

Ezi-MOTIONGATE

PROFIBUS-DPV1 **PLC 예제 (MELSEC-Q)**



www.fastech.co.kr

- 목 차 -

1. 안전상의 주의사항	6
2. 제품 사양 및 크기	9
2.1 개요	9
2.2 네트워크 별 제품 특성 표	10
2.3 제품 외형	11
2.3.1 제품 치수	11
2.3.2 각 부의 명칭	12
2.3.3 FASTECH RS485 통신 접속 커넥터	13
2.3.4 RS485 보오레이트 설정 및 종단저항 선택 스위치 (SW4)	14
2.3.5 상태 표시 LED (LED1..LED4)	15
2.3.6 네트워크 국번 설정 (SW1, SW2)	18
2.3.7 전원 커넥터(DC POWER)	18
3. 설치 및 결선 방법	19
3.1 시스템 구성도	19
3.2 PROFIBUS 네트워크의 결선	20
4. 동작 원리	21
4.1 PROFIBUS 네트워크의 모션게이트 시스템 개요	21
4.1.1 PROFIBUS 의 개요	21
4.1.2 모션게이트의 개요 및 네트워크 구성	23
4.2 제어명령 비트영역(Input Map)과 상태정보 비트영역(Output Map)	26
Input-Map 의 비트 구성	26
Output-Map 의 비트 구성	30
4.3 IO Map 의 동작 순서 및 동작 조건	33
4.3.1 IO-Map 의 비트명령 방법	33

4.3.2 IO-Map의 제어 명령 준비 순서	35
5. IO-Map의 사용 예제	37
5.1. MELSOFT GX-developer 프로젝트 설정	37
5.2. PLC LADDER 프로그래밍 예제	52
■ 데이터 교환 버퍼 영역	52
■ PROFIBUS Buffer 주소의 할당	52
■ PLC PROFIBUS 마스터 모듈의 데이터 교환 시작 명령	54
■ CONNECT	55
■ ENABLE 명령과 E-ETOP 명령	57
■ ALARM 상태 확인	59
■ CANCEL 명령	61
■ HOLD 명령	63
■ RESPONSE TYPE 설정	65
■ 상태 정보 확인	71
■ CMD START 명령	75
5.2.1 조그 운전 명령	78
■ JOG Move – Speed Step Move or Speed Ratio Move	78
■ JOG Move – Speed Value Move	82
5.2.2 스텝 이동 명령	85
5.2.3 영점 이동 명령	88
5.2.4 위치 이동 명령	91
■ 상대 위치 이동	91
■ 절대 위치 이동	94
※ 위치 이동 명령에 대한 응답확인	97
5.2.5 PT 운전 명령	98
■ 일반 PT 운전	98
■ 싱글 PT 운전	101
■ PT 운전 명령에 대한 응답확인	104
5.2.6 원점 이동 명령	105
5.2.7 파라미터 설정	107
■ 파라미터 확인 명령	107
■ 파라미터 변경 명령	110
■ 파라미터 저장	113

5.2.8 위치 지정.....	115
5.2.9 알람 내역 요청 및 초기화.....	118
■ 알람 내역 확인.....	118
■ 알람 내역 초기화	121
5.2.10 모션게이트 버전 확인.....	123

예제 목차

예제 1. PROFIBUS 네트워크 데이터교환 승인 프로그램.....	54
예제 2. 각 축의 활성화 명령 및 응답 비트 확인.....	55
예제 3. 모터 활성화 명령과 비상정지 명령.....	57
예제 4. 알람상태 확인 및 알람해제 명령.....	59
예제 5. 실행취소 명령.....	61
예제 6. Axis - 0 의 일시정지 명령.....	63
예제 7. 응답 데이터 설정.....	65
예제 8. 응답 데이터 설정.....	71
예제 9. 모션 모드의 CMD START 실행 방법.....	75
예제 10. 설정 모드의 CMD START 실행 방법.....	76
예제 11. Speed Step Move 또는 Speed Ratio Move 형태의 조그 운전 명령.....	78
예제 12. Speed Value Move 형태의 조그 운전 명령.....	82
예제 13. 스텝 이동 명령.....	85
예제 14. 영점이동 명령.....	88
예제 15. 상대위치 이동 명령.....	91
예제 16. 절대위치 이동 명령.....	94
예제 17. 위치이동 명령에 대한 응답 확인 방법.....	97
예제 18. PT 운전 명령.....	98
예제 19. 싱글 PT 운전 명령.....	101
예제 20. 위치이동 명령에 대한 응답 확인 방법.....	104
예제 21. 원점 이동 명령.....	105
예제 22. 파라미터 확인 명령.....	107
예제 23. 파라미터 변경 명령.....	110
예제 24. 파라미터 저장 명령.....	113
예제 25. 위치값 변경 명령.....	115
예제 26. 알람 내역 요청.....	118
예제 27. 알람 내역 초기화 명령.....	121
예제 28. 모션게이트의 버전 정보 요청 명령.....	123

※ 사용하시기 전에 ※

- 파스텍 Ezi-MOTIONGATE 를 구입해 주셔서 대단히 감사합니다.
- Ezi-MOTIONGATE 는 32bit 고성능 ARM 프로세서를 탑재한 Fieldbus to FASTECH protocol Gateway Unit 입니다.
- 이 사용자 설명서에는 Ezi-MOTIONGATE 의 취급 방법, 안전상의 주의 사항, 이상진단과 처치방법 및 사양 등이 기재 되어있습니다.
- 사용자 설명서를 잘 이해하신 후에 Ezi-MOTIONGATE 를 안전하게 사용하여 주십시오.
- 사용자 설명서를 다 읽으신 후에는 본 제품을 사용하는 사람이 언제든지 볼 수 있도록 잘 보관해 주십시오.

1. 안전상의 주의사항

◆ 일반 주의사항

- 사용자 설명서는 제품 개선이나 사양 변경 또는 사용자 설명서 자체를 이해하기 쉽게 하기 위하여 고지 없이 변경 될 수 있습니다.
- 사용자 설명서를 손상 또는 분실해서 새로 주문할 경우에는 구입하신 대리점이나 본사로 문의해 주십시오.
- 사용자 임의로 제품을 개조하는 것은 당사의 보증 범위 밖이므로 당사에서 책임지지 않습니다.

◆ 안전 주의사항

- 설치, 운전, 점검, 보수 등을 하기 전에는 반드시 사용자 설명서를 읽어서 그 내용을 충분히 숙지하신 후에 실시 해 주십시오, 또한, 기계에 관한 지식, 안전에 관한 정보나, 주의사항을 충분히 숙지하신 후 제품을 사용하여 주십시오.
- 사용자 설명서는 안전에 관한 주의사항의 정도를 **주의**와 **경고**로 구분하여 기재하고 있습니다.




주의: 잘못 취급했을 경우 위험한 상황을 초래하여 중상 또는 경상을 입을 가능성이 있는 경우, 그리고 대물 손해만이 발생할 가능성이 있는 경우





경고: 잘못 취급 하였을 경우 전기 감전 등의 위험한 상황을 초래하여, 사망 또는 중상을 입을 가능성이 있는 경우

- 기재된 내용 중에 주의에 해당하는 것일지라도, 상황에 따라서 중대한 결과를 야기시킬 가능성이 있습니다. 반드시 지켜 주십시오.



◆ 제품 상태

 주의	<p>제품이 손상되어 있거나 또는 부품이 빠져있는지 확인하십시오. 비정상적인 제품을 설치, 운전할 경우 기계파손 또는 부상의 위험이 있습니다.</p>
---------------------------------------------------------------------------------------------	-----------------------------------------------------------------------------------------

◆ 설치

 주의	<p>운전 시에는 충분히 주의하십시오 떨어지면 제품이 파손되거나, 발에 떨어지면 부상의 위험이 있습니다.</p> <p>제품을 취급할 장소에는 금속 등 불연 물질을 사용하여 주십시오. 화재가 날 위험이 있습니다.</p> <p>여러 대의 Ezi-MOTIONGATE를 하나의 밀폐된 공간에 설치할 때는, 냉각장치 등을 설치하시어 주위 온도가 50℃이하가 되도록 해주십시오. 과열로 화재 또는 그 밖의 사고로 이어질 위험이 있습니다.</p>
 경고	<p>설치, 접속, 운전, 조작, 점검 및 고장 진단 작업은 적합한 자격을 가진 사람이 실시하여 주십시오. 화재, 부상, 장치 파손의 원인이 됩니다.</p>

◆ 배선

 주의	<p>드라이브의 전원 입력 전압은 정격 범위를 반드시 지켜 주십시오. 화재 및 고장의 원인이 됩니다.</p> <p>접속은 배선도에 따라 확실히 실시하여 주십시오. 화재 및 오작동의 원인이 됩니다.</p>
 경고	<p>입력 전원이 OFF 되어 있는 것을 확인한 후에 작업해 주십시오. 감전 또는 화재의 위험이 있습니다.</p> <p>본 Ezi-MOTIONGATE 케이스는 콘덴서에 의해 내부회로의 GND와 절연되어 있으므로, 반드시 접지를 시켜주십시오. 감전 또는 화재의 위험이 있으며, 제품 오작동의 원인이 됩니다.</p>

◆ 운전 및 설정 변경

**주의**

드라이브의 보호기능이 작동하면 원인을 제거한 후에 보호 기능을 해제하여 주십시오.

원인을 제거하지 않고 운전을 계속하면 모터 및 드라이브가 오작동되어 부상, 장치 파손의 원인이 됩니다.

드라이브에 전원을 투입할 때에는 드라이브의 제어 입력을 모두 OFF로 한 후에 투입하여 주십시오.

모터가 가동되어 부상, 장치파손의 원인이 됩니다.

본 Ezi-MOTIONGATE 의 모든 값들은 출하 시 적절히 설정해 놓았습니다. 설정 변경 시에는 충분히 사용자 설명서를 숙지한 후 변경해 주십시오. 기계가 파손되거나 제품이 고장 날 수 있습니다.

◆ 보수 및 점검

**경고**

본 Ezi-MOTIONGATE는 주 회로 전원을 차단한 후, 충분히 시간이 경과한 후에 보수, 점검을 해주십시오.

콘덴서 전원이 남아 있으므로, 감전 등의 위험이 있습니다.

통전 중에는 배선 변경을 하지 마십시오.

감전 또는 제품 파손, 기계파손 등의 위험이 있습니다.

제품의 개조는 절대로 하지 마십시오.

감전 또는 제품파손, 기계파손 등의 위험이 있으며, 해당 제품은 당사의 A/S를 받을 수 없습니다.

설치 시 주의사항.

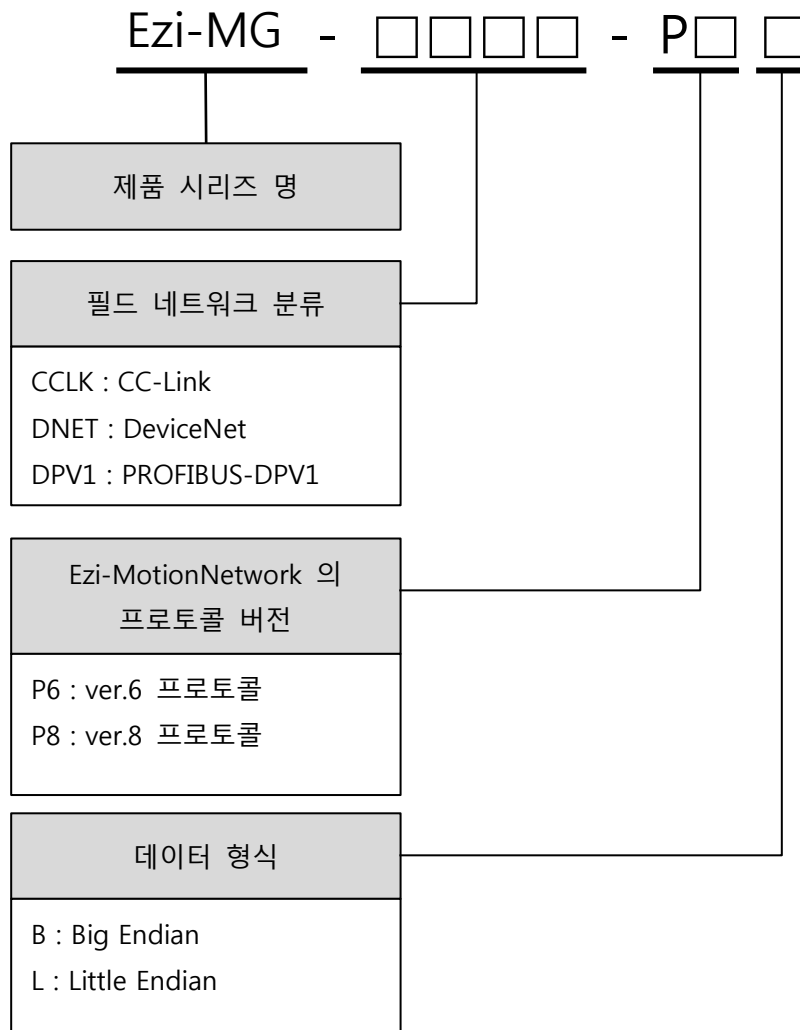
- 1) 실내에서 사용해야 하며, 실내 주의온도는 0°~55°C 에서 사용해야 합니다.
- 2) 케이스가 50°C 이상이 되면 외부에 방열을 시켜주어야 합니다.
- 3) 직사광선, 자석물체, 방사선 물체는 피해서 설치해 주어야 합니다.
- 4) 드라이브를 2대 이상 나란히 설치 시에는 수직방향은 20mm 이상, 수평방향은 50mm 이상 거리를 두고 설치해 주어야 합니다.

2. 제품 사양 및 크기

2.1 개요

- Ezi-MotionGate(이하 모션게이트)는 산업용 네트워크에서 슬레이브로 연결하여, FASTECH RS485 로 구성된 모션 드라이브를 마스터로 제어하는 모션 게이트웨이 장치 입니다.
- 모션게이트를 슬레이브로 연결하여 사용 가능한 가능한 최대 수량은 적용되는 산업용 네트워크에서 지원되는 수량입니다.
- 모션게이트에서 연결 가능한 모터 드라이브(Axis)는 적용되는 산업용 네트워크 마다 최대 0~15 까지의 모터 ID 를 부여할 수 있습니다.

2.2 제품 형명



2.3 제품 특성 표

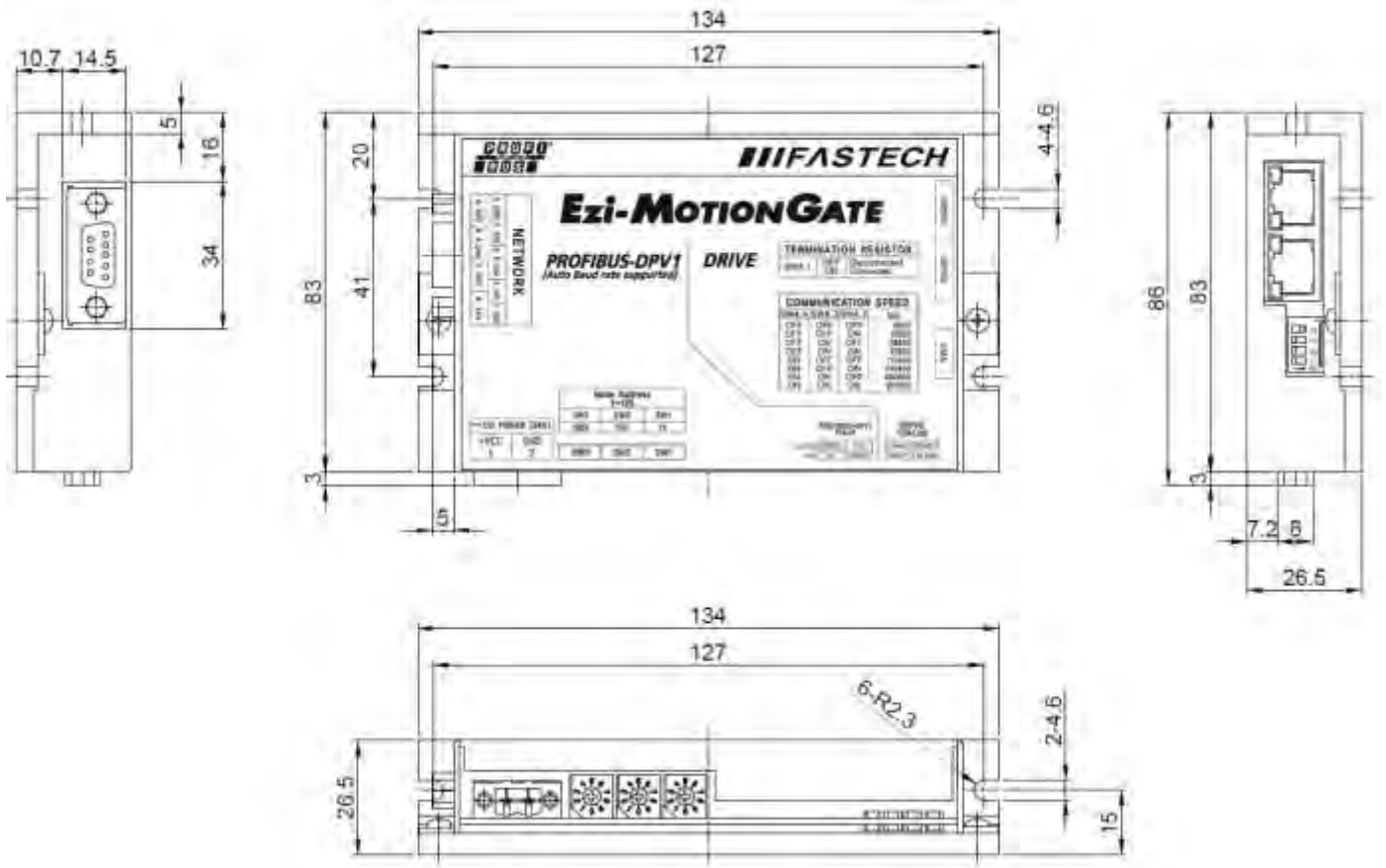
네트워크		PROFIBUS								
입력전압		24VDC ±10%								
제어방식		산업용 네트워크의 I/O 데이터를 사용하여 다축 제어가 가능한 모션 게이트웨이								
다 축 제어		1 Station – 9 Axis								
소비전류		최대 500mA								
환경	온도	사용 : 0~55℃ 보관 : -27~70℃								
	습도	사용 : 35~85℃ (결로는 없을 것) 보관 : -10~90℃(결로는 없을 것)								
	내 진동	0.5G								
기능	스위치 선택	네트워크 국번(Station No.) 설정								
	LED 표시	네트워크 이상, 마스터 연결 이상, 드라이브의 Servo On 상태, 드라이브의 알람 상태								
특수기능	조그 제어	4-Speed Step 모드, Speed Ratio 모드								
	스텝 이동 제어	4-Step Distance								
	통신 기능	Ezi-STEP Plus-R, Ezi-SERVO Plus-R series								
FASTECH RS-485		Baud-Rate (bps)	9600	19200	38400	57600	115200	230400	460800	921600
		케이블 길이 (m)	1200	1150	1100	1000	1000	880	550	300
		RJ-45 커넥터 LED	YELLOW : RS485 송신 상태 (TX from MOTIONGATE) GREEN : RS485 수신 상태 (RX to MOTIONGATE)							

NOTE 1: 케이블의 길이는 네트워크 상태가 최적일 때의 최대 연결거리입니다.

NOTE 2: 산업용 네트워크의 통신규약을 준수 합니다.

2.4 제품 외형

2.4.1 제품 치수

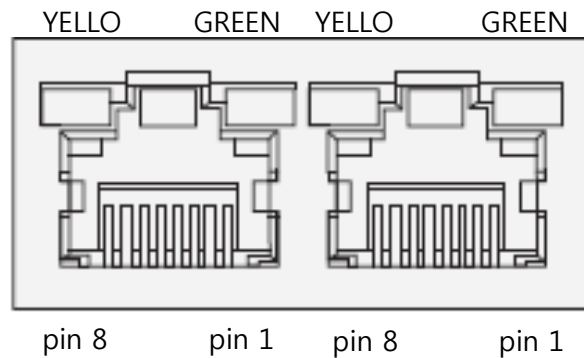


2.4.2 각 부의 명칭



2.4.3 FASTECH RS485 통신 접속 커넥터

통신 커넥터는 RJ-45 를 사용하여 연결합니다.



통신 커넥터의 핀 맵 (RS485)

핀 번호	기능
1, 2, 4, 5, 7, 8	GND
3	DATA +
6	DATA -
CASE	Frame GND


통신 커넥터의 LED

디스플레이	색상	발광 형태
RS485 TX	녹색	RS485 데이터를 송신 시 점멸
RS485 RX	황색	RS485 데이터를 수신 시 점멸

2.4.4 RS485 보오레이트 설정 및 종단저항 선택 스위치 (SW4)

SW4 는 모터 드라이브와 연결된 RS-485 통신 네트워크의 보오레이트(Baud-Rate : 통신속도) 설정을 위한 스위치 입니다. 만약 모션게이트가 네트워크 segment 의 가장 끝 단에 연결된 상태일 경우, 종단 저항을 사용 여부를 결정할 수 있습니다.

SW4.1 은 종단 저항의 사용을 결정하며, SW4.2~SW4.4 는 다음과 같이 통신속도를 설정 하는데 사용됩니다.

SW4.1	SW4.2	SW4.3	SW4.4	Speed baud[bps]	<p>*1 : 초기 설정 값</p>  <p>SW4.1 가 OFF : 종단 저항이 OFF 상태. SW4.1 가 ON : 종단 저항이 ON 상태.</p>
X	OFF	OFF	OFF	9600	
X	ON	OFF	OFF	19200	
X	OFF	ON	OFF	38400	
X	ON	ON	OFF	57600	
X	OFF	OFF	ON	115200 *1	
X	ON	OFF	ON	230400	
X	OFF	ON	ON	460800	
X	ON	ON	ON	921600	

⚠ 주의

하나의 RS485 네트워크에 연결된 드라이브 모듈들의 통신 속도 설정 값은 모두 동일한 값으로 지정해 주어야 합니다.

2.4.5 상태 표시 LED (LED1~LED4)

상태 표시 LED 는 상황에 따라 LED1~LED4 를 동시에 동작 하거나, 단독적으로 동작하여 상태를 표시합니다.

LED 동작 표시 (LED1~LED4)

LED 번호	동작 상태	설명
LED1 LED2 LED3 LED4	소등	전원 OFF, 타임 아웃 상태, 네트워크 미 접속
	녹색, 적색 동시 점등	모션게이트에 전원공급으로 부팅되는 상태 국번 지정 스위치 또는 네트워크 통신 속도 선택 스위치를 변경 하여 재 부팅되는 상태 * 녹색과 적색이 동시 점등하므로 LED 의 색상은 주황색으로 보입니다.
	녹색 동시 점멸	모션게이트의 자기 진단 상태. ✓ 커넥터 미 연결 ✓ 잘못된 네트워크 보오레이트 설정 ✓ 잘못된 네트워크 국번 지정
LED3 LED4	녹색, 적색 동시 점등	모션게이트의 네트워크 장치가 인식 불가 상태 * 본사 또는 대리점으로 연락하여 조치를 취하십시오.

드라이브 상태 표시 LED (DRIVE STATUS)

LED 번호	LED 상태 정보	LED 이름	동작 상태	설명
LED1	드라이브의 접속 상태	ENABLE (녹색)	점등	CONNECT 된 드라이브 중 모터가 활성화 됨.
			점멸	CONNECT 된 드라이브 중 1 개 이상의 드라이브의 모터가 ENABLE 되지 않음
	드라이브 알람	ALARM (적색)	점멸	1 개 이상의 드라이브가 알람을 발생한 상태
LED2	모터 드라이브 연결 양호	CONNECT (녹색)	점등	모터 드라이브가 CONNECT 명령이 실행되어 모션게이트와 정상적으로 통신 중
		ERROR (적색)	소등	
	드라이브 접속 상태 오류	CONNECT (녹색)	소등	RS-485 네트워크에 연결된 모터 드라이브가 CONNECT 명령이 실행된 모터 드라이브가 없음. 모터 드라이브와 모션게이트가 통신중인 상태가 아님.
		ERROR (적색)	점등	드라이브와 통신되지 않음 RS485 네트워크의 통신연결이 끊김 보오레이트 설정 오류
	통신 에러	CONNECT (녹색)	점등	모터 드라이브와 통신 오류 발생 (CRC 에러 발생)
		ERROR (적색)	무작위 점멸	
	다축 연결 상태의 통신 에러	CONNECT (녹색)	점등	RS485 에 연결된 하나 이상의 모터드라이브에 CONNECT 명령에 응답 하지 않음. ✓ 네트워크의 연결이 끊김 ✓ 토폴로지 구성에 이상이 있음. ✓ 없는 모터드라이브의 IO-Map 영역에 CONNECT 명령을 실행함.
		ERROR (적색)	점등	

**주의**

드라이브 상태 표시 LED는 모션게이트와 드라이브의 통신 상태를 점검 후 모터 활성화 상태를 확인하도록 하십시오.

PROFIBUS 네트워크 상태 표시 LED (PROFIBUS-DP1 STATUS)

번호	LED	동작 상태	내용	조치
LED 3	RUN (GREEN)	소등	초기 값이 없음.	상위 제어기의 PROFIBUS 의 초기화가 되지 않음.
		점등	초기화 됨	상위 제어기와 연결되어, 정상적인 데이터의 교신 상태입니다.
		점멸	자기 진단 상태	초기화가 되었으나, 자가 진단 상태입니다.
	ERROR (RED)	점등	예외 적인 오류 발생	모션게이트의 허용버퍼를 초과 하였습니다. 상위 제어기의 송수신 버퍼를 확인 바랍니다.
LED 4	POEWR (GREEN)	소등	전원 OFF	전원 상태를 확인하십시오
			네트워크 미 접속	PROFIBUS 네트워크 케이블의 연결 상태 및 상위 제어기의 상태를 확인 확인하십시오
		점등	전원 ON	정상적인 동작 상태입니다.
	BF (RED)	점멸 1	대기 상태	상위 제어기에 대한 통신을 기다리는 상태 입니다.
		점멸 2	파라미터 설정 오류	상위 제어기의 파라미터 설정 값을 확인하십시오.

**주의**

네트워크 상태 LED는 산업용 네트워크의 규약으로 인하여 네트워크의 상태에 따라 정해진 표현으로 동작합니다.

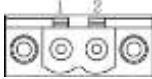
2.4.6 네트워크 국번 설정 (SW1, SW2)

PROFIBUS 네트워크의 국번(상위제어기 마다 Node No, FDL Address, Station No 등으로 표기됩니다)를 설정하는 로터리 스위치로, 네트워크에서 설정 하고자 하는 국번으로 지정 할 수 있습니다. 모션게이트에서 지원 하는 PROFIBUS 에서 지원하는 국번의 범위는 0~125 입니다.

스위치 값 (SW1)	ID번호 X100 (100자리)	스위치 값 (SW1)	ID번호 X10 (10자리)	스위치 값 (SW2)	ID번호 X1 (1자리)	
0	000	0	00	0	0	
1	100	1	10	1	1	
2	N.C	2	20	2	2	
3	N.C	3	30	3	3	
4	N.C	4	40	4	4	
5	N.C	5	50	5	5	
6	N.C	6	60	6	6	
7	N.C	7	70	7	7	
8	N.C	8	80	8	8	
9	N.C	9	90	9	9	

2.4.7 전원 커넥터(DC POWER)

전원 공급을 위한 커넥터 입니다.

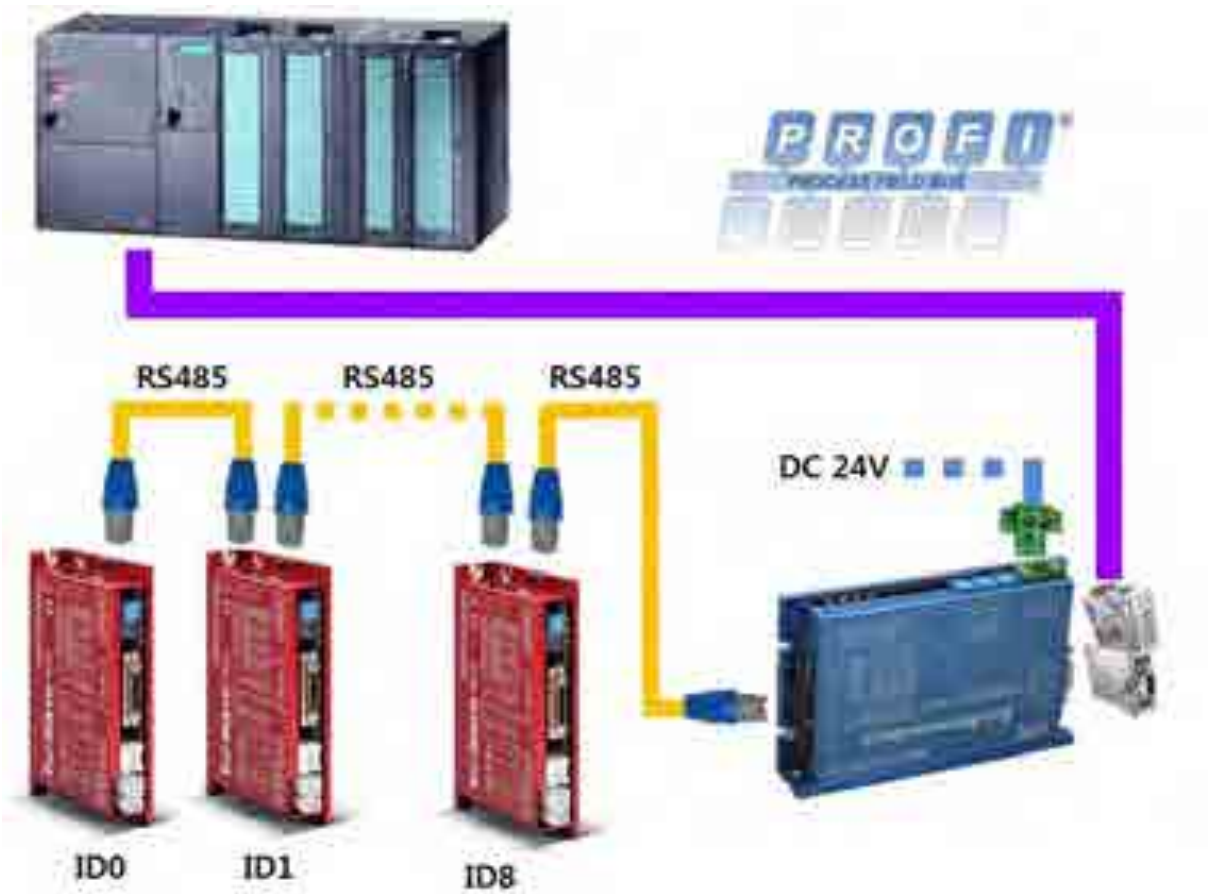
번호	기능	핀 배치도
1	입력전원 : 24VDC \pm 10%	
2	입력전원 : GND	

3. 설치 및 결선 방법

3.1 시스템 구성도



모션게이트의 배선도

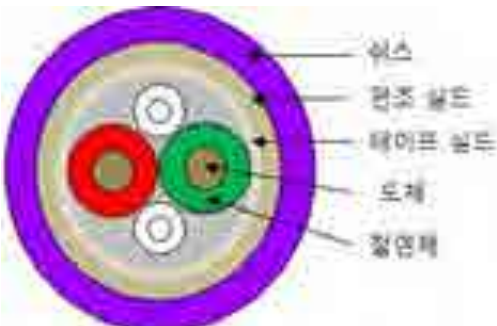



3.2 PROFIBUS 네트워크의 결선

PROFIBUS DP 커넥터의 핀 맵

번호	기능	입/출력	핀 배치도
1	NC	NC	 <p>*모션게이트의 NETWORK 단자입니다.</p>
2	NC	NC	
3	B Line	입/출력	
4	RTS.	입/출력	
5	GND	입력	
6	5v	전원 출력(60mA)	
7	NC	NC	
8	A Line	NC	
9	NC	NC	

PROFIBUS DP 케이블의 구조

PROFIBUS DP 케이블	PROFIBUS DP 커넥터						
 <table border="1"> <thead> <tr> <th>심선 종류</th><th>데이터 명칭</th></tr> </thead> <tbody> <tr> <td>녹색</td><td>A Line</td></tr> <tr> <td>적색</td><td>B Line</td></tr> </tbody> </table>	심선 종류	데이터 명칭	녹색	A Line	적색	B Line	
심선 종류	데이터 명칭						
녹색	A Line						
적색	B Line						


주의

네트워크 결선 시 PROFIBUS DP 전용 커넥터와 전용 케이블 사용을 권장 합니다.

4. 동작 원리

4.1 PROFIBUS 네트워크의 모션게이트 시스템 개요

4.1.1 PROFIBUS 의 개요

PROFIBUS 는 SIEMENS 를 중심으로 개발 발전되어온 EN 50170 과 EN 50254 에 명시 된 개방형 필드버스(Fieldbus)입니다. 이 네트워크는 서로 다른 제조사의 디바이스간 특별한 인터페이스가 없이 통신이 가능하도록 하여 주며 마스터와 슬레이브 형태로 디바이스를 구성합니다.

마스터 디바이스는 버스상의 데이터 통신을 결정하고, 버스 액세스 권한이 정지되었을 때 내부 요청 없이 메시지를 보낼 수 있으며, 항상 동작중인 스테이션(Station)의 부름을 받습니다.

슬레이브 디바이스는 I/O 디바이스, 드라이브 디바이스, 측정장비와 같은 주변장치들을 말합니다. 이들은 버스 액세스 권한이 없으며, 오직 네트워크상의 메시지로 수신되는 마스터의 요청이 있을 때 응답 메시지를 보내는 역할만을 수행합니다.

모션게이트는 PROFIBUS 네트워크 상에서 슬레이브 디바이스로서 마스터로부터 명령 메시지를 수신 받아 응답하는 장치 입니다.

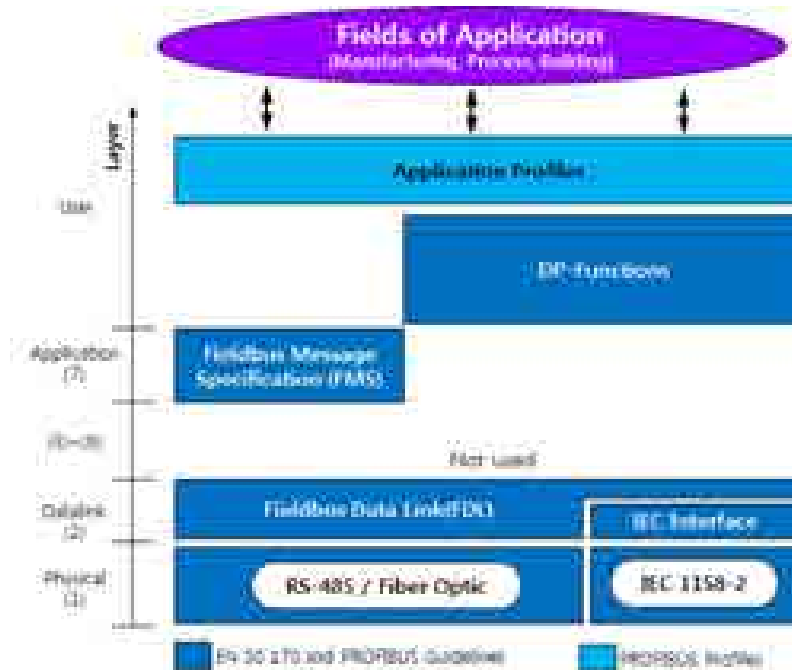
PROFIBUS 는 DP, FMS, PA 세 종류가 있습니다.

