

**RS** OEMax

자동화 설계를 위한

# PLC X8 Series 입문

교육용 매뉴얼

## 일러두기

이 문서에서 설명하는 제품은 다양한 설비에 사용될 수 있습니다. 본 제품의 적용 설비가 다양하므로 본 제품의 모든 사용자와 관리자는 사용하고자 하는 설비와 관련된 운전 및 안전상의 제 법령 및 규정과 준수사항을 반드시 확인하고 지켜야 합니다.

사용자가 본 제품을 사용 및 응용함에 따른 직접 또는 간접적인 손상에 대해서 알에스오토메이션(주)은 그 책임을 지지 않습니다.

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)은 그 책임을 지지 않습니다.

알에스오토메이션(주)은 본 매뉴얼에 포함된 소프트웨어, 장비, 회로 또는 정보 사용에 대해 특허 책임을 지지 않습니다.

알에스오토메이션(주)의 서면 승인 없이 본 매뉴얼의 내용 전부 또는 일부를 복제하는 것은 금지됩니다.

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

<p><b>WARNING</b></p> 	<p>주어진 정보 또는 절차를 잘못 취급할 경우, 사망, 중상 또는 재산상의 손실을 야기할 수 있음을 나타냅니다.</p>
<p><b>IMPORTANT</b></p>	<p>주어진 정보가 제품을 이해하고 활용함에 있어 중요한 정보임을 나타냅니다.</p>
<p><b>ATTEN-</b></p> 	<p>주어진 정보 또는 절차를 잘못 취급할 경우, 경상 또는 재산상의 손실을 야기할 수 있음을 나타냅니다. 그렇지만, 상황에 따라서 심각한 결과를 가져올 수도 있습니다.</p>
<p><b>SHOCK HAZ-</b></p> 	<p>잘못 취급했을 경우 감전에 의한 위험한 상황이 발생하여 사망 또는 중상을 입을 수 있음을 나타냅니다.</p>
<p><b>BURN HAZ-</b></p> 	<p>잘못 취급했을 경우 열에 의한 위험한 상황이 발생하여 사망 또는 중상을 입을 수 있음을 나타냅니다.</p>

# 매뉴얼 개정이력

---

매뉴얼 개정이력에서는 매뉴얼 버전 변경시마다 해당 변경 내용에 대한 간단한 설명을 살펴볼 수 있습니다.

매뉴얼 버전	변경 내용	변경 년/월
A	초판	2014년 9월
B	기능내용 업그레이드	2015년 5월
c	기능내용 업그레이드	2017년 12월



	일러두기.....	ii
	이 매뉴얼의 사용자.....	P-1
	참고 자료.....	P-1
<b>1장.</b>	<b>자동화 시스템.....</b>	<b>1-1</b>
	자동화 시스템 개요.....	1-2
	컴퓨터 통합 생산공정과 네트워크.....	1-2
	자동화의 역사와 목적.....	1-4
	자동화의 역사.....	1-4
	자동화의 정의.....	1-4
	자동화의 5대 요소.....	1-5
	환경변화와 자동화.....	1-5
	자동 제어의 효과.....	1-5
	자동화를 촉진하는 요소.....	1-5
	자동화와 생산비용의 관계.....	1-6
	자동화의 단점.....	1-6
	저투자성 자동화.....	1-6
	계장제어.....	1-6
	계장시스템의 목적.....	1-8
	계장제어의 역사적 배경.....	1-9
	계장 시스템의 역사.....	1-10
<b>2장.</b>	<b>시퀀스 제어.....</b>	<b>2-1</b>
	시퀀스 제어의 종류.....	2-2
	릴레이 시퀀스.....	2-2
	무접점 시퀀스.....	2-2
	시퀀스.....	2-2
	주요 시퀀스 회로 예.....	2-2

자기유지회로 ..... 2-2

인터록(Interlock) 회로 ..... 2-3

### 3장.

## ***X8 PLC 하드웨어*..... 3-1**

PLC의 구성요소 ..... 3-2

PLC의 종류 ..... 3-2

블럭타입 PLC ..... 3-2

모듈타입 PLC ..... 3-3

PLC 사용시의 장점 ..... 3-4

PLC 개념의 이해 ..... 3-5

하드와이어드 와 소프트와이어드 ..... 3-5

PLC의 기본적인 개념 ..... 3-5

작동원리 ..... 3-7

릴레이 시퀀스와 PLC 프로그램 차이점 ..... 3-8

PLC의 특징 및 발전 동향 ..... 3-10

PLC의 기능 ..... 3-10

기본연산기능 ..... 3-10

응용 연산기능 ..... 3-10

MICOM 연산 ..... 3-11

SUBROUTINE 연산 ..... 3-11

연산능력 ..... 3-11

MEMORY 용량 ..... 3-12

PLC의 기타 기능 ..... 3-12

PLC의 특징 ..... 3-12

현재 PLC의 발전 동향 ..... 3-12

PLC의 구성 ..... 3-13

프로세서 ..... 3-13

메모리의 내용 ..... 3-14

	전원부 .....	3-15
	입출력부.....	3-15
	아날로그/디지털 변환 유니트.....	3-23
	고속 카운트 유니트.....	3-27
	위치 결정 유니트.....	3-28
	<b>PLC 사용상의 주의점 및 대책.....</b>	<b>3-37</b>
	동작환경.....	3-38
<b>4장.</b>	<b><i>X8 PLC 하드웨어.....</i></b>	<b>4-1</b>
	<b>X8 PLC .....</b>	<b>4-2</b>
	모델.....	4-2
	<b>Network PLC X8 의 미래 네트워크.....</b>	<b>4-5</b>
	<b>일반사양 및 신뢰성 규격 .....</b>	<b>4-6</b>
	<b>PLC X8 외부 구성 .....</b>	<b>4-6</b>
	<b>통신 케이블 .....</b>	<b>4-7</b>
	<b>CPU모듈의 규격사양.....</b>	<b>4-9</b>
	CPU 성능규격.....	4-9
	<b>입력 사양.....</b>	<b>4-10</b>
	<b>출력 사양.....</b>	<b>4-10</b>
	릴레이 및 FET 출력.....	4-10
	릴레이 출력 .....	4-10
	<b>X8 CPU/확장I/O 결선.....</b>	<b>4-12</b>
	아날로그 결선 .....	4-14
	X8-ANALOG 입력 타입 선택 .....	4-14
	X8-ANALOG 출력 타입 선택 .....	4-14
	단자대 구성 .....	4-15
<b>5장.</b>	<b><i>XGPC 운영.....</i></b>	<b>5-1</b>

	<b>XGPC Software 구성 화면 및 기능 설명.....</b>	<b>5-2</b>
	Online Toolbar 정보.....	5-2
	Project Window.....	5-3
	Project Tree Actions .....	5-3
	툴바설명 .....	5-4
	도구 메뉴.....	5-17
	I/O 설정 .....	5-22
<b>6장.</b>	<b><i>데이터</i>.....</b>	<b>6-1</b>
	데이터의 표현(Number System) .....	6-2
	기타 데이터 표현.....	6-2
	진법 변환과 연산.....	6-3
	각 진법의 수체계 .....	6-3
<b>7장.</b>	<b><i>PLC 프로그래밍</i>.....</b>	<b>7-1</b>
	프로그램 언어 .....	7-2
	프로그램 언어.....	7-2
	니모닉 프로그램 .....	7-2
	래더 다이어그램 .....	7-3
	PLC Programming용어의 정의.....	7-3
	PLC의 메모리 번지 구성.....	7-6
	외부입출력 영역(입력 X, 출력 Y).....	7-6
	내부접점 (B: Memory).....	7-6
	정전유지접점 (모든 메모리의 설정에 따라 가능하다)....	7-7
	타이머 (TM).....	7-7
	카운터 (CT).....	7-7
	데이터 워드 (N/L/F).....	7-7
	특수 접점 (SR).....	7-7
	특수 기능 데이터 테이블(SFR) .....	7-7
	주소표현방법 .....	7-9

비트 번지지정 .....	7-9
워드 번지지정 .....	7-9
더블 워드 번지 지정(L,F) .....	7-9
표현방법 .....	7-10
<b>PLC PROGRAMMING .....</b>	<b>7-12</b>
<b>운전 모드(mode)의 개요 .....</b>	<b>7-12</b>
CPU 운전모드란? .....	7-12
RUN 모드(운전상태) .....	7-13
STOP 모드(정지상태) .....	7-13
PAUSE 모드(일시정지 상태) .....	7-13
ERROR 모드(에러상태) .....	7-13
<b>CPU 처리순서 .....</b>	<b>7-13</b>
프로그램 처리순서 .....	7-13
<b>Programming 명령어 종류 .....</b>	<b>7-15</b>
그룹별 X8명령어 .....	7-15
명령어에 대한 설명 .....	7-17
어드레싱 모드 .....	7-17
Indirect Addressing 을 통한 데이터 테이블의 지정 .....	7-19
Indirect Addressing 의 비트 제어 응용 .....	7-19
<b>8장. PLC명령어 .....</b>	<b>8-1</b>
<b>기본명령 .....</b>	<b>8-2</b>
NOC, NCC 명령 .....	8-2
OUT(OUT)명령 .....	8-2
INV명령 .....	8-2
SET연산결과 ON유지 / RST연산결과 OFF유지 .....	8-3
EGR, EGF명령 .....	8-3
ONSR(One Shot Rising), ONSF(One Shot Falling) .....	8-4
타이머(한시동작 순서정지) .....	8-6
카운터 명령 .....	8-12
<b>비교 명령어 .....</b>	<b>8-16</b>

EQ - Compare for Equal, NE - Compare for Not Equal	8-17
GT(Compare for Greater Than), LT(Compare for Less Than)	8-18
GE(Compare for Greater Than or Equal)	8-19
LE(Compare for Less Than or Equal)	8-19
EQM - Mask Compare for Equal	8-20
LIM - Limit Test	8-21

## **수학 및 산술 명령어.....8-22**

수학 및 산술 명령	8-22
수학, 산술 명령에 대한 설명	8-22
부동 소수점	8-24
변환 명령어	8-34
논리 연산 명령어	8-38
데이터 이동 명령어	8-41
데이터 테이블 명령어	8-43
Shift 및 FIFO, LIFO 명령어	8-47

## **Sequencing (순차 제어) 명령어 .....8-54**

프로그램 제어 명령어	8-58
입력 / 출력 명령어	8-62
인터럽트 명령어	8-64

## **9장.**

### **X8 CPU 특수 기능 .....9-1**

#### **고속카운터 HSC기능 (HSC; High Speed Counter) .....9-2**

카운터 기능이란?	9-3
고속 카운터 MODE	9-3
고속 카운터 기능 설정 방법	9-10
고속 출력(PTO / PWM) 기능	9-16
PTO 특수기능 SFR 구성원	9-21

#### **확장 PTO(원점복귀) 기능.....9-24**

(2) 확장 PTO 출력 기능(원점복귀기능)	9-26
--------------------------	------

#### **PWM 출력 기능 .....9-40**

	PWM (Pulse Width Modulation).....	9-40
	PWM 출력 기능 설정 방법.....	9-40
	PWM SFR의 구성원.....	9-40
<b>10장.</b>	<b><i>X8-POSITION CONTROL.....</i></b>	<b><i>10-1</i></b>
	<b>개요.....</b>	<b>10-2</b>
	Target Applications.....	10-2
	Prerequisites.....	10-2
	Test Environment & Equipment.....	10-2
	Related Publication.....	10-2
	<b>Configuration.....</b>	<b>10-3</b>
	설치배선(Wiring).....	10-3
	결선도.....	10-5
	X8-POSITION MODULE Setting.....	10-9
	<b>Example.....</b>	<b>10-13</b>
	X8-PM소프트웨어로 X8-POSITION MODULE을 이용하여 모션제어.....	10-13
	XGPC LADDER 이용하여 CSD5/7 Servo Drive Control... 17	10-17
<b>11장.</b>	<b><i>X8 PLC통신.....</i></b>	<b><i>11-1</i></b>
	<b>PLC 통신.....</b>	<b>11-2</b>
	Target Applications.....	11-2
	실행조건.....	11-2
	Test Environment & Equipment.....	11-2
	Related Publication.....	11-2
	<b>Configuration.....</b>	<b>11-3</b>
	Wiring.....	11-3
	X8 Setting.....	11-4
	<b>Example.....</b>	<b>11-9</b>
	NX750(또는 NX-Alpha Series)를 연결할 경우.....	11-9

NX-CPUp(또는 NX-Plus Series)를 연결할 경우 ..... 11-15

## **12장. X8 CPU ETHERNET DATA SYNC기능..... 12-1**

**개요** ..... 12-2

    목표(Target Applications)..... 12-2

    실행조건 ..... 12-2

    Test Environment & Equipment..... 12-3

    Related Publication..... 12-3

**Configuration** ..... 12-3

    Wiring..... 12-3

    X8 Setting..... 12-3

## **13장. X8 PID AUTO TUNING ..... 13-1**

**개요** ..... 13-2

**프로세스 제어** ..... 13-2

    비례제어 (P 동작)..... 13-2

    적분제어 (I 동작) ..... 13-3

    미분제어 (D 동작) ..... 13-4

**PID 제어란** ..... 13-6

    PID제어 알고리즘..... 13-6

    각동작에서파라미터의효과..... 13-8

**PID 예제** ..... 13-11

## **14장. X8 CSD7N OVER ETHERCAT..... 14-1**

**X8 CPU OS FRN 11.10**..... 14-2

**지원 가능한 EtherCAT Slave**..... 14-2

**장점** ..... 14-3

**CSD7 서보 드라이브**..... 14-4

---

	예제 - CSD7과 사용시 시스템 구성도.....	14-5
	XGPC 설정 개요.....	14-5
	확장 I/O 설정.....	14-7
	MCST 명령어.....	14-9
	MOVB 명령어.....	14-27
	MSG를 사용한 COE SDO 설정.....	14-28
	CPU 분산 처리를 이용한 6 축 이상의 시스템.....	14-32
<b>15장.</b>	<b>예제처리.....</b>	<b>15-1</b>
	예제처리.....	15-2
	참고사항.....	15-14
	펌웨어 업그레이드 방법.....	15-14
	참고문헌.....	31
	참고사항 1. 고정 IP Address가 설정 되어 있지 않은 경우	
	참고사항 2. Network 방화벽 설정(Windows XP)	
	참고사항 3. Network 방화벽 설정(Windows 7)	
	참고사항 4. CSD3, CSD7 - X8 14DDT, 32DDT 결선도	



이 매뉴얼의 목적이 매뉴얼은 자동화 입문자가 X8 PLC Series를 이용하여 자동화 설계를 하기 위해 자세히 설명합니다.

## 이 매뉴얼의 사용자

이 매뉴얼은 X8 PLC Series의 설치 및 배선 연결에 직접 관여하는 엔지니어 또는 기술자와 운영, 현장 유지 관리 및 X8 PLC Series 의 통합에 직접 관여하는 프로그래머를 대상으로 만들어 졌습니다.

## 참고 자료

다음 문서들은 알에스오토메이션 제품들에 대한 추가 정보를 포함하고 있습니다.

매뉴얼	내용
X8 CPU 설치 매뉴얼	X8 PLC CPU 설치 정보
X8 PLC SERIES 사용자 매뉴얼	X8 PLC SERIES사용 정보

당사 대리점 또는 이 사용자 매뉴얼 뒷면에 기재되어 있는 가까운 영업소에 문의해 주십시오. 또는 당사의 아래 홈페이지를 방문하여 필요한 문서를 다운로드 할 수 있습니다.

[www.rsautomation.co.kr](http://www.rsautomation.co.kr)



## 1장. 자동화 시스템

본 장에서는 X8 자동화 시스템 개요에 대해서 설명합니다.

---

자동화 시스템 개요.....	1-2
컴퓨터 통합 생산공정과 네트워크.....	1-2
자동화의 역사와 목적.....	1-4
계장제어.....	1-6

## 자동화 시스템 개요

하나의 완성된 제품이 나오기까지의 과정을 설계과정과 생산과정으로 구분할 수 있다.

설계과정은 제품에 따라 많이 달라질 수 있으나 생산과정은 어느 정도 체계적인 면을 갖추고 있다고 볼 수 있다. 생산기술을 통하여 제품의 생산비를 낮추고 품질을 높이며 생산소요 시간을 단축할 수 있기 때문에 생산기술은 국가경쟁력에 대단히 중요한 역할을 한다.

생산을 자동화하기 위해서는 프로그램 가능 논리 제어기(PLC: Programmable Logic Controller), 로봇(Robot), 컴퓨터 수치 제어기(CNC: Computerized Numerical Controller) 등 여러 가지 자동화기기를 활용해야 하며 이것들을 서로 연계하여 작업을 수행해야 한다.

여러 가지 자동화 기기들을 서로 유기적으로 연계하여 작업을 수행하기 위해서는 이들 자동화기기 사이에 필요한 정보가 교환되어야 하는데, 공장의 규모가 점점 커짐에 따라 교환되어야 하는 정보의 양은 점점 더 많아지고 있다.

특히 관리 및 경영에 필요한 정보의 양이 방대해 지고 있는 실정이다. 따라서 이런 모든 기능을 갖는 컴퓨터 통합 생산공정이 필요하게 된다.

## 컴퓨터 통합 생산공정과 네트워크

오늘날과 같이 제조업계의 치열한 시장경쟁, 소비자의 다양해진 욕구, 작업자의 임금상승 등의 어려운 여건 속에서 살아남기 위해 제조업계는 생산공정을 자동화시키기 위하여 전력을 다하고 있다.

이러한 방안의 하나가 제품의 주문으로부터 설계, 생산, 관리 및 판매 등 제조업에 관련된 모든 업무들을 수행하는데 있어서 컴퓨터의 기능을 이용하고 이들을 수평적, 수직적으로 통합함으로써 생산성을 극대화시키고자 하는 것인데, 이런 시스템을 컴퓨터 통합 생산공정이라고 한다.

컴퓨터 통합 생산공정은 현재 많은 분야에서 사용되고 있고 꾸준히 연구되고 있는 미래의 제조 기술로써 국내외적으로 크게 각광을 받고 있다.

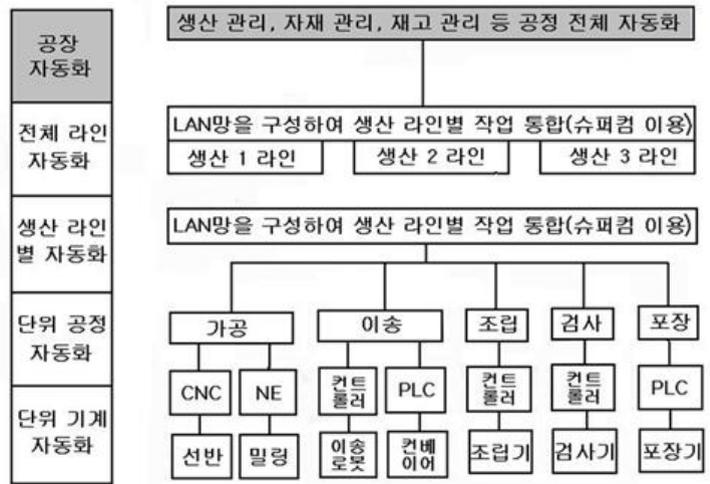
컴퓨터 통합 생산공정은 제조 공정에 컴퓨터를 이용하여 공정 처리에 자동화를 추진하고 이들을 통합하여 전체 공정을 일관되게 관리함으로써, 공정관리에 유연성을 주어 다품종의 소량 생산은 물론 생산 계획과 생산 공정의 급격한 변화에 대하여서도 적절히 대처할 수 있다. 또한 제품의 품질과 생산성을 향상시킬 뿐만 아니라 생산 기간을 단축시키며 고가의 장비들을 공유화하고 노동력 및 불필요한 재고를 감소시킴으로써 생산비를 절감시키는 효과를 가져올 수 있다.

최근에 급격히 이루어지고 있는 컴퓨터의 고기능화 및 저가격화, 통신 네트워크 및 데이터베이스 기술의 발달에 의한 정보 시스템의 진전, 설계 및 개발 소프트웨어의 급격한 발전 등 하드웨어·소프트웨어 양면에서 컴퓨터 통합 생산공정 도입의 환경이 정비됨으로써 컴퓨터 통합 생산공정 도입이 급격히 진전되고 있다.

컴퓨터 통합 생산공정과 같은 대형의 복잡한 시스템을 설계 및 운용하기 위해서는 각각의 공정과 주위 환경에 관련된 방대한 정보의 흐름을 파악할 수 있고, 각각의 요소에 필요한 정보를 적시에 제공할 수 있는 기능이 갖추어져야 한다. 통신 네트워크는 각각의 시스템들을 연결하여 시스템간의 정보 교환을 가능케 함으로써 컴퓨터 통합 생산공정에서의 중추적 역할을 담당하고 있으며 궁극적으로 모든 공정들의 통합을 추구하는 컴퓨터 통합 생산공정에 있어서 가장 핵심이 되는 기술 중의 하나이다.

1 경영정보 경영예측 시스템	2 Level 4
3 인사, 영업관리 시스템	4 Level 3
5 생산(공정) 정보 시스템	6 Level 2
7 제어기 (PLC, DDC 등)	8 Level 1
9 기계(작동기기) 센서(Sensor)	10 Level 0

[그림 1-1] 자동화의 계층구조



[그림 1-2] 자동화의 단계

## 자동화의 역사와 목적

### 자동화의 역사

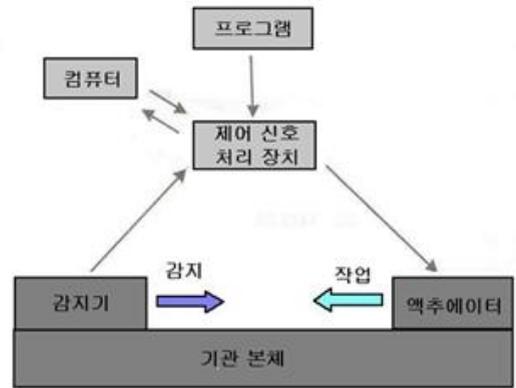
- ① 1920년 : 온도, 압력, 유량, 액면, pH 등의제어 기술
- ② 제2차 대전 중 : 서보 제어 기술→배, 항공기 등의 자동조정 장치
- ③ 전기, 전자, 반도체 -> 대용량의 정보 처리를 신속, 정확하게 처리
- ④ 무인 자동화 공장

### 자동화의 정의

- ① 기계화  
기계가 사람 대신 일을 하는 것 (콤바인, 트랙터 등)
- ② 자동화  
작업의 전부 또는 일부를 사람이 직접 조작하지 않고 컴퓨터 시스템 등을 이용한 기계 장치에 의하여 자동적으로 작동하게 하는 것 (CNC 공작 기계)

## 자동화의 5대 요소

감지기, 제어신호 처리장치, 액추에이터, 소프트웨어 기술, 네트워크 기술



[그림 1-3] 자동화의 5대 요소

## 환경변화와 자동화

- ① 수작업 -> 도구화, 기계화, 자동화
- ② 소품종 소량 생산
- ③ 소품종 대량 생산
- ④ 다품종 소량 생산

## 자동 제어의 효과

- ① 제품의 균일 및 향상.
- ② 생산성 향상.
- ③ 사고의 예방 및 설비의 수명연장.
- ④ 조업의 합리화에 따른 원가절감.
- ⑤ 생산의 증가
- ⑥ 작업원의 노동력감소.
- ⑦ 작업원의 숙련의 불필요

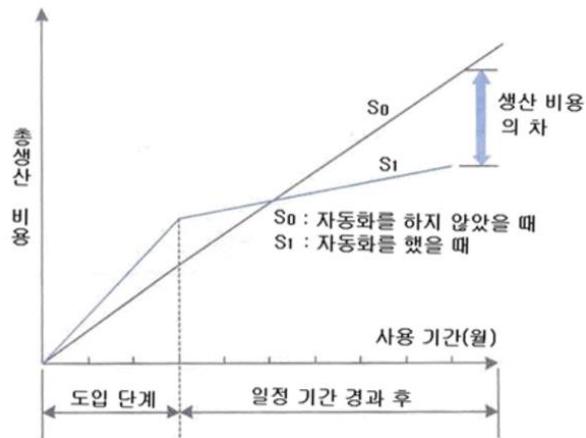


보다 높고 보다 많은 제품을 싸게 생산해서 이익을 높이는 데 목적이 있음

## 자동화를 촉진하는 요소

- ① 작업자의 안전 확보
- ② 3D 업종의 인력 부족
- ③ 기업의 경쟁력 심화
- ④ 소비자 욕구의 다양화
- ⑤ 노사의 이해 대립(일은 적게, 보수는 많이)
- ⑥ 생산 시스템의 거대화

### 자동화와 생산비용의 관계



[그림 1-4] 자동화와 생산비용의 관계

### 자동화의 단점

- ① 초기 설비 투자, 시스템 구축 방법에 따른 운영, 유지비의 변동성
- ② 설치, 운영, 유지, 보수에 높은 기술
- ③ 소비자의 욕구에 따른 제품 수명의 단기성
  - 생산 환경의 변화에 따른 유연성 저하

### 저투자성 자동화

- ① 간단한 원리 적용하고, 스스로 설계
- ② 꼭 필요한 기능만 자동화
- ③ 사용하고 있는 장비 활용하여 간단하게.
- ④ 단계별로 자동화
- ⑤ 회사 내에서 자체적으로 자동화 시스템을 구축
  - 기술적 독립 →유지비 절감

## 계장제어

계장제어는 주로 계장이라는 이름으로 현장에서는 잘 알려져 있으나 그 원류는 자동제어에 속한다. 계장 시스템에서 다루는 분야는 간단히 자동화 시스템이라고도 부를 수 있으나 그 범위는 흔히 대학과정에서 제어이론 정도만 취급하는 자동제어에 비해 매우 포괄적이어서 관련 이론도 넓은 지식이 필요하다.

계장 시스템의 효시는 1950년대 공정계통의 자동화를 목적으로 미국의 ISA(Instrument Society of)가 자동화에 필요한 각종 규격

(Code)과 관련이론 등을 정립하여 제시함으로써 시작되었다. 따라서 계장 시스템의 원어는 Instrumentation이라는 언어로 시작되었고 이를 일본에서 計裝(계기를 설치, 설계하고 관련 설비를 운용하는 기술의 의미)이라는 단어로 번역되어 사용하게 되었다. 그 후 우리나라에서는 이 언어를 그대로 사용함으로써 계장이라는 단어로 자리를 굳혀 일반적으로 사용되게 되었다.

계장관련 기술이 프로세스의 자동화에 대한 관련 분야이므로 그 의미도 매우 포괄적인데 흔히 사용하는 계측(Instrument), 제어(Control), 설치(Installation), 설계(Engineering), 운전(Operation)등의 의미를 포함하게 된다. 이를 간단히 계장(Instrumentation)이라는 단어로 사용하는 일이 많아지게 되어 Instrumentation이라는 언어에는 위에서 언급한 내용들이 포함된 것으로 볼 수 있다.

특히 최근에는 산업시설이 방대하고 그 제어대상도 매우 다양하여 가는 추세에 있어 초기와는 달리 계장 시스템은 공정계통은 물론 자동화에 필요한 각종 설비의 설치, 운전, 설계 엔지니어링, 여러 자동제어 이론의 효과적인 응용 등은 물론 이를 효과적으로 구현하기 위한 컴퓨터의 기술도 필수가 되어가고 있다.

즉, 계장 시스템은 플랜트의 자동화에 필요한 각종 복합적인 기술을 어떻게 도입하여 경제성, 신뢰성, 안정성 등을 확보하면서 가능한 생산성을 증대 시키고자 하는데 목적이 있다고 볼 수 있다. 이들의 목적을 달성하기 위해서 많은 측정설비(센서, 계측장비)는 물론 제어이론과 장비가 이용되어야 하고 프로세스의 특성에 따라 적절한 장비를 선정하고 설계 할 수 있는 엔지니어링 기술, 설치된 모든 장비를 효과적으로 운용 할 수 있는 운용기술도 포함하게 된다. 특히 최근에는 컴퓨터를 이용한 계장 시스템의 구현이 일반화되고 있어 실질적인 공정제어를 구현하기 위해서는 다양한 기술이 필요하다.

이와 같은 계장 시스템의 효시를 굳이 언급한다면 그 개념은 아주 오랜 농경문화 시절부터 인간의 중심 생활에 아주 깊숙이 이용되어 온 것으로 볼 수 있으나 이의 개념 및 이론이 산업시설에 이용되기 시작한 것은 산업혁명 시절부터 보는 것이 좋다. 계장 시스템의 자동화에 대한 이론은 자동제어 이론에 근간을 두게 되므로 자동화에 대한 이론적 배경은 자동제어이론과 같다고 볼 수 있다.

선진 산업국인 미국, 일본, 유럽 등은 자동제어의 중요성을 인식하여 관련 시스템에 대한 많은 연구를 연구소와 대학이 공동으로 기초, 응용, 실험을 다방면에 걸쳐 이룩하였고, 최근에는 현대 제어 이론에 의한 연구가 단순히 이론으로서만이 아닌 인간지능에 대한 적용성 여부 문제를 활발히 연구 중에 있다. 그러나 이론적 배경만을 가지고는 플랜트의 자동화 구현은 매우 어렵다. 즉, 플랜트를 자동화하기 위해서는 이에 필요한 각종 설비 즉, 밸브, 센서, 유압장치, 로봇 기구 등은 물론 이들을 관리하고 제어하기 위한 각종 제어기 및 컴퓨터 등이 필요하게 된다.

이들 장비들이 복합적으로 구성되어 하나의 자동화 시스템이 이루어지게 되므로 이들을 어떻게 설치하고 운용하느냐에 따라 그 효과도 매우 다르게 되므로 이들의 설치 및 운용 기술도 매우 중요하게 된다.

따라서, 계장시스템은 이들을 설계하고, 운용하는 기술까지도 포함하는 것이 보통이다. 그러므로 계장설비를 효과적으로 공부하기 위해서는 다양한 관련기술과 이론적 배경이 요구된다.

## 계장시스템의 목적

최근에 산업이 급격히 발달하면서 계장 시스템이라는 단어가 각 요소에 흔히 인용되고 있다.

그러면 이러한 계장 시스템이란 무엇인가?

이와 같은 물음에 대한 것은 한마디로 정의를 내리기는 어려우나 우리들 신체의 각 기관이 자연환경이나 외부의 사회적 변화에 대해 정신적, 육체적으로 균형을 이루어 원활한 사회활동을 할 수 있도록 각각의 기관의 특성, 서로의 관계 등을 파악하여 원활한 사회활동을 할 수 있도록 하는 것과 같다고 비교 할 수 있다.

이러한 균형을 유지하기 위해 우리의 신체에 있는 모든 기관들은 매우 복잡한 자동제어(균형유지)를 수행하게 되는데 만일 이러한 균형이 깨져 병이 난다든가 또는 암과 같은 불치의 병이 걸리게 되면 우리는 원만한 사회 활동을 할 수 없게 된다.

우리의 사회에 있어서도 이와 같아 계장 시스템은 모든 분야 즉, 경제, 사회, 문화, 군사, 정치 등에 이용 할 수 있다고 볼 수 있다.

이들 분야에 있어서는 모두 계획(예상), 실행(과정), 궤도수정(개선 등)의 과정을 거치면서 적절한 균형을 이루어가고 있기 때문에 우리들은 거대한 자동제어 집단의 한 요소(Element)로 작용하면서 살아가고 있다고 볼 수 있다.

이와 같이 자동제어의 범위는 대단히 넓으므로 이들을 모두 언급한다는 것은 매우 어렵고 그 의미도 없으므로 본 교재에서는 공학적인 면만을 다룬다.

이러한 계장제어의 목적은 역사적 변천과정과 매우 밀접한 관계가 있어 목적도 역시 변천에 따라 다소 변하고 있다. 즉, 초기의 계장제어는 인간의 힘으로는 할 수 없는 것을 자연이나 다른 기계적인 힘으로 해결 하고자 하는 것이 주요 목적이었으나 산업혁명이 일어나고 부터는 실질적인 계장제어의 목적인 정밀조작을 위해 계장제어의 개념이 대폭적으로 도입되었다.

최근에는 고전적인 계장시스템의 개념에서 탈피하여 인간의 사고력을 도입하기 위한 각종 제어알고리즘(퍼지제어, 신경망 제어, 혼돈이론 제어)은 물론 앞의 개요에서 설명한 바와 같은 목적을 위해 다양한 기술(계측, 설치, 센싱, 제어, 운용 등)을 이용하여 자동화를 구현하는데 계장 시스템의 목적이 있다고 하겠다.

## 계장제어의 역사적 배경

계장제어가 인류와 같이 사용되기 시작한 것은 인류 문명의 발달과 거의 일치하는 것으로 알려지고 있으나 그 물리적 특성과 수학적 개념이 도입된 것은 훨씬 뒤의 일이다.

특히 고대 히랍 국가나 이집트의 피라미드 축조시는 대규모 인원과 어떤 종류의 기계 즉, 활차와 같은 것을 이용한 것으로 생각되는데 이러한 것도 원시적이기는 하나 계장설비의 한 단계라 할 수 있다.

그 후 인류가 화석연료를 이용하면서부터 기계공학이 적극적으로 이용되어 자동제어도 활발히 이용되기 시작했다 볼 수 있다.

고대의 나일강에서 배를 이용해 하구의 목적지까지 운반할 때 풍향에 따라 배의 운반속도, 방향을 조정했던 것이 고대 자동제어의 시초로 언급되고 있다.

학문적으로 최초로 언급된 것은 Norber Wiener가 사용한 Cybernetics로 이는 히랍어로 타수(舵手)의 의미로 배의 선수를 조정하는 것을 의미한다.

또한 관개농업에서 유입물의 양과 수위정도, 등잔불의 기름양과 불의 밝기정도 등의 기초적 응용단계를 거쳐 산업계에 최초로 이용되기 시작한 것은 1700년대의 증기기관차에 이용된 것으로 보는 것이 일반적이다.

일본의 경우는 앞서 설명한 ISA의 기술을 곧 받아들여 다양한 기술을 발전 시켜 독립적인 학회의 설립은 물론 국가고시(계장기사)를 배출하고 있으나 우리나라는 아직 미흡한 실정이다. 그러나 최근 이에 대한 관련 기술과 인적 배출이 요구되어 사회적으로 고무되어 가고 있음은 매우 다행한 일이라 할 수 있다.

## 계장 시스템의 역사

계장 시스템에 대한 역사는 공정제어의 역사와 같으나 화학프로세스의 제어가 1940년대까지는 주로 수동식 제어가 이루어져 왔고 따라서 운전원이 플랜트의 많은 변수를 직접 감시 했다.

1950년대 초반까지 이들의 재래식 방법에 의한 자동제어의 기술을 이용해 고정밀 대용량등을 구현하기 위해 발전이 이루어져 왔고 1960년대는 이론적인 면이 발전하여 프로세스의 동특성 분석 및 제어이론을 프로세스에 적용하고자 하는 연구가 수행되었고 1970년대는 대량생산 저렴화의 생산이 주요 목적이 되어 왔다.

1980년대 이후는 각종 제어이론과 컴퓨터의 발달로 분산제어를 수행 함은 물론 일반적인 계획, 생산, 자료전송 및 저장 등 많은 분야를 컴퓨터에 의존하면서 총합제어(Total control)을 수행하고 있다.

이와 같이 플랜트는 다양하므로 그 구현 방법이 매우 다양하여 어떠한 플랜트에는 어떤 제어 알고리즘과 계장설비가 꼭 좋다는 정의를 내릴 수 없다.

따라서 플랜트의 특성과 경제성, 안정성, 신뢰성 등을 종합적으로 검토하여 가장 적합한 장비와 이론을 플랜트에 적용 할 수밖에 없다.

일반적인 경우는 제어 이론면에서도 다양한 이론이 시도는 되고 있으나 계장에서는 경제성과 실질적인 플랜트의 운전에 목적을 두게 되므로 피드백 및 피드 포워드 제어, PID제어 등이 이용되고 있다.

한편, 최근에는 퍼지제어, 신경망 제어, 유전자알고리즘 제어 가 오스 이론을 이용한 제어 등, 다양한 형태의 제어이론을 프로세스에 많이 이용하여 가는 추세이다. 그러나 실 플랜트에서는 최근에 연구된 많은 것들의 종류에도 불구하고 아직도 약 90[%]이상이 고전제어의 대표격인 PID제어기를 이용하거나 이를 변형시킨 형이 이용되고 있는 실정이다.

계장 시스템에 필요한 요소 기술 앞서 설명한 바와 같이 계장설비를 구성하고 운용하기 위해서는 다양한 기술이 요구되므로 이의관련 요소기술도 여러 가지가 필요하다.

따라서 플랜트에서 적용되고 있는 계장시스템에 대한 요소 기술을 공부하기 위해서는 다음과 같은 부분에 대한 기초 이론이 필수적이다.

- ① 기초이론(재료, 물리, 기계, 유체역학, 자기학, 전자, 디지털 등)
- ② 센서기술(온도, 압력 레벨, 유량, 반도체, 성분, 가스, 습도 등)
- ③ 제어이론(고전적인 PID이론, 자동제어 기초, 다변수제어, 캐스케이드 제어, 퍼지제어, 신경망제어, 유전적 알고리즘제어, 현대제어 등)
- ④ 컴퓨터 기술(H/W, S/W, 센서와 인터페이스기술, 컴퓨터간의 인터페이스 기술, 액츄이에터와의 인터페이스 기술)
- ⑤ 통신기술(컴퓨터 통신, 신호전송 등)



## 2장. 시퀀스 제어

본 장에서는 시퀀스 제어에 대해서 설명합니다.

---

시퀀스 제어의 종류.....2-2

주요 시퀀스 회로 예.....2-2

## 시퀀스 제어의 종류

### 릴레이 시퀀스

주위 온도나 써어지 전압에 대한 내력특성은 좋으나 소비전력이 크고 접점동작 속도가 느리고, 진동 및 충격 등에 약하며 접점에 수명이 있어서 고장이 많다.

### 무접점 시퀀스

제어 회로에 사용되는 소자는(IC, Diode, Transistor, SCR 등) 동작 속도가 빠르고 정밀하며 수명이 길다.

진동, 충격에 강하고 장치가 소형화되지만 주위온도에 민감하고 써어지 전압 발생 시 오작동의 우려가 있으며 동작확인이 어렵다.

### 시퀀스

PLC 시퀀스는 릴레이 시퀀스에서 사용하는 릴레이, 타이머, 카운터 등의 기능을 반도체를 사용하며 조립한 소형컴퓨터라고 생각하면 된다.

손쉽게 프로그램을 바꿀 수 있으며 공정단축 및 제어반의 소형화 등이

가능하나 내부회로에 접근하기 위해서는 전용 로더나 PC에 전용 소프트웨어를 설치해서 사용하기 때문에 보수유지에는 기능적인 것을 많이 요구 한다.

## 주요 시퀀스 회로 예

### 자기유지회로

자기유지회로(self-holding circuit)에는 자기유지를 시켜 주기 위한 ON신호가 자기유지를 해제하기 위한 OFF신호보다 우선하는 ON우선회로(Dominant ON)와 OFF신호가 ON신호보다 우선하는 OFF우선회로(Dominant OFF)의 두 가지가 있다.

#### ON우선 자기유지회로

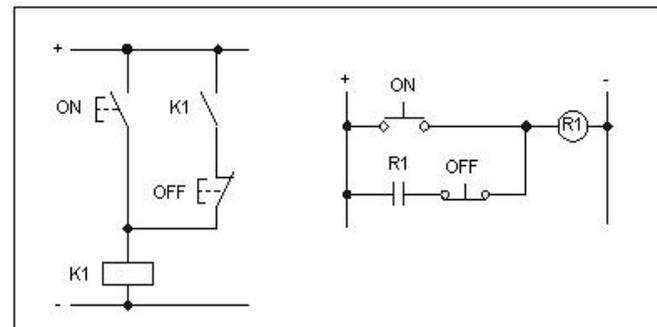
릴레이를 작동시키기 위한 전기신호가 짧은 기간 동안만 존재하다 없어져도 그 릴레이가 계속하여 작동된 상태로 존재하기 위

해서는 그 릴레이가 갖고 있는 접점 중의 하나를 이용하여 자기 자신에게 전기를 공급하여 주면 된다.

아래의 그림에서와 같이 ON 스위치를 작동시키면 릴레이가 작동하게 되고 자신의 접점 K1을 이용하여 릴레이의 코일에 전류가 공급되므로 ON스위치가 원위치 되어도 릴레이 K1은 계속하여 작동된 상태를 유지할 수 있게 된다.

이러한 회로를 자기유지회로라고 하며 릴레이 K1을 OFF시키기 위해서는 OFF스위치를 작동시키면 된다. ON스위치와 OFF스위치를 같이 작동시키면 릴레이가 OFF스위치와는 관계없이 ON스위치에 의하여 작동되므로 이와 같은 자기유지회로를 ON우선 자기유지회로라고 한다.

PLC 에서는 보통 내부 메모리를 이용하여 프로그램으로 작성된다.



[그림 2-1] ON우선 자기유지회로

## 인터록(Interlock) 회로

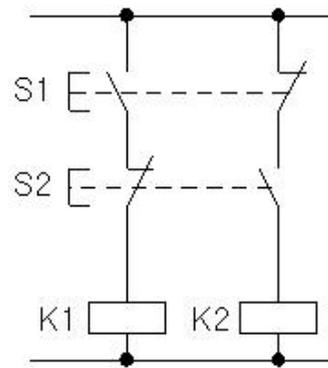
전기 회로에서 인터록 회로라 함은 서로 반대되는 신호가 존재할 때 어느 한 신호가 유효하게 되면 그 반대되는 신호가 더 이상 입력될 수 없도록 인위적으로 차단시켜 주는 회로이다.

이 회로를 구성하기 위해서는 다접점 스위치를 이용하거나 릴레이의 접점을 이용하여 반대 신호를 차단시켜 주게 된다.

또는 다접점 스위치와 릴레이 접점을 이용하여 이중으로 인터록 회로를 구성할 수도 있다.

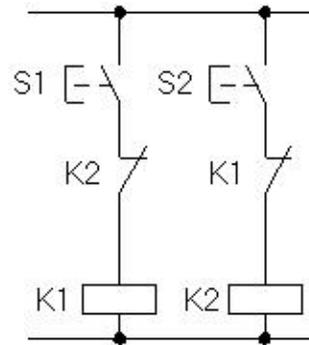
PLC에서는 프로그램을 이용하여 간단하게 구현할 수 있다.

푸시 버튼의 다접점 스위치를 이용한 인터록 회로



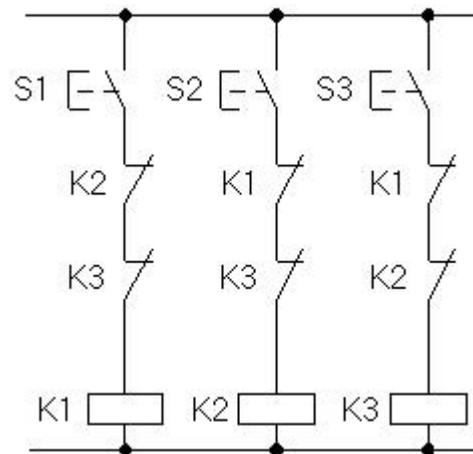
[그림 2-3] 푸시 버튼의 다접점 스위치를 이용한 인터록 회로

릴레이 접점을 이용한 인터록 회로(입력 신호가 두 개인 경우)

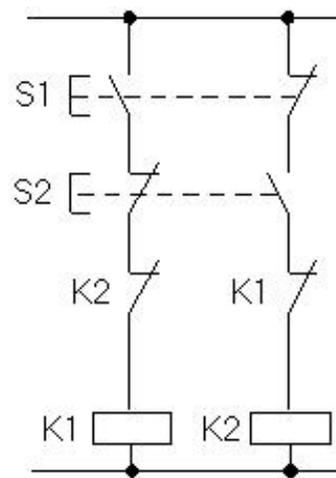


[그림 2-4] 릴레이 접점을 이용한 인터록 회로

릴레이 접점을 이용한 인터록 회로(입력 신호가 세 개인 경우)



[그림 2-5] 릴레이 접점을 이용한 인터록 회로

릴레이와 푸시 버튼을 이용한 이중 인터록 회로

[그림2-6] 릴레이와 푸시 버튼을 이용한 이중 인터록 회로



## 3장. X8 PLC 하드웨어

본 장에서는 PLC제어 설계 및 운영에 대해서 설명합니다.

---

PLC의 구성요소 .....	3-2
PLC의 종류 .....	3-2
PLC 사용시의 장점 .....	3-4
PLC 개념의 이해 .....	3-5
릴레이 시퀀스와 PLC 프로그램 차이점 .....	3-8
PLC의 특징 및 발전 동향 .....	3-10
PLC의 기능 .....	3-10
PLC의 기타 기능 .....	3-12
PLC의 특징 .....	3-12
현재 PLC의 발전 동향 .....	3-12
PLC의 구성 .....	3-13
PLC 사용상의 주의점 및 대책 .....	3-37
PID 예제 .....	13-11

## PLC의 구성요소



[그림 3-1] PLC의 구성요소

## PLC의 종류

PLC는 크기와 형태에 의하여 블록타입과 모듈타입으로 분류하며 대부분 블록타입은 소형 PLC, 모듈타입은 대형 PLC이다.

### 블럭타입 PLC

블록 모듈 타입의 PLC는 PLC 구성도에 있는 모든 기능들이 하나의 케이스 안에 설치되어 있어 소형이다. 즉 전원장치, CPU, 메모리, 입력 및 출력의 모든 기능이 집적되어 있어 취급이 간단하고 저가이다.

그러나 입출력 점수가 제한되고 확장 유닛을 장착하여 입출력 점수를 늘릴 수 있으나 한계가 있다.

또한 PLC간의 통신과 ANALOG신호의 처리, 위치결정과 같은 고도의 기능의 발휘가 어려우며, 로직의 STEP 양도 제한적이고, 상위 컴퓨터와의 작업정보의 전송이나 작업지시 정보의 교환 등이 제한된다.

블록 모듈 타입의 PLC는 간단한 설비의 자동화나 제어 단독 설비의 자동화가 주목적이며 장래의 확장성이 요구되는 설비에서의 채택은 바람직하지 않다.



[그림 3-2] 블록타입 PLC

- 모든 장치가 집적되어 소형
- 시리얼 통신은 가능하나 고도의 통신은 불가
- 아날로그신호처리, 위치결정과 같은 고도의 기능 발휘가불가
- 접점확장이 불가
- 적재 가능한 로직의 스텝수가 제한적
- 저가

### 모듈타입 PLC

모듈타입의 PLC는 베이스유닛에 전원 유닛과 CPU 유닛을 기본으로 설치하고 자동화 대상의 기능에 따라 여러 종류의 입출력 장치를 추가하여 사용한다.

입출력 점수의 변경이 요구될 시는 필요한 기능의 입출력장치를 추가시킬 수 있다. 또한 CPU의 기능이 향상되고 고속처리 및 대용량 데이터의 처리가 가능하며 PLC간의 통신이나 상위 컴퓨터와의 다양하고 고속의 통신이 지원된다.

이와 같은 기능을 이용하여 CNC, ROBOT, 상위 컴퓨터와 연결하여 대단위의 자동화가 가능하며 컴퓨터 통신을 이용하여 수주에서 생산 판매에 이르는 전 과정을 자동화하는 총합화 생산시스템을 구성할 수 있다.

또한 모든 생산 활동이 기계에 의해 움직이는 무인 공장의 등장이 가능 하도록 하였다. 유닛은 입력, 출력, 입/출력 혼용, 통신,

고속 카운터, PID제어, 아날로그 입력, 아날로그 출력 위치 결정 유닛 등이 있다.



[그림 3-3] 모듈타입 PLC



[그림 3-4] 복합형 모듈타입 PLC

- 입출력점수의 변경이 자유롭다
- 고도의 기능을 발휘하는 입출력장치 자유롭게 추가 가능
- 고속CPU의 장착으로 신호의 고속처리 및 대용량의 데이터처리가 가능
- PLC간의 통신이나 상위 컴퓨터간의 다양한 고속통신지원
- 고가

## PLC 사용시의 장점

와이어 LOGIC형의 기존 제어 방법은 상당한 지식과 시간이 필요한 방법이었으나 PLC의 구현으로 시간과 인력구성이 용이하여 설치가 쉬울 뿐만 아니라 프로그램으로 다시 다른 작업을 할 수 있게 하므로 하드웨어 낭비가 거의 없는 것이 장점이다.

- ① 공정을 생략할 수 있고, 기획성이 우수하다.
- ② 높은 제어성을 확보할 수 있다.(마이크로프로세서를 이용)
- ③ 제어반의 크기를 줄일 수 있다.(소형화)
- ④ 에너지를 절약할 수 있다.
- ⑤ 기밀을 유지할 수 있다.

- ⑥ 배선 작업이 간단하다.(제어 계통 배선이 생략되며 프로그램으로 구현)
- ⑦ 신뢰성 및 보수성이 향상된다. (NOISE에 취약한 점이 있지만 전자 부품의 신뢰성 확보와 자가진단 및 모듈 특성향상으로 결점보완이 이루어지고 있음)
- ⑧ 높은 경제성을 실현시켜 준다.
- ⑨ 컴퓨터 네트워킹이 가능하다.

## PLC 개념의 이해

### 하드와이어드 와 소프트와이어드

종래의 릴레이 제어 방식은 일의 순서를 회로도에 전개하여 그곳에 필요한 제어 기기를 결합하여 리드선으로 배선 작업을 해서 요구하는 동작을 실현한다.

이 같은 방식을 하드와이어드 로직(hardwired logic)이라고 한다. 하드와이어드 로직 방식에서는 하드(기기)와 소프트가 한 쌍이 되어 있어 사양이 변경되면 하드와 소프트를 모두 변경해야 하므로, 이것이 갖가지 문제를 발생시키는 원인이 된다. 따라서, 하드와 소프트를 분리하는 연구 끝에 컴퓨터 방식이 개발되었다.

컴퓨터는 하드웨어(hardware)만으로는 동작할 수 없다. 하드웨어 속에 있는 기억 장치에 일의 순서를 넣어야만 비로소 기대되는 일을 할 수가 있다. 이 일의 순서를 프로그램이라 하며 기억 장치인 이 메모리에 일의 순서를 넣는 작업을 프로그래밍이라 한다.

이는 마치 배선작업과 같다고 생각하면 된다. 이 방식을 소프트와이어드로직(soft\_wired logic)이라 하며 PLC는 이 방식을 취하고 있다.

### PLC의 기본적인 개념

1) PLC는 컴퓨터이다.

PLC는 산업체에서 사용하는 제어기라는 생각보다 컴퓨터는 인식을 먼저 할 필요가 있다. 제어기도 역시 컴퓨터이지만, PLC가 작동하는 기본 원리는 컴퓨터를 기본으로 하기 때문에 컴퓨터를 배운다는 인식을 먼저 하는 것이 좋겠다.

2) 컴퓨터는 CPU에 의해 순서대로 하나씩만 명령을 수행한다.

컴퓨터는 여러 가지 일을 동시에 처리할 수 있는 장치가 아니고, 순서(프로그램)에 맞춰 하나씩 처리하는 장치다. 아침에 일어나서 "신문보고 식사하고 난 뒤 화장실가기" 의 순서로 하는 것이지 " 화장실에 앉아서 식사하면서 신문보기"는 할 수 없다.

3) 컴퓨터는 수학을 엄청 잘 한다.(처리속도가 빠르다.)

컴퓨터는 덧셈, 뺄셈과 같은 사칙연산을 비롯하여 AND, OR와 같은 논리 연산은 너무 엄청 빨리 수행한다. 연산 하나 처리하는데, 약 1 $\mu$ Sec정도다. 즉, 1초에 백만번 연산을 할 수 있는 정도다. 이 보다 빠른 PLC도 많다.

4) 컴퓨터가 다루는 수의 세계는 단순하다.

컴퓨터가 사는 세상은 오로지 "0"과 "1"만 있고 2,3,4...는 없다. 2,3,4와 같은 숫자도 "0"과 "1"의 조합으로 표시한다. 0010은 2로, 0011은 3으로, 0100은 4로 표시한다. 그리고, 논리연산에 사용되는 거짓(False)과 참(True)도 "0"과 "1"로 표시한다. "0"은 거짓, "1"은 참이다. PLC제어에서는 "1"과 "0"을 스위치나 램프가 ON과 OFF로 생각한다.

5) PLC는 자동화에 관해서만 강력한 기능을 가진 컴퓨터

일반 컴퓨터가 주로 하는 문서작성이나, 그림 그리기, 음악 만들기 같은 것은 못해도, 공장자동화를 하기 위한 여러 기능들은 어떤 컴퓨터도 흉내를 내지 못 할 정도로 강력한 능력을 가졌다. 자동화에 필요 없는 기능은 과감이 없애고, 오로지 자동화에 필요한 기능들만 갖춘 자동화 전용컴퓨터이다.

6) PLC는 모니터가 없다.

컴퓨터에서는 CPU에서 작업한 내용이 모니터를 통하여 출력되어 우리에게 전달된다. 그 외의 출력장치는 스피커, 프린터 등이 있지만, PLC에서는 여러 전선 단자(端子)를 통하여 전구에 불이 오게 하거나, 모터를 회전시키는 출력을 한다. 따라서 PLC에 연결된 전선을 통하여 공장의 여러 기계를 원하는 때에 작동시킬 수 있다.

7) PLC는 키보드가 없다.

키보드는 컴퓨터에 글자를 입력할 때에 사용된다.

그렇지만, PLC에는 글자를 입력시킬 필요가 없으므로 키보드가 없다. 그 대신 기계를 움직이려는 [START]스위치나 정지하라는 [STOP]스위치 등이 필요하다.

이런 스위치를 접점(接點)이라 한다. 이런 접점 스위치를 연결하는 전선을 PLC의 입력으로 배선하여 키보드를 대신하여 입력장치로 사용한다.

## 작동원리

PLC의 작동은 CPU라는 작업자에 의해서만 모두 처리된다. PLC는 컴퓨터의 일종으로 CPU에 의해 모든 동작이 이루어진다. 따라서 PLC의 작동원리를 이해하고자 한다면 여러분이 CPU가 된 것처럼 모든 작동을 머리로 상상해야 한다.

1) 작업자가 일하는 공간에는 입력부 메모리, 출력부 메모리, 내부 메모리, 작업대가 준비되어 있다.

접점에 의해 입력되는 "1"과 "0"의 정보를 저장하는 메모리를 입력부 메모리라 한다. 접점이 ON되어 있으면, "1"로 인식하고, 접점이 OFF되어있으면, "0"으로 인식한다. (입력접점정의)

2) PLC에서 계산된 결과를 저장하여 외부로 신호를 내 보낼 수 있는 메모리를 출력부 메모리라 한다.

저장된 메모리의 값이 "1"이면, 그 메모리와 연결된 램프에는 불이 들어오고, "0"이면 불이 들어오지 않는다.

램프 대신 모터를 연결해두면, 지정된 출력부 메모리의 값을 1/0으로 함으로써 모터를 작동/정지시킬 수 있다. (출력접점정의)

3) 내부 메모리는 프로그램 연산에 따라 계산된 결과를 출력부 메모리와 같이 저장하는 곳이지만, 외부로 신호를 내보내지 않고 PLC내부에서만 그 결과를 참조한다.

입력되어 있는 프로그램 순서대로 [작업자]가 AND, OR와 같은 연산을 한 스텝씩 수행하면서 그 결과를 임시로 저장하는 곳을 [작업대]라 비유한다.

입력/출력/내부 메모리의 정보를 이 [작업대]로 불러 들이고, 원하는 연산을 한 뒤, 작업대에 있는 연산 결과를 출력/내부 메모리로 보낸다.

결론적으로 모든 정보의 이동은 이 작업대를 통하여 이루어진다.

4) 작업자 CPU는 프로그램을 빠르게 반복 수행한다.

일반 계산프로그램은 입력 데이터를 받고, 원하는 계산을 수행한 후, stop, end를 거쳐 종료된다. 그러나, PLC제어는 RUN 수행하는 동안에는 계속적으로 PLC는 작동하여야 한다.

따라서 [입력] → [프로그램연산] → [출력] (→자기진단) → [입력] → ...을 반복하면서 무한 루프를 돌며 지속적으로 연산 작업을 한다. 한 루프의 대부분 시간은 연산 작업을 한다.

예를 들어 한 스텝 수행하는데 걸리는 시간이  $1\mu\text{sec}/\text{step}$  이고, 프로그램은 1000step이라면, 한 루프를 도는데 걸리는 시간은  $1000\text{step} * 1\mu\text{sec}/\text{step} = 1\text{msec}$ 이다. 이 시간을 1 scan time이라 한다.

결과적으로 1초에 1000번 반복 순환한다.

5) 입력신호를 받아들이는 것도 스캔타임단위(예로서 1msec)로 단속적으로 받아들인다. 입력 메모리를 갱신한 후 프로그램 수행 중에는 비록 입력신호가 바뀌더라도 입력되지 않고, 다음 입력 메모리 갱신 때까지 기다렸다가 입력된다.

출력도 마찬가지로 프로그램 수행중에 여러 번 바뀌더라도 그 순간 순간 출력이 변하지 않고, 마지막 결과만 출력 메모리로 갱신되어 변경된다.

따라서 입력과 출력신호의 변경은 실제 시간보다 약간씩 지연이 발생한다.

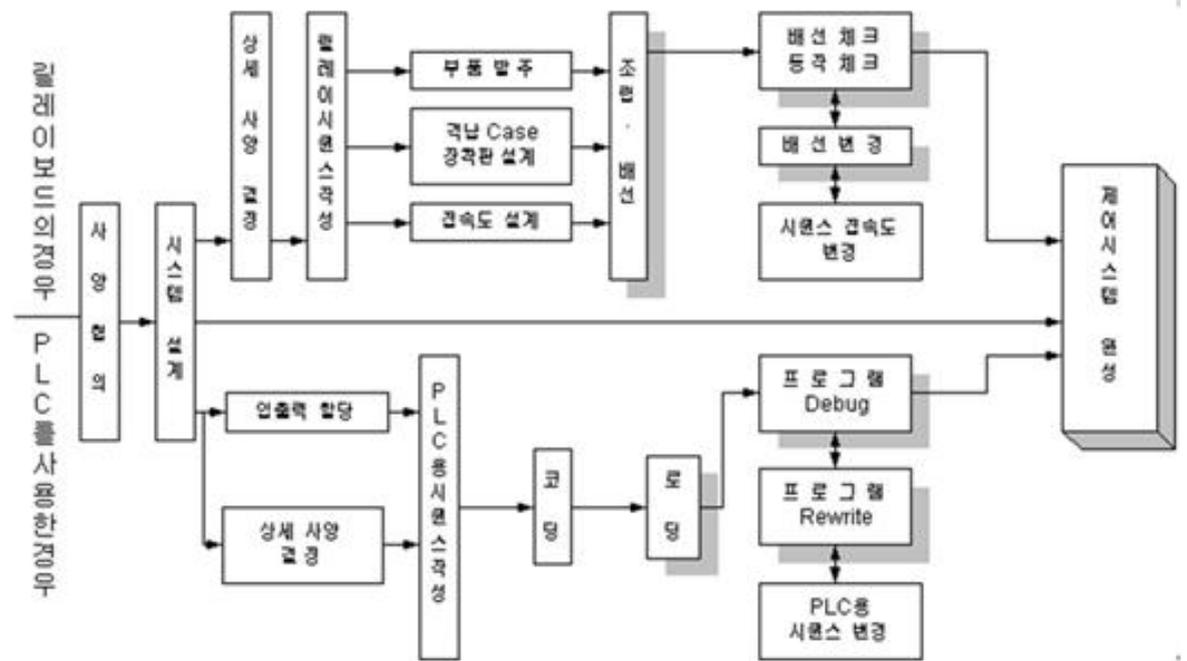
## 릴레이 시퀀스와 PLC 프로그램 차이점

PLC는 LSI등의 전자 부품의 집합으로 릴레이 시퀀스와 같은 접점이나 코일은 존재하지 않으며 접점이나 코일을 연결하는 동작은 소프트웨어로 처리되므로 실제로 눈에 보이는 것이 아니다.

또, 동작도 코일이 여자되면 접점이 닫혀 회로가 활성화되는 릴레이 시퀀스와는 달리 메모리에 프로그램을 기억시켜 놓고 순차적으로 내용을 읽어서 그 내용에 따라 동작하는 방식이다.

제어는 프로그램의 내용에 의하여 좌우된다.

따라서 사용자는 자유자재로 원하는 제어를 할 수 있도록 프로그램의 작성 능력이 요구된다.



[그림 3-5] 릴레이 제어반과 PLC의 설계 순서 비교

구분	PLC	릴레이 제어반
제어방식	프로그램이라는 소프트웨어에 의해 제어되는 소프트 로직	부품간의 배선에 의해 로직이 결정되는 하드 로직
제어기능	릴레이(AND, OR, NOT 등) 업 다운 카운터 시프트 레지스터 산술 연산, 논리연산 전송 (기능은 한정적이고 규모에 따라 대형화)	릴레이(직/병렬에 의한 AND, OR) 타이머 단순한 프리셋 카운터 (고기능, 대규모 제어를 소형으로 실현)
제어요소	유접점 (한정된 수명 속도제어)	무접점 (고신뢰성, 긴수명, 고속제어)
제어내용변경	모든 배선의 철거 및 재시공	프로그램의 변경만으로 가능
보 전 성	보수 및 수리 곤란	고 신뢰성 유지, 보수가 용이함
확장성	시스템의 확장이 곤란	시스템의 확장이 용이하고 컴퓨터와 연결 가능하여 작업정보를 송수신 할 수 있다.
크기	소형화가 곤란	소형화 가능

[표 3-1] PLC와 릴레이 제어반과의 비교

구 분	릴레이 제어반	디지털 로직	컴퓨터	PLC
가격	매우 저가	저가	고가	저가
크기	대형	매우 소형	적당	매우 소형
처리 속도	느림	매우 빠름	매우 빠름	빠름
노이즈	우수	양호	아주 우수	양호
실 장	설계와 설치시 많은 시간 소요	설계시 많은 시간 소요	Program시 아주 많은 시간 소요	간단
복합기능	없다	있다	있다	있다
기능의 변화	매우 어렵다	어렵다	아주 쉽다	아주 쉽다
유지보수	매우 어렵다	어렵다	아주 쉽다	아주 쉽다

[표 3-2] PLC와 기타 제어 시스템의 비교

## PLC의 특징 및 발전 동향

PLC는 일종의 마이크로 컴퓨터(Micro computer)이다. 종래의 릴레이 시퀀스 제어는 각종 유접점 기기를 전선으로 배선하여 제어 내용을 구성하는 하드 와이어드(Hard wired) 방식이었으나 PLC 제어는 프로그램에 의하여 제어 내용을 만드는 소프트 와이어드(Soft wired) 방식이 그 특징이다.

## PLC의 기능

### 기본연산기능

PLC의 기본적인 연산 기능은 다음의 3가지 조건이 있다.

① 접점의 직,병렬 접속에 의해 제어의 지령, 조건, INTERLOCK의 제어 내용을 실현하는 연산으로 가장 기본적인 것이다.

② TIME 연산

제어상의 TIMING을 취하는 시한 요소 연산이다.

③ COUNTER 연산

동작횟수 등을 COUNTER하는 계수 요소의 연산이다. 이들 3개의 연산이 SEQUENCER의 기본 연산으로 유접점 RELAY에 의한 SEQUENCE에 회로와 같은 기능을 실현할 수 있다.

### 응용 연산기능

기본 연산 기능에 덧붙여 다음과 같은 응용적인 연산 기능이 있다.

① MASTER CONTROL

공통 모선의 교환을 시행하는 연산이다.

## ② SHIFT 연산

SHIFT REGISTER 기능의 연산이다.

## ③ PULSE 연산

신호의 상승부를 검출해 PROGRAM의 1주기분의 PULSE를 생성하는 연산이다.

## ④ DATA 연산

수직 DATA를 다루는 연산으로 다음과 같은 것이 있다. (DATA 전송, DATA SHIFT, DATA 입/출력, DATA의 대소 일치의 판정 사칙연산, CODE 변환, DATA의 논리연산, DECODE / ENCODE DATA의 결합 및 분리)

**MICOM 연산**

내장하고 있는 MICRO-PROCESSOR의 연산으로 SEQUENCER로는 불가능한 연산을 실현할 수 있다.

**SUBROUTINE 연산**

반복처리를 시행하는 연산이다.

SEQUENCER의 제어 능력을 결정하는 항목으로서 다음의 종류가 있다.

1. 연산의 능력
2. MEMORY 용량
3. 연산속도
4. 입, 출력 점수
5. 입, 출력 기기와의 접속 능력
6. SYSTEM 구성능력

**연산능력**

연산기능을 어느 만큼 갖고 있는가 하는 것인데 사람에 있어서 두뇌가 얼마만큼 좋고, 나쁜가를 비교할 수 있다.

단순한 유접점 RELAY 대신 SEQUENCE 제어이면 전항의 기본연산 기능으로 충분하다. 그러나 RELAY 제어를 초월한 제어이거나 많은 제어를 행하는 것이면 DATA 연산을 포함한 고기능 연산이 필요하다.

SEQUENCE를 폭 넓게 사용하려면 될 수 있는 한 연산능력이 높은 것을 선택해 주면 좋다.

## MEMORY 용량

어느 정도의 PROGRAM을 기억할 수 있는 가로 제어의 복잡함에 관계한다. 제어능력이 복잡하면 또 제어규모가 크면 PROGRAM 량이 많게 되고 MEMORY의 용량도 크게 되지 않으면 안 된다. MEMORY 용량은 대강 다음의 계산식으로 산출할 수 있다.

$$\text{MEMORY 용량} = \text{입, 출력점수} * K(\text{STEP})$$

## PLC의 기타 기능

- ① 릴레이 시퀀스 회로의 기능을 수행
- ② 인터럽트처리가 가능
- ③ 모니터링이 가능
- ④ 아날로그 데이터의 입출력이 가능
- ⑤ 프로그램 보존기능
- ⑥ 컴퓨터와 링크기능
- ⑦ Soft/Hardware적 자기고장 진단 기능

## PLC의 특징

- ① 기능의 다양화
- ② 조작의 간편성
- ③ 설치의 간편성
- ④ 유지보수의 편리성
- ⑤ 고 신뢰성
- ⑥ 프로그램의 고기능성(제어회로 설계가 용이)

## 현재 PLC의 발전 동향

- ① 고속화, 고기능화, 대용량화
- ② 저가격화, 소형화
- ③ 위치제어, PID제어 등의 피드백 제어기술 실현
- ④ 네트워크 및 데이터 베이스에 의한 시스템화
- ⑤ 고성능명령어 추가
- ⑥ 네트워크기술을 이용한 분산시스템
- ⑦ 멀티테스킹과 병렬 처리
- ⑧ 이중화 및 고장 허용
- ⑨ 표준버스 시스템의 사용 (VME, PC 버스 등)

## PLC의 구성

### 프로세서

프로세서(Processor)는 PLC의 핵심으로서 PLC 모든 동작을 제어하기 위한 중앙연산장치(CPU)부와 실행 프로그램 사용자 프로그램 및 데이터 저장을 위한 메모리부, 외부 입출력 장치를 스캐닝(Scanning)하기 위한 입출력 인터페이스 회로부, 외부장치들과 통신하기 위한 통신회로부가 있고 이외에 Watch dog timer, Real time clock 프로세서 상태표시 회로부 메모리 에러 Check하는 회로부 등의 부수적 회로들이 있다.

### CPU

CPU는 메모리부, 연산 제어부, 입출력 제어부로 구성되어 있다. 특히 연산제어부와 메모리부를 합해서 CPU모듈이라고 한다.

제어용 메모리는 PLC를 동작시키는 데 필요한 프로그램을 기억하는 부분이기 때문에 ROM을 사용한다. 프로그램 메모리는 시퀀스를 기억하는 부분이다. 시퀀스는 변경하는 경우가 있으므로 프로그램을 필요에 따라 마음대로 기억 또는 소멸시킬 수 있는 RAM(주로 SRAM)을 사용한다.

입출력 메모리는 입력데이터나 출력 데이터를 일시에 기억하는 부분이다.

연산제어부는 메모리에서 프로그램 내용이나 데이터를 받아 논리연산이나 산술연산을 한 후 입출력메모리에 그 결과를 보내어 기억시킨다.

입출력 제어부는 입출력 신호의 모으기 또는 보내기의 타이밍 메모리 등의 번지에 입력 데이터를 기억시키거나 또는 어느 번지에서 데이터를 호출할 것인가를 지시, 명령하는 부분이다.

#### CPU의 기능으로는

1. PROGRAM의 실행
2. 주변기기와의 INTERFACE
3. 자기진단 (연산장치 검출, MEMORY 이상검출 전원)

## 메모리의 내용

### 프로그램 메모리

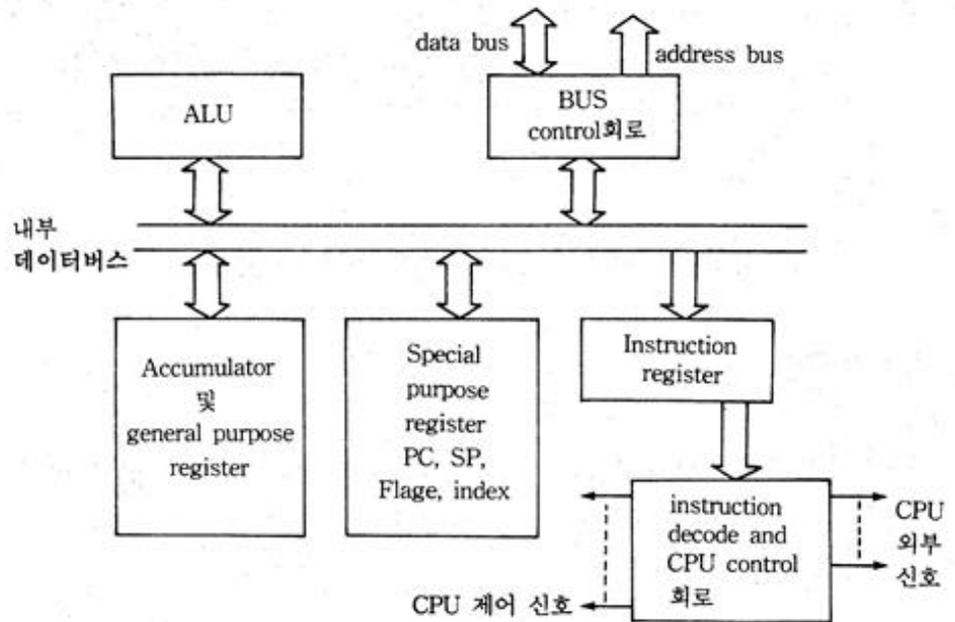
제어하고자 하는 시스템의 사양에 따라 사용자가 작성한 프로그램이 저장되는 영역 (RAM으로 저장했다가 프로그램이 완전하면 ROM으로 운전가능)

### 데이터 메모리

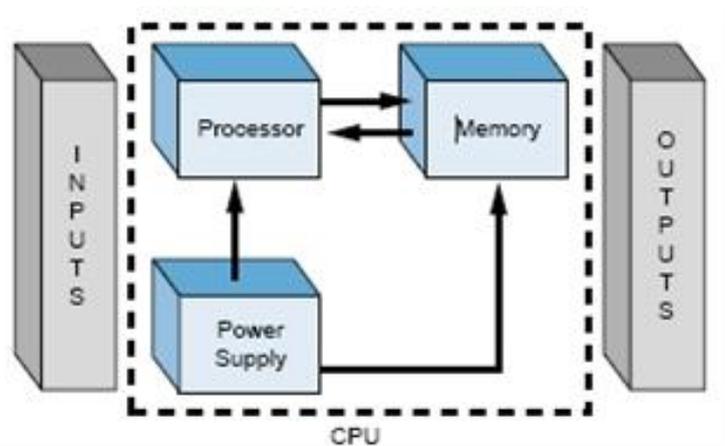
입,출력릴레이, 보조릴레이, 타이머, 카운터 등의 접점상태 및 설정값, 현재값 등의 정보가 저장되는 영역 (정보가 수시로 바뀌므로 RAM을 사용)

### 시스템 메모리

PLC의 제작사에서 작성한 시스템이 저장되는 영역 (PLC 기능과 성능을 결정하는 중요한 프로그램으로 ROM를 사용)



[그림3-7] PLC의 CPU부의 구성 상세



[그림 3-8] PLC의 CPU부의 구성2

## 전원부

PLC는 CPU와 입출력장치 등에 공급되는 전원이 제각기 다르기 때문에 각 장치에 공급되는 전원을 잘 알아야 한다. 이는 제조회사의 MANUAL과 각 주변기기의 전원을 참조하면 된다.

교류 100V 또는 220V의 상용 전원을 PLC에서 사용하는 직류 5V 및 24V의 전압으로 변환하는 장치이며 잡음, 서지를 없앤다. 전원의 필요조건은 다음과 같다.

- ① 교류전원에서의 노이즈 제거, 노이즈에 의한 오동작의 방지
- ② 입출력전압과 내부 구동전압과 절연
- ③ 정전압이어야 한다.
- ④ 전원 고장이 다른 부분에 파급되지 않도록 한다.

대개 PLC 입력전원은 AC 100V,200V이며 IC구동에는 5V, 10V, 24V 등의 직류전압, 입출력 기기와의 인터페이스에는 교류 100V, 200V, 직류100V, 48V, 24V, 12V, 9V, 5V 등의 전압이 사용된다.

## 입출력부

외부기기(신호원)으로부터 신호를CPU에 전달하는 부분 : 포토커플러를 사용하여 외부기기와 내부회로는 전기적으로 절연되어 내 잡음성이 강해진다.

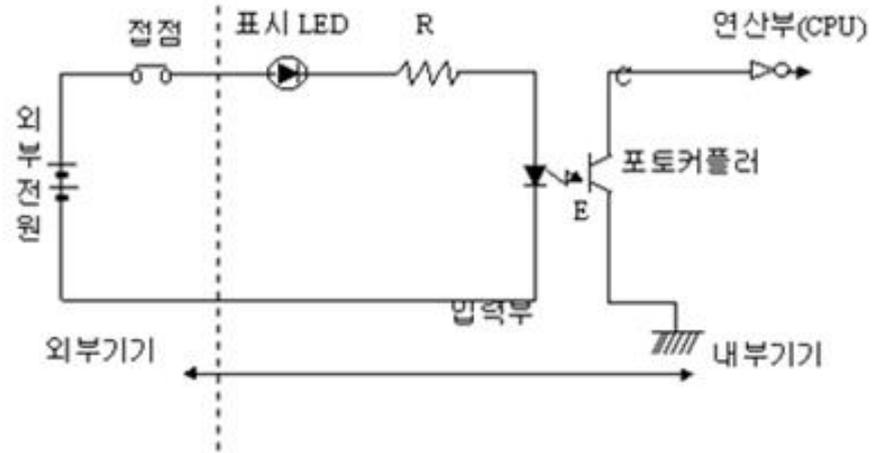
PLC의 입/출력부는 PLC와 기계나 장치와의 인터페이스이다.

PLC는 현장에서 기계나 장치와 직결하여 사용하는 것을 목적으로 하고 있으므로 입/출력부는 PLC의 중요한 요소다. 입력인터페이스에는 조작스위치나 검출 스위치 등과 같은 입력기기가 접속된다.

그리고 출력인터페이스에는 전자 계전기나 접속기의 코일, 솔레노이드, 히터, 밸브, 표시 등과 같은 출력기기가 접속된다.

PLC의 입/출력부는 현장의 외부 기기에 직접 접속하여 사용한다.

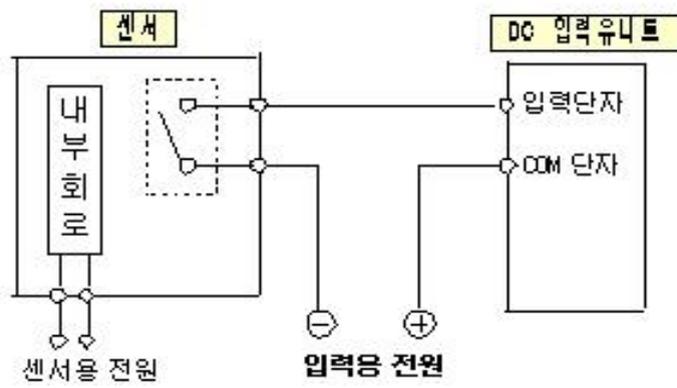
PLC 내부는 DC+5(V)의 전원(TTL 레벨)을 사용하지만 입/출력부는 다른 전압 레벨을 사용하므로 PLC 내부와 입/출력의 접속(interface)은 시스템 안정에 결정적인 요소가 된다.



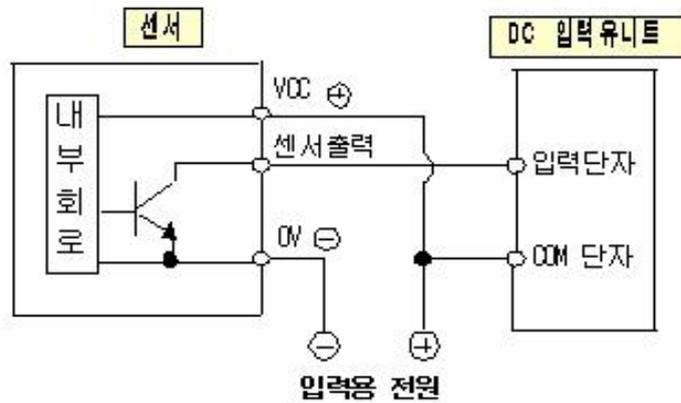
[그림 3-9] 입력부 회로 예

입력측 배선의 예

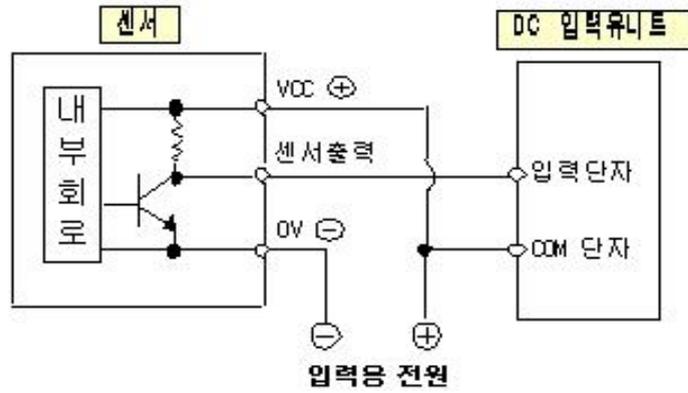
1) 릴레이 출력 타입



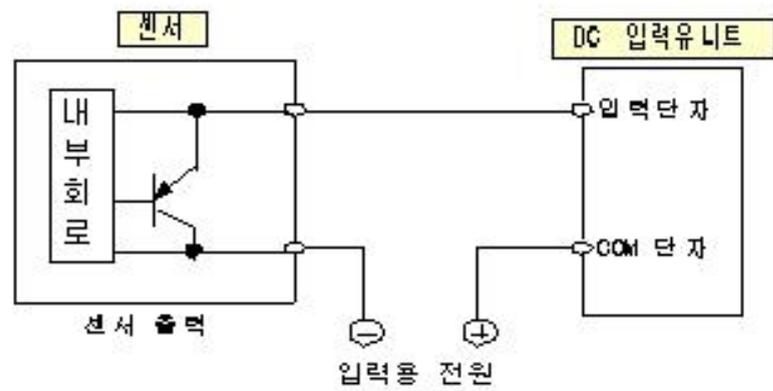
2) 릴레이 출력 타입



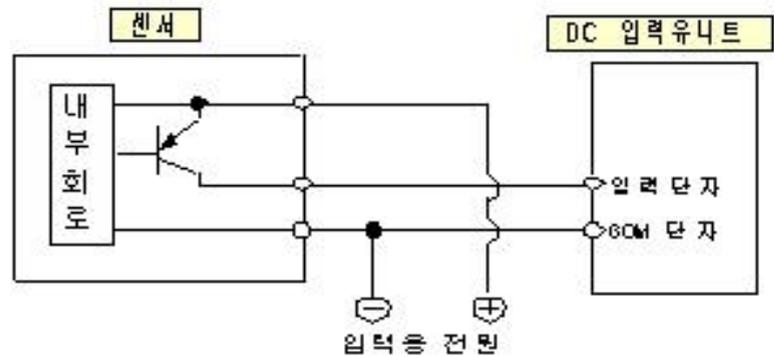
3) 전압출력 타입



4) 2선식 타입



5) PNP오픈 콜렉터 출력타입



출력부는 CPU로 부터 신호를 구동부(외부)에 전달해주는 부분 : 출력의 종류는 TR출력타입, 릴레이출력타입, SSR(Solid State Relay) 출력타입이 있다.

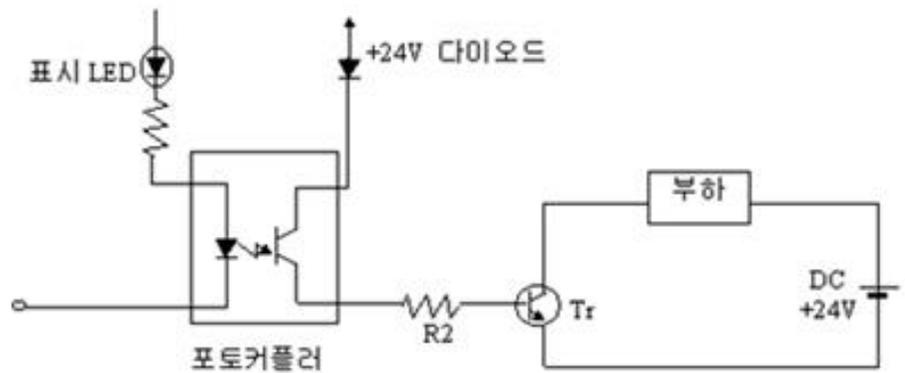
내부 연산의 결과를 외부에 접속된 전자 접촉기나 솔레노이드에 전달하여 구동시키는 부분이다.

PLC의 출력부는 시퀀스 프로그램에 기초하여 출력되는 신호를 외부의 출력기기로 전달하는 것뿐만 아니라 전자밸브 등의 출력

기기를 동작시키는데 필요한 전압 레벨로 변환하거나 전력을 증폭하는 역할도 하고 있다.

PLC로 제어하는 출력기기는 수십 [mW]의 표시등에서부터 수십 [kW]나 되는 대용량의 모터까지 다양하다. 그러나 PLC내의 IC 회로에서는 이들의 출력기기를 직접 동작시킬 수가 없기 때문에 연산부에서 출력된 IC회로 신호는 트랜지스터나 릴레이 등을 이용한 출력부로 전달된다.

표시등과 소형 릴레이 등은 이 출력부의 회로로 직접 동작시킬 수가 있지만 대용량의 모터나 고전압을 사용하는 기기는 출력 내부의 트랜지스터나 릴레이보다 한층 능력이 큰 릴레이나 전자 개폐기 등을 통해 동작시켜야 한다.



[그림 3-10] 트랜지스터 출력부 회로 예

출력측의 배선

출력유닛 배선의 체크 포인트

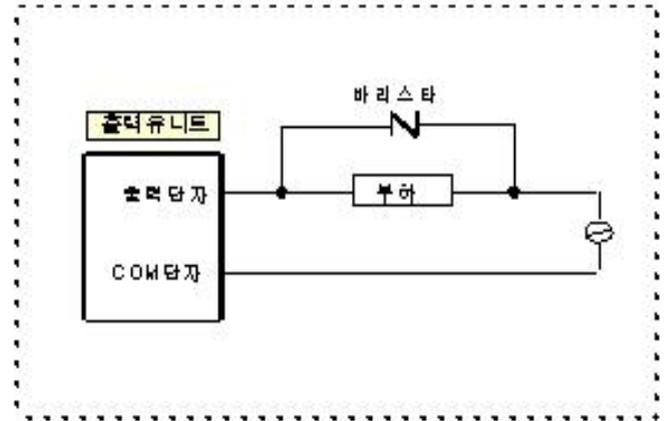
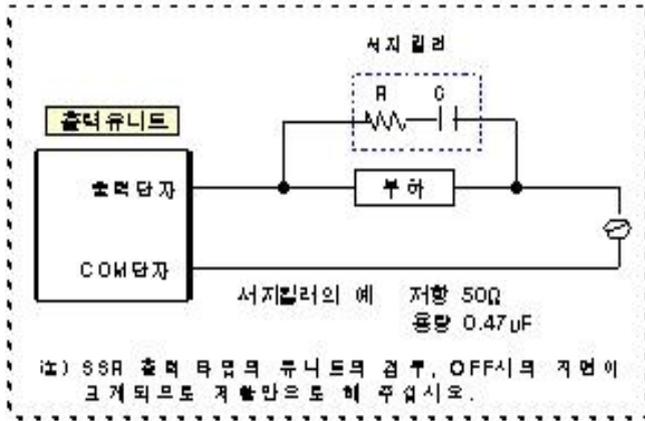
- 유닛의 종류에 따라 동시 ON 점수 또는 부하전류에 제한이 있는 것이 있습니다. 각 유닛의 사양을 참조하십시오. 특히 주의온도가 높은 곳에서 사용하는 경우는 주의 바랍니다.
- 유도성 부하, 용량성 부하등에 대해서는 아래와 같이 보호회로를 설치하여 주십시오.
- 출력측 유닛에는 Common당 전류제한이 있는 것이 있으므로 그 범위내에서 사용 해 주십시오.

유도부하의 보호회로

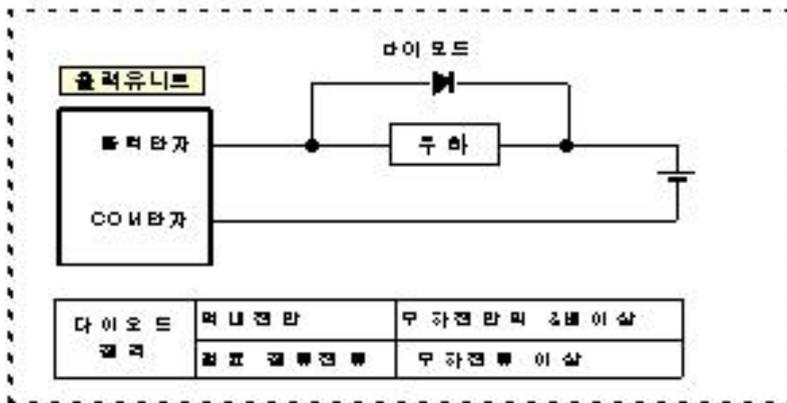
- 유도부하인 경우는 부하와 병렬로 보호회로를 마련해 주십시오.

- 특히, 릴레이 출력타입으로 DC 유도부하를 개폐하는 경우는 보호회로의 유무가 수명에 크게 영향을 주므로 반드시 부하의 양쪽에 다이오드를 설치해 주십시오.

1) AC 부하의 경우(릴레이 출력타입)

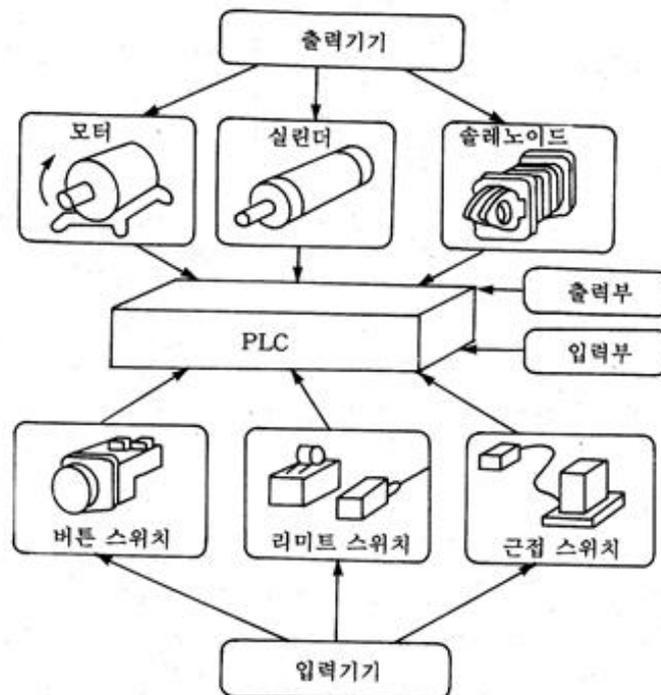


2) DC 부하의 경우



I/C	구분	부착 장소	외부 기기의 명칭
입력부	조작 입력	제어반 및 조작반	푸시 버튼 스위치 선택 스위치 토글 스위치
	검출 입력 (센서)	기계 장치	리미트 스위치 광전 스위치 근접 스위치 레벨 스위치
출력부	표시 경고 출력	제어반 및 조작반	파일럿 램프 부저
	구동 출력 (액추에이터)	기계 장치	전자 밸브 전자 클러치 전자 브레이크 전자 개폐기

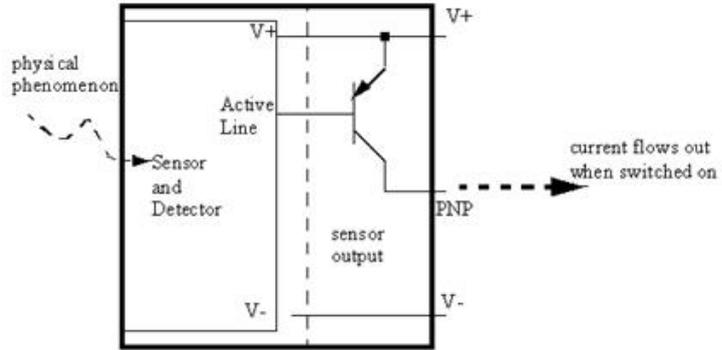
[표 3-3] 입, 출력부에 접속되는 외부 기기의 예



[그림 3-11] 입/출력부에 연결되는 외부 기기 예

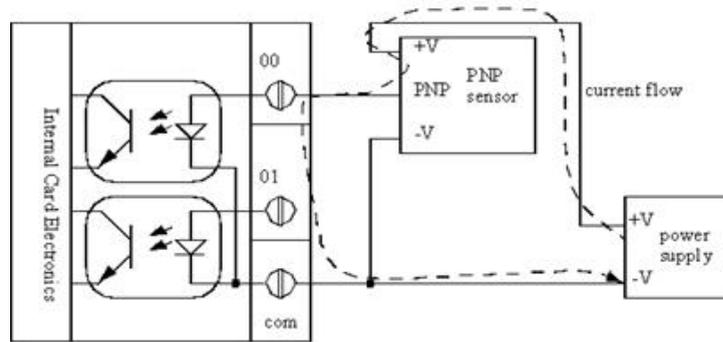
입출력 모듈의 연결 방식(싱크/소스)

Sourcing (supplies voltage/current)



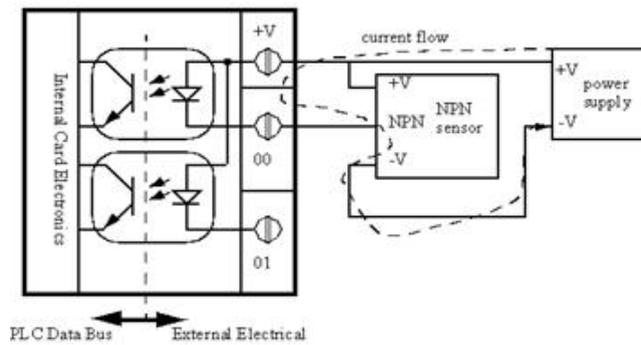
[그림 3-12] 소상(Sourcing)

PLC Input card for Sourcing (PNP) sensors



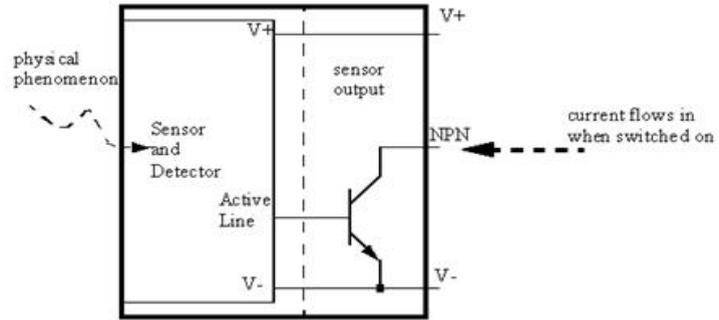
[그림 3-13] PLC Input card for Sourcing (PNP) sensors

Sinking (to the common)



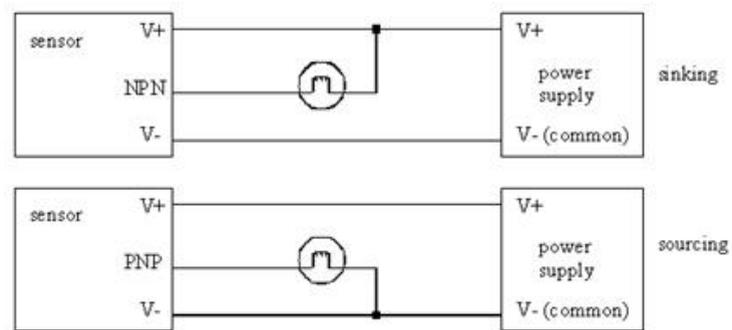
[그림 3-14] 싱크(Sinking)

A PLC Input Card For a Sinking (NPN) Sensor



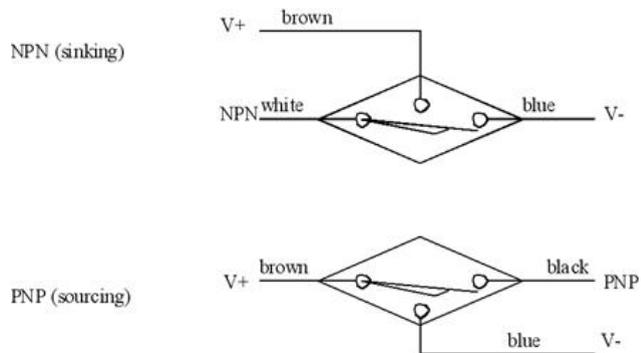
[그림 3-15] A PLC Input Card For a Sinking (NPN) Sensor

싱크형 입력을 갖는 PLC의 경우 센서를 연결하는 방법은 PLC의 COM단자에 +전압을 연결하고 센서에서 0V 즉, -출력이 나오도록 배선하여 필요한 입력 단자에 연결하여야 한다는 의미입니다. 소스형은 이와 반대의 경우가 됩니다.



[그림 3-16] 예

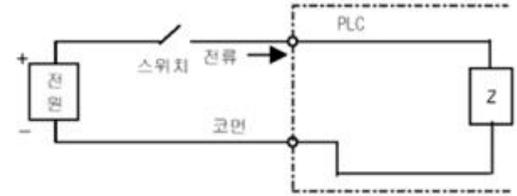
센서의 경우 출력이 +신호가 나오는 PNP형 센서와 출력이 0V가 나오는 NPN형 센서가 있습니다. 따라서 연결할 때 주의하여야 합니다.



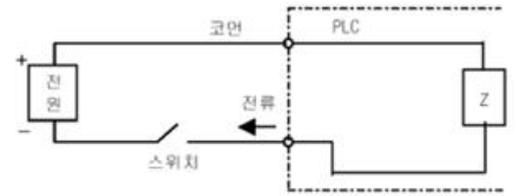
[그림 3-17] Sensor 예

**입력모듈**

SINK입력:  
센서가 ON되면 PLC로  
전류가 유입

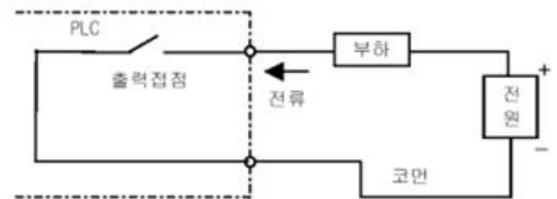


SOURCE입력:  
센서가 ON되면  
PLC로부터  
스위치로 전류가 유입

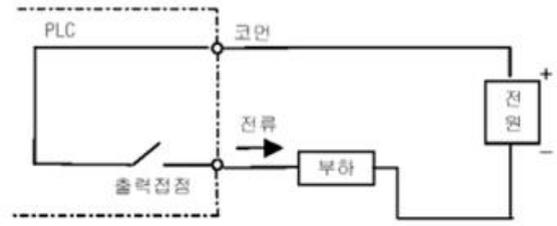


[그림 3-18] 입력 모듈에서의 Sinking/Sourcing

SINK출력:  
출력접점이 ON되면  
PLC로 전류가 유입  
출력모듈



SOURCE출력:  
출력접점이 ON되면  
PLC로부터  
부하로 전류가 유입



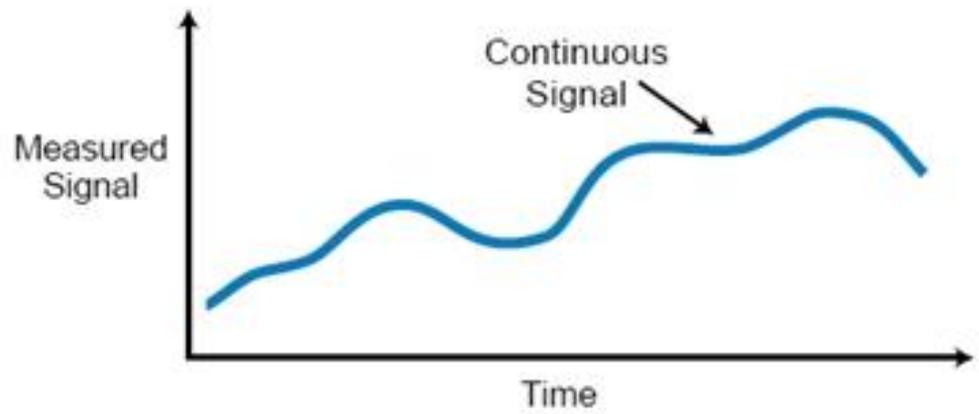
[그림 3-19] 출력 모듈에서의 Sinking/Sourcing

**아날로그/디지털 변환 유니트**

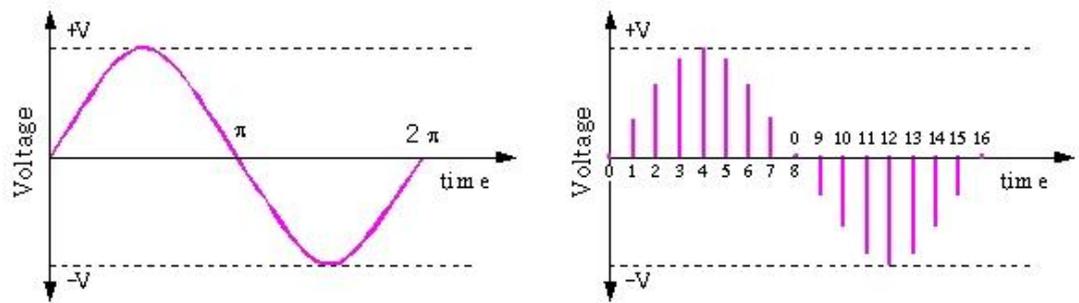
전압, 전류, 온도, 압력, 유량, 속도 등과 같이 연속해서 변화하는 물리량을 아날로그 값이라 부른다.

아날로그 신호는 컴퓨터나 PLC와 같은 디지털 기기의 CPU에 신호로 입력할 수 없다. 따라서 물리량(아날로그 값)을 전기적 신호 즉, 직류전압 또는 전류 형태로 변환하고 이 전기신호를 디지털로 변환하여 입력 하여야 한다.

또한 아날로그 신호로 동작하는 서보모터나, 모터 가변속 장치와 같은 기계장치에 아날로그 신호를 출력하기 위해서는 CPU의 디지털 신호를 아날로그로 변화 시켜 출력 한다.



[그림 3-20] 연속된 아날로그 신호

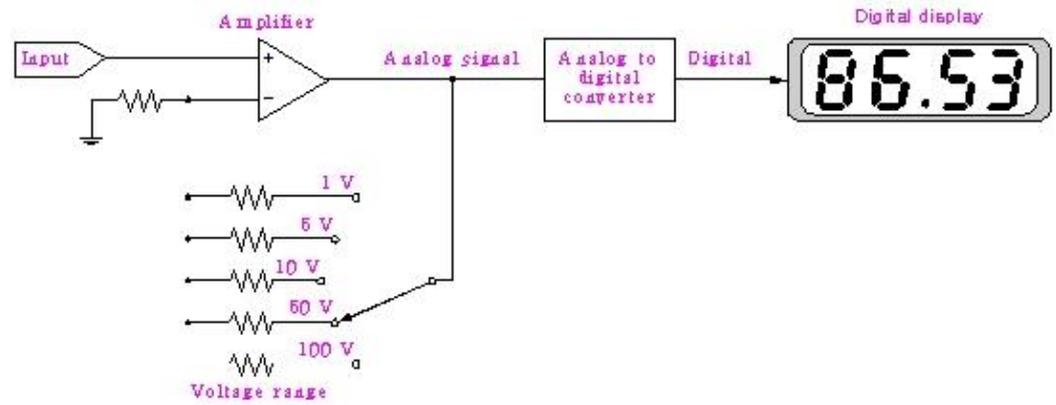


아날로그적 표현 이산적인 표현

[그림 3-21] 아날로그 /디지털 신호

아날로그신호: 연속적으로 표현된 신호

디지털신호: 이산적으로 표현된 신호



[그림 3-22] 디지털 전압계 예

### 디지털 입력 유니트(A/D UNIT)

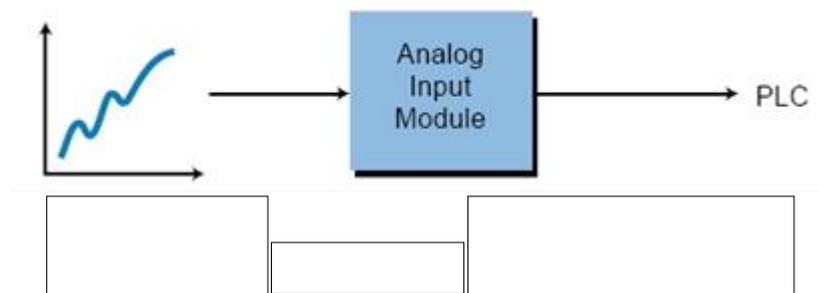
PLC의 디지털 입력 유니트는 입력기기로 부터 ON, OFF신호를 입력 받을 때 사용한다.

이것은 기기의 상태정보를 입력 받는데 주로 사용된다.

조작 패널의 푸시버튼이나 스위치, 기계에 부착된 각종 SENSOR가 연결되어 있는데 이들은 각각 다른 전력과 전압을 사용하므로 이것에 적합한 입력 유니트를 선정하여야 한다.

모듈타입의 PLC의 경우는 적용업무에 적합한 입력를 선정하여 사용하며 블록 타입의 PLC는 입력기기에 적합한 부가회로를 추가하여 적용전압과 전력으로 변환하여 사용해야 한다.

입력 전압의 형태에 따라 DC 입력, AC 입력 유니트가 있고 각각은 절연형과 비절연형이 있다. 여기서 절연형이라 함은 입력부에 포토 커플러를 두어 외부 입력 신호와 내부 입력신호사이에 절연한 것으로 노이즈에 강한 구조로 되어 있다.



[그림 3-23] Digitization of an analog signal

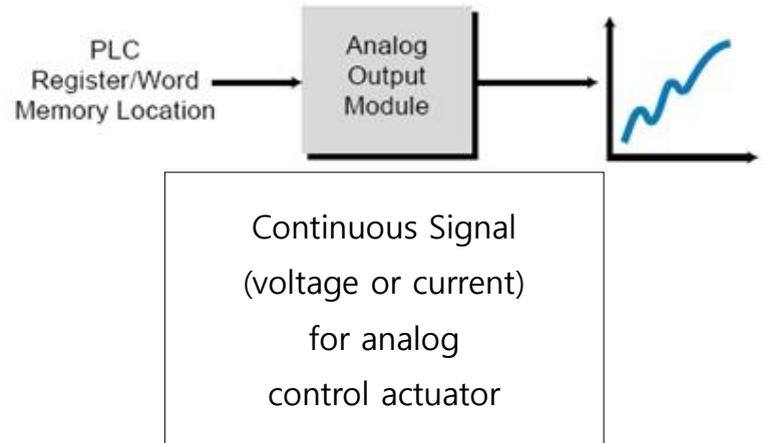
### 디지털 출력 유니트(D/A UNIT)

디지털 출력 유니트는 ON 또는 OFF 신호를 작동기기에 출력하여 기계장치를 동작시킨다.

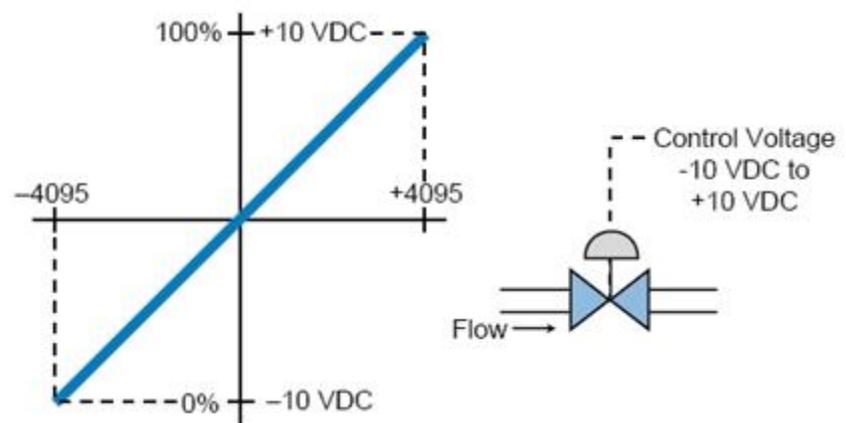
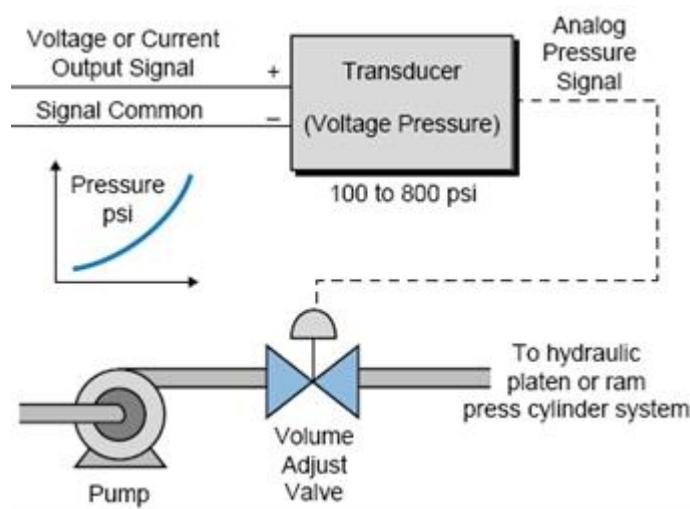
연결된 출력장치는 주로 램프, 마그네트(MC), 릴레이등이며 마그네트나 릴레이의 후단에는 작동기기인 모터나 밸프 등에 연결되어 있다. 이들 작동기기는 각각 다른 전력과 전압을 사용하므로 이것에 적합한 출력 유니트를 선정하여야 한다.

모듈타입의 PLC의 경우는 적용업무에 적합한 출력를 선정하여 사용하거나 적합한 부가회로를 추가하여 사용하며 블록 타입의 PLC는 출력기기에 적합한 부가회로를 추가하여 적용전압과 전력으로 변환하여 사용해야 한다.

출력 유닛은 출력 접점의 형태에 따라 릴레이출력, TR 출력, SSR 출력 유닛이 있다.



[그림 3-24] 디지털 신호의 아날로그화



[그림 3-25] counts, voltage, and percentage와의 관계

### 고속 카운트 유니트

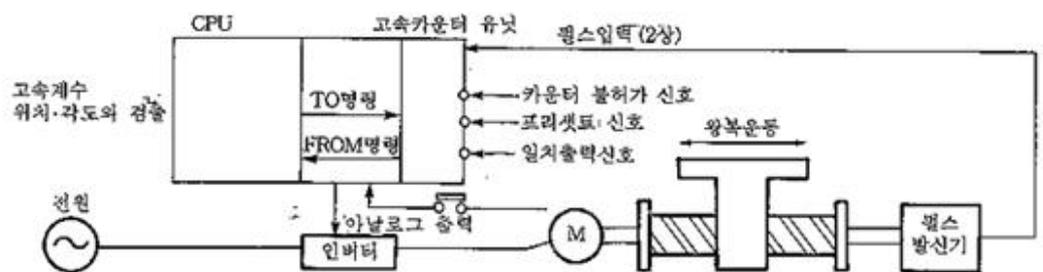
CPU 내의 카운터는 프로그램을 스캔(SCAN)하므로 스캔 주기는 수 ms이하일 경우가 대부분 이다. 그래서 이보다 더 짧은 펄스 신호를 카운터 할 수 없다. 초당 1000회 정도의 계수가 한계이다.

따라서 고속 카운터 유니트는 PLC의 독립적으로 펄스를 카운터 하여 10μs 이상의 펄스를 카운터 한다. 고속 카운트 유닛을 사용하면 메이커의 기종에 따라서는 1만 카운트/초에서 5만 카운트/초까지 계수가 가능하므로 펄스 인코더 등에서 발신하는 신호를 수신할 수 있다.

