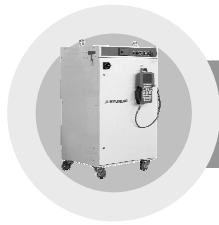


경고

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



Hi4a 제어기 기능설명서

내장 PLC



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

본 설명서는 사전 예고없이 변경될 수 있습니다.

Printed in Korea - 2007 년 10 월 . 3 판
Copyright © 2007 by Hyundai Heavy Industries Co., Ltd



목 차

1. 개요	1-1
2. 제어기 설정	2-1
2.1. 내장 PLC 기능 유효/무효	2-2
2.2. 내장 PLC 동작모드 선택	2-2
2.3. 제어기 TP 상에서 릴레이 상태 모니터링	2-3
2.4. 내장 Scan Time	2-3
3. 입출력 Block Diagram	3-1
4. Relay 사양	4-1
4.1. 릴레이 점수	4-2
4.2. 릴레이 설명	4-3
4.3. 타이머 & 카운터 릴레이	4-6
5. 명령어 사양	5-1
5.1. 기본 명령어	5-2
5.2. 사용가능한 Operands	5-3
5.3. 래더 diagram 와 니모닉의 관계	5-5
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):엔래치 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

목차

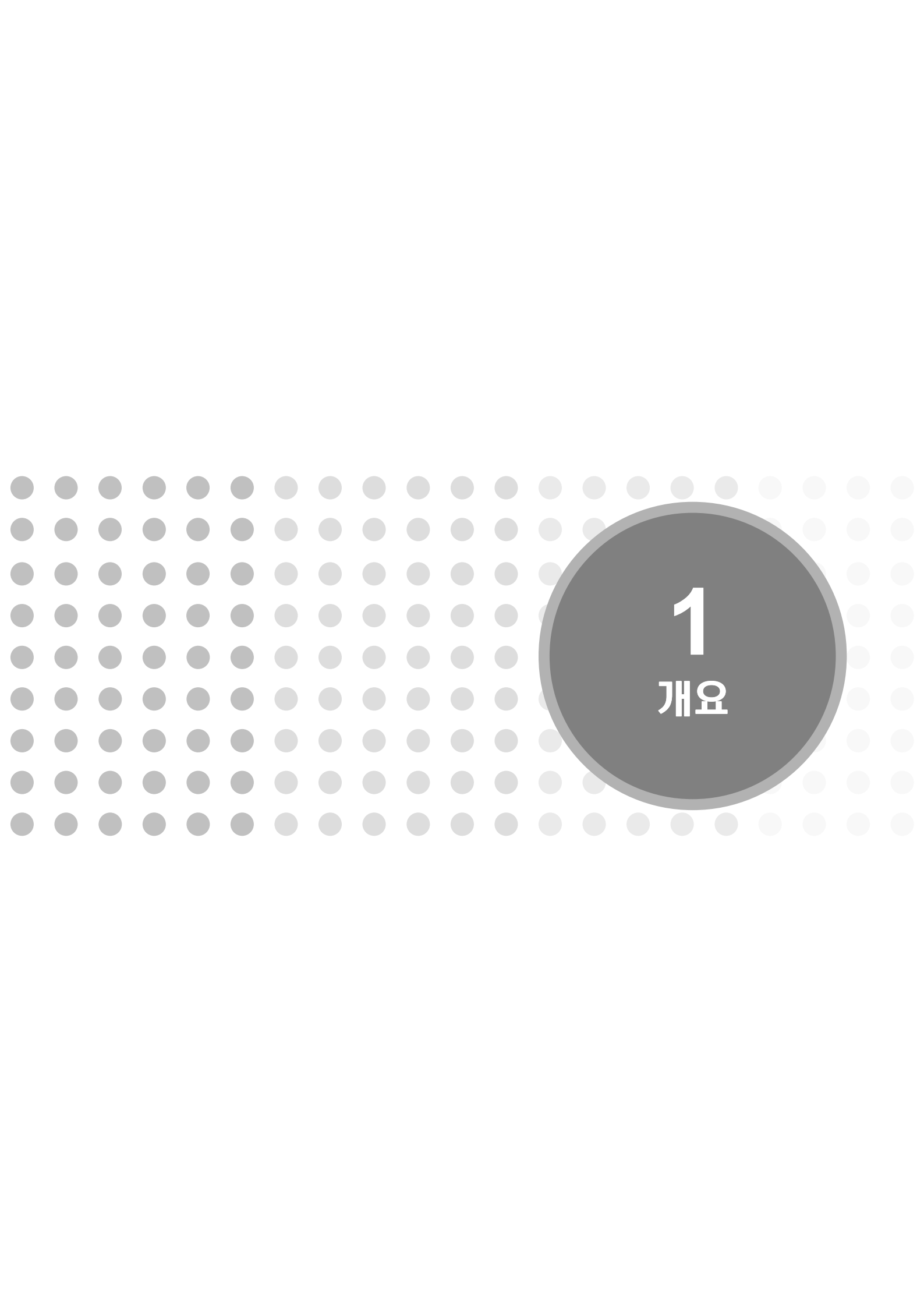
6.16. CTD(Count Down):카운터	6-9
6.17. ADD(Add):더하기	6-10
6.18. SUB(Subtract):빼기	6-10
6.19. MUL(Multiply):곱하기	6-11
6.20. DIV(Divide):나누기	6-12
6.21. POW(Power):거듭제곱	6-13
6.22. TOD(Convert to BCD): BCD 값으로 변환	6-14
6.23. FRD(Convert form BCD to Integer): integer 로 변환	6-15
6.24. SEG(7'Segment):7' 세그먼트값으로 변환	6-16
6.25. MOV(Move):이동	6-17
6.26. COP(Copy data):복사	6-18
6.27. CCOP(Conditional copy data):조건별 복사	6-19
6.28. ROT(Rotating Output):순차 출력	6-20

그림 목차

그림 1.1 BD420 필드버스(Slave/Master)	1-2
그림 3.1 입출력 Block Diagram	3-2

표 목차

표 4-1 릴레이 점수	4-2
표 5-1 기본 명령어	5-2
표 5-2 사용가능한 Operands	5-3



1 개요



1. 개요

내장 PLC 기능설명서

Hi4a 제어기의 내장 PLC 는 상용 PLC 의 기능을 제어기에 구현한 기능입니다. 아래 그림과 같이, 제어기에 연결된 PC 나 Notebook 에서 HRLadder 를 실행하여 래더 프로그램을 작성/편집하고, 제어기에 다운로드 하거나, 제어기에서 실행되고 있는 래더 프로그램을 업로드할 수 있으며, 제어기에서 실행되고 있는 상태를 모니터링 하는 것도 가능합니다. HRLadder 는 별도 기능설명서를 참조하십시오.

BD430/BD431 과 BD420 을 함께 사용하는 경우, 내장 PLC 는 하위의 장치들을 하드-와이어드 I/O 와 필드버스 I/O 로 제어할 수 있습니다. 또한, BD420 의 필드버스 슬레이브 기능을 이용하여, 상위의 라인공정 PLC 와 필드버스 I/O 를 제어할 수 있습니다.(하드-와이어드 I/O 도 가능)

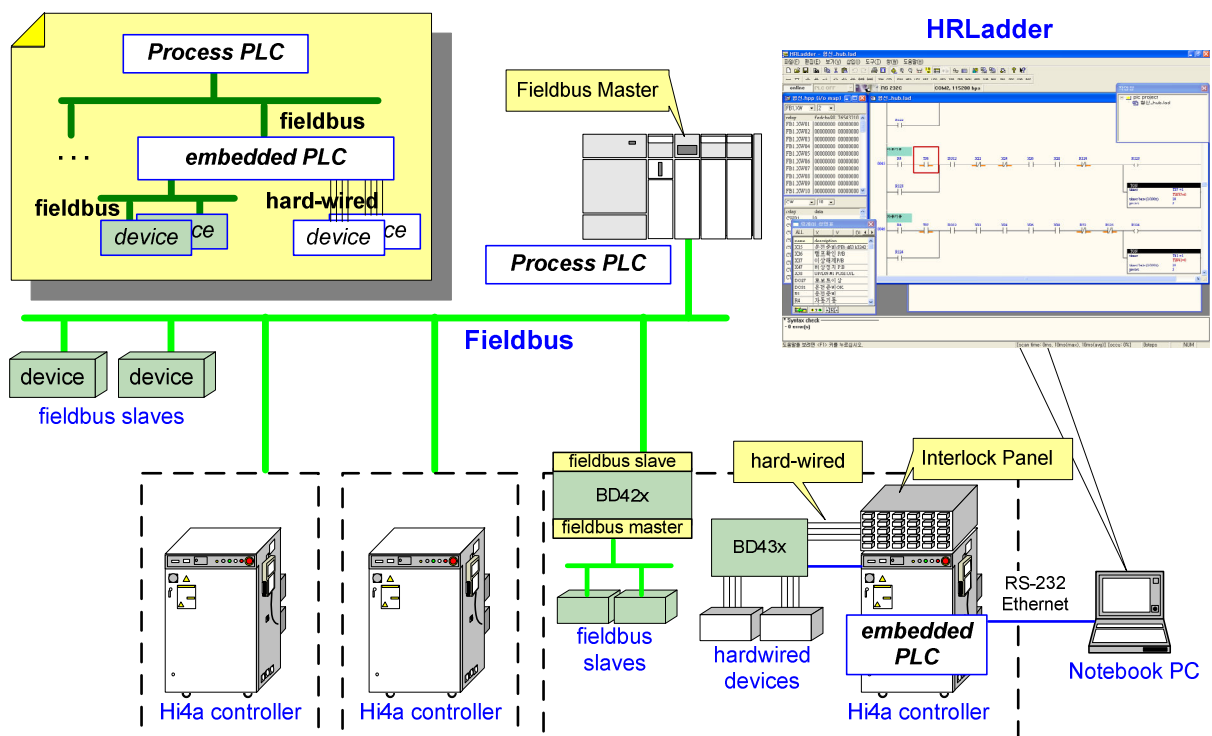


그림 1.1 BD420 필드버스(Slave/Master)

The background features a light gray grid of dots. A dark gray circle is positioned on the right side of the page, containing the number 2 and the text 제어기 설정.

