MODBUS Organization

MODBUS Master Series

MODBUS-TCP Client (Master) Driver

Supported version TOP Design Studio V1.4.5 or higher

CONTENTS

We want to thank our customers who use the Touch Operation Panel.

1. System configuration

This driver is the "MODBUS-TCP Client (Master)" among the "MODBUS Protocol" of "MODBUS Organization".

Depending on the external device (MODBUS Slave Protocol supported), you may set the "command code", "protocol frame format" etc., of the driver separately. In this case, set the detailed settings according to the external device side based on the communication method.

The system configuration with an external device supported by this driver is as follows:

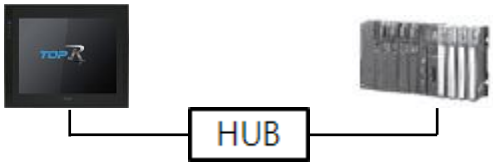
Series	CPU	Link I/F	Communication method	System setting	Cable
MODBUS Slave/Server Device			Ethernet (TCP / UDP)	3. TOP communication setting 4. External device setting	Twisted pair cable*Note 1)

*Note 1) Twisted pair cable

- Refer to STP (Shielded Twisted Pair Cable) or UTP (Unshielded Twisted Pair Cable) Category 3, 4, 5.
- Depending on the network configuration, you can connect to components such as the hub and transceiver, and in this case, use a direct cable.

■ Connectable configuration

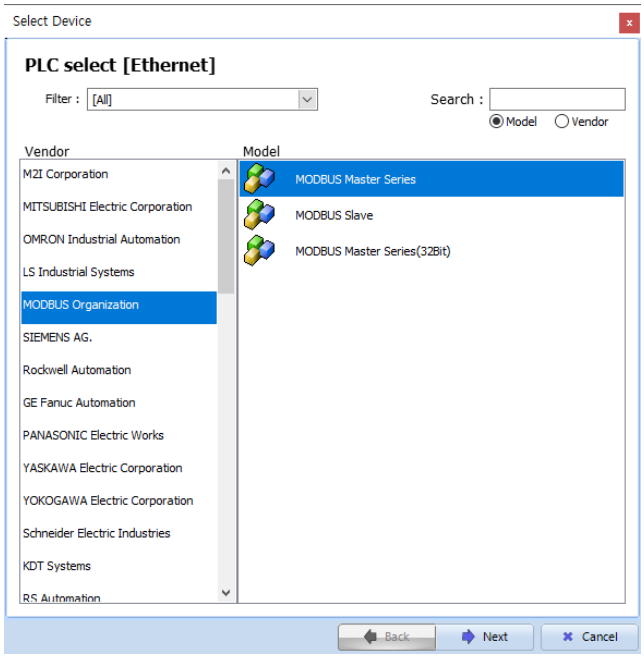
・1:1 connection

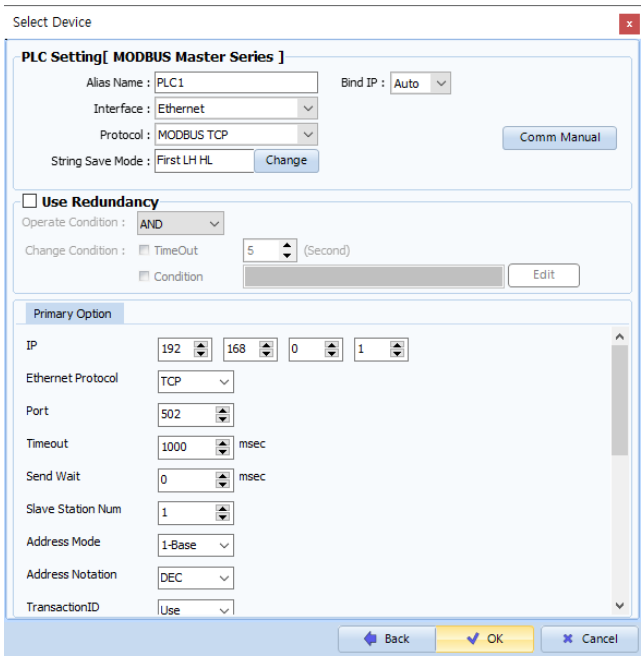


・1:N connection

2. External device selection

■ Select a TOP model and a port, and then select an external device.





Settings		Contents
TOP	Model	Select the TOP model.
External device	Vendor	Select the vendor of the external device to be connected to TOP. Please select "MODBUS Organization".
	PLC	Select the external device to be connected to the TOP.

Model	Interface	Protocol
MODBUS Master Series	Ethernet	MODBUS TCP

Please check the system configuration in Chapter 1 to see if the external device you want to connect is a model whose system can be configured.

3. TOP communication setting

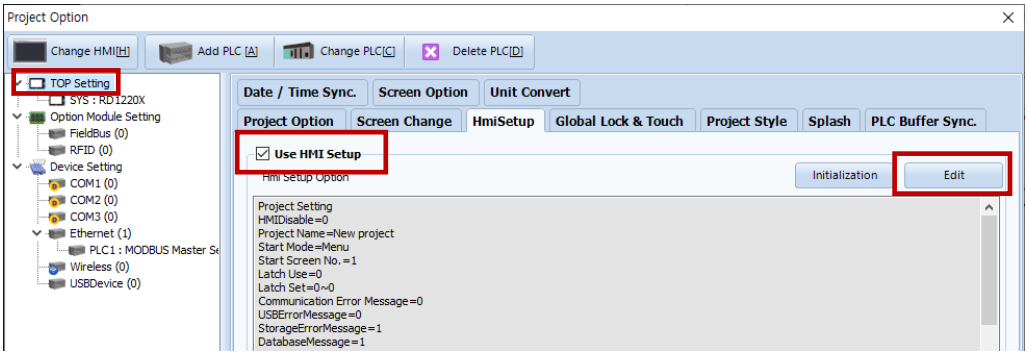
The communication can be set in TOP Design Studio or TOP main menu. The communication should be set in the same way as that of the external device.

