RS OEMax

NX70 Plus 시스템 (NX70-CPU70p1, NX70-CPUp2)

사용 설명서

Catalog Number(s): NX70-CPU70p1, NX70-CPU70p2)

RS Reliable & Smart Automation

일러두기

이 문서에서 설명하는 제품은 다양한 애플리케이션에 사용될 수 있습니다. 본 제품의 적용 애플리케이션이 다양하므로 본 제품의 모든 사용자와 관리자는 사용하고자 하는 애플리케이션과 관련된 운전 및 안전상의 제 법령 및 규정과 준수사항을 반드시 확인하고 지켜야합니다.

사용자가 본 제품을 사용 및 응용함에 따른 직접 또는 간접적인 손상에 대해서 알에스오토메이션(주)는 그 책임을 면합니다.

이 문서에 사용된 예제, 삽화, 도표, 데이터 등은 설명을 위한 예시적인 자료일 뿐입니다. 제품이 설치 및 사용되는 구체적인 상황에 따라 많은 변수와 충족되어야 할 전제 조건들이 있습니다. 이러한 사유로 이 문서에서 제시된 예시, 도표 및 데이터의 사용 및 이에 따른 직간접적 결과에 대해 알에스오토메이션(주)는 그 책임을 면합니다.

알에스오토메이션(주)는 본 매뉴얼에 포함된 소프트웨어, 장비, 회로 또는 정보 사용에 대해 특허 책임을 면합니다.

사용자의 안전 및 효과적인 정보 전달을 위해 이 문서는 다음 기호들을 사용하고 있습니다

경고

주어진 정보 또는 절차를 잘못 취급할 경우, 사망, 중상 또는 재산상의 손실을 야기할 수 있음을 나타냅니다.

중요

주어진 정보가 제품을 이해하고 활용함에 있어 중요한 정보임을 나타냅니다.

주의

주어진 정보 또는 절차를 잘못 취급할 경우, 경상 또는 재산상의 손실을 야기할 수 있음을 나타냅니다. 그렇지만, 상황에 따라서 심각한 결과를 가져올 수도 있습니다.

목 차

1. 시스템 구성

2. 각부의 명칭과 기능,사양

3. 내· 외 번지 지정

4. 설치와 배선

5. 시운전과 이상시 조치사항

6. 명령어 규격

7. 제품일람 및 외형치수도

부록. 통신규약과 절차

안전에 관한 주의사항

반드시 지켜 주십시오.

설치, 운전, 보수, 점검에 앞서 반드시 이 매뉴얼을 읽고 올바르게 사용하여 주십시오.
기기에 대한 지식, 안전정보, 기타 주의사항을 모두 숙지하신 후 사용하시기 바랍니다.

⚠ 경고

잘못 취급하였을 경우 사용자가 사망 또는 중상을 입는 위험상태가 발생할 것으로 예상되는 경우

- 인체사고나 중대한 손해로 확대될 것으로 예측되는 용도로 사용하실 경우에는 이중 안전기구 등 안전대책을 세워 주십시오.
- 연소성 가스가 있는 곳에서는 사용하지 마십시오.
  폭발의 원인이 됩니다.

⚠ 주의

잘못 취급하였을 경우 사용자가 상해를 입거나 또는 물적 손해가 발생하는 위험상태가 발생할 것으로 상정되는 경우

- 비상정지, 인터록회로는 외부회로에서 구성해 주십시오.
- 전선은 단자나사로 확실히 조여 주십시오.
  접속이 불량일 경우 이상발열이나 고장의 원인이 됩니다.
- 정격사양, 환경등의 사양범위 이외에서는 사용하지 마십시오.
  이상발열이나 고장의 원인이 됩니다.
- 분해나 개조하지 마십시오.
  감전이나 고장의 원인이 됩니다.
- 전류가 흐르고 있는 동안에는 단자를 만지지 마십시오.
  감전의 우려가 있습니다.

■ NX70 PLC 설치환경

⚠ 주의 **이런환경은 피해야…**

1. 주변온도가 0~55℃를 넘는장소
2. 직사광선에 직접 노출된 장소
3. 습도가 30~85%를 넘는 장소
4. 전자부품에 영향을 주는 화공약품을 취급하는 장소
5. 지나치게 먼지, 염분이 많은 장소
6. 주변에 고전압, 강한자장, 강한전자파가 있는 장소
7. 충격, 진동이 심한 장소

⚠ 주의 **정전기에 관해서 .....**

건조한 장소에는 과대한 정전기가 발생할 가능성이 있으므로, 유니트에 접촉할 경우에는 어스된 금속성등에 접촉해서 정전기를 방전시켜 주십시오.

⚠ 주의 **청소에 관해서 .....**

신나류는 보드(PCB 기판)를 녹이거나 변색되거나 하므로 절대로 사용하지 마십시오.

⚠ 주의 **전원시퀀스에 관해서**

1. PLC의 기동은, 입출력 기기, 동력기기가 기동하고나서 수행해 주십시오.
   (예, PROG.모드에서 전원을 켜고나서, RUN모드로 전환 해 주십시오.)
2. PLC의 정지는, PLC 운전이 정지하고나서 입출력기기의 전원이 OFF되도록 해 주십시오.
3. PLC 본체가 운전중인 상태에서 입추력 기기의 전원을 ON/ OFF하면 PLC가 입력신호의 변화를 검출하고 오동작 할 수가 있습니다.

⚠ 주의 **전원입력전에 관해서**

처음으로 전원을 입력할때에는, 아래의 점에 주의해 주십시오.

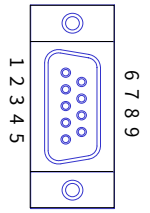
1. 시공시의 배선층, 특히 도전물이 붙어있지 않은가를 확인해 주십시오.
2. 전원배선, 입출력 배선, 전원전압이 틀리지 않은가를 확인해 주십시오.
3. 설치나사, 단자나사를 확실히 체결해 주십시오.
4. 모드전환스윗치를 PROG.모드로 해 주십시오.

■ NX70(NX70-CPU70p1, NX70-CPU70p2) CPU모듈 통신포트

- NX70-CPU70p1 모듈은 통신용 포트가 1개 준비되어 있습니다. (RS232C/ RS485 x 1포트)
- NX70-CPU70p2 모듈은 통신용 포트가 2개 준비되어 있습니다. (RS232C/ RS485 x 2포트)
- PLC Nplus-시리즈 (SPC10, N70plus, N700plus, NX7, NX70(CPU70p1, CPU70p2), NX700(CPU700p) PLC)는 프로그래밍 툴인 WinGPC S/W 를 사용하고 있습니다.

■ CPU모듈 통신용 케이블 결선도

● CPU모듈과 컴퓨터와의 접속



※ 당사제품인 CPL5530(2m), CPL5531(5m)을 접속하여 사용할 수 있습니다.

CPU 유니트측(9핀)

신호명	PIN No.
TXD	2
RXD	3
-	4
S.G	5
485(-)	6
485(+)	7
-	8
Vcc	9

컴퓨터측, 9핀

PIN No.	신호명
2	RXD
3	TXD
4	DTR
5	S.G
6	DSR
7	RTS
8	CTS
9	

■ NX70 PLC의 CPU 모듈종류

CPU 제품코드	제품사양	비 고	관련 메뉴얼	사용자 S/W
NX70-CPU70p1	• 9K Words (기본내장), 0.2μs/ STEP, RS232C/485 1포트, • FLASH ROM 내장 • 2K Data Register		NX70 (NX70-CPU70p1, p2) 시스템메뉴얼	WinGPC S/W
NX70-CPU70p2	• 20K Words (기본내장), 0.2μs/ STEP, RS232C/485 2포트, • 시계기능, FLASH ROM내장, PID기능, 사용자정의 통신 • 4K Data Register			

■ 특수유니트 사용제한

CPU 제품코드	사용불가 유니트	비 고
NX70-CPU70p1	ROM 버젼 2.0 이하 : CCU, W-Link, MW-Link, HSC 4Ch, Pulse4, 위치결정등 사용불가. ROM 버젼 2.0 이상 : CCU 유니트 사용불가.	
NX70-CPU70p2	CCU 유니트,	ROM 버젼 2.0 이상

*** CPU ROM버젼에 따라 기능이 제한될 수도 있습니다.**
*** CCU 유니트는 NX알파 시리즈 전용이며, NX plus시리즈는 CCU+ 기종을 사용합니다.**

■ NX70 PLC와 N70 PLC의 모듈 호환성

N70 PLC와 NX70 PLC는 CPU 모듈, 마더보드 모듈, POWER 모듈을 제외하고는 모든 모듈은 상호 호환성이 있습니다.

ex1) N70plus CPU(CPL9215)를 NX70 PLC의 마더보드(NX70-BASE02 NX70-BASE12)에 실장할 수는 없습니다.

ex2) N70의 POWER 모듈(CPL9631등)을 NX70 PLC의 마더보드(NX70-BASE02 NX70-BASE12)에 실장할 수는 없습니다.

ex3) N70의 마더보드(CPL9502 CPL9508)상에 Version-Up된 Analog 모듈은 실장 가능합니다.

ex4) NX70 PLC의 마더보드(NX70-BASE02 NX70-BASE12)상에 기존 N70의 I/O 및 특수모듈은 실장할 수 있습니다.

■ 참고 사항

● PLC는 N-시리즈와 N-plus 시리즈가 있습니다. (명령어 및 사용자 S/W가 다르므로 주의바랍니다.)

● 특히, N70 시리즈, N700 시리즈, NX70 시리즈, NX700 시리즈는 CPU 모듈외 모든 모듈(POWER, I/O, 마더보드, 특수모듈...)을 공통으로 사용하고 있습니다.

시리즈명	제품명	CPU 모듈명	비 고
N-plus, NX-plus 시리즈	**SPC 시리즈**	**SPC10, SPC24S, SPC120S, SPC100, SPC300, A200 등**	**WinGPC S/W 사용**
	N70 시리즈	**CPL9215A, CPL9216A**	
	N700 시리즈	**CPL7215A**	
	NX7 시리즈	**NX7-28ADR, NX7-28ADT, NX7-48ADR, NX7-48ADR.....**	
	NX70 시리즈	**NX70-CPU70p1, NX70-CPU70p2**	
	NX700 시리즈	**NX-CPU700p**	

N-시리즈, NX-알파 시리즈	**N7 시리즈**	CPL02323, CPL02343C, CPL02543C.....	**WinFPST S/W 사용**
	N70 시리즈	CPL9210A, CPL9211A	
	N700 시리즈	CPL7210A, CPL7211A, CPL6210A, CPL6210B, CPL6215A	
	NX70 시리즈	NX70-CPU70, NX70-CPU750	
	NX700 시리즈	NX-CPU750A, NX-CPU750B, NX-CPU750C, NX-CPU750D, NX-CPU700	

1장

시스템 구성

1-1. CPU70plus(NX70) PLC의 특징 및 시스템 구성

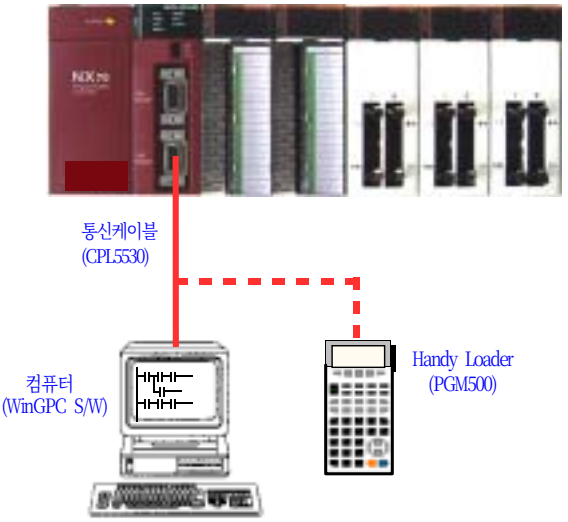
(1) 시스템 접속

NX70-CPU70p1

- 9.6k STEP 기본메모리
- 1 port (RS232/ RS485)

NX70-CPU70p2

- 20k STEP 기본메모리
- 2 port (RS232/ 485)



(2) CPU70plus(NX70) PLC의 특징

1. 명령어 처리속도

고속처리용 전용 IC를 채용하여 기본 명령어 처리속도가 0.2μ sec/STEP 으로 고속처리가 가능합니다.

2. RUN중 프로그램 수정

CPU 운전중에도 시스템을 중단하지 않고 프로그램을 수정할 수 있습니다.

3. Clock(RTC : Real Time Clock) function

내장형 시계를 이용하여 공정을 시간, 날짜별로 프로그램 할 수 있습니다 (NX70-CPU70p2 모듈에서만 지원)

4. 충분한 프로그램 용량/백업 기능

최대20k 워드(NX70-CPU70p2), 9.6k 워드(NX70-CPU70p1)까지 프로그램이 가능하고, FLASH 백업방식(EEPROM이 기본내장)을 채택하여 프로그램을 별도로 보관할 수 있도록 하였습니다.

5. 자기진단 기능

편리한 자기진단 기능이 있어 시스템 에러 및 고장진단에 편리합니다.

6. 최대 384점까지 제어가능

7. 다양한 입출력 및 특수모듈 지원

DC24V 입력(16/32점), AC110V 입력, AC220V 입력, 릴레이 출력, TR 출력(16점/32점), SSR출력, A/D(4CH), D/A(2CH, 4CH), RTD(4CH), TC(4CH), 고속카운터, SCU(시리얼 통신장치), MW-Link, 위치결정, 펄스출력등을 처리할 수 있습니다.

8. WinGPC 소프트웨어 사용

CPU제어를 위한 전용 그래픽 소프트웨어인 WinGPC S/W를 이용하여, CPU 상태 체크 및 강제입출력, 입출력 모니터링, 프로그램 다운로드/업로드 등이 편리합니다.

9. 다양한 마더보드 준비 (12슬롯 마더보드까지)

시스템을 구성할때 각각 2슬롯, 3슬롯, 5슬롯, 6슬롯, 8슬롯, 10슬롯, 12슬롯 마더보드를 선택하여 구축할 수 있습니다.

10. RS232C/ RS485, 2포트 탑재 (NX70-CPU70p2 모듈)

2채널의 통신 포트 내장으로 컴퓨터 및 TOUCH PANEL과 직접 접속, 고속으로 대용량의 DATA교환이 가능하며, **COM2포트는 사용자정의 통신방식을 지원**하여 바코드 리더, 인버터, 서보를 통신으로 연결,제어할 수 있습니다. (바이너리 통신 가능)

(3) 슬롯별 기본구성 및 I/O 점수

■ 2모듈 시스템

(NX70-BASE02)

32점: 16점 I/O 구성시
64점: 32점 I/O 구성시

■ 3모듈 시스템

(NX70-BASE03)

48점: 16점 I/O 구성시
96점: 32점 I/O 구성시

■ 5모듈 시스템

(NX70-BASE05)

80점: 16점 I/O 구성시
160점: 32점 I/O 구성시

■ 6모듈 시스템

(NX70-BASE06)

96점: 16점 I/O 구성시
192점: 32점 I/O 구성시

■ 8모듈 시스템 (NX70-BASE08)

128점: 16점 I/O 구성시
256점: 32점 I/O 구성시

■ 10모듈 시스템 (NX70-BASE10)

160점: 16점 I/O 구성시
320점: 32점 I/O 구성시

■ 12모듈 시스템 (NX70-BASE12)

192점: 16점 I/O 구성시
384점: 32점 I/O 구성시

■ 자유로운 시스템 구성 : 마더보드 종류는 7종 보유 (2슬롯, 3슬롯, 5슬롯, 6슬롯, 8슬롯, 10슬롯, 12슬롯)

- NX70 PLC에는 7종류의 마더보드가 있으며, 각종 입출력 유니트를 자유롭게 장착할 수 있습니다.
  (7종 : 2, 3, 5, 6, 8, 10, 12모듈 타입)
- I/O 모듈, 전원모듈, 특수모듈, 마더보드는 CPU 기종에 관계없이 공통입니다.

■ 최대 제어 I/O 점수 384점

- NX70 PLC는 12 슬롯 마더보드를 사용할 경우, 최대 384점(터미널블럭은 192점)까지 사용할 수 있습니다.
- NX70 PLC는 증설을 사용할 수 없습니다. (기본 마더보드밖에 없습니다.)

■ NX70 PLC와 N70 PLC의 모듈 호환성

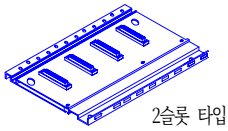
N70 PLC와 NX70 PLC는 CPU 모듈, 마더보드 모듈, POWER 모듈을 제외하고는 모든 모듈은 상호 호환성이 있습니다.

ex1) N70plus CPU(CPL9215)를 NX70 PLC의 마더보드(NX70-BASE02 NX70-BASE12)에 실장할 수는 없습니다.
ex2) N70의 POWER 모듈(CPL9631등)을 NX70 PLC의 마더보드(NX70-BASE02 NX70-BASE12)에 실장할 수는 없습니다.
ex3) N70의 마더보드(CPL9502 CPL9508)상에 Version-Up된 Analog 모듈은 실장가능합니다.
ex4) NX70 PLC의 마더보드(NX70-BASE02 NX70-BASE12)상에 기존 N70의 I/O 및 특수모듈은 실장할 수 있습니다.

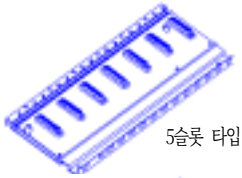
1-2. 유니트(모듈) 종류와의 조합

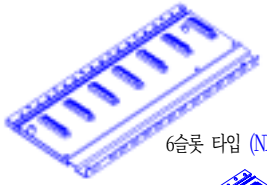
(1) 마더보드와 유니트 조합일람

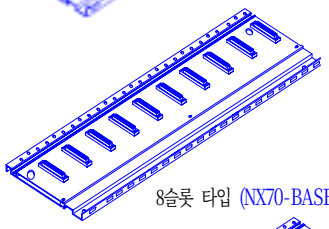
【마더보드】

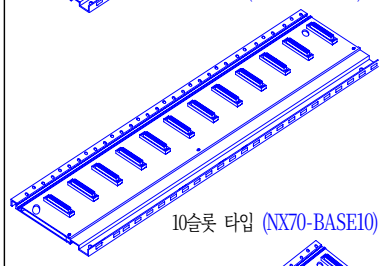

2슬롯 타입 (NX70-BASE02)

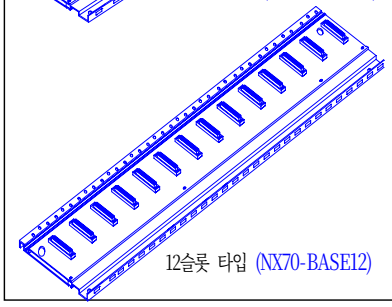

3슬롯 타입 (NX70-BASE03)


5슬롯 타입 (NX70-BASE05)


6슬롯 타입 (NX70-BASE06)


8슬롯 타입 (NX70-BASE08)


10슬롯 타입 (NX70-BASE10)


12슬롯 타입 (NX70-BASE12)

【CPU 유니트】

NX70-CPU70p1

- 9.6k STEP 기본메모리
- 1 port (RS232/ RS485)

NX70-CPU70p2

- 20k STEP 기본메모리
- 2 port (RS232/ 485)

【전원유니트】

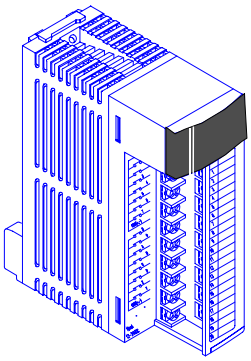
NX70-POWER1
AC110-220V Free Voltage 입력
5V 3.5A, 24V 0.3A

NX70-POWER2
AC110-220V Free Voltage 입력
5V 4.5A

NX70-PWRDC
DC24V 입력
5V 4.5A

★표는 ROM Version 2.0 이상에서 지원됩니다.

【입력유니트/ 출력유니트】

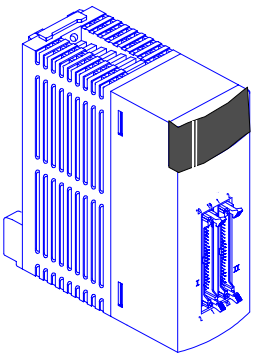


■ 16점 입력모듈
- 24V DC IN (NX70-X16D) (NX70-X16D1)
- 110V AC IN (NX70-X16A110)
- 220V AC IN (NX70-X16A220)

■ 16점 출력모듈
- Relay OUT (NX70-Y16R) (NX70-Y16RV)
- TR OUT (NX70-Y16T)
- SSR OUT (NX70-Y16SSR)

■ 8점 출력모듈
- Relay OUT (NX70-Y8R)

■ 16점 입출력 혼합모듈
- DC 8IN/Ry 8out (NX70-XY16)

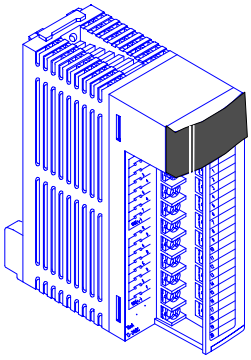


■ 32점 입력모듈
- 24V DC IN (NX70-X32D) (NX70-X32D1)

■ 32점 출력모듈
- TR OUT (NX70-Y32T) (NX70-Y32P)

■ 32점 입출력 혼합모듈
- DC 16IN/TR 16out (NX70-XY32)

【특수유니트】



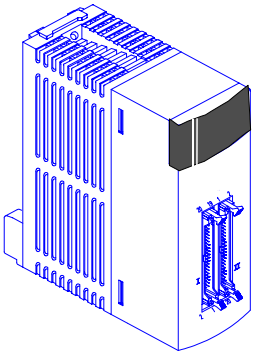
■ Analog 입력모듈
- 8CH, 전류입력 (NX70-AI8C)
- 8CH, 전압입력 (NX70-AI8V)
- 4CH, 전류입력 (NX70-AI4C)
- 4CH, 전압입력 (NX70-AI4V)

■ Analog 출력모듈
- 4CH, 전류출력 (NX70-AO4C)
- 4CH, 전압출력 (NX70-AO4V)
- 2CH, 전류출력 (NX70-AO2C)
- 2CH, 전압출력 (NX70-AO2V)

■ RTD 입력모듈
- 4CH, RTD (NX70-RTD4)

■ TC 입력모듈
- 4CH, TC (NX70-TC4)

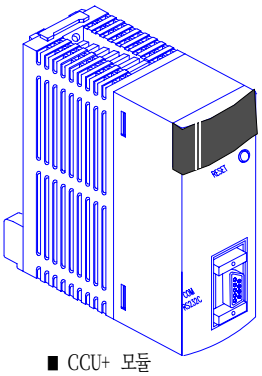
■ 고속카운터 모듈1
- 2CH, HSC (NX70-HSC2)
- 1CH, HSC (NX70-HSC1)



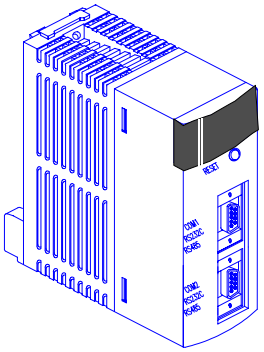
★ ■ 고속카운터 모듈2
- 4CH, HSC (NX70-HSC4)

★ ■ 펄스출력 모듈
- 4CH,Pulse (NX70-PULSE4)

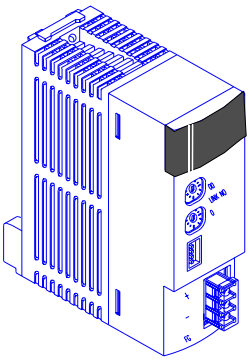
★ ■ 위치결정 모듈
- 2CH, 4Ch (NX70-POSI2) (NX70-POSI4)



■ CCU+ 모듈
(NX70-CCU+)



■ SCU 모듈
(NX70-SCU)

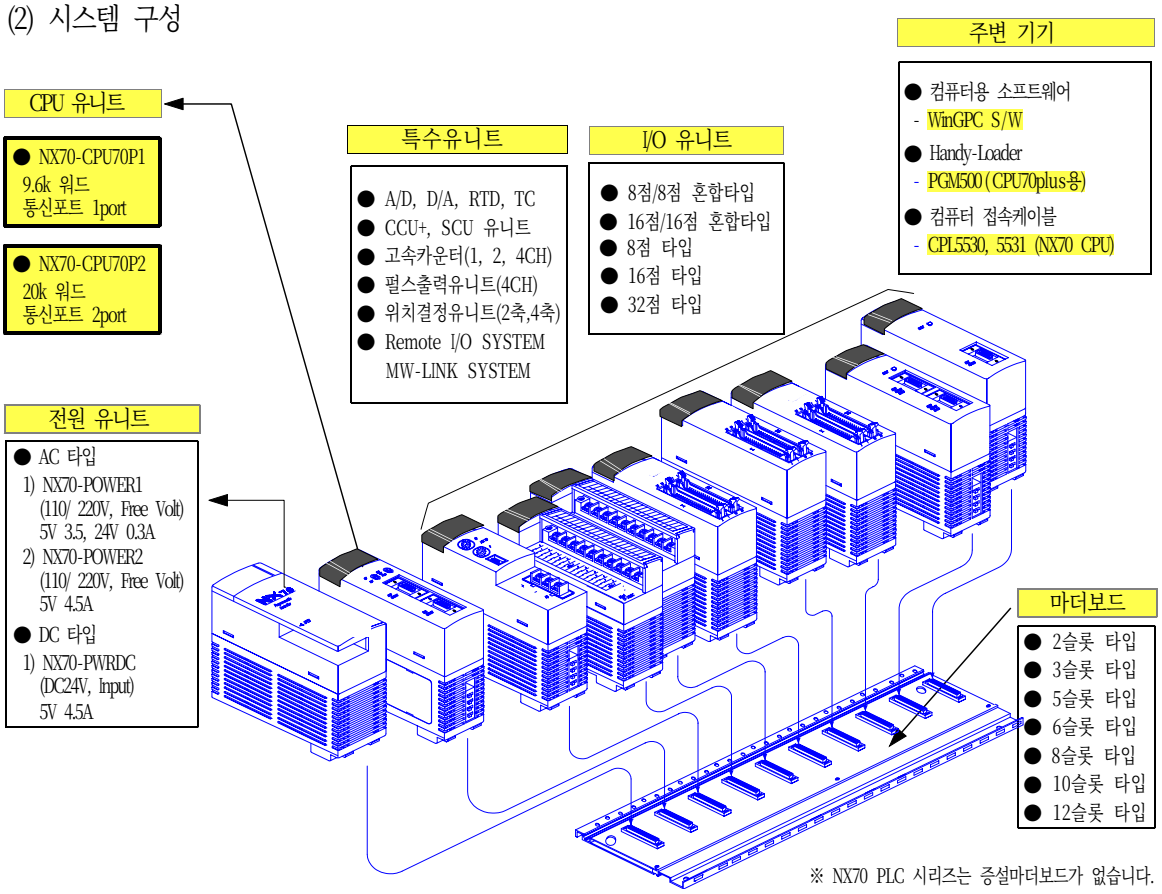


■ MW-LINK 모듈 ★
(NX70-MWLINK)
Multi W-Link (MW-Link)
- W-Link 기능

★ ■ W-LINK 모듈
(NX70-MWLINK)

■ Slave 모듈
(NX70-SLAVE)

(2) 시스템 구성



※ NX70 PLC 시리즈는 증설마더보드가 없습니다.

■ 유니트 종류에 의한 제한

○ : 사용가능 △ : 버젼에 따라 사용가능

CPU 유니트 종류 \ 일반 유니트 종류		기본 유니트					특수 유니트							내트워크 유니트			비 고
		마더보드 (2, 3, 5, 6, 8, 10, 12 슬롯) (NX70-BASE02, 03, 05, 06, 08, 10, 12)	AC 전원 유니트 (전 슬롯용) (NX70-POWER1, NX70-POWER2)	DC 전원 유니트 (전 슬롯용) (NX70-PWRDC)	입력 유니트	출력 유니트	A/D, D/A, RTD, TC 유니트	고속카운터 유니트(1CH / 2CH)	고성능 고속카운터 유니트(4CH)	펄스출력 유니트(4CH)	위치결정 유니트(2축, 4축)	SCU 유니트 (2CH)	CCU+ 유니트 (1CH)	W-Link 유니트 (W- 모드)	MW-Link 유니트 (W- 모드, W2- 모드)	Remote Slave 유니트	
일반 CPU	NX70-CPU70p1	○	○	○	○	○	○	○	△	△	△	○	△	△	△	○	
고급 CPU	NX70-CPU70p2	○	○	○	○	○	○	○	△	△	△	○	△	○	△	○	

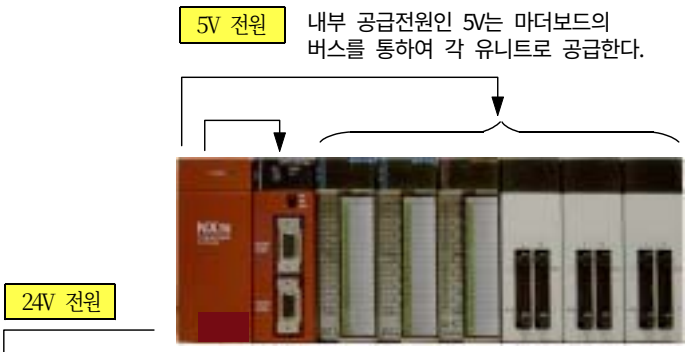
※ △ 표시는 ROM 버젼 2.0이상에서 지원됩니다.

(3) 사용가능한 조합과 제한

※1. 네트워크 유니트의 실장의 제한

유니트 종류	NX70-CPU70p1, NX70-CPU70p2 모듈일 경우	비 고
CCU+ 유니트	ROM 2.0 이상 사용 1대만 사용가능	
W-Link 유니트	최대 2대까지 사용가능	
MW-Link 유니트 (W-모드)	사용가능	

(4) 소비전류에 의한 제한



전원유니트	정격 출력전류	
	5V	24V
NX70-POWER1	3.5A	0.3A
NX70-POWER2	4.5A	없음
NX70-PWRDC	4.5A	없음

주1) 전원유니트에서 제공되는 24V 출력 전원은, 24V를 필요로 하는 각 입출력유니트용으로 사용할 수 있지만, 전류용량이 제한되어 있으므로 24V 전원은 시중에서 구입 사용할 것을 권장합니다.
(NX70-POWER1 : 24V 0.3A)
(NX70-POWER2 : 24V 출력 없음)

주2) 입출력유니트로의 24V 공급은,
① 전원유니트의 24V 전원과
② 시중에서 판매되는 전원중
어느것이나 사용할 수 있지만,
병렬접속은 하지마십시오

■ 내부 공급전원과 외부 공급전원

내부 공급전원 (5V)

각 유니트의 내부회로 구동용으로 사용되는 5V전원은, 전원유니트에서 마더보드의 내부버스를 통하여 공급됩니다. (별도공급 불필요)

외부 공급전원 (24V)

- 입력유니트의 입력용 전원, 그리고 출력유니트의 출력 회로 구동용 전원등으로 사용되는 24V 전원은 외부단자로부터 공급해야 합니다.
- 24V 전원은, 전원유니트의 서비스전원 또는 시판전원을 사용하면 됩니다.
- 전원이 병렬운전(자체 공급전원과 외부에서 구입한 시판전원) 되지 않도록 해야 합니다.
- 단, NX70-POWER2 및 NX70-PWRDC는 24V 서비스 전원이 없습니다. (5V 출력만 가능)

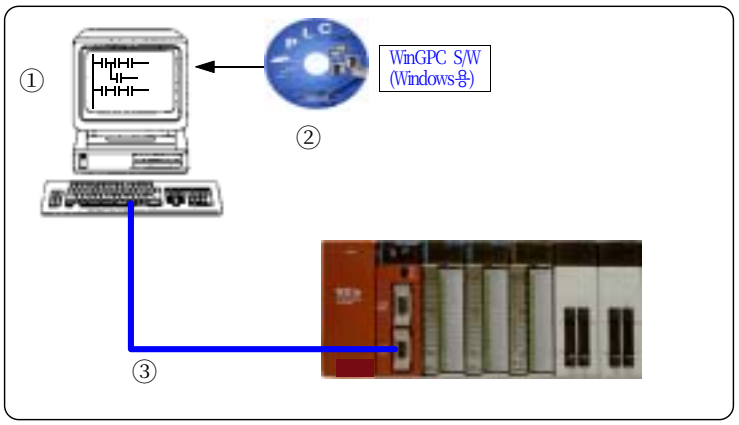
■ 각 유니트에 소비되는 전류용량

각 유니트에서 소비되는 전원용량은, 5V, 24V 사용전원의 정격용량을 넘지 않도록 주의하십시오.

1-3. 프로그래밍 Tool

(1) 프로그래밍에 필요한 도구(Tool)

WinGPC S/W를 사용하여 프로그래밍 할때 다음과 같은 도구가 요구됩니다.



① 컴퓨터

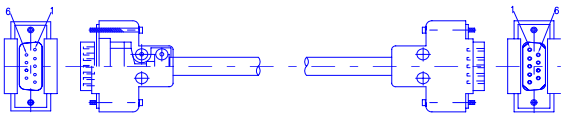
② WinGPC S/W (Windows용)

PLC Nplus-시리즈의 통합 소프트웨어로서, 프로그램 편집 · 디버그용 소프트웨어입니다.
WinGPC S/W는 Windows용으로 준비되어 있습니다. (현재, WinGPC S/W Ver 3.**입니다.)

※ PLC Nplus-시리즈란, SPC10, N70plus, N700plus, NX7, NX70(CPU70p1, CPU70p2), NX700(CPU700p) PLC CPU 기종을 말합니다.

③ 프로그래밍용 케이블

CPU70plus(NX70) CPU 모듈의 통신포트와 접속시 사용되는 프로그래밍 케이블은 당사 판매품(CPL5530, CPL5531)을 구매하여 사용할 수 있으나, 아래와 같은 통신 사양으로 사용자가 직접 조립 할 수도 있습니다.



CPU 유니트측(9핀)-수컷

신호명	PIN No.
TXD	2
RXD	3
-	4
S.GND	5
485(-)	6
485(+)	7
-	8
Vcc	9

컴퓨터측, 9핀-암컷

PIN No.	신호명
2	RXD
3	TXD
4	DTR
5	GND
6	DSR
7	RTS
8	CTS
9	

2 장

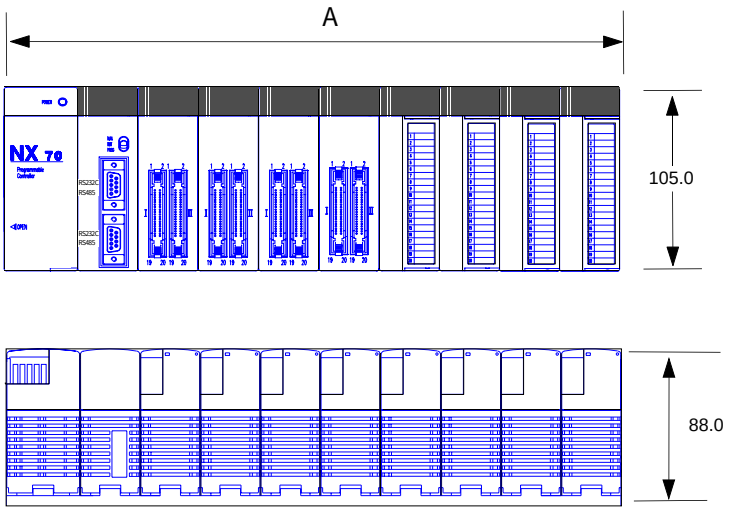
각부의 명칭과 기능·사양

2-1. 전체 사양

(1) 일반 사양

항 목		사 양
주변 온도	사용 온도	0 55℃
	보존 온도	-25 70℃
주변 습도	사용 습도	30 85% RH (단, 이슬이 없을때)
	보존 습도	30 85% RH (단, 이슬이 없을때)
내 전 압		AC 외부단자 <-> 어스 사이, AC 1500V 1분간
		DC 외부단자 <-> 어스 사이, AC 500V 1분간
절연 저항		외부단자 <-> 어스 사이, 100MΩ 이상(DC 500V메가에서)
내 진 동		10 55Hz 1掃引/ 1분간, 복진폭 0.75mm, X, Y, Z 각 방향 10분간
내 충 격		98 m/s² 이상 , X, Y, Z 각방향 4회
내 노이즈성		1500 Vp-p 펄스폭 50ns, 1μs (노이즈 시뮬레이터법에 의함)
사 용 환 경		IP20 (부식성 가스가 없을것 , 먼지가 심하지 않을것)

(2) 외형 치수도 (단위 mm)



단위 (mm)

	2슬롯 타입	3슬롯 타입	5슬롯 타입	6슬롯 타입	8슬롯 타입	10슬롯 타입	12슬롯 타입
	NX70-BASE02	NX70-BASE03	NX70-BASE05	NX70-BASE06	NX70-BASE08	NX70-BASE10	NX70-BASE12
A(mm)	149.5	185.0	256.0	291.5	362.5	398.0	433.5

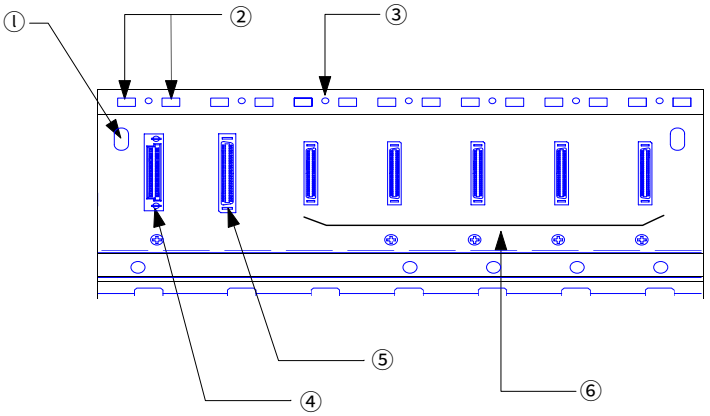
(3) 성능 사양

CPU 명		NX70-CPU70p1	NX70-CPU70p2
제 어 방 식		프로그램 저장, 반복연산 방식	
외부입출력	디지탈	384점 (32점 모듈 / 12 Slots)	
	아날로그	48채널 (4채널 모듈 / 12 Slots)	
명령어	기본명령	30종	
	응용명령	147 여종	
처리속도	기본명령	0.2 μs/step	
	응용명령	1.0 ~ 수십 μs/step	
프로그램 용량		9.6K 워드	20K 워드
메모리 용 량	입출력(R)	2,048점 (R0.0 ~ R127.15)	
	링크 접점 (L)	1,024점 (L0.0 ~ L63.15)	
	내부 접점 (M)	2,048점 (M0.0 ~ M127.15), (단, NX70-CPU70p2에서는 링크접점 영역으로 사용가능, 64워드)	
	정전유지 (K)	2,048점 (K0.0 ~ M127.15)	
	특수접점 (F)	256점 (F0.0 ~ F15.15)	
	타이머/ 카운터 (TC 또는 TIM)	256채널 (타이머+카운터), 설정치 : 0 ~65535 타이머 : 0.01초 : CH000~CH063 (64채널) , 0.1초 : CH064~CH255 (192채널) 카운터 : CH000 ~ CH255 (256채널)	
	데이터 워드 (W)	2,048워드 (W0000 ~ W2047)	4,096워드 (W0000~ W2047), (W3072~ W5119)
	특수워드 (W, SR)	512워드 W2560(=SR000) ~ W3071(=SR511)	
시계기능 (RTC)		없 음	년, 월, 일, 시, 분. 초, 요일
통신기능	Port 1	RS232C/ RS485 겸용, 4800/ 9600/ 19200/ 38400bps	
	Port 2	없 음	RS232C/ RS485 겸용, 4800 38400bps 사용자정의 지원가능
FLASH ROM 지원 (BACK-UP)		모두 지원 (CPU 모듈에 내장되어 있음)	

- K, W, Counter PV Register는 정전유지됩니다.
- CPU유니트는 Battery가 꺼져도 내부 Super Capacitor에 의해 48시간 동안 사용자 프로그램과 특정 Register들을 Back-Up할 수 있습니다.

2-2. 마더보드

(NX70-BASE02, NX70-BASE03, NX70-BASE05, NX70-BASE06, NX70-BASE08, NX70-BASE10, X70-BASE12)



5슬롯 마더보드의 예 (NX70-BASE05)

■ 각부의 명칭과 기능

① 마더보드 설치구멍

마더보드를 제어판(콘트롤 박스) 등에 설치하기 위한 구멍입니다. 설치시 M5 나사를 사용해 주십시오.

② 유니트 가이드 구멍

각각의 유니트를 마더보드에 설치할 때, 이 구멍과 유니트의 돌기를 맞추어 주십시오.

※ 좌측부터 전원유니트, CPU유니트, I/O(특수)유니트 순으로 설치해 주십시오.

③ 유니트 설치구멍

각종 유니트와 마더보드를 고정하기 위한 구멍이며 고정방법은 각 유니트에 붙어있는 나사로 체결해 주십시오.

④ 전원유니트용 커넥터

⑤ CPU 유니트용 커넥터

CPU 유니트를 설치합니다. CPU유니트는 반드시 전원 유니트옆에 설치하십시오.

⑥ I/O 유니트 (또는 특수유니트) 모듈용 커넥터

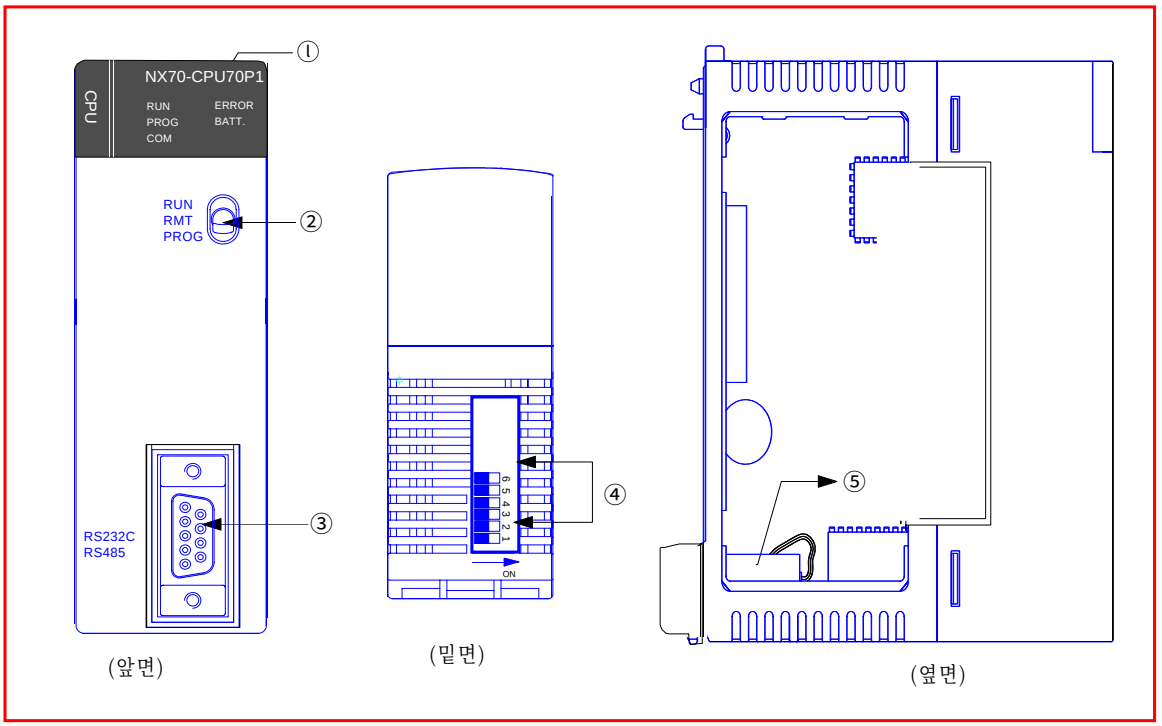
I/O(특수) 유니트를 설치합니다.

■ 마더보드의 종류

슬롯 수	제품 번호	비 고
2	NX70-BASE02	
3	NX70-BASE03	
5	NX70-BASE05	
6	NX70-BASE06	
8	NX70-BASE08	
10	NX70-BASE10	
12	NX70-BASE12	

2-3. CPU 유니트

(1-1) CPU 유니트 (NX70-CPU70p1)



■ 각부의 명칭과 기능

① 상태표시 LED

PLC의 운전/ 정지, 에러/ 알람상황등 동작상태를 표시합니다.

② 모드 전환스위치

PLC의 운전모드를 전환하는 스위치입니다.

③ 통신포트 COM (RS232C/ RS485 통신), 9핀

프로그래밍 툴(WinGPC S/W) 및 T/P 등의 접속이 가능한 통신포트입니다.

④ 동작모드 설정스윗치

DIP S/W (6폴, 종단저항 설정 스윗치 및 통신방식, 프로그램 부팅용 스윗치) 입니다.

⑤ 메모리 백업용 전지

내장 메모리(RAM)의 백업용 전지입니다.
출하시에는 커넥터가 접속되어 있지 않습니다.

■ 상태표시 LED

상태 표시 LED		
LED	색깔	기 능
RUN	녹	CPU가 RUN 일때 점등
PROG.	녹	Program 변경가능 상태때 점등
COMM.	녹	CPU가 통신중일때 깜박거림.
ERROR	적	CPU가 Error발생시 점등
BATT.	적	Battery전압이 낮거나 미장착시 점등

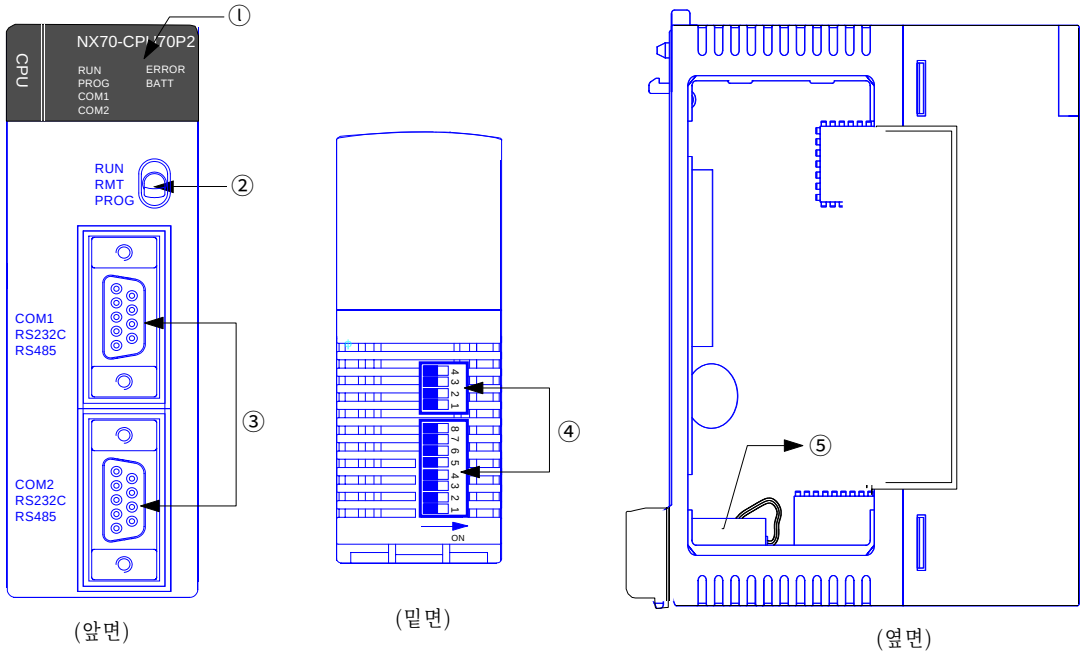
■ 모드전환 스윗치

상태	기 능
RUN	CPU를 RUN Mode로 함.
REMOTE	CPU를 RUN 또는 PROG Mode로 함.
PROG.	CPU를 Stop, 즉 Program모드로 변경

■ 동작모드 설정스윗치

PIN번호		스위치		기 능	Dip Switch	비 고
6	5	OFF	OFF	9600 bps 통신 속도 설정		통신방식/ 프로그램 부팅 설정용 스위치
		ON	OFF	19200 bps 통신 속도 설정		
		OFF	ON	38400 bps 통신 속도 설정		
		ON	ON	4800 bps 통신 속도 설정		
4		OFF		RS-232C 통신 방식설정		
		ON		RS-485 통신방식 설정		
3		OFF		내장 RAM에 저장된 프로그램으로 운전시		
		ON		FLASH ROM에 저장된 프로그램으로 운전시		
2, 1		ON	ON	RS-485 통신시 : 종단국일때 셋트 (RS-485방식의 통신사용시, 종단국일때 셋트함)		종단저항 설정 스위치
		OFF	OFF	RS-485통신시 : 종단국이 아닐때 셋트 (RS-485방식의 통신사용시, 종단국이 아닐때 셋트함)		

(1-2) CPU 유니트 (NX70-CPU70p2)



■ 각부의 명칭과 기능

① 상태표시 LED

PLC의 운전/ 정지, 에러/ 알람상황등 동작상태를 표시합니다.

② 모드 전환스위치

PLC의 운전모드를 전환하는 스위치입니다.

③ 통신포트 COM1, COM2 (RS232C/ RS485 통신), 9핀

프로그래밍 툴(WinGPC S/W) 및 T/P 및 MMI, 사용자 정의 통신(COM2 포트)이 가능한 통신 포트입니다.

④ 동작모드 설정스윗치

DIP S/W 1(4폴, 종단저항 설정 스윗치),
DIP S/W 2(8폴, 통신방식, 프로그램 부팅용 스윗치) 입니다.

⑤ 메모리 백업용 전지

내장 메모리(RAM)의 백업용 전지입니다.
출하시에는 커넥터가 접속되어 있지 않습니다.

■ 상태표시 LED

LED	색깔	기 능
RUN	녹	CPU가 RUN 일때 점등
PROG.	녹	Program 변경가능 상태때 점등
COM1	녹	CPU가 COM1과 통신중일때 깜빡거림
COM2	녹	CPU가 COM2와 통신중일때 깜빡거림
ERROR	적	CPU가 Error발생시 점등
BATT	적	내부 밧데리에 문제발생시 점등

■ 모드전환 스윗치

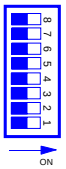
상태	기 능
RUN	CPU를 RUN Mode로 함.
REMOTE	CPU를 RUN 또는 PROG Mode로 함.
PROG.	CPU를 Stop, 즉 Program모드로 변경

■ 동작모드 설정스윗치

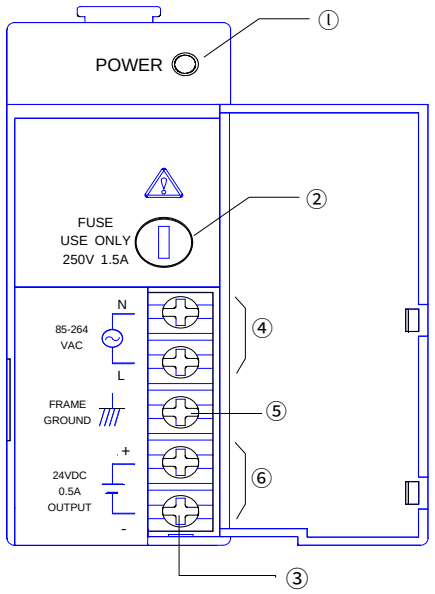
● 종단저항 설정 스위치 ("DIP 스위치 1“)

PIN번호	스위치		기 능	Dip Switch1
3, 4	ON	ON	RS-485 통신시 : 종단국일때 셋트 (COM1 단자에 종단 저항 설정)	
	OFF	OFF	RS-485통신시 : 종단국이 아닐때 셋트 (COM1 단자에 종단 저항 미설정)	
1, 2	ON	ON	RS-485 통신시 : 종단국일때 셋트 (COM2 단자에 종단 저항 설정)	
	OFF	OFF	RS-485통신시 : 종단국이 아닐때 셋트 (COM2 단자에 종단 저항 미설정)	

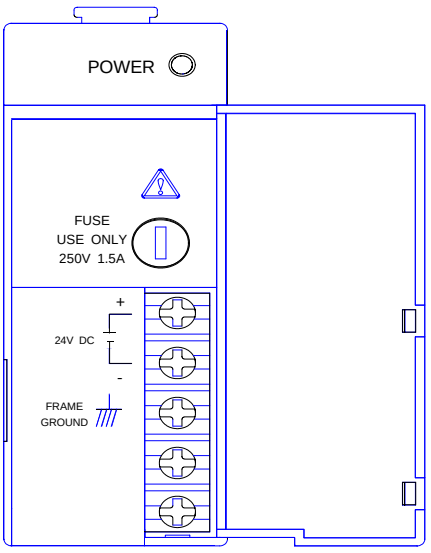
● 통신방식/ 프로그램부팅 설정용 스위치 ("DIP 스위치 2“)

PIN 번호	스위치		기 능	Dip Switch2
8,7	OFF	OFF	COM2단자 통신속도 9600 bps 설정	
	ON	OFF	COM2단자 통신속도 19200 bps 설정	
	OFF	ON	COM2단자 통신속도 38400 bps 설정	
	ON	ON	COM2단자 통신속도 4800 bps 설정	
6,5	OFF	OFF	COM1단자 통신속도 9600 bps 설정	
	ON	OFF	COM1단자 통신속도 19200 bps 설정	
	OFF	ON	COM1단자 통신속도 38400 bps 설정	
	ON	ON	COM1단자 통신속도 4800 bps 설정	
4	ON		COM1의 RS-485 방식 선택	
	OFF		COM1의 RS-232C 방식 선택	
3	ON		COM2의 RS-485 방식 선택	
	OFF		COM2의 RS-232C 방식 선택	
2	OFF		항상 OFF로 고정 시킴 (시스템 설정용)	
1	ON		전원 ON시 EEPROM (Flash ROM)에서 프로그램 Load.	
	OFF		전원 ON시 RAM으로 프로그램 운전	

2-4. 전원유니트



NX70-POWER1
NX70-POWER2



NX70-PWRDC

■ 각부의 명칭과 기능

① POWER 표시 LED
전원이 공급되고 있을때 켜집니다.

② 전원 퓨즈홀더

③ 단자대
전원배선용 단자대입니다. M3.5용의 압착단자를 사용할 수 있습니다.

④ 전원 입력단자
AC110-240V Free Voltage 전원입력 단자입니다. (단, NX70-PWRDC는 DC24V 입력 전용입니다.)

⑤ Frame Ground (FG) 단자
대지 접지단자로 마더보드의 금속부와 접속되어 있습니다. 감전방지를 위해 제3종 접지를 해 주십시오.

⑥ 서비스 전원 단자 (DC24V)
입출력유니트등의 DC전원 공급용으로 사용합니다.

주의 이 서비스 전원단자는 시중 전원기기와 병렬로 접속하지 마십시오. 에러의 원인이 됩니다.

■ 사양 일람

제품 번호	NX70-POWER1	NX70-POWER2
입력 정격전압	AC 110 220V, Free Voltage	
허용 전압범위	AC 85 264V	
입력 전원주파수	47 63Hz	47 63Hz
돌입 전류	20A 이하	20A 이하
정격 출력전류 (5V)	5V 3.5A	5V 4.5A
정격 출력전류 (24V)	24V 0.3A	없 음

제품 번호	NX70-PWRDC
입력 정격전압	DC 24V
허용 전압범위	DC 21.6 26.4V
정격 출력전류 (5V)	5V 4.5A

주의 ※ NX70-POWRER1, NX70-POWER2 모듈은, AC110V에서는 순간정전을 보증할 수 없습니다.

■ 사용상의 주의사항

전원 유니트는 마더보드에 장착되어 있는 모든 유니트에 5V 전원을 공급합니다.
따라서 각 유니트의 소비전류(5V)와 서비스 전원인 24V를 부하에 맞게 사용해 주십시오.

주의 사항

(1) 각 마더보드에 장착하는 유니트의 소비전류가 전원유니트의 용량을 초과하지 않도록 전원유니트를 사용해 주십시오.

(2) 5V 전원은 그 유니트가 장착되어 있는 마더보드의 전원유니트에서 공급됩니다.

(3) 서비스 전원인 24V전원은 입출력 유니트의 DC전원용으로 사용합니다.
단, 서비스전원인 자체 24V와 시중 전원기기와 병렬로 접속치 마십시오. 에러의 원인이 됩니다.

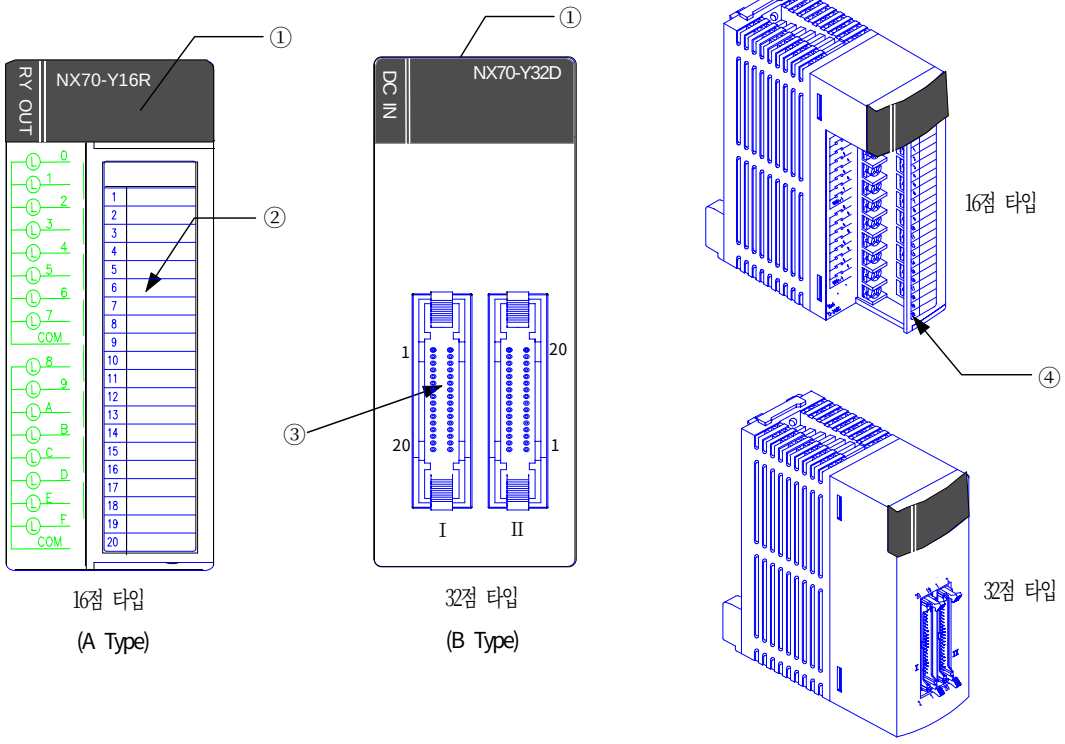
(4) AC 전원용 FUSE는 250V 1A의 FUSE를 사용하고 있습니다. (퓨즈는 전원유니트 내부에 내장되어 있습니다)

(5) 또한 외부단자에 있는 24V측은 전류용량을 초과하지 않도록 사용해야 합니다.
(시스템 에러를 유발할 수 있습니다.)

(6) NX70-POWRER1, NX70-POWER2 모듈은, AC110V에서는 순간정전을 보증할 수 없습니다.

2-5. 입력, 출력유니트

(1) 각부의 명칭과 기능



① 입출력 표시 LED

입출력의 ON/ OFF상태를 나타냅니다.

② 단자대 (20P)

입출력 및 전원배선용 단자대입니다.
M3.5용의 압착단자를 사용할 수 있습니다.
「4-4. 단자대 TYPE의 배선」을 참조하십시오.

③ 커넥터 32점 (20핀 x 2)

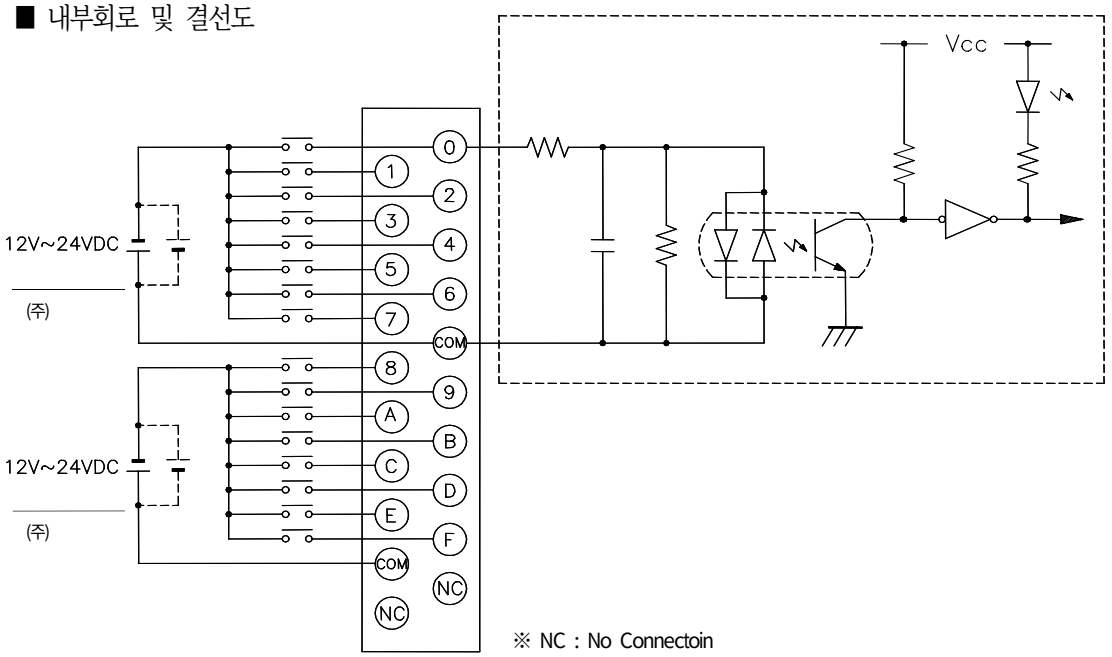
입출력 접점 및 전원배선용 커넥터입니다.
PIN 타입 또는 플랫케이블용 하네스를 사용해 주십시오. 상세한 것은 「4-5. 커넥터 TYPE의 배선」을 참조하십시오.

④ 단자대 커버 (COVER)

(2) 입력유니트 사양

제품 명칭		DC 입력유니트	
제품 코드		NX70-X16D (CPL93023)	NX70-X16D1 (CPL93033)
입력 점수		16점	
절연 방식		포토커플러	
정격 입력전압		12 ~ 24V DC	24V DC
사용 전압범위		10.2 ~ 26.4V DC	21.6V ~ 26.4V DC
최대 입력전류		10mA 이하	
ON 전압		9.6V 이상	20V 이상
OFF 전압		2.5V 이하	7V 이하
입력임피던스		약 3KΩ	
응답 시간	OFF → ON	2ms 이하	
	ON → OFF	2ms 이하	
내부소비전류 (5V)		50mA 이하	
COMMON 방식		8점/ 1COM (극성+,− 공통)	
동작 표시		LED 표시	
외선 접속방식		단자대 접속 (M3.0)	
적합 전선사이즈		0.5 1.25 mm²	
중 량		약 160g	
형 상		(A) Type	

■ 내부회로 및 결선도



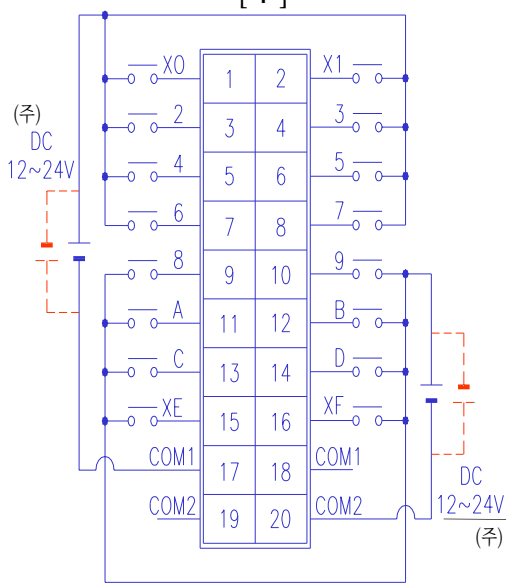
(주) NX70-X16D1 유니트는 입력전압이 24V DC입니다.

제품 명칭		DC 입력유니트	
제품 코드		NX70-X32D (CPL93024)	NX70-X32D1 (CPL93034)
입력 점수		32점	
절연 방식		포토커플러	
정격 입력전압		12 ~ 24V DC	24V DC
사용 전압범위		10.2 ~ 26.4V DC	21.6V ~ 26.4V DC
최대 입력전류		10mA 이하	
ON전압 / ON 전류		9.6V 이상	20V 이상
OFF전압 / OFF전류		2.5V 이하	7 V 이하
입력임피던스		약 3KΩ	
응답 시간	OFF → ON	2ms 이하	
	ON → OFF	2ms 이하	
내부소비전류 (5V)		90mA 이하	
COMMON 방식		8점/ 1COM (극성+,− 공통)	
동작 표시		LED 표시	
외선 접속방식		커넥타 접속 (20핀 x 2)	
적합 전선사이즈		0.2 mm²	
중 량		약 130g	
형 상		(B) Type	

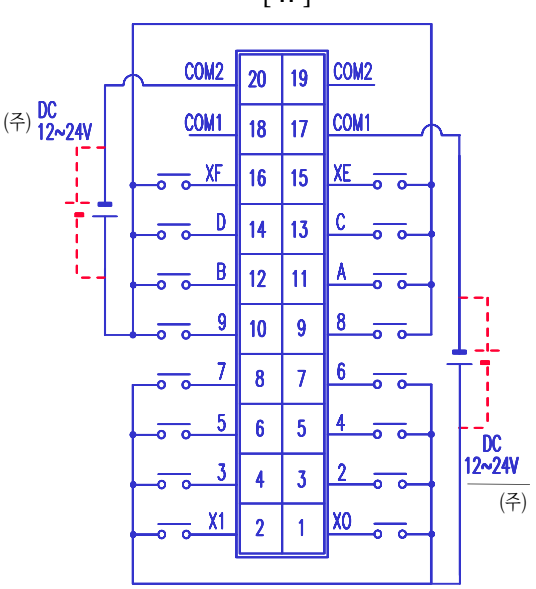
■ 내부회로 및 결선도

주 의 아래그림의 번호(1 20)는 제품 전면에 인쇄되어 있는 번호를 나타냅니다.

