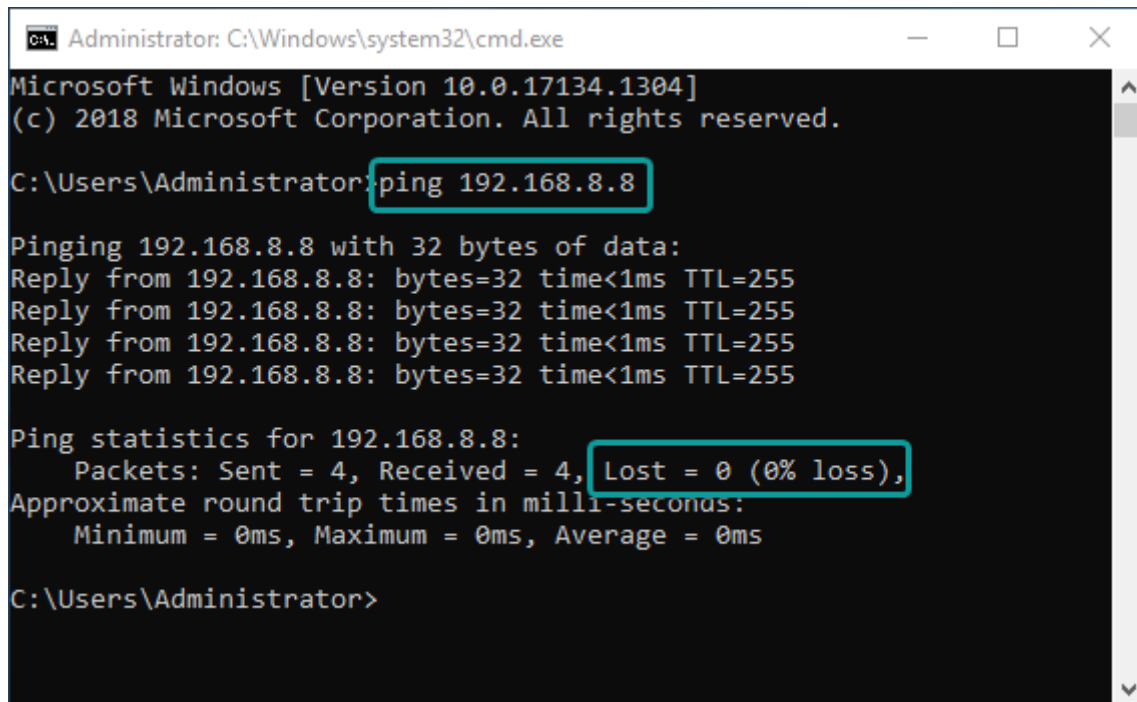
C:\Users\Administrator>ping 192.168.8.8

Pinging 192.168.8.8 with 32 bytes of data:
General failure.
General failure.
General failure.
General failure.

Ping statistics for 192.168.8.8:
    Packets: Sent = 4, Received = 0, Lost = 4 (100% loss),

C:\Users\Administrator>
```

2) Input "ping 192.168.8.8 ", and enter. If it display "0% loss", it means that it could be connected to PLC.



```
Administrator: C:\Windows\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.1304]
(c) 2018 Microsoft Corporation. All rights reserved.

C:\Users\Administrator>ping 192.168.8.8

Pinging 192.168.8.8 with 32 bytes of data:
Reply from 192.168.8.8: bytes=32 time<1ms TTL=255
Reply from 192.168.8.8: bytes=32 time<1ms TTL=255
Reply from 192.168.8.8: bytes=32 time<1ms TTL=255
Reply from 192.168.8.8: bytes=32 time<1ms TTL=255

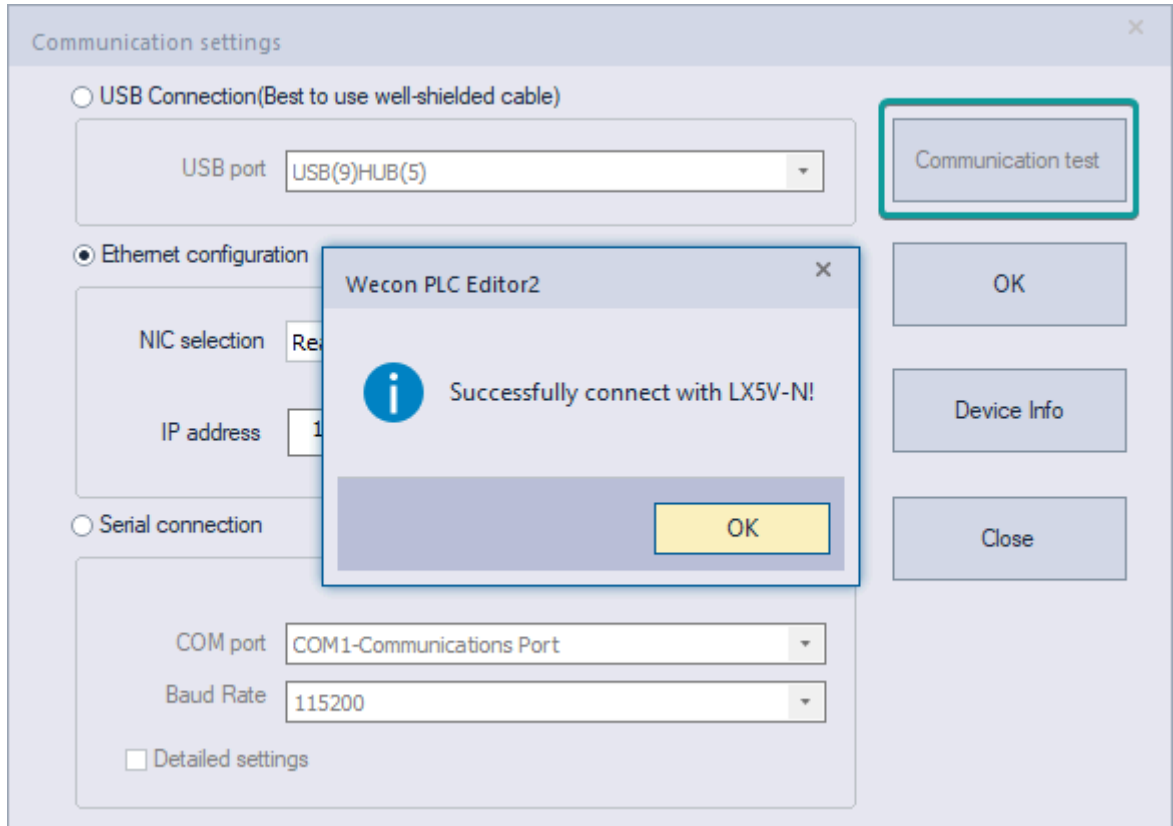
Ping statistics for 192.168.8.8:
    Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
    Approximate round trip times in milli-seconds:
        Minimum = 0ms, Maximum = 0ms, Average = 0ms

C:\Users\Administrator>
```

3) The command "ping network device IP" could only be use four times. To ping network devices continuously, run "ping network device ip-t" command, it is shown as below.

The screenshot shows a 'Communication settings' dialog box with three main sections: 'USB Connection', 'Ethernet configuration', and 'Serial connection'. The 'Ethernet configuration' section is selected with a radio button. It contains a 'NIC selection' dropdown menu set to 'Realtek PCIe GBE Family Controller', an 'IP address' text field containing '192 . 168 . 8 . 8' (highlighted with a red box), and a 'Devices search' button. The 'USB Connection' section is unselected and shows a 'USB port' dropdown set to 'USB(9)HUB(5)'. The 'Serial connection' section is unselected and shows a 'COM port' dropdown set to 'COM1-Communications Port' and a 'Baud Rate' dropdown set to '115200'. There is also an unchecked 'Detailed settings' checkbox. On the right side of the dialog, there are four buttons: 'Communication test', 'OK', 'Device Info', and 'Close'.

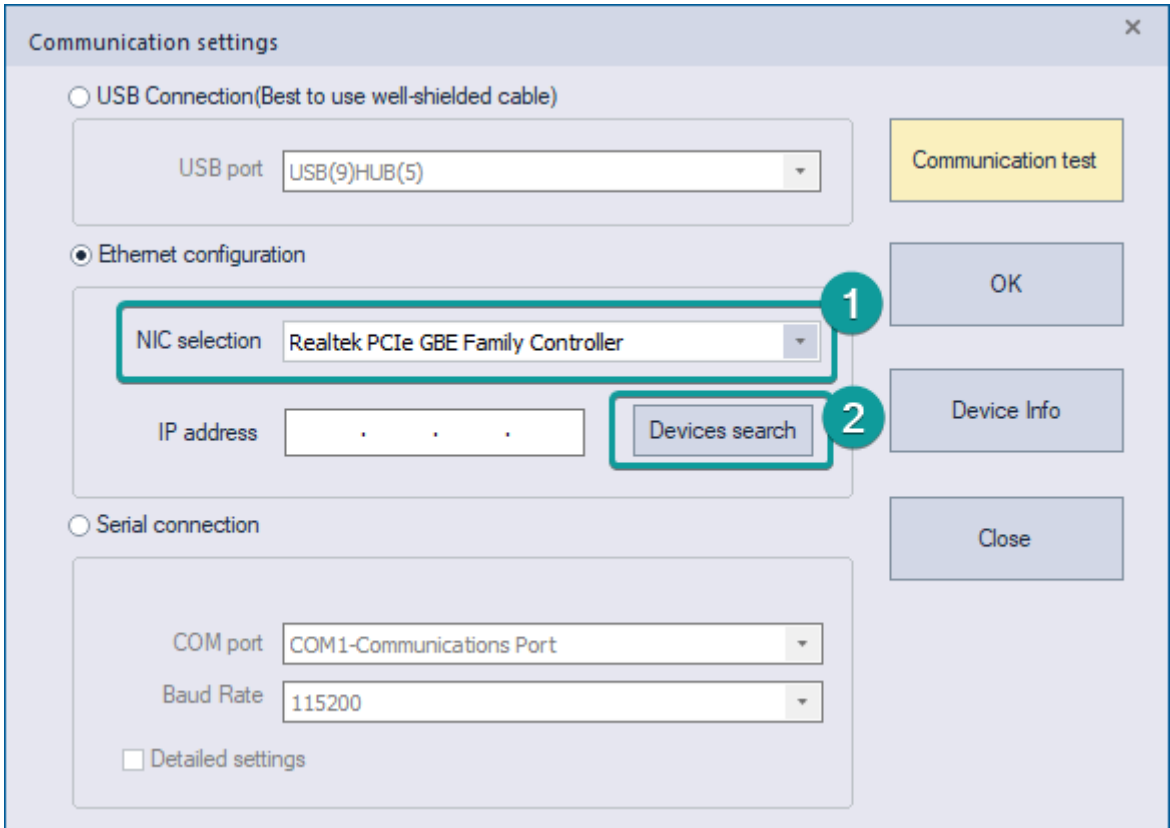
(2) Click "Communication test" to confirm the communication.