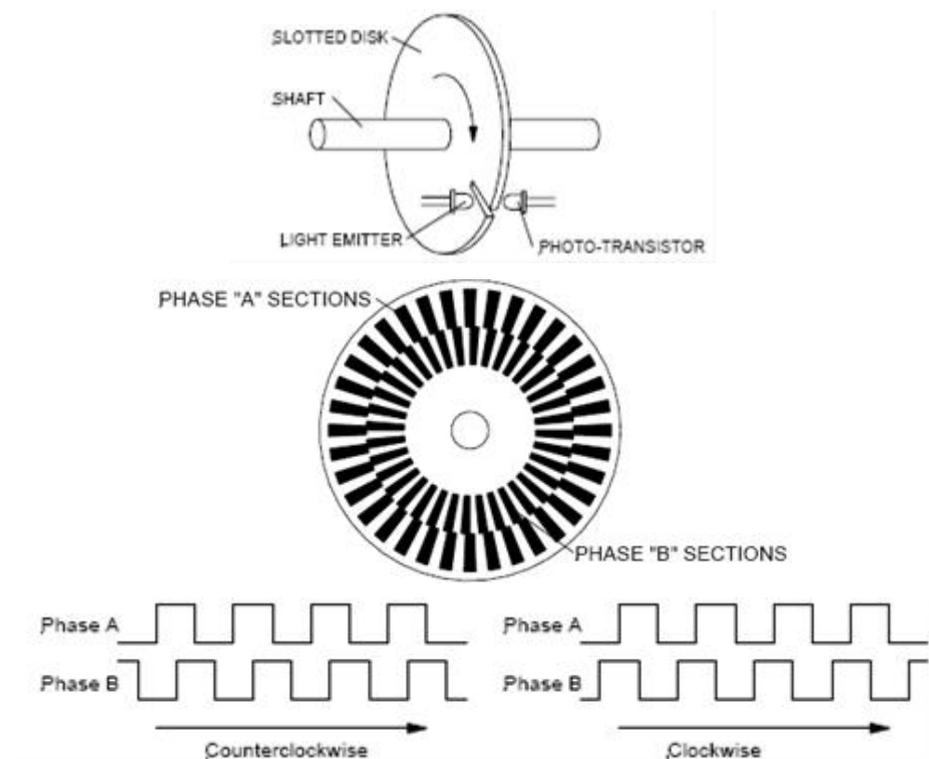
또한 비교 기능을 가지고 설정치와 CPU와 카운터 값의 상태( <, >, = )을 출력하여 준다.

또한 가, 감산 카운터를 지원하여 CNC와 같은 기계의 위치 검출에도 사용 되며 설정치와 카운터 값이 일치할 때의 외부 출력 접점을 가지고 있다.

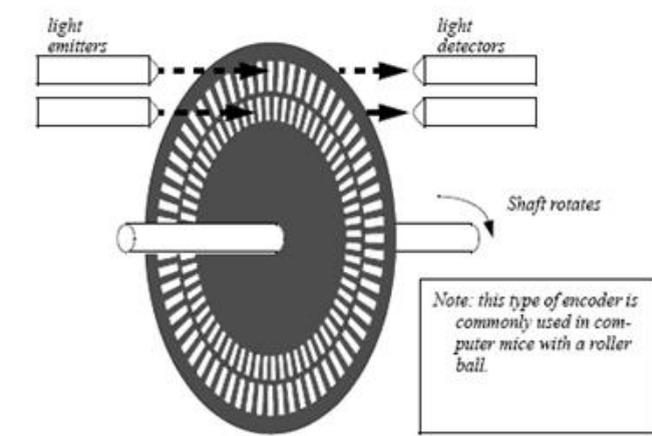
- ① CPU의 카운트 C는 설정값에 도달되어도 카운트는 정지하지 않고 최대값까지 카운트한다.
- ② 고속 카운트 유닛은 가산 및 감산이 가능하다.
- ③ 고속 카운트 유닛은 설정값과 일치 신호 외에 대소 비교 신호가 있다.
- ④ 고속 카운트 유닛은 설정값의 일치 신호가 유닛 자체에서 출력이 가능하다.



[그림 3-26] 고속 카운트 유닛을 적용한 시스템의 구성 예



[그림 3-27] Incremental Encoder Output Waveform



[그림 3-28] Encoder의 모양

### 위치 결정 유니트

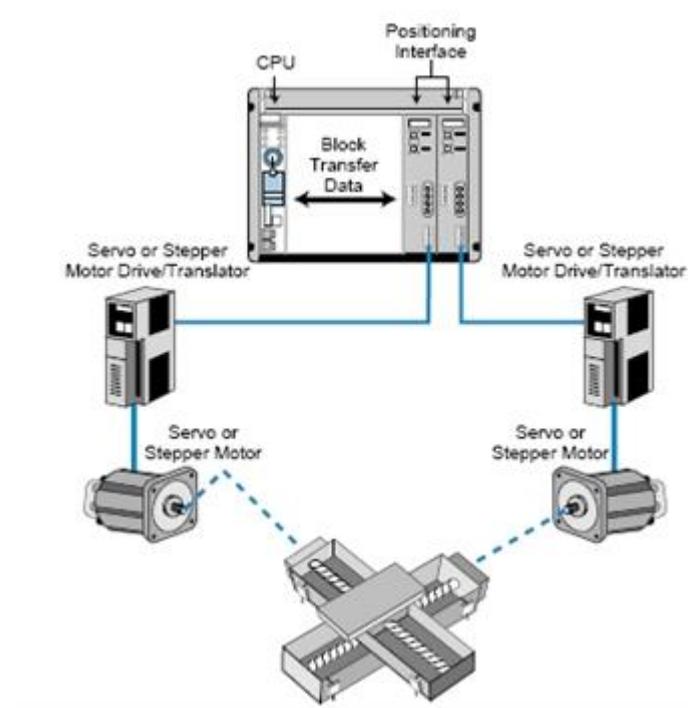
서보모터나, 스텝핑모터에 의한 정밀한 모터의 제어를 위해 위치 결정유니트가 사용된다.

X-Y 테이블의 위치 결정 등에 응용되며 등속, 가속 감속 등의 다양한 운전 패턴을 지원한다.

지금까지의 제어는 펌프 구동용 모터의 운전정지 등 ON/OFF 제어가 많았지만 자동화기기의 발전과 더불어 고정도의 위치 결정

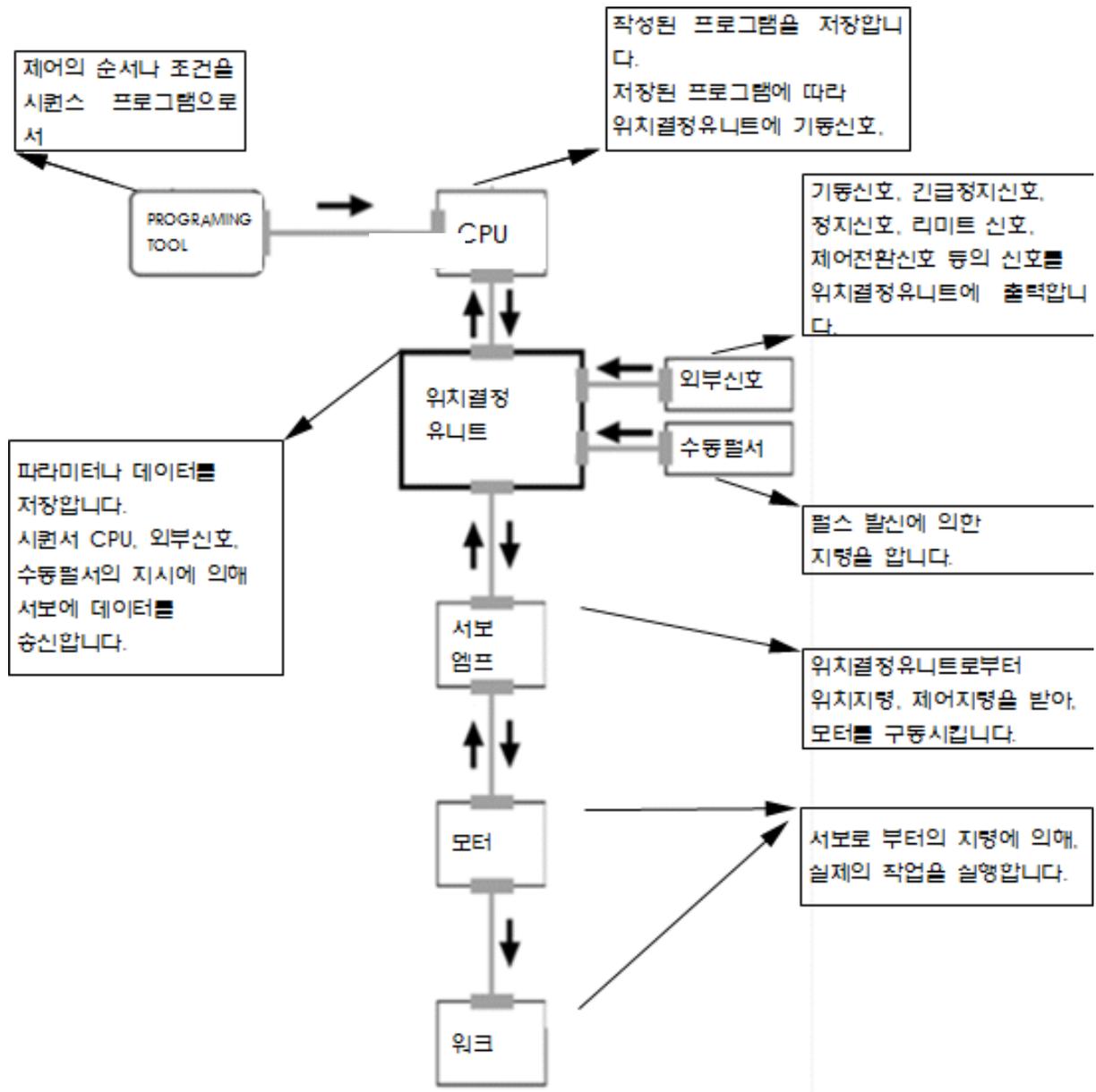
제어가 많이 요구되고 있다. 위치 결정 제어는 ON/OFF 제어와는 달리 「스텝1에서 52mm, 스텝2에서 84mm 이동시킨다」와 같이 제어량의 변화를 제어한다.

이 제어에는 서보모터, 스텝핑모터 등이 사용되고 운전 패턴에도 가속, 정속, 감속운전 등 전문지식이 필요하므로 일반 기술자는 취급하기 어려웠으나 최근의 PLC는 위치 제어 전용 유닛을 사용하여 간단히 위치 결정 제어가 가능하게 되었으며, 정전이나 역전의 조건을 위치 결정 유닛에 입력해 두고 PLC에서 위치 결정 유닛으로 신호를 출력하고, 다시 위치 결정 유닛에서 모터 드라이버로 신호를 출력시켜 모터를 구동시킨다.

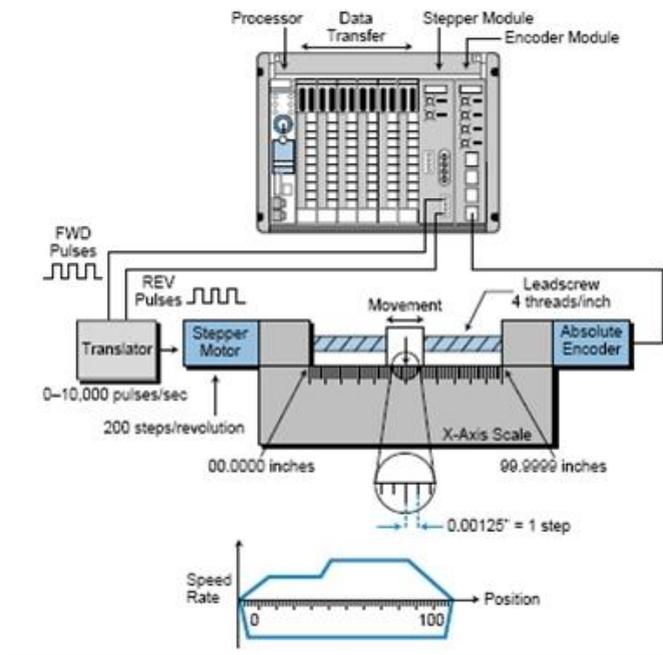


[그림 3-30] 위치결정 유닛 개념도

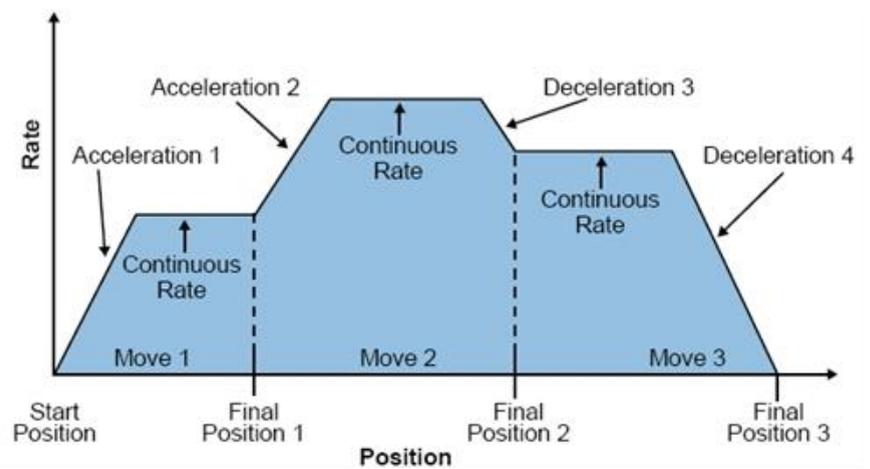
위치결정제어의 구조



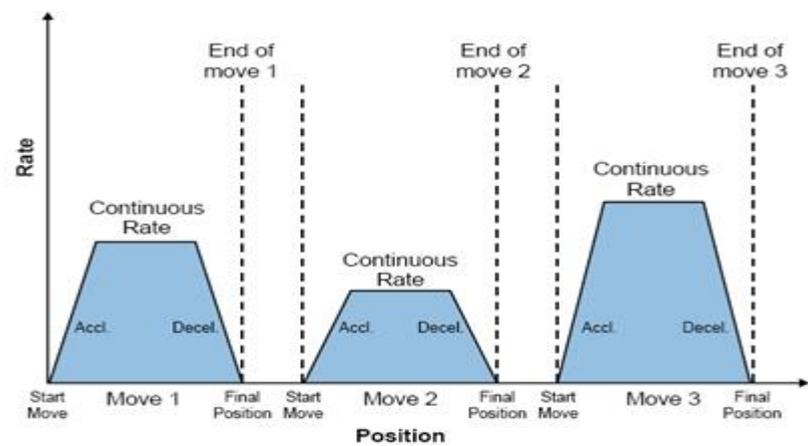
[그림 3-31] 위치결정 제어의 구조



[그림 3-29] 펄스신호를 이용한 Motion동작



Single-step profile mode.(PASS MOVE)



Continuous profile mode. (End MOVE)

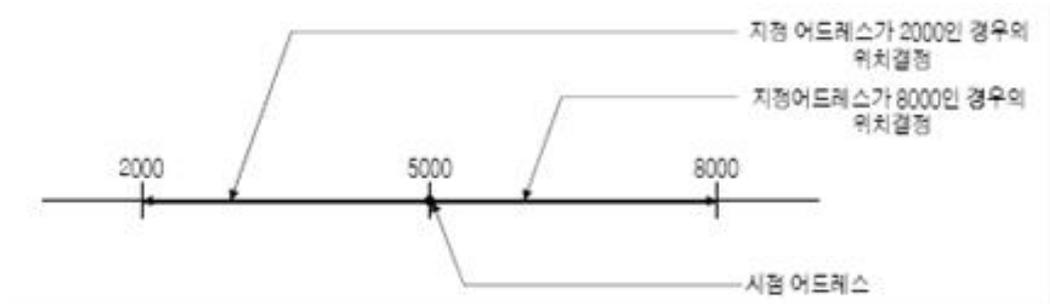
[그림 3-29-1] 펄스신호를 이용한 Motion동작 그래프

**위치제어**

지정된 축을 시점 어드레스(현재의 정지위치)로부터 지정한 위치 간에서의 위치결정 제어를 실행합니다.

1. 앵슬루트 방식에 의한 제어

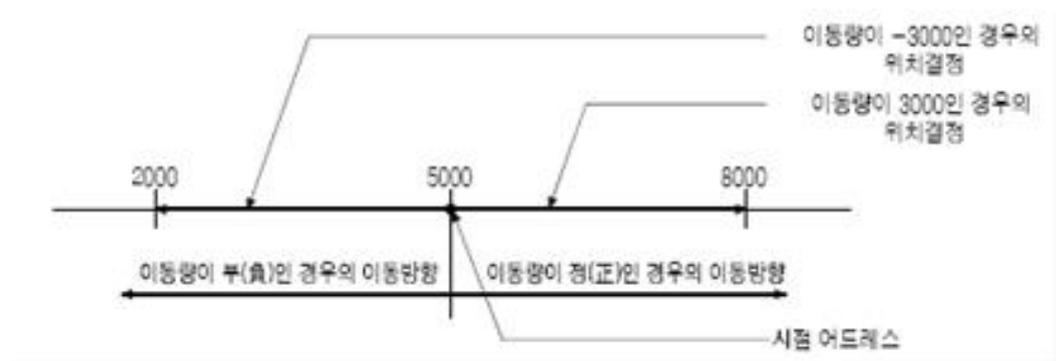
- ① 시점 어드레스로부터, 지정된 어드레스로의 위치결정을 실행합니다.
- ② 이동방향은, 시점 어드레스와 지정 어드레스에 의해 정해집니다.



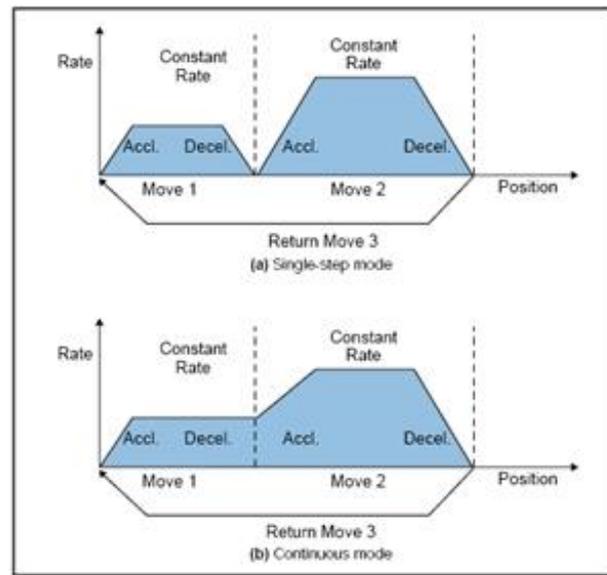
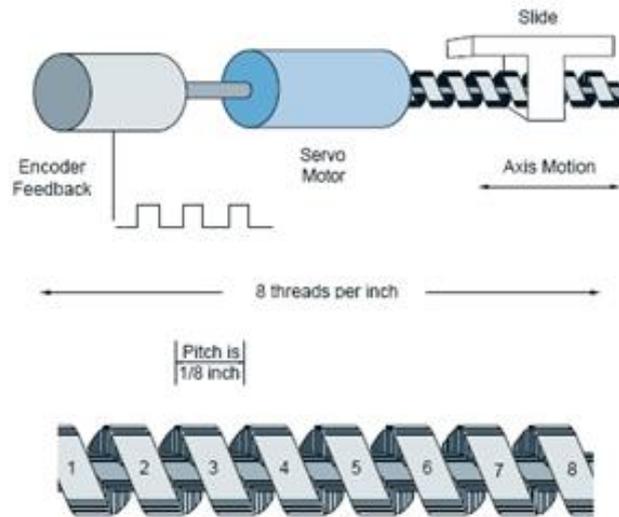
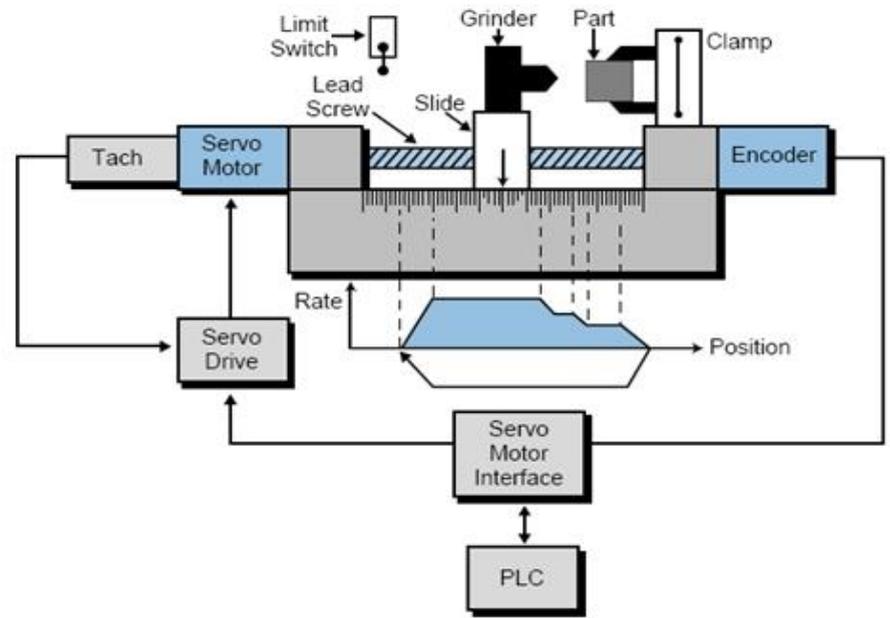
[그림 3-32] 위치결정 앵슬루트 방식에 의한 제어

2. 인크리먼트(increment) 방식에 의한 제어

- ① 시점 어드레스로부터, 지정된 이동량 분의 위치결정제어를 실행합니다.
- ② 이동방향은, 이동량의 부호에 의해 정해집니다.
  - 이동방향이 정(正)일때..정방향(어드레스 증가방향)으로의 위치결정
  - 이동방향이 부(負)일때..역방향(어드레스 감소방향)으로의 위치결정



[그림 3-33] 위치결정 인크리먼트 방식에 의한 제어

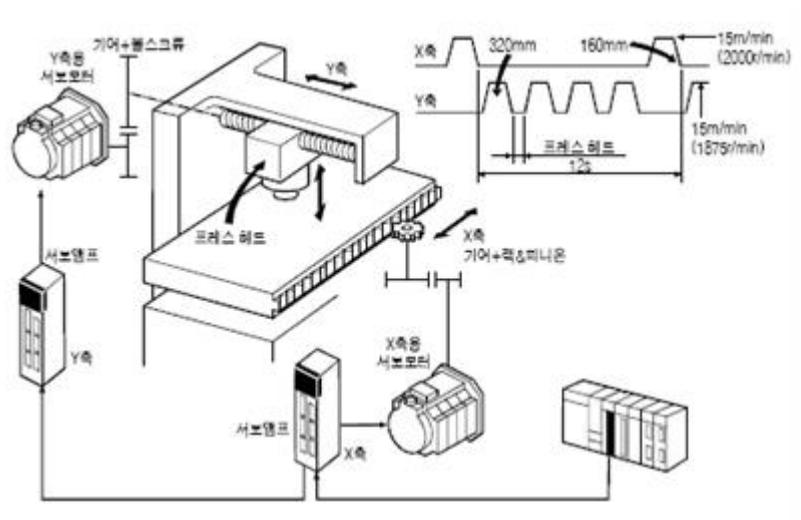


[그림 3-34] 위치결정 제어

### 위치결정 유니트의 사용 예

#### 1. 펀치 프레스(X, Y의 이송 위치결정)

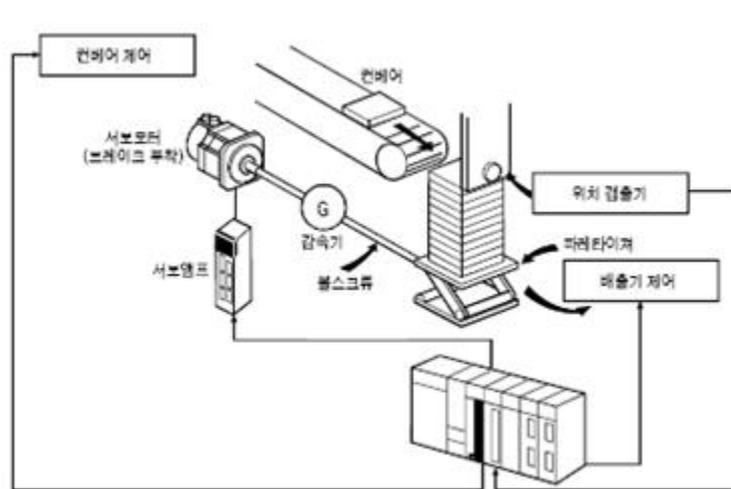
- ① 절연재료나 가죽 등을 단일형으로 구멍을 뚫기 위해서, X축, Y축의 2축의 서보를 사용해서 순차적인 위치결정을 실행합니다.
- ② X축의 서보로 이송반을 위치결정 후, Y축의 서보로 프레스헤드를 위치결정하여, 프레스로 구멍을 뚫습니다.
- ③ 재료의 종류나 형상이 바뀔 때는, 프레스 헤드의 형태를 교환함과 동시에, 위치결정 패턴을 새로 교체합니다.



[그림 3-35] 위치결정 제어\_펀치 프레스

#### 2. 파레타이저

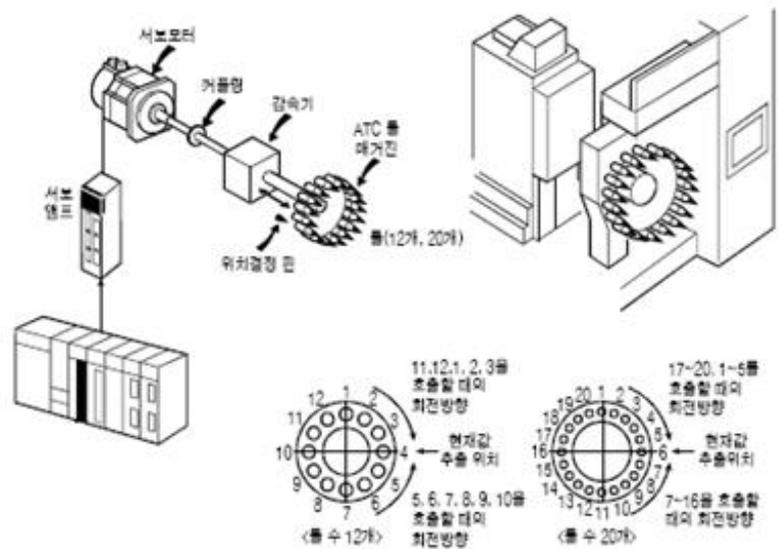
- ① 1축의 서보를 사용해서, 파레타이저의 고정밀도 위치결정을 실행합니다.
- ② 재료의 두께에 대응하는 파레타이저의 강하 이동량을 저장 시킵니다.



[그림 3-36] 위치결정 제어\_파레타이저

3. 소형머시닝 센터(ATC 매거진 위치결정)

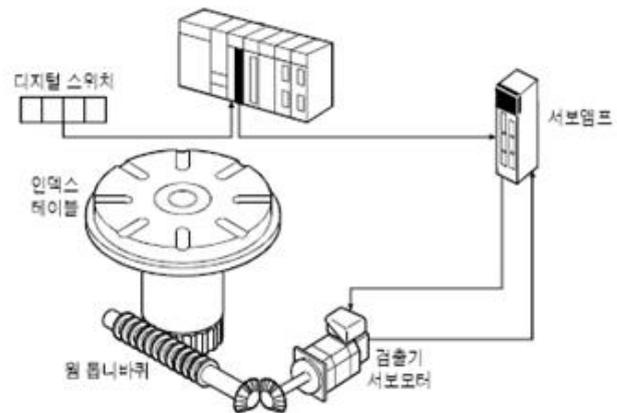
- ① 소형 머시닝 센터 ATC 톨 매거진의 위치결정을 실행합니다.
- ② 매거진의 현재값과 목적값과의 관계를 연산하고, 정회전 또는 역회전으로 최단 액세스가 되는 방향의 위치결정을 실행합니다.



[그림 3-37] 위치결정 제어\_소형머시닝 센터

4. 인덱스 테이블(각도의 고정밀도 분할)

1축의 서보를 사용해, 인덱스 테이블의 고정밀도 위치결정을 실행합니다.

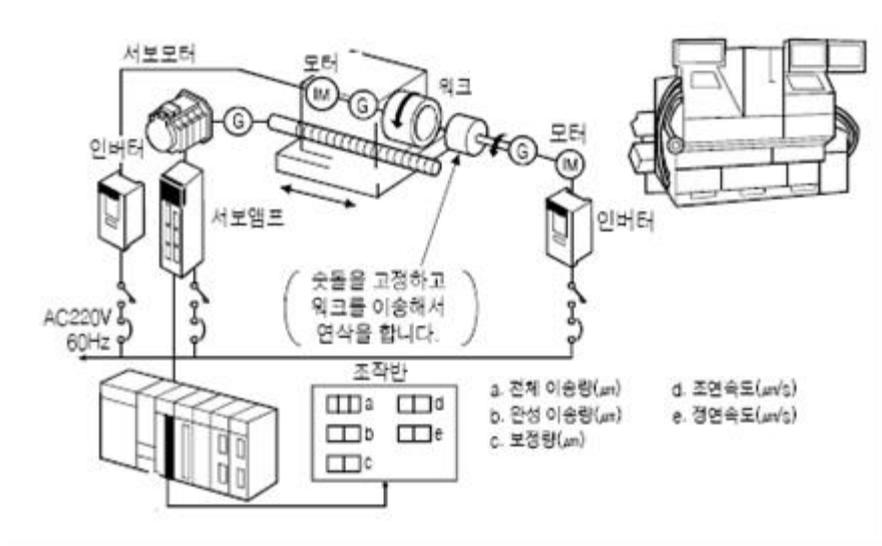


[그림 3-38] 위치결정 제어\_인덱스 테이블

5. 내면 연삭반

- ① 서보와 인버터를 사용해, 워크 내면의 연삭을 제어합니다.
- ② 1축의 인버터로 워크의 회전을 제어하고, 2축의 인버터로 슷돌의 회전을 제어합니다.

3축의 서보로 워크를 보내, 연삭을 실행합니다.

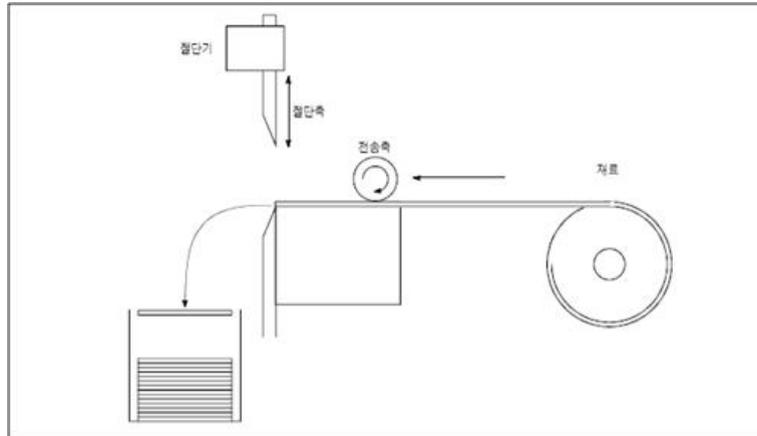


[그림 3-39] 위치결정 제어\_내면 연삭반

6. 절단기

- ① 전송축에서 재료를 절단기에 보냅니다.
- ② 절단기로 절단 합니다.

반복해서 재료를 정치수로 절단하는 시스템 예입니다.



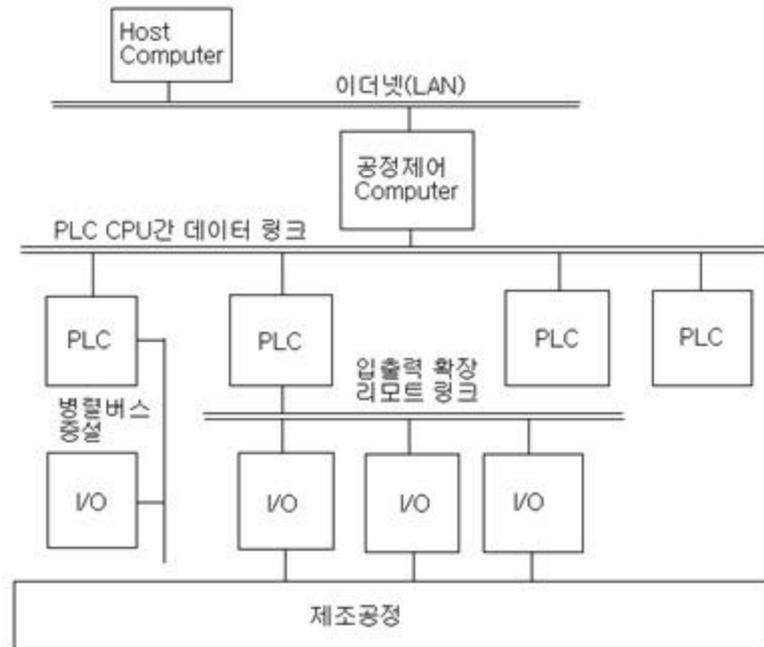
[그림 3-40] 위치결정 제어\_절단기

7. 데이터 링크 유닛

대규모 시스템에서 PLC는 분산된 원거리의 입출력기거나 다른 PLC, 상위 컴퓨터 등과의 데이터의 교환이 요구된다.

따라서 데이터 링크 유닛은 원거리의 입출력 접점과의 배선공사를 줄이고, 분산제어, 데이터의 집중 관리 및 Monitoring을 가능토록 하기위해 사용된다.

PLC의 정보 수집을 위한 기기와 입출력 접점을 확장하기 위한 기기가 있다.



[그림 3-41] 데이터 링크

## PLC 사용상의 주의점 및 대책

PLC를 실장한 제어장치가 기능을 충분히 발휘하도록 하고, 안정되게 가동되기 위해서는 Programming상의 연구도 물론 중요하지만, Noise 대책이나 PLC가 설치될 환경에 대한 고려와 조치가 상당히 중요한 부분을 차지하게 된다.

Noise 대책이나 설치환경에 대한 Hardware적인 조치와 대책은 제어장치의 가격에 직접적으로 영향을 미치며 게다가 가격이 고가로 되는 경우가 많다.

따라서 모든 경우에 만전을 기하여 대책을 실시하는 것은 어렵지만, 상황이 허락하는 범위 내에서의 대책 수립은 충분히 고려해야 할 필요가 있다.

## 동작환경

### 1. 전원

#### 전원계통

PLC를 사용하여 시스템을 구성할 경우의 전원계통은 PLC자체 전원계통 외에 제어계통, 입·출력 전원계통, 외부기기 계통 등이 있다.

따라서 시스템의 신뢰성을 높이기 위해서는 전원공급을 각각 계통별로 분리하는 것이 바람직하며, 이렇게 하는 것은 입·출력 부분의 사소한 오류로 인하여 제어 계통 전체로 영향이 파급되는 것을 방지하기 위함이다.

#### 전원공급용 전선

전원 공급용 전선은 동력기기의 Surge 전류나 시스템 전체의 소비전류를 인해 전압강하를 일으키지 않도록 전류용량에 충분한 여유가 있는 전선을 사용한다.

#### 전원의 전압변동

PLC의 전압변동 범위는 일반적으로 +10% ~ -15% 이내인 경우가 많으나 빈번한 전압변동은 허용범위 이내라 하더라도 PLC에 나쁜 영향을 끼친다.

공정에서 중요한 역할을 담당하는 PLC의 경우에는 갑작스런 정전이나 전압의 변동으로 인한 PLC의 Trouble을 방지하기 위해서는 UPS 전원과 Normal 전원을 각각 연결하는 것이 좋다.

일반적으로 PLC는 10ms 정도의 순간 정전이 일어나도 운전을 계속하는 성능을 가지고 있다.

#### 입·출력 기기용 전원

입·출력 Unit용 외부전원의 전압변동범위는 각 Unit에 따라 다르기 때문에 각각의 설비사양에 기재된 범위를 준수해야 한다.

직류 전원의 리플 허용값은 전압의 변동 범위내이기에 입·출력선이 긴 경우에는 전압강하가 발생하여 전압부족이 되는 경우도 있으므로 주의하여야 한다.

### 2. 온도

#### 고온 대책

직사광선을 쬐는 곳, 옥외, Furnace 등의 근처, 천장이 가까운 장소, Control Panel 내에 발열소자가 있을 때 등에는 PLC의 고온화 현상에 대한 주의를 기울여야 한다.

PLC를 고온으로 사용했을 때는 IC등 반도체 부품의 열화 및 수명저하, 고장빈도의 증가, 정밀도 저하 등의 이상이 발생하는 경우가 있다.

그러므로 PLC는 항온 항습이 유지되는 I/O Room 내에 위치시키는 것이 좋으며 그렇지 못한 경우에는 Control Panel 내의 Fan 설치, 직사광선의 차단, 양호한 통풍 유지 등의 대책이 필요하다.

#### 급격한 온도변화

1시간에  $\pm 10^{\circ}\text{C}$  이상의 온도변화가 있을 때 PLC내 부품의 팽창률의 차이로 인한 팽창, 수축으로 조임 나사가 풀리는 경우가 있다.

또 반도체 부품의 열화나 회로의 온도특성에 의한 조정의 변화 등 나쁜 상태가 발생하는 경우도 있을 수 있다.

#### 통풍

PLC의 통풍을 좋게 하는 것도 하나의 온도대책이 될 수 있다.

PLC의 통풍을 좋게 하기 위해서는 PLC 본체의 상부, 하부는 구조물이나 부품과의 거리를 적어도 50mm이상 두어야 한다. 또 배선용 덕트를 설치할 때는 통풍에 방해가 되어서는 안 된다.

### **3. 습도**

일반적으로 PLC의 사용중의 습도는 35% ~ 85% RH(상대습도)이다. 특히 고 습도에서 장시간 사용하면 절연성이 떨어지므로 세심한 주의가 필요하다.

매우 건조한 상태에서는 절연물상의 정전기에 의한 대전이 있을 수 있으므로 건조 상태에서 Unit의 장착이나 점검을 할 때는 반드시 인체의 전기를 방전한 후에 한다.

### **4. 진동/충격**

PLC를 설치할 때는 진동에 주의하지 않으면 안되며, 진동과 충격이 미약하더라도 장기간 지속되는 진동은 조임 부분의 풀림, 전기부품의 피로파괴, 커넥터의 스프링 피로로 인한 접촉 불량 등이 발생한다.

정상적으로 진동이 있는 경우나 큰 충격이 있는 경우, 설치한 Control Panel 이나 전자기기류에서 문제가 발생하는 경우 등에는 다음과 같은 대책을 실시한다.

- Control Panel에 방진고무를 부착한다.

- Control Panel이 공진하지 않도록 구조를 강화한다.
- 진동원에서 가능한 먼 곳에 Control Panel을 설치한다.
- 내진형 Terminal Block를 사용한다.

### 5. 공기중 부유물(분위기)

먼지, 도전성 분말, 부식성 가스, 수분, 유분, Oil Mist, 유기용제, 염분 등이 있는 장소에서는 다음과 같은 불량 발생한다.

- 먼지로 인한 접촉부분의 접촉 불량
- 먼지로 인해 필터가 막혀 일어나는 Control Panel의 온도상승
- 도전성 분말로 인한 오동작, 절연 열화와 단락
- 유분, Oil Mist 인한 접촉 불량
- 부식성 가스, 염분에 의한 프린트 기판패턴이나 부품 리드선의 부식, 릴레이, 스위치류의 접촉 불량

이와 같은 분위기에서는 다음과 같은 대책을 세운다.

- Control Panel을 밀폐구조로 한다.

이때는 온도상승에 주의한다.

- Air Purge를 실시한다.

Control Panel의 내압이 외부공기보다 높아지도록 청정한 공기를 압송하고 필요시 양압 관련 Alarm 기능을 설치한다.

### Noise 대책

Noise의 발생원으로는 여러 가지 있지만 대표적인 것은 다이리스터기기, 전자개폐기의 코일이나 접점 등이 있다.

PLC의 입·출력부에는 신호의 입·출력선으로 부터의 Noise 침입이나, 유도전압에 대한 배려에서 동작전압을24V로 하거나 Photo coupler로 절연하는 등의 사전 대책이 취해지고 있지만, PLC 내부의 회로는 마이크로세서를 중심으로 5V라는 낮은 전압, 더욱이 2~3MHz 이상의 고주파수로 동작하고 있어서 Noise에 대해서는 열악한 구조를 가지고 있다고 할 수 있다.

## 6. 접지

접지의 목적으로는 다음과 같은 것을 생각할 수 있다.

- PLC와 Control Panel 및 대지간의 전위차가 없게 하여, 전위의 차이로 인한 노이즈 전류를 감소시킨다.
- 전원 및 입력 신호선에 혼입한 Noise를 대지로 방전하여 Noise의 영향을 감소시킨다.
- 누설 전류에 의한 감전을 방지한다.

이와 같이 접지는 Noise로 인한 오동작을 방지하는 유효한 Noise 대책이 된다.

접지 방법은 다음과 같은 사항을 따르도록 한다.

- 접지는 PLC만을 접지하는 전용접지가 가장 좋으므로 될 수 있으면 전용 접지로 한다.
- 전용 접지를 할 수 없을 때는 한 접지점에서 다른 기기의 접지와 접속되는 공용접지로 한다.
- 다른 기기와 공통의 접지선을 사용하는 공통 접지는 될 수 있는 대로 하지 않는다.

특히 전동기, 변압기 등의 전력 기기와 공통 접지는 절대로 피해야 한다. 또 단지 감전방지가 목적이고 많은 기기가 접속되어 있는 접지선이나 철골 등에 접지하는 것도 피해야 한다.

- 외함의 접지는 제 3종 접지(100hm 이하)를 사용하며, Signal 접지는 되도록 10ohm이하의 접지를 사용하며 가능하면 외함의 접지와 Signal 접지는 분리시키도록 한다.
- 접지선은 될 수 있는 대로 굵은 전선을 사용한다.
- 접지점은 될 수 있는 대로 PLC 본체 가까이에 설정한다. 거리는 50m이하가 되도록 한다.
- 접지선의 배선에서는 강전회로, 주회로의 전선에서 될 수 있는대로 떨어지고 또 평행하는 거리를 될 수 있는 한 짧게 한다.

## 7. 전원부의 대책

전원부의 배선에 관해서는 Twist된 배선을 사용함과 동시에 1차측과 2차측(PLC측)의 배선을 접근시키거나, 절대로 묶어서 배선하지 않는 것이다.

1차측과 2차측을 묶어 배선하면 Noise 억제효과가 거의 없어진다. 또 공급전원은 주전원회로 나 입·출력기기용, 동력전원의 회로와는 분리하고 사용 트랜스는 용량[VA]적으로도 여유를 두는 것이 좋으며, Regulation이 좋은 것을 사용한다.

### 8. 입·출력부 선정과 배열에 의한 대책

입출력 Unit을 Noise원에서 보면 다음과 같이 생각할 수 있다.

- 입·출력 신호와 내부회로가 비절연인 것보다 절연한 것이 Noise 내력이 높다.
- 접점 출력보다 무접점 출력(Transister, Triac 등)이 PLC쪽이 받는 Noise의 영향이 적다.
- 입력 Unit에서는 ON 전압과 OFF 전압의 차가 큰 것일수록 Noise에 강하다.

또 입력 응답시간이 긴 것일수록 Noise에 강하다.

따라서 이런점을 고려하여 입·출력 Unit을 사용한다.

- Noise가 많은 환경에서는 절연형 Unit을 사용한다.
- 제어대상에 부착하는 입·출력기기의 입·출력 Unit은 절연형을 사용한다.
- 외부 Noise가 혼입하지 않는 Control Panel 이나 조작반내의 입·출력 기기의 Unit은 비절연형이라도 좋다.
- 코일 부하의 구동에는 접점 출력보다 트랜지스터, 트라이악 등의 무접점 출력이 좋다.

사용하는 입·출력 Unit의 배열은 Noise 대책을 고려하여 배정한다. 기본적인 방안은 CPU Unit을 될 수 있는 대로 Noise 발생원으로부터 멀리하는 것이다.

### 9. 입·출력부 선정과 배열에 의한 대책

입력기기 측에서 침입하는 Noise에는, 입력선간 Noise(Normal Mode Noise)와 내부회로의 Common Line의 전위를 상승시키는 입력 신호선과 대지간 Noise(Common mode Noise) 이외에, 주 회로나 동력선에 흐르는 대전류에 의하여 발생하는 유도전압, 입력기기 그 자체가 발생시키는 Noise나 Surge가 있다.

Normal mode Noise는 PLC입력부의 필터 등으로 감쇠시킬 수 있고, Common mode Noise는 접지에 의한 대책이 있다.

#### 유도전압 대책

유도전압은 입력선간의 용량전류(콘덴서)나 대전류가 흐르는 동력선과 입력 신호선간에 존재하는 용량전류에 의한 전기적인 결합에 의하여 발생한다.

이에 대해서는 다음과 같은 대책이 있다.

- 가능한 입력전원을 AC 보다는 DC를 사용토록 한다.

- 입력단자와 Common 단자간에 Dummy 저항이나 Surge Killer를 삽입하고 입력 임피던스를 낮추어 여기에서 발생하는 전압을 낮춘다.
- 유도전압 대책의 핵심은 입력기기부터 입력부까지의 신호배선의 거리를 가능한 짧게하고, 대전류선을 가까이 하지 않는 것이다.

장거리 배선이나 유도전압이 클 때에는 Shield 케이블을 채용하거나 또는 Interposing 릴레이로 중계하는 것도 효과적이다.

나) 입력기기로부터 침입하는 Noise 대책

Limit Switch로 부하를 On-Off함과 동시에 부하의 동작지령을 확인하기 위해 그 신호를 PLC에 입력하는 경우가 있다. (그러나 일반적으로는 Limit Switch 신호를 PLC에 입력하고, 이 입력신호에 따라 부하를 On-Off 하는 경우가 많다)

이때 이 부하가 유도부하인 경우에는 커다란 Noise나 서지전압을 발생시킨다. 그 대책은 입력전원이 AC인 경우에는 CR식 스파크 킬러를, DC인 경우에는 다이오드를 삽입한다.

출력기기로부터 침입하는 Noise 대책

PLC의 출력부와 접속하는 대부분의 출력기기는 주로 Magnetic Contact나 Solenoid와 같은 유도부하이며, 대소의 차는 있어도 돌입전류(Off On)나 역기전류(On Off)에 의한 Noise를 발생하고, 접점에서 Arch에 의한 Noise를 발생한다.

Noise 방지의 대원칙은 발생원에서 억제하는 것이다. 따라서 부하에 직접 CR식 Surge Killer나 다이오드를 접속하는 것이 좋다. 접점의 Arch Noise에 대해서는 접점간에 Surge Killer를 삽입하는데 이 경우에는 누설전류에 주의하지 않으면 안 된다. 동력전원에 대한 출력에는 반드시 Interposing Relay를 사용한다.



---

## 4장. X8 PLC 하드웨어

X8 PLC의 하드웨어 및 운영 대해서 설명합니다.

---

X8 PLC .....	4-2
Network PLC X8 의 미래 네트워크 .....	4-5
일반사양 및 신뢰성 규격 .....	4-6
PLC X8 외부 구성 .....	4-6
통신 케이블 .....	4-7
CPU모듈의 규격사양 .....	4-9
입력 사양 .....	4-10
출력 사양 .....	4-10
X8 CPU/확장I/O 결선 .....	4-12

## X8 PLC

X8 PLC는 RSAutomation에서 직접 개발한 제품으로 산업계에 널리 사용되고 있는 범용제품으로 브릭형 제품과 빌딩블럭형 제품의 장점을 모두 가지고 있으며 이더넷 포트를 내장하고 있어 통신에 관련하여 다양하게 응용될 수 있게 설계된 제품입니다.

### 모델

#### 1. CPU: 3종



X8-M32DDT



X8-M16DDR



X8-M14DDT

- X8-M32DDT : In 16, Out 16
- X8-M16DDR : In 8, Out 8 Relay
- X8-M14DDT : In 8, Out 6

Ethernet, USB, 고속 시리얼포트 기본 내장  
 Battery가 필요 없는 중형 PLC급 프로그램 메모리 지원  
 사용자 편의를 위한 기본 내장 - DC 파워, 디지털 입출력, 고속 카운터, PTO/PWM, RTC, SD 카드.

강력한 확장 모듈 지원 기능

RoHS 및 CE, KC, cULus 국제 안전 규격

#### 2. IO모듈



카탈로그번호	사양
디지털	
X8-XU16	16점싱크/소스 24V DC 입력모듈
X8-YN16	16점소스 24V DC 출력모듈
X8-YR16	16점 AC/DC 릴레이출력모듈
X8-XU32	32점싱크/소스 24V DC 입력모듈
X8-YN32	32점소스 24V DC 출력모듈
아날로그	
X8-AI8	8채널전압/전류아날로그입력모듈
X8-AO4	4채널전압/전류아날로그출력모듈
X8-AI4AO2	콤보 4채널입력 2채널출력전압/전류아날로그모듈
X8-RTD6	4채널 RTD/ 저항입력모듈
X8-TC6	4채널열전대/mV 입력모듈
포지션	
X8-POS1	1 AXIS 포지션 모듈 4MPPS
X8-POS2	2 AXIS 포지션 모듈 4MPPS
X8-POS4	4 AXIS 포지션 모듈 4MPPS
통신	
X8-SCU2	2PROT 422/484/232 시리얼
X8-RNM	리모트 마스트 유닛
X8-ETN2	2PROT EtherNet 유닛
파워	
X8-POWER3	POWER 모듈 24V 3A

### 3. Network 기능

국제 표준 산업용 Ethernet 프로토콜

- EtherNet/IP , Modbus/TCP
- PCCC over EtherNet/IP , SNMP, Bootp/DHCP, HTTP 등

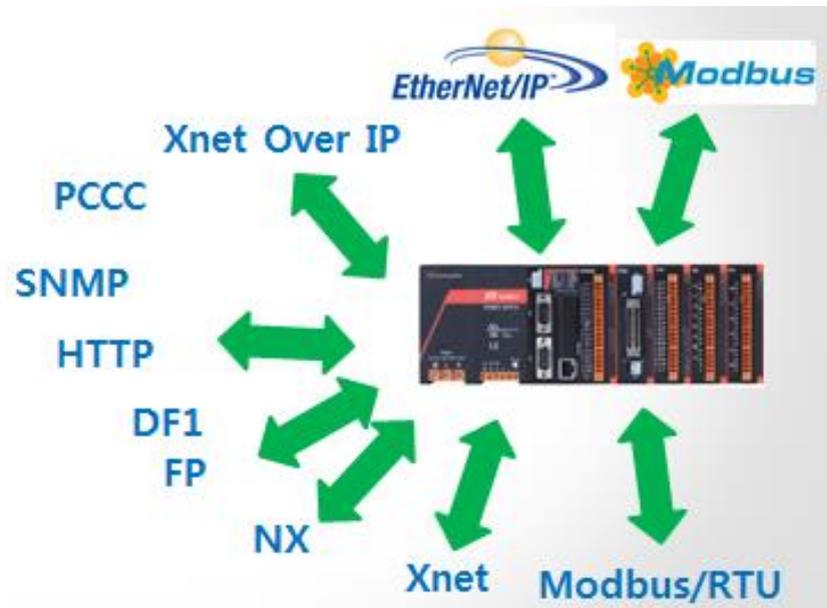
국제 표준 산업용 Serial 프로토콜

- Modbus/RTU
- DF1

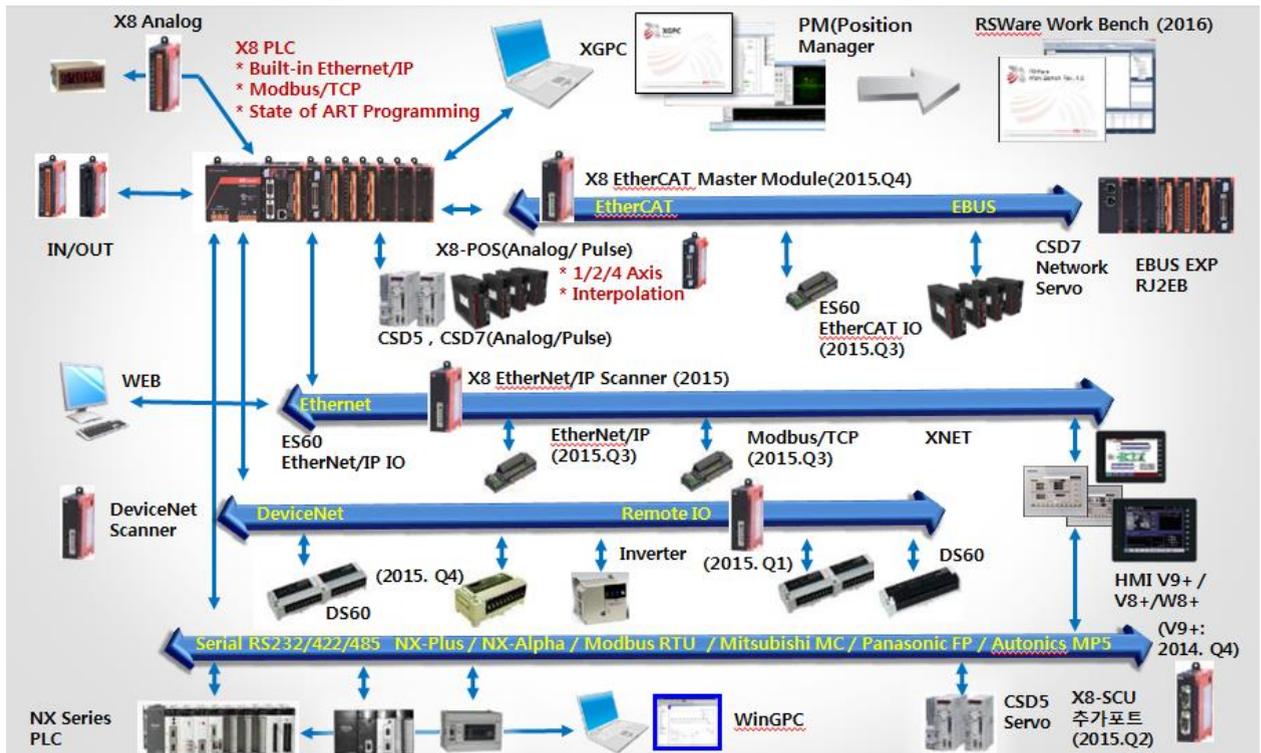
RSA 프로토콜

- Xnet over IP , Xnet
- NX Plus Protocol Master
- ASCII (User Defined Protocol)
- NX @ Protocol
- Panasonic(FP Series)

내장 Web Server



## Network PLC X8 의 미래 네트워크

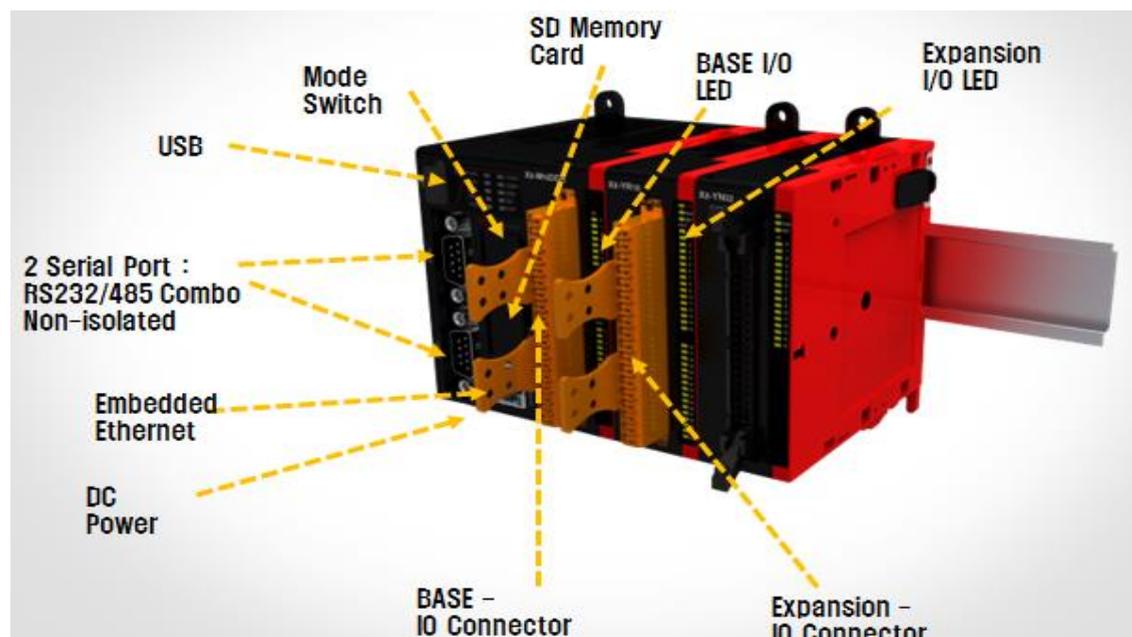


- Easy To Use
- 이 기종간의 통신
- Integration , 통합
- 호환성
- 표준 프로토콜 지원
- 강력한 성능
- Down-Sizing
- 경제적인 가격
- 확장성

## 일반사양 및 신뢰성 규격

항목	사양
사용온도/보존온도	-15~65°C, '-25~85°C Topr=-20~70°C, Tstg=-40~85°C
주변습도	30~85%RH(단, 이슬이 없을 때) 0~95%RH((단, 이슬이 없을 때)
내전압	I/O터미널(DC) <-> FrameGround(PowerUnit), AC500V1분간
절연저항	I/O터미널 <-> FrameGround(PowerUnit), 100MΩ 이상(DC500V메가에서)
내진동(Vibration)	(1)IEC 60028-2-6 (Test Fc, Operating) : <a href="#">3g@10...500Hz</a>
내충격(Shock)	Operating : IEC 60028-2-27 (Test Ea, Unpackaged Shock) : 30g NonOperating : IEC 60028-2-27 (Test Ea, Unpackaged Shock) : 50g(Panel mounting), 40g(Din mounting)
내노이즈성	모든 EMC 규격은 CE/EMC 기준에 따르며, 개발 스펙은 MLX와 동격
사용환경	부식성 가스가 없을 것, 먼지가 심하지 않을 것
규격인증	cULus, CE, KCC

## PLC X8 외부 구성



## 통신 케이블

RS X8 컨트롤러에는 다음과 같은 통신 케이블만 사용합니다. 이 케이블들은 Class I Division 2 어플리케이션에 사용됩니다.

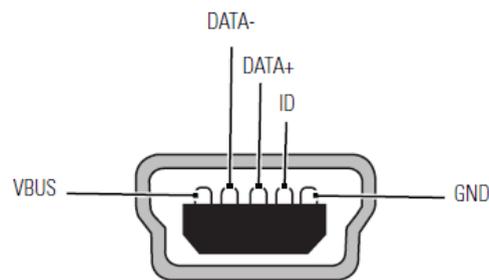
### 프로그래밍

RS X8 컨트롤러의 프로그램은 XGPC 1.00 이상으로 작성합니다. 프로그래밍에 사용하는 통신 케이블은 컨트롤러 및 소프트웨어와 별도로 사용할 수 있습니다.

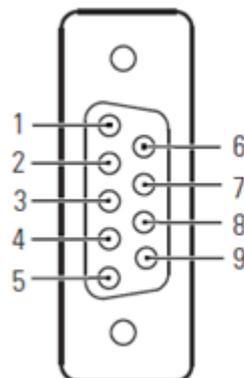
### 통신 옵션

RS X8 컨트롤러는 네개의 통신 포트 즉, USB포트(채널 0), 겸용 RS-232/485 통신 포트 (채널 1,2), Ethernet 포트(채널 3)를 지원합니다.

#### PLC PORT1 USB(MINI B TYPE) 각부분의 명칭



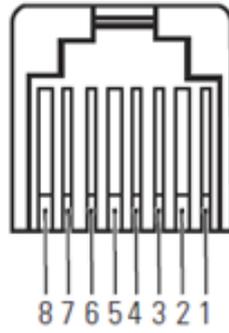
#### COM0/COM1(9PIN DSUB)



#### 각부 이름

No	X8 -COM1/2
1	DCD
2	RX
3	TX
4	485P +
5	GND
6	485N -
7	RTS
8	CTS
9	NC

#### EtherNet PORT(RJ45)



각부 이름

No	X8-RJ45
1	TD+
2	TD-
3	RD+
4	TCT
5	NC
6	RD-
7	TCT
8	NC

RS X8 의 채널 1 포트와 채널 2 포트를 다음과 같은 기기에 연결할 수 있습니다.

- 사용자 인터페이스, 개인용 컴퓨터 등. using DF1 Full Duplex point-to-point
- DH-485 네트워크
- DF1 무선 모뎀 네트워크
- RTU 마스터 또는 RTU 슬레이브로 사용하는 DF1 반 이중 네트워크
- RTU 마스터 또는 RTU 슬레이브로 사용하는 Modbus 네트워크
- ASCII 네트워크
- Ethernet 포트를 사용하는 Ethernet 네트워크

슬레이브로 사용하는 DNP3 네트워크

DH-485, DF1 반 이중 마스터 /슬레이브, Modbus RTU 마스터 / 슬레이브 또는 DNP3 슬레이브 프로토콜을 사용하여 RS-485 네트워크에 연결할 때, RS X8을 고급 인터페이스 컨버터 (Advanced Interface Converter, 카탈로그 번호 X8-AIC) 없이 채널 1로 직접 연결할 수 있습니다. 채널 1콤보 포트는 RS-232와 RS-485 연결을 둘 다 제공합니다. 사용자가 통신 케이블을 선택하면 그것에 맞추어 전기 인터페이스가 선택됩니다. 통신 케이블이 RS-232 드라이버에 인터페이스를 제공합니다.

채널 1 구성 설정 값으로 바코드 리더, 저울, 직렬 프린터 및 기타 지능형 기기에 ASCII 를 사용하여 연결할 수 있습니다. RS X8 을 채널 0을 통해 RS-485 네트워크에 ASCII를 사용하여 직접 연결할 수 있습니다.

RS X8은 Ethernet 통신 채널 3을 통해 Ethernet 통신을 지원합니다. 10 Mbps 또는 100 Mbps로 다양한 기기 간에 통신을 제공하는 LAN에 컨트롤러를 연결할 수 있습니다. 이 포트는 CIP explicit 메시지 기능 (메세지 교환)만을 지원합니다. 컨트롤러는 CIP implicit 메시지 기능(실시간 I/O 메시지 기능)에 사용할 수 없습니다. 또한 컨트롤러는 표준 웹 브라우저로 모듈정보, TCP/IP 구성 및 진단 정보를 보여주고, 데이터 테이블 맵 과 데이터 테이블 감시 화면을 가지고 있는 웹 서버를 내장하고 있습니다. 사용할 수 있는 통신 옵션의 연결에 대한 자세한 정보는 4장을 참조하십시오.

## CPU모듈의 규격사양

### CPU 성능규격

설명	X8-14DDT	X8-16DDR	X8-32DDT
연산방식	반복연산, 정주기연산, 인터럽트 연산		
입출력 제어방식	스캔동기 일괄처리, 명령어에 의한 다이렉트방식		
프로그램언어	Ladder Diagram		
I/O 수	IN:DC INPUT 8P OUT:DC OUTPUT 6P	IN:DC INPUT 8P OUT:AC/DC OUTPUT 8P	IN:DC INPUT 16P OUT:DC OUTPUT 16P
전원 공급 장치 전압	24 V dc(-15%, +10%) Class 2 SELV		
방열	공냉식		
전원 공급 장치 돌입 전류	24 V dc: 20 ms 동안 15 A		
소비 전력	7.5~53 W		
24 V dc 센서 전원	없음		
입력 회로 타입	디지털 : 24 V dc 싱크/ 소스(표준 및 고속)		
출력 회로 타입	FET(싱크)	릴레이	FET(싱크)
외함 타입 등급	없음 (개방형)		
단자 나사 토크	0.791 Nm(7.0 in-lb) 정격		

## 입력 사양

종류	DC INPUT
On 상태 전압 범위	4.5~24 V dc (14~26.4 V dc(+10%) @ 65 °C/149 °F) (14~30 V dc(+25%) @ 30 °C/86 °F)
Off 상태 전압 범위	0~5 V dc
작동 주파수	0 Hz~100 kHz
On 상태 전류 최소값	3.0 mA @ 10 V dc
공칭값	5.0 mA @ 24 V dc
최대값	5.5 mA @ 30 V dc
Off 상태 누출 전류	최대 0.1 mA
공칭 임피던스	2.0 kΩ
최대 돌입 전류 @ 120 V ac	30 A

## 출력 사양

### 릴레이 및 FET 출력

설명	FET	
최대 제어 부하	1440 VA	
최대 연속 전류		
그룹 Common당 전류	5 A	
컨트롤러당 전류	@ 최대 150 V	30 A 또는 접점당 총 부하 중 작은 것
	@ 최대 240 V	20 A 또는 접점당 총 부하 중 작은 것

### 릴레이 출력

설명	Relay
Turn On 시간/Turn Off 시간	최대 10 ms <sup>(1)</sup>
부하 전류	최소 10 mA

(1) (스캔 시간에 따라 다름)

설명	일반 작동	고속 작동(1) <sup>(1)</sup> (출력 2와 3 전용)
전원 공급 장치 전압	12/24 V dc(-15%, +10%)	
On 상태 전압 저하 :		
최대 부하 전류 시	1 V dc	사용 불가
최대 서지 전류 시	2.5 V dc	사용 불가
접점당 정격 전류		
최대 부하	아래 그래프 참조	100 mA
최소 부하	mA	10 mA
최대 누출	1.0 mA	1.0 mA

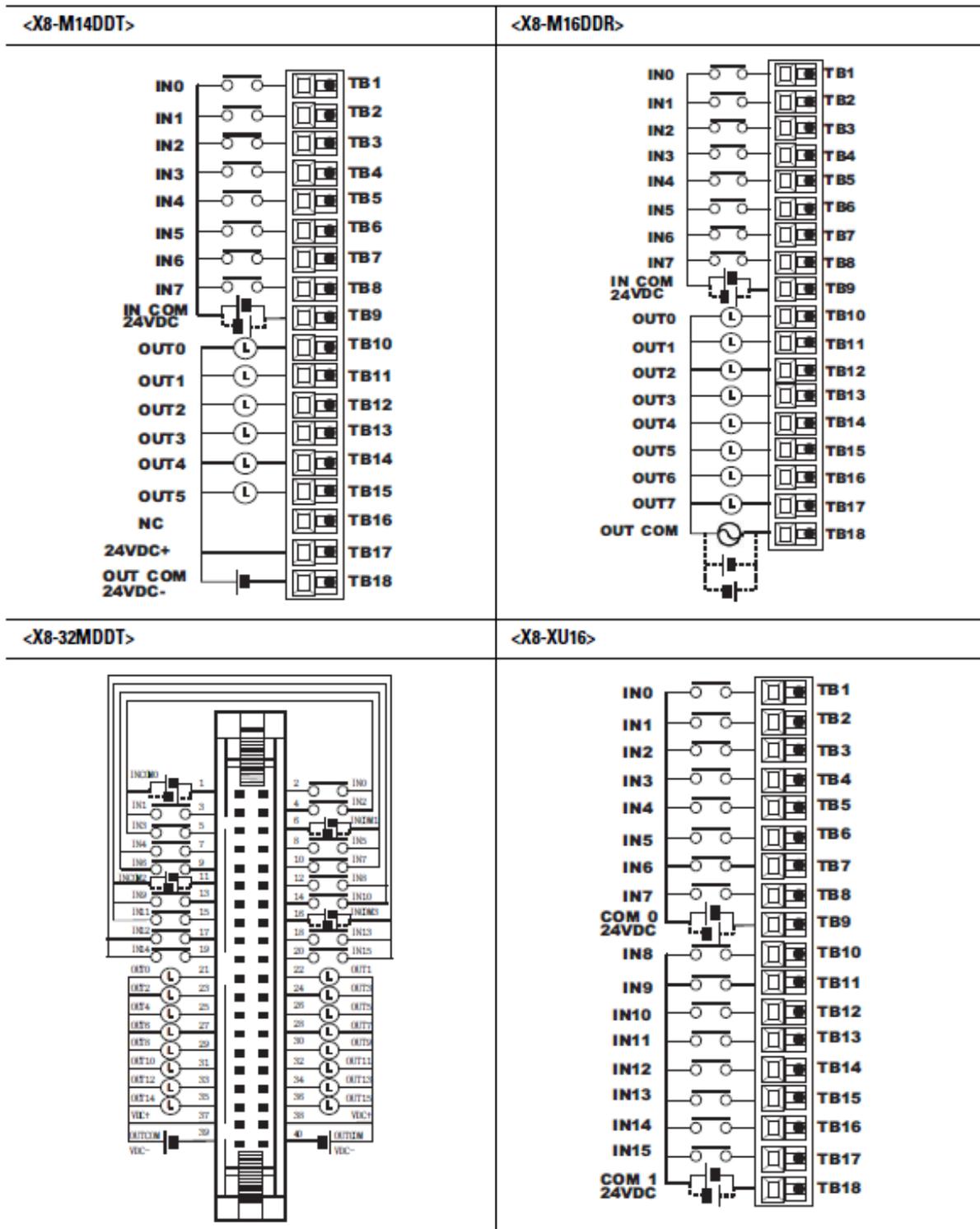
최대 출력 전류 (온도에 따라 다름) :

출력 2와 3은 다른 FET 출력을 통해 향상된 기능을 제공하도록 설계되었습니다. 출력 2와 3은 다른 FET 트랜지스터 출력처럼 사용할 수 있지만, 이에 더하여 제한된 전류 범위 내에서 더 높은 속도로 작동할 수 있습니다. 또한 출력 2와 3은 펄스 열 출력 (PTO) 또는 펄스 폭 변조 출력 (PWM) 기능도 제공합니다.

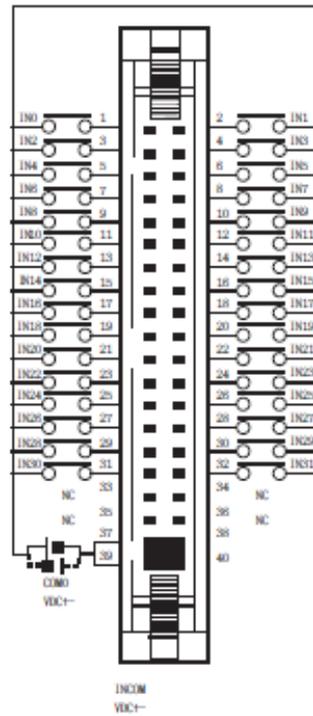
## X8 CPU/확장I/O 결선

디지털 결선도

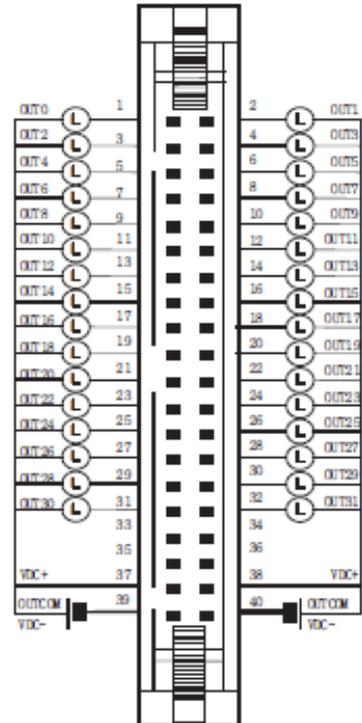
아래 그림은 디지털 확장 I/O 결선도 입니다.



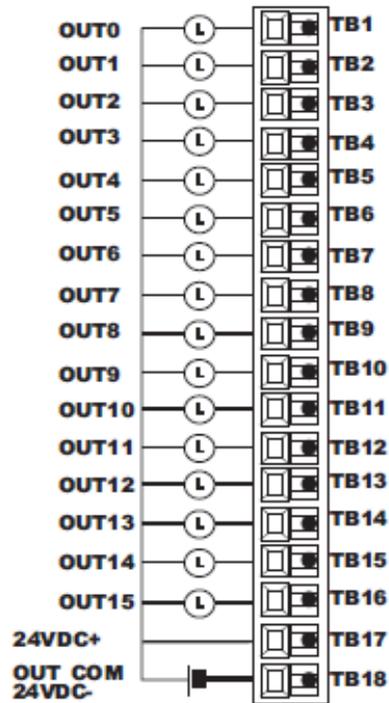
<X8-XU32>



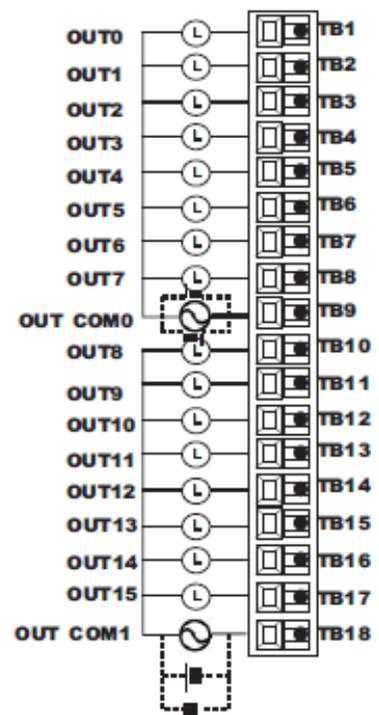
<X8-YN32>



<X8-YN16>



<X8-YR16>



## 아날로그 결선

아날로그 모듈을 결선할 때에는 다음 사항에 유의하십시오.

- 아날로그 공통 (COM) 은 모듈 내부의 대지 접지에 연결합니다. 단자들은 모두 시스템과 전기적으로 절연되어 있습니다.
- 채널들은 서로 절연되어 있지 않습니다.
- .Belden 8761 또는 이에 상당하는 실드선을 사용하십시오.
- 표준 상태에서 접지선 (실드)을 금속 장착판 (대지 접지)에 연결해야 합니다. 대지 접지에 연결하는 실드선은 최대한 짧게 유지하십시오.
- 전압 타입 입력에 최적의 정확도를 확보하기 위해 모든 아날로그 케이블을 최대한 짧게 유지하여 전체 케이블 임피던스를 제한하십시오. I/O 시스템을 전압 타입 센서 또는 액추에이터에 최대한 가까운 곳에 놓으십시오.
- 모듈은 아날로그 입력에 루프 전원을 공급하지 않습니다. 입력 트랜스미터 사양에 맞는 전원 공급 장치를 사용하십시오.

## X8-ANALOG 입력 타입 선택

모듈의 구성 데이터 파일 (Configuration Data File)에 있는 입력 타입 /범위 선택 비트를 사용하여 입력 타입 (전류 또는 전압)을 선택하십시오.

RS X8 PLC 명령어 세트 참조 매뉴얼 (Publication X8-RM001)을 참조하십시오.

## X8-ANALOG 출력 타입 선택

출력 타입 (전류 또는 전압)은 적절한 단자 Iout 또는 Vout에 결선하여 그리고 구성 데이터 파일 (Configuration Data File)에 있는 타입 /범위 선택 비트를 사용하여 선택할 수 있습니다. RS X8 PLC 명령어 세트 참조 매뉴얼 (Publication X8-RM001)을 참조하십시오.

### 주의

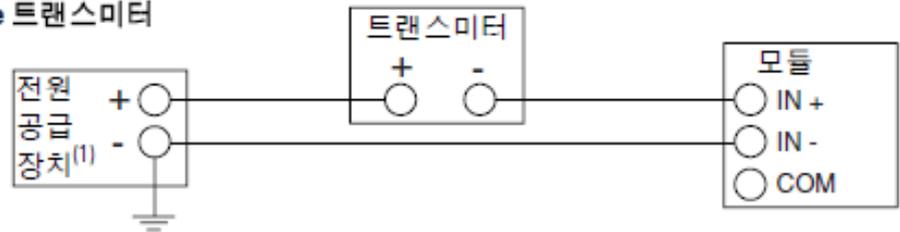


아날로그 출력은 전원을 인가하거나 제거할 때 잠깐 동안 (1초 이내) 변동할 수 있습니다. 이것은 대부분의 아날로그 출력에 공통된 특성입니다. 대부분의 부하가 이 짧은 신호를 검출하지 못하지만, 그렇더라도 연결된 장비가 영향을 받지 않게 하기 위해 방지 수단을 확보하는 것이 좋습니다.

## 단자대 구성

### 단일 종단 센서/트랜스미터 타입

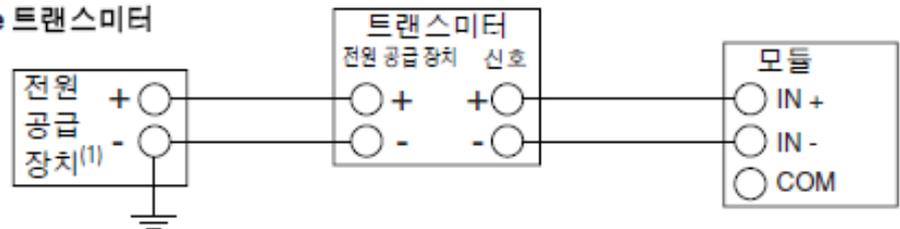
#### 2-Wire 트랜스미터



#### 3-Wire 트랜스미터



#### 4-Wire 트랜스미터



(1) 전원 공급 장치는 모두 N.E.C. Class 2 정격입니다.

### 팁

일반적으로 케이블 실드를 모듈 종단에만 접지해도 충분한 내노이즈성을 제공합니다. 그러나 최상의 케이블 실드 효율을 얻기 위해 양쪽 끝을 모두 대지 접지하십시오. 필요한 경우에는 AC 전원 접지 전류를 막기 위해 0.01 $\mu$ F 커패시터를 사용하십시오.

1. 전원 공급 장치는 모두 N.E.C. Class 2 정격입니다.

#### X8-AO4 출력 타입 선택

출력 타입 (전류 또는 전압)은 적절한 단자 Iout 또는 Vout에 결선하여 그리고 구성 데이터 파일 (Configuration Data File)에 있는 타입 /범위 선택 비트를 사용하여 선택할 수 있습니다.



## 5장. XGPC 운영

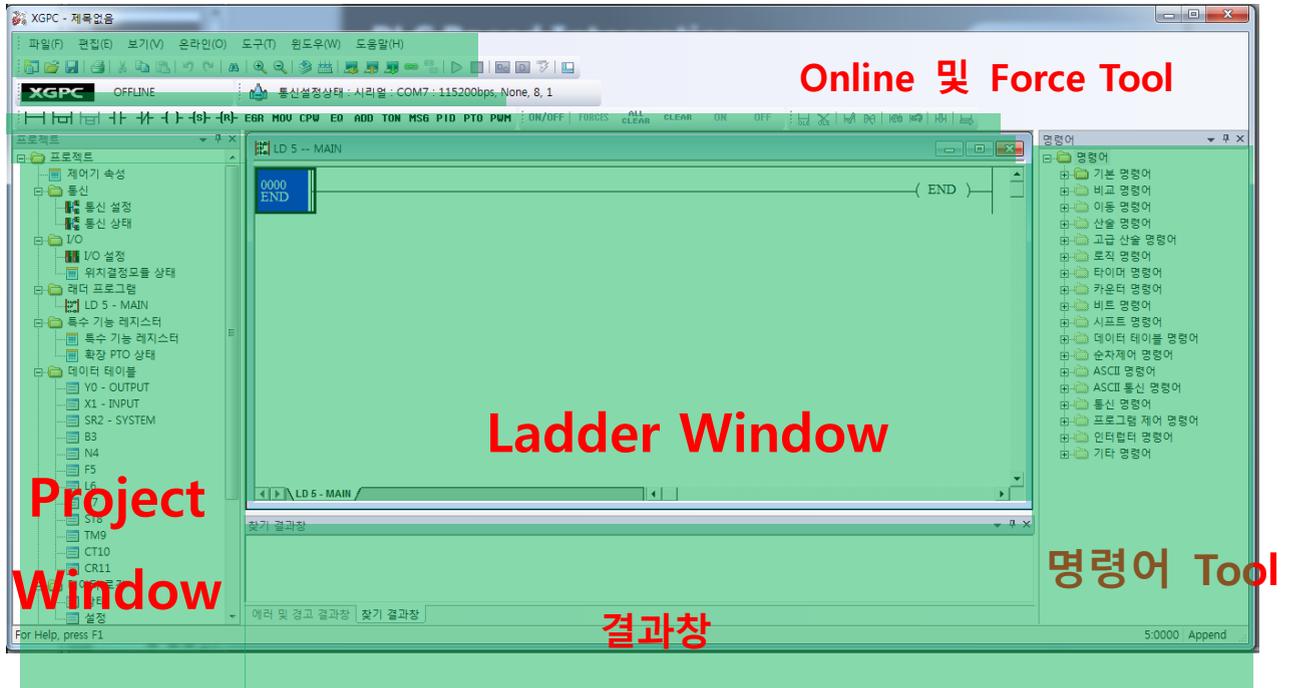
본 장에서는 X8 PLC의 운영소프트웨어 XGPC대해서 설명합니다.

---

XGPC Software 구성 화면 및 기능 설명.....5-2

## XGPC Software 구성 화면 및 기능 설명

### 기본 메뉴 및 Icon Tool



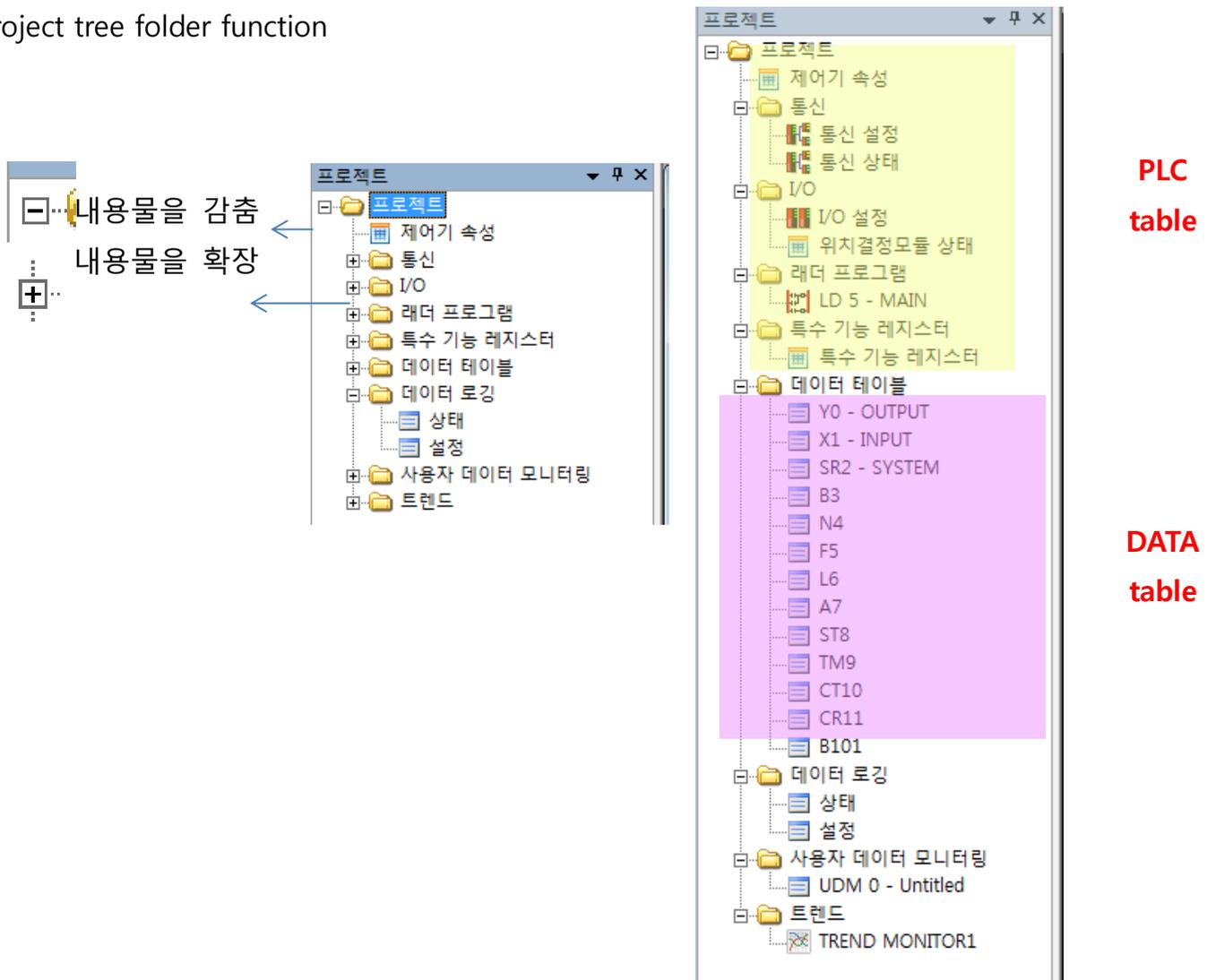
### Online Toolbar 정보



- PLC의 가동 상태
- Online ladder program에서 편집 상태
- Ladder program에서 force 존재
- Ladder program에서 force 상태
- 현재 driver 정보
- PLC의 노드 수

## Project Window

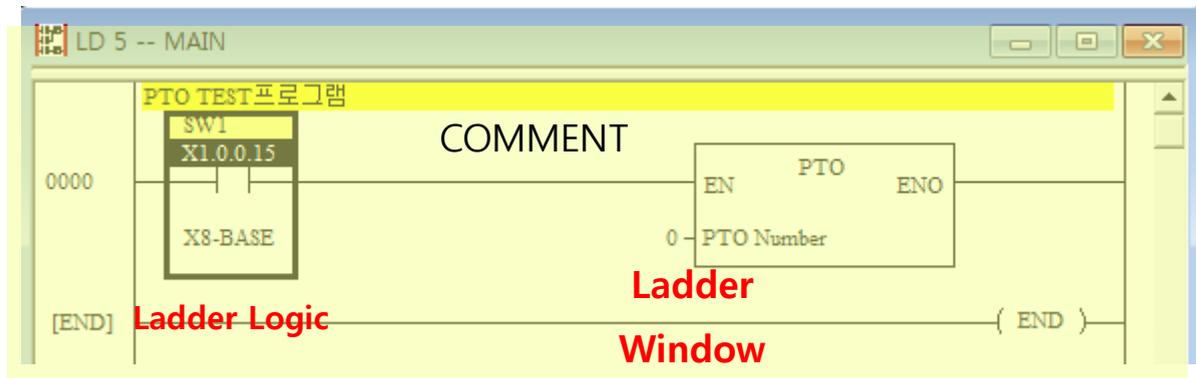
Project tree folder function



## Project Tree Actions

- Create data table
- Delete data table
- Copy data table
- Open data table
- Rename data table

**Open Ladder Program**



**툴바설명**

**기본툴바**

XGPC는 기본툴바에서 다음과 같은 기본 버튼을 제공합니다



아이콘	설명	단축키	아이콘	설명	단축키
	새 프로젝트를 생성합니다.	Ctrl + N		프로그램을 컴파일 합니다.	
	프로젝트를 엽니다.	Ctrl + O		프로젝트를 컴파일 합니다.	
	현 프로젝트를 저장합니다.	Ctrl + S		통신을 설정합니다.	
	현 프로젝트를 출력합니다.			프로젝트를 업로드 합니다.	
	잘라내기를 합니다.	Ctrl + X		프로젝트를 다운로드 합니다.	
	복사를 합니다.	Ctrl + C		프로젝트를 온라인합니다.	
	붙여넣기를 합니다.	Ctrl + V		프로젝트를 오프라인합니다.	
	실행 취소를 합니다.	Ctrl + Z		런 모드로 변경합니다.	
	이전 작업을 다시 실행합니다.	Ctrl + Y		프로그램 모드로 변경합니다.	
	오퍼랜드를 찾습니다.	Ctrl + F		연속 디버깅 모드로 변경합니다.	
	래더 편집을 확대합니다.			단일 디버깅 모드로 변경합니다.	
	래더 편집을 축소합니다.			Fault 비트를 초기화 합니다.	

### 상태표시바

X8 PLC의 현재 상태를 표시해주는 툴바입니다.

#### PLC MODE별 상태표시

X8 PLC의 모드별 상태표시창에 다른 아이콘색깔로 구분할수 있습니다.

- RUN MODE



- REMOTE MODE



- PROGRAM MODE



- 컬러별 구분

상태	설명
OFFLINE	오프라인 모드
RUN	런 모드
REMOTE RUN	
PROGRAM	프로그램 모드
REMOTE PROG	
SUSPEND	Suspend 모드
FAULTED	Fault 모드

### 명령어 툴바

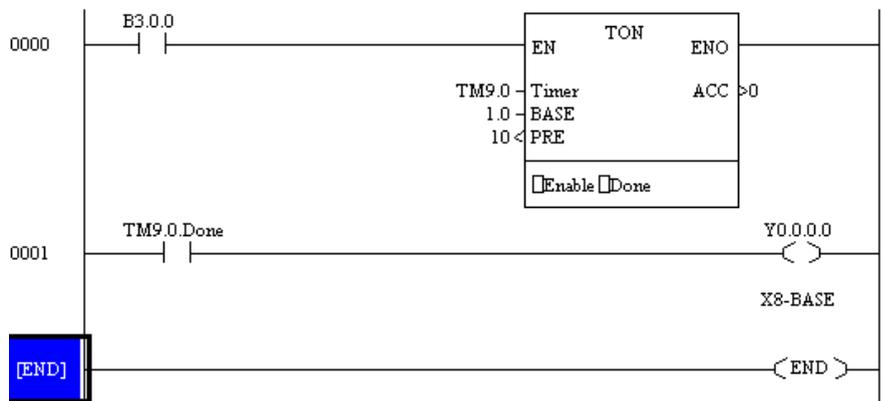
기본명령어를 모아 편리하게 만들어놓은 툴바



### 래더 만들기

1. 명령어바에서 령버튼을 두번 클릭하여 령을 2개 만듭니다.
2. 첫번째 령을 마우스로 선택합니다. A 접점을 만들기 위해서 -|- 버튼을 누릅니다. 편집창이 나타나면 "B3.0.0"를 입력하고 엔터 키를 누릅니다.
3. 타이머 명령어를 만들기 위해서 TON 버튼을 누르고, 편집창에 TM9.0를 입력하고, PRE에는 10을 입력하고 엔터키를 누릅니다.
4. 두번째 령을 선택합니다. A 접점과 OUT 명령어를 만들기 위해서 -|- 와 -( )-버튼을 눌러 명령어를 만든후에 아래 그림과 같이 오퍼랜드를 입력합니다.

위의 모든 과정을 수행하였다면, 다음과 같은 래더 프로그램이 만들어 집니다.



### 도구모음 및 상태바

#### 비트 반전/강제출력 상태 툴바

XGPC는 비트 반전/강제출력 상태 툴바에서 비트 상태를 반전시키거나, 입력/출력 데이터 테이블에서 강제출력을 설정할 수 있습니다.



버튼에 대한 상세한 설명은 다음과 같습니다.

아이콘	설명
	비트 상태를 반전합니다.
	강제출력 상태를 보여줍니다.
	모든 강제출력을 삭제합니다.
	하나의 강제 출력을 삭제합니다.
	I/O 데이터 테이블에서 강제로 ON 상태로 만들어 줍니다.
	I/O 데이터 테이블에서 강제로 OFF 상태로 만들어 줍니다.

### 온라인 편집 툴바

XGPC는 온라인 편집 툴바에서 다음과 같은 버튼을 제공합니다.



아이콘	설명
	온라인 편집 시작
	온라인 편집 취소
	신규 링 다운로드
	온라인 편집 링 제거
	신규 링 실행
	이전 링 실행
	신규 링 적용

### 에러 결과창

작성된 래더 프로그램을 컴파일하면, XGPC는 래더 프로그램을 PLC코드로 변경하게 됩니다. 이 래더 프로그램에 에러를 포함하고 있으면 에러 결과창에 에러 메시지가 표시됩니다. 에러 결과창에 있는 에러 메시지를 더블 클릭하게 되면 에러 링 또는 에러 명령어로 이동하게 됩니다. 이 기능은 에러에 대한 디버깅을 용이하게 합니다. 에러를 가진 링은 링 앞에 'e' 표시가 나타납니다. 'e'표시는 컴파일이 안되었을 경우도 나타납니다.

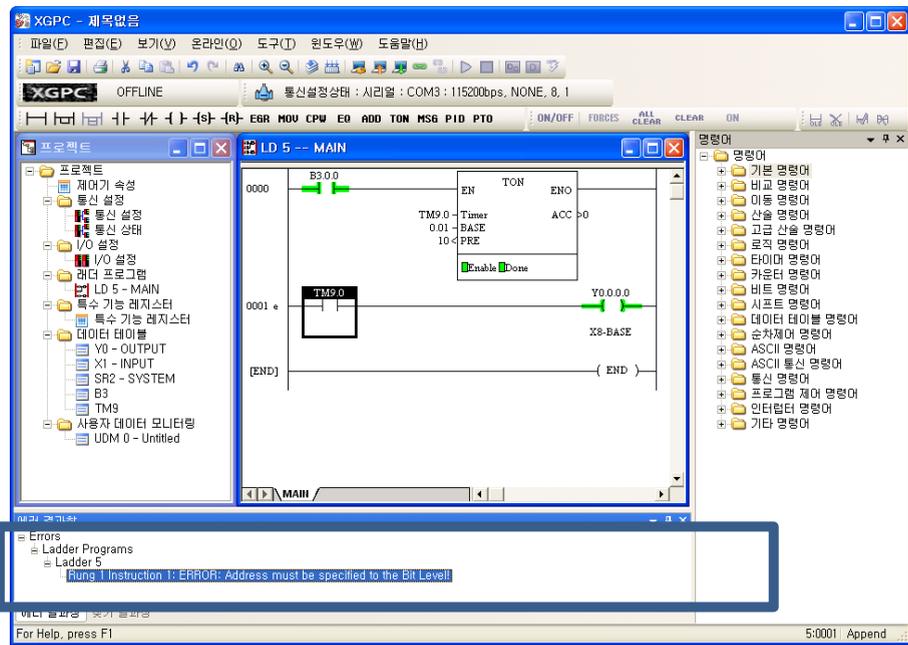


그림 1 래더 검사중 에러 발견

에러가 없다면 그림 2와 같이 결과가 표시됩니다

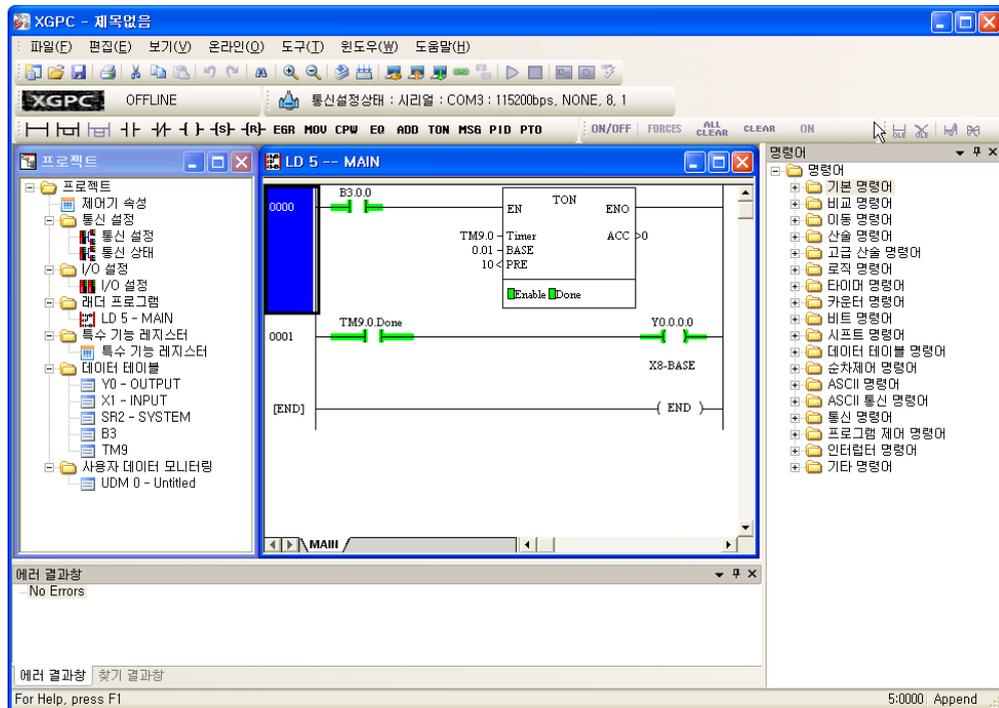
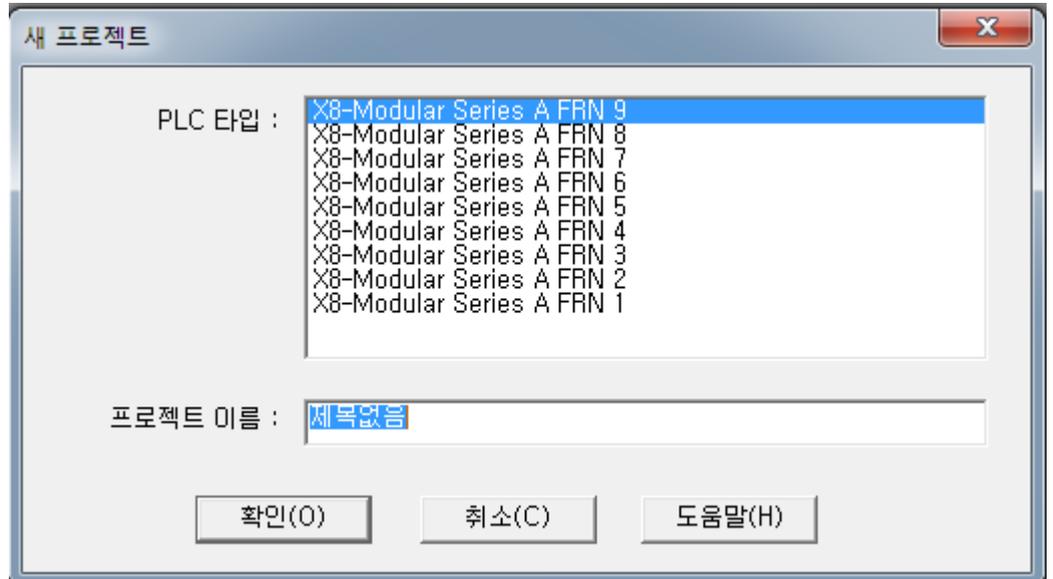


그림 2. 검사결과 에러 없음

### New 프로젝트 만들기

프로젝트를 만들기 위해서 메뉴 > 파일 > 새 프로젝트...  
또는 기본 툴바에 있는 버튼을 선택합니다.



#### 프로젝트 이름 입력

PLC 타입과 프로젝트 이름을 입력합니다.

##### 1. PLC 타입

새 프로젝트에서 동작하게 될 PLC 타입을 선택합니다.

##### 2. 프로젝트 이름

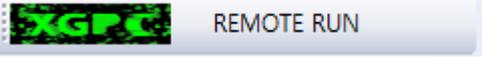
프로젝트의 이름을 입력합니다. 생성되는 파일은 프로젝트명에 확장자 XLD가 붙습니다.

3. FRN은 펌웨어 버전이 올라가면서 기능이 추가된 내용이며 높을수록 많은 기능을 활용할 수 있습니다.

단 CPU의 펌웨어 버전이 동일하게 버전업이 되어 있어야 사용 가능합니다.

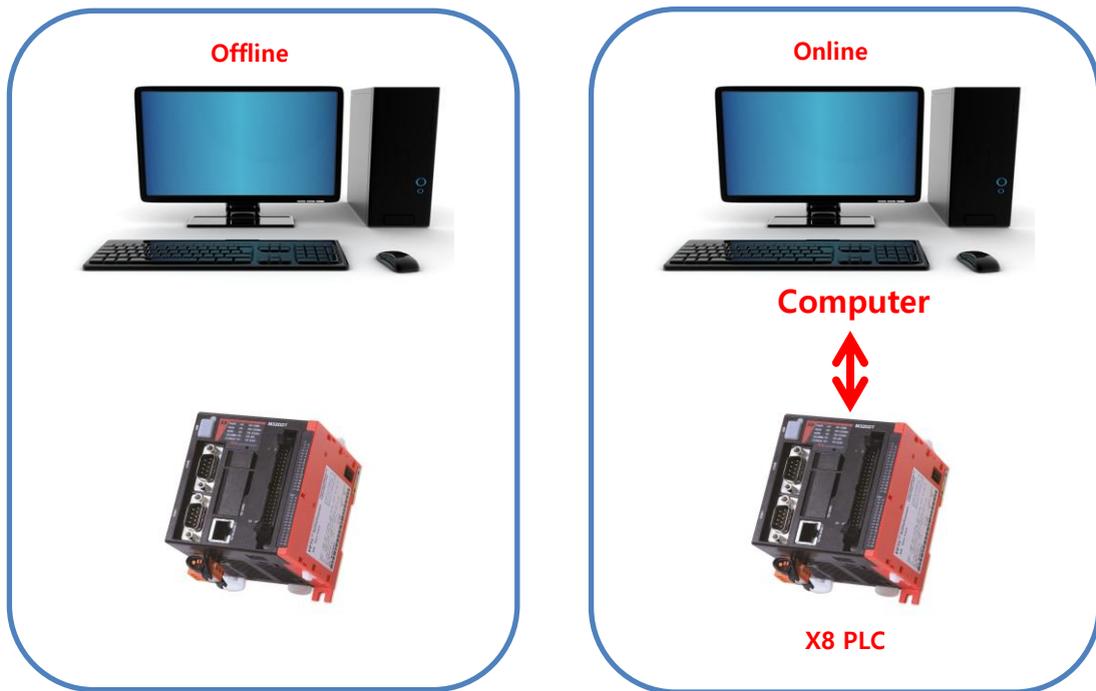
### ONLINE 하기

 온라인 및 오프라인 상태 버튼  
PLC와 통신하고 있지 않으면 온라인 상태/모드 툴바에  
 과 같이 표시되고, 온라인  
하게 되면

 과 같이 표시 됩니다.

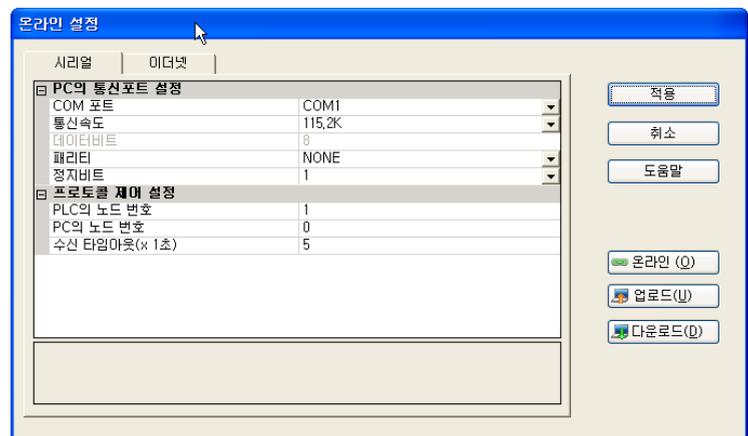
래더 UPLOAD 및 다운로드를 RS-232/485 통신,Ethernet 통신

RS-232 비절연 USB 통신 모두 가능합니다.



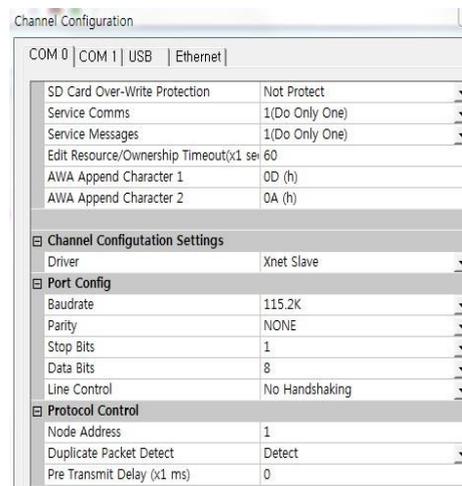
## XGPC Software 온라인 위한 설정

- XGPC는 사용가능한 COM 포트 또는 이더넷 포트를 보여줍니다. 이더넷 포트는 멀리서 PLC에 연결할 수 있도록 도움을 줍니다.
- PLC에 연결하기 위해서, PLC의 연결 포트, 보드레이트, 데이터 비트수, 패리티 비트 등을 선택합니다. 그리고, PC와 PLC를 통신케이블로 연결합니다.
- 아래와 같은 통신설정을 하기 위해서는 기본 툴바에서 "온라인 설정"을 선택하거나 메뉴에서 온라인 > 온라인 설정(O)을 선택하면 됩니다.
- Online Settings



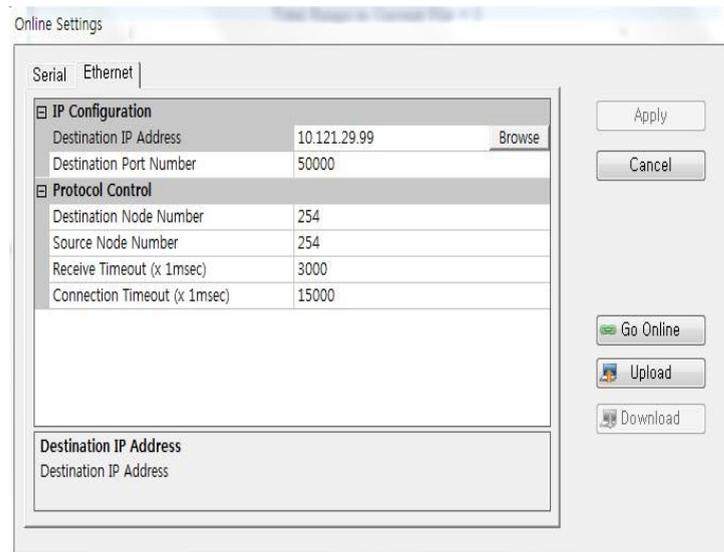
PC에 대한 통신 설정값 입력

- Serial 설정

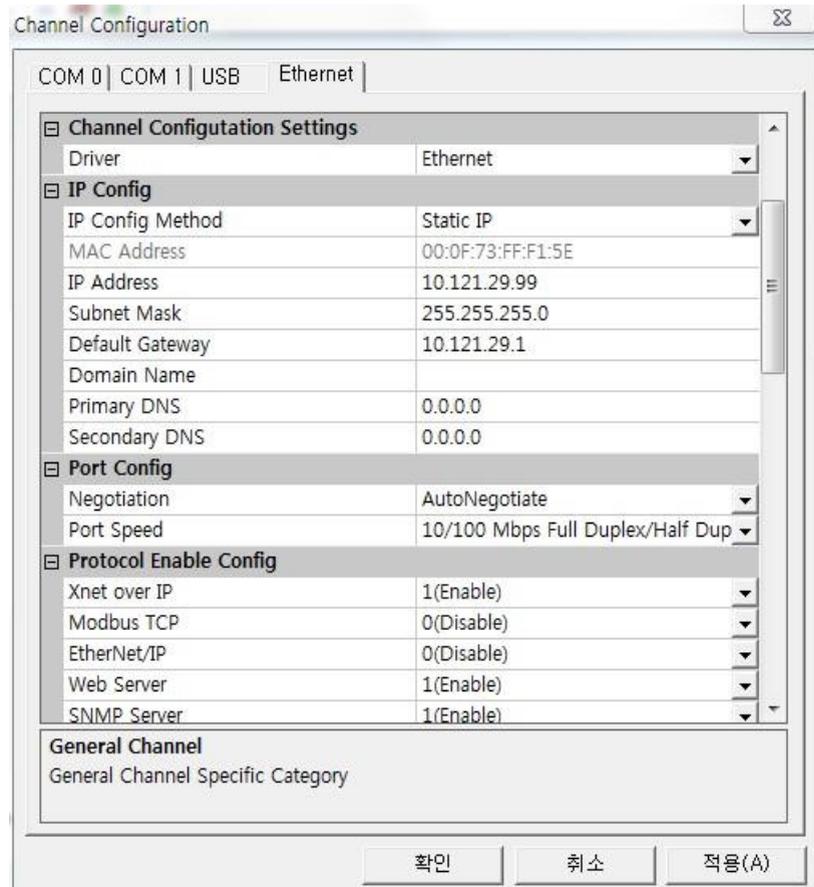


PLC에 대한 설정값을 입력

- Ethernet 설정

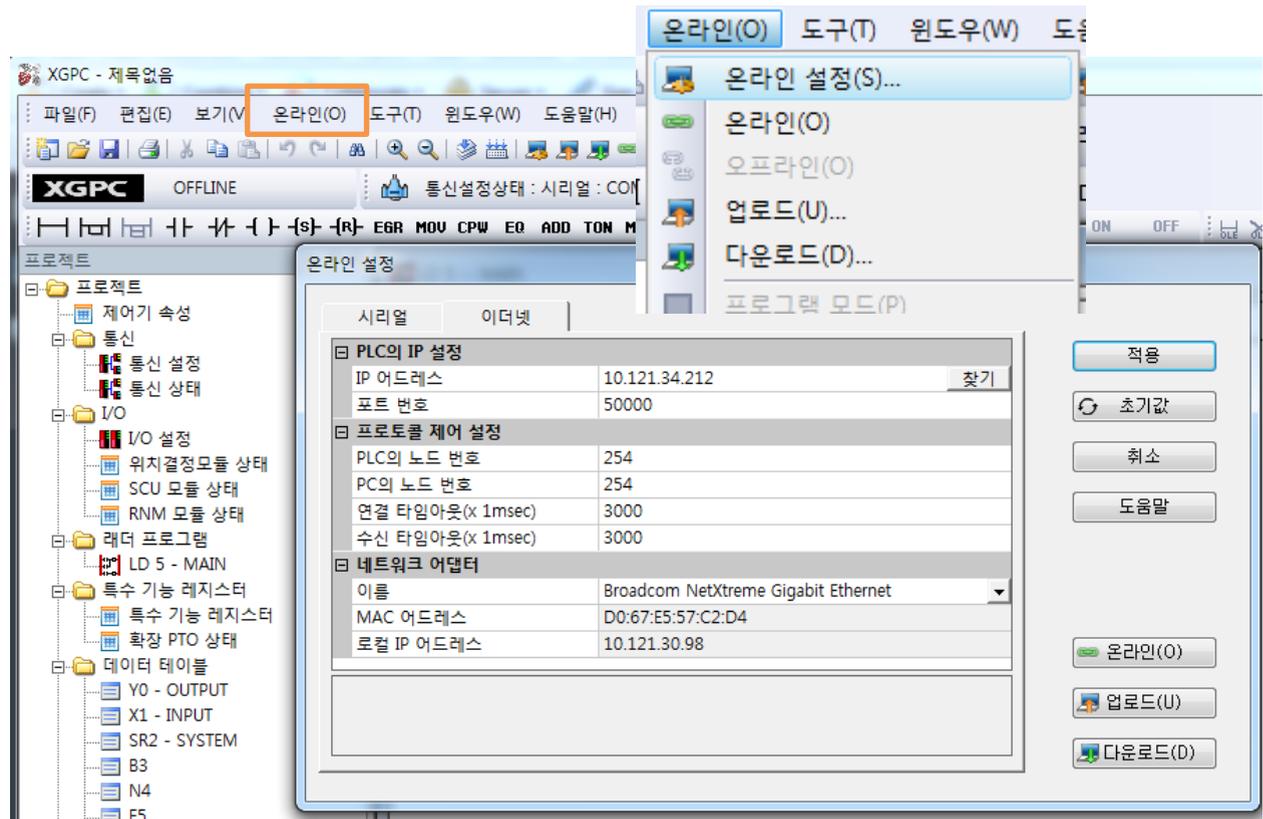


PC에 대한 통신 설정값 입력



PLC에 대한 설정값을 입력

## PC에서 PLC로 Online 연결

온라인 설정.