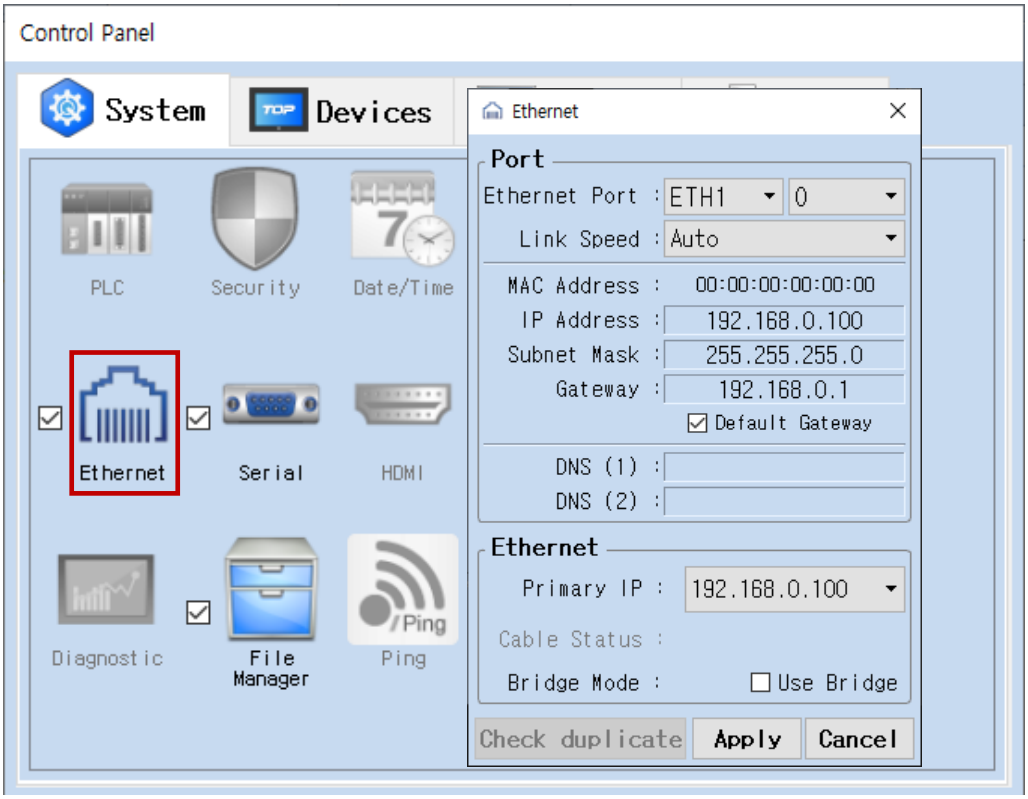
3.1 Communication setting in TOP Design Studio

(1) Communication interface setting

■ [Project] → [Property] → [TOP Setting] → [HMI Setup] → [Use HMI Setup Check] → [Edit] → [Ethernet]

– Set the TOP communication interface in TOP Design Studio.





Items	TOP	External device	Remarks
IP Address*Note 1) Note 2)	192.168.0.100	192.168.0.51	
Subnet Mask	255.255.255.0	255.255.255.0	
Gateway	192.168.0.1	192.168.0.1	

*Note 1) The network addresses of the TOP and the external device (the first three digits of the IP, 192 . 168 . 0 . 0) should match.

*Note 2) Do not use duplicate IP addresses over the same network.