[I]



[II]



(주) NX70-X32D1 모듈은 입력전압이 24V DC입니다.

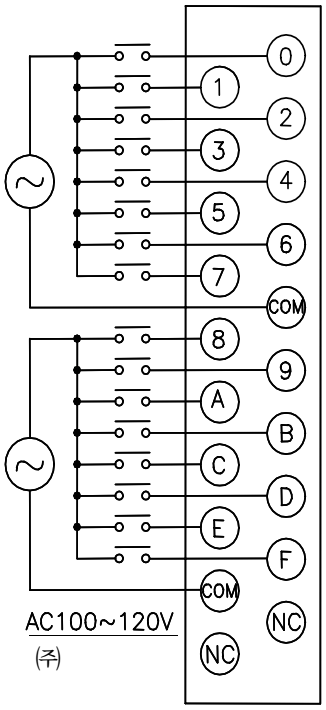
※ 내부회로는 앞페이지의 NX70-X16D 모듈과 동일합니다.

※ 커넥타의 [I], [II] 방향이 반대이므로 결선시 주의하십시오.

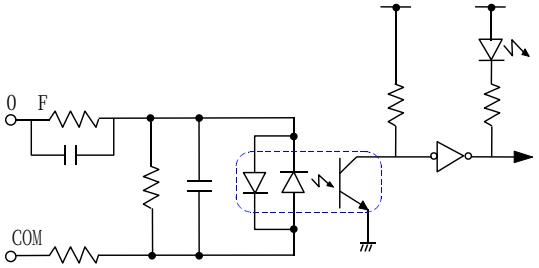
※ 커넥타 외부접속을 위하여 케이블 하네스 (CPL8800) 및 Pin Type Ass'y (CPL8880) 제품을 별도로 구매하여 사용하시기 바랍니다. (배선방법에 대해서는, 「4-5. 커넥터 타입 유니트의 배선」을 참조하십시오.)

제품 명칭		AC 입력유니트	
제품 코드		NX70-X16A110 (CPL93043)	NX70-X16A220 (CPL93053)
입력 점수		16점	
절연 방식		포토커플러	
정격 입력전압		AC 100 ~120V	AC 200 ~240V
사용 전압범위		AC 85 ~ 132V	AC 170 ~ 264V
최대 입력전류		20mA 이하	
ON 전압 / ON 전류		80V 이하 / 6mA 이하	160V 이하 / 6mA 이하
OFF 전압 / OFF 전류		30V 이상 / 3mA 이상	50V 이상 / 3mA 이상
입력임피던스		약 15KΩ	약 20KΩ
응답 시간	OFF → ON	15ms 이하	
	ON → OFF	15ms 이하	
내부소비전류 (5V)		80mA 이하	
COMMON 방식		8점/ 1COM	
동작 표시		LED 표시	
외선 접속방식		단자대 접속 (M3.0)	
적합 전선사이즈		0.5 1.25 mm^2	
중 량		약 160g	
형 상		(A) Type	

■ 결선도



■ 내부회로



※ NC : No Connectoin

(주) NX70-X16A220 : AC 200-240V

(3) 출력유니트 사양

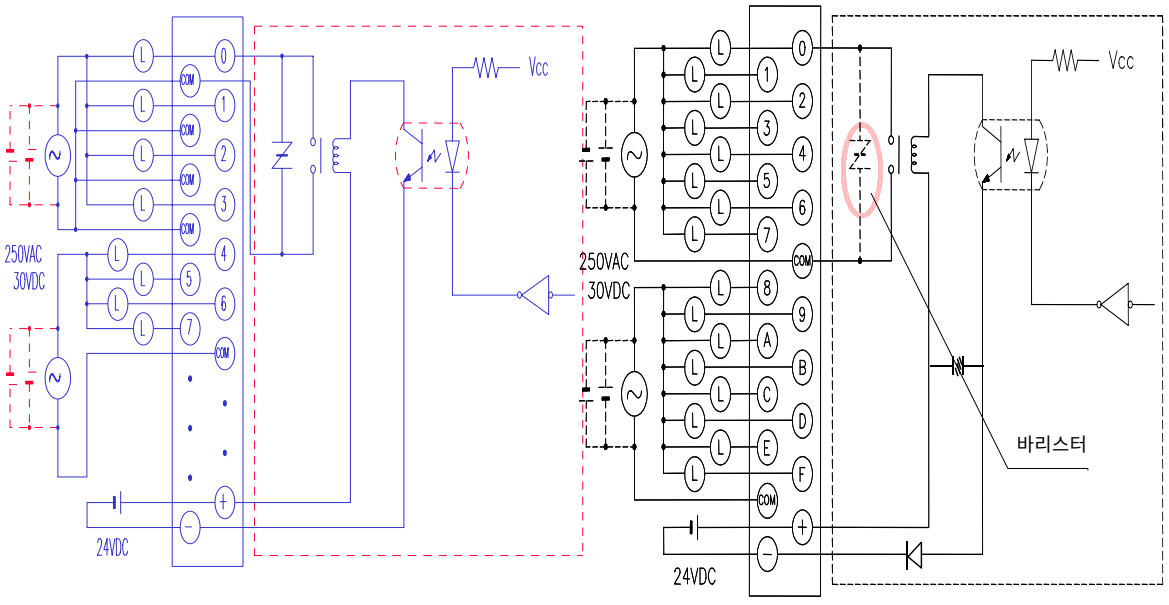
제품 명칭		RELAY 출력유니트		
제품 코드		NX70-Y8R (CPL93202)	NX70-Y16R (CPL93103)	NX70-Y16RV (CPL93203)
입력 점수		8점	16점	
절연 방식		포토커플러		
정격부하전압		250V AC, 30V DC		
사용 부하전압범위		85 V ~ 264V AC		
최대부하전류		3A /점	1A /점	
응답 시간	OFF → ON	10ms 이하		
	ON → OFF	10ms 이하		
외부공급 전원		24V 150mA 이하	24V 150mA 이하	
서지 보호 회로		바리스타	없음	바리스타
내부소비전류 (5V)		60mA 이하	100mA 이하	
COMMON 방식		4점 /COM, 1점 /COM x 4	8점 /1COM	
동작 표시		LED 표시		
외선접속방식		단자대 접속 (단자대 M3.0)		
적합전선 사이즈		0.5 1.25 mm²		
중 량		약 200g	약 300g	
형 상		(A) Type		

■ 내부회로 및 결선도

● NX70-Y8R

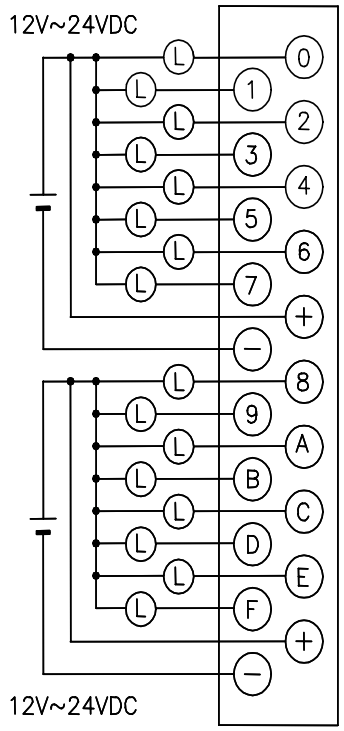
● NX70-Y16R : 바리스타 없음

● NX70-Y16RV : 바리스타 있음

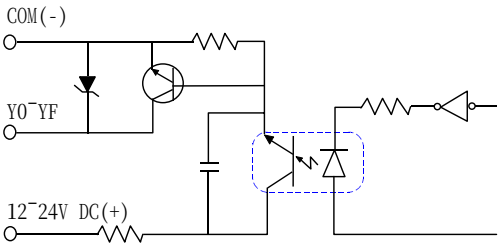


제품 명칭		TR 출력유니트 (NPN)
제품 코드		NX70-Y16T (CPL93483)
입력 점수		16점
절연 방식		포토커플러
정격부하전압		12 ~ 24V DC
사용 부하전압범위		10 ~ 30V AC
최대부하전류		0.6A /점
OFF시 누설전류		100μA 이하
응답 시간	OFF → ON	1ms 이하
	ON → OFF	1ms 이하
내부소비전류 (5V)		80mA 이하
서지 킬러		제너 다이오드
COMMON 방식		8점 /1COM(-)
동작 표시		LED 표시
외선접속방식		단자대 접속 (단자대 M3.0)
적합전선 사이즈		0.5 1.25 mm²
중 량		약 160g
형 상		(A) Type

■ 결선도



■ 내부회로

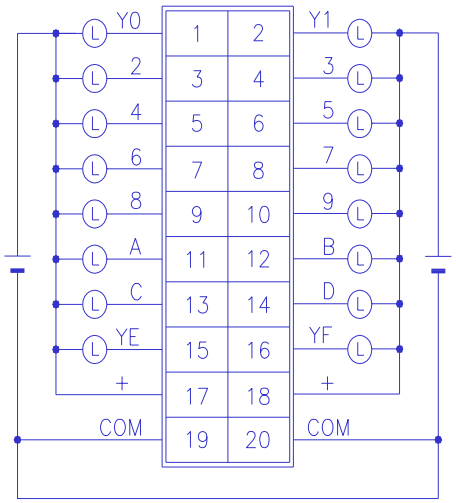


제품 명칭		TR 출력유니트
제품 코드		NX70-Y32T (NPN) (CPL93484)
입력 점수		32점
절연 방식		포토커플러
정격 부하전압		12 ~ 24V DC
사용 부하전압범위		10 ~ 30V AC
최대 부하전류		0.4A /점
OFF시 누설전류		100μA 이하
응답 시간	OFF → ON	1ms 이하
	ON → OFF	1ms 이하
내부소비전류 (5V)		140mA 이하
서지 킬러		제너 다이오드
COMMON 방식		16점/ 1COM (-)
동작 표시		LED 표시
외선 접속방식		커넥타 접속 (20핀 x 2)
적합 전선사이즈		0.2 mm^2
중 량		약 120g
형 상		(B) Type

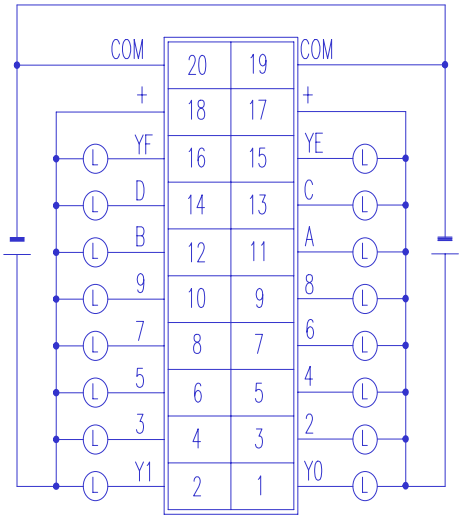
■ 내부회로 및 결선도

주 의 아래그림의 번호(1 20)는 제품 전면에 인쇄되어 있는 번호를 나타냅니다.

[I]



[II]



※ 커넥타의 [I], [II] 방향이 반대이므로 결선시 주의하십시오.

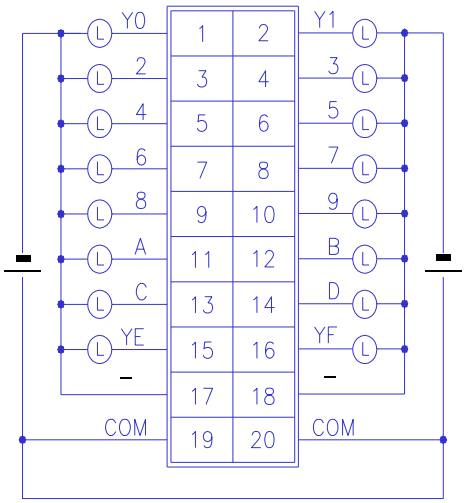
※ 커넥타 외부접속을 위하여 케이블 하네스 (CPL8810) 및 Pin Type Ass'y (CPL8880) 제품을 별도로 구매하여 사용하시기 바랍니다. (배선방법에 대해서는, 「4-5. 커넥터 타입 유니트의 배선」을 참조하십시오.)

제품 명칭		TR 출력유니트
제품 코드		NX70-Y32P (PNP) (CPL93584)
입력 점수		32점
절연 방식		포토커플러
정격 부하전압		12 ~ 24V DC
사용 부하전압범위		10 ~ 30V AC
최대 부하전류		0.4A /점
OFF시 누설전류		100μA 이하
응답 시간	OFF → ON	1ms 이하
	ON → OFF	1ms 이하
내부소비전류 (5V)		140mA 이하
서지 킬러		제너 다이오드
COMMON 방식		16점/ 1COM (+)
동작 표시		LED 표시
외선 접속방식		커넥타 접속 (20핀 x 2)
적합 전선사이즈		0.2 mm^2
중 량		약 120g
형 상		(B) Type

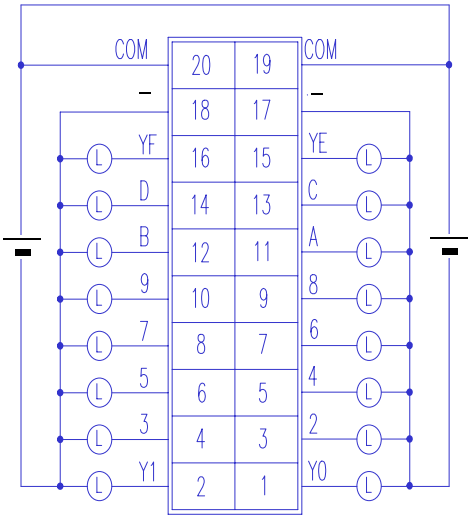
■ 내부회로 및 결선도

주 의 아래그림의 번호(1 20)는 제품 전면에 인쇄되어 있는 번호를 나타냅니다.

[Ⅰ]



[Ⅱ]



※ 커넥타의 [Ⅰ], [Ⅱ] 방향이 반대이므로 결선시 주의하십시오.

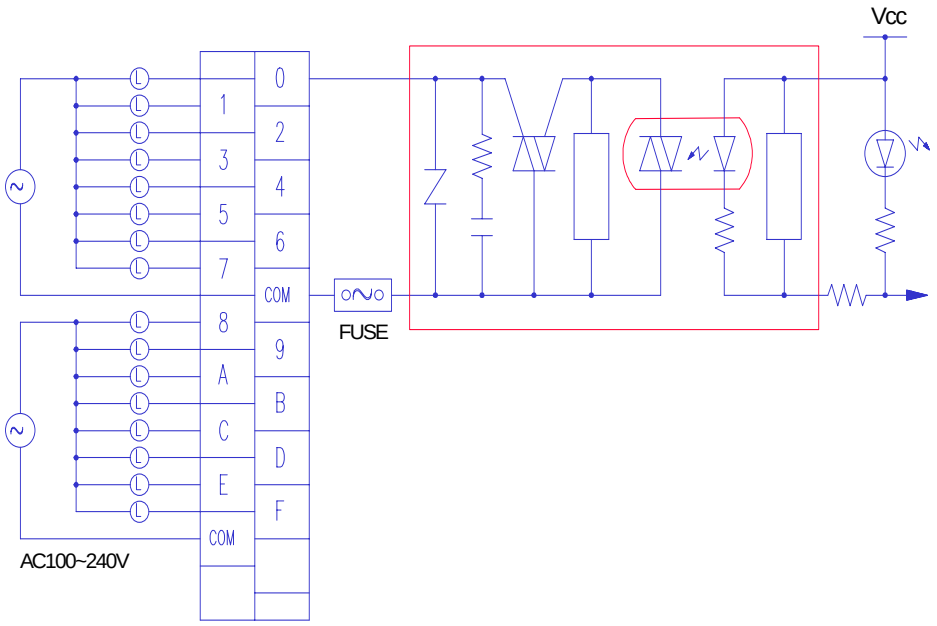
※ 커넥타 외부접속을 위하여 케이블 하네스 (CPL8810) 및 Pin Type Ass'y (CPL8880) 제품을 별도로 구매하여 사용하시기 바랍니다. (배선방법에 대해서는, 「4-5. 커넥터 타입 유니트의 배선」을 참조하십시오.)

※ NPN출력(NX70-Y32T) 제품과 컴먼방식이 다르므로 전원 연결 방향에 주의를 바랍니다.

- **17, 18번 핀 : - (VDC-)**
- **19, 20번 핀 : COM (VDC+)**

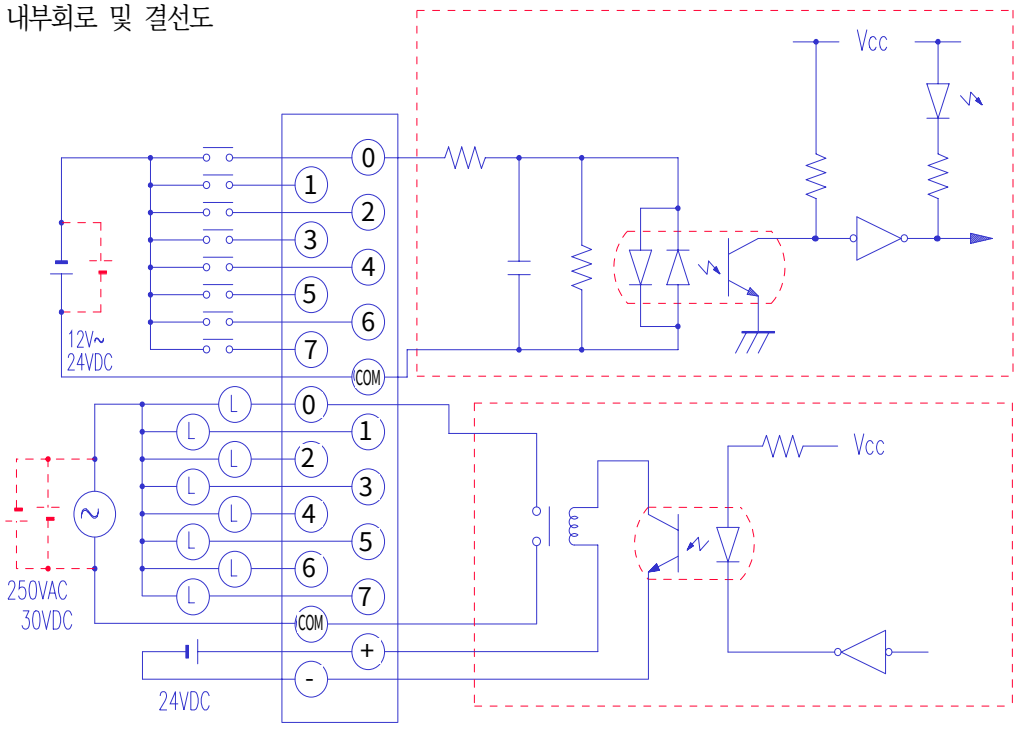
제품 명칭		SSR 출력유니트
제품 코드		NX70-Y16SSR (CPL93703)
입력 점수		16점
절연 방식		S S R
정격 부하전압		100 ~ 240V AC
사용 부하전압범위		85 ~ 264V AC
최대 부하전류		0.5A /점
OFF시 누설전류		100μA 이하
응답 시간	OFF → ON	1ms 이하
	ON → OFF	0.5 CYCLE + 1ms 이하
내부 소비전류 (5V)		250mA 이하
퓨즈 정격		3A
COMMON 방식		8점/ 1COM
동작 표시		LED 표시
외선 접속방식		단자대 접속 (단자대 M 3.0)
적합 전선사이즈		0.5 1.25 mm²
중 량		약 300g
형 상		(A) Type

■ 내부회로 및 결선도



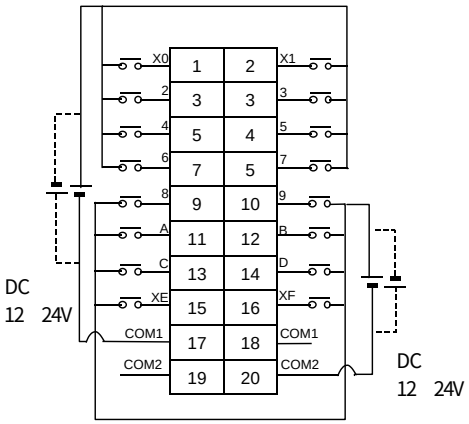
제품 명칭		DC입력 / RELAY출력 혼합유니트	
제품 코드		NX70-XY16 (CPL93088)	
입출력 점수(16점)		입력 8점 (DC Input)	출력 8점 (Relay Output)
절연 방식		포토커플러	
정격입력전압		12 ~ 24V DC	
사용전압범위		10.2 ~ 26.4V DC	
최대입력전류		10mA 이하	
정격부하전압/ 전류			250V AC, 30V DC, 1A /점
ON 전압 / ON 전류		9.6V 이하 / 4mA 이하	
OFF 전압 / OFF전류		2.5V 이하 / 1.5mA 이하	
입력임피던스		약 3KΩ	
외부공급 전원		필요 없음	24V 150mA 이하
응답 시간	OFF → ON	10ms 이하	
	ON → OFF	10ms 이하	
내부소비전류 (5V)		80mA 이하	
COMMON 방식		8점/ 1COM (극성+,− 공통)	8점 /1COM
동작 표시		LED 표시	
외선접속방식		단자대 접속 (M3.0)	
적합전선 사이즈		0.5 1.25 mm²	
중 량		약 220g	
형 상		(A) Type	

■ 내부회로 및 결선도

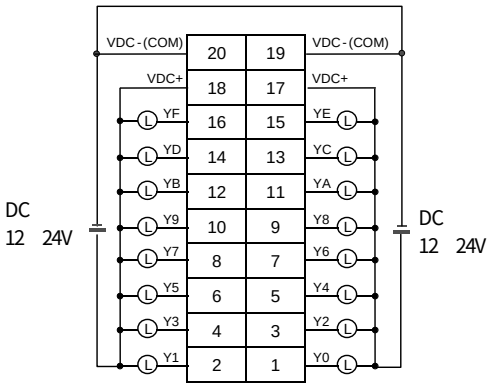


제품 명칭		DC입력 / TR 출력 혼합유니트	
제품 코드		NX70-XY32	
입출력 점수(32점)		입력 16점 (DC Input)	출력 16점 (TR Output, NPN)
절연 방식		포토커플러	
정격입력전압		12 ~ 24V DC	
사용전압범위		10.2 ~ 26.4V DC	
최대입력전류		10mA 이하	
정격부하전압/ 전류			12 ~ 24V DC, 0.4A /점
ON 전압 / ON 전류		9.6V 이하 / 4mA 이하	
OFF 전압 / OFF전류		2.5V 이하 / 1.5mA 이하	
입력임피던스		약 3KΩ	
서지 킬러			제너 다이오드
외부공급 전원		필요 없음	
응답 시간	OFF → ON	10ms 이하	1ms 이하
	ON → OFF	10ms 이하	1ms 이하
내부소비전류 (5V)		120mA 이하	
COMMON 방식		8점/ 1COM (극성+,− 공통)	16점/ 1COM (-)
동작 표시		LED 표시	
외선접속방식		커넥타 접속 (20핀 x 2)	
적합전선 사이즈		0.2 mm²	
중 량		약 120g	
형 상		(B) Type	

■ 내부회로 및 결선도



입력 16점 (DC Input)



출력 16점 (TR Output)

※ 커넥타 외부접속을 위하여 케이블 하네스(DC Input : CPL8800, TR Output :CPL8810) 또는 Pin Type Ass'y (CPL8880) 제품을 별도로 구매하여 사용하시기를 추천합니다.
(자세한 배선방법에 대해서는,「시스템메뉴얼」을 참조하십시오.)

3장

내·외부 번지 지정

3-1. 내·외부 번지지정

■ 모든 외부 입·출력모듈과 내부 데이터 처리를 위한 메모리에는 번지(Address)와 Data가 항상 동시에 존재합니다.

■ 메모리 번지 지정은 그 종류에 따라 R, L, M, K, F, TC, W 등으로 구분합니다.

종 류	범 위	설 명
외부 입출력 영 역 (R)	R0.0 ~ R127.15	- 입출력 모듈구성시 지정가능한 영역 (Local) - 2048점, 128 워드
링크 접점 영역 (L)	L0.0 ~ L63.15	- 링크 접점 공유 영역,1024점, Loop 0 - 링크 사용 않을시 내부접점 사용 가능
	M0.0 ~ M63.15	- 링크 접점 공유 영역, 1024점, Loop 1
내부접점 영역 (M)	M0.0 ~ M127.15	- 내부 보조접점 영역 - 2048점, 128워드
정전유지 접점영역 (K)	K0.0 ~ K127.15	- 정전기억 가능한 내부보조 접점영역 - 2048 점, 128워드 - CPU초기화 또는 정전영역 초기화를 실행해야 Clear됩니다
특수접점 영역 (F)	F0.0 ~ F15.15	- 특수 내부접점영역 - 256점, 16워드
타이머/ 카운터 영역 (TC)	TC0 ~ TC255 설정치 : W2048 (=SV0) ~W2303 (=SV255) 현재치 : W2304 (=PV0) ~W2559 (=PV255)	- 256채널 공동사용(타이머, 카운터) - TC는 접점표시 - 설정치는 SV로 사용하며, 현재치(경과치)는 PV로 사용. - SV는 0 ~ 65535까지 지정가능
데이터 워드영역1 (W)	W0000 ~ W2047 W0~W255는 링크공유 겸용	- 정전시 데이터 보존영역 - Bit로 지정불가 - CPU초기화 또는 정전영역 초기화를 실행해야 Clear됩니다 -주) W000 ~ W255 까지는 링크데이터 공유영역으로도 사용이 가능 합니다.
데이터 워드영역2 (W)	W3072 ~ W5119	- 확장된 데이터 레지스터 영역 (CPU Ver2.0 이상 지원) - 정전시 데이터 보존됩니다. - Bit로 지정불가 - NX70_CPU70p2 모델에서 지원됩니다.
특수워드 영역 (SR)	SR000 ~ SR511	- CPU상태표시, RTC등의 특수 내부 데이터 영역

● R, L, M, K, F, TC 영역은 Bit, Word로 데이터처리가 가능합니다.

● W 영역은 Word 데이터처리만 가능합니다.

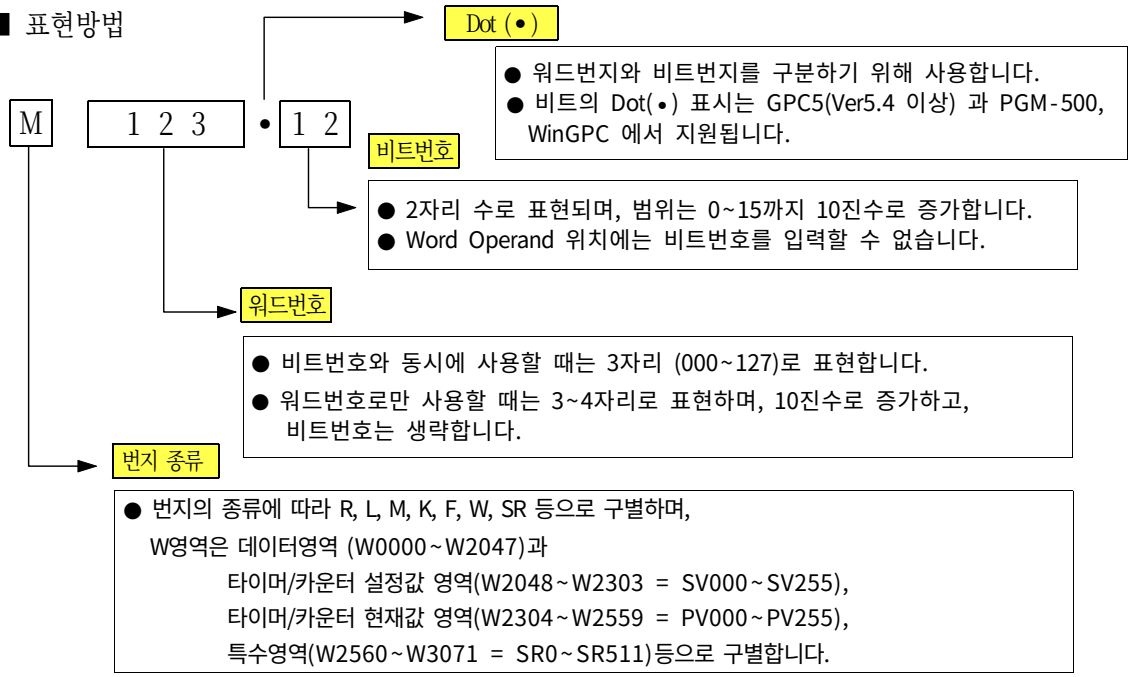
● L(링크영역)은 내부접점으로 사용가능합니다.

● K,W, Timer/Counter PV Register는 정전시 Data가 유지됩니다.

■ 비트, 워드 번지지정

- 비트 번지는 종류를 구별하는 문자인 R, L, M, K, F 등과 워드를 표시하는 3자리의 10진수 숫자 (000~127까지)와 비트를 표시하는 2자리의 10진수(00~15까지) 숫자로 구성하며, 타이머 및 카운터의 접점은 TC000과 같이 문자(TC)와 채널번호(000~255)로 구성합니다.
- 워드번지는 문자 R, L, M, K, F, W, SV, PV, SR등과 워드를 표시하는 3자리(000~127까지) 또는 4자리(0000~2047까지)의 10진수 숫자로 구성되며, 특수영역인 SR000~SR511까지는 W2560~W3071처럼 사용하기도 합니다.
- 비트 번지와 워드번지를 동시에 사용할 수 있는 R, L, M, K, F등은 명령어의 종류에 따라 비트번지와 워드번지가 자동으로 구분되므로 사용할 때 주의해야 합니다.
- 비트 번지는 ON(1)또는 OFF(0)등의 상태를 나타내며, 워드 번지는 16bit로 구성되어 데이터 값을 0~65535까지 처리할 수 있고, 더블 워드를 처리하는 명령어는 32bit로 구성되어 값을 0~4,294,967,295 까지 처리할 수 있습니다.

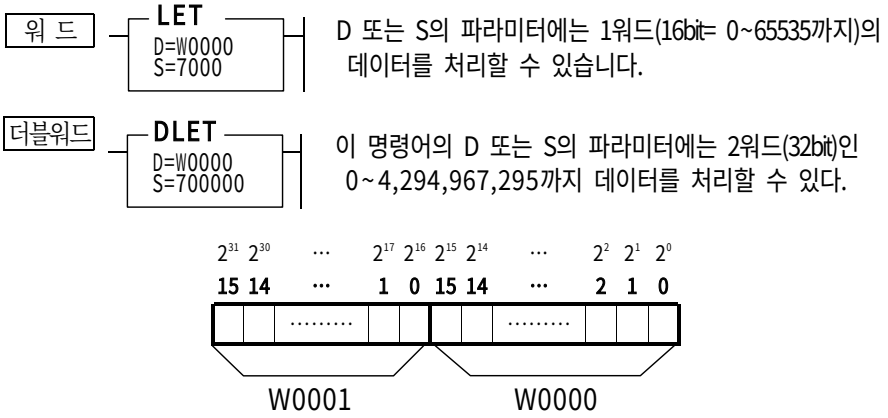
■ 표현방법



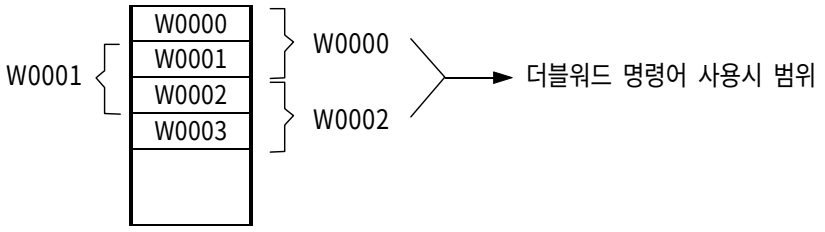
3-2. 더블워드 번지지정

■ 더블워드 지정은 워드번호지정 방법과 동일하고, 지정한 번호와 그 다음 번호가 합쳐져 32bit의 데이터를 처리합니다.
■ 워드와 더블워드의 구별은 명령어로 합니다.
■ GPC5를 이용하여 더블워드 비교명령어를 사용할 때는 편집상태에서 'Ctrl+T' 키를 눌러 모드를 'Double' 상태에 두고 비교명령을 입력하면 더블워드 비교명령이 처리됩니다. 래더의 모양도 명령어 박스 안에 'D'글자가 나타나며, 더블워드 데이터 입력 및 처리를 한다.

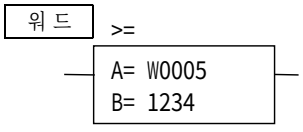
예 1) 워드와 더블워드 구분



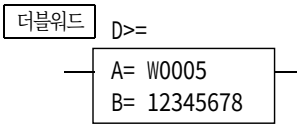
즉, W0는 W0와 W1이 합쳐져 구성되고 W0는 LSB이고, W1은 MSB에 위치한다.
또한, W1로 지정하면 W1 과 W2가 합쳐진 형태가 된다.
데이터를 중복으로 사용하면 문제가 발생할 수 있으므로 주의를 해야한다.



예 2) 비교명령 표시 (WinGPC 에서)



- WinGPC 화면상에서 응용명령 입력을 선택하고 부호를 입력하면 좌측과 같이 나타나고
- W0005와 1234등의 오퍼랜드는 워드 값으로 16bit 데이터를 처리한다.



- WinGPC 화면상에서 응용명령 입력을 선택하고 부호 앞에 D를 먼저 입력하고 부호를 입력하면 더블워드 지정이 되며, 2워드의 데이터를 처리합니다.
- W0005는 W0005와 W0006이 합쳐져 있고, 32bit 데이터를 처리할 수 있다.

3-3. 절대번지 지정

■ 통신포트를 이용한 PLC 레지스터를 제어하거나 LDR, DLDR, STO, DSTO 등의 명령에서 레지스터의 번지를 간접지정 하여 프로그램 처리를 할 경우 절대번지를 이용합니다.

구분	레지스터 번지	절대번지(워드)	
		dec.	hex.
외부입출력	R0000	0000	$0000
	R0001	0001	$0001
	R0002	0002	$0002
	...	...	...
	R0126	0126	$007E
	R0127	0127	$007F
링크접점	L0000	0128	$0080
	L0001	0129	$0081
	L0002	0130	$0082
	...	...	...
	L0062	0190	$00BE
	L0063	0191	$00BF
내부접점	M0000	0192	$00C0
	M0001	0193	$00C1
	...	...	...
	M0063	0255	$00FF
	M0064	0256	$0100
	...	...	...
	M0127	0319	$013F
정전유지접점	K0000	0320	$0140
	K0001	0321	$0141
	K0002	0322	$0142
	...	...	...
	K0126	0446	$01BE
	K0127	0447	$01BF
내부특수접점	F0000	0448	$01C0
	F0001	0449	$01C1
	...	...	...
	F0014	0462	$01CE
	F0015	0463	$01CF

구분	레지스터 번지	절대번지(워드)	
		dec.	hex.
T/C 접점	TC00~15	0464	$01D0
	TC16~31	0465	$01D1
	...	...	...
	TC240~255	0479	$01DF
데이타영역	W0000	0512	$0200
	W0001	0513	$0201
	...	...	...
	W1024	1536	$0600
	...	...	...
	W2046	2558	$09FE
	W2047	2559	$09FF
T/C 설정치	SV000	2560	$0A00
	SV001	2561	$0A01
	...	...	...
	SV255	2815	$0AFF
T/C 현재치	PV000	2816	$0B00
	PV001	2817	$0B01
	...	...	...
	PV255	3071	$0BFF
상태표시	SR000	3072	$0C00
	SR001	3073	$0C01
	...	...	...
	SR256	3328	$0D00
	...	...	...
	SR511	3583	$0DFF
확장데이터영역 **주1)**	W3072	3584	$0E00
	...	...	...
	W4096	4608	$1200
	...	...	...
	W5119	5631	$15FF

주1) 확장 데이터 영역은 NX-CPU70P2기종에 적용된 확장된 2048워드의 절대번지입니다.

참고)

- 통신기능에서 사용하는 **비트절대번지**는 워드절대번지와 해당워드의 비트번호 (0~15 or $0 $F)로 구성됩니다.

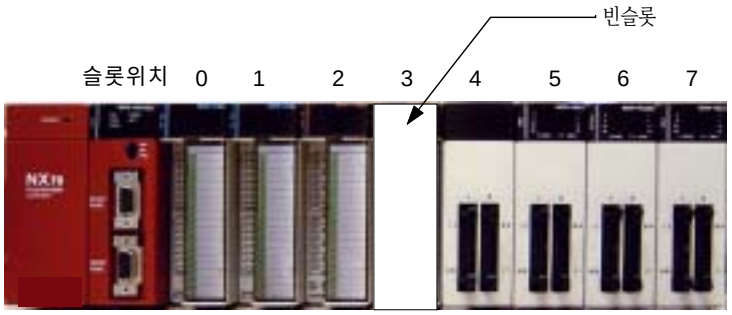
1워드표시

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
워드 절대번지												비트번호			

예) K127.12 내부출력에 대한 비트절대번지는 $1BFC(Hex)입니다.
("**워드 절대번지=$01BF" + "비트번호=$C" = $1BFC**)

3-4. 입출력 번지 지정 - 모듈위치에 따른 번호 지정

[예 1]



(8슬롯 시스템의 경우)

■ 번호지정 예 (상기그림예)

슬롯위치		00	01	02	03	04	05	06	07
워드번호	CPU 유니트	R0	R1	R2		R3 R4	R5 R6	R7 R8	R9 R10
비트번호		R0.0 R0.1 R0.2 : R0.15	R1.0 R1.1 R1.2 : R1.15	R2.0 R2.1 R2.2 : R2.15	**빈슬롯**	R3.0 R3.1 R3.2 : R4.15	R5.0 R5.1 R5.2 : R6.15	R7.0 R7.1 R7.2 : R8.15	R9.0 R9.1 R9.2 : R10.15

[예 2]

(10슬롯 시스템의 경우)

■ 번호지정 예 (상기그림예)

슬롯위치		00	01	02	03	04	05	06	07	08	09
워드번호	CPU 유니트	R0	R1	R2	R3	R4	R5 R6	R7 R8	R9 R10	R11 R12	R13 R14
비트번호		R0.0 R0.1 R0.2 : R0.15	R1.0 R1.1 R1.2 : R1.15	R2.0 R2.1 R2.2 : R2.15	R3.0 R3.1 R3.2 : R3.15	R4.0 R4.1 R4.2 : R4.15	R5.0 R5.1 R5.2 : R6.15	R7.0 R7.1 R7.2 : R8.15	R9.0 R9.1 R9.2 : R10.15	R11.0 R11.1 R11.2 : R12.15	R13.0 R13.1 R13.2 : R14.15

■ 슬롯별 번지지정 기준

- CPU에서 입력번지와 출력번지를 워드 단위로 자동으로 구별합니다.
- 16점 입·출력 모듈은 1워드 단위로 지정이 되며, 혼합모듈은 입력용 1워드와 출력용 1워드로 구별이 되고 16점 모듈일때 하위 8bit(0~7) 까지만 사용 가능합니다.
- CPU모듈에서 가까운 슬롯(왼쪽)이 오른쪽 슬롯의 번지보다 높게 지정할 수는 없습니다.
- 빈 슬롯은 모듈이 장착되지 않은 상태를 말하며, 더미(NX-DUMMY) 유니트는 장착할 수 있고, 자동으로 설정할 때 (CPU자동설정)는 번지가 할당되지 않습니다.

■ 모듈 종류별 번지지정 (자동설정의 경우, NX-CPU70p에 한함)

구분	종류	할당워드	입력	출력
입력	NX70-X16D	1	1	
	NX70-X16D1	1	1	
	NX70-X16A110	1	1	
	NX70-X16A220	1	1	
	NX70-X32D	2	2	
	NX70-X32D1	2	2	
출력	NX70-Y8R	1		1
	NX70-Y16R	1		1
	NX70-Y16RV	1		1
	NX70-Y16T	1		1
	NX70-Y16SSR	1		1
	NX70-Y32T	2		2
	NX70-Y32P	2		2
혼합	NX70-XY16	2	1	1
	NX70-XY32	2	1	1

구분	종류	할당워드	입력	출력
A/I	NX70-AI8C	8 (1)	8 (1)	
	NX70-AI8V	8 (1)	8 (1)	
	NX70-AI4C	4 (1)	4 (1)	
	NX70-AI4V	4 (1)	4 (1)	
	NX70-RTD4	4 (1)	4 (1)	
	NX70-TC4	4 (1)	4 (1)	
A/O	NX70-AO4C	4 (1)	(1)	4
	NX70-AO4V	4 (1)	(1)	4
	NX70-AO2C	2 (1)	(1)	2
	NX70-AO2V	2 (1)	(1)	2
특수	NX70-HSC1	2	1	1
	NX70-HSC2	2	1	1
	NX70-HSC4	4	2	2
	NX70-PULSE4	4	2	2
	NX70-POSI2	4	2	2
	NX70-POSI4	8	4	4
통신	NX70-CCU+	0	-	-
	NX70-SCU	2	1	1
	NX70-MWLink	0	-	-
	NX70-DNS	0	-	-

주) - 할당워드의 ( ) 안의 숫자는 번지 지정방식을 공유메모리 방식으로 설정할 때 해당됩니다.
- 디지탈 입출력, 아날로그 등은 번지지정시 수동설정으로 할 수 있으며, 할당된 워드값을 초과하여 지정 할 수도 있습니다.

■ NX70 PLC와 N70 PLC의 모듈 호환성 관련

N70 PLC와 NX70 PLC는 CPU, 마더보드, POWER 모듈을 제외하고, 기타 모든 모듈은 상호 호환성이 있습니다.

ex1) NX70 마더보드(NX70-BASE02 NX70-BASE12)상에 N70의 CPU(CPL9215..) 및 POWER 모듈(CPL9631..)은 실장할 수는 없습니다.

ex2) NX70 마더보드(NX70-BASE02 NX70-BASE12)상에 기존 N70의 I/O 및 특수모듈은 실장할 수 있습니다.

ex3) N70 마더보드(CPL9502 CPL9508)상에 NX70의 CPU 및 POWER 모듈은 실장할 수 없습니다.

3-5. 특수 내부레지스터 번지

1. F000 F015 워드레지스터 기능

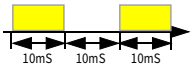
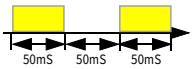
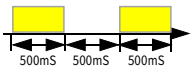
워드번지	기 능	상 세 내 용
F000	시스템 점검/제어	시스템을 자체 점검/프로그램 점검, 동작 제어
F001	시스템 점검/Clock	내부클럭, 연산 결과,Carry Flag
F002~F007	Link 제어 영역	Link 장착, 동작 모드, 상태정보 등
F008~F010	Remote 제어 영역	Remote 동작제어 Flag, 상태정보 등
F011~F013	시스템 제어	사용자통신, RTC 장착, 시스템 제어 등
F014	PID 제어	PID 동작모드 및 운전제어 플래그(채널0~3)
F015	PID 제어	PID 동작모드 및 운전제어 플래그(채널4~7)

2. F000 워드(F0.0 F0.15)의 기능

번지	기 능	상 세 내 용
F0.00	시스템 점검	전원투입시 시스템을 자체 점검하여 고장시 ON됨
F0.01	CPU ROM점검 (ROM CheckSum)	전원투입시 시스템 ROM을 자체 점검하여 고장시 ON되며, Error램프가 켜지고, 출력과 운전이 정지됨
F0.02	CPU RAM검사	전원ON시 RAM을 진단하여 에러발생시 ON되며 운전정지됨.
F0.03	사용자 메모리 에러	사용자 프로그램 메모리가 손상되거나 프로그램 내용이 파괴된 경우 ON되며, Error램프ON, 출력과 운전이 정지됨
F0.04	프로그램 문법 에러	CPU가 최초로 운전(RUN)을 시작할때 프로그램의 문법을 점검하여 이상시 Error램프 ON, 출력과 운전이 정지됨
F0.05	I/O모듈범위에러	R64 워드 이상의 번지 입력시 ON됨
F0.06	모듈변경에러	운전중에 입출력 시스템에 이상이 발생시 ON되며, 스위치가 Remote상태에서는 운전이 정지하며, 스위치가 RUN에서는 에러 램프는 켜지고 CPU는 계속 RUN된다.
F0.07	모듈종류에러 (비정상 모듈)	CPU에 저장된 입출력 모듈 종류와 실제 장착된 모듈이 다를때 ON되며, Error램프가 켜지고 운전(RUN)이 정지됨
F0.08	입력 데이터 제어	CPU운전중 입력모듈의 데이터를 프로그램과 차단 (Input Update를 No로함) 하고자 할때 OFF시킴
F0.09	출력 데이터 제어	CPU운전중 연산결과를 차단하고자 할때 OFF(Output Update를 No로함)시키며, OFF직전의 데이터가 출력모듈에 유지됨
F0.10	모든 출력 OFF	CPU운전중 모든 출력을 OFF(Output Enable을 No로함)시킴
F0.11	정주기 인터럽트	정주기 인터럽트 명령어 수행시 ON됨
F0.12	위치독 에러	1스캔시간이 워치독 타임 설정치보다 길때 ON됨
F0.13	모듈종류무시	최초운전시 입출력에 상관없이 프로그램을 점검할때 ON
F0.14	운전(RUN)중 프로그램변경	운전중 사용자 프로그램을 수정하고자 할때 ON시키며, 문법 에러가 발생하면 운전(RUN)이 정지됨
F0.15	CPU가 운전중일때 ON됨	CPU가 운전(RUN)중일때 ON되며, 운전이 정지(Stop)되거나 일시정지(Pause)될 때도 OFF로 된다.

3. F001 워드(F001.0 F001.15)의 기능

비트처리만 됨

번지	기 능	상 세 내 용	비 고
F1.0	최초1스캔 ON	STOP→ RUN 상태로 전환시 1스캔동안 ON됨	
F1.1	스캔 클럭	매 스캔마다 ON/OFF 반전됨 (1SCAN ON, 1SCAN OFF)	
F1.2	0.02초 클럭	10ms : ON, 10ms : OFF 	
F1.3	0.1초 클럭	50ms : ON, 50ms : OFF클럭 	
F1.4	1초 클럭	500ms : ON, 500ms : OFF클럭 	
F1.5	순간정전표시	20ms이상 전원이 OFF되면 ON됨	유지됨
F1.6	CPU스위치 실행상태 표시	CPU 스위치가 RUN 위치에서 운전중 (RUN상태)일때 ON됨 주의) CPU 스위치가 Remote에서 RUN운전 중에는 동작하지 않으므로 F0.15접점으로 사용바랍니다.	
F1.7	K영역 에러표시	K영역의 데이터가 파괴, 변경된 경우 ON됨	
F1.8	자리올림 (캐리)	연산명령어 사용시 캐리가발생하면 ON됨	
F1.9	나눗셈 연산에러	프로그램의 나누기 명령시 분모가 0 일때 ON	
F1.10	범위지정 에러	절대번지사용시 지정범위 초과시 ON됨	
F1.11	Reserved(예약)	시스템 사용 영역	사용금지
F1.12	W영역 에러표시	시스템 사용 영역	
F1.13	Reserved(예약)	시스템 사용 영역	사용금지
F1.14	Reserved(예약)	시스템 사용 영역	사용금지
F1.15	Reserved(예약)	시스템 사용 영역	사용금지

참고사항

F001워드의 총 16비트는 CPU의 특수기능 및 자체진단 결과를 제공하는 비트로서 출력만 되고 수정, 입력은 안됩니다. (단, F001.05 순간정전 표시 접점은 사용자가 OFF (해제)시킬수 있음.)

4. F011 워드의 기능 : COM2 단자의 사용자 정의 통신용(CPU70p2용)

번지	기 능	상 세 내 용
F11.00	사용자 데이터 송신 요구	1: 송신 요구
F11.01	송신 실패 보고	1: 송신 실패 표시
F11.02	ASCII 통신 모드시 시작 코드 조건 사용	1: 시작 코드 사용
F11.03	ASCII 통신 모드시 종료 코드 조건 사용	1: 종료 코드 사용
F11.04	사용자 데이터 수신 완료	1: 수신완료 표시
F11.05	수신 메모리 초기화 (수신데이터 Read후 '1'로SET)	1:수신 메모리 초기화
F11.06	수신 데이터 중복	1: 수신데이터 중복
F11.07	수신 데이터 에러	1: 에러 발생
F11.08	ASCII 데이터 수신시 ASCII 데이터형태로 표시	1: ASCII 데이터 갱신
F11.09	수신중 에러 발생시 이의 무시	1: 에러 발생 무시
F11.10	페러티 동작 여부	1: 페러티 조건 사용
F11.11	ODD/EVEN 페러티 정의	0: ODD, 1: EVEN
F11.12	7 또는 8 Bit 통신 모드 설정	0: 8 비트, 1: 7 비트
F11.13	Port 2의 통신 데이터 형태 지정	0: ASCII, 1: HEX ,주2)
F11.14	COM2 Modbus통신 정상	MODBUS 통신시 정상 상태일 때 ON (V2.30이상)
F11.15	Reserved(예약)	시스템 사용 영역

5. F012 워드의 기능 : 시스템 제어용

번지	기 능	상 세 내 용
F12.00	RTC존재확인	RTC기능이 있을때 ON됨.
F12.01	REMOTE MAP설정	REMOTE I/O MAP설정이 되어 있으면 ON됨.
F12.02	COM1 Modbus지정	COM1 Port를 Modbus RTU모드로 전환할 때 Set (V2.0 이상)
F12.03	FLASH 유,무	FLASH 메모리가 정착되어 있을때 ON됨.
F12.05	Battery 에러	밧데리가 연결되지 않았거나, Backup 전압보다 낮으면 Error LED 가 깜박임.
F12.07	정주기스캔이상	정주기 프로그램수행중 이상이 발생하면 ON됨.
F12.08	사용자통신 사용	통신용 COM2 단자를 사용자정의 통신 모드로 전환함
F12.09	COM2 Modbus지정	COM2 단자를 Modbus 통신용으로 전환함
F12.10	RTC설정 에러	RTC설정 오류발생시 ON됨(정상적이면 OFF됨.)
F12.11	FLASH저장 및 부팅검사	F12.15를 ON하여 FLASH에 프로그램이 정상적으로 저장되면 OFF됨.
F12.12	EEPROM백업확인	EEPROM 과 RAM의 프로그램이 동일하면 ON됨
F12.13	RTC 설정1	년/월/요일/일을 변경하고자 할때 ON됨.
F12.14	RTC 설정2	시/분/초를 변경시 ON되며. 설정이 정상적이면 OFF됨
F12.15	FLASH 저장	사용중인 프로그램을 FLASH에 저장, 보관하려 할때 ON시킴. 정상시 : OFF됨.

6. F014 ~F015 워드의 기능 : PID 제어용

번지	기 능	상 세 내 용
F14.00	PID Loop0 제어	Loop 0 의 1=PID동작, 0=PID정지
F14.01		파라미터 범위설정 1=에러, 0=정상
F14.02		PID연산모드 1=수동, 0=자동
F14.03		PID실행완료 표시 1=완료, 0=연산중
F14.04~ F14.07	PID Loop1 제어	Loop 1 의 PID제어 (기능은 Loop0 내용과 동일)
F14.08~ F14.11	PID Loop2 제어	Loop 2 의 PID제어 (기능은 Loop0 내용과 동일)
F14.12~ F14.15	PID Loop3 제어	Loop 3 의 PID제어 (기능은 Loop0 내용과 동일)
F15.00~ F15.03	PID Loop4 제어	Loop 4 의 PID제어 (기능은 Loop0 내용과 동일)
F15.04~ F15.07	PID Loop5 제어	Loop 5 의 PID제어 (기능은 Loop0 내용과 동일)
F15.08~ F15.11	PID Loop6 제어	Loop 6 의 PID제어 (기능은 Loop0 내용과 동일)
F15.12~ F15.15	PID Loop7 제어	Loop 7 의 PID제어 (기능은 Loop0 내용과 동일)

7. 특수 레지스터 SR영역

■ SR영역은 SR000(절대번지 $0C00) ~ SR511(절대번지 $0DFF)까지를 말하며, 대부분 시스템 제어용으로 사용하므로 사용 가능한 영역만 주의하여 사용해야만 합니다.

1) SR000~SR511 번지 기능 요약

워드 번지		기 능	상 세 내 용
SR000~SR004	W2560~W2564	CPU 상태	CPU에서 운전중인 상태 데이터를 표시함.
SR005~SR007	W2565~W2567	Reserved(예약)	시스템 사용 영역
SR008	W2568	PID 제어용	PID 제어함수 설정용 시작번지 지정
SR009~SR016	W2569~W2576	Reserved(예약)	시스템 사용 영역
SR017~SR048	W2577~W2608	프로그램 에러 표시	프로그램 처리중 에러발생 내용 표시영역
SR049~SR288	W2609~W2848	Reserved(예약)	시스템 사용 영역
SR289~SR297	W2849~W2857	시계 기능	시계기능 설정 및 표시영역
SR298~SR373	W2858~W2933	사용자정의 통신	COM2 단자의 사용자정의 통신 Protocol 제어용
SR374~SR511	W2934~W3071	Reserved(예약)	시스템 사용 영역

2) SR000(W2560) ~ SR029(W2589) 번지 기능 :

(CPU, 링크, 리모트 및 에러상태 표시)

번지		기 능	상 세 내 용
SR000	W2560	CPU ID번호	CPU의 접속번호(CPU ID Number)를 나타내며 하위 8bit로 표현되고, 0~223까지는 사용자 지정 영역이고, 255번은 Default로 접속가능한 번호입니다.
SR001	W2561	CPU상태 표시	CPU가 운전, 정지, 원격제어, 에러등의 상태를 표시함.  즉, Stop=010, REM/Pause=011, REM/RUN=111, RUN/RUN=101
SR002	W2562	워치독타임	사용자가 설정한 워치독 타임표시 (단위 : mSec)
SR003	W2563	스캔 타임	프로그램 스캔시간 표시, 매스캔마다 갱신됨(mSec)
SR004	W2564	최대스캔타임	프로그램 운전중 최대 스캔타임 값이 저장됨
SR005~ SR007	W2565~ W2567	Reserved (예약)	시스템 사용 영역
SR008	W2568	PID 제어용	PID 제어함수 설정용 시작번지 지정 참조
SR009~ SR016	W2569~ W2576	Remote I/O 정보	Remote Master의 베이스워드, Complement값 지정 (리모트 매뉴얼 참조)
SR017	W2577	시스템 에러 정보 표시	전원투입시 시스템 자기진단 결과 이상시 F0.0을 ON을 시키며, 그 에러 내용을 저장하는 영역임 
SR018	W2578	미정의명령 위치	정의되지 않은 명령어를 사용한 스텝번호 표시
SR019	W2579	Reserved	시스템 사용 영역
SR020	W2580	곱셈값 하위	곱셈 명령 수행후 결과의 하위값 저장
SR021	W2581	곱셈값 상위	곱셈 명령 수행후 결과의 상위값 저장
SR022	W2582	나머지값하위	나눗셈명령 수행후 나머지의 하위바이트 값을 저장
SR023	W2583	나머지값상위	나눗셈명령 수행후 나머지의 상위바이트 값을 저장
SR024~ SR029	W2584~ W2589	Reserved (예약)	시스템 사용 영역

3) SR30(W2590) ~ SR48(W2608)번지 기능 : 문법에러 정보

번지		기 능	상 세 내 용
SR030	W2590	에러 정보 표시	Bit 0 = 비트명령의 입출력번호가 지정범위를 벗어난 경우에 ON. Bit 1 = 타이머나 카운터 채널 번호 범위가 255를 초과하였거나 채널번호가 중복된 경우 ON됨 Bit 2 = 응용 명령에서 비트나 워드번호가 지정된 범위를 넘거나 사용할 수 없는 외부번지를 지정하였을 경우 ON됨 Bit 3 = INPR/ OUTR 명령어 사용시 워드번호가 지정된 범위를 벗어나 사용하였을 경우 ON됨 Bit 4 = 규정되지 않은 명령이 있을 경우 ON됨 Bit 5 = 사용자 프로그램 메모리에 쓰기 에러 발생시 ON됨 Bit 6 = 기타 에러 발생시 ON됨 Bit 7 = 사용자 프로그램 메모리가 비정상일때 ON됨 Bit 8 = 입출력번지와 사용한 비트/워드/더블워드번호 오류시 ON Bit 9 = JMP명령과 CALL명령의 레이블번호가 63을 초과 하였거나 CALL에서 지정한 회로블럭(SBR~RET)이 존재하지 않을경우 또는 JMP/ CALL 이전에 LBL/ SBR명령이 있을 경우 ON. Bit10 = LBL명령의 번호가 63을 초과했거나 중복사용시 ON Bit11 = JMPS/ JMPE 명령어를 쌍으로 사용하지 않았을 경우 ON됨 Bit12 = FOR/ NEXT 명령어를 쌍으로 사용하지 않았거나 하나의 루프내에 4개이상의 Loop를 사용하였을 경우 ON됨. Bit13 = SBR/ RET 명령어를 쌍으로 사용하지 않았거나 레이블 번호가 63을 초과하였을 경우 ON됨. Bit14 = INT/ RETI 명령어를 잘못 사용했을 경우 ON됨. Bit15 = END명령이 자동으로 삽입되지 않은 경우 ON됨
SR031	W2591	-	시스템 사용 영역
SR032	W2592	CPU 에러 정보 표시	비트범위 사용 Error시 사용한 스텝번호
SR033	W2593		T/C 범위 사용 Error시 사용한 스텝번호
SR034	W2594		워드범위 사용 Error시 사용한 스텝번호
SR035	W2595		I/O 리프레쉬 사용 Error시 사용한 스텝번호
SR036	W2596		비정상코드 사용 Error시 사용한 스텝번호
SR037	W2597		프로그램 메모리
SR038	W2598		기타
SR039	W2599		시스템 동작
SR040	W2600		입출력 문법에러 발생시 사용한 스텝번호.
SR041	W2601		JMP/Call 문법 사용 Error시 사용한 스텝번호
SR042	W2602		LBL 번호 사용 Error시 사용한 스텝번호
SR043	W2603		JMPS/JMPE 문법 사용 Error시 사용한 스텝번호
SR044	W2604		FOR/NEXT 문법 사용 Error시 사용한 스텝번호
SR045	W2605		SBR/RET 문법 사용 Error시 사용한 스텝번호
SR046	W2606		INT/RETI 문법 사용 Error시 사용한 스텝번호
SR047	W2607		END명령 사용 Error시 사용한 스텝번호
SR048	W2608		시스템 사용 영역

4) SR49(W2609) ~ SR251(W2811)번지 기능 : 시스템 사용 영역

번지		기 능	상 세 내 용
SR049~ SR288	W2609~ W2848	Reserved(예약)	시스템 사용 영역

5) SR289~SR297(W2849~W2857) 번지 기능

시계(RTC)기능이 내장된 경우 시간을 맞추거나 현재시간을 저장/표시하는 영역이며 각각은 BCD로 표시, 구성됩니다.

구분	번지		내용 \ 비트번지	상 세 내 용															
				15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
현재시간	SR289	W2849	년(4자리 BCD)	○	○	×	×	×	×	×	×	×	×	×	×	×	×	×	×
	SR290	W2850	일(日):요일	○	○	×	×	×	×	×	×	○	○	○	○	○	×	×	×
	SR291	W2851	년(年):월(月)	○	×	×	×	×	×	×	×	○	○	○	×	×	×	×	×
	SR292	W2852	초(秒):00	○	○	×	×	×	×	×	×	○	○	○	○	○	○	○	○
	SR293	W2853	시(時):분(分)	○	○	×	×	×	×	×	×	○	×	×	×	×	×	×	×
시간설정	SR294	W2854	일(日):요일	○	○	×	×	×	×	×	×	○	○	○	○	○	×	×	×
	SR295	W2855	년(年):월(月)	○	×	×	×	×	×	×	×	○	○	○	×	×	×	×	×
	SR296	W2856	초(秒):00	○	○	×	×	×	×	×	×	○	○	○	○	○	○	○	○
	SR297	W2857	시(時):분(分)	○	○	×	×	×	×	×	×	○	×	×	×	×	×	×	×

주) '×' 표시는 Data가 변하는 영역 표시입니다.
년,월,일,요일 값을 저장후 F12.13 비트를 ON하면 수정이 되며, 시,분,초 값을 변경할 때는 F12.14비트 접점을 ON시켜야 변경이 됩니다.

6) SR298~SR373(W2858~W2933)번지 기능

사용자 정의 통신영역은 통신용 COM2 단자(Modular Jack형태)를 통해 다른 PLC 및 통신 가능한 타 기종과 데이터를 교환할 수 있습니다.

번지		기 능	상 세 내 용
SR298~ SR333	W2858 ~ W2893	송신 데이터 영역	36 워드,절대번지 3370
SR334~ SR369	W2894 ~ W2929	수신 데이터 영역	36 워드,절대번지 3406
SR370	W2930	송신 데이터 길이	송신 데이터의 Byte 길이
SR371	W2931	수신 데이터 길이	수신되는 데이터의 Byte 길이 정보
SR372	W2932	시작코드 보관영역	ASCII 통신시 시작코드를 보관하는 영역 (하위 Byte 만 사용)
SR373	W2933	종료코드 보관영역	

7) SR번지 기타영역 기능

정의되지 않은 SR영역은 시스템용으로 예약되어 있어 사용자가 임의로 제어하면 고장이나 오류를 일으킬 수 있습니다.

3-6. 타이머/카운터 영역(TC0-255)

1. 타이머/카운터 설정값, 현재값 고유번지

채널	설정값(SV)	현재값(PV)
0	W2048	W2304
1	W2049	W2305
2	W2050	W2306
3	W2051	W2307
4	W2052	W2308
5	W2053	W2309
6	W2054	W2310
7	W2055	W2311
8	W2056	W2312
9	W2057	W2313
10	W2058	W2314
11	W2059	W2315
12	W2060	W2316
13	W2061	W2317
14	W2062	W2318
15	W2063	W2319
16	W2064	W2320
17	W2065	W2321
18	W2066	W2322
19	W2067	W2323
20	W2068	W2324
21	W2069	W2325
22	W2070	W2326
23	W2071	W2327
24	W2072	W2328
25	W2073	W2329
26	W2074	W2330
27	W2075	W2331
28	W2076	W2332
29	W2077	W2333
30	W2078	W2334
31	W2079	W2335
32	W2080	W2336
33	W2081	W2337
34	W2082	W2338
35	W2083	W2339
36	W2084	W2340
37	W2085	W2341
38	W2086	W2342
39	W2087	W2343

채널	설정값(SV)	현재값(PV)
40	W2088	W2344
41	W2089	W2345
42	W2090	W2346
43	W2091	W2347
44	W2092	W2348
45	W2093	W2349
46	W2094	W2350
47	W2095	W2351
48	W2096	W2352
49	W2097	W2353
50	W2098	W2354
51	W2099	W2355
52	W2100	W2356
53	W2101	W2357
54	W2102	W2358
55	W2103	W2359
56	W2104	W2360
57	W2105	W2361
58	W2106	W2362
59	W2107	W2363
60	W2108	W2364
61	W2109	W2365
62	W2110	W2366
63	W2111	W2367
64	W2112	W2368
65	W2113	W2369
66	W2114	W2370
67	W2115	W2371
68	W2116	W2372
69	W2117	W2373
70	W2118	W2374
71	W2119	W2375
72	W2120	W2376
73	W2121	W2377
74	W2122	W2378
75	W2123	W2379
76	W2124	W2380
77	W2125	W2381
78	W2126	W2382
79	W2127	W2383

채널	설정값(SV)	현재값(PV)
80	W2128	W2384
81	W2129	W2385
82	W2130	W2386
83	W2131	W2387
84	W2132	W2388
85	W2133	W2389
86	W2134	W2390
87	W2135	W2391
88	W2136	W2392
89	W2137	W2393
90	W2138	W2394
91	W2139	W2395
92	W2140	W2396
93	W2141	W2397
94	W2142	W2398
95	W2143	W2399
96	W2144	W2400
97	W2145	W2401
98	W2146	W2402
99	W2147	W2403
100	W2148	W2404
101	W2149	W2405
102	W2150	W2406
103	W2151	W2407
104	W2152	W2408
105	W2153	W2409
106	W2154	W2410
107	W2155	W2411
108	W2156	W2412
109	W2157	W2413
110	W2158	W2414
111	W2159	W2415
112	W2160	W2416
113	W2161	W2417
114	W2162	W2418
115	W2163	W2419
116	W2164	W2420
117	W2165	W2421
118	W2166	W2422
119	W2167	W2423

참고사항 GPC5, WinGPC등을 사용할 경우 상기의 W레지스터는 다음과 같이 사용합니다.