온라인 설정은 PC에서 PLC와 통신하기 위해서 시리얼 포트 설정 또는 이더넷 설정을 하는 것을 말합니다. "온라인" 버튼을 누르면 현재 설정을 가지고 PLC와 모니터링 하려고 합니다. "업로드" 버튼을 누르면 PLC에 있는 래더 프로그램을 업로드 하고자 합니다. "다운로드" 버튼을 누르면 현재 작업중인 래더 프로그램을 PLC로 다운로드 하고자 합니다. "적용" 버튼을 누르면 현재 설정된 통신 설정을 기억합니다.

**주의**

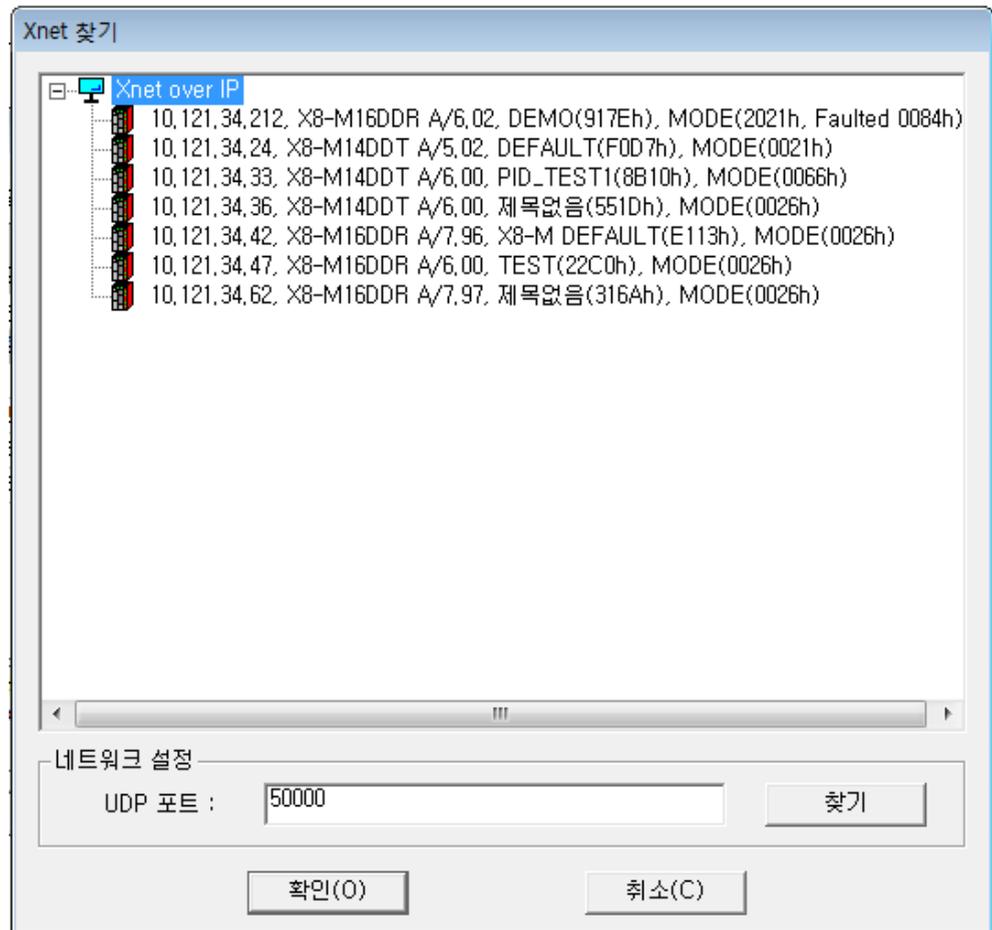
**업로드 : PLC의 프로젝트를 PC로 업로딩**

**다운로드 : PC에 있는 프로젝트를 PLC로 다운로드**

**항상 PC 클라이언트로 처리하여 상위로 처리함. (일반적인 인터넷은 서버가 상위임.)**

### 이더넷 연결 X8 통신 찾기

이더넷에 연결된 X8의 IP를 직접 몰라도 IP옆에 있는 찾기버튼으로 로컬에 같이 있는 그룹을 확인하고 연결할수 있습니다.



#### 주의.

연결이 보지 않는 경우는 컴퓨터 방화벽 문제로 제어판에서 XGPC를 활성화 시켜주시면 됩니다.

UDP포트는 꼭 50000으로 설정하셔야합니다.

### 온라인상태에서 편리한 모니터링 기능

래더 모니터링 이나 데이터 테이블을 이용해서 값을 모니터링 할 수 있습니다. 현재의 래더프로그램과 PLC의 래더 프로그램이 같으면 래더 모니터링을 할 수 있습니다. (XGPC가 체크해서 PLC 와 현재의 프로젝트가 틀릴경우 업로드해서 모니터링하게 됩니다.)

### 래더 모니터링

래더 모니터링하고자 하는 래더를 선택합니다. 모니터링 동안 ON이된 비트 값은 녹색으로 표시가 되고, 데이터 값은 계속 업데이트 됩니다.

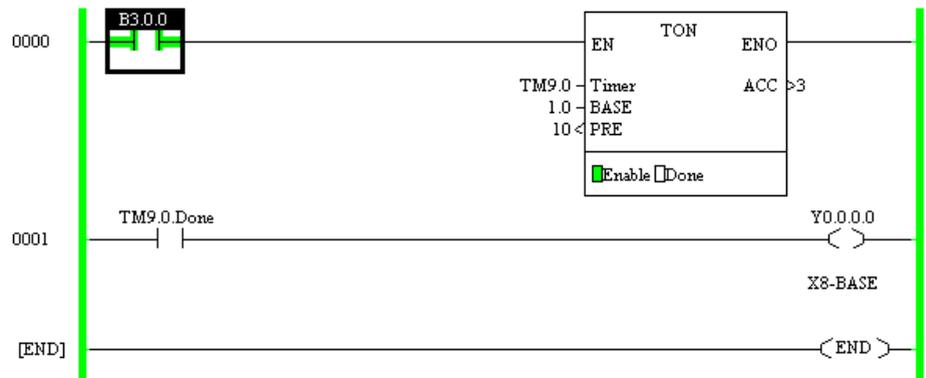


그림 1. 래더 모니터링

### 데이터 테이블

데이터 테이블 모니터링은 PLC에 있는 데이터 테이블을 모니터링 하는 것입니다. 데이터 모니터링 중 원하는 값으로 변경도 가능합니다. 입력/출력 데이터 테이블은 강제출력 기능을 이용해서 래더를 디버깅할 수 있습니다.

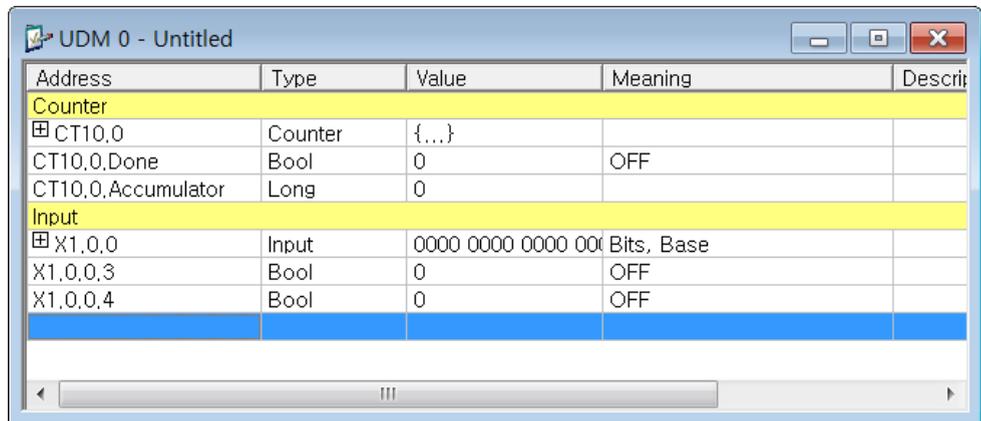
주소	데이터 타입	값	의미	설명
Y0.0.0	Output	0000 0000 0000 0001	Bits, Base	
.0	Bool	1	ON	
.1	Bool	0	OFF	
.2	Bool	0	OFF	
.3	Bool	0	OFF	
.4	Bool	0	OFF	
.5	Bool	0	OFF	
.6	Bool	0	OFF	
.7	Bool	0	OFF	
.8	Bool	0	OFF	
.9	Bool	0	OFF	
.10	Bool	0	OFF	
.11	Bool	0	OFF	
.12	Bool	0	OFF	
.13	Bool	0	OFF	
.14	Bool	0	OFF	
.15	Bool	0	OFF	
Y0.0.1	Output	0000 0000 0000 0000	Bits, Base	

그림 2. 출력 데이터 테이블

USER 데이터 테이블

유저 데이터 테이블 모니터링은 PLC에 있는 데이터 테이블을 유저가 등록하여 모니터링하고자 하는 것입니다. 데이터 모니터링 중 원하는 값으로 변경도 가능합니다. 입력/출력 데이터 테이블은 강제출력 기능을 이용해서 래더를 디버깅할 수 있습니다.

UDM (User Data Monitor)

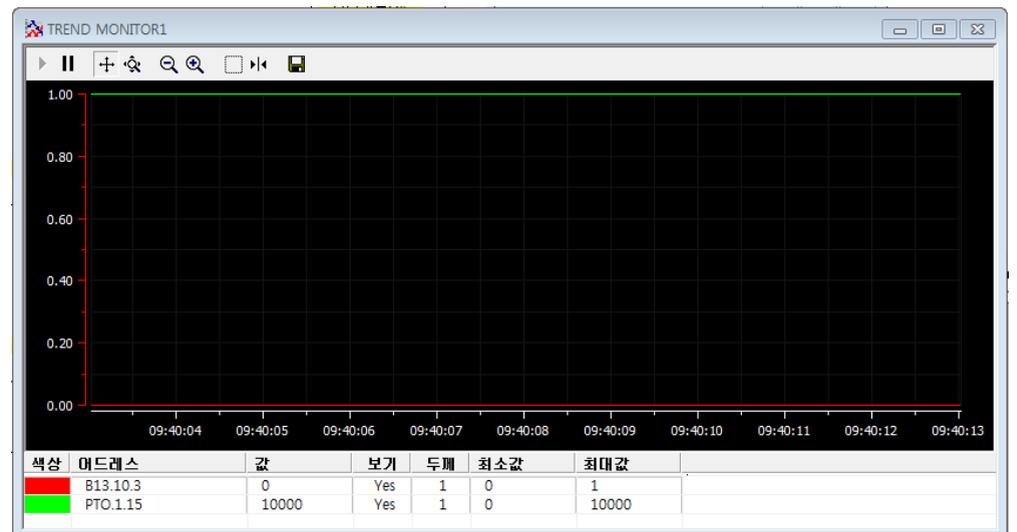


트랜드 모니터링

트랜드 모니터링은 PLC에서 처리되는 데이터값들을 연속적으로 모니터링할 때 유용합니다.

일시적으로 변경되거나 변경되는 시점을 모니터링할 때 같이 복수로 설정하여 동시에 모니터링이 가능합니다.

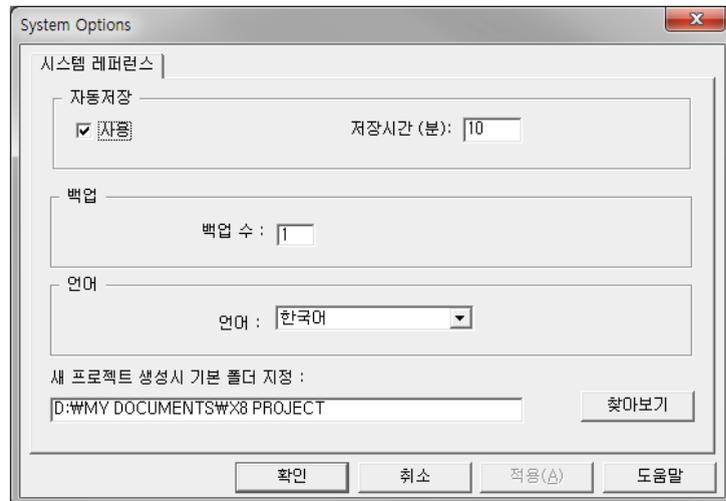
트랜드 모니터링 창 또한 복수로 등록할 수 있어 필요한 데이터만 모아서 따로 창을 관리 할 수 있습니다.



## 도구 메뉴

### OPTION

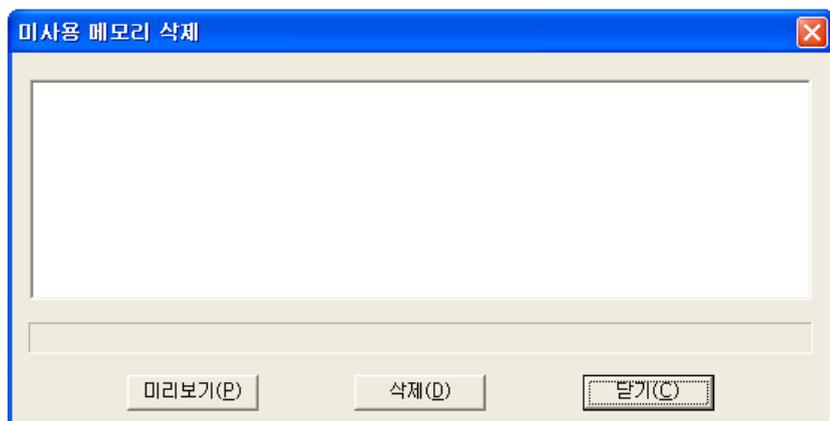
시스템 레퍼런스는 자동저장 시간 및 프로젝트 파일의 기본 폴더를 설정할 수 있습니다.



메뉴	설명
자동저장	자동저장 기능을 사용 여부를 설정하고, 자동 저장에 대한 시간을 설정합니다.
백업	파일이 저장될때 얼마나 많은 백업파일 만드는지 설정합니다.
언어	XGPC의 사용언어를 설정합니다.
새 프로젝트 생성시 기본 폴더 지정	열기화면 또는 저장 화면에서의 기본 폴더를 설정합니다.

### 미사용 메모리 삭제 처리

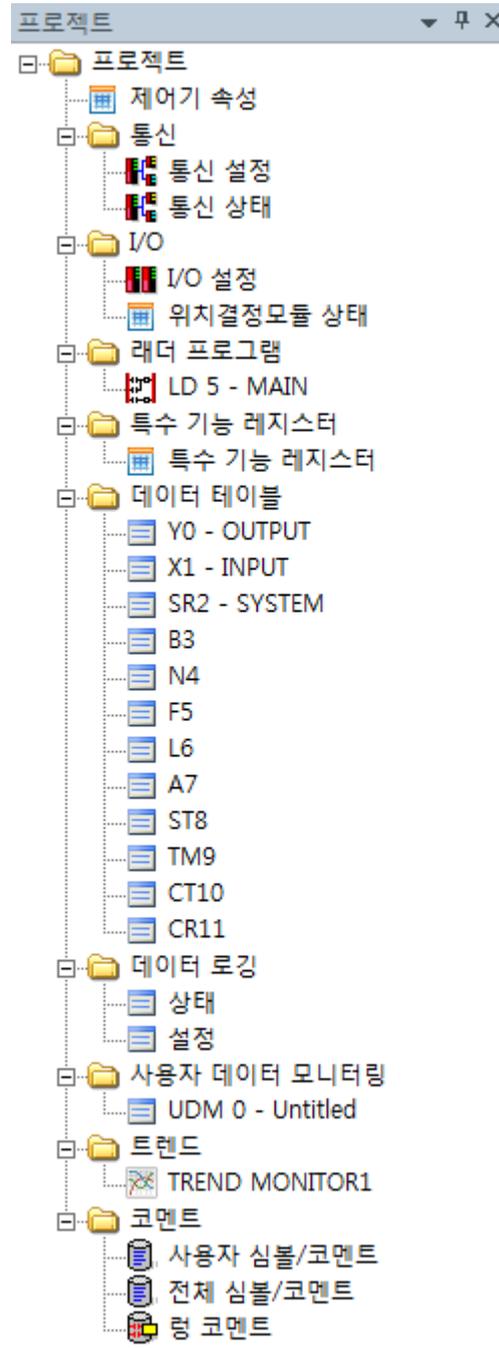
래더에서 사용하지 않는 데이터 테이블 및 메모리를 정리합니다



버튼	설명
미리보기	사용되지 않는 데이터테이블을 계산해 봅니다.
삭제	사용되지 않는 데이터테이블을 삭제합니다.

### 프로젝트 트리

프로젝트 트리는 PLC에서 가지고 있는 정보를 그대로 볼수 있게끔 처리되어 있습니다. 설정을 변경하거나 래더를 변경후 반드시 다운로드하시어 최종 프로젝트를 일치 하셔야 합니다.





### 통신 설정

이 다이얼로그는 PLC 시리얼 포트에 대한 드라이버의 통신 설정을 보여줍니다. 드라이버를 변경하게 된다면 그에 맞게 파라미터를 설정해 주어야 합니다.

### 시리얼 통신설정

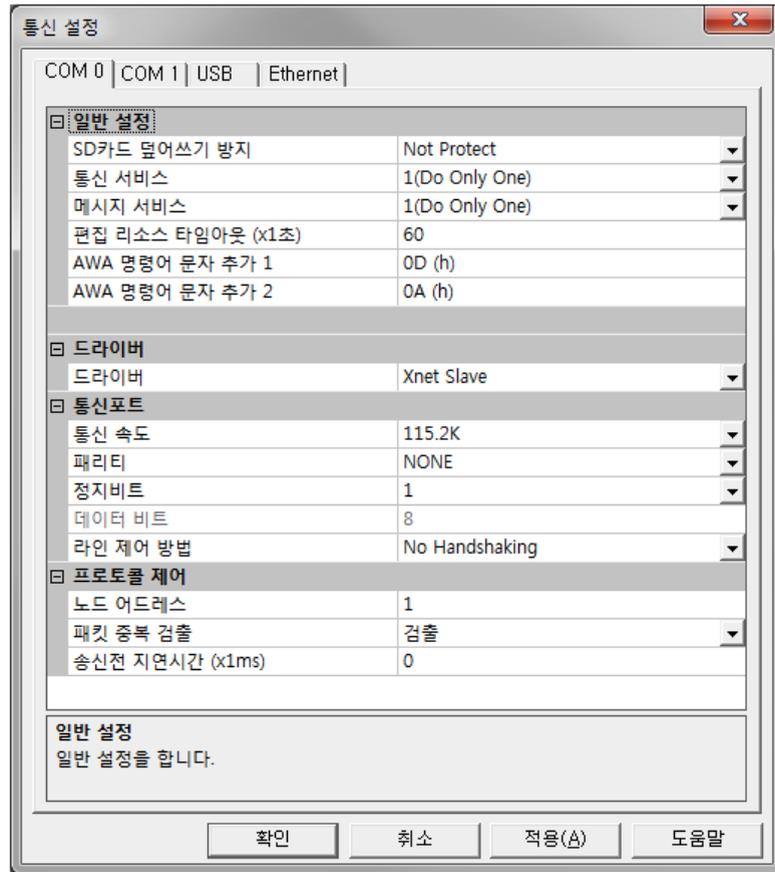
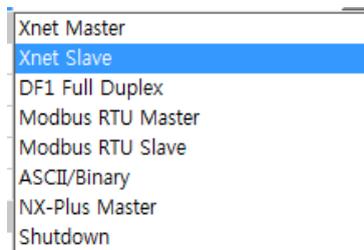


그림 2. 시리얼 포트의 통신 설정

시리얼 통신 드라이버는 Xnet Master와 Xnet Slave, DF1 Full Duplex, Modbus Master, Modbus Slave, ASCII/Binary, NX-Plus Master, NX-Alpha Master가 있습니다. 원하는 드라이버를 선택하고 통신포트 및 프로토콜 제어에 대한 파라미터를 설정하면 됩니다



## 이더넷 통신 설정



그림 3. 이더넷 포트의 통신 설정

이더넷 드라이버는 IP 어드레스 및 포트 속도, Xnet over IP, Modbus TCP, Ethernet/IP, **Socket Message**, Web 서버, SNMP 서버 등을 설정합니다. .

## 통신상태

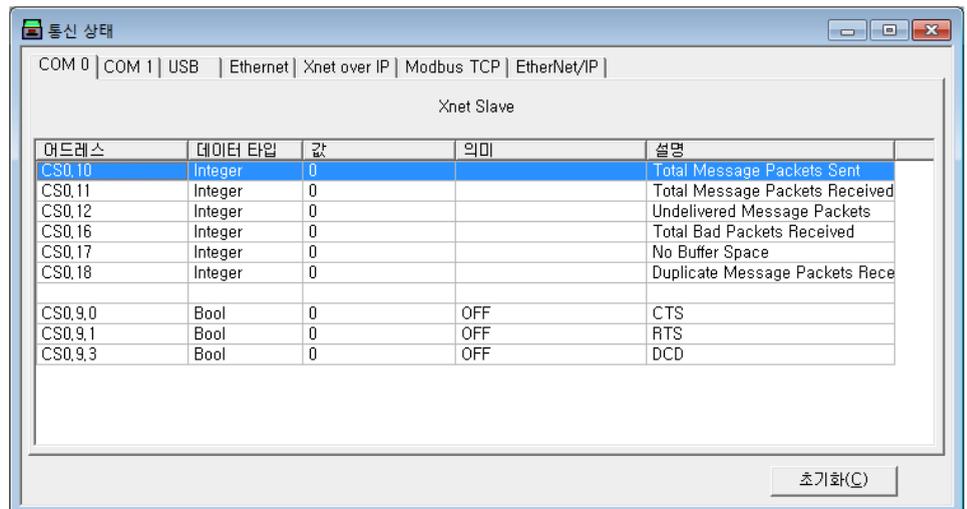


그림 4. 통신상태 다이얼로그

이 다이얼로그는 각 통신 포트에 대한 정보를 보여줍니다. 드라이버가 변경되면, 표시되는 파라미터도 변경이 됩니다.

## I/O 설정

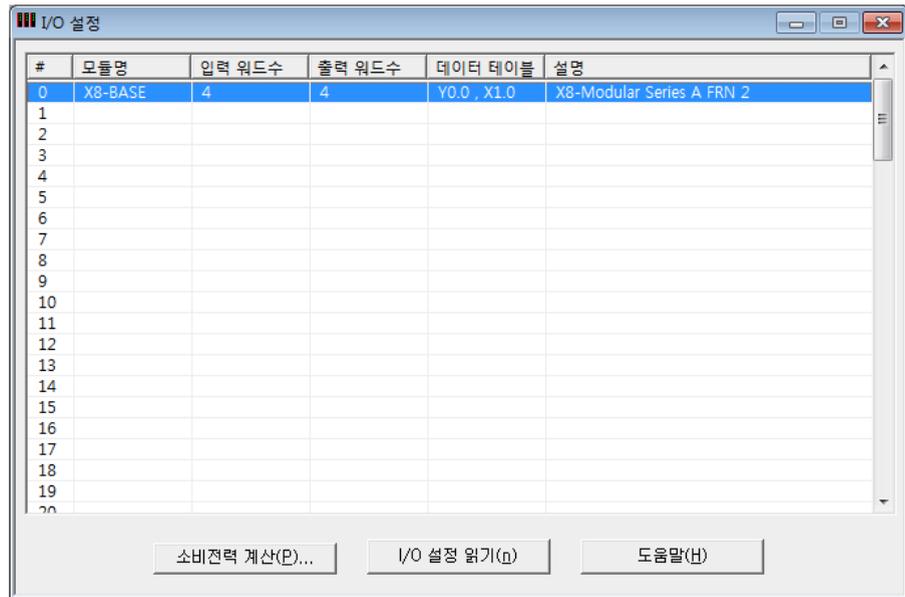


그림 2. I/O 설정 다이얼로그

이 다이얼로그에서는 I/O 모듈에 대해서 수동/자동 설정할 수 있습니다.

- I/O 설정에서 사용하고자 하는 I/O 모듈에 대해서 자동 또는 수동으로 설정할 수 가 있습니다.
- 소비전력 계산 : 선택된 I/O의 파워 소비를 계산합니다.

I/O 설정 읽기 : PLC에서 실제 I/O 맵을 읽습니다.

IO 모듈 추가

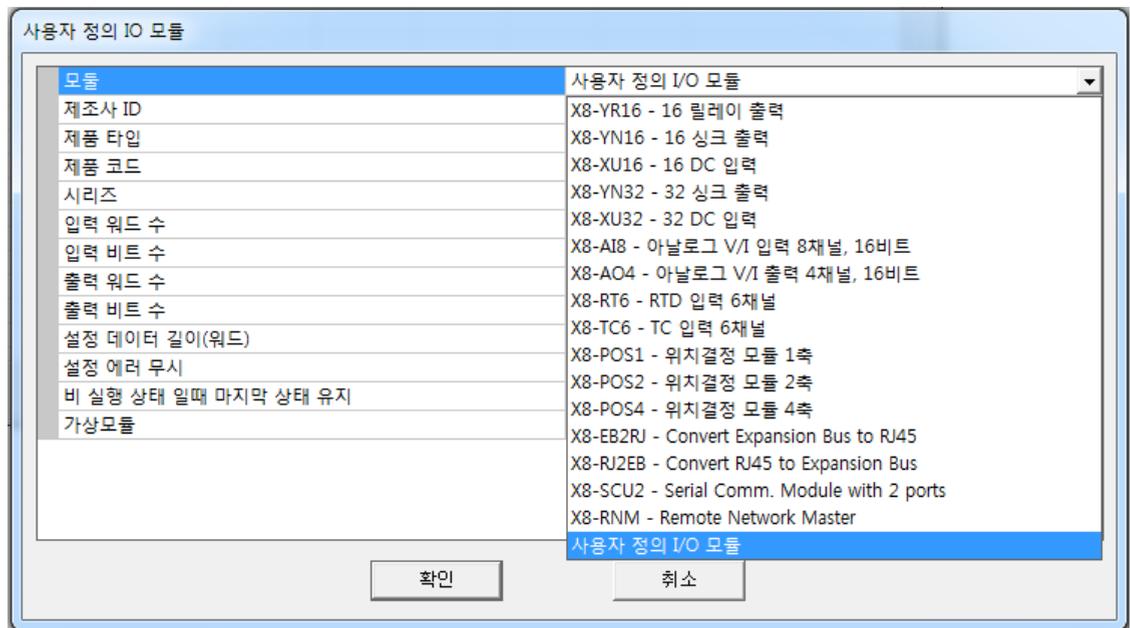
I/O 모듈을 설정하는 방법은 2가지 있습니다.

1. "I/O 설정 읽기" 버튼을 누르면 XGPC는 PLC와 통신을 해서 연결된 I/O리스트를 화면 보여줍니다.



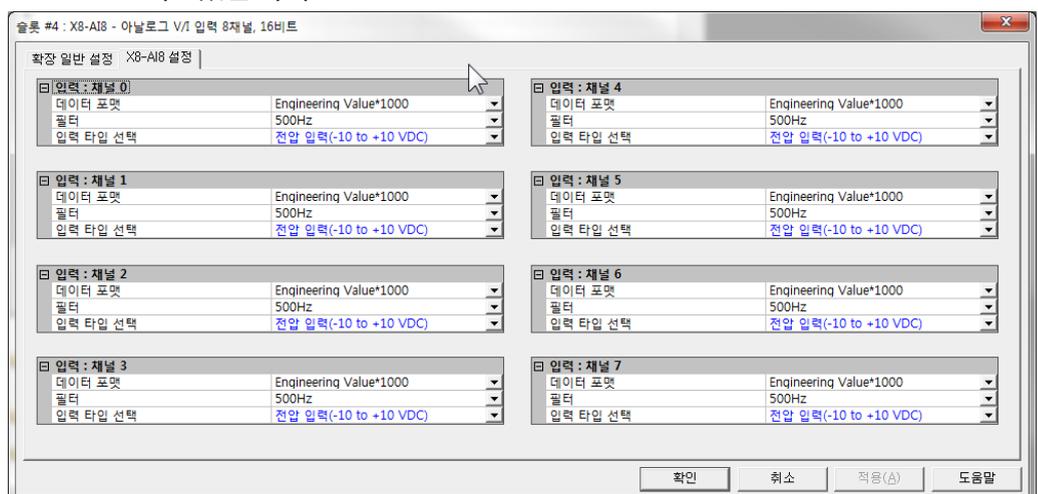
2. "I/O 설정 읽기" 버튼을 누르면 XGPC는 PLC와 통신을 해서 연결된 I/O리스트를 화면 보여줍니다.

슬롯 번호의 열을 더블클릭합니다. '사용자정의 IO 모듈'의 다이얼로그 화면이 나타납니다. 모듈에 있는 콤보박스 리스트에서 원하는 모듈을 선택합니다. 그리고 '확인' 버튼을 눌러 선택을 완료합니다. 리스트에서 선택된 모듈을 볼수 있습니다. 모듈을 더블클릭하면 설정된 모듈에 대해서 상세 설정을 할 수 있습니다.



### 3. I/O 모듈 설정

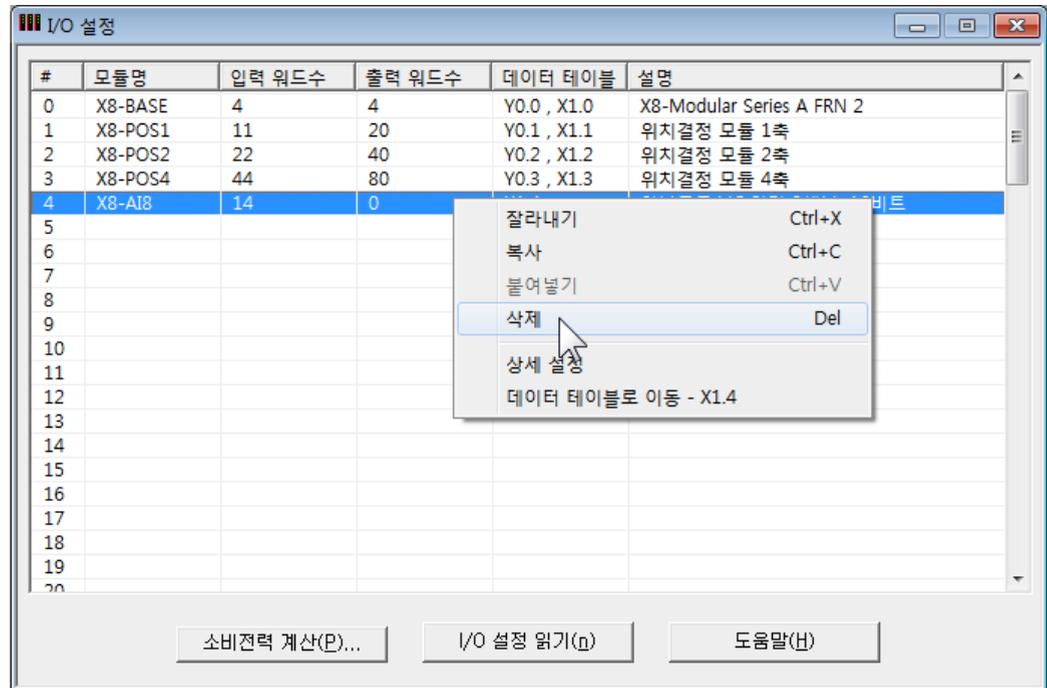
특수 유니트나 CPU에서는 모듈별 설정을 소프트웨어로 설정할 수 있습니다.



4. 사용자 정의 I/O 모듈 제거

삭제하고자 하는 모듈을 선택하고 오른쪽 마우스 버튼을 누릅니다.

팝업 메뉴에서 "삭제 버튼을 누릅니다.

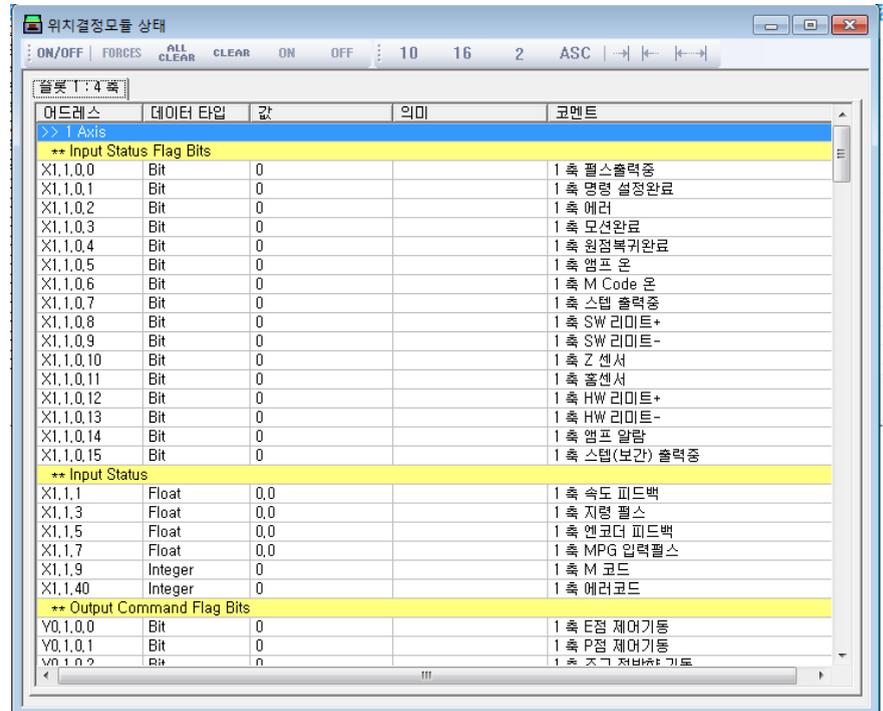


**중요 :**

모든 I/O 설정은 오프라인에서만 변경할 수 있습니다.

### 위치결정/SCU/RNM 모듈상태

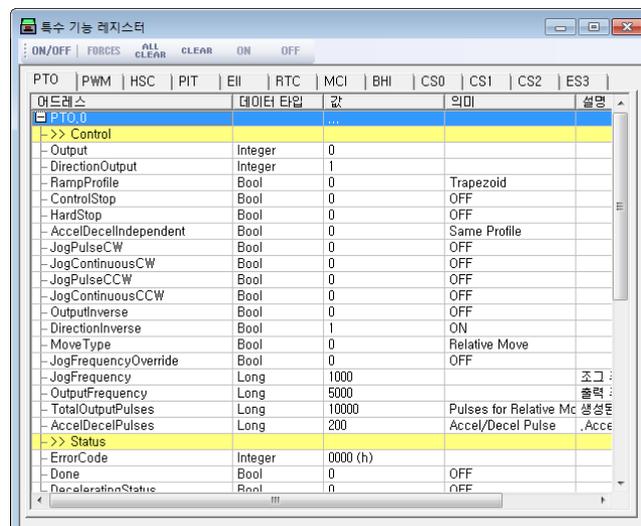
I/O모듈중 위치결정/SCU/RNM 모듈을 장착하고 설정했을 때 보여지는 데이터를 한곳에 모아 처리하기 쉽게 만들어 놓은 모니터링 창입니다.



이 다이얼로그에서는 X8 CPU에서 지원하는 특수모듈 기능들을 바로 설정하고 실행할 수 있습니다.

### 특수기능 레지스터

이 다이얼로그에서는 X8 CPU에서 지원하는 특수 기능들을 바로 설정하고 실행할 수 있습니다.



현재 X8 CPU가 지원하는 특수 기능은 PTO, PWM, HSC, PIT, EII, RTC, MCI, BHI를 지원하고 있습니다.



## 6장. 데이터

X8 PLC의 데이터 처리에 대해서 설명합니다.

---

데이터의 표현(Number System) .....	6-2
기타 데이터 표현.....	6-2
진법 변환과 연산.....	6-3

## 데이터의 표현(Number System)

컴퓨터에서 데이터를 처리하는 경우, 데이터는 기억 장치에 저장되거나 장치들 사이를 이동하여야 하므로 이에 적합한 형태로 표현 되어야 하며, 수치 데이터, 문자 데이터 등 이 형태가 다양하므로 각각의 특성에 맞게 표현되어야 한다.

컴퓨터 내부에서는 모든 데이터의 표현은 두 가지 상태(0 또는 1)로 나타내며 이들의 조합을 부호화하는데 이것을 Code라고 한다.

이 코드는 2진수와 밀접한 관계가 있다.

## 기타 데이터 표현

### 1. 논리 데이터

참(True) : 0이 아닌 값으로 표현(일반적으로 1로 사용)

거짓(False) : 0으로 표현

### 2. 영상 데이터

Graphics : 영상데이터를 영상 표시 장치에 나타내는 방식

Pixel(화소) : 화소의 밝기나 색깔을 나타낸다( 2진수로 표시)

512 \* 512개의 Pixel을 갖고 , 3 bit를 사용하는 영상 표시 장치에 한 화면을 나타내기 위한 영상 데이터는 512 \* 512 \* 3 Bit의 모임으로 구성된다.

### 3. 일반데이터-TEXT 및 데이터

### 4. 음성 데이터

음성 인식 시스템(Speech recognition system) :

음성 자체가 갖는 파형을 Digital신호로 변환하여 2진 코드로 표현한 것이다.

## 진법 변환과 연산

### 각 진법의 수체계

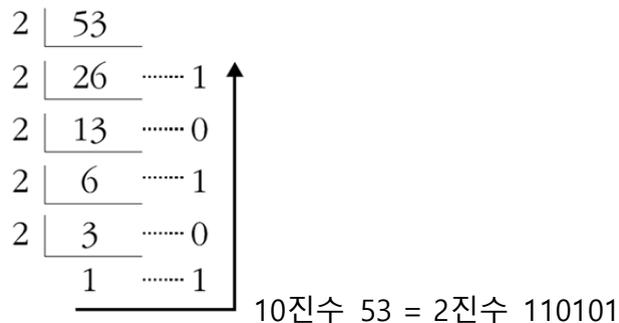
- 10진법 : 0, 1, 2, ..., 9로 표현되는 10개의 디지털로 숫자를 표현
- 2진법 : 0, 1로 표현되는 2개의 디지털로 숫자를 표현
- 8진법 : 0, 1, 2, ..., 7로 표현되는 8개의 디지털로 숫자를 표현
- 16진법 : 0, 1, 2, ..., 9, A, B, C, D, E, F로 표현되는 16개의 디지털로 숫자를 표현

### 진수 (Decimal number)

- 1) 일상생활에서 널리 사용되고 있는 10가지의 수  
 (0, 1, 2, 3, 4, 5, 6, 7, 8, 9)  
 10진수의 자리수는 10개의 수로써 나타내며 자리값(weight)은 10의 거듭제곱으로 나타낸다.  
 ex) 10진수 593.21의 각 자리값  
 우리들이 일상적으로 사용하고 있는 수는 0~9까지인 10진법 숫자 253이라는 숫자는 3자리숫자로 위치에 따라 2는 100의 자리, 5는 10의 자리, 3은 1의 자리에 있다고 약속하여 사용

- 2) 10진수를 2진수로 변환  
 ※ 소수부를 포함하는 10진수를 2진수로 변환하는 경우에는 정수부와 소수부를 분리하여 변환한다.
  - ① 10진수를 2로 나누어 나머지를 구하고 몫이 0이 될 때까지 계속 몫을 2로 나눈다.
  - ② 최초의 나머지를 2진수의 최하위 비트 (LSB: Least Significant Bit)가 되며
  - ③ 최후의 나머지를 2진수의 최상위 비트 (MSB: Most Significant Bit)가 된다.
  - ④ 2진수는 최하위 비트(LSB)에서 최상위비트(MSB)까지 포함하면 된다.

예제) 10진수 53를 2진수로 표시하면



**진수 체계**

2진 숫자(binary digit)를 흔히 비트(bit)라고 말한다.  
 2진수로 정보를 표현하기 위해서 두 가지 디지털 회로를 사용한  
 다면 4가지의 다른 비트 조합 "00", "01", "10", "11"로 표현될 수  
 있다. 2진수 한자리로 표현한 것보다 시스템의 복잡도가 두 배가  
 되었지만 시스템이 표현할 수 있는 정보 상태의 수 또한 두 배  
 가 된다.

3개의 디지털 신호선을 사용했을 때의 비트조합은 8가지가 된다.  
 수학적으로 n개의 2진 신호선 또는 비트에 대해서 가능한 상태  
 의 수를 2n으로 예측할 수 있다. 따라서 디지털 신호선의 숫자를  
 증가시키면 수천, 수만의 다른 2진수를 생성할 수 있다.

이들 2진수 비트들의 일정한 묶음에 이름을 부여하고 있는데, 예  
 를 들면 8비트를 바이트(byte), 4비트는 니블(nibble)이라고 한다.  
 16비트를 워드(word), 32비트를 2중 워드(double word)라 부른다.

**8진수(Octal number)**

1) 8진수

- 디지털 시스템이나 계산기 등에서 2진수를 효과적으로 표현  
 하려고 많이 사용
- 8진수의 자리수 :0, 1, 2, 3, 4, 5, 6, 7을 사용하며, 자리값은 8  
 의 거듭제곱으로 나타냄

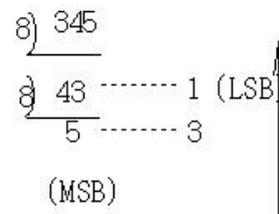
ex) 8진수 237.54의 각 자리값

8진법은 2진수 3자리를 하나로 묶어 0에서 7까지 나타내도록 하  
 고, 16진법은 2진수 4자리를 하나로 묶어 0에서 15까지 표현

2진수 (11010111)를 8진수로 나타낼 경우는 수의 낮은 자리부터  
 3자리씩 마다 8진수의 1자리로 표시하면 다음과 같이 됩니다.

$$\begin{array}{c} \text{11010111} \\ \boxed{\text{110}} \boxed{\text{101}} \boxed{\text{111}} \\ \text{3} \quad \text{2} \quad \text{7} \end{array} \quad (11010111)_2 = (327)_8$$

[예제] 10진수 345(10)를 8진수로 변환하라.



2) 8진수를 2진수로 변환

- 8진수를 2진수로의 변환은 8진수의 각 자리수를 3자리의 2진수로 나타내는 것으로서 쉽게 변환

8진수	0	1	2	3	4	5	6	7
2진수	000	001	010	011	100	101	110	111

8진수를 2진수로 변환시 3비트 사용

3) 2진수를 8진수로 변환

- 2진수를 8진수로의 변환
- 8진수를 2진로 변환하는 역순으로 2진수의 최하위 비트(LSB)를 기준으로 3비트씩 묶어 8진수로 변환

**16진수(Hexadecimal number)**

1) 16진수

16진수 체계도 PLC에서 사용되어진다. 그 이유는 한 워드의 구성이 16비트 또는 8비트로 구성되기 때문이다.

디지털 시스템에서 2진수 4비트를 간소화하여 효율적으로 표기 하는데 사용

16진수의 자리수는 10개의 10진 숫자와 6개의 영문자를 사용하여 16가지의 숫자와 문자로 구성된다.

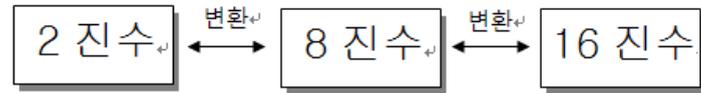
숫자는 0 ~ 9, 문자는 A~ F까지 사용한다.

십진수	16진수
10	A
11	B
12	C
13	D
14	E
15	F

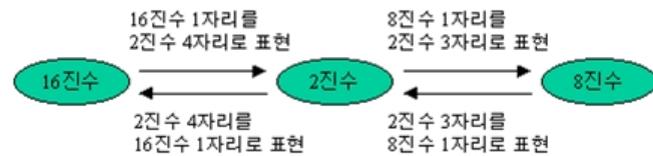


**2진수, 8진수, 10진수의 상호 변환**

8진수와 16진수의 상호변환은 각 진수를 2진수로 변환하여 다시 각 진수로 변환



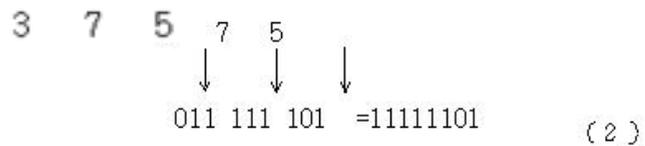
2진수, 8진수, 10진수의 상호 변환



[예제] 8진수 375(8)를 16진수로 변환하라.

$\therefore 375(8) = FD(16)$

풀이) 먼저 8진수를 2진수로 변환



다시 2진수를 16진수로 변환

$$\frac{11111}{F} \frac{101}{D} (2) = FD (16)$$

**BCD(Binary Coded Decimal)**

BCD / 2진 부호화 10진수 (binary coded decimal)란 비트의 2진수를 이용하여 1자리의 10진수를 표현한 코드 방식을 말한다.

예를 들면, 10진수 547은 다음과 같이 표시된다.

0101 0100 0111

BCD 표현 방식의 장점은 수의 크기에 제한이 없다는 것이다.

10진수 1자리를 추가할 경우, 새로이 4-비트만 더 부가하면 된다. 그리고 4비트의 2진수에 2개의 비트를 추가하여 알파벳 및 특수 기호 등을 부호화하였는데 이를 확장 BCD 코드(Extended BCD code)라 한다. (zone bit 2개 digit bit 4개)

- 10진수에 대한 2진 코드는 적어도 4개 이상의 비트를 필요로 한다.
- 8421코드 : 2진화 10진 코드(BCD)의 형식

10진 숫자 0에서 9까지의 표현을 4비트의 2진 숫자로 구성한 것으로 8421은 각각 2<sup>3</sup>, 2<sup>2</sup>, 2<sup>1</sup>, 2<sup>0</sup>의 자리 값을 의미하며 자릿값 코드(weighted code)라 부른다.

- 실제로 8421의 4개 비트로는 0000~1111까지의 모두 16개를 나타낼 수 있으나,
- BCD 코드에서는 0~9까지의 10개만을 사용하고 나머지 6개(1010, 1011, 1100, 1101, 1110, 1111)는 사용하지 않는다.

Advantage : 10진수와의 변환이 다른 코드에 비해 쉽게 처리될 수 있다.

Disadvantage : BCD 코드로 표시된 수 전체에 대하여 2진 가산법이 적용되는 것이 아니고 4비트에 대해서만 적용

먼저 BCD 코드의 덧셈에서는 다음 2가지 조건을 적용해야 한다.  
 첫째, 두 BCD수를 덧셈 결과 각 자릿수의 4비트가 10개의 BCD수(0000~1001)에 포함된 경우  
 → 그 자체가 BCD값이 된다.

둘째, BCD수를 더한 결과 각 자릿수의 4비트가 10개의 BCD수를 초과할 경우  
 → 캐리를 상위 BCD수에 더하기 시킨다.

BCD(8421)코드	10진수
8421	
0000	0
0001	1
0010	2
0011	3
0100	4
0101	5
0110	6
0111	7
1000	8
1001	9
00010000	10

## 7장. PLC 프로그래밍

본 장에서는 X8 PLC 운영을 위한 PLC 프로그래밍 대해서 설명합니다.

---

프로그램 언어 .....	7-2
PLC Programming 용어의 정의 .....	7-3
PLC의 메모리 번지 구성 .....	7-6
주소표현방법 .....	7-9
PLC PROGRAMMING .....	7-12
운전 모드(mode)의 개요 .....	7-12
CPU 처리순서 .....	7-13
Programming 명령어 종류 .....	7-15

## 프로그램 언어

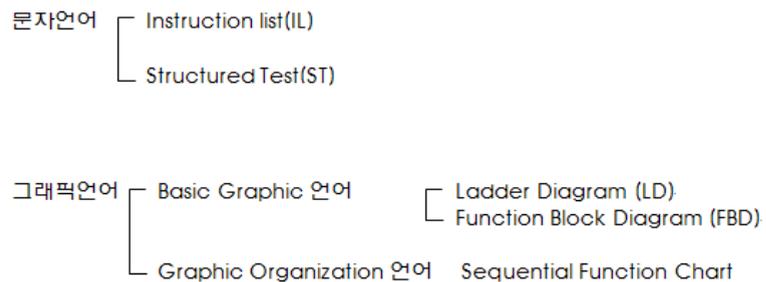
### 프로그램 언어

PLC와 일반 컴퓨터가 구별되는 주요 항목 중에 하나가 프로그래밍 언어의 차이이다. 즉, PLC는 논리 표현 및 변경의 용이성을 중시하는 경향이 있는데 여기에는 사용자와 PLC를 중계하는 언어가 큰 비중을 차지한다.

따라서, 특별한 용도를 위한 각 기종의 특징을 발휘하기 위해서는 다른 언어 체계를 갖는다.

IEC에서 최근 표준을 발표하였다. 그 중에서 전통적으로 많이 사용해온 것이 니모닉방식과 래더 다이어그램 방식이다.

또한 고급 언어 방식은 일반 컴퓨터에서 사용되는 Basic이나 Fortran과 같은 형태이며 플로우 차트방식인 SFC (Sequential Function Chart)가 있다.



[그림5-1] PLC Programming Language의 분류

### 니모닉 프로그램

시퀀스의 접속 상태를 특정 명령어를 사용하여 프로그램 하는 방법을 말한다.

시퀀스회로를 컴퓨터 저급 언어 방식을 도입한 프로그램 방식으로 명령어 방식이라고도 한다. 현장의 경우 로더를 이용하여 프로그램을 하는 경우가 많은데 (과거에 많이 사용하던 방식) 이러한 경우에 주로 사용한다.

어드레스	명령어	오퍼랜드
00000	NOC	B3.0.0
00001	OUT	B3.0.1
00002	NOC	B3.0.2
00003	OUT	B3.0.4
00004	OUT	B3.0.5

[표 5-1] 니모닉 방식의 프로그램 예

## 래더 다이어그램

PLC 프로그램에서 가장 많이 사용되는 방식으로서 릴레이 회로를 직접 기입하거나 표시할 수 있도록 도면을 표시하는 화면 출력장치가 필요한데 이는 그래픽 로더나 컴퓨터를 많이 사용하고 있다.

래더 다이어그램 방식은 래더 심볼로 대체되는데, 이때 래더 심볼이나 접속선 등을 표시하는 키를 차례로 눌러 디스플레이상에 회로를 그리면 완료된다.

이 방식은 회로를 눈으로 확인하면서 프로그램 할 수 있으므로 단순하면서도 직관적으로 이해할 수 있다.

## PLC Programming 용어의 정의

### 1) 번지 (= 레지스터)

외부 입, 출력모듈 및 각 내부 메모리의 위치를 지정한 메모리의 주소입니다. 본 교재에서 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 사용 할 때는 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.

### 5) 더블워드(Double Word)

32비트로 구성되고 범위는 -2,147,483,648~2,147,483,647 (16진수:0~FFFFFFFF)이며 하나의 독립된 번로 사용될 수 있습니다.

6) 엣지(Edge)

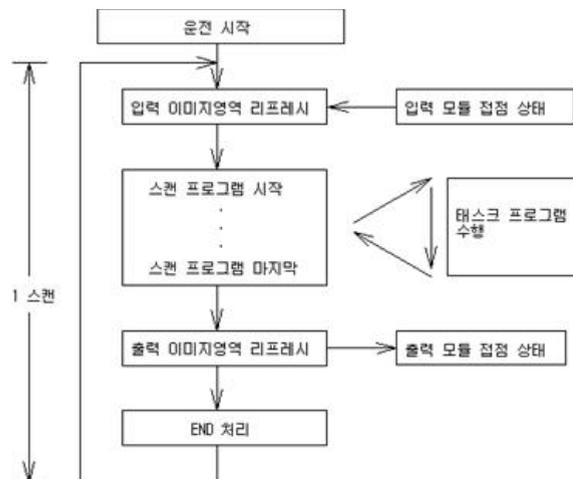
펄스의 상승/하강시점을 말하며, 1스캔동안 ON되는 명령입니다.

- OFF →ON 유지입력접점의 경우 최초 ON될 때 한 번의 상승 엣지가 발생합니다.
- ON →OFF 유지입력접점의 경우 최초 OFF될 때 한 번의 하강 엣지가 발생합니다.

7) 스캔타임(Scan Time)

CPU는 RUN중 프로그램 처리뿐만 아니라, 주기적으로 통신처리, 입출력 처리, 강제 ON/OFF 값 처리등을 수행하는데 이런 주기적인 1사이클을 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.



[그림 5-2] 스캔타임(Scan Time)

입력 Refresh후 프로그램번 스텝부터 END 까지 수행하고, 자기진단 후 출력Refresh를 수행하게 됩니다. 이후 다시 입력 Refresh부터 같은 동작을 반복 수행하게 됩니다.

- 입력 Refresh : 프로그램을 수행하기 전에 입력 Unit 에서 입력 Data 를 Read 하여 Data의 입력용 영역에 일괄 저장합니다.

- 출력 Refresh : 프로그램 수행 완료 후 Data Memory 의 입력용 영역의 Data 를 출력Unit 에 일괄 출력합니다.
- 즉시 입출력 명령을 사용한 경우 : 명령에서 설정된 입출력 카드에 대하여 프로그램 실행 중 입출력을 Refresh합니다.
- 출력의 OUT 명령을 실행한 경우 : Program의 연산 결과를 Data의 출력용 영역에 저장하고 END 명령수행 후 출력 Refresh에 해당 접점을 ON 또는 OFF 시킵니다.

#### 8) BCD(Binary Coded Decimal)

십진수의 1자리 (Digit)를 4비트로 표현하는 숫자체계로서 0-9 숫자로 표시됩니다.

BCD 값 변환은 16진수 값 계산처럼 하면 됩니다.

예) 59(BCD)=59(HEX), 32(BCD)=(32HEX)

#### 9) FLASH ROM

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

#### 10) 점(Point)

입력 8 점, 출력 6 점의 PLC 는 스위치나 센서 등 입력기기를 최대 8 개, 램프나 릴레이 등 출력기기를 6 개를 연결할 수 있습니다.

PLC 의 입출력 용량을 표시할 때 사용합니다.

#### 11) 스텝(Step)

PLC 명령어의 최소 단위로 A 접점, B 접점, 출력 코일 등의 명령이 1 스텝에 해당하는 명령이고 기타 응용 명령어의 경우 하나의 명령어가 다수의 스텝을 점유합니다.

프로그램 용량 및 CPU 속도를 표시하는 단위로 사용 됩니다.

(예, 용량: 30k step, 속도:)

#### 12) WDT( Watch Dog Timer)

프로그램 연산 폭주나 CPU 기능고장에 의하여 출력을 하지 못할 경우 설정한 시간(WDT)대기 후 에러를 발생시키는 시스템 감시 타이머입니다.

기본 3000ms 로 설정되어 있으며 파라미터 지정에 의해 변경 시킬 수 있습니다.

### 13) 파라미터(Parameter)

프로그램과 함께 PLC 에 저장되는 운전 데이터로 통신, 시스템 환경 등을 지정합니다.

## PLC의 메모리 번지 구성

PLC의 메모리는 사용용도에 따라서 분류되며, 이 분류를 기준으로 식별자로 구분하게 된다. 그러므로 모든 외부 입출력장치가 내부 데이터를 처리를 위한 메모리에는 번지(ADDRESS)와 데이터가 항상 동시에 존재한다.

메모리 번지 지정은 그 종류에 따라서 B,N,F,L,A,ST등으로 구분하는데 종류에 대한 식별자는 PLC제조회사에 따라서 또 모델명에 따라서 조금씩 다르므로 프로그램 작성 전에 미리 확인하여야 한다.

### 외부입출력 영역(입력 X, 출력 Y)

외부 입출력영역은 직접 외부 기기와 연결되는 주소를 나타낸다. 비트/워드/더블워드로 사용 가능하다. 입출력 X/Y는 외부 기기와 대응되는 영역으로서 입력기기로 사용되는 누름버튼 스위치,(PBS : Push Button Switch), 절환 스위치, 리미트 스위치 등의 신호를 받아들이는 입력장치와 출력기기로 사용되는 솔레노이드 밸브, 모터, 램프 등에 연산 결과를 전달하는 출력장치가 해당된다. 입력장치 X에 대해서는 PLC 내부의 메모리에 입력 상태가 보존되므로 a, b접점 모두 사용 가능하고 출력장치 Y에 대해서는 a접점만이 가능하다.

### 내부접점 (B: Memory)

PLC 내부의 내부 접점 영역으로서 보조 접점으로 사용한다. 비트/워드/더블워드로 사용이 가능하다. 전원이 인가될 때나 프로그램이 수행이 시작될 때 내부값은 설정에 따라 전부 클리어 된다. 내부 접점은 외부로 직접출력이 불가능하지만 입출력 X/Y과 연결한다면 외부 출력이 가능하다. 내부 접점은 a,b접점으로 사용 가능하다.

**정전유지접점** (모든 메모리의 설정에 따라 가능하다)

정전유지접점은 내부 접점과 유사하지만 전원이 인가될 때나 프로그램의 실행 시작 시에도 그전의 데이터를 유지하는 영역이다. 비트/워드/더블워드로 사용이 가능하다. 내부 접점과 마찬가지로 a, b접점으로 사용 가능하다.

**타이머 (TM)**

메모리의 용량만큼 설정으로 제공하고 있으며, 1/10/100/1000msec의 기본주기를 기준으로 하는 타이머로 사용할 수 있는 영역으로 TON, TOF, TONR등 3가지 명령어로 구성되어 있다. 최대 설정 값은 32BIT로 다양한 설정 값의 타이머를 만들 수 있다.

**카운터 (CT)**

카운터는 한 번 사용한 채널은 중복하여 사용할 수 없다. 그리고 CTU, CTD등 2가지 명령어로 사용할 수 있다. 최대 설정 값은 32BIT이다.

**데이터 워드 (N/L/F)**

데이터 워드 영역은 정전시에도 데이터가 보존되며 설정으로 초기화가 가능하다. 16BIT, 32BIT구분되며 실수형 데이터를 따로 처리할 수 있다.

**특수 접점 (SR)**

특수 접점은 시스템이 제공하는 특수 기능을 활용할 때 사용되며, PLC의 RUN/STOP등 시스템의 상태 및 동작을 제어할 때 사용한다.

**특수 기능 데이터 테이블(SFR)**

특수 한 기능들을 이미 구현하여 필요한 데이터를 입력하면 사용 가능하도록 만들어 놓은 메모리로 동작을 편하게 구현 할 때 사용합니다.

## 특수 기능 레지스터 (SFR, Special Function Register)

SFR 이름	SFR 약자	설명
High-Speed Counter	<b>HSC</b>	HSC 는 고속 카운터 입력에 관련된 데이터를 모아 놓은 데이터 테이블이다.
Pulse Train Output	<b>PTO</b>	PTO 는 펄스 출력에 관련된 데이터를 모아 놓은 데이터 테이블이다.
Pulse Width Modulation	<b>PWM</b>	PWM은 펄스 폭 변조 명령과 연관된다. 폭 변조 (PWM ) 기능 펄스 참조.
Selectable Timed Interrupt	<b>STI</b>	STI는 선택 가능한 시간 제한 인터럽트 함수 와 연관된다. 선택 가능한 시간 제한 인터럽트 ( STI ) 기능 사용을 참조하십시오.
Event Input Interrupt	<b>EII</b>	EII는 이벤트 입력 인터럽트 명령과 연관된다. 이벤트 입력 인터럽트 ( EII ) 기능 파일 사용을 참조하십시오.
Real-Time Clock	<b>RTC</b>	RTC는 실시간 클록 기능 ( 시간대 ) 와 연관된다. 실시간 시계 기능 파일을 참조하십시오.
Memory Module Information	<b>MMI</b>	MMI는 메모리 모듈 에 대한 정보가 포함되어 있습니다. 메모리 모듈 정보 기능 을 참조하십시오.
Base Hardware Information	<b>BHI</b>	BHI는 컨트롤러의 하드웨어에 대한 정보가 포함되어 있습니다. 기본 하드웨어 정보 기능 파일을 참조하십시오.
Communications Status File for Channel 0	<b>CS0</b>	CS0는 컨트롤러의 하드웨어에 대한 정보가 포함되어 있습니다. 기본 하드웨어 정보 기능 파일을 참조하십시오.
Communications Status File for Channel 2	<b>CS2</b>	
I/O Status File	<b>IOS</b>	IOS는 컨트롤러 I / O 에 대한 정보가 포함되어 있습니다. 입력 / 출력 상태 파일을 참조하십시오.
Ethernet Status File for Channel 1	<b>ES1</b>	ES1은 컨트롤러와 이더넷 통신 에 대한 정보가 포함되어 있습니다.

## 주소표현방법

메모리를 사용하기 위해서는 메모리번지를 표현하는 방법이 있어야 한다. PLC의 경우 주소 표현방법은 사용자 프로그램을 작성할 때 식별자를 사용할 경우, 각 영역 전체를 워드 또는 비트단위로 각각의 주소를 지정하여 사용한다. 주소는 식별자와 워드 또는 비트 번지를 조합하여 사용한다. 비트번지와 워드번지를 동시에 사용할 수 있는 B,N등은 명령어의 종류에 따라 비트번지와 워드번지가 자동으로 구분되므로 사용할 때 주의하여야 한다.

비트 번지는 ON(1)또는 OFF(0)등의 상태를 나타내며, 워드 번지는 16bit로 구성되어 데이터 값을 -32767~32768까지 처리할 수 있습니다.

### 비트 번지 지정

비트 번지는 종류를 구별하는 문자인 B,N 등과 워드를 표시하는 3자리의 10진수 숫자(000~127까지)와 비트를 표시하는 2자리의 10진수(00~15까지) 숫자로 구성합니다.

### 워드번지 지정

워드번지는 문자 모든 오프랜드등과 워드를 표시하는 10진수 숫자로 구성되며, 특수영역인 SR2.1.0~SR2.109 사용하기도 합니다.

### 더블 워드 번지 지정(L,F)

더블워드 지정은 지정된 메모리 오프랜드로 더블워드 사용이 가능합니다.

## 표현방법

어드레싱 기술 방법

# Df.S.W.B

D : 데이터 타입

데이터 종류를 결정하고 사용하다.

예)

X : 외부 입력

Y : 외부 출력

SR : 스페셜 레지스터

B : 바이너리 데이터

N : 16비트 10진수 데이터

F : 실수 데이터

L : 32 비트 10진수 데이터

..

f : 파일 넘버

연속적으로 발생하는 넘버.

데이터 타입은 중복되나 넘버 중복되지 않는다.

S : 슬롯 넘버

외부 입출력 유니트에 할당되며 CPU부터 0이 시작된다.

W : 워드 넘버

모든 데이터 타입의 워드 번호이다.

(16비트로 구분된다.)

B : 비트 넘버

워드 번호의 할당된 비트 넘버

∴ 모든 구분은 .으로 한다.

I/O 어드레싱 구조

Format	Explanation		
Df.S.W.B	<b>D</b>	데이터 종류	Input(X), Output(Y), Status(SR), Binary(B), N(Integer), TM(Timer) CT(Counter), CR(Control), A(ASCII), F(Float), ST(STRING), L(Long) MG(Message), PD(PID), PS(Pulse Limited Switch), RP(Routing Path)
	<b>f</b>	데이터 테이블 번호	0 = output, 1 = input → 고정 3번부터 사용자 임의로 사용 가능
	.	슬롯 구분자	
	<b>S</b>	슬롯 번호	내장 I/O 슬롯 번호 : 0 확장 I/O 슬롯 번호 : 1~40
	.	워드 구분자	
	<b>W</b>	워드 번호	워드 데이터의 Read 또는 Write 시에 필요 만약 Bit 번호가 15 이상일 경우 필요
	.	비트 구분자	
	<b>B</b>	비트 번호	0 to 15

어드레싱 예제

어드레싱 방법	어드레싱 예	슬롯	워드	비트
비트 어드레싱	<b>X1.0.0</b>	<b>0</b>	<b>Word 0</b>	<b>Input bit 0</b>
	<b>Y0.0.1</b>	<b>0</b>	<b>Word 0</b>	<b>Output bit 1</b>
	<b>B3.3.15</b>	<b>X</b>	<b>Word 3</b>	<b>Binary bit 15</b>
	<b>N4.20.4</b>	<b>X</b>	<b>Word 20</b>	<b>Integer bit 4</b>

"X" 의 의미는 사용하지 않음.

내부 메모리는 슬롯번호가 삭제됩니다.

어드레싱 방법	어드레싱 예	슬롯	비트
워드 어드레싱	<b>X1.1.1</b>	<b>1</b>	<b>Word 1</b>
	<b>Y0.3.3</b>	<b>3</b>	<b>Word 3</b>
	<b>ST8.7</b>	<b>X</b>	<b>Word 7</b>
	<b>B3.13</b>	<b>X</b>	<b>Word 13</b>
	<b>N4.20</b>	<b>X</b>	<b>Word 20</b>

## PLC PROGRAMMING

### 1. 문제분석



### 2. 입출력기기의 할당

각종 입출력기기의 고유번호 등을 부여한 동작 상태를 정의 내 부릴레이, 타이머, 카운터 등의 할당



### 3. 코딩

시퀀스 프로그램을 PLC의 메모리에 기억시키기 위해 PLC용 래더도에 따라 순서대로 니모닉 명령어로 기입해 넣는 작업



### 4. 로딩

프로그램 입력장치를 이용하여 PLC메모리에 기억시키는 작업



### 5. 디버그(시뮬레이션)

프로그램의 틀린 곳을 찾아 수정 한다.



### 6. 운전

## 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 유닛은 외부접점의 신호를 읽은후 RAM에(또는 NVSRAM) 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 NVSRAM 메모리에 프로그램을 저장할 수 있는 유일한 모드입니다.)

## PAUSE 모드(일시정지 상태)

PAUSE 모드에서는 사용자 프로그램을 1스캔동안 운전할 수 있고, 그 결과 값을 유지할 수 있는 모드입니다. 1스캔씩 프로그램 수행이 가능하므로 프로그램 디버깅시 편리합니다. STOP 모드와 거의 동일하나 Data를 초기화시키지 않습니다.

## ERROR 모드(에러상태)

ERROR 모드는 유닛이 자체진단 하여 내부 오류를 발견 할 경우, 발생하는 모드로서, 정상적인 기능 수행이 불가능합니다. 에러가 발생되면 CPU는 프로그램 실행을 중단하고 모든 출력을 OFF시킵니다. 에러모드가 발생하면, 사용자는 에러코드를 검색하여 적절한 해제조치를 취한 후 전원을 OFF→ON시키거나, XGPC의 CPU 오류클리어 버튼을 눌러서 Error를 해제할 수 있습니다.

## CPU 처리순서

### 프로그램 처리순서

#### 1. 프로그램 해석처리 과정

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

#### 2. 주변장치 신호처리 과정

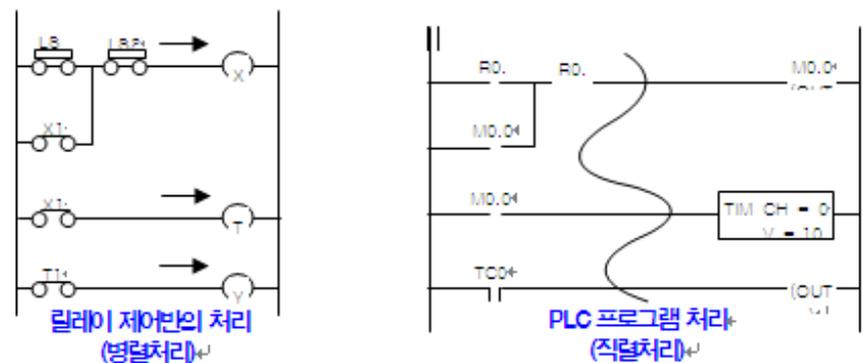
통신모듈 또는 주변장치로부터 데이터를 내부 메모리에 저장합니다.

3. 강제입출력처리 과정

강제 입출력 설정 비트가 존재할 경우 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.



[그림 5-3] 입출력처리 과정

5. 위치독 시간 초기화과정

위치독 경과치를 0으로 합니다. (다음 스캔까지의 위치독 계산 시점이 됨) 아래는 릴레이제어반과 PLC의 시퀀스 처리과정의 다른점을 보여줍니다. 즉 릴레이제어반은 모든 시퀀스가 동시에 병렬처리 되는 반면, PLC는 프로그램의 처음스텝부터 최종스텝까지 주기적으로 직렬 처리됨을 보여줍니다.

## Programming 명령어 종류

아래의 테이블은 X8 PLC의 각 기능별로 나눈 명령어를 보여 주고 있습니다.

### 그룹별 X8명령어

명령어 그룹	설명
고속 카운터 입력	HSCS, HSCC – 고속 카운터 명령을 사용하여 DC 입력 포트를 통해 들어오는 데이터의 모니터링 및 제어를 할 수 있습니다.
PTO/PWM 출력	PTO, PWM PTO 및 PWM 출력 명령을 사용하여 DC 입력포트를 통해 들어오는 데이터의 모니터링 및 제어를 할 수 있습니다.
릴레이, 비트처리	NO, NC, OUT, SET, RST, ONSR, EGR, ONSF 릴레이 및 접점 처리 명령을 통해 입력 접점의 모니터링 및 상태 비트를 제어 할 수 있습니다.
타이머 및 카운터	TON, TOF, TONR, CTU, CTD, RSTA 타이머와 카운터 명령을 사용하여 시간 및 이벤트 관련 사항의 제어를 할 수 있습니다.
비교 명령	EQ, NE, LT, LE, GT, GE, EQM, LIM 비교 명령을 통해 각종 데이터와의 비교 기능을 제공합니다.
수학 연산 명령	ADD, SUB, MUL, DIV, NEG, CLR, ABS, SQRT, SCAL, SCAP, SWAP, CALC, COS, ATAN, ASIN, ACOS, SIN, TAN, POW, LN, LOG, DEG, RAD 다양한 종류의 수학 명령어를 지원합니다.
Application Specific	CLKR, CLKC 성능 진단을 위한 명령어들로 클럭 데이터를 읽고 비교 할 수 있습니다.
변환 명령	DECO, ENCO, BCD, BIN, GRAY 변환 명령을 사용하여, 멀티플렉스, 디멀티플렉스, 그레이 코드, BCD 변환 등의 작업을 수행 할 수 있습니다.
논리 연산 명령	AND, OR, XOR, NOT 논리 연산 명령을 사용하여 논리 비트 연산을 할 수 있습니다.
데이터 이동 명령	MOV, MOVW 데이터 이동 명령을 사용하여 데이터 이동 및 수정을

	할 수 있습니다.
데이터 테이블	COPW, COPT, FILT, BSL, BSR, FIFOL, FIFOU, LIFOL, LIFOU, SWAP 데이터 테이블에 관련된 기능을 수행합니다.
순차 제어	SEQIC, SEQO, SEQL 반복적인 설비 제어나 프로세스 처리를 위한 순차제어 처리 시 순차 제어 명령어를 사용하여 처리할 수 있습니다.
프로그램 제어 명령	JUMP, LBL, CALL, SBR, RET, SUSP, ENDT, MCR, END 프로그램 제어 명령을 사용하여 래더 프로그램 수행 제어를 할 수 있습니다.
입출력 명령	REFI, REFO, EOS 입출력 명령을 통해서 입력, 출력 스캔과 관계없이, 즉각적인 입출력 값의 처리를 할 수 있습니다.
인터럽트 명령	PITS, INT, INTD, INTE, INTF 사용자 정의 인터럽트 제어 명령을 통해 정의된 이벤트의 인터럽트 제어가 가능합니다.
PID 명령	PID PID 명령을 사용하여 Closed-Loop 제어를 할 수 있습니다.
아스키 명령	ARNL, ARNC, ATOI, ACB, ACN, AEX, AHS, ITOA, ARC, ARL, ASCH, ACMP, AWA, AW 아스키 명령을 사용하여 형식 변환 및 아스키 문자 처리를 할 수 있습니다.
통신 명령	MSG, SVC 통신 명령을 사용하여, 다른 기기로 데이터를 읽거나 쓸 수 있습니다.
레시피 명령	RCP 레시피 명령을 사용하여 레시피 데이터베이스와 사용자 데이터 테이블 구성원간의 데이터 전송을 할 수 있습니다.
데이터 로깅 명령	DLOG 데이터 로깅 명령을 사용하여, 시간 및 데이터를 로깅할 수 있습니다.

### 명령어에 대한 설명

모든 명령어에 대한 설명에 있어서, 아래와 같은 서로 유사한 테이블을 사용하여, 각 명령어의 가능 입력 조건, 가능 출력 조건들에 대한 내용들을 설명합니다.

예) 적용 가능한 어드레싱 모드 및 데이터 테이블

파라미터	데이터 테이블																펄스션										어드레스 모드			어드레스레벨			
	Y	X	SR	B	TM, CT, CR	N	F	ST	A	L	MG, PD	RP	PLS	ASCII	RTC	HSC	PTO, PWM	STI	EII	BHI	MMI	CS - Comms	IOS - I/O	DLS - Data Log	Immediate	Direct	Indirect	Bit	Word	Long Word	Element		
입력 A	○	○	○	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
입력 B	○	○	○	○	○	○	○	○	○	○		○		○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○	○			
출력	○	○	○	○	○	○	○	○	○	○		○		○	○	○	○	○								○	○		○	○			

Indirect Addressing 모드로 SR, MG, PD, RTC, HSC, PTO, PWM, STI, EII, BHI, MMI, CS, IOS, DLS 는 사용 할 수 없습니다..

### 어드레싱 모드

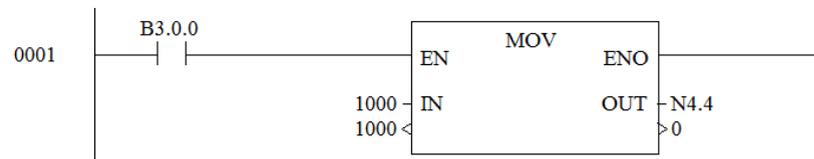
X8 PLC 는 3가지 종류의 어드레싱 모드를 지원합니다.

- Immediate
- Direct
- Indirect

X8 PLC는 Indexed Addressing은 지원하지 않는 대신, Indirect Addressing모드를 통해서 그 기능을 대신 할 수 있습니다. 본 3가지의 어드레싱 모드를 통해서 다양한 응용을 할 수 있습니다.

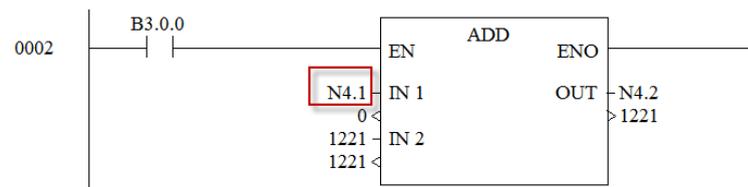
**Immediate Addressing**

가장 기본이 되는 어드레싱 모드로, X8 PLC 데이터 테이블에 직접 상수를 입력하는 방식을 의미합니다.



**Direct Addressing**

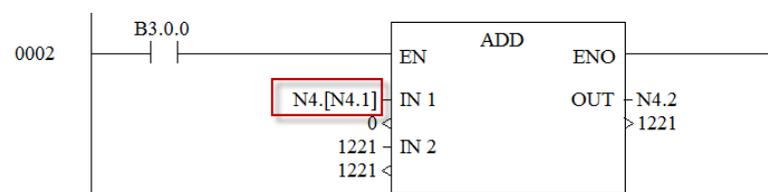
Direct Addressing은 아래의 예와 같이 X8 PLC내부의 데이터 테이블 구성 요소(아래의 예는 N4.1)를 직접 입력하는 방식입니다.



**Indirect Addressing**

Indirect Addressing 은 포인터의 개념을 통한 래더 프로그램을 가능케 합니다. 본 기능은 레시피, 배치 프로세싱 등의 응용에 사용 될 수 있습니다.

Indirect Addressing은 대괄호 “[ ]”를 사용하여, 어드레스를 표현합니다.



위의 예는 N4.[N4.1] 로 표현되어 있는 데이터 테이블의 값과 1221이라는 값을 더해서 N4.2에 저장을 하는 래더 프로그램입니다.

여기서 N4.1 의 값이 10일 경우는, N4.10 이라는 데이터 테이블로부터 데이터를 가져와 1221 이라는 값과 더해서 N4.2 에 저장을 하게 됩니다.

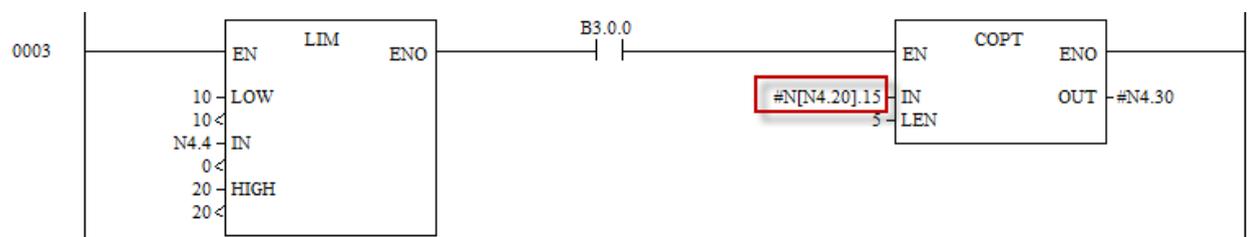
여기서 N4.1 의 범위는 X8 PLC에서 지원하는 범위인 0 ~ 1534 까지 , 총 1535입니다.

**TIP**

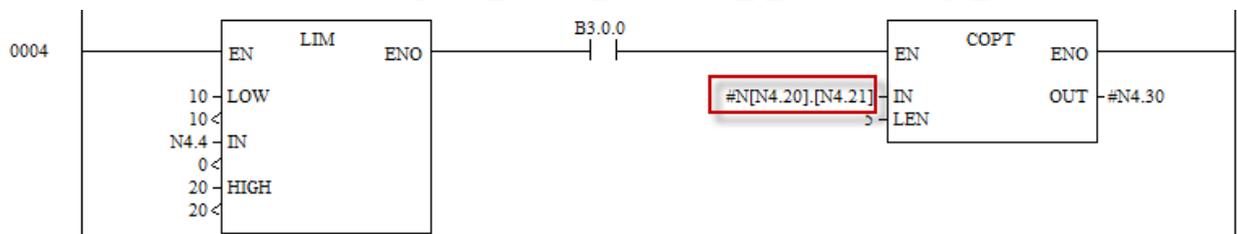
만약, 데이터 테이블에서 정의해 놓은 구성 요소 개수 이상을 Indirect Addressing 방식을 통해 액세스하게 될 경우, 28H 에러 (런타임에러, 래더 프로그램 에러)가 발생하게 됩니다. 물론 처음 컴파일 시에는 문법적으로 잘못된 것이 없으므로 정상 처리가 됩니다.

### Indirect Addressing 을 통한 데이터 테이블의 지정

위에서 설명한 데이터 테이블 구성 요소에 대한 Indirect Addressing 접근 방법 외에 데이터 테이블 자체를 Indirect Addressing 을 통해 지정하고, 그 데이터의 참조가 가능합니다.

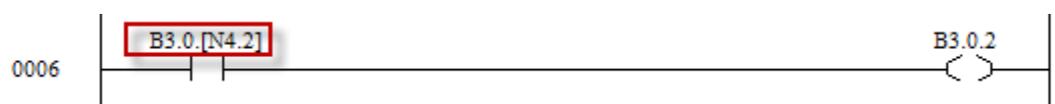


또한 아래의 예처럼, 데이터 테이블 번호 및 그 구성 요소 번호 까지도 Indirect Addressing 을 지원하므로, 마치 C 언어에서의 이중 포인터 처럼 보다 세밀한 제어가 가능합니다.



### Indirect Addressing 의 비트 제어 응용

위의 워드 데이터의 처리사와 같은 방법으로, 응용 할 수 있습니다.



**TIP**

Indirect Addressing 방법에서 데이터 테이블의 최대 크기인 4096이상의 값이 할당되지 않도록

주의하여 주십시오.

## 8장. PLC명령어

X8 PLC를 사용하기 위해서 필요한 명령어를 정리하여 설명합니다.

---

기본명령.....8-2

비교 명령어.....8-16

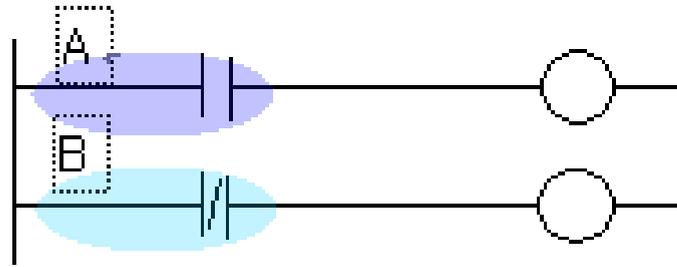
수학 및 산술 명령어.....8-22

Sequencing (순차 제어) 명령어.....8-54

## 기본명령

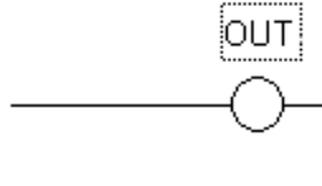
### NOC, NCC 명령

프로그램이 시작되는 가지(Branch)는 대부분 NOC나 NCC로 시작하여야 한다. 만약 없다면 항상 동작을 실행하게 된다. NOC, NCC으로 시작된 한 가지(Branch)는 반드시 하나 이상의 접점 삽입 후 종료 한다. 응용 명령어와 관련된 회로의 시작은 각각의 명령어에 따라 다르다.



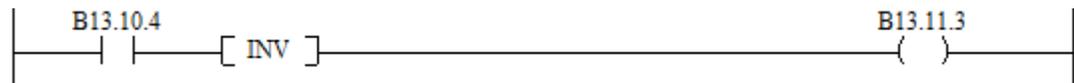
### OUT(OUT)명령

입력조건을 만족하는 지정한 접점을 ON시키는 명령으로 반드시 우측모선에 연결되어 있어야 하며 회로 중간에 올 수 없다. 외부 입출력(Y), 내부 릴레이(B) 접점을 사용한다.



### INV명령

이전의 상태를 반전시켜 다음 입력으로 전달하는 명령으로 A까지의 상태를 반전하여 B로 출력한다. 즉 INV명령은 그 명령의 직전까지의 연산 결과를 반전시키게 된다. 주의할 점은 INV명령은 모선에 직접 연결할 수 없다. INV명령은 이전 단계의 회로 상태를 직접 반전하므로 회로의 검증이나 실험단계에서 많이 사용할 수 있다.



### NOT 실행결과

실행전 실행후

A (ON) → (OFF)

A (OFF) → (ON)

하나의 회로가지에서 이전단계의 논리 상태를 반전하여 출력함.

## SET연산결과 ON유지 / RST연산결과 OFF유지

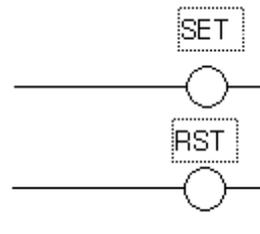
### 1) SET

- 외부출력(Y)/내부(B) 접점에 사용. 다중 사용가능.
- 입력조건이 만족되어 한번"ON"되면 RESET시키기 전까지 "ON" 유지.
- STOP 모드에서는 모든 출력이 OFF됨

### 2) RST

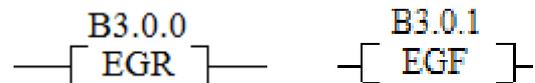
- 외부입출력/내부/정전유지 접점에 사용 다중사용가능. SET된 접점을 해제하는 기능.
- STOP 모드에서는 모든 출력이 OFF됨.

B영역을 OUT,SET,RST하면 STOP 및 전원 OFF후에도 그 상태가 유지됨. OUT명령어와 달리 출력 접점에도 두 번 이상 사용할 수 있다. 또한 정전유지 접점에 SET출력을 사용하면 정전이 되더라도 그 상태가 계속 유지 된다.

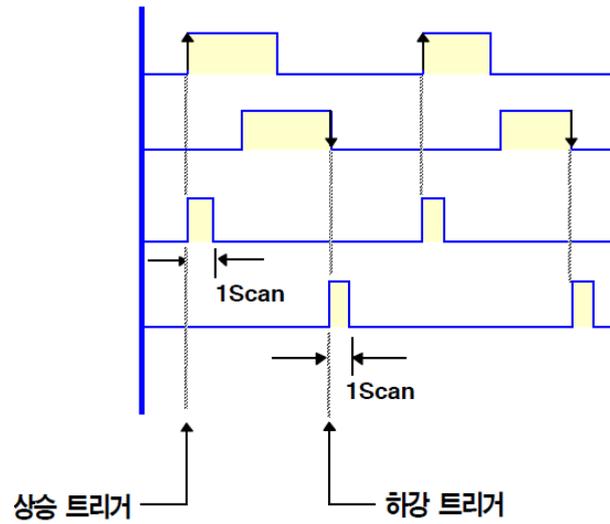


## EGR, EGF명령

EGR 및 EGF 명령은 해당 접점의 상승 또는 하강 시점을 검출하는 명령이다. 즉, 1스캔 시간 동안 ON되는 접점 명령을 말한다.

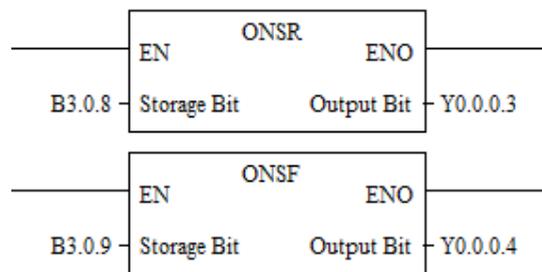


- ① EGR, EGF 명령은 같은 링 영역에서 중복사용 할 수 있습니다.
- ② EGR 명령은 OFF→ON으로 신호가 변화할 때 1스캔동안만 "ON"되는 접점이므로, OFF 혹은 ON으로 신호가 계속 유지될 때는"OFF"됨.
- ③ EGF 명령은 ON→OFF로 신호가 변화할 때 1스캔동안만 "ON"되는 접점이므로, OFF 혹은 ON으로 신호가 계속 유지될 때는"OFF"됨.
- ④ 하나의 번지로 상승, 하강엿지를 다중으로 사용가능합니다.



### ONSR(One Shot Rising), ONSF(One Shot Falling)

ONSR 과 ONSF 명령은 령의 상태 변화에 따른 이벤트의 발생을 감지하는 래칭 기능을 내장하고 있고, 여기에 출력 기능까지 추가된 명령입니다.



- ONSR : 이벤트의 발생 시점이 OFF → ON 일 경우 (상향 예지)
- ONSF : 이벤트의 발생 시점이 ON → OFF 일 경우 (하향 예지)

본 명령어는 2개의 파라미터를 가지고 있습니다.

- 저장 비트 : 비트 형태의 어드레스로, 바로 전의 스캔시 처리 결과를 저장 합니다.
- 출력 비트 : ONSR 또는 ONSF 의 결과에 따른 1 스캔 동안의 출력 데이터가 발생 됩니다.
- ONSR 이나 ONSF 명령을 재 실행 할 경우는, 령의 상태 가 TRUE 또는 FALSE 상태로 변경되어야 합니다.

• ONSR 명령 테이블

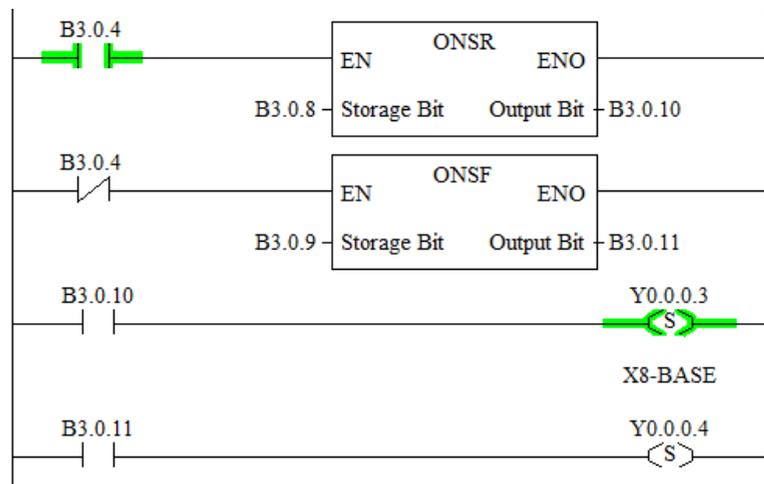
령의 상태 전이	저장비트	출력 비트
False → True (one scan)	bit is set	bit is set
True → True	bit is set	bit is reset
True → False, False → False	bit is reset	bit is reset

• ONSF 명령 테이블

령의 상태 전이	저장 비트	출력 비트
False → True (one scan)	bit is reset	bit is set
True → True	bit is reset	bit is reset
True → False, False → False	bit is set	bit is reset

**팁**

출력 비트의 데이터는 1스캔 동안 유지가 되므로, 유지 할 경우는 아래의 예와 같이 SET 명령등을 응용 할 수 있습니다.



### 타이머(한시동작 순서정지)

타이머 및 카운터 명령어는 출력 명령이며 시간에 관련된 제어를 할 수 있습니다.

명령어	설명
TON	타이머 ON
TOF	타이머 OFF
TONR	자기유지 타이머
RSTA	어큐물레이터 초기화(Reset)

타이머를 사용할 경우는 TM 이라는 타이머 데이터 테이블을 사용해야 하며, 그 데이터 테이블을 통하여 설정 및 모니터링을 할 수 있습니다.

타이머 데이터 테이블에는 타이머의 제어를 위한 3가지 구성 요소 있습니다.

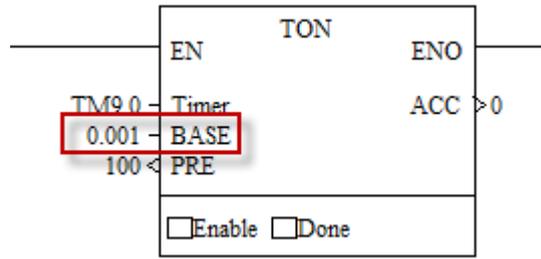
- 타이머 제어 및 상태 관리  
 설정치 (Preset) : 타이머 값의 목표 값입니다. 타이머 값이 설정치 값에 이르게 되면, 데이터 테이블의 "Done" 플래그가 ON 으로 세팅 됩니다.
- 어큐물레이터(Accumulator) : 타이머에서의 현재 누적된 타이머 값을 가지고 있습니다. 범위는 0 ~ 32767 입니다.

#### 타이머 설정 및 구성 요소

- 타이머의 설정 시간

설정 시간	타이머 범위	데이터 테이블 설정 값	
		TimeBase 1	TimeBase 0
0.001 초	0 ~ 32.767 초	0	1
0.01 초	0 ~ 327.67 초	0	0
0.1 초	0 ~ 32.767 초	1	0
1.00 초	0 ~ 32,767 초	1	0

위 테이블의 TimeBase 0, TimeBase1 값은 Ladder 상에서 자동 설정 됩니다. 아래 그림에서 BASE 값을 0.001로 설정할 경우, 여기와 연계된 TM 데이터 테이블의 TimeBase 0, TimeBase1 값이 자동 설정 됩니다.



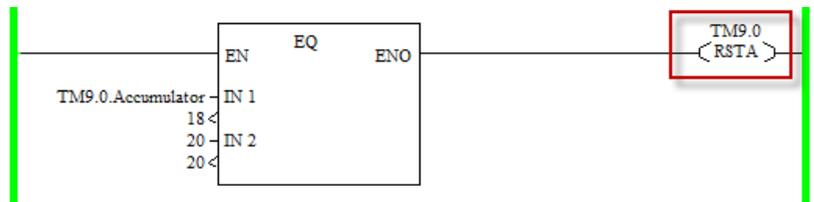
타이머 데이터 테이블의 내부 구성 요소 입니다.

구성원	데이터 종류	권한	설명
TimeBase0	Bit	RW	타이머의 기반 시간 설정
TimeBase0	Bit	RW	
Done	Bit	RO	타이머 동작 완료
TimerTiming	Bit	RO	타이머 동작 중
Enable	Bit	RO	타이머 사용 가능
Preset	Long Word	RW	설정치
Accumulator	Long Word	RW	어큐물레이터(누적치)

- 주의

타이머의 내부 구성 요소들을 타이머가 Enable 된, 동작 상태에서는 가능한 Access 하지 마십시오. 시스템 구조상 타이머 운영 시에는, 내부 인터럽트 처리가 사용되므로, 외부로부터의 타이머 구성원 참조 시 타이머의 정확성을 떨어트릴 수 있습니다.
- 팁

RSTA (Reset Accumulator) 명령을 사용하여, 타이머의 Accumulator 및 모든 비트들을 Reset 할 수 있습니다.



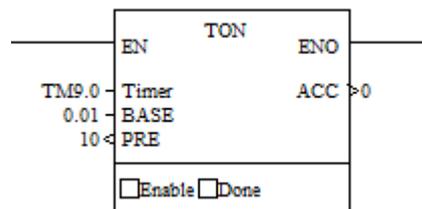
타이머의 정확도

타이머의 정확도는 타이머가 Enable 된 순간부터 타이머 주기가 완료 되었을 때까지의 총 시간을 참조하여 계산 되어 질 수 있습니다.

기반 시간	정확도
0.001 초	-0.001 ~ 0.00
0.01 초	-0.01 ~ 0.00
0.1 초	-0.1 ~ 0.00
1.00 초	-1.00 ~ 0.00

- 만일 타이머가 동작 중인 상황에서 래더 프로그램 중 JUMP, LBL, CALL, SBR 명령어에서 타이머가 위치해 있는 령을 건너 뛰게 될 경우, 타이머의 정확도가 떨어 질 수 있습니다.
- 만일 위의 명령어들(JUMP, LBL, CALL, SBR)을 통한 타이머가 위치한 지연 시간이 2.5초 이내의 경우는 시간 오차가 발생하지 않지만, 2.5초를 넘어서게 될 경우, 문제가 발생할 수 있습니다. 따라서 서브루틴 등의 타이머에 영향을 미치는 명령 사용시, 적어도 2.5초 안에는 타이밍 에러의 방지를 위해 타이머가 위치한 령이 스캔 되도록 프로그램이 되어야 합니다.

**TON - Timer, On-Delay**



명령어 유형 : 출력

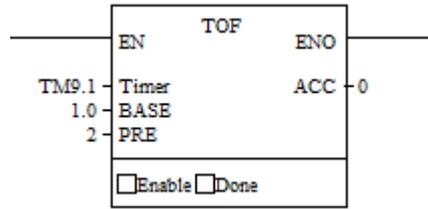
TON 명령은 타이머의 구동을 통한 출력 ON의 지연 기능을 수행 합니다.

TON 명령이 위치한 령의 조건의 참(TRUE) 일 경우, TON 명령에 설정 된 설정 값에 따라 타이머 동작을 하게 되며, Accumulator 에 저장되고 있는 누적 값이 Preset 설정치에 도달하게 되면, 타이머의 동작을 멈추게 됩니다.

만약, 타이머 명령이 위치한 령의 입력 조건의 거짓(FALSE)일 경우에는 타이머의 값 및 조건에 관계 없이, 누적 타이머 값을 가지고 있는 어큐물레이터 값은 리셋됩니다.

또한, Power Cycle (전원 ON/OFF시)이나 PLC의 작동 모드의 변경 시에도 타이머는 리셋 됩니다.

**TOF - Timer, Off-Delay**



명령어 유형 : 출력

TOF 명령은 타이머의 구동을 통한 출력 ON의 지연 기능을 수행합니다.

TOF 명령이 위치한 령의 조건의 거짓(FALSE) 일 경우, TOF 명령에 설정된 설정 값에 따라 타이머 동작을 하게 되며, Accumulator에 저장되고 있는 누적 값이 Preset 설정치에 도달하게 되면, 타이머의 동작을 멈추게 됩니다.

만약, 타이머 명령이 위치한 령의 입력 조건의 참(TRUE)일 경우에는 타이머의 값 및 조건에 관계 없이, 누적 타이머 값을 가지고 있는 어큐물레이터 값은 리셋됩니다.

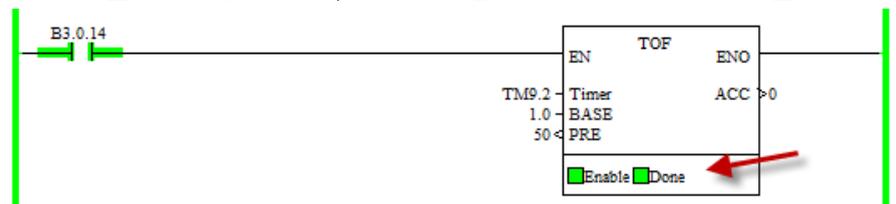
또한, Power Cycle (전원 ON/OFF시)이나 PLC의 작동 모드의 변경시에도 타이머는 리셋 됩니다.

주의 해야 할 점은, 모든 상태 플래그등의 모든 상태 표시가 TON 명령과는 과는 반대라는 것입니다.

아래는 타이머의 시작부터 완료 때까지의 상태 표시를 나타내주는 그림 입니다.

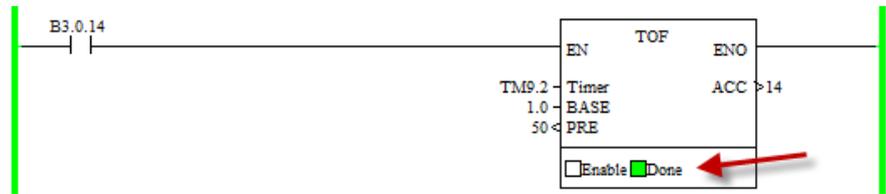
1) 령의 조건이 참(TRUE) 이므로, TOF가 미 동작.

TON 과는 다르게 Enable, Done의 값이 SET 으로 되어있음



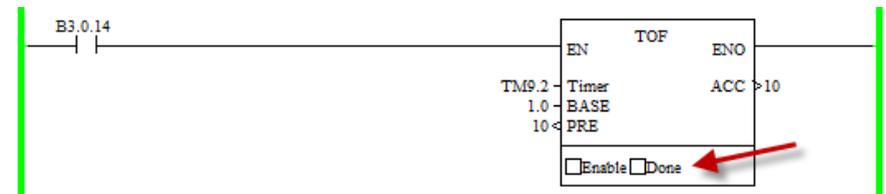
2) 령의 조건이 거짓(FALSE) 이므로, TOF가 동작

TOF가 동작하므로 Enable비트가 Clear 상태로 변경되고, 현재 동작 중이므로 Done 비트가 Set 되어 있음



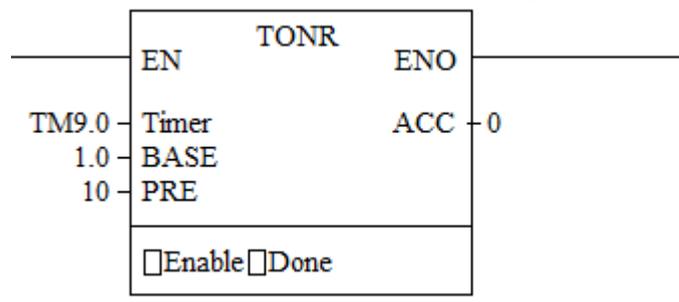
3) 령의 조건이 거짓(FALSE) 이므로, TOF가 동작 중이고 타이머가 종료 되었을 경우

Enable 및 Done 비트가 모두 Clear 상태로 되어 있음.



RSTA 명령어는 어큐물레이터 및 내부 플래그를 모두 리셋시키므로, TOF명령어 사용 시 주의를 요합니다.

**TONR – Retentive Timer On, On-Delay**



명령어 유형 : 출력

TOR 명령은 타이머의 구동을 통한 출력 ON의 지연 기능을 수행 합니다.

TON 명령이 위치한 령의 조건의 참(TRUE) 일 경우, TON 명령에 설정 된 설정 값에 따라 타이머 동작을 하게 되며, Accumulator 에 저장되고 있는 누적 값이 Preset 설정치에 도달하게 되면, 타이머의 동작을 멈추게 됩니다.

TOR 명령어는 아래의 조건에서 어큐물레이터 값을 저장/유지 합니다.

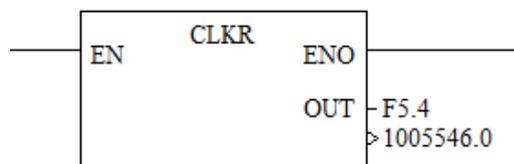
- 령의 조건이 거짓(FALSE) 가 될 때
- PLC 의 동작 모드가 RUN 모드 또는 TEST 모드에서 프로그램 모드로 변경 될 때
- 예러가 발생 했을 경우

또한 위 3가지 조건으로 인해 저장/유지되고 있는 어큐물레이터 값은 아래의 조건에서 기존의 저장/유지하고 있는 어큐물레이터 값으로부터 타이머가 동작 됩니다.

- 령의 조건이 참(TRUE) 로 변경되었을 경우
- PLC의 운영 모드가 프로그램 모드에서 RUN 또는 TEST 모드로 변경되었을 경우
- 될 때, 현재 저장된 값 이후로 타이머가 계속 가동됩니다.

Power Cycle (전원 ON/OFF) 및 동작 모드 변경 시 TOR 타이머는 계속 유지가 됩니다.

**CLKR - Read High Speed Clock Time**



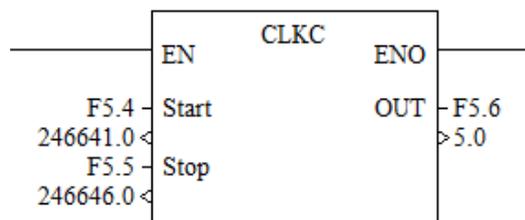
명령어 유형 : 출력

CLKR명령은 고성능의 타임 스탬프 기능을 출력합니다.

전원을 켜 후 매 10 μs 단위로 32비트의 할당된 데이터 값을 증가 시킵니다.

클럭 값이 0xFFFFFFFF (42949.67295 초)에 다다르면, 0으로 초기화되고 계속 적인 증가를 하게 됩니다.

**CLKC – Compute Time Difference**



명령어 유형 : 출력

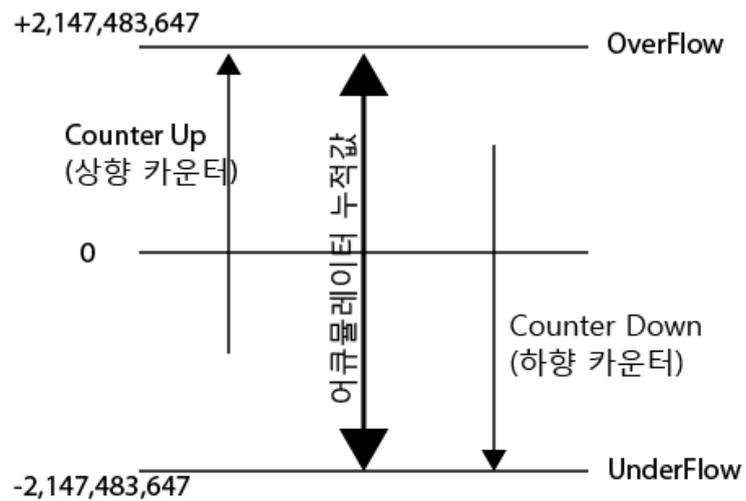
CLKR 명령을 통해 얻은 2개의 시간 값과의 차이를 계산 합니다. 즉, 두개의 이벤트가 발생 했을 때 각각의 시간값의 차이를 구할 수 있는 시간 간격을 계산 할 수 있는 함수입니다.

두개의 시간 값을 계산하므로 두개의 입력 값의 데이터 형태는 같아야 합니다.

### 카운터 명령

아래의 그림은 카운터의 사용에 대해서 나타내주고 있습니다. 카운터 값의 범위는 하한치인 -2,147,483,647 로부터 상한치 +2,147,483,647 까지 사용이 되며, 카운터의 값이 증가되어 +2,147,483,647을 넘어서면 Overflow 플래그가 SET(1)되고, 카운터 값이 감소되어 -2,147,483,647보다 낮아지면 Underflow 플래그가 SET(1) 이 됩니다.

카운터를 리셋하기 위해서는 RSTA 명령어가 사용 됩니다.



### CTU 와 CTD 명령

카운터 명령인 CTU와 CTD를 사용 하기 위해서는 데이터 테이블의 형태 중 CT를 사용하게 되고, CT의 구성원은 아래 그림과 같습니다.

Address	Type	Value	Meaning	Description
CT10.0	Counter	{...}		
- Underflow	Bool	0	OFF	
- Overflow	Bool	0	OFF	
- Done	Bool	0	OFF	
- CountDown	Bool	0	OFF	
- CountUp	Bool	0	OFF	
- Preset	Long	0		
- Accumulator	Long	0		
CT10.1	Counter	{...}		
CT10.2	Counter	{...}		
CT10.3	Counter	{...}		
CT10.4	Counter	{...}		

팁:

- CTD 명령 시에 카운터의 값이 Preset 값보다 작을 경우 카운트가 계속 됩니다
- CTU 명령 시에 카운터의 값이 Preset 값보다 클 경우 카운트가 계속 됩니다.

#### CT.10.0.Underflow

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.Underflow	bit	0 또는 1	status	RO

Accumulator 값이 -2,147,483,647보다 작을 때 ON(1) 됩니다.

#### CT.10.0.Overflow

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.Overflow	bit	0 또는 1	status	RO

Accumulator 값이 2,147,483,647보다 클 때 ON(1) 됩니다.

#### CT.10.0.Done

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.Done	bit	0 또는 1	status	RO

어큐물레이터 값이 Preset 값이 도달했을 때 ON(1) 됩니다.

#### CT.10.0.CountDown

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.CountDown	bit	0 또는 1	status	RO

CTD 명령에 의해 카운터가 감소되고 있을 경우, ON(1) 됩니다.

#### CT.10.0.CountUp

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.CountUp	bit	0 또는 1	status	RO

CTU 명령에 의해 카운터가 감소되고 있을 경우, ON(1) 됩니다.

#### CT.10.0.Preset

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.Preset	Long word	-32768 ~32767.	control	RW

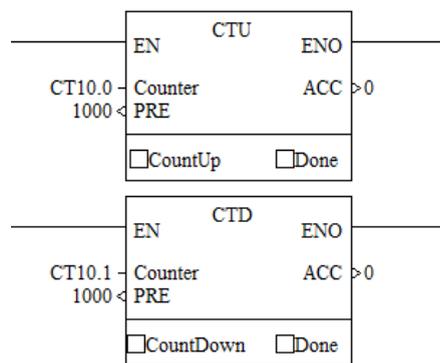
카운터의 설정치입니다.어큐물레이터의 값이 Preset 설정치에 다르면, Done 플래그가 ON (1) 됩니다.

**CT.10.0.Accumulator**

Address	데이터 형식	범위	종류	접근 권한
CT.10.0.Accumulator	Long word	-32768 ~32767	control	RW

현재 카운터의 값이 저장되어 있습니다. 령의 상태가 OFF(0)→ON(0) 으로 변경될 때 CTU명령에서는 어큐물레이터 값이 증가되고, CTD명령에서는 감소됩니다.  
 어큐레이터의 값은 자기 유지가 되어, 령의 상태가 OFF(0, FALSE) 상태가 되어도 그 값을 유지를 하게 되면, 파워 사이클에서도 그 값을 유지 합니다.  
 어큐레이터 값은 RSTA 명령에 의해서만 리셋 됩니다.

1) CTU(Up Counter), CTD(Down Counter)



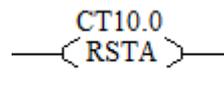
명령어 유형 : 출력

CTU 와 CTD 명령은 령의 상태가 FALSE → TRUE 로 변경될 때 어큐물레이터 값을 증가 또는 감소 시키는 명령어 입니다.

팁:

만약 령의 상태가 FALSE → TRUE 로 변경되는 신호가 외부 기기로부터 입력되는 신호일 경우, ON↔OFF 의 폭이 PLC 에서의 스캔 타임보다는 작을 경우의 처리를 위해, 카운터에서의 령의 상태 변이 체크를 위해 이 신호를 유지 시킬 수 있도록 해주어야 합니다.

