MODBUS Organization, Inc.

MODBUS-TCP Client(Master) Driver

지원버전	OS	V4.0 이상
	XDesignerPlus	4.0.0.0 이상

CONTENTS

본사 ㈜M2I의 "Touch Operation Panel(M2I TOP) Series"를 사용해주시는 고객님께 감사드립니다. 본 매뉴얼을 읽고 "TOP–외부장치"의 접속 방법 및 절차를 숙지해 주십시오.

1. 시스템 구성 2 페이지

접속에 필요한 기기, 각 기기의 설정, 케이블, 구성 가능한 시스템에 대해 설명합니다.
본 절을 참조하여 적절한 시스템을 선정하십시오.

2. TOP 기종과 외부 장치 선택 3 페이지

TOP 기종과 외부 장치를 선택합니다.

3. 시스템 설정 예제 4 페이지

본 기기와 해당 외부 단말기의 통신 접속을 위한 설정 예제를 설명 합니다.
"1. 시스템 구성"에서 선택한 시스템에 따라 예제를 선택 하십시오.

4. 통신 설정 항목 6 페이지

TOP 통신 설정 하는 방법에 대해서 설명합니다.
외부 장치의 설정이 바뀔 경우 본 장을 참조 하여 TOP의 설정도 외부 장치와 같게 설정하십시오.

5. 지원 어드레스 8 페이지

본 절을 참조하여 외부 장치와 통신 가능한 어드레스를 확인하십시오.

APPENDIX A. MODBUS Protocol 9 페이지

본 기기의 "MODBUS Serial Master Driver"가 지원하는 MODBUS 프로토콜 명령어 및 디바이스에 대해 설명 합니다.

1. 시스템 구성

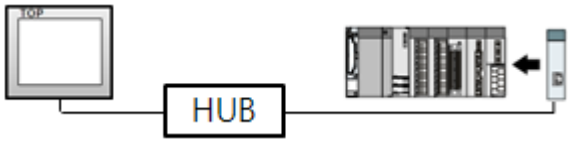
본 드라이버는 "MODBUS Organization, Inc."의 "MODBUS Protocol" 중 "MODBUS-TCP Client(Master)" 입니다.
외부 장치(MODBUS Slave Protocol 지원)에 따라서 드라이버의 "명령어 코드", "프로토콜 프레임 형식" 등을 별도 설정 해야 할 수 있습니다. 이 경우 통신 방식에 따른 세부 설정 사항을 외부 장치 측에 맞추어 설정 해주십시오.
본 드라이버가 지원하는 외부 장치와의 시스템 구성은 아래와 같습니다.

시리즈	CPU	Link I/F	통신 방식	시스템 설정	케이블
MODBUS Slave/Server Device			Ethernet (UDP)	3.1 설정 예제 1 (페이지)	트위스트 페어 케이블*주1)
			Ethernet (TCP)	3.2 설정 예제 2 (페이지)	

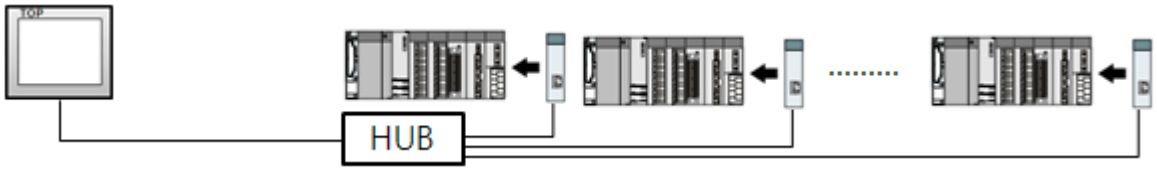
*주1) 트위스트 페어 케이블
- STP(실드 트위스트 페어 케이블) 혹은 UTP(비실드 트위스트 페어 케이블) 카테고리 3, 4, 5 를 의미 합니다.
- 네트 워크 구성에 따라 허브, 트랜시버 등의 구성기기에 접속 가능하며 이 경우 다이렉트 케이블을 사용 하십시오.