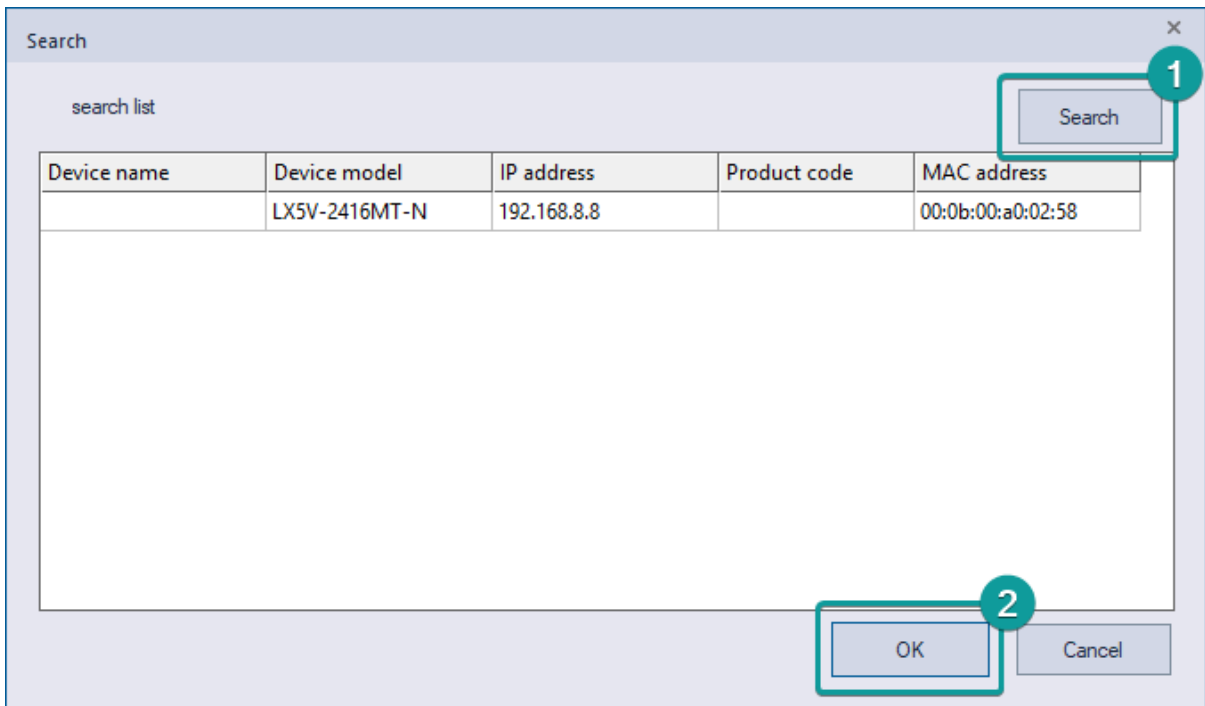
(3) After successful connection, PLC is able to operate.

PLC Editor2 Ethernet search funtion

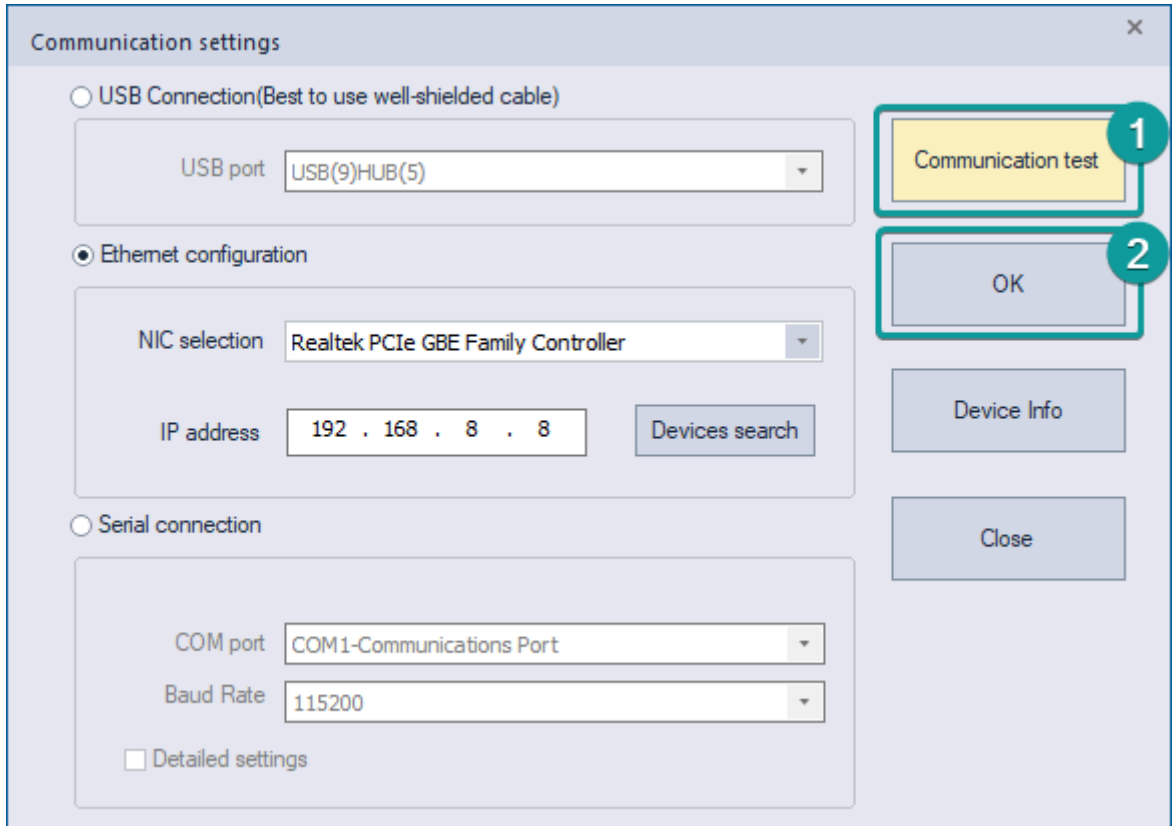
(1) Transfer settings → NIC configuration → Device search. (**#Note:** The address of NIC must be on the same network segment as that of the PLC.)



(2) The search interface is as below. Click search to display the PLC devices in the LAN, and select the corresponding device and click OK to communicate.



(3) The IP address of one is filled in automatically.



Ethernet configuration

Hardware interface

LX5V is RJ45 specification

Contents

Transmission speed

Modulation

Topology

Transmission medium

Transmission distance

Linking number

Ethernet interface

10Mbps: 10BASE-T

100Mbps: 100BASE-TX

10Mbps/100Mbps self-adaptive

Bandwidth

Starlike

Class 5 or above twisted pairs or shielded twisted pairs with aluminum foil and woven mesh

The distance between nodes: 100m or less

8

Total numbers of links supported

When LX5V-N series PLC is powered on, ModbusTCP server monitor is automatically enabled by default. 2 to 8 ModbusTCP clients are supported, and the port number is 502. (PLC host computer upload and download, monitor and HMI communication protocol are supported by the ModbusTCP server.)

The number of configurable links is 6. The free configurations of TCP server free protocol, TCP client free protocol, ModbusTCP server and ModbusTCP client are supported.

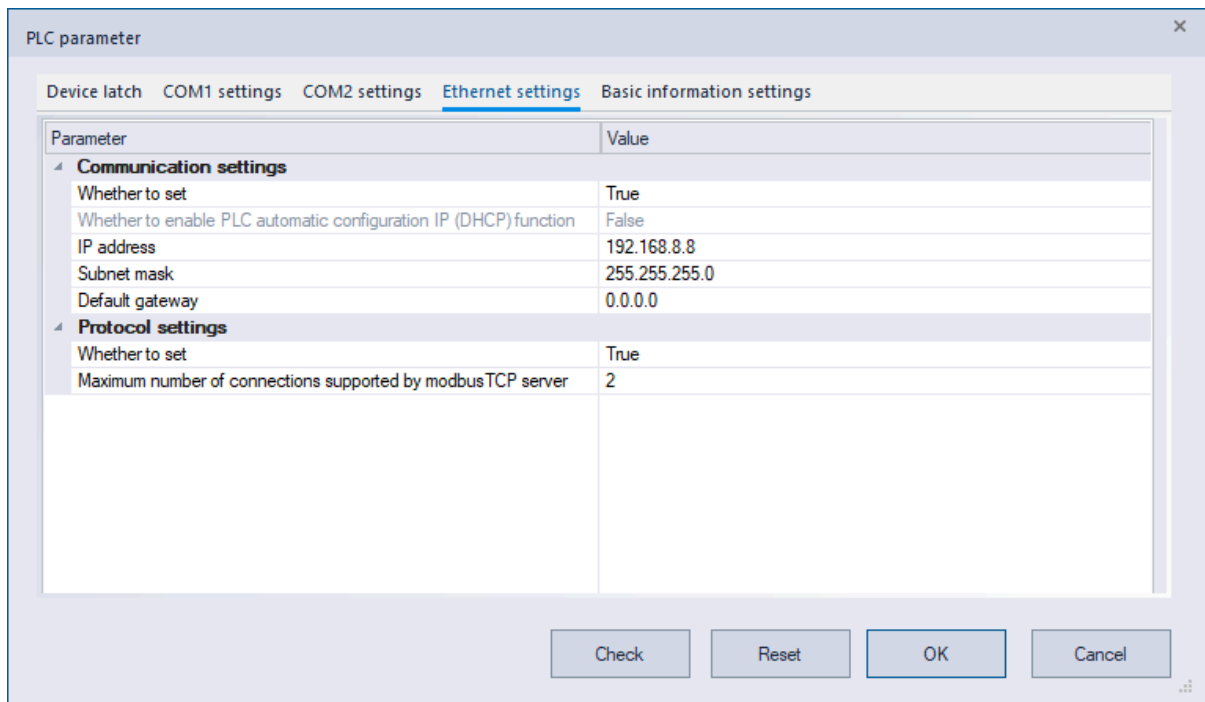
Communication protocol	Maximum links supported
ModbusTCP server	8
ModbusTCP client	6
Free TCP server	6
Free TCP client	6

IP address settings

(1) Set by programming software.

Project manager →Parameter→PLC parameters→Ethernet configuration. Download selected parameters through PLC after modification. The download takes effect after STOP->RUN is complete.

#Note: The maximum link supported of ModbusTCP servers is used to set the maximum number of external ModbusTCP servers that could connect to PLC simultaneously. The range is from 2 to 8.



(2) Set by special device.

Write IP address, subnet mask, and default gateway in SD2680 to SD2691.

SM2680 is set to ON, static IP function is enable. (**#Note:** DHCP function is not supported by LX5V currently.)

SM2683 is set to ON, IP identification could be modified.

New IP address takes effect when STOP->RUN or after power-on again.