2) RSTA - Reset Accumulator



명령어 유형 : 출력

RSTA 명령어는 카운터 , 타이머 및 각종 제어 구성 요소를 리셋 합니다.

각 요소 별 RSTA 명령에 의한 리셋 결과

타이머 구성원	카운터 구성원	제어용 구성 요소
Accumulator → 0	Accumulator → 0	Position → 0
Done → 0	Overflow → 0	Enable → 0
TimerTiming → 0	Underflow → 0	Enable Unload → 0
Enable → 0	Done → 0	Done → 0
	CountUp → 0	Empty → 0
	CountDown → 0	Error → 0
		Unload bit → 0

## 비교 명령어

다음은 비교 명령어들의 정리입니다.

명령어	설명
EQ : Compare for equal	== 두 개의 값이 같은지를 비교 합니다.
NE : Compare for not equal	!= , <>, ≠ 두 개의 값이 다른지를 비교 합니다.
LT : Compare for Less Than	< 첫 번째 인수가 두 번째 인수보다 작은지 비교합니다.
LE : Compare for Less Than or Equal	<= 첫 번째 인수가 두 번째 인수보다 작거나 같은지 비교합니다.
GT : Compare for Greater Than	> 첫 번째 인수가 두 번째 인수보다 큰지 비교합니다.
GE : Compare for Greater Than or Equal	>= 첫 번째 인수가 두 번째 인수보다 크거나 같은지 비교합니다.
EQM : Masked Compare for Equal	두개의 입력 값을 각각 Mask 데이터와 AND 후 비교
LIM - Limit Test	입력 값이 하한치와 상한치 사이에 있는 지를 비교

모든 비교 명령은 두 개의 입력 인자를 작게 됩니다.(EQM, LIM 명령은 추가되는 입력 인수가 있음.)

유의할 점은 2개의 인수모두 상수 데이터가 입력 될 수 없다는 것입니다.

단 2개의 인수 중 1개는 가능 합니다.

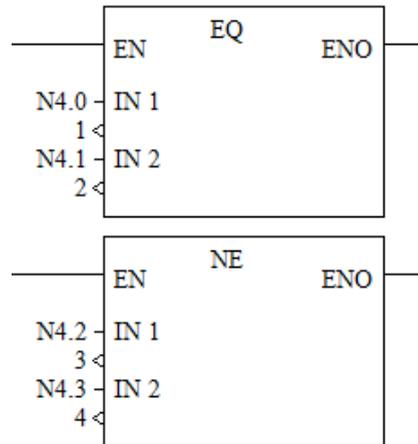
다음은 입력 인수에 대한 데이터 범위 입니다.

- WORD (16Bit) : 0 ~ 32,767
- LONG WORD(32Bit) : 0 ~2,147,483,647

다음은 비교 명령어를 사용하는데 있어서 주의해야 할 사항입니다.

- EQ, GE, GT, LE, LT 명령에서는 입력 A의 인자가 숫자가 아닐 경우, 령의 상태가 FALSE 로 변경됩니다.
- NE 명령에서는 두 개의 인수 모두가 숫자일 경우에도 령의 상태는 TRUE 로 유지 됩니다.

### EQ - Compare for Equal, NE - Compare for Not Equal



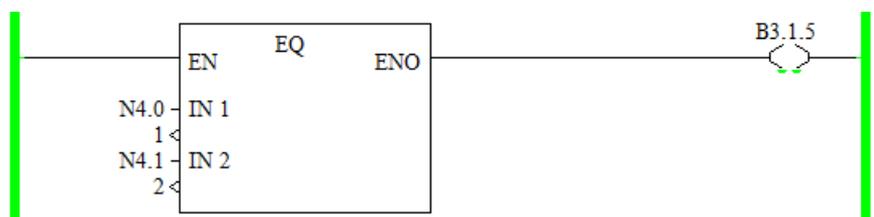
명령어 유형 : 입력

- EQ : 입력 인자 두 개의 데이터가 같은지를 비교합니다.
- NE : 입력 인자 두 개의 데이터가 다른지를 비교합니다.

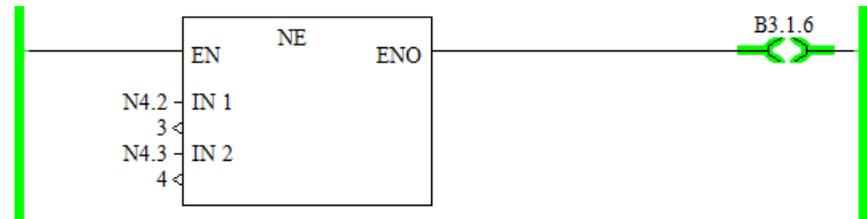
명령어	연산 방법	연산 결과
EQ	A = B	TRUE
	A ≠ B	FALSE
NE	A = B	FALSE
	A ≠ B	TRUE

사용 예

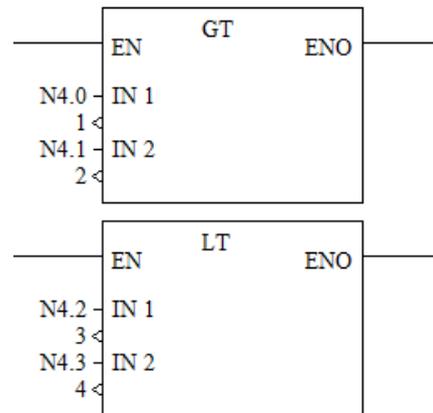
- EQ 명령. IN1 =1 , IN2=2 일 경우, 두 개의 입력 데이터가 서로 다르므로, 출력 B3.1.5 비트는 OFF 상태가 유지 됩니다.



- NE 명령. IN1 =3 , IN2=4 일 경우, 두 개의 입력 데이터가 서로 다르므로, 출력 B3.1.6 비트는 ON 상태가 됩니다.



**GT(Compare for Greater Than), LT(Compare for Less Than)**



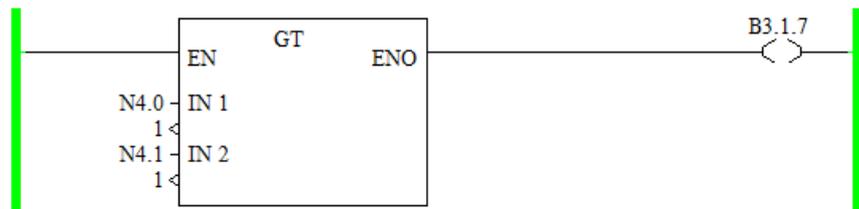
명령어 유형 : 입력

- GT : 입력 A 가 입력 B 보다 큰지를 비교 합니다.
- LT : 입력 B 가 입력 A 보다 큰지를 비교 합니다.

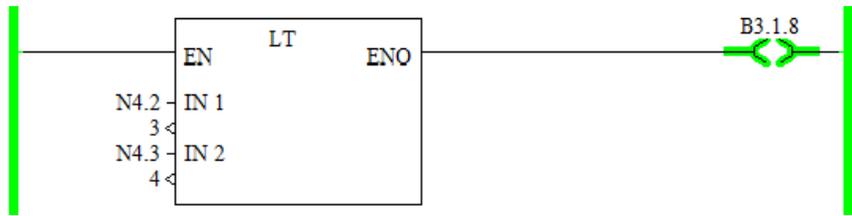
명령	연산 방법	연산 결과
GT	$A > B$	TRUE
	$A \leq B$	FALSE
LT	$A \geq B$	FALSE
	$A < B$	TRUE

사용 예

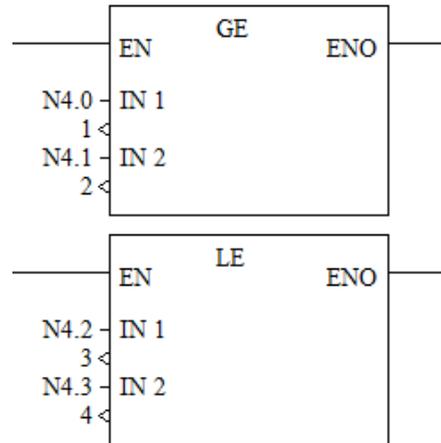
- GT 명령. IN1 =1 , IN2=1 일 경우, IN 1가 IN 2보다 큰지 비교를 하는데, 두 개의 입력 데이터가 서로 같으므로, 출력 B3.1.7 비트는 OFF 상태가 유지 됩니다.



- LT 명령. IN1 =3 , IN2=4 일 경우, IN 1가 IN 2보다 큰지 비교를 하는데, IN 2가 더 크므로 출력 B3.1.8 비트는 ON 상태가 유지 됩니다.



**GE(Compare for Greater Than or Equal),  
LE(Compare for Less Than or Equal)**



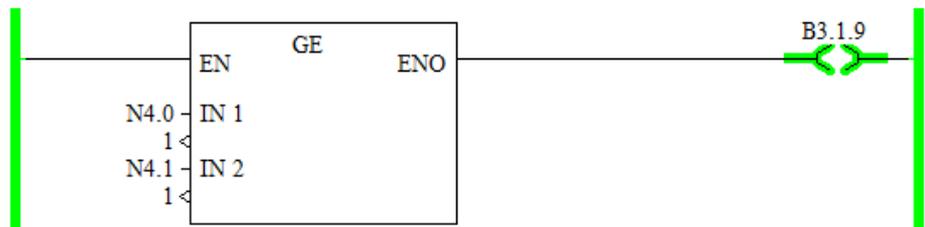
명령어 유형 : 입력

- GE : 입력 A 가 입력 B 보다 크거나 같은지를 비교 합니다.
- LE : 입력 B 가 입력 A 보다 크거나 같은지를 비교 합니다.

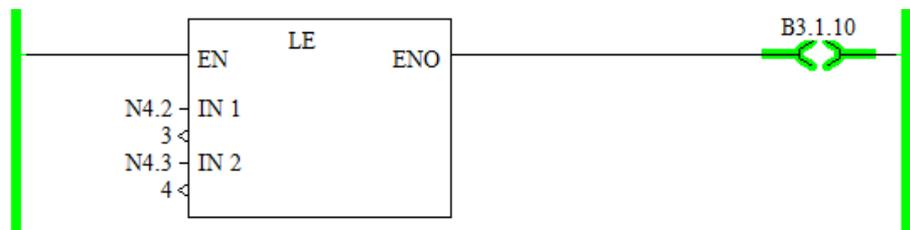
명령	연산 방법	연산 결과
GE	$A \geq B$	TRUE
	$A < B$	FALSE
LE	$A > B$	FALSE
	$A \leq B$	TRUE

사용 예

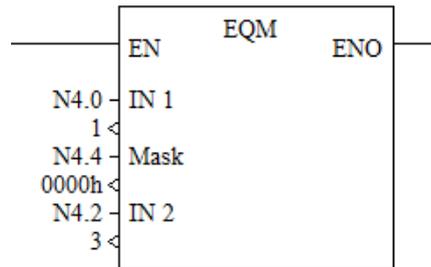
- GE 명령. IN1 =1 , IN2=1 일 경우, IN 1가 IN 2보다 크거나 같은지 비교를 하는데, 두 개의 입력 데이터가 서로 같으므로, 출력 B3.1.9 비트는 ON 상태가 유지 됩니다.



- LE 명령. IN1 =3 , IN2=4 일 경우, IN 2가 IN 1보다 크거나 같은지 비교를 하는데, IN 2가 더 크므로 출력 B3.1.10 비트는 ON 상태가 유지 됩니다.



### EQM - Mask Compare for Equal

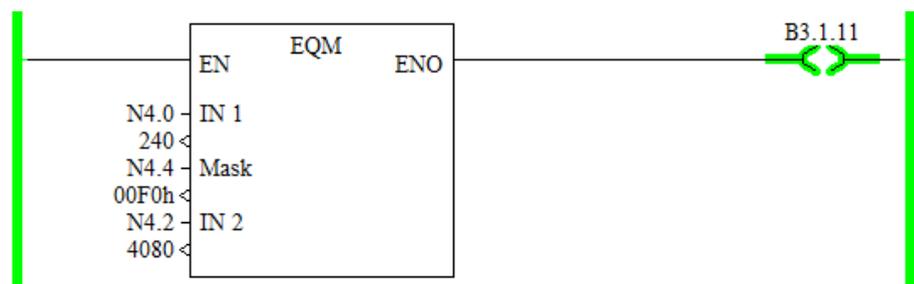


명령어 유형 : 입력

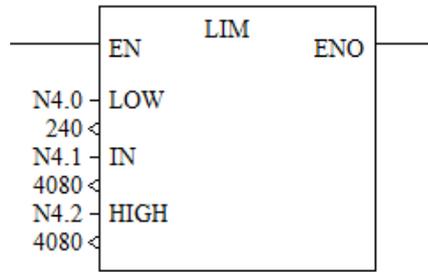
EQM 명령은 입력 A와 Mask 데이터의 AND 결과 값과 입력 B와 Mask 데이터와 AND 결과값과의 결과가 같은지를 비교 합니다.

사용 예

- EQM 명령. IN1 =240 (0F0H) , IN2=4080 (FF0H) 일 경우, 두 개의 값을 F0H와 Andmftlzlz 후, 두개의 결과 같이 같은 지를 비교 한다.
  - IN1=240 (0F0H) AND MASK 0F0H = F0H
  - IN1=240 (FF0H) AND MASK 0F0H = F0H
  - 두 개의 입력 데이터의 Mask 한 결과가 서로 같으므로, 출력 B3.1.11 비트는 ON 상태가 유지 됩니다.



### LIM - Limit Test



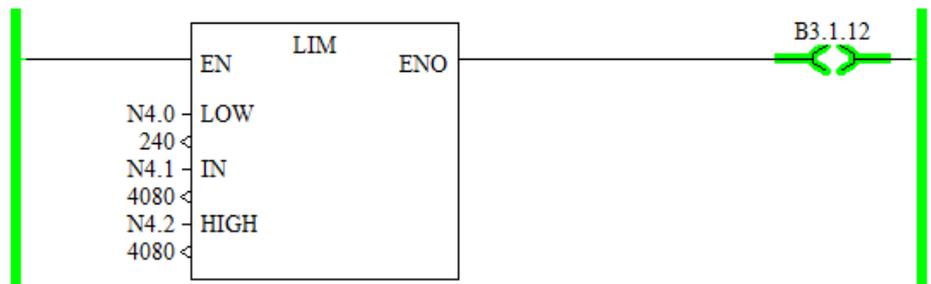
명령어 유형 : 입력

EQM명령은 입력 값이 하한치와 상한치 사이에 있는 지를 비교합니다.

연산 조건	결과값
하한치 ≤ 입력 ≤ 상한치	TRUE
입력 < 하한치 또는 입력 > 상한치	FALSE
상한치 < 입력 < 하한치	FALSE
입력 ≥ 상한치 또는 입력 ≤ 하한치	TRUE

사용 예

- LIM 명령. IN =4080 ,하한치 =240, 상한치=4080 일 경우, 입력 값이 상한치 하한치 사이에 있으므로 출력 B3.1.12 비트는 ON 상태가 유지 됩니다.



## 수학 및 산술 명령어

다음은 X8 PLC에서 제공하는 다양한 수학 명령 및 기타 연산 명령어들에 대한 설명입니다.

### 수학 및 산술 명령

명령어	설명
ADD	가산, 덧셈
SUB	감산, 뺄셈
MUL	승산, 곱셈
DIV	제산, 나눗셈
NEG	부호 반전
CLR	0으로 초기화(클리어)
ABS	절대값 계산
SQRT	제곱근, Root
SCAL	Scale a value
SCAP	입력 값의 사용자 정의 범위의 비례 변환
SIN	SIN()
COS	COS()
TAN	TAN()
ASIN	ARC SIN()
ACOS	ARC COS()
ATAN	ARC TAN()
DEG	Radian 값을 Degrees 으로 변경
RAD	Degrees 값을 Radian 으로 변경
LN	자연 로그
LOG	일반 로그
POW	승. 거듭 제곱
CALC	입력된 수식에 의한 연산 결과를 대상에 입력

### 수학, 산술 명령에 대한 설명

#### 상수의 범위

Word : -32768 ~ 32767

Long Word : -2,147,483,648...2,147,483,647

입력 1, 입력 2는 각각 상수가 될 수 있으나, 동시에 상수는 지원이 되지 않습니다.

입력 1 과 입력 2 는 서로 다른 데이터 형태를 가질 수 있습니다. 그러나 출력은 출력의 데이터 형태에 맞추어 저장 됩니다.

Overflow 가 발생 했을 경우, 데이터 테이블의 특수 레지스터 (SR2)의 구성 요소 중 Math부분의 SR2.14 를 사용하여, Overflow 가 발생 했을 경우 처리 방법을 다르게 할 수 있습니다.

- SET(1) : 입력 값이 양수 일 경우 32,767 (WORD) 또는 2,147,483,647 (Long Word) 가 출력 값으로 저장 되고, 음수 일 경우 -32,767 (WORD) 또는 -2,147,483,647 (Long Word) 가 저장 됩니다.
- CLEAR(0) : 각 데이터 종류에 따라, 범위가 벗어난 부호 없는 값이 출력에 저장 됩니다.

#### 수학 산술 관련 플래그

수학, 산술 명령의 상태를 표시해주는 플래그는 SR 데이터 테이블의 Math 항목에 위치해 있으며, 각각의 의미를 설명합니다.

Address	Type	Value	Meaning	Description
SR2.2.14	Bool	0	OFF	Math Overflow Selection
SR2.5.0	Bool	0	OFF	Overflow Trap
SR2.0.0	Bool	0	OFF	Math, Carry Flag
SR2.0.1	Bool	0	OFF	Math, Underflow/Overflow Flag
SR2.0.2	Bool	0	OFF	Math, Zero Flag
SR2.0.3	Bool	0	OFF	Math, Sign Flag
SR2.13-14	Long	0		Math Register Low
SR2.13	Integer	0		Math Register Low
SR2.14	Integer	0		Math Register High

- SR2.2.14 : Math Overflow Selection  
Overflow 발생시에 본 비트의 설정에 따라 출력 값이 다르게 처리 됩니다.
- SR2.5.0 : Overflow Trap  
Overflow 플래그가 SET 되었을 때 SET 됩니다.  
Overflow 플래그가 시스템에 영향을 미칠 수 있으므로, 본 비트를 래더상에서 RST명령을 통해 강제 클리어 시켜 놓을 수 있습니다.
- SR2.0.0 : Carry Flag  
Carry 발생시 SET 됩니다.

- SR2.0.1 : Underflow , Overflow Flag  
연산 결과가 출력 대상의 범위에 들지 않을 경우 발생합니다.
  - SR2.0.2 : Zero Flag  
출력이 0일 때 설정 됩니다.
  - SR2.0.3 : Sign Flag  
출력 결과가 음수 일 때 설정 됩니다. 음수의 경우는 MSB (Most Significant Bit)가 1로 표기 됩니다.
  - SR2.13 : Register Low : 나눗셈 시 나머지가 저장 됩니다
  - SR2.14 : Register High : 나눗셈 시 몫이 저장 됩니다.
- X8의 연산, 산술 명령에서 데이터의 종류는 아래의 테이블에 따라 결정 됩니다.

입력 1	입력 2	출력
Integer	Integer	Integer
Integer	Long	Long
Integer	Float	Float
Long	Integer	Long
Long	Long	Long
Long	Float	Float
Float	Integer	Float
Float	Long	Float
Float	Float	Float

### 부동 소수점

X8 PLC 는 부동 소수점 형식 (F)의 32비트 크기의 데이터 테이블을 지원하며, 부동 소수점은 IEEE-754 사양을 준수합니다. 다른 데이터 테이블과 마찬가지로 1536개의 내부 구성 요소를 지원 합니다.

#### 부동 소수점 형식의 데이터 구조

MSB												LSB
31	30	29	....	23	22	21	20	19	...	2	1	0
부호	지수					가수						

- Bit 31은 MSB 로써, 부호를 의미합니다.

- Bit 30 ~ Bit 23 은 지수를 의미합니다.
- Bit 22 ~ Bit 00 은 가수를 의미합니다.

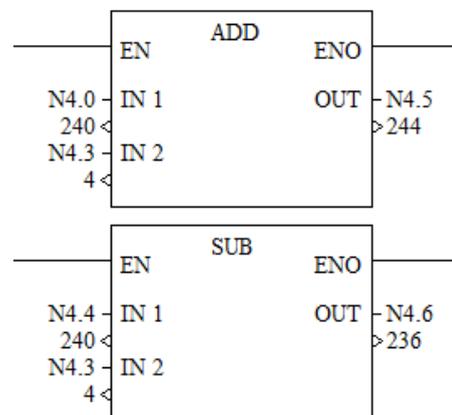
LSB에 대한 Round to Even 규칙

- 중간값인 5 에 대해서 반올림 자리 바로 윗자리 숫자가 짝수에 가까워지도록 올림이나 내림을 수행합니다

부동 소수점의 예외 값

- 0 : 0의 가수와 지수로 표현됩니다.
- Denormalized : 아주 작은 값으로 0에 가까운 무의미한 값입니다. 실행 속도를 높이기위해 0으로 취급됩니다.
- ∞ 는 255의 지수값과 0의 가수 부분으로 표현되는데, 오버플로 에러 발생시 양과 음의 무한대가 모두 발생 합니다.
- NaN(Not a Number)은 연산 과정에서 잘못된 입력을 받았음을 나타내는 기호입니다. 특히 부동 소수점 연산에서 사용을 하는데, 예를 들어, 대부분의 부동 소수점 장치는 음수에 대해서 제곱근을 구하려는 연산에 대해서 이것은 불가능(Invalid)하다는 메시지와 함께 NaN 값을 반환하게 됩니다.

**ADD – Add, SUB - Subtract**

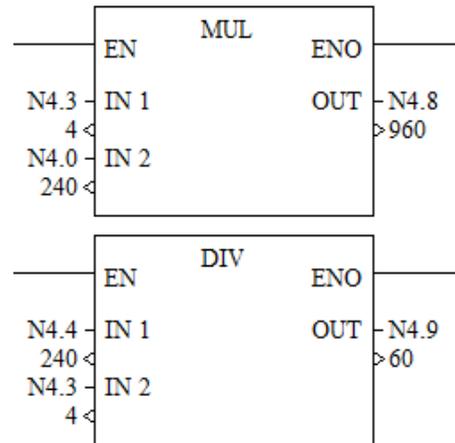


명령어 유형 : 출력

ADD : 입력 1 와 입력 2의 값을 합산하여 그 결과를 출력합니다.

SUB : 입력 1 에서 입력 2의 값을 빼서 그 결과를 출력합니다.

**MUL - Multiply      DIV - Divide**

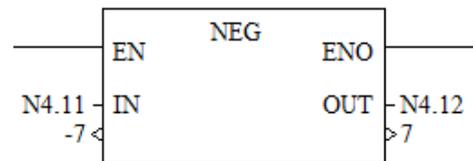


명령어 유형 : 출력

MUL : 입력 1 와 입력 2의 값을 곱하여 그 결과를 출력합니다.

DIV : 입력 1 에서 입력 2의 값을 나누어서 그 결과를 출력합니다.

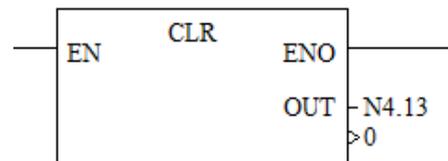
**NEG - Negate**



명령어 유형 : 출력

입력 데이터의 부호를 반전 시켜 출력 합니다.

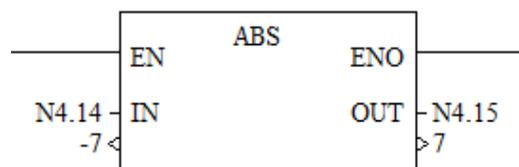
**CLR - Clear**



명령어 유형 : 출력

지정된 데이터 테이블의 값을 0으로 초기화 시킵니다.

**ABS - Absolute Value**



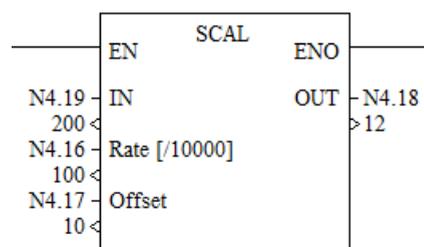
명령어 유형 : 출력

지정된 데이터의 절대값을 출력 합니다.

ABS 연산의 결과가 정상적이지 않을 경우 , 아래와 같은 제한이 따르게 됩니다.

입력	출력	제한 사항
F	N	반올림된 데이터가 출력 반올림 후, Overflow 가 발생하면 32767 또는 2147836647 이 출력되고 , SR2.2.14 Math Overflow Selection 비트는 무시 됩니다. Carry 플래그는 RESET 출력대상이 모두 0인 경우 Zero 플래그 SET Overflow Trap 플래그는 Overflow 플래그가 SET일 경우 SET 부호있는 결과 값이 ∞, 숫자가 아닐 경우 또는 출력 대상에 입력을 할 수 없을 경우 SET 출력의 MSB 가 SET 되어 있을 경우 Sign 플래그는 SET
∞	N	32767 또는 2147836647 이 출력
N	N	SR2.2.14 Math Overflow Selection 비트가 0 인 경우, 설정된 경우, 32767 또는 2147836647 이 출력됩니다. 입력이 음수일 때 Carry 플래그는 SET 결과 값이 부호가 있고, 대상에 입력이 안될 경우 Overflow 플래그는 SET 출력대상이 모두 0인 경우 Zero 플래그 SET Overflow Trap 플래그는 Overflow 플래그가 SET일 경우 SET

**SCAL – Scale with Slope**

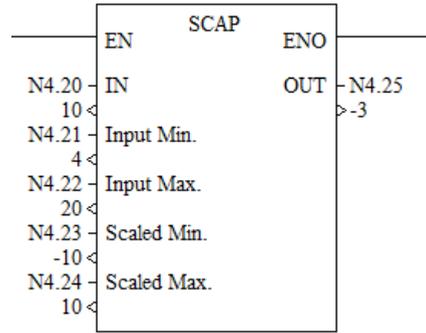


명령어 유형 : 출력

입력 값과 기울기를 곱한 후, 오프셋 값을 더하여 출력 합니다.

$$\text{즉, Scale 출력 값} = \frac{\text{입력} \times \text{기울기}}{10000} + \text{오프셋}$$

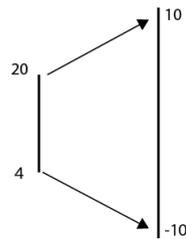
**SCAP – Scale with Parameter**



명령어 유형 : 출력

입력값이 입력 범위의 최소, 최대 값 안에서 입력 될 경우, 이에 비례한 SCALE된 값으로 변환 됩니다.

좌측의 래더에 표현된 숫자의 예는 입력의 범위가 4 ~ 20 이고 이에 비례한 출력 값을 -10 ~ 10 이라고 했을 때, 입력이 10 이 입력 되면, 여기에 비례한 출력으로 -3이 출력 되는 예제입니다. (4~20mA 입력을 -10V ~ 10V 출력으로 변경할 경우)

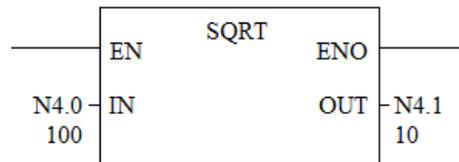


이를 수식으로 나타낸다면

$$Scale\ 출력 = \left[ \frac{(y1 - y0)}{(x1 - x0)} \right] \times (x - x0) + y0$$

로 표현 될 수 있습니다.

**SQRT - Square Root**



명령어 유형 : 출력

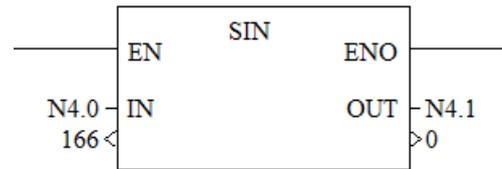
SQRT 명령은 루트 계산을 합니다.

SR2 내부의 Math 부분의 구성원인 Math Carry Flag 는 그 결과 값이 (-)일 경우 ON으로 세팅 됩니다.

SQRT의 데이터 테이블 형태에 따른 값의 범위는 아래와 같습니다.

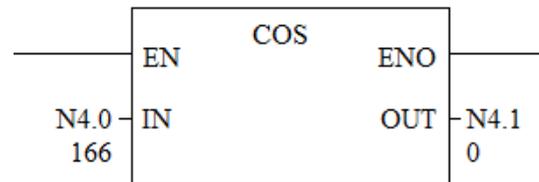
WORD범위 : -32768 ~ 32767

LONG WORD 범위 : -2,147,483,648 ~ 2,147,483,647

**SIN - Sine**

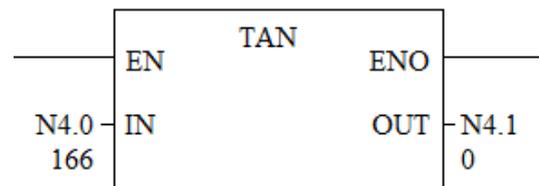
명령어 유형 : 출력

입력 데이터 테이블의 라디안 값에 대한 SIN 계산 값을 출력 대상에 저장 합니다. 입력에 적용 될 수 있는 데이터의 종류로는 WORD, LONG WORD, FLOAT등의 모든 경우의 데이터를 사용 할 수 있으며, 이 데이터는 FLOAT 형태로 변경되어 계산이 되고, 그 결과 값은 출력 데이터의 종류에 맞게 자동 변환 됩니다.

**COS - Cosine**

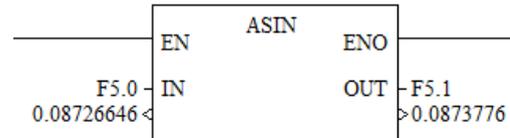
명령어 유형 : 출력

입력 데이터 테이블의 라디안 값에 대한 COS 계산 값을 출력 대상에 저장 합니다. 입력에 적용 될 수 있는 데이터의 종류로는 WORD, LONG WORD, FLOAT등의 모든 경우의 데이터를 사용 할 수 있으며, 이 데이터는 FLOAT 형태로 변경되어 계산이 되고, 그 결과 값은 출력 데이터의 종류에 맞게 자동 변환 됩니다.

**TAN - Tangent**

명령어 유형 : 출력

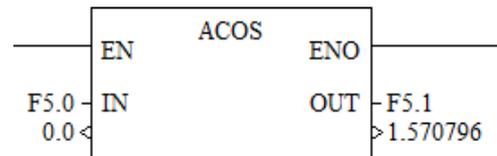
입력 데이터 테이블의 라디안 값에 대한 TAN 계산 값을 출력 대상에 저장 합니다. 입력에 적용 될 수 있는 데이터의 종류로는 WORD, LONG WORD, FLOAT등의 모든 경우의 데이터를 사용 할 수 있으며, 이 데이터는 FLOAT 형태로 변경되어 계산이 되고, 그 결과 값은 출력 데이터의 종류에 맞게 자동 변환 됩니다.

**ASIN - Arc Sine**

명령어 유형 : 출력

입력 데이터 테이블의 래디안 값에 대한 Arc SIN 계산 값을 출력 대상에 저장 합니다. 입력에 적용 될 수 있는 데이터의 종류로는 WORD, LONG WORD, FLOAT등의 모든 경우의 데이터를 사용 할 수 있으며, 이 데이터는 FLOAT 형태로 변경되어 계산이 되고, 그 결과 값은 출력 데이터의 종류에 맞게 자동 변환 됩니다.

입력 데이터는 -1 보다 크거나 같아야 하고 1보다 작거나 같아야 합니다. 출력 데이터는  $-\frac{\pi}{2}$  보다 크거나 같고,  $\frac{\pi}{2}$ 보다 작거나 같습니다.

**ACOS - Arc Cosine**

명령어 유형 : 출력

입력 데이터 테이블의 래디안 값에 대한 Arc SIN 계산 값을 출력 대상에 저장 합니다. 입력에 적용 될 수 있는 데이터의 종류로는 WORD, LONG WORD, FLOAT등의 모든 경우의 데이터를 사용 할 수 있으며, 이 데이터는 FLOAT 형태로 변경되어 계산이 되고, 그 결과 값은 출력 데이터의 종류에 맞게 자동 변환 됩니다.

입력 데이터는 -1 보다 크거나 같아야 하고 1보다 작거나 같아야 합니다. 출력 데이터는 0보다 크거나 같고,  $\pi$  보다 작거나 같습니다.

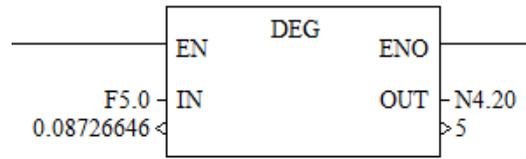
**ATAN - Arc Tangent**

명령어 유형 : 출력

입력 데이터 테이블의 래디안 값에 대한 Arc TAN 계산 값을 출력 대상에 저장 합니다. 입력에 적용 될 수 있는 데이터의 종류로는 WORD, LONG WORD, FLOAT등의 모든 경우의 데이터를 사용 할 수 있으며, 이 데이터는 FLOAT 형태로 변경되어 계산이 되고, 그 결과 값은 출력 데이터의 종류에 맞게 자동 변환 됩니다.

출력 데이터는  $-\frac{\pi}{2}$  보다 크거나 같고,  $\frac{\pi}{2}$ 보다 작거나 같습니다.

**DEG - Radians to Degrees**

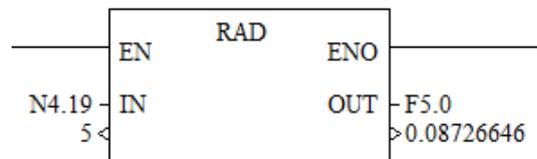


명령어 유형 : 출력

입력 데이터 테이블의 래디안을 Degree(도) 로 변경 출력 합니다.

계산 수식은 Degree = 입력 ×  $\frac{180}{\pi}$  입니다.

**RAD - Degrees to Radians**

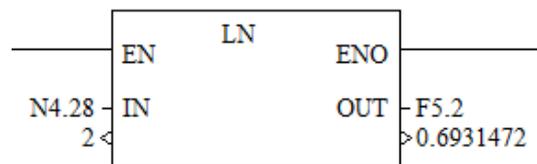


명령어 유형 : 출력

입력 데이터 테이블의 Degree (도)를 래디안으로 변경 출력 합니다.

계산 수식은 Radian = 입력 ×  $\frac{\pi}{180}$  입니다.

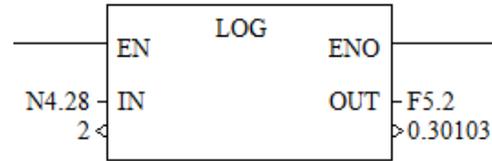
**LN - Natural Log**



명령어 유형 : 출력

입력 데이터 테이블의 값을 자연 로그를 연산 후 , 출력합니다.

**LOG - Base 10 Logarithm**

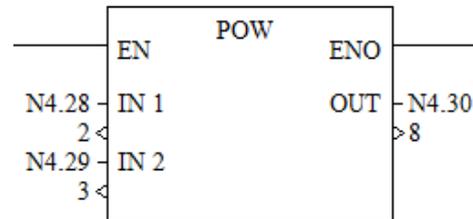


명령어 유형 : 출력

입력 데이터 테이블의 값을 기수 10 로그를 연산 후 , 출력합니다.

입력 데이터는 0보다 커야 합니다.

**POW - X Power Y**

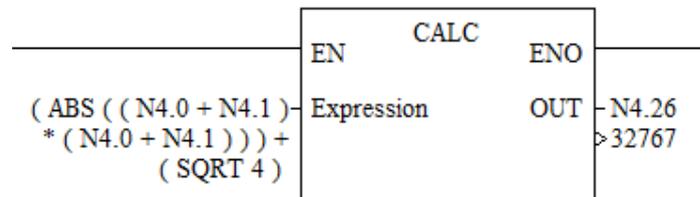


명령어 유형 : 출력

입력 1(x) 과 입력 2(y) 의 거듭 제곱을 계산하여 출력합니다.

$$\text{출력} = x^y$$

**CALC - Calculate**



명령어 유형 : 출력

CALC 명령어는 X8 PLC를 사용 할 때 사용자의 편의를 줄 수 있는 기능으로, 4칙 연산, 각종 산술 식 및 변환을 본 기능을 통해서 단 1개의 명령어로 바로 처리 할 수 있는 편리한 기능입니다. 아래의 테이블은 계산 수식으로 지원되는 함수의 목록입니다.

명령	기호	의미
ADD	+	덧셈
SUB	-	뺄셈
MUL	*	곱셈
DIV	/	나눗셈
SQRT	SQRT	루트(근)
SIN	SIN	사인
COS	COS	코사인
TAN	TAN	탄젠트
ASIN	ASIN	아크 사인
ACOS	ACOS	아크 코사인
ATAN	ATAN	아크 탄젠트
AND	AND	논리곱
OR	OR	논리합
XOR	XOR	배타적 논리합
NOT	NOT	논리적 반전
NEG	- (NEG)	부호 반전
LN	LN	자연 로그
LOG	LOG	10 기반 로그
ABS	ABS	절대치
DEG	DEG	라디안 → 디그리
RAD	RAD	디그리 → 라디안
POW	POW	거듭제곱
BCD	BCD	BIN → BCD
BIN	BIN	BCD → BIN

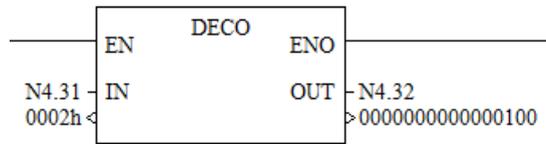
수식을 입력하면, 시스템에서 자체적으로 괄호( ' ( ) ' ) 를 자동 입력 합니다.

## 변환 명령어

본 명령어들은 멀티플렉스, 디멀티플렉스 명령외에 2진수와 10진수를 서로 변환하는 명령어들입니다.

명령어	설명
DECO	Decode 4 to 1-of-16 4비트의 데이터를 16비트 데이터로 만들어 (Multiplexer) 해당 1 비트를 ON 시킵니다.
ENCO -	Encode 1-of-16 to 4 16비트의 데이터를 4비트로 인코딩합니다.
BIN	Convert From Binary Coded Decimal 입력 BCD 값을 10진수로 변경 출력 합니다.
BCD -	Convert to Binary Coded Decimal 10진수 값을 BCD 로 변환합니다.
GRAY	Gray Code 그레이 코드를 10진수로 변환 출력합니다.

**DECO - Decode 4 to 1-of-16**



명령어 유형 : 출력

DECO 명령은 입력 데이터의 하위 4비트를 사용하여, 16비트로 멀티플렉스 한 값을 출력 합니다.

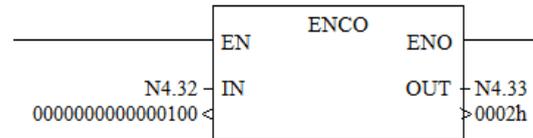
다음은 4비트의 입력 값에 대한 출력값을 표시합니다.

**Decode 4 to 1-of-16**

입력					출력															
15 to 04	03	02	01	00	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
x	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
x	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0
x	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0
x	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0
x	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
x	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0
x	0	1	1	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
x	0	1	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0
x	1	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
x	1	0	0	1	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0
x	1	0	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
x	1	0	1	1	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0
x	1	1	0	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
x	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0
x	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
x	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

x = Not Used

**ENCO -Encode 1-of-16 to 4**



명령어 유형 : 출력

ENCO 명령은 입력 데이터의 하위 4비트를 사용하여, 16비트로 멀티플렉스 한 값을 출력 합니다.

명령 실행에 따른 플래그 변화

SR주소	플래그	동작 설명
SR2.0.0	Carry Flag	항상 리셋
SR2.0.1	Overflow Flag	입력에서 2개의 비트가 SET일 경우 SET되고 나머지는 Reset.
SR2.0.2	Zero Flag	출력값이 0 일 경우 SET 이외는 Reset
SR2.0.3	Sign Flag	항상 Reset

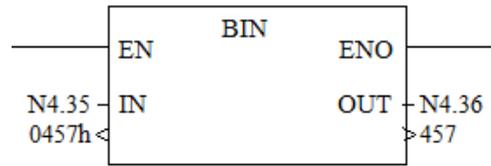
다음은 16비트의 입력 값에 대한 출력값을 표시합니다.

**Encode 1-of-16 to 4**

입력																출력				
15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00	15 to 04	03	02	01	00
x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0
x	x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	1
x	x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	1	0
x	x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	1
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	1	0	0
x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	1	0	1
x	x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	1	1	0
x	x	x	x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	1	1	1
x	x	x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1
x	x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
x	x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	0
x	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1

x = determines the state of the flag

**BIN - Convert from BCD**



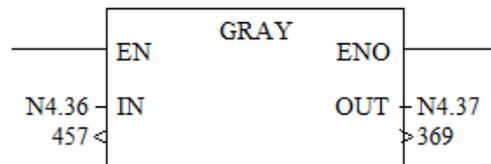
명령어 유형 : 출력

BIN 명령은 입력 BCD데이터를 정수 값으로 출력 합니다.

SR주소	플래그	동작 설명
SR2.0.0	Carry Flag	항상 리셋
SR2.0.1	Overflow Flag	입력에서 2개의 비트가 SET일 경우 SET 되고 나머지는 Reset.
SR2.0.2	Zero Flag	출력값이 0 일 경우 SET 이외는 Reset
SR2.0.3	Sign Flag	항상 Reset

좌측 예제는 입력 1111 (457H) 를 입력했을 때의 출력인 457을 보여 주고 있습니다.

**GRAY - Gray Code**



명령어 유형 : 출력

입력의 GRAY 코드 데이터를 10진수로 변환 출력 합니다.

SR주소	플래그	동작 설명
SR2.0.0	Carry Flag	항상 리셋
SR2.0.1	Overflow Flag	입력된 GRAY코드가 (-) 일 경우 SET일 경우 SET되고 나머지는 Reset.
SR2.0.2	Zero Flag	출력값이 0 일 경우 SET 이외는 Reset
SR2.0.3	Sign Flag	항상 Reset
SR2.5.0	Overflow Trap	Overflow 플래그가 SET 일 경우 SET. 이외는 Reset

### 논리 연산 명령어

X8PLC의 논리 연산 명령어는 4개가 있으며, 아래와 같습니다.

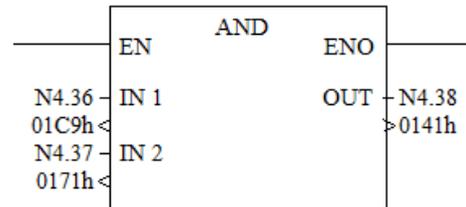
명령	설명
AND	Bit-Wise AND 두 개의 입력에 대한 논리곱을 출력합니다.
OR	Logical OR 두 개의 입력에 대한 논리합을 출력합니다.
XOR	Exclusive OR 두 개의 입력에 대한 배타적 논리합 결과를 출력합니다.
NOT	Logical NOT 입력에 대한 반전 값을 출력합니다.

논리 연산 명령어의 데이터 범위는 아래와 같습니다.

- WORD : -32768 ~ 32767
- LONG WORD : -2,147,483,648 ~ 2,147,483,647

SR주소	플래그	동작 설명
SR2.0.0	Carry Flag	항상 리셋
SR2.0.1	Overflow Flag	항상 리셋
SR2.0.2	Zero Flag	출력값이 0 일 경우 SET 이외는 Reset
SR2.0.3	Sign Flag	출력값이 (-)일 경우 SET, 이외는 Reset

#### AND - Bit-Wise AND



명령어 유형 : 출력

두 개의 입력을 논리곱(AND) 조건으로 연산 후, 출력합니다.

		입력	
		0	1
입력	0	0	0
	1	0	1

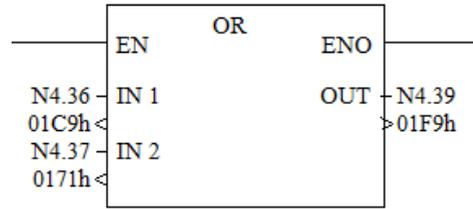
좌측의 예는

입력1 : 01C9H = 0000 0001 1100 1001

입력2 : 0171H = 0000 0001 0111 0001

출력 = 0000 0001 0100 0001 = 0141H-

### OR - Logical OR



명령어 유형 : 출력

두 개의 입력을 논리합(OR) 조건으로 연산 후, 출력합니다.

		입력	
		0	1
입력	0	0	1
	1	1	1

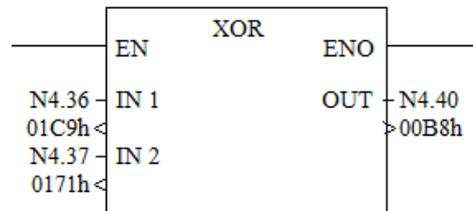
좌측의 예는

입력1 : 01C9H = 0000 0001 1100 1001

입력2 : 0171H = 0000 0001 0111 0001

출력 = 0000 0001 1111 1001 = 01F9H

### XOR - Logical OR



명령어 유형 : 출력

두 개의 입력을 배타적 논리합(XOR) 조건으로 연산 후, 출력합니다.

		입력	
		0	1
입력	0	0	1
	1	1	0

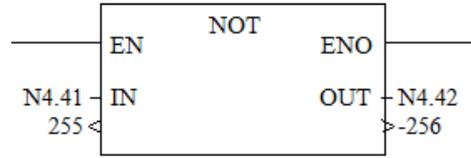
좌측의 예는

입력1 : 01C9H = 0000 0001 1100 1001

입력2 : 0171H = 0000 0001 0111 0001

출력 = 0000 0000 1011 1000 = 00B8H

**NOT - Logical NOT**



명령어 유형 : 출력

입력에 대한 반전 값을 출력합니다.

입력	0	1
	1	0

좌측의 예는

입력1 : 255 = 00FFH = 0000 0000 1111 1111

출력 : = 1111 1111 0000 0000 = -256

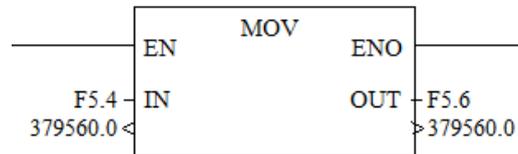
---

## 데이터 이동 명령어

데이터 이동 명령어는 2가지가 있습니다.

명령	설명
MOV	Move 입력 데이터를 지정된 출력 데이터 테이블로 이동 합니다.
MOVM	Masked Move 입력 데이터를 마스크 후, 지정된 출력 데이터 테이블로 이동 합니다.

### MOV - Move



명령어 유형 : 출력

입력 데이터를 출력에 지정된 데이터 테이블로 이동 합니다.

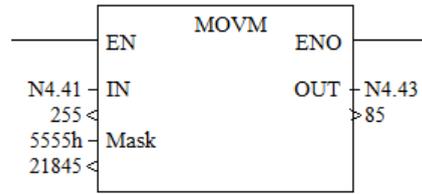
만일 입력 데이터의 크기와 출력 데이터의 크기가 다를 경우, 모든 기준은 출력에 맞추어 저장 됩니다.

SR주소	플래그	동작 설명
SR2.0.0	Carry Flag	항상 리셋
SR2.0.1	Overflow Flag	무한대, 오버플로우, NaN 조건일 경우 SET되고, 나머지는 리셋 됩니다.
SR2.0.2	Zero Flag	출력값이 0 일 경우 SET 이외는 Reset
SR2.0.3	Sign Flag	출력값이 (-)일 경우 SET, 이외는 Reset
SR2.5.0	Overflow Trap	오버플로우 플래그가 SET되면 본 플래그가 SET되고, 이외는 기존 마지막 값을 유지합니다.

입력 데이터가 상수 일 경우

- WORD : -32768 ~ 32767
- LONG WORD : -2,147,483,648 ~ 2,147,483,647의 범위 값을 사용 할 수 있습니다.

**MOVM - Masked Move**



명령어 유형 : 출력

입력 데이터1을 Mask 데이터와 AND (논리곱) 시킨 데이터를 출력에 지정된 데이터 테이블로 이동 합니다.

입력 데이터1, 입력 데이터2, 출력 데이터는 모두 같은 형태의 데이터 이어야 합니다.

좌측의 예는

입력1 : 00FFH = 0000 0000 1111 1111

MASK : 5555H = 0101 0101 0101 0101

출력 = 0000 0000 0101 0101 = 0055H = 85

SR주소	플래그	동작 설명
SR2.0.0	Carry Flag	항상 리셋
SR2.0.1	Overflow Flag	항상 리셋
SR2.0.2	Zero Flag	출력값이 0 일 경우 SET 이외는 Reset
SR2.0.3	Sign Flag	출력값이 (-)일 경우 SET, 이외는 Reset

데이터 범위는 아래와 같습니다.

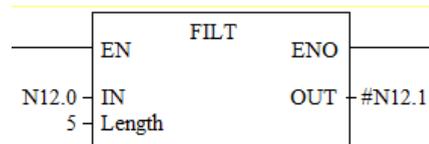
- WORD : -32768 ~ 32767
- LONG WORD : -2,147,483,648 ~ 2,147,483,647

## 데이터 테이블 명령어

데이터 이동 명령어는 2가지가 있습니다.

명령	설명
FILT	Fill Table 입력 데이터를 지정된 출력 데이터 테이블로 이동 합니다.
COPT	Copy Table 입력 데이터를 마스크 후, 지정된 출력 데이터 테이블로 이동 합니다.
SWAP	Swap Word
COPW	Copy Word

### FILT – Fill Table



명령어 유형 : 출력

입력 데이터1을 출력 대상 데이터 테이블에 Length 갯수 만큼 채웁니다.

좌측의 예제의 경우 N12.0 의 데이터를 N12.1 부터 5개를 채워 주는 예제입니다.

Address	Type	Value	Meaning	Description
N12.0	Integer	1		Source
N12.1	Integer	1		Destination 1
N12.2	Integer	1		Destination 2
N12.3	Integer	1		Destination 3
N12.4	Integer	1		Destination 4
N12.5	Integer	1		Destination 5
N12.6	Integer	0		

입력 데이터 범위는 아래와 같습니다.

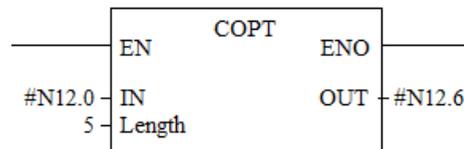
- WORD : -32768 ~ 32767
- LONG WORD : -2,147,483,648 ~ 2,147,483,647
- IEEE-754 32-bit 값

데이터 Length 범위

- WORD : 1 ~ 128
- LONG WORD : 1 ~ 64

3 WORD크기의 데이터 : 1 ~ 42

**COPT – Copy Table**



명령어 유형 : 출력

입력 데이터를 시작으로 Length 만큼의 데이터를 출력 데이터 테이블로 복사합니다.

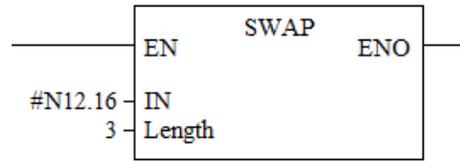
좌측의 예제는 N12.0 ~ N12.4 의 데이터를 N12.11 ~ N12.15 까지 복사하는 예제입니다.

Address	Type	Value	Meaning	Description
N12.6	Integer	1		Input #1
N12.7	Integer	2		Input #2
N12.8	Integer	3		Input #3
N12.9	Integer	4		Input #4
N12.10	Integer	5		Input #5
N12.11	Integer	1		Output #1
N12.12	Integer	2		Output #2
N12.13	Integer	3		Output #3
N12.14	Integer	4		Output #4
N12.15	Integer	5		Output #5
N12.16	Integer	0		

Length 길이 제한

- WORD : 1 ~ 128
- LONG WORD : 1 ~ 64
- 3 WORD : 1 ~ 42
- String : 1 ~ 3

### SWAP-Swap Word



명령어 유형 : 출력

입력 데이터를 시작으로 지정된 Length 개수만큼의 데이터 내에서, 각각을 구성하는 모든 워드의 각각 상위, 하위 바이트 데이터를 맞교환 합니다.

좌측 명령의 수행 결과를 아래 그림을 통해 확인 할 수 있습니다.

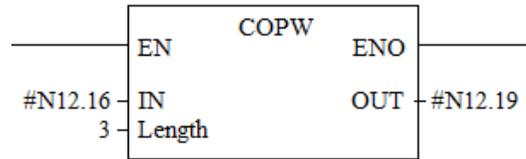
[수행 전]

Address	Type	Value	Meaning	Description
N12.16	Integer	1234 (h)		
N12.17	Integer	5678 (h)		
N12.18	Integer	9ABC (h)		
N12.19	Integer	0 (h)		
N12.20	Integer	0 (h)		

[수행 후]

Address	Type	Value	Meaning	Description
N12.16	Integer	3412 (h)		
N12.17	Integer	7856 (h)		
N12.18	Integer	BC9A (h)		
N12.19	Integer	0 (h)		
N12.20	Integer	0 (h)		

### COPW- Copy Word



명령어 유형 : 출력

입력 데이터를 기점으로 Length 만큼의 데이터를 출력 번지를 기점으로 Length만큼 복사합니다.

COPT 명령과 기능은 흡사하나 COPW명령은 입력 데이터 형과 출력 데이터 형을 서로 다르게 설정 가능 합니다.

- INTEGER → LONG WORD
- LONG WORD → FLOATING POINT
- LONG WORD → INTEGER
- INTEGER → SFR PTO 데이터 테이블
- Length는 128 이 최대 크기입니다.

만약 명령 수행 시 데이터 테이블의 크기를 초과하거나, 각 데이터 테이블의 최대/최소 값을 초과 할 경우, Fault 가 발생 합니다.

좌측 명령의 수행 결과를 아래 그림을 통해 확인 할 수 있습니다.

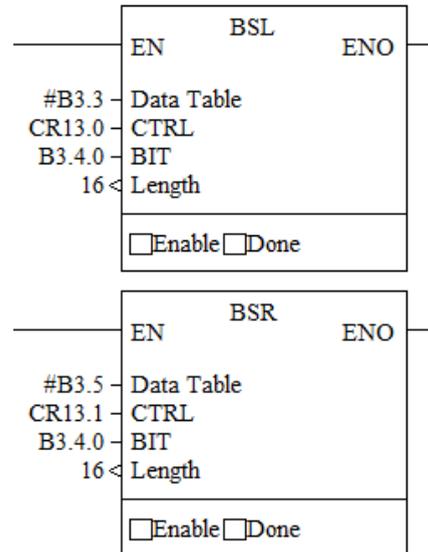
Address	Type	Value	Meaning	Description
N12.16	Integer	1234 (h)		Input #1
N12.17	Integer	5678 (h)		Input #2
N12.18	Integer	9ABC (h)		Input #3
N12.19	Integer	1234 (h)		Output #1
N12.20	Integer	5678 (h)		Output #2
N12.21	Integer	9ABC (h)		Output #3
N12.22	Integer	0		

## Shift 및 FIFO, LIFO 명령어

데이터 내부에서의 비트 Shift 명령어로는 아래와 같은 명령어가 있습니다.

명령	설명
BSL	Bit Shift Left 령 상태가 참(TRUE)일 때, 지정된 BIT값을 LSB에 삽입하고 전체 비트를 좌측으로 1 비트 이동 시킵니다.
BSR	Bit Shift Right 령 상태가 참(TRUE)일 때, 지정된 BIT값을 MSB에 삽입하고 전체 비트를 우측으로 1 비트 이동 시킵니다.
FIFOL	FIFO Queue Load 령 상태가 참(TRUE)일 때, 지정된 데이터를 Queue 로 지정된 데이터 테이블에 로드합니다.(PUSH)
FIFOU	FIFO Queue Unload 령 상태가 참(TRUE)일 때, 지정된 Queue 로부터 데이터를 Unload 하여 출력합니다.(POP)
LOFOL	LIFO Queue Load Stack 과 같은 저장 구조에서, 령 상태가 참(TRUE)일 때, 지정된 데이터를 Stack으로 지정된 데이터 테이블에 로드합니다.(PUSH)
LIFOU	LIFO Queue Unload Stack 과 같은 저장 구조에서, 령 상태가 참(TRUE)일 때, 지정된 Stack 로부터 데이터를 Unload 하여 출력합니다.(POP)

**BSL - Bit Shift Left    BSR - Bit Shift Right**



명령어 유형 : 출력

령 상태가 참(TRUE)일 때,

- BSL : 지정된 BIT값을 LSB에 삽입하고 데이터 테이블의 전체 비트를 좌측으로 1 비트 이동 시킵니다.
- BSR : 지정된 BIT값을 MSB에 삽입하고 데이터 테이블의 전체 비트를 우측으로 1 비트 이동 시킵니다.

데이터 설정

- Data Table : Shift 시킬 데이터 테이블
- CTRL : CR 데이터 테이블. BSL 명령 실행 시, 내부의 플래그등을 제어하기위한 목적.
- BIT : 삽입할 데이터를 가지고 있는 비트
- LENGTH : 데이터 테이블의 비트 길이

좌측의 예제의 결과를 아래 그림을 통해 확인 할 수 있습니다.

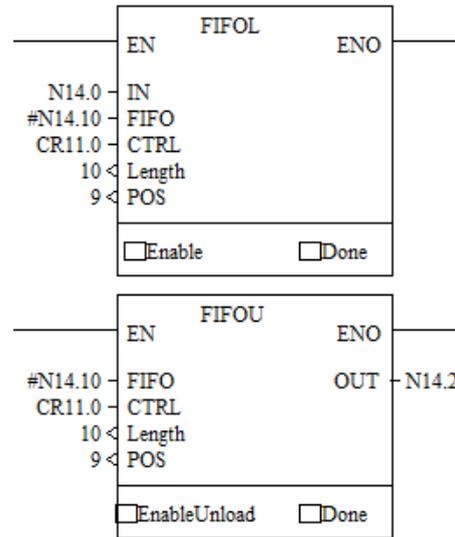
비트 쉬프트를 시킬 데이터 테이블로 B3.3 을 지정하고, 여기에 삽입할 비트 데이터를 B3.4.0 으로 지정 했을 때, B3.3 및 B3.5의 Bit Shift Left, Bit Shift Right 기능이 수행되어 비트 데이터가 Shift 된 것을 확인 할 수 있습니다.

Address	Type	Value	Meaning
B3.3	Integer	0000 1111 1111 1111	Bits
B3.4	Integer	0000 0000 0000 0001	Bits
B3.5	Integer	1111 1111 1111 0000	Bits
B3.6	Integer	0000 0000 0000 0000	Bits
B3.7	Integer	0000 0000 0000 0000	Bits

BSL또는 BSR명령에 의해 Shift가 이루어지게되면, MSB 또는 LSB로 마지막 데이터가 사라지게 되는데, 이 데이터는 CR 데이터 테이블의 Unload 플래그를 통해 확인 할 수 있습니다.

Address	Type	Value	Meaning
CR13.0	Control	{...}	
- Found	Bool	0	OFF
- Inhibit	Bool	0	OFF
- Unload	Bool	1	ON
- Error	Bool	0	OFF
- Empty	Bool	0	OFF
- Done	Bool	0	OFF
- EnableUnload	Bool	0	OFF
- Enable	Bool	0	OFF
- Length	Integer	16	
- Position	Integer	0	
CR13.1	Control	{...}	
- Found	Bool	0	OFF
- Inhibit	Bool	0	OFF
- Unload	Bool	1	ON
- Error	Bool	0	OFF
- Empty	Bool	0	OFF
- Done	Bool	0	OFF
- EnableUnload	Bool	0	OFF
- Enable	Bool	0	OFF
- Length	Integer	16	
- Position	Integer	0	

**FIFOL – FIFO QUEUE LOAD ,FIFOU - FIFO QUEUE UNLOAD**



명령어 유형 : 출력

령 상태가 참(TRUE)일 때,

- FIFOL : 령 상태가 참(TRUE)일 때, 지정된 데이터를 Queue 로 지정된 데이터 테이블에 로드합니다.(Push)
- FIFOU : 령 상태가 참(TRUE)일 때, 지정된 Queue 로부터 데이터를 Unload 하여 출력합니다.(POP)

**FIFOL데이터 설정**

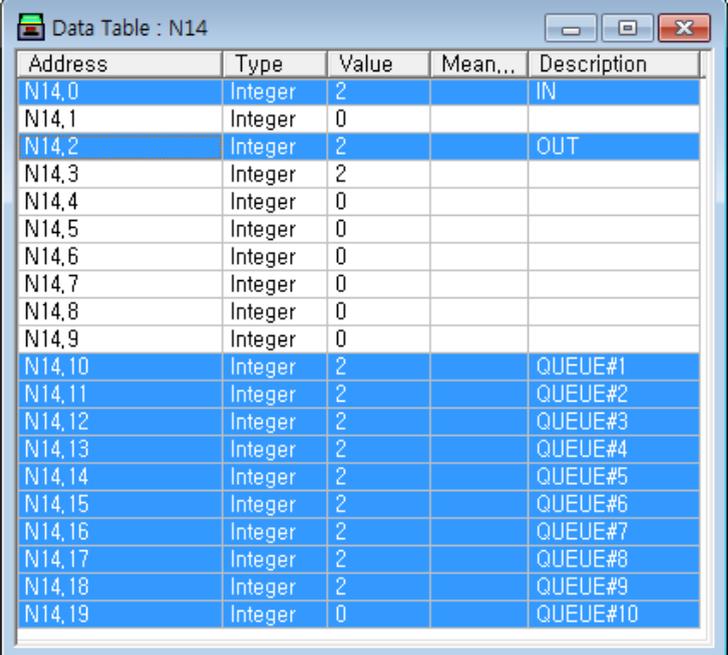
- IN : Queue에 삽입할 데이터 지정
- FIFO : QUEUE로 지정된 데이터 테이블
- CTRL : CR 데이터 테이블. FIFOL, FIFOU 명령 실행 시, 내부의 플래그등을 제어하기위한 목적.
- LENGTH : QUEUE로 지정된 데이터 테이블 크기
- POS : 가장 먼저 삽입한 데이터의 위치 (Pointer)

**FIFOU데이터 설정**

- FIFO : QUEUE로 지정된 데이터 테이블
- CTRL : CR 데이터 테이블. BSL 명령 실행 시, 내부의 플래그등을 제어하기위한 목적.
- LENGTH : 데이터 테이블의 비트 길이
- POS : 데이터를 추출 할 가장 먼저 저장된 데이터 위치 (Pointer)

- OUT : QUEUE로부터 데이터 언로드 후, 저장할 데이터 테이블

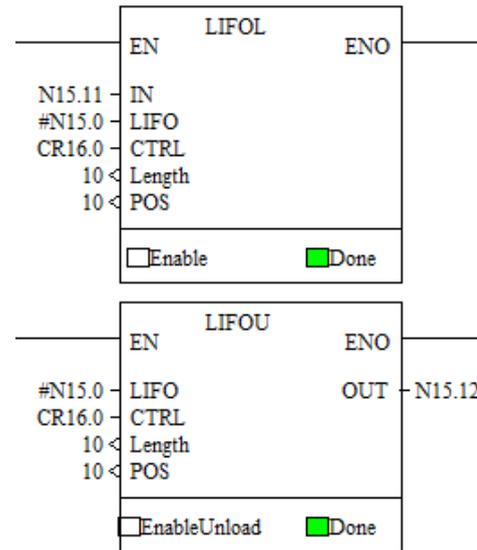
위 명령어 예제의 결과를 아래 그림을 통해 확인 할 수 있습니다.



Address	Type	Value	Mean...	Description
N14,0	Integer	2		IN
N14,1	Integer	0		
N14,2	Integer	2		OUT
N14,3	Integer	2		
N14,4	Integer	0		
N14,5	Integer	0		
N14,6	Integer	0		
N14,7	Integer	0		
N14,8	Integer	0		
N14,9	Integer	0		
N14,10	Integer	2		QUEUE#1
N14,11	Integer	2		QUEUE#2
N14,12	Integer	2		QUEUE#3
N14,13	Integer	2		QUEUE#4
N14,14	Integer	2		QUEUE#5
N14,15	Integer	2		QUEUE#6
N14,16	Integer	2		QUEUE#7
N14,17	Integer	2		QUEUE#8
N14,18	Integer	2		QUEUE#9
N14,19	Integer	0		QUEUE#10

N14.0 의 데이터가 IN , N14.10 ~ N14.19 가 QUEUE 영역, N14.2 는 OUT 으로 설정되어 FIFOL 명령 실행 시 N14.0 의 데이터가 N14.10 ~ N14.19의 저장 공간 중 (QUEUE) POS 에 나타나있는 위치에 저장이 되고, FIFO 명령 실행 시 N14.10 ~ N14.19에 설정된 영역 중 POS의 위치 (가장 먼저 로드된 데이터)에서 데이터를 빼내, OUT 저장소인 N14.2 에 저장이 됩니다.

## LIFOL – LIFO QUEUE LOAD ,LIFOU - LIFO QUEUE UNLOAD



명령어 유형 : 출력

령 상태가 참(TRUE)일 때,

- LIFOL : 령 상태가 참(TRUE)일 때, 지정된 데이터를 Stack 로 지정된 데이터 테이블에 로드합니다.(Push)
- LIFOU : 령 상태가 참(TRUE)일 때, 지정된 Stack 로부터 데이터를 Unload 하여 출력합니다.(POP)

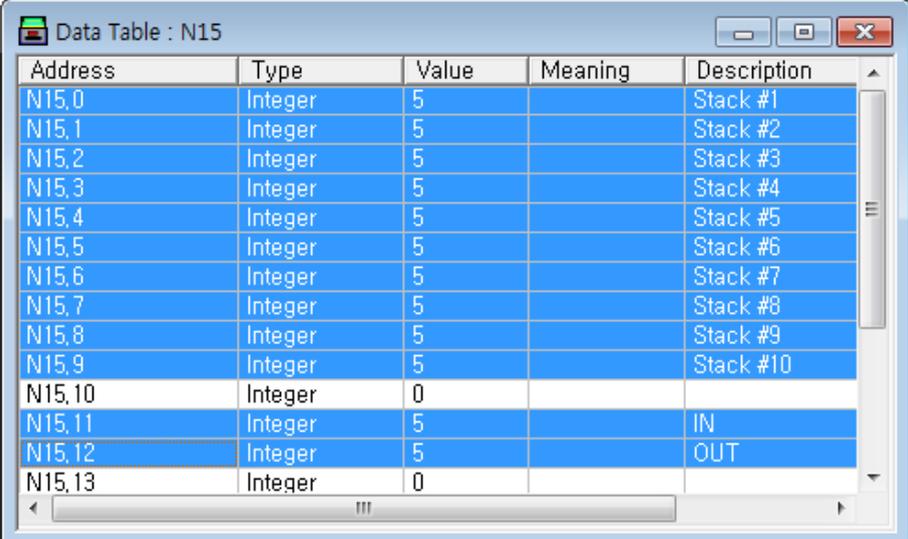
## LIFOL데이터 설정

- IN : Stack에 삽입할 데이터 지정
- LIFO : Stack으로 지정된 데이터 테이블
- CTRL : CR 데이터 테이블. FIFOL, LIFOU 명령 실행 시, 내부의 플래그등을 제어하기위한 목적.
- LENGTH : Stack로 지정된 데이터 테이블 크기
- POS : 데이터를 Load 포인터의 위치 (Stack Pointer)

## LIFOU데이터 설정

- OUT : Stack으로부터 빼낸 데이터 출력 위치
- LIFO : Stack으로 지정된 데이터 테이블
- CTRL : CR 데이터 테이블. FIFOL, LIFOU 명령 실행 시, 내부의 플래그등을 제어하기위한 목적.
- LENGTH : Stack로 지정된 데이터 테이블 크기
- POS : 데이터를 UnLoad 할 포인터의 위치 (Stack Pointer)

좌측의 예제의 결과를 아래 그림을 통해 확인 할 수 있습니다.



Address	Type	Value	Meaning	Description
N15.0	Integer	5		Stack #1
N15.1	Integer	5		Stack #2
N15.2	Integer	5		Stack #3
N15.3	Integer	5		Stack #4
N15.4	Integer	5		Stack #5
N15.5	Integer	5		Stack #6
N15.6	Integer	5		Stack #7
N15.7	Integer	5		Stack #8
N15.8	Integer	5		Stack #9
N15.9	Integer	5		Stack #10
N15.10	Integer	0		
N15.11	Integer	5		IN
N15.12	Integer	5		OUT
N15.13	Integer	0		

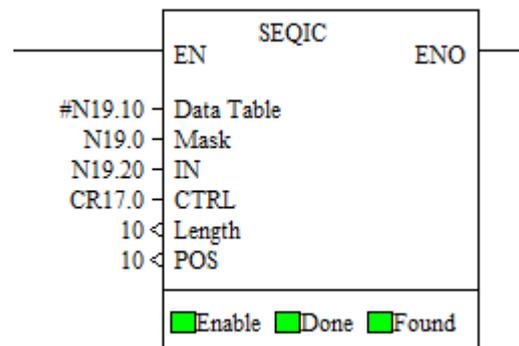
N15.11 의 데이터가 IN , N15.0 ~ N15.9 가 Stack 영역, N15.12 는 OUT 으로 설정되어 LIFOL 명령 실행 시 N15.11 의 데이터가 N15.0 ~ N15.9의 저장 공간 중 POS 에 나타나있는 위치에 저장 이 되고, LIFOU 명령 실행 시 N15.0 ~ N15.9에 설정된 영역 중 POS의 위치 (가장 늦게 로드 된 데이터)에서 데이터를 빼내, OUT 저장소인 N15.12 에 저장 이 됩니다.

## Sequencing (순차 제어) 명령어

데이터 내부에서의 비트 Shift 명령어로는 아래와 같은 명령어가 있습니다.

명령	설명
SEQIC	Sequencer Input Compare 령 상태가 참(TRUE)일 때, IN의 데이터와 마스크한 결과와 같은 값이, 데이터 테이블 안에 있는지를 검사하고, 있을 경우, CTRL 로 지정된 CR 데이터 테이블 안의 구성 요소인 Found 를 세팅합니다.
SEQO	Sequencer Output 령 상태가 참(TRUE)일 때, POS에 위치한 데이터 테이블의 값과 마스크 로 설정된 데이터를 AND 시킨 후, OUT출력에 설정한 데이터 테이블로 복사를 합니다. 지정된 개수 만큼 복사가 이루어지면, CTRL 로 지정된 CR 데이터 테이블 안의 구성 요소인 DONE 플래그를 세팅합니다.
SEQL	Sequencer Load 령 상태가 참(TRUE)일 때, IN 에 지정된 데이터 테이블의 값을 POS가 가르키는 데이터 테이블에 순차적으로 복사합니다. 지정된 개수 만큼(데이터 테이블의 크기) 복사가 이루어 지면, CTRL 로 지정된 CR 데이터 테이블 안의 구성 요소인 DONE 플래그를 세팅합니다.

### SEQIC- Sequencer Input Compare



명령어 유형 : 출력

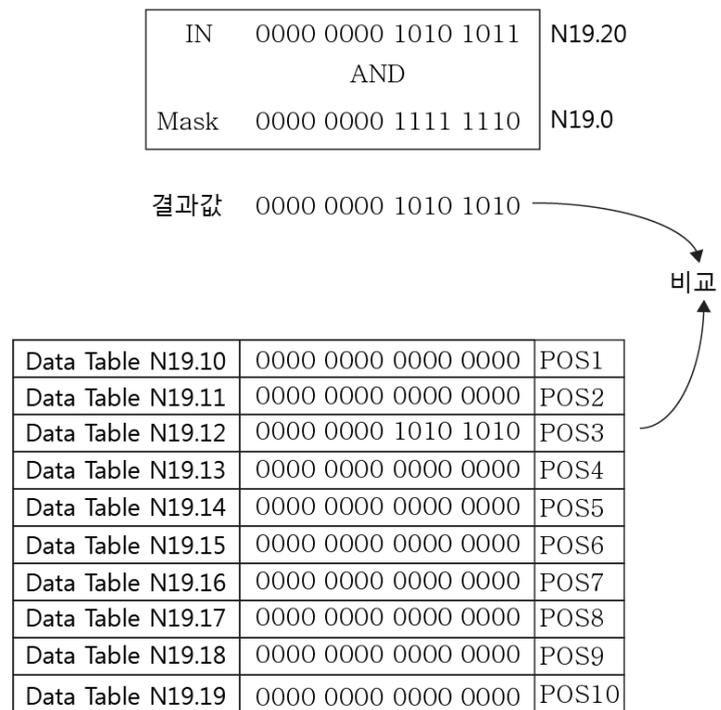
령 상태가 참(TRUE)일 때, IN의 데이터와 마스크 한 결과와 같은 값이 , 데이터 테이블 안에 있는지를 검사하고, 있을 경우, CTRL 로 지정된 CR 데이터 테이블 안의 구성 요소인 Found 플래그를 세팅합니다.

FOUND 플래그가 세팅되면 좌측의 그림과 같이 래더 상의 FOUND에 색깔이 변경 됩니다.

주의사항

데이터 테이블과 마스크, 비교 값은 모두 같은 형식의 데이터이어야 합니다. 즉, 데이터 테이블이 INTEGER (N) 일 경우, Mask 및 IN의 형식이 모두 INTEGER (N) 이어야 합니다.

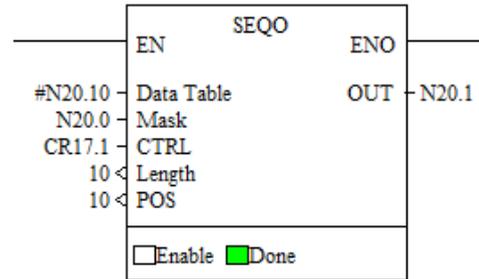
SEQIC 명령의 운영 메커니즘을 설명하면 아래와 같습니다..



N19.20 에 비교할 원본 데이터와 N19.0 에 마스크 데이터를 서로 AND 후, 그 결과 값이 N19.10 ~ N19.19 데이터 테이블에 있는지를 검사하게 됩니다. 위의 령이 TRUE 상태가 되면, 자동적으로 POS 값이 +1 씩 증가 되면서 N19.10 ~ N19.19 데이터 테이블의 데이터와 비교를 하게 됩니다.

위의 그림에서는 N19.12 에 같은 데이터가 위치하고 있는데, 이를 발견하게 되면, CR (Control) 데이터 테이블 내의 FOUND 플래그가 ON 으로 세팅 됩니다.

### SEQO- Sequencer Output

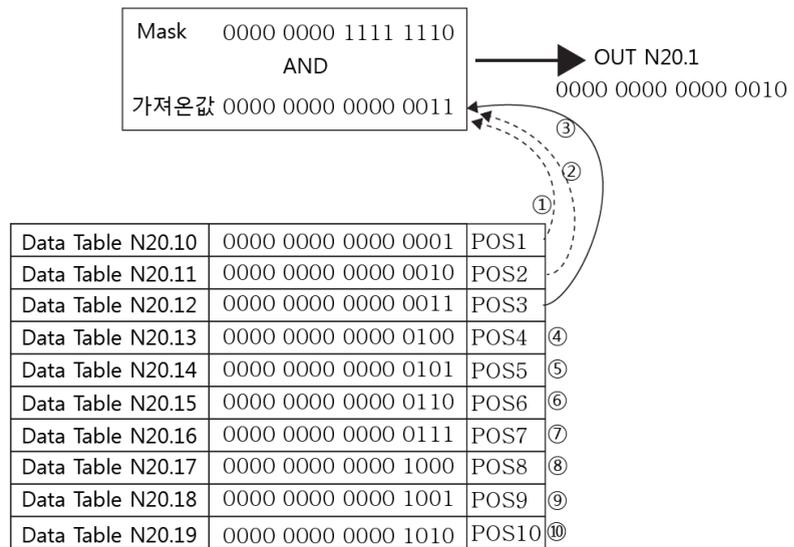


명령어 유형 : 출력

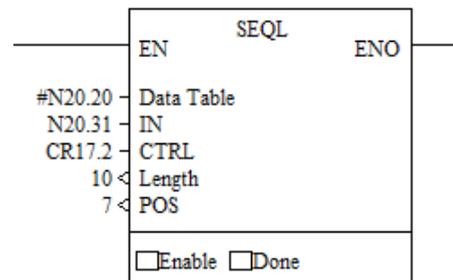
링 상태가 참(TRUE)일 때, POS에 위치한 데이터 테이블의 값과 마스크 로 설정된 데이터를 AND 시킨 후, OUT출력에 설정한 데이터 테이블로 복사를 합니다. 지정된 개수 만큼 복사가 이루어지면, CTRL 로 지정된 CR 데이터 테이블 안의 구성 요소인 DONE 플래그를 세팅합니다.

#### 주의사항

데이터 테이블과 마스크, 비교 값은 모두 같은 형식의 데이터이어야 합니다. 즉, 데이터 테이블이 INTEGER (N) 일 경우, Mask 및 IN의 형식이 모두 INTEGER (N) 이어야 합니다.



### SEQL- Sequencer Load

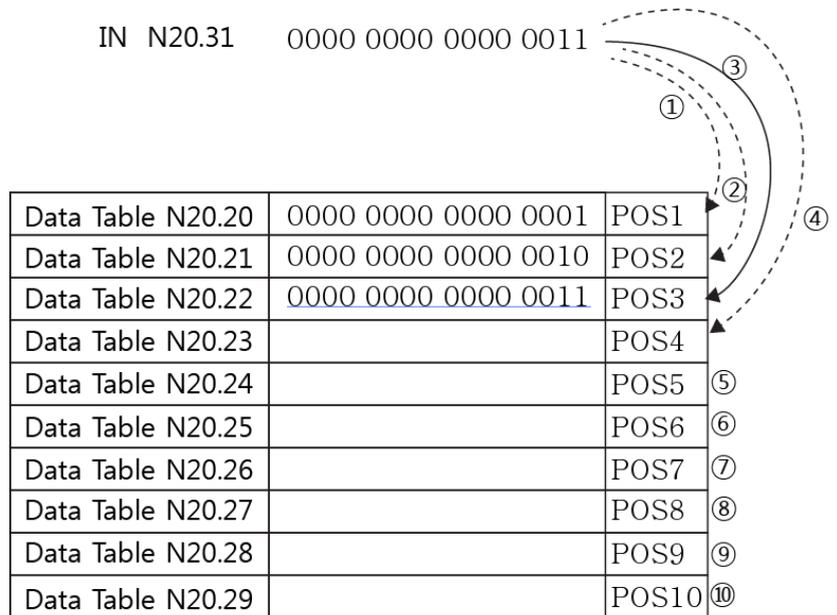


명령어 유형 : 출력

령 상태가 참(TRUE)일 때, IN 에 지정된 데이터 테이블의 값을 POS가 가르키는 데이터 테이블에 순차적으로 복사합니다. 지정된 개수 만큼(데이터 테이블의 크기) 복사가 이루어 지면, CTRL 로 지정된 CR 데이터 테이블 안의 구성 요소인 DONE 플래그를 세팅합니다.

주의사항

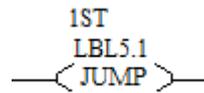
데이터 테이블과 마스크, 비교 값은 모두 같은 형식의 데이터이어야 합니다. 즉, 데이터 테이블이 INTEGER (N) 일 경우, Mask 및 IN의 형식이 모두 INTEGER (N) 이어야 합니다.



## 프로그램 제어 명령어

프로그램 제어 명령어로는 아래와 같은 명령어가 있습니다.

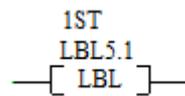
명령	설명
JUMP	Jump to Label 프로그램 실행 위치를 Label이 명시된 곳으로 옮깁니다.
LBL	Label JUMP 명령어의 사용을 위한, 위치를 표시합니다.
CALL	Call Subroutine 서브루틴을 호출합니다.
SBR	Subroutine Label 서브루틴의 시작을 나타내며, 반드시 RET와 한쌍을 이루어야 합니다.
RET	Return from Subroutine 서브루틴의 마지막에서 사용되며, 서브루틴을 호출한 다음 명령어로 복귀를 합니다.
SUSP	Suspend 프로그램의 디버깅 및 테스트를 위한 명령어로, 본 명령어가 실행되면 PLC는 IDLE 상태로 진입합니다.
ENDT	Temporary End 프로그램의 실행을 임시로 중지 시킵니다. 프로그램의 실행 도중에 ENDT를 만나게 될 경우, 이후의 명령은 더 이상 PLC에서 스캔 하지 않습니다.
END	End of a Ladder Program 래더 프로그램의 맨 마지막을 나타냅니다. 또한 래더 프로그램의 스캔시 맨 마지막이라는 것을 PLC로 알려주게 됩니다.
MCR	Master Control Relay MCR 명령 2개가 한쌍으로 동작을 하게 되며, 내부의 래더 프로그램을 제어 합니다.

**JUMP - Jump to Label**

명령어 유형 : 출력

프로그램 실행 위치를 Label이 명시된 곳으로 옮깁니다.

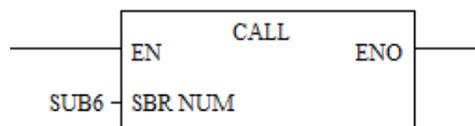
래더 프로그램에서 이동 할 수 있는 라벨의 범위는 0 ~ 999 까지 입니다.

**LBL - Label**

명령어 유형 : 입력

JUMP 명령어의 사용을 위한, 위치를 표시합니다.

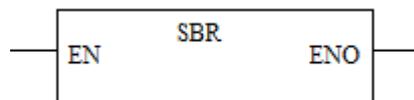
사용 할 수 있는 라벨의 범위는 0 ~ 999 까지 입니다.

**CALL - Call Subroutine**

명령어 유형 : 출력

별도의 서브루틴을 호출할 때 사용 합니다. 서브루틴의 NUM은 XGPC 상에서의 서브루틴의 번호입니다.

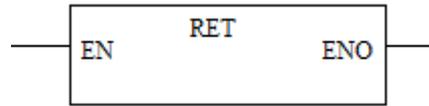
서브루틴에서는 RET 명령을 통해 중간에 복귀하거나, END를 통해 서브루틴을 수행 완료한 후에 복귀되는데, 복귀되는 위치는 CALL 명령이 발생한 다음 령입니다.

**SBR - Subroutine**

명령어 유형 : 입력

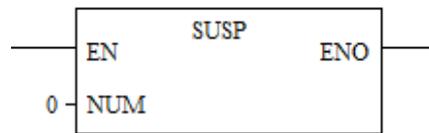
PLC에서 실제로는 사용하지 않는 코드를 가진 명령어로, XGPC에서 서브루틴의 첫 번째 행이라는 것을 나타냅니다.

**RET - Return from Subroutine**



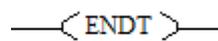
명령어 유형 : 출력  
 서브루틴의 마지막에서 사용되며, 서브루틴을 호출한 다음 명령어로 복귀를 합니다.

**SUSP - Suspend**



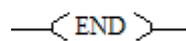
명령어 유형 : 출력  
 프로그램의 디버깅 및 테스트를 위한 명령어로, 본 명령어가 실행되면 PLC는 IDLE 상태로 진입하게 되어 현재 RUN 상태인 PLC가 동작 중지를 하게 되고, 모든 출력도 중단 됩니다.  
 본 명령어가 다수 개 사용될 경우의 구분을 위한, 의 NUM 의 범위는 -32768 ~ 32767 입니다.

**ENDT – Temporary End**



명령어 유형 : 출력  
 프로그램의 실행을 임시로 중지 시킵니다.  
 프로그램의 실행 도중에 ENDT를 만나게 될 경우, 이후의 명령은 더 이상 PLC에서 스캔 하지 않습니다.명령어 특성상, 반드시 수행 후 완료가 되어야 하는 인터럽트 서비스 루틴, HSC 관련 서브루틴 등에서는 사용 할 수 없습니다.

**END – End of ladder Program**



명령어 유형 : 출력  
 래더 프로그램의 맨 마지막을 나타냅니다. 또한 래더 프로그램의 스캔시 맨 마지막이라는 것을 PLC 로 알려주게 됩니다.

서브루틴, 인터럽트 루틴등에서 본 명령을 만나면 호출된 곳으로 복귀하는 역할도 수행합니다.

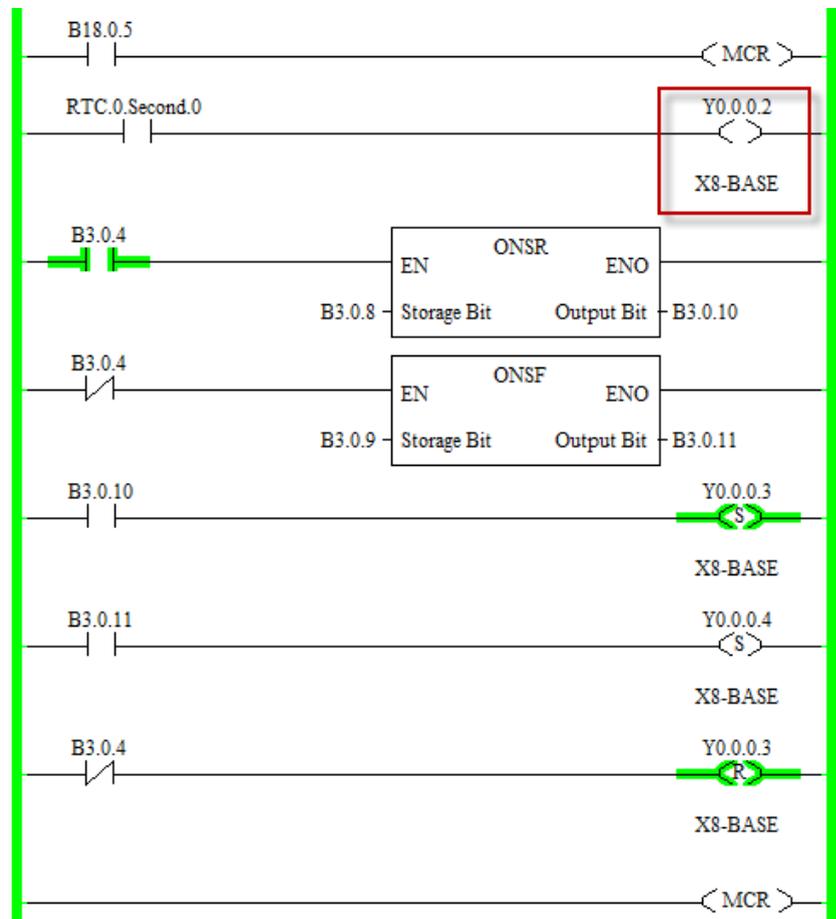
**MCR - Master Control Relay**

—〈MCR〉—

명령어 유형 : 출력

MCR 명령 2개가 한 쌍으로 동작을 하게 되며, 내부의 래더 프로그램을 제어 합니다. 본 명령어를 통해, 자기 유지가 되지 않은 출력등을 제어 하지 않음으로써, 스캔 타임을 줄일 수 있습니다.

아래 예제의 경우, 시작 MCR 령이 활성화 될 경우, 적색 네모칸의 Y0.0.02의 출력이 되지만, MCR 령이 비활성화 일 경우, 입력 단인 RTC.0.Second.0 이 참(TRUE) 상태 일지라도 Y0.0.02의 출력이 안나가게 됩니다.



주의 사항

- MCR 로 시작되서 반드시 MCR로 종료를 해야 합니다.
- MCR 안에 MCR이 중첩될 수 없습니다.

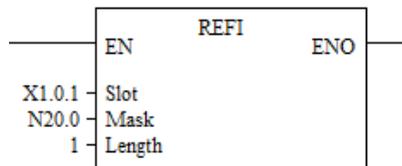
- 프로그램의 제어 로직이 올바르게 동작 되기 위해서는 MCR 영역으로 JUMP 하는 것을 추천하지 않습니다.

### 입력 / 출력 명령어

입출력 제어 명령어로는 아래와 같은 명령어가 있습니다.  
 본 명령어들을 통해서 PLC 내부의 스캔을 기다릴 필요 없이, IO 포트에 대한 즉각적인 상태 Update 가 가능합니다.

명령	설명
REFI	Embedded Input Refresh CPU 모듈에 내장된 Input 포트의 상태를 즉시 갱신합니다.
REFO	Embedded Output Refresh CPU 모듈에 내장된 Output 포트의 상태를 즉시 갱신합니다.
EOS	End Of Scan 전체 IO 읽기 쓰기 및 통신관련 상태를 갱신합니다.

#### REFI – Embedded Input Refresh



명령어 유형 : 출력

CPU 모듈에 내장된 Input 포트의 상태를 즉시 갱신합니다.  
 일반적으로 I/O 포트의 갱신은 래더 프로그램의 스캔 시 이루어지게 됩니다.

즉시, 입력 포트의 상태의 즉시 갱신이 필요할 경우, REFI 명령을 사용하면, 즉시 Input Port의 상태를 갱신 할 수 있습니다.

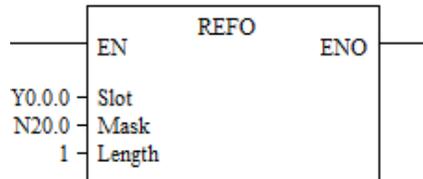
본 명령어는 CPU 모듈에 내장된 Input 포트에서만 동작 됩니다.

본 명령을 사용하기 위한 입력 사항은 아래와 같습니다.

- Slot : 입력 포트의 슬롯 위치를 지정 합니다. 좌측의 예제의 경우 X1 이라는 입력 변수에 슬롯 0, 1번 WORD 를 의미합니다.

- Mask : 입력된 데이터와의 마스크 조건입니다. 상수 또는 마스크 데이터가 저장되어 있는 데이터 테이블도 가능합니다.
- Length : 입력될 데이터의 워드 수입니다.

**REFO – Embedded Output Refresh**



명령어 유형 : 출력

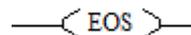
CPU 모듈에 내장된 Output 포트의 상태를 즉시 갱신합니다. 일반적으로 I/O 포트의 갱신은 래더 프로그램의 스캔 시 이루어지게 됩니다.

즉시, 출력 포트의 상태의 즉시 갱신이 필요할 경우 REFO 명령을 사용하면, 즉시 Output Port의 상태를 갱신 할 수 있습니다. 본 명령어는 CPU 모듈에 내장된 Output 포트에서만 동작 됩니다.

본 명령을 사용하기 위한 입력 사항은 아래와 같습니다.

- Slot : 출력 포트의 슬롯 위치를 지정 합니다. 좌측의 예제의 경우 Y0 이라는 출력 변수에 슬롯 0, 0번 WORD 를 의미합니다.
- Mask : 출력될 데이터와의 마스크 조건입니다. 상수 또는 마스크 데이터가 입력된 데이터 테이블도 가능합니다.
- Length : 출력될 데이터의 워드 수입니다.

**EOS – End of Scan**



명령어 유형 : 출력

전체 IO 읽기 쓰기 및 통신관련 상태를 갱신합니다. EOS 명령은 PLC 내부의 모든 포트에 대한 동시 실행을 하기 때문에 별도의 입력 데이터가 필요 없습니다. 령의 조건의 참(TRUE)이 되어 EOS 명령을 실행 하게 되면 모든 IO 포트 및 통신 관련 상태 갱신 후, EOS 다음부터 스캔이 재개 됩니다.

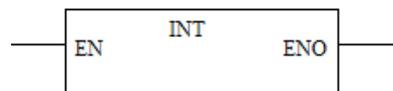
본 명령어는 상대적으로 속도가 느린 IO 포트에 대한 갱신을 하는 작업을 하게 되고 또 EOS 실행 후, 내부의 와치도그 타이머 나 스캔 타이머가 초기화 되므로, 시간에 영향을 받는 STI, HSC 루틴, TEI 루틴, 에러 처리 루틴 등에서는 사용 할 수 없습니다.

## 인터럽트 명령어

인터럽트 명령어로는 아래와 같은 명령어가 있습니다.

명령	설명
INT	Interrupt Subroutine 인터럽트의 시작을 알리는 라벨의 의미입니다.
PITS	PIT(Programmable Interrupt Timer) Start
INTD	Interrupt Disable 사용자 인터럽트를 비활성화 시킵니다.
INTE	Interrupt Enable 사용자 인터럽트를 활성화 시킵니다.
INTF	Interrupt Flush 시스템에서 실행 대기중인 각종 사용자 인터럽트를 시스템에서 제거합니다.

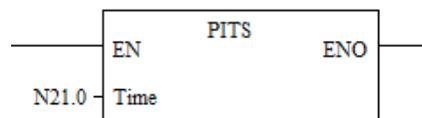
### INT – Interrupt Subroutine



명령어 유형 : 출력

앞에서 설명을 했던 SBR명령과 비슷한 개념으로, INT 명령은 인터럽트 루틴의 라벨과 같은 역할을 하는 명령어 입니다.

### PITS – Programmable Interrupt Timer Start



명령어 유형 : 출력

Programmable Interrupt Timer의 XGPC 내부에서의 설정은 SFR 내부의 "PIT" 부분을 통해 설정 할 수 있습니다.

Address	Type	Value	Meaning	Description
PIT_0		{...}		
<b>-&gt;&gt; Control</b>				
- LadderProgramNumber	Integer	0		
- UserInterruptEnable	Bool	1	ON	
- TimedInterruptEnabled	Bool	0	OFF	
- AutoStart	Bool	0	OFF	
- SetPointMsec	Integer	0		Set Point Msec (between
<b>-&gt;&gt; Status</b>				
- ErrorCode	Integer	0		
- UserInterruptExecuting	Bool	0	OFF	
- UserInterruptLost	Bool	0	OFF	
- UserInterruptPending	Bool	0	OFF	
- ErrorDetected	Bool	0	OFF	

### Control

- LadderProgramNumber
- UserInterruptEnable
- TimedInterruptEnabled
- AutoStart
- SetPointMsec

### Status

- ErrorCode
- UserInterruptExecuting
- UserInterruptLost
- UserInterruptPending
- ErrorDetected



## 9장. X8 CPU 특수 기능

X8 PLC로 특수한 기능을 사용하기 위해서 필요한 방법을 정리하여 설명합니다.

---

고속카운터 HSC기능 (HSC; High Speed Counter) .....9-2

확장 PTO(원점복귀) 기능.....9-24

PWM 출력 기능 .....9-40

## 고속카운터 HSC기능 (HSC; High Speed Counter)

기본 X8 Base Module PLC는 최대 100KHz의 주기를 갖는 4 채널의 고속 카운터와 1KHz의 주기를 갖는 2채널 서브고속 카운터 (이하 고속 카운터 또는HSC)를 갖고 있습니다. 0,1,2,3채널은 메인 고속 카운터이며, 나머지 4,5 채널은 서브 고속 카운터로 구성되어 있습니다. HSC 기능은 PLC 내장 I/O를 통해서만 사용되며, 확장 I/O에서는 지원되지 않습니다.

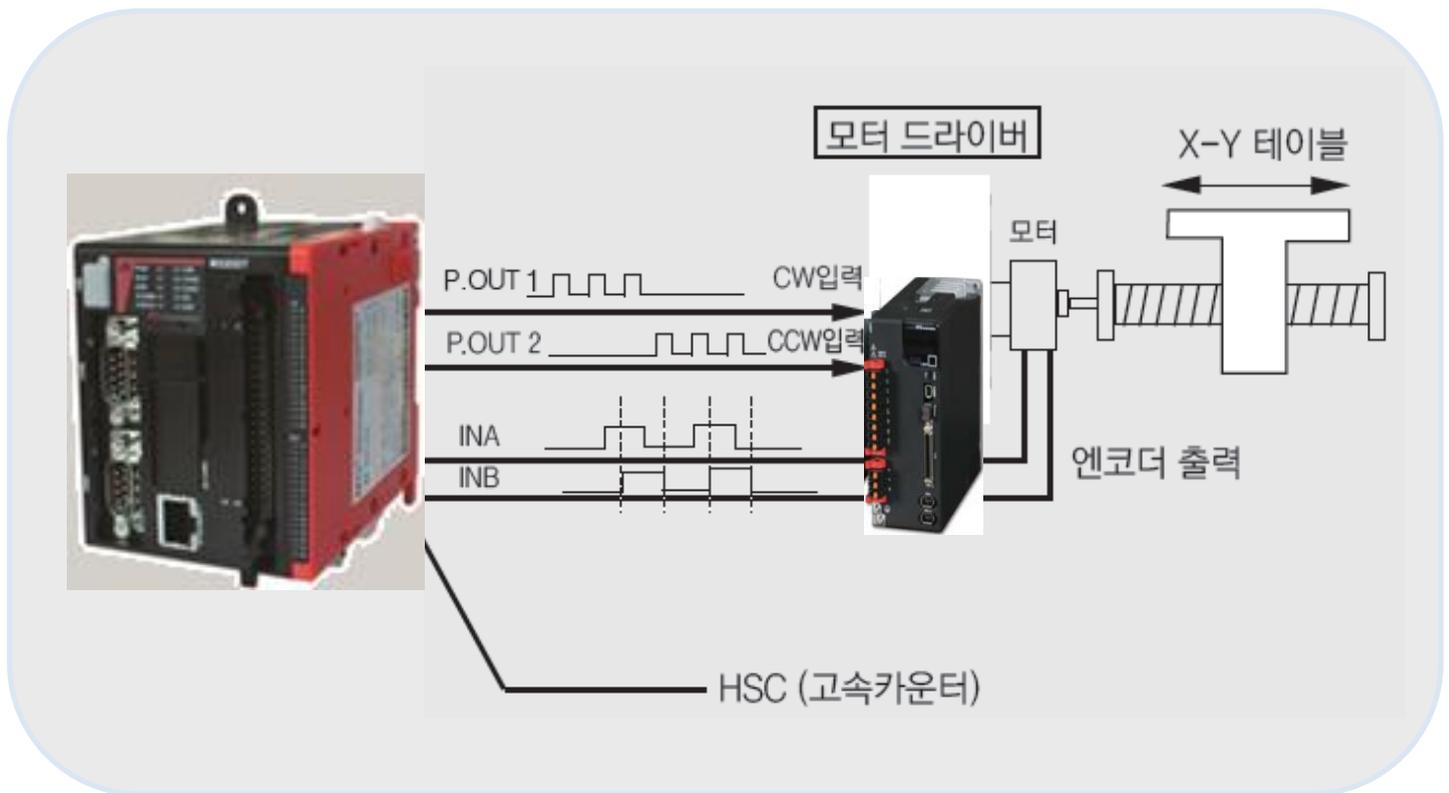
### 고속 카운터 사양

항목		규격
채널 수		6채널
Mode 수		10개의 Mode (각각 XGPC에서 설정 가능)
외부 출력	출력 점수	1채널 당 1점
	출력 형태	Open Collector 출력
데이터 처리		32비트(±2,147,483,647)
카운트		프로그램으로 지정 (Enable상태에서 카운트)

## 카운터 기능이란?

카운터 기능이란, 입력된 펄스 수를 카운트하여 경과 값에 반영하는 기능입니다. 또한 경과값에 데이터를 기록함으로써 오프셋 값을 설정할 수 있습니다.

본 X8 PLC에서 사용되는 고속 카운터는 고속 펄스 출력(PTO/PWM)과 조합하여 사용할 수 있습니다. 또한 사용자의 시스템과 병렬로 운영되기 때문에 운영이 쉬운 장점이 있습니다.



## 고속 카운터 MODE

고속 카운터 MODE는 프로그램(XGPC)에서 MODE 변수를 통해 10 종류의 운영방식을 설정할 수 있습니다. HSC.0의 서브 카운터는 HSC.1, HSC.2의 서브 카운터는 HSC.3, HSC.4의 서브 카운터는 HSC.5입니다. 아래의 표는 각각의 카운터에 대한 입력 포트를 보여줍니다.

모든 모드 포트별 결선 사항

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	A/C	B/D	Reset	Hold								
HSC.1			A/C	B/D								
HSC.2					A/C	B/D	Reset	Hold				
HSC.3							A/C	B/D				
HSC.4									A/C	B/D	Reset	Hold
HSC.5											A/C	B/D

HSC 동작 Mode

MODE	설명
0	Up Counter 고속 카운터의 Accumulator value가 High Preset에 도달했을 때, Accumulator value가 0으로 Reset 됩니다. Mode 0에서는 Low Preset value는 설정하지 않습니다.
1	Up Counter with External Reset/Hold 고속 카운터의 Accumulator value가 High Preset에 도달했을 때, Accumulator value가 0으로 Reset 됩니다. Mode 1에서는 Low Preset value는 설정하지 않습니다.
2	Up Counter with External Direction 고속 카운터의 Direction 설정이 가능한 모드입니다.
3	Up Counter with External Direction Reset/Hold 고속 카운터의 Direction 설정, Reset 및 Hold 신호 입력이 가능한 모드입니다.
4	Two Input Counter Up/Down 카운터용 2개의 입력이 가능한 모드입니다.
5	Two Input (Up and Down) Counter with External Reset/Hold Up/Down 카운터용 2개의 입력과 외부로부터의 Reset 및 Hold 신호 입력이 가능한 모드입니다.
6	Quadrature Counter A상, B상의 입력을 받아 처리하는 모드입니다.
7	Quadrature Counter with External Reset/Hold A상, B상의 입력, 외부로부터의 Reset 및 Hold 신호 입력이 가능한 모드입니다.
8	Quadrature X4 Counter A상, B상의 입력을 통하여 4체배의 처리가 가능한 모드입니다.
9	Quadrature X4 Counter with External Reset/Hold A상, B상의 입력을 통하여 4체배의 처리, 외부로부터의 Reset 및 Hold 신호 입력이 가능한 모드입니다.

## HSC Mode에 따른 입력 포트

## Mode 0; Up Counter

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	Count											
HSC.1			Count									
HSC.2					Count							
HSC.3							Count					
HSC.4									Count			
HSC.5											Count	

## Mode 1; Up Counter with External Reset/Hold

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	Count		Reset	Hold								
HSC.1												
HSC.2					Count		Reset	Hold				
HSC.3												
HSC.4									Count		Reset	Hold
HSC.5												

**Mode 2; Up Counter with External Direction**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	Count	Direction										
HSC.1			Count	Direction								
HSC.2					Count	Direction						
HSC.3							Count	Direction				
HSC.4									Count	Direction		
HSC.5											Count	Direction

**Mode 3; Counter with External Direction, Reset and Hold**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	Count	Direction	Reset	Hold								
HSC.1												
HSC.2					Count	Direction	Reset	Hold				
HSC.3												
HSC.4									Count	Direction	Reset	Hold
HSC.5												

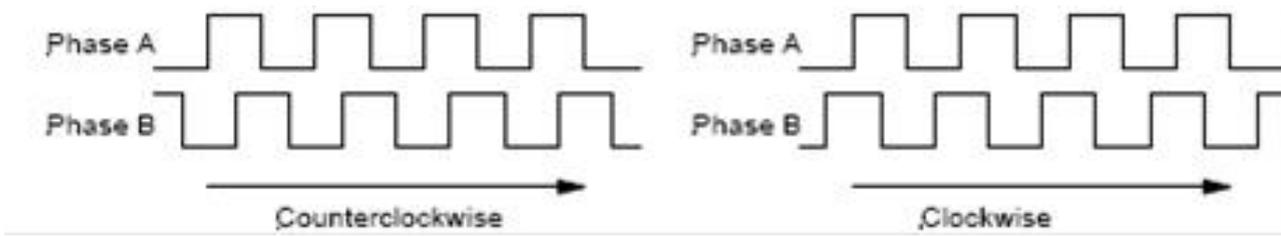
**Mode 4; Two Input Counter**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	Count Up	Count Down										
HSC.1			Count Up	Count Down								
HSC.2					Count Up	Count Down						
HSC.3							Count Up	Count Down				
HSC.4									Count Up	Count Down		
HSC.5											Count Up	Count Down

**Mode 5; Two Input Counter with External Reset and Hold**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	Count Up	Count Down	Reset	Hold								
HSC.1												
HSC.2					Count Up	Count Down	Reset	Hold				
HSC.3												
HSC.4									Count Up	Count Down	Reset	Hold
HSC.5												

**Mode 6; Quadrature Counter**



	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	A Phase Input	B Phase Input										
HSC.1			A Phase Input	B Phase Input								
HSC.2					A Phase Input	B Phase Input						
HSC.3							A Phase Input	B Phase Input				
HSC.4									A Phase Input	B Phase Input		
HSC.5											A Phase Input	B Phase Input

**Mode 7; Quadrature Counter with External Reset and Hold**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	A Phase Input	B Phase Input	Z Phase Reset	Hold								
HSC.1												
HSC.2					A Phase Input	B Phase Input	Z Phase Reset	Hold				
HSC.3												
HSC.4									A Phase Input	B Phase Input	Z Phase Reset	Hold
HSC.5												

**Mode 8; Quadrature X4 Counter**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	A Phase Input	B Phase Input										
HSC.1			A Phase Input	B Phase Input								
HSC.2					A Phase Input	B Phase Input						
HSC.3							A Phase Input	B Phase Input				
HSC.4									A Phase Input	B Phase Input		
HSC.5											A Phase Input	B Phase Input

**Mode 9; Quadrature X4 Counter with External Reset and Hold**

	X1.0.0	X1.0.1	X1.0.2	X1.0.3	X1.0.4	X1.0.5	X1.0.6	X1.0.7	X1.0.8	X1.0.9	X1.0.10	X1.0.11
HSC.0	A Phase Input	B Phase Input	Z Phase Reset	Hold								
HSC.1												
HSC.2					A Phase Input	B Phase Input	Z Phase Reset	Hold				
HSC.3												
HSC.4									A Phase Input	B Phase Input	Z Phase Reset	Hold
HSC.5												

## 고속 카운터 기능 설정 방법

### 사용 순서 1. CPU 모델별 결선

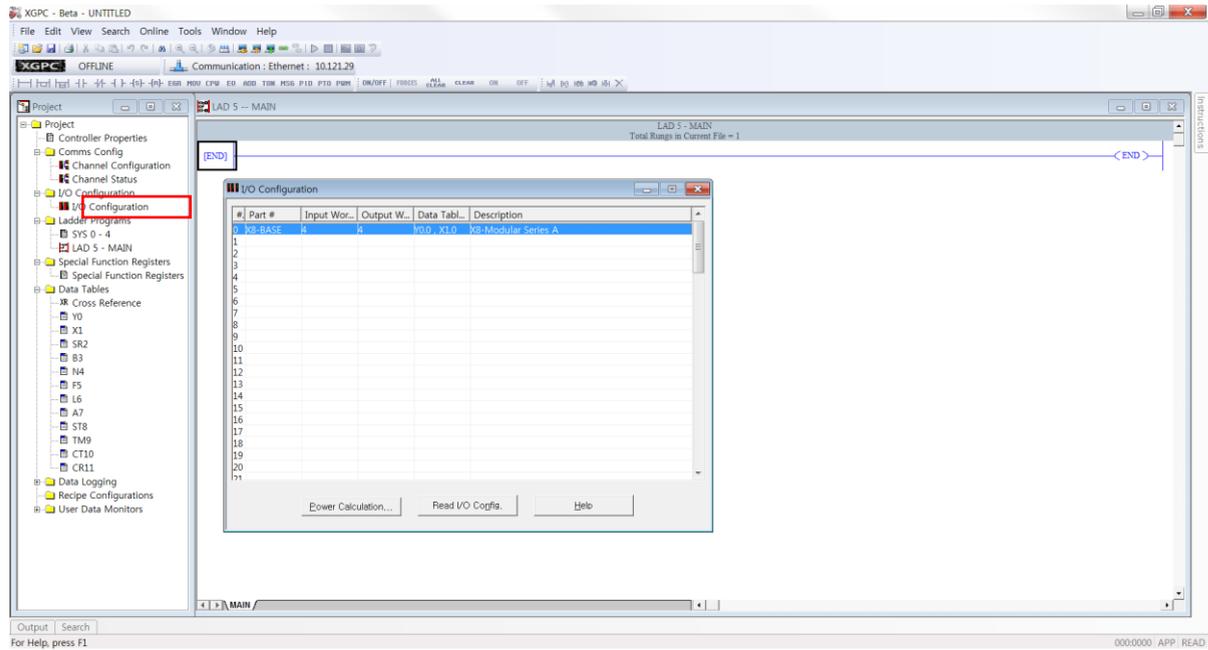
### 사용 순서 2. Input Filtering 설정

고속 카운터는 짧은 시간 내에 고속의 신호나 짧은 펄스 신호의 처리를 하기 때문에 PLC가 이러한 신호를 인식하는 것이 절대적인 요소입니다. X8 PLC Series는 이러한 신호 처리를 간단한 Input Filtering 기능을 이용하여 사용환경에 맞게 조절이 가능합니다. 이 기능은 고속 카운터 외에도 래칭 입력, 인터럽트 처리에 사용할 수 있습니다.

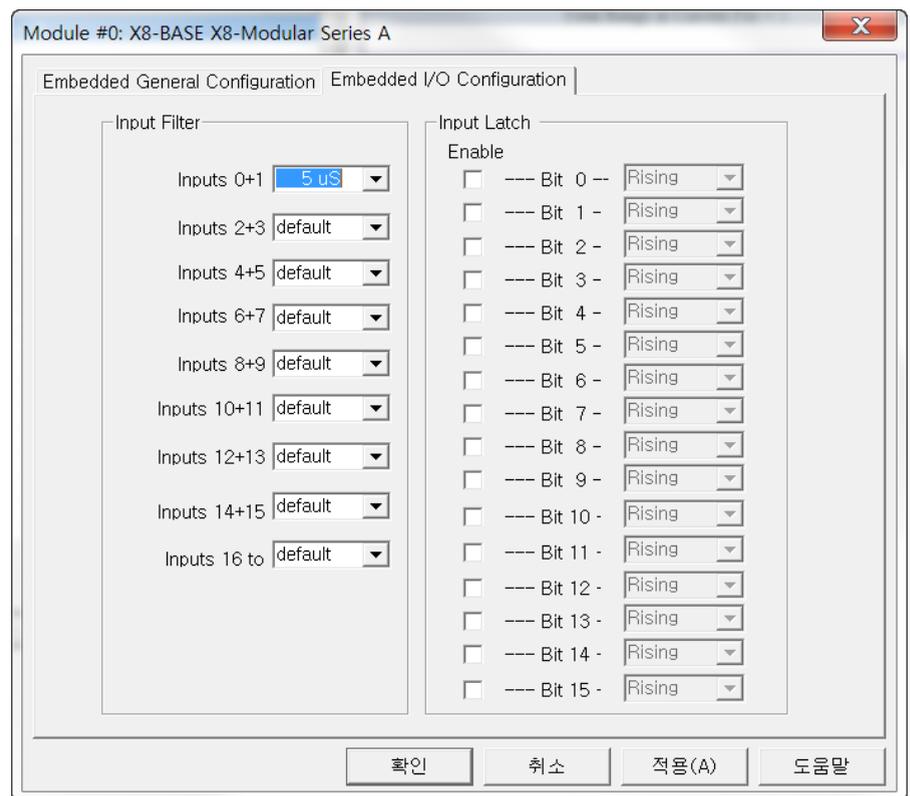
Input Filtering은 X8 PLC 소프트웨어인 XGPC를 통해 설정 가능합니다.