■ 연결 가능 구성

• 1 : 1 연결(TOP 1 대와 외부 장치 1 대) 연결

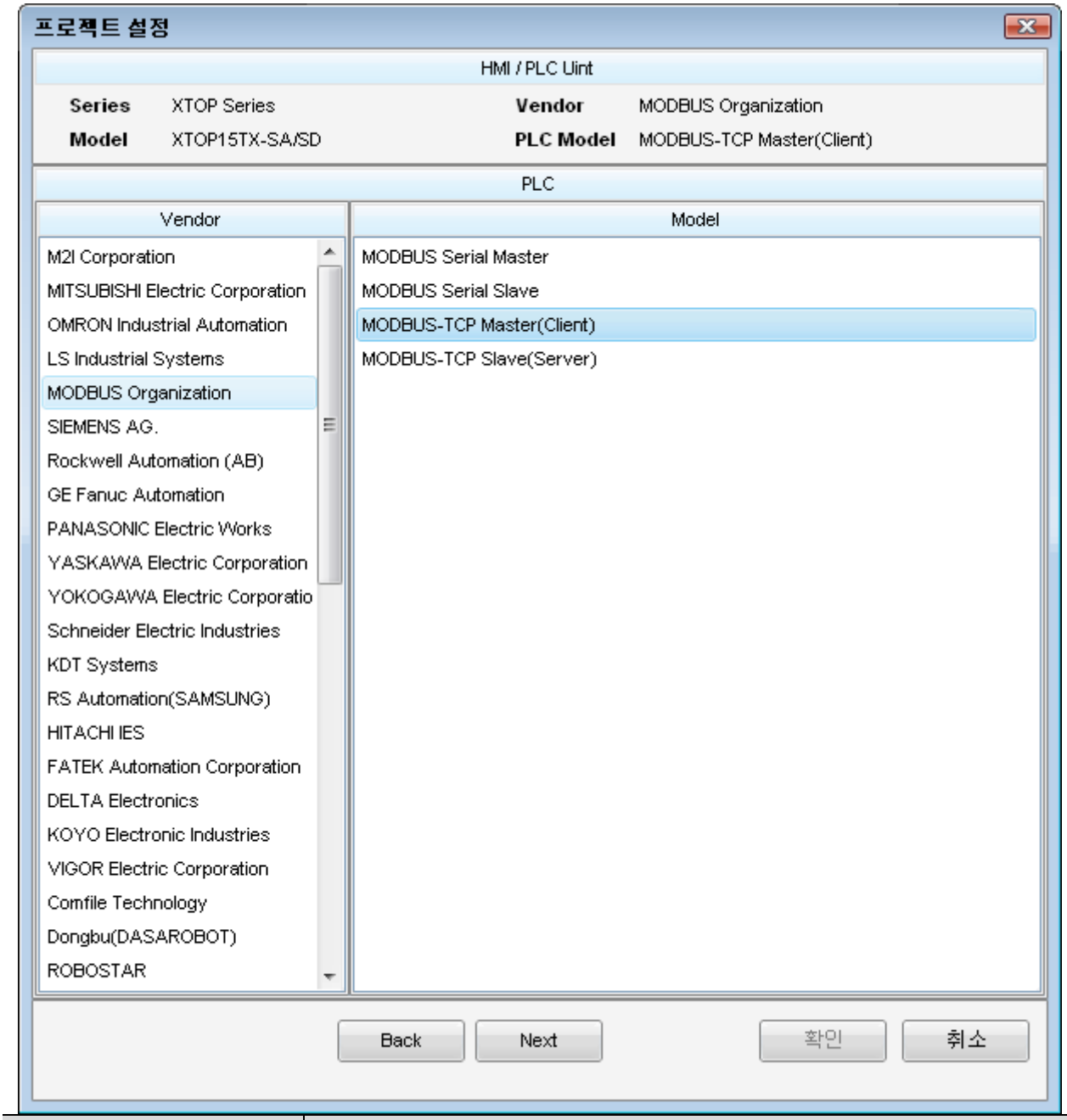


• 1 : N 연결(TOP 1 대와 외부 장치 여러 대) 연결



2. TOP 기종과 외부 장치 선택

TOP와 연결 될 외부 장치를 선택 합니다.



<table>
<tr><th colspan="2">설정 사항</th><th>내용</th></tr>
<tr><td rowspan="2">TOP</td><td>Series</td><td>PLC와 연결할 TOP의 시리즈 명칭을 선택합니다.
설정 내용을 Download 하기 전에 TOP의 시리즈에 따라 아래 표에 명시된 버전의 OS를 인스톨 하십시오.
<table>
<tr><th>시리즈</th><th>버전 명칭</th></tr>
<tr><td>XTOP / HTOP</td><td>V4.0</td></tr>
</table></td></tr>
<tr><td>Name</td><td>TOP 제품 모델명을 선택합니다.</td></tr>
<tr><td rowspan="2">외부 장치</td><td>제조사</td><td>TOP와 연결할 외부 장치의 제조사를 선택합니다.
“MODBUS Organization, Inc.”를 선택 하십시오.</td></tr>
<tr><td>PLC</td><td>TOP에 연결 될 외부 장치의 모델 시리즈를 선택 합니다.
“MODBUS TCP Client(Master)”를 선택 하십시오.
연결을 원하는 외부 장치가 시스템 구성 가능한 기종인지 1장의 시스템 구성에서 확인 하시기 바랍니다.</td></tr>
</table>

3. 시스템 설정 예제

TOP와 "MODBUS TCP Slave Device"의 통신 인터페이스 설정을 아래와 같이 권장 합니다.