SM number	Name	Comment
SM2680	Static set IP switch	ON:SD2680 Static set 1st set byte of IP address automatically configure IP

SM2681	Display current network information	<p>address by router DHCP, and could not be modify IP. When STOP->RUN takes effect.</p> <p>Reset SM2681 current IP by gateway subnet IP mask address after ON.</p> <p>Turn OFF after the fresh is complete.</p>
SM2682	Display current MAC information	<p>Reset SM2682 current MAC byte after of ON. IP address</p> <p>Turn OFF after the fresh is complete.</p>
SM2683	The modification flag of IP, subnet mask and gateway	<p>Reset SM2683 Modifiable byte of Unmodifiable address (After setting to ON, modify when stop->run, and automatically turn OFF after the modification)</p>

SM2684	SD2684	1st byte of subnet mask
SM2685	SD2685	2nd byte of subnet mask
SM2686	SD2686	3rd byte of subnet mask
SM2687	SD2687	4th byte of subnet mask
SM2688	SD2688	1st byte of default gateway
SM2689	SD2689	2nd byte of default gateway
SM2690	SD2690	3rd byte of default gateway
SM2691	SD2691	4th byte of default gateway

TCP protocol

TCP protocol, short for Transport Control Protocol, is a connection-oriented and reliable transport layer protocol. Connection-oriented means that a normal TCP transmission need to be completed by establishing a specific virtual circuit connection between TCP client and TCP server. To transfer data over TCP, a connection must be established between hosts at both ends.

UDP protocol

UDP protocol, short for User Datagram Protocol, is a connectionless transport layer protocol. There is no guarantee of data order, a risk of data loss. It provides a simple and unreliable information transfer service for transactions and is Mainly used in data broadcasting.

Socket

When the application layer communicates data over the transport layer, TCP encounters the problem of providing concurrent services to multiple application processes at the same time. Multiple TCP connections or multiple application processes may require data to be transmitted over the same TCP protocol port. To distinguish between different application processes and connections, many computer operating systems provide interfaces called sockets for applications to interact with the TCP/IP protocol.

To generate a socket, there are three main parameters: the IP address of the destination of the communication, the transport layer protocol used (TCP or UDP) used, and the port number used. By combining these three parameters and binding to a socket, the application layer and the transport layer can distinguish communication from different application processes or network connections through the socket interface, realizing concurrent services for data transmission.

Establish an Ethernet link by socket

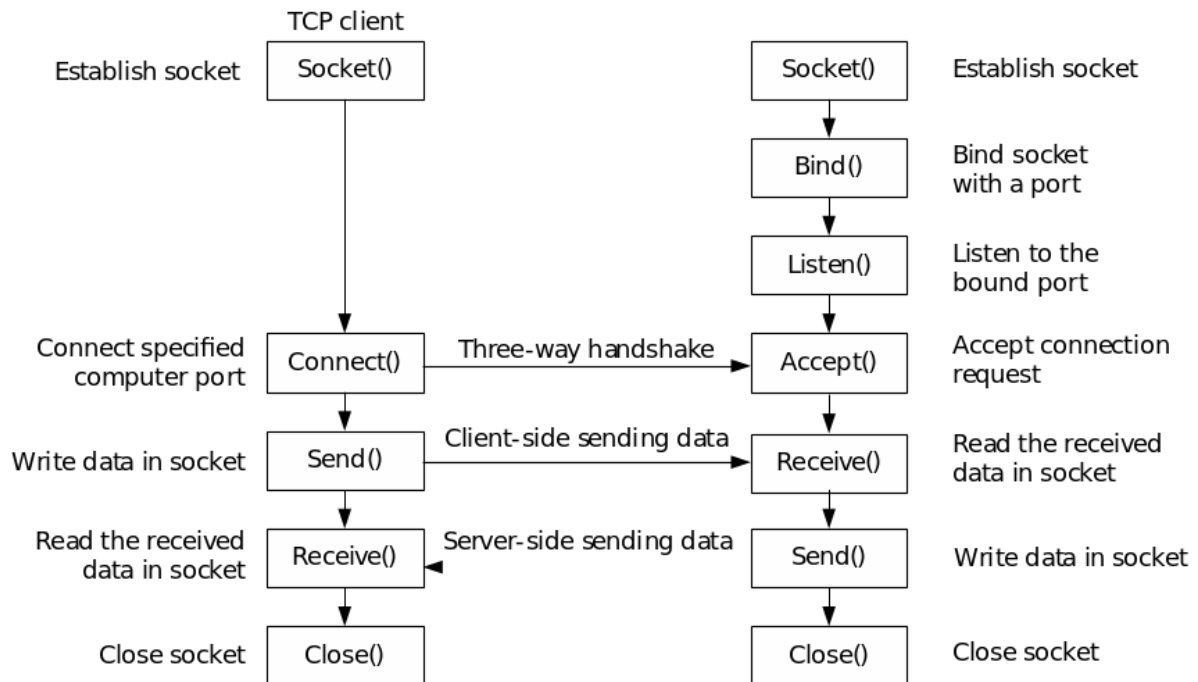
At least one pair of sockets is required to establish a socket link.

For TCP, the two sockets, one running on the TCP client and the other running on the TCP server. The connection process between sockets is divided into three steps: server monitor, client request, connection confirmation, also known as the three-way handshake.

Server monitor: After the server socket is enabled, it does not locate the specific client socket, but is in a state of waiting for the connection, monitoring the network status in real time, and waiting for the client's connection request.

Client request: Refers to a connection request made by a client-side socket, and the target of the connection is the server-side socket. To do this, the client-side socket must first describe the socket of the server to which it is connecting, indicate the address and port number of the server-side socket, and then make a connection request to the server-side socket

Connection confirmation: Refers to when the server-side socket listens to or receives a connection request from the client socket, it responds to the client socket request, establishes a new thread, sends the description of the server-side socket to the client. Once the client confirms this description, the connection is established. The server-side socket continues to listen and continues to receive connection requests from other client-side sockets.



In order to simplify the complexity of ladder programming, sockets have been partially simplified:

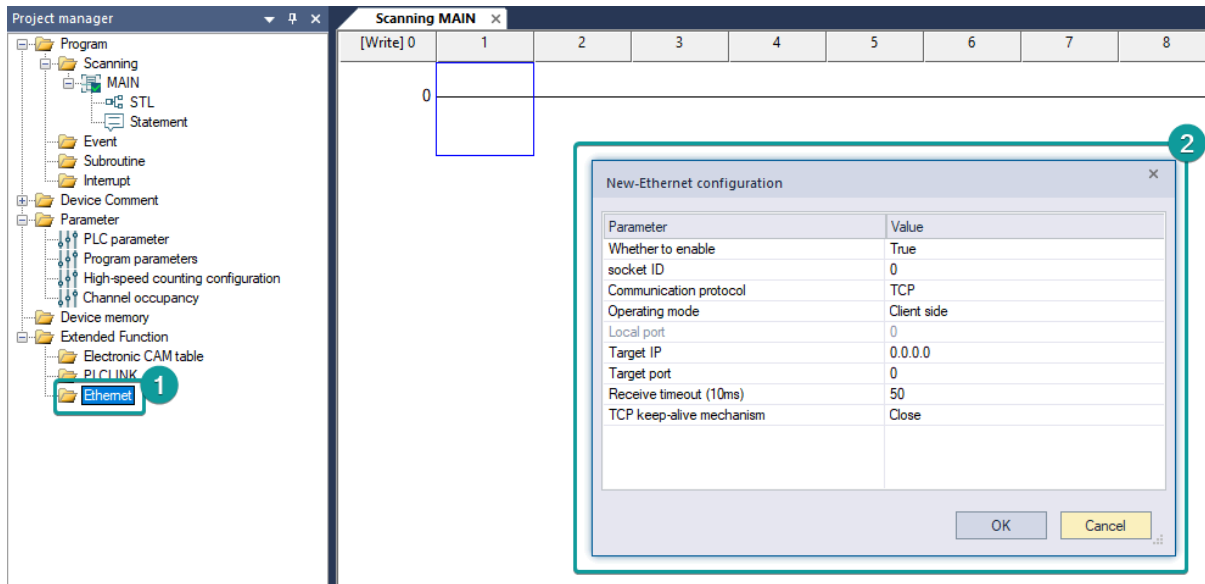
For TCP clients, merge socket() and connect() into SOCOOPEN instructions. After this function is enabled, automatically connect to the TCP server.

For TCP server, merge socket(), bind(), listen(), and accept() into SOCOOPEN instructions. After this function is enabled, automatically listen to server connection. If the server is successfully connected, the corresponding position is marked and the IP address and port information of the server are displayed.

For UDP, there is no concept of client and server. Creating a UDP socket only requires local address information and remote address information, without connection operations. Communication could be made when the address information of the local socket and the remote socket could be matched, that is, the remote address of the local socket is the same as the local address of the remote socket, and the local address of the local socket is the same as the remote address of the remote socket. For UDP connections, the connection could be established immediately by calling the SOCOOPEN instruction.

LX5V-N socket configuration instructions

LX5V socket could be configured in Project manager→Extended function→Ethernet, right click to create socket configuration, as shown below.



Socket ID: The number of the socket ranges from K0 to K5, and a total of six are supported. The socket is used to specify links, and each ID could be used for one link and could not be defined repeatedly.

Communication protocol: TCP protocol and UDP protocol are supported.

Operating mode: For TCP, client and server could be selected. For UDP, this is meaningless.

Local port:

For TCP client mode, the local port would be automatically allocated by PLC without setting.

For TCP server mode, the local port ranges from 1 to 65535. Port 502 is used for internal ModbusTCP and can not be set to port 502.

For UDP mode, the local port ranges from 1 to 65535. Port 1092 is used for scanning protocol of Wecon and can not be set to port 1092.

Destination IP: It is valid in TCP client mode or UDP mode, and specify the IP of opposite end device to be linked.

Destination port: It is valid in TCP client mode or UDP mode, and specify the port number of opposite end device to be linked.

Receive timeout period(10ms): After the PLC sends the data, If the response of the opposite end device exceeds the timeout period, it is considered that the network has an abnormality and sets the wrong flag.

TCP keep-alive mechanism: When using the TCP protocol for communication, if the communication line is idle in most cases, there is only a small amount of data to be sent and received, but it is necessary to keep the link open continuously, or disconnect in time in the case of a drop, crash or forced end of the process at the other end of the communication, the keep-alive mechanism can be used to communicate.

When the keep-alive function is turned on, after the two parties stop communicating for 5 seconds, the TCP connection that opens the keep-alive function will send a survival confirmation message to the other party. If the other party responds, it means that the other party is alive and online. The connection is normal, and the survival confirmation message is sent again after 5 seconds to continue to confirm. If the other party does not confirm the survival, it means that the other party has a problem, the end that opens the keep-alive will continue to send it a survival confirmation message after 5 seconds. When the opposite end does not respond for 9 consecutive times, it means that the opposite end communication is abnormal, and the end that opens the keep-alive will actively disconnect.

Ethernet instruction

SOCOPEN/Create a socket link

Create socket link specified by (s), and update the data information of this socket link to (d1) and the status information to (d2).

-[SOCOPEN (s) (d1) (d2)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(d1)	The start device that stores the data information of socket links	-	Signed BIN 16 bit	ANY_ELEMENTARY
(d2)	The start device that stores the status information of socket links	-	Bit	ANY_BOOL

Device used

Instruction	Parameter	Devices												Offset [D]	Pulse modification XXP
		Y	M	S	SM	D.b	T	C	D	R	SD	K	H		
SOCOPEN	Parameter 1											●	●		
	Parameter 2					●	●	●	●	●					
	Parameter 3	●	●	●	●	●									

Features

#Create the socket link specified in (s) and update the link information in (d1) and (d2).

