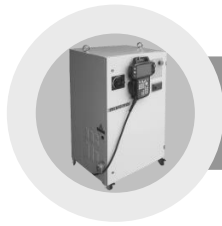




경고

모든 설치 작업은 반드시 자격 있는
설치기사에 의해 수행되어야 하며
관련 법규 및 규정을 준수하여야 합니다.





Hi5a 제어기 기능설명서

내장 PLC





본 제품 설명서에서 제공되는 정보는 현대로보틱스의 자산입니다.
현대로보틱스의 서면에 의한 동의 없이 전부 또는 일부를 무단 전재 및 재배포할 수 없으며,
제 3 자에게 제공되거나 다른 목적에 사용할 수 없습니다.

본 설명서는 사전 예고 없이 변경될 수 있습니다.

Printed in Korea - 2023 년 4 월. 10 판
Copyright © 2023 by Hyundai Robotics Co., Ltd



목 차

1. 개요	1-1
2. 제어기 설정.....	2-1
2.1. 내장 PLC의 모드 설정.....	2-2
2.2. 제어기 TP 상에서 릴레이 상태 모니터링	2-4
2.3. 내장 Scan Time	2-4
2.4. 내장 PLC 실행 할당시간 조정.....	2-4
3. 입출력 Diagram	3-1
4. 릴레이 사양	4-1
4.1. 릴레이 점수.....	4-2
4.2. 릴레이 설명	4-4
4.3. 릴레이 간접 주소 지정.....	4-26
4.4. 타이머 & 카운터 릴레이.....	4-27
5. 명령어 사양.....	5-1
5.1. 명령어 일람	5-2
5.2. 사용 가능한 Operands.....	5-5
5.3. 래더 Diagram 과 니모닉의 관계	5-8
6. 명령어 설명	6-1
6.1. XIC(Examine if Closed) : Close 검사.....	6-2
6.2. XIO(Examine if Open) : Open 검사	6-2
6.3. INV(Inverting) : 인버팅	6-3
6.4. EQU(Equal) : 같은지 검사.....	6-3
6.5. NEQ(Not Equal) : 다른지 검사.....	6-4
6.6. LES(Less Than) : 작은지 검사.....	6-4
6.7. GRT(Greater Than) : 큰지 검사.....	6-5
6.8. LEQ(Less Than or Equal) : 작거나 같은지 검사.....	6-5
6.9. GEQ(Greater Than or Equal) : 크거나 같은지 검사	6-6
6.10. OTE(Output Energize) : 일반출력.....	6-6
6.11. OTL(Output Latch) : 래치 출력.....	6-7
6.12. OTU(Output Unlatch) : 언래치 출력	6-7
6.13. OSR(One Shot Rising) : 원 샷 출력.....	6-8
6.14. RES(Reset) : 리셋	6-8
6.15. TON(Time On Delay) : 타이머	6-9

6.16. CTD(Count Down) : 카운터.....	6-10
6.17. ADD(Add) : 더하기.....	6-10
6.18. SUB(Subtract) : 빼기.....	6-11
6.19. MUL(Multiply): 곱하기.....	6-11
6.20. DIV(Divide): 나누기.....	6-12
6.21. POW(Power): 거듭제곱.....	6-12
6.22. TOD(Convert to BCD): BCD 값으로 변환.....	6-13
6.23. FRD(Convert form BCD to Integer): integer 로 변환.....	6-14
6.24. SEG(7'Segment): 7' 세그먼트 값으로 변환.....	6-15
6.25. MOV(Move): 이동.....	6-16
6.26. COP(Copy data): 복사.....	6-17
6.27. CCOP(Conditional Copy data): 조건 별 복사.....	6-18
6.28. ROT(Rotating Output): 순차 출력.....	6-19
6.29. FOR(FOR): 블록 반복.....	6-20
6.30. NEXT(NEXT): 블록 next.....	6-21
6.31. LBL(Label): 레이블 지정.....	6-21
6.32. JMP(Jump): 점프.....	6-22
6.33. CALL(Call): 서브래더 호출.....	6-23
6.34. [(End): 레더 종료.....	6-24



그림 목차

그림 1.1 Hi5a I/O Connection.....	1-2
그림 2.1 내장 PLC 모드 설정.....	2-2
그림 2.2 내장 PLC Off 상태.....	2-2
그림 2.3 내장 PLC STOP 상태.....	2-2
그림 2.4 내장 PLC 실행시간 설정.....	2-4
그림 3.1 입출력 Diagram.....	3-2

표 목차

표 4-1 릴레이 점수.....	4-2
표 5-1 명령어 표.....	5-2
표 5-2 사용 가능한 Operands	5-5





HD

HYUNDAI
ROBOTICS

1

개요



1. 개요

내장 PLC

Hi5a 제어기의 내장 PLC 는 상용 PLC 의 기능을 제어기에 구현한 기능입니다. 아래 그림과 같이, 제어기에 연결된 PC 나 노트북에서 HRLadder 를 실행하여 래더 프로그램을 작성/편집하고, 제어기에 다운로드 하거나, 제어기에서 실행되고 있는 래더 프로그램을 업로드 할 수 있으며, 제어기에서 실행되고 있는 상태를 모니터링 하는 것도 가능합니다. HRLadder 는 별도 기능설명서를 참조하십시오.

Hi5a 제어기의 I/O 연결은 상위 공정반과 필드버스 또는 리얼타임 이더넷으로 연결할 수 있고, 내장 DeviceNet 마스터를 이용하여 DeviceNet 슬레이브 모듈들을 연결할 수 있으며, BD580 이나 BD58A 를 이용하여 여러 장치들을 물리적인 신호로 직접 연결할 수도 있습니다. 내장 PLC 기능은 상기와 같이 연결된 입출력 신호들을 Ladder Logic 으로 제어하는 기능입니다.

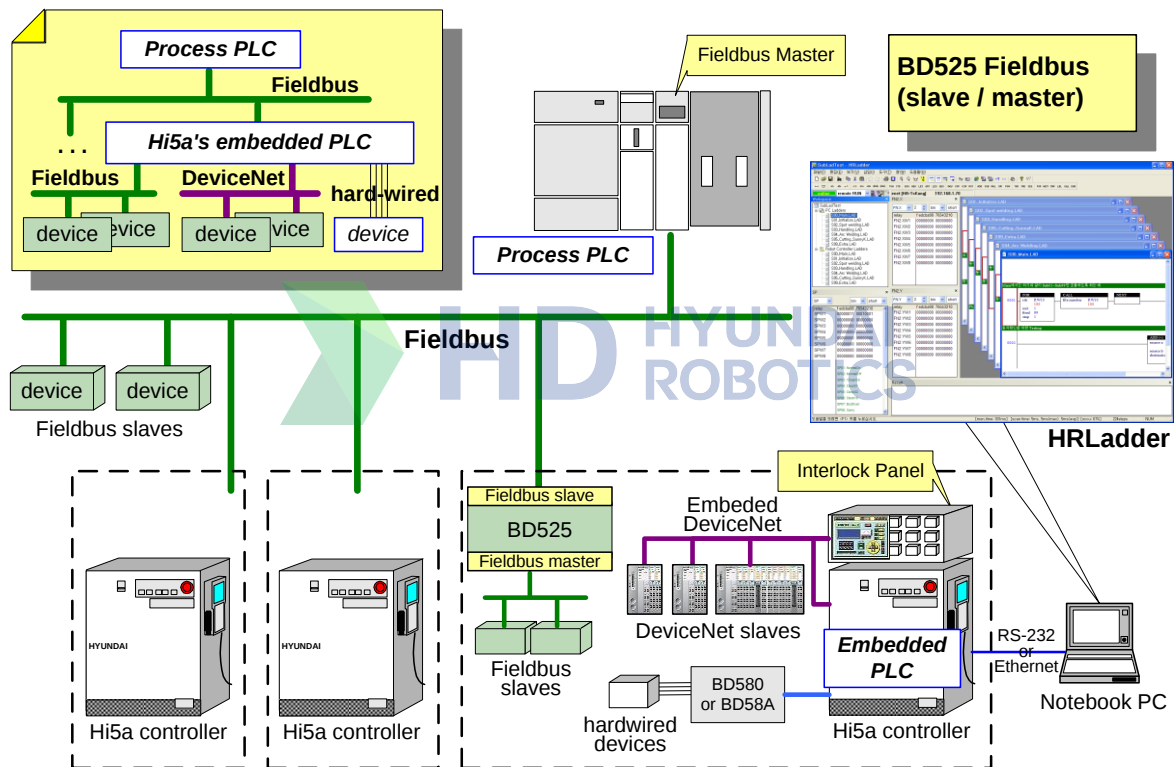


그림 1.1 Hi5a I/O Connection



HD

HYUNDAI
ROBOTICS

2

제어기 설정



2. 제어기 설정

내장 PLC

2.1. 내장 PLC의 모드 설정

내장 PLC의 동작모드는 『[F7]: 조건설정』 → 『[F1]: 응용조건』 → 『5: 내장 PLC 모드』에서 <Off, Stop, R-Stop, R-Run, Run>중 하나를 선택할 수 있습니다.

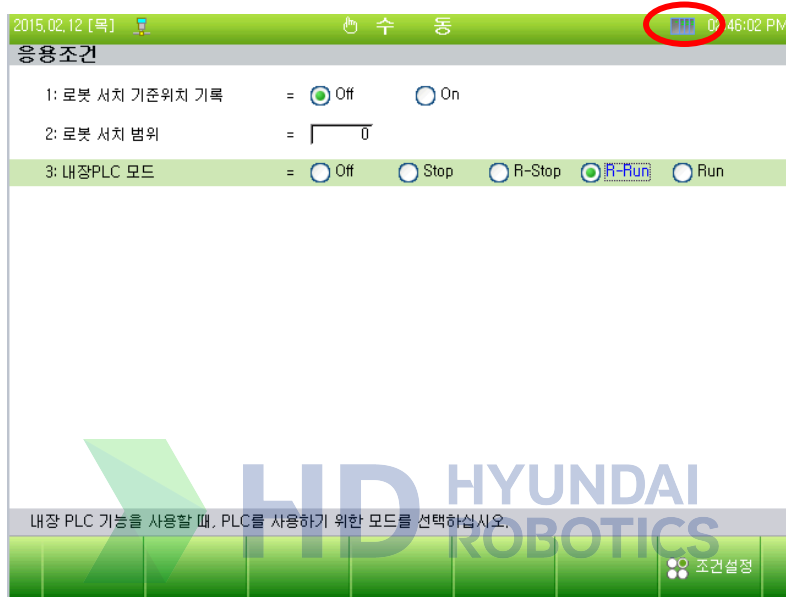


그림 2.1 내장 PLC 모드 설정

선택된 모드에 따라 TP 화면 우측상단에 아이콘으로 상태를 표시합니다. 즉, PLC=<R-Run 이나 Run>상태이면 상기 그림과 같이 PLC 그림이 표시되고, PLC=Off 인 경우에는 하기 그림과 같이 아이콘이 사라지며, PLC=STOP 상태이면 PLC 그림에 적색금지마크가 표시됩니다.

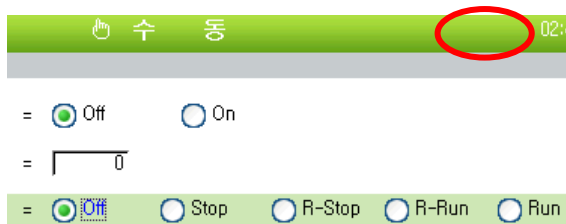


그림 2.2 내장 PLC Off 상태

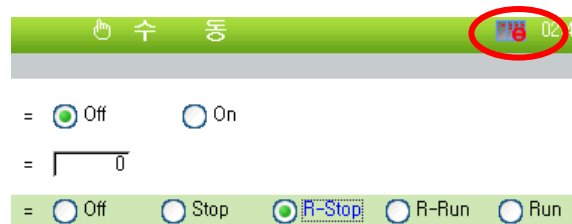


그림 2.3 내장 PLC STOP 상태

■ Off

내장 PLC 의 기능이 무효로 됩니다. 이 경우, 로봇제어기의 디지털 출력(Digital Output)인 DO1~DO4096 이 물리적 출력(Physical Output)인 Y1~Y4096 으로 자동 출력되고, 물리적 입력인 X1~X4096 이 DI1~DI4096 으로 자동 입력됩니다. 이 경우에도 DI, DO, X, 그리고 Y 에 대한 속성이 각각 적용됩니다. 예를 들어 DO1 이 부논리이고, Y1 도 부논리인 경우, DO1=1 로 출력하면 DO1 이 부논리라서 0 가 Y1 로 입력되고 Y1 의 속성도 부논리이므로 실제 출력되는 값은 Y1=1 이 됩니다.

■ R-Stop / Stop

내장 PLC 의 동작을 정지시킵니다. R-Stop 은 HRLadder 에서 변경할 수 있는 리모트(Remote)상태를 나타내며, STOP 으로 설정하면 HRLadder 에서 동작모드를 변경할 수 없습니다. 내장 PLC 가 Stop 될 때에는 PLC 의 출력신호인 DI, Y, R 릴레이가 자동으로 0 이 됩니다. 다만, PLC 외의 HRBasic 이나 할당에서도 Y 출력에 값을 출력할 수 있는 조건인 SP11=On 인 경우에는 Y 출력은 이전 값을 유지합니다.

■ R-Run / Run

내장 PLC 를 실행시킵니다. R-Run 은 HRLadder 에서 변경할 수 있는 리모트(Remote)상태를 나타내며, Run 으로 설정하면 HRLadder 에서 동작모드를 변경할 수 없습니다.



2.2. 제어기 TP 상에서 릴레이 상태 모니터링

『[F1]: 서비스』 → 『1: 모니터링』 → 『7: PLC Relay 데이터』 → 『1: PLC X Relay(외부입력)』 → 『11: PLC SW Relay(시스템 메모리)』로 릴레이 상태를 모니터링 할 수 있습니다.

2.3. 내장 Scan Time

HRLadder의 하단 상태 바에 “scan time”로 표시되며, 스텝수가 증가하여 5msec 안에 처리할 수 없으면 scan time 이 자동으로 증가(5msec 단위)합니다.

2.4. 내장 PLC 실행 할당시간 조정

특수한 용도에서 내장 PLC 스캔 타임을 빠르게 설정할 필요가 있을 경우, 당사 엔지니어에게 요청하여 Hi5a 로봇에서 PLC를 위한 배당시간을 조정할 수 있습니다. 수동모드에서 『[F2]: 시스템』 → 『2: 제어 파라미터』 → 『1: 제어 환경 설정』 → 『7: PLC 실행시간 설정』에서 배당시간을 조정하기 위한 메뉴가 준비되어 있습니다. PLC 실행시간 설정이란 로봇제어기에서 처리해야 하는 작업들 중에 PLC 처리를 위해 얼마의 시간을 배정할 것인가를 설정하는 것이며, 디폴트 값은 2.0/20[ms]으로 되어 있습니다만, 이 값을 너무 크게 설정하면, 로봇이동을 위한 모션계산 등을 처리하지 못할 수 있으므로 반드시 당사 엔지니어에게 요청하시어 변경하고자 하는 시스템의 잉여자원이 얼마나 되는지를 조사한 다음, 이를 바탕으로 배당시간을 결정해야 합니다.