채널번호	설정값(SV)		현재값(PV)		출력접점(TC)	
	PGM사용	WinGPC	PGM사용	WinGPC	PGM사용	WinGPC
0	W2048	SV000	W2304	PV000	TIM000	TC000
.....	.....	.....	.....	.....	.....	.....
255	W2303	SV255	W2559	PV255	TIM255	TC255

타이머/카운터 설정값 현재값 고유번지 (계속)

채널	설정값(SV)	현재값 (PV)
120	W2168	W2424
121	W2169	W2425
122	W2170	W2426
123	W2171	W2427
124	W2172	W2428
125	W2173	W2429
126	W2174	W2430
127	W2175	W2431
128	W2176	W2432
129	W2177	W2433
130	W2178	W2434
131	W2179	W2435
132	W2180	W2436
133	W2181	W2437
134	W2182	W2438
135	W2183	W2439
136	W2184	W2440
137	W2185	W2441
138	W2186	W2442
139	W2187	W2443
140	W2188	W2444
141	W2189	W2445
142	W2190	W2446
143	W2191	W2447
144	W2192	W2448
145	W2193	W2449
146	W2194	W2450
147	W2195	W2451
148	W2196	W2452
149	W2197	W2453
150	W2198	W2454
151	W2199	W2455
152	W2200	W2456
153	W2201	W2457
154	W2202	W2458
155	W2203	W2459
156	W2204	W2460
157	W2205	W2461
158	W2206	W2462
159	W2207	W2463
160	W2208	W2464
161	W2209	W2465
162	W2210	W2466
163	W2211	W2467
164	W2212	W2468
165	W2213	W2469

채널	설정값(SV)	현재값(PV)
166	W2214	W2470
167	W2215	W2471
168	W2216	W2472
169	W2217	W2473
170	W2218	W2474
171	W2219	W2475
172	W2220	W2476
173	W2221	W2477
174	W2222	W2478
175	W2223	W2479
176	W2224	W2480
177	W2225	W2481
178	W2226	W2482
179	W2227	W2483
180	W2228	W2484
181	W2229	W2485
182	W2230	W2486
183	W2231	W2487
184	W2232	W2488
185	W2233	W2489
186	W2234	W2490
187	W2235	W2491
188	W2236	W2492
189	W2237	W2493
190	W2238	W2494
191	W2239	W2495
192	W2240	W2496
193	W2241	W2497
194	W2242	W2498
195	W2243	W2499
196	W2244	W2500
197	W2245	W2501
198	W2246	W2502
199	W2247	W2503
200	W2248	W2504
201	W2249	W2505
202	W2250	W2506
203	W2251	W2507
204	W2252	W2508
205	W2253	W2509
206	W2254	W2510
207	W2255	W2511
208	W2256	W2512
209	W2257	W2513
210	W2258	W2514
211	W2259	W2515

채널	설정값(SV)	현재값(PV)
212	W2260	W2516
213	W2261	W2517
214	W2262	W2518
215	W2263	W2519
216	W2264	W2520
217	W2265	W2521
218	W2266	W2522
219	W2267	W2523
220	W2268	W2524
221	W2269	W2525
222	W2270	W2526
223	W2271	W2527
224	W2272	W2528
225	W2273	W2529
226	W2274	W2530
227	W2275	W2531
228	W2276	W2532
229	W2277	W2533
230	W2278	W2534
231	W2279	W2535
232	W2280	W2536
233	W2281	W2537
234	W2282	W2538
235	W2283	W2539
236	W2284	W2540
237	W2285	W2541
238	W2286	W2542
239	W2287	W2543
240	W2288	W2544
241	W2289	W2545
242	W2290	W2546
243	W2291	W2747
244	W2292	W2548
245	W2293	W2549
246	W2294	W2550
247	W2295	W2551
248	W2296	W2552
249	W2297	W2553
250	W2298	W2554
251	W2299	W2555
252	W2300	W2556
253	W2301	W2557
254	W2302	W2558
255	W2303	W2559

■ 용어의 정의

1. 번지 (= 레지스터)

외부 입,출력모듈 및 각 내부 메모리의 위치를 지정한 메모리의 주소입니다.
본 설명서에서 흔히 사용되며 하나의 번지는 그 종류에 따라 1비트나 16비트(=워드) 혹은 32비트(=더블워드)로 구별되어 설정됩니다.

2. 비트(Bit)

연산의 최소 단위. 1 또는 0의 값으로 표현됨.

3. 바이트(Byte)

8비트로 구성되고 범위는 0~255(16진수 :0 FF)까지 이므로 바이트를 사용할때는 이보다 큰 값을 취급할 수 없습니다.

4. 워드(Word)

16비트로 구성되고 범위는 0 65,535 (16진수 : 0 FFFF)이며 하나의 독립된 번지로서 사용될 수 있습니다.
plus용 PLC는 R, M, K, F, W영역을 워드 영역으로 설정하였으며 별도의 조작없이 워드 처리 가능합니다.

5. 더블워드(Double Word)

32비트로 구성되고 범위는 0 4,294,967,295 (16진수 : 0 FFFFFFFF)까지입니다
"더블워드 번지" = "시작워드번지" + "다음워드번지"로 구성됩니다.
더블워드번지는 두개의 연속하는 워드번지로 구성됩니다.

예) W003을 사용할 경우
W003(더블워드번지) = W003(시작워드번지) + W004(다음워드번지)

6. 스캔타임(Scan Time)

CPU는 RUN중 프로그램 처리뿐만 아니라, 주기적으로 통신처리, 입출력 처리, 강제 ON/OFF 값 처리등을 수행하는데 이런 주기적인 1싸이클을 1스캔이라고 하며 이때 소요되는 시간을 스캔타임이라고 합니다.
스캔타임의 대부분은 프로그램처리에 소요됩니다. 프로그램이 여러 입출력을 처리하도록 복잡하게 되어있는 경우 프로그램처리 시간이 길어지므로, 이때는 스캔타임을 고려하여 프로그램하여야 합니다.

7. 엣지(Edge)

펄스의 상승/하강시점을 말하며, 1스캔동안 ON되는 명령입니다.
OFF →ON 유지입력접점의 경우 최초 ON될때 한번의 상승엣지가 발생합니다.
ON →OFF 유지입력접점의 경우 최초 OFF될때 한번의 하강엣지가 발생합니다.

8. BCD(Binary Coded Decimal)

십진수의 1자리 (Digit)를 4비트로 표현하는 숫자체계로서 0-9 숫자로 표시됩니다.
BCD 값 변환은 16진수 값 계산처럼 하면됩니다.

예) 59(BCD)=59(HEX), 32(BCD)=(32HEX)

9. FLASH ROM

프로그램이 저장될수 있는 ROM(EEPROM)으로서 전기적으로 그 내용을 삭제할 수 있는 장점 때문에 PLC등 프로그램을 취급하는 기기에 많이 사용됩니다.

■ 레지스터 사용방법

1. R(Relay)레지스터…(비트/워드/더블워드 사용가능)

일반 입출력 모듈과 직접 연결되는 외부 입출력 레지스터를 지칭합니다.

2. M(Memory)레지스터… (비트/워드/더블워드 사용가능)

내부보조 릴레이로서 일반 연산 및 프로그램 변수로 사용됩니다. PLC의 전원이 꺼지거나 CPU가 STOP(PROG.)상태이면 레지스터값이 지워집니다.

3. W(Word)레지스터… (비트/워드/더블워드 사용가능)

M레지스터와 쓰임새가 같습니다, 전원 OFF후에도 값이 보존되지만 프로그램 다운로드 및 특수명령어에 의하여 클리어 됩니다.

● 주의 : 비트로는 사용되지 않습니다.(워드전용 레지스터)

4. K(Keep)레지스터… (비트/워드/더블워드 사용가능)

일반 연산 및 프로그램 변수로 사용될 수 있으며 전원 OFF후에도 값이 보존 됩니다.

5. F(Flag)레지스터… (비트 처리만 하십시오)

시스템이 제공하는 특수기능을 활용할 때 사용되며 PLC의 RUN / STOP(PROG.)등 시스템 상태 및 동작을 제어할 때 사용됩니다.

6. 계산값이나 입력값이 65,535(FFFFh)을 넘을때는…

65,535이상의 값을 저장, 계산할 수 있는 더블워드(명령어)를 쓰십시오. 더블워드명령을 사용하면, 4,294,967,295(2^{32})까지 확장하여 사용할수 있습니다. 워드명령어를 사용할 경우 K, M, R, W 영역은 0 65,535값만을 처리합니다.

7. 오늘 작업한 내용이나 값을 다음에 불러쓰고 싶을 때에는…

K, W영역을 사용하세요. K, W영역은 특별하게 지우지 않는한 값이 보존되며 W영역은 프로그램 다운로드 또는 특별한 명령에 의해 지워집니다.

8. A+B=C, 34X45=D, A1>C1, 대입등 수학적인 표현이 필요할 때는…

R, M, W, K영역을 사용하세요, R번지를 사용하면 계산값을 입출력 모듈에서 참조, 출력할 수 있습니다.

9. 비트의 회전, 전송등이 필요한 경우에는…

M, K, R영역을 쓰세요. 주의 : W영역은 비트연산이 않됩니다.

10. 타이머, 카운터의 설정값을 참조/수정하고 싶을 때는…

W2048부터 W2303번지 또는 SV0부터 SV255영역을 참조/ 수정하십시오.

11. 타이머, 카운터의 현재값을 참조/수정하고 싶을 때는…

1) W2304부터 W2559번지 또는, PV0부터 PV255영역을 참조/ 수정하십시오.
이영역은 STOP(PROG.)상태에서도 값이 유지됩니다.

2) 카운터의 현재값(PV)은 전원 OFF후에도 유지됩니다.

3-7. CPU 운전 모드(mode)의 개요

CPU 운전모드란?

CPU가 외부의 RUN /REMOTE/ PROG 스위치를 읽은 상태, 자체적으로 시스템을 점검한 PLC의 현재 상태를 말합니다. 아래의 4가지의 운전모드를 가집니다.

RUN 모드(운전상태)

RUN 모드에서 plus용 CPU 유니트는 외부접점의 신호를 읽은후 RAM에(또는 FLASH ROM) 저장한 사용자 프로그램을 실행합니다, 또한 프로그램 결과대로 매 스캔마다 외부로 출력신호를 내보냅니다.

STOP 모드(정지상태)

STOP 모드에서는 실행중인 사용자 프로그램이 정지되며, 이와 동시에 모든 출력신호를 OFF 시킵니다.
STOP운전모드에서는 프로그램 수정/ 삭제/ 전송이 자유롭습니다.
(내장되어 있는 FLASH 메모리에 프로그램을 저장할수 있는 유일한 모드입니다.)

PAUSE 모드(일시정지상태)

PAUSE 모드에서는 사용자 프로그램을 1스캔동안 운전할수 있고, 그 결과값을 유지할수 있는 모드입니다. 1스캔씩 프로그램수행이 가능하므로 프로그램 디버깅시 편리합니다. STOP 모드와 거의 동일하나 Data를 초기화시키지 않습니다.

ERROR 모드(에러상태)

ERROR 모드는 Plus용(제품코드뒤에 "p"가 붙은 CPU) CPU 유니트가 자체진단 하여 내부 오류를 발견할경우, 발생되는 모드로서, 정상적인 기능 수행이 불가능합니다. 에러가 발생되면 CPU는 프로그램 실행을 중단하고 모든 출력을 OFF시킵니다. 에러모드가 발생하면, 사용자는 에러코드를 검색하여 적절한 해제조치를 취한후 전원을 OFF→ON시키거나, 모드전환스위치를 PROG.상태에 둔후에 INITIALIZE KEY를 눌러서 Error를 해제할 수 있습니다.

■ CPU 모드 전환스위치별 운전모드기능

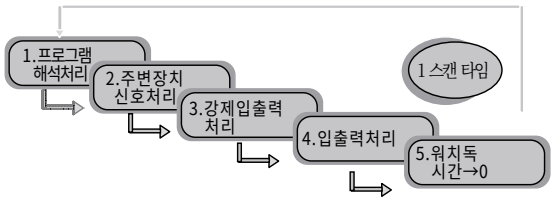
☼ :점등 ◑ :점멸 ● :소등

모드전환 스위치	운전 모드	LED상태 RUN LED	LED상태 PROG LED	프로그램 변경	데이터 변경	INITIALIZE 스위치	전원OFF → ON시 운전모드
RUN	RUN	☼	●	불가능	가능	0	RUN
	STOP	●	☼	불가능	가능	0	
REMOTE	RUN	☼	☼	가능	가능	0	RUN
	PAUSE	◑	☼	가능	가능	0	PAUSE
PROG.	STOP	●	☼	가능	가능	0	STOP

- PROG. LED가 점등되면 사용자 프로그램의 변경이 가능하다는 것입니다.
- PROG. 모드일때, INITIALIZE 스윗치를 누르면 ERROR를 CLEAR시킵니다.
- 모드 전환스위치가 REMOTE일때는 전원 OFF →ON시 운전모드가 기억됩니다.
- 사용자 프로그램 디버깅시 모드전환스위치를 REMOTE에 두면 편리합니다.

3-8. CPU 처리순서

프로그램 처리순서



■ PLC의 프로그램 처리과정을 나타냅 니다.
이때 CPU는 1→5 과정을 주기적으로 반복하는데
이와같은 1싸이클을“1스캔타임”이라고 합니다.

1. 프로그램 해석처리 과정
작성된 프로그램의 선두스텝부터 최종스텝까지 실행하고 외부출력 및 내부 출력을 내부메모리(Working RAM)에 저장합니다.

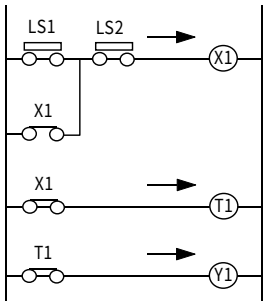
2. 주변장치 신호처리 과정
통신모듈 또는 주변장치로부터 데이타를 내부 메모리에 저장합니다.

3. 강제입출력처리 과정
강제 입출력 설정 비트가 존재할경우 해당비트를 ON/ OFF시킵니다.

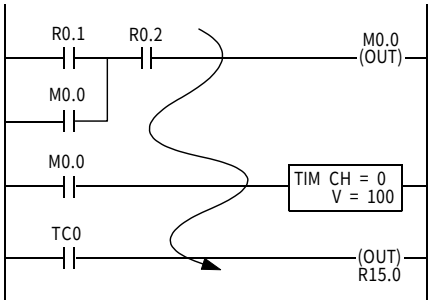
4. 입출력처리 과정
외부입출력의 ON/OFF상태를 보관하여 다음 스캔의 입력으로 사용합니다.
(정확한 처리를 위해서는 입력을 1스캔타임이상 지속되도록 하여야 합니다.)
프로그램 처리결과를 내부메모리에서 외부로 보냅니다.

5. 워치독 시간 초기화과정
위치독 경과치를 0으로 합니다. (다음 스캔까지의 워치독 계산 싯점이됨)

아래는 릴레이제어반과 PLC의 시퀀스 처리과정의 다른점을 보여줍니다. 즉 릴레이제어반은 모든시퀀스가 동시에 병렬처리되는 반면, PLC는 프로그램의 처음스텝부터 최종스텝까지 주기적으로 직렬처리됨을 보여줍니다.



릴레이 제어반의 처리
(병렬처리)



PLC프로그램 처리
(직렬처리)

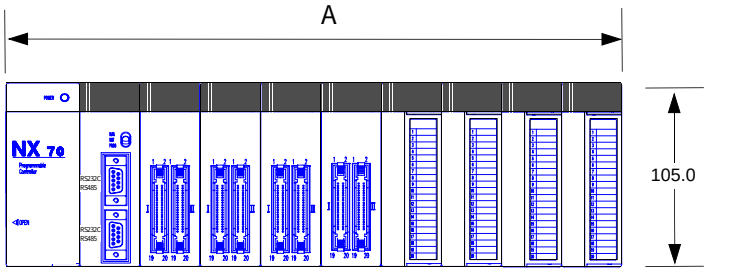
4 장

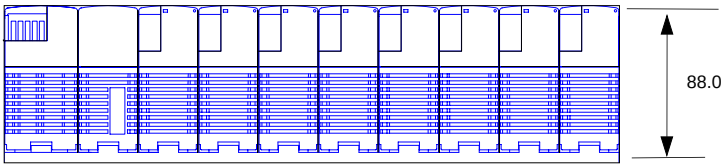
설치와 배선

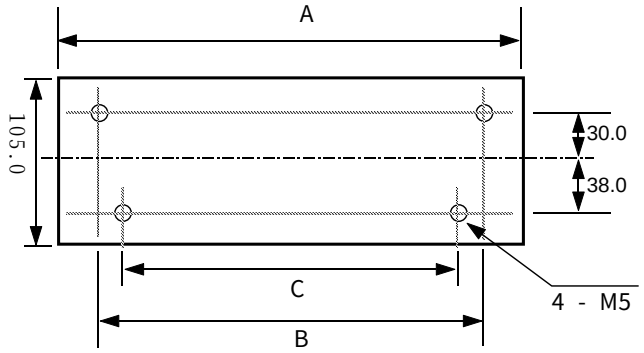
4-1. 설 치

(1) 설치공간과 설치환경

■ 외형 치수도(단위 : mm)





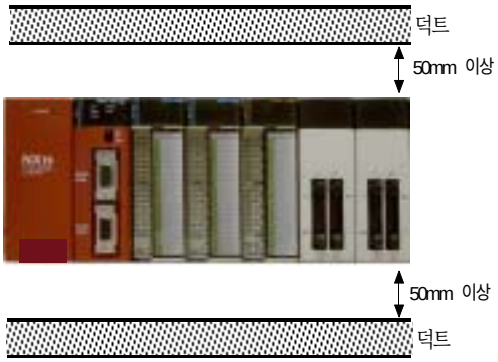


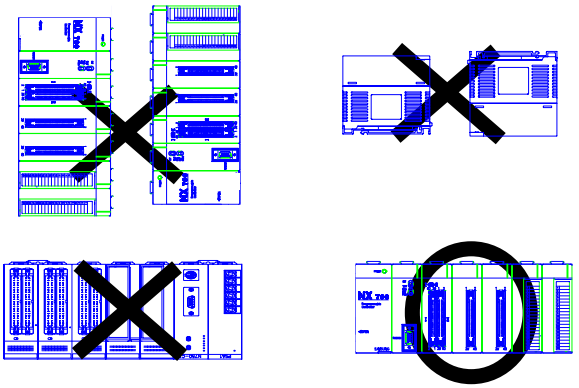
단위 (mm)

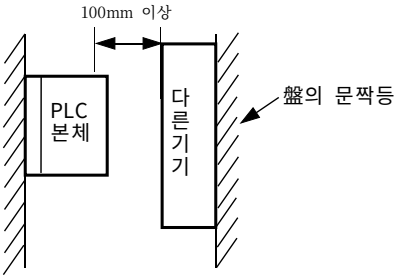
슬롯 타입	제품 번호	치 수 (A)	치 수 (B)	치 수 (C)
2 슬롯형	NX70-BASE02	157.2	128.6	114.6
3 슬롯형	NX70-BASE03	192.4	163.8	149.8
5 슬롯형	NX70-BASE05	262.8	234.2	220.2
6 슬롯형	NX70-BASE06	298.0	269.4	255.4
8 슬롯형	NX70-BASE08	368.4	339.8	325.8
10 슬롯형	NX70-BASE10	438.8	410.2	396.2
12 슬롯형	NX70-BASE12	508.6	480.6	466.6

■ 판넬(Control Box)내 설치방법

- 통풍 스페이스를 확보하기 위해, 상하부는 다른기기와, 배선 닥트등에서 충분히 거리를 이격시켜 주십시오.
- 세로로 세워서 설치하거나 수평으로 설치한다거나 하면 PLC내부의 이상발열의 원인이 되므로 하지 마십시오.
- 히터, 트랜스, 대용량 저항등의 발열량이 큰 기기의 바로위에는 설치하지 마십시오.
- 각 유니트(모듈)의 표면은 방사노이즈의 영향을 피하기 위해 동력선, 전자 개폐기기등과 100mm이상 거리를 이격시켜 주십시오. 특히, 제어반의 문짝에 그와 같은 기기를 설치할 경우, 그 거리를 주의해 주십시오.

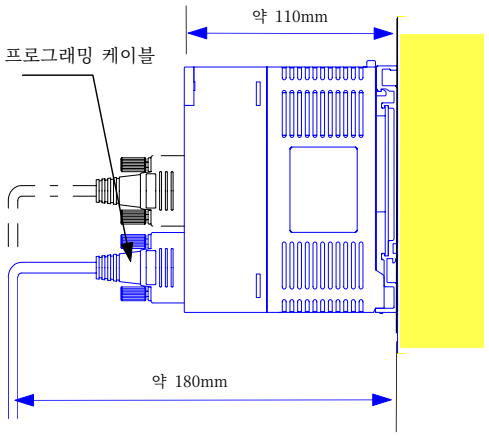






■ 툴(TOOL) 배선을 위한 여유

프로그래밍 툴이나 케이블을 접속하기 위해 CPU 유니트의 설치면에서 180mm이상 스페이스를 확보해 주십시오.

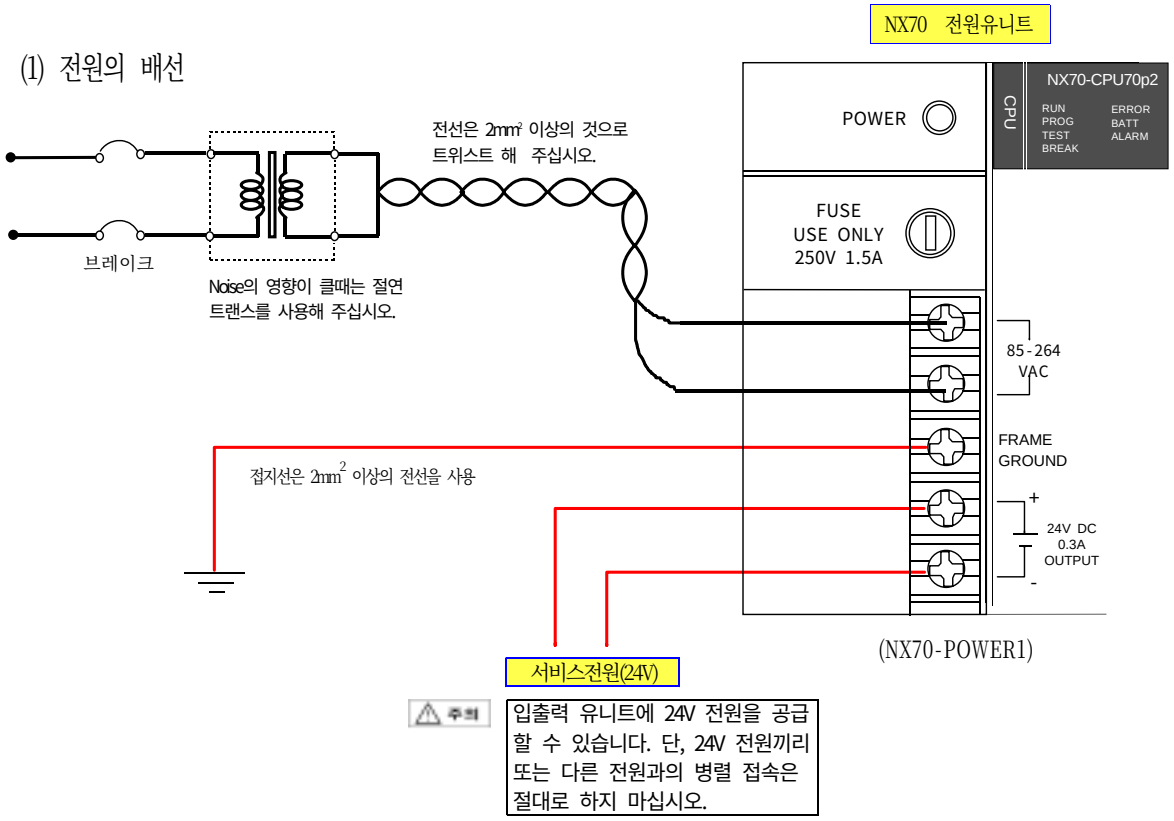


■ 설치시 피해야 할 장소

- 주위온도가 0 55°C의 범위를 넘는 장소
- 주위온도가 30 85%RH의 범위를 넘는 장소
- 급격한 습도변화로 결로될 우려가 있는 장소
- 부식성 가스, 가연성 가스의 환경중
- 진애, 철분, 염분이 많은 장소
- 벤젠, 신너, 알코올 등의 유기용제 또는 암모니아 및 가성소다등의 강알칼리 물질이 부착될 우려가 있는 장소 또는 그 환경장소
- 진동 또는 충격이 심한 장소
- 직접 일광이 닿는 장소
- 물, 기름, 약품등이 닿을 우려가 있는 장소
- 고압선, 고압기기, 동력선, 동력기기 외에, 커다란 개폐서지를 발생시키는 기기로부터는 가능한 분리하여 설치해 주십시오.(특히 100mm 이내)

4-2. 전원의 배선

(1) 전원의 배선



■ 전원전압은 허용전압 범위에서

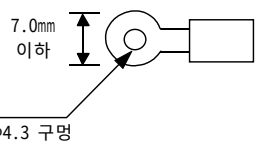
제품명	정격입력전압	허용전압범위
NX70-POWER1 NX70-POWER2	AC110-220V	AC85 -264V

■ 전원유니트 단자나사

- 단자나사는 M3.5 나사를 사용해 주십시오.
- 단자의 배선은 압착단자를 사용할 것을 권합니다.

● 개방형 단자

● 환형 단자



■ 전원공급은 2㎟의 트위스트 전선으로

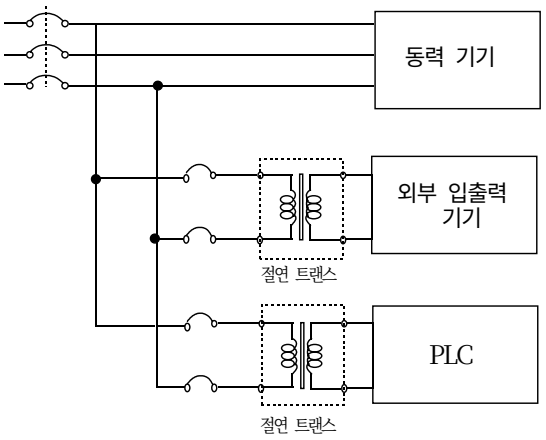
- 전선은 전압강하를 작게 하기위해 2㎟ 이상의 전선을 사용해 주십시오.
- 노이즈의 영향을 적게 하기위해 전선은 트위스트로 사용해 주십시오.

■ 적합한 압착단자

- 나환형 단자 (O형 Lug)
- 절연지가 붙은 나환형 단자
- 앞쪽이 개방된 단자(Y형 Lug)

■ 전원계통의 분리

● PLC, 입출력기기, 동력기기의 배선은 아래 배선과 같이 각각 계통을 분리해 주십시오.



■ 노이즈가 많을때는 절연트랜스 사용

● 전원에는 노이즈가 적은 것을 사용해 주십시오.

● 노이즈가 많을 경우는 위의 그림과 같이 절연트랜스로 노이즈를 감쇠시키고 나서 전원을 공급해 주십시오.

(2) 접지

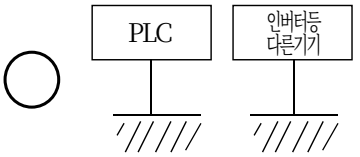
■ 노이즈 영향이 클때는 접지사용

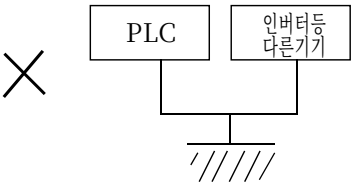
● Frame Ground 단자(FG)는, 大地 접지단자로 마더보드의 금속부와 접속되어 있습니다.

● 전선은 2mm² 이상의 것을 사용하고, 접지저항 100Ω 이하의 제3종 접지로 해 주십시오.

● 接地点은 가능한 한 PLC 근처로 하고, 접지선의 길이를 짧게 해 주십시오.

● 접지를 다른기기와 공용하면 역효과가 날 경우가 있으므로 전용 접지로 해 주십시오.





4-3. 입출력의 배선

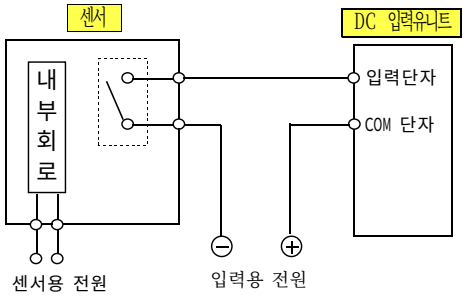
(1) 입력측의 배선

■ 입력유니트 배선의 체크포인트

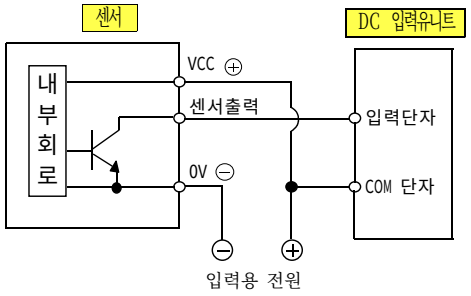
1. 유니트의 종류에 따라 동시 ON점수에 제한되는 것이 있습니다.
   각 입력유니트의 사양항목에서 확인바라며, 특히 주위온도가 높은곳에서 사용할 경우는 주의하십시오.
2. 센서등 입력기기의 종류에 따른 접속방법은 아래 항목과 같습니다.

■ 광전센서, 근접센서의 접속방법

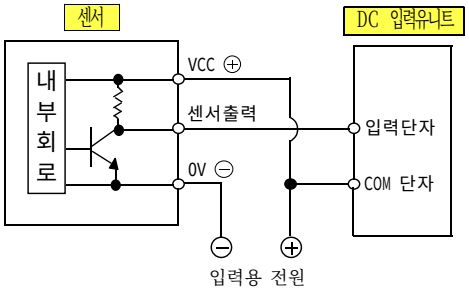
1) 릴레이출력 타입



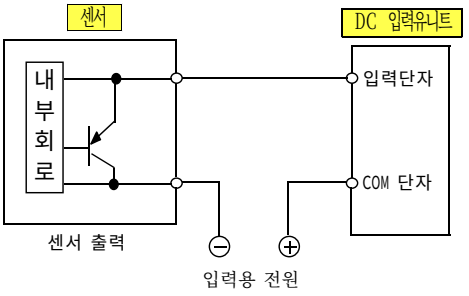
2) NPN 오픈콜렉터 출력타입



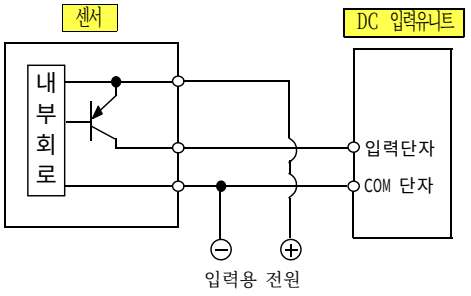
3) 전압출력 타입



4) 2선식 타입

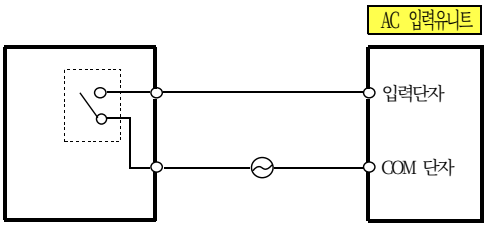


5) PNP 오픈콜렉터 출력타입

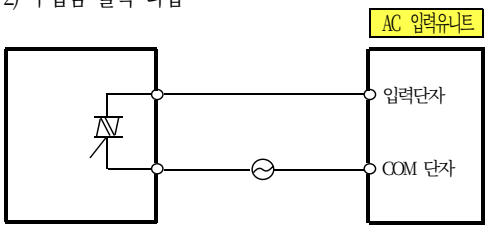


■ AC 입력기기와의 접속예

1) 유접점출력 타입

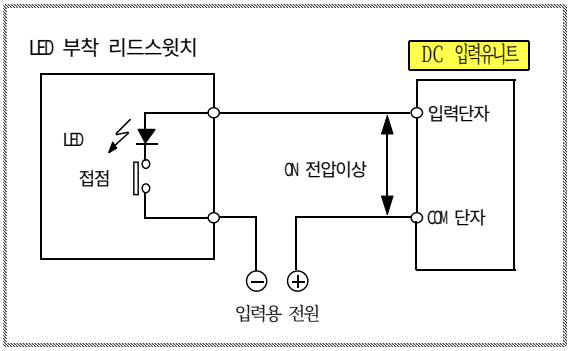


2) 무접점 출력 타입



■ LED가 붙은 리드스위치 사용시의 주의점

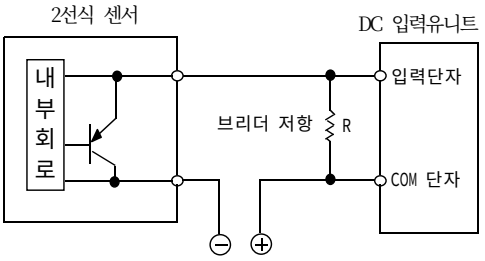
LED가 붙은 리드스위치등, 입력접점에 직렬 LED가 내장 되어있는 경우라도 PLC입력단자에는 ON전압 이상의 전압이 걸리도록 해 주십시오. 특히 직렬로 복수의 스위치를 접속할 경우는 주의하시기 바랍니다.



■ 2선식 센서를 사용할 때의 주의점

2선식의 광전스위치, 근접스위치를 사용할 경우는 누설전류의 영향으로 PLC로의 입력이 OFF가 되지 않을 경우는 아래와 같이 브리더 저항을 접속해 주십시오.

DC 12-24V 타입의 입력유니트의 경우
(OFF전압 2.5V, 입력 임피던스 3kΩ)



I : 센서의 누설 전류(mA)
R : 브리더 저항치(kΩ)

입력의 OFF전압은 2.5V이므로 COM 단자, 입력단자간의 전압이 2.5V이하가 되도록 R의 값을 정합니다.
입력 임피던스는 3kΩ

$$I \times \frac{3R}{3+R} \leq 2.5 \text{에 의해} \quad R \leq \frac{7.5}{3I-2.5} (k\Omega)$$

저항의 와트수 W는

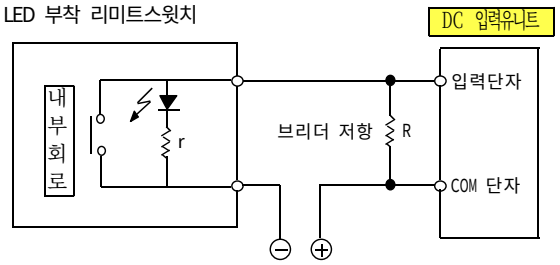
$$W = \frac{(\text{전원 전압})^2}{R}$$ 으로 구해지며 통상,

이값의 3 5배에서 선정해 주십시오.

■ LED가 부착된 리미트스윗치 사용시의 주의점

LED가 붙어있는 리미트 스윗치를 사용할 경우, 누설전류의 영향으로 PLC로의 입력이 OFF하지 않는다거나 또는 LED가 잘못 점등할 경우는 아래와 같이 브리더저항을 접속해 주십시오.

DC 12-24V 타입의 입력 유니트의 경우
(OFF 전압 2.5V, 입력 임피던스 3kΩ)



r : 리미트스위치의 내부저항(kΩ)
R : 브리더 저항치(kΩ)

입력의 OFF 전압은 2.5V이므로 전원전압이 24V일 때,

$$I = \frac{24 - 2.5}{r} \quad \text{이상}$$

흐르도록 R의 값을 정합니다. I를 구하고 앞페이지의 2선식 센서를 사용할 때와 똑같이 구해집니다.

$$R \leq \frac{7.5}{3I - 2.5}\ (k\Omega) \qquad W = \frac{(\text{전원 전압})^2}{R} \times (3 \sim 5\text{배})$$

(2) 출력측의 배선

■ 출력유니트 배선의 체크포인트

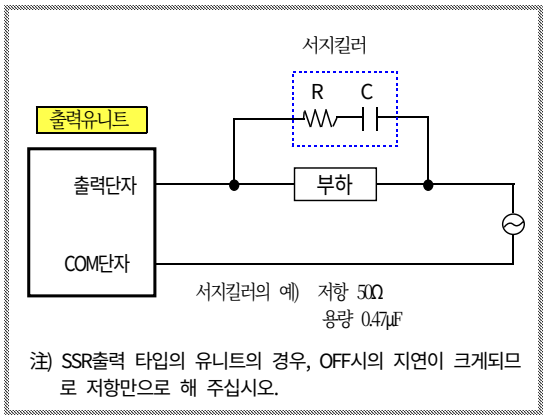
1. 유니트의 종류에 따라 동시 ON점수 또는 부하전류에 제한이 있는 것이 있습니다. 각 유니트의 사양을 참조하십시오. 특히 주위온도가 높은 곳에서 사용하는 경우는 주의 바랍니다.
2. 유도성 부하, 용량성 부하등에 대해서는 아래와 같이 보호회로를 설치해 주십시오.
3. 출력유니트에는 Common당 전류제한이 있는것이 있으므로 그 범위내에서 사용해 주십시오.

■ 유도부하의 보호회로

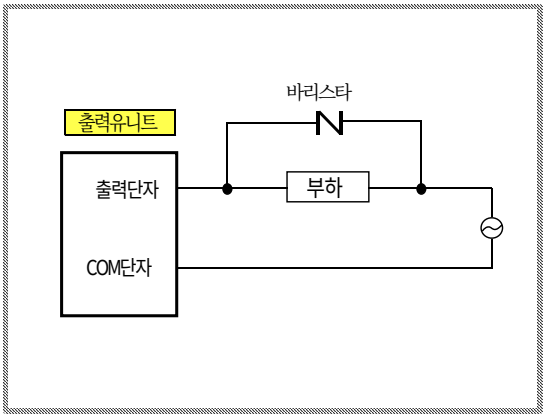
● 유도부하인 경우는 부하와 병렬로 보호회로를 마련해 주십시오.

● 특히 릴레이출력 타입으로 DC 유도부하를 개폐하는 경우는, 보호회로의 유무가 수명에 크게 영향을 주므로 반드시 부하의 양쪽에 다이오드를 설치해 주십시오.

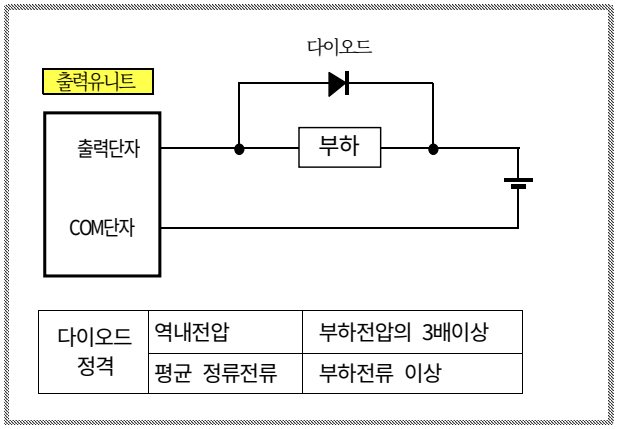
1) AC부하의 경우 (릴레이출력 타입)



注) SSR출력 타입의 유니트의 경우, OFF시의 지연이 크게되므로 저항만으로 해 주십시오.



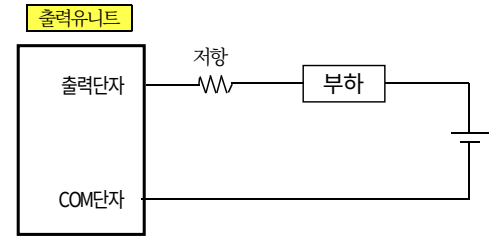
2) DC부하의 경우

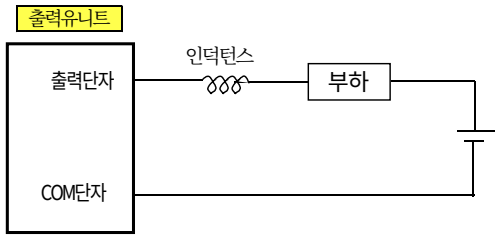


다이오드 정격	역내전압	부하전압의 3배이상
	평균 정류전류	부하전류 이상

■ 용량성 부하를 사용할 때의 주의점

돌입전류가 큰 부하를 접속할 경우는 그 영향을 작게 하기 위해 다음과 같이 보호회로를 준비해 주십시오.



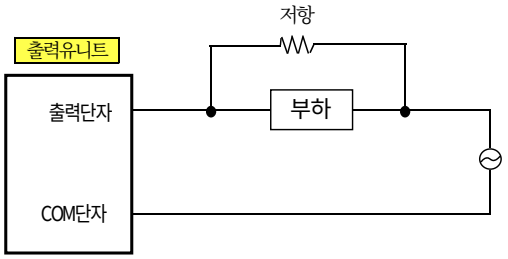


■ 과부하 보호는 외부퓨즈로

- 퓨즈가 내장되어 있는 유니트는 출력의 단락시등에 소손을 방지하는 것이 목적입니다.
- 퓨즈가 내장되어 있는 유니트라도 과부하에 대해 각 소자의 보호는 할 수 없으므로 1점마다 외부에 퓨즈를 설치할 것을 권합니다.
- 단, 단락시등의 경우에는 출력유니트의 소자를 보호할 수 없는 경우가 있습니다.

■ 누설전류에 대한 주의점

SSR 출력유니트에서는 저전류 부하의 경우에 누설전류가 원인이 되어 부하가 OFF되지 않는 경우가 있습니다. 그러한 경우는 부하와 병렬로 저항을 접속해 주십시오.

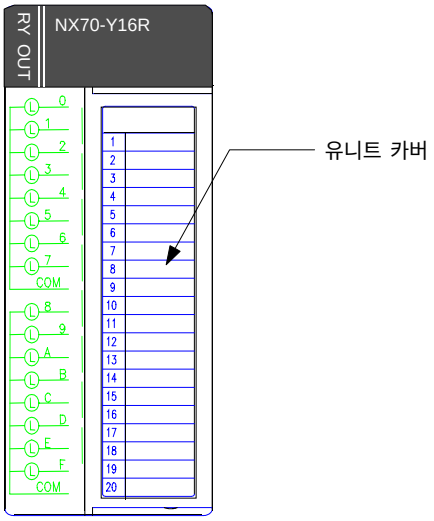


(3) 입출력유니트 공통의 주의사항

■ 입력/ 출력/ 동력선은 각각 분리

- 입력배선과 출력배선은 가능한 한 동력선과 멀리하여 배선해 주십시오. 또한, 동일 덕트를 통과시키거나 함께 묶지 말아 주십시오.
- 입출력 배선과 동력선, 또는 고압선과는 100mm 이상 거리를 떨어지게 해 주십시오.

■ 유니트 커버



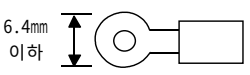
4-4. 단자대 타입 유니트의 배선

■ 압착단자는 M3.0용

NX70 PLC의 입출력 모듈(단자대 타입)의 단자대는 M3.0의 단자나사를 사용하고 있습니다. 단자와의 배선은 아래와 같은 압착단자를 사용할 것을 권합니다.

● 개방형 단자

● 환형 단자



4-5. 커넥타 타입 유니트의 배선

(1) 배선 방법

■ 결선 방법

NX70 PLC 32점 입력모듈 (NX70-X32D, NX70-X32D1) 및 32점 출력모듈 (NX70-Y32T, NX70-Y32P)은, 20핀의 MIL Type 커넥타를 사용하고 있습니다.

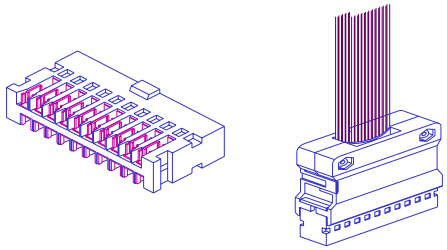
외부기기와 접속할 수 있는 방법은

1) 낱개의 Pin으로 소켓에 각각 끼워 접속하는 방법과

2) 플렛 케이블을 이용한 하네스처리를 하는 방법의 2가지가 있습니다. (제품으로 판매하고 있음)

1) PIN 타입의 접속방법

낱개의 PIN으로, 소켓에 각각 끼워 접속하는 방법으로 배선방법은 다음페이지에 소개합니다.

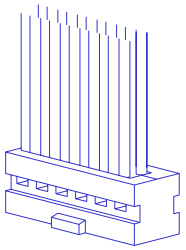


제품명	제품코드	제품사양
I/O 커넥타 ASS'Y (Pin 타입)	CPL8880	· 20핀 1) PIN 20개 2) 커넥타 후드 3) 소켓이 포함되어 있습니다.

적용 제품	
32점 입력모듈용	NX70-X32D (DC IN 32점, 12-24V) NX70-X32D1(DC IN 32점), 24V)
32점 출력모듈용	NX70-Y32T (TR OUT 32점, NPN) NX70-Y32P (TR OUT 32점, PNP)

2) 하네스 접속방법 (플랫케이블 커넥타 이용)

본 제품은 20핀 플랫케이블 커넥타로 처리되어 있으며, 끝부분에 압착단자가 20가닥 접속되어 있고, 길이가 1.5m로 구성되어 있습니다. 유니트에 바로 접속하여 사용할 수 있습니다.



제품명	제품코드	적용 사양
케이블 ASS'Y	CPL8800	DC IN 32점, 커넥타 하네스케이블 1.5m
	CPL8810	TR OUT 32점 커넥타 하네스케이블 1.5m

적용 제품	
32점 입력모듈용	NX70-X32D (DC IN 32점, 12-24V) NX70-X32D1(DC IN 32점), 24V)
32점 출력모듈용	NX70-Y32T (TR OUT 32점, NPN) NX70-Y32P (TR OUT 32점, PNP)

■ 참고사항 (제품코드 변환표)

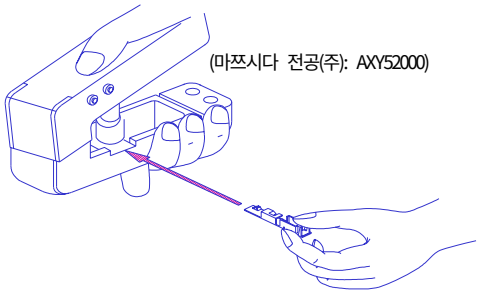
제품명	제품 번호	
	신규 코드	(구코드)
입력 유니트	NX70-X16D	(CPL93023)
	NX70-X16D1	(CPL93033)
	NX70-X32D	(CPL93024)
	NX70-X32D1	(CPL93034)
	NX70-X16A110	(CPL93043)
	NX70-X16A220	(CPL93053)
출력 유니트	NX70-Y8R	(CPL93202)
	NX70-Y16R	(CPL93103)
	NX70-Y16RV	(CPL93203)
	NX70-Y16T	(CPL93483)
	NX70-Y32T	(CPL93484)
	NX70-Y32P	(CPL93584)
	NX70-Y16SSR	(CPL93703)
입출력 혼합유니트	NX70-XY16	(CPL93088)
	NX70-XY32	-

■ PIN 타입 접속시 압접소켓 사용방법

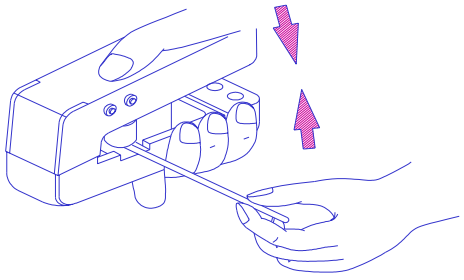
피복은 그대로 다이렉트로 압접하며, 배선의 수고를 줄일 수 있습니다.

● 배선방법

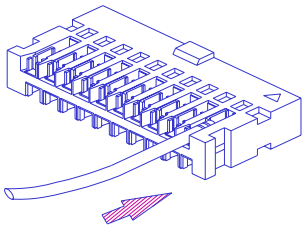
(1) 콘택트를 캐리어에서 꺾어 압접공구에 셋트해 주십시오.



(2) 피복된 채로의 전선을 맞닿을 때까지 삽입하고, 공구를 가볍게 쥐어 주십시오.

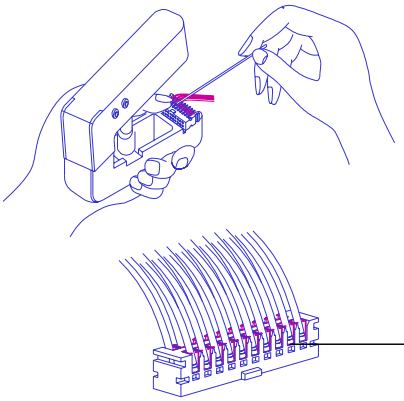


(3) 압접후 전선을 하우징에 삽입해 주십시오.



※ 배선실수시, 다시 할수 있는 콘택트제거 핀

배선미스, 케이블 압접에러인 경우, 공구에 접속되어 있는 콘택트 제거핀을 사용하여 콘택트를 뺄 수가 있습니다.



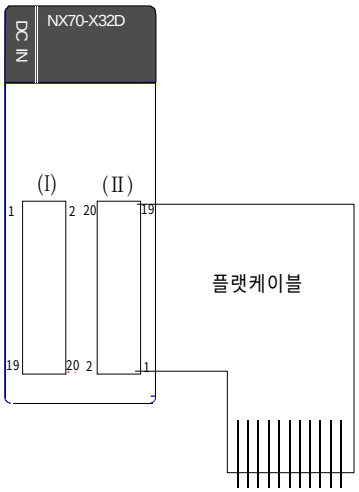
이 부분에 컨택트 제거용 핀이 닿도록 하우징을 압접 공구에 대고 누른다.

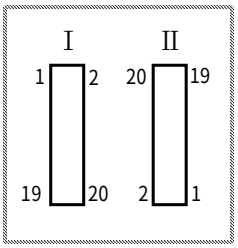
(2) 플랫케이블 커넥타의 접속

■ 플랫케이블 커넥터를 사용할 경우의 주의

플랫케이블 커넥터를 사용하여 직접 결선할 경우,
케이블의 No.와 I/O 번호의 관계는 아래와 같이 대응되므로 잘 확인해 주십시오.

● 32점의 커넥타 타입의 접속





제품명	제품코드	적용 모듈
플랫케이블 (1.5m)	CPL8800	DC IN 32점용 (NX70-X32D) (NX70-X32D1)
	CPL8810	TR OUT 32점 (NX70-YX32T) (NX70-YX32P)

● 플랫케이블 No.와 I/O 대응표 (32점)

커넥터 (I)

유니트의 기호	NX70-X32D NX70-X32D1	NX70-Y32T NX70-Y32P
I 1	X 0	Y 0
I 2	X 1	Y 1
I 3	X 2	Y 2
I 4	X 3	Y 3
I 5	X 4	Y 4
I 6	X 5	Y 5
I 7	X 6	Y 6
I 8	X 7	Y 7
I 9	X 8	Y 8
I 10	X 9	Y 9
I 11	X A	Y A
I 12	X B	Y B
I 13	X C	Y C
I 14	X D	Y D
I 15	X E	Y E
I 16	X F	Y F
I 17	COM1	+
I 18	COM1	+
I 19	COM2	COM
I 20	COM2	COM

커넥터 (II)

유니트의 기호	NX70-X32D NX70-X32D1	NX70-Y32T NX70-Y32P
II 20	COM2	COM
II 19	COM2	COM
II 18	COM1	+
II 17	COM1	+
II 16	X1F	Y1F
II 15	X1E	Y1E
II 14	X1D	Y1D
II 13	X1C	Y1C
II 12	X1B	Y1B
II 11	X1A	Y1A
II 10	X19	Y19
II 9	X18	Y18
II 8	X17	Y17
II 7	X16	Y16
II 6	X15	Y15
II 5	X14	Y14
II 4	X13	Y13
II 3	X12	Y12
II 2	X11	Y11
II 1	X10	Y10

커넥터 (I)과 (II)의 핀 방향이 반대이므로 주의하여 접속해 주십시오.

4-6. 안전 대책

■ 시스템 설계상의 주의

PLC를 사용한 시스템에서는 다음과 같은 요인에 의해 오동작을 일으키는 수가 있습니다.

① PLC의 전원과 입출력기기 그리고 동력기기의 개시와 종료차이

② 순간정전에 의한 응답시간의 차이

③ PLC본체, 외부 전원, 다른 기기의 이상일 때 이와 같은 오동작이 시스템 전체의 이상 또는 사고로 이어지지 않도록 다음과 같은 안전 대책을 실시해 주십시오.

■ 인터록 회로는 PLC 외부에도

모터의 정전, 역전등 상반되는 동작을 제어할 경우는 PLC 외부에도 인터록회로를 설치해 주십시오.

■ 비상정지 회로도 PLC 외부에

출력기기의 전원을 끊는 회로를 PLC 외부에 설치해 주십시오.

■ PLC의 기동은 다른기기보다 늦게 (전원 시퀀스)

PLC의 기동은, 입출력 기기, 동력기기가 시작되고 나서 기동하도록 해 주십시오.

- 방법①…… PLC의 전원이 시작되고나서 RUN모드로 해 주십시오.
- 방법②…… 타이머 회로를 마련하여 PLC의 기동을 지연시켜 주십시오.

■ 워치독 타이머

- 워치독 타이머는 프로그램 이상 또는 하드웨어 이상을 검출하는 타이머로, 스캔시간이 640ms 이상이 되었을 경우에 ON됩니다.
- 워치독 타이머가 작용하면 ALARM LED가 점등됨과 동시에 전원 유니트의 ALARM 접점이 ON되고, 출력 유니트의 모든 출력을 OFF로 하여 정지상태가 됩니다. (프로그래밍 툴등과의 통신도 무처리 상태가 됩니다.)

4-7. N-plus CPU 모듈 통신사양

* PGM-500은 RS232C, RS485 둘다 지원합니다.

접속 규격	RS485	RS232C	비고
전송거리(max)	1.2㎞	15m	
전송속도	38,400, 19,200, 9,600, 4,800 bps		Dip 스위치 조정
동기방식	Half Duplex Asyncronous/ Polling		
패리티	NO Parity		
Stop비트	1 Stop bit		
전송케이블	Twisted Pair Cable		Shield Cable
접속주변기기	*PGM-500	IBM-PC (RS-232C) *PGM-500	

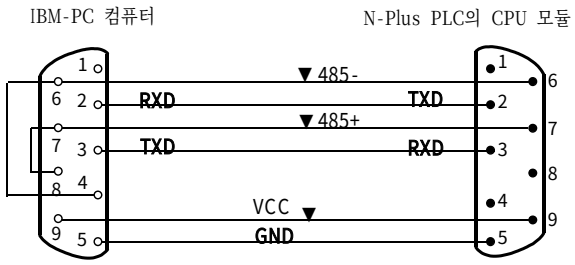
*1. N-plus CPU란, SPC10, N70plus, N700plus, NX7, NX70(CPU70p1, CPU70p2), NX700(CPU700p) PLC를 말합니다. (WinGPC S/W 사용)

참고사항

■ N-plus용 CPU 모듈은 기존의 SPC 시리즈와 통신 Protocol이 동일합니다.

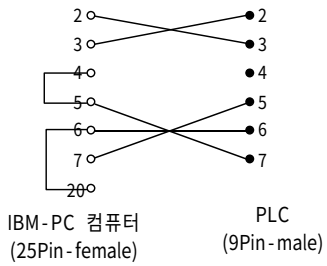
■ RS232C/ RS485 케이블 결선도 : CPL5530(2m), CPL5531(5m)

CPL5530(2m)
CPL5531(5m)



RS-232C 통신을 하고자 할때는 485+, 485-, VCC 신호를 연결하지 않아도 통신을 할수 있습니다. ▼ 표시된 배선

■ 참고사항 (25Pin-9Pin) 결선도



4-8. EEPROM 백업기능(NX70-CPU70p1, NX70-CPU70p2, NX-CPU700p, N70plus, N700plus CPU용) 사용방법

(1) EEPROM Backup 기능이란?

EEPROM(Electric Erease Programmable Read Only Memory)은 전원(밧데리)이 없어도 데이터가 지워지지 않으며, 전원을 인가하여 데이터를 지우거나 기록할 수 있습니다.

이 기능을 활용하여 평소에는 전원이 없어도 PLC의 프로그램을 영원히 보존하기 위한 기능으로 활용하고, 프로그램 수정이나 저장할 때 전원을 인가하여 기존 프로그램을 지우고 새로운 프로그램을 기록하는 역할을 합니다.

(2) 적용모델

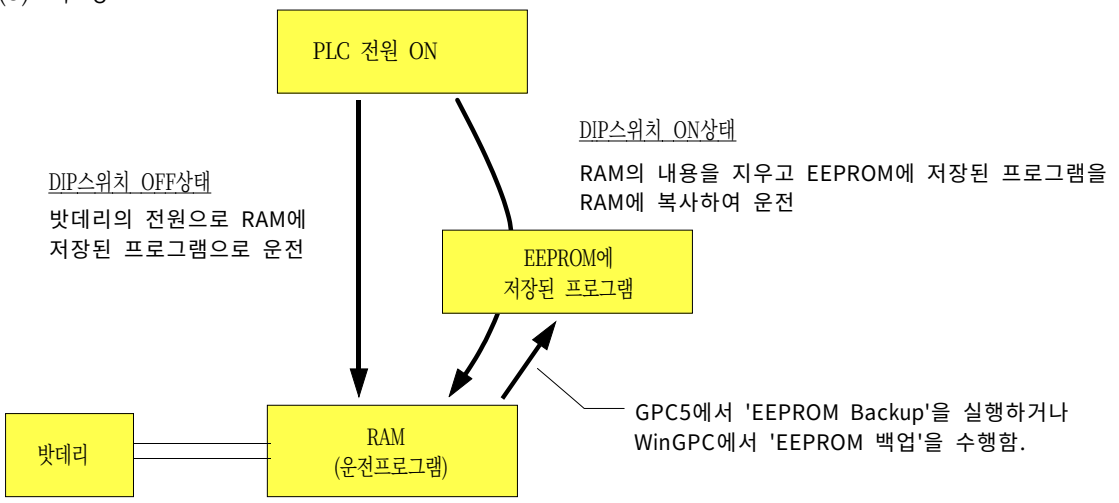
■ EEPROM의 종류는 특성에 따라 다양하게 구별이 됩니다.

PLC NX70(NX70-CPU70p1 및 NX70-CPU70p2) PLC, NX700(NX-CPU700p) PLC 그리고 N70plus, N700 plus용 CPU는 플래시 메모리(Flash Memory)를 사용합니다.

■ 이 기능은 사용하기 쉽고, 보관이나 전송하기 쉬운 장점이 있어 널리 활용되고 있으며, NX700 (NX-CPU700p) 및 NX70 PLC(NX70-CPU70p2), 그리고 N70plus(CPL9216)에서는 기본으로 내장되어 있고, N700plus시리즈는 옵션품으로 CPL7203이라는 모델을 사용하고 있습니다.

■ EEPROM은 29EE512를 사용하며, 플래시 메모리는 기록(Write) 가능한 회수(최소 3000회 이상)가 존재하며, 이것을 넘을 때는 메모리를 교환해야 하므로 주의하시기 바랍니다.

(3) 기 능



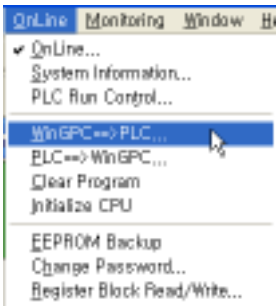
■ 백업기능은 WinGPC, PGM-500 등을 이용하여 수행합니다.

1) WinGPC 3.xx에서 사용방법

- PLC와 Online 접속 합니다.
- 완성된 프로그램을 PLC에 저장(Download : WinGPC => PLC)합니다.
- 온라인'메뉴에서 'EEPROM 백업(E)'을 선택하면 됩니다.

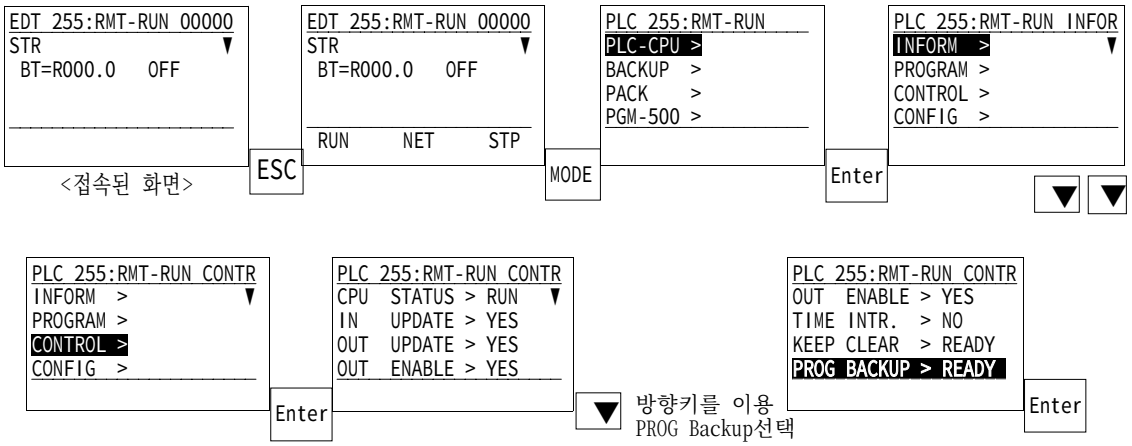
2) WinGPC 4.xx 에서 사용방법 (4.xx버젼에서도 WinGPC3 버젼과 사용법이 동일합니다.)

- OnLine 접속.
- 완성된 프로그램을 PLC에 저장(Download : WinGPC => PLC)
- '온라인'메뉴에서 'EEPROM 백업(E)'을 선택하면 됩니다.



3) PGM-500 에서 사용방법

- 접속후 PLC 제어상태로 만들고, PROG Backup을 실행합니다.

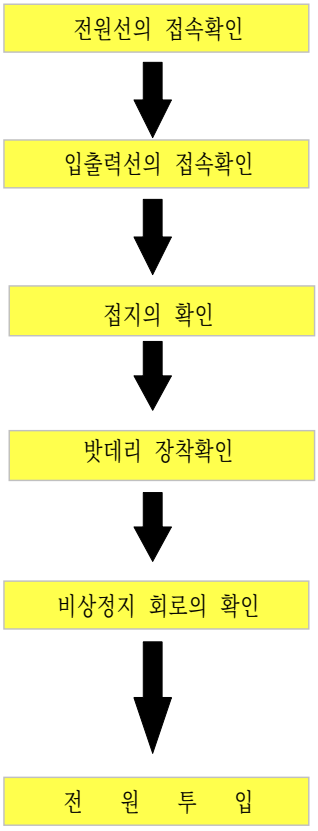


5장

시운전과 이상시 조치사항

5-1. 시운전에 앞선 확인사항

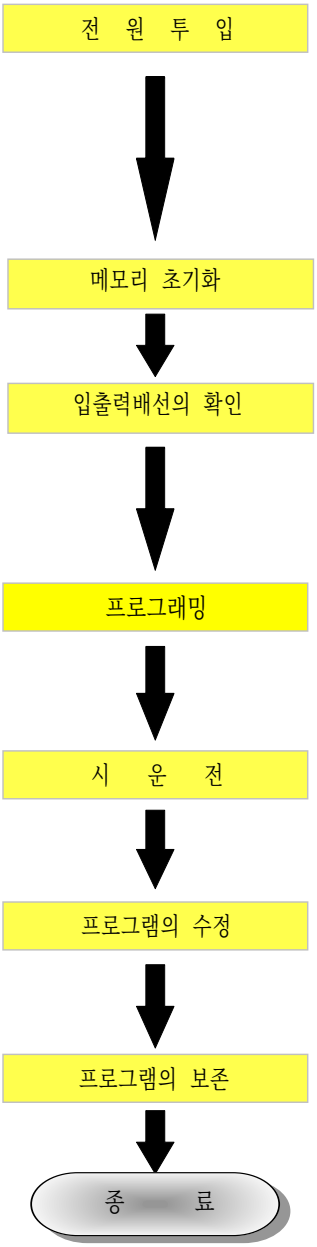
PLC의 입출력 배선을 완료하고 전원을 투입하기 전에 아래항목을 확인해 주십시오.



확 인 사 항	내 용
전원선 및 입출력선의 접속확인	1. 배선은 올바른가? 2. 단자대의 나사는 헐겁지 않은가? 3. 커넥터의 접속부는 확실한가? 4. 입출력모듈의 고정은 확실한가? 5. 전원선의 연결은 이상이 없는가? 6. 전선의 사이즈는 맞는가?
접지의 확인	1. 접지(제3종접지)는 확실히 전용으로 되었는가?
밧데리장착확인	1. CPU모듈내의 밧데리 홀더에 밧데리 장착은 되었는가? 2. 밧데리 커넥터가 CPU기판에 연결되었는가?
비상정지 회로의 확인	1. PLC의 외부에 이상시 동작한 비상정지 회로가 정확하게 구성되었는가? (PLC 오동작시 비상정지용)
전원투입	1. 전원공급원의 전원전압은 규격치 이내인가? AC110V사용시 (AC90~132V) AC220V사용시 (AC180~264V) 2. AC입력모듈 등의 입력전원을 외부에서 공급할때 전압은 규격치 이내인가?

