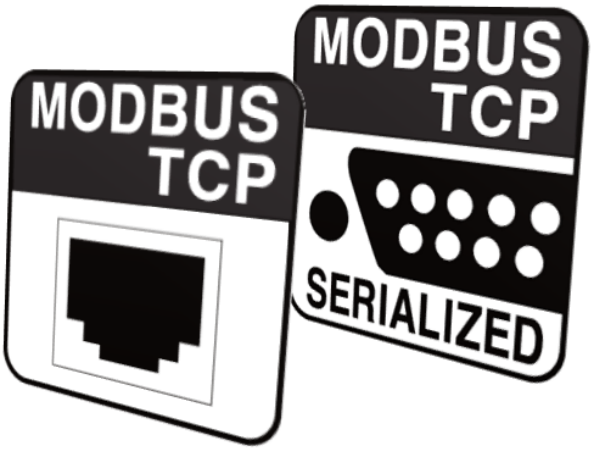
Application Note

ezTCP 의 Modbus/TCP

Version 1.7



솔내시스템㈜

https://www.sollae.co.kr/

1 목차

2 개요

2.1 개요

MODBUS는 PLC(Programmable Logic Controller) 대표되는 각종 산업용 자동화 장비들의 감시, 제어에 널리 사용되고 있는 통신 프로토콜입니다. 최초에는 시리얼 통신용으로 개발되었지만 네트워크 통신 환경에서의 적용 필요성으로 TCP/IP 통신용 버전까지 확장되었습니다. 시리얼 버전(이하 Modubs RTU/ASCII)과 TCP/IP 버전(이하 Modbus/TCP)의 프로토콜은 유사하지만 동일하지 않으므로 구분할 필요가 있으며 솔내시스템 I/O 게이트웨이는 Modbus/TCP 프로토콜을 지원합니다(클라우드 전용 제품 제외).

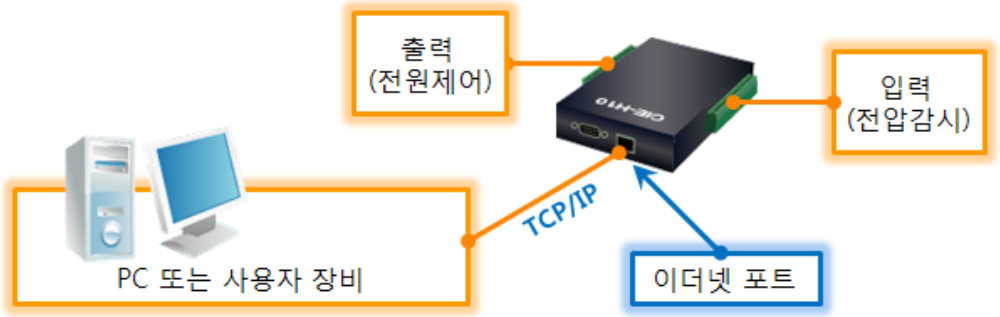


그림 2-1 Modbus/TCP 사용 구성 예

☞ ***이 문서는 솔내시스템 I/O 게이트웨이에 적용된 Modbus/TCP 프로토콜 안내가 주 목적입니다. Modbus RTU/ASCII 프로토콜은 관련 표준 문서를 참고하시기 바랍니다.***

일부 제품은 시리얼 포트를 통한 장비 감시, 제어를 위해 "시리얼 Modbus/TCP" 모드를 지원합니다(해당 제품: CIE 제품군). "시리얼 Modbus/TCP"는 표준 MODBUS를 의미하는 것이 아니며 앞서 언급한 Modbus/TCP 프로토콜 데이터를 그대로 사용하여 직렬통신 (RS232) 방식으로 송/수신하는 모드입니다.

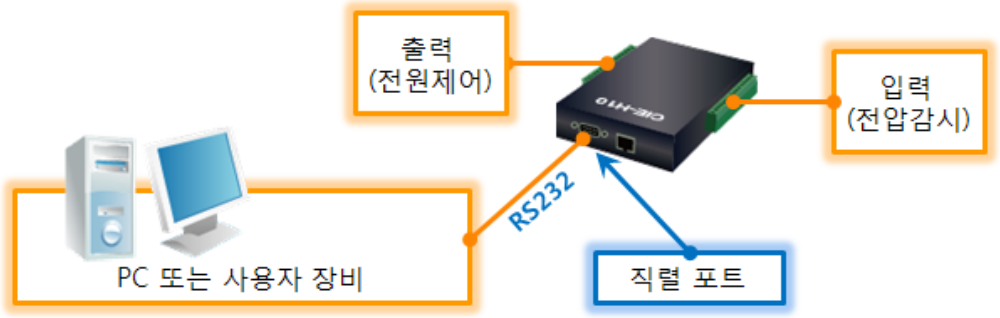


그림 2-2 시리얼 Modbus/TCP 사용 구성 예

2.2 적용 제품

- SIG 제품군 – SIG-5430, SIG-5440, SIG-5450, SIG-5600
- CIE 제품군 – CIE-H12A, CIE-H14A, CIE-H10A, CIE-M10A 등
- EZI-10

2.3 용어

2.3.1 포트 종류

다음 3가지 종류의 포트가 존재합니다.

- 디지털 입력포트
- 아날로그 입력포트
- 디지털 출력포트

2.3.2 포트 번호

제품 입력/출력 포트 각각 첫 번째 포트는 숫자 0부터 시작하고 이후 1씩 증가하며 '#' 기호와 함께 표시합니다. 예를 들어 CIE-H10A는 디지털 입력포트와 디지털 출력포트 각각 8개가 있으며 첫 번째 포트는 "#0"으로 두 번째 포트는 "#1"로 마지막으로 8번째 포트는 '#7'로 표시합니다.

2.3.3 MSB / LSB

숫자 데이터를 비트 단위로 나타냈을 때 비트의 위치에 따라 구분하는 방법입니다.

그림 2-3 MSB와 LSB

- MSB(Most Significant Bit) – 가장 큰 값에 해당하는 비트, 가장 왼쪽 비트
- LSB(Least Significant Bit) – 가장 작은 값에 해당하는 비트, 가장 오른쪽 비트

3 프로토콜 개요

3.1 MODBUS

3.1.1 특징

- 응용 계층 프로토콜 (OSI 7 계층)
- PDU (Protocol Data Unit)
  하위 계층과 관계없이 독립적 데이터 구조

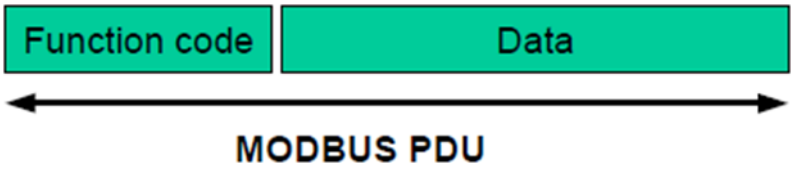


그림 3-1 MODBUS PDU

- ADU (Application Data Unit)
  하위 계층에 따라 최종 결정되는 데이터 구조

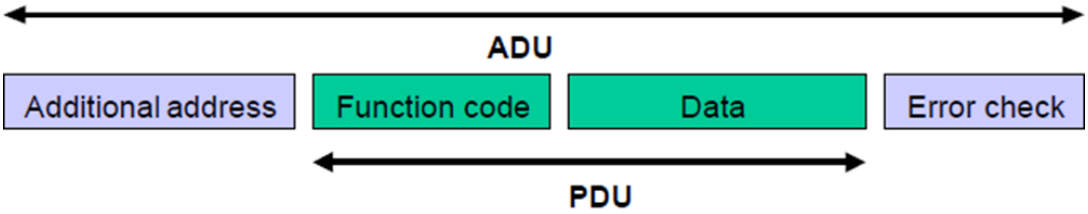


그림 3-2 일반적인 MODBUS 프레임 구조

- 클라이언트/서버 모델

그림 3-3 MODBUS 클라이언트/서버 모델

3.1.2 데이터 인코딩 (Data Encoding)

MODBUS는 빅 엔디언(Big-endian) 방식을 사용합니다. 빅 엔디언이란, 한 바이트 이상의 메모리를 차지하는 값이 전송될 때 큰 값이 먼저 전송되는 방식을 의미합니다. 예를 들어 16비트 데이터 0x1234는 0x12, 0x34의 순서로 전송됩니다.

3.1.3 MODBUS 데이터 모델

MODBUS 데이터는 그 유형과 읽기/쓰기 가능 여부 기준으로 나눠 총 4가지로 분류되며 분류별로 각각 1에서 65,536까지의 번호가 매겨집니다.

데이터 분류	데이터 유형	읽기/쓰기	설명
Discrete Inputs	비트	읽기	장비 포트 상태
Coils	비트	읽기/쓰기	응용 프로그램에 의해 변경 가능
Input Registers	워드 (16비트)	읽기	장비 포트 상태
Holding Registers	워드 (16비트)	읽기/쓰기	응용 프로그램에 의해 변경 가능

표 3-1 Modbus 데이터 모델

3.1.4 I/O 게이트웨이 메모리 구조

MODBUS 장비의 메모리 구조는 MODBUS 프로토콜의 범위를 벗어나는 부분이므로 표준에 정의되어 있지 않습니다. 따라서 제조사의 구현 방식과 정책에 따라 그 구조가 달라질 수밖에 없습니다. 크게 보면 데이터 종류별로 블록을 각각 지정하여 4개의 데이터 블록을 사용하는 방식과 모든 데이터를 하나의 데이터 블록에 연결시켜 사용하는 방식 2가지를 들 수 있습니다. 아래 그림은 하나의 데이터 블록만 사용하는 I/O 게이트웨이 메모리 구조를 간략히 나타냅니다.

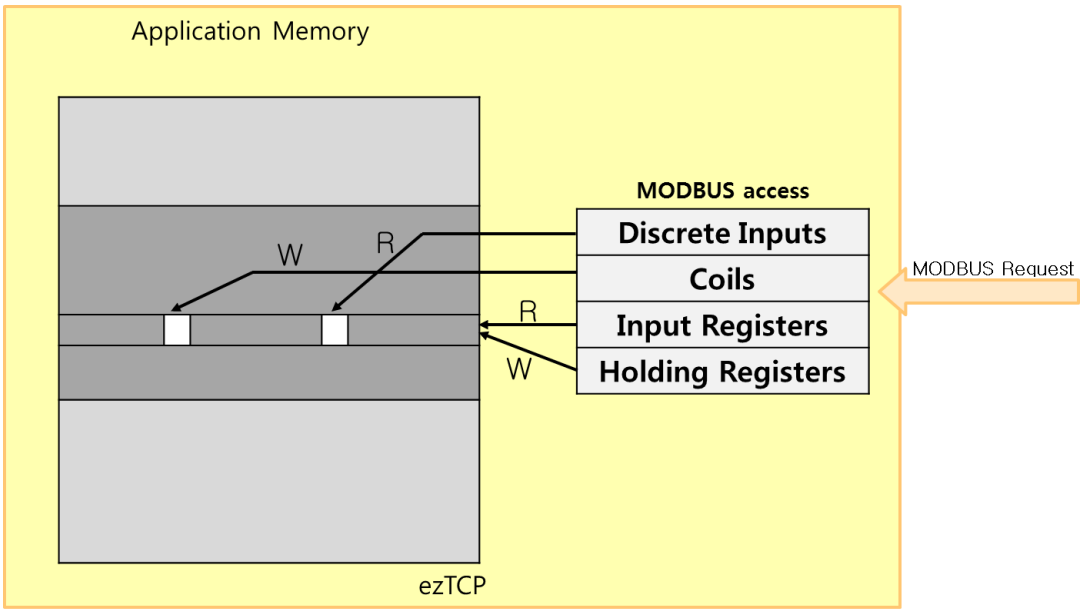


그림 3-4 하나의 데이터 블록을 사용하는 I/O 게이트웨이의 메모리 구조

3.1.5 MODBUS 주소 사용

MODBUS 표준은 16비트 체계를 적용해 0에서 65,535까지(0x0000 ~ 0xFFFF)의 값을 주소로 사용하도록 정의하고 있지만 이 주소를 MODBUS 데이터와 어떻게 연결시킬지에 대한 정의는 없습니다. 따라서 메모리 구조와 마찬가지로 데이터와 메모리 주소간 매핑 방식은 장비 제조사마다 다르며 이번 절은 I/O 게이트웨이와 통신을 통해 MODBUS 데이터를 가져오기 위한 주소 사용에 대한 안내를 목적으로 합니다.