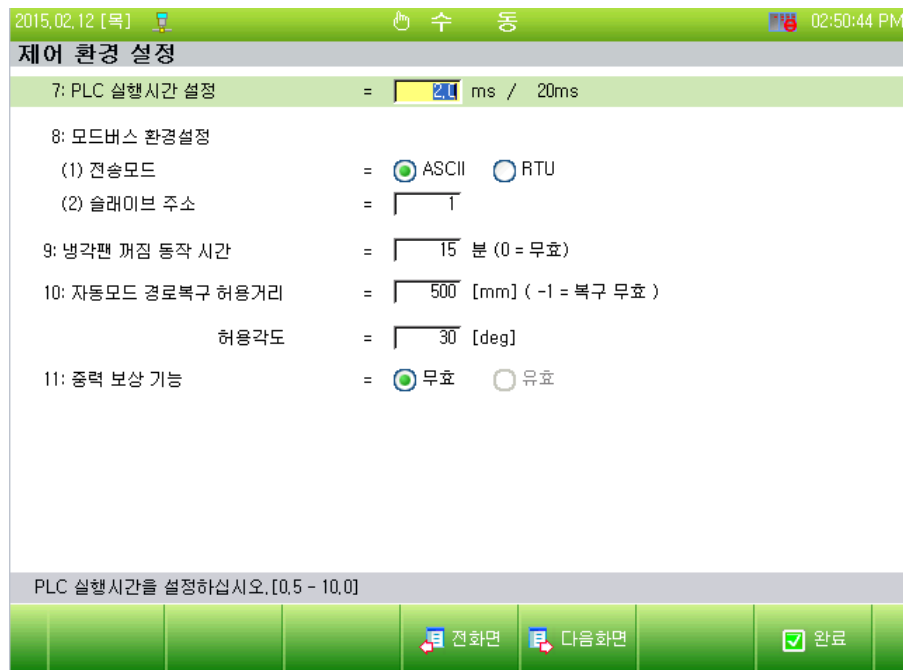


그림 2.4 내장 PLC 실행시간 설정



HD

HYUNDAI
ROBOTICS

3

입출력
Diagram



3. 입출력 Diagram

내장 PLC

Hi5a 로봇제어기의 입출력 Diagram 은 다음 그림과 같습니다.

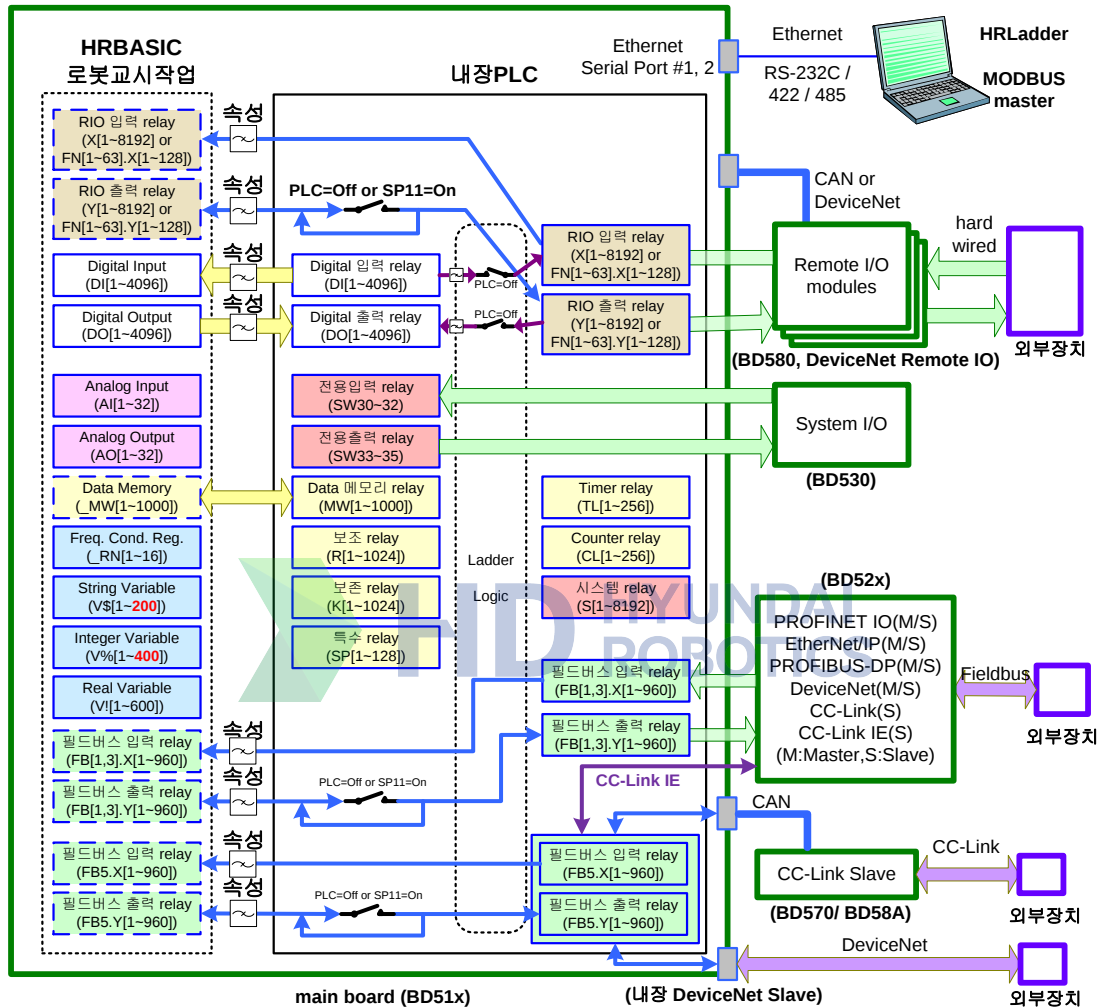


그림 3.1 입출력 Diagram

- PLC 사용 중이더라도 SP11 을 On 하면 로봇 작업 프로그램(HRBASIC)에서 RIO 출력(Y[1~8192], FN[1~63].Y[1~128]) 또는 필드버스 출력(FB[1,3,5].Y[1~960]) 릴레이를 직접 제어할 수 있습니다.

그러나 입력 릴레이들(X[1~8192], FN[1~63].X[1~128], FB[1,3,5].X[1~960])의 경우에는 PLC 사용 유무와 관계 없이 로봇 작업 프로그램(HRBASIC)에서 항상 읽을 수 있습니다.

- 상기 그림의 시스템 메모리(System Memory)는 특수한 용도로 사용하기 위해 예약(reserved)된 것으로서, 향후 제어기 버전에 따라 다른 용도로 변경될 수 있습니다.



HD

HYUNDAI
ROBOTICS

4

릴레이 사양



4. 릴레이 사양

내장 PLC

4.1. 릴레이 점수

표 4-1 릴레이 점수

릴레이 명칭	점수	릴레이(1bit)	릴레이(Byte, Word)	(Long, Float)
디지털 입출력 릴레이	입력 4096 점	DI1~4096	DIB1~512	DIL1~128
			DIW1~256	DIF1~128
	출력 4096 점	DO1~4096	DOB1~512	DOL1~128
			DOW1~256	DOF1~128
RIO 입출력 릴레이 (BD580/DeviceNet)	입력 8192 점 노드당 128 점 (Max 64 노드)	FN1~64.X1~128 (or X1~8192)	FN1~64.XB1~16 (or XB1~1024)	FN1~64.XL1~4 (or XL1~256)
			FN1~64.XW1~8 (or XW1~512)	FN1~64.XF1~4 (or XF1~256)
	출력 8192 점 노드당 128 점 (Max 64 노드)	FN1~64.Y1~128 (or Y1~8192)	FN1~64.YB1~16 (or YB1~1024)	FN1~64.YL1~4 (or YL1~256)
			FN1~64.YW1~8 (or YW1~512)	FN1~64.YF1~4 (or YF1~256)
필드버스 릴레이 (PROFINET IO, EtherNet/IP, CC-Link IE, PROFIBUS-DP, DeviceNet)	입력 960 점	FB1.X1~960	FB1.XB1~120	FB1.XL1~30
			FB1.XW1~60	FB1.XF1~30
	출력 960 점	FB1.Y1~960	FB1.YB1~120	FB1.YL1~30
			FB1.YW1~60	FB1.YF1~30
	입력 960 점	FB3.X1~960	FB3.XB1~120	FB3.XL1~30
			FB3.XW1~60	FB3.XF1~30
	출력 960 점	FB3.Y1~960	FB3.YB1~120	FB3.YL1~30
			FB3.YW1~60	FB3.YF1~30
필드버스 릴레이 (CC-Link, CC-Link IE)	입력 960 점	FB5.X1~960	FB5.XB1~120	FB5.XL1~30
			FB5.XW1~60	FB5.XF1~30
	출력 960 점	FB5.Y1~960	FB5.YB1~120	FB5.YL1~30

4. 릴레이 사양

릴레이 명칭	점수	릴레이(1bit)	릴레이(Byte, Word)	(Long, Float)
			FB5.YW1~60	FB5.YF1~30
보조 릴레이	1024 점	R1~1024	RB1~128	RL1~32
			RW1~64	RF1~32
보존 릴레이	1024 점	K1~1024	KB1~128	KL1~32
			KW1~64	KF1~32
특수 릴레이	128 점	SP1~128	SPB1~16	SPL1~4
			SPW1~8	SPF1~4
타이머 릴레이	256 점	T1~256	TB1~1024	TL1~256
			TW1~512	TF1~256
카운터 릴레이	256 점	C1~256	CB1~1024	CL1~256
			CW1~512	CF1~256
데이터 메모리	16000 점	M1~16000	MB1~2000	ML1~500
			MW1~1000	MF1~500
시스템 메모리	8192 점 (Reserved)	S1~8192 (Reserved)	SB1~1024	SL1~256
			SW1~512	SF1~256

참고) 릴레이 형식: Byte=8bit, Word=16bit, Long=32bit, Float=(32bit)

4.2. 릴레이 설명

각각의 릴레이에 대해 접두문자(B/W/L/F)를 사용하여 bit, byte, word, long, 그리고 float 형식으로 지정할 수 있습니다. 다만, 타이머와 카운터의 bit(T/C)는 long(TL/CL)의 값이 0 이면 활성화(activation)이고, 0 이 아니면 비활성입니다.

- (1) 디지털 입출력 릴레이: HRBasic 이나 각종 입출력 할당에서 사용할 수 있는 신호입니다.
- (2) RIO 입출력 릴레이:
BD580 보드의 입출력 신호들이나, 내장 디바이스넷 마스터에 연결된 디바이스넷 슬레이브 노드들의 입출력 신호들입니다.
- (3) 필드버스 릴레이:
BD521, BD522, BD525, BD570, BD58A 보드 등을 통해 연결된 필드버스 또는 리얼타임 이더넷의 입출력 신호들입니다.
- (4) 보조 릴레이: PLC 프로그램에서 사용하는 보조 릴레이입니다.
- (5) 보존 릴레이: 전원을 Off 해도 On/Off 상태가 보존되는 릴레이입니다.
- (6) 특수 릴레이: 특수한 목적을 위해 정의된 릴레이입니다. 4.2.1 절을 참조하세요.
- (7) 타이머 릴레이:
타이머 동작을 위한 릴레이며, 값이 0 일 때 접점이 On 됩니다. (정전 시 보존)
- (8) 카운터 릴레이:
카운터 동작을 위한 릴레이며, 값이 0 일 때 접점이 On 됩니다. (정전 시 보존)
- (9) 데이터 메모리:
응용명령에서 임의의 Data 를 저장하거나 읽어 들일 때 사용하며, HRBasic 에서도 access 할 수 있으므로 HABasic 과 많은 양의 데이터를 교신할 때에도 사용할 수 있습니다. (정전 시 보존)
- (10) 시스템 메모리(Reserved):
제어기 내의 시스템 값을 읽거나 쓰는 용도로 사용합니다. 4.2.2 절을 참조하세요.

4.2.1. SP 릴레이

릴레이 번호	설명	기타
SP01	상시 On 릴레이	Controller states
SP02	상시 Off 릴레이	
SP03	운전 시작의 한 스캔만 On 되는 릴레이	Internal timer
SP04	0.1 초 clock (0.05 초 On → 0.05 초 Off)	
SP05	0.2 초 clock (0.1 초 On → 0.1 초 Off)	
SP06	1 초 clock (0.5 초 On → 0.5 초 Off)	
SP07	BCD 연산에서 변환이 안될 경우에 On.	TOD 나 FRD 실행 시
SP08	연산결과 carry 가 있으면 On.	산술명령 실행 시
SP09	2 초 clock (1 초 On → 1 초 Off)	Internal timer
SP10	4 초 clock (2 초 On → 2 초 Off)	
SP11	On 이면, PLC 외의 응용(HRBasic, 할당)에서 Y 릴레이에 직접 출력하는 것을 허용	
SP12	Label 이 상수가 아닐 때 On	
SP13	Label 의 개수가 100 이상일 때 On	
SP14	Label 이 중복될 때 On	
SP15	Label 이 0 이하이거나, Jump 할 Label 번호가 없을 경우에 On	
SP16	Simulation 상태에서 Modbus Test 용	
SP17	Scan Time 이 5 초를 초과할 때 On (이 경우 5 초 이후에 있는 명령은 실행되지 않습니다)	
SP18	Call 로 호출되는 Subladder 가 없을 경우에 On	
SP19	0 = 내장 디바이스넷 마스터 에러/경고 활성화(기본 값) 1 = 내장 디바이스넷 마스터 에러/경고 생략	V40.06-00 버전부터 지원

릴레이 번호	설명	기타
SP20	0 = FB1 통신 채널 에러/경고 활성화(기본 값) 1 = FB1 통신 채널 에러/경고 생략	
SP21	0 = FB3 통신 채널 에러/경고 활성화(기본 값) 1 = FB3 통신 채널 에러/경고 생략	
SP22	0 = FB5 통신 채널 에러/경고 활성화(기본 값) 1 = FB5 통신 채널 에러/경고 생략	
SP23~SP128	Reserved	

4.2.2. SW 릴레이

번호	설 명	기 타
SW3	PLC 실행모드(4: PLC OFF, 5: 프로그램 없음, 0:STOP, 1:R.STOP, 2:R.RUN, 3:RUN)	
SW4	확장 IO 보드 수	
SW5	Main SW Version 의 2 nd + 1 st	40.05-04v → &H05 28
SW6	I/O version + Main SW Version 의 3 rd	40.05-04 → &H43 04
SW10	Scan time	
SW11	할당 시간	
SW12	최대 점유시간	
SW13	평균 점유시간	
SW14	Ladder 의 총 스텝수	
SW15	Ladder 프로그램의 CHECKSUM	
SW17	점유율 (%)	
SW20	제어기의 현재 프로그램 번호	

4. 릴레이 사양

번호	설 명	기 타
SW21	제어기의 현재 스텝 번호	
SW22	제어기의 현재 평선 번호	
SW23	제어기의 메인 프로그램 번호	
SW24	FB1~FB4 active	0th~3rd bit
SW25	힐서 상태정보 읽기(1~4) (1:FB1 정보, 2: FB2 정보...)	
SW26	LSB: 힐서 COM GlobalBits MSB: reserved	(마스터만 유효)
SW27	LSB: 에러노드번호 MSB: 에러코드	(마스터만 유효)
SW28	버스에러 카운터	(마스터만 유효)
SW29	DeviceNet: 버스 OFF 카운터 Profibus-DP: 타임아웃 카운터	(마스터만 유효)
SW30	전용 입력 1	
SW31	전용 입력 2	
SW32	전용 입력 3	
SW33	전용 출력 1	
SW34	전용 출력 2	
SW35	전용 출력 3	
SW37	LSB: 지정한 node 번호에 대해 1 초마다 slave_diag 요청	
SW38	slave_diag 요청에 대한 리턴값	
SW39	slave_diag 요청 결과의 StationStatus LSB: StationStatus1 MSB: StationStatus2	
SW40~59	로봇 현재 pose 읽기. 4.2.3 절, 4.2.4 절 참조.	