2

제어기 설정



2. 제어기 설정

2.1. 내장 PLC 기능 유효/무효

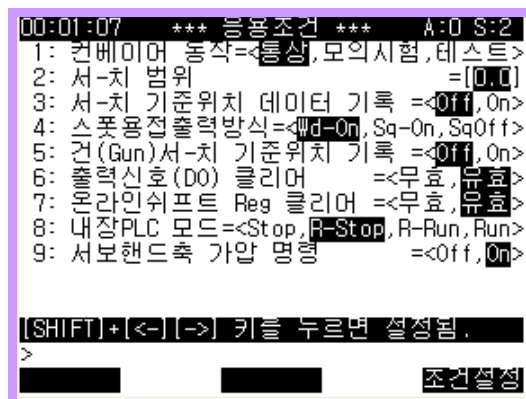
내장 PLC 의 기능을 유효로 하려면, 메인 CPU 보드(BD411/BD412)의 DIP 스위치 DIP5 를 On 으로 하십시오.

내장 PLC 의 기능을 무효로 하려면, DIP 스위치 DIP5 를 Off 로 하십시오.

2.2. 내장 PLC 동작모드 선택

(1) 조작

『[PF5]: 조건설정』 → 『[PF1]: 응용조건』을 누르면, 『8: 내장 PLC 모드 =<Stop, R-Stop, R-Run, Run>』이 표시되며, [SHIFT]+[←][→]로 동작모드를 선택할 수 있습니다.



(2) Stop

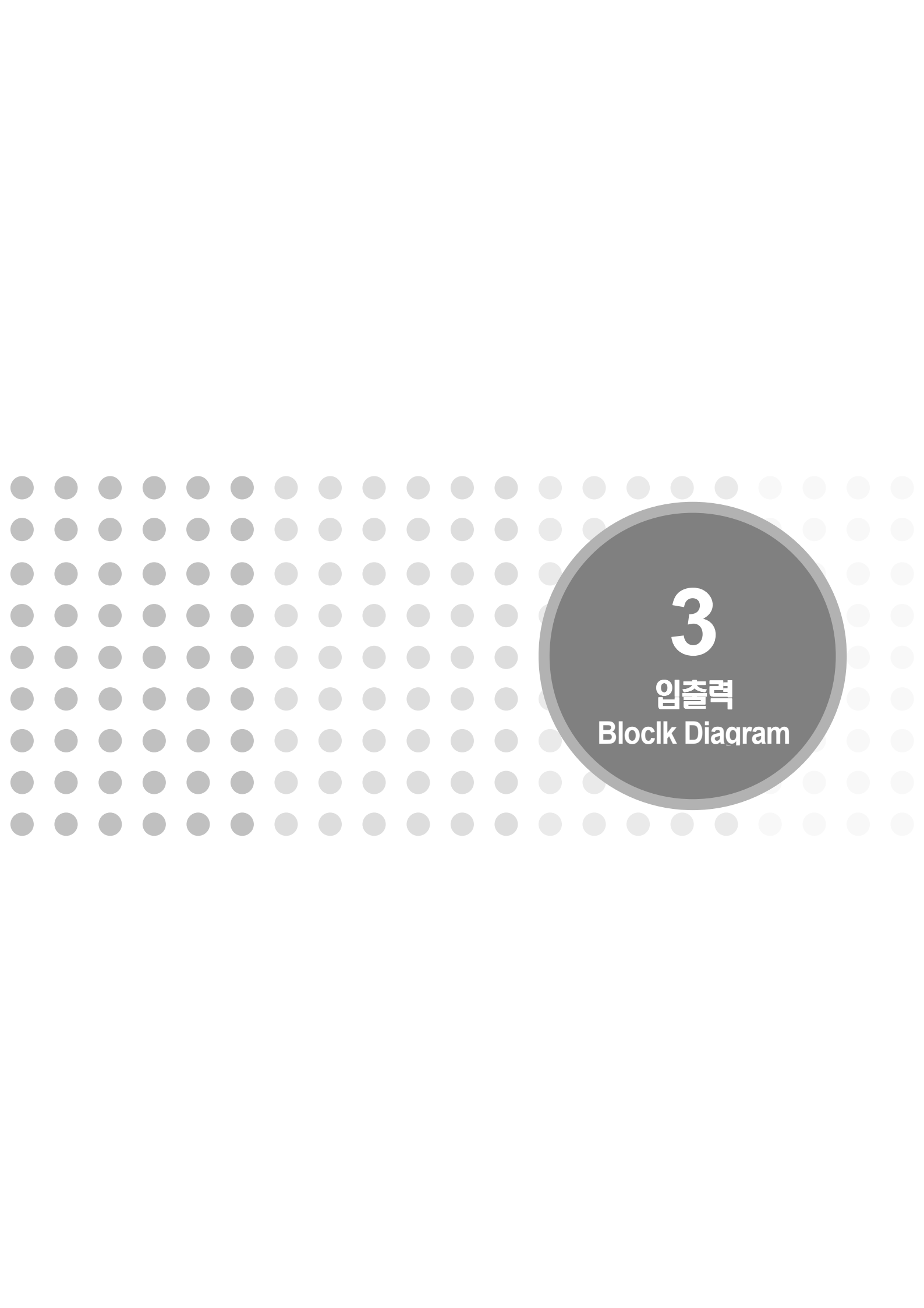
내장 PLC 동작을 정지시킵니다. (Stop 과 동시에 DI,Y,R 릴레이가 clear 됩니다.)

(3) R-Stop/R-Run

Remote 모드이며, 제어기에 연결된 PC 의 HRLadder 에서 R-Stop 과 R-Run 을 선택할 수 있습니다. (R-Stop 과 동시에 DI,Y,R 릴레이가 clear 됩니다)

(4) Run

PLC 실행상태이며 PC 의 HRLadder 에서는 모니터링만 가능합니다.



3

입출력
Block Diagram



3. 입출력 Block Diagram

내장 PLC 기능설명서

- 시스템 메모리(시스템 메모리)는 특수한 용도로 사용하기 위해 예약(reserved)된 것이므로 사용에 제약이 있으며, 향후 제거기 버전에 따라 다른 용도로 변경될 수 있습니다.