3.1 설정 예제 1

구성한 시스템을 아래와 같이 설정 한다.

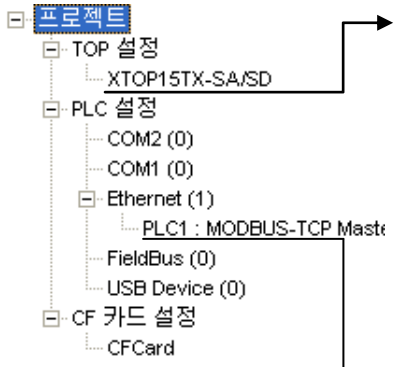
항목	TOP	Slave Device	비고
IP Address*주1)주2)	192.168.0.50	192.168.0.51	유저 설정
포트	Don`t Care	502	유저 설정
프로토콜	UDP		유저 설정

*주1) TOP와 외부 장치의 네트워크 주소 (IP 앞 세자리 192.168.000)는 일치 해야 한다.

*주2) 동일 네트워크 상에서 중복된 IP 주소를 사용하지 마십시오.

(1) XDesignerPlus 설정

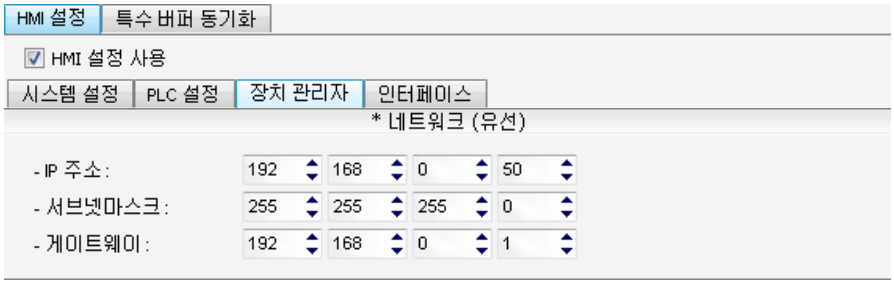
[프로젝트 > 프로젝트 설정]에서 아래 내용을 설정 후, TOP 기기로 설정 내용을 다운로드 합니다..



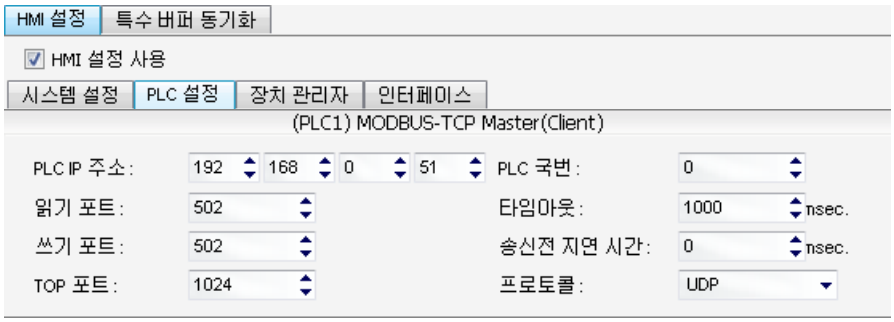
■ [프로젝트 > 프로젝트 속성 > 프로젝트 > 설정 > TOP Name].

TOP 기기의 통신 인터페이스를 설정 합니다.

– 우측 윈도우에서 [HMI 설정 > HMI 설정 사용 체크 > 장치 관리자]

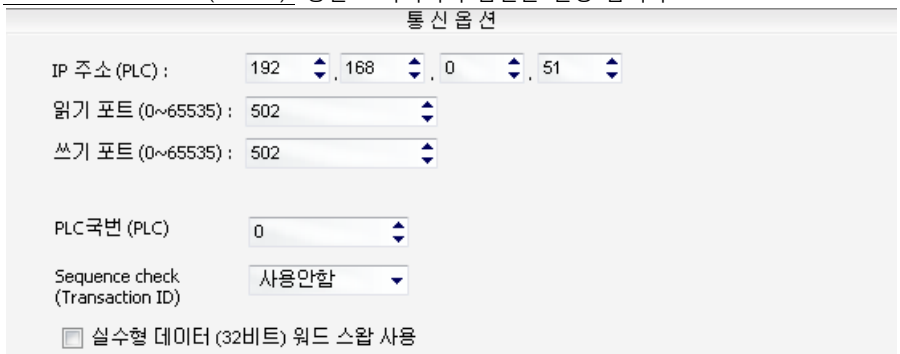


– 우측 윈도우에서 [HMI 설정 > HMI 설정 사용 체크 > PLC 설정]



■ 외부 장치 설정

"MODBUS TCP Client(Master)" 통신 드라이버의 옵션을 설정 합니다.