번호	설 명	기 타
SW60~69	간접주소지정을 위한 영역(4.3 간접 주소 지정 참조)	-1:SW61 ~ -9:SW69
SW70	가동시간 종류 (0:무, 1:가동시간(누적), 2:통산가동시간) 주의) SW(n):MSW(상위), SW(n+1):LSW	
SW71~SW72	측정시간(10msec 단위)	
SW73~SW74	싸이클 시간(10msec 단위)	
SW75	싸이클 수	
SW76~SW77	용접기 1 용접시간(10msec 단위)	
SW78	용접기 1 용접횟수	
SW79~SW80	용접기 2 용접시간(10msec 단위)	
SW81	용접기 2 용접횟수	
SW82~SW83	용접기 3 용접시간(10msec 단위)	
SW84	용접기 3 용접횟수	
SW85~SW86	용접기 4 용접시간(10msec 단위)	
SW87	용접기 4 용접횟수	
SW88~SW89	Wait, DI 대기 시간(10msec 단위)	
SW90~SW91	타이머 대기 시간(10msec 단위)	
SW70	가동시간 종류 (0:무, 3:최근 가동시간, 4:통산가동시간) 최근 가동시간을 추가하면서, SL 를 사용할 수 있도록 SW 를 Swapping 시킴 즉, SW(홀수):LSW(하위), SW(짝수):MSW	
SL36	측정시간(10msec 단위)	

4. 릴레이 사양

번호	설 명	기 타
SL37	싸이클 시간(10msec 단위)	
SL38	싸이클 수	
SL39	용접기 1 용접시간(10msec 단위)	
SL40	용접기 1 용접횟수	
SL41	용접기 2 용접시간(10msec 단위)	
SL42	용접기 2 용접횟수	
SL43	용접기 3 용접시간(10msec 단위)	
SL44	용접기 3 용접횟수	
SL45	용접기 4 용접시간(10msec 단위)	
SL46	용접기 4 용접횟수	
SL47	Wait, DI 대기 시간(10msec 단위)	
SL48	타이머 대기 시간(10msec 단위)	10msec 단위
SW70	가동시간 종류 (5:가동정보(아크)) 최근 가동시간을 추가하면서, SL 를 사용할 수 있도록 SW 를 Swapping 시킴 즉, SW(홀수):LSW(하위), SW(짝수):MSW	
SL36	측정시간(10msec 단위)	
SL37	싸이클 시간(10msec 단위)	
SL38	싸이클 수	
SL39	아크용접 수행시간(10msec 단위), 1 일 단위로 리셋	
SL40	아크용접 수행일자(1 일 단위)	
SW81	ARC Retry 수행 횟수	

번호	설 명	기 타																
SW82	ARC Overlap 수행 횟수. 1:Arc Off																	
SW83	ARC Overlap 수행 횟수. 2:Wire Off																	
SW84	ARC Overlap 수행 횟수. 3:Limit Over																	
SW85	ARC Overlap 수행 횟수. 4:Gas Off																	
SW86	ARC Overlap 수행 횟수. 5:Coolant Off																	
SW87	ARC 자동용착해제 수행 횟수																	
SL45	1 Cycle 동안 아크 가동시간																	
SL46	팁 사용 시간																	
SL47	팁 사용 시간 제한치																	
SW99	모드스위치 절환 시 모터Off 로 처리할지 설정 (0: 미처리, 1: 수동(-)자동 전환 시 모터Off , 2 : 모드 전환 시 모터 Off)																	
SW100~ SW109	Program Count 관련 reserved																	
SW110	각축 속도 Enable(SW110=0xKLMN, 16 진수방식) <table border="1"><tr><td>B15</td><td>B14</td><td>B13</td><td>B12</td><td>B11</td><td>B10</td><td>B9</td><td>B8</td><td>B7</td><td>B6</td><td>B5</td><td>B4</td><td>B3</td><td>B2</td><td>B1</td><td>B0</td></tr></table> N(B3~B0): 분해능(0x1=10 배,0x2=100 배,0x3=1000 배) M: 기타옵션(B4:1=절대값, B5:0=축속도,1=모터속도)	B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0	Bit 별 설정값 정한 후 10 진수로 변경하여 입력하여야 함. Ex) 분해능 1000 배, 절대값없음, 모터속도 설정일 경우 N = 0x3, M = 0x2 => 0x23 = '35' 입력하기
B15	B14	B13	B12	B11	B10	B9	B8	B7	B6	B5	B4	B3	B2	B1	B0			
SW111~ SW126	각축(또는 모터)의 속도(16 축분)																	
SW130~ SW133	TP KEYPAD 정보																	
SW135	Get Applet																	
SW136	Set Applet																	
SW140	내장 디바이스넷 마스터 에러 슬레이브 리스트(0~15)	Bit0=1: Node 0 Error Bit15=1: Node 15 Error																

번호	설 명	기 타
SW141	내장 디바이스넷 마스터 에러 슬레이브 리스트(16~31)	Bit0=1: Node 16 Error Bit15=1: Node 31 Error
SW142	내장 디바이스넷 마스터 에러 슬레이브 리스트(32~47)	Bit0=1: Node 32 Error Bit15=1: Node 47 Error
SW143	내장 디바이스넷 마스터 에러 슬레이브 리스트(48~63)	Bit0=1: Node 48 Error Bit15=1: Node 63 Error
SW144	내장 디바이스넷 마스터 상태*1)	
SW145	내장 디바이스넷 마스터 연결 슬레이브 상태 정보 요청을 위한 노드 번호	
SW146	SW145 노드 번호의 슬레이브 상태*2)	
SW150	Arc Welding Info	1=Enable(Digital Only)
SW151	Arc Welding Real Current	
SW152	Arc Welding Real Voltage	
SW153	Arc Welder Error1	
SW154	Arc Welder Error2	
SW155	Wire Feeding Speed	

*1) 내장 디바이스넷 마스터 상태

Bit 15~9 : Reserved

Bit 8 : 초기화 상태(1=초기화 수행 중, 0=초기화 완료)

Bit 4 : 마스터 운전 상태(0=RUN, 1=IDLE(PLC STOP))

Bit 3~0 : 디바이스넷 상태(0=Running, 1=Reset(Out-of-box), 2=Init(Out-of-box),
3=Reset(Normal), 4=Init(Normal), 5=Duplicated MAC ID Check,
6=Fault)

*2) 슬레이브 상태

Bit 15 : 1=사용 노드, 0=비 사용 노드

Bit 14 : 제조사(1=CREVIS, 0=기타)

Bit 13~12 : 노드 상태(1=Standby, 2=Connection Fault, 3=Configuration Fault)

Bit 11 : 0=IO mode(마스터와 IO 데이터 교환 상태), 1=IO mode가 아님

Bit 9~0 : 슬레이브 상태

0x000=Online, 0x001=Offline, 0x002=IO Connection Closed,

0x004=Input Size Error, 0x008=Output Size Error, 0x010=Vendor ID mismatch,

0x020=Device type mismatch, 0x040=Production code mismatch,

0x080=CCV mismatch, 0x100=IO size is too big, 0x200=No connection

번호	설 명	기 타
SW156	Arc Welding SI 1~8 번(하위 byte), SO 1~8 번(상위 byte)	V40.04-00 버전부터 지원
SW160	디바이스넷 Explicit message 요청 플래그 - 0x0001= 요청 디바이스넷 Explicit message 응답 플래그 - 0x001X=Explicit message 요청 진행 중 - 0x0000=Explicit message 처리 완료 - 0x002X=Explicit message 타임 아웃 - 기타= 에러 응답	결과 =>SW166 ~ SW179
SW161	Explicit message Request: Node ID	
SW162	Explicit message Request: Service	
SW163	Explicit message Request: Class	
SW164	Explicit message Request: Instance	
SW165	Explicit message Request: Attribute	
SW166	Explicit message Request/Response: Size of Data	Max. 26
SW167~ SW179	Explicit message Request/Response: Service Data (SB333~SB358)	Max. 26 bytes
SW180	SPOT GUN 의 마모량을 얻기 위한 GUN 번호 (0: 무효, 1~: GUN 번호)	
SW181	이동전극 마모량	
SW182	고정전극 마모량	
SW183	건서치 상태	
~SW189	SPOT GUN 을 위한 reserved	
SW190	MON_AXIS_CTRL_OFF	Bit 당 1 축
SW195	롤러헤밍 가압력[N]	

4. 릴레이 사양

번호	설 명	기 타
SW200	축별 누적 거리 모니터링 모드 (dword 지원은 V31.11-00~, V32.01-00~)	0:Off, 1:read, 2:write(word) 3:read, 4:write(dword)
SW201~ SW216	축별 누적 거리 모니터링 (1~16 축) dword 인 경우는 SL101~SL108 (1~8 축)	단위: km, rad 오버플로우 발생하면 다시 0 이 됨.
SW220	슬롯 1 코드 (0: 기능 Off)	[_IDX] 0-base: 값 읽기 30000-base: 값 설정 예 1) SW223 에 352 설정하면, SW224 에 AO2 값 반영됨.
SW221	슬롯 1 데이터 하위 워드	
SW222	슬롯 1 데이터 상위 워드	
SW223~ SW224, 225	슬롯 2 코드 슬롯 2 데이터 하위, 상위 워드	
SW226~ SW227, 228	슬롯 3 코드 슬롯 3 데이터 하위, 상위 워드	
SW229~ SW230, 231	슬롯 4 코드 슬롯 4 데이터 하위, 상위 워드	
SW232~ SW233, 234	슬롯 5 코드 슬롯 5 데이터 하위, 상위 워드	
SW235~ SW236, 237	슬롯 6 코드 슬롯 6 데이터 하위, 상위 워드	
SW238~ SW239, 240	슬롯 7 코드 슬롯 7 데이터 하위, 상위 워드	
SW241~ SW242, 243	슬롯 8 코드 슬롯 8 데이터 하위, 상위 워드	
SW244~ SW245, 246	슬롯 9 코드 슬롯 9 데이터 하위, 상위 워드	
SW247~ SW248, 249	슬롯 10 코드 슬롯 10 데이터 하위, 상위 워드	
SW250	(서브태스크 1) 생성 상태	
SW251	(서브태스크 1) 현재 프로그램 번호	
SW252	(서브태스크 1) 현재 스텝 번호	
SW253	(서브태스크 1) 현재 평선 번호	
~SW259	(서브태스크 1)을 위한 reserved	

번호	설 명	기 타
SW260	(서브태스크 2) 생성 상태	
SW261	(서브태스크 2) 현재 프로그램 번호	
SW262	(서브태스크 2) 현재 스텝 번호	
SW263	(서브태스크 2) 현재 평선 번호	
~SW269	(서브태스크 2)을 위한 reserved	
SW270	(서브태스크 3) 생성 상태	
SW271	(서브태스크 3) 현재 프로그램 번호	
SW272	(서브태스크 3) 현재 스텝 번호	
SW273	(서브태스크 3) 현재 평선 번호	
~SW279	(서브태스크 3)을 위한 reserved	
SW280	(서브태스크 4) 생성 상태	
SW281	(서브태스크 4) 현재 프로그램 번호	
SW282	(서브태스크 4) 현재 스텝 번호	
SW283	(서브태스크 4) 현재 평선 번호	
~SW289	(서브태스크 4)을 위한 reserved	
SW290	(서브태스크 5) 생성 상태	
SW291	(서브태스크 5) 현재 프로그램 번호	
SW292	(서브태스크 5) 현재 스텝 번호	
SW293	(서브태스크 5) 현재 평선 번호	
~SW299	(서브태스크 5)을 위한 reserved	

4. 릴레이 사양

번호	설 명	기 타
SW300	(서브태스크 6) 생성 상태	
SW301	(서브태스크 6) 현재 프로그램 번호	
SW302	(서브태스크 6) 현재 스텝 번호	
SW303	(서브태스크 6) 현재 평선 번호	
~SW309	(서브태스크 6)을 위한 reserved	
SW310	(서브태스크 7) 생성 상태	
SW311	(서브태스크 7) 현재 프로그램 번호	
SW312	(서브태스크 7) 현재 스텝 번호	
SW313	(서브태스크 7) 현재 평선 번호	
~SW319	(서브태스크 7)을 위한 reserved	
SW320	현재 로봇의 톨 번호	
SW321	PRM 회생전력: 현재 값(하위 16Bit)	SL161
SW322	PRM 회생전력: 현재 값(상위 16Bit)	
SW323	PRM 회생전력: 누적 값(하위 16Bit)	SL162
SW324	PRM 회생전력: 누적 값(상위 16Bit)	
SW325	PRM 회생전력: 5 분 평균값(하위 16Bit)	SL163
SW326	PRM 회생전력: 5 분 평균값(상위 16Bit)	
~SW329	Reserved	
SW330	BD525 마스터: 장치 상태 ^{*3)}	

*3) BD525 마스터: 장치 상태

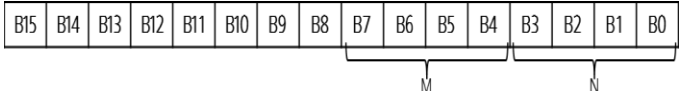
번호	설 명	기 타
SW331	BD525 마스터: 설정 슬레이브 개수	0~65535
SW332 ~ SW339	BD525 마스터: 설정 슬레이브 리스트	비트 리스트
SW340	BD525 마스터: 통신 상태(하위 8bit) ^{*4)} BD525 마스터: 슬레이브 상태(상위 8Bit) ^{*5)}	
SW341	BD525 마스터: 동작 슬레이브 개수	0~65535
SW342 ~ SW349	BD525 마스터: 동작 슬레이브 리스트	비트 리스트
SW350	BD525 마스터: 통신 에러 누적 횟수	0~65535
SW351	BD525 마스터: 에러 슬레이브 개수	0~65535
SW352 ~ SW359	BD525 마스터: 에러 슬레이브 리스트	비트 리스트
SW360	CC-Link IE 링크 및 운전 상태	CC-Link IE 기능 설명서 참조
SW361	CC-Link IE 사이클릭 통신 상태	CC-Link IE 기능 설명서 참조
SW366	하위바이트 : <code>iot_converter_client</code> 상위바이트 : <code>iot_converter_server</code>	Hi5a V40.25-00 부터 지 원 ON: 1, OFF: 0
SW367	하위바이트 : <code>iot_connector</code> 상위바이트 : <code>iot_ftp</code>	

Bit 0: 1=준비(Ready)
 Bit 1: 1=운전(Running)
 Bit 2: 1=통신(Bus On)
 Bit 3: 1=설정 잠금(Configuration locked)
 Bit 4: 1=새로운 설정 대기(New Configuration)
 Bit 5: 1=리셋 필요(Restart required)
 Bit 6: 1=리셋 활성화(Restart required Enable)
 Bit 7~31: Reserved

^{*4)} BD525 마스터: 통신 상태
 0=Unknown, 1=NOT Configured, 2=STOP, 3=IDLE, 4=OPERATE

^{*5)} BD525 마스터: 슬레이브 상태
 0=UNDEFINED, 1=OK(No Fault), 2=FAILED(하나 이상의 Slave 에러)

4. 릴레이 사양

번호	설 명	기 타
SW370 ~ SW385	축 별 엔코더 온도 (J1~J16) 서비스 / 모니터링 / 시스템 특성 데이터 / 모터 엔코더 데이터 / 각축 엔코더 온도의 현재값을 모니터링 참고) 모터 온도를 간접적으로 모니터링	Hi5a V40.15-00 부터 지 원 ° C
SW388	main b/d 온도 (단위: 0.1° C)	Hi5a V40.17-00 부터 지 원
SW389	main b/d 의 CPU 온도 (단위: 0.1° C)	Hi5a V40.17-00 부터 지 원
SW390	제어기 시리얼 번호 서비스 / 시스템 진단 / 시스템 버전화면에 일련번호를 모니터링	Hi5a V40.15-00 부터 지 원
~SW394	Reserved	
SW395	로봇 상태 모니터링 (0: stop, 1: run, 2: wait)	Hi5a V40.15-00 부터 지 원
SW396	경로이탈여부 (1: 경로 이탈, 0: 정상 경로, <0: 판단 불가)	Hi5a V40.19-00 부터 지 원
SW397	Step Forward/Backward 제한 (1 입력: 제한, 그 외 숫자 입력: 정상 동작)	
~SW399	Reserved	
SW400	각축 속도 Enable(SW110=0xKLMN, 16 진수방식)  N(B3~B0): 분해능(0x1=10 배, 0x2=100 배, 0x3=1000 배) M: 기타옵션(B4:1=절대값, B5:0=축속도, 1=모터속도) 참고) SW110 과 동일. 2 가지 옵션을 동시에 보기 원할 경우 SW110 과 함께 사용.	Hi5a V40.15-00 부터 지 원 속도단위 - 직동축: mm/sec - 회전축: deg/sec - 모터속도 : RPM
SW401~	각축(또는 모터)의 속도(16 축분) 모니터링	Hi5a V40.15-00 부터 지 원
SW420	서비스 / 모니터링 / 시스템 특성데이터 / 부하율 데이터 (1: I/Ir , 2: I/Ip, 3: 연속부하율)	
SW421~ SW436	각축(또는 모터)의 부하율(16 축분) 모니터링	단위: %
SW440	그리스 자동급유기 적용 요소품 설정	Hi5a V40.18-00 부터 지 원
SW441~ SW456	요소품별 그리스 자동급유 기준거리	Hi5a V40.18-00 부터 지 원 거리단위 - 직동축: km - 회전축: rad