#When the instruction is turned on, the devices specified in (d1) and (d2) will be used in other Ethernet instructions using the same socket ID. (SOCSEND, SOCRECV, SOCCLOSE, SOCMTCP)

#(d1) Specifies the following information (a total of 14 word devices):

Device	Function
(d1)	Local port number
(d1+1)	The 1st segment of the destination IP
(d1+2)	The 2nd segment of the destination IP
(d1+3)	The 3rd segment of the destination IP
(d1+4)	The 4th segment of the destination IP
(d1+5)	Destination port number
(d1+6)	Receive timeout period(10ms)
(d1+7)	Actual receiving length (byte)
(d1+8)	Current link error code
(d1+9)	Numbers of communication errors high bit
(d1+10)	Numbers of communication errors low bit
(d1+11)	Reserved
(d1+12)	Reserved
(d1+13)	Reserved

#(d2) Specifies the following information (a total of 14 bit devices):

Device	ON status	OFF status
(d2)	Connecting	The connection is not turned on
(d2+1)	Connection completed	Connecting or not connected
(d2+2)	Sending data(used by SOCSSEND instruction)	Data is not sent or data sending is complete
(d2+3)	Data sending completed(used by SOCSSEND instruction)	The instruction is not started or being sent.
(d2+4)	Receiving data(used by SOCRECV instruction)	No data or receiving is completed
(d2+5)	Data receiving completed(used by SOCRECV instruction)	The instruction is not started or received
(d2+6)	Connection is closing	The instruction is not started or is receiving
(d2+7)	Connection close completed	The instruction is not started or close is complete
(d2+8)	Communication completed(used by SOCMTCP instruction)	In communication
(d2+9)	Connection error	No error in connection
(d2+10)	Reserved	Reserved
(d2+11)	Reserved	Reserved
(d2+12)	Reserved	Reserved
(d2+13)	Reserved	Reserved

Features

Local port number:

Establish a TCP client: PLC automatically allocates the local communication port, ranging from 49152 to 65535. The port number is automatically incremented by 1 each time it is turned on.

Establish a TCP server: specified by Ethernet socket configuration of the host computer.

Establish a UDP connection: specified by Ethernet socket configuration of the host computer.

Destination IP:

Establish a TCP client: The destination address is specified by Ethernet socket configuration of the host computer.

Establish a TCP server: After the remote client connection is successful, display the IP address of the remote connection.

Establish a UDP connection: The destination address is specified by Ethernet socket configuration of the host computer.

Destination port number:

Establish a TCP client: The destination port number is specified by Ethernet socket configuration of the host computer.

Establish a TCP server: After the remote client connection is successful, display the port number of the remote connection.

Establish a UDP connection: The destination port number is specified by Ethernet socket configuration of the host computer.

Receive timeout period(10ms): specified by Ethernet socket configuration of the host computer.

Actual receiving length: This parameter is valid only when the SOCRECV instruction is used. It indicates the number of bytes received after the instruction is enabled.

Current link error code: Display the current error information. For details, Refer to Ethernet error code List.

Numbers of communication errors: total number of communication errors after successful connection (double word).

Error codes

Error code	Content
4085H	The device specified in application instruction (d1) and (d2) exceeds the corresponding device range.
5080H	The specified socket is already connected and cannot be opened again.
5082H	The socket used by parameter 1 exceeds the range of 0 to 5.
5083H	Failed to establish TCP server.
5084H	Failed to create links.
5086H	The specified (d) is not configured socket or the socket is not enabled.
5089H	502 port could not be used on the TCP server because the 502 port is enabled by default.

SOC_CLOSE/Close socket link

Close socket link specified by (s).

-[SOC_CLOSE (s)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16

Device used

Instruction	Parameter	Devices	Offset modification	Pulse extension
SOC_CLOSE	Parameter 1	K ●	H ● [D]	XXP

Features

#Close the socket link specified in (s).

#When the TCP server is closed, the reset request will be sent to the remote client. At the moment, in bit device specified by SOCOPEN, the status of connection closure will be set. The socket is not actually released until the connection closure state is set and the next connection is opened

#If the socket specified by (s) is not connected to the remote end, it cannot be closed and the instruction error occurs.

Error codes

Error code	Content
5081H	The socket specified by is not connected, and could not be closed
5082H	The data specified in (s) exceeds the range of 0 to 5

SOCSSEND/Ethernet free-form communication sending

Send the data in (s2) to the socket link specified by (s1) at the length specified by (S3).

-[SOCSSEND (s1) (s2) (s3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(s2)	The start device that send the data	-	Signed BIN 16 bit	ANY_ELEMENTARY
(s3)	Sent length	1 to 256	Bit	ANY16

Device used

Instruction	Parameter							Devices	Offset modification [D]	Pulse extension XXP
	T	C	D	R	SD	K	H			
SOCSSEND						●	●			
Parameter 1										
Parameter 2	●	●	●	●	●					
Parameter 3	●	●	●	●	●	●	●			

Features

#Send the data specified in (s2) from the socket connected to (s1), and the length is (s3).

#According to the devices specified by SOCOPEN, the information such as the sending status and the total sending length could be queried. For details, refer to the SOCOPEN instruction.

#It must be used with the SOCOPEN instruction, and data can only be sent after a full link has been established.

Error codes

Error code	Content
4084H	The data in (s3) exceeds the specified range.
5081H	The socket specified by is not connected, and could not be sent.
5082H	The data specified in (s) exceeds the range of 0 to 5.

SOCRECV/Ethernet free-form communication receiving

Receive the data from the socket link in (s1) and store in the start device of (s2) at the length of (S3).

-[SOCRECV (s1) (S2) (S3)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(s2)	The start device that receive the data	-	Signed BIN 16 bit	ANY_ELEMENTARY
(s3)	Receive length	1 to 256	Bit	ANY16

Device used

Instruction	Parameter	Devices						Offset modification [D]	Pulse extension XXP		
		T	C	D	R	SD	K			H	
SOCRECV	Parameter 1									●	●
	Parameter 2	●	●	●	●	●					
	Parameter 3	●	●	●	●	●				●	●

Features

#Receive the data from the socket link in (s1) and store in the start device of (s2) at the length of (S3).

#According to the devices specified by SOCOPEN, the information such as the sending status and the total sending length could be queried. For details, refer to the SOCOPEN instruction.

#It must be used with the SOCOPEN instruction, and data can only be sent after a full link has been established.

#When used with SOCSSEND, it could not be opened at the same time.

Error codes

Error code	Content
4084H	The data in (s3) exceeds the specified range.
5081H	The socket specified by is not connected, and could not be sent.
5082H	The data specified in (s) exceeds the range of 0 to 5.
5087H	Receiving data timeout

SOCMTCP/Ethernet ModbusTCP communication

Ethernet ModbusTCP client communication instruction

-[SOCMTCP (s1) (s2) (s3) (s4) (s5)]

Content, range and data type

Parameter	Content	Range	Data type	Data type (label)
(s1)	Socket ID	0 to 5	Signed BIN 16 bit	ANY16
(s2)	High byte is station number, low byte is function code	-	Signed BIN 16 bit	ANY_ELEMENTARY

(s3)	The Modbus address that need communication	1 to 256	Unsigned BIN 16 bit	ANY16
(s4)	Sent length or received length		Signed BIN 16 bit	ANY16
(s5)	Sent or received start device		Signed BIN 16 bit	ANY_ELEMENTARY

Device used

Instruction	Parameter	Devices						Offset modification [D]	Pulse extension XXP
		T	C	D	R	SD	K		
SOCMTCP	Parameter 1							●	●
	Parameter 2	●	●	●	●	●	●	●	●
	Parameter 3	●	●	●	●	●	●	●	●
	Parameter 4	●	●	●	●	●	●	●	●
	Parameter 5	●	●	●	●	●			

Features

#(s1) specify the socket link. The other parameters are compatible with RS instruction Modbus master protocol.

#(s2) high byte is station number. For ModbusTCP, the station number could be set at will.

#(s2) low byte is function code. For details, refer to [Modbus protocol description](#).

#(s3) Modbus communication address, ModbusTCP server address that needs to be read or written.

#(s4): the length read or written by Modbus.

#(s5): the start device that Modbus receive read data or or store written data.

#It must be used with the SOCOPEN instruction, and data can only be sent after a full link has been established.

#This instruction can only be used when a TCP client socket link is established.

#The communication completion information and the number of received and transmitted could be viewed in the soft devices specified in the SOCOPEN instruction.

Error codes

Error code	Content
5081H	The socket specified by is not connected, and could not communicate.
5082H	The data specified in (s1) exceeds the range of 0 to 5.
5086H	The socket specified by (s1) is not configured in the host computer or enabled.
5088H	The SOCMTCP instruction only supports TCP client mode.