DP(Decentralized Peripherals)는 실시간 처리의 능력을 갖는 빠른 속도 보장으로 주변장치와의 연결 시에 비용 부담의 절감효과로 특히 Field Level 안에서 자동화 시스템과 주변장치 사이의 통신을 위해 널리 이용되고 있습니다. 기존의 전압신호(24V), 전류 신호(0~20mA) 전송을 대체 하는데 적합합니다.

FMS(Fieldbus Message Specification)는 Cell Level 의 통신에 적용하기 위해 개발된 통신 방식으로 여러 대의 마스터를 가진 시스템(Multi master-System)에서 Cell 상호간의 통신을 위한 Backbone 으로 응용됩니다. FMS 는 여러 가지 응용과 폭넓은 유연성을 제공하므로 다양한 통신이 필요한 곳에 많이 사용됩니다.

PA(Process Automation)는 화공분야에 연계되어 폭발의 위험이 존재하는 영역에서 Sensor 와 Transmitter 의 연결이 가능한 고유 안전성 기능을 갖습니다. 또한 국제표준 IEC 1158-2 에 준해 한 개(두 가닥 선)의 Cable 로 장비의 전원 공급과 Data 통신이 가능합니다.

PROFIBUS 의 Protocol 구조



PROFIBUS 네트워크는 ISO 7498 에 의한 OSI(Open System Interconnection)참조모델을 준용하여 PROFIBUS-FMS 의 경우 1 계층(물리영역), 2 계층(데이터링크), 7 계층(응용)으로 세 개의 계층을 사용합니다. 이에 반하여 모션게이트가 지원하는 PROFIBUS-DP 는 1 계층과 2 계층을 사용하여 매우 빠르고 효율적인 데이터 전송을 보장하고, Direct Data Link Mapper(DDLM)으로 사용자 인터페이스가 보다 쉽게 2 계층 영역에 접속 할 수 있도록 합니다.

모션게이트가 지원하는 PROFIBUS-DP 의 물리영역은 RS485 의 토폴로지를 사용합니다. 따라서 네트워크에 연결되는 모든 디바이스는 직선처럼 연결되고, 장치의 추가 또는 제거할 때 기존에 설치된 다른 디바이스에 아무런 영향을 주지 않습니다.

이러한 특징으로 대용량 데이터를 고속처리가 가능하고 현장 정보의 신속한 상위 연결이 가능하여 생산성이 향상되는 효과가 있습니다. 또한 RS485 의 토폴로지에 대한 특징으로 설치 및 유지보수가 간편합니다.

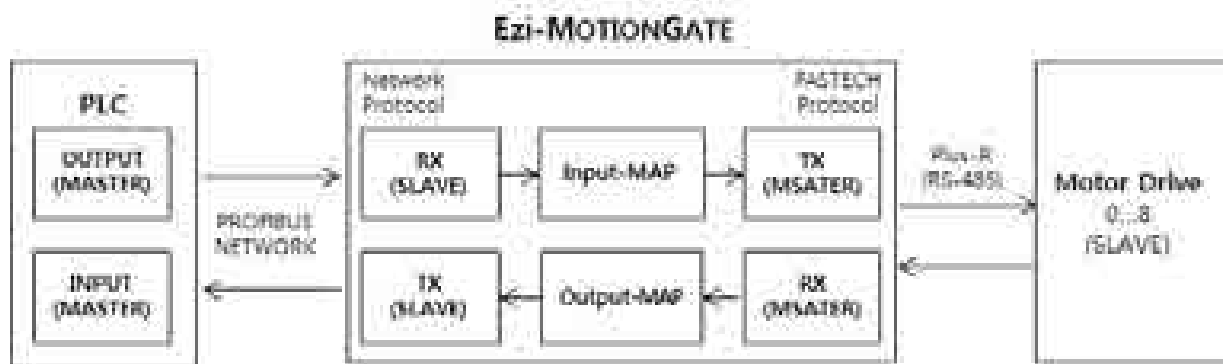
항목	사양	
스테이션 수	32 개/1 Segment (Repeater 의 미사용) 126 개(Repeater 의 사용)	
최대 I/O 수	6,000 포인트 (마스터에 따라서 다름)	
통신속도와 케이블 연장 거리	12Mbps : 100m 1.5Mbps : 200m 0.5Mbps : 400m	187.5Kbps : 900m 93.75Kbps : 1200m 19.2Kbps : 1200m 9.6Kbps : 1200m
통신 방식	Broad Casting Pooling	

4.1.2 모션게이트의 개요 및 네트워크 구성

PROFIBUS-DPV1 네트워크에서 사용하는 모션게이트는 최대 9 개(Axis 0~8)의 모터 드라이브를 지원합니다. (모션게이트 별 모터드라이브의 최대 연결 수량은 지원되는 네트워크에 따라 달라집니다. 네트워크 별 지원 가능한 수량 확인은 *2.2 네트워크 별 제품 특성표 에서 확인할 수 있습니다.)

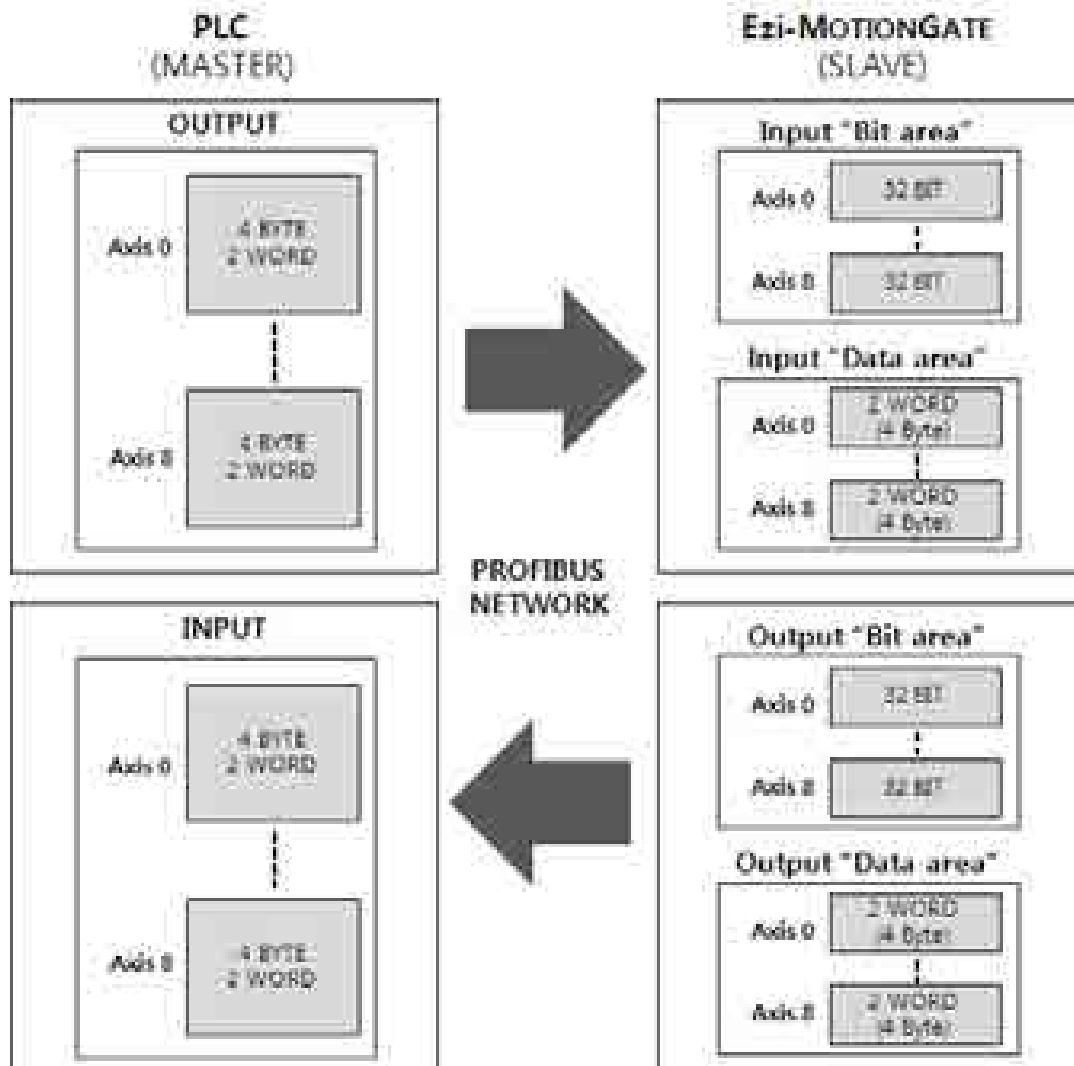
상위 제어기(PLC)는 모션게이트와 연결된 송수신 구간에 대한 메모리 주소를 액세스 가능한 마스터 시스템이 되어야 합니다. 그리고 모션게이트는 연결된 모터 드라이브에 대한 송수신 구간을 액세스가 가능한 마스터 시스템이 됩니다.

PROFIBUS-DPV1 네트워크로부터 수신된 Input-Map 데이터가 해당하는 축을 제어하고, 해당 축에서 응답한 정보는 Output-Map 으로 구성되어 PLC 로 상위 제어기에 대한 데이터를 송신합니다.



PROFIBUS-DPV1 네트워크에서의 모션게이트에 연결된 각 모터드라이브에 대한 명령은 IO-Map 으로 구성된 데이터를 사용하여 명령 및 정보 요청을 합니다. 여기서 IO-Map 의 어드레스 영역은 각 축에 대한 제어 명령, 응답 정보 확인을 위한 영역입니다. IO-Map 의 데이터 구성은 한 축에 대하여 32bit 로 구성된 비트영역(Bit area)와 2WORD(4Byte)로 구성된 데이터영역(Data area)으로 나뉩니다.

상위 제어기(PLC)는 IO-Map 의 데이터 어드레스 영역으로 제어와 응답 데이터를 확인 할 수 있습니다.

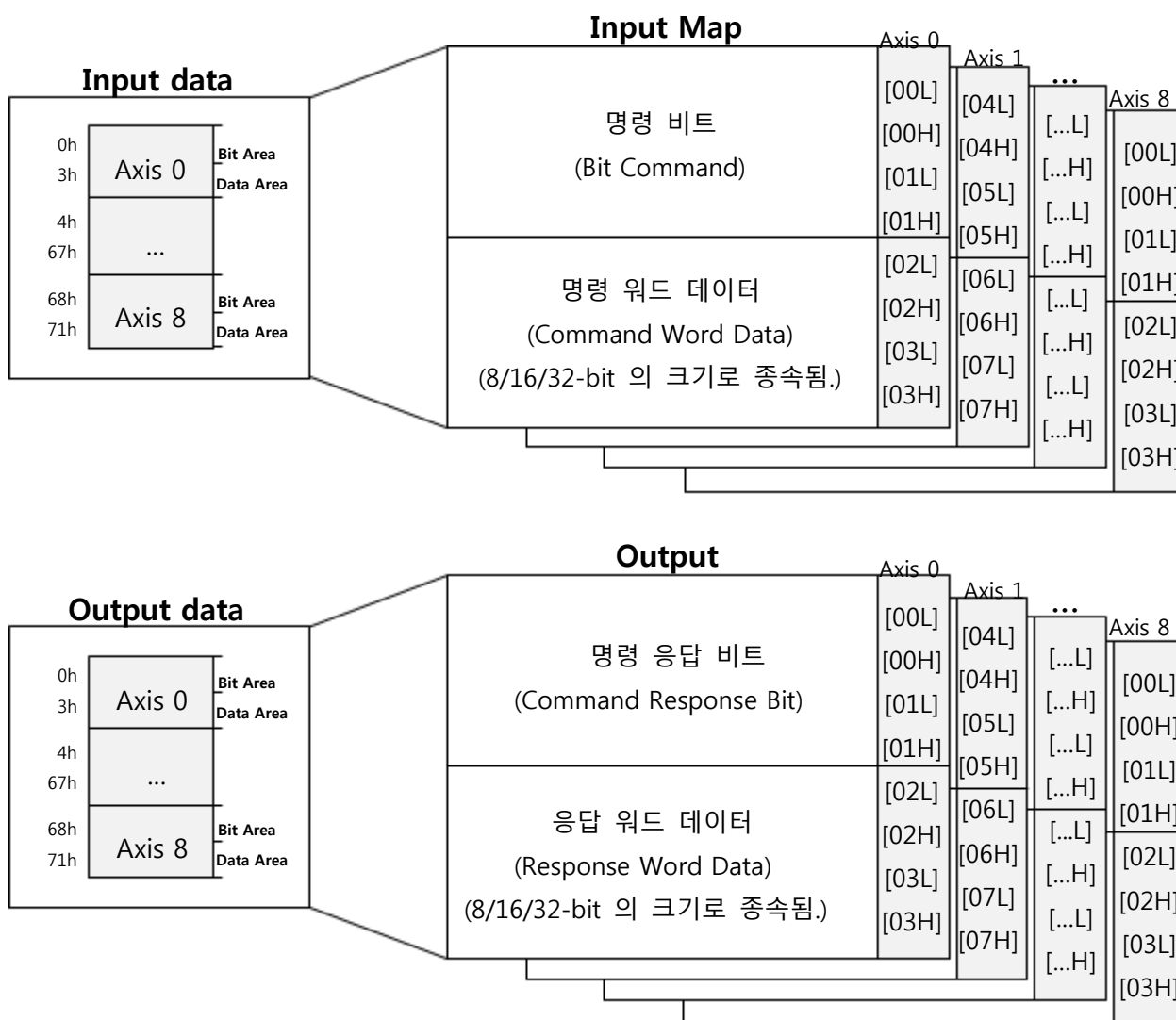


모션게이트의 IO-Map 데이터 어드레스 영역은 상위 제어기의 송수신 버퍼 영역 72byte 영역을 점유 합니다. (한 축에 대한 Input-Map 과 Output-Map 은 각각 8Byte[4WORD]를 점유합니다.)

IO-Map 은 비트영역[0-3]과 데이터영역[4-7]으로 나누어 집니다.

Input-Map 의 비트영역은 모터축의 비트명령을 위한 구간으로 사용되며, 데이터영역은 비트명령에 해당하는 데이터 정보를 입력하는 구간으로 사용됩니다. Output-Map 의 비트영역은 해당 축의 상태플래그 또는 제어 명령에 대한 응답 비트로 사용되고, 워드 데이터영역은 Input-Map 의 명령으로 요청한 데이터가 지정 되는 구간입니다.

IO-Map 의 구조는 맨 처음의 주소에서 4 번째의 주소까지의 바이트 영역은 0 번축에 대한 IO-Map 이고 뒤로 1 번축의 IO-Map 으로 이어져 8 번축의 송신구간으로 8 바이트 간격으로 이어집니다.



NOTE 1: 상위 제어기의 디바이스 메모리 1WORD 는 16Bit 데이터로 2byte 의 크기를 갖고 있습니다. 그러므로 디바이스 메모리의 주소 0h 의 00.0~00.7 영역은 IO-Map 의 0Byte 영역을 점유하고, 00.8~00.15 는 1Byte 영역을 점유합니다.

NOTE 2: 데이터 영역이 사용하는 디바이스 점유 메모리는 2WORD 입니다. 따라서 데이터영역의 시작 주소에 대한 DWORD 주소를 사용할 수 있습니다.

4.2 제어명령 비트영역(Input Map)과 상태정보 비트영역(Output Map)

Input-Map의 비트 구성

Input-Map 은 모터 드라이브의 제어를 지령 하는 영역입니다. 명령에 대한 비트의 조합으로 모터 드라이브의 모션 제어의 선택 및 응답 정보 형식 설정, 파라미터 또는 PT 정보 등의 값을 설정할 수 있습니다.

Input-Map의 비트 영역(상위 4 바이트 영역)

BYTE offset	BIT	비트 이름	동작 레벨	Description
0	0	CONNECT	상승 엣지	<p>이 비트의 설정으로 해당 축의 사용여부를 결정하며, '1'로 세트 하였을 때 해당 축과 통신연결을 시도하며, 해당 축과 통신이 필요하지 않을 경우 이 비트의 설정을 '0'으로 설정합니다. '0'으로 설정 되었을 때 모션게이트는 해당 축과의 통신처리는 제외되며, 어떠한 명령도 실행 하지 않습니다.</p> <p>여러 축에 대한 명령이벤트가 동시에 발생되었을 시 모터 드라이브의 번호가 낮은 순에서 높은 순으로 처리가 됩니다. 한 축에 대한 이벤트의 처리가 완료되었다면, 다음 ID의 축에 대한 프로세스 작업으로 변경 됩니다.</p> <p>해당된 축에서 명령 또는 이벤트가 없을 경우, MotionGate 는 해당 축의 상태정보와 응답요청 항목에 대한 데이터를 수신 합니다.</p> <ul style="list-style-type: none"> - 해당 축의 상태정보(flags FLAG-define) - 지령 위치 값 (signed long 32-bit) - 실제 위치 값 (signed long 32-bit) - 위치 오차 값 (signed long 32-bit) - 현재 운전 속도 값 (signed long 32-bit) - 현재 운전중인 PT 번호 <p>NOTE 1: 모션게이트는 모터드라이브와 수시로 Fas_GetAllStatus() 함수의 명령을 실행 합니다.</p> <p>NOTE 2: 모터제어의 지연시간은 동일한 지연시간을 갖는 모터일 경우 연결 축수에 대한 배가 됩니다.</p>

BYTE offset	BIT	비트 이름	동작 레벨	Description
	1	ENABLE / IGNORED	상승 엷지	SERVO Drive : 해당 축의 상태를 모션동작이 가능한 상태로 전환합니다. 0 : Servo OFF 1 : Servo ON STEP Drive : 이 비트의 명령은 무시 됩니다.
	2	nESTOP	하강 엷지	모션 또는 모든 명령 실행 정지. (비상 정지) * 0: E-Stop 명령 실행, 1: E-Stop 명령 대기
	3	ALARM_RESET / MOTOR_FREE	상승 / 하강	SERVO Drive : 발생된 알람을 해제 할 때 사용 (상승 엷지 동작) * STEP Drive 는 MOTOR_FREE 비트를 '1'로 유지할 경우 STEP 드라이브는 Motor Free 상태가 유지되고, '0'으로 전환되는 하강엷지에서 스텝모터 알람 리셋 명령이 실행 됩니다.
	4	CMD_START	상승 엷지	위치이동 또는 PT 운전, 원점이동의 명령을 실행 시 사용
	5	-	-	-
	6	-	-	-
	7	MOTION / SETTING	H/L	MotionGate 의 Map 의 구성을 모션 및 설정으로 선택하는 비트 0: 모션제어 모드 1: 설정 모드
1	0	CMD_CODE0	H/L	모션 제어 모드일 때 0000(0): 일반 모션(Jog, Step, 영점 이동) 0001(1): 상대치 이동[Incrometal Move], 절대치 이동[Absolute Move] 0100(4): PT 운전 (PT 운전, 싱글 PT 운전) 0111(7): 원점 이동 (Origin) 설정 모드 일 때 0000(0): 명령 없음 0101(5): 버전 정보 확인 1000(8): 파라미터 요청 1001(9): 파라미터 쓰기 1010(10): 위치정보 변경 1100(12): 알람내역 요청 1101(13): 알람내역 삭제 1110(14): 파라미터 저장
	1	CMD_CODE1		
	2	CMD_CODE2		
	3	CMD_CODE3		

BYTE offset	BIT	비트 이름	동작 레벨	Description
	4	RESPONSE_TYPE0	H/L	해당 축에 대한 RX 구간으로부터 수신을 원하는 응답데이터의 응답형식을 지정 0000(0): 응답데이터를 요청하지 않음. 0001(1): 지령 위치 0010(2): 실제 위치 0011(3): 위치 오차 0100(4): 현재 속도 0101(5): 운전중인 PT 번호 1000(8): 현재 발생된 알람 번호
	5	RESPONSE_TYPE1		
	6	RESPONSE_TYPE2		
	7	RESPONSE_TYPE3		* RESPONSE_TYPE 비트 영역은 설정 모드에서는 사용되지 않습니다.
2	0	CANCEL	상승 엷지	모션에 대한 일반 정지
	1	HOLD	상승 엷지	모션 중 일시 정지
	2	-	-	-
	3	GO_ZERO_POS	상승 엷지	해당 축의 드라이브에서 지정된 영점으로 이동 (위치 값: 0)
	4	-JOG_MOV	상승 엷지	역방향 조그(JOG)운전 데이터 영역의 입력 값: 속도 비율, 속도 값, Speed Step 번호.
	5	+JOG_MOV	상승 엷지	정 방향 조그(JOG)운전 데이터 영역의 입력 값: 속도 비율, 속도 값, Speed Step 번호.
	6	-STEP_MOV	상승 엷지	MotionGate 의 내부 파라미터 값인 위치 값과, 속도 값으로 가감이동 데이터 영역의 입력 값: 위치 값의 번호(0~3) *사용자에 의하여 재 정의 가능
	7	+STEP_MOV	상승 엷지	MotionGate 의 내부 파라미터 값인 위치 값과, 속도 값으로 증감이동 워드영역의 입력 값: 위치 값의 번호(0~3) *사용자에 의하여 재 정의 가능

BYTE offset	BIT	비트 이름	동작 레벨	Description
3	0	INC/ABS	H/L	제어 방법이 위치이동일 때(CMD_CODE:0001) 상대치 이동 또는 절대치 이동을 선택하는 비트 0: 상대치 이동 1: 절대치 이동
	1	-	-	-
	2	SPD_MODE	H/L	제어 방법이 일반 모션일 때(CMD_CODE: 0000) Jog 이동 시 사용 0: 입력된 비율 값 또는 Speed Step 번호로 Jog 운전 1: 입력된 속도 값으로 Jog 운전
	3	-	-	-
	4	SINGLE_PT	H/L	제어 방법이 PT 운전일 때(CMD_CODE:0100), 일반 PT 운전, 또는 싱글 PT 운전을 선택하는 비트. 0: 일반 PT 운전 1: 싱글 PT 운전
	5	-	-	-
	6	-	-	-
	7	-	-	-

NOTE: Input Map 은 모션게이트가 PLC 또는 Master 로 명령을 입력 받는 영역 입니다.

Output-Map의 비트 구성

Output-Map의 구간은 데이터 플래그와 비트명령에 대한 루프-백(Loop-Back: 되돌림)비트가 존재합니다. 루프-백 비트는 해당 비트의 명령 이벤트와 동일하게 반응하는 비트로 Input-Map의 비트 입력 여부를 확인할 수 있습니다. 상태 플래그는 해당 모터드라이브와 통신하여 수신된 데이터 정보를 기반으로 나타냅니다.

Output-Map의 비트 영역(상위 4 바이트 영역)

Byte offset	bit	비트 이름	동작 레벨	비트 유형	Description
0	0	CONNECTED	H	상태 비트	해당 축의 Plus-R과 연결되었을 경우 이 비트는 '1'로 세트
	1	ENABLED MOTOR_FREE_RESP (STEP)	H	상태 비트	해당축의 Servo ON 또는 스텝모터의 노말 상태일 때 '1'로 세트 *STEP Drive 일 때, Motor Free 명령에 대한 응답 비트가 됨.
	2	ESTOP_RESP	H	루프-백	Input-Map의 nESTOP 비트의 루프-백 비트로 비상정지 명령이 실행되면 '1'로 세트
	3	ALARM_ERROR	H	상태 비트	해당 축의 모터 드라이브에서 알람이 발생되었을 때 자동으로 '1'로 세트, 알람이 해제되면 '0'으로 클리어
	4	CMD_RESP	H	루프-백	Input-Map의 CMD_START 비트의 루프-백 비트
	5	OUT_RANGE	H	상태 비트	Input-Map의 데이터영역의 값이 해당 명령의 값의 범위에 맞지 않았을 때 '1'로 세트
	6	READY	H	상태 비트	현재 해당 축에 대한 명령이 실행 가능한 상태일 때 '1'로 세트 됩니다. 이 비트가 '0'인 상태에서 어느 명령도 동작하지 않습니다. NOTE 1: 세팅 모드에서 READY 비트가 '1'로 세트 되었을 때, 다른 축의 제어가 가능합니다.
	7	SET_MOV_RESP	H/L	루프-백	현재 Output-Map의 데이터가 세팅 모드일 때는 '1'로 세트 되고, 모션모드일 때 '0'으로 클리어

Byte offset	bit	비트 이름	동작 레벨	비트 유형	Description
	0	CMD_CODE_RESP0	H/L	루프-백	명령 받은 모션의 종류를 응답 0000(0): 일반 모션(Jog, Step, 영점 이동) 0001(1): 상대이동[Incrometal Move], 절대값이동[Absolute Move] 0100(2): PT 운전 (PT 운전, 싱글 PT 운전) 0111(3): 원점 이동 (Origin)
	1	CMD_CODE_RESP 1	H/L		
	2	CMD_CODE_RESP 2	H/L		
	3	CMD_CODE_RESP 3	H/L		
	4	RESPONSE_TYPE_RESP0	H/L	루프-백	Word 영역에 지정되는 되는 응답데이터의 정보를 응답 0000(0): 응답데이터를 요청하지 않음. 0001(1): 명령위치 0010(2): 실제 위치 0011(3): 위치 오차 0100(4): 현재 속도 0101(5): 운전중인 PT 번호 1000(8): 현재 발생된 알람 번호
	5	RESPONSE_TYPE_RESP1	H/L		
	6	RESPONSE_TYPE_RESP2	H/L		
	7	RESPONSE_TYPE_RESP3	H/L		
2	0	MOTIONNING	H/L	상태 비트	해당 축의 모션상태일 때 '1'로 세트
	1	HOLD_RESP	H/L	상태 비트	운전 중 HOLD 비트의 명령으로 일시 정지된 상태일 때 '1'로 세트
	2	-	-	-	-
	3	GO_ORIGIN_RESP	H	상태 비트	해당 축의 Plus-R 의 파라미터로 원점복귀를 실행 중 일 때, '1'로 세트
	4	-	-	-	-
	5	JOG_RESP	H	상태 비트	해당 축이 Jog 운전 중일 때
	6	-	-	-	-
	7	STEP_RESP	H	상태 비트	해당 축이 Step 운전 중일 때

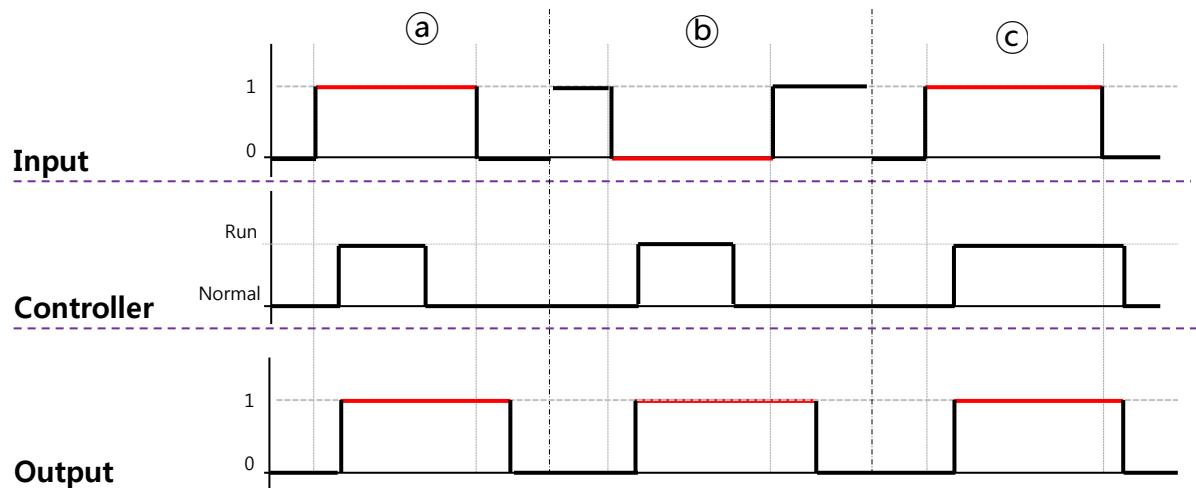
Byte offset	bit	비트 이름	동작 레벨	비트 유형	Description
	0	PT_RUNNING	L/H	상태 비트	해당 축이 위치이동 중일 때
	1	MOV DIR	L/H	상태 비트	모터의 회전방향을 표시 0 : CW(+) 1 : CCW(-) * 이 비트를 확인 시 FLAG_IN_MOTION 비트가 '1'로 세트 되었을 때 갱신된 값으로 확인해야 합니다. 논리연산 (FLAG_IN_MOTION & FLAG_nDIR)
	2	INP	L/H	상태 비트	모터가 In position 이 완료되었을 때 '1'로 세트 됩니다. * 모터가 STEP 일 경우 이 비트는 동작되지 않습니다.
	3	ORIGIN_SENSOR	H	상태 비트	원점센서가 ON 이 된 경우 '1'로 세트 됩니다.
	4	SW_LIMIT_N	H	상태 비트	'-' 방향 프로그램 리미트를 초과한 경우 '1'로 세트 됩니다.
	5	SW_LIMIT_P	H	상태 비트	'+' 방향 프로그램 리미트를 초과한 경우 '1'로 세트 됩니다.
	6	HW_LIMIT_N	H	상태 비트	'-' 방향 리미트 센서가 ON 이 된 경우 '1'로 세트 됩니다.
	7	HW_LIMIT_P	H	상태 비트	'+' 방향 리미트 센서가 ON 이 된 경우 '1'로 세트 됩니다.

NOTE: Output Map 은 모션게이트가 PLC 또는 Master 로 상태정보를 출력하는 영역 입니다.

4.3 IO Map 의 동작 순서 및 동작 조건

4.3.1 IO-Map 의 비트명령 방법

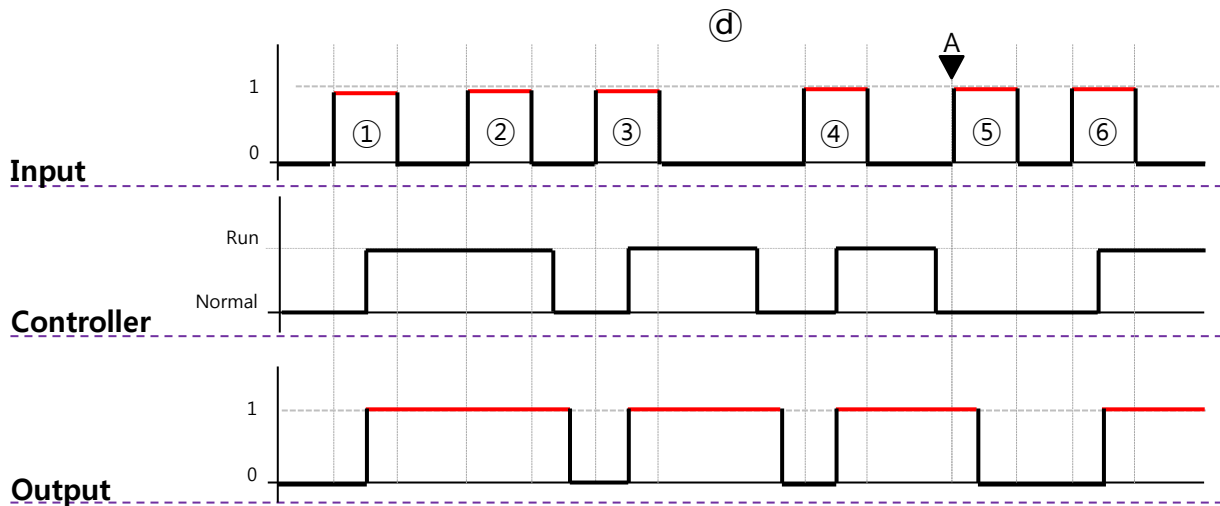
비트 명령은 상승엣지와 하강엣지 명령으로 구분됩니다.



Input 의 상승 엣지의 명령 시작은 ㉠구간과 같이 '0' 상태에서 '1'로 변경되는 시점입니다. 이 명령을 받은 모션게이트는 해당 축으로 명령을 하달하고, 그 명령이 실행될 때 Output 으로 명령에 대하여 응답합니다.

하강 엣지의 명령 시작은 ㉢의 구간은 와 같이 Input 명령이 '1' 상태에서 '0'으로 변경되는 시점입니다. 이 이벤트로 모션게이트는 해당 축으로 명령을 하달하고, 그 명령이 실행 될 때 Output 으로 명령에 대하여 응답합니다.

㉢구간과 같은 비트 명령은 Input 의 상승엣지 명령으로 모션게이트가 해당 축의 동작명령을 하달하여, 하강 엣지의 명령이 있을 때까지 지속적인 명령이 유지 되는 명령 입니다. 이 명령의 순서는 Input 의 상승 엣지로 해당축의 동작이 되면, 동작에 대하여 Output 으로 응답 합니다. 그리고 Input 의 하강 엣지 명령으로 해당축의 동작이 정지 되면, Output 으로 동작의 정지 대하여 응답됩니다.



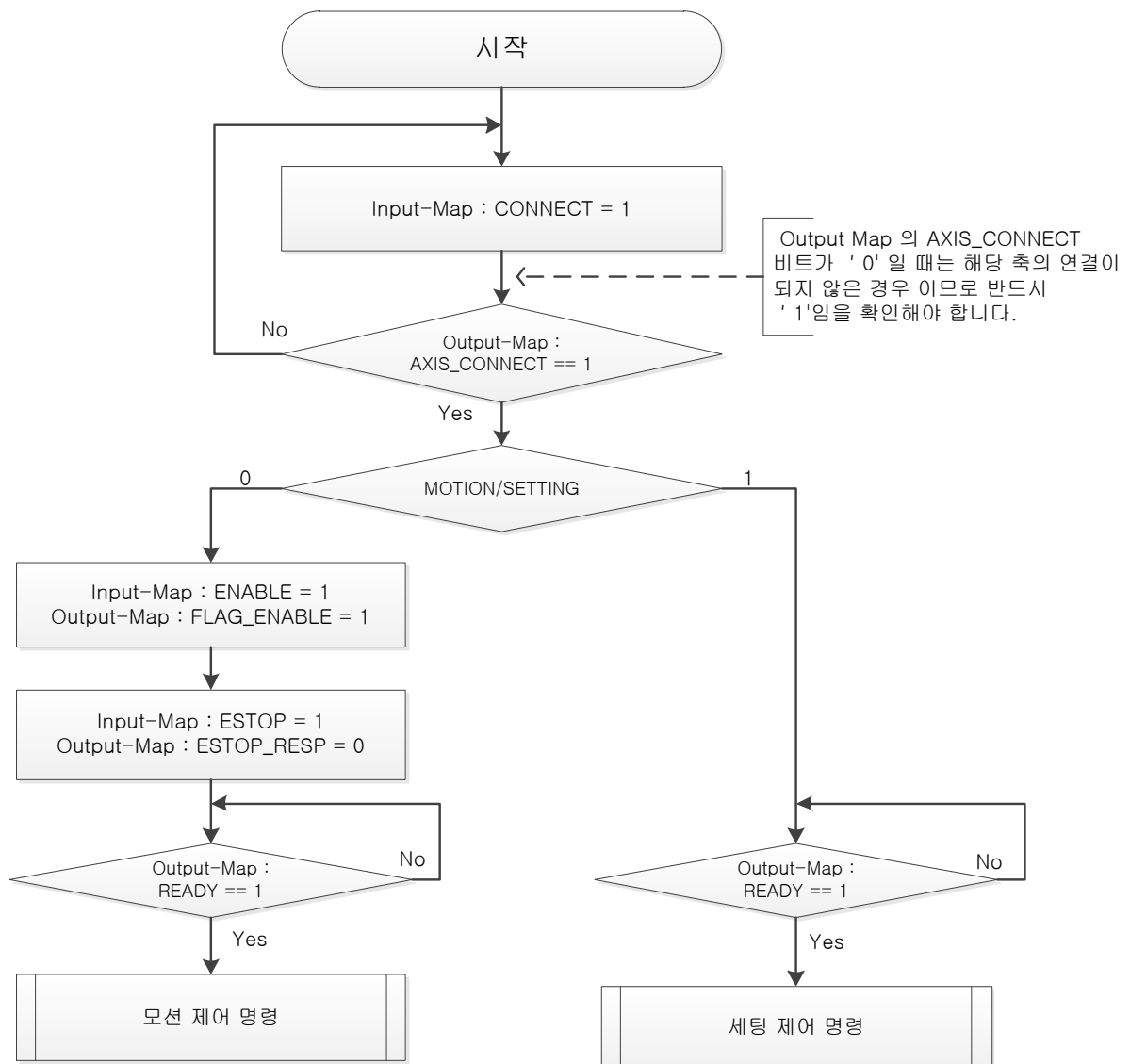
㉔구간은 Input 명령이 연속적으로 동작 했을 경우 입니다. 이러한 경우에는 ①의 명령으로 ㉔구간과 동일하게 명령이 시작됩니다. 이때 모션게이트가 동작 중일 때 입력된 명령 ② 으로 동작 되지 않습니다. 그리고 ①의 명령으로 동작한 모션게이트의 동작이 완료 된 후 입력된 ③의 명령으로는 동작이 실행됩니다.