– IP 주소 (PLC): 외부 장치에 할당한 IP 번호를 기입합니다.

– 읽기 포트 / 쓰기 포트: 외부 장치의 이더넷 통신에 사용할 포트 번호를 선택합니다.

– PLC 국번(PLC) : 외부장치 설정 국번.

– Sequence check(Transaction ID) : 처리하는 과정을 한번 더 체크합니다.

– 실수형 데이터(32비트)워드 스왑 사용 : 실수 형 데이터의 High/ Low Word의 순서를 Low/High Word 순서로 설정 합니다.

(2) 외부 장치 설정

외부 장치의 사용자 매뉴얼을 참조하여 외부기기 I/F에 "MODBUS Serial Slave Driver"를 설정 하십시오.

- 동일 네트워크 상에서 IP 어드레스를 중복 사용하지 마십시오.
- 외부 장치 측 어드레스 맵 내용을 확인하고 그 내용에 따라 통신 어드레스를 사용하십시오.

3.2 설정 예제 2

구성한 시스템을 아래와 같이 설정 한다.

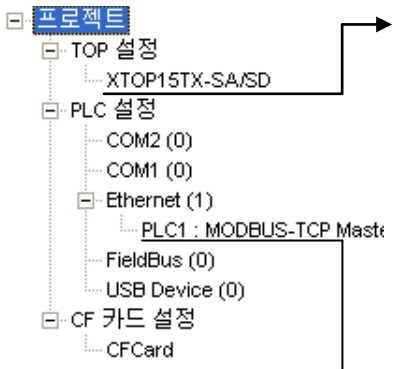
항목	TOP	Slave Device	비고
IP Address*주1)주2)	192.168.0.50	192.168.0.51	유저 설정
포트	Don`t Care	502	유저 설정
프로토콜	TCP		유저 설정

*주1) TOP와 외부 장치의 네트워크 주소 (IP 앞 세자리 192.168.000)는 일치 해야 한다.

*주2) 동일 네트워크 상에서 중복된 IP 주소를 사용하지 마십시오.

(1) XDesignerPlus 설정

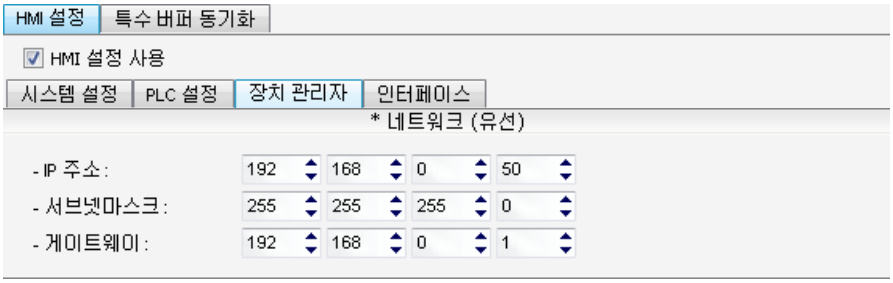
[프로젝트 > 프로젝트 설정]에서 아래 내용을 설정 후, TOP 기기로 설정 내용을 다운로드 합니다..



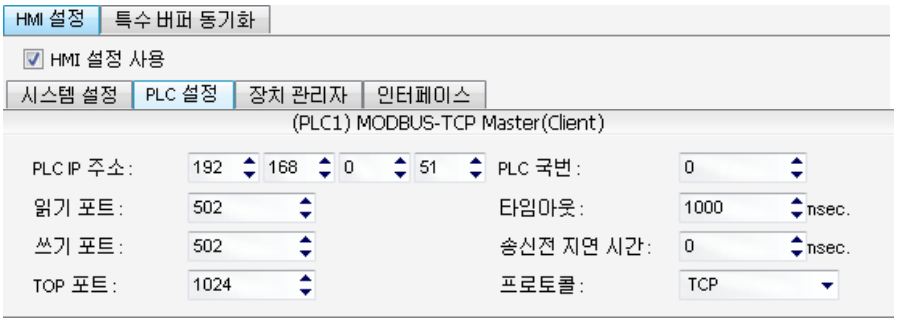
■ [프로젝트 > 프로젝트 속성 > 프로젝트 > 설정 > TOP Name].

TOP 기기의 통신 인터페이스를 설정 합니다.