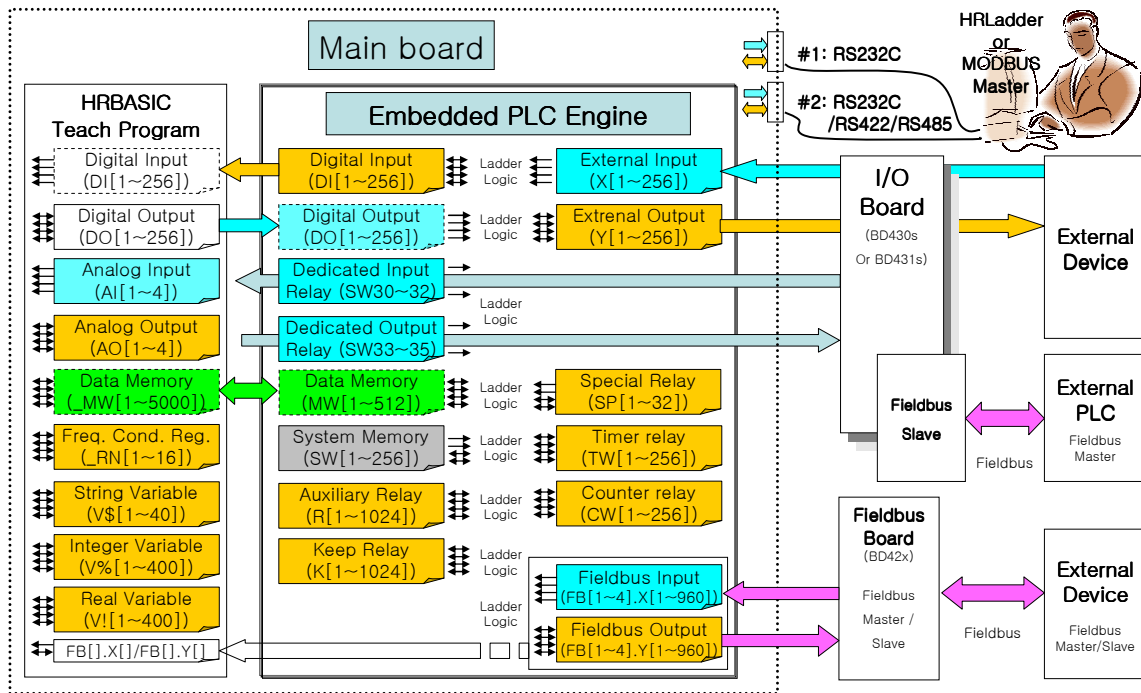


그림 3.1 입출력 Block Diagram



4

Relay 사양



4. Relay 사양

4.1. 릴레이 점수

표 4-1 릴레이 점수

		점수	릴레이(1bit)	번호	릴레이(16/8bit)	번호
외부 입출력 릴레이 (BD430, BD431)	기본	입력 32 점	X01~X32	1	XW01~XW02	2
					XB01~XB04	3
	확장 1 매당 (64/64) Max.3	출력 32 점	Y01~Y32	4	YW01~YW02	5
					YB01~YB04	6
		입력 192 점	X33~X224	1	XW03~XW14	2
					XB05~XB28	3
		출력 192 점	Y33~Y224	4	YW03~YW14	5
					YB05~YB28	6
BD471(CC-Link) 장착시, CC-Link I/O 가 X, Y 릴레이에 선택적으로 매핑됨. CC-Link 설정에 따라, 최대 256 점씩을 각각 X1~X256, Y1~Y256 에 매핑가능.						
제어기 입출력 릴레이 (티칭/재생)	범용	입력 256 점	DI001~DI256	7	DIW01~DIW16	8
					GI01~GI32	9
		출력 256 점	DO001~DO256	10	DOW01~DOW16	11
					G001~G032	12
보조 릴레이		1024 점	R0001~R1024	31	RW01~RW64	32
					RB01~RB128	33
보존 릴레이		1024 점	K0001~K1024	34	KW01~KW64	35
					KB01~KB128	36
특수 릴레이		32 점	SP01~SP32	37	SPW01~SPW02	38
					SPB01~SPB04	39
타이머 릴레이		256 점	T01~T256	40	TW01~TW256	41
					TB01~TB512	42
카운터 릴레이		256 점	C001~C256	43	CW001~CW256	44
					CB001~CB512	45
데이터 메모리		8192 점	M001~M8192	46	MW001~MW512	47
					MB001~MB1024	48
시스템 메모리		4096 점 (Reserved)	S001~S4096 (Reserved)	49	SW001~SW256	50
					SB001~SB512	51
필드버스 릴레이 (DeviceNet/Profibus-DP)		입력 960 점	FB1.X001~X960	55	FB1.XW01~XW60	56
					FB1.XB01~XB120	57
		출력 960 점	FB1.Y001~Y960	67	FB1.YW01~YW60	68
					FB1.YB01~YB120	69
		입력 960 점	FB2.X001~X960	58	FB2.XW01~XW60	59
					FB2.XB01~XB120	60
		출력 960 점	FB2.Y001~Y960	70	FB2.YW01~YW60	71
					FB2.YB01~YB120	72
		입력 960 점	FB3.X001~X960	61	FB3.XW01~XW60	62
					FB3.XB01~XB120	63
		출력 960 점	FB3.Y001~Y960	73	FB3.YW01~YW60	74
					FB3.YB01~YB120	75
		입력 960 점	FB4.X001~X960	64	FB4.XW01~XW60	65
					FB4.XB01~XB120	66
		출력 960 점	FB4.Y001~Y960	76	FB4.YW01~YW60	77
					FB4.YB01~YB120	78

4.2. 릴레이 설명

동일한 릴레이를 3 가지 형식 지정(비트, 바이트, 워드)으로 데이터 사이즈를 선택할 수 있습니다. 다만, 타이머와 카운터는 bit(T/C)와 word(TW/CW)가 동일한 영역을 나타내며, bit(T/C)는 TW/CW 값이 0 인 경우가 활성(active)된 것이다.

(1) 외부 입출력 릴레이 : BD430 과 BD431 를 통해서 입출력되는 실제 신호들

(2) 제어기 범용 입출력 릴레이 : HRBASIC 에서 사용하는 입출력 신호

(3) 필드버스 릴레이 : BD42X 보드를 통해 입출력되는 필드서브 신호들

메인보드 버전	MV10.07 ~ 25 이하	X001 ~ X256, XW01 ~ 16, XB01 ~ XB32
	MV10.07 ~ 26 이상	X001 ~ X960, XW01 ~ 60, XB01 ~ XB120

(4) 보조 릴레이 : PLC program 에서 사용하는 보조 릴레이

(5) 보존 릴레이 : 전원을 Off 해도 On/Off 상태가 보존되는 릴레이

(6) 특수 릴레이 : 특수한 목적을 위해 정의된 릴레이

릴레이 번호	설 명	기 타
SP01	상시 On 릴레이	Controller states
SP02	상시 Off 릴레이	
SP03	운전 개시시 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 경우 On	TOD 나 FRD 실행시
SP08	연산결과 carry 가 있으면 On.	산술명령 실행시
SP09	2 초 clock (1 초 On → 1 초 Off)	Internal 타이머
SP10	4 초 clock (2 초 On → 2 초 Off)	
SP09~SP32	Reserved	

- (7) 타이머 릴레이 :
타이머 동작을 위한 릴레이며, 값이 0 일 때 접점이 On 된다.(정전시 보존)
- (8) 카운터 릴레이 :
카운터 동작을 위한 릴레이며, 값이 0 일 때 접점이 On 된다.(정전시 보존)
- (9) 데이터 메모리 :
응용명령에서 임의의 Data 를 저장하거나 읽어들이기 때 사용.(정전시 보존)
- (10) 시스템 메모리 :
다음과 같은 특수 용도로 사용하고 있으며, 향후 사용자에게 통보없이 용도가 변경될 수 있습니다.

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 I/O 보드 수	
SW5	Main SW Version 의 2 nd + 1 st	20.03-10v → &H0314
SW6	I/O version + Main SW Version 의 3 rd	20.03-10 → &H430A
SW10	Scan time	
SW11	할당 시간	
SW12	최대 점유시간	
SW13	평균 점유시간	
SW14	Ladder 의 총 스텝수	
SW20	제어기의 현재 프로그램 번호	
SW21	제어기의 현재 스텝 번호	
SW22	제어기의 현재 평선 번호	
SW24	FB1~FB4 active	0 th ~3 rd bit
SW25	힐서 상태정보 읽기(1~4) (1:FB1 정보, 2: FB2 정보...)	(since mv20.03-10)
SW26	LSB: Hilscher COM GlobalBits (& 0x84 == 0 : master OK) MSB: Hilscher COM HostFlag (& 0x20 != 0 : slave OK)	
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	

4. 릴레이 사양

번호	설 명	기 타
SW33	전용 출력 1	
SW34	전용 출력 2	
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	로봇위치 읽기 (0:무,1:현재치,2:지령치)	
SW41	로봇 베이스좌표계의 X 값	mm 단위
SW42	로봇 베이스좌표계의 Y 값	mm 단위
SW43	로봇 베이스좌표계의 Z 값	mm 단위
SW44	로봇 베이스좌표계의 RX 값	0.1 deg 단위
SW45	로봇 베이스좌표계의 RY 값	0.1 deg 단위
SW46	로봇 베이스좌표계의 RZ 값	0.1 deg 단위
SW60~69	상대(indirect) 어드레싱을 위한 영역	-1:SW61, -2:SW62, ...
SW70	가동시간 종류(0:무, 1:가동시간, 2:통산가동시간)	
SW71~SW72	가동시간(HH:MM:SS.CC)	10msec 단위
SW73~SW74	싸이클 시간(HH:MM:SS.CC)	10msec 단위
SW75	싸이클 수	
SW76~SW77	용접기 1 용접시간(HH:MM:SS.CC)	10msec 단위
SW78	용접기 1 용접횟수	
SW79~SW80	용접기 2 용접시간(HH:MM:SS.CC)	10msec 단위
SW81	용접기 2 용접횟수	
SW82~SW83	용접기 3 용접시간(HH:HMM:SS.CC)	10msec 단위
SW84	용접기 3 용접횟수	
SW85~SW86	용접기 4 용접시간(HH:MM:SS.CC)	10msec 단위
SW87	용접기 4 용접횟수	
SW88~SW89	Wait, DI 대기 시간(HH:MM:SS.CC)	10msec 단위
SW90~SW91	타이머 대기 시간(HH:MM:SS.CC)	10msec 단위

4.3. 타이머 & 카운터 릴레이

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

- 타이머 base 는 10msec 단위로 사용자가 설정할 수 있습니다.
- 타이머값은 내부적으로 16bit 의 값으로 처리되기 때문에 $32767 \times 10 = 327670\text{msec}$ 까지 counting 할 수 있습니다.

(2) 타이머 / 카운터의 값은 다음과 같은 의미가 있습니다.

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

(3) 타이머 / 카운터 릴레이가 비활성이면,

- TON : TW(타이머)의 값을 -1 로 합니다.
- CTD : CW(카운터)의 값을 계속 유지합니다.

(4) 타이머 / 카운터 릴레이가 활성화되어 있는 동안,

- TON
만일 TW의 값이 0 보다 작으면, TW의 초기값은 “타이머 base x preset” 이 저장되고, TW 값이 0 보다 크면, 10msec 마다 1 씩 감소합니다.
- CTD
만일 CW값이 0 보다 작으면, CW의 초기값은 preset 값이 되고, 만일 CW값이 0 보다 크면, 비활성에서 활성으로 변할때마다 1 씩 감소합니다.