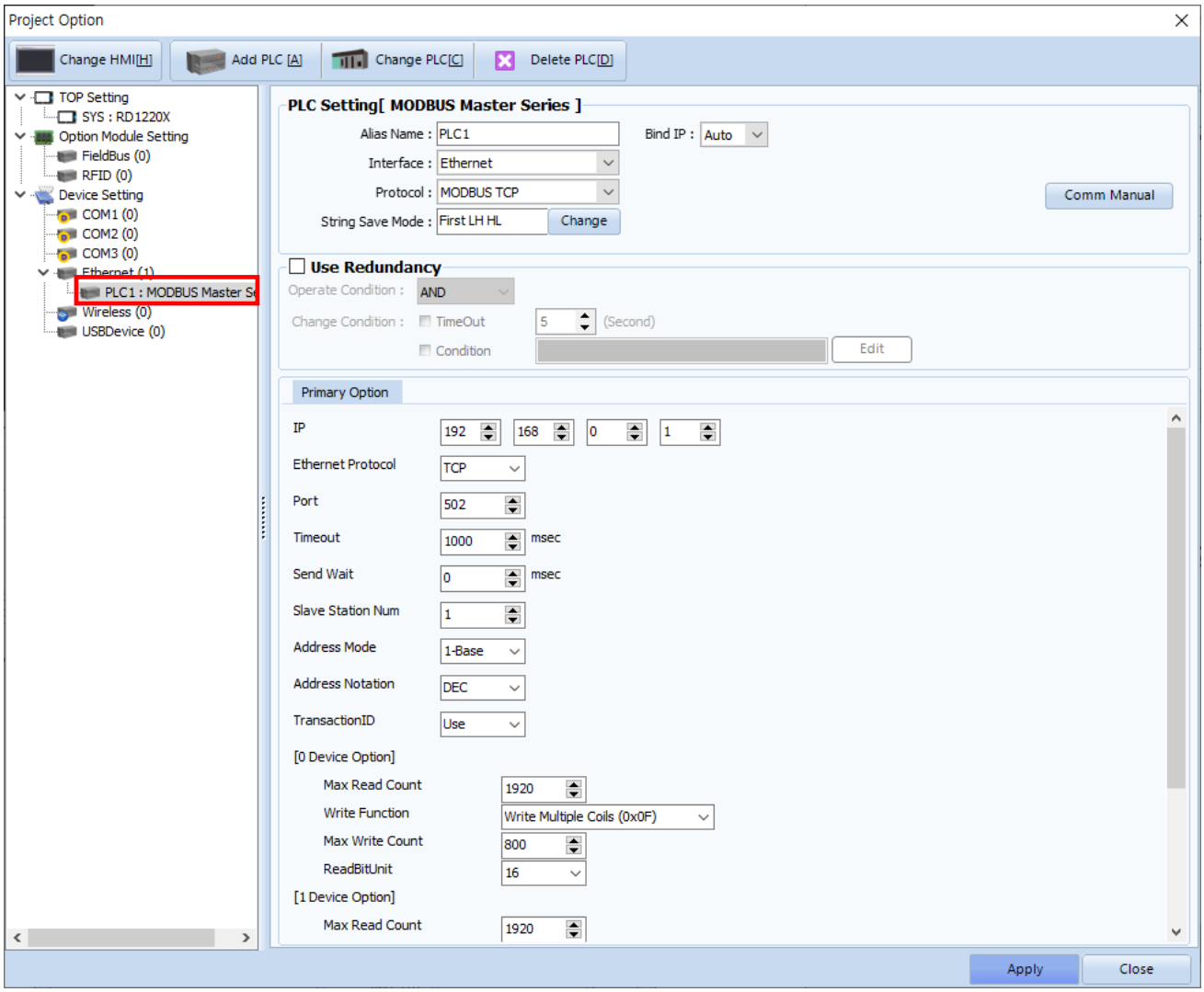
* The above settings are examples recommended by the company.

Items	Description
IP Address	Set an IP address to be used by the TOP to use over the network.
Subnet Mask	Enter the subnet mask of the network.
Gateway	Enter the gateway of the network.

(2) Communication option setting

■ [Project > Project properties > PLC settings > ETHERNET > PLC1 : MODBUS Master Series]

– Set the options of the communication driver of MODBUS-TCP Client (Master) in TOP Design Studio.



Items		Settings	Remarks
Interface		Select "Ethernet".	Refer to "2. External device selection".
Protocol		Select "MODBUS TCP".	
String save mode		Select the method to save the string.	
Redundancy	Use Redundancy	Check whether redundancy settings are used or not.	
	Operation Condition	Set the operation condition for the change condition. AND: change Primary ↔ Secondary if all change conditions checked are satisfied. OR: change Primary ↔ Secondary if any of change conditions checked are satisfied.	
	Change Condition	Set Primary ↔ Secondary change condition.	
IP		Enter the IP address of the external device.	
Ethernet Protocol		Select the Ethernet protocol between the TOP and an external device.	
Port		Enter the Ethernet communication port number of an external device.	
TimeOut (ms)		Set the time for the TOP to wait for a response from an external device.	
SendWait (ms)		Set the waiting time between TOP's receiving a response from an external device and sending the next command request.	
Slave Station Num		Enter the prefix of an external device.	
Address Mode		Select the address input method. When set to "1-base", the request is made by subtracting 1 from the address entered in the drawing during communication. Ex) When 400001 is registered, request 0000(hex) of Holding Register.	
Address Notation		Select the address notation.	
Transaction ID		Set whether to apply Transaction ID of Modbus TCP packets or not. Fixed to 0 when not used	
[0 Device Option]		**Coil**	
Max Read Count		Set the maximum count at which a request can be made at one time when coil read is requested.	*Note *Note 2)
Write Function		Set the coil write request command. 0x05 : Force Single Coil (Write in 1-bit unit. Only bit unit operation can be used.) 0x0F : Force Multiple Coils (Write in 16-bit unit) Auto : Request as 0x05 or 0x0F depending on the number of data.	*Note 3)
Max Write Count		Sets the maximum count at which a request can be made at one time when Coil Write is requested.	*Note 2)
Read Bit Unit		Sets the number of bits requested when Coil Write is requested. If the set value is 16 and the address following the screen is registered, data is requested up to "Max Read Count" at one time.	
[1 Device Option]		**Discrete Input**	
Max Read Count		Set the maximum count at which a request can be made at one time when Discrete Input is requested.	*Note *Note 2)
Read Bit Unit		Sets the number of bits requested when Discrete Input is requested. If the set value is 16 and the address following the screen is registered, data is requested up to "Max Read Count" at one time.	
[3 Device Option]		**Input Register**	
Max Read Count		Set the maximum count at which a request can be made at one time when Input Register Read is requested.	*Note *Note 2)
[4 Device Option]		**Holding Register**	
Max Read Count		Set the maximum count at which a request can be made at one time when Holding Register Read is requested.	*Note 1)
Write Function		Set the Holding Register write request command. 0x06 : Preset Single Register (write 1) 0x10 : Preset Multiple Registers (write n) Auto : Request as 0x06 or 0x10 depending on the number of data.	*Note 3)
Max Write Count		Set the maximum count at which a request can be made at one time when requesting Holding Register data write with command 0x10.	*Note 2)

*Note 1)

- The Max Read Count of each device is also used as the address range to requested at one time without communicating several times when the addresses registered on the screen are not consecutive.

