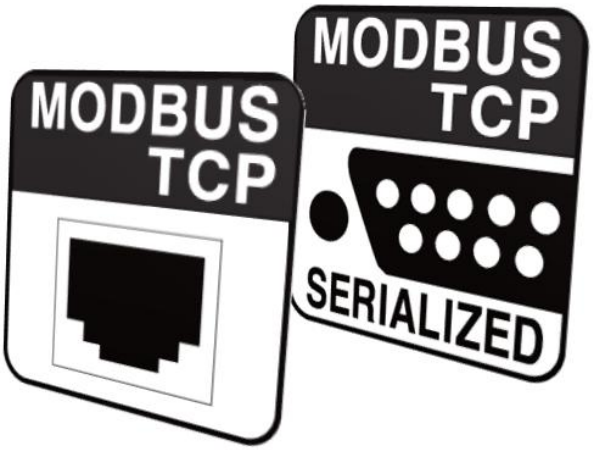
ezTCP 기술자료

모드버스 TCP

Version 1.1

2010-07-20



☞ *주의: 이 문서의 내용은 제품 성능향상을 위해서 예고 없이 변경될 수 있습니다.*

솔내시스템㈜

http://www.sollae.co.kr

목차

1 개요 - 2 -

2 모드버스TCP - 3 -

2.1 특징 - 3 -
2.2 구성요소 - 3 -
2.2.1 모드버스TCP 마스터 - 3 -
2.2.2 모드버스TCP 슬레이브 - 3 -
2.3 헤더 데이터 구조 - 4 -
2.4 클래스 0의 함수들 - 4 -
2.4.1 Read Multiple Registers - 4 -
2.4.2 Write Multiple Registers - 4 -
2.5 통신 데이터 종류 - 5 -
2.5.1 Command Request of Read Multiple Registers - 5 -
2.5.2 Command Response of Read Multiple Registers - 6 -
2.5.3 Command Exception of Read Multiple Registers - 7 -
2.5.4 Command Request of Write Multiple Registers - 8 -
2.5.5 Command Response of Write Multiple Registers - 9 -
2.5.6 Command Exception of Write Multiple Registers - 10 -
2.6 사용하기 - 11 -
2.6.1 모드버스TCP 설정 - 11 -
2.6.2 설정 예 - 12 -
2.7 샘플 코드 - 13 -
2.7.1 주요 함수 - 13 -

3 시리얼 모드버스TCP - 14 -

3.1 특징 - 14 -
3.2 사용하기 - 14 -
3.2.1 설정 방법 - 14 -
3.3 시험 작동 - 15 -
3.3.1 통신 준비 - 15 -
3.3.2 시험 데이터 전송 - 16 -

4 문서 변경 이력 - 17 -

1 개요

모드버스(Modbus)는 전세계적으로 PLC(Programmable Logic Controller) 등에 널리 사용되고 있는 직렬 통신 프로토콜 입니다. 모드버스의 버전 중 모드버스TCP는 TCP/IP 네트워크 상에서 수행되며 현재 ezTCP 원격 디지털 입/출력 컨트롤 제품들에 사용되고 있습니다.

모드버스TCP를 사용하려면 제품의 이더넷 포트를 이용해야 합니다. 하지만 사용자에 따라서는 시리얼 포트를 이용해 장비들을 제어/감시 하는 경우가 종종 있습니다. ezTCP의 시리얼모드버스TCP 모드는 이 경우를 위해 만들어졌습니다.

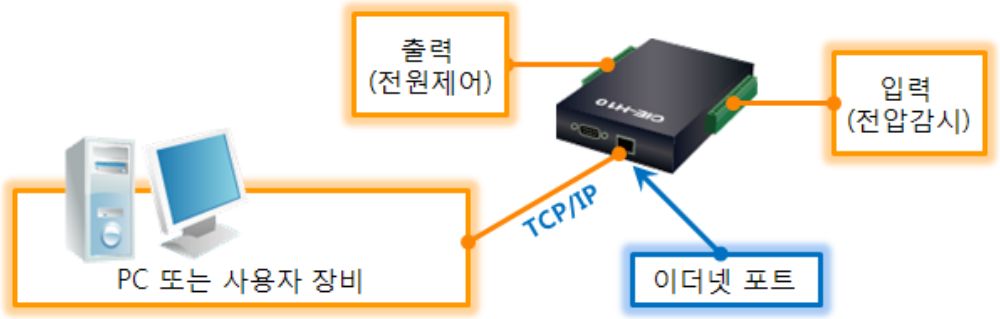


그림 1-1 모드버스TCP 사용 구성 예

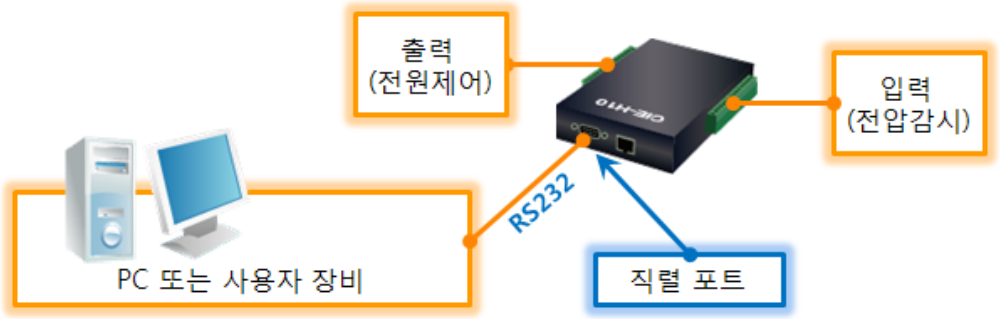


그림 1-2 시리얼모드버스TCP 사용 구성 예

ezTCP의 시리얼모드버스TCP는 앞서 언급한 모드버스TCP 프로토콜의 데이터를 직렬통신(RS232) 방식으로 송/수신하는 모드 입니다.

