Modbus RTU Master Manual

V2.3.7

본 매뉴얼은 MP STUDIO에서 Modbus RTU Master를 사용하는 방법에 대하여 설명합니다.

1. Introduction to Modbus RTU Master

Modbus는 1979년 Modicon 회사에서 산업에서 사용할 수 있도록 개발한 범용 프로토콜입니다. 지금은 산업에서 사용하는 통신이 되는 장비들은 대부분 Modbus를 지원하고 있습니다.

Modbus는 통신 프로토콜의 데이터 형식에 따라 Modbus RTU와 Modbus ASCII로 나뉘어 집니다. Modbus RTU는 데이터타입이 Hex로 되어 있고, ASCII는 데이터타입이 ASCII로 되어 있습니다. 두 프로토콜 중에서 Modbus RTU를 사용하는 장비들이 많습니다.

Modbus RTU 통신 프로토콜에서 지령을 내리는 장비가 Modbus RTU Master가 되고 지령을 받아서 응답을 하는 장비가 Modbus RTU Slave가 됩니다.

저희 아이로직스 제품중 통신이 되는 장비는 모두 Modbus RTU Master 및 Modbus RTU Slave를 지원합니다.

일반적으로 HMI와 같은 터치디스플레이가 Modbus RTU Master가 되고 PLC가 Modbus RTU Slave가 됩니다. 경우에 따라서, PLC와 PLC의 통신 또는 PLC와 인버터 등의 통신이 필요한 경우가 있습니다. 이럴 때에는, PLC가 Modbus RTU Master가 되어야 합니다. 이때, MP STUDIO에서 통신 프로토콜을 Modbus RTU Master로 설정하여 이용할 수 있습니다.

2. 주의사항

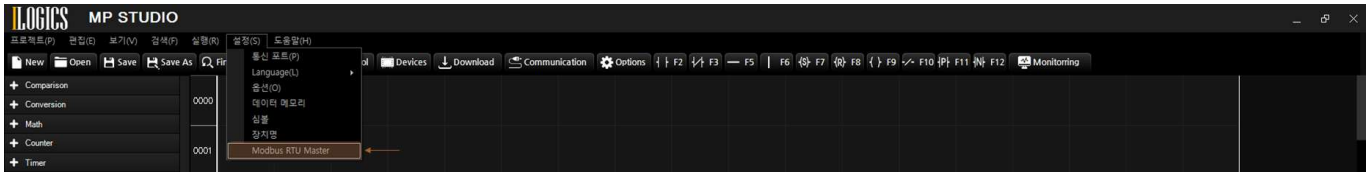
Modbus RTU Master는 1개의 채널에서만 지원됩니다. 예를들어 MPS-16A8R8T 제품은 채널1:RS-232, 채널2:RS-485, 채널3:UART 통신을 사용할 수 있습니다. 총 채널3개에서 채널2:RS-485를 Modbus RTU Master로 설정하시면, 채널1과 채널3에서는 Modbus RTU Master를 사용할 수 없습니다. (Modbus RTU Slave 등은 사용가능합니다)

RS-485를 Modbus RTU Master로 설정하시면, Modbus RTU Slave를 지원하는 여러대의 슬레이브 장비와 통신이 가능합니다.

3. Getting Started

3.1. Modbus RTU Master 설정창 실행

MP STUDIO의 메뉴에서 설정 -〉 Modbus RTU Master를 실행합니다.



3.2. 포트 설정

Modbus RTU Master로 사용할 통신포를 설정하고 Modbus RTU Slave와 동일하게 통신속도(BaudRate), 데이터 비트(Data Bit), 패리트 비트(Parity Bit), 스톱 비트(Stop Bit)를 설정합니다.

타임아웃은 Modbus RTU Master에서 Slave로 패킷을 송신하였는데, 타임아웃 설정시간 동안 안오면 타임아웃으로 간주합니다. 타임아웃이 발생하면 패킷을 재송신하지 않고 다음 패킷을 송신합니다.

지연시간은 Modbus RTU Slave에서 Master로 패킷이 회신되어 정상적으로 통신을 마친 이후에 Master에서 다음 패킷을 송신하기 까지의 시간을 지연시킵니다. Slave에서 너무 빠른 통신으로 수신된 데이터를 버퍼에서 지우지 못할 경우, 에러가 발생합니다. 1(Master):1(Slave) 통신에서는 지연시간은 필요 없을 가능성이 높습니다. 다만, 1:N 통신일 경우, 필요합니다.