Ex 1) If 400001, 400002, 400003, 400004, 400005, and 400120 are registerd on the screen as a numeric object and the max read count of 4 devices is set to 120, assuming that the addresses are continuous from 400001 to 400120, it reads data with one request by 120 words from 400001.

Ex 2) If 400001, 400002, 400003, 400004, 400005, and 400120 are registerd on the screen as a numeric object and the max read count of 4 devices is set to 3, it reads data with three requests by 3 words from 400003, 2 words from 400004 to 400005, and 1 word from 400120.

Ex 3) If 400001, 400010, 400011, 400021, 400031, and 400041 are registerd on the screen as a numeric object and the max read count of 4 devices is set to 10, it reads data with five request by 10 words from 400001 to 400010, 1 word from 400011, 1 word from 400021, 1 word from 400031, and 1 word from 400041.

- If Max Read Count is set to 0, up to 120 words only for consecutive addresses are requested.

*Note 2)

- Refer to the manual for the external device to check how many data can be read/written from the registered address at a time.
If a setting is made larger than the range supported by an external device, communication is not made normally.
Ex) If the holding register (4 devics) of the external device can respond only up to 10 words per communication, set the max read count of 4 devices among TOP's communication options to 10 according to the specifications of the external device.

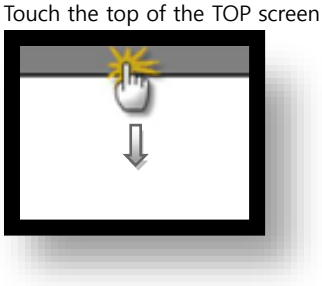
*Note 3)

- Refer to the manual of the external device and set it according to the supported write command.
If you set a write command which is not supported, data write operation is not made.

3.2. Communication setting in TOP

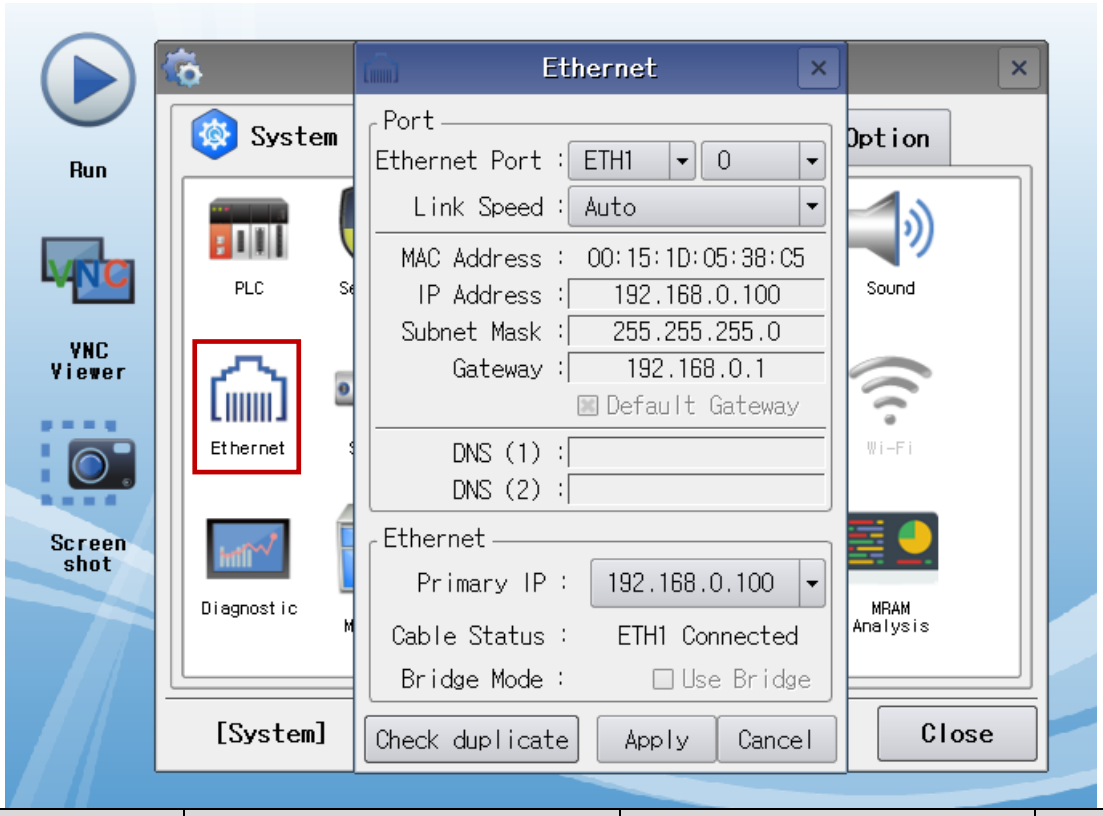
*** This is a setting method when "Use HMI Setup" in the setting items in "3.1 TOP Design Studio" is not checked.**

■ Touch the top of the TOP screen and drag it down. Touch "EXIT" in the pop-up window to go to the main screen.



(1) Communication interface setting

■ [Control Panel] → [Ethernet]



Items	TOP	External device	Remarks
IP Address	192.168.0.100	192.168.0.51	*Note 1) Note 2)
Subnet Mask	255.255.255.0	255.255.255.0	
Gateway	192.168.0.1	192.168.0.1	

*Note 1) The network addresses of the TOP and the external device (the first three digits of the IP, 192 . 168 . 0 . 0) should match.