번호	설 명	기 타
SW461~ SW476	요소품별 할당된 축 번호	Hi5a V40.19-00 부터 지원
SW480	Low Grease 경고 (0 : 이상없음, 1~16 : 해당 요소품 급유기 이상)	Hi5a V40.19-00 부터 지원
SW481	Empty 경고 (0 : 이상없음, 1~16 : 해당 요소품 급유기 이상)	Hi5a V40.19-00 부터 지원
SW482	Overload 경고 (0 : 이상없음, 1~16 : 해당 요소품 급유기 이상)	Hi5a V40.19-00 부터 지원
SW483	FAN 이상 정보 모니터링 - 0 번째 Bit : R-FAN 보드 정보 (0:정상, 1:이상) - 8 번째 Bit : R-FAN 보드 정보 (0:정상, 1:이상) 15 8 0 예비7 예비6 예비5 예비4 예비3 예비2 예비1 C-FAN 예비7 예비6 예비5 예비4 예비3 예비2 예비1 R-FAN	Hi5a V40.22-00 부터 지원
SW484	LCD 핸드 충돌감지 센서데이터 모니터링 - Bit 별 정보 (0:미감지, 1:감지) (Bit0 : LSR1, Bit1 : LSR2, Bit8 : RSR1, Bit9 : RSR2) 15 9 8 7 1 R-예비6 R-예비5 R-예비4 R-예비3 R-예비2 R-예비1 RSR2 RSR1 L-예비6 L-예비5 L-예비4 L-예비3 L-예비2 L-예비1 LSR2 LSR1 - 충돌감지 센서 사용 시에만 모니터링 데이터 올라옴 (경로 : F2.시스템 -> 4:응용파라미터 -> 29:Hand 충돌감지 센서 설정)	Hi5a V40.26-00 부터 지원
SW485, SW486	SW485: 에러, 경고 번호 SW486: 최상위 비트(0:에러, 1:경고) 1~15 번째 비트: 에러 부가 정보(축 번호 등)	Hi5a V40.25-00 부터 지원

4.2.3. SW40-59 릴레이 – 로봇 현재 pose 16 bit 로 읽기

릴레이 번호		설명	기타
	SW40	로봇위치 16bit 로 읽기 (SW41~SW56) 0:무 1:현재 base 좌표 값(1mm or 0.1deg) 2:지령 base 좌표 값(1mm or 0.1deg) 3:현재 축 좌표 값(0.1mm or 0.1deg) 4:지령 축 좌표 값(0.1mm or 0.1deg) 5:현재 축 좌표 값(1mm or 0.1deg) 6:지령 축 좌표 값(1mm or 0.1deg)	
SL21	SW41	X 좌표 값 or J1 축 값	
	SW42	Y 좌표 값 or J2 축 값	
SL22	SW43	Z 좌표 값 or J3 축 값	
	SW44	RX 좌표 값 or J4 축 값	
SL23	SW45	RY 좌표 값 or J5 축 값	
	SW46	RZ 좌표 값 or J6 축 값	
SL24	SW47 ~52	T1~T6 축 값	

4.2.4. SW40-59 릴레이 – 로봇 현재 pose 32 bit 로 읽기

릴레이 번호		설명	기타
	SW40	로봇위치 32bit 로 읽기 (SL21~SL29) 0:무 ... 10: 현재 base 좌표 값 + 부가 3 축 20: 지령 base 좌표 값 + 부가 3 축 1m: 현재 축 좌표 값 (m 축 ~ m+8 축) 2m: 지령 축 좌표 값 (m 축 ~ m+8 축)	(0.001mm or 0.001deg) V40.25-00 부터 지원
SL21	SW41	X 좌표 값	
	SW42	Jm+0 축 값	
SL22	SW43	Y 좌표 값	
	SW44	Jm+1 축 값	
SL23	SW45	Z 좌표 값	
	SW46	Jm+2 축 값	
SL24	SW47	RX 좌표 값	
	SW48	Jm+3 축 값	
SL25	SW49	RY 좌표 값	
	SW50	Jm+4 축 값	
SL26	SW51	RZ 좌표 값	
	SW52	Jm+5 축 값	
SL27	SW53	T1 좌표 값	
	SW54	Jm+6 축 값	
SL28	SW55	T2 좌표 값	
	SW56	Jm+7 축 값	
SL29	SW57	T3 좌표 값	
	SW58	Jm+8 축 값	

4.2.5. SW220~249 릴레이 – 시스템 변수

● 코드

10	SPDRATE ; 재생속도 (%)
20	JOGSPD ; 스텝 전/후진 시 최고속 (mm/sec)

● 읽기/쓰기

읽기	계속 갱신됨.
쓰기	계속 갱신됨.

● 예시 (예를 들어, slot1 을 이용하는 경우.)

SW220	10(dec)	SPDRATE 값 읽기. 85%
SW221	85(dec)	
SW222	-	
SW220	30020(dec)	JOGSPD=100 mm/sec
SW221	100	
SW222		

4.2.6. SW220~249 릴레이 – 메인보드 저장공간

- 코드

210	하위워드 Free Block 수, 상위워드 Total Block 수 읽기만 가능
-----	---



4.2.7. SW220~249 릴레이 – 아날로그 입출력

● 코드

301~316	AI1 ~ AI16 (단위 : 0.01V) 값 : 하위워드 읽기만 가능
351~366	AO1 ~ AO16 (단위 : 0.01V) 값 : 하위워드

● 예시 (예를 들어, slot1 을 이용하는 경우,)

SW220	302(dec)	AI2 값 읽기. -9.25V
SW221	-925(dec)	
SW222	-	
SW220	30353(dec)	AO3=3.80V
SW221	380(dec)	
SW222	-	

4.2.8. SW220~249 릴레이 – 날짜/시간

- 코드

401	날짜 : 상위워드 YYYY(dec), 하위워드 MMDD(dec)
402	시간 (24 시간제) : 상위워드 00HH(dec), 하위워드 MMSS(dec)

- 읽기/쓰기

읽기	Hi5a 제어기의 현재 날짜/시간값이 슬롯 데이터로 계속 갱신됨. (1 초에 1 번 정도의 주기)
쓰기	코드 값이 다른 값에서 30401 이나 30402 로 변경되는 순간, 슬롯 데이터로 입력해둔 날짜/시간값이 Hi5a 제어기에 설정됨.

- 예시 (예를 들어, slot1, slot2 를 이용하는 경우.)

SW220	401(dec)	Index: 날짜 읽기 (0-based)
SW221	0523(dec)	2012 년 5 월 23 일
SW222	2012(dec)	
SW223	402(dec)	Index: 시간 읽기 (0-based)
SW224	3650(dec)	15 시 36 분 50 초
SW225	15(dec)	
SW220	30401(dec)	Index: 날짜 설정 (30000-based)
SW221	0523(dec)	2012 년 5 월 23 일
SW222	2012(dec)	
SW223	30402(dec)	Index: 시간 설정 (30000-based)
SW224	3650(dec)	15 시 36 분 50 초
SW225	15(dec)	

4.2.9. SW220~249 릴레이 - GE 변수

● 코드

1101~1128	GE1 ~ GE128 GE 변수의 index 는 코드 - 1000
-----------	---

● 읽기/쓰기

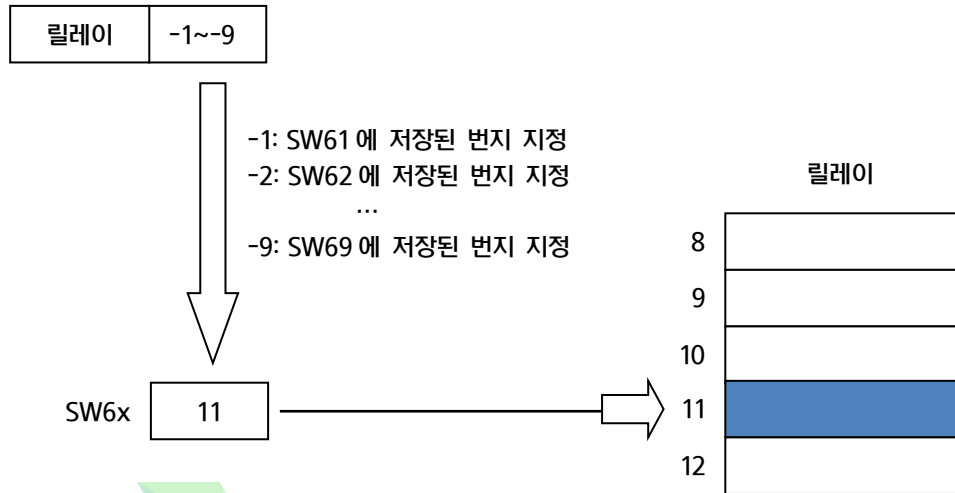
읽기	계속 갱신됨. GE[index] ~ GE[index+3]의 4byte 를 하위워드, 상위워드로 읽음.
쓰기	시스템변수 인덱스 값이 변하는 순간 (31001~31128 이어야 함.) 데이터 하위워드에 입력해둔 값을 바이트로 형변환 해서 GE 변수에 설정됨.

● 예시 (예를 들어, slot1, slot2 를 이용하는 경우,)

SW220	31001(dec)	Index: GE 설정 (30000-based)
SW221	2(hex)	GE1 에 2 설정
SW222	-	
SW223	31004(dec)	
SW224	37(hex)	GE4 에 55 설정
SW225	-	
SW220	1001(dec)	Index: GE1~ 읽기 (0-based)
SW221	0002(hex)	
SW222	3700(hex)	
SW223	1002(dec)	Index: GE2~ 읽기 (0-based)
SW224	0000(hex)	
SW225	0037(hex)	

4.3. 릴레이 간접 주소 지정

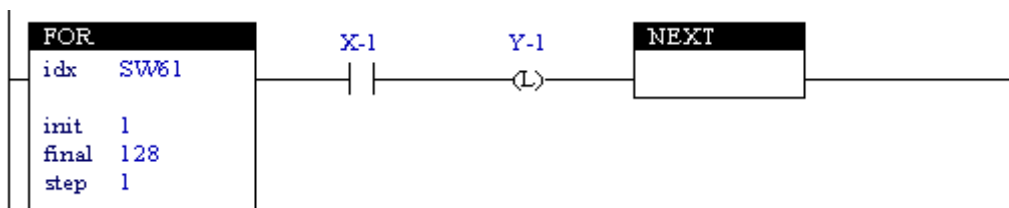
SW61~SW69 는 간접 주소 지정을 위한 시스템 메모리입니다. 릴레이 형식에 관계없이 릴레이 번지를 “-1” ~ “-9” 의 값으로 지정하면 SW61~SW69 에 저장된 값의 번지로 지정됩니다.



예를 들어 SW61=12, SW65=34, SW69=56 일 경우 아래와 같이 릴레이가 지정됩니다.

- MW-1 → MW12, MW-5 → MW34, MW-9 → MW56
- FN-1.X3 → FN12.X3, FN-5.X3 → FN34.X3, FN-9.X3 → FN56.X3
- X-1 → X12, X-5 → X34, X-9 → X56
- Y-1 → Y12, Y-5 → Y34, Y-9 → Y56

아래 내장 PLC 예제는 X1~X128 까지의 입력 신호에 해당하는 Y1~128 신호를 출력하는 동작을 FOR/NEXT 문과 간접 주소 지정방식을 활용해 작성한 예입니다.



4.4. 타이머 & 카운터 릴레이

(1) 타이머와 카운터 릴레이는 전부 down-counting 만을 지원합니다.

- 타이머 베이스는 10msec 단위로 사용자가 설정할 수 있습니다.
- 타이머 값은 내부적으로 32bit 의 값으로 처리되기 때문에 2,147,483,647[msec](약 597 시간)까지 counting 할 수 있습니다.

(2) 타이머 / 카운터의 값은 다음과 같은 의미가 있습니다.

타이머 & 카운터 값	설 명
0	접점 On (=counting 완료)
-1	접점 Off
그 외	접점 Off; timing & counting (진행중)

(3) 타이머 / 카운터 릴레이가 연결된 Rung 이 비활성이면,

- TON: TL(타이머)의 값을 -1로 됩니다.
- CTD: CL(카운터)의 값은 계속 유지합니다.

(4) 타이머 / 카운터 릴레이가 연결된 Rung 이 활성화되어 있는 동안,

- TON
만일 TL의 값이 0보다 작으면, TL의 초기값은 “타이머 베이스 x 프리셋 x 10”이 저장되고, TL 값이 0보다 크면, 5msec 마다 5씩 감소합니다.
- CTD
만일 CL 값이 0보다 작으면, CL의 초기값은 프리셋 값이 되고, 만일 CL 값이 0보다 크면, 비활성에서 활성으로 변할 때마다 1씩 감소합니다.