### 설정 방법

1. Project Tab을 선택합니다.
2. I/O설정 I/O Configuration을 더블 클릭 합니다.



3. 고속 카운터는 기본 내장 I/O에서 가능한 기능이므로, X8-BASE를 클릭합니다. Module Configuration 창이 뜨면 베이스 모듈을 사용하고 있는 CPU모델로 선정합니다. 내부 I/O설정 Tab을 선택합니다. 아래의 그림과 같이 Input Filter 및 Input Latch 방법 설정을 할 수 있습니다.



4. 고속 카운터의 입력 접점에 맞춰 Filter 속도를 지정할 수 있습니다. 즉, Inputs 0+1의 속도는 X1.0.0.0과 X1.0.0.1의 입력 Filter 속도이며, 이것을 5 $\mu$ s의 속도로 신호를 인식한다는 의미가 됩니다. Filtering의 최대값은 16ms 이고, 최소값은 5 $\mu$ s입니다. 5us라고 설정되면 100Khz의 펄스를 받아 들일 수 있습니다.

### 사용 순서 3. 변수 설정

고속 카운터의 변수는 XGPC 내부의 SFR(Special Function Register) 테이블을 통해 각종 변수를 설정 및 Monitoring 합니다.

다음은 HSC SFR에 대해 간단히 설명합니다.

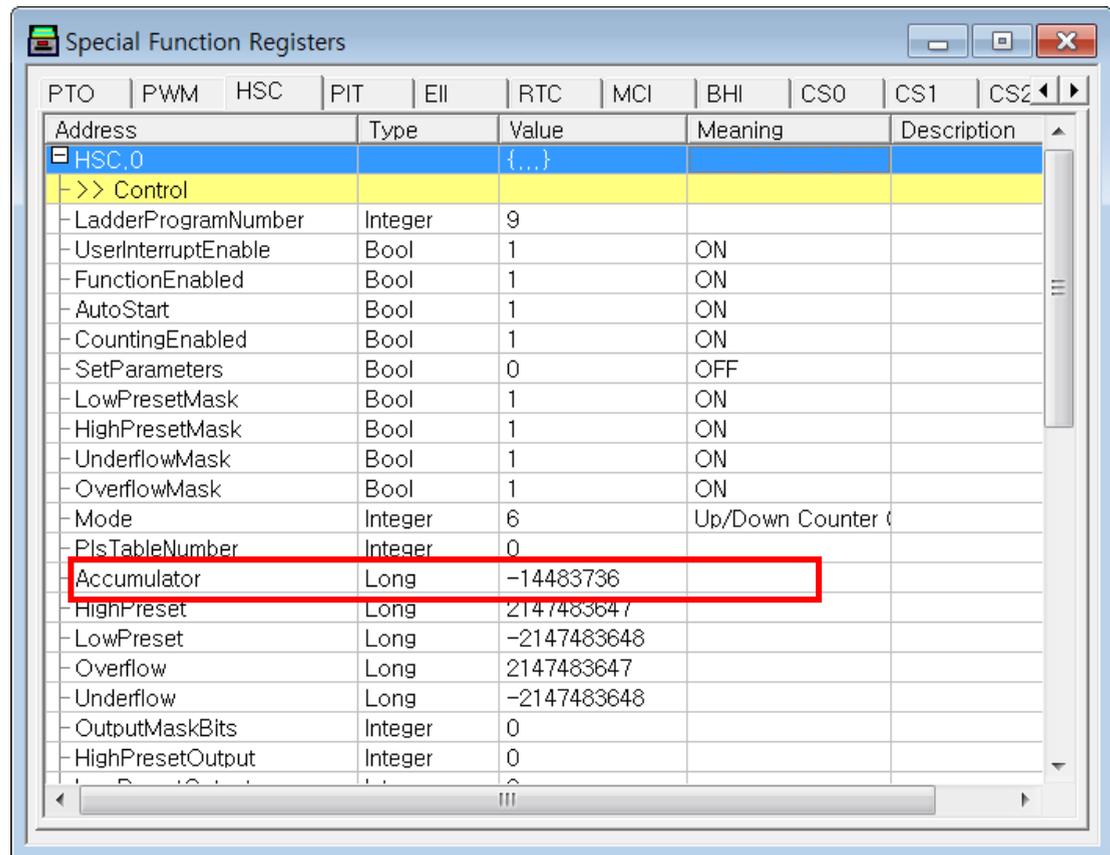
HSC SFR 구성원

Address	Data Format	HSC Mode	User Program Access	Explanation
Control				
LadderProgram Number	Word(INT)	0...9	Read Only	HSC Accumulator value가 High Preset, Low Preset, Overflow, Underflow에 도달했을 때, 실행되는 Subroutine 번호를 의미합니다. 본 데이터 값은 6~511까지의 범위를 가집니다.
UserInterrupt Enable	bit	0...9	Read/Write	HSC Accumulator value가 High Preset, Low Preset, Overflow, Underflow의 조건에 도달했을 때, 서브루틴의 수행여부를 설정합니다.
Function Enabled	bit	0...9	Read/Write	HSC Interrupt가 언제 Enable 상태가 되는지를 정의한 상태 및 제어 모두 처리 가능한 비트로, 내부적인 우선 순위에 의해 동작됩니다. 본 비트는 사용자 래더 프로그램에 의해 제어되거나, AutoStart 비트가 Enable 될 경우 자동적으로 설정이 됩니다.
AutoStart	bit	0...9	Read Only	사용자 프로그램의 일부로 저장이 되고, X8 PLC가 Run 또는 Test Mode에서 HSC 기능을 자동으로 시작되게 할 경우 본 비트를 사용합니다.
CountingEnabled	bit	0...9	Read/Write	HSC 기능의 사용 여부를 설정하는 비트입니다.
SetParameters	bit	0...9	Read/Write	새로운 값들을 HSC에 설정할 때 사용합니다.
LowPresetMask	bit	2...9	Read/Write	Low Preset Interrupt를 제어할 때 사용합니다.
HighPreset Mask	bit	0...9	Read/Write	High Preset Interrupt를 제어할 때 사용합니다.
UnderflowMask	bit	2...9	Read/Write	Underflow Interrupt를 제어할 때 사용합니다.
OverflowMask	bit	0...9	Read/Write	Overflow Interrupt를 제어할 때 사용합니다.
Mode	Word(INT)	0...9	Read Only	Mode 변수를 통해 10 종류의 HSC 운영 방식을 설정할 수 있습니다.
PlsTableNumber	Word(INT)	0...9	Read/Write	Programmable Limit Switch 기능을 사용할 때 설정합니다.
Accumulator	Long word(INT)	0...9	Read/Write	HSC에서 측정된 결과 값이 저장되어 있습니다.
HighPreset	Long word(INT)	0...9	Read/Write	HSC의 Accumulator에 대한 High Preset 설정 값을 의미합니다.
LowPreset	Long word(INT)	2...9	Read/Write	HSC의 Accumulator에 대한 Low Preset 설정 값을 의미합니다.
Overflow	Long word(INT)	0...9	Read/Write	HSC의 Accumulator value가 High Preset 또는 Overflow value를 초과했을 경우 ON 됩니다.
Underflow	Long word(INT)	2...9	Read/Write	HSC의 Accumulator value가 Low Preset 또는 Underflow value 이하로 감소했을 경우 ON 됩니다.
OutputMaskBits	Word(16 bit binary)	0...9	Read Only	PLC의 어떤 출력을 제어 할 것인지 정의하는 기능입니다.
HighPreset Output	Word(16 bit binary)	0...9	Read/Write	HSC의 Accumulator value가 High Preset에 도달했을 때의 출력 상태를 표시합니다. 그러나 실제적인 출력 값은 OutputMaskBits에 의해 결정됩니다.
LowPreset Output	Word(16 bit binary)	2...9	Read/Write	HSC의 Accumulator value가 Low Preset에 도달했을 때의 출력 상태를 표시합니다. 그러나 실제적인 출력 값은 OutputMaskBits에 의해 결정됩니다.
Status				

ErrorCode	Word(INT)	0...9	Read Only	HSC Error 발생 시, Error code를 저장합니다. (상세한 Error code는 Instruction Manual 참조)
UserInterrupt Executing	bit	0...9	Read Only	Accumulator 값이 High Preset / Low Preset에 도달하였을 때, 또는 Overflow / Underflow 조건이 되었을 경우 내부 프로세스를 시작할 때 'ON' 됩니다.
UserInterrupt Lost	bit	0...9	Read/Write	최대 2개의 Pending 된 인터럽트 유실 여부를 표시해 줍니다.
UserInterrupt Pending	bit	0...9	Read Only	어떤 조건들에 의해 현재 수행 중이던 인터럽트 기능이 정지 되었을 때 'ON' 됩니다.
ErrorDetected	bit	0...9	Read Only	HSC에서 Error 발생 시 'ON' 되는 상태 비트입니다.
LowPreset Interrupt	bit	2...9	Read/Write	HSC의 Accumulator value가 Low Preset에 도달하고, 이의 처리를 위한 인터럽트가 발생했을 때 'ON'이 되는 상태 플래그입니다.
HighPreset Interrupt	bit	0...9	Read/Write	HSC의 Accumulator value가 High Preset에 도달하고, 이의 처리를 위한 인터럽트가 발생했을 때 'ON'이 되는 상태 플래그입니다.
Underflow Interrupt	bit	2...9	Read/Write	HSC의 Accumulator value가 Underflow value에 도달하고, 이의 처리를 위한 인터럽트가 발생했을 때 'ON'이 되는 상태 플래그입니다.
Overflow Interrupt	bit	0...9	Read/Write	HSC의 Accumulator value가 Overflow value에 도달하고, 이의 처리를 위한 인터럽트가 발생했을 때 'ON'이 되는 상태 플래그입니다.
LowPreset Reached	bit	2...9	Read Only	HSC의 Accumulator value가 Low Preset value 보다 작거나 같을 경우 'ON' 되는 플래그입니다.
HighPreset Reached	bit	2...9	Read Only	HSC의 Accumulator value가 High Preset value 보다 크거나 같을 경우 'ON' 되는 플래그입니다.
CountDirection	bit	0...9	Read Only	HSC의 방향을 표시합니다. HSC Accumulator가 증가 할 경우 'ON'으로 설정되며, 감소 할 경우 'OFF'로 설정됩니다.
Underflow	bit	0...9	Read/Write	Accumulator의 Low Preset value를 지정합니다.
Overflow	bit	0...9	Read/Write	Accumulator의 High Preset value를 지정합니다.
ModeDone	bit	0 또는 1	Read/Write	HSC의 Mode가 0 또는 1로 설정되고, Accumulator가 증가되고 있을 경우 'ON'으로 설정됩니다.
CountDown	bit	2...9	Read Only	HSC의 Mode 2~9에서 양방향 카운터에 사용됩니다.
CountUp	bit	0...9	Read Only	HSC의 Mode 0~9에서 사용됩니다.

**사용 순서 4. Accumulator value 읽기**

SFR(Special Function Register) 테이블에서 HSC Tab 영역을 들어가 보면, HSC 관련 변수들을 확인할 수 있습니다. 사용 하는 HSC의 Accumulator value를 볼 수 있습니다.



**사용 순서 5. Accumulator value 입력**

사용하는 HSC의 Accumulator value를 입력하기 위해서는 SFR에서 Accumulator를 사용할 위치로 Drag 하여 옮기거나, 사용하고 있는 HSC.0.Accumulator (HSC0의 Accumulator일 경우)라고 입력을 하면 됩니다. Drag를 하여 옮길 때는 Drag 할 위치가 빨간색이 아닌 초록색으로 변하는 곳으로 옮겨지기 때문에 원하는 위치가 초록색이 되었을 때 Drop을 하면 됩니다. 키보드로 직접 입력할 때는 관련 내용들이 Down drop list로 나오기 때문에 원하는 내용을 선택하면 쉽게 사용 가능합니다. Accumulator value 외에 다른 SFR 변수들도 사용할 위치로 Drag 하여 옮기거나 직접 키보드로 입력하여 사용할 수 있습니다.

Special Function Registers

Address	Type	Value	Meaning	Description
<b>&gt;&gt; Control</b>				
-LadderProgramNumber	Integer	9		
-UserInterruptEnable	Bool	0	OFF	
-FunctionEnabled	Bool	1	ON	
-AutoStart	Bool	1	ON	
-CountingEnabled	Bool	1	ON	
-SetParameters	Bool	0	OFF	
-LowPresetMask	Bool	1	ON	
-HighPresetMask	Bool	1	ON	
-UnderflowMask	Bool	1	ON	
-OverflowMask	Bool	1	ON	
-Mode	Integer	6	Up/Down Counter (	
-PlsTableNumber	Integer	0		
-Accumulator	Long	-12		
-HighPreset	Long	2147483647		
-LowPreset	Long	-2147483648		
-Overflow	Long	2147483647		
-Underflow	Long	-2147483648		
-OutputMaskBits	Integer	0		
-HighPresetOutput	Integer	0		
-LowPresetOutput	Integer	0		
<b>&gt;&gt; Status</b>				
-ErrorCode	Integer	0		
-UserInterruptExecuting	Bool	0	OFF	

HSC.0.ACC

- Accumulator, Long
- AutoStart, Bool
- CountDirection, Bool
- CountDown, Bool
- CountingEnabled, Bool
- CountUp, Bool
- ErrorCode, Integer
- ErrorDetected, Bool
- FunctionEnabled, Bool

### 고속 출력(PTO / PWM) 기능

이 장에서는 기본 X8이 내장하고 있는 고속 출력 기능을 설명합니다. X8의 고속 출력(PTO; Pulse Train Output / PWM; Pulse Width Modulation)기능은 100KHz의 PTO와 40KHz의 PWM을 지원하며, 본 기능을 사용하여 간이 모션 제어 및 고속 펄스 출력 등의 응용을 할 수 있습니다. PTO 및 PWM 기능은 PLC 내장 I/O를 통해서만 사용되며, 확장 I/O에서는 지원되지 않습니다. 또한 X8 PLC Series 중 내장 I/O의 종류가 릴레이 출력(예, X8-M16DDR)일 경우 고속 수행 동작을 릴레이가 따라가지 못하기 때문에 사용하기 어렵습니다.

#### 고속 출력 사양

항 목	규 격
채널 수	6 채널
최대 출력 주파수	100KHz(PTO), 40KHz(PWM)

#### 고속 출력 기능

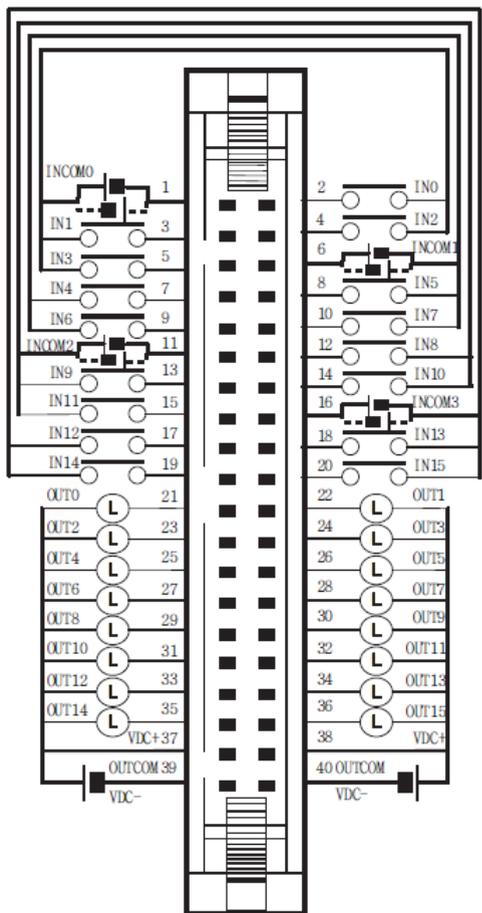
##### 1) PTO(Pulse Train Output)

펄스 출력 기능이란 1Hz~100kHz 내에서 임의의 주파수를 출력하는 기능입니다.

본 X8 PLC에서 사용되는 펄스 출력 기능은 최대 100kHz의 주파수를 출력하며, 이 출력 포트들은 기본적인 출력 포트로 사용하거나 각각의 포트 별로 설정 후 사용할 수 있습니다.

**X8-M32DDT에 연결되는 40 Pin map 예제**

	1	HSC COM0	2	0 AXIS A
	3	0 AXIS B	4	1 AXIS A
	5	1 AXIS B	6	HSC COM1
	7	2 AXIS A	8	2 AXIS B
	9	3 AXIS A	10	3 AXIS B
	11	COM2	12	HOME0 SENSOR
	13	HOME1 SENSOR	14	HOME2 SENSOR
	15	HOME3 SENSOR	16	COM3
	17	LIMIT0 SENSOR	18	LIMIT1 SENSOR
	19	LIMIT2 SENSOR	20	LIMIT3 SENSOR
Pulse	21	PTO0 OUT ①AXIS	22	PTO1 OUT ②AXIS
	23	PTO2 OUT ③AXIS	24	PTO3 OUT ④AXIS
	25	PTO4 OUT ⑤AXIS	26	PTO5 OUT ⑥AXIS
Sign	27	SIGN0 OUT 1AXIS	28	SIGN1 OUT 2AXIS
	29	SIGN2 OUT 3AXIS	30	SIGN3 OUT 4AXIS
	31	SIGN4 OUT 5AXIS	32	SIGN5 OUT 6AXIS
	33	SV0 ON OUT	34	SV1 ON OUT
	35	SV2 ON OUT	36	SV3 ON OUT
	37	VDC+	38	VDC+
	39	VDC-	40	VDC-



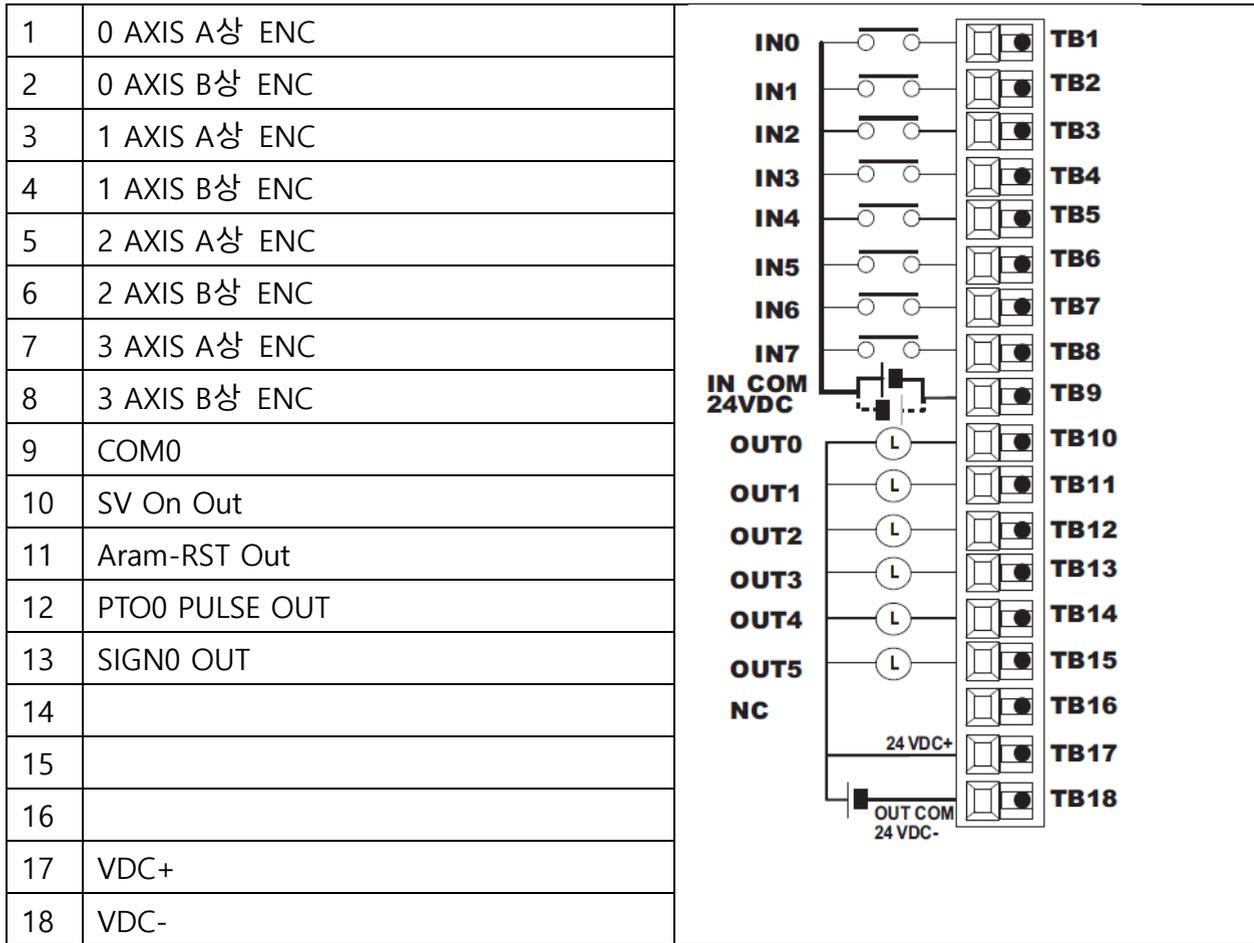
X8-M32DDT와 CSD5 Servo Drive를 연결할 경우, 다음과 같이 결선합니다.

X8-M32DDT		CSD5 Servo Drive	
SV On Out	21	3	SV ON IN
Aram-RST Out	22	7	Aram-RST IN
PTO OUT	23	12	24 PULS- IN
SIGN OUT	24	49	24 PULS+ IN
VDC+	37	14	SIGN- IN
VDC+	38	25	SIGN+ IN
VDC-	39	1	VDC+
VDC-	40	22	VDC-(SV GND)

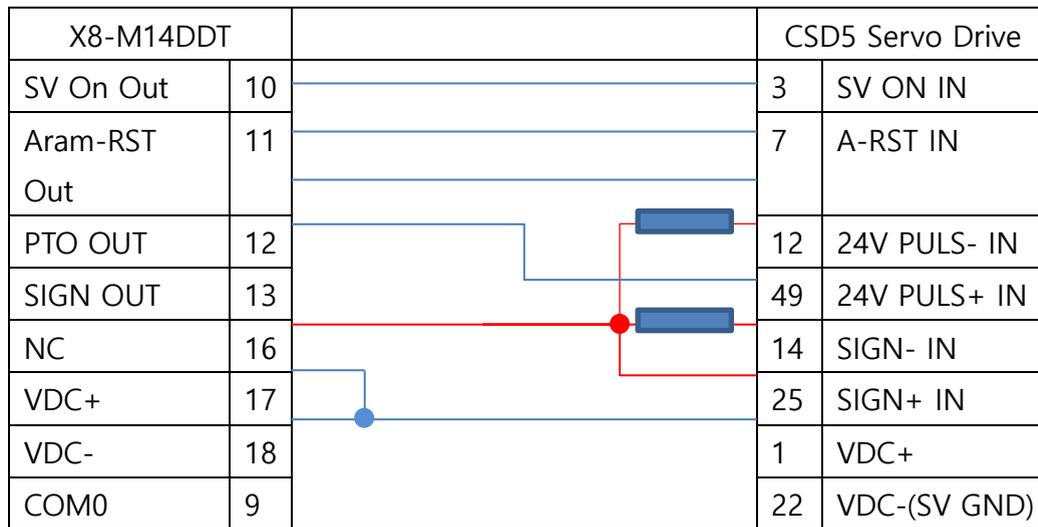
※ CSD5 Servo Drive에서 24V가 아닌 5V에 연결할 경우, 다음과 같이 결선합니다.

24 PULS+ IN	49	→	11	5V PULS+ IN
SIGN+ IN	25		13	5V SIGN+ IN

**X8-M14DDT에 연결되는 18점 Cable**



X8-M14DDT와 CSD5 Servo Drive를 연결할 경우, 다음과 같이 결선합니다.

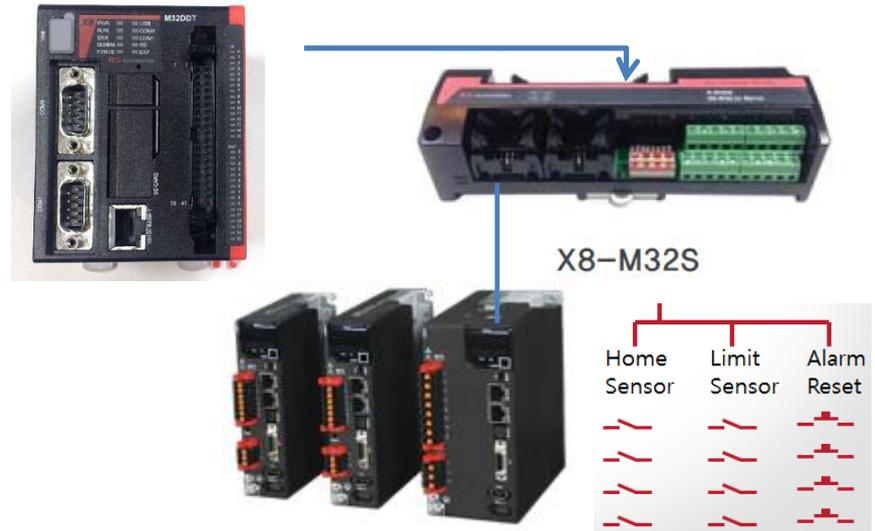


※ CSD5 Servo Drive에서 24V가 아닌 5V에 연결할 경우, 다음과 같이 결선합니다.

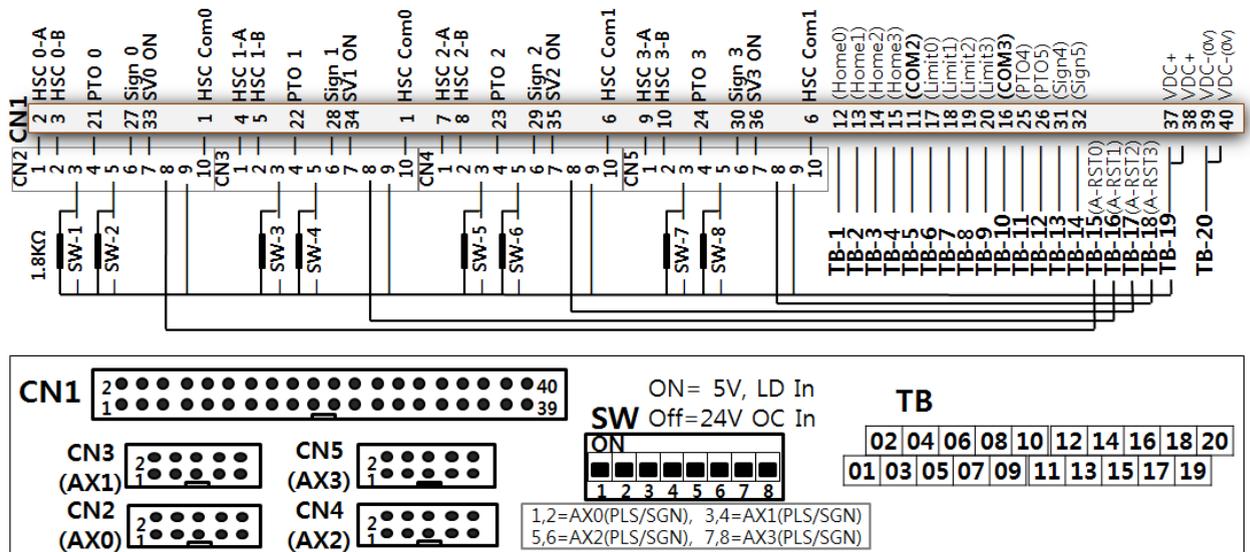
24 PULS+ IN	49	→	11	5V PULS+ IN
SIGN+ IN	25		13	5V SIGN+ IN

**X8-M32DDT HSC & PTO Interface Board**

X8-M32DDT Module로 HSC와 PTO를 이용할 경우 Interface Board를 이용하면 결선이 용이합니다. (CABLE : X8-MCBL1,MCBLSE1)



X8-M32S 결선도 사양



### 사용 순서 2. 변수 설정

고속 펄스 출력의 변수는 XGPC 내부의 특수 기능 레지스터 SFR(Special Function Register) 테이블을 통해 각종 변수를 설정 및 Monitoring 합니다.

다음은 PTO 특수 기능 레지스터 SFR에 대해 간단히 설명합니다.

어드레스	데이터 타입	값	의미	코멘트
PTO.0	...	...	...	...
<b>&gt;&gt; Control</b>				
- Output	Integer	-1		
- DirectionOutput	Integer	321		
- RampProfile	Bool	1	S-Curve	
- ControlStop	Bool	0	OFF	
- HardStop	Bool	0	OFF	
- AccelDecelIndependent	Bool	1	Separated Profile	
- AdjustAccelDecel	Bool	1	ON	
- JogPulseCW	Bool	0	OFF	
- JogContinuousCW	Bool	0	OFF	
- JogPulseCCW	Bool	0	OFF	
- JogContinuousCCW	Bool	0	OFF	
- JogAccelDecel	Bool	0	OFF	
- OutputInverse	Bool	0	OFF	
- DirectionInverse	Bool	0	OFF	
- Move Type	Bool	0	Relative Move	
- JogFrequencyOverride	Bool	0	OFF	
- JogFrequency	Unsigned Long	0		조그 주파수 (Hz)
- OutputFrequency	Unsigned Long	0		출력 주파수 (Hz)
- TotalOutputPulses	Long	0	Pulses for Relative Move	생성된 전체 출력 펄스
- AccelDecelPulses	Long	0	Data Table : L?.0	.AccelDecelIndependent0
- ExtendedPtoDataTblNumber	Integer	500	DataTable : L500	
<b>&gt;&gt; Status</b>				
- ErrorCode	Integer	0000 (h)		
- Done	Bool	0	OFF	
- DeceleratingStatus	Bool	0	OFF	
- RunStatus	Bool	0	OFF	
- AcceleratingStatus	Bool	0	OFF	
- IdleStatus	Bool	0	OFF	
- ErrorDetectedStatus	Bool	0	OFF	
- NormalOperationStatus	Bool	0	OFF	
- JogPulseStatus	Bool	0	OFF	
- JogContinuousStatus	Bool	0	OFF	
- EnableStatus	Bool	0	OFF	동작 상태 (형 상태에 따라)
- DirectionStatus	Bool	0	OFF	
- JogFrequencyOverridden	Bool	0	OFF	
- AccumulatedOutputOverflow	Bool	0	OFF	
- AccumulatedOutputUnderflow	Bool	0	OFF	
- AdjustAccelDecelStatus	Bool	0	OFF	
- OperationFrequency	Long	0		동작 주파수 상태 (Hz)
- OutputPulses	Long	0		
- AccumulatedOutputPulses	Long	0		
PTO.1	...	...	...	...

특수기능 레지스터 PTO설정

## PTO 특수기능 SFR 구성원

Address	Data Format	Range	User Program Access	Explanation
Control				
Output	Word(IN T)	0~5	Read/Write	PTO 및 PTO 명령에서 제어할 출력 포트를 정의합니다. 출력 포트 0은 Y0.0.0.0을 의미합니다.
Direction Output	Word(IN T)	0~15	Read/Write	PTO 및 PTO 명령에서 DirectionOutput 제어할 출력 포트를 정의합니다. 출력 포트 0은 Y0.0.0.0을 의미합니다.
Ramp Profile	bit	0 or 1	Read/Write	가속 또는 감속 시에 Output Frequency에 저장되어 있는 주파수를 어떻게 발생시킬 것인가 지정합니다. 0 : Trapezoid 사다리꼴 가감속 1 : S-Curve 가감속
Control Stop	bit	0 or 1	Read/Write	현재 실행 중인 PTO 동작을 감속구간 후 강제 정지 시킵니다.
Hard Stop	Bit	0 or 1	Read/Write	현재 실행 중인 PTO 동작을 긴급 강제 정지 시킵니다.
AccelDecel Pulse Independent	bit	0 or 1	Read/Write	가/감속시의 간격을 같게 할 것인지, 또는 서로 다른 간격으로 가/감속을 진행 시킬 지를 지정합니다. 가/감속 데이터를 독립적으로 사용하고자 할 때 LONG 데이터 파일번호를 입력하여 사용하면됩니다. 예) 500 -> L500.0:가속구간 L500.1:감속구간
Adjust AccelDecel Pulse	bit	0 or 1	Read/Write	가/감속시의 데이터가 등속구간보다 클 때 에러로 처리하지 않고 50%로 이하로 진행할 것 인지 지정합니다.
Jog Pulse CW	bit	0 or 1	Read/Write	CW(시계방향)방향으로 하나의 펄스를 발생하기 위해 사용되고, 펄스의 폭은 JogFrequency에 의해 정의됩니다. 조그 미세 조정시 사용됩니다.
Jog Continuous CW	bit	0 or 1	Read/Write	CW(시계방향)방향으로 PTO에서 연속적인 펄스를 발생하게 할 때 'ON'으로 설정하며, 주파수는 JogFrequency에서 설정합니다. 방향설정은 DirectionOutput포트에서 자동으로 처리 되어 on/off됩니다.
Jog Pulse CCW	bit	0 or 1	Read/Write	CCW(시계반대방향)방향으로 하나의 펄스를 발생하기 위해 사용되고, 펄스의 폭은 JogFrequency에 의해 정의됩니다. 조그 미세 조정시 사용됩니다.
Jog Continuous CCW	bit	0 or 1	Read/Write	CCW(시계반대 방향)방향으로 PTO에서 연속적인 펄스를 발생하게 할 때 'ON'으로 설정하며, 주파수는 JogFrequency에서 설정합니다. 방향설정은 DirectionOutput포트에서 자동으로 처리 되어 on/off됩니다.
Jog AccelDecel	bit	0 or 1	Read/Write	jog기동시 가감속 설정을 사용할 것 인지 지정합니다.
Output inverse	bit	0 or 1	Read/Write	PTO출력시 HIGH -> LOW로 시작할것인지 LOW -> HIGH로 시작할것인지 지정할 수 있습니다. 보통 1펄스씩 틀리게 동작할 때 서보설정 값을 바꾸지 않

				고 X8 CPU에서 설정할 수 있습니다.
Direction inverse	bit	0 or 1	Read/Write	PTO동작을 시켰을 때 방향이 반대로 움직일 때 서보에서 변경가능하나 이 비트를 ON 시키면 방향이 바뀝니다.
Move Type	bit	0 or 1	Read/Write	좌표데이터 처리를 상대 좌표 제어와 절대 좌표 제어를 선택할 수 있습니다. 의미에서 확인 가능합니다.
Jog Frequency Override	bit	0 or 1	Read/Write	JOG동작중 속도를 변경 하고자 할 때 사용하는 설정 비트입니다. 속도 제어가 가능해집니다.
Jog Frequency	Long word	0~100,000	Read/Write	Jog Phase(위상)의 주파수를 입력 합니다.
Output Frequency	Long word	0~100,000	Read/Write	PTO에서 RUN 상태에 발생 시킬 Pulse의 주파수 데이터를 입력합니다.
Total Output Pulses	Long word		Read/Write	PTO에서 발생시킬 총 Pulse 수를 지정합니다.
AccelDecel Pulses	Long word		Read/Write	가/감속 시에 얼마나 많은 Pulse를 발생 시킬 것인가를 지정합니다.(자세한 설명은 Instruction Manual을 참조)
Extended Pto DataTb Number	Integer		Read/Write	PTO 확장기능을 사용하고자 할 때 LONG 데이터 파일번호를 입력하여 사용하면됩니다. 확장기능이라고 함은 홈 기능을 이야기합니다. 예) 600 -> L600.0:확장 PTO기능설정구간 시작
Status				
Error Code	Word(IN T)	-2~7	Read Only	Error 발생 시, Error Code를 나타냅니다. 의미 부분에서 에러 데이터 확인 가능합니다.
Done	Bit	0 or 1	Read Only	PTO 동작이 완료 되었을 때, 'ON'으로 상태가 변경됩니다.
Decelerating Status	Bit	0 or 1	Read Only	현재 PTO 동작 모드가 감속 구간에 있을 때 'ON'으로 상태가 변경됩니다.
Run Status	Bit	0 or 1	Read Only	현재 PTO 동작 모드가 가속 구간을 지나 RUN Mode에 있을 때 'ON'으로 상태가 변경됩니다.
Accelerating Status	Bit	0 or 1	Read Only	현재 PTO 동작 모드가 가속 구간에 있을 때 'ON'으로 상태가 변경됩니다.
Idle Status	Bit	0 or 1	Read Only	현재 PTO 동작이 Idle 상태일 경우 'ON'으로 상태가 변경됩니다.
Error Detected Status	Bit	0 or 1	Read Only	PTO 동작 시 에러가 발생할 경우 'ON' 상태가 되고, 에러의 종류는 ErrorCode의 내용을 확인하여 원인을 알 수 있습니다.
Normal Operation Status	Bit	0 or 1	Read Only	PTO 상태가 정상일 경우 'ON' 상태가 되고, 정상 상태의 의미는 에러 발생 없이 가/감속, RUN, DONE 상태일 경우를 의미합니다.
Jog Pulse Status	Bit	0 or 1	Read Only	PTO 기능에서 Jog Pulse의 발생을 감지 했을 때 'ON'으로 상태가 변합니다.
Jog Continuout Status	Bit	0 or 1	Read Only	JogContinuous가 'ON'이 되어 연속적인 펄스를 발생 시킬 때 현재의 상태를 알려줍니다.

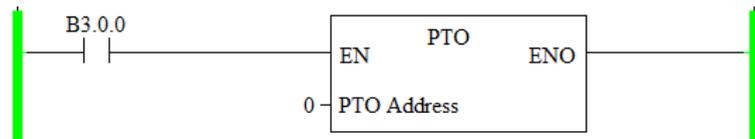
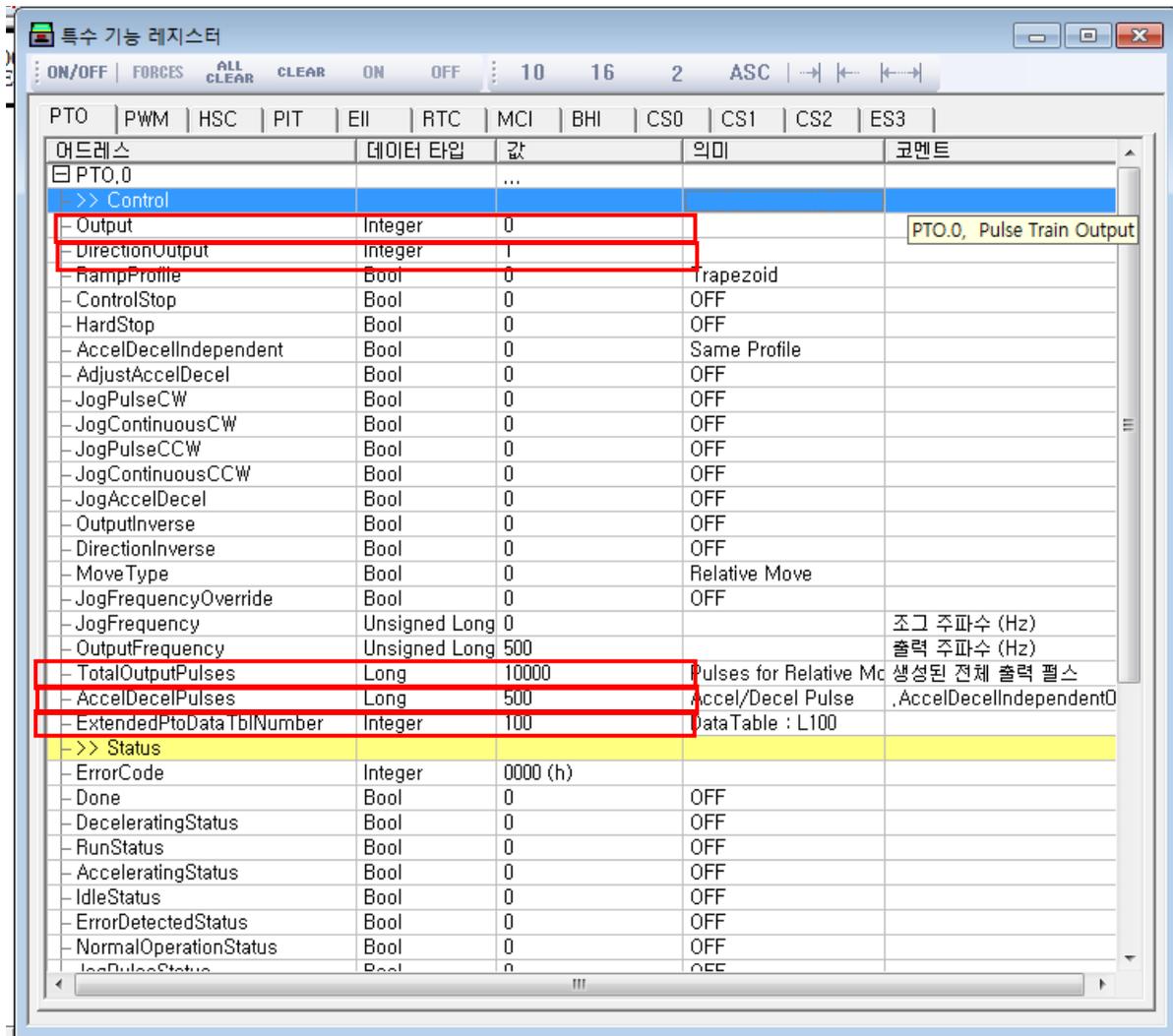
Enable Status	Bit	0 or 1	Read Only	PTO를 구동하는 기본 조건으로 래더 상에 PTO의 입력단인 EN=TRUE 상태가 되어야 PTO 구동이 가능합니다.
Direction Status	Bit	0 or 1	Read Only	PTO Direction를 구동하는 기본 조건으로 래더 상에 PTO의 입력단인 EN=TRUE 상태가 되어야 PTO 구동이 가능합니다.
JOG frequency Override	Bit	0 or 1	Read Only	OGfrequencyOverriden를 구동하는 기본 조건으로 래더 상에 PTO의 입력단인 EN=TRUE 상태가 되어야 PTO 구동이 가능합니다.
Operation Frequency Status	Long word	0~100,000	Read Only	현재 PTO에서 발생 중인 펄스 주파수 데이터입니다.
Output Pulses Produced	Long word	0~2,147,483,647	Read Only	현재 PTO에서 발생시킨 총 펄스 수 입니다.
Accumulated Output Pulses	Long word	0~2,147,483,647	Read Only	PTO에서 발생시킨 총 누적 펄스 수 입니다.

사용자는 SFR(Special Function Register)의 PTO 설정에서

- 1) 출력 포트 번호 설정(Output)
- 2) 출력방향 포트번호 설정(Directioin Output)
- 3) 발생시킬 펄스 수(Total Output Pulses)
- 4) 가/감속 간격(Accel Decel Pulses)
- 5) 실행간격(Output Frequency)

총 5개의 항목만을 입력 한 후, 래더 프로그램 상에 PTO 명령을 사용하면 간단히 펄스 출력 예제를 만들 수 있습니다.

- Output : 0(Y0.0.0.0), DirectionOutput : 1(Y0.0.0.1)
- OutputFregency : 500pps,
- TotalOutputPulses : 10000ps,
- AccelDecelPulses : 500 으로 설정



위와 같이 SFR에서 5개의 변수들을 지정한 값으로 설정 해 준 후, 래더 프로그램 상에서 B3.0.0이 'ON' 되면 SFR에서 설정한 값으로 PTO가 동작합니다.

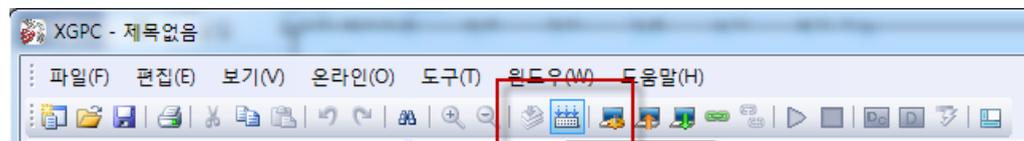
## 확장 PTO(원점복귀) 기능

X8 CPU에서 원점복귀 기능을 구현하기 위해서는 기존에는 래더로 구성하여 사용하였으나 FRN8부터 확장 PTO기능이 지원된다. 6가지 모드가 지원되면 필요한 원점복귀 모드를 사용하시면됩니다.

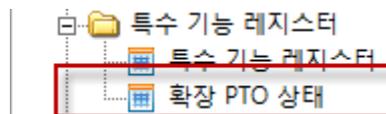
사용을 하려면 우선 특수기능레지스터의 PTO레지스터에서 Extended Pto Data Tb Number에 32bit Long파일번호를 넣고 컴파일을 시키면 파일번호의 레지스터가 생성되고 확장 PTO기능이 활성화 된다.

PTO	PWM	HSC	PIT	EII	RTC	MCI	BHI	CS0	CS1	CS2	ES3
주소	데이터 타입	값	의미	코멘트							
PTO.0											
->> Control											
- Output	Integer	-1									
- DirectionOutput	Integer	-1									
- RampProfile	Bool	0		Trapezoid							
- ControlStop	Bool	0		OFF							
- HardStop	Bool	0		OFF							
- AccelDecelIndependent	Bool	0		Same Profile							
- AdjustAccelDecel	Bool	0		OFF							
- JogPulseCW	Bool	0		OFF							
- JogContinuousCW	Bool	0		OFF							
- JogPulseCCW	Bool	0		OFF							
- JogContinuousCCW	Bool	0		OFF							
- JogAccelDecel	Bool	0		OFF							
- OutputInverse	Bool	0		OFF							
- DirectionInverse	Bool	0		OFF							
- MoveType	Bool	0		Relative Move							
- JogFrequencyOverride	Bool	0		OFF							
- JogFrequency	Unsigned Long	0			조그 주파수 (Hz)						
- OutputFrequency	Unsigned Long	0			출력 주파수 (Hz)						
- TotalOutputPulses	Long	0			Pulses for Relative Move						
- AccelDecelPulses	Long	0			Accel/Decel Pulse 생성된 전체 출력 펄스						
- ExtendedPtoDataTblNumber	Integer	100		Data Table : L100	.AccelDecelIndependent0						
->> Status											
- ErrorCode	Integer	0000 (h)									
- Done	Bool	0		OFF							
- DeceleratingStatus	Bool	0		OFF							
- RunStatus	Bool	0		OFF							
- AcceleratingStatus	Bool	0		OFF							
- IdleStatus	Bool	0		OFF							
- ErrorDetectedStatus	Bool	0		OFF							
- NormalOperationStatus	Bool	0		OFF							
- JogPulseStatus	Bool	0		OFF							

PTO 설정 레지스터



컴파일 아이콘



주소	데이터 타입	값	의미	코멘트
PTO.0 : L100				
>> PTO.0				
** 제어/상태 플래그				
L100.0	Long	0	PTO/Jog 모드	제어 플래그
L100.1	Long	0	-	상태 플래그
** 리미트 스위치 입력 설정				
L100.2.0	Bool	0	통전시 입력유효(A접점 - HW 리미트 스위치 타입	
L100.2.1	Bool	0	통전시 입력유효(A접점 원점 스위치 타입	
L100.2.2	Bool	0	통전시 입력유효(A접점 원점 근점 스위치 타입	
L100.2.3	Bool	0	통전시 입력유효(A접점 + HW 리미트 스위치 타입	
L100.3	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx] - HW 리미트 스위치 : 입력 접점 어드레스	
L100.4	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx] 원점 스위치 : 입력 접점 어드레스	
L100.5	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx] 원점 근점 스위치 : 입력 접점 어드레스	
L100.6	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx] + HW 리미트 스위치 : 입력 접점 어드레스	
** PTO/Jog 모드를 SW 리미트 설정				
L100.8	Long	0	pulses	- SW 리미트
L100.9	Long	0	pulses	+ SW 리미트
** 원점복귀 모드를 원점복귀 설정				
L100.10	Long	0	-	원점복귀 모드
L100.11	Long	0	-	목표 속도(pulses/second)
L100.12	Long	0	-	Creep 속도(pulses/second)
L100.13	Long	0	-	가속펄스수
L100.14	Long	0	-	감속펄스수
L100.15	Long	0	-	원점복귀 완료 후 , AccumulatedOutputPulses

확장PTO 설정창

## (2) 확장 PTO 출력 기능(원점복귀기능)

외부 센서(리미트+, -), 홈센서, 홈근접센서, Z상펄스 등으로 결선 후 설정창에 입력 포트를 설정 후 실행시키면 동작합니다. 주의 사항으로는 원점복귀기능과 PTO기능은 동시에 사용 할 수 없습니다. 제어 모드를 확인 하시고 동작시키시면 됩니다.

사용방법은 순서에 맞게 결선, 설정을 완료하시고 동작실행 하시면 됩니다.

### 사용 순서 1. 결선

- -리미트 센서(Negative Limit Switch : L(-))
- 홈 센서(Home Switch or Z-Pulse : H or Z)
- 홈 근접센서(Home Proximity Switch : HP)
- +리미트 센서(Positive Limit Switch : L(+))
- PTO결선은 동일하게 연결하시면 됩니다.

### 사용 순서 2. 변수 설정

- 원점복귀제어모드 및 센서 속도설정

확장 PTO기능은 총 7 가지의 중요 모드가 있습니다.

모드	설명
0	시계반대 방향(-방향) -리미트 서치 후 원점 근접센서 확인 후 속도변경 후 원점센서 시작점 원점 복귀
1	시계방향(+방향) +리미트 서치 후 원점 근접센서 확인 후 속도변경 후 원점센서 시작점 원점 복귀
2	시계반대 방향(-방향) -리미트 시작점 원점 복귀
3	시계방향(+방향) +리미트 서치 시작점 원점 복귀
4	시계반대 방향(-방향) -리미트 서치 후 원점센서 끝지점 원점 복귀
5	시계방향(+방향) +리미트 서치 후 원점센서 끝지점 원점 복귀
6	현재 위치 원점

## 모드별 센서 지원사항

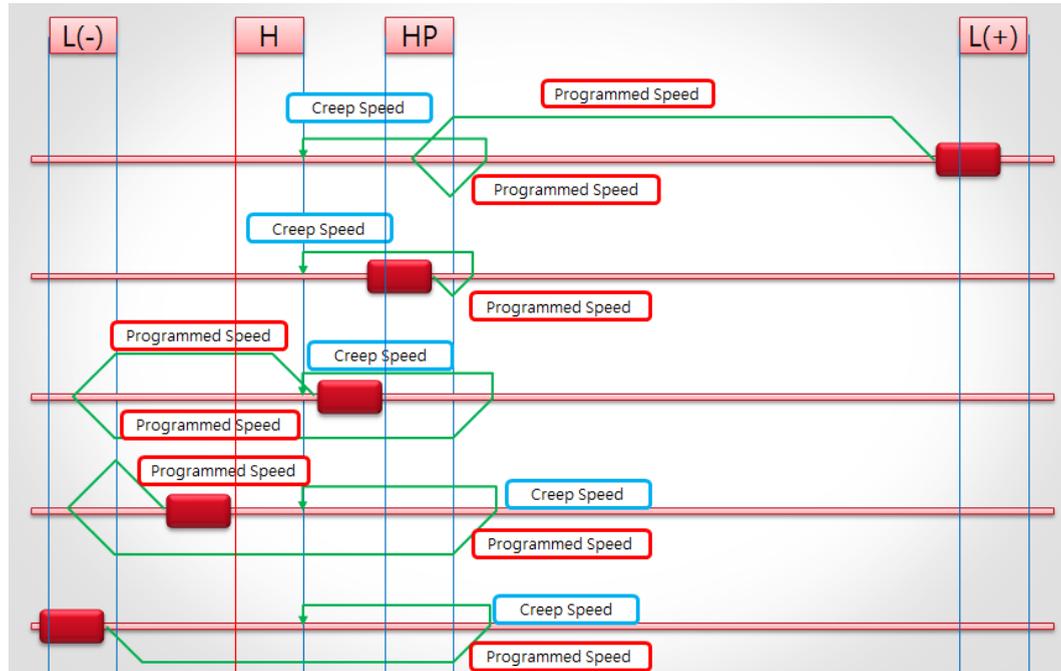
모드	-리미트센서	홈센서(Z상포함)	홈근접센서	+리미트센서
0	○	○	○	▲
1	▲	○	○	○
2	○	X	X	▲
3	▲	X	X	○
4	○	○	X	▲
5	▲	○	X	○
6	X	X	X	X

○ : 사용함 ▲ : 사용은 하나 꼭필요치않음 X : 사용하지않음

모드별 상세 설명

- 0번모드(1번모드:반대방향)

시계반대 방향(-방향) -리미트 서치 후 원점 근접센서 확인 후 속도변경 후 원점센서 시작점 원점 복귀



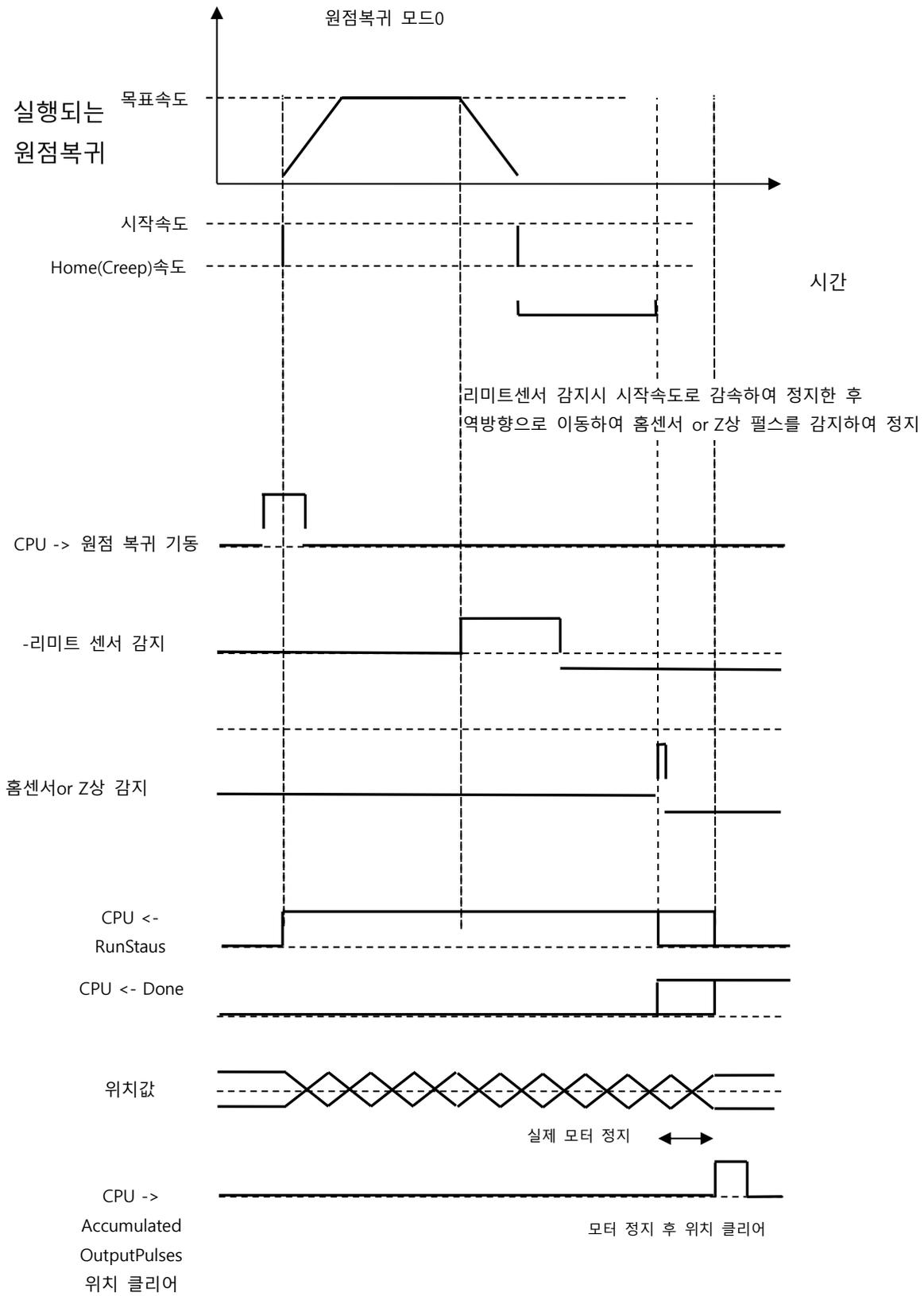
모드	-리미트센서 (L(-))	홈센서(Z상포함) (H)	홈근접센서 (HP)	+리미트센서 (L+)
0	○	○	○	▲
1	▲	○	○	○

리미트 센서가 감지되면 정지한 후 역방향으로 저속 회전하여 Z상이 감지되면 정지하는 모드입니다.

최초에 리미트 센서가 감지될 때의 정지는 설정데이터에서 Stop Action Mode (Limit Action)의 설정에 따라 감속하여 정지하거나 즉시 정지의 동작을 합니다.

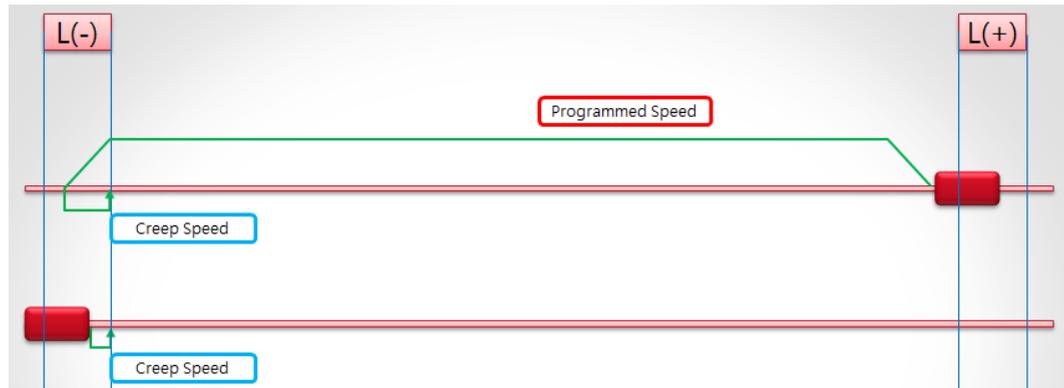
**\*1번 모드는 +방향으로 반대로 동작합니다.**

■ 동작 구간별 플래그 동작상태



**2번모드(3번모드:반대방향)**

시계반대 방향(-방향) -리미트 서치 후 시작점 원점 복귀



모드	-리미트센서 (L(-))	홈센서(Z상포함) (H)	홈근접센서 (HP)	+리미트센서 (L+)
2	○	x	x	▲
3	▲	x	x	○

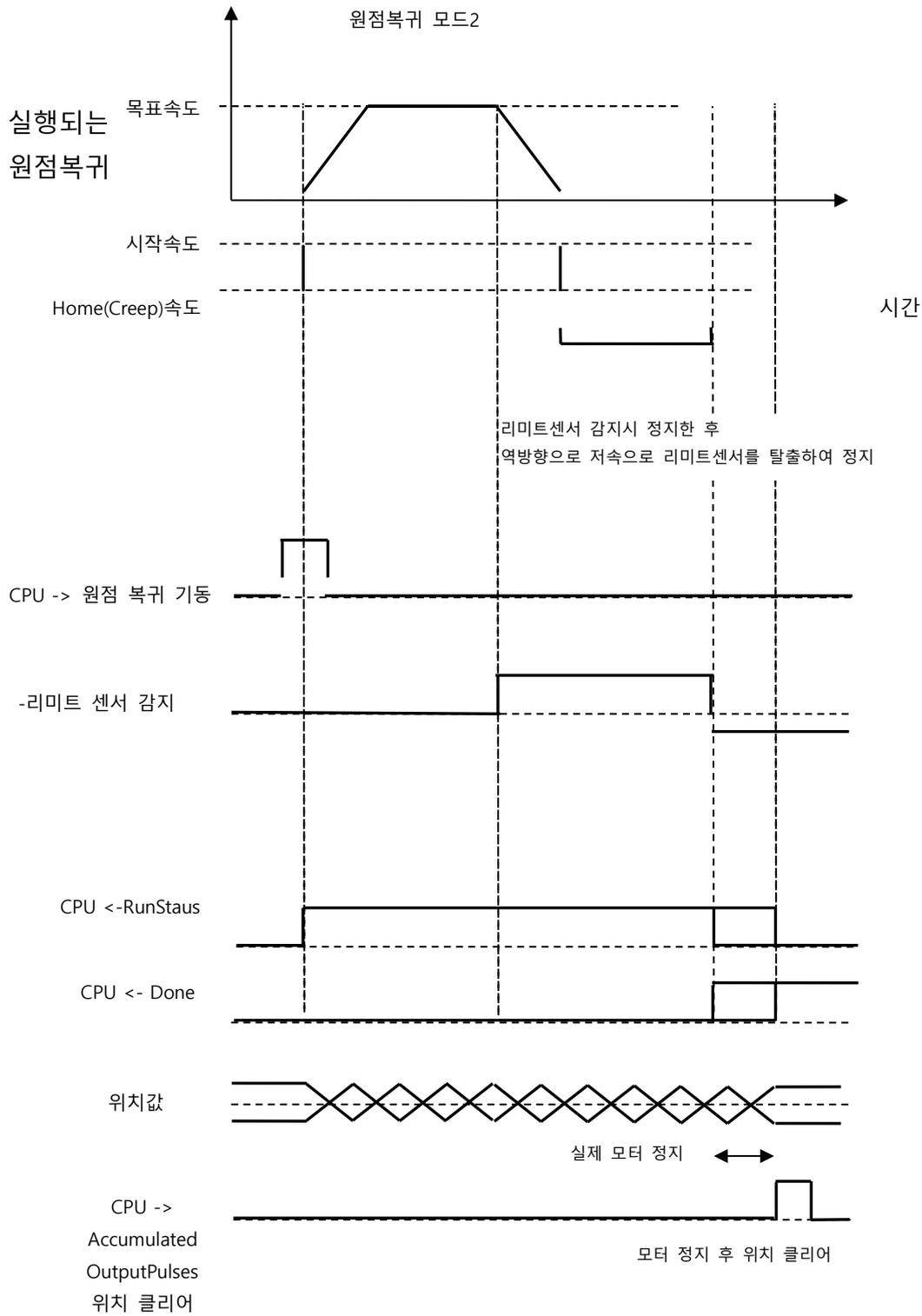
리미트 센서가 감지되면 정지한 후 역방향으로 저속 이동하여 리미트 센서를 탈출한 후 정지하는 모드입니다.

최초에 리미트 센서가 감지될 때의 정지는 설정데이터에서 설정에 따라 감속하여 정지하거나 즉시 정지의 동작을 합니다.

최초에 리미트 센서가 감지되어 정지되었을 때 이미 리미트 센서를 진행방향으로 넘어갔다면 그 자리에서 원점 동작이 종료되므로 감속정지를 사용하는 경우는 원점복귀의 가감속을 충분히 작은 값으로 하여 리미트 센서를 지나치는 경우가 없도록 하여야 합니다.

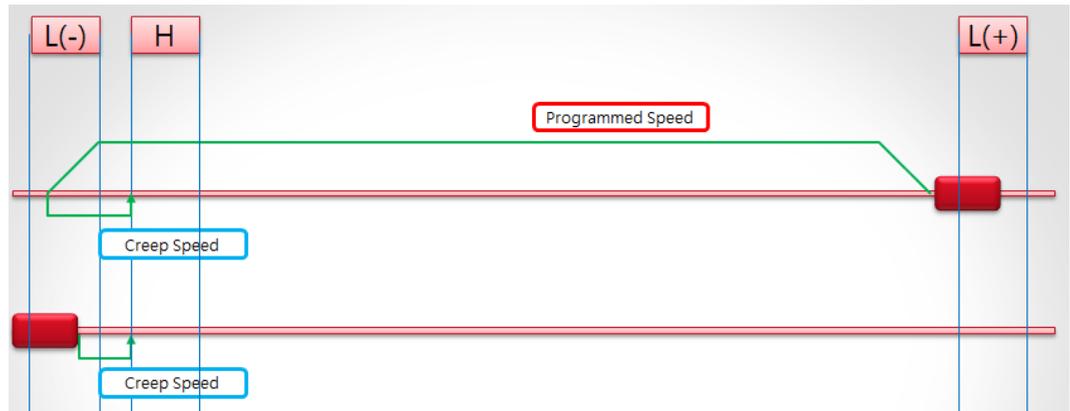
**\*3모드는 +방향으로 반대로 동작합니다.**

■ 동작 구간별 플래그 동작상태



**4번모드(5번모드:반대방향)**

시계반대 방향(-방향) -리미트 서치 후 원점시작점 원점 복귀



모드	-리미트센서 (L(-))	홈센서(Z상포함) (H)	홈근접센서 (HP)	+리미트센서 (L(+))
4	○	○	x	▲
5	▲	○	x	○

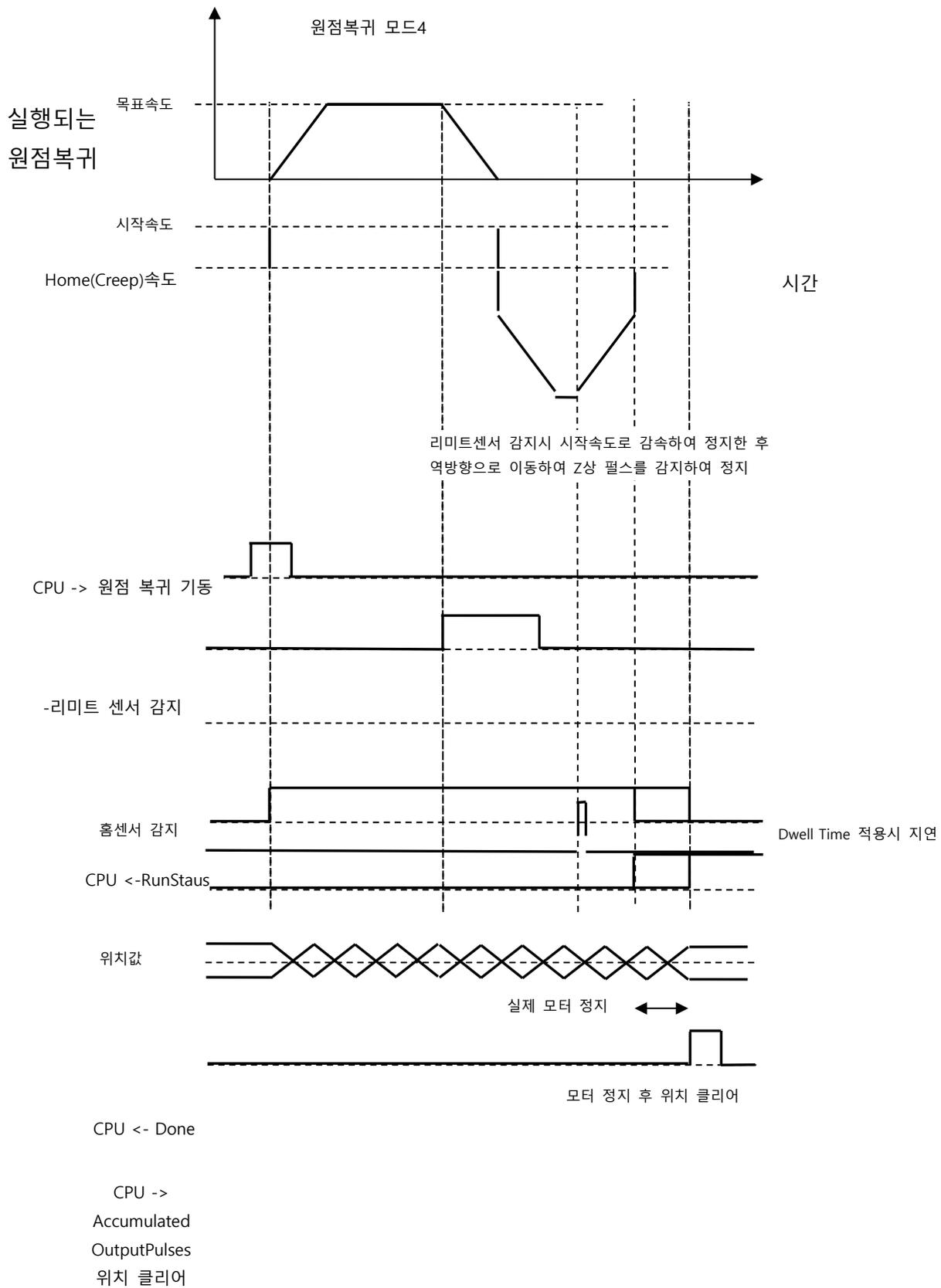
리미트 센서가 감지되면 정지한 후 역방향으로 저속 이동하여 리미트 센서를 탈출한 후 홈센서 시작부분에 정지하는 모드입니다.

최초에 리미트 센서가 감지될 때의 정지는 설정 데이터에서 설정에 따라 홈센서 감지후 감속하여 정지하거나 즉시 정지의 동작을 합니다.

최초에 리미트 센서가 감지되어 정지되었을 때 이미 리미트 센서를 진행방향으로 넘어갔다면 그 자리에서 원점 동작이 종료되므로 감속정지를 사용하는 경우는 원점복귀의 가감속을 충분히 작은 값으로 하여 리미트 센서를 지나치는 경우가 없도록 하여야 합니다.

**\*5모드는 +방향으로 반대로 동작합니다.**

■ 동작 구간별 플래그 동작상태



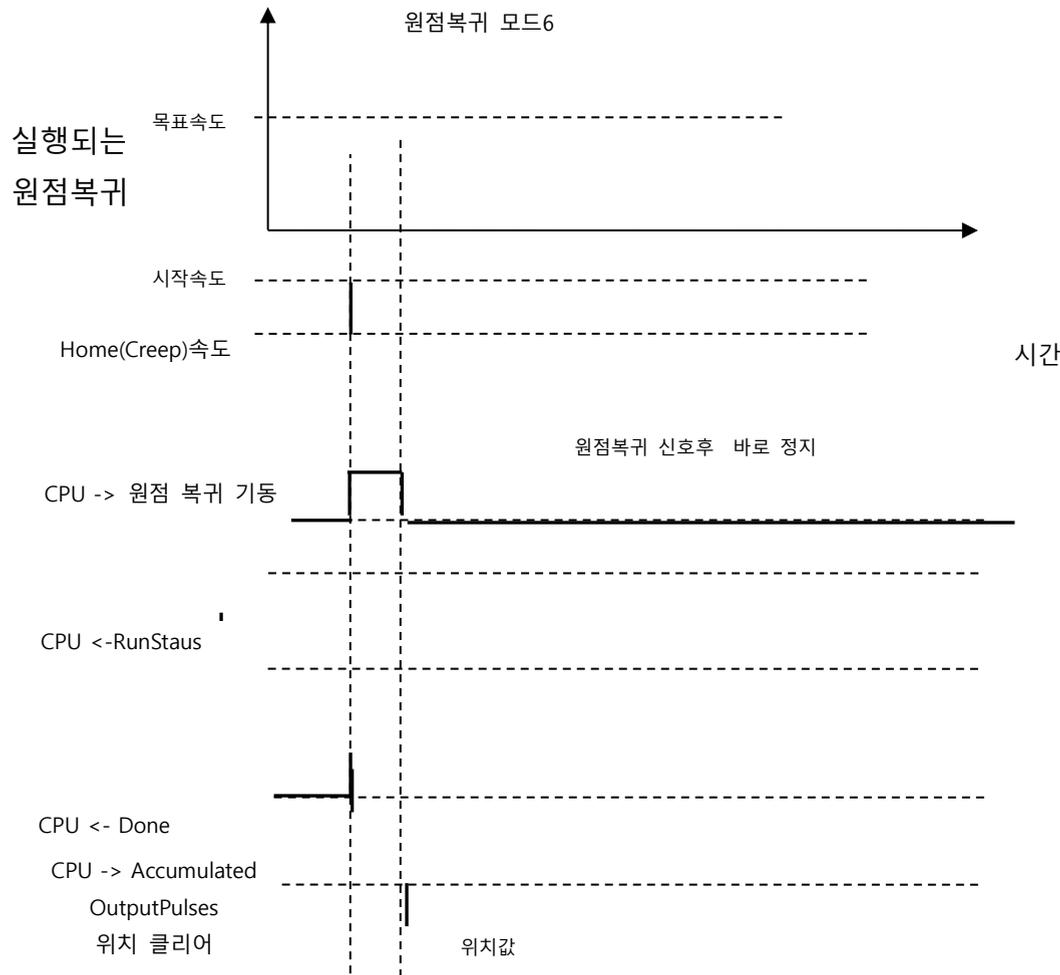
**6번모드**

현재 위치 원점시작점 원점 복귀



모드	-리미트센서 (L-)	홈센서(Z상포함) (H)	홈근접센서 (HP)	+리미트센서 (L+)
6	x	x	x	x

원점복귀 실행 후 이동 없이 현재 위치를 원점으로 시작한다.



원점복귀의 변수는 XGPC 내부의 설정된 파일 레지스트 테이블을 통해 각종 변수를 설정 및 Monitoring 합니다.  
다음은 확장 PTO 변수에 대해 간단히 설명합니다.

**확장 PTO의 구성원**

Address	Data Format	Range	Access	Explanation
<b>** 제어/상태 플래그</b>				
제어플래그	LONG	0 or 1	Read/Write	확장 PTO를 사용하고자 할 때 활성화 되며제어플래그의 값에 따라 PTO명령이 실행됨. 0 : PTO/Jog모드 1 : 원점 복귀모드
상태플래그	LONG	0 or 1	Read/Write	원점복귀 모드가 완료가 되면 상태 플래그는 완료 플래그 "1"이 발생됨
<b>** 리미트 스위치 입력 설정</b>				
-HW리미트 스위치 타입	bit	0 or 1	Read/Write	-HW리미트 스위치 타입을 설정하는 플래그 0 : 통전시 입력 유효(A접점,NOC) 1 : 비통전시 입력 유효(B접점,NCC)
원점 스위치 타입	bit	0 or 1	Read/Write	원점 스위치 타입을 설정하는 플래그 0 : 통전시 입력 유효(A접점,NOC) 1 : 비통전시 입력 유효(B접점,NCC)
원점 근접 스위치 타입	bit	0 or 1	Read/Write	원점 근접 스위치 타입을 설정하는 플래그 0 : 통전시 입력 유효(A접점,NOC) 1 : 비통전시 입력 유효(B접점,NCC)
+HW리미트 스위치 타입	bit	0 or 1	Read/Write	+HW리미트 스위치 타입을 설정하는 플래그 0 : 통전시 입력 유효(A접점,NOC) 1 : 비통전시 입력 유효(B접점,NCC)
-HW리미트 스위치 : 입력 접점 어드레스	LONG (HEX)		Read/Write	-,+HW 리미트 스위치,원점 스위치, 원점 근접 스위치는 동일 하게 구성된다. 입력 접점 어드레스로서 HEX8자리로 구성된다. 예제1 ) 00000007 앞4자리 슬롯번호 : XXXX(예:CPU 0슬롯 0) 뒷4자리 비트번호 : XXXX(예:7번째 비트 7) 예제2)00010010 앞4자리 슬롯번호 : XXXX(예:XU16 1슬롯 1) 뒷4자리 비트번호 : XXXX(예:10번째 비트 10)
원점 스위치 : 입력 접점 어드레스	LONG (HEX)		Read/Write	
원점 근접 스위치 : 입력 접점 어드레스	LONG (HEX)		Read/Write	
+HW리미트 스위치 : 입력 접점 어드레스	LONG (HEX)		Read/Write	

드레스				
<b>**PTO/Jog 모드용 SW리미트 설정</b>				
-SW리미트	LONG		Read/Write	시계반대 방향 영역 소프트 리미트
+SW리미트	LONG		Read/Write	시계 방향 영역 소프트 리미트
<b>**원점복귀 모드용 원점복귀 설정</b>				
원점복귀모드	LONG		Read/Write	<p>지원하는 7가지 모드(0~6)를 선택할 수 있다.</p> <p>0 모드 시계반대 방향(-방향) -리미트 서치 후 원점 근접센서                      확인 후 속도변경 후 원점센서 시작점 원점 복귀</p> <p>1 모드 시계방향(+방향) +리미트 서치 후 원점 근접센서                      확인 후 속도변경 후 원점센서 시작점 원점 복귀</p> <p>2 모드 시계반대 방향(-방향) -리미트 시작점 원점 복귀</p> <p>3 모드 시계방향(+방향) +리미트 서치 시작점 원점 복귀</p> <p>4 모드 시계반대 방향(-방향) -리미트 서치 후 원점센서                      끝지점 원점 복귀</p> <p>5 모드 시계방향(+방향) +리미트 서치 후 원점센서 끝지점                      원점 복귀</p> <p>6 모드 현재 위치 원점</p>
목표속도 (Pulses/sec)	LONG		Read/Write	원점복귀 등속구간 목표속도입니다.
Creep 속도 (Pulses/sec)	LONG		Read/Write	원점 근접 센서 감지 후 원점 속도 입니다.
가속펄스 수	LONG		Read/Write	원점 복귀 등속구간까지 가속펄스 수 입니다.
감속펄스 수	LONG		Read/Write	원점 복귀 감속펄스 수 입니다.
원점복귀 완료 후. AccumulatedOutput pulses의 초기값	LONG		Read/Write	원점 복귀 완료 후 내부 메모리의 현재 위치 펄스 값. 기본값 : 0

사용자는 확장 PTO 설정(원점복귀설정)에서 아래 설정을 완료하고 PTO 명령을 실행하면 원점복귀 명령을 확인 할 수 있다.

- 아 래 -

주소	데이터 ...	값	의미	코멘트
<b>&gt;&gt; PTO 0</b>				
<b>** 제어/상태 플래그</b>				
L100,0	Long	0	PTO/Jog 모드	제어 플래그
L100,1	Long	0	-	상태 플래그
<b>** 리미트 스위치 입력 설정</b>				
L100,2,0	Bool	0	통전시 입력유효(A접점)	- HW 리미트 스위치 타입
L100,2,1	Bool	0	통전시 입력유효(A접점)	원점 스위치 타입
L100,2,2	Bool	0	통전시 입력유효(A접점)	원점 근접 스위치 타입
L100,2,3	Bool	0	통전시 입력유효(A접점)	+ HW 리미트 스위치 타입
L100,3	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx]	- HW 리미트 스위치 : 입력 접점 어드레스
L100,4	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx]	원점 스위치 : 입력 접점 어드레스
L100,5	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx]	원점 근접 스위치 : 입력 접점 어드레스
L100,6	Long	FFFFFFFF (h)	- [xxxxyyyy(h) : xxx]	+ HW 리미트 스위치 : 입력 접점 어드레스
<b>** PTO/Jog 모드를 SW 리미트 설정</b>				
L100,8	Long	0	pulses	- SW 리미트
L100,9	Long	0	pulses	+ SW 리미트
<b>** 원점복귀 모드용 원점복귀 설정</b>				
L100,10	Long	0	-	원점복귀 모드
L100,11	Long	0	-	목표 속도(pulses/second)
L100,12	Long	0	-	Creep 속도(pulses/second)
L100,13	Long	0	-	가속펄스수
L100,14	Long	0	-	감속펄스수
L100,15	Long	0	-	원점복귀 완료 후 ,AccumulatedOutputPulses

우선 PTO설정은 위 PTO기능에서 예제로 되어 있는 설정을 기본적으로 사용 한다.

원점복귀명령에 필요한 각 설정을 완료한다.

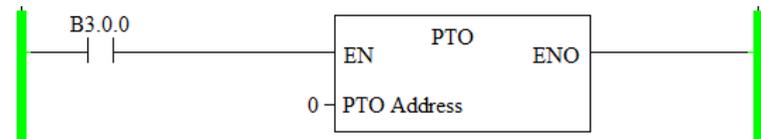
- 1) 모드제어플래그(" 1:원점복귀모드 ")
- 2) 스위치 타입설정(0 : A접점, 1 : B접점)
- 3) 센서 입력 스위치 접점어드레스(XXXX : SLOT, XXXX : BIT NO)
- 4) 원점복귀 모드(0~6)
- 5) 목표 속도 ( 원점 목표 속도)
- 6) Creep 속도 (원점근접 센스 확인후 저속 속도)
- 7) 가속펄스수, 감속 펄스수(가감속 펄스 수)
- 8) 원점복귀 완료 후 내부 시작 펄스값

총 8개의 항목을 입력 한 후, 래더 프로그램 상에 PTO 명령을 사용하면 간단히 펄스 출력 예제를 만들 수 있습니다.

- 1) 모드제어플래그(" 1:원점복귀모드 ")
- 2) 스위치 타입설정(0 : A접점)
- 3) 센서 입력 스위치 접점어드레스(0 : SLOT, F : 15BIT NO X1.0.0.15)
- 4) 원점복귀 모드(0)
- 5) 목표 속도 ( 50000pps 원점 목표 속도)
- 6) Creep 속도 (5000pps 원점근접 센스 확인후 저속 속도)
- 7) 가속펄스수, 감속 펄스수(100 가감속 펄스 수)
- 8) 원점복귀 완료 후 시작 펄스값(0)

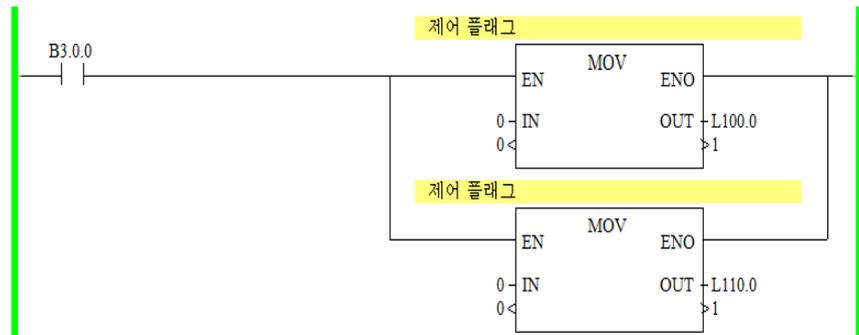
으로 설정

어드레스	데이터 ...	값	의미	코멘트
<b>&gt;&gt; PTO 1</b>				
<b>** 제어/상태 플래그</b>				
L110.0	Long	1	원점복귀 모드	제어 플래그
L110.1	Long	0		상태 플래그
<b>** 리미트 스위치 입력 설정</b>				
L110.2.0	Bool	0	통전시 입력유효(A접점)	- HW 리미트 스위치 타입
L110.2.1	Bool	1	비통전시 입력유효(B접점)	원점 스위치 타입
L110.2.2	Bool	0	통전시 입력유효(A접점)	원점 근접 스위치 타입
L110.2.3	Bool	0	통전시 입력유효(A접점)	+ HW 리미트 스위치 타입
L110.3	Long	00000000 (h)	입력 : X0.0.0.13	- HW 리미트 스위치 : 입력 접점 어드레스
L110.4	Long	00000009 (h)	입력 : X0.0.0.9	원점 스위치 : 입력 접점 어드레스
L110.5	Long	00000009 (h)	입력 : X0.0.0.9	원점 근접 스위치 : 입력 접점 어드레스
L110.6	Long	0000000F (h)	입력 : X0.0.0.15	+ HW 리미트 스위치 : 입력 접점 어드레스
<b>** PTO/Jog 모드용 SW 리미트 설정</b>				
L110.8	Long	0	-	- SW 리미트
L110.9	Long	0	-	+ SW 리미트
<b>** 원점복귀 모드용 원점복귀 설정</b>				
L110.10	Long	0	왼쪽에서 시작해서, 원점 스위	원점복귀 모드
L110.11	Long	50000	pulses/second	목표 속도(pulses/second)
L110.12	Long	5000	pulses/second	Creep 속도(pulses/second)
L110.13	Long	100	pulses	가속펄스수
L110.14	Long	100	pulses	감속펄스수
L110.15	Long	0	pulses	원점복귀 완료 후 AccumulatedOutputPulses



위와 같이 SFR에서 8개의 변수들을 지정한 값으로 설정 해 준 후, 래더 프로그램 상에서 B3.0.0이 'ON' 되면 설정한 값으로 원점복귀가 동작합니다.

기존 변수값을 변경하고자 할 때는 직접 변수 값을 바꿀 수도 있지만 래더에서도 변경이 가능합니다.



## PWM 출력 기능

### PWM (Pulse Width Modulation)

임의주파수의 Duty를 0% ~ 100% 범위 내에서 1%단위로 가변 출력할 수 있는 기능입니다. 임의 주파수는 1Hz ~ 30kHz 이내로, 1Hz 단위로 설정할 수 있습니다.

PWM은 아래 두 가지의 중요 기능이 있습니다.

- 주파수 발생
- Duty Cycle 간격 조절

### PWM 출력 기능 설정 방법

#### 사용 순서 1. 결선

앞의 PTO와 같습니다.

#### 사용 순서 2. 변수 설정

고속 펄스 출력의 변수는 XGPC 내부의 SFR(Special Function Register) 테이블을 통해 각종 변수를 설정 및 Monitoring 합니다.

다음은 PTO SFR에 대해 간단히 설명합니다.

### PWM SFR의 구성원

Address	Data Format	Range	User Program Access	Explanation
Control				
Output	Word(INT)	2~4	Read Only	PWM 명령에 의해 실제 출력이 나가는 출력 포트를 설정합니다.
Profile Parameter Select	bit	0 or 1	Read/Write	어떤 파형으로 출력할 것인지 설정합니다. 주파수(Frequency)출력 시 : ON Duty Cycle 설정 시 : OFF
Enable Hard Stop	bit	0 or 1	Read/Write	임의적으로 PWM의 동작을 중단 시킬 때 사용합니다.
Output Frequency	Long word	0~100,000	Read/Write	PWM의 주파수(Frequency)를 지정합니다.
Duty Cycle	Word	0~1000	Read/Write	Duty Cycle을 정의합니다. 1000 : 100% Output ON (Constant, no waveform) 500 : 50% Output ON, 50% output OFF 250 : 25% Output ON, 75% output OFF

AccelDecel Delay	Word	0~32,767	Read/Write	0Hz ~20kHz까지 주파수 발생 시 10ms 범위 안에서의 지연 시간을 설정할 수 있고, 언제든지 값의 변경이 가능합니다.
Status				
Error Code	Word(INT)	-2~5	Read Only	Error 발생 시 Error Code를 나타냅니다.
Decelerating Status	Bit	0 or 1	Read Only	PWM의 출력상황이 감속 위상 구간에 있을 때 'ON'으로 설정됩니다.
Run Status	Bit	0 or 1	Read Only	PWM의 상태가 실행 중일 때 'ON'으로 설정됩니다.
Accelerating Status	Bit	0 or 1	Read Only	PWM의 출력 상황이 가속 위상 구간에 있을 때 'ON'으로 설정됩니다.
Idle Status	Bit	0 or 1	Read Only	PWM의 상태가 Idle일 때 'ON'으로 설정됩니다.
Error Detected Status	Bit	0 or 1	Read Only	PWM의 Error 발생 시 'ON'으로 설정됩니다.
Normal Operation Status	Bit	0 or 1	Read Only	PWM이 Error가 없는 상태에서 가/감속, RUN 상태에 있을 경우 'ON'으로 설정됩니다.
Enable Status	Bit	0 or 1	Read Only	PWM이 동작 가능하도록 하는 입력 신호입니다.
Operation Frequency Status	Long word	0~100,000	Read Only	PWM이 동작하고 있는 주파수(Frequency)를 표시해 줍니다.
Duty Cycle Status	Word(INT)	1~1000	Read Only	PWM이 동작하고 있는 Duty Cycle 정보를 표시해 줍니다.



## 10장. X8-POSITION Control

본 10장은 X8-POSITION MODULE을 이용하여 CSD7 Servo Drive와 Motor의 위치 제어 하는 방법을 제시합니다. X8-POSITION MODULE은 총 4축의 고속 출력(라이드라이브)이 가능하며 4Mpps 까지 지원합니다. 포지션 모듈은 증설유닛으로 장착하여 1,2,4축으로 사용 가능 합니다. 추가 장착이 가능하나 소비전류를 잘 파악하셔야 하며 20축이상도 가능합니다.

---

개요.....	10-2
Configuration.....	10-3
Example.....	10-13

## 개요

### Target Applications

본 Application Note는 X8-POSITION MODULE 이 지원하는 포지션 기능으로 독립모션 제어 및 보간제어 등을 응용할 수 있습니다.

### Prerequisites

- Knowledge of XGPC, X8-PM and X8-POSITION MODULE Programming
- Knowledge of CSD5/7 Servo Drive
- Knowledge of Miscellaneous wiring

### Test Environment & Equipment

- X8-POSITION MODULE PLC, CSD5/7 Servo Drive, Motor, (또는 PLC, Servo, Motor, Touch까지 구성된 X8-POSITION MODULE-DEMOKIT)
- XGPC 1.0.0.8 Software, PC window 7

### Related Publication

- X8-POSITION MODULE Installation Manual (RSA홈페이지에서 제공. <http://www.rsautomation.co.kr>)
- X8-POSITION MODULE User Manual (RSA홈페이지에서 제공)
- CSD5 Servo Drive Installation Instruction (RSA홈페이지에서 제공)
- CSD5 Servo Drive User Manual (RSA홈페이지에서 제공)

# Configuration

## 설치배선(Wiring)

### AX1 AX2

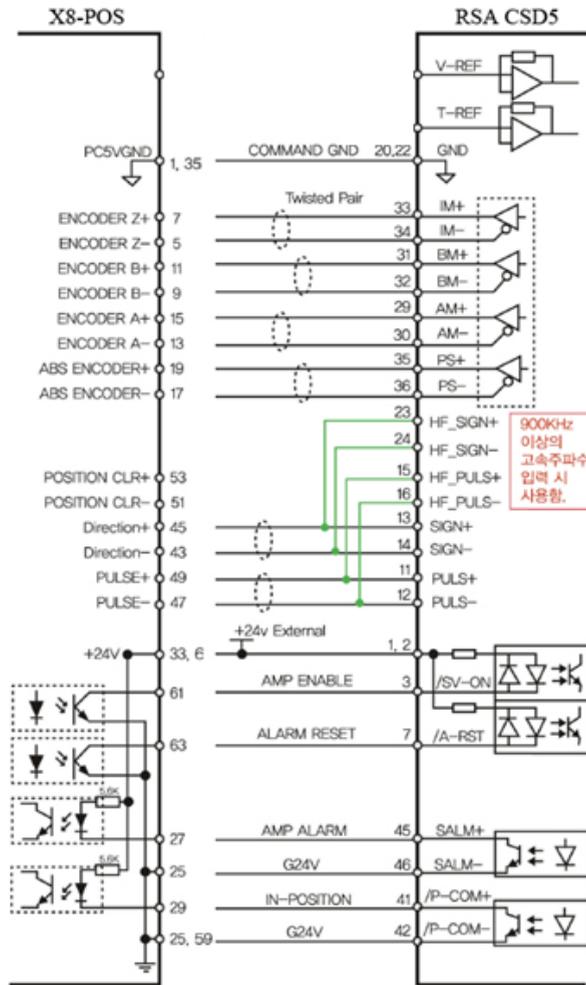
핀	신호	축	핀	신호	축
1	G5V (GND)	1	2	G5V (GND)	2
3	5V (+5V)	1	4	5V (+5V)	2
5	EZ- (Encoder)	1	6	EZ- (Encoder)	2
7	EZ+ (Encoder)	1	8	EZ+ (Encoder)	2
9	EB- (Encoder)	1	10	EB- (Encoder)	2
11	EB+ (Encoder)	1	12	EB+ (Encoder)	2
13	EA- (Encoder)	1	14	EA- (Encoder)	2
15	EA+ (Encoder)	1	16	EA+ (Encoder)	2
17	ABS- (Encoder)	1	18	ABS- (Encoder)	2
19	ABS+ (Encoder)	1	20	ABS+ (Encoder)	2
21	Catch In 1	1	22	Catch In 2	2
23	Home Sensor Input	1	24	Home Sensor Input	2
25	G24V (EX GND)	1	26	G24V (EX GND)	2
27	Servo Drive Alarm Input	1	28	Servo Drive Alarm Input	2
29	Servo Drive Inposition Input	1	30	Servo Drive Inposition Input	2
31	Positive Limit Sensor Input	1	32	Positive Limit Sensor Input	2
33	24V (EX +24V)	1	34	24V (EX +24V)	2
35	G5V (GND)	1	36	G5V (GND)	2
37	-	1	38	-	2
39	Catch In 3	1	40	Catch In 4	2
41	-	1	42	-	2
43	CCW- (Sign-)	1	44	CCW- (Sign-)	2
45	CCW+ (Sign+)	1	46	CCW+ (Sign+)	2
47	CW- (Pulse-)	1	48	CW- (Pulse-)	2
49	CW+ (Pulse+)	1	50	CW+ (Pulse+)	2
51	5V TTL Output-	1	52	5V TTL Output-	2
53	5V TTL Output+	1	54	5V TTL Output+	2
55	-	1	56		2
57	MPG A*	1	58	MPG B*	2
59	G24V (EX GND)	1	60	G24V (EX GND)	2
61	Servo Drive On Output	1	62	Servo Drive On Output	2
63	Servo Drive Reset Output	1	64	Servo Drive Reset Output	2
65	Negative Limit Sensor Input	1	66	Negative Limit Sensor Input	2
67	24V (EX +24V)	1	68	24V (EX +24V)	2

**AX3 AX4**

핀	신호	축	핀	신호	축
1	G5V (GND)	3	2	G5V (GND)	4
3	5V (+5V)	3	4	5V (+5V)	4
5	EZ- (Encoder)	3	6	EZ- (Encoder)	4
7	EZ+ (Encoder)	3	8	EZ+ (Encoder)	4
9	EB- (Encoder)	3	10	EB- (Encoder)	4
11	EB+ (Encoder)	3	12	EB+ (Encoder)	4
13	EA- (Encoder)	3	14	EA- (Encoder)	4
15	EA+ (Encoder)	3	16	EA+ (Encoder)	4
17	ABS- (Encoder)	3	18	ABS- (Encoder)	4
19	ABS+ (Encoder)	3	20	ABS+ (Encoder)	4
21	Compare Out 1-1	3	22	Compare Out 1-2	4
23	Home Sensor Input	3	24	Home Sensor Input	4
25	G24V (EX GND)	3	26	G24V (EX GND)	4
27	Servo Drive Alarm Input	3	28	Servo Drive Alarm Input	4
29	Servo Drive Inposition Input	3	30	Servo Drive Inposition Input	4
31	Positive Limit Sensor Input	3	32	Positive Limit Sensor Input	4
33	24V (EX +24V)	3	34	24V (EX +24V)	4
35	G5V (GND)	3	36	G5V (GND)	4
37	-	3	38	-	4
39	Compare Out 2-1	3	40	Compare Out 2-2	4
41	Compare Out 3-1	3	42	Compare Out 3-2	4
43	CCW- (Sign-)	3	44	CCW- (Sign-)	4
45	CCW+ (Sign+)	3	46	CCW+ (Sign+)	4
47	CW- (Pulse-)	3	48	CW- (Pulse-)	4
49	CW+ (Pulse+)	3	50	CW+ (Pulse+)	4
51	5V TTL Output-	3	52	5V TTL Output-	4
53	5V TTL Output+	3	54	5V TTL Output+	4
55	Compare Out 4-1	3	56	Compare Out 4-2	4
57	-	3	58	-	4
59	G24V (EX GND)	3	60	G24V (EX GND)	4
61	Servo Drive On Output	3	62	Servo Drive On Output	4
63	Servo Drive Reset Output	3	64	Servo Drive Reset Output	4
65	Negative Limit Sensor Input	3	66	Negative Limit Sensor Input	4
67	24V (EX +24V)	3	68	24V (EX +24V)	4

### 결선도

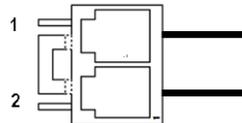
Case 1. X8-POSITION MODULE과 CSD5/7 Servo Drive를 바로 연결할 경우, 다음과 같이 결선합니다.



※ CSD5/7 Servo Drive와 포지션 모듈을 바로 연결 하실 때에는 24V전압입력을 포지션 모듈에 있는 2P커넥트에 결선합니다.

#### ④ 외부 전원공급 포트

외부 24VDC SMPS전원 연결용 포트입니다.

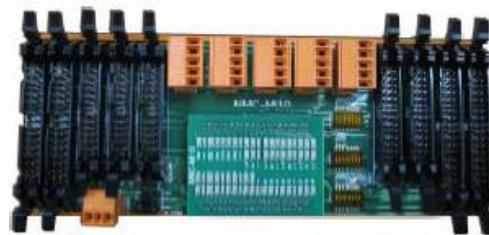


핀	신호
1	24V(+)
2	G24V(-)

Case 2. X8-POSITION MODULE To Option B'd MI10 연결할 경우

- MMC-MI10 연결사양

결선용 인터페이스 모듈



위치결정 모듈 커넥터(J11, J12, J21, J22) pin Map

MMC에서 MI 10 커넥터 J11~J22 Pin MAP						
MMC Con (34pin)	신호명				그룹	설명
	J11(0축)	J12(1축)	J21(2축)	J22(3축)		
1	GND	GND	GND	GND	Power Source	5V GND
2	VCC	VCC	VCC	VCC	Power Source	5V GND
3	A0-Z	A1-Z	A2-Z	A3-Z	Encoder	Encoder Z-
4	A0+Z	A1+Z	A2+Z	A3+Z	Encoder	Encoder Z+
5	A0-B	A1-B	A2-B	A3-B	Encoder	Encoder B-
6	A0+B	A1+B	A2+B	A3+B	Encoder	Encoder B+
7	A0-A	A1-A	A2-A	A3-A	Encoder	Encoder A
8	A0+A	A1+A	A2+A	A3+A	Encoder	Encoder A+
9	ABS0-	ABS1-	ABS2-	ABS3-	Encoder	Encoder Abs
10	ABS0+	ABS1+	ABS2+	ABS3+	Encoder	Encoder Abs
11	Input0	Input1	Output0	Output1	I/O	User I/O
12	HOME0	HOME1	HOME2	HOME0	Sensor	Home Sensor Input
13	G24V	G24V	G24V	G24V	Power Source	External 24V GND
14	SV0ERR	SV1ERR	SV2ERR	SV3ERR	Motion I/O	AMP Fault Input
15	PCIN0	PCIN1	PCIN2	PCIN3	Motion I/O	In-Position Input

MMC에서 MI 10 커넥터 J11~J22 Pin MAP						
MMC Con (34pin)	신호명				그룹	설명
	J11(0축)	J12(1축)	J21(2축)	J22(3축)		
16	PLIM0	PLIM1	PLIM2	PLIM3	Sensor	Positive Limit Input
17	+24V	+24V	+24V	+24V	Power Source	External 24V Power
18	GND	GND	GND	GND	Power Source	5V GND
19	SV0AO	SV1AO	SV2AO	SV3AO	I/O	Analog Signal Output
20	Input 2	Input 3	Output 2	Output 3	I/O	User I/O
21	Input 4	Input 5	Output 4	Output 5	I/O	User I/O
22	A0-DIR	A1-DIR	A2-DIR	A3-DIR	Pulse Output	CCW Pulse & Direction Signal Output(/CCW)
23	A0+DIR	A1+DIR	A2+DIR	A3+DIR	Pulse Output	CCW Pulse & Direction Signal Output(CCW)
24	A0-CLK	A1-CLK	A2-CLK	A3-CLK	Pulse Output	CW Pulse & Pulse Output(/CW)
25	A0+CLK	A1+CLK	A2+CLK	A3+CLK	Pulse Output	CW Pulse & Pulse Signal Output(CW)
26	PCLR0-	PCLR1-	PCLR2-	PCLR3-	Pulse Output	Position Clear Output(/P-CLR)
27	PCLR0+	PCLR1+	PCLR2+	PCLR3+	Pulse Output	Position Clear Output(/P-CLR)
28	Input 6	Input 7	Output 6	Output 7	I/O	User I/O
29	Input 8	Input 9	Output 8	Output 9	I/O	User I/O
30	G24V	G24V	G24V	G24V	Power Source	External 24V GND
31	SV0ON	SV1ON	SV2ON	SV3ON	Motion I/O	AMP Enable(Servo On) Output
32	SV0RST	SV1RST	SV2RST	SV3RST	Motion I/O	AMP Fault Reset Output
33	NLIM0	NLIM1	NLIM2	NLIM3	Sensor	Negative Limit Input
34	+24V	+24V	+24V	+24V	Power Source	External 24V Power

\* AMP(Servo) 커넥터 (AX0, AX1, AX2, AX3) Pin MAP

MMC에서 MI 10 AMP(Servo) 커넥터 AX0 ~ AX3 Pin MAP						
Limit Con (5Pin)	커넥터				그룹	설명
	AX0	AX1	AX2	AX3		
1	SV0AO	SV1AO	SV2AO	SV3AO	Motion I/O	Analog Signal Output
2	GND	GND	GND	GND	Power Source	5V GND
3	A0+Z	A1+Z	A2+Z	A3+Z	Encoder	Encoder Z+
4	A0-Z	A1-Z	A2-Z	A3-Z	Encoder	Encoder Z-
5	A0+B	A1+B	A2+B	A3+B	Encoder	Encoder B+
6	A0-B	A1-B	A2-B	A3-B	Encoder	Encoder B-
7	A0+A	A1+A	A2+A	A3+A	Encoder	Encoder A+

8	A0-A	A1-A	A2-A	A3-A	Encoder	Encoder A-
9	ABS0+	ABS1+	ABS2+	ABS3+	Encoder	Encoder Abs
10	ABS0+	ABS1+	ABS2+	ABS3+	Encoder	Encoder Abs
11	PCLR0+	PCLR1+	PCLR2+	PCLR3+	Pulse Output	Position Clear Output(P-CLR)
12	PCLR0-	PCLR1-	PCLR2-	PCLR3-	Pulse Output	Position Clear Output(/P-CLR)
13	A0+DIR	A1+DIR	A2+DIR	A3+DIR	Pulse Output	CCW Pulse & Direction Signal Output(CCW)
14	A0-DIR	A1-DIR	A2-DIR	A3-DIR	Pulse Output	CCW Pulse & Direction Signal Output(/CCW)
15	A0+CLK	A1+CLK	A2+CLK	A3+CLK	Pulse Output	CW Pulse & Pulse Signal Output(CW)
16	A0-CLK	A1-CLK	A2-CLK	A3-CLK	Pulse Output	CW Pulse & Pulse Output(/CW)
17	-	-	-	-	-	Non Connection
18	-	-	-	-	-	Non Connection
19	SV0ON	SV1ON	SV2ON	SV3ON	Motion I/O	AMP Enable(Servo On) Output
20	SV0ERR	SV1ERR	SV2ERR	SV3ERR	Motion I/O	AMP Fault Input
21	SV0RST	SV1RST	SV2RST	SV3RST	Motion I/O	AMP Fault Reset Output
22	PCIN0	PCIN1	PCIN2	PCIN3	I/O	In-Position Input
23	-	-	-	-	-	Non Connection
24	-	-	-	-	-	Non Connection
25	G24V	G24V	G24V	G24V	Power Source	External 24V GND
26	+24V	+24V	+24V	+24V	Power Source	External 24V Power

\* Limit 커넥터 Pin MAP (LIMIT0, LIMIT1, LIMIT2, LIMIT3)

MI10에서 Limit 센서 커넥터 LIMIT0~LIMIT3) Pin MAP

Limit Con (5Pin)	커넥터				설명
	Limit0	Limit1	Limit2	Limit3	
1	+24V	+24V	+24V	+24V	External 24V
2	PLMT0	PLMT1	PLMT2	PLMT3	Positive Limit
3	HOME0	HOME1	HOME2	HOME0	Home Sensor
4	NLMT0	NLMT1	NLMT2	NLMT3	Negative Limit
5	GND	GND	GND	GND	24V GND

\* MPG 입력 Pin MAP (MPGA)

Limit Con (5Pin)	커넥터	설명	MPGA Con (5Pin)	커넥터	설명
	Limit0			Limit0	
1	+24V	EXT 24V	4	-	Non Connection
2	User In 8	MPG Input Phase-A(or B)	5	GND	24V GND
3	User In 9	MPG Input Phase-B(or A)			

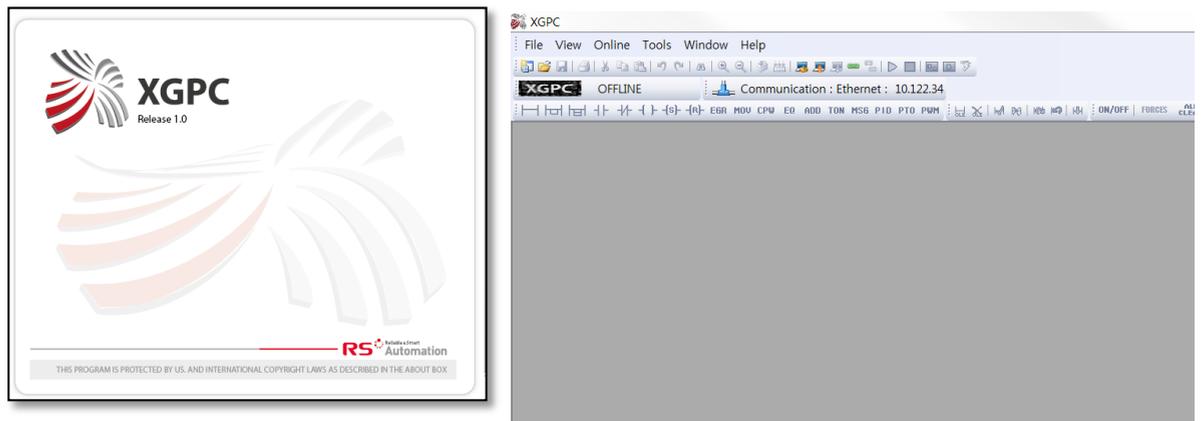
### I/O 보드 연결사양

\* 사용자 I/O 커넥터 Pin MAP (IOCON)

Pin No.	User I/O	Pin No.	User I/O	Pin No.	User I/O	Pin No.	User I/O
1	User Out 0	8	User Out 7	15	User In 4	22	Non Connection
2	User Out 1	9	User Out 8	16	User In 5	23	24V GND
3	User Out 2	10	User Out 9	17	User In 6	24	EXT 24V
4	User Out 3	11	User In 0	18	User In 7	25	24V GND
5	User Out 4	12	User In 1	19	User In 8	26	EXT 24V
6	User Out 5	13	User In 2	20	User In 9		
7	User Out 6	14	User In 3	21	Non Connection		

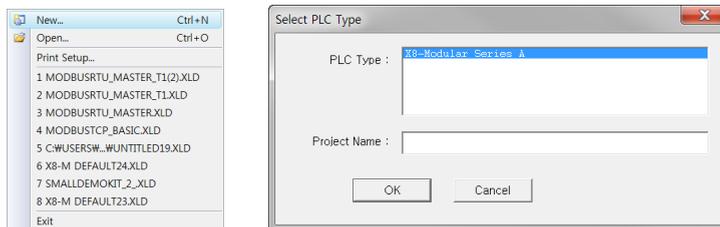
### X8-POSITION MODULE Setting

Step 1. X8 전용 래더 소프트웨어인 XGPC를 설치하고 실행.



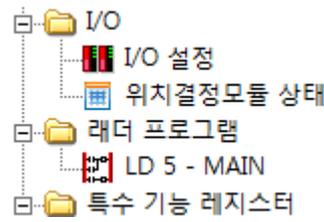
Step 2. 새 프로젝트 생성

파일 메뉴에서 New 버튼을 누릅니다. PLC type 창에서 X8-CPU SERIES 하드웨어 모듈을 선택하고, 프로젝트 이름을 입력합니다.