④의 명령으로 실행된 동작이 완료되고 Output 의 응답하기 전의 시점인 A 에서 ⑤의 명령이 입력되면, 이 입력은 무시됩니다. 그러나 Output 이 응답된 후에 입력된 ⑥의 명령으로 동작합니다. 즉, Input 명령으로 모션게이트의 동작이 실행되고, Output 에 동작이 완료된 상태의 응답이 있을 때의 Input 명령이 유효 합니다.

4.3.2 IO-Map의 제어 명령 준비 순서

모션게이트는 명령을 실행할 때 아래의 순서의 과정이 필요 합니다.

순서도 1. 모션 및 설정 제어 명령의 활성화 조건



※ 모션게이트의 명령

- ① Input-Map 의 CONNECT 비트를 '1'로 세트 하여 명령을 실행 <참고: [IO-Map 사용 설명서 *2.2.1](#)>
 - CONNECT 비트는 해당축의 사용을 선택하는 비트 이므로 반드시 '1'로 세트
 - Output-Map 의 AXIS_CONNECT 비트의 응답 상태가 '1'임을 확인
- ② Input-Map 의 MOTION/SETTING 비트를 선택 <참고 : [IO-Map 사용 설명서 *2.1](#)>
 - 모션 제어는 '0', 세팅 제어는 '1'로 선택 합니다.
- ③ 모션 제어는 Input-Map 의 ENABLE 비트와 ESTOP 비트를 '1'로 세트 <참고: [IO-Map 사용 설명서 *2.2.1](#)>
 - Output-Map 의 응답비트 FLAG_ENABLE 비트 '1'임을 확인
 - ESTOP_RESP 비트가 '0' 을 확인
- ④ 명령을 실행 하고자 할 때 Output-Map 의 READY 비트의 상태를 확인 <참고: [IO-Map 사용 설명서 * 2.7](#)>
 - 다른 명령이 실행 중 일 때 READY 비트는 '0' 상태로 유지 됨.
 - 모션 명령이 없을 때 READY 비트는 '1' 상태로 유지 됨.
 - 세팅 명령을 할 때, 해당 명령이 완료 될 때까지 '0' 상태로 유지 됨.
- ⑤ 드라이브의 모션 제어는 모션 명령에 대한 IO-Map 의 비트 조합으로 수행합니다.
 <참고: [IO-Map 사용 설명서 *2.1.1, *2.1.2](#)>
 - Input-Map 의 MOTION/SETTING 비트를 '0'으로 설정
 - 모션 제어의 명령 실행은 "CONNECT= 1, ENABLE=1, nESTOP=1" 을 반드시 설정 해야 함
- ⑥ 드라이브 및 모션게이트의 설정 값에 대한 확인 및 수정은 세팅 제어에 대한 IO-Map 의 비트 조합으로 수행 됨 <참고: [IO-Map 사용 설명서 *6. *7. *8. *9](#)>
 - Input-Map 의 MOTION/SETTING 비트를 '1'로 설정
 - 세팅 제어의 명령 실행은 "CONNECT= 1" 을 반드시 설정 해야 함

5. IO-Map 의 사용 예제

5.1. MELSOFT GX- developer 프로젝트 설정

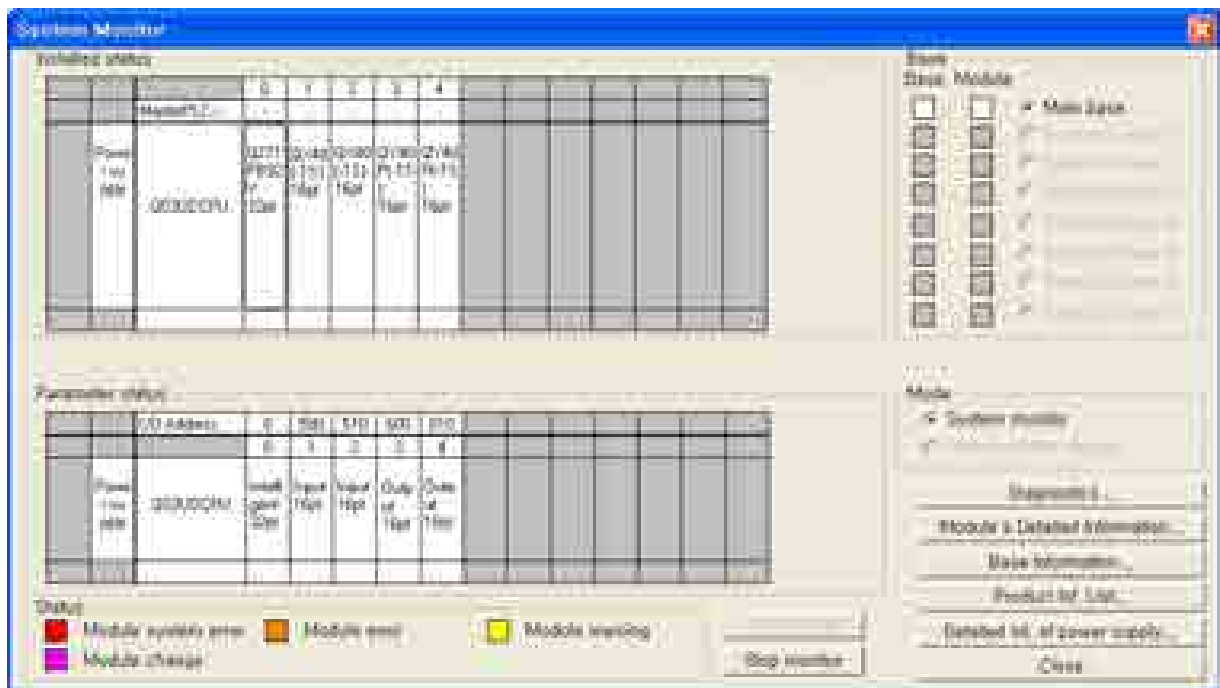
- 네트워크 파라미터의 설정시의 주요 항목

- FDL address : Fieldbus Data Link Address 로 네트워크의 주소를 말합니다.
(Node No. 또는 Station No.)
- Starting I/O Number : PLC 에 연결된 PROFIBUS Master 모듈에 할당된 I/O 어드레스 입니다.
- Block Transfer : PROFIBUS Master 모듈의 데이터 교환 구간 입니다.
 - Input : 모션게이트의 Output Map 의 데이터 영역으로 사용됩니다.
 - Output : 모션게이트의 Input Map 의 데이터 영역으로 사용됩니다.

※ 본 매뉴얼에서의 예제는 다음과 같은 PLC 기반으로 제작 되었습니다.

- PLC 시리즈 : MELSEC-Q
- PLC 의 유형 : Q03UD
- PROFIBUS-DP 모듈 : QJ71PB92V
- MELSOFT GX Configurator-DP
- MELSOFT 시리즈 GX developer 8

예제에서 사용된 PLC 시스템

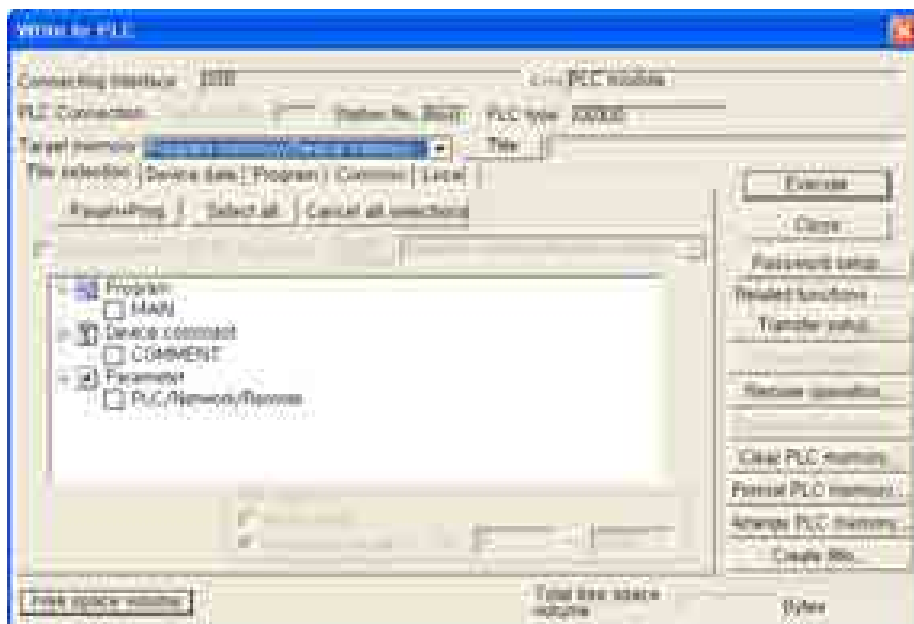


■ PLC 파라미터 설정

- PLC 프로그램 GX developer 의 프로젝트에서 PLC 파라미터 설정을 그림 5.1 과 같이 I/O 할당란을 설정 합니다.



■ PLC 파라미터 다운로드

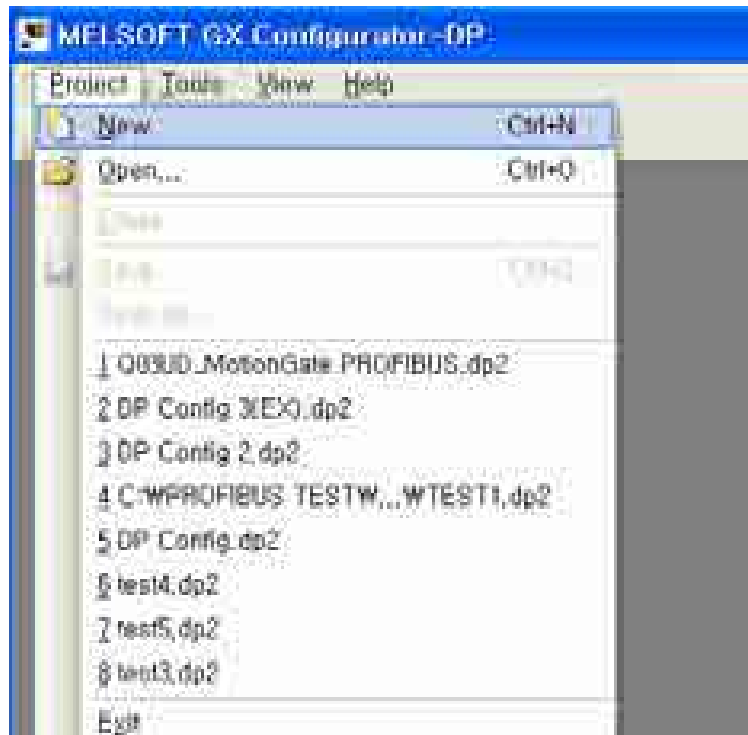


주의

PLC 의 최초 사용시 파라미터 정보를 저장하여 PLC 가 정상적인 "RUN" 상태가 되도록 준비 합니다.

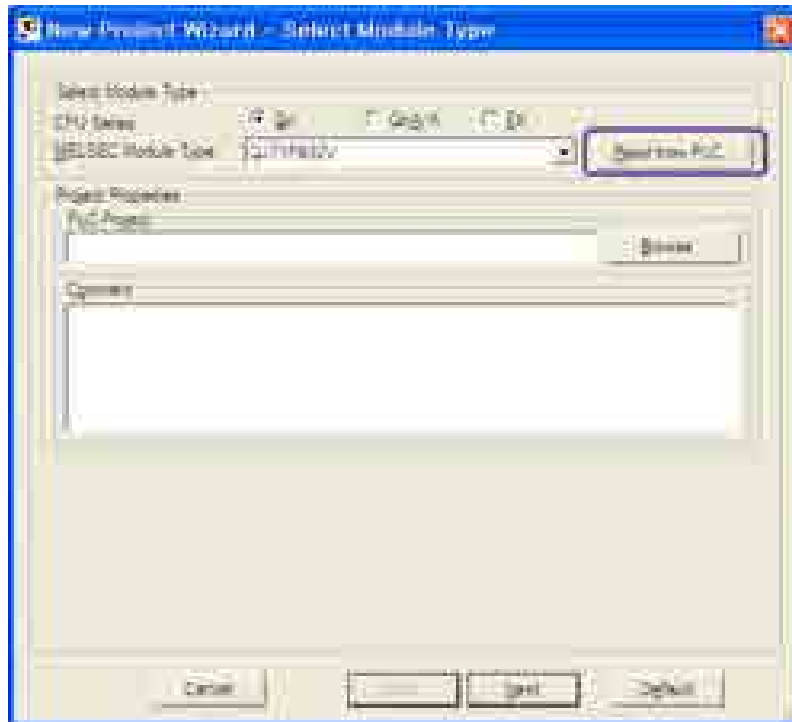
■ PROFIBUS Master 설정

- PLC 프로그램 GX Configurator-DP 로 PROFIBUS Master 모듈을 설정 합니다
- 메인메뉴의 [Project]->[New] 를 선택하여 신규 프로젝트를 제작합니다.

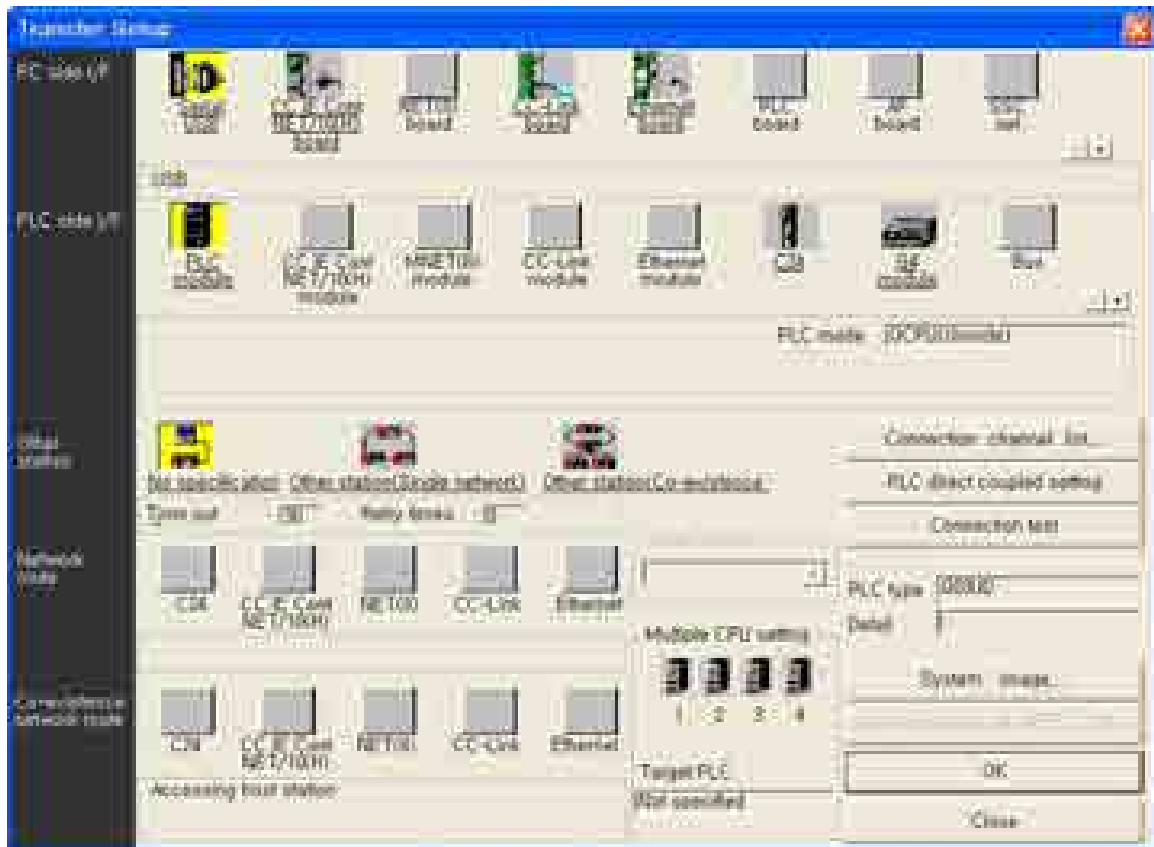


◆ 모듈선택

- Read from PLC 를 클릭하여 사용하고자하는 PLC 를 선택합니다.

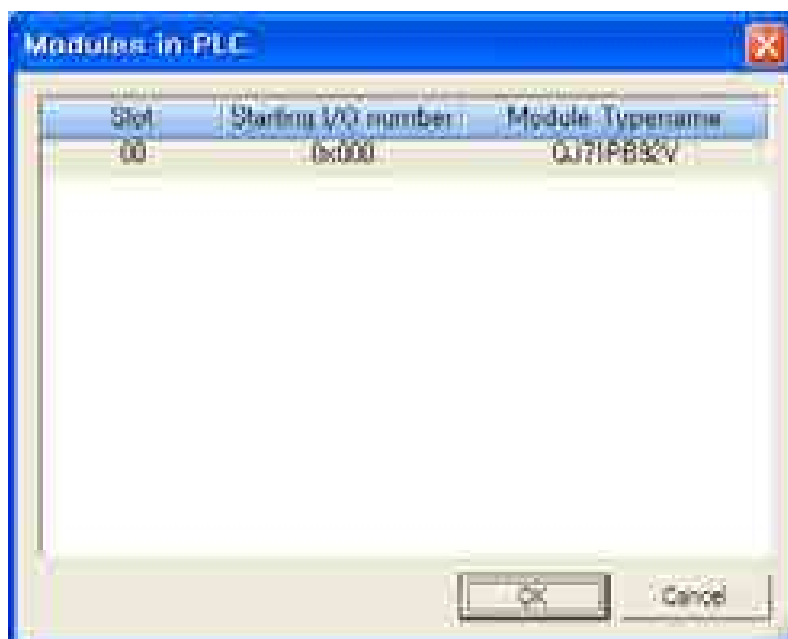


- PLC 의 연결 상태를 확인합니다.



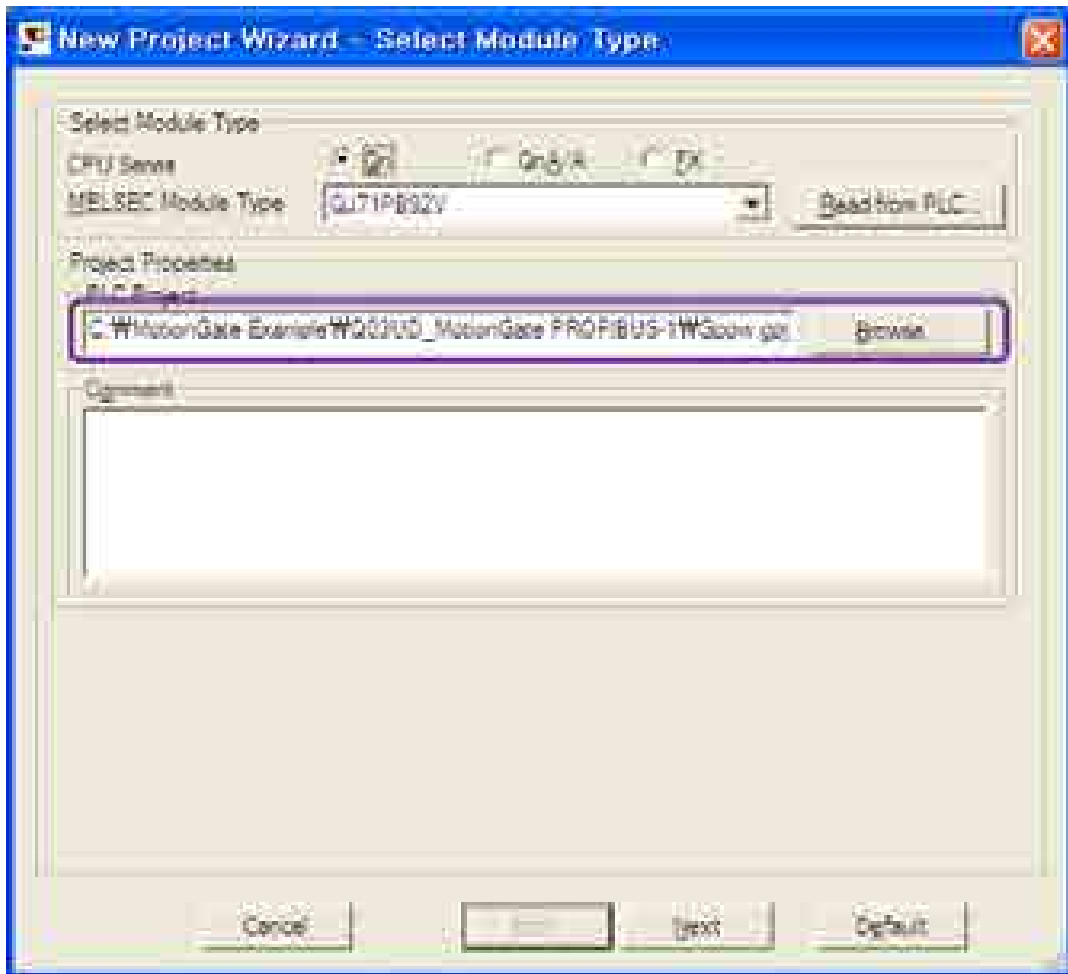
- GX Configurator-EP 프로그램은 자동으로 연결된 PLC 의 파라미터 정보를 읽은 후 PROFIBUS Master 모듈의 정보를 보여줍니다.

- 사용할 모듈을 선택후 OK 버튼을 클릭합니다.



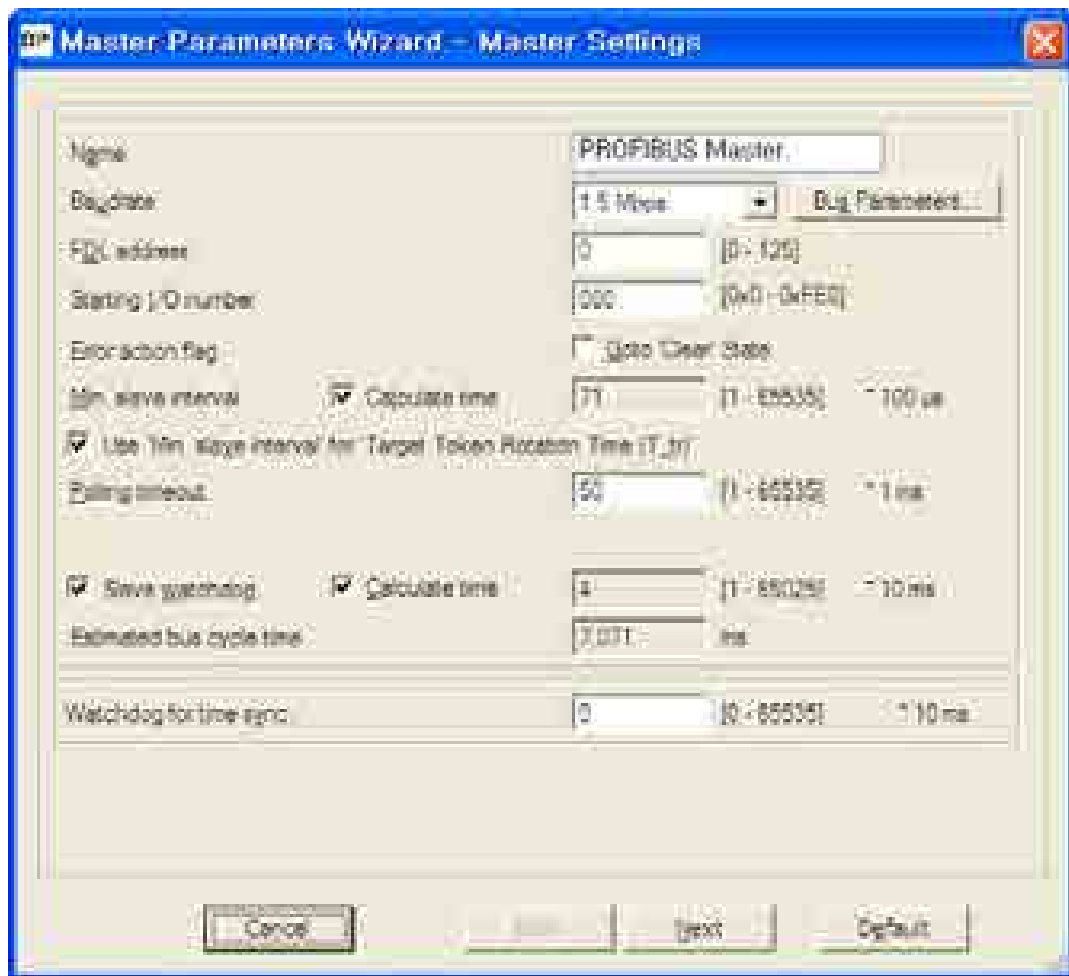
◆ PLC 프로젝트 선택

- 파라미터 설정 값을 추가할 GX developer 프로젝트 파일을 선택합니다.



NOTE: 본 예제는 PLC 파라미터의 설정시 제작한 GX-Developer 프로젝트 파일을 선택하였습니다.

◆ PROFIBUS Master 모듈의 설정



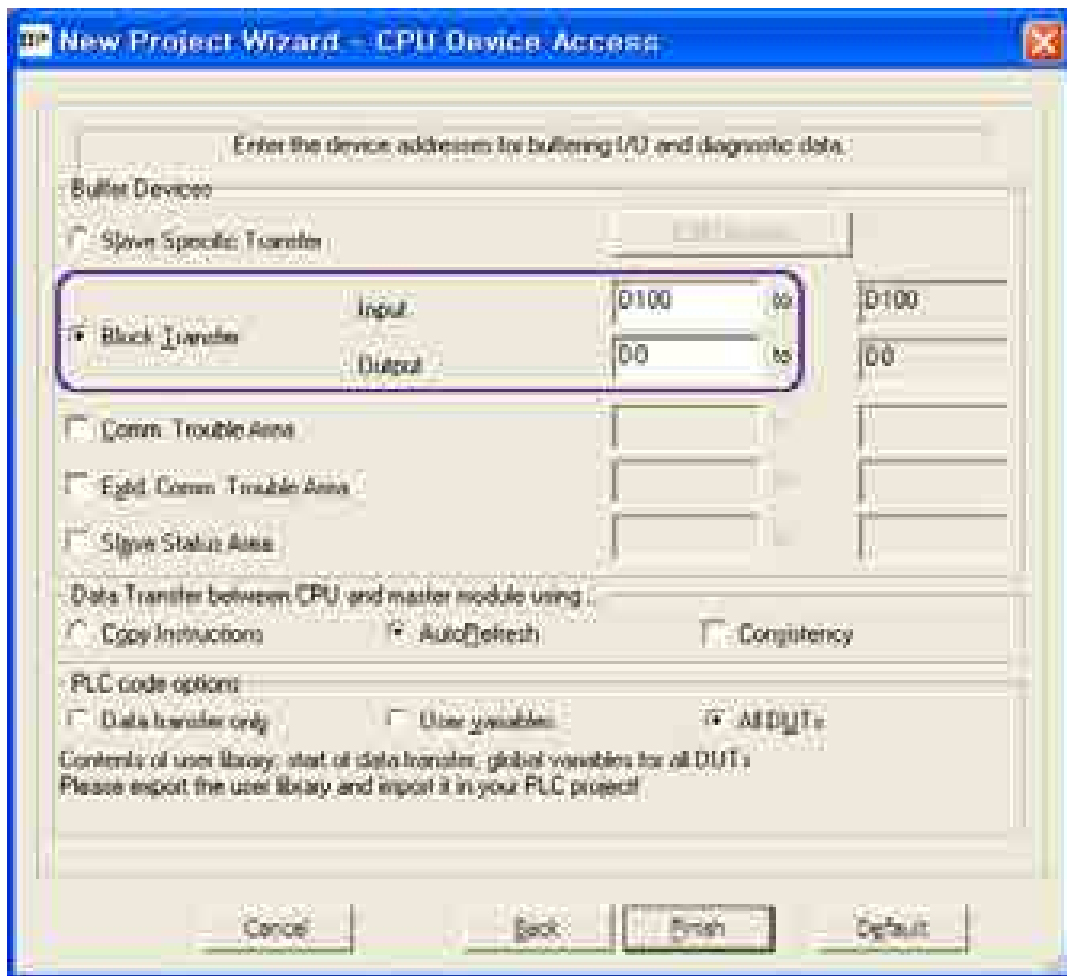
- Name : 모듈의 이름을 설정합니다
- Baudrate : PROFIBUS 네트워크의 보오레이트 설정치 입니다. 모션게이트는 Auto Baudrate 기능이 있어, 여기에 설정된 값으로 동작 됩니다. 만약 변동사항이 있다면, 모션게이트를 재부팅 하십시오.
- Starting I/O Number : PLC 파라미터에 설정된 시작 IO 어드레스를 입력합니다. 만약 앞의 과정에서 PLC의 파라미터를 성공적으로 읽어 왔을 경우, 자동으로 입력됩니다.

⚠ 주의

본 매뉴얼에서 설명하는 이외의 설정 사항은 사용자의 선택 범위이므로, 시스템의 환경에 따라 변동될 수 있습니다.

◆ CPU 데이터 설정

- PROFIBUS 네트워크의 데이터가 교환되는 영역을 설정합니다.



- Block Transfer 의 Input 으로 지정된 『D100』은 Output Map 의 데이터 영역의 시작 위치입니다.
- Block Transfer 의 Output 으로 지정된 『D0』은 Output Map 의 데이터 영역의 시작 위치입니다.

NOTE: 비활성화 된 영역의 주소는 PROFIBUS 네트워크에 사용되는 영역을 나타냅니다. 따라서 슬레이브 장비가 사용되는 I/O 메모리의 할당량에 따라서 범위가 바뀌게 됩니다.

- GX Configurator 프로젝트에 MOTIONGATE PROFIBUS-DPV1 의 정보 추가
 - 마스터 설정이 완료된 상태를 확인 합니다.
 - Add GSD File 를 클릭하여 모션게이트의 GSD 파일을 추가 합니다.



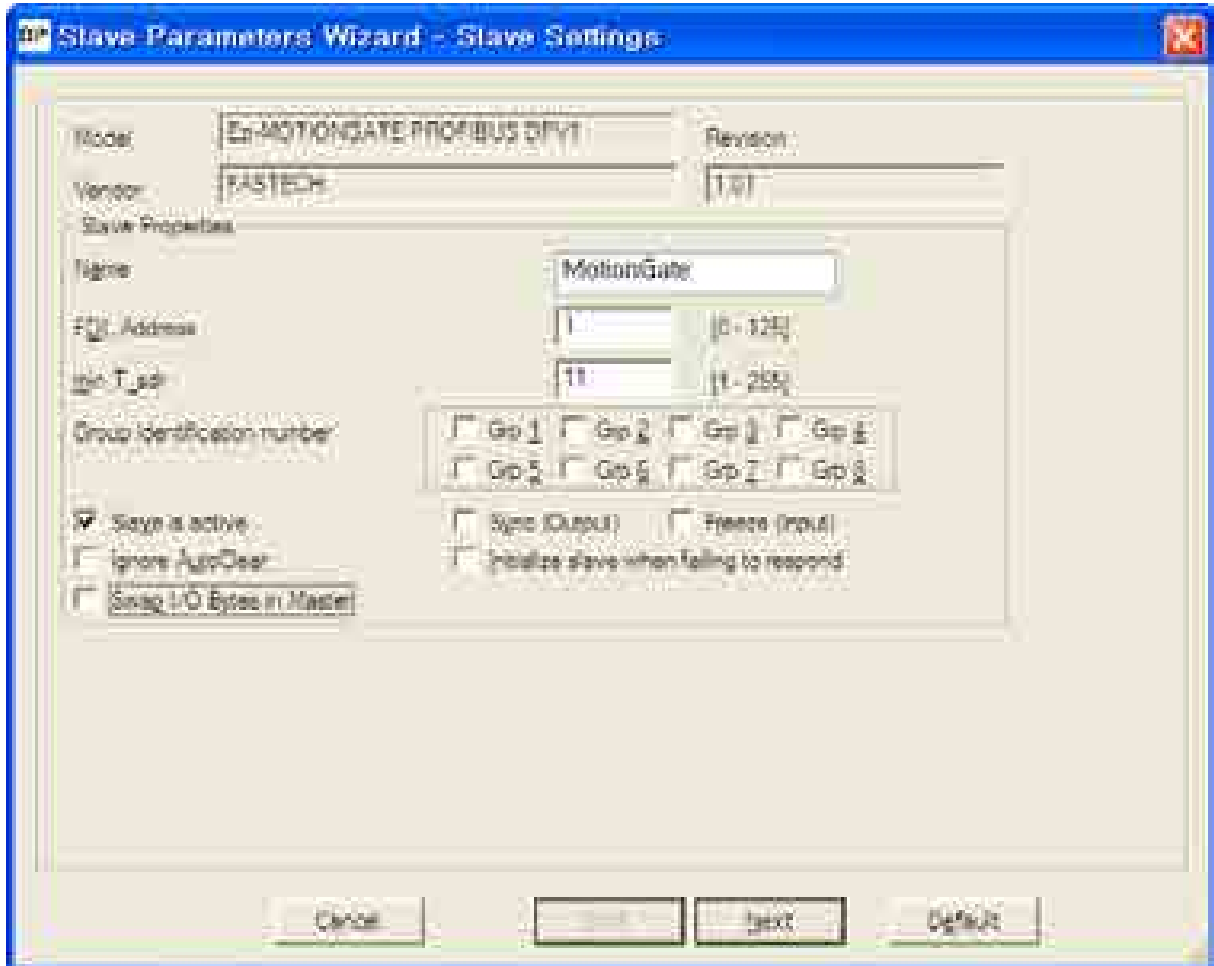
- GSD 데이터베이스에 모션게이트의 정보가 추가 되었음을 확인 합니다.



NOTE: 모션게이트의 GSD 파일은 동봉된 CD 또는 파스텍 홈페이지(www.fastech.co.kr)의 [파스텍 자료실]-[소프트웨어 프로그램] 에서 받으실 수 있습니다.

◆ 슬레이브 파라미터 설정

- 모션게이트의 GSD 데이터 베이스를 더블 클릭합니다.
- 슬레이브 디바이스의 정보를 확인 합니다.



- Name : 슬레이브 장비의 이름을 설정 합니다.
- FDL Address : PROFIBUS 의 노드 번호를 선택하는 영역입니다. 0~99 의 영역을 값을 입력 하십시오.
- Slave is active 를 체크 하여 슬레이브 장비인 모션게이트의 사용 활성화 합니다.