HD

HYUNDAI
ROBOTICS

5

명령어 사양



5. 명령어 사양

내장 PLC

5.1. 명령어 일람

표 5-1 명령어 표

명령어				설명
번호	니모닉	이름	심볼	
	RUNG	Rung		링(rung)을 나타냄
	BST	Branch Start		브랜치(branch)의 시작
	BND	Branch End		브랜치(branch)의 끝
	NXB	Nested Branch		브랜치(branch)의 중복
로직 검사 명령				검사결과, 참이면 Rung 활성화/(거짓:비활성)
1	XIC	Examine if Closed		접점이 닫혔는가를 검사(A 접점)
2	XIO	Examine if Open		접점이 열렸는가를 검사(B 접점)
3	INV	Inverting		Rung 의 결과 반전(inverting)
4	EQU	Equal		같은지(=) 검사
5	NEQ	Not Equal		다른지(<>) 검사
6	LES	Less Than		작은지(<) 검사
7	GRT	Greater Than		큰지(>) 검사
8	LEQ	Less Than or Equal		작거나 같은지(<=) 검사
9	GEQ	Greater Than or Equal		크거나 같은지(>=) 검사
출력 명령				
10	OTE	Output Energize		Rung 의 상태를 출력(활성:ON/비활성:OFF)
11	OTL	Output Latch		Rung 이 활성화면, ON(high)으로 출력
12	OTU	Output Unlatch		Rung 이 활성화면, OFF(low)로 출력

5. 명령어 사양

명령어				설명
번호	니모닉	이름	심볼	
13	OSR	One Shot Rising	-(OSR)-	Rung 이 활성화면, 한 scan 동안만 ON 출력
14	RES	Reset	-(RES)-	Rung 이 활성화면, 타이머나 카운터를 리셋
타이머 및 카운터 명령				
15	TON	Time On Delay	-  -	Rung 이 활성화인 동안 타이머 동작
16	CTD	Count Down	-  -	Rung 의 활성화(비활성→활성)을 다운-카운트
산술연산 명령				
17	ADD	Add	-  -	Rung 이 활성화면, (+)연산
18	SUB	Subtract	-  -	Rung 이 활성화면, (-)연산
19	MUL	Multiply	-  -	Rung 이 활성화면, (x)연산
20	DIV	Divide	-  -	Rung 이 활성화면, (/)연산
21	POW	Power	-  -	Rung 이 활성화면, (^: 거듭제곱)연산
데이터 변환 명령				
22	TOD	Convert int. to BCD	-  -	Rung 이 활성화면, BCD 로 변환
23	FRD	Convert form BCD to int.	-  -	Rung 이 활성화면, integer 로 변환.
24	SEG	7'Segment	-  -	Rung 이 활성화면, 7'세그먼트 값으로 변환
이동 및 복사 명령				
25	MOV	Move	-  -	Rung 이 활성화면, 데이터 한 개를 복사
26	COP	Copy data	-  -	Rung 이 활성화면, 데이터 여러 개를 복사
27	CCOP	Conditional Copy data	-  -	Rung 상태에 따라 데이터 여러 개를 복사

명령어				설명
번호	니모닉	이름	심볼	
응용 명령				
28	ROT	Rotating Output		Rung 이 활성화면, 순차적으로 출력
블록제어 명령				
29	FOR	For		Rung 이 활성화면, NEXT 까지 반복 실행
30	NEXT	Next		반복횟수 이내이면, FOR 문으로 JUMP
31	LBL	Label		Jump 명령으로 Jump 할 위치 지정
32	JMP	Jump		Rung 이 활성화면, Label 위치로 Jump (Label<0 이면, -n 개의 Next 까지 건너뛴)
33	CALL	Call		Rung 이 활성화면, Sub-ladder 호출
34	END	End		Rung 이 활성화면, Sub-ladder end

5.2. 사용 가능한 Operands

표 5-2 사용 가능한 Operands

inst	Relay	Input		Output		Timer		Counter		Memory		Const (32bit) (L/F)
	Type	X,DO		Y,DI,R,K,SP		T		C		M,S		
	arg	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	
XIC			x		x		x		x		x	x
XIO			x		x		x		x		x	x
EQU	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
LES	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
GRT	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
LEQ	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
GEQ	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
OTF		x	x		x	x	x	x	x	x	x	x
OTL		x	x		x	x	x	x	x	x	x	x
OTU		x	x		x	x	x	x	x	x	x	x
OSR		x	x		x	x	x	x	x	x	x	x
RES		x	x	x	x		x		x	x	x	x
TON	tm	x	x	x	x		x	x	x	x	x	x
	bs	x		x		x		x		x		
	pst	x		x		x		x		x		
CTD	cnt	x	x	x	x	x	x	x	x	x	x	x
	pst	x		x		x		x		x		
ADD	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
SUB	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
MUL	sA	x		x		x		x		x		
	sB	x		x		x		x		x		

inst	Relay	Input		Output		Timer		Counter		Memory		Const (32bit) (L/F)
	Type	X,DO		Y,DI,R,K,SP		T		C		M,S		
	arg	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	
	dst	x	x	x		x		x		x		x
DIV	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
POW	sA	x		x		x		x		x		
	sB	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
TOD	src	x	u	x	u	x	u	x	u	x	u	u
	dst	x	x	x	u	x	x	x	x	x	u	x
FRD	src	x	u	x	u	x	u	x	u	x	u	u
	dst	x	x	x	u	x	x	x	x	x	u	x
SEG	src	x	u	x	u	x	u	x	u	x	u	u
	dst	x	x	x	u	x	x	x	x	x	u	x
MOV	src	x		x		x		x		x		
	dst	x	x	x		x		x		x		x
COP	src					x		x				
	dst	x	x			x		x				x
	len	x		x		x		x		x		
CCOP	sA					x		x				
	sB					x		x				
	dst	x	x			x		x				x
	len	x		x		x		x		x		
ROT	st	x		x		x		x		x		x
	cnt	x		x	x	x		x		x		
	tm	x	x	x	x		x	x	x	x	x	x
	rep	x		x		x		x		x		
	out	x	x	x		x	x	x	x	x		x
	rst		x		x		x		x	x	x	x
	tmp	x	x	x		x	x	x	x	x		x
FOR	idx	x	x	x		x	x	x	x	x		x
	init	x		x		x		x		x		
	final	x		x		x		x		x		
	step	x		x		x		x		x		

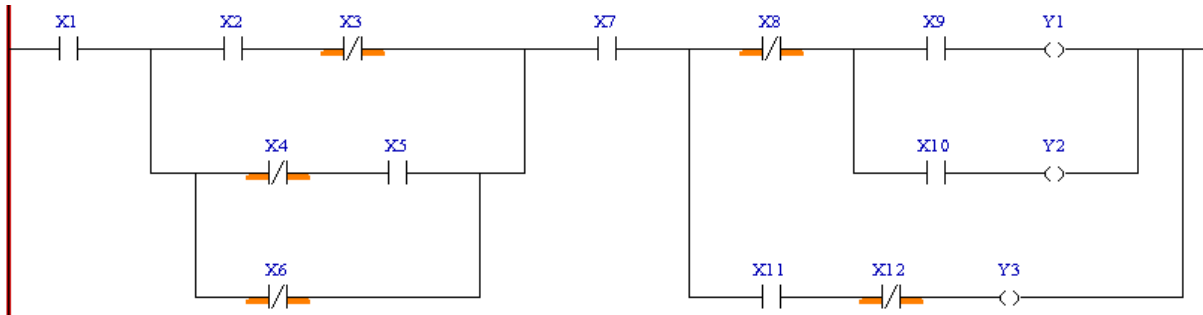
inst	Relay	Input		Output		Timer		Counter		Memory		Const (32bit) (L/F)
	Type	X,DO		Y,DI,R,K,SP		T		C		M,S		
	arg	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	n	(B,W,L,F)m	
LRI	label	x	x	x	x	x	x	x	x	x	x	
JMP	label	x		x		x		x		x		
CALL	S_file	x		x		x		x		x		

- (1) 표의 'x'표기는 사용할 수 없음을 나타냅니다.
- (2) 릴레이(relay)의 n 은 비트(bit) 번호를 나타내며, (B,W,L,F)m 은 Byte(8bit), Word(16bit), Long(32bit), Float 형식을 구분하는 첨자이고, m 은 각 형식의 데이터 번호를 의미합니다. (예, X1, XB2, XW2, XL2, XF2)
- (3) 표의 'u'표기는 부호 없는(unsigned) 값으로 처리됨을 나타냅니다.
- (4) 따라서 명령어의 인수 입력시 릴레이 형식(B, W, L, F)을 선택하는 것에 의해 데이터 사이즈(data size)를 선택할 수 있습니다.
- (5) 단 COP(복사)와 CCOP(조건별 복사) 명령의 경우, src 가 상수가 아니면, src와 dst 의 릴레이 형식은 같아야 합니다.
- (6) 비트에 대한 값을 입력할 때에는 바이트(B)나 워드(W), 혹은 롱워드(L)의 상대 값으로 입력할 수 있습니다. 즉, X34=XB5/2=XW3/2=XL2/2 와 같이 동일한 bit 번지를 여러 가지 형태로 입력할 수 있습니다.
- (7) 상수(constant)
상수의 데이터 사이즈는 32bit 이고, 데이터 형식은 2 진수, 10 진수, 16 진수가 있으며, &B 로 시작하면 2 진수를 의미하고, &H 로 시작하면 16 진수를 의미하며, 접두어 없는 숫자는 10 진수를 의미합니다. 따라서 125 는 &H7D 나 &B01111101 과 같은 형식으로 입력할 수 있습니다.
- (8) 표에 'u'가 표기가 없는 전 릴레이의 (B, W, L, F)는 부호 있는(signed) 값으로 처리합니다.
예를 들어, MOV 와 같은 명령에서 B(8bit)값을 W(16bit)에 옮기는 경우에는 부호비트(signed bit)가 확장되기 때문에 주의해야 합니다. 즉, RB(-1=&HFF)를 RW 에 옮기면 &H00FF(255)가 아닌 0xFFFF(-1)이 됩니다.
다만, BCD 데이터에 관련된 TOD 와 FRD 명령, 그리고 7'세그먼트 데이터로 변환하는 SEG 명령은 부호 없는 값(unsigned)으로 처리합니다.

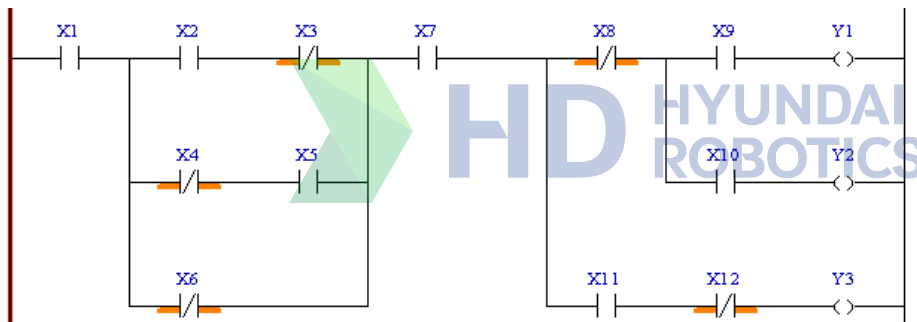
5.3. 래더 Diagram 과 니모닉의 관계

래더로 작성된 파일은 니모닉으로 자동 변환되어 제어기로 송신합니다.

(1) 래더 diagram



혹은 HRLadder 의 브랜치 간격(branch gap) [] 을 클릭하여 다음과 같이 표시할 수 도 있습니다.



(2) 니모닉(mnemonic)

```
SOR XIC X1 BST XIC X2 XIO X3 NXB BST XIO X4 XIC X5 NXB XIO X6 BND BND
XIC X7 BST XIO X8 BST XIC X9 OTE Y1 NXB XIC X10 OTE Y2 BND
NXB XIC X11 XIO X12 OTE Y3 BND
```



HD

HYUNDAI
ROBOTICS

6

명령어 설명



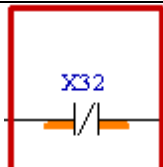
6. 명령어 설명

내장 PLC

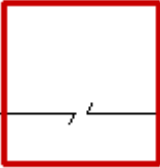
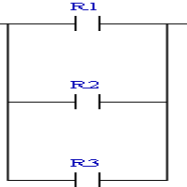
6.1. XIC(Examine if Closed) : Close 검사

심볼	사용할 수 있는 Relay 조합																																
X34	<table><tr><td></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td></td><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>arg.</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>XIC</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	XIC		X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																										
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																										
XIC		X	X	X	X	X	X																										
설명	인수의 값이 1 이면 Rung 을 활성(active), 0 이면 비활성 합니다.																																
사용 예	XIC(X2); XIC(DO2); XIC(Y2); XIC(DI2); XIC(R2); XIC(K2); XIC(SP1); XIC(T2); XIC(C2);																																
입력 방법	X18 을 XB3/2 이나 XW2/2, 혹은 XL1/2 로 입력할 수도 있습니다.																																

6.2. XIO(Examine if Open) : Open 검사

심볼	사용할 수 있는 Relay 조합																												
X32 	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(L/F)</td></tr><tr><th>inst</th><td>arg.</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td></tr><tr><td>XIO</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	XIO		X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const (32bit)																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																							
inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m																							
XIO		X	X	X	X	X																							
설명	인수의 값이 0 이면 Rung 을 활성화(active), 1 이면 비활성 합니다.																												
사용 예	XIO(X2); XIO(DO2); XIO(Y2); XIO(DI2); XIO(R2); XIO(K2); XIO(SP1); XIO(T2); XIO(C2);																												
입력 방법	X18 을 XB3/2 이나 XW2/2, 혹은 XL1/2 로 입력할 수도 있습니다.																												

6.3. INV(Inverting) : 인버팅

심볼	설명
	Rung 의 이전까지의 결과를 반전(활성$\leftrightarrow$비활성)시킵니다. 이 처리는 드모르강의 법칙에 따라 $\neg(A \times B) = \neg A + \neg B$, 또는 $\neg(A + B) = \neg A \times \neg B$로 되기 때문에, 여러 개의 브랜치로 OR 로직을 구성해야 하는 것을 브랜치 없이, 간단하게 AND 로직으로 구성할 수 있습니다. 즉, $(R1 + R2 + R3) = \neg(\neg R1 \times \neg R2 \times \neg R3)$이므로 하기 두 로직은 같은 결과가 됩니다.  

6.4. EQU(Equal) : 같은지 검사

심볼	EQU source aXB3 0 source b100																																			
Relay 조합	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td>EQU</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	EQU	sA	X	X	X	X			sB	X	X	X	X	
relay	Input	Output	Timer	Counter	Memory	const																														
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																														
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																														
EQU	sA	X	X	X	X																															
	sB	X	X	X	X																															
인수	sA: 비교하려는 첫 번째 릴레이(혹은 숫자) sB: 비교하려는 두 번째 릴레이(혹은 숫자)																																			
설명	두 값을 비교하여 같으면, Rung 이 활성화 됩니다. (점점활성)																																			
사용 예	EQU(XB3,100): XB3 에 있는 값이 100 이면 Rung 이 활성화 됩니다. EQU(XB3,RW4): XB3 과 RW4 에 있는 값이 같으면 Rung 이 활성화 됩니다.																																			

6.5. NEQ(Not Equal) : 다른지 검사

심볼	NEQ source aXB3 0 source b100																																							
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th></th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td>NEQ</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	inst	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)		arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	NEQ	sA	X	X	X	X	X			sB	X	X	X	X	X	
inst	relay		Input	Output	Timer	Counter	Memory	const																																
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																	
	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																	
NEQ	sA	X	X	X	X	X																																		
	sB	X	X	X	X	X																																		
인수	sA: 비교하려는 첫 번째 릴레이(혹은 숫자) sB: 비교하려는 두 번째 릴레이(혹은 숫자)																																							
설명	두 값을 비교하여 같지 않으면, Rung 이 활성화 됩니다. (점점활성)																																							
사용 예	NEQ(XB3,100): XB3 의 값이 100 이 아니면 Rung 이 활성화 됩니다. NEQ(XB3,RW4): XB3 의 값이 RW4 의 값과 같지 않으면 Rung 이 활성화 됩니다.																																							

6.6. LES(Less Than) : 작는지 검사

심볼	LES source aXB3 0 source b100																																					
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td></tr><tr><th>arg.</th><td></td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td></td></tr><tr><td rowspan="2">LES</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	inst	relay	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	arg.		n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m		LES	sA	X	X	X	X	X		sB	X	X	X	X	X	
inst	relay		Input	Output	Timer	Counter	Memory	const (32bit) (L/F)																														
	type	X, DO	Y, DI, R, K, SP	T	C	M, S																																
arg.		n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m																																
LES	sA	X	X	X	X	X																																
	sB	X	X	X	X	X																																
인수	sA: 비교하려는 첫 번째 릴레이(혹은 숫자) sB: 비교하려는 두 번째 릴레이(혹은 숫자)																																					
설명	sA 가 sB 보다 작으면, Rung 이 활성화 됩니다. (점점활성)																																					
사용 예	LES(XB3,100): XB3 의 값이 100 보다 작으면 Rung 이 활성화 됩니다. LES(XB3,RW4): XB3 의 값이 RW4 의 값보다 작으면 Rung 이 활성화 됩니다.																																					

6.7. GRT(Greater Than) : 큰지 검사

심볼	GRT source aXB3 0 source b100																																					
Relay 조합	<table><tr><th rowspan="2">inst</th><th rowspan="2">relay type</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>(L/F)</th></tr><tr><td rowspan="2">GRT</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	inst	relay type	Input		Output		Timer	Counter	Memory	const (32bit)	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	GRT	sA	X	X	X	X	X	X	X		sB	X	X	X	X	X	X	X	
inst	relay type			Input		Output		Timer	Counter	Memory	const (32bit)																											
		arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																													
GRT	sA	X	X	X	X	X	X	X																														
	sB	X	X	X	X	X	X	X																														
인수	sA: 비교하려는 첫 번째 릴레이(혹은 숫자) sB: 비교하려는 두 번째 릴레이(혹은 숫자)																																					
설명	sA 가 sB 보다 크면, Rung 이 활성화 됩니다. (점점활성)																																					
사용 예	GRT(XB3,100): XB3 의 값이 100 보다 크면 Rung 이 활성화 됩니다. GRT(XB3,RW4): XB3 의 값이 RW4 의 값보다 크면 Rung 이 활성화 됩니다.																																					