5-2. 시운전의 순서

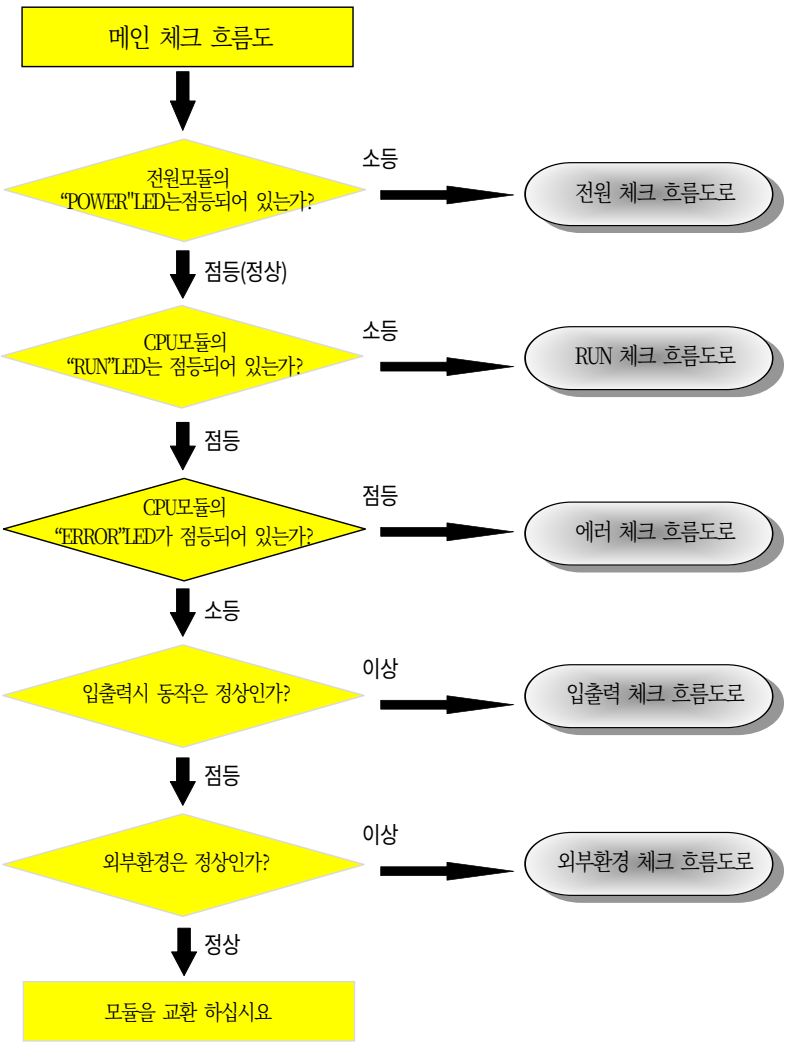
PLC의 취부 및 배선을 완료하고, 시운전에 앞선 확인을 끝낸후에는 아래의 순서대로 시운전하십시오.



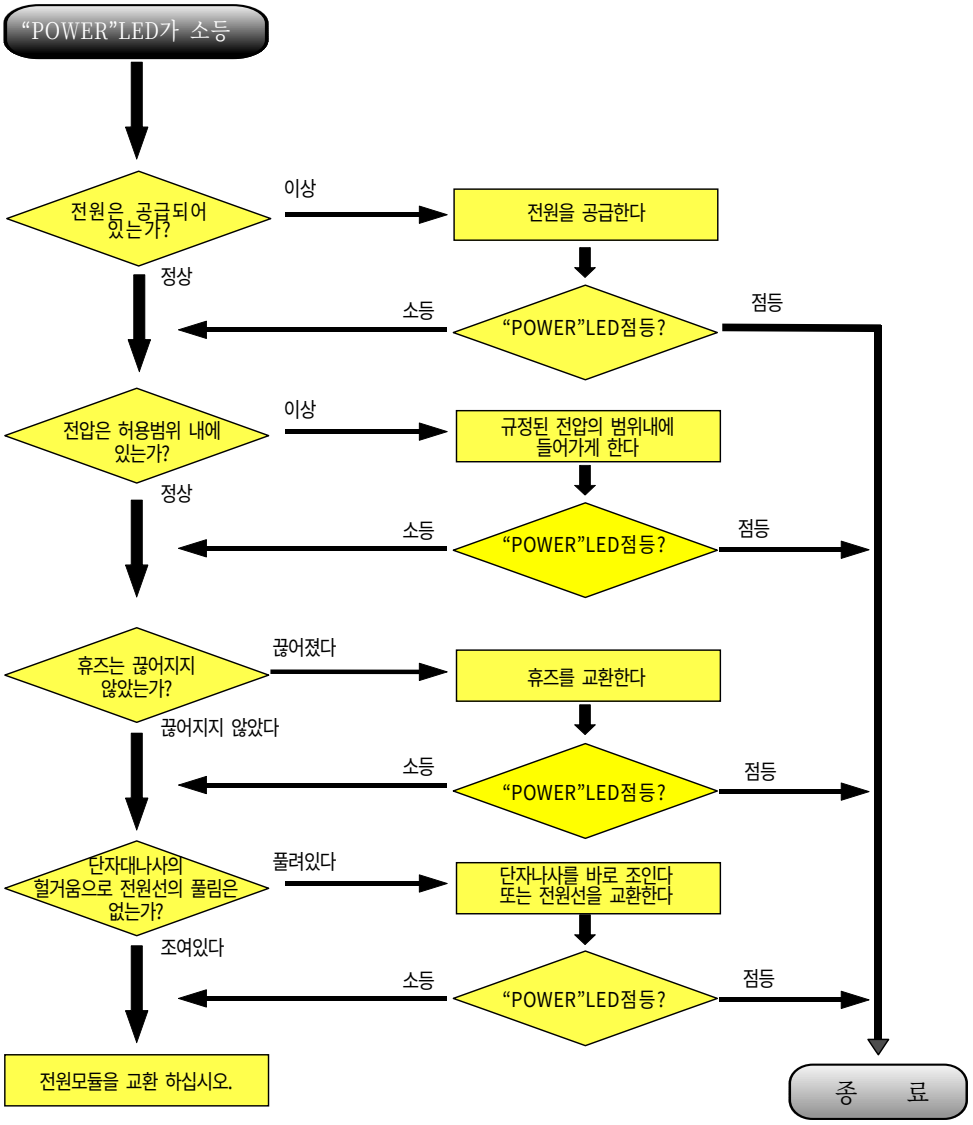
확 인 사 항	내 용
전원투입	1. 전원모듈의 입력전압을 확인합니다. 2. 입출력모듈용 전원전압을 확인합니다. 3. WinGPC또는 프로그램머(PGM-500)를 CPU 모듈에 접속합니다. (이때 CPU모듈은 PROG모드에 둡니다) 4. 전원을 투입합니다. 5. 전원 모듈의 LED의 점등을 확인합니다.
메모리 초기화	1. WinGPC 또는 프로그래머로 PLC를 초기화시킵니다.
입출력배선의 확인	1. 입력부의 배선 체크는 입력기기를 동작시킨뒤 입력모듈의 LED표시 또는 WinGPC나 프로그래머의 모니터 기능 등에서 확인합니다. 2. 출력부의 배선 체크는 WinGPC 또는 프로그래머를 이용하여 강제 출력기능으로 출력을 ON/OFF 시켜가며 확인합니다. (이때 CPU모듈은 RUN모드에 둡니다)
프로그래밍	1. WinGPC 또는 프로그래머를 이용하여 프로그램을 입력합니다. 2. WinGPC에 프로그램이 되어있으면 CPU모듈에 다운 로드합니다.
시운전	1. CPU모듈의 모드 스위치를 RUN위치로 올립니다. 2. RUN LED의 점등을 확인합니다. 3. 시퀀스 동작을 확인합니다.
프로그래밍 수정	1. 프로그램의 오류가 있으면 수정합니다.
프로그램의 보존	1. 프로그램을 플로피 디스크 또는 HDD 등에 저장합니다. 2. 프린터에 프로그램내용 (래더, 니모닉)을 프린트하여 보관합니다.
주) 기록한 프로그램에는 PLC기종명, 프로그램용량, 설치명칭, 날짜등을 기록해 두는것이 좋습니다.	

5-3. 시운전 이상시 조치 흐름도

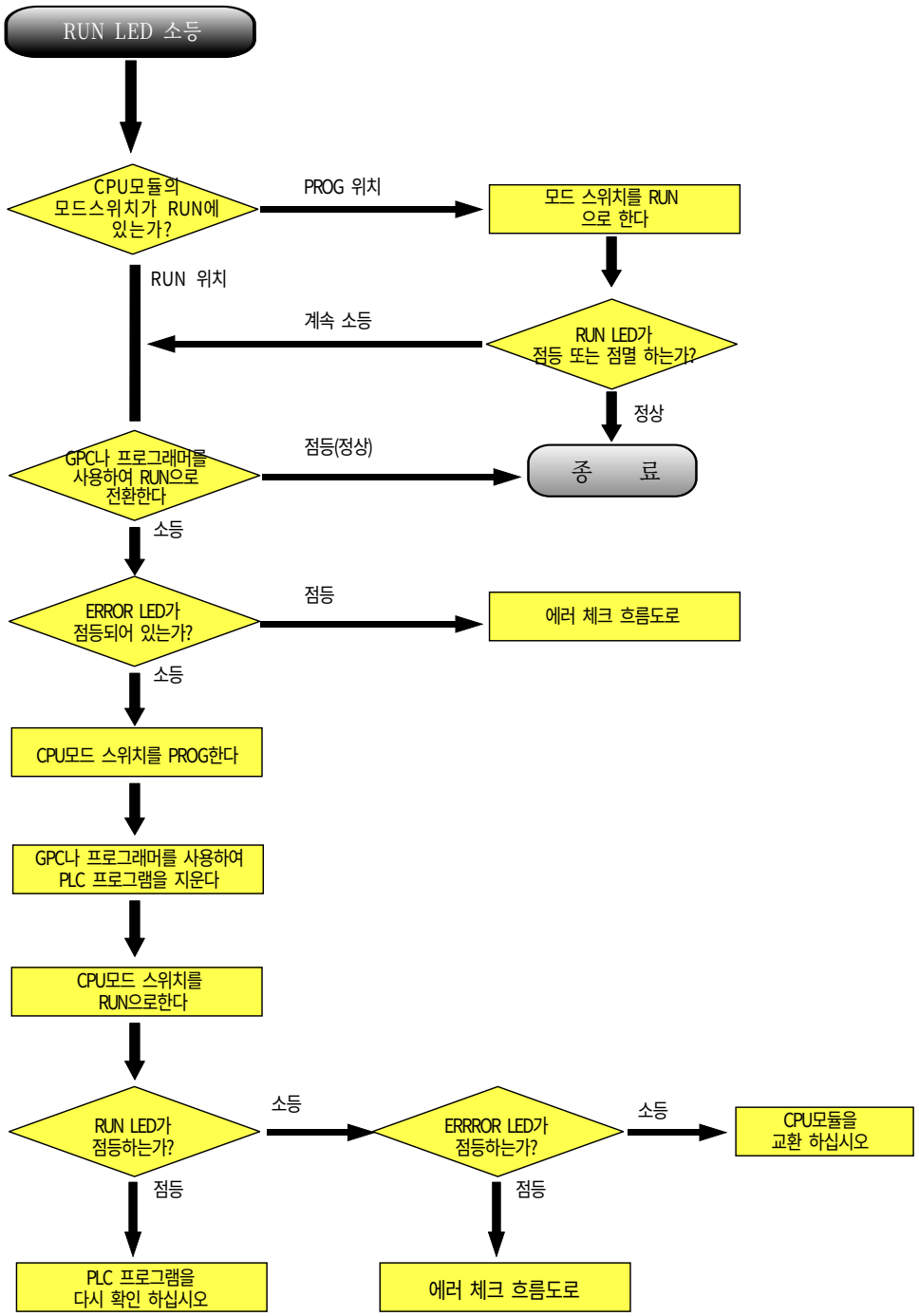
(1) 메인체크 흐름도 - 시운전 이상이 발생할 경우 먼저 상황을 충분히 파악하고 이상이 재현되는지의 여부, 타기기와의 관련성등을 정확하게 판단한 후 아래의 흐름도에 따라 체크 하십시오.



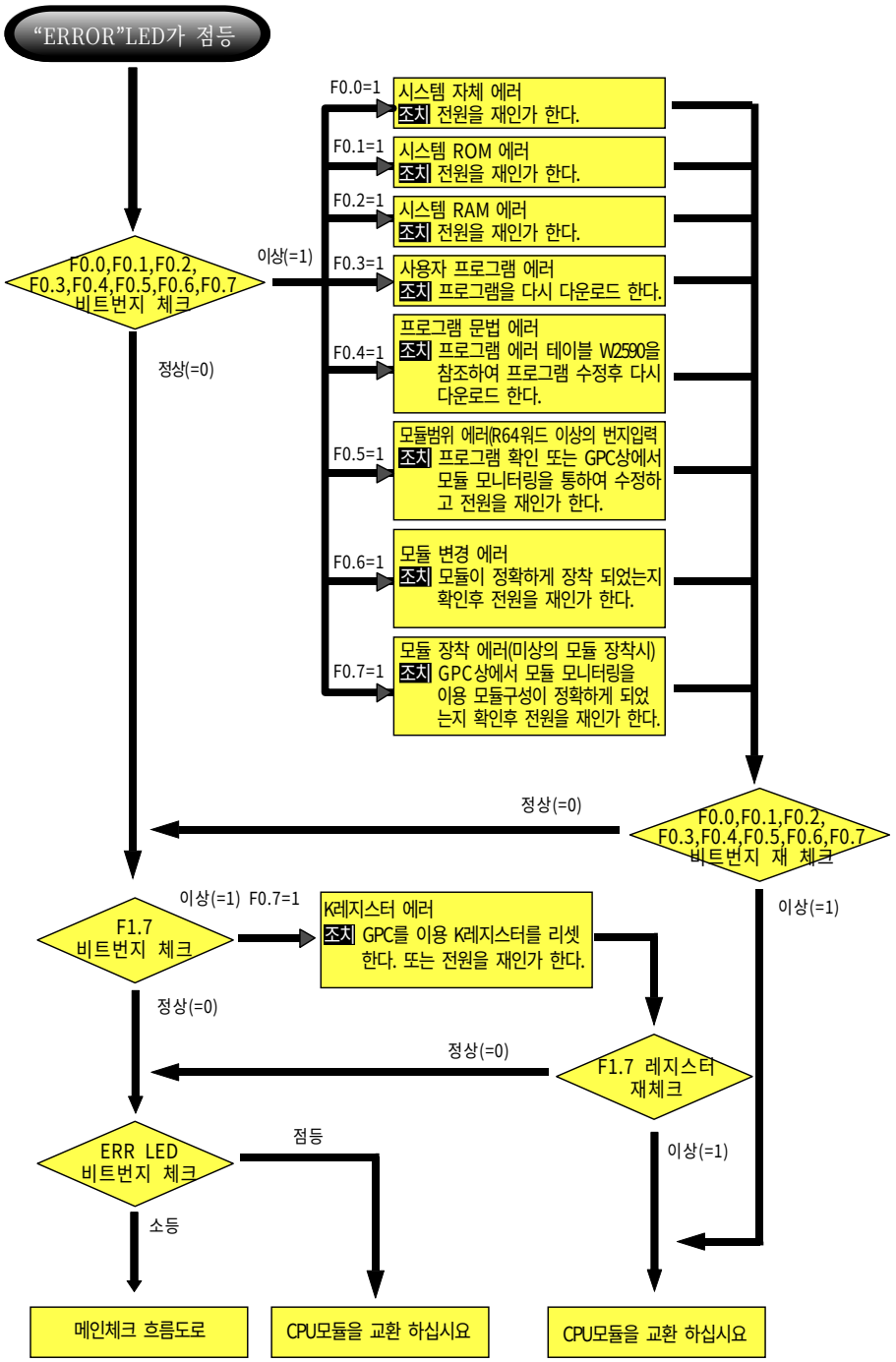
(2) 전원 체크 흐름도


"POWER"LED가 소등
전원은 공급되어 있는가?
이상
전원을 공급한다
정상
소등
"POWER"LED점등?
점등
전압은 허용범위 내에 있는가?
이상
규정된 전압의 범위내에 들어가게 한다
정상
소등
"POWER"LED점등?
점등
휴즈는 끊어지지 않았는가?
끊어졌다
휴즈를 교환한다
끊어지지 않았다
소등
"POWER"LED점등?
점등
단자대나사의 헐거움으로 전원선의 풀림은 없는가?
풀려있다
단자나사를 바로 조인다 또는 전원선을 교환한다
조여있다
소등
"POWER"LED점등?
점등
전원모듈을 교환 하십시오.
종 료

(3) RUN 체크 흐름도

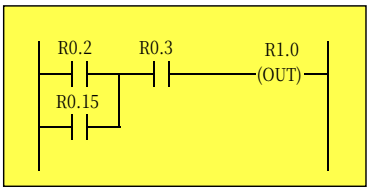

RUN LED 소등
CPU모듈의 모드스위치가 RUN에 있는가?
PROG 위치
모드 스위치를 RUN 으로 한다
RUN 위치
계속 소등
RUN LED가 점등 또는 점멸 하는가?
정상
GPC나 프로그래머를 사용하여 RUN으로 전환한다
점등(정상)
종 료
소등
ERROR LED가 점등되어 있는가?
점등
에러 체크 흐름도로
소등
CPU모드 스위치를 PROG한다
GPC나 프로그래머를 사용하여 PLC 프로그램을 지운다
CPU모드 스위치를 RUN으로한다
RUN LED가 점등하는가?
소등
ERRROR LED가 점등하는가?
소등
CPU모듈을 교환 하십시오
점등
점등
PLC 프로그램을 다시 확인 하십시오
에러 체크 흐름도로

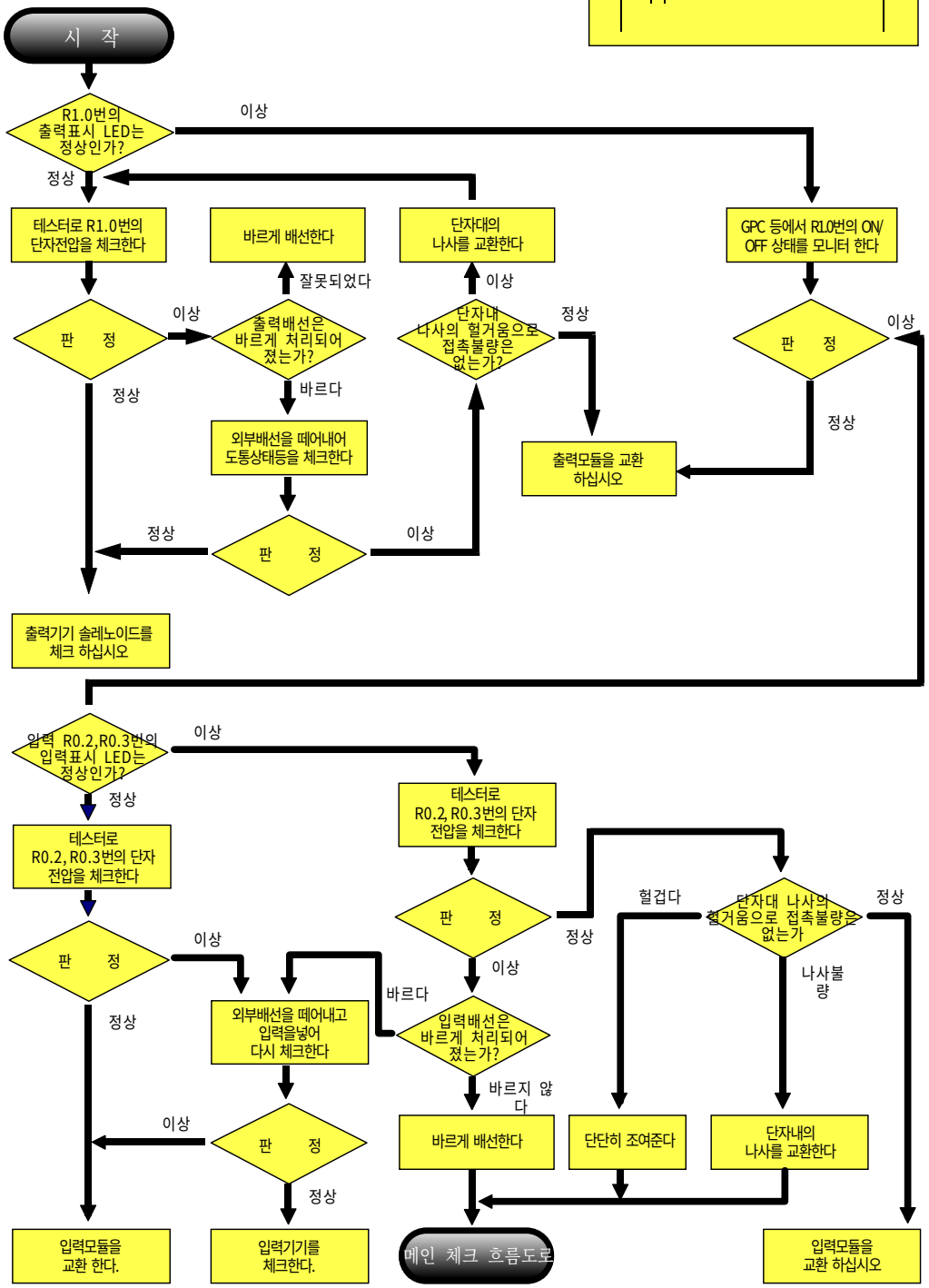
(4) 에러 체크 흐름도


"ERROR"LED가 점등
F0.0,F0.1,F0.2,
F0.3,F0.4,F0.5,F0.6,F0.7
비트번지 체크
이상(=1)
정상(=0)
F0.0=1
시스템 자체 에러
조치 전원을 재인가 한다.
F0.1=1
시스템 ROM 에러
조치 전원을 재인가 한다.
F0.2=1
시스템 RAM 에러
조치 전원을 재인가 한다.
F0.3=1
사용자 프로그램 에러
조치 프로그램을 다시 다운로드 한다.
F0.4=1
프로그램 문법 에러
조치 프로그램 에러 테이블 W2590을 참조하여 프로그램 수정후 다시 다운로드 한다.
F0.5=1
모듈범위 에러(R64워드 이상의 번지입력
조치 프로그램 확인 또는 GPC상에서 모듈 모니터링을 통하여 수정하고 전원을 재인가 한다.
F0.6=1
모듈 변경 에러
조치 모듈이 정확하게 장착 되었는지 확인후 전원을 재인가 한다.
F0.7=1
모듈 장착 에러(미상의 모듈 장착시)
조치 GPC상에서 모듈 모니터링을 이용 모듈구성이 정확하게 되었는지 확인후 전원을 재인가 한다.
F0.0,F0.1,F0.2,
F0.3,F0.4,F0.5,F0.6,F0.7
비트번지 재 체크
정상(=0)
이상(=1)
F1.7
비트번지 체크
이상(=1) F0.7=1
K레지스터 에러
조치 GPC를 이용 K레지스터를 리셋 한다. 또는 전원을 재인가 한다.
정상(=0)
F1.7 레지스터
재체크
정상(=0)
이상(=1)
ERR LED
비트번지 체크
점등
소등
메인체크 흐름도로
CPU모듈을 교환 하십시요
CPU모듈을 교환 하십시요

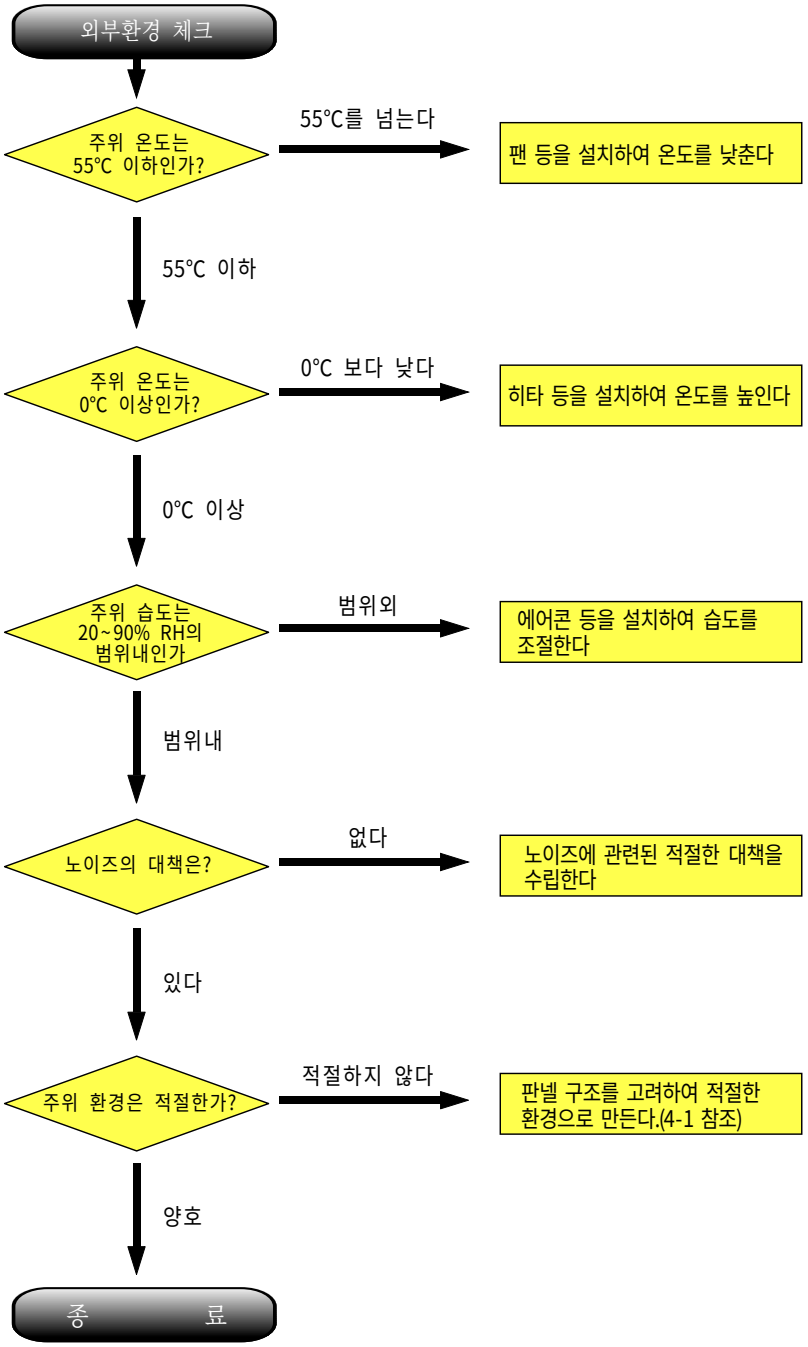
(5) 입출력 체크 흐름도

우측 회로를 사례로 체크한 경우입니다.





(6) 외부환경 체크 흐름도


외부환경 체크
주위 온도는 55℃ 이하인가?
55℃를 넘는다
팬 등을 설치하여 온도를 낮춘다
55℃ 이하
주위 온도는 0℃ 이상인가?
0℃ 보다 낮다
히타 등을 설치하여 온도를 높인다
0℃ 이상
주위 습도는 20~90% RH의 범위내인가
범위외
에어콘 등을 설치하여 습도를 조절한다
범위내
노이즈의 대책은?
없다
노이즈에 관련된 적절한 대책을 수립한다
있다
주위 환경은 적절한가?
적절하지 않다
판넬 구조를 고려하여 적절한 환경으로 만든다.(4-1 참조)
양호
종 료

5-4. 보수와 점검

(1) 보수와 점검

이 상 현 상	추 정 원 인	조 치
POWER LED가 점등되지 않는다.	휴즈가 끊어짐	휴즈교환
휴즈가 자꾸 끊어진다.	패턴의 단락 또는 부품이상	전원 또는 CPU 모듈 교환
RUN LED가 동작하지 않는다.	프로그램 이상	프로그램 수정
	전원라인 불량	CPU모듈 교환
RUN중 출력이 ON하지 않는다.	회로패턴 단락 또는 단선	CPU모듈 교환
특별한 접점번호이후가 동작하지 않는다.	I/O 버스 불량	마더보드 모듈 교환
입력 또는 출력모듈이 특정번호만 ON된다	I/O 버스 불량	마더보드 모듈 교환
출력모듈의 접점이 전부 동작하지 않는다	I/O 버스 불량	마더보드 모듈 교환

(2) 입력모듈

이 상 현 상	추 정 원 인	조 치
입력이 전부 ON되지 않는다. (LED 소등)	외부 입력전원이 공급되지 않음	전원공급
	외부 입력전압이 낮다	정격전원전압 입력
	단자대의 나사가 헐거움/ 접촉불량	연결부위의 확실한 결합
입력이 전부 ON 되지 않는다(LED점등)	입력회로 불량	입력모듈 교환
입력이 전부 OFF 되지 않는다.	입력회로 불량	입력모듈 교환
특정한 접점번호의 입력이 ON되지 않는다.	입력모듈에 연결된 기기불량	입력기기 교환
	입력배선의 단선	입력배선 연결
	외부입력의 ON 시간이 짧다	입력기기 조정
	단자대의 나사가 헐거움/ 접촉불량	연결부위의 확실한 결합
특정한 접점번호의 입력이 OFF되지 않는다	입력회로 불량	입력모듈 교환
입력이 불규칙하게 ON / OFF된다.	외부 입력전압이 낮다	정격 입력전압 공급
	노이즈에 따른 오동작	노이즈에 대한 대책수립
	단자대의 나사가 헐거움/ 접촉불량	연결부위의 확실한 결합
입력표시 LED가 점등하지 않는다. (동작은 정상)	LED 불량	입력모듈 교환

(3) 출력모듈

이 상 현 상	추 정 원 인	조 치
출력이 전부 ON되지 않는다	부하전원이 공급되지 않음	전원공급
	부하전원 전압이 낮다	정격전원전압 입력
	단자대의 나사가 헐거움/ 접촉불량	연결부위의 확실한 결합
	I/O커넥터 접촉불량	출력모듈 교환
	출력회로 불량	출력모듈 결합
출력이 전부 OFF 되지 않는다	출력회로 불량	출력모듈 교환
특정한 접점번호의 입력이 ON되지 않는다 (LED 소등)	출력 ON시간이 짧다	프로그램 수정
	출력회로 불량	출력모듈 교환
특정한 접점번호의 입력이 ON되지 않는다. (LED 점등)	출력에 연결된 부하의 이상	출력부하 교환
	출력배선의 단선	출력배선 연결
	단자대의 나사가 헐거움/ 접촉불량	연결부위의 확실한 결합
	출력접점 불량	출력모듈 또는 릴레이 교환
	출력회로 불량	출력모듈 교환
특정한 접점번호의 출력이 OFF되지 않는다. (LED 점등)	출력접점 불량	출력모듈 또는 릴레이 교환
	누설전류, 잔류전압에 따른 복귀불량	외부부하 교환
특정한 접점번호의 출력이 OFF되지 않는다. (LED 점등)	출력회로 불량	출력모듈 교환
출력이 불규칙하게 ON / OFF한다	부하전원 전압이 낮다	정격입력전압 공급
	노이즈에 의한 오동작	노이즈에 대한 대책수립
	단자대의 나사가 헐거움/ 접촉불량	연결부위의 확실한 결합
이상동작의 접점번호가 8단위로 동작한다	커먼단자의 나사가 헐겁다	연결부위의 확실한 결합
	단자대 커넥터의 접촉불량	연결부위의 확실한 결합
	CPU모듈 불량	CPU모듈 교환
출력표시 LED가 점등하지 않는다	LED 불량	출력모듈의 교환

(4) 이상시 추정원인과 조치

PLC는 평소 최상의 상태로 사용될 수 있도록 정기적인 점검이 필요합니다.
정기적인 점검은 6개월에 1회정도를 표준으로 합니다만 주위의 환경에 따라 점검기간을 당길 수 있습니다.

점검항목	점 검 내 용	판정기준	비고
공급전원	전원단자대에서 측정하여 전압변동은 기준내에 기준내에 있는가?	전원모듈 사양의 입력 전압규격 이내일것	테스터기
주위환경	주위온도(판넬내 온도)는 적당한가?	0~55°C	온도계
	주위습도(판넬내 습도)는 적당한가?	35~85%RH	습도계
	먼지등이 쌓이지 않았는가?	없을것	육안
입·출력용전원	입출력의 단자대에서 측정하여 전압변동은 기준내에 있는가?	각 입출력 사양의 규격치 이내일것	테스터기
모듈고정상태	각모듈은 확실히 고정되어 있는가?	확실히 고정되어 있을것	드라이버
	접속케이블의 커넥터는 완전하게 삽입 및 고정되어 있는가?		
	외부배선의 나사는 느슨하지 않은가?		
부품의 수명	접점릴레이	전기적수명 10~30만회	
	밧데리	3년 (25°C)	

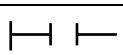
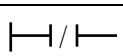
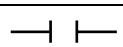
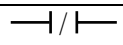
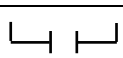
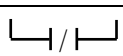
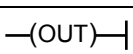
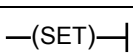
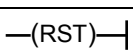
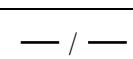
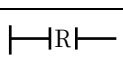
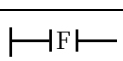
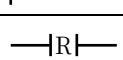
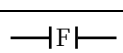
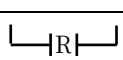
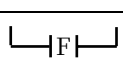
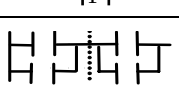
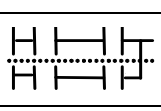
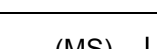
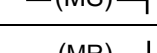
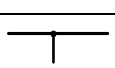
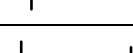
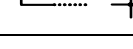
점검 주의사항

1. 모듈을 끼우고 뺄때는 전원을 OFF한뒤 실시합니다.
2. 불량을 교환할때는 다시한번 모듈의 이상여부를 확인해 주십시오.
3. 불량모듈의 수리를 위해 반품할 경우는 이상현상을 상세하게 기재하여 반품합니다.
4. 접촉불량등이 발생한 경우는 해당부위를 알콜 및 깨끗한 솜으로 닦아준뒤 다시 장착하여 점검해 주십시오.
5. 청소하는 경우 신나류를 사용하면 외관이 변색될 수 있으므로 사용하지 마십시오.

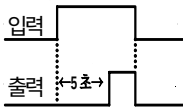
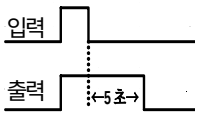
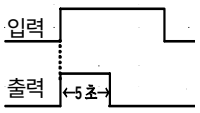
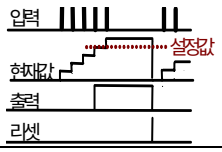
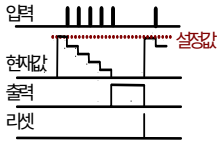
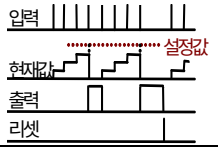
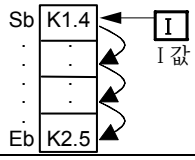
6 장

명령어 규격

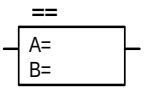
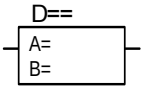
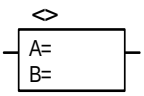
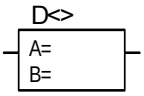
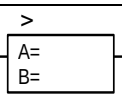
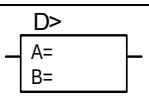
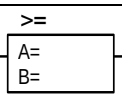
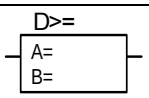
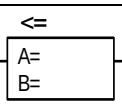
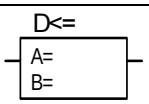
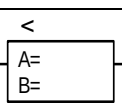
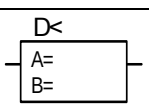
6-1. 기본 명령어

니모닉	명 칭	래더 심볼	설 명
STR	Start		a 접점 시작
STN	Start Not		b 접점 시작
AND	And		a 접점 직렬회로
ANN	And Not		b 접점 직렬회로
OR	Or		a 접점 병렬회로
ORN	Or Not		b 접점 병렬회로
OUT	Out		연산결과 출력
SET	Set		출력 SET(ON 유지)
RST	Reset		출력 RESET(OFF 유지)
NOT	Not		회로반전
STR DIF	Start Differential		상승엣지 접점 시작 ( )
STR DFN	Start Dif. Not		하강엣지 접점 시작 ( )
AND DIF	And Dif.		상승엣지 직렬 접속 ( )
AND DFN	And Dif. Not		하강엣지 직렬 접속 ( )
OR DIF	Or Dif		상승엣지 병렬 접속 ( )
OR DFN	Or Dif. Not		하강엣지 병렬 접속 ( )
ANB	And Block		회로블럭의 직렬 접속
ORB	Or Block		회로블럭의 병렬 접속
MS	Master block Set		마스터 블럭 시작 (CPU ROM V2.0 이상)
MR	Master block Reset		마스터 블럭 종료 (CPU ROM V2.0 이상)
MCS	MCS		회로분기 시작
MCR	MCR		분기 종료
-	확장 (Entension)		확장 (AND 조건으로 확장할 때 쌍으로 사용) (WinGPC3, GPC5 등 전용)

6-2. 타이머/카운터/SR 명령어

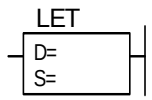
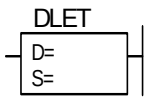
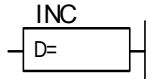
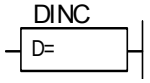
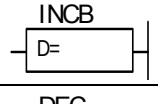
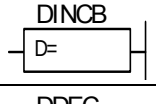
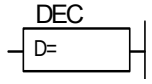
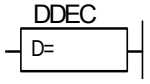
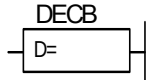
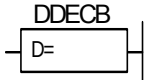
니모닉	명 칭	래더 심볼	설 명	비 고
TIM	On Delay Timer	TIM Ch=00010 SV=00050	**한시동작 순시정지 타이머** 	Time Base: Ch 0~63= 0.01s Ch 64~255= 0.1s 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
TOF	Off Delay Timer	TOF Ch=00064 SV=00005	**순시동작 한시정지 타이머** 	Time Base: Ch 0~63= 0.01s Ch 64~255= 0.1s 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
SST	Single Shot Timer	SST Ch=00100 SV=00005	**순시동작/정지 타이머** 	Time Base: Ch 0~63= 0.01s Ch 64~255= 0.1s 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
UC	Up Counter	UC U Ch=020 SV=004 R	**상승 카운터** 	채널범위 : Ch 0~255 (타이머와 공용) 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
DC	Down Counter	DC D Ch=021 SV=005 R	**하강 카운터** 	채널범위 : Ch 0~255 (타이머와 공용) 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
RCT	Ring Counter	RCT T Ch=022 SV=004 R	**링(회전) 카운터** 	채널범위 : Ch 0~255 (타이머와 공용) 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
UDC	Up-Down Counter	UDC U Ch=023 SV=003 D R	**상승/하강 카운터** 	채널범위 : Ch 0~255 (타이머와 공용) 설정범위 : SV = 0~65535 접점표시 : TC + 채널번호
SR	Shift Register	SR I Sb=K1.4 Eb=K2.5 P R	**시프트 레지스터** 	번지 사용영역 (Sb, Eb): M, K 영역 (동일한 종류(M,K 등)범위내) P 단자 입력시마다 1Bit 씩 시프트 P 단자입력시 I 상태값이 Sb 에 저장 최대 사용갯수 : 256 개 명령

6-3. 비교 명령어

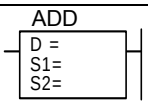
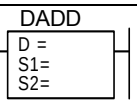
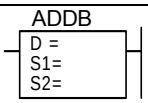
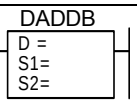
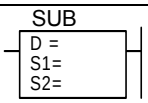
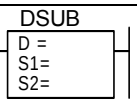
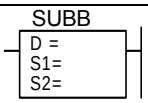
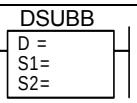
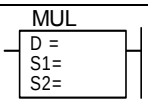
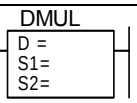
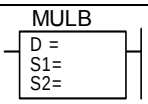
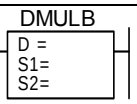
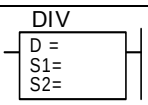
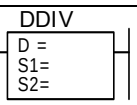
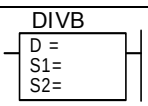
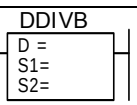
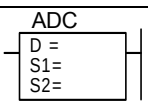
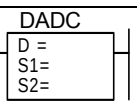
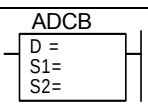
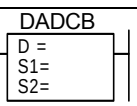
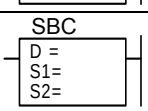
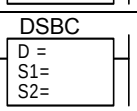
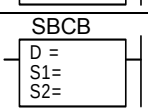
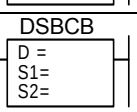
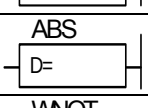
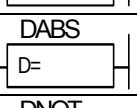
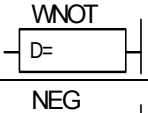
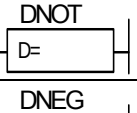
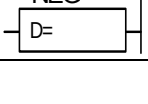
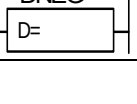
니모닉	명 칭	워드 심볼	더블워드 심볼	설 명
STR == AND == OR ==	START == AND == OR ==			A 값과 B 값이 같을때 ON. A 및 B 는 워드/더블워드 또는 Data 값
STR <> AND <> OR <>	START <> AND <> OR <>			A 값과 B 값이 다를때 ON. <>은 ≠ 과 동일한 의미, A 및 B 는 워드/더블워드 또는 Data 값
STR > AND > OR >	START > AND > OR >			A 값이 B 값보다 클때 0N.
STR >= AND >= OR >=	START >= AND >= OR >=			A 값이 B 값보다 크거나 같을때 ON.
STR <= AND <= OR <=	START <= AND <= OR <=			A 값이 B 값보다 작거나 같을때 ON.
STR < AND < OR <	START < AND < OR <			A 값이 B 값보다 작을때 ON.

주) 니모닉 프로그램에서의 더블워드 비교명령은 부호 앞에 ‘D’를 삽입해야 합니다.

6-4. 대입. 증가/감소 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
LET (DLET)	Let (대입)			S 의 값을 D 에 대입(저장)
INC (DINC)	Increment (Binary 증가)			입력이 ON 될때마다 D 의 값을 1 씩 증가
INCB (DINCB)	BCD increment (BCD 증가)			입력이 ON 될때마마 BCD 단위로 D 의 값을 1 씩 증가
DEC (DDEC)	Decrement (Binary 감소)			입력이 ON 될때마다 D 의 값을 1 씩 감소
DECB (DDECB)	BCD decrement (BCD 감소)			입력이 ON 될때마다 BCD 단위로 D 의 값을 1 씩 감소

6-5. 산술연산 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
ADD (DADD)	Decimal addition (10 진수 덧셈)			D = S1 + S2 (10 진수 연산)
ADDB (DADDB)	BCD addition (BCD 덧셈)			D = S1 + S2 (BCD 연산)
SUB (DSUB)	Decimal subtraction (10 진수 뺄셈)			D = S1 - S2 (10 진수 연산)
SUBB (DSUBB)	BCD subtraction (BCD 뺄셈)			D = S1 - S2 (BCD 연산)
MUL (DMUL)	Decimal multiplication (10 진수 곱셈)			D = S1 × S2 (10 진수 연산)
MULB (DMULB)	BCD multiplication (BCD 곱셈)			D = S1 × S2 (BCD 연산)
DIV (DDIV)	Decimal division (10 진수 나눗셈)			D = S1÷S2 (10 진수 연산), S2 값이 0 일 때 에러로 처리함.
DIVB (DDIVB)	BCD division (BCD 나눗셈)			D = S1÷S2 (BCD 연산) S2 값이 0 일 때 에러로 처리함.
ADC (DADC)	Decimal addition w/carry (캐리포함 10 진수 덧셈)			D = S1 + S2 + CY (10 진수 덧셈 연산, 캐리 포함)
ADCB (DADCB)	BCD addition w/carry (캐리포함 BCD 덧셈)			D = S1 + S2 + CY (BCD 덧셈 연산, 캐리 포함)
SBC (DSBC)	Decimal subtraction w/carry (캐리포함 10 진수 뺄셈)			D = S1 - S2 – CY (10 진수 뺄셈연산, 캐리 포함)
SBCB (DSBCB)	BCD subtraction w/carry (캐리포함 BCD 뺄셈)			D = S1 - S2 - CY (BCD 뺄셈연산, 캐리 포함)
ABS (DABS)	Absolute value (절대값 연산)			D = |D| (절대값 연산)
WNOT (DNOT)	NOT (반전, 1 의보수)			D 에 1 의 보수값(반전)을 D 에 저장
NEG (DNEG)	Negative (2 의 보수)			D 에 2 의 보수값을 D 에 저장 (1 의 보수+1)(-처리값)

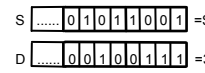
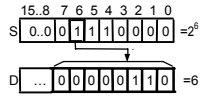
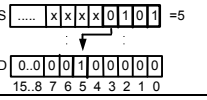
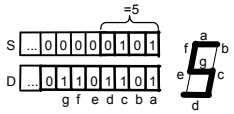
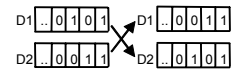
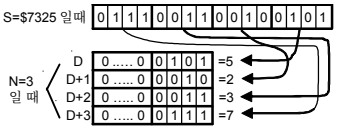
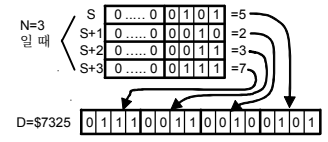
6-6. 논리 연산 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
WAND (DAND)	AND (논리곱)	WAND D= S1= S2=	DAND D= S1= S2=	S1 과 S2 의 논리곱을 D 에 저장 S1 0 0 1 1 S2 0 1 0 1 D 0 0 0 1
WOR (DOR)	OR (논리 합)	WOR D= S1= S2=	DOR D= S1= S2=	S1 과 S2 의 논리합을 D 에 저장 S1 0 0 1 1 S2 0 1 0 1 D 0 1 1 1
WXOR (DXOR)	Exclusive OR (배타적 논리합)	WXOR D= S1= S2=	DXOR D= S1= S2=	S1 과 S2 의 배타적 논리합을 D 에 저장 S1 0 0 1 1 S2 0 1 0 1 D 0 1 1 0
WXNR (DXNR)	Exclusive OR NOT (동등 회로)	WXNR D= S1= S2=	DXNR D= S1= S2=	S1 과 S2 의 배타적 논리합의 반전을 D 에 저장(동등할 경우 1(ON)처리) S1 0 0 1 1 S2 0 1 0 1 D 1 0 0 1

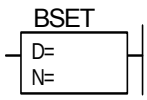
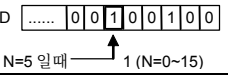
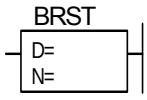
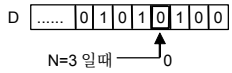
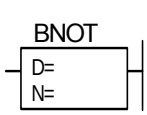
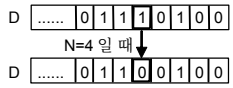
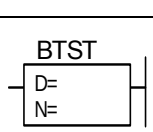
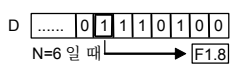
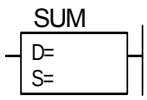
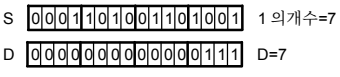
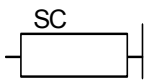
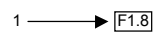
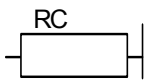
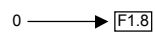
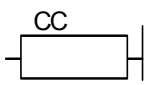
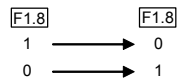
6-7. 회전 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
RLC (DRLC)	Rotate left (캐리와 무관, 좌로 회전)	RLC D= N=	DRLC D= N=	N 비트씩 좌로(하위-〉 상위로)회전 F1.8 ← 15 ...D 워드 .. 00 ←
RRC (DRRC)	Rotate right (캐리와 무관, 우로 회전)	RRC D= N=	DRRC D= N=	N 비트씩 우로(상위-〉 하위로)회전 → 15 ...D 워드 .. 00 → F1.8
ROL (DROL)	Rotate left (캐리포함, 좌로 회전)	ROL D= N=	DROL D= N=	N 비트씩 좌로(하위-> 상위로)회전(이동) (하위비트는 F1.8 값이 입력) 15 ...D 워드 .. 00 ← F1.8
ROR (DROR)	Rotate right (캐리포함, 우로 회전)	ROR D= N=	DROR D= N=	N 비트씩 우로(상위-〉 하위로)회전(이동) (상위비트는 F1.8 값이 입력) → 15 ...D 워드 .. 00 → F1.8
SHL (DSHL)	Shift left (좌로 시프트)	SHL D= N=	DSHL D= N=	N 비트씩 좌로이동(하위비트는 0 값이 입력) F1.8 ← 15 ...D 워드 .. 00 ← 0
SHR (DSHR)	Shift right (우로 시프트)	SHR D= N=	DSHR D= N=	N 비트씩 우로이동 (상위비트는 0 으로 입력) 0 → 15 ...D 워드 .. 00 → F1.8

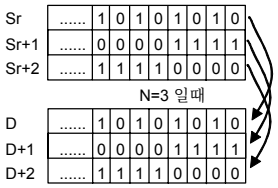
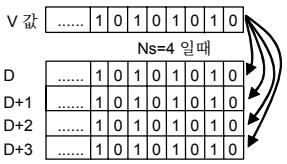
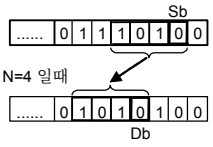
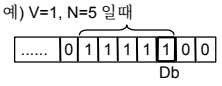
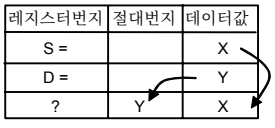
6-8. 워드변환 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
BCD (DBCD)	BCD 변환	BCD D= S=	DBCD D= S=	S 의 2 진수 값을 BCD 로 변환하여 D 에 저장 
BIN (DBIN)	Binary 변환	BIN D= S=	DBIN D= S=	S 의 BCD 값을 2 진수로 변환하여 D 에 저장. 
ENCO	엔코더 변환 Encode	ENCO D= S=	———	1 이 있는 최상위 비트의 비트번지 값을 (2^n 의 n 값) D 의 하위 8bit 에 저장  주의) SPC 시리즈(2^n +1)와 기능이 다름
DECO	디코더 변환 Decode	DECO D= S=	———	S 의 하위 4 비트 값을 2^n 승 값으로 하여 D 에 저장 
SEG	7-세그먼트 7-Segment	SEG D= S=	———	S 의 하위 4 비트 값을 7-Segment 로 변환하여 D 에 저장 
XCHG (DXCHG)	교환 (Exchange)	XCHG D1= D2=	DXCHG D1= D2=	D1 과 D2 의 값을 서로 교환 
DIS	분리 Dissemble	DIS D= N= S=	———	S 를 4 비트씩 N+1 개로 분리하여 D 부터 D+N 워드의 하위 4 비트에 저장 
UNI	결합 Unify	UNI D= S= N=	———	S 부터 S+N 워드의 하위 4 비트를 D 에 하위비트부터 저장(N= 0~3) 

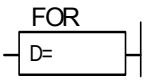
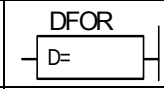
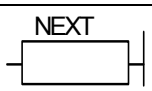
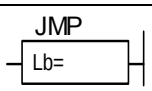
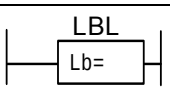
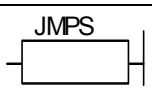
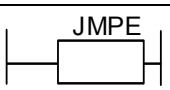
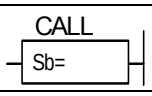
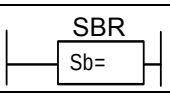
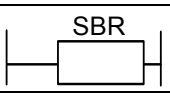
6-9. 비트 변환 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
BSET	비트 셋(SET) Bit Set		——	D 워드의 N 번째 비트 값을 1 로 SET 시킴 
BRST	비트 리셋 Bit Reset		——	D 워드의 N 번째 비트 값을 0 으로 저장 
BNOT	비트 반전 Bit Not		——	D 워드의 N 번째 비트 값을 반전시킴. 
BTST	비트 테스트 Bit Test		——	D 워드의 N 번째 비트 값을 F1.8 에 저장 
SUM	1 인 비트의 수 Sum		——	S 워드에서 1 인 비트의 개수를 D 에 저장 
SC	캐리 셋(SET) Set Carry		——	캐리 비트(F1.8)를 1 로 SET 함 
RC	캐리 리셋 Reset Carry		——	캐리 비트(F1.8)를 0 으로 RESET 함 
CC	캐리 반전 Complement Carry		——	캐리 비트(F1.8)값을 반전 시킴. 

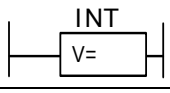
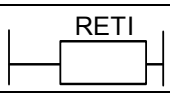
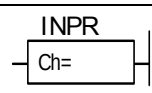
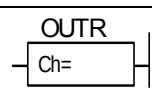
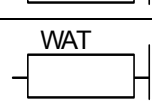
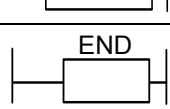
6-10. 전송 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
MOV	워드 전송 Move	MOV D= S= N=	———	S 워드부터 N 개 워드(S 포함)를 D 워드부터 N 개로 값을 복사함 
FMOV	워드 다중 전송 Fill Move	FMOV D= N= V=	———	V 값을 D 부터 N 개 워드로 복사함 
BMOV	비트 전송 Bit Move	BMOV Db= Sb= N =	———	Sb 비트로부터 N 개 비트를 Db 비트부터 N 개로 값을 복사함. (Sb, Db 는 비트번지만 지정가능) 
BFMV	비트 다중 전송 Bit Fill Move	BFMV Db= N = V =	———	V 값을 Db 부터 N 개 비트로 복사함. (V=0 또는 1)(N=1…256, 영역내) (Db 는 비트번지) 
LDR (DLDR)	간접지정 번지값의 데이터 불러오기 Load D←(S)	LDR D= S=	DLDR D= S=	S 워드의 값을 절대번지로 하는 레지스터의 값을 D 에 저장 (절대번지는 매뉴얼 참조) 
STO (DSTO)	간접지정 번지값에 S 데이터 저장 Store (D)←S	STO S= D=	DSTO S= D=	S 워드값을 D 워드값을 절대번지로 하는 레지스터에 저장 (절대번지는 매뉴얼 참조) 

6-11. 블록 처리 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
FOR (DFOR)	FOR 루프 제어 For Loop			D 값을 1 씩 감소시키면서 NEXT 명령 까지 D 회 (D 가 0 일때까지) 반복수행
NEXT	FOR 루프 종료 Next		——	FOR 명령어부터 반복 수행
JMP	LBL 명령의 Lb 번 까지 점프		——	LBL 명령의 L 번까지의 점프 L=0~63(중복 사용 불가능)
LBL	JMP 점프 종료		——	JMP 로 이동되는 위치 L=0~63(중복 사용 불가능)
JMPS	JMPE 까지 점프 Jump Start		——	JMPE 명령어 까지 점프
JMPE	JMPS 점프 종료 Jump End		——	점프 끝 (JMPS 와 쌍으로 구성) (JMPS 명령 이후에 있어야 함)
CALL	서브루틴 호출 Call Subroutine		——	서브루틴 호출 명령
SBR	서브루틴 시작 Subroutine Start		——	서브루틴 시작 명령 Sb=0~63(64 개 사용, 중복사용불가능)
RET	서브루틴 완료 Subroutine Return		——	서브루틴 끝

6-12. 시스템 제어 명령어

니모닉	명 칭	워드 심볼	더블워드	설 명
INT	정주기 스캔 시작 Interrupt		——	정주기 스캔 시작명령(1 개만사용가능) N:=1~999(20msec 10sec) 정주기시간=(Ni+1)×0.01 초
RETI	정주기 스캔 끝 Return Interrupt		——	정주기 스캔 끝
INPR	입력신호 받기 Input Refresh		——	외부입력 리프레쉬(프로그램 수행중 입력신호 받아들임) CH 는 외부 입력 워드번지
OUTR	출력신호 보내기 Output Refresh		——	외부출력 리프레쉬(프로그램 수행중 출력신호 보냄) CH 는 외부출력 워드번지
WAT	워치독경과치 리셋 Watchdog Timer		——	워치독 타이머 경과치 클리어
END	프로그램 끝 END		——	프로그램 수행종료(처음부터 수행) 마지막부분은 CPU 에서 자동생성 됩니다.

6-13. 통신제어 명령어

니모닉	명 칭	워드 심볼	더블 워드	설 명
READ	데이터 읽기 (고기능 UNIT 의 공유메모리영역)	READ TO=RR1 SZ=NR3 FR=NN5:NR6	——	지정 Slot(NN5)의 내부위치(NN6)부터 지정워드수(NR3)를 읽어내어 레지스터(RR1)에 차례로 기입.
WRITE	데이터 쓰기 (고기능 UNIT 의 공유메모리영역)	WRITE TO=NN1:NR2 SZ=NR3 FR=NN5	——	숫자/레지스터(NR5)부터 지정워드수(NR3)를 읽어내어 지정 Slot(NN1)의 내부위치(NR2)에 차례로 기입.
RMRD	리모트 슬레이브 데이터 읽기 (고기능 UNIT 의 공유메모리영역)	RMRD TO=NR1:RR2 NT=NN3:NN4 FR=NN5:NR6	——	지정 Remote Network Loop(NN3)상의 Station(NN4)에 대한 Slot(NN5)의 내부위치(NR6)부터 지정워드수(NR1)를 읽어내어 레지스터(RR2)에 차례로 기입.
RMWR	리모트 슬레이브 데이터 쓰기 (고기능 UNIT 의 공유메모리영역)	RMWR NT=NN1:NN2 TO=NN3:NR4 FR=NR5:NR6	——	숫자/레지스터(NR6)부터 지정워드수(NR5)를 읽어내어 지정 Remote Network Loop(NN1)상의 Station(NN2)에 대한 Slot(NN3)의 내부위치(NR4)에 차례로 기입.
RECV	링크 데이터 워드수신	RECV TO=NR1:RR2 NT=NN3:NN4 FR=NN5:NR6	——	지정 Link Network Loop(NN3)상의 Station(NN4)에 대한 레지스터종류(NN5)의 지정위치(NR6)부터 지정워드수(NR1)를 읽어내어 레지스터(RR2)에 차례로 기입.
SEND	링크 데이터 워드송신	SEND NT=NN1:NN2 TO=NN3:NR4 FR=NN5:NR6	——	숫자/레지스터(NR6)부터 지정워드수(NR5)를 읽어내어 지정 Link Network Loop(NN1)상의 Station(NN2)에 대한 레지스터종류(NN3)의 지정위치(NR4)에 차례로 기입.
RECVB	링크 데이터 비트수신	RECVB TO=BR1 NT=NN3:NN4 FR=NN5:NR6	——	지정 Link Network Loop(NN3)상의 Station(NN4)에 대한 레지스터종류(NN5)의 지정 Bit 위치(NR6)의 값을 읽어내어 Bit 레지스터(BR1)에 기입.
SENDB	링크 데이터 비트송신	SENDB NT=NN1:NN2 TO=NN3:NR4 FR=NB5	——	Bit 레지스터(NB5)의 값을 지정 Link Network Loop(NN1)상의 Station(NN2)에 대한 레지스터종류(NN3)의 지정 Bit 위치(NR4)에 기입.

6-14. 명령어 상세설명

- 기본 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
STR	┣┫	a 접점 시작	1	전기종
STN	┣┫	b 접점 시작	1	전기종
AND	─┫	a 접점 직렬접속	1	전기종
ANN (주 1)	─┫	b 접점 직렬접속	1	전기종
OR	┗┫	a 접점 병렬접속	1	전기종
ORN	┗┫	b 접점 병렬접속	1	전기종

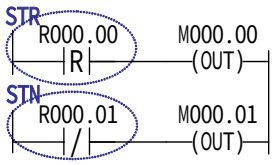
주 1) ANN 명령은 핸디로더의 종류에 따라 ADN 으로 표기된 기종(구형)도 있습니다.

도움말

1. ‘a 접점’은 지정한 번지의 값이 ‘1’일 때 ON 이 되며, 신호가 없는 Normal 상태인 ‘0’일 때는 Off 되는 NO(Normal Open) 접점 상태를 말한다.
2. ‘b 접점’은 지정한 번지의 값이 Normal 상태인 ‘0’일 때 회로가 도통되는 ON 상태가 되며, ‘1’일 때는 Off 가 된다. 즉, 신호가 없는 ‘0’상태 에서 연결이 되는 NC(Normal Close) 접점 상태를 말한다.
3. 접점의 상태는 ‘a 접점’과 ‘b 접점’으로 나누며, 니모닉 명령어로는 직렬접속(AND 조건)과 병렬접속(OR 조건)으로 나누어 AND/ANN, OR/ORN 등으로 구별하여 표현한다.
4. 번지 지정은 비트번지 영역만 사용 가능하고, R,L,M,K,F,TC 등을 사용할 수 있다.

사용예제 1

STR, STN 명령



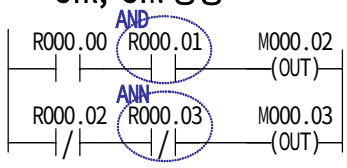
도움말

R000.00 의 a 접점이 ON 되면 출력 M000.00 이 ON 되고, 접점이 OFF 되면 출력도 OFF 가 됩니다.
b 접점의 경우 R000.01 이 OFF 상태에서 출력이 ON 되며, 접점이 ON 되면 출력이 OFF 된다.

니모닉 프로그램 예

래더 프로그램을 니모닉으로 변환하면 STR, OUT, STN, OUT 의 순서로 된다.
즉, 니모닉 프로그램으로 확인해 보면
STR R000.00, OUT M000.00
STN R000.01, OUT M000.01 이된다

STR, STN 명령



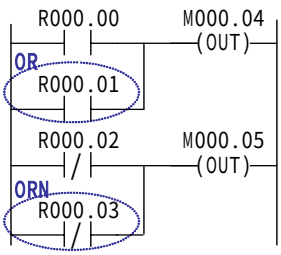
도움말

a 접점의 R0.0 이 ON 상태에서 R0.1 이 ON 되어야(AND 조건) 출력이 ON 되며, b 접점의 R000.02 는 OFF 상태이고, R000.03 도 OFF 상태(AND NOT 조건)에서 출력이 ON 된다.

니모닉 프로그램 예

니모닉 프로그램으로 변환하면
STR R000.00, AND R000.01, OUT M000.02
STN R000.02, ANN R000.03, OUT M000.03
의 순서로 된다.

OR, ORN 명령



도움말

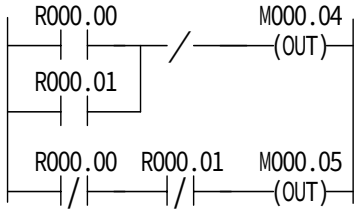
‘a 접점’의 R0.0 또는 R0.1 에서 하나만 ON 되어도(OR 조건) 출력(M0.4) 이 ON 되는 상태를 말하며, 부울대수로 OR 회로로 출력 Y=A+B 로 표현한다.
‘b 접점’의 R0.2 와 R0.3 중에 하나만 OFF 가 되어도 출력 M0.5 가 ON 되는 회로를 ‘NAND 회로’(AND NOT)라고 하며, 출력(Y)=$\overline{A*B}$ 로 표현하고 이것을 변환하면 Y=$\overline{A}+\overline{B}$ 이며 래더와 같다

니모닉 프로그램 예

니모닉 프로그램으로 확인 해 보면 구성하고 있는 순서에 따라 작동된다.
즉, STR R000.00,
OR R000.01,
OUT M000.04
STN R000.02,
ORN R000.03,
OUT M000.05 의 순서로 된다.

응용예제 1

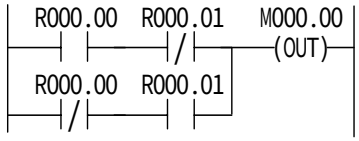
NOR(OR NOT) 회로



도움말

NOR(OR NOT)회로는 $Y=\overline{A+B}$ 를 말하며, 이것은 $Y=\overline{A}*\overline{B}$ 와 동일하고 왼쪽의 래더와 같으며, 출력 M0.4 와 M0.5 는 동일한 프로그램이 된다.

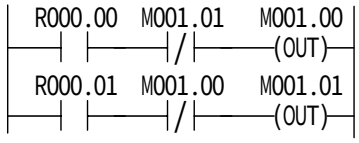
XOR(Exclusive OR) 회로



도움말

XOR(배타적 논리합)회로는 입력 A,B 중 하나만 ON 일때 출력이 ON 되는 회로를 말하며, $Y=A*\overline{B}+\overline{A}*B$ 로 표현한다

인터록 회로

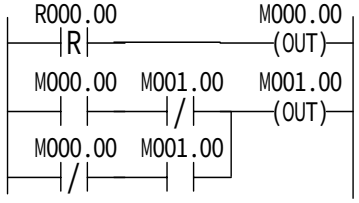


도움말

인터록 회로는 출력 X 및 출력 Y 중에서 하나가 ON 작동을 하고 있으면 다른쪽 출력이 ON 되지 못하도록 제한하는 회로를 말하며, $X=A*\overline{Y}$, $Y=B*\overline{X}$ 로 표현한다.

응용예제 2

교번동작 회로



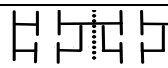
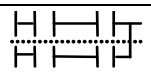
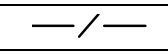
도움말

교번동작 회로는 입력접점이 한번씩 ON,OFF 될때마다 출력이 교번으로 한번은 ON, 한번은 OFF 를 반복하는 회로를 말한다. 즉, 입력이 ON 된 후 OFF 가 되어도 출력은 계속 ON 을 유지하고 있으며, 다시 입력이 ON, OFF 되면 출력이 OFF 상태로 유지된다.

주 1) 교번동작 회로의 예제에서 입력은 R0.0, 출력은 M1.0, 내부접점으로 M0.0 으로 작동되도록 프로그램 되어 있습니다.

주 2) 입력접점의 래더표시 ⊣R⊢ 은 ‘상승엣지’를 뜻하며, 입력이 OFF 에서 ON 으로 변하는 순간의 1Scan 동안만 작동하는 접점입니다. 상세내용은 DIF 명령어 부문을 참조바랍니다.

- 기본 명령어

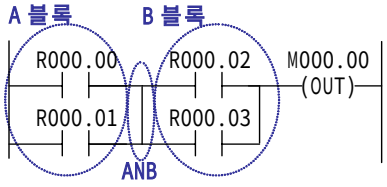
니모닉 명령	래더 표시	설 명	워드수	적용기종
ANB		회로블럭의 직렬 접속	1	전기종
ORB		회로블럭의 병렬 접속	1	전기종
NOT		회로 반전 (이전의 연산결과를 반전시킴)	1	전기종

도움말

1. ANB(AND Block)은 블록으로 구성된 것을 직렬로 접속하는 것을 말하며, ORB(OR Block)은 회로 블록을 병렬로 접속하는 경우를 말한다.
2. NOT 명령은 NOT 이전의 상태를 반전시키는 역할을 하며, 번지는 지정하지 않는다.
3. 이 3 개의 명령어는 번지 지정은 하지 않으며, 명령어만 사용한다.
4. 블록으로 구성하는 버퍼(Stack)는 최대 16 개까지 가능하며, CPU 기종별로 다릅니다.