장비와 통신을 통해 MODBUS 데이터를 가져와 사용자에게 제공하는 클라이언트(이하 HMI – Human Machine Interface)는 번호는 같지만 분류가 다른 데이터를 구별할 목적으로 접두사를 사용합니다. 다음 표는 MODBUS 데이터 분류에 따른 HMI에서 사용되는 접두사, MODBUS 표준에 정의된 사용할 수 있는 함수(실제 제공되는 서비스), 그리고 I/O 게이트웨이의 포트 정보를 나타냅니다.

데이터 분류	구분 접두사	함수	I/O 게이트웨이 포트
Coils	0X reference	01, 05, 15	출력 포트 (디지털)
Discrete Inputs	1X reference	02	입력 포트 (디지털)
Input Registers	3X reference	04	입력 포트 (디지털/아날로그)
Holding Registers	4X reference	03, 06, 16	입력/출력 포트 (디지털/아날로그)

표 3-2 Modbus 데이터 주소 및 I/O 게이트웨이의 구조

일반적으로 HMI는 사용자 인터페이스에서 MODBUS 주소를 바로 사용하지 않고 구분 접두사와 MODBUS 데이터 번호를 조합한 값을 사용합니다. 이는 사용자 입장에서 해당 데이터 분류를 가져올 때 필요한 함수 코드를 몰라도 통신 설정이 가능하도록 해주며 단순히 주소만으로도 해당 데이터의 종류를 알 수 있게 해줍니다. MODBUS 데이터 번호는 1부터 시작(주소는 0부터 시작)되며 구분 접두사까지 사용되므로 HMI에서 주소와 I/O 게이트웨이 장비에서 주소는 같지 않을 수 있습니다. 다음의 예들은 I/O 게이트웨이 MODBUS 데이터에 접근하기 위한 HMI에서 주소를 장비 주소 및 포트 종류 기준으로 나타냅니다.

- (예) 디지털 입력포트 (장비 주소가 '0'일 때)

장비 주소	종류	함수	HMI에서 주소
0	Coils	01, 05, 15	사용 불가
	Discrete Inputs	02	10001
	Input Registers	04	30001
	Holding Registers	03	40001

표 3-3 디지털 입력 포트 주소 표시 예

- (예) 디지털 출력포트 (장비 주소가 '8'일 때)

장비 주소	종류	함수	HMI에서 주소
8	Coils	01, 05, 15	00009
	Discrete Inputs	02	사용 불가
	Input Registers	04	사용 불가
	Holding Registers	03, 06, 16	40009

표 3-4 디지털 출력 포트 주소 표시 예

- (예) 아날로그 입력포트 (장비 주소가 '4'일 때)

장비 주소	종류	함수	HMI에서 주소
4	Coils	01, 05, 15	사용 불가
	Discrete Inputs	02	사용 불가
	Input Registers	04	30004
	Holding Registers	03	40004

표 3-5 아날로그 입력 포트 주소 표시 예

☞ ***일부 HMI에서는 MODBUS 주소와 데이터간 시작 번호 차이에 따른 혼란을 없애기 위해 MODBUS 데이터 시작 번호를 0으로 설정하는 옵션을 지원하니 참고하시기 바랍니다.***

3.2 Modbus/TCP

3.2.1 특징

- MODBUS 프로토콜의 TCP/IP용 버전
- TCP 접속 과정이 선행됨
  이름에서 알 수 있듯이 Modbus/TCP는 TCP를 사용합니다. 기본 포트번호는 TCP 502번 입니다.

3.2.2 통신 모델

표준 Modbus/TCP는 클라이언트/서버 모델을 정의합니다. 클라이언트가 요청(Request)을 보내면 서버가 그에 대한 응답(Response)을 보내는 구조입니다. Modbus/TCP 클라이언트는 동시에 TCP 클라이언트이고 마찬가지로 Modbus/TCP 서버는 TCP 서버 역할을 수행합니다.

하지만 I/O 게이트웨이는 클라이언트/서버 모델이 아닌 마스터/슬레이브 형태로 동작합니다. TCP 접속의 시작은 마스터/슬레이브 여부와 독립적으로 사용자 설정에 의해 결정됩니다.

- 마스터
  TCP 접속의 시작 여부를 제외하고는 표준에서 클라이언트를 의미합니다.
- 슬레이브
  TCP 접속의 시작 여부를 제외하고는 표준에서 서버를 의미합니다.

3.2.3 Modbus/TCP 프레임

아래의 그림은 Modbus/TCP 프레임 구조를 나타냅니다.

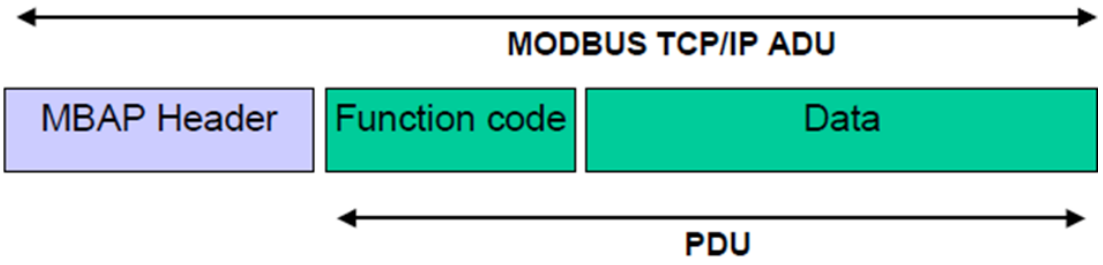


그림 3-5 Modbus/TCP 프레임 구조

3.2.4 MBAP 헤더

MBAP는 Modbus Application Protocol의 약자로 헤더는 다음과 같이 4가지 항목으로 이루어져 있습니다.

항목	길이	설명
Transaction Identifier	2 bytes	요청/응답을 한 쌍의 작업으로 구분
Protocol Identifier	2 bytes	0 = MODBUS protocol
Length	2 bytes	프레임의 남은 길이
Unit Identifier	1 byte	TCP/IP가 아닌 다른 통신 선로를(예: 시리얼) 통해 연결되어 있는 슬레이브 구분

표 3-6 MBAP 헤더

전체 Modbus/TCP 프레임은 다음 구조로 되어 있습니다.

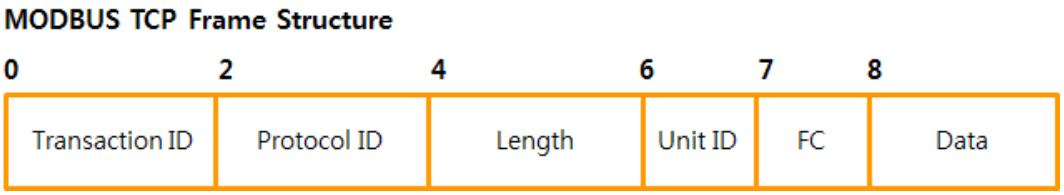


그림 3-6 Modbus/TCP 프레임 구조

- byte 0 ~ 1: 트랜잭션 아이디 (Transaction Identifier)
  요청 및 응답을 한 쌍의 작업으로 구분하기 위해 사용되는 번호이며 마스터에 의해 설정됩니다. I/O 게이트웨이 마스터 동작모드는 매 명령마다 1씩 증가시키며 값을 사용하며 슬레이브는 마스터의 요청에 있는 값을 그대로 복사해 사용합니다.

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

- byte 2 ~ 3: 프로토콜 아이디 (Protocol Identifier)
  프로토콜의 ID를 나타내며 0x0000으로 고정 값입니다.
- byte 4 ~ 5: 길이 (Length)
  Length 필드 이후부터 해당 프레임의 마지막까지의 길이를 나타냅니다. (단위: Byte)
- byte 6: 유니트 아이디 (Unit Identifier)
- byte 7: 함수 코드 (Function Code)
- byte 8 ~: 함수 코드에 따른 데이터 등 (Data)

☞ ***통신 선로로 시리얼을 사용하는 Modbus와 호환을 위해 Modbus/TCP 프레임 하나의 최대 크기는 260 바이트로 제한됩니다.***

3.3 함수 (Function Codes)

3.3.1 함수 종류

함수는 Modbus 프로토콜에서 실제 제공되는 서비스를 정의합니다. 함수는 Modbus 프레임에서 1 바이트의 공간을 차지하고 있으므로 사용 가능한 영역은 1 ~ 255입니다. 이중 실제로는 1 ~ 127 사이의 값을 사용하며 128 ~ 255의 값은 에러가 발생할 때 사용되는 예외 응답을 위해 사용됩니다. 함수 코드 0은 사용할 수 없습니다. 일부 함수는 여러 동작 지원을 위해 서브 함수 코드가 추가로 사용될 수 있습니다.

함수 종류는 그 목적에 따라 크게 3가지로 분류됩니다.

- Public Function Codes
  표준 문서에 정의되어 있는 함수입니다.
  1 ~ 64, 73 ~ 99, 111 ~ 127
- User-Defined Function Codes
  표준 문서에는 정의되어 있지 않고 장비 제조사에서 직접 구현한 기능과 관련된 함수입니다.
  65 ~ 72, 100 ~ 110
- Reserved Function Codes
  Public 영역 중 일부 제조사의 구형 장비에 사용되는 함수로 공식적으로 사용이 불가능한 함수입니다.
  8/19, 8/21~65535, 9, 10, 13, 14, 41, 42, 90, 91, 125, 126, 127

Number	Function Codes
111 ~ 127	Public
100 ~ 110	User-Defined
73 ~ 99	Public
65 ~ 72	User-Defined
1 ~ 64	Public

표 3-7 함수 코드 종류

3.3.2 Public Function Codes

다음은 솔내시스템 I/O 게이트웨이가 지원하는 Public 함수 코드입니다.