– 우측 윈도우에서 [HMI 설정 > HMI 설정 사용 체크 > 장치 관리자]

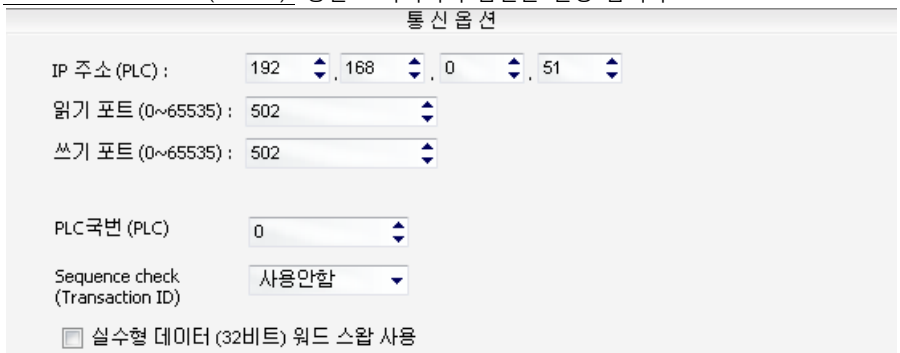


– 우측 윈도우에서 [HMI 설정 > HMI 설정 사용 체크 > PLC 설정]



■ 외부 장치 설정

"MODBUS TCP Client(Master)" 통신 드라이버의 옵션을 설정 합니다.



– IP 주소 (PLC): 외부 장치에 할당한 IP 번호를 기입합니다.

– 읽기 포트 / 쓰기 포트: 외부 장치의 이더넷 통신에 사용할 포트 번호를 선택합니다.

– PLC 국번(PLC) : 외부장치 설정 국번.

– Sequence check(Transaction ID) : 처리하는 과정을 한번 더 체크합니다.

– 실수형 데이터(32비트)워드 스왑 사용 : 실수 형 데이터의 High/ Low Word의 순서를 Low/High Word 순서로 설정 합니다.

(2) 외부 장치 설정

외부 장치의 사용자 매뉴얼을 참조하여 외부기기 I/F에 "MODBUS Serial Slave Driver"를 설정 하십시오.

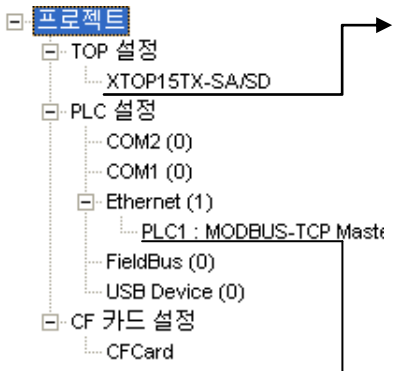
- 동일 네트워크 상에서 IP 어드레스를 중복 사용하지 마십시오.
- 외부 장치 측 어드레스 맵 내용을 확인하고 그 내용에 따라 통신 어드레스를 사용하십시오.

4. 통신 설정 항목

통신 설정은 XDesignerPlus 혹은 TOP 메인 메뉴에서 설정 가능 합니다. 통신 설정은 외부 장치와 동일하게 설정 해야 합니다.

4.1 XDesignerPlus 설정 항목

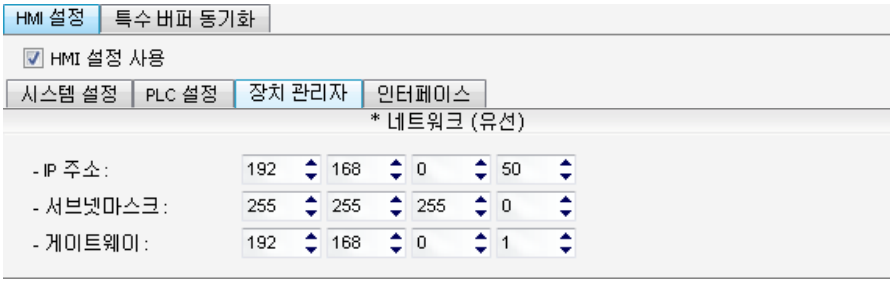
아래 창의 내용을 나타내기 위해서 [프로젝트 > 프로젝트 속성]을 선택 하십시오



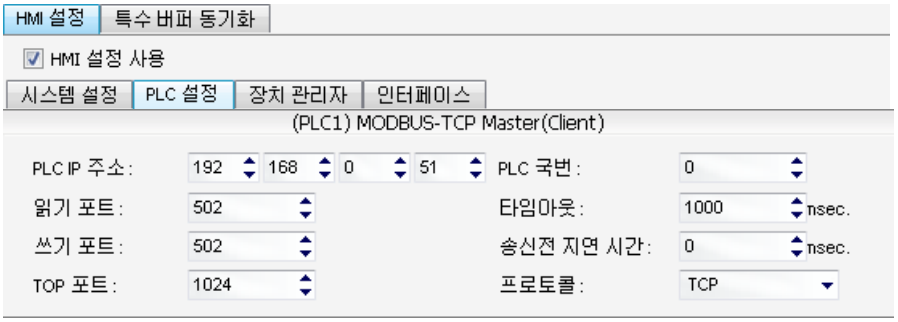
■ [프로젝트 > 프로젝트 속성 > 프로젝트 > 설정 > TOP Name].