사용예제

ANB 명령



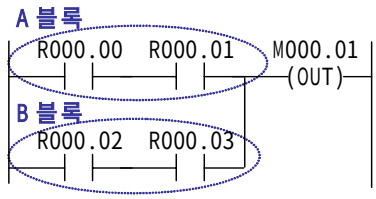
도움말

A 블록과 B 블록을 직렬 AND 조건으로 연결하는 것이 ANB 이며, A 블록과 B 블록이 모두 ON 되면 출력이 ON 된다

니모닉 프로그램 예

니모닉으로 확인해 보면
STR R000.00, OR R000.01,
STR R000.02, OR R000.03 으로 되고,
다음으로 ANB, OUT M000.00 이된다

ORB 명령



도움말

A 블록과 B 블록을 OR 조건의 병렬로 연결하는 것이 ORB 이며, 블록중에서 하나가 연결되면 출력이 ON 된다.

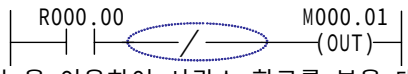
니모닉 프로그램 예

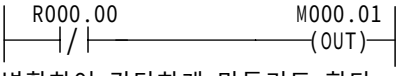
니모닉 프로그램으로 변환하면
STR R000.00, AND R000.01,
STR R000.02, AND R000.03,
ORB, OUT M000.01 의 순서로 된다.

NOT 명령

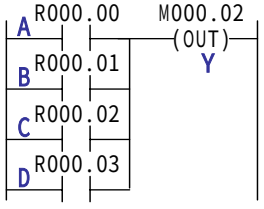
NOT 은 이전의 연산결과를 반전시키는 기능이며, 접점상태의 반전과 직렬회로 또는 병렬회로의 반전도 가능하다.

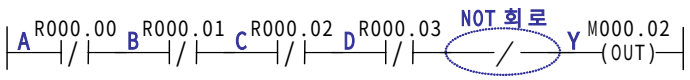
다음과 같은 두 회로는 같은 기능을 한다. 즉, b 접점의 기능과 a 접점의 반대가 기능이 같다.





이 기능을 이용하여 시퀀스 회로를 부울 대수식으로 변환하여 간단하게 만들기도 한다.
즉, 아래와 같이 OR 회로(병렬회로)로 구성된 경우를 변환하면 다음과 같은 AND 회로로 된다.





왼쪽의 OR 회로를 부울 대수식으로 표현하면
Y=A+B+C+D 로 나타내고, 이것을 변환하면
$\overline{Y}=\overline{(A+B+C+D)}=\overline{A}*\overline{B}*\overline{C}*\overline{D}$ 이다. 여기서 출력 $\overline{Y}$ 는
Y 앞에 NOT 회로를 삽입하면 동일한 결과가 된다.

이와 같은 형식으로 복잡한 AND 회로을 OR 회로로 변환하여 쉽게 해석해도 동일한 결과를 얻을 수 있으며, 각종 인터록 회로를 쉽게 이해할 수 있는 프로그램으로 변환하기도 합니다.

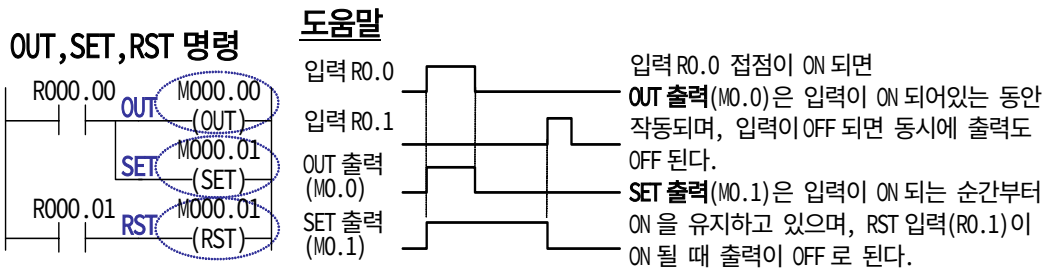
- 기본 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
OUT	—(OUT)—	연산결과 출력	1	전기종
SET	—(SET)—	출력을 ON 으로 SET	2	전기종
RST	—(RST)—	출력을 OFF 로 RESET	2	전기종

도움말

1. OUT 출력은 프로그램이 수행되는 매 스캔마다 연산이 된다.
2. OUT, SET, RST 등의 출력은 접점으로 사용되는 R,L,M,K 영역만 사용 가능하고, 특수 내부접점인 F 영역은 일부만 출력용으로 사용가능하다.
3. 입출력 번지의 R 영역의 경우 입력모듈이 장착된 번지는 출력으로 사용할 수 없다.
4. OUT 출력의 경우 입출력의 R 영역은 하나만 사용할 수 있으며, 내부접점인 L,M,K,F 영역은 두개 이상 사용이 가능하지만 최후에 사용한 OUT 명령으로 연산되므로 다중으로 사용할 경우에는 주의해야 한다.
5. SET 또는 RST 명령의 경우, 다중으로 사용이 가능하며, 접점을 한번 SET 하면 RST 명령으로 RESET 될때 까지 연산결과가 유지된다.
6. 정전유지 접점인 K 영역으로 SET 또는 RST 을 사용하면 전원이 OFF 되기 이전의 값을 유지하게 된다.
7. 출력명령 OUT,SET,RST 등은 입력이 없이 ROOT 에서 직접 출력으로 사용할 수 없으며, 항상 작동해야 하는 경우에는 운전중 항상 ON 되는 F0.15 번지의 a 접점을 사용한다.

사용예제



니모닉 프로그램 예

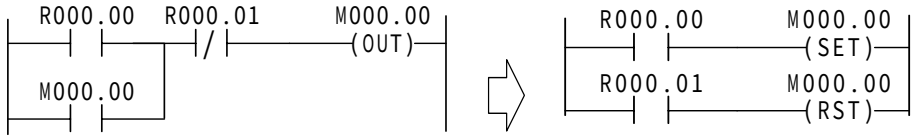
래더를 니모닉으로 변환하면

STR R000.00, OUT M000.00, SET M000.01, STR R000.01, RST M000.01 이 된다.

여기서 출력이 병렬로 연결된 회로는 출력회로를 반복적으로 사용하면 된다.

응용 1

다음과 같은 자기유지 접점을 SET, RST 명령으로 사용할 수도 있다.



- 기본 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
STR DIF	⊢R⊢	상승엣지 접점 시작 (⌐↑)	1	전기종
AND DIF	—R—	상승엣지 직렬 접속 (⌐↑)	1	전기종
OR DIF	└R┘	상승엣지 병렬 접속 (⌐↑)	1	전기종
STR DFN	⊢F⊢	하강엣지 접점 시작 (↓⌐)	1	전기종
AND DFN	—F—	하강엣지 직렬 접속 (↓⌐)	1	전기종
OR DFN	└F┘	하강엣지 병렬 접속 (↓⌐)	1	전기종

도움말

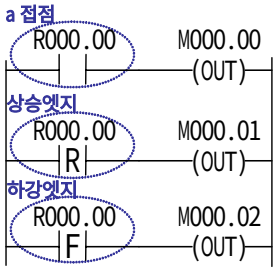
1. 엣지 접점은 값(상태)이 변하는 시점의 1SCAN 동안 ON 되는 접점을 말한다.
2. 상승엣지(DIF)는 접점이 OFF 에서 ON 으로 변하는 1Scan 동안 ON 으로 작동하고
   하강엣지(DFN)는 접점이 ON 상태에서 OFF 로 변하는 1Scan 동안 ON 되는 접점입니다.
3. 번지 지정은 아래와 같은 비트번지 영역에서 중복으로 사용할 수 있습니다.
   - NX-CPU700P, NX70-CPU70p2, NX70-CPU70p1(Ver2.0 이상), CPL9216A, CPL7215A =
     => R,L,M,K,F,TC 등의 전영역 사용이 가능합니다.
   - NX70-CPU70p1(V1.x), CPL9215A = R 전체, M0.0~M63.15, F, TC 의 전영역 사용가능.

   주의 : CPU70p1, 9215A 모델은 내부접점 M64.0~M127.15 까지, K 영역을 엣지로 사용할 수 없으며, 프로그램 S/W 에서는 에러처리를 하지 않고 CPU 에 Download 를 하면 에러로 처리하여 운전(RUN 작동)이 되지 않습니다.

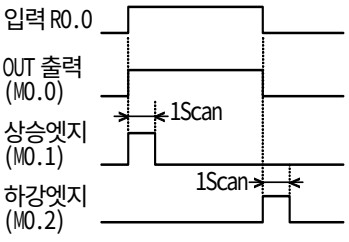
   주의 : SPC 시리즈(SPC-10, SPC-100/24S/120S/300 등)는 1 개 비트마다 1 개의 명령어만 사용할 수 있으며, 두번째 이후의 명령어는 에러 처리 없이 작동하지 않습니다.
4. 하나의 번지로 상승엣지 및 하강엣지를 다중으로 사용할 수 있습니다.
5. 엣지명령어도 시퀀스와 같이 순차적으로 작동하므로 프로그램 중간에 출력을 변화시키면 출력이전의 엣지동작과 출력이후의 엣지동작이 달라질 수도 있습니다.

사용예제

상승,하강엣지



도움말



입력 R0.0 접점이 ON 되면
OUT 출력(M0.0)은 입력과 같으며,
상승엣지는 입력이 OFF 에서 ON 되는 1 Scan 동안 출력이 ON 되는 기능이며,
하강엣지는 입력이 ON 에서 OFF 로 하강하는 1 Scan 동안 ON 되는 접점이다.

니모닉 프로그램 예

엣지 접점도 a 접점이나 b 접점과 동일한 형태로 사용되며, 시작회로 와 직렬회로, 병렬회로등에 따라 니모닉 명령어가 달라진다. 사용예제를 니모닉으로 변환하면,
STR R00.00, OUT M00.00, STR DIF R00.00, OUT M00.01, STR DFN R00.00, OUT M00.02
등으로 변환할 수 있다

- 기본 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
MS	—(MS)—	마스터 블럭(일괄처리) 시작	1	주 1)
MR	—(MR)—	마스터 블럭(일괄처리) 종료	1	주 1)
MCS	—┬—	회로분기 시작	1	전기종, 주 2)
MCR	└····—	분기 종료 (분기 끝에 존재)	1	전기종, 주 2)

주 1) - MS, MR 명령은 CPU F/W 버전 2.0 이상 및 WinGPC 3.70 이상에서 작동됩니다.
- WinGPC V3.70 이상, PLC V2.0 이상에서 MCS, MCR 명령이 분기로 변환(Upgrade)되었습니다.

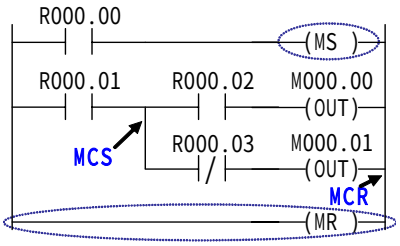
주 2) – WinGPC 버전에 따라 표현방법이 다릅니다.
- WinGPC 버전 3.70 이상에서 출력으로 MS, MR 로 작동하고, WinGPC 3.6 이하 에서는 회로분기 형태로 변환이 되며, 2.xx 이하 버전은 모두 출력으로 표현됩니다.

도움말

1. 출력 MS 및 MR 명령은 별도의 번지를 지정하지 않고, 명령어만 사용되며, 마스터 블록 비트로 작동합니다.
2. MS, MR 은 항상 쌍으로 구성해야 하며, 네스팅(MS 영역내에서 반복사용하는 구조)을 최대 16 개 까지 사용할 수 있고, 그 이상에서는 OverFlow 현상이 생길 수 있습니다. 네스팅을 사용하는 수량은 CPU 기종별로 차이가 납니다(예 SPC 는 8 개).
3. CPU 에 저장된 MCS,MCR 을 WinGPC3.0 이상에서 구동하면 회로분기 형식으로 나타나며, 니모닉에서 확인되며, WinGPC V2.xx 이하에서는 출력 MCS, MCR 로 표시됩니다.
4. MS 명령 앞에는 항상 명령어가 있어야 하며, ROOT 에 직접 연결할 수 없으며, MR 명령어 앞에는 접점명령을 사용하지 않고 Root 에 직접 연결해야 합니다.
5. MCS, MCR 명령어는 WinGPC3.70 버젼 이상에서는 회로분기 형식으로만 사용할 수 있고, 니모닉 명령어에서 확인 가능합니다.

사용예제 1

<그림 1>(CPU V1.3~, WinGPC 3.5~이상)

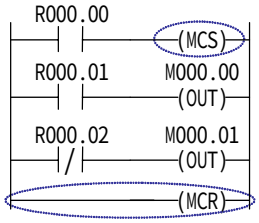


도움말

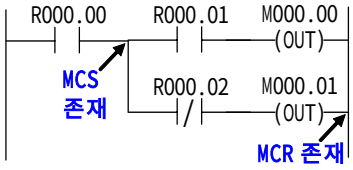
MS 조건의 R0.0 접점이 ON 된 후 MS/MR 블록의 접점이 ON 되어야 출력이 ON 됩니다.
즉,R0.0, R0.1, R0.2 가 ON 되면 출력 M0.0 ON 이고 R0.0= ON, R0.1= ON, R0.3= OFF 이면 M0.1 이 ON, R0.0 이 OFF 되면 모든 출력이 OFF 됩니다.

사용예제 2 : MS/MR, MCS/MCR 명령어 기능비교(WinGPC3.x 이하버젼)

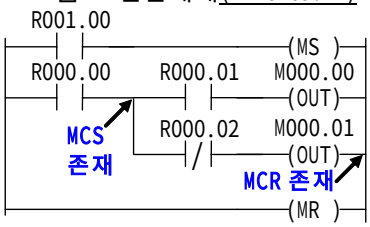
<그림 2> WinGPC2.x 버전



<그림 3>변환예제(WinGPC3.1x)



<그림 4>변환예제(WinGPC3.7x)



도움말

1. WinGPC 2.x 버전에서는 MCS, MCR 기능은 <그림 1, 2>와 같이 표현 및 사용할 수 있으며, <그림 3>은 WinGPC3.0, 3.1 버전에서 표시방법이고, <그림 4>는 WinGPC3.7x 이상에서 표시가능한 형식이며, MS, MR 명령은 3.7 버전 이하에서 사용할 수 없습니다.

2. 래더로 프로그래밍 작업시에는 MCS/MCR 명령어를 사용하지 않고, 분기 형식으로 작업을 해도 실제 CPU 에 운영되는 프로그램에는 자동으로 MCS, MCR 명령이 포함되도록 되어 있으며, CPU 의 프로그램을 Upload 하면 작업한 형식으로 나타나지 않을 수도 있습니다.

3. 래더 프로그램을 PC 에 저장(Compile)하여 PLC 로 전송할 경우에는 작업한 형태와 동일하게 표현되지만, PLC 에 있는 프로그램을 Upload 하여 래더로 변환(Decompile)하면 WinGPC 의 버전에 따라 그림 2,3,4 와 같이 다르게 표현됩니다.

   즉, WinGPC V3.1x 이하에서 <그림 2>와 같이 프로그램을 하여 PLC 에 저장한 후, WinGPC V3.x 이상에서 Upload 하면 <그림 3>과 같이 표현되며, WinGPC2.x 이하에서 Upload 하면 항상 <그림 2>와 같이 표시됩니다.

4. WinGPC V3.7x 에서는 <그림 1>과 같이 프로그램을 할 수 없으며, CPU 버젼 V1.2x 이전에서는 동작을 하지 않고, PLC V2.0 이상에서는 정상으로 동작합니다. 이전 버전(PLC V1.2x 이하)은 그림 3 과 같은 방식으로 프로그램 해야 정상 동작하게 됩니다.

5. WinGPC 3.7 이상 에서는 CPU 의 ROM 버전에 상관없이 자동으로 MCS/MCR, MS/MR 명령어로 전환되도록 수정되었습니다.

니모닉 프로그램 예

1. 사용예제를 니모닉으로 변환하면, <그림 2, 그림 3 예제>

   WinGPC 에서 니모닉변환은 STR R0.0, MCS, STR R0.1, OUT M0.0, STN R0.2, OUT M0.1, MCR 등으로 동일하지만, WINGPC V3.5x 이상 및 CPU 버젼 2.0x 이상에서는 MS, MR 명령을 추가로 사용할 수 있으며, 이전 버전에는 저장할 수 없습니다.

- 기본 명령어

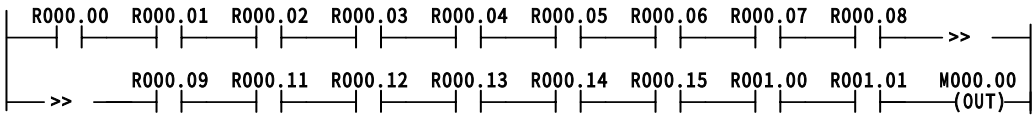
니모닉 명령	래더 표시	설 명	워드수	적용기종
-	—≫—	확장 (Entension) (AND 조건 확장시 사용) (WinGPC3, GPC5 등 전용)	-	전기종

도움말

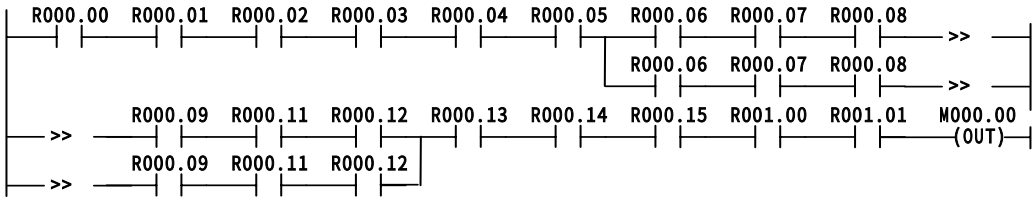
1. 확장기능은 연속으로 한번만 사용할 수 있으며, 병렬구조의 회로로 사용이 가능하고, 항상 쌍으로 구성이 되어야 한다.
2. 확장이 포함된 렁(RUNG)은 동일한 하나의 RUNG 으로 되며, 다른 RUNG 으로 확장할 수는 없습니다.
3. 확장기능은 명령어가 아니며, CPU 에서 별도의 메모리를 할당 받지 않고 WinGPC 상에서 표현하는 기호입니다.

사용예제 : 확장기능 사용예

<그림 1>

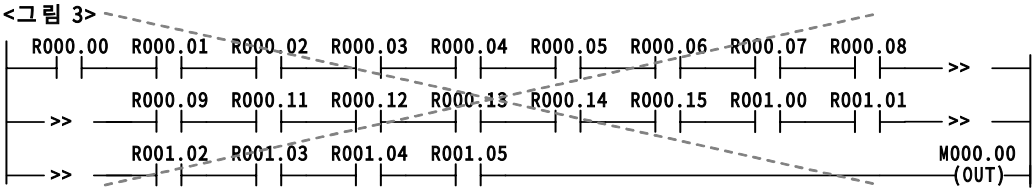


<그림 2>



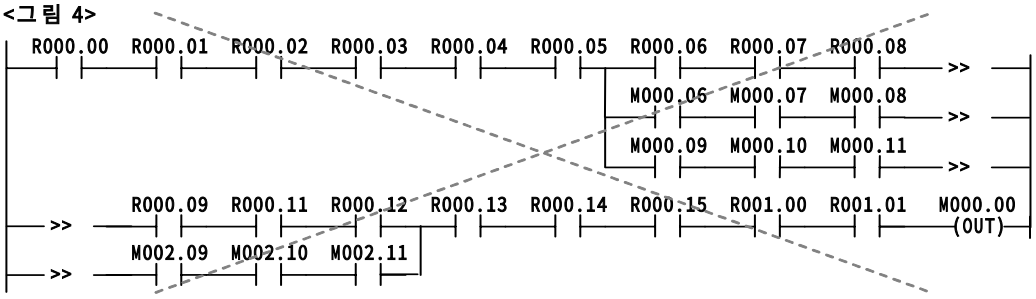
확장기능을 잘못 사용한 예제

<그림 3>



➔ 확장기능을 연속적으로 사용할 수 없으며, 이러한 경우 R001.01 이후에 내부접점 출력을 사용하고 입력에 사용한 내부접점을 이용하면 됩니다.

<그림 4>



➔ 확장은 항상 쌍으로 구성되어야 하므로 보내는 개수와 받는 수량이 동일해야 한다.

- 타이머/ 카운터/ SR 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
TIM	TIM Ch= SV=	ON 딜레이 타이머 (한시동작 순시정지 타이머)	3	전기종
TOF	TOF Ch= SV=	OFF 딜레이 타이머 (순시동작 한시정지 타이머)	3	주 1)
SST	SST Ch= SV=	싱글 쇼트 타이머 (순시동작/정지 타이머)	3	전기종

주 1) 적용기종 (TOF 명령)

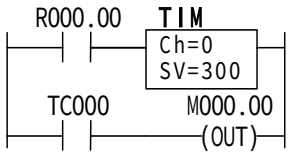
기 종	SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
지원여부	X	X	X	○	○	○	○	○

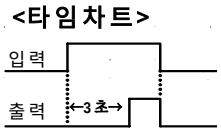
도움말

1. 타이머 기능은 3 종류가 있고, 카운터와 공용으로 최대 256 개를 사용할 수 있습니다.
2. 타이머와 카운터는 공용의 채널번호를 사용하며, 순서에 무관하게 0~255 번까지 지정하고, 동일한 채널번호를 사용할 수는 없습니다.
3. 타이머로 사용할 경우 채널번호 0~63 번 까지 지정하면, 자동으로 Time Base 가 10mSec (0.01 초)단위로 계산되고, 채널번호 64~255 번까지 지정하면 Time Base 가 100mSec(0.1 초)단위로 계산이 됩니다.
4. 설정치(SV)는 16 비트로 0 ~ 65,535 까지 지정할 수 있습니다.
5. 타이머나 카운터의 출력접점 번지는 " TC + 채널번호 " 로 지정합니다.
6. 각각의 채널별로 타이머로 사용할 경우에는 경과치 값이 정전유지가 안되며, 카운터로 사용할 경우에는 경과치 값이 정전유지가 됩니다.
7. 경과치가 증가하는 타이머(TIM, SST 등)는 설정치를 지나 출력이 ON 된 이후에도 경과치는 계속 증가하여 최대치(65535)에서 멈추게 되며, 입력조건이 OFF 되면 경과치가 초기화 됩니다.
8. 경과치가 감소하는 타이머(TOF)는 타이머가 작동할 때부터 경과치가 감소되고, 경과치가 ' 0 ' 이 되면 출력이 ON 되고 경과치가 멈추게 됩니다.
9. TIM 은 입력이 ON 된 후부터 타이머가 작동해서 지정시간 이후에 출력이 ON 되는 기능의 ON 딜레이 타이머 입니다.
10. TOF 는 입력이 OFF 된 시점에 타이머가 작동해서 지정시간 이후에 출력이 OFF 되는 기능의 OFF 딜레이 타이머 입니다.
11. SST 는 입력이 ON 되면서 동시에 타이머가 작동해서 지정한 시간이 되면, 출력이 OFF 되는 타이머를 말한다.

사용예제

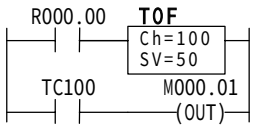
< ON 딜레이 타이머 - TIM >

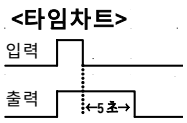




타이머 입력조건이 ON 되면 타이머 경과치가 증가하여 설정치 이상이 되면 출력접점이 ON 됩니다.
입력이 OFF 되면 타이머 경과치가 초기화 되며, 출력도 OFF 됩니다.

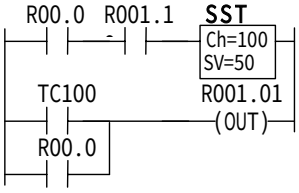
< OFF 딜레이 타이머 – TOF >

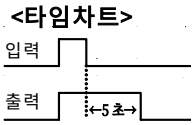




타이머 입력조건이 ON 되면 타이머 출력도 ON 되며, 입력이 OFF 되는 순간, 경과치가 설정치부터 감소하게 되며, 경과치가 '0' 이 되면 출력접점이 OFF 됩니다.

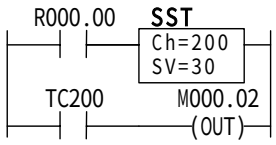
< 응용예제 : SST 를 이용한 OFF 딜레이 타이머 >

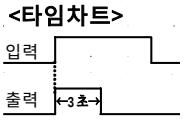




입력 접점과 동시에 출력도 ON 되며, 입력이 OFF 되는 순간부터 SST 타이머가 작동하고 설정값까지 증가한 후, 경과치와 설정치가 동일하면 출력접점이 OFF 됩니다.

< 싱글 쇼트 타이머 – SST >



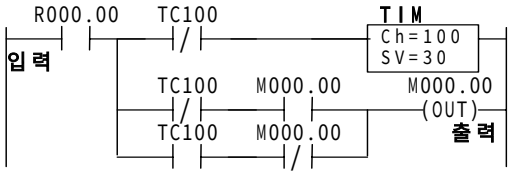


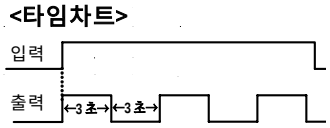
타이머 입력조건이 ON 되면 출력도 ON 되고, 타이머 경과치가 증가하여 설정치 이상이 되면 출력이 OFF 됩니다. 경과치가 설정치에 도달하기 전에 입력이 OFF 되면, 출력도 OFF 됩니다.

응용예제 1

■ 타이머 1 개를 이용한 플리커 회로

➔ 입력이 ON 되면 3 초간 ON, 3 초간 OFF 가 반복되는 회로(플리커 회로)

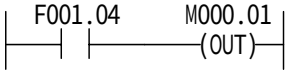


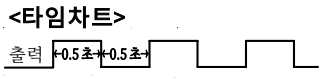


타이머의 SV 값을 조정하면 플리커 간격을 조정할 수 있습니다.
이 회로는 타이머와 교번동작 회로를 이용한 방법입니다.

■ 간단한 특수접점을 이용한 1 초 플리커 회로

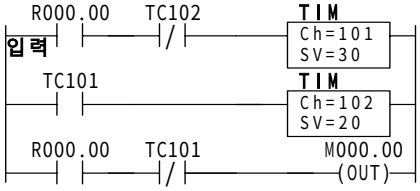
➔ PLC 내부 특수접점(F 영역)을 이용하여 1 초간격의 플리커 회로를 활용할 수도 있습니다.

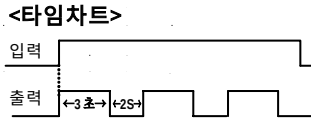




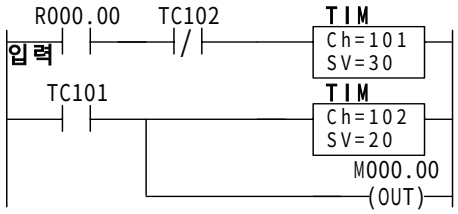
■ 타이머 2 개를 이용한 플리커 회로 2

➔ 입력이 ON 되면 3 초간 ON, 2 초간 OFF 가 반복되는 플리커 회로

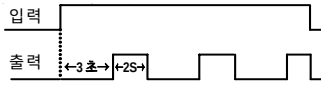




< 플리커 응용 회로 2 >



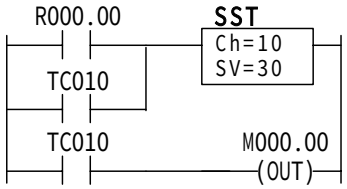
<타임차트>
응용 표현



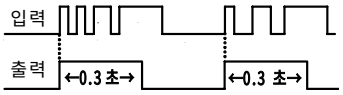
응용예제 2

■ 불안정한 신호의 안정화 회로

➔ 입력의 펄스 노이즈와 같은 불안정한 회로를 안정화 시킬 수 있는 회로



<타임차트>

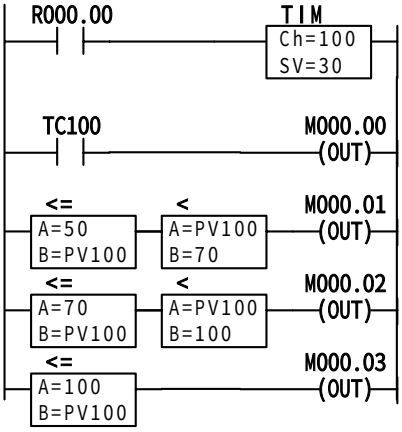


■ 타이머의 경과치를 이용한 단계별 제어회로

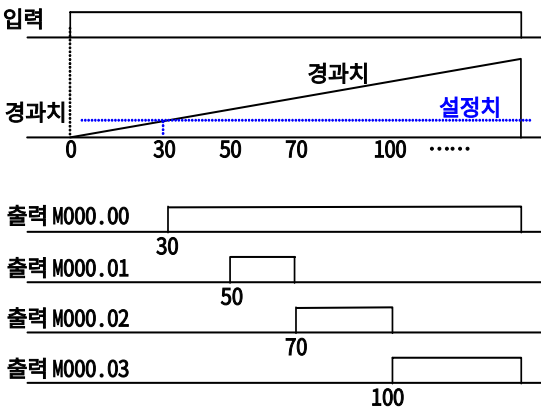
➔ 타이머의 경과치(PV)를 이용하여 단계별 동작회로를 구성할 경우에 이용합니다.

(비교명령어 부분은 별도로 참고 바랍니다)

<래더 프로그램>



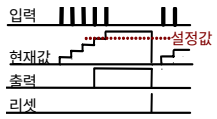
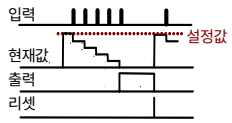
<타임 차트 비교>



도움말

1. 타이머 입력조건이 작동되고 있는 동안은 경과치(PV 값)가 최대치 까지 증가하므로 타이머의 경과치를 비교명령어로 영역을 설정하여 시간에 대한 구간별로 작동 프로그램을 구축하여 효과적인 회로구성을 할 수 있습니다.
2. 입력조건이 정지하면 모든 출력은 정지하게 됩니다.

- 타이머/ 카운터/ SR 명령어

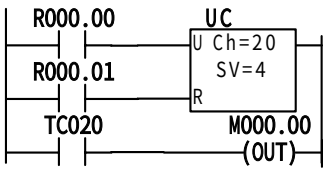
니모닉 명령	래더 표시	설 명	워드수	적용기종
UC	UC U Ch=020 SV=004 R	상승카운터 	3	전기종
DC	DC D Ch=021 SV=005 R	하강카운터 	3	전기종

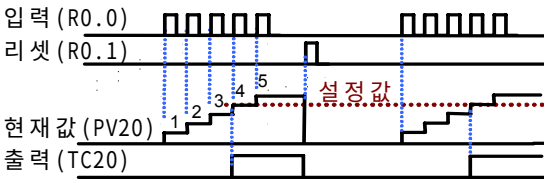
도움말

1. 카운터는 4 종류(UC, DC, RCT, UDC)가 있고, 타이머와 공용으로 최대 256 개를 사용할 수 있습니다.
2. 타이머와 카운터는 공용의 채널번호를 사용하며, 순서에 무관하게 0~255 번까지 지정하고, 동일한 채널번호를 사용할 수는 없습니다.
3. 설정치(SV)는 16 비트(0 ~ 65,535 까지)를 지정할 수 있습니다.
4. 타이머나 카운터의 출력접점 번지는 “TC + 채널번호”로 지정합니다.
5. 각 채널별로 카운터로 사용할 경우에는 경과치 값이 정전유지가 됩니다.
6. UC(상승 카운터)는 입력 접점이 OFF 에서 ON 으로 변경될 때마다 1 씩 카운터 되며, 카운터 값이 0 에서부터 최대치(65,535)까지 증가하고, 최대치에서는 입력에 무관하게 값을 유지합니다.
7. DC(하강 카운터)는 리셋 입력이 ON 되면 현재치가 설정치로 지정되며, 입력이 OFF 에서 ON 으로 변경될 때 마다 1 씩 감소하여 값이 ‘0’이 될 때, 출력접점이 ON 되고, 카운터가 정지됩니다. Reset 입력이 ON 되면, 출력이 OFF 로 변하고 현재치는 설정치로 지정됩니다. 리셋 입력이 OFF 된 후부터 카운터는 작동을 합니다. 초기 프로그램 다운로드 후 작동시에는 리셋입력을 ON 시킨 후 작동시키십시오.
8. 모든 카운터의 리셋입력이 ON 되어 있으면 입력의 영향을 받지 않고 초기화 됩니다.

사용예제

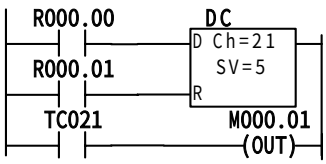
< UP Counter >

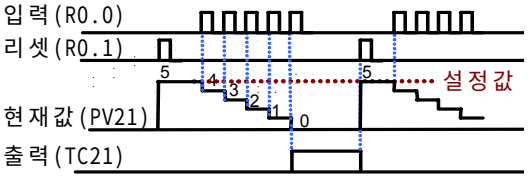




상승 카운터는 입력이 ON 될 때마다 현재치가 1 씩 증가하며, 설정값 과 동일하거나 이상이 되면 출력접점이 ON 됩니다. 리셋입력이 ON 되면 0 으로 된다.

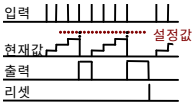
< Down Counter >





하강 카운터는 입력이 ON 될 때마다 현재치가 1 씩 감소하여 ‘0’이 되면 출력접점이 ON 됩니다. 리셋입력이 ON 되면 출력은 정지하고 현재치가 설정치로 된다.

- 타이머/ 카운터/ SR 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
RCT	RCT T Ch=022 SV=004 R	링 카운터 	3	주 1)
UDC	UDC U Ch=023 SV=003 D R	상승/하강 카운터 	3	전기종

주 1) 적용기종

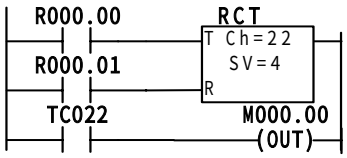
SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	○	○	○	○	○

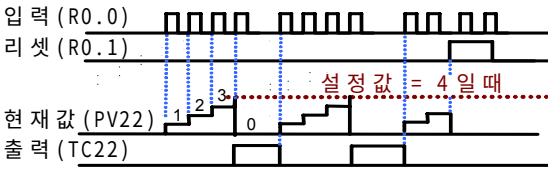
도움말

1. 카운터는 4 종류가 있고, 타이머와 공용으로 최대 256 개를 사용할 수 있습니다.
2. 타이머와 카운터는 공용의 채널번호를 사용하며, 순서에 무관하게 0~255 번까지 지정하고, 동일한 채널번호를 사용할 수는 없습니다.
3. 설정치(SV)는 16 비트(0 ~ 65,535 까지)를 지정할 수 있습니다.
   타이머나 카운터의 출력접점 번지는 " TC + 채널번호 " 로 지정합니다.
4. 각 채널별로 카운터로 사용할 경우에는 경과치 값이 정전유지가 됩니다.
5. RCT(링 카운터)는 입력 접점이 OFF 에서 ON 으로 변경될 때마다 ' 0 ' 부터 1 씩 카운터 되며, 카운터 횟수가 설정치가 되면 0 으로 값이 변하면서 출력이 ON 됩니다. 그 다음 입력부터 다시 증가하면서 카운터 값이 반복됩니다.
6. UDC(상승/하강 카운터)는 상승입력(U) 조건이 OFF 에서 ON 으로 변경될때 카운터가 1 씩 증가 되며, 하강입력(D) 조건이 OFF 에서 ON 으로 변경될때 카운터가 1 씩 감소하고 0 에서는 감소하지 않습니다. 카운터 현재치가 설정치 보다 크거나 ' 0 ' 이 되면 출력이 ON 되며, 리셋입력이 ON 되면 출력은 OFF 되고, 경과치는 ' 0 ' 이 됩니다.
   (단, SPC 시리즈는 하강 카운터가 현재치 0 에서도 (-) 로 감소됩니다.
7. 카운터의 리셋입력이 ON 되어 있으면 입력의 영향을 받지 않고 초기화 됩니다.

사용예제

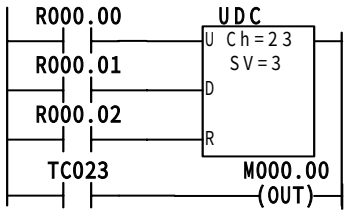
< Ring Counter >

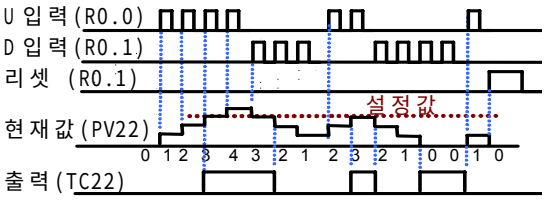




링 카운터는 설정치 주기로 현재치가 반복되며, 현재치가 0 이 되면서 출력이 ON 됩니다. 리셋 입력이 ON 되면 값은 0 으로 변하면서 출력은 OFF 상태가 됩니다.

< UP-Down Counter >





UDC 명령은 U 입력은 상승하고 D 입력은 하강한다. 출력은 설정값 이상과 0 에서 ON 된다.

- 타이머/ 카운터/ SR 명령어

니모닉 명령	래더 표시	설 명	워드수	적용기종
SR	SR I Sb=K1.4 Eb=K2.5 P R	시프트 레지스터 Sb K1.4 Eb K2.5 I I 값	4	주 1)

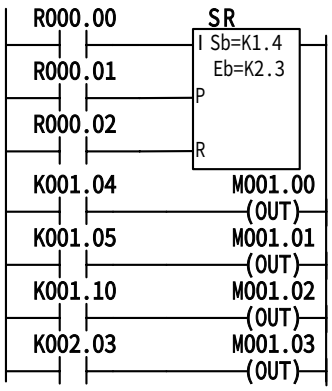
주 1) 적용기종

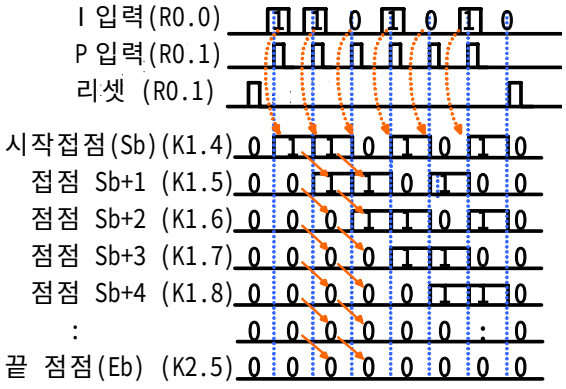
SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	○	○	○	○	○

도움말

1. SR 명령은 최대 256 개를 사용할 수 있으며, 자동으로 사용 갯수를 점검합니다.
2. 번지(ADDRESS)는 동일한 종류의 M 또는 K 영역에서 사용 가능하며, 지정한 비트의 범위는 2 ¯ 1024 비트까지 지정가능하고, K 영역을 사용하면 정전유지가 됩니다.
3. Sb 또는 Eb 의 번지는 시프트 방향을 표시하며, 중복하여 사용할 수도 있고, 역방향(Sb 번지가 Eb 보다 낮은 경우)의 번지지정도 가능합니다. 즉, Sb < Eb 일 경우 상위비트로 시프트 되고, Sb > Eb 일때는 하위비트로 시프트 됩니다.
4. ‘ 입력 I ’ 는 시작번지(Sb)의 데이터 값을 결정하며, ‘ 입력 P ’ 가 ON 될 때마다 전체 데이터를 1 비트씩 Sb 에서 Eb 방향으로 시프트 합니다.
5. 리셋입력이 ON 되면 지정된 모든 Data 가 ‘ 0 ’ 으로 초기화 됩니다.

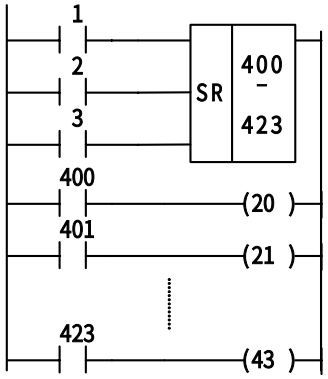
사용예제

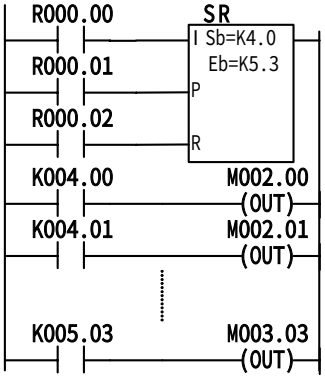




1) 리셋 입력이 ON 되면 모든 데이터가 0 으로 초기화 됩니다.
2) 처음 ‘ I 입력 ’ (입력 Data)이 ON 상태에서 ‘ P 입력 ’ (펄스 시프트 지령)이 ON 되면 데이터값 1 이 Sb 에 저장되며, 영역의 모든 비트 값은 Sb 비트에서 Eb 비트 방향으로 1 비트씩 시프트 됩니다.
3) 두번째 P 입력이 ON 되는 순간의 I 입력 Data (ON)가 Sb 에 저장되고, 기존 Sb 의 값은 Sb+1 비트로 각각 시프트 됩니다.
4) P 입력이 ON 되는 시점까지 데이터는 유지되며, ON 되는 순간에 시프트 동작이 반복되며, 마지막 비트(Eb)의 값은 Over flow 하고, 리셋 입력이 ON 되면 모든 데이터를 0 으로 초기화 됩니다.

응용예제 : SPC-120, SPC24 기종의 SR 명령을 대체하는 방법





<SPC120/24 기종의 예제> **<NX-p, Nplus 기종의 예제>**

1) SPC-120/24 기종의 경우 8 비트 단위로 번지지정이 되며, NX plus 기종은 16 비트 단위로 번지 지정이 됩니다.

2) 즉, 예제에서 각각 대응되는 번지는 다음 표와 같습니다.

SPC-120/24	NX-p,Nplus	SPC-120/24	NX-p,Nplus
400	K4.00	412	K4.10
401	K4.01	413	K4.11
402	K4.02	414	K4.12
403	K4.03	415	K4.13
404	K4.04	416	K4.14
405	K4.05	417	K4.15
406	K4.06	420	K5.00
407	K4.07	421	K5.01
410	K4.08	422	K5.02
411	K4.09	423	K5.03

SPC-120/24	NX-p,Nplus	SPC-120/24	NX-p,Nplus
20	M2.00	32	M2.10
21	M2.01	33	M2.11
22	M2.02	34	M2.12
23	M2.03	35	M2.13
24	M2.04	36	M2.14
25	M2.05	37	M2.15
26	M2.06	40	M3.00
27	M2.07	41	M3.01
30	M2.08	42	M3.02
31	M2.09	43	M3.03

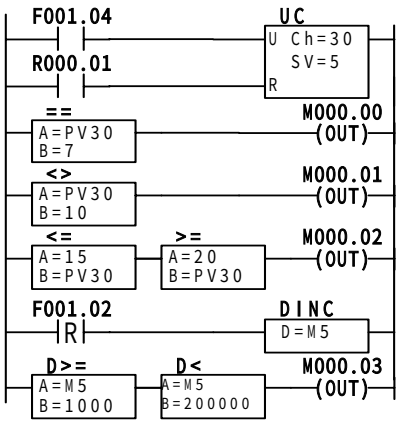
- 비교 명령어

니모닉명령	워드수	래더 표시	기능	설 명	적용기종
STR == AND == OR ==	3	== A= B=	A = B	A 값과 B 값이 같을때 ON. A,B 는 워드번지 또는 워드값	전기종
STR D== AND D== OR D==	4	D== A= B=	A = B	A 값과 B 값이 같을때 ON. A,B 는 더블워드번지 또는 더블워드값	전기종
STR <> AND <> OR <>	3	<> A= B=	A ≠ B	A 값과 B 값이 다를때 ON. <>은 ≠과 동일 A,B 는 워드번지 또는 워드값	전기종
STR D<> AND D<> OR D<>	4	D<> A= B=	A ≠ B	A 값과 B 값이 다를때 ON. <>은 ≠과 동일 A,B 는 더블워드번지 또는 더블워드값	전기종
STR > AND > OR >	3	> A= B=	A > B	A 값이 B 값보다 클때 ON. A,B 는 워드번지 또는 워드값	전기종
STR D> AND D> OR D>	4	D> A= B=	A > B	A 값이 B 값보다 클때 ON. A,B 는 더블워드번지 또는 더블워드값	전기종
STR >= AND >= OR >=	3	>= A= B=	A ≥ B	A 값이 B 값보다 크거나 같을때 ON. A,B 는 워드번지 또는 워드값	전기종
STR D>= AND D>= OR D>=	4	D>= A= B=	A ≥ B	A 값이 B 값보다 크거나 같을때 ON. A,B 는 더블워드번지 또는 더블워드값	전기종
STR <= AND <= OR <=	3	<= A= B=	A ≤ B	A 값이 B 값보다 작거나 같을때 ON. A,B 는 워드번지 또는 워드값	전기종
STR D<= AND D<= OR D<=	4	D<= A= B=	A ≤ B	A 값이 B 값보다 작거나 같을때 ON. A,B 는 더블워드번지 또는 더블워드값	전기종
STR < AND < OR <	3	< A= B=	A < B	A 값이 B 값보다 작을때 ON. A,B 는 워드번지 또는 워드값	전기종
STR D< AND D< OR D<	4	D< A= B=	A < B	A 값이 B 값보다 작을때 ON. A,B 는 더블워드번지 또는 더블워드값	전기종

도움말

1. 비교명령의 A 또는 B 영역은 R,L,M,K,W,SV,PV,SR 등의 워드(더블워드)번지 또는 데이터 값을 지정할 수 있습니다.
2. 데이터값 지정시 워드명령은 16 비트(65,535)이내, 더블워드는 32 비트(4,294,967,295) 까지 값을 지정할 수 있습니다.
3. 니모닉 명령은 접점의 형식처럼 각 비교명령마다 STR, AND, OR 등이 함께 사용됩니다.
4. 입력이 ON 되고, A 와 B 의 값을 비교한 결과에 따라 출력됩니다.

사용예제



UP 카운터를 이용하여 데이터가 증가하는 기능입니다.
F1.4 접점은 1 초마다 반복되는 신호임.

카운터의 경과치(PV30)가 7 일 때 출력 M0.0 이 ON.

카운터 경과치가 10 이 아닐 때 출력 M0.1 이 ON

경과치 PV30 이 15 보다 크고, 20 보다 작을 때 출력 M0.2 가 ON 됨.

F1.2 접점은 20mS 마다 ON/OFF 반복되는 접점이며, DINC 명령은 20mS 마다 1 씩 증가합니다.
M0005 의 더블워드가 1000 이상 200000 이하의 범위에서 출력 M0.3 이 ON 됩니다.

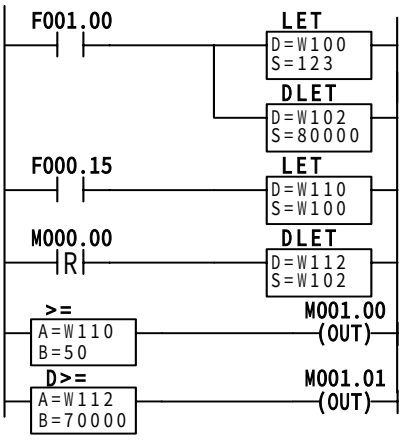
- 대입 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
LET	3	LET D= S=	수치(데이터) 대입 (S=Source, D=Destination) (D= 워드번지, S= 워드번지,워드값)	전기종
DLET	4	DLET D= S=	더블워드 대입 (S 의 값을 D 에 저장) (D= 더블워드번지, S= 더블워드, 더블워드값)	전기종

도움말

1. 대입명령은 목적지 D 번지에 S 값을 직접 입력하거나 다른 워드(더블워드)번지의 값을 대입(전송) 하는 기능을 가지고 있습니다.
2. 대입명령의 D 영역은 R,L,M,K,W,SV,PV,SR 등의 워드(더블워드)번지를 지정할 수 있으며, S 영역은 LET 명령에서는 워드번지 또는 16 비트(65,535)이내, DLET 명령에서는 더블워드번지 또는 32 비트(4,294,967,295) 까지 값을 지정할 수 있습니다.
3. 대입명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 항상 동작해야 하는 경우에는 F0.15 번지(운전시 항상 ON)를 ‘a 접점’으로 사용합니다.

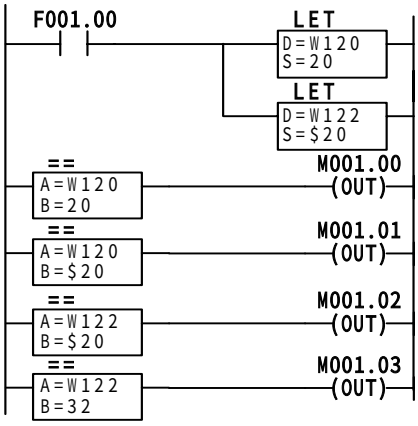
사용예제



F1.0 접점은 최초 1Scan 동안 ON 되는 신호임.
PLC 가 처음 운전(수행)되는 SCAN 에서 W100 워드에 데이터값 123 을 삽입(전송)합니다.
처음 운전 스캔에서 W102 번지에 데이터값 80000 을 대입(전송)합니다.

PLC 가 운전하는 매 스캔마다 S 의 W100 의 값을 D 의 W110 에 저장합니다.
접점 M0.0 이 ON 되는 순간 W102 값을 W112 워드에 대입(전송)함. *주의) DLET 명령은 더블워드 단위로 전송하므로 W102,W103 의 값을 W112,W113 에 대입하는 기능을 가지고 있습니다.
W110 워드가 50 보다 크면 M001.00 이 ON 됩니다.
W112 더블워드가 70000 보다 크면 M001.01 이 ON 됩니다.

응용예제 : 16 진수(HEX) 값 입력방법



PLC 가 처음 운전(수행)되는 SCAN 에서
W120 워드에 데이터값 20(10 진수)을 삽입(전송)하며,
W122 워드에 데이터값 $20(16 진수표현)(10 진수=32)을 삽입(전송)합니다.
(16 진수 입력은 숫자 앞에 $를 삽입하면 됩니다.)

저장된 데이터를 확인하는 프로그램입니다.
즉, W120 값이 20 이면 출력 M0.0 이 ON 됩니다.
W120 값이 $20(16 진수)일 때 출력 M0.1 이 ON 됩니다.
(10 진수로 32 값이므로 출력은 OFF 상태가 됩니다)

W122 값이 $20(16 진수)이면 출력 M0.2 가 ON 됩니다.

W122 값이 32(10 진수)이면 출력 M0.3 이 ON 됩니다.
즉, $20 또는 32 로 입력해도 동일한 값이 입력됩니다.

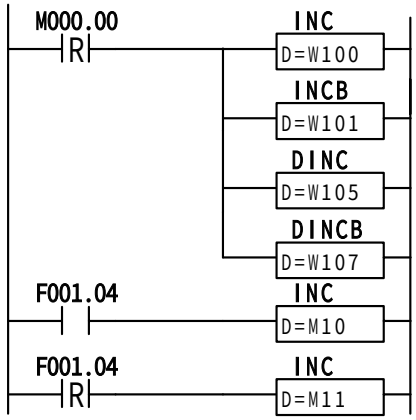
- 증가 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
INC	2	INC D=	2 진수 증가 (10 진수 증가) - 1 워드 (Binary Increment)	전기종
DINC	2	DINC D=	2 진수 증가 (10 진수 증가) - 더블워드	전기종
INCB	2	INCB D=	BCD 증가 - 1 워드 (BCD Increment)	전기종
DINCB	2	DINCB D=	BCD 증가 - 더블 워드	전기종

도움말

1. 증가명령은 D 번지의 값을 1 씩 증가시키는 기능을 가지고 있으며, Binary 증가 또는 BCD 증가 명령어가 있습니다.
2. Binary 증가는 비트단위로 증가하고, BCD 증가는 BCD 숫자단위로 구성하는 4 개의 비트단위로 증가하며, 16 진수의 A~F 까지 값을 제외시킨 값과 동일합니다.
3. D 번지는 R,L,M,K,W,SV,PV,SR 등의 워드(더블워드)번지를 지정할 수 있으며, 워드명령은 16 비트(65,535)이내, 더블워드명령은 32 비트(4,294,967,295) 까지 값이 증가 합니다.
4. 증가된 값이 최대치를 넘으면 Overflow 가 되어 0 값이 되고, 계속해서 처음부터 다시 카운터 합니다.
5. 증가명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 카운터 기능과 같이 증가시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제



M1.0 접점이 ON 될 때마다 1Scan 동안(한번) 증가명령이 작동되며, 각각의 명령어가 모두 다른 기능을 가지고 있습니다.
즉, INC 명령은 Binary(10 진수값) 단위로 1 워드,
INCB 명령은 BCD 단위로 1 워드 증가,
DINC 명령은 Binary(10 진수값) 단위로 2 워드,
DINCB 명령은 BCD 단위로 2 워드 증가 기능.
M10 과 M11 은 입력조건을 접점으로 하는 경우와 엣지명령으로 하는 경우의 차이점을 표시합니다.
즉, F1.4 접점은 0.5 초동안 ON, 0.5 초 동안 OFF 동작을 하므로 ‘ a 접점 ’ 조건의 M10 워드는 ON 상태에서 SCAN 할때마다 1 씩 증가하고, 엣지명령 조건의 M11 워드는 1 초마다 1 씩 증가합니다.

주) BCD 란?

BCD 는 ‘ Binary Coded Decimal’ 의 약자로 ‘ 2 진 부호화 10 진수’ 라고 부르며, 4 개의 비트를 이용하여 10 진수를 표현한 코드방식을 말한다.
즉, 10 진법에서 각 자리수의 10 진 숫자를 2 진 숫자로 나타내는 방식이다.
예를들어 10 진수 23 은 BCD 로 0010 0011 이며, 16 진수 표현은 $23 이 된다.
10 진수 57 은 BCD 로 0101 0111 로 표현되고, 16 진수로 $57 이며,10 진수로는 87 이된다.
워드의 BCD 값은 0 ~ $9999(10 진수로는 39,321 까지 처리되며, 더블워드 BCD 값은 0~$99999999 (10 진수로 2,576,980,377)까지 값을 처리할 수 있습니다.

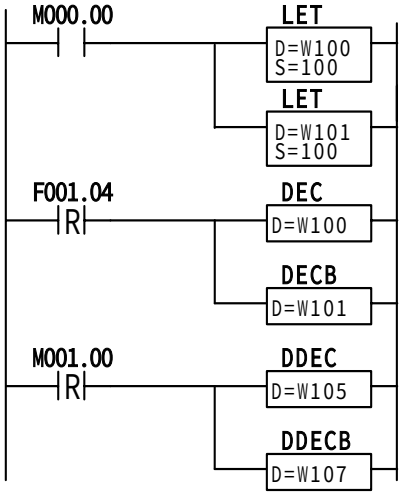
- 감소 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
DEC	2	DEC D=	2 진수 감소 (10 진수 감소) - 1 워드 (Binary Decrement)	전기종
DDEC	2	DDEC D=	2 진수 감소 (10 진수 감소) - 더블워드	전기종
DECB	2	DECB D=	BCD 감소 - 1 워드 (BCD Decrement)	전기종
DDECB	2	DDECB D=	BCD 감소 - 더블 워드	전기종

도움말

1. 감소명령은 D 번지의 값을 1 씩 감소시키는 기능을 가지고 있으며, Binary 감소 또는 BCD 감소 명령어가 있습니다.
2. Binary 감소는 비트단위로 감소하고, BCD 감소는 BCD 숫자단위로 구성하는 4 개의 비트단위로 감소하며, 16 진수의 A~F 까지 값을 제외시킨 값과 동일합니다(별도 참조).
3. D 번지는 R,L,M,K,W,SV,PV,SR 등의 워드(더블워드)번지를 지정할 수 있으며, 워드명령은 16 비트(65,535)이내, 더블워드명령은 32 비트(4,294,967,295) 이내에서 값이 감소 합니다.
4. 값이 0 에서 감소가 되면 Overflow 가 되고, 계속해서 최대치부터 다시 1 씩감소 합니다.
5. 감소명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 카운터 기능과 같이 감소시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제



M0.0 접점이 ON 되면 W100, W101 에 값이 100 으로 저장됩니다.

F1.4 접점에 의해 1 초 단위로 ON 될 때마다 한번씩 감소명령이 작동되며, 각각의 명령어가 모두 다른 기능을 가지고 있습니다.

즉, DEC 명령은 Binary 단위로 1 씩 감소하고,
DECB 명령은 BCD 단위로 1 워드 범위내 감소합니다

DDEC 명령과 DDECB 명령의 W105, W107 워드는 초기조건이 ‘ 0 ’ 상태에서 1 씩 감소하므로 최대치부터 감소하는 값이 되며,
DDEC 명령은 Binary 단위로 2 워드내 감소,
DDECB 명령은 BCD 단위로 2 워드내 감소합니다.

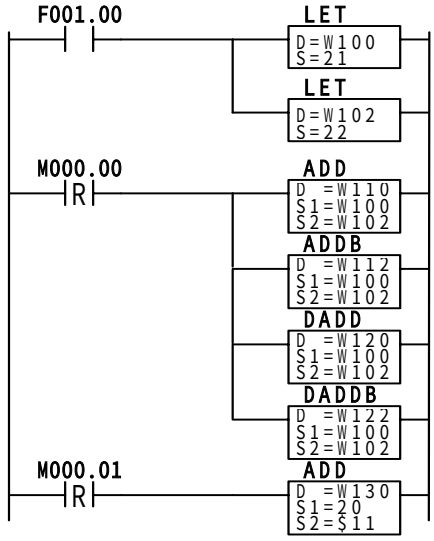
- 산술연산 명령어 (덧셈)

니모닉명령	워드수	래더 표시	설 명	적용기종
ADD	4	ADD D = S1= S2=	2 진수 덧셈 (10 진수 덧셈) - 1 워드 (Binary addition)	전기종
DADD	5	DADD D = S1= S2=	2 진수 덧셈 (10 진수 덧셈) - 더블워드	전기종
ADDB	4	ADDB D = S1= S2=	BCD 덧셈 - 1 워드 (BCD addition)	전기종
DADDB	5	DADDB D = S1= S2=	BCD 덧셈 - 더블 워드	전기종

도움말

1. 덧셈명령은 S1 과 S2 의 값을 더하여 D 에 저장하는 기능을 가지고 있으며, Binary 덧셈과 BCD 덧셈의 연산명령이 있습니다.
2. Binary 덧셈은 비트단위로 덧셈하고, BCD 덧셈은 BCD 숫자단위로 구성하는 4 개의 비트단위로 덧셈하며, HEX 로 표시되는 숫자의 덧셈 표현과 동일합니다(예제참조).
3. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드)번지만 지정할 수 있습니다.
4. BCD 연산명령(ADDB, DADDB 등)의 S1 과 S2 에 숫자(값)를 입력할 경우에는 BCD 단위로 입력하며, 숫자 앞에 $를 붙여 값을 입력합니다. (예, $100, $300 등)
5. Binary 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 연산할 수 있고, BCD 워드는 0 ~ $9999(10 진수로는 39,321)까지 연산되며, 더블워드 BCD 값은 0~$99999999 (10 진수로 2,576,980,377)까지 연산이 가능합니다.
6. 연산한 값이 최대치를 넘으면 캐리비트를 ON 시킨 후 최대치를 지나 나머지 값이 저장됩니다. 즉, 워드연산에서 65530 + 10 을 하면 캐리비트 F1.8 을 ON 시키고, 최대치 65536 을 뺀 나머지 값인 4 를 저장하게 됩니다.
7. 덧셈명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



처음 CPU 가 운전될 때 F1.0 접점이 한 SCAN ON 되어
W100 에 10 진수 21(16 진수 $15)을 저장하고
W102 에 10 진수 22(16 진수 $16)을 저장합니다.

M000.00 접점에 의해 명령어 4 개가 동작하며,
처리된 값은 다음과 같다.

ADD 명령은 10 진수 덧셈으로 21+22=43 이 된다.
ADDB 는 BCD 덧셈으로 21+22=49 가 된다.
쉽게 설명하여 BCD 로 표현을 하면
$15 + $16 = $31 이 되고, $31=49(10 진수)이다.
DADD 와 DADDB 는 더블워드 연산으로 16 비트의
연산은 동일하며, W120 의 값은 43 이 되고,
W122 는 값이 $31(10 진수로 49) 로 연산된다.

명령어의 S1,S2 에는 숫자를 직접 입력해도 되며,
연산결과 D 에는 22+17($11)=42 의 값이 저장된다.

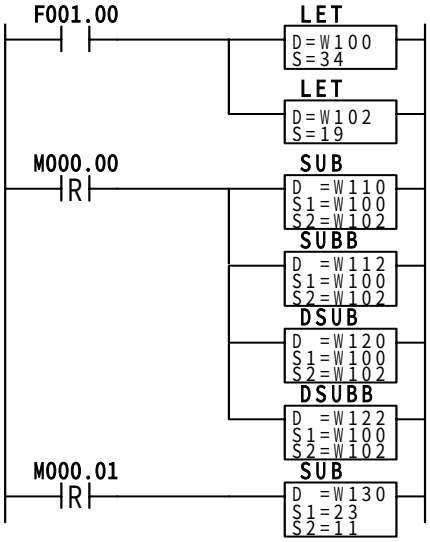
- 산술연산 명령어 (뺄셈)

니모닉명령	워드수	래더 표시	설 명	적용기종
SUB	4	SUB D = S1= S2=	2 진수 뺄셈 (10 진수 뺄셈) - 1 워드 (Binary subtraction)	전기종
DSUB	5	DSUB D = S1= S2=	2 진수 뺄셈 (10 진수 뺄셈) - 더블워드	전기종
SUBB	4	SUBB D = S1= S2=	BCD 뺄셈 - 1 워드 (BCD subtraction)	전기종
DSUBB	5	DSUBB D = S1= S2=	BCD 뺄셈 - 더블 워드	전기종

도움말

1. 뺄셈명령은 S1 에서 S2 를 감산하여 D 에 저장하는 기능을 가지고 있으며, Binary 뺄셈과 BCD 뺄셈의 명령이 있습니다.
2. Binary 뺄셈은 비트단위로 뺄셈하고, BCD 뺄셈은 BCD 숫자단위로 구성하는 4 개의 비트단위로 뺄셈하며, HEX 로 표시되는 숫자의 뺄셈 표현과 동일합니다(예제참조)
3. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드) 번지만 지정할 수 있습니다.
4. BCD 연산명령(SUBB, DSUBB 등)의 S1 과 S2 에 숫자(값)를 입력할 경우에는 BCD 단위로 입력하며, 숫자 앞에 $를 붙여 값을 입력합니다. (예, $10, $30 등)
5. Binary 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 연산할 수 있고, BCD 워드는 0 ~ $9999(10 진수로는 39,321)까지 연산되며,더블워드 BCD 값은 0~$99999999(10 진수로 2,576,980,377)까지 연산됩니다.
6. 연산한 값이 음수(마이너스)가 되면 캐리비트를 ON 시킨 후 음수(마이너스)값이 표현되지만 10 진수로 볼 경우 최대치에서 뺀 것처럼 표현됩니다. 즉, 워드연산에서 10-20 을 하면 -10 이 되지만 결과는 캐리비트 F1.8 을 ON 시키고, 최대치 65536 에서 10 을 뺀 65526 값이 저장됩니다. 이것을 부호를 포함한 값으로 보면 -10 이 됩니다.
7. 뺄셈명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



처음 CPU 가 운전될 때 F1.0 접점이 한 SCAN ON 되어 W100 에 10 진수 34(16 진수로 $22)을 저장하고 W102 에 10 진수 19(16 진수로 $13)을 저장합니다.

M000.00 접점에 의해 명령어 4 개가 동작하며, 처리된 값은 다음과 같다.

SUB 명령은 10 진수 뺄셈으로 34-19=15 가 된다.
SUBB 는 BCD 뺄셈으로 34-19=9 가 된다.
쉽게 설명하여 BCD 로 표현을 하면
$22 - $13 = $9 가 되고, $9=9(10 진수)이다.

DSUB 는 더블워드 뺄셈으로 연산결과는 15 이고, DSUBB 는 더블워드 BCD 뺄셈으로 연산결과 $9 가 W122 에 저장된다.

명령어의 S1,S2 에는 숫자를 직접 입력해도 되며, 연산결과 W130 에는 23-11=12 의 값이 저장된다.