2 모드버스 TCP

2.1 특징

- 모드버스 프로토콜의 TCP/IP용 버전
- TCP 접속 과정이 선행 됨
  표준 모드버스TCP에서는 슬레이브가 항상 TCP 서버로 동작하지만 ezTCP는 클라이언트로도 동작이 가능합니다. 포트번호는 TCP 502번을 사용합니다.
- 마스터와 슬레이브로 구성
  표준 모드버스TCP와는 달리 슬레이브 뿐만 아니라 마스터로도 동작이 가능합니다.
- 모드버스 데이터는 빅 엔디언(Big-endian) 사용
  빅 엔디언 방식이란, 최 상위 바이트(MSB-Most Significant Byte)가 가장 낮은 주소를 갖는 방식입니다. 예를 들어 0x1234는 0x12 0x34의 순서로 정해집니다.

☞ ***MSB: 숫자 값의 크기에 가장 크게 영향을 미치는 바이트, 맨 왼쪽의 바이트***

☞ ***LSB: 숫자 값의 크기에 가장 영향을 덜 미치는 바이트, 맨 오른쪽의 바이트***

- ezTCP는 클래스 0의 함수들만 지원

표 2-1 클래스 0의 함수

함수	이름	의미
0x03	Read Multiple Registers	마스터: 입력포트 상태 정보 질의
		슬레이브: 입력포트 상태 정보 응답
0x10	Write Multiple Registers	마스터: 출력포트 출력 명령
		슬레이브: 출력포트 출력 후 응답

2.2 구성요소

2.2.1 모드버스 TCP 마스터

모드버스TCP 데이터의 요청(Request)패킷을 보냅니다. 이를 쿼리(Query)라고 하며, 일정한 주기마다 슬레이브로 보내고 응답을 수신 합니다.

2.2.2 모드버스 TCP 슬레이브

마스터의 쿼리를 수신하고 그에 대한 응답(Response)패킷을 보냅니다.

☞ ***모드버스TCP 표준에서의 슬레이브는 마스터의 쿼리에 의해서만 응답하지만, ezTCP의 [입력포트 변경 알림]기능을 이용하면, 슬레이브가 먼저 입력포트 변경 데이터를 보낼 수 있습니다.***

2.3 헤더 데이터 구조

모드버스TCP 헤더는 다음의 구조를 가지며, 모든 모드버스TCP 데이터에 선행됩니다.

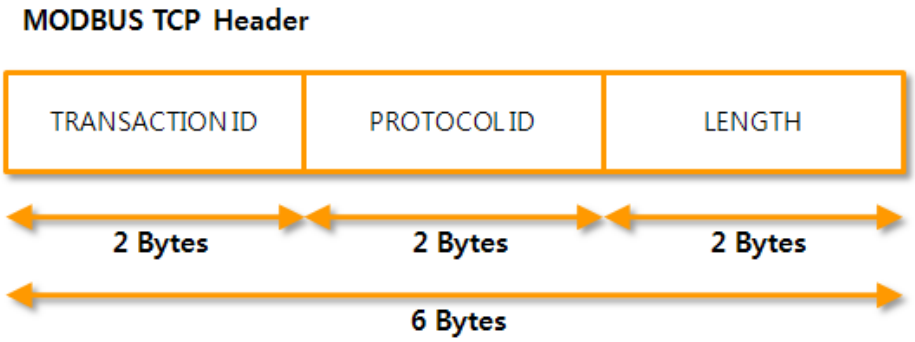


그림 2-1 모드버스TCP 헤더 데이터 구조

- TRANSACTION ID(트랜잭션 ID)
  쿼리 및 응답에 관련 한 작업의 순서번호를 나타내며 마스터에 의해 설정 됩니다. ezTCP가 마스터로 동작 할 때 0x0000부터 매 명령마다 1씩 증가시킵니다. 슬레이브로 동작 할 때에는 마스터가 설정 한 값에 맞게 응답합니다. (항상 0x0000으로 설정해도 무방)

☞ ***HEX: 이 문서에서 HEX, 또는 0xABCD로 표현된 수는 16진수를 의미합니다.***

- PROTOCOL ID(프로토콜 ID)
  프로토콜의 ID를 나타내며 0x0000으로 고정 값 입니다.
- LENGTH(길이)
  LENGTH필드 이후부터 해당 프레임의 마지막까지의 길이를 나타냅니다. (단위: Byte)

2.4 클래스 0의 함수들

ezTCP는 표준 모드버스TCP의 클래스 0에 해당하는 함수들만 지원합니다. 다음의 두 가지 함수가 여기에 해당합니다.