5

명령어 사양



5. 명령어 사양

5.1. 기본 명령어

표 5-1 기본 명령어

명령어				설 명
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	-//-	Rung의 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	-()-	Rung의 상태를 출력(활성:ON/비활성:OFF)
11	OTL	Output Latch	-(L)-	Rung이 활성화되면, ON(high)으로 출력
12	OTU	Output Unlatch	-(U)-	Rung이 활성화되면, OFF(low)로 출력
13	OSR	One Shot Rising	-(OSR)-	Rung이 활성화되면, 한 scan 동안만 ON 출력
14	RES	Reset	-(RES)-	Rung이 활성화되면, 타이머나 카운터를 리셋
타이머 및 카운터 명령				
15	TON	Time On Delay	-□-	Rung이 활성화인 동안 타이머 동작
16	CTD	Count Down	-□-	Rung의 활성화(비활성→활성)을 다운-카운트
산술연산 명령				
17	ADD	Add	-□-	Rung이 활성화이면, (+)연산
18	SUB	Subtract	-□-	Rung이 활성화이면, (-)연산
19	MUL	Multiply	-□-	Rung이 활성화이면, (x)연산
20	DIV	Divide	-□-	Rung이 활성화이면, (/)연산
21	POW	Power	-□-	Rung이 활성화이면, (^:거듭제곱)연산
데이터 변환 명령				
22	TOD	Convert int. to BCD	-□-	Rung이 활성화이면, BCD로 변환
23	FRD	Convert form BCD to int.	-□-	Rung이 활성화이면, integer로 변환
24	SEG	7'Segment	-□-	Rung이 활성화이면, 7' 세스먼트값으로 변환
이동 및 복사 명령				
25	MOV	Move	-□-	Rung이 활성화이면, 데이터 한 개를 복사
26	COP	Copy data	-□-	Rung이 활성화이면, 데이터 여러 개를 복사
27	CCOP	Conditional Copy data	-□-	Rung 상태에 따라 데이터 여러 개를 복사
응용 명령				
28	ROT	Rotating Output	-□-	Rung이 활성화이면, 순차적으로 출력

5-3

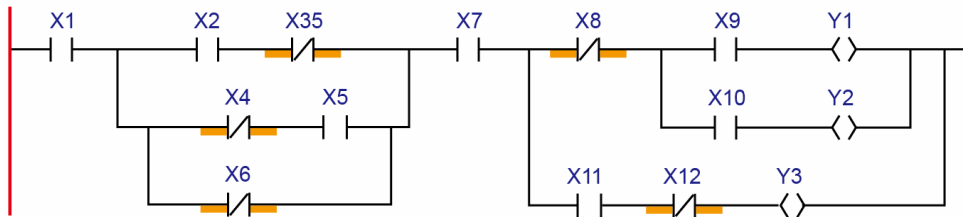

HYUNDAI
 HEAVY INDUSTRIES CO., LTD.

- (1) 표의 'X' 표기는 사용할 수 없음을 나타냅니다.
- (2) 표의 'u' 표기는 부호없는(unsigned) 값으로 처리됨을 나타냅니다.
- (3) 표에 'o' 로 표기된 것은 (2007.4.13)이후 버전부터 사용할 수 있습니다.
- (4) 릴레이(relay)의 n은 비트(bit) 번호를 나타내며, Bn는 바이트(8bit)를 의미하고, Wn는 워드(16bit)를 의미합니다. 다만, DI나 DO의 바이트(8bit) 형식의 명칭은 HRBasic에서와 동일하게 GIn(group Input)과 GOn(Group Output)을 사용하고 있습니다.
- (5) 따라서 명령어에서 릴레이 형식(B, W)을 선택하는 것에 의해 데이터 사이즈(data size)를 선택할 수 있습니다.
- (6) 단 COP(복사)와 CCOP(조건별 복사) 명령의 경우, src가 상수가 아니면, src와 dst의 릴레이 형식은 같아야 합니다.
- (7) 비트에 대한 값을 입력할 때에는 바이트(B)나 워드(W)의 시작위치로 부터의 값으로 입력할 수 있습니다. 예를 들어, X18를 XB3/2나 XW2/2와 같은 형태로 입력할 수 있습니다.
- (8) 상수(Constant)
워드(16bit) size의 2진수, 10진수, 16진수가 있으며, &B로 시작하면 2진수를 의미하고, &H로 시작하면 16진수를 의미하며, 접두어 없는 숫자는 10진수를 의미합니다. 따라서 125는 &H7D나 &B01111101과 같은 형식으로 입력할 수 있습니다.
- (9) 표에 'u'가 표기가 없는 전 릴레이의 W와 B(GI,GO)는 부호있는(signed) 값으로 처리합니다.
예를 들어, MOV와 같은 명령에서 B(8bit)값을 W(16bit)에 옮기는 경우에는 부호비트(signed bit)가 확장되기 때문에 주의해야 합니다. 즉, RB(-1=&HFF)를 RW에 옮기면 &H00FF(255)가 아닌 0xFFFF(-1)이 됩니다.
다만, BCD 데이터에 관련된 TOD와 FRD 명령, 그리고 7' 세그먼트 데이터로 변환하는 SEG 명령은 부호없는 값으로 처리합니다.

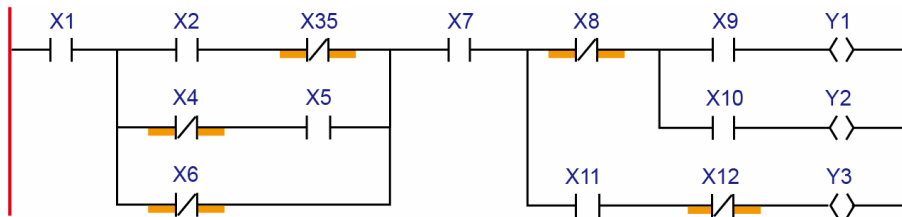
5.3. 래더 diagram 와 니모닉의 관계

- 래더로 작성된 파일은 니모닉으로 자동변환되어 제어기로 송신합니다.

(1) 래더 diagram



또는 편집기 symbol(branch gap) click 으로 다음과 같이 표시할 수 도 있습니다.



(2) 니모닉(Mnemonic)

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




6

명령어 설명



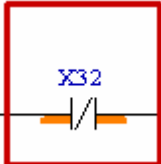
6. 명령어 설명

내장 PLC 기능설명서

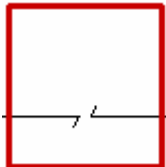
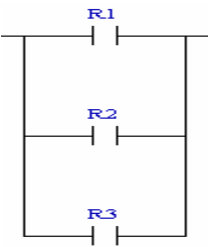
6.1. XIC(Examine if Closed):Close 검사

심볼	X34																
Relay 조합	relay type inst arg.	Input (FB.)X, DO			Output (FB.)Y,DI,R,K,SP			Timer T			Counter C			Memory M(DM), S(SM)			const (16bit)
	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn		
XIC		x	x		x	x		x	x		x	x	o	x	x		
설명	인수의 값이 1 이면 Rung 을 활성(active), 0 이면 비활성 합니다.																
사용 예	XIC(X2); XIC(D02); XIC(Y2); XIC(DI2); XIC(R2); XIC(K2); XIC(SP1); XIC(T2); XIC(C2);																
입력 방법	X18 을 XB3/2 이나 XW2/2 로 입력할 수도 있습니다.																

6.2. XIO(Examine if Open):Open 검사

심볼																																																												
Relay 조합	<table><tr><td rowspan="3">relay</td><td>type</td><td colspan="2">Input</td><td colspan="2">Output</td><td colspan="2">Timer</td><td colspan="2">Counter</td><td colspan="2">Memory</td><td rowspan="3">const (16bit)</td></tr><tr><td>inst</td><td colspan="2">(FB.)X, DO</td><td colspan="2">(FB.)Y,DI,R,K,SP</td><td colspan="2">T</td><td colspan="2">C</td><td colspan="2">M(DM), S(SM)</td></tr><tr><td>arg.</td><td>n</td><td>Bn Wn</td><td>n</td><td>Bn Wn</td><td>n</td><td>Bn Wn</td><td>n</td><td>Bn Wn</td><td>n</td><td>Bn Wn</td></tr><tr><td>XIO</td><td></td><td></td><td>x x</td><td></td><td>x x x</td><td></td><td>x x</td><td></td><td>x x</td><td></td><td>o x x</td><td>x</td></tr></table>	relay	type	Input		Output		Timer		Counter		Memory		const (16bit)	inst	(FB.)X, DO		(FB.)Y,DI,R,K,SP		T		C		M(DM), S(SM)		arg.	n	Bn Wn	n	Bn Wn	n	Bn Wn	n	Bn Wn	n	Bn Wn	XIO			x x		x x x		x x		x x		o x x	x											
relay	type		Input		Output		Timer		Counter		Memory		const (16bit)																																															
	inst		(FB.)X, DO		(FB.)Y,DI,R,K,SP		T		C		M(DM), S(SM)																																																	
	arg.	n	Bn Wn	n	Bn Wn	n	Bn Wn	n	Bn Wn	n	Bn Wn																																																	
XIO			x x		x x x		x x		x x		o x x	x																																																
설명	인수의 값이 0 이면 Rung 을 활성(active), 1 이면 비활성 합니다.																																																											
사용 예	XIO(X2); XIO(D02); XIO(Y2); XIO(DI2); XIO(R2); XIO(K2); XIO(SP1); XIO(T2); XIO(C2);																																																											
입력 방법	X18 을 XB3/2 이나 XW2/2 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 결과 가 됩니다.  