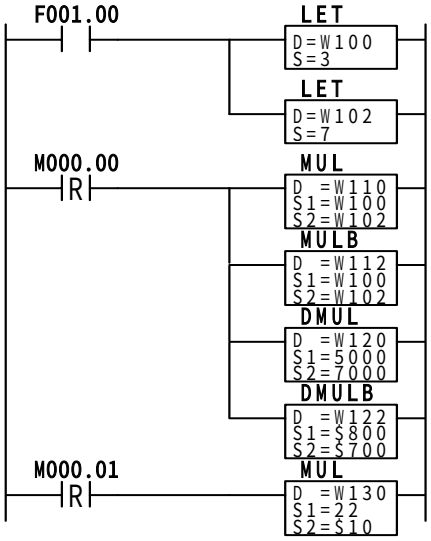
- 산술연산 명령어 (곱셈)

니모닉명령	워드수	래더 표시	설 명	적용기종
MUL	4	MUL D = S1= S2=	2 진수 곱셈 (10 진수 곱셈) - 1 워드 (Binary multiplication)	전기종
DMUL	5	DMUL D = S1= S2=	2 진수 곱셈 (10 진수 곱셈) - 더블워드	전기종
MULB	4	MULB D = S1= S2=	BCD 곱셈 - 1 워드 (BCD multiplication)	전기종
DMULB	5	DMULB D = S1= S2=	BCD 곱셈 - 더블워드	전기종

도움말

1. 곱셈명령은 S1 과 S2 를 곱셈하여 D 에 저장하는 기능을 가지고 있으며, Binary 곱셈과 BCD 곱셈의 명령이 있습니다.
2. Binary 곱셈은 비트단위로 곱셈하고, BCD 곱셈은 BCD 단위로 곱셈하며, HEX 로 표시되는 숫자의 곱셈 표현과 동일합니다(사용예제 참조).
3. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드) 번지만 지정할 수 있습니다.
4. BCD 연산명령(MULB, DMULB 등)의 S1 과 S2 에 숫자(값)를 입력할 경우에는 BCD 단위로 입력하며, 숫자 앞에 $를 붙여 값을 입력합니다. (예, $100, $300 등)
5. Binary 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 연산할 수 있고, BCD 워드는 0 ~ $9999(10 진수로는 39,321)까지 연산되며,더블워드 BCD 값은 0~$99999999(10 진수로 2,576,980,377)까지 연산됩니다
6. 연산한 값이 최대치를 넘게 되면 상위워드의 값은 특수워드 SR20 에 저장하고, 캐리비트를 ON 시킨 후 최대치만큼 뺀 나머지 하위 워드값을 D 에 저장합니다. 즉, 워드연산의 MUL 명령에서 20*3500 을 하면 값은 70000 이 되지만 결과는 SR20 에 1(65536)을 저장하고, 캐리 F1.8 을 ON 시키며,하위 워드값 4464 를 D 에 저장합니다.
7. 곱셈명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



처음 CPU 가 운전될 때 F1.0 접점이 한 SCAN ON 되어 W100 에 10 진수 3(16 진수로도 $3)을 저장하고 W102 에 10 진수 7(16 진수로 $7)을 저장합니다.

M000.00 접점이 ON 될 때마다 명령어 4 개가 동작하며, 처리된 값은 다음과 같다.
MUL 명령은 10 진수 연산으로 3×7=21 이므로 D 의 W110 워드에는 21 값이 저장된다.
MULB 는 BCD 곱셈으로 $3×$7=$21 이 되므로, D 에는 BCD 로 $21 이 저장되며, 10 진수로는 33 값이 된다.

DMUL 은 더블워드 곱셈으로 5000×7000=35000000 의 값이 D 에 저장된다.
DMULB 는 더블워드 BCD 곱셈 명령으로 $800×$700=$560000 이 D 에 저장되고, 10 진수로는 ' 5636096 ' 이 된다.
숫자를 직접 입력할 경우 10 진수 또는 16 진수로 입력해도 되며, 연산결과 D 에는 22×16($10)=352 의 값이 저장된다.

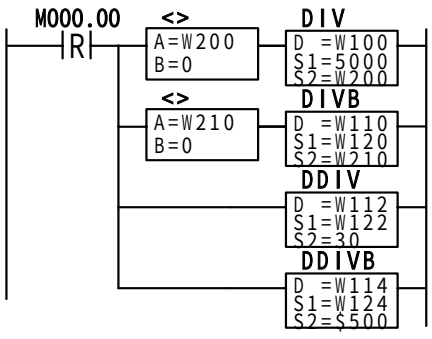
- 산술연산 명령어 (나눗셈)

니모닉명령	워드수	래더 표시	설 명	적용기종
DIV	4	DIV D = S1= S2=	2 진수 나눗셈 (10 진수 나눗셈) - 1 워드 (Binary division)	전기종
DDIV	5	DDIV D = S1= S2=	2 진수 나눗셈 (10 진수 나눗셈) - 더블워드	전기종
DIVB	4	DIVB D = S1= S2=	BCD 나눗셈 - 1 워드 (BCD division)	전기종
DDIVB	5	DDIVB D = S1= S2=	BCD 나눗셈 - 더블워드	전기종

도움말

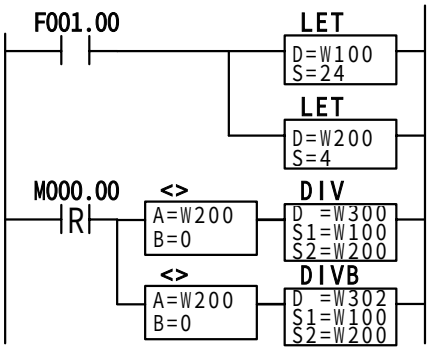
1. 나눗셈 명령은 분자 S1 에 분모 S2 를 나눈값(D = S1÷S2) 을 D 에 저장하는 기능을 가지고 있으며, Binary 나눗셈과 BCD 나눗셈의 명령이 있습니다.
2. 연산한 결과의 나머지값(소수점 이하값)은 내부 특수워드에 더블워드(SR22) 상태로 임시 저장합니다. 즉, 더블워드의 상위워드 SR23 에 나머지값의 상위값이 저장되고, 하위워드 SR22 에는 나머지의 하위값이 저장됩니다.
3. 나눗셈 명령 수행시 분모(S2)의 값이 ‘ 0 ’ 일때는 연산결과가 무한대(∞)가 되므로 CPU 에서 에러로 처리하고, 에러비트(F1.9)(나눗셈 연산시 분모가 0 인 에러)를 ON 시키며, 에러 LED 를 깜박입니다. 대부분 이러한 에러를 방지하기 위해서 나눗셈 명령어 입력조건에 ‘ 0 ’ 이 아닐 때 동작하도록 비교명령을 사용하여 오동작이 없도록 합니다(예제 참조).
4. Binary 나눗셈은 비트단위로 나눗셈하고, BCD 나눗셈은 BCD 단위로 나눗셈하며, HEX 로 표시되는 숫자의 나눗셈 표현과 동일합니다(사용예제 참조).
5. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드) 번지만 지정할 수 있습니다.
6. BCD 연산명령(DIVB, DDIVB 등)의 S1 과 S2 에 숫자(값)를 입력할 경우에는 BCD 단위로 입력하며, 숫자 앞에 $를 붙여 값을 입력합니다. (예, $100, $300 등)
7. Binary 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 연산할 수 있고, BCD 워드는 0 ~ $9999(10 진수로는 39,321)까지 연산되며, 더블워드 BCD 값은 0~$99999999 (10 진수로 2,576,980,377)까지 연산이 가능합니다.
8. 나눗셈명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제



5000÷W200 = W100 에 저장하는 명령으로 W200 이 ‘ 0 ’ 이 아닐 때 작동하도록 비교명령을 AND 조건으로 사용하였다.
DIVB 명령에서도 S2 인 W200 값이 0 이 아닐 때 작동하도록 비교명령을 사용하였고, 연산결과가 1 워드 범위내에서 처리됩니다.
DDIV 명령에서 분모값이 항상 30 으로 오류가 생기지 않기 때문에 동작조건만으로 가능합니다.
DDIVB 명령의 경우에도 S2 가 0 이 아니므로 오류가 생기지 않고, BCD 값을 지정하기 위해 숫자 앞에 $를 삽입하였다.

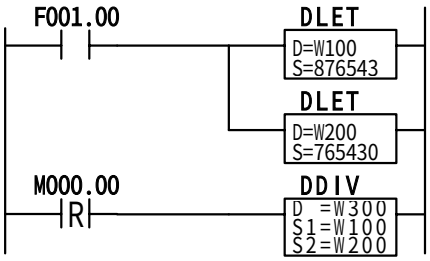
응용예제 1 (나눗셈)



첫 Scan 에서 W100 에 24, W200 에 4 의 Data 를 저장한다.

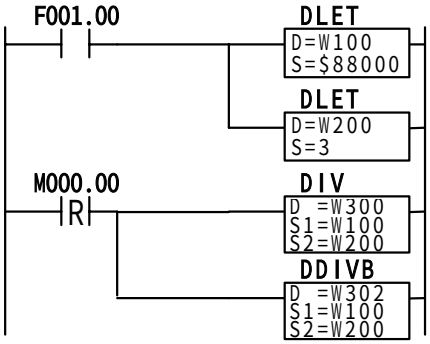
M0.0 접점이 ON 되고 나눗셈의 분모인 W200 이 0 이 아닐 때 W100÷W200 = W300 에 저장하게 된다. 즉, 연산결과는 24÷4 = 6 의 값이 저장된다.
BCD 나눗셈의 연산결과는 W100 의 BCD 값이 $18(24) 이므로 $18÷$4 = 몫이 4, 나머지가 2 이므로 W302 에는 $4 가 저장되고, 나머지 2 는 SR022 에 저장된다.

응용예제 2 (더블워드 나눗셈)



첫 Scan 에서 W100 에 876543, W200 에 765430 의 Data 를 저장한다.
M0.0 접점을 ON 하면, 더블워드 나눗셈 연산기능을 수행하며, 연산결과는 876543÷765430= 몫이 1 이고, 나머지 111113 ($1B209)값이 된다. 이 나머지 값은 더블워드로 SR22 에 저장되며, 상위워드(SR23)에 $0001 이 저장되고, 하위워드(SR22)에 45577($B209)가 저장된다.

응용예제 3



첫 Scan 에서 더블워드 W100 에 $88000(10 진수 557056)을 저장하고, W200 에 3 을 저장한다.

M0.0 접점을 ON 하면 연산이 되고, 워드 DIV 명령에 의해 W100 은 1 워드값이 되며, W100 은 $8000 이고, 연산은 32768($8000)÷3 = 몫이 10922, 나머지가 2 값이 되며, W300 에는 10922 가 저장되고, 나머지 값은 SR22 에 2 가 저장됩니다.

BCD 더블워드 나눗셈의 DDIVB 의 연산은 $88000÷$99000= 몫은 0, 나머지는 $88000 이 됩니다
결과치는 W302 에 0 이 저장되며, 더블워드 SR22= $88000 이며, 실제 SR22 에 $8000, SR23 에 $0008 입니다.

- 산술연산 명령어 (캐리포함 덧셈)

니모닉명령	워드수	래더 표시	설 명	적용기종
ADC	4	ADC D = S1= S2=	D = S1 + S2 + CY , (워드) 캐리 포함 2 진수 덧셈 (Decimal addition with carry)	주 1)
DADC	5	DADC D = S1= S2=	D = S1 + S2 + CY , (더블워드) 캐리 포함 2 진수 덧셈 (Decimal addition with carry)	주 1)
ADCB	4	ADCB D = S1= S2=	D = S1 + S2 + CY , (워드) 캐리 포함 BCD 덧셈 (BCD addition with carry)	주 1)
DADCB	5	DADCB D = S1= S2=	D = S1 + S2 + CY , (더블워드) 캐리 포함 BCD 덧셈 (BCD addition with carry)	주 1)

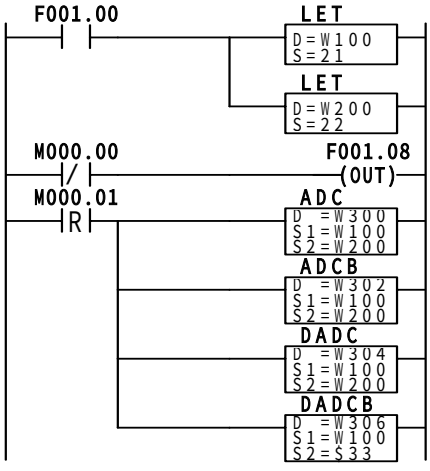
주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	○	○	○	○	○	○

도움말

1. ADC(캐리포함 덧셈명령)은 S1 과 S2, 캐리비트(F1.8)의 값을 더하여 D 에 저장하는 기능을 가지고 있으며, Binary 덧셈과 BCD 덧셈의 연산명령이 있습니다.
2. Binary 덧셈은 비트단위로 덧셈하고, BCD 덧셈은 BCD 숫자단위로 구성하는 4 개의 비트단위로 덧셈하며, HEX 로 표시되는 숫자의 덧셈 표현과 동일합니다(사용예제 참조).
3. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드)번지만 지정할 수 있습니다.
4. BCD 연산명령(ADCB, DADCB 등)의 S1 과 S2 에 숫자(값)를 입력할 경우에는 BCD 단위로 입력하며, 숫자 앞에 $를 붙여 값을 입력합니다. (예, $100, $300 등)
5. Binary 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 연산할 수 있고, BCD 워드는 0 ~ $9999(10 진수로는 39,321)까지 연산되며, 더블워드 BCD 값은 0~$99999999 (10 진수로 2,576,980,377)까지 연산이 가능합니다.
6. 연산한 값이 최대치를 넘으면 캐리비트를 ON 시키고 나머지 값(하위워드 값)이 저장됩니다. 즉, 워드연산에서 65530 + 10 을 하면 캐리비트 F1.8 을 ON 시키고, 하위워드의 값 4 를 저장하게 됩니다.
7. 덧셈명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



처음 CPU 가 운전될 때 F1.0 접점이 한 SCAN ON 되어 W100 에 10 진수 21(16 진수 $15)을 저장하고 W200 에 10 진수 22(16 진수 $16)를 저장합니다.

M000.00 접점에 의해 캐리비트(F1.8)을 임의로 ON 또는 OFF 시킬 수 있도록 합니다.
아래 연산은 캐리비트(F1.8)가 ON 으로 가정하여 연산한 결과의 값입니다.
M000.01 접점에 의해 명령어 4 개가 동작하며, 처리된 값은 다음과 같다.
ADC 명령으로 10 진수 연산의 21+22+1=44 가 된다.
ADCB 는 BCD 연산으로 $15+$16+1=$32 가 된다.

DADC 와 DADCB 는 더블워드 연산으로 16 비트의 연산은 동일하며, W304 의 값은 21+22+1=44 이고,

W306 은 $15+$33+1=$49 값이 된다.

- 산술연산 명령어 (캐리포함 뺄셈)

니모닉명령	워드수	래더 표시	설 명	적용기종
SBC	4	SBC D = S1= S2=	D = S1 - S2 - CY, (워드) 캐리 포함 2 진수 뺄셈 (Binary subtraction W/Carry)	주 1)
DSBC	5	DSBC D = S1= S2=	D = S1 - S2 - CY, (더블 워드) 캐리 포함 2 진수 뺄셈 (Binary subtraction W/Carry)	주 1)
SBCB	4	SBCB D = S1= S2=	D = S1 - S2 - CY , (워드) 캐리 포함 BCD 뺄셈 (BCD subtraction W/Carry)	주 1)
DSBCB	5	DSBCB D = S1= S2=	D = S1 - S2 - CY , (더블 워드) 캐리 포함 BCD 뺄셈 (BCD subtraction W/Carry)	주 1)

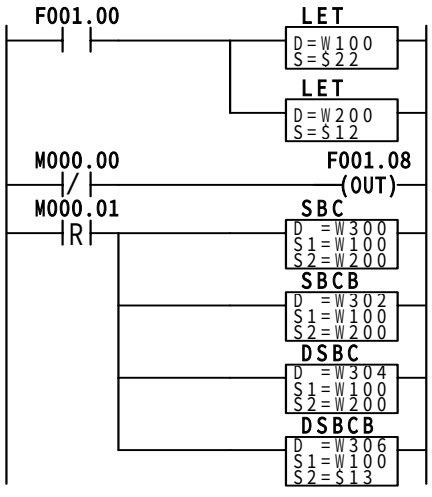
주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	○	○	○	○	○	○

도움말

1. SBC(캐리포함 뺄셈명령)는 S1 에서 S2 와 캐리(F1.8)를 감산하여 D 에 저장하는 기능을 가지고 있으며, Binary 뺄셈과 BCD 뺄셈의 명령이 있습니다.
2. Binary 뺄셈은 비트단위로 뺄셈하고, BCD 뺄셈은 BCD 숫자단위로 뺄셈하며, HEX 로 표시되는 숫자의 뺄셈 표현과 동일합니다(사용예제 참조).
3. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드) 번지만 지정할 수 있습니다.
4. BCD 연산명령(SBC, DSBC 등)의 S1 과 S2 에 숫자(값)를 입력할 경우에는 BCD 단위로 입력하며, 숫자 앞에 $를 붙여 값을 입력합니다. (예, $10, $30 등)
5. Binary 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 연산할 수 있고, BCD 워드는 0 ~ $9999(10 진수로는 39,321)까지 연산되며, 더블워드 BCD 값은 0~$99999999 (10 진수로 2,576,980,377)까지 연산이 가능합니다.
6. 연산한 값이 음수(마이너스)가 되면 캐리비트를 ON 시킨 후 음수(마이너스)값이 표현되지만 10 진수로 볼 경우 최대치에서 뺀 것처럼 표현됩니다. 즉, 워드연산에서 10-20 을 하면 -10 이 되지만 결과는 캐리비트 F1.8 을 ON 시키고, 최대치 65536 에서 10 을 뺀 65526 값이 저장됩니다. 이것을 부호를 포함한 값으로 보면 -10 이 됩니다.
7. 뺄셈명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



처음 CPU 가 운전될 때 F1.0 접점이 한 SCAN ON 되어 W100 에 10 진수 $22(10 진수 34)를 저장하고 W200 에 10 진수 $12(10 진수 18)을 저장합니다.

M000.00 접점에 의해 캐리비트(F1.8)을 임의로 ON 또는 OFF 시킬 수 있도록 합니다.
아래 연산은 캐리비트(F1.8)가 ON 으로 가정하여 연산한 결과의 값입니다.
M000.01 접점에 의해 다음과 같이 연산됩니다.

SBC 명령으로 D 는 34-18-1=15 가 된다.
SBCB 는 BCD 연산으로 $22-$12-$1=$9 가 된다.
DSBC 와 DSBCB 는 더블워드 연산으로 16 비트의 연산은 동일하며, W304 의 값은 34-18-1=15 이고,

W306 은 $22-$13-$1=$8 값이 된다.

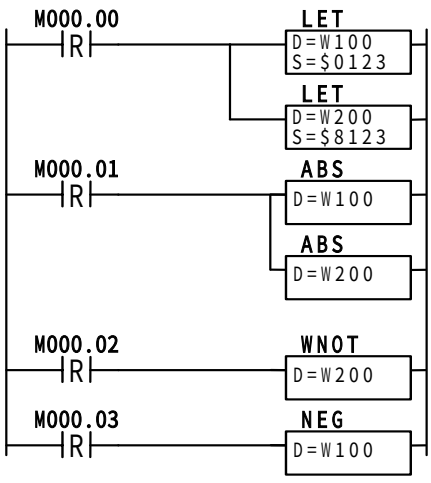
- 산술연산 명령어 (절대값,보수처리)

니모닉명령	워드수	래더 표시	설 명	적용기종
ABS	2	ABS D=	Absolute value (절대값 연산, D = |D|) (워드)	전기종
DABS	2	DABS D=	Absolute value (절대값 연산, D = |D|) (더블 워드)	전기종
WNOT	2	WNOT D=	NOT (워드 반전, 1 의보수) (워드)	전기종
DNOT	2	DNOT D=	NOT (더블워드 반전, 1 의보수) (더블 워드)	전기종
NEG	2	NEG D=	2 의 보수값을 D 에 저장 (1 의 보수+1 = 마이너스 값) Negative (워드)	전기종
DNEG	2	DNEG D=	2 의 보수값을 D 에 저장 (1 의 보수+1 = 마이너스 값) Negative (더블 워드)	전기종

도움말

1. ABS(절대값)명령은 최상위 비트(MSB)가 1 이면 2 의 보수를 취하고, 0 이면 그대로 둡니다. 즉, 10 진수 연산으로 마이너스 값은 +값으로 바꾸고, +값은 그대로 둡니다.
2. 절대값 명령은 ABS(워드 절대값) 및 DABS(더블워드 절대값)명령이 있습니다.
3. WNOT(1 의 보수) 명령은 워드(더블워드)의 각 비트를 반전시킨 것과 동일합니다.
4. 반전 명령은 WNOT(워드반전) 및 DNOT(더블워드 반전)명령이 있습니다.
5. NEG(2 의 보수, 마이너스, Negative 등) 명령은 워드(더블워드)값을 마이너스 처리한 것과 같으며, 각 비트를 반전시킨 후 1 을 덧셈한 것과 동일합니다.
6. 마이너스 명령은 NEG(워드 마이너스) 및 DNEG(더블워드 마이너스)명령이 있습니다.
7. D 영역은 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있습니다.
8. 워드명령은 16 비트(65,535) 이내, 더블워드명령은 32 비트(4,294,967,295) 범위 내에서 값을 처리합니다.
9. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제



워드영역에 데이터를 저장하기 위해 M000.00 접점을 ON 하면, W100 에 $0123(10 진수로 291)을 저장하고, W200 에는 $8123(10 진수로 33059)(부호포함 -32477)이 저장됩니다.(WinGPC 의 레지스터 모니터로 확인 가능)
M000.01 접점이 ON 되면 ABS(절대치)명령에 의해 W100, W200 의 값을 절대값으로 처리합니다.
즉, W100 의 $0123(0000 0001 0010 0011)(291)은 양수(+)이므로 절대치 변환은 동일한 값이 유지 됩니다.
W200 의 $8123(1000 0001 0010 0011)(-32477)은 음수값으로 절대값은 양수(+)값인 32477(0111 1110 1101 1101)($7EDD)이 됩니다. 즉, 2 의보수(비트를 반전시킨 후 1 을 더한 값)이다

M0.2 에서 WNOT 명령은 비트를 반전시킨 값이 되고, $8123 이 $7EDC(0111 1110 1101 1100)으로 변한다.
M0.3 의 NEG 명령은 양은 음으로,음은 양으로 한다. $0123(291)은 -291(1111 1110 1101 1101)이 된다.

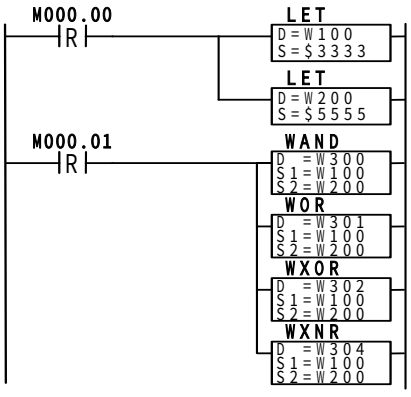
- 논리 연산 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
WAND	4	WAND D = S1= S2=	논리곱 (Word AND : 워드 연산) S1 과 S2 의 논리곱을 D 에 저장함. S1 0 0 1 1 S2 0 1 0 1 D **0 0 0 1**	전기종
DAND	5	DAND D = S1= S2=	논리곱 (D-Word AND : 더블 워드) S1 과 S2 의 논리곱을 D 에 저장함. S1 0 0 1 1 S2 0 1 0 1 D **0 0 0 1**	전기종
WOR	4	WOR D = S1= S2=	논리합 (Word OR : 워드 연산) S1 과 S2 의 논리합을 D 에 저장함. S1 0 0 1 1 S2 0 1 0 1 D **0 1 1 1**	전기종
DOR	5	DOR D = S1= S2=	논리합 (D-Word OR : 더블 워드) S1 과 S2 의 논리합을 D 에 저장함. S1 0 0 1 1 S2 0 1 0 1 D **0 1 1 1**	전기종
WXOR	4	WXOR D = S1= S2=	배타적 논리합 (Word Exclusive OR) S1 과 S2 의 배타적 논리합을 D 에 저장함. S1 0 0 1 1 S2 0 1 0 1 D **0 1 1 0**	전기종
DXOR	5	DXOR D = S1= S2=	배타적 논리합 (Exclusive OR : 더블 워드) S1 과 S2 의 배타적 논리합을 D 에 저장함. S1 0 0 1 1 S2 0 1 0 1 D **0 1 1 0**	전기종
WXNR	4	WXNR D = S1= S2=	동등회로 (Word Exclusive OR NOT) S1, S2 의 동등논리를 D 에 저장함. (배타적 논리합의 반전 값) S1 0 0 1 1 S2 0 1 0 1 D **1 0 0 1**	전기종
DXNR	5	DXNR D = S1= S2=	동등회로 (Exclusive OR Not : 더블워드) S1, S2 의 동등논리를 D 에 저장함. (배타적 논리합의 반전 값) S1 0 0 1 1 S2 0 1 0 1 D **1 0 0 1**	전기종

도움말

1. WAND(DAND)의 논리곱 연산은 S1 과 S2 의 각 비트번지를 논리적으로 곱셈(S1 의 각 비트와 S2 의 비트가 모두 1 일 때 결과값이 1 이 됨)하여 D 의 해당 비트번지에 저장합니다.
2. WOR(DOR)명령의 논리합 연산은 S1, S2 의 비트를 논리합(S1,S1 중 1 개라도 1 이면 결과가 1) 연산하여 D 에 저장합니다.
3. WXOR(DXOR)명령의 배타적논리합 연산은 S1,S2 의 비트중 1 개만 1 일때 결과가 1 로 연산하여 D 에 저장합니다.
4. WXNR(DXNR)명령의 배타적 논리합의 반전(동등회로) 연산은 S1, S2 의 비트가 같을때 결과를 1 로 연산하여 D 에 저장합니다.
5. S1, S2 영역은 숫자 또는 워드(더블워드)번지 (R,L,M,K,W,SV,PV,SR 등)를 지정할 수 있으며, 저장영역인 D 번지는 워드(더블워드) 번지만 지정할 수 있습니다.

사용예제

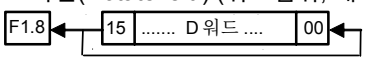
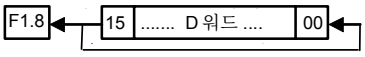
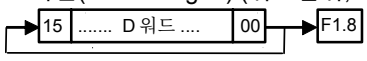
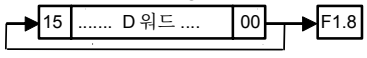


워드영역에 데이터를 저장하기 위해 M000.00 접점을 ON 하면, W100 에 $3333(0011 0011 0011 0011)을 저장하고, W200 에는 $5555(0101 0101 0101 0101)이 저장됩니다.

M000.01 접점이 ON 되면 논리연산 명령이 한번 작동하며, 각 명령의 결과가 D 에 저장됩니다.

즉, WAND 명령의 W300 은 $1111(0001 0001 0001 0001)이고 WOR 명령의 W301 에는 $7777(0111 0111 0111 0111), WXOR 명령의 W302 에는 $6666(0110 0110 0110 0110), WXNR 명령의 W304 에는 $9999(1001 1001 1001 1001)가 저장 됩니다.

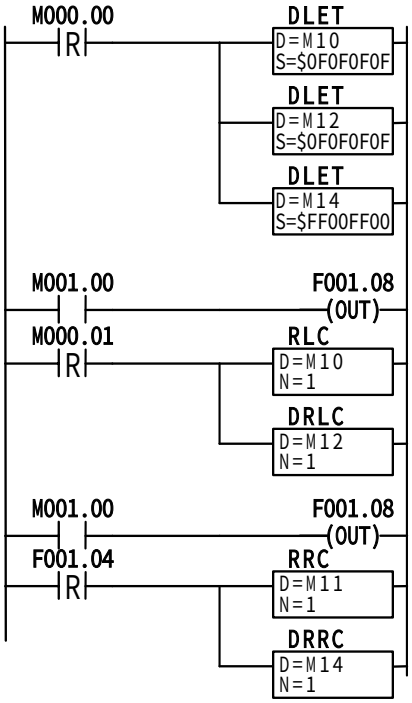
- 회전 명령어(좌/우로 회전)

니모닉명령	워드수	래더 표시	설 명	적용기종
RLC	3	RLC D= N=	좌로 회전(Rotate left) (워드범위, 캐리와 무관) 	전기종
DRLC	4	DRLC D= N=	좌로 회전(Rotate left) (더블워드, 캐리와 무관) 	전기종
RRC	3	RRC D= N=	우로 회전(Rotate Right) (워드범위, 캐리와 무관) 	전기종
DRRC	4	RRC D= N=	우로 회전(Rotate Right) (더블워드, 캐리와 무관) 	전기종

도움말

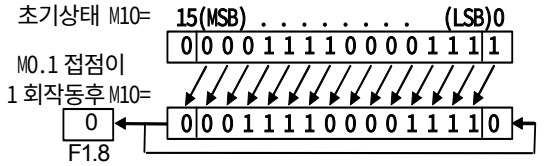
1. RLC(DRLC)는 지정된 D 번지의 각 비트를 N 비트씩 좌측으로(하위비트에서 상위로) 이동하며, 상위비트값(MSB)이 캐리(F1.8)비트에 저장됩니다.
2. RRC(DRRC)는 지정된 D 번지의 각 비트를 N 비트씩 우측으로(상위비트에서 하위로) 이동하며, 하위비트값(LSB)이 캐리(F1.8)비트에 저장됩니다.
3. D 영역은 워드(더블워드)번지 (R,L,M,K,W,SR 등)를 지정할 수 있으며, N 은 0~15(워드명령) 또는 0~31(더블워드 명령)까지 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



워드영역에 데이터를 저장하기 위해 M000.00 접점을 ON 하면, M10 에 $0F0F0F0F 을 저장합니다. 이것은 M10 워드에 $0F0F 를 저장하고 M11 워드에 $0F0F 를 동시에 저장한다. 또 M12 와 M13 워드에 $0F0F 를 저장하고, M14 와 M15 에는 $FF00 를 저장하게 만든다.

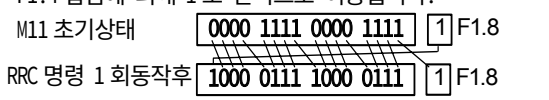
F1.8 의 캐리비트 출력은 접점에 반영여부를 판단한다.



M12 초기상태 0000 1111 0000 1111 0000 1111 0000 1111

DRLC 동작후 0001 1110 0001 1110 0001 1110 0001 1110

F1.4 접점에 의해 1 초 간격으로 이동됩니다.



M14 초기상태 1111 1111 0000 0000 1111 1111 0000 0000

DRRC 동작후 0111 1111 1000 0000 0111 1111 1000 0000

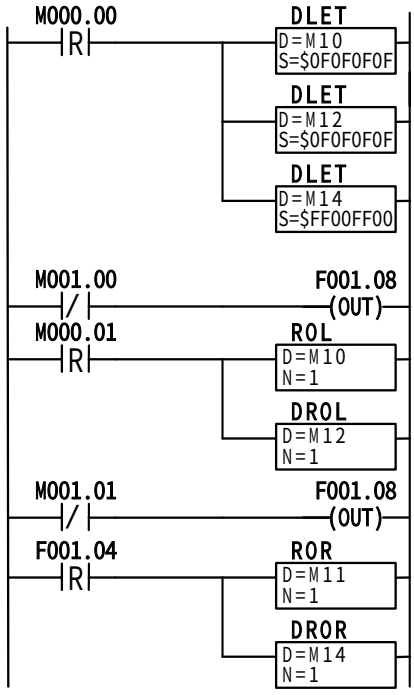
- 회전 명령어(캐리포함 좌/우로 회전)

니모닉명령	워드수	래더 표시	설 명	적용기종
ROL	3	ROL D= N=	좌로 회전(Rotate left) (워드범위, 캐리 포함) F1.8 ◀ 15 ..... D 워드 .... 00 ◀	전기종
DROL	4	DROL D= N=	좌로 회전(Rotate left) (더블워드, 캐리 포함) F1.8 ◀ 15 ..... D 워드 .... 00 ◀	전기종
ROR	3	ROR D= N=	우로 회전(Rotate Right) (워드범위, 캐리 포함) 15 ..... D 워드 .... 00 ▶ F1.8	전기종
DROR	4	DROR D= N=	우로 회전(Rotate Right) (더블워드, 캐리 포함) 15 ..... D 워드 .... 00 ▶ F1.8	전기종

도움말

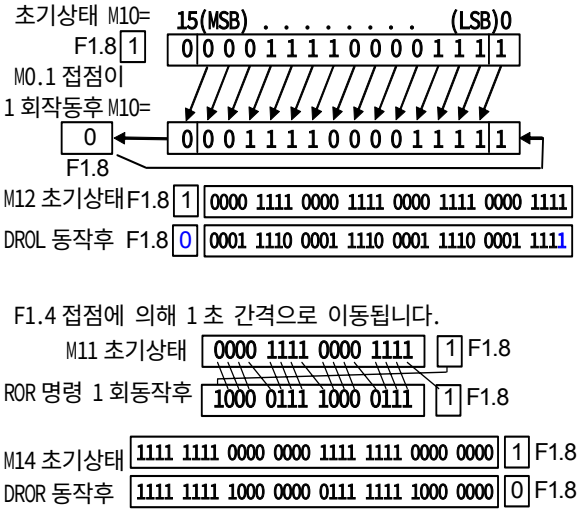
1. ROL(DROL)는 지정된 D 번지의 각 비트를 N 비트씩 좌측으로(하위비트에서 상위로) 이동하며, 상위비트값(MSB)은 캐리(F1.8)비트로 이동되며, 캐리비트값이 하위(LSB)에 저장됩니다.
2. ROR(DROR)는 지정된 D 번지의 각 비트를 N 비트씩 우측으로(상위비트에서 하위로) 이동하며, 하위비트값(LSB)은 캐리(F1.8)비트로 이동되며, 캐리비트값이 상위(MSB)에 저장됩니다.
3. D 영역은 워드(더블워드)번지 (R,L,M,K,W,SR 등)를 지정할 수 있으며, N 은 0~15(워드명령) 또는 0~31(더블워드 명령)까지 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



M0.0 접점을 ON 하면, M10, M11 에 $0F0F 를 저장합니다.
또 M12 와 M13 워드에도 $0F0F 를 저장하고,
M14 와 M15 에는 $FF00 를 저장하게 만든다.

F1.8 의 캐리 출력은 접점에 반영여부를 판단하기 위함이며, 접점이 ON 상태에서 변경되는 값을 확인하자.



F1.4 접점에 의해 1 초 간격으로 이동됩니다.

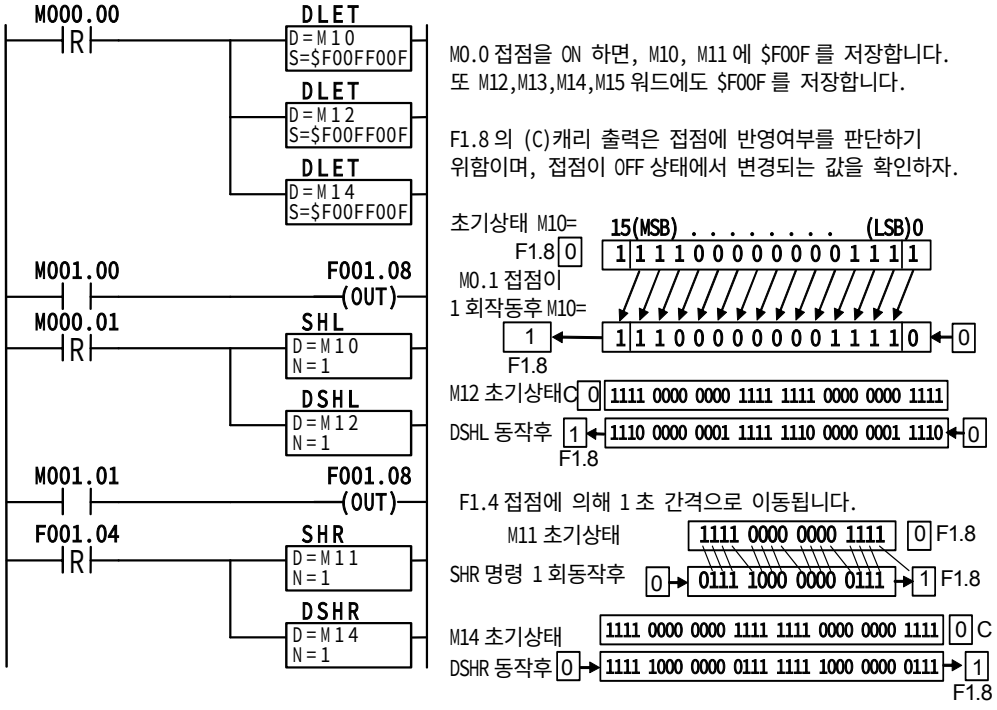
- 회전 명령어(좌/우로 시프트)

니모닉명령	워드수	래더 표시	설 명	적용기종
SHL	3	SHL D= N=	좌로 이동(Shift left) (워드범위, 0 입력) F1.8 ← 15 ..... D 워드 .... 00 ← 0	전기종
DSHL	4	DSHL D= N=	좌로 이동 (Shift left) (더블워드, 0 입력) F1.8 ← 15 ..... D 워드 .... 00 ← 0	전기종
SHR	3	SHR D= N=	우로 이동 (Shift Right) (워드범위, 0 입력) 0 → 15 ..... D 워드 .... 00 → F1.8	전기종
DSHR	4	DSHR D= N=	우로 이동 (Shift Right) (더블워드, 0 입력) 0 → 15 ..... D 워드 .... 00 → F1.8	전기종

도움말

1. SHL(DSHL)는 지정된 D 번지의 각 비트를 N 비트씩 좌측으로(하위비트에서 상위로) 이동하며, 상위비트값(MSB)은 캐리(F1.8)비트로 이동되며, 0 값이 하위(LSB)에 저장됩니다.
2. SHR(DSHR)는 지정된 D 번지의 각 비트를 N 비트씩 우측으로(상위비트에서 하위로) 이동하며, 하위비트값(LSB)은 캐리(F1.8)비트로 이동되며, 0 값이 상위(MSB)에 저장됩니다.
3. D 영역은 워드(더블워드)번지 (R,L,M,K,W,SR 등)를 지정할 수 있으며, N 은 0~15(워드명령) 또는 0~31(더블워드 명령)까지 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



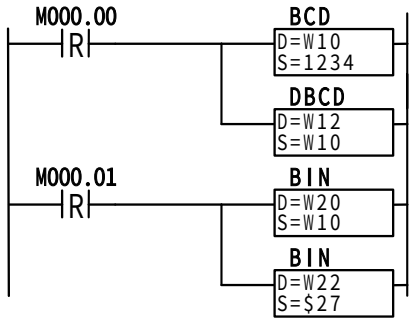
- 워드변환 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
BCD	3	BCD D= S=	S 의 2 진수값을 BCD 로 변환하여 D 에 저장(워드) 예) S= 00063 (10 진수) → D= $0063(BCD) 0000 0000 0011 1111 → 0000 0000 0110 0011	전기종
DBCD	3	DBCD D= S=	S 의 2 진수값을 BCD 로 변환하여 D 에 저장(더블워드)	전기종
BIN	3	BIN D= S=	S 의 BCD 값을 2 진수로 변환하여 D 에 저장(워드) 예) S= $0039 (BCD) → D= 00039(10 진수, Bianry) 0000 0000 0011 1001 → 0000 0000 0010 0111	전기종
DBIN	3	DBIN D= S=	S 의 BCD 값을 2 진수로 변환하여 D 에 저장(더블워드)	전기종

도움말

1. BCD(DBCD)는 S 번지의 Binary 값을 BCD 로 변환하여 D 영역에 저장하며, S 의 값은 BCD 단위의 최대치 범위내에 있어야 한다.
2. BIN(DBIN)명령은 S 번지의 BCD 값을 binary 로 변환하여 D 영역에 저장하며, S 의 값은 BCD 의 최대치 범위내에 있으면 된다.
3. 최대치는 워드명령(BCD, BIN)에서 0 ~ 9999(10 진수) 까지(16 진수로 $270F)이며, 더블워드(DBCD,DBIN)에서는 0 ~ 99999999 까지 (16 진수로 $05F5E0FF) 범위 내에 존재해야 합니다.
4. D 영역은 워드(더블워드)번지 (R,L,M,K,W,SR 등)를 지정할 수 있으며, S 는 숫자 또는 워드/더블워드 번지를 지정할 수 있습니다.
5. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



M0.0 접점을 ON 하면, 10 진수의 1234 값이 BCD 로 변환되어 W10 에 저장됩니다.

10 진수 1234	0000 0100 1101 0010	
BCD 변환값 W10 ($1234)	0001 0010 0011 0100	(=4660)
BCD 변환값 W12 ($4660)	0100 0110 0110 0000	(=18016)
BIN 변환값 W20 (1234)	0000 0100 1101 0010	
BCD $0027	0000 0000 0010 0111	(=39)
BIN 변환값 W22 (00027)	0000 0000 0001 1011	(=$001B)

주) BCD 란?

BCD 는 ' Binary Coded Decimal' 의 약자로 ' 2 진 부호화 10 진수' 라고 부르며, 4 개의 비트를 이용하여 10 진수를 표현한 코드방식을 말한다.
즉, 10 진법에서 각 자리수의 10 진 숫자를 2 진 숫자로 나타내는 방식이다.
예로 10 진수 23(0001 0111)을 BCD 로 변환하면 0010 0011 이며, 16 진수 표현은 $23 이 된다.
10 진수 57 은 BCD 로 0101 0111 로 표현되고, 16 진수로 $57 이며, 10 진수로는 87 이된다.

WinGPC 를 이용하여 프로그램 편집상태에서 도구(T)/숫자변환(N) 메뉴를 열면 숫자를 10 진수/16 진수/2 진수로 변환하여 확인할 수 있습니다.

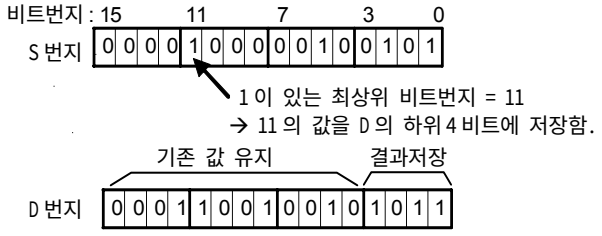
워드의 BCD 값은 0 ~ $9999(10 진수로는 39,321 까지 처리되며, 더블워드 BCD 값은 0~$99999999 (10 진수로 2,576,980,377)까지 값을 처리할 수 있습니다.

- 워드변환 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
ENCO	3	ENCO D= S=	S 워드의 상위비트값을 8421 Encode 변환후 D 에 저장	전기종
DECO	3	DECO D= S=	S 워드의 값을 8421 Decode 변환 후 D 에 저장	전기종

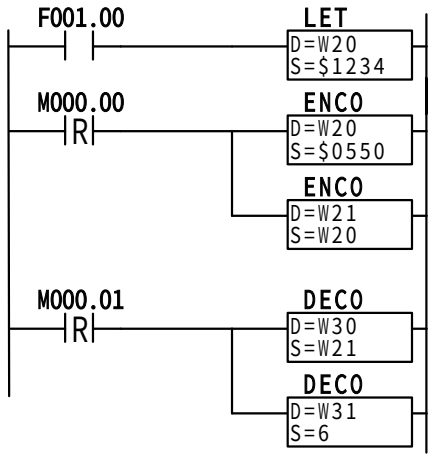
도움말

1. ENCO 명령은 S 번지에서 1 이 있는 최상위비트의 비트번지 값을 엔코드(2^n의 n 값)변환하여 변환된값을 D 영역의 하위 8 비트에 저장하며, D 의 상위 바이트의 값은 유지되고, 1 이 2 개 이상 존재하면 최상위에 있는 비트값만 처리됩니다.
   (단, SPC 시리즈는 엔코드 변환을 2^n의 n+1 값으로 변환하여 D 에 저장합니다)



2. DECO 명령은 S 번지의 하위 4 비트 값을 디코드(2^n 값)로 변환하여 D 영역에 저장합니다.
3. D 영역은 워드번지 (R,L,M,K,W,SR 등)를 지정할 수 있으며, S 영역은 워드번지 또는 숫자를 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제

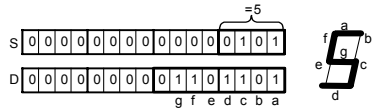


초기조건으로 W20 워드에 $1234 값을 대입시킨다.
M0.0 접점을 ON 하면, 16 진수의 $0550 값을 Encode 로 변환하여 W10 에 저장합니다.

초기값 W20 ($1234) 0001 0010 0011 0100
16 진수$0550 0000 0101 0101 0000
1 이 있는 최상위 비트번지=10,
즉, 2^{10}의 10 을 W20 워드의 하위 바이트에 저장합니다.

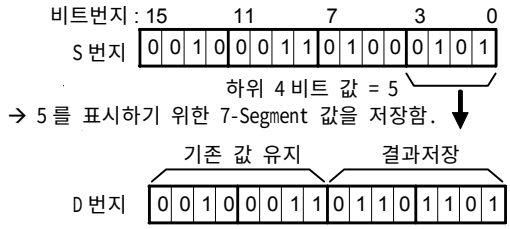
ENCO 변환값 W20 ($120A) 0001 0010 0000 1010
상위 비트=12 에서 2^{12}의 12 를 저장
ENCO 변환값 W21 ($000C) 0000 0000 0000 1100
Decode 변환은 S 의 값을 2^n처리함
DECO 변환값 W30 ($1000) 0001 0000 0000 0000
6 의 DECO 변환값 W31 ($0040) 0000 0000 0100 0000

- 워드변환 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
SEG	3	SEG D= S=	S 의 하위 4 비트 값을 7-Segment 로 변환하여 D 의 하위 바이트에 저장 	전기종

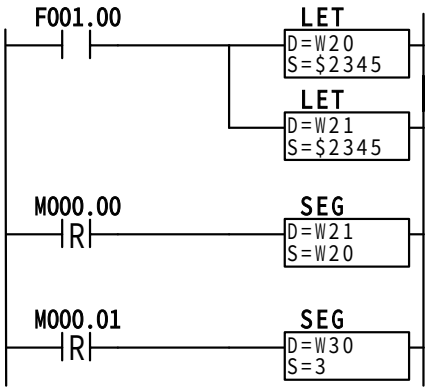
도움말

1. SEG 명령은 S 번지에서 하위 4 비트의 값을 7-세그먼트로 변환하여 D 워드의 하위 8 비트에 저장하는 기능입니다.



2. SEG 명령 D 번지의 상위 바이트 값은 변경되지 않고 기존 값을 유지합니다.
3. D 영역은 워드번지 (R,L,M,K,W 등)를 지정할 수 있으며, S 영역은 워드번지 또는 숫자를 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



초기조건으로 W20 과 W21 워드에 $2345 값을 대입시킨다.

M0.0 접점을 ON 하면, S 번지의 하위 4 비트의 Data 5 를 7-Segment 로 변환하여 D 의 하위 8 비트(1 바이트)에 저장합니다.

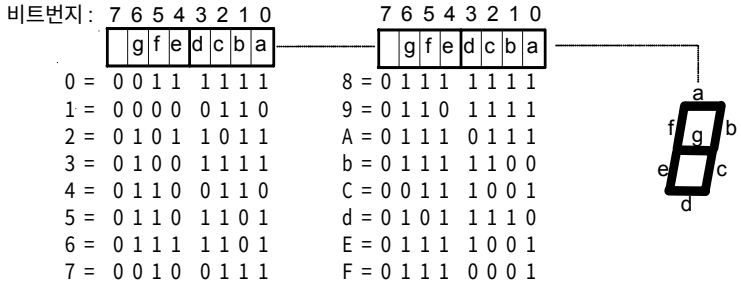
초기값 S = W20 ($2345) **0010 0011 0100 0101**

연산결과 D= W21($236D) **0010 0011 0110 1101**

M0.1 접점을 ON 하면 S 데이터의 3 을 변환하여 D 영역에 저장합니다.

연산결과 D=W30 ($004F) **0000 0000 0100 1111**

주) 7-Segment 표시 비트 예



- 워드변환 명령어

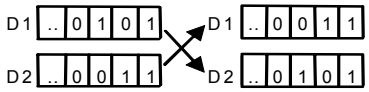
니모닉명령	워드수	래더 표시	설 명	적용기종
XCHG	3	XCHG D1= D2=	D1 과 D2 의 값을 서로 교환 (워드)	주 1)
DXCHG	3	DXCHG D1= D2=	D1 과 D2 의 값을 서로 교환 (더블워드)	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	○	○	○	○	○	○

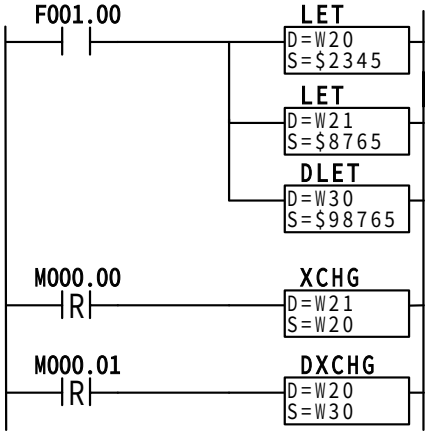
도움말

1. XCHG(DXCHG)명령은 D1 번지와 D2 번지의 값을 서로 교환하는 기능입니다.

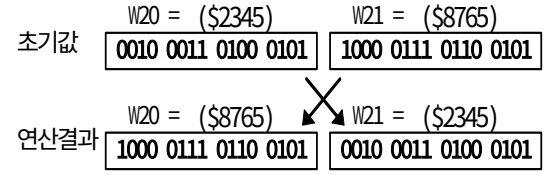


2. D1, D2 영역은 모두 워드번지(R,L,M,K,W 등)만 지정할 수 있습니다.
3. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

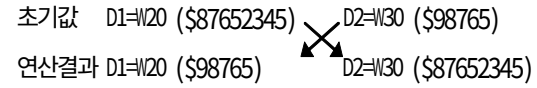
사용예제



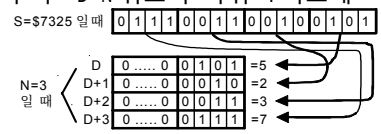
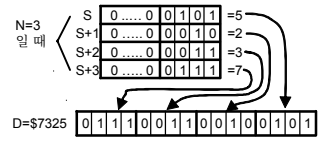
초기조건으로 W20 워드에 $2345 값을 대입하고,
W21 워드에 $8765 값을 대입시킨다.
M0.0 접점을 ON 하면, D1 번지와 D2 번지의 Data 를
서로 교환합니다.



M0.1 접점을 ON 하면 더블워드의 D1 과 D2 값을 교환합니다.



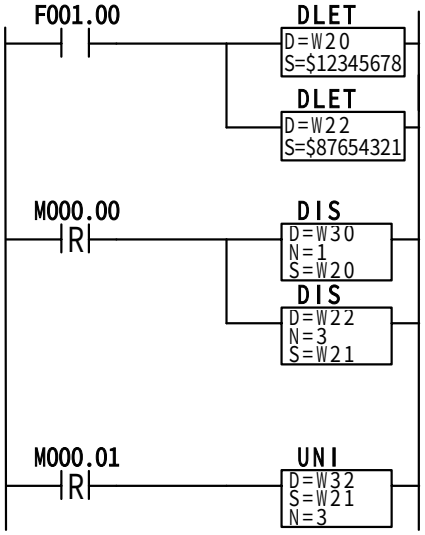
- 워드변환 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
DIS	4	DIS D= N= S=	S 를 4 비트씩 N+1 개로 분리하여 D 부터 D+N 워드의 하위 4 비트에 저장 	전기종
UNI	4	UNI D= S= N=	S 부터 S+N 워드의 하위 4 비트를 D 에 하위비트부터 저장(N= 0~3) 	전기종

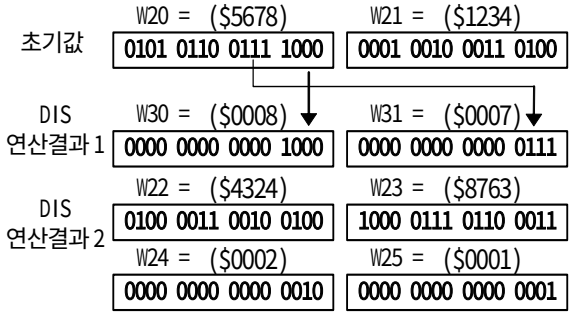
도움말

1. DIS 명령은 4 비트(1 자리수) 단위로 데이터를 분리하는 기능입니다.
2. UNI 명령은 분리된 4 비트씩을 1 개 워드로 결합하는 기능입니다.
3. D, S 영역은 워드번지(R,L,M,K,W 등)를 지정할 수 있고, N 은 개수(숫자)로 0~3 까지 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

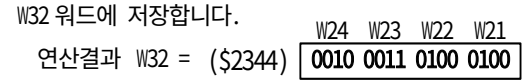
사용예제



초기조건으로 W20 워드에 $12345678 값을 대입하고,
W22 워드에 $87654321 값을 대입시킨다.
M0.0 접점을 ON 하면, S 의 W20 번 워드값의 하위 2 자리 수를 분리하여 D 워드(W30), D+1 워드(W31)에 저장합니다.



M0.1 접점을 ON 하면 W21~W24 의 하위 4 비트씩 값을 결합하여 W32 워드에 저장합니다.



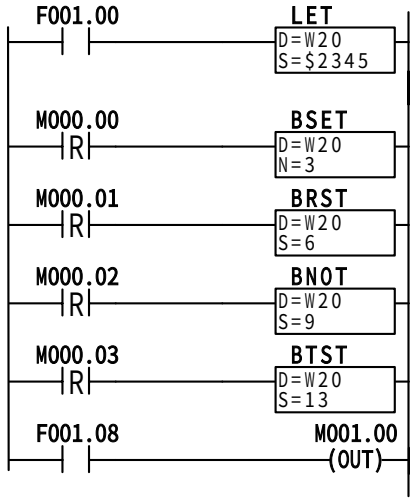
- 비트변환 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
BSET	3	BSET D= N=	D 워드의 N 번째 비트 값을 1 로 SET 시킴 D ..... 0 0 1 0 0 1 0 0 N=5 일때 ——↑ (N=0~15)	전기종
BRST	3	BRST D= N=	D 워드의 N 번째 비트 값을 0 으로 저장(리셋시킴) D ..... 0 1 0 1 0 1 0 0 N=3 일때 ——↑ 0	전기종
BNOT	3	BNOT D= N=	D 워드의 N 번째 비트 값을 반전시킴. (0 → 1, 1→ 0) D ..... 0 1 1 1 0 1 0 0 N=4 일 때 ↓ D 0 1 1 0 0 1 0 0	
BTST	3	BTST D= N=	D 워드의 N 번째 비트 값을 F1.8(캐리비트)에 저장 D ..... 0 1 1 1 0 1 0 0 N=6 일 때 ——→ F1.8	

도움말

1. 비트변환 명령은 워드번지에서 각각의 비트값을 제어하기 위한 기능입니다.
2. N 으로 지정된 비트 이외의 것은 기존의 값을 유지 합니다.
3. D 영역은 워드번지 (R,L,M,K,W, SR 등)를 지정할 수 있으며, N 영역은 숫자 0~15 까지를 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



초기조건으로 W20 워드에 $2345 값을 대입시킨다.
초기값 D = W20 ($2345) 0010 0011 0100 0101

M0.0 접점을 ON 하면, W20 워드의 3 번 비트를 1 로 Set 함.
연산결과 D=W20 ($234D) 0010 0011 0100 1101
3 번 비트 Set

M0.1 접점을 ON 하면 W20 워드의 6 번비트를 Reset 합니다.
연산결과 D=W20 ($230D) 0010 0011 0000 1101
6 번 비트 Reset

M0.2 접점을 ON 하면 W20 워드의 9 번비트를 반전합니다.
연산결과 D=W20 ($210D) 0010 0001 0000 1101
9 번 비트 반전

M0.3 접점을 ON 하면 W20 워드의 13 번비트의 값을 F1.8 에 저장합니다. 동시에 M1.0 출력도 ON 됩니다.
연산결과 D=W20 ($210D) 0010 0001 0000 1101
13 번비트 저장(F1.8)

- 비트변환 명령어

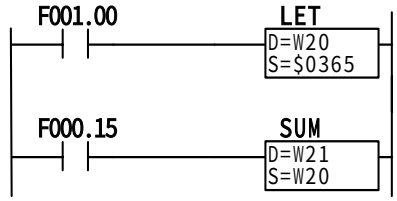
니모닉명령	워드수	래더 표시	설 명	적용기종
SUM	3	SUM D= S=	S 워드에서 1 인 비트의 개수를 D 에 저장 S 0001101001101001 1 의개수=7 D 0000000000000111 D=7	전기종

도움말

1. SUM 명령은 S 워드에서 ON(=1)으로 Set 된 비트의 개수를 D 에 저장하는 기능입니다.
2. D 와 S 는 모두 워드번지 (R,L,M,K,W, SR 등)를 지정할 수 있습니다.
3. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제

1. 주로 접점의 상태를 점검하거나 데이터의 변화를 감시하기 위한 용도로 많이 사용합니다.



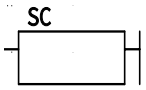
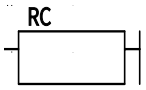
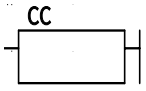
초기조건으로 W20 워드에 $0365 값을 대입시킨다.

초기값 S = W20 ($0365) **0000 0011 0110 0101**

항상 ON 되어 있는 F0.15 접점으로 S 워드의 값이 변경되면 즉시 D 영역에 Data 를 저장합니다.

연산결과 D=W21 ($0006) **0000 0000 0000 0110**

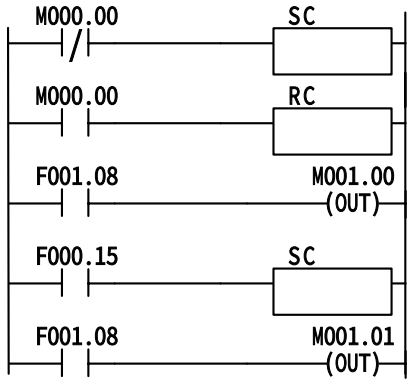
- 비트변환 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
SC	1		캐리 비트(F1.8)를 1 로 SET 함 1 → F1.8	전기종
RC	1		캐리 비트(F1.8)를 0 으로 RESET 함 0 → F1.8	전기종
CC	1		캐리 비트(F1.8)값을 반전 시킴. F1.8 (1 → 0) F1.8 (0 → 1)	전기종

도움말

1. 캐리비트(F1.8)는 내부 특수 프래그로서 프로그램 연산중에 값의 범위를 넘거나 부호의 표시, 회전이나 이동시에 Set 또는 Reset 으로 변경되기도 하므로 현재의 캐리값을 제어하기 위해 사용합니다.
2. 캐리비트(F1.8)를 제어하기 위한 명령으로 SC, RC, CC 등이 있습니다.
3. 캐리비트의 조작은 계산의 조건과 결과를 바꿀 수도 있으므로 사용시 주의 해야 합니다.
4. 캐리명령의 오퍼랜드 변수는 없으며, 항상 캐리비트(F1.8)접점을 제어하게 됩니다.
5. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



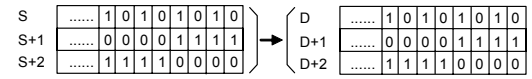
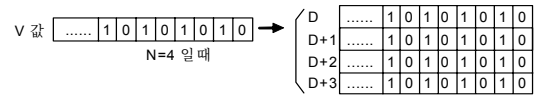
M0.0 접점의 조건에 따라 캐리비트(F1.8)를 Set 또는 Reset 으로 지정한다. 즉, M0.0 이 OFF 상태에서는 F1.8 을 Set 으로 지정하고, M0.0 이 ON 상태에서는 F1.8 이 Reset 으로 된다.

F1.8 의 상태에 따라 출력 M1.0 출력이 ON/OFF 된다.

항상 ON 되는 F0.15 접점으로 CC 명령이 작동되고, 캐리가 반전됩니다.

F1.8 접점에 의해 M1.1 출력은 항상 M1.0 과 반대로 된다.

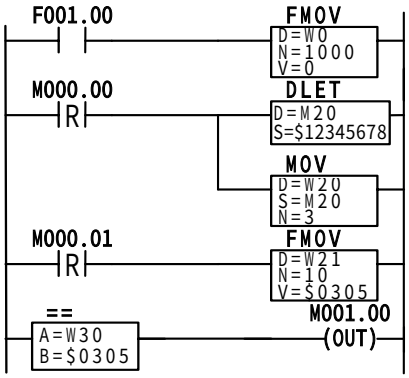
- 전송 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
MOV	4	MOV D= S= N=	S 워드부터 N 개 워드(S 포함)를 D 워드부터 N 개로 값을 복사함. 	전기종
FMOV	4	FMOV D= N= V=	V 값을 D 부터 N 개 워드로 복사함 	전기종

도움말

1. 워드블럭 단위로 데이터를 전송(복사)하고자 할 경우 MOV 명령을 사용하며, 어떤 동일한 데이터를 지정한 레지스터 블록에 전송하고자 할 때에는 FMOV 명령을 사용합니다.
2. MOV 명령에서 D 와 S 는 모두 워드번지 (R,L,M,K,W, SR 등)를 지정할 수 있으며, N 값은 동일한 종류의 레지스터 범위내의 숫자를 지정할 수 있습니다.
3. FMOV 명령에서 D 는 워드번지 (R,L,M,K,W,SR 등)를 지정할 수 있으며, N 값은 동일한 레지스터 범위의 숫자를 지정할 수 있고, V 는 16 비트 이내의 값을 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.
5.

사용예제



F1.0 접점으로 첫 Scan 에서 W0 부터 W999 까지 1000 개워드를 0 으로 초기화 시킨다.

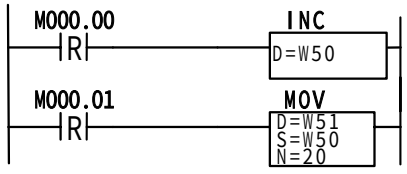
M0.0 접점이 ON 되면 M20 워드에 $5678 이 저장되고,
M21 워드에 $1234 값을 저장합니다.
동시에 M20~M22 워드 값을 W20~W22 워드까지 저장(복사)합니다.

M0.1 접점이 ON 되면 W21 워드부터 W30 워드까지 10 개를 $0305 값으로 저장합니다.

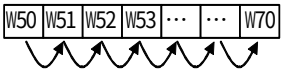
W30 워드값이 $0305 값과 동일하면 출력 M1.0 이 ON 됩니다.

응용예제

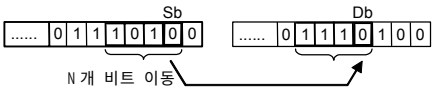
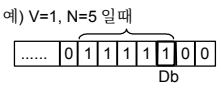
1. 과거의 Data 를 워드단위로 시프트 시키면서 저장하는 예제 입니다.
   즉, 한번씩 작동할 때 마다 W50 은 W51 로 보내고, W51 은 W52 로 보내는 등의 기능입니다.



M0.0 접점이 ON 될때마다 INC 명령으로 W50 은 1 씩 증가하며,
M0.1 접점이 ON 될때마다 W50 번부터 W69 까지 20 개 워드를 W51~W70 으로 데이터를 전송합니다.
즉, W50 의 값은 W51 에 저장되고,W51 은 W52 에,W52 는 W53 에 … W69 는 W70 에 저장되며, W70 은 소멸됩니다.



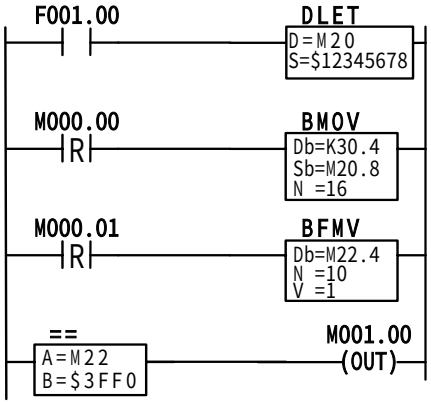
- 전송 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
BMOV	4	BMOV Db= Sb= N =	Sb 비트로부터 N 개 비트를 Db 비트부터 N 개로 값을 복사함. (Sb, Db 는 비트번지만 지정가능) 	전기종
BFMV	4	BFMV Db= N = V =	V 값을 Db 부터 N 개 비트로 복사함. (V=0 또는 1)(N=1…256, 영역내) (Db 는 비트번지) 	전기종

도움말

1. 비트블럭 단위로 데이터를 전송(복사)하고자 할 경우 BMOV 명령을 사용하며, 어떤 동일한 데이터(0 또는 1)를 지정한 비트블럭에 전송하고자 할 때에는 BFMV 명령을 사용합니다.
2. BMOV 명령에서 Db 와 Sb 는 모두 비트번지 (R,L,M,K 등)를 지정할 수 있으며, N 값은 1~ 256 이내의 값(비트수)을 지정할 수 있습니다.
3. BFMV 명령에서 Db 는 비트번지 (R,L,M,K 등)를 지정할 수 있으며, N 은 1~ 256 이내의 비트수, V 는 0 또는 1 의 값을 지정할 수 있습니다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제



F1.0 접점으로 첫 Scan 에서 M20 워드에 $5678 을 저장하고, M21 워드에 $1234 값을 저장한다.

M0.0 접점이 ON 되면 M20.8 비트부터 16 개비트(M21.7 까지) 값을 K30.4 비트부터 16 개(K31.3 까지)로 저장합니다.

M0.1 접점이 ON 되면 M22.4 비트부터 10 개비트(M22.13 까지)를 1 로 저장합니다.

실제 M22 워드의 값이 $3FF0 (0011 1111 1111 0000)상태가 되면 출력 M1.0 접점이 ON 됩니다.

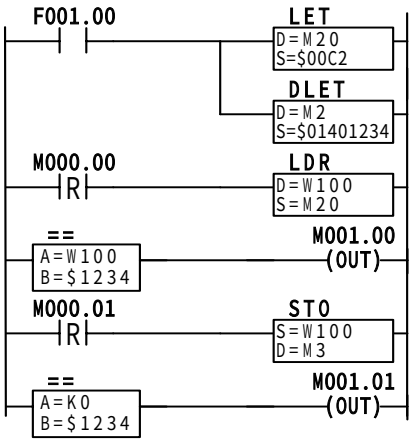
- 전송 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
LDR	3	LDR D= S=	간접지정 번지값의 데이터 불러오기 (S 워드의 값이 절대번지인 번지의 값을 D 에 저장) 	전기종
DLDR	3	DLDR D= S=		
STO	3	STO S= D=	간접지정 번지값에 S 데이터 저장 (D 워드값이 절대번지인 장소에 S 워드값을 저장) 	
DSTO	3	DSTO S= D=		

도움말

1. 간접지정 명령은 고정된 번지에 데이터를 저장하지 않고, 지정한 번지의 값에 따라 그 값이 지정한 번지(절대번지)에 데이터를 저장 및 이동시키는 명령을 말합니다.
2. 간접지정 명령에서는 절대번지를 사용하며, 절대번지는 CPU 의 메모리에 할당된 번지를 종류(R,M,K,W 등)에 상관없이 순서대로 지정된 번지를 의미하며, 주로 16 진수 단위로 구별이 됩니다. (본 매뉴얼의 ‘ 절대번지 지정 ’ 참조)
3. LDR, DLDR 명령은 S 번지의 값에 따라 저장하고자 하는 D 의 데이터 값이 달라지며, STO, DSTO 명령은 D 번지의 값에 따라 저장하고자 하는 번지가 달라지는 기능이다.
4. 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제



F1.0 접점으로 첫 Scan 에서 M20 워드에 $00C2 를 저장하고, M2 워드에 $1234, M3 워드에 $0140 값을 저장한다.