종류	접근	포트	이름	함수코드	
				코드	서브 코드
데이터	비트	입력포트 (디지털)	Read Discrete Inputs	02 (0x02)	
		출력포트 (디지털)	Read Coils	01 (0x01)	
			Write Single Coil	05 (0x05)	
			Write Multiple Coils	15 (0x0F)	
	16비트	입력포트	Read Input Registers	04 (0x04)	
		입력포트 출력포트	Read Holding Registers	03 (0x03)	
		출력포트	Write Single Register	06 (0x06)	
		출력포트	Write Multiple Registers	16 (0x10)	
진단			Read Exception Status	07 (0x07)	
			Read Device Identification	43 (0x2B)	14 (0x0E)
기타			Encapsulated Interface Transport	43 (0x2B)	

표 3-8 I/O 게이트웨이가 지원하는 Public 함수 코드

☞ ***제품 종류와 펌웨어 버전에 따라 지원하는 함수 코드는 다를 수 있습니다.***

3.3.3 User-Defined Function Codes

솔내시스템 I/O 게이트웨이가 지원하는 사용자 정의 함수는 다음과 같습니다.

종류	접근	포트	이름	함수코드	
				코드	서브 코드
데이터	비트	출력포트 (디지털)	Write Pulse	105 (0x69)	
데이터	16비트	입력포트 출력포트	Send Notify	108 (0x6C)	

표 3-9 I/O 게이트웨이가 지원하는 사용자 정의 함수

4 Public 함수

4.1 Read Coils (FC 01)

디지털 출력포트 상태 확인에 사용됩니다.

4.1.1 요청

Request of Read Coils

그림 4-1 Request of Read Coils

- byte 0: 함수 코드
  Read Coils의 함수 코드는 0x01 입니다.
- byte 1~2: 시작 주소
  상태 값을 읽을 첫 번째 디지털 출력포트의 주소입니다.
- byte 3~4: 출력포트 개수
  읽을 디지털 출력포트 수를 지정합니다. 사용 가능한 값의 범위는 1 ~ n까지입니다.

☞ ***n: 각 제품의 디지털 출력포트 개수***

4.1.2 응답

Response of Read Coils

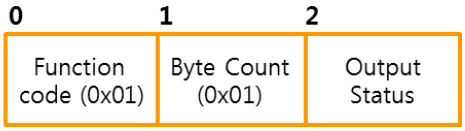


그림 4-2 Response of Read Coils

- byte 0: 함수 코드 (0x01)
- byte 1: 바이트 카운트 (0x01)
  (출력포트 개수+7) / 8
- byte 2: 출력포트 상태
  디지털 출력포트 상태를 나타냅니다. 포트 개수에 따라 바이트 단위로 추가되며 1바이트에 최대 8개의 포트를 비트 단위로 나타냅니다. 시작 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 표시되고 비트 값 0은 OFF를 1은 ON을 의미하며 제품에 없는 포트에 해당되는 비트는 0으로 채워집니다.

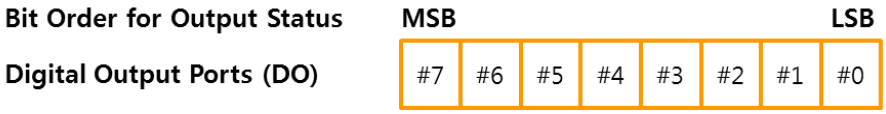


그림 4-3 디지털 출력포트 상태

4.1.3 예외

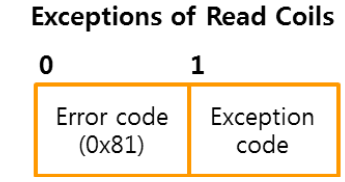


그림 4-4 Exception of Read Coils

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x81 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02 또는 0x03 입니다.

4.1.4 사용 예

다음은 시작 주소가 "8"일 때 디지털 출력포트 #0 ~ #7을 읽는 사용 예입니다.

● 요청

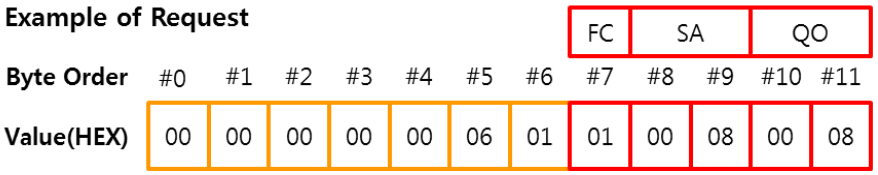


그림 4-5 요청 예

바이트 순서	값(HEX)	의미
7	0x01	함수 코드 01
8~9	0x0008	읽어올 주소 8로 설정
10~11	0x0008	시작 주소부터 8개의 디지털 출력포트를 읽음

표 4-1 요청 예

● 응답

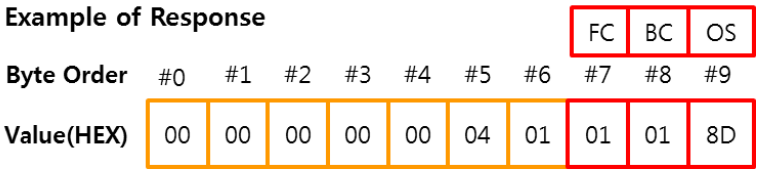


그림 4-6 응답 예

바이트 순서	값(HEX)	의미
7	0x01	함수 코드 01
8	0x01	1바이트, 1 ~ 8개 사이의 출력포트
9	0x8D	(1000 1101) #0, 2, 3, 7 ON / #1, 4 ~ 6 OFF

표 4-2 응답 예

4.2 Read Discrete Inputs (FC 02)

디지털 입력포트 상태 확인에 사용됩니다.

4.2.1 요청

Request of Read Discrete Inputs

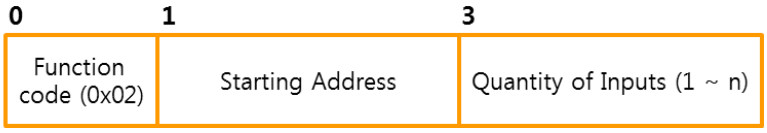


그림 4-7 Request of Read Discrete Inputs

- byte 0: 함수 코드
  Read Discrete Inputs의 함수 코드는 0x02 입니다.
- byte 1~2: 시작 주소
  상태 값을 읽을 첫 번째 디지털 입력포트의 주소입니다.
- byte 3~4: 입력포트 개수
  읽을 입력포트 수를 지정합니다. 사용 가능한 값의 범위는 1 ~ n까지입니다.

☞ ***n: 각 제품의 디지털 입력포트 개수***

4.2.2 응답

Response of Read Discrete Inputs

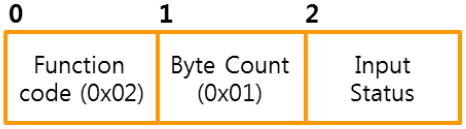


그림 4-8 Response of Read Discrete Inputs

- byte 0: 함수 코드 (0x02)
- byte 1: 바이트 카운트 (0x01)
  (입력포트 개수+7)/8
- byte 2: 입력포트 상태
  디지털 입력포트 상태를 나타냅니다. 포트 개수에 따라 바이트 단위로 추가되며 1바이트에 최대 8개의 포트를 비트 단위로 나타냅니다. 시작 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 표시되고 비트 값 0은 OFF를 1은 ON을 의미하며 제품에 없는 포트에 해당되는 비트는 0으로 채워집니다.

그림 4-9 디지털 입력포트 상태

4.2.3 예외

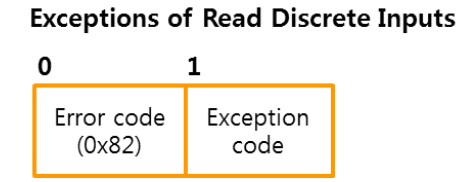


그림 4-10 Exception of Read Discrete Inputs

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x82 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02 또는 0x03 입니다.

4.2.4 사용 예

다음은 시작 주소가 "0"일 때 디지털 입력포트 #0 ~ #7을 읽는 사용 예입니다.

● 요청

Example of Request

								FC	SA		QI	
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	02	00	00	00	08

그림 4-11 요청 예

바이트 순서	값(HEX)	의미
7	0x02	함수 코드 02
8~9	0x0000	읽어올 주소 0으로 설정
10~11	0x0008	시작 주소부터 8개의 디지털 입력포트를 읽음

표 4-3 요청 예

● 응답

Example of Response

								FC	BC	IS
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9
Value(HEX)	00	00	00	00	00	04	01	02	01	15

그림 4-12 응답 예

바이트 순서	값(HEX)	의미
7	0x02	함수 코드 02
8	0x01	1 바이트, 1 ~ 8개 사이의 입력포트
9	0x15	(0001 0101) #0, 2, 4 포트 ON / #1, 3, 5 ~ 7 포트 OFF

표 4-4 응답 예

4.3 Read Holding Registers (FC 03)

디지털/아날로그 입력포트, 디지털 출력포트 상태 확인에 사용됩니다.

4.3.1 요청

Request of Read Holding Registers

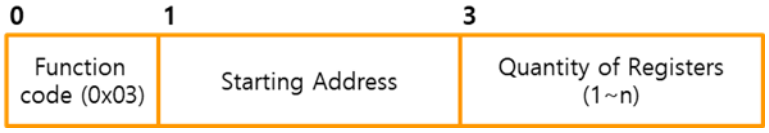


그림 4-13 Request of Read Holding Registers

- byte 0: 함수 코드
  Read Holding Registers의 함수 코드는 0x03 입니다.
- byte 1~2: 시작 주소
  상태 값을 읽을 첫 번째 레지스터 주소입니다.
- byte 3~4: 레지스터 개수
  값을 읽을 레지스터 수를 지정합니다. 사용 가능한 값의 범위는 1 ~ n입니다.

☞ ***n: 125 (SIG 제품군), 8 (CIE 제품군), 1 (EZI-10)***

4.3.2 응답

Response of Read Holding Registers

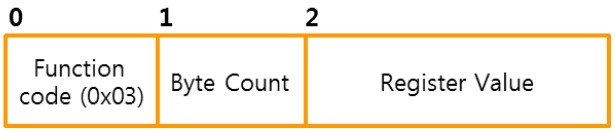


그림 4-14 Response of Read Holding Registers

- byte 0: 함수 코드 (0x03)
- byte 1: 바이트 카운트
  레지스터 개수 × 2
- byte 2 ~: 레지스터 값
  디지털/아날로그 포트의 상태를 나타냅니다. 디지털 포트는 레지스터 하나에 최대 16개의 포트를 비트 단위로 나타냅니다. 시작 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 표시되고 비트 값 0은 OFF를 1은 ON을 의미하며 제품에 없는 포트에 해당되는 비트는 0으로 채워집니다.

Register Value for Digital Inputs / Outputs (DI / DO)

15															0
0	0	0	0	0	0	0	0	#7	#6	#5	#4	#3	#2	#1	#0
MSB															LSB

그림 4-15 디지털 포트에 대한 레지스터 값

아날로그 포트는 레지스터 하나에 포트 1개를 나타냅니다(데이터 범위: 0 ~ n).