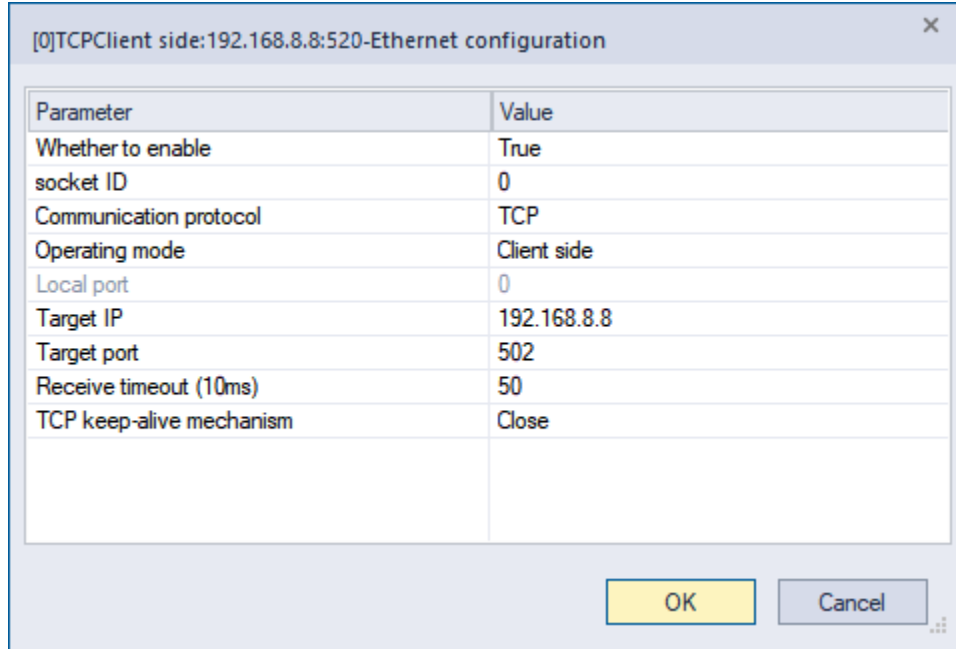
Ethernet applications

Data exchange between two PLCs through ModbusTCP

Parameters	PLC No.1	PLC No.2
------------	----------	----------

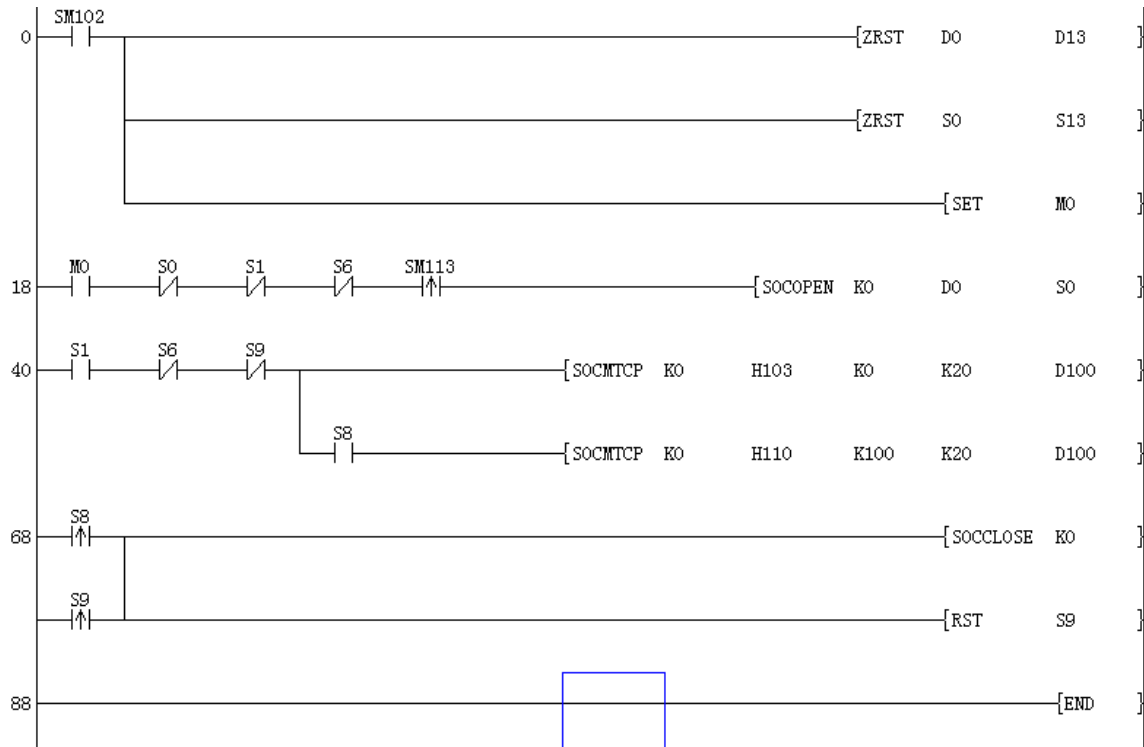
Port number	Free internal distribution	502
IP address	192.168.8.10	192.168.8.8
Protocol type	ModbusTCP client	ModbusTCP server

The socket configuration of PLC No.1



Ladder diagram logic: Automatically connect socket 0 after power on 1s. Read the 0 address length 20 of PLC No.2 to D100 to D119 after the link is successful, and set the value of D100 to D119 to address 100 of PLC No.2 after the communication is successful. Close the link when communicate successfully again, and wait 1s to re-connect after closing successfully. Repeat the actions above.

The ladder diagram of PLC No.1



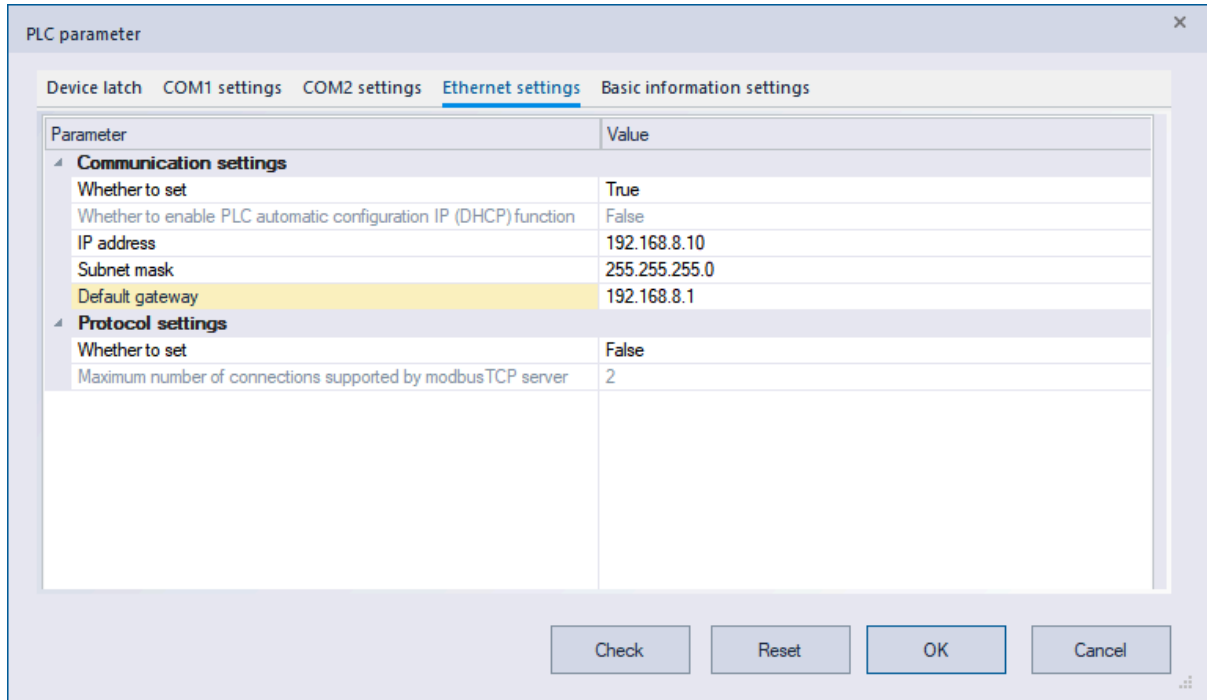
As a ModbusTCP server, PLC No. 2 does not need to write instructions. (Open two links by default, and could be modified in [PLC parameters]→[Ethernet settings]. A maximum of eight links are supported.)

Data exchange between two PLCs through Free TCP

Parameters	PLC No.1	PLC No.2
Port number	Free internal distribution	520
IP address	192.168.8.10	192.168.8.8
Protocol type	Free TCP client	Free TCP server

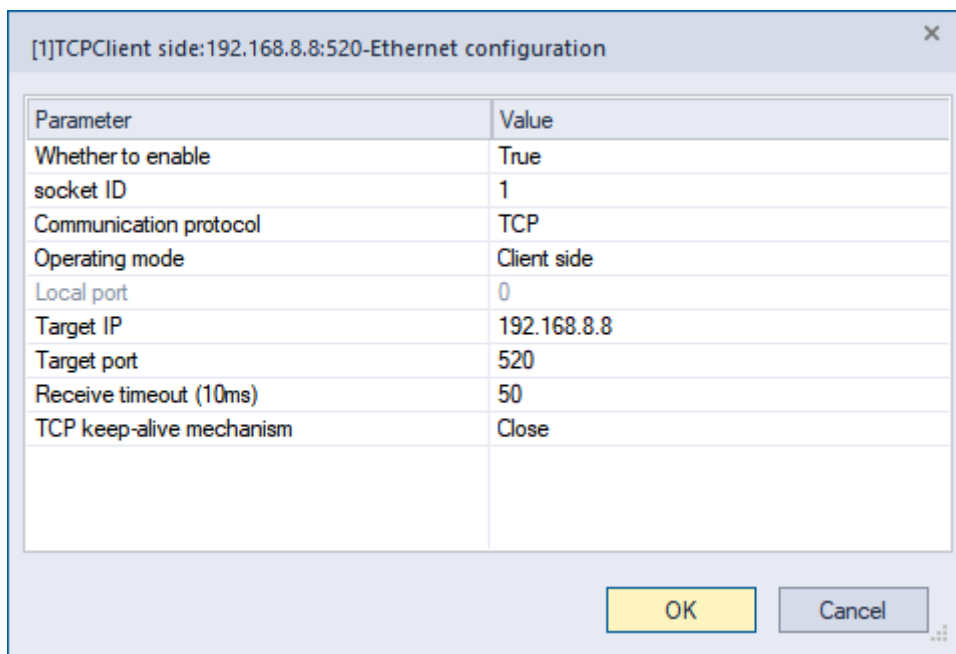
The IP setting of PLC No.1

[Project manager]→[Parameter]→[PLC parameter]→[Ethernet settings]



The socket configuration of PLC No.1

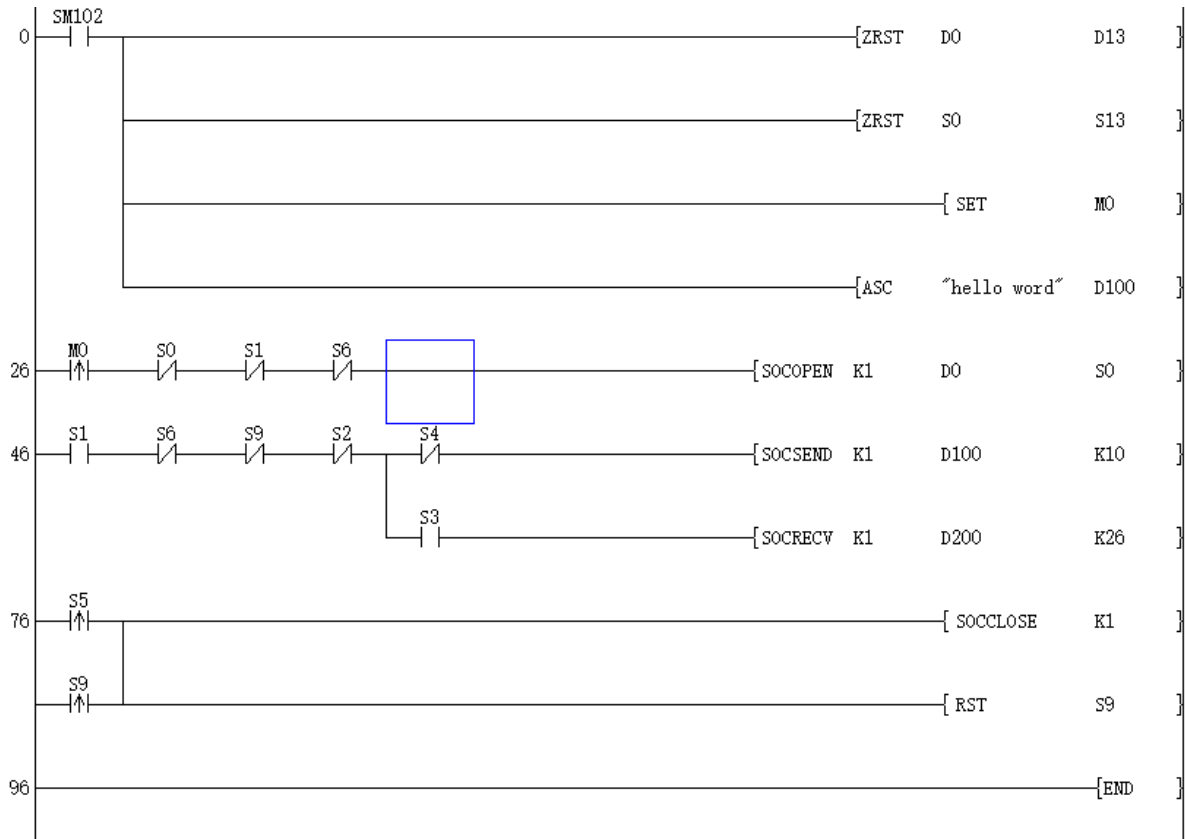
[Project manager]→[Extended function]→[Ethernet],and right click to create.