M0.0 접점이 ON 되어 LDR 명령의 기능을 분석해 보면
D 에 저장하는 값은 S 의 값에 따라 결정되며 S=$00C2 이다.
즉, 절대번지 $00C2 는 M0002 번지이므로 D 번지(W100)에는 M0002 워드의 값($1234)이 저장됩니다.
W100 번지에 $1234 값과 동일하면 출력 M1.0 이 ON 됩니다.

M0.1 접점에 의한 STO 명령은
저장하는 번지를 결정하는 D 워드값(M3)이 $0140 이며,
$0140 의 절대번지는 K0 워드에 해당됩니다.
즉, K0 워드에 S 값(W100=$1234)을 저장하는 명령입니다.
K0 의 값이 $1234 이면 출력 M1.1 접점이 ON 됩니다.

절대번지 대응표 (상세내용은 매뉴얼 참조)

레지스터	절대번지	레지스터	절대번지	레지스터	절대번지	레지스터	절대번지	레지스터	절대번지
R0000	$0000	L0000	$0080	M0000	$01C0	K0000	$0140	W0000	$0200
R0001	$0001	L0001	$0081	M0001	$01C1	K0001	$0141	W0001	$0201
R0002	$0002	L0002	$0082	M0002	$01C2	K0002	$0142	W0002	$0202
.....		.....		.....		.....		.....	
R0126	$007E	L0062	$00BE	M0126	$013E	K0126	$01BE	W2046	$09FE
R0127	$007F	L0063	$00BF	M0127	$013F	K0127	$01BF	W2047	$09FF

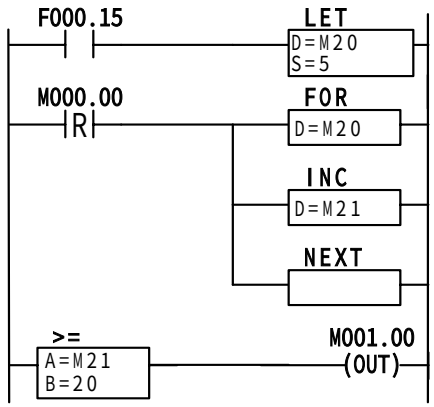
- 블록처리 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
FOR	3	FOR D=	D 값(워드)을 1 씩 감소시키면서 NEXT 명령까지 D 회 (D 가 0 일때까지) 반복수행(FOR Loop 시작)	전기종
DFOR	3	DFOR D=	D 값(더블워드)을 1 씩 감소시키면서 NEXT 명령까지 D 회 (D 가 0 일때까지) 반복수행(FOR Loop 시작)	전기종
NEXT	2	NEXT	FOR 명령어부터 반복 수행 종료(FOR Loop 끝)	전기종

도움말

1. FOR(DFOR) 명령은 D 워드의 값이 0 이 될 때 까지 반복해서 NEXT 까지 구간을 수행하는 기능이며, FOR ~ NEXT 명령 구간에서도 일반 명령어나 분기명령(JMP, CALL 등)을 사용할 수 있고, 실행 횟수(D 값)는 프로그램 수행중에도 변경이 가능합니다.
2. FOR 명령의 D 값이 0 이면 NEXT 이후의 명령으로 분기합니다.
3. FOR(DFOR) 명령과 NEXT 명령어는 항상 쌍으로 구성되어야 하며, NEXT 명령어는 오퍼랜드가 존재하지 않습니다.
4. NEXT 명령어의 입력조건이 OFF 상태에서는 FOR(DFOR)명령 루프가 한번만 작동하고, 입력조건이 ON 상태에서만 FOR(DFOR) ~ NEXT 명령 루프가 반복해서 수행 합니다.
5. D 영역은 워드번지 (R,L,M,K,W 등)를 지정할 수 있으며, FOR 명령은 워드범위, DFOR 명령은 더블워드 값을 지정할 수 있습니다.
6. FOR(DFOR) 명령을 사용해서 D 값을 크게 하면, 많은 루프 수행을 하기 때문에 스캔타임이 길어지므로 사용에 주의 바랍니다.
7. FOR, DFOR 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제



F0.15 접점은 운전중일 때 항상 ON 되는 접점이며, 매 Scan 마다 M20 워드에 5 의 값을 저장합니다.

M0.0 접점이 ON 될 때마다 FOR, NEXT 구간이 반복되며, 한번씩 수행될 때 마다 M20 워드가 1 씩 감소되어 0 이 되면 NEXT 다음 명령어로 넘어갑니다.
즉, M0.0 접점이 한번 ON 될때마다 M20 이 5 이므로 5 번 수행하며, INC 명령에 의해 M21 워드가 1 SCAN 에서 5 씩 증가하게 됩니다.

M21 워드값이 20 이상에서(4 번 이상 수행후) 출력 M1.0 이 ON 됩니다.

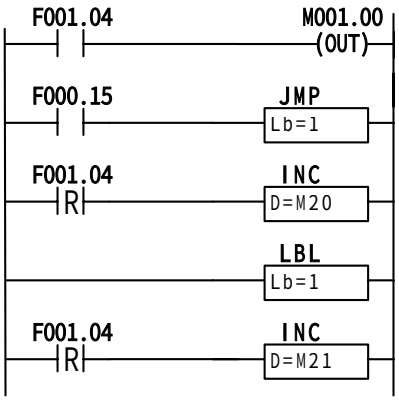
- 블록처리 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
JMP	3	JMP Lb=	LBL 명령의 Lb 번까지 점프(포인터에 의한 점프) Lb=0~63 (중복 사용 불가)	전기종
LBL	2	LBL Lb=	JMP 로 이동되는 위치(점프 포인터 지정) Lb=0~63 (중복 사용 불가)	전기종

도움말

1. JMP 명령은 프로그램 처리중에 회로의 일부를 수정하지 않고 강제로 점프하고자 할 때 사용하며, Lb 의 범위가 0 ~ 63 까지 64 개 까지 사용이 가능 합니다.
2. JMP 명령과 LBL 명령은 항상 쌍을 이루어야 하며, Lb 번호를 중복해서 사용할 수 없습니다.
3. 점프 시작명령(JMP) 보다 항상 점프위치(LBL)명령이 뒤에 위치해야 합니다.
4. JMP 명령어와 쌍을 이루는 LBL 명령어 사이에서도 JMP 명령어를 사용할 수 있습니다.
5. JMP 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.
6. LBL 명령은 항상 Root 에서 직접 연결되어야 하며, 입력조건 없이 사용해야 합니다.

사용예제

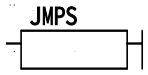
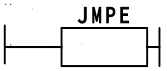


F1.4 접점은 1 초 간격으로 ON/OFF 를 반복하는 내부접점이며, 동시에 출력 M1.0 은 0.5 초 동안 ON, 0.5 초 동안 OFF 로 작동합니다. (작동상황 모니터링 용)
항상 ON 되는 F0.15 접점으로 JMP 명령이 작동되므로 내부의 F1.4 접점과 INC(M20 워드)명령은 작동하지 않습니다.

LBL 명령으로 Lb 1 번의 점프 쌍이 완료됩니다.

LBL 명령으로 점프가 완료되고 난 후, F1.4 접점은 작동하며, INC 명령의 M21 워드값은 1 초마다 1 씩 증가합니다.

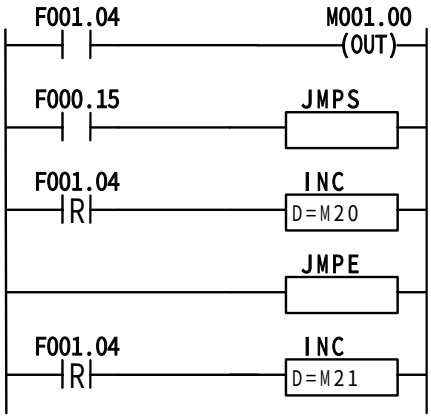
- 블록처리 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
JMPS	2		JMPE 명령어 까지 점프(점프 시작)	전기종
JMPE	1		점프 끝 (JMPS 와 쌍으로 구성) (JMPS 명령 이후에 있어야 함)	전기종

도움말

1. JMPS 명령은 프로그램 처리중에 회로의 일부를 수정하지 않고 강제로 점프하고자 할 때 사용하며, 오퍼랜드가 없이 JMPE 명령어 까지 점프 합니다.
2. JMPS 명령과 JMPE 명령은 항상 쌍을 이루어야 하며, JMPS 보다 LBL 명령이 항상 뒤에 위치해야 합니다.
3. JMPS 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.
4. JMPE 명령은 항상 Root 에서 직접 연결되어야 하며, 입력조건 없이 사용해야 합니다.

사용예제



F1.4 접점으로 출력 M1.0 은 0.5 초 동안 ON, 0.5 초 동안 OFF 로 작동합니다.

F0.15 접점으로 JMPS 명령이 작동되어 JMPE 명령까지 점프하므로 INC 의 M20 워드는 증가하지 않습니다.

JMPE 명령으로 점프가 완료되고, F1.4 접점은 작동하며, INC 명령의 M21 워드값은 1 초마다 1 씩 증가합니다.

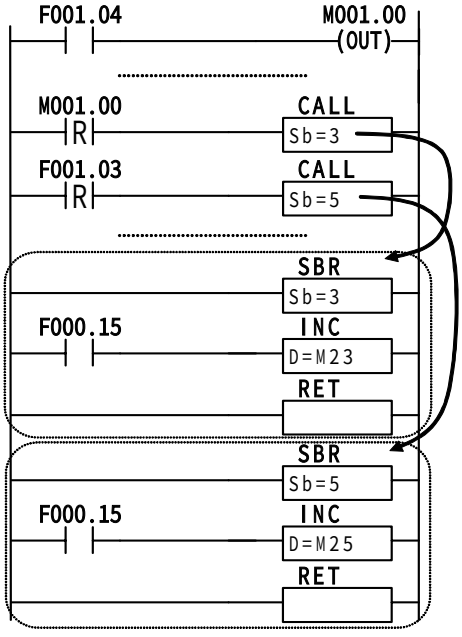
- 블록처리 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
CALL	3	CALL Sb=	서브루틴 호출 명령	전기종
SBR	2	SBR Sb=	서브루틴 시작 명령 Sb=0~63(64 개 사용, 중복사용 불가)	전기종
RET	1	RET	서브루틴 끝 (Subroutine Return)	전기종

도움말

1. CALL 명령은 서브루틴을 호출해서 실행하는 명령어로 서브루틴 번호(Sb 번호)의 모든 프로그램을 불러와서 처리하는 명령어입니다.
2. 하나의 서브루틴이 끝나는 RET 명령어를 수행한 후에는 CALL 명령 다음으로 복귀합니다.
3. CALL 및 SBR 명령의 Sb 는 0~63 까지 64 개의 서브루틴을 지정할 수 있으며, CALL 명령어는 SBR/RET 명령어 이전에 존재해야 합니다.
4. CALL 명령의 Sb 번호는 다중으로 사용이 가능하며, SBR 과 RET 명령어는 중복해서 사용할 수 없고, 항상 쌍을 이루어야 하며, SBR 보다 RET 명령이 항상 뒤에 위치해야 합니다.
5. CALL 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.
6. SBR 과 RET 명령은 항상 Root 에서 직접 연결되어야 하며, 입력조건 없이 사용해야 합니다.
7. SBR 과 RET 명령어로 구성되는 서브루틴 블록 이후에는 프로그램 수행을 하지 않으므로 시퀀스 처리용 프로그램은 항상 SBR 명령어 이전에 있어야 작동합니다.

사용예제



F1.4 접점으로 출력 M1.0 은 0.5 초 동안 ON, 0.5 초 동안 OFF 로 작동합니다.

M1.0 에 의한 CALL 3 의 동작은 1 초마다 서브루틴 3 번 프로그램을 수행합니다.
서브루틴에서는 F0.15 접점에 의해 INC 명령이 작동하고 동작할 때 마다 1 씩 증가하므로 M23 워드는 1 초마다 1 씩 증가합니다.

F1.3 은 0.2 초 클럭의 내부접점이며, 0.1 초동안 ON, 0.1 초동안 OFF 로 작동합니다.
즉, CALL 5 의 동작은 0.2 초마다 서브루틴 5 번 프로그램을 수행하며, 서브루틴 5 번의 INC 명령으로 M25 워드는 0.2 초마다 1 씩 증가합니다.

- 시스템 제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
INT	2	INT V=	정주기 스캔 시작명령(1 개만 사용가능) V=1~999 까지 지정가능, 제어비트= F0.11 정주기시간=(V+1)×0.01 초 (20msec 10sec)	주 1)
RETI	1	RETI	정주기 스캔 끝	주 1)

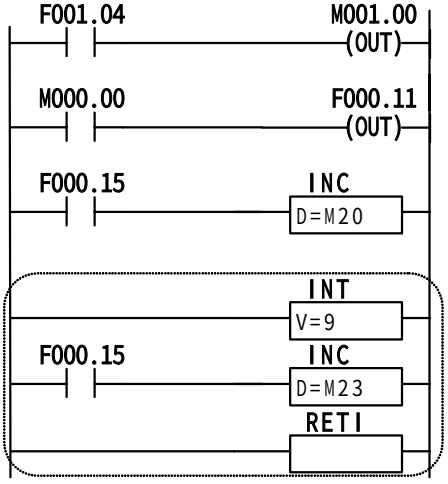
주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	○	○	○	○	○	○

도움말

1. 정주기 스캔 명령은 스캔타임과 상관없이 특정 프로그램 구간(INT 명령~RETI 명령까지)을 일정한 주기로 작동시키고자 할 경우에 사용하며, 0.02 초 ~ 10 초까지 지정할 수 있다.
2. 정주기 스캔의 시간설정은 10mSec 단위로 V 값+1 으로 정해지며, 수식으로는 정주기 시간 = (V+1)*0.01 초 이다. 즉, V 값이 3 이면, (3+1)*0.01=40mSec 이 된다.
3. 프로그램 내부에서 정주기 스캔 명령은 하나만 사용할 수 있습니다.
4. 정주기 명령의 동작은 내부접점 F0.11 으로 제어하며, 이 접점이 ON 상태에서만 작동합니다.
5. INT 와 RETI 명령은 항상 Root 에서 직접 연결되어야 하며, 입력조건 없이 사용해야 합니다.
6. 정주기 루틴을 수행하는 시간이 메인 프로그램의 스캔타임 보다 크면 프로그램이 정상적으로 작동하지 않으므로 정주기 루틴은 최소 스텝으로 작성해야 합니다.

사용예제



F1.4 접점으로 출력 M1.0 은 0.5 초 동안 ON, 0.5 초 동안 OFF 로 작동합니다.

M0.0 이 ON 상태에서 INT 명령이 작동하며, OFF 상태에서는 INT 명령부터 RETI 명령까지 동작하지 않습니다.

F0.15 접점에 의해 메인 프로그램이 동작하는 매 스캔마다 M20 워드 값이 1 씩 증가합니다.

정주기 영역은 V 값이 9 이므로 (9+1)*10mS=100mSec 단위로 작동하며, M23 워드는 100mSec 마다 1 씩 증가합니다.
단, M0.0 접점이 ON 상태에서 F0.11 이 ON 되어야 정주기 프로그램이 작동합니다.

- 시스템 제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
INPR	2	INPR Ch=	외부입력 리프레쉬(프로그램 수행중 입력신호 받아들임) Ch 는 외부 입력 워드번지	주 1)
OUTR	2	OUTR Ch=	외부출력 리프레쉬(프로그램 수행중 출력신호 보냄) Ch 는 외부출력 워드번지	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	○	○	○	○	○	○

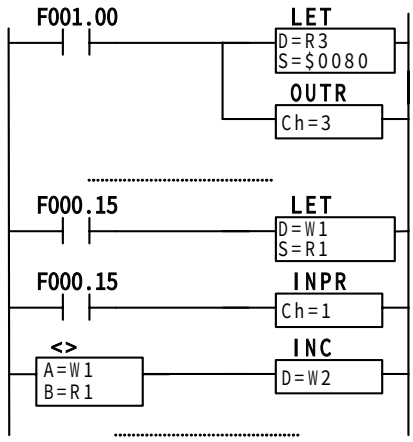
도움말

1. PLC 의 입출력 데이터 처리의 순서는 ‘CPU 진단 (입력데이터 전송 (프로그램 연산 (출력 데이터 처리 (통신 및 기타 데이터 처리 ’의 순서로 진행됩니다.
2. INPR 및 OUTR 명령은 입출력 데이터 전송을 프로그램 스캔 중에 실시하고자 할 경우에 사용하며, 빠른 입력펄스를 감지하거나 특수모듈의 초기 데이터 설정등에 주로 사용합니다.
3. INPR 명령은 프로그램 수행 중 데이터를 입력모듈로부터 입력레지스터로 Update 시키는 기능이며, Ch 값은 지정된 입력의 워드번지(R 을 제외한 숫자)를 뜻합니다.
4. OUTR 명령은 프로그램 수행 중인 출력 레지스터값을 출력모듈로 전송하여 Update 시키는 기능이며, Ch 값은 지정된 출력의 워드번지(R 을 제외한 숫자)를 뜻합니다.
5. Ch 값은 0 ~ 63 까지 지정할 수 있으며, Local(기본 및 확장베이스)에 장착된 입출력 모듈의 번지를 지정하며, 리모트 영역의 입출력은 사용할 수 없습니다.

사용예제

장착된 입출력 모듈의 번지가 아래와 같이 입력 R0, R1, 고속카운터(입력 R2, 출력 R3)등의 경우를 예제로 간단하게 설명한 것입니다.

POWER	CPU	IN 16 R0	IN 16 R1	HSC (32) R2,3	OUT 16 R4	OUT 16 R5	→ 모듈종류 → 점유 점수 → 입출력 번지(워드)



프로그램이 시작되는 한 Scan 에서 F1.0 접점이 ON 되며, 출력모듈 R3 워드에 값을 $0080 으로 설정되고, 직접 출력모듈에 데이터를 전송하게 됩니다.

매 스캔마다 입력모듈의 데이터를 받아 프로그램을 수행하며, 프로그램 진행중에 입력모듈의 Data 가 변화된 것이 있는지 확인하고자 하는 프로그램이다.
즉, 프로그램 시작시점의 입력 R1 워드를 W1 에 저장하고, 다시 입력 모듈로부터 변화된 값을 INPR 명령어로 불러와서 R1 에 저장하며, 기존의 W1 값과 R1 값이 다르면 입력 접점에 변화가 생긴 것을 의미합니다.
매 스캔마다 이것을 점검하여 차이가 나는 회수를 카운터하여 W2 에 저장합니다.

- 시스템 제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
WAT	1	WAT	워치독 타이머의 경과치 값을 리셋시킴 (0 으로 처리)	주 1)
END	1	END	프로그램 수행종료(끝) (마지막 부분은 CPU 에서 자동생성)	전기종

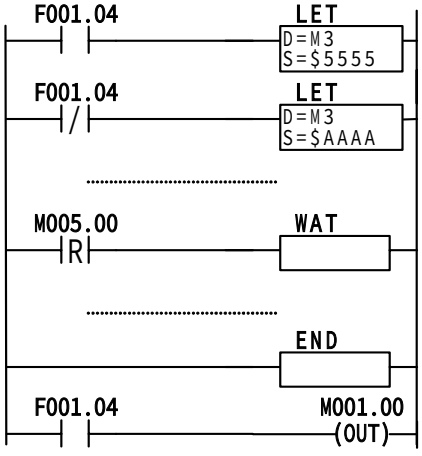
주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,9216A	NX70 plus CPU70p1/V2,CPU70p2/V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	○	○	○	○	○	○

도움말

1. WAT 명령은 프로그램 수행중에 워치독 타이머의 경과치를 리셋시켜 스캔타임이 워치독시간을 초과하여도 프로그램이 정지되지 않도록 하는 명령어 입니다.
2. CPU 에서는 프로그램을 스캔단위로 처리하며, 프로그램 수행과 입출력신호 및 기타 처리시간을 합친 시간을 스캔시간(1 Scan Time)이라 합니다.
3. 스캔타임이 지정한 워치독 타임보다 길어지면 프로그램 처리에 문제가 생긴것으로 판단하고 CPU 는 프로그램 수행을 중단하며, Error 를 발생시킵니다.
4. 스캔타임은 수행할 때 마다 달라지며, 이것을 점검하는 시계가 워치독 타이머이고, 워치독 타임은 프로그램 수행에 문제가 생겨 정지시키고자 하는 최대의 시간을 지정합니다. 최초에 기본으로 설정된 워치독 타임은 3000mSec(3 초)입니다.
5. 특정프로그램 수행시 워치독 타임보다 길어지는 경우가 예상되면 WAT 명령어로 워치독 경과시간을 초기화 하여 프로그램 수행에 이상이 없도록 해 줍니다.
6. WAT 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.
7. END 명령은 프로그램의 수행을 종료하고 다음 단계로 넘기는 명령이며, 프로그램 마지막 부분에는 자동으로 CPU 에서 삽입시켜 줍니다.
8. 프로그램을 부분적으로 점검하거나 강제로 프로그램을 종료하고자 할 경우에 사용합니다.
9. END 명령은 항상 Root 에서 직접 연결되어야 하며, 입력조건 없이 사용해야 합니다.
10. WAT, END 명령은 오퍼랜드가 없이 명령어만 사용합니다.

사용예제



F1.4 접점으로 M3 워드는 0.5 초 동안 $5555 을 대입하고, 0.5 초 동안은 $AAAA 를 대입합니다.
→ 일반적인 프로그램이 작동하는 예제입니다.

M5.0 이 ON 상태가 되면 워치독 타임의 경과치를 초기화 시킵니다.

프로그램 중간에 END 명령을 사용하여 강제로 프로그램을 종료시키면 F1.4 접점에 의한 M1.0 출력접점은 동작하지 않습니다.

- 통신제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
READ	5	READ TO=RR1 SZ=NR3 FR=NN5:NR6	고기능 입출력 모듈의 공유 메모리 데이터 읽기 -지정 Slot(NN5)의 내부번지(NN6) 부터 지정워드 수(NR3)를 읽어내어 레지스터(RR1)에 차례로 저장함.	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

참조) 오퍼랜드 형식 : NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

도움말

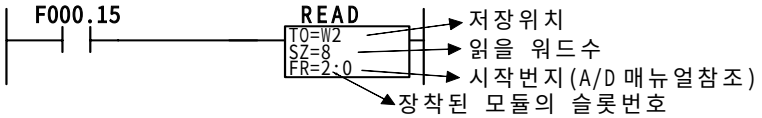
1. 고기능 입출력 모듈은 입출력 모듈에 전용의 프로세서와 메모리를 내장하여 PLC CPU 와 데이터를 교환하는 구조를 가진 것을 말하며, 아날로그 입출력, 고속카운터, 위치결정 모듈, SCU 통신모듈, DeviceNet, 리모트모듈, 링크모듈 등이 있다.
2. READ 명령어는 고기능 입출력 모듈에 내장된 메모리(공유메모리)의 데이터를 PLC CPU 가 읽어내기 위해 사용합니다.
3. RR1= 읽은 데이터를 보관할 워드의 시작번지(PLC CPU 의 레지스터),
   NR3= 읽고자 하는 데이터 또는 레지스터의 갯수,
   NN5= 입출력 모듈이 장착된 슬롯번호(0 부터 바이트 단위의 숫자, 슬롯번호 부문 참조)
   NR6= 고기능 입출력 모듈에서 읽고자 하는 공유메모리의 시작번지(숫자/레지스터)
4. READ 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제

고기능의 아날로그 입력모듈을 공유메모리 방식으로 번지지정을 하고, 공유 메모리 데이터를 읽기 위한 예제 입니다.

POWER	CPU	IN	IN	A/I	OUT	OUT	
		16	32	8Ch	16	32	→ 점유 점수(비트)
		R0	R1,2	R3	R4	R5,6	→ 입출력 번지(워드)
		0	1	2	3	4	→ 슬롯 번호

→ 공유메모리 방식으로 설정한 경우에 해당됨
(입력 접점방식의 경우 채널 수 만큼 워드지정(8 채널=8 워드 할당됨))



아날로그 입력모듈의 공유메모리 영역(0~52 번 워드까지)의
데이터 중 0~7 번까지 8 개 워드의 내용을 읽어서
W2 워드부터 W9 워드까지 저장하는 명령어 입니다.

아날로그 모듈의 공유메모리 영역

공유 메모리	데이터 값
00	500
01	1000
02	1500
03	2000
04	2500
05	3000
06	3500
07	4000
…	…
52	…

PLC CPU 내부 레지스터 영역

CPU 메모리	데이터 값
W0000	…
W0001	…
W0002	500
W0003	1000
W0004	1500
W0005	2000
W0006	2500
W0007	3000
W0008	3500
W0009	4000
W0010	…

- 통신제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
WRITE	5	WRITE TO=NN1:NR2 SZ=NR3 FR=NR5	고기능 입출력 모듈의 공유 메모리에 데이터 쓰기 PLC 레지스터/숫자(NR5)에서 워드수(NR3)만큼 읽어내어 지정슬롯(NN1)의 모듈레지스터(RR2)에 차례로 저장함.	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

참조) 오퍼랜드 형식 : NN=숫자 표시, NR=숫자 또는 레지스터, RR=레지스터 표시

도움말

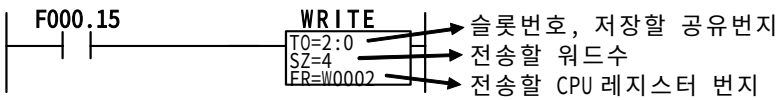
1. 고기능 입출력 모듈은 입출력 모듈에 전용의 프로세서와 메모리를 내장하여 PLC CPU 와 데이터를 교환하는 구조를 가진 것을 말하며, 아날로그 입출력, 고속카운터, 위치결정 모듈, SCU 통신모듈, DeviceNet, 리모트모듈, 링크모듈 등이 있다.
2. WRITE 명령어는 PLC CPU 의 레지스터 데이터를 고기능 입출력 모듈에 내장된 메모리 (공유메모리)에 데이터를 쓰기(전송하기) 위해 사용합니다.
3. NN1= 입출력 모듈이 장착된 슬롯번호(0 부터 바이트 단위의 숫자, 슬롯번호 부문 참조)
   NR2= 고기능 입출력 모듈에 쓰고자 하는 모듈 공유메모리의 시작번지(숫자/레지스터)
   NR3= 쓰고자 하는 데이터 또는 레지스터의 갯수
   NR5= 쓰고자 하는 PLC CPU 의 레지스터 시작번지(숫자/레지스터)
4. WRITE 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없고, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제

고기능의 아날로그 출력모듈을 공유메모리 방식으로 번지지정을 하고, 모듈의 공유 메모리에 데이터를 쓰기 위한 예제 입니다.

POWER	CPU	IN	IN	A/O	OUT	OUT	
		16	32	4Ch	16	32	→ 점유 점수(비트)
		R0	R1,2	R3	R4	R5,6	→ 입출력 번지(워드)
		0	1	2	3	4	→ 슬롯 번호

→ 공유메모리 방식으로 설정한 경우에 해당됨
(입력 접점방식의 경우 채널 수 만큼 워드지정(4 채널=4 워드 할당됨))



아날로그 출력모듈의 공유메모리 영역(0~47 번 워드까지)의 0~3 번 영역에 CPU 의 W2~W5 워드의 4 개를 전송하는 명령어 입니다.

PLC CPU 내부 레지스터 영역

CPU 메모리	데이터값
W0000	…
W0001	…
W0002	1000
W0003	1500
W0004	2000
W0005	2500
W0006	…
W0007	…
…	…

아날로그 출력모듈의 공유메모리 영역

공유 메모리	데이터값
00	1000
01	1500
02	2000
03	2500
04	…
05	…
…	…
…	…
47	…

- 통신제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
RMRD	7	RMRD TO=NR1:RR2 NT=NN3:NN4 Fr=NN5:NR6	리모트 슬레이브(자국)의 베이스에 장착된 고기능 입출력 모듈의 공유 메모리 데이터 읽기 리모트 Loop(NN3)의 Station(NN4)에 대한 Slot 번호(NN5)의 공유메모리(NR6)에서 워드수(NR1)를 읽어 레지스터(RR2)에저장	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

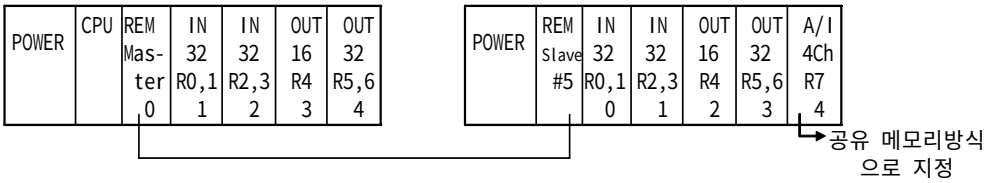
참조) 오퍼랜드 형식 : NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

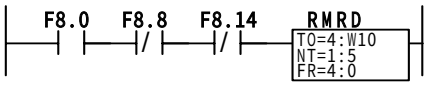
도움말

1. 고기능 입출력 모듈은 입출력 모듈에 전용의 프로세서와 메모리를 내장하여 PLC CPU 와 데이터를 교환하는 구조를 가진 것을 말하며, 아날로그 입출력, 고속카운터, 위치결정 모듈, SCU 통신모듈, DeviceNet, 등이 있다.
2. RMRD 명령어는 리모트 슬레이브(자국)의 베이스에 장착된 고기능 입출력 모듈의 공유메모리 데이터를 PLC CPU 가 읽어내기 위해 사용합니다.
3. NR1= 읽고자 하는 데이터 또는 레지스터의 갯수,
   RR2= 읽은 데이터를 보관할 워드의 시작번지(PLC CPU 의 레지스터),
   NN3= 리모트 네트워크의 번호(1, 2 또는 3 = 리모트 Master 가 1 개의 경우 1)
   NN4= 해당 리모트 네트워크에 연결된 Slave Station 의 번호
   NN5= 읽고자 하는 모듈이 장착된 슬롯 번호
   NR6= 고기능 입출력 모듈에서 읽고자 하는 공유메모리의 시작번지(숫자/레지스터)
4. RMRD 명령은 SCAN 완료후에 실행 하므로 1Scan 내에 해당 리모트 네트워크에 중복으로 수행되지 않도록 프로그램 작업시 주의해야 합니다.
5. RMRD 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제

리모트 Master 가 장착된 상태에서 Slave ID 5 번의 4 번슬롯 번호에 있는 고기능의 아날로그 입력모듈을 공유메모리 방식으로 번지지정을 하고, 공유 메모리 데이터를 읽기 위한 예제.





리모트 통신이 정상일 때, Slave 에 장착된 4 채널 아날로그 입력모듈의 입력 데이터를 읽어 W10~13 번 워드에 저장하는 명령어 입니다.

아날로그 입력모듈의 공유메모리 영역

공 유 메 모 리	데 이 터 값
0 0	1 0 0 0
0 1	1 5 0 0
0 2	2 0 0 0
0 3	2 5 0 0
0 4	…
0 5	…
…	…
…	…
5 2	…

PLC CPU 내부 레지스터 영역

CPU 메 모 리	데 이 터 값
…	…
W 0 0 0 9	…
W 0 0 1 0	1 0 0 0
W 0 0 1 1	1 5 0 0
W 0 0 1 2	2 0 0 0
W 0 0 1 3	2 5 0 0
W 0 0 1 4	…
W 0 0 1 5	…
…	…

- 통신제어 명령어

니모닉명령	워드수	래더 표시	설 명	적용기종
RMWR	7	RMWR NT=NN1:NN2 TO=NN3:NR4 FR=NR5:NR6	리모트 슬레이브(자국)의 베이스에 장착된 고기능 입출력 모듈의 공유 메모리에 데이터 쓰기 CPU 의 데이터/레지스터(NR6)부터 워드수(NR5)를 읽어 리모트 Loop(NN1) Station(NN2)의 Slot 번호(NN3)의 공유메모리(NR4)에 저장	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

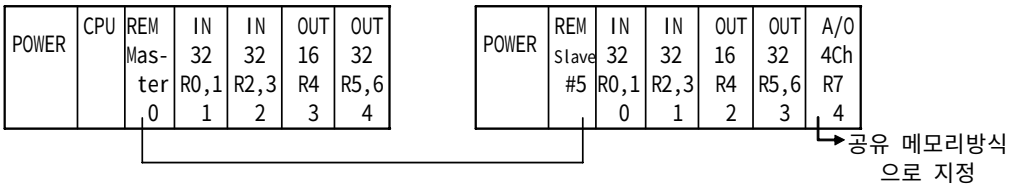
참조) 오퍼랜드 형식 : NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

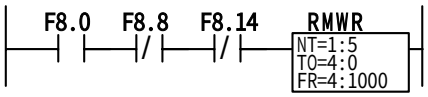
도움말

1. 고기능 입출력 모듈은 입출력 모듈에 전용의 프로세서와 메모리를 내장하여 PLC CPU 와 데이터를 교환하는 구조를 가진 것을 말하며, 아날로그 입출력, 고속카운터, 위치결정 모듈, SCU 통신모듈, DeviceNet, 등이 있다.
2. RMWR 명령어는 리모트 슬레이브의 베이스에 장착된 고기능 입출력 모듈의 공유 메모리에 데이터를 쓰기 위해 사용합니다.
3. NN1= 리모트 네트워크의 번호(1, 2 또는 3 = 리모트 Master 가 1 개의 경우 1)
   NN2= 해당 리모트 네트워크에 연결된 Slave Station 의 번호
   NN3= 쓰고자 하는 모듈이 장착된 슬롯 번호
   NR4= 고기능 입출력 모듈에 쓰고자 하는 공유메모리의 시작번지(숫자/레지스터)
   NR5= 쓰고자 하는 데이터 또는 레지스터의 갯수,
   RR6= 쓰고자 하는 데이터가 저장되어 있는 CPU 의 시작번지(PLC CPU 의 레지스터),
4. RMWR 명령은 SCAN 완료후에 실행 하므로 1Scan 내에 해당 리모트 네트워크에 중복으로 수행되지 않도록 프로그램 작업시 주의해야 합니다.
5. RMWR 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제

리모트 Master 가 장착된 상태에서 Slave ID 5 번의 4 번슬롯 번호에 있는 고기능의 아날로그 출력모듈을 공유메모리 방식으로 번지지정을 하고, 공유 메모리에 데이터를 쓰기 위한 예제.





리모트 통신이 정상일 때, Slave 에 장착된 4 채널 아날로그 출력모듈에 데이터 1000 을 전송(쓰기)하는 명령어 입니다.

<PLC CPU 내부 레지스터 영역>

<아날로그 출력모듈의 공유메모리 영역>

데 이 터
1000

공 유 메 모 리	데 이 터 값
0 0	1 0 0 0
0 1	1 0 0 0
0 2	1 0 0 0
0 3	1 0 0 0
0 4	…
0 5	…
…	…
…	…
4 7	…

- 통신제어 명령어 (RECV)

니모닉명령	워드수	래더 표시	설 명	적용기종
RECV	7	RECV TO=NR1:RR2 NT=NN3:NN4 FR=NN5:NR6	링크통신 모듈을 이용한 워드 데이터 수신명령 링크 네트워크(NN3)의 Station(NN4)에 대한 레지스터 종류(NN5)의 지정위치(NR6)에서 지정워드 수(NR1)를 읽어 레지스터(RR2)에 차례로 저장함.	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

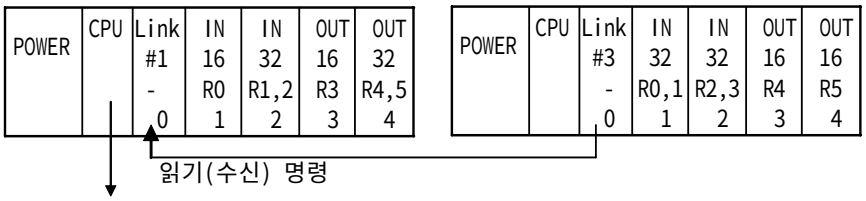
참조) 오퍼랜드 형식 : NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

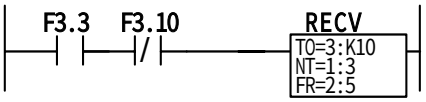
도움말

1. RECV 명령어는 링크 네트워크를 구성한 상태에서 다른 ID 를 가진 CPU 의 데이터를 읽어내기 위해 사용합니다.
2. NR1= 수신(읽고자 하는) 데이터 또는 레지스터의 개수.
   RR2= 수신(읽은) 데이터를 보관할 워드의 시작번지(PLC CPU 의 레지스터),
   NN3= 링크 네트워크의 Loop 번호(1, 2 또는 3 = 링크가 CPU 마다 1 개씩 장착된의 경우 1)
   NN4= 해당 링크 네트워크에 연결된 Station 의 번호(ID 번호)
   NN5= 읽고자 하는 레지스터의 종류(0= L 레지스터, 1=M, 2=R, 3=K, 4=SV, 5=PV,6=W,7=F)
   NR6= 읽고자 하는 레지스터의 시작번지(숫자/레지스터) – 원격 CPU
3. RECV 명령은 SCAN 완료후에 실행 하므로 1Scan 내에 해당 링크 네트워크에 중복으로 수행되지 않도록 프로그램 작업시 주의해야 합니다.
4. RECV 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ‘ 엣지접점 ’ 으로 사용해야 합니다.

사용예제

링크 통신모듈이 1 개씩 장착된 상태에서 Link ID 3 번 CPU 에 있는 R 레지스터(종류 2)의 5 번 워드부터 3 개를 읽어 자신의 K10 번지부터 저장하는 예제.





링크가 장착되어 있고, 송신 완료된 상태에서
ID #3 번의 R5 번지부터 3 개워드를 읽어
ID #1 번의 K10~12 번 워드에 저장하는 기능입니다.

PLC CPU #1 레지스터 영역

CPU #1 메모리	데이터값
…	…
K0009	…
K0010	$2345
K0011	$3456
K0012	$4567
K0013	…
K0014	…
…	…

PLC CPU #3 레지스터 영역

CPU #3 메모리	데이터값
…	…
R0003	…
R0004	$1234
R0005	$2345
R0006	$3456
R0007	$4567
R0008	…
R0009	…
…	…

- 통신제어 명령어 (SEND)

니모닉명령	워드수	래더 표시	설 명	적용기종
SEND	7	SEND NT=NN1:NN2 TO=NN3:NR4 FR=NN5:NR6	링크통신 모듈을 이용한 워드 데이터 송신명령 지정레지스터/숫자(NR6)의 지정워드 수(NR5)를 읽어 링크 네트워크(NN1)의 Station(NN2)에 레지스터 종류(NN3)의 지정위치(NR4)에 차례로 저장함	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

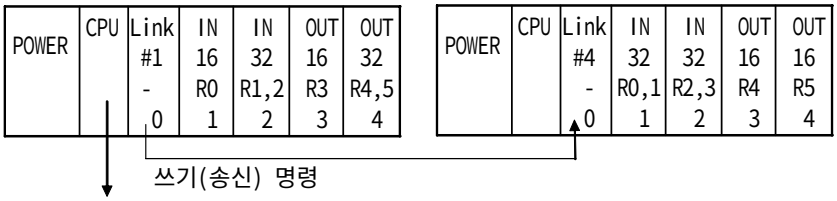
참조) 오퍼랜드 형식 : NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

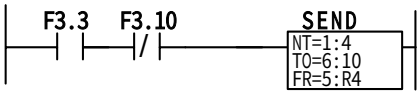
도움말

1. SEND 명령어는 링크 네트워크를 구성한 상태에서 CPU 의 데이터를 다른 ID 를 가진 CPU 에 데이터를 전송하기 위해 사용합니다.
2. NN1= 링크 네트워크의 Loop 번호(1, 2 또는 3 = 링크가 CPU 마다 1 개씩 장착된의 경우 1)
   NN2= 해당 링크 네트워크에 연결된 Station 의 번호(ID 번호)
   NN3= 저장할 레지스터의 종류(0= L 레지스터, 1=M, 2=R, 3=K, 4=SV, 5=PV,6=W,7=F)
   NR4= 전송한 레지스터를 저장할 시작번지(숫자/레지스터) - 원격지 CPU.
   NR5= 전송하고자 하는 데이터 또는 레지스터의 개수.
   NR6= 전송하고자 하는 데이터의 레지스터 번지
3. SEND 명령은 SCAN 완료후에 실행 하므로 1Scan 내에 해당 링크 네트워크에 중복으로 수행되지 않도록 프로그램 작업시 주의해야 합니다.
4. SEND 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

사용예제

링크 통신모듈이 1 개씩 장착된 상태에서 Link ID 4 번 CPU 의 W10~14 영역에
ID 1 번 CPU 의 R4 워드부터 5 개를 송신(쓰기)하는 명령 예제.





링크 네트워크 = 1, Station (ID)번호= 4
송신 저장번지 종류 W=6, 시작 번지 = 10
송신워드 개수 = 5, 송신워드 시작번지 = R4

PLC CPU #1 레지스터 영역

CPU #1 메모리	데이터값
…	…
R0003	…
R0004	$1234
R0005	$2345
R0006	$3456
R0007	$4567
R0008	$5678
R0009	…
…	…

PLC CPU #4 레지스터 영역

CPU #4 메모리	데이터값
…	…
W0009	…
W0010	$1234
W0011	$2345
W0012	$3456
W0013	$4567
W0014	$5678
…	…

- 통신제어 명령어 (RECVB)

니모닉명령	워드수	래더 표시	설 명	적용기종
RECVB	6	RECVB TO=BR1 NT=NN3:NN4 FR=NN5:NR6	링크통신 모듈을 이용한 비트 데이터 수신명령 링크 네트워크(NN3)의 Station(NN4)에 대한 레지스터 종류(NN5)의 지정 비트위치(NR6)의 값을 읽어 비트 레지스터(BR1)에 저장함.	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

참조) 오퍼랜드 형식 : BR=비트 레지스터 표시, NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

도움말

1. RECVB 명령어는 링크 네트워크를 구성한 상태에서 다른 ID 를 가진 CPU 의 비트 데이터를 읽어내기 위해 사용합니다.
2. BR1= 수신(읽고자 하는) 데이터를 저장할 비트번지.
   NN3= 링크 네트워크의 Loop 번호(1, 2 또는 3 = 링크가 CPU 마다 1 개씩 장착된의 경우 1)
   NN4= 해당 링크 네트워크에 연결된 Station 의 번호(ID 번호)
   NN5= 읽고자 하는 레지스터의 종류(0= L 레지스터, 1=M, 2=R, 3=K, 4=SV, 5=PV,6=W,7=F)
   NR6= 수신(읽고자 하는) 레지스터의 비트번지(숫자/레지스터) – 원격 CPU, 16 진수로 표시
   예) 3 번째 워드의 5 번째 비트를 지정할 경우 : NR6=$0035 로 지정함.
3. RECVB 명령은 SCAN 완료후에 실행 하므로 1Scan 내에 해당 링크 네트워크에 중복으로 수행되지 않도록 프로그램 작업시 주의해야 합니다.
4. RECVB 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

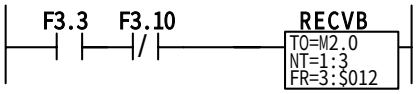
사용예제

링크 통신모듈이 1 개씩 장착된 상태에서 Link ID 3 번 CPU 에 있는 K 레지스터(종류 3)의 비트위치 $0012(1 번워드 비트번호 2(K1.2))를 읽어 자신(ID #1)의 M2.0 비트에 저장.

POWER	CPU	Link	IN	IN	OUT	OUT
		#1	16	32	16	32
		-	R0	R1,2	R3	R4,5
		0	1	2	3	4

POWER	CPU	Link	IN	IN	OUT	OUT
		#3	32	32	16	16
		-	R0,1	R2,3	R4	R5
		0	1	2	3	4

비트읽기(수신) 명령



수신 값 저장 비트번지 = M2.0
링크 네트워크 = 1, Station (ID)번호= 3
수신 저장번지 종류 K=3, 저장 비트번지=01.2

PLC CPU #1 레지스터 영역

CPU #1 메모리	데 이 터 값 (비 트)
M0000	…
M0001	…
M0002	00000000 00000001
M0003	00000000 00000000
M0004	00000000 00000000
M0005	…
M0006	…
…	…

PLC CPU #3 레지스터 영역

CPU #3 메모리	데 이 터 값 (비 트)
K0000	…
K0001	00001000 01000100
K0002	00000000 00000000
K0003	00000000 00000000
K0004	00000000 00000000
K0005	…
K0006	…
…	…

- 통신제어 명령어 (SENDB)

니모닉명령	워드수	래더 표시	설 명	적용기종
SENDB	6	SENDB NT=NN1:NN2 TO=NN3:NR4 FR=NB5	링크통신 모듈을 이용한 비트 데이터 송신명령 비트 레지스터/숫자(NB5)의 값을 읽어 링크 네트워크(NN1)의 Station(NN2)에 레지스터 종류(NN3)의 지정 비트위치(NR4)에 저장함	주 1)

주 1) 적용기종

SPC-10	SPC100 24S/120S	SPC300	A200	N70 plus 9215A,CPU70p1,V2	NX70 plus 9216A,CPU70p2,V2	N/NX700plus 7215A,CPU700p	NX700plus CPU700p/V2
×	×	×	×	×	○	○	○

참조) 오퍼랜드 형식 : BR=비트 레지스터 표시, NN=숫자표시, NR=숫자 또는 레지스터, RR=레지스터 표시

도움말

1. SENDB 명령어는 링크 네트워크를 구성한 상태에서 CPU 의 비트 데이터를 다른 ID 를 가진 CPU 에 전송하기 위해 사용합니다.
2. NN1= 링크 네트워크의 Loop 번호(1, 2 또는 3 = 링크가 CPU 마다 1 개씩 장착된의 경우 1)
   NN2= 해당 링크 네트워크에 연결된 Station 의 번호(ID 번호)
   NN3= 저장할 레지스터의 종류(0= L 레지스터, 1=M, 2=R, 3=K, 4=SV, 5=PV,6=W,7=F)
   NR4= 저장할 비트 번지(숫자/레지스터) - 원격지 CPU, 16 진수로 표시
   예) 3 번째 워드의 5 번째 비트를 지정할 경우 : NR6=$0035 로 지정함.
   NB5= 전송하고자 하는 비트 데이터 값.
3. SENDB 명령은 SCAN 완료후에 실행 하므로 1Scan 내에 해당 링크 네트워크에 중복으로 수행되지 않도록 프로그램 작업시 주의해야 합니다.
4. SENDB 명령의 동작은 접점과 같은 입력조건이 있어야 하며, Root 에서 직접 연결될 수 없으며, 입력조건에 따라 한번 작동시킬 경우에는 입력 접점을 ' 엣지접점 ' 으로 사용해야 합니다.

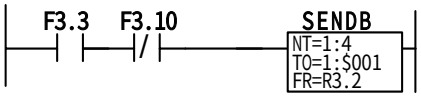
사용예제

링크 통신모듈이 1 개씩 장착된 상태에서 Link ID 4 번 CPU 의 M0.1 비트 영역에 ID 1 번 CPU 의 R3.2 비트값을 송신(쓰기)하는 명령 예제.

POWER	CPU	Link	IN	IN	OUT	OUT
		#1	16	32	16	32
		-	R0	R1,2	R3	R4,5
		0	1	2	3	4

POWER	CPU	Link	IN	IN	OUT	OUT
		#4	32	32	16	16
		-	R0,1	R2,3	R4	R5
		0	1	2	3	4

비트쓰기(송신) 명령



링크 네트워크 = 1, Station (ID)번호= 4
저장할 비트번지의 종류 M=1, 비트 번지 = 00.1
송신할 비트번지 = R3.2

PLC CPU #1 레지스터 영역

CPU #1 메모리	데이터 값(비트)
R0000	…
R0001	…
R0002	00000000 10000001
R0003	00000000 01000100
R0004	00000000 00000000
R0005	…
R0006	…
…	…

PLC CPU #4 레지스터 영역

CPU #3 메모리	데이터 값(비트)
M0000	00000000 00000010
M0001	00001000 01000100
M0002	00000000 00000000
M0003	00000000 00000000
M0004	00000000 00000000
M0005	…
M0006	…
…	…

7 장

제품일람 및 외형치수도

7-1. NX70 PLC 제품 일람

■ CPU유니트 (plus-시리즈)

WinGPC S/W 사용

유니트명	제품코드	제품사양	비 고
CPU 유니트	NX70-CPU70p1	• 9K Words (기본내장), 0.2μs/ STEP, RS232C/485 1포트, • FLASH ROM 내장 • 2K Data Register	
	NX70-CPU70p2	• 20K Words (기본내장), 0.2μs/ STEP, RS232C/485 2포트, • 시계기능, FLASH ROM내장, PID기능, 사용자정의 통신 • 4K Data Register	

■ CPU유니트 통신케이블

유니트명	제품코드	제품사양		비 고
CPU용 통신케이블	CPL5530	2m	NX70, NX700 CPU유니트 통신용 케이블	
	CPL5531	5m		

■ 마더보드

유니트명	제품코드	제품사양	비 고
기본 마더보드	NX70-BASE02	2슬롯 타입	NX70 PLC는 기본마더보드만 있습니다.
	NX70-BASE03	3슬롯 타입	
	NX70-BASE05	5슬롯 타입	
	NX70-BASE06	6슬롯 타입	
	NX70-BASE08	8슬롯 타입	
	NX70-BASE10	10슬롯 타입	
	NX70-BASE12	12슬롯 타입	

■ 전원유니트

유니트명	제품코드	제품사양	비 고
전원 유니트	NX70-POWER1	AC 110/ 220V Free Voltage, 5V 3.5A, 24V 0.3A	AC 입력 Type
	NX70-POWER2	AC 110/ 220V Free Voltage, 5V 4.5A	
	NX70-PWRDC	DC 24V Input, 5V 4.5A	DC 입력 Type

■ 입출력 유니트

유니트명		제품코드	(구 코드)	제품사양	비 고
입력 유니트	16점	NX70-X16D	CPL93023	DC 12~24V, 단자대 20핀, 8점/ COM, (극성 +,- 공통)	단자대 타입
		NX70-X16D1	CPL93033	DC 24V, 단자대 20핀, 8점/ COM, (극성 +,- 공통)	
		NX70-X16A110	CPL93043	AC 100~120V, 단자대 20핀, 8점/ COM	
		NX70-X16A220	CPL93053	AC 200~240V, 단자대 20핀, 8점/ COM	
	32점	NX70-X32D	CPL93024	DC 12~24V, 커넥타 20핀 x 2, 8점/ COM, (극성 +,- 공통)	커넥타 타입
		NX70-X32D1	CPL93034	DC 24V, 커넥타 20핀 x 2, 8점/ COM, (극성 +,- 공통)	
출력 유니트	8점	NX70-Y8R	CPL93202	RELAY 출력, 단자대 20핀, 250V 3A, (4점/ COM x 1, 1점/ COM x 4)	단자대 타입
	16점	NX70-Y16R	CPL93103	RELAY 출력, 단자대 20핀, 250V 1A, 8점/ COM	
		NX70-Y16RV	CPL93203	RELAY 출력, 단자대 20핀, 250V 1A, 8점/ COM, 바리스타 있슴	
		NX70-Y16T	CPL93483	TR 출력(NPN), 단자대 20핀, 12~24V 0.6A, 8점/ COM	
		NX70-Y16SSR	CPL93703	SSR 출력, 단자대 20핀, 100~220V, 0.5A, 8점/ COM	
	32점	NX70-Y32T	CPL93484	TR 출력(NPN), 커넥타 20핀 x 2, 12~24V 0.4A, 16점/ COM(-)	커넥타 타입
		NX70-Y32P	CPL93584	TR 출력(PNP), 커넥타 20핀 x 2, 12~24V 0.4A, 16점/ COM(+)	
혼합 유니트	16점	NX70-XY16	CPL93088	• DC12~24V, 8점, 8점/ COM, (극성 +,- 공통), • RELAY 출력, 8점, 250V 1A, 8점/ COM, 단자대 20핀	입력 8점, 출력 8점
	32점	NX70-XY32	-	• DC12~24V 16점, 16점/ COM (극성 +,- 공통), 커넥타 20핀 • TR 출력(NPN) 16점, DC12~24V 0.4A, 16점/ COM	입력 16점, 출력 16점 (출시예정)
더미 유니트		NX70-DUMMY	CPL93000	Dummy(더미) 유니트	

■ 아날로그 유니트

유니트명	제품코드	제품사양	비 고
아날로그 입력 (A/D)	NX70-AI8C	8CH, Current Input, 16Bit A/D Converter, ±20mA, 0 20mA, 4 20mA 분해능 (0.519uA 2.0uA) 변환속도 1.25ms/Ch	단자대 타입 20핀 성능개선품 (Ver 2.0)
	NX70-AI8V	8CH, Voltage Input, 16Bit A/D Converter, ±5V, ±10V, 0 5V, 0 10V 분해능 (0.153mV 1.0mV), 변환속도 1.25ms/Ch	
	NX70-AI4C	4CH, Current Input, 16Bit A/D Converter, ±20mA, 0 20mA, 4 20mA 분해능 (0.519uA 2.0uA) 변환속도 1.25ms/Ch	
	NX70-AI4V	4CH, Voltage Input, 16Bit A/D Converter, ±5V, ±10V, 0 5V, 0 10V 분해능 (0.153mV 1.0mV), 변환속도 1.25ms/Ch	
아날로그 출력 (D/A)	NX70-AO4V	4CH Voltage Output, 14bit D/A Converter, ±10V, ±5V, 0 10V, 0 5V 분해능 (0.305mV 1.0mV), 변환속도 2.5ms/ Ch	
	NX70-AO4C	4CH Current Output, 14bit D/A Converter, 0 20mA, 4 20mA 분해능 (0.037uA 2.0uA) 4uA, 변환속도 2.5ms/ Ch	
	NX70-AO2V	2CH Voltage Output, 14bit D/A Converter, ±10V, ±5V, 0 10V, 0 5V 분해능 (0.305mV 1.0mV), 변환속도 2.5ms/ Ch	
	NX70-AO2C	2CH Current Output, 14bit D/A Converter, 0 20mA, 4 20mA 분해능 (0.037uA 2.0uA) 4uA, 변환속도 2.5ms/ Ch	
측온저항체입력 (RTD)	NX70-RTD4	4CH 3-Wire 방식, Pt100, Pt200, Pt500, Pt1000, JPt100, JPt200, JPt500, JPt1000, NI100, NI120, CU50, 300Ω, 600Ω, 2000Ω 분해능 0.1°C, 0.1°F, 10mΩ, 20mΩ, 변환속도 60ms/Ch	
열전대입력 (ThermoCouple)	NX70-TC4	4CH Type : B/ R/ S/ N/ K/ E/ J/ T(센서타입에 따라 온도범위가 다름) ±30mV (1uV /Bit), ±60mV (2uV /Bit) 분해능 0.1°C/ 0.1°F/ 1uV/ 2uV, 변환속도 60ms/Ch	

■ 통신 유니트

유니트명	제품코드	제품사양	비 고
컴퓨터 커뮤니케이션 유니트 (CCU+)	NX70-CCU+	NX Protocol 포트확장용 RS232C/ RS485 x 1포트 (1개 사용가능) (단 CPU Version 2.0 이상)	ROM V2.0이상
시리얼 커뮤니케이션 유니트 (SCU)	NX70-SCU	RS232C/ RS485 x 2포트 (래더프로그램으로 데이타 처리) • 각종 RS232C/ RS485 기기와의 Master 송수신가능 (Binary/ ASCII 코드처리_)	

■ 고속카운터/펄스출력 유니트

유니트명	제품코드	제품사양	비 고
고속카운터 유니트	NX70-HSC1	1CH 고속카운터 입력기능 (100Kcps) 바이너리 24Bit Up/Down 카운터	
	NX70-HSC2	2CH 고속카운터 입력기능 (100Kcps) 바이너리 24Bit Up/Down 카운터	
	NX70-HSC4	4CH 고속카운터 입력기능(200Kcps), 인터럽터 8점입력, 비교출력8점 내장 바이너리 32Bit Up/Down 카운터	CPU70p2전용 ROM V2.0이상
펄스출력	NX70-PULSE4	펄스출력 4Ch, 고속카운터 4Ch 겸용, 인터럽터 8점입력, 비교출력8점 내장 펄스출력 100KHz, CW/CCW, HSC 200Kcps, 32Bit, PWM 30KHz, Duty 0~100%(1%단위)	

■ PLC 네트워크용 유니트 (CPU 공통)

유니트명	제품코드	제품사양	비 고
W-Link MW-Link 유니트	NX70-WLINK NX70-MWLINK	W-Link 기능 (W-모드) • 기본마더보드에 장착하여 사용 • CCU포함 최대 3대까지 장착 • 기능 : PLC링크 (16대, L : 1,024점, W : 128워드), 컴퓨터링크, 데이터전송(16워드/ 1명령), 리모트 프로그래밍, • 전송속도 : 0.5Mbps, 전송거리: 총 800m, 2Line ※ 사용가능 기종 : NX70-CPU70p2	Twisted-Pair 케이블 사용

■ REMOTE I/O 시스템

제품 종류	제품 번호	제품 사양	비 고
REMOTE I/O SLAVE 유니트	NX70-SLAVE	마더보드의 CPU유니트에 실장함 상위 마스터모듈 : N700 PLC (CPL7740) 및 NX700 PLC (NX-MASTER)모듈과 통신	
I/O Link (SLAVE 유니트)	NX-IOLINK	• 독립(Stand Alone Type) 유니트 • Slave 모듈 + 1슬롯 마더보드로 구성 • NX70용 입출력 모듈을 빈 슬롯에 실장(1개)	NX700, N700 Master에 Slave로 사용

■ 프로그래밍 기기

유니트명	제품코드	제품사양	적용 제품	비 고
Handy-Loader	PGM-500	· 프로그램 작성,편집,모니터링, BACK LIGHT 부착 · 메모리 BACK-UP 가능 내장 · 폭넓은 LCD 지원 · RS-232C/ 485 통신지원	· SPC10 PLC . N70plus (CPL9215A) (CPL9216A) . N700plus (CPL7215A) . NX7 PLC . NX70 PLC (NX70-CPU70p1) (NX70-CPU70p2) . NX700 PLC (NX-CPU700p)	케이블 포함
사용자 S/W	WinGPC S/W (Window용)	· 컴퓨터에서 PLC 프로그램 작성 및 모니터링, FILE 관리/ 보관기능/ ON-LINE 편집기능 /에러 및 PLC 상태 체크기능/ 네트워크 체크기능/ I/O MAP 기능 / TIME CHART 모니터링 기능등을 수행할 수 있는 컴퓨터용 S/W . 현 WinGPC(Window용) Ver 3.xx임		Windows 98 이상

■ CPU 유니트 통신케이블

통신포트명	제품 코드	제품 사양	용도	비 고
CPU 유니트용 케이블	NX-CBLCPU2 (CPL5530)	2m	WinFPST S/W(컴퓨터) 접속용	첫페이지 통신핀사양 참조할것
	NX-CBLCPU5 (CPL5531)	5m		

■ 입출력 하네스류

유니트명		제품 코드	제품 사양	비 고
I/O 케이블 ASS'Y	NX70 I/O (커넥타 타입)	CPL8800	DC IN 32점/ 64점 커넥타 하네스	하네스 길이 1.5m 핀수 : 20핀
		CPL8810	TR OUT 32점/ 64점 커넥타 하네스	
		CPL8820	RELAY OUT 32점 커넥타 하네스	
I/O 커넥타 ASS'Y	NX70 I/O (커넥타 타입)	CPL8880	20핀 (PIN 20개, 커넥타 후드포함)	NX70, N70, N700 I/O의 커넥타 타입 유니트용

■ 참고 사항

● PLC는 N-시리즈와 N-plus 시리즈가 있습니다. (명령어 및 사용자 S/W가 다르므로 주의바랍니다.)

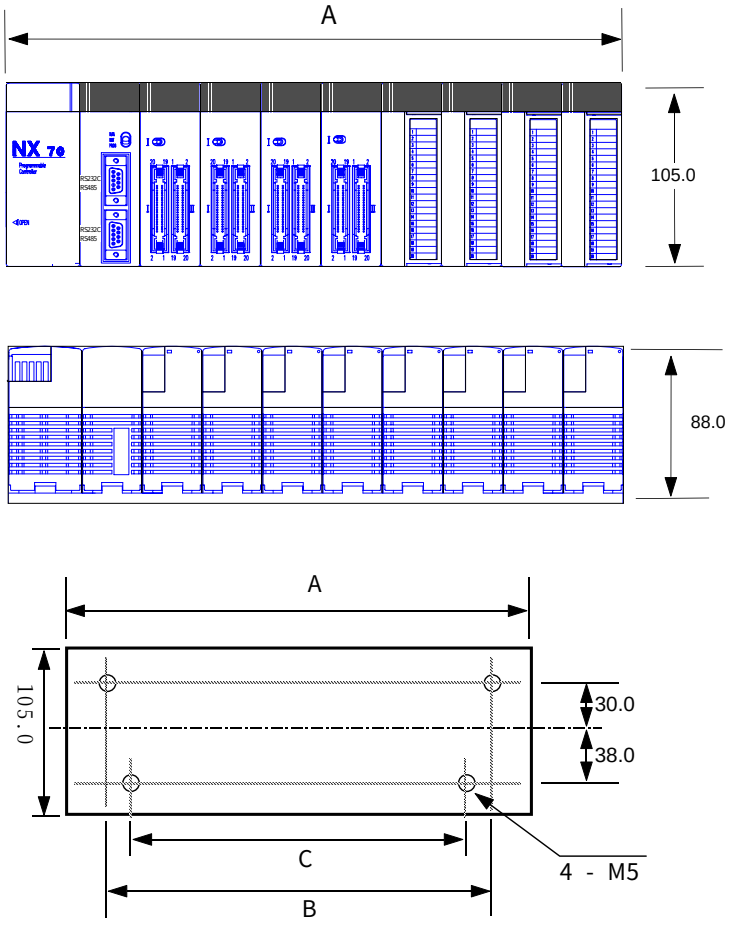
● 특히, N70 시리즈, N700 시리즈, NX70 시리즈, NX700 시리즈는 CPU 모듈외 모든 모듈(POWER, I/O, 마더보드, 특수모듈...)을 공통으로 사용하고 있습니다.