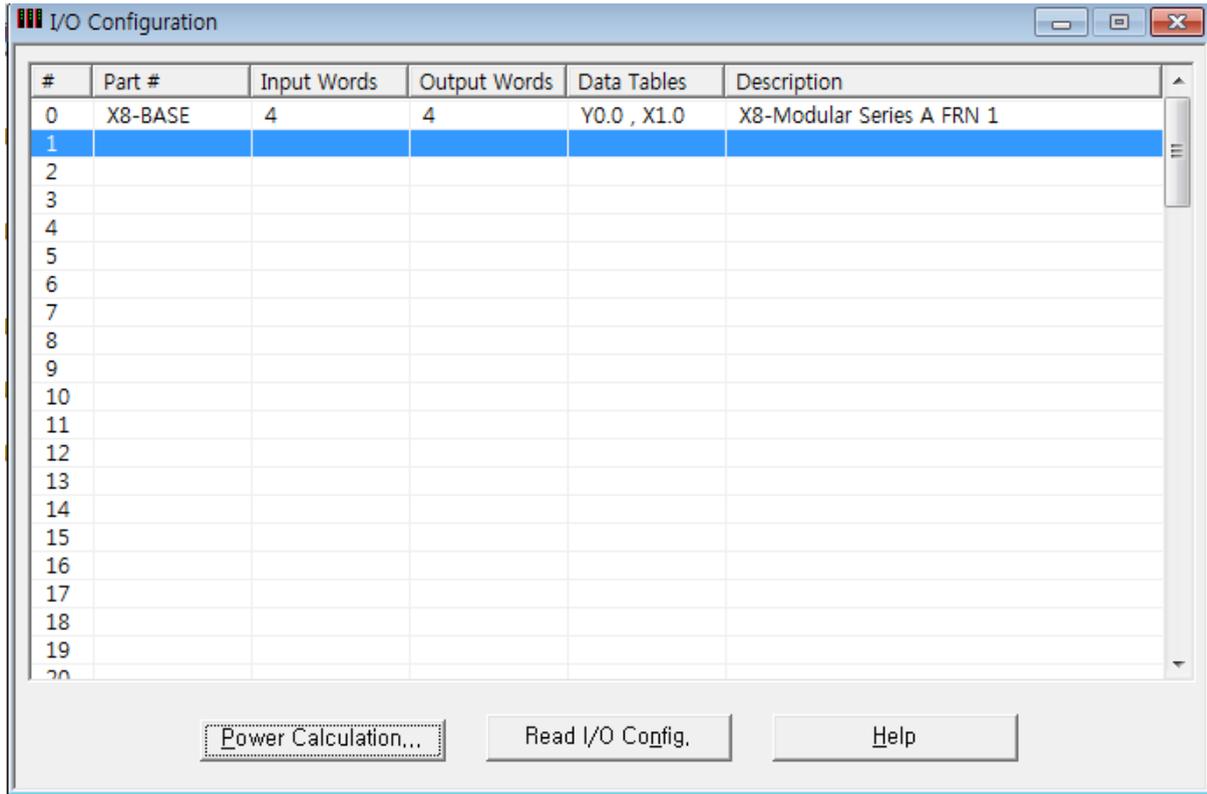


Step 3. PLC Read I/O(CPU에 포지션 인식시키기)

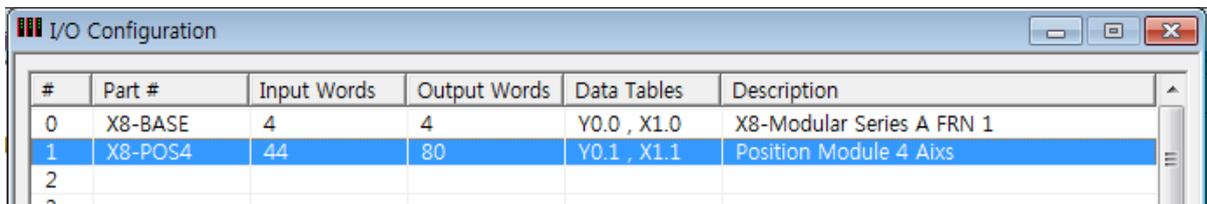
파란색위치의 I/O Configuration 을 클릭한다.



클릭하면 아래 창이 나타납니다.



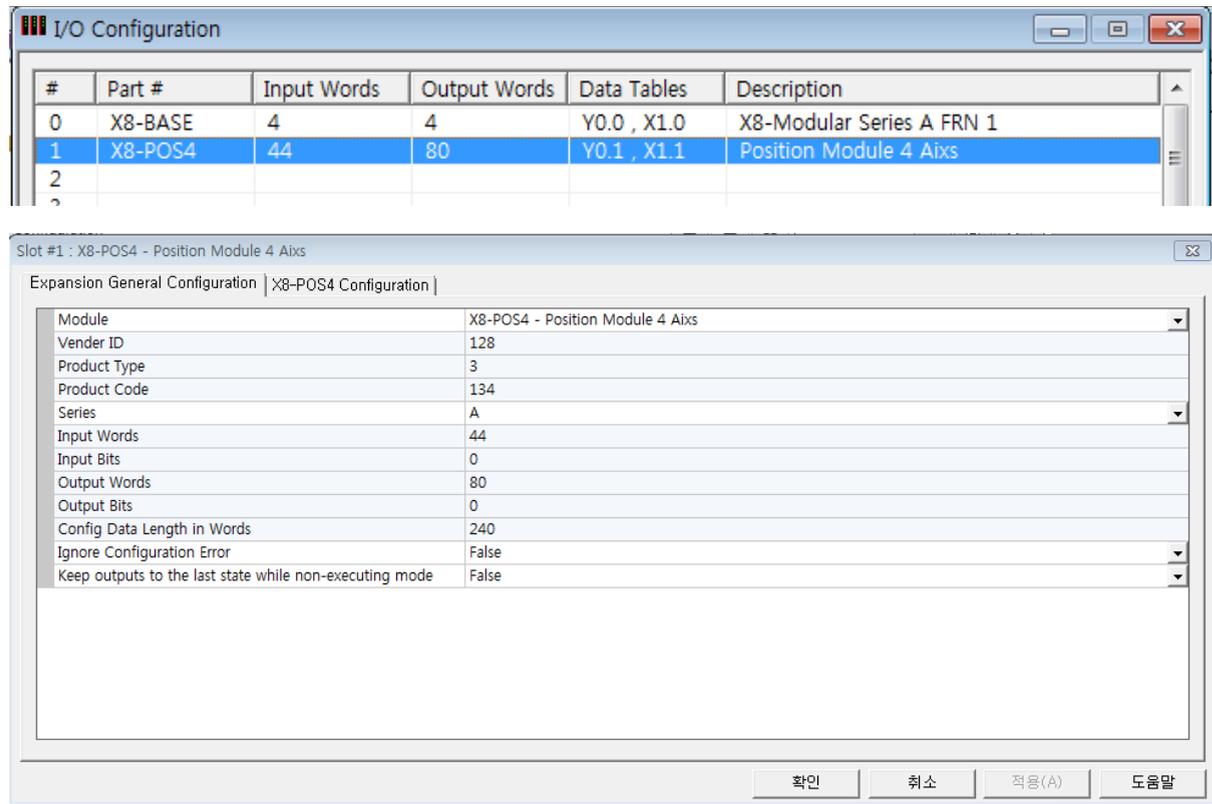
그리고 제품을 장착후 READ I/O Config 버튼을 클릭하시면 포지션 제품이 나타납니다.



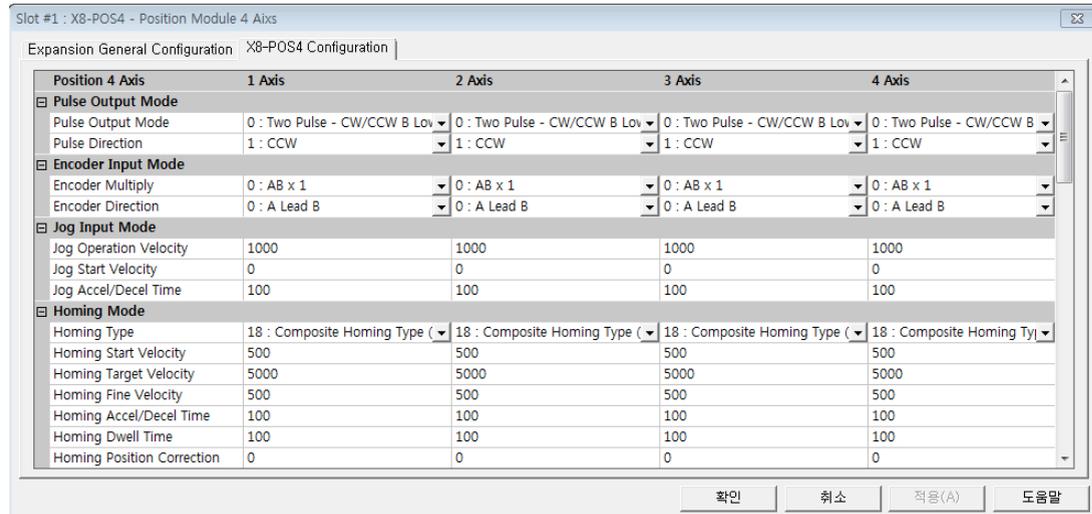
**Step 4. POSITION MODULE Configuration**

POSITION MODULE Configuration 설정은 CSD5 SERVO와 ROBOT의 다양한 CONFIG사항을 포지션 모듈에서 정리할 수 있습니다. 예를 들어 엔코드 값이 반대로 들어오거나 사용자가 원하는 방향으로 ROBOT가 구동되지 않을 시 방향을 바꿀 수 있습니다.

파란색 선택되어진 포지션모듈을 더블 클릭하시면 Configuration 창으로 이동 하실 수 있습니다.



여기에서 X8-POS4 Configuration 탭을 선택하시면 됩니다.



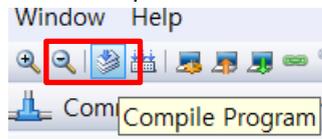
모든 설정이 완료 시 다운로드 하시면 됩니다. 적용은 CPU RUN 시 반영됩니다.

**Step 5 PLC Online 및 Download**

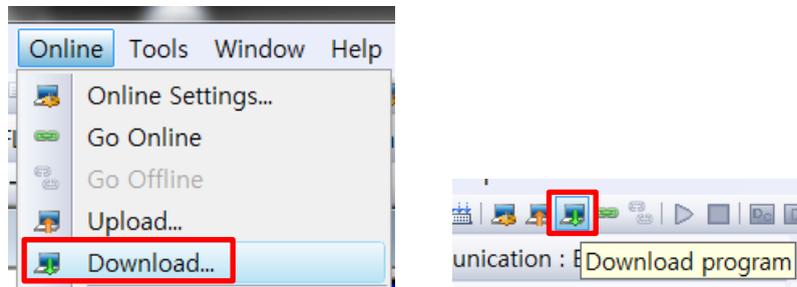
XGPC에서 Programming을 끝나치고 난 후, Compile 버튼을 실행하여 Programming상 오류가 있는지 확인합니다. Compile 버튼은 Compile Program과 Compile Project가 있습니다.

Compile Program : 현재 Programming한 Ladder Program만 Compile

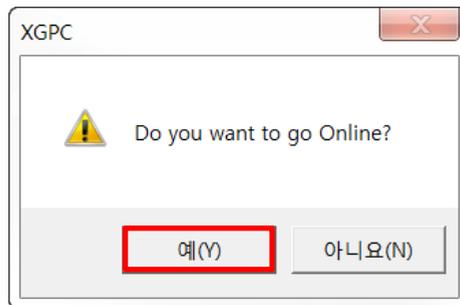
Compile Project : 현재 열려 있는 Project에 있는 모든 Ladder Program이 Compile



Compile이 완료된 후, 작성한 Project를 PLC로 전송하기 위해 Download 버튼을 누릅니다. Download 버튼은 Online 메뉴에서 클릭하거나, 아이콘 Toolbar에서 사용할 수 있습니다.



Download를 하고 나면, Online을 할 것인지 메시지 창이 뜹니다. '예'를 누르면 Online이 되는 것을 볼 수 있습니다.



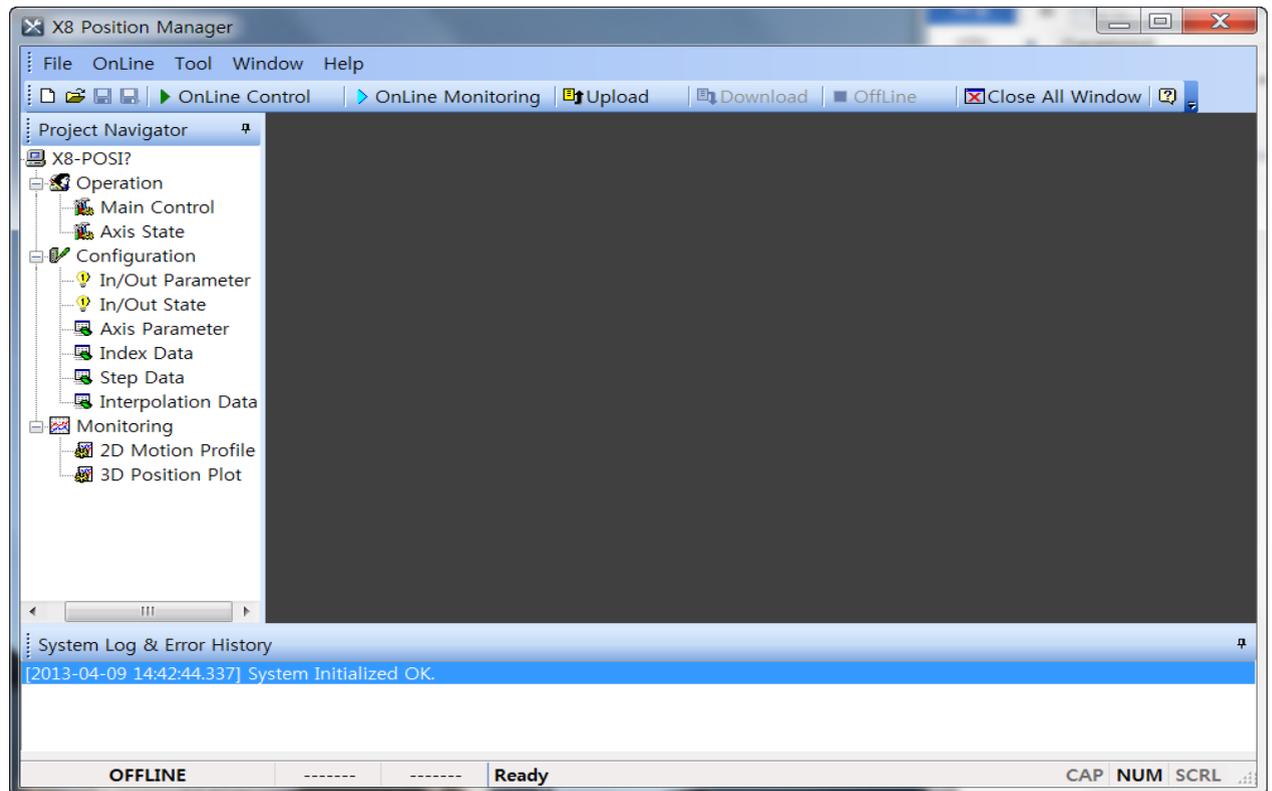
※ Download를 하지 않고, Online을 누르면 본래 있던 PLC의 데이터를 갖고 옵니다.

## Example

### X8-PM소프트웨어로 X8-POSITION MODULE을 이용하여 모션제어

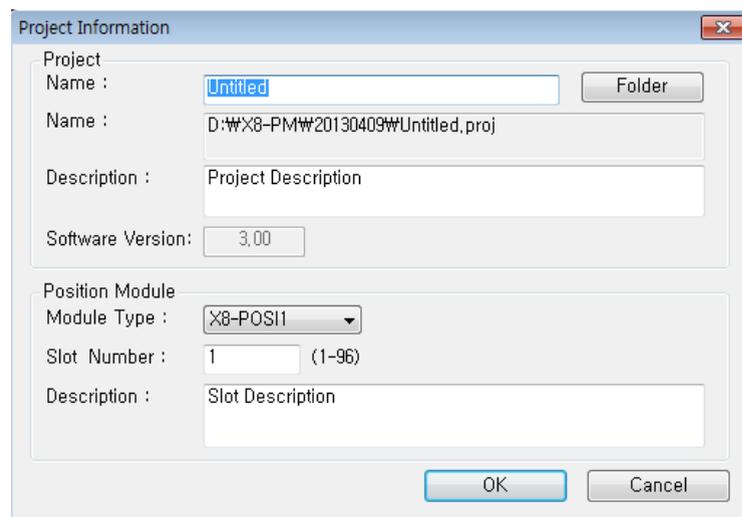
이 장에서는 간단한 E-MOTION제어를 해보겠습니다.

#### Step 1. 소프트웨어 기동



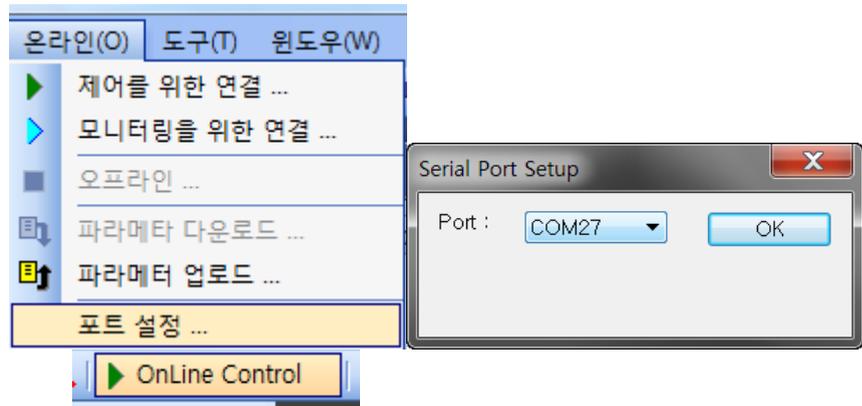
#### New

새로운 프로젝트를 생성합니다.



- Port Setup

현재 Position Module과 연결 되어있는 USB Port를 설정합니다.(온라인메뉴 -> 포트설정)

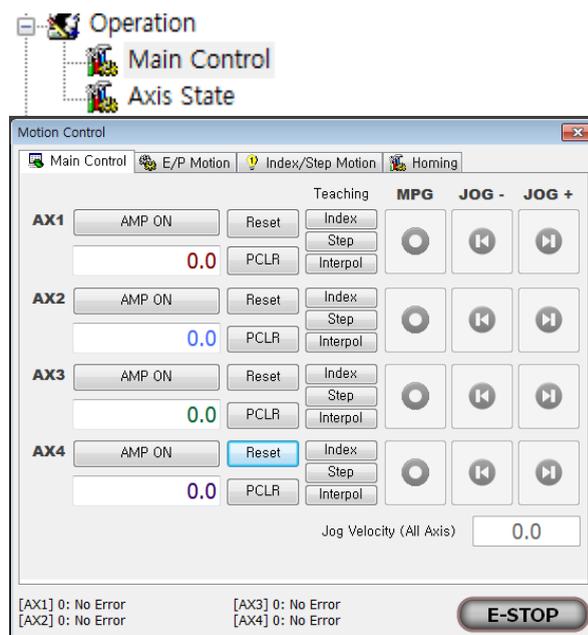


Port Setup이 완료후 버튼을 클릭하시면 온라인상태가 됩니다.



### Step 2. 모션기능 사용

왼쪽 프로젝트 트리에 있는 메인제어(Main Control)버튼을 더블 클릭하시면 메인 컨트롤 창이 나타납니다.

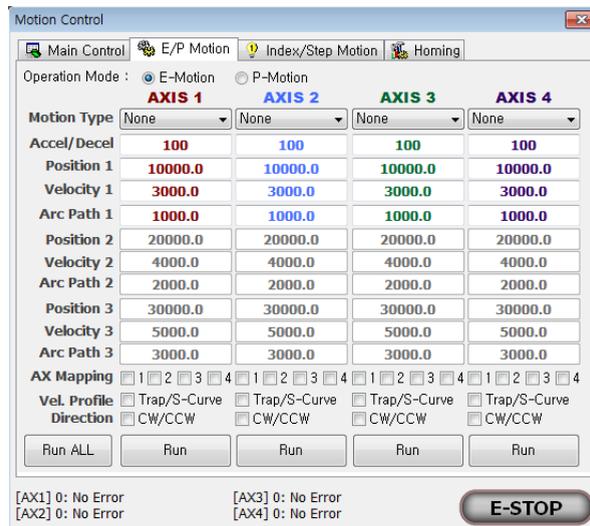


이 탭에서는 JOG기능과 MPG기능,AMP ON,ERROR RESET, PCLR, Teaching이 가능한 창입니다.

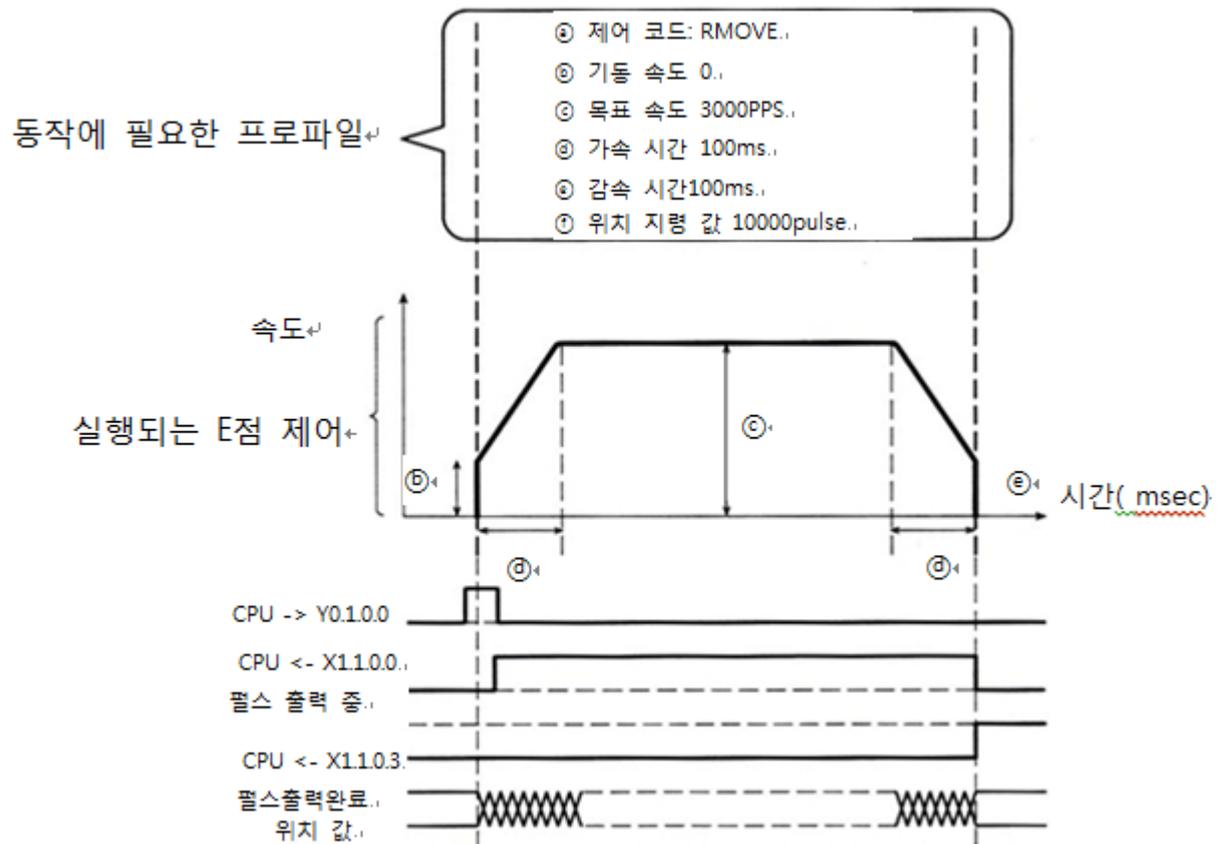
에러명과 현재 CMD PULSE의 상태도 확인 가능합니다.

우선 AMP ON이 완료된 후 작업이 가능합니다.

E/P Motion 탭으로 이동합니다.



Motion Type을 RMOVE로 선택합니다. RMOVE는 상대좌표로 이동을 합니다. AMOVE는 절대치제어입니다. 그리고 ROBOT의 상태를 확인 후 원하시는 축의 RUN버튼을 클릭하시면 됩니다. 현재 모터는 원점을 잡은 상태로 0으로 있는 상태입니다. 프로파일은 Accel/Decel은 100ms이며 이동좌표는 10000펄스,속도는 3000pps입니다.

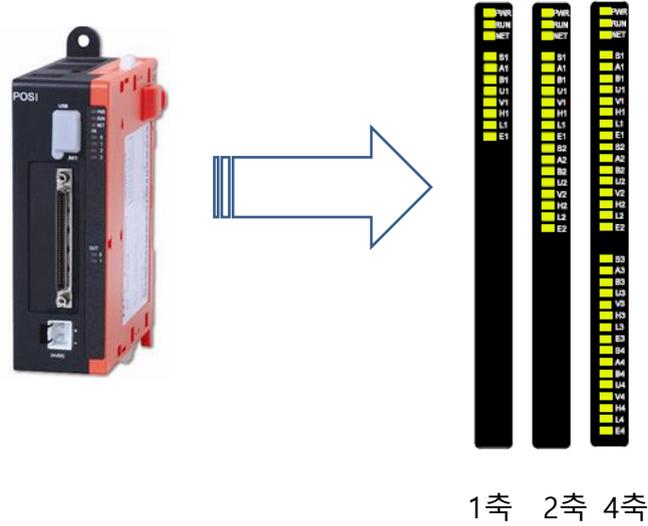


**Step 3. PLC 및 Motor 확인**

모터가 회전하는 것과 LED의 출력을 확인하실 수 있습니다.

LED 표시부

각 축마다 동작상태 LED는 다음과 같습니다,



상태표시 LED

LED 명칭	기능
PWR	전원 입력 상태
RUN	실행 상태
NET	모듈 연결 상태

동작상태 표시 LED

LED 명칭		기능
순서	기호	
0	S	앰프 온 신호 표시
1	A	펄스출력A 신호 표시*1
2	B	펄스출력 B 신호 표시*1
3	U	엔코더 A 신호 표시*2
4	V	엔코더 B 신호 표시*2
5	H	홈신호 표시*3
6	L	리미트 신호 표시*3
7	E	에러발생 표시

## XGPC LADDER 이용하여 CSD5/7 Servo Drive Control

1번 슬롯에 장착된 1번축 기동 파라미터 예제 (기동 명령 : Y0.1.0, E점 제어 기동)

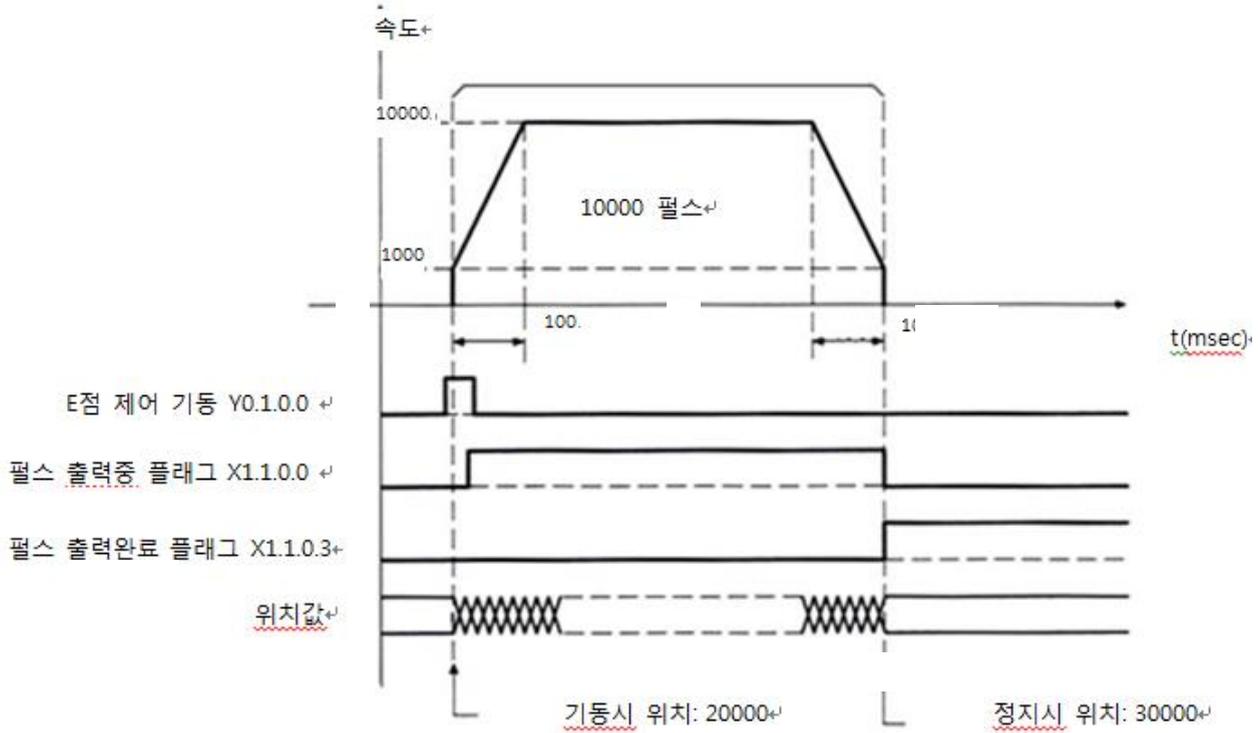
출력데이터 테이블 설정

파라미터 변수	출력데이터 테이블	1속재 설정값
제어코드	Y0.1.1	1
확장 제어코드	Y0.1.2	0
가속 시간[ms]	Y0.1.3	100
감속 시간[ms]	Y0.1.4	100
드웰 시간[ms]	Y0.1.13	0
위치지령값	Y0.1.5/6	+10000.0
기동 속도	Y0.1.7/8	1000.0
목표 속도	Y0.1.9/10	10000.0

사용자 데이터 테이블을 위한 설정

파라미터 변수	데이터 테이블	1속재 설정값	설정 가능 범위
제어코드	N4.0	인크리먼트	1 : 인크리먼트 2 : 엡솔루트
확장 제어코드	N4.1	사다리꼴	0 : 사다리꼴 1 : S커브
가속 시간 [ms]	N4.2	100	32,767
감속 시간 [ms]	N4.3	0	32,767
드웰 시간 [ms]	N4.4	0	P점 제어에서는 사용하지 않음
위치지령값	F12.0	+10000.0	-2,147,483,648 ~ 2,147,483,647
기동 속도	F12.0	1000	0~1,000,000
목표 속도	F12.2	10000.0	1~1,000,000

주의 : 목표위치, 시작속도, 최대속도는 Float 형 데이터임 (2 위드 크기)



### 각 플래그의 동작

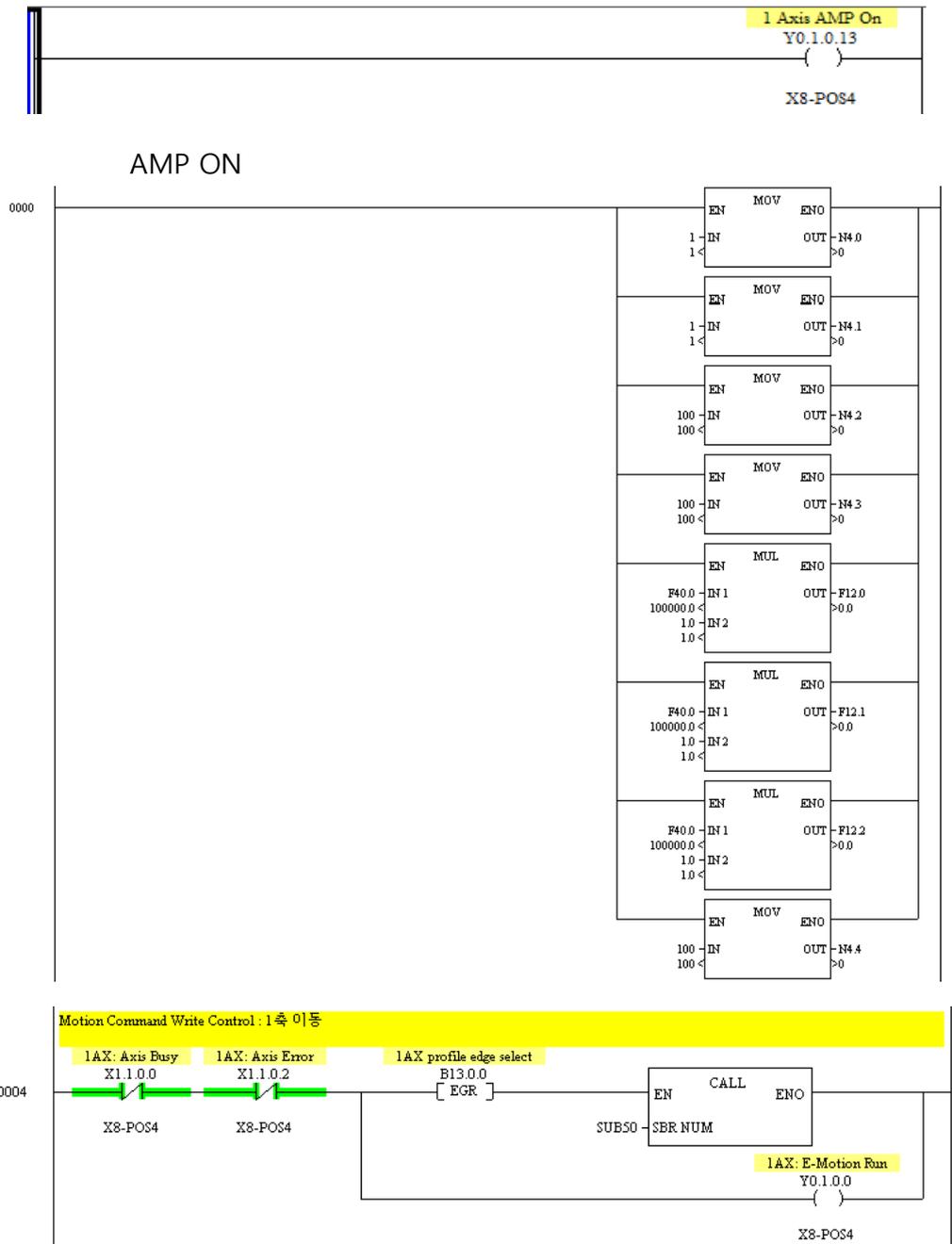
펄스 출력중 플래그(X1.1.0.0)는, E점 제어 기동시에 ON되고 펄스 출력이 완료되면 OFF가 됩니다.

펄스출력 완료 플래그(X1.1.0.3)는, 펄스출력이 완료되면 ON되고 다음으로 E점 제어, P점 제어, JOG운전, 원점 복귀, 펄스발생기 입력 허가 중 한가지라도 동작이 기동될 때까지는 유지됩니다.

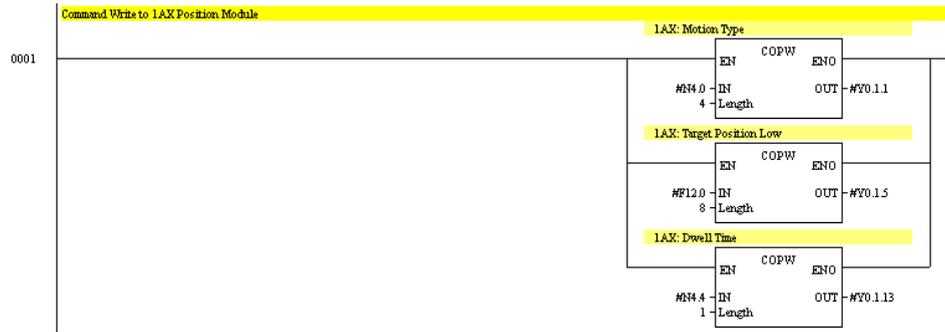
위치값은 위치결정 유니트의 내부 카운터에 절대값으로 보관됩니다.

프로그램 예제

프로파일 입력 및 기동



출력데이터 테이블에 WRITE 프로그램



프로그램상의 주의

각 제어 파라미터를 입력하는 출력데이터 테이블은 가감속 제어, JOG운전, 원점복귀등 다른 제어와 공통으로 사용됩니다. 다른 조건으로 덮어쓰게 되지 않도록 하십시오.

기동속도, 목표속도, 가감속 시간, 위치 지령값의 각각의 값이 설정 가능범위를 넘을 경우, 설정값 에러가 발생하여 기동되지 않습니다.

기동 접점번호는 유니트의 축수, 장착위치에 따라 변합니다  
지정할 슬롯 번호, 출력 데이터 테이블 어드레스는 위치결정 유니트의 슬롯 위치, 축 번호에 따라 변합니다.

## 11장. X8 PLC통신

본 Application Note는 RSA의 제품인 NX-Plus와 NX-Alpha PLC를 X8과 통신하는 방법을 설명합니다. X8 PLC는 모두 Serial 통신으로 NX-Plus Master와 NX-Alpha Master를 지원합니다.

---

PLC 통신 ..... 11-2

Configuration ..... 11-3

Example ..... 11-9

## PLC 통신



위 그림은 본 Application Note에서 예제로 사용되는 제품의 개략도입니다. X8이 Master로 설정되고, 상대 제품(NX-Plus/ NX-Alpha)은 Slave로 설정됩니다. X8이 상대 제품의 데이터를 자유자재로 Read/Write 할 수 있습니다.

### Target Applications

본 Application Note는 X8이 지원하는 NX-Plus Master와 NX-Alpha Master 프로토콜을 이용하여 기존에 설치된 NX 시스템의 데이터를 Read/Write 할 수 있습니다.

### 실행조건

- 1) XGPC로 X8 Programming
- 2) WINGPC로 NX-Plus PLC Programming
- 3) WINFPST로 NX-CPU750A/B/C/D PLC Programming

### Test Environment & Equipment

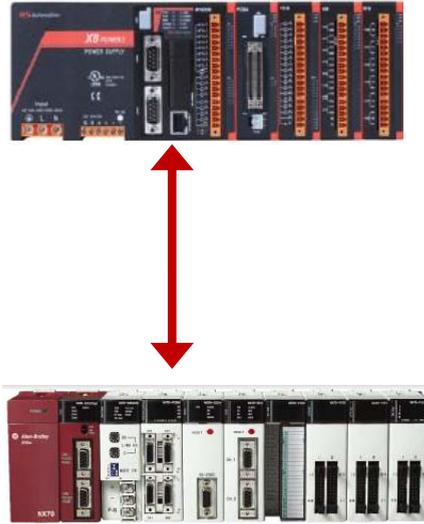
- 1) X8 PLC, NX-CPUplus PLC, NX-CPU750 PLC
- 2) XGPC 1.1.1.0 Software(X8 PLC 전용), WinGPC 4 Software(NX-Plus PLC 전용),
- 3) WinFPST 2 Software(NX-Alpha PLC 전용), PC window XP/ 7 32bit/64bit
- 4) 232/485 Serial Cable 2개

### Related Publication

X8 User Manual, NX-Plus PLC User Manual, NX-Alpha PLC User Manual

(홈페이지에서 제공. <http://www.rsautomation.co.kr>)

## Configuration



## Wiring

Case 1. X8과 NX750(또는 NX-Alpha Series)를 연결할 경우

X8 Female D-sub 9Pin			NX750 Male D-sub 9Pin	
DCD	1		1	F.G
RXD	2		2	TXD
TXD	3		3	RXD
485+	4		4	-
GND	5		5	SG
485-	6		6	-
RTS	7	┌───┐	7	RTS
CTS	8	└───┘	8	CTS
NC	9		9	ER

X8의 통신포트 핀맵은 컴퓨터와 동일합니다. NX750과 PC와 연결하는 Cable을 사용하시면 됩니다.

**Case 2. X8과 NX-CPU700/70p(1/2)를 연결할 경우**

파란선은 RS232C 연결 핀 맵 이고, 빨간선은 RS485 연결 핀 맵 입니다.

X8 Female D-sub 9Pin			NX-CPU Male D-sub 9Pin	
DCD	1		1	F.G
RXD	2	—————	2	TXD
TXD	3	—————	3	RXD
485+	4	—————	4	-
GND	5	—————	5	SG
485-	6	—————	6	485-
RTS	7	———	7	485+
CTS	8	———	8	-
NC	9		9	Vcc

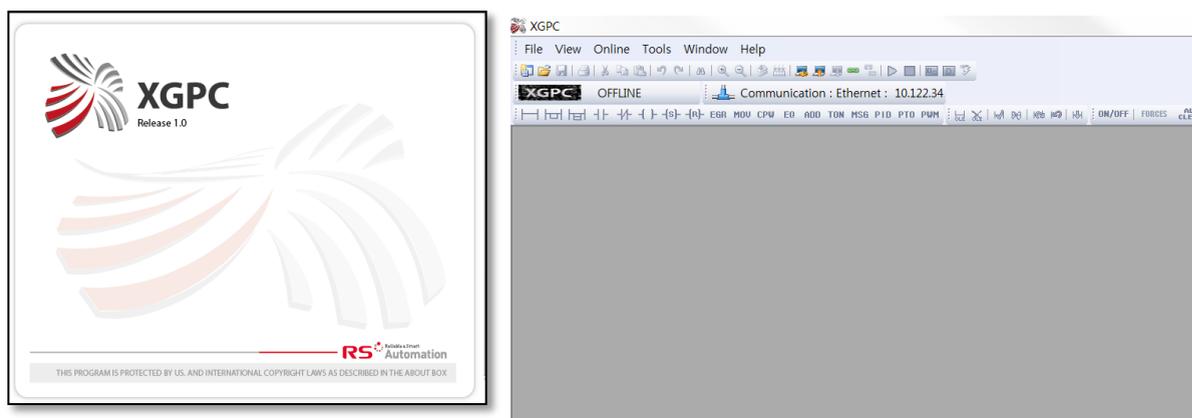
X8의 포트 핀맵은 동일합니다. PC와 NX-CPU와 PC와 연결하는 Cable을 사용하시면 됩니다.

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

NX-CPU는 RS232C/RS485 통신 설정을 DIP Switch를 이용하여 변경합니다.

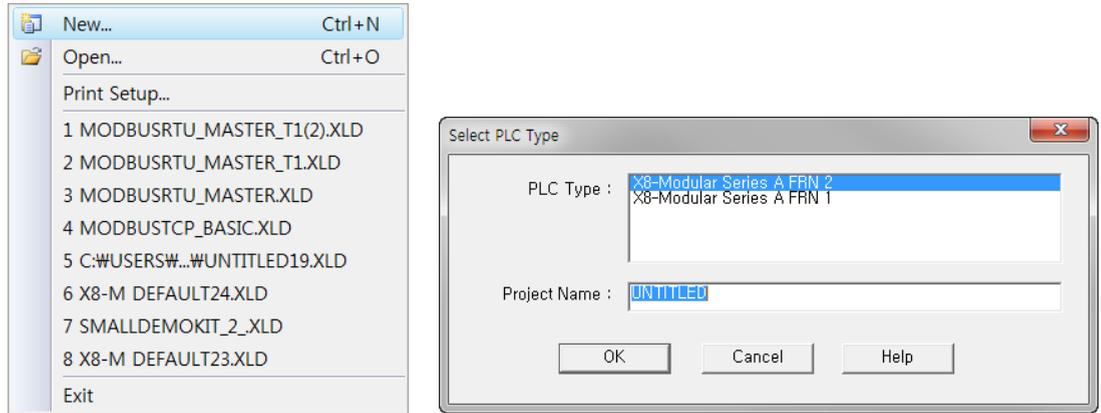
**X8 Setting**

Step 1. X8 전용 래더 소프트웨어인 XGPC를 설치하고 실행.



**Step 2. 새 프로젝트 생성**

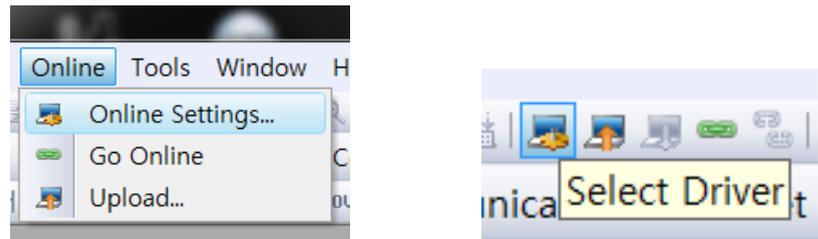
파일 메뉴에서 New 버튼을 누릅니다. PLC type 창에서 X8 하드웨어 모듈을 선택하고, 프로젝트 명을 입력합니다. NX-Alpha Master 설정은 X8 PLC의 FRN(Firmware Revision Number) 2 또는 그 이상부터 지원됩니다.



**Step 3. Online Settings**

Online Settings에서는 연결하고자 하는 PLC의 설정을 입력하여, 해당하는 PLC를 찾아 프로젝트를 Upload 하거나 XGPC와 PLC를 Online 하는 기능입니다.

Online 메뉴에서 Online Settings, 또는 아이콘에서 Online Settings(Select Driver)를 클릭합니다.



**Case 1. Serial 또는 USB로 연결**

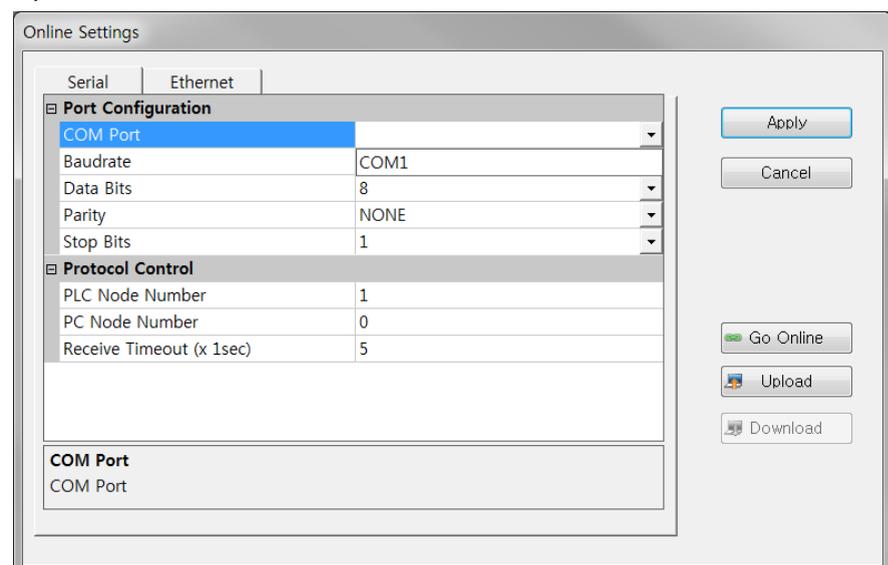
Serial 또는 USB로 연결할 경우, Online Settings 창의 Serial 탭으로 들어갑니다. USB로 연결하였을 때 Serial 탭으로 들어가는 이유는, USB Driver가 통신 포트를 Serial Port로 매핑(mapping)하여 연결하기 때문입니다.

Online Settings 창에 들어가면, 자동으로 COM port 번호가 인식되어 나타납니다. PLC에 적혀 있는 COM0, COM1이 아니라, 컴퓨터에서 인식한 COM port번호를 선택합니다.

COM Port 번호는 내 컴퓨터 > 속성 > 장치관리자로 들어가면 확인할 수 있습니다.



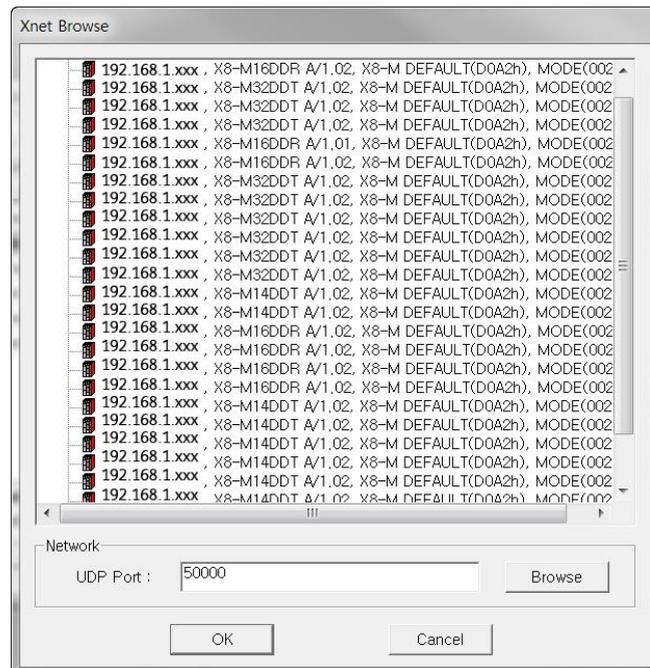
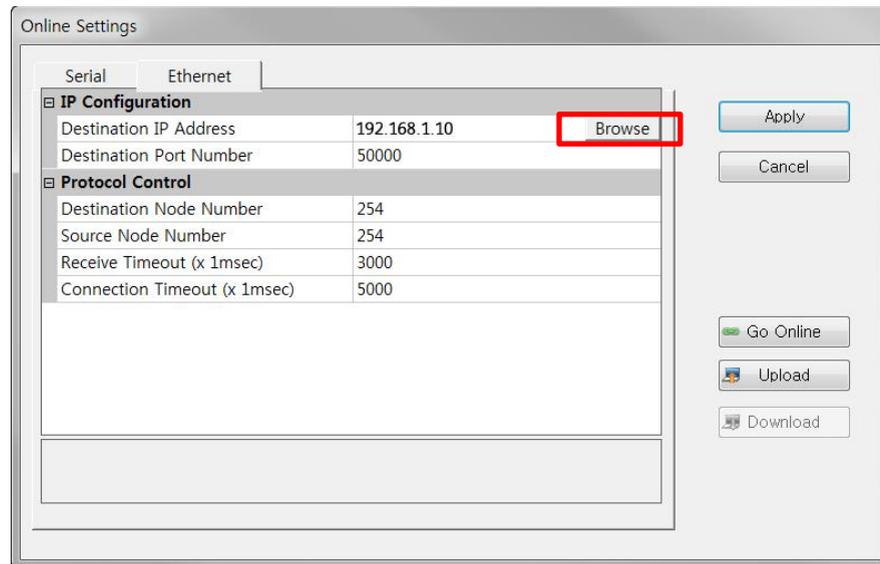
PLC와 연결된 Computer의 COM port번호를 선택 한 후, Apply 를 누릅니다. 이 창에서 Go Online 또는 Upload를 누르면, 바로 PLC와 연결하여 데이터를 가지고 오거나 Online을 할 수 있습니다.



### Case 2. Ethernet으로 연결

Ethernet으로 PLC와 연결할 경우, Ethernet 탭으로 들어갑니다. Destination IP Address에 연결하고자 하는 PLC의 IP주소를 입력 합니다.

컴퓨터와 같은 Network 상으로 연결되어 있는지 확인하고 싶은 경우, 오른쪽의 'Browse'버튼을 클릭하여 연결하고자 하는 PLC의 IP를 선택할 수도 있습니다.

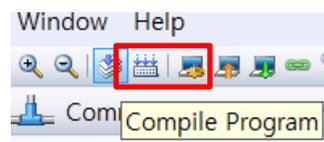


Computer와 연결할 PLC의 IP를 선택 또는 입력한 후, Apply를 누릅니다. 이 창에서 Go Online 또는 Upload를 누르면, 바로 PLC와 연결하여 데이터를 가지고 오거나 Online을 할 수 있습니다.

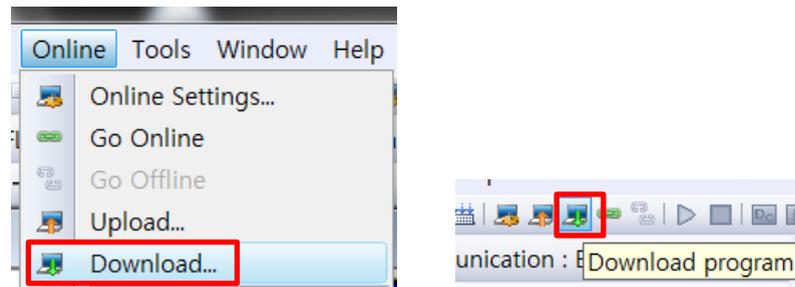
#### Step 4. PLC Online 및 Download

XGPC에서 Programming을 끝나치고 난 후, Compile 버튼을 실행하여 Programming상 오류가 있는지 확인합니다. Compile 버튼은 Compile Program과 Compile Project가 있습니다.

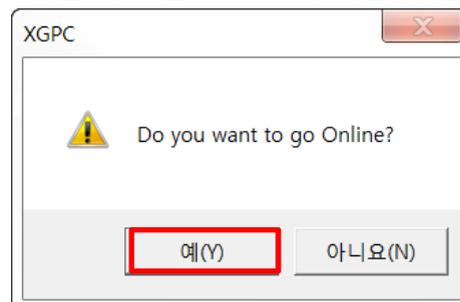
- Compile Program : 현재 Programming한 Ladder Program만 Compile
- Compile Project : 현재 열려 있는 Project에 있는 모든 Ladder Program과 설정들을 Compile



Compile이 완료된 후, 작성한 Project를 PLC로 전송하기 위해 Download 버튼을 누릅니다. Download 버튼은 Online 메뉴에서 클릭하거나, 아이콘 Toolbar에서 사용할 수 있습니다.



Download를 하고 나면, Online을 할 것인지 메시지 창이 뜹니다. '예'를 누르면 Online이 되는 것을 볼 수 있습니다.



※ Download를 하지 않고, Online을 누르면 본래 있던 PLC의 데이터를 가지고 옵니다.

## Example

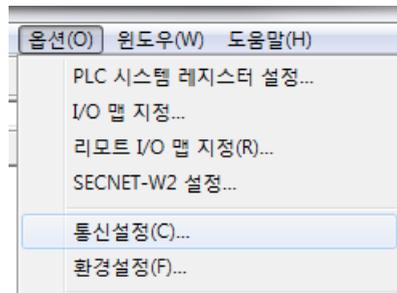
### NX750(또는 NX-Alpha Series)를 연결할 경우

#### Step 1. 통신 설정 변경

X8 PLC와 NX-Alpha Series PLC(이하 NX-Alpha PLC)가 연결되었을 때는 X8이 Master가 되고, NX-Alpha PLC가 Slave가 됩니다. X8 PLC와 NX-Alpha PLC가 연결하기 위해서는 서로의 통신 설정이 같아야 합니다.

#### NX-Alpha PLC

NX-Alpha PLC 전용 프로그램인 WinFPST 4를 실행합니다. X8과 연결하려고 하는 NX-Alpha PLC를 연결하고 옵션의 통신설정으로 들어갑니다.



다음 그림은 NX-Alpha PLC의 RS232C Default 통신 설정입니다.

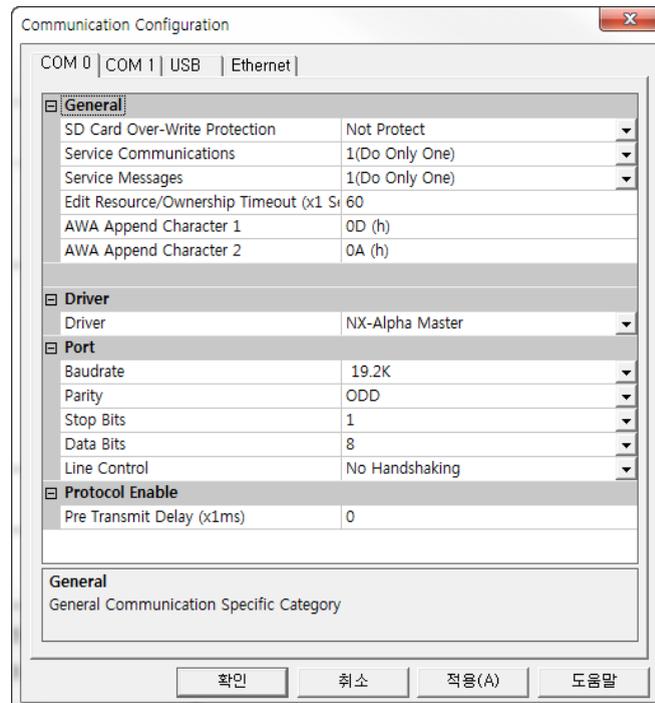


파라미터	값
Baud rate	57600 bps
Data Bits	8
Stop Bits	1
Parity	ODD

위와 같이 설정한 것을 확인하고, 이와 같은 설정을 NX-Alpha PLC로 Download 합니다.

## X8 PLC

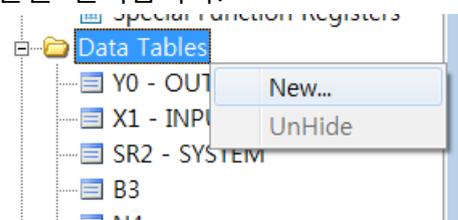
X8 PLC의 통신 설정을 변경하기 위해 Project 창의 Communication Configuration(통신 설정)으로 들어갑니다. 본 Note에서는 Com0 포트에서 통신 설정을 변경 하도록 하겠습니다.



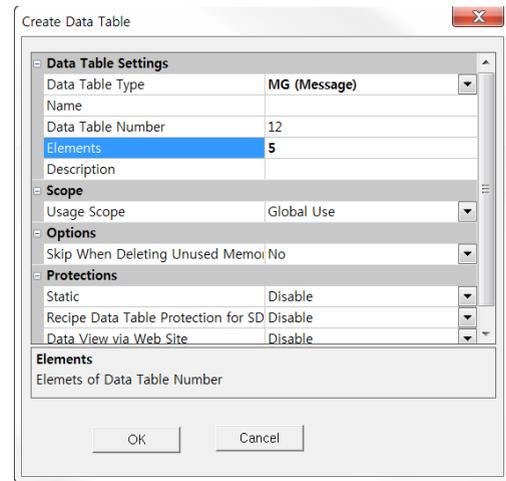
Driver를 NX-Alpha Master로 변경합니다. Baud rate와 Parity, Stop Bits, Data Bits를 NX-Alpha PLC의 Default 통신 설정과 동일하게 설정합니다.

### Step 2. XGPC에서 MSG 생성

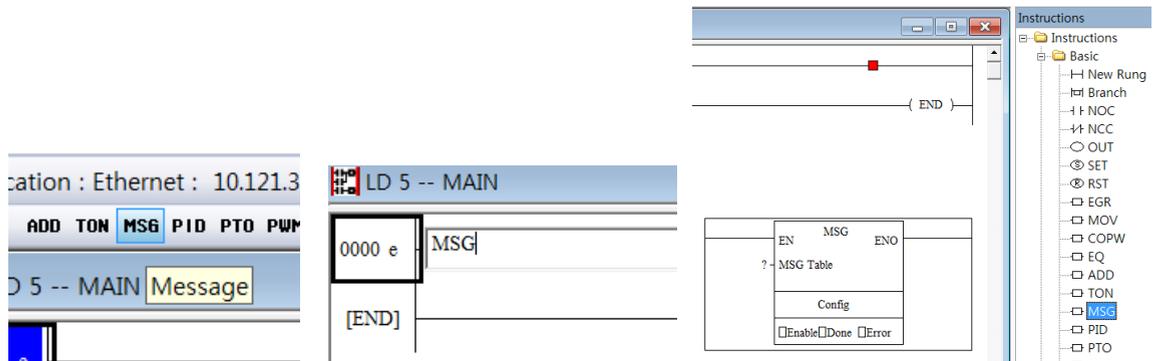
X8 PLC가 Master일 경우에는 MSG 명령어를 사용합니다. MSG 명령어를 사용하기 위해 Data table에 MG Data table을 생성해야 합니다. Data table 생성 및 삭제는 XGPC가 Offline 상태일 때만 가능합니다. Data Table 폴더를 마우스의 오른쪽 버튼으로 클릭하여 New 버튼을 선택합니다.



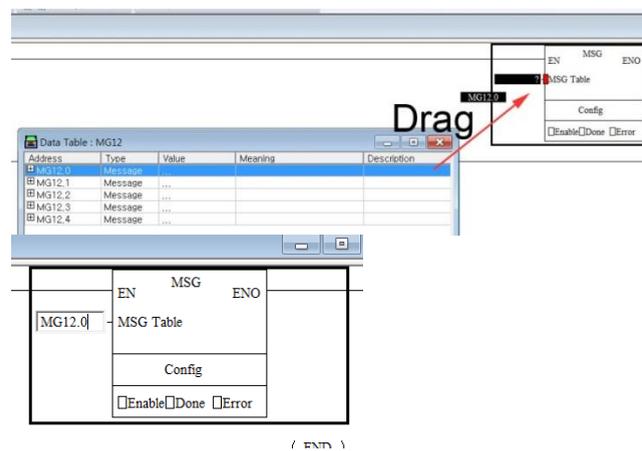
Create Data Table 창이 열리면 Data Table Type을 MG(Message)로 선택합니다. Data Table Number는 자동으로 Data Table에 없는 번호로 지정됩니다. 또는 최대 1535번까지 사용자가 번호를 선택할 수 있습니다. Element는 MSG 명령어를 사용할 개수만큼 생성합니다. 하나의 MG Data Table 당 1022개의 Element를 생성할 수 있습니다.



NX-Alpha Data를 Read할 부분에 MSG 명령어를 생성합니다. MSG 명령어는 명령어 Toolbar에서 선택/ 키보드로 입력 또는 Instruction window에서 드래그 하여 생성할 수 있습니다.



MSG 명령어에 생성한 MG(Message) Data Table을 입력 또는 드래그 합니다.

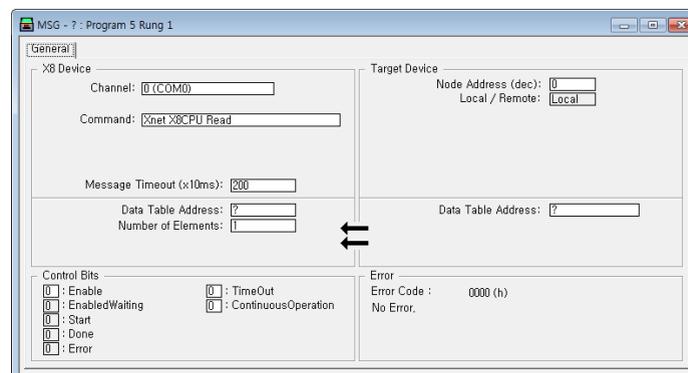


Step 3. X8이 지원하는 NX-Alpha Command와 Operand  
X8이 지원하는 NX-Alpha Command와 Operand 입니다. Command는 Read와 Write로 나뉘며 Bit와 Word에 따라 구분 지을 수 있습니다.

Command	Target Address	Local Address Example
	Target Address Example	
RCP(Read Bits)(01)	X,Y,R,L,T,C	B3.0.0, N4.0.0
	X100A, Y100A, R100A, L100A, T100, C100	
WCP(Write Bits) (02)	Y,R,L	B3.0.0, N4.0.0
	Y100A, R100A, L100A	
RCC(Read Words) (03)	WX, WY, WR, WL, T, C	B3.0, N4.0, F5.0, L6.0
	WX100, WY100, WR100, WL100, T100, C100	
WCC(Write Words) (04)	WY, WR, WL	B3.0, N4.0, F5.0, L6.0
	WY100, WR100, WL100	
RD(Read Words) (05)	DT, FL, LD, IX, IY, IXIY	B3.0, N4.0, F5.0, L6.0
	DT100, FL100, LD100, IX, IY, IXIY	
WD(Write Words) (06)	DT, FL, LD, IX, IY, IXIY	B3.0, N4.0, F5.0, L6.0
	DT100, FL100, LD100, IX, IY, IXIY	
RS(Read T/C SV) (07)	SV	B3.0, N4.0
	SV100	
WS(Read T/C SV) (08)	SV	B3.0, N4.0
	SV100	
RK(Read T/C EV) (09)	EV	B3.0, N4.0
	EV100	
WK(Write T/C EV) (0A)	EV	B3.0, N4.0
	EV100	

#### Step 4. NX-Alpha Word Data Read 설정

MSG의 상세 설정을 위해 MSG 명령어 하단의 Config를 더블 클릭합니다.



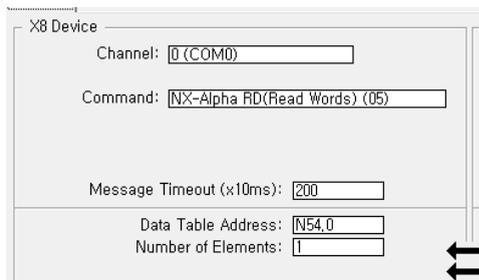
Channel은 X8이 연결하려고 하는 장치가 직접적/간접적으로 연결되어 있는 통신 포트를 설정하는 곳입니다.

Command는 X8이 연결한 장치와 프로토콜에서 정의한 명령 코드를 설정하는 곳입니다.

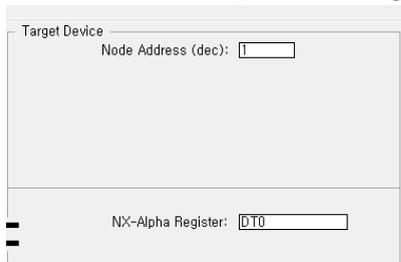
Data Table Address는 NX-Alpha PLC에서 읽을 데이터가 저장될 X8 어드레스를 입력합니다. Number of Elements는 어드레스에 입력되는 데이터 단위로 적습니다. 만약 Data Table Address에 L6.0을 쓰고 읽을 데이터가 L6.0, L6.1로 저장하기 위해선 Number of Element를 2로 입력합니다.

X8 Device 부분을 살펴 보겠습니다. Channel은 NX-Alpha PLC와 연결되어 있는 Com0 포트로 설정합니다. Command는 "NX-Alpha RD(Read Words) (05)"로 설정합니다.

아래의 그림에서는 NX-Alpha PLC에서 읽는 하나의 Word가 X8의 N54.0에 저장합니다.



Target Device 부분을 살펴 보겠습니다. Node Address는 X8과 연결하는 NX-Alpha PLC의 국번을 입력합니다. NX-Alpha Register는 X8이 읽을 NX-Alpha PLC Register (Address)를 입력합니다.



MSG Control Bits는 다음과 같습니다. 일반적으로 MSG는 한번 실행하도록 되어 있습니다. 오른쪽하단의 ContinuousOperation 비트를 ON으로 설정하면, 해당하는 MSG 명령어가 지속적으로 실행됩니다.



### Step 5. NX-Alpha Word Data Write 설정

새로운 MSG 명령어를 생성하고 MG12.1을 MSG 명령어에 입력합니다. 상세 설정을 하기 위해 Config를 클릭합니다(Step 2,3 참고). Channel은 "0(Com0)", Command는 "NX-Alpha WD(Write Word) (06)"으로 설정합니다.

아래 그림에서는 X8의 N55.0의 데이터가 NX-Alpha PLC로 전송됩니다.

X8 Device

Channel:

Command:

Message Timeout (x10ms):

Data Table Address:

Number of Elements:

Target Device 부분을 살펴 보겠습니다. Node Address는 X8과 연결하는 NX-Alpha PLC의 국번을 입력합니다. X8의 N55.0의 데이터가 저장될 NX-Alpha PLC Register (Address)를 입력합니다.

Target Device

Node Address (dec):

NX-Alpha Register:

Step 4와 같이 지속적인 MSG 실행을 하기 위해서 Control Bits의 ContinuousOperation 비트를 On으로 설정합니다.

### Step 6. X8과 NX-Alpha PLC 연결 확인

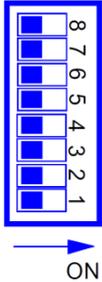
XGPC에 설정한 프로젝트를 X8 PLC로 다운로드 합니다. X8의 Com0포트와 연결된 NX-Alpha PLC의 모드를 확인합니다. X8의 MSG 명령에 따라 NX-Alpha PLC의 데이터가 Read/Write 되는 것을 확인할 수 있습니다.

### NX-CPU(또는 NX-Plus Series)를 연결할 경우

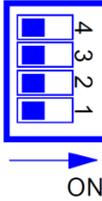
Step 1. 통신 설정 변경

#### NX-Plus PLC

NX-CPU(이하 NX-Plus)의 통신 설정은 Software에서 하지 않고, 외부에 있는 DIP Switch 통해 변경합니다.

PIN 번호		스위치		기능	DIP Switch2
8	7	OFF	ON	COM2단자 통신속도 38400 bps	
		ON	OFF	COM2단자 통신속도 19200 bps	
		OFF	OFF	COM2단자 통신속도 9600 bps	
		ON	ON	COM2단자 통신속도 4800 bps	
6	5	OFF	ON	COM1단자 통신속도 38400 bps	
		ON	OFF	COM1단자 통신속도 19200 bps	
		OFF	OFF	COM1단자 통신속도 9600 bps	
		ON	ON	COM1단자 통신속도 4800 bps	
4	ON		COM1의 RS485 방식 선택		
	OFF		COM1의 RS232C 방식 선택		
3	ON		COM2의 RS485 방식 선택		
	OFF		COM2의 RS232C 방식 선택		
2	OFF		항상 OFF로 고정 시킴 (시스템 설정용)		
	1	ON		전원 ON시 EEPROM (Flash ROM)에서 프로그램 램 Load	
OFF		전원 ON시 RAM으로 프로그램 운전			

또한 RS485 통신할 시, 종단저항을 DIP Switch로 설정할 수 있습니다.

PIN 번호		스위치		기능	DIP Switch1
4	3	ON	ON	RS485 통신시 : 종단국일 때 ON (COM1 단자에 종단 저항 설정)	
		OFF	OFF	RS485 통신시 : 종단국이 아닐 때 ON (COM1 단자에 종단 저항 미설정)	
2	1	ON	ON	RS485 통신시 : 종단국일 때 ON (COM2 단자에 종단 저항 설정)	
		OFF	OFF	RS485 통신시 : 종단국이 아닐 때 ON (COM2 단자에 종단 저항 미설정)	

파라미터	값
Baud rate	19200 bps
Data Bits	8
Stop Bits	1
Parity	NONE

NX-Plus의 설정은 다음과 같습니다.

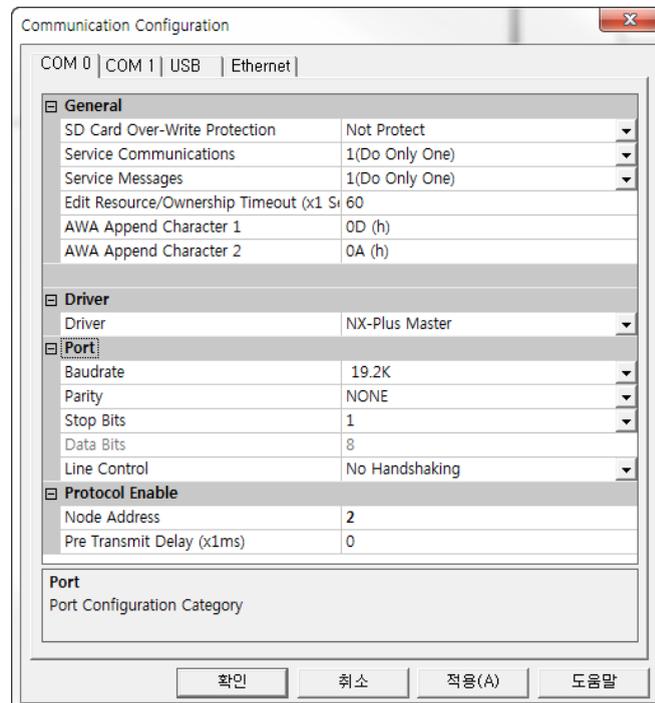
DIP Switch 2의 6번을 ON으로 변경합니다. NX-Plus의 통신 설정이 위의 표와 같은 지 확인 합니다.

WinGPC와 NX-Plus를 연결하여 Node Address(국번)를 확인합니다. 본 Application Note에서는 1로 설정합니다.

### X8 PLC

X8 PLC의 통신 설정을 변경하기 위해 Project 창의 Communication Configuration(통신 설정)으로 들어갑니다.

본 Note에서는 Com0 포트에서 통신 설정을 변경 하도록 하겠습니다.



Driver를 NX-Plus Master로 설정하고, NX-CPU와 같도록 Baudrate, Parity, Stop Bits를 선택합니다. 본 Application Note에서는 NX-Plus와 X8이 RS232C로 연결합니다. RS485로 연결할 경우, X8의 통신 설정 중 Line Control을 "No Handshaking (RS485 Network)" 로 변경합니다.

본 Application note에서는 X8 PLC의 Node Address를 2로 설정합니다. XGPC에서 MSG 생성하는 부분은 예제 3.1의 Step 2를 참조 하세요.

**Step 2. X8이 지원하는 NX-Plus Command와 Operand**

X8이 지원하는 NX-Plus Command와 Operand 입니다. Command는 Read와 Write로 나뉘며 Bit와 Word에 따라 구분 지을 수 있습니다.

Command	Target Address	Local Address
	Target Address Example	Example
Read Bits (21)	R,L,M,K	B3.0.0, N4.0.0
	R0.0, L0.0, M0.0, K0.0	
Write Bits (22)	R,L,M,K	B3.0.0, N4.0.0
	R0.0, L0.0, M0.0, K0.0	
Read Words (23)	R,L,M,K,F,W,SV,PV,SR,D	B3.0, N4.0, F5.0, L6.0
	R0, L0, M0, K0, ...	
Write Words (24)	R,L,M,K,F,W,SV,PV,SR,D	B3.0, N4.0, F5.0, L6.0
	R0, L0, M0, K0, ...	

**Step 3. NX-Plus Bits Data Read 설정**

새로운 MSG 명령어를 생성하고, MG12.2 Data table을 MSG의 Operand로 입력합니다. 생성한 MSG 상세 설정을 위해 명령어 하단의 'Config'를 더블 클릭합니다.

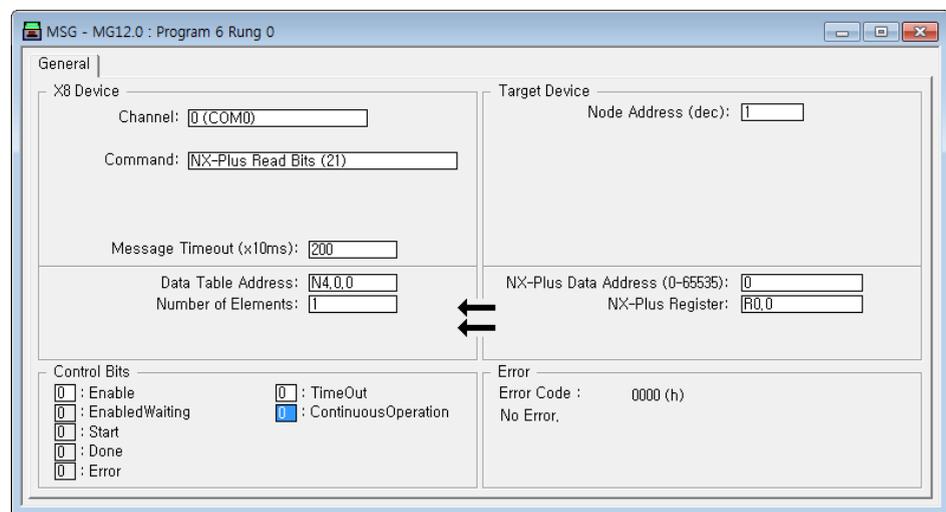
아래 그림과 같이 Channel은 COM0로 설정합니다. X8의 COM0는 NX-Plus와 연결되어 있는 Serial Port 입니다. Command는 "NX-Plus Read Bits (21)"로 설정합니다.

Target Device에는 X8과 연결하는 NX-Plus Node Address를 입력합니다. Step 1의 1-1에서 설정한 Node Address를 입력합니다. NX-Plus Data Address (0-65535) 또는 NX-Plus Register(Address)를 입력합니다. 본 Application Note에서는 R0.0 비트 값을 읽어 오도록 하겠습니다.

X8 Device에서 Data Table Address는 N4.0.0으로 입력합니다. 읽어오는 NX-Plus의 R0.0 비트는 N4.0.0으로 전송됩니다.

Number of Elements는 R0.0 하나의 비트가 전송되어 1로 설정합니다. 만약 비트가 32개, 즉 워드로 2Word가 X8로 전송 될 경우, Number of Elements는 2로 설정합니다.

MSG 명령어는 기본적으로 한번만 실행 됩니다. Control Bits의 ContinuousOperation 비트를 Off에서 On으로 변경하면 메시지가 지속적으로 수행됩니다.



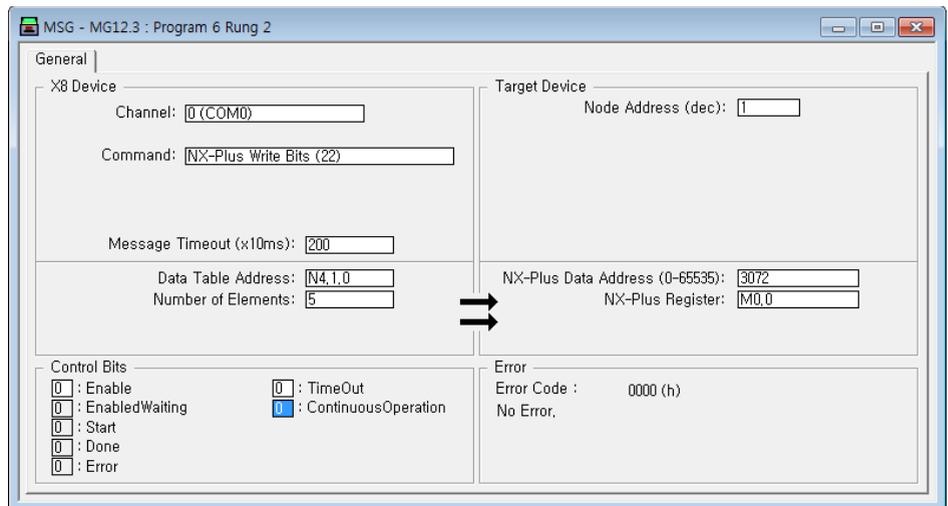
#### Step 4. NX-Plus Bits Data Write 설정

새로운 MSG 명령어를 생성하고, MG12.3 Data Table을 MSG의 Operand로 입력합니다. 상세 설정을 위해 명령어 하단의 'Config'를 더블 클릭합니다.

Channel은 NX-Plus와 연결되어 있는 '0(COM0)', Command는 'NX-Plus Write Bits (22)' 로 설정합니다.

Target Device에는 X8과 연결하는 NX-Plus의 Node Address와 전송될 NX-Plus의 Register(Address)를 입력합니다. 본 Application Note는 M0.0으로 설정합니다.

X8 Device에는 NX-Plus로 전송시킬 X8 Data Table Address와 전송 개수를 입력합니다. 아래의 그림에서는 N4.1.0부터 총 5개의 Word인 N4.1.4까지 데이터를 M0.0부터 M0.4까지 전송합니다.



지속적인 MSG 명령어를 수행하기 위해 Control Bits의 ContinuousOperation 비트를 OFF에서 ON로 변경합니다.

### Step 5. X8과 NX-Plus PLC 연결 확인

X8에 설정한 프로젝트를 다운로드 합니다. X8의 COM0포트와 연결된 NX-Plus PLC의 모드를 확인합니다. X8의 MSG 명령에 따라 NX-Plus PLC의 데이터가 Read/Write 되는 것을 확인할 수 있습니다.





## 12장. X8 CPU EtherNet DATA

### SYNC기능

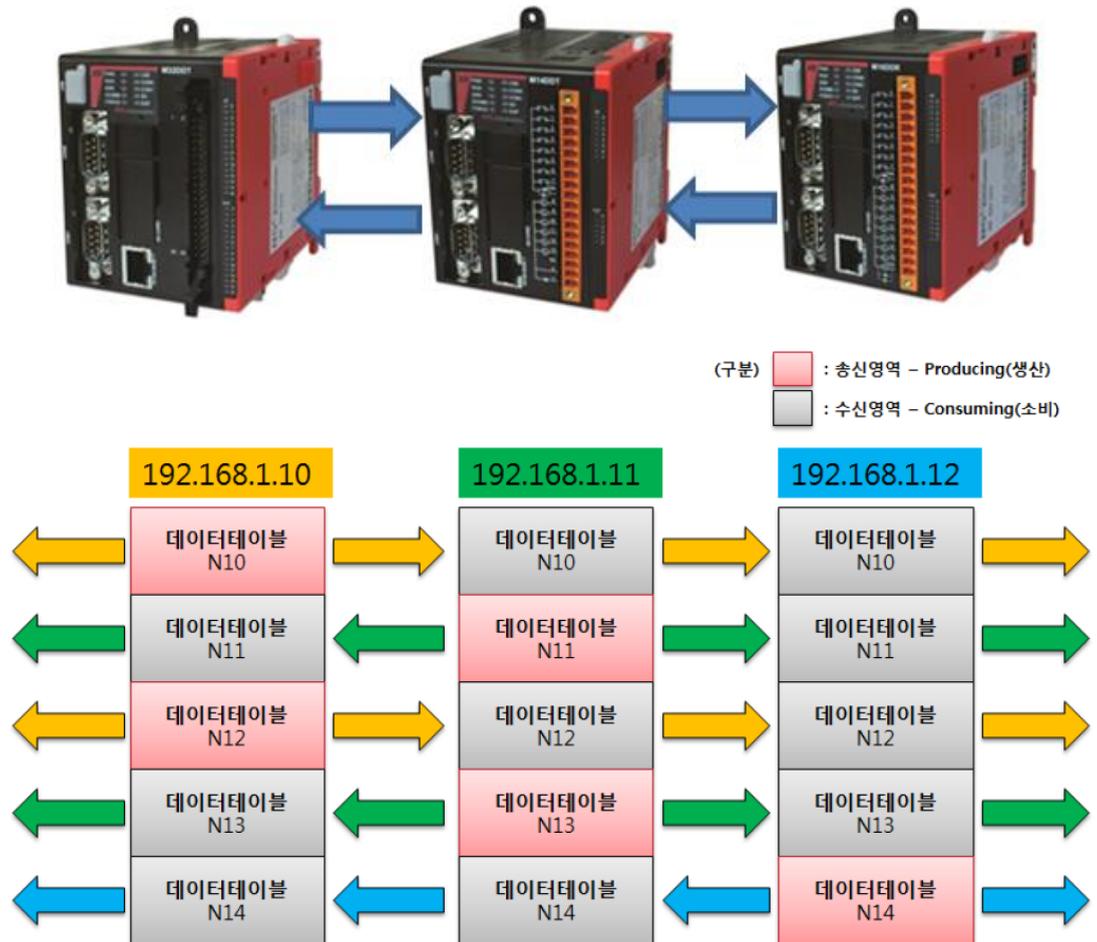
본 장에서는 X8 CPU 끼리 설정만으로 데이터 통신을 할 수 있는 DATASYNC기능을 설명합니다.

---

개요.....12-2

Configuration.....12-3

## 개요



이더넷을 이용하여 X8 CPU 끼리 데이터를 공유하는 기능입니다. X8 CPU 사이의 데이터를 교환하기 위해서는 보통 MSG 명령어를 사용합니다. 그러나, Data Sync에서는 래더가 없이 SR2레지스터의 설정에 의해서 이루어지게 됩니다. Data Sync기능은 생산자/소비자(Producer/Comsumer) 형태로 동작합니다. 또한 멀티캐스트 형태로 이더넷 네트워크에 사용됩니다.

### 목표(Target Applications)

X8 PLC의 DATASYNC 이용하여 데이터를 공유 할 수 있습니다.

### 실행조건

- 1) XGPC로 X8로 이더넷 연결
- 2) X8 CPU 셋팅
- 3) X8 DATA공유

## Test Environment & Equipment

### X8 PLC

XGPC 1.8.2 Software(X8 PLC 전용)

EtherNet Cable 4개

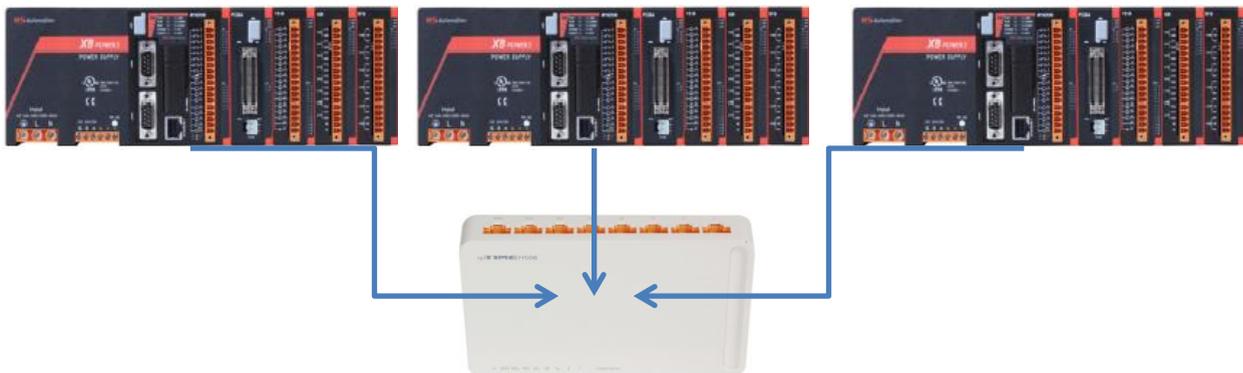
EtherNet 스위칭허브

## Related Publication

X8 User Manual

(홈페이지에서 제공. <http://www.rsautomation.co.kr>)

## Configuration



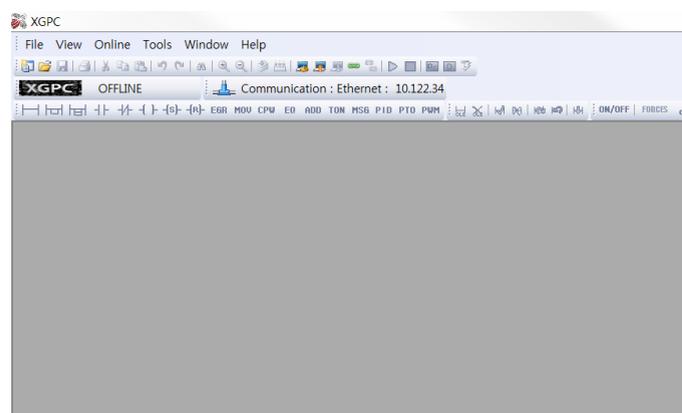
## Wiring

### Case 1. X8과 스위칭 허브와 연결할 경우

X8의 이더넷 포트 핀맵은 일반 랜케이블과 동일합니다. 시중에 판매되는 이더넷 Cable을 사용하시면 됩니다.

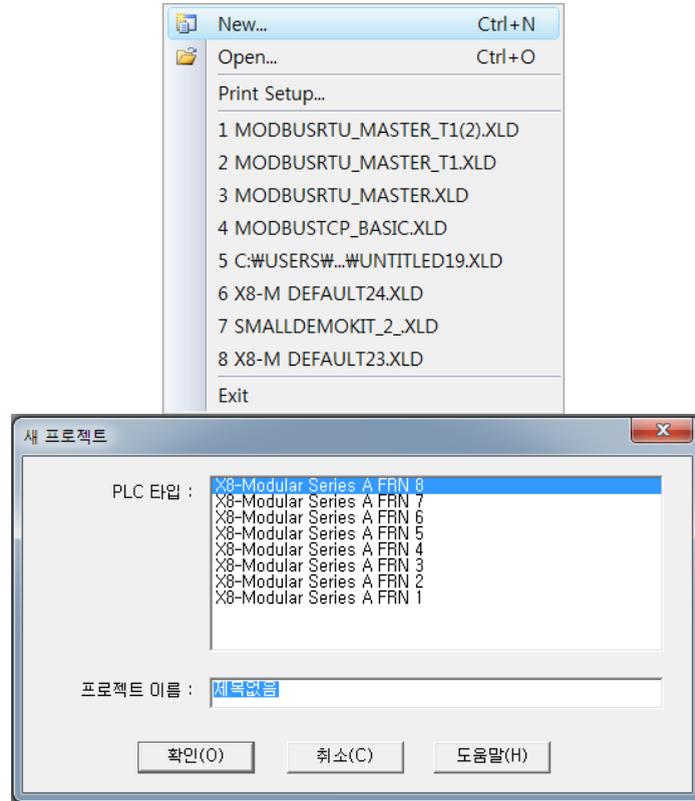
## X8 Setting

Step 1. X8 전용 래더 소프트웨어인 XGPC를 설치하고 실행.



**Step 2. 새 프로젝트 생성**

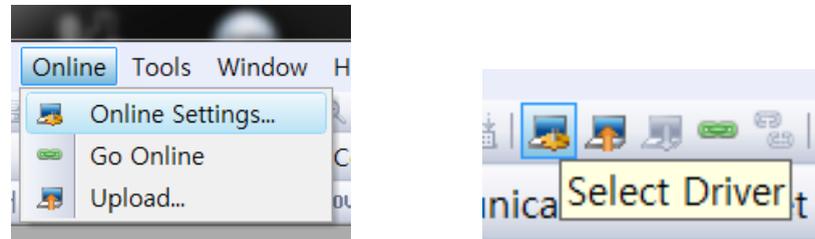
파일 메뉴에서 New 버튼을 누릅니다. PLC type 창에서 X8 하드웨어 모듈을 선택하고, 프로젝트 명을 입력합니다. NX-Alpha Master 설정은 X8 PLC의 FRN(Firmware Revision Number) 8 또는 그 이상부터 지원됩니다.



**Step 3. Online Settings**

Online Settings에서는 연결하고자 하는 PLC의 설정을 입력하여, 해당하는 PLC를 찾아 프로젝트를 Upload 하거나 XGPC와 PLC를 Online 하는 기능입니다.

Online 메뉴에서 Online Settings, 또는 아이콘에서 Online Settings(Select Driver)를 클릭합니다.

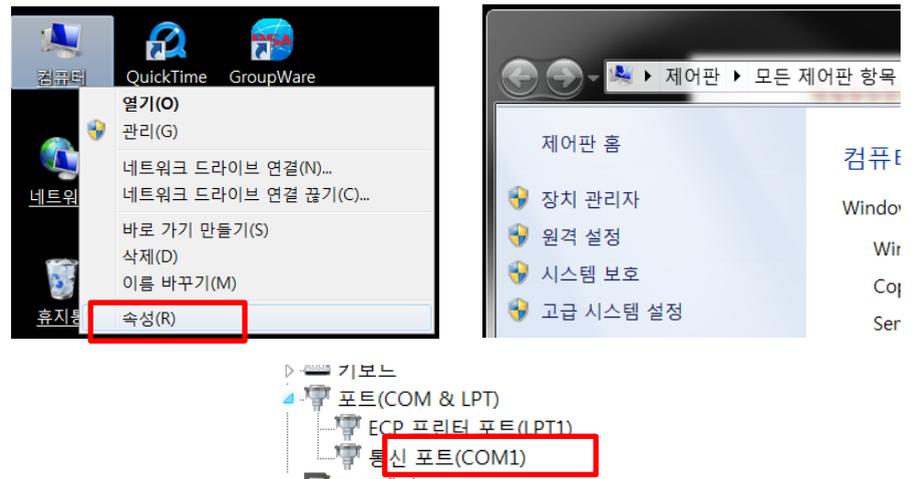


**Case 1. Serial 또는 USB로 연결**

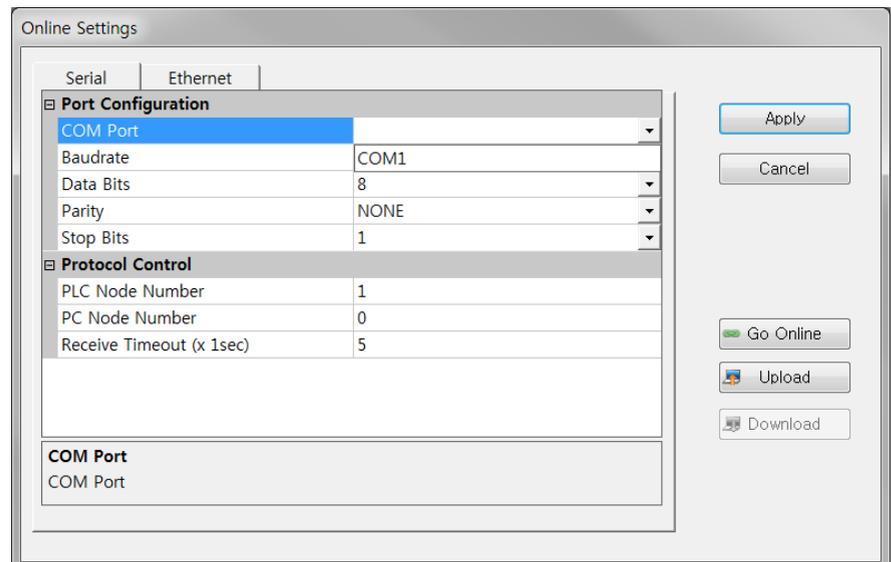
Serial 또는 USB로 연결할 경우, Online Settings 창의 Serial 탭으로 들어갑니다. USB로 연결하였을 때 Serial 탭으로 들어가는 이유는, USB Driver가 통신 포트를 Serial Port로 매핑(mapping)하여 연결하기 때문입니다.

Online Settings 창에 들어가면, 자동으로 COM port 번호가 인식되어 나타납니다. PLC에 적혀 있는 COM0, COM1이 아니라, 컴퓨터에서 인식한 COM port번호를 선택합니다.

COM Port 번호는 내 컴퓨터 > 속성 > 장치관리자로 들어가면 확인할 수 있습니다.



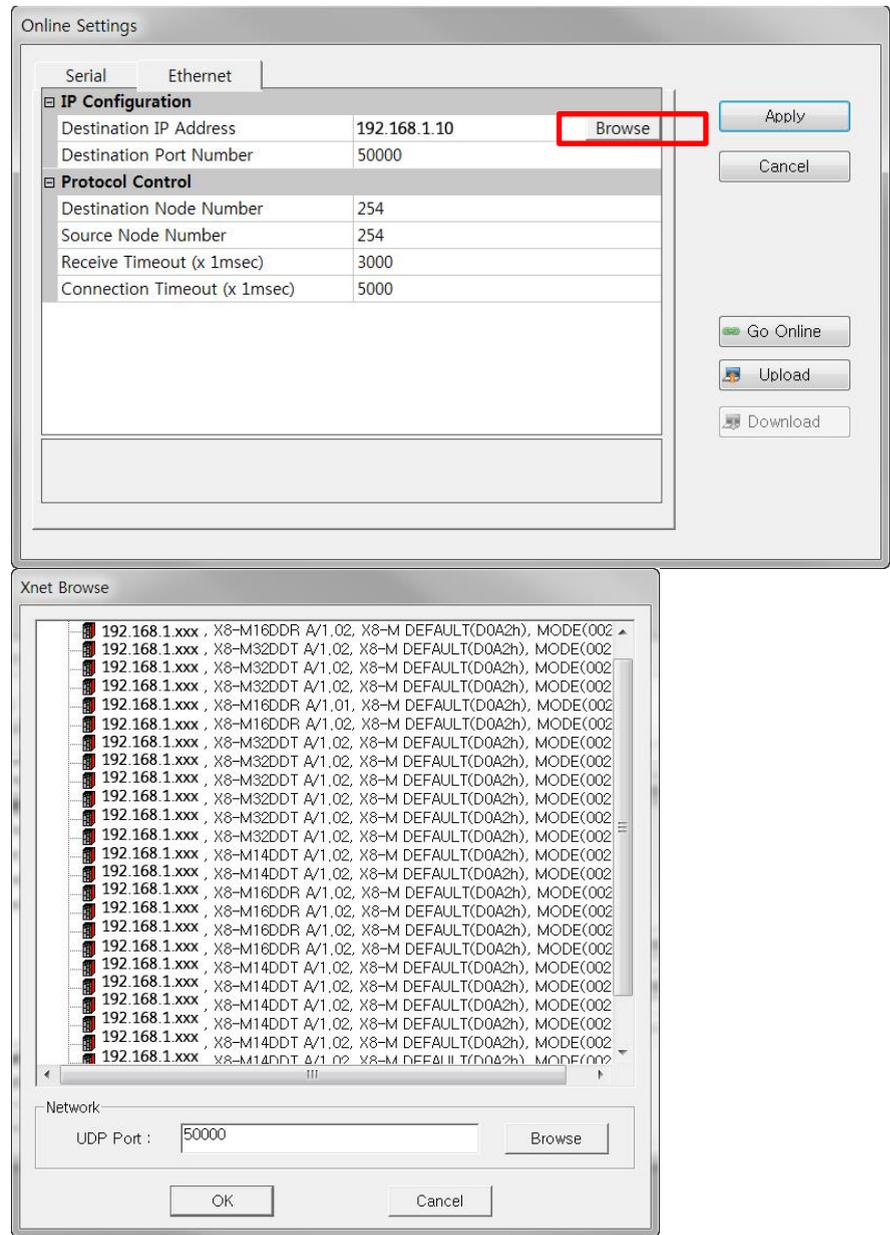
PLC와 연결된 Computer의 COM port번호를 선택 한 후, Apply를 누릅니다. 이 창에서 Go Online 또는 Upload를 누르면, 바로 PLC와 연결하여 데이터를 가지고 오거나 Online을 할 수 있습니다.



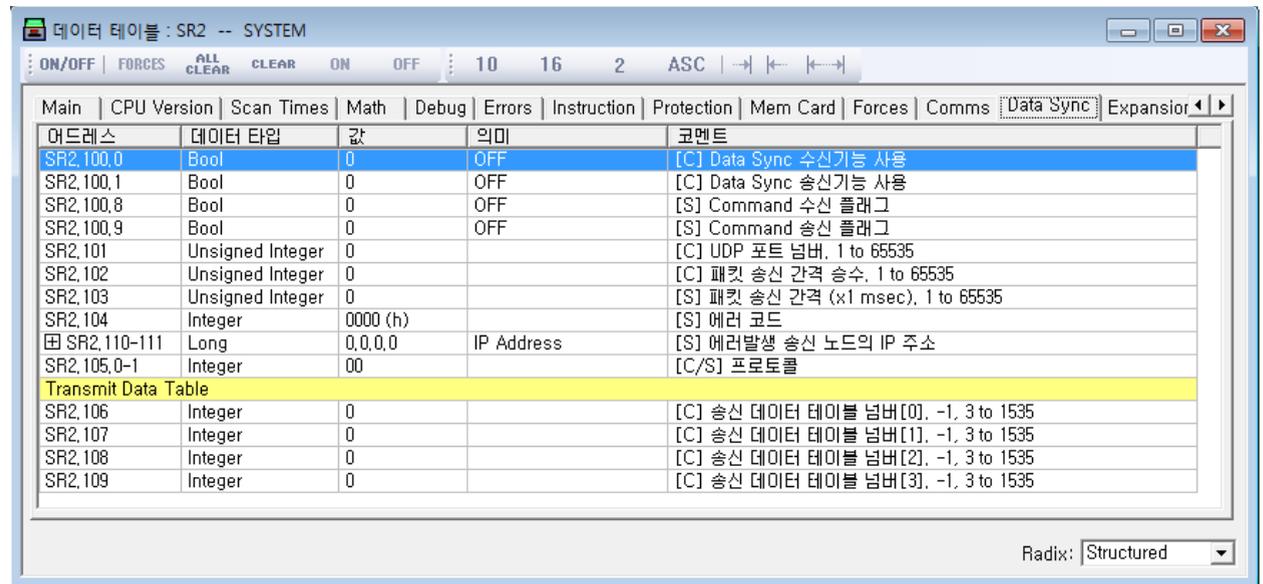
**Case 2. Ethernet으로 연결**

Ethernet으로 PLC와 연결할 경우, Ethernet 탭으로 들어갑니다. Destination IP Address에 연결하고자 하는 PLC의 IP주소를 입력합니다.

컴퓨터와 같은 Network 상으로 연결되어 있는지 확인하고 싶은 경우, 오른쪽의 'Browse'버튼을 클릭하여 연결하고자 하는 PLC의 IP를 선택할 수도 있습니다.



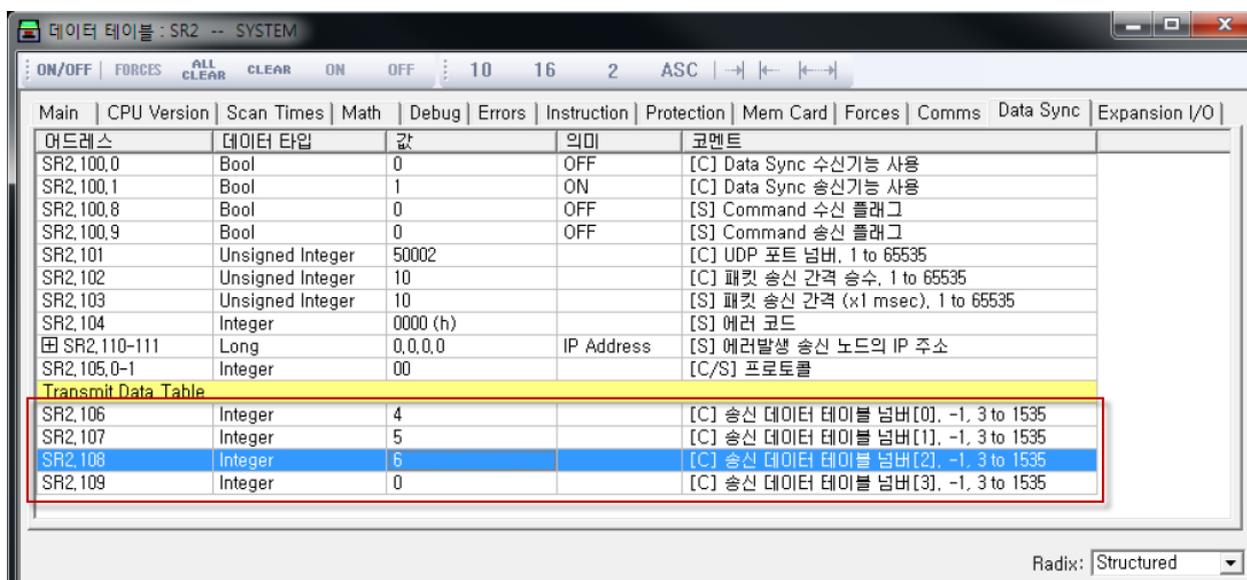
Computer와 연결할 PLC의 IP를 선택 또는 입력한 후, Apply를 누릅니다. 이 창에서 Go Online 또는 Upload를 누르면, 바로 PLC와 연결하여 데이터를 가지고 오거나 Online을 할 수 있습니다.



### 데이터싱크 셋팅 전 상태

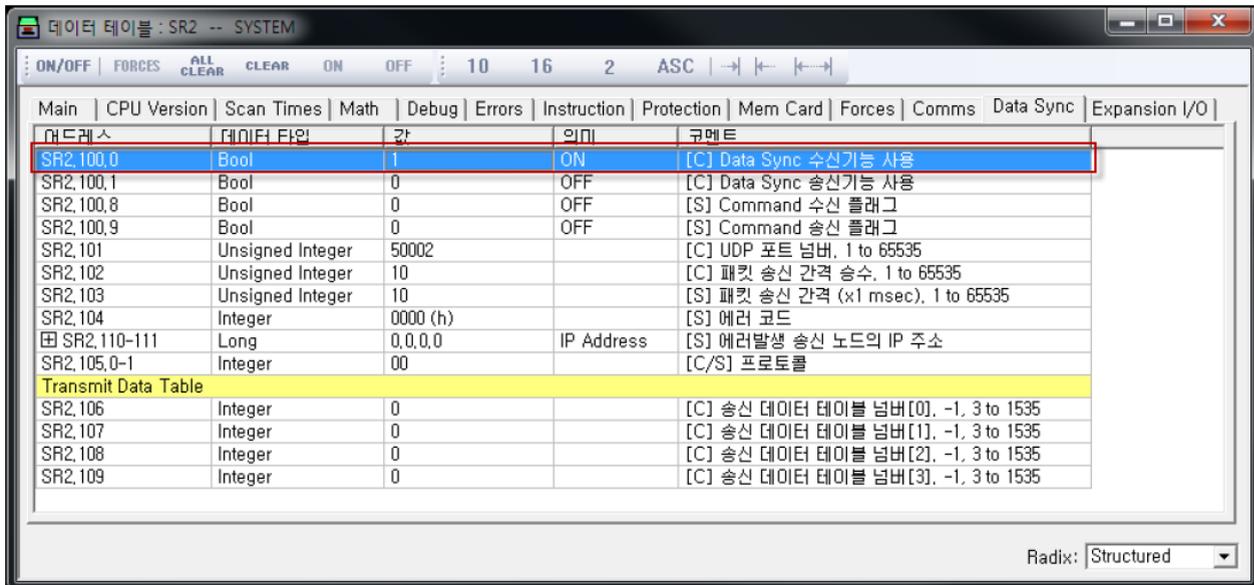
#### 셋팅 순서(송신기능)

1. SR2 데이터 테이블의 Data Sync 탭을 선택합니다.
2. SR2.100.1(Data Sync 송신기능 사용)의 비트를 ON 합니다.
3. SR2.101의 UDP 포트번호를 입력합니다.(예 50002)
4. SR2.102에 패킷 송신 간격 승수에 값을 입력합니다.(예 10)
5. SR2.106 ~ 109에 데이터 동기화할 데이터 테이블 번호를 입력합니다.(예 N4, F5, L6)



셋팅(수신기능)

1. SR2 데이터 테이블의 Data Sync 탭을 선택합니다.
2. SR2.100.0(Data Sync 수신기능 사용)의 비트를 ON 합니다.
3. SR2.101의 UDP 포트번호를 입력합니다.(예 50002)
4. SR2.102에 패킷 송신 간격 승수에 값을 입력합니다.(예 10)
5. 5. Data Sync를 송신하는 데이터 테이블 번호와 타입이 맞는지 확인합니다.



Step 4. PLC Online 및 Download

XGPC에서 셋팅을 끝나치고 난 후, Compile 버튼을 실행하여 Programming상 오류가 있는지 확인합니다. Compile 버튼은 Compile Program과 Compile Project가 있습니다.

## 참조

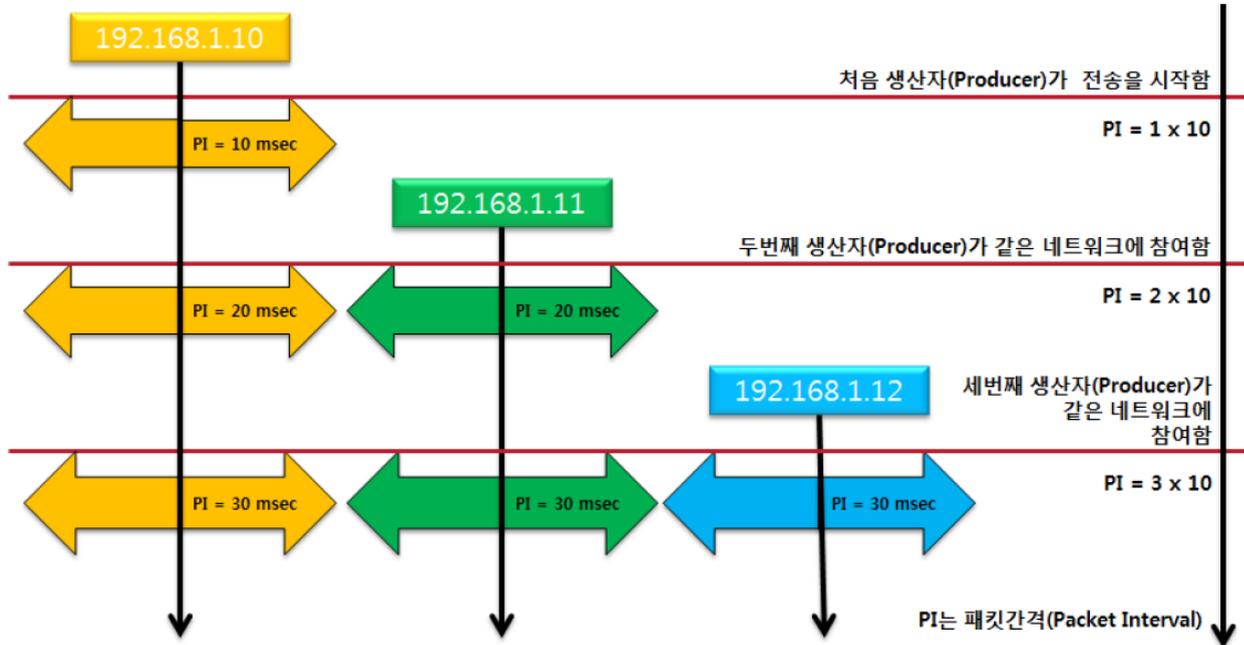
사용가능한 데이터 테이블과 최대 엘리먼트 개수는 다음과 같습니다.

데이터 타입	최대 엘리먼트 개수	최대 바이트
B	512	1024 byte(=512x2)
N	512	1024 byte(=512x2)
F	256	1024 byte(=256x4)
L	256	1024 byte(=256x4)
A	512	1024 byte(=512x2)
ST	12	1024 byte(=12x84)

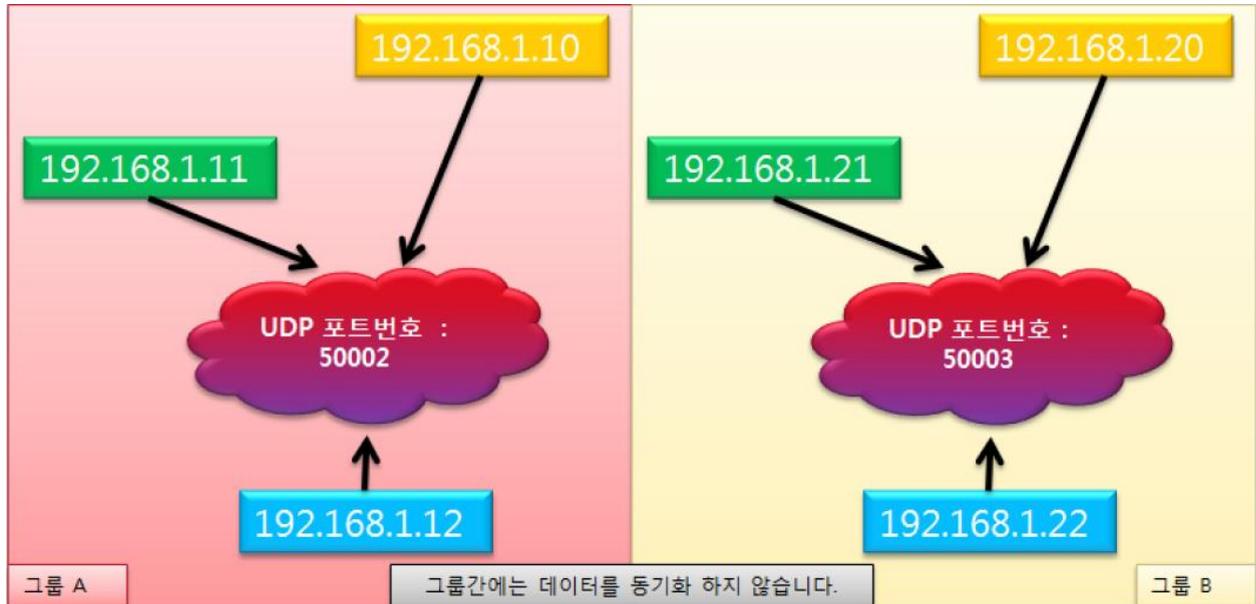
최대 엘리먼트를 초과하는 엘리먼트는 데이터를 동기화 하지 않습니다.