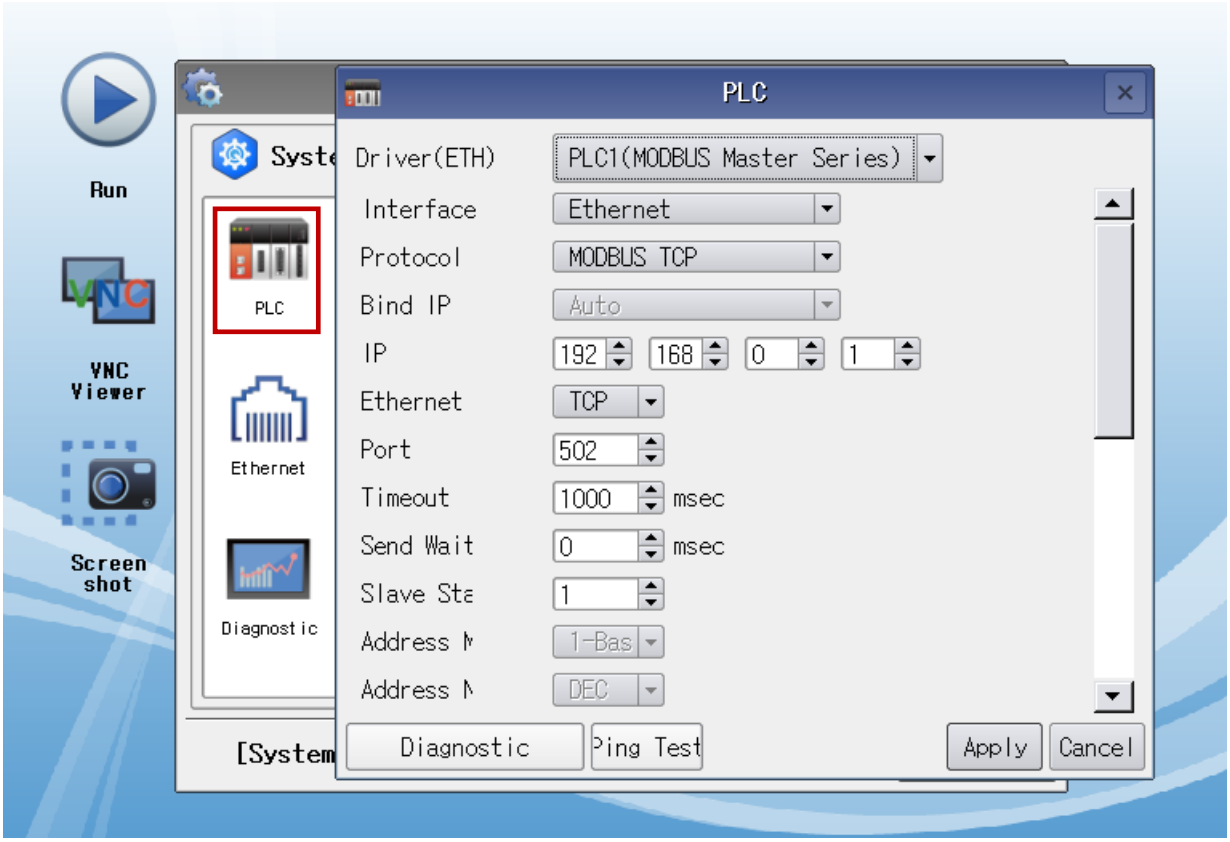
*Note 2) Do not use duplicate IP addresses over the same network.

* The above settings are examples recommended by the company.

Items	Description
IP Address	Set an IP address to be used by the TOP to use over the network.
Subnet Mask	Enter the subnet mask of the network.
Gateway	Enter the gateway of the network.

(2) Communication option setting

■ [Control panel] → [PLC]



Items	Settings	Remarks
Interface	Select "Ethernet".	Refer to "2. External device selection".
Protocol	Select "MODBUS TCP".	
IP	Enter the IP address of the external device.	
Ethernet Protocol	Select the Ethernet protocol between the TOP and an external device.	
Port	Enter the Ethernet communication port number of an external device.	
TimeOut (ms)	Set the time for the TOP to wait for a response from an external device.	
SendWait (ms)	Set the waiting time between TOP's receiving a response from an external device and sending the next command request.	
Slave Station Num	Enter the prefix of an external device.	
Address Mode	Select the address input method. 1-base: The memory address of an device at 1. Request data to registered address -1. 0-base: The memory address of an device at 0. Request data to registered address.	
Address Notation	Select the address notation.	
Transaction ID	Set whether to apply Transaction ID of Modbus TCP packets or not. Fixed to 0 when not used	
[0 Device Option]	**Coil**	
Max Read Count	Set the maximum count at which a request can be made at one time when coil read is requested.	*Note *Note 2)
Write Function	Set the coil write request command. 0x05 : Force Single Coil (Write in 1-bit unit. Only bit unit operation can be used.) 0x0F : Force Multiple Coils (Write in 16-bit unit) Auto : Request as 0x05 or 0x0F depending on the number of data.	*Note 3)
Max Write Count	Sets the maximum count at which a request can be made at one time when Coil Write is requested.	*Note 2)
Read Bit Unit	Sets the number of bits requested when Coil Write is requested. If the set value is 16 and the address following the screen is registered, data is requested up to "Max Read Count" at one time.	
[1 Device Option]	**Discrete Input**	