3.3. 통신 리스트 추가

추가 버튼을 눌러서 통신 리스트를 추가합니다.

- **모드버스 코드** : Modbus의 통신 패킷중 Slave에게 명령할 명령어의 종류입니다. FC01과 FC02와 같이 같은 목적이지만 명령어가 다른 것은, Slave 장치에서 FC01만을 사용할 수도 있고 FC02만을 사용할 수도 있고 FC01과 FC02 모두 사용할 수도 있습니다. 또한 FC01로 엑세스 되는 메모리와 FC02로 엑세스 되는 메모리가 다를 수 있습니다. 이 부분은 Slave 장치의 사용설명서를 참조해 주시기 바랍니다.

모드버스 코드	펑션 코드	설 명
BIT READ (FC01)	0x01	여러개의 비트의 정보를 Slave에게 요청합니다.
BIT READ (FC02)	0x02	여러개의 비트의 정보를 Slave에게 요청합니다.
WORD READ (FC03)	0x03	여러개의 워드의 정보를 Slave에게 요청합니다.
WORD READ (FC04)	0x04	여러개의 워드의 정보를 Slave에게 요청합니다.
BIT WRITE (FC05)	0x05	한 개의 비트를 변경하라고 Slave에게 요청합니다.
WORD WRITE (FC06)	0x06	한 개의 워드를 변경하라고 Slave에게 요청합니다.
MULTIPLE WORD WRITE (FC16)	0x10	여러개의 워드를 변경하라고 Slave에게 요청합니다. (명령어코드의 해석은 여러개의 워드를 변경하는 것이지만 여기서는 한 개의 워드만 변경하는 것으로 동작됩니다)

- **국번** : 통신할 Slave 장비의 슬레이브 어드레스를 입력합니다.

- **주소** : Slave 장비의 엑세스할 메모리 주소값을 입력합니다. Slave 장비마다 사용하는 주소와 주소에 매칭된 메모리의 정보가 다릅니다.

- **바인딩** : 바인딩할 MP STUDIO의 메모리 정보를 입력합니다.
  BIT READ : Slave 장비로부터 읽어온 비트를 저장할 메모리를 입력합니다. (M0, M1...)
  BIT WRITE : Slave 장비에 쓰기할 비트의 값으로 참조할 메모리를 입력합니다. (M0, M1...)
  WORD READ : Slave 장비로부터 읽어온 워드를 저장할 메모리를 입력합니다. (D0, D1...)
  WORD WRITE : Slave 장비에 쓰기할 워드의 값으로 참조할 메모리를 입력합니다. (D0, D1...)

※ MULTIPLE WORD WRITE (FC16)을 사용해도 여러개의 워드를 Slave 장비로 쓰기하지 않고 한 개의 워드만 쓰기합니다. FC16 명령어 코드는 Slave 장비가 WORD WRITE(FC03 , FC04)를 허용하지 않고 FC16만을 허용하는 장비가 있어서 추가하게 되었습니다.