TOP 기기의 통신 인터페이스를 설정 합니다.

– 우측 윈도우에서 [HMI 설정 > HMI 설정 사용 체크 > 장치 관리자]

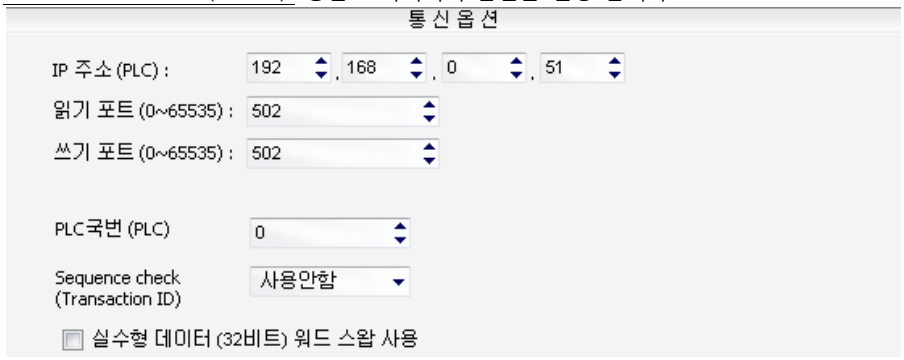


– 우측 윈도우에서 [HMI 설정 > HMI 설정 사용 체크 > PLC 설정]



■ 외부 장치 설정

"MODBUS TCP Client(Master)" 통신 드라이버의 옵션을 설정 합니다.

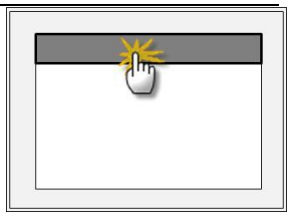


■ 통신 인터페이스 설정

항목	내용
IP 주소	네트워크 상에서 TOP에 부여하는 IP 주소를 설정 합니다.
서브넷마스크	네트워크의 서브넷마스크를 기입합니다.
게이트웨이	네트워크의 서브넷마스크를 기입합니다.
PLC IP 주소	외부 장치에 할당한 IP 번호를 기입합니다.
읽기 포트 / 쓰기 포트	외부 장치의 이더넷 통신에 사용할 포트 번호를 선택합니다.
TOP 포트	외부장치와 이더넷 통신 할 경우 포트 번호는 자동 설정 됩니다.
PLC 국번. [0~65535]	상대 기기의 국번입니다. [0 – 65535] 사이의 값을 선택합니다.
이더넷 타임아웃	TOP가 외부 장치로부터의 응답을 기다리는 시간을 [0 – 99] x 100 mSec 로 설정합니다.
송신전 지연시간 [x1 mSec]	TOP가 외부 장치로부터 응답 수신 – 다음 명령어 요청 전송 간에 대기하는 시간을 [0 – 5000] x 1 mSec 로 설정합니다.
프로토콜	외부장치와 설정 포트 번호에 따라 허용된 프로토콜 방식을 선택 합니다.

4.2 TOP 메인 메뉴 설정 항목

- 전원을 리셋 중 부저음이 울릴 때 LCD 상단 1점을 터치하여 "TOP 관리 메인" 화면으로 이동합니다.
- TOP에서 드라이버 인터페이스 설정은 아래의 **Step1** → **Step2** 내용을 따라 설정합니다.
  (**Step 1**.에서 "TOP 이더넷 설정"을 누르시면 **Step2**.에서 설정을 바꾸실 수 있습니다.)



Step 1. [PLC 설정] – 드라이버 인터페이스를 설정 합니다.

PLC 설정	
PLC IP : 192 . 168 . 0 . 51 프로토콜 : UDP PLC 읽기 포트 : 502 PLC 쓰기 포트 : 502 TOP 포트 : 1024 PLC 국번 : 0 타임아웃 : 1000 [mSec] 송신전 지연 시간 : 0[mSec] TOP IP : 192 . 168 . 0 . 50 TOP 이더넷 설정 통신 진단	통신 인터페이스 설정

Step 1–Reference.