2.4.1 Read Multiple Registers

- 입력 값을 읽는 함수
- 함수 코드(Function code)는 0x03(HEX)

2.4.2 Write Multiple Registers

- 출력 값을 쓰는 함수
- 함수 코드(Function code)는 0x10(HEX)

2.5 통신 데이터 종류

ezTCP의 모드버스TCP 통신 데이터 종류는 예외 코드를 제외하고 마스터와 슬레이브가 각각 3(0x03)번과 16(0x10)번의 함수를 사용하여 통신하므로 총 4가지 입니다.

2.5.1 Command Request of Read Multiple Registers

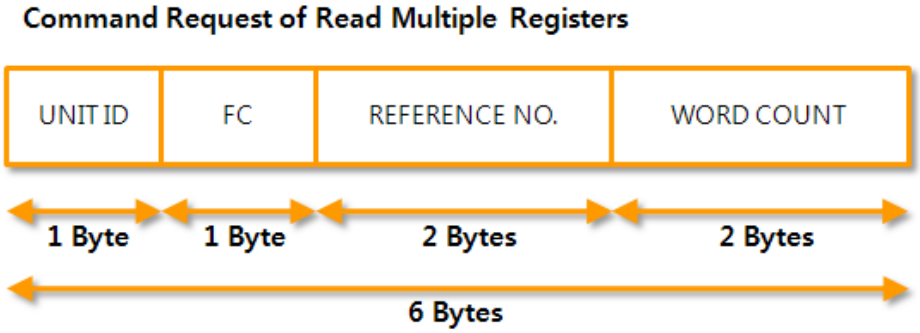


그림 2-2 마스터의 Read Multiple Registers

- UNIT ID(유니트 ID)
  설정 된 마스터와 슬레이브의 유니트 ID를 나타냅니다.
- FC(Function Code): 함수 코드
  Read Multiple Registers의 함수 코드는 0x03(HEX) 입니다.
- REFERENCE NO. (Reference Number)
  입력포트의 시작주소로 설정 된 값을 나타냅니다. (기본 값: 0x0000)
- WORD COUNT
  ezTCP는 WORD COUNT 기본 값이 0x01 입니다.
- 전송 예

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	03	00	00	00	01

그림 2-3 전송 예

표 2-2 전송 예에 대한 설명

바이트 순서	값(HEX)	의미
0~1	0x0000	트랜잭션 ID를 0x0000으로 설정
2~3	0x0000	프로토콜 ID (고정 값)
4~5	0x0006	이후 6 Bytes 까지가 한 프레임임을 나타냄 (고정 값)
6	0x01	유니트 ID가 1로 설정 됨
7	0x03	Read Multiple Registers 함수임을 나타냄 (고정 값)
8~9	0x0000	입력포트 시작주소가 0으로 설정 됨
10~11	0x0001	Word Count 기본 값 (고정 값)

2.5.2 Command Response of Read Multiple Registers

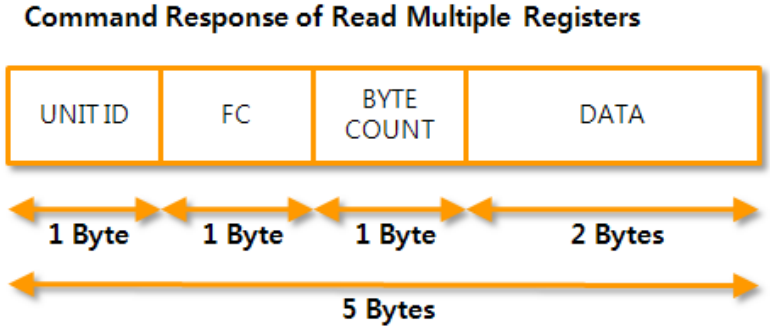


그림 2-4 슬레이브의 Read Multiple Registers

- UNIT ID
- FC

  Read Multiple Registers 이므로 0x03 입니다.
- BYTE COUNT

  BYTE COUNT는 WORD COUNTER×2 로 ezTCP의 기본 값은 0x02 입니다.
- DATA (Register Values)

  실제 입력포트의 상태를 나타냅니다. 해당 포트를 나타내는 비트의 값이 0일 경우 OFF를, 1일 경우 ON을 나타냅니다.

표 2-3 비트 값과 그 의미

값	설명
0	입력 OFF
1	입력 ON

CIE-M10/H10의 경우 입력포트가 8포트 이며 다음 그림에서와 같이 최 하위 비트가 0번 입력포트를 나타냅니다. EZI-10은 1포트 이므로 최 하위 비트만 사용됩니다.