Max Read Count	Set the maximum count at which a request can be made at one time when Discrete Input is requested.	*Note *Note 2)
Read Bit Unit	Sets the number of bits requested when Discrete Input is requested. If the set value is 16 and the address following the screen is registered, data is requested up to "Max Read Count" at one time.	
[3 Device Option]	**Input Register**	
Max Read Count	Set the maximum count at which a request can be made at one time when Input Register Read is requested.	*Note *Note 2)
[4 Device Option]	**Holding Register**	
Max Read Count	Set the maximum count at which a request can be made at one time when Holding Register Read is requested.	*Note 1)
Write Function	Set the Holding Register write request command. 0x06 : Preset Single Register (write 1) 0x10 : Preset Multiple Registers (write n) Auto : Request as 0x06 or 0x10 depending on the number of data.	*Note 3)
Max Write Count	Set the maximum count at which a request can be made at one time when requesting Holding Register data write with command 0x10.	*Note 2)

*Note 1)

- The Max Read Count of each device is also used as the address range to requested at one time without communicating several times when the addresses registered on the screen are not consecutive.

Ex 1) If 400001, 400002, 400003, 400004, 400005, and 400120 are registerd on the screen as a numeric object and the max read count of 4 devices is set to 120, assuming that the addresses are continuous from 400001 to 400120, it reads data with one request by 120 words from 400001.

Ex 2) If 400001, 400002, 400003, 400004, 400005, and 400120 are registerd on the screen as a numeric object and the max read count of 4 devices is set to 3, it reads data with three requests by 3 words from 400003, 2 words from 400004 to 400005, and 1 word from 400120.

Ex 3) If 400001, 400010, 400011, 400021, 400031, and 400041 are registerd on the screen as a numeric object and the max read count of 4 devices is set to 10, it reads data with five request by 10 words from 400001 to 400010, 1 word from 400011, 1 word from 400021, 1 word from 400031, and 1 word from 400041.

- If Max Read Count is set to 0, up to 120 words only for consecutive addresses are requested.

*Note 2)

- Refer to the manual for the external device to check how many data can be read/written from the registered address at a time.
If a setting is made larger than the range supported by an external device, communication is not made normally.
Ex) If the holding register (4 devics) of the external device can respond only up to 10 words per communication, set the max read count of 4 devices among TOP's communication options to 10 according to the specifications of the external device.

*Note 3)

- Refer to the manual of the external device and set it according to the supported write command.
If you set a write command which is not supported, data write operation is not made.

3.3 Communication diagnostics

■ Check the interface setting status between the TOP and an external device.

- Touch the top of the TOP screen to drag it down. Touch "EXIT" in the pop-up window to move to the system screen.
- Check whether the connected port setting is correct in [Control panel] → [Ethernet].

■ Diagnosis of whether the port communication is normal or not

- Touch "Communication diagnostics" in [Control Panel] → [PLC].
- Check whether communication is connected or not.

Communication diagnostics succeeded	**Communication setting normal**
Error message	**Communication setting abnormal** - Check the cable, TOP, and external device settings. (Refer to Communication diagnostics sheet.)

■ Communication diagnostics sheet

- If there is a problem with the communication connection with an external terminal, please check the settings in the sheet below.

Items	Contents		Check		Remarks
System configuration	How to connect the system		OK	NG	1. System configuration
	Cable		OK	NG	
TOP	Version information		OK	NG	2. External device selection 3. TOP communication setting
	Communication port		OK	NG	
	Communication driver and protocol		OK	NG	
	Other detailed settings		OK	NG	
	Relative prefix	Project setting	OK	NG	
		Communication diagnostics	OK	NG	
	Ethernet port setting	IP Address	OK	NG	
		Subnet Mask	OK	NG	
		Gateway	OK	NG	
External device	CPU name		OK	NG	4. External device setting
	Communication port		OK	NG	
	Protocol		OK	NG	
	Prefix		OK	NG	
	Other detailed settings		OK	NG	
	Ethernet port setting	IP Address	OK	NG	
		Subnet Mask	OK	NG	
		Gateway	OK	NG	
	Check address range		OK	NG	5. Supported addresses (For details, please refer to the PLC vendor's manual.)

4. External device setting

Refer to the vendor's user manual to set as a Modbus Slave (server).

- Check the memory address of the external device.

5. Supported addresses

The devices available in TOP are as follows:

The device range (address) may differ depending on the CPU module series/type. The TOP series supports the maximum address range used by the external device series. Please refer to each CPU module user manual and be take caution to not deviate from the address range supported by the device you want to use.

	Bit	Word	Remarks
Coil	000001 – 065536	000001 – 065521	
Discrete Input	100001 – 165536	100001 – 165521	*Note 1)
Input Register	300001.00 – 365536.15	300001 – 365536	*Note 1)
Holding Register	400001.00 – 465536.15	400001 – 465536	

***Note 1)** Write-only

Appendix A. MODBUS TCP/IP ADU Frame(Data Frame)

Describes MODBUS protocol commands and devices supported by "MODBUS TCP Client (Master) Driver" of this device.

WHAT IS MODBUS?

The MODBUS protocol was developed in 1979 by Modicon, Incorporated, for industrial automation systems and Modicon programmable controllers. It has since become an industry standard method for the transfer of discrete/analog I/O information and register data between industrial control and monitoring devices. MODBUS is now a widely-accepted, open, public-domain protocol that requires a license, but does not require royalty payment to its owner.