☞ ***n: 4095 (SIG 제품군, 분해능 12비트), 1023 (CIE 제품군, 분해능 10비트)***

Register Value for Analog Inputs (AI)

15				11		9	0
0	0	0	0			Analog Input Value	

그림 4-16 아날로그 포트에 대한 레지스터 값

4.3.3 예외

Exceptions of Read Holding Registers

0	1
Error code (0x83)	Exception code

그림 4-17 Exception of Read Holding Registers

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x83 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02 또는 0x03 입니다.

4.3.4 사용 예

다음은 시작 주소가 "0"일 때 디지털 입력포트 상태를 확인하는 사용 예입니다.

- 요청

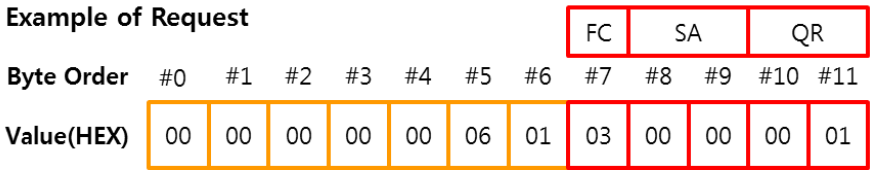


그림 4-18 요청 예

바이트 순서	값(HEX)	의미
7	0x03	함수 코드 03
8~9	0x0000	읽어올 주소 0으로 설정
10~11	0x0001	시작 주소부터 1개의 레지스터 읽음

표 4-5 요청 예

- 응답

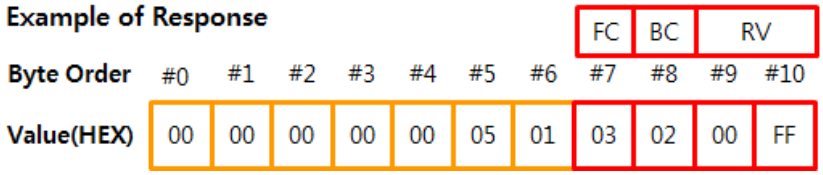


그림 4-19 응답 예

바이트 순서	값(HEX)	의미
7	0x03	함수 코드 03
8	0x02	2 바이트 즉, 1개의 레지스터
9~10	0x00FF	(1111 1111) 디지털 입력포트 #0 ~ 7 ON

표 4-6 응답 예

4.4 Read Input Registers (FC 04)

디지털/아날로그 입력포트 상태 확인에 사용됩니다.

4.4.1 요청

Request of Read Input Registers

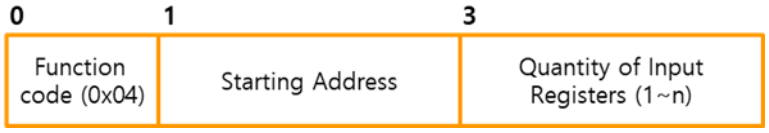


그림 4-20 Request of Read Input Registers

- byte 0: 함수 코드
  Read Input Registers의 함수 코드는 0x04 입니다.
- byte 1~2: 시작 주소
  값을 읽을 첫 번째 레지스터 주소입니다.
- byte 3~4: 레지스터 개수
  값을 읽을 레지스터 수를 지정합니다. 사용 가능한 값의 범위는 1~n입니다.

☞ ***n: 125 (SIG 제품군), 8 (CIE 제품군), 1 (EZI-10)***

4.4.2 응답

Response of Read Input Registers

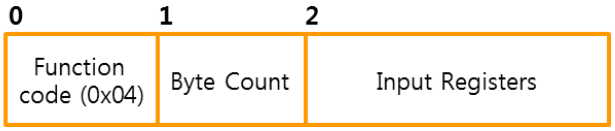


그림 4-21 Response of Read Input Registers

- byte 0: 함수 코드 (0x04)
- byte 1: 바이트 카운트
  레지스터 개수 × 2
- byte 2 ~: 레지스터 값
  디지털/아날로그 입력포트의 상태를 나타냅니다. 디지털 포트는 레지스터 하나에 최대 16개의 포트를 비트 단위로 나타냅니다. 시작 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 표시되고 비트 값 0은 OFF를 1은 ON을 의미하며 제품에 없는 포트에 해당되는 비트는 0으로 채워집니다.

Register Value for Digital Inputs (DI)

15															0
0	0	0	0	0	0	0	0	#7	#6	#5	#4	#3	#2	#1	#0
MSB															LSB

그림 4-22 디지털 입력포트에 대한 레지스터 값

아날로그 포트는 레지스터 하나에 포트 1개를 나타냅니다(데이터 범위: 0 ~ n).

☞ ***n: 4095 (SIG 제품군, 분해능 12비트), 1023 (CIE 제품군, 분해능 10비트)***

Register Value for Analog Inputs (AI)

15				11	9	0
0	0	0	0			Analog Input Value

그림 4-23 아날로그 포트에 대한 레지스터 값

4.4.3 예외

Exceptions of Read Input Registers

0	1
Error code (0x84)	Exception code

그림 4-24 Exception of Read Input Registers

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x84 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02 또는 0x03 입니다.

4.4.4 사용 예

다음은 시작 주소가 "0"일 때 디지털 입력포트 상태를 확인하는 사용 예입니다.

- 요청

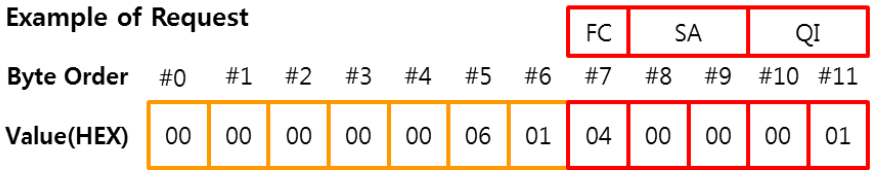


그림 4-25 요청 예

바이트 순서	값(HEX)	의미
7	0x04	함수 코드 04
8~9	0x0000	읽어올 주소 0으로 설정
10~11	0x0001	시작 주소부터 1개의 레지스터 읽음

표 4-7 요청 예

- 응답

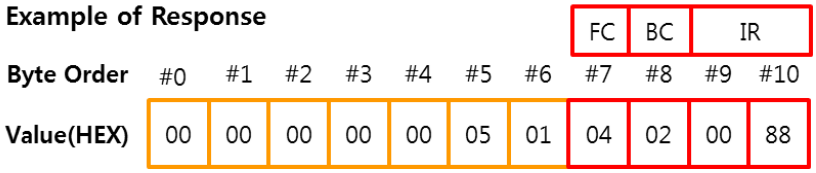


그림 4-26 응답 예

바이트 순서	값(HEX)	의미
7	0x04	함수 코드 04
8	0x02	2 바이트, 1개의 레지스터
9~10	0x0088	(1000 1000) #3, 7 포트 ON / #0 ~ 2, 4 ~ 6 포트 OFF

표 4-8 응답 예

4.5 Write Single Coil (FC 05)

하나의 디지털 출력포트 ON/OFF 제어에 사용됩니다.

4.5.1 요청 / 응답

Request / Response of Write Single Coil

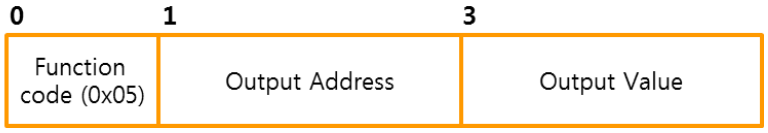


그림 4-27 Request / Response of Write Single Coil

- byte 0: 함수 코드
  Write Single Coil의 함수 코드는 0x05 입니다.
- byte 1~2: 출력포트 주소
  제어할 디지털 출력포트의 주소입니다.
- byte 3~4: 데이터 값
  0xFF00은 출력포트 ON을 0x0000은 OFF를 하기 위해 사용됩니다.

☞ ***Write Single Coil은 요청 프레임과 응답 프레임의 구조가 동일합니다.***

4.5.2 예외

Exceptions of Write Single Coil

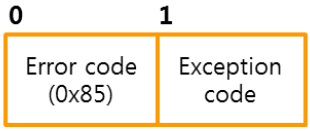


그림 4-28 Exception of Write Single Coil

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x85 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02, 0x03, 0x04 또는 0x06 입니다.

4.5.3 사용 예

다음은 시작 주소가 "8"일 때 디지털 출력포트 #0을 ON하는 사용 예입니다.

- 요청 및 응답

Example of Request / Response

								FC	OA		OV	
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	05	00	08	FF	00

그림 4-29 요청 / 응답 예

바이트 순서	값(HEX)	의미
7	0x05	함수 코드 05
8~9	0x0008	제어할 주소 8로 설정
10~11	0xFF00	데이터 값 0xFF00 (출력포트 ON)

표 4-9 요청 / 응답 예

4.6 Write Single Register (FC 06)

디지털 출력포트 ON/OFF 제어에 사용됩니다.

4.6.1 요청 / 응답

Request / Response of Write Single Register

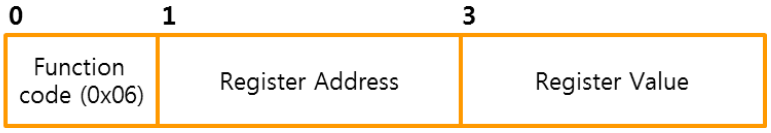


그림 4-30 Request / Response of Write Single Register

- byte 0: 함수 코드
  Write Single Register의 함수 코드는 0x06 입니다.
- byte 1~2: 레지스터 주소
  제어할 디지털 출력포트의 주소입니다.
- byte 3~4: 레지스터 값
  출력포트 제어에 이용할 값이며 레지스터 하나에 최대 16개의 출력포트를 비트 단위로 나타냅니다. 레지스터 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 적용되고 비트 값 0은 OFF를 1은 ON을 의미합니다. 출력포트 개수를 넘거나 제품에 없는 포트에 해당되는 비트의 값은 무시됩니다(일부 제품 제외, "SIG 제품군"은 예외 응답 처리).

Register Value for Digital Outputs (DO)

그림 4-31 디지털 출력포트에 대한 레지스터 값

☞ ***Write Single Register는 요청 프레임과 응답 프레임의 구조가 동일합니다.***

4.6.2 예외

Exceptions of Write Single Register

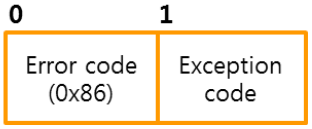


그림 4-32 Exception of Write Single Register

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x86 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02 또는 0x04 입니다.

4.6.3 사용 예

다음은 시작 주소가 "8"일 때 디지털 출력포트를 제어하는 사용 예입니다.

- 요청 및 응답

Example of Request / Response

								FC	RA		RV	
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	06	00	08	00	48

그림 4-33 요청 / 응답 예

바이트 순서	값(HEX)	의미
7	0x06	함수 코드 06
8~9	0x0008	제어할 주소 8로 설정
10~11	0x0048	(0100 1000) #3, 6 포트 ON / #0 ~ 2, 4, 5, 7 포트 OFF

표 4-10 요청 / 응답 예

4.7 Read Exception Status (FC 07)

Read Exception Status는 예외 응답과 무관하며 제품 디지털 출력포트 중 매크로가 설정된 포트를 확인합니다.