항목	내용
PLC IP	외부 장치에 할당한 IP 번호입니다.
프로토콜	외부장치와 설정 포트 번호에 따라 허용된 프로토콜 방식을 선택 합니다.
PLC 읽기 포트	외부 장치의 이더넷 통신에 사용할 포트 번호입니다.
PLC 쓰기 포트	외부 장치의 이더넷 통신에 사용할 포트 번호입니다.
TOP 포트	외부장치와 이더넷 통신 할 경우 포트 번호는 자동 설정 됩니다.
PLC 국번. [0~65535]	상대 기기의 국번입니다. [0 – 65535] 사이의 값을 선택합니다.
타임아웃 [x1 mSec]	TOP가 외부 장치로부터의 응답을 기다리는 시간을 [0 – 5000] x 1 mSec 로 설정합니다.
송신전 지연 시간 [x1 mSec]	TOP가 외부 장치로부터 응답 수신 – 다음 명령어 요청 전송 간에 대기하는 시간을 [0 – 5000] x 1 mSec 로 설정합니다.
TOP IP	네트워크 상에서 TOP에 부여하는 IP 주소를 설정 합니다

Step 2. [PLC 설정] >[TOP 이더넷 설정] - 해당 포트의 시리얼 파라미터를 설정 합니다.

포트 설정	
* 이더넷 통신 + 네트워크 설정 - MAC : 00 – 15 – ID – 00 – 30 – 52 (기기마다 다른 고유 주소) - IP 주소 : 192 . 168 . 0 . 50 - 서브넷마스크 : 255 . 255 . 255 . 0 - 게이트웨이 : 192 . 168 . 0 . 1	이더넷 포트 통신 인터페이스 설정

Step 2-Reference.

항목	내용
MAC	네트워크 상의 물리적인 고유 주소입니다.
IP 주소	네트워크 상에서 TOP에 부여하는 IP 주소를 설정 합니다
서브넷마스크	IP주소에 대한 네트워크 아이디와 호스트 아디를 구분하는 주소입니다.
게이트웨이	네트워크와 다른 네트워크가 연결되는 주소입니다.

4.3 통신 진단

■ TOP – 외부 장치 간 인터페이스 설정 상태를 확인

- TOP의 전원을 리셋 하면서 LCD 창의 상단을 클릭하여 메뉴 화면으로 이동한다.
- [메인 메뉴 >통신 설정] 20~24 번 내용이 "■설정 예제 1"의 설정 내용과 같은지 확인한다
- PLC 설정 > TOP 이더넷 "통신 진단"의 버튼을 클릭한다.
- 화면 상에 Diagnostics 다이얼로그 박스가 팝업 되며, 박스의 3번 항에 표시된 내용에 따라 진단 상태를 판단한다.

OK!	**통신 설정 정상**
Time Out Error!	통신 설정 비 정상 - 케이블 및 TOP/외부 장치의 설정 상태를 에러 **(참조 : 통신 진단 시트)**

■ 통신 진단 시트

- 외부 단말기와 통신 연결에 문제가 있을 경우 아래 시트의 설정 내용을 확인 바랍니다.

항목	내용				확인	
TOP	버전 정보		xDesignerPlus :	O.S :		
	드라이버 명칭				OK	NG
	외부 장치 정보 (xDesignerPlus의 프로젝트 설정)	IP Address			OK	NG
		서브넷마스크			OK	NG
		게이트 웨이			OK	NG
	TOP 정보 (본체 메뉴설정)	프로토콜	UDP/IP	TCP/IP	OK	NG
		IP Address			OK	NG
		서브넷마스크			OK	NG
		게이트 웨이			OK	NG
	기타 세부 설정 사항				OK	NG
시스템 구성	시스템 연결 방법		1:1	1:N	N:1	OK	NG
	케이블 명칭(허브 사용 유무)		다이렉트(허브사용)	크로스(허브미사용)		OK	NG
외부 장치	CPU 명칭					OK	NG
	통신 모듈 명칭					OK	NG
	프로토콜(모드)					OK	NG
	기타 세부 설정 사항					OK	NG
	IP Address		(Local)	(Destination)		OK	NG
	포트 번호		(Local)	(Destination)		OK	NG
	서브넷 마스크					OK	NG
	게이트 웨이					OK	NG
	어드레스 범위 확인(별도자료)					OK	NG

5. 지원 어드레스

TOP에서 사용 가능한 디바이스는 아래와 같습니다.

CPU 모듈 시리즈/타입에 따라 디바이스 범위(어드레스) 차이가 있을 수 있습니다. TOP 시리즈는 외부 장치 시리즈가 사용하는 최대 어드레스 범위를 지원합니다. 사용하고자 하는 장치가 지원하는 어드레스 범위를 벗어 나지 않도록 각 CPU 모듈 사용자 매뉴얼을 참조/주의 하십시오.

	Bit Address	Word Address	32 bits	Remarks
Coil	000001 – 065536	000001 – 065521	L/H	
Discrete Input	100001 – 165536	100001 – 165521		*주1)
Input Register	300001.00 – 365536.15	300001 – 365536		*주1)
Holding Register	400001.00 – 465536.15	400001 – 465536		

***주1)** 쓰기 불가능(읽기 전용)

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

본 기기의 "MODBUS TCP Client(Master) Driver"가 지원하는 MODBUS 프로토콜 명령어 및 디바이스에 대해 설명 합니다.