The ladder diagram of PLC No.1

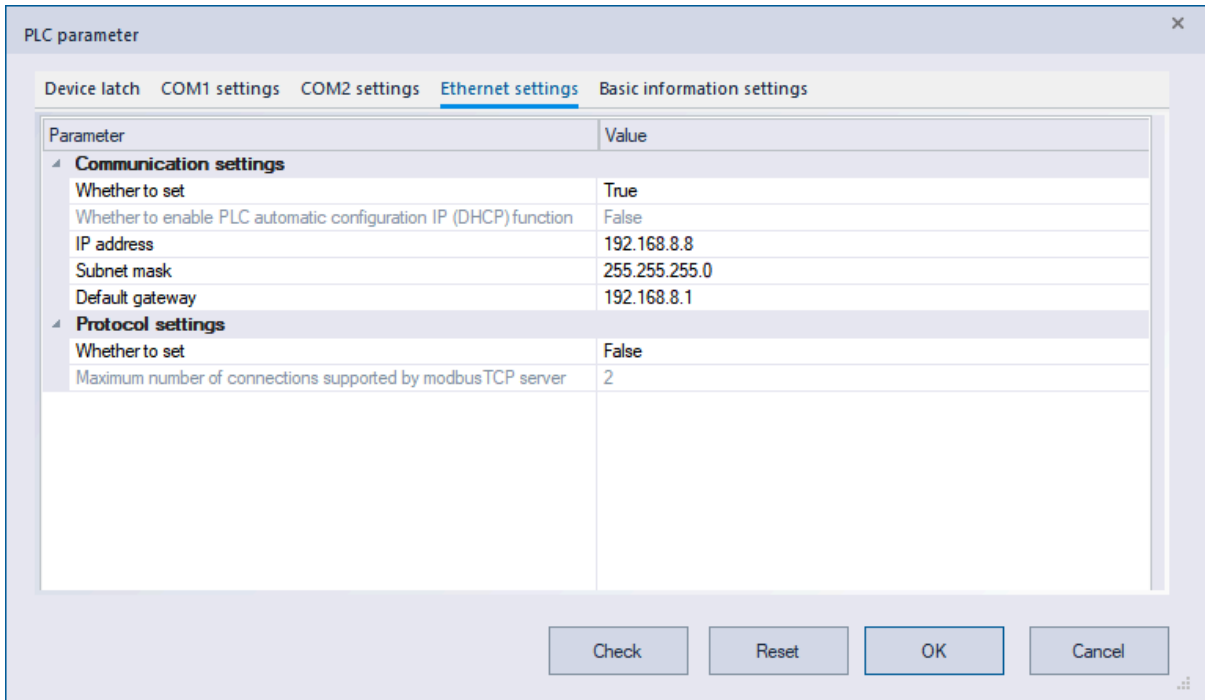
Ladder diagram logic: Automatically connect socket one after power on. Send character string "hello word" initiatively to PLC No.2 after connecting successfully.

After receiving "hello word" and verifying it correctly, PLC No.2 would reply "abcdefghijklmnopqrstuvwxy". If PLC No.1 receives the reply of PLC No.2, the link closed.



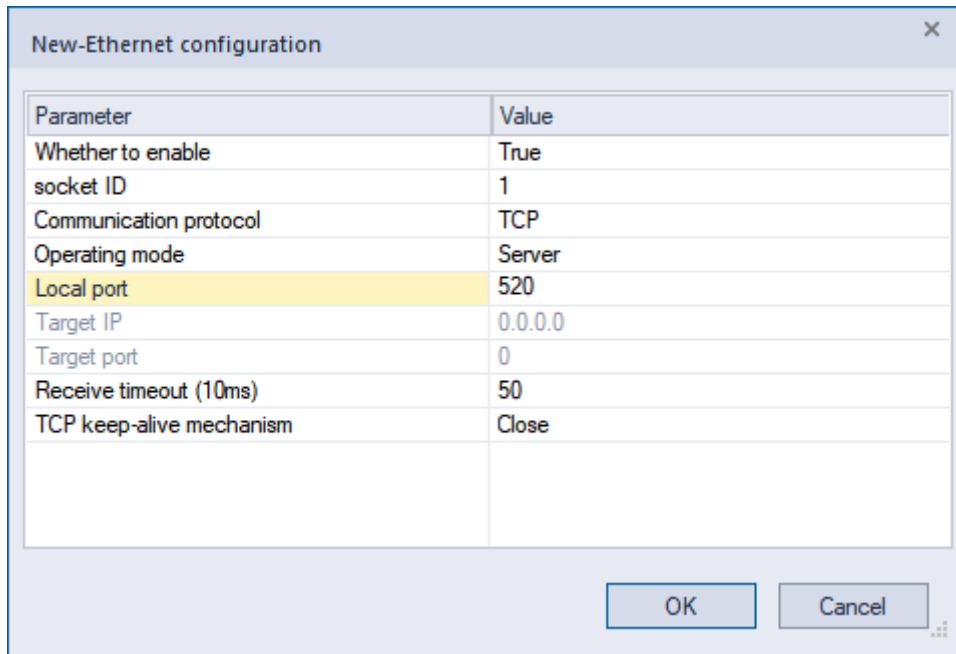
The socket configuration of PLC No.2

[Project manager]→[Parameter]→[PLC parameter]→[Ethernet settings]

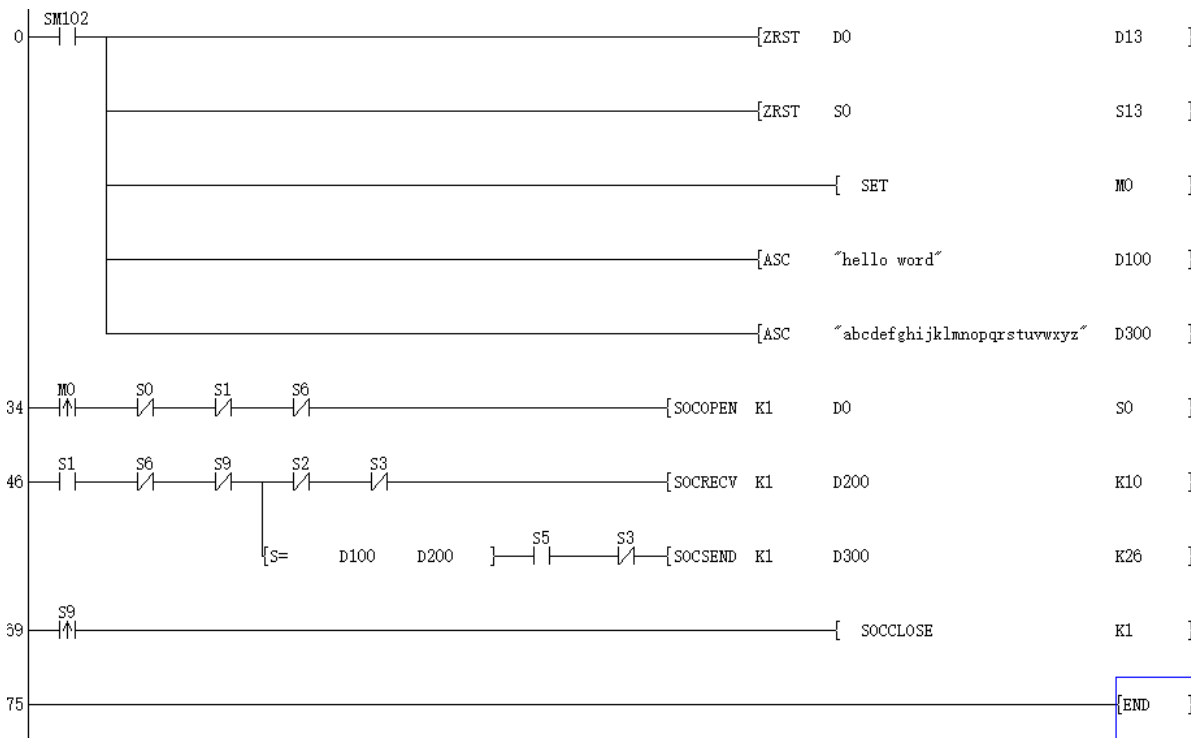


The socket configuration of PLC No.2

[Project manager]→[Extended function]→[Ethernet],and right click to create.



The ladder diagram of PLC No.2

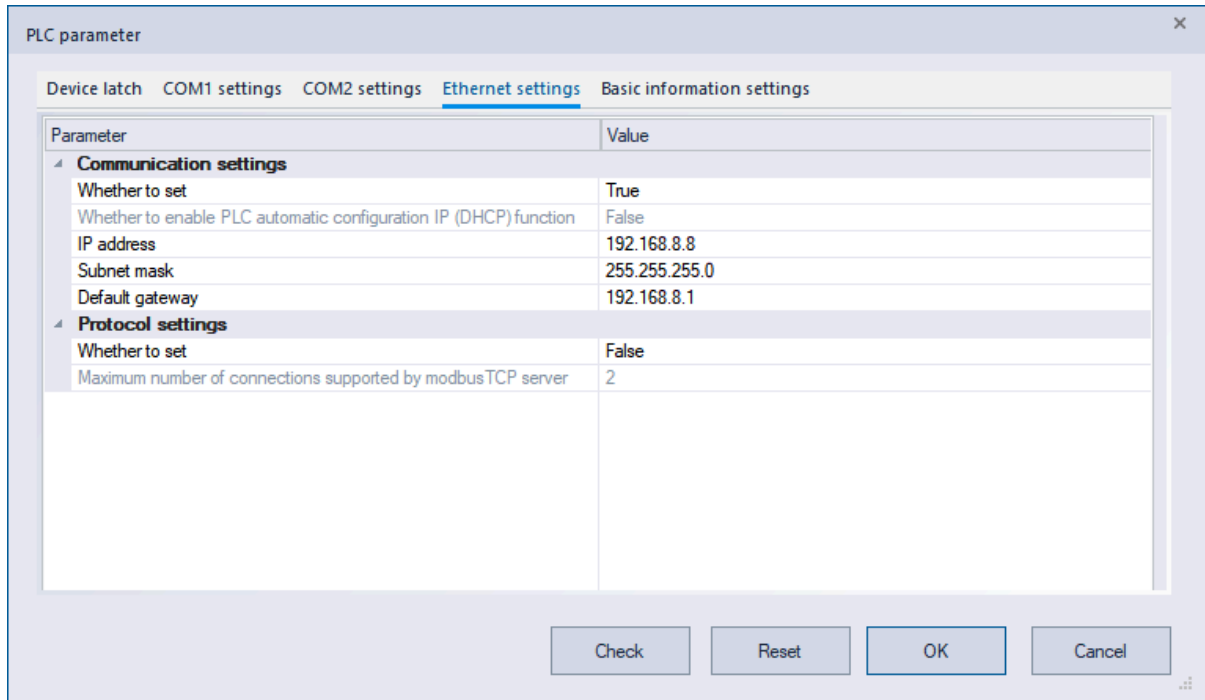


Ladder diagram logic: Automatically open the monitor server link of socket one after power on. The data sent by the client is continuously read after connecting successfully. After receiving "hello word", PLC No.2 would reply "abcdefghijklmopqrstuvwxyz".