☞ ***"SIG 제품군"은 FC 07을 지원하지 않습니다.***

4.7.1 요청

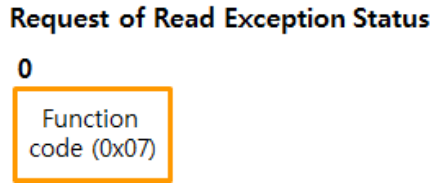


그림 4-34 Request of Read Exception Status

- byte 0: 함수 코드
  Read Exception Status의 함수 코드는 0x07 입니다.

4.7.2 응답

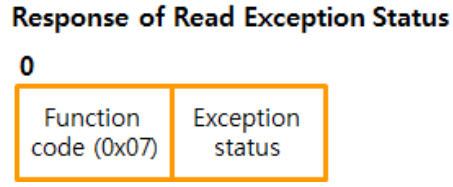


그림 4-35 Response of Read Exception Status

- byte 0: 함수 코드 (0x07)
- byte 1: 포트 상태 값 (Exception Status)
  매크로모드가 설정된 출력포트는 비트 1로, 그렇지 않은 포트는 비트 0으로 표시됩니다. 첫 번째 출력포트부터 LSB에서 MSB 방향으로 표시되고 제품에 없는 포트에 해당되는 비트의 값은 0으로 채워집니다.

4.7.3 예외

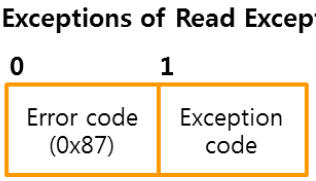


그림 4-36 Exception of Read Exception Status

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x87 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01 입니다.

4.7.4 사용 예

매크로 모드가 설정된 디지털 출력포트를 확인하는 사용 예입니다.

- 요청

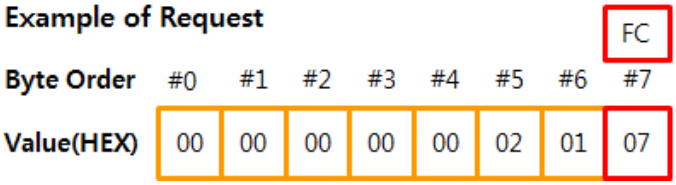


그림 4-37 요청 예

바이트 순서	값(HEX)	의미
7	0x07	함수 코드 07

표 4-11 요청 예

- 응답

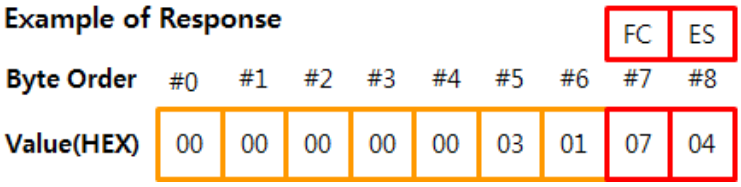


그림 4-38 응답 예

바이트 순서	값(HEX)	의미
7	0x07	함수 코드 07
8	0x04	(0000 0100) #2 포트 매크로 ON

표 4-12 응답 예

4.8 Write Multiple Coils (FC 15)

연속적으로 있는 여러 개의 디지털 출력포트 ON/OFF 제어에 사용됩니다.

4.8.1 요청

Request of Write Multiple Coils

0	1	3	5	6
Function code (0x0F)	Starting Address	Quantity of Outputs (1 ~ n)	Byte Count (0x01)	Output Value

그림 4-39 Request of Write Multiple Coils

- byte 0: 함수 코드
  Write Multiple Coils의 함수 코드는 0x0F 입니다.
- byte 1~2: 시작 주소
  제어할 첫 번째 디지털 출력포트의 주소입니다.
- byte 3~4: 출력포트 개수
  제어할 디지털 출력포트 수를 지정합니다. 사용 가능한 값의 범위는 1 ~ n까지입니다.

☞ ***n: 각 제품의 디지털 출력포트 개수***

- byte 5: 바이트 카운트 (0x01)

  (출력포트 개수+7) / 8

- byte 6: 출력포트 값
  디지털 출력포트 제어에 이용할 값이며 포트 개수에 따라 바이트 단위로 추가되며 1바이트에 8개의 출력포트를 비트 단위로 나타냅니다. 시작 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 적용되고 비트 값 0은 OFF를 1은 ON을 의미하며 출력포트 개수를 넘거나 제품에 없는 포트에 해당되는 비트의 값은 무시됩니다.

Output Value for Digital Outputs (DO)

MSB							LSB
#7	#6	#5	#4	#3	#2	#1	#0

그림 4-40 디지털 출력포트에 대한 출력포트 값

4.8.2 응답

바이트 카운트와 출력포트 값 부분을 제외하고 요청 패킷과 동일합니다.

Response of Write Multiple Coils

그림 4-41 Response of Write Multiple Coils

- byte 0: 함수 코드 (0x0F)
- byte 1~2: 시작 주소
- byte 3~4: 출력포트 개수

4.8.3 예외

Exceptions of Write Multiple Coils

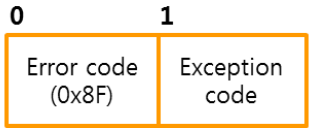


그림 4-42 Exception of Write Multiple Coils

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x8F 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02, 0x03, 0x04 또는 0x06 입니다.

4.8.4 사용 예

다음은 시작 주소가 "8"일 때 디지털 출력포트 4개를 제어하는 사용 예입니다.

● 요청

Example of Request

								FC	SA		QO		BC	OV
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11	#12	#13
Value(HEX)	00	00	00	00	00	08	01	0F	00	08	00	04	01	03

그림 4-43 요청 예

바이트 순서	값(HEX)	의미
7	0x0F	함수 코드 15
8~9	0x0008	제어할 주소 8로 설정
10~11	0x0004	시작주소부터 4개의 디지털 출력포트 제어
12	0x01	1 바이트, 1 ~8개 사이의 디지털 출력포트
13	0x03	(0000 0011) #0, 1 포트 ON / #2, 3 포트 OFF

표 4-13 요청 예

● 응답

Example of Response

								FC	SA		QO	
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	0F	00	08	00	04

그림 4-44 응답 예

바이트 순서	값(HEX)	의미
7	0x0F	함수 코드 15
8~9	0x0008	제어할 주소 8로 설정
10~11	0x0004	시작주소부터 4개의 디지털 출력포트 제어

표 4-14 응답 예

4.9 Write Multiple Registers (FC 16)

출력포트 ON/OFF 제어에 사용됩니다.

4.9.1 요청

Request of Write Multiple Registers

0	1	3	5	6
Function code (0x10)	Starting Address	Quantity of Registers (0x0001)	Byte Count (0x02)	Register Value

그림 4-45 Request of Write Multiple Registers

- byte 0: 함수 코드
  Write Multiple Registers의 함수 코드는 0x10 입니다.
- byte 1~2: 시작 주소
  값을 쓸 첫 번째 레지스터 주소입니다.
- byte 3~4: 레지스터 개수 (0x0001)
  값을 쓸 레지스터 수를 지정합니다. 사용 가능한 값은 1 입니다.
- byte 5: 바이트 카운트 (0x02)
  레지스터 개수 × 2
- byte 6~7: 레지스터 값
  출력포트 제어에 이용할 값이며 레지스터 하나에 16개의 출력포트를 비트 단위로 나타냅니다. 시작 주소에 해당되는 포트부터 LSB에서 MSB 방향으로 적용되고 비트 값 0은 OFF를 1은 ON을 의미합니다. 출력포트 개수를 넘거나 제품에 없는 포트에 해당되는 비트의 값은 무시됩니다(일부 제품 제외, "SIG 제품군"은 예외 응답 처리).

Register Value for Digital Outputs (DO)

15															0
0	0	0	0	0	0	0	0	#7	#6	#5	#4	#3	#2	#1	#0
MSB															LSB

그림 4-46 디지털 출력포트에 대한 레지스터 값

4.9.2 응답

바이트 카운트와 레지스터 값 부분을 제외하고 요청 패킷과 동일합니다.

Response of Write Multiple Registers

그림 4-47 Response of Write Multiple Registers

- byte 0: 함수 코드 (0x10)
- byte 1~2: 시작 주소
- byte 3~4: 레지스터 개수

4.9.3 예외

Exceptions of Write Multiple Registers

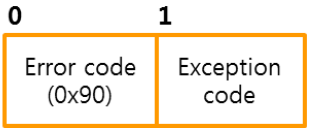


그림 4-48 Exception of Write Multiple Registers

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0x90 입니다.
- byte 1: 예외 코드
  예외 코드는 0x01, 0x02, 0x03 또는 0x04 입니다.

4.9.4 사용 예

다음은 시작 주소가 "8"일 때 디지털 출력포트를 제어하는 사용 예입니다.

● 요청

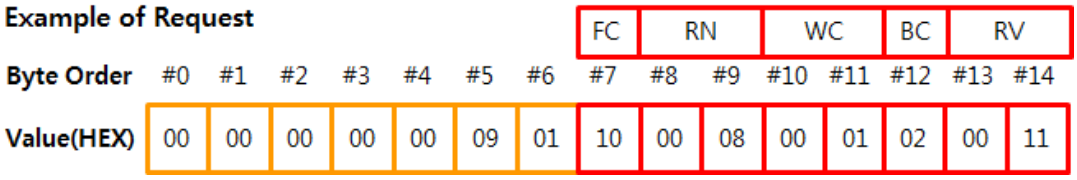


그림 4-49 요청 예

바이트 순서	값(HEX)	의미
7	0x10	함수 코드 16
8~9	0x0008	제어할 주소 8로 설정
10~11	0x0001	시작 주소부터 1개의 레지스터 값 쓰기
12	0x02	2 바이트, 1개 레지스터
13~14	0x0011	(0001 0001) #0, 4 포트 ON / #1 ~3, 5 ~ 7 포트 OFF

표 4-15 요청 예

● 응답

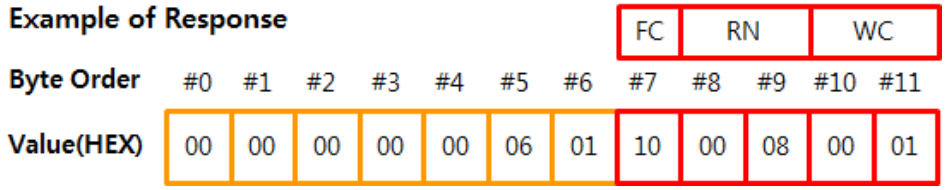


그림 4-50 응답 예

바이트 순서	값(HEX)	의미
7	0x03	함수 코드 16
8~9	0x0008	제어할 주소 8로 설정
10~11	0x0001	시작 주소부터 1개의 레지스터 값 쓰기

표 4-16 응답 예

4.10 Encapsulated Interface Transport (FC 43)

Modbus/TCP가 아닌 다른 프로토콜에 사용되는 통신 패킷을 Modbus/TCP 프로토콜 데이터 부분에 실어서 통신하기 위한 함수 코드이며 이러한 통신 구조를 MEI(Modbus Encapsulated Interface)라고 부릅니다. 캡슐화되는 프로토콜의 종류에 따라 MEI 타입으로 구분하며 13(0x0D) – CANopen General Reference와 14(0x0E) – Read Device Identification 총 2개의 MEI 타입이 있습니다.