6.4. EQU(Equal):같은지 검사

심볼	EQU source aXB3 0 source b100																																																																																				
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB.)X, DO</th><th colspan="3">(FB.)Y, DI, R, K, SP</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th>arg.</th><th></th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td rowspan="2">EQU</td><td>sA</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td></tr></table>	inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB.)X, DO			(FB.)Y, DI, R, K, SP			T			C			M(DM), S(SM)			arg.		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	EQU	sA	X			X			X			X			X			sB	X			X			X			X			X		
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																			
	type	(FB.)X, DO			(FB.)Y, DI, R, K, SP			T			C			M(DM), S(SM)																																																																							
arg.		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																					
EQU	sA	X			X			X			X			X																																																																							
	sB	X			X			X			X			X																																																																							
인수	sA: 비교하려는 첫번째 릴레이(혹은 숫자) sB: 비교하려는 두번째 릴레이(혹은 숫자)																																																																																				
설명	두 값을 비교하여 같으면, Rung 이 활성화됩니다.(점점활성)																																																																																				
사용 예	EQU(XB3,100):XB3 에 있는 값이 100 이면 Rung 이 활성화됩니다. EQU(XB3,RW4):XB3 과 RW4 에 있는 값이 같으면 Rung 이 활성화됩니다.																																																																																				

6.5. NEQ(Not Equal):다른지 검사

심볼	NEQ source aXB3 0 source b100														
Relay 조합	relay type inst arg. NEQ	Input (FB.)X, DO n	Output (FB.)Y,DI,R,K,SP BnWn	Timer T BnWn	Counter C BnWn	Memory M(DM), S(SM) BnWn	const (16bit)								
인수	sA: 비교하려는 첫번째 릴레이(혹은 숫자) sB: 비교하려는 두번째 릴레이(혹은 숫자)														
설명	두 값을 비교하여 같지 않으면, Rung 이 활성화됩니다.(점점활성)														
사용 예	EQU(XB3,100):XB3 의 값이 100 이 아니면 Rung 이 활성화됩니다. EQU(XB3,RW4):XB3 의 값이 RW4 의 값과 같지 않으면 Rung 이 활성화됩니다.														

6.6. LES(Less Than):작은지 검사

심볼	LES source aXB3 0 source b100														
Relay 조합	relay type inst arg. LES	Input (FB.)X, DO nBnWn	Output (FB.)Y,DI,R,K,SP nBnWn	Timer T nBnWn	Counter C nBnWn	Memory M(DM), S(SM) nBnWn	const (16bit)								
인수	sA: 비교하려는 첫번째 릴레이(혹은 숫자) sB: 비교하려는 두번째 릴레이(혹은 숫자)														
설명	sA 가 sB 보다 작으면 , Rung 이 활성화됩니다.(점점활성)														
사용 예	EQU(XB3,100):XB3 의 값이 100 보다 작으면 Rung 이 활성화됩니다. EQU(XB3,RW4):XB3 의 값이 RW4 의 값보다 작으면 Rung 이 활성화됩니다.														

6.7. GRT(Greater Than):큰지 검사

심볼	GRT source aXB3 0 source b100																																																																																																				
Relay 조합	<table><tr><td rowspan="2">relay</td><td rowspan="2">type</td><td colspan="3">Input</td><td colspan="3">Output</td><td colspan="3">Timer</td><td colspan="3">Counter</td><td colspan="3">Memory</td><td rowspan="2">const (16bit)</td></tr><tr><td colspan="3">(FB,)X, DO</td><td colspan="3">(FB,)Y,DI,R,K,SP</td><td colspan="3">T</td><td colspan="3">C</td><td colspan="3">M(DM), S(SM)</td></tr><tr><td>inst</td><td>arg.</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td></tr><tr><td>GRT</td><td>sA</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td></td><td>sB</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td></tr></table>	relay	type	Input			Output			Timer			Counter			Memory			const (16bit)	(FB,)X, DO			(FB,)Y,DI,R,K,SP			T			C			M(DM), S(SM)			inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	GRT	sA	x			x			x			x			x				sB	x			x			x			x			x																		
relay	type			Input			Output			Timer			Counter			Memory				const (16bit)																																																																																	
		(FB,)X, DO			(FB,)Y,DI,R,K,SP			T			C			M(DM), S(SM)																																																																																							
inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																					
GRT	sA	x			x			x			x			x																																																																																							
	sB	x			x			x			x			x																																																																																							
인수	sA: 비교하려는 첫번째 릴레이(혹은 숫자) sB: 비교하려는 두번째 릴레이(혹은 숫자)																																																																																																				
설명	sA 가 sB 보다 크면 , Rung 이 활성화됩니다.(점점활성)																																																																																																				
사용 예	EQU(XB3,100):XB3 의 값이 100 보다 크면 Rung 이 활성화됩니다. EQU(XB3,RW4):XB3 의 값이 RW4 의 값보다 크면 Rung 이 활성화됩니다.																																																																																																				

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 colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB,)X, DO</th><th colspan="3">(FB,)Y,DI,R,K,SP</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td>LEQ</td><td>SA</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td></td><td>SB</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td></tr></table>	inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB,)X, DO			(FB,)Y,DI,R,K,SP			T			C			M(DM), S(SM)			arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	LEQ	SA	x			x			x			x			x				SB	x			x			x			x			x																	
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																		
	type	(FB,)X, DO			(FB,)Y,DI,R,K,SP			T			C			M(DM), S(SM)																																																																																						
arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																					
LEQ	SA	x			x			x			x			x																																																																																						
	SB	x			x			x			x			x																																																																																						
인수	sA: 비교하려는 첫번째 릴레이(혹은 숫자) sB: 비교하려는 두번째 릴레이(혹은 숫자)																																																																																																			
설명	sA 가 sB 보다 작거나 같으면, Rung 이 활성화됩니다.(점점활성)																																																																																																			
사용 예	EQU(XB3,100):XB3 의 값이 100 보다 작거나 같으면 Rung 이 활성화됩니다. EQU(XB3,RW4):XB3 의 값이 RW4 의 값보다 작거나 같으면 Rung 이 활성화됩니다.																																																																																																			

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

심볼	GEQ source aXB3 0 source b100																																																																																														
Relay 조합	<table><tr><th rowspan="2">relay</th><th rowspan="2">type</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>(FB,X, DO)</th><th colspan="2"></th><th>(FB,Y,DI,R,K,SP)</th><th colspan="2"></th><th>T</th><th colspan="2"></th><th>C</th><th colspan="2"></th><th>M(DM), S(SM)</th><th colspan="2"></th></tr><tr><th>inst</th><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td rowspan="2">GEQ</td><td>SA</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td></tr><tr><td>SB</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td></tr></table>	relay	type	Input			Output			Timer			Counter			Memory			const (16bit)	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)			inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	GEQ	SA	x			x			x			x			x			SB	x			x			x			x			x													
relay	type			Input			Output			Timer			Counter			Memory				const (16bit)																																																																											
		(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)																																																																																	
inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																															
GEQ	SA	x			x			x			x			x																																																																																	
	SB	x			x			x			x			x																																																																																	
인수	sA: 비교하려는 첫번째 릴레이(혹은 숫자) sB: 비교하려는 두번째 릴레이(혹은 숫자)																																																																																														
설명	sA 가 sB 보다 크거나 같으면, Rung 이 활성화됩니다.(접점활성)																																																																																														
사용 예	EQU(XB3,100):XB3 의 값이 100 보다 크거나 같으면 Rung 이 활성화됩니다. EQU(XB3,RW4):XB3 의 값이 RW4 의 값보다 크거나 같으면 Rung 이 활성화됩니다.																																																																																														

6.10. OTE(Output Energize):일반출력

심볼	Y23																
Relay 조합	relay inst OTE type arg.	Input (FB,X, DO) nBnWn			Output (FB,Y,DI,R,K,SP) nBnWn			Timer T nBnWn			Counter C nBnWn			Memory M(DM), S(SM) nBnWn			const (16bit) x
설명	Rung 상태에 따라 출력신호를 출력합니다. 즉, Rung 이 활성화면 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 로 입력할 수도 있습니다.																

6.11. OTL(Output Latch):래치 출력

심볼	Y20 (L)												
Relay 조합	relay type inst OTL Input (FB.)X, DO nBnWn x x x Output (FB.)Y,DI,R,K,SP nBnWn x x x Timer T nBnWn x x x Counter C nBnWn x x x Memory M(DM), S(SM) nBnWn x x x const (16bit) 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 로 입력할 수도 있습니다.												

6.12. OTU(Output Unlatch):엔래치 출력

심볼	Y21 (U)																																																																																		
Relay 조합	<table><tr><td rowspan="3">relay type inst</td><td rowspan="3">arg.</td><td colspan="3">Input</td><td colspan="3">Output</td><td colspan="3">Timer</td><td colspan="3">Counter</td><td colspan="3">Memory</td><td rowspan="3">const (16bit)</td></tr><tr><td colspan="3">(FB.)X, DO</td><td colspan="3">(FB.)Y,DI,R,K,SP</td><td colspan="3">T</td><td colspan="3">C</td><td colspan="3">M(DM), S(SM)</td></tr><tr><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td></tr><tr><td colspan="2">OTU</td><td>x</td><td>x</td><td>x</td><td></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><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	relay type inst	arg.	Input			Output			Timer			Counter			Memory			const (16bit)	(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)			n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	OTU		x	x	x		x	x	x	x	x	x	x	x	x	x	x	x																
relay type inst	arg.			Input			Output			Timer			Counter			Memory				const (16bit)																																																															
				(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)																																																																			
		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																			
OTU		x	x	x		x	x	x	x	x	x	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 로 입력할 수도 있습니다.																																																																																		