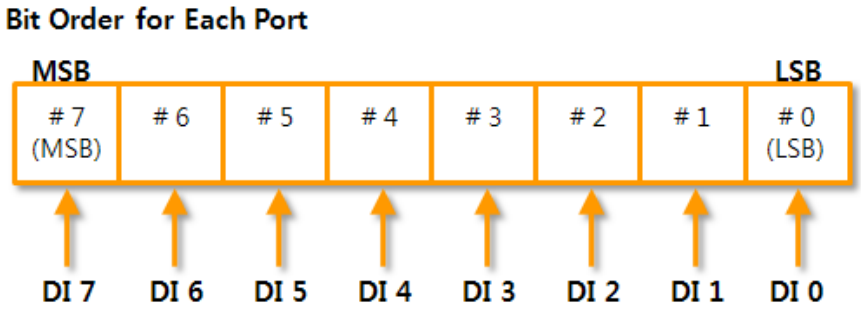


그림 2-5 비트 순서와 포트 지정

● 전송 예

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Value(HEX)	00	00	00	00	00	05	01	03	02	00	FF

그림 2-6 전송 예

표 2-4 전송 예에 대한 설명

바이트 순서	값(HEX)	의미
0~1	0x0000	트랜잭션 ID를 0x0000으로 설정
2~3	0x0000	프로토콜 ID (고정 값)
4~5	0x0005	이후 5 Bytes 까지가 한 프레임임을 나타냄 (고정 값)
6	0x01	유니트 ID가 1로 설정 됨
7	0x03	Read Multiple Registers 함수임을 나타냄 (고정 값)
8	0x02	Byte Count = Word Count * 2 (고정 값)
9~10	0x00FF	0번부터 7번까지 8개의 입력포트가 ON인 상태 (비트 표현: 0000 0000 1111 1111)

2.5.3 Command Exception of Read Multiple Registers

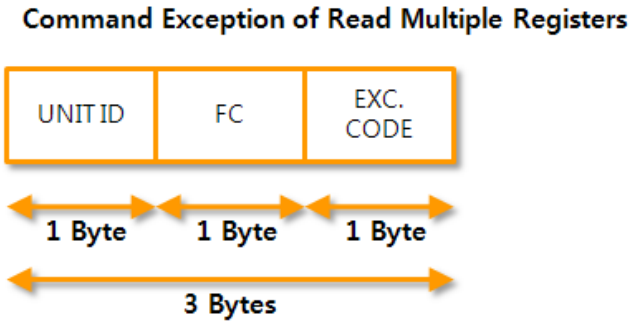


그림 2-7 Command Exception of Read Multiple Registers

● UNIT ID

● FC

0x83 입니다.

● EXCEPTION CODE

0x01 또는 02

표 2-5 exception code

값	설명
0x01	잘못된 FC이거나 슬레이브가 응답할 수 없는 경우
0x02	Reference Number 값이 잘 못 된 경우

2.5.4 Command Request of Write Multiple Registers

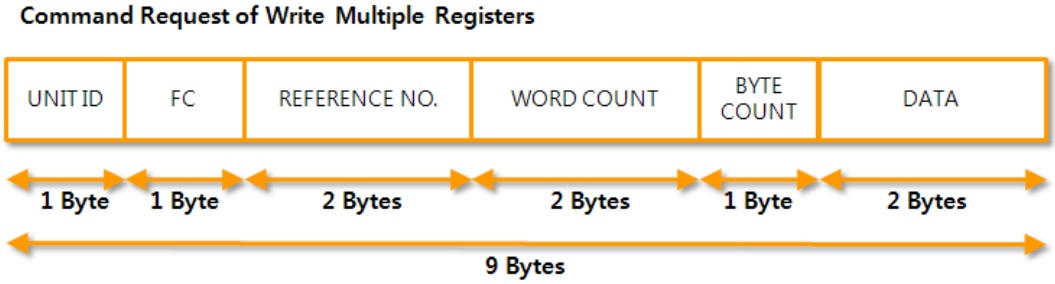


그림 2-8 마스터의 Write Multiple Registers

- UNIT ID
- FC(Function Code): 함수 코드
  Write Multiple Registers의 함수 코드는 0x10(HEX) 입니다.
- REFERENCE NO. (Reference Number)
  출력포트의 시작주소로 설정 된 값을 나타냅니다. (기본 값: 0x0008)
- WORD COUNT
- BYTE COUNT
- DATA (Register Values)
  값을 ON 하기 위한 출력포트를 나타냅니다. 해당 포트를 나타내는 비트의 값이 0일 경우 OFF를, 1일 경우 ON을 나타냅니다.
  입력포트의 데이터와 마찬가지로, CIE-M10/H10의 경우 출력포트가 8포트 이며 다음 그림에서와 같이 최 하위 비트가 0번 출력포트를 나타냅니다. EZI-10은 1포트 이므로 최 하위 비트만 사용됩니다.

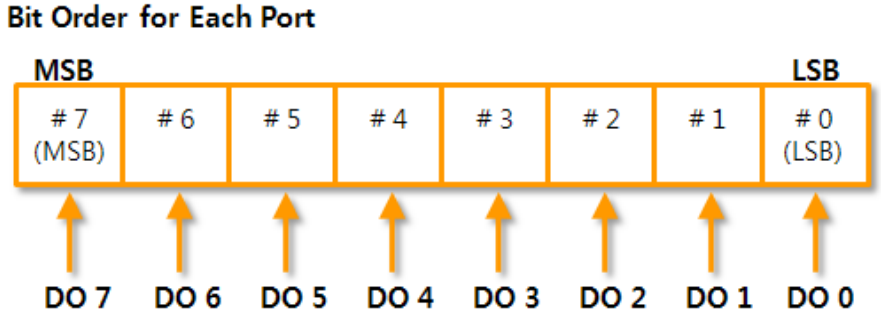


그림 2-9 비트 순서와 포트 지정

● 전송 예

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13	#14
Value(HEX)	00	00	00	00	00	09	01	10	00	08	00	01	02	00	11