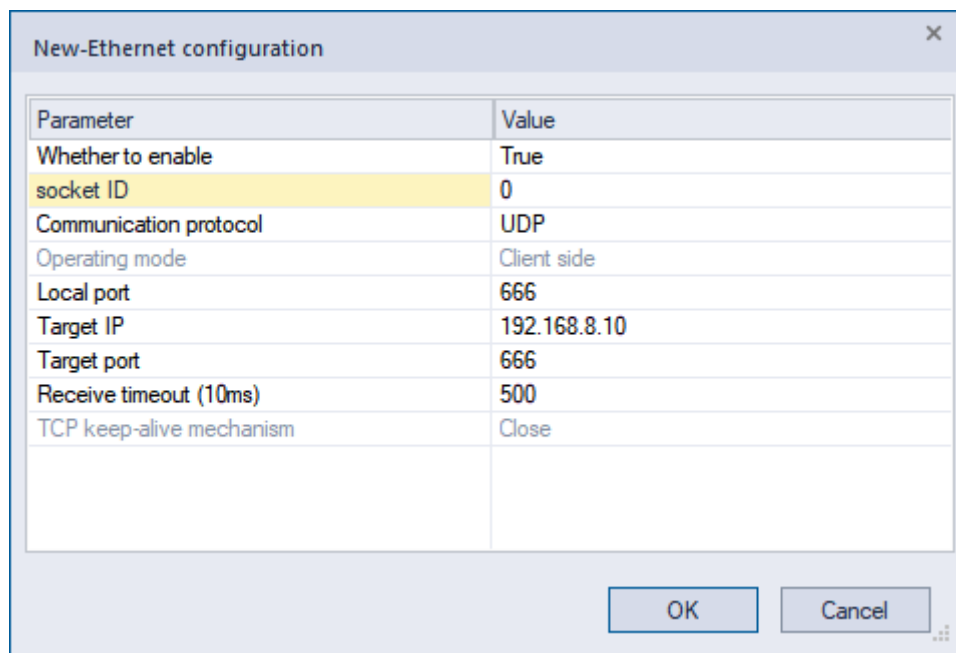
Data exchange between two PLCs through Free UDP

Parameters	PLC No.1	PLC No.2
Port number	666	666
IP address	192.168.8.10	192.168.8.8
Protocol type	Free UDP	Free UDP

The IP setting of PLC No.1

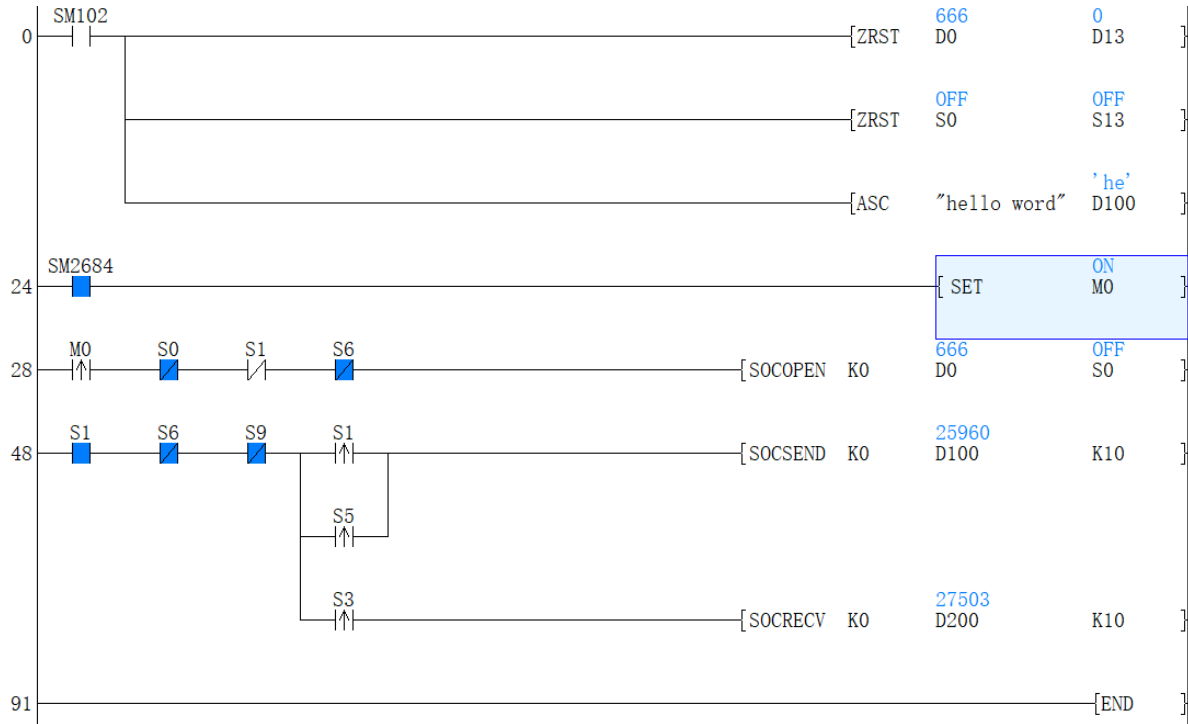


The socket configuration of PLC No.1

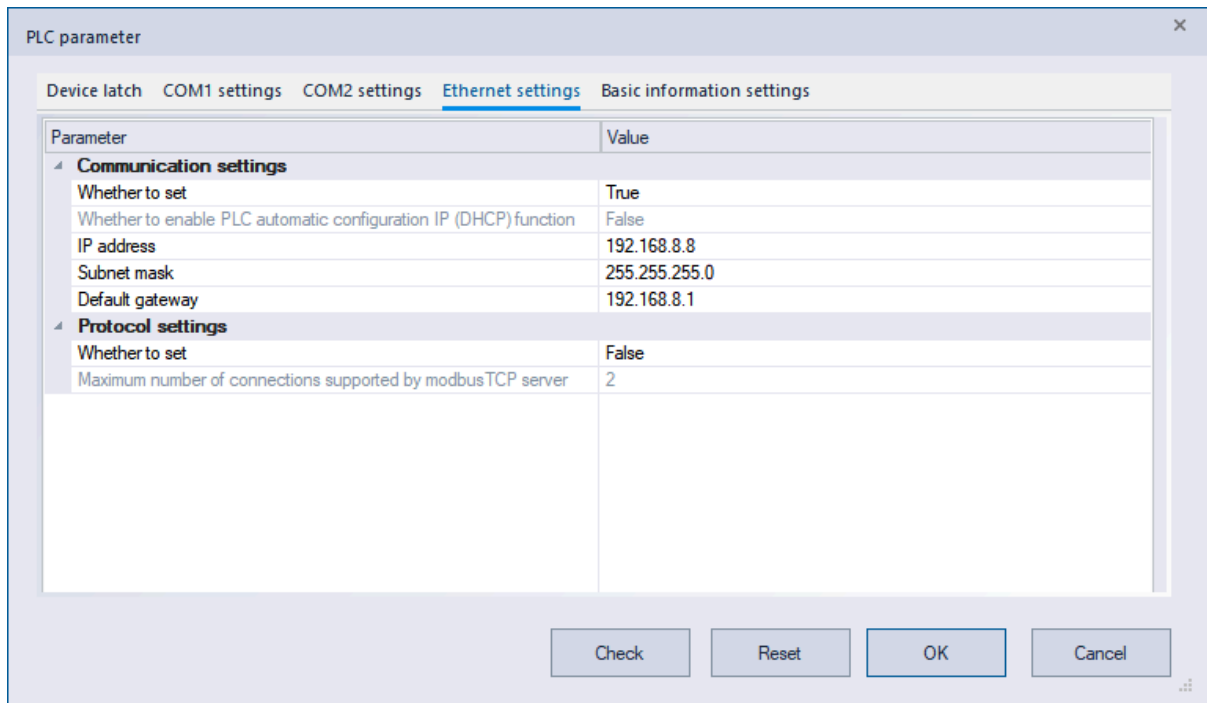


The ladder diagram of PLC No.1

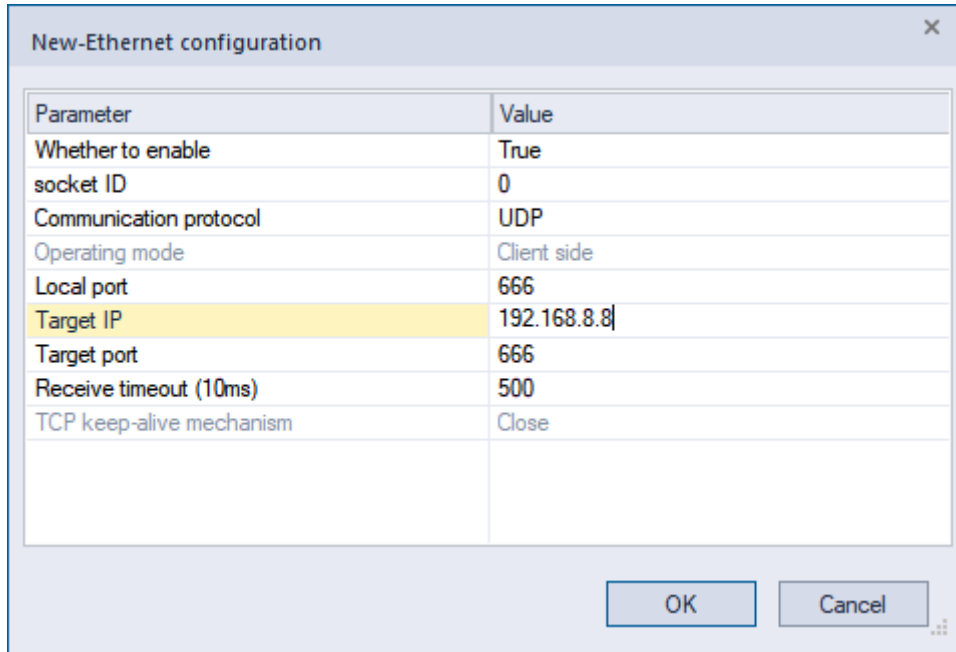
Ladder diagram logic: After setting the NIC state bit, establish UDP socket. After the link is established successful, send a data of 20 bytes that start from D100 to 192.168.8.10: 666. After the data is sent successfully, wait for the reply data of the other party. After the reply succeeds, continues the process, and so on



The IP address configuration of PLC No.2

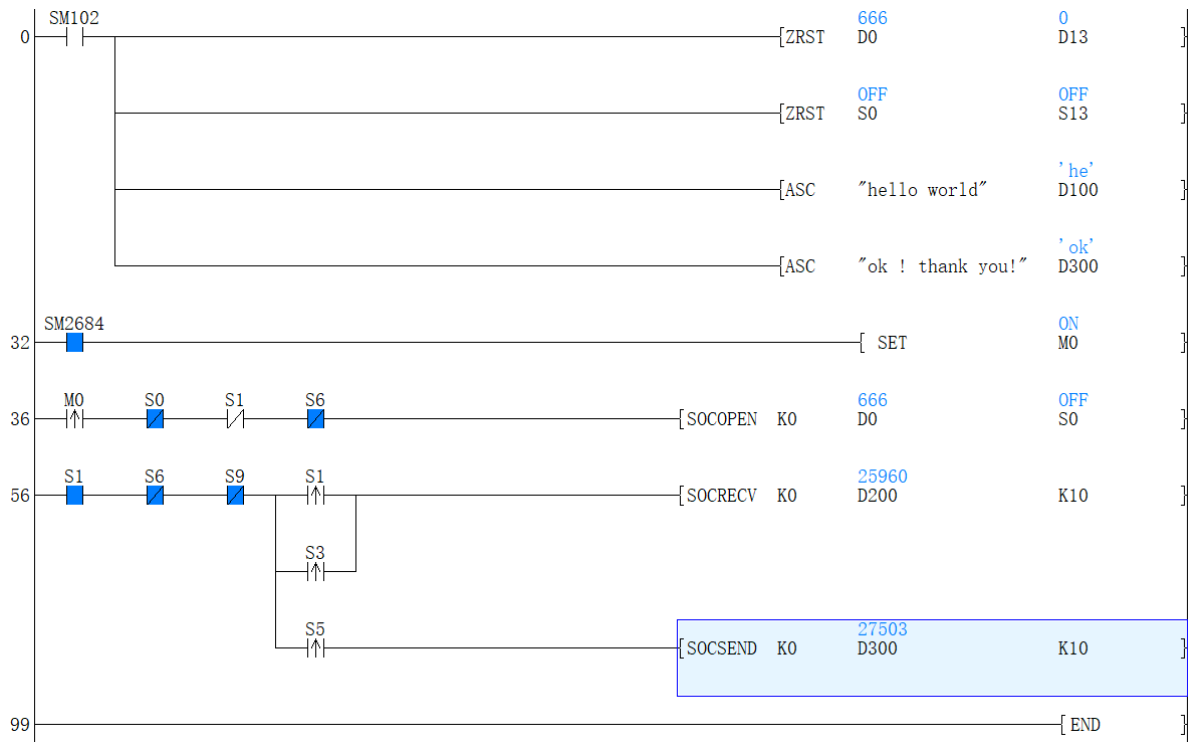


The socket configuration of PLC No.2



The ladder diagram of PLC No.2

Ladder diagram logic: After setting the NIC state bit, establish UDP socket. After the link is established successfully, send a data of 20 bytes that start from D300 to 192.168.8.10: 666. After the data is sent successfully, wait for the reply data of the other party, and so on.