⚠주의

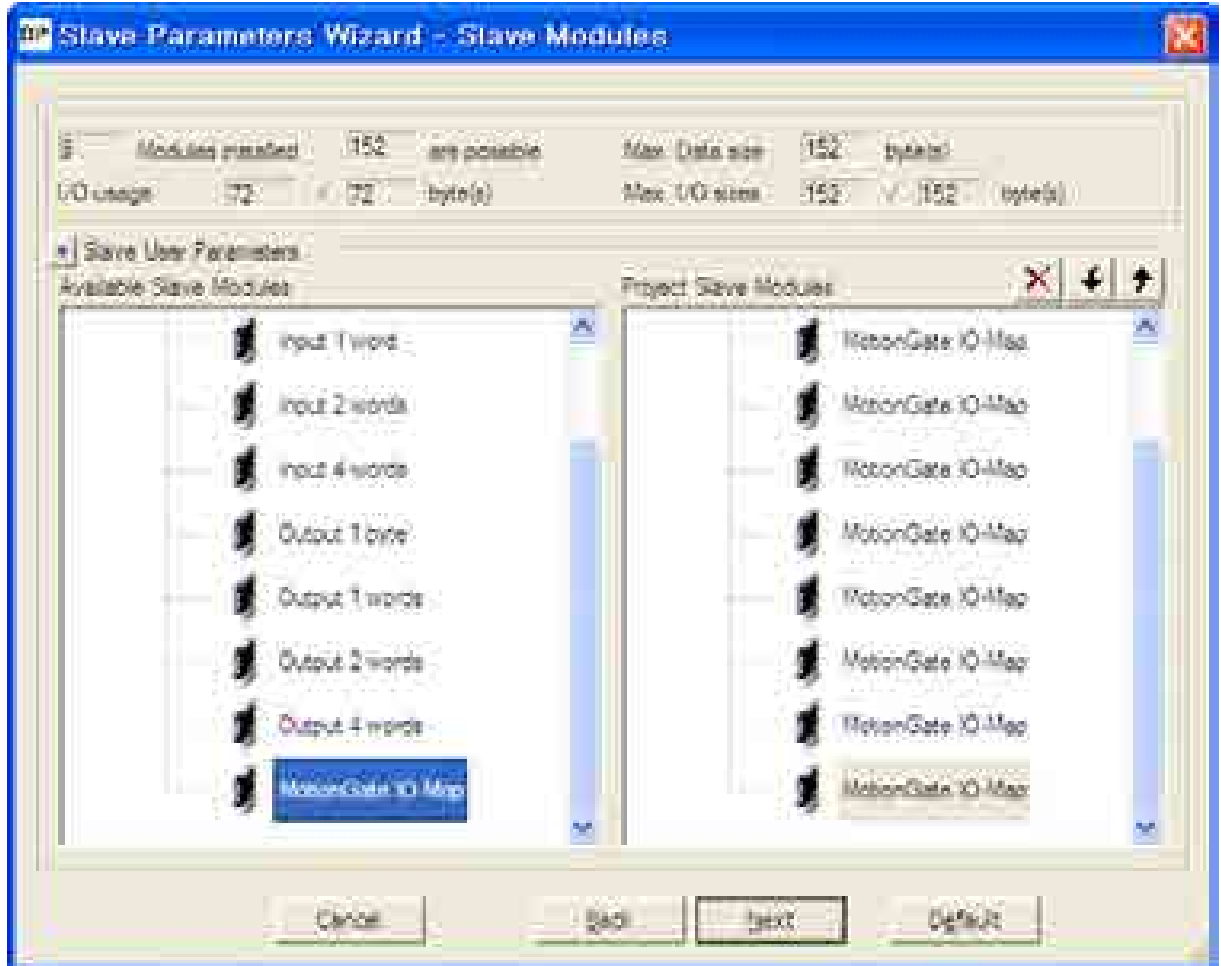
모션게이트의 슬레이브 설정 시 『Swap I/O Byte in Master』의 체크를 제거 하십시오.

⚠주의

본 매뉴얼에서 설명하는 이외의 설정 사항은 사용자의 선택 범위이므로, 시스템의 환경에 따라 변동될 수 있습니다.

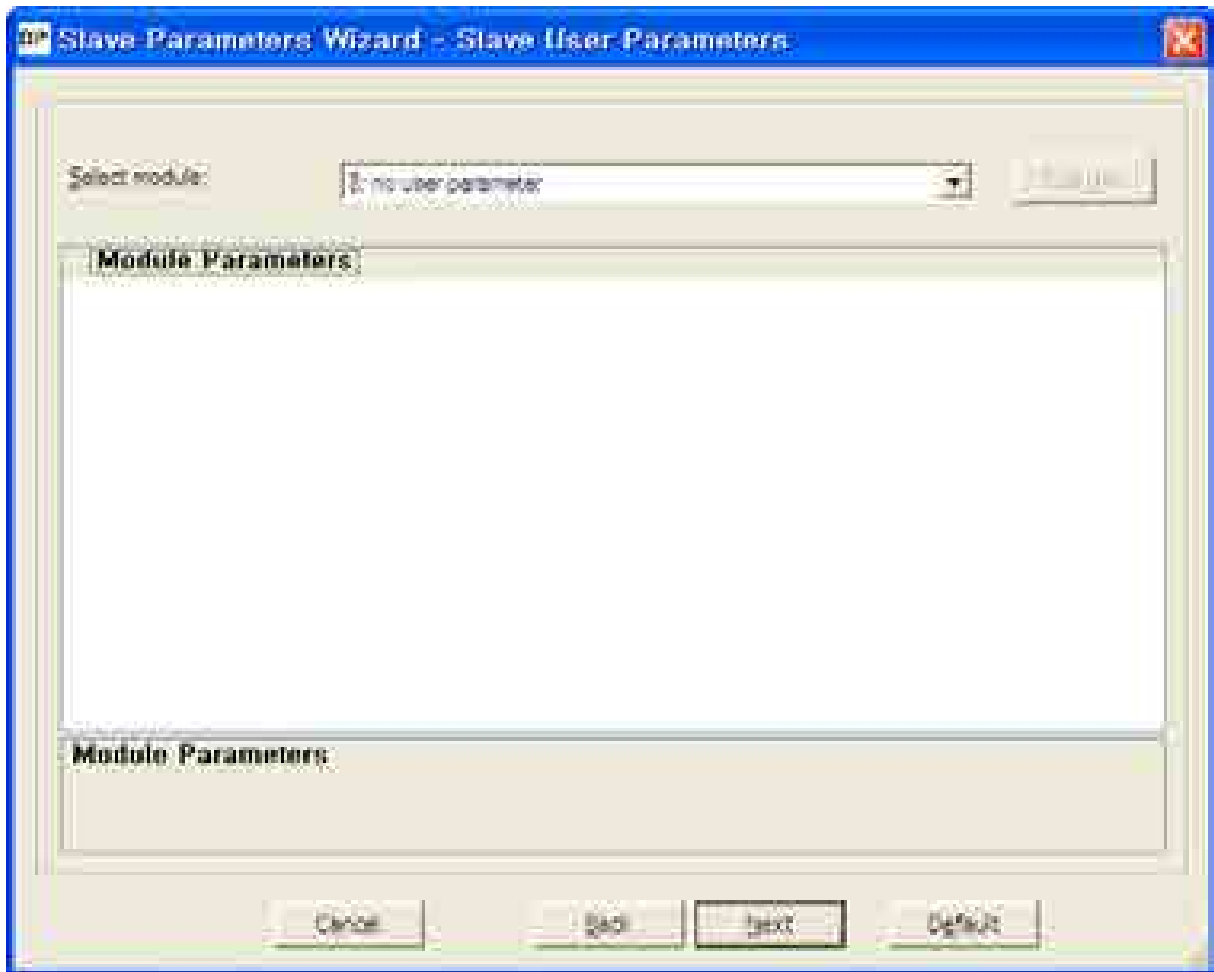
◆ 슬레이브 장비의 사용 메모리 설정

- 모션게이트가 사용하는 메모리의 공간을 할당하는 영역으로, Input 72byte, Output 72byte 의 공간이 필요합니다.
- Available Slave Modules 로 모션게이트에 연결되는 모터드라이브의 수(n) x 8byte 로 Input 와 Output 의 크기를 선정 합니다.



◆ 모듈 파라미터 설정

- 모션게이트는 PROFIBUS 슬레이브 파라미터가 없으므로, Next 버튼을 클릭하십시오.



- ◆ GX Configurator 프로젝트의 PROFIBUS I/O-Mapping 정보
- GX Configurator 는 프로젝트의 네트워크 설정 정보확인이 가능합니다.

Project: Q03UD_MotionGate PROFIBUS.dp2

FDL Addr.	Name	Model	Modules		
			Slot	Model	Global Var.
1	MotionGate	E2-MOTIONGATE PROFIBUS DPV1	0	MotionGate IO-Map	vHA60SLV1MOD0
			1	MotionGate IO-Map	vHA60SLV1MOD1
			2	MotionGate IO-Map	vHA60SLV1MOD2
			3	MotionGate IO-Map	vHA60SLV1MOD3
			4	MotionGate IO-Map	vHA60SLV1MOD4
			5	MotionGate IO-Map	vHA60SLV1MOD5
			6	MotionGate IO-Map	vHA60SLV1MOD6
			7	MotionGate IO-Map	vHA60SLV1MOD7
			8	MotionGate IO-Map	vHA60SLV1MOD8

Global Variables

MotionGate.Module Slot 0 : vHA60SLV1MOD0

Element Identifier	Element Type	Class	User MIT-Address	Global Var. Identifier	Buffer MIT-Address
input	ARRAY (0..3) OF WORD	Input	—	—	0100
output	ARRAY (0..3) OF WORD	Output	—	—	06

MotionGate.Module Slot 1 : vHA60SLV1MOD1

Element Identifier	Element Type	Class	User MIT-Address	Global Var. Identifier	Buffer MIT-Address
input	ARRAY (0..3) OF WORD	Input	—	—	0104
output	ARRAY (0..3) OF WORD	Output	—	—	04

●
●
●

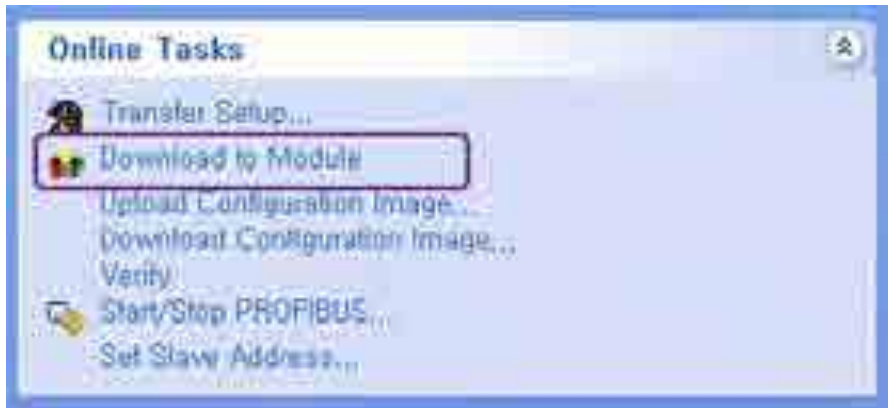
MotionGate.Module Slot 8 : vHA60SLV1MOD8

Element Identifier	Element Type	Class	User MIT-Address	Global Var. Identifier	Buffer MIT-Address
input	ARRAY (0..3) OF WORD	Input	—	—	0132
output	ARRAY (0..3) OF WORD	Output	—	—	002

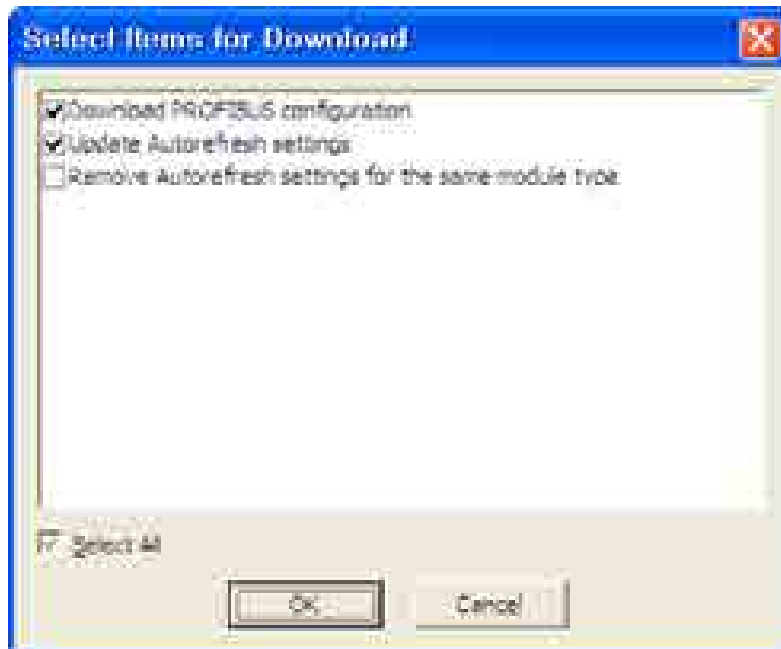
* Buffer MIT-Address 는 각각의 모터드라이브의 IO-Map 의 시작 어드레스가 됩니다.

■ GX Configurator 프로젝트 다운로드

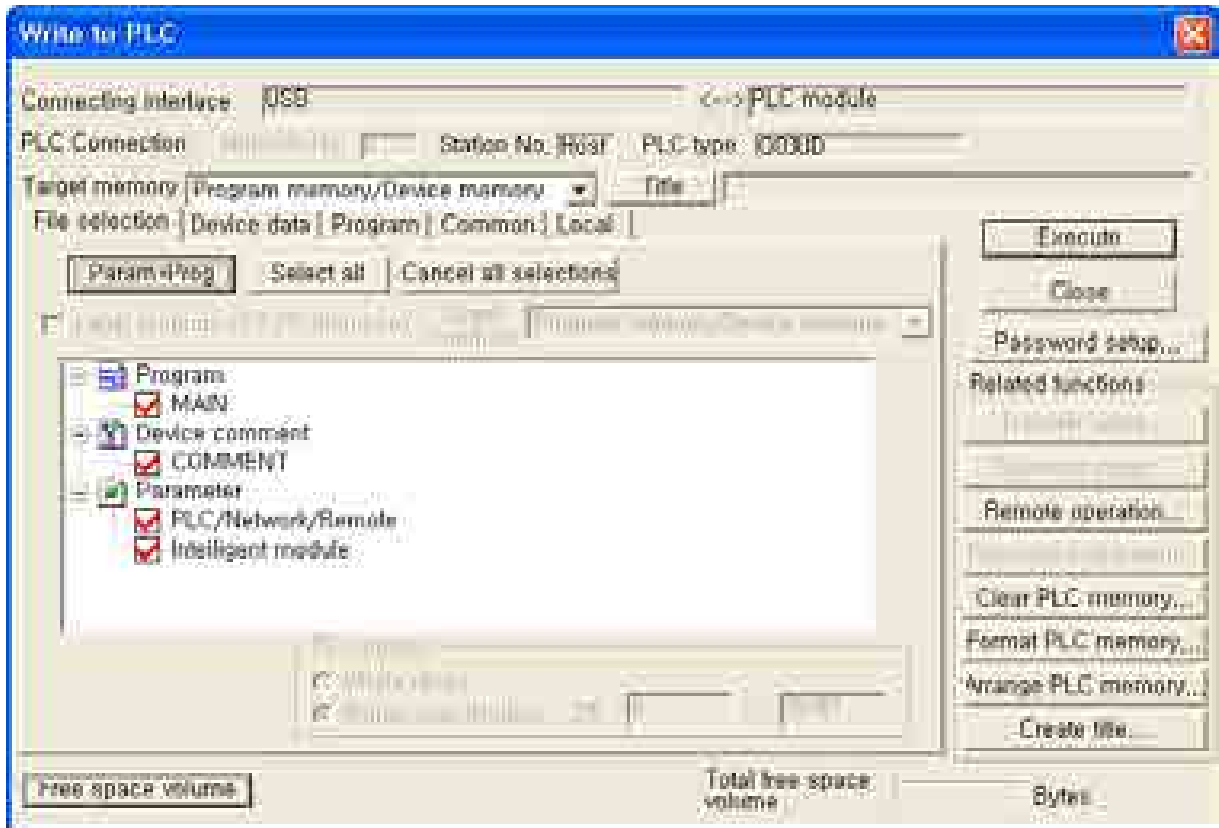
- a. GX Configurator 프로젝트의 마스터 설정 및 슬레이브 설정을 완료합니다.
- b. Download to module 을 클릭하여 다운로드를 실행합니다.



- c. 다운로드할 정보를 선택후 OK 버튼을 클릭합니다..



- d. 다운로드가 완료되면, GX-Developer 에서의 프로그램 다운로드시 인텔리전트 기능 모듈란이 추가 되었음을 확인할 수 있습니다.



NOTE: 초기에 설정한 GX Developer 의 PLC 프로젝트에 GX Configurator 에서 생성한 파라미터의 설정값이 포함 됩니다. 새로운 GX Developer 프로젝트를 생성시 GX Configurator 의 설정을 새로 시작 해야 합니다.

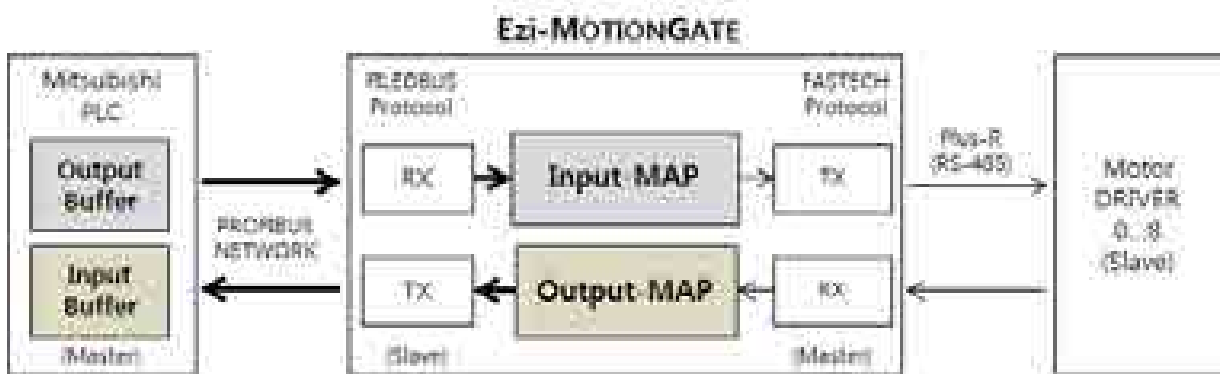
5.2. PLC LADDER 프로그래밍 예제

■ 데이터 교환 버퍼 영역

모션게이트는 PROFIBUS 네트워크를 통하여 상호간의 데이터 교환을 워드단위(16bit)의 구간을 갖습니다

- ✓ Input Buffer MIT Address
- ✓ Output Buffer MIT Address

본 매뉴얼에서 말하는 모션게이트의 Input Map 은 PLC 상의 PROFIBUS 마스터 설정 값인 Output Buffer 로 정의 됩니다. 그리고 모션게이트의 Output Map 은 Input Buffer 로 정의 됩니다.



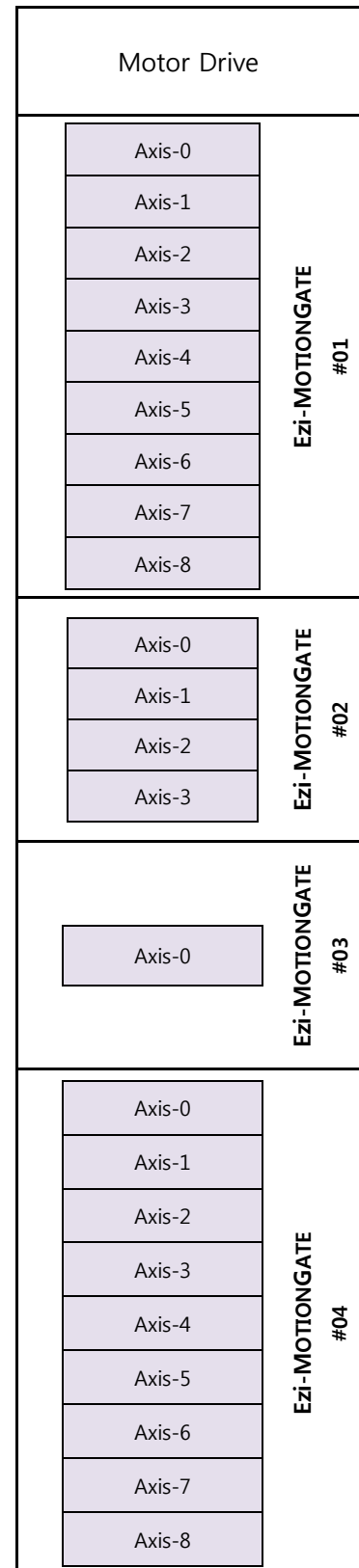
■ PROFIBUS Buffer 주소의 할당

모션게이트의 IO-Map 데이터 어드레스 영역은 상위 제어기의 PROFIBUS Buffer 를 최대 152Byte 까지 사용할 수 있습니다. 그러나하나의 모션게이트의 연결시 Input 72byte Output 72byte 의 버퍼메모리 공간을 점유하기 때문에, PROFIBUS 의 슬레이브 장비의 설정시, 슬레이브의 버퍼 할당은 Input 에 72byte, Output 72Byte 를 설정 합니다.

모션게이트는 하나의 모터 드라이브에 8Byte 의 Input/Output 버퍼가 필요하므로 모션게이트가 9 축 모두 사용하기 위해서 Input/Output 을 72byte 를 할당해야 합니다. 이때 슬레이브 1 은 xB#00 번지부터 사용하게 되며, 4 축을 사용하는 슬레이브 2 의 모션게이트는 xB#72 번지부터 32byte 범위로 할당됩니다. 또한 1 축을 사용하는 슬레이브 3 의 모션게이트는 슬레이브 2 의 주소에 이어서 xB#104 번지부터 8byte 범위로 할당됩니다. 이후에 지정되는 슬레이브는 할당된 버퍼의 끝 주소인 xB#112 번지부터 이어서 할당 됩니다.

PROFIBUS Buffer 시작 주소

Node Address	Buffer Start address			Slave Modules
	DWORD [ID/QD]	WORD [IW/QW]	BYTE [IB/QB]	
SLAVE 1	0	0	0	In/Out. 0 [8 BYTE]
	2	4	8	In/Out. 1 [8 BYTE]
	4	8	16	In/Out. 2 [8 BYTE]
	6	12	24	In/Out. 3 [8 BYTE]
	8	16	32	In/Out. 4 [8 BYTE]
	10	20	40	In/Out. 5 [8 BYTE]
	12	24	48	In/Out. 6 [8 BYTE]
	14	28	56	In/Out. 7 [8 BYTE]
	16	32	64	In/Out. 8 [8 BYTE]
SLAVE 2	18	36	72	In/Out. 0 [8 BYTE]
	20	40	80	In/Out. 1 [8 BYTE]
	22	44	88	In/Out. 2 [8 BYTE]
	24	48	96	In/Out. 3 [8 BYTE]
SLAVE 3	26	52	104	In/Out. 0 [8 BYTE]
SLAVE 4	28	56	112	In/Out. 0 [8 BYTE]
	30	60	120	In/Out. 1 [8 BYTE]
	32	64	128	In/Out. 2 [8 BYTE]
	34	68	136	In/Out. 3 [8 BYTE]
	36	72	144	In/Out. 4 [8 BYTE]
	38	76	152	In/Out. 5 [8 BYTE]
	40	80	160	In/Out. 6 [8 BYTE]
	42	84	168	In/Out. 7 [8 BYTE]
	44	88	176	In/Out. 8 [8 BYTE]



■ PLC PROFIBUS 마스터 모듈의 데이터 교환 시작 명령

미쯔비시 PLC 의 PROFIBUS 마스터 모듈은 PROFIBUS 네트워크의 데이터 교환 여부를 결정하는 비트가 존재 합니다. 또한 데이터 교환 상태를 확인 할 수 있는 비트가 있습니다.

본 예제에서는 이 기능이 지속적으로 실행 한다는 조건에서 실행하는 예제 입니다. 또한, PROFIBUS 마스터 모듈인 QJ71PB92V 가 PLC 슬롯 0 에 삽입 되었을 경우에 대한 명령입니다. (5.1 의 예제 시스템 구성을 참조).

X 로 시작하는 장치 번호는 QJ71PB92V 에서 QCPU 의 입력 신호를 표시하고, Y 로 시작하는 장치 번호는 QCPU 에서 QJ71PB92V 로 출력 신호를 나타냅니다. 또한 SM400 은 미쯔비시 PLC 의 특수 릴레이로 항상 ON 상태를 유지하는 릴레이 입니다. 이로써 처음 PLC 의 부팅 후 PROFIBUS 의 데이터 교환 시작명령이 실행 되도록 하는 명령이 됩니다.

예제 1. PROFIBUS 네트워크 데이터교환 승인 프로그램



※ 선두 I/O 의 주소가 000 일 때

- ① 입력 디바이스의 시작 주소 : X0
- ② 출력 디바이스의 시작 주소 : Y0

NOTE: 예제 1 은 PROFIBUS 네트워크의 교환 시작 명령으로 Y0 의 비트를 ON 하지 않을 경우 모든 슬레이브의 장비와 통신이 이루어 지지 않습니다.

■ CONNECT

CONNECT 명령은 해당축의 사용을 결정시 사용합니다. 예제 2. 는 Axis-0 과 Axis-1 에 대한 CONNECT 명령에 대한 예제 입니다.

예제 2. 각 축의 활성화 명령 및 응답 비트 확인

✓ 제어 비트맵의 주소

- Input-Map

CONNECT – D000

- Output-Map

CONNECTED – D100

✓ 입출력 정보

- 입력신호

CONNECT - X500

- 출력신호

CONNECT RESP - Y600

✓ IO-Map 의 명령 및 응답 형태

- CONNECT 명령의 실행 전

Input-Map								
	T	B	I	B	I	C	I	C
D000	V	I	O	C	B	A	B	I
	T	B	I	A	I	A	I	C
D001	V	I	O	C	B	A	B	I
D002								
D003								
D004								

CONNECT bit = 0

Output-Map								
	V	I	O	C	B	A	B	I
D100	V	I	O	C	B	A	B	I
	V	I	O	C	B	A	B	I
D101	V	I	O	C	B	A	B	I
	V	I	O	C	B	A	B	I
D102								
D103								
D104								

CONNECTED bit = 0

- CONNECT 명령의 실행 후

Input-Map								Output-Map							
D000	T	0	1	1	1	1	1	D100	T	0	1	1	1	1	1
	V	1	0	0	0	0	0		V	1	0	0	0	0	0
D001	T	0	1	1	1	1	1	D101	T	0	1	1	1	1	1
	V	1	0	0	0	0	0		V	1	0	0	0	0	0
D002								D102							
D003								D103							

CONNECT bit = 1

CONNECTED bit = 1

LADDER



NOTE: 예제 2.은 Axis-0 의 이후의 예제에 포함 되는 기본 예제이며, 이 명령이 실행된 상태에서 이후의 예제가 실행 가능합니다.

✓ 명령 순서

- ① X500 의 입력으로, 0 번축의 CONNECT 비트(D0.0)를 세트 합니다.
- ② Axis-0 의 접속이 정상 접속이 되었을 시 응답 데이터가 출력됩니다.
- ③ 0 번축 사용 명령에 대한 응답 상태를 Y600 에 출력하여 확인 합니다.
- ④ 만약 접속이 되지 않았을 경우 D100 의 0 번 비트를 제외한 모든 4WORD 의 데이터 레지스터의 비트는 1 로 세트 되어 있습니다.



주의

모터 드라이브의 상태 정보, 명령 응답, 응답 데이터의 모든 정보는 **CONNECTED** 비트 [Output-Map 0:0]가 '1'이라는 조건을 포함 해야 합니다.

■ ENABLE 명령과 E-ETOP 명령

ENABLE 명령은 E-STOP 명령이 비활성화 된 상태에서 동작됩니다. 예제 3은 Axis-0 의 모터 활성화 명령과 E-STOP 명령에 대한 예제 입니다.

예제 3. 모터 활성화 명령과 비상정지 명령

✓ 제어 비트맵의 주소

- Input-Map

ENABLE - D000.1

E-STOP - D002

- Output-Map

ENABLED - D100.1

E-STOP_RESP - D100.2

✓ 입출력 정보

- 입력신호

ENABLE - X501

E-STOP - X502

- 출력신호

ENABLE RESP - Y601

E-STOP LoopBack - Y602

✓ IO-Map 의 명령 및 응답 형태

- ENABLE 명령을 실행

Input-Map									
D000	0	0	0	0	0	1	0	0	0
D001	0	0	0	0	0	0	0	0	0
D002	0	0	0	0	0	0	0	0	0
D003	0	0	0	0	0	0	0	0	0
D004	0	0	0	0	0	0	0	0	0

ENABLE bit = 1

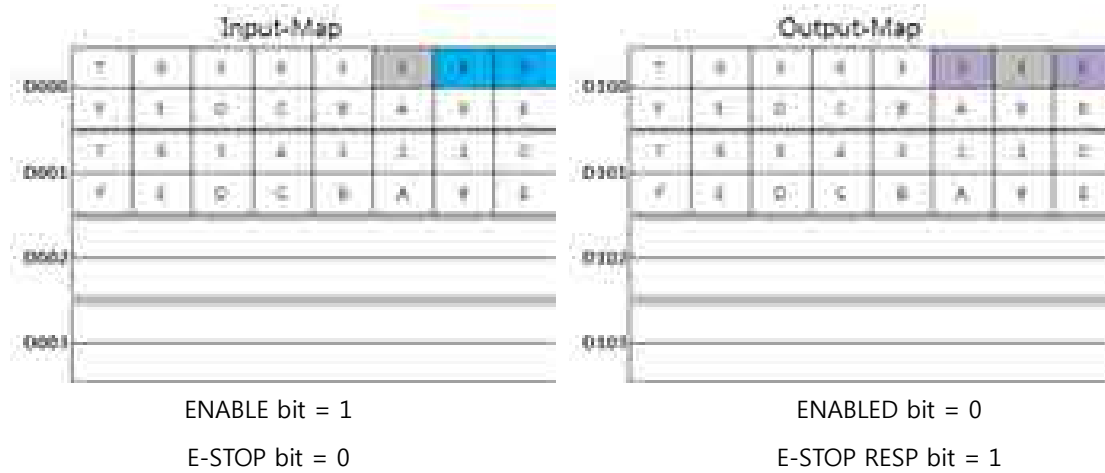
E-STOP bit = 1

Output-Map									
D100	0	0	0	0	0	0	0	0	0
D101	0	0	0	0	0	0	0	0	0
D102	0	0	0	0	0	0	0	0	0
D103	0	0	0	0	0	0	0	0	0
D104	0	0	0	0	0	0	0	0	0

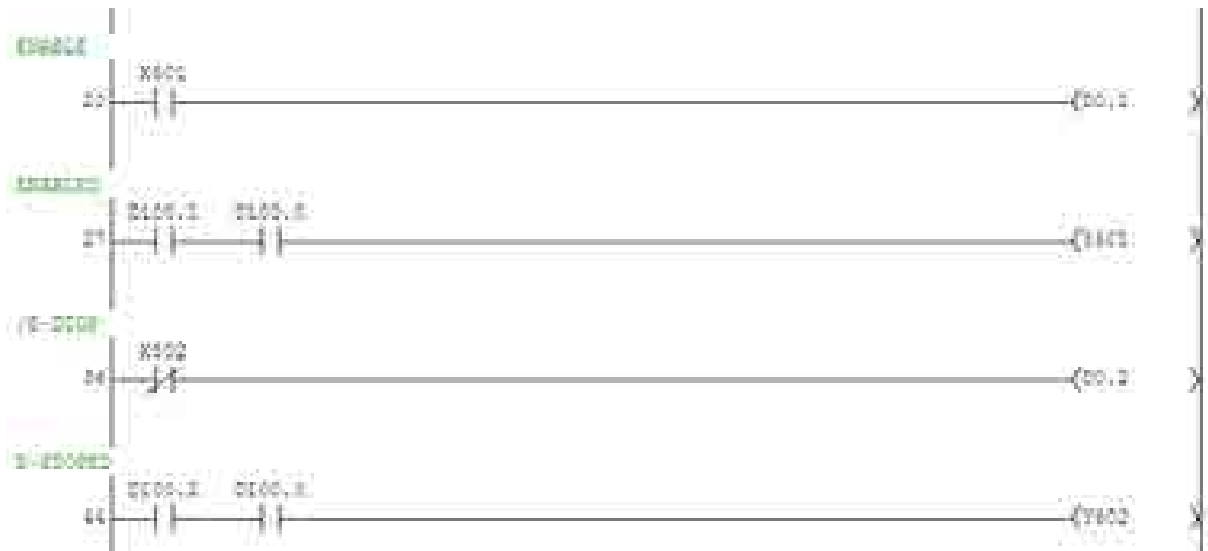
ENABLED bit = 1

E-STOP RESP bit = 0

- E-STOP 명령을 실행



✓ LADDER



NOTE: 예제 3의 래더는 한 축(Axis-0)에 대한 명령입니다. 이 예제는 이 후의 예제에서 포함됩니다.

✓ 명령 순서

- ① 모터 활성화 명령은 X501의 입력으로 ENABLE 비트(Y1)를 세트합니다.
- ② 모터의 활성화 상태가 되었을 때, 상태비트인 ENABLED 비트(X1001)의 상태로 Y601에 출력됩니다.
- ③ X502의 N/C 입력으로 /E-STOP 비트를 제어합니다. 만약 X502의 입력이 Close 상태이면 X501로 ENABLE 명령을 실행하여도 ENABLE 명령은 동작되지 않으며, E-STOPED 비트(D100.2)가 활성화됩니다. 또한 모든 모션제어 명령은 /E-STOP 명령이 활성화된 상태에서 명령제어가 가능합니다.

■ ALARM 상태 확인

알람상태는 ALM/ERR 비트로 확인할 수 있습니다. 예제 4. 는 Axis-0 의 알람상태 확인 방법과 알람 해제에 대한 예제 입니다.

예제 4. 알람상태 확인 및 알람해제 명령

✓ 제어 비트맵의 주소

- Input-Map

ENABLE – D000.1
E-STOP – D000.2
ALARM_RESET – D000.3

- Output-Map

ENABLED – D100.1
E-STOP_RESP – D100.2
ALARM_ERROR – D100.3

✓ 입출력 정보

- 입력신호

ALM_RST – X503

- 출력신호

ALM_STAT – Y603

✓ IO-Map 의 명령 및 응답 형태

- 알람 발생 시

Input-Map									
D000	T	R	I	R	R	ALM_RST	0	1	0
D001	T	R	I	A	A	1	1	0	0
D002	T	R	I	C	B	A	1	0	0
D003									
D004									
D005									

ALARM RESET bit = 0

ENABLE bit = 1

E-STOP bit = 1

Output-Map									
D100	T	R	I	R	R	ALM_STAT	1	0	0
D101	T	R	I	A	A	1	0	0	0
D102	T	R	I	C	B	A	0	1	0
D103									
D104									
D105									

ALARM bit = 1

ENABLED bit = 0

E-STOP RESP bit = 0

- 알람 해제 명령을 실행

Input-Map								Output-Map							
D000	T	B	I	E	P	C	A	D100	T	B	I	E	P	C	A
D001	V	I	O	C	B	A	F	D101	V	I	O	C	B	A	F
D002	T	B	I	A	I	I	C	D102	T	B	I	A	I	I	C
D003	F	I	O	C	B	A	F	D103	F	I	O	C	B	A	F
D004								D104							
D005								D105							
D006								D106							
D007								D107							
D008								D108							
D009								D109							
D010								D110							

ALARM RESET bit = 1

ENABLE bit = 1

E-STOP bit = 1

ALARM bit = 0

ENABLED bit = 0

E-STOP RESP bit = 0

✓ LADDER



NOTE: 예제 4 의 래더는 예제 3 에 추가된 명령으로, 핸드셰이크와 CONNECT, ENABLE, E-STOP 명령이 생략되었습니다.