그림 2-10 전송 예

표 2-6 전송 예에 대한 설명

바이트 순서	값(HEX)	의미
0~1	0x0000	트랜잭션 ID를 0x0000으로 설정
2~3	0x0000	프로토콜 ID (고정 값)
4~5	0x0005	이후 9 Bytes 까지가 한 프레임임을 나타냄 (고정 값)
6	0x01	유니트 ID가 1로 설정 됨
7	0x10	Write Multiple Registers 함수임을 나타냄 (고정 값)
8~9	0x0008	출력포트 시작주소가 8로 설정 됨
10~11	0x0001	Word Count 기본 값 (고정 값)
12	0x02	Byte Count = Word Count * 2 (고정 값)
13~14	0x0011	0번과 4번 출력포트에 ON을 출력 (비트 표현: 0000 0000 0001 0001)

2.5.5 Command Response of Write Multiple Registers

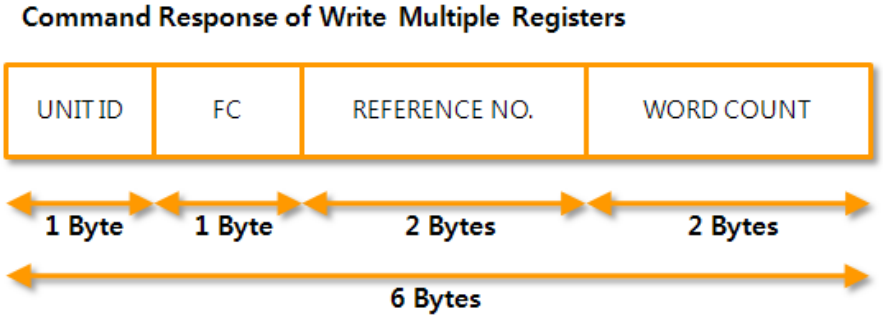


그림 2-11 Command Response of Write Multiple Registers

● UNIT ID

● FC

Write Multiple Registers 이므로 0x10 입니다.

● REFERENCE NO. (Reference Number)

출력포트의 시작주소로 설정 된 값을 나타냅니다. (기본 값: 0x0008)

● WORD COUNT

● 전송 예

Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	10	00	08	00	01

그림 2-12 전송 예

표 2-7 전송 예에 대한 설명

바이트 순서	값(HEX)	의미
0~1	0x0000	트랜잭션 ID를 0x0000으로 설정
2~3	0x0000	프로토콜 ID (고정 값)
4~5	0x0006	이후 6 Bytes 까지가 한 프레임임을 나타냄 (고정 값)
6	0x01	유니트 ID가 1로 설정 됨
7	0x10	Write Multiple Registers 함수임을 나타냄 (고정 값)
8~9	0x0008	출력포트 시작주소가 8로 설정 됨
10~11	0x0001	Word Count 기본 값 (고정 값)

2.5.6 Command Exception of Write Multiple Registers

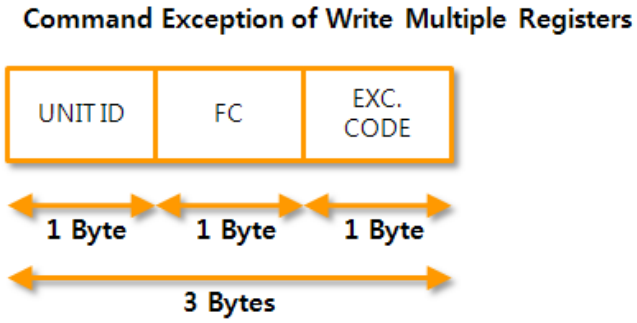


그림 2-13 Command Exception of Write Multiple Registers

● UNIT ID

● FC

0x90 입니다.

● EXCEPTION CODE

0x01 또는 02

표 2-8 exception code

값	설명
0x01	잘못된 FC이거나 슬레이브가 응답할 수 없는 경우
0x02	Reference Number 값이 잘 못 된 경우

2.6 사용하기

2.6.1 모드버스 TCP 설정

- CIE-M10/H10

  ezManager를 통해 제품을 검색 하고 ③번에 표시된 부분을 설정합니다.