패킷간격(Packet Interval)은 다음 공식에 의해서 자동으로 계산되어 집니다.

패킷 간격(Packet Interval) = 생산자 수(Number of Producers) \* 패킷간격 승수(Packet Interval Multiplier) 예를 들어 5개의 생산자(Producer)가 네트워크에 있고 패킷간격 승수(Packet Interval Multiplier)가 10이면, 생산자(Producer)는 50msec마다 패킷을 생산해서 보냅니다. 5개 생산자(Producer)가 100msec 간격으로 패킷을 보내려고 하려면, 패킷간격 승수(Packet Interval Multiplier)를 20으로 설정하면 됩니다. 네트워크 트래픽이 다른 패킷에 의해서 높아지는 경우, 패킷간격 승수(Packet Interval Multiplier)는 자동적 또는 임의적으로 10 또는 10보다 큰수로 조정됩니다. 다음은 패킷간격 승수(Packet Interval Multiplier)이 10일때 생산자 수(Number of Producers)에 따른 패킷 간격(Packet Interval)의 변화 그림입니다.



- 최대 만들 수 있는 생산자(Producer)는 256개의 PLC입니다.
- 최대 보낼 수 있는 데이터케이블 개수는 4개 입니다.
- X8 CPU는 같은 네트워크에 연결되어 다른 UDP포트번호를 설정해서 그룹화 시킬 수 있습니다.



- 데이터 싱크 패킷을 받고 데이터테이블에 업데이트하게 되면 SR2.100.8(Command Received Flag)가 ON 됩니다.
- 데이터 싱크 패킷을 보내게 되면 SR2.100.9(Command Sent Flag)가 ON 됩니다.
- 데이터 싱크 에러코드는 Low byte는 에러코드를 나타내고, High byte는 데이터 테이블의 번호를 나타냅니다.

에러코드	설명
0x0080	Invalid Tx Data Table
0x0081	Invalid Tx Data Type, high byte is data table offset
0x0082	Invalid Tx Data Table Number, high byte is data table offset
0x0083	Duplicate Tx Data Table Number detected on the network, high byte is data table offset
0x0084	Invalid Packet Received
0x0085	Mismatched Data Table Type Received
0x0086	Reached Maximum Producers on the network

다음과 같은 순서로 에러 코드를 찾습니다.

1. SR2 데이터 테이블의 Data Sync탭을 선택합니다.
2. SR2.104(에러코드)의 값을 확인합니다.



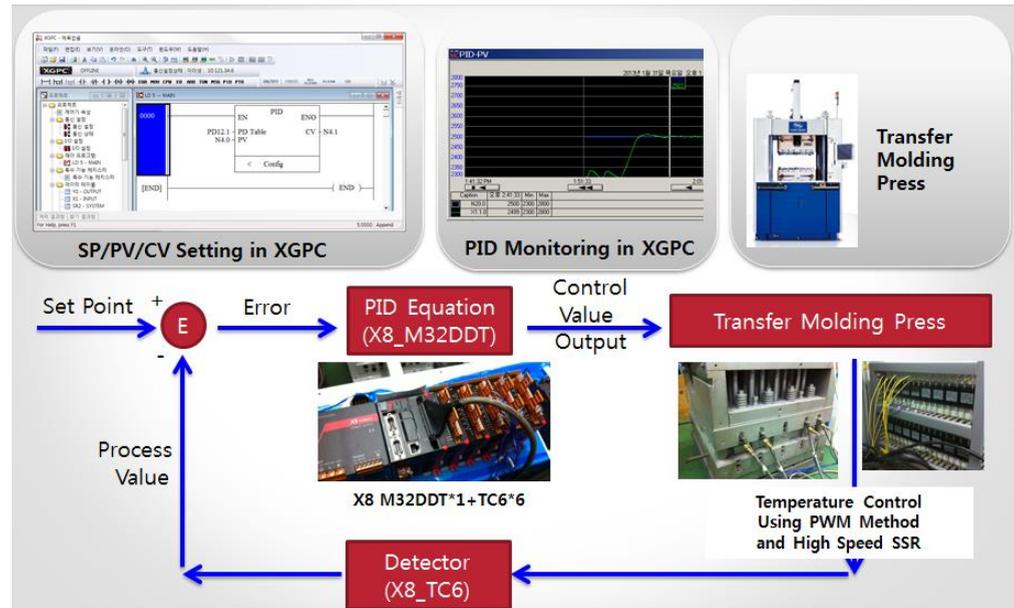
## 13장. X8 PID AUTO TUNING

본 장에서는 X8 PID AUTOTUNING기능을 설명합니다.

---

개요.....	13-2
프로세스 제어 .....	13-2
PID 제어란 .....	13-6
PID 예제 .....	13-11

## 개요



PID 제어에 앞서...

## 프로세스 제어

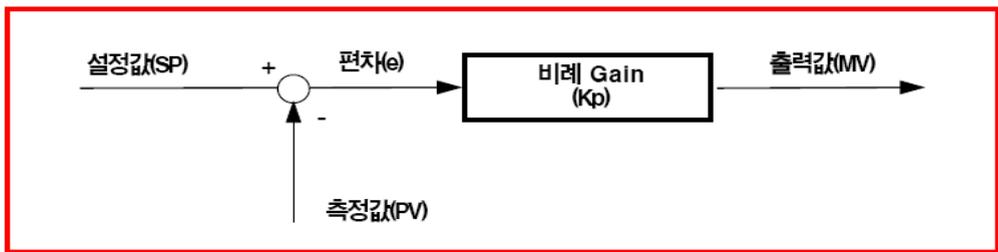
프로세스 제어에서는 설정한 목표값(SP)과 실제로 제어대상으로부터의 측정값(PV)을 비교하여, 편차(e)를 구하고, 그것들을 일치시키도록 출력값(CV)을 출력합니다. 또 프로세스 제어에는 구한 편차(e)에서 출력값(CV)을 구하는 방식으로 P동작, I동작, D동작 등이 있습니다. 여기에서는 그러한 각 제어방식의 동작에 대해 설명하겠습니다

### 비례제어 (P 동작)

비례제어는 제어목표값과 측정값의 편차(e)에서 출력값(MV)을 얻는 제어방식입니다. 편차신호를 e, 출력값을 MV라 하면 양자의 관계식은 다음과 같이 표시됩니다.

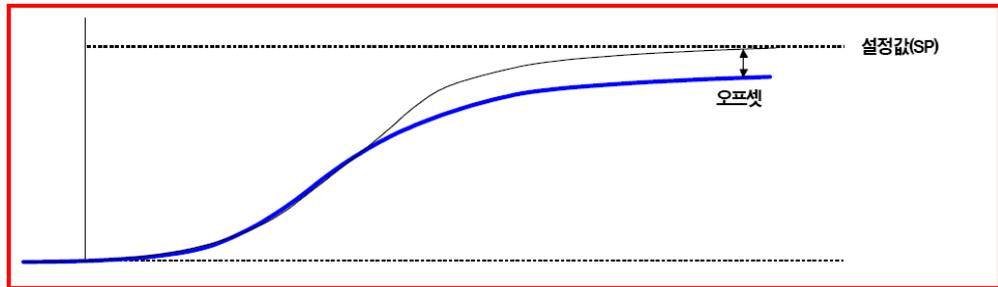
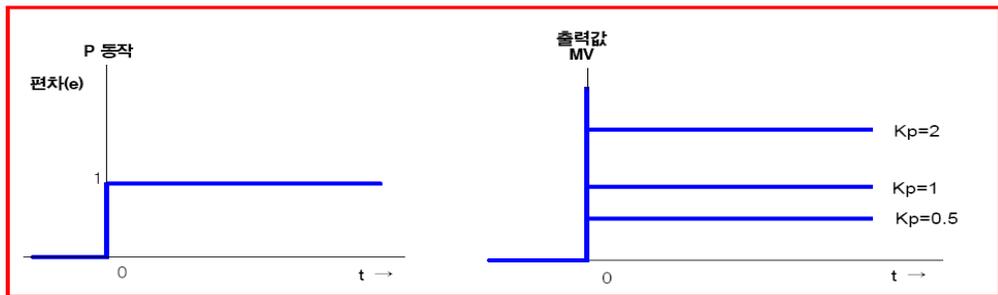
$$MV = K_p \times e$$

상기의  $K_p$ 는 비례 Gain 이라합니다. 상기와 같이 비례제어에서는 편차 e(입력)와 출력치(CV)는 같은 형태가 됩니다. 이 비례 Gain 을 크게하면 커다란 출력값이 출력되고, 비례 Gain 이 작으면 출력 값도 작아집니다



**비례제어의 오프셋(편차,e)**

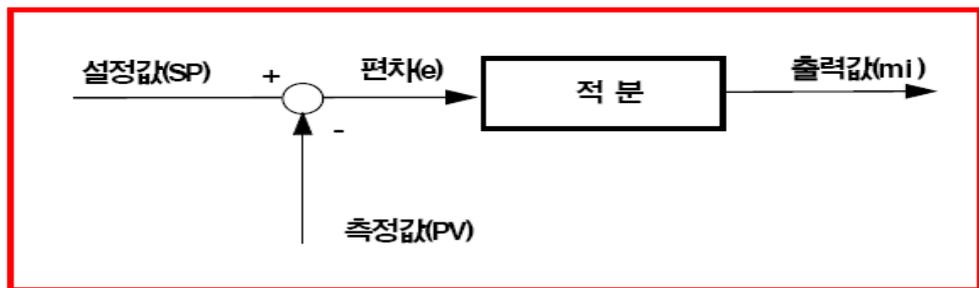
비례동작에서는 편차를 "0"으로 하는 방향으로 출력값을 조정하는데, 편차가 "0"이 되면 출력값도 "0"이 되기 때문에 "0"이 될 수가 없습니다. 그 때문에 제어대상의 전달함수등에 의해 오프셋(정상편차)이 남습니다.



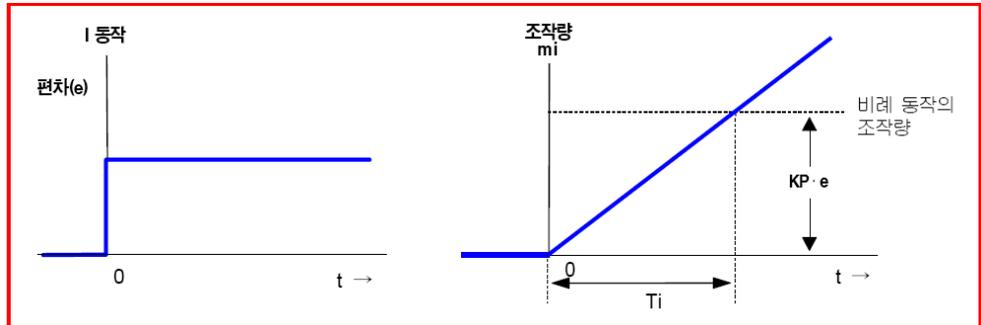
**적분제어 (I 동작)**

적분제어는, 편차를 없애도록 연속적으로 출력량을 변화시키는 제어동작입니다. 적분동작은 비례동작을 기본으로하여 정의되며, 입출력의 관계는 다음식으로 표시됩니다.

$$m_i = \frac{1}{T_i} \int e dt$$



상기의  $T_i$ 는 적분시간이라하며 스텝상의 편차가 생기고부터 적분 동작의 출력량이 비례동작과 같아 질때까지의 시간을 나타냅니다

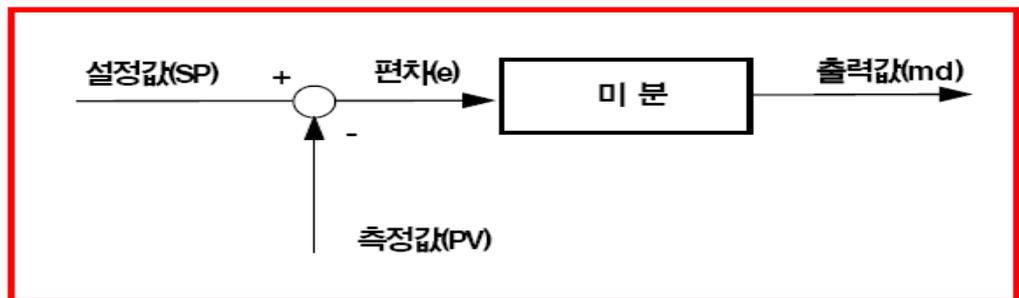


적분시간( $T_i$ )을 작게하면 조작량의 각도가 커져 적분동작이 강해 집니다. 반대로 적분시간( $T_i$ )을 크게하면 조작량의 각도가 완만해져 적분동작이 약해 집니다. 적분동작은 단독으로는 사용할 수 없습니다. 비례동작 또는 비례동작과 조합하여 사용해주십시오.

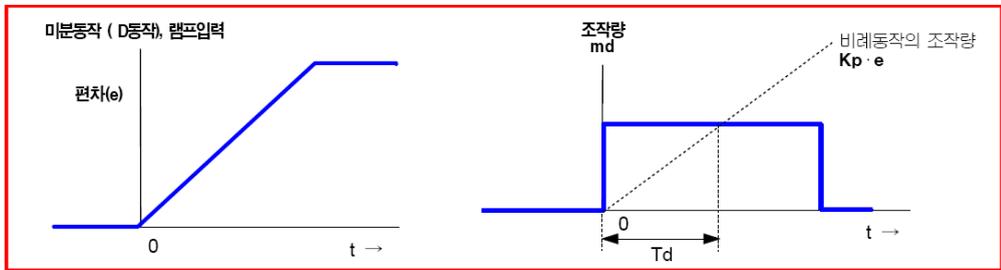
### 미분제어 (D 동작)

미분제어는 편차의 변화속도(각도)에 따라 조작량을 출력합니다. 이 때문에 편차가 발생한 시점에서 크게 정정동작을 실행합니다. 따라서 편차에 대하여 앞을 내다본듯 조작량을 수정하므로 외란에 의한 제어대상의 커다란 변동을 방지합니다. 미분동작의 관계는 다음식으로 표시됩니다.

$$m_d = T_d \cdot \frac{de}{dt}$$

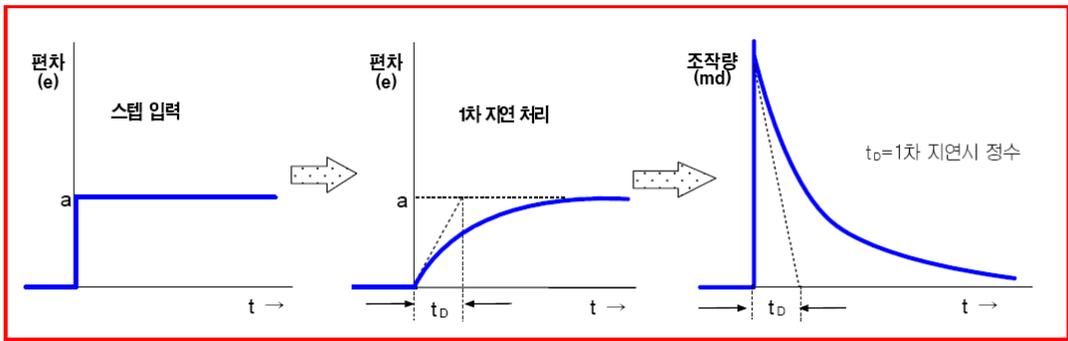


상기Td는 미분시간이라 하며, 램프상의 편차가 발생했을 때 미분 동작만의 조작량과 비례동작만의 조작량이 같아지기까지의 시간을 나타냅니다.



상기그림과 같은 편차입력의 경우, 편차의 약간의 시작으로 조작량을 출력합니다. 그후편차의 변화속도(각도)가 같으면 조작량도 일정합니다.

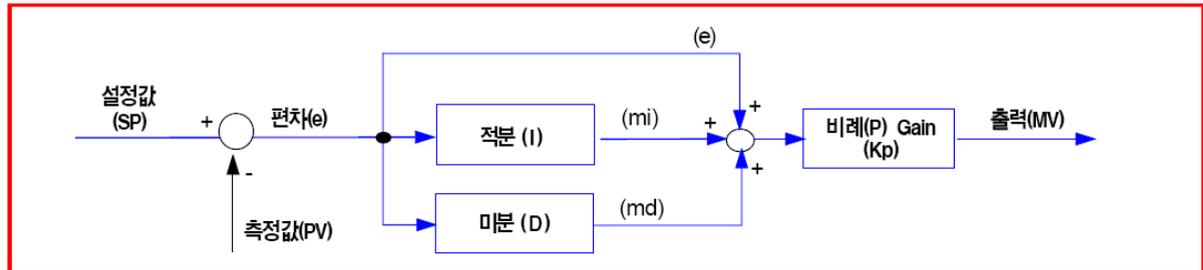
불완전미분에 대하여



순미분동작의경우, 스텝상의 입력이 있으면 그순간 커다란 조작량을 출력하여 제어대상에 악영향을 미칩니다. 그 때문에 PID제어 시스템에서는 상기 그림과 같이 1차지연의 필터를 설정하여 불완전 미분을 실현하고, 스텝입력에 대한 급격한 출력변화를 개선 시킵니다. 미분동작도 단독으로는 사용할 수 없습니다. 비교 동작 또는 비례동작과 조합하여 사용해주시오.

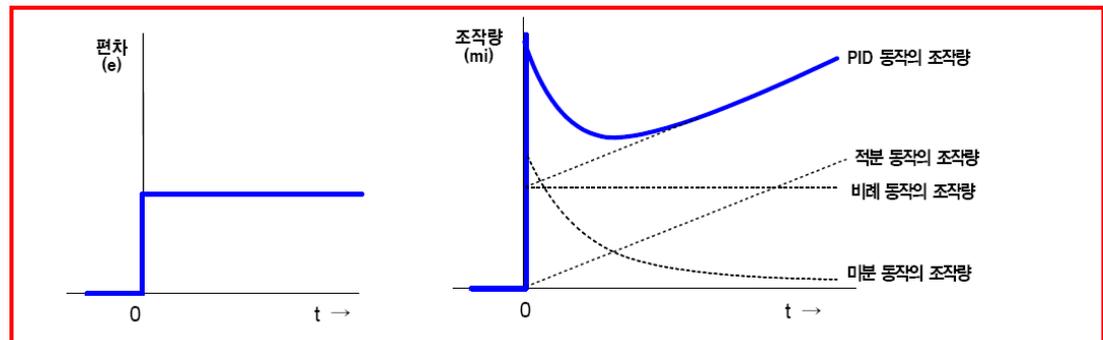
## PID 제어란

PID제어는 비례, 적분, 미분의 3가지를 조합한 제어방식입니다. 비례동작(P)으로 발생하는 오프셋량도 적분동작(I)을 가함으로써 해소되며, 다시 미분동작(D)을 가하면 이상적인 제어동작이 실현됩니다.



### 조작출력

PID제어에서는 비례(P), 적분(I), 미분(D) 각각의 동작을 합한 특성의 조작량 출력을 수행합니다. PID제어에서 각각의 동작비율(강도)은 파라미터 또는 선택하는 변환 알고리즘으로 조정 가능합니다.

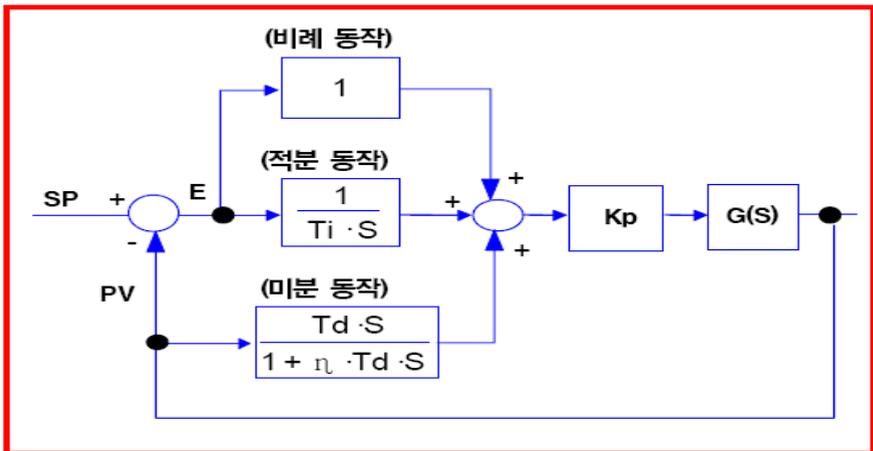


### PID제어 알고리즘

비례,적분,미분제어를 조합한 제어방식이 PID제어이며, 또한, PID의 연산방식에는 측정값 미분 선행형과 측정값 비례 미분 선행형의 2가지 방식이 있습니다.

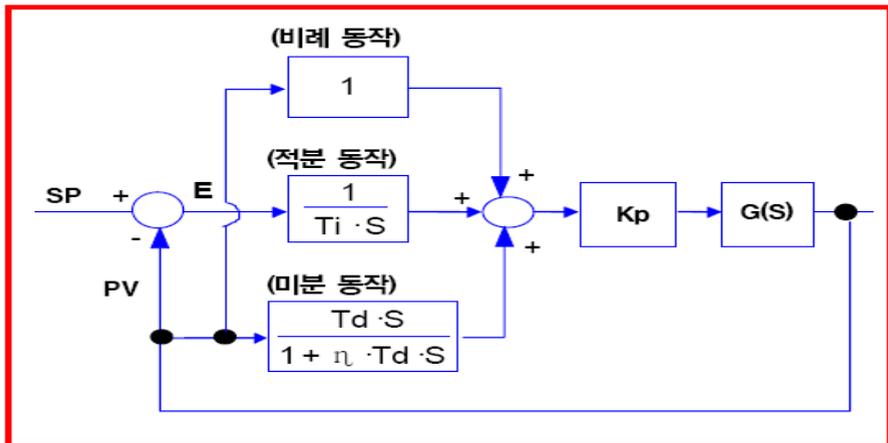
#### PI-D 제어(미분선행형)

제어 목표값이 스텝함수상으로 변화했을 경우, 미분동작의 원리적으로 조작량이 예리한 펄스상의 응답을 나타냅니다. 그러나, 실제 제어를 해보면 이와 같은 제어는 불가능하며, 설비에 미치는 영향도 바람직하지 않습니다. 그래서 목표값에 대한 미분처리를 하지 않고 측정값 만에 미분처리가 작용 하도록 한 연산이 PI-D(미분선행형PID제어)입니다.

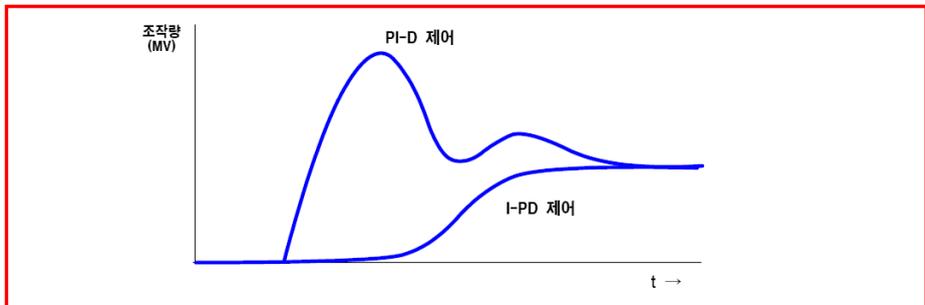


**I-PD제어(비례미분선행형)**

상기 PI-D(미분 선행형 PID제어)에서도 제어 목표값이 스텝상으로 변화했을 경우, 비례동작에는 스텝함수가 포함되어 있습니다. 따라서, 미분동작에 더하여 다시 비례동작도 측정값에만 작용 하도록 한 것이 I-PD(비례 선행형PID제어)입니다.



목표값을 변경 했을때의 제어의 응답



일반적으로는 각각의 제어특성을 활용하여 I-PD는 상수(constant) 값 제어(프로세스제어)에 가장효과적이며 ,PI-D는 follow-up 제어에 가장 효과적으로 사용됩니다.

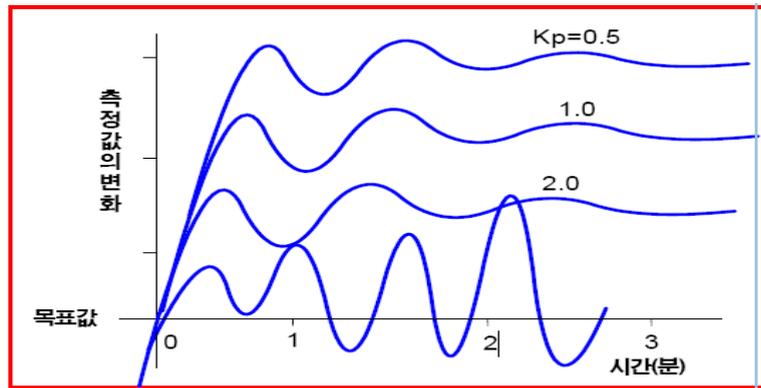
### 각동작에서파라미터의효과

P동작,I동작,D동작 각각의 제어의 포인트가 되는 파라미터를 변화시켰을 경우의 효과와 영향을 간단하게 설명하겠습니다. PID제어에서는 이러한 파라미터의 설정에 의해 외란에 대한 제어의 응답이 변화하고, 설정이 최적이면 제어량을 목표값에 의해 빨리 일치시켜 일정하게 유지할 수가 있습니다.

#### 비례제어(P동작)

P동작에서는  $K_p$ (비례Gain)를 크게하면 오프셋량이 감소하여 첫 번째 산도 작아지는데, 너무크면 진동적이 되고, 진동의 주기가 짧아져 결국 발산합니다.

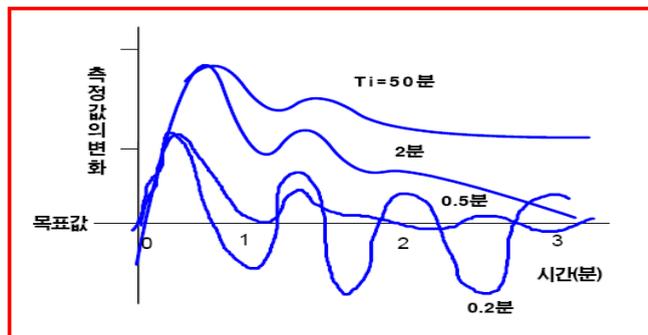
비례Gain의 변경 예)



#### 비례적분제어(PI동작)

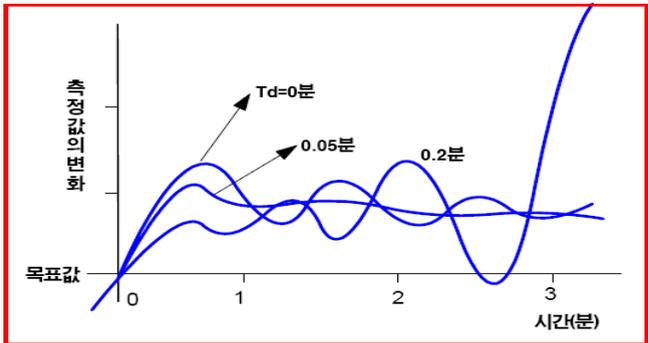
PI동작에서는  $T_i$ (적분시간)를 짧게하면 제어목표값으로 처음으로 돌아가는 시간이 짧아져 첫 번째 산이 작아지는데, 너무 짧아지면 진동적이 되어 결국은 발산합니다.

적분시간의 변경 예)



**비례미분제어(PD동작)**

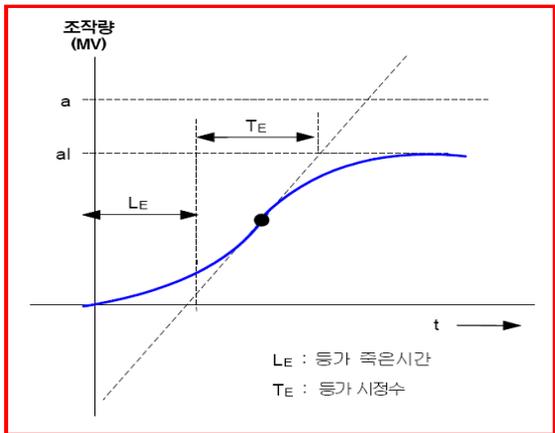
PD동작에서는 Td(미분시간)를 크게하면 진동이 억압되어 첫번째 산이 작아지는데, 너무크면 다시 진동적이되고, 진동의 주기가 짧아져 결국 발산합니다.



**PID제어의 파라미터 설정방법**

PID제어에서 프로세스제어의 각 파라미터를 구하는 경우 복잡한 계산에 의해 구하는 이외에 실제의 제어계에 부가하여 직접 실행한값을 구하는 실험적인 방법이 있습니다. 이것을 Ziegler-Nichols의 스텝응답법이라고 부릅니다. 25% 댄핑의 제어특성을 실험과 간단한 계산에서 얻는 것이 가능합니다. 아래에 이방법에 대해 설명합니다.

아래의 그림은 제어대상에 스텝입력a를 인가한 제어응답의 결과입니다. 이그림의 응답곡선의 변곡점에 접선을 연장하여 끝난시간(LE), 시정수(TE)을 구하고 파라메타를 얻을 수 있습니다.



<p><b>P 동작</b></p> $K_p = \frac{T_E}{K \cdot L_E}$ $T_i = \infty$ $T_d = 0$	<p><b>PI 동작</b></p> $K_p = 1.2 \times \frac{T_E}{K \cdot L_E}$ $T_i = 3.3 \times L_E$ $T_d = 0$
<p><b>PID 동작</b></p> $K_p = 1.2 \times \frac{T_E}{K \cdot L_E}$ $T_i = 2.0 \times L_E$ $T_d = 0.5 \times L_E$	

㉞ 각 계산식에서 사용하는 파라메타 "K"는 위그림에서 a,a1 이 얻어지며, 다음 방정식으로 구합니다.  $K = a / a1$

참조사항

25% 덤핑(Dumping)에 대해서?

실제로 제어할 때 ,측정치의 지속이 25%(곡선크기가1/4)에 되는 지점을 표시합니다.

주의사항

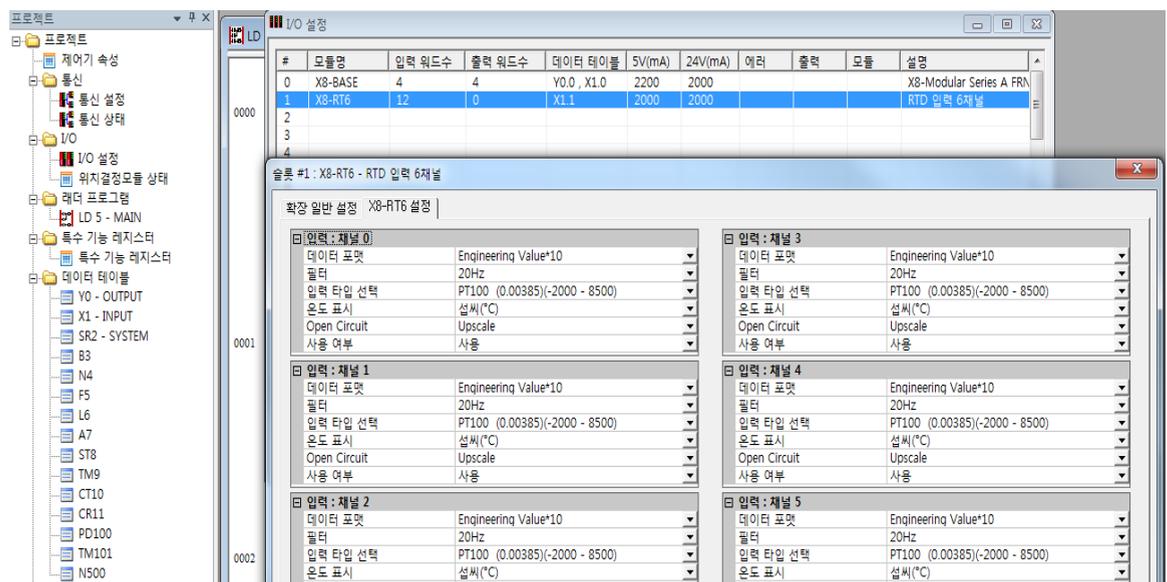
상기에서 구한값은 이론적인 값이며, 실제는 이값을 직접보면서 제어대상을 조정해 주십시오.

# PID 예제

※다음은 X8 PID AUTO 튜닝 기능을 통해 제어를 하기 위한 절차입니다.

- 1) 우선적으로 온도모니터링을 하기 위한 X8 온도입력 모듈을 설정시킵니다.

위 그림은 RT6 모듈에 PT100을 사용하였을 경우에 예시이고, 제어를 위한 센서가 다르다면 입력 타입센서를 맞추어 설정합니다.



아래는 PID 의 연산을 수행하는 명령어 및 데이터 처리에 대하여 설명합니다.

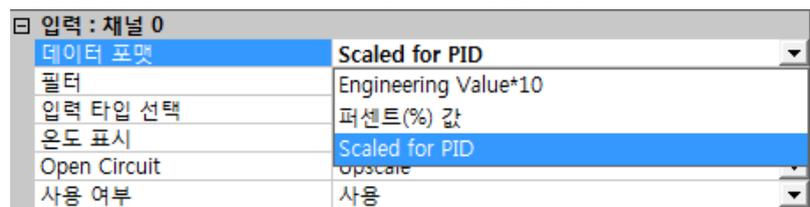
### ※PV & SV DATA 생성

(SCAP명령어는 Input Min과 Input Max의 범위에 대해 Saled Min과 Saced Max 범위로 배율한 Data를 변화시킵니다.)

- 2) PID 제어를 사용하기 위해서는 Saled Min과 Saced Max를 0~16383으로 설정하기 바라며, INPUT DATA의 기준에 대하여 정의를 하였으면 PV,SV 모두 동일한 범위로 설정해주시기 바랍니다.

이 명령어 사용을 원치 않으시면 데이터 포맷을 Scaled for PID 로 선택하시면 됩니다.

사용범위가 0~16383으로 바로 표기됩니다.



====PV 값 계산====		PV입력계산 DATA	
	EN	SCAP	ENO
X1.1.0	IN		OUT -N4.3
602	<		>4060
-2000	Input Min.		
-2000	<		
8500	Input Max.		
8500	<		
0	Scaled Min.		
0	<		
16383	Scaled Max.		
16383	<		
====SV 값 계산====			
	EN	SCAP	ENO
N4.0	IN		OUT -PD100.0.SPcaled
600	<		>4057
-2000	Input Min.		
-2000	<		
8500	Input Max.		
8500	<		
0	Scaled Min.		
0	<		
16383	Scaled Max.		
16383	<		

**PV값 계산**

X1.1.0은 앞장에서 설정한 RTD를 통해 들어오는 입력 DATA 입니다

이 DATA를 제어를 하고 싶은 범위를 Input 범위에 입력합니다.

Scaled Data는 변환할 비율을 입력하는데, 본 PID 제어를 하기 위해선 반드시 0~16383의 범위를 입력합니다.

N4.3은 X1.1.0의 배율을 적용한 DATA가 출력으로 나오게 됩니다.

**SV값 계산**

N4.0은 목표온도를 설정하기 위한 DATA를 넣게 됩니다.

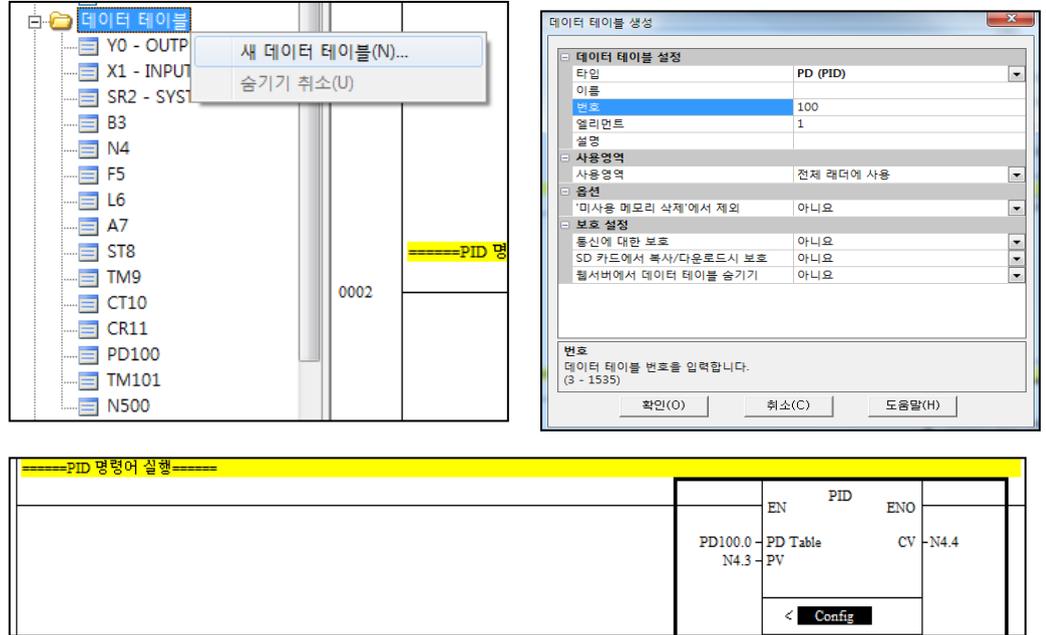
이 DATA를 제어를 하고 싶은 범위를 Input 범위에 입력합니다.

Scaled Data는 변환할 비율을 입력하는데, 본 PID 제어를 하기 위해선 반드시 0~16383의 범위를 입력합니다.

PD100.0 SPcaled는 N4.0의 배율을 적용한 DATA가 출력으로 나오게 됩니다.

3) DATA 테이블 생성 및 셋팅

※PID 명령어 생성합니다. 이를 위해선 PD DATA를 생성해주시기 바랍니다



PD Table은 생성한 PD Data를 입력하고 앞장에서 설정한 PV 출력 메모리를 입력합니다.

Config를 클릭하여 세부설정을 합니다.

Scan에 따라 PID제어를 하기 위해서 Timed Or PIT를 "1"로 설정합니다.

Forward Acting은 Cooling시스템에서,Reverse Acting은 Heating 시스템에서 사용합니다.

Reduce Overshoot는 튜닝data 찾는 과정에서 목표온도 이상 사용여부에 대해 선택합니다.

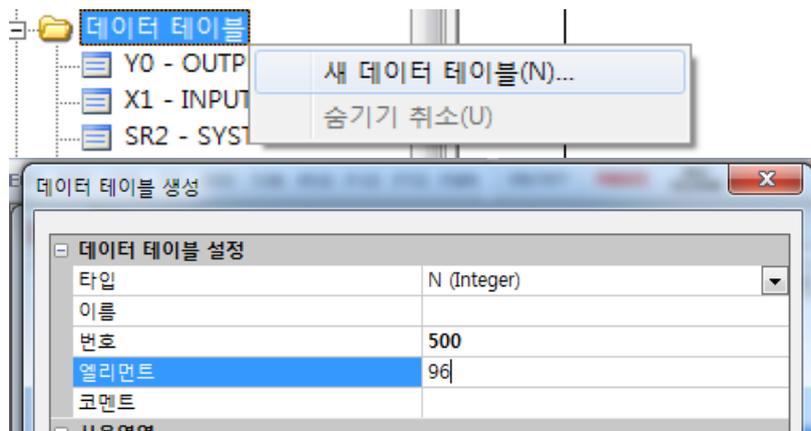
0:SV온도이상사용 1:SV온도이상미사용

어드레스	데이터 타입	값	의미	설명
PD100,0	PID	...		
->> Tuning Parameters				
- TimedOrPIT	Bool	1	Timed	
- AutoOrManual	Bool	0	Auto Mode	
- ForwardReverseActing	Bool	0	Reverse Acting	
- OutputLimitingEnable	Bool	0	Output Limit Disable	
- GainRangeSelection	Bool	0	Kc = Value * 0,1 and Ti = Value * 0,1	
- SPScalingEnabled	Bool	0	Scale Disabled	
- LoopUpdatesTooFast	Bool	0	Slow update	
- DerivativeInputOnError	Bool	0	Disabled	
- PVInDeadband	Bool	0	Deadband Disabled	PV In Deadband
- UpperOutputLimit	Bool	0		Output >= Upper Output Limit
- LowerOutputLimit	Bool	0		Output <= Lower Output Limit
- SPOutOfRange	Bool	0		
- PVOutOfRange	Bool	0		
- Done	Bool	0		
- Enabled	Bool	0	Disabled	
- ReduceOvershoot	Bool	0	Disable	
- Kc	Integer	0	0,0	
- Ti	Integer	0	0,0 min/repeat	
- Td	Integer	0	0,00 min	
- FeedForwardBias	Integer	0		
- ZeroCrossingDeadband	Integer	0		
- LoopUpdateTime	Integer	0	0,00 sec	
- IntegralSum	Long	0		
- AlteredDerivativeTerm	Long	0		
- SPScaledOld	Integer	0		
->> Inputs				
- SPScaled	Integer	0		
- SPMax	Integer	0	Max value	
- SPMin	Integer	0	Min value	
- PVScaled	Integer	0		
->> Outputs				
- CVPercent	Integer	0	%	
- CVHighLimit	Integer	0	%	
- CVLowLimit	Integer	0	%	
- ScaledError	Integer	0		
- ErrorCode	Integer	0000 (h)		
->> Extended PID				
- ExtendedPidData TbNumber	Integer	500		

4) AUTO PID TUNING사용

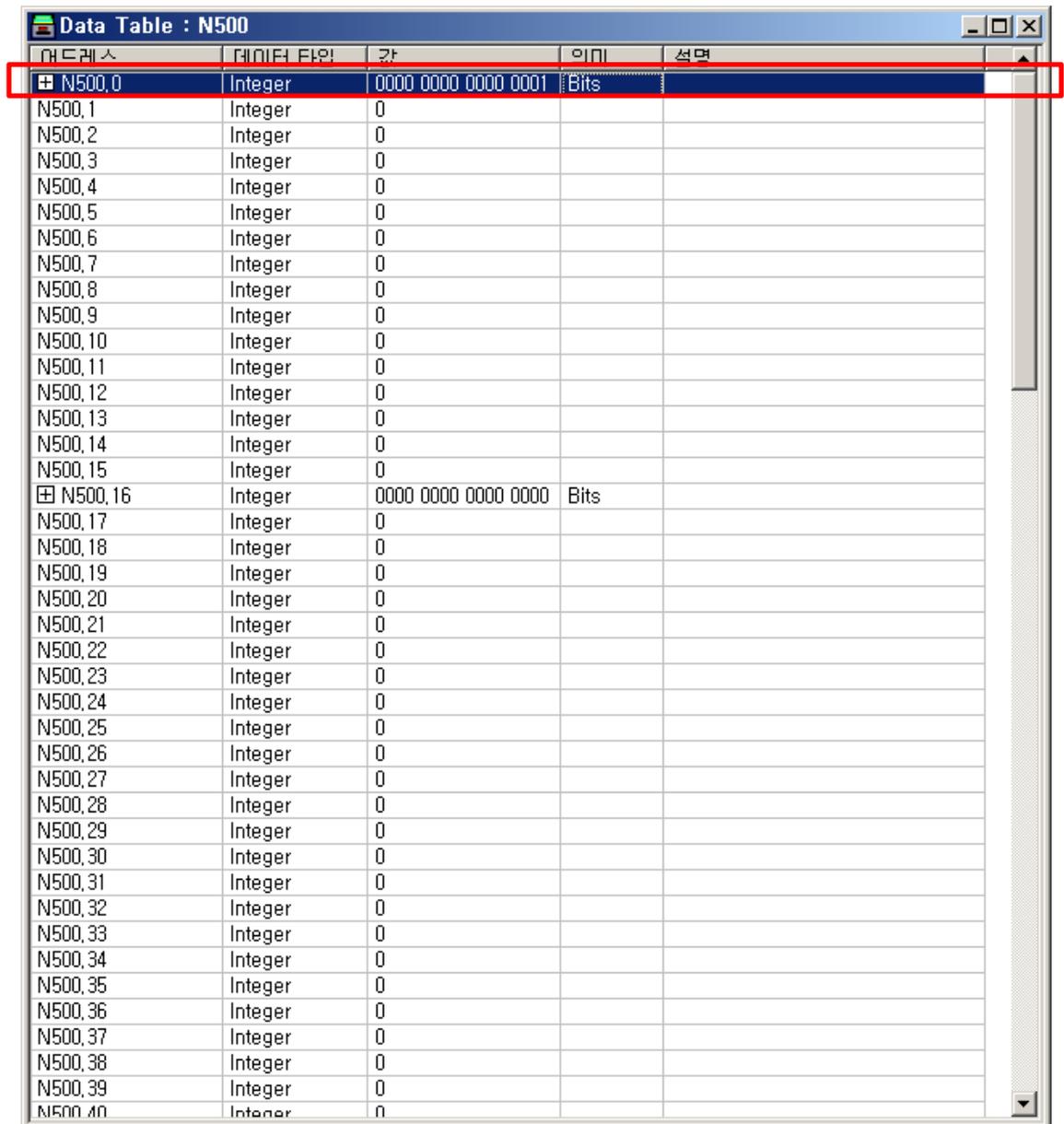
Extend Pid Data Tb Number 사용을 위해 96개의 Word Data가 필요합니다.

때문에 96개의 연속적인 Data를 포함시킬 메모리 주소를 설정합니다. ex)N500



ExtendPidDataTbNumber 에서 선택한 DATA 테이블에 0번 WORD를 "1"설정하게 되면 AUTO 튜닝을 사용하게 됩니다.

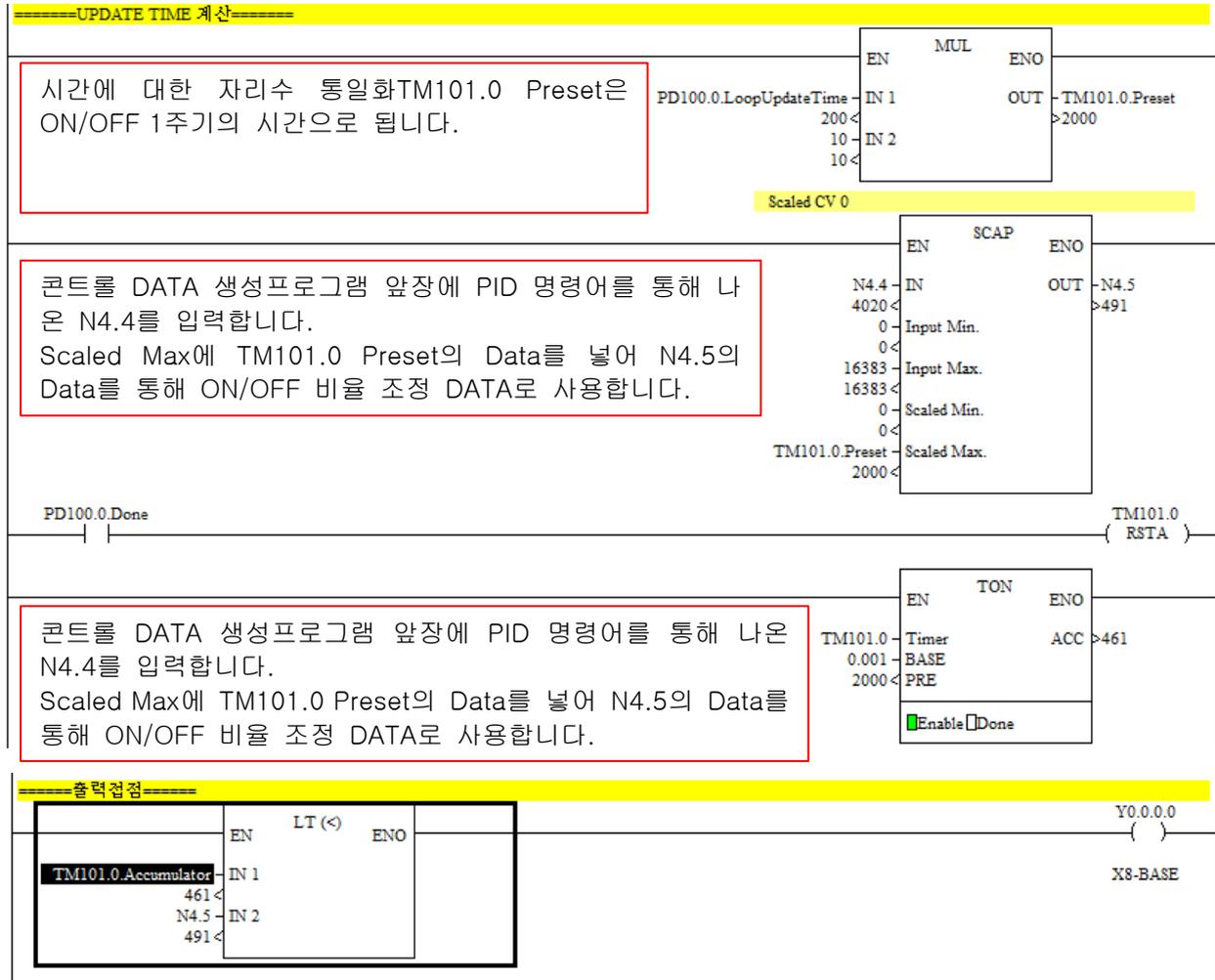
Ex) ExtendPidDataTbNumber 500설정 N500으로 사용하였을 경우



주소	데이터 타입	값	의미	설명
N500,0	Integer	0000 0000 0000 0001	Bits	
N500,1	Integer	0		
N500,2	Integer	0		
N500,3	Integer	0		
N500,4	Integer	0		
N500,5	Integer	0		
N500,6	Integer	0		
N500,7	Integer	0		
N500,8	Integer	0		
N500,9	Integer	0		
N500,10	Integer	0		
N500,11	Integer	0		
N500,12	Integer	0		
N500,13	Integer	0		
N500,14	Integer	0		
N500,15	Integer	0		
N500,16	Integer	0000 0000 0000 0000	Bits	
N500,17	Integer	0		
N500,18	Integer	0		
N500,19	Integer	0		
N500,20	Integer	0		
N500,21	Integer	0		
N500,22	Integer	0		
N500,23	Integer	0		
N500,24	Integer	0		
N500,25	Integer	0		
N500,26	Integer	0		
N500,27	Integer	0		
N500,28	Integer	0		
N500,29	Integer	0		
N500,30	Integer	0		
N500,31	Integer	0		
N500,32	Integer	0		
N500,33	Integer	0		
N500,34	Integer	0		
N500,35	Integer	0		
N500,36	Integer	0		
N500,37	Integer	0		
N500,38	Integer	0		
N500,39	Integer	0		
N500,40	Integer	0		

5) 래더작성

※최종 출력접점을 ON/OFF 하기 위한 주기를 계산하기 위한 프로그램 입니다.



앞장에서 설정한 CV DATA와 타이머에 ACCUMULATOR를 통해 출력접점을 제어하게 됩니다..

---

## 14장. X8 CSD7N OVER ETHERCAT

---

X8 CPU OS FRN 11.10.....	14-2
지원 가능한 EtherCAT Slave.....	14-2
장점.....	14-3
CSD7 서보 드라이브.....	14-4
예제 - CSD7과 사용시 시스템 구성도.....	14-5
XGPC 설정 개요.....	14-5
확장 I/O 설정.....	14-7
MCST 명령어.....	14-9
MOVB 명령어.....	14-27
MSG를 사용한 COE SDO 설정.....	14-28
CPU 분산 처리를 이용한 6 축 이상의 시스템.....	14-32

## X8 CPU OS FRN 11.10

- ▣ CSD7N을 포함한 서보 드라이브 및 EtherCAT Slave 모듈을 지원
- ▣ EtherCAT Free Run 모드만 지원. EtherCAT DC (Distributed Clock) 모드는 지원하지 않는다.
- ▣ EtherCAT XML 은 사용하지 않음. COE PDO 매핑은 EtherCAT EEPROM 내용만으로 이루어 진다.
- ▣ COE SDO 설정
  - 각 Slave에 대한 COE SDO 설정은 RUN 모드의 시작에서 I/O 설정에 의해 이루어 진다.
  - 각 Slave에 대한 COE SDO 설정은 RUN 모드에서는 MSG 명령어에 의해 이루어 진다.
- ▣ 서보 드라이브를 테스트 하거나 모션 제어를 위한 프로파일을 생성하기 위해서는 MCST 명령어를 사용한다.

## 지원 가능한 EtherCAT Slave

다음 기기들이 동시에 지원된다

- ▣ X8 확장 I/O 모듈
- ▣ CSD7 서보 드라이브
  - 펌웨어 버전 V2.00 이후는 EtherCAT Free Run 모드를 지원
  - 펌웨어 버전 V2.00 이후는 PDO 매핑 테이블을 EtherCAT EEPROM 내에서 지원
- ▣ 다음 항목을 만족하는 일반적인 EtherCAT Slave 모듈
  - Free Run 모드가 지원되어야 한다.
  - PDO 매핑 테이블이 EtherCAT EEPROM 내에 있어야 한다

## 장점

### ▣타사 제품과 비교한 X8 EtherCAT 모션 기능의 장점

- 사용이 쉬움.
- 저렴한 가격.
- 간단한 Ladder 프로그래밍.

### ▣Pulse 타입과 비교한 X8 EtherCAT 모션 기능의 장점

- 간단한 하드웨어 케이블링.
- 고속 모션.
- 하드웨어 추가 없이 6 축 모션 지원.

## CSD7 서보 드라이브

EtherCAT EEPROM 의 PDO 매핑

PDO Map		Total Bytes	Size(byte)	Index	Name	Type
1th rxPDO Mapping (0x1600)	16	2	0x6040	Control word	UINT	
		4	0x607A	Target position	DINT	
		4	0x60FF	Target Velocity	DINT	
		2	0x6071	Target Torque	INT	
		1	0x6060	Mode of Operation	SINT	
		1	0x0000	Padding	SINT	
		2	0x60B8	Touch probe function	UINT	
		2	0x6041	Status word	UINT	
		4	0x6064	Position actual value	DINT	
		4	0x606C	Velocity actual value	DINT	
1th txPDO Mapping (0x1A00)	26	2	0x6077	Torque actual value	INT	
		1	0x6061	Modes of operation display	SINT	
		1	0x1001	Error Register	SINT	
		4	0x60F4	Following error actual value	DINT	
		2	0x60B9	Touch probe status	UINT	
		2	0x603F	Error Code	UINT	
2nd rxPDO Mapping (0x1601)	6	4	0x60FD	Digital inputs	UDINT	
		2	0x6040	Control word	UINT	
2nd txPDO Mapping (0x1A01)	24	4	0x607A	Target position	DINT	
		2	0x6041	Status word	UINT	
		4	0x6064	Position actual value	DINT	
		4	0x606C	Velocity actual value	DINT	
		2	0x6077	Torque actual value	INT	
		4	0x60F4	Following error actual value	DINT	
		2	0x60B9	Touch probe status	UINT	
		2	0x603F	Error Code	UINT	
4	0x60FD	Digital inputs	UDINT			

1<sup>st</sup>  
PDO  
(default)

2<sup>nd</sup>  
PDO

## 예제 - CSD7과 사용시 시스템 구성도



- ▣ X8-CPU OS 펌웨어 FRN 11.10 이후
- ▣ X8-EB2RJ 모듈 추가

- ▣ CSD7 OS 펌웨어 V2.00 이후
- ▣ EtherCAT Free Run 모드
- ▣ EtherCAT EEPROM에 의한 모듈 식별
- ▣ COE CSP 모드

- ▣ X8 확장 I/O 또는 일반적인 EtherCAT Slave 모듈
- ▣ EtherCAT Free Run 모드
- ▣ EtherCAT EEPROM에 의한 모듈 식별

## XGPC 설정 개요

- ▣ 단계 1) I/O 설정
  - 자동 I/O 찾기 또는 수동 설정
  - PDO 매핑 할당
  - Startup COE SDO 설정
- ▣ 단계 2) Y / X 데이터 테이블
  - RxPDO 와 TxPDO 는 자동적으로 각각 Y 와 X 데이터 테이블로 매핑됨
- ▣ 단계 3) 래더 프로그램
  - MCST 명령어를 사용한 서보 ON/OFF
  - MCST 명령어를 사용한 Jog
  - MCST 명령어를 사용한 Move
  - 서보 기능을 사용한 Homing

The image displays the Siemens SIMATIC Manager interface for configuring an X8 PID auto-tuning system. It is divided into three main sections:

### I/O Configuration

#	Part #	Input Words	Output Words	Data Tables	5V(mA)
0	X8-BASE	4	4	Y0.0, X1.0	2200
1	X8-EB2RJ	0	0		2050
2	CSD7_01BN1 100W	13	8	Y0.2, X1.2	2050
3	CSD7_01BN1 100W	13	8	Y0.3, X1.3	2050
4	CSD7_01BN1 100W	13	8	Y0.4, X1.4	2050
5	CSD7_01BN1 100W	13	8	Y0.5, X1.5	2050
6	X8-RJ2EB	1			
7	X8-XU16	1			
8	X8-YR16	0			

### MCST Ladder Logic

The ladder logic for MCST (Motion Control Servo Tuning) is shown in three rungs:

- Rung 0000:** Command Flags/Config Flags. EN MCST ENO. Parameters: 2 Slot, #L1400.0 ControlDataTbl, 80 ControlLength.
- Rung 0001:** Command Flags/Config Flags. EN MCST ENO. Parameters: 3 Slot, #L1401.0 ControlDataTbl, 80 ControlLength.
- Rung 0002:** Command Flags/Config Flags. EN MCST ENO. Parameters: 4 Slot, #L1402.0 ControlDataTbl, 80 ControlLength.

### MCST - L1400.0 : Program 402 Rung 0

The configuration window for the servo driver shows the following settings:

- General:** Exp. Slot Number: 0, Mode of Operation Display: 0, Current RX PDO Map: 0, Servo Error Code: 0h, Current TX PDO Map: 0.
- Configuration:** Move Type: Absolute, Stop Type: Hard, Jerk Type: units/sec<sup>3</sup>, Mode of Operation: 6, Software Negative Limit: -2000000 pulses, Ratio Numerator: 0, Software Positive Limit: 2000000 pulses, PDO Mem Copy: Auto, Ratio Denominator: 0, In Position Range: 10 units, In Position Hold Time: 0 scans.
- Monitoring:** Current Time: 0 usec. Command and Actual values for Position, Velocity, Accel, Decel, Jerk, and Torque.
- Command:** Servo ON, Set Position, Clear Fault, STOP, Jog -, Jog +, Move.
- Digital Input:** 32-bit input bus (0-31).
- Status Flags:** Servo ON/OFF, Set Position, Clear Servo Fault, Stop, Jog -, Jog +, Move, Enabled, Executing, Done, Error, Accel, Run, Decel, In Position, Profile Adjusted, Accumulated Position, Accel, Decel, Overflow, Underflow.

## 확장 I/O 설정

### 모든 Slave에 대한 I/O 설정

#	Part #	Input Words	Output Words	Data Tables	5V(mA)	24V(mA)	Error	Outputs	Module	WatchDog	Description
0	X8-BASE	4	4	Y0.0 , X1.0	2200	2000					X8-Modular Series A FRN 11
1	X8-EB2RJ	0	0		2050	2000					Convert Expansion Bus to RJ45
2	CSD7_01BN1 100W	13	8	Y0.2 , X1.2	2050	2000					EtherCAT Device : 2010001h
3	CSD7_01BN1 100W	13	8	Y0.3 , X1.3	2050	2000					EtherCAT Device : 2010001h
4	CSD7_01BN1 100W	13	8	Y0.4 , X1.4	2050	2000					EtherCAT Device : 2010001h
5	CSD7_01BN1 100W	13	8	Y0.5 , X1.5	2050	2000					EtherCAT Device : 2010001h
6	X8-RJ2EB	1	0	X1.6	2200	2000					Convert RJ45 to Expansion Bus
7	X8-XU16	1	0	X1.7	2050	2000					16 DC Input
8	X8-YR16	0	1	Y0.8	1890	1850		Keep			16 Relay Output
9											
10											
11											
12											
13											
14											
15											
16											
17											
18											
19											
20											

### 각 EtherCAT Slave 에 대한 설정

- PDO Member Offset 을 변경하였을 경우, 서보 드라이브의 전원을 재 인가해야 할 수 있음.

Slot #2 : ECAT-DEVICE - EtherCAT Device : 2010001h	
Expansion General Configuration   Startup COE SDO Configuration   CSD7N	
Module	ECAT-DEVICE - EtherCAT Device
Vendor ID	829 (33Dh) [RS Automation Co., Ltd.]
Product Type	4096
Product Code	33619969 (2010001h) [CSD7_01BN1 100W]
Revision Number	0 (0h)
Series	A
Input Words	13
Input Bits	0
Output Words	8
Output Bits	0
Config Data Length in Words	112
Ignore Configuration Error	False
Keep outputs to the last state while non-executing mode	False
Virtual Module	False
<b>PDO Member</b>	
Rx PDO Member Offset	0
Tx PDO Member Offset	0

### COE SDO 설정 – Homing Parameter 설정 예

- \* 항목이 "Enabled" 에 체크가 되지 않으면, 설정되지 않는다.
- \* Hexadecimal 값을 입력할 경우에는 숫자 뒤에 'h'를 입력한다.

Slot #2 : ECAT-DEVICE - EtherCAT Device : 2010001h

Expansion General Configuration | Startup COE SDO Configuration | CSD7N

Copy Row(s) | Paste Row(s) | Clear Row(s) | Copy All Rows | Paste All Rows | Clear All Rows

No	Enable	Index	SubIndex	BitLength	Data
0	1: Enable	24728 (6098h)	0 (00h)	8 (0008h)	19 (00000013h)
1	1: Enable	24700 (607Ch)	0 (00h)	32 (0020h)	0 (00000000h)
2	1: Enable	24729 (6099h)	1 (01h)	32 (0020h)	60000 (0000EA60h)
3	1: Enable	24729 (6099h)	2 (02h)	32 (0020h)	60000 (0000EA60h)
4	1: Enable	24730 (609Ah)	0 (00h)	32 (0020h)	
5	1: Enable	9486 (250Eh)	0 (00h)		
6	1: Enable	9488 (2510h)	0 (00h)		
7	0: Disable	0 (0000h)	0 (00h)		
8	0: Disable	0 (0000h)	0 (00h)		
9	0: Disable	0 (0000h)	0 (00h)		
10	0: Disable	0 (0000h)	0 (00h)		
11	0: Disable	0 (0000h)	0 (00h)		
12	0: Disable	0 (0000h)	0 (00h)		
13	0: Disable	0 (0000h)	0 (00h)		
14	0: Disable	0 (0000h)	0 (00h)		
15	0: Disable	0 (0000h)	0 (00h)		
16	0: Disable	0 (0000h)	0 (00h)		
17	0: Disable	0 (0000h)	0 (00h)		
18	0: Disable	0 (0000h)	0 (00h)		
19	0: Disable	0 (0000h)	0 (00h)		

0 : 기본값  
1 : EEPROM 내의 첫 번째 PDO 매핑

Homing Parameters			
Index	SubIndex	BitLength	Data
0x6098	0	8	0x00000013
0x607C	0	32	0x00000000
0x6099	1	32	0x0000EA60
0x6099	2	32	0x0000EA60
0x609A	0	32	0x000927C0
0x250E	0	32	0x00000000
0x2510	0	16	0x0000001E

■TxPDO 는 X 데이터 테이블로 매핑된다

Data Table : X1 -- INPUT

ON/OFF | FORCES | ALL CLEAR | CLEAR | ON | OFF | 10 | 16 | 2 | ASC |

Address	Type	Value	Meaning	Description
X1.0.0	Input	0000 0000 0000 0000	Bits, Base	
X1.0.1	Input	0000 0000 0000 0000	Bits, Base	
X1.0.2	Input	0000 0000 0000 0000	Bits, Base	
X1.0.3	Input	0000 0000 0000 0000	Bits, Base	
X1.2.0	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.1	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.2	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.3	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.4	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.5	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.6	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.7	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.8	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.9	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.10	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.11	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.2.12	Input	0000 0000 0000 0000	Bits, Slot #2	
X1.3.0	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.1	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.2	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.3	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.4	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.5	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.6	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.7	Input	0000 0000 0000 0000	Bits, Slot #3	
X1.3.8	Input	0000 0000 0000 0000	Bits, Slot #3	

▣RxPDO 는 Y 데이터 테이블로 매핑된다

Address	Type	Value	Meaning	Description
Y0.0.0	Output	0000 0000 0000 0000	Bits, Base	
Y0.0.1	Output	0000 0000 0000 0000	Bits, Base	
Y0.0.2	Output	0000 0000 0000 0000	Bits, Base	
Y0.0.3	Output	0000 0000 0000 0000	Bits, Base	
Y0.2.0	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.2.1	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.2.2	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.2.3	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.2.4	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.2.5	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.2.6	Output	0000 0000 0000 0110	Bits, Slot #2	
Y0.2.7	Output	0000 0000 0000 0000	Bits, Slot #2	
Y0.3.0	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.3.1	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.3.2	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.3.3	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.3.4	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.3.5	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.3.6	Output	0000 0000 0000 0110	Bits, Slot #3	
Y0.3.7	Output	0000 0000 0000 0000	Bits, Slot #3	
Y0.4.0	Output	0000 0000 0000 0000	Bits, Slot #4	
Y0.4.1	Output	0000 0000 0000 0000	Bits, Slot #4	
Y0.4.2	Output	0000 0000 0000 0000	Bits, Slot #4	
Y0.4.3	Output	0000 0000 0000 0000	Bits, Slot #4	
Y0.4.4	Output	0000 0000 0000 0000	Bits, Slot #4	
Y0.4.5	Output	0000 0000 0000 0000	Bits, Slot #4	

## MCST 명령어

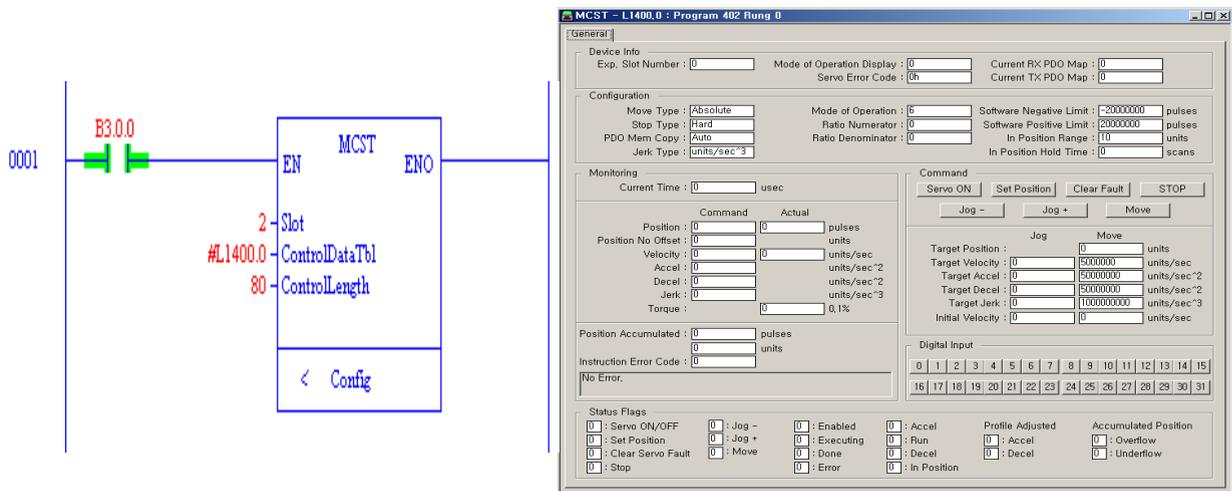
▣MCST(Motion Control Servo Test) 명령어는 서보 드라이브를 테스트하는 데 사용될 수 있다. 다음의 기능을 수행 할 수 있다.

- Position, Velocity, Acceleration, Deceleration, Jerk, Initial velocity 에 의한 Profile 생성
  - Servo ON/OFF
  - Clear Servo Fault
  - Set Position
  - Jog-, Jog+
  - Move Absolute, Move Relative
  - Hard Stop, Controlled Stop
  - In Position, In Velocity
  - Position Unit Conversion
  - Negative, Positive Software Limit
  - Negative, Positive Hardware Limit (OS FRN 12.04 or Higher)
  - Gearing
  - Teaching
  - Monitoring variables

■Homing 은 이 명령어 내에서 직접 지원하지 않는다. 하지만, 서보 드라이브 내의 Homing 기능을 사용할 수 있다.

■MCST 명령어는 Slot, ControlDataTbl, ControlLength 등 3개의 오퍼랜드를 가진다.

- 설정/명령/상태 창은 명령어 상의 "Config" 를 클릭하면 호출된다.



■ 관련 파라미터 - Instruction

- Ele 1, Bit 16 : [S] Instruction Enabled
- Ele 1, Bit 17 : [S] Instruction Executing
- Ele 1, Bit 18 : [S] Instruction Done
- Ele 1, Bit 19 : [S] Instruction Error
- Ele 1, Bit 20 : [S] Instruction Accel Executing
- Ele 1, Bit 21 : [S] Instruction Run Executing
- Ele 1, Bit 22 : [S] Instruction Decel Executing
- Ele 1, Bit 24 : [S] Profile Adjusted Accel
- Ele 1, Bit 25 : [S] Profile Adjusted Decel
- Ele 1, Bit 26 : [S] Accumulated Position Overflow
- Ele 1, Bit 27 : [S] Accumulated Position Underflow

#### ■ 관련 파라미터 - Instruction

- Ele 2, Uint16 : ExpansionSlot, [S] Expansion I/O Slot Number. 1, 2,..., 96
- Ele 3, Uint16 : RxPdoMapCurrent, [S] Rx PDO Map Current. 0, 1, 2,... for CSD7N
- Ele 4, Uint16 : TxPdoMapCurrent, [S] Tx PDO Map Current. 0, 1, 2,... for CSD7N
- Ele 50, Sint32 : PositionCmd,[S] Position Command No Offset, in units
- Ele 51, Sint32 : VelocityCmd,[S] Velocity Command, in units per sec
- Ele 52, Sint32 : AccelCmd,[S] Accel Command, in units per sec<sup>2</sup>
- Ele 53, Sint32 : DecelCmd,[S] Decel Command, in units per sec<sup>2</sup>
- Ele 54, Sint32 : JerkCmd,[S] Jerk Command, in units per sec<sup>3</sup>
- Ele 58, Sint32 : PositionCmdAccumulated,[S] Position Command Accumulated, in pulses
- Ele 59, Sint32 : PositionCmdAccumulatedInUnits,[S] Position Command Accumulated, in units

#### ■ 관련 파라미터 - Servo ON/OFF

- Ele 0, Bit 0 : [C] Servo ON/OFF Command
- Ele 1, Bit 0 : [S] Servo ON/OFF Executing Status

#### ■ 기능 - Servo ON/OFF

- 서보 드라이브를 동작하기 전에, 서보 드라이브는 "Operation enabled" 되어야 한다.
- Operation enable 하기 위해, Servo ON/OFF Command 비트를 set 한다.
- Operation disable 하기 위해, Servo ON/OFF Command 비트를 clear 한다.

■ 관련 파라미터 - Clear Servo Fault

- Ele 0, Bit 2 : [C] Clear Servo Fault Command
- Ele 1, Bit 2 : [S] Clear Servo Fault Executing Status
- Ele 1, Bit 19 : [S] Instruction Error
- Ele 40, Uint16 : ErrorCodeInst, [S] Instruction Error Code

■ 기능 - Clear Servo Fault

- MCST 명령어에서 에러가 발생할 경우, Instruction Error 비트와 ErrorCodeInst 에 값이 설정된다.
- 서보 드라이브나 명령어에서의 에러를 초기화하기 위하여, Clear Servo Fault Command 비트를 set 한다.

■ 관련 파라미터 - Set Position

- Ele 0, Bit 3 : [C] Set Position Command
- Ele 1, Bit 3 : [S] Set Position Executing Status
  - Ele 58, Sint32 : PositionCmdAccumulated,[S] Position Command Accumulated, in pulses
  - Ele 59, Sint32 : PositionCmdAccumulatedInUnits,[S] Position Command Accumulated, in units

■ 기능 - Set Position

- PositionCmdAccumulated 를 초기화 한다.
- Homing 이 끝난 후 실행하여야 한다. Set Position 은 기준 위치를 정하는 기능이기 때문에 Homing 이 끝난 후 한번만 실행하여야 한다. 빈번한 실행은 기준 위치를 변경하게 한다.
- Move 나 Jog 가 실행 중이면 이 명령은 실행되지 않는다.

#### ■ 관련 파라미터 - Jog-, Jog+

- Ele 0, Bit 4 : [C] Jog- Command
- Ele 0, Bit 5 : [C] Jog+ Command
- Ele 1, Bit 4 : [S] Jog- Executing Status
- Ele 1, Bit 5 : [S] Jog+ Executing Status
- Ele 21, Sint32 : VelocityTargetJog, [C] Jog, Target Velocity, in units per sec
- Ele 22, Sint32 : AccelTargetJog, [C] Jog, Target Acceleration, in units per sec<sup>2</sup>
- Ele 23, Sint32 : DecelTargetJog, [C] Jog, Target Deceleration, in units per sec<sup>2</sup>
- Ele 24, Sint32 : JerkTargetJog, [C] Jog, Target Jerk, in units per sec<sup>3</sup>
- Ele 25, Sint32 : VelocityInitialJog, [C] Jog, Initial Velocity, in units per sec

#### ■ 기능 - Jog-, Jog+

- Jog 기능을 사용하면 구동 부를 움직이게 할 수 있다.

#### ■ 관련 파라미터 - Move Absolute, Move Relative

- Ele 0, Bit 8 : [C] Move Command
- Ele 1, Bit 8 : [S] Move Executing Status
- Ele 0, Bit 16 : [C] Relative Bit, Absolute(0) or Relative(1)
- Ele 10, Sint32 : PositionTarget, [C] Move, Target Position, in units
- Ele 11, Sint32 : VelocityTarget, [C] Move, Target Velocity, in units per sec
- Ele 12, Sint32 : AccelTarget, [C] Move, Target Acceleration, in units per sec<sup>2</sup>
- Ele 13, Sint32 : DecelTarget, [C] Move, Target Deceleration, in units per sec<sup>2</sup>

- Ele 14, Sint32 : JerkTarget, [C] Move, Target Jerk, in units per sec<sup>3</sup>
- Ele 15, Sint32 : VelocityInitial, [C] Move, Initial Velocity, in units per sec

▣기능 - Move Absolute, Move Relative

- Move 기능을 사용하면 구동 부를 움직이게 할 수 있다.

▣관련 파라미터 - Stop

- Ele 0, Bit 10 : [C] Stop Command
- Ele 1, Bit 10 : [S] Stop Executing Status
- Ele 0, Bit 18 : [C] Controlled Stop Bit, Hard Stop(0) or Controlled Stop(1)

▣기능 - Stop

- STOP 명령을 사용하여 구동 부를 멈추게 할 수 있다.
- Controlled Stop 비트가 set 되어 있는 경우, 구동 부는 즉시 멈추지 않고 설정된 감속 후 멈춘다.
- Stop 이 실행된 후 Move 나 Jog 명령 비트는 clear 된다.

▣관련 파라미터 - In Position

- Ele 1, Bit 28 : [S] In Position Bit
- Ele 6, Uint16 : InPositionRange, [C] In Position Range, in units
- Ele 7, Uint16 : InPositionHoldTime, [C] In Position Hold Time, in scans
- Ele 7, Uint16 : InPositionHoldTimeStatus, [S] In Position Hold Time Status, in scans

#### ■기능 - In Position

- Move 가 끝난 후, Done 비트가 set 되고 Target Position 과 Actual Position 의 차이가 설정된 값 (InPositionRange) 내에 있고 설정된 시간 (InPositionHoldTime) 동안 지속되면 In Position 비트가 set 된다.
- In Position 비트는 Move 가 끝난 후, Move 명령이 clear 될 때 clear 된다.

#### ■관련 파라미터 - In Velocity

- Ele 1, Bit 29 : [S] In Velocity Bit
- Ele 26, Uint16 : InVelocityRange, [C] In Velocity Range, in units
- Ele 27, Uint16 : InVelocityHoldTime, [C] In Velocity Hold Time, in scans
- Ele 27, Uint16 : InVelocityHoldTimeStatus, [S] In Velocity Hold Time Status, in scans

#### ■기능 - In Velocity

- Move 가 진행 중일 동안, Target Velocity 와 Actual Velocity 의 차이가 설정된 값(InVelocityRange )내에 있고 설정된 시간(InVelocityHoldTime ) 동안 지속되면 In Velocity 비트가 set 된다.

#### ■관련 파라미터 - Position Unit Conversion

- Ele 8, Sint32 : RatioNumerator, [C] Ratio Numerator
- Ele 9, Sint32 : RatioDenominator, [C] Ratio Denominator

#### ■기능 - Position Unit Conversion

- Position 값은 설정된 비율에 따라서 변경 될 수 있다.
- $\text{Ratio} = \text{RatioNumerator} / \text{RatioDenominator}$
- $\text{Position} = \text{Position} \times \text{Ratio}$

- Position 값에 각도가 사용된다면, 다음과 같이 파라미터를 설정한다.
  - RatioNumerator = 131072 (17 bit encoder), 8388608 (23 bit encoder)
  - RatioDenominator = 360, 3600, 10000 or ...

▣ 관련 파라미터 - Software Limit

- Ele 18, Sint32: SoftwareLimitNegative, [C] Software Negative Limit, in pulses
- Ele 19, Sint32: SoftwareLimitPositive, [C] Software Positive Limit, in pulses

▣ 기능 - Software Limit

- Software Limit 은 Pulse 값으로 설정된다.

▣ 관련 파라미터 - Hardware Limit

- Ele 0, Bit 29 : [C] Disable(0), Enable(1) Stop at Hardware Limit
- Ele 0, Bit 30 : [C] Active Low(0), Active High(1) for Negative Hardware Limit
- Ele 0, Bit 31 : [C] Active Low(0), Active High(1) for Positive Hardware Limit

▣ 기능 - Hardware Limit

- OS FRN 12.04 or Higher

▣ 관련 파라미터 - Gearing

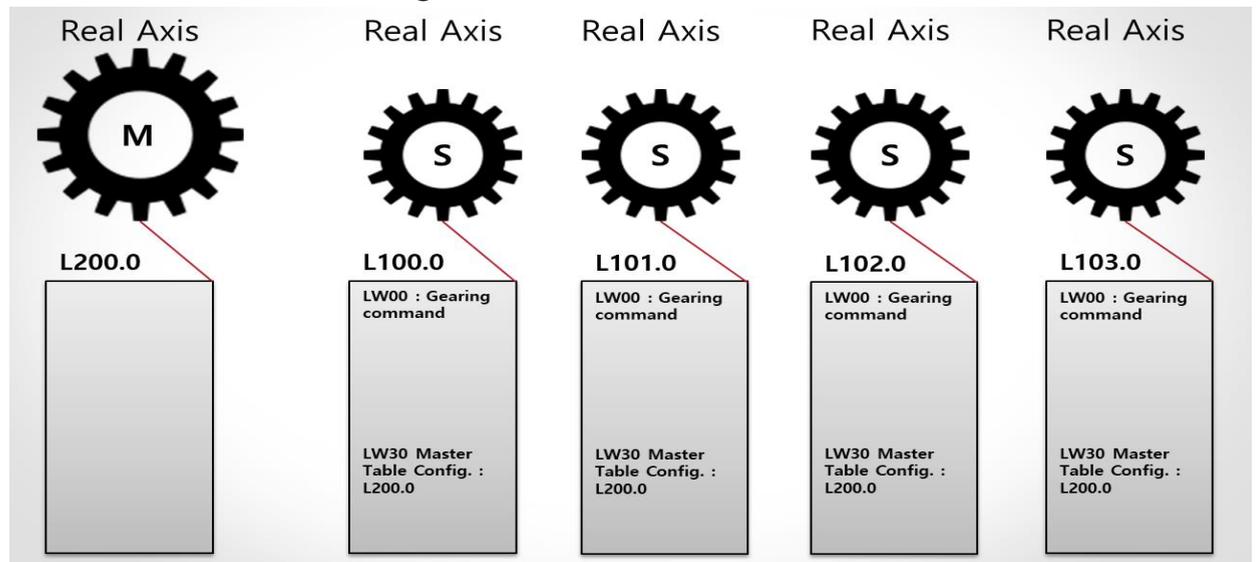
- Ele 0, Bit 12 : [C] Gearing Command
- Ele 1, Bit 12 : [S] Gearing Execution Status
- Ele 0, Bit 19 : [C] Gearing to Master Command Pos(0), Actual Pos(1)
- Ele 28, Sint32 : GearingRatioNumerator, [C] Gearing Ratio Numerator

- Ele 29, Sint32 : GearingRatioDenominator, [C] Gearing Ratio Denominator
- Ele 30, Uint32 : MasterControlDataTbl, [C] Master Control Data Table

#### ■기능 - Gearing

- 마스터 축은 설정된 Position, Velocity, Acceleration, Deceleration, Jerk 로 움직이고 있을 때, 추종 축은 설정된 비율로 마스터 축을 추종한다.
  - $GearingRatio = \frac{GearingRatioNumerator}{GearingRatioDenominator}$
  - $FollowerAxisPosition = MasterAxisPosition \times GearingRatio$

#### ■Gearing



#### ■관련 파라미터 - Teaching

- Ele 0, Bit 13 : [C] Teaching Command
- Ele 1, Bit 13 : [S] Teaching Executing Status
- Ele 31, Uint32 : TeachingPositionDataTbl, [C] Teaching Position Data Table
- Ele 32, Uint16 : TeachingPositionLength, [S] Length of Teaching Position

■기능 - Teaching

•Teaching 기능을 사용하여, 명령 비트를 셋하여 현재의 Position 을 수집할 수 있다.

과정은 아래와 같다.

1. Homing 후, Position Command Accumulated 를 Reset 한다.
2. 축을 원하는 만큼 움직인다.
3. Teaching Command bit 를 Set 한다. 그러면, Length of Teaching Position 는 값이 증가하고, Position Command Accumulated 의 값은 설정된 Teaching Position Data Table 로 복사된다.
4. Length of Teaching Position 이 Teaching Position Data Table 의 엘리먼트 개수만큼 증가하게 된다. Length 가 최대 엘리먼트 개수와 같으면 더 이상 실행되지 않는다.
5. 티칭 명령이 실행될 때 서보는 ON 또는 OFF 일 수 있다.

■관련 파라미터 - PDO Mem Copy

- Ele 0, Bit 17 : [C] PDO Mem External Copy Bit, PDO Mem Auto Copy(0), PDO Mem External Copy(1)

■기능 - PDO Mem Copy

- 이 비트가 clear 이고 서보 드라이브가 RSA CSD7N 이면, PDO 메모리는 자동적으로 복사된다.
- 메모리 맵이 다를 경우, 이 비트를 set 하고 ladder program 을 사용하여 다음과 같이 복사한다.

X Data Table => Element 70-79 in L Control Data Table of MCST instruction

Y Data Table <= Element 60-69 in L Control Data Table of MCST instruction

#### ■ 관련 파라미터 - Keep Position Accumulated

- Ele 0, Bit 20 : [C] Keep Position Accumulated Bit, Init Pos Accumulated at First Scan(0), Keep(1)

#### ■ 기능 - Keep Position Accumulated

- 이 비트가 Set 일 경우, Position Accumulated 엘리먼트 값이 MCST명령어의 시작에서 초기화되지 않고 유지된다. 이 기능은 절대치 엔코더를 사용할 경우 Set 하여야 한다.

#### ■ 관련 파라미터 - Jerk Unit % of Time

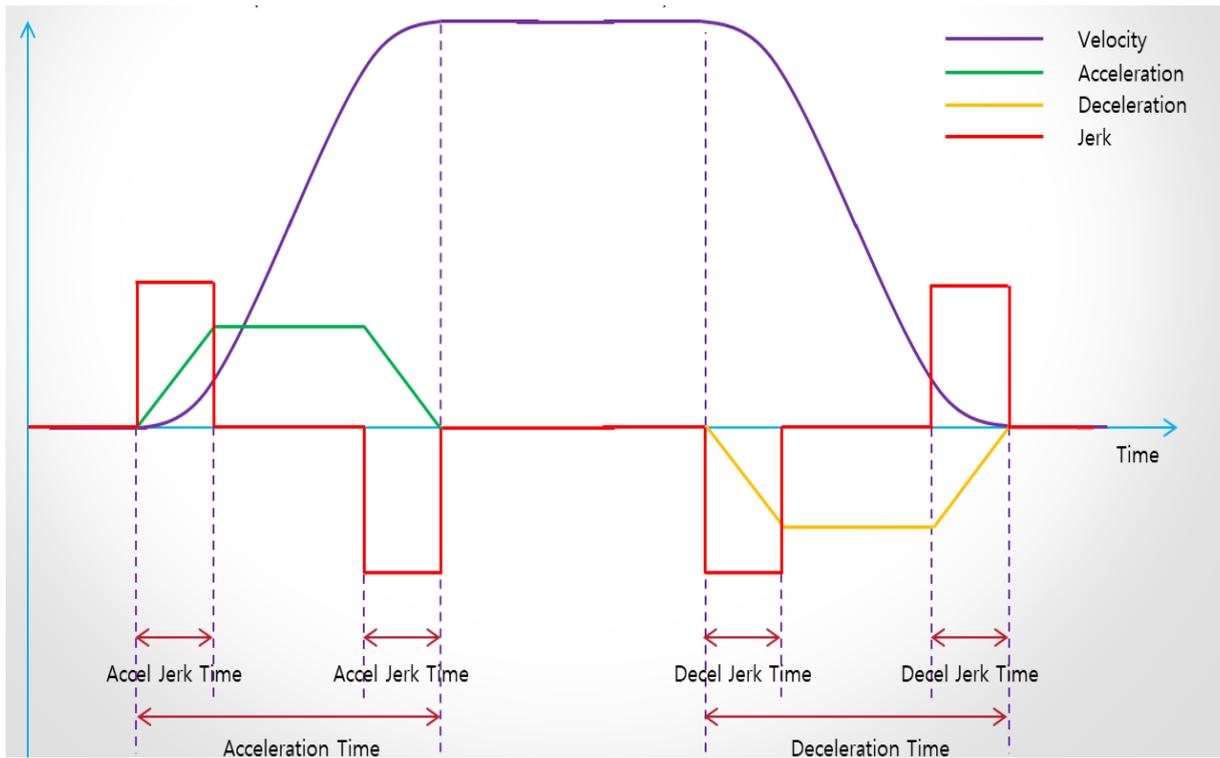
- Ele 0, Bit 21 : [C] Jerk Unit % of Time Bit, Jerk is units per  $\text{sec}^3$ (0), 0.1 % of time(1)
  - Ele 14, Sint32 : JerkTarget, [C] Move, Target Jerk, in 0.1 % of time
  - Ele 24, Sint32 : JerkTargetJog, [C] Jog, Target Jerk, in 0.1 % of time

#### ■ 기능 - Jerk Unit % of Time

- 이 비트를 사용하게 되면, 사용자는 또 다른 방법으로 Jerk 값을 설정할 수 있다.
- 이 비트가 clear 되면, Jerk 단위는  $\text{units per sec}^3$  이다.
- 이 비트가 사용자에게 의해 set 되면, Jerk 단위는 0.1 % of time 이다. 이 경우, Jerk 값은 Acceleration 이나 Deceleration time 의 % 동안의 값으로 환산된다.

▣ Jerk Unit 0.1 % of Time

- For Acceleration,  
 $0.1 \% \text{ of Time} = (\text{Accel Jerk Time} / \text{Acceleration Time}) * 100 * 10$
- For Deceleration,  
 $0.1 \% \text{ of Time} = (\text{Decel Jerk Time} / \text{Deceleration Time}) * 100 * 10$



▣ 변수 모니터링

- Actual Position
- Actual Velocity
- Actual Torque
- Following Error Actual Value
- Digital Inputs

▣ Mode of Operation

- 지원하는 Mode of Operation 은 아래와 같음
  - 6 : Homing Mode
  - 8 : CSP(Cycle Synchronization Position) Mode
  - 9 : CSV(Cycle Synchronization Velocity) Mode
  - 10 : CST(Cycle Synchronization Torque) Mode, FRN 12.00 or later

- CSV 모드로 사용하기 위해서는, CSV 모드를 위한 target velocity 값을 Control Data Table 의 엘리먼트 62 에 직접 Write 하여야 한다.
- CST 모드로 사용하기 위해서는, CST 모드를 위한 target torque 값을 Control Data Table 의 엘리먼트 63 에 직접 Write 하여야 한다.
- Configuration의 Mode of Operation 이 서보 드라이브에서 문제없이 설정되면, Device Info의 Mode of Operation Display 에 동일한 값이 표시됨.
- 만약 X8 CPU 가 Mode of Operation을 지원하지 않는 버전이면, MCST Error Code 5 가 발생함.

#### ▣Rotary 축

- CPU OS FRN 12.00 또는 그 이상에서 지원함
- 축의 3 가지 타입

Rotary Positive Path Axis Type

Rotary Negative Path Axis Type

Rotary Shortest Path Axis Type

- 관련 파라미터

Ele 0, Bit 24 : [C] Rotary Axis Bit - When this bit is set by user, the axis is considered as a Rotary axis. Linear Axis(0), Rotary Axis(1).

Ele 0, Bit 25 : [C] Rotary Shortest Path Bit - When this bit is set by user, the axis moves to shortest path. None(0), Rotary Shortest Path(1).

Ele 0, Bit 26 : [C] Rotary Negative Path Bit - When this bit is set by user, the axis moves to negative path. Rotary Positive Path(0), Rotary Negative Path(1).

Ele 16, Sint32 : [C] Move, Rotary Position Unwind, in units

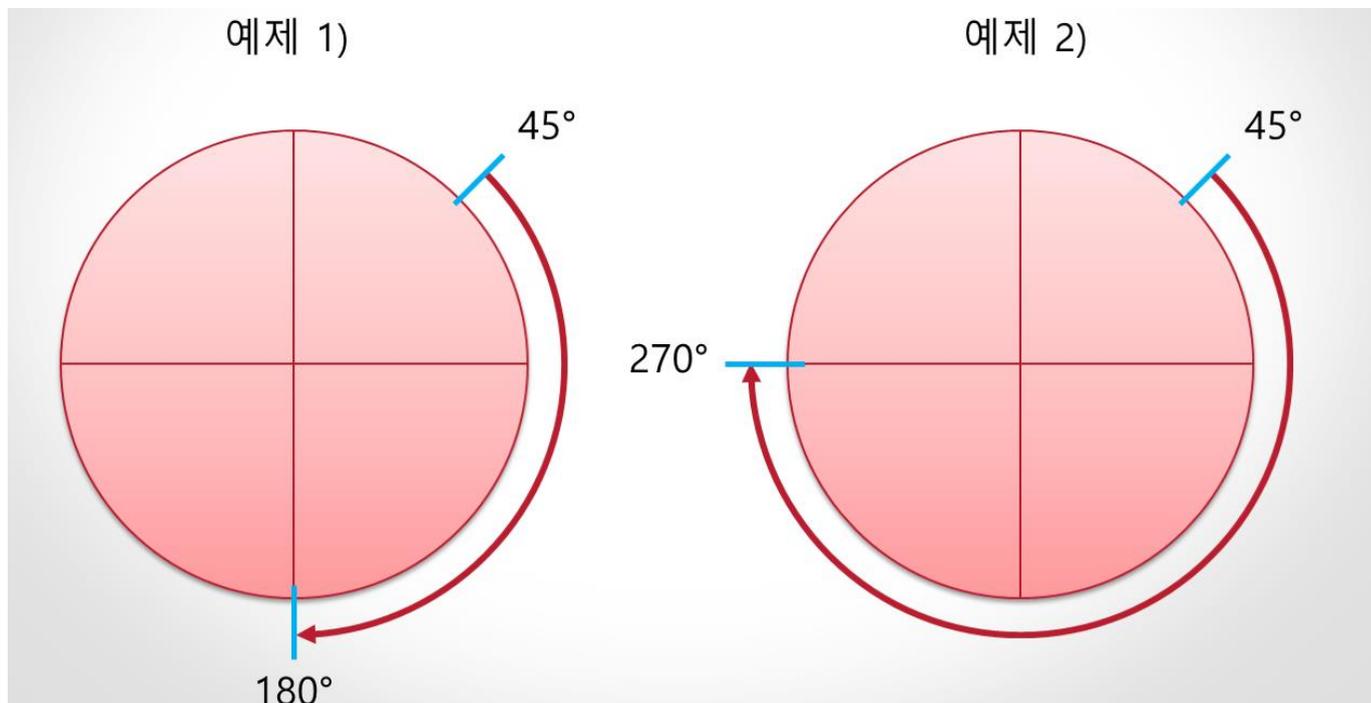
▣Rotary 축

- 라미터 간의 올바른 값

Parameters	Linear Axis	Rotary Positive Path Axis Type	Rotary Negative Path Axis Type	Rotary Shortest Path Axis Type
Ele 0, Bit 24 : [C] Rotary Axis Bit	0	1	1	1
Ele 0, Bit 25 : [C] Rotary Shortest Path Bit	0	0	0	1
Ele 0, Bit 26 : [C] Rotary Negative Path Bit	0	0	1	0
Ele 16, Sint32 : [C] Rotary Position Unwind	Don't care	Positive Value, >= Target Position	Positive Value, >= Target Position	Positive Value, >= Target Position

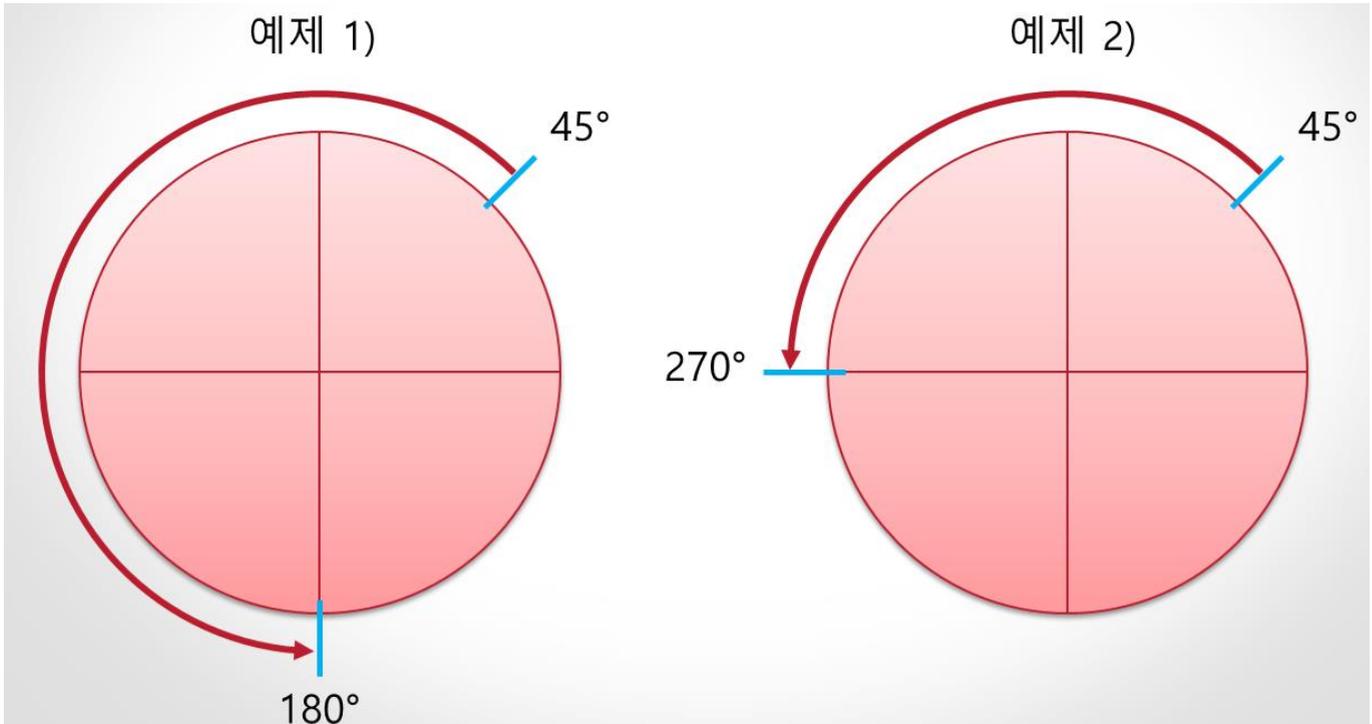
▣Rotary Axis

- Rotary Positive Path Axis Type
- 항상 (+) 방향으로 움직임



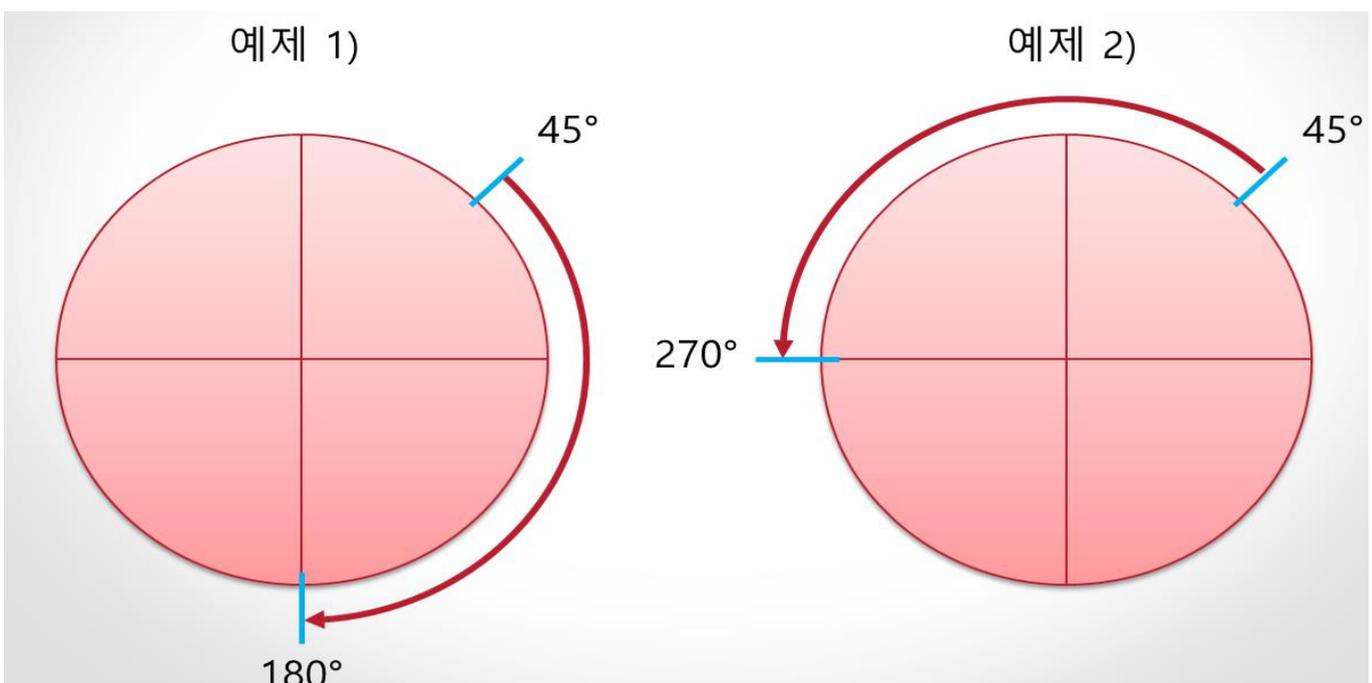
▣ Rotary Axis

- Rotary Positive Path Axis Type
- 항상 (-) 방향으로 움직임



▣ Rotary Axis

- Rotary Shortest Path Axis Type
- 짧은 경로 방향으로 움직임.



▣Rotary Axis

- In OS CPU FRN 12.00, Target Position cannot be the same as Position Unwind.
- In OS CPU FRN 12.01 , if (Target Position == Position Unwind),

For Rotary Positive Path Axis Type, the axis moves to Positive path as much as Position Unwind value.

For Rotary Negative Path Axis Type, the axis moves to Negative path as much as Position Unwind value.

For Rotary Shortest Path Axis Type, the axis does not move.

▣서보 기능을 사용한 Homing

- MCST 명령어 내에는 자체적인 Homing 기능이 내장되어 있지 않다. 하지만, MCST 명령어를 사용하여, 서보 기능을 사용한 Homing 을 할 수 있다.

- 관련 파라미터

Ele 64, [C] Mode Of Operation, uint8 :  
6:Homing

Ele 74, [S] Mode Of Operation Display, uint8 :  
6:Homing

Ele 60, Bit 4, [C] Control Word - Operation Mode Specific  
: Homing Start/Stop

Ele 70, Bit 10, [S] Status Word - Target Reached  
: Homing Done

Ele 70, Bit 13, [S] Status Word - Operation Mode Specific:  
Homing Error

- Homing Method, Homing Speed 등, Homing Parameter 는 I/O 설정의 Startup COE SDO Configuration 에서 설정된다. 이 과정은 제어기 모드가 RUN 으로 전환되기 전에 이루어져야 한다.
- Homing Parameter 는 MSG 명령어를 사용하여 RUN 중 에 설정 할 수도 있다

### ■ 확장 I/O Homing Parameter 설정

- \* 항목이 "Enabled" 에 체크가 되지 않으면, 설정되지 않는다.
- \* Hexadecimal 값을 입력할 경우에는 숫자 뒤에 'h'를 입력한다.

The screenshot shows the 'Slot #6 : ECAT-DEVICE - EtherCAT Device : 2010001h' configuration window. The 'Startup COE SDO Configuration' tab is active, displaying a table of parameters. A blue arrow points from the 'Homing Parameters' table below to the corresponding rows in the software interface.

No	Enable	Object Parameter	Data
0	1 : Enable	0x6098, (8) ECAT Homing Method	19 (00000013h)
1	1 : Enable	0x607C, (32) ECAT Homing Offset	0 (00000000h)
2	1 : Enable	0x6099:01, (32) Speed During Search for Switch	60000 (0000EA60h)
3	1 : Enable	0x6099:02, (32) Speed During Search for Zero	60000 (0000EA60h)
4	1 : Enable	0x609A, (32) ECAT Homing Acceleration	600000 (000927C0h)
5	1 : Enable	0x250E, (32) ECAT Abs Origin Offset	0 (00000000h)
6	1 : Enable	0x2510, (16) ECAT Homing Timeout	30 (0000001Eh)
7	0 : Disable	-	0 (00000000h)
8	0 : Disable	-	0 (00000000h)
9	0 : Disable	-	0 (00000000h)
10	0 : Disable	-	0 (00000000h)
11	0 : Disable	-	0 (00000000h)
12	0 : Disable	-	0 (00000000h)
13	0 : Disable	-	0 (00000000h)
14	0 : Disable	-	0 (00000000h)
15	0 : Disable	-	0 (00000000h)
16	0 : Disable	-	0 (00000000h)
17	0 : Disable	-	0 (00000000h)
18	0 : Disable	-	0 (00000000h)
19	0 : Disable	-	0 (00000000h)

Homing Parameters			
Index	SubIndex	BitLength	Data
0x6098	0	8	0x00000013
0x607C	0	32	0x00000000
0x6099	1	32	0x0000EA60
0x6099	2	32	0x0000EA60
0x609A	0	32	0x000927C0
0x250E	0	32	0x00000000
0x2510	0	16	0x0000001E

### ■ 에러 코드

- 0 : No Error.
- 1 : Slot is invalid.
- 2 : Tx PDO Map is invalid.
- 3 : Rx PDO Map is invalid.
- 4 : Servo drive caused an error.
- 5 : Mode of Operation is invalid.
- 10 : Invalid Velocity was configured. Velocity shall be larger than initial velocity configured.
- 11 : Invalid Software Limit was configured. Positive Software Limit shall be larger than Negative Software Limit configured. Position Accumulated shall be in the range of Software Limit.
- 12 : Invalid Denominator was configured. Denominator shall not be 0.

13 : Invalid Numerator was configured. Numerator shall not be 0.

20 : Invalid Master Control Data Table configured. Data Table Type shall be L only.

21 : Invalid Master Control Data Table Length. Master Control Data Table Length is 80 in elements.

22 : Invalid Gearing Denominator was configured. Gearing Denominator shall not be 0.

30 : Invalid Teaching Position Data Table configured. Data Table Type shall be L only.

31 : Invalid Teaching Position Data Table Length. Maximum Teaching Position Data Table Length is 1536 in elements.

32 : Invalid Teaching Position Length. Maximum Teaching Position Length is 1536 in elements.

90 : Invalid Parameter configured in the reserved elements.

100 : Stopped by Hard Stop.

101 : Stopped by Controlled Stop.

110 : Moving part reached Negative Software Limit during operation.

111 : Moving part reached Positive Software Limit during operation.

■ Error Codes – CPU OS FRN 12.00 or later

40 : Invalid Rotary Configuration. For Rotary, Move Type shall be Absolute.

41 : Invalid Rotary Configuration. For Rotary, Position Unwind value shall be positive.

42 : Invalid Rotary Configuration. For Rotary, Shortest path and Negative path cannot be set together.

43 : Invalid Rotary Configuration. For Rotary, Position Target value shall be positive value or less than Position Unwind value.

■ Error Codes – CPU OS FRN 12.06 or later

19 : Cannot generate the profile with the configuration.

120 : Moving part reached Negative Hardware Limit during operation.

121 : Moving part reached Positive Hardware Limit during operation.

130 : Multiple commands defined.

## MOVB 명령어

▣MOVB 명령어는 워드나 더블-워드 간 비트들을 복사하는 데 사용된다.

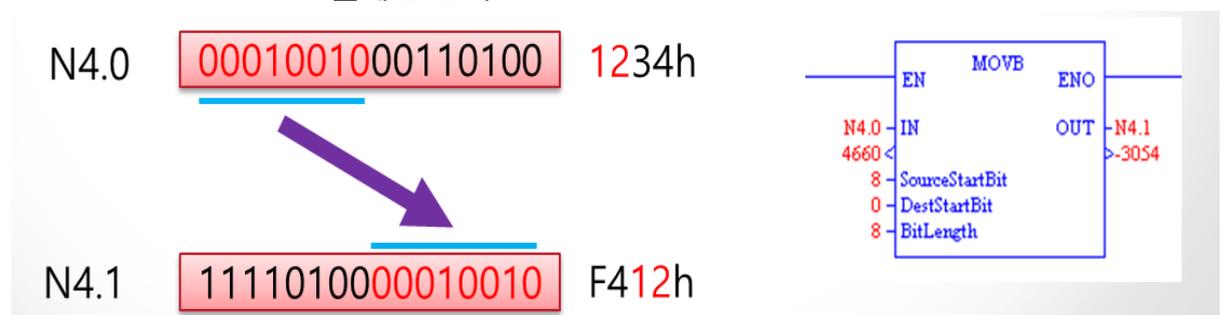
▣MOVB 명령어는 워드나 더블-워드 내에서 비트 시프트 하는 데 사용된다.

▣MOVB 명령어는 아래와 같은 데이터 복사에 사용될 수 있다.

- X 데이터 테이블 to N 데이터 테이블
- N 데이터 테이블 to Y 데이터 테이블

▣EtherCAT Slave 에 따라서, PDO 맵이 바이트 단위로 구성되어 질 수 있다. 이 경우 MOVB 명령어로 바이트 복사에 사용할 수 있다.

▣예를 들어:



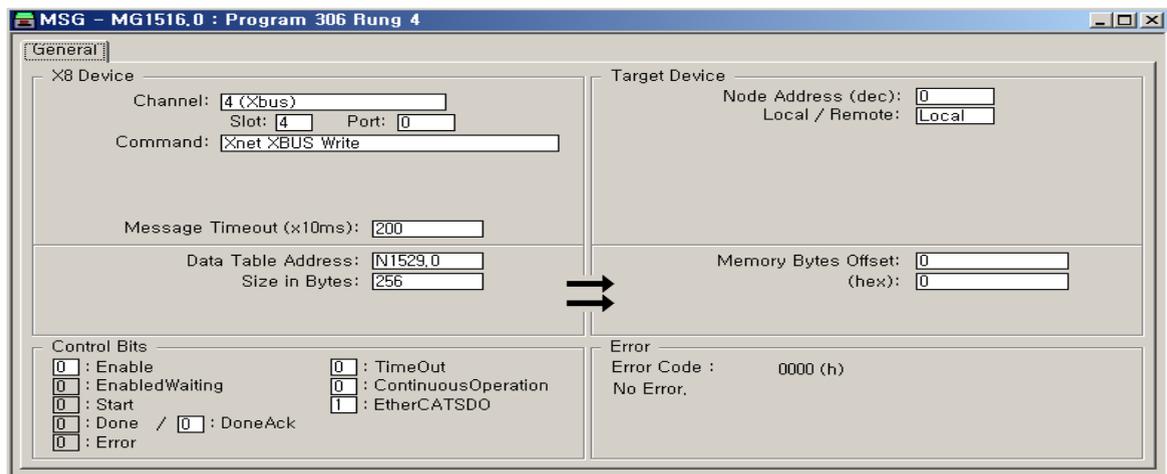
- Source 는 변경되지 않는다.
- Destination에서, 설정된 비트들을 제외한 나머지 비트들은 변경되지 않는다.