6.8. LEQ(Less Than or Equal) : 작거나 같은지 검사

심볼	LEQ source aXB3 0 source b100																																						
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>arg.</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>(L/F)</th></tr><tr><td rowspan="2">LEQ</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	inst	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	arg.	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	(L/F)	LEQ	sA	X	X	X	X	X		sB	X	X	X	X	X	
inst	relay		Input	Output	Timer	Counter	Memory	const																															
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																
arg.	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	(L/F)																																
LEQ	sA	X	X	X	X	X																																	
	sB	X	X	X	X	X																																	
인수	sA: 비교하려는 첫 번째 릴레이(혹은 숫자) sB: 비교하려는 두 번째 릴레이(혹은 숫자)																																						
설명	sA 가 sB 보다 작거나 같으면, Rung 이 활성화 됩니다. (점점활성)																																						
사용 예	LEQ(XB3,100): XB3 의 값이 100 보다 작거나 같으면 Rung 이 활성화 됩니다. LEQ(XB3,RW4): XB3 의 값이 RW4 의 값보다 작거나 같으면 Rung 이 활성화 됩니다.																																						

6.9. GEQ(Greater Than or Equal) : 크거나 같은지 검사

심볼	GEQ source aXB3 0 source b100																																			
Relay 조합	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><th>arg.</th><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>GEQ</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td></td><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	GEQ	sA	X	X	X	X			sB	X	X	X	X	
relay	Input	Output	Timer	Counter	Memory	const																														
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																														
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																														
GEQ	sA	X	X	X	X																															
	sB	X	X	X	X																															
인수	sA: 비교하려는 첫 번째 릴레이(혹은 숫자) sB: 비교하려는 두 번째 릴레이(혹은 숫자)																																			
설명	sA 가 sB 보다 크거나 같으면, Rung 이 활성화 됩니다. (점점활성)																																			
사용 예	GEQ(XB3,100): XB3 의 값이 100 보다 크거나 같으면 Rung 이 활성화 됩니다. GEQ(XB3,RW4): XB3 의 값이 RW4 의 값보다 크거나 같으면 Rung 이 활성화 됩니다.																																			

6.10. OTE(Output Energize) : 일반출력

심볼	사용할 수 있는 Relay 조합																																
Y23	<table><tr><td></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td></td><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>arg.</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>OTE</td><td></td><td>X X</td><td>X</td><td>X X</td><td>X X</td><td>X X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	OTE		X X	X	X X	X X	X X	X
	relay	Input	Output	Timer	Counter	Memory	const																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																										
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																										
OTE		X X	X	X X	X X	X X	X																										
설명	Rung 상태에 따라 출력신호를 출력합니다. 즉, Rung 이 활성화되면 출력신호를 On(high)으로, 비활성이면 Off(low)로 출력합니다.																																
인수 예	Y2; DI2; R2; K2; SP1;																																
입력 방법	Y18 을 YB3/2 이나 YW2/2, 혹은 YL1/2 로 입력할 수도 있습니다.																																

6.11. OTL(Output Latch) : 래치 출력

심볼	사용할 수 있는 Relay 조합																												
Y20 (L)	<table><tr><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>arg.</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>OTL</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)	OTL	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																							
inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	(L/F)																							
OTL	X	X	X	X	X	X																							
설명	Rung 이 활성화면, 출력신호를 On(high)으로 출력합니다. 그러나 Rung 이 비활성이면, 출력은 변하지 않습니다.																												
인수 예	Y2; DI2; R2; K2; SP1;																												
입력 방법	Y18 을 YB3/2 이나 YW2/2, 혹은 YL1/2 로 입력할 수도 있습니다.																												

6.12. OTU(Output Unlatch) : 언래치 출력

심볼	사용할 수 있는 Relay 조합																																			
Y21	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(L/F)</td></tr><tr><th>inst</th><td>n</td><td>n</td><td>n</td><td>n</td><td>n</td><td>(L/F)</td></tr><tr><th>arg.</th><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>OTU</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	n	n	n	n	n	(L/F)	arg.	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(L/F)	OTU	X	X	X	X	X	X
relay	Input	Output	Timer	Counter	Memory	const (32bit)																														
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																														
inst	n	n	n	n	n	(L/F)																														
arg.	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(L/F)																														
OTU	X	X	X	X	X	X																														
설명	Rung 이 활성화면, 출력신호를 Off(low)로 출력합니다. 그러나 Rung 이 비활성이면, 출력은 변하지 않습니다.																																			
사용 예	Y2; DI2; R2; K2; SP1;																																			
입력 방법	Y18 을 YB3/2 이나 YW2/2, 혹은 YL1/2 로 입력할 수도 있습니다.																																			

6.13. OSR(One Shot Rising) : 원 샷 출력

심볼	사용할 수 있는 Relay 조합																																
Y22 (OSR)	<table><tr><td rowspan="3"></td><td>relay</td><td colspan="2">Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td rowspan="3">const (32bit) (L/F)</td></tr><tr><td>type</td><td colspan="2">X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td></tr><tr><td>inst</td><td>arg.</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td><td>n (B, W, L, F) m</td></tr><tr><td>OSR</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>		relay	Input		Output	Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO		Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	OSR		x	x	x	x	x	x	x
	relay		Input		Output	Timer	Counter	Memory	const (32bit) (L/F)																								
	type		X, DO		Y, DI, R, K, SP	T	C	M, S																									
	inst	arg.	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m	n (B, W, L, F) m																										
OSR		x	x	x	x	x	x	x																									
설명	Rung 이 활성화되면, 한 scan 동안만 On(high)으로 출력합니다. 즉, Rung 이 비활성이었다가 활성화로 될 때 한 scan 동안만 해당 릴레이가 On 됩니다.																																
인수 예	Y2; DI2; R2; K2; SP1;																																
입력 방법	Y18 을 YB3/2 이나 YW2/2, 혹은 YL1/2 로 입력할 수도 있습니다.																																

6.14. RES(Reset) : 리셋

심볼	사용할 수 있는 Relay 조합																																			
T3 (RES)	<table><tr><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>n</td><td>n</td><td>n</td><td>n</td><td>n</td><td>(L/F)</td></tr><tr><td>arg.</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td>(B, W, L, F) m</td><td></td></tr><tr><td>RES</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	n	n	n	n	n	(L/F)	arg.	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m		RES	x	x	x	x	x	x
relay	Input	Output	Timer	Counter	Memory	const																														
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																														
inst	n	n	n	n	n	(L/F)																														
arg.	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m	(B, W, L, F) m																															
RES	x	x	x	x	x	x																														
설명	Rung 이 활성화면, 출력을 클리어(-1)시킵니다.																																			
인수 예	T2; C2;																																			

6.15. TON(Time On Delay) : 타이머

심볼	TON timerT2 =0 TL 2=-1 timer base(1/100s)100 preset2																																																
Relay 조합	<table><tr><th rowspan="2">relay type</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th></th></tr><tr><td rowspan="3">TON</td><td>tm</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>bs</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>pst</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>	relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m		TON	tm	X	X	X	X	X	X	X	bs	X		X	X	X	X		pst	X		X	X	X	X	
relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																									
	X, DO	Y, DI, R, K, SP	T	C	M, S																																												
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m																																										
TON	tm	X	X	X	X	X	X	X																																									
	bs	X		X	X	X	X																																										
	pst	X		X	X	X	X																																										
인수	tm: 타이머 릴레이 번호를 지정합니다. bs: 타이머의 단위(100 이면 1 초 단위, 10 이면 0.1 초 단위, ……)를 릴레이나 숫자로 설정합니다. pst: 프리셋 값(시간[msec]=bs*pst*10)을 릴레이나 숫자로 설정합니다.																																																
설명	Rung 이 활성인 시간을 계산하여 설정한 시간(bs x pst x 10) [msec]이후에 해당 타이머 릴레이가 On(high)으로 됩니다. 그러나 Rung 이 비활성이면, 즉시 클리어(-1) 설정됩니다. 참고) TL 의 값은 1msec 단위입니다.																																																
사용 예	TON(T2,100,5): T2 를 (100x5x10[msec]=5[sec])초 타이머로 설정, TON(T3,RW3,RW4): T3 를 (RW3)x(RW4)x10 의 값 만큼 타이머로 설정																																																

6.16. CTD(Count Down) : 카운터

심볼	CTD counter C2=1 CL2=0 preset 100																																							
Relay 조합	<table><tr><th rowspan="2">relay type</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th></tr><tr><td rowspan="2">CTD</td><td>cnt</td><td>X X</td><td>X X</td><td>X X</td><td>X X</td><td>X X</td><td>X X</td><td>X X</td></tr><tr><td>pst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	CTD	cnt	X X	X X	X X	X X	X X	X X	X X	pst	X	X	X	X	X	X	X
relay type	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																
	X, DO	Y, DI, R, K, SP	T	C	M, S																																			
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m																																	
CTD	cnt	X X	X X	X X	X X	X X	X X	X X																																
	pst	X	X	X	X	X	X	X																																
인수	cnt: 카운터 릴레이 번호를 지정합니다. pst: 프리셋 값을 릴레이나 숫자로 설정합니다.																																							
설명	Rung의 상승(비활성에서 활성)을 다운-카운트합니다 해당 CL의 값이 0이면 해당 카운터가 On(high)으로 되고, 더 이상 카운트 하지 않습니다. Rung이 활성일 때, 해당 CL의 값이 음수이면, pst 값을 CL에 저장합니다. 참고) Rung이 비활성이라도 CL는 클리어(-1) 하지 않습니다.																																							
사용 예	CTD(C2,100): C2를 (100)개 다운-카운터로 설정 CTD(C3,RW3): C3를 (RW3)의 값 만큼 다운-카운터로 설정																																							

6.17. ADD(Add) : 더하기

심볼	ADD source aXB3 0 source b50 destinationRW3 0																																																																				
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><th colspan="2">X, DO</th><th colspan="2">Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>arg.</th><th>n</th><th>(B,W,L,F)</th><th>m</th><th>n</th><th>(B,W,L,F)</th><th>m</th><th>n</th><th>(B,W,L,F)</th><th>m</th><th>n</th><th>(B,W,L,F)</th><th>m</th></tr><tr><td rowspan="3">ADD</td><td>sA</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td></tr></table>	inst	relay	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO		Y, DI, R, K, SP		T	C	M, S	arg.	n	(B,W,L,F)	m	n	(B,W,L,F)	m	n	(B,W,L,F)	m	n	(B,W,L,F)	m	ADD	sA	X		X		X		X		X			sB	X		X		X		X		X			dst	X	X	X		X		X		X		X
inst	relay		Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																																											
	type	X, DO		Y, DI, R, K, SP		T	C	M, S																																																													
arg.	n	(B,W,L,F)	m	n	(B,W,L,F)	m	n	(B,W,L,F)	m	n	(B,W,L,F)	m																																																									
ADD	sA	X		X		X		X		X																																																											
	sB	X		X		X		X		X																																																											
	dst	X	X	X		X		X		X		X																																																									
인수	sA: 연산을 위한 첫 번째 릴레이(혹은 숫자) sB: 연산을 위한 두 번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																																				
설명	Rung 이 활성이면, sA 값에 sB 의 값을 더하여 dst 에 저장합니다. 만일, 연산결과에 오버플로우가 발생하면 SP8=1 로 설정합니다.																																																																				
사용 예	ADD(XB3,RB2,RW3): (RW3)=(XB3)+(RB2) ADD(XB3,RB2,RB2): (RB2)=(XB3)+(RB2) ADD(XB3,50,RW3): (RW3)=(XB3)+50																																																																				

6.18. SUB(Subtract) : 빼기

심볼	SUB source aXB3 0 source b50 destinationRW3 0																																																			
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="2">Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><th colspan="2">X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>arg.</th><th></th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>(L/F)</th></tr><tr><td rowspan="3">SUB</td><td>sA</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	inst	relay	Input		Output	Timer	Counter	Memory	const	type	X, DO		Y, DI, R, K, SP	T	C	M, S	(32bit)	arg.		n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	(L/F)	SUB	sA	X		X	X	X	X		sB	X		X	X	X	X		dst	X	X	X	X	X	X	X
inst	relay		Input		Output	Timer	Counter	Memory	const																																											
	type	X, DO		Y, DI, R, K, SP	T	C	M, S	(32bit)																																												
arg.		n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	(L/F)																																												
SUB	sA	X		X	X	X	X																																													
	sB	X		X	X	X	X																																													
	dst	X	X	X	X	X	X	X																																												
인수	sA: 연산을 위한 첫 번째 릴레이(혹은 숫자) sB: 연산을 위한 두 번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																			
설명	Rung 이 활성화되면, sA 값을 sB 의 값으로 빼서 dst 에 저장합니다.																																																			
사용 예	SUB(XB3,RB2,RW3): (RW3)=(XB3)-(RB2) SUB(XB3,RB2,RB2): (RB2)=(XB3)-(RB2) SUB(XB3,50,RW3): (RW3)=(XB3)-50																																																			

6.19. MUL(Multiply): 곱하기

심볼	MUL source aXB3 0 source b50 destinationRW3 0																																																	
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="2">Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>arg.</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th><th>n</th><th>(B, W, L, F) m</th></tr><tr><td rowspan="3">MUL</td><td>sA</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	inst	relay	Input		Output	Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	arg.	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	MUL	sA	X		X	X	X	X		sB	X		X	X	X	X		dst	X	X	X	X	X	X	X
inst	relay		Input		Output	Timer	Counter	Memory	const (32bit) (L/F)																																									
	type	X, DO	Y, DI, R, K, SP	T	C	M, S																																												
arg.	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m	n	(B, W, L, F) m																																										
MUL	sA	X		X	X	X	X																																											
	sB	X		X	X	X	X																																											
	dst	X	X	X	X	X	X	X																																										
인수	sA: 연산을 위한 첫 번째 릴레이(혹은 숫자) sB: 연산을 위한 두 번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																	
설명	Rung 이 활성화면, sA 값에 sB 의 값을 곱하여 dst 에 저장합니다. 만일, 연산결과에 오버플로우가 발생하면 SP8=1 로 설정합니다.																																																	
사용 예	MUL(XB3,RB2,RW3): (RW3)=(XB3)x(RB2) MUL(XB3,RB2,RB2): (RB2)=(XB3)x(RB2) MUL(XB3,50,RW3): (RW3)=(XB3)x50																																																	

6.20. DIV(Divide): 나누기

심볼	DIV source aXB30 source b50 destinationRW30																																					
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay type</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit) (L/F)</th></tr><tr><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th></th></tr><tr><td rowspan="3">DIV</td><td>sA</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	inst	relay type	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m		DIV	sA	X	X	X	X	X		sB	X	X	X	X	X		dst	X	X	X	X	X	X
inst	relay type		Input	Output	Timer	Counter	Memory	const (32bit) (L/F)																														
	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m																																
DIV	sA	X	X	X	X	X																																
	sB	X	X	X	X	X																																
	dst	X	X	X	X	X	X																															
인수	sA: 연산을 위한 첫 번째 릴레이(혹은 숫자) sB: 연산을 위한 두 번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																					
설명	Rung 이 활성화되면, sA 값을 sB 의 값으로 나누어 dst 에 저장합니다. 만일, sB 가 0 이거나, 연산결과에 오버플로우가 발생하면 SP8=1 로 설정합니다.																																					
사용 예	DIV(XB3,RB2,RW3): (RW3)=(XB3)/(RB2) DIV(XB3,RB2,RB2): (RB2)=(XB3)/(RB2) DIV(XB3,50,RW3): (RW3)=(XB3)/50																																					