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.

다음 페이지에 계속 됩니다.

WHAT IS MODBUS TCP/IP?

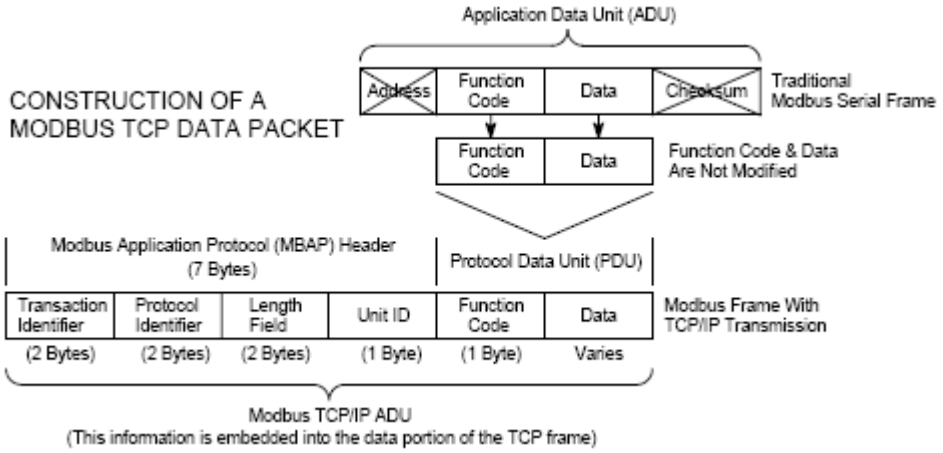
MODBUS TCP/IP (also MODBUS-TCP) is simply the MODBUS RTU protocol with a TCP interface that runs on Ethernet.
The MODBUS messaging structure is the application protocol that defines the rules for organizing and interpreting the data independent of the data transmission medium.

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

MASTER 기기에서 Slave 기기 측(국번:17번)의 "**0**00020–**0**00056 Coil" 데이터를 읽어 오는 예제를 통해 "01"명령어 프레임을 설명 합니다.

■ RTU Mode

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 규격)	명령어	전부디바이스		디바이스 점수	
	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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 규격)	명령어	데이터 바이트 개수(byte)	데이터				
	H	L	H	L	H	L				Coils 27~20	Coils 35~28	Coils 43~36	Coils 51~44	Coils 56~52
										L	–	–	–	H
Hex	00	01	00	00	00	08	11	01	05	CD	6B	B2	0E	1B

■ Coils 데이터 상태

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

MASTER 기기에서 Slave 기기 측의 Coil **0**00173 에 FORCE "ON" 하는 예제를 통해 "05"명령어 프레임을 설명 합니다.

■ RTU Mode

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 규격)	명령어	전부디바이스		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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 규격)	명령어	전부디바이스		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

MASTER 기기에서 Slave 기기 측(국번:17번)의 "**1**00197–**1**00218 Input" 데이터를 읽어 오는 예제를 통해 "02"명령어 프레임을 설명합니다.

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	첫번째 디바이스		디바이스 점수	
	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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	데이터 개수(byte)	데이터(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 데이터 상태

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

MASTER 기기에서 Slave 기기 측(국번:17번)의 "**3**00009 Register" 데이터를 읽어 오는 예제를 통해 "03"명령어 프레임을 설명 합니다.

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	전부디바이스		디바이스점수 (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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	데이터 개수(byte)	데이터 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

MASTER 기기에서 Slave 기기 측(국번:17)의 "**4**00108 – **4**00110 Register" 데이터를 읽어 오는 예제를 통해 "03"명령어 프레임을 설명 합니다.

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	전두디바이스		디바이스점수	
	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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	데이터 개수(byte)	데이터					
										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

Slave 기기 측의 **4**00002 Register 에 00 03 (hex) 데이터를 입력 하는 예제를 통해 "06"명령어 프레임을 설명 합니다.

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	전두디바이스		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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 국번)	명령어	전두디바이스		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

Slave 기기 측의 **4**00002 Register 에 "00 0A (hex)", "01 02 (hex)" 연속한 두 개의 데이터를 입력 하는 예제를 통해 "10"명령어 프레임을 설명 합니다. (Error Code : 90_H)

(Master → Slave : 요청 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 규격)	명령어	전부디바이스		Quantity of Register (Word Count)		데이터 개수(Byte)	데이터			
														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 : 응답 프레임)

Comment	Transaction Identifier		Protocol Identifier		Length Field		Unit ID (Slave 규격)	명령어	전부디바이스		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