## MSG를 사용한 COE SDO 설정

■기본적으로, 각 slave에 대한 COE SDO 설정은 RUN 모드의 시작에서 I/O 설정에 의해 이루어질 수 있다. 그러나, RUN 모드에서 설정되어야 한다면, COE SDO 설정은 MSG 명령어를 사용하여 할 수 있다.

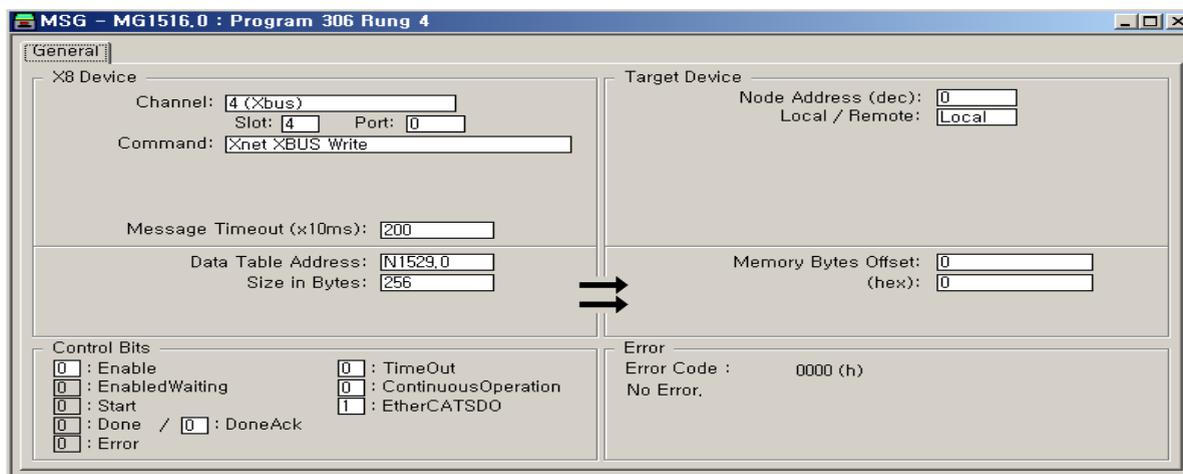
### ■Xnet XBUS Write/Read MSG 사용

- Channel : 4 (Xbus)
- Command : Xnet XBUS Write or Read
- Size in Bytes : 256



### ■Xnet XBUS Write/Read MSG 사용

- Data Table Address : N1529.0 (예외 경우)
- Download 나 Upload 를 위한 COE SDO 는 EtherCATSDO 비트를 set 한 상태에서, Data Table Address 내에서 설정된다



■MG 데이터 테이블에서 비트 .EtherCATSDO

- 이 비트가 clear 되면, 있는 그대로의 데이터가 확장 슬롯의 모듈로 송신된다.
- 이 비트가 set 되면, EtherCAT COE SDO 메시지가 N 데이터 테이블에 설정된 대로 만들어져서 EtherCAT slave 모듈로 전송된다. N 데이터 테이블의 내용은 구조화 된 12 워드 이어야 한다. 다음 장 slide 참고.

Word Offset	Configuration	Number of bytes
0	Reserved (High byte) Command Specifier (Low byte)	1 Byte 1 Byte
1	reserved	2 Bytes
2	reserved	2 Bytes
3	reserved	2 Bytes
4	Index	2 Bytes
5	Reserved (High byte) Sub Index (Low byte)	1 Byte 1 Byte
6	Bit Length	2 Bytes
7	Data Low Word	2 Bytes
8	Data High Word	2 Bytes
9	reserved	2 Bytes
10	reserved	2 Bytes
11	reserved	2 Bytes

■Xnet XBUS Write 명령의 경우, ASCII 인라인 간접주소 변환은 EtherCAT COE SDO 메시지를 만들기 전에 실행된다.

■N 데이터 테이블 주소에서 COE SDO 설정

■N 데이터 테이블 주소에서 COE SDO 설정

- Command Specifier : 1 (1 for Download, 2 for Upload)
- Index : 6060h
- SubIndex : 00h
- BitLength : 08h
- Data : 08h

Address	Type	Value	Meaning	Description
N1528.0	Integer	0001 (h)		Command Specifier/Reserved
N1528.1	Integer	0000 (h)		Reserved
N1528.2	Integer	0000 (h)		Reserved
N1528.3	Integer	0000 (h)		Reserved
N1528.4	Integer	6060 (h)		Index
N1528.5	Integer	0000 (h)		SubIndex
N1528.6	Integer	0008 (h)		BitLength
N1528.7	Integer	0008 (h)		Data(Low Word)
N1528.8	Integer	0000 (h)		Data(High Word)
N1528.9	Integer	0000 (h)		
N1528.10	Integer	0000 (h)		
N1528.11	Integer	0000 (h)		
N1528.12	Integer	0000 (h)		

■ASCII 인라인 간접주소 변환에 의한 N 데이터 테이블 주소에서 COE SDO 설정

- Command Specifier : 1 (1 for Download, 2 for Upload)
- Index : 6060h
- SubIndex : 00h
- BitLength : 08h
- Data : 08h (N1535.6 의 내용)

[N1535.6/b] : decimal 값을 hex 값으로 변환

Address	Type	Value	Meaning	Description
N1528.0	Integer	0001 (h)		Command Specifier/Reserved
N1528.1	Integer	0000 (h)		Reserved
N1528.2	Integer	0000 (h)		Reserved
N1528.3	Integer	0000 (h)		Reserved
N1528.4	Integer	6060 (h)		Index
N1528.5	Integer	0000 (h)		SubIndex
N1528.6	Integer	0008 (h)		BitLength
N1528.7	Integer	[N (a)		Data(Low Word)
N1528.8	Integer	15 (a)		Data(High Word)
N1528.9	Integer	35 (a)		
N1528.10	Integer	,6 (a)		
N1528.11	Integer	/b (a)		
N1528.12	Integer	] (a)		

▣ Ladder Program flow to write COE SDO parameters

- Xnet XBUS Write MSG instruction with Command Specifier 1 (for Download Request).
- Xnet XBUS Read MSG instruction. Download Response, 3, will be returned in Command Specifier.

▣ Ladder Program flow to read COE SDO parameters

- Xnet XBUS Write MSG instruction with Command Specifier 2 (for Upload Request).
- Xnet XBUS Read MSG instruction. Upload Response, 2, will be returned in Command Specifier.

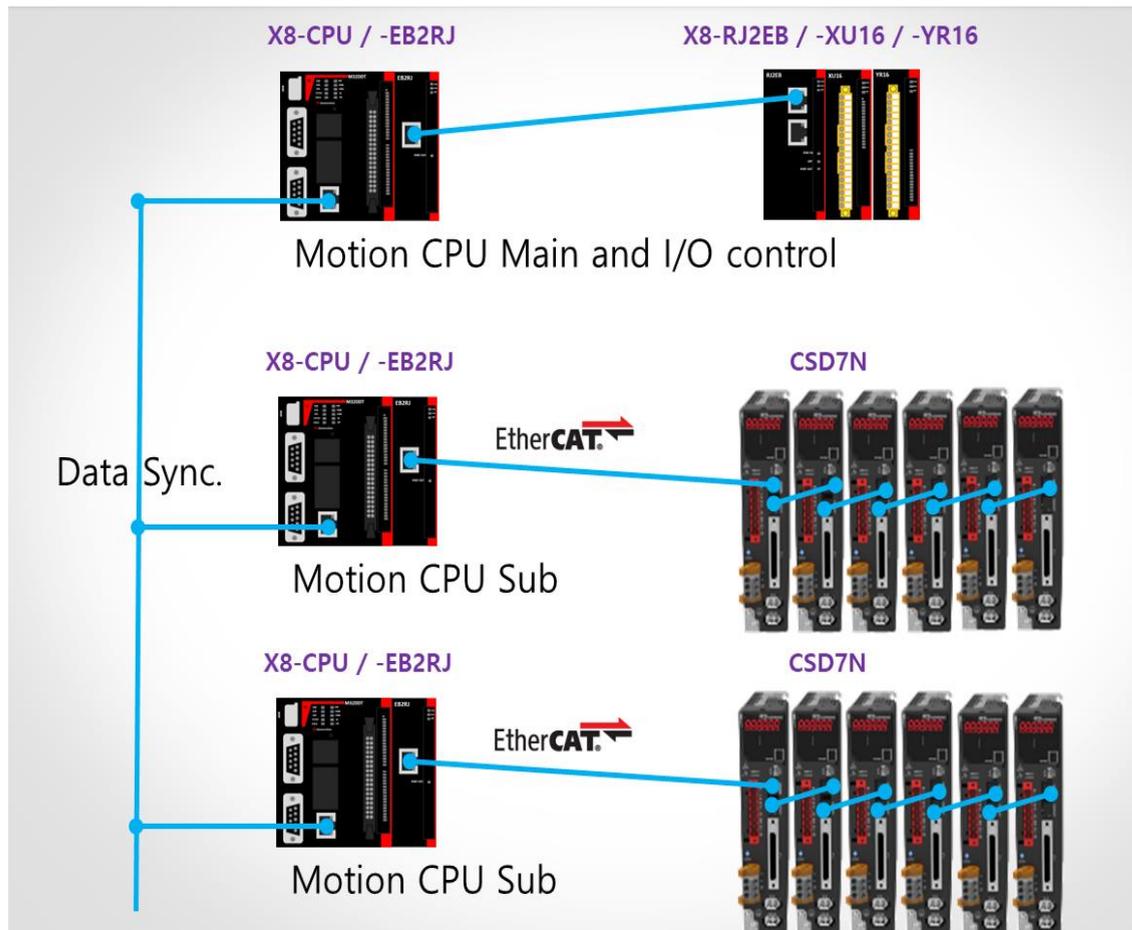
▣ 확장 I/O 슬롯 번호 지정

- 비트, SR2.90.1, “[C] MSG/통신시 I/O 설정에 의한 Slot 번호 지정”.
- I/O 설정에서 가상 모듈이 있을 경우, 슬롯 번호가 CPU 모듈로부터의 하드웨어 오프셋과 I/O 설정에서의 슬롯 번호가 다를 수 있다. 이 경우, 이 비트는 확장 I/O 슬롯 번호 지정을 어떤 방법으로 할 것인지를 정의할 수 있다.

비트, SR2.90.1 이 0 이면, 슬롯 번호 지정은 하드웨어 설정에 따른다. I/O 설정의 가상 모듈 설정에 상관없이, 슬롯 번호는 CPU 모듈로부터의 하드웨어 오프셋에 의해 지정된다.

비트, SR2.90.1 이 1 이면, 슬롯 번호 지정은 I/O 설정에 따른다.

## CPU 분산 처리를 이용한 6 축 이상의 시스템



## 15장. 예제처리

X8 PLC로 응용예제처리에 대해 기술합니다.

---

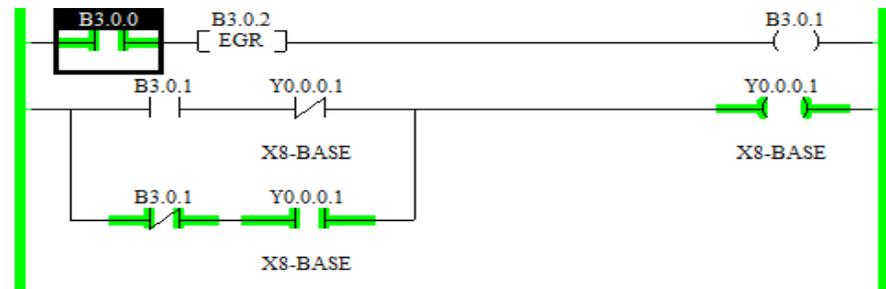
예제처리 .....	15-2
참고사항 .....	15-14

## 예제처리

PLC의 시퀀스회로는 비슷한 구조로 형성되어 있습니다. 기본 구조를 이해하시고 프로그램을 작성하시면 빠른 시퀀스 구성이 가능하오니 미리 익혀 프로그램을 시작하시는 것이 바람직합니다.

응용예제1)

### 교번회로



설명

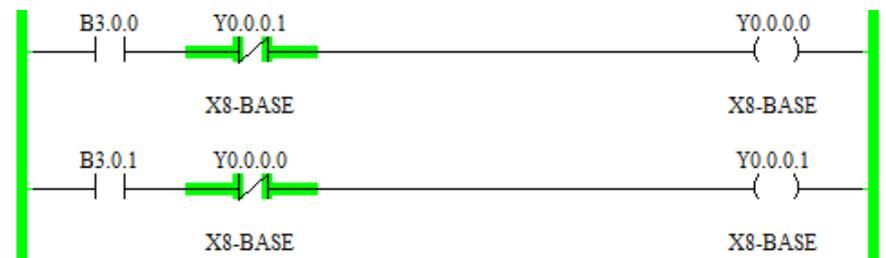
교번동작 회로는 입력접점이 한번씩 ON,OFF될때마다 출력이 교번으로 한번은 ON, 한번은OFF를 반복하는 회로를 말한다. 즉, 입력이 ON된 후 OFF가 되어도 출력은 계속 ON을 유지하고 있으며, 다시 입력이 ON, OFF되면 출력이 OFF상태로 유지된다.

주1) 입력접점의 래더표시 [EGR] 은 '상승엣지'를 뜻하며, 입력이 OFF에서 ON으로 변하는 순간의 1Scan동안만 작동하는 접점입니다.

상세내용은 EGR 명령어 부문을 참조바랍니다.

응용예제2)

### 인터록 회로

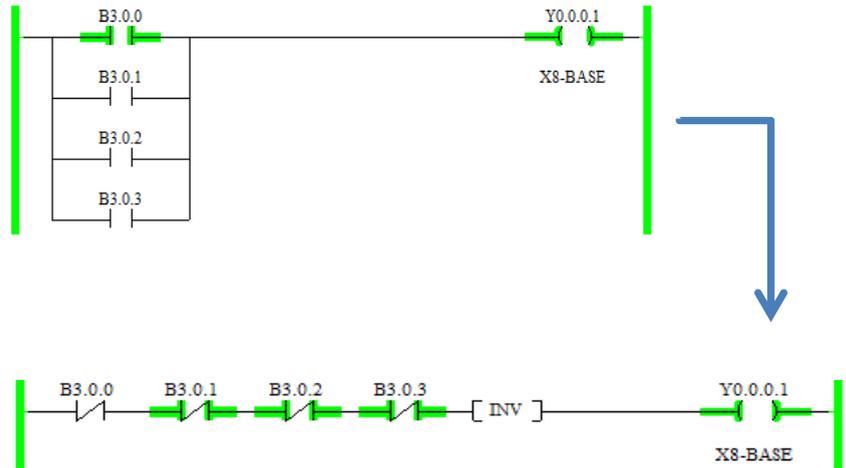


설명

인터록 회로는 출력X 및 출력Y중에서 하나가 ON작동을 하고 있으면 다른쪽 출력이 ON되지 못하도록 제한하는 회로를 말하며,  $X=A*Y$ ,  $Y=B*X$  로 표현한다

응용예제 3)

INV(NOT) 명령 응용 회로



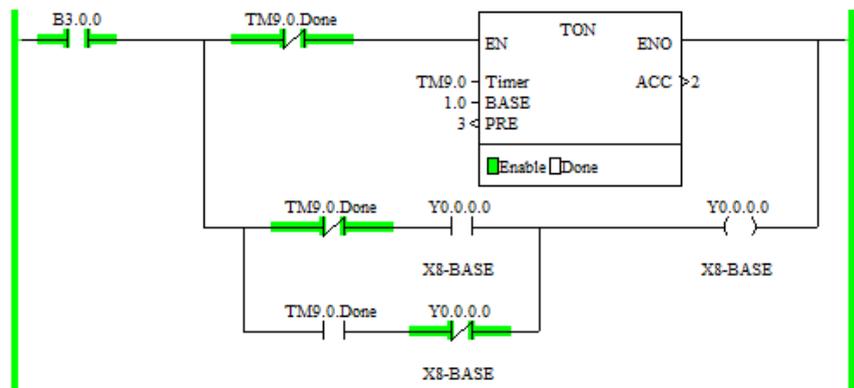
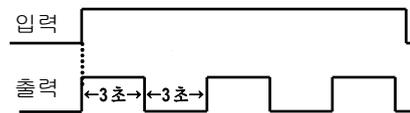
설명

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

응용예제 4)

타이머 응용 회로(타이머 1개를 이용한 플러커 회로)

<타임차트>



설명

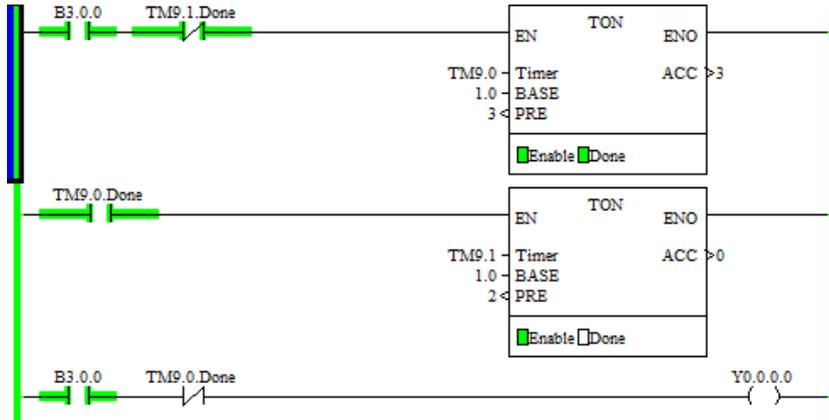
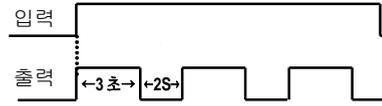
입력이 ON되면 3초간 ON, 3초간 OFF가 반복되는 회로(플러커 회로)  
타이머의 SV값을 조정하면 플러커 간격을 조정할 수 있습니다.

이 회로는 타이머와 교번동작 회로를 이용한 방법입니다.

응용예제5)

타이머 응용 회로(타이머 2개를 이용한 ON/OFF플러커 회로)

<타임차트>



설명

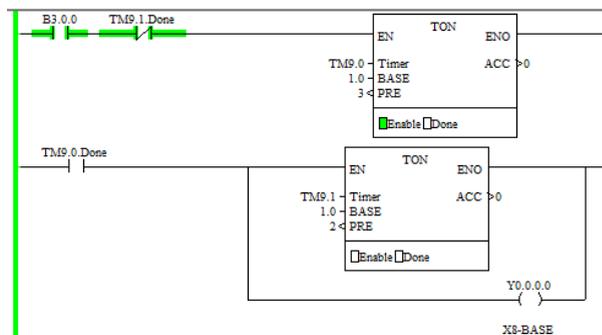
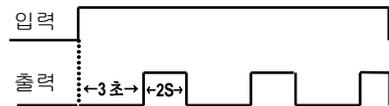
입력이 ON되면 3초간 ON, 2초간 OFF가 반복되는 플러커 타이머의 SV값을 조정하면 플러커 간격을 조정할 수 있습니다. 이 회로는 타이머와 교번동작 회로를 이용한 방법입니다.

응용예제6)

타이머 응용 회로(타이머 2개를 이용한 OFF/ON플러커 회로)

<타임차트>

응용 표현

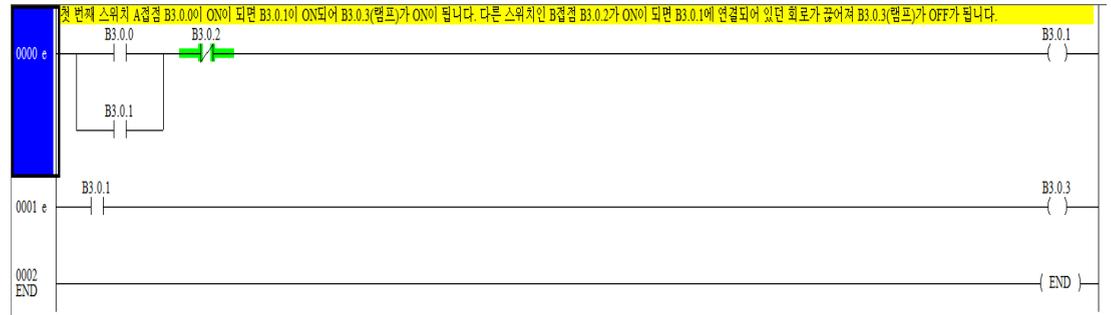


설명

입력이 ON되면 3초간 OFF, 2초간 ON가 반복되는 플러커회로 타이머의 SV값을 조정하면 플러커 간격을 조정할 수 있습니다. 이 회로는 타이머와 교번동작 회로를 이용한 방법입니다.

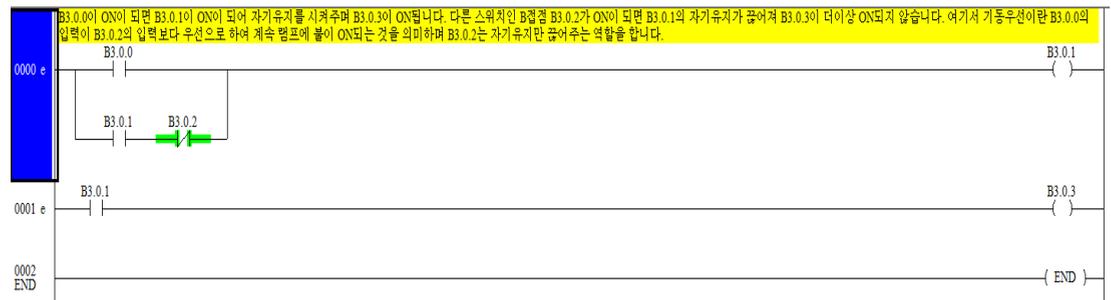
**실습 예제01) 정지우선 자기유지 회로**

정지우선 자기유지 회로는 기동신호와 정지신호가 동시에 입력 될 경우 출력을 내보내지 않는 회로를 말한다. 기동신호를 B3.0.0, 정지신호를 B3.0.2, 출력을 B3.0.3이라고 하면 다음과 같다.



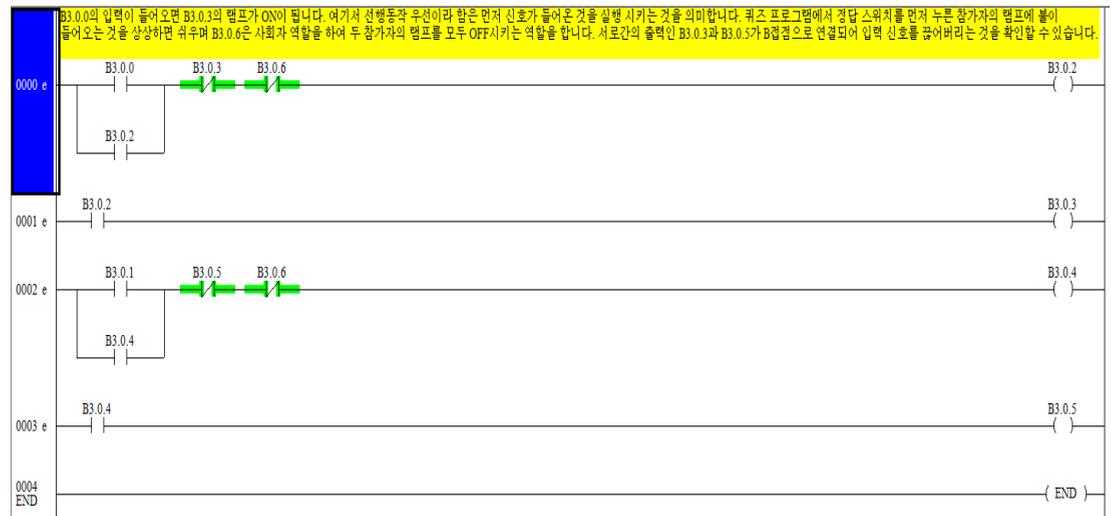
**실습 예제02) 기동우선 자기유지 회로**

기동우선 자기유지회로는 기동신호와 정지신호가 동시에 입력될 경우 기동신호가 우선하여 출력신호를 내보내는 회로를 말한다.



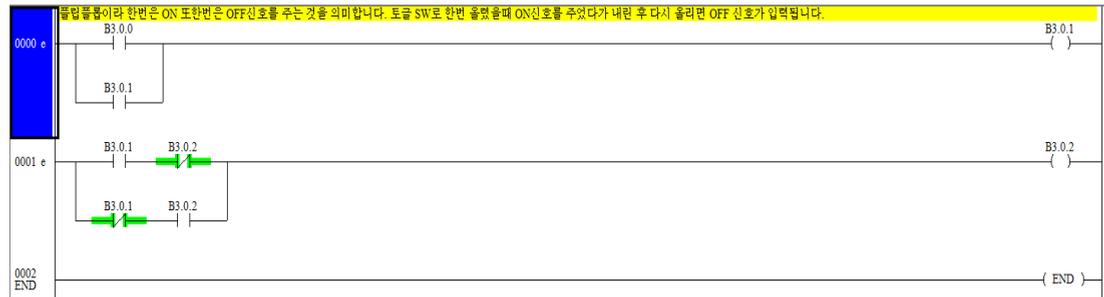
**실습 예제03) 선행동작 우선회로 (인터록 회로)**

선행동작 우선회로는 입력에 들어가는 여러 신호 중 가장 먼저 들어온 신호를 우선으로 동작시키고 뒤에 들어온 신호에 의한 동작은 금지시키는 회로를 말한다.



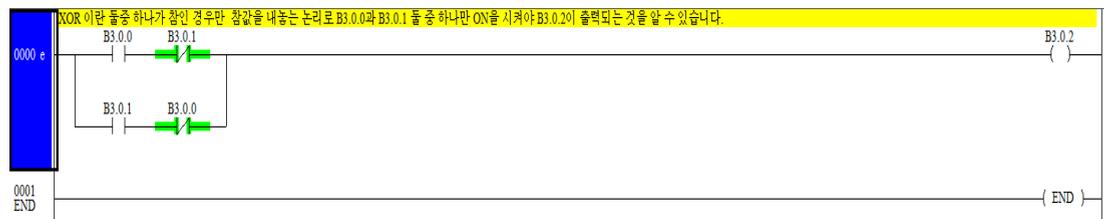
**실습 예제04) 플립플롭회로 (FLIP FLOP)**

플립플롭 회로는 쌍안정 멀티바이브레이터(Bistable N4.ultivibB3.atoB3.) 또는 래치(Latch)회로 등으로 부르는 회로로 설정과 복귀라는 두 입력을 가지고 입력의 상태에 따라 출력이 정해지며 출력상태가 결정되면 입력이 없어도 출력이 그대로 유지되는 특성을 가진 회로이다.



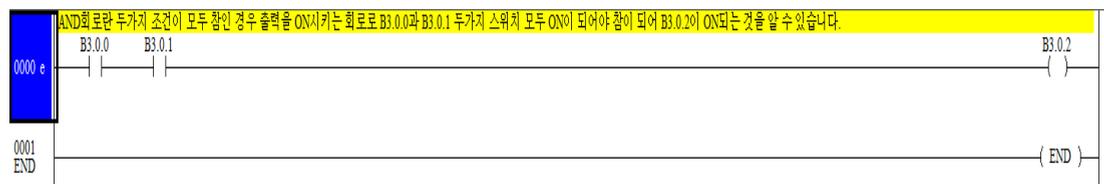
**실습 예제05) 반일치 회로(Exclusive OR회로)**

반일치 회로는 입력이 두 개인 경우 두 입력의 상태가 모두 같을 때는 출력이 0이고, 다른 경우에는 출력이 1이 되는 회로를 말한다.(XOR)



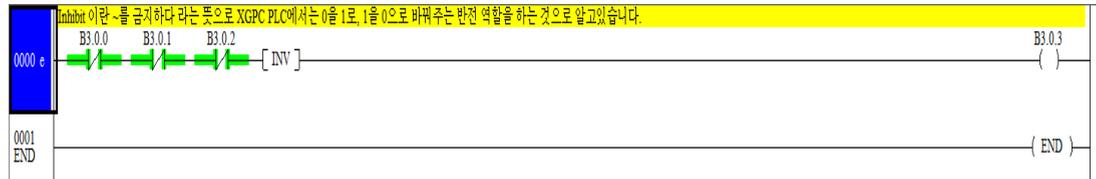
**실습 예제06) 일치 회로 (Coincidence 회로)**

일치회로란 두 입력의 상태가 같을 때만 출력이 ON되고 다른 경우에는 OFF되는 회로를 말한다. 즉 반 일치회로와는 반대의 동작이다.



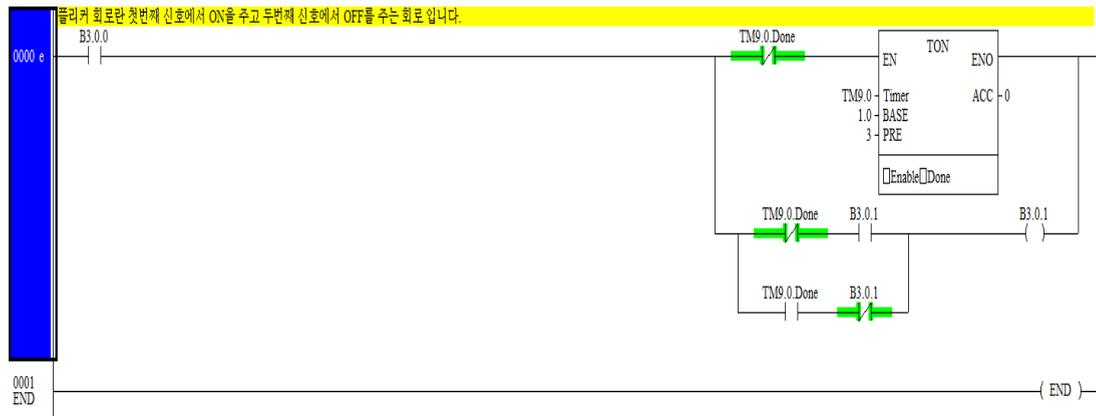
**실습 예제07) 금지 회로(Inhibit 회로)**

금지 회로란 AND로 구성된 다수 입력의 a접점 외에 b접점 한 개를 포함한 회로를 말한다. 즉 여러 개의 AND 회로 중 한 개의 AND는 금지 입력으로 NOT 회로를 조합시킨 회로이다.



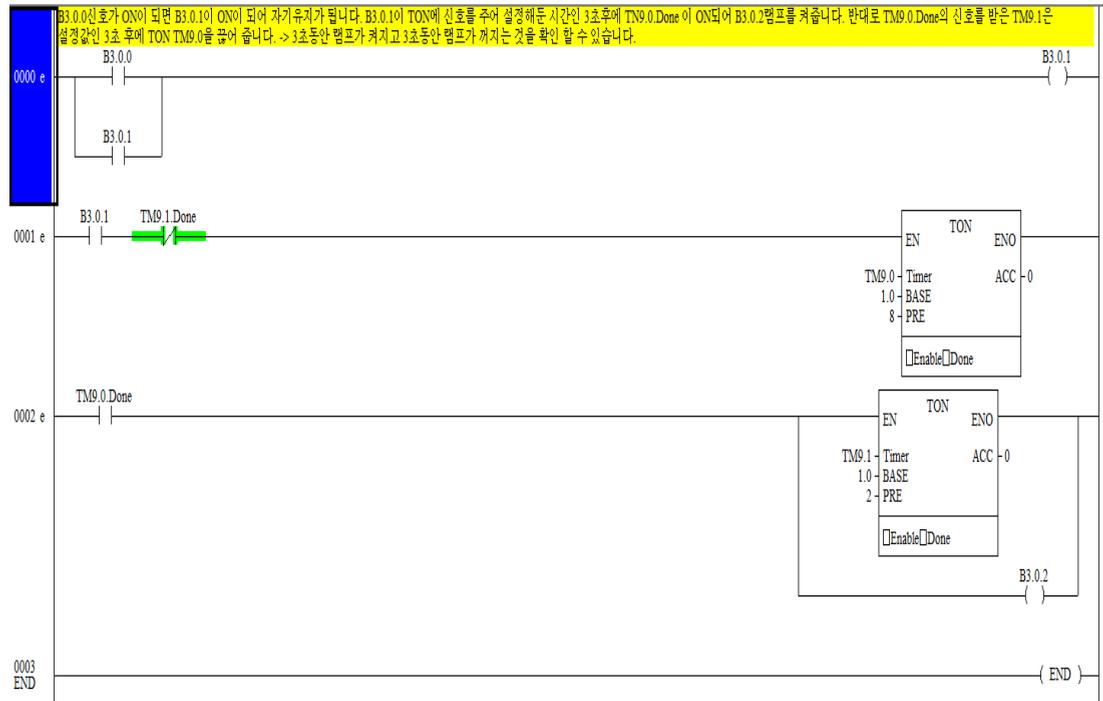
**실습 예제08) 타이머 1 개를 이용한 플리커 회로**

입력이 ON 되면 3 초간 ON, 3 초간 OFF 가 반복되는 회로



**실습 예제09) 타이머 2 개를 이용한 플리커 회로**

입력이 B3.0.0이 ON 됨과 동시에 램프가 3초 간격으로 OFF→ON→OFF→ON을 반복한다. 타이머0은 타이머1을, 타이머1은 타이머0을 각각 리셋시키도록 설계된 프로그램이다.



**실습 예제10) 불안정한 신호의 안정화 회로**

불안정한 입력신호가 출력신호로 바로 연결되는 경우 출력 신호가 입력 신호를 바로 따라 동작하므로 인하여 불안정해지는 경우가 많다.

이러한 경우 입력 신호가 ON된 경우부터 출력을 설정된 일정 시간 동안 유지시켜 출력을 안정화시켜야 한다.

여기에서 우리는 불안정한 입력 신호를 안정한 출력신호로 안정화시키는 프로그램을 작성해 보자.

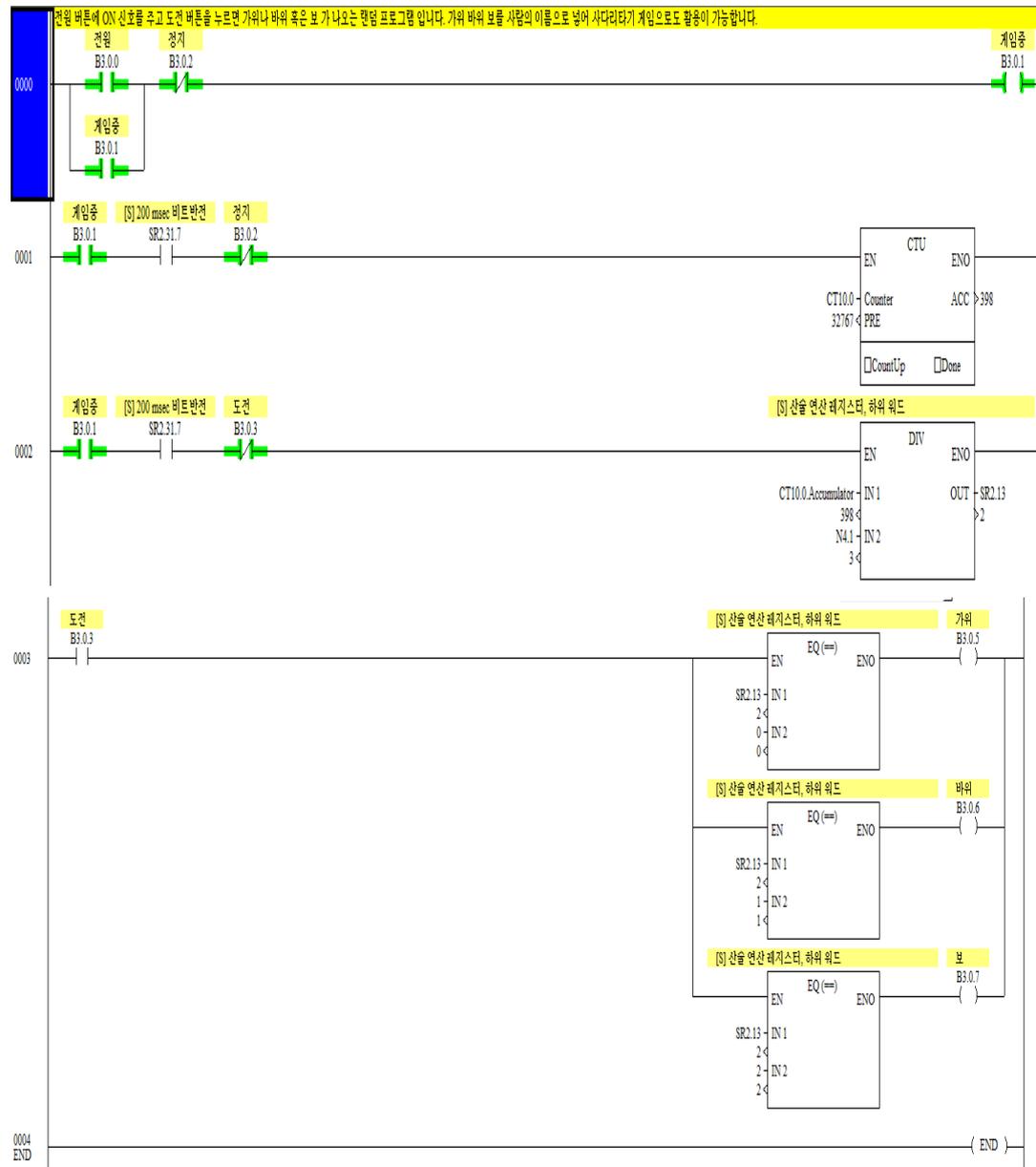
컨베이어 벨트 위로 움직이는 음료수 병을 감지하는 센서는 일반적으로 광센서를 사용하는데 병이 최초에 감지되는 경우에는 반사되는 광신호가 병에 의해 산란되어 수신 신호가 불안정하게 된다.

그러나 병이 감지되면 일정시간 내에 음료수를 병에 주입하는 시스템에 이 프로그램을 적용할 수 있다.



### 실습 예제11) 랜덤 가위바위보 프로그램

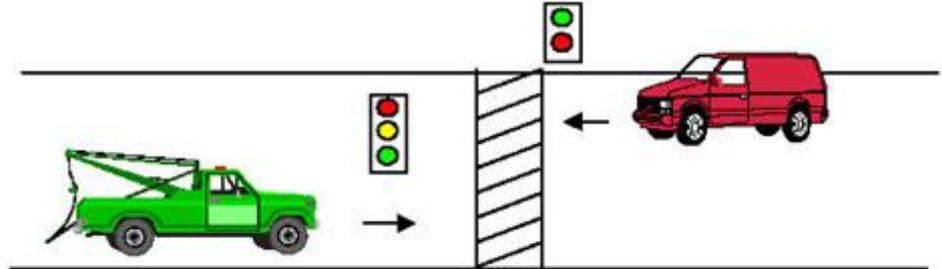
전원 버튼에 ON 신호를 주고 도전 버튼을 누르면 가위나 바위 혹은 보 가 나오는 랜덤 프로그램이다. (CTU : PRE 32767 / Counter CT 10.0 / DIV : CT10.0 Accumulator)



\*EQ IN1 값 : SR2.13 // IN2 값 : 0(가위), 1(바위), 2(보)

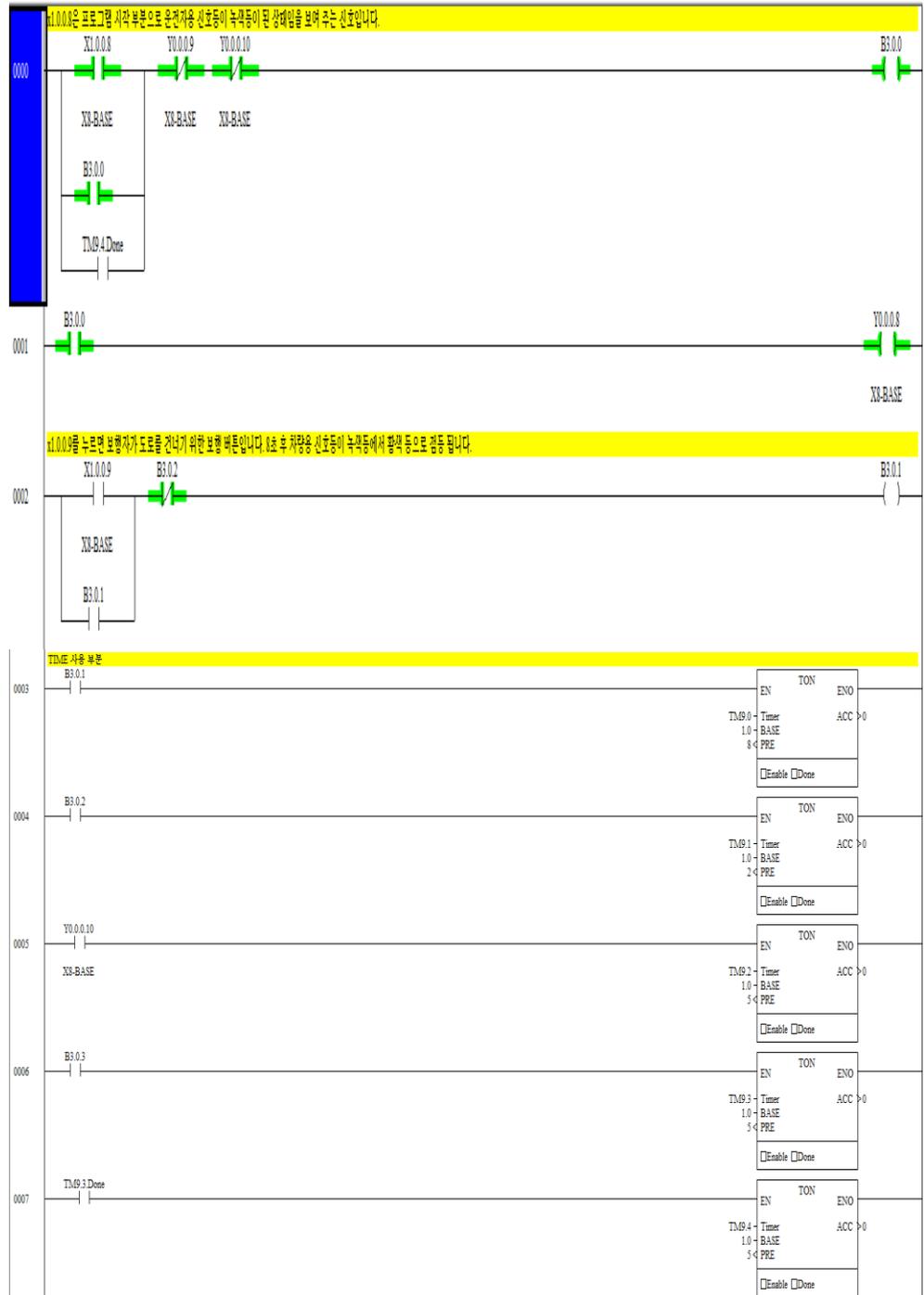
**실습 예제12) 신호등 제어**

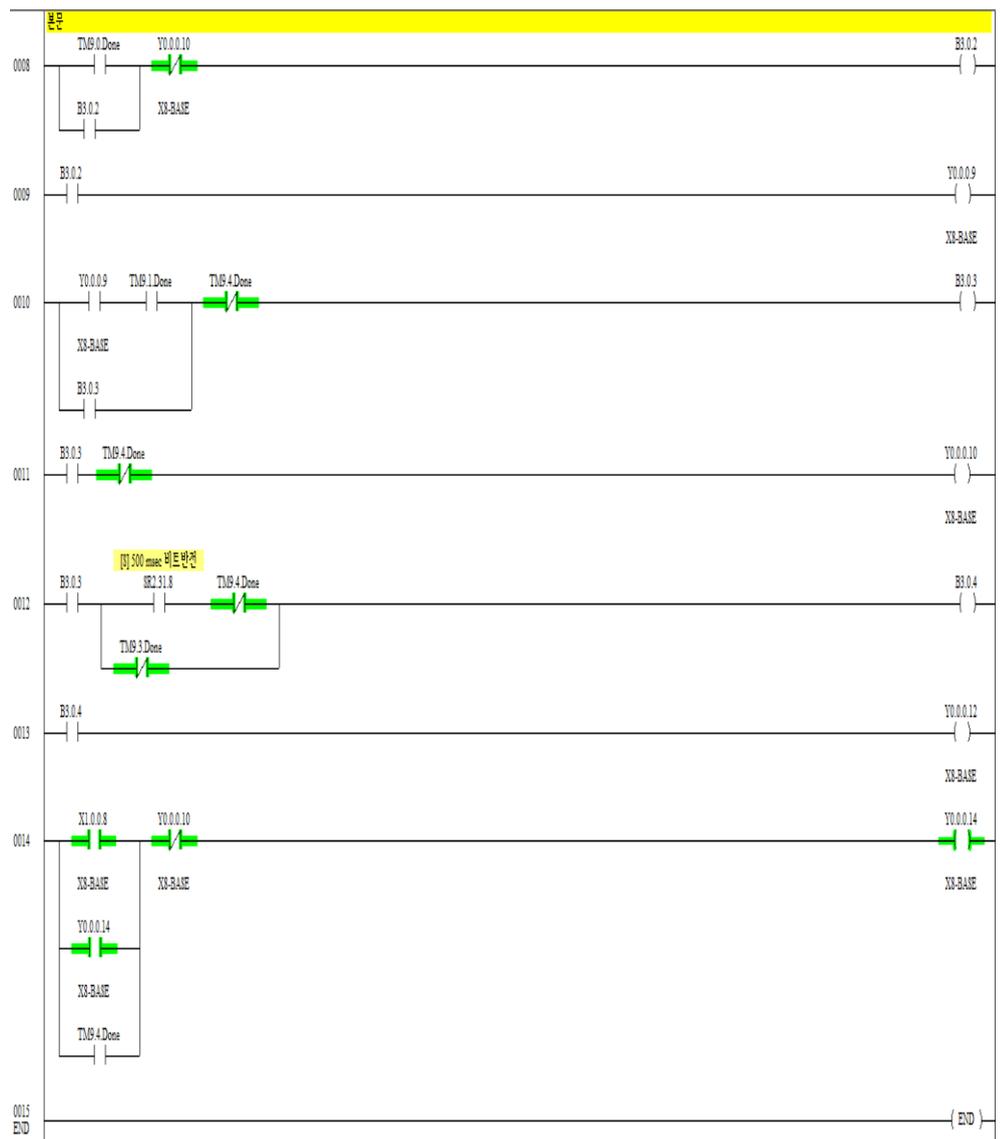
차도를 횡단하기 위한 보행자용 횡단보도가 있으며 보행자가 신호등에 부착된 버튼을 누르는 경우 이후 진행 시간에 따라 신호등은 제어된다.

**동작 순서**

- ① 보행자가 도로를 건너기 위하여 보행 버튼을 누르면 8 초 후 차량 운전자용 신호등이 녹색등에서 황색등으로 점등된다.
- ② 이후 2 초가 경과되면 차량 운전용 신호등은 적색 램프로 바뀌며 보행자용 신호등은 적색에서 녹색으로 점등된다.
- ③ 이후 5 초가 경과되면 5 초간 보행자용 신호등의 녹색 램프가 0.5초 주기로 점멸하며 점멸주기가 경과하면 적색 램프가 점등되며 동시에 차량 운전자용 신호등에는 적색등에서 황색등이 점등된다.
- ④ 이후 2 초가 경과되면 차량 운전자용 신호등에는 황색등에서 녹색등이 점등된다.

\*(내부접점이 아닌 입출력 활용)





## 참고사항

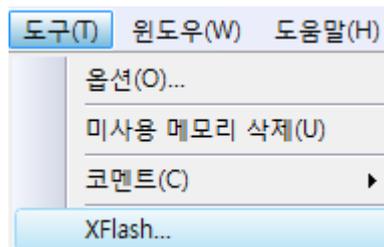
### 펌웨어 업그레이드 방법

1. X8 PLC Firmware 업그레이드는 Ethernet Port를 통해서만 가능합니다. 따라서 X8 시리즈PLC(이하 PLC)가 PC와 동일한 게이트웨이의 Ethernet Network로 연결되어 있어야 합니다. PC와 직접 연결하거나, 허브 또는 Ethernet 스위치로 연결합니다.

2. PLC에 고정 IP Address가 설정되어 있지 않고, BOOTP protocol 이 Enable(Default)되어 있는 경우, \*참고 사항 1을 보시고 IP Address를 설정합니다. 이 경우에 X8 PLC 는 PC 에서 동작하는 BOOTP Server 로부터 IP Address를 받아오게 됩니다.

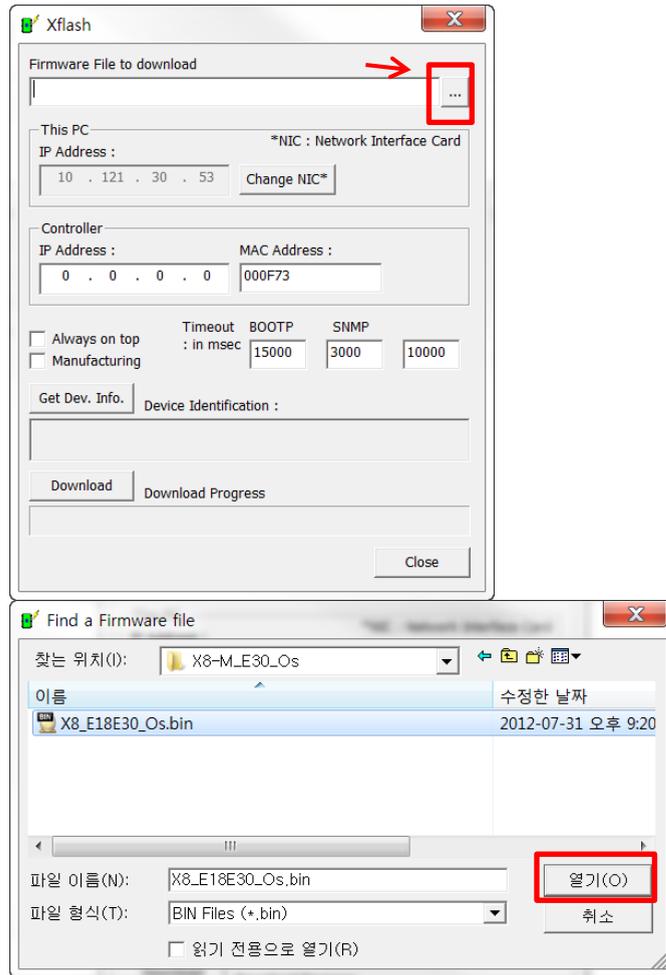
3. PLC에 고정 IP Address가 설정되어 있는 경우, Xflash.exe를 실행합니다.

도구 메뉴 참조

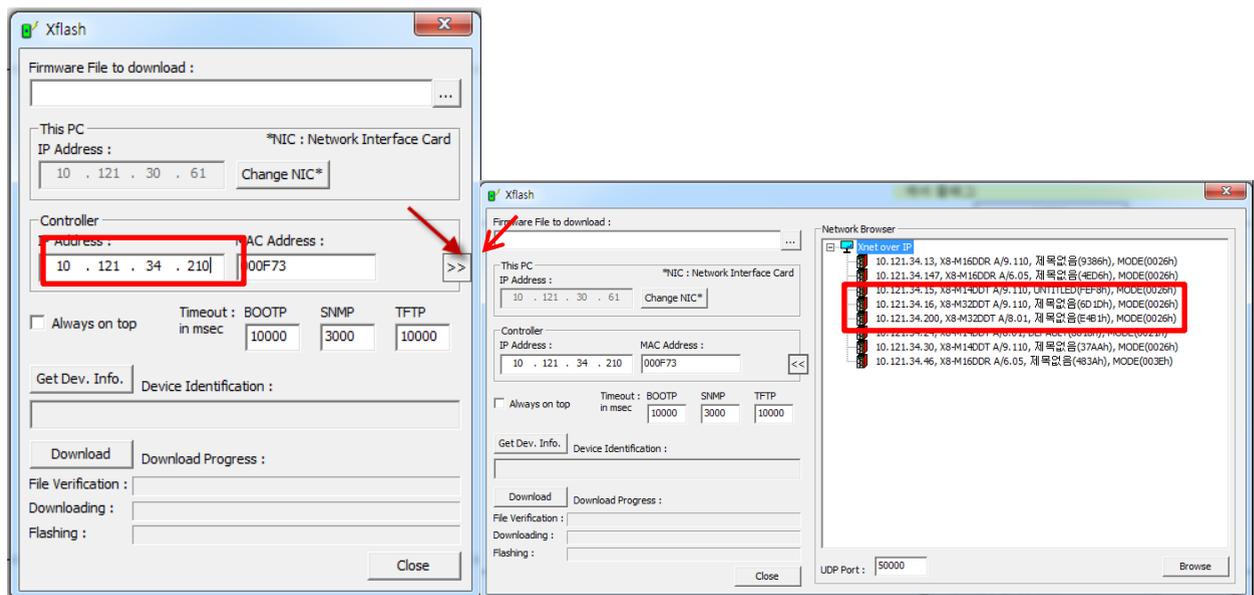


4. Windows 방화벽에 의해 Xflash 프로그램이 차단될 경우, \*참고 사항 2(Windows XP)또는 \*참고 사항 3(Windows 7)을 보시고 방화벽을 설정합니다.

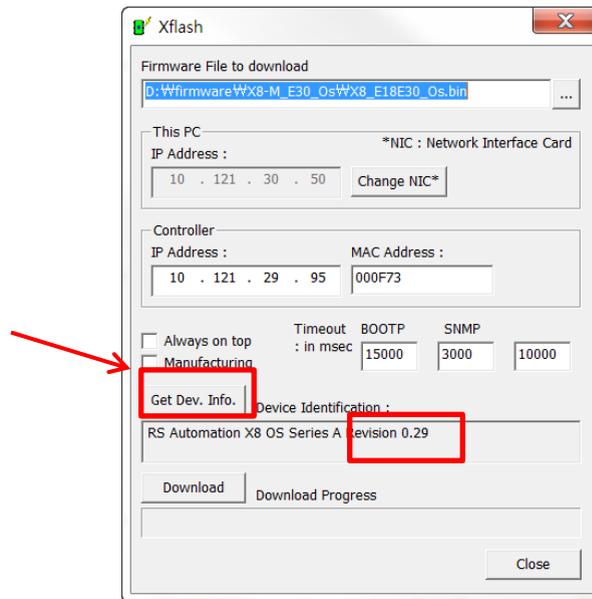
5. Xflash.exe를 실행한 후, 번호 순서대로 실행합니다. 화살표가 가리킨 버튼을 눌러 \*\_OS.bin 파일을 불러옵니다.



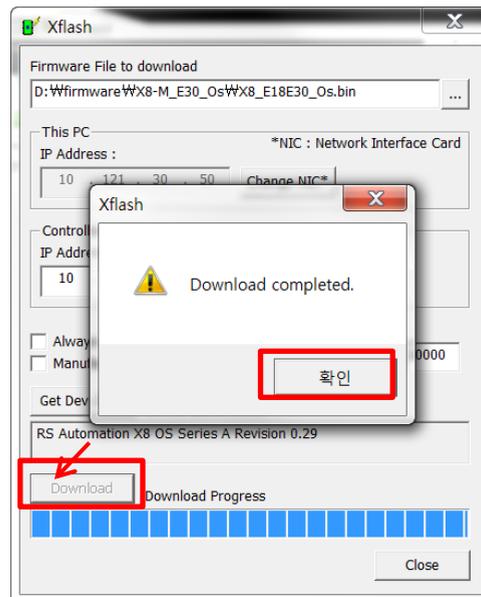
6. 화살표가 가리킨 곳에 PLC의 IP Address를 직접기입 하거나 >>화살표를 눌러 선택을 합니다.



7. 화살표가 가리킨 버튼을 눌러 현재 PLC의 OS 버전을 확인합니다.



8. 꼭 먼저 X8 PLC가 프로그램 모드 상태인지 확인 하시고 화살표가 가리킨 버튼을 눌러 OS를 다운로드 합니다. OS를 다운로드 하는 중에는 PLC의 왼쪽 한 열의 LED가 순서대로 점멸을 반복하게 됩니다. 다운로드가 끝나면 왼쪽 한 열의 LED가 모두 켜진 후, PLC가 자동으로 재부팅을 하게 됩니다. 재부팅 후 PWR LED만 켜지면 펌웨어 업데이트가 완료된 것입니다. 다운로드가 완료되면 확인 창이 아래와 같이 나타납니다.

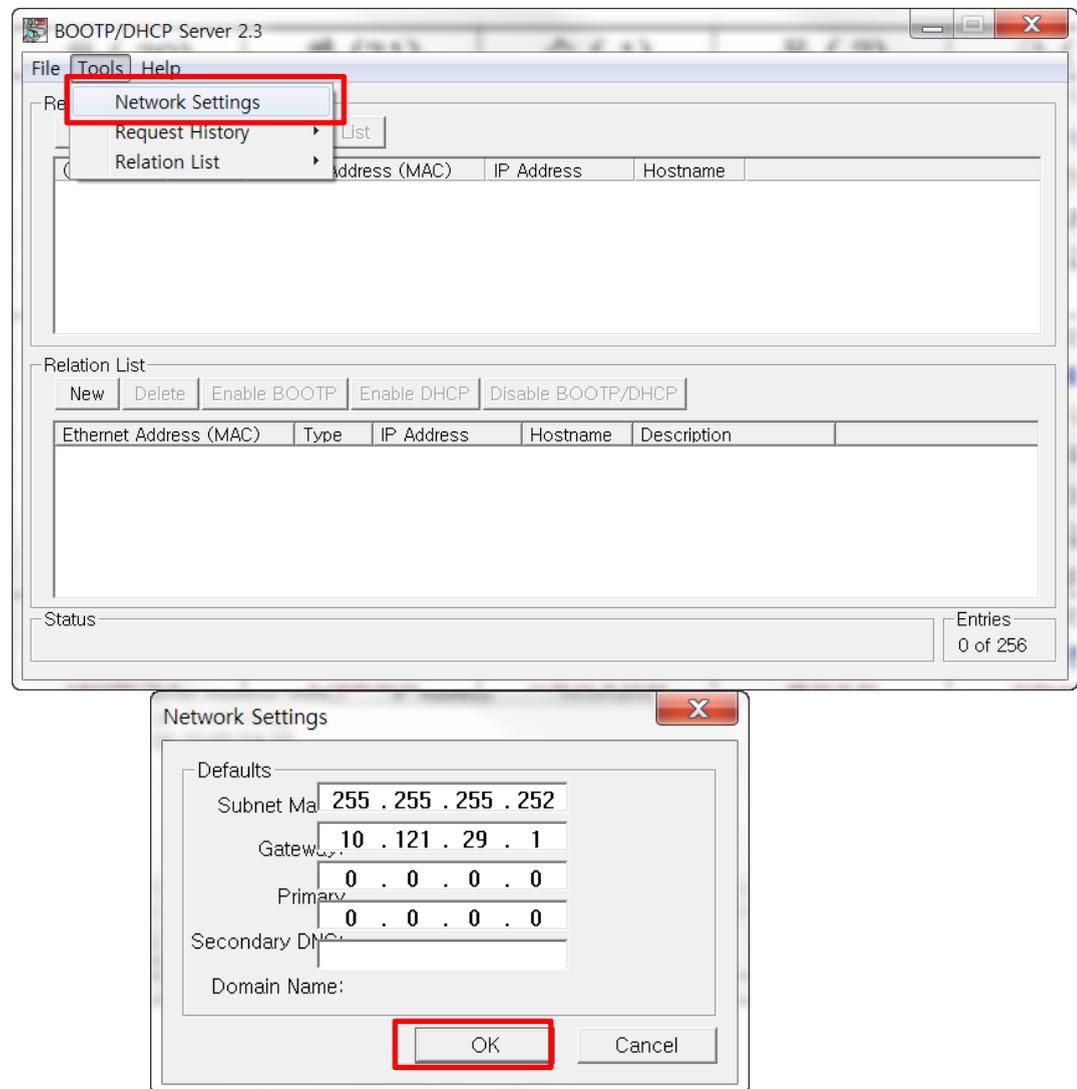


9. 5번부터 8번 과정에서 펌웨어 업그레이드가 정상적으로 되지 않은 경우, 2번의 과정부터 재시도를 할 수 있습니다.

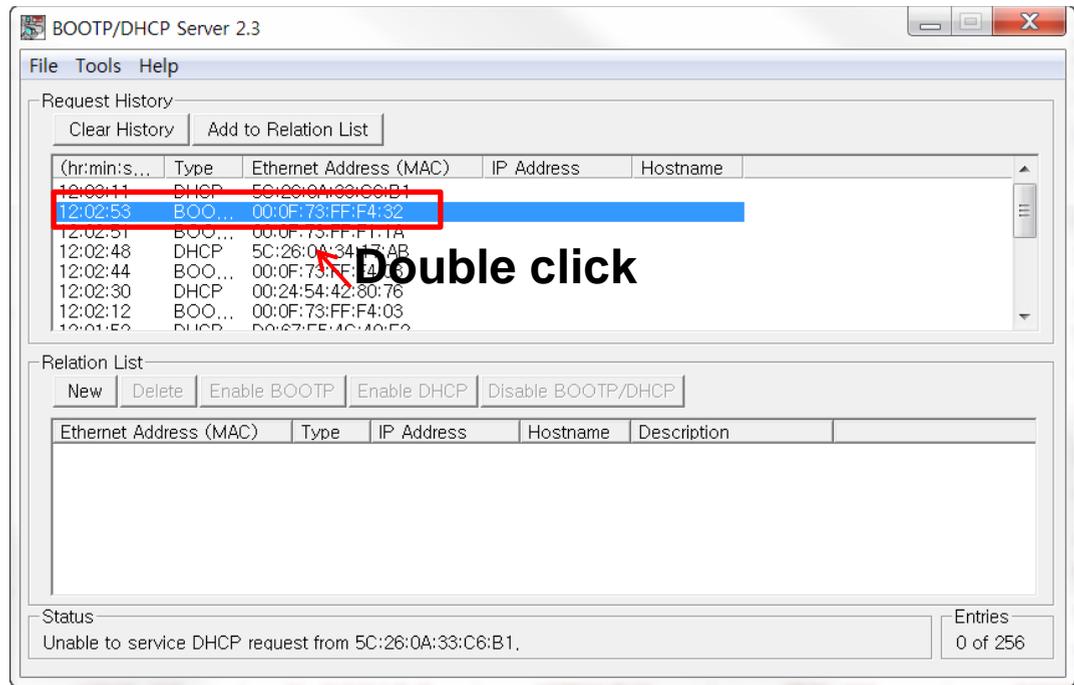
**\*참고사항 1. 고정 IP Address가 설정 되어 있지 않은 경우**

1. RSA에서 제공하는 BOOT/DHCP Server 프로그램을 실행합니다.  
(현재 프로그램이 나오지 않은 관계로 AB에서 제공하는 프로그램을 설치해 주시기 바랍니다. 프로그램은 <http://www.ab.com/networks/ethernet/bootp.html>에서 다운받아 실행합니다.)

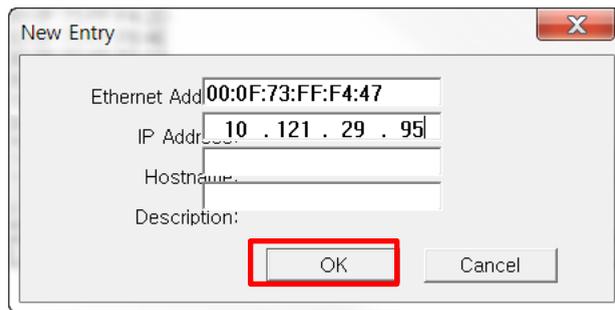
2. 처음 프로그램을 실행하면 Network Settings 창이 나옵니다. 혹은 Tools 메뉴에서 Network Settings를 실행합니다. 컴퓨터의 Subnet Mask와 Gateway Address를 입력하고 OK 버튼을 누릅니다. Subnet Mask와 Gateway Address 를 모르시는 경우에는 네트워크 관리자에게 문의하십시오.



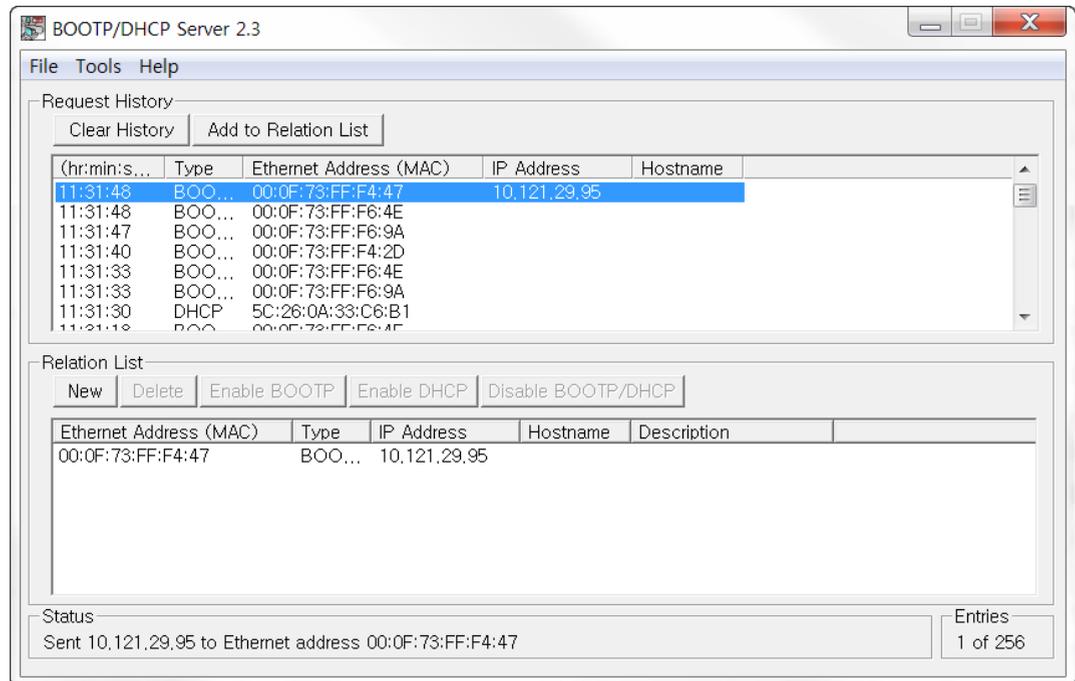
3. Network Setting이 완료되면 PC와 동일한 네트워크 상에 있는 제품들의 MAC ID가 Request History 창에 나타납니다. 펌웨어 업그레이드를 할 PLC의 MAC ID를 찾아서 더블 클릭합니다



4. New Entry 창이 나오면 PLC에 설정할 IP Address를 입력한 후, OK 버튼을 누릅니다. 그러면, Relation List 에 항목이 추가됩니다.



5. 잠시 후, Request History 창에 IP Address를 설정한 PLC의 MAC ID와 IP Address가 아래의 그림과 같이 나오면 PLC에 IP Address가 설정된 것입니다.



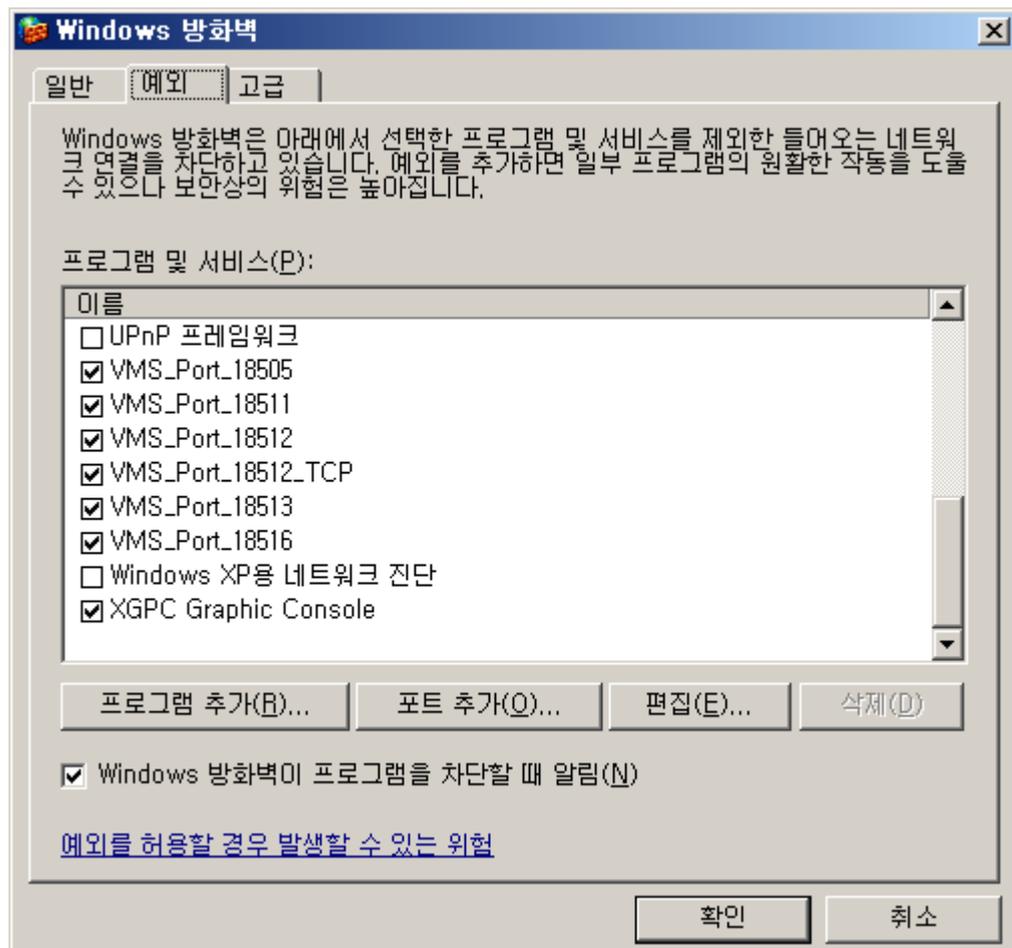
### \*참고사항 2. Network 방화벽 설정(Windows XP)

1. 다음의 과정들은, PC의 OS가 Windows XP 일 경우, Xflash.exe program이 Windows 방화벽에 막혀 통신을 하지 못하는 경우의 해결 방법입니다.

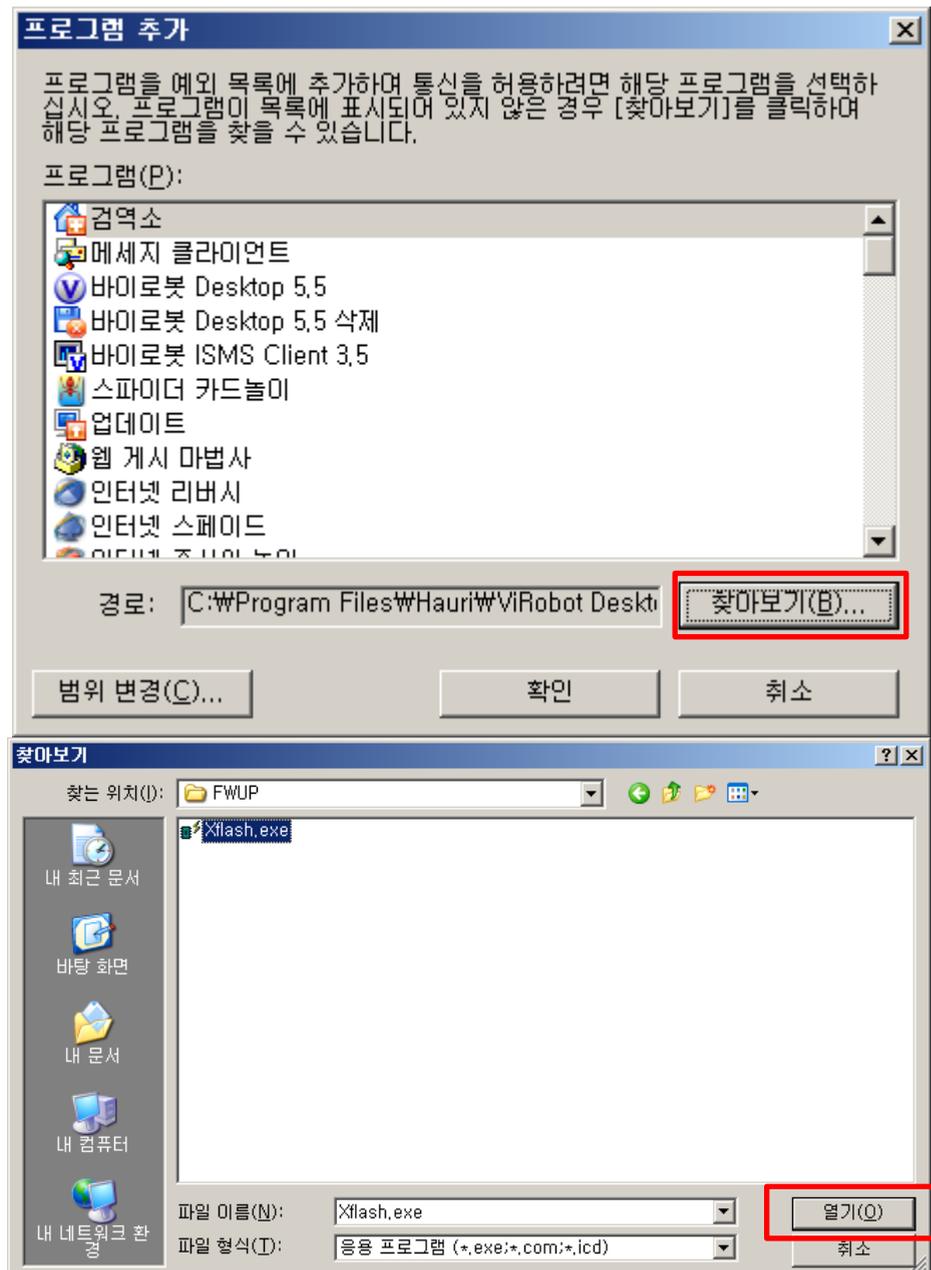
2. 아래 메뉴를 따라 Windows 방화벽을 실행합니다.

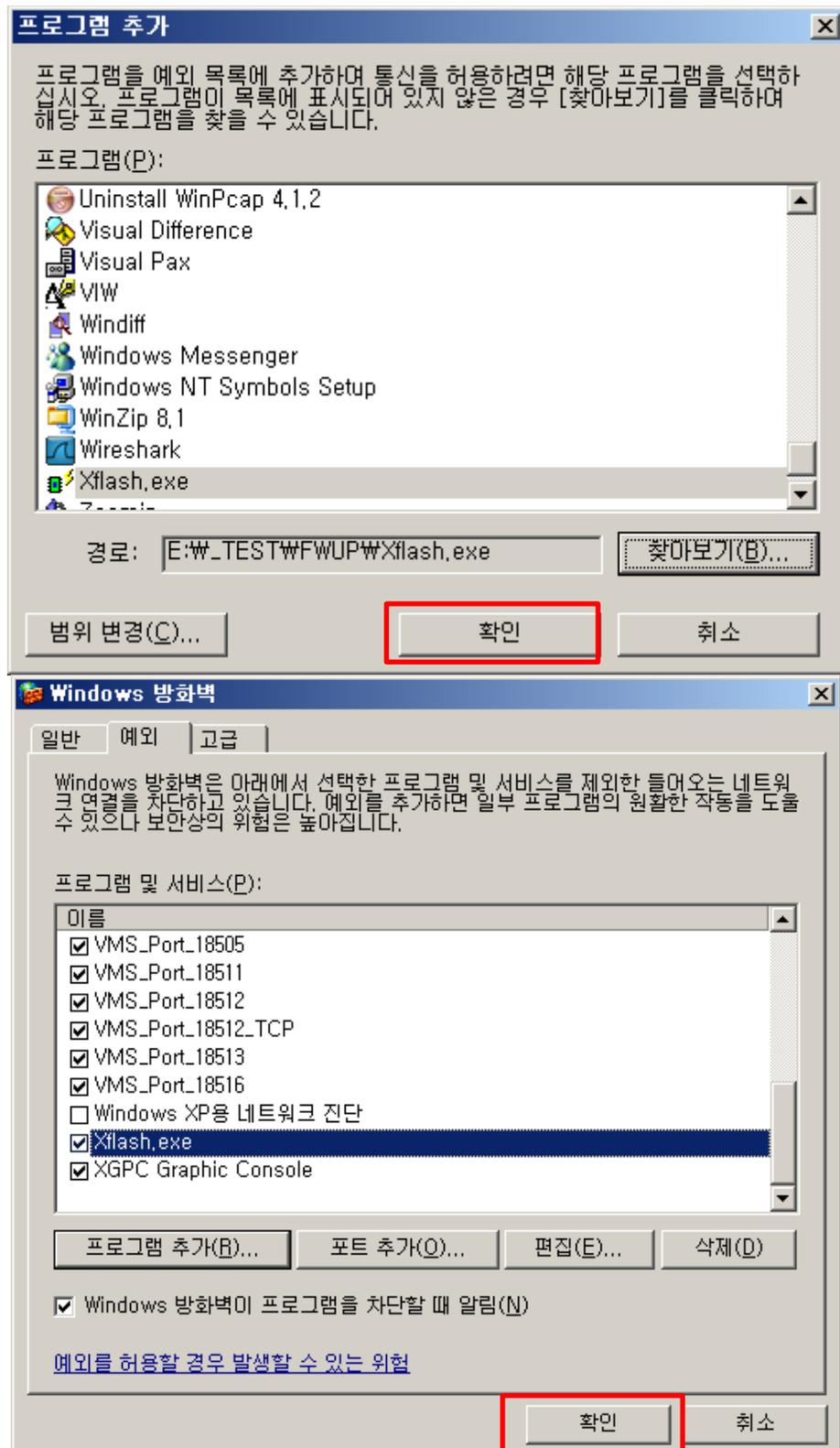
시작 > 설정 > 제어판 내 Windows 방화벽

3. 예외 탭으로 이동합니다. Xflash 프로그램이 예외 처리 리스트에 등록되었는지 확인합니다.



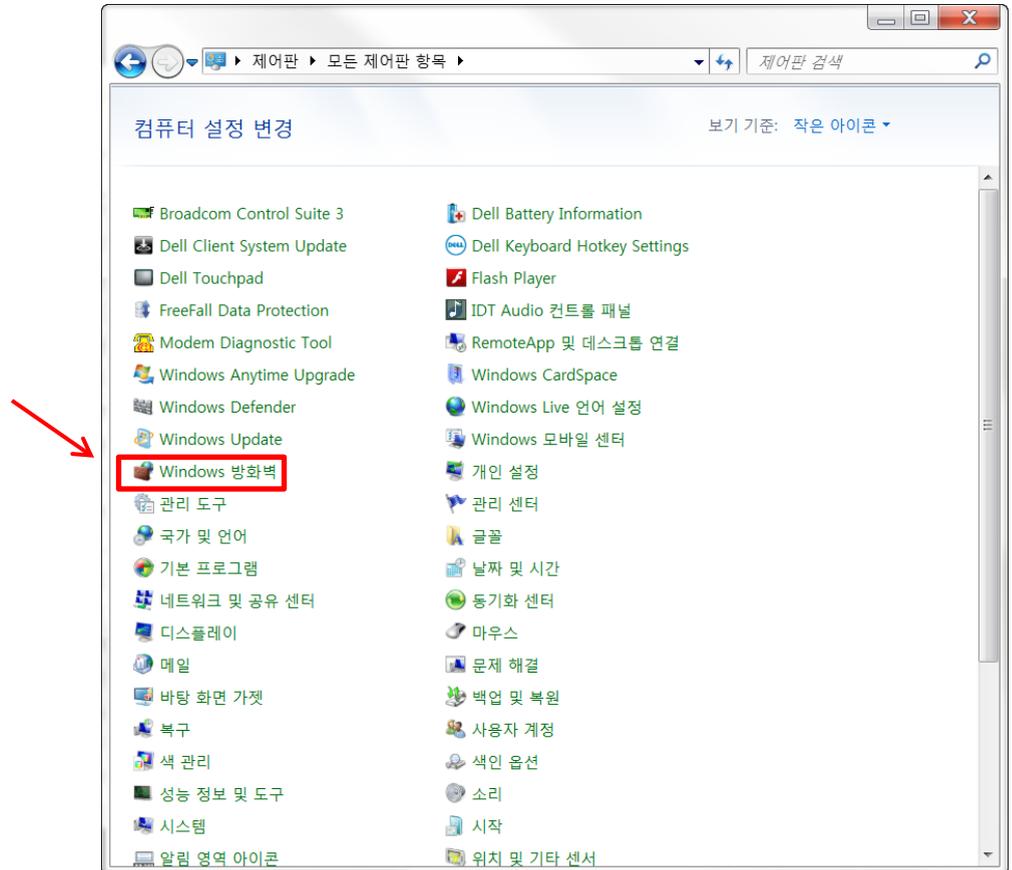
4. 리스트에 있을 경우, 다음 과정을 생략합니다. 리스트에 없을 경우, Xflash 프로그램을 프로그램 추가 버튼을 눌러 추가합니다.





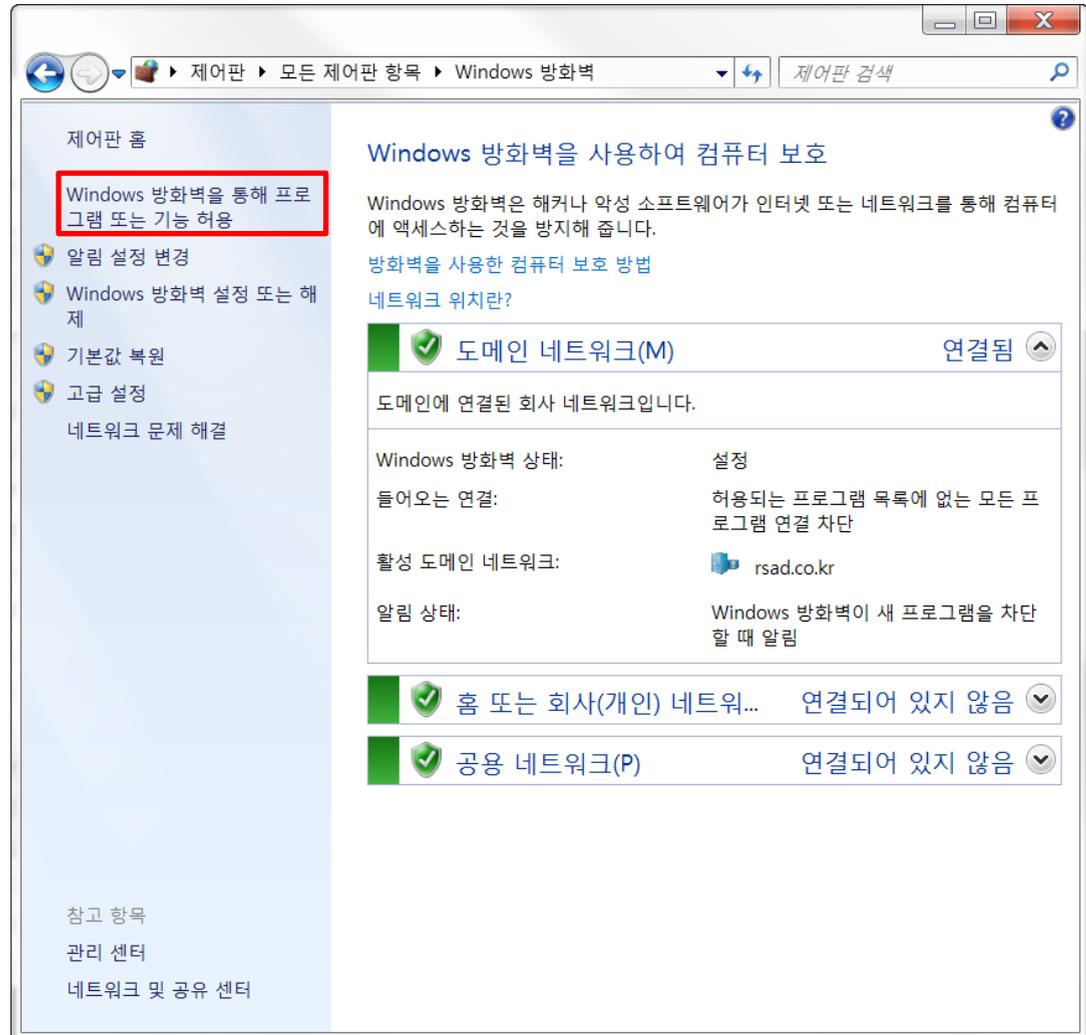
**\*참고사항 3. Network 방화벽 설정(Windows 7)**

1. 방화벽 설정을 위해 시작-제어판을 실행합니다. 제어판에서 Windows 방화벽을 클릭하여 실행합니다.

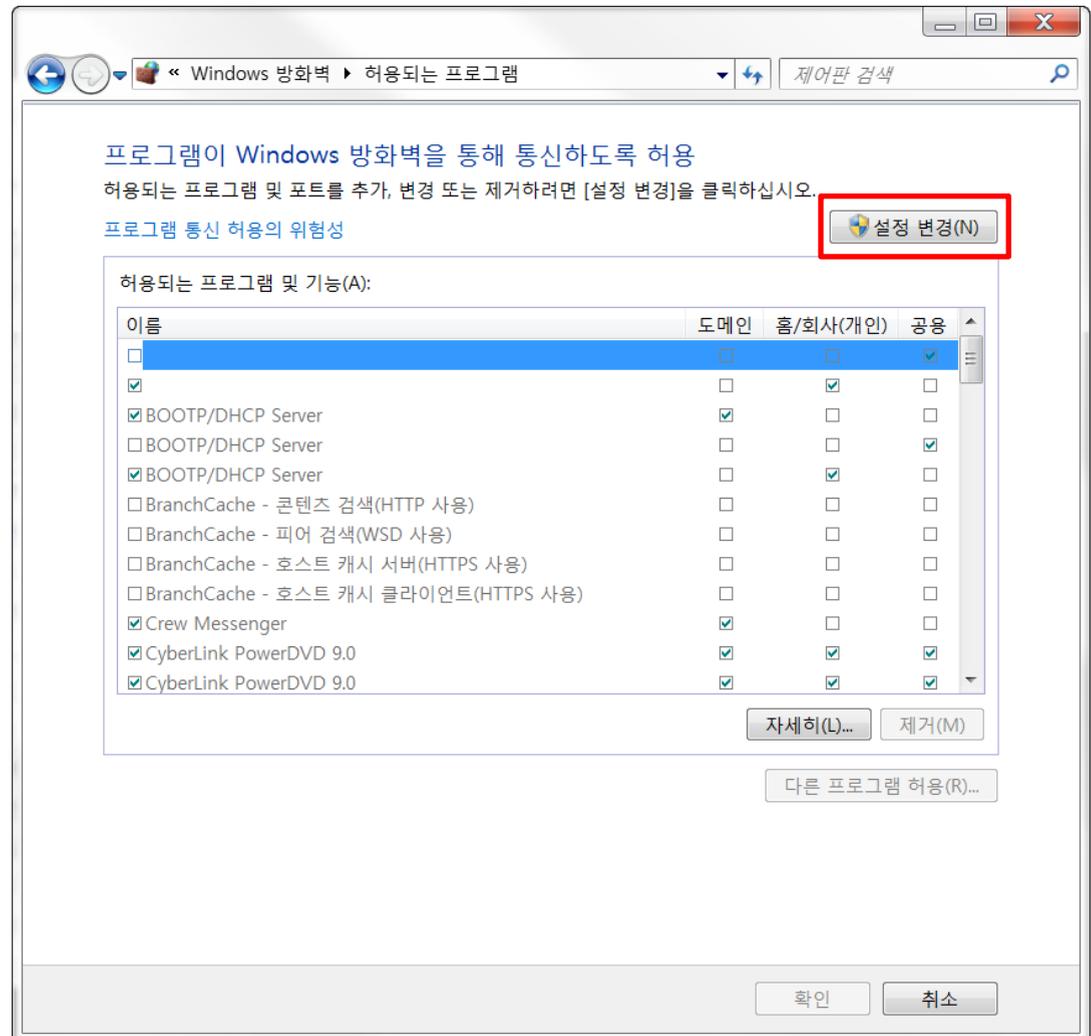


3. Windows 방화벽 창의 왼쪽 상단에 있는 'Windows 방화벽을 통해 프로그램 또는 기능 허용'을 클릭하여 실행합니다.

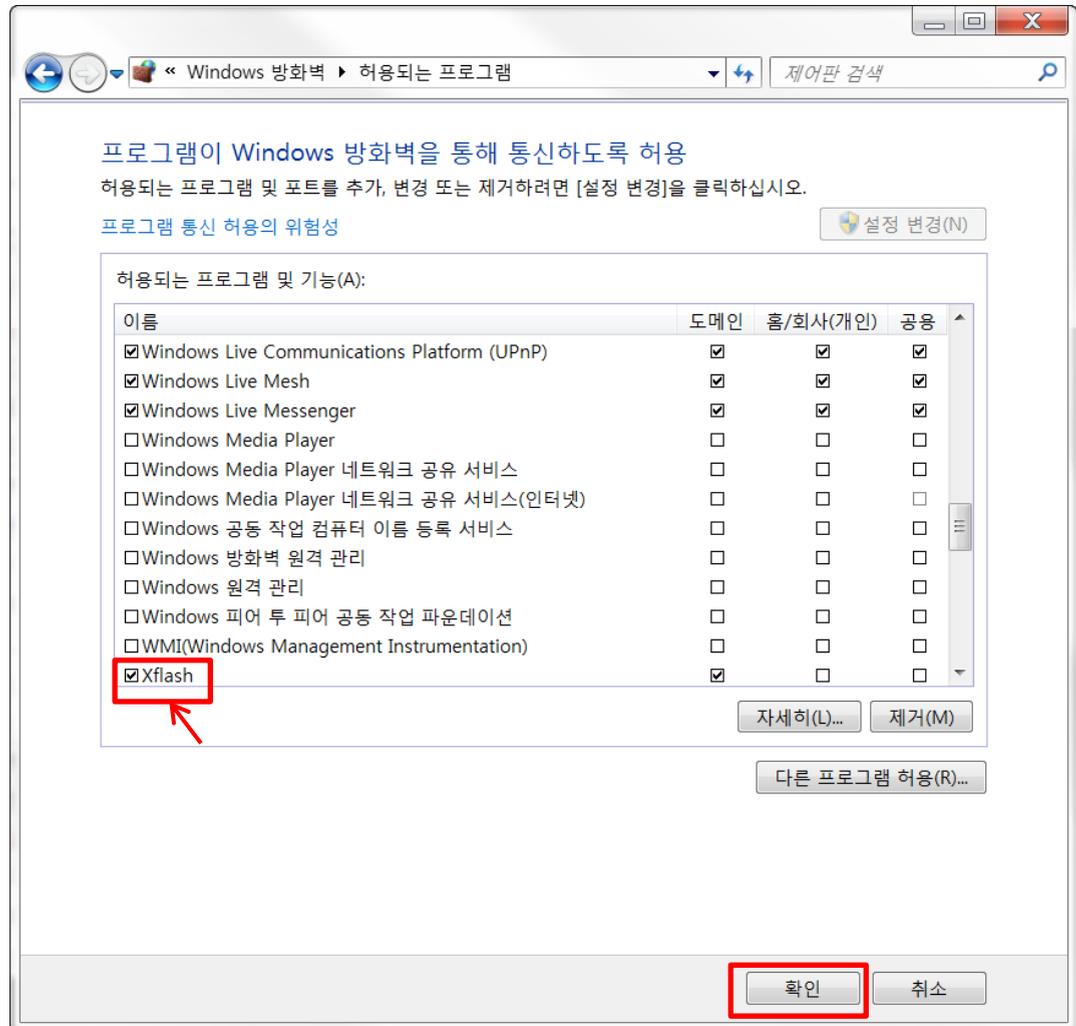
4.



3. 허용되는 프로그램 및 포트를 추가하기 위해서 설정 변경 버튼을 누릅니다.



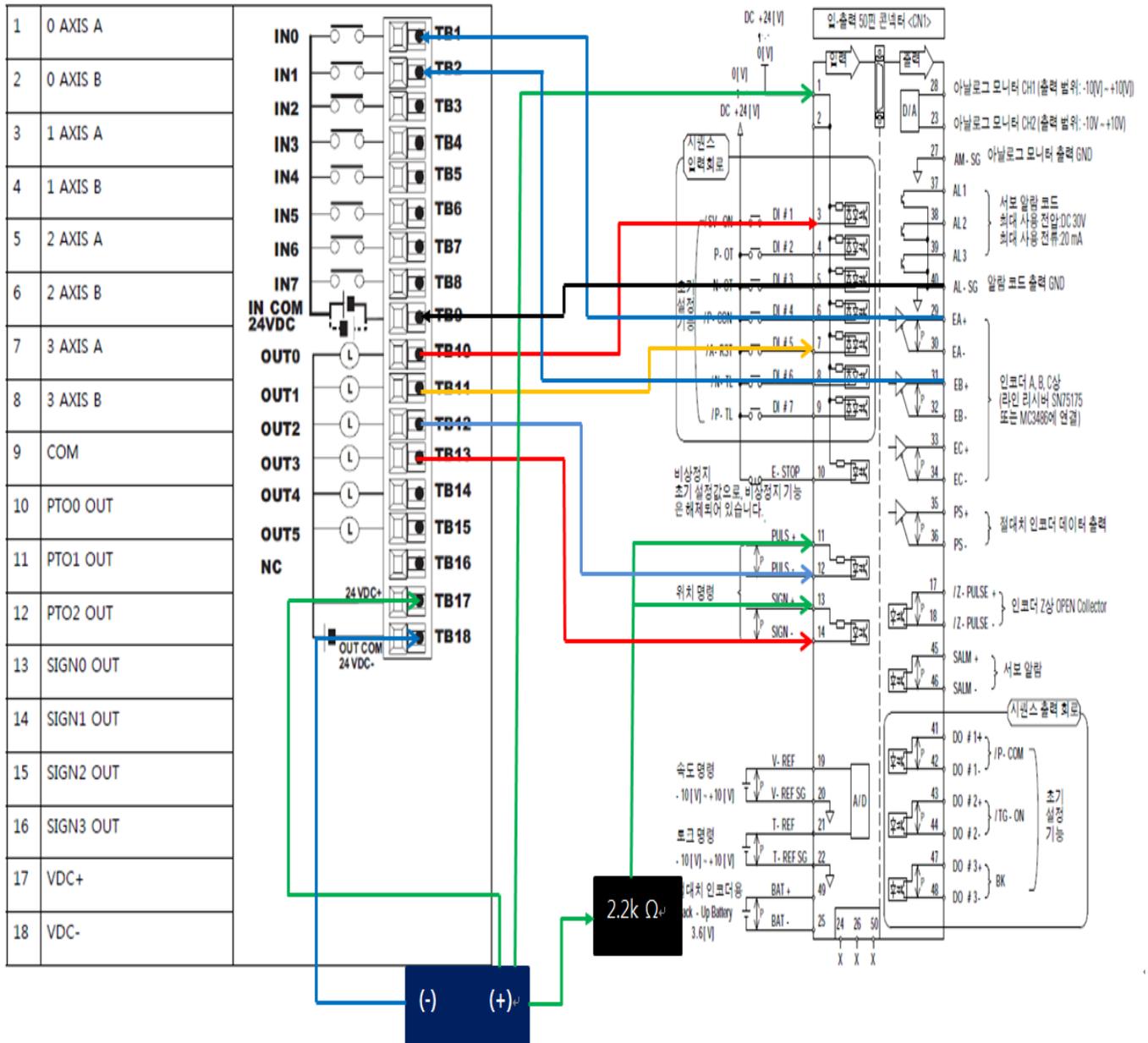
4. 설정 변경 버튼을 누르면 비활성화되었던, 허용되는 프로그램 및 기능의 내용들이 활성화됩니다. Xflash 프로그램을 찾아 클릭하여 check합니다. 확인 버튼을 누르면 Xflash 프로그램에 대한 방화벽 설정이 완료됩니다.



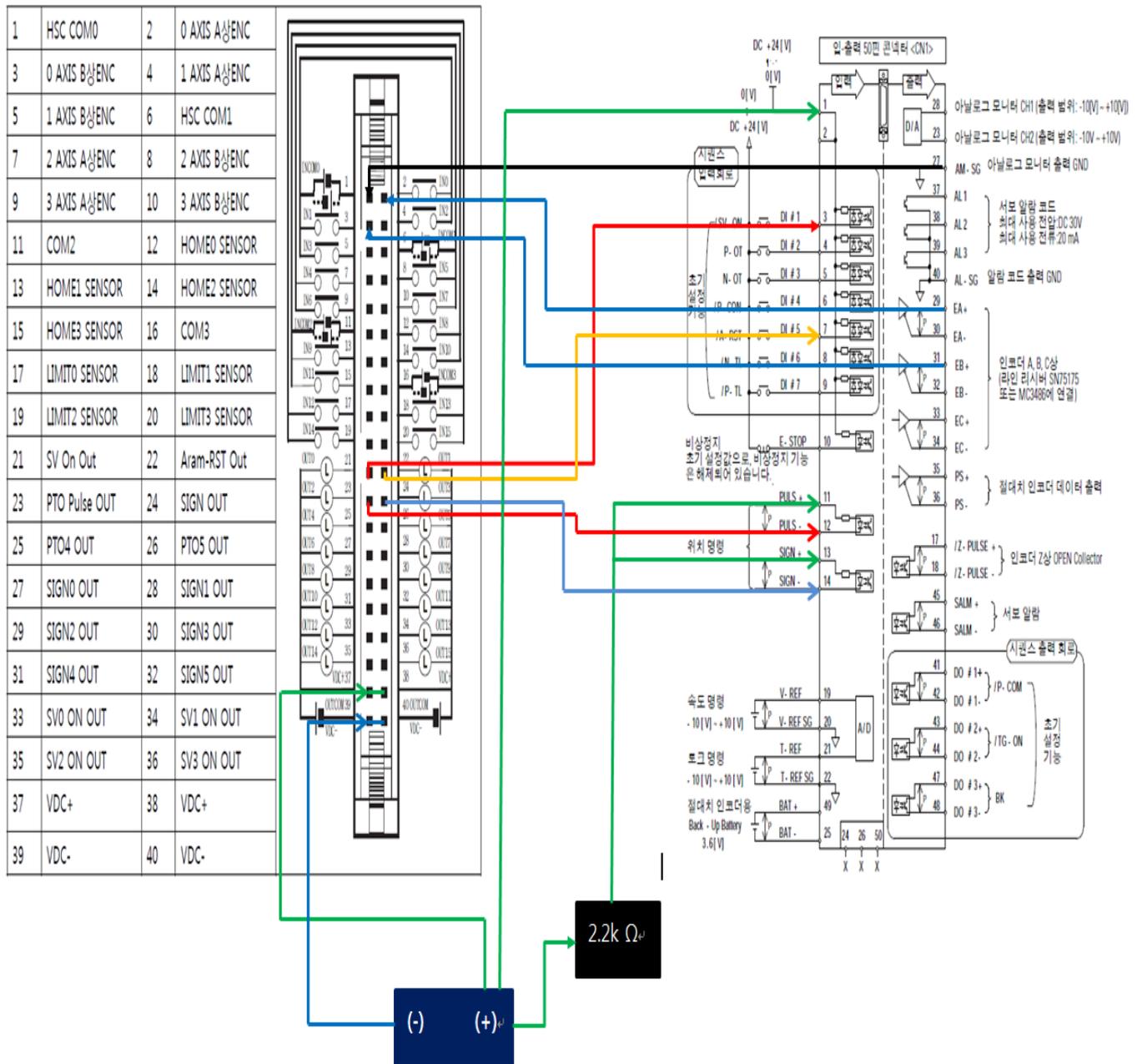
**\*참고사항 4. CSD3, CSD7 - X8 14DDT, 32DDT 결선도(CSD5는 CSD7과 동일)**

**1. X8-M14DDT to CSD3 Servo Position control**

X8-M14DDT에 연결되는 18점 Cable

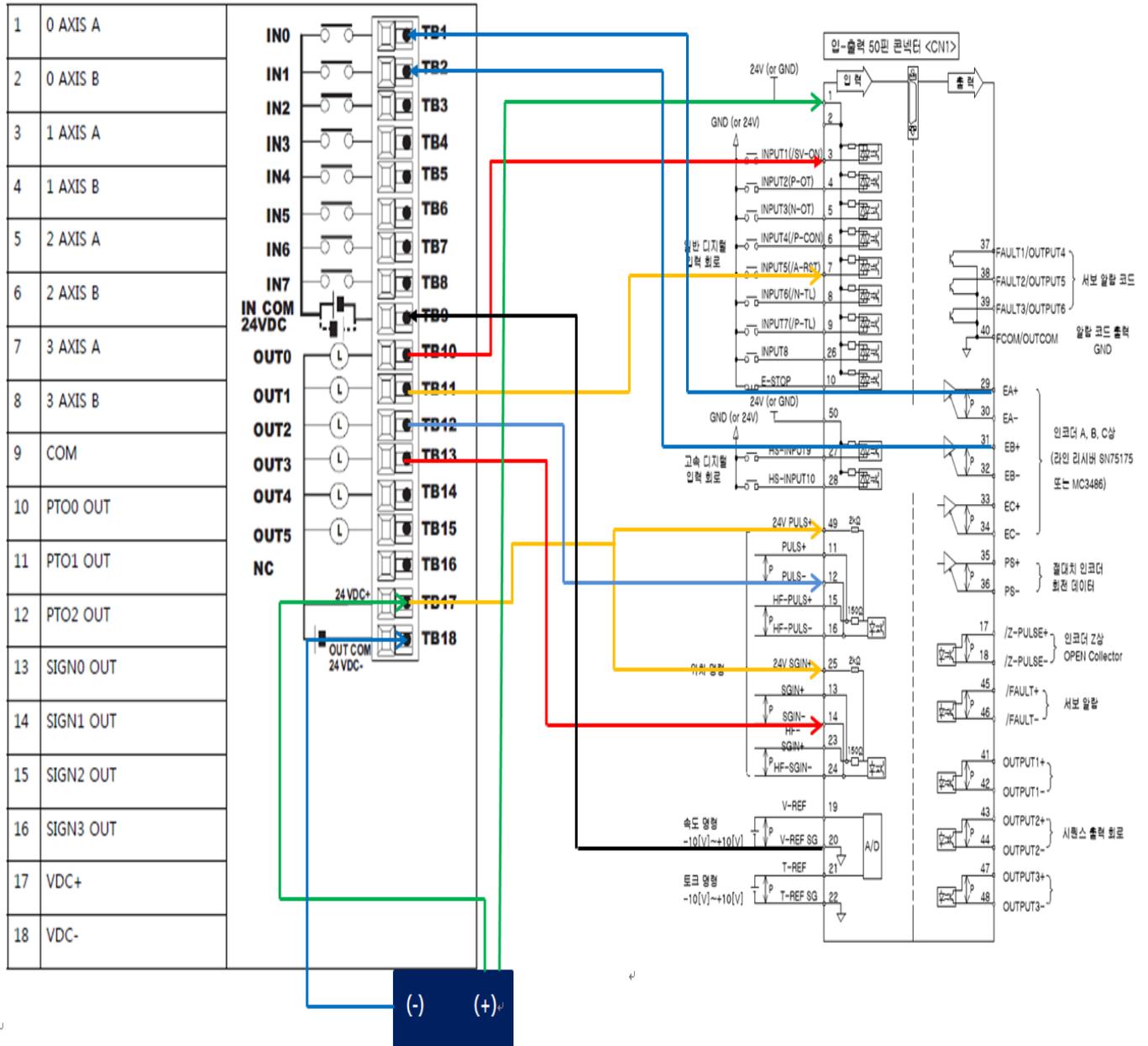


2. X8-M32DDT to CSD3 Servo Position control

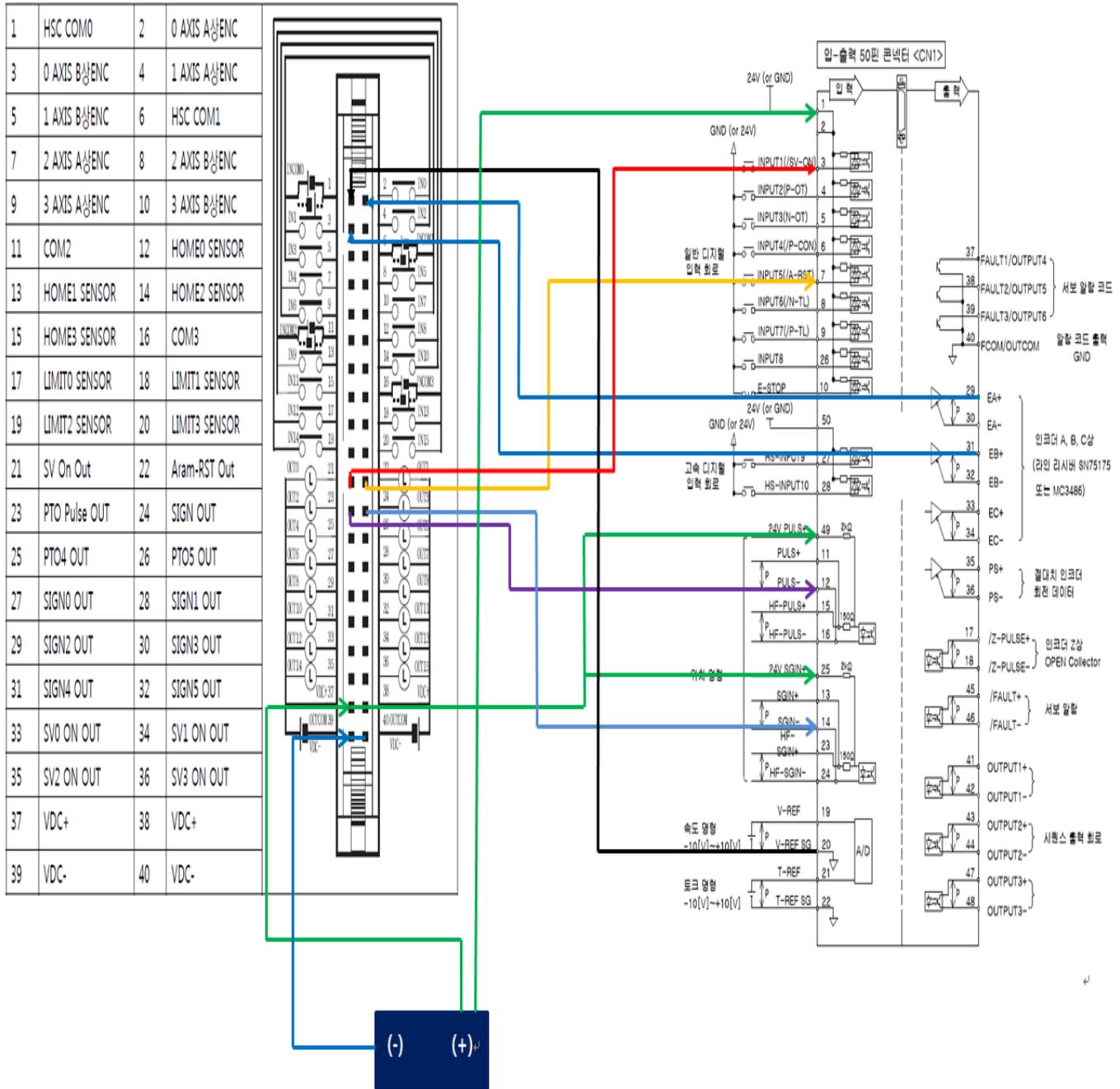


3. X8-M14DDT to CSD7 Servo Position control

X8-M14DDT에 연결되는 18점 Cable



4. X8-M32DDT to CSD7 Servo Position control



## 참고문헌

자동화 설계를 위한 plc nx series 입문 (nplus type) 한국기술교육대학교 산학협력단

자동화 설계를 위한 PLC X8 Serise 입문 (XGPC Type)

발행 2015년 5월 12일

발행인 알에스오토메이션

발행처 알에스오토메이션

문의처 경기 평택 진위면 청호리 진위산단로 38

Tel (031) 685-9300

Fax (031) 685-9500

<http://rsautomation.co.kr>

저작권법에 따라 이 책의 무단복사나 복제를 금지합니다.

## 알에스오토메이션주식회사

[www.rsautomation.co.kr](http://www.rsautomation.co.kr)

경기도 평택시 진위면 진위산단로 38 #17709

T. 031-685-9300 F. 031-685-9500

· 부산지사 부산광역시 사상구 대동로 303 벽산디지털단지 313호 #46981  
T. 051-329-7802/3, F. 051-329-7804

알에스오토메이션 서비스센터 전국 어디서나 **1588-5298**

· 본사센터 경기도 평택시 진위면 진위산단로 38 #17709

T. 031-685-9593(9594), F. 031-685-9500

· 부산센터 부산광역시 사상구 대동로 303 벽산디지털벨리 313호 #46981

T. 051-329-7802(7803), F. 051-329-7804

## RS Automation Co., Ltd.

[www.rsautomation.biz](http://www.rsautomation.biz)

38, Jinwisandan-ro, Jinwi-myeon, Pyeongtaek-si, Gyeonggi-do, Korea,  
Zip code : 17709

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

RS Automation Global Business Support  
[rsagbs@rsautomation.biz](mailto:rsagbs@rsautomation.biz)

京畿道平泽市振威面振威产团路38 #17709

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

RS自动化全球商户支持  
[rsagbs@rsautomation.biz](mailto:rsagbs@rsautomation.biz)