✓ 명령 순서

- ① 알람 상태는 ALM/ERR 비트(D100.3)의 상태로 Y603 에 출력됩니다.
- ② 발생한 알람은 X503 의 Close 입력으로, 발생한 알람을 리셋 할 수 있습니다.
- ③ 알람이 발생되면 모터는 비활성화 되고, ENABLE 비트(D100.1)는 Open 상태가 됩니다.

■ CANCEL 명령

실행취소 명령은 모션정지, 일시정지 명령 취소, PT 운전 정지를 위하여 사용됩니다. 예제 5 는 Axis-0 의 실행취소 명령에 대한 예제 입니다.

예제 5. 실행취소 명령

✓ 제어 비트맵의 주소

- Input-Map

MOTION/SETTING – D000.7

CANCEL – D000.A

- Output-Map

MOTION/SETTING_RESP – D100.7

MOTIONING – D100.A

✓ 입출력 정보

- 입력신호

CANACLE – X504

- 출력신호

MOTIONING – Y604

✓ IO-Map 의 명령 및 응답 형태

- CANCEL 명령을 실행

Input-Map								
D000	0	0	0	0	0	0	0	0
D001	1	1	0	0	0	0	0	0
D002	1	1	1	1	1	1	1	1
D003	1	1	0	0	0	0	0	0
D004								
D005								
D006								
D007								

MOTION/SETTING bit = 0

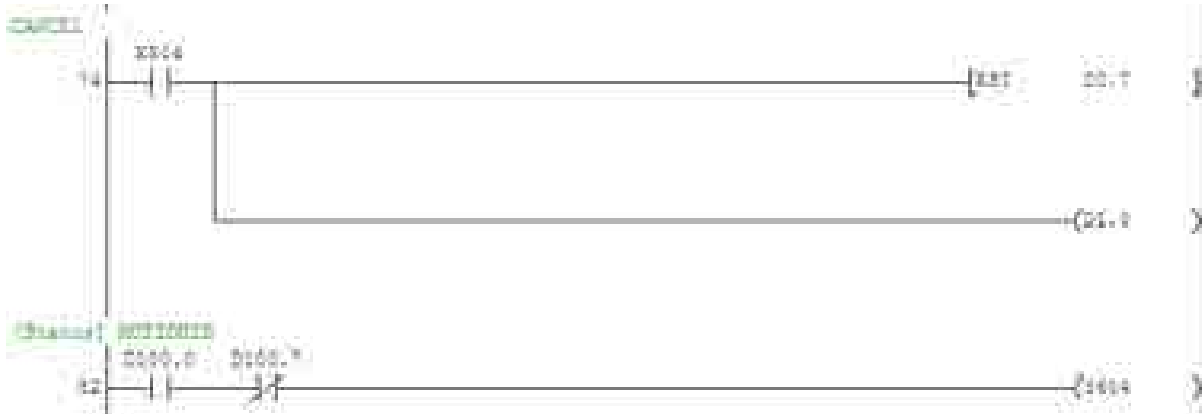
CANCEL bit = 1

Output-Map								
D100	0	0	0	0	0	0	0	0
D101	1	1	0	0	0	0	0	0
D102	1	1	1	1	1	1	1	1
D103	1	1	0	0	0	0	0	0
D104								
D105								
D106								
D107								

MOTION/SETTING_RESP bit = 0

MOTIONING bit = 0

✓ LADDER



NOTE: 예제 5 의 래더는 예제 4 에 추가된 명령으로, 앞의 예제 내용이 생략되었습니다.

✓ 명령 순서

- ① X504 의 Close 입력으로 MOTION/SETTING(Y0007)비트를 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② 실행취소 명령은 CMD_CODE 의 값과 무관하게 CANCEL 비트(D000.A)가 ON 이 될 때 적용됩니다.
- ③ 0 번축의 모션이 실행 중 일 때 MOTIONING 비트(D100.A)는 Close 상태가 되어 Y604 로 출력됩니다. 그러나 X504 를 통한 실행 취소시 MOTIONING 비트(D100.A)는 Open 상태가 됩니다.

* 만약 PT 운전중에서 CANCEL 명령이 입력되면, PT 운전은 정지 됩니다.

■ HOLD 명령

일시정지 명령은 모션 명령을 일시적으로 정지하고, 재개 할 수 있도록 하는 명령입니다. 예제 6. 는 Axis - 0 의 일시정지 명령에 대한 예제 입니다.

예제 6. Axis - 0의 일시정지 명령

✓ 제어 비트맵의 주소

- Input-Map

MOTION/SETTING – D000.7

HOLD – D00.B

- Output-Map

MOTION/SETTING_RESP – D100.7

MOTIONING – D100.A

HOLD_RESP – D100.B

✓ 입출력 정보

- 입력신호

HOLD – X505

- 출력신호

MOTIONING – Y604

HOLD Resp – Y605

✓ IO-Map 의 명령 및 응답 형태

- HOLD 명령을 실행하여 모션 일시정지

Input-Map									
D000	0	0	0	0	0	0	0	0	0
D001	0	0	0	0	0	0	0	0	0
D002	0	0	0	0	0	0	0	0	0
D003	0	0	0	0	0	0	0	0	0
D004	0	0	0	0	0	0	0	0	0
D005	0	0	0	0	0	0	0	0	0
D006	0	0	0	0	0	0	0	0	0
D007	0	0	0	0	0	0	0	0	0
D008	0	0	0	0	0	0	0	0	0
D009	0	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

HOLD bit = 1

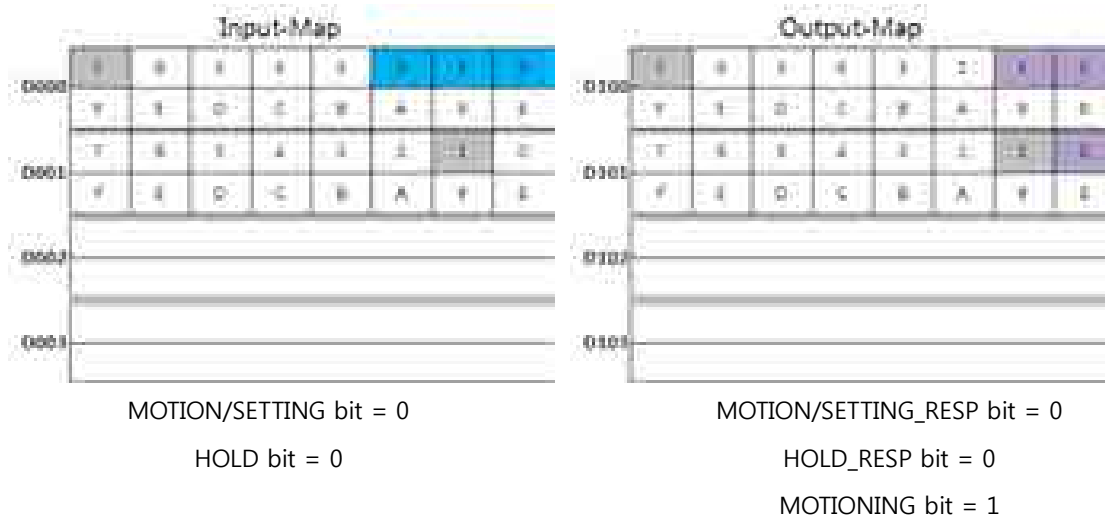
Output-Map									
D100	0	0	0	0	0	0	0	0	0
D101	0	0	0	0	0	0	0	0	0
D102	0	0	0	0	0	0	0	0	0
D103	0	0	0	0	0	0	0	0	0
D104	0	0	0	0	0	0	0	0	0
D105	0	0	0	0	0	0	0	0	0
D106	0	0	0	0	0	0	0	0	0
D107	0	0	0	0	0	0	0	0	0
D108	0	0	0	0	0	0	0	0	0
D109	0	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

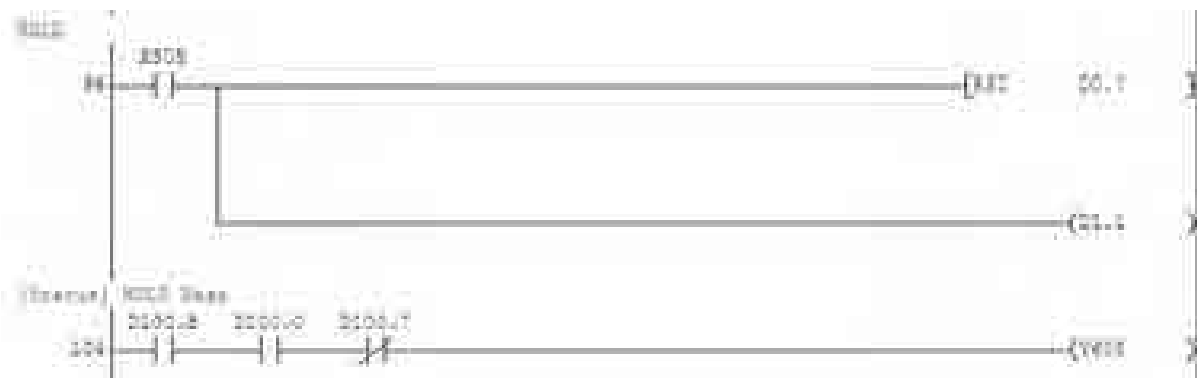
HOLD_RESP bit = 1

MOTIONING bit = 0

- HOLD 명령을 해제하여 모션 재개



✓ LADDER



NOTE: 예제 6. 의 래더는 예제 5. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

✓ 명령 순서

- ① X505 의 Close 입력으로 MOTION/SETTING(D000.7)비트를 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② 실행취소 명령은 CMD_CODE 의 값과 무관하게 HOLD 비트(D000.B)이 ON 이 될 때 적용됩니다.
- ③ HOLD 명령이 실행된 상태에서 X505 가 Open 입력이 되면, HOLD 비트 (D000.B)이 OFF 로 전환되어 일시정지된 모션이 재개 됩니다.
- ④ HOLD 명령이 실행되면 HOLD_RESP 비트(D100.B)가 Close 상태가 되어 Y605 로 출력됩니다.
- ⑤ HOLD 명령의 동작상태는 Input 버퍼로부터 수신되는 MOTION/SETTING_RESP. 비트 (D100.7)의 Close 상태에서 HOLD_RESP. 비트 (D100.B)로 확인할 수 있습니다.

■ RESPONSE TYPE 설정

RESPONSE TYPE(응답 데이터 설정) 설정은 모션모드(MOTIONION)상태에서 확인할 수 있습니다.
예제 7 은 Axis-0 의 RESPONSE TYPE 설정 방법에 대한 예제 입니다.

예제 7. 응답 데이터 설정

✓ 제어 비트맵의 주소

- Input-Map

MOTION/SETTING – D000.7
RESPONSE_TYPE 0~3 – D000.C~D000.F

- Output-Map

MOTION/SETTING_RESP – D100.7
RESPONSE_TYPE_RESP 0~3 – D100.C~D100.F
RESPONSE_DATA – D102~D103 (D102 [1 DWORD])

✓ 입출력 정보

- 입력신호

Response Type 0~3

- 출력데이터(DWORD)

Command Position – D2000
Actual Position – D2002
Position Error – D2004
Actual Velocity – D2006
Current PT No – D2008
Current Alarm No – D2010

✓ IO-Map 의 명령 및 응답 형태

- 응답 데이터를 요청하지 않음

Input-Map							
0000	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0
0004	0	0	0	0	0	0	0
0005	0	0	0	0	0	0	0
0006	0	0	0	0	0	0	0
0007	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

RESPONSE_TYPE 0~3 = 0000b

Output-Map							
0100	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0
0104	0	0	0	0	0	0	0
0105	0	0	0	0	0	0	0
0106	0	0	0	0	0	0	0
0107	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

RESPONSE_TYPE 0~3 = 0000b

RESPONSE_DATA = 0

- 지령 위치 값(Command Position)을 요청

Input-Map							
0000	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0
0004	0	0	0	0	0	0	0
0005	0	0	0	0	0	0	0
0006	0	0	0	0	0	0	0
0007	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

RESPONSE_TYPE 0~3 = 0001b

Output-Map							
0100	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0
0104	0	0	0	0	0	0	0
0105	0	0	0	0	0	0	0
0106	0	0	0	0	0	0	0
0107	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

RESPONSE_TYPE 0~3 = 0001b

RESPONSE_DATA = Command Position

- 현재 위치 값(Actual Position)을 요청

Input-Map							
0000	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0
0004	0	0	0	0	0	0	0
0005	0	0	0	0	0	0	0
0006	0	0	0	0	0	0	0
0007	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

RESPONSE_TYPE 0~3 = 0010b

Output-Map							
0100	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0
0104	0	0	0	0	0	0	0
0105	0	0	0	0	0	0	0
0106	0	0	0	0	0	0	0
0107	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

RESPONSE_TYPE 0~3 = 0010b

RESPONSE_DATA = Actual Position

- 위치 오차 값(Position Error)을 요청

Input-Map							
(X000)	0	0	0	0	0	1	0
(X001)	0	0	0	0	0	0	0
(X002)	0	0	0	0	0	0	0
(X003)	0	0	0	0	0	0	0

MOTION/SETTING bit = 0
RESPONSE_TYPE 0~3 = 0011b

Output-Map							
(Y100)	0	0	0	0	0	0	0
(Y101)	0	0	0	0	0	0	0
(Y102)	0	0	0	0	0	0	0
(Y103)	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0
RESPONSE_TYPE 0~3 = 0011b
RESPONSE_DATA = Position Error

- 현재 속도 값(Actual Velocity)을 요청

Input-Map							
(X000)	0	0	0	0	0	1	0
(X001)	0	0	0	0	0	0	0
(X002)	0	0	0	0	0	0	0
(X003)	0	0	0	0	0	0	0

MOTION/SETTING bit = 0
RESPONSE_TYPE 0~3 = 0101b

Output-Map							
(Y100)	0	0	0	0	0	0	0
(Y101)	0	0	0	0	0	0	0
(Y102)	0	0	0	0	0	0	0
(Y103)	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0
RESPONSE_TYPE 0~3 = 0101b
RESPONSE_DATA = Actual Velocity

- 현재의 PT 번호 요청(Current PT No.)을 요청

Input-Map							
(X000)	0	0	0	0	0	1	0
(X001)	0	0	0	0	0	0	0
(X002)	0	0	0	0	0	0	0
(X003)	0	0	0	0	0	0	0

MOTION/SETTING bit = 0
RESPONSE_TYPE 0~3 = 0101b

Output-Map							
(Y100)	0	0	0	0	0	0	0
(Y101)	0	0	0	0	0	0	0
(Y102)	0	0	0	0	0	0	0
(Y103)	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0
RESPONSE_TYPE 0~3 = 0101b
RESPONSE_DATA = Current PT No.

- 현재의 알람정보(Current Alarm Info.)을 요청

	Input-Map							
	0	1	2	3	4	5	6	7
0000	0	1	2	3	4	5	6	7
0001	8	9	A	B	C	D	E	F
0002								
0003								

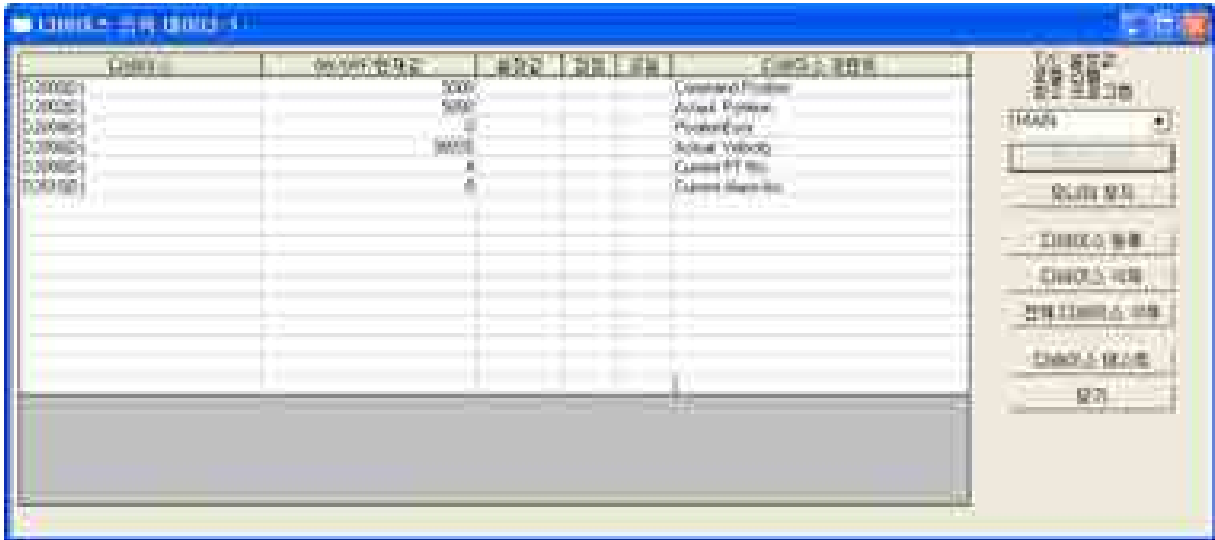
MOTION/SETTING bit = 0
 RESPONSE_TYPE 0~3 = 0100b

	Output-Map							
	0	1	2	3	4	5	6	7
0100	0	1	2	3	4	5	6	7
0101	8	9	A	B	C	D	E	F
0102								
0103								

MOTION/SETTING_RESP bit = 0
 RESPONSE_TYPE 0~3 = 0100b
 RESPONSE_DATA = Current Alarm Info.

✓ LADDER





NOTE: 예제 7 의 명령으로 응답데이터를 분리한 데이터를 모니터링한 결과입니다.

✓ 명령 순서

- ① X506 의 입력으로 MOTION/SETTING(D000.7)비트를 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② 요청하고자 하는 데이터에 대한 코드를 RESPONSE_TYPE 영역(D000.C~D00.F)에 지정 합니다. (예제 7.은 X51A~X51D 의 입력 값에 따라 응답데이터 코드를 지정하고, 각각의 응답 데이터를 D2000~D20010 에 저장하는 명령 입니다.)
- ③ MOTION/SETTING_RESP. 비트(D100.7)가 Close 상태에서 RESPONSE_TPYE_RESP. 영역 (D100.C~D100.F)값이 요청한 응답데이터 코드와 같을 때, 응답 데이터 영역(D100)의 값은 요청한 응답 데이터로 분류 합니다.

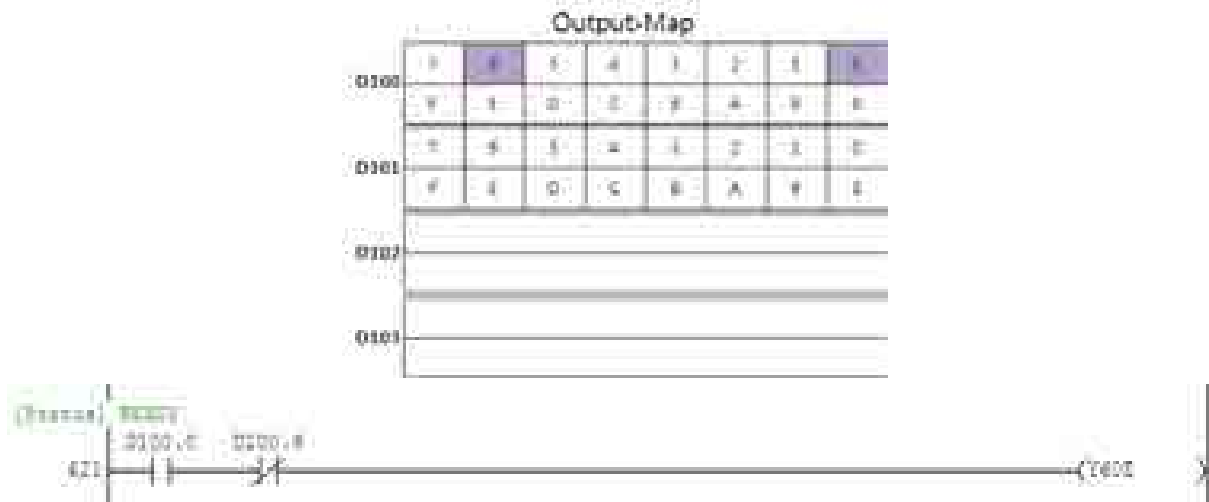
NOTE: 예제 7.의 디바이스 등록 데이터는 응답데이터 형식을 순차적으로 변화 시켜 얻은 데이터로, PT 항목 3 번에서 4 번으로 변경되는 상태에서 추종위치 값, 실제 위치값, 위치오차, 운전속도, PT 번호를 순서대로 얻은 데이터입니다.

■ 상태 정보 확인

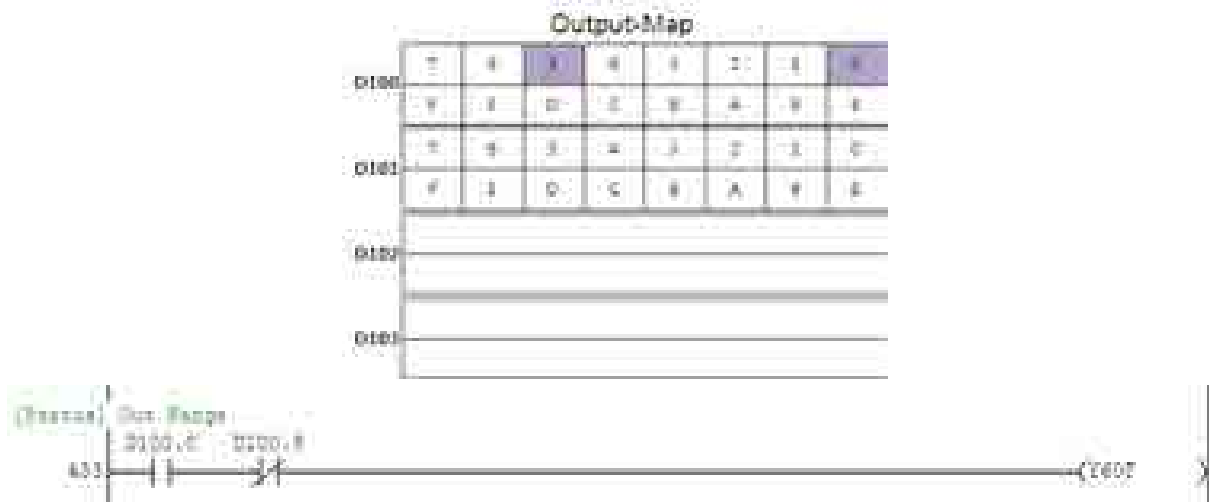
상태 정보 확인은 모션모드(MOTIONION)상태에서 확인할 수 있습니다. 예제 8 은 Axis-0 의 상태 정보 확인 방법에 대한 예제 입니다.

예제 8. 응답 데이터 설정

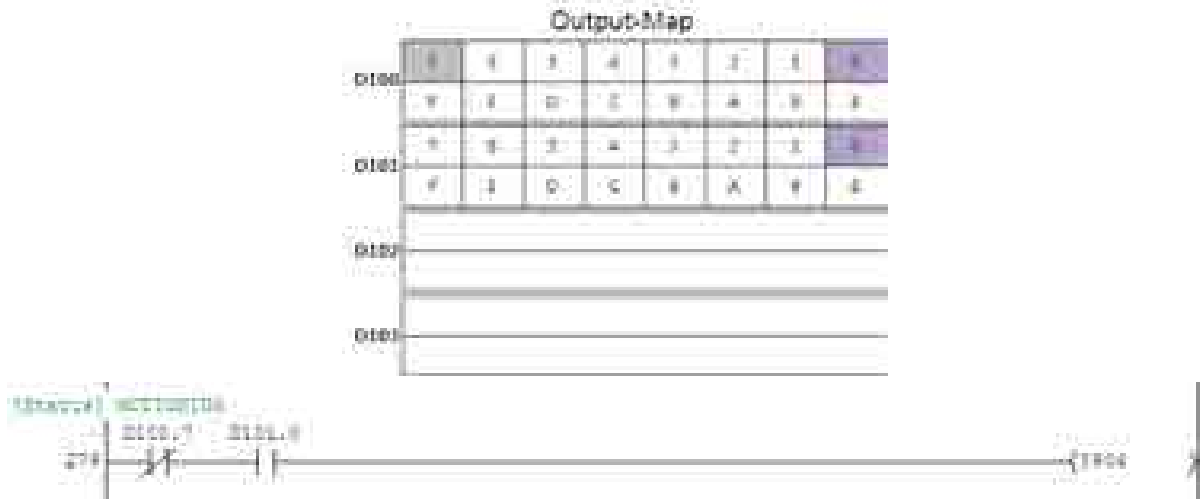
- ✓ READY 비트 : IO-Map 에 대한 명령이 가능 할 때 ON 됩니다.



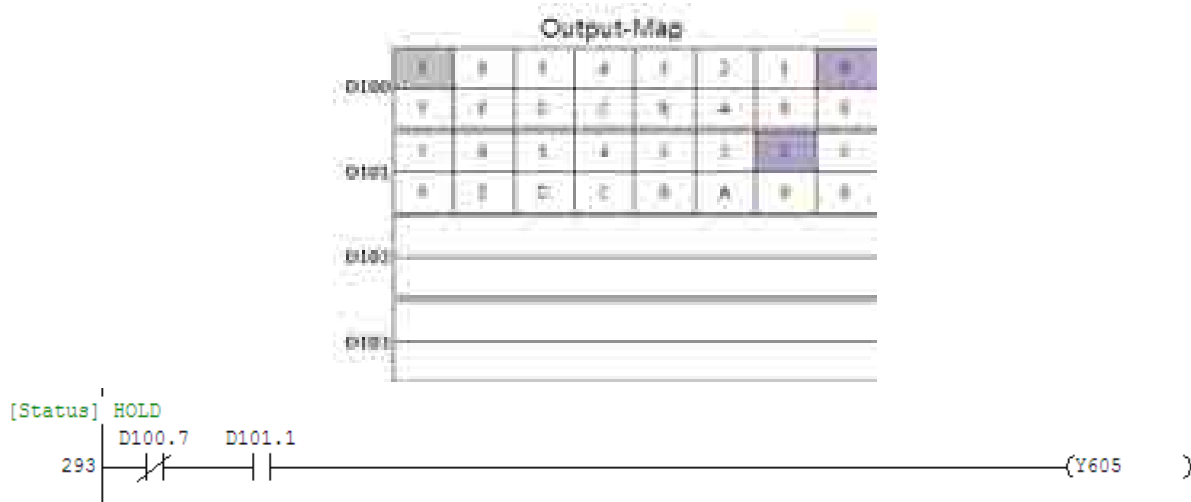
- ✓ Out. Range 비트 : IO-Map 에 대한 명령시 데이터 범위를 초과 하였을 때 ON 됩니다.



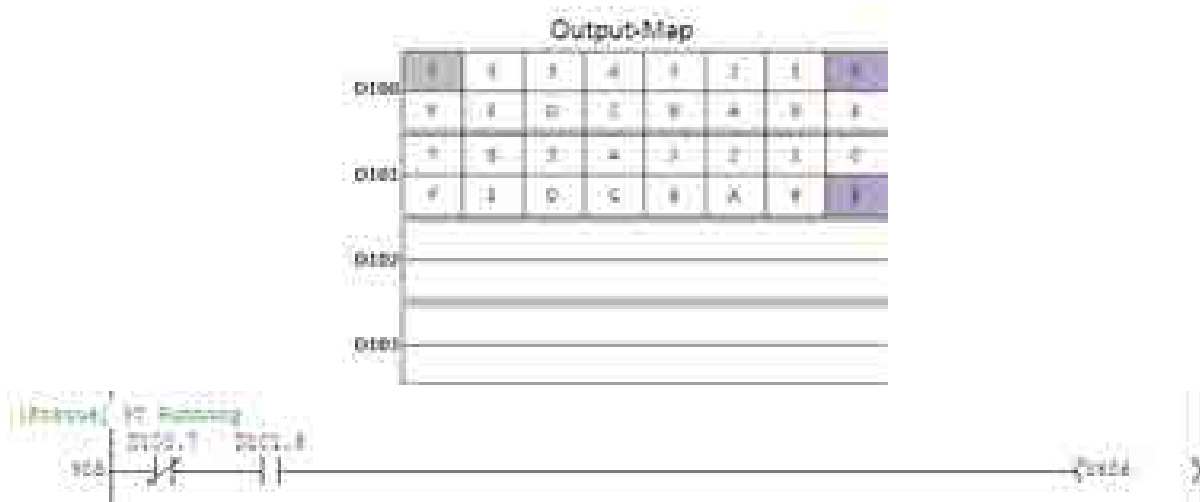
- ✓ MOTIONING 비트 : 모터가 구동중일 때 ON 됩니다.



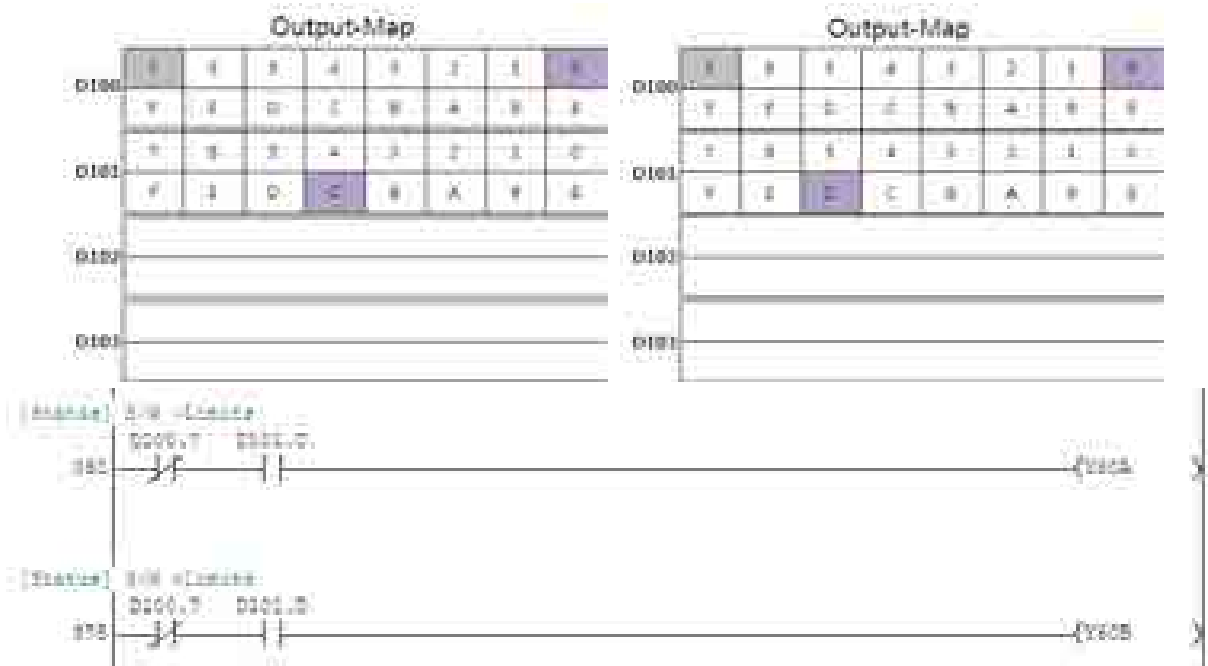
- ✓ HOLD RESP. 비트 : 운전중 일시정지 상태로 되었을 때 ON 됩니다.



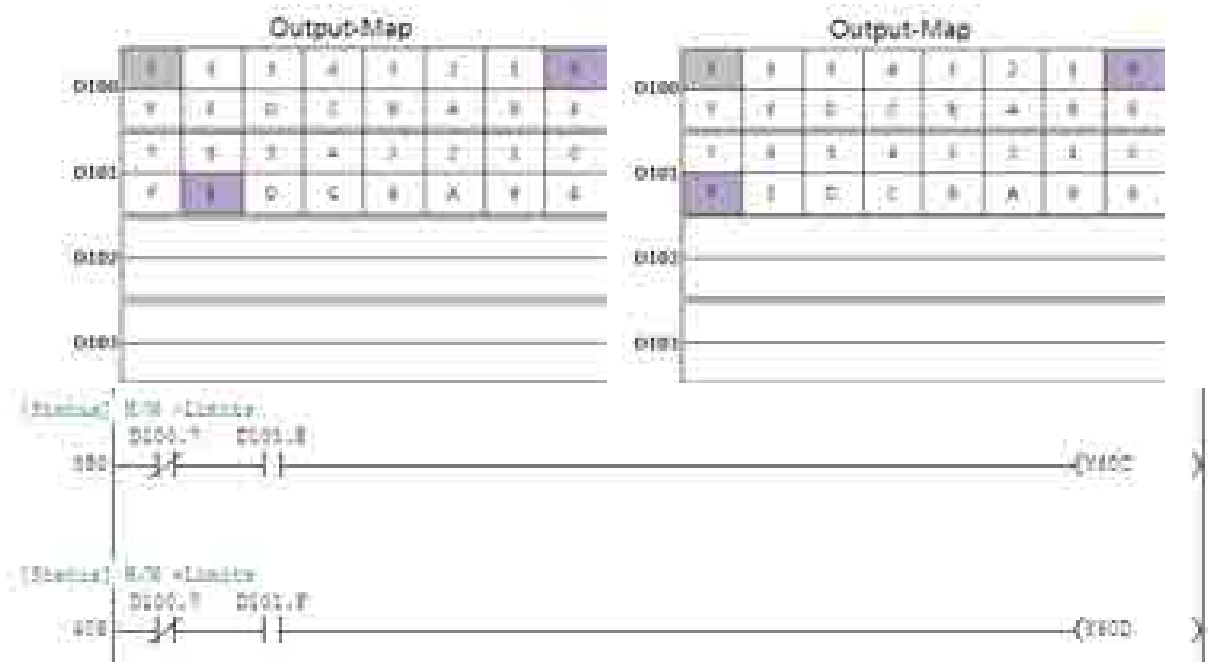
- ✓ PT RUNNING 비트 : PT 운전 상태일 때 ON 됩니다.



- ✓ S/W -LIMIT 비트 : -방향 프로그램 리미트를 초과한 경우 일때 ON 됩니다.
- ✓ S/W +LIMIT 비트 : -방향 프로그램 리미트를 초과한 경우 일때 ON 됩니다.



- ✓ H/W -LIMIT 비트 : -방향 리미트 센서가 ON 이 된경우 일때 ON 됩니다.
- ✓ H/W +LIMIT 비트 : +방향 리미트 센서가 ON 이 된경우 일때 ON 됩니다.



NOTE: 상태정보의 확인은 IO-Map 이 모션모드에서 확인이 가능합니다.

■ CMD START 명령

CMD START 명령은 IO-Map의 모션 모드(MOTION/SETTING = 0)인 위치이동, PT 운전, 원점이동과 설정 모드(MOTION/SETTING = 1)에서의 명령 실행으로 사용됩니다.

이 명령은 예제 9와 예제 10의 PLC 회로로 구성하여 사용합니다.