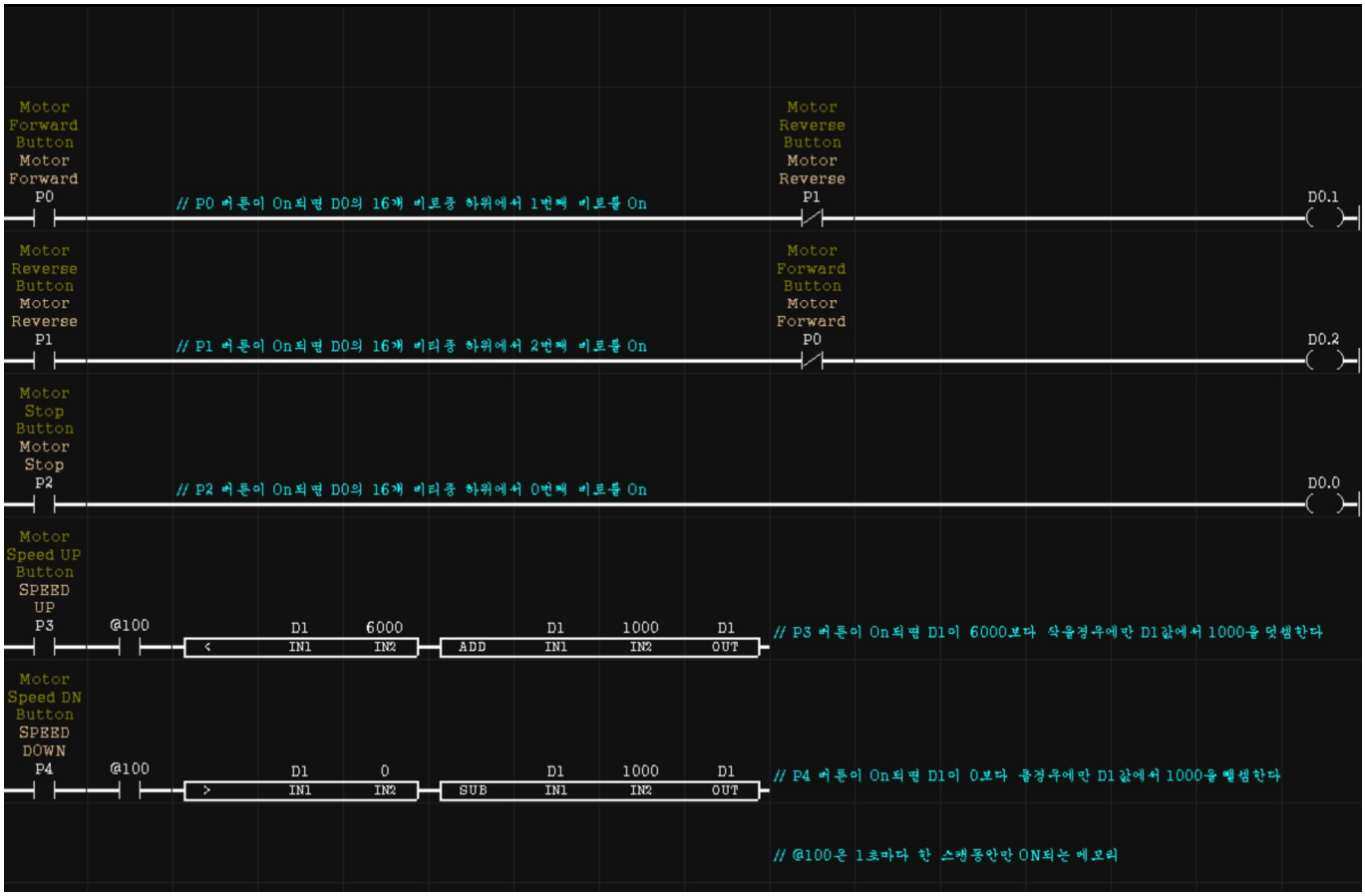
4. S100 인버터 통신

LS산전의 S100 인버터와 통신하는 예시입니다.

아래는 S100 인버터의 Modbus 통신맵입니다.

통신 번지	파라미터	스케일	단위	R/W	비트별 할당 내용	
0h0000	인버터 모델	-	-	R	6: S100	
0h0001	인버터 용량	-	-	R	0: 0.75kW, 1: 1.5kW, 2: 2.2kW 3: 3.7kW, 4: 5.5kW, 5: 7.5kW 6: 11kW, 7: 15kW, 8: 18.5kW 9: 22kW, 10: 30kW, 11: 37kW 12: 45kW, 13: 55kW, 14: 75kW 256 : 0.4kW, 257 :1.1kW, 258:3.0kW 259: 4.0kW	
0h0002	인버터 입력 전압	-	-	R	0: 220V급, 1: 440V급	
0h0003	버전	-	-	R	(예제) 0h0100: Version 1.00 0h0101: Version 1.01	
0h0004	Reserved	-	-	R/W	-	
0h0005	목표 주파수	0.01	Hz	R/W	-	
0h0006	운전 지령(옵션)	-	-	R	B15	Reserved
					B14	0: Keypad Freq
					B13	1: Keypad Torq
					B12	2~16: 단자대 다단속
					B11	17: Up, 18: Down
					B10	19: STEADY
					B9	22: V1, 24: V2, 25: I2 26: Reserved 27: 내장형 485 28: 통신 옵션 30: JOG, 31: PID
					B8	0: Keypad
					B7	1: Fx/Rx-1
					B6	2: Fx/Rx-2 3: 내장형 485 4: 통신 옵션
				R/W	B5	Reserved
					B4	비상 정지
					B3	W: Trip 초기화(0 →1) R: Trip 상태
					B2	역방향 운전(R)
					B1	정방향 운전(F)
					B0	정지(S)

아래는 MP STUDIO에서 S100 인버터를 제어하기 위해 코딩한 래더로직입니다.