MODBUS devices communicate using a master-slave (client-server) technique in which only one device (the Client(Master)) can initiate

transactions (called queries). The other devices (slaves/servers) respond by supplying the requested data to the master, or by taking the action requested in the query. A slave is any peripheral device (I/O transducer, valve, network drive, or other measuring device) which processes information and sends its output to the master using MODBUS. The Acromag I/O Modules form slave/server devices, while a typical master device is a host computer running appropriate application software. Other devices may function as both clients (masters) and servers (slaves).

Masters can address individual slaves, or can initiate a broadcast message to all slaves. Slaves return a response to all queries addressed to them individually, but do not respond to broadcast queries. Slaves do not initiate messages on their own, they only respond to queries from the master.

A master's query will consist of a slave address (or broadcast address), a function code defining the requested action, any required data, and an error checking field. A slave's response consists of fields confirming the action taken, any data to be returned, and an error checking field. Note that the query and response both include a device address, a function code, plus applicable data, and an error checking field. If no error occurs, the slave's response contains the data as requested. If an error occurs in the query received, or if the slave is unable to perform the action requested, the slave will return an exception message as its response (see MODBUS Exceptions). The error check field of the slave's message frame allows the master to confirm that the contents of the message are valid. Traditional MODBUS messages are transmitted serially and parity checking is also applied to each transmitted character in its data frame.

At this point, It's important to make the distinction that MODBUS itself is an application protocol, as it defines rules for organizing and interpreting data, but remains simply a messaging structure, independent of the underlying physical layer. As it happens to be easy to understand, freely available, and accessible to anyone, it is thus widely supported by many manufacturers.

Continued on next page.

WHAT IS MODBUS TCP/IP?

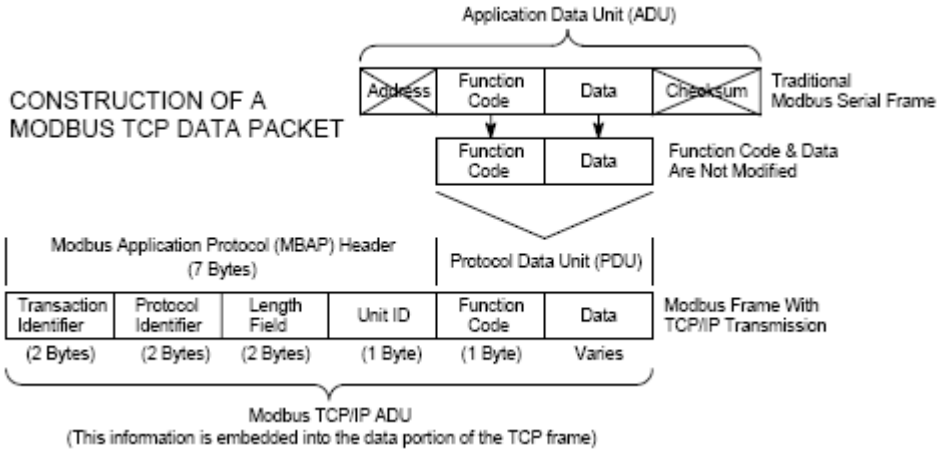
MODBUS TCP/IP (also MODBUS-TCP) is simply the MODBUS RTU protocol with a TCP interface that runs on Ethernet.
The MODBUS messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

TCP/IP refers to the Transmission Control Protocol and Internet Protocol, which provides the transmission medium for MODBUS TCP/IP messaging.

Simply stated, TCP/IP allows blocks of binary data to be exchanged between computers. It is also a world-wide standard that serves as the foundation for the World Wide Web. The primary function of TCP is to ensure that all packets of data are received correctly, while IP makes sure that messages are correctly addressed and routed. Note that the TCP/IP combination is merely a transport protocol, and does not define what the data means or how the data is to be interpreted (this is the job of the application protocol, MODBUS in this case).

So in summary, MODBUS TCP/IP uses TCP/IP and Ethernet to carry the data of the MODBUS message structure between compatible devices. That is, MODBUS TCP/IP combines a physical network (Ethernet), with a networking standard (TCP/IP), and a standard method of representing data (MODBUS as the application protocol). Essentially, the MODBUS TCP/IP message is simply a MODBUS communication encapsulated in an Ethernet TCP/IP wrapper.

In practice, MODBUS TCP embeds a standard MODBUS data frame into a TCP frame, without the MODBUS checksum, as shown in the following diagram.



The MODBUS commands and user data are themselves encapsulated into the data container of a TCP/IP telegram without being modified in any way. However, the MODBUS error checking field (checksum) is not used, as the standard Ethernet TCP/IP link layer checksum methods are instead used to guaranty data integrity. Further, the MODBUS frame address field is supplanted by the unit identifier in MODBUS TCP/IP, and becomes part of the MODBUS Application Protocol (MBAP) header (more on this later).

From the figure, we see that the function code and data fields are absorbed in their original form. Thus, a Modbus TCP/IP Application Data Unit (ADU) takes the form of a 7 byte header (transaction identifier + protocol identifier + length field + unit identifier), and the protocol data unit (function code + data). The MBAP header is 7 bytes long and includes the following fields:

- **Transaction/invocation Identifier (2 Bytes):** This identification field is used for transaction pairing when multiple messages are sent along the same TCP connection by a client without waiting for a prior response.
- **Protocol Identifier (2 bytes):** This field is always 0 for MODBUS services and other values are reserved for future extensions.
- **Length (2 bytes):** This field is a byte count of the remaining fields and includes the unit identifier byte, function code byte, and the data fields.
- **Unit Identifier (1 byte):** This field is used to identify a remote server located on a non TCP/IP network (for serial bridging). In a typical MODBUS TCP/IP server application, the unit ID is set to 00 or FF, ignored by the server, and simply echoed back in the response.