☞ ***EZI-10 제품은 FC 43을 지원하지 않습니다.***

4.10.1 요청

Request of Encapsulated Interface Transport

그림 4-51 Request of Encapsulated Interface Transport

- byte 0: 함수 코드

  Encapsulated Interface Transport의 함수 코드는 0x2B 입니다.

- byte 1: MEI 종류 (0x0D 혹은 0x0E)

  솔내시스템 I/O 게이트웨이는 14(0x0E) – Read Device Identification만 지원합니다.

- byte 2~: 실제 데이터(n bytes)

  MEI 종류에 따라 내용이 다릅니다.

4.10.2 응답

Response of Encapsulated Interface Transport

그림 4-52 Response of Encapsulated Interface Transport

- byte 0: 함수 코드 (0x2B)
- byte 1: MEI 종류 (0x0D 혹은 0x0E)
- byte 2~: 실제 데이터(n bytes)

4.10.3 예외

Exceptions of Encapsulated Interface Transport

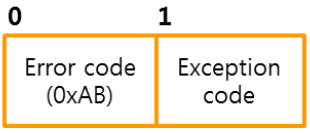


그림 4-53 Exception of Encapsulated Interface Transport

- byte 0: 에러 코드

  에러 코드는 "함수 코드 + 0x80", 즉 0xAB 입니다.

- byte 1: 예외 코드

  예외 코드는 0x01, 0x02, 0x03 또는 0x04 입니다.

4.10.4 사용 예

"4.11 Read Device Identification (FC 43 / 14)"를 참조하십시오.

4.11 Read Device Identification (FC 43 / 14)

Modbus 서버 장비 정보 확인을 위해 사용되며 각각의 장비 정보를 오브젝트라 부르고 오브젝트는 그 특징에 따라 크게 3 종류로 나뉩니다.

- 기본 정보 (Basic Device Identification) – 필수
  제조사, 제품 코드, 펌웨어 버전
- 정규 정보 (Regular Device Identification) – 옵션
  제조사 홈페이지 주소, 제품명, 모델명, 사용자 응용프로그램명
- 확장 정보 (Extended Device Identification) – 옵션
  I/O 게이트웨이는 제품 설명 환경 값, 제품 MAC 주소 등 총 8가지 항목을 지원합니다.

종류	아이디	오브젝트 이름 및 설명	데이터 형태	필수여부
Basic	0x00	VendorName	ASCII String	**필수**
	0x01	ProductCode	ASCII String	**필수**
	0x02	MajorMinorRevision	ASCII String	**필수**
Regular	0x03	VendorUrl	ASCII String	옵션
	0x04	ProductName	ASCII String	옵션
	0x05	ModelName	ASCII String	옵션
	0x06	UserApplicationName	ASCII String	옵션
	0x07~0x7F	Reserved		옵션
Extended	0x80	Comment	Binary	옵션
	0x81	MAC Address	ASCII String	옵션
	0x82	Macro Mode	Binary	옵션
	0x83	TCP Session ID	Binary	옵션
	0x84	1-bit ADC Mode	Binary	옵션
	0x85	DO Pulse Mode	Binary	옵션
	0xA0 + n	Input Comments	Binary	옵션
	0xB0 + n	Output Comments	Binary	옵션

표 4-17 오브젝트 아이디

☞ ***n: 각 제품의 디지털 입력/출력포트 개수***

4.11.1 요청

Request of Read Device Identification

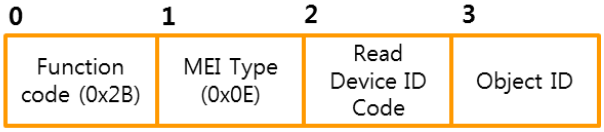


그림 4-54 Request of Read Device Identification

- byte 0: 함수 코드 (0x2B)
- byte 1: MEI 종류 (0x0E – Read Device Identification)
- byte 2: 디바이스 아이디(0x01 / 0x02 / 0x03 / 0x04)
  0x01: Basic 전체 요청
  0x02: Regular 전체 요청
  0x03: Extended 전체 요청
  0x04: Basic/Regular/Extended 구분 없이 특정 오브젝트 하나만 요청

  요청하는 정보를 구분합니다. 데이터가 많아서 전체 요청의 응답을 한 번에 보낼 수 없으면 트랜잭션(요청과 응답)이 여러 번 필요할 수 있습니다. 솔내시스템 I/O 게이트웨이는 Basic/Regular 전체 요청은 한번의 트랜잭션으로 끝나고 Extended에 대해서는 여러 번의 트랜잭션이 필요합니다.

- byte 3: 오브젝트 아이디
  첫 번째로 수신할 오브젝트 아이디를 의미합니다.
  전체 오브젝트 요청: 첫 번째 트랜잭션 – 0x00
  전체 오브젝트 요청: 두 번째 및 그 이후 트랜잭션 – 직전 응답에서 받은 값
  단일 오브젝트 요청: 실제 수신하고자 하는 오브젝트 아이디 값

  전체 오브젝트 요청일 때(디바이스 아이디가 0x01, 0x02, 혹은 0x03) 두 번째 및 그 이후 트랜잭션의 오브젝트 아이디 값이 적절하지 않으면 첫 번째 트랜잭션에 해당되는 응답(오브젝트 아이디 0x00)을 함으로서 트랜잭션을 처음부터 다시 시작합니다.

4.11.2 응답

Response of Read Device Identification

0	1	2	3	4	5	6	7	8	9
Function code (0x2B)	MEI Type (0x0E)	Read Device ID Code	Conformity Level	More Follows	Next Object ID	Number of Objects	Object ID	Object Length	Object Value

그림 4-55 Response of Read Device Identification

- byte 0: 함수 코드 (0x2B)
- byte 1: MEI 종류 (0x0E – Read Device Identification)
- byte 2: 디바이스 아이디(0x01, 0x02, 0x03, 0x04) – 요청과 동일
- byte 3: Conformity Level
  지원하는 오브젝트 종류와 요청 형태를 구분하며 I/O 게이트웨이는 0x83을 사용합니다.
  0x01: Basic (전체 요청만 지원)
  0x02: Regular (전체 요청만 지원)
  0x03: Extended (전체 요청만 지원)
  0x81: Basic (전체/단일 요청 둘 다 지원)
  0x82: Regular (전체/단일 요청 둘 다 지원)
  0x83: Extended (전체/단일 요청 둘 다 지원)
- byte 4: More Follows
  전체 오브젝트 요청이고 여러 번의 트랜잭션이 필요할 때 사용됩니다.
  0x00: 오브젝트 더 이상 없음, 마지막 트랜잭션을 의미
  0xFF: 오브젝트 더 있음, 추가적인 트랜잭션이 필요함
  단일 오브젝트 요청: 0x00으로 고정
- byte 5: 다음 오브젝트 아이디
  More Follows가 0xFF일 때: 다음 요청에 사용돼야 할 오브젝트 아이디
  More Follows가 0x00일 때: 0x00
- byte 6: 오브젝트 개수
  전체 오브젝트 요청: 응답되는 오브젝트의 개수
  단일 오브젝트 요청: 0x01
- byte 7: 오브젝트 아이디
  전체 오브젝트 요청: 첫 번째 오브젝트
  단일 오브젝트 요청: 요청된 오브젝트
- byte 8: 오브젝트 길이 – 첫 번째 오브젝트 데이터의 길이 (단위: 바이트)
- byte 9~: 오브젝트 데이터
  첫 번째 오브젝트 데이터입니다. 응답 오브젝트의 개수가 여러 개면 두 번째 오브젝트부터 아이디/길이/데이터 항목이 반복적으로 추가됩니다.

4.11.3 예외

Exceptions of Read Device Identification

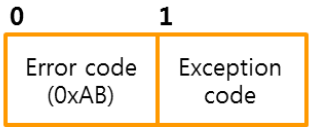


그림 4-56 Exception of Read Device Identification

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0xAB 입니다.
- byte 1: 예외 코드
  예외 코드는 0x01, 0x02, 0x03 또는 0x04 입니다.

4.11.4 사용 예 – TCP Session ID

다음은 현재 접속된 TCP 세션 아이디를 읽어오는 사용 예입니다.

- 요청

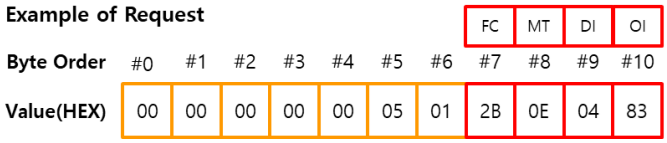


그림 4-57 요청 예

바이트 순서	값(HEX)	의미
7	0x2B	함수 코드 43
8	0x0E	Read Device Identification
9	0x04	특정 오브젝트 하나만 요청
10	0x83	TCP Session ID 요청

표 4-18 요청 예

- 응답

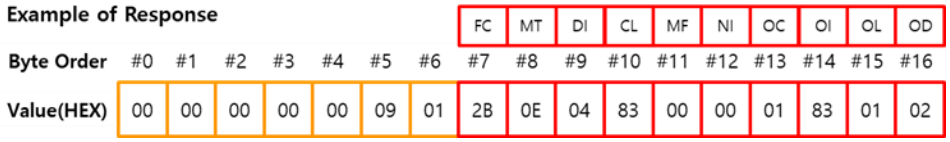


그림 4-58 응답 예

바이트 순서	값(HEX)	의미
7	0x2B	함수 코드 43
8	0x0E	Read Device Identification
9	0x04	특정 오브젝트 하나만 요청
10	0x83	Extended 전체/단일 요청까지 지원
11	0x00	추가 트랜잭션 없음
12	0x00	마지막 트랜잭션
13	0x01	1개의 오브젝트 정보 포함
14	0x83	TCP Session ID
15	0x01	1 바이트
16	0x02	TCP 세션 아이디 0x02

표 4-19 응답 예

4.11.5 사용 예 – Basic Device Identification

다음은 CIE-H10A의 Basic 디바이스 정보를 읽어오는 사용 예입니다.

- 요청

이름	값(HEX)	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x01	Basic 오브젝트 전체 요청
오브젝트 아이디	0x00	VendorName (전체 요청 시작)

표 4-20 요청 예

- 응답

이름	값	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x01	Basic 오브젝트 전체 요청
Conformity Level	0x83	Extended 전체/단일 요청까지 지원
More Follows	0x00	추가 트랜잭션 없음
다음 오브젝트 아이디	0x00	마지막 트랜잭션
오브젝트 개수	0x03	3개의 오브젝트 정보 포함
오브젝트 아이디	0x00	VendorName
오브젝트 길이	0x0E	14 바이트
오브젝트 데이터	"Sollae Systems"	
오브젝트 아이디	0x01	ProductCode
오브젝트 길이	0x02	2 바이트
오브젝트 데이터	"20"	CIE-H10A의 Product Code
오브젝트 아이디	0x02	MajorMinorRevision
오브젝트 길이	0x05	5 바이트
오브젝트 데이터	"V2.2B"	버전 2.2B

표 4-21 응답 예

4.11.6 사용 예 – Extended Device Identification

다음은 CIE-H10A의 Extended 디바이스 정보를 읽어오는 사용 예입니다.