P0: 정회전 모터 시작버튼, P1: 역회전 모터 시작버튼, P2: 모터 정지버튼
P3: 모터 주파수 10Hz 상승버튼(가속버튼), P4: 모터 주파수 10Hz 하강버튼(감속버튼)

모터는 정회전과 역회전이 동시에 작동하면 안되므로 B접점으로 인터락 접점을 P0버튼쪽에는 P1을 P1버튼쪽에는 P0를 만들었습니다.

D0.0이 On되면, D0의 값은 1이 됩니다. (B0)
D0.1이 On되면, D0의 값은 2가 됩니다. (B1)
D0.2가 On되면, D0의 값은 4가 됩니다. (B2)

A접점으로 @100은 1초마다 한 스캔동안만 On이 되는 접점입니다. @100 접점이 없으면 너무 빠르게 주파수가 상승/하강하기 때문에 1초마다 10hz가 상승/하강 하도록 하기위해 사용되었습니다.

Modbus RTU Master 설정은 아래와 같습니다.

WORD WRITE(FC06) 으로 D0 메모리와 D1 메모리를 S100 모드버스 통신맵에 있는 5번지(목표주파수) 와 6번지(운전지령)으로 쓰기 합니다.

WORD READ(FC03) 으로 S100 인버터의 현재 목표주파수 및 운전지령 상태값을 각각 D10, D11로 읽어와서 상태를 확인할 수 있습니다.

5. 실수 값 읽기

중국에서 저렴한 가격으로 구매한 전력량계로 테스트 해보았습니다.

A상, B상, C상의 전압을 읽이 위한 전력량계의 모드버스 주소는 아래와 같습니다.

주소	프로젝트
9, 10 Ua(3상 4선식) A상 전압	
11, 12 Ub(3상 4선식) B상 전압	
13, 14 Uc(3상 4선식) C상 전압	

A상 전압은 모드버스 주소 9,10번을 사용합니다. 32비트 실수(IEEE-574)를 사용하므로 2개의 워드 메모리를 가지고 있기 때문에 주소값이 2개입니다.

B상 전압은 모드버스 주소 11,12번을 사용합니다.

C상 전압은 모드버스 주소 13,14번을 사용합니다.

Modbus RTU Master 설정은 아래와 같습니다.

이렇게 동작을 해보니 이상한 값이 표시되었습니다. 국내에서 구입한 다른 제품은 잘 표기가 되던데 이상했습니다. 래더 모니터링으로 비트로 쪼개서 읽어보니 실수 메모리의 상위 메모리와 하위 메모리가 서로 바뀌어있습니다. 전력량계에서 보내주는 워드의 순서가 다른 것입니다. 그래서 래더 프로그램을 아래와 같이 하였습니다.

R0의 상위 워드와 하위 워드를 서로 교환하여 R1에 저장하여 올바른 값을 읽어올 수 있습니다.

6. 통신실패 확인하기

Modbus RTU Slave가 되는 상대방 기기와 통신이 정상적으로 이루어지지 않았을 경우, 통신이 실패되었는지 확인할 수 있습니다.

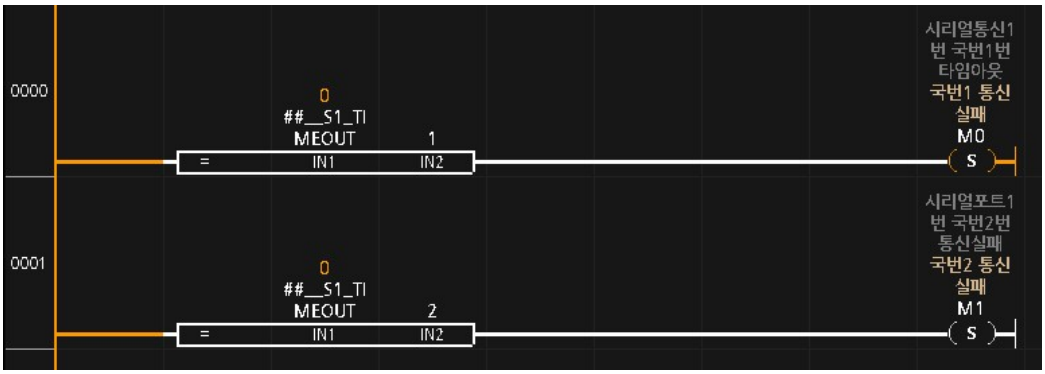
아래와 같이 슬레이브 제품 2대(국번이 1번과 2번)와 통신을 할 경우를 예를 들어보겠습니다.