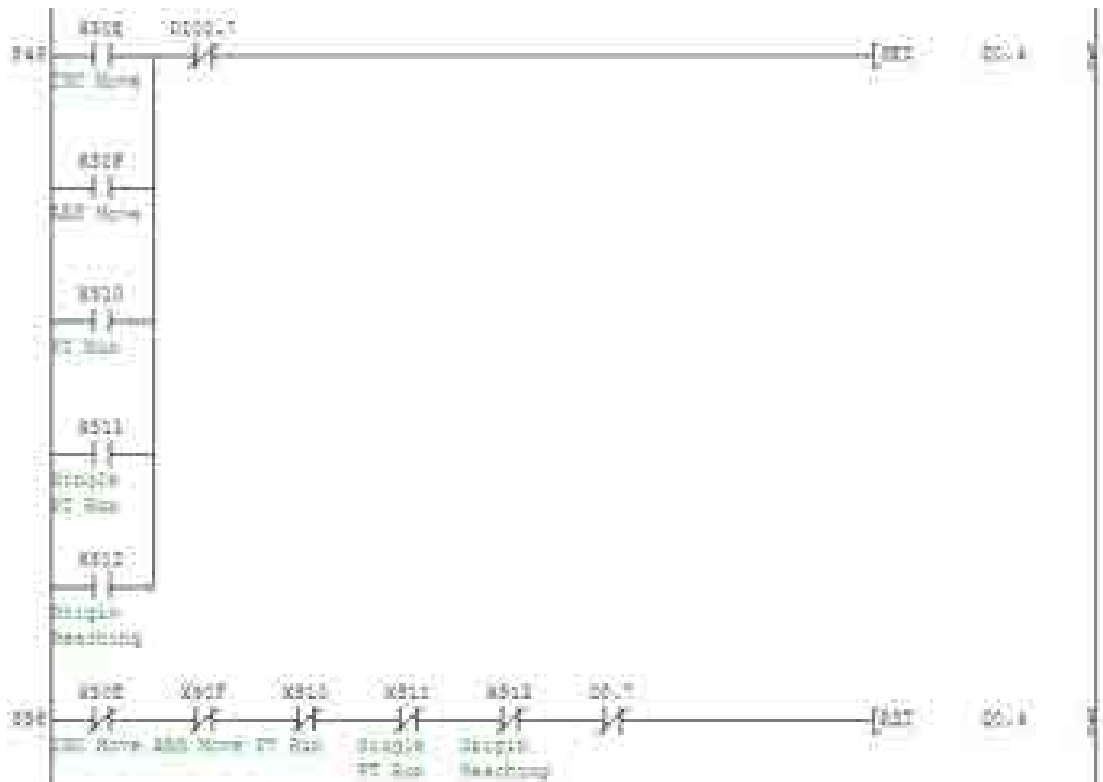
예제 9. 모션 모드의 CMD START 실행 방법

✓ IO-Map의 명령 및 응답 형태

- 모션 모드에서의 명령 및 응답

Input-Map								Output-Map							
0000	0	1	2	3	4	5	6	0100	0	1	2	3	4	5	6
	13	14	15	16	17	18	19		13	14	15	16	17	18	19
0001	7	8	9	10	11	12	13	0101	7	8	9	10	11	12	13
	20	21	22	23	24	25	26		20	21	22	23	24	25	26
0002								0102							
0003								0103							

✓ LADDER



NOTE : 예제 9의 래더는 예제 15, 16, 18, 19, 21에 추가된 명령으로, 해당 명령을 실행하기 위하여 추가된 예제입니다.

✓ 명령 순서

- ① 해당 명령의 Open 입력신호와 명령 선택으로 인하여 변경된 리모트 입력의 D000.7 비트가 NC 상태의 값과, AND 회로로 구성된 신호로 CMD_START 비트(D000.4)를 세트 합니다.
- ② 해당 명령의 입력이 Close 되었을 때, CMD_STAR 비트(D000.4)를 리셋 합니다.

예제 10. 설정 모드의 CMD START 실행 방법

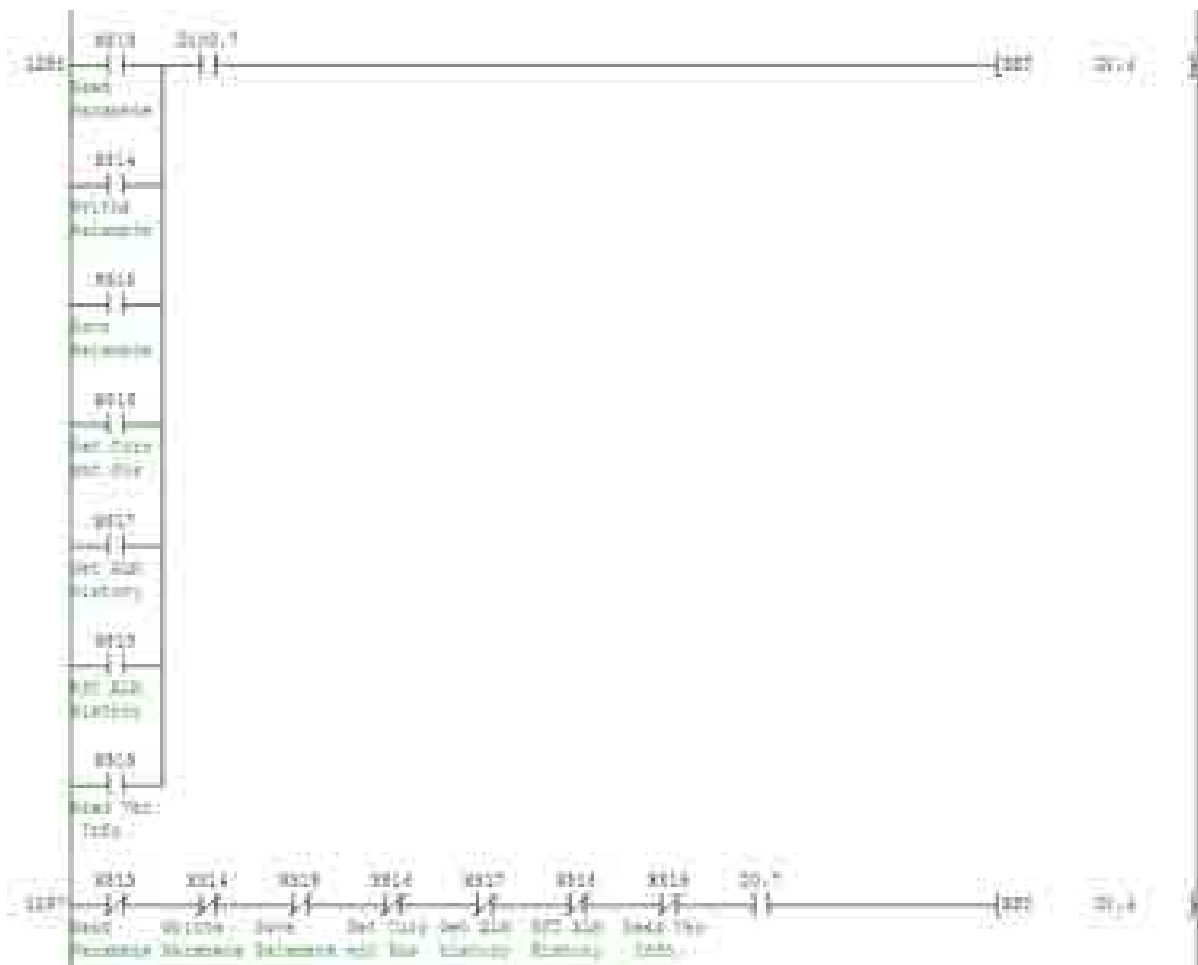
✓ IO-Map의 명령 및 응답 형태

- 모션 모드에서의 명령 및 응답

Bit	0	1	2	3	4	5	6	7
D000	10	10	10	10	10	10	10	10
D001	10	10	10	10	10	10	10	10
D002	10	10	10	10	10	10	10	10
D003	10	10	10	10	10	10	10	10

Bit	0	1	2	3	4	5	6	7
D000	10	10	10	10	10	10	10	10
D001	10	10	10	10	10	10	10	10
D002	10	10	10	10	10	10	10	10
D003	10	10	10	10	10	10	10	10

✓ LADDER



www.fastech.co.kr

NOTE : 예제 10 의 래더는 예제 22~28 에 추가된 명령으로, 해당 명령을 실행하기 위하여 해당 예제를 사용 시 반드시 추가 해야하는 래더입니다.

✓ 명령 순서

- ① 해당 명령의 Open 입력신호와 명령 선택으로 인하여 변경된 리모트 입력의 D000.7 비트가 NO 상태의 값과, AND 회로로 구성된 신호로 CMD_START 비트(D000.4)를 세트 합니다.
- ② 해당 명령의 입력이 Close 되었을 때, CMD_STAR 비트(D000.4)를 리셋 합니다.

5.2.1 조그 운전 명령

조그 운전은 모션모드(MOTIONION)상태의 명령 코드(CMD_CODE) '0'에서 동작합니다. Speed Step Move 또는 Speed Ratio Move 과 Speed Value 의 조그 운전 명령 응답 확인은 JOG_Resp. 비트로 확인 가능합니다.

■ JOG Move – Speed Step Move or Speed Ratio Move

조그 운전의 Speed Step Move 과 Speed Ratio Move 는 운전 명령 방법은 동일합니다. 이 운전 방법은 모션게이트 파라미터 PN#0104 『Use Jog Speed Ratio』의 설정 값으로 선택 됩니다.

조그 운전의 Speed Step Move 는 저장된 0~3 의 속도 단계로 조그 운전하는 모션입니다. 조그 운전의 Speed Ratio Move 는 저장된 파라미터의 PN#0105 『Move Speed for Jog Move: Ratio』의 비율로 운전합니다.

예제 11. Speed Step Move 또는 Speed Ratio Move 형태의 조그 운전 명령

✓ 제어 비트맵의 주소

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D0008~D000.C
 SPD.MODE – D001.A
 -JOG – D000.E
 +JOG – D000.F
 Command Data Area – D002~D003 (D002 [1 DWORD])

- Output-Map

MOTION/SETTING_RESP – D100. 7
 CMD_CODE_RESP – D100.8~D100.B
 JOG_RESP – D101.5

✓ 입출력 정보

- 입력 정보

JOG- : X507
 JOG+ : X508
 JOG Speed Step No. (입력 범위 : 0~3 [DWORD])
 JOG Speed Ratio Value (입력 범위 : 1~255 [DWORD])

- 출력 정보

MOTIONING – Y604
 JOG Resp – Y610

✓ IO-Map 의 명령 및 응답 형태

- -JOG 명령 실행

Input-Map									
0000	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0	0
0004	0	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b , SPD_MODE bit = 0;

+JOG bit = 0, -JOG bit = 1

Output-Map									
0100	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0	0	0
0104	0	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

JOG_RESP bit = 1, MOTIONING bit = 1

- +JOG 명령 실행

Input-Map									
0000	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0	0
0004	0	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b , SPD_MODE bit = 0;

+JOG bit = 1, -JOG bit = 0

Output-Map									
0100	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0	0	0
0104	0	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

JOG_RESP bit = 1, MOTIONING bit = 1

- -JOG , +JOG 비트가 '0'인 상태

Input-Map									
0000	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0	0
0004	0	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b , SPD_MODE bit = 0;

+JOG bit = 0, -JOG bit = 0

Output-Map									
0100	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0	0	0
0104	0	0	0	0	0	0	0	0	0

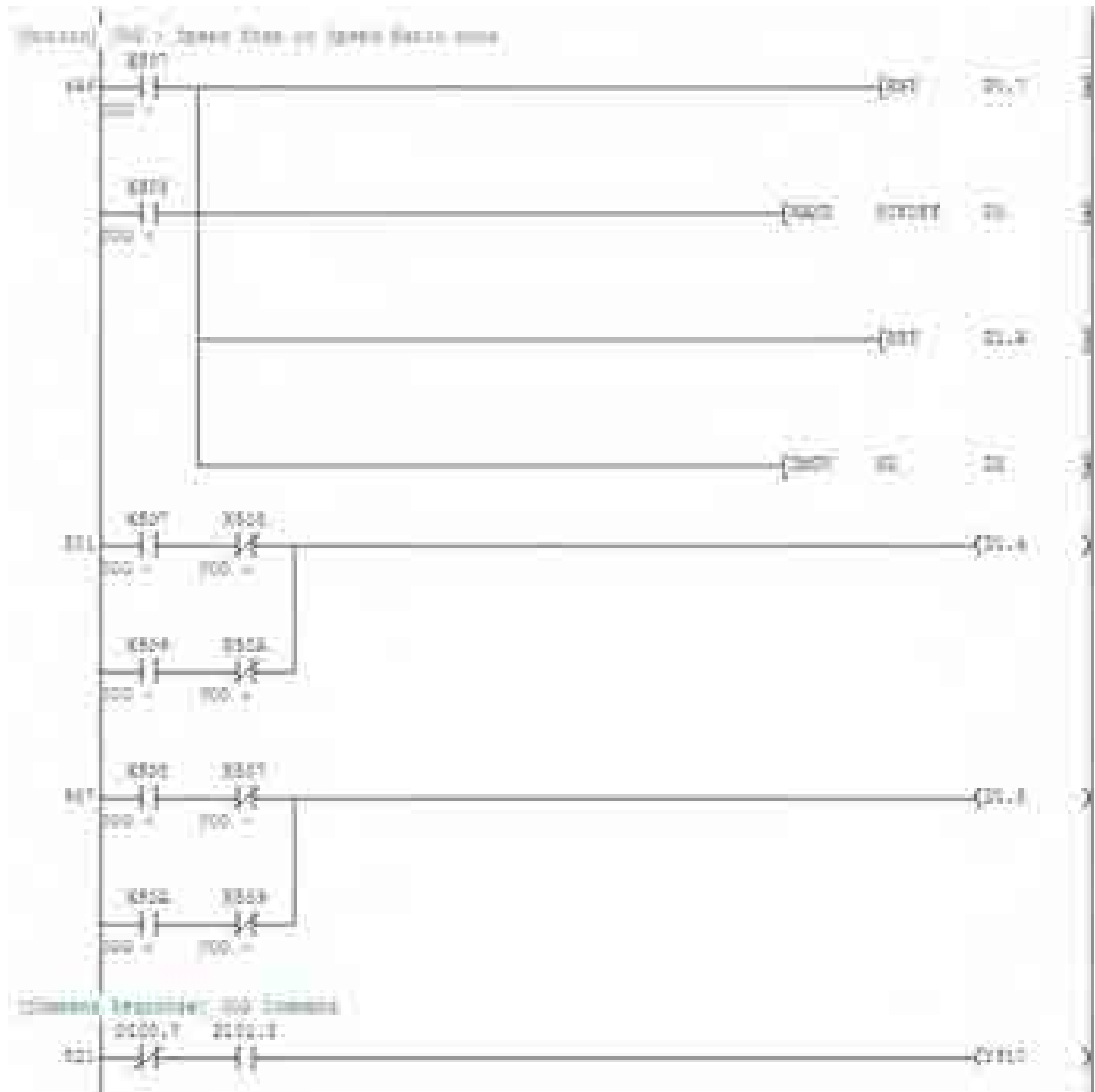
MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

JOG_RESP bit = 0, MOTIONING bit = 0

www.fastech.co.kr

✓ **LADDER**



NOTE 1: 예제 11 의 래더는 예제 7.에 추가된 명령으로, 앞의 예제를 생략 되었습니다.

NOTE 2: 파라미터 PN#0104 『Use Jog Speed Ratio』 값이 '1'일 때 Speed Ratio Move 모드에서 예제 11 를 실행 하면, PN#0105 『Move Speed for Jog Move: Ratio』 값의 2%의 속도로 운전합니다.

NOTE 3: 본 예제는 예제 12의 명령 입력이 포함 되어 있습니다.

✓ 명령 순서

- ① X507 또는 X508 의 Close 입력으로 MOTION/SETTING (D000.7)비트를 클리어 하여 IO-Map 을 모션 모드 상태로 전환 합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '0'을 입력합니다.
- ③ SPD_MODE 비트(D001.A)을 '0'으로 클리어 합니다.
- ④ Speed Step 번호를 Axis-0 의 데이터영역(D002)에 입력 합니다.

(예제 11 에서는 DMOV 명령의 'S'에 Speed Step 의 번호 '2'을 D002 영역에 입력한 래더입니다.)

- ⑤ -JOG 비트(D001.4) 또는 +JOG 비트(D001.5)를 ON 하여 명령을 실행합니다.
- ⑥ X507 또는 X508 의 입력이 Open 되면, 조그운전은 정지 합니다.
- ⑦ 조그 운전 명령에 대한 응답은 MOTION/SETTING_RESP. 비트(D100.7)의 NC 입력과, JOG_Resp. 비트(D101.5)의 AND 회로로 구성하여 확인 가능합니다.

■ JOG Move – Speed Value Move

조그 운전의 Speed Value Move 는 입력된 값을 실제 속도에 적용하여 조그 운전하는 모션입니다.
예제 12 는 Axis-0 의 조그 운전의 Speed Value Move 에 대한 예제 입니다.

예제 12. Speed Value Move 형태의 조그 운전 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 SPD.MODE – D001.A
 -JOG – D001.4
 +JOG – D001.5
 Command Data Area – D002~D002 (D002 [1 DWORD])

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 JOG_RESP – D101.5

✓ 입출력 정보

- 입력 정보

JOG- : X509
 JOG+ : X50A
 JOG Speed Value (입력 범위 : 1~500,000 pps [DWORD])

- 출력 정보

MOTIONING – Y604
 JOG Resp – Y610

✓ IO-Map 의 명령 및 응답 형태

- -JOG 명령 실행

Input-Map

Y0000	0	0	0	0	0	0	0	0
Y0001	0	0	0	0	0	0	0	0
Y0010	0	0	0	0	0	0	0	0
Y0011	0	0	0	0	0	0	0	0
Y0000	0	0	0	0	0	0	0	0
Y0001	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b , SPD_MODE bit = 1;

+JOG bit = 0, -JOG bit = 1

Output-Map

X1000	0	0	0	0	0	0	0	0
X1001	0	0	0	0	0	0	0	0
X1010	0	0	0	0	0	0	0	0
X1011	0	0	0	0	0	0	0	0
X1000	0	0	0	0	0	0	0	0
X1001	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

JOG_RESP bit = 1, MOTIONING bit = 1

- +JOG 명령 실행

Input-Map

Y0000	0	0	0	0	0	0	0	0
Y0001	0	0	0	0	0	0	0	0
Y0010	0	0	0	0	0	0	0	0
Y0011	0	0	0	0	0	0	0	0
Y0000	0	0	0	0	0	0	0	0
Y0001	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b , SPD_MODE bit = 1;

+JOG bit = 1, -JOG bit = 0

Output-Map

X1000	0	0	0	0	0	0	0	0
X1001	0	0	0	0	0	0	0	0
X1010	0	0	0	0	0	0	0	0
X1011	0	0	0	0	0	0	0	0
X1000	0	0	0	0	0	0	0	0
X1001	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

JOG_RESP bit = 1, MOTIONING bit = 1

- -JOG , +JOG 비트가 '0'인 상태

Input-Map

Y0000	0	0	0	0	0	0	0	0
Y0001	0	0	0	0	0	0	0	0
Y0010	0	0	0	0	0	0	0	0
Y0011	0	0	0	0	0	0	0	0
Y0000	0	0	0	0	0	0	0	0
Y0001	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b , SPD_MODE bit = 1;

+JOG bit = 0, -JOG bit = 0

Output-Map

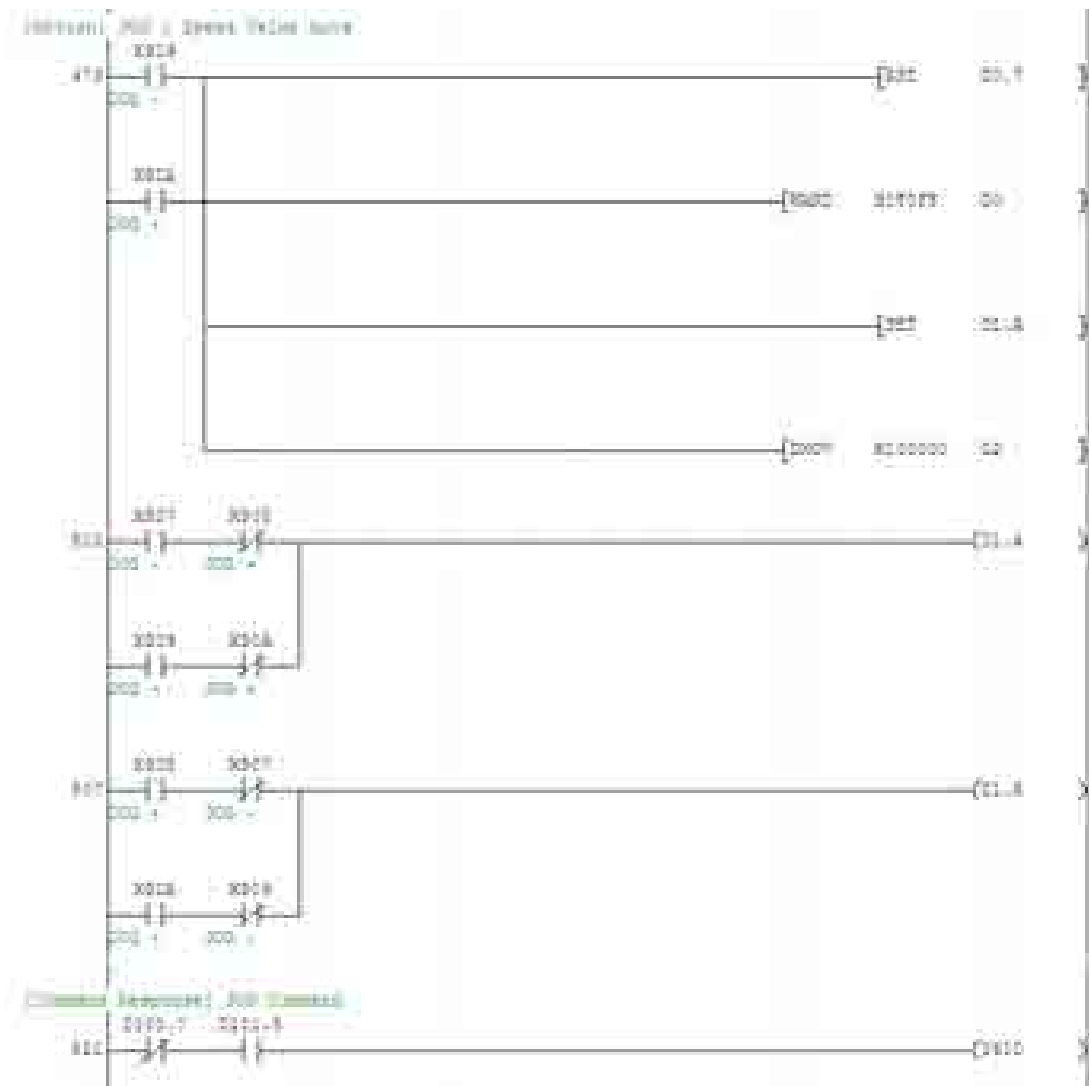
X1000	0	0	0	0	0	0	0	0
X1001	0	0	0	0	0	0	0	0
X1010	0	0	0	0	0	0	0	0
X1011	0	0	0	0	0	0	0	0
X1000	0	0	0	0	0	0	0	0
X1001	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

JOG_RESP bit = 0, MOTIONING bit = 0

✓ **LADDER**



NOTE 1: 예제 12 의 리더는 예제 11 에 추가된 명령으로, 앞의 예제를 생략 되었습니다.

NOTE 2: 본 예제는 예제 11의 명령입력이 포함 되어 있습니다.

✓ 명령 순서

- ① X509 또는 X50A 의 Close 입력으로 MOTION/SETTING(D000.7)비트를 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '0'을 입력합니다.
- ③ SPD_MODE 비트(D001.A)을 '1'로 세트 합니다.
- ④ 조그 운전 속도값을 Axis-0 의 데이터영역(D002)에 입력 합니다.
(예제 12. 에서는 DMOV 명령의 'S'에 조그 운전 속도 100,000pps 을 D002 영역에 입력한 래더입니다.)
- ⑤ -JOG 비트(D001.4)를 ON 하여 명령을 실행합니다.
- ⑥ X509 또는 X50A 의 입력이 Open 되면, 조그운전은 정지 합니다.
- ⑦ 조그 운전 명령에 대한 응답은 MOTION/SETTING_RESP. 비트(D100.7)의 NC 입력과, JOG Resp. 비트(D101.5)의 AND 회로로 구성하여 확인 가능합니다.

5.2.2 스텝 이동 명령

스텝 이동은 모션모드(MOTIONION)상태의 명령 코드(CMD_CODE) '0'에서 동작하며, 스텝이동 거리 번호 0~3 의 값을 지정하여 이동합니다. 예제 13 은 Axis-0 의 스텝이동에 대한 예제 입니다.

예제 13. 스텝 이동 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 -STEP – D001.6
 +STEP – D001.7
 Command Data Area – D002~D003 (D002 [1 DWORD])

- Output-Map

MOTION/SETTING_RESP – D100.7
 MOTIONING – D101.0
 STEP_RESP – D101.7

✓ 입출력 정보

- 입력 정보

STEP- – X50B
 STEP+ – X50C
 STEP Step Distance No. (입력 범위 : 0~3 [DWORD Type])

- 출력 정보

MOTIONING – Y604
 Step Resp – Y611

✓ IO-Map 의 명령 및 응답 형태

- -SETP Move 명령 실행

Input-Map

0000	0	0	0	0	0	1	1	1
0001	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b

+STEP bit = 0, -STEP bit = 1

Output-Map

0100	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

STEP_RESP bit = 1, MOTIONING bit = 1

- + SETP Move 명령 실행

Input-Map

0000	0	0	0	0	0	1	1	1
0001	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b

+STEP bit = 1, -STEP bit = 0

Output-Map

0100	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

STEP_RESP bit = 1, MOTIONING bit = 1

- -STEP, +STEP 비트가 '0'인 상태

Input-Map

0000	0	0	0	0	0	1	1	1
0001	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0
0003	0	0	0	0	0	0	0	0

MOTION/SETTING bit = 0

CMD_CODE = 0000b

+STEP bit = 0, -STEP bit = 0

Output-Map

0100	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0
0102	0	0	0	0	0	0	0	0
0103	0	0	0	0	0	0	0	0

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

STEP_RESP bit = 0, MOTIONING bit = 0

5.2.3 영점 이동 명령

영점 이동은 모션모드(MOTIONION)상태의 명령 코드(CMD_CODE) '0'에서 동작하며, 입력 데이터(D002)의 값은 무시되며 이동합니다. 예제 14 는 Axis-0 의 영점 이동에 대한 예제 입니다.

예제 14. 영점이동 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 GO_ZERO_POS – D001.3

- Output-Map

MOTION/SETTING_RESP – D100.7
 MOTIONING – D101.1
 GO_ZERO_POS_RESP – D101.3

✓ 입출력 정보

- 입력 정보

Go Zero Position – X50D

- 출력 정보

MOTIONING – Y604
 Go Zero POS Resp – Y612

✓ IO-Map 의 명령 및 응답 형태

- 영점 이동 명령을 실행

Input-Map									
D000	0	1	2	3	4	5	6	7	8
D001	0	1	2	3	4	5	6	7	8
D002	0	1	2	3	4	5	6	7	8
D003	0	1	2	3	4	5	6	7	8

MOTION/SETTING bit = 0

CMD_CODE = 0000b

GO_ZERO_POS bit = 1

Output-Map									
D100	0	1	2	3	4	5	6	7	8
D101	0	1	2	3	4	5	6	7	8
D102	0	1	2	3	4	5	6	7	8
D103	0	1	2	3	4	5	6	7	8

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0000b

GO_ZERO_POS_RESP bit = 1,

MOTIONING bit = 1

- 영점 이동 중 GO_ZERO_POS 비트를 해제하여도, 영점 이동상태는 유지

Input-Map							
0000	0	1	2	3	4	5	6
0001	7	8	9	10	11	12	13
0002	14	15	16	17	18	19	20
0003	21	22	23	24	25	26	27
0004	28	29	30	31	32	33	34
0005	35	36	37	38	39	40	41
0006	42	43	44	45	46	47	48
0007	49	50	51	52	53	54	55
0008	56	57	58	59	60	61	62
0009	63	64	65	66	67	68	69
0010	70	71	72	73	74	75	76
0011	77	78	79	80	81	82	83
0012	84	85	86	87	88	89	90
0013	91	92	93	94	95	96	97
0014	98	99	100	101	102	103	104
0015	105	106	107	108	109	110	111
0016	112	113	114	115	116	117	118
0017	119	120	121	122	123	124	125
0018	126	127	128	129	130	131	132
0019	133	134	135	136	137	138	139
0020	140	141	142	143	144	145	146
0021	147	148	149	150	151	152	153
0022	154	155	156	157	158	159	160
0023	161	162	163	164	165	166	167
0024	168	169	170	171	172	173	174
0025	175	176	177	178	179	180	181
0026	182	183	184	185	186	187	188
0027	189	190	191	192	193	194	195
0028	196	197	198	199	200	201	202
0029	203	204	205	206	207	208	209
0030	210	211	212	213	214	215	216
0031	217	218	219	220	221	222	223
0032	224	225	226	227	228	229	230
0033	231	232	233	234	235	236	237
0034	238	239	240	241	242	243	244
0035	245	246	247	248	249	250	251
0036	252	253	254	255	256	257	258
0037	259	260	261	262	263	264	265
0038	266	267	268	269	270	271	272
0039	273	274	275	276	277	278	279
0040	280	281	282	283	284	285	286
0041	287	288	289	290	291	292	293
0042	294	295	296	297	298	299	300
0043	301	302	303	304	305	306	307
0044	308	309	310	311	312	313	314
0045	315	316	317	318	319	320	321
0046	322	323	324	325	326	327	328
0047	329	330	331	332	333	334	335
0048	336	337	338	339	340	341	342
0049	343	344	345	346	347	348	349
0050	350	351	352	353	354	355	356
0051	357	358	359	360	361	362	363
0052	364	365	366	367	368	369	370
0053	371	372	373	374	375	376	377
0054	378	379	380	381	382	383	384
0055	385	386	387	388	389	390	391
0056	392	393	394	395	396	397	398
0057	399	400	401	402	403	404	405
0058	406	407	408	409	410	411	412
0059	413	414	415	416	417	418	419
0060	420	421	422	423	424	425	426
0061	427	428	429	430	431	432	433
0062	434	435	436	437	438	439	440
0063	441	442	443	444	445	446	447
0064	448	449	450	451	452	453	454
0065	455	456	457	458	459	460	461
0066	462	463	464	465	466	467	468
0067	469	470	471	472	473	474	475
0068	476	477	478	479	480	481	482
0069	483	484	485	486	487	488	489
0070	490	491	492	493	494	495	496
0071	497	498	499	500	501	502	503
0072	504	505	506	507	508	509	510
0073	511	512	513	514	515	516	517
0074	518	519	520	521	522	523	524
0075	525	526	527	528	529	530	531
0076	532	533	534	535	536	537	538
0077	539	540	541	542	543	544	545
0078	546	547	548	549	550	551	552
0079	553	554	555	556	557	558	559
0080	560	561	562	563	564	565	566
0081	567	568	569	570	571	572	573
0082	574	575	576	577	578	579	580
0083	581	582	583	584	585	586	587
0084	588	589	590	591	592	593	594
0085	595	596	597	598	599	600	601
0086	602	603	604	605	606	607	608
0087	609	610	611	612	613	614	615
0088	616	617	618	619	620	621	622
0089	623	624	625	626	627	628	629
0090	630	631	632	633	634	635	636
0091	637	638	639	640	641	642	643
0092	644	645	646	647	648	649	650
0093	651	652	653	654	655	656	657
0094	658	659	660	661	662	663	664
0095	665	666	667	668	669	670	671
0096	672	673	674	675	676	677	678
0097	679	680	681	682	683	684	685
0098	686	687	688	689	690	691	692
0099	693	694	695	696	697	698	699
0100	700	701	702	703	704	705	706
0101	707	708	709	710	711	712	713
0102	714	715	716	717	718	719	720
0103	721	722	723	724	725	726	727
0104	728	729	730	731	732	733	734
0105	735	736	737	738	739	740	741
0106	742	743	744	745	746	747	748
0107	749	750	751	752	753	754	755
0108	756	757	758	759	760	761	762
0109	763	764	765	766	767	768	769
0110	770	771	772	773	774	775	776
0111	777	778	779	780	781	782	783
0112	784	785	786	787	788	789	790
0113	791	792	793	794	795	796	797
0114	798	799	800	801	802	803	804
0115	805	806	807	808	809	810	811
0116	812	813	814	815	816	817	818
0117	819	820	821	822	823	824	825
0118	826	827	828	829	830	831	832
0119	833	834	835	836	837	838	839
0120	840	841	842	843	844	845	846
0121	847	848	849	850	851	852	853
0122	854	855	856	857	858	859	860
0123	861	862	863	864	865	866	867
0124	868	869	870	871	872	873	874
0125	875	876	877	878	879	880	881
0126	882	883	884	885	886	887	888
0127	889	890	891	892	893	894	895
0128	896	897	898	899	900	901	902
0129	903	904	905	906	907	908	909
0130	910	911	912	913	914	915	916
0131	917	918	919	920	921	922	923
0132	924	925	926	927	928	929	930
0133	931	932	933	934	935	936	937
0134	938	939	940	941	942	943	944
0135	945	946	947	948	949	950	951
0136	952	953	954	955	956	957	958
0137	959	960	961	962	963	964	965
0138	966	967	968	969	970	971	972
0139	973	974	975	976	977	978	979
0140	980	981	982	983	984	985	986
0141	987	988	989	990	991	992	993
0142	994	995	996	997	998	999	1000

MOTION/SETTING bit = 0

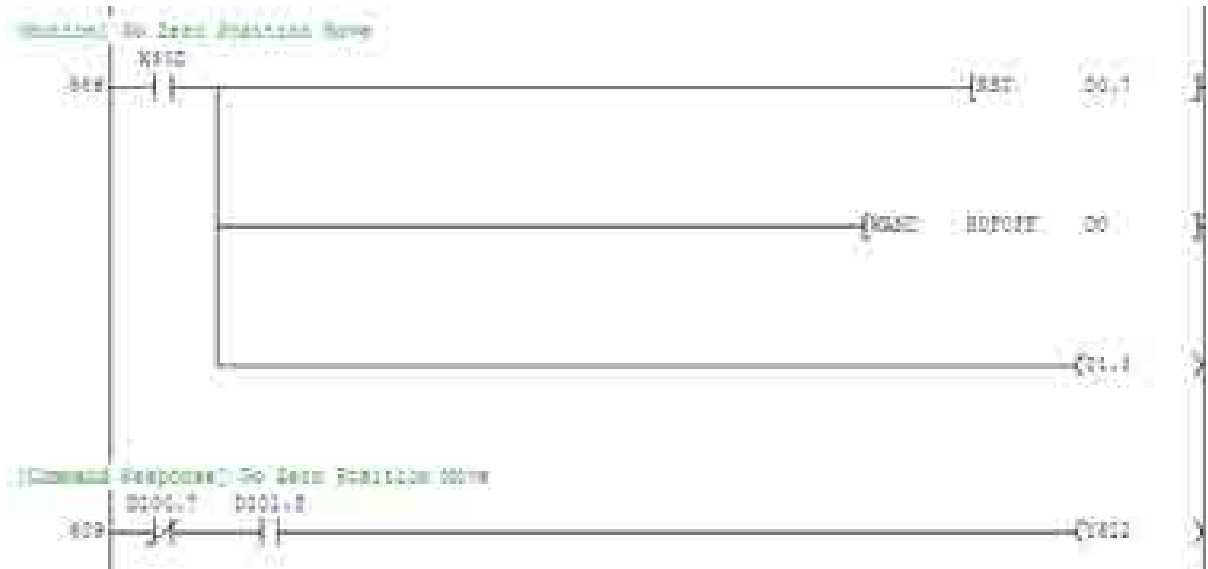
CMD_CODE = 0000b

RESPONSE_TYPE = 0001b

GO_ZERO_POS bit = 0

Output-Map							
0100	0	1	2	3	4	5	6
	7	8	9	10	11	12	13
0101	14	15	16	17	18	19	20
	21	22	23	24	25	26	27
0102	Response Data # 0						
0103							
0104							
0105							
0106							
0107							
0108							
0109							
0110							
0111							
0112							
0113							
0114							
0115							
0116							
0117							
0118							
0119							
0120							
0121							
0122							
0123							
0124							
0125							
0126							
0127							

✓ LADDER



NOTE : 예제 14의 래더는 예제 13에 추가된 명령으로, 앞의 예제가 생략되었습니다.