6.13. OSR(One Shot Rising):원 샷 출력

심볼	Y22 (OSR)																																																																																			
Relay 조합	<table><tr><td rowspan="3">relay type inst</td><td rowspan="3">arg.</td><td colspan="3">Input</td><td colspan="3">Output</td><td colspan="3">Timer</td><td colspan="3">Counter</td><td colspan="3">Memory</td><td rowspan="3">const (16bit)</td></tr><tr><td colspan="3">(FB.)X, DO</td><td colspan="3">(FB.)Y,DI,R,K,SP</td><td colspan="3">T</td><td colspan="3">C</td><td colspan="3">M(DM), S(SM)</td></tr><tr><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td></tr><tr><td>OSR</td><td></td><td>x</td><td>x</td><td>x</td><td></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><td>x</td><td>x</td><td>x</td><td>x</td></tr></table>	relay type inst	arg.	Input			Output			Timer			Counter			Memory			const (16bit)	(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)			n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	OSR		x	x	x		x	x	x	x	x	x	x	x	x	x	x	x																	
relay type inst	arg.			Input			Output			Timer			Counter			Memory				const (16bit)																																																																
				(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)																																																																				
		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																				
OSR		x	x	x		x	x	x	x	x	x	x	x	x	x	x	x																																																																			
설명	Rung 이 활성화면, 한 scan 동안만 On(high)으로 출력합니다. 즉, Rung 이 비활성이었다가 활성화로 될 때 한 scan 동안만 On 됩니다.																																																																																			
인수 예	Y2; DI2; R2; K2; SP1;																																																																																			
입력 방법	Y18 을 YB3/2 이나 YW2/2 로 입력할 수도 있습니다.																																																																																			

6.14. RES(Reset):리셋

심볼	T3 (RES)																																																																				
Relay 조합	<table><tr><td rowspan="2"></td><td>relay</td><td colspan="3">Input</td><td colspan="3">Output</td><td colspan="3">Timer</td><td colspan="3">Counter</td><td colspan="3">Memory</td><td rowspan="2">const (16bit)</td></tr><tr><td>type</td><td colspan="3">(FB.)X, DO</td><td colspan="3">(FB.)Y,DI,R,K,SP</td><td colspan="3">T</td><td colspan="3">C</td><td colspan="3">M(DM), S(SM)</td></tr><tr><td>inst</td><td>arg.</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td></tr><tr><td>RES</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td></td><td>x</td><td>x</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 (16bit)	type	(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)			inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	RES		x	x	x	x	x	x		x	x		x	x	x	x	x
	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																			
	type	(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)																																																							
inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																					
RES		x	x	x	x	x	x		x	x		x	x	x	x	x																																																					
설명	Rung 이 활성화면 , 출력을 clear(-1)시킵니다.																																																																				
인수 예	T2; C2;																																																																				

6.15. TON(Time On Delay):타이머

심볼	TON timer T2 =1 TW2=0 timer base(1/100s) 100 preset 2																																																																																																																			
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB,X, DO)</th><th colspan="3">(FB,Y,DI,R,K,SP)</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th></th><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</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></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><tr><td>bs</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td></tr><tr><td>pst</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td></tr></table>																inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)				arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	TON	tm	x	x	x	x	x	x		x	x	x	x	x	x	x	x	bs	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	pst	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o
inst	relay	Input			Output			Timer			Counter			Memory				const (16bit)																																																																																																		
	type	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)																																																																																																						
	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																																				
TON	tm	x	x	x	x	x	x		x	x	x	x	x	x	x	x																																																																																																				
	bs	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o																																																																																																				
	pst	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o																																																																																																				
인수	tm: 타이머 릴레이 번호를 지정합니다. bs: 타이머의 단위(100 이면 1 초단위, 10 이면 0.1 초단위, ……)를 릴레이나 숫자로 설정합니다. pst: 프리셋 값(시간=bs*pst[10msec])을 릴레이나 숫자로 설정합니다.																																																																																																																			
설명	Rung 이 활성화인 시간을 계산하여 설정한 시간(bs x pst) 이후에 해당 타이머 릴레이가 On(high)으로 됩니다. 그러나 Rung 이 비활성이면, 즉시 클리어(-1) 설정됩니다. 참고) TW 의 값은 10msec 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)의 값 만큼 타이머로 설정																																																																																																																			

6.16. CTD(Count Down):카운터

심볼	CTD counter C2=1 CW2=0 preset 100																																																																																												
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay type</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>arg.</th><th>(FB,X, DO)</th><th colspan="2">(FB,Y,DI,R,K,SP)</th><th colspan="2">(FB,Y,DI,R,K,SP)</th><th colspan="2">T</th><th colspan="2">C</th><th colspan="2">M(DM), S(SM)</th></tr><tr><td></td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td></tr><tr><td rowspan="2">CTD</td><td>cnt</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><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td></tr><tr><td>pst</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td></tr></table>	inst	relay type	Input			Output			Timer			Counter			Memory			const (16bit)	arg.	(FB,X, DO)	(FB,Y,DI,R,K,SP)		(FB,Y,DI,R,K,SP)		T		C		M(DM), S(SM)			n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	CTD	cnt	x	x	x	x	x	x	x	x	x	x	x	x	x	x	pst	x	o	o	x	o	o	x	o	o	x	o	o	x	o															
inst	relay type		Input			Output			Timer			Counter			Memory			const (16bit)																																																																											
	arg.	(FB,X, DO)	(FB,Y,DI,R,K,SP)		(FB,Y,DI,R,K,SP)		T		C		M(DM), S(SM)																																																																																		
	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																														
CTD	cnt	x	x	x	x	x	x	x	x	x	x	x	x	x	x																																																																														
	pst	x	o	o	x	o	o	x	o	o	x	o	o	x	o																																																																														
인수	cnt: 카운터 릴레이 번호를 지정합니다. pst: 프리셋 값을 릴레이나 숫자로 설정합니다.																																																																																												
설명	Rung의 상승(비활성에서 활성)을 다운-카운트합니다 해당 CW의 값이 0이면 해당 카운터가 On(high)으로 되고, 더 이상 카운트 하지 않습니다. Rung이 활성화일때, 해당 CW의 값이 음수이면, pst 값을 CW에 저장합니다. 참고) Rung이 비활성이라도 CW는 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="3">inst</th><th colspan="2">relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="3">const (16bit)</th></tr><tr><th rowspan="2">type</th><th rowspan="2">arg.</th><th colspan="3">(FB,X, DO)</th><th colspan="3">(FB,Y,DI,R,K,SP)</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td rowspan="3">ADD</td><td>sA</td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>sB</td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr><tr><td>dst</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td></tr></table>	inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)	type	arg.	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)			n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	ADD	sA		x			x			x			x			x				sB		x			x			x			x			x				dst		x	x	x	x			x			x			x			x
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																									
	type		arg.	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)																																																																																											
		n		Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																										
ADD	sA		x			x			x			x			x																																																																																												
	sB		x			x			x			x			x																																																																																												
	dst		x	x	x	x			x			x			x			x																																																																																									
인수	sA: 연산을 위한 첫번째 릴레이(혹은 숫자) sB: 연산을 위한 두번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																																																																										
설명	Rung 이 활성화되면, 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 aXB30 source b50 destinationRW30																																																																																																														
Relay 조합	<table><tr><th rowspan="2">inst</th><th colspan="2">relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th>arg.</th><th colspan="3">(FB,X, DO)</th><th colspan="3">(FB,Y,DI,R,K,SP)</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th></th><th></th><th></th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th></th></tr><tr><td rowspan="3">SUB</td><td>SA</td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td>SB</td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td></td></tr><tr><td>dst</td><td></td><td>X</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td></tr></table>	inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)	type	arg.	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)						n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn		SUB	SA		X			X			X			X			X				SB		X			X			X			X			X				dst		X	X	X	X			X			X			X			X
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																													
	type	arg.	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)																																																																																																
			n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																														
SUB	SA		X			X			X			X			X																																																																																																
	SB		X			X			X			X			X																																																																																																
	dst		X	X	X	X			X			X			X			X																																																																																													
인수	sA: 연산을 위한 첫번째 릴레이(혹은 숫자) sB: 연산을 위한 두번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																																																																														
설명	Rung 이 활성화면, 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	<pre> MUL source a XB3 0 source b 50 destination RW3 0 </pre>																																																																																																																									
Relay 조합	<table border="1"> <thead> <tr> <th rowspan="3">inst</th><th colspan="2">relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="3">const (16bit)</th></tr> <tr> <th rowspan="2">type</th><th rowspan="2">arg.</th><th colspan="3">(FB,X, DO)</th><th colspan="3">(FB,Y,DI,R,K,SP)</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr> <tr> <th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr> </thead> <tbody> <tr> <td rowspan="3">MUL</td><td>sA</td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr> <tr> <td>sB</td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td></td></tr> <tr> <td>dst</td><td></td><td>x</td><td>x</td><td>x</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td></tr> </tbody> </table>	inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)	type	arg.	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)			n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	MUL	sA		x			x			x			x			x				sB		x			x			x			x			x				dst		x	x	x	x			x			x			x			x															
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																																								
	type		arg.	(FB,X, DO)			(FB,Y,DI,R,K,SP)			T			C			M(DM), S(SM)																																																																																																										
		n		Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																																									
MUL	sA		x			x			x			x			x																																																																																																											
	sB		x			x			x			x			x																																																																																																											
	dst		x	x	x	x			x			x			x			x																																																																																																								
인수	sA: 연산을 위한 첫번째 릴레이(혹은 숫자) sB: 연산을 위한 두번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																																																																																									
설명	Rung 이 활성화되면, 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 aXB3 0 source b50 destinationRW3 0																																																																																																				
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB,)X, DO</th><th colspan="3">(FB,)Y, DI, R, K, SP</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th></th><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td rowspan="3">DIV</td><td>sA</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td></tr><tr><td>sB</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></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>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td>X</td></tr></table>	inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB,)X, DO			(FB,)Y, DI, R, K, SP			T			C			M(DM), S(SM)				arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	DIV	sA	X			X			X			X			X			sB	X			X			X			X			X			dst	X	X	X	X			X			X			X		X
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																			
	type	(FB,)X, DO			(FB,)Y, DI, R, K, SP			T			C			M(DM), S(SM)																																																																																							
	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																					
DIV	sA	X			X			X			X			X																																																																																							
	sB	X			X			X			X			X																																																																																							
	dst	X	X	X	X			X			X			X		X																																																																																					
인수	sA: 연산을 위한 첫번째 릴레이(혹은 숫자) sB: 연산을 위한 두번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																																																																				
설명	Rung 이 활성화이면, 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="3">inst</th><th>relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="3">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB.)X, DO</th><th colspan="3">(FB.)Y,DI,R,K,SP</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td rowspan="3">POW</td><td>sA</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>o</td></tr><tr><td>sB</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>o</td></tr><tr><td>dst</td><td>x</td><td>x</td><td>x</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td><td>o</td><td>o</td><td>x</td></tr></table>	inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)			arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	POW	sA	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	o	sB	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	o	dst	x	x	x	x	o	o	x	o	o	x	o	o	x	o	o	x															
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																																				
	type		(FB.)X, DO			(FB.)Y,DI,R,K,SP			T			C			M(DM), S(SM)																																																																																																							
	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																																						
POW	sA	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	o																																																																																																					
	sB	x	o	o	x	o	o	x	o	o	x	o	o	x	o	o	o																																																																																																					
	dst	x	x	x	x	o	o	x	o	o	x	o	o	x	o	o	x																																																																																																					
인수	sA: 연산을 위한 첫번째 릴레이(혹은 숫자) sB: 연산을 위한 두번째 릴레이(혹은 숫자) dst: 결과를 저장할 릴레이																																																																																																																					
설명	Rung 이 활성화면, 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><th rowspan="2">relay</th><th rowspan="2">type</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>(FB.)X, DO</th><th colspan="2">(FB.)Y, DI, R, K, SP</th><th colspan="2">(FB.)Y, DI, R, K, SP</th><th colspan="2">T</th><th colspan="2">C</th><th colspan="2">M(DM), S(SM)</th></tr><tr><th>inst</th><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td>TOD</td><td>src</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td></tr><tr><td></td><td>dst</td><td>x</td><td>x</td><td>x</td><td>x</td><td>u</td><td>u</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>u</td><td>u</td></tr></table>	relay	type	Input			Output			Timer			Counter			Memory			const (16bit)	(FB.)X, DO	(FB.)Y, DI, R, K, SP		(FB.)Y, DI, R, K, SP		T		C		M(DM), S(SM)		inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	TOD	src	x	u	u	x	u	u	x	u	u	x	u	u	x	u	u		dst	x	x	x	x	u	u	x	x	x	x	x	x	x	u	u
relay	type			Input			Output			Timer			Counter			Memory				const (16bit)																																																													
		(FB.)X, DO	(FB.)Y, DI, R, K, SP		(FB.)Y, DI, R, K, SP		T		C		M(DM), S(SM)																																																																						
inst	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																	
TOD	src	x	u	u	x	u	u	x	u	u	x	u	u	x	u	u																																																																	
	dst	x	x	x	x	u	u	x	x	x	x	x	x	x	u	u																																																																	
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이																																																																																
설명	Rung 이 활성화면, src 의 값을 BCD 로 변환하여 dst 에 저장합니다. 이 명령은 BCD 형식으로 7' 세스먼트에 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. 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 sourceXB30 destinationRB30																																																																																																				
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB,)X, DO</th><th colspan="3">(FB,)Y, DI, R, K, SP</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th></th><th>arg.</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td>FRD</td><td>src</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td><td>x</td><td>u</td><td>u</td></tr><tr><td></td><td>dst</td><td>x</td><td>x</td><td>x</td><td>x</td><td>u</td><td>u</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>x</td><td>u</td><td>u</td></tr></table>	inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB,)X, DO			(FB,)Y, DI, R, K, SP			T			C			M(DM), S(SM)				arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	FRD	src	x	u	u	x	u	u	x	u	u	x	u	u	x	u	u		dst	x	x	x	x	u	u	x	x	x	x	x	x	x	u	u															
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																			
	type	(FB,)X, DO			(FB,)Y, DI, R, K, SP			T			C			M(DM), S(SM)																																																																																							
	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																					
FRD	src	x	u	u	x	u	u	x	u	u	x	u	u	x	u	u																																																																																					
	dst	x	x	x	x	u	u	x	x	x	x	x	x	x	u	u																																																																																					
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 캄스위치의 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, &H35(53)를 RB2 에 저장합니다.																																																																																																				