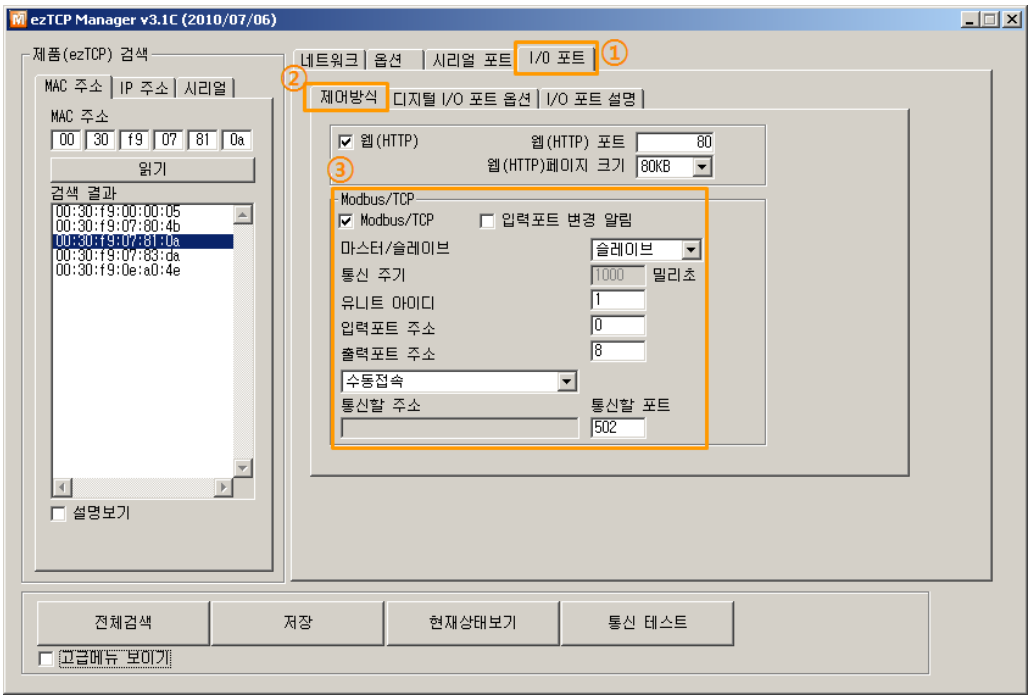


그림 2-14 ezManager 모드버스TCP 설정

- EZI-10

  ezConfigIO를 통해 제품을 검색 하고 표시된 부분을 설정합니다.

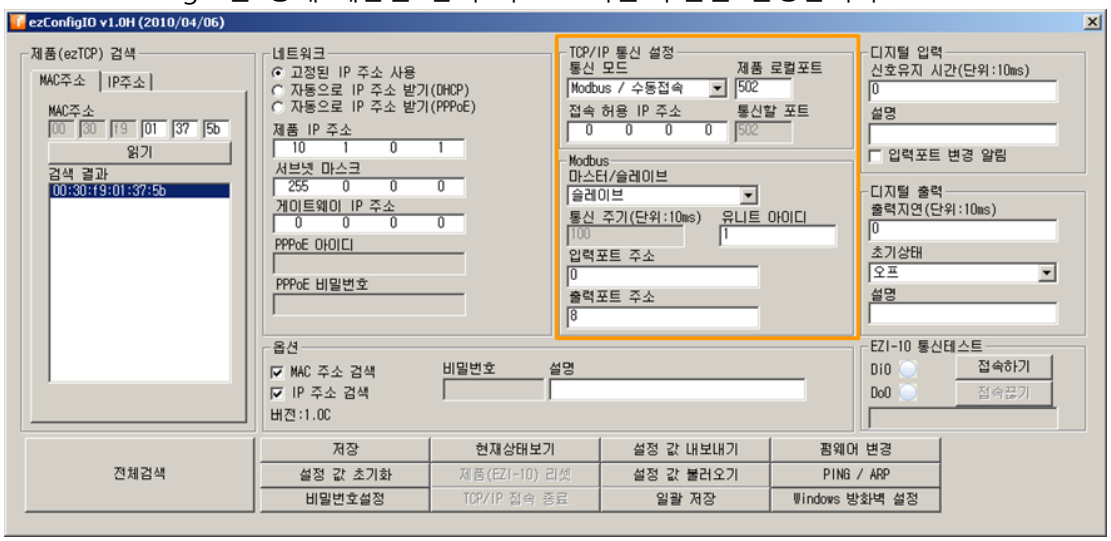


그림 2-15 ezConfigIO 모드버스 TCP 설정

● 설정 항목

표 2-9 Modbus/TCP 설정 항목

항목	설명
Modbus/TCP	해당 프로토콜 사용 여부
입력포트 변경 알림	슬레이브가 마스터의 쿼리 없이도 입력포트 변경 시 바로 알려주는 기능
마스터 / 슬레이브	모드버스TCP 동작 유형
통신 주기	마스터가 쿼리를 보내는 주기 (단위: 밀리 초)
유니트 아이디	마스터와 슬레이브의 아이디
입력포트 주소	입력포트 시작주소
출력포트 주소	출력포트 시작주소
수동/능동 접속	능동: 접속 시도, 수동: 접속 대기
통신할 주소	능동접속일 때 접속 할 주소
통신할 포트	통신할 포트 번호

2.6.2 설정 예

표 2-10 설정 예

항목	ezTCP	또 다른 ezTCP 또는 Modbus/TCP 프로그램
로컬 IP주소	192.168.0.10	192.168.0.20
서브넷 마스크	255.255.255.0	255.255.255.0
Modbus/TCP	체크 또는 선택	-
마스터 / 슬레이브	슬레이브	마스터
통신 주기	-	1,000ms (1초)
유니트 아이디	1	1
입력포트 주소	0	0
출력포트 주소	8	8
수동/능동 접속	수동접속	능동접속
통신할 주소	-	192.168.0.10
통신할 포트	-	502
제품 로컬포트	502	-

2.7 샘플 코드

당사에서는 ezTCP 디지털 입/출력 컨트롤러 사용자를 위해 모드버스TCP 샘플코드를 제공하고 있습니다. 프로그램 구현에 활용하시기 바랍니다.

☞ ***샘플코드는 당사 홈페이지 [고객지원]>>[기술자료]>>[샘플코드] 메뉴의 "Visual C++ CIE-H10", "Visual Basic CIE-H10" 에서 다운로드 할 수 있습니다.***

- CModBusEngine
  모드버스TCP를 구현한 클래스(Class) 입니다.