✓ 명령 순서

- ① X50D의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 클리어 하여 IO-Map을 모션모드 상태로 전환합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '0'을 입력합니다.
- ③ -GO_ZERO_POS 비트(D001.3)을 ON 하여 명령을 실행합니다.
- ④ 영점이동 명령에 대한 응답은 MOTION/SETTING_RESP. 비트(D100.7)의 NC 입력과, GO_ZERO_POS_Resp. 비트(D101.7)의 AND 회로로 구성하여 확인 가능합니다.

5.2.4 위치 이동 명령

상대위치 이동은 모션모드(MOTIONION/SETTING = 0)상태의 명령 코드(CMD_CODE) '1'에서 동작하며, 입력 데이터(D002)에 위치값으로 상대치 또는 절대치 이동을합니다.

■ 상대 위치 이동

상대 위치 이동은 입력된 위치 값으로 상대이동을 하는 명령입니다. 예제 15 은 Axis-0 의 상대위치 이동에 대한 예제 입니다.

예제 15. 상대위치 이동 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 INC/ABS – D001.8
 CMD_START – D000.4
 Command Data Area – D002~D003 (D002 [1 DWORD])

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 MOTIONING – D101.1
 CMD_RESP – D100.4

✓ 입출력 정보

- 입력 정보

INC Move – X50E
 Incremental Position (입력 범위 : -2,147,483,648 ~ -2,147,483,647 [DWORD])

- 출력 정보

MOTIONING – Y604
 POS MOV Resp – Y613 *)

*) 위치이동 명령의 응답 확인은 『예제 17. 위치이동 명령에 대한 응답 확인 방법』을 참조 하세요

✓ IO-Map 의 명령 및 응답 형태

- INC Move 명령 실행

Input-Map

0000	0	1	2	3	4	5	6	7
0001	0	1	2	3	4	5	6	7
0002	0	1	2	3	4	5	6	7
0003	Incremental Position Value							
0004	Incremental Position Value							

MOTION/SETTING bit = 0

CMD_CODE = 0001b

INC/ABS bit = 0

Command Data = 상대 위치 값

CMD_START bit = 1

Output-Map

0100	0	1	2	3	4	5	6	7
0101	0	1	2	3	4	5	6	7
0102								
0103								
0104								

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0001b

CMD_RESP bit = 1, MOTIONING bit = 1

- INC Move 명령 후 CMD_START 비트의 해제

Input-Map

0000	0	1	2	3	4	5	6	7
0001	0	1	2	3	4	5	6	7
0002	0	1	2	3	4	5	6	7
0003	Incremental Position Value							
0004	Incremental Position Value							

MOTION/SETTING bit = 0

CMD_CODE = 0001b

INC/ABS bit = 0

Command Data = 상대 위치 값

CMD_START bit = 0

Output-Map

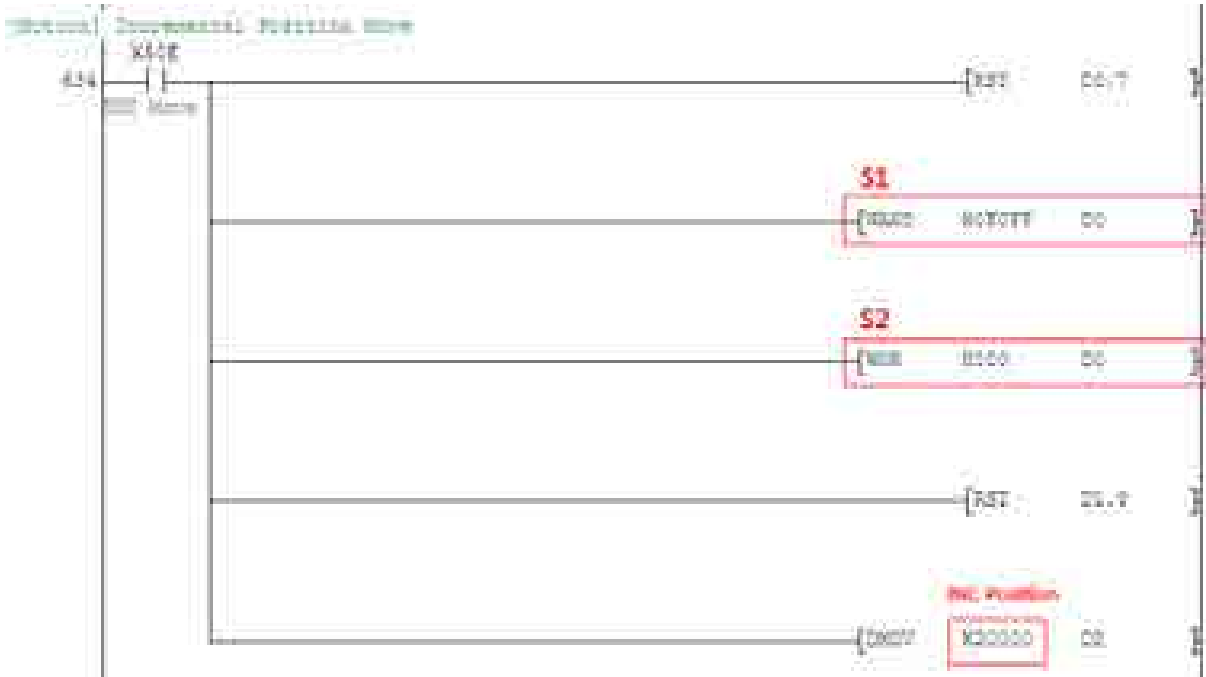
0100	0	1	2	3	4	5	6	7
0101	0	1	2	3	4	5	6	7
0102								
0103								
0104								

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0001b

CMD_RESP bit = 0, MOTIONING bit = 1

✓ LADDER



NOTE : 예제 15의 래더는 예제 14. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

✓ 명령 순서

- ① X50E 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② MOV 명령으로 'S'영역에 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '1'을 입력 합니다.(S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '1'로 세트합니다.)
- ③ INC/ABS 비트(D001.8)을 리셋 합니다.
- ④ 상대 위치 값을 Axis-0 의 데이터영역(D002)에 입력 합니다.
(예제 14.는 DMOV 명령으로 상대위치 값 20000pulse 를 D002 영역에 입력한 래더 입니다.)
- ⑤ 예제 9 를 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.

■ 절대 위치 이동

절대 위치 이동은 입력 데이터(D002)에 위치값으로 절대치 이동을합니다. 예제 16. 는 0 번축의 절대위치 이동에 대한 예제입니다.

예제 16. 절대위치 이동 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 INC/ABS – D001.8
 CMD_START – D000.4
 Command Data Area – D002~D003 (D002 [1 DWORD])

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – X1008~D100.B
 MOTIONING – D101.1
 CMD_RESP – D100.4

✓ 입출력 정보

- 입력 정보

ABS Move - X50F
 Absolute Position (입력 범위 : -2,147,483,648 ~ -2,147,483,647 [DWORD])

- 출력 정보

MOTIONING - Y604
 POS MOV Resp - Y613^{*)}

*) 위치이동 명령의 응답 확인은 『예제 17. 위치이동 명령에 대한 응답 확인 방법』을 참조 하세요

✓ IO-Map 의 명령 및 응답 형태

- ABS Move 명령 실행

Input-Map

0000	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0	0
0003	Absolute Position Value								
0004	Absolute Position Value								

MOTION/SETTING bit = 0

CMD_CODE = 0001b

INC/ABS bit = 1

Command Data = 절대 위치 값

CMD_START bit = 1

Output-Map

0100	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0	0
0102									
0103									
0104									

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0001b

CMD_RESP bit = 1, MOTIONING bit = 1

- ABS Move 명령 후 CMD_START 비트의 해제

Input-Map

0000	0	0	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0	0	0
0003	Absolute Position Value								
0004	Absolute Position Value								

MOTION/SETTING bit = 0

CMD_CODE = 0001b

INC/ABS bit = 1

Command Data = 절대 위치 값

CMD_START bit = 0

Output-Map

0100	0	0	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0	0	0
0102									
0103									
0104									

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0001b

CMD_RESP bit = 0, MOTIONING bit = 1

✓ LADDER



NOTE : 예제 16.의 래더는 예제 15. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

✓ 명령 순서

- ① X50F 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② MOV 명령으로 'S'영역에 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '1'을 입력 합니다.(S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '1'로 세트합니다.)
- ③ INC/ABS 비트(D001.8)을 클리어 합니다.
- ④ 절대 위치 값을 Axis-0 의 데이터영역(D002)에 입력 합니다.
(예제 16. 에서는 DMOV 명령으로 절대위치 값을 20000pulse 을 D002 영역에 입력한 래더 입니다.)
- ⑤ 예제 9 를 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.

※ 위치 이동 명령에 대한 응답확인

위치 이동 명령의 응답 비트는 조그운전 또는 스텝이동, 영점 이동과 같이 IO-Map 으로 확인할 수 없습니다. 따라서, 예제 17 과 같이 IO-Map 의 조합으로 확인 할 수 있습니다.

예제 17. 위치이동 명령에 대한 응답 확인 방법

Output-Map

	7	6	5	4	3	2	1	0
D100	7	6	5	4	3	2	1	0
D101	7	6	5	4	3	2	1	0
D102								
D103								

✓ LADDER



NOTE : 예제 17.의 래더는 예제 16. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

- ① 위치 이동 명령의 응답은 IO-Map 의 조합으로 확인할 수 있습니다.
- ▶ 위치이동 명령의 응답은 MOTION/SETTING_RESP 비트가(D100.7)의 N.C 입력과, CMD_CODE_RESP 영역(D100.8~D100.B)이 '1'일 때의 상태 값, CMD_START_RESP (D100.4)를 AND 회로로 구성된 결과 값입니다.

5.2.5 PT 운전 명령

PT 운전은 모션모드(MOTIONION/SETTING = 0)상태의 명령 코드(CMD_CODE) '4'에서 동작하며, 입력 데이터(D002)에 PT 항목의 번호를 입력하여 운전을 실행 합니다.

■ 일반 PT 운전

일반 PT 운전은 입력된 값(D002)부터 PT 운전을 시작합니다. 예제 18 은 Axis-0 의 PT 운전 명령에 대한 예제입니다.

예제 18. PT 운전 명령

✓ 제어 비트맵 정보

- **Input-Map**
 - MOTION/SETTING – D000.7
 - CMD_CODE – D000.8~D000.B
 - SINGLE_PT – D001.C
 - CMD_START –D000.4
 - Command Data Area – D002~D003 (D002 [1 DWORD])
- **Output-Map**
 - MOTION/SETTING_RESP – D100.7
 - CMD_CODE_RESP – D100.8~D100.B
 - MOTIONING – D101.1
 - CMD_ RESP – D100.4
 - PT_RUNUNG – D101.8

✓ 입출력 정보

- **입력 정보**
 - PT Run – Y510
 - PT No. (PT 항목이 저장 되어있는 번호 [DWORD Type])
- **출력 정보**
 - MOTIONING – Y604
 - PT CMD Resp – Y614^{*)}
 - PT Running – Y606

*) PT 운전 명령의 응답 확인은 『예제 20. PT 운전 명령에 대한 응답 확인 방법』을 참조 하세요

✓ IO-Map 의 명령 및 응답 형태

- PT RUN 명령 실행

Input-Map

0000	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0
0003	Position Table Number						
0004	Position Table Number						

MOTION/SETTING bit = 0

CMD_CODE = 0100b

SINGLE_PT bit = 0

Command Data = PT 번호

CMD_START bit = 1

Output-Map

0100	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0
0102							
0103							
0104							

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0100b

CMD_RESP bit = 1, PT_RUNNING bit = 1

- PT RUN 명령 후 CMD_START 비트의 해제

Input-Map

0000	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0
0003	Position Table Number						
0004	Position Table Number						

MOTION/SETTING bit = 0

CMD_CODE = 0100b

SINGLE_PT bit = 0

Command Data = PT 번호

CMD_START bit = 0

Output-Map

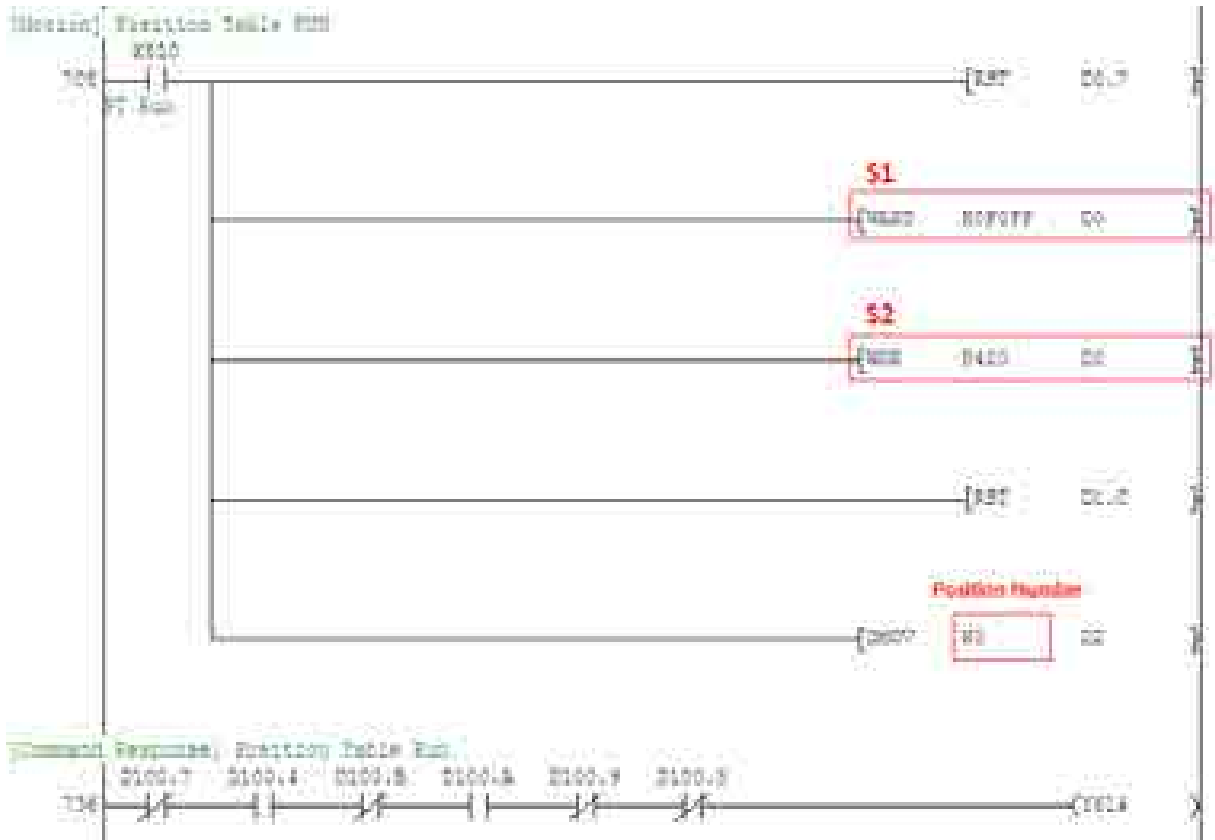
0100	0	0	0	0	0	0	0
0101	0	0	0	0	0	0	0
0102							
0103							
0104							

MOTION/SETTING_RESP bit = 0

CMD_CODE_RESP = 0100b

CMD_RESP bit = 0, PT_RUNNING bit = 1

✓ LADDER



NOTE 1: 예제 18.의 래더는 예제 17. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

✓ 명령 순서

- ① X511 의 Close 입력으로 MOTION/SETTING 비트(Y0007)을 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② MOV 명령으로 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '4'을 입력 합니다 (S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '4'로 세트).
- ③ SINGLE_PT 비트(D001.C)을 클리어 합니다.
- ④ 시작할 PT 번호를 Axis-0 의 데이터 영역(D002)에 입력 합니다.
(예제 18. 에서는 DMOV 명령의 'S'에 PT 번호 '0'을 D0000 에 지정한 래더 입니다.)
- ⑤ 예제 9 를 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.

■ 싱글 PT 운전

싱글 PT 운전은 입력된 값(D0000)부터 하나의 PT 항목에 대하여 운전합니다. 예제 19 는 0 번축의 싱글 PT 운전에 대한 예제입니다.

예제 19. 싱글 PT 운전 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 SINGLE_PT – D001.C
 CMD_START – D000.4
 Command Data Area – D002~D003 (D002 [1 DWORD])

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 MOTIONING – D101.1
 CMD_RESP – D100.4
 PT_RUNUNG – D101.8

✓ 입출력 정보

- 입력 정보

Single PT Run
 PT No. (PT 항목이 저장 되어있는 번호 [DWORD])

- 출력 정보

MOTIONING – Y604
 PT CMD Resp – Y614*)
 PT Running – Y606

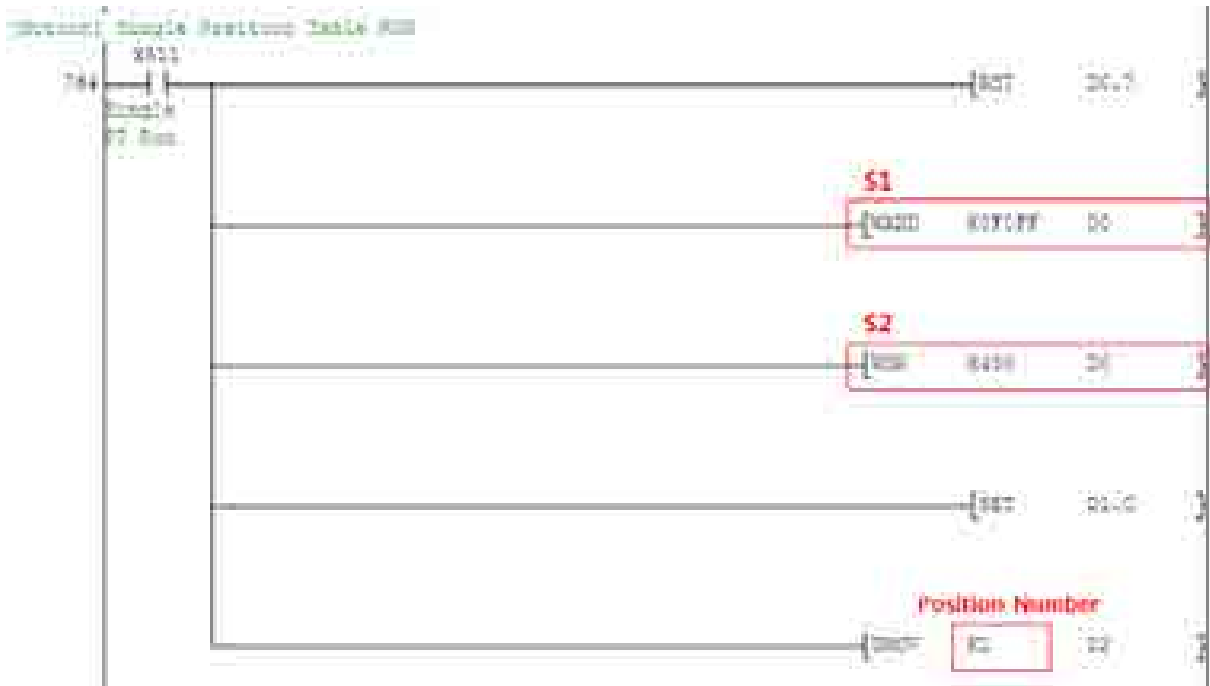
*) PT 운전 명령의 응답 확인은 『예제 20. PT 운전 명령에 대한 응답 확인 방법』을 참조 하세요

✓ IO-Map 의 명령 및 응답 형태

- 싱글 PT RUN 명령 실행

Input-Map							
0000	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
0001	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
0002	0	0	0	0	0	0	0
	0	0	0	0	0	0	0
0003	Position Table Number						
0003							

✓ LADDER



NOTE 1: 예제 19.의 래더는 예제 15. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

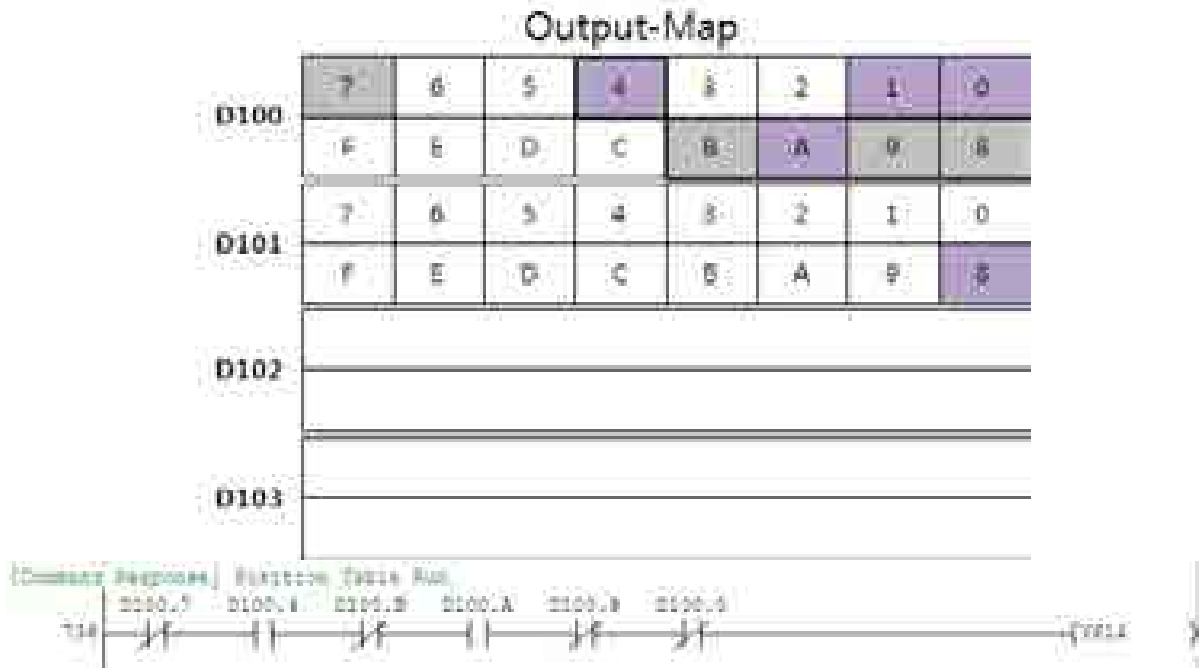
✓ 명령 순서

- ① X50F 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② MOV 명령으로 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '4'을 입력 합니다 (S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '4'로 세트).
- ③ SINGLE_PT 비트(D001.C)을 세트 합니다.
- ④ 운전할 PT 번호를 Axis-0 의 데이터 영역(D002)에 입력합니다.
(예제 19.는 DMOV 명령으로 실행할 PT 번호 '1'을 D002 영역에 입력한 래더입니다.)
- ⑤ 예제 9 를 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.

■ PT 운전 명령에 대한 응답확인

PT 운전의 명령 응답 비트는 조그운전 또는 스텝이동, 영점 이동과 같이 IO-Map 으로 확인할 수 있습니다. 따라서, 예제 20 과 같이 IO-Map 의 조합으로 확인 할 수 있습니다.

예제 20. PT운전 명령에 대한 응답 확인 방법



NOTE : 예제 20.의 래더는 예제 19. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

- ① PT 운전 명령의 응답은 IO-Map 의 조합으로 확인할 수 있습니다.
 - ▶ PT 운전 명령의 응답은 MOTION/SETTING_RESP 비트가(D100.7)의 N.C 입력과, CMD_CODE_RESP 영역(D100.8~D100.B='4')의 정보, CMD_START_RESP(X1004)의 응답 정보를 AND 회로로 구성된 결과 값입니다.

5.2.6 원점 이동 명령

원점 이동 명령은 모션모드(MOTIONION/SETTING = 0)상태의 명령 코드(CMD_CODE) '7'에서 동작하며, 데이터영역(D002)에 입력된 값과 무관하게 동작합니다. 예제 21 은 Axis-0 의 원점이동 명령 대한 예제입니다.

예제 21. 원점 이동 명령

✓ 제어 비트맵 정보

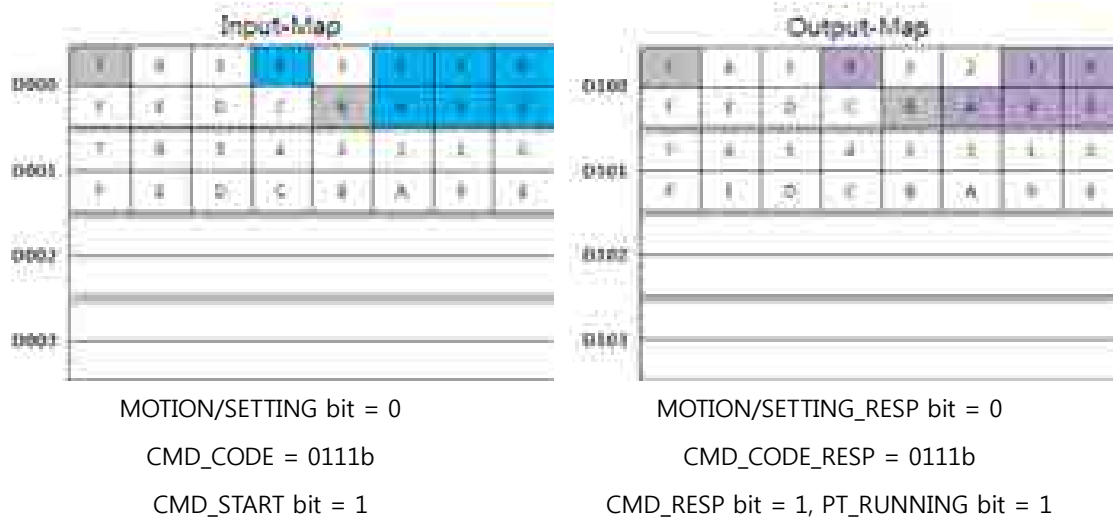
- **Input-Map**
 MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 CMD_START – D000.4
- **Output-Map**
 MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 MOTIONING – D101.1
 CMD_RESP – D100.4

✓ 입출력 정보

- **입력 정보**
 ORIGIN Search – X512
- **출력 정보**
 MOTIONING – Y604
 Origin Mov Resp – Y60B

✓ IO-Map 의 명령 및 응답 형태

- 원점 이동 실행



✓ LADDER



NOTE 1: 예제 21.의 래더는 예제 20. 에 추가된 명령으로, 앞의 예제가 생략 되었습니다.

✓ 명령 순서

- ① X50E 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 클리어 하여 IO-Map 을 모션모드 상태로 전환 합니다.
- ② MOV 명령으로 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '7'을 입력 합니다 (S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '7'로 세트).
- ③ 예제 9 를 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ④ 이 명령의 응답은 원점 이동 명령의 IO-Map 을 조합하여 확인 가능합니다.
 - ▶ 위치이동 명령의 응답은 MOTION/SETTING_RESP 비트가(D100.07)의 N.C 입력과, CMD_CODE_RESP 영역(D100.8~D100.B='7')의 정보, CMD_START_RESP(X1004)의 응답 정보를 AND 회로로 구성된 결과 값입니다.

www.fastech.co.kr

5.2.7 파라미터 설정

파라미터 설정은 해당 축의 모터 드라이브에대한 파라미터와 PT 정보의 확인 및 설정, 저장을 할 수 있습니다.

■ 파라미터 확인 명령

파라미터 확인은 설정모드(MOTIONION/SETTING=1)상태의 명령 코드(CMD_CODE) '8' 에서 동작하며, 인덱스 영역 D001 에 해당하는 파라미터 또는 PT 정보 등의 값을 요청하는 명령입니다. 이때 데이터영역(D002)에 입력된 값과 무관하게 동작합니다. 예제 22. 은 Axis-0 의 파라미터 확인 명령에 대한 예제입니다.

예제 22. 파라미터 확인 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 INDEX_VALUE– D001 (1WORD)
 CMD_START – D000.4

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 INDEX_VALUE_RESP – D101 (1WORD)
 RESPONSE
 CMD_RESP – D100.4

✓ 입출력 정보

- 입력 정보

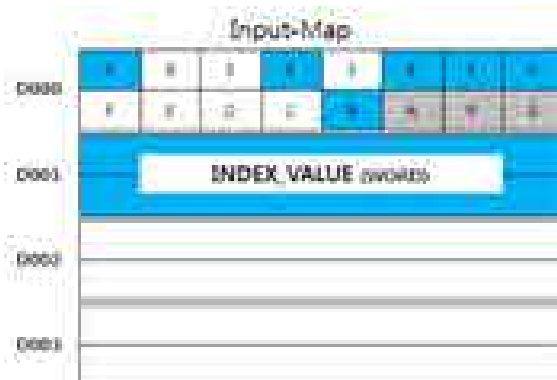
Read Parameter – X513

- 출력 정보

Read Parameter Response Value

✓ IO-Map 의 명령 및 응답 형태

- 원점 이동 실행

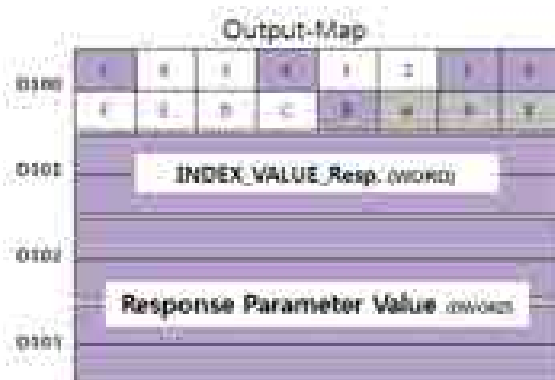


MOTION/SETTING bit = 1

CMD_CODE = 1000b

INDEX_VALUE = Index Value (Parameter No)

CMD_START bit = 1



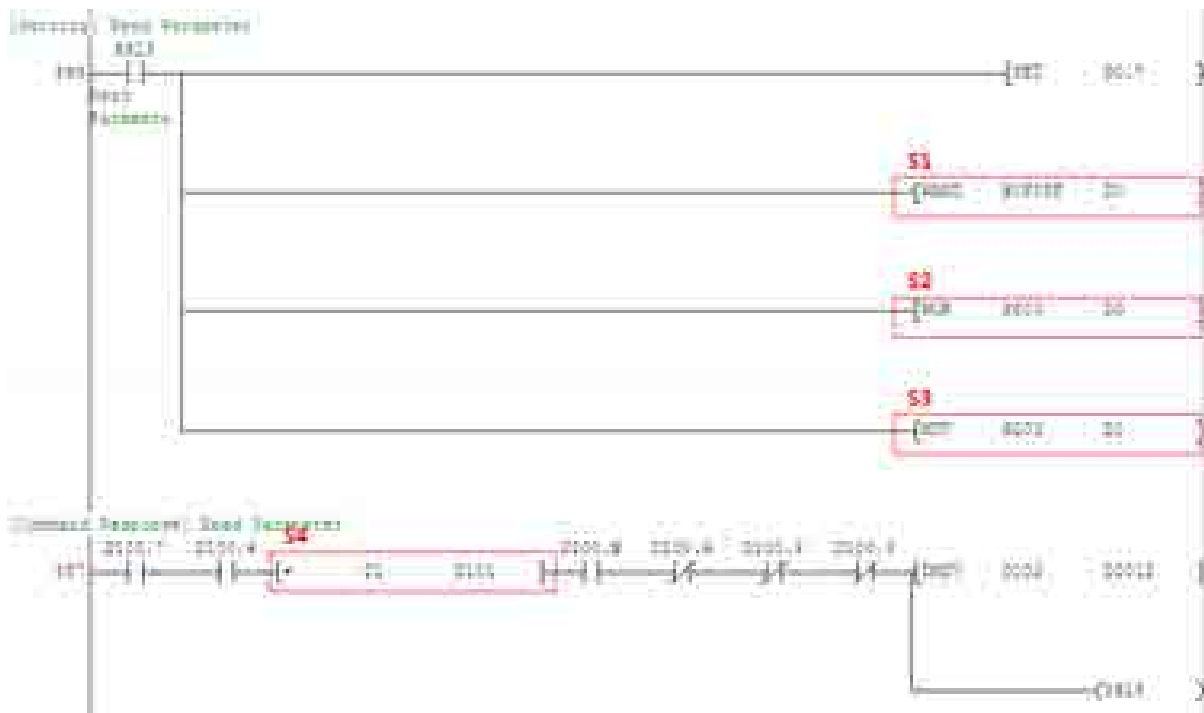
MOTION/SETTING_RESP bit = 1

CMD_CODE_RESP = 1000b

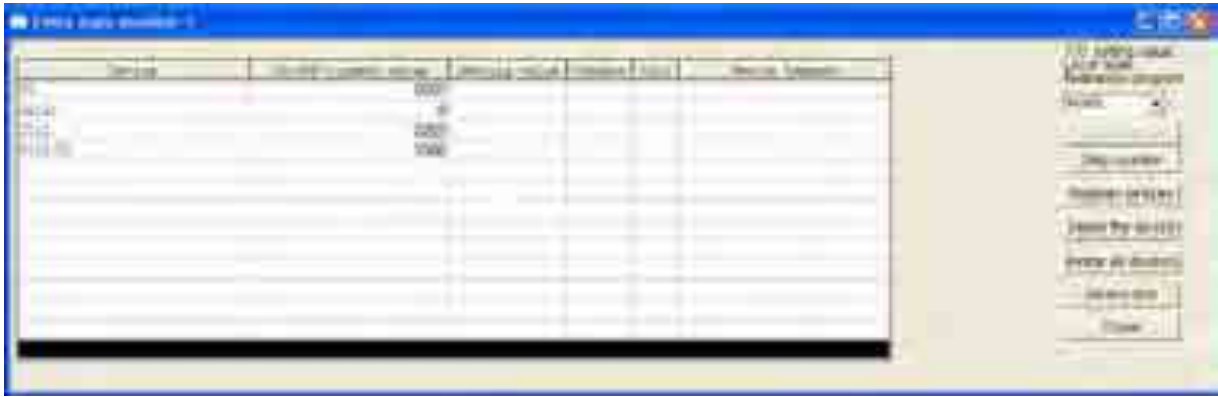
INDEX_VALUE_RESP = Index Value (Parameter No)

CMD_RESP bit = 1

✓ LADDER



NOTE 1: 예제 22.의 래더는 예제 21. 에서 추가된 명령으로, 앞의 예제가 생략 되었습니다.



NOTE 2: 예제 22 의 실행으로 D100 으로 요청한 파라미터 값을 얻을 수 있습니다.

NOTE 3: 예제 22 에서 인덱스 영역에 입력한 파라미터 번호는 INDEX_RESP. 영역(X101) 에 16 비트 데이터로 루프-백 됩니다.

✓ 명령 순서

- ① X513 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 세트 하여 IO-Map 을 설정 모드 상태로 전환합니다.
- ② MOV 명령으로 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '8'을 입력 합니다 (S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '8'로 세트).
- ③ INDEX 영역(D001)에 파라미터 번호를 입력합니다.
(S3 에서 MOV 명령으로 파라미터 PN#0203 『Step Distance 3』를 입력)
- ④ 예제 10 을 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ⑤ 파라미터 확인 명령의 응답은 IO-Map 의 조합으로 확인할 수 있습니다.
 - ▶ 파라미터 확인 명령의 응답은 MOTION/SETTING_RESP 비트가(D100.7)의 N.O 입력과, CMD_CODE_RESP 영역(D100.8~D100.B)을 '8'로 비교한 상태값, S4 의 비교 수식으로 INDEX_Resp.영역(D101)의 값과 파라미터 번호(D001)를 비교한 결과값, 그리고 CMD_START_RESP(D100.4)를 AND 회로로 구성된 결과값이 됩니다.
(본 예제에서는 요청한 파라미터의 값을 D2012 에 저장하도록 한 예제입니다.)