6.21. POW(Power): 거듭제곱

심볼	POW source aXB3 source b3 destinationRW3																																																								
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="2">Input</th><th colspan="2">Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>type</th><th>X, DO</th><th></th><th>Y, DI, R, K, SP</th><th></th><th>T</th><th>C</th><th>M, S</th></tr><tr><th></th><th>arg.</th><th>n (B,W,L,F)</th><th>m</th><th>n (B,W,L,F)</th><th>m</th><th>n (B,W,L,F)</th><th>m</th><th>n (B,W,L,F)</th><th>m</th></tr><tr><td rowspan="3">POW</td><td>sA</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td><td>X</td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td><td></td><td>X</td><td>X</td></tr></table>	inst	relay	Input		Output		Timer	Counter	Memory	const (32bit) (L/F)	type	X, DO		Y, DI, R, K, SP		T	C	M, S		arg.	n (B,W,L,F)	m	n (B,W,L,F)	m	n (B,W,L,F)	m	n (B,W,L,F)	m	POW	sA	X		X		X		X		sB	X		X		X		X		dst	X	X	X		X		X	X
inst	relay		Input		Output		Timer	Counter	Memory	const (32bit) (L/F)																																															
	type	X, DO		Y, DI, R, K, SP		T	C	M, S																																																	
	arg.	n (B,W,L,F)	m	n (B,W,L,F)	m	n (B,W,L,F)	m	n (B,W,L,F)	m																																																
POW	sA	X		X		X		X																																																	
	sB	X		X		X		X																																																	
	dst	X	X	X		X		X	X																																																
인수	sA: 연산을 위한 첫 번째 릴레이(혹은 숫자) sB: 연산을 위한 두 번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																								
설명	Rung 이 활성화되면, sA 값을 sB 의 값으로 거듭제곱하여 dst 에 저장합니다. 만일, 연산결과에 오버플로우가 발생하면 SP8=1 로 설정합니다.																																																								
사용 예	POW(2,3,RW3): (RW3)=2^3=2*2*2=8 POW(XB3,RB2,RW3): (RW3)=(XB3)^(RB2) POW(XB3,RB2,RB2): (RB2)=(XB3)^(RB2) POW(XB3,3,RW3): (RW3)=(XB3)^3																																																								

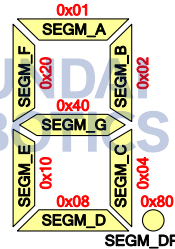
6.22. TOD(Convert to BCD): BCD 값으로 변환

심볼	TOD sourceXB30 destinationRB20																																								
Relay 조합	<table><tr><td></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td></td><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>arg.</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>TOD</td><td>src</td><td>Xu</td><td>Xu</td><td>Xu</td><td>Xu</td><td>Xu</td><td>u</td></tr><tr><td></td><td>dst</td><td>Xx</td><td>Xu</td><td>Xx</td><td>Xx</td><td>Xu</td><td>x</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	TOD	src	Xu	Xu	Xu	Xu	Xu	u		dst	Xx	Xu	Xx	Xx	Xu	x
	relay	Input	Output	Timer	Counter	Memory	const																																		
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																		
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																																		
TOD	src	Xu	Xu	Xu	Xu	Xu	u																																		
	dst	Xx	Xu	Xx	Xx	Xu	x																																		
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이																																								
설명	Rung 이 활성화되면, src 의 값을 BCD 로 변환하여 dst 에 저장합니다. 이 명령은 BCD 형식으로 7'세그먼트에 값을 표시하는 장치를 사용할 때 편리합니다. 만일, dst 가 바이트(B) 형식이면, 2 개의 디지털로 변환하고, 워드(W) 형식이면, 4 개의 디지털로 변환합니다. Src 의 값이 변환하는 디지털 수 보다 크면, SP7=1 로 설정합니다.																																								
사용 예	TOD(XB3,RB2): XB3 의 값을 BCD 로 변환하여 RB2 에 저장합니다. (참고, BCD(Binary Coded Decimal)란 4bit 의 코드 값이 0~9 범위의 값을 가질 수 있는 숫자를 의미합니다. 즉 BCD 에서는 4bit 로 표시할 수 있는 숫자 0~F 중에서 A~F 를 사용하지 않습니다) 만일 (XB3)=&H7B(123)이면, &H23(35)를 RB2 에 저장하고, &H7B(123)>&H63(99)이기 때문에 SP7=1 로 설정합니다.																																								

6.23. FRD(Convert form BCD to Integer): integer 로 변환

심볼	FRD sourceXB3 0 destinationRB3 0																																			
Relay 조합	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><td>arg.</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>FRD</td><td>src</td><td>Xu</td><td>Xu</td><td>Xu</td><td>Xu</td><td>u</td></tr><tr><td></td><td>dst</td><td>Xx</td><td>Xx</td><td>Xx</td><td>Xx</td><td>x</td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	FRD	src	Xu	Xu	Xu	Xu	u		dst	Xx	Xx	Xx	Xx	x
relay	Input	Output	Timer	Counter	Memory	const																														
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																														
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																														
FRD	src	Xu	Xu	Xu	Xu	u																														
	dst	Xx	Xx	Xx	Xx	x																														
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이																																			
설명	Rung 이 활성화면, src 의 BCD 값을 integer 로 변환하여 dst 에 저장합니다. 이 명령은 BCD 형태로 출력되는 캠 스위치의 값을 입력으로 받아드리는 경우에 편리하게 사용할 수 있습니다. 만일, src 의 값이 BCD 가 아닌 경우에는 SP7=1 로 설정합니다. 또한, src 가 워드(W) 형식이고, dst 가 바이트(B) 형식인 경우, 변환할 src 의 최대값은 &H9999 이므로 integer 의 변환결과가 9999(&H270F)로 되어 바이트의 범위 &Hff 를 초과하므로 오버플로우가 발생하게 되며, 이 경우에는 SP7=1 로 설정합니다.																																			
사용 예	FRD(XB3,RB2): XB3 의 값(BCD)을 integer 로 변환하여 RB2 에 저장합니다. 만일 (XB3)=&H23(35)이면, 23=&H17 를 RB2 에 저장합니다.																																			

6.24. SEG(7'Segment): 7'세그먼트 값으로 변환

심볼	SEG source XB3 0 destination RW3 0																																								
Relay 조합	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>(L/F)</th></tr><tr><th>SEG</th><th>src</th><td>X</td><td>U</td><td>X</td><td>U</td><td>X</td><td>U</td></tr><tr><th></th><th>dst</th><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	SEG	src	X	U	X	U	X	U		dst	X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																		
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																		
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																																		
SEG	src	X	U	X	U	X	U																																		
	dst	X	X	X	X	X	X																																		
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이																																								
설명	Rung 이 활성화면, src 의 값을 7'세그먼트 값(8bit)으로 변환하여 dst 에 저장합니다. dst 가 워드(W)형식이면, 2 개의 7'세그먼트 값(8bit)을 dst 에 저장합니다.																																								
7' 세그먼트 데이터	#define SEGM_A 0x01 #define SEGM_B 0x02 #define SEGM_C 0x04 #define SEGM_D 0x08 #define SEGM_E 0x10 #define SEGM_F 0x20 #define SEGM_G 0x40 #define SEGM_DP 0x80  #define SEG_D_0 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_E SEGM_F) #define SEG_D_1 (SEGM_B SEGM_C) #define SEG_D_2 (SEGM_A SEGM_B SEGM_G SEGM_E SEGM_D) #define SEG_D_3 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_G) #define SEG_D_4 (SEGM_B SEGM_C SEGM_F SEGM_G) #define SEG_D_5 (SEGM_A SEGM_C SEGM_D SEGM_F SEGM_G) #define SEG_D_6 (SEGM_A SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_7 (SEGM_A SEGM_B SEGM_C) #define SEG_D_8 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_9 (SEGM_A SEGM_B SEGM_C SEGM_F SEGM_G) #define SEG_D_A (SEGM_A SEGM_B SEGM_C SEGM_E SEGM_F SEGM_G) #define SEG_D_B (SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_C (SEGM_A SEGM_D SEGM_E SEGM_F) #define SEG_D_D (SEGM_B SEGM_C SEGM_D SEGM_E SEGM_G) #define SEG_D_E (SEGM_A SEGM_D SEGM_E SEGM_F SEGM_G) #define SEG_D_F (SEGM_A SEGM_E SEGM_F SEGM_G)																																								
사용 예	SEG(XB3,RW3): XB3 의 값에 해당하는 7'세그먼트 값을 RW3 에 저장 만일 (XB3)=(&H17)인 경우, 상기 SEG_D_1(SEGM_B SEGM_C=0x02 0x04=0x06)=&H06 과 상기 SEG_D_7(SEGM_A SEGM_B SEGM_C=0x01 0x02 0x04=0x07)=&H07 이 결합된 &H0607 를 RW3 에 저장합니다.																																								

6.25. MOV(Move): 이동

심볼	MOV source55 destinationRB2 0																																
Relay 조합	<table><tr><th rowspan="2">relay type</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th>inst</th><th>arg.</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th><th>n (B,W,L,F) m</th></tr><tr><td rowspan="2">MOV</td><td>src</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	relay type	Input	Output	Timer	Counter	Memory	const (32bit) (L/F)	X, DO	Y, DI, R, K, SP	T	C	M, S	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	MOV	src	X	X	X	X	X	dst	X	X	X	X	X
relay type	Input		Output	Timer	Counter	Memory	const (32bit) (L/F)																										
	X, DO	Y, DI, R, K, SP	T	C	M, S																												
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m																											
MOV	src	X	X	X	X	X																											
	dst	X	X	X	X	X																											
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이																																
설명	Rung 이 활성화면, src 의 값을 dst 에 복사합니다. 만일, src 가 워드(W)형식이고 dst 가 바이트(B)형식이면, src 의 값 중 하위 바이트만 dst 에 복사됩니다. 그리고, 내장 PLC 의 모든 데이터는 부호가 있는 데이터로 처리하고 있기 때문에, src 가 바이트(B) 형식이고 값이 -1(&Hff)인 경우 워드(W)형식의 dst 로 복사하면, -1(&HFFFF)로 복사합니다.(&H00ff 는 255 의 값이 됩니다.)																																
사용 예	MOV(XB3,RB2): XB3 의 값을 RB2 에 복사합니다. MOV(55,RB2): 55 를 RB2 에 복사합니다. MOV(-30,RB2): -30 을 RB2 에 복사합니다.																																

6.26. COP(Copy data): 복사

심볼	COP sourceXB20 destinationYB30 length3																																														
Relay 조합	<table><tr><td></td><td>relay</td><td>Input</td><td>Output</td><td>Timer</td><td>Counter</td><td>Memory</td><td>const</td></tr><tr><td></td><td>type</td><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><td>inst</td><td>arg.</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td rowspan="3">COP</td><td>src</td><td></td><td></td><td>X</td><td>X</td><td></td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td></tr><tr><td>len</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	COP	src			X	X			dst	X	X	X	X		X	len	X	X	X	X	X	
	relay	Input	Output	Timer	Counter	Memory	const																																								
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																								
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																																								
COP	src			X	X																																										
	dst	X	X	X	X		X																																								
	len	X	X	X	X	X																																									
인수	src: 원본(source) 릴레이 혹은 채우고자 하는 값(const) dst: 목적지(destination) 릴레이 len: 개수(length) 릴레이(혹은 숫자)																																														
설명	Rung 이 활성화되면, src 의 위치에서부터 len 수 만큼 dst 의 위치로 값들을 복사합니다. src 가 숫자인 경우, dst 에서부터 len 수 만큼의 src 값으로 채워집니다. 이 경우 dst 가 bit 형식이면, src 의 숫자가 0 이면 OFF, 0 이 아니면 ON으로 채워집니다. src 가 릴레이인 경우, src 와 dst 의 데이터 형은 같아야 합니다. 즉, src 가 비트이면 dst 도 비트, src 가 바이트(B)이면 dst 도 바이트(B), src 가 워드(W)이면 dst 도 워드(W)라야 합니다. 만일 src+len 이 src 릴레이 최대개수보다 크거나, dst+len 이 dst 릴레이 최대개수보다 큰 경우, 릴레이 최대개수까지만 복사를 진행합니다.																																														
사용 예	지정한 개수만큼 복사하는 예) COP(X2,Y3,4): Y3=X2, Y4=X3, Y5=X4, Y6=X5 COP(XB2,YB3,3): YB3=XB2, YB4=XB3, YB5=XB4 COP(XW2,YW3,2): YW3=XW2, YW4=XW3 지정한 값(const)으로 채우는 예) COP(0,Y3,4): Y3=OFF, Y4=OFF, Y5=OFF, Y6=OFF COP(1,Y3,4): Y3=ON, Y4=ON, Y5=ON, Y6=ON COP(25,Y3,4): Y3=ON, Y4=ON, Y5=ON, Y6=ON COP(25,YB3,3): YB3=25, YB4=25, YB5=25 COP(&H55AA,YW3,2): YW3=&H55AA, YW4=&H55AA COP(0,MW3,50): MW3=0, MW4=0, MW5=0, MW6=0, ~, MW52=0																																														

6.27. CCOP(Conditional Copy data): 조건 별 복사

심볼	<table><tr><th colspan="2">CCOP</th></tr><tr><td>source a</td><td>X2</td></tr><tr><td></td><td>0</td></tr><tr><td>source b</td><td>R5</td></tr><tr><td></td><td>0</td></tr><tr><td>destination</td><td>Y3</td></tr><tr><td></td><td>0</td></tr><tr><td>length</td><td>3</td></tr></table>	CCOP		source a	X2		0	source b	R5		0	destination	Y3		0	length	3																																					
CCOP																																																						
source a	X2																																																					
	0																																																					
source b	R5																																																					
	0																																																					
destination	Y3																																																					
	0																																																					
length	3																																																					
Relay 조합	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>n : (B,W,L,F) m</th><th>(L/F)</th></tr><tr><td rowspan="4">CCOP</td><td>sA</td><td></td><td></td><td>X</td><td>X</td><td></td><td></td></tr><tr><td>sB</td><td></td><td></td><td>X</td><td>X</td><td></td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td>X</td></tr><tr><td>len</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	(L/F)	CCOP	sA			X	X			sB			X	X			dst	X	X	X	X		X	len	X	X	X	X	X	
	relay	Input	Output	Timer	Counter	Memory	const																																															
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																															
inst	arg.	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	n : (B,W,L,F) m	(L/F)																																															
CCOP	sA			X	X																																																	
	sB			X	X																																																	
	dst	X	X	X	X		X																																															
	len	X	X	X	X	X																																																
인수	sA: Rung 이 활성일 때, 복사하고자 하는 원본(source) 릴레이 혹은 채우고자 하는 값(const) sB: Rung 이 비활성일 때, 복사하고자 하는 원본(source) 릴레이 혹은 채우고자 하는 값(const) dst: 목적지(destination) 릴레이 len: 개수(length) 릴레이(혹은 숫자)																																																					
설명	Rung 상태에 따라 sA 나 sB 의 위치에서부터 len 수 만큼 dst 의 위치로 값들을 복사합니다. src 가 숫자인 경우, dst 에서부터 len 수 만큼의 sA/sB 값으로 채워집니다. 이 경우 dst 가 bit 형식이면, sA/sB 의 숫자가 0 이면 OFF, 0 이 아니면 ON 으로 채워집니다. sA/sB 가 릴레이인 경우, sA/sB 와 dst 의 데이터 형은 같아야 합니다. 즉, sA/sB 가 비트이면 dst 도 비트, sA/sB 가 바이트(B)이면 dst 도 바이트(B), sA/sB 가 워드(W)이면 dst 도 워드(W)라야 합니다. 만일 (sA/sB)+len 이 sA/sB 릴레이 최대개수보다 크거나, dst+len 이 dst 릴레이 최대개수보다 큰 경우, 릴레이 최대개수까지만 복사를 진행합니다.																																																					
사용 예	지정한 개수만큼 복사하는 예) CCOP(X2,R5,Y3,3): Rung 이 활성이면, Y3=X2, Y4=X3, Y5=X4 Rung 이 비활성이면, Y3=R5, Y4=R6, Y5=R7 CCOP(1,0,Y3,3): Rung 이 활성이면, Y3=ON, Y4=ON, Y5=ON Rung 이 비활성이면, Y3=OFF, Y4=OFF, Y5=OFF 참고) 인수 사용 예는 COP 명령과 유사하므로 COP 명령을 참고하세요.																																																					

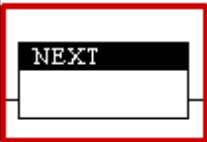
6.28. ROT(Rotating Output): 순차 출력

심볼	ROT start relay MW51 0 count 6 timer relay T5 1 repeat time(1/100s) 200 out relay MW58 0 reset relay X3 0 temp relay MW60 0																																																																																	
Relay 조합	<table><tr><th rowspan="2">inst</th><th rowspan="2">relay type</th><th colspan="2">Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th rowspan="2">const (32bit) (L/F)</th></tr><tr><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th></tr><tr><th></th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th></th></tr><tr><td rowspan="6">ROT</td><td>st</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>cnt</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>tm</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>rep</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>out</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>rst</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>tmp</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>	inst	relay type	Input		Output	Timer	Counter	Memory	const (32bit) (L/F)	X, DO	Y, DI, R, K, SP	T	C	M, S		arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m		ROT	st	X	X	X	X	X	X	X	cnt	X	X	X	X	X	X	X	tm	X	X	X	X	X	X	X	rep	X	X	X	X	X	X	X	out	X	X	X	X	X	X	X	rst	X	X	X	X	X	X	X	tmp	X	X	X	X	X	X	X	X
inst	relay type			Input		Output	Timer	Counter	Memory		const (32bit) (L/F)																																																																							
		X, DO	Y, DI, R, K, SP	T	C	M, S																																																																												
	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m																																																																											
ROT	st	X	X	X	X	X	X	X																																																																										
	cnt	X	X	X	X	X	X	X																																																																										
	tm	X	X	X	X	X	X	X																																																																										
	rep	X	X	X	X	X	X	X																																																																										
	out	X	X	X	X	X	X	X																																																																										
	rst	X	X	X	X	X	X	X																																																																										
tmp	X	X	X	X	X	X	X	X																																																																										
인수	st: 시작(start) 릴레이 cnt: 개수(count) 릴레이(혹은 숫자) tm: 타이머(timer) 릴레이(1/100sec 단위) rep: 반복(repeat) 시간을 지정하는 릴레이(혹은 숫자) out: 결과를 출력(output)할 릴레이 rst: 출력상태를 리셋(reset)할 릴레이 tmp: 연산을 위해 어디에 있는 값을 출력하고 있는지를 나타내는 릴레이																																																																																	
설명	Rung 이 활성화이면, st 릴레이부터 cnt 범위내의 릴레이 값이 0 이 아닌 값을 rep 시간 동안 out 릴레이에 반복적으로 출력합니다. 만일, rst 릴레이에 신호가 입력되면, st 릴레이부터 cnt 수 만큼 전부 0 으로 채우고, 타이머 값을 rep 값으로 초기화 하며, out 릴레이도 0 으로 출력합니다. 이 명령은 에러번호를 출력하는 장치가 한 개 밖에 없으나, 발생하는 에러의 종류가 많을 경우, 발생한 에러번호를 지정한 시간 동안 출력하는 용도에 매우 편리하게 사용할 수 있습니다.																																																																																	
사용 예	------(에러조건 1)-----[MOV(21,MW51)]--- ------(에러조건 2)-----[MOV(22,MW52)]--- ------(에러조건 3)-----[MOV(23,MW53)]--- ------(에러조건 4)-----[MOV(24,MW54)]--- ------(에러조건 5)-----[MOV(25,MW55)]--- ------(에러조건 6)-----[MOV(26,MW56)]--- -----[ROT(MW51,6,T5,200,MW58,X3,MW60)]--- -----[TOD(MW58,YB3)]----- 상기 로직은 에러조건 1~6 중에 한 개 이상의 에러가 발생하면, 에러번호가 MW51~MW56 에 저장되고, ROT 명령에 의해 발생한 에러번호가 MW58 에 2 초 동안 저장되고, TOD 명령으로 BCD 값으로 변환되어 YB3 에 연결된 표시장치에 순차적으로 표시됩니다. 만일, 외부 에러리셋에 연결된 X3 에 신호가 입력되면, 에러번호가 저장된 MW51~MW56 의 내용이 0 으로 클리어 되고, MW58 과 MW60 도 0 으로 클리어 되어, 표시장치에 0 이 표시됩니다.																																																																																	

6.29. FOR(FOR): 블록 반복

심볼	FOR idx SW61 initial 1 final 256 step 1																																																																
Relay 조합	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><th>X, DO</th><th>Y, DI, R, K, SP</th><th>T</th><th>C</th><th>M, S</th><th>(32bit)</th></tr><tr><th>inst</th><th>arg.</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>n : (B, W, L, F) m</th><th>(L/F)</th></tr><tr><td></td><td>idx</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td></td><td>init</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td></td><td>final</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td></td><td>step</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr><tr><td>FOR</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td><td>X</td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)		idx	X	X	X	X	X	X		init	X	X	X	X	X	X		final	X	X	X	X	X	X		step	X	X	X	X	X	X	FOR		X	X	X	X	X	X
	relay	Input	Output	Timer	Counter	Memory	const																																																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																																																										
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																																																										
	idx	X	X	X	X	X	X																																																										
	init	X	X	X	X	X	X																																																										
	final	X	X	X	X	X	X																																																										
	step	X	X	X	X	X	X																																																										
FOR		X	X	X	X	X	X																																																										
인수	idx: 인덱스(index) 릴레이 init: 초기값(initial value) 릴레이 혹은 숫자 final: 종료값(final value) 릴레이 혹은 숫자 step: 스텝(step)으로 사용할 릴레이 혹은 숫자																																																																
설명	Rung 이 활성화면, NEXT 까지의 블록을 init 부터 final 까지 idx 릴레이 값을 step 씩 증가하면서 반복 실행합니다. FOR 문을 실행할 때 무조건 idx 에 init 값을 대입합니다. FOR/NEXT 문은 최대 10 개까지 중첩시킬 수 있습니다. → FOR() FOR() FOR() ...NEXT NEXT NEXT step>0 인 경우, init>final 이면 실행하지 않고 NEXT 로 Jump 함. step<0 인 경우, init<final 이면 실행하지 않고 NEXT 로 Jump 함. Final 과 step 은 변수로 지정해도 for 문을 처음 시작할 때의 값만을 사용합니다. 특수하게 FOR 문의 중간에서 빠져나가고자 할 때에는 뒤에서 설명할 JMP(음수)명령을 사용할 수 있습니다. (JMP 명령설명 참조) 주의) FOR 명령은 브랜치를 위한 별도의 처리를 하지 않습니다.																																																																
사용 예	FOR(RB10,1,4,1): RB10=1 부터 4 까지 1 씩 증가하면서 NEXT 명령까지 반복 실행합니다. {[--[FOR(SW61,1,256,1)]---[XIC(X-1)]---[OTL(Y-1)]---[NEXT]--]}: SW61=1 부터 256 까지 1 씩 증가하면서 {XIC(X-1), OTL(Y-1)}명령을 반복 실행 합니다. 즉, idx 가 상대어드레싱(SW61~SW69)을 위한 릴레이를 사용하고 있고, 그 XIC 의 X 릴레이와 OTL 의 Y 릴레이가 “-1”이므로 SW61 의 값에 있는 번호가 적용되기 때문에, X1~X256 중 High 인 신호번호에 해당하는 Y 릴레이 번호만 High 로 출력하고 입력되지 않는 번호의 Y 출력은 이전상태를 유지합니다. 참고) 상대어드레싱이란 어떤 형식의 릴레이든지 그 번지를 -1~-9 값으로 설정하면 SW61~SW69 에 저장된 값으로 릴레이 번지가 지정되는 방식을 말합니다.																																																																

6.30. NEXT(NEXT): 블록 next

심볼	
인수	없음
설명	FOR 명령의 step 에 따라 다음과 같이 동작합니다. step>0 인 경우, idx 릴레이 값이 final 보다 적거나 같을 때까지 반복 실행하고, step<0 인 경우, idx 릴레이 값이 final 보다 크거나 같을 때까지 반복 실행합니다. FOR 문이 없이 NEXT 문이 실행되는 경우, NEXT 명령이 무시됩니다. 주의) FOR/NEXT 명령은 브랜치를 위한 별도의 처리를 하지 않고 있으므로, 브랜치 내에 FOR 명령을 기록하고, 브랜치 밖에 혹은 다른 브랜치 내에 NEXT 를 기록하면 FOR 명령이 잘못 동작합니다.
사용 예	FOR 명령의 사용 예를 참고하십시오.

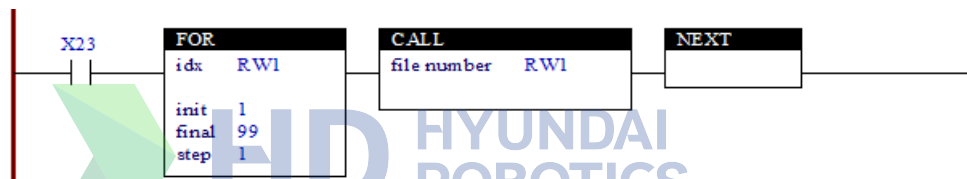
6.31. LBL(Label): 레이블 지정

심볼	LBL label 99																																
Relay 조합	<table><tr><th></th><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th></th><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><th>arg.</th><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>LBL</td><td>label</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td></td></tr></table>		relay	Input	Output	Timer	Counter	Memory	const		type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	LBL	label	x	x	x	x	x	
	relay	Input	Output	Timer	Counter	Memory	const																										
	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																										
inst	arg.	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																										
LBL	label	x	x	x	x	x																											
인수	label: 레이블 번호, 0 보다 큰 숫자(const)로 번호를 지정합니다.																																
설명	JMP 명령으로 jump 할 레이블의 위치를 지정합니다. LBL 명령은 Rung 이 활성/비활성에 무관하게 위치가 지정됩니다. 한 file 내에 동일한 번호의 LBL 명령이 있을 경우,																																
사용 예	JMP 명령과 함께 사용하므로 JMP 명령 설명을 참고하십시오.																																

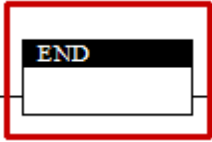
6.32. JMP(Jump): 점프

심볼	JMP label 99																																		
Relay 조합	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(32bit)</td></tr><tr><th>inst</th><th>arg.</th><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>n : (B, W, L, F) m</td><td>(L/F)</td></tr><tr><td>JMP</td><td>label</td><td>x</td><td>x</td><td>x</td><td>x</td><td></td></tr></table>							relay	Input	Output	Timer	Counter	Memory	const	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)	inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)	JMP	label	x	x	x	x	
relay	Input	Output	Timer	Counter	Memory	const																													
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(32bit)																													
inst	arg.	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	n : (B, W, L, F) m	(L/F)																													
JMP	label	x	x	x	x																														
인수	label: 레이블 번호인 숫자(const)나 릴레이																																		
설명	Rung 이 활성화되면, label 로 지정한 레이블의 값과 일치하는 LBL 명령이 있는 위치로 점프합니다. 특별하게 label<0 인 경우에는 FOR 문의 중간에서 빠져나가는 기능으로 사용할 수 있습니다. (음수로 지정한 수만큼 skip 함) 주의 1) 레이블의 위치가 JMP 명령보다 위쪽에 있고, JMP 명령 앞에 조건이 없을 경우에는 무한 루프에 빠질 수 있으므로 주의하십시오. 이 경우, 스캔타임이 5 초를 초과하기 때문에 SP17=On 으로 설정됩니다. 주의 2) FOR/NEXT 명령 블록 내에서 JMP(양수)를 사용하여 블록을 빠져나가는 경우, 별도의 고려를 하지 않고 있으므로 블록제어가 잘못될 수 있습니다. 이 경우에는 JMP(음수)를 이용하여 NEXT 명령까지 건너뛸 수 있도록 작업하셔야 합니다.																																		
사용 예	{[--[XIC(X21)]--[JMP(25)]--[.....]--[LBL(25)]--I}: X21 릴레이가 1 이면, JMP(25)명령에 의해 LBL(25) 명령이 있는 위치로 점프합니다. LBL 명령이 다른 Rung 에 있어도 정상 동작합니다.																																		

6.33. CALL(Call): 서브래더 호출

심볼	CALL file number 21																												
Relay 조합	<table><tr><th>relay</th><th>Input</th><th>Output</th><th>Timer</th><th>Counter</th><th>Memory</th><th>const (32bit)</th></tr><tr><th>type</th><td>X, DO</td><td>Y, DI, R, K, SP</td><td>T</td><td>C</td><td>M, S</td><td>(L/F)</td></tr><tr><th>inst</th><td>arg. n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>n (B,W,L,F) m</td><td>(L/F)</td></tr><tr><td>CALL</td><td>S_file x</td><td>x</td><td>x</td><td>x</td><td>x</td><td></td></tr></table>	relay	Input	Output	Timer	Counter	Memory	const (32bit)	type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)	inst	arg. n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)	CALL	S_file x	x	x	x	x	
relay	Input	Output	Timer	Counter	Memory	const (32bit)																							
type	X, DO	Y, DI, R, K, SP	T	C	M, S	(L/F)																							
inst	arg. n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	n (B,W,L,F) m	(L/F)																							
CALL	S_file x	x	x	x	x																								
인수	file number: 서브래더 파일번호(1~99)를 지정하는 릴레이 혹은 숫자																												
설명	Rung 이 활성화되면, (file number)으로 지정하는 번호의 서브래더를 호출합니다. 서브래더의 파일명은 S01xxxx.LAD ~ S99xxxx.LAD 으로 최대 99 개가 있을 수 있으며, 파일 명의 xxxx 은 최대 15 글자까지 사용자가 임의로 붙일 수 있습니다.																												
사용 예	 X23 신호가 입력되고 있으면, S01xxxx.LAD 부터 S99xxxx.LAD 의 파일을 순차 호출한다. CALL 명령 실행결과, 해당 번호의 서브래더가 존재하지 않거나 값의 범위가 1~99 를 벗어나면 SP18=On 으로 설정되고, 정상적으로 실행되면 SP18=Off 로 됩니다. 따라서 필요에 따라 서브래더가 있어야만 하는 경우에는 호출 후 SP18 을 이용하여 에러검출이 가능합니다. 메인 래더에서 CALL 명령으로 1~99 번까지의 서브래더를 호출하도록 하고, 응용 별로 서브래더 번호를 배정하면, 응용에 따라 필요한 서브래더만 제어로 로딩하면 해당응용의 래더가 자동으로 실행되는 효과를 기대할 수 있습니다.																												

6.34. [(End): 레더 종료

심볼	
인수	없음
설명	Rung 이 활성화면, 현재 실행되고 있는 래더를 종료합니다. 만일 현재 실행하고 있는 래더가 서브래더이면, 메인래더로 리턴 하고, 현재 실행하고 있는 래더가 메인래더이면, 실행을 종료하고 처음부터 다시 실행합니다.
사용 예	초기화 처리가 필요한 경우, 초기화 처리를 S01_initialize.LAD 라는 파일로 작성하여 메인래더에서 1 번을 호출하도록 하고, S01 의 첫 번째 랭(rung)에  을 추가하면, SP3 은 운전시작 시 한 스캔 동안만 On 되는 릴레이이므로 S01 이 한번만 실행되고, 다음 스캔부터는 첫 번째 Rung 에서 리턴 됩니다.





● **Daegu Office (Head Office)**

50, Techno sunhwan-ro 3-gil, yuga, Dalseong-gun, Daegu, 43022, Korea

● **GRC**

477, Bundangsuseo-ro, Bundang-gu, Seongnam-si, Gyeonggi-do, 13553, Korea

● **대구 사무소**

(43022) 대구광역시 달성군 유가읍 테크노순환로 3 길 50

● **GRC**

(13553) 경기도 성남시 분당구 분당수서로 477

● **ARS : +82-1588-9997 (A/S center)**

● **E-mail : robotics@hyundai-robotics.com**