- 요청 1

이름	값(HEX)	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x03	Extended 오브젝트 전체 요청
오브젝트 아이디	0x00	Comment (전체 요청 시작)

표 4-22 요청 1

- 응답 1

이름	값	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x03	Extended 오브젝트 전체 요청
Conformity Level	0x83	Extended 전체/단일 요청까지 지원
More Follows	0xFF	추가 트랜잭션 있음
다음 오브젝트 아이디	0xA0	입력포트 설명
오브젝트 개수	0x03	3개의 오브젝트 정보 포함
오브젝트 아이디	0x80	Comment
오브젝트 길이	0x00	0 바이트
오브젝트 아이디	0x81	MAC Address
오브젝트 길이	0x11	17 바이트
오브젝트 데이터	"00:30:F9:00:00:01"	제품 MAC 주소
오브젝트 아이디	0x82	Macro Mode
오브젝트 길이	0x01	1 바이트
오브젝트 데이터	0x81	#0, 7 포트 매크로 모드

표 4-23 응답 1

- 요청 2

이름	값(HEX)	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x03	Extended 오브젝트 전체 요청
오브젝트 아이디	0xA0	입력 포트 설명

표 4-24 요청 2

● 응답 2

이름	값	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x03	Extended 오브젝트 전체 요청
Conformity Level	0x83	Extended 전체/단일 요청까지 지원
More Follows	0xFF	추가 트랜잭션 있음
다음 오브젝트 아이디	0xB0	출력포트 설명
오브젝트 개수	0x08	8개의 오브젝트 정보 포함
오브젝트 아이디	0xA0	입력포트 #0 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI0"	기본 설정 값
오브젝트 아이디	0xA1	입력포트 #1 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI1"	기본 설정 값
오브젝트 아이디	0xA2	입력포트 #2 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI2"	기본 설정 값
오브젝트 아이디	0xA3	입력포트 #3 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI3"	기본 설정 값
오브젝트 아이디	0xA4	입력포트 #4 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI4"	기본 설정 값
오브젝트 아이디	0xA5	입력포트 #5 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI5"	기본 설정 값
오브젝트 아이디	0xA6	입력포트 #6 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI6"	기본 설정 값
오브젝트 아이디	0xA7	입력포트 #7 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DI7"	기본 설정 값

표 4-25 응답 2

● 요청 3

이름	값(HEX)	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x03	Extended 오브젝트 전체 요청
오브젝트 아이디	0xB0	오브젝트 아이디

표 4-26 요청 3

● 응답 3

이름	값	의미
함수 코드	0x2B	함수 코드 43
MEI 종류	0x0E	Read Device Identification
디바이스 아이디	0x03	Extended 오브젝트 전체 요청
Conformity Level	0x83	Extended 전체/단일 요청까지 지원
More Follows	0x00	추가 트랜잭션 없음
다음 오브젝트 아이디	0x00	마지막 트랜잭션
오브젝트 개수	0x08	8개의 오브젝트 정보 포함
오브젝트 아이디	0xB0	출력포트 #0 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO0"	기본 설정 값
오브젝트 아이디	0xB1	출력포트 #1 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO1"	기본 설정 값
오브젝트 아이디	0xB2	출력포트 #2 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO2"	기본 설정 값
오브젝트 아이디	0xB3	출력포트 #3 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO3"	기본 설정 값
오브젝트 아이디	0xB4	출력포트 #4 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO4"	기본 설정 값
오브젝트 아이디	0xB5	출력포트 #5 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO5"	기본 설정 값
오브젝트 아이디	0xB6	출력포트 #6 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO6"	기본 설정 값
오브젝트 아이디	0xB7	출력포트 #7 Comment
오브젝트 길이	0x03	3 바이트
오브젝트 데이터	"DO7"	기본 설정 값

표 4-27 응답 3

5 사용자 정의 함수

5.1 Write Pulse (FC 105)

출력포트를 일정 시간 동안만 ON 혹은 OFF 상태를 유지시키고 다시 원래 상태로 돌아가는 펄스 형태로 제어하기 위해 사용됩니다.

☞ ***"SIG 제품군"은 FC 105를 지원하지 않습니다.***

5.1.1 요청 / 응답

Request / Response of Write Pulse

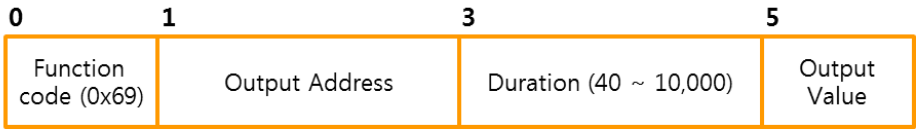


그림 5-1 Request / Response of Write Pulse

- byte 0: 함수 코드
  Write Pulse의 함수 코드는 0x69 입니다.
- byte 1~2: 출력포트 주소
  제어할 디지털 출력포트의 주소입니다.
- byte 3~4: 유지 시간
  단위는 밀리 초(ms)이며 설정 가능한 범위는 40 ~ 10,000 (0x0028 ~ 0x2710) 입니다.
- byte 5: 출력포트 값
  출력포트 ON을 유지하기 위해 0xFF 또는 OFF를 유지하기 위해 0x00을 설정합니다. 출력포트 값이 현재의 출력포트와 같은 값이면 예외 코드 0x04를 응답합니다.

☞ ***Write pulse는 요청 프레임과 응답 프레임의 구조가 동일합니다.***

5.1.2 예외

Exceptions of Write Pulse

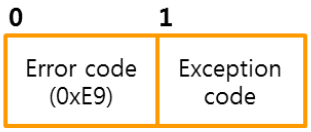


그림 5-2 Exception of Write Pulse

- byte 0: 에러 코드
  에러 코드는 "함수 코드 + 0x80", 즉 0xE9 입니다.
- byte 1: 예외 코드(Exception code)
  예외 코드는 0x01, 0x02, 0x03 또는 0x04 입니다.

5.1.3 사용 예

- 요청 / 응답

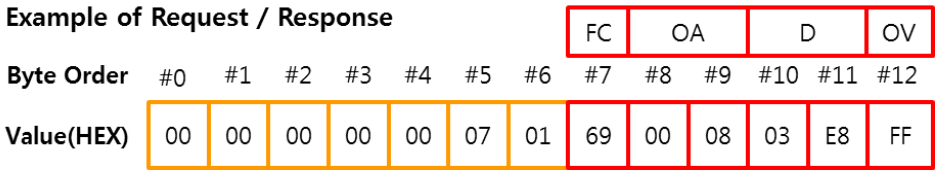


그림 5-3 요청 / 응답 예

바이트 순서	값(HEX)	의미
7	0x69	함수 코드 105
8~9	0x0008	제어할 출력포트 주소를 나타냄
10~11	0x03E8	1초 (1000ms = 0x03E8) 동안 유지
12	0xFF	데이터 값 0xFF (ON 상태로 유지)

표 5-1 요청 / 응답 예

☞ ***현재 출력포트가 이전에 요청받은 FC 105로 제어중이거나 매크로 모드일 때는 제어할 수 없습니다.***

5.2 Send Notify (FC 108)

포트 상태가 변경될 때 알림용으로 사용되며 응답 패킷만 정의되어 있습니다.

☞ ***"CIE 제품군"과 EZI-10 제품은 FC 108을 지원하지 않습니다.***

5.2.1 응답

Response of Send Notify

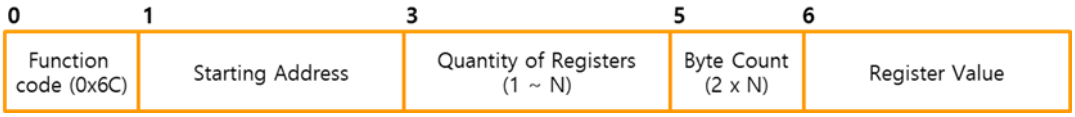


그림 5-4 Response of Send Notify

- byte 0: 함수 코드

  Send Notify의 함수 코드는 0x6C 입니다.

- byte 1~2: 시작 주소

  전송할 레지스터 시작 주소이며 주요 포트들의 시작 주소는 다음과 같습니다.

이름	주소	설명
DI Counter	0x00A0	디지털 입력포트 카운터 값
DI Value	0x00F0	디지털 입력포트 상태
1-bit ADC	0x00FA	아날로그 입력포트 상태를 1비트 디지털로 변환한 값
DO Value	0x0140	디지털 출력포트 상태

표 5-2 각 포트 시작 주소

- byte 3~4: 레지스터 개수 (1 ~ N)

  전송할 레지스터 개수를 지정합니다. 사용 가능한 값은 1~N입니다.

- byte 5: 바이트 카운트 (2 x N)

  레지스터 개수 × 2

- byte 6~ (2 x N + 5): 레지스터 값

5.2.2 사용 예

다음은 디지털 입력포트 상태 변경을 알리는 응답 패킷 예입니다(시작 주소 "240").

- 응답

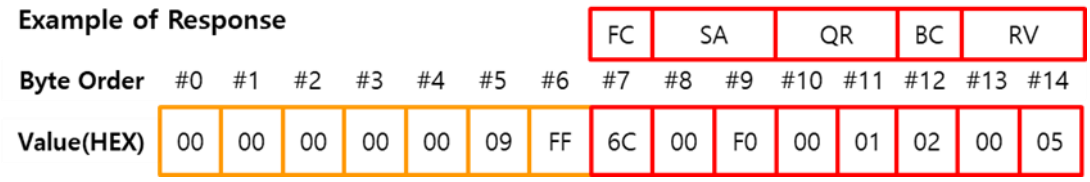


그림 5-5 응답 예

바이트 순서	값(HEX)	의미
7	0x6C	함수 코드 108
8~9	0x00F0	시작 주소 240(디지털 입력포트 상태 값 저장 주소)
10~11	0x0001	시작 주소부터 1개의 레지스터 값 전송
12	0x02	2 바이트, 1개 레지스터
13~14	0x0005	(0000 0101) #0, 2 포트 ON / #1, 3 포트 OFF

표 5-3 응답 예

6 기타 알아두어야 할 사항

6.1 예외 코드와 의미

예외 코드	명칭	의미
0x01	Illegal Function	함수 코드 오류
0x02	Illegal Data Address	시작 주소 오류
0x03	Illegal Data Value	데이터 값 오류
0x04	Server Device Failure	요청 명령 실행 실패 (CIE 제품군, EZI-10)
0x06	Slave Device Busy	요청 명령 실행 실패 (SIG 제품군)

표 6-1 예외 코드

6.2 CIE-M10A 아날로그 포트 값 읽기

6.2.1 요청

CIE-M10A 아날로그 입력포트는 FC 04 (Read Input Registers)를 이용해 읽을 수 있으며 이 때 읽기 시작할 레지스터 주소를 "[디지털 입력포트 시작주소] + 4"로 지정해주어야 합니다(참고로 "SIG 제품군"은 0으로 지정). 예를 들어 [디지털 입력포트 시작주소] 설정 값이 0이면, 아날로그 입력포트의 주소는 4번지가 됩니다. 따라서 전송 예는 다음과 같습니다.

- 요청 예

Example of Request								FC	SA		QR	
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10	#11
Value(HEX)	00	00	00	00	00	06	01	04	00	04	00	01

그림 6-1 CIE-M10A 아날로그 포트 값 요청 예

6.2.2 응답

CIE-M10A 아날로그 입력포트 읽기 요청에 대한 응답은 다음과 같습니다.