2.7.1 주요 함수

- SendReadRequest
  ezTCP의 입/출력 포트의 값을 읽는 함수 입니다.

표 2-11 SendReadRequest 함수의 파라미터

파라미터	설명
Transaction_id	트랜잭션 아이디
Unit_id	제품의 유니트 아이디
address	제품의 입/출력 포트 주소

- SendWriteRequest
  ezTCP의 출력 포트로 출력을 내보내는 함수 입니다.

표 2-12 SendWriteRequest 함수의 파라미터

파라미터	설명
Transaction_id	트랜잭션 아이디
Unit_id	제품의 유니트 아이디
address	제품의 입/출력 포트 주소
value	출력 포트 출력 값

- OnReceive
  모드버스TCP 응답 패킷을 처리하는 함수 입니다.

3 시리얼 모드버스 TCP

시리얼모드버스TCP는 CIE-M10/H10만 지원합니다. RS232 포트를 이용해 I/O를 감시/제어하며 이 모드를 사용할 때 다른 통신방식은 모두 사용할 수 없습니다.

3.1 특징

- 기존 모드버스TCP 데이터를 RS232로 송/수신
- 직렬포트를 이용한 디지털 입/출력 컨트롤
- 접속과정이 없고 단순히 데이터를 송/수신 함
  케이블 등의 접촉상태에 따라 데이터가 유실 될 수 있으므로, 이를 방지하고자 하는 경우에는 하드웨어 흐름제어(RTS/CTS)를 사용하시기 바랍니다.

3.2 사용하기

3.2.1 설정 방법

- 시리얼모드버스TCP 모드 설정

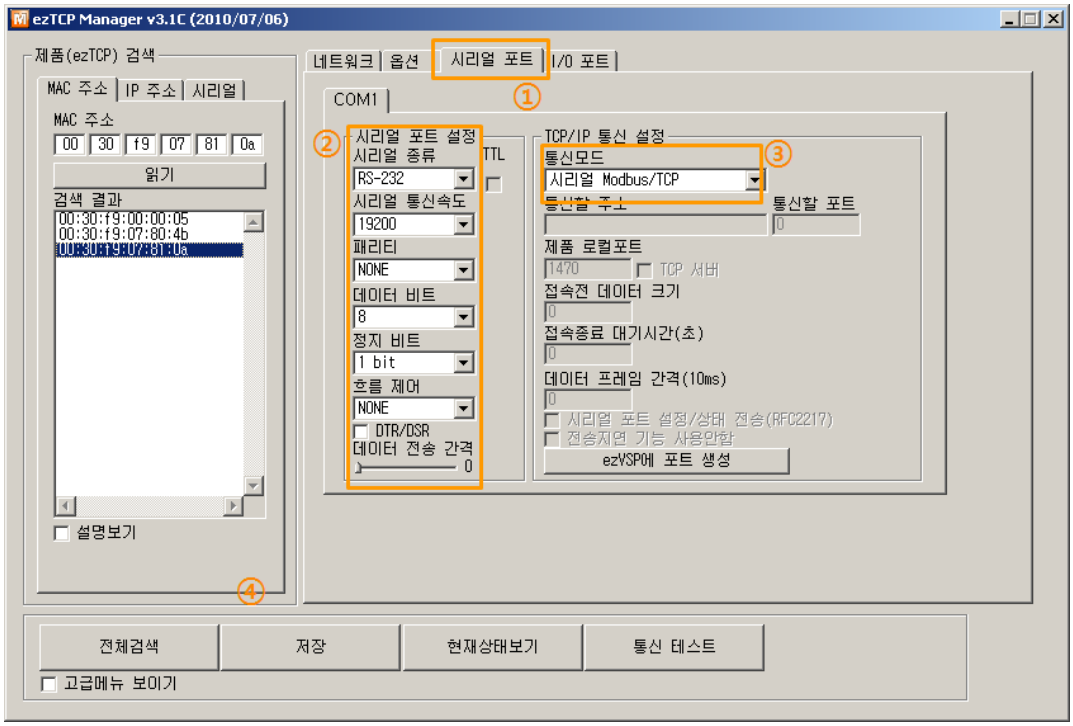


그림 3-1 시리얼모드버스TCP 모드 설정

① [시리얼 포트] 탭으로 이동
② 시리얼 포트 항목 설정
③ [TCP/IP 통신설정]에서 통신모드를 [시리얼 Modbus/TCP]로 선택
④ [저장]버튼으로 환경 값 저장

3.3 시험 작동

3.3.1 통신 준비

시리얼모드버스TCP 동작을 시험하기 위해 다음처럼 구성해 주시기 바랍니다.

☞ ***랜 케이블은 연결하지 않아도 무방합니다.***

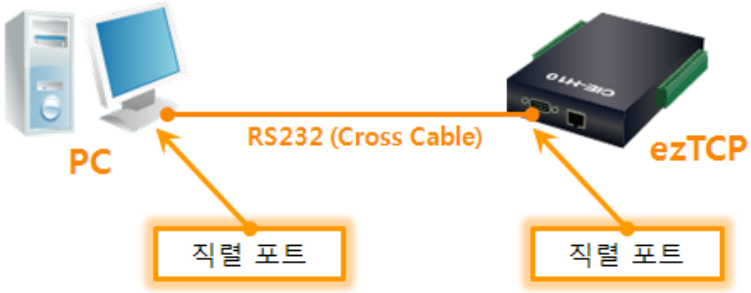


그림 3-2 통신 준비

시험을 위해 모드버스TCP 설정은 다음과 같이 기본 값을 유지하시기 바랍니다.