The complete MODBUS TCP/IP Application Data Unit is embedded into the data field of a standard TCP frame and sent via TCP to well-known system port 502, which is specifically reserved for MODBUS applications. MODBUS TCP/IP clients and servers listen and receive MODBUS data via port 502.
We can see that the operation of MODBUS over Ethernet is nearly transparent to the MODBUS register/command structure. Thus, if you are already familiar with the operation of traditional MODBUS, then you are already very with the operation of MODBUS TCP/IP.

A.1 "0" Device (Coil)

(1) Read Single Coil : 01

Describes "01" command frame through the example where "**0**00020–**0**00056 Coil" data of the Slave device side (prefix: 17) is read from the MASTER device.

■ RTU Mode

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Device score	
	H	L	H	L	H	L			H	L	H	L
Hex	00	01	00	00	00	06	11	01	00	13	00	25

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Number of data (bytes)	Data				
										Coils 27~20	Coils 35~28	Coils 43~36	Coils 51~44	Coils 56~52
	H	L	H	L	H	L				L	–	–	–	H
Hex	00	01	00	00	00	08	11	01	05	CD	6B	B2	0E	1B

■ Coils data status

Coils	27	26	25	24	23	22	21	20
on/off	1	1	0	0	1	1	0	1
Coils	35	34	33	32	31	30	29	28
on/off	0	1	1	0	1	0	1	1
Coils	43	42	41	40	39	38	37	36
on/off	1	0	1	1	0	0	1	0
Coils	51	50	49	48	47	46	45	44
on/off	0	0	0	0	1	1	1	0
Coils	59	58	57	56	55	54	53	52
on/off	–	–	–	1	1	0	1	1

0: OFF /

(2) Force Single Coil : 05

Describes "05" command frame through an example where FORCE "ON" is done on Coil 000173 of the Slave device side in the MASTER device.

■ RTU Mode

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Force data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	02	00	00	00	06	11	05	00	AC	FF	00

■ Force Data

	High	Low
Force ON	FF_H	00_H
Force OFF	00_H	00_H

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Force data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	02	00	00	00	06	11	05	00	AC	FF	00

A.2 "1" Device (Discrete Input)

(1) Read Input Status : 02

Describes "02" command frame through an example where "**1**00197–**1**00218 Input" data of the Slave device side (prefix: 17) is read from the MASTER device.

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Device score	
	H	L	H	L	H	L			H	L	H	L
Hex	00	03	00	00	00	06	11	02	00	C4	00	16

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Number of data	Data (Inputs)		
	H	L	H	L	H	L				10204~10197	10212~10205	10218~10213
Hex	00	03	00	00	00	06	11	02	03	AC	DB	35

■ Coils data status

Coils	204	203	202	201	200	199	198	197
on/off	1	0	1	0	1	1	0	0
Coils	212	211	210	209	208	207	206	205
on/off	1	1	0	1	1	0	1	1
Coils	220	219	218	217	216	215	214	213
on/off	–	–	1	1	0	1	0	1

0: OFF / 1:ON

A.3 "3" Device (Input Register)

(1) Read Input Registers : 04

Describes "03" command frame through an example where "**3**00009 Register" data of the Slave device side (prefix: 17) is read from the MASTER device.

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Device score (Word Count)	
	H	L	H	L	H	L			H	L	H	L
Hex	00	04	00	00	00	06	11	04	00	08	00	01

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Number of data (bytes)	Data 30009 Register	
	H	L	H	L	H	L				H	L
Hex	00	04	00	00	00	05	11	04	02	00	0A

A.4 "4" Device (Holding Register)

(1) Read Holding Registers : 03

Describes "03" command frame through an example where "**4**00108 – **4**00110 Register" data of the Slave device side (prefix: 17) is read from the MASTER device.

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Device score	
	H	L	H	L	H	L			H	L	H	L
Hex	00	05	00	00	00	06	11	03	00	6B	00	03

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Number of data (bytes)	Data					
										Register 40108		Register 40109		Register 40110	
	H	L	H	L	H	L				H	L	H	L	H	L
Hex	00	05	00	00	00	09	11	03	06	02	2B	00	00	00	64

(2) Preset Single Register : 06

Describes "06" command frame through an example where 00 03 (hex) data is entered in **4**00002 Register of the Slave device side .

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Preset data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	06	00	00	00	06	11	06	00	01	00	03

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Preset data	
	H	L	H	L	H	L			H	L	H	L
Hex	00	06	00	00	00	06	11	06	00	01	00	03

(3) Preset Multiple Register : 10

Describes "10" command frame through an example where two consecutive data, "00 0A (hex)", "01 02 (hex)" are entered in **4**00002 Register of the Slave device side. (Error Code : 90_H)

(Master → Slave: request frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Quantity of Register (Word Count)		Number of data (bytes)	Data			
														Register 40002		Register 40003	
	H	L	H	L	H	L			H	L	H	L		H	L	H	L
Hex	00	07	00	00	00	0B	11	10	00	01	00	02	04	00	0A	01	02

(Slave → Master: response frame)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave prefix)	Command	Leading device		Quantity of Register (Word Count)	
	H	L	H	L	H	L			H	L	H	L
Hex	00	07	00	00	00	06	11	10	00	01	00	02