- 응답 예

Example of Response								FC	BC	RV	
Byte Order	#0	#1	#2	#3	#4	#5	#6	#7	#8	#9	#10
Value(HEX)	00	00	00	00	00	05	01	04	02	02	7F

그림 6-2 CIE-M10A 아날로그 포트 값 응답 예

위 예에서 레지스터 값 0x027F는 10진수로 표현하면 639가 됩니다.

6.3 샘플 코드

당사에서는 I/O 게이트웨이 사용자를 위해 Modbus/TCP 샘플코드를 제공하고 있습니다. 프로그램 구현에 활용하시기 바랍니다.

☞ ***당사 홈페이지 [다운로드]>>[자료실] 게시판에서 다운로드 구분 [샘플코드] 선택 후 검색하면 다운로드 받을 수 있습니다. (https://www.sollae.co.kr/kr/download/pds.php)***

6.3.1 제공 버전

- C++ (Visual Studio 2008)

7 시리얼 Modbus/TCP

시리얼 Modbus/TCP는 제품의 시리얼포트를 이용해 I/O를 감시/제어합니다. 시리얼포트가 있는 I/O 게이트웨이는 시리얼 Modbus/TCP 모드를 지원합니다.

7.1 특징

- 기존 Modbus/TCP 데이터를 시리얼포트로 송/수신
- 시리얼포트를 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.

7.2 사용하기

7.2.1 설정 방법

- 시리얼 Modbus/TCP 모드 설정

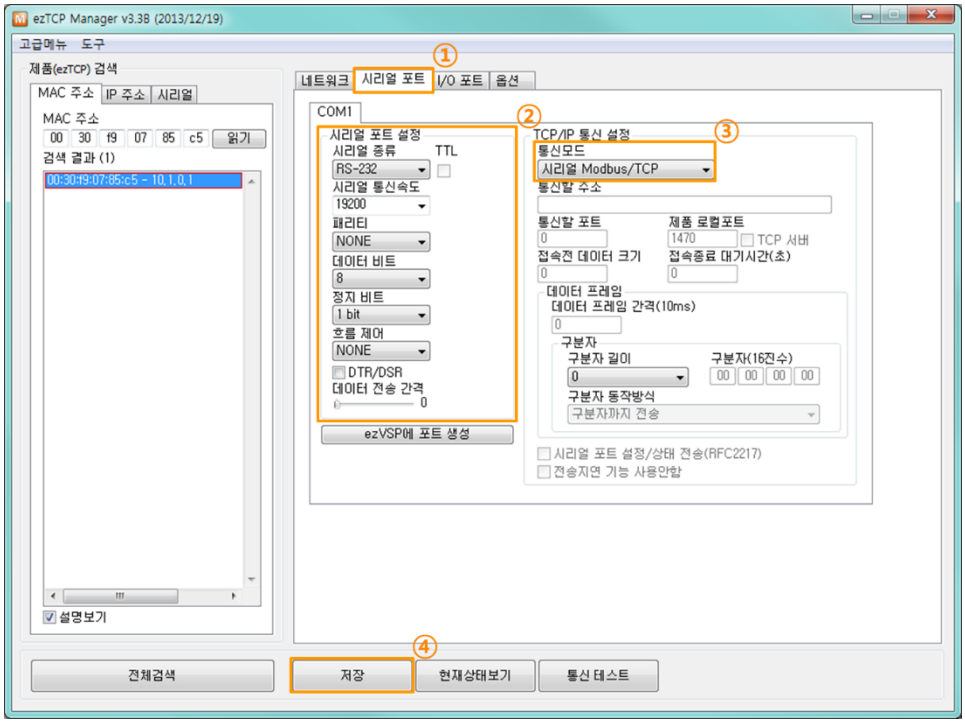


그림 7-1 시리얼 Modbus/TCP 모드 설정

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

7.3 시험 작동

7.3.1 통신 준비

시리얼 Modbus/TCP 동작을 시험하기 위해 다음처럼 구성해 주시기 바랍니다.

☞ ***LAN 케이블은 연결하지 않아도 무방합니다.***

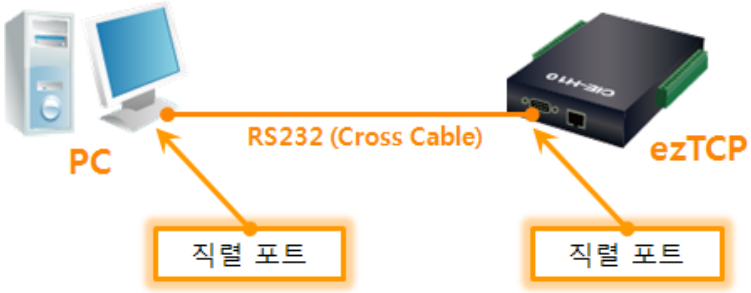


그림 7-2 통신 준비

시험을 위해 Modbus/TCP 설정은 다음과 같이 기본 값을 유지하시기 바랍니다.

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

표 7-1 Modbus/TCP 설정 기본 값

7.3.2 시험 데이터 전송

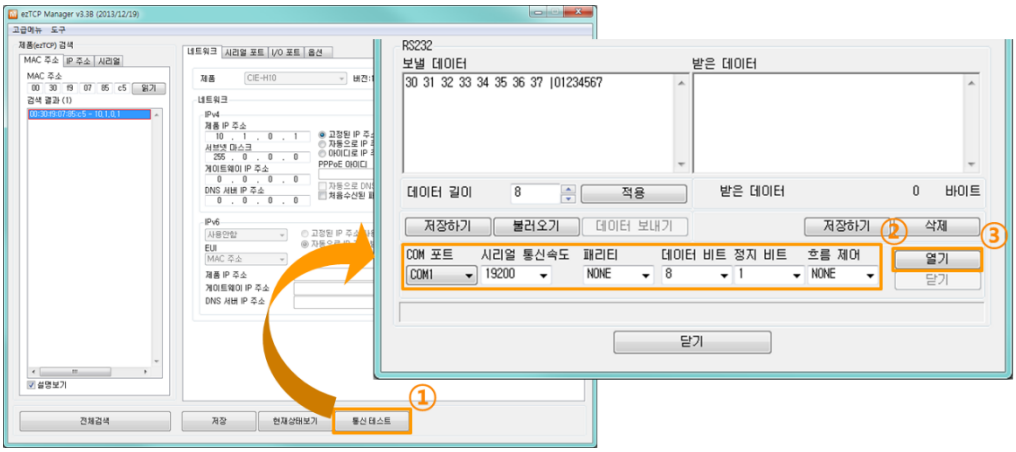


그림 7-3 시험 데이터 전송 1

① ezManager의 [통신 테스트]버튼 클릭

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

③ [열기]버튼 클릭하여 포트 열기

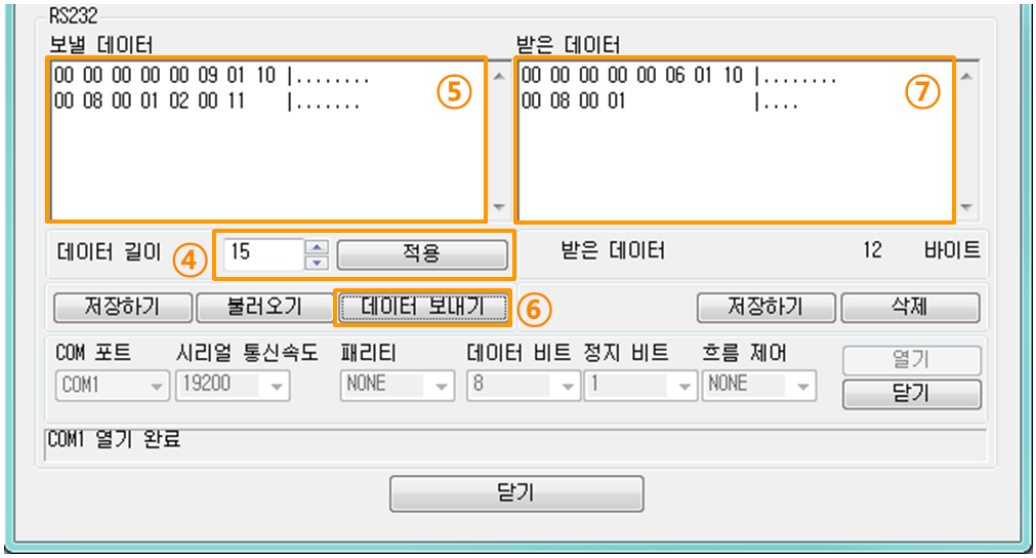


그림 7-4 시험 데이터 전송 2

④ 데이터 길이를 15(Bytes)로 설정하고 [적용]버튼 클릭

⑤ [보낼 데이터]에 write multiple registers 함수의 예제 데이터를 입력

⑥ [데이터 보내기]버튼 클릭

⑦ [받은 데이터]에 나타난 ezTCP의 응답 데이터가 위 그림과 같은지 확인

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

8 주의 사항

- 본 문서는 표준 문서 "MODBUS Application Protocol Specification (v1.1b3)"과 "MODBUS Messaging Implementation Guide (v1.0b)"를 기반으로 솔내시스템 I/O 게이트웨이가 지원하는 Modbus/TCP 프로토콜에 대해 설명합니다.
- 당사는 본 문서를 작성함에 있어서 충분한 검토를 거쳤으나 문서 내 설명에 대해 어떠한 보증도 하지 않으며 사전 예고 없이 변경될 수 있습니다.
- 좀 더 자세한 내용은 MODBUS 프로토콜 표준 문서를 참조해주시기 바랍니다.

9 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
2010.11.23.	1.2	○ ADC값 읽기/응답 내용 추가 ○ 표지의 날짜 표시 제거	이인
2011.06.24.	1.3	○ 추가된 함수 코드 내용 포함(FC 1, 2, 4, 5, 6, 7, 15, 105) ○ 기존 함수 코드 내용 중 그림 수정 ○ 설정 툴 캡쳐 화면 수정 ○ 문서 구조 및 일부 용어 변경 ○ 문서 제목 수정	이인
2014.04.30.	1.4	○ 용어 설명 추가 ○ Modbus 데이터 및 메모리 구조 설명 추가 ○ 함수 코드 클래스 구분 삭제 및 종류 설명 추가 ○ 기존 함수 코드 설명 수정 및 그림 교체 ○ 함수 코드 추가 (FC 43, FC 43 / 14) ○ 설정 프로그램 캡쳐 화면 수정 ○ 문서 제목 수정	이성운
2015.02.13.	1.5	○ 오탈자 수정	이성운
2017.08.04.	1.6	○ 2.1 개요 내용 수정 ○ MODBUS 데이터, 메모리, 주소 관련 내용 수정 ○ 샘플 코드 설명 삭제 및 링크 오류 수정	이성운
2021.09.08.	1.7	○ 신제품(SIG-5430/5440/5450/5600) 관련 내용 반영 ○ 사용 예 추가 (FC 43, TCP Session ID) ○ 함수 코드 추가 (FC 108) ○ 일부 오류 수정	이성운