표 3-1 모드버스TCP 설정 기본 값

항목	기본 값
Modbus/TCP	체크
입력포트 변경 알림	체크 안 됨
마스터/슬레이브	슬레이브
통신 주기	1,000
유니트 아이디	1
입력포트 주소	0
출력포트 주소	8

3.3.2 시험 데이터 전송

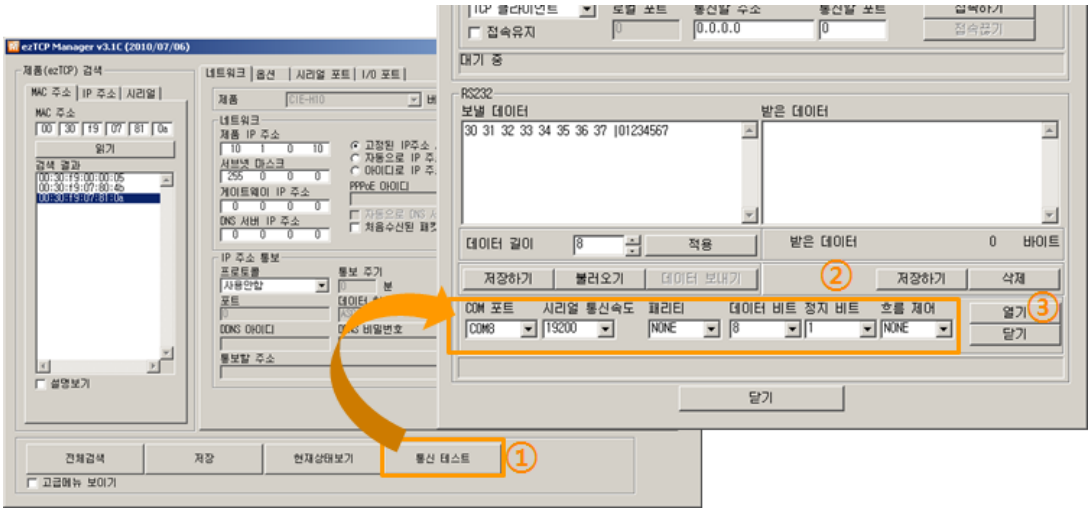


그림 3-3 시험 데이터 전송 1

① ezManager의 [통신 테스트]버튼 클릭

② ezTCP와 연결 된 PC의 COM포트를 선택하고 시리얼 포트 설정 값 확인

③ [열기]버튼 클릭하여 포트 열기

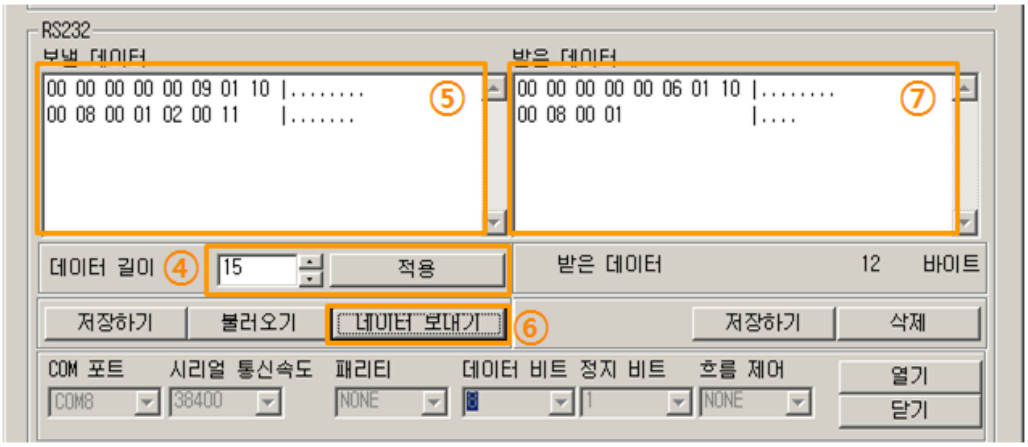


그림 3-4 시험 데이터 전송 2

④ 데이터 길이를 15(Bytes)로 설정하고 [적용]버튼 클릭

⑤ [보낼 데이터]에 그림 2-10의 데이터를 입력

⑥ [데이터 보내기]버튼 클릭

⑦ [받은 데이터]에 나타난 ezTCP의 응답 데이터가 그림 2-12와 같은지 확인

☞ ***⑤ 에서 보낸 데이터는 슬레이브의 0번, 4번 출력포트를 ON 하라는 마스터의 명령입니다(Write Multiple Registers). 따라서 슬레이브는 그에 대한 응답으로 ⑦번에 나타난 데이터를 보내야 합니다.***

4 문서 변경 이력

날짜	버전	변경내용	작성자
2010.03.08	1.0	○ 최초 작성	이인
2010.07.20	1.1	○ 문서 이름 변경 ○ Modbus/TCP 문서 통합 ○ EZI-10 내용 추가	이인