List of special device related to Ethernet

SM number	Name
SM2681	Display current network information
SM2682	Display current MAC information
SM2683	The modification flag of IP, subnet mask and gateway
SM2684	The connecting status of NIC

SM2692 MAC address modification flag

SM2700 ModbusTCP keep-alive mechanism

SM2701 ModbusTCP server force close

SM2710 Ethernet error flag

SM2740 ModbusTCP server connection status 1

SM2760 ModbusTCP server connection status 2

SM2780 ModbusTCP server connection status 3

SM2800 ModbusTCP server connection status 4

SM2820 ModbusTCP server connection status 5

SM2840 ModbusTCP server connection status 6

SM2860 ModbusTCP server connection status 7

SM2880 ModbusTCP server connection status 8

SD number	Name
SD2680	The 1st byte of IP address
SD2681	The 2nd byte of IP address
SD2682	The 3rd byte of IP address
SD2683	The 4th byte of IP address
SD2684	The 1st byte of subnet mask
SD2685	The 2nd byte of subnet mask
SD2686	The 3rd byte of subnet mask
SD2687	The 4th byte of subnet mask
SD2688	The 1st byte of default gateway
SD2689	The 2nd byte of default gateway
SD2690	The 3rd byte of default gateway
SD2691	The 4th byte of default gateway

SD2692	The 1st byte of MAC
SD2693	The 2nd byte of MAC
SD2694	The 3rd byte of MAC
SD2695	The 4th byte of MAC
SD2696	The 5th byte of MAC
SD2697	The 6th byte of MAC
SD2700	Communication speed display
SD2702	Maximum link number supported by ModbusTCP server
SD2703	The number of links of ModbusTCP
SD2710	Error code
SD2711	The socket ID of current error

SD2720	Input low bit of number of ping request
SD2721	Input high bit of number of ping request
SD2722	Input low bit of number of ping response
SD2723	Input high bit of number of ping response
SD2724	Input low bit of number of ping request
SD2725	Input high bit of number of ping request
SD2726	Input low bit of number of ping response
SD2727	Input high bit of number of ping response
SD2728	The number of arp pack sent
SD2729	The number of arp pack received
SD2730	The number of IP pack sent
SD2731	The number of IP pack received
SD2732	The number of TCP pack sent

soc
err
The
pur
of
ext
inp
pin
con
The
pur
of
rep
afte
rec
ext
pin
con
The
pur
of
rep
afte
rec
ext
pin
con
ser
The
of
the
num
of
arp
pac
ser
The
num
of
arp
pac
rec
The
num
of
IP
pac
ser
The
num
of
IP
pac
rec
The
num

SD2733	The number of TCP pack received
SD2734	The number of UDP pack sent
SD2735	The number of UDP pack received
SD2740	Connection one Local port number
SD2741	Connection one The 1st byte of IP address
SD2742	Connection one The 2nd byte of IP address
SD2743	Connection one The 3rd byte of IP address
SD2744	Connection one The 4th byte of IP address
SD2745	Connection one Opposite end port number
SD2746	Reserved
SD2747	Reserved
SD2748	Connection one Error code
SD2749	Connection one Error communication times low word
SD2750	Connection one Error communication times high word
SD2760	Connection two Local port number
SD2761	Connection two The 1st byte of IP address
SD2762	Connection two The 2nd byte of IP address
SD2763	Connection two The 3rd byte of IP address
SD2764	Connection two The 4th byte of IP address
SD2765	Connection two Opposite end port number
SD2766	Reserved
SD2767	Reserved
SD2768	Connection two Error code
SD2769	Connection two Error communication times low word
SD2770	Connection two Error communication times high word
SD2780	Connection three Local port number
SD2781	Connection three The 1st byte of IP address
SD2782	Connection three The 2nd byte of IP address
SD2783	Connection three The 3rd byte of IP address
SD2784	Connection three The 4th byte of IP address
SD2785	Connection three Opposite end port number
SD2786	Reserved
SD2787	Reserved

SD2788	Connection three Error code
SD2789	Connection three Error communication times low word
SD2780	Connection three Error communication times high word
SD2800	Connection four Local port number
SD2801	Connection four The 1st byte of IP address
SD2802	Connection four The 2nd byte of IP address
SD2803	Connection four The 3rd byte of IP address
SD2804	Connection four The 4th byte of IP address
SD2805	Connection four Opposite end port number
SD2806	Reserved
SD2807	Reserved
SD2808	Connection four Error code
SD2809	Connection four Error communication times low word
SD2810	Connection four Error communication times high word
SD2820	Connection five Local port number
SD2821	Connection five The 1st byte of IP address
SD2822	Connection five The 2nd byte of IP address
SD2823	Connection five The 3rd byte of IP address
SD2824	Connection five The 4th byte of IP address
SD2825	Connection five Opposite end port number
SD2826	Reserved
SD2827	Reserved
SD2828	Connection five Error code
SD2829	Connection five Error communication times low word
SD2830	Connection five Error communication times high word
SD2840	Connection six Local port number
SD2841	Connection six The 1st byte of IP address
SD2842	Connection six The 2nd byte of IP address
SD2843	Connection six The 3rd byte of IP address
SD2844	Connection six The 4th byte of IP address
SD2845	Connection six Opposite end port number
SD2846	Reserved
SD2847	Reserved
SD2848	Connection six Error code
SD2849	Connection six Error communication times low word
SD2850	Connection six Error communication times high word
SD2860	Connection seven Local port number
SD2861	Connection seven The 1st byte of IP address
SD2862	Connection seven The 2nd byte of IP address
SD2863	Connection seven The 3rd byte of IP address
SD2864	Connection seven The 4th byte of IP address
SD2865	Connection seven Opposite end port number
SD2866	Reserved
SD2867	Reserved
SD2868	Connection seven Error code
SD2869	Connection seven Error communication times low word
SD2870	Connection seven Error communication times high word

SD2880	Connection eight Local port number
SD2881	Connection eight The 1st byte of IP address
SD2882	Connection eight The 2nd byte of IP address
SD2883	Connection eight The 3rd byte of IP address
SD2884	Connection eight The 4th byte of IP address
SD2885	Connection eight Opposite end port number
SD2866	Reserved
SD2867	Reserved
SD2888	Connection eight Error code
SD2889	Connection eight Error communication times low word
SD2890	Connection eight Error communication times high word

Ethernet error codes table

Operational error

Error code	Description
3680	Ethernet data reception error
3681	Ethernet data reception timeout
3684	ModbusTCP station number configuration error
3685	ModbusTCP send buffer overflow
3686	ModbusTCP function code error
3687	ModbusTCP address error
3688	ModbusTCP length error
3689	ModbusTCP data error
368A	ModbusTCP slave station is busy
368B	ModbusTCP slave station does not support function code
368C	ModbusTCP slave station fault
368D	ModbusTCP slave station confirmation
368E	ModbusTCP protocol currently does not support this instruction
368F	Network port sending timeout
3690	Receiving cache overflow
36A0	ModbusTCP unavailable gateway
36A1	ModbusTCP No response was received from the target device. Generally it means that the device is not on the network.

36C0	ModbusTCP transaction identifier error
36C1	ModbusTCP The server is full of available links
36C8	The Ethernet protocol stack is running out of space
36C9	The number of links exceeded the limit
36CA	The last sending is not complete
36CB	TCP abnormal write
36CC	TCP abnormal output
36CD	The IP address has been used
36CE	The server receiving link error
36CF	TCP receiving buffer overflow
36D0	TCP connection failed
36D1	Abnormal when closing the link initiative
36D2	An abnormal shutdown occurred inside the protocol stack
36D3	Initiate an RST link on the opposite end
36D4	A single-ended shutdown of the protocol stack occurs
36D5	There is an IP address conflict
36D6	There is an MAC address conflict
36D7	TCP sending buffer overflow
36D8	UDP abnormal connection
36D9	UDP sending buffer overflow
36DA	UDP insufficient memory space when sending
36DB	UDP failed to send
36DC	UDP memory release failure
36DD	UDP receiving buffer overflow
4084	The data input in the application instruction exceeds the specified range.
4085	The output result in the read application instruction exceeds the device range.
4086	The output result in the read application instruction exceeds the device range.
5080	The Ethernet socket is already linked and could not be opened again
5081	The Ethernet socket is not opened and could not be operated
5082	The socket ID that Ethernet instruction inputs exceeds the range
5083	Failed to create TCP server
5084	Failed to create links
5086	The socket ID used by Ethernet instruction is not configured in the host computer or is not enabled after configuration
5087	SOCRECV instruction reception timeout
5088	The socket specified by SOCMTCP instruction uses the configuration mode of non-TCP client
5089	When Ethernet socket configures a TCP server, specify the local port as 502
508A	The UDP port is set to 1092
5090	Abnormal network cable connection