■ 파라미터 변경 명령

파라미터 변경은 설정모드(MOTIONION/SETTING=1)상태의 명령 코드(CMD_CODE) '9' 에서 동작하며, 인덱스 영역(K4Y0010)에 작성된 파라미터 번호에 입력 데이터영역(D0000)에 지정된 값으로 변경하는 명령입니다. 예제 23 은 0 번축의 파라미터 변경 명령에 대한 예제입니다.

예제 23. 파라미터 변경 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 INDEX_VALUE – D001 [1 WORD]
 COMMAND_WORD_DATA – D002 [2 DWORD]
 CMD_START – D000.4

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 INDEX_VALUE_RESP – D101 [1WORD]
 CMD_RESP – X1004
 RESPONSE_DATA – D102~D103 (D102 [1 DWORD])

✓ 입출력 정보

- 입력 정보

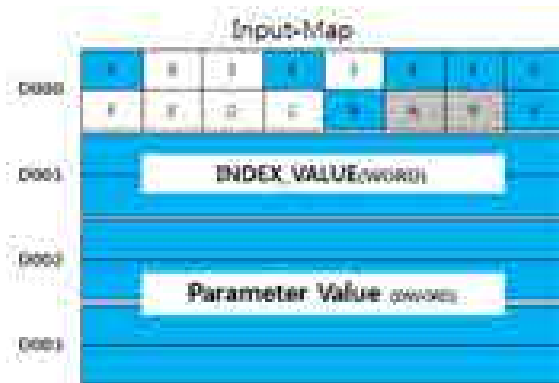
Write Parameter – X514
 Write Parameter Value

- 출력 정보

Read Parameter Value

✓ IO-Map 의 명령 및 응답 형태

- 파라미터 값 변경 실행



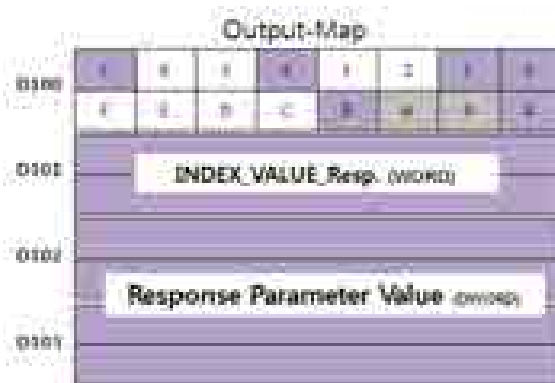
MOTION/SETTING bit = 1

CMD_CODE = 1001b

INDEX_VALUE = Index Value (Parameter No)

COMMAND_WORD_DATA = Write Parameter Value

CMD_START bit = 1



MOTION/SETTING_RESP bit = 1

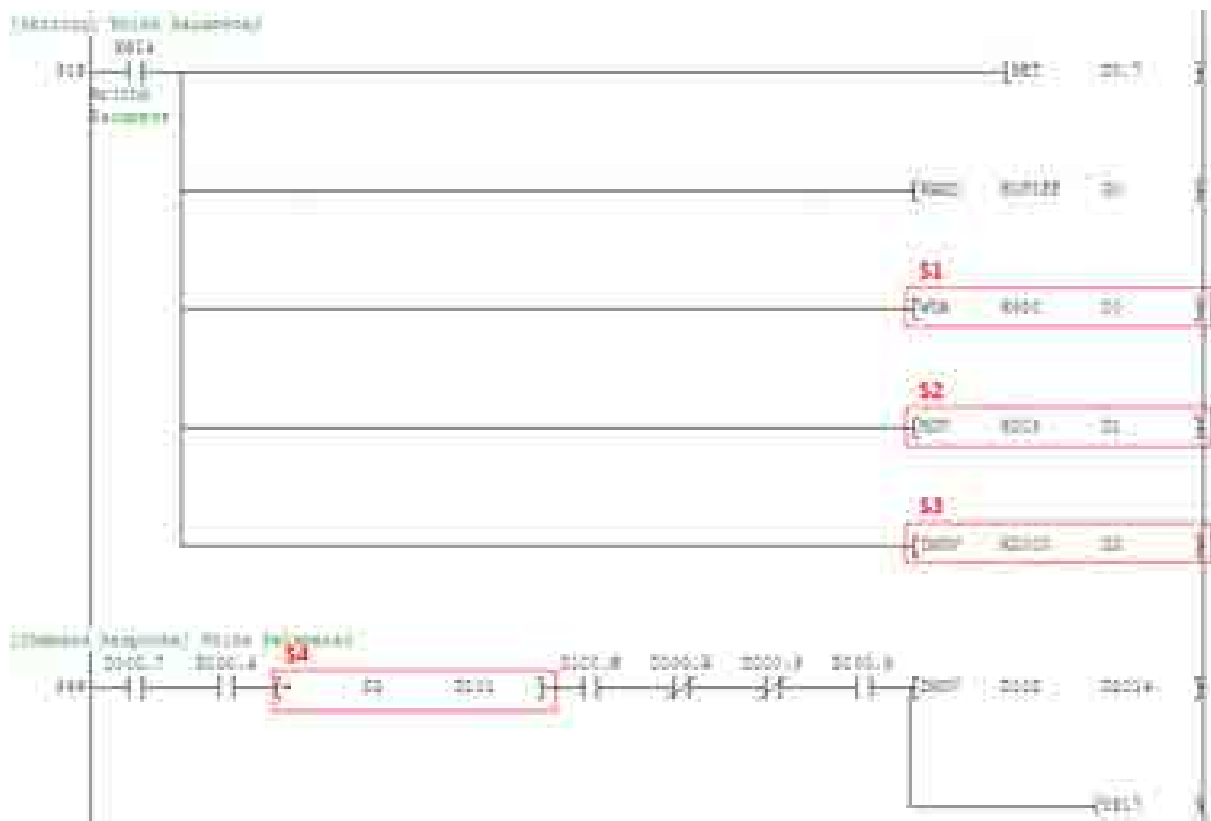
CMD_CODE_RESP = 1001b

INDEX_VALUE_RESP = Index Value (Parameter No)

RESPONSE_DATA = Response Parameter Value

CMD_RESP bit = 1

✓ LADDER



NOTE : 예제 23의 래더는 예제 22. 에서 추가된 명령으로, 앞의 예제가 생략 되었습니다.

■ 파라미터 저장

파라미터 저장은 설정모드(MOTIONION/SETTING=1)상태의 명령 코드(CMD_CODE) '14' 에서 동작하여 ROM 영역에 파라미터 값을 저장 합니다. 예제 20. 은 0 번축의 파라미터 저장 명령에 대한 예제입니다.

예제 24. 파라미터 저장 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 INDEX_VALUE – D001 [1 WORD]
 CMD_START – D000.4

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 INDEX_VALUE_RESP – D101 (1WORD)
 CMD_RESP – D100.4

✓ 입출력 정보

- 입력 정보

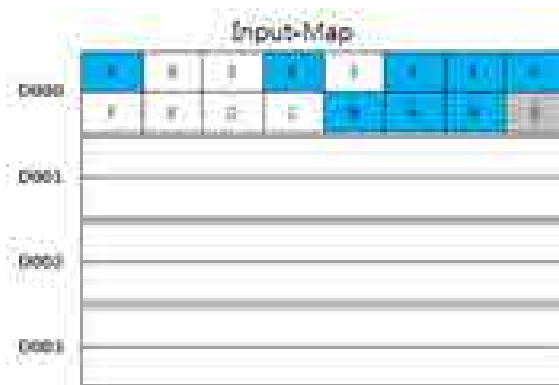
Save Parameter – X515

- 출력 정보

Save Parameter Resp – Y616

✓ IO-Map의 명령 및 응답 형태

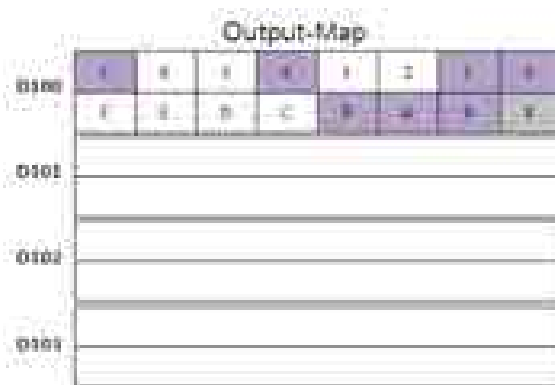
- 파라미터 저장 명령



MOTION/SETTING bit = 1

CMD_CODE = 1110b

CMD_START bit = 1

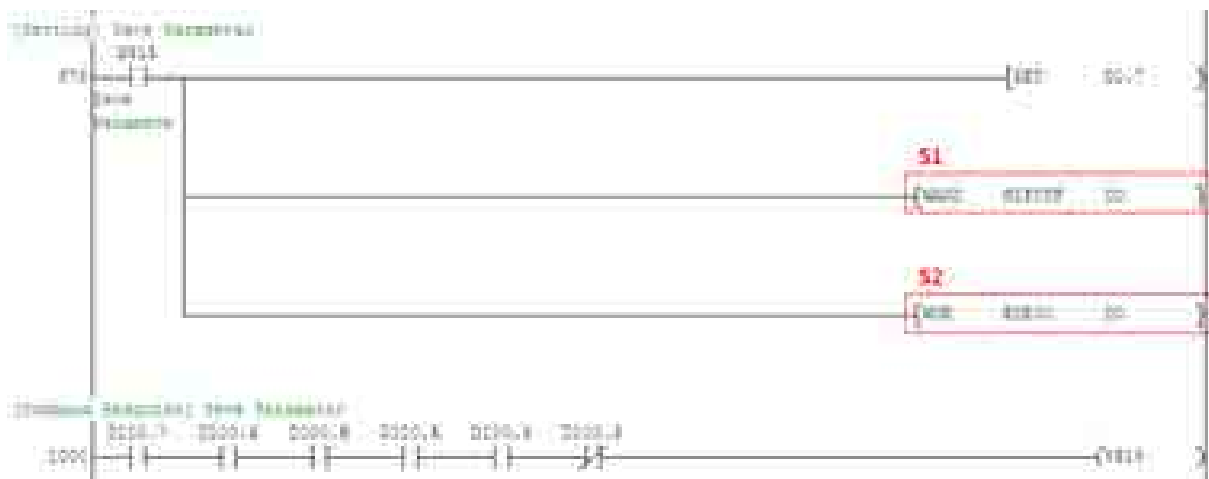


MOTION/SETTING_RESP bit = 1

CMD_CODE_RESP = 1110b

CMD_RESP bit = 1

✓ **LADDER**



NOTE 1: 예제 24.의 리더는 예제 23. 에서 추가된 명령으로, 앞의 예제가 생략 되었습니다.

✓ 명령 순서

- ① X515 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 세트 하여 IO-Map 을 설정 모드 상태로 전환합니다.
- ② MOV 명령으로 CMD_CODE 영역(D000.8~D000.B)에 명령코드 '14'을 입력 합니다 (S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '14'로 세트).
- ③ 예제 10 을 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ④ 이 명령의 응답은 IO-Map 의 조합으로 확인할 수 있습니다.
 - ▶ 파라미터 저장 명령의 응답은 MOTION/SETTING_RESP 비트(D100.7)의 N.O 입력, CMD_CODE_RESP 영역(D100.8~D100.B)을 '14'가 되는 상태값과 CMD_START_RESP (D100.4)를 AND 회로로 구성된 결과값이 됩니다.)

5.2.8 위치 지정

위치 지정 명령은 설정모드(MOTIONION/SETTING=1)상태의 명령 코드(CMD_CODE) '10' 에서 동작하여 추종중인 위치값(Command Position)을 변경합니다. 예제 25. 은 Axis-0 의 추종중인 위치값 변경 명령에 대한 예제입니다.

예제 25. 위치값 변경 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 COMMAND_WORD_DATA – D002~D003 [DWORD]
 CMD_START – D000.4

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 CMD_RESP – D100.4
 RESPONSE_DATA – D102~D103 (D102 [DWORD])

✓ 입출력 정보

- 입력 정보

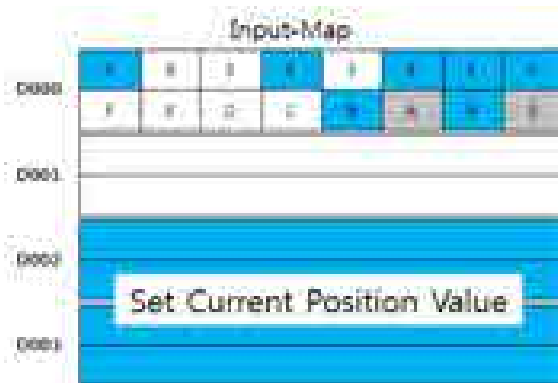
Set Position – X516
 Set Current Position Data

- 출력 정보

Response Current Position

✓ IO-Map 의 명령 및 응답 형태

- 현재 위치 지정 명령

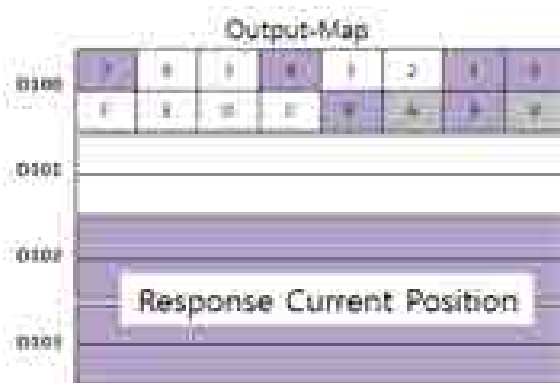


MOTION/SETTING bit = 1

CMD_CODE = 1010b

COMMAND_WORD_DATA = Set Current Position

CMD_START bit = 1



MOTION/SETTING_RESP bit = 1

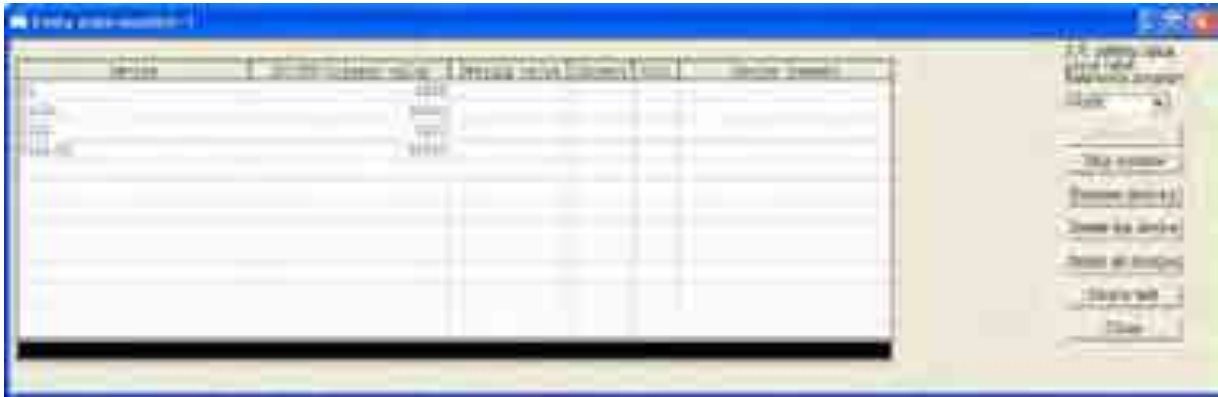
CMD_CODE_RESP = 1010b

RESPONSE_DATA = Response Current Position

CMD_RESP bit = 1

✓ LADDER

**NOTE 1:** 예제 25 의 래더는 예제 24. 에서 추가된 명령으로, 앞의 예제가 생략 되었습니다.



NOTE 2: 예제 25 을 실행하여, D102 영역으로 변경된 추종 위치값을 얻습니다.



NOTE 4: 예제 25 을 실행 후, 예제 7 을 실행하여 현재 위치 추종값을 모니터링 한 화면입니다.

✓ 명령 순서

- ① X516 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 세트 하여 IO-Map 을 설정 모드 상태로 전환합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '10'을 입력 합니다.
(S1 에서 CMD_CODE 를 '0'으로 초기화 하고, S2 에서 '10'로 세트).
- ③ 예제 10 을 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ④ 위치 지정 명령 응답은 IO-Map 의 조합으로 확인할 수 있습니다.
 - ▶ 파라미터 저장 명령의 응답은 MOTION/SETTING_RESP 비트가(D100.7)의 N.O 입력, CMD_CODE_RESP 영역(D100.8~D100.B)을 '10'로 비교한 상태값, 그리고 CMD_RESP(D100.4)를 AND 회로로 구성된 결과 값이 됩니다.)
- ⑤ 이 명령을 실행 후 현재 추종위치 값을 요청 하면, 이 값이 변동하였음을 확인 할 수 있습니다.

5.2.9 알람 내역 요청 및 초기화

모션게이트는 최근에 발생한 알람 부터 4 개에 대한 알람내역을 관리 할 수 있습니다.

■ 알람 내역 확인

알람 내역 요청은 설정모드(MOTIONION/SETTING=1)상태의 명령 코드(CMD_CODE) '12' 에서 동작하여 해당축에 발생한 알람 내역(Alarm History)를 확인 할 수 있습니다. 예제 26. 은 Axis-0 의 알람 내역 요청에 대한 예제입니다.

예제 26. 알람 내역 요청

✓ 제어 비트맵 정보

- **Input-Map**
 MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 CMD_START – D000.4
- **Output-Map**
 MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 CMD_RESP – D100.4
 RESPONSE_DATA – D102~D103 (D102 [DWORD])

✓ 입출력 정보

- **입력 정보**
 ALM History – X517
- **출력 정보**
 ALM His Resp – Y61A

✓ IO-Map 의 명령 및 응답 형태

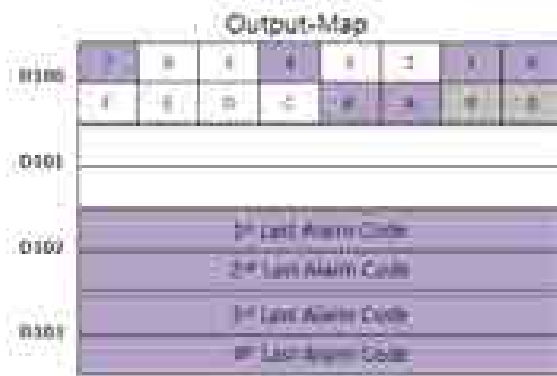
- 알람 내역 확인 명령 실행



MOTION/SETTING bit = 1

CMD_CODE = 1100b

CMD_START bit = 1



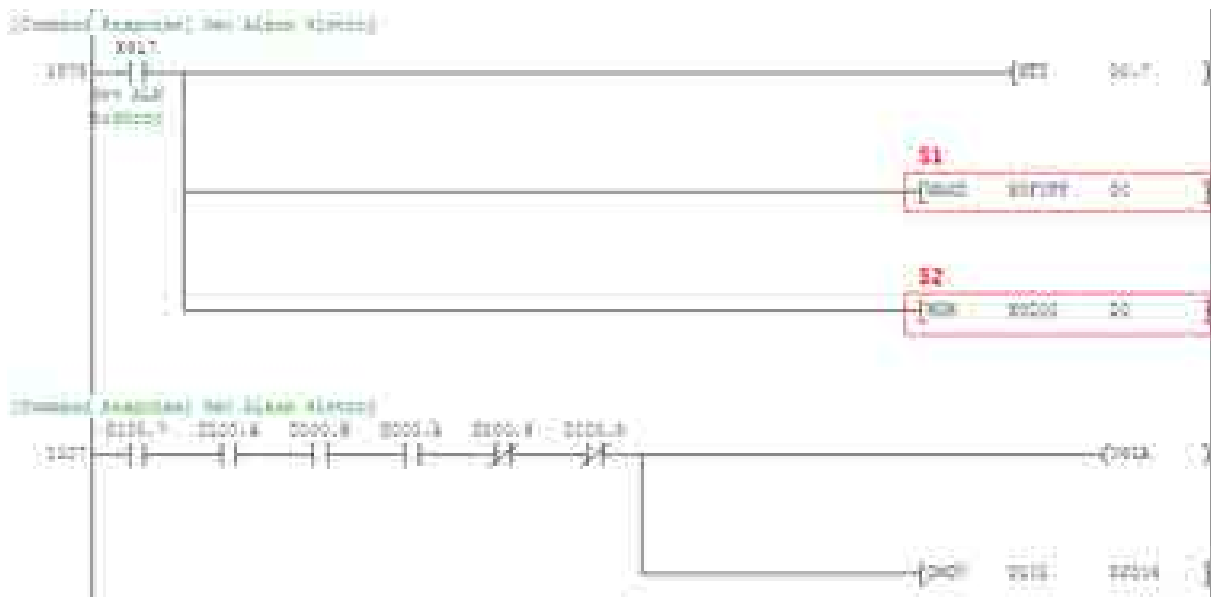
MOTION/SETTING_RESP bit = 1

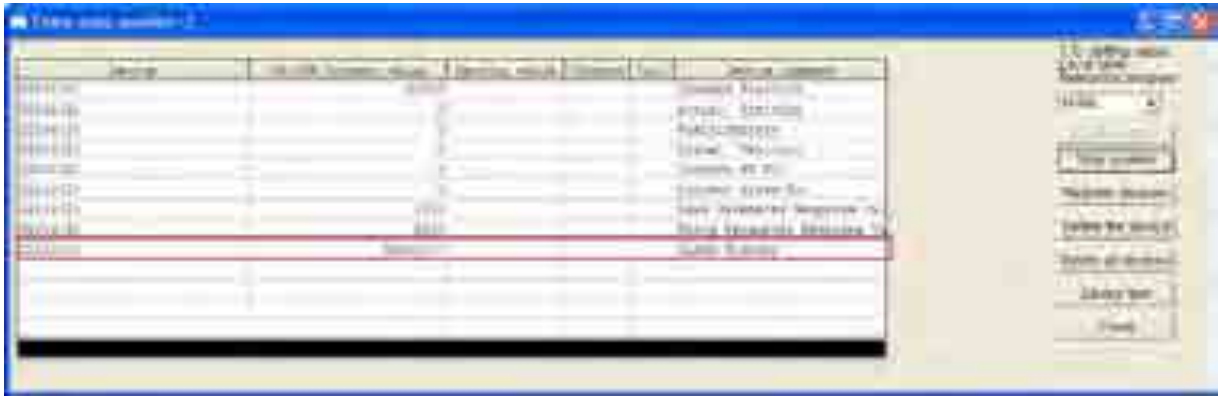
CMD_CODE_RESP = 1100b

CMD_RESP bit = 1

RESPONSE_DATA = Alarm History

✓ LADDER

**NOTE 1:** 예제 26의 래더는 예제 25에서 추가된 명령으로, 앞의 예제가 생략되었습니다.



NOTE 2: 인위적으로 발생시킨 알람을 알람내역 요청 명령을 실행하여 수신한 정보입니다.

NOTE 3: 알람 내역은 알람 리셋 명령을 기점으로 저장됩니다.

✓ 명령 순서

- ① X517 의 Close 입력으로 MOTION/SETTING 비트(Y0007)을 세트 하여 IO-Map 을 설정 모드 상태로 전환합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '12'를 입력합니다.
- ③ 예제 10 을 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ④ 알람 내역 확인 명령의 IO-Map 의 조합으로 확인할 수 있습니다.
 - ▶ 파라미터 저장 명령의 응답은 MOTION/SETTING_RESP 비트가(X1007)의 N.O 입력, CMD_CODE_RESP 영역(D100.8~D100.B)을 '12'로 비교한 상태값, 그리고 CMD_RESP(D100.4)를 AND 회로로 구성된 결과 값이 됩니다.)
- ⑤ 본 예제는 알람 내역요청 명령이 정상적으로 실행 되었을 때(명령 응답 확인) D2016 으로 저장되도록 한 예제입니다.
- ⑥ 이 명령을 실행 후 얻은 데이터는 16 진수 형태로 4 개의 데이터가 있습니다.
 - ▶ D2016.0~D2016.7 구간의 데이터(0x07 : 모터 접속 에러)은 가장 최근에 발생한 알람 정보입니다.
 - ▶ D2016.8~D2016.F 구간의 데이터(0x07 : 모터 접속 에러)은 최근에 발생한 알람 정보의 두번째 이전 정보입니다.
 - ▶ D2016.0~D2016.7 구간의 데이터(0x04 : 로드 과부하)은 최근에 발생한 알람 정보의 세번째 이전 정보입니다.
 - ▶ D2016.8~D2016.F 구간의 데이터(0x0A : 엔코더 접속 에러)은 최근에 발생한 알람 정보의 네번째 이전 정보입니다.

■ 알람 내역 초기화

알람 내역 초기화는 설정모드(MOTIONION/SETTING=1)상태의 명령 코드(CMD_CODE) '13' 에서 동작하여 해당축에 발생한 알람 내역(Alarm History)를 확인 할 수 있습니다. 예제 27 은 Axis-0 의 알람 내역 초기화에 대한 예제입니다.

예제 27. 알람 내역 초기화 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 CMD_START – D000.4

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 CMD_RESP – D100.4

✓ 입출력 정보

- 입력 정보

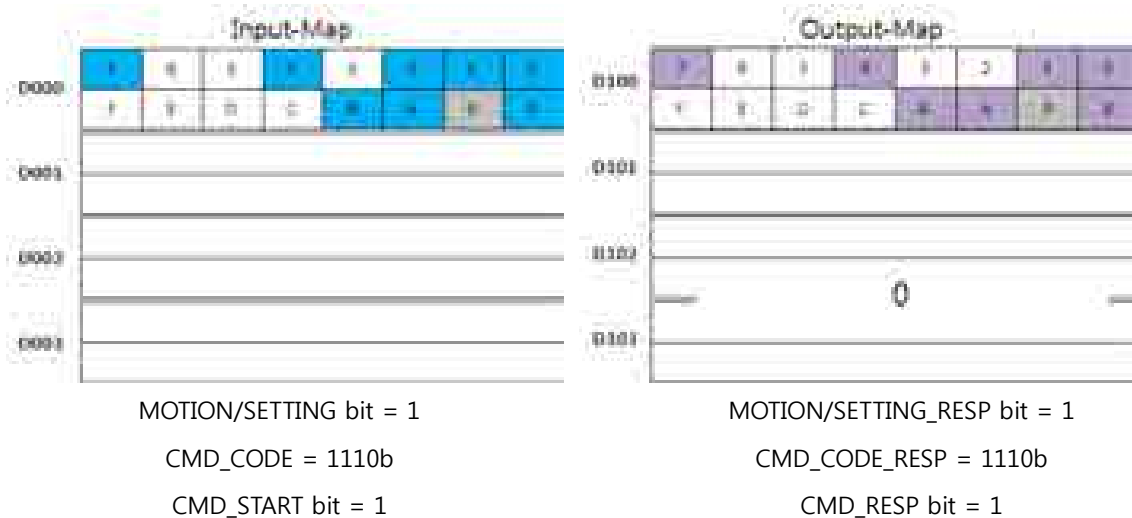
Reset ALM His – X518

- 출력 정보

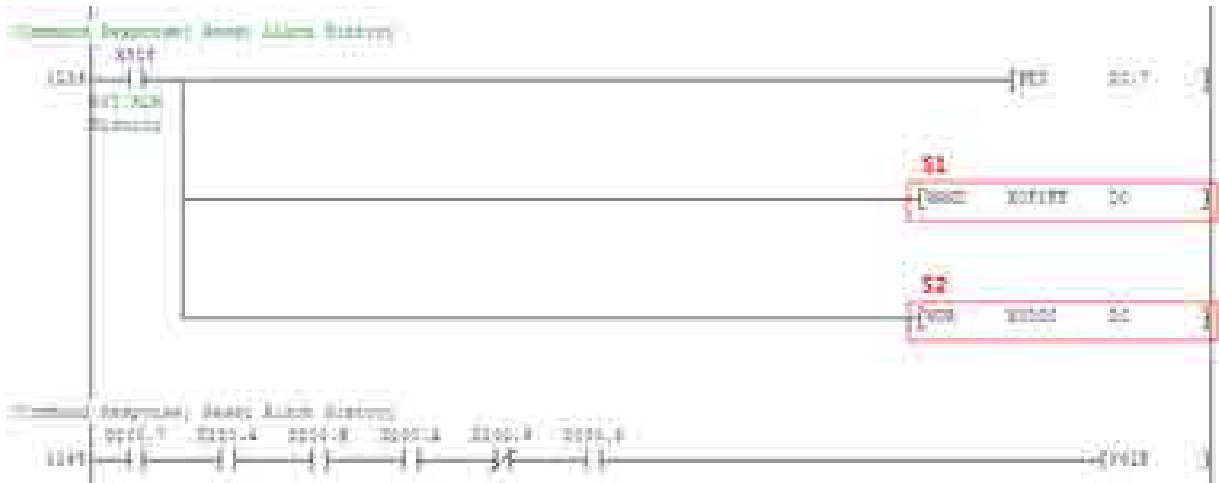
ALM His CLR Resp – Y61B

✓ IO-Map 의 명령 및 응답 형태

- 알람 내역 초기화 명령 실행



✓ LADDER



NOTE: 예제 27의 래더는 예제 26에서 추가된 명령으로, 앞의 예제가 생략되었습니다.

✓ 명령 순서

- ① X518의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 세트 하여 IO-Map을 설정 모드 상태로 전환합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '13'를 입력합니다.
- ③ 예제 10을 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ④ 이 명령을 실행 후 알람내역은 0x00, 0x00, 0x00, 0x00으로 클리어 됩니다.
- ⑤ 알람 내역 초기화 명령의 IO-Map의 조합으로 확인할 수 있습니다.
 - ▶ 파라미터 저장 명령의 응답은 MOTION/SETTING_RESP 비트가(X1007)의 N.O 입력, CMD_CODE_RESP 영역(D100.8~D100.B)을 '13'로 비교한 상태값, 그리고 CMD_RESP(D100.4)를 AND 회로로 구성된 결과 값이 됩니다.

5.2.10 모션게이트 버전 확인

모션게이트의 버전 확인 명령은 어느 축에서 명령을 실행하여도 모션게이트의 버전 정보를 얻을 수 있습니다. 이 명령으로 설정모드(MOTIONION/SETTING=1)상태의 명령 코드 (CMD_CODE) '5' 에서 동작하여 확인할 수 있습니다. 예제 28. 은 모션게이트의 버전 정보 요청 방법에 대한 예제입니다.

예제 28. 모션게이트의 버전 정보 요청 명령

✓ 제어 비트맵 정보

- Input-Map

MOTION/SETTING – D000.7
 CMD_CODE – D000.8~D000.B
 CMD_START – D000.4

- Output-Map

MOTION/SETTING_RESP – D100.7
 CMD_CODE_RESP – D100.8~D100.B
 CMD_RESP – D100.4

✓ 입출력 정보

- 입력 정보

Read Ver Info – X519

- 출력 정보

Read Ver Resp – Y61C

✓ IO-Map 의 명령 및 응답 형태

- 버전 정보 요청 명령



MOTION/SETTING bit = 1

CMD_CODE = 0101b

CMD_START bit = 1



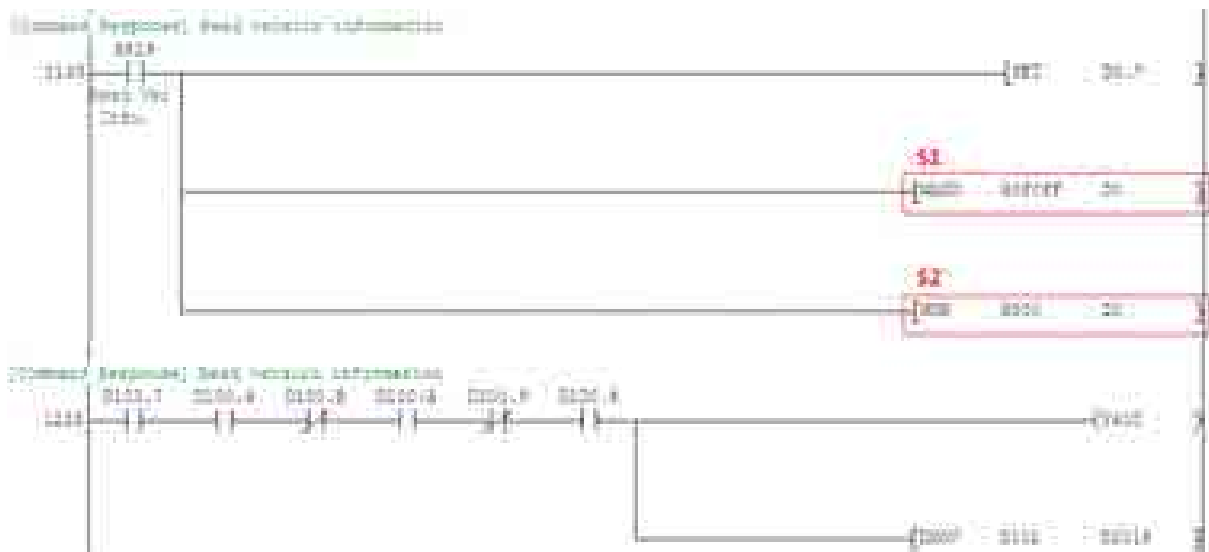
MOTION/SETTING_RESP bit = 1

CMD_CODE_RESP = 0101b

CMD_RESP bit = 1

RESPONSE_DATA = Version Information

✓ LADDER

**NOTE 1:** 예제 28 의 래더는 예제 27 에서 추가된 명령으로, 앞의 예제가 생략 되었습니다.**NOTE 2:** 예제 28 는 같은 모션게이트에 연결된 모든 축으로 버전 정보 요청 명령을 실행하여도 동일한 정보를 얻습니다.



✓ 명령 순서

- ① X50A 의 Close 입력으로 MOTION/SETTING 비트(D000.7)을 세트 하여 IO-Map 을 설정 모드 상태로 전환합니다.
- ② CMD_CODE 영역(D000.8~D000.B)에 명령코드 '5'를 입력합니다.
- ③ 예제 10 을 참조하여 CMD_START 비트(D000.4)을 ON 하여 명령을 실행 합니다.
- ④ 모션게이트 버전 확인 명령의 응답은 IO-Map 의 조합으로 확인할 수 있습니다.
 - ▶ 모션게이트 버전 확인 명령의 응답은 해당 IO-Map 의 응답 비트인 MOTION/SETTING_RESP 비트가(D100.7)의 N.O 입력, CMD_CODE_RESP 영역 (d100.8~D100.B)을 '5'로 비교한 상태값, 그리고 CMD_RESP (D100.4)를 AND 회로로 구성된 결과 값이 됩니다.
- ⑦ 이 명령을 실행 후 얻은 데이터는 16 진수 형태로 4 개의 데이터를 수신 합니다. (예제 28 에서 확인된 버전은 1.0.0.0 입니다.)
 - ▶ D2018.0~D2018.7 구간의 데이터는 릴리즈 번호(Release No.)입니다.
 - ▶ D2018.8~D2018.F 구간의 데이터는 버그 수정 번호(Bug Fix. No.)입니다.
 - ▶ D2019.0~D2019.7 구간의 데이터는 마이너 버전(Minor Version)입니다.
 - ▶ D2019.8~D2019.F 구간의 데이터는 메이저 버전(Major Version)입니다.



- 사용자설명서의 일부 또는 전부를 무단 기재하거나 복제하는 것은 금지되어 있습니다.
- 손상이나 분실등으로 사용자설명서가 필요할때에는 본사 또는 가까운 대리점에 문의하여 주십시오.
- 사용자설명서는 제품의 개량이나 사양변경 및 사용자 설명서의 개선을 위해서 예고없이 변경되는 경우가 있습니다.
- Ezi-MOTIONGATE 는 국내에 등록된 FASTECH Co.LTD.의 등록상표입니다.

© Copyright 2008 FASTECH Co.,Ltd.
All Rights Reserved. Jul 25 2012 Rev.01.02.00

www.fastech.co.kr