시리즈명	제품명	CPU 모듈명	비 고
N-plus 시리즈	**SPC10 시리즈**	SPC-28ADR, SPC-28ADT, SPC-10ADR, SPC-10ADT, SPC-10DR.....	**WinGPC S/W 사용**
	NX7 시리즈	NX7-28ADR, NX7-28ADT, NX7-48ADR, NX7-48ADR.....	
	N70 시리즈	CPL9215A, CPL9216A	
	N700 시리즈	CPL7215A	
	NX70 시리즈	NX70-CPU70p1, NX70-CPU70p2	
	NX700 시리즈	NX-CPU700p	
N-시리즈	**N7 시리즈**	CPL02323, CPL02343C, CPL02543C.....	**WinFPST S/W 사용**
	N70 시리즈	CPL9210A, CPL9211A	
	N700 시리즈	CPL7210A, CPL7211A, CPL6210A, CPL6210B, CPL6215A	
	NX70 시리즈	NX70-CPU70, NX70-CPU750	
	NX700 시리즈	NX-CPU750A, NX-CPU750B, NX-CPU750C, NX-CPU750D, NX-CPU700	

7-2. NX70 PLC 외형치수도

1) 시스템 치수도(mm)

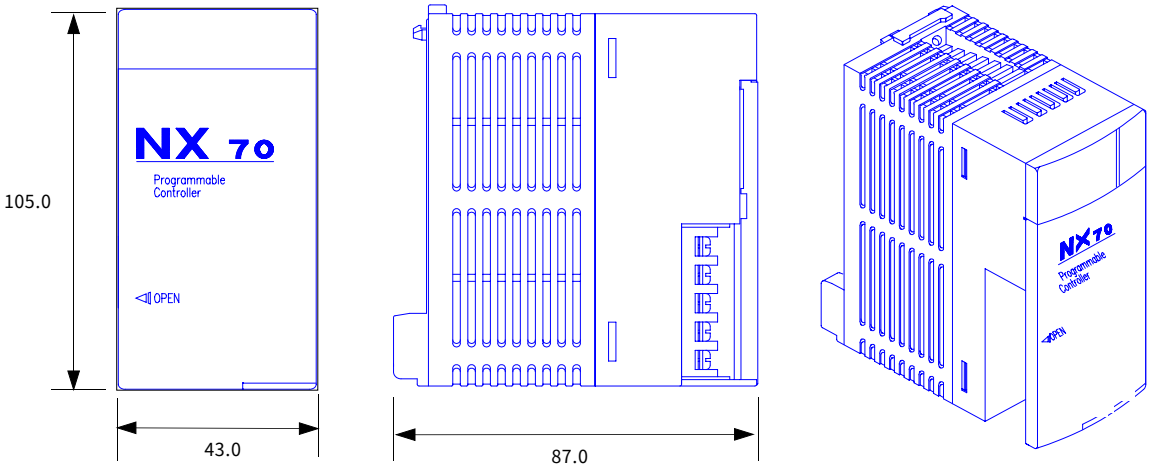
■ 외형 치수도(단위 : mm)



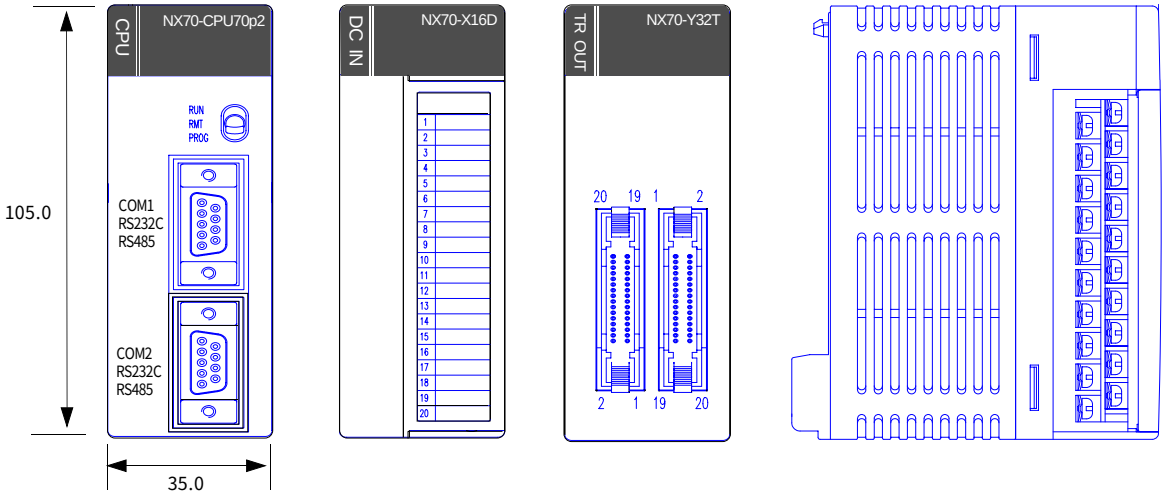
단위 (mm)

슬롯 타입	제품 번호	치 수 (A)	치 수 (B)	치 수 (C)
2 슬롯형	NX70-BASE02	157.2	128.6	114.6
3 슬롯형	NX70-BASE03	192.4	163.8	149.8
5 슬롯형	NX70-BASE05	262.8	234.2	220.2
6 슬롯형	NX70-BASE06	298.0	269.4	255.4
8 슬롯형	NX70-BASE08	368.4	339.8	325.8
10 슬롯형	NX70-BASE10	438.8	410.2	396.2
12 슬롯형	NX70-BASE12	508.6	480.6	466.6

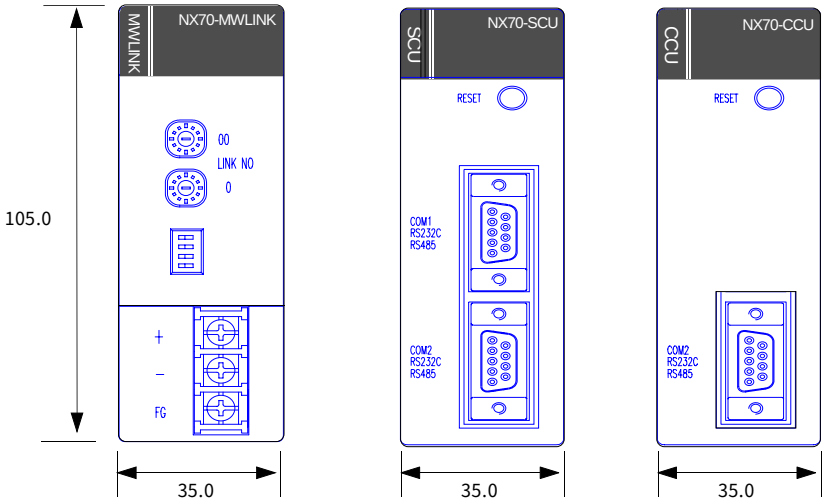
2) 전원유니트 치수도(mm)



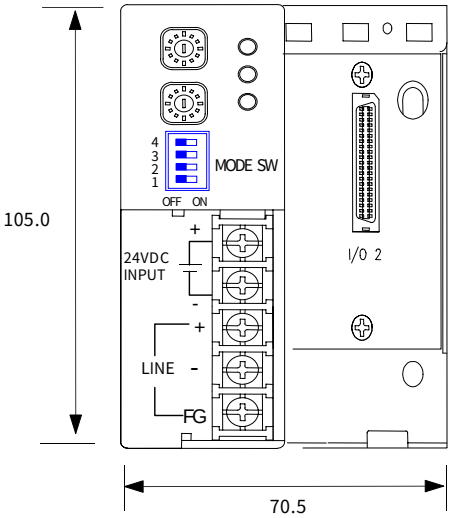
3) CPU, I/O유니트 치수도(mm)



4) 특수유니트 치수도(mm)



5) NX-IOLINK 유니트 (mm)



부록

통신규약 및 절차

1-1. NX-plus CPU 의 통신규약

통신을 위한 기본사양은 인터페이스 및 전송속도를 조정할 수 있으며, 기타 통신방식은 표와 같이 정의되어 있습니다.

항 목	사 양
인터페이스	RS-232C/RS-485
전송 속도	4800/ 9600/ 19200/38400 bps
통신 방식	반 2중 비동기방식 (Half DUPLEX Asynchronous)
동기 방식	Polling 방식
전송 코드	Binary (HEX)
전송데이타 포맷	데이타 길이 : 8bit
	STOP BIT : 1bit
	Parity : 없음 (NO Parity)
에러 체크 방식	CRC-16 방식
데이타 전송 순서	Byte 단위로 하위 바이트 부터 전송
전송 케이블	Twisted Pair Cable

1-2. NX-plus CPU 의 통신 프로토콜(Protocol)

N plus계열의 CPU는 통신을 위한 Protocol을 2가지 방법으로 제공합니다. Master(PC등)에서 통신절차를 Q(Query), QA(Query Acknowledge), RR(Response Request), R(Response)등을 거쳐 완성시키는 4단계 통신이 있고, Master에서 Q를 보내면 즉시 R로 회신을 하는 2단계 통신을 모두 지원합니다. 이 통신의 단계는 기능코드에 따라 달라집니다.

1단계 Q — **Query**

통신을 원하는 PLC의 고유번호와 Function Code를 설정하여 주변장치(Master)에서 PLC로 보내는 신호

2단계 QA — **Query Acknowedge**

해당 고유번호의 PLC가 주변장치(Master)로부터 Q신호를 정상적으로 수신했다는 인식신호로써 PLC에서 주변기기(Master)로 회신하는 신호.

3단계 RR — **Response Request**

주변장치(Master)가 PLC에서 보낸 QA신호를 정상적으로 인색했다는 인식신호로써 주변장치에서 PLC로 보내는 신호.
QA 신호가 정상적인 경우에 보내는 신호.

4단계 R — **Response**

PLC는 주변장치가 보낸 RR을 정상적으로 수신했을 경우 주변장치와의 통신이 원활하다고 판단하여 주변장치에서 최초에 보낸 Q에 실려 있었던 Function Code에 대한 처리를 마치고 그 결과를 주변장치로 보내주는 신호. PLC에서 R을 보냄으로써 한 Function Code에 대한 통신 사이클을 종료한다.

■ CPU 모듈의 통신 지연시간

PLC는 Q나 RR을 수신한 뒤 일정시간내에 반드시 다음 신호를 되돌려 주게 된다. 그러나 통신선로 이상, CRC값, 통신속도 불일치 등으로 PLC가 주변장치부로터 신호를 받지 못하는 경우가 발생할 수도 있다. 이때 주변장치는 Q나 RR에 대한 응답을 기다리다가 3초 이내에 응답이 없으면 통신에 이상이 생긴 것으로 간주하고 다시 Q나 RR을 송신하여야 한다.

■ CPU 모듈의 고유번호 (CPU ID)

NX-plus PLC CPU 에는 항상 고유번호를 가지고 있다. 이것은 CPU가 별도의 장치없이 RS-485 통신망을 구성할 수 있게 된다. 통신망에 연결된 모든 장치들은 각각의 통신을 위한 고유번호가 할당되어져야 하며 통신시에 그 고유번호와 통신이 이루어져야 한다.
N plus Series는 고유번호를 0 191까지 사용자가 임의로 설정하여 사용할 수 있다.
그러나, 한번 지정된 번호를 다른 CPU모듈의 고유번호로 중복 지정하여 통신망을 형성하면 안된다. 하나의 CPU모듈과 하나의 주변장치로 연결된 경우에 고유번호는 통상적으로 0, 1 또는 255로 지정되지만 여러대의 CPU모듈이 하나의 통신망에 연결되어 있으면 서로 다른 ID를 사용하여야 한다.
단, 고유번호를 255로 지정하면 모든 CPU에서 회신을 할 수 있는 상태가 되며, ID를 지정하지 않는 것과 동일하다. 즉, 통신망에 연결된 모든 CPU모듈과 통신하도록 할당된 번호이다. 그러나, N-plus Series는 어느 순간에 두대 이상의 CPU모듈과 동시에 통신을 할수 없으므로 255번 ID를 동시에 두대 이상의 CPU모듈의 ID로 지정하여 사용하면 통신에러가 발생한다. 두대 이상의 CPU모듈과 통신하고자 할 때는 반드시 다른 ID를 사용하여야 한다.
고유번호는 S/W(GPC5,WinGPC등) 또는 주변장치로 지정할 수 있으며, 래더 프로그램으로도 지정할 수 있고, Link 모듈을 사용할 경우에는 Link모듈의 ID번호를 따르게 된다.

■ 통신 단계

N plus CPU에서 2단계와 4단계 통신을 지원하고 있으며, 2단계와 4단계 통신의 구별은 Q(Query) Frame에 기능코드(Function Code)를 구별하여 전송함으로써 쉽게 구현할 수 있습니다. 2단계통신이나 4단계통신일지라도 반복되는 기능에 대해서는 2단계로 통신을 수행할 수 있습니다. 반복되는 기능이란 Query로 송신했던 프래임을 다시 수행하고자 할 경우로 RR 신호만을 반복적으로 송신하여 결과를 수신하는 것을 말한다. 이것은 4단계 통신에서 빠른 데이터 수신을 위해 사용할 때 자주 이용된다.

● 2단계 통신방법

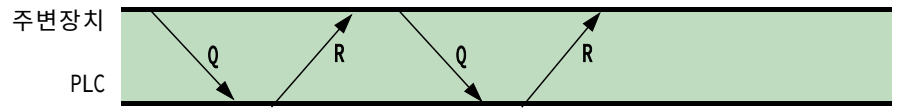
2단계 통신은 N plus CPU에서 지원되며, 유사한 기종의 SPC시리즈에는 제공이 되지 않습니다.
이 방식을 사용하면 기본적으로 Q신호 송신에 대한 R신호 회신이 이루어져 데이터를 빠르게 교환할 수 있습니다.
즉, 2단계 구성 : Q(1단계) → R(2단계)
반복기능 코드 : Q(1단계) → R(2단계) → RR(1단계) → R(2단계) .. 등으로 구성할 수 있다.
단, 반복기능 코드는 일부 CPU에서 제공되지 않으며, Q신호로 전송하면 동일한 속도가 가능하다.

● 4단계 통신방법

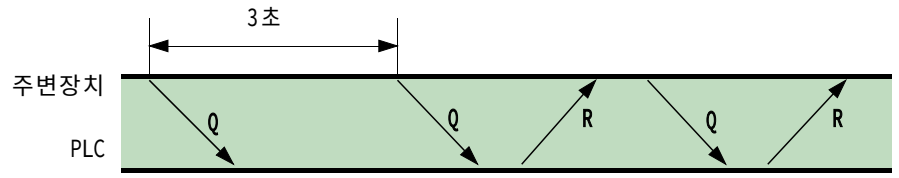
4단계 통신은 N plus CPU 및 SPC시리즈에도 제공이 되는 Protocol이며, 기본적으로 Q → QA → RR → R 의 4단계로 구성됩니다. 여기서 RR신호를 이용하여 반복되는 기능코드로 전송하면 2단계로 통신이 이루어 집니다.
즉, 4 단계 구성 : Q(1단계) → QA(2단계) → RR(3단계) → R(4단계)
반복기능 코드 : Q(1단계) → QA(2단계) → RR(3단계) → R(4단계) → RR(1단계) → R(2단계)..
등으로 구성하여 제한적으로 빠른 회신을 얻을 수도 있다.

(1) 2단계 통신 방법

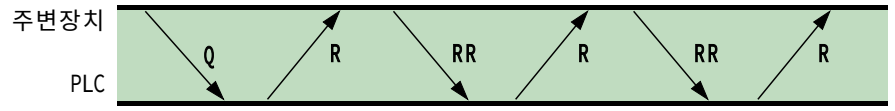
1) 통신에러가 없는 경우



2) R이 수신되지 않는 경우

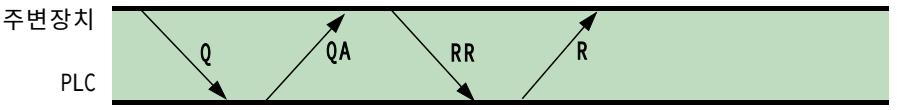


3) 반복되는 Function Code에 대한 응답

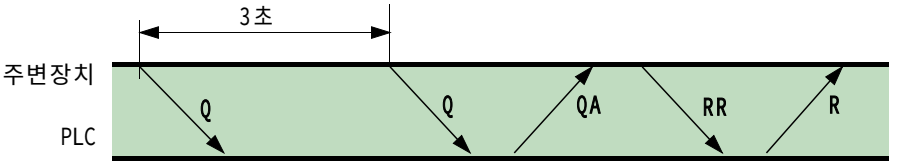


(2) 4단계 통신방법

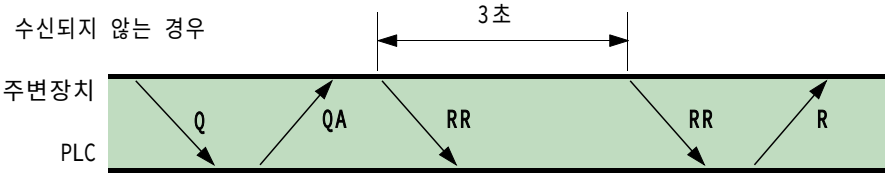
1) 통신에러가 없는 경우



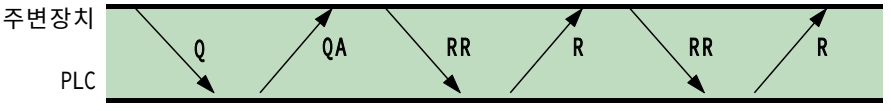
2) QA가 수신되지 않는 경우



3) R이 수신되지 않는 경우



4) 반복되는 Function Code에 대한 응답



(3) 쿼리(Query)에 실리는 기능코드(Function Code)

● 각각의 기능코드는 1바이트로 구성되며 PLC에서 정상적으로 쿼리(Q)신호를 수신했을때, R신호의 프레임에는 쿼리에서 보낸 기능코드에 $80(Hex)를 더하여 보낸다. 예를들면, PLC의 워드값을 읽는 FC(기능코드)인 $23을 Q신호를 통해 전송하면, R로 회신을 할때 $A3 ($80을 더한 값)의 FC로 회신이 된다. 2단계통신과 4단계의 통신의 차이점은 이 기능코드에 따라 다르게 회신하며, 쿼리나 Response 의 기능코드가 $20(Hex)의 차이가 있습니다.

● 주변기기는 자신이 보낸 쿼리를 PLC가 정상적으로 수신하였는지를 여부를 R 프레임의 Q Function Code를 검사하여 확인할 수 있다.

통신 기능 코드

주) $표시는 Hex(16진수)표기를 나타냄

통신 기능	쿼리(Q)의 Function Code		Response(R)의 Function Code		비 고
	2단계	4단계	2단계	4단계	
비트 읽기	$21	$01	$A1	$81	상세설명 참조
비트 쓰기	$22	$02	$A2	$82	〃
워드 읽기	$23	$03	$A3	$83	〃
워드 쓰기	$24	$04	$A4	$84	〃
비트/워드 혼용 읽기	$25	$05	$A5	$85	〃
비트/워드 혼용 쓰기	$26	$06	$A6	$86	〃
프로그램 읽기	$27	$07	$A7	$87	
프로그램 쓰기	$28	$08	$A8	$88	
명령어 읽기	$29	$09	$A9	$89	
명령어 변경	$2A	$0A	$AA	$8A	
오퍼랜드 변경	$2B	$0B	$AB	$8B	
명령어 삽입	$2C	$0C	$AC	$8C	
명령어 삭제	$2D	$0D	$AD	$8D	
명령어 찾기	$2E	$0E	$AE	$8E	
오퍼랜드 찾기	$2F	$0F	$AF	$8F	
프로그램 전부/부분 삭제	$20	$10	$A0	$90	
NO SERVICE	$00	$00	$00(Hex)	$00(Hex)	

참고

● 통신프레임의 비트/워드 번지지정은 절대번지 방식으로 사용하므로 절대번지 변환표를 참조 바랍니다.

● 프로그램 읽기, 쓰기등 기타 명기되지 않은 기능코드에 대한 자세한 내용은 당사 기술부서로 연락 바랍니다.

● Query란 ?

사전적인 의미로써 질문, 의문, 질문부호를 나타내며 통신에서 사용될 때는 사용자나 응용프로그램이 상대측에 어떠한 정보를 요구하는 것을 말합니다.

(4) 에러 체크 방식(CRC-16, Cycle Redundancy Checking : 순환 중복 검사)

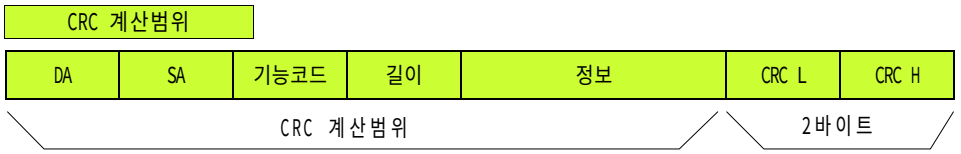
● 통신의 신뢰성 확보를 위해 에러를 체크하여 오류를 최소화 해야 합니다.
CPU 에서는 CRC-16방식(Cyclic Redundancy Checking, 순환중복검사)으로 에러를 점검하여 마지막 2Byte에 연결하여 전송한다.

● CRC 는 통신 프레임(Frame)이 에러없이 전송되었느지 여부를 확인하기 위하여 송신 프레임에 붙이는 일종의 체크섬 코드(Check Sum Code)정보로써 2바이트로 구성된다.

● 송/수신측은 수신데이타를 1바이트 송/수신할 때마다 CRC를 계산하여야 하므로 통신프로그램 작성시 CRC계산에 많은 시간이 소요되므로 통신속도 개선 및 통신에러를 방지하기 위해서는 이 부분의 속도를 향상시키도록 염두해 두어야 한다.

■ CRC-16의 계산

● CRC-16 은 통신 메세지의 전 블록(DA부터 정보의 마지막 데이터까지)를 직렬로 연결시켜 이 데이터를 정해진 17비트의 2진수(1 1000 000 000 0101)로 나눗셈한 나머지 2Byte값 입니다.



□ C++로 작성한 CRC-16계산 프로그램 예제 1

```
/*--------------------------------------------------------------------*/
// CRC-16 Routine with traditional method
// method 1. No need CRC table
/*--------------------------------------------------------------------*/
void CRC16_Test(int count)
{
    int i;

    Crc = 0xffff;
    for(i=0; i<count; i++)
    {
        CrcOld = Crc;
        Crc = Crc ^ (Data[i] & 0x00FF);
        for(int bit=0; bit<8; bit++)
        {
            if((Crc & 0x0001) == 0x0001)
                Crc = (Crc>>1) ^ 0xA001;
            else
                Crc = Crc>>1;
        }
        PRINT_CRC_JOBPROCESS(i, Data[i]);
    }
}
```

☐ C++로 작성한 CRC-16계산 프로그램 예제 2 (Table을 이용한 예)

```
/*--------------------------------------------------------------------*/
// CRC-16 Routine with CrcTable
// method 2. Need CRC table
/*--------------------------------------------------------------------*/
void CRC16_TestFast(int count)
{
    int i;
    unsigned int Temp;

    Crc = 0xffff;
    for(i=0; i<count; i++)
    {
        CrcOld = Crc;
        Temp = Crc ^ Data[i];
        Crc = (Crc>>8) ^ Crc16Table[Temp&0x00ff];
        PRINT_CRC_JOBPROCESS(i, Data[i]);
    }
}
```

Table 파일 (CrcTest.h)

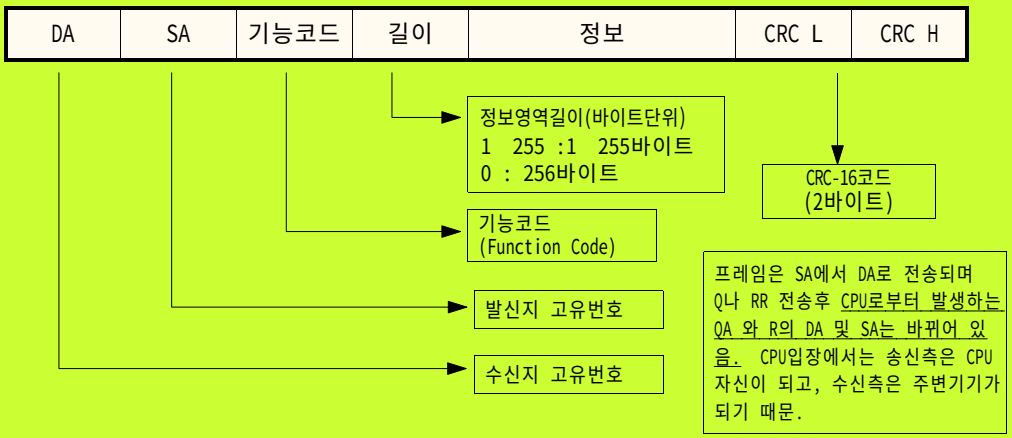
```
unsigned int Crc, CrcOld;

const unsigned int Crc16Table[256] =
{
    0x0000,0xc0c1,0xc181,0x0140,0xc301,0x03c0,0x0280,0xc241,
    0xc601,0x06c0,0x0780,0xc741,0x0500,0xc5c1,0xc481,0x0440,
    0xcc01,0x0cc0,0x0d80,0xcd41,0x0f00,0xcfc1,0xce81,0x0e40,
    0x0a00,0xcac1,0xcb81,0x0b40,0xc901,0x09c0,0x0880,0xc841,
    0xd801,0x18c0,0x1980,0xd941,0x1b00,0xdbc1,0xda81,0x1a40,
    0x1e00,0xdec1,0xdf81,0x1f40,0xdd01,0x1dc0,0x1c80,0xdc41,
    0x1400,0xd4c1,0xd581,0x1540,0xd701,0x17c0,0x1680,0xd641,
    0xd201,0x12c0,0x1380,0xd341,0x1100,0xd1c1,0xd081,0x1040,
    0xf001,0x30c0,0x3180,0xf141,0x3300,0xf3c1,0xf281,0x3240,
    0x3600,0xf6c1,0xf781,0x3740,0xf501,0x35c0,0x3480,0xf441,
    0x3c00,0xfcc1,0xfd81,0x3d40,0xff01,0x3fc0,0x3e80,0xfe41,
    0xfa01,0x3ac0,0x3b80,0xfb41,0x3900,0xf9c1,0xf881,0x3840,
    0x2800,0xe8c1,0xe981,0x2940,0xeb01,0x2bc0,0x2a80,0xea41,
    0xee01,0x2ec0,0x2f80,0xef41,0x2d00,0xedc1,0xec81,0x2c40,
    0xe401,0x24c0,0x2580,0xe541,0x2700,0xe7c1,0xe681,0x2640,
    0x2200,0xe2c1,0xe381,0x2340,0xe101,0x21c0,0x2080,0xe041,
    0xa001,0x60c0,0x6180,0xa141,0x6300,0xa3c1,0xa281,0x6240,
    0x6600,0xa6c1,0xa781,0x6740,0xa501,0x65c0,0x6480,0xa441,
    0x6c00,0xacc1,0xad81,0x6d40,0xaf01,0x6fc0,0x6e80,0xae41,
    0xaa01,0x6ac0,0x6b80,0xab41,0x6900,0xa9c1,0xa881,0x6840,
    0x7800,0xb8c1,0xb981,0x7940,0xbb01,0x7bc0,0x7a80,0xba41,
    0xbe01,0x7ec0,0x7f80,0xbf41,0x7d00,0xbdc1,0xbc81,0x7c40,
    0xb401,0x74c0,0x7580,0xb541,0x7700,0xb7c1,0xb681,0x7640,
    0x7200,0xb2c1,0xb381,0x7340,0xb101,0x71c0,0x7080,0xb041,
    0x5000,0x90c1,0x9181,0x5140,0x9301,0x53c0,0x5280,0x9241,
    0x9601,0x56c0,0x5780,0x9741,0x5500,0x95c1,0x9481,0x5440,
    0x9c01,0x5cc0,0x5d80,0x9d41,0x5f00,0x9fc1,0x9e81,0x5e40,
    0x5a00,0x9ac1,0x9b81,0x5b40,0x9901,0x59c0,0x5880,0x9841,
    0x8801,0x48c0,0x4980,0x8941,0x4b00,0x8bc1,0x8a81,0x4a40,
    0x4e00,0x8ec1,0x8f81,0x4f40,0x8d01,0x4dc0,0x4c80,0x8c41,
    0x4400,0x84c1,0x8581,0x4540,0x8701,0x47c0,0x4680,0x8641,
    0x8201,0x42c0,0x4380,0x8341,0x4100,0x81c1,0x8081,0x4040
};
```

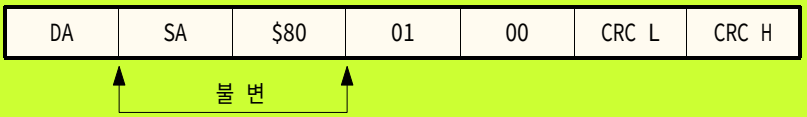
1-3 통신 프레임의 구조

● 기능코드는 2,4단계통신을 기준으로 Query 및 Response Frame의 예제를 설명하였습니다.

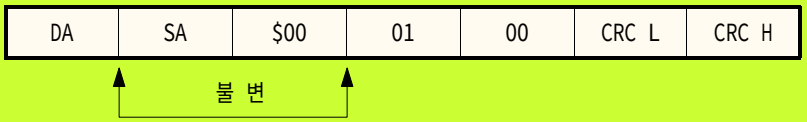
Q(Query) 및 R(Response) 프레임



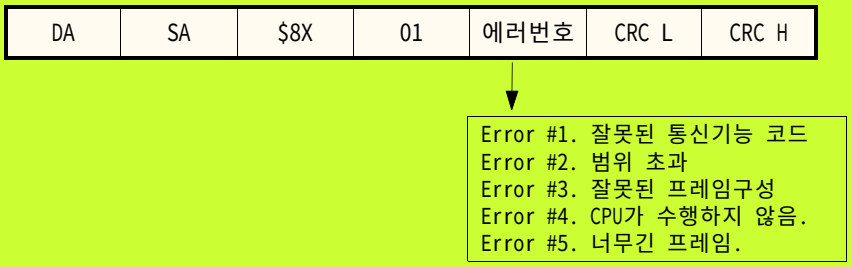
QA(Query Acknowledge) 프레임



RR(Response Request) 프레임



에러발생시 R(Response) 프레임

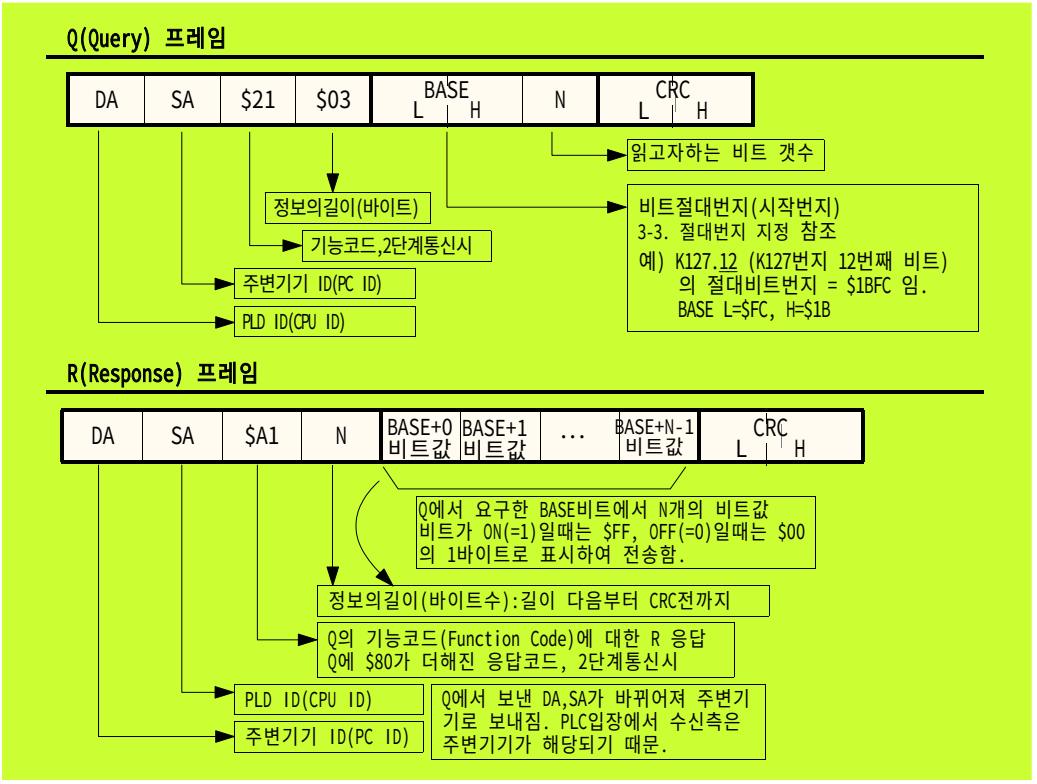


● 여기서 DA는 목적지의 ID번호, SA는 보내는 곳의 ID번호를 뜻하며, 먼저 보내는 Q의 경우 DA는 PLC의 ID번호이며, SA는 PC의 ID번호이다. 다시 회신하는 QA는 DA가 PC이며, PLC는 SA가 된다. 그러므로 주고받는 PLC의 ID번호에 맞게 프로그램을 해야 한다. 각각의 영역별로 1바이트씩(2 Digits) 값이 표시되어 있고, 정보 영역은 지정한 바이트에 따라 길이가 달라질 수 있다.

● 길이는 바로뒤에 있는 정보영역의 길이를 말하며, 00~FF까지 바이트 단위를 말하며, 256바이트까지 설정할 수 있다.

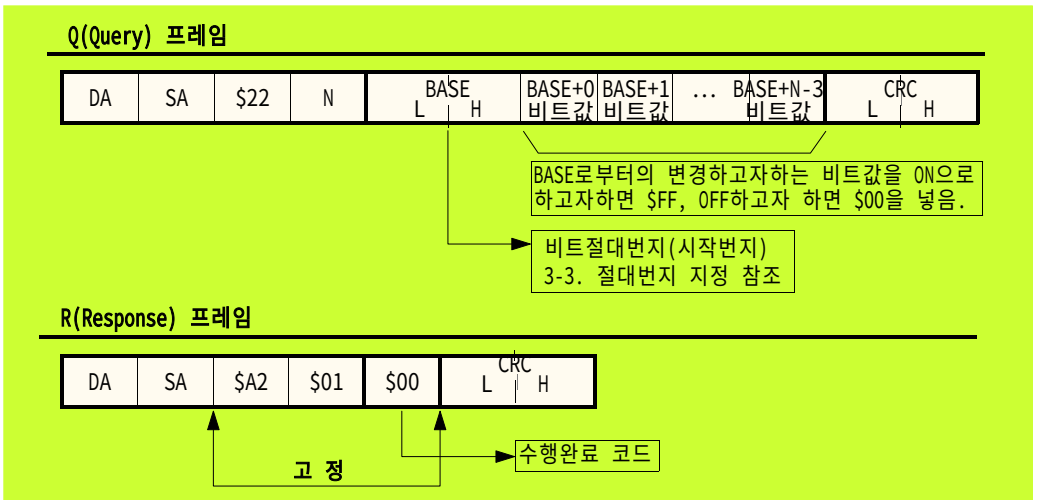
(1) 비트읽기

● 절대번지로 지정된 비트(R, L, M, K, F, TC)의 내용을 읽음.
● 연속하는 N개의 비트내용(ON/ OFF)를 읽을 수 있음.



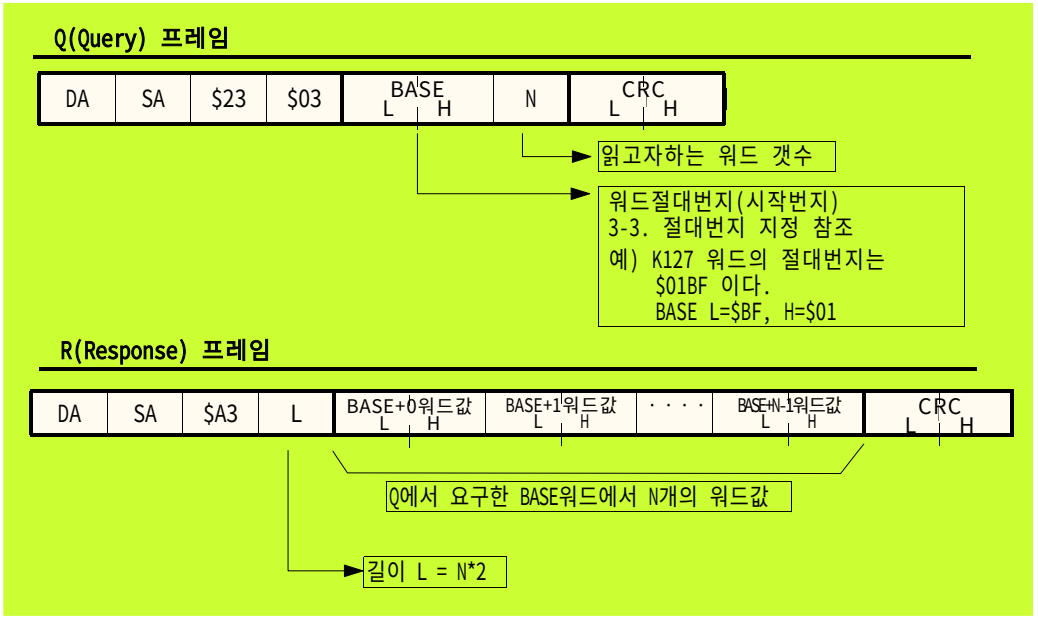
(2) 비트쓰기

● 절대번지로 지정된 비트(R, L, M, K, F, TC)의 내용을 변경함. ON/ OFF를 변경함.
● 연속하는 여러개의 비트 내용를 변경할 수 있음.



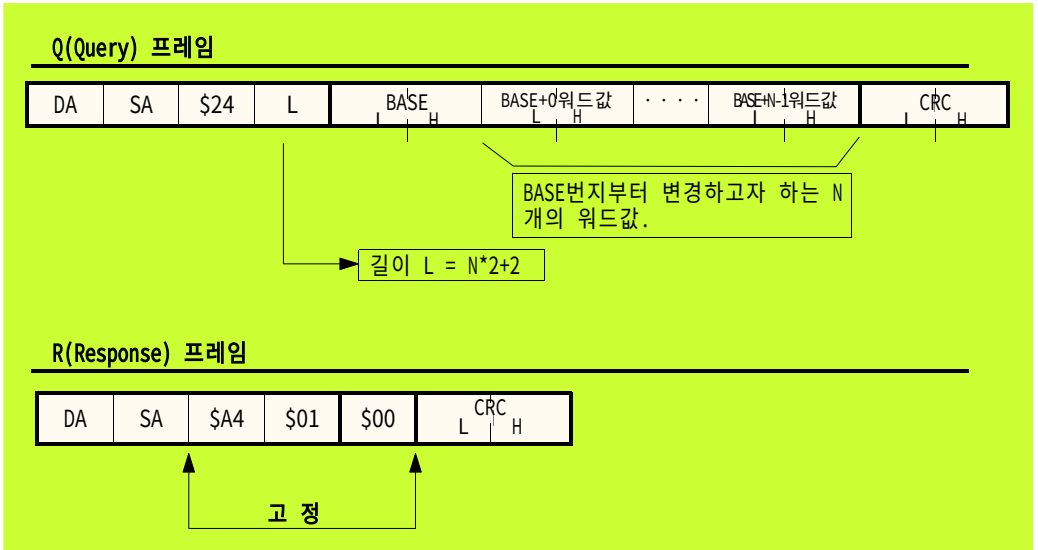
(3) 워드 읽기

● 절대번지로 지정된 워드(R, L, M, K, F, W)의 내용을 읽음.
● 연속하는 N개의 워드 내용을 읽을 수 있음.



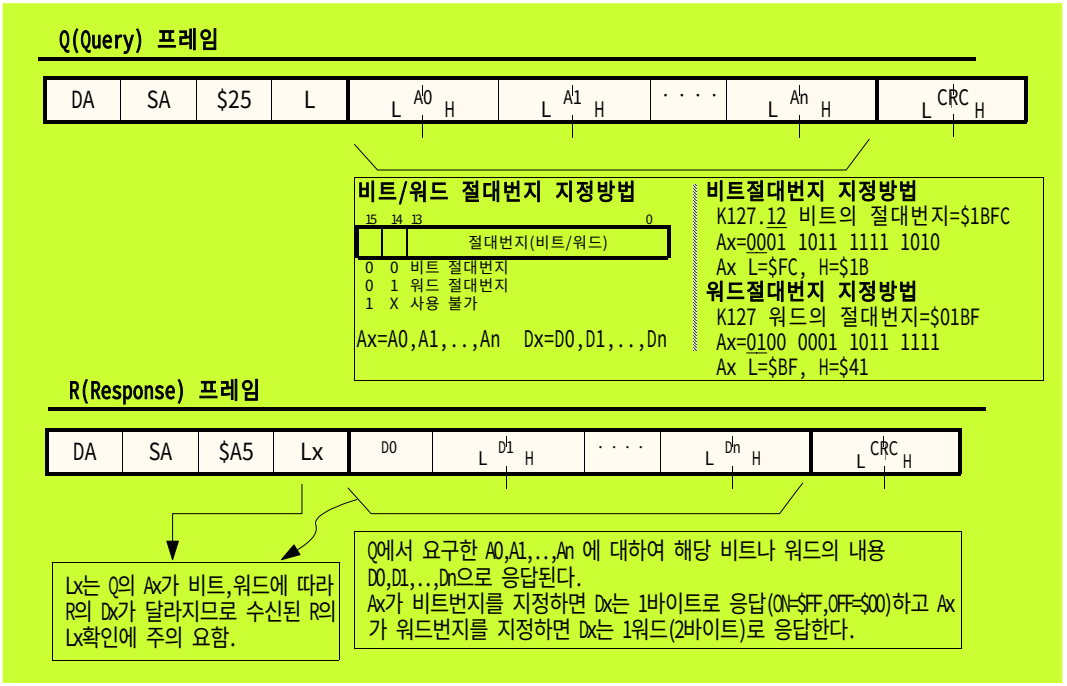
(4) 워드 쓰기

● 절대번지로 지정된 워드(R, L, M, K, F, W)의 내용을 변경함.
● 연속하는 N개의 워드 내용을 변경할 수 있음.



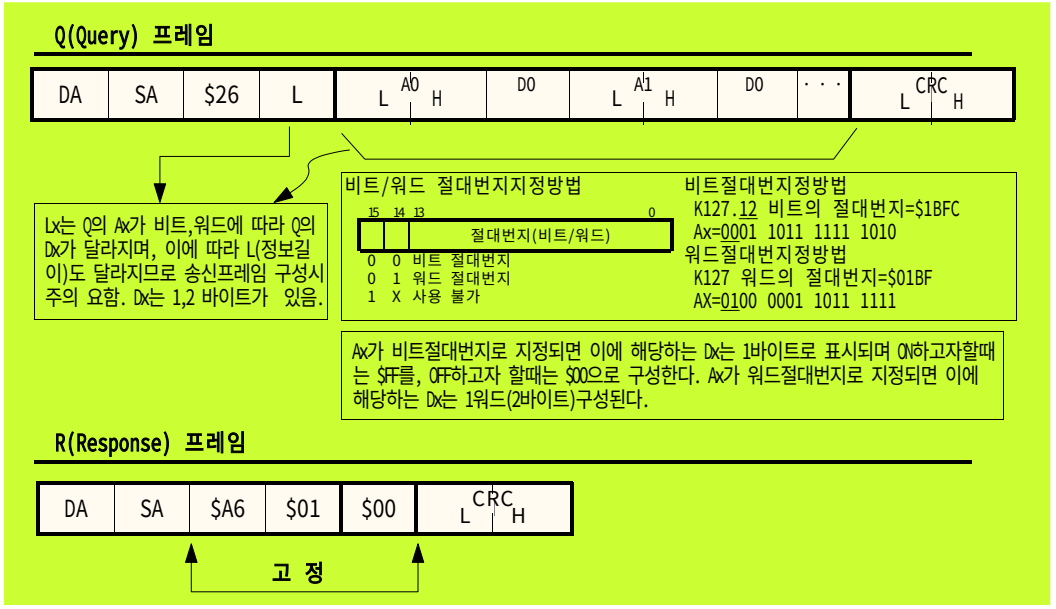
(5) 비트/워드 혼용하여 읽기

- 절대번지로 지정된 비트나 워드내용를 읽음.
- 비트와 워드를 순서없이 혼용하여 읽을수 있음.



(6) 비트/워드 혼용하여 쓰기

- 절대번지로 지정된 비트나 워드내용를 변경함.
- 비트와 워드를 순서없이 혼용하여 변경할수 있음.



1-4. 통신 프로그램 예제

- 본 예제는 사용자가 직접 통신프로그램을 작성을 위한 예제입니다.
- 본 예제는 C++로 작성된 통신 프로그램이며, Visual Basic 또는 다른 프로그램 예제는 문의 바랍니다.

프 로 그 램	설 명
`/////////////////////////////////////////////////////////////////////`	
`// NpDemo.cpp : Defines the entry point for the console application.`	
`//`	
`/////////////////////////////////////////////////////////////////////`	
`#include "NpDemo.h"`	
`#include <stdio.h>`	
`#include <conio.h>`	
`int main(int argc, char* argv[])`	
`{`	
`printf("************************************************************\n");`	
`printf("* N-plus series PLC Communication Test Program *\n");`	
`printf("************************************************************\n");`	
`//-----Serial port initialization`	
`printf("\nReady serial port init... ");`	
`if(OnlineConnection(1))`	COM1포트 접속
`{`	
`printf("Port open success!\n");`	
`}`	
`else`	
`{`	
`CloseHandle(hPort);`	포트 접속이 비정상적인 경우 포트를 닫고 종료
`printf("Port open error!\n");`	
`}`	
`//-----COMMUNICATION TEST`	
`if(Test_Read()!=CS_END)`	
`printf("Test_Read() failed!\n");`	테스트용 워드단위 읽기 함수 호출
`DsplPacket();`	결과출력
`if(Test_Write()!=CS_END)`	
`printf("Test_Write() failed!\n");`	테스트용 워드단위 쓰기 함수 호출
`DsplPacket();`	결과출력
`//-----End process`	
`if(hPort!=NULL)`	
`{`	
`if(CloseHandle(hPort))`	테스트후 사용했던 포트를 닫아 주어 다른 응용프로그램에서 사용할 수 있도록 한다
`printf("\nPort close success!\n");`	
`else`	
`printf("\nPort close failed!\n");`	
`}`	
`return 0;`	
`}`	
`/*-----------------------------------------------------------------*/`	
`// N-plus READ_WORD TEST`	
`// R0 : 1 word read`	
`/*-----------------------------------------------------------------*/`	
`BYTE Test_Read()`	
`{`	
`int i=0;`	
`Tx_Data.Data[i++] = 0xFF;`	
`Tx_Data.Data[i++] = 0xE1;`	받는 측 (PLC) ID 지정
`Tx_Data.Data[i++] = FC_READ_WORD_2STEP;`	보내는 측 (PC) ID 지정
`Tx_Data.Data[i++] = 0x03;`	2단계 통신용 워드읽기 통신 코드
`Tx_Data.Data[i++] = 0x00;`	정보 길이 지정
`Tx_Data.Data[i++] = 0x00;`	읽기 시작 어드레스L. 절대번지로 지정한다
`Tx_Data.Data[i++] = 0x01;`	읽기 시작 어드레스H
`CRC16(Tx_Data.Data, i++);`	읽고자 하는 워드 개수
`i--;`	CRC-16 계산
`Tx_Data.Data[i++] = LOBYTE(Crc);`	
`Tx_Data.Data[i++] = HIBYTE(Crc);`	CRC결과L
`Tx_Data.Size = i;`	CRC결과H
`RRA_LengthCalc(FC_READ_WORD_2STEP);`	
`printf("\nREAD_WORD Function Code Test\n");`	PLC로부터 수신될 개수 계산
`return StateMachine();`	
`}`	실제 데이터 송수신부 호출
`/*-----------------------------------------------------------------*/`	
`// N-plus READ_WORD TEST`	
`// M0~M4 : 5 word write`	
`/*-----------------------------------------------------------------*/`	

프 로 그 램	설 명
84 BYTE Test_Write() 85 { 86 int i=0, j, WordNumber=5; 87 88 Tx_Data.Data[i++] = 0xFF; 89 Tx_Data.Data[i++] = 0xE1; 90 Tx_Data.Data[i++] = FC_WRITE_WORD_2STEP; 91 Tx_Data.Data[i++] = WordNumber*2+2; 92 Tx_Data.Data[i++] = LOBYTE(0xc0); //0xc0 is M0 address 93 Tx_Data.Data[i++] = HIBYTE(0xc0); 94 95 for(j=0; j<WordNumber; j++) 96 { 97 Tx_Data.Data[i+j*2+0] = rand(); 98 Tx_Data.Data[i+j*2+1] = rand(); 99 } 100 i += WordNumber*2; 101 CRC16(Tx_Data.Data, i++); 102 i--; 103 Tx_Data.Data[i++] = LOBYTE(Crc); 104 Tx_Data.Data[i++] = HIBYTE(Crc); 105 Tx_Data.Size = i; 106 107 RRA_LengthCalc(FC_WRITE_WORD_2STEP); 108 printf("\nWRITE_WORD Function Code Test\n"); 109 return StateMachine(); 110 } 111 113 /*---*/ 114 // Query Acknwoledgement 115 /*---*/ 116 bool Q_Process() 117 { 118 BOOL bRet; 119 DWORD dwWritten, dwErr; 120 COMSTAT CommState; 121 122 printf("Q : "); 123 ClearCommError(hPort, &dwErr, &CommState); 124 PurgeComm(hPort, MY_PURGE_EVENT); 125 bRet = WriteFile(hPort, 126 Tx_Data.Data, 127 Tx_Data.Size, 128 &dwWritten, 129 NULL); 130 131 if(!bRet) return false; 132 133 while(dwWritten!=Tx_Data.Size); 134 135 return true; 136 } 137 138 /*---*/ 139 // Request Response Acknwoledgement 140 /*---*/ 141 bool RRA_Process() 142 { 143 BOOL bRet; 144 DWORD dwReaded=0, dwRdTotal=0, dwErr, StartTime; 145 COMSTAT CommState; 146 147 printf("RRA: "); 148 ClearCommError(hPort, &dwErr, &CommState); 149 PurgeComm(hPort, MY_PURGE_EVENT); 150 StartTime = GetTickCount(); 151 152 while((StartTime+1000)>GetTickCount()) 153 { 154 bRet = ReadFile(hPort, 155 (Rx_Data.Data+dwRdTotal), 156 Rx_Data.Size-dwRdTotal, 157 &dwReaded, 158 NULL); 159 if(!bRet) return false; 160 if(dwReaded!=0) 161 { 162 dwRdTotal+=dwReaded; 163 if(dwRdTotal==Rx_Data.Size) 164 return true; 165 } 166 } 167 168 return false; 169 }	받는 측 (PLC) ID 지정 보내는 측 (PC) ID 지정 2단계 통신용 워드쓰기 통신 코드 정보 길이 지정 쓰기 시작 어드레스L. 절대번지로 지정한다 쓰기 시작 어드레스H 실제 쓰고자 하는 값L 대입 실제 쓰고자 하는 값H 대입 쓰고자 하는 워드 개수 CRC-16 계산 CRC결과L CRC결과H PLC로부터 수신될 개수 계산 실제 데이터 송수신부 호출 새로운 데이터를 송수신하기 위하여 포트가 이전에 에러가 발생했는지, 쓰레기값들이 있는지 검사하고 이를 클리어 함. 포트와 데이터 포인터, 개수를 지정하고 WriteFile을 호출하면 포트로 데이터가 출력됨. 이때 출력된 개수를 dwWritten 로 리턴해주므로, 이것을 검사하면 실제 데이터가 정상적으로 출력되었는지 확인 할 수 있음. PLC로 데이터 전송(Q)후 1000ms 이내에 응답이 없을경우 통신 무시. PLC에서 송신하는 데이터를 PC에서 수신함. Rx_Data.Size=읽고자하는 총갯수는 앞서 결정됨. 1000ms 동안 수신되는 데이터를 계속 읽어내어 원하는 개수만큼 데이터을 얻어 냄.

프 로 그 램	설 명
171 /*---*/ 172 // Win32 Serial Communication Port Open 173 // method 1. No need CRC table 174 /*---*/ 175 bool OnlineConnection(int PortNumber) 176 { 177 char cpName[256]; 178 DCB CommDCB; 179 BOOL fRet; 180 181 hPort = NULL; 182 sprintf(cpName,"COM%d", PortNumber); 183 hPort = CreateFile(cpName, 184 GENERIC_READ|GENERIC_WRITE, 185 0, 186 NULL, 187 OPEN_EXISTING, 188 FILE_ATTRIBUTE_NORMAL, 189 NULL); 190 191 if(hPort == INVALID_HANDLE_VALUE) return FALSE; 192 193 // set buffer size 194 fRet = SetupComm(hPort, MAX_FRAME_LEN, MAX_FRAME_LEN); 195 if(!fRet) return FALSE; 196 197 // device clear 198 PurgeComm(hPort, MY_PURGE_EVENT); 199 if(!fRet) return FALSE; 200 201 // Comm timeout set 202 COMMTIMEOUTS CommTimeOuts; 203 CommTimeOuts.ReadIntervalTimeout = 1; 204 CommTimeOuts.ReadTotalTimeoutMultiplier = 5; 205 CommTimeOuts.ReadTotalTimeoutConstant = 100; 206 CommTimeOuts.WriteTotalTimeoutMultiplier = 5; 207 CommTimeOuts.WriteTotalTimeoutConstant = 100; 208 fRet = SetCommTimeouts(hPort, &CommTimeOuts); 209 if(!fRet) return FALSE; 210 211 // DCB Block Set 212 CommDCB.DCBlength = sizeof(DCB) ; 213 fRet = GetCommState(hPort, &CommDCB) ; 214 if(!fRet) return FALSE; 215 CommDCB.BaudRate = CBR_9600; 216 CommDCB.Parity = NOPARITY; // Parity Setting 217 CommDCB.ByteSize = 8; // Data Size Setting 218 CommDCB.StopBits = ONESTOPBIT; // Stop Bit Setting 219 fRet = SetCommState(hPort, &CommDCB); 220 if(!fRet) return FALSE; 221 222 return TRUE; 223 } 225 226 /*---*/ 227 // CRC-16 Routine with traditional method 228 // method 1. No need CRC table 229 /*---*/ 230 void CRC16(BYTE *src, int count) 231 { 232 int i; 233 234 Crc = 0xffff; 235 for(i=0; i<count; i++) 236 { 237 CrcOld = Crc; 238 Crc = Crc ^ (*(src+i) & 0x00FF); 239 for(int bit=0; bit<8; bit++) 240 { 241 if((Crc & 0x0001) == 0x0001) 242 Crc = (Crc>>1) ^ 0xA001; 243 else 244 Crc = Crc>>1; 245 } 246 } 247 } 248 249 250 /*---*/ 251 // Recieve length calcuration 252 /*---*/ 253 void RRA_LengthCalc(BYTE fc) 254 { 255 switch(fc) 256 {	포트 접속 루틴 포트이름 지정(95/98과 NT/2000/XP의 경우 포트이름 지정방법이 다르므로 API 설명서를 참조한다.) 본예제에서는 NON_OVERRAPPED 비동기식 방식을 선택하였음. 쓰레드를 사용하여 통신프로그램을 구현할경우 오버랩드 타입으로 지정하여야 함. 지정된 포트가 사용할 버퍼크기를 지정. 포트상태 초기화. 통신 타임아웃값을 설정. 자세한 내용은 API 설명서를 참조한다. 보드레이트, 패리티비트, 스톱비트등 포트 파라미터 설정. 재설정시 까지 현재 상태 유지됨. 보드레이트는 가변가능. 고정: Plus 통신규약 임 고정: Plus 통신규약 임 고정: Plus 통신규약 임 CRC 테이블에 의존하지 않은 고전적인 CRC-16 계산 루틴 Q 전송시 PLC로부터 수신될 개수는 결정되는데 이것을 계산하기 위한 루틴임.

프 로 그 램	설 명
257 case FC_READ_WORD_2STEP: 258 Rx_Data.Size = 6+Tx_Data.Data[6]*2; //6=4(header)+2(Crc) 259 break; 260 261 case FC_WRITE_WORD_2STEP: 262 Rx_Data.Size = 7; //7=4(header)+1(info)+2(Crc) 263 break; 264 265 default: 266 break; 267 } 268 269 }	PLC 가 송신하게 될 데이터 내용과 길이는 통신규약을 참조한다. PLC 가 송신하게 될 데이터 내용과 길이는 통신규약을 참조한다.
270 271 /*----------*/ 272 // N-plus protocol establish 273 /*----------*/ 274 BYTE StateMachine() 275 {	통신 프로토콜 구현부
276 bool bRet; 277 278 //-----Q 279 bRet = Q_Process(); 280 if(!bRet) { printf("TX error!\n"); 281 return CS_ERR; 282 } 283 printf("%d bytes transmit success!\n", Tx_Data.Size); 284 285 //-----RRA 286 bRet = RRA_Process(); 287 if(!bRet) { printf("%d bytes received error!\n", Rx_Data.Size); 288 return CS_ERR; 289 } 290 printf("%d bytes received success!\n", Rx_Data.Size); 291 292 //-----packet verify 293 bRet = VerifyFrame(); 294 if(!bRet) { printf("Received packet informaton invalid!\n"); 295 return CS_ERR; 296 } 297 printf("Received packet informaton success!\n"); 298 299 return CS_END; 300 }	Q 과정 RRA 과정 수신데이터 검사 과정
301 302 /*----------*/ 303 // Packet verify 304 /*----------*/ 305 bool VerifyFrame() 306 {	수신데이터 검사부
307 DWORD rxCrc; 308 309 310 rxCrc = (DWORD)(Rx_Data.Data[Rx_Data.Size-2] + 311 Rx_Data.Data[Rx_Data.Size-1]*0x100); 312 CRC16(Rx_Data.Data, Rx_Data.Size-2); 313 314 if((Tx_Data.Data[0]==Rx_Data.Data[1]) && 315 ((Tx_Data.Data[2]+0x80)==Rx_Data.Data[2]) && 316 (Crc==rxCrc)) 317 return true; 318 else 319 return false; 320 }	프레임의 끝 2바이트에 실려있는 CRC값과 수신된 데이터의 CRC16값을 계산한 결과가 일치하면 수신된 데이터가 유효하다고 볼수 있음. 또한, 수신된 데이터가 Q에서 지정한 PLC로부터 온 데이터 인지를 RRA의 헤더부를 검사하면 알수 있다. (RS-485 통신을 하는 경우, 여러대의 PLC가 연결되어 있으므로 반드시 보낸쪽 ID를 검사하여야 한다)
321 322 323 /*----------*/ 324 // Sent and Received packet data display 325 /*----------*/ 326 void DsplPacket() 327 {	
328 int i; 329 330 printf("TxPacket:"); 331 for(i=0; i<Tx_Data.Size; i++) 332 printf("%02X ",Tx_Data.Data[i]); 333 printf("\nRxPacket:"); 334 335 for(i=0; i<Rx_Data.Size; i++) 336 printf("%02X ",Rx_Data.Data[i]); 337 printf("\n"); 338 }	송수신 데이터 표시부

● 통신 프로그램 예제 (Header File)

프 로 그 램	설 명
`` 1 // 2 // NpDemo.h : NpDemo source Header file 3 // 7 // 8 9 #include <windows.h> 10 11 //-----DEFINITION PART------------------------- 12 const unsigned int MAX_FRAME_LEN = 262; 13 const unsigned int MY_PURGE_EVENT = PURGE_TXABORT | PURGE_RXABORT | 14 PURGE_TXCLEAR | PURGE_RXCLEAR; 15 const BYTE CS_RDY = 0; 16 const BYTE CS_ING = 1; 17 const BYTE CS_END = 2; 18 const BYTE CS_ERR = 3; 19 20 const BYTE FC_READ_BIT_2STEP = 0x21; 21 const BYTE FC_WRITE_BIT_2STEP = 0x22; 22 const BYTE FC_READ_WORD_2STEP = 0x23; 23 const BYTE FC_WRITE_WORD_2STEP = 0x24; 24 const BYTE FC_READ_BITWORD_2STEP = 0x25; 25 const BYTE FC_WRITE_BITWORD_2STEP = 0x26; 26 27 struct Tx_RxData 28 { 29 public: 30 BYTE Data[MAX_FRAME_LEN]; 31 WORD Size; 32 }; 33 34 35 //-----VARIABLE PART-------------------------- 36 unsigned int Crc, CrcOld; 37 HANDLE hPort; 38 Tx_RxData Tx_Data,Rx_Data; 39 40 //-----FUNCTION PART-------------------------- 41 bool OnlineConnection(int PortNumber); 42 void CRC16(BYTE *src, int count); 43 bool Q_Process(); 44 bool RRA_Process(); 45 void RRA_LengthCalc(BYTE fc); 46 BYTE StateMachine(); 47 BYTE Test_Read(); 48 BYTE Test_Write(); 49 bool VerifyFrame(); 50 void DsplPacket(); ```	**262: N-plus 통신 규약상 한번에 송수신되는 최대 바이트 수임.** **통신 준비 상태** **통신중인 상태** **통신 완료 상태** **통신 에러 상태** **2단계통신용 비트 읽기 코드** **2단계통신용 비트 쓰기 코드** **2단계통신용 워드 읽기 코드** **2단계통신용 워드 쓰기 코드** **2단계통신용 비트/워드 혼합 읽기 코드** **2단계통신용 비트/워드 혼합 쓰기 코드** **데이터 수신을 위한 구조체 선언**

1-5. 10진수 /BIN /HEX /BCD /Gray 코드 대응표

10진수 (Decimal)	16진수 (Hecadecimal)	BIN 2진수 (Binary)	BCD 2진화 10진수(4자리) (Binary Coded Decimal)	Gray 코드
0	0000	00000000 00000000	0000 0000 0000 0000	0000 0000 0000 0000
1	0001	00000000 00000001	0000 0000 0000 0001	0000 0000 0000 0001
2	0002	00000000 00000010	0000 0000 0000 0010	0000 0000 0000 0011
3	0003	00000000 00000011	0000 0000 0000 0011	0000 0000 0000 0010
4	0004	00000000 00000100	0000 0000 0000 0100	0000 0000 0000 0110
5	0005	00000000 00000101	0000 0000 0000 0101	0000 0000 0000 0111
6	0006	00000000 00000110	0000 0000 0000 0110	0000 0000 0000 0101
7	0007	00000000 00000111	0000 0000 0000 0111	0000 0000 0000 0100
8	0008	00000000 00001000	0000 0000 0000 1000	0000 0000 0000 1100
9	0009	00000000 00001001	0000 0000 0000 1001	0000 0000 0000 1101
10	000A	00000000 00001010	0000 0000 0001 0000	0000 0000 0000 1111
11	000B	00000000 00001011	0000 0000 0001 0001	0000 0000 0000 1110
12	000C	00000000 00001100	0000 0000 0001 0010	0000 0000 0000 1010
13	000D	00000000 00001101	0000 0000 0001 0011	0000 0000 0000 1011
14	000E	00000000 00001110	0000 0000 0001 0100	0000 0000 0000 1001
15	000F	00000000 00001111	0000 0000 0001 0101	0000 0000 0000 1000
16	0010	00000000 00010000	0000 0000 0001 0110	0000 0000 0001 1000
17	0011	00000000 00010001	0000 0000 0001 0111	0000 0000 0001 1001
18	0012	00000000 00010010	0000 0000 0001 1000	0000 0000 0001 1011
19	0013	00000000 00010011	0000 0000 0001 1001	0000 0000 0001 1010
20	0014	00000000 00010100	0000 0000 0010 0000	0000 0000 0001 1110
21	0015	00000000 00010101	0000 0000 0010 0001	0000 0000 0001 1111
22	0016	00000000 00010110	0000 0000 0010 0010	0000 0000 0001 1101
23	0017	00000000 00010111	0000 0000 0010 0011	0000 0000 0001 1100
24	0018	00000000 00011000	0000 0000 0010 0100	0000 0000 0001 0100
25	0019	00000000 00011001	0000 0000 0010 0101	0000 0000 0001 0101
26	001A	00000000 00011010	0000 0000 0010 0110	0000 0000 0001 0111
27	001B	00000000 00011011	0000 0000 0010 0111	0000 0000 0001 0110
28	001C	00000000 00011100	0000 0000 0010 1000	0000 0000 0001 0010
29	001D	00000000 00011101	0000 0000 0010 1001	0000 0000 0001 0011
30	001E	00000000 00011110	0000 0000 0011 0000	0000 0000 0001 0001
31	001F	00000000 00011111	0000 0000 0011 0001	0000 0000 0001 0000
32	0020	00000000 00100000	0000 0000 0011 0010	0000 0000 0011 0000
63	003F	00000000 00111111	0000 0000 0110 0011	0000 0000 0010 0000
64	0040	00000000 01000000	0000 0000 0110 0011	0000 0000 0110 0000
255	00FF	00000000 11111111	0000 0010 0101 0101	0000 0000 0110 0000

1.6 ASCII 코드표

b8	b7	b6	b5	b4	b3	b2	b1	R \ C	0	1	2	3	4	5	6	7
								b8								
								b7	0	0	0	0	1	1	1	1
								b6	0	0	1	1	0	0	1	1
								b5	0	1	0	1	0	1	0	1
				0	0	0	0	0	NUL	DEL	SPACE	0	@	P	`	p
				0	0	0	1	1	SOH	DC_1	!	1	A	Q	a	q
				0	0	1	0	2	STX	DC_2	"	2	B	R	b	r
				0	0	1	1	3	ETX	DC_3	#	3	C	S	c	s
				0	1	0	0	4	EOT	DC_4	$	4	D	T	d	t
				0	1	0	1	5	ENQ	NAK	%	5	E	U	e	u
				0	1	1	0	6	ACK	SYN	&	6	F	V	f	v
				0	1	1	1	7	BEL	ETB	'	7	G	W	g	w
				1	0	0	0	8	BS	CAN	(	8	H	X	h	x
				1	0	0	1	9	HT	EM	)	9	I	Y	i	y
				1	0	1	0	A	LF	SUB	*	:	J	Z	j	z
				1	0	1	1	B	VT	ESC	+	;	K	[	k	{
				1	1	0	0	C	FF	FS	‘	<	L	\	l	|
				1	1	0	1	D	CR	GS	-	=	M	]	m	}
				1	1	1	0	E	SO	RS	.	>	N	^	n	~
				1	1	1	1	F	SI	US	/	?	O	_	o	DEL

Publication NX70-UM003A-70P-KO – December 2010

알에스오토메이션주식회사

www.rsautomation.co.kr

경기도 평택시 진위면 청호리 진위산업단지 32-1-1 블록 알에스오토메이션빌딩 # 451-862

T 031-685-9300, F 031-685-9500

부산 지사 부산광역시 사상구 괘법동 578 산업용품유통상가 27동 203호 #617-726
T 051-319-2890, F 051-319-2894

대구 지사 대구광역시 북구 산격2동 1665번지 전기재료관 다동 223호 #702-717
T 053-944-7783, F 053-944-7784

광주 지사 광주광역시 광산구 우산동 1589-1 광주무역회관 10층 #506-721
T 062-945-8408, F 062-945-8670

알에스오토메이션 서비스센터

전국 어디서나 **1588-5298**

동탄 센터 경기도 화성시 동탄면 청계리 401-12번지 # 445-811
T 031-373-3744, F 031-372-6446

안양 센터 안양시 동안구 호계2동 894번지 피카빌딩 2층 #431-836
T 031-455-8686, F 031- 455-8656

광주 센터 광주광역시 광산구 우산동 1589-1 광주무역회관 10층 #506-721
T 062-945-8665, F 062-945-8664

부산 센터 부산광역시 사상구 괘법동 578 산업용품유통상가 27동 103호 #617-726
T 051-319-1802/3, F 051-319-1834

RS Automation Co., Ltd.

www.oemax.com

RS Automation Building, 32-1-1 Block, Jinwi Industrial Complex, Cheongho-ri, Jinwi-myeon, Pyeongtaek-si, Gyeonggi-do, Korea, zip code : 451-862

T 82-31-685-9300, F 82-31-685-9500

RS Automation Global Business Support
rsagbs@rsautomation.co.kr

韩国京畿道平泽市振威面清湖里振威工业园32-1-1区RS自动化大厦 邮编：451-862

T 82-31-685-9300, F 82-31-685-9500

RS自动化全球商户支持
rsagbs@rsautomation.co.kr