6.24. SEG(7' Segment):7' 세그먼트값으로 변환

심볼	SEG sourceXB30 destinationRW30															
Relay 조합	relay inst type arg. SEG src dst Input (FB.)X, DO nBnWn xuxu xuxx Output (FB.)Y,DI,R,K,SP nBnWn xuuu xuuu Timer T nBnWn xuxx xuxx Counter C nBnWn xuuu xxux Memory M(DM), S(SM) nBnWn xuuu xuuu const (16bit) u x															
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이															
설명	Rung이 활성화면, src의 값을 7' 세그먼트값(8bit)으로 변환하여 dst에 저장합니다. dst가 워드(W)형식이면, 2개의 7' 세그먼트값(8bit)을 dst에 저장합니다.															
7' 세스먼트 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								0x01 SEGM_A 0x02 SEGM_B 0x04 SEGM_C 0x08 SEGM_D 0x10 SEGM_E 0x20 SEGM_F 0x40 SEGM_G 0x80 SEGM_DP							
	#define SEGD_0 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_E SEGM_F) #define SEGD_1 (SEGM_B SEGM_C) #define SEGD_2 (SEGM_A SEGM_B SEGM_G SEGM_E SEGM_D) #define SEGD_3 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_G) #define SEGD_4 (SEGM_B SEGM_C SEGM_F SEGM_G) #define SEGD_5 (SEGM_A SEGM_C SEGM_D SEGM_F SEGM_G) #define SEGD_6 (SEGM_A SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEGD_7 (SEGM_A SEGM_B SEGM_C) #define SEGD_8 (SEGM_A SEGM_B SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEGD_9 (SEGM_A SEGM_B SEGM_C SEGM_F SEGM_G) #define SEGD_A (SEGM_A SEGM_B SEGM_C SEGM_E SEGM_F SEGM_G) #define SEGD_B (SEGM_C SEGM_D SEGM_E SEGM_F SEGM_G) #define SEGD_C (SEGM_A SEGM_D SEGM_E SEGM_F) #define SEGD_D (SEGM_B SEGM_C SEGM_D SEGM_E SEGM_G) #define SEGD_E (SEGM_A SEGM_D SEGM_E SEGM_F SEGM_G) #define SEGD_F (SEGM_A SEGM_E SEGM_F SEGM_G)															
사용 예	SEG(XB3,RW3): XB3의 값에 해당하는 7' 세그먼트값을 RW3에 저장. 만일 (XB3)=(&H17)인 경우, 상기 SEGD_1(SEGM_B SEGM_C=0x02 0x04=0x06)=&H06과 상기 SEGD_7(SEGM_A SEGM_B SEGM_C=0x01 0x02 0x04=0x07)=&H07이 결합된 &H0607를 RW3에 저장합니다.															

6.25. MOV(Move):이동

심볼	MOV source55 destinationRB2 0																																																																																						
Relay 조합	<table><tr><td rowspan="2">inst</td><td>relay type</td><td colspan="3">Input (FB,X, DO)</td><td colspan="3">Output (FB,Y,DI,R,K,SP)</td><td colspan="3">Timer T</td><td colspan="3">Counter C</td><td colspan="3">Memory M(DM), S(SM)</td><td rowspan="2">const (16bit)</td></tr><tr><td>arg.</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td><td>n</td><td>Bn</td><td>Wn</td></tr><tr><td rowspan="2">MOV</td><td>src</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></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></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td></tr></table>	inst	relay type	Input (FB,X, DO)			Output (FB,Y,DI,R,K,SP)			Timer T			Counter C			Memory M(DM), S(SM)			const (16bit)	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	MOV	src	x			x			x			x			x				dst	x	x	x	x			x			x			x			x																	
inst	relay type		Input (FB,X, DO)			Output (FB,Y,DI,R,K,SP)			Timer T			Counter C			Memory M(DM), S(SM)			const (16bit)																																																																					
	arg.	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																							
MOV	src	x			x			x			x			x																																																																									
	dst	x	x	x	x			x			x			x			x																																																																						
인수	src: 원본(source) 릴레이(혹은 숫자) dst: 목적지(destination) 릴레이																																																																																						
설명	Rung 이 활성화되면, src 의 값을 dst 에 복사합니다. 만일, src 가 워드(W)형식이고 dst 가 바이트(B)형식이면, src 의 값중 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 source XB2 0 destination YB3 0 length 3																																																																																																									
Relay 조합	<table><tr><th rowspan="2">inst</th><th>relay</th><th colspan="3">Input</th><th colspan="3">Output</th><th colspan="3">Timer</th><th colspan="3">Counter</th><th colspan="3">Memory</th><th rowspan="2">const (16bit)</th></tr><tr><th>type</th><th colspan="3">(FB.)X, DO</th><th colspan="3">(FB.)Y, DI, R, K, SP</th><th colspan="3">T</th><th colspan="3">C</th><th colspan="3">M(DM), S(SM)</th></tr><tr><th>arg.</th><th></th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td>COP</td><td>src</td><td>0</td><td></td><td></td><td>0</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>0</td><td></td><td></td><td>0</td></tr><tr><td></td><td>dst</td><td>X</td><td>X</td><td>X</td><td>0</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>0</td><td></td><td></td><td>X</td></tr><tr><td></td><td>len</td><td>X</td><td>0</td><td>0</td><td>X</td><td>0</td><td>0</td><td>X</td><td>0</td><td>0</td><td>X</td><td>0</td><td>0</td><td>X</td><td>0</td><td>0</td><td>0</td></tr></table>	inst	relay	Input			Output			Timer			Counter			Memory			const (16bit)	type	(FB.)X, DO			(FB.)Y, DI, R, K, SP			T			C			M(DM), S(SM)			arg.		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	COP	src	0			0			X			X			0			0		dst	X	X	X	0			X			X			0			X		len	X	0	0	X	0	0	X	0	0	X	0	0	X	0	0	0
inst	relay		Input			Output			Timer			Counter			Memory			const (16bit)																																																																																								
	type	(FB.)X, DO			(FB.)Y, DI, R, K, SP			T			C			M(DM), S(SM)																																																																																												
arg.		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																										
COP	src	0			0			X			X			0			0																																																																																									
	dst	X	X	X	0			X			X			0			X																																																																																									
	len	X	0	0	X	0	0	X	0	0	X	0	0	X	0	0	0																																																																																									
인수	src: 원본(source) 릴레이 혹은 채우고자 하는 값(const) dst: 목적지(destination) 릴레이 len: 개수(length) 릴레이(혹은 숫자)																																																																																																									
설명	Rung 이 활성화면, src 의 위치에서부터 len 수 만큼 dst 의 위치로 값들을 복사합니다. src가 숫자인 경우, dst 에서부터 len수 만큼의 src값으로 채워집니다. 이 경우 dst 가 bit 형식이면, src의 숫자가 0 이면 OFF, 0이 아니면 ON으로 채워집니다. src가 릴레이인 경우, src와 dst 의 데이터 형은 같아야 합니다. 즉, src가 비트이면 dst 도 비트, src가 바이트(B)이면 dst 도 바이트(B), src가 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):조건별 복사