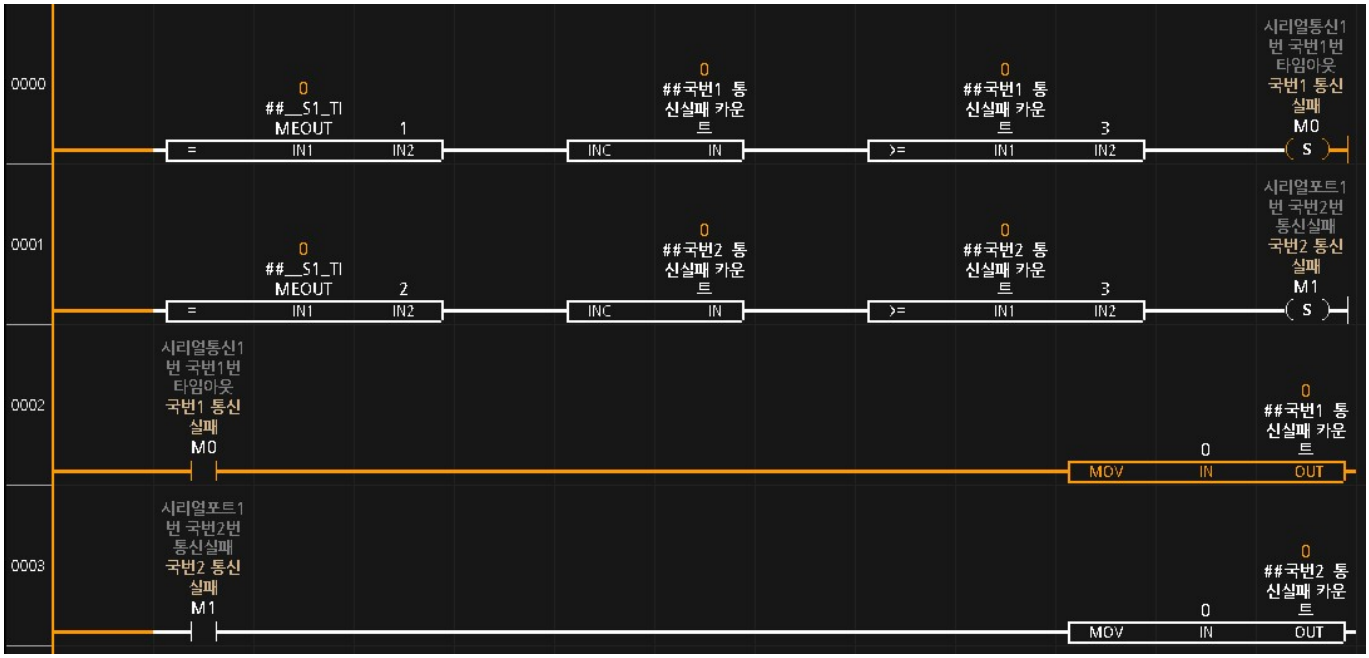
Serial Port 1번이 통신이 실패될 경우, 유저 메모리 워드값인 ##__S1_TIMEOUT 에 통신이 실패된 국번 값이 입력됩니다. 하지만, 통신이 실패한 다음 스캔에서 0으로 초기화 되므로 모니터링으로 그 값을 확인할 수는 없습니다.

Serial Port 2번이 통신이 실패될 경우, ##__S2_TIMEOUT 에 통신이 실패된 국번 값이 입력됩니다.

Serial Port 3번이 통신이 실패될 경우, ##__S3_TIMEOUT 에 통신이 실패된 국번 값이 입력됩니다.

펄스처럼 한 스캔만 통신이 실패된 국번 값이 입력되므로 아래처럼 타임아웃을 프로그램할 수 있습니다. Serial Port 1번의 1번 국번 장비와 통신이 실패될 경우, M0가 SET 되는 프로그램입니다.



M0(국번1 통신 실패)가 SET 되면 RESET을 하지 않는한 계속 통신 실패가 되고 있으므로, 특정 조건(푸쉬버튼 누름, HMI에서 리셋버튼 누름)으로 M0를 RESET 하는 프로그램을 하셔야 합니다.

위처럼 프로그램 할 경우, 단 한번의 통신실패에도 통신실패가 되므로, 통신실패가 3회 실패시 M0가 SET이 되게 아래처럼 프로그램도 가능합니다.