심볼	CCOP source aX20 source bR50 destinationY30 length3																																																																																																		
Relay 조합	<table><tr><th rowspan="2">inst</th><th rowspan="2">relay type arg.</th><th colspan="3">Input (FB.)X, DO</th><th colspan="3">Output (FB.)Y,DI,R,K,SP</th><th colspan="3">Timer T</th><th colspan="3">Counter C</th><th colspan="3">Memory M(DM), S(SM)</th><th rowspan="2">const (16bit)</th></tr><tr><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th><th>n</th><th>Bn</th><th>Wn</th></tr><tr><td rowspan="4">CCOP</td><td>sA</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>sB</td><td></td><td></td><td></td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td></td><td></td><td></td></tr><tr><td>dst</td><td>X</td><td>X</td><td>X</td><td></td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td>X</td></tr><tr><td>len</td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td><td>X</td><td></td><td></td></tr></table>	inst	relay type arg.	Input (FB.)X, DO			Output (FB.)Y,DI,R,K,SP			Timer T			Counter C			Memory M(DM), S(SM)			const (16bit)	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	CCOP	sA							X			X						sB							X			X						dst	X	X	X				X			X			X		X	len	X			X			X			X			X		
inst	relay type arg.			Input (FB.)X, DO			Output (FB.)Y,DI,R,K,SP			Timer T			Counter C			Memory M(DM), S(SM)				const (16bit)																																																																															
		n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn	n	Bn	Wn																																																																																			
CCOP	sA							X			X																																																																																								
	sB							X			X																																																																																								
	dst	X	X	X				X			X			X		X																																																																																			
	len	X			X			X			X			X																																																																																					
인수	sA: Rung 이 활성화일 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 det 의 위치로 값들을 복사합니다. src가 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 채워 집니다. 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 활성화이면, Y3=X2, Y4=X3, Y5=X4 Rung 이 비활성이면, Y3=R5, Y4=R6, Y5=R7 CCOP(1,0,Y3,3): Rung 이 활성화이면, Y3=ON, Y4=ON, Y5=ON Rung 이 비활성이면, Y3=OFF, Y4=OFF, Y5=OFF 참고) 인수사용예는 COP 명령과 유사하므로 COP 명령을 참고하세요.																																																																																																		

6.28. ROT(Rotating Output):순차 출력

심볼	ROT start relayMW510 count6 timer relayT51 repeat time(1/100s)200 out relayMW580 reset relayX30 temp relayMW600																																																																																																												
Relay 조합	<table><tr><th rowspan="2">inst</th><th rowspan="2">relay type</th><th colspan="2">Input (FB,X,DO)</th><th colspan="2">Output (FB,Y,D,I,R,K,SP)</th><th colspan="2">Timer T</th><th colspan="2">Counter C</th><th colspan="2">Memory M(DM), S(SM)</th><th rowspan="2">const (16bit)</th></tr><tr><th>arg. n</th><th>BnWn</th><th>nBnWn</th><th>nBnWn</th><th>nBnWn</th><th>nBnWn</th><th>nBnWn</th><th>nBnWn</th></tr><tr><td rowspan="5">ROT</td><td>st</td><td>x</td><td></td><td></td><td>x</td><td></td><td>x</td><td></td><td>x</td><td></td><td>x</td><td>x</td></tr><tr><td>cnt</td><td>x</td><td></td><td></td><td>x</td><td>x</td><td>x</td><td></td><td>x</td><td></td><td>x</td><td>x</td></tr><tr><td>tm</td><td>x</td><td></td><td></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><tr><td>rep</td><td>x</td><td>0</td><td>0</td><td>x</td><td>0</td><td>0</td><td>x</td><td>0</td><td>0</td><td>x</td><td>0</td></tr><tr><td>out</td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td><td></td><td>x</td><td></td></tr><tr><td></td><td>rst</td><td>x</td><td></td><td></td><td>x</td><td></td><td>x</td><td></td><td>x</td><td></td><td>x</td><td>x</td></tr><tr><td></td><td>tmp</td><td>x</td><td></td><td></td><td>x</td><td></td><td>x</td><td></td><td>x</td><td></td><td>x</td><td>x</td></tr></table>	inst	relay type	Input (FB,X,DO)		Output (FB,Y,D,I,R,K,SP)		Timer T		Counter C		Memory M(DM), S(SM)		const (16bit)	arg. n	BnWn	nBnWn	nBnWn	nBnWn	nBnWn	nBnWn	nBnWn	ROT	st	x			x		x		x		x	x	cnt	x			x	x	x		x		x	x	tm	x			x	x	x	x	x	x	x	x	rep	x	0	0	x	0	0	x	0	0	x	0	out	x			x			x			x			rst	x			x		x		x		x	x		tmp	x			x		x		x		x	x
inst	relay type			Input (FB,X,DO)		Output (FB,Y,D,I,R,K,SP)		Timer T		Counter C		Memory M(DM), S(SM)			const (16bit)																																																																																														
		arg. n	BnWn	nBnWn	nBnWn	nBnWn	nBnWn	nBnWn	nBnWn																																																																																																				
ROT	st	x			x		x		x		x	x																																																																																																	
	cnt	x			x	x	x		x		x	x																																																																																																	
	tm	x			x	x	x	x	x	x	x	x																																																																																																	
	rep	x	0	0	x	0	0	x	0	0	x	0																																																																																																	
	out	x			x			x			x																																																																																																		
	rst	x			x		x		x		x	x																																																																																																	
	tmp	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 활성화면, st 릴레이부터 cnt 범위내의 릴레이값이 0 이 아닌 값을 rep 시간동안 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 시간동안 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 초동안 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 클리어되고, MW58 과 MW60 도 0 으로 클리어 되어, 표시장치에 0 이 표시됩니다.																																																																																																												



● **Head Office**

Tel. 82-52-202-7901 / Fax. 82-52-202-7900
1, Jeonha-dong, Dong-gu, Ulsan, Korea

● **A/S Center**

Tel. 82-52-202-5041 / Fax. 82-52-202-7960

● **Seoul Office**

Tel. 82-2-746-4711 / Fax. 82-2-746-4720
140-2, Gye-dong, Jongno-gu, Seoul, Korea

● **Ansan Office**

Tel. 82-31-409-4945 / Fax. 82-31-409-4946
1431-2, Sa-dong, Sangnok-gu, Ansan-si, Gyeonggi-do, Korea

● **Cheonan Office**

Tel. 82-41-576-4294 / Fax. 82-41-576-4296
355-15, Daga-dong, Cheonan-si, Chungcheongnam-do, Korea

● **Daegu Office**

Tel. 82-53-746-6232 / Fax. 82-53-746-6231
223-5, Beomeo 2-dong, Suseong-gu, Daegu, Korea

● **Gwangju Office**

Tel. 82-62-363-5272 / Fax. 82-62-363-5273
415-2, Nongseong-dong, Seo-gu, Gwangju, Korea

● **본사**

Tel. 052-202-7901 / Fax. 052-202-7900
울산광역시 동구 전하동 1번지

● **A/S 센터**

Tel. 82-52-202-5041 / Fax. 82-52-202-7960

● **서울 사무소**

Tel. 02-746-4711 / Fax. 02-746-4720
서울특별시 종로구 계동 140-2번지

● **안산 사무소**

Tel. 031-409-4959 / Fax. 031-409-4946
경기도 안산시 상록구 사동 1431-2번지

● **천안 사무소**

Tel. 041-576-4294 / Fax. 041-576-4296
충남 천안시 다가동 355-15번지

● **대구 사무소**

Tel. 053-746-6232 / Fax. 053-746-6231
대구광역시 수성구 범어 2동 223-5번지

● **광주 사무소**

Tel. 062-363-5272 / Fax. 062-363-5273
광주광역시 서구 